An Extensible Software and Communication Platform for Distributed Energy Resource Management

Gabe Fierro
Dept. of Electrical Engineering and Computer Science
University of California, Berkeley
Berkeley, USA
gtfierro@cs.berkeley.edu

Keith Moffat
Dept. of Electrical Engineering and Computer Science
University of California, Berkeley
Berkeley, USA
keithm@berkeley.edu

Jasper Pakshong
Dept. of Civil and Environmental Engineering
University of California, Berkeley
Berkeley, USA
jasperpak@berkeley.edu

Alexandra von Meier
California Institute for Energy and Environment
Dept. of Electrical Engineering and Computer Science
University of California, Berkeley
Berkeley, USA
vonmeier@berkeley.edu

***Abstract*—This paper introduces a novel Distributed Extensible Grid Control (DEGC) software and communication platform to facilitate the control of distributed energy resources on electric grids. The DEGC software platform leverages state-of-the-art advances in secure, distributed communication and decentralized authorization and authentication. We discuss how these advances enable the kind of robust and secure communication required for a distributed grid control platform, and show how DEGC applies these technologies to the agile development and deployment of grid software through an extensible and flexible API. We describe how DEGC can implement both Volt-VAR voltage magnitude control and Phasor-Based Control as sample applications and demonstrate the DEGC platform in hardware with the demanding Phasor-Based Control test case, and provide performance metrics.**

***Index Terms*—distributed control, communication, grid security, phasor-based control.**

I. Introduction

The intelligent recruitment of diverse distributed energy resources (DER) is a vital component of electric grid modernization. As installed capacity of solar photovoltaic (PV) generation, electric vehicle (EV) charging and distributed energy storage grows, the focus must shift from mitigating adverse impacts of intermittent power injections—often by constraining permissible interconnections—to leveraging controllable DER to *support* the grid. However, coordinating spatially distributed resources throughout an electric grid in a secure and scalable manner is nontrivial.

A software and communication platform that supports DER integration must address the following three challenges: 1) DER heterogeneity: there are many types of DER (e.g. PV, battery energy storage, and EVs), and many different products/vendors; 2) Communication network inconsistency: Many different methods are used to communicate with DER. The communication networks cannot be presumed to be 100% reliable; 3) Security concerns: The electric grid is critical infrastructure for modern society. Thus, any distributed control platform is a potential target for cyberattacks.

The best way to control and coordinate DER continues to be an active field of research. Communication platforms for grid monitoring and control must ensure the privacy and integrity of all transmitted data, in addition to providing scalable mechanisms for managing and auditing access to information and actuators [1]–[3]. The increased penetration of distributed controllers and DER offer an opportunity to shift away from traditional one-way communication and towards more flexible, bi-directional flows of information [4], [5].

Recent research establishes that message buses are an effective solution for dynamic and scalable communication between distributed resources [6]–[8]. However, this knowledge has resulted in few published implementations, of which VOLTTRON [9] is the most prominent. Such solutions largely do not address reliable delivery and often rely upon central authorities for handling permissions.

In this paper we introduce the Distributed Extensible Grid Control (DEGC) platform, which combines recent advances in message buses and decentralized security to implement a Python-based software framework which facilitates the development and deployment of grid monitoring and control. DEGC incorporates a publish-subscribe message bus [10] which allows distributed devices and processes to communicate securely without being aware of the communication network topology. DEGC also incorporates the WAVE [11] authorization and authentication platform to protect devices,

978-1-7281-6127-3/20/$31.00 ©2020 IEEE

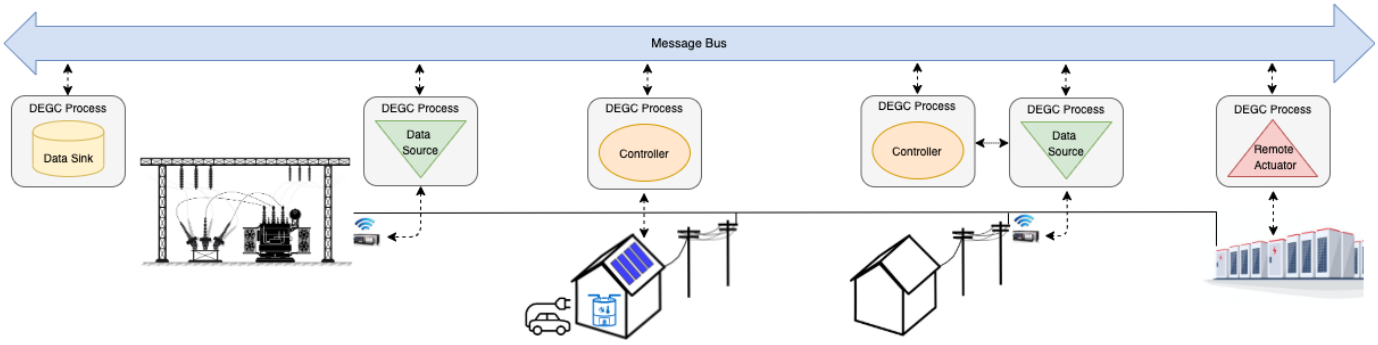

Fig. 1. Logical architecture of the platform, showing how distributed processes communicate through the message bus, and how the processes interact and interface with DERs

processes and other distributed resources. The distributed nature of WAVE helps to scale the commissioning process while providing flexible configuration of access to data sources and controllable resources. DEGC provides simple abstractions over these features to implement a secure, agile, and extensible distributed control platform. The novelty of the platform lies in the incorporation of state-of-the-art communication and security technologies into an operable system with a flexible API.

DEGC's extensibilty provides a broad application space. The DEGC platform can be applied to any layer of the transmission-distribution spatial hierarchy, and used to implement the control decisions, communications, and optimizations required for all of the layers of the traditional primary/secondary/tertiary control temporal hierarchy [12]. The DEGC application space is general, though, and not limited to traditional grid control distinctions [13].

In this paper, we focus on DER control in the distribution network context, which clearly demonstrates DEGC's strengths. The most prominent control technique for DER voltage magnitude control is distributed Volt-VAR control [14], in which a centralized optimization determines the Volt-VAR curve setpoints, and the DER implement the Volt-VAR curve using local measurements [15], [16]. However [17] demonstrates that a system of Volt-VAR voltage magnitude controllers that use only local information can fail to maintain the voltage at all locations in the network, despite having sufficient actuation capabilities.

Another example of distributed control is Phasor-Based Control (PBC), a novel control framework described in [18] which assigns voltage phasor (magnitude and angle) targets to distributed controllers located at strategic locations on a distribution grid. The distributed controllers track the phasor targets using feedback control based on measurements produced by phasor measurement units (PMU) at both the local and reference nodes. PBC is an exceptionally demanding distributed control method because it requires low-latency alignment of microsecond-accuracy voltage phasor angle measurements between remote locations.

The Distributed Extensible Grid Control (DEGC) platform described in this paper provides the distributed control research community with a software platform that facilitates the implementation of general distributed grid control systems. DEGC consists of a secure, extensible communication platform (§II) and an extensible software platform (§III). Sensors, actuators, controllers and other processes communicate with one another by sending messages via a shared publish-subscribe message bus. The software platform defines high-level APIs that enable the flexible and robust implementation of arbitrary grid controllers (§IV). The platform has been used to implement PBC [18] in a real testbed deployment, and demonstrated delivery of 99.9% of all messages (§IV).

II. Secure, Distributed Extensible Grid Control Communication Platform

While previous research has addressed the design and implementation of communication platforms for the grid [5], [6], [19], the question of how these secure and distributed platforms impact the construction and deployment of real grid control has largely been left unanswered. Below, we present the design of the DEGC communication platform in the context of established and emerging security and communication requirements for grid control platforms.

A. Architecture

Figure 1 presents the logical architecture of the platform, which consists of distributed, persistent processes communicating over a shared publish-subscribe message bus. Processes abstract away the diversity of physical location and networked interfaces for the family of DER, devices, equipment, systems, cloud services and other cyberphysical and networked "things" in an electric grid. The message bus allows processes to interact with remote devices, services and data sources as easily as if they were local, without having to configure network access or translate between different protocols. This drastically simplifies the development and configuration of grid-centric controls and analytics.

B. Security Requirements

It is well established [1] that the primary security properties of a grid communication platform include **availability** (information should be made available in a timely and reliable manner), **integrity** (information should not be modified while

in transit) and **confidentiality** (information should be protected and only made available to authorized parties and processes).

Established solutions for authentication, authorization, data privacy and data integrity often depend on trusting centralized authorities and services. A central service is a central point of attack [20] and can also view and manipulate all permissions [21]. For example, if a distribution system operator administers time-varying prices to participating grid assets, this operator should not have access to information exchanged among privately-owned assets, such as battery state-of-charge messages exchanged between members of a virtual power plant. The presence of a centralized authority with access to all permissions makes some distributed control strategies less viable. Omnipotence on the part of any entity creates cyber-vulnerabilities as well as opportunities for conflicts of interest.

The recently published WAVE framework for decentralized authorization offers a path forward [11]. WAVE allows *entities* (including devices, software processes and people) to grant and revoke access to *resources* (including API endpoints, data sources and actuators) without relying upon a central authority. We adapt WAVE to the electric grid domain as follows. We define the following entities as WAVE principals (the entities that can be authenticated and authorized):

- Grid-based data sources, characterized by measurement and monitoring equipment such as PMUs, SCADA remote terminal units, and smart meters
- Actuators, including any controllable device that can change the voltage or power flows on the grid, including power electronic devices, switches, capacitor banks, and voltage regulators
- Software processes, characterized by instances of controllers, analytics and alarm systems
- People, such as grid operators, market managers, technicians and engineers

Through WAVE, each entity can be granted the ability to take certain actions (e.g. the ability to read data from a certain source or the ability to send a command to a certain device) on certain resources—these are called *permissions*. Permissions are kept encrypted and are only visible to authorized entities. WAVE resources are hierarchically structured labels that represent data sources (such as synchrophasor (PMU) sensors) and data sinks (such as API endpoints).

Adopting the WAVE authorization framework provides a secure, decentralized base for the communication platform, described below, and the grid control software platform, described in §III. It is able to provide the key security properties described above and can do so in a manner that respects the administrative constraints of electric grids.

C. Communication Requirements

The message bus implements a publish-subscribe communication paradigm in which discrete messages are passed between loosely-coupled data producers and data consumers [22]. Data consumers register an interest in one or more data resources; data producers broadcast messages associated with a data resource; the message bus routes messages from data producers to consumers.

The publish-subscribe approach is a notable departure from existing, centralized systems for grid monitoring and control, such as SCADA, because it *decouples* data producers and data subscribers—data producers do not know which processes are subscribed to them. This decoupling simplifies the reconfiguration of communication paths, and contributes to the scalability of the system. Because the message bus handles the distribution of messages, a single (popular) data producer does not have to manage hundreds or thousands of simultaneous connections in order to transmit its data to all interested parties; it simply delivers its data over a single connection to the message bus. This also simplifies the data consumer. It does not need to reach across firewalls and perform complex handshakes with a diverse array of producers; instead, it simply registers its interests with the message bus, which routes the correct messages to the data consumer.

The DEGC communication platform builds on WAVEMQ, a message bus which implements authentication and authorization with WAVE [10], [23]. Thus, data resources, data consumers and producers in DEGC inherit the same privacy and decentralized properties as WAVE. All interaction between resources, processes and users in the system is performed through the message bus and is protected by WAVE. This allows the message bus to act as a layer of protection against DOS attacks and other network-based attacks: WAVEMQ blocks all traffic that does not provide sufficient authorization before it reaches sensitive processes or resources.

WAVEMQ implements a "tiered" architecture: local instances of the message bus perform routing within a reliable local area network, corresponding to a single deployment site. This ensures that processes within a site can continue to communicate even if their connection to remote resources is disrupted. The local instance not only performs all necessary networking to the rest of the message bus, but also handles all necessary encryption, decryption and validation for all traffic routed in and out of a site.

D. Challenges for Distributed Grid Control

Adopting a publish-subscribe communication platform with decentralized authorization provides the necessary flexibility and security for distributed grid control, but also presents several implementation challenges.

Handling Intermittent Connectivity: Network connectivity in real deployments can fail unexpectedly and for extended periods of time. "Reliable delivery" as a property of a communication system does not necessarily mean that data is delivered promptly—only that it is delivered eventually. WAVEMQ instances buffer data when they lose connectivity, and redeliver that data when connectivity is restored. However, this data could be hours or even days old depending on the length of the connectivity outage. If processes that consume data are not careful, they might mistakenly treat old data as recent data. A software platform for distributed grid control must provide processes with a way to reason about the

difference in time from when a message was first published to the message bus and when that message was received.

Handling Asynchronous Communication: The decoupled nature of publish-subscribe systems presents a challenge for traditional approaches to implementing grid control algorithms. Many control loops are written "synchronously"—that is, they repeatedly execute a set of sequential steps that are each run to completion before the next step begins. For example, a generic control loop may first read some sensor data, perform some computation, produce a control decision, and send that control decision to relevant actuators before beginning again.

Unlike the usual point-to-point or client-server regime in which a process initiates and maintains a persistent connection with another process, processes in a publish-subscribe system do not communicate directly. This means that incoming messages may arrive at any time, out of sync with the usual sequential progression of a control loop. A software platform for distributed grid control must provide processes with a way to implement familiar control loops over multiple concurrent subscriptions which may deliver data asynchronously.

III. Distributed Extensible Grid Control Software Platform

The DEGC software platform abstracts the components of distributed grid control as persistent, networked computational entities called *processes*. A process interacts directly with the WAVEMQ message bus and contains an instance of an **application class**, of which there are four main flavors: Controller, Remote Actuator, Data Source and Data Sink.

All DEGC processes interact with each other by publishing and subscribing on the WAVEMQ message bus. This ensures that all messages are encrypted, delivered reliably, and are only made available to authorized entities. By building on WAVEMQ, DEGC processes only ever receive authorized and authenticated messages. This essentially eliminates the need for a process to implement its own access control.

We discuss how the DEGC software platform is made extensible through a layered approach to the implementation of a process before exploring how the software platform enables the four application classes described above.

A. Architecture of a DEGC Process

A DEGC Process adopts the layered structure described in Figure 2. Each layer provides an API that abstracts unnecessary details away from software written against the API. The implementation of these layers, and the APIs they define, constitute the DEGC software platform. This design is extensible because it allows new kinds of processes to be developed without changing the architecture or implementation of the rest of the system.

The **communication layer** is a software library that implements all communication with the message bus and defines the Application Programming Interface (API) for the process runtime. All logic in a DEGC process is ultimately scheduled by calls to this API.

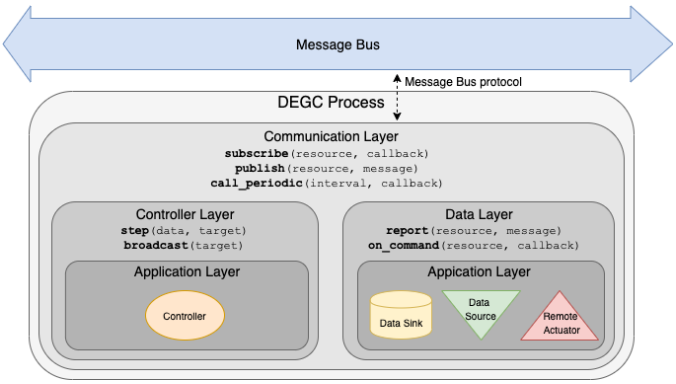

Fig. 2. The architecture of a DEGC process. Each layer lists its respective API.

- **`subscribe`**`(resource, callback)`: registers a subscription to data delivered on the given resource; when messages are received, trigger the function `callback` with the received message as an argument
- **`publish`**`(resource, msg)`: publishes the given message on the given resource
- **`call_periodic`**`(interval, callback)`: schedules the function `callback` to be called at a regular interval. This allows processes to perform computation that is not tied directly to the receipt of a message

The **controller layer** and **data layer** wrap the communication layer's API to implement additional logic useful for constructing controllers and data-oriented processes. The `on_command` function is essentially identical to the `subscribe` function defined by the communication layer, but contains additional logic to filter out "old" commands that were delayed due to buffering in the message bus.

- **`broadcast`**`(target)`: publishes a computed target as input for other controllers
- **`step`**`(data, targets)`: implements one controller iteration
- **`report`**`(resource, msg)`: publishes sensor data, equipment status on the given resource; ensures timestamps are associated with reported data
- **`on_command`**`(resource, callback)`: triggers the provided `callback` function when a command message is received on the given resource; used to invoke actuation of local DER.

The **application layer** consists of application, device, and controller-specific logic that makes use of the APIs defined above to implement one or more of the DEGC process roles: controller, remote actuator, data source and data sink/processor.

B. DEGC Application Classes

The four (primary) DEGC application classes are Data Sources, Data Sinks, Remote Actuators and Controllers. Via the data layer API, these classes can publish sensor, actuator and other kinds of data on the message bus and to respond to data published on the message bus.

Data Sources interface with sensors and equipment such as PMUs, batteries, inverters and loads, and publish their current values and statuses on the message bus periodically (see Figure 2). Data sources define the set of *resources* for a particular sensor or equipment. These resources represent different data channels (such as phases of a PMU or the names of specific loads), and can be subscribed to individually.

Data Sinks subscribe to data from many resources and perform computation on that data. An important example in Figure 2 is the "data historian", which logs all published data in a timeseries database for later retrieval.

Remote Actuators interface with equipment such as inverters and controllable loads. They define and subscribe to a set of resources representing the inputs to the underlying equipment. When Remote Actuators receive command messages that are published on the bus, they can choose to implement the desired action on the equipment they represent.

Controllers use the controller layer API to implement grid control. Unlike the other API calls which are implemented by the platform, `step` contains user-defined control logic. The user-defined control logic allows developers to implement any type of controller.

The DEGC process runtime periodically calls `step` on a configurable interval (this is implemented with `call_periodic`). DEGC provides each invocation of `step` with (1) a buffer of all sensor/equipment data that was received since the last time `step` was called, and (2) a list of the most recently received controller targets. The particular data streams and controllers whose outputs are included in the buffer and provided to `step` are determined by the configuration of the controller. The return value of `step` is the status of the DEGC process after applying the targets contained in the call to `step`; the controller layer automatically publishes the status on the message bus.

IV. Implementing Grid Control Frameworks

A. Application to Voltage Control

To illustrate the flexibility of DEGC, consider three distinct approaches to voltage control in distribution systems that recruit DER. In decentralized voltage control, resources adjust reactive power output on a Volt-VAR droop curve based on strictly local voltage magnitude measurement [17], [24]. Here, local controllers would run all optimization computations internally, and communicate their status via the publish-subscribe message bus.

Controllers may also receive instructions for updated parameters of the Volt-VAR curve via DEGC. In a more centralized version of Volt-VAR control, a single supervisory controller computes the droop curves for each local controller/actuator pair. Such a schema is readily implemented in the DEGC platform by defining two Controller Processes: one supervisory controller per network, and as many distributed controllers as there are control nodes. The latter command their respective Remote Actuator processes based on local measurements, which are in turn provided by Data Source processes that each publish a single voltage magnitude measurement.

In a further refinement, phasor-based control (PBC) [18] extends to both voltage magnitude V and phase angle δ, which constitute the voltage *phasor* at any given network node, using phasor measurement units (PMUs) that provide ultra-precise 120-Hz data [25], [26]. In PBC, a supervisory controller computes voltage phasor targets through optimization, and broadcasts these targets to local controllers that drive DER toward maintaining their target using both real and ractive power. This control framework demands a particularly sophisticated and secure information infrastructure if it is to include many independent, geographically separated participants.

B. Deployment Demonstration

The DEGC platform was deployed to test PBC using real hardware. The Hardware-in-the-Loop (HIL) setup consisted of a grid simulator, solar PV and battery-connected inverters, and load banks acting as the Remote Actuators. Controllers and sensors were added accordingly to implement PBC algorithms. DEGC processes supported the Controller, Data Source, and Data Sink application classes of the setup, consisting of one supervisory controller, multiple distributed controllers, four PMUs, and the data historian, respectively. We note that the Remote Actuators did not communicate via WAVEMQ due to restrictions of the actuator hardware itself.

Table I provides an overview of the kinds of process deployed as part of the PBC demonstration, the number of each process type deployed, the number of channels each process type subscribes to and publishes on, as well as the associated rates of messages sent or received each minute. For context, we also provide numbers of a corresponding Volt-VAR example setup.

C. Communication Platform Performance and Reliability

To measure the performance of the DEGC communication platform we monitored the deployment described above for 27 days. Over this time, the message bus (WAVEMQ) delivered 354,254,138 messages, of which only 353,967 were dropped for any reason—this means that over 99.9% of messages were delivered successfully. The count of dropped messages includes those that were buffered for redelivery to processes that never came back online, so the 99.9% statistic is an underestimate of the actual reliability of the system.

We also conducted experiments to measure the latency of messages routed through the communication platform. We configured one process on one server to periodically publish a message, which was received and then re-published by another process on another server. The first process measures the amount of time that elapses between sending the first message and receiving the reply; we can estimate the one-way communication latency by dividing this number in half. Over 100,000 messages, we observed a median latency of 8.7ms and a 99th percentile latency of 10.6ms.

While a more in-depth evaluation of the subject is beyond the scope of this paper, these preliminary numbers demonstrate that DEGC, and its components WAVEMQ and WAVE, are both reliable and performant.

Control Framework	Process Type	Application Class	# Deployed	# Sub Channels	Msg/min (Recv)	# Pub Channels	Msg/min (Send)
PBC	PMU Driver	Data Source	4	0	0	6	360
	Distributed Ctrl	Controller	3	13	721	1	60
	Supervisory Ctrl	Controller	1	9	540	3	3
	Data Historian	Data Sink	1	30	1623	0	0
Volt-VAR	Vmag Sensor	Data Source	3	0	0	3	180
	Distributed Ctrl	Controller	3	4	181	1	60
	Supervisory Ctrl	Controller	1	3	180	3	3
	Data Historian	Data Sink	1	15	723	0	0

TABLE I
AN ENUMERATION OF THE PROCESSES DEPLOYED AS PART OF THE PBC DEMONSTRATION AS WELL AS AN EXAMPLE OF A CORRESPONDING VOLT-VAR SETUP, WITH HOW MANY SUBSCRIPTIONS AND MESSAGES ARE EXCHANGED BETWEEN THEM

V. CONCLUSION

We have described a novel communication platform, DEGC, that meets the security and scalability requirements of advanced schemes for controlling distributed energy resources. It is based on the WAVE framework for decentralized authorization, which enables diverse actors to securely participate in processes involving sensitive grid information and control actuation. DEGC was implemented in HIL testing, where it exhibited high reliability and short latencies. The platform is available for download at https://github.com/gtfierro/DEGC.

ACKNOWLEDGMENT

This work was supported by the U.S. Department of Energy, SETO, Award DE-EE0008008.

REFERENCES

[1] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer networks*, vol. 57, no. 5, pp. 1344–1371, 2013.

[2] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012.

[3] J. Liu, Y. Xiao, S. Li, W. Liang, and C. P. Chen, "Cyber security and privacy issues in smart grids," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 981–997, 2012.

[4] D. B. Rawat and C. Bajracharya, "Cyber security for smart grid systems: Status, challenges and perspectives," in *SoutheastCon 2015*. IEEE, 2015, pp. 1–6.

[5] Y.-J. Kim, M. Thottan, V. Kolesnikov, and W. Lee, "A secure decentralized data-centric information infrastructure for smart grid," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 58–65, 2010.

[6] M. Albano, L. L. Ferreira, L. M. Pinho, and A. R. Alkhawaja, "Message-oriented middleware for smart grids," *Computer Standards & Interfaces*, vol. 38, pp. 133–143, 2015.

[7] S. G. C. SI, "Smart grid reference architecture," CEN-CENELEC-ETSI, Tech. Rep., 2012. [Online]. Available: https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf

[8] S. Rusitschka, K. Eger, and C. Gerdes, "Smart grid data cloud: A model for utilizing cloud computing in the smart grid domain," in *2010 First IEEE International Conference on Smart Grid Communications*, 2010, pp. 483–488.

[9] J. Haack, B. Akyol, N. Tenney, B. Carpenter, R. Pratt, and T. Carroll, "Volttron™: An agent platform for integrating electric vehicles and smart grid," in *2013 International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 2013, pp. 81–86.

[10] M. P. Andersen, "Decentralized authorization with private delegation," Ph.D. dissertation, EECS Department, University of California, Berkeley, Aug 2019. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-113.html

[11] M. P. Andersen, S. Kumar, M. AbdelBaky, G. Fierro, J. Kolb, H.-S. Kim, D. E. Culler, and R. A. Popa, "WAVE: A decentralized authorization framework with transitive delegation," in *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 1375–1392. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/andersen

[12] J. D. Glover, M. S. Sarma, and T. Overbye, *Power system analysis & design, SI version*. Cengage Learning, 2012.

[13] F. Dörfler, J. Simpson-Porco, and F. Bullo, "Breaking the hierarchy: Distributed control & economic optimality in microgrids, 2014," *URL http://arxiv. org/pdf/1401.1767 v1. pdf,(under review)*.

[14] J. Smith, "Modeling high-penetration pv for distribution interconnection studies," *Electric Power Research Institute, Tech. Rep*, 2013.

[15] F. Ding, A. Nagarajan, S. Chakraborty, M. Baggu, A. Nguyen, S. Walinga, M. McCarty, and F. Bell, "Photovoltaic impact assessment of smart inverter volt-var control on distribution system conservation voltage reduction and power quality," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2016.

[16] M. Farivar, C. R. Clarke, S. H. Low, and K. M. Chandy, "Inverter var control for distribution systems with renewables," in *2011 IEEE international conference on smart grid communications (SmartGridComm)*. IEEE, 2011, pp. 457–462.

[17] S. Bolognani, R. Carli, G. Cavraro, and S. Zampieri, "On the need for communication for voltage regulation of power distribution grids," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 1111–1123, 2019.

[18] A. von Meier, E. L. Ratnam, K. Brady, K. Moffat, and J. Swartz, "Phasor-based control for scalable integration of variable energy resources," *Energies*, vol. 13, no. 1, 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/1/190

[19] K. Demir, "A secure and reliable communication platform for the smart grid," Ph.D. dissertation, Technische Universität, 2017.

[20] S. Larson. (2017, October) Every single yahoo account was hacked - 3 billion in all. CNN. [Online]. Available: http://web.archive.org/web/20200611022311/https://money.cnn.com/2017/10/03/technology/business/yahoo-breach-3-billion-accounts/index.html

[21] H. Kelly. (2018, June) Facebook bug set 14 million users' sharing settings to public. CNN. [Online]. Available: http://web.archive.org/web/20200527014946/https://money.cnn.com/2018/06/07/technology/facebook-public-post-error/index.html

[22] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM computing surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, 2003.

[23] (2020, May) WAVEMQ - Tiered message bus for WAVE 3. UC Berkeley. [Online]. Available: https://github.com/immesys/wavemq

[24] B. Zhang, A. Y. Lam, A. Dominguez-Garcia, and D. Tse, "Optimal distributed voltage regulation in power distribution networks," *Cornell University Library Web Page. Available online: http://arxiv.org/abs/1204.5226 v1 (accessed on 23 April 2012)*, vol. 108, 2012.

[25] A. von Meier, E. Stewart, A. McEachern, M. Andersen, and L. Mehrmanesh, "Precision micro-synchrophasors for distribution systems: A summary of applications," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2926–2936, Nov. 2017.

[26] A. von Meier, D. Culler, A. McEachern, and R. Arghandeh, "Micro-synchrophasors for distribution systems," in *Proc. IEEE PES Conf. on Innovative Smart Grid Technologies*, Washington, DC, Feb. 2014, pp. 1–5.