Large Scale Rollout of Smart Grid Services

Florian Kintzler*, Tobias Gawron-Deutsch*, Stephan Cejka*, Judith Schulte†, Mathias Uslar†,
Eric MSP Veith†, Ewa Piatkowska‡, Paul Smith‡, Friederich Kupzog‡,
Henrik Sandberg§, Michelle S. Chong§, David Umsonst§, and Marco Mittelsdorf¶
*Corporate Technology, Siemens AG Austria, Vienna, Austria
Email: {florian.kintzler, tobias.gawron-deutsch, stephan.cejka}@siemens.com
†R&D Division Energy, OFFIS e.V. - Institut für Informatik, Oldenburg, Germany,
Email: {judith.schulte, mathias.uslar, eric.veith}@offis.de
‡Center for Digital Safety & Security, AIT Austrian Institute of Technology, Vienna, Austria,
Email: {ewa.piatkowska, paul.smith, friederich.kupzog}@ait.ac.at
§Department of Automatic Control, School of Electrical Engineering and Computer Science,
KTH Royal Institute of Technology, Stockholm, Sweden
Email: {hsan, mchong, umsonst}@kth.se
¶Fraunhofer Institute for Solar Energy Systems, Freiburg, Germany,
Email: marco.mittelsdorf@ise.fraunhofer.de

***Abstract*—In the domain of energy automation, where a massive number of software-based IoT services interact with a complex dynamic system, processes for software installation and software update become more important and more complex. These processes have to ensure that the dependencies on all layers are fulfilled, including dependencies arising due to the energy system controlled by IT components being a hidden communication channel between these components. In addition, the processes have to be resilient against faults in and attacks to both the energy grid and the communication network.**

The ERA-Net funded project LarGo! aims at developing and testing processes for the large scale rollout of software applications in the power grid domain as well as the user domain. This article describes a work in progress and the project's roadmap to solve the technical issues. It investigates the problems that arise from the interlocking of the two networks – the power grid and the communication network. Based on this analysis a first set of requirements for a rollout process in such a Smart Grid is derived and the chosen approach to verify the resilience of the developed processes under research is described.

***Index Terms*—Application Management, Rollout, Smart Grid, Processes.**

I. Introduction

A. Problem Statement

The paradigm in which classical Remote Terminal Units (RTUs) are connected to a Supervisory Control And Data Acquisition (SCADA) system using an isolated communication infrastructure cannot be applied to Smart Grid operation. In contrast to RTU-based distribution grid control, modern Information and Communications Technology (ICT) based control in Smart Grids requires sophisticated device- and application-management as well as security maintenance processes and technology. Distributed monitoring and control schemes are required in both the distribution (e.g., in smart secondary substations) and customer (e.g., using energy management systems) domains, which are connected using a public or shared ICT infrastructure.

In previous projects (e.g., for Smart City Demo Aspern [1], enera [2], and ZEROPlus [3]), Smart Grid applications were developed to support economic and ecological energy supply using this form of infrastructure. These applications were manually deployed to the field devices. In real world environments, hundreds of secondary substations and (hundreds of) thousands of devices in the customer domain are to be managed. These nodes are distributed and connected via an ICT infrastructure that varies greatly in quality. In these environments the manual approach is not feasible anymore.

In the smart phone sector the installation or the update of a software component (app, driver, operating system, etc.) does only affect the device itself. Industrial IoT apps can interact with an external system (e.g., the power grid) that becomes a hidden communication channel. Thus for these applications it is important to ensure that the running apps (or an area-related subset of these apps) work together correctly. The deployment process must therefore be resilient to faults and attacks in both the ICT system and the power grid system.

The ERA-NET [4] funded project LarGo! investigate the mass rollout of Smart Grid applications for energy and grid management. It tackles the challenge of stable and resilient system operation in a setting where communication systems are used for both Smart Grid runtime operation (such as monitoring), controls and ICT maintenance (such as application deployment and patching), as well as remote configuration.

B. Scientific Approach

We pose the hypothesis that ICT maintenance cannot be conducted independently of the runtime operation of a Smart Grid. For example, on a utility scale, the time required for deployment and ICT maintenance processes overlaps significantly with operational periods (i.e., these two aspects cannot be readily separated). Furthermore, the exchange of operational data will use the same communication channels as used for ICT maintenance. As an example of the problems this causes, we have observed that the testing of novel features for smart meters is difficult because remotely updating firmware (required to deploy new features) and the standard process of meter reading adversely interfere with each other on the communication channel. Furthermore, firewall updates in distribution substations can cause them to be offline for substantial periods.

To test this hypothesis and evaluate solutions, LarGo! will implement a utility-scale and highly accurate emulation of the required systems for ICT maintenance. To support this, monitoring and grid control approaches from national demonstrators (Smart City Demo Aspern in Austria [1] and enera [2], ZEROPlus in Germany [3]) are scaled-up and operated in this realistic environment, allowing different design options to be analysed. We will use co-simulation to examine the interaction between ICT and physical power systems, which could result in the degradation of operational quality indices, such as power quality or losses.

Existing tools and methods for Smart Grid co-simulation (the Mosaik framework [5], OpenMUC [6], [7], modular Smart Grid applications [8], [9], application lifecycle management [9]–[11], a time-series store [11], [12], testing methods [13], and requirements elicitation methods [14]) that have been developed by the project partners will be used and extended. Additionally, co-simulation results can be verified with Controller Hardware in the Loop (CHIL) and Power Hardware in the Loop (PHIL) experiments. Finally, to gain real-world insights, selected Smart Grid applications will be rolled out in the Smart City Aspern and ZEROPlus testbeds.

The Smart Grid applications that will be considered in the LarGo! project will be described through a set of use cases. Here, a use case is a specification of a set of actions performed by a system, which yields an observable result that is of value for one or more actors. The overall integration of use cases will be illustrated with the help of the Smart Grid Architecture Model (SGAM) Framework [15] and the IEC 62559 use case template [14]. Thus, requirements focus not only on functionality, but also gaps in the system design can be analysed.

C. Outline

In the following sections we describe the current state of research in the LarGo! project. We first describe the requirements elicitation in the light of a secure and resilient massive service rollout. We describe basic domain requirements and how LarGo! is going to approach the resulting challenges (Section II). Thereafter (Section III) we discuss several benign and malicious ways that the rollout of Smart Grid services could be challenged, resulting in failure. In Section IV we discuss the security and resilience aspects of the massive deployment of services in a Smart Grid. The environment in which LarGo! is going to test the developed algorithms and mechanisms is then described in Section V. In Section VI we describe the next steps the LarGo! project is going to take to achieve the described goals. The paper is concluded with a summary in Section VII.

II. Rollout Process - Requirements

A. Requirements Elicitation

One important aspect in the system design-phase is the process of requirements elicitation. Stakeholders have to provide both the functional as well as the non-functional requirements for the engineer to implement a meaningful system and behavior. Given the usual design process, it is meaningful to spend effort at the early stages on high-quality requirements documentation as it will save expenditures in the later stages of the projects where the costs will even be higher to correct faulty design decisions. One particular way to deal with a meaningful process of requirements elicitation is the use of the IEC 62559 use case case template [14]. Within the M/490 mandate by the European Commission to CEN/CENELEC and ETSI, the template was further elaborated on and can be seen as the state-of-the-art method of documenting domain-specific knowledge from stakeholders for Smart Grid applications [16].

Still, the original scope of IEC 62559 template was on the envisioned behavior of both actors and systems in scope. Within the scope of defining resilient systems and their requirements, we need to shift from documenting observed intended behavior of a system to possible non-intended behavior. This is often called a mis-use case, the term being derived from and meaning the inverse of use case. It describes the process of executing a malicious act against a system, while use case can be used to describe any action taken by the system. Within LarGo!, we will use an extended version of the IEC 62559 template to cover mis-use cases in order to elicit information for resilience criteria analysis.

B. The Domain

A process to rollout applications to a massive number of devices must take dependencies on various levels into account. As depicted in Figure 1 these dependencies range from device level software version dependencies, and system level dependencies like protocol version numbers, to domain specific dependencies.

In general the dependencies of an application and thus its impact on the power grid can either be system wide, or limited to a sub-scope of the system. For example, an update of an application that optimizes the energy consumption within a household by using a battery for specific consumers (e.g.,

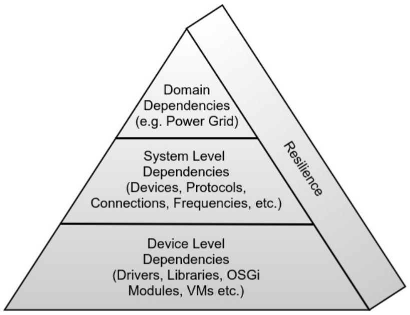
Domain Dependencies (e.g. Power Grid)
System Level Dependencies (Devices, Protocols, Connections, Frequencies, etc.)
Device Level Dependencies (Drivers, Libraries, OSGi Modules, VMs etc.)
Resilience

Fig. 1. Requirements Pyramid

garden lightning) but without influence on the power grid, can be rolled out to specific devices without taking other grid components into account. In case the application has an effect on the consumption from the power grid, the update of this application in multiple households connected to the same branch in the low voltage grid might have an effect on the voltage stability of this branch, but is not likely to have an effect beyond the border between low voltage and medium voltage sections of the grid. The application could thus be rolled out in several steps, each step covering a separate low voltage grid section to avoid a blackout of all branches at the same time and to be able to stop and rollback the rollout process in case of failures.

Before an application can be started, the dependencies on all levels (cf. Figure 1) must be satisfied. To ensure that all technical dependencies are fulfilled on the System Level and the Device Level the most common approach is to use identifiers and version numbers. There are domain level dependencies, which are directly coupled to one of the lower levels. For example, the stability of the power could depend on the usage of specific versions of two otherwise non-coupled applications, which can be ensured using the common techniques already applied on the System and Device Level. However since these dependencies are non-technical on the Component Layer, Communication Layer, and Information Layer, methods are needed to manage and make these dependencies explicit on the Function Layer or Business Layer (for the different layers see SGAM [15]).

To ensure that a rolled out application works correctly, its output can be validated against a model of the environment. This validation can not only be done in fully operational mode, but in addition in a standby phase in which the application receives input, but is not allowed to provide its output to other components and thus alter the state of the environment (cf. [17]). Depending on the scope in which the application interacts with the environment, the validation can also be scope specific (e.g., because of performance restrictions) or cover a larger scope (e.g., to detect impacts on a larger scope via the hidden communication channel power grid):

- In a Building Energy Management System (BEMS), the output of the applications could be checked against a model of the building.
- In a secondary substation the output of apps installed on devices in the substation or even of applications in a BEMS that consume power via the substation, can be validated against a model that includes the substation itself, multiple buildings and the power lines between these components [18].
- In Central Control (CC) a high level model could be used to check the output of applications on devices in the substations. Since CC is the most likely place for extensive computing power, the model could also be much more detailed than the models used for verification in the substations and the BEMSs (e.g., the open Common Information Model – CIM standard [19], [20]).

In addition to ensuring that the described dependencies are satisfied on all layers during the complete process, the rollout of applications to a massive number of devices which interact with a complex environment must fulfill at least the following basic cross-use-case requirements:

- Minimize interference with operational processes.
- Provide automatic rollback in case of a failure.
- Require as little human interaction as possible.
- Resilience against faults of and attacks against the ICT network and the power grid (see Sections III and IV).

Analogue to transactions in the database domain, the rollout process needs mean to automatically rollback the state of all involved distributed devices to a previous state. However a complete rollback to the previous state can only be ensured in the ICT domain. When the applications already altered the controlled complex system (i.e., the state of the power grid), the recovery to a stable, full functional state of the overall subsystems might not be possible. In this case human interaction is unavoidable.

Figure 2 shows an example for an application lifecycle based on the OSGi app lifecycle [21] that fulfills the basic requirements needed to support the domain specific use-cases. It focuses only on one device or runtime environment, which does not necessarily require a permanent connection to the operator backend. Once an application's installation is initiated (i.e., the respective command is either executed by the operator by use of a backend dashboard or by a scheduled event), the respective app is downloaded to the intelligent Substation Node (iSSN) and the app's lifecycle is started. All lifecycle tasks (i.e., install, start, stop, uninstall) include an immediate state step in which the respective tasks are executed. The lifecycle is enriched by the two sub-states *In Verification* and *Active* to implement the described app verification mechanism. This need not be a direct sub-state in the app lifecycle but can be indirectly implemented by using configuration interfaces to the application.

In real life scenarios, the devices to be managed by the software rollout system will be inhomogeneous. Thus the system will have to include means to handle different app lifecycles or wrap them, so that they fulfill the described basic requirements.

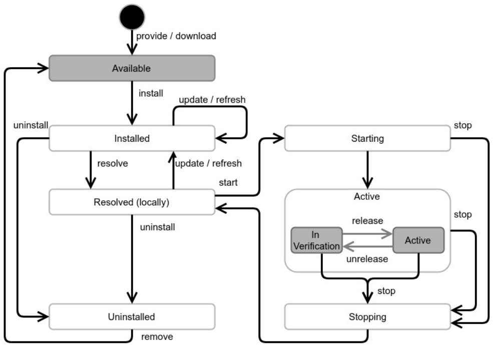

Fig. 2. Basic application lifecycle to support domain-specific use-cases.

III. Challenges to Service Rollouts

There are several ways that the rollout of Smart Grid services could be challenged, resulting in failure. These challenges can either be *benign* or *malicious*. Regarding benign challenges, we foresee four main types: *(i)* issues related to Quality of Service (QoS), for example, in the communication network; *(ii)* faulty hardware, software, or (change management) processes; *(iii)* incomplete state awareness; and *(iv)* human error.

The communication networks that support Smart Grid services can use a range of technologies with different QoS characteristics; for example, related to bandwidth, delay, error rates, and losses. In many cases, a range of technologies will be used along an end-to-end path in the Smart Grid, including optical fiber, Wireless LAN (WLAN), Power-Line Communication (PLC), GSM, Zigbee, etc. Insufficient QoS, for instance caused by transient network demand or losses caused by environmental factors, can result in service rollouts not completing or taking unacceptably extended periods to complete.

The potential situations that faults could be triggered, resulting in a rollout failure, are manifold: in the Smart Grid, there is a combination of legacy systems, which are relatively fragile to change, as well as novel devices and software. The deployment of novel services, which may contain faults (software bugs) could result in service failures. This may also occur when novel services negatively interact with existing and legacy systems. Furthermore, faults in the software and processes that are used to support service rollouts could result in a novel service not being successfully deployed.

Closely related to these challenges are failures that are caused by incomplete or incorrect awareness of the system state, and how it is typically operated and functions. A good example of this form of challenge are differences in hardware configurations, such as available volatile memory and storage, that are used for testing new services – prior to deployment – and those that are deployed in the field. Similarly, there may be differences between organizational processes that describe how systems are used, which are used as a basis for supporting rollout decisions, and the way the system is used; these inconsistencies could lead to failures of the rollout process or the novel service being deployed.

Perhaps one of the most prevalent and difficult to manage challenges to software rollouts are human errors. These can be caused by insufficient knowledge and training in the service rollout processes and systems, for example. A key distinction that can be made regarding human error, is whether the person involved is unwittingly (or not) being manipulated as part of an attack that is using a social engineering component. In this case, behavioral analysis techniques can be used to detect abnormal patterns that could indicate that a misuse of systems is taking place.

There are several malicious approaches that can be used to cause failures in service rollouts that need to be addressed. In general, from an operational perspective, an attacker will aim to compromise the integrity and availability of various aspects associated with rollouts. For example, an attacker can aim to compromise the integrity of software modules, which are to be deployed, that reside in repositories or while they are being communicated to target devices. As a further example, a Denial of Service (DoS) attack could be targeted at the systems and networks that are involved in the service rollout process.

The target of a cyber-attack could be the systems being used for service rollouts, such as software repositories, management systems, or communication networks. Furthermore, to inhibit the successful rollout of a service, an attacker may choose to compromise the environment the services are being deployed into. For example, this could take the form of an attacker manipulating the measured state of the system, resulting in failures that are like those mentioned earlier regarding inconsistent state awareness. Similarly, an attacker could seek to drive the system (either its cyber or physical systems) into a non-nominal state, which could result in a service rollout being curtailed. Previous works have proven that system redundancy is key in mitigating such forms of attacks [22]–[24]. However, in large-scale and distributed systems such as the Smart Grid, the challenge is to keep infrastructure redundancy minimal while ensuring resilience against attacks, which we aim to do through the development of clever algorithms (e.g., [25], [26]).

In short, the potential threat actors (e.g., nation state actors, industrial competitors, . . .), attacker vectors (e.g., social engineering, supply chain attacks, physical and blended attacks, . . .), and types of attack (e.g., DoS attacks, Man-in-The-Middle attack, . . .) are manifold. In this context, it is critical that operators regularly perform an assessment of cyber security risks and deploy appropriate measures to mitigate identified risks.

IV. Secure and Resilient Rollout

A. Secure Architecture

The overall architecture in LarGo! shall be prone to certain kind of attacks which can already be envisioned. Different repositories documenting attacks already exist; for example, the Open Web Application Security Project (OWASP) repository dealing with vulnerabilities of web-based systems, the Repository of Industrial Security Incidents (RISI) database or the Common Attack Pattern Enumeration and Classification (CAPEC) dictionary and classification taxonomy database [27].

Those vulnerabilities provide good means as a starting point to take into account certain decisions at design time of the LarGo! system and its rollout. In addition, requirements from stakeholders have to be taken into account as documented in Section II.A. While positive system functionality is covered by use case management, the aforementioned misuse cases and a new template will provide a way to document the unintended behavior and needed mitigations to deal with threats. Defined scenarios will provide threats and typical attacks from the aforementioned databases, and annotate mitigations in both design process and operations to the corresponding scenarios;

thus, addressing known and generic vulnerabilities at design-time of both the system and its operational procedures. This will make for a good traceability of requirements for a secure and safe system operation and provide trust in the system developed [28].

B. Resilient Control

The aim of resilient control in the face of faults and attacks on the power grid connected via an ICT infrastructure is to ensure its smooth operation through smart algorithms [29]. Mathematical control and systems engineering presents tools that are amenable for the design of control algorithms that are distributed, automated, and resilient. Typically, the algorithms employ physical models and measurement redundancy for detecting and isolating misbehaving devices (e.g., [24]). In response to a detected misbehavior, the controller could automatically reconfigure, or ask for human intervention.

LarGo! focuses on the safe and secure rollout of new applications in the Smart Grid, and resilient controllers can assist in multiple ways. Two key points prone to benign and malicious faults and attacks, especially during rollout, are *(i)* the metering and actuation devices; and *(ii)* the Supervisory Control and Data Acquisition system (SCADA). During the rollout, resilient control methodologies can be used to monitor the system state and alert the operator of any mis-use.

Further, resilient control strategies provide a systematic methodology for ensuring the patched software performs as desired when deployed in the Smart Grid through modular design and checkable criteria that can be performed offline. Additionally, software deployment is also rolled out in a careful manner to ensure maximal protection against system failure and attacks. For this, we need to investigate the design and tuning of model-based anomaly detectors (e.g., [30], [31]).

V. Co-Simulation

In the past, approaches to simulating a part of the Smart Grid a researcher was interested in has always used one of the most prudent approaches: simplifying the model. In short, while a researcher models the part of the simulation he is familiar with in intricate detail, all other parts get either abstracted away or are assumed to “just work”. In an integrated Smart Grid environment, this can be a dangerous assumption.

As described above, the ICT infrastructure, the power grid, and the applications that will in future be a substantial part of the overall power system are interlocked. How will an unsuccessful application deployment impact the grid operations? Can a failure in the ICT infrastructure endanger the provisioning of real or reactive power? What happens on a brownout with subsequent loss of the computer networks? Domains that have once been largely seen as a commodity (such as the communication infrastructure that powers the Internet) have, with the notion of the Smart Grid and the beginning pervasion of applications on top of the tradition open- and closed loop control systems, become critical to the operation of the power grid.

However, there is no one software to simulate them all. Every domain has its experts with their detailed knowledge; creating a software package that would include ICT simulations, power grid simulations, power flow studies, and virtual environments for real-world software applications, such as the LarGo! project needs, would be a herculean effort at best and would most probably fail. Therefore, our attention shifted to the coupling of domain-specific simulators. Coupling, enabling data exchange and synchronizing these simulators in a divide-et-impera approach is the better solution.

A co-simulation framework such as Mosaik [5] allows to couple software of different vendors, executed on different machines with different operating systems. It allows to let these simulation softwares exchange data, such as supplying a voltage reading from a power flow study, routing it as a network packet through the ICT simulation and presenting it to an application tied to a third simulator. LarGo! will make heavy use of this ability to the extend to not only couple simulators, but also the project partners through VPN tunnels, allowing each to contribute their expertise to an integrated, albeit distributed, simulation environment with real Controller and Power Hardware in the loop (CHIL/PHIL, see Figure 3). The following paragraphs describe our assumptions for the creation of the co-simulation testbed.

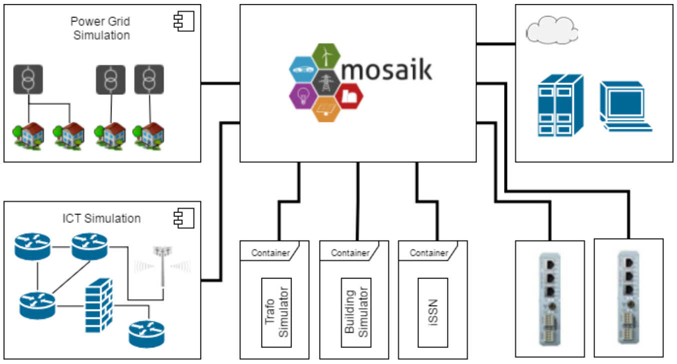

Fig. 3. Architecture of Simulation Environment

To this end, we have created an ICT network using a private IP network range. This range is subdivided into three distinct, but interconnected, network areas to model different network archetypes. Here, we assume that mainly services based on the Transmission Control Protocol (TCP) will see deployment. Considering the standard literature [32], [33], as well as analyses (e.g., [34]–[36]), we can safely assume that an initial ICT modeling concept that concentrates on the delay as perceived by the application layer is largely sufficient for the task at hand.

The *Dedicated Network Area* serves to model the best-case condition: links dedicated to the ICT requirements of the power grid operator. We assume that neither broken links nor any packet loss due to congestion happens, as the dedicated network features reliable links and sufficiently-sized hardware. Therefore, delay can be proportional to:

$$d \sim 10 + 50 \cdot \mathrm{f}(x, 1) \ [\mathrm{ms}] \ , \tag{1}$$

where $\mathrm{f}(x, 1)$ denotes the Probability Density Function (PDF) of an exponential distribution with $\lambda = 1$.

The second defined area is the *Shared Links Area*. It covers most current ICT for distribution system operators that chose to use encrypted communication over public telecommunications infrastructure. Here, we will see occasionally congestion, but no line drops. The delay follows a normal distribution:

$$d \sim \mathcal{N}(50; 250) \ [\mathrm{ms}] \ . \tag{2}$$

The extreme end is denoted by the *High-impairment Network Area*. It assumes many low data-rate wireless links with volatile link quality, letting packet transmission stall or even abort at uncontrollable intervals. This network area will be the litmus test for a resilient rollout process in its modeling of network transmission delay:

$$d \sim \mathcal{U}[0; \infty] \ [\mathrm{ms}] \ . \tag{3}$$

In addition to the simulated ICT network, the applications that see deployment in LarGo! – such as the ones powered by the openMUC framework [6] – are connected to the ICT simulation using a virtual network interface, a TAP device [37]. The respective software is deployed in Docker containers, having its usual software environment, while the virtual network device sets the default route upon initialization and routes all traffic into the ICT simulation, where it is routed like any real traffic and treated according to the above network area specifications.

The power grid simulation (see Figure 3) is designed using the co-simulation middleware LabLink [38] and is not limited to software only components. It interacts with the real controller and power hardware in addition to power gird modeled with the leading power system analysis tool DIgSILENT PowerFactory [39].

VI. Future Work

The LarGo! project is currently finalizing the specification of detailed use-cases for service rollouts that are to be implemented in the described simulation environment. From these use-cases requirements for the secure and resilient rollout process are derived using the described requirements elicitation process.

Existing technologies, for example including resin.io [40], Eclipse hawkBit [41] and others, will be evaluated against the derived domain specific requirements and analysed with respect to the described security and resilience requirements. Based on the derived requirements, and on the analysis of state of the art solutions for software rollout, a process for massive service rollout in Smart Grids will be defined and tested in the described simulation environment. We expect a detailed evaluation of the large co-simulation environment. In future publications, we will show how the co-simulation approach influences the assessment of the software roll-out process and describe the runtime characteristics.

The LarGo! project will end with a field test in the test-beds of Smart City Aspern [1] and ZEROPlus [3].

VII. Conclusion

Ongoing digitalization in the power system domain results in a changing environment for ICT in electric distribution grids. LarGo! aims at creating three central innovative contributions:

(1) LarGo! develops an open and standardized deployment process that can be applied in the grid and customer domain. This output has a strong impact on the efficiency of Smart Grid rollouts, the creation of marketplaces for Smart Grid applications and the adoption potential of new Smart Grid solutions.

(2) LarGo! seeks for a resilient system and controls design that tolerates the temporarily unavailability of ICT components as a result of patching processes, technical failures or even malicious actions. This contribution is unique because it is made with a holistic view on operational resilience of the power grid infrastructure itself, the necessary communication infrastructure, the design of applications and the necessary ICT maintenance and deployment processes. This will also result in requirements of future application design for mass rollout.

(3) LarGo! will realize a unique up-scaling and validation environment based on a large-scale and highly detailed simulation approach with several (> 10) primary and > 500 secondary substations. Applications get deployed, maintained and operated in an emulated cyber-physical environment with primary and secondary substations of a distribution segment, together with many active management systems on customer sites. This environment, capable of enabling coupling of real-world controller and power hardware for detailed analyses, will have a tremendous potential for re-use after the project.

Acknowledgments

The presented work is conducted in the framework of the joint programming initiative ERA-Net Smart Grids Plus [4], with support from the European Unions Horizon 2020 research and innovation programme. On national level the work was funded and supported by the Austrian Climate and Energy Fund (KLIEN, ref. 857570), by German BMWi (FKZ 0350012A), and by the Swedish Energy Agency (Project number 42794-1).

References

[1] Aspern Smart City Research (ASCR). [Online]. Available: http//www.ascr.at/

[2] enera. [Online]. Available: http://www.energie-vernetzen.de

[3] ZeroPLUS. [Online]. Available: https://www.ise.fraunhofer.de/de/forschungsprojekte/fellbach-zeroplus.html

[4] ERA-Net Smart Grids Plus. [Online]. Available: http://www.eranet-smartgridsplus.eu/

[5] Mosaik. [Online]. Available: https://mosaik.offis.de/

[6] OpenMUC. [Online]. Available: https://www.openmuc.org/

[7] S. Feuerhahn, M. Zillgith, R. Becker, and C. Wittwer, "Implementation of an open smart metering reference platform - OpenMUC," *ETG-Kongress*, 2009.

[8] A. Einfalt, S. Cejka, K. Diwold, A. Frischenschlager, M. Faschang, M. Stefan, and F. Kupzog, "Interaction of smart grid applications supporting plug & automate for intelligent secondary substations," *CIRED - Open Access Proceedings Journal*, vol. 2017, no. 1, pp. 1257–1260, Oct 2017.

[9] T. Gawron-Deutsch, K. Diwold, S. Cejka, M. Matschnig, and A. Einfalt, "Industrial IoT for Smart Grid in-field Analytics," *e & i Elektrotechnik und Informationstechnik*, vol. 135, no. 3, 2018, to appear.

[10] S. Cejka, A. Hanzlik, and A. Plank, "A framework for communication and provisioning in an intelligent secondary substation," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sept 2016.

[11] M. Faschang, S. Cejka, M. Stefan, A. Frischenschlager, A. Einfalt, K. Diwold, F. Pröstl Andrén, T. Strasser, and F. Kupzog, "Provisioning, deployment, and operation of smart grid applications on substation level," *Computer Science - Research and Development*, vol. 32, no. 1, pp. 117–130, Mar 2017.

[12] S. Cejka, R. Mosshammer, and A. Einfalt, "Java embedded storage for time series and meta data in Smart Grids," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2015, pp. 434–439.

[13] F. Schloegl, M. Buescher, K. Diwold, S. Lehnhoff, L. Fischer, F. Zeilinger, and T. Gawron-Deutsch, "Performance testing smart grid applications using a distributed co-simulation approach," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 6305–6310.

[14] M. Gottschalk, C. Delfs, and M. Uslar, *The Use Case and Smart Grid Architecture Model Approach: The IEC 62559-2 Use Case Template and the SGAM applied in various domains*, ser. SpringerBriefs in Energy. Springer, Jan 2017.

[15] Smart Grid Coordination Group, "Smart grid reference architecture," CEN-CENELEC-ETSI, Tech. Rep., 2012. [Online]. Available: ftp://ftp.cencenelec.eu/EN/EuropeanStandardization/HotTopics/SmartGrids/Reference_Architecture_final.pdf

[16] M. Uslar, "Energy Informatics: Definition, State-of-the-Art and New Horizons," vol. 6, 2015, pp. 15–26.

[17] S. Cejka, A. Frischenschlager, M. Faschang, M. Stefan, and K. Diwold, "Operation of modular smart grid applications interacting through a distributed middleware," *Open Journal of Big Data (OJBD)*, vol. 4, no. 1, pp. 14–29, 2018.

[18] M. Stefan, M. Faschang, S. Cejka, K. Diwold, A. Einfalt, and A. Frischenschlager, "Distribution grid topology validation and identification by graph-based load profile analysis," in *IEEE International Conference on Industrial Technology (ICIT)*, Feb 2018.

[19] "IEC 61970: Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base, Third Edition," Tech. Rep., 2009.

[20] M. Uslar, M. Specht, S. Rohjans, J. Trefke, and J. Gonzalez, *The Common Information Model CIM: IEC 61968/61970 and 62325 - A practical introduction to the CIM*. Springer, 2012.

[21] OSGi. [Online]. Available: https://www.osgi.org/developer/specifications/

[22] Y. Shoukry, M. Chong, M. Wakaiki, P. Nuzzo, A. Sangiovanni-Vincentelli, S. A. Seshia, J. P. Hespanha, and P. Tabuada, "SMT-Based Observer Design for Cyber-Physical Systems Under Sensor Attacks," *ACM Transactions on Cyber-Physical Systems*, vol. 2, no. 1, pp. 5:1–5:27, Jan. 2018.

[23] Z. Tang, M. Kuijper, M. Chong, I. Mareels, and C. Leckie, "Linear system security – detection and correction of adversarial attacks in the noise-free case," *ArXiv e-prints*, Nov. 2017.

[24] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Distributed fault detection and isolation resilient to network model uncertainties," *IEEE Transactions on Cybernetics*, vol. 44, no. 11, pp. 2024–2037, Nov 2014.

[25] S. B. Andrade, M. Pignati, G. Dan, M. Paolone, and J.-Y. Le Boudec, "Undetectable PMU timing-attack on linear state-estimation by using rank-1 approximation," *IEEE Transactions on Smart Grid*, 2016.

[26] J. Milošević, T. Tanaka, H. Sandberg, and K. H. Johansson, "Exploiting submodularity in security measure allocation for industrial control systems," in *Proceedings of the 1st ACM Workshop on the Internet of Safe Things*, ser. SafeThings'17. New York, NY, USA: ACM, 2017, pp. 64–69.

[27] S. Barnum, "Common attack pattern enumeration and classification (capec) schema description."

[28] C. Neureiter, G. Eibl, D. Engel, S. Schlegel, and M. Uslar, "A concept for engineering smart grid security requirements based on SGAM models," *Computer Science - Research and Development*, vol. 31, no. 1, pp. 65–71, May 2016.

[29] C. G. Rieger, D. I. Gertman, and M. A. McQueen, "Resilient control systems: Next generation design research," in *2009 2nd Conference on Human System Interactions*, May 2009, pp. 632–636.

[30] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the impact of stealthy attacks on industrial control systems," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 1092–1105.

[31] D. Umsonst, H. Sandberg, and A. A. Crdenas, "Security analysis of control system anomaly detectors," in *2017 American Control Conference (ACC)*, May 2017, pp. 5500–5506.

[32] K. R. Fall and W. R. Stevens, *TCP/IP illustrated, volume 1: The protocols*. Addison-Wesley, 2011.

[33] "Transmission Control Protocol," RFC 793, Sep. 1981. [Online]. Available: https://rfc-editor.org/rfc/rfc793.txt

[34] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-friendly High-speed TCP Variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, Jul. 2008.

[35] D. J. Leith, R. N. Shorten, and G. McCullagh, "Experimental evaluation of Cubic-TCP," in *International Workshop on Protocols for Fast Long-Distance Networks*, 2008.

[36] I. Abdeljaouad, H. Rachidi, S. Fernandes, and A. Karmouch, "Performance analysis of modern TCP variants: A comparison of Cubic, Compound and New Reno," in *2010 25th Biennial Symposium on Communications*, May 2010, pp. 80–83.

[37] M. Krasnyansky and M. Yevmenkin, "Universal TUN/TAP device driver," Linux Kernel Documentation, Tech. Rep., 2010. [Online]. Available: https://www.kernel.org/doc/Documentation/networking/tuntap.txt

[38] AIT LabLink. [Online]. Available: https://www.ait.ac.at/en/research-fields/smart-grids/network-operators-and-energy-service-providers/ait-lablink/

[39] DIgSILENT. [Online]. Available: https://www.digsilent.de/

[40] Resin.io. [Online]. Available: https://resin.io

[41] Eclipse hawkBit. [Online]. Available: https://projects.eclipse.org/projects/iot.hawkbit