©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

Advanced Communication Modes for the Publish/Subscribe C-DAX Middleware

Michael Hoefling, Florian Heimgaertner, and Michael Menth
University of Tuebingen, Chair of Communication Networks, Tuebingen, Germany,
Email: {hoefling,florian.heimgaertner,menth}@uni-tuebingen.de

***Abstract*—C-DAX is a cyber-secure publish/subscribe middleware tailored to the needs of smart grids, offering end-to-end security, and scalable and resilient communication among participants in a smart grid. While C-DAX' broker-based publish/subscribe mechanisms are well-suited for scalable information dissemination with regard to high numbers of publishers and subscribers, (1) additional transmission delays are inherent to the design because of multi-hop application layer forwarding, and (2) interactive (probably legacy) applications are prohibited due to the one-way publish/subscribe paradigm and potential dependencies on IP communication. This work presents two advanced communication modes for the C-DAX architecture, addressing those issues: (1) broker-less publish/subscribe for delay-sensitive applications, and (2) transparent IP-tunneling over publish/subscribe for legacy applications. Those modes further improve C-DAX' suitability for smart grid applications, including enhanced real-time application support, and transparent support for legacy smart grid communication protocols.**

I. Introduction

Electrical power distribution networks are undergoing major changes in operational procedures and monitoring, thereby evolving from passive to *active distribution networks* (ADNs) [1]. Advanced smart monitoring tools result in faster and more reliable *real-time state estimation* (RTSE) [2]. Nevertheless, traditional mission-critical electrical power network control systems such as *supervisory control and data acquisition* (SCADA) remain important building blocks of the current and future smart electrical power grid. The main obstacles to the deployment of *smart grid* (SG) applications are the limited scalability, reliability, and security of today's utility communication infrastructures.

C-DAX [3] is a communication middleware addressing these issues by applying the *publish/subscribe* (pub/sub) [4] paradigm to the electric utility network of sensors and controls. The main motivation for the introduction of pub/sub communication in SGs is improved scalability with regard to the number of communication partners, and ease of application development [4], [5]. *Publishers* and *subscribers* are decoupled in the sense that they do not need to know each other a priori, rather a *message broker* ensures that messages are relayed between registered publishers and subscribers of a topic. In this context, a *topic* is an abstract representation of a unidirectional information channel, addressed by its unique name and possibly attributes.

The main contribution of this paper is a detailed presentation of two advanced communication modes for the C-DAX architecture: (1) broker-less publish/subscribe for delay-sensitive applications, and (2) transparent IP-tunneling over publish/subscribe for legacy applications. While C-DAX' broker-based publish/subscribe mechanisms are well-suited for scalable information dissemination, additional transmission delays are inherent to the baseline design, and interactive applications are prohibited by the pub/sub paradigm. The new communication modes address those issues and further improve C-DAX' suitability for smart grid applications, including enhanced real-time application support, and transparent support for legacy smart grid communication protocols.

This work is structured as follows. We review selected examples for current and future SG applications in Section II, and give an overview of C-DAX in Section III. In Section IV, we describe the advanced communication modes and show how SG applications can benefit from them. We discuss related work in Section V, and draw conclusions in Section VI.

II. Examples of Smart Grid Applications

We review SCADA and synchrophasor-based RTSE as examples of SG applications. SCADA is one of the most-widely deployed legacy SG applications in today's electrical power grids and will remain an important building block of future smart electrical power networks, too. In contrast, RTSE represents a potential future SG application which is already deployed on the transmission grid level in some countries [6], [7], and which may be massively deployed on the distribution grid level in the future. We briefly introduce each application and summarize its respective communication requirements.

A. Supervisory Control and Data Acquisition

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. The currently widely-used communication standard IEC 60870-5-104 [8] defines *remote terminal units* (RTUs) which are deployed at the substations, and which communicate over TCP/IP with a SCADA *master control* and other systems in the utility's *distribution control center* (DCC). 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.

978-1-5090-0223-8/16/$31.00 © 2016 IEEE
IFIP/IEEE Workshop on the Management of Fog Computing and the Internet of Things (ManFIoT 2016) collocated with the IEEE/IFIP Network Operations and Management Symposium (NOMS 2016), April 2016, Istanbul, Turkey

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 IV how C-DAX enables transparent integration of legacy applications such as SCADA.

B. *Synchrophasor-Based Real-Time State Estimation of Active Distribution Networks*

The lack of distributed measurement infrastructures at the distribution grid level is one of the main obstacles for distribution network operators to develop adequate controls capable of enabling the seamless integration of distributed energy resources. One of the most promising ADN monitoring technologies is synchrophasor-based RTSE [1], [2]. The base components of this technology are *phasor measurement units* (PMUs) and *phasor data concentrators* (PDCs). PMUs measure the equivalent phasor representation of the power-system waveforms at different points of the power grid. The measurement data are accurately time-stamped using a reliable time source, and sent to the PDC with a refresh rate up to 50 times per second [9]. 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. The time-aligned and aggregated measurement data is fed into a mathematical model of the distribution grid to estimate the current state of the grid. The outcome of the estimation may be used by several power-system applications, 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 distribution network operators a complete and real-time view and control of their ADNs.

We proposed an adapter-based solution to easily connect and integrate entities in a synchrophasor network over a pub/sub communication architecture in [10]. Even though this solution works satisfactory in a controlled communication network environment, the mandatory intermediary message broker in a broker-based pub/sub communication architecture causes additional delay on the communication path between PMUs and PDCs which might distort the estimation result. We will, therefore, introduce an enhancement to the existing data streaming mode in C-DAX for delay-sensitive SG applications in Section IV.

III. C-DAX: A Cyber-Secure Data and Control Cloud for Power Grids

C-DAX (*Cyber-secure Data And Control Cloud for power grids*) [3] is an FP7 project funded by the European Commission which adapts the pub/sub paradigm to the needs of power grids. It aims at developing a cyber-secure and

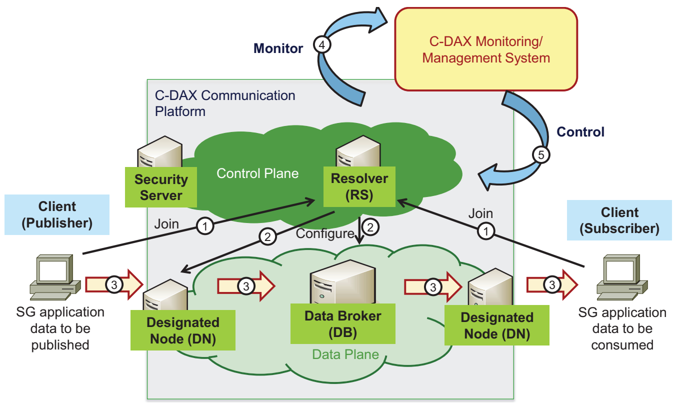


Fig. 1. The C-DAX architecture. Basic signaling steps include client join (step 1), data plane configuration (step 2), and topic data transmission (step 3). Further signaling includes monitoring (step 4) and general control of the C-DAX cloud (step 5).

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. Further information on C-DAX' resilience concept and its security architecture can be found in [11] and [12].

A. *Components*

Figure 1 illustrates the basic structure and interactions of the C-DAX architecture. It is composed of C-DAX clients and the C-DAX cloud, and organizes information in *topics*. A topic is an abstract representation of a unidirectional information channel, addressed by its unique name and probably attributes, e.g., phasor measurement data for a specific geographic region inside the distribution grid. SG applications use *C-DAX clients* as interface to the C-DAX cloud, which handle all C-DAX signaling transparent 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. We briefly describe the functions from bottom to top, and assign them to their respective plane, e.g., data, control, or management plane.

1) Data Plane: Designated nodes (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. *Data brokers (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.

2) Control Plane: Resolvers (RSes) hold the topic-to-DB mappings and provide a resolution interface for other nodes. There may be several RSes in a C-DAX cloud, e.g., for resiliency or extensibility reasons. Security-related functionalities are provided by a *security server (SecServ)*, e.g., authentication, authorization, and key distribution.

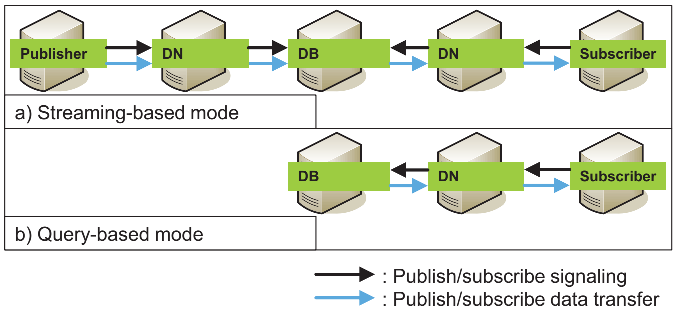


Fig. 2. Communication modes of C-DAX. Streaming-based (a) and query-based mode (b) are part of the initial C-DAX specification.

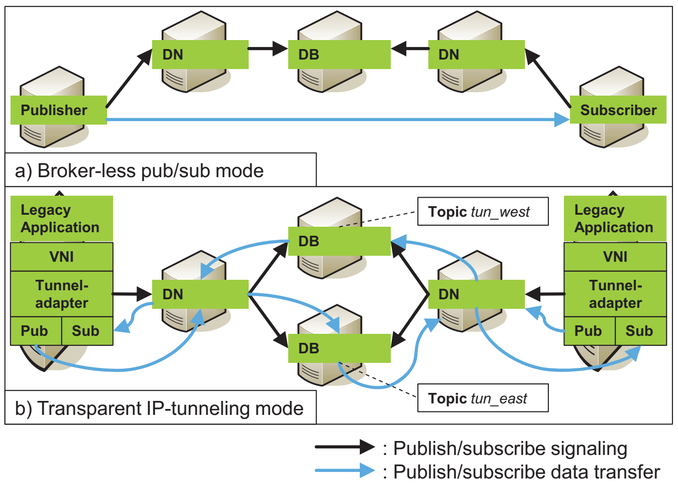

DN
DB
DN
Publisher
Subscriber
a) Broker-less pub/sub mode
Legacy Application
VNI
Tunnel-adapter
Pub
Sub
DB
Topic tun_west
DN
DN
DB
Topic tun_east
b) Transparent IP-tunneling mode
: Publish/subscribe signaling
: Publish/subscribe data transfer

Fig. 3. 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.

3) Management Plane: Management is provided by the *management system (MgmSys)*, which 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. Node management enables addition and removal of a C-DAX node from the cloud. Monitoring is provided by the *monitoring system (MonSys)*, which provides mechanisms to gather, and aggregate monitoring information.

B. Basic Interactions

1) Publication of Topic Data: Initial message exchange prior to topic data publication is shown on the left side of Figure 1. When the publisher wants to publish topic data, it first sends a join message over its DN to the RS 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).

2) Subscription to Topic Data: Topic data retrieval works similarly. Initial message exchange prior to topic data retrieval is shown on the right side of Figure 1. When the subscriber wants to retrieve topic data, it first sends a join message over its DN to the RS 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).

C. Communication Modes

The initial C-DAX specification contains only two communication modes which are illustrated in Figure 2: streaming-based and query-based communication. In *streaming-based mode* (see Figure 2a), subscribers continuously receive topic data after successfully joining a topic without requiring further explicit requests. In *query-based mode* (see Figure 2b), subscribers have to send explicit topic data queries to fetch specific topic data, e.g., a snapshot of streamed data.

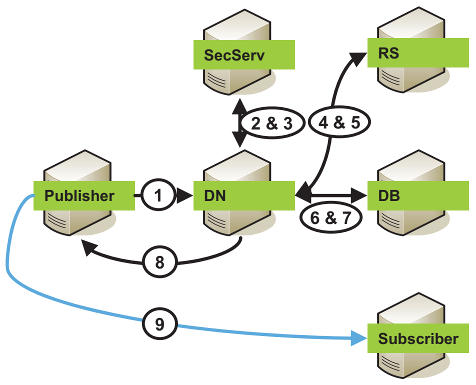
IV. Advanced Communication Modes

Future SG applications can be designed specifically for the pub/sub paradigm or can be adapted to it [10]. 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*.

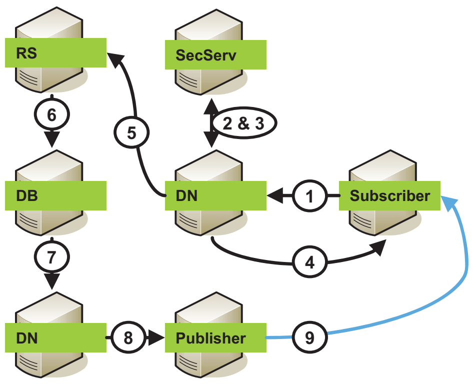
A. Broker-less Pub/Sub Mode

In *broker-less pub/sub mode* (see Figure 3a), 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.

1) 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. Additional 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



(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 from the publishers.

Fig. 4. Basic signaling for broker-less pub/sub mode.

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.

2) Basic Signaling: We assume that publishers and subscribers join a broker-less topic in the following. As depicted in Figure 3a, join-specific signaling is sent over the C-DAX cloud whereas the actual data transmission takes place between publishers and subscribers only.

a) Publisher-Join Signaling: When a publisher joins a broker-less topic, as shown in Figure 4a, 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.

b) Subscriber-Join Signaling: Subscriber join signaling works similarly. When a subscriber joins a broker-less topic, as shown in Figure 4b, 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).

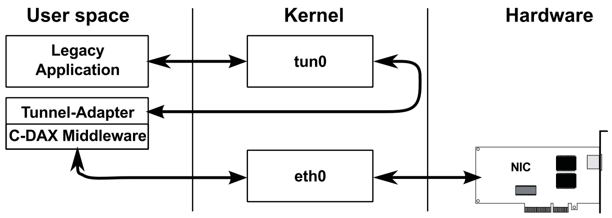


Fig. 5. Information exchange and signal flow for transparent IP-tunneling over virtual and physical network interfaces.

B. Transparent IP-Tunneling Mode

In *transparent IP-tunneling mode* (see Figure 3b), 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.

1) 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 3b 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.

2) Implementation and Configuration: The prototype implementation is based on the Linux tun/tap [13] 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 5 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 an entry for the remote IP prefix with the respective remote endpoint as gateway.

C. 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. 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 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, as long as IP communication is supported.

V. Related Work

Cugola et al. [14] investigate several options for adding replies to pub/sub communication to enable remote procedure calls (RPCs) over pub/sub, which requires bidirectional communication. They propose a few protocols which may be implemented on top of any pub/sub architecture, utilizing either out-of-band signaling or in-band signaling. One of their proposed in-band signaling protocols utilizes pub/sub messages as replies which is comparable to C-DAX' transparent IP-tunnel mode.

The OMG Data-Distribution Service (DDS) [15] is a pub/sub architecture which targets fault-tolerant real-time communication. A prominent user of DDS in the SG area is NASPInet [6]. Direct point-to-point connections between publishers and subscribers yield minimal delay and latency, i.e., no brokers are involved in the data communication. Discovery of publishers and subscribers is handled via a distributed architecture comparable to C-DAX' RS component. DDS' data transmission is comparable with C-DAX' broker-less mode.

GridStat [16] is a broker-based pub/sub middleware for wide-area monitoring systems and was one of the candidate architectures for NASPInet. By default, GridStat only supports broker-based unidirectional communication; a broker-less communication mode is not available. Vidall et al. [17] propose the 2WoPS protocol (*2 Way over Publish-Subscribe*) to extend GridStat with bidirectional communication capabilities. In combination with the Ratatoskr framework, 2WoPS facilitates QoS-managed RPCs over wide-area networks. This is comparable to C-DAX' transparent IP-tunnel mode.

The SeDAX [5] architecture uses geographic routing on an overlay network to forward messages to responsible brokers. An extension to support distributed load balancing has been proposed in [18], and can be used to optimize end-to-end delays for real-time applications. SeDAX is limited to broker-based communication. Other pub/sub architectures such as DataTurbine [19], MQTT [20], and RabbitMQ [21] use a similar approach and are purely broker-based pub/sub architectures.

The ZeroMQ [22] high-performance asynchronous messaging library provides a message queue for scalable distributed and concurrent applications. By default, ZeroMQ is a broker-less system but instructions for implementation of brokers are available at [23]. 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. The authors of ZeroMQ discuss and propose the use of brokers as forwarding entities and/or directory services for publisher/subscriber discovery in [24] to enable broker-based and broker-less communication in ZeroMQ. In this respect, ZeroMQ brokers provide the same functionalities as RSes and DBs in C-DAX. Compared to C-DAX, ZeroMQ provides a similar level of flexibility to adapt its data plane to different use case communication requirements.

The Java Message Service (JMS) [25] can be operated in both broker-less and broker-based mode, and includes methods for bidirectional communication. The actual realization of those functionalities depends on the underlying implementation and the configuration thereof.

VI. Conclusion

The C-DAX project aims at providing and investigating a communication middleware for SGs to address the limited scalability, reliability, and security of today's utility communication infrastructures. In this paper, we introduced two advanced communication modes for C-DAX to better cope with the needs of SG applications: (1) broker-less pub/sub mode, and (2) 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. Furthermore, the transparent IP-tunneling mode is implemented in the C-DAX prototype and will be used alongside the general streaming mode in a field trial in the Alliander LiveLab [26] SG test site.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7-ICT-2011-8 under grant agreement n° 318708 (C-DAX). The authors alone are responsible for the content of this paper.

The authors thank Roelof Klein, Martijn Kammerling, Paolo Romano, Jimmie Adolph, and Vidar Gronas for valuable input and stimulating discussions.

References

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

[2] 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.

[3] C-DAX Consortium, "Cyber-secure Data And Control Cloud for Power Grids," 2015. [Online]. Available: http://www.cdax.eu/

[4] 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.

[5] Y.-J. Kim, J. Lee, G. Atkinson, H. Kim, and M. Thottan, "SeDAX: A Scalable, Resilient, and Secure Platform for Smart Grid Communications," *IEEE JSAC*, vol. 30, no. 6, 2012.

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

[7] 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.

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

[9] IEEE, "IEEE Standard for Synchrophasor Measurements for Power Systems," IEEE C37.118.{1,2}-2011, Dec. 2011.

[10] 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.

[11] 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.

[12] 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.

[13] M. Krasnyansky and M. Yevmenkin, "Universal tun/tap device driver," 2007. [Online]. Available: http://vtun.sourceforge.net/tun/

[14] G. Cugola, M. Migliavacca, and A. Monguzzi, "On Adding Replies to Publish-Subscribe," in *Inaugural Conference on Distributed Event-Based Systems (DEBS)*, 2007.

[15] G. Pardo-Castellote, "OMG Data-Distribution Service: Architectural Overview," in *ICDCS Workshops*, 2003.

[16] 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.

[17] E. S. Viddal, D. E. Bakken, K. H. Gjermundrød, and C. H. Hauser, "Wide-Area Actuator RPC over GridStat with Timeliness, Redundancy, and Safety," in *International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, 2010.

[18] 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.

[19] 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.

[20] A. Stanford-Clark and A. Nipper, "MQ Telemetry Transport," 2014. [Online]. Available: http://www.mqtt.org/

[21] A. Videla and J. J. W. Williams, *RabbitMQ in Action: Distributed Messaging for Everyone*. Shelter Island NY: Manning, 2012.

[22] P. Hintjens, *ZeroMQ: Messaging for Many Applications*. O'Reilly Media, Inc., 2013.

[23] ——, "ZeroMQ - The Guide," 2013. [Online]. Available: http://zguide.zeromq.org/page:all

[24] iMatix Corporation, "ZeroMQ: Broker vs. Brokerless," 2012. [Online]. Available: http://zeromq.org/whitepapers:brokerless

[25] Oracle Corporation, "Java Message Service 2.0 Released," 2013. [Online]. Available: http://www.oracle.com/technetwork/java/jms/index.html

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