Design, Evaluation, and Optimization of Communication Architectures for Smart Grids

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Michael Jürgen Höfling
aus Würzburg

Tübingen
2016

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	24.02.2017
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Michael Menth
2. Berichterstatter:	Prof. Dr. Andreas Zell

Meinen Eltern in größter Dankbarkeit gewidmet.

Kurzfassung

Zukünftige Stromnetze (Smart Grids) erlauben durch den Einsatz von Informations- und Kommunikationstechnik eine bessere Steuerung des Stromnetzes als es in der Vergangenheit möglich war. Teilziele sind eine effiziente Nutzung erneuerbarer Energien zu ermöglichen, die Versorgungssicherheit zu gewährleisten und neue Strommarktstrukturen zu unterstützen. Dazu müssen Daten zwischen verschiedensten existierenden und neuen Anwendungen im Smart Grid ausgetauscht werden. Um dies zu erleichtern, wird eine flexible Kommunikationsplattform benötigt. Wie diese Kommunikationsplattform aussehen soll ist eine zentrale Frage dieser Arbeit. In Simulationen, Analysen und Prototypen werden Konzepte für ausfallsichere Kommunikation, Integration von existierenden und neuen Smart Grid Anwendungen, sowie sichere Kommunikation über nicht vertrauenswürdige Infrastrukturen entworfen, evaluiert und optimiert. Darüber hinaus werden am Beispiel der elektrischen Lastverteilung die Kommunikationsdynamiken einer Handelsplattform für den zukünftigen Strommarkt analysiert. Als weitere notwendige Bausteine für das Smart Grid werden intelligente Stromzähler (Smart Meter) und deren Kommunikationskomponenten (Smart Meter Gateways) untersucht.

Abstract

Future electrical power grids (smart grids) use information and communication technology to improve power system control beyond what has been possible in the past. The objectives are, among others, to enable efficient use of renewable energy resources, to ensure security of electricity supply and to support future energy market structures. This requires data exchange between various legacy and future applications in the smart grid. To facilitate this, a flexible communication platform is required. A central question of this work is what such a communication platform should look like. Concepts for resilient communication, integration of legacy and future smart grid applications, and secure communication over untrusted infrastructures are designed, evaluated and optimized using simulations, analysis and prototypes. In addition, the communication dynamics of a trading platform for the future retail energy market are analyzed, and smart meters and their communication components (smart meter gateways) are investigated as further important building blocks for the smart grid.

Danksagung

Mehr als sechs Jahre wissenschaftlicher Arbeit liegen nun hinter mir und ich schaue voller Vorfreude und Zuversicht auf die neuen Herausforderungen in meiner beruflichen Laufbahn. Dass ich für diese Aufgaben so gut gewappnet bin, verdanke ich vielen Menschen.

Ich danke meinem Doktorvater Prof. Dr. Michael Menth, der es mir möglich gemacht hat, die vorliegende Arbeit zu verfassen und mir mit seinem Wissen und wertvollen Tipps jederzeit zur Seite stand. Schon früh übertrug er mir Verantwortung, ermöglichte mir die Teilnahme an zahlreichen Projekttreffen und Konferenzen und legte damit den Grundstein für meine fachliche Entwicklung sowie die Zusammenarbeit mit anderen Wissenschaftlern. Ich möchte ihm für das entgegengebrachte Vertrauen danken.

Bedanken möchte ich mich auch bei Prof. Dr. Andreas Zell, der die vorliegende Arbeit begutachtete, und bei Prof. Dr. Thomas Walter und Prof. Dr. Oliver Bringmann, die als Prüfer bei meiner Disputation fungierten.

Meinen Tübinger Kollegen Wolfgang Braun, Jakob Breu, Frederik Hauser, Florian Heimgärtner, Alfons Martin, Mark Schmidt, Andreas Stockmayer und Sebastian Veith danke ich für die tolle Gemeinschaft und die inspirierende Zusammenarbeit in Form von vielen fachlichen, manchmal hitzig geführten Diskussionen. Insbesondere bedanke ich mich bei Florian Heimgärtner, mit dem ich mehrere Jahre ein Büro und Whiteboard geteilt habe, und bei Mark Schmidt, der mir administrativ immer zur Seite stand. Ganz herzlich möchte ich mich auch bei meinen ehemaligen Würzburger Kollegen Dr. Thomas Zinner und Dr. Matthias Hartmann bedanken, die mir bei der Anfertigung der Dissertation mit Anregungen und Kommentaren zur Seite standen.

Das C-DAX Projekt lieferte die Grundlage für große Teile meiner Arbeit. Als Leiter des Arbeitspakets für Leistungsbewertung möchte ich allen beteiligten Projektpartnern danken, die durch ihre Beiträge das Projekt zu einem gemeinschaftlichen Erfolg werden ließen. Stellvertretend danke ich Dr. Marina Thottan, Thierry Pollet, Dr. Young-Jin Kim, Marcel Mampaey und Michel Hasz von Alcatel-Lucent, Herman Bontius und Wilfred Smith von Alliander, Prof. Dr. Mario Paolone und Dr. Paolo Romano von EPFL, Prof. Dr. Chris Develder und Dr. Matthias Strobbe von iMinds, Jimmie Adolph und Vidar Grönas von National Instruments, Dr. Konstantinos V. Katsaros und Dr. Wei Koong Chai von UCL, Dr. Ning Wang von UNIS, sowie Prof. Dr. Erik Poll von RUN.

Ein besonderer Dank geht auch an Susanna Uresch und Gülsen Ergün-Karagkiozidou für ihre vielfältige administrative Unterstützung, insbesondere bei Dienstreisen und bei der Projektorganisation. Ich konnte mich immer auf ihre herzliche Tatkräftigkeit verlassen.

Besondere Freude hat mir während meiner Zeit am Lehrstuhl die Zusammenarbeit mit 'meinen' Studenten bereitet, ganz besonders mit Cynthia Mills und Daniel Fuchs. Ich bedanke mich für tolle Ideen und großes Engagement in Abschlussarbeiten, Projekten und auch als Hilfskräfte, wodurch meine Arbeiten unterstützt und erleichtert wurden.

Abschließend möchte ich mich herzlich bei meinen Freunden und meiner Familie bedanken, allen voran bei meinen Eltern Elisabeth und Jürgen (†2012), die mir das Informatikstudium ermöglichten und mich in all den Jahren stets unterstützten. Meinen tiefsten Dank möchte ich meiner Partnerin Petra aussprechen für ihren Rückhalt und ihr Verständnis während meiner unzähligen Dienstreisen und Abende am Lehrstuhl.

Contents

1 Introduction

The electrical power grid has evolved from a centralized system with a few large power plants to a scattered system incorporating distributed energy resources (DERs)[1] including weather-dependent renewables. This next-generation electrical power grid is called smart grid (SG) and uses digital information and control technology to improve reliability, security, and efficiency of the electric grid. The main obstacles to the deployment of SG applications are the limited scalability, reliability, and security of today's utility communication infrastructures [39].

A typical example of a SG application requiring scalability is demand-response (DR), i.e., distributed matching of electricity demand and supply inside the grid. It starts with all parties exchanging electricity demand and supply information with each other, and eventually reaching an agreement on the matching. That means, all DR parties need dedicated connections to each DR party if traditional client-server communication is used. Such an approach does not scale for high numbers of parties n because the number of necessary connections grows by $\frac{n \cdot (n-1)}{2}$. A promising and widely used solution for such large-scale many-to-many communication is the publish/subscribe (pub/sub) paradigm [40]. All members of the same conversation (called a *topic*) communicate over a shared communication node (*broker*) which handles message forwarding, i.e., they are decoupled and do not have to keep connection states about each other. Applied to the example, all DR participants have to maintain only one connection to the broker and register for the DR topic.

[1] In this monograph the term DER describes "distributed generation, demand response, and electricity storage connected to the distribution grid" [38].

The *Cyber-secure Data And Control Cloud for power grids* (C-DAX) [41] project adapted the pub/sub paradigm to the needs of SGs. It was funded by the European Commission (EC) in the 7th Framework Programme for Research and Technological Development (FP7). The project aimed at developing a cyber-secure and scalable communication middleware for SGs to facilitate the flexible integration of emerging SG applications, and proves its benefits by suitable use cases, a prototype, and a field trial. Furthermore, it aimed at improving scalability compared to traditional client-server communication and facilitating the development of new communication-based applications by providing a standardized transparent interface [39, 40, 42].

The objectives of this monograph are as follows. The first goal is to provide a thorough understanding about the structure of a SG, its typical applications, and the involved communication parties. We then investigate and optimize the performance of the SEcure Data-centric Application eXtension (SeDAX) middleware which initially served as blueprint for the C-DAX architecture. Furthermore, we design and optimize the novel C-DAX middleware, and compare it to alternative approaches. An additional objective is to investigate the future retail energy market (REM) as exemplary SG use case by evaluating the communication performance of the PowerMatcher (PM) trading platform and by proposing an experimentation framework for smart metering communication.

In the following, a detailed description of the scientific contribution in this monograph is given. The chapter is concluded by an outline of the remainder of this monograph.

1.1 Scientific Contribution

This section summarizes the contributions of the author in the field of smart grid communications. It gives an overview of the content of the studies presented in this monograph and explains their relations. Subsequently, it provides a summary of the author's contributions beyond this monograph. All studies are based on scientific publications of the author.

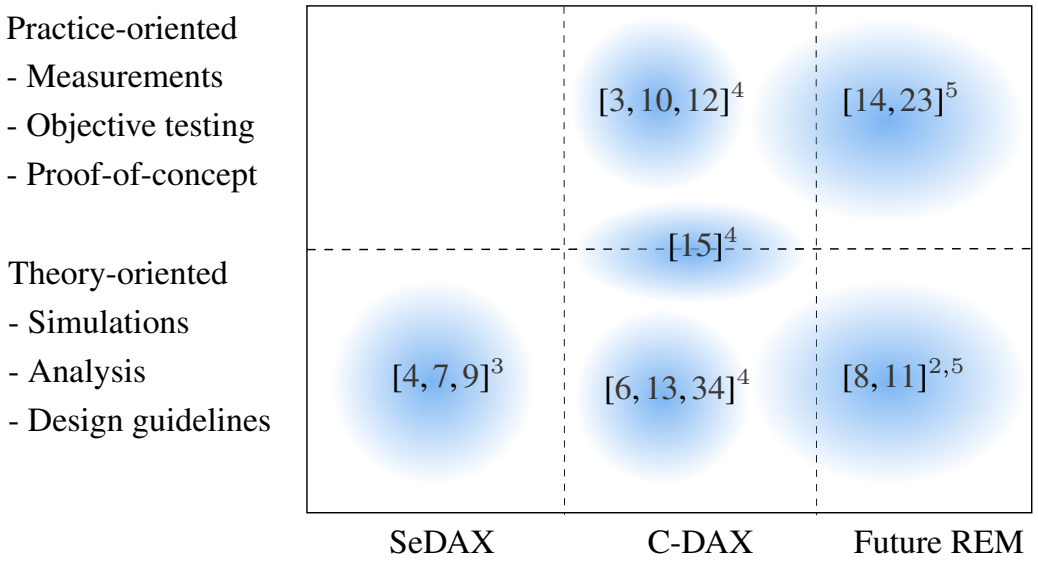


Figure 1.1: *Contribution of this work illustrated as a classification of the research studies conducted by the author. The notation* $[x]^y$ *indicates that the scientific publication* $[x]$ *is discussed in Chapter* y *of this monograph.*

Figure 1.1 gives an overview of the contribution of this work. The individual research studies carried out during the course of this work are classified according to their used methodology. In particular, they are classified with respect to practice-oriented methods like measurements, objective testings, and proof-of-concept implementations, or theory-oriented methods like simulations, analysis, and design guidelines. The respective focus of the studies cover SeDAX, C-DAX, and future REM. The markers $[x]^y$ indicate the scientific publications $[x]$ which provide the basis for Chapter y.

The first major contribution presented is the performance evaluation of the SeDAX architecture to identify potential resource management issues. In the original SeDAX architecture, geographic hashing determines the coordinates of topics on the Delaunay-triangulated (DT) overlay, i.e., the mapping of topics to storage nodes. We showed that this static assignment of topics to coordinates can lead to severe load imbalance on SeDAX nodes and developed a Monte-Carlo op-

timization for node placement in SeDAX to minimize storage requirements. We further derived the storage requirements of SeDAX under optimal conditions and showed that they exceed those of an idealized storage system. In a next step, we proposed a modification allowing dynamic reassignment of topics to coordinates while retaining the benefits of SeDAX, i.e., resilient overlay forwarding, decentralized control, and the ability to cope without a mapping system. We developed load balancing algorithms and demonstrated that they work well for static topic sizes. We further showed that a balanced SeDAX system may run out of balance if topic sizes change over time. Therefore, we presented a distributed algorithm for continuous load balancing offering a single parameter to trade off load balancing quality against load balancing effort in terms of moved load rates. In our evaluations, it kept a balanced system well balanced when topic sizes grew exponentially over time with different rates.

The second major contribution presented in this monograph considers the design of the novel C-DAX architecture as a mean to overcome, among others, the resource management issues of the original C-DAX blueprint SeDAX architecture. Our evaluations showed that SeDAX is inflexible with regard to resource management and that those issues are inherent to its design. We clarify the need for a novel C-DAX architecture, specify the initial architecture, and detail the core features of the architecture. The new architecture is a cyber-secure pub/sub middleware tailored to the needs of SGs, offering end-to-end security, and scalable and resilient communication among participants in a SG. It further includes enhanced real-time application support, and transparent support for legacy SG communication protocols. We describe a simulation of C-DAX in the OMNeT++ simulation framework and the prototype implementation. Furthermore, we analyze the strength and weakness of the C-DAX architecture with respect to alternative communication solutions. Eventually, we give recommendations for the potential re-use of C-DAX concepts and components in other architectures.

The third part of this monograph presents the future REM as a use case for SG communication. Our well-educated assumption is that the future REM will have many more participants and see more volatile prices than today, creating the need

for new communication and trading infrastructures [43–45]. Accordingly, we review the PowerMatcher (PM) as a possible approach for such a trading infrastructure, and analytically evaluate its communication characteristics. Our results show that PM enables scalable retail energy transactions (RETs) with millions of participants requiring only moderate resources on the communication's side. Besides a trading infrastructure, advanced metering infrastructures (AMIs) in the distribution grid (DG) are necessary as an enabling technology to provide automatic billing, and acquisition of network status data. Different standards and communication protocols exist for smart metering, ranging from transmission protocols to architectural recommendations. We present the concept of the German AMI as defined in BSI TR-03109 [46], review implementations of smart metering protocols and architectures, and provide a Java-based open-source smart meter gateway experimentation framework (jOSEF).

Beyond this monograph, the author contributed to several other studies and projects in the field of future networks, data center monitoring, and IT education. We delved into future Internet research and developed a taxonomy for mapping systems for locator/identifier split Internet routing architectures. We provided a comprehensive review of recently proposed mapping systems and classified them into our proposed categories [2, 18]. Based on our extensive literature study on mapping systems, we developed FIRMS [1, 17–19], a fast, scalable, reliable, and secure future Internet routing mapping system. During our work on future Internet protocols, we proposed improvements to the LISP mobile node architecture [5,20]. Together with "'science+computing ag"' in Tuebingen, Germany, we developed the InfiniBand performance monitoring tool (IBPM) [21]. Our tool analyzes InfiniBand data center networks and presents a comprehensible visualization of the performance and health of the network. InfiniBand network operators can use the tool to detect potential bottlenecks and optimize the overall performance of their network. Finally, we improved the hands-on networking courses offered at the University of Tuebingen by moving from an outdated physical infrastructure to a modern virtualized infrastructure while preserving the ability for students to physically interact with the networking experiments [22].

1.2 Thesis Outline

The remainder of this monograph is structured as follows. Chapter 2 introduces the technological background. Chapter 3 investigates the performance of the SeDAX architecture using simulations, and analytical modeling to highlight key resource management issues of the original architecture as well as proposing and evaluating mechanisms for distributed load balancing. In Chapter 4, the final C-DAX architecture is presented including a detailed description of the C-DAX core features, the proof-of-concept and a comparison of C-DAX to alternative approaches. Chapter 5 focuses on the future REM. In particular, we estimate the amount of signaling traffic of the PM trading infrastructure. Finally, we discuss how AMI will be deployed in Germany and propose the jOSEF as a tool for further experimental exploitation. Chapter 6 summarizes this work and draws conclusions. A table with abbreviations is provided in the appendix.

2 Technological Background

This chapter introduces the technical background of this monograph. After a short overview on the structure of power grids, we give a perspective on the future retail energy market (REM) as published in [8]. Then, we briefly introduce the pub/sub paradigm as published in [12] and present the C-DAX project. Subsequently, we describe the smart grid (SG) use cases which were considered in the C-DAX project. At the end of this chapter, we provide a brief outlook on selected SG research projects.

2.1 Structure of Power Grids

Power grids are hierarchically structured. They can be divided broadly into three different domains: power generation, power transmission, and power consumption. In addition to the domains, there are four different voltage levels: extra high-voltage (EHV), high voltage (HV), medium voltage (MV), and low voltage (LV). Substations transform between the voltage levels. Figure 2.1 shows the structure of a typical power grid.

The *power generation* domain consists of power plants, e.g., coal, nuclear, or hydro-electric plants, but also DERs, e.g., wind farms or photo-voltaic (PV) panels. The transmission grid transports power over long distances, sometimes even across international borders. The distribution grid (DG) facilitates regional distribution of power. Combined, both grids form the *power transmission* domain. The *power consumption* domain covers all service locations consuming power, e.g., industrial consumers and residential buildings. Prosumers are special entities which belong to both the power generation and the power consumption domain.

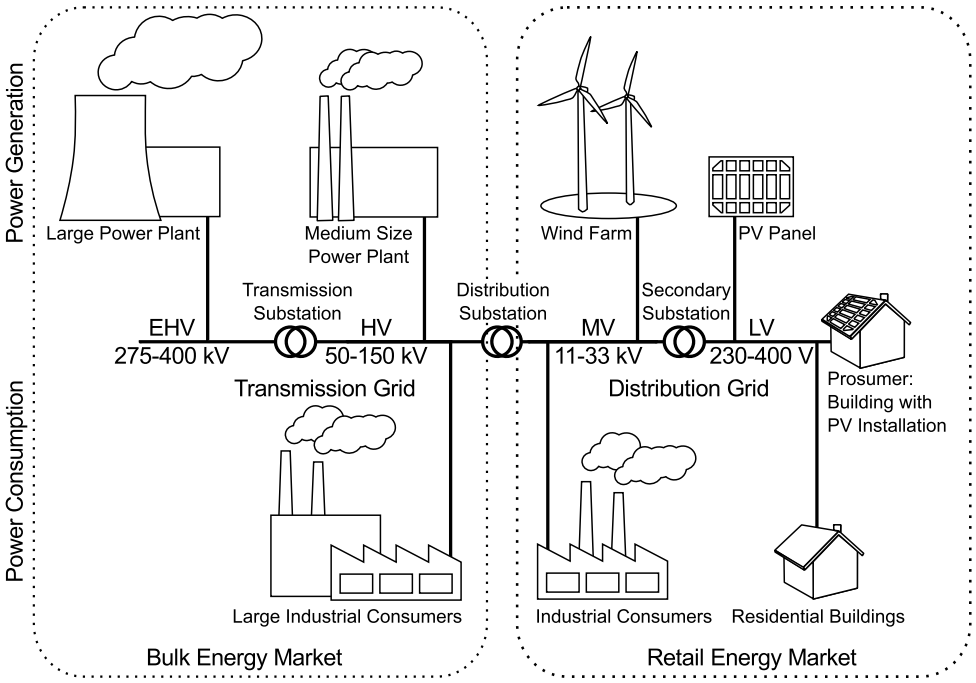


Figure 2.1: *The structure of a typical power grid including the general boundaries of the electrical energy market. Power is generated at the top, transferred over the transmission and distribution grid, and consumed at the bottom. Prosumers are positioned in-between the generation and consumption domain as they are part of both domains. Voltage levels decrease from left to right, i.e., from EHV level to LV level.*

They may produce power and feed-in to the grid, but they may also consume power. The normal power flow is unidirectional: top-down from the generation domain to the consumption domain, and from left to right in the transmission domain in Figure 2.1, i.e., from EHV level to LV level. With the increasing number of DERs, bidirectional power flow inside the transmission domain is possible, e.g., from LV to MV level.

2.2 Bulk and Retail Energy Market

We now take a closer look at today's electrical energy market and its market mechanisms. From an economic point of view, electrical energy is a commodity which can be bought, sold, and traded. Depending on which participants interact with each other on what voltage level, we differentiate between two markets: bulk energy market (BEM) and retail energy market (REM). Figure 2.1 illustrates the boundaries of BEM and REM. In practice, there is no sharp border between both markets.

2.2.1 Market Structures

The *BEM*, sometimes referred to as wholesale market, consists of three major participants on the EHV, HV, and MV level of the power grid: suppliers of energy, retailers, and large consumers. Competing suppliers of energy offer their electrical energy on the BEM to retailers or large consumers of electrical energy, e.g., aluminum plants. Large consumers buy electrical energy through the BEM directly. Energy trading normally takes place on trading platforms similar to the stock exchange. However, BEM transactions are also possible without involving a trading platform. An example for a BEM trading platform is the European Energy Exchange (EEX) [47], which spans Germany, France, Austria, and Switzerland. Typical time scales for BEM transactions on the EEX vary between hours and years.

The REM consists of two major participants on the MV and LV level of the power grid: retailers and clients. *Retailers* buy electrical energy on the BEM, and resell it through the REM to clients not participating in the BEM. *Clients* buy or sell electrical energy on the REM. Examples for clients are consumers, prosumers, and DERs. The REM enables clients to choose their electrical energy supplier from competing retailers. In contrast to the BEM, energy on the REM is not traded directly between all participants but indirectly through the retailer. That is, clients can buy or sell energy only through retailers.

2.2.2 Retail Energy Transactions

All transactions between consumers of energy and suppliers of energy on the REM are called RETs. Today's RETs include three consecutively executed phases: retailer selection by clients, delivery of electrical energy, and accounting for the delivered electrical energy. While the meaning of each phase is self-explanatory their exact realization in today's REMs is subject to country-specific legislation. Today's RETs are based on fixed-price contract models, i.e., a client buys or sells a certain amount of electrical energy at a fixed price per energy unit for a specified period on the REM. The time scale of today's RETs is given by the accounting period of the electricity contract, e.g., one month, one year, or even longer. However, no generally agreed fixed time scale for today's RETs is given in the literature.

2.2.3 Common Market Trading

Depending on the covered time period, the actual energy trading takes place on specific markets which are distinguished in derivatives market, spot market and intra-day market. Parts of the following description are adapted from [48]. The derivatives market is standardized in monthly, quarterly, and annual contracts, and trades electricity for the coming years. The objective of the spot market is to compensate for production and consumption of the coming day; it is also called day-ahead market. The intra-day market allows to react on deviations of actual load and generation after the spot market trading has already been completed. The EEX allows intra-day trading down to 30 minutes before delivery; over-the-counter intra-day trading is possible down to 15 minutes before delivery. Subsequent financial compensation is called day-after trading. Besides the actual demand and supply, operating reserves are traded as well. The latter are a crucial concept to ensure the grid's operation in case of variations in the load profile or if faults occur [49].

2.3 Publish/Subscribe Basics

We now review components and signaling in typical pub/sub systems. The basic idea of the pub/sub paradigm is the decoupling of communication partners in space, time, and synchronization. The goal is to improve scalability compared to traditional client-server communication, and to facilitate development of new communication-based applications by providing a standardized transparent interface [39,40]. The pub/sub paradigm is similar to the well-known *observer-pattern* in software engineering [50].

A pub/sub communication architecture consists of at least four components: publishers, subscribers, brokers, and a broker discovery service. The actual pub/sub communication can be topic-oriented, content-based, or type-based [40]. We are interested in topic-oriented pub/sub communication only. In this context, a topic is an abstract representation of a unidirectional information channel, and is addressed using its unique name and probably attributes, e.g., data type, location, and time. An example for a topic is measurement data for a specific geographic region inside the DG. First, *publishers* and *subscribers* register with a *broker* for a certain topic. Then, publishers send messages for that topic to that broker which forwards them to the subscribers. A *broker discovery service* tells publishers and subscribers what broker supports the communication on a certain topic. Depending on deployment, the functionality of brokers may be collocated with publishers or subscribers.

The signaling interactions of typical pub/sub architectures are illustrated in Figure 2.2. In the Step 1, publishers and subscribers query the broker discovery service to find the appropriate broker for further communication on a certain topic. In Step 2, publishers and subscribers send a join message to the broker; the message also indicate the role of publishers and subscribers. After successful join, publishers may start sending data to the broker in Step 3. In Step 4, the broker eventually starts forwarding data to the registered subscribers in Step 4.

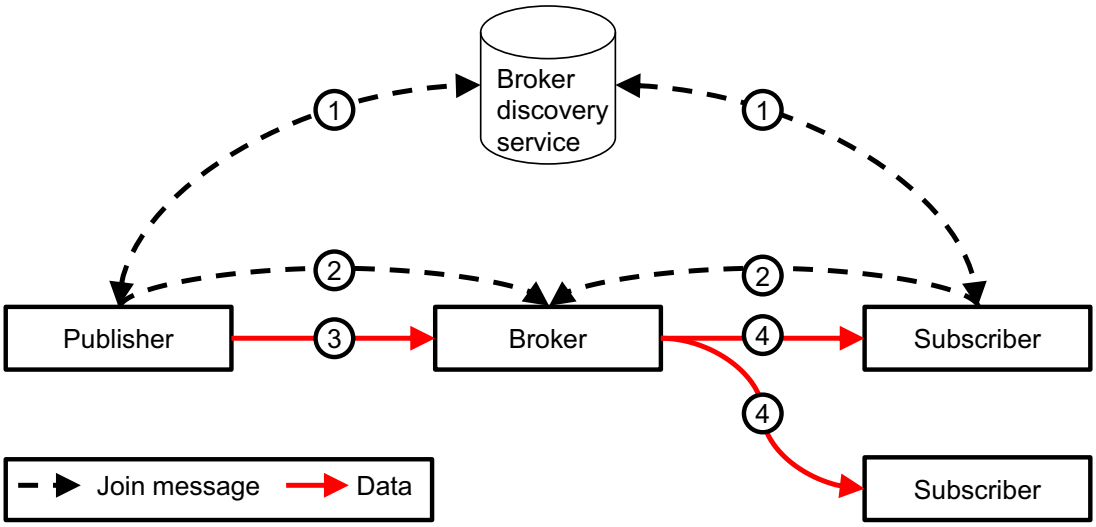


Figure 2.2: *Basic pub/sub communication. After successful broker discovery (Step 1) and client join (Step 2), publishers send data to brokers (Step 3) which forward data to subscribers (Step 4).*

2.4 The C-DAX Project

The Cyber-secure Data And Control Cloud for power grids (C-DAX) project [41] is an FP7 project funded by the EC. It aims to develop a cyber-secure communication middleware for SGs, applying the pub/sub paradigm to enable scalable, transparent, and secure end-to-end communication [13, 51] between publishers and subscribers. Additionally, the C-DAX architecture provides resilient communication [10], advanced communication modes [15], inter-domain communication, and support for real-time applications [3]. The foundations of the C-DAX architecture are based on the initial SeDAX concepts developed at Alcatel-Lucent Bell Labs [39, 52, 53]. We review relevant aspects of the SeDAX architecture in Section 3.1, and give a motivation for and a detailed description of the current C-DAX architecture in Chapter 4.

The C-DAX project originally investigated specific use cases in three application domains: LV pervasive DERs, MV DERs and islanding, and RETs. During the course of the project, the actual project use cases have been refined to better map the project partner's areas of expertise. We will detail the refined use cases in the next section. A prototype of the C-DAX architecture is implemented, both as simulation and as field trial. The latter evaluated the real-time performance of C-DAX when used in LiveLab [54], a power distribution network owned by Alliander, a member of the C-DAX consortium.

2.5 Considered Use Cases

The C-DAX project targets three use cases on the MV and LV level of the DG which can be summarized as follows. The short descriptions are adapted from the official project reports in [24, 25].

- *Use case 1 (UC1)* covers grid controlling normal operating network stability in MV networks. For this use case we consider the communication between remote terminal units (RTUs) and intelligent electronic devices (IEDs) in distribution substations with supervisory control and data access (SCADA) master control and other systems in the utility distribution control center (DCC).
- *Use case 2 (UC2)* covers monitoring the DG. For this use case we consider the communication between the phasor measurement units (PMUs) deployed along the MV distribution lines and phasor data concentrators (PDCs) located at the distribution substations and other communication required for distribution management implementation based on state estimation using the PMU measurements.
- *Use case 3 (UC3)* covers retail and market facilitation. For this use case we consider RETs between the consumers of energy and owners of distributed generation including those owned and located at consumer premises. These transactions facilitate the matching of demand with supply and/or the operation of DR mechanisms.

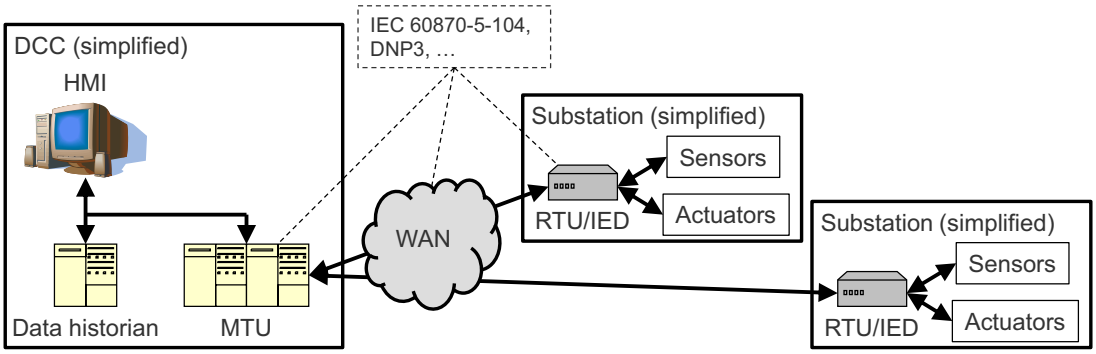


Figure 2.3: *Example of a SCADA system. RTUs connect sensors and actuators in the field to the HMI in the DCC.*

In the following, we give a more detailed use case description including SG actors, communication profiles (e.g. data volume, message rate) and requirements (e.g. delay), and a brief overview of the security aspects for each use case. The remaining content of this subsection is mainly taken from studies and public project reports that we published and presented in [10, 12, 13, 15, 24, 25, 30].

2.5.1 Use Case 1: Telecontrol

SCADA systems are used by utilities for collecting electrical power grid data at periodic intervals as well as reporting asynchronous events in the grid based on detected faults, and for automatically controlling operations of actuating elements. Examples for electrical power grid data are measurements of voltages and currents at several points in a substation. Examples for asynchronous events are alarms, and examples for actuating elements are circuit breakers. The currently widely-used communication standard IEC 60870-5-104 [55] defines RTUs which are deployed at the substations, and which communicate over transmission control protocol (TCP)/Internet protocol (IP) with a SCADA *master control* and other systems in the utility's DCC.

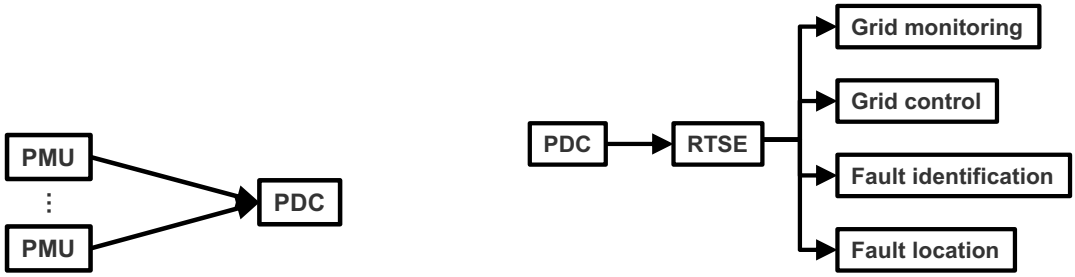
Figure 2.3 illustrates an example of a SCADA system with one DCC and two substations. RTUs are responsible for collecting all measurement data and generated events in a substation, and for eventually forwarding them to the DCC. Additionally, RTUs receive control signals from the SCADA system in the DCC and forward them to the actuators in the substation. On the DCC side, data is handled by a master terminal unit (MTU) that further communicates with an HMI and a SCADA historian. The latter is responsible for recording and maintaining a historical database of the SCADA measurements. Substations are evolving to support the IEC 61850 set of standards [56] that will eventually replace an RTU and associated substation equipment with IEDs.

SCADA communication requires bidirectional client-server communication, irrespective of the underlying communication protocol, i.e., RTUs need to be able to send data to and receive data from the DCC and vice versa. This makes direct integration of SCADA applications in a traditional pub/sub system difficult or even impossible without modifications to the actual SCADA software. We will show in Section 4.2.2 how C-DAX enables transparent integration of legacy applications such as SCADA.

IED/RTU data is not privacy-sensitive information, thus the most important security requirement that must be enforced here is end-to-end integrity. Irrespective of the type of pub/sub based communication, all the messages exchanged between communicating parties shall be either digitally signed or protected by a message authentication code (MAC).

2.5.2 Use Case 2: Synchrophasor-Based Real-Time State Estimation of Active Distribution Networks

Currently, the lack of available distributed measurement infrastructures at the DG level represents one of the main obstacles for distribution network operators (DNOs) to develop adequate controls capable to enable the seamless integration of DERs. Within this context, one of the most promising technologies for monitoring such active distribution networks (ADNs) is associated to the concept of the synchrophasor-based real-time state estimation (RTSE) [57–62].



(a) An example of an IEEE C37.118 synchrophasor network.

(b) Examples of several power-system applications that are based on RTSE.

Figure 2.4: *Synchrophasor-based RTSE of ADNs.*

The technical base components of this technology are PMUs and PDCs as depicted in Figure 2.4(a). PMU devices measure the equivalent phasor representation of the power system waveforms (i.e., voltages and currents) in different points of the power grid. The measurement data are accurately time-stamped using a reliable time source, such as the coordinated universal time derived from the global positioning system (UTC-GPS), and sent to the PDC with a refresh rate of up to 50 times per second [63, 64]. PDCs receive, time-align and aggregate measurement data from different PMUs based on the time-stamp, and provide the aggregated data to the RTSE application. In turn, this feeds the time-aligned and aggregated measurement data into a mathematical model of the DG to estimate the current state of the grid. The outcome of the estimation may be used by several power-system applications as depicted in Figure 2.4(b), e.g., grid monitoring and control, and fault identification and location. Compared to traditional SCADA systems, synchrophasor-based RTSE allows estimating the system's state with increased accuracy, high refresh rate and reduced time latencies, providing DNOs a complete and real-time view and control of their ADNs.

Today, PMU measurement technology is already deployed on the transmission grid level in several countries around the world, e.g., the North American SynchroPhasor Initiative (NASPI) operates a large-scale measurement infrastructure called NASPInet [65, 66], or the Synchrophasor Initiative in India [67]. Still,

PMU measurement technology has not been widely deployed on the DG level yet. As part of the C-DAX field trial, PMUs have been installed in a real-world DG and transferred their measurement data using the C-DAX middleware to a PDC in the DCC; eventually RTSE was performed. More information on the C-DAX field trial will be given in Section 4.3.2.

PMUs continuously stream their measurements with a refresh rate of 50 times per second to the PDCs. The IEEE C37.118 standard allows for higher and lower data rates, too. In general, the acceptable delay between PMUs and PDCs depends on the SG application that is interested in the data. For the RTSE to work properly, the underlying network has to provide low latency and low jitter. That means, any additional middleware between the physical network underlay and the SG application must not add significant processing delay. Furthermore, the IEEE C37.118 standard offers two different modes for client-server communication, but cannot be used unchanged over pub/sub communication architectures. In Section 4.2.4, we provide an adapter-based solution to easily connect and integrate entities in a synchrophasor network over the pub/sub C-DAX communication architecture.

End-to-end confidentiality is not always going to be an issue in this use case, i.e., encryption is not always required. However, end-to-end integrity is of utmost importance. Any unauthorized modification of the content of the messages sent from the PMUs to the PDC/RTSE can damage the entire grid operation. Due to the stringent allowed time delays in this use case, end-to-end integrity of very-high priority messages has to rely on MACs. Symmetric keys have to be generated and distributed by a security server to all clients publishing/subscribing very-high priority topic data.

2.5.3 Use Case 3: Retail Energy Transactions

In the future REM, any participant will be able to trade energy. Consumers will have dynamic pricing based on predicted supply and demand instead of a fixed-price contract model [68]. Electricity trading intervals will be on the order of minutes or hours, i.e., significantly shorter than today's accounting intervals [45].

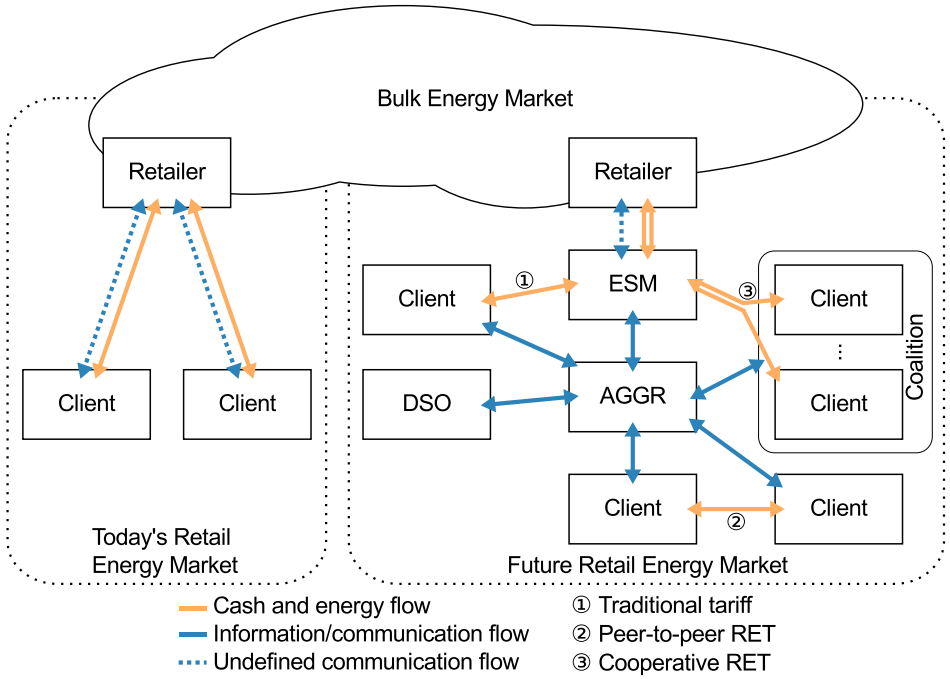


Figure 2.5: *Cash, energy, and communication flows in today's and the future REM. In today's REM, clients can only sell or buy energy through retailers. Trading between clients is only possible indirectly using retailers. In the future REM, in addition to traditional tariffs (1), AGGRs enable clients to directly trade their energy with each other (2). Groups of clients may form coalitions and participate in collaborative RETs (3), e.g., to maximize profit. ESMs guarantee energy balance inside DGs while DSOs verify physical constraints of RETs.*

As a consequence, the future REM will have many more participants and see more volatile prices than today. In the literature there are various definitions of future market participants and their functions [24, 43, 45, 69–71]. We provide a unified view thereof in Figure 2.5. The figure shows what a future REM may look like compared to today's REM. Besides additional participants, cash and energy flows, and communication flows will change.

The future REM comprises of five classes of participants: clients, AGGRs, ESMs, DSOs, and regulators (not shown in the figure). *Clients* in the future REM cannot only buy and sell electrical energy from or to retailers, but they can also trade their electrical energy directly on the REM. They have to provide proper forecasts of their energy demand and supply, possibly based on weather forecasts if their power production is weather-dependent. *AGGRs* supervise demand supply matching (DSM). They mediate between clients for DSM inside the DG, and between clients and ESMs for DSM between the DG and the transmission grid. AGGRs are the only authoritative entity in the future REM to initialize and supervise auctions, and they prevent trades that cannot meet physical constraints. *ESMs* are responsible for balancing the energy in the DG. For example, if the energy demands of DGs exceed their internal production, ESMs acquire additional electrical energy on the BEM to ensure proper energy supply in the DGs. *DSOs* are control instances of DGs. They operate DGs and validate the outcomes of auctions, so-called power transaction plans. That is, if the outcome of an auction would lead to an unstable grid configuration violating physical constraints, the auction is invalidated and AGGRs may be asked to restart the auctions. *Regulators* are independent authorities that determine or approve the electricity market rules, and monitor RETs to ensure compliance with regulations and rules.

Normally, each client acts as an individual participant in a RET. The minimum achievable profit by a single client is given by the so-called *self-value* [72]. The self-value depends on client-specific parameters, e.g., estimated weather-dependent energy production, or the geographical location of the client. The future REM introduces client coalitions to maximize client profits [72–75] or to create efficient virtual power plants (VPPs) [76]. Client coalitions are temporary groups of clients, not necessarily geographically close to each other, pursuing short-term common economic interests. Coalition formation is a distributed process which enables clients to find and agree on potential coalition partners. During coalition formation, each client calculates its self-value and disseminates it to all other clients through the AGGR. Coalition decisions are then made based on the self-values, i.e., each client independently determines whether a coalition

with one or more clients matches its economic objectives. From the market's perspective, coalitions are virtual clients with their own self-value participating in RETs. A VPP is an example for such a client coalition, i.e., prosumers and DERs are aggregated into a virtual equivalent of a large power plant [16]. Coalitions are included here because they are an active research area, but RETs are possible without coalitions as well, i.e., coalitions are an optional feature. We will use the term clients interchangeably for both clients and coalitions.

The future REM supports three different types of future RETs: traditional tariff, peer-to-peer (P2P), and collaborative. Traditional tariff RETs are comparable with today's RETs based on fixed-price contracts. However, communication flows for traditional tariffs differ as shown in Figure 2.5. Clients communicate with retailers through AGGRs and ESMs. P2P RETs [43, 77] are direct transactions between two clients which have been coordinated using the AGGR. Collaborative RETs [73–76] are transactions between coalitions and clients, or coalitions and coalitions.

In contrast to today's RETs, the retailer selection phase is replaced by a two-stage process consisting of *coalition formation* and *auctions* in future RETs. Coalition formation is optional. The auction phase between clients is initialized and coordinated by the AGGR. That is, each client sends its demand and supply prediction to the AGGR which then matches the received demands and supplies. The outcome of the auction is a *power transaction plan* which is sent to the DSO for approval considering the physical constraints of the DG. If the approval is successful, the AGGR sends a binding agreement to the clients. After the delivery of electrical energy, the accounting phase matches actual demands and supplies with their originally predicted values. Clients which did not fulfill their demand or supply prediction are penalized.

Contrary to UC1 and UC2, this use case requires end-to-end confidentiality between communicating parties since it deals with private consumer data. Publishers have to encrypt and sign or MAC all exchanged data. Encryption can be either symmetric or asymmetric. Asymmetric encryption requires a key pair: the public key is used for encryption and the secret key for decryption. On the other hand, on

symmetric encryption the same key is used for both encryption and decryption. Since in this use case speed is not an issue, one can rely on asymmetric schemes to enforce end-to-end confidentiality between publishers and subscribers. The National Institute for Standards and Technology (NIST) report *Guidelines for Smart Grid Cyber Security* [78] recommends the use of specific cipher suites for public-key encryption.

2.6 Related Smart Grid Research Projects

In the following, we briefly introduce four FP7-funded and three country-funded SG projects that we consider complementary to the C-DAX project. An almost complete overview on European SG projects can be found in the very comprehensive *Smart Grid Projects Outlook* 2014 [79], provided by the EC Joint Research Centre. Parts of the FP7 project descriptions were adapted from information available in the EC Community Research and Development Information Service (CORDIS) [80]; the other project descriptions have been adapted from the projects' websites.

The FP7 INCREASE project [81] focuses on managing renewable energy sources in LV and MV networks, to provide ancillary services[1] towards DSOs and transmission system operators (TSOs), in particular voltage control and the provision of reserve. It investigates the regulatory framework, grid code structure and ancillary market mechanisms, and proposes adjustments to facilitate successful provisioning of ancillary services that are necessary for the operation of the electricity grid, including flexible market products. INCREASE enables DERs and loads to go beyond just exchanging power with the grid which will enable the DSO to evolve from a congestion manager to capacity manager[2]. They assume that this may result in a more efficient exploitation of the current grid capacity, thus facilitating higher DER penetration at reduced cost. A simulation platform enables the validation of the proposed solutions and provides DSOs with a tool

[1] Ancillary services are a general term in power systems and comprise any operations beyond generation and transmission that are required to maintain grid stability and security.

[2] Congestion and capacity here refer to the power system, not the communication system.

for investigating the influence of DERs on their DG. Furthermore, the solutions are validated by lab tests and in three field trials in real-life operational DGs in Austria, Slovenia and the Netherlands.

The FP7 SUNSEED project [82] proposes an evolutionary approach to utilize already present communication networks from both energy and telecom operators to form a converged communication infrastructure for future smart energy grids offering open services. SUNSEED analyzes the regional overlap of energy and telecommunications operator infrastructures and identifies vital DSO energy and support grid locations (e.g. distributed energy generators, transformer substations, cabling, ducts) that are covered by both energy and telecom communication networks. According to SUNSEED, interconnection assures secure end-to-end communication on the physical layer between energy and telecom, whereas interoperation provides network visibility and reach of SG nodes from both operator (utility) sides. Monitoring, control and management gathers measurement data from a wide area of sensors and smart meters (SMs) and assures stable DG operation by using novel intelligent real time analytical knowledge discovery methods. SUNSEED further proposes the development of applications build on open standards with exposed application programming interfaces (APIs) to third parties to enable the creation of new businesses related to energy and communication sectors. Finally, the project claims that its approach leads to much lower investments and total cost of ownership for future smart energy grids with dense distributed energy generation and prosumer involvement.

The FP7 SmartC2Net project [83] aimed[3] at developing, implementing, and validating robust solutions that enable SG operation on top of heterogeneous off-the-shelf communication infrastructures with varying properties. They designed mechanisms for adaptive network and grid monitoring, strategies to control communication network configurations and Quality of Service (QoS) settings, and extended information models and adaptive information management procedures. They further investigated how power control algorithms can benefit from improved awareness of the communication network properties and their impact on

[3] The SmartC2Net project officially ended on 2015-11-30.

information quality. The project results were validated in representative use-cases of the active operation of DERs connected to MV and LV DGs, and investigated in three complementary lab prototypes. SmartC2Net showed that intelligent DG operation can be realized in a robust manner over existing communication infrastructures even despite the presence of accidental faults and malicious attacks.

The FP7 eBadge project [84] proposed[4] an optimal pan-European intelligent balancing mechanism to integrate VPPs by means of an integrated communication infrastructure that can assist in the management of the electricity transmission grids and DGs in an optimized, controlled and secure manner. They implemented a simulation and modeling tool to study the integrated *balancing and reserve market*. Subsequently, they developed a unified data-exchange standard for *balancing and reserve entities* on top of the RabbitMQ [85] message bus. Further, they investigated VPP data analysis, optimization and control strategies. These components were then integrated into a pilot *eBadge Energy Cloud* that has been validated through tests in the lab and a real world field trial. Finally, related business models integrating energy, information and communication technology, and residential consumer benefits have been developed and evaluated.

The Belgium SWIFT project [86] investigated[5] several technical and economical active power network management approaches to integrate renewable energy resources (especially wind turbines) to the electricity grid more quickly and cost-efficiently. Their main objective was to maximize the availability of green energy while mitigating the threat of uncontrolled peak production. In particular, they explored the effect of demand-response (DR), dynamic line rating, and fine-grained curtailment. Dynamic line rating facilitates transmitting electricity peaks over cables while avoiding lifetime-reducing damage to the cables by actively monitoring the cables' temperature band. They developed and validated curtailment approaches for fine-grained control of wind turbine electricity output in case of excess wind. The project's findings were evaluated in a real world test bed in the harbor of Antwerp, Belgium.

[4]The eBadge project officially ended on 2015-11-30.

[5]The SWIFT project officially ended on 2015-12-31.

The Belgium MonIEflex project [87] applied[6] data analysis and machine learning techniques to characterize and unlock additional power flexibility in the process industry. They developed a tool that forecasts flexible capacity at industrial consumers which can be freed up in a non-intrusive way and integrated it in the commercial REstore demand supply matching (DSM) [88] platform. Furthermore, they commercialized the project results into a new service in Belgium and the UK. In the UK, the new service helps industrial power consumers to avoid peak consumption while maintaining production during winter which would otherwise result in mandatory fines. In Belgium, the new service helps grid operators to predict if strategic reserves need to be activated to avoid energy shortages in a timely fashion.

The German cooperation network *Virtuelles Kraftwerk Neckar-Alb* (VPP Neckar-Alb) [89] is an incubator for innovative smart energy projects and represents a variety of regional partners from industry and academia located in the south-west of Germany; we are also actively involved here. The *demonstration project VPP Neckar-Alb* [16] is a result of this incubator and aims at building a demonstration site connecting and integrating different VPP building blocks at the Reutlingen University campus. This demonstrator constitutes a flexible environment for research and teaching to investigate interactions between the components. Additionally, it will provide an opportunity for visiting to interested companies, leading to increased acceptability and better understanding.

[6] The MonIEflex project officially ended on 2015-12-31.

3 Performance Evaluation of SeDAX: the C-DAX Blueprint Architecture

In the early phase of the C-DAX project, the SeDAX [39] architecture was discussed and initially used as technological foundation of the C-DAX architecture. SeDAX is a resilient pub/sub information-centric networking (ICN) architecture where publishers send messages to the appropriate message broker over a Delaunay-triangulated (DT) overlay network. Overlay nodes and topics are addressed via geographic coordinates. A topic is stored on primary and secondary nodes, those nodes that are closest and second-closest to the topic's coordinate.

This chapter summarizes our investigations of resource management issues of the SeDAX architecture. In general, SeDAX statically assigns topics to coordinates by hashing the topic's name to a coordinate. If a SeDAX node is accidentally primary or secondary node for too many or too large topics, it may become overloaded in terms of storage capacity. The original SeDAX architecture does not provide any features to take away topic responsibility from such a node.

We first investigate the impact of optimized node placement on storage requirements of overlay nodes. For that, we develop a simple Monte-Carlo optimization for node placement in SeDAX to minimize storage requirements. We evaluate the capacity requirements of SeDAX with optimized node placement for homogeneous and heterogeneous node provisioning. We analytically derive the least storage requirements of SeDAX under optimal conditions and compare them to those of an idealized storage system.

We then develop a topic delegation mechanism to make the assignment of topics to nodes dynamic. Our proposed mechanism is the only existing method to improve the flexibility and resource management of the SeDAX architecture so far. We suggest a distributed resource management system that detects traffic imbalances among SeDAX nodes and re-assigns topics to other coordinates for load balancing purposes.

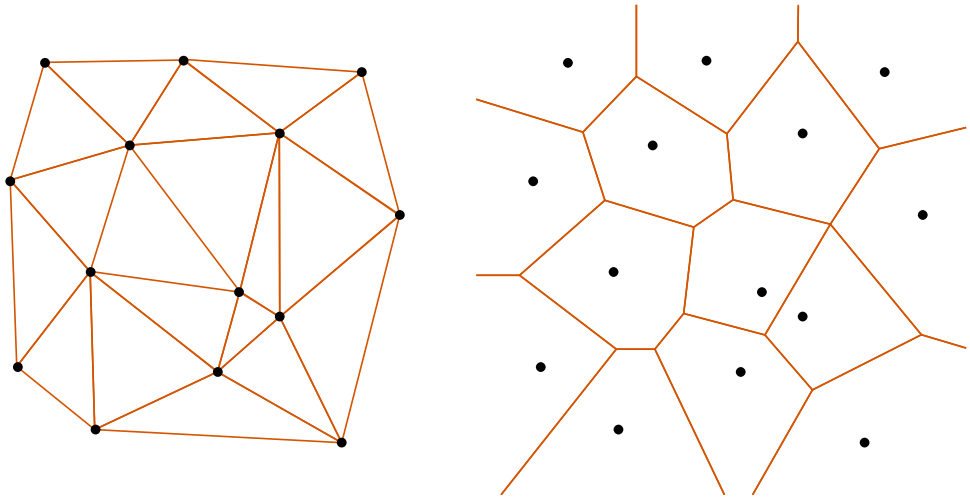
The content of this chapter is mainly taken from [4, 7, 9]. Its remainder is organized as follows. We review relevant aspects of the SeDAX architecture and discuss related work in Section 3.1. We introduce some commonly used definitions and nomenclature in Section 3.2 and investigate the impact of optimized node placement on storage requirements in Section 3.3. Based on these results, we propose an improvement to the resource management system of SeDAX facilitating distributed load balancing in Section 3.4 which is evaluated in Section 3.5. Finally, Section 3.6 summarizes some condensed insights.

3.1 Background and Related Work

We now review relevant parts of the SeDAX architecture necessary to understand the performance evaluation. Then, we discuss related work in the context of data placement and load balancing in ICN, P2P, and distributed computing.

3.1.1 Delaunay Triangulation

We briefly introduce the concept of Delaunay triangulation which is used as structure for the DT overlay network in SeDAX [39]. Parts of this description are adapted from [39,90,91]. In general, Delaunay triangulation is a commonly used method for generating triangular networks from a point set [90]. Figure 3.1(a) illustrates the Delaunay triangulation of a discrete point set in general position. As shown, it is a triangulation such that no point is inside the circumcircle of any triangle in the Delaunay triangulation [91]. There are several algorithms for DT construction which are nicely summarized in [92]. The Delaunay triangulation



(a) Delaunay triangulation of a point set.

(b) Voronoi diagram for the same point set.

Figure 3.1: *Supporting figure for the explanation of DT. Taken from [90].*

of a discrete point set in general position corresponds to the dual graph of the Voronoi diagram as shown in Figure 3.1(b). The area surrounding each point is called its *Voronoi cell* and contains all coordinates that are closest to that point.

3.1.2 The SeDAX Architecture

SeDAX is a SG communication middleware initially developed by the former C-DAX project partner Alcatel-Lucent Bell Labs[1] that uses geographic routing over a DT overlay network for information dissemination. Its pub/sub communication paradigm decouples information contributors from information consumers by organizing information into *topics*. Formally, SeDAX stores a set of topics $\mathcal{T}$ on a set $\mathcal{V}$ of brokers which are called SeDAX nodes. An overlay network steers messages addressed to a certain topic to the right SeDAX node. Thus, publishers and subscribers do not need to know the IP addresses of the corresponding SeDAX node to send registration and data messages, they just need to have access

[1] Now part of Nokia.

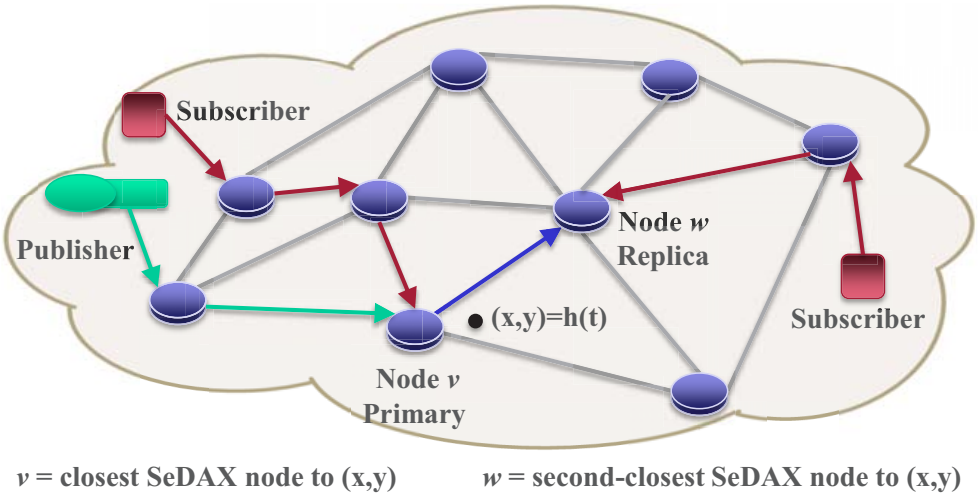


Figure 3.2: *Topic-group communication in SeDAX uses geographic forwarding.*

to the overlay network. As a result, SeDAX does not require a mapping system, that may be compromised or fail, to resolve topics to SeDAX nodes.

Figure 3.2 illustrates the overlay network which is organized as follows: SeDAX nodes $v \in \mathcal{V}$ are equipped with geo-coordinates denoted as $C(v)$. Nodes are connected to selected geographic neighbors via TCP connections to form a DT overlay network. The DT overlay network enables SeDAX nodes to forward a message addressed to a certain coordinate to the closest SeDAX node. All coordinates for which a node v is closest form its Voronoi cell $Voronoi(v)$. A geographic hashing function (GHF) derives a Euclidean coordinate $(x, y) = h(t)$ from the name of a topic t. A topic is stored on the node closest to that coordinate, i.e., on the node with the least Euclidean distance $d(C(v), h(t)), v \in \mathcal{V}$. The GHF and the DT overlay enable other SeDAX nodes to forward messages destined to a topic to the node responsible for that topic. The SeDAX authors [39] have shown that this kind of overlay forwarding creates only little path stretch compared to the shortest path in the overlay. Furthermore, the DT overlay is self-healing: if a node fails, the DT property is restored after some local and self-organized reconfiguration.

SeDAX can be made resilient against node failures. Data and metadata of a topic t are stored on SeDAX nodes that are closest (primary) and second-closest (secondary) to the topic's coordinate $h(t)$. This is simple, as they are neighboring nodes. Node failures are detected via broken TCP connections, and trigger self-healing of the DT overlay [39]. Failed nodes are excluded from the network of forwarding nodes. Messages for topic t are then automatically forwarded to the respective alternative node, which starts delivering messages to subscribers. This resilience concept may be extended to protect against consecutive failures by ensuring that topic data and metadata are always kept on the closest and second-closest working SeDAX node. Thus, the self-healing property of the DT overlay combined with the backup concept constitutes a simple and effective resilience concept in SeDAX that can survive even multiple consecutive failures.

3.1.3 Related Work

SeDAX [39] builds upon prior work in the area of pub/sub [40], data-centric storage [93] and ICN [94]. It specifically addresses the requirements of the SG. A security framework [52] covers security considerations for SeDAX as a cyber-physical system. In recent work [7], we investigated the storage requirements of SeDAX necessary to survive the failure of multiple SeDAX nodes without storage shortages. This led to high storage requirements on SeDAX nodes that could be reduced by assignment of optimized coordinates to SeDAX nodes, which is generally difficult to implement.

SeDAX uses the DT overlay and GHF to locate its pub/sub-based message brokers. Most existing ICN architectures such as PSIRP/PURSUIT [95], 4WARD/SAIL [96], NDN/CCNx [97,98], DONA [99], and CAN [100] are based on distributed hash tables (DHTs) and pub/sub. They differ in the way topic names are resolved, data is forwarded, and whether the organization of data distribution is hierarchical [101] or flat as in SeDAX. QoS constraints for replication in more complex topologies with hierarchical data stores are discussed in [102,103]. LIPSIN [104] uses bloom filters to quickly resolve names and find topic stores.

Chord [105] allocates coordinates on a ring to predecessor and successor nodes. Other architectures like CAN [100] allocate rectangular areas to a primary node, further subdividing or combining rectangles as nodes join or exit the network. Greedy routing schemes like SeDAX organize the space into Voronoi cells so that the closest node to a coordinate is the home node for that coordinate, thus avoiding the need to maintain routing tables.

ICN systems can be viewed as structured P2P systems [106]. In a structured system like SeDAX, some nodes may provide more centralized services such as directory services (e.g. maintaining a lookup table of underloaded nodes) or security services (authoritatively authenticating a node, publisher, or subscriber). Most load balancing approaches in P2P systems focus on unstructured P2P systems [106] where nodes with different capacities frequently join and leave the network. SeDAX nodes are both more structured and less ephemeral whereas SeDAX publishers and subscribers can readily be mobile without requiring updates to the node routing overlay.

Load balancing schemes differ as well. Felber et al. [107] give an excellent overview on current load balancing mechanisms for peer-to-peer systems based on DHTs. Their taxonomy divides load balancing into three different categories: object placement, routing, and underlay. Our approach falls into the object placement category, i.e., load balancing is achieved by placing objects (topics in SeDAX) or nodes on the overlay so that the load among nodes is equalized. We briefly summarize solutions that have been surveyed in [107] and that come close to the proposed SeDAX load balancing approach.

Kenthapadi et al. [108] propose a mechanism for load balanced overlay node addition by placing new overlay nodes between most loaded nodes. Stoica et al. [105] propose virtual servers for load balancing. In a nutshell, a physical node may host several virtual servers that are each responsible for a certain identifier (coordinate in SeDAX). Physical nodes can exchange virtual servers to achieve better load balance, i.e., virtual servers facilitate fair (virtual) overlay node placement. This scheme may be very difficult to implement for SeDAX because of SeDAX' resilience scheme, i.e., primary and backup virtual server have to be

adjacent nodes on the overlay but should be hosted on two different physical nodes. Rao et al. [109] propose three methods for physical nodes to exchange load information about their virtual servers; their methods are comparable to our distributed coordinate selection algorithms in Section 3.4.3. Godfrey et al. [110] extend the methods of [109] by periodic load balancing and emergency load balancing. Byers et al. [111] propose the use of multiple hashing functions to find storage nodes on the overlay, and the use of redirection pointers at destination nodes resembles our topic delegation mechanism in Section 3.4.1.

Most load balancing approaches, including those described in this paper, benefit from the "power of 2 choices" described by Mitzenmacher [112, 113] in ball-bin load balancing. As Bridgewater et al. summarize in Balanced Overlay Networks (BON) [114], "The important result from ball-bin systems is that if one probes the population of more than one bin prior to assigning a ball, the population of the most full bin will be reduced exponentially in N." Even and Medina [115] further discuss lower bounds for ball-bin load balancing.

In BON, nodes change the number of immediate incoming neighbors in response to the node's availability. Thus, the overlay network can be viewed as a directed graph that is dynamically reconfigured to reflect the current system load. BON uses random walks through the directed graph to select the least loaded node on the path. BON's target application is job allocation in grid computing. In this environment jobs enter and leave the network frequently whereas SeDAX's storage requirements tend to be of longer if not permanent duration. However, an implementation of the SeDAX random query approach might use such a random walk to include the least loaded (best) node on the path to the queried location, effectively increasing the scope of queries.

3.2 Definitions and Nomenclature

We introduce auxiliary functions that facilitate later definitions used to evaluate the performance of SeDAX.

- $\mathcal{V}$: set of SeDAX nodes.
- $\mathcal{T}$: set of topics.
- $C(v), v \in \mathcal{V}$: coordinate of node v.
- $C(t), t \in \mathcal{T}$: (delegate) coordinate of topic t.
- $N_j(c)$: node whose coordinate is j-closest to coordinate c among all other SeDAX nodes, e.g., $N_1(c)$ is the closest node, $N_2(c)$ is the second-closest, etc.
- $\mathcal{T}_j(v) = \{t : t \in \mathcal{T}, N_j(C(t)) = v\}$; set of topics for which v is the j-closest node.
- $L_T(t), t \in \mathcal{T}$: load of topic t.

Each topic $t \in \mathcal{T}$ induces a certain load $L_T(t)$ on the node on which it is stored. Since topic data may expire, SeDAX nodes require only sufficient capacity to store current, i.e., non-expired, topic data, and are not intended for archival purposes. Therefore, limited storage is sufficient for the data of a topic $t \in \mathcal{T}$ which is given by the topic load $L_T(t)$. As an alternative to storage capacity, load may be measured in terms of required processing power or I/O capacity if these quantities are the limiting system resource. To facilitate further considerations and calculations, we assume the topic load to be an additive metric.

3.3 Impact of Optimized Node Placement on Storage Requirements

In this section, we investigate the impact of optimized node placement on the storage requirements of SeDAX nodes. We first introduce the performance metrics of interest. Then, we conduct a simulative storage analysis of the SeDAX architecture and suggest a Monte-Carlo based optimization for SeDAX node placement to minimize storage requirements. Finally, we present theoretical lower bounds for SeDAX's storage requirements and for an idealized storage system, and compare them with the simulative results.

3.3.1 Performance Metrics

We denote $\mathcal{S}$ as the set of all considered failure scenarios. A failure scenario $s \in \mathcal{S}$ represents a set of failed nodes including the failure-free case. Given a maximum number n_{fail}^{max} of failed nodes, $\mathcal{S}$ contains all combinations of up to n_{fail}^{max} simultaneously SeDAX node failures. Next, we define performance metrics that can be applied when the coordinates of all topics $h(t)$, $t \in \mathcal{T}$ and nodes $v \in \mathcal{V}$ as well as each topic's storage requirements $L_T(t)$, $t \in \mathcal{T}$ are known.

3.3.1.1 Node Load

The *node load* $L_N(v, s)$ is the sum of the storage requirements of topics for which node v is the primary or secondary node under failure scenario s:

$$\begin{aligned} \mathcal{T}_1(v,s) &= \{t \in T : \forall w \in V \setminus \{v,s\}, d(v,t) < d(w,t)\} \\ \mathcal{T}_2(v,s) &= \{t \in T : \exists^{=1} u \in V \setminus \{v,s\} \forall w \in V \setminus \{v,u,s\}, \\ &\quad (d(u,t) < d(v,t)) \wedge (d(v,t) < d(w,t))\} \\ L_N(v,s) &= \sum_{t \in (\mathcal{T}_1(v,s) \cup \mathcal{T}_2(v,s))} L_T(t). \end{aligned} \tag{3.1}$$

3.3.1.2 Capacity Requirements

Based on the node load, we define capacity requirements for SeDAX nodes and the system. Capacity is given in storage units. To generalize results, we express them relative to the system load c_{load}, i.e., the sum of the storage requirement for all topics $c_{load} = \sum_{t \in T} L_T(t)$. As an example, the required system capacity is 200% relative to the system load when each topic is stored on exactly two nodes in the failure-free scenario, independent of node and topic coordinates.

Capacity Requirement per Node The *node capacity requirement* $c_{node}(v)$ specifies the minimum capacity required to store the node load in all failure scenarios $s \in \mathcal{S}$:

$$c_{node}(v) = max_{s \in \mathcal{S}} \left(L_N(v,s) \right). \tag{3.2}$$

Maximum Capacity Requirement per Node The *maximum node capacity requirement* c_{node}^{max} is defined as the largest capacity requirement $c_{node}(v)$ of any node $v \in V$ in all failure scenarios $s \in \mathcal{S}$:

$$c_{node}^{max} = max_{v \in V} \left(c_{node}(v) \right). \tag{3.3}$$

It specifies the minimum storage requirements of nodes if all nodes are provisioned uniformly or *homogeneously*, i.e., all nodes have equal storage.

System Capacity The *system capacity* c_{sys} specifies the SeDAX network-wide storage required to survive all failures $s \in \mathcal{S}$ without storage shortages if each node is individually or *heterogeneously* provisioned with its minimum needed storage:

$$c_{sys} = \sum_{v \in V} c_{node}(v). \tag{3.4}$$

3.3.2 Simulative Storage Analysis

This section presents a simulative analysis of storage requirements for SeDAX nodes. First, the experiment setup for the evaluation is given together with an optimization scheme for SeDAX node placement. The simulation results show storage requirements for SeDAX with optimized SeDAX node placement.

3.3.2.1 Experiment Setup and Optimization of Node Placement

To evaluate storage requirements for SeDAX under failures, we choose a square plane as coordinate space and create topic and node patterns. We generate random coordinates for $n_{topics} = 100$ topics and for $n_{nodes} = \{5, 10, 20\}$ nodes. Then we calculate the performance metrics for up to $n_{fail}^{max} = 3$ simultaneous node failures.

We perform Monte Carlo optimization of node placement to reduce storage requirements. We produce $n_{patterns}^{node} = 200$ different node patterns and choose the one that requires least system capacity. The results of these experiments depend on the topic pattern. Therefore, we repeat them for $n_{patterns}^{topic} = 40$ different topic

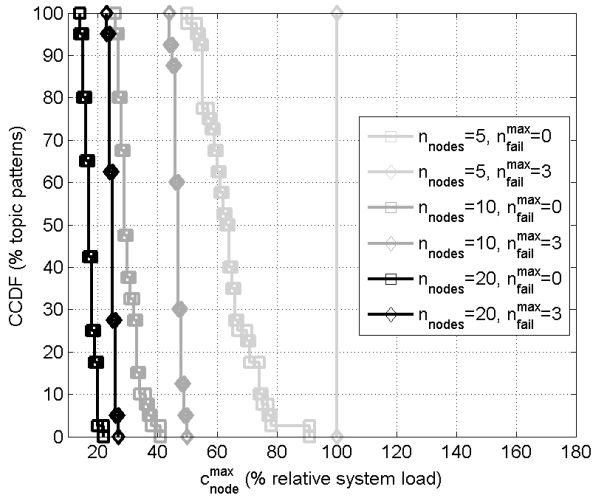


Figure 3.3: *Impact of the number of nodes n_{nodes} and number of failed nodes n_{fail}^{max} on the maximum node capacity c_{node}^{max} under optimized node placement.*

patterns and express the results as complementary cumulative distribution functions (CCDFs) based on the topic patterns. To simplify the analysis, we set $L_T(t)$ to one storage unit, but this is not a constraint for the presented optimization and evaluation framework.

3.3.2.2 Storage for Optimized Node Placement

Maximum Node Capacity Requirements We first assume that all nodes in SeDAX are provisioned with the same amount of storage. In order to survive node failures without storage shortage, all nodes need at least the maximum node capacity requirements c_{node}^{max} as defined in Equation (3.3). Therefore, we optimize the node placement to minimize c_{node}^{max}.

Figure 3.3 shows the CCDF of the results of the maximum node capacity requirements for the optimized node placement. We interpret the figure as follows: for each maximum node capacity requirement x on the x-axis, the y-axis gives the percentage of topic patterns whose maximum node capacity requirements X are greater than x.

We observe that the maximum node capacity requirements decrease with an increasing number of SeDAX nodes in the system. This is a trivial result: as the number of topics and their data volume is the same in all experiments, the average load per node is inversely proportional to the number of nodes n_{nodes}, at least for $n_{fail}^{max} = 0$.

We recognize that the maximum node capacity depends on the specific topic pattern. For $n_{nodes} = 20$ nodes, the maximum node capacities range in the failure-free case between 14% and 22%. If up to three nodes fail, the maximum node capacities range between 23% and 27% relative system load. More storage capacity is needed for the SeDAX system to survive additional node failures without storage shortages. When all nodes in a SeDAX system are provisioned homogeneously, the system-wide capacity requirement is $n_{nodes} \cdot c_{node}^{max}$. For $n_{nodes} = 20$ nodes, a system-wide capacity between 460% and 540% is required.

System Capacity Node-specific storage provisioning is an alternative to homogeneous provisioning. That means, each node is provisioned with its individual node capacity $c_{node}(v)$ to survive up to a given number of node failures n_{fail}^{max} without storage shortages. We now optimize the placement of SeDAX nodes to minimize c_{sys}.

Figure 3.4 shows the CCDF of the system capacities for the optimized node placement whose mean values are summarized in Table 3.1. It illustrates that the SeDAX system requires significantly more storage to survive up to n_{fail}^{max} node failures compared to the failure-free case. We observe that the system capacity depends on the topics patterns. This is because the backup capacity can be shared more efficiently for some topic patterns than for others. For 20 nodes, the required system capacity is between 254% and 263% for $n_{fail}^{max} = 1$, between 311% and 327% for $n_{fail}^{max} = 2$, and between 368% and 383% for $n_{fail}^{max} = 3$. The figure further shows that the required system capacity is about the same for $n_{nodes} = 10$ and $n_{nodes} = 20$ nodes and only for a very small number of nodes like $n_{nodes} = 5$, the relative system capacity is clearly larger.

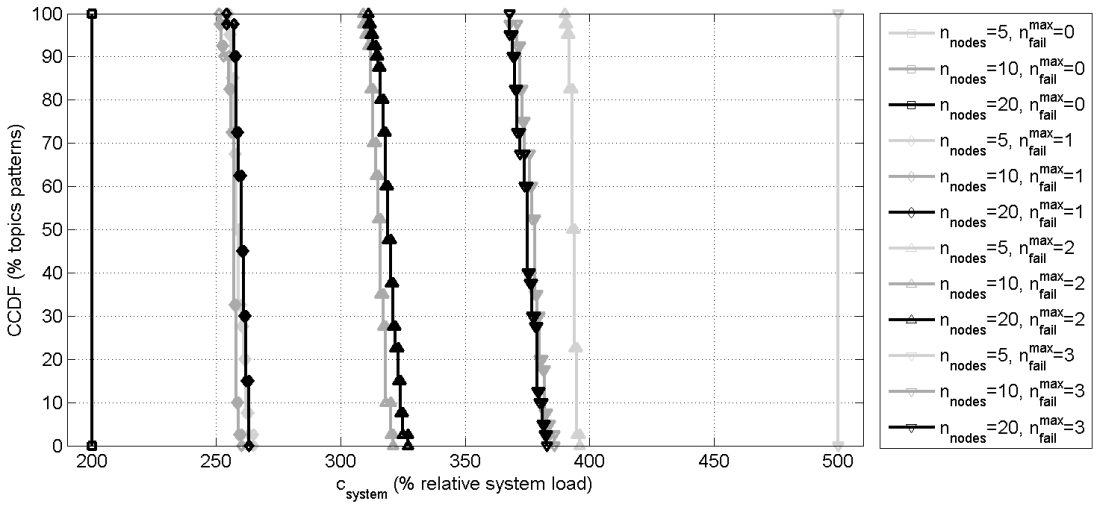


Figure 3.4: *Impact of the number of nodes n_{nodes} and number of failing nodes n_{fail}^{max} on the system capacity c_{sys} under optimized node placement.*

Node-specific storage provisioning leads to significant storage savings compared to homogeneous node storage provisioning. For 20 nodes and a maximum number of $n_{fail}^{max} = 3$ failed nodes, savings up to

$$\frac{n_{nodes} \cdot c_{node}^{max} - c_{sys}}{n_{nodes} \cdot c_{node}^{max}} = \frac{540\% - 368\%}{540\%} \approx 32\%$$

are possible. In other words, homogeneous node storage provisioning requires 68% more storage than node-specific storage provisioning to provide the same level of protection.

The outcome of the optimization may seem difficult to implement as node placement in practice is typically determined by operational necessities. However, the assignment of virtual coordinates and their use for the DT overlay combines arbitrary physical placement of nodes with the use of optimized coordinates of SeDAX nodes. The drawback of that approach may be longer paths in the DT overlay.

3.3.3 Analytical Lower Bounds

We derive lower bounds for SeDAX system capacity requirements that could suffice under optimal conditions. Then, we calculate lower bounds for an idealized storage system. Numerical results are compared with those from simulations.

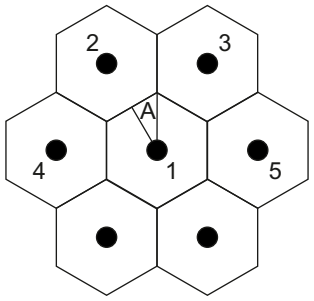
3.3.3.1 Bounds for SeDAX

In a SeDAX system, the overall storage requirements are smallest when the same amount of data is distributed equally across all SeDAX nodes in the failure-free case and each node shares its load equally among the maximum number of equidistant neighbors. This consideration is the basis for the following analysis.

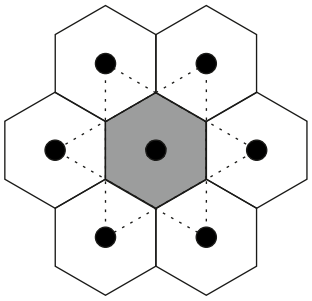
We consider an infinite plane. A GHF maps a vast number of topics with equal storage requirements evenly over this plane. Since a triangular node arrangement maximizes the number of equidistant neighbors, we use it for the placement of SeDAX nodes.

The *topic area* for which a SeDAX node is the closest node thus forms a hexagon with six adjacent neighbors, as shown by the gray area in Figure 3.5(a). For area A these nodes are listed in order of proximity. Normally, a topic that maps to area A is assigned to the closest node (node 1) as primary, and its secondary to the second-closest node (node 2). When one of these nodes (node 1 or 2) fails, the third-closest node (node 3) becomes the secondary node. When two of the three closest nodes fail, the affected topic is stored on the fourth-closest node (node 4). For larger numbers of adjacent node failures, the topic responsibility is shifted in the same way.

Failure-Free Condition For all topics that the GHF hashes into a hexagon, see the gray area in Figure 3.5(b), the primary node is located in the center. We denote the load created by the topics located in a single hexagon as 100% load. Due to the assumption that topics are evenly distributed over the plane, each node carries 100% primary load. There are six triangular areas adjacent to this hexagon. They form the area for which the central node is second-closest (see the areas bounded by the dashed lines in Figure 3.5(b)) and thereby contribute another 100% load to the central node. Thus, each node carries 200% load in the failure-free scenario.

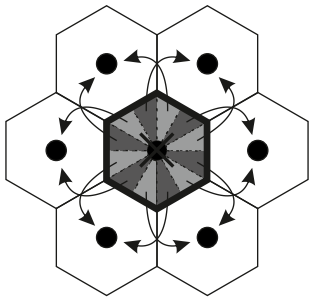


(a) The five closest SeDAX nodes of topic area A.

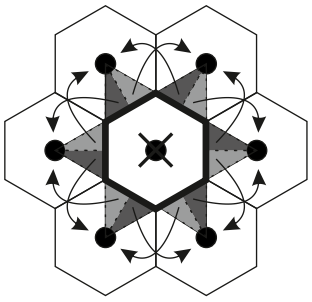


(b) The node in the middle is closest to the gray area and second-closest to the dashed triangles.

Figure 3.5: *Supporting figures for the analysis.*



(a) Topic areas reassigned to previously second-closest nodes.



(b) Topic areas reassigned to previously third-closest nodes.

Figure 3.6: *Reassignment of topic area responsibilities if one SeDAX node fails.*

Single Node Failures In Figure 3.6(a), the center node serves as primary node for the topics mapped to the shaded triangles. When it fails, the secondary nodes take over as primary and the topics are reassigned to new secondary nodes as indicated by the arrows. The load of one triangle corresponds to a load of $\frac{1}{12} \cdot 100\%$.

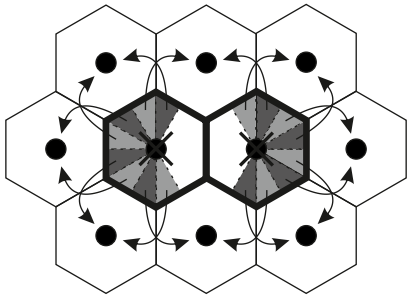
In Figure 3.6(b), the center node serves as secondary node for the topics mapped to the shaded triangles. When it fails, new secondary nodes are reassigned to the topics that are mapped to the respective triangles. These nodes are indicated by the arrows.

In both figures together, we count four arrows towards the failed node's neighbors. Thus, each of those nodes receives an additional load of $\frac{4}{12} \cdot 100\% \approx 33.3\%$ so that it must carry an overall load of 233.3%.

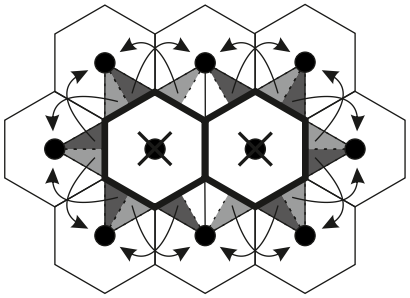
Double Node Failures We analyze the cases in which the failed nodes are adjacent to each other, separated by exactly one node, or separated by more than one node.

Two adjacent nodes fail. We use the same approach as for the single node failure in Section 3.3.3.1 to analyze the failure of two adjacent nodes. Figure 3.7(a) shows the areas that lose their closest node, but not their second- and third-closest nodes. Thus, a copy of the topics mapped to these areas is added to the third-closest nodes as indicated by the arrows in the figure. Figure 3.7(b) shows the areas that lose their second-closest nodes, but not their closest and third-closest nodes. Thus, a copy of the topics mapped to these areas is also added to the third-closest nodes as indicated in the figure. Figure 3.7(c) shows the areas that lose their closest and third-closest node or their second- and third-closest node, but not their fourth-closest node. Thus, a copy of the topics mapped to these areas is added to the fourth-closest nodes as indicated in the figure. Figure 3.7(d) shows the areas that lose their closest and second-closest node, but not their third- and fourth-closest nodes. Thus, a copy of the topics mapped to these areas is added to both the third- and fourth-closest nodes.

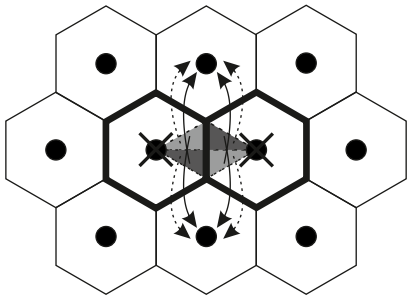
Adding up all reassignments of topic area responsibilities, we see that neighboring nodes of the failed nodes receive additional load from 4, 6, or 8 triangles, which results in a maximum additional load of $\frac{8}{12} \cdot 100\% \approx 66.7\%$. The most heavily loaded nodes are the direct neighbors of the two failed nodes; they must be able to carry a load of up to 266.7%.



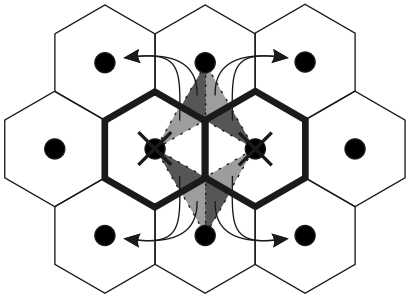
(a) When only the closest node to an area fails, its load is addded to the third-closest nodes.



(b) When only the second-closest node to an area fails, its load is added to the third-closest nodes.



(c) When the first- and second-closest nodes to an area fail, its load is added to the third- and fourth-closest nodes.



(d) When the first- and third-closest nodes or the second- and third-closest nodes to an area fail, its load is added to the fourth-closest node.

Figure 3.7: *Reassignment of topic area responsibilities if two adjacent SeDAX nodes fail.*

Two nodes fail with one node in between. If two non-adjacent nodes fail that are separated only by a single intermediate node, this node receives 33.3% additional load from each of its failed neighbors. This is the worst case which amounts to a maximum load of 266.7%.

Two nodes fail with more than one node in between. If two non-adjacent nodes fail that are separated by more than a single intermediate node, their neighboring nodes receive additional load only from one of the failed nodes. Therefore, the maximum load is 33.3% like in the single node failure scenario.

Triple Node Failures For the sake of brevity, we consider only the worst case in terms of additional load. When three contiguous neighbors of a node fail, the node next to the three failed nodes needs to carry at most a load of 316.7%.

3.3.3.2 Bounds for an Idealized System

In an idealized storage system, each topic's data is simultaneously stored on both a primary and a secondary node. When one of these nodes fails, its topic data is instantaneously replicated to yet another node so that two copies of the same topic are always available in the system. When a node or topic is added or removed, its topic data is distributed evenly over all nodes. This idealized load distribution leads to theoretical minimum storage requirements.

Due to the idealized load distribution, each of the n_{nodes} storage nodes carries $\frac{200\%}{n_{nodes}}$ of the system load. When n_{fail}^{max} nodes fail, each of the remaining $n_{nodes} - n_{fail}^{max}$ nodes now carries $\frac{200\%}{n_{nodes}-n_{fail}^{max}}$ of the system load so that the network-wide system capacity requirement of the idealized storage system c_{sys}^{ideal} is defined as

$$c_{sys}^{ideal} = \frac{n_{nodes}}{n_{nodes} - n_{fail}^{max}} \cdot 200\%. \tag{3.5}$$

3.3.4 Insights

We compare the system capacity requirements of the idealized storage system with simulation results and the lower bounds for SeDAX in Table 3.1. While the idealized storage system uses only 36% extra system capacity to provide enough capacity to accommodate backup copies if up to $n_{fail}^{max} = 3$ nodes fail, SeDAX requires $168\% - 186\%$ extra capacity. In contrast to SeDAX, the idealized storage system leverages perfect load balancing, so its capacity requirements are independent of topic coordinates and node placement.

Table 3.1: *System capacity requirements for up to n_{fail}^{max} node failures: simulation results and analytical lower bounds for SeDAX together with lower bounds of an idealized storage system.*

n_{nodes}	n_{fail}	**SeDAX simulation**	**Idealized system**	**Lower bounds for SeDAX**
5	0	200%	200%	200%
	1	255% – 265%	250%	233.3%
	2	390% – 396%	333%	266.7%
	3	500%	500%	316.7%
10	0	200%	200%	200%
	1	251% – 260%	222%	233.3%
	2	309% – 321%	250%	266.7%
	3	368% – 386%	285%	316.7%
20	0	200%	200%	200%
	1	254% – 263%	211%	233.3%
	2	311% – 327%	222%	266.7%
	3	368% – 383%	236%	316.7%

The analytical lower bounds for SeDAX are derived for an infinite plane with an infinite number of nodes. We compare them with the system capacity requirements of the idealized storage system. For $n_{fail}^{max} = 3$, we have 116.7% extra capacity compared to 36% extra capacity for $n_{nodes} = 20$ nodes. Even though the lower bounds for SeDAX were calculated for optimal conditions, it is still considerably less efficient than the idealized storage system. We see this deviation because SeDAX cannot efficiently distribute capacity. When a node fails, only its closest neighbors copy its data and provide backups; available capacity on distant nodes cannot be used for that purpose.

3.4 Improving Load Distribution in SeDAX

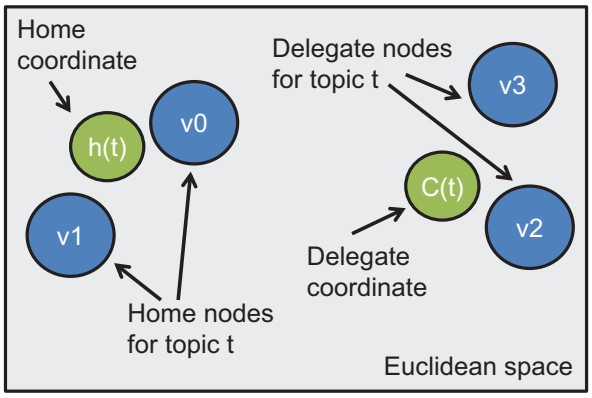
In the previous section, we showed the storage requirements of SeDAX necessary to survive the failure of multiple SeDAX nodes without storage shortages. The high storage requirements on SeDAX nodes could be reduced by assignment of optimized coordinates to SeDAX nodes. An alternative to explore would be to keep the coordinates of the nodes and optimize the placement of topics, which would require larger architectural modifications to SeDAX.

In the following, we propose a topic delegation mechanism to make the assignment of topics to nodes dynamic. For this elaborated SeDAX approach, we suggest a distributed resource management system that detects traffic imbalances among SeDAX nodes and re-assigns topics to other coordinates for load balancing purposes. The proposed mechanism is the only existing method to improve the flexibility and resource management of the SeDAX architecture so far.

We first describe the topic delegation mechanism and give definitions for the loads of SeDAX nodes and coordinates for different levels of resilience. The latter also includes a definition for best coordinates on the overlay which is an essential part of the distributed coordinate selection algorithms that are described thereafter. Finally, we detail several distributed load balancing algorithms. We investigate their impact in Section 3.5.

3.4.1 Topic Delegation Mechanism

If a SeDAX node is overloaded, diverting load to other nodes may be helpful. However, due to static assignment of topic coordinates $C(t)$ to coordinates $h(t)$, the original SeDAX architecture cannot support load shifting by design. We propose topic delegation for SeDAX which uses $h(t)$ as the default coordinate of a topic, but allows for a reassignment of $C(t)$ to any other coordinate. Topic delegation adds flexibility to SeDAX without sacrificing its benefits, e.g., resilient overlay forwarding, decentralized control, and the ability to cope without a mapping system. In the following, we explain the principle and operation of topic delegation in SeDAX.



Delegate list of nodes v0, v1, v2 and v3

Topic	Coordinate	Timestamp	…
t	h(t)	0	…
t	C(t)	2	…
…	…	…	…

Figure 3.8: *Topic delegation principle in SeDAX. The home nodes ($v0$ and $v1$) are responsible for all messages of topic t from time zero until the next entry on the delegate list, time 2. All messages on or following time 2 are the responsibility of the delegate nodes ($v2$ and $v3$). The delegate list is synchronized among home and delegate nodes.*

3.4.1.1 Topic Delegation Principle

The node closest to a topic's default coordinate $h(t)$ is the topic's *home node*. By default, the topic coordinate $C(t)$ equals the topic's hash value $h(t)$ and is called *home coordinate* of topic t. When the topic coordinate $C(t)$ is set to a value other than $h(t)$, the coordinate $C(t)$ is called *delegate coordinate* of topic t and the node closest to that coordinate is called the *delegate node* for topic t. Delegate nodes are responsible for the topic, i.e., they store published topic data and metadata.

Home nodes track where all topic data is stored via a *delegate list*. A delegate list holds the active topic coordinates $C(t)$ for topics $t \in \mathcal{T}$ and a timestamp of the first message stored at the respective coordinate; this list is shared and synchronized among home and delegate nodes.

Figure 3.8 shows a delegate list shared among topic t's home and delegate nodes. The current coordinate of a topic is the most recent coordinate on the delegate list. The home nodes ($v0$ and $v1$) are responsible for all messages of topic t from time zero until the next entry on the list, time 2. All messages on or following time 2 are the responsibility of the delegate nodes ($v2$ and $v3$). If the delegate nodes at coordinate $C(t)$ have retired themselves from service for topic t, the home nodes resume responsibility by default and enter the home coordinate $h(t)$ as most recent topic coordinate on the list.

Once home and delegate nodes agree to participate in a forwarding relationship, the home nodes add to their list of delegates an entry containing the delegate coordinate $C(t)$ and the *start time*. The start time is the timestamp of the first data packet stored at $C(t)$. If the start time is not known, e.g., no data has been stored at $C(t)$ yet, the start time is the time at which forwarding to $C(t)$ began. When a topic moves from one delegate node to another, registrations are transferred to the new delegate node. It is up to the implementer whether the old or new delegate node informs the clients about that event. The first entry in the delegate list always contains the topic coordinates $h(t)$ and start time zero to ensure that the home nodes remain responsible for all requests prior to any other delegation.

Delegate nodes that wish to retire, e.g., because they have become overloaded themselves, simply inform the home nodes that they are retiring. Retiring delegate nodes should exit gracefully to minimize both the potential for data loss and unnecessary network traffic, normally by waiting until all of their data has expired, alternatively by gradually shifting their load to their predecessor and/or successor. When retiring delegate nodes no longer contain data for a topic, they notify the home nodes and the delegation is deleted from the delegate list.

This simple yet robust arrangement elegantly enables the continuous service of requests while avoiding the excessive data transfers, forwarding loops, data loss, and service delays that commonly accompany such transitions.

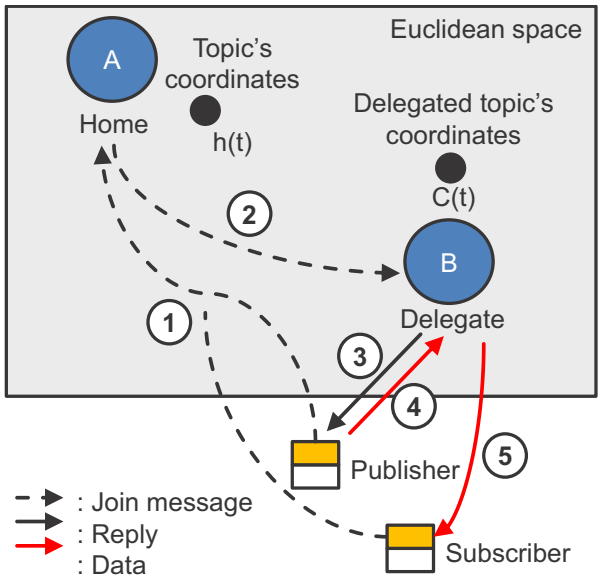


Figure 3.9: *Topic delegation operation in SeDAX. Clients send registrations to $h(t)$ (Step 1). Responsible home node forwards them to delegate node at $C(t)$ (Step 2). Delegate node returns delegate coordinate to publisher upon registration (Step 3). Publisher sends traffic to delegate coordinate $C(t)$ instead of $h(t)$ (Step 4). Subscriber receives traffic from delegate node instead of home node (Step 5).*

3.4.1.2 Topic Delegation Operation

When a SeDAX client (publisher or subscriber) joins a topic, the client first sends a join message over the overlay to the topic's home coordinate $h(t)$ so that the message reaches the home node, as illustrated in Figure 3.9 (Step 1). The topic's home node checks its delegate list for that topic. If there is no entry, the home node itself is the message broker for that topic; no modification to the existing SeDAX architecture is needed. If the delegate list holds an entry for that topic, the home node forwards the join message to the delegate coordinate $C(t)$ (Step 2); this can be achieved by encapsulation to the delegate coordinate or rewrite of the destination coordinate. Upon receipt of the join message, the delegate node registers the client for the requested topic and informs the client to use the new topic

coordinate $C(t)$ instead of $h(t)$ in all subsequent messages (Step 3). In particular, publishers will address all data messages to $C(t)$ instead of $h(t)$ (Step 4); subscribers will receive all data messages from the delegate node instead of the home node (Step 5). Should the topic be moved for some reason to another node, all registered clients are informed of the new delegate coordinate.

3.4.1.3 Robustness Considerations

Each topic data store, whether at the default topic coordinates $h(t)$ or the delegate coordinates $C(t)$, has a secondary node ($v1$ and $v3$ in Figure 3.8) to which it replicates the topic data and control structures. Should a home or delegate node fail, the secondary node seamlessly takes over since it is now the node nearest to the coordinates in question.

Since a secondary node has already been pre-populated with appropriate topic data and metadata, it can now start replicating to a new secondary node, and so forth. Note that a returning primary node must check with its secondary node before resuming operations. The secondary node then becomes a delegate node for topic data that was stored during the primary node's absence. It is not necessary to copy the interim data back to the primary node unless other factors, such as load balancing, make the shift of data desirable. The adjacent delegate shifting mechanism can then facilitate an orderly, efficient, and gradual shift of topic data under home's direction, even after cascading node failures.

3.4.2 Load Definitions for SeDAX Nodes and Coordinates

We consider three different levels of resilience for the operation of SeDAX. We define load metrics for SeDAX nodes and coordinates, based on which we determine a SeDAX node's best coordinate. These concepts are used by the coordinate selection and load balancing algorithms presented in Section 3.4.3 and Section 3.4.4, respectively.

3.4.2.1 Considered Resilience Levels

We consider three different resilience levels for SeDAX operation.

1. No resilience. Topic data and topic information are stored only on SeDAX node $N_1(C(t))$. If the node fails, the topic information is lost.
2. Resilience against one node failure. Topic data and information are stored redundantly on two SeDAX nodes $N_1(C(t))$ and $N_2(C(t))$. If $N_1(C(t))$ fails, messages are automatically rerouted to $N_2(C(t))$ so that they can be forwarded to the registered subscribers. If both $N_1(C(t))$ and $N_2(C(t))$ fail, the topic information is lost and publishers cannot longer reach a broker.
3. Resilience against two node failures. Topic information is stored redundantly on two SeDAX nodes $N_1(C(t))$ and $N_2(C(t))$ like above. If $N_1(C(t))$ fails, messages are automatically rerouted to $N_2(C(t))$ so that they can be forwarded to the registered subscribers. In addition, if $N_1(C(t))$ or $N_2(C(t))$ fails, topic data and information are copied to SeDAX node $N_3(C(t))$. Should the remaining node $N_1(C(t))$ or $N_2(C(t))$ also fail, then $N_3(C(t))$ takes over.

More than two successive node failures are repetitions of the two node failure scenario.

3.4.2.2 Load Definitions

We provide definitions for a topic's *load on a SeDAX node* and the *load on a coordinate* for different resilience levels. While the node loads serve to quantify load imbalance among nodes, the coordinate loads are used to find appropriate coordinates for load balancing.

Node Load $L_N^i(v)$ The node load $L_N^i(v)$ is the maximum load on a node $v \in \mathcal{V}$ induced by topics in any failure scenario considered by resilience level i. It is the minimum capacity for v to guarantee operation on resilience level i without capacity shortage.

Resilience Level 1 A SeDAX node v is responsible only for topics $t \in \mathcal{T}$ for which it is the closest node. The maximum load induced by topics on this node is

$$L_N^1(v) = \sum_{t \in \mathcal{T}_1(v)} L_T(t). \tag{3.6}$$

Resilience Level 2 A SeDAX node v is responsible for topics for which it is the closest or second-closest node. The maximum load induced by topics on this node is

$$L_N^2(v) = \sum_{t \in (\mathcal{T}_1(v) \cup \mathcal{T}_2(v))} L_T(t). \tag{3.7}$$

Resilience Level 3 As above, a SeDAX node v is responsible for topics for which it is the closest or second-closest node; the resulting base load is $L_N^2(v)$. With resilience level 3, node v becomes responsible for additional topics for which it is third-closest if their closest or second-closest node fails. The imposed load depends on the failure of a specific primary or secondary node $x \in \mathcal{V}$. Therefore, we determine the maximum load over all relevant single node failures. The failure of a specific node $x \in \mathcal{V}$ is relevant only if it is closest or second-closest for a topic $u \in \mathcal{T}$, i.e., $N_1(C(u)) = x$ or $N_2(C(u)) = x$, for which the considered node v is third-closest, i.e., $\{u \in \mathcal{T}_3(v)\}$. Thus, the maximum additional load imposed on node v in case of a node failure is:

$$L_{aN}^3(v) = \max_{w \in \left\{ \substack{x : x \in \mathcal{V}, u \in \mathcal{T}_3(v), \\ N_1(C(u)) = x \vee \\ N_2(C(u)) = x} \right\}} \sum_{t \in \left\{ \substack{s : s \in \mathcal{T}_3(v), \\ N_1(C(s)) = w \vee \\ N_2(C(s)) = w} \right\}} L_T(t) \tag{3.8}$$

and the node load for resilience level 3 is

$$L_N^3(v) = L_N^2(v) + L_{aN}^3(v). \tag{3.9}$$

Minimum and Maximum Coordinate Load ($L_{min}^i(c)$ and $L_{max}^i(c)$) We define the minimum (maximum) load of a coordinate c as the minimum (maximum) of all node loads that are affected by topics assigned to coordinate c.

Resilience Level 1 A topic assigned to coordinate c is stored only on the closest node $N_1(c)$ so that $N_1(c)$ stores only information of topics for which it is closest node. The (minimum and maximum) coordinate load is

$$L^1_{min}(c) = L^1_{max}(c) = L^1_N(N_1(c)). \tag{3.10}$$

Resilience Level 2 A topic assigned to coordinate c is stored on the closest node $N_1(c)$ and on the second-closest node $N_2(c)$. These nodes store the information of topics for which they are closest or second-closest. The coordinate loads are

$$L^2_{min}(c) = \min\left(L^2_N(N_1(c)), L^2_N(N_2(c))\right) \text{ and} \tag{3.11}$$

$$L^2_{max}(c) = \max\left(L^2_N(N_1(c)), L^2_N(N_2(c))\right). \tag{3.12}$$

Resilience Level 3 Like above, a topic assigned to coordinate c is stored on the closest node $N_1(c)$ and on the second-closest node $N_2(c)$. The maximum load of those nodes is $L^3_N(N_1(c))$ and $L^3_N(N_2(c))$. Moreover, the topic may be stored on the third-closest node $N_3(c)$ if $N_1(c)$ or $N_2(c)$ fails. That node $N_3(c)$ carries the load $L^2_N(N_3(c))$ from topics for which it is closest or second-closest node. If $N_1(c)$ or $N_2(c)$ fails, node $N_3(c)$ carries in addition the load from all topics that have $N_1(c)$ or $N_2(c)$ as closest or second-closest node, and $N_3(c)$ as third-closest node. Thus, the failure-set-specific additional node load $L^3_{faN}(N_3(c), c)$ of $N_3(c)$ for coordinate c is

$$L^3_{faN}(N_3(c), c) = \max_{w \in \{N_1(c), N_2(c)\}} \sum_{t \in \left\{ \begin{smallmatrix} s : s \in \mathcal{T}_3(N_3(c)), \\ N_1(C(s)) = w \vee \\ N2(C(s)) = w \end{smallmatrix} \right\}} L_T(t). \tag{3.13}$$

Hence, the coordinate loads are

$$\begin{aligned} L^3_{min}(c) = {} & \min(L^3_N(N_1(c)), L^3_N(N_2(c)), \\ & L^2_N(N_3(c)) + L^3_{faN}(N_3(c), c)) \text{ and} \end{aligned} \tag{3.14}$$

$$\begin{aligned} L^3_{max}(c) = {} & \max(L^3_N(N_1(c)), L^3_N(N_2(c)), \\ & L^2_N(N_3(c)) + L^3_{faN}(N_3(c), c)). \end{aligned} \tag{3.15}$$

3.4.2.3 Definition of Best Coordinates

We defined and experimented with several heuristics for best coordinates. In the following, we present the heuristic that proved best during our work. We define $\mathcal{C}^*$ as a set of coordinates. If a new topic should be assigned to a coordinate from that set, the coordinate should be carefully selected such that it minimizes the maximum load of all nodes, maximizes the minimum load of all nodes, and minimizes the required backup capacity. This translates to the following three criteria based on the metrics L^i_{max}, L^i_{min}, and coordinate-specific spare capacity:

1. Select a coordinate with a small maximum coordinate load L^i_{max}.

2. Select a coordinate with a small minimum coordinate load L^i_{min}.

3. Only for resilience level 3: select a coordinate with a large coordinate-specific spare capacity on the coordinate's third-closest node $N_3(c)$. It is the spare capacity on $N_3(c)$ if either the closest node $N_1(c)$ or the second-closest node $N_2(c)$ fails. That capacity is calculated as $L^3_N(N_3(c)) - (L^2_N(N_3(c)) + L^3_{faN}(N_3(c), c))$.

We define that a coordinate c_0 is better than a coordinate c_1 if it is better in the first criterion (small $L^i_{max}(c)$). Or if it is equal in the first criterion but better in the second one (small $L^i_{min}(c)$). Or if it is equal in the first two criteria and better in the third one (coordinate-specific spare capacity). A coordinate of a coordinate set $\mathcal{C}^*$ is best if there is no better coordinate in that set. Several best coordinates may exist. These criteria combine the best heuristics in our experiments.

For resilience level 1, all coordinates of a Voronoi cell $Voronoi(v)$ of a node $v \in \mathcal{V}$ are equally good. This is different for resilience level 2 and 3. Here, a mathematical analysis yields the area of best coordinates. Alternatively, a best (or at least a good) coordinate may be found empirically by selecting the best coordinate of a set of random coordinates within a node's Voronoi cell. This is much simpler, but may not find the absolute best coordinate.

3.4.3 Distributed Coordinate Selection Algorithms

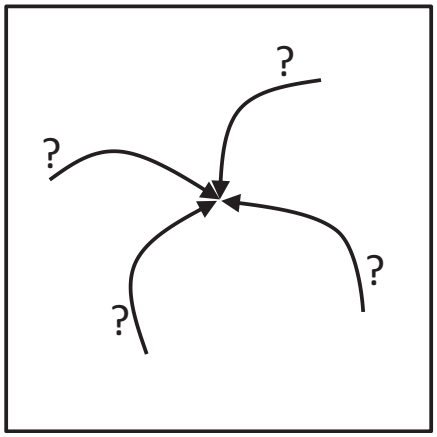
We present four different algorithms for distributed coordinate selection in SeDAX that support resilience levels 1, 2, and 3. If a node v wants to delegate a topic with a coordinate $C(t) \in Voronoi(v)$ within its own Voronoi cell to another coordinate, we call it a delegating node. This delegating node needs to find a better coordinate according to the definitions in Section 3.4.2.3. Load metrics in this section should be computed excluding the topic to be delegated.

3.4.3.1 Querying for Individual Coordinates (IndCoord)

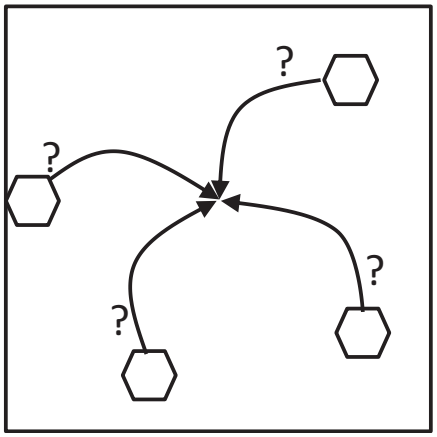
A delegating node may send a query to a random coordinate c that is forwarded to its closest node $N_1(c)$ over the DT overlay. This node locally computes the metrics L^i_{max}, L^i_{min}, and coordinate-specific spare capacity as proposed in Section 3.4.2.3 and returns them to the delegating node. The delegating node may issue $n_{queries}$ such queries so that it eventually knows the relevant loads of $n_{queries}$ other coordinates and the load $L_T(t)$ of the topic to be delegated. On this basis the delegating node can choose the best coordinate and assign the topic. This method is illustrated in Figure 3.10(a).

3.4.3.2 Determining Locally Best Coordinates

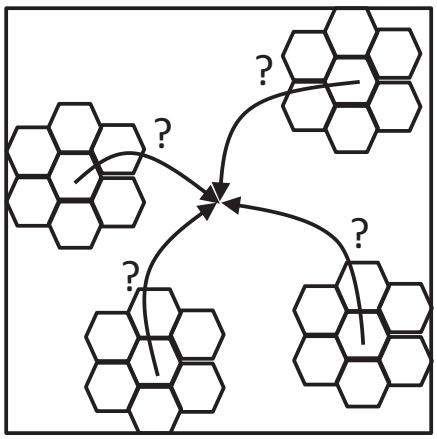
In our simulation implementation, we use a heuristic coordinate generation approach to produce best coordinates for querying nodes. We generate 200 random coordinates per SeDAX node which must lie in the Voronoi cell of the respective node and which serve as best coordinate candidates. We calculate the load metrics for each coordinate according to the desired resilience level and cache them. When a node is queried by another node, it returns its best coordinate according to the preferred selection mechanism. Cache refresh is necessary when topics are added to or removed from the system, and when topics are delegated from one coordinate to another. Appropriate data structures allow for a significant reduction of recalculations for the latter because only affected nodes at the delegation source and destination have to refresh the load metrics of their best coordinate candidates.



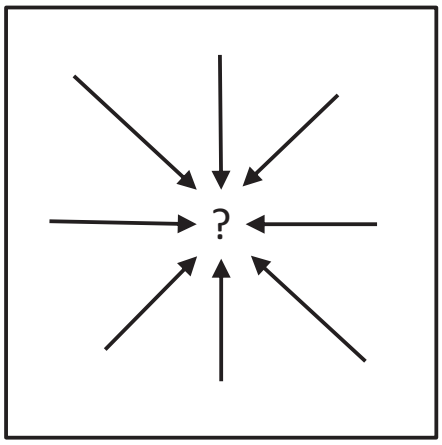
(a) IndCoord: Query load state from individual random coordinates.



(b) BestLocalCoord: Query best coordinate from random cell.



(c) BestRegionalCoord: Query best coordinate from random region.



(d) BestGlobalCoord: Choose globally best coordinate based on flooding.

Figure 3.10: *Algorithms for finding a delegation coordinate.*

3.4.3.3 Querying Locally Best Coordinates (BestLocalCoord)

This differs from IndCoord in that the node $N_1(c)$ determines a locally best coordinate within its Voronoi cell $Voronoi(v)$ according to Section 3.4.2.3. It returns that coordinate including the relevant metrics to the delegating node. Thus, the delegating node receives $n_{queries}$ locally best coordinates and also computes its

own locally best coordinate. The topic is assigned to the best coordinate among them. This method is illustrated in Figure 3.10(b). It causes more computational overhead than IndCoord, but it is likely to find good coordinates more efficiently.

3.4.3.4 Querying Regionally Best Coordinates (BestRegionalCoord)

Here, the node receiving the query returns a regionally best coordinate selected from the coordinates of its own cell and of those cells within n_{hops} hops. Thus, the delegating node receives $n_{queries}$ regionally best coordinates and also computes its own regionally best coordinate. The topic is assigned to the best coordinate among those. This method is illustrated in Figure 3.10(c). It causes more computational overhead and involves more communication than the methods presented above, but it is more likely to find better coordinates.

3.4.3.5 Determining Globally Best Coordinates Based on Flooding (BestGlobalCoord)

The delegating node floods a request to all other nodes (or at least one node in each region) for their best coordinates. The responses allow the delegating node to determine a globally best coordinate to which the topic is assigned. This method is illustrated in Figure 3.10(d). It may require more computation and communication than the methods presented above, but it is able to find a network-wide best delegation coordinate given the current network state.

3.4.4 Distributed Load Balancing Algorithms

We distinguish two types of load balancing for SeDAX: load-balanced topic addition and continuous load balancing. In the following, we show the basic steps for each type and briefly discuss the differences. A combined version of the two approaches is briefly presented at the end of this section.

Algorithm 1 Load-balanced topic addition.

Require: balanced SeDAX network, $t \notin \mathcal{T}$

1: $C(t) \leftarrow h(t)$ {Geographical hash of topic t.}
2: $C^* \leftarrow$ Distributed coordinate selection
3: $c_{cand}^{best} \leftarrow$ Tiebreaker(C^*) {Best coordinate from C^*.}
4: **if** c_{cand}^{best} is better than $C(t)$ **then**
5: $\quad C(t) \leftarrow c_{cand}^{best}$ {Delegate t to $c_{cand}^{best} \in C^*$.}
6: **end if**
7: $\mathcal{T} \leftarrow \mathcal{T} \cup t$ {Add topic to SeDAX network.}
8: **return** balanced SeDAX network.

3.4.4.1 Load-balanced Topic Addition

Load balancing by *load-balanced topic addition* means that best coordinates for new topics are determined before the actual topic addition to the overlay. Topic coordinates are chosen so that new topics have minimal negative impact on the load imbalance on the overlay.

Algorithm 1 shows the simplified steps that are necessary when a topic t is added to a balanced overlay. Regardless of whether a topic will be delegated or not, its home coordinate $h(t)$ and original coordinate $C(t)$ is calculated via geographic hashing. A set $\mathcal{C}^*$ of best delegation coordinates is constructed using one of the distributed coordinate selection algorithms described in Section 3.4.3. Applying a tie-breaker of Section 3.4.2.3 on this set yields the best coordinate c_{cand}^{best}. Topic t is delegated to c_{cand}^{best} if c_{cand}^{best} is better than $C(t)$. Otherwise, topic t is stored at $C(t)$. Eventually, topic t is added to the set $\mathcal{T}$ of SeDAX topics.

When topic loads remain static, any topic addition to a balanced overlay leads to a still balanced overlay with only minimal signaling effort because at most the newly added topic is delegated. When topic loads change, the optimized yet static topic assignment may lead to load imbalance on the overlay. We will investigate the effects of changing topic loads on an initially balanced SeDAX network in Section 3.5.2.

Algorithm 2 Continuous load balancing every $\Delta\tau$.

Require: node $v \in \mathcal{V}$, resilience level i

1: $t^v_{min} \leftarrow \underset{t \in \bigcup_{j=1}^{i} \mathcal{T}_j(v)}{\arg\min} L_T(t)$ {Smallest topic of v.}
2: $C^* \leftarrow$ Distributed coordinate selection
3: $c^{best}_{cand} \leftarrow$ Tiebreaker(C^*) {Best coordinate from C^*.}
4: **if** c^{best}_{cand} is better than $C(t^v_{min})$ **then**
5: $\quad C(t^v_{min}) \leftarrow c^{best}_{cand}$ {Delegate t^v_{min} to $c^{best}_{cand} \in C^*$.}
6: **end if**
7: **return** more balanced SeDAX network.

3.4.4.2 Continuous Load Balancing

In contrast, *continuous load balancing* means that balancing decisions are made during system runtime and that existing topics may be relocated. Each SeDAX node has a dedicated load balancer process that is triggered from time to time. The load balancer tries to shift topic load away from its node and then waits for another $\Delta\tau$ time. This process runs on each SeDAX node and does not require additional synchronization for triggering. This fully distributed approach allows the overlay to react on topic load changes.

Algorithm 2 shows the simplified steps when load balancing is triggered after $\Delta\tau$ for node v and resilience level i. The rationale is discharging nodes by delegating their smallest topics to other less-loaded nodes. We select the smallest topic t^v_{min} of all topics for which v is responsible for according to the desired resilience level; this may include topics for which node v is closest, second-closest or third-closest depending on the resilience level. The best coordinate c^{best}_{cand} for a potential topic delegation is determined analogously to Algorithm 1. If delegating t^v_{min} to c^{best}_{cand} decreases the node load imbalance compared to the original coordinate $C(t^v_{min})$, t^v_{min} is delegated to c^{best}_{cand}. Otherwise, t^v_{min} remains at $C(t^v_{min})$. Finally, the load balancer waits for the next load balancing trigger event after $\Delta\tau$.

3.4.4.3 Combined Approach

Load-balanced topic addition and continuous load balancing can be integrated into a *combined approach* to take advantage of the benefits of both approaches. In such a combined approach, load-balanced topic addition minimizes the impact of new topics to node load imbalance, and continuous load balancing adapts the topic coordinates to changing topic loads. We will investigate the impact of $\Delta\tau$ on load balancing quality and signaling effort when applying the combined approach to a SeDAX network in Section 3.5.2.

3.5 Impact of Distributed Load Balancing on Load Distribution

In this section, we investigate the impact of the proposed distributed load balancing algorithms on load distribution in SeDAX overlays. We evaluate the load imbalance for the different resilience levels (c.f. Section 3.4.2.1), for different topic characteristics, and in particular for topics with storage requirements growing over time. Finally, we investigate the trade-off between load balancing quality and signaling overhead.

3.5.1 Static Topic Sizes

This subsection investigates potential load imbalance in SeDAX overlays for static topic sizes by simulation experiments. First, the simulation setup is described. The CCDFs of node loads illustrate that the existing SeDAX can lead to significant load imbalance for which we analyze the causes. We show that load-balanced topic addition based on global information can equalize the load among all nodes and highlight the importance of respecting the resilience level for load balancing. As global information may be difficult to obtain, we show that simpler coordinate selection approaches can also lead to good load balancing results.

3.5.1.1 Experiment Setup and Methodology

We use a square plane as coordinate space on which n_{nodes} nodes are randomly positioned. Each node is assigned n_{node}^{topics} topics on average. We generate $n_{topics} = n_{node}^{topics} \cdot n_{nodes}$ topics, and each t of these topics comes with a random coordinate $h(t)$ in the square plane. These topics are iteratively added to SeDAX. When load-balanced topic addition is enabled, a load balancer may reassign each topic to a different coordinate $C(t)$ in the square plane depending on the current load situation in the overlay; otherwise the original random topic coordinates remain.

We study three choices for static topic loads.

- Homogeneous topic load: each topic has the same load $L_T = 1$.
- Heterogeneous topic load: 80% of the topics have load $L_T = \frac{1}{4}$, 20% of the topics have load $L_T = 4$. This distribution yields also an average load $E[L_T] = 1$ and its coefficient of variation is 1.5.
- Exponentially distributed topic sizes: $L_T(i) = e^{\lambda \cdot \frac{i}{(n-1)}}$ with $0 \leq i < n$. Their mean and coefficient of variation is $E[L_T] = 3.9247$ and $c_{var}[L_T] = 0.6472$ for $n = 100$ different topics and $\lambda = 2.3026$. We use that model in Section 3.5.1.4. Parameter λ is chosen that the topic sizes equal those of the topic growth model in Section 3.5.2.

After the successive generation of topics, assignment to coordinates, and load balancing, node loads are calculated for all nodes $v \in \mathcal{V}$ and the CCDF of these loads is determined. We perform each experiment 100 times, average the CCDFs from single simulation runs, and show 95% confidence intervals where appropriate. We use the same seeds for all corresponding experiments to use the same topic coordinates and sizes, i.e., to make the simulation results comparable with each other. The quantiles in the following evaluations are derived from the averaged CCDFs.

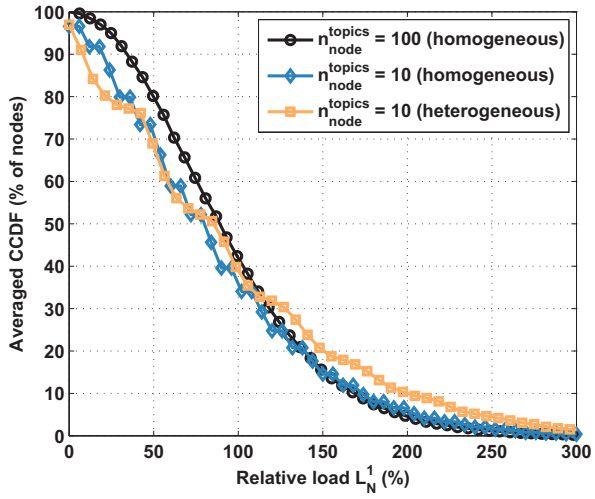


Figure 3.11: *CCDFs of the node loads L_N^1 for resilience level 1 demonstrate a significant load imbalance.*

3.5.1.2 Load Distribution without Topic Delegation

We first investigate the load distribution in SeDAX without load balancing. We simulate $n_{nodes} = 100$ nodes in the plane with $n_{node}^{topics} \in \{1000, 100, 10\}$ homogeneous-load topics per node on average and $n_{node}^{topics} \in \{100, 10\}$ heterogeneous-load topics per node on average. Figure 3.11 shows the CCDF of the node loads $L_N^1(v)$ for resilience level 1 for $n_{node}^{topics} \in \{100, 10\}$. The curve for $n_{node}^{topics} = 1000$ is omitted in the figure as it visually coincides with the curve for $n_{node}^{topics} = 100$. Node loads are relative, i.e., 100% relative load corresponds to a node load of n_{node}^{topics}. The lines are interpreted as follows: for a node load x on the x-axis, the y-axis gives the percentage of nodes whose node load X is greater than x. Thus, equal load on any node would result in a vertical line at 100% node load. The figure rather shows a continuous decrease over a load range between 0% and 250% for $n_{node}^{topics} = 100$. The curves for $n_{node}^{topics} = 10$ homogeneous-load topics have a slightly greater load imbalance which increases for heterogeneous-load topics.

Table 3.2: *Mean value $\bar{x}$, 5% and 95% quantiles of node load $L_N^i(v)$ for $n_{nodes} = 100$ without topic delegation.*

n_{node}^{topics}	L_N^i	$q_{5\%}$	$\bar{x}$	$q_{95\%}$	$q_{5\%}$	$\bar{x}$	$q_{95\%}$
		homogeneous topic loads			heterogeneous topic loads		
1000	L_N^1	29.5%	100.0%	194.7%			
	L_N^2	85.0%	200.0%	332.9%			
	L_N^3	122.6%	257.4%	411.6%			
100	L_N^1	24.8%	100.0%	192.5%	24.7%	100.0%	197.3%
	L_N^2	80.2%	200.0%	335.3%	78.6%	200.0%	345.7%
	L_N^3	117.3%	258.0%	410.6%	113.5%	259.2%	419.2%
10	L_N^1	12.0%	100.0%	204.0%	7.1%	100.0%	239.7%
	L_N^2	60.8%	200.0%	349.6%	37.3%	200.0%	391.7%
	L_N^3	102.0%	262.9%	433.5%	78.2%	271.3%	488.8%

Figure 3.12(a) shows in addition to the distribution of node load L_N^1 the distribution of node loads L_N^2 and L_N^3, i.e., the loads for resilience levels 2 and 3. The loads are significantly larger than the load of resilience level 1. While the L_N^1 loads have a mean of 100%, the L_N^2 loads have a mean of 200% because each topic has to be stored twice, and they range between 0% and 450% per node. The L_N^3 loads have a mean of about 260% and range between 0% and 550%. The mean of the L_N^3 load is less than 300% because topics can share the normally unused backup capacity of SeDAX nodes if they have different primary and secondary nodes. As exact values for load imbalance are hard to determine from the figures, Table 3.2 shows the 5% and 95% quantiles of the loads. We observe that the relative load imbalance increases with fewer topics per node and with increasing variance of topic loads. Furthermore, these values increase with increasing resilience level. The 95% quantiles may be useful for capacity provisioning. They can easily amount to 200% – 250% of the respective mean values. This is highly inefficient but necessary in the absence of load balancing capabilities.

Table 3.3: *Correlation coefficients between Voronoi cell size $A(v)$ and node load $L_N^i(v)$ for $n_{nodes} = 100$.*

$corr$	$n_{node}^{topics} = 1000$	$n_{node}^{topics} = 100$		$n_{node}^{topics} = 10$	
	homogeneous	homo-geneous	hetero-geneous	homo-geneous	hetero-geneous
L_N^1	0.9984	0.9844	0.9522	0.8680	0.6996
L_N^2	0.6830	0.6740	0.6528	0.6026	0.4914
L_N^3	0.6028	0.5954	0.5770	0.5336	0.4365

A good part of the strong load imbalance is caused by the strong imbalance of the Voronoi cell sizes. The average Voronoi cell size is $\frac{A_{square}}{n_{nodes}}$, where A_{square} is the area of the coordinate space in our experiment. If we take this as 100%, the 5% and 95% quantile of the cell sizes is 29.0% and 191.4%. This is very close to the quantiles of the load distribution with $n_{node}^{topics} = 1000$ homogeneous-load topics. Table 3.3 shows the correlation coefficients between the Voronoi cell size and the load of SeDAX nodes for different topic loads and resilience levels. We observe high correlations for all cases. The correlation is largest for resilience level 1 and 1000 homogeneous-load topics per node, and decreases for fewer topics per node, heterogeneous topic loads, and higher resilience levels. Thus, the observed load imbalance is largely due to different cell sizes. [2]

3.5.1.3 Load Distribution with Topic Delegation

We now examine the impact of load-balanced topic addition and the various coordinate selection algorithms presented in Section 3.4.3 on the load balancing outcome.

[2] We also conducted experiments with more and fewer nodes, but the results are so similar that we omit them here.

Load-Balanced Topic Addition Using Global Knowledge We first investigate load-balanced topic addition using coordinate selection based on global knowledge as proposed in Section 3.4.3.5. We add topics one after another to SeDAX and perform a load balancing decision for each new topic, i.e., whether it should be assigned to its default coordinate $C(t) = h(t)$ or to another recommended coordinate $C(t)$.

To validate the correctness of the load balancing results for load balancing goal L_N^3, we check that the following equation is met after the assignment of a topic with load $L_T^{assigned}$:

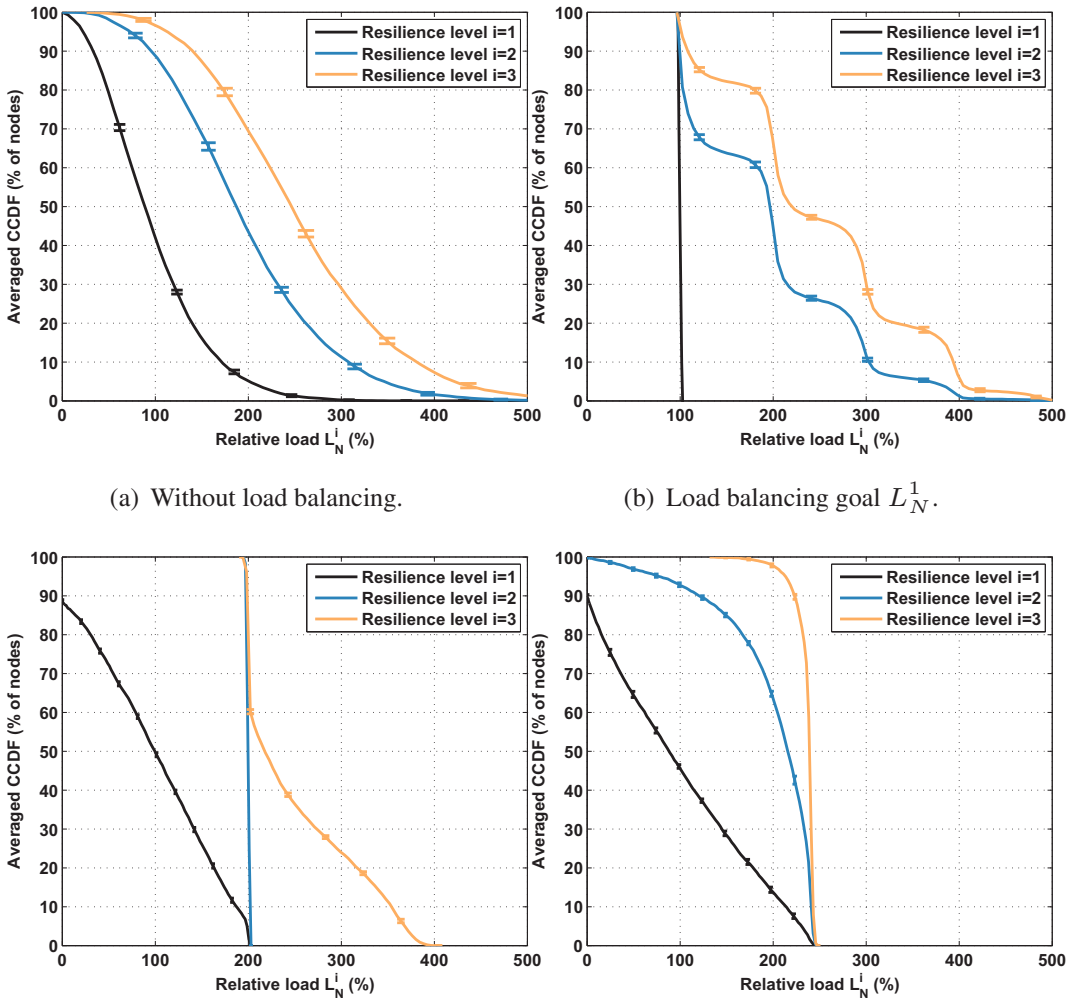
$$\min(L^3_{max,old}(c)) \leq \max(\max_{v \in \mathcal{V}}(L^3_{N,new}(v)) - L_T^{assigned}, \\ L^3_{N,new}(N_3(c_{assigned}))) \tag{3.16}$$

The subscripts "old" and "new" in the equation refer to the respective metric before and after topic addition, and $L_T^{assigned}$ and $c_{assigned}$ refer to the load and the coordinate of the last assigned topic.

In the following, we perform load-balanced topic addition with various objectives, namely to equalize the L_N^1, L_N^2, or L_N^3 load.

Equalizing L_N^1 Node Load Figure 3.12(b) illustrates the CCDF of the node loads L_N^1, L_N^2, and L_N^3 when topics are load-balanced for L_N^1. The L_N^1 load is well balanced over all nodes and the maximum L_N^1 load is near 100%. However, the L_N^2 load ranges between 100% and 500%. Thus, this simple load balancing approach does not lead to equalized data volumes on SeDAX nodes when SeDAX is operated under failure-free conditions in a resilient mode. For resilience level 3, the node load also ranges between 100% and 500%.

Equalizing L_N^2 Node Load Figure 3.12(c) shows the respective results when L_N^2 is used as load balancing goal. The L_N^1 load is almost equally distributed between 0% and 200% which is far from being equally balanced. However, the L_N^2 load is well equalized among all nodes, which is the balancing goal. That means, the data volumes on SeDAX nodes are about the same on all nodes when SeDAX is operated under failure-free conditions in a resilient mode. The



(a) Without load balancing.

(b) Load balancing goal L_N^1.

(c) Load balancing goal L_N^2.

(d) Load balancing goal L_N^3.

Figure 3.12: *CCDF of node loads L_N^1, L_N^2, and L_N^3 without load balancing and for load balancing goals L_N^1, L_N^2, and L_N^3.*

CCDF of the L_N^3 load shows the distribution of the maximum node load during single node failures. During single node failures, heavy load spikes in terms of additional load from other topics can occur on nodes with values ranging from 200% to 400%.

Equalizing L_N^3 Node Load Figure 3.12(d) presents the load distribution for load balancing objective L_N^3. The L_N^1 load is approximately uniformly distributed between 0% and 240% whereas the L_N^2 load is not. The maximum load node L_N^3 is about 240%; this means that no SeDAX node carries much more than 240% even during single node failures. This is a desirable feature even though the distribution of the actual load under failure-free operation is far from being equalized.

These investigations demonstrate that the load balancing objective for SeDAX needs to be carefully chosen. The simple L_N^1 load balancing goal cannot equalize the load of resilient SeDAX under failure-free conditions. The more complex L_N^2 load balancing goal achieves that objective, but cannot avoid load spikes during single node failures. Only the more complex L_N^3 load balancing goal is able to minimize load spikes during single node failures.

Load-Balanced Topic Addition Using Limited Knowledge Load balancing with coordinate selection based on global knowledge requires the calculation of the best coordinates of all SeDAX nodes and their communication to the load balancing node. That can be expensive in networks with many nodes and topics, so it is important to explore coordinate selection approaches that require less effort.

In the following, we examine the impact of the various coordinate selection algorithms presented in Section 3.4.3 on the load balancing outcome. We focus on balancing of the L_N^3 load with $n_{nodes} = 100$ nodes and $n_{node}^{topics} = 100$ heterogeneous-load topics. All investigated approximation algorithms are based on the principle of random queries. In all experiments, we use $n_{queries} = \{1, 10, 100\}$ queries per topic delegation decision, and perform load-balanced topic addition.

Table 3.4: *Impact of coordinate selection algorithms and $n_{queries}$ on mean value $\bar{x}$, 5% and 95% quantiles of node load L_N^3.*

	IndCoord			**BestLocalCoord**		
$n_{queries}$	$q_{5\%}$	$\bar{x}$	$q_{95\%}$	$q_{5\%}$	$\bar{x}$	$q_{95\%}$
—	113.5%	259.2%	419.2%	113.5%	259.2%	419.2%
1	176.8%	254.0%	282.3%	175.9%	245.3%	312.7%
10	225.5%	245.8%	251.1%	230.6%	235.6%	237.9%
100	234.2%	237.7%	239.1%	223.8%	235.4%	238.4%
	BestRegionalCoord			**BestGlobalCoord**		
$n_{queries}$	$q_{5\%}$	$\bar{x}$	$q_{95\%}$	$q_{5\%}$	$\bar{x}$	$q_{95\%}$
—	113.5%	259.2%	419.2%	113.5%	259.2%	419.2%
1	226.8%	235.8%	239.4%	213.5%	236.2%	243.3%
10	226.2%	235.1%	238.4%	213.5%	236.2%	243.3%
100	214.6%	236.1%	242.0%	213.5%	236.2%	243.3%

Table 3.4 shows the mean L_N^3 load, the 5% and the 95% quantiles of the averaged CCDFs of the experiments including the values without load balancing from Table 3.2 for comparison. The simulation results show that *all* selection algorithms can limit the 95% load quantile to about 240% while the 95% load quantile without load balancing is 419%, i.e., they reduce the 95% quantile of the load by as much as $\frac{419.2\%-237.9\%}{419.2\%} \approx 43\%$. However, IndCoord and BestLocalCoord require at least $n_{queries} = 10$ to achieve good results but that is feasible. Hence, distributed load balancing yields similar results as load balancing with global knowledge (BestGlobalCoord), but is more scalable. Nevertheless, all presented approaches are heuristics. The general load balancing problem maps to the NP-hard 0/1 knapsack problem and all proposed algorithms greedily assign topics to coordinates, one after another. Therefore, none of the results is likely to be fully optimal.

Table 3.5: *Distribution of node load L_N^3 for exponentially distributed topic sizes and varying topic addition order.*

Topic addition order	$q_{5\%}$	$\bar{x}$	$q_{95\%}$
Ascending size	207.7%	248.6%	264.6%
Descending size	205.7%	248.6%	257.1%
Random size	207.0%	248.3%	261.3%

The fact that coordinate selection algorithms with limited knowledge can outperform the coordinate selection algorithm with global knowledge seems surprising. By incrementally equalizing existing load before adding large topics, BestGlobalCoord can cause load spikes on a few nodes. In contrast, coordinate selection algorithms with limited knowledge equalize the load for only a limited set of coordinates, leading to a globally imperfect balance with larger load differences between coordinates. This leaves room for larger topics to be more evenly distributed, since when a large topic is assigned, the probability of a coordinate having significantly less load is larger than for BestGlobalCoord. Although this helps explain the observed phenomena, it also hints that future research can further improve coordinate selection algorithms, particularly for the investigation of load balancing in larger networks.

3.5.1.4 Impact of Topic Addition Order on Load Distribution

Finally, we investigate the impact of the topic addition order on the load balancing result. We simulate $n_{nodes} = 10$ and $n_{node}^{topics} = 10$ topics per node on average. We use exponentially distributed topic sizes (see Section 3.5.1.1) and perform load-balanced topic addition with BestGlobalCoord as coordinate selection algorithm to balance the node loads in each experiment run. We add the topics in *ascending topic size order*, *descending topic size order*, and *random order*. We perform each experiment 100 times and use the 5% and 95% quantiles of the averaged CCDFs of the experiments to calculate the load imbalance.

Table 3.5 shows the mean L_N^3 load, the 5% and 95% quantiles of the averaged CCDFs of the experiments. We observe that the topic addition order has some effect on the load balancing quality. Adding topics in descending topic size order leads to the best load balancing results because this addition order always leaves room for smaller topics to fill holes in the overlay. Conversely, adding topics in ascending order makes it more challenging for the last few topics to be placed on an already well-balanced overlay without causing some load imbalance. Random topic addition leads to an imbalance between both topic size orders.

3.5.2 Dynamic Topic Sizes

In this subsection, we assume that topic sizes grow over time, some grow slowly and some grow fast. We first present a model for this growth. We use it to illustrate the impact of heterogeneously growing topic sizes on load distribution in SeDAX without any load balancing. Then, we show how heterogeneous topic growth impacts load distribution after load balancing. Finally, we assume that topics are initially added to the system in a load-balanced way and then investigate the impact of continuous load balancing. The latter algorithm has only a single parameter and we illustrate its impact.

3.5.2.1 Model for Topic Growth

We assume n topics t whose sizes $L_T(t, \tau)$ grow exponentially over time $\tau \in [0, D]$ within an experimentation interval of duration D. Initially, all topics t have equal size $L_T(t, 0) = 1$. They grow with different rates so that the smallest topic is still of size 1 at the end of the experimentation interval and the largest topic is $L_T^{max} = 10$ large. Thus, the largest growth rate is

$$\lambda_{max} = \frac{ln\,(L_T^{max})}{D} \tag{3.17}$$

while the growth rate of the other topics is linearly spaced within $[0, \lambda_{max}]$. This yields exponentially distributed topic sizes as already used in Section 3.5.1.4.

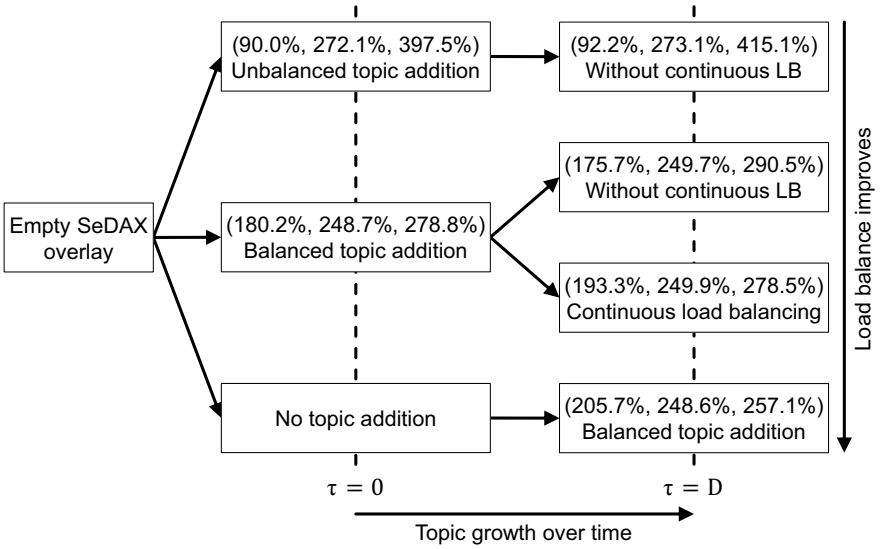


Figure 3.13: *Distribution of node load L_N^3 for dynamic topic sizes before topic growth (left-hand) and after topic growth (right-hand) under different load balancing configurations. The (x, y, z) values correspond to the 5% quantile, the mean, and the 95% quantile of the averaged CCDFs of the node load L_N^3.*

We simulate $n_{nodes} = 10$ nodes and $n_{node}^{topics} = 10$ topics per node on average. We quantify the load imbalance by the mean node load, and the 5% and 95% quantiles of the averaged CCDFs of the node load. We normalize node loads after topic growth for better comparison. That means, for all following experiments 100% normalized load corresponds to the sum of all topic loads $L_T(t, D)$ at the end of the experiment divided by n_{nodes}.

3.5.2.2 Impact of Topic Growth without Load Balancing

We first conduct reference measurements with an unbalanced system. We add all topics to the overlay without balanced topic addition, and then let all topics grow according to the topic growth model. The top row in Figure 3.13 shows the mean L_N^3 load, the 5% and 95% quantiles of the averaged CCDFs of the experiments before (left-hand) and after (right-hand) topic growth. We observe only minor changes in the mean load from 272.1% to 273.1% but more significant changes

in the load imbalance. The 95% quantile increases from 397.5% to 415.1%. This is because topic growth leads to a heterogeneous topic size distribution and increases the variance of the node loads. This observation is consistent with our initial investigation from Section 3.5.1.2 and the illustration in Figure 3.11. That means, fewer topics per node and fewer nodes in the system amplify this effect. Conversely, this also means topic growth has only a minor effect on load imbalance for higher numbers of topics per node and higher numbers of nodes. We use the values from the top row in Figure 3.13 as reference for comparison in the remaining part of the performance evaluation.

3.5.2.3 Impact of Topic Growth on Load-Balanced Topic Addition

We now investigate the impact of topic growth on load distribution in a balanced system. In contrast to the previous experiment, we now use load-balanced topic addition with BestGlobalCoord as coordinate selection algorithm to initially balance the node loads in each experiment run. When all topics have been added to the system, we let all topics grow according to the topic growth model without any further load balancing. The middle row in Figure 3.13 shows the mean L_N^3 load, the 5% and 95% quantiles of the averaged CCDFs of the experiments before (left-hand) and after (right-hand, upper box) topic growth. The mean load changes from 248.7% to 249.7% and the 95% quantile increases from 278.8% to 290.5%. That means, we observe a similar trend of load imbalance change after topic growth like in an unbalanced system. For completeness, we included the results for load-balanced topic addition after topic growth from Section 3.5.1.4 in the bottom row of Figure 3.13.

3.5.2.4 Benefits of Continuous Load Balancing

As final experiment, we perform continuous load balancing on initially balanced SeDAX overlays, i.e., effectively a combined approach, showing the influence and tradeoffs of the control parameter $\Delta\tau$ on the load balancing quality. A load balancer is triggered every $\Delta\tau$ for a randomly selected node which may reassign its smallest topic t to a different coordinate $C(t)$ based on the current load situation in the overlay.

Table 3.6: *Impact of control parameter $\Delta\tau$ on mean value $\bar{x}$, 5% and 95% quantiles of node load L_N^3.*

$\boldsymbol{\Delta\tau}$	$\boldsymbol{q_{5\%}}$	$\boldsymbol{\bar{x}}$	$\boldsymbol{q_{95\%}}$
—	175.7%	249.7%	290.5%
0.1	188.6%	249.2%	282.8%
0.01	193.3%	249.9%	278.5%
0.001	199.0%	249.8%	276.0%

Impact of Continuous Load Balancing on Load Distribution The middle row in Figure 3.13 shows the mean L_N^3 load, the 5% and 95% quantiles of the averaged CCDFs of the experiments before (left-hand) and after (right-hand, lower box) topic growth for $\Delta\tau = 0.01$. Mean load and 95% quantile change only minimally from 248.7% to 249.9% and from 278.8% to 278.5%. This is a significant improvement compared to the previous experiments without continuous load balancing, and demonstrates that our algorithm keeps the load imbalance constant over time for heterogeneously growing topics.

We now investigate the impact of different $\Delta\tau$ on load balancing quality and the necessary communication effort.

Impact of $\Delta\tau$ on Load Balancing Quality Table 3.6 shows the mean L_N^3 load, the 5% and 95% quantiles of the averaged CCDFs of the experiments for $\Delta\tau = \{0.1, 0.01, 0.001\}$. For easier comparison, we include the results from the experiments without continuous load balancing in the first row of the table. We observe that the load imbalance improves for smaller values of $\Delta\tau$. Compared to the result for topic growth after load-balanced topic addition (see Section 3.5.2.3), the 95% quantile improves by $\frac{290.5\%-276.0\%}{276.0\%} \approx 5\%$ but falls behind the heuristically achievable load balancing results for exponentially distributed topic sizes in Table 3.5 by $\frac{276.0\%-257.1\%}{276.0\%} \approx 7\%$. Nevertheless, the results for continuous load balancing are a good indicator that the proposed mechanisms can well handle dynamic topic load changes.

Impact of $\Delta\tau$ on Moved Load Rate Finally, we investigate the impact of $\Delta\tau$ on the *moved load rate* R_{ML}. This allows quantifying the tradeoff between improving load balancing and minimizing the moved load. We first give the necessary definitions to quantify R_{ML}.

Let $\delta(\tau) = [\delta_1(\tau), \delta_2(\tau), \ldots, \delta_n(\tau)]$ be a vector of size $||\mathcal{T}||$. Each $\delta_t(\tau)$ equals 1 if topic t was delegated at time τ; otherwise $\delta_t(\tau)$ equals 0. The *volume of moved load* $V_{ML}(\tau)$ at time τ is defined as

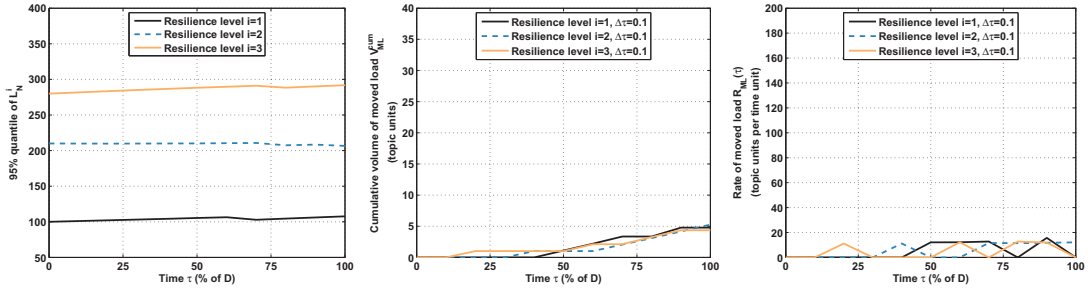
$$V_{ML}(\tau) = \sum_{t \in \mathcal{T}} \delta_t(\tau) \cdot L_T(t, \tau). \tag{3.18}$$

We calculate the rate of moved load $R_{ML}(\tau)$ over a time window of size $W = \frac{1}{10} \cdot D$ by

$$R_{ML}(\tau) = \frac{1}{min(\tau, W)} \cdot \sum_{\tau' = max(0, \tau - W) + 1}^{\tau} V_{ML}(\tau'). \tag{3.19}$$

Figure 3.14 shows the 95% quantile of node load L_N^i, the cumulative volume of moved load V_{ML}^{cum} and the moved load rate R_{ML} over the experimentation period D for load balancing goals L_N^1, L_N^2 and L_N^3, and varying control parameter $\Delta\tau$. The cumulative moved load volume V_{ML}^{cum} is given in topic units, the rate of moved load R_{ML} is given in topic units per time unit, and the 95% quantile is given as normalized node load. In this context, one topic unit corresponds to the load of a topic t with $L_T(t) = 1$. The plotted values represent the results from one arbitrarily selected simulation run of the 100 distinct simulation runs.

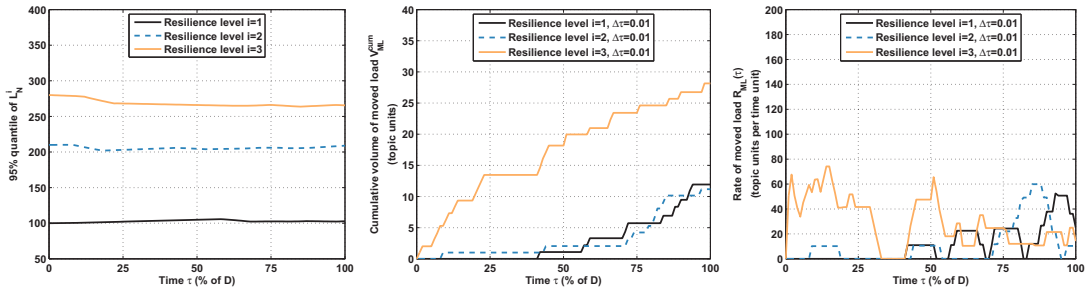
The time-dependent evolution of the 95% quantile of L_N^i in Figure 3.14(a), Figure 3.14(d) and Figure 3.14(g) shows that continuous load balancing can keep the system balanced over experimentation period D. Figure 3.14(b), Figure 3.14(e) and Figure 3.14(h) illustrate the influence of $\Delta\tau$ on the cumulative volume of moved topic load over time. We observe a significant increase in the amount of moved load between $\Delta\tau = 0.1$ and $\Delta\tau = 0.01$, and a saturation with regard to the absolute amount of moved load when comparing $\Delta\tau = 0.01$ and $\Delta\tau = 0.001$. Figure 3.14(f) and Figure 3.14(i) show the rates of moved load



(a) 95% quantile of L_N^i for $\Delta\tau = 0.1$.

(b) Cumulative sum of V_{ML} for $\Delta\tau = 0.1$.

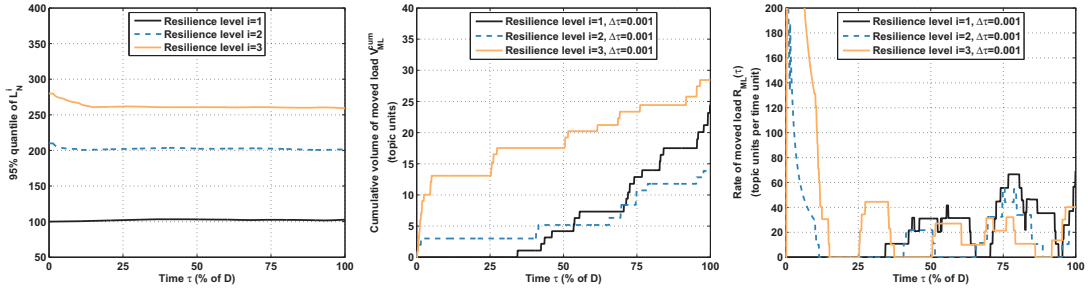
(c) Load move rate R_{ML} for $\Delta\tau = 0.1$.



(d) 95% quantile of L_N^i for $\Delta\tau = 0.01$.

(e) Cumulative sum of V_{ML} for $\Delta\tau = 0.01$.

(f) Load move rate R_{ML} for $\Delta\tau = 0.01$.



(g) 95% quantile of L_N^i for $\Delta\tau = 0.001$.

(h) Cumulative sum of V_{ML} for $\Delta\tau = 0.001$.

(i) Load move rate R_{ML} for $\Delta\tau = 0.001$.

Figure 3.14: 95% *quantile of* L_N^i, *cumulative volume of moved load* V_{ML}^{cum} *and rate of moved load* R_{ML} *over experimentation period* D *for load balancing goals* L_N^1, L_N^2 *and* L_N^3 *and varying control parameter* $\Delta\tau$ *of a selected simulation run of* 100 *simulation runs.*

R_{ML} for $\Delta\tau = 0.01$ and $\Delta\tau = 0.001$. We observe that the fast saturation of the cumulative volume of moved load for $\Delta\tau = 0.001$ comes at the price of a large and erratic moved load rate R_{ML} at the beginning of the experiment. In our experiments, $\Delta\tau = 0.1$ achieves worse load balancing results than $\Delta\tau = 0.01$. In contrast, $\Delta\tau = 0.001$ leads to equally good load balancing results as $\Delta\tau = 0.01$ at the price of very high communication overhead. $\Delta\tau = 0.01$ provides a good tradeoff between reasonable communication overhead and still very good load balancing results. We conclude that re-assignment during operation is challenging and causes significant communication overhead in the form of large, probably erratic load move rates. Therefore, the $\Delta\tau$ should be set to a reasonable value. This value depends on the number of topics, number of nodes, and their placement.

3.6 Lessons Learned

The objective of this chapter was to investigate the resource management issues of the SeDAX architecture. In the original SeDAX architecture, geographic hashing determines the coordinates of topics on the DT overlay. That means, neither load balancing nor delay optimizations are possible. We showed that this static assignment of topics to coordinates can lead to severe load imbalance on SeDAX nodes. We further observed that the strong load imbalance in existing SeDAX is due to topological structures, i.e., varying Voronoi cell sizes, and does not vanish with scaling to larger number of topics or nodes.

We investigated the impact of node placement on the load distribution. We developed a Monte-Carlo optimization for node placement in SeDAX to minimize storage requirements. We evaluated the capacity requirements of SeDAX with optimized node placement for homogeneous and heterogeneous node provisioning. The latter requires significantly less storage. In general, storage requirements depend on topic patterns. We derived the least storage requirements of SeDAX under optimal conditions and showed that they far exceed those of an idealized storage system. The reason is the inflexibility of the topic location in SeDAX.

We proposed a modification allowing dynamic reassignment of topics to coordinates while retaining the benefits of SeDAX, i.e., resilient overlay forwarding, decentralized control, and the ability to cope without a dedicated mapping system. Assignment of topics to chosen coordinates is desirable because it allows the system to move topics away from overloaded nodes or to move topics and clients closer to each other, i.e., enabling load balancing and delay optimization. We defined metrics for load on SeDAX nodes for three different levels of resilience to quantify the effect of topic movement. We developed load balancing algorithms and demonstrated that they work well for all considered resilience levels, i.e., they significantly reduce the 95% quantile of the load on all nodes. For resilient SeDAX that survives at least two node failures, the relative reduction of the 95% quantile of the load is 43% and also the amount of shared backup capacity is clearly reduced. As load balancing using global knowledge requires many information updates, which may raise scalability concerns, we also proposed simpler coordinate selections algorithms that work with only limited knowledge.

Further, we showed that a balanced SeDAX system may run out of balance if topic sizes change over time. Therefore, we presented a distributed algorithm for continuous load balancing offering a single parameter to trade load balancing quality against load balancing effort in terms of moved load rates. In our evaluations, it kept a balanced system well balanced when topic sizes grew exponentially over time with different rates.

We conclude that the resource management issues of SeDAX are inherent to its design. Although efficient use of available storage is not the primary goal of SeDAX, our optimizations improve the resource management of SeDAX while maintaining its compelling properties, namely scalability, automatic resilience, and security. We evaluated the load imbalance for different resilience levels, different topic characteristics, and in particular for topics with storage requirements growing over time. The proposed load balancing algorithms lead to well balanced load on SeDAX nodes while keeping load redistribution at a reasonable level. Thus, the distributed and resilient SeDAX pub/sub system can be managed in a distributed way both at its initialization and during operation.

4 Design and Evaluation of the C-DAX Architecture

In the previous chapter, we investigated the SeDAX architecture which was discussed and used as technological foundation for the C-DAX architecture at the beginning of the C-DAX project. Our evaluations showed that SeDAX is inflexible with regard to resource management and that those issues are inherent to its design. Eventually, our investigations convinced the C-DAX project consortium, and led to the design and specification of a new C-DAX architecture. In this chapter, we describe and evaluate this final C-DAX architecture.

The content of this chapter is mainly taken from [10,12,13,15,29,34] and organized as follows. We clarify the need for a novel C-DAX architecture and specify the initial architecture in Section 4.1. We detail the core features of the architecture in Section 4.2. We describe a simulation of C-DAX in the OMNeT++ simulation framework and the prototype implementation in Section 4.3. We analyze the strength and weakness of the C-DAX architecture with respect to alternative communication solutions in Section 4.4. Finally, Section 4.5 summarizes some condensed insights.

4.1 Background

We first discuss the reasons that eventually lead to the design of the novel C-DAX architecture. Finally, we give a description of the overall C-DAX architecture.

4.1.1 The Need for a Novel C-DAX Architecture

At the beginning of the C-DAX project, our former project partner Alcatel-Lucent introduced SeDAX as technological foundation. It utilizes the pub/sub and ICN communication paradigm, and comes with a cyber-security concept. The DT-based overlay network uses geographic routing for message distribution and offers self-healing against network and node failures. The latter provides automatic resource management to a certain extent.

We brought up our doubts that the resource management concept of SeDAX may lead to inefficient use of resources. To confirm our initial claims, we conducted several simulation and analytical studies. The outcome of these studies was the storage capacity analysis accompanied with a node placement optimization (see Section 3.3), and the design of an advanced resource management concept and its evaluation for SeDAX (see Section 3.4 and Section 3.5). We were able to significantly improve resource management in SeDAX, but some problems persist that cannot be solved without significantly changing the SeDAX architecture.

Long discussions in the C-DAX consortium eventually led to the decision to design and implement a novel architecture. We briefly summarize and justify the enhancements and changes during the transition from SeDAX to C-DAX in Table 4.1. The most significant architectural changes during the transition from SeDAX to C-DAX were:

1. Separation of control and data plane for improved robustness against node failures and improved flexibility of resource management

2. Replacement of the geographic routing and forwarding engine by a traditional, robust broker-based forwarding engine for reduced signaling complexity and improved forwarding performance

3. Relaxation of primary and backup topic placement constraints for improved resiliency against adjacent node failures and improved flexibility of resource management

4. Implementation of an own security architecture to primarily overcome the US export restrictions of the original security architecture of the SeDAX code base, and to apply established security primitives and libraries to C-DAX control and data plane traffic

5. Implementation of protocol-specific and generic adapters to interconnect SG application transparently over C-DAX; integrating existing and future SG applications in the original SeDAX architecture has not been defined

The first three changes were triggered by the investigations documented in Chapter 3 about the inflexibilities of the SeDAX architecture with regard to resource management and about the potential problems of the SeDAX resiliency mechanism. The rationale behind change four is trivial. The last change was necessary to add protocol support to the C-DAX prototype for the laboratory tests and the field trial. The concepts behind the last two changes may be adapted by other SG communication architectures, too.

Table 4.1: Summary of changes and enhancements during the transition from SeDAX to C-DAX.

Functionality	**Realization in SeDAX**	**Realization in C-DAX**	**Enhancement**
Broker discovery	Multi-hop geographic DT overlay	1-level or 2-level mapping system	Reduced lookup time
Data forwarding	Multi-hop geographic DT overlay	4-hop application layer forwarding (normal mode); 1-hop application layer forwarding (point-to-point mode)	Reduced number of hops; reduced end-to-end delay; reduced jitter
Resiliency	Primary and backup copies of a topic are handled on adjacent nodes in the DT overlay; data and control plane functionality is collocated on the same overlay nodes	Primary and backup copies of a topic are handled on arbitrary nodes; separate data and control plane	Increased robustness against node failures; flexible resource management

Table 4.1: Summary of changes and enhancements during the transition from SeDAX to C-DAX. (continued)

Functionality	Realization in SeDAX	Realization in C-DAX	Enhancement
Security	Not applicable due to US export restrictions	Topic key-based security architecture based on established security primitives and libraries guaranteeing end-to-end security, topic access control, and source authentication	Provides at least the same level of security as the original architecture but without the US export restrictions
Resource management	Inflexible and static; geographic hashing function determines where a topic (and its backup) will be stored in the DT overlay	Flexible and dynamic; management system (MgmSys) allows to place topics (and its backup) on arbitrary nodes, and to migrate topics to different nodes during runtime	Increased flexibility with regard to cloud operation, administration, and management
Scalability	Addition of more DT overlay nodes; overlay node coordinate selection has direct impact on overall topic placements and may cause significant signaling traffic	Control plane and data plane can be scaled independently by adding more designated nodes (DNs), data brokers (DBs), and resolvers (RSes)	Independent elastic scalability of control and data plane of the C-DAX cloud
Protocol adaptation	Not specified	Generic tunnel adapters; protocol-specific adapters	Increased support for a wide variety of current and future SG applications including legacy applications

4.1.2 The C-DAX Architecture

We give a broad overview on the C-DAX architecture, its design rationales, components, basic interactions, and briefly introduce its more advanced features. Details on C-DAX core features are omitted in the following for the sake of brevity and are, therefore, presented in Section 4.2 instead.

4.1.2.1 Design Rationale

Traditional power grid communication solutions are based on the client-server communication model. This requires both communication end-points to be aware of each other. Clients need to be configured with detailed communication parameters, e.g., IP addresses and port numbers of servers, and probably more communication protocol-specific parameters. Servers need to be configured properly to allow only access from trustworthy clients. When servers undergo a service cycle, all clients need to be re-configured to communicate with backup servers. When new clients are added to the system, the access control of the servers needs to be re-configured.

C-DAX uses the information-centric communication model instead of the client-server communication model. Information is organized in so-called *topics*. A topic is an abstract representation of a unidirectional information channel with a certain storage capacity; the storage capacity is the validity period of the stored information. A topic is addressed using its unique name and probably attributes, e.g., data type, location, and time. This allows C-DAX to support different applications over a unified communication solution at the same time, e.g., measurements, and grid control. An example for a topic is phasor measurement data for a specific geographic region inside the DG. Topics and topic names are key elements for the pub/sub and ICN paradigm.

The basic idea of the pub/sub paradigm is the decoupling of communication partners in space, time, and synchronization [40, 116]. Publishers and subscribers register at a *broker* for a certain topic. Publishers send messages for that topic to the broker, which eventually forwards them to the subscribers. In the RTSE use case, PMUs are publishers and PDCs are subscribers. The ICN paradigm is a global-scale version of the pub/sub paradigm. It provides finer-grained in-

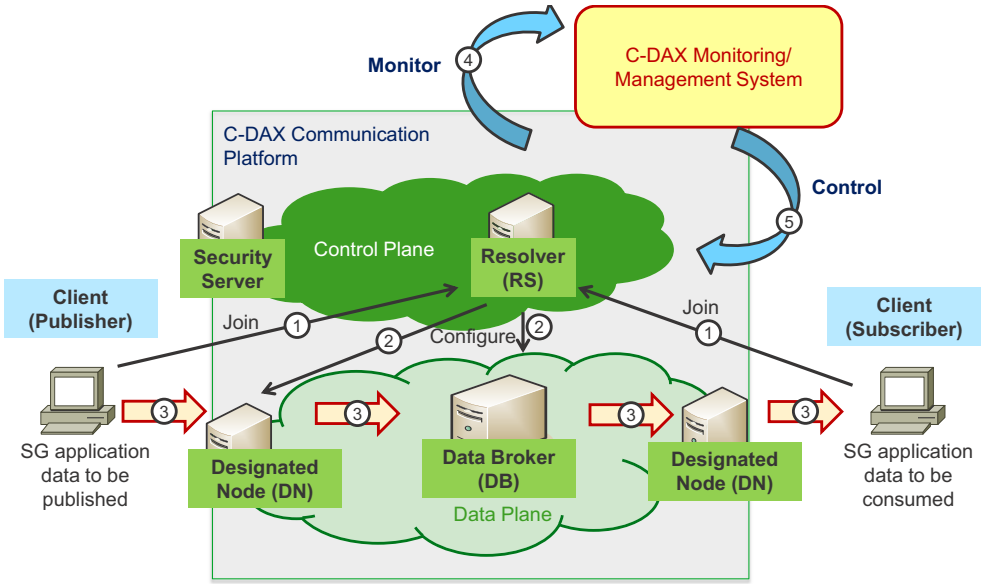


Figure 4.1: *The C-DAX architecture. Basic signaling steps include client join (step 1), data plane configuration (step 2), and topic data transmission (step 3). Further signaling includes monitoring (step 4) and general control of the C-DAX cloud (step 5).*

terface semantics for accessing information in the network compared to pure pub/sub, universal in-network caching, and content-oriented security [94]. The goal of applying pub/sub and ICN in C-DAX is to improve scalability compared to traditional client-server communication, and to facilitate development of new communication-based applications by providing a standardized transparent interface [39, 40, 42].

4.1.2.2 Components

Figure 4.1 illustrates the basic structure and interactions of the C-DAX architecture. It is composed of C-DAX clients and the C-DAX cloud. SG applications use *C-DAX clients* as interface to the C-DAX cloud, which handle all C-DAX signaling transparently to the respective application. *Publishers* are C-DAX clients generating data for a specific topic. *Subscribers* are C-DAX clients interested in certain topic data.

C-DAX nodes form the *C-DAX cloud*, and provide a specific set of functions to the cloud and clients. Possible functions are storage of topic data, resolving topic-to-node mappings, providing security functionalities, providing monitoring facilities, and providing management interfaces for operators. We briefly describe the functions from bottom to top, and assign them to their respective plane, e.g., data, control, or management plane.

Data Plane The *DNs* provide access for clients to the C-DAX cloud. They act as first point of contact and are responsible for forwarding topic data to and from the cloud, i.e., clients are pre-configured with DNs. The *DBs* store and forward topic data to DNs. Each topic is assigned to a DB where its publishers send topic data to. DBs store topic data for a certain time, and forward it to the topic's subscribers. The exact assignment of topics to DBs is subject to management decisions, and may be changed during runtime. Actual data plane communication is supported over TCP and user datagram protocol (UDP), configurable per topic.

Control Plane Topic names need to be mapped to DBs so that join requests can be sent to appropriate DBs that manage registrations. To that end, *RSes* hold topic-to-DB mappings and provide a resolution interface through which they answer mapping requests of other nodes. There may be several RSes in a C-DAX cloud, e.g., for resiliency or extensibility reasons. In that case, a *resolver discovery system (RDS)* is necessary which provides a mechanism to discover RSes when given a topic name. Security-related functionalities are provided by a *security server (SecServ)*, e.g., authentication, authorization, and key distribution. TCP is used for control plane communication.

Management Plane Management and monitoring is provided by the respective *MgmSys* and *monitoring system (MonSys)*. The MgmSys is responsible for topic and node management, and provides an operator interface for remote management. Topic management includes creation, deletion, migration, and configuration of topics during runtime. Topic migration allows operators to move topics from one set of DBs to another set of DBs, e.g., to perform load balancing. Topic configuration allows operators to change the attributes for a topic, e.g., changes

in the access control list of a topic. Node management enables addition and removal of a C-DAX node from the cloud. The MonSys provides mechanisms to gather and aggregate monitoring information. Depending on the actual management plane action, either C-DAX' pub/sub mechanism or TCP is used for communication.

4.1.2.3 Basic Interactions

We explain how topic data is published in C-DAX and how a subscriber can retrieve such topic data from C-DAX.

Publication of Topic Data Initial message exchange prior to topic data publication is shown on the left side of Figure 4.1. We assume that the publisher is authenticated by the SecServ and authorized to publish data to a topic. When the publisher wants to publish topic data, it first sends a join message to the RS over its DN using the topic identifier (step 1). The RS looks up its database for the topic-to-DB mapping. If such a mapping exists, the RS sends the responsible topic-to-DB mapping to the DN which installs a forwarding entry for that topic in its internal forwarding table (step 2). The publisher starts pushing data to its DN which forwards it to the responsible DB which stores the topic data (step 3).

Subscription to Topic Data Topic data retrieval works similar. Initial message exchange prior to topic data retrieval is shown on the right side of Figure 4.1. We again assume that the subscriber is authenticated by the SecServ and authorized to retrieve data of the topic. When the subscriber wants to retrieve topic data, it first sends a join message to the RS over its DN using the topic identifier (step 1). At the same time, the DN installs a topic-to-client entry in its internal forwarding table. The RS looks up its database for the topic-to-DB mapping. If such a mapping exists, the RS forwards the join message to the responsible DB which installs a topic-to-subscriber's-DN entry in its internal forwarding table (step 2), and starts pushing topic data to all registered subscriber's DNs (step 3).

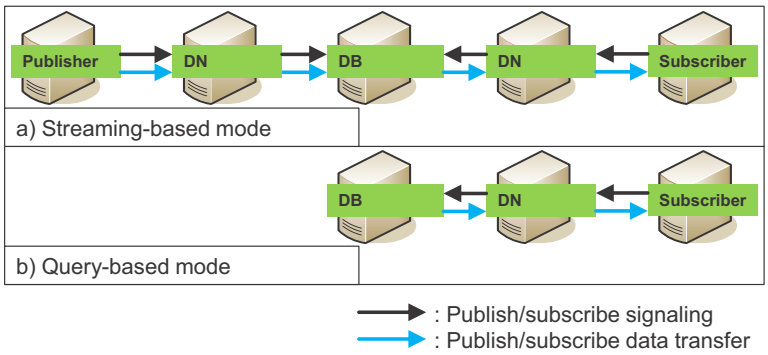


Figure 4.2: *Communication modes of C-DAX. Streaming-based (a) and query-based mode (b) are part of the initial C-DAX specification [26].*

Monitoring and Control of the C-DAX Cloud Any C-DAX node is a publisher to a special *monitoring* topic and publishes its node state information to that topic. This information is gathered and aggregated by the MonSys, which is a subscriber of this topic (step 4 in Figure 4.1). The MgmSys issues management commands to individual C-DAX nodes in order to perform topic and node management operations (step 5 in Figure 4.1).

4.1.2.4 Communication Modes

Figure 4.2 illustrates the initially specified communication modes of C-DAX: streaming-based and query-based communication. In *streaming-based mode* (see Figure 4.2a), subscribers continuously receive topic data after successfully joining a topic without requiring further explicit requests. In *query-based mode* (see Figure 4.2b), subscribers have to send explicit topic data queries to fetch specific topic data, e.g., a snapshot of streamed data. Modes are set per topic to fit the requirements of the application, e.g., low latency for PMUs or improved scalability for RETs on the REM [8]. While C-DAX' broker-based pub/sub mechanisms are well-suited for scalable information dissemination with regard to high numbers of publishers and subscribers, additional transmission delays are inherent to the initial design because of multi-hop application layer forwarding, and inter-

active (probably legacy) applications are prohibited due to the one-way pub/sub paradigm and potential dependencies on IP communication. Therefore, we introduce two advanced communication modes for the C-DAX architecture in Section 4.2.2, addressing those issues.

4.1.2.5 Security Concept

C-DAX security rationales are strong authentication of clients and nodes based on asymmetric cryptography, end-to-end security for topic data, minimal trust in the underlying infrastructure, and a flexible match of security parameters to the requirements of use cases. C-DAX nodes do not have to trust each other for secure operation, and clients do not have to trust the C-DAX cloud for guaranteed end-to-end security. We provide a more detailed description of the C-DAX security architecture and the key update mechanisms in Section 4.2.3.

4.1.2.6 Inter-Domain Concept

C-DAX enables utilities to cluster their infrastructure into *C-DAX domains*, i.e., sets of components of the same jurisdiction. Direct communication between clients and nodes of different domains may be restricted, e.g., due to business reasons, laws, operations rules, or security. Each domain operates DNs at its domain borders which provide a uniform interface for external subscribers, and hide the domain's network. DNs are responsible for forwarding inter-domain traffic, and for enforcing inter-domain security policies. A domain operator may operate multiple DNs to balance inter-domain traffic. The SecServ of the each domain manages the respective rights for its internal and external subscribers. We provide a more detailed description of the inter-domain concept in Section 4.2.5.

4.2 Core Features

In this section, we present the core features of the C-DAX architecture. We first present the resilience concept [10], advanced communication modes [15], security architecture [13], IEEE C37.118 adapter [12], and finally the inter-domain concept [29].

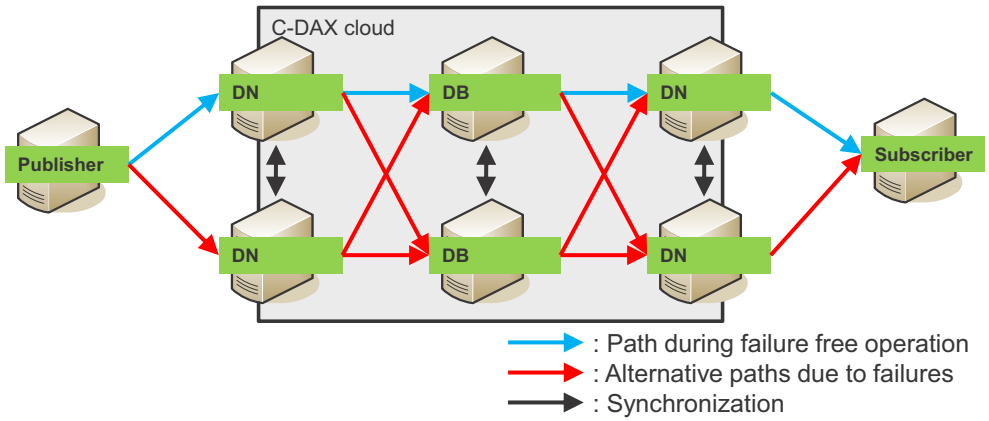


Figure 4.3: *The C-DAX resilience concept. Topic data is stored on two DBs. Each critical communication path is divided into a path during failure free operation (top) and alternative paths due to failures (bottom).*

4.2.1 Resilience Concept

We now describe the resilience concept of the C-DAX architecture. We first discuss the design rationale behind the concept and the envisioned resilience support levels. Then we specify the required signaling and depict actions upon node failure detection. Finally, we show experimental performance evaluation results based on the prototype implementation.

4.2.1.1 Design Rationale

Topic data should be highly available to SG applications, even in case of C-DAX component failures. In addition, resiliency should be transparent to and configurable by the actual SG application. Figure 4.3 shows the basic idea of the resilience concept in the C-DAX architecture. Component and data redundancy yields a simple yet robust resilience concept, enabling the infrastructure to survive in case of any component failure. Robustness here means that C-DAX should be able to cope with single component failures without additional communication with the MgmSys. Each client is configured with at least one primary and

backup DN with whom it may communicate, and each topic is stored on at least one primary and backup DB. Each critical communication path is divided into a path during failure-free operation (top paths in Figure 4.3) and alternative paths due to failures (bottom paths in Figure 4.3). Node failure detection is based on a heartbeat mechanism which we will elaborate on in Section 4.2.1.4.

4.2.1.2 Protected Failures

C-DAX' resilience mechanism primarily addresses the failure of DBs which are needed as forwarding nodes in a classical pub/sub system. Network failures such as link, switch, or router failures, should be rather protected by re-routing mechanisms. However, C-DAX' resilience mechanism can also limit the impact of a network failure when the network breaks into disconnected islands. Then, communication is possible among all C-DAX clients and nodes that still have a working path via reachable primary or backup DB.

4.2.1.3 Resilience Support Levels

A SG application may tolerate data loss, data delay, and failover delay to some extent. *Failover delay* includes the time for failure detection and successful failure recovery. It gives the lower bound of service unavailability time in case of a failure which must be dealt with by the SG application. *Data delay* means that time-stamped data may not be delivered with the original data rate. Reasons for data delay may be, e.g., intermediary buffering, network congestion, or retransmissions. *Data loss* means that topic data sent by publishers is not received by subscribers. Reasons for data loss may be, e.g., node failures and network failures.

Component and data redundancy allows for several meaningful communication patterns between publishers and subscribers. Depending on the communication pattern, different levels of resilience quality can be realized, which we summarize under the generic term resilience support levels (RSLs). We define four different RSLs as listed in Table 4.2, and describe them in detail in the following. RSLs are configured per topic during topic creation time.

Table 4.2: *Overview on C-DAX resilience support levels.*

Level	**Data loss** (during failover)	**Data delay** (during failover)	**Complexity**
RSL-0	Y	Y	Low
RSL-1	Y	N	Low
RSL-2	N	Y	Middle
RSL-3	N	N	High

Resilience Support Level 0: No Resilience For completeness, we include RSL-0 as the no resilience mode of C-DAX. There are certainly use cases where resiliency may not be necessary because the underlying applications can cope with temporary service degradation. Topics in RSL-0 are only stored on the primary DB, i.e., there are no backup DBs for topics. If the primary DB fails, data forwarding is interrupted until the DB problem is resolved, e.g., by restarting the failed DB, or by moving the topics to a non-failed DB.

Resilience Support Level 1: Data Loss Possible RSL-1 is the simplest resilience mode of C-DAX and it is the least complex RSL with regard to signaling and provisioning. In contrast to RSL-0, topic data is stored on primary and backup DBs. Topic data is sent unreliably[1] from publishers over the C-DAX cloud to subscribers. Should any intermediary node between publishers and subscribers fail, topic data will be dropped until the upstream node of the failed node switches to a configured backup node. That means, data loss depends on the response time of the node failure detection mechanism. The advantage of this RSL is that neither publishers nor intermediary nodes need retransmission buffers, i.e., it is cheap to implement.

Resilience Support Level 2: No Data Loss, But Delays Possible RSL-2 builds on top of RSL-1 and adds reliable data transmission. Topic data is now sent reliably from publishers over the respective primary DNs and primary DBs to the subscribers. Should any intermediary node between publishers and

[1] Depending on the underlying transmission protocol, see Section 4.1.2.2.

subscribers fail, topic data will be buffered at the upstream nodes of the failed node. After the upstream nodes successfully switched over to a pre-configured backup node, they re-send the buffered topic data to the backup node. Subscribers will not notice data loss but may experience data delay during the switchover process. That means, the experienced data delay depends on the response time of the node failure detection mechanism. Compared to RSL-1, RSL-2 requires more resources because retransmission buffers are necessary at publishers and intermediary nodes. Still, well-considered placement of topics on primary and backup nodes may allow for efficient backup capacity sharing.

Resilience Support Level 3: Near Real-Time Resilience RSL-2 is an improvement to RSL-1 with regard to data loss. Still, data delay may be a problem for near real-time SG applications. Using RSL-2 for such applications would require a very fast and highly reliable node failure detection mechanism which may itself introduce significant signaling load on the communication substrate. We therefore propose RSL-3 for near real-time resilience which resembles $1 + 1$ protection.

The key concept behind RSL-3 is simultaneous topic data transmission on disjunct data paths from publishers to subscribers. Within the limits of the system, RSL-3 provides reliable topic data delivery and close-to-zero data delay. Prerequisites for RSL-3 are perfect subscription synchronization of primary and backup nodes, and appropriate provisioning of the communication substrate. During failure-free operation, subscribers receive all topic data twice and perform duplicate data removal before handing the data over to the SG application. Should any intermediate node fail, data is still delivered to the subscriber. RSL-3 is the most expensive and complex solution compared to RSL-1 and RSL-2 because it also requires careful communication substrate planing and provisioning.

4.2.1.4 Node Failure Detection

Node failure detection has direct impact on the performance of RSL-1 and RSL-2. The involved components and the necessary signaling of the node failure detection mechanism of C-DAX are shown in Figure 4.4. It is implemented using hello messages and timers. That means, one component is periodically sending hello

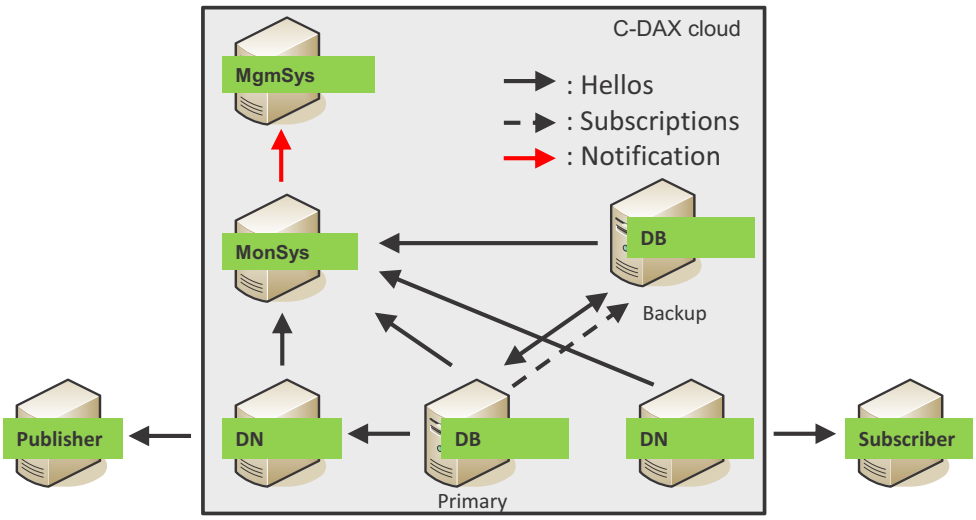


Figure 4.4: *Resilience signaling in C-DAX. Node failure detection is based on a heartbeat mechanism using periodic hello messages. Missing hello messages indicate node failures. Primary and backup nodes synchronize their subscriptions to guarantee smooth switchovers.*

messages and another component is receiving hello messages. After the reception of a hello message, the receiving component starts an internal timer. When the receiving component receives another hello message from the same sending component before the timer expires, the sending component is considered alive, the timer is restarted and the receiving component awaits the next hello message. When the timer expires before another hello message is received, the sending component is considered failed, and a failure event is raised at the receiving component. The timer value at the receiving component, called vulnerability window, has to be set carefully because network disruptions may cause hello messages to be dropped during regular operation, too. Otherwise, the receiving component may falsely assume a failed sending component.

Hello message signaling is applied in C-DAX as follows. All cloud nodes periodically send hello messages to the MonSys. In case of DNs, this information is only logged for monitoring purposes. In case of DBs, additional steps may take

place should a node failure be detected, e.g., determination and selection of a new primary or backup DB for the failed DB, triggering of topic migration operations to make the system ready for the next DB failure, and notification of the MgmSys. Clients receive hello messages from their connected DNs. This allows for a faster switchover to a backup DN should the primary DN fail compared to periodically querying the DN for availability. The MgmSys selects primary and secondary DBs at topic creation time while primary nodes synchronize subscriptions with backup nodes during operation, as will be elaborated in the following. In the latter, nodes refers to both DBs and DNs.

4.2.1.5 Subscription Synchronization

Subscription synchronization among primary and backup nodes yields fast node switchover without service degradation should the respective primary node fail because all necessary forwarding information is already available at the backup node. The subscriptions for a topic are stored on primary and backup DBs, and forwarding state is synchronized between the primary and backup DNs as well. There are several possible implementation options for subscription synchronization. One approach is to include proactive synchronization in the client join and leave process, e.g., clients send their join messages to the primary and backup nodes, which in turn have to know if they are the primary and backup node for the requested topic. When clients leave the cloud, their subscriptions are removed from any respective node. Another approach is to have a reactive synchronization signaling scheme in place, i.e., primary nodes in the cloud update the state of the backup nodes whenever a change in the subscriptions or forwarding occurs. This is also necessary when topics shall be migrated to different DBs inside the cloud. For the prototype implementation, we used the proactive subscription synchronization.

4.2.1.6 Actions Upon Failure Detection

C-DAX provides autonomous operation of the system should primary or backup nodes fail with minor service degradation and with only limited interaction with the MgmSys. We now describe the actions that take place upon failure detection.

Primary DB Fails When the primary DB fails, the MgmSys promotes the backup DB to the new primary DB. Then, the MgmSys selects a new backup DB and informs the new primary DB. The new primary DB synchronizes its subscriptions with the new backup DB. Publishers' DNs, aware of the primary DB failure, may query the RDS/RS for the new backup node, and send their data to the new primary DB. Eventually, the new primary DB sends the data to the subscriber DNs.

Backup DB Fails When the backup DB fails, the MgmSys selects a new backup DB and informs the primary DB. The primary DB synchronizes its subscriptions with the new backup DB. Publishers' DNs aware of the backup DB failure may query the RDS/RS for the new backup node but continue to send their data to the primary DB.

Primary and Backup DB Fail Simultaneously When both DBs fail simultaneously, the subscriptions are temporarily lost. In that case, the MgmSys selects a new primary and backup DB for the topics, and the publishers' and subscribers' DNs re-register via the RDS/RS and receive information about the new DBs.

DN of Publishers Fails When the primary DN of a publisher fails, the publisher may switch over to its backup DN, and send its data to the backup DN. Should the backup DN of a publisher fail instead, the publisher will notice this event, but not take any further actions. To make publisher more robust against DN failures, it may be configured with more than two DNs.

DN of Subscribers Fails When the primary DN of a subscriber fails, the subscriber will receive topic data from its backup DN instead. When the backup DN of a subscriber fails, the subscriber will continue to receive topic data from its primary DN. No additional signaling is necessary from the subscriber's perspective. Like for publishers, subscribers can be made more robust against DN failures by configuring more than two DNs.

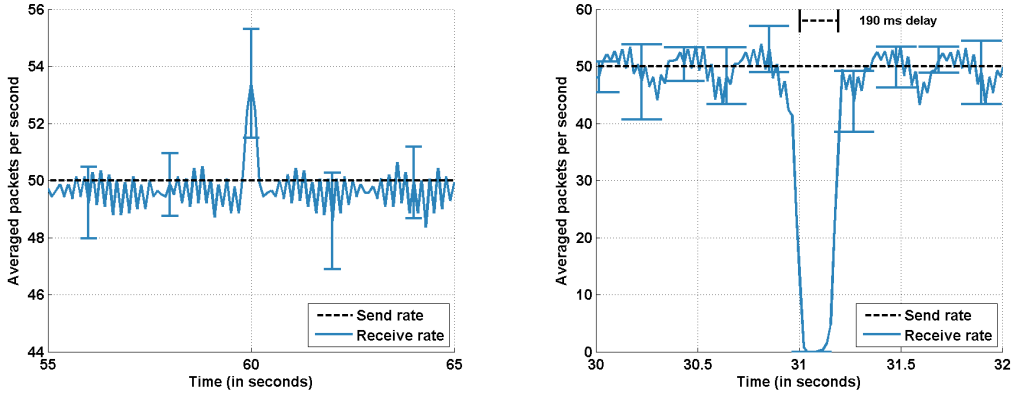
Cloud Core Components Fail Cloud core components are central components of the cloud and can fail as well. In C-DAX, this includes the MgmSys, the MonSys, the RDS/RS, and an initial set of DBs and DNs. In order to avoid a single point of failure, a redundant array of cloud core service nodes is operated which synchronizes its information. Thus, topic-to-RS and topic-to-DB mapping information is highly available.

4.2.1.7 Performance Evaluation

We now investigate the performance of the presented resilience concept by experimentation with the C-DAX prototype. The setup of experiments is described first, followed by experimental results from traffic experiments during failure-free operation and during a DB failure. Our results show data throughput for C-DAX before, during, and after a DB failure, and further demonstrate that the designed resilience mechanism performs fast and reliably.

Experiment Setup and Methodology To evaluate the performance of the resilience concept, we created a dumbbell-like topology with one publisher on the left side, the C-DAX cloud in the middle, and one subscriber on the right side. The C-DAX cloud is configured with one DN for publisher and subscriber each, and two DBs; the current prototype implementation supports RSL-2 only. We created one topic for PMU measurement data to which the publisher and the subscriber join. We used recorded IEEE C37.118-compliant [64] PMU measurement data provided by our C-DAX consortium partner École Polytechnique Fédérale de Lausanne (EPFL) as realistic workload; the publisher replayed the data set and sent 50 packets per second, and one interleaved configuration frame every 60 seconds.

We deployed our setup on a dedicated network testbed with 100 Mbit/s link bandwidth, and measured the data throughput of the C-DAX cloud at the subscriber side. This enabled us to measure the time and quality of service degradation during the actual DB switchover. Our data throughput measurement method is based on packet arrival timestamp sampling. We first log the time of each



(a) Failure-free operation.

(b) One DB failure after 30 seconds.

Figure 4.5: *Averaged packet receive rate at the subscriber side including* 95% *confidence intervalls. The dashed line represents the send rate at the publisher side. The peak at* 60 *seconds is part of the IEEE C37.118 PMU communication protocol [64] and represents a periodically interleaved configuration frame.*

packet arrival at the subscriber. Then, we sample the recorded timestamps with a higher frequency than the send rate, i.e., we count the number of packet arrivals during one sample period, and retrieve the receive rate. We performed each experiment 50 times with each experiment running for 70 seconds, averaged the throughput measurements, and show the 95% confidence intervalls.

Failure-Free Operation We first assume that no nodes fail. We start the data replay at the publisher and measure the data throughput at the subscriber. The results are shown in Figure 4.5(a); the dashed line represents the send rate of the publisher. The subscriber receives topic data with a rate of 50 packets per second with a small peak at 60 seconds as expected. We recognize fluctuations in the data throughput which stem from the network substrate of the network testbed. We use these values as a benchmark for failure-free operation.

One DB Failure We re-use our previous experiment setup and emulate the failure of a DB during regular operation. First, all nodes and clients are started, the publisher replays the data, and we wait until we have a stable receiving rate at the subscriber. After 30 seconds, we disconnect the primary DB of the PMU measurement topic, and measure the time until the receiving rate at the subscriber is stable again, i.e., until the switchover finished successfully. The results are shown in Figure 4.5(b); the dashed line represents the send rate of the publisher. Before the DB failure, the subscriber receives topic data with an average rate of 50 packets per second; these results are in-line with our measurements during failure-free operation. During the switchover, we can see a drop in data throughput down to 0 packets per second. After successful switchover, the data throughput returns to the same level as before the DB failure. The switchover time is less than 190 milliseconds. This shows that our proposed mechanism works fast and reliably.

4.2.2 Advanced Communication Modes

Future SG applications can be designed specifically for the pub/sub paradigm or can be adapted to it [12]. However, legacy applications like SCADA involve bidirectional communication or other paradigms incompatible with pub/sub and delay-sensitive SG applications such as RTSE may benefit from a direct communication mode to minimize end-to-end delay. Therefore, we introduce two advanced communication modes for the C-DAX architecture which address the requirements of delay-sensitive and interactive SG applications in the following: *broker-less pub/sub mode* and *transparent IP-tunneling mode*.

4.2.2.1 Broker-Less Pub/Sub Mode

In *broker-less pub/sub mode* (see Figure 4.6a), publishers send data directly to subscribers without DNs and DBs involved in the actual data transmission. This violates the decoupling of publishers and subscribers but is the only option for use cases requiring extremely low latency, e.g., RTSE.

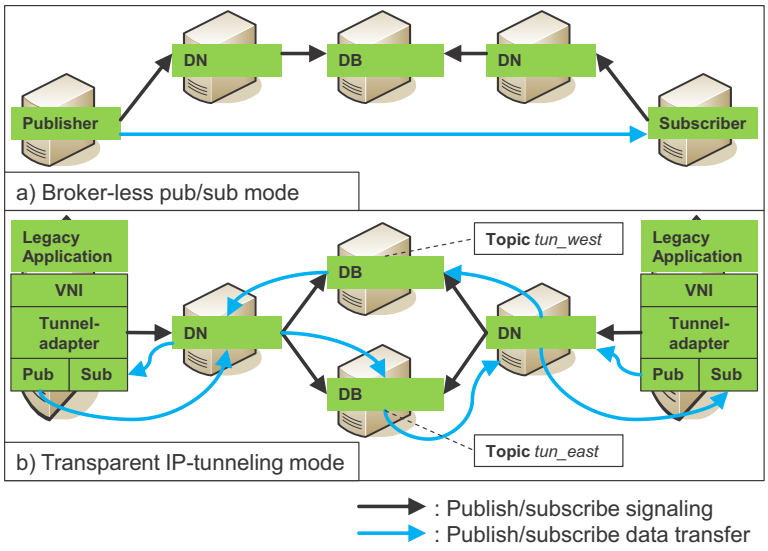
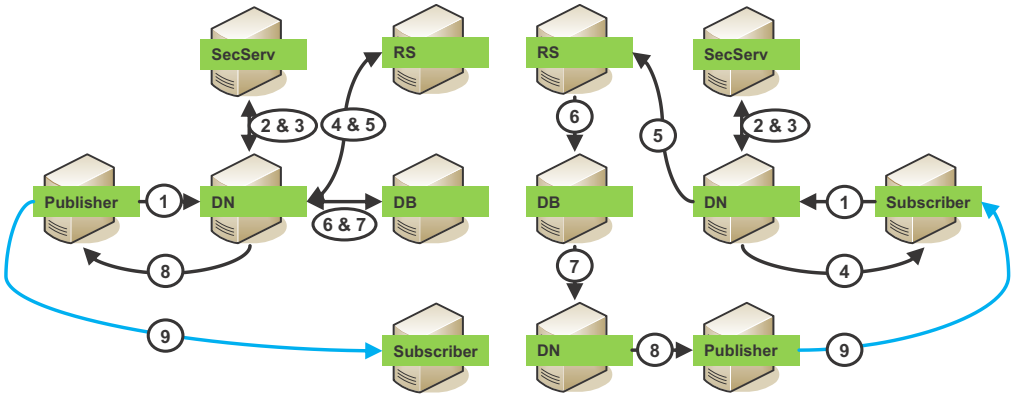


Figure 4.6: *Advanced communication modes for C-DAX. Broker-less pub/sub mode (a) uses pub/sub signaling for publisher and subscriber discovery during client join. Transparent IP-tunneling mode (b) uses two topics (e.g.,* `tun_east`, `tun_west`*) to realize bidirectional pub/sub communication.*

Design Rationale The broker-less pub/sub mode requires a different signaling compared to broker-based pub/sub communication. DNs remain as first point of contact for clients, but additional information needs to be stored at DBs and publishers. Additionally to topic-to-subscriber-DN mappings for broker-based pub/sub, DBs store two new kinds of mappings for broker-less pub/sub: (1) topic-to-publisher-DN mappings, and (2) topic-to-subscriber mappings. The rationale behind storing mapping (1) at the DB instead of at the publisher is that clients must not interact with other C-DAX nodes but DNs by design. Publishers store topic-to-subscriber mappings. This is only necessary for real-time topics and is expected to be manageable because of the potentially small number of subscribers in such use cases, e.g., a utility may run one or two PDCs for all its deployed PMUs, thus, requiring only up to two entries per C-DAX PMU client.



(a) Publisher-join signaling. After successfully joining C-DAX in steps 1 to 3, publishers discover their subscribers in steps 4 to 8, and eventually start forwarding data in step 9.

(b) Subscriber-join signaling. After successfully joining C-DAX in steps 1 to 4, publishers are notified about the newly joined subscriber in steps 5 to 8. In step 9, the newly joined subscriber starts receiving topic data.

Figure 4.7: *Basic signaling for broker-less pub/sub mode.*

Basic Signaling We assume that publishers and subscribers join a broker-less topic in the following. As depicted in Figure 4.6a, join-specific signaling is sent over the C-DAX cloud whereas the actual data transmission takes place between publishers and subscribers only.

Publisher-Join Signaling When a publisher joins a broker-less topic, as shown in Figure 4.7(a), it sends a join message to its DN (step 1), which in turn will authenticate and authorize the publisher against the SecServ (steps 2 and 3). The DN queries the RS for the DB responsible for the topic (steps 4 and 5) and forwards the join message to the responsible DB (step 6). The DB returns the list of subscribers for the broker-less topic to the DN (step 7), which in turn forwards the list to the publisher (step 8). The publisher updates its internal topic-to-client mappings and starts forwarding data to its subscribers.

Subscriber-Join Signaling Subscriber join signaling works similarly. When a subscriber joins a broker-less topic, as shown in Figure 4.7(b), it sends a join message to its DN (step 1), which in turn will authenticate and authorize the subscriber against the SecServ (steps 2 and 3), and signals successful join back to the subscriber (step 4). The DN forwards the join message inside the C-DAX cloud to the RS, which in turn forwards the join message to the DB responsible for the topic (steps 5 and 6). The DB internally looks up the list of responsible publisher DNs for the topic and forwards the join message to all responsible DNs, which in turn forward the join message to the appropriate publishers for the broker-less topic (steps 7 and 8). Finally, the publishers update their internal topic-to-subscriber mappings and start forwarding data to their subscribers (step 9).

4.2.2.2 Transparent IP-Tunneling Mode

In *transparent IP-tunneling mode* (see Figure 4.6b), any IP-based application can communicate over C-DAX, taking advantage of C-DAX' security, management, and resilience features; it is a compatibility feature for transparent integration of IP-based legacy applications in C-DAX, e.g., SCADA.

Design Rationale The transparent IP-tunneling mode uses virtual network interfaces (VNIs) and *tunnel adapters* to connect IP-based applications over C-DAX. The tunnel adapter is a C-DAX client which acts as a publisher and a subscriber, and provides secure, resilient bidirectional communication over C-DAX. The bidirectional communication of the tunnel is mapped to topic-based pub/sub by using a special topic per tunnel endpoint. That means, the tunnel adapters at both ends of a tunnel need to join the topic associated with the local end as a subscriber and the topic corresponding to the remote end as a publisher. Figure 4.6b depicts the tunneled communication over one topic per tunnel direction. Each tunnel has exactly two endpoints, and each tunnel adapter is part of exactly one tunnel.

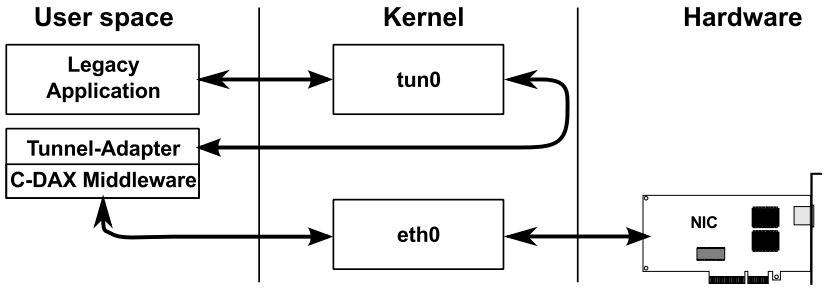


Figure 4.8: *Information exchange and signal flow for transparent IP-tunneling over virtual and physical network interfaces.*

Implementation and Configuration The prototype implementation is based on the Linux tun/tap [117] VNI. We use the `tun` mode of the tun/tap driver to provide the network traffic to a user space application as IP packets. IP packets sent over the tun interface are redirected to user space software reading from a file descriptor. IP packets written to that file descriptor appear as received packets at the tun interface.

Figure 4.8 illustrates the operation of the tunnel adapter. Legacy applications send or receive data over the virtual tun interface. The tunnel adapter is connected to the tun interface, encapsulates IP packets received from the tun interface into C-DAX messages, and sends the encapsulated messages over the physical network interface to the next C-DAX node. If a message is received from the physical network interface, the tunnel adapter extracts the inner IP packet from the C-DAX message. This IP packet is sent over the virtual tun interface to the legacy application.

Configuring the tun interfaces as point-to-point interfaces with the remote end IP address as peer address is sufficient if the applications are running on the tunnel endpoints. For applications running on dedicated hardware, modifications to the forwarding tables are required. At the application, the local tunnel endpoint needs to be configured as gateway for the respective remote application's IP prefix and vice versa. Additionally, each tunnel endpoint needs to have a entry for the remote IP prefix with the respective remote endpoint as gateway.

4.2.2.3 Discussion

The C-DAX streaming mode involves multi-hop application layer forwarding for data dissemination. Without violating the basic C-DAX signaling and data plane forwarding, four hops are necessary to forward topic data from a publisher over the publisher DN, the DB, and the subscriber DN to finally arrive at the subscriber. Each application layer hop increases the end-to-end delay by processing delay, e.g., performing security checks, internal lookups, or interaction with the network. Furthermore, additional path stretch is possible due to non-least-cost routing.

In contrast, broker-less pub/sub mode enables one-hop data dissemination avoiding additional path stretch and intermediary processing delays because only publishers and subscribers are involved in the actual communication. This makes it the communication mode of choice for real-time applications. The only drawback is the more communication-intensive client join signaling compared to the broker-based pub/sub mode signaling. Still, the minimized end-to-end delays for the actual data transmission outweigh this drawback.

While the pub/sub paradigm is well-suited for scalable information dissemination, interactive applications relying on the traditional client/server communication paradigm are prohibited by design. The transparent IP-tunneling mode addresses this shortcoming. IP-based legacy applications can be supported without the need to modify existing legacy hardware and software, or to implement protocol-specific compatibility layers. Proprietary applications can even be supported without knowledge of the protocol characteristics (i.e., any specifics defined above the IP protocol), as long as IP communication is supported.

4.2.3 Security Architecture

We now describe the security architecture of C-DAX, present methods for distributing updated symmetric keys for data plane communication and discuss their properties. We first discuss the design rationale behind the security concept, introduce the basic terminology, and finally specify the required mechanisms and keys which are used to implement those properties in C-DAX.

4.2.3.1 Design Rationale and Terminology

Topic data transmission should be protected end-to-end because

1. only legitimate publishers may publish data for a certain topic,
2. only legitimate subscribers may receive data from a certain topic, and
3. third parties (including DNs, DBs, and malicious clients) must not modify or spoof topic data without detection.

The actually required security properties for the topic data transmission may vary depending on the smart grid applications and the C-DAX middleware must be capable of supporting them.

The security architecture of C-DAX provides *topic access control*, *end-to-end integrity* and *end-to-end confidentiality* of published data, and *authentication* of clients and nodes. We describe those security features in detail below. In contrast to more innovative solutions for security in information-centric SG middleware presented in [51], the current C-DAX middleware uses authentication and encryption mechanisms based on standard cryptographic primitives, i.e., it can be implemented based on established and trusted security libraries. The C-DAX security architecture does not restrict the type of cryptographic primitives (i.e., symmetric or asymmetric) used to secure the communication. Nevertheless, for performance reasons we rely mainly on symmetric primitives to enforce the data plane security properties.

We write $\mathcal{T}$ for the set of all topics and K_t for a *topic key* associated with a topic $t \in \mathcal{T}$. Topic keys are generated by the SecServ, and the SecServ distributes the topic keys to the respective components as part of the join response message. Table 4.3 provides an overview of the keys used in C-DAX, the component or topic the key is associated with, and the components that know the key.

4.2.3.2 Security Properties

We now detail the security properties.

Table 4.3: *Overview on key types in C-DAX.*

Name	Description	Associated With	Known By
K_c^-	Component private key	component c	only known by component c
K_c^+	Component public key	component c	may be known by all components
$K_{SecServ}^+$	SecServ public key	SecServ	must be known by all components
K_t^{auth}	Topic access control key	topic t	publishers, DNs, and DBs for topic t
K_t^{e2e}	End-to-end security key	topic t	publishers and subscribers for topic t
$K_t^{e2e_x}$	Diversified end-to-end security key	topic t, publisher x	only known by publisher x for topic t

Source Authentication Source authentication is required for control plane messages. When processing request messages, the SecServ needs to verify the identity of the component before looking up the permissions of the requesting party in its access control list (ACL). The same requirement applies to configuration messages where DBs need to verify that such a message originates from an authorized node, e.g., an RS.

Source authentication is realized using asymmetric cryptography., e.g., Rivest-Shamir-Adleman (RSA). Each component is assigned a public/private key pair (K^+, K^-). Control plane messages are digitally signed using the private key K_{sender}^- of the respective sender. The receiver can verify the signatures using K_{sender}^+.

As usual, certificates are used to link these keys to identities. Certificates issued and signed by the SecServ provide identity information, the associated public key, and additional attributes. The additional attributes include C-DAX function information about permission to modify node configurations, e.g., for the RS function. Certificates can be attached to the signed messages. Additionally, the SecServ provides a certificate revocation list (CRL) to allow certificates to be revoked.

Because of the decoupling of publishers and subscribers, source authentication for data messages is not available in most pub/sub systems. Even though source authentication is not needed for pure topic based communication, there are SG

applications (e.g., RETs) where messages from a publisher could lead to a binding contract. For such applications, digital signatures generated with K^{-}_{sender} can be used to authenticate the sender of a data plane message.

Topic Access Control Topic access control is required for all topics to prevent unauthorized clients from publishing data. We use a shared symmetric key K_t^{auth} to implement topic access control. This key is used to compute hash MACs, and is shared among authorized publishers and involved forwarding nodes for topic t. When publishing a data message msg for topic t, publishers use this key to add MAC(K_t^{auth}, msg) to the message. The forwarding nodes verify the MACs of incoming messages and only forward messages with valid MACs; otherwise messages are discarded.

End-to-End Integrity End-to-end integrity for all topics enables subscribers to verify the integrity of received topic data without having to trust intermediate forwarding nodes. The topic key K_t^{e2e} is introduced to implement end-to-end integrity in C-DAX, and is used as a shared secret to generate a MAC. In contrast to K_t^{auth}, the SecServ distributes K_t^{e2e} only to the publishers and the subscribers of topic t, i.e., the forwarding nodes do not know K_t^{e2e}. Subscribers can verify that an original message msg was not altered during forwarding when the received message contains MAC(K_t^{e2e}, msg).

End-to-End Confidentiality Confidentiality means that only the intended receivers of a message can read the message content. Because control plane and data plane in C-DAX do not share the same concept of receivers, we use different mechanisms to ensure end-to-end confidentiality for control plane and data plane messages.

Control Plane Messages C-DAX control plane communication consists of point-to-point messages, i.e., only the single intended receiver of a message should be able to read the message content. End-to-end confidentiality is especially important for all control plane messages containing topic keys. We use asymmetric cryptography to achieve this requirement. The SecServ uses the public key K_c^{+} of a component c to encrypt the topic key K_t.

Data Plane Messages Data plane communication in C-DAX is essentially many-to-many communication, i.e., only the subscribers of topic t should be able to read messages published to that topic. End-to-end confidentiality is required for transmission of personal data, e.g., smart metering data for residential buildings. Asymmetric encryption using individual public keys of the subscribers is not possible for data plane messages. As the pub/sub paradigm decouples publishers and subscribers, the publishers do not know the subscribers and their respective public keys. Therefore, we use symmetric ciphers to encrypt the payload of pub/sub data plane messages using the topic key K_t^{e2e}, e.g., Advanced Encryption Standard (AES).

As mentioned above, only publishers and subscribers receive K_t^{e2e}. DNs and DBs cannot decrypt the actual message content but can detect and discard unauthenticated messages because they possess K_t^{auth}. However, if the forwarding configuration can be manipulated, publishers are able to decrypt messages sent to the same topic by other publishers. Diversified keys for publishers can be used to prevent this, as shown in Figure 4.9. Then subscribers are still supplied with K_t^{e2e}, which now acts as a master key, while each publisher receives a unique identifier x and derived key $K_t^{e2e_x}$, derived from the master key for this value of x using some KDF: $K_t^{e2e_x} = \text{KDF}(K_t^{e2e}, x)$. Messages encrypted by a publisher using $K_t^{e2e_x}$ now need to include the publisher's x in the unencrypted message header. On reception of a message, subscribers derive the symmetric key for decryption and MAC verification $K_t^{e2e_x}$ using the KDF, K_t^{e2e}, and x. Publishers cannot derive the keys of other publishers without knowing the master key K_t^{e2e}, so publishers can no longer decrypt any data plane messages except their own. A similar approach is proposed in the Resilient End-to-end Message Protection framework (REMP) [52] protocol, a security protocol for SeDAX.

4.2.3.3 Application of the Security Mechanisms

We now show how the security properties are ensured by applying the security mechanisms described above. We give an example of how the mechanisms are applied during publication of confidential topic data, and provide a summary of the mechanisms used in C-DAX.

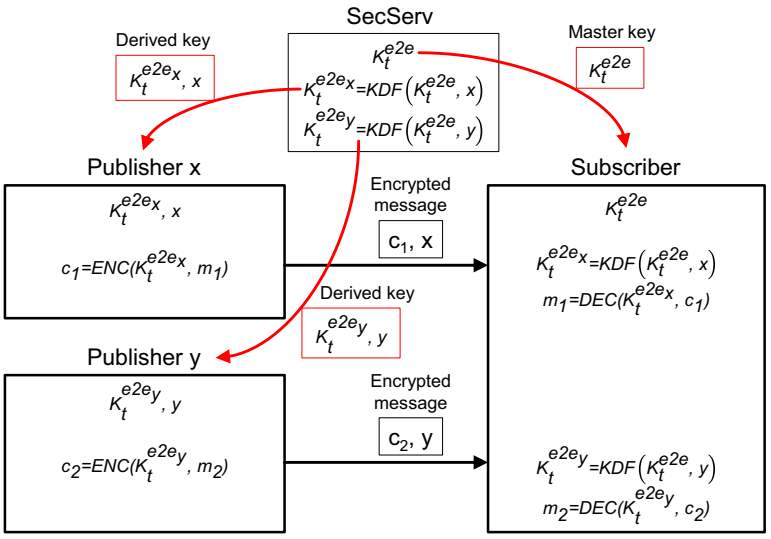


Figure 4.9: *Diversified keys prevent publishers from decrypting messages sent to the same topic by other publishers. The SecServ derives publisher keys ($K_t^{e2e_x}$ and $K_t^{e2e_y}$) from the master key K_t^{e2e}, distributes those keys together with unique identifiers (x and y) to publishers and distributes the master key K_t^{e2e} to subscribers. Plaintexts (m_1 and m_2) are encrypted with the publisher's key and sent as cryptotexts (c_1 and c_2) together with the publisher's identifier to the subscribers. Subscribers derive the decryption key from the master key K_t^{e2e} using the KDF and the publisher's identifier, and decrypt the cryptotexts to plaintexts (m_1 and m_2).*

Figure 4.10 depicts the publication of data over the C-DAX cloud. The SecServ distributes the topic keys encrypted with the public keys of the clients and nodes. Publishers and subscribers receive both K_t^{auth} and K_t^{e2e} (step 1) while the forwarding nodes only receive K_t^{auth} (step 2). The publisher encrypts the message using K_t^{e2e} and generates one MAC using K_t^{auth} and another MAC using K_t^{e2e} (step 3). The DNs and DBs forward the message after verifying the MAC using K_t^{auth} (step 4). After receiving the message, the subscriber uses K_t^{e2e} to verify the MAC and decrypt the payload (step 5).

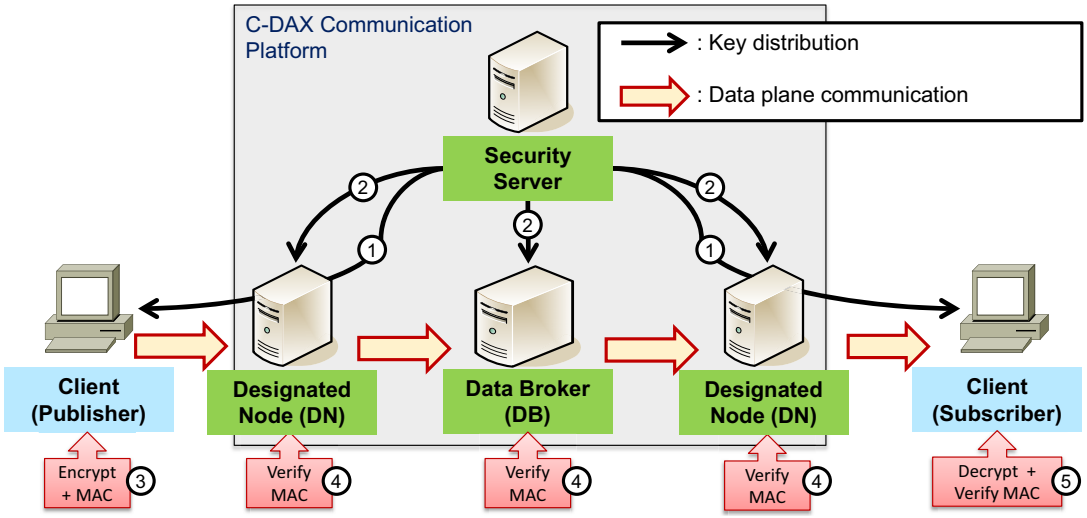


Figure 4.10: *C-DAX security mechanisms applied for publication of topic data.*

Table 4.4: *C-DAX components and supported operations.*

Component	Mechanisms										
	Key generation	ACL lookup	Certificate revocation	Signing	Signature verification	MAC generation	MAC verification	Asymmetric encryption	Asymmetric decryption	Symmetric encryption	Symmetric decryption
SecServ	X	X	X	X	X	-	-	X	-	-	-
DB	-	-	-	X	X	-	X	-	X	-	-
DN	-	-	-	X	X	-	X	-	X	-	-
Publisher	-	-	-	X	-	X	-	-	X	X	-
Subscriber	-	-	-	X	X	-	X	-	X	-	X

X: supported; -: not supported

Table 4.4 maps the C-DAX components to the mechanisms they need to support for their operation. While key generation, authorization, and asymmetric encryption is only performed by the SecServ, all components have to support signing. The components involved in data plane communications (i.e., clients and forwarding nodes) need to support asymmetric decryption of topic keys. Additionally, the forwarding nodes need to verify signatures and MACs. The publishers need to generate MACs and perform symmetric encryption. Subscribers have to verify MACs and need to do symmetric decryption. For use cases like retail energy transactions they also need to verify publisher signatures.

4.2.3.4 Key Distribution Mechanisms

We now describe key distribution mechanisms to securely distribute new topic keys based on the pub/sub mechanisms already provided by the C-DAX infrastructure. We first describe the requirements and prerequisites for secure distribution, propose two key distribution mechanisms, and finally discuss approaches for scheduling key updates

We use the notation of the symmetric topic key $K_t^{*,i}$, where K_t^* can be one of the keys K_t^{e2e} or K_t^{auth}, and i is the index in a chronological series of K_t^* for topic t. When a key update is performed for a topic t with the current key $K_t^{*,i}$, the updated key is denoted as $K_t^{*,i+1}$. Because updated keys need to be delivered not only to subscribers but also to publishers, we define a corresponding *key-update topic* t' for each regular topic t. The original subscribers and publishers for topic t are subscribers of topic t', and the SecServ is the only publisher for topic t'.

Requirements To make sure that messages for a topic originate from legitimate publishers and can only be read by legitimate subscribers, *backward secrecy* and *forward secrecy* are required for the topic keys. We use the definitions from Steiner, Tsudik and Weidner [118] that have been adopted for the terms forward and backward secrecy in later literature [119]. *Backward secrecy* is defined as the guarantee that "old, previously used group keys must not be discovered by new group members". *Forward secrecy* is defined as the guarantee that "new keys must remain out of reach of former group members".

In the case of K_t^{e2e}, full forward and backward secrecy is required to prevent subscribers from decrypting messages that were not published during their subscription period. That means, the topic encryption key needs to be changed each time a subscriber joins or leaves the topic. For K_t^{auth} only forward secrecy is required because MACs generated with previous keys cannot be used for publishing data. Therefore, K_t^{auth} only needs to be changed when a publisher leaves the topic t.

To maintain forward secrecy, the new topic key $K_t^{*,i+1}$ cannot be transmitted encrypted using the old topic key $K_t^{e2e,i}$. Therefore, we must rely on asymmetric encryption to distribute the new keys and individually encrypt $K_t^{*,i+1}$ using $K_{c_1}^{+}...K_{c_n}^{+}$, with $\mathcal{C}_t = \{c_1, ..., c_n\}$ being the set of publishers and subscribers for topic t.

Distribution of Asymmetric Keys As a prerequisite for secure key distribution in C-DAX, the component's asymmetric key pair (K_c^+, K_c^-) and the public key $K_{SecServ}^{+}$ of the SecServ are pre-installed on each component c. Those key pairs are intended to be long-term keys, i.e., they are only changed if the original key is considered compromised or otherwise insecure. As there is no secure way to remotely install a new key on a device whose keys can no longer be trusted, manual intervention is required anyway. Therefore, we do not define automated update mechanisms for this.

Topic Key Update: a Push Approach As a naïve solution, the SecServ can distribute updated keys by publishing them to t'. The distribution of the updated keys could be done in individual messages or concatenated to one large message. This approach has two major scalability drawbacks.

The first problem is that the SecServ needs to transmit n encrypted keys through all DBs and DNs. Clients would receive multiple keys but can only decrypt one of them. To reduce this overhead at the receiver side, filters deployed at DNs can reduce the number of keys delivered to the individual clients at the cost of increasing DN complexity. Alternatively, separate topics could be created per client at the cost of increasing the complexity of DBs and topic management.

The second problem is that the SecServ is required to know the current subscription state of each topic to select the required set of public keys for encryption of the topic keys. Keeping the subscription state can be avoided by using the ACL as source for the set $\mathcal{C}_t$. On the other hand, this could lead to unnecessary key transmissions and would prevent wildcards from being used in the ACLs.

Topic Key Update: a Pull Approach Alternatively, a pull mechanism can be used for key update notification which does not suffer from the drawbacks of the push mechanism. For this we use the topic-based pub/sub communication only to advertise the key update event, but not to publish the actual keys. The SecServ publishes a simple unencrypted notification message to topic t', and the clients are responsible for requesting a new key upon reception of the key update notification message.

The procedure of notification and subsequent key retrieval request is shown in Figure 4.11. The update notification is published by the SecServ to the DB and forwarded via DNs to the clients (step 1). The clients then send a signed message to the SecServ via their DN to request the new topic keys (step 2). The SecServ sends the new topic keys encrypted with the respective public key to the clients (step 3). For the sake of readability, requests of DBs and DNs to retrieve K_t^{auth} are omitted in the figure. The key retrieval process is similar to the topic join process described in Section 4.1.2.

Compared to the push approach, the pull method needs additional messages (notification and retrieval request). On the other hand, the pull method helps to avoid unnecessary key transmissions, and the SecServ does not need to keep track of the current subscription state. However, this optimization of the SecServ comes with an increased risk of denial of service (DoS) attacks because by providing a mechanism to actively retrieve topic keys from the SecServ, clients have the opportunity to trigger expensive asymmetric encryption operations at the SecServ by sending multiple key retrieval requests. To prevent clients from overloading the SecServ with unnecessary key retrieval requests, DNs may cache the encrypted topic keys for the clients they are serving. Any subsequent key retrieval requests for the same client and topic can be handled by the DN without involving

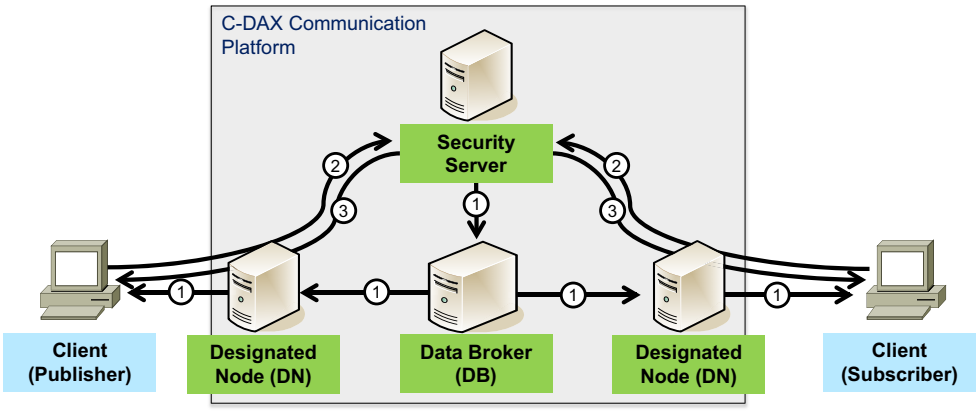


Figure 4.11: *Update notification and key retrieval (Pull mechanism).*

the SecServ while the key is valid. DNs can remove the old cached key should they receive an unencrypted key update notification from the SecServ, and cache the new key during the topic key retrieval process of a client.

Key Update Triggers Key updates can be scheduled periodically by configuring a key lifetime and replacing it after expiration. Key updates can also be triggered by join or leave events and ACL changes. The advantage of periodical key updates is that there is no means to attack the SecServ by intentionally causing key updates. However, topics with low fluctuation in the set of publishers and subscribers benefit from event-triggered key updates because unnecessary key updates can be avoided.

Key Transitions We need a mechanism for a seamless transition because the simultaneous replacement of $K_t^{*,i}$ by $K_t^{*,i+1}$ at all involved parties is impossible. Therefore, subscribers need to preserve the outdated key $K_t^{*,i}$ for a short time after the key update. If publishers switch to the new keys with a small additional delay and include an identification number like the index i of the key used into the message, the transition $K_t^{*,i} \rightarrow K_t^{*,i+1}$ can be performed without the risk of delivering messages to subscribers that are not yet or no longer in possession of the required key.

4.2.4 IEEE C37.118 Adapter

In this subsection, we review the communication part of IEEE C37.118, the current standard for synchrophasor measurements in power systems, and provide an adapter-based solution to easily connect and integrate entities in a synchrophasor network over the pub/sub C-DAX architecture. IEEE C37.118 offers two different modes for client-server communication, but cannot be used unchanged over pub/sub communication architectures. The proposed adapters offer standard-compliant communication between the synchrophasor measurement network entities to facilitate the exchange of measurement data. This work was an essential enabler for the C-DAX project's field trial, which will be briefly summarized in Section 4.3.2.

4.2.4.1 IEEE C37.118: A Standard for Synchrophasor Measurement in Power Systems

IEEE C37.118 is the current standard for synchrophasor measurement in power systems and divided into two documents: one document describing the phasor measurement in power grids [63] and one document describing the communication architecture [64]. In the following, we describe the components and the communication in synchrophasor networks according to that standard.

Synchrophasor Networks A *synchrophasor network* is a hierarchically organized network and consists of two components: PMUs and PDCs. PMUs measure the equivalent phasor representation of the power-system waveforms (i.e., voltages and currents) in different points of the power grid, time-stamp each measurement using a reliable time source, such as global positioning system (GPS), and send the time-stamped measurement data to the PDC. PDCs receive measurement data from PMUs, aggregate data from different PMUs based on the time-stamp, and optionally forward the aggregated data to superordinate PDCs or provide them to applications. Figure 4.12 shows an example of a two-level synchrophasor network and illustrates that measurement data flows are unidirectional from lower-layer to higher-layer devices.

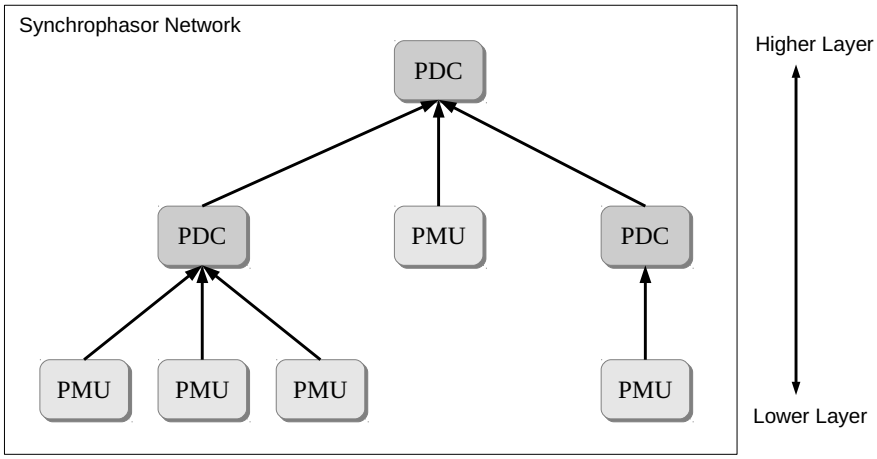


Figure 4.12: *An example of an IEEE C37.118 synchrophasor network.*

The standard defines the terms *server* and *client* as follows. A server is a function or device providing phasor measurement data, e.g., a PMU or an intermediate PDC. A client is a function or device receiving phasor measurement data, e.g., a PDC. An intermediate PDC may also provide measurement data to another PDC. As a result, a device can be both server and client. While the server-side function is providing measurement data to other devices, the client-side function is receiving measurement data. Each stream in a synchrophasor network device is identified and addressed by a 16-bit *IDCODE*, i.e., IEEE C37.118 can be deployed on top of any transport protocol because information from lower protocol layers is ignored, e.g., IP address and port number.

Messages The standard defines four types of messages: command (CMD), data (DATA), header (HDR), and configuration (CFG). *CMD messages* are sent from clients to servers and contain the server stream's IDCODE. They may switch data streaming on and off, or request CFG and HDR messages. *DATA messages* are sent from servers to clients, and contain the server stream's IDCODE, and single or aggregated PMU measurement data. They cannot be interpreted without knowing the current configuration of the PMU. Servers send client *HDR messages* containing general information about the PMU(s), scaling,

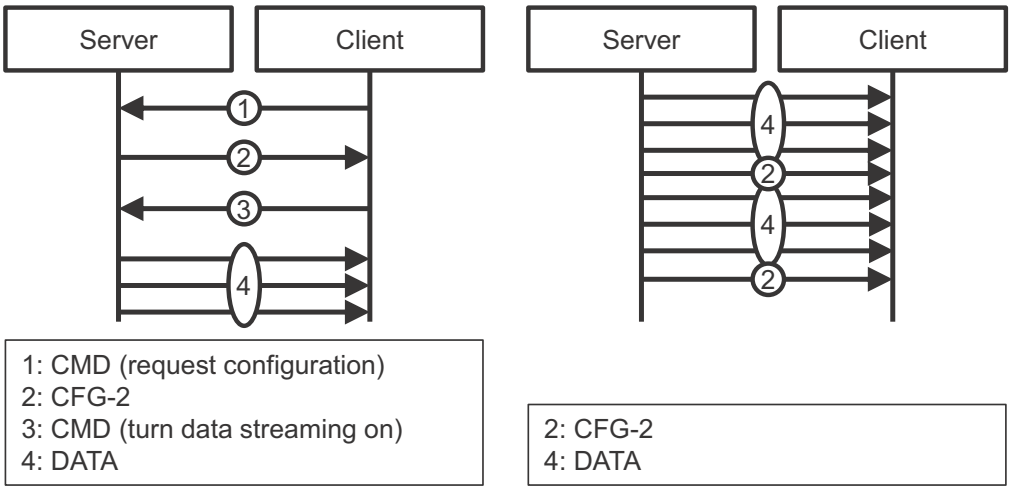
algorithms, and filtering. HDR messages are not used for synchrophasor data streaming.

The standard defines three types of *CFG messages*: CFG-1, CFG-2, and CFG-3. CFG messages are sent from servers to clients and contain the server stream's IDCODE. CFG-1 messages contain generic capability information of the queried PMU device and are not used in the data streaming context. CFG-2 messages contain the current configuration of the PMU device which is necessary to interpret the measurement data. CFG-3 messages are extended CFG-2 message enabling advanced phasor measurement features, e.g., flexible framing or global PMU IDs. We will use CFG and CFG-2 for configuration messages interchangeably.

Communication Modes The standard describes two modes of operation with regard to communication: commanded mode and spontaneous mode.

Commanded Mode PMUs and PDCs interact with each other using bidirectional communication in *commanded mode*, as shown in Figure 4.13(a). First, the PDC requests the PMU configuration by sending a CMD message with the *request-CFG-2* option to the PMU. After successful retrieval of the CFG-2 message, the PDC switches data streaming on by sending a CMD message with the *turn-streaming-on* option to the PMU. Eventually, the PMU starts to continuously stream measurement data to the PDC. The PMU triggers its PDC to actively request a new CFG-2 message if the PMU configuration changes. The PDC sends a CMD message with the *turn-streaming-off* option to the PMU to switch data streaming off. The standard recommends using UDP for measurement data but supports TCP as well. Further, having a TCP-based control channel *and* a UDP-based data channel is also possible.

Spontaneous Mode In contrast to commanded mode, PMUs send unsolicited DATA and CFG messages to PDCs over UDP in *spontaneous mode*, i.e., there is no communication in the reverse direction, as shown in Figure 4.13(b). The rate of DATA messages depends on the PMU measurement configuration.



(a) Commanded mode. Servers send DATA and CFG messages to clients on request.

(b) Spontaneous mode. Servers send DATA and CFG messages to clients unsolicitedly.

Figure 4.13: *IEEE C37.118 communication modes. Servers are functions or devices providing phasor measurement data, e.g., PMUs or intermediate PDCs. Clients are functions or devices receiving phasor measurement data, e.g., PDCs.*

The rate of CFG-2 messages depends on the general PMU configuration. Streaming CFG-2 messages in regular intervals is necessary because PDCs can interpret the received measurement data only after receiving the current configuration of the PMUs, i.e, the PDC automatically learns about a PMU configuration change without any additional communication overhead.

4.2.4.2 Integration of IEEE C37.118 in Publish/Subscribe Communication

In this section, we clarify the need for integration of IEEE C37.118 in pub/sub communication. We propose the concept of publisher and subscriber adapters, and show how they can solve the problem. We discuss implementation alternatives, their pros and cons, and configuration considerations. Finally, we summarize the current implementation.

The Need for Integration of IEEE C37.118 in Publish/Subscribe Communication Existing products for PMUs and PDCs implement IEEE C37.118 as communication interface. Commanded mode is not suitable for pub/sub communication because it requires bidirectional one-to-one communication, and pub/sub systems support unidirectional many-to-many communication only. Spontaneous mode does not require a reverse channel, i.e., data delivery over pub/sub is meaningful but additional configuration on the PMU side is necessary to properly set a streaming target. Commanded mode is widely used in existing large-scale PMU installations, e.g., the Synchrophasors Initiative in India [67]. Additionally, off-the-shelf products for PMUs and PDCs may offer only commanded mode prohibiting data transport over pub/sub communication infrastructures, e.g., the open-source iPDC [120] simulator suite. To include such PMUs or PDCs in settings with pub/sub communication, translations between commanded mode and spontaneous mode are needed before and after data are transmitted over pub/sub. Furthermore, IEEE C37.118 messages need to be translated into a format compatible with the pub/sub data plane and converted back when delivered to the application. Otherwise, the pub/sub communication architecture cannot process and forward IEEE C37.118 messages.

Publisher and Subscriber Adapters We propose publisher and subscriber adapters that perform communication mode and message translation to integrate IEEE C37.118 with pub/sub communication. Adapters provide interfaces for native pub/sub and IEEE C37.118 communication. They are full-fledged publishers and subscribers, and allow clients and servers of a synchrophasor network to exchange data over a pub/sub communication infrastructure.

When a publisher or subscriber adapter is started, it first joins the pub/sub network as a publisher or subscriber for its pre-configured topics. After appropriate signaling with a server (PMU or PDC), the publisher adapter sends synchrophasor data over the network. Likewise, the subscriber adapter forwards synchrophasor data to the client PDC after appropriate signaling.

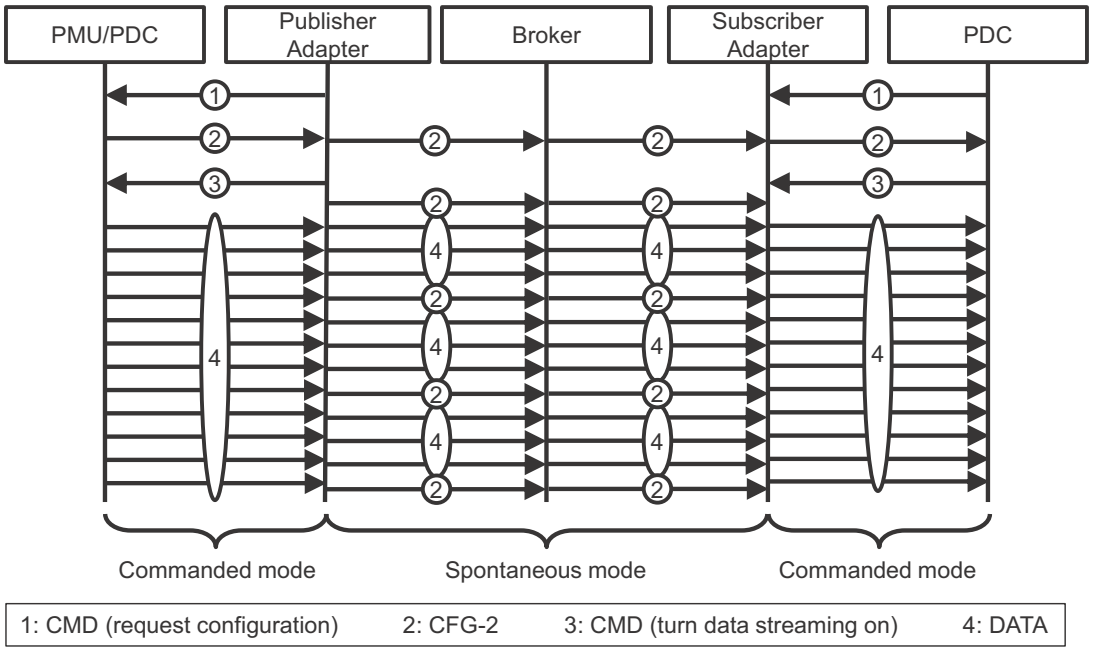


Figure 4.14: *Communication mode translation of publisher and subscriber adapters. The publisher adapter requests CFG-2 frames from the server, toggles data streaming, and forwards DATA frames with interspersed CFG-2 frames over the pub/sub communication infrastructure to the subscriber adapter. The subscriber adapter listens for incoming commands from the client and forwards CFG-2 and DATA frames on request.*

All subscriber adapters receive the same topic data should they join the same topic, e.g., a second PDC is able to process the topic data because it receives the same CFG-2 and DATA messages as the first PDC. Should no topic data be available for a topic because no publishers exists yet, the subscriber adapter signals that according to the IEEE C37.118 protocol. Subscriber adapters silently discard all received DATA messages if their PDC did not switch on data streaming; this is valid behavior because applications like RTSE rely on timely DATA messages.

Communication Mode Translation We introduce two types of translation behaviors that enable an adapter to translate between spontaneous and commanded mode, depending on whether the adapter is located at server/publisher side or client/subscriber side. Communication mode translation is not necessary when PMUs and PDCs are both operated in spontaneous mode.

Figure 4.14 illustrates the basic idea behind communication mode translation. In the first case, data come from a server (PMU or PDC) that uses commanded mode and is translated to spontaneous mode before being passed to a publisher to be carried over the pub/sub communication infrastructure (left side of Figure 4.14). In the second case, data come from the pub/sub communication infrastructure via a subscriber in spontaneous mode and are translated to commanded mode before being passed to a client that uses commanded mode (right side of Figure 4.14). Publisher adapters perform commanded-to-spontaneous mode translation; subscriber adapters perform spontaneous-to-commanded mode translation. We explain the translation operations inside both adapters.

Publisher Adapter A publisher adapter initiates data transmission with a server which may be a PMU or a sending PDC. It translates from commanded mode to spontaneous mode which is depicted in Figure 4.14. The adapter communicates with the server (PMU) in commanded mode. It requests a CFG-2 message from the server and then stores it internally, so that the configuration remains available. The streaming from the server to the adapter is enabled by a CMD message. The adapter starts sending the CFG-2 message spontaneously over the pub/sub communication infrastructure via a subscriber to the client (PDC) in user-defined intervals. The adapter forwards the incoming DATA messages to the pub/sub communication infrastructure.

Subscriber Adapter A subscriber adapter communicates with the client PDC in commanded mode, meaning that it listens for CMD messages from the client PDC, as depicted in Figure 4.14. It receives CFG-2 and DATA messages spontaneously from the pub/sub communication infrastructure via the subscriber,

and stores CFG-2 messages internally for future use. The client PDC requests a CFG-2 message from the adapter to correctly interpret the measurement data. Then, the client sends a CMD message to the adapter to turn on the data streaming, which is in fact a forwarding of DATA messages from the server. Finally, the client turns off the data streaming of the adapter in a similar manner. Without translation at the subscriber side, the receiving PDC is served exclusively and transparently in spontaneous mode.

Message Translation We propose to use data decapsulation and re-encapsulation as a straightforward solution for translating IEEE C37.118 messages to pub/sub data plane messages. The message translation step may be simplified if the adapters are integrated on the PMU/PDC platform.

Publisher Adapter A server (PMU or PDC) sends messages to a publisher adapter which strips off the TCP/UDP and IP headers to obtain the actual IEEE C37.118 messages. The publisher adapter re-encapsulates these messages in the data plane format of the underlying pub/sub architecture. Additionally, it may extract data fields from the original IEEE C37.118 messages and embed this information in the data plane message header, e.g., for in-network filtering if the pub/sub architecture supports this feature.

Subscriber Adapter On the receiving side, a subscriber adapter translates the pub/sub data plane message back to IEEE C37.118 before handing it over to the PDC. This is done by decapsulating the original IEEE C37.118 message from the data plane message, encapsulating it in TCP/UDP and IP, and re-sending it towards the PDC. Additionally, the adapter performs *stream demultiplexing* towards the PDC if the PDC cannot separate distinct PMU streams received over the same TCP/UDP socket, e.g., by assigning a unique source port number to each PMU stream.

Implementation Options for Publisher and Subscriber Adapters The adapter can be implemented (1) as an additional software module of the target PMU/PDC platform, or (2) as an extra logical and physical entity. We briefly discuss the advantages and disadvantages of both approaches. The advantage of approach (1) is that adapters integrated into the PMU/PDC platform do not add transmission delay, but its disadvantage is increased integration effort as coding on the PMU/PDC platform is required. It may be even difficult or impossible to add code on these platforms so that only the second approach may be feasible. The advantage of approach (2) is that PMU and PDC platforms do not need to be modified. Their configuration just has to be pointed to the adapters which take care of the necessary signaling and data forwarding. Its disadvantage is that the adapters introduce extra hops on the communication path which adds delay, e.g., on the station bus. If PMU and PDC manufacturers take care of the integration of IEEE C37.118 in pub/sub, they should follow the first approach which is feasible for them and does not cause additional communication delay.

Configuration Considerations The configuration of the synchrophasor network has a direct influence on the configuration of the adapters. In general, two cases have to be considered: the synchrophasor network is operated in spontaneous mode and the synchrophasor network is operated in commanded mode. We assume that it is best practice to operate synchrophasor networks using the same communication mode for all deployed devices, i.e., we omit discussing cases with PMUs and PDCs using different communication modes inside the same synchrophasor network. Independent of the communication mode of the synchrophasor network, PMU adapters and PDC adapters always have to be configured with the correct topic names and the correct pub/sub credentials.

Synchrophasor Network Operated in Spontaneous Mode PMUs have to be configured with the IP address and port number of the publisher adapter. The subscriber adapter has to be configured with the IP address and port number of the client PDC so that it can correctly forward data received over the pub/sub architecture towards that PDC. Publisher adapters and client PDCs do not need any further configuration.

Synchrophasor Network Operated in Commanded Mode Client PDCs have to be configured with the IP address, port number, and the IDCODEs of all PMU streams of interest of the subscriber adapter. Subscriber adapters only have to be configured to translate between spontaneous and commanded mode, i.e., they automatically learn the IDCODEs of all PMU streams for which they are responsible for by processing the data received from the pub/sub communication infrastructure, and they react on IEEE C37.118 communication from the PDC. Publisher adapters have to be configured with the IP address, port number, and IDCODEs of their respective PMU streams, and they have to be configured to translate between commanded and spontaneous mode, i.e., they need to initiate communication with the PMUs. PMUs do not need any further configuration.

Implementation in C-DAX We implemented the described publisher and subscriber adapters in the C-DAX prototype, and deployed our software on the Virtual Wall network testbed [121, 122], EPFL's network simulator (see Section 4.3.2.2), and Alliander's LiveLab [54] SG test site. Currently, the implementation supports spontaneous mode PMUs and PDCs over UDP only, performs stream demultiplexing at the subscriber adapter, and is realized as extra entities (see approach (2) in Section 4.2.4.2). In the Virtual Wall network testbed setup, we used our adapter implementation to connect four PMUs over the C-DAX communication architecture to one PDC. Additionally, we have a running proof-of-concept implementation of the communication mode translation, allowing to interconnect spontaneous and commanded mode PMUs and PDCs in any meaningful combination. The current implementation supports UDP as transport protocol only. We used the PMU Connection Tester [123] analysis software to verify the correctness of our implementation. The PMU Connection Tester supports IEEE C37.118 in commanded and spontaneous mode, and allows to investigate PMU communication. Further, we enhanced the iPDC [120] PDC/PMU simulator software with spontaneous mode to have a test target for our adapter, e.g., to generate spontaneous mode PMU data for testing and evaluating our adapter. We used the enhanced version of iPDC as a blueprint for our adapter during the early stage of our prototype development.

4.2.5 Inter-Domain Concept

We assume that in the future, all utilities will operate a communication middleware for their SG communication, be it C-DAX or any other comparable system. We further assume that utilities want to provide restricted access to their data for interested parties. Interested parties may be their own customers, energy market participants or other third-parties. In any case, the system needs to be able to support the concept of domains. In general, a domain is the set of components of the same jurisdiction. A utility may even cluster its infrastructure into several domains. Direct communication between clients and nodes of different domains shall be restricted for example due to business reasons, laws, operations rules, or security. To enable communication between these domains, the communication architecture needs to provide and support an inter-domain communication concept.

4.2.5.1 Actors

The inter-domain concept of C-DAX involves four actors: companies, DNs, external subscribers, and SecServs. Companies define C-DAX domains and want to exchange information. They define what information should be accessible for intra-domain communication only and what information should be accessible for inter-domain communication as well. DNs provide access for external subscribers to the domain's C-DAX cloud, and are the only point of contact for external subscribers. They further trigger authentication and authorization of external subscribers, manage external subscriptions, and eventually forward data from internal DBs to external subscribers. External subscribers are basically C-DAX clients that are associated with a different domain. They may re-publish the received topic data in their own domain, i.e., in their own C-DAX cloud. The SecServs of each domain are responsible for managing the respective access rights.

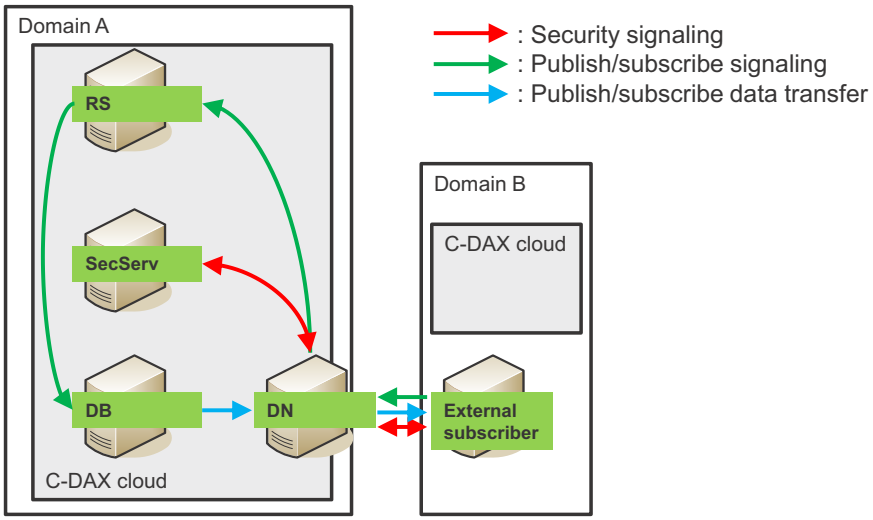


Figure 4.15: *The inter-domain concept in C-DAX allows each C-DAX domain to operate its own infrastructure while enabling access for and to other domains. C-DAX DNs forward inter-domain traffic, and enforce domain-based security policies.*

4.2.5.2 Basic Interactions

The inter-domain concept of C-DAX permits only reading access of topic data of each domain. Write access of external publishers is explicitly forbidden by design which avoids potential security threats. We therefore describe basic signaling if an external subscriber from domain B wants to retrieve topic data from the C-DAX cloud of domain A.

In general, the join of an external subscriber follows the same steps as the join of an internal subscriber. The main difference is that the SecServ has to consult a different internal information base for external clients to properly authenticate and authorize the joining subscriber. The actual subscription of the external subscriber is managed at the DN facing the external subscriber. The DB involved in the topic data dissemination is only aware of that DN but not of any external subscribers. The basic interactions are illustrated in Figure 4.15.

4.2.5.3 Security Considerations

The inter-domain security concept is built on top of the DNs and the respective SecServs of each domain. The DNs hide the domain's internal C-DAX cloud and network structure. Furthermore, external subscribers may only access the domain's C-DAX cloud through the DNs. The SecServs are responsible for managing the access rights to the C-DAX cloud (authentication) and the actual access rights to the topic data (authorization). That means, the domain owners have to agree on an access relationship between their domains on a per-topic basis and configure their SecServs accordingly to allow inter-domain communication.

4.2.5.4 Implementation Remarks

Even though we strongly believe in the usefulness of the inter-domain concept for SG communication middlewares in general, we assigned to the actual implementation of the inter-domain concept in the C-DAX prototype a low priority. Both laboratory tests and field trial of C-DAX did not require inter-domain communication and in the end, the consortium decided to not implement this feature in the final prototype of the C-DAX middleware.

4.3 Proof of Concept

To validate the design, and to evaluate the baseline communication and management functionalities of C-DAX, we implemented a detailed simulation of C-DAX in the OMNeT++ framework. In addition to the detailed simulation, we also implemented a C-DAX prototype that was deployed and evaluated in three different environments.

4.3.1 OMNeT++ Simulation

We provide an overview over the implementation of C-DAX in OMNeT++. OMNeT++ [124, 125] is a C++ based discrete event simulator which can be used for modeling computer networks. The INET framework [126] works on top of OMNeT++ and implements several protocols which are needed for modeling IP networks, e.g., IPv4, IPv6, Ethernet, TCP, UDP, and routing protocols. The C-DAX simulation framework is built on top of the INET framework.

4.3.1.1 Components

For the functional simulation of C-DAX, we implemented all important C-DAX components.

PublisherApp: A client application producing data. It periodically sends topic data and topic join request messages to the DN.

SubscriberApp: A client application consuming data. It periodically sends topic join request messages and receives topic data messages from the DN.

DesignatedNodeApp: Represents the DN as described in Section 4.1.2. The DesignatedNodeApp forwards control plane messages received from clients to the SecurityServerApp and to RSes via RDS. The DN forwards topic data messages received from publishers to DBs and topic data messages received from DBs to subscribers.

DataBrokerApp: Represents the DB as described in Section 4.1.2. The DataBrokerApp receives topic data messages from publisher DNs and forwards them to subscriber DNs.

RDS: Represents the RDS as described in Section 4.1.2. The RDS maintains topic-to-RS mappings configured by the ManagementSystem. Requests received from DNs are forwarded to the responsible RSes. Details of possible RDS instantiations are left open to the implementer of the C-DAX architecture, i.e., it has not been fully specified in [26, 29]. The simulation framework uses a simple server application for an RDS node which can be placed either on hosts placed at any location in the network or collocated on same hosts as DNs.

ResolverApp: Represents the RS as described in Section 4.1.2. The ResolverApp maintains topic-to-DB mappings. The ResolverApp receives control plane messages sent by DNs via the RDS and sends control plane messages to the DB responsible for the respective topic.

SecurityServerApp: Represents the SecServ as described in Section 4.1.2.

ManagementSystem: Represents the MgmSys described in Section 4.1.2. The module controls the creation and removal of topics based on an eXtensible Markup Language (XML) configuration file, and accesses the affected components through direct method invocation.

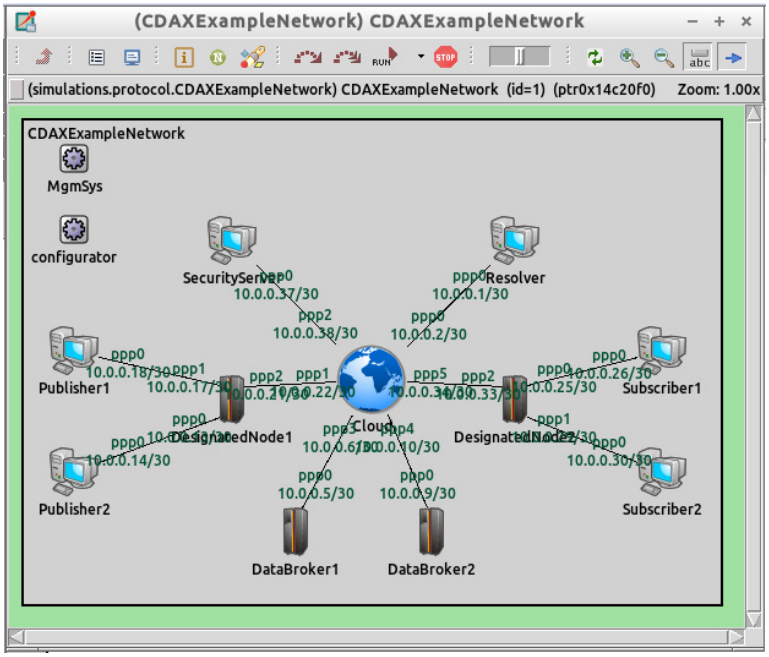


Figure 4.16: *Simulation setup for the protocol simulation of the C-DAX architecture shown in the OMNeT++ simulation GUI.*

4.3.1.2 Protocol Simulation

To verify the design and functionality of the C-DAX architecture, we evaluated several test cases. We used the simplistic network model shown in Figure 4.16 to make the visualization of the protocol simulation in the OMNeT++ graphical user interface (GUI) more comprehensible. It consists of two publishers, two subscribers, two DNs, two DBs, one RS, one SecServ, and one MgmSys. All components are connected via an IP-router. Each DN host is configured with a RDS application. The MgmSys configures all RDS running hosts automatically with the same knowledge about topic-to-RS mappings.

We verified that our implementation works under normal conditions, i.e., publishers and subscribers can join the C-DAX cloud and reliable disseminate topic data. In additional scenarios, we showed that C-DAX continues to work when different DBs fail or clients crash. Finally, we demonstrated that corner cases are handled properly by the forwarding engine, e.g., last subscriber leaves.

4.3.2 Prototype Implementation and Field Trial

After we verified that C-DAX works in the simulator, we implemented the basic C-DAX functionality in a distributed C++ program to enable further tests and verification in a real world environment. For this purpose, we deployed the C-DAX prototype in three different environments throughout its development.

4.3.2.1 iMinds' Virtual Wall

We first deployed the C-DAX prototype on the Virtual Wall network testbed [121, 122]. The latter was provided by iMinds, a C-DAX consortium partner. The main objective of this deployment was to evaluate basic signaling, security, resilience, and IEEE C37.118 protocol support. The C-DAX consoritum chose RTSE as application to be run on top of this deployment. At this time, we had no access to the PMUs developed by EPFL and National Instruments (NI), i.e., we could only stream pre-recorded PMU data over C-DAX. However, this still allowed us to demonstrate that data can be disseminated reliably over C-DAX. A byproduct of this deployment is the JReplayClient, a small Java program that enables parallel replay of several pre-recorded PMU data streams over UDP. This handy software has since been actively used by EPFL during their PDC and RTSE development. The Virtual Wall deployment was also part of the first C-DAX project review.

4.3.2.2 EPFL's Network Simulator

We then deployed the next iteration of the C-DAX prototype on EPFL's network simulator, a hardware power network simulator used during the PMU, PDC, and RTSE development. The main objective of this deployment was to evaluate IEEE C37.118 protocol support with existing PMU hardware, multiple subscriber support including filtering, and resilience [35, 37]. In contrast to the Virtual Wall deployment, we now had physical access to the PMU prototypes (shown in Figure 4.17). The PMUs were directly connected to the power network simulator and had C-DAX clients running to publish their measurement data. The PMUs should stream their data over C-DAX to two PDC/RTSE instances, each interested in a subset of the original data. PMUs, C-DAX nodes and PDCs were connected via an Ethernet switch. This setup allowed us to demonstrate that the IEEE C37.118

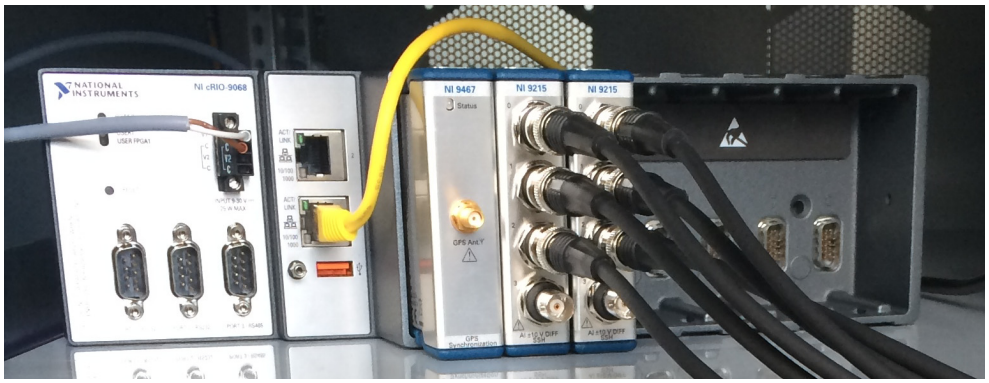


Figure 4.17: *The developed PMU prototype, based on a NI CompactRIO platform (NI 9068) including the GPS receiver (NI 9467) and two ADCs (NI-9215) that sample the input voltage and current waveforms. Taken from [36].*

protocol support works as expected without inducing any measurable delay. Furthermore, the setup allowed us to demonstrate the fast switchover of C-DAX' resilience mechanism by physically unplugging a DB from the Ethernet switch to mimic a DB failure. We used per-subscriber filtering to stream all PMU data to a primary RTSE instance showing the current state of the (simulated) power network, and to stream a subset of all PMU data to a secondary RTSE instance showing advanced state estimation features such as fault detection and fault location. This deployment was part of the second C-DAX project review. A video of the C-DAX prototype live demo is available on YouTube [37].

4.3.2.3 Alliander's Livelab

Finally, we deployed the last iteration of the C-DAX prototype as part of the project's field trial in Alliander's LiveLab [54] SG test site, a DG in the Bommelerwaard region in the Netherlands [3, 36]. The main objective of this deployment was to evaluate C-DAX and RTSE in a real DG, and to evaluate the tunnel adapter concept. 10 PMUs and 1 PQ meter were installed in selected substations alongside the BML 2.10 feeder as shown in Figure 4.18. The substations were chosen by EPFL after a preceding power network observability analysis.

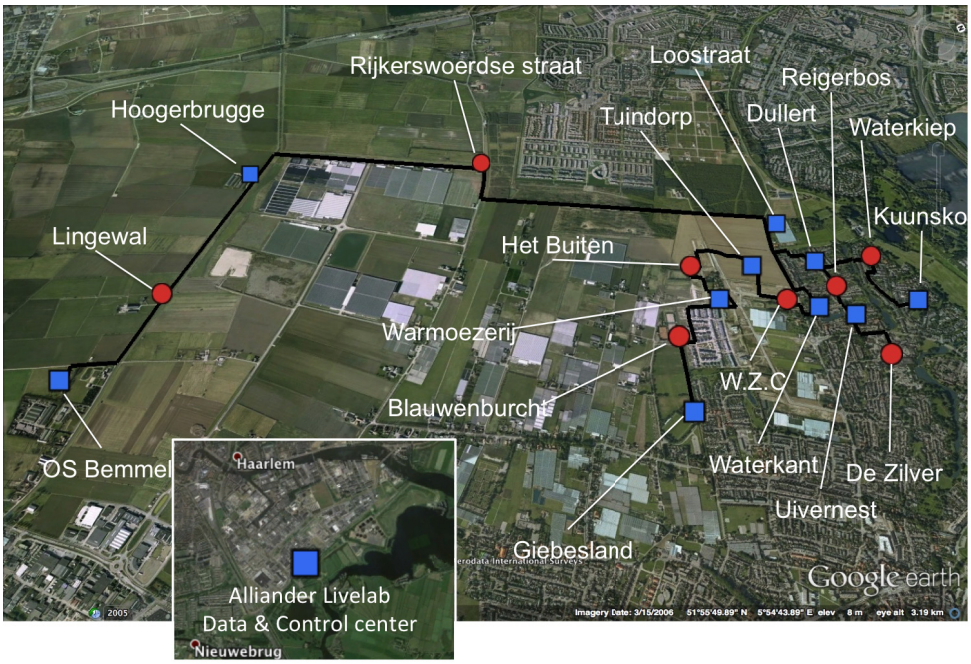


Figure 4.18: *The BML 2.10 feeder topology (the substation names are indicated) and Alliander's Data center in Haarlem. Substations drawn as rectangles have a PMU installed. Taken from [36].*

Figure 4.19 shows the overall field trial setup on the map of the Netherlands. The C-DAX cloud components had to run in Alliander's high-security data center in Haarlem, Netherlands, due to the company's strict standard operation procedures. That means, the actual field trial was about 110 km away from the C-DAX cloud as shown in Figure 4.19. Each PMU and the PQ meter was connected to a LTE mobile access gateway for Internet connectivity via the Vodafone mobile network as shown in Figure 4.20. To the best of our knowledge, this is the first time ever that PMU-based RTSE has been deployed and demonstrated on the DG level. Furthermore, our setup showed that SG communication over LTE is, in fact, practical. Our tunnel adapter allowed to utilize the deployed PQ meter, even though it was not used for any RTSE-related operations. This deployment was part of the final C-DAX project review.

Figure 4.19: *Map of the Netherlands that highlights the physical location of the BML 2.10 feeder and the Alliander data center where the PDC has been deployed (approximate distance: 110km). Taken from [36].*

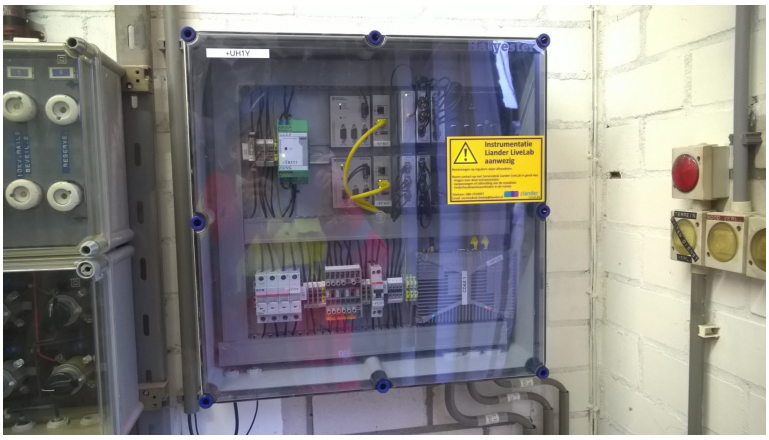


Figure 4.20: *The CompactRIOs (PMU (upper) and PQ (lower)) with LTE mobile access gateway installed in a substation. Taken from [36].*

4.4 Strength and Weakness Analysis of C-DAX

Besides the development of the novel C-DAX architecture, we continuously analyzed the strengths and weaknesses of the C-DAX architecture with respect to alternative communication solutions. The original goal was to conduct a quantitative comparison, but we decided to perform a qualitative comparison of C-DAX with other approaches instead. We justify our decision for this deviation from the original work plan before presenting the actual strength and weakness analysis. Eventually, recommendations for the potential re-use of C-DAX concepts and components in other architectures will be given.

4.4.1 Qualitative vs. Quantitative Comparison

The main objective of this subsection is to clearly understand and point out the advantages and drawbacks of the C-DAX approach. C-DAX is as flexible as all other compared systems and more flexible than the C-DAX blueprint architecture SeDAX. Even though we originally planned to perform a quantitative comparison of C-DAX with alternative approaches, we decided to do a qualitative comparison instead. The main reasoning behind that decision is that our own C-DAX architecture is efficient by design whereas the C-DAX blueprint architecture SeDAX came with a complex geographic hashing-based resource management system. Achieving a comparable level of flexibility and efficiency with SeDAX would have required significant changes to the core architecture, as has been shown in Chapter 3. Furthermore, the C-DAX software is still in prototype state whereas most of the other systems have matured over years of development and professional use. A quantitative comparison with regard to system stability or memory footprint would, therefore, give those probably less-advanced systems an advantage over our prototype software.

4.4.2 Comparison Metrics

Our qualitative comparison focuses on common and unique selling functionalities of each system in contrast to the respective functionalities of the C-DAX architecture. During our literature study, we identified a few criteria (comparison metrics) to differentiate the investigated communication architectures from the C-DAX architecture. We introduce each of these criteria and briefly explain their respective meaning.

4.4.2.1 Security

Security is a very important feature that communication architectures should provide when they are used for SG communication. Depending on the actual use case, different levels of security need to be supported. In our comparison, we distinguish between (1) no security, (2) transport-layer security, and (3) end-to-end security. Figure 4.21 illustrates the three levels of security. No security is obvious (see topmost illustration in Figure 4.21). Transport-layer security means that the communication between two nodes of the system is secured on the transport layer using well-established protocols like transport layer security (TLS) or datagram transport layer security (DTLS), or a similar technique. Transport-layer security does not guarantee that a third-party may be able to read the transferred data when an intermediate forwarding node of the system has been taken over (see middle illustration in Figure 4.21). In contrast, end-to-end security provides this level of security. That means, end-to-end security guarantees that only legitimate receivers of data are able to actually read and interpret the transferred data. All intermediate nodes may only be able to validate the integrity of transferred and forwarded data but cannot make any assumptions about the actual content (see lowermost illustration in Figure 4.21).

4.4.2.2 Resilience

SG communication involves mission-critical communication like for example fault protection, SCADA, or RTSE. To provide the necessary level of availabil-

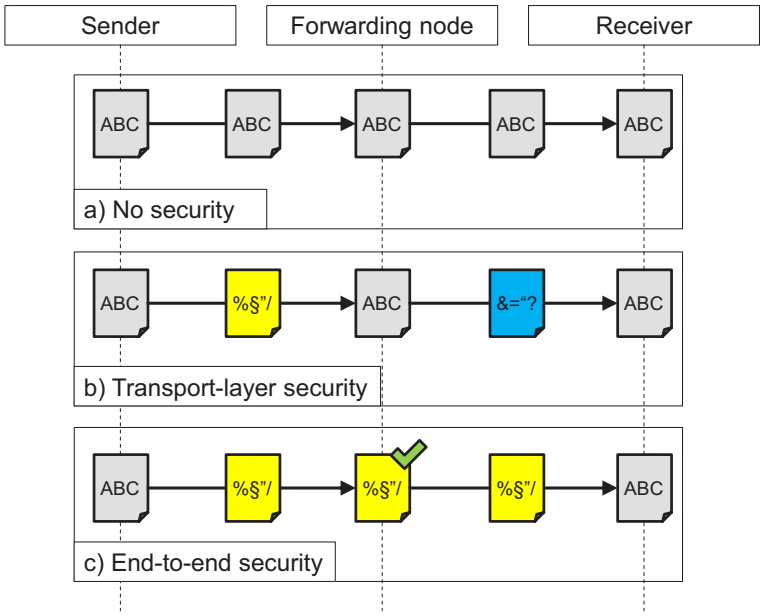


Figure 4.21: *Classification of the three security levels a) no security, b) transport-layer security, and c) end-to-end security. The green checkmark represents a message integrity check at an intermediate forwarding node.*

ity to the respective SG application running on top of the communication architecture, it has to be robust against network and hardware failures. Forwarding nodes may fail or become unavailable for several reasons including planned service cycles, network links may be impaired or break due to construction works or natural disasters. Either way, the communication architecture should provide mechanisms for resilient data forwarding should any component fail.

Besides the resilience mechanisms, the actual message delivery semantics [116] are also of interest in a comparison. We will, therefore, include information on message delivery semantics where appropriate. Possible message delivery semantics are (1) at most once, (2) exactly once, and (3) at least once. At most once message delivery means that messages are transmitted only once and no retransmissions are triggered should they get lost on the way to receivers, i.e., messages are either received successfully or not at all. Exactly once message de-

livery means that messages are transmitted and possibly retransmitted until they are received exactly once at receivers, i.e., messages are always transferred reliably. At least once message delivery means that messages are sent several times by senders or replicated by intermediate nodes, and receivers are responsible for handling duplicate messages, i.e., message are received successfully at least once.

4.4.2.3 Message Persistence

Message persistence is a feature of a communication architecture that is sometimes mistaken as resilience. Message persistence means that a communication system stores all sent messages between senders and receivers for future use. Examples for such future use are archival purposes, failure recovery of receivers, or support for query mode. The meaning of archival purposes is obvious. Failure recovery of receivers means that a re-started message receiver may actively request all old messages that have been lost due to its failure. Similarly, the query mode allows requesting specific data from the system based on query criteria, e.g., topic name or topic attributes. In ICN systems, message persistence is often called in-network storage and in-network caching.

4.4.2.4 Communication Modes

All investigated communication systems fall into the category of pub/sub, ICN, or message-queuing, i.e., publishers/content providers send data over the communication system to subscribers/content consumers. Depending on how the actual data transmission is realized, we differentiate between (1) broker-based communication and (2) broker-less communication. Broker-based communication means that at least one intermediate node between publisher and subscriber is involved in the data transmission, called a broker. While this approach provides great scalability with regard to the number of supportable publishers and subscribers, and the necessary network bandwidth between the communication parties, it also has its drawbacks like for example increased end-to-end delay (additional hops due to application layer forwarding), and vulnerability against intermediate node fail-

ures (which needs to be dealt with by applying appropriate resilience mechanisms). In contrast, broker-less communication allows for the lowest physically achievable end-to-end delay given an appropriately provisioned underlying physical network. However, limited scalability with regard to the maximum number of reasonably supportable communication parties is a limiting factor for broker-less communication because each communicating pair of nodes needs dedicated resources, i.e., a direct connection between publisher and subscriber. An ideal communication middleware for SGs provides both communication modes, configurable for each topic.

4.4.2.5 Inter-Domain Communication

We assume that in the future, all utilities will operate a communication middleware for their SG communication, be it C-DAX or any other comparable system. We further assume that utilities want to provide restricted access to their data to their own customers and selected access to their data to energy market participants or third-parties. In any case, the system needs to be able to support domains. In general, a domain is the set of components of the same jurisdiction. A utility may even cluster its infrastructure into several domains. Direct communication between clients and nodes of different domains shall be restricted for example due to business reasons, laws, operations rules, or security. To enable communication between these domains, the communication architecture(s) need to provide and support an inter-domain communication concept.

4.4.3 Comparison of C-DAX with Alternative Approaches

We now compare the C-DAX architecture with alternative communication architectures from academia and industry. They all share the basic pub/sub, ICN or message-queuing communication principle as common ground and allow for a similarly flexible resource management as C-DAX with the exception of SeDAX.

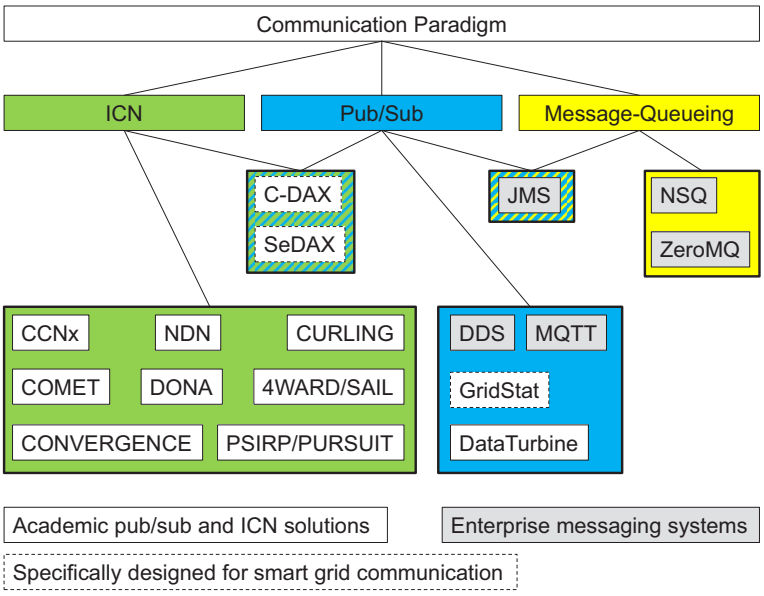


Figure 4.22: *Classification of C-DAX and alternative communication solutions by communication paradigm, academic or industrial background, and their specificity for SG communication.*

Figure 4.22 gives a broad overview on all investigated architectures, a classification by communication paradigm (indicated by different background colors of the grouping rectangles), a classification by academic or industrial background (indicated by the different background colors of the individual rectangles), and their specificity for SG communication (indicated by the line style of the individual rectangles).

In the following, we will use the classification by academic or industrial background for better comprehension and cover the relatively large set of ICN architectures (see the green rectangle in the lower left part of Figure 4.22) separately.

4.4.3.1 Academic Publish/Subscribe and ICN Solutions

SeDAX The SeDAX [39] architecture is an ICN and pub/sub system that uses geographical routing on a DT overlay network to forward messages to the responsible broker. Geographic hashing assigns static overlay network coordinates

to topics. Storage requirements for resilient SeDAX operation have been investigated in [7], and an extension to support distributed load balancing has been proposed in [9]. The REMP [52] of SeDAX is purely based on symmetric encryption. It uses long-term keys for each participating overlay node, which are assigned during node authentication. The actual end-to-end communication between publishers and subscribers in SeDAX is protected using encryption with diversified keys derived from the long-term node keys. Topic data is stored on adjacent primary and backup brokers which are closest and second-closest to the topic's coordinate to make the system resilient against node failures. After a node failure, the backup broker takes over automatically, and the overlay reconfigures itself to restore the overlay DT properties and heal the forwarding. Messages in SeDAX can be stored on the nodes for query-based retrieval. SeDAX provides only broker-based communication. A direct communication mode is not available by design. Inter-domain communication is not available in SeDAX.

DataTurbine The DataTurbine [127] pub/sub architecture proposes a Ring Buffer Network Bus (RBNB) comprised of either a single broker or a federated set of brokers. DataTurbine is format agnostic, i.e., messages are treated as binary data. The target application domain of DataTurbine is transmission of environmental sensor data. DataTurbine supports transport security using TLS. End-to-end security is not available on the middleware level. DataTurbine implements exactly-once delivery but allows for rate adaptation through so-called monitor subscriptions should downstream bandwidth exceed. That means, if subscribers are only interested in receiving most recent topic data and not necessarily all data, they make a monitor subscription. Brokers can be mirrored to make the system resilient against network failures. A ring buffer is used to store the most recent messages in memory and on disk. This enables applications to pause, rewind, or replay streams. DataTurbine only supports broker-based communication. A topic in DataTurbine corresponds to a single publisher, i.e., the native communication paradigm is one-to-many. The available literature did not indicate if inter-domain communication is available in DataTurbine.

GridStat GridStat [128] is a broker-based pub/sub middleware for wide area monitoring systems and was one of the candidate architectures for NASPInet.. It is still subject to ongoing research. The GridStat control plane consists of a hierarchical set of Quality of Service (QoS) brokers. The data brokers in GridStat are called *Status Routers*. Neither an open source nor a commercial version is publicly available at this time. GridStat proposes an end-to-end security scheme based on symmetric encryption and multicast authentication [129] [130]. The latter means that a data stream can be verified even if packets are dropped or reordered. GridStat subscribers can specify QoS requirements for subscription. As part of the QoS requirements a subscriber can request a specific level of redundancy, so that data is being forwarded over multiple disjoint network paths. GridStat is focused on delivery of real-time sensor data. To achieve very low latencies, GridStat does not implement any message persistence. Messages that cannot be forwarded immediately, are dropped by the brokers [131]. GridStat only supports broker-based communication. A broker-less communication mode is not available. GridStat allows creating a hierarchy of QoS brokers on the management plane. However, it is not clear from the available literature to what extent this enables inter-domain communication.

4.4.3.2 Enterprise Messaging Systems

Object Management Group Data Distribution Service The Object Management Group (OMG) Data-Distribution Service (DDS) [132] is a pub/sub architecture which targets real-time communication. DDS uses the Real-Time Publish/Subscribe protocol (RTPS) which can be described as reliable multicast. Common implementations of DDS are RTI Connext DDS, OpenSplice, and OpenDDS. Some DDS implementation support transport encryption based on the DTLS protocol. Future versions of the DDS specification will include a pluggable security architecture. Features like authentication, access control, and cryptography will be implementable as plugins. DDS proposes the concept of data-stream ownership [133] to provide fault tolerance and automatic failover. That means, publishers and subscribers communicate over so-called data-streams

which have owners with pre-configured ownership-strength assigned. In case of node failures, the next-strongest data-stream owner takes over. Message persistence is available in DDS and can be configured using a fine-grained set of QoS policies. Direct point-to-point connections between publishers and subscribers yield minimal delay and latency, i.e., no brokers are involved in the communication. The QoS policies further allow to install filters alongside the subscriptions so that only messages matching a filter are delivered to subscribers, e.g., `TIME_BASED_FILTER` defines a minimum time interval between two message deliveries. The latter facilitates rate-adaptation in DDS. Besides QoS policies, DDS supports so-called *Content Filtered Topics* which are topics with filtering properties. It makes it possible to subscribe to topics and at the same time specify that subscribers are only interested in a subset of the topic's data [134]. The topic space of a DDS installation constitutes its dataspace which is comparable to a C-DAX domain. DDS allows partitioning the topic space into DDS domains which are under the same jurisdiction as opposed to C-DAX domains. A client can be member of multiple domains without the need for specific inter-domain mechanisms. The DDS dataspace Interconnection Service (DDS-IS) [135] extends DDS with an inter-domain mechanism to interconnect the dataspaces of different DDS installations. The latter resembles the inter-domain concept of C-DAX.

Java Message Service Java Message Service (JMS) [136] is an API for message-oriented middleware specified as part of the Java Enterprise Edition Platform. JMS can be used with a single broker, a "master/slave" configuration or a "network of brokers". Common implementations of JMS providers are Apache ActiveMQ, IBM WebsphereMQ, Oracle Glassfish, and TIBCO EMS. JMS implementations support transport encryption using the TLS protocol. End-to-end security is not specified in the standard but existing JMS implementations provide this feature as payload encryption [137]. While failover mechanisms are not defined in the JMS API, common implementations provide high-availability schemes including failover for node failures. JMS provides mechanisms for de-

livery acknowledgements which are used to realize message persistence. JMS brokers cache messages until they received delivery acknowledgements from all active subscribers. Subscriptions can also be in an inactive state while a subscriber is disconnected. Messages for inactive subscriptions are stored until the inactive subscribers become active again. JMS provides mechanisms to install filters alongside the subscriptions so that only messages matching a filter are delivered to subscribers. JMS supports both point-to-point and pub/sub messaging modes. JMS-JMS-connectors enable exchanging data among different JMS instances, effectively representing inter-domain communication.

NSQ The distributed messaging platform NSQ [138] uses a broker-less architecture with topic discovery using a redundant set of resolvers. The knowledge of the resolvers is so-called eventually consistent, i.e., joining subscribers (consumers) have to query all their configured resolvers to find all responsible producers (publishers) for a certain topic. Each publisher forwards received topic data to all interested subscribers. NSQ supports transport security using TLS. A separate end-to-end security scheme is not required as NSQ does not forward data over intermediate nodes. NSQ implements "at-least-once" delivery. Subscribers have to handle duplicate message reception themselves. The architecture provides means for reliable data transfer using acknowledgments, and in-network caching. If a client does not send acknowledgements, messages are automatically retransmitted. Message persistence is only available for the retransmission of unacknowledged messages. NSQ only supports direct communication. Brokers are not supported, but broker-like components could be implemented using NSQ building blocks. The available literature did not indicate if inter-domain communication is available in NSQ.

ZeroMQ (∅MQ) The ZeroMQ [139, 140] high-performance asynchronous messaging library provides a message queue for scalable distributed and concurrent applications. Starting with version 4, ZeroMQ supports a CurveCP [141] based transport encryption scheme called CurveZMQ [142]. A separate end-

to-end security scheme is not required as ZeroMQ does not forward data over intermediate nodes. Although ZeroMQ does not provide resilience or failover mechanisms, it offers a framework to implement them. ZeroMQ can be used to build complex pub/sub architectures, using socket polling and heartbeating for reliable node failure detection, and primary-backup server pairs to provide high-availability. Message persistence is not natively supported by ZeroMQ but could be implemented using the ZeroMQ building blocks. ZeroMQ is a broker-less system, but instructions for implementation of brokers and are available. The available literature did not indicate if inter-domain communication is available in ZeroMQ.

MQTT The Message Queue Telemetry Transport (MQTT) [143] is a protocol for machine-to-machine (M2M) communication originally developed by IBM. MQTT is a light-weight, broker-based pub/sub architecture. MQTT supports transport security using TLS. End-to-end security is not available on the middleware level. Delivery semantics of MQTT can be configured using three QoS levels: "at most once", "at least once", and "exactly once". MQTT implementations [144] provide high availability schemes with clustered brokers. Messages persist on the brokers until they are acknowledged by all receivers. MQTT also supports persistent sessions. A subscription of a persistent session remains valid even when the subscriber disconnects. Messages that were sent while the subscriber was offline are resent after the client reconnects. MQTT only supports broker-based communication. A broker-less communication mode is not available. Filtering is supported through hierarchical topics and wildcards in topic subscriptions. The available literature did not indicate if inter-domain communication is available in MQTT.

4.4.3.3 Other ICN architectures

C-DAX has followed the ICN paradigm with the purpose of adopting the identified benefits of inherent anycast, multicast, mobility, caching and security support. A set of ICN architectures has been proposed during the last years, aiming at harnessing these benefits [98, 145–152]. These architectures have been proposed as alternatives to the current public Internet architecture and as such have not been

tailored specifically for SG applications, as is the case with C-DAX. Nevertheless, for completeness reasons in this section, we provide a high level comparison of the differences between C-DAX and these architectures, highlighting the distinctive features of C-DAX. Similar to the previous sections, in this comparison, we take into account the aspects of deployability, security, resilience, message persistence and inter-domain communications. It must be noted that a thorough description of these architectures is considered beyond the scope of this work; a detailed overview of the considered architectures can be found in [153].

Deployability The difficulties faced in deploying a new network architecture play a vital role in the adoption of the architecture in practice. Most ICN architectures so far follow clean slate designs, targeting a replacement of the current TCP/IP protocol stack. CCN/NDN assumes routing and forwarding being based on a content-oriented Forwarding Information Base (FIB) and a Pending Interest Table (PIT) denoting the network interface content requests have received from/forwarded to [98, 151, 152]. PSIRP/PURSUIT also uses a novel Bloom filter based routing and forwarding plane, replacing IP as well [145, 146]. A similar approach has been followed by the NetInf architecture [147, 148]. The CONVERGENCE architecture follows an approach similar to CCN/NDN with the exception of considering an off-path name resolution system to provide the name-to-location binding information [150]. In all these approaches, and contrary to the overlay approach of C-DAX, IP is replaced by its ICN counterpart. As an effect, the deployment of C-DAX is facilitated on top of existing network infrastructures, as is the case with the LTE network used in the field trial. In this respect, by following an overlay approach, C-DAX directly enables the adoption of the ICN paradigm benefits without heavily disrupting the existing networking infrastructure and practices. DONA [99] and CURLING [149, 154], on the other hand, maintain full backwards compatibility with IP. In effect, their deployability can be compared to that of C-DAX. However, their focus is on global, Internet-scale, content-centric communications. As such, they do not focus on highly important aspects such as security and resilience.

Security C-DAX gains the standard security benefit of most ICN (and pub/sub) solutions, namely the decoupling of clients and subscribers. Having no direct communication between clients and between clients and subscribers lessens the attack surface of nodes within C-DAX. In general, ICN solutions introduce more, or at least new security weaknesses, compared to traditional networking solutions [155]. However, most of these weaknesses stem from using ICN solutions to replace the current TCP/IP networking infrastructure. The more limited setting of SGs for which C-DAX is designed allows us to have an encompassing security architecture, which deals with a closed user group which greatly simplifies the key setup and registration. The setting of SGs also introduces some unique security challenges, e.g., in public smart charging of electrical vehicles (EVs), where the sudden availability of charge details introduces a new incentive for attacks. C-DAX can be a good way of securing the various information flows in such new architectures, as we argued in [156].

There have been several surveys comparing the security of ICN solutions [155, 157] which note that it is important in an ICN to secure the data, not the connections. Additionally, end-to-end security over untrusted clouds is seen as an important feature mostly found in recent academic proposals. C-DAX implements security on the data layer, not on the connections and achieves end-to-end security as a result. Furthermore, C-DAX has a highly configurable security architecture, with choices between symmetric or asymmetric cryptography, depending on the available resources on the clients. The most important security aspect that C-DAX offers lies in the resilience and availability guarantees provided by C-DAX. While traditionally thought of as a safety and correctness feature, these features also provide excellent security benefits.

Resilience Most ICN architectures proposed so far do not focus on resilience. This is justified by their focus on the novel, clean slate functional design aspects such as routing and forwarding, simply demonstrating a different focus area. However, C-DAX focuses on a realistic path towards the immediate adoption of the ICN principles and their straightforward application on the field, with a particular attention paid to SG application requirements, such as resilience. In

addition, C-DAX is able to exploit specific characteristics of power grids (e.g., grid topologies, device locations) to enhance the robustness of the communication infrastructure.

Message Persistence All ICN architectures consider content (rather than message) persistence in the form of in-network caching. An extensive volume of ICN research has been devoted to the design of efficient caching schemes [158]. Though it bears some similarity to message persistence, (in-network) caching in ICN is in principle different in that it presents an opportunistic character, i.e., no guarantees are provided for the availability of messages within the network, as caching mechanisms aim to adapt to the dynamic conditions in the network on a best effort basis. In contrast, C-DAX focuses on the specific application domain of mission-critical SGs, where content availability needs to be guaranteed, changing the design objectives of the architecture.

Communication Modes It can be argued that to a large extent all ICN architectures adopt the pub/sub paradigm in that in all cases, a recipient (subscriber) must express explicit interest to a named content item before a sender (publisher) can start transmitting the requested item. In this respect, the functionality of brokers in pub/sub systems corresponds to the name resolution functionality in ICN. Then, considering broker-based or broker-less operation in ICN becomes equivalent to whether name resolution takes place on the data path or not. In CCN/NDN, name resolution is performed by the routers themselves, i.e., no separate name resolution service is supported [98, 151, 152]. FIBs/PITs are populated in such a way that data always follows the shortest path to the recipients of the information. In contrast, PSIRP/PURSUIT decouples name resolution by realizing a separate Rendezvous (name resolution) System and a Topology Manager service whose role is to further support source routing [145, 146]. In particular, rendezvous nodes (i.e., broker nodes) are aware of both publishers and subscribers to a particular information item and are responsible for notifying the Topology Manager about this match. The Topology Manager is then responsible for constructing a special type of Bloom filter containing the identifiers of all network links to be followed by the corresponding data packets. Publishers and routers

use this Bloom filter to forward data to the subscribers. In this process, no broker belongs to the data path, which equals to the union of the shortest paths from the publisher to each subscriber. A similar source routing scheme is used in Net-Inf, though without employing Bloom filters, but rather label stacks. In contrast, broker entities in C-DAX belong to the data path and a non-careful configuration/selection of the broker nodes may result in inefficient, stretched data paths. However, the simplicity of the SG network infrastructure (e.g., star topology in simple LTE environments), as well as the inherent ability to support multiple brokers per topic provide C-DAX operators with the tools to avoid inefficient routing. The broker-based mode of communication also enables the implicit construction of simplified multicast trees, where publishers need only to unicast their data to the broker(s) which is (are) then responsible of replicating it to the different subscribers. Focusing in principle on inter-domain communications, DONA relies on IP multicast for this operation on an intra-domain level [99]. CURLING, on the other hand, allows the configuration of forwarding devices (equivalent to brokers) on both an inter-domain and intra-domain level [149] supporting thus the formation of efficient data delivery structures.

Inter-Domain Communication The concept of inter-domain communications in C-DAX refers to the exchange of information between different administrative domains (e.g., DSOs) with a particular focus on access rights i.e., managing the rights publishers and subscribers residing at different administrative domains to publish or subscribe to information of another domain. So far, work on inter-domain communications in ICN architectures has only focused on enabling name resolution at a global, inter-domain level, i.e., resolving name identifiers at Internet scale. However, work in this area has mostly focused on the design of a scalable name resolution system, able to accommodate vast volumes of resolution information, e.g., 10^{13} distinct names to be resolved [159]. Moreover, limited efforts have been devoted in designing access control mechanisms for ICN architectures, e.g., [160]. Moreover, access control becomes more important in the C-DAX environment since applications may often be mission-critical, affecting the stability of the power grid itself.

4.4.4 Summary of Analysis

Table 4.5 summarizes the strength and weakness analysis of C-DAX in comparison with alternative communication solutions from academia and industry. We conclude that the C-DAX concept is competitive with existing pub/sub, ICN and message-queuing communication solutions. From a direct comparison of features, DDS and JMS could be alternatives to C-DAX but without a quantitative comparison of all three architectures under comparable operational conditions, we cannot make any general statement on what architecture is the most appropriate for C-DAX use cases. However, with regard to the special requirements for SG communication, C-DAX stands out as a potential blueprint for improving existing well-established communication architectures. Especially the security architecture, the dual communication mode (broker-based and broker-less), and the adapter concept could be ported and re-used in alternative architectures.

4.5 Lessons Learned

The objective of this chapter was to describe and evaluate the C-DAX architecture. We first discussed the reasons that eventually led to the design of the novel C-DAX architecture. We summarized and justified the enhancements and changes during the transition from the project's blueprint architecture SeDAX to C-DAX. In a next step, we gave a broad overview on the final C-DAX architecture, its design rationales, components, basic interactions, and its advanced features.

We detailed core features of the C-DAX architecture. We extended the initial C-DAX specification with four resilience support levels: no resilience (RSL-0), with packet loss during switchover (RSL-1), with packet delay but without packet loss during switchover (RSL-2), and without packet loss and delay during switchover (RSL-3). RSL-0 and RSL-2 are implemented in the C-DAX prototype, and we presented measurement data of the switchover process for RSL-2.

Table 4.5: *Summary of the strength and weakness analysis of C-DAX in comparison with alternative communication solutions. Adapted from [34].*

Architecture	Security		Resilience	Message persistence	Comm. mode		Advanced features		
	Transport	End-to-end			Broker-based	Broker-less	Rate adaptation	Filtering	Inter-domain
C-DAX	X	X	X	X	X	X	X	X	X
SeDAX	?	X	X	X	X	-	-	X	-
DataTurbine	X	-	(X)[a]	X	X	-	X	-	-
GridStat	?	X	X	-	X	-	X	?	-
DDS	(X)[b]	(X)[c]	X	X	-	X	X	X	X
JMS	X	X[d]	X[d]	X	X	X	-[e]	X	X
NSQ	X	N/A	X	(X)[f]	-	X	?	?	-
ZeroMQ	X	N/A	-	-	-	X	?	?	-
MQTT	X	-	X	X	X	-	?	X	?
CCN/NDN	?	(X)	-	X	X	-	?	X	(X)
PSIRP/PURSUIT	?	(X)	-	X	-	X	?	X	(X)
NetInf	?	(X)	-	X	-	X	?	X	(X)
CONVERGENCE	?	(X)	-	X	?	?	?	X	(X)
DONA	?	(X)	-	X	?	X	?	X	X
CURLING/CURLING	?	(X)	-	X	X	?	?	X	X

X: available; (X): partially available; -: not available;

N/A: not applicable; ?: unclear from literature.

[a] Resilience mechanism covers only link failures

[b] Not part of the DDS specification

[c] Planned feature for future versions of DDS

[d] Not part of the JMS specification

[e] Some JMS implementations have work-arounds to handle slow information consumers that would otherwise slow down or block information producers but those work-arounds cannot be seen as rate adaptation

[f] Only applicable to retransmissions

We introduced two advanced communication modes for C-DAX to better cope with the needs of SG applications: broker-less pub/sub mode and transparent IP-tunneling mode. The broker-less pub/sub mode clearly improves the end-to-end delay for real-time applications because no intermediary application layer hops are involved in the data transmission from publishers to subscribers. The transparent IP-tunneling mode enables legacy SG applications to communicate transparently over C-DAX utilizing advanced features such as end-to-end security, resiliency, and flexibility. The transparent IP-tunneling mode is implemented in the C-DAX prototype and was used alongside the general streaming mode in the C-DAX field trial.

We proposed a security architecture for the C-DAX middleware. We defined the security properties *source authentication*, *topic access control*, *end-to-end integrity*, and *end-to-end confidentiality* for C-DAX and presented the mechanisms used to enforce them. We described how keys are initially distributed and how they are updated either in regular intervals or as a response to topic joins and leaves. A subset of this architecture has been implemented in the C-DAX prototype and is used in the C-DAX field trial to securely exchange phasor measurement data.

We proposed a solution to easily connect and integrate entities in an IEEE C37.118 synchrophasor network over a pub/sub communication infrastructure. We introduced publisher and subscriber adapters as interfaces for entities of the synchrophasor network with the pub/sub architecture. The adapters translate between IEEE C37.118 commanded and spontaneous mode which is necessary as commanded mode requires a back channel that is unavailable in pub/sub communication. They also translate the message format so that the data can be forwarded over the pub/sub communication infrastructure. We explained these procedures in detail, discussed implementation options, and clarified configuration issues. Our proposed method allows transparent integration of all IEEE C37.118-compliant hardware and software in pub/sub architectures. The adapter concept is implemented in the C-DAX prototype as a standalone solution and is used in the C-DAX field trial.

We verified the design with an OMNeT++ simulation of the communication architecture that can be used for systems research. We also implemented a proof-of-concept of C-DAX that was deployed and evaluated on iMinds' Virtual Wall network testbed [12], EPFL's power network simulator [35, 37], and Alliander's LiveLab [54] SG test site [3,36]. The deployment on Alliander's LiveLab SG test site also represents the C-DAX field trial. To the best of our knowledge, this is the first time ever that PMU-based RTSE has been deployed and demonstrated on the DG level. Furthermore, our setup showed that SG communication over LTE is, in fact, practical.

We conclude that the C-DAX concept is competitive with existing pub/sub, ICN and message-queuing communication solutions, in particular also with the feature-rich DDS and JMS architectures. With regard to the special requirements for SG communication, the security architecture, the dual communication mode (broker-based and broker-less) and the adapter concept of C-DAX could be ported and re-used for improving existing well-established communication architectures.

5 Use Case Study: Future Retail Energy Market

In the previous chapter, we presented the C-DAX architecture, which enables cyber-secure, scalable, and resilient communication for SG communication. We used C-DAX use case (UC)1 (*Telecontrol*, see Section 2.5.1) and UC2 (*Synchrophasor-Based Real-Time State Estimation of Active Distribution Networks*, see Section 2.5.2) as a vehicle to illustrate the necessity and functionality of some of its core features. This chapter summarizes our investigations in the context of UC3 (*Future Retail Energy Market*).

In the future retail energy market (REM), any participant will be able to trade energy. As a consequence, the future REM for electrical energy will have many more participants and see more volatile prices than today, creating the need for new communication and trading infrastructures [43–45]. The PowerMatcher (PM) is a multi-agent based approach for such a trading infrastructure which enables market integration of DERs and automatic DSM. While the trading side of the framework is well understood, there is no study that considers the communication side. We review PM and analytically evaluate its communication characteristics.

Besides a trading infrastructure, advanced metering infrastructures (AMIs) in the DG are necessary as an enabling technology to provide automatic billing, and acquisition of network status data. Different standards and communication protocols exist for smart metering, ranging from transmission protocols to architectural recommendations. We present the concept of the German AMI as defined in BSI TR-03109 [46], review implementations of smart metering protocols and

architectures, and provide a Java-based open-source smart meter gateway experimentation framework (jOSEF).

The content of this chapter is mainly taken from [8, 11, 14] and structured as follows. Section 5.1 recaps the future REM and discusses related work. In Section 5.2, we briefly review PM and analytically evaluate its communication characteristics. In Section 5.3, we propose jOSEF that combines and extends established protocol frameworks to provide an extensible tool for the validation of smart metering communication. The lessons learned are summarized in Section 5.4.

5.1 Background and Related Work

In this section, we recap today's and future REM [8]. In addition, we discuss related work in the area of SG communication traffic estimation and characterization, and review relevant protocols for smart metering.

5.1.1 Recap of Today's and Future Retail Energy Market

Electrical power generation is currently changing from a centralized system with predictable and controllable outputs to a system integrating DERs including weather-dependent renewables. Such renewable energy sources are hard to predict and impossible to control [69,161]. As a direct consequence, electrical power distribution networks are undergoing major changes in operational procedures and monitoring, thereby evolving from passive to active networks [57, 162]. The downside is that we will face variations in supply, with periods of higher or lower renewable energy offers. The deficit must be compensated by other, probably more expensive energy sources to avoid outages. This will affect future markets for electrical energy, e.g., future prices for electrical energy will fluctuate more than today.

Nevertheless, a normal household will still be able to buy electrical energy for a fixed price per period from a retailer, but at increased cost. Consumers may

be better off buying power directly from prosumers or DERs than from retailers on the REM, thus taking advantage of lower prices at certain times, and possibly shifting parts of their demand to other times of day, which is a desired behavior [68]. Today, DERs like PV panels or wind farms sell their generated power for a fixed, subsidized price. When the fixed-price contract model expires, they may sell their energy on the REM, too. As a consequence, the *future REM* for electrical energy will have many more participants and see more volatile prices than today, creating the need for new communication and trading infrastructures [43–45]. In a later section of this chapter, we will review the PowerMatcher (PM) as a possible approach for such a trading infrastructure, and analytically evaluate its communication characteristics.

5.1.2 Related Work

This subsection details related work on traffic estimation of SG communication, and gives a brief overview on protocols for smart metering.

5.1.2.1 Traffic Estimation of Smart Grid Communication

Budka et al. discuss SG bandwidth requirements in LTE macrocells in [163]. They estimate the worst case bandwidth requirements of different SG applications, e.g., SCADA, synchrophasors, closed-circuit television (CCTV), mobile workforce, and AMI. They apply these estimates to different LTE deployment scenarios and evaluate them with and without meter concentrators placed at substations. They conclude that the frequency spectrum has a direct impact on the bandwidth requirement, which is caused by the LTE cell size. They further claim that bandwidth requirements for smart grid applications may not exceed 5 MB/s per investigated applications per LTE macrocell.

Karagiannis et al. [164] investigate the suitability of LTE for SG communication as well. In contrast to [163], they focus on the established manufacturing message specification (MMS) framework of the IEC 61850 SG protocol suite [56] as communication protocol. Using an NS-3-based simulation

model [165], they examine whether MMS over LTE can satisfy the performance requirements for smart metering and remote control communications, and propose architectural modifications. The performance evaluation shows that LTE can be used for the investigated applications as underlying communication technology, given those modifications are applied.

Kansal et al. [166] investigate bandwidth and latency requirements for synchrophasor measurements on the transmission grid level. They propose an evaluation framework based on the NS-2 simulator [167], and apply their tool on the Polish power system to evaluate the communication requirements for different zones inside that system. They conclude that the average link bandwidth for the investigated SG application should be in the range of $5 - 10$ Mb/s within one zone, and in the range of $25 - 75$ Mb/s for inter-zone communication. They further claim that 100 ms latency requirements can be achieved when utilities use a meshed topology for communication.

Deconinck [168] analyzes data volumes and real-time requirements for advanced metering with focus on the two-way property of the communication. He investigates the applicability of powerline communications, smallband and broadband communication over telephone line or cable, 2G and 3G mobile telephone systems, and other radio technologies for advanded metering in the Flanders region of Belgium. He compares those access technologies regarding costs, reachability, bandwidth, latency and reliabilty, and concludes that hybrid communication solutions are needed to satisfy all requirements.

In [169], Luan et al. describe a bottom-up method for SG communication network capacity planning. They estimate hourly traffic profiles based on message sizes and intervals for metering, monitoring and telecontrol applications. Based on the traffic profiles and the forecasted number of devices, they derive regional bandwidth requirements for a *blue sky day* scenario featuring normal operation conditions and a *storm day* scenario including large power outages.

5.1.2.2 Protocols for Smart Metering

The DLMS/COSEM suite is a set of standards for the exchange of energy meter data, comprising of Device Language Message Specification (DLMS) [170] as an application layer protocol for communication with metering devices, and COmpanion Standard for Energy Metering (COSEM) [171] as a system for object-oriented modeling of energy metering equipment. DLMS/COSEM uses the OBject Identification System (OBIS) [172] to identify data objects in energy metering systems, and *COSEM services* enable clients to query specific attributes of objects, assign values to attributes of objects, or execute methods of objects.

The smart message language (SML) [173] is a message-oriented protocol for communication with SMs. The SML application protocol defines *SML files* consisting of one or multiple *SML messages*. An *SML message* can be either a request or a response. SMs act as servers, receiving SML files from clients, and processing the contained SML messages in order of reception. Starting with version 1.04, SML supports COSEM services, i.e., the COSEM object model can be used with the SML application protocol. Currently, SML is not widely used outside Germany but international use is expected to increase if plans to adopt SML as part of the DLMS/COSEM suite [174] are successful.

M-Bus is a protocol suite for communication with SMs. M-Bus is defined in the European standard EN 13757 which comprises data model [175], application layer [176], and both wired [177] and wireless [178] specifications for the physical layer. The Open Metering System (OMS) [179, 180] is a smart metering communication architecture based on M-Bus. OMS proposes several modifications to the M-Bus protocols, and adds an optional authentication and fragmentation layer to the M-Bus protocol stack.

The Dutch smart meter requirements (DSMR) [181] are a joint specification of the Dutch grid operators. DSMR is based on DLMS/COSEM and M-Bus, and defines a data model for SMs including corresponding OBIS codes. We use selected parts of DSMR to fill the technical gaps of the German BSI TR-03109 for our smart metering experimentation framework.

5.2 Performance Evaluation of the PowerMatcher Application

In this section, we give a broad overview on the PowerMatcher (PM) architecture, its general idea, components, and basic interactions. The PM communication framework [182, 183] developed by the Netherlands Organisation for Applied Scientific Research (TNO) aims at providing a communication and trading infrastructure at DSO scale, i.e., in the order of millions of customers.

The trading aspects of PM are well understood and have been evaluated in simulation studies and field tests [38, 184–186]. The communication side of PM has only been investigated with regard to latency measurements in a simulation study [38] demonstrating the scalability of PM for one million households. Investigations of the communication part beyond latency measurements do not exist.

This section addresses that gap through an analytical performance evaluation of the communication part of PM based on a realistic DG model provided by Alliander N.V. and TNO. We omit trading-related details because they are not within the scope of this work; for further information see [38, 184–186].

5.2.1 System Description

PM aims at (1) automatically balancing demand and supply in a cluster of DERs, and (2) market integration of DERs. It builds on a hierarchical multi-agent based approach. Within a PM cluster, agents are organized into a logical tree where DERs represent leafs and a so-called Auctioneer Agent (AA) forms the root.

There are two generic *agent roles* in PM as shown in Figure 5.1: agent and matcher. An *agent* expresses bids to its *matcher* based on the flexibility in supply and demand it represents. The matcher determines the price for its agents based on the supply and demands bids. Any agent is associated with exactly one matcher, and any matcher may be associated with any number of agents.

Besides the generic agent roles, the PM architecture comprises four *agent types*: Device Agents (DAs), one AA, Concentrator Agents (CAs), and option-

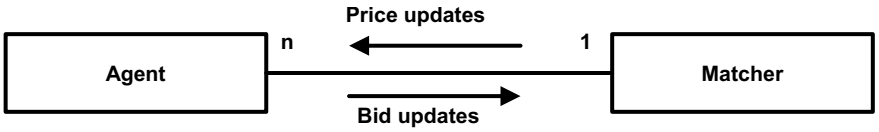


Figure 5.1: *Generic agent roles in PM: agent and matcher. Agents express bids to a matcher based on the flexibility in supply and demand they represent. The matcher determines the price for its agents based on the supply and demands bids.*

ally one Objective Agent (OA). Figure 5.2 gives an overview of a possible PM architecture and the respective interactions.

5.2.1.1 Components and Interactions

A *Device Agent* (DA) represents a DER device in the PM cluster. It is a control agent which tries to operate the associated physical device in an economically optimal way. An example for such a device may be a PV panel or controllable consumers, e.g., a fridge and a washing machine. The agent coordinates its actions with all other agents in the cluster by buying or selling energy consumed or produced by the device on an electricity market.

The *Auctioneer Agent* (AA) is the central entity that performs the price-forming process. It concentrates the bids of all DAs, CAs, and the OA directly connected to it in a single bid, searches for the equilibrium price and communicates a price update back whenever there is a significant price change.

A *Concentrator Agent* (CA) represents a sub-cluster of DAs or CAs. It concentrates the bids of all subordinate agents in a single bid and communicates this *aggregated bid* to the AA or to its superordinate CA if it is an intermediate CA. In the opposite direction, it disseminates price updates to the agents in its sub-cluster. A CA may perform bid and price transformation, i.e., intermediate agents can be configured with constraints causing localized price changes. An example would be cutting off the maximum power running over a certain node in the DG by increasing the price. PM calls this feature *congestion management*.

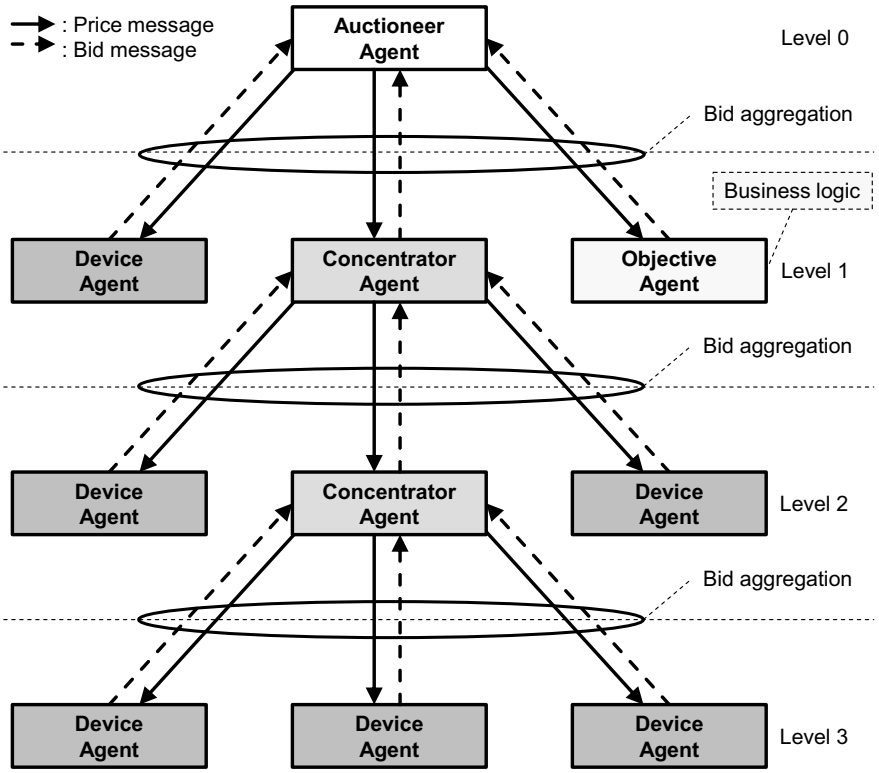


Figure 5.2: *Overview of the PM architecture and respective interactions. A hierarchy of CAs disseminates current price information from a central AA down to DAs. It further aggregates bids of DAs towards the AA. The behavior of the overall system can be influenced by an OA.*

The *Objective Agent* (OA) is an optional agent which allows to change the goal of the cluster. The default goal of the cluster is to balance the demand and supply automatically. If an OA is present, the goal of a cluster might be different, e.g., operation of the cluster as a virtual power plant (VPP). This agent interfaces with the business logic of the specific application for the cluster.

Signaling of all interactions is based on two message primitives: bid and price. A *price message* contains the minimum price, maximum price and the number of possible price points n_{steps}. The original price update message which is disseminated from the AA to next-lower agents is also called *market base message*. A

bid message contains a so-called *bidcurve* which is a vector of bids sampled according to the predefined settings received by the price message, i.e., the bidcurve comprises n_{steps} price points with each price point having a value between the minimum and the maximum price. At each CA, these bidcurves are aggregated to one bidcurve, and the AA uses these curves to perform its price-forming process.

5.2.1.2 Mapping to Publish/Subscribe Communication

The current implementation of PM runs over the MQTT message bus middleware [143], a broker-based light-weight pub/sub architecture which runs on top of TCP/IP. Mapping PM communication to pub/sub communication is straightforward, and we use Figure 5.2 as an example.

Figure 5.2 shows three levels of bid and price aggregation. The AA is located on the top level and acts as publisher for the price topic and as subscriber for the bid topic. Each level of aggregation shall only contain bids and prices of the respective level. To realize this using pub/sub communication, each CA needs to have its own independent set of bid and price topics for its subordinate CAs and DAs. We apply this to the given example in Figure 5.2, use the abbreviations for PM participants given in Table 5.1, and summarize the necessary topics and their corresponding publishers and subscribers in Table 5.2.

The process of mapping PM communication to pub/sub communication can be formalized. We denote n_{bid} as the number of bid topics, n_{price} as the number of price topics, and n_{topics} as the overall number of required topics. We further denote n_{AA} as the number of AAs, and n_{CA} as the number of CAs. One can derive n_{topics}, n_{bid}, and n_{price} based on n_{AA} and n_{CA} using the following formula.

$$n_{topics} = (n_{bid} + n_{price}) = 2 \cdot (n_{AA} + n_{CA}) = 6 \tag{5.1}$$

Applied to our example given in Figure 5.2, we have $n_{AA} = 1$ and $n_{CA} = 2$. Thus, we calculate $n_{topics} = 6$ distinct topics. This is in line with the previously conducted manual mapping of PM communication to pub/sub communication.

Table 5.1: *Abbreviations for the PM participants in Figure 5.2.*

Short Form	Long Form
AA	Auctioneer Agent
OA	Objective Agent
CA-x-y	Concentrator Agent $x - y$; x gives the level of the agent, e.g., 1, 2, or 3; y gives the horizontal position of the agent, e.g., left (L), middle (M), or right (R)
DA-x-y	Device Agent $x - y$; x gives the level of the agent, e.g., 1, 2, or 3; y gives the horizontal position of the agent, e.g., left (L), middle (M), or right (R)

Table 5.2: *Overview on topics necessary to map PM communication to pub/sub communication in Figure 5.2.*

Topic	Publisher	Subscriber
Auct_Bid	DA-1-L, CA-1-M, OA	AA
Auct_Price	AA	DA-1-L, CA-1-M, OA
CA-1_Bid	DA-2-L, CA-2-M, DA-2-R	CA-1-M
CA-1_Price	CA-1-M	DA-2-L, CA-2-M, DA-2-R
CA-2_Bid	DA-3-L, DA-3-M, DA-3-R	CA-2-M
CA-2_Price	CA-2-M	DA-3-L, DA-3-M, DA-3-R

5.2.2 Traffic Model

We now investigate the performance of PM communication. We base our studies on a DG model provided by the Dutch utility Alliander N.V. and TNO. We first give a brief description of the model, define our metrics, and analyze the model.

The model comprises of 2 million households or prosumers. Each household is represented by a CA, and has internally between 1 and 20 DAs. Two million

Table 5.3: *Key parameters for the performance evaluation of the investigated PM scenario.*

Variable	Value	Description
$n_{households}$	2000000	Number of households
$n_{clusters}$	20	Number of clusters
$n_{households}^{cluster}$	100000	Households per cluster
$n_{AA}^{cluster}$	1	Number of AAs per cluster
$n_{OA}^{cluster}$	1	Number of OAs per cluster
n_{CA}^{1}	$100 - 1000$	Number of CAs on level 1
n_{CA}^{2}	$1000 - 100$	Number of CAs on level 2 per CA on level 1
$n_{CA}^{cluster}$	$n_{CA}^{1} + n_{CA}^{1} \cdot n_{CA}^{2}$	Overall number of CAs per cluster
n_{DA}	$1 - 20$	Number of DAs per CA on level 2
s_{price}	16 B	Size of a price message
s_{bid}	2 kB	Size of a bid message
f_{price}^{min}	$\frac{1}{5} \cdot \frac{1}{\text{min}}$	Minimum price update rate

households are typically subdivided into 20 to 50 trusted clusters, each containing an AA. That means, there are at most $n_{households}^{cluster} = \frac{2000000}{20} = 100000$ households per trusted cluster. Within each cluster, there are either 100 concentrators active, each concentrating 1000 households, or 1000 concentrators active, each concentrating 100 households. Computing power and network bandwidth are limiting factors for the size of each concentrator's subcluster.

The communication behavior of the model is as follows. The AA sends out a price update at least every 5 minutes. Each price message is 16 B large. Each bid and aggregated bid message is 2 kB large. Each CA and DA reacts immediately on the price update and may reply with a bid message also at least every 5 minutes. MQTT is used as communication middleware between the DAs at the households and the AA of each cluster. Table 5.3 summarizes the important key parameters for the performance evaluation of the investigated scenario.

5.2.3 Performance Metrics and Analysis

We derive and calculate performance metrics per cluster first, and scale it to full model size later. For each cluster, the number of publishers and subscribers that need to be supported can be calculated by counting the number of involved AAs, OAs, CAs and DAs. Each of them acts as publisher and subscriber, i.e., the number of publishers and subscribers is equal because the communication in PM follows a bidirectional pattern. We base our calculations for $n^{min}_{subscriber}$ on the minimum number of DAs per household $n^{min}_{DA} = 1$, and the minimum number of CAs per cluster $min(n^{cluster}_{CA})$ which means $n^{1}_{CA} = 100$ and $n^{2}_{CA} = 1000$. The calculations for $n^{max}_{subscriber}$ are based on the respective maximum values $n^{max}_{DA} = 20$, and $n^{1}_{CA} = 1000$ and $n^{2}_{CA} = 100$.

$$\begin{aligned} n^{min}_{subscriber} =& n^{min}_{publisher} \\ =& n_{AA} + n^{cluster}_{OA} + min(n^{cluster}_{CA}) + n^{min}_{DA} = 200102 \qquad (5.2) \\ n^{max}_{subscriber} =& n^{max}_{publisher} \\ =& n_{AA} + n^{cluster}_{OA} + max(n^{cluster}_{CA}) + n^{max}_{DA} = 2101002 \qquad (5.3) \end{aligned}$$

This gives a lower and upper bound for the number of publishers and subscribers that need to be supported if each household has between 1 and 20 DAs running. For the remainder of this performance evaluation, we use $max(n^{cluster}_{CA})$ for the number of CAs per cluster. We derive the number of topics which have to be supported based on Equation (5.1).

$$n_{topics} = (n_{bid} + n_{price}) = 2 \cdot \left(1 + max(n^{cluster}_{CA})\right) = 202002 \qquad (5.4)$$

We estimate the minimum data rate that each agent is expected to handle. The AA receives bids only from its subordinate CAs and the OA, and sends price updates to these nodes. Therefore, the expected minimum load for the AA is derived as follows.

$$
\begin{aligned}
L_{sent}^{AA} =& s_{price} \cdot f_{price}^{min} = 0.43\,\text{b/s} \\
L_{recv}^{AA} =& (n_{OA}^{cluster} + n_{CA}^{1}) \cdot s_{bid} \cdot f_{price}^{min} = 54.67\,\text{kb/s} \\
L_{AA} =& max\left(L_{sent}^{AA}, L_{recv}^{AA}\right) = 54.67\,\text{kb/s}
\end{aligned} \tag{5.5}
$$

Intermediate CAs receive price updates from the AA, bids from their subordinate CAs or DAs. In the other direction, intermediate CAs send the aggregated bid to the AA and forward the price update to all subordinate CAs and DAs. For each intermediate CA, the expected minimum load $L_{CA}^{intermediate}$ is derived as follows.

$$
\begin{aligned}
L_{CA}^{intermed.,sent} &= (s_{price} + s_{bid}) \cdot f_{price}^{min} = 55.04\,\text{b/s} \\
L_{CA}^{intermed.,recv} &= \left(n_{CA}^{2} \cdot s_{bid} + n_{AA}^{cluster} \cdot s_{price}\right) \cdot f_{price}^{min} = 5.46\,\text{kb/s} \\
L_{CA}^{intermediate} =& max\left(L_{CA}^{intermed.,sent}, L_{CA}^{intermed.,recv}\right) = 5.46\,\text{kb/s}
\end{aligned} \tag{5.6}
$$

Household CAs receive price updates from their superordinate CA and bids from their subordinate DAs. In the other direction, they send the aggregated bid to the superordinate CA and forward the price update to all subordinate DAs. For each household CA, the expected minimum load is derived as follows.

$$
\begin{aligned}
L_{CA}^{household,sent} &= (s_{price} + s_{bid}) \cdot f_{price}^{min} = L_{CA}^{intermed.,sent} = 55.04\,\text{b/s} \\
L_{CA}^{household,recv} &= (n_{DA} \cdot s_{bid} + s_{price}) \cdot f_{price}^{min} = 1.07\,\text{kb/s} \\
L_{CA}^{household} =& L_{CA}^{household,sent} + L_{CA}^{household,recv} = 1.12\,\text{kb/s}
\end{aligned} \tag{5.7}
$$

Finally, DAs receive price updates from their superordinate CA and send bids to it. For each DA, the expected minimum load is derived as follows because bid messages are larger than price messages.

$$
L_{DA} = s_{bid} \cdot f_{price}^{min} = 54.61\,\text{b/s} \tag{5.8}
$$

As shown above, the expected peak load for the AA is largest. When we scale the numbers to 2 million households, the required network I/O capacity per node and also the necessary network bandwidth remain in a manageable region so that it can be realized with off-the-shelf technology.

Another important metric is the expected data rate per topic because this has a direct impact on the provisioning of the MQTT brokers responsible for these topics. We derive minimum and maximum data rates per topic on the assumptions that bid topics in general are larger because bid messages are 128 times larger than price messages. Further, we consider the minimum and maximum number of agents which corresponds to 1 (DAs) and 1000 (CA) respectively. Based on that, we calculate the minimum and maximum expected data rate per topic as follows.

$$L_{topic}^{min} = s_{bid} \cdot f_{price}^{min} = 54.61\,\text{b/s} \tag{5.9}$$

$$L_{topic}^{max} = max(n_{CA}^{1}, n_{CA}^{2}) \cdot s_{bid} \cdot f_{price}^{min} = 54.61\,\text{kb/s} \tag{5.10}$$

These rates are lower bounds as the actual message rate may be higher than f_{price}^{min}. We take the maximum expected topic data rate and extrapolate the system-wide overall topic load which has to be managed by the MQTT brokers. This load has to be distributed appropriately among all MQTT brokers of the system.

$$L_{topic}^{all} = n_{topics} \cdot L_{topic}^{max} = 11.03\,\text{Gb/s} \tag{5.11}$$

We assume that MQTT brokers are able to process data with at least $L_{broker}^{throughput}$[1]. We can therefore express the minimum number of brokers needed to handle the overall topic data load as a function of $L_{broker}^{throughput}$.

$$n_{broker}^{min}(L_{broker}^{throughput}) = \left\lceil \frac{L_{topic}^{all}}{L_{broker}^{throughput}} \right\rceil \tag{5.12}$$

[1] This summarizes processing and network throughput, whichever dominates as limiting factor.

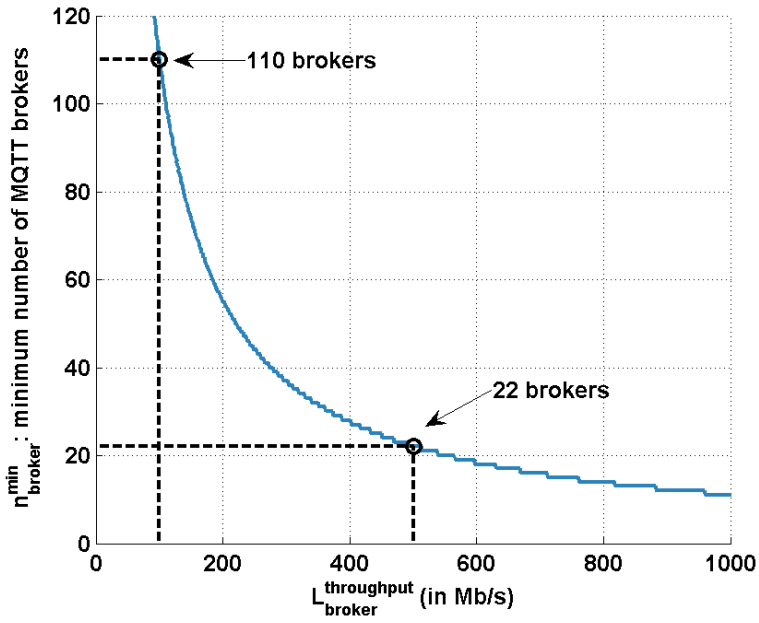


Figure 5.3: *The minimum number of MQTT brokers n^{min}_{broker} depends on the maximum data throughput of a broker $L^{throughput}_{broker}$. For a cluster with* 100.000 *households,* 22 *brokers with a throughput of* 500 *Mb/s are necessary to handle the overall topic data load. In contrast,* 110 *brokers would be necessary for $L^{throughput}_{broker} = 100$ Mb/s.*

Figure 5.3 shows the minimum number of brokers n^{min}_{broker} for varying broker throughputs $L^{throughput}_{broker}$ ranging from 100 Mb/s to 1000 Mb/s. The line is interpreted as follows: for a maximum broker throughput x on the x-axis, the y-axis gives the minimum number of brokers that are necessary to handle the overall topic load L^{all}_{topic} when the topic load is evenly distributed among all brokers. When brokers are able to process and transfer data with at least 500 Mb/s, a minimum number of 22 brokers is necessary for a cluster with 100.000 households. When brokers are significantly slower, e.g., $L^{throughput}_{broker} = 100$ Mb/s, a minimum number of 110 brokers is necessary for the same cluster.

CAs and the AA have to cache the last bids from all their subordinate CAs and DAs until a new bid is received. Because each agent shall send or resend its bid every 5 minutes, we propose a minimum caching time of 10 minutes for each bid.

$$t_{cache}^{min} = 10 \text{ minutes} \tag{5.13}$$

Based on this, we can estimate the minimum storage needed per agent. We denote the respective capacities as C_x^y with x representing the agent type and y the subclass, if applicable. We base our calculations on the assumption that $n_{CA}^1 = 1000$, $n_{CA}^2 = 100$, and $n_{DA} = 20$.

$$C_{AA} = (1 + n_{CA}^1) \cdot s_{bid} = 1.96\,\text{MB} \tag{5.14}$$

$$C_{CA}^{intermediate} = n_{CA}^2 \cdot s_{bid} = 0.20\,\text{MB} \tag{5.15}$$

$$C_{CA}^{household} = n_{DA} \cdot s_{bid} = 20.00\,\text{kB} \tag{5.16}$$

$$C_{DA} = s_{bid} = 2.00\,\text{kB} \tag{5.17}$$

These relatively small numbers for a base of 100.000 households is because each intermediate CA aggregates all bids into one bid, i.e., strong aggregation significantly reduces the amount of data which needs to be cached. When we scale these numbers to 2 million households, the AA still only has to have about 40 MB of storage for the last bids. The numbers scale linearly with the storage time should old bids be stored longer for statistical analysis, e.g., if is set to a very large value.

5.2.4 Numerical Results and Insights

We summarize the evaluation results in Figure 5.4. The shown results are valid for clusters of 100000 households, 1 AA, 1000 first-level CAs, 100 second-level CAs per first-level CA, and up to 20 DAs per second-level CA. When we scale our results for 2000000 households, we split the households into clusters of 100000 households each, i.e., 20 clusters in total. For each of those clusters,

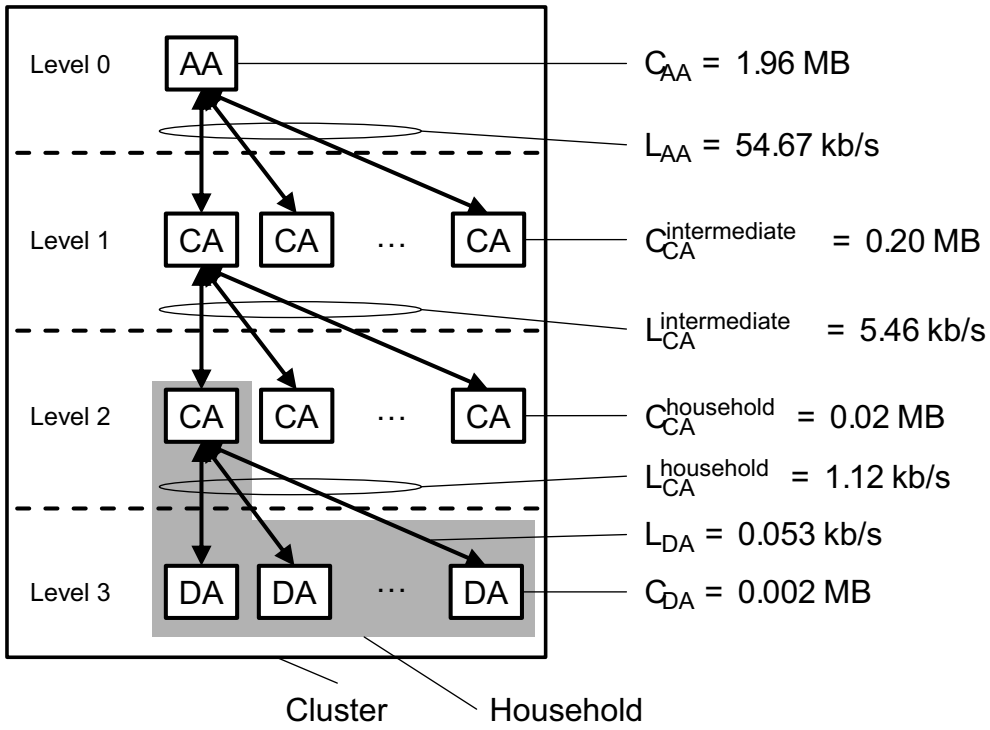


Figure 5.4: *Estimated loads and capacity requirements for PM clusters of* 100000 *households,* 1 *AA,* 1000 *first-level CAs,* 100 *second-level CAs per first-level CA, and* 20 *DAs per second-level CA.*

we effectively have the same load and storage capacity characteristics as shown in Figure 5.4. However, some values such as the number of publishers and subscribers, as well as the overall topic load do change with increasing numbers of households. We summarize the linearly scaled values for 2000000 households in Table 5.4 and compare them to those for 100000 households.

Our results show that the communication requirements of a large-scale PM deployment can be handled with todays communication technology. RETs with millions of participants possibly require only moderate resources on the communication's side. We identify two main reasons for the observed traffic characteristics. The first reason is price and bid aggregation at each intermediate agent which leads to a significant reduction in traffic volume. The second reason is the use of

Table 5.4: *Evaluation results for* 100000 *households (second column) and* 2000000 *households (third column).*

Variable	100000 **households**	2000000 **households**
$n^{min}_{subscriber}$	200102	4002040
$n^{max}_{subscriber}$	2101002	42020040
$n^{min}_{publisher}$	200102	4002040
$n^{max}_{publisher}$	2101002	42020040
n_{topics}	202002	4040040
L^{all}_{topic}	11.03 Gb/s	220.60 Gb/s
n^{min}_{broker}	22	440

pub/sub as information dissemination paradigm on the communication layer, i.e., each agent only has to publish one message to the pub/sub framework instead of sending separate messages to subordinate agents. The latter simplifies the agents internal communication logic.

5.3 A Java-Based Open-Source Smart Meter Gateway Experimentation Framework (jOSEF)

In the previous section, we investigated and evaluated the communication characteristics of the PM trading infrastructure. Regardless of whether PM or any similar trading architecture will be deployed for the future REM, *advanced metering infrastructures (AMIs)* in the DG are necessary as an enabling technology to provide automatic billing, and acquisition of network status data.

In this section, we present the concept of the German smart meter gateway (SMGW)-based AMI as defined in TR-03109 [46] of the Federal Office for Information Security (BSI). Different standards and communication protocols exist for smart metering, ranging from transmission protocols to architectural recommendations. While existing implementations allow the isolated simulation

and evaluation of certain smart metering communication aspects, a framework providing the minimally necessary building blocks for a BSI-03109-compliant SMGW-based architecture has been missing in both literature and in practice. We address this issue and introduce the *Java-based open-source smart meter gateway experimentation framework* (jOSEF). The framework allows to model an SMGW-based smart metering architecture utilizing open-source components only. We focus on the German AMI approach [46] but consider the Dutch AMI approach [181] for technical details that have not been defined for Germany yet.

5.3.1 Smart Meter Gateways: A Communication Topology for Smart Metering

SMGWs are the central communication components in the future smart metering infrastructure in Germany [46, 187]. The two most important functionalities of SMGWs are (1) gathering of metering data from SMs, and (2) providing a unified interface for metering data retrieval to interested and legitimate external market participants (EMPs).

In general, the SMGW mediates between three networks, as shown in Figure 5.5: the local metrological network (LMN), the home area network (HAN), and the wide area network (WAN). The LMN connects SMs to the SMGW only. The HAN connects end consumers, service technicians, and controllable local systems (CLSes) to the SMGW, e.g., EVs, PV panels, and remote-controllable heating and air conditions. The WAN connects administrators and EMPs to the SMGW, e.g., DSOs, metering point operators, and suppliers of electric energy.

5.3.1.1 Functionalities and Communication

The functionalities and the used communication protocols of SMGWs can be differentiated by the networks they mediate between.

In the LMN, SMGWs are responsible for gathering metering data from SMs according to metering profiles, time-stamping the measurements based on an externally synchronized time source, tariffing, and finally storing the time-stamped,

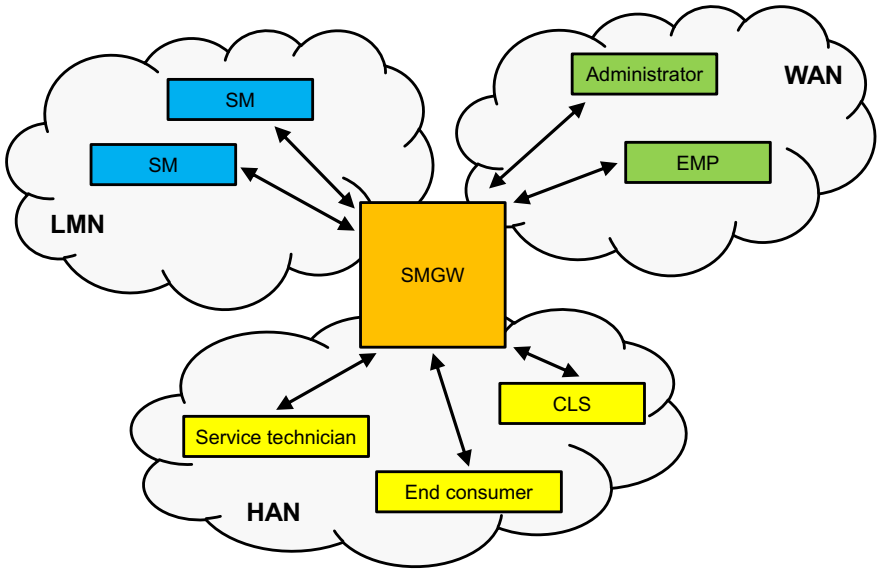


Figure 5.5: *System boundaries of the SMGW architecture according to [187]. The SMGW mediates between LMN, HAN, and WAN.*

tariffed metering data for further dissemination to EMPs. SMGWs support bidirectional and unidirectional communication with SMs. Bidirectional communication involves interactive communication between SMGWs and SMs to poll for metering data or to manage SMs. Unidirectional communication stands for unsolicited metering data dissemination from SMs to SMGWs. Generally, COSEM [171] with OBIS [172] codes are used as data model between SMs and SMGWs. Depending on the underlying physical layer, M-Bus [175–178] or SML [173] is used as transport protocol.

In the HAN, SMGWs provide read-only access to their internally stored metering data and status messages to end consumers. SMGWs can support several end consumers facilitating multi-client operation, e.g., in an environment involving many SMs and many households. Service technicians must only access status messages of SMGWs. SMGWs relay control messages between CLSes and EMPs as configured by administrators. [187] does not specify protocols between SMGWs and potential HAN communication partners but security mechanisms to

be used, e.g., secure transport layer communication, and mandatory authentication of clients against the SMGW. Essentially, any IP-based protocol may be used between SMGWs and HAN entities, e.g., end consumers or service technicians.

In the WAN, SMGWs are responsible for forwarding their internally stored metering data to interested and legitimate EMPs based on communication profiles. SMGWs must not accept connections from the WAN for security reasons but a wake-up service facilitates remote SMGW administration. When SMGWs receive specific control packets from the WAN, they contact an external administrator for maintenance, e.g., for firmware updates, changes in the communication profiles, time synchronization, or access to status messages. WAN communication is based on representational state transfer (REST) web services as defined in [187], and SMGWs act as REST web service clients because they must not accept connections from the WAN. EMPs must provide the server side of a REST web service according to the interface definitions in [187,188]. As for LMN communication, COSEM with OBIS codes are used as data model between SMGWs and EMPs but XML and cryptographic message syntax (CMS) [189] are used as transport protocol on top of REST. Time synchronization of SMGWs is handled over the network time protocol (NTP) instead of web services.

5.3.1.2 Security

The BSI SMGW protection profile (SMGW-PP) [190] requires all LMN, HAN and WAN communication to be secured by TLS in combination with a public-key infrastructure (PKI) [191, 192]. WAN communication is further protected by CMS between SMGWs and EMPs. SMGWs are equipped with a security module which provides cryptographic functions, e.g., generation and secure storage of encryption keys, and verification of digital certificates. The security module is realized as a smart card. Further information on the security module and its requirements can be found in [193–195].

These security requirements limit the suitability of the C-DAX middleware for smart metering in Germany because C-DAX provides its own strong security mechanisms [13] but does not support TLS between communication part-

ners without modification. However, if the BSI security regulations would permit replacing TLS by other security mechanisms with the same level of security, C-DAX may be used as communication middleware for HAN communication, e.g., between SMGWs and EMPs. In that case, C-DAX' pub/sub mechanisms would allow scalable, secure, and resilient dissemination of tariff information or firmware updates to all SMGWs, or transparent and secure remote control of a customers CLSes.

5.3.2 Existing Implementations

In this subsection, we review selected open-source implementations and discuss their suitability for the development of a BSI TR-03109 compliant SMGW.

OpenMUC [196] is an open-source implementation of a multi utility communication controller (MUC) developed at Fraunhofer ISE. OpenMUC is implemented in Java and licensed under the terms of the GNU General Public License (GPL). The core component of OpenMUC is the data manager which interfaces to optional components like data server, logger, protocol drivers, and custom applications. The OpenMUC framework provides the functionality specified in a previous draft standard for a German AMI. While BSI TR-03109 requires the use of XML for the REST web service, OpenMUC uses the JavaScript Object Notation (JSON) format. Additionally, the uniform resource identifier (URI) hierarchy used by OpenMUC differs from the BSI specification, and the protocol drivers do not satisfy the minimum requirements. Implementing a BSI TR-03109 compliant SMGW based on OpenMUC would require major changes to the OpenMUC code. However, the OpenMUC framework also includes protocol libraries that can be used independently, e.g., jDLMS and jSML. We discuss those in the following subsections.

jDLMS [197] is a Java implementation of the DLMS/COSEM protocol available under the terms of the GNU Lesser General Public License (LGPL). jDLMS supports the DLMS/COSEM application layer protocol over serial lines using high-level data link control (HDLC), or over TCP or UDP. As the current version

0.9.0 only implements the client side of the DLMS/COSEM protocol, and does not include the COSEM object model, jDLMS is not suitable for implementing a SMGW according to BSI TR-03109.

jSML [198] is a Java implementation of SML available under the terms of the LGPL. jSML supports SML communication over TCP/IP or over serial line. HDLC support is currently unavailable. jSML implements the SML message format and encoding, the SML data types, and the SML transport layer version 1. The current version 1.0.17 is based on SML 1.03, i.e., jSML does not yet support COSEM services. For implementing a SMGW according to BSI TR-03109, jSML needs to be extended to support the current SML version.

Open Gateway Energy Management (OGEMA) [199] is an OpenMUC-based software platform for building automation and load management. OGEMA is implemented in Java and licensed under the terms of the GPL. Since OGEMA is based on OpenMUC and focuses rather on the HAN side than on the SMGW, it is not suitable for implementing a SMGW according to BSI TR-03109.

Gurux [200] is a collection of smart metering software components developed by Gurux Ltd. Gurux code contains implementations in C#, C++, Java, and Delphi and is licensed under the terms of the GPL. Gurux supports DLMS/COSEM, Modbus, and M-Bus. However, the COSEM object model is not separated from the DLMS/COSEM application protocol in the Gurux code. Using the COSEM object model with other application protocols as required by BSI TR-03109, would require major modifications to the code.

5.3.3 Experimentation Framework

We now present jOSEF, a Java-based open-source smart meter gateway experimentation framework licensed under the terms of the GPL version 2 or later. We describe its architecture, specify its operation, and discuss the deviation from BSI TR-03109. The implementation utilizes the jSML library [198] for LMN communication that was extended as part of this work to support SML version 1.04 [173]. Additionally, we used the COSEM implementation of Gurux [200]

as a blueprint to implement DSMR's COSEM object model. We used DSMR's COSEM object model because BSI TR-03109 has not defined a companion standard for its COSEM object model yet.

5.3.3.1 Components

The framework comprises three main components: a minimal SMGW, an SM simulator, and a simple EMP. The *minimal SMGW* represents an SMGW which provides the minimally necessary functionalities to control SMs and to send meter data to EMPs. It is equipped with a GUI for configuration and operation, and allows several SMs to be connected, as shown in Figure 5.6. The *SM simulator* represents an SM and can be configured with standard load profiles for energy generation and consumption to simulate SM behavior. It is controlled over a command line interface (CLI), as shown in Figure 5.7. The *simple EMP* provides an BSI TR-03109-compliant REST web service towards the SMGW and acts as a data sink for meter data. It can be accessed using any hypertext transfer protocol (HTTP) client, e.g., a web browser.

5.3.3.2 Meter Data Retrieval

When the SMGW wants to retrieve meter data from an SM, it first sends an SML message to the SM requesting all internal COSEM object IDs to discover the SM's internal data model. The returned list of COSEM object IDs is then used by the SMGW to build the actual meter data retrieval request by filtering for metering object IDs based on OBIS codes. The SMGW generates a new SML message containing explicit requests for details on the metering object IDs, and sends the message to the SM. The SM returns the actual metering objects to the SMGW that can perform further processing on the data, e.g., time stamping, tariffing, buffering, or dissemination to EMPs. The SMGW re-sends the meter data request message to the SM to receive new meter data; rediscovery of the SM's internal data model by the SMGW is only necessary when the SM configuration changes.

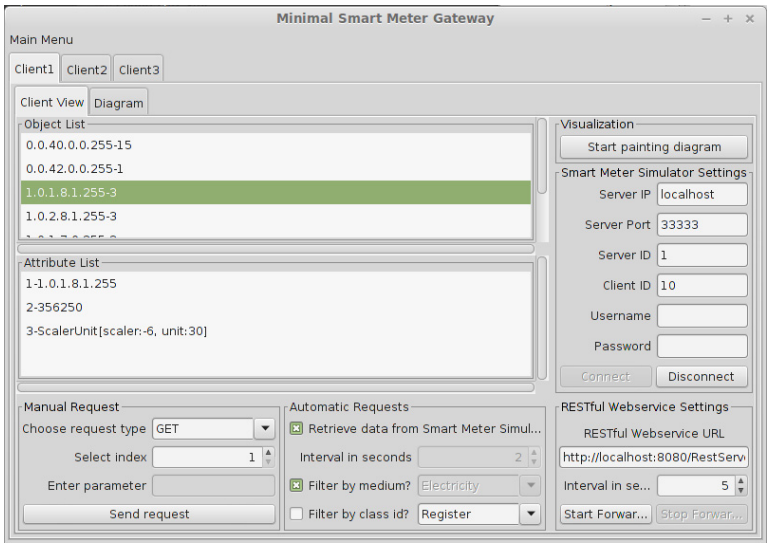


Figure 5.6: *Screenshot of the SMGW GUI of jOSEF. Client* 1, 2, *and* 3 *correspond to three different SMs that are connected to the SMGW, and the attribute list pane shows a detailed view of attribute* 1.0.1.8.1.255 *(electricity consumption in Wh) of client* 1.

```
Welcome to the Smart Meter Simulator user interface!
Menu:
[1] View server settings.
[2] View simulator settings.
[0] Exit.
1
View server settings:
-> Server is listening on port [33333]
-> Server is not using authentication for clients
Menu:
[1] View server settings.
[2] View simulator settings.
[0] Exit.
2
View simulator settings:
-> Simulated annual power consumption in kWh [1000.0]
-> Simulated annual power infeed in kWh [1000.0]
-> Simulation-time of a 'realtime-quarter-hour' in seconds [4]
-> Simulation resolution in miliseconds [1000]
Menu:
[1] View server settings.
[2] View simulator settings.
[0] Exit.
0
Closing program...
```

Figure 5.7: *Console log of the SM simulator CLI. The CLI allows the user to view its configuration. The configuration may only be changed on SM simulator startup via a configuration file.*

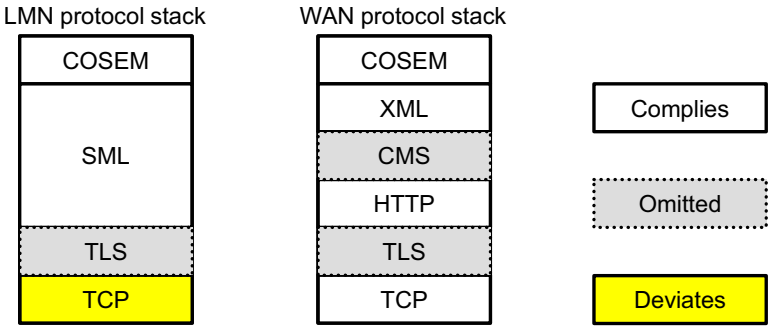


Figure 5.8: *Deviations of the LMN and WAN protocol stack of the current jOSEF implementation from BSI TR-03109. Currently omitted protocol layers are shown as gray, dashed boxes, and deviating protocol layers are shown as yellow, solid boxes.*

5.3.3.3 Meter Data Dissemination

When the SMGW retrieved new meter data from an SM, data conversion is necessary before actual dissemination to EMPs because LMN communication is based on COSEM over SML, and WAN communication is based on COSEM over XML. The common data model between SM, SMGW and EMP is COSEM so that data conversion works straightforward, i.e., a mapping of COSEM objects to XML is defined in [188]. After conversion to COSEM over XML, the SMGW sends the meter data to the EMP using the appropriate REST web service endpoint and HTTP methods. The EMP stores the received meter data and can perform further processing on the data.

5.3.3.4 Limitations

Our current framework implementation deviates from BSI TR-03109 in some minor points which we consider not important if the framework is used for laboratory communication experimentations only. These deviations need to be considered when using the framework for experiments involving insecure network connections between framework components. Minor deviations include that we do not support HDLC and serial links between the SMGW and the SM at the

moment because the SM simulator uses TCP to communicate with the SMGW, as shown in Figure 5.8. Further, the SMGW does not perform tariffing on received meter data, it does not support remote administration over the WAN, and it does not perform pseudonymization of meter data before sending them to EMPs. The framework implements only a limited subset of security functionalities, e.g., password-based authentication of SMGWs against SMs is available, but no authentication between SMGWs and EMPs. Additionally, we do not use TLS for LMN and WAN communication, and we do not use CMS to further secure WAN communication, as shown in Figure 5.8.

5.3.4 Illustration

We now illustrate the functionality of the proposed framework by experimentation. The setup of the experiment is described first, followed by experimental results from traffic experiments. Our results show that our framework enables easy modeling of typical smart metering topologies and communication patterns.

5.3.4.1 Experiment Setup and Methodology

To illustrate the functionality of the proposed framework, we created a simple dumbbell-like topology with one SM on the left side, one SMGW in the middle, and one EMP on the right side, as shown in Figure 5.9. The SM was simulated by an SM simulator instance, configured to use a H0 load profile [201] and a E0 generation profile [202]. The SMGW was configured to actively poll its associated SM every 2 seconds for new meter readings, and to forward all internally buffered metering data unsolicited to the EMP every 5 seconds. The EMP buffered meter readings from the SMGW, queryable via a REST web service interface. We deployed our setup on two end-hosts connected via a 100 Mbit/s Ethernet link.

5.3.4.2 Basic LMN and WAN Communication

First, the SM simulator, the SMGW and the EMP are started. The SMGW contacts the SM, queries for the SM's internal object list, and then subsequently

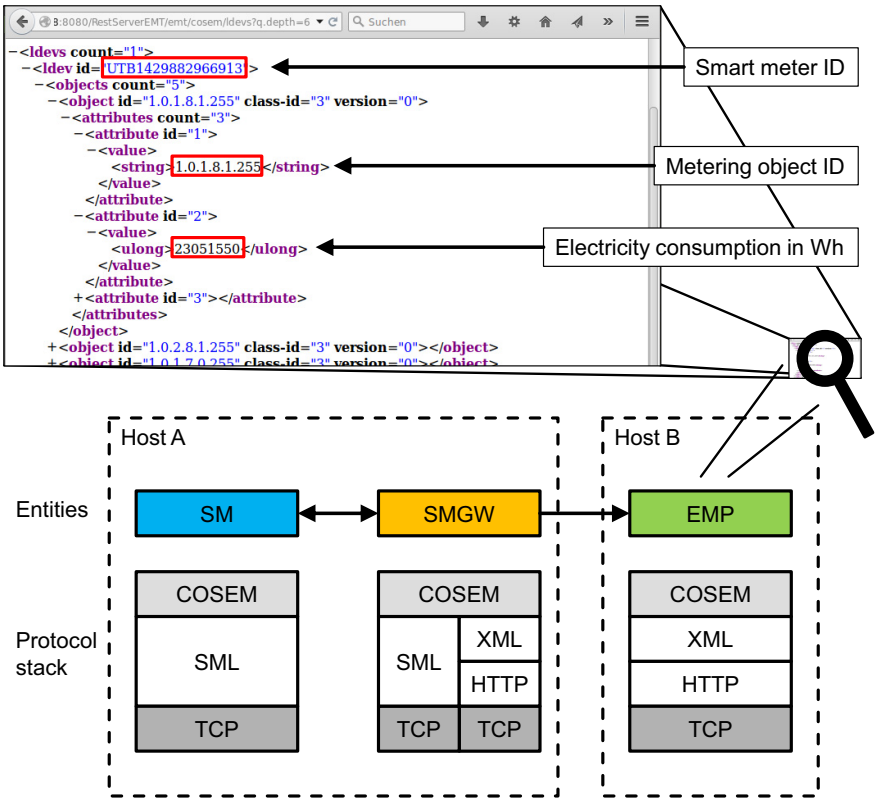


Figure 5.9: *Basic LMN and WAN communication including involved entities and protocol layers. A SMGW requests meter data from an SM, translates from COSEM over SML to COSEM over XML, and forwards the meter data to an EMP. The screenshot on the top shows what the XML structure of meter data looks like at the EMP.*

polls for electricity objects only. When the SMGW receives the first meter data from the SM over SML, it translates from COSEM over SML to COSEM over XML, and starts forwarding the meter data to the EMP using the HTTP PUT method. The EMP stores the received meter data and provides access to it over a REST web service. The screenshot on top of Figure 5.9 shows what meter data looks like at the EMP when the EMP is queried via the HTTP GET method, e.g., using a web browser. We can see that the SM with device id `UTB1429882966913` consists of five electricity objects. For better illustration,

only object `1.0.1.8.1.255` has been expanded in the figure, and the electricity consumption in Watt hours can be read.

5.3.5 Insights

BSI TR-03109 defines an SMGW-based smart metering infrastructure as it will be deployed in Germany. In this work, we presented the concept of BSI TR-03109, briefly reviewed implementations of smart metering protocols and architectures, and constituted that they only allow limited evaluation of smart metering communication aspects. Therefore, we proposed jOSEF, a Java-based open-source framework for smart metering communication experimentation, and evaluated its functionality. Our proposed framework combines and extends established protocol frameworks, thus providing a flexible tool for SMGW-based smart metering communication validation, e.g., adapting BSI TR-03109 to the C-DAX communication middleware. Furthermore, our extension of the jSML protocol library allows the implementation of independent programs. The source code of jOSEF and its subcomponents like the SML v.1.04 extension of the jSML library is available online [23].

5.4 Lessons Learned

The objective of this chapter was to investigate C-DAX UC3: the *future retail energy market (REM)*. In the future REM, any participant will be able to trade energy based on predicted supply and demand. As a consequence, the future REM will have many more participants and see more volatile prices than today. New trading and measurement infrastructures are necessary as enabling technology.

We analytically evaluated the communication performance of the PM architecture for a large-scale deployment model provided by Alliander N.V. and TNO. The PM communication framework by TNO aims at providing a communication and trading infrastructure at DSO scale. Our results show that PM enables scalable RETs with millions of participants requiring only moderate resources on the communication's side. The main reasons for scalable and efficient communication in PM are price and bid aggregation on the application layer, and the use of pub/sub as information dissemination paradigm on the communication layer.

We further shed light on smart metering as it will be deployed in Germany. We presented the concept of the future legally binding standard BSI TR-03109, briefly reviewed implementations of smart metering protocols and architectures, and constituted that they only allow limited evaluation of smart metering communication aspects. Therefore, we proposed jOSEF, a Java-based open-source framework for smart metering communication experimentation, and evaluated its functionality. Our proposed framework combines and extends established protocol frameworks, thus providing a flexible tool for SMGW-based smart metering communication validation. Furthermore, our extension of the jSML protocol library allows the implementation of independent programs. The source code of jOSEF and its subcomponents like the SML v.1.04 extension of the jSML library is available online [23].

6 Conclusion

Electrical power grids are undergoing major changes in operational procedures and monitoring. The retail energy market (REM) and distributed energy resources (DERs) are only two of many prominent examples of smart grid (SG) applications that in the long run will lead to the transformation of the classical electrical power grid towards a SG. Those applications demand changes from the electrical engineering and communications side. The main obstacles to the deployment of such SG applications are the limited scalability, reliability, and security of today's utility communication infrastructures.

In this monograph, we studied communication middleware and use cases in the context of SGs as part of the C-DAX project. The project aimed at developing a cyber-secure and scalable communication middleware for SGs to facilitate the flexible integration of emerging SG applications, and proved its benefits by suitable use cases, a prototype, and a field trial. Furthermore, it aimed at improving scalability compared to traditional client-server communication, and facilitating the development of new communication-based applications by providing a standardized transparent interface [39, 40, 42].

We first investigated the resource management issues of the C-DAX blueprint architecture SeDAX in Chapter 3. In the original SeDAX architecture, geographic hashing determines the coordinates of topics on the DT overlay, i.e., the mapping of topics to storage nodes. We showed that this static assignment of topics to coordinates can lead to severe load imbalance on SeDAX nodes and developed a Monte-Carlo optimization for node placement in SeDAX to minimize storage requirements. We further derived the storage requirements of SeDAX under optimal conditions and showed that they exceed those of an idealized storage system.

In a next step, we proposed a modification allowing dynamic reassignment of topics to coordinates while retaining the benefits of SeDAX, i.e., resilient overlay forwarding, decentralized control, and the ability to cope without a mapping system. We developed load balancing algorithms and demonstrated that they work well for static topic sizes. We further showed that a balanced SeDAX system may run out of balance if topic sizes change over time. Therefore, we presented a distributed algorithm for continuous load balancing offering a single parameter to trade off load balancing quality against load balancing effort in terms of moved load rates. In our evaluations, it kept a balanced system well balanced when topic sizes grew exponentially over time with different rates.

In Chapter 4, we described and evaluated the final C-DAX architecture. We discussed the reasons that eventually led to the design of the novel C-DAX architecture, and summarized and justified the enhancements and changes during the transition from the project's blueprint architecture SeDAX to C-DAX. In a next step, we gave a broad overview on the final C-DAX architecture, its design rationales, components, basic interactions, and its advanced features. We detailed several core features that represent a significant improvement in the original architecture specification. We simulated C-DAX using OMNeT++ to demonstrate its functionality. We also implemented a proof-of-concept of C-DAX that was deployed and evaluated on iMinds' Virtual Wall network testbed, EPFL's power network simulator, and Alliander's LiveLab SG test site. To the best of our knowledge, the project's field trial in Alliander's LiveLab is the first time ever that PMU-based RTSE has been deployed and demonstrated on the DG level. Subsequently, we conducted an extensive literature study and concluded that the C-DAX concept is competitive with existing pub/sub, ICN and message-queuing communication solutions, in particular also with the feature-rich DDS and JMS architectures. With regard to the special requirements for SG communication, the security architecture, the dual communication mode (broker-based and broker-less) and the adapter concept of C-DAX could be ported and re-used for improving existing well-established communication architectures.

Finally, in Chapter 5, we investigated C-DAX UC3: the *future retail energy market (REM)*. We assume that in the future REM, any participant will be able to trade energy based on predicted supply and demand. As a consequence, the future REM will have many more participants and see more volatile prices than today, and new trading and measurement infrastructures are necessary as enabling technology. The PM communication framework aims at providing such a trading infrastructure at DSO scale. We analytically evaluated the communication performance of the PM architecture for a realistic large-scale deployment model, and showed that PM enables scalable RETs with millions of participants requiring only moderate resources on the communication's side. We further shed light on smart metering as it will be deployed in Germany according to the BSI TR-03109 standard. We presented the concepts of the standard, and briefly reviewed implementations of related smart metering protocols and architectures. We concluded that existing implementations only allow limited evaluation of smart metering communication aspects. Consequently, we proposed jOSEF, a Java-based open-source framework for smart metering communication experimentation, and evaluated its functionality. jOSEF is a flexible tool for SMGW-based smart metering communication validation.

In the course of this monograph, we showed that SeDAX is not well suited for SG communication after all. In particular, the resource management issues of SeDAX are inherent to its design and can only be improved to a certain degree. As part of the C-DAX project, we developed mechanisms to improve pub/sub for SG communication. In particular, we showed how pub/sub can be made resilient against node failures, and how legacy and future SG applications can be integrated using adapters. We further showed how security can be realized in minimally trusted pub/sub architectures, i.e., if clients do not trust intermediary forwarding nodes. Beyond that, we improved the understanding of traffic dynamics and scalability of demand-response (DR), and proposed jOSEF as a flexible tool for validating smart metering communication.

Acronyms

AA	Auctioneer Agent 156
ACL	Access control list 103
ADC	Analog-to-digital converter
ADN	Active distribution network 15
AES	Advanced Encryption Standard 105
AGGR	Aggregator
AMI	Advanced metering infrastructure 5
API	Application programming interface 22
BEM	Bulk energy market 9
BON	Balanced Overlay Networks 31
BSI	Federal Office for Information Security 168
C-DAX	Cyber-secure Data And Control Cloud for power grids
CA	Concentrator Agent 156
CCDF	Complementary cumulative distribution function 35
CCTV	Closed-circuit television 153
CLI	Command line interface 174
CLS	Controllable local system 169
CMS	Cryptographic message syntax 171
CORDIS	Community Research and Development Information Service . . . 21
COSEM	COmpanion Standard for Energy Metering 155
CRL	Certificate revocation list 103
DA	Device Agent 156
DB	Data broker 80
DCC	Distribution control center 13

DDS-IS DDS dataspace Interconnection Service 139
DDS Data-Distribution Service 138
DER Distributed energy resource 1
DG Distribution grid 5
DHT Distributed hash table 29
DLMS Device Language Message Specification 155
DNO Distribution network operator 15
DN Designated node 80
DoS Denial of service 110
DR Demand-response 1
DSMR Dutch smart meter requirements 155
DSM Demand supply matching 19
DSO Distribution system operator
DTLS Datagram transport layer security 132
DT Delaunay-triangulated 3
EC European Commission 2
EEX European Energy Exchange 9
EHV Extra high-voltage 7
EMP External market participant 169
EPFL École Polytechnique Fédérale de Lausanne 94
ESM Energy supply manager
EV Electrical vehicle 143
FIB Forwarding Information Base 142
FIRMS Future Internet Routing Mapping System
FP7 7th Framework Programme for Research and Technological Development 2
GHF Geographic hashing function 28
GPL GNU General Public License 172
GPS Global positioning system 112
GUI Graphical user interface 126
HAN Home area network 169

HDLC High-level data link control 172
HMI Human-machine interface
HTTP Hypertext transfer protocol 174
HV High voltage 7
IBPM InfiniBand performance monitoring tool 5
ICN Information-centric networking 25
IED Intelligent electronic device 13
IP Internet protocol 14
JMS Java Message Service 139
jOSEF Java-based open-source smart meter gateway experimentation framework 5
JSON JavaScript Object Notation 172
KDF Key derivation function
LGPL GNU Lesser General Public License 172
LISP Locator/Identified Split Protocol
LMN Local metrological network 169
LTE Long Term Evolution
LV Low voltage 7
M2M Machine-to-machine 141
MAC Message authentication code 15
MgmSys Management system 80
MMS Manufacturing message specification 153
MonSys Monitoring system 83
MQTT Message Queue Telemetry Transport 141
MTU Master terminal unit 15
MUC Multi utility communication controller 172
MV Medium voltage 7
NASPI North American SynchroPhasor Initiative 16
NIST National Institute for Standards and Technology 21
NI National Instruments 127
NTP Network time protocol 171

OA Objective Agent 157
OBIS OBject Identification System 155
OGEMA Open Gateway Energy Management 173
OMG Object Management Group 138
OMS Open Metering System 155
P2P Peer-to-peer 20
PDC Phasor data concentrator 13
PIT Pending Interest Table 142
PKI Public-key infrastructure 171
PMU Phasor measurement unit 13
PM POWERMATCHER 2
PQ Power quality
pub/sub Publish/subscribe 1
PV Photo-voltaic 7
QoS Quality of Service 22
RBNB Ring Buffer Network Bus 137
RDS Resolver discovery system 83
REMP Resilient End-to-end Message Protection framework 105
REM Retail energy market 2
REST Representational state transfer 171
RET Retail energy transaction 5
RSA Rivest-Shamir-Adleman 103
RSL Resilience support level 88
RS Resolver 80
RTPS Real-Time Publish/Subscribe protocol 138
RTSE Real-time state estimation 15
RTU Remote terminal unit 13
SCADA Supervisory control and data access 13
SecServ Security server 83
SeDAX SEcure Data-centric Application eXtension 2
SG Smart grid 1

SMGW-PP SMGW protection profile 171
SMGW Smart meter gateway 168
SML Smart message language 155
SM Smart meter 22
TCP Transmission control protocol 14
TLS Transport layer security 132
TNO Netherlands Organisation for Applied Scientific Research 156
TSO Transmission system operator 21
UC Use case 151
UDP User datagram protocol 83
URI Uniform resource identifier 172
UTC-GPS Coordinated universal time derived from the global positioning system 16
VNI Virtual network interface 99
VPP Virtual power plant 19
WAN Wide area network 169
XML EXtensible Markup Language 125

Bibliography and References

Bibliography of the Author

— Journal Papers —

[1] M. Menth, M. Hartmann, and M. Hoefling, "FIRMS: A Mapping System for Future Internet Routing," *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Internet Routing Scalability*, vol. 28, no. 8, pp. 1326 – 1331, Oct. 2010. [Online]. Available: https://doi.org/10.1109/JSAC.2010.101010

[2] M. Hoefling, M. Menth, and M. Hartmann, "A Survey of Mapping Systems for Locator/Identifier Split Internet Routing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1842 – 1858, Nov. 2013. [Online]. Available: https://doi.org/10.1109/SURV.2013.011413.00039

[3] W. K. Chai, N. Wang, K. V. Katsaros, G. Kamel, S. Melis, M. Hoefling, B. Vieira, P. Romano, S. Sarri, T. Tesfay, B. Yang, F. Heimgaertner, M. Pignati, M. Paolone, M. Menth, G. Pavlou, E. Poll, M. Mampaey, H. Bontius, and C. Develder, "An Information-Centric Communication Infrastructure for Real-Time State Estimation of Active Distribution Networks," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 2134 – 2146, Jul. 2015. [Online]. Available: https://doi.org/10.1109/TSG.2015.2398840

[4] M. Hoefling, C. G. Mills, and M. Menth, "Distributed Load Balancing for the Resilient Publish/Subscribe Overlay in SeDAX," *accepted for publication in IEEE Transactions on Network and Service Management (TNSM)*, 2017. [Online]. Available: http://dx.doi.org/10.1109/TNSM.2016.2647678

— Conference Papers —

[5] D. Klein, M. Hoefling, M. Hartmann, and M. Menth, "Integration of LISP and LISP-MN into INET," in *5th International Workshop on OMNeT++*, Desenzano, Italy, Mar. 2012, pp. 299 – 306.

[6] M. Hoefling, F. Heimgaertner, and M. Menth, "C-DAX: A Secure and Resilient Communication and Information Infrastructure for Power Grids," in *2nd Workshop on "Renewable Energies, Smart Grid, and Green ICT"*, Stuttgart, Germany, Nov. 2012.

[7] M. Hoefling, C. G. Mills, and M. Menth, "Analyzing Storage Requirements of the Resilient Information-Centric SeDAX Architecture," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Vancouver, Canada, Oct. 2013. [Online]. Available: https://doi.org/10.1109/SmartGridComm.2013.6687975

[8] M. Hoefling, F. Heimgaertner, B. Litfinski, and M. Menth, "A Perspective on the Future Retail Energy Market," in *Workshop on Demand Modeling and Quantitative Analysis of Future Generation Energy Networks and Energy Efficient Systems (FGENET)*, Mar. 2014.

[9] M. Hoefling, C. G. Mills, and M. Menth, "Distributed Load Balancing for Resilient Information-Centric SeDAX Networks," in *IEEE Network Operations and Management Symposium (NOMS)*, Krakow, Poland, May 2014. [Online]. Available: https://doi.org/10.1109/NOMS.2014.6838254

[10] M. Hoefling, F. Heimgaertner, M. Menth, K. V. Katsaros, P. Romano, L. Zanni, and G. Kamel, "Enabling Resilient Smart Grid Communication over the Information-Centric C-DAX Middleware," in *ITG/GI International Conference on Networked Systems (NetSys)*, Cottbus, Germany, Mar. 2015. [Online]. Available: https://doi.org/10.1109/NetSys.2015.7089080

[11] M. Hoefling, F. Heimgaertner, M. Menth, and H. Bontius, "Traffic Estimation of the PowerMatcher Application for Demand Supply Matching in Smart Grids," in *ITG/GI Workshop on Middleware for a Smarter Use of Electric Energy (MidSEE)*, Cottbus, Germany, Mar. 2015. [Online]. Available: https://doi.org/10.1109/NetSys.2015.7089087

[12] M. Hoefling, F. Heimgaertner, D. Fuchs, M. Menth, P. Romano, T. Tesfay, M. Paolone, J. Adolph, and V. Gronas, "Integration of IEEE C37.118 and Publish/Subscribe Communication," in *IEEE International Conference on Communications (ICC)*, Jun. 2015. [Online]. Available: https://doi.org/10.1109/ICC.2015.7248414

[13] F. Heimgaertner, M. Hoefling, B. Vieira, E. Poll, and M. Menth, "A Security Architecture for the Publish/Subscribe C-DAX Middleware," in *Workshop on Security and Privacy for Internet of Things and Cyber-Physical Systems (IoT/CPS-Security) in conjunction with IEEE International Conference on Communications (ICC)*, London, UK, Jun. 2015. [Online]. Available: https://doi.org/10.1109/ICCW.2015.7247573

[14] M. Hoefling, F. Heimgaertner, D. Fuchs, and M. Menth, "jOSEF: A Java-Based Open-Source Smart Meter Gateway Experimentation Framework," in *D-A-CH Conference on Energy Informatics*, Karlsruhe, Germany, Nov. 2015.

[15] M. Hoefling, F. Heimgaertner, and M. Menth, "Advanced Communication Modes for the Publish/Subscribe C-DAX Middleware," in *IFIP/IEEE International Workshop on the Management of Fog Computing and the Internet of Things (ManFIoT) in conjunction with IEEE Network Operations and Management Symposium (NOMS)*, Istanbul, Turkey, Apr. 2016. [Online]. Available: https://doi.org/10.1109/NOMS.2016.7503009

[16] F. Heimgaertner, M. Hoefling, M. Menth, E. Schur, F. Truckenmueller, H. Hagenlocher, J. Zunke, A. Frey, D. Ebinger, S. Jaegers, L. Duerr, T. Roeger, K. Lindner, and C. Kahlert, "The Demonstration Project Virtual Power Plant Neckar-Alb," in *VDE Kongress 2016: Internet der Dinge*, Mannheim, Germany, Nov. 2016.

— Others —

[17] M. Menth, M. Hartmann, and M. Hoefling, "Demo: a Future InteRnet Mapping System (FIRMS)," in *Würzburg Workshop on IP: Visions of Future Generation Networks (EuroView)*, Würzburg, Germany, Jul. 2009.

[18] ——, "Mapping Systems for Loc/ID Split Internet Routing," University of Würzburg, Institute of Computer Science, Technical Report, No. 472, May 2010.

[19] M. Hartmann, D. Hock, M. Hoefling, T. Neubert, and M. Menth, "Demo: Demonstration of the Future InteRnet Mapping System (FIRMS) in the G-Lab Experimental Facility," in *Würzburg Workshop on IP: Visions of Future Generation Networks (EuroView)*, Würzburg, Germany, Aug. 2010.

[20] D. Klein, M. Hartmann, M. Hoefling, and M. Menth, "Demo: Improvements to LISP Mobile Node Including NAT Traversal," in *Würzburg Workshop on IP: Visions of Future Generation Networks (EuroView)*, Würzburg, Germany, Aug. 2010.

[21] M. Hoefling, M. Menth, C. Kniep, and M. Camen, "Demo: IBPM: An Open-Source-Based Framework for InfiniBand Performance Monitoring," in *GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB)*, Kaiserslautern, Germany, Mar. 2012.

[22] M. Schmidt, F. Heimgaertner, M. Hoefling, and M. Menth, "Demo: Enabling Physical Interaction with Virtualized Testbeds for Hands-on Networking Courses," in *ITG/GI International Conference on Networked Systems (NetSys)*, Cottbus, Germany, Mar. 2015.

[23] "jOSEF: A Java-Based Open-Source Smart Meter Gateway Experimentation Framework," http://kn.inf.uni-tuebingen.de/software/josef, last visited 14.08.2015.

[24] C-DAX Consortium and M. Hoefling (Contributor), "C-DAX Deliverable D2.1: C-DAX Requirements - Use Case Descriptions for Domains 1, 2 and 3 and Derived C-DAX Requirements R1.0," Apr. 2013.

[25] ——, "C-DAX Deliverable D2.2: C-DAX Requirements - Use Case Descriptions for Domains 1, 2 and 3 and Derived C-DAX Requirements R2.0," Mar. 2014.

[26] ——, "C-DAX Deliverable D3.1: Specification of the Initial C-DAX Architecture and Basic Mechanisms, Protocols and Algorithms," Sep. 2013.

[27] ——, "C-DAX Deliverable D3.2: Specification of the Security and Privacy Techniques for the C-DAX Infrastructure," Mar. 2014.

[28] ——, “C-DAX Deliverable D3.3: Specification of the C-DAX Information Models and Configuration Management,” Jun. 2014.

[29] ——, “C-DAX Deliverable D3.5: Specification of the Final C-DAX Infrastructure,” Sep. 2015.

[30] ——, “C-DAX Deliverable D4.1: Modeling of Use Cases,” Jun. 2013.

[31] M. Hoefling (Editor) and C-DAX Consortium, “C-DAX Deliverable D4.2: Validation of C-DAX for Use Cases,” Dec. 2013.

[32] ——, “C-DAX Deliverable D4.3: Provisioning Strategies for C-DAX,” Jun. 2014.

[33] C-DAX Consortium and M. Hoefling (Contributor), “C-DAX Deliverable D4.4: Optimization of Resource Allocation in C-DAX,” Dec. 2014.

[34] M. Hoefling (Editor) and C-DAX Consortium, “C-DAX Deliverable D4.5: Strength/Weakness Analysis of C-DAX,” Sep. 2015.

[35] C-DAX Consortium and M. Hoefling (Contributor), “C-DAX Deliverable D5.1: Laboratory Communications Tests Results of C-DAX Compatible Grid Devices,” Nov. 2014.

[36] ——, “C-DAX Deliverable D5.2: Test Results of Field Trial with C-DAX Compatible Grid Devices,” Feb. 2016.

[37] M. Hoefling, “C-DAX prototype live demo during second C-DAX project review,” YouTube, Mar. 2015. [Online]. Available: https://www.youtube.com/watch?v=Qp0-lx2bVls

General References

[38] J. Kok, B. Roossien, P. MacDougall, O. Pruissen, G. Venekamp, I. Kamphuis, J. Laarakkers, and C. Warmer, “Dynamic Pricing by Scalable Energy Management Systems - Field Experiences and Simulation Results using PowerMatcher,” in *IEEE Power and Energy Society General Meeting*. San Diego, CA, USA: IEEE, Jul. 2012.

[39] Y.-J. Kim, J. Lee, G. Atkinson, H. Kim, and M. Thottan, “SeDAX: A Scalable, Resilient, and Secure Platform for Smart Grid Communications,” *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 30, no. 6, 2012.

[40] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “The Many Faces of Publish/Subscribe,” *ACM Computing Surveys*, vol. 35, no. 2, pp. 114 – 131, 2003.

[41] C-DAX Consortium, “Cyber-secure Data And Control Cloud for Power Grids,” 2014. [Online]. Available: http://www.cdax.eu/

[42] K. V. Katsaros, W. K. Chai, N. Wang, G. Pavlou, H. Bontius, and M. Paolone, “Information-Centric Networking for Machine-to-Machine Data Delivery - A Case Study in Smart Grid Applications,” *IEEE Network, Special Issue on Information-Centric Networking Beyond Baseline Scenarios: Research Advances and Implementation*, vol. 28, no. 3, 2014.

[43] N. Capodieci, G. A. Pagani, G. Cabri, and M. Aiello, "Smart Meter Aware Domestic Energy Trading Agents," in *E-Energy Market Challenge Workshop*, 2011.

[44] C. Bendel, D. Nestle, and J. Ringelstein, "Bidirektionales Energiemanagement im Niederspannungsnetz: Strategie, Umsetzung und Anwendungen," *e&i Elektrotechnik und Informationstechnik*, vol. 125, 2008.

[45] C. Block, J. Collins, and W. Ketter, "Agent-based Competitive Simulation: Exploring Future Retail Energymarkets," in *International Conference on Electronic Commerce*, Honolulu, HI, USA, 2010.

[46] Bundesamt für Sicherheit in der Informationstechnik, "Technische Richtlinie BSI TR-03109."

[47] European Energy Exchange AG, "European Energy Exchange (EEX)," 2013, last visited November 2013. [Online]. Available: http://www.eex.com/

[48] Wikipedia, "Stromhandel," 2016, last visited 26.08.2016. [Online]. Available: https://de.wikipedia.org/w/index.php?title=Stromhandel&oldid=157356098

[49] Wikipedia, "Operating reserve," 2016, last visited 26.08.2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Operating_reserve&oldid=711911738

[50] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[51] B. Vieira and E. Poll, "A Security Protocol for Information-centric Networking in Smart Grids," in *ACM Workshop on Smart Energy Grid Security (SEGS)*, Nov. 2013.

[52] Y.-J. Kim, V. Kolesnikov, H. Kim, and M. Thottan, "Resilient End-to-End Message Protection for Large-scale Cyber-Physical System Communications," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov. 2012.

[53] Y.-J. Kim, J. Lee, G. Atkinson, and M. Thottan, "GridDataBus: Information-centric Platform for Scalable Secure Resilient Phasor-Data Sharing," in *IEEE INFOCOM Smart Grid Workshop*, 2012.

[54] Alliander N.V., "LiveLab," 2015. [Online]. Available: https://www.alliander.com/en/innovation/our-innovations

[55] IEC, "IEC 60870-5-104 (IEC 104): Network access for IEC 60870-5-101 using standard transport profiles," 2000.

[56] IEC, "IEC 61850: Power Utility Automation," 2003.

[57] G. T. Heydt, "The Next Generation of Power Distribution Systems," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, 2010.

[58] R. Singh, B. Pal, and R. Jabr, "Choice of Estimator for Distribution System State Estimation," *IET Generation, Transmission & Distribution*, vol. 3, no. 7, 2009.

[59] J. Liu, J. Tang, F. Ponci, A. Monti, C. Muscas, and P. A. Pegoraro, "Trade-Offs in PMU Deployment for State Estimation in Active Distribution Grids," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, 2012.

[60] P. Romano and M. Paolone, "Enhanced Interpolated-DFT for Synchrophasor Estimation in FPGAs: Theory, Implementation, and Validation of a PMU Prototype," *IEEE Transactions on Instrumentation and Measurement*, 2014.

[61] L. Zanni, M. Pignati, S. Sarri, R. Cherkaoui, and M. Paolone, "Probabilistic Assessment of the Process-Noise Covariance Matrix of Discrete Kalman Filter State Estimation of Active Distribution Networks," in *International Conference of Probabilistic Methods Applied to Power Systems (PMAPS)*, Durham, UK, 2014.

[62] M. Pignati, L. Zanni, S. Sarri, R. Cherkaoui, J.-Y. L. Boudec, and M. Paolone, "A Pre-Estimation Filtering Process of Bad Data for Linear Power Systems State Estimators Using PMUs," in *Power Systems Computation Conference (PSCC)*, Wroclaw, Poland, 2014.

[63] "IEEE Standard for Synchrophasor Measurements for Power Systems," IEEE C37.118.1-2011, December 2011.

[64] "IEEE Standard for Synchrophasor Data Transfer for Power Systems," IEEE C37.118.2-2011, December 2011.

[65] P. T. Myrda and K. Koellner, "NASPInet - The Internet for Synchrophasors," in *International Conference on Systems Sciences (HICSS)*. IEEE, 2010.

[66] North American Synchro-Phasor Initiative, "Data Bus Technical Specifications for North American Synchro-Phasor Initiative Network," 2009. [Online]. Available: https://www.naspi.org/File.aspx?fileID=587

[67] Power System Operation Corporation Limited, "Synchrophasors Initiative in India," Power System Operation Corporation Limited, Power Grid Corporation of India Limited, New Delhi, India, technical report, Dec. 2013.

[68] A. Eßer, M. Franke, A. Kamper, and D. Möst, "Future Power Markets – Impacts of Consumer Response and Dynamic Retail Prices on Electricity Markets," *WIRTSCHAFTSINFORMATIK*, vol. 49, 2007.

[69] M. Franke, D. Rolli, A. Kamper, A. Dietrich, A. Geyer-Schulz, P. Lockemann, H. Schmeck, and C. Weinhardt, "Impacts of Distributed Generation from Virtual Power Plants," in *International Sustainable Development Research Conference*, vol. 11, Helsinki, Finland, Jun. 2005.

[70] C. Bendeli, D. Nestle, J. Ringelstein, A. Eßer, D. Möst, O. Rentz, M. Franke, and A. Geyer-Schulz, "Marktmodell für ein dezentral organisiertes Energiemanagement im elektrischen Verteilnetz - Grundlage für ein internetbasiertes Managementsystem," in *ETG-Kongress, Fachtagung Webbasierte Automatisierung in der elektrischen Energietechnik*, Karlsruhe, Germany, 2007.

[71] L. Gkatzikis, I. Koutsopoulos, and T. Salonidis, "The Role of Aggregators in Smart Grid Demand Response Markets," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 31, no. 7, Jul. 2013.

[72] C. Yeung, A. Poon, and F. Wu, "Game Theoretical Multi-Agent Modelling of Coalition Formation for Multilateral Trades," *IEEE Transactions on Power Systems*, vol. 14, no. 3, 1999.

[73] J. Contreras, O. Candiles, J. de la Fuente, and T. Gomez, "Auction Design in Day-ahead Electricity Markets," *IEEE Transactions on Power Systems*, vol. 16, no. 1, 2001.

[74] C. J. Hazard and P. R. Wurman, "The Game of Scale: Decision Making with Economies of Scale," in *International Conference on Electronic Commerce*, Minneapolis, MN, USA, 2007.

[75] C. Corchero, E. Mijangos, and F.-J. Heredia, "A New Optimal Electricity Market Bid Model Solved Through Perspective Cuts," *TOP*, vol. 21, no. 1, 2013.

[76] G. Chalkiadakis, V. Robu, R. Kota, A. Rogers, and N. R. Jennings, "Cooperatives of Distributed Energy Resources for Efficient Virtual Power Plants," in *International Conference on Autonomous Agents and Multiagent Systems*, Taipei, Taiwan, May 2011.

[77] N. Capodieci, "P2P Energy Exchange Agent Platform Featuring a Game Theory Related Learning Negotiation Algorithm," Master's thesis, Universita degli Studi di Modena e Reggio Emilia, 2011.

[78] The Smart Grid Interoperability Panel Cyber Security Working Group, "NISTIR 7628: Guidelines for Smart Grid Cyber Security," Sep. 2010. [Online]. Available: http://www.nist.gov/smartgrid/upload/nistir-7628_total.pdf

[79] C. F. Covrig, M. Ardelean, J. Vasiljevska, A. Mengolini, G. Fulli, and E. Amoiralis, *Smart Grid Projects Outlook 2014*. Luxembourg: Publications Office of the European Union, 2014. [Online]. Available: http://ses.jrc.ec.europa.eu/sites/ses.jrc.ec.europa.eu/files/u24/2014/report/ld-na-26609-en-n_smart_grid_projects_outlook_2014_-_online.pdf

[80] European Commission, "CORDIS – Community Research and Development Information Service," 2016, last visited 27.08.2016. [Online]. Available: http://cordis.europa.eu/

[81] European Commission, "FP7 Project: Increasing the penetration of renewable energy sources in the distribution grid by developing control strategies and using ancillary services (INCREASE)," 2016, last visited 27.08.2016. [Online]. Available: http://cordis.europa.eu/project/rcn/109974_en.html

[82] European Commission, "FP7 Project: Sustainable and robust networking for smart electricity distribution (SUNSEED)," 2016, last visited 27.08.2016. [Online]. Available: http://cordis.europa.eu/project/rcn/189098_en.html

[83] European Commission, "FP7 Project: Smart Control of Energy Distribution Grids over Heterogeneous Communication Networks (SmartC2Net)," 2016, last visited 27.08.2016. [Online]. Available: http://cordis.europa.eu/project/rcn/106172_en.html

[84] European Commission, "FP7 Project: Development of Novel ICT tools for integrated Balancing Market Enabling Aggregated Demand Response and Distributed Generation Capacity (EBADGE)," 2016, last visited 27.08.2016. [Online]. Available: http://cordis.europa.eu/project/rcn/105542_en.html

[85] A. Videla and J. J. W. Williams, *RabbitMQ in Action: Distributed Messaging for Everyone*. Shelter Island NY: Manning, 2012.

[86] iMinds, "ICON Project: Smart WInd Farm conTrol (SWIFT)," 2016, last visited 29.08.2016. [Online]. Available: http://www.iminds.be/en/projects/swift

[87] iMinds, "ICON Project: optimizing Monetization of Industrial Energy flexibility (MonIEflex)," 2016, last visited 29.08.2016. [Online]. Available: http://www.iminds.be/en/projects/monieflex

[88] REstore N.V., "Demand Response and flexible energy management | REstore," 2016, last visited 29.08.2016. [Online]. Available: https://www.restore.eu/en/homepage

[89] "Virtuelles Kraftwerk Neckar-Alb," 2016, last visited 29.08.2016. [Online]. Available: http://www.virtuelles-kraftwerk-neckar-alb.de/

[90] Wikipedia, "Delaunay-Triangulierung," 2016, last visited 24.08.2016. [Online]. Available: https://de.wikipedia.org/w/index.php?title=Delaunay-Triangulierung&oldid=151088647

[91] Wikipedia, "Delaunay triangulation," 2016, last visited 24.08.2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Delaunay_triangulation&oldid=735203323

[92] F. Aurenhammer and R. Klein, "Voronoi Diagrams," University of Hagen, Chair for Practical Computer Science VI, Technical Report, No. 198, 1996.

[93] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-Centric Storage in Sensornets with GHT, a Geographic Hash Table," *Mobile Networks and Applications (MONET)*, vol. 8, no. 4, 2003.

[94] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, and J. Wilcox, "Information-Centric Networking: Seeing the Forest for the Trees," in *ACM Workshop on Hot Topics in Networks (HotNets)*, Nov. 2011.

[95] D. Trossen and G. Parisis, "Designing and Realizing an Information-Centric Internet," *IEEE Mobile Communications*, vol. 50, no. 7, 2012.

[96] R. Alimi, L. Chen, D. Kutscher, H. H. Liu, A. Rahman, H. Song, Y. R. Yang, D. Zhang, and N. Zong, "An Open Content Delivery Infrastructure using Data Lockers," in *ACM SIGCOMM Workshop on Information-Centric Networking (ICN)*, 2012.

[97] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive Forwarding in Named Data Networking," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 42, no. 3, 2012.

[98] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *ACM CoNEXT*, 2009.

[99] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinsky, K. H. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and Beyond) Network Architecture," in *ACM SIGCOMM*, Aug. 2007.

[100] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 31, no. 4, 2001.

[101] A. Ghose, J. Grossklags, and J. Chuang, "Resilient Data-Centric Storage in Wireless Ad-Hoc Sensor Networks," in *International Conference on Mobile Data Management (MDM)*, 2003.

[102] M. Shorfuzzaman, P. Graham, and R. Eskicioglu, "Distributed Placement of Replicas in Hierarchical Data Grids with User and System QoS Constraints," in *IEEE 3PGCIC*, 2011.

[103] X. Tang and J. Xu, "QoS-Aware Replica Placement for Content Distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 10, 2005.

[104] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed publish/subscribe inter-networking," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 39, no. 4, 2009.

[105] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 31, no. 4, 2001.

[106] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials (COMST)*, vol. 7, no. 2, 2005.

[107] P. Felber, P. Kropf, E. Schiller, and S. Serbu, "Survey on Load Balancing in Peer-to-Peer Distributed Hash Tables," *IEEE Communications Surveys & Tutorials (COMST)*, vol. 16, no. 1, pp. 473 – 492, 2014.

[108] K. Kenthapadi and G. S. Manku, "Decentralized Algorithms Using both Local and Random Probes for P2P Load Balancing," in *Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, Jul. 2005.

[109] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, "Load Balancing in Structured P2P Systems," in *International Workshop on Peer-to-Peer Systems (IPTPS)*, Feb. 2003.

[110] B. Godfrey, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, "Load Balancing in Dynamic Structured P2P Systems," in *IEEE Infocom*, Mar. 2004.

[111] J. Byers, J. Considine, and M. Mitzenmacher, "Simple Load Balancing for Distributed Hash Tables," in *International Workshop on Peer-to-Peer Systems (IPTPS)*, Feb. 2003.

[112] M. D. Mitzenmacher, "The Power of Two Choices in Randomized Load Balancing," Ph.D. dissertation, University of California at Berkeley, 1996.

[113] M. Mitzenmacher, A. W. Richa, and R. Sitaraman, "The power of two random choices: A survey of techniques and results," in *Handbook of Randomized Computing*. Kluwer, 2000, pp. 255 – 312.

[114] J. S. A. Bridgewater, P. O. Boykin, and V. P. Roychowdhury, "Balanced Overlay Networks (BON): An Overlay Technology for Decentralized Load Balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 8, pp. 1122–1133, Aug. 2007.

[115] G. Even and M. Medina, "Parallel Randomized Load Balancing: A Lower Bound for a More General Model," in *SOFSEM 2010: Theory and Practice of Computer Science*. Springer Berlin Heidelberg, 2010, vol. 5901.

[116] P. Bellavista, A. Corradi, and A. Reale, "Quality of Service in Wide Scale Publish-Subscribe Systems," *IEEE Communications Surveys & Tutorials (COMST)*, vol. 16, no. 3, pp. 1591–1616, 2014.

[117] M. Krasnyansky and M. Yevmenkin, "Universal tun/tap device driver," 2007.

[118] M. Steiner, G. Tsudik, and M. Waidner, "CLIQUES: A New Approach to Group Key Agreement," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, May 1998.

[119] Y. Kim, A. Perrig, and G. Tsudik, "Communication-Efficient Group Key Agreement," in *IFIP Conference on Information Security (IFIP/Sec'01)*, Jun. 2001.

[120] N. Pandit and K. Khandeparkar, "Design and Implementation of IEEE C37.118 based Phasor Data Concentrator & PMU Simulator for Wide Area Measurement System," Indian Institute of Technology, Bombay, Tech. Rep., 2012.

[121] iMinds, "Virtual Wall - Generic test environment for advanced network, distributed software and service evaluation, and scalability research," 2014. [Online]. Available: http://www.iminds.be/en/develop-test/ilab-t/virtual-wall

[122] S. Bouckaert, P. Becue, B. Vermeulen, B. Jooris, I. Moerman, and P. Demeester, "Federating Wired and Wireless Test Facilities through Emulab and OMF: The iLab.t Use Case," in *International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*. Ghent University, Department of Information Technology, 2012.

[123] Grid Protection Alliance, "PMU Connection Tester." [Online]. Available: http://pmuconnectiontester.codeplex.com/

[124] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *European Simulation and Modelling Conference (ESM)*, Prague, Czech Republic, Jun. 2001.

[125] A. Varga, "OMNeT++ 4.3 released," Apr. 2013, last visited 27.06.2016. [Online]. Available: http://www.omnetpp.org/

[126] A. Varga, "INET-2.2.0 stable released," Aug. 2013, last visited 27.06.2016. [Online]. Available: http://inet.omnetpp.org/

[127] S. Tilak, P. Hubbard, M. Miller, and T. Fountain, "The Ring Buffer Network Bus (RBNB) DataTurbine Streaming Data Middleware for Environmental Observing Systems," in *IEEE Conference on e-Science and Grid Computing*, 2007.

[128] H. Gjermundrod, D. E. Bakken, C. H. Hauser, and A. Bose, "GridStat: A Flexible QoS-Managed Data Dissemination Framework for the Power Grid," *IEEE Transactions on Power Delivery*, vol. 24, no. 1, 2009.

[129] E. Solum, C. Hauser, R. Chakravarthy, and D. E. Bakken, "Modular Over-the-Wire Configurable Security for Long-Lived Critical Infrastructure Monitoring Systems," in *ACM International Conference on Distributed Event-Based Systems (DEBS)*, Nashville, TN, USA, Jul. 2009.

[130] R. Chakravarthy, C. Hauser, and D. E. Bakken, "Long-lived Authentication Protocols for Process Control Systems," *International Journal of Critical Infrastructure Protection*, vol. 3, no. 3-4, pp. 174 – 181, Dec. 2010.

[131] D. E. Bakken, C. Hauser, and H. Gjermundroed, "Periodically Updated Variables: Wide-Area Publish-Subscribe Middleware Supporting Electric Power Monitoring, Control, and Protection," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, Montreal, QC, Canada, Jun. 2009.

[132] G. Pardo-Castellote, "OMG Data-Distribution Service: Architectural Overview," in *ICDCS Workshops*, 2003, pp. 200–206.

[133] S. Schneider and B. Farabaugh, "Is DDS for You?" RTI Whitepaper, 2009.

[134] F. Garcia, "Data Distribution Service (DDS) Community RTI Connext Users - Content Filtered Topics," 2013, last visited 29.09.2015. [Online]. Available: https://community.rti.com/examples/content-filtered-topic

[135] J. M. Lopez-Vega, J. Povedano-Molina, G. Pardo-Castellote, and J. M. Lopez-Soler, "A Content-aware Bridging Service for Publish/Subscribe Environments," *Journal of Systems and Software*, vol. 86, no. 1, pp. 108 – 124, Jan. 2013.

[136] Oracle Corporation, "Java Message Service 2.0 Released," 2013. [Online]. Available: http://www.oracle.com/technetwork/java/jms/index.html

[137] M. Richards, R. Monson-Haefel, and D. A. Chappell, "Java Message Service," 2009.

[138] M. Reiferson and J. Czebotar, "NSQ - a real-time distributed messaging platform," 2014. [Online]. Available: http://nsq.io/

[139] P. Hintjens, *ZeroMQ: Messaging for Many Applications*. O'Reilly Media, Inc., 2013.

[140] P. Hintjens, "Zeromq - the guide," 2013. [Online]. Available: http://zguide.zeromq.org/page:all

[141] D. J. Bernstein, "CurveCP: Usable Security for the Internet," 2014, last visited 30.12.2015. [Online]. Available: http://curvecp.org

[142] iMatrix Corporation, "CurveZMQ - Security for ZeroMQ," 2013, last visited 30.12.2015. [Online]. Available: http://curvezmq.org

[143] A. Stanford-Clark and A. Nipper, "MQ Telemetry Transport," 2014. [Online]. Available: http://www.mqtt.org/

[144] dc-square GmbH, "HiveMQ Enterprise MQTT Broker," 2015, last visited 30.12.2015. [Online]. Available: http://www.hivemq.com

[145] D. Trossen et al., "FP7 Project: PURSUIT: Conceptual Architecture: Principles, Patterns and Sub-components Descriptions," May 2011. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[146] "FP7 Project: Publish-Subscribe Internet Routing Paradigm (PSIRP)," 2010. [Online]. Available: http://www.psirp.org/

[147] "FP7 Project: Scalable and Adaptive Internet Solutions (SAIL)," 2013. [Online]. Available: http://www.sail-project.eu/

[148] P. A. Aranda et al., "FP7 Project: 4WARD: Final Architectural Framework," Jun. 2010. [Online]. Available: http://www.4ward-project.eu/

[149] "FP7 Project: COntent Mediator architecture for content-aware nETworks (COMET)," 2013. [Online]. Available: http://www.comet-project.org/

[150] "FP7 Project: CONVERGENCE," 2013. [Online]. Available: http://www.ict-convergence.eu/

[151] "NSF Project: Named Data Networking (NDN)," 2015. [Online]. Available: http://www.named-data.net/

[152] "Paolo Alto Research Center (PARC) Project: Content Centric Networking (CCNx)," 2015. [Online]. Available: http://www.ccnx.org/

[153] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A Survey of Information-Centric Networking Research," *IEEE Communications Surveys & Tutorials (COMST)*, vol. 16, no. 2, pp. 1024 – 1049, Jul. 2014.

[154] W. K. Chai, N. Wang, I. Psaras, G. Pavlou, C. Wang, G. de Blas, F. Ramon-Salguero, L. Liang, S. Spirou, A. Beben, and E. Hadjioannou, "CURLING: Content-Ubiquitous Resolution and Delivery Infrastructure for Next Generation Services," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 112 – 120, Mar. 2011.

[155] E. AbdAllah, H. Hassanein, and M. Zulkernine, "A Survey of Security Attacks in Information-Centric Networking," *IEEE Communications Surveys & Tutorials (COMST)*, vol. 17, no. 3, pp. 1441 – 1454, Jan. 2015.

[156] F. van den Broek, E. Poll, and B. Vieira, "Securing the Information Infrastructure for EV Charging," in *International Workshop on Communication Applications in Smart Grid (CASG 2015)*, Bradford, UK, Jul. 2015.

[157] C. Esposito and M. Ciampi, "On Security in Publish/Subscribe Services: a Survey," *IEEE Communications Surveys & Tutorials (COMST)*, to appear.

[158] M. Zhang, H. Luo, and H. Zhang, "A Survey of Caching Mechanisms in Information-Centric Networking," *IEEE Communications Surveys & Tutorials (COMST)*, vol. 17, no. 3, pp. 1473 – 1499, Apr. 2015.

[159] K. V. Katsaros, X. Vasilakos, T. Okwii, G. Xylomenos, G. Pavlou, and G. C. Polyzos, "On the Inter-domain Scalability of Route-by-Name Information-Centric Network Architectures," in *IFIP-TC6 Networking*, Toulouse, France, May 2015.

[160] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access Control Enforcement Delegation for Information-Centric Networking Architectures," in *ACM SIGCOMM Workshop on Information-Centric Networking (ICN)*, New York, NY, USA, Aug. 2012.

[161] F. Borggrefe and A. Nüßler, "Auswirkungen fluktuierender Windverstromung auf Strommärkte und Übertragungsnetze," *uwf UmweltWirtschaftsForum*, vol. 17, no. 4, 2009.

[162] CIGRE Working Group C6.11, "Development and Operation of Active Distribution Networks," Apr. 2011.

[163] K. C. Budka and J. G. Deshpande, "Smart Grid Bandwidth Requirements in LTE Macrocells," Alcatel-Lucent Technical Whitepaper, Bell Labs, Whitepaper, 2008.

[164] G. Karagiannis, G. T. Pham, A. D. Nguyen, G. J. Heijenk, B. R. Haverkort, and F. Campfens, "Performance of LTE for Smart Grid Communications," in *GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB)*, 2014.

[165] S. McCanne, S. Floyd, and et al., "The Network Simulator - ns-3," 2014. [Online]. Available: http://www.nsnam.org/

[166] P. Kansal and A. Bose, "Bandwidth and Latency Requirements for Smart Transmission Grid Applications," *IEEE Transactions on Smart Grid (ToSG)*, vol. 3, no. 3, 2012.

[167] S. McCanne and S. Floyd, "The Network Simulator - ns-2," 2011. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/Main_Page

[168] G. Deconinck, "An Evaluation of Two-Way Communication Means for Advanced Metering in Flanders (Belgium)," in *IEEE Instrumentation and Measurement Technology Conference*, 2008.

[169] W. Luan, D. Sharp, and S. Lancashire, "Smart Grid Communication Network Capacity Planning for Power Utilities," in *IEEE PES Transmission and Distribution Conference and Exposition*, 2010.

[170] International Electrotechnical Comission, "Electricity Metering Data Exchange - The DLMS/COSEM Suite - Part 5-3: DLMS/COSEM Application Layer," IEC 62056-5-3 ed1.0, 2013.

[171] International Electrotechnical Comission, "Electricity Metering Data Exchange - The DLMS/COSEM Suite - Part 6-2: COSEM Interface Classes," IEC 62056-6-2 ed1.0, 2013.

[172] International Electrotechnical Comission, "Electricity Metering Data Exchange - The DLMS/COSEM Suite - Part 6-1: Object Identification System (OBIS) ," IEC 62056-6-1 ed1.0, 2013.

[173] Bundesamt für Sicherheit in der Informationstechnik, "BSI TR-03109-1 Anlage IV: Feinspezifikation Drahtgebundene LMN-Schnittstelle, Teil b: SML - Smart Message Language," SML Version 1.04.

[174] International Electrotechnical Comission, "Electricity Metering Data Exchange - Part 5-3-8 Smart Message Language SML," IEC 62056-5-3-8 (future standard).

[175] European Committee for Standardization, "Communication Systems for and Remote Reading of Meters - Part 1: Data Exchange," EN 13757-1:2015-01, 2015.

[176] European Committee for Standardization, "Communication Systems for and Remote Reading of Meters - Part 4: Wireless Meter Readout," EN 13757-4:2014-02, 2014.

[177] European Committee for Standardization, "Communication Systems for and Remote Reading of Meters - Part 2: Physical and Link Layer," EN 13757-2:2004, 2004.

[178] European Committee for Standardization, "Communication Systems for and Remote Reading of Meters - Part 3: Dedicated Application Layer," EN 13757-3:2013-08, 2013.

[179] OMS Group, "Open Metering System Specification, Volume 1: General Part," OMS Spec Vol1 1.4.0, 2011.

[180] OMS Group, "Open Metering System Specification, Volume 2: Primary Communication, Version 4.0.2," OMS Spec Vol2 4.0.2, 2014.

[181] Netbeheer Nederland, "Dutch Smart Meter Requirements: P1 Companion Standard," DSMR Version 5.0.

[182] K. Kok, C. Warmer, and R. Kamphuis, "PowerMatcher: multiagent control in the electricity infrastructure," in *Proceedings of the 4th int. joint conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, vol. industry track. New York, NY, USA: ACM Press, Jul. 2005.

[183] TNO, "PowerMatcher Smart Grid Technology," 2014. [Online]. Available: http://www.powermatcher.net/

[184] B. Roossien, "Field-Test Upscaling of Multi-Agent Coordination in the Electricity Grid," in *Proceedings of the 20th International Conference on Electricity Distribution (CIRED).* Prague, Poland: IET-CIRED, Jun. 2009.

[185] K. Kok, Z. Derzsi, J. Gordijn, M. Hommelberg, C. Warmer, R. Kamphuis, and H. Akkermans, "Agent-based electricity balancing with distributed energy resources, a multiperspective case study," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS)*. Washington, DC, USA: IEEE Computer Society, Jan. 2008.

[186] B. Roossien, M. Hommelberg, C. Warmer, K. Kok, and J. W. Turkstra, "Virtual power plant field experiment using 10 micro-CHP units at consumer premises," in *Proceedings of the CIRED Seminar SmartGrids for Distribution*. Frankfurt, Germany: IET-CIRED, Jun. 2008.

[187] Bundesamt für Sicherheit in der Informationstechnik, "Anforderungen an die Interoperabilität der Kommunikationseinheit eines intelligenten Messsystems, Technische Richtlinie BSI TR-03109-1, Version 1.0."

[188] Bundesamt für Sicherheit in der Informationstechnik, "BSI TR-03109-1 Anlage II: COSEM/HTTP Webservices."

[189] Bundesamt für Sicherheit in der Informationstechnik, "BSI TR-03109-1 Anlage I: CMS-Datenformat für die Inhaltsdatenverschlüsselung und -signatur."

[190] Bundesamt für Sicherheit in der Informationstechnik, "Schutzprofil für die Kommunikationseinheit eines intelligenten Messsystems für Stoff- und Energiemengen, Version 1.3," BSI SMGW-PP 1.3.

[191] Bundesamt für Sicherheit in der Informationstechnik, "Kryptographische Vorgaben für die Infrastruktur von intelligenten Messsystemen, Technische Richtlinie BSI TR-03109-3, Version 1.1."

[192] Bundesamt für Sicherheit in der Informationstechnik, "Public Key Infrastruktur für Smart Meter Gateways, Technische Richtlinie BSI TR-03109-4, Version 1.0."

[193] Bundesamt für Sicherheit in der Informationstechnik, "Smart Meter Gateway - Anforderungen an die Funktionalität und Interoperabilität des Sicherheitsmoduls, Technische Richtlinie BSI TR-03109-2, Version 1.1."

[194] Bundesamt für Sicherheit in der Informationstechnik, "Kryptographische Vorgaben für Projekte der Bundesregierung, Teil 3 - Intelligente Messsysteme, Technische Richtlinie BSI TR-03116-3."

[195] Bundesamt für Sicherheit in der Informationstechnik, "Schutzprofil für das Sicherheitsmodul der Kommunikationseinheit eines intelligenten Messsystems für Stoff- und Energiemengen, Version 1.02," BSI SecMod-PP 1.02.

[196] S. Feuerhahn, M. Zillgith, R. Becker, and C. Wittwer, "Implementation of an Open Smart Metering Reference Platform - OpenMUC," in *ETG-Kongress*, 2009.

[197] K. Mueller-Bier, "jDLMS," Jul. 2013. [Online]. Available: http://www.openmuc.org/index.php?id=42

[198] S. Feuerhahn and M. Buehrer, "jSML," Jun. 2014. [Online]. Available: http://www.openmuc.org/index.php?id=63

[199] Fraunhofer IWES, “Open Gateway Energy MAnagement - OGEMA,” Apr. 2014. [Online]. Available: http://www.ogema.org/

[200] Gurux Ltd., “Gurux Open Source Device Communication.” [Online]. Available: http://www.gurux.fi/

[201] KommEnergie, “Lastprofile von KommEnergie,” last visited 14.08.2015. [Online]. Available: http://www.kommenergie.de/?id=140

[202] Stadtwerke Emmendingen, “Lastprofile der Stadtwerke Emmendingen,” last visited 14.08.2015. [Online]. Available: https://swe-emmendingen.de/netz/strom-netz/lastprofile/