RESEARCH

An ICT Architecture for Enabling Ancillary Services in Distributed Renewable Energy Sources Based on the SGAM Framework

Armin Stocker[1*], Ali Alshawish[1], Martin Bor[2], John Vidler[2], Antonios Gouglidis[2], Andrew Scott[2], Angelos Marnerides[2,3], Hermann De Meer[1] and David Hutchison[2]

*Correspondence:
stockera@fim.uni-passau.de
[1]Computer Networks and Computer Communications, University of Passau, Passau, Germany
[2]School of Computing and Communications, Lancaster University, Lancaster, United Kingdom
[3]School of Computing Science, University of Glasgow, Glasgow, United Kingdom
Full list of author information is available at the end of the article

Abstract

Smart Grids are electrical grids that require a decentralised way of controlling electric power conditioning and thereby control of the production and distribution of energy. Yet, the integration of Distributed Renewable Energy Sources (DRESs) in the Smart Grid introduces new challenges with regards to electrical grid balancing and the storing of electric power, as well as additional monetary costs. Furthermore, the future smart grid also has to take over the provision of Ancillary Services (ASs) by synchronous generators. In this paper, a distributed ICT infrastructure to solve such challenges specifically related to ASs in future Smart Grids is described. The proposed infrastructure is developed on the basis of the Smart Grid Architecture Model (SGAM) framework, which is defined by the European Commission in Smart Grid Mandate M/490. A testbed that provides a flexible, secure, and low-cost version of this architecture, illustrating the separation of systems and responsibilities, and supporting both emulated DRESs and real hardware has been developed. The resulting system supports the integration of a variety of DRESs, with a secure two-way communication channel between the monitoring and controlling components. It assists in the analysis of various inter-operabilities and in the verification of eventual system designs. To validate the system design, the mapping of the proposed architecture to the testbed is presented. Further work will help improve the architecture in two directions; first, by investigating specific-purpose use cases, instantiated using this more generic framework; and second, by investigating the effects a realistic number and variety of connected devices within different grid configurations has on the testbed infrastructure.

Keywords: Ancillary Services; SGAM; DRES; Smart Grid

Introduction

There is an emerging decentralisation of electrical energy system infrastructures. The two main sources of this decentralisation are the generation sources themselves and the topological structure.

The widespread introduction of Distributed Energy Resources (DERs) as a new form of generation has caused a substantial phasing out of synchronous generators, which of course were the former main input source. However, the inclusion of DERs which are fed by renewable primary energy sources has many serious implications due to the inherently unpredictable nature of typical renewable energy sources.

These resources form a sub-class of DER referred to in this paper as a Distributed Renewable Energy Source (DRES).

From a topological point of view, the former centralised power grid, operated in a top-down manner whereby power flows consistently from upstream generators to downstream consumers. This directionality is now being challenged with small, distributed generation sources pressing the requirement for bottom-up operational control.

This bottom-up mode leads to many as of yet unsolved challenges, such as satisfying the information exchange amongst the participants, the structural changes, and the organisational changes. Relatively slow electromechanical dynamics of synchronous generators are replaced by the much faster dynamics of power electronics within smart converters with a much greater potential for flexibility.

The required smartness, flexibility and scalable aggregation of widely dispersed and heterogeneous power sources depend predominantly on a yet-to-be standardised ICT architecture with appropriate levels of abstraction and these abstractions themselves may not necessarily coincide precisely on a one-to-one basis with power grid components on any aggregation level.

The ICT system forms part of a cyber-physical infrastructure which will be key to resource optimisation for the services to be provided, both locally and globally. The architecture will necessarily need to tolerate faults (both ICT and generation related) and be resilient to IT-security challenges. There are many components and agents to be integrated and several views to be respected. A schematic overview of this structure is given in Figure 1, highlighting the heterogeneous nature of the components and the need for the ICT system to accommodate stakeholders with different business interests.

Designing and developing such a complex distributed cyber-physical system calls for a systematic approach; here this paper uses two main steps. First, the approach prescribed by the Smart Grid Architecture Model (SGAM) framework is used; this helps mitigate complexities as early as possible by focusing on proper interface definitions for all possible forms of interactions among all identified agents and levels.

Second, a quick-as-possible testing approach is enabled by developing an integrated testbed, by which we can validate whether our developing architecture fulfils the relevant system requirements, or if it needs further revisions.

Following this schema, leads to a stepwise design, starting with the component level of the smart converter interfaced DRESs, based on an amalgam of power electronics and Information and Communications Technology (ICT) components such as communication buses, Single Board Computers (SBCs), Programmable Logic Controllers (PLCs) and Digital Signal Processors (DSPs), at the edge of the overall cyber-physical system.

This paper describes these two stages without attempting to cover the full architecture. In particular the aggregation aspects of the cyber-physical distributed system and the accounting middleware are not included in their full scope in the present system.

With this in mind three main contributions of this paper are identified:

1 A first step in the modelling of an ICT architecture, specifically, for the Ancillary Service (AS) portion of the general smart grid which is presented in a structured way using the SGAM framework.
2 A prototypical implementation of an integrated testbed based on this description.
3 Mapping of the described architecture and the testbed to validate the architecture.

The remainder of this paper is structured as follows: The section Related Work is an overview of the current state of research, and how this work is different. The Designing a New Smarter Grid with SGAM section gives an introduction into the SGAM framework and presents a flexible ICT architecture model for AS enabled DRESs based on SGAM. Based on the ICT architecture model, the paper demonstrates a low-cost Testbed Prototype, which integrates real and simulated hardware in a flexible, secure, and low-cost way. Next, a Discussion about some of the challenges in building such an ICT architecture is included. Finally, the Conclusion show the main outcomes of this paper and outline future work.

Related Work

Taking into consideration the requirements and restrictions described in the introduction, there is a requirement for developing a flexible ICT architecture for AS in DRESs (Jindal et al, 2020) to ease their integration into the grid (Jindal et al, 2019) in a secure way (Gouglidis et al, 2018; Jindal et al, 2019). Existing work considering how DRESs can contribute to the provision of ASs only makes rudimentary considerations for the required ICT infrastructure. Two examples of this are in Kryonidis et al (2019) and in Yuen et al (2011).

In Kryonidis et al (2019) a control scheme for voltage regulation via reactive power is proposed. The control scheme is based on a central decision maker that collects measurements from the DRES in a Medium Voltage (MV) grid. Based on these measurements and grid topology an optimal operation scheme for each DRES is computed. However, the actual amount of reactive power that each DRES contributes to the service is determined in a decentralised manner based on the local information of the DRES. In Yuen et al (2011) possibilities for the provision of frequency control reserves by microgrids are explored. In particular this paper highlights the possibility of frequency reserves being request via a central controller managed by an aggregator, an entity that collects the individual contributions of DRES and offers the sum of these contributions to the DSO, or in a decentralised manner where each DRES is only configured on a longer time scale like weeks or months with a droop curve.

When dealing with optimisations on the medium voltage level as in the above publications it is important to note that common grid representations include the Low Voltage (LV) grids as the aggregations of their load and (renewable) generation. To realise a setpoint on an aggregated LV grid requires additional computation and communication. This paper aims to start working on overcoming this gap. To this end a structured formulation of an ICT architecture capable of handling the requesting, monitoring and control of DRES in order to provide ASs is shown. Special care is taken to show that the control and monitoring concepts can deal with multiple levels of aggregation.

In Kim et al (2011) a cloud-based system is presented which aims to enable demand response. They employed the cloud system to determine the cost-optimal distribution of demand response to the individual loads. Their findings show that this system is scalable by lowering the information burden required from the System Operator (SO). The architecture presented later in the present work aims to improve upon this by offering general purpose tools to perform multiple ancillary services. Additionally, a hierarchical organisation of the aggregation system retains the benefit of above method while allowing for the integration of Transmission System Operators (TSOs) and Distribution System Operators (DSOs).

The authors of Hammad et al (2019) present a control strategy to enable virtual inertia by batteries in MV systems. Their evaluation also considered the performance of the proposed system when varying communication delays are present. However, how realistic the presented communication delays are is not considered as the underlying communication infrastructure or intermediary aggregation levels are not shown.

The SGAM framework employed in this paper has been widely adopted by groups looking to build modern grid systems: In Messinis et al (2016) the authors examine the use of virtual power plants for providing demand response via modelling, which is aligned with SGAM. By first defining and partitioning their model into smaller functions and functional groups, they refine their approach. In their next step, the use cases built from these functions are used to define the components on the different layers of the SGAM framework. Their experience shows that this tool is useful in finding interoperability issues and thereby, helps in creating a more robust ICT infrastructure.

The goal to enable more frequent and complete data exchange between DSOs and TSOs is examined in Hooshyar and Vanfretti (2017). In this work the focus is on a monitoring system using phasor measurement units. As part of this, the required ICT systems and their mapping to physical components is shown using the layered architecture of the SGAM framework. Additionally, in Estebsari et al (2019) the use case of wide area monitoring of PhotoVoltaic (PV) systems is tackled using the SGAM cube with the aim to reuse many of the existing components in the model. While these works provide a deep dive into how the monitoring of the system can be achieved using their respective technologies, our work aims to give a more general view of what is required of a system to provide ancillary services. High fidelity monitoring of the grid state is only a part of these requirements.

In Radi et al (2019) a cloud-based infrastructure is created to enable bidirectional communications between the TSO and DSO. To validate this approach the mapping to the National Institute of Standards and Technology (NIST) interoperability layers and the mapping to the SGAM layers is examined. The main focus of this work is to show that such a data exchange is in fact possible and what standards exist to facilitate this exchange. To complete their approach, they present a specification of which data can be accessed by which party. Our work seeks to utilise such a data exchange to show how this data exchange allows for the provision of ASs.

The web of cells is another architecture presented in Luciano et al (2017). The grid is partitioned into non-overlapping cells. Each cell aims to be self-sufficient in its regulation of voltage and frequency. Only in exceptional cases is communication

required with other cells. While parts of the data exchange are detailed, Merino et al (2017) shows that to employ such a solution a novel communication system is required. Our work aims to present a hierarchical architecture which fills this gap by focusing also on ICT related aspects.

In Tian et al (2016) an architecture for management of microgrid communities is presented. They define a microgrid as a LV grid with distributed generation units connected. Further, the microgrid community is defined as a structure where multiple microgrid band together to coordinate the operation of their generation units. Utilising this structure, the paper proposes a control strategy to provide controllable active and reactive power generation or contribution to frequency and voltage regulation services. The proposed control architecture has both centralised and hierarchical control. On the one hand, the microgrid community controller is in charge of multiple microgrid controllers; on the other hand, each of these controllers is the central entity in charge of the respective part of the power system. The approach to structuring their ICT system for the microgrid communities is similar to the one described in this paper, which shows that such an approach is feasible. However, they assume that the DRES connected to their system are fixed in their operation and cannot contribute to the grid stability. Our model and testbed seek to show that this contribution of DRESs to grid stability via AS can be enabled.

The aspect of virtualisation is also shown in Leal and Botero (2021) and Leal and Botero (2019). These publications present an SDN-based architecture for communication inside a substation. Our aim is to present a flexible, virtual grid architecture but on a larger scale allowing to communication across the whole grid and not only within any one substation.

Sirviö et al (2021) gives an overview over the historical, current, and future of the smart grid. The focus is to highlight the viewpoints of different stakeholder and described their functional requirements on a high level. The present work takes a more concrete approach by presenting not only an architecture but also giving insight into aspects of its implementation.

A scalability analysis is conducted in Potenciano Menci et al (2020) for the centralised control of the grid via a virtual power plant. Their work presents a simulation setup which showed that their control system scales well with upwards of 100s of nodes. It is important to note that their work did not consider the links connecting the DRES with the VPP as critical links. With an increase in the number of connected DRES, the number of messages required to be sent from the VPP as well as the number of communications due to device failures increases. Therefore, the VPP can become a bottleneck. By introducing an aggregation hierarchy, the present work aims to reduce this effect.

The authors in Thornton et al (2017) build a hardware-in-the-loop testbed for simulating demand response. This testbed is specifically designed to simulate a small subset of the Internet of Things (IoT) communication, sensors, and actuation, whereby the demand response and grid are simulated in software via PSIM. This differs from our approach, where a testbed is built that reflects a full demonstrator for a real-world deployment, instead of a supporting a single type of AS, without any actual DRESs.

Prior work examines simulated environments bounded to discrete components (or sets of components) within the larger AS or DRES space. To gain a more

complete understanding of the complex interactions between these interconnected components, and to provide methods for simulating specific functions of an AS, a full-stack demonstrator with hardware in-the-loop inclusion is required. Using the capability of performing end-to-end test in this demonstrator validates our proposal for an ICT architecture for ASs that is missing in the related work so far.

Currently in the Nordic countries data hubs are being established by NordREG. These data hubs tackle management of energy trading for both retail and wholesale markets which can be seen as a precursor to the trading of ASs. Many of the lessons learned outlined in the report on the current state as of December 2021 Langset (2021) correspond to tasks tackled in the design of the testbed presented in this paper. The most important claim is that thorough end-to-end testing is required before deploying the system, further justifying the use case for the testbed presented in this paper. Additionally, it is stated that a cloud-based solution matching the approach of this paper is more cost-effective than an in-house solution and should be preferred. While on a design perspective it is highlighted that the roles of market players and especially third parties are important to consider for the systems design which the SGAM framework used as a tool in this paper enables the presented architecture to do.

Designing a New Smarter Grid with SGAM

To better understand the conceptual space in which the efforts presented in the previous section exist, requires to first examine the core idea; in this case, the SGAM framework. This section highlights some of the important components relevant to distributed control and monitoring. A general overview of the important parts of the SGAM framework itself is given in Appendix 1. The full detail can be found in the related standard Bruinenberg et al (2012).

As was shown in the related work section in the literature there exist a variety of approaches that build upon centralised control of the system and a hierarchical aggregation structure. The centralised control allows for the computation of optimal configuration of the DRES operation with high accuracy as several commonly used optimisation techniques can be applied in this setting. The hierarchical aggregation structure mimics the hierarchical structure in the grid where the different voltage levels are separated by transformers substations. An architecture approach that utilises this paradigm therefore, closely resembles the existing structures in the power grid of today. In this section a simplified version of an ICT architecture SGAM model for an hierarchical structure with centralised control, but distributed provision of AS is presented.

Business Layer

As a start a brief stakeholder analysis of the envisioned system is presented. First, this analysis considers the SOs which from the perspective of the AS serve as the customers. Then the resources which are used to provide these services are considered and finally, additional third parties are taken into account.

The role of the SO is encapsulating the needs of the power system side of the overall system. The goal is to have a stable grid which is operated efficiently i.e. with a minimal amount of losses. Therefore in our context, the SO fills the role of the customer seeking certain grid services.

The role of the third party is to perform the aggregation and optimisation required to provide the services to the SOs. This role is introduced to provide the system with flexibility regarding future market or regulatory developments. As such, an actual third party may be a subsidiary of a SO or a completely separate entity. This party also serves as the broker between a potential large amount of connected DRES and a small number of SOs consuming services.

Distribution System Operators (DSO)
DSOs are charged with managing both the medium and low voltage distribution grids, supplying the end consumer with electricity directly. In this case, their primary challenge is the reversal of power flow in situations where end-consumers also have some generation capacity, such as on-site solar installations. This in turn leads to concerns around voltage spikes as power from these sites may spike unpredictably causing overly high back-feed into the low voltage grid. Traditionally, these problems are solved through the implementation of grid reinforcements, preventing the propagation of these problems upstream.

With the introduction of smart grid architectures, the DSOs can take direct action to control the power feeds and flows, enabling them to become an active component in the management of the grid, rather than simply a passive consumer of higher-voltage ingress. This involves the integration of vastly higher fidelity energy monitoring in the form of smart metering, and smart converter devices. These two technologies place the DSO in a prime position to execute very fine-grained control over their grid. This has real, tangible benefits as it allows the DSO to reduce their dependence on upstream power generation when dealing with the management of reactive power, line losses and the bidirectional power flow with renewable injection.

Transmission System Operator (TSO)
TSOs, on the other hand, are responsible for maintaining the balance of power in the grid, and have control over the high-voltage power lines transmitting power from traditional bulk generators over long distances where the connections are made with the DSOs to step down into the medium-voltage grids. This naturally has wide-reaching effects, as balancing supply and demand at this scale will have knock-on effects with bulk generators and DSOs supply alike, and any deviation in this balance will immediately become apparent in globally monitored parameters such as grid frequency. As such, ASs dealing with grid inertial response, primary frequency response and fault currents (and fault mitigation) are required to be provided at this scale to keep the grid operation stable and safe.

DRES owners
Finally, the DRES owners are the people actually providing contributions to grid services via their DRES. This term generalises installations containing different types of energy resources. Batteries, capacitors banks, super capacitors, controllable loads, Electric Vehicle (EV) charging stations, PV systems and many more resources can all provide valuable contributions to controlling the operation within limits of the power grid. The variety of generation sources implies also differences in the business goals of the DRES owners. As an example, the owner of a large scale PV

system might be solely focused on maximising their profits to increase the returns on their initial investment. Owners of rooftop PV systems for their own home might primarily be interest in optimising their self-consumption and only sell excess energy to the power grid. Finally, owners of EVs and charging stations equipped with PV panels may want to ensure their transportation is available for their trips to and from their workplace.

In order to simplify the model, it is assumed that these considerations can be handled on a local level. This means especially that when a DRES owner reports the capacity to provide ASs of their installation, their own goals are already accounted for.

Third Parties

Several other parties have a interest in the energy system coupled with an ICT system. Regulators and law makers give restrictions on the operation conditions to be ensured and the framework for trading of energy and services between the SO and DRES owners. Regulatory policies in the future might forbid the SO from being the consumer of ASs and at the same time selling the aggregated capacities of the devices in their respective grids to other SOs. Further, the trading of ASs offers a new business opportunity which previously not involved parties might attempt to seize. This might lead to them becoming an intermediary that contracts individual DRES owners to sell their aggregated capacity to SOs. Finally, cloud service providers might be integral to setting up and managing the communication links to the distributed components in the grid while offering a platform for other parties to host their required services on.

Role Model

The main focus of the present architecture is the monitoring and control of ASs on a moment to moment basis. As such considerations towards the law makers and cloud service providers are less relevant as these need to happen in advance to any service provision. Further, as already mentioned when talking about the DRES owner the multitude of different DRES types is resolved on a local level. With this only one actor is created in the model to serve the role of providers of ASs.

As was just described the DSO and the TSO both seek to have stable operation of their respective grid levels. However, the services they require and therefore request from the system are different. This difference is assumed to be minor enough that a general actor, the SO, serving as the client who request any AS is sufficient.

To model either the separation of concerns within a SO or the integration of other third parties serving as aggregators of ASs a separate role is introduced. This third party has the task of controlling and coordinating the grid-wide operation of the DRES they are responsible for. Their business interest is two fold: On the one hand, they want to earn money from their aggregation of services i.e.; they seek to optimise the operation of DRES with respect to monetary gains; on the other hand, as a requirement from the side of the SOs, they have to ensure the operation of their DRES stays within permissible grid limits.

As a final consideration to satisfy the business interests of the described actors, the existence of a mutual contract between these three actors, the third party, the

SO and the DRES owners, is assumed. Models for these contracts exists when it comes to trading of wholesale energy as can be made evident by considering the operation of companies such as Kiwi Power Kiwi Power (2022). When buying energy on the wholesale market the customer expects that the requested amount of energy is fed in during the request period. The trading of ASs is different however in this key aspect. When requesting an AS the customer expects a certain behaviour to be present when a contingency in the grid arises. On the one hand, this means that the DRES is uncertain when exactly it must provide the service; on the other hand, this raises the importance of the requirement to keep track of the state of the DRES as it must be available otherwise critical emergency resources are lacking. This changes the requirements in a way that the existing contracts may not translate on a one to one basis. As there is no regulatory framework for this exact relation to the best of our knowledge until now, the exact details of this contract are intentionally left vague. However, it is assumed to serve as the contractual basis that regulates the provision and remuneration of the different ASs.

Function Layer

The high level goal of the system of providing ASs requires a set of services to be available.

First, the participants in the system need to be aware of the available resources and the system state. This is enabled by a service referred to as monitoring. Considering that each stakeholder has different business interests they may be concerned with different monitored values. Furthermore, monitoring needs to be split in monitoring done for a human operator and monitoring done for a software system to ensure the proper operation of the DRES.

Second, once the SOs are aware of the grid state and the available resources, they are in a position to formulate requests for certain ASs. These request need to be mapped in an appropriate way onto the DRESs available to contribute to the respective AS. Usually, it is not enough to just forward these request but an optimal distribution to the DRES is desirable. The optimisation employed should consider grid constraints and economic benefits for the actors. As the goal of this paper is not to develop novel optimisation algorithms, further details on the design of such software is not included.

Finally, after the system has received a request, processed it to configure the DRES and these devices have provided the respective AS, the DRES owner expects to be remunerated for this in some way. To enable the cashflow between the participants, trustworthy accounting is required. To this end considerations with regards to non-repudiation, reliable metering, and storage of data, as well as storage solutions offering enough throughput and disk space are required. As mentioned in the Introduction the full scope of this function is not included in this paper. Instead we focus on structuring the information exchange from the meters to the storage system and leave the details with regards to the structure / protocol for this information exchange intentionally open.

The services described above are the functionalities considered for the function layer. From a functional point of view the system must be able to do the following: Take an request from the SO and send it to the third party. In order to determine

how this request is to be realised the third party should have monitoring data about the grid available. Using the available data a optimal usage of the available contributions from the individual DRES is computes and communicated to them. Once the signal has reached the DRES each of them can change its local behaviour to provide the service.

This functional decomposition of the system is also shown in Fig. 2. Starting from the top in this figure the interface to the SO is the function *User Input Handling*. As this represents the front end that the SO interacts with, it is located in the market zone and distribution domain. It is assumed that a SO knows which ASs with which parameters is required. As such this function provides to them the possibility to enter this demand into the system. The sum of their AS requests is forwarded to the third party where this request is then translated into a provision of the service via the available DRES.

Next the role of the third party is split into three functions: *Monitoring*, *Setpoint Communication* and *Optimal Setpoint Computation*. These functions aggregate the behaviour of the DRES located in the distribution grid to a distribution grid level service. Therefore they are located in the distribution domain and operation zone. The computation of optimal setpoints and setpoint communication happens in multiple aggregation levels as mentioned at the start of the section. Take as an example a medium voltage grid which is connected to many different low voltage grids. This structure would indicate two hierarchy levels. First, the optimal setpoints for all the low voltage grids in the medium voltage grids can be computed. Then these setpoints can be sent to these low voltage grids and inside of them they can be dispatched again by a similar procedure to the individual DRESs.

Finally, the functionality the DRES owner is responsible for is twofold. Firstly, the local control of a DRES and the actual provision of contributions by low level controller and actual hardware is modelled by the function *Grid Service Provision*). This function is located in the DER domain and spans the process and field zones since it involves both the control of the generation hardware and the generation hardware itself. Once a control signal has reached this function the appropriate changes to the settings of the DRES are made to provide the required response. Secondly, the management of the DRES is modelled by the function *DRES Local Control*. This function deals with the coordination of the different devices inside the DRES. As such any inputs received from the *Setpoint Communication* function need to be translated to appropriate output to the *Grid Service Provision*. Considering the flow of monitoring data from DRES to *Third Party* within this function the business goals outlined on the business layer for the DRES owner are to be resolved. This means that when reporting the available resources for ASs an appropriate amount of the actual resources is reserved to ensure these business goals are met. This function represents a form of operational control. Thus, it is located on the operation zone of the DER domain.

Information Layer

When considering communication links there are four different sections of the infrastructure. The link between the user and the system, the internal communication links within the system, the communication link from the system to the DRES

gateways, and finally a communication link between the DRES gateways and the different components of the DRES. These four links are what is described further in the following paragraphs from the viewpoint of the information and communication layer.

From the viewpoint of the information layer the message exchange between the user and the system is modelled as requests for collections of services. Then, the message a user sends to the system is a collection of requests for individual AS with the corresponding parameters set by the user. Thinking in terms of a collection of services has the advantage that it closely mirrors the utility provided to the system by traditional synchronous generators. Synchronous generators do not only provide a single AS like inertia but also simultaneously may contribute to the reactive power balance of the system, inject the required high currents during faults and many other ASs.

For the message exchange within the system there is a degree of freedom still left in the model. The hierarchical aggregation schema along the hierarchical optimisation impose different requirements on the information objects for determining the available amount of each AS (aggregation) and for a given request computing a effective dispatch (optimisation). Further, each AS has different requirements when it comes to the involved information. It is therefore required to come up with flexible formats and protocols for both aggregation and optimisation. One possibility solution is the use of loosely structured data objects like for example JSON provides. From an implementation perspective this has the benefit that the code for managing the message exchange can be the same and only the pieces of code for (de-)serialising to JSON objects need to be created. Additionally, this reduces the required effort to include new ASs with new optimisation and aggregation procedures.

For the aggregation and optimisation schemes the DRES serve as the smallest quantity one can talk about. As such the DRES gateways are the final smart entities involved in these processes. Therefore, the same format for data exchange as with higher levels of aggregation is suitable for this link.

For the final link between the DRES gateways and the DRES themselves, one has to keep in mind that in a future smart grid different types of DRES with different devices connected to them which are produced by different vendors will be the norm. As such the DRES gateway has the important additional task to translate the information objects received in an appropriate manner to fit with the information objects required by the different installations. As an example the received JSON objects may be required to be translated to appropriate Modbus registers. While doing so one has to keep in mind things like the number of bits available for each data point i.e. the precision to round to.

The figure for the information layer largely coincides with the communication links shown in Fig. 3 as blue lines. Therefore the figure is omitted.

Communication Layer

For the link between the DRES gateway and the DRES shown in Fig. 3 multiple communication protocols need to be supported. Which protocol is to be supported depends on the interface that the smart converter offers. The DRES gateway therefore was introduced to serve as the mediator between the ICT system and smart

converter. This flexibility with regards to the employed protocol allows for converters from different manufacturers to easily be integrated in the system.

Apart from this connection, two different technologies for the communication links are used. In order to communicate the dispatch of an AS to individual DRES installations point to point links between the optimisation and the individual DRES are required. These links must be able to reliably deliver messages for different ASs that can differ greatly in the required parameters. In simple cases, the payload can only be a Boolean value which needs to be communicated. In other cases, a set of numbers indicative of total energy amounts or to be interpreted as a droop curve needs to be sent. Standard web technology, like HTTP, is suitable for dealing with this task.

The second protocol is to deal with monitoring requirements. In order to take an optimal decision inside the ICT system up to date measurements of some DRES parameters are required. For example, take the state of charge of a battery system or available active power from the primary source of a DRES system. When deciding if and to what extent a DRES can provide an AS these information must be available. Further, optimisation algorithms for different AS may be interested in different measurements from different device. For this reason, a publish-subscribe scheme allowing for a flexible distribution of the measurements to the interested parties is proposed.

Considering the requirements outlined in this section the OPC-UA protocol described in Lehnhoff et al (2012) also offers the required capabilities. On the one hand, this technology offers to create sessions between a client and a server using HTTP to exchange variables as required for the setpoint communication; on the other hand, the subscription mechanism it offers is suitable to allow for a publish-subscribe scheme to be implemented. This system further offers the additional capability of implementing events and alarms to notify listeners of imminent changes in the production of the DRES.

The figure for the communication layer coincides with the communication links shown in Fig. 3 as blue lines. Therefore the figure is omitted.

Component Layer

A possible simplified technical realisation in software and hardware components is shown in Fig. 3.

For the user input handling some hardware owned by the DSO is required to run or access the frontend of the system. This DSO hardware needs to be connected with the third party hardware. A set of virtual entities is located on this third party hardware. These virtual entities are organised hierarchically to facilitate the hierarchical optimisation described in the function layer section. Thus creating a virtual representation of the actual physical entities in the power grid and their aggregation hierarchy.

To this end an exchange of information between the virtual entities for higher and lower hierarchy levels and from the lowest level virtual entities to the physical DRES location is required. In order to enable this communication a directory service is envisioned allowing each virtual entity to lookup the communication address of other virtual entities and DRES.

The virtual entities and the directory service together realise the setpoint communication function. Further, the virtual entities alone are enough to realise the monitoring through collection of data received from the DRES and aggregating this information towards the top of the virtual entity hierarchy.

Finally, a service performing the optimisations is required. Virtual entities which represent an aggregate, can provide measuring data to this service, and receive the optimal allocation of contribution to the virtual entities they aggregated. As such the optimisation service realises the *Optimal Setpoint Communication* function.

In order to realise the *AS Provision* function two components are involved: the smart converter controller, as the controlling device for the DRES generation hardware, and the DRES generation hardware itself. The DRES generation hardware can for example be a smart converter connected to a battery system or a PV system. It is responsive to certain changes in its parameters made by the smart converter controller.

Inside this smart converter controller real-time sensitive processing of data and determination of operational setpoints for the DRES hardware is done. It also forwards the required measurements from the DRES hardware to the gateway. The job of the DRES gateway is to translate the signals received from the third party hardware to a format that is understood by the smart converter controller and sending measurements taken from the DRES back to the third party hardware.

Testbed Prototype

For the EASY-RES project, a testbed designed to compare, and validate the general performance of designs based around the SGAM model, is developed. It further serves as the platform for building a full-stack demonstrator, paving the path forward to a flexible ICT architecture for ASs in the Smart Grid. The testbed allows to gather hands on experience with DRES communication, assess the orchestration and performance of communication, along with the impact of virtualisation and containers.

Given the aforementioned motivation, the following four requirements for the testbed are designed:

- Low Cost: DRES testbed setups are often expensive due to the use of High Voltage (HV) lab equipment. Being able to set up a testbed on a low budget greatly expands the opportunity for validation and reproducibility in this research field. Naturally, this does require some parts to be emulated at various levels of fidelity depending on the model and the compute power available. As the testbed is focused on prototyping an ICT architecture, the emphasis is on building a communication network, for which relatively inexpensive SBCs like the Raspberry Pi is used.
- Layered Approach: The testbed software stack should be built on interchangeable layers, aligned to the layers of the SGAM framework. This makes it easy to swap out individual components through following a modular design. This in turn also promotes easy transfer of services between hosts, as they are not tied to a physical device, which gives a higher fault tolerance.
- Secure: ICT security is essential, and cannot be left as an afterthought, especially for critical infrastructure like the electrical grid. With the testbed also

functioning as a prototype, security should be a first-class citizen included from the outset, and not bolted on. While this makes it more involved to develop software, it enforces a security-first mindset and promotes the hardening of software designs.

- Flexible: The testbed should be flexible to scale in a number of dimensions based on required resources; be those financial, compute, size, or others. Furthermore it should be easy to add real hardware as hardware-in-the-loop simulation has been proven to be much more accurate (Barragán-Villarejo et al, 2020; Thornton et al, 2017). Using a layered approach helps in facilitating this flexibility through the concretion of specific interfaces at the points where layers meet.

Testbed Setup

Given the requirements set out in the previous section, a geographically distributed testbed is developed, using modern container technology to provide a flexible ICT infrastructure which serves as the base of our work. The parts of services of the testbed are easy interchangeable between software emulation and real hardware implementations through strict message-based interfaces between the layers.

Hardware Stack

Figure 4 shows an overview of the physical components of the testbed, as implemented on three geographically distributed sites, and a centralised Manager. This particular setup is a snapshot of one instance of the testbed.

Site A is a research lab, which consists of all physical assets: an experimental DRES, with an experimental controller, monitored by a PLC (Schneider Electric Modicon TM241CEC24R), and a Human-Machine Interface (HMI) (Schneider Electric Magelis HMISTW6400) for local monitoring and control. The gateway is a Raspberry Pi 3 SBC, which facilitates a secure connection to the central Manager. The information sent to the Manager are the metrics of the DRES, as measured by the controller and send via the PLC, and set points from the Manager back to the controller. The PLC communicates via Modbus TCP to the Raspberry Pi, and via Controller Area Network (CAN) to the Controller.

Site B consists of one Raspberry Pi 3 SBC that acts as a physical gateway, and also runs an emulated versions of the PLC, controller, and DRES as present at Site A.

Site C is similar setup as Site B, but consists of four Raspberry Pi 3 SBC, each acting as a physical gateway, and also running an emulated PLC, controller, and DRES like Site B.

The Manager acts as a central hub, receiving all metrics and telemetry information from the distributed energy sources, running optimisation algorithms, sending control signals and set points back, and managing the deployment and health of software and network infrastructure. In our testbed, the Manager runs in two geographically distributed clusters of Virtual Machines (VMs). One cluster is provisioned via VMware vSphere, the other cluster is provisioned via OpenStack.

Software stack
Figure 5 shows an overview of the software stack on the manager and one (virtualised) gateway node. Both the Manager and the Gateway node run various services, some directly on the host operating system (as indicated by red boxes), or in a container (as indicated by blue boxes).

Containers Both the Manager and Gateway use the Docker runtime to run most of the network services in containers. Running services in containers has several advantages.

- A container gives a known, well-defined environment, allowing the easy deployment on any system (with the right architecture) that runs a container runtime.
- Containers and container-centric design facilitate the horizontal scaling of software within a cluster; instances can be easily duplicated to handle additional load, for example.
- Compared to other virtualisation techniques like virtual machines and emulation, containers run with minimal overheads, relying on the underlying operating system to provide the process isolation (Gerend et al, 2019). No work is required to replicate the operating system (like with virtual machines), or outright translate the machine instructions (as with emulation).
- Containers are an immutable snapshot of a particular service setup that can be versioned, which allows for the auditing of changes between versions, and facilitates "rolling back" to previous (known working) versions in case of failures.

Overlay Network All network communication in the testbed is done via an overlay network. Each node is connected to the Manager via a WireGuard VPN[1], creating a secure communication link for each node over the wider (insecure) Internet, ensuring that only authorised nodes can communicate with the server and other authorised nodes. WireGuard as chosen as it is faster, leaner, and more performant than the more commonly used IPsec or OpenVPN (Donenfeld, 2017).

Container Orchestration The containers running on the nodes are organised via Hashicorp Nomad. Nomad[2] is a workload orchestration engine that takes care of the deployment, execution, and halting of containers on all nodes in the cluster. The Manager acts as *server* node, while all other nodes act as *client* node. The server node monitors the health of each node, and can restart containers in case of failures, or report an error to the system administrator.

System Management To further help the setup and maintenance of the testbed itself, Ansible[3] is deployed, which is a provisioning, configuration management, and application-deployment tool enabling infrastructure as code. This approach allows to programmatically and verifiably manage all nodes in the network, as opposed

[1] https://www.wireguard.com
[2] https://www.nomadproject.io
[3] https://www.ansible.com/

to manual maintaining each node. Ansible uses the SSH services running on each gateway node for access.

Vault provides each service with the proper and up to date credentials (like SSL certificates) to enable secure and authenticated communication between the gateway node, the manager, and other systems in the network.

All these services work together to maintain the operation of a manager and gateway node. Further services, such as the MQTT agent and OpenEMS[4] Edge service, are used to facilitate DRES operations.

DRES functions consist mostly of sending metrics such as active power, frequency and supplied ancillary service function upstream; along with receiving commands from cloud-based controllers.

Application Communication Infrastructure For our communications infrastructure, we use MQTT, a lightweight asynchronous publish-subscribe network protocol. All relevant metrics are sent (published) by the MQTT agent to the MQTT broker, which then forwards them to the relevant receivers (subscribers). We further employ Telegraf[5] to receive the metrics and store them in a Time Series Database (TSDB), InfluxDB[6]. Grafana[7] provides an interface for viewing the data stored in InfluxDB in interactive visualisations.

System and Communication Overheads

While the addition of an orchestration layer on the testbed naturally increases the workload on the cloud services and edge compute devices (the Raspberry Pis), the actual effect on the systems used is minimal. Containers present a minimal processing and memory footprint, with the focus on securing access to resources on the host through kernel name spacing mechanisms rather than full virtualisation.

On the cloud host machine used for the testbed, the overall CPU usage for the docker daemon and its associated sub-processes amounts to under 0.7% of the available compute time, along with 0.4% of the host memory, including all coordination and orchestration tasks of the other devices. On the Raspberry Pi hosts, the CPU time is less than 0.1% (it actually reported as zero, as the tools available have only 1 decimal place of accuracy), while the memory usage peaked at around 5% with the devices not running any services beyond the core swarm communications and management.

The light load is also reflected in the minimal communication *jitter* seen when performing bandwidth tests between devices in the swarm. Jitter is a measure of deviation from an expected periodicity for a particular operation, in this case, sending or responding to sent data as part of the bandwidth test.

If the devices were under high load or performing particularly compute- or network-intensive operations, it is extremely likely this would be reflected in an increased deviation in packet timeliness. With the base testbed configuration, this results in a packet jitter of far below a millisecond for edge-to-cloud communications

[4] https://openems.github.io/openems.io

[5] https://www.influxdata.com/time-series-platform/telegraf/

[6] https://www.influxdata.com/products/influxdb/

[7] https://grafana.com/

as shown in Figure 6, and under 3 milliseconds for edge-to-edge communication as seen in Figure 7; this is despite using docker overlay networking through the WireGuard VPN, all of which is transmit through the wider internet.

Throughput figures for the swarm network are also fairly performant. The edge devices were connected to a 1000 Mbit/sec switch with a shared 56 Mbit/sec upstream connection to the internet, but are limited by the Raspberry Pi hardware itself, as the network device on the Raspberry Pi is only capable of 100 Mbit/sec. With this network configuration, the Raspberry Pi hosts attain an average throughput of approximately 34.5 Mbit/sec when communicating with the cloud host in both directions (both sending to and receiving from) which is drawn in Figure 8, and an average of approximately 27 Mbit/sec when communicating with other Raspberry Pis which is plotted in Figure 9.

Mapping SGAM with the Testbed

This section highlights how components located in a given layer, zone and domain of the SGAM plane are mapped into the testbed. First, how the testbed can aid in the realisation of certain layers is considered followed by a discussion of the components located in one of the domains across the distribution and DERdomain.

The testbed should offer a platform to implement a prototype of the EASY-RES ICT architecture, and when considering the description of the ICT architecture SGAM model described in Designing a New Smarter Grid with SGAM this means it needs to realise certain components of the proposed model. The mapping of the components mentioned in Figure 5 to the SGAM model is shown in Figure 10 which will be detailed more in the following.

Mapping by Layer

Communication and Component layer

From the standpoint of the communication and component layers, there must be certain components in the testbed which have communication links between them. On the one hand, this is fulfilled by including real hardware in the testbed; on the other hand, the virtualisation tools in the testbed offer the capability to design a stub which services as a replacement for the hardware component in tests. This approach for virtualisation was used for instances of the PLC and DER controller in the DER domain and Field / Process zones.

Information and Function layer

Moving to the information and function layers, there must be certain applications realised in the testbed as a set of processes which interact among one another (functionality), and this interaction is characterised by certain standards (defined information objects). This strongly, depends on the application in question of course. However, the flexible virtualisation architecture allows for arbitrary processes to be run on different machines connected to the testbed. This makes it straightforward to design and implement backend services, realising the functionality required for optimisation, which are run on the cloud machines and edge services, realising the local process of a DRES, which are then run on the Raspberry Pis.

Business Layer

Furthermore, while it is not possible to implement business goals held by certain stakeholders directly, the testbed can be utilised to run simulations, which check for the properties that the system should exhibit in order to fulfil the business goals. In this sense the testbed does not show how a business layer can be implemented but can validate that a system behaves as desired in simulations (with varying setups of components).

Mapping by Zone

Process

Process components are directly related to the generation and transmission of power. For the distribution domain, this is the hardware related to distributing the electricity inside the distribution network. To put it more concretely these are the power lines, transformers, and fuses. For the DER domain, the generation hardware includes primary energy sources, batteries, and the converter. As was outlined earlier in this section it is possible to include real deployments of transformers and converters in the testbed and locate a Raspberry Pi as a controller next to them. Alternatively, these components can be included by using a simulator on a cloud machine to compute power flows or by designating a Raspberry Pi as an emulator for a converter. In the present testbed emulated DER controller are included. These are part of the components realising the Grid Service Provision function and consequentially are located in the Process Zone and DER Domain of Figure 10

Field

Field equipment is the local control of the devices included in the process domain. In this case this can be for example the local control of a transformer, or the micro controllers located in a smart converter. In case that simulated components are included in the power grid the logic of these controller can be integrated as part of the simulation. Otherwise, the controller of the real deployment determine where these components are located. In our testbed the task is to control the ASs provision via DRES which is realised via (emulated) PLCs that fit into the Field Zone and DER domain.

Station

Station is the local aggregation level of the field level. In the context of the distribution domain this can be local Supervisory control and data acquisition (SCADA) or subsystem automation systems. For the DER domain, this can be a local control system that handles the local dispatching of multiple deployments co-located at the same site. It is possible to determine a Raspberry Pi as a higher level of aggregation and deploy the respective controller to it. In case said Raspberry Pi is located on the same site as the components of the field level this best mirrors the situation of multiple individual components controlled by a higher-level local controller. The deployment of our test bed decides to exclude such components which is the reason that the Station domain remains empty in Figure 10.

Operation

In the operation zone the power system control for the respective domain is located. As these processes for the distribution domain likely are not located on individual sites but at a central entity (e.g., the DSO or an aggregator) they should be located on the cloud machine. The operational control of the DER domain is distributed alongside the resources and therefore should be located on a Raspberry Pi rather than on a cloud machine. An example from our testbed are the backend and edge of OpenEMS. The OpenEMS backend is involved with the setpoint computation and communication functionalities while OpenEMS edge takes the responsibility of controlling the DRES locally. Its communication is aided by an MQTT agent to realize publish-subscribe style communication for monitoring. The overlay network is created using WireGuard and access to it requires to be part of the PKI which is built by the Vault Agent as the client software and the Vault PKI as the server. Taken together this enables a secure overlay network over which communication takes place. Within the SGAM plane the OpenEMS Backend, Vault PKI and MQTT broker are treated as centralised entities located in the Distribution domain, while the OpenEMS edge, MQTT agent and Vault agent components are required locally at the Raspberry Pi and therefore located in the DER domain of Figure 10.

Enterprise and Market

Components located in the Enterprise zone are responsible for the commercial and organisation processes while the Market zone contains trading processes. For completeness a cloud service is included in the mapping of Figure 10 that simulates request for AS calculated through mechanisms in these two zones. Despite this the operation of these zones is not of primary interest to this paper as no new approach to structuring the business case between the SO, third party and DRES owner is presented. Companies (e.g. Kiwi Power (2022)) are starting to push into these fields with integrating DRES better to aid synchronous generation. As a basis for the future presentation of business cases deployed to the testbed, we give an intuition of where to locate these components within the testbed and SGAM.

For the enterprise layer a main concern for the operation of the business processes is the accounting of services provided. For the distribution domain the accounting generally requires information from many other components in the system and are usually executed centralised at the corresponding organisation. Therefore, they can best be located on a machine inside the cloud of the testbed. Owners of multiple DRES may wish to deploy applications that provide aggregated billing and accounting for their DRES which are then located in this domain but in the DER zone instead. For owners of multiple DRES installations applications providing aggregated billing information about these DRES are located in this domain. These systems are likely centralised or even self-hosted by the owners of multiple DRES which indicates they should be located on a cloud machine.

The market zone deals with trading processes. Our stand-in for this is the aforementioned service simulating the request for AS. Usually, the market platforms to which offers are submitted will be realised in the cloud rather than on edge devices. Therefore, these would be deployed as cloud services. The resulting service is then located in the Distribution domain of the market zone. Whereas actors submitting their offers to the market may indeed be deployed on the Raspberry Pis.

Discussion

The realisation of a system which complies with the SGAM model has historically been quite challenging, with much of the functionality tied to the specific hardware components used. However, with modern design and development practices, coupled with new orchestration techniques it is possible to create a platform on to which novel testing regimes can be constructed.

The testbed design presented here supplies the needs of research in addition to closely aligning with the requirements of a large-scale distributed production system. Hardware in-the-loop based designs are increasingly necessary for accurate simulation of complex systems as the interactions between components reach higher complexities. While at the current stages our design is mainly focused on the lower levels of the SGAM model; the flexibility included allows the precise level at which the software stack can be used to be altered to match the requirements of any simulation, paving the way for future expansion.

Unfortunately, the application of the SGAM model to large-scale projects is a double-edged sword, whereby the tools it provides are particularly useful for formalising interactions between components, and where the responsibility for individual components starts and ends; it is also unfortunately a complex and (albeit necessarily) difficult tool to initially apply.

The different levels of abstraction when describing the SGAM model and the testbed structure are an example of this; the testbed is a concrete, technology focused implementation using specific software and tools, and is subject to any number of real-world constraints — environmental, human, and otherwise — whereas the SGAM model is intentionally very abstract and focuses on systems and interactions from a high-level. This difference in approaches can mean that consolidating the system entire into a single homogeneous design view is quite tricky, necessitating the inclusion of out-of-model descriptions for particular configurations, as the model itself does not capture these directly.

In particular, the mapping from specific SGAM zones and domains to specific software and hardware components has to be initially developed in parallel, including the implementation and the model, as the decisions made required knowledge of both and a certain amount of balancing and trade-off to reach a complete system structure.

Security of the Testbed

Security of the test bed is achieved by deploying containers and connecting them via the overlay network established using Wireguard VPN aided by the PKI established using Vault. For a device to be able to participate in this network in the first place, it must be assigned a pair of a public and a private key in the PKI system. On the one hand, this enables the encryption of messages which prevents man-in-the-middle or eavesdropping type attacks; on the other hand, a message signed using a specific key can be used to identify the sender. Therefore, if messages containing malicious data are detected, the participant sending them can be determined and appropriate action can be taken such as removing the offender from the overlay network.

The deployment of containers offers another advantage. When a container is detected to malfunction either due to malicious actions taken against, accidents or environmental influence it can be rebuilt from its image to restore its operation to the initial settings which can remove the cause for the malfunction in many cases. Furthermore, this ease of re-instantiating a container can help to deflect denial-of-service type attacks by migrating to another more powerful machine to withstand the attack. Further such attacks are made difficult as the attacker first, needs to gain access to a sufficiently large number of devices participating in the overlay network.

Issues with Geo-diverse Testbed deployments

The testbed structure described in this paper is intended to be flexible and have minimal impact on any existing structure at implementer sites, although does sometimes fall foul of particularly exotic host site network configurations. Notably, the technologies used here mean that only the coordinating host be visible externally, and all other participating nodes can connect to one another via this central point without the introduction of a dedicated piece of ingress controlling software.

This does bring up the obvious concerns related to bandwidth availability between equipment at the edges of the network, as they would necessitate communications with one or more intermediate hosts rather than using a direct link. However, as the vast majority of the network traffic to and from edge devices is likely to be directed between the edge and the cloud services, this deficit in a direct link should pose no problem under normal circumstances.

It is possible to peer directly between sites using the WireGuard VPN, should this become a particular pinch point for performance. Additional links could be created to offer shorter, faster, alternate routes, in addition to providing further resilience against failures. Indeed, in a large-scale production deployment, it would be expected that such links be created as a matter of course to reduce single-point failure probabilities.

Hardware Concerns

The Raspberry Pi platform is more than powerful enough to run all services as well as communication and management tasks local to the edges, and consumes only small amounts of energy to do so. Unfortunately, however, the choices for some hardware components have been made based on keeping the price of the unit down, rather than to ensure long-running stability; namely, the choices for storage and network hardware.

The Raspberry Pi uses an SD card for storage, and as flash storage has a limited lifespan — despite efforts to extend this Chang et al (2007) — it presents problems for systems working with large amounts of ephemeral data or exceptionally long runtimes. In the case of a short-term testbed deployment, this is not a concern, but if the hardware is to be suggested for longer-term usage in real-world deployments, then industrialised versions of the standard hardware, or better storage media (such as USB-attached solid-state disks) should be used, despite the additional costs and configuration changes.

Furthermore, the Ethernet network connection on the Raspberry Pi is only capable of 100 Mbit/s, which, while more than adequate for single site deployments,

could be saturated if a single Raspberry Pi were used as a gateway for an entire DRES deployment.

Conclusion

The Smart Grid is an important evolution of the electricity grid, changing from a top-heavy, strictly hierarchical, and downstream distribution of energy generation and consumption to a distributed bi-directional generation and consumption system through the growing use of Distributed Renewable Energy Sources (DRESs), complementing or even replacing the traditional centralised fossil-fuel powered energy sources.

While decentralising energy production, there is a need for distributed control to balance the grid, ensuring demand and supply are in equilibrium, to provide proper operation of the electrical grid. The mechanisms used to achieve this balance, such as frequency, voltage control and inertial response, are often referred to as Ancillary Service (AS). How AS can be implemented is well understood for traditional energy sources - such as synchronous generators - but the distributed and often intermittent nature of DRESs poses new challenges. The goal of the EASY-RES project is to develop new methods and approaches to ease the integration of DRES into the Smart Grid, making them behave more like traditional macro-scale sources.

An important part in enabling a distributed grid control design is having a well understood, distributed ICT infrastructure, and reasoning about such an ICT architecture is challenging in isolation. The Smart Grid Architecture Model (SGAM) is an architectural framework, covering all aspects of ICT, technical, and business functions within larger grid designs; it includes many parties involved in the Smart Grid (DSOs, TSOs, and other related industries). The common framework it provides helps in understanding, developing, and integrating the Smart Grid. Within this framework, an architecture to enable the provision of ASs using the capacities of DRESs was presented here.

A demonstration of the architecture was developed as a low-cost Smart Grid testbed which currently implements the lower-level parts of this architecture. The testbed is designed to be flexible by making extensive use of virtualisation. This method allows us to cleanly separate various systems and responsibilities which enables swapping out parts if required. The testbed also allows replacement of an emulated DRESs with real hardware, without any changes to the code. The deployment presented in this paper was found to provide sufficient performance with an average throughput of 27 MBit/sec between the Raspberry Pis and of 34.5 Mbit/sec between a Raspberry Pi and the cloud. The jitter in these cases was found to be below 3 milliseconds and below 1 millisecond respectively.

Finally, we explained how the testbed maps to the SGAM model to show how it helps in analysing various instances of inter-operability. In doing so the system design has been substantially verified.

Future Work

With the initial testbed deployment, only a small network was set up, comprised of only 4 sites. We are now working on adding more geographically distributed DRESs, both in virtualised and physical form to identify issues with the scalability of this approach, and further proving the generality of our approach.

Additionally, the testbed has currently focused on the 'edge' of a Smart Grid, integrating various DRESs and ensuring a secure bidirectional communication channel to management systems for monitoring and control. The next step is to integrate the optimisation work that has already been done in the project, and the accounting work which will extend the functionality across low-level DRES controllers and towards a greater complexity of cloud services. Ultimately, the goal is to provide an end-to-end testing environment of a full Smart Grid with geographically distributed DRESs involving hardware in-the-loop equipment and involving the complete set of stakeholders (DSOs, TSOs, DRES owners).

The future plan is for this end-to-end testing environments to be instantiated for different business cases related to the trading of AS either on a per AS basis or as aggregated provision of bundled AS. Such studies can yield insights into key performance indicators of the testbed such as adequacy of the possible monitoring resolution, stress on the data storage in the system in both the read and write direction and the delay times for scheduling the AS provision during normal and emergency operations. The architecture presented in this paper is intended to be a foundation for building such scenarios in the near future.

Appendix

1 SGAM

The smart grid architecture model (SGAM) was created by CEN, CENELEC, ETSI and is defined in *CEN-CENELEC-ETSI Smart Grid Co-Ordination Group Smart Grid Reference Architecture* Bruinenberg et al (2012). The main purpose of SGAM is to model interactions between different components of a system, where mainly the exchange of information is characterised. The framework models these interactions in three dimensions: Domains, Zones and Layers.

The five layers represent the interoperability from different viewpoints on the system, allowing for the consideration of the business goals and involved stakeholders, a functional view of the system, the information exchange and related standards, communication protocols and finally, the hardware and software components used to implement the system.

On each layer of the model a plane composed of domains and zones is presented. The five domains, generation, transmission, distribution, DER and customer premises, are derived from the different domains of the electrical energy production and conversion chain. The six zones are based on the automation pyramid; process, field, station and operation, along with two additional zones for enterprise and market concerns. A visual representation of this three dimensional model is shown in Figure 11.

Employing the SGAM framework offers several advantages, as can be seen by the variety of already preexisting approaches which utilise it as were described in Uslar et al (2019). The structure of model allows for a holistic view on the system, which is useful for finding interoperability problems Messinis et al (2016), and ensures consistency by tracing the components through the SGAM layers, highlighting the places where a given architecture design may be insufficient.

Funding

This work has received funding from the European Union's Horizon 2020 research and innovation programme via the "EASY-RES" project under grant agreement No 764090.

Abbreviations

AS: Ancillary Service; CAN: Controller Area Network; DER: Distributed Energy Resource; DRES: Distributed Renewable Energy Source; DSO: Distribution System Operator; DSP: Digital Signal Processor; EV: Electric Vehicle; HMI: Human-Machine Interface; HV: High Voltage; ICT: Information and Communications Technology; IoT: Internet of Things; LV: Low Voltage; MV: Medium Voltage; NIST: National Institute of Standards and Technology; PLC: Programmable Logic Controller; PV: PhotoVoltaic; RES: Renewable Energy Source; SBC: Single Board Computer; SCADA: Supervisory control and data acquisition; SG: Synchronous Generator; SGAM: Smart Grid Architecture Model; SO: System Operator; TSDB: Time Series Database; TSO: Transmission System Operator; VM: Virtual Machine; VPN: Virtual Private Network

Competing interests

The authors declare that they have no competing interests.

Consent for publication

The authors read and approved the published version of this article.

Authors' contributions

Armin Stocker (AS1), Ali Alshawish (AA), Martin Bor (MB), John Vidler (JV), Antonios Gouglidis (AG), Andrew Scott (AS2), Angelos Marnerides (AM), Hermann De Meer (HDM), and David Hutchison (DH) conceptualization, methodology: AS1, HDM, AG, AM; software, investigation: AS1, AA, JV, MB; writing-original draft preparation: AS1, JV, MB; writing-review and editing: AA, HDM, AG, DH; supervision: AG, AS2, AM, HDM; funding acquisition: HDM, AM, DH.

Author details

[1]Computer Networks and Computer Communications, University of Passau, Passau, Germany. [2]School of Computing and Communications, Lancaster University, Lancaster, United Kingdom. [3]School of Computing Science, University of Glasgow, Glasgow, United Kingdom.

References

Barragán-Villarejo M, García-López FdP, Marano-Marcolini A, Maza-Ortega JM (2020) Power system hardware in the loop (pshil): A holistic testing approach for smart grid technologies. Energies 13(15), , URL https://www.mdpi.com/1996-1073/13/15/3858

Bruinenberg J, Colton L, Darmois E, Dorn J, Doyle J, Elloumi O, Englert H, Forbes R, Heiles J, P H, et al (2012) Cen-cenelec-etsi smart grid co-ordination group smart grid reference architecture. URL https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf

Chang YH, Hsieh JW, Kuo TW (2007) Endurance enhancement of flash-memory storage systems: An efficient static wear leveling design. In: Proceedings of the 44th annual Design Automation Conference, pp 212–217

Donenfeld JA (2017) Wireguard: Next generation kernel network tunnel. In: NDSS

Estebsari A, Barbierato L, Bahmanyar A, Bottaccioli L, Macii E, Patti E (2019) A sgam-based test platform to develop a scheme for wide area measurement-free monitoring of smart grids under high pv penetration. Energies 12(8), , URL https://www.mdpi.com/1996-1073/12/8/1417

Gerend J, Coulter D, Lohr H, Ross E (2019) Microsoft: Containers vs. virtual machines. https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

Gouglidis A, Green B, Hutchison D, Alshawish A, de Meer H (2018) Surveillance and security: protecting electricity utilities and other critical infrastructures. Energy Inform 1(1):15, , URL https://doi.org/10.1186/s42162-018-0019-1

Hammad E, Farraj A, Kundur D (2019) On effective virtual inertia of storage-based distributed control for transient stability. IEEE Transactions on Smart Grid 10(1):327–336,

Hooshyar H, Vanfretti L (2017) A sgam-based architecture for synchrophasor applications facilitating tso/dso interactions. In: IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT), Arlington, VA, USA, pp 1–5

Jindal A, Marnerides AK, Gouglidis A, Mauthe A, Hutchison D (2019) Communication standards for distributed renewable energy sources integration in future electricity distribution networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, pp 8390–8393,

Jindal A, Marnerides AK, Scott A, Hutchison D (2019) Identifying security challenges in renewable energy systems: A wind turbine case study. In: Proceedings of the Tenth ACM International Conference on Future Energy Systems, Association for Computing Machinery, New York, NY, USA, e-Energy '19, p 370–372, , URL https://doi.org/10.1145/3307772.3330154

Jindal A, Kronawitter J, Kühn R, Bor M, de Meer H, Gouglidis A, Hutchison D, Marnerides AK, Scott A, Mauthe A (2020) A flexible ict architecture to support ancillary services in future electricity distribution networks: an accounting use case for dsos. Energy Inform 3(1):6, , URL https://doi.org/10.1186/s42162-020-00111-x

Kim H, Kim YJ, Yang K, Thottan M (2011) Cloud-based demand response for smart grid: Architecture and distributed algorithms. In: 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp 398–403,

Kiwi Power (2022) Kiwi powered website. URL https://www.kiwipowered.com/about

Kryonidis GC, Demoulias CS, Papagiannis GK (2019) A new voltage control scheme for active medium-voltage (mv) networks. Electr Power Syst Res 169:53 – 64, , URL http://www.sciencedirect.com/science/article/pii/S0378779618304115

Langset T (2021) Implementation of data hubs in the Nordic countries. Tech. rep., Nordic Energy Regulators, URL http://www.nordicenergyregulators.org/wp-content/uploads/2021/12/6.1-NordREG-Status-report-on-data-hubs-2021.pdf

Leal A, Botero F (2021) An architecture for power substations communication networks based on sdn and virtualization paradigms. Revista Facultad de Ingeniería, Universidad de Antioquia (100):48–66

Leal A, Botero JF (2019) Defining a reliable network topology in software-defined power substations. IEEE Access 7:14,323–14,339,

Lehnhoff S, Rohjans S, Uslar M, Mahnke W (2012) Opc unified architecture: A service-oriented architecture for smart grids. In: First International Workshop on Software Engineering Challenges for the Smart Grid (SE-SmartGrids), Zurich, Switzerland, pp 1–7,

Luciano M, Helfried B, Emilio R, Chris C, Thomas I S, Graeme M B (2017) Grid of the future and the need for a decentralised control architecture: the web-of-cells concept. CIRED - Open Access Proceedings Journal 2017:1162–1166(4)

Merino J, Rodriguez-Seco JE, Garcia-Villalba I, Temiz A, Caerts C, Schwalbe R, Strasser T (2017) Electra irp voltage control strategy for enhancing power system stability in future grid architectures. CIRED - Open Access Proceedings Journal 2017:1068–1072(4),

Messinis G, Dimeas A, Hatziargyriou N, Kokos I, Lamprinos I (2016) Ict tools for enabling smart grid players' flexibility through vpp and dr services. In: 13th International Conference on the European Energy Market (EEM), Porto, Portugal, pp 1–5

Potenciano Menci S, Le Baut J, Matanza Domingo J, López López G, Cossent Arín R, Pio Silva M (2020) A novel methodology for the scalability analysis of ict systems for smart grids based on sgam: the integrid project approach. Energies 13(15):3818

Radi M, Taylor G, Uslar M, Köhlke J, Suljanovic N (2019) Bidirectional power and data flow via enhanced portal based tso-dso coordination. In: 54th International Universities Power Engineering Conference (UPEC), Bucharest, Romania, pp 1–5

Sirviö KH, Laaksonen H, Kauhaniemi K, Hatziargyriou N (2021) Evolution of the electricity distribution networks—active management architecture schemes and microgrid control functionalities. Applied Sciences 11(6):2793

Thornton M, Smidt H, Schwarzer V, Motalleb M, Ghorbani R (2017) Internet-of-things hardware-in-the-loop simulation testbed for demand response ancillary services. Materials for Energy, Efficiency and Sustainability, TechConnect Briefs pp 66–69

Tian P, Xiao X, Wang K, Ding R (2016) A hierarchical energy management system based on hierarchical optimization for microgrid community economic operation. IEEE Trans Smart Grid 7(5):2230–2241,

Uslar M, Rohjans S, Neureiter C, Pröstl Andrén F, Velasquez J, Steinbrink C, Efthymiou V, Migliavacca G, Horsmanheimo S, Brunner H, Strasser TI (2019) Applying the smart grid architecture model for designing and validating system-of-systems in the power and energy domain: A european perspective. Energies 12(2), , URL https://www.mdpi.com/1996-1073/12/2/258

Yuen C, Oudalov A, Timbus A (2011) The provision of frequency control reserves from multiple microgrids. IEEE Trans Ind Electron 58(1):173–183,

Figures

Figure 1: Schematic representation of the future smart grid.

Figure 2: Function layer of the proposed architecture.

Figure 3: Component layer of the proposed architecture.

Figure 4: Testbed system architecture. Solid boxes indicate physical assets, dashed boxes indicate virtualised assets.

Figure 5: Testbed software architecture. Red boxes are services running directly on the operating system, blue boxes are services which run in containers. White dashed boxes are emulated functions, which in this case also run in a container.

Figure 6: iperf3 test results for jitter on the testbed for a 30 second window connecting from an edge device to the cloud services host

Figure 7: iperf3 test results for jitter on the testbed for a 30 second window connecting from an edge device to another edge device

Figure 8: iperf3 test results for throughput on the testbed for a 30 second window connecting from an edge device to the cloud services host.

Figure 9: iperf3 test results for throughput on the testbed for a 30 second window connecting between two edge devices.

Figure 10: Mapping of the test bed components into the SGAM plane.

Figure 11: Graphical representation of the SGAM framework as shown in Bruinenberg et al (2012).