Pacific Northwest
NATIONAL LABORATORY

PNNL-33845

Distributed Grid-Sensing Service Layer

A System Architecture Blueprint

January 2023

Md Touhiduzzaman
David J Sebastian-Cardenas
Alka Singh
Jaime T Kolln
James P Ogle

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <https://www.ntis.gov/about>
Online ordering: http://www.ntis.gov

PNNL-33845

Distributed Grid-Sensing Service Layer

A System Architecture Blueprint

January 2023

Md Touhiduzzaman
David J Sebastian-Cardenas
Alka Singh
Jaime T Kolln
James P Ogle

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Summary

In this report, a scalable, decentralized, sensor-oriented communication blueprint is introduced. The proposed architecture has been specifically designed to support the mass deployment of sensors and also enable data consumers to access the underlying data streams using a distributed systems approach. With the proposed approach, sensors and data clients can be decoupled from the physical communication infrastructure and migrated into a modern, software-defined infrastructure ecosystem that can be configured to suit the end application's requirements. In its current iteration, this blueprint defines the interactions, processes, and expected outcomes that each individual component within the architecture must support the overall ecosystem's needs. Within this work, such an assembly of services is called the data bus; it represents the heart of the distributed grid sensing services architecture, a multiyear project that aims to design, implement, and test a reference sensor-data dissemination network that can support the needs of the extended grid state. We expect this blueprint to guide our future implementation efforts.

Acknowledgment

This work is supported by the Department of Energy's Office of Electricity and authored by Pacific Northwest National Laboratory, which is operated by Battelle under contract number DE-AC05-76RL01830.

Acronyms and Abbreviations

AMI	advanced metering infrastructure
API	application programming interface
BFT	Byzantine fault tolerance
BTM	behind the meter
CFT	crash fault tolerance
CIM	Common Information Model
DDS	data distribution service
DER	distributed energy resource(s)
DERMS	distributed energy resources management system
DGSS	distributed grid sensing service
DNP3	Distributed Network Protocol 3
DSO	distribution system operator
GOSS	GridOPTICS Software System
GPS	Global Positioning System
IEC	International Electrotechnical Commission
MW	megawatt(s)
OpenFMB	Open Field Message Bus
QoS	quality of service
UML	Unified Modeling Language
VLAN	virtual local area network
Pub/Sub	Publication/Subscription

Table of Contents

Figures

Tables

1.0 Background

Power grids rely on a wide array of sensing devices, including dedicated, stand-alone sensors and ones that are part of other control devices and/or systems. These grid sensors provide supervisory agents with a continuous stream of data that provides visibility into the system's condition. Currently, most of this data is forwarded and aggregated through a centralized architecture using dedicated communication networks (Taft, Sensor Network Issues for Power Grid 2017). This communication network for grid sensors typically manages sensing and measurement information collected from grid sensor nodes or responds to requests from grid application services. However, such sensor networks should be capable of managing and storing a large volume of information received from thousands of grid sensor nodes in a centralized manner and thus enable centralization. The current trend of communication networks for grid sensing and measurement is toward more speed, more data, and more endpoints. The need for dedicated communication paths stems from a combination of legacy connectivity specifications and the need for secure and predictable levels of service.

In this context, Figure 1 represents a typical sensor deployment. As can be observed, sensors are often located within specialized networks that tend to isolate individual sensors from neighboring systems. This leads to data silos, which effectively isolate sensor data, making them available only to a limited set of higher-level data consumers. Grid sensors have generally been treated as network structures with architecture and associate standards. Figure 1 also represents a logical architecture for a grid sensor network. In this architecture, sensors are rigidly bound to an application system, often forming a disjointed set. The logical architecture addresses the issues of query mode, programming, communication modes, information abstraction, and logical structure.

Although data interoperability improvement efforts can lessen these silos, they tend to rely on data bridges at the top layer, often disregarding the resources already deployed at the lower levels (e.g., to avoid complications). Furthermore, as the number of sensors and types of client systems continue to evolve, there is an increased need for a connectivity model that supports the everything-to-everything paradigm. For example, the use of distributed grid analytics and controls requires decision makers to have timely access to all the relevant data, regardless of their location, and often under the assumption that there is no central controller (James P Ogle 2021).

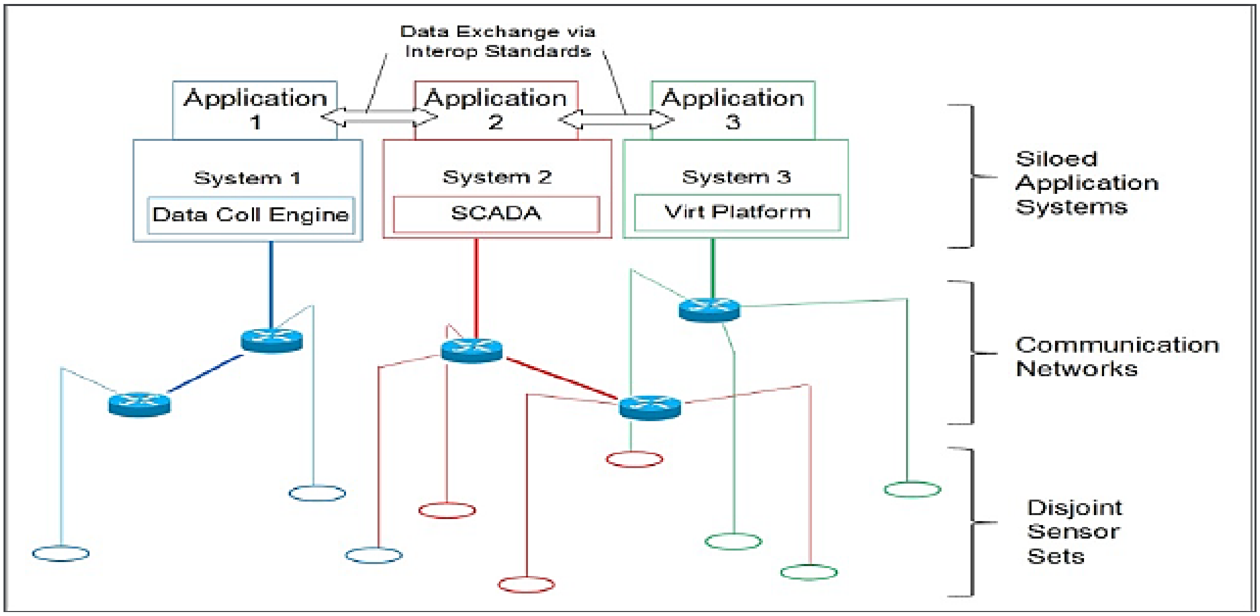

Figure 1. Traditional Grid Sensor System Architecture. (Taft, Sensor Network Issues for Power Grid 2017)

1.1 Limitation of Existing Grid Sensor Communication Networks

Siloed and coupled application. The current communication structure for grid sensor networks is vertical and generally has hub-and-spoke configuration. Such communication systems are often siloed, with centralized data collection head-end and application systems.

Expensive integration. Back-end integration costs make this multiple vertical-silo structure expensive to implement and coupling of applications makes it brittle—failure in one can ripple through to degrade others. The system integration costs for such systems deployed over the past 10 years have increased multiple times over the operational software licensing cost (Muammar 2022).

Limited flexibility. The current architecture has limited extensibility, because when applications are added or removed, existing coupled applications must be updated to integrate them. Standardization efforts attempt to simplify the integration problem but themselves create silos between systems when developers, vendors, and integrators use a variety of standards and information models.

Long Latency. A centralized communication structure may not have high data latency within a given silo system; however, it adds significant latency to any other application that must request data from outside the system that initially acquired it. Each siloed system can perform data conditioning appropriate for the applications in that system, but this treatment may not be appropriate for other systems requesting the data.

As grid dynamics become faster, the effects of latency on the functioning and stability of closed-loop controls must be considered. In siloed system architectures, the need to pass sensor data from system to system via application-level interfaces is often a source of severe latency.

1.2 Extended Grid State

The landscape of sensing and measurement has been expanded to include characteristics of the extended grid state (Taft, Li and Stewart, Extended Grid State Definition 2019). The *grid state* is a collection of voltages, currents, phase angles, system frequencies, and power flows. Sensor data captured from multiple points supports *state estimation*, which defines the existing condition of the transmission and distribution systems. The collected grid-state information is used in advanced grid applications to operate the system and provide resilience. In recent years, power system dynamics has changed drastically with the variability of renewable energy, increasing electrification of loads that traditionally used fossil fuel, and integration of distributed energy resources (DERs). These changes in the power system have expanded the state estimation process into the *extended grid state.* This extended grid state is a combination of physical (asset state), electrical (system state), and ambient-condition parameters collected from different sensors and used in diverse grid applications for advanced system operations and increased resilience. In the extended grid state, the asset state includes status, type/configuration, condition, and geospatial information; system state includes structure, market, electrical, thermal, and communications status; ambient state includes weather, water flow levels, air quality, and the like.

Figure 2 shows a taxonomy diagram of an extended grid state that has been structured into a set of layers, proceeding from the extended grid state to what have been arbitrarily labeled Categories (blue), Groups (green), Kinds (orange), Sets (white), and Types (aqua), just to distinguish the layers.

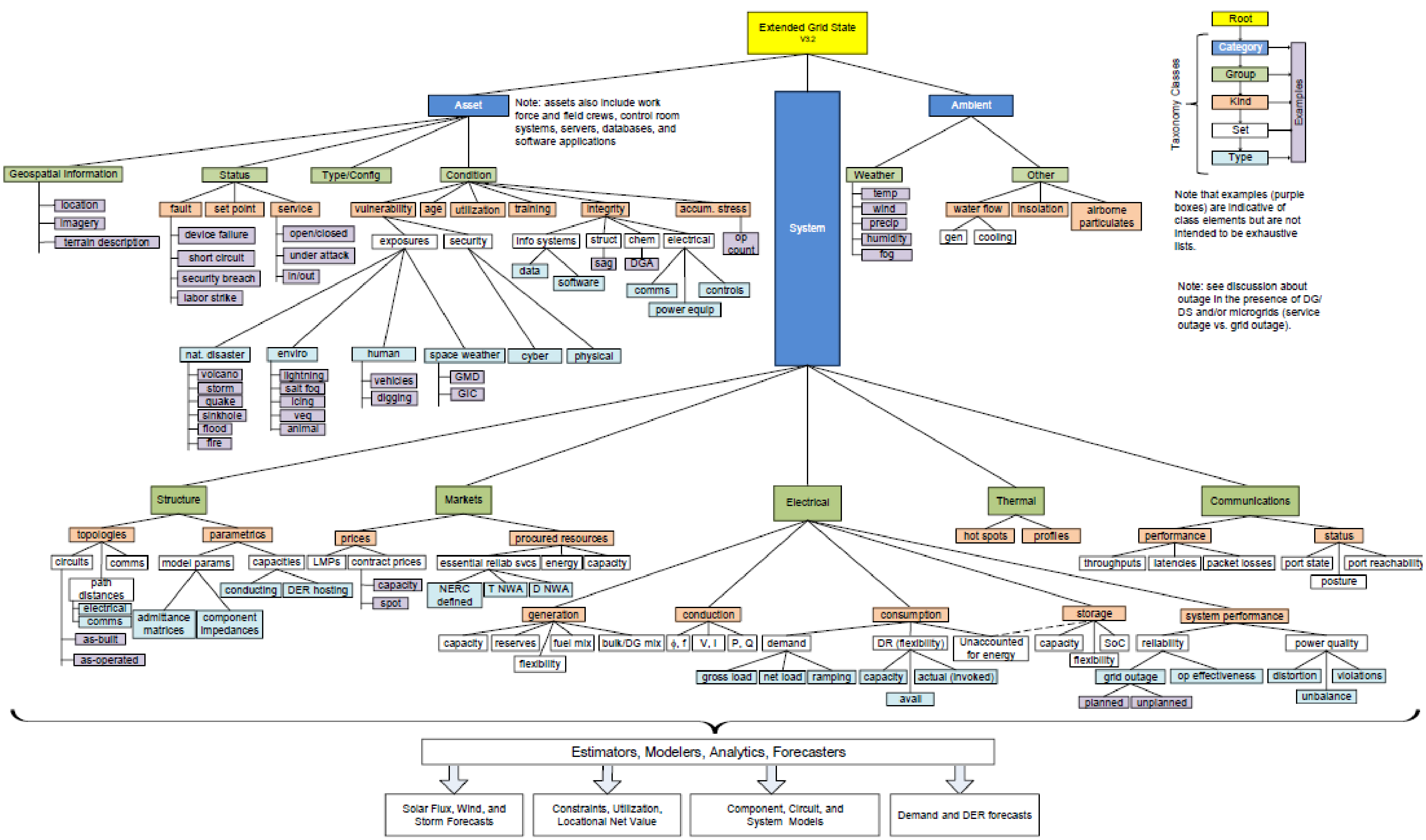

Figure 2. Extended Grid State. (*Extended Grid State Definition* 2019)

1.3 Grid Sensor and Measurement

As modern grid complexity increases, the move toward synchronized measurement that is necessitated by advanced optimization and control requirements leads to the use of smart sensors, including phasor measurement units, on distribution feeders. A smart sensor is one that contains (Bhattarai, et al. 2019) a digital processor with memory, embedded software, possibly downloaded applications, and digital communications capability. Traditional line sensors are implemented as a combination of a set of sensors and a remote terminal unit with embedded processing capability, as well as one or more communication interfaces. The measurements are then transmitted to a central location for developing the grid state model.

1.3.1 Emerging Trends in Grid Sensors and Measurement

Over the past several years, the Department of Energy, utilities, and public/private stakeholders made a massive investment in modernizing the traditional grid to accommodate the diverse requirements of the 21st century. This grid modernization effort ranges from transmission and distribution hardening to DER integration. The list below presents emerging trends that relate to grid sensors and measurements (Xue, et al. 2022):

- Metadata management. Advanced grid sensors, especially at the distribution level, produce large amounts of time-varying metadata. Most distribution grid ICT (interconnecting transformer) and control systems do not handle metadata well, but more importantly, the metadata is often inaccurate or not available. This is a special issue for distribution connectivity because it is often poorly known (connectivity models are only 50% to 80% accurate) and it is the context for all distribution grid data, events, and control.
- Increasing data volumes from the grid, and increasing variety of data from more diverse device types, and increasing observability. Newer protection and control systems needed for advanced distribution grid functionality will generate enormous volumes of sensor data that must be transported, processed, and consumed in real time and be stored for off-line analysis. All told, the utility industry will experience an expansion of data collection, transport, storage, and analysis needs of several orders of magnitude by 2030 (Bhattarai, et al. 2019).
- Faster system dynamics. The new grid capabilities have changed the power system dynamics in the distribution grid. New functionalities of the grid system have started leveraging diverse datasets to shorten operational time scale and thereby improve system dynamics..
- Latency hierarchy. The grid sensor data and measurement need to properly manage quality of service(QoS) to achieve control at a seconds to milliseconds level. QoS matters because some grid services and their analytics operate in real time, meaning that the results must be produced from newly sensed data and delivered for action within strict time constraints.
- Vastly increasing number of endpoints attached to the grid that must be managed, sensed, and/or controlled. In recent days, grid sensor devices such as smart meters and advanced inverters are adding drastically to behind-the-meter measurement for its new functionalities that enhance control and improve security. Widespread implementation of smart sensor devices implies that a distribution grid control system may have to handle large number of endpoints, which existing grid control currently cannot accommodate.

- New desired capabilities raise issues regarding data privacy and confidentiality. Some approaches for transactive and other large-scale coordination methods require that some information from prosumers flows in the control systems. It may be beneficial to share certain data to facilitate transactions, but general security concerns still apply. Most schemes for secondary control of large numbers of endpoints assume sharing of some kinds of data that are not shared today.

1.3.2 Systematic Issue on Sensor and Measurement

Increased data volume and sensor device diversity. The volume of data is continuing to increase as the deployment of advanced sensor networks grows exponentially, including advanced metering infrastructure (AMI) at the distribution level (customer-side metering). This issue will continue to expand as the number of non-utility interconnections grows with DERs, microgrids, electric vehicles, and smart buildings.

Siloed sensor data into applications. Most grid sensors are rigidly bound to the isolated/siloed sensor network or application. This isolation along with other technological constraints prevent sharing data among applications. Also, siloed data loses synchronization over time and the data integrity is threatened by those data being stored in multiple centralized databases. Other than that, as data are bound to applications, sharing data across applications requires point-to-point, back-end software integrations through some intermediate layer of software (shadow historian or other middleware) which places scaling, cost, and schedule burdens on access to extended grid state observability.

QoS management. The increasing presence of technologies such as photovoltaics, inverters, and power electronics have changed power system dynamics, particularly in the distribution grid. Active time scales are shrinking to subseconds and even to milliseconds. The existing QoS management are operating in old fashioned manner and not able accommodate the large volume of data and grid system dynamics

Variation of timing distribution. Grid operations require synchronization of applications and data that reside in geographically dispersed locations. Applications include acquisition and processing of grid sensor data and measurements. Without precise time synchronization through the network, applications lack the fidelity needed for responses that will support critical services such as fast frequency response. For a grid-edge device such as a phasor measurement unit, it is common to have Global Positioning System (GPS) clocks, which are expensive and not fully flexible. Hence, the sensor and measurement system should reduce dependence on GPS clocks and choose synchronization techniques (e.g., precise timing distribution via Precision Time Protocol) that are more flexible.

2.0 Distributed Grid Sensing Service Architectural Overview

Clearly, based on the previously discussed grid sensor and measurement limitations and systematic issues, it can be established that legacy, centralized communication architectures have become a technological barrier toward the deployment of efficient and scalable solutions. Further, a more dynamic grid requires analytics and controls distributed throughout the grid with timely access to data at multiple layers of the system to provide fast response to changes in the extended grid state. To address the limitation and issues, we need an approach that can liberate the data from these silos so that they can be available where and when they are needed for grid operations in centralized, distributed, or fragmented conditions. In this report, we have introduced a service-oriented middleware called Distributed Grid Sensing Services Layer (DGSS) that will provide advanced analytic and control applications with shared access to the growing number of grid sensors and a diverse set of data for local visibility of the extended grid state. The platform has been built from the ground up to satisfy the following goals:

- Support a grid-oriented data distribution service (DDS), with reliability and end-to-end security as core features.
- Leverage features associated with distributed networks, including service decentralization and fault tolerance, regardless of the supporting network topology.
- Provide standard interfaces to clients (data producers and consumers), enabling them to interact with other systems, while abstracting hardware and software differences.

Based on the aforementioned goals, a reference design was constructed by adopting the internet's ad hoc architecture as a starting point and expanding it with the publish/subscribe capabilities provided by a distributed stream processing system. The resulting approach seeks to leverage existing technologies, taking advantage of their maturity, while at the same time making sure that the sensor distribution requirements are met. In addition to identifying the required components, the resulting architectural blueprint has been complemented with behavioral and life cycle management definitions that satisfy the design goals. In summary, the contributions of this paper include the following:

- Presenting a distributed data architecture with decoupled services to enable cross-level sensor with data access.. Data can be used to provide visibility and/or to support control actions at a local or remote site, thereby supporting the deployment of extended grid state applications.
- Building upon mature distributed network approaches to provide an ad hoc, scalable, and security-oriented publish/subscribe messaging system that remains agnostic to the underlying network topology.
- Providing an overview of individual components, and their interactions within the architecture. Furthermore, it provides life cycle management definitions to capture expected client interactions and to describe the internal processes and features supported by the proposed solution.

2.1 DGSS Definition

A Grid Modernization Laboratory Consortium (GMLC) grid architecture document introduced the layering and platform concepts that are applied to electricity infrastructure, sensing and measurement, and communications for electricity distribution to define a platform for resilient electricity distribution operations (Taft and Becker-Dippmann, Grid Architecture 2015). The

DGSS layer is a fundamental layer of GMLC grid architecture that resides between the physical layer and the application layer. It represents an abstraction layer that decouples sensor networks from higher-level applications. From an architectural point of view, the DGSS layer bundles the sensor and communication networks and treats them as an integrated structure; this facilitates grid resilience, flexibility, and future-proofing its operational systems. It also enables service decoupling, eliminating application silos, and enhancing data availability. Therefore, the DGSS is a foundational enabler to enhance visibility and observability at both the bulk power and distribution levels (Taft, Sensor Network Issues for Power Grid 2017). This supports advanced applications that enhance grid capabilities using multiple services such as distributed control, protection, and state estimation applications, and also increases situational awareness via analytics.

From an implementation aspect, the DGSS is a service-oriented, software-defined middleware that lies between applications and sensor networks to provide a reliable DDS. The resulting middleware design and implementation must be capable of supporting near-real-time or real-time heterogeneous data transmittal, while also being able to filter, route, and deliver data ased on the end-application requirements. The end result is a software layer (i.e., the middleware) that acts as a virtual communication network, capable of providing connectivity to multiple components and services while maintaining security and reliability. A successful DGSS implementation abstracts the component-level differences introduced by the data formats/protocols, network transport mechanisms, and operating systems. Therefore, the DGSS concept simplifies the overall integration process by enabling solution developers to focus on satisfying end-application goals and functionalities rather than working the low-level interoperability and connectivity requirements.

2.2 DGSS Target Stakeholders and Actors

The DGSS architecture has a strong focus on satisfying the needs and requirements of distribution network ecosystems. This encompasses the sensors (i.e., the data producers), the communication networks, the control systems, and the stakeholders/actors who depend on sensor data to operate. These actors may use data to support day-to-day operations or to drive planning. High-quality data streams enable stakeholders to optimize operations and to prioritize and select future investments. The DGSS architectural approach proposed in this work addresses several regulatory and business considerations related to utility investment planning, regulatory review, and authorizations. Specifically, it highlights the importance of treating a sensor network as a core investment toward a modern grid.

In this section, four prototypical, high-level stakeholders (i.e., system actors) are used to represent and model key functionalities, which in turn have been used to define the DGSS architecture. These stakeholders are the utility executive, the communications engineer, the power system engineer, and the utility data architect. These represent highly generalized roles of people who are typically in charge of the planning and operation of the distribution system. These roles may be filled by one or more individuals and assigned at different hierarchies and/or job descriptions but are trying to achieve a common goal. An initial summary of these actors is provided in Table 1.

Table 1. DGSS Stakeholders and Their Needs

Actor	Stakeholder's Goal	Stakeholder's Typical Needs
Utility Executive	Optimize investments in the power system operations to maximize the impact and value provided to customers, members, and the community. Such actions are subject to the utility mission, strategy, and business constraints.	Requires a long-term, cost-effective sensor network with substantial future-proofing guarantees. This requires appropriate risk assessment to avoid having premature technology obsolescence, i.e., stranded investments. These investments are balanced between the need to provide reliable electric service and cost to deliver service.
Power System Engineer	Maintain reliable and safe system operation. Optimize the use of existing resources to satisfy daily operational needs (e.g., support power delivery, quality).	Develops control and protection schema that depend on data based on regulatory requirements and the need to maintain a stable system, balancing supply with demand and maintaining frequency, voltage, power factor, and phase balance.
Communication Engineer	Develops and maintains interoperable and reliable communications among utility, distribution, transmission, and balancing authority systems and devices.	Integrates new and legacy sensors and equipment into control and protection systems. Also may have a role in managing databases and back-office use of data, working closely with the data architect within a unified sensor network structure.
Utility Data Architect	Provide flexible grid sensor data access by maintaining the QoS requirement for grid application	Integrate utility data platform to enable the acquisition, storage, preparation, delivery, and governance of its data, and support the security layer for utility customer and grid applications.
System Planning Engineer	Develop solutions to current and future system issues by considering sensor data requirements.	Develop appropriate system modeling architecture by properly considering a utility data model to monitor and control the system; perform system studies.

2.3 Desirable System Properties of DGSS

The characteristics of a complex system can be classified by the system's qualities and properties. System qualities are the characteristics as seen by end users and other stakeholders from an external viewpoint. Qualities are a form of high-level requirements that may be expressed qualitatively or quantitatively. The system properties represent an internal view of the system. The system properties can be thought of as internal requirements for the desired functions of the system. Figure 3 illustrates the relationship.

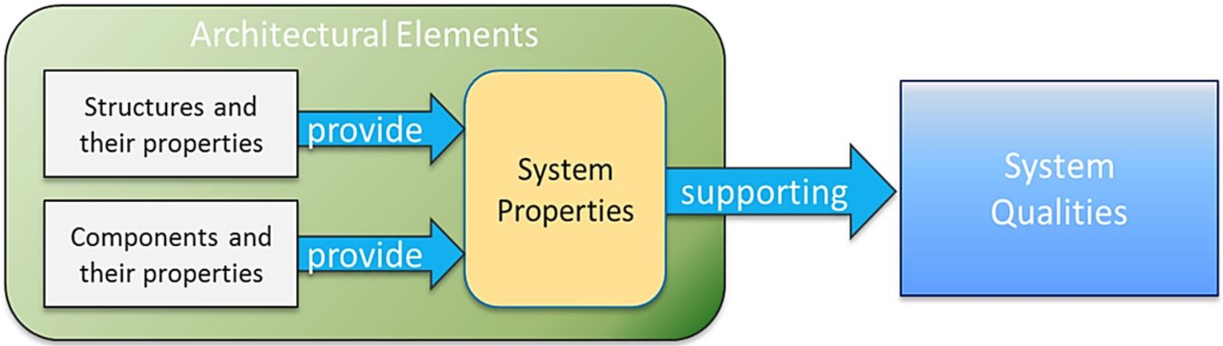

Figure 3. Relationship between System Qualities and Properties

The set of sensing and measurement qualities and properties and associated performance measures must be developed together to develop DGSS architectures. Furthermore, developing sensing and measurement properties and qualities enables powerful methods for evaluating proposed DGSS options. Below are some high-level qualities and functionalities required in DGSS architecture.

2.3.1 DGSS System Qualities

Security. Security is an integral part of power grid operations, including sensor networks and the associated data. Security mechanisms are often treated as overlays that are coupled with specific groups and applications. This application-centric security integration introduces cyber air gaps between layers. Hence, existing grid sensing and measurement networks are vulnerable to diverse attack vectors and scenarios. These attack vectors range from independent components to applications and the data exchange among them. The sensor network's security should be treated an integral part of the sensing layer, actively seeking to limit the number of attack vectors introduced by its adoption, and furthermore, seeking to eliminate existing threats.

The DGSS middleware includes security as one of its core capabilities and can provide data security across multiple typical data flows to the benefit of the end applications. This capability can be leveraged by supervisory agents to determine when sensors are malfunctioning or whether sensor data is being manipulated.

Resilience. Changes in the grid state require reliable and predictable responses that satisfy the timing requirements of the end application. The correctness of this response is a critical requirement of eventually providing system-level resilience, particularly as adoption of highly variable renewables and other DERs increases and more distributed control mechanisms are implemented. In this context, it is clear that a well-designed sensor network that enables crosscutting interoperability is an essential tool that can be leveraged by advanced analytics. A successful architecture must liberate its data from silos, enabling the data to be used when and where they are needed, regardless of the type of application being deployed (e.g., centralized or distributed), thereby enabling the end application to provide increased resilience.

Data/Application Separation (architectural approach). Existing sensor-driven applications remain largely siloed, thereby restricting data access to only the systems involved in its collection, processing, or storage. Once a data source is siloed, successful data sharing requires complex retrofits that often rely on inflexible point-to-point integrations. In addition, these may be often centralized at the back end (rather than sharing data from its inception point), thereby resembling a *patch* rather than a long-term solution. This retrofit-based solution is very expensive and difficult to maintain due to the architectural and interoperability challenges of making sure that the original applications continue to work, while the “patch” achieves its data import/export goals. As an alternative to point-to-point integrations, the DGSS offers a distributed, multi-tiered management system that effectively stores data across multiple layers. This platform solution is vertically structured and introduces high scalability, a benefit of decoupled data access. DGSS uses a distributed data architecture concept to provide a uniform, yet abstracted data access from multiple sensor each deployed at multiple levels of the grid, removing the application/data silo constraints. This enables end users to dynamically group the sensor data and create overlapping sets.

Data accessibility. The abundance of valuable sensor data from grid sensors presents an opportunity to accelerate grid modernization. But the data accessibility barrier prevents full utilization of these sensor data. Although powerful solutions already exist that provide more intuitive and zero-learning-curve data access features by supporting a broader range of user types, these solutions are not specifically developed for grid sensor systems. DGSS uses a publication-subscription mechanism as its data delivery/flow solution, which can support enhanced data access any time it is needed.

2.3.2 DGSS System Functionalities

Synchronized Data Sampling. DGSSs have synchronized sampling functionalities to support advanced grid applications involving fast dynamics. Synchronized sample functionalities minimize or eliminate sample skew and clock deviations. In DGSS, this functionality is accomplished by considering three elements:

- precision time distribution
- synchronized data sample acquisition
- time stamping.

Note that time stamping alone is not sufficient. The samples must be acquired at as simultaneously as possible, not just annotated with the time of acquisition. To accomplish this, each sampling device, whether it is a smart sensor or a data acquisition engine, must be precisely synchronized with the other clocks , which can be accomplished via network time protocols. The distribution of timing in networks is somewhat complex but is well developed. The complexity arises from the need to operate across network domains, leading to multiple classes of network clocks and protocols for timing distribution.

Dynamic Sensor Grouping. DGSSs can group a subset of sensor data from a given network based on applications or use cases. Grid sensor applications can subscribe to the appropriate sets and subsets of sensor data, which do not have to be disjoint but can be treated independently and/or grouped virtually by applications as needed. The DGSS layer and the applications must have certain capabilities for this to work autonomously. The capabilities required include

- Advertisement. Sensors must be able to advertise their presence on the network and describe their capabilities and characteristics while maintaining their privacy.
- Binding and access control. Sensors and applications must have methods and protocols to agree on data stream subscriptions, including function, location, and security criteria.

- Precedence resolution. For sensors that require control inputs to set parameters, a mechanism is needed to resolve which application has precedence when sensors can be shared, how precedence is established, when it expires, etc.

Sensor Discovery. Sensor discovery refers to the announcement and registration of new grid sensors with the DGSS layer. Other than announcement and registration, all the applications that are tied to DGSS must have ways to discover the grid sensor data that they need.

Layered Coordination Framework. In the DGSS architecture, sensing and measurement systems can be viewed abstractly in layered format (Taft and Becker-Dippmann, Grid Architecture 2015). In the case of DGSS, the sensor layer can separate and isolate the application layer from the communication layer. This prevents requiring changes in either the application or communication layer from propagating when changes are made in one or more layers.

Network Level Cybersecurity. Security features in DGSS should be an interlayer in the sensor network, rather than an overlay. In other words, security should be an integral part of the DGSS layer, not something that is added after the network deployment. DGSS capabilities and services comprise four categories:

- Data integrity, confidentiality, and privacy – includes encryption (internodal or end-to-end), key and certificate management, Internet Protocol Security.
- Device and platform integrity – methods to make sure devices and systems have not been compromised at the hardware or code levels, including methods for supporting supply chain integrity, tamper resistance/detection, signed firmware images, posture assessment, secure software life cycles, etc.
- Access control – identification, authentication, network access control, subscription control
- Intrusion detection and mitigation – signature and behavioral analysis, traffic analysis, node exclusion, nonrepudiation, network segmentation, VLAN (virtual local area network).

Interoperable. DGSSs should have system-level interfaces that support the interoperability requirements of stakeholders. A DGSS will adopt recognized and widely accepted standards and protocol using the International Electrotechnical Commission (IEC) Common Information Model (CIM). The DGSS will leverage the GridAPPS-D platform (DOE 2016) and its standards-based application programming interface (API) for application portability. Noted that the GridAPPS-D platform has a system-level interface module that can translate different protocols and standards into CIM format in an ad hoc manner.

2.4 Key Architectural Structure of DGSS

2.4.1 Layered Architecture

In layered architecture, system elements are grouped into separate layers and each layer manages complexity through encapsulation and isolation. Functionality associated with those system elements is thus encapsulated within a layer. Other layers can access the services provided by that layer's functionality through well-defined interfaces without needing to know anything about the underlying details contained in the serving layer.

The DGSS layer serves as a fundamental middle layer, as shown in Figure 4. This layering concept was adapted from a grid architectural viewpoint (Taft and Becker-Dippmann, Grid

Architecture 2015). This DGSS middle layer can isolate the physical layer from the application layer. The DGSS layer is a combination of a distributed grid sensor network and DGSS middleware. The architectural view of the distributed grid sensor network treats sensors and the communication network as an integrated structure. Along with the smart sensors, this DGSS sensor network also includes legacy sensors, embedded sensors and indirect sensors, enabled by incorporating the concept of "virtualization." DGSS middleware is a software package that can be deployed in the DGSS sensor network throughout the grid in utility, non-utility, and run-on equipment such as intelligent devices and sensors in transmission, distribution, or behind-the-meter domains. This DGSS layer operates as a publish/subscribe data system and sends information between the physical layer and application layer through DDS. The middleware helps to achieve all the DGSS system qualities by incorporating the characteristics of DGSS functionalities through sensor orchestration, data analytics, security, network management.

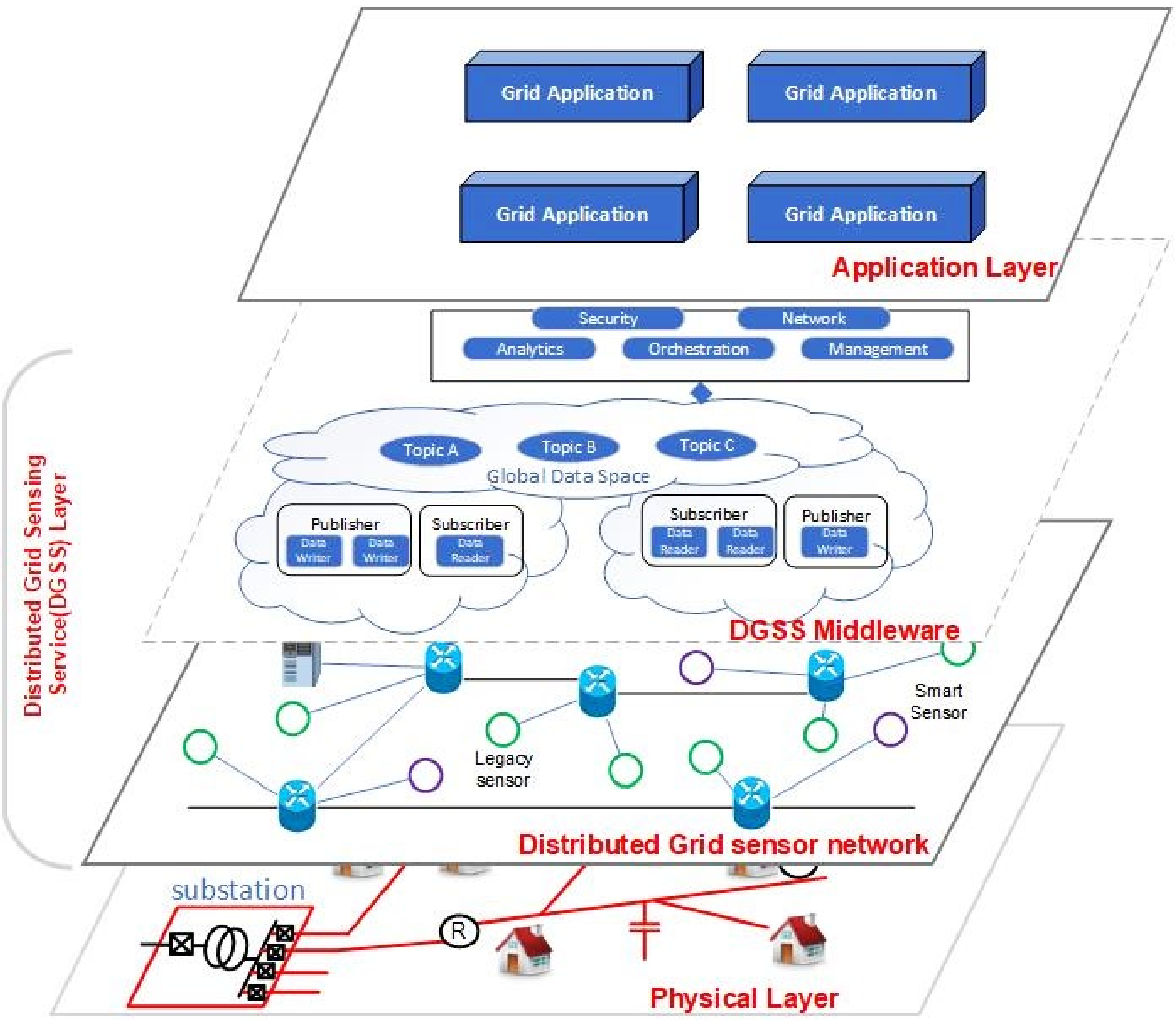

Figure 4. DGSS Layered Architecture

2.4.2 Enterprise-Level Solution Architecture

Enterprise-level solution architecture is based on the structured architecture that converts a business requirement into a solution design by satisfying the organizational or system need. Enterprise solution architecture tells a solution architect how to turn business requirements into solutions or products. DGSS enterprise architecture, shown in Figure 5, is symmetrical with Purdue reference architecture (Rathwell n.d.), starting from the edge/facility domain and ending in the application domain.

- Facility domain – The facility domain is the lowest layer in DGSS enterprise architecture, and it hosts the physical sensors/actuators along with their associated network access points. It enables sensors to forward domain-specific information into the data bus. Power grids use a wide array of sensing devices, including those that are integrated into complex devices/systems (e.g., an actuator), as well as dedicated sensors. In the facility domain, all the data produced by sensors (e.g., electrical and environmental) is packetized using a dedicated field bus wrapper and transformed into DGSS-aware packets that can be forwarded to other domains.
- Substation domain – Within the DGSS architecture, this layer can bridge the *control and automation domain* with the *facility domain* in a local or remote context. The dual connectivity profiles (remote and local) are achieved by supporting the deployment of fully distributed, autonomous data buses across all participating substations. Naturally, each data bus is capable of bridging both domains within the context of a local substation, but each local bus also can interact with other remote buses (e.g., other substations or control centers) by dynamically binding to remote endpoints, and thus effectively creating a network tunnel that abstracts the geographic or hierarchical differences of each endpoint.
- Control and automation domain – The control and automation domain binds to the DGSS data bus, enabling entities to send/receive information to/from the *substation domain* (and thus the *facility domain*). This domain enables data readers and writers to subscribe (and publish) to the underlying data streams, while inheriting the DGSS functionalities.
- Application domain – The application domain is the upper-level domain of DGSS architecture that deals with grid applications by sending/receiving information to/from the control and automation domain.

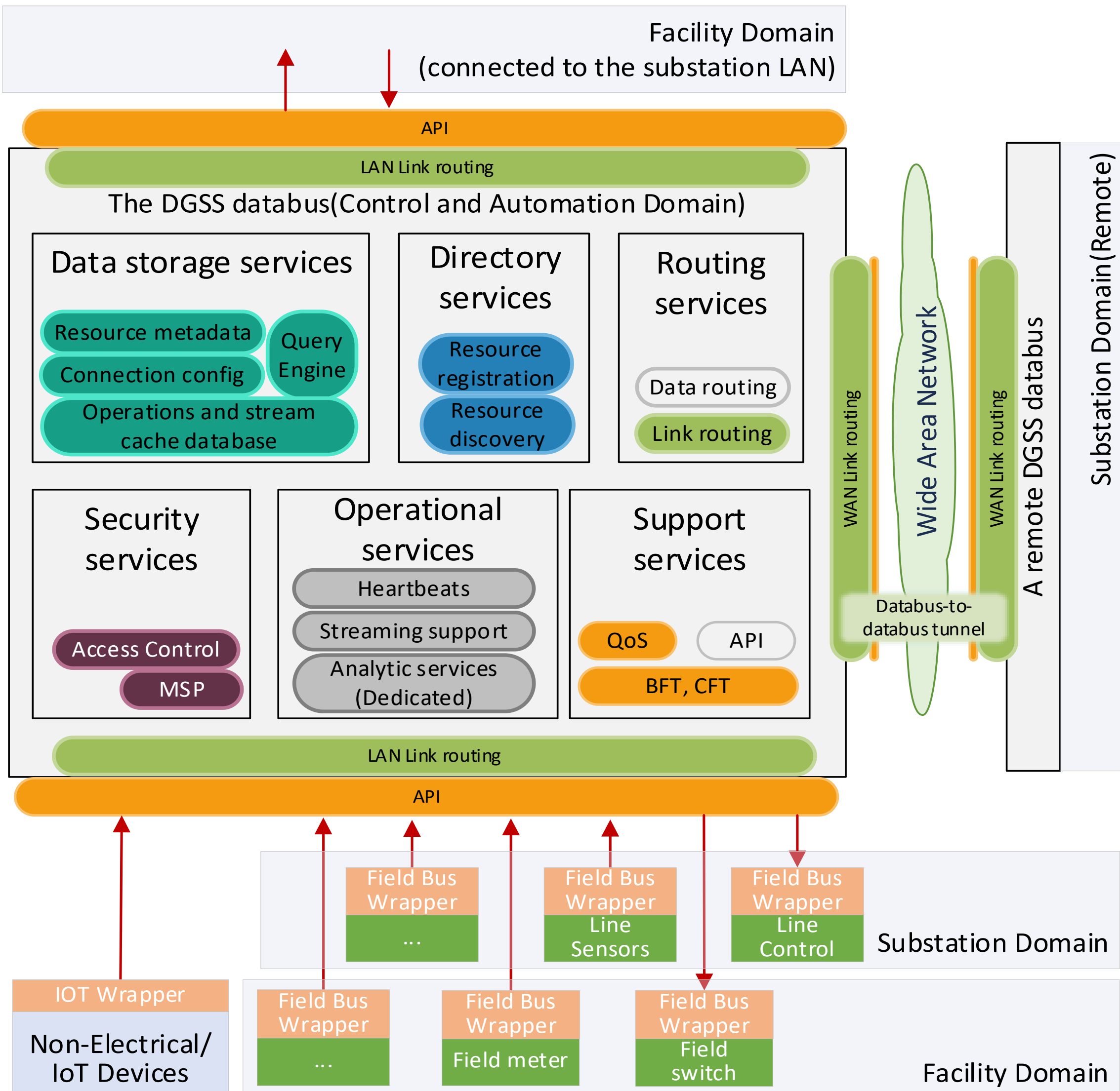

Figure 5. DGSS Enterprise Architecture

DGSS Data Bus Definition

As mentioned earlier, the DGSS middleware relies on a publish-and-subscribe architecture to support its operational goals. Under this context, the data bus acts as the heart of the solution, responsible for decoupling interactions, the data bus offers participants the ability to interact with all others via a common interoperability layer, thereby enabling all participants to exchange data regardless of their actual communication and/or grid hierarchy, enabling multi-system integration. According to RTI, "A data bus is a data-centric software framework for distributing and managing real-time data in intelligent distributed systems. It allows applications and devices to work together as one, integrated system" (RTI 2022). The key attributes of the DGSS data bus include the following:

- Participants directly interface with the data bus through a standardized API.

- The DGSS data bus is a software-defined component that leverages the capabilities provided by distributed networks, enabling a wide array of user and application-level customization.
- The data bus follows a data-centric approach; it is context aware and thus can perform low-level data-handling tasks (e.g., to support down-sampling, aggregation, and filtering, assuming that end-to-end encryption is not required).
- The DGSS data bus provides mechanisms to actively manage the link layer properties, which make sure QoS and connectivity metrics can be achieved (subject to link availability and capacity).

2.5 Protocols and Standards Involved in DGSS

The DGSS can use existing communication network protocols, along with additional services supplied by attached processors, to form a complete sensing and measurement platform. Among the key protocols and services are

- direct access to sensors via the IPv4/IPv6[1] protocol suite, including broadcast modes
- MPLS[2] and PIM/SSM9[3] for handling streaming data and providing a publish-and-subscribe mechanism for sensor data
- standard network management functions with extensions for sensor monitoring
- IEC CIM interface services – Generic Data Access, High Speed Data Access, Time Series Data Access, and Generic Eventing and Subscription
- sensor registry service or service advertisement
- IISO/IEC/IEEE 21451-4:2010TEDS (Transducer Electronics Data Sheets) service(IEEE/ISO/IEC 2010)
- resource metadata (e.g., geolocation, coupling point)
- timing distribution
- software-defined networking interface services.

[1] Internet Protocol version 4/version 6
[2] multi-protocol label switching
[3] protocol independent multicast / source specific multicast

3.0 DGSS Services and Life Cycle Management

3.1 DGSS Services

The DGSS layer can manage information collected from grid sensor nodes and promptly respond to queries from grid application services. It also efficiently maintains metadata information and successfully removes the silos by separating the data from applications. All the DGSS functions are achieved by following the standard security characteristics.

DGSS is a service-oriented, layer-based architecture where various services are inserted into DGSS structure via attached servers or through integration into network management systems. These services support different DGSS functions and thus help to achieve DGSS qualities. Various DGSS services include data storage service, standard network management, and security functions as well as grid-specific capabilities such as sensor metadata management, interfacing CIM to IEC 61850 (IEC 2022), and grid topology/connectivity. Figure 5 in the previous section shows the different types of DGSS services that reside in the data bus. Each of those services is described below.

3.1.1 Data Storage Service

As expected from a publish/subscribe architecture, the DGSS relies on data streaming for data transport, yet it complements streaming with a data caching mechanism that enables the system to offer limited data-at-rest capabilities. These transient data-storage capabilities enable the system to offer clients support for data time-shifting, thereby enabling clients to recover data that were missed due to a disconnection or malfunction.

The data storage architecture also enables data replication, which can provide an additional operational guarantee to the system. The data storage services consist of

- operation and stream cache database – supports the pub/sub mechanism
- connection/configuration database – enables the system to keep track of clients and connections
- resource metadata index – keeps track of participant capabilities
- cold-storage system – a dedicated system that enables persistence and limits the metadata storage time to a user-defined period.

3.1.2 Resource Directory Service

The resource directory service is responsible for maintaining a sensor/subscriber registry that enables services to be mutually discoverable via a custom Lightweight Directory Access Protocol (LADP n.d.)implementation. The two main components include resource discovery and resource registration. Resource registration can keep track of all the sensors connected to DGSS middleware using a service directory. The resource discovery service enables data clients to search for data providers using flexible queries. This pair of services (along with the service directory) enables data consumers to dynamically find data producers based on parameters such as nameplate details, location/ownership properties, or effective access permissions. Characteristics of these two services are described below:

- **Resource Discovery Services**. Discovery is an essential step in the process of allocating and managing the system's resources and providing applications with the optimal set of resources to fulfill their requests. Furthermore, the discovery process requires continuous access to all available resources in the system. Resource discovery enables searching for grid sensors/resources based on the system/service requirements. Resource discovery services face many challenges:

  - Distributed resources: Resources are traditionally announced within a local scope (e.g., siloed). Creating a universal, globally searchable metadata store requires solution architects to redesign this approach.
  - Heterogeneity: Devices are highly specialized, making it difficult to create comprehensive taxonomies to efficiently describe their capabilities.
  - Scalability: Supporting large data indices, such as those required for global-scale metadata stores, requires planning and decentralization (to avoid having a single, large database that could be a single point of failure).
  - Timeliness: Not all devices are operated on the same time clock.

- **Resource Registration**. This registry system can keep track of all the sensors connected to DGSS middleware via metadata registry management. The grid sensors connected to DGSS are added to a resource registry when complete information regarding its capabilities and availability is provided. This registry system helps to identify the available grid sensors based on the DGSS user requirements.

3.1.3 DGSS Routing Service

The DGSS routing service provides for delivery of sensor data to its consumers. This routing service operates on two different levels to provide service flexibility. At the data level, dynamic filters can be set to selectively set the delivery path based on message contents, etc. At the link layer, routing can be configured to optimize for different network needs (e.g., based on topology); this includes edge-to-edge, edge-to-remote, and hybrid architectures (e.g., edge-to-edge-with-remote). This routing flexibility is expected to enable future performance optimizations to satisfy the end-user needs (e.g., to accommodate the communication network topology or capacity constraints).

3.1.4 Security Service

The DGSS is intended to embrace security by design. To this effect, an internal security model with a configurable service plugin interface can be used to provide ad hoc security constructs to the data bus, enabling plugins to secure a wide array of services and processes. Typical examples may include (a) securing the sensor registration/discovery process; (b) securing end-to-end configurations (e.g., securing routing and QoS settings); and (c) providing optional in-transit and at-rest confidentiality guarantees. Some currently planned plugins are listed here:

- Authentication service plugin – provides the means to verify the identity of the application and/or user that invokes operations on the DDS. Includes facilities to perform mutual authentication between participants and establish a shared secret.
- Access control service plugin – provides the means to enforce policy decisions on what DDS-related operations an authenticated user can perform. For example, which domains it can join, or which topics it can publish or subscribe to.
- Cryptographic service plugin – implements (or interfaces with libraries that implement) all cryptographic operations including encryption, decryption, hashing, digital signatures, This includes the ways to derive keys from a shared secret.

- Logging and monitoring service plugin – supports auditing of all DDS security-relevant events.
- Data tagging service plugin – provides a way to add tags to data samples.

3.1.5 Operational Service

This service contains the DGSS workhorse components responsible for carrying the data-streaming tasks and delivering data to all subscribers. It also provides active monitoring (via heartbeats) and logging functionalities.

3.1.6 Support Service

This service category includes the components required for the data bus to interact with clients (i.e., the API). It also includes the components required for service decentralization, fault-tolerant handlers, and, generally speaking, any component that helps the data bus abstract the software from the network's or host's particularities.

3.2 DGSS Life Cycle Management

Generally speaking, any platform that seeks to support the presence of ad hoc actors must clearly define mechanisms to enable new and returning actors to join, use, and eventually abandon their services. This is often referred to as an operational life cycle and is used to denote the various steps (and transitions) that actors must follow to transition from a "starting" point into a "finalized" state. The DGSS builds upon this requirement not only to support the presence of ad hoc actors but also to encourage predictable behaviors among all participants. In DGSS, the actors are sensors, data subscribers, and data publishers. A graphical overview of the operational life cycle as proposed within the DGSS architecture is presented in Figure 6. DGSS life cycle management provides a framework for DGSS middleware development and also helps manage this middleware over time. In the upcoming paragraphs, extended descriptions of each phase will be provided, along with how it makes use of the services provided by the internal components.

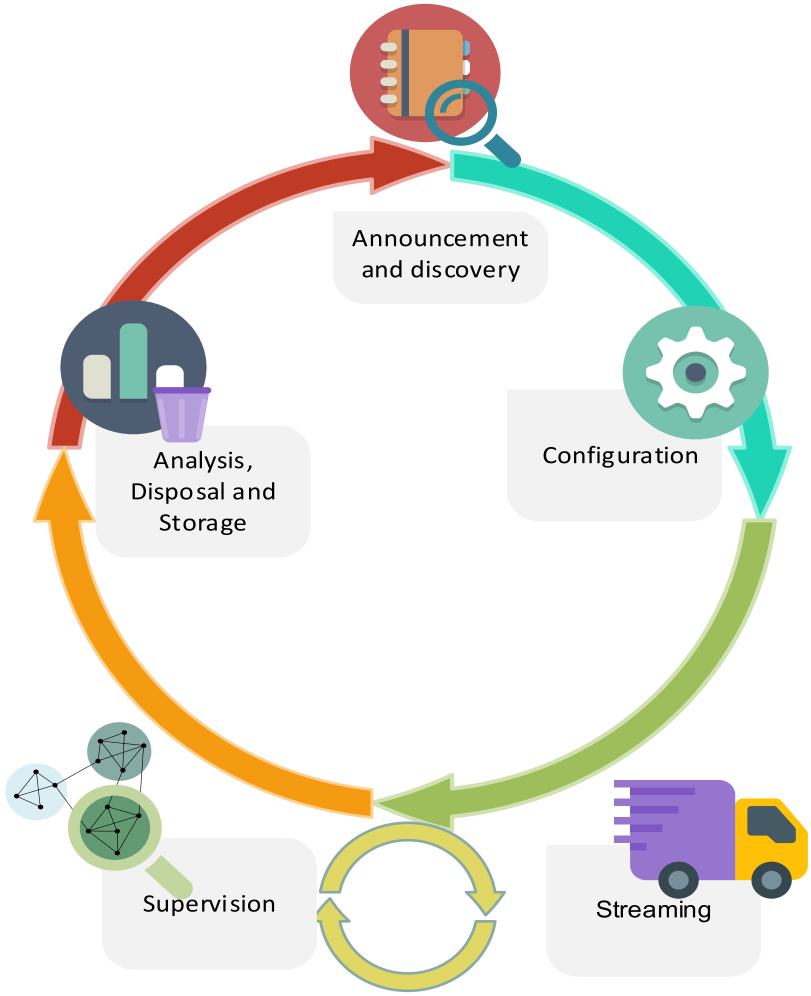

Figure 6. High-Level Overview of the Life Cycle Management Phase

Announcement and Discovery. This stage acts as the entry point to the platform. Because data producing and data consumer actors are present, two alternatives may occur. Either data producers reach out to the DGSS middleware layer to announce new capabilities/services, or data consumers issue discovery queries to find previously announced capabilities/services. This phase is mainly supported by the directory services component, a custom implementation of a Lightweight Active Directory Services provider. This stage relies on the functionalities provided by the data storage services component to host the service directory data store, and the security services component to make sure access control rules are followed—e.g., to identify who is authorized to access, discover, or register within the network.

Configuration phase. During this stage, the software-defined data bus settings can be provisioned on a global or a per-application basis. This may include (a) defining the network (i.e., creating and setting preferred network paths, defining QoS goals); (b) defining the fault tolerance policy; and (c) defining the data stream properties (e.g., cache time, size limits, end-of-life procedures).

Streaming. During this phase, the actual data transmission occurs. Actors participate either as data sinks or data sources, constantly receiving or sending data across the data bus. Hence, the data bus becomes responsible for providing reliable end-to-end data delivery with minimal oversight. During this stage, the internal components work to make sure global and application-specific requirements are met.

Supervision Phase. During this phase, periodic heartbeats are used to detect malfunctions at the edge (and act upon them). Also, within this phase, the data bus performs internal checks and logs operational metrics for future analysis (e.g., QoS metrics, link health, traffic counters).

Analysis and End of Life. After a disconnection message is received (or a timeout condition is reached), a data cleanup process is initiated (leading to data disposition or archiving). Operational logs are made available to analysts parallelly.

4.0 Overview of DGSS Middleware

DGSS middleware will be constructed by following the DGSS life cycle management and will leverage GridAPPS-D platform for its implementation. In this section, we will briefly describe how the information will flow through DGSS middleware based on the life cycle phases. Other than the information flow, this section also provides a basic understanding of different implementation components (GridAPPS-D platform, CIM model, OpenFMB) that will shape the DGSS middleware.

4.1 DGSS Middleware Information Flow

In DGSS, the life cycle phases are modeled using a simplified three actor model. These actors can be described as follows:

Data producers. Data generators include sensors, but also data-augmentation applications (e.g., a state estimator). These actors must “announce” their capabilities to the platform, and continuously “stream” data to the data bus (particularly when other actors require such data). Data producers interact through the common API, which is in charge of providing and abstracting low-level connection and context management tasks (i.e., sending periodic heartbeats).

Data consumers. The data consumers category is expected to include most grid applications. These actors perform on-demand service “discovery” to identify relevant data sources. Applications then configure “downstream” connections, performing configuration on a per-source basis. This is similar to data consumers interacting through the common API.

Data bus internal manager. This internal actor is responsible for safeguarding the continuous operation of the data bus using a continuous supervision and active maintenance model. Although this actor is invisible to other actors (and should remain so), it exposes a set of dedicated interfaces and services that can be used by system administrators to monitor and manage its behavior.

4.1.1 Announcement and Discovery Phase

The announcement and discovery phase serves as the main entryway to the platform. Two main data flows exist:

- Data producers announce their “capabilities” to the *resource registration component*. Internal mechanisms then verify and confirm the actors’ identity and permissions before enabling the registration to be reflected in the *resource metadata database*.
- Data consumers invoke a search request via the *resource discovery component*. As before, internal mechanisms verify and confirm the actors’ identity and permissions before forwarding the request to reach the *query engine*, which searches the *resource metadata database*.

Data flows in the announcement and discovery phase are diagrammed in Figure 7.

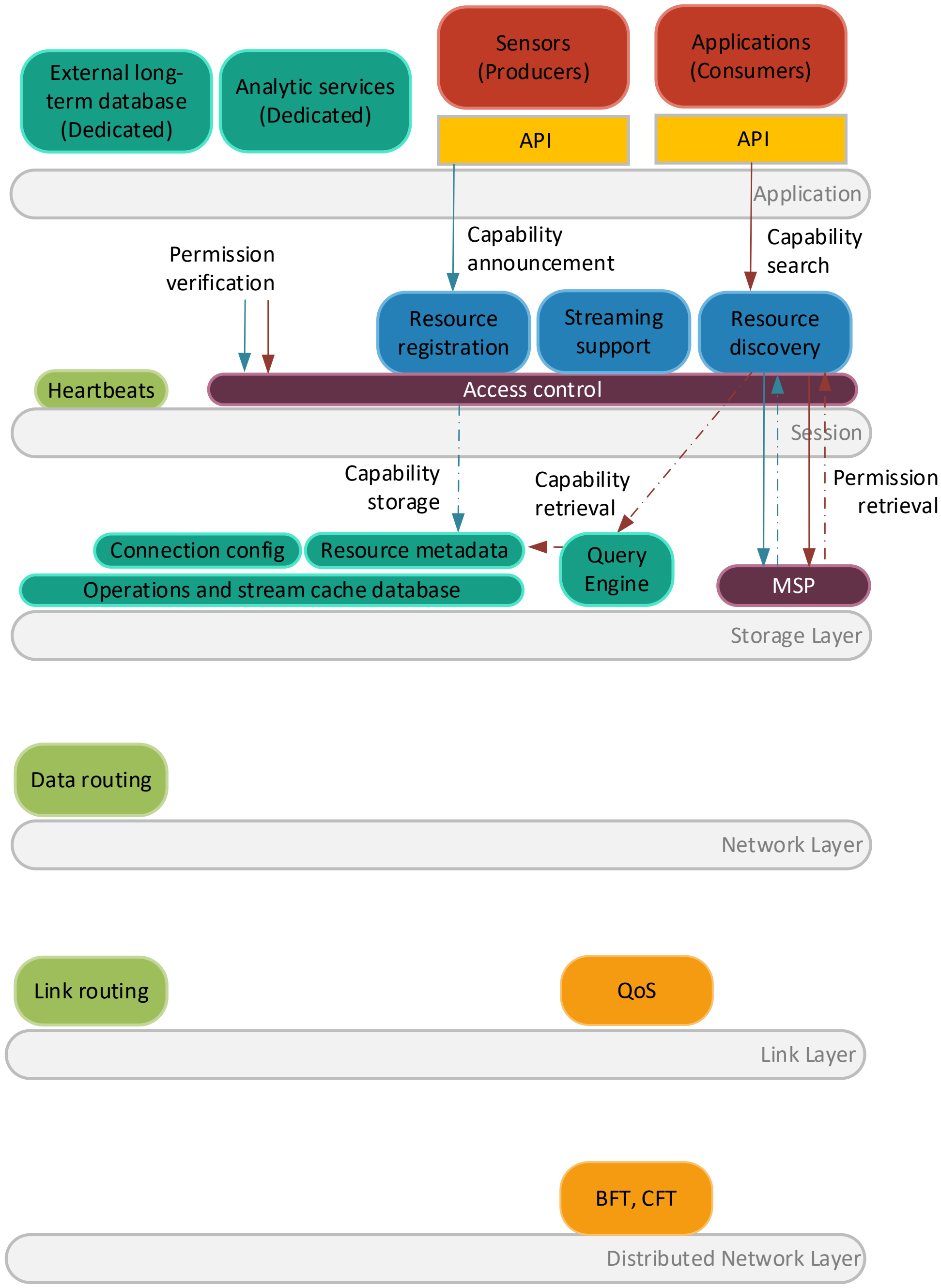

Figure 7. DGSS Life Cycle Announcement and Discovery Phase. MSP is managed service provider.

4.1.2 Configuration Phase

During this phase, actors request and provision the operational settings that will be used during the data transmittal stage. Once configured, the settings will be kept within the *connection* *configuration database*. Depending on the actor's role (data producer or data consumer), the following options for operational settings exist:

Data producers. Actors request the data bus to provision the following services:

- *QoS*: Usually defined and limited by the actor's own QoS. Options include required reliability rate (e.g., 99%, 99.9%, 99.99%), maximum actor-to-data-bus time delay (e.g., 10 ms, 100 ms), and maximum end-to-end time delay (for data to be considered "valid").
- *Architecture fault tolerance*: Three options are available to all participants: (a) crash fault tolerance (CFT), in which data is replicated across multiple hosts, so it can withstand unmalicious failures; (b) Byzantine fault tolerance (BFT), which is an extension of the CFT option but also renders the datacapable of withstanding failure due to malicious actors; or (c) no fault tolerance. Each option incurs additional computation and communication overheads, which must be weighted.
- *Link routing*: This enables the producer to define the preferred links used to reach the host data bus (which could be a local or remotely located entity). The host bus in this case represents the set of systems that are responsible for providing fault tolerance and thus will host the data cache.

Data consumers. As with the data producers, data consumers must provision the data bus before being capable of receiving data. A configuration profile is applied on a per-connection basis. Profiles can be saved for future use, and default profiles (e.g., per-device class or per-application class) may be defined. A summary of configurable attributes is as follows:

- *QoS*: Options include required reliability rate (e.g., 99%, 99.9%, 99.99%), maximum data-bus-to-actor delivery delay (e.g., 10 ms, 100 ms), and maximum end-to-end time delay (for data to be considered valid). These values are limited by the stream source's QoS (e.g., a reliability of 99.99% cannot be achieved if the source's reliability is only 99%).
- *Architecture Fault Tolerance*: This option follows the data producers' s settings.
- *Link routing*: This enables the producer to define the preferred network paths used to push data from the data bus to the consumer.
- *Data routing*: This option configures the data bus internal data handler to perform data-level filtering and aggregation. During its first iteration, the data handler can provide (a) down-sampling (e.g., filter one out of every *n* messages) or (b) aggregation (perform average, minimum, and maximum functions, over custom-sized windows) before publishing to the data consumer.

Figure 8 is a diagram of the configuration phase.

As a final note, both actors also configure the termination behavior, which will be described later.

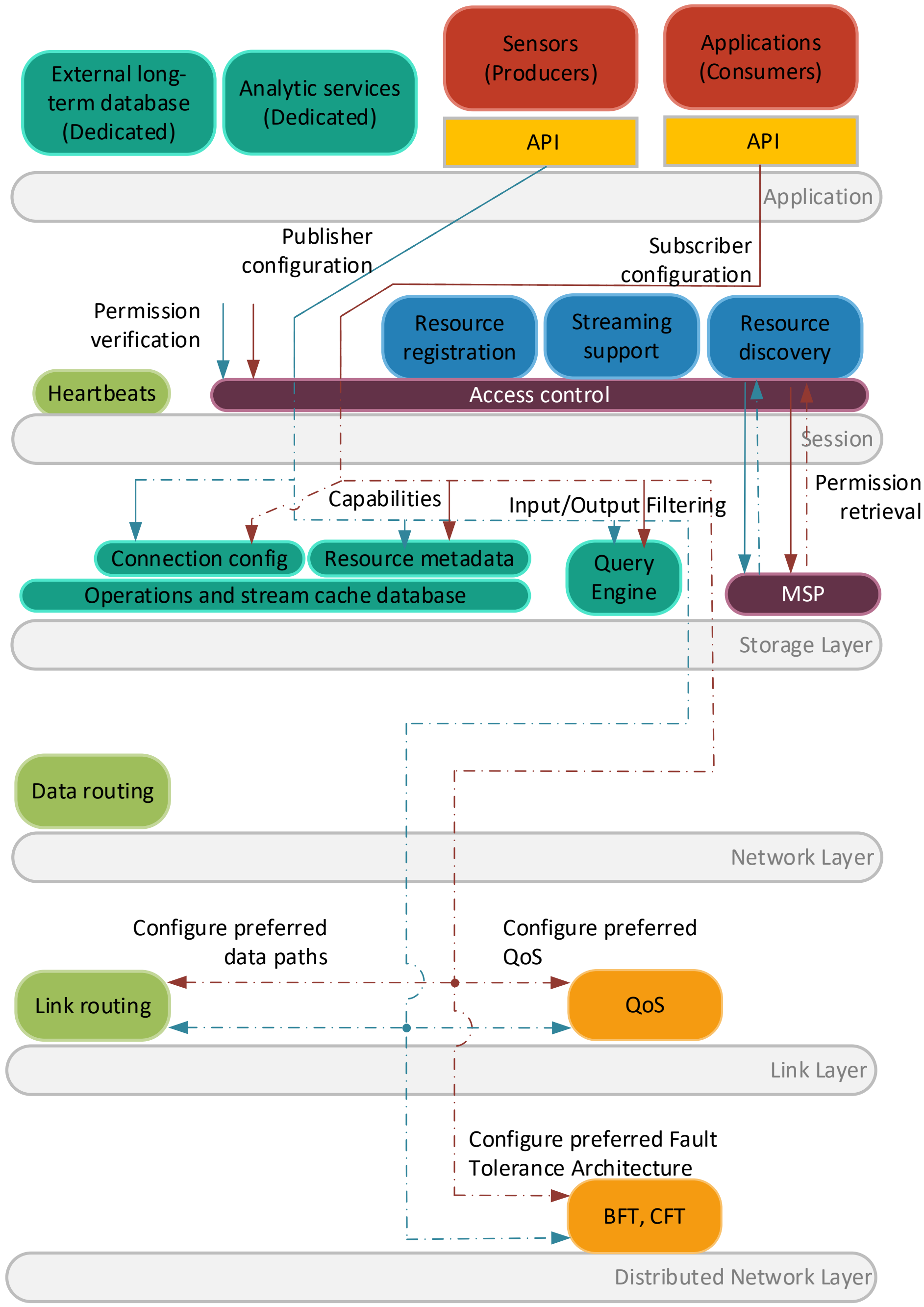

Figure 8. DGSS Life Cycle Configuration Phase

4.1.3 Streaming Phase

During the streaming phase, the actual data transmittal occurs. Both data producers and data consumers actively rely on libraries to abstract much of the connection management (e.g., packaging and unpacking data according to the API definition). However, during this phase, the data bus internal manager is actively working to enable end-to-end communication; this includes enforcing the security rules (via the *access control* module), and managing the communication paths, while attempting to satisfy QoS and fault-tolerance requirements (as defined within the *connection configuration datastore*). In addition, components such as the *query engine* and *data* *routing* modules constantly filter data. This filtering can be represented as a selective copy-paste mechanism between an input buffer and an output buffer (e.g., producers populate the input buffer, while consumers get their data from the output buffer). The streaming phase is shown in Figure 9.

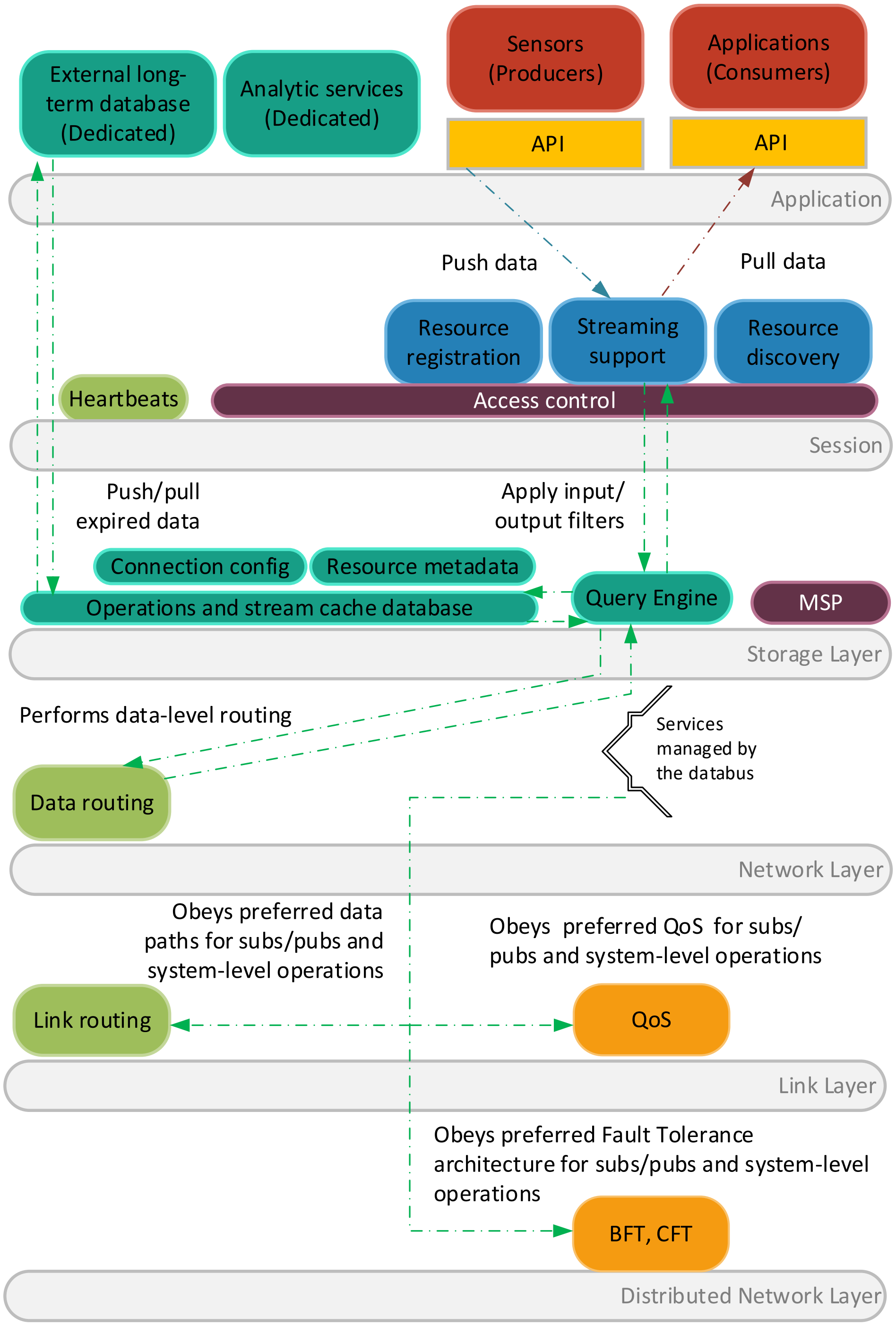

Figure 9. DGSS Life Cycle Streaming Phase

4.1.4 Supervision Phase

The supervision phase runs parallel to the streaming phase. During this stage, the data bus internal manager actively monitors and logs performance records. These are stored in the *operations and stream cache database* for future analysis. In addition to the logging activity, an internal security function performs routine audits to make sure participants' access has not been revoked (which would enable connection termination). In addition, during this phase, the CFT and BFT modules actively supervise and manage their resources to maintain their operational gurantees. Figure 10 is a diagram of the supervision phase.

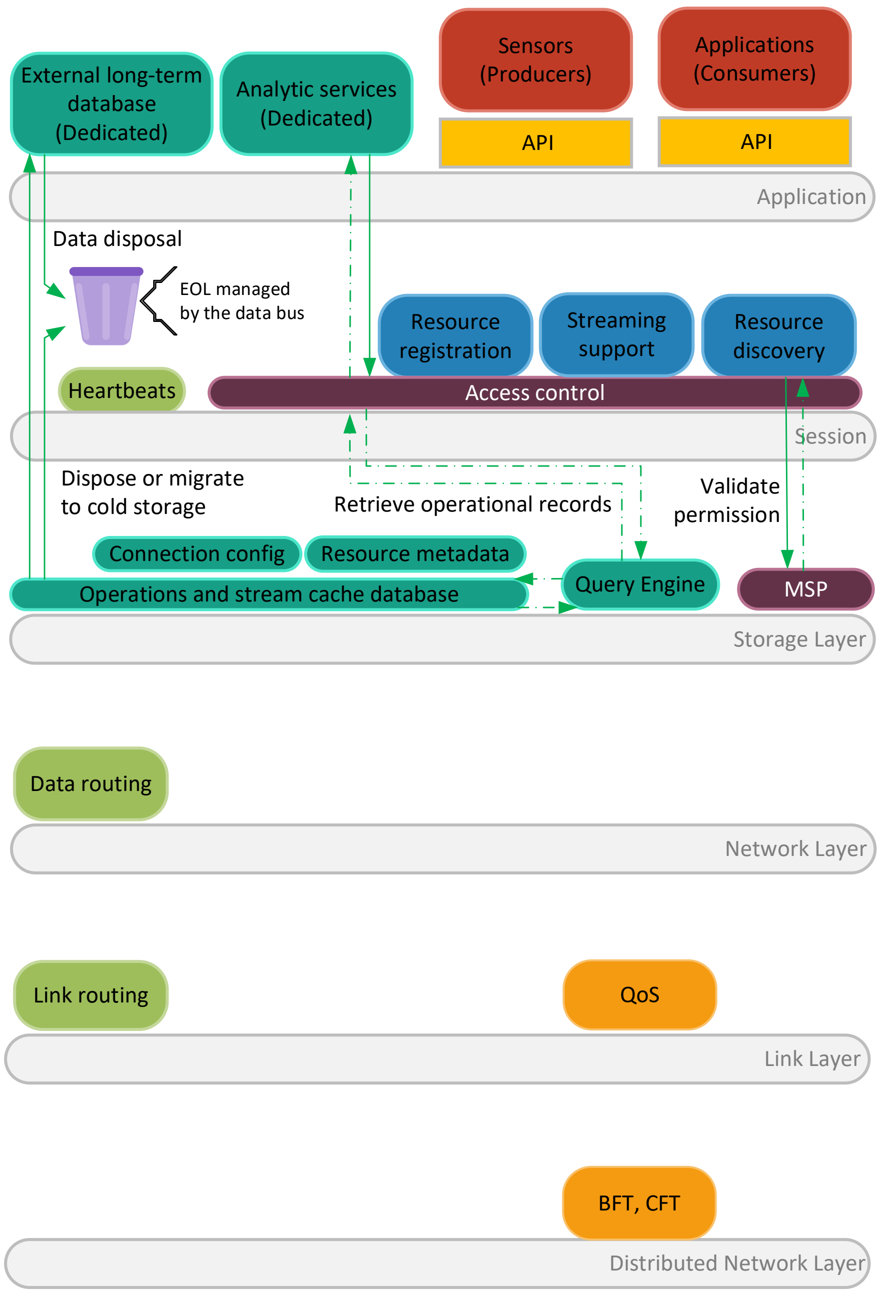

Figure 10. DGSS Life Cycle Supervision Phase. EOL is end of life.

4.1.5 Analysis and End-of-Life Phase

The Analysis and End-of-Life phase, shown in Figure 11, is responsible for handling the actor's terminal state. Such a state can be reached from two fronts:

- If a producer gracefully (or abruptly) terminates the connection to the data bus, data input stops and a chain reaction begins that disconnects all data consumers (which are notified of the event). In addition, all relevant producer-to-consumer mappings are invalidated (e.g., input/output buffers enter a decommissioned state and filter handlers are dropped).
- If a data consumer terminates the connection (gracefully or abruptly), a cleanup operation begins from the consumer side. Within this context, output buffers enter a decommissioned state and filter handlers are dropped (unless other consumers share the resource).

Notice that when a buffer is terminated, data may optionally be moved to the long-term database or discarded (based on the actors' profile configurations). Also at this point, in-depth analytics can be executed to gain visibility into historical events (e.g., connection start, data moves).

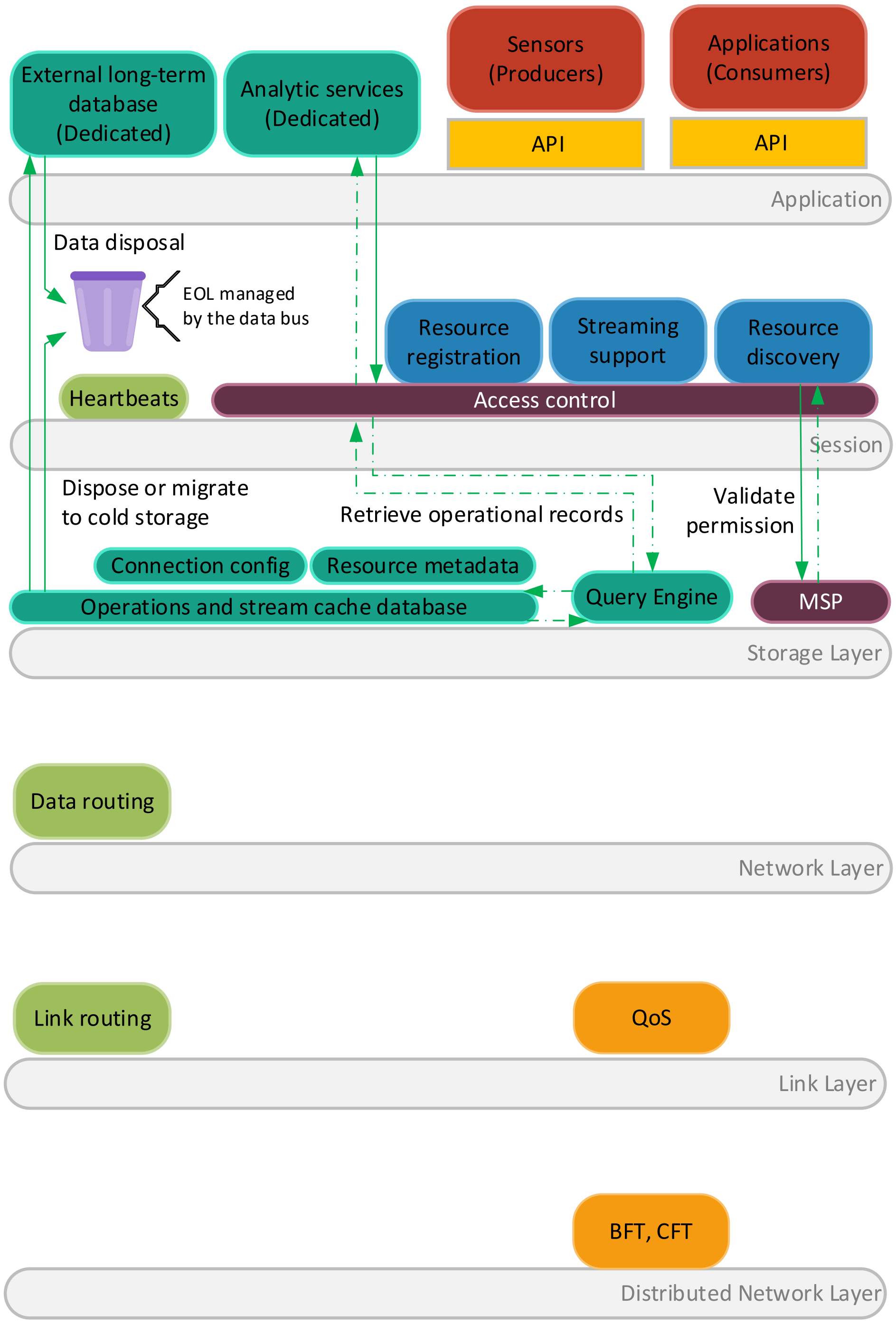

Figure 11. DGSS Analysis and End-of-life Phase

4.2 GridAPPS-D Platform Description

As mentioned in Section 2.3.2, the DGSS will leverage the GridAPPS-D platform and its standards-based API for application portability. The current section provides a general description of the GridAPPS-D platform.

GridAPPS-D is an open-source platform developed by Pacific Northwest National Laboratory. It is a standards-based software platform for data integration and application development for advanced electricity distribution management and operations. This platform was developed to reduce the cost and time required to integrate advanced functionality into distribution operations, to create a more reliable and resilient grid. GridAPPS-D has implemented many applications and services needed for distributed operation, such as state estimation service, model validation, distributed optimal power flow, and DNP3 service (IEEE 2010). Figure 12 shows a conceptual architecture of GridAPPS-D with functional elements such as these:

1. **Tools** help developers enhance the functionality of their applications. Examples might include off-line power flow, optimization toolboxes, state estimators, or statistical processing.
2. **I/O** (input/output) allows convenient access to the power system model and data through standards-based queries and messages. Conversely, applications can send control signals to the simulator using standard message schemas.
3. **Development utilities** include loggers, debuggers, access control, test managers, user interface toolkits, and other application support functions.
4. **Data bus** is based on industry standards like IEC 61968 (IEC 2003) and IEC 61970 (IEC 2022) (i.e., the Common Information Model), plus more to be identified in future work.
5. **Distribution simulator** represents the power system operating in real time. Initially, this will be GridLAB-D (DOE 2022), but future versions may include the Electric Power Research Institute, Inc.'s, OpenDSS (EPRI n.d.), ns-3 for communications (nsnam 2022), and other federated co-simulators.
6. **Field Bus Manager** at Data Ingest is an application that allows exchange of data with commercial tools. Field Bus Manager will provide API functions to the applications for accessing the network and device data model and interacting with the grid devices in a standardized format.

In GridAPPS-D, the platform data and message bus are managed by the GridOPTICS Software System (GOSS) (Akyol, et al. 2014), and the time clock and message traffic between platform simulators are managed by the Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS) (GMLC 2022). GOSS is a comprehensive software framework developed to integrate various software capabilities in GridAPPS-D. Use of GOSS enabled plug and play of various analysis, modeling, and visualization software tools to improve the reliability and efficiency of the grid.

The HELICS-GOSS bridge is a data management infrastructure that serves as input and output (for the publication/subscription mechanism) from the simulator to the rest of GridAPPS-D. It receives initialization, time step, and update and finalize requests from the simulation manager and other applications. It publishes output from the simulator on a predefined topic for the simulation manager and other applications. GridAPPS-D uses the field-interface concept to map among the data models used in the field devices (Sharma, et al. 2022). The field-interface manager has an added provision to support different types of protocols for integrating different applications. This featurewill create an integrated application for exchanging data using different standards.

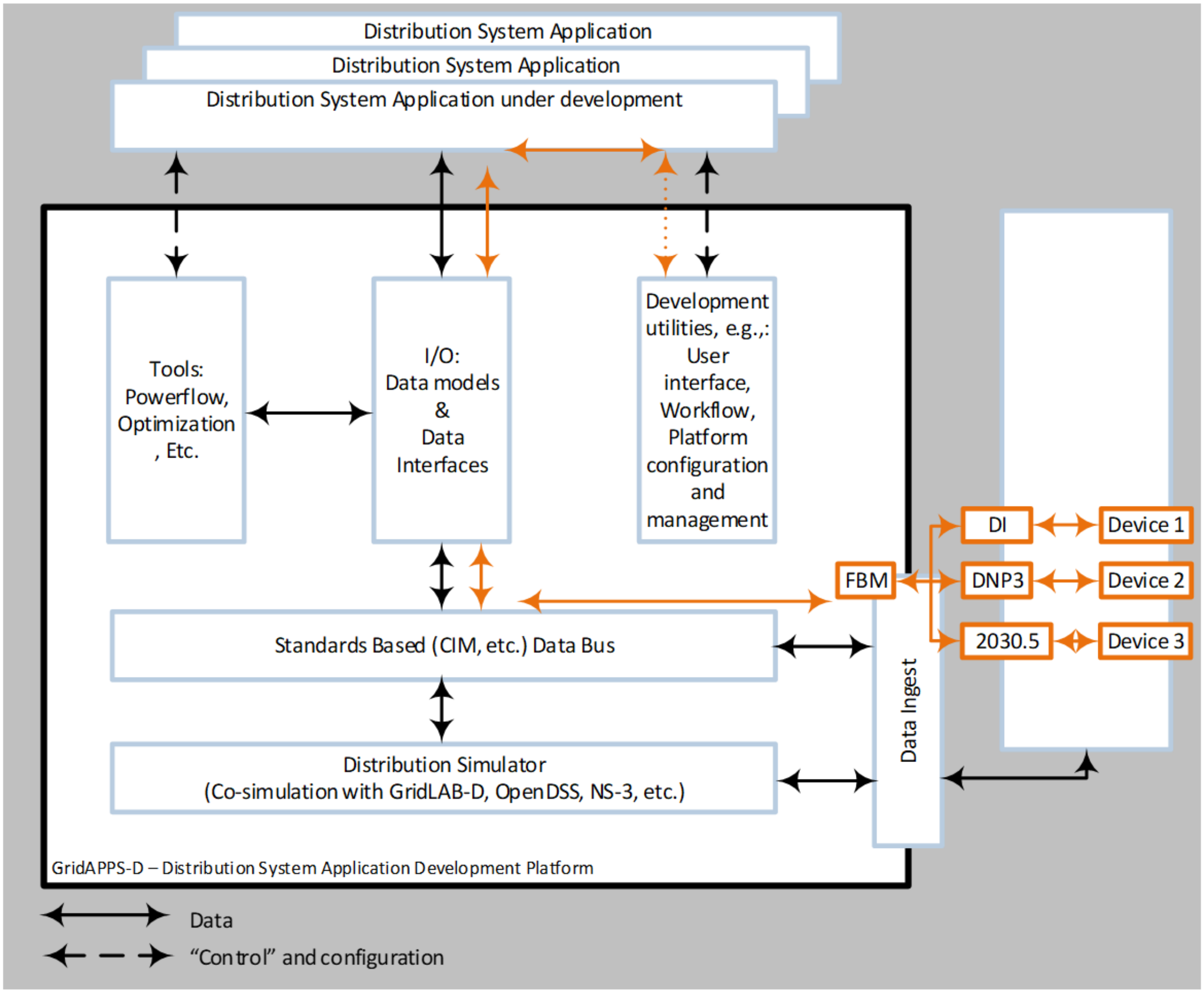

Figure 12. GridAPPS-D Conceptual Architecture [Adapted from (Sharma, et al. 2022)]

4.3 The Common Information Model

GridAPPS-D uses the CIM standard and its relevant extension to power systems, covering the majority of IEC 61970, IEC 61968, and IEC 62325 (IEC 2018) for all the grid applications and its own services. CIM is extendable and already covers a lot of power grid-related elements and functions. It provides a common semantic understanding for systems and subsystems in the platform. The CIM standard is developed by the IEC and represents a generalized model of all utility objects and their relationships. It is an object-oriented representation of a power system. It uses mRIDs (master resource identifiers) for measurements. The CIM provides the blueprints for application software like GridAPPS-D to represent data structures, message payloads, and information exchanges between applications.

CIM messages for substations, such as those shown in Figure 13, use the Unified Modeling Language (UML) to represent a distribution system model. In GridAPPS-D, UML diagrams are used as a human-readable, intuitive profile descriptions of application-specific uses of the extended CIM. In the GridAPPS-D platform, various applications and services exchange data using a publish/subscribe mechanism or Python API. A publish/subscribe mechanism can be implemented using any of the language bindings for ActiveMQ messaging framework (Apache 2022). There are various input/output topics for the data exchange. For example, a simulation message from the simulator looks like the code in Figure 13, where each component in a feeder has an mRID and the magnitude and angle are sent in a JavaScript Object Notation (JSON) format.


```
{
    "simulation_id": "12ae2345",
    "message": {
        "timestamp": "1357048800",
        "measurements": {
            "123a456b-789c-012d-345e-678f901a234b": {
                "measurement_mrid": "123a456b-789c-012d-345e-678f901a234b"
                "magnitude": 3410.456,
                "angle": -123.456
            }
        }
    }
}
```


Figure 13. Example CIM Message from Simulation Output Topic

In Figure 12 the architecture shows all the applications that view devices as sets of available functions and data points within the common data model. Devices, on the other hand, only see messages from applications or other devices in their native communications protocols.

Syntactic interoperability is trickier, because there is a vast number of relevant communications standards and proprietary technologies. For implementation within the GridAPPS-D platform, JSON is the selected data exchange format (ECMA 2013).

The GridAPPS-D DNP3 (Singh, et al. 2020) service running on the GridAPPS-D server subscribes to all messages on the simulation output topic, and it expects them to be in CIM format. The GridAPPS-D server is the outstation in this case and the SCADA (supervisory control and data acquisition) is the master server, which polls the outstation every 60 seconds (this frequency can be modified as needed). The first step of integrating GridAPPS-D with SCADA was to create a service code that can connect to the master, send the SCADA points where it will integrate, update the real-time values, and process the input from the master. The application service consists of multiple Python codes. The conversion code parses the CIM model dictionary file containing the model data and converts the data into DNP3 data points, then sends it to the master. When it receives a control command from the master (in DNP3 format), it converts the command into CIM format and finally updates the status of device. The complete loop operates in real time and can accept commands from devices at multiple internet protocols and at multiple ports. We have tested this application with a 9500-node model that had ~30K DNP3 points.

GridAPPS-D IEEE **2030.5** service (IEEE 2018) (Sharma, et al. 2022) is designed to implement a REST (representational state transfer) architecture. This service can dynamically generate end devices and client/device certificates from GridAPPS-D. During the runtime, this server will update the end devices data through the GridAPPS-D Field Bus Manager API. The 2030.5 server and client enable the integration of DERs and the devices using this protocol. The devices connected through the field bus interface send the information to the GridAPPS-D platform. The service deployed in the platform translates the 2030.5 data to CIM and vice versa.

Other than the CIM model, DGSS data bus also supports **Open Field Message Bus (OpenFMB)** (UCAIug 2022) to inform the extended grid state data definitions through different DGSS services. OpenFMB supports distributed intelligence and grid edge interoperability that leverage existing standards to federate data between devices.. Like the CIM, OpenFMB framework also uses UML to represent use-case requirements and exchange information between actors.

5.0 Use Case Detail

Employing use cases is a valuable way to document applications and processes to define requirements. Use cases provide a representation of the way a system, and its associated actors/stakeholders, will be used to achieve specific goals that address one or more business objectives. Business drivers are the impetus for the creation of any utility system. To satisfy the needs of the business, certain stakeholders require one or more features to be available; these are described in this document as user stories. User stories are a concise means to define goals and reasons for a feature from the perspective of a single actor. The realization of user stories involves one or more use cases, which generate one or more systems.

In the upcoming Fiscal Year 2023, the project team plans to implement a reference use case on top of the DGSS platform, enabling the team to accurately measure the platform's capabilities and identify possible shortcomings. As a value-added proposition, this demonstration will create a sample use case that can then be modified or expanded by the end user. In this section, we'll describe the planned use case and describe its primary functionalities.

Use Case Title – Energy Schedule Service Use Case

Use Case Description – As the power grid transitions from dependence on the transmission system and large generators for the supply of electricity and as distributed resources, including renewables, provide our electricity, we see markets evolving to allow the participation of such resources. Coordinating these assets on a large scale can be challenging. Market participation requirements such as scheduling, measurement and validation, and settlement add additional challenges that could benefit from a distributed sensor network and the associated data. The following use case explores how a distributed sensor network architecture might benefit market activities related to a simple energy schedule service.

A distribution system operator (DSO) DER management system (DERMS) platform uses sensor data from the distribution level to inform decisions related to the balance and stability of the electricity grid. The DERMS has the role of a subscriber but also could act as a publisher to provide sensor data to the DSO or to third-party actors, which may include device vendors, (e.g., ecobee or Tesla). This bidirectional data flow can help with coordinating various assets, validating that the performance expectations of a grid service were met, and appropriate compensation is offered. A DER facility may have multiple sensors, even if they only account for interval service metering. For example, a microgrid such as that for Pacific Northwest National Laboratory's Marine Coastal Research Laboratory has two service entrances but is billed cumulatively. The AMI would act as a publisher, providing sensor data to other users for purposes of billing, or in the case of providing grid services, performance validation. Other publishers could include individual assets such as electric vehicle service equipment, which is typically accessed via the cloud, but for this use case could be posted to a message or data bus. To abstract an individual asset including its type, these publishing actors could be considered DER Facilities, including one or many DERs. These devices would be managed individually by the DER management function, as shown in Figure 14.

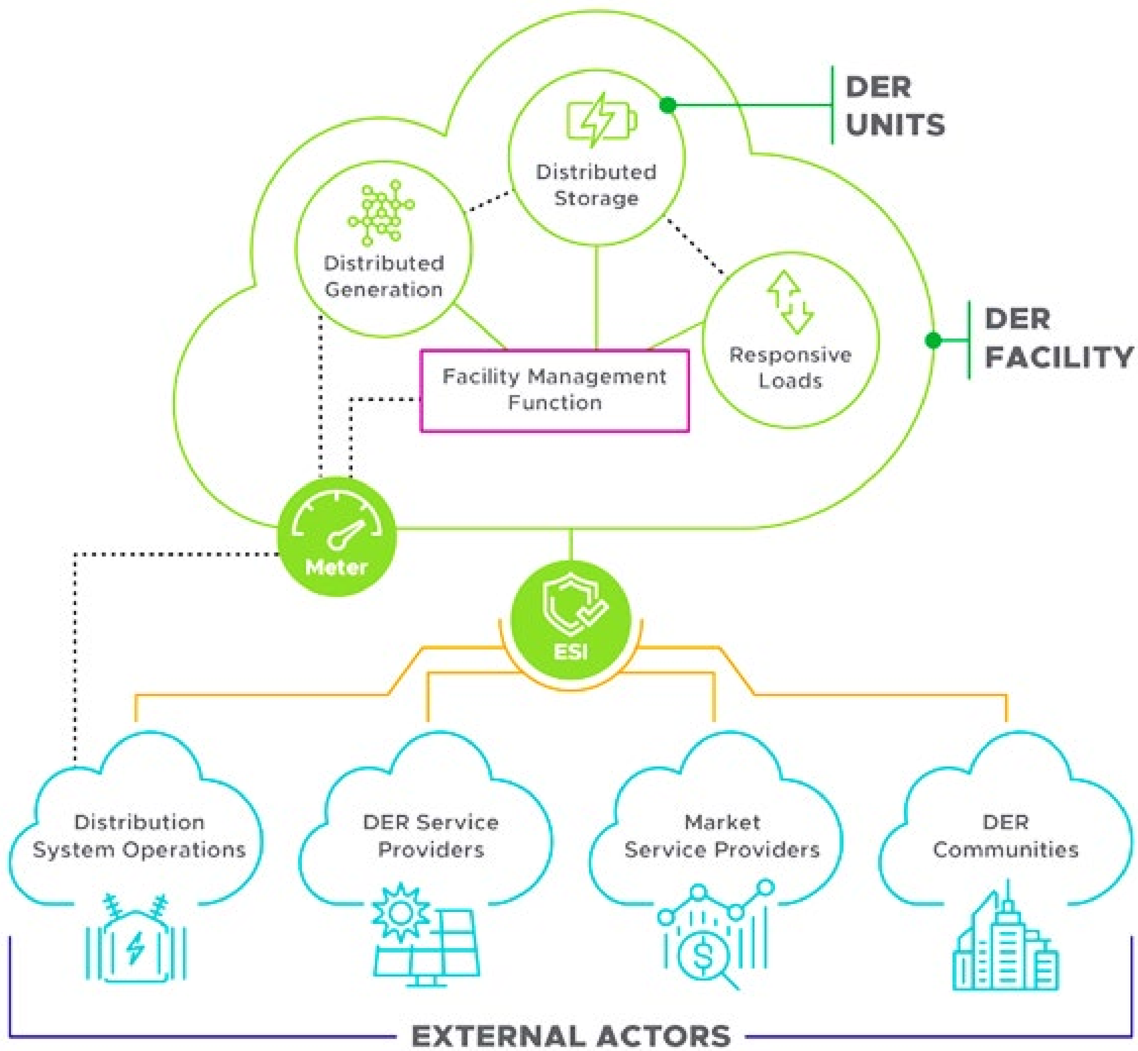

Figure 14. DER System Structure. The DER Facility May Include One or Many DERs, Which Are Managed by the DER Management Function (GMLC 2018). ESI is electrical service identifier.

This abstraction allows us to focus on the interaction between a customer and a grid operator rather than the control of a specific DER, preventing exposure of what could be considered private behaviors behind the customer premises AMI.

Consider the case of energy schedule service. This service requires that energy provided over a scheduled period meets a certain performance metric. In this case, the DSO DERMS platform is subscribed to sensor data related to grid balance and stability. The DERMS platform understands that during a period of peak loading from 5 p.m. to 7 p.m., balancing the distribution network will require that 5 MW of deficit will be incurred. To avoid this deficit, 5 MW can be either shed or generated. In this sense, load reduction is the same as generation that will be needed in a region.

Based on 10 prior days' sensor data that has been evaluated by the DERMS, the platform can average the data from a particular service location meter and predict what the likely consumption or production will be at a given future time on a given point. The DERMS can make a request or post a price per megawatt to incentivize a DER facility to reduce consumption or generate energy. For the purposes of this example, control decisions are less important as we focus on how the sensor data will be used.

The DERMS reads the published sensor data from the message bus to determine whether the DER facility adjusted its performance accordingly. After the measurement and verification have been performed, settlement can occur, and a preselected subset of sensor data is placed in long-term storage for the DSO’s billing purposes and for distribution system planning. Planning would include studying changes in average demand and peak loading, and prioritizing infrastructure improvements.

6.0 Future Work

From an implementation aspect, the DGSS is a service-oriented, software-defined piece of middleware that sits between the applications and the sensor network plane to provide a reliable DDS. A successful middleware design and implementation must be able to support near-real-time or real-time heterogeneous data transmittal while also being able to filter, route, and deliver data according to the end-application requirements. Aiming toward these goals, this work has presented an architectural blueprint that effectively presents itself as a virtual communication plane, capable of providing network- and device-agnostic connectivity while meeting the security and reliability goals. The resulting architecture abstracts the component-level differences introduced by the data formats/protocols, network transport mechanisms, and operating systems and unifies them in a highly interoperable environment. Therefore, the DGSS concept simplifies the overall integration process by enabling solution developers to focus on satisfying end-application goals and functionalities rather than addressing the low-level interoperability and connectivity requirements.

In Fiscal Year 2023, the project team will focus on developing a reference implementation of the DGSS middleware based on the architecture presented herein. Also, a utility use case will be demonstrated on this platform to support DGSS functionality. The DGSS is expected to leverage the GridAPPS-D platform along with its standardized communication APIs (e.g., during later implementation and testing phases). In addition, existing standards such as the IEC Common Information Model, IEC 61850, and North American Energy Standards Board OpenFMB will be bridged into the platform to assess its capabilities and performance under real-world conditions.

Although the DGSS has been designed with a data-agnostic mindset, future iterations will embrace the CIM as its main, transport-level data format. This is expected to enable future interoperability while also enabling the team to more easily adopt and transition into the GridAPPS-D platform.

Existing GridAPPS-D capabilities may enable the DGSS to expand its user base, because the GridAPPS-D platform can already exchange data with external devices/systems using protocols such as Distributed Network Protocol and IEEE 2030.5 via its Field Bus Manager (Sharma, et al. 2022). This may aid in supporting future interconnections between DNP3 and SCADA systems. Similarly, the 2030.5 service is under development and may enable the connection of DERs and behind-the-meter inverters to the distribution control centers.

7.0 References

Akyol, Bora A, Selim Ciraci, Tara Gibson, Mark Rice, Poorva Sharma, Jian Yin, and Craig Allwardt. 2014. *GridOPTICS Software System.* February 24. https://www.osti.gov/biblio/1231728.

Apache. 2022. *ActiveMQ: Flexible & Powerful Open Source Multi-Protocol Messaging.* Apache Software Foundation. https://activemq.apache.org/.

Bhattarai, Bishnu P, Sumit Paudyal, Yusheng Luo, Manish Mohanpurkar, Kwok Cheung, Reinaldo Tonkoski, Rob Hovsapian, et al. 2019. "Big data analytics in smart grids: state-of-the-art, challenges, opportunities, and future directions." *IET Smart Grid* (International Electrotechnical Commission) 2 (2): 141-154. doi:10.1049/iet-stg.2018.0261.

DOE. 2016. "GridAPPS-D." U.S. Department of Energy. https://gridappsd.readthedocs.io/en/latest/hosted_applications/index.html.

—. 2022. *GridLAB-D: A Unique Tool to Design the Smart Grid.* U.S. Department of Energy. https://www.gridlabd.org/.

ECMA. 2013. *The JSON Data Interchange Format.* Standard, Geneva, Switzerland: Ecma International. https://www.ecma-international.org/wp-content/uploads/ECMA-404_1st_edition_october_2013.pdf.

EPRI. n.d. *OpenDSS.* Electric Power Research Institute. https://www.epri.com/pages/sa/opendss.

GMLC. 2022. *Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS).* Department of Energy Grid Modernization Laboratory Consortium. https://helics.org/.

GMLC. 2018. *Interoperability Strategic Vision.* PNNL-27320, Grid Modernization Laboratory Consortium, Washington, D.C: Department of Energy. https://gmlc.doe.gov/resources/interoperability-strategic-vision.

IEC. 2003. *Application integration at electric utilities - System interfaces for distribution management.* Standard, Geneva, Switzerland: International Electrotechnical Commission.

IEC. 2022. *Communication Networks and Systems for Power Utility Automation.* Standard, Geneva, Switzerland: International Electrotechnical Commission. https://webstore.iec.ch/.

IEC. 2022. *Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base.* Standard, Geneva, Switzerland: International Electrotechnical Commission. https://webstore.iec.ch/publication/62698.

IEC. 2018. "Framework for energy market communications - Part 301: Common information model (CIM) extensions for markets." Standard, International Electrotechnical Commission, Geneva, Switzerland. https://webstore.iec.ch/publication/31487.

IEEE. 2010. *IEEE Standard for Electric Power Systems Communications -- Distributed Network Protocol (DNP3).* New York, NY: Institute of Electrical and Electronics Engineers. doi:10.1109/IEEESTD.2010.5518537.

IEEE. 2018. *IEEE Standard for Smart Energy Profile Application Protocol.* New York, NY: Institute of Electrical and Electronics Engineers. https://standards.ieee.org/ieee/2030.5/5897/.

IEEE/ISO/IEC. 2010. *Standard for Information technology -- Smart transducer interface for sensors and actuators -- Part 4: Mixed-mode communication protocols and Transducer Electronic Data Sheet (TEDS) formats.* Institute of Electrical and Electronics Engineers, International Organization for Standardization, and International Electrotechnical Commission. https://www.techstreet.com/ieee/standards/ieee-iso-iec-21451-4-2010?product_id=1771626.

James P Ogle, Ronald B Melton, Kevin P Schneider, and Rohit Jinsiwale. 2021. "Enhancing Responsiveness and Resilience with Distributed Applications in the Grid." *2021 IEEE Rural Electric Power Conference (REPC).* Savannah, GA. 15-20. doi:10.1109/repc48665.2021.00012.

Muammar, Nada. 2022. *What are the cost drivers for system integration in SAAS for the utilities?* May 12. https://aplines.com/what-are-the-cost-drivers-for-system-integration-in-saas-for-the-utilities/.

nsnam. 2022. *ns-3 Network Simulator.* nsnam. https://www.nsnam.org/.

Rathwell, Gary. n.d. *PERA Enterprise Integration Web Site.* http://www.pera.net/.

RTI. 2022. *What is a Databus?* Real Time Innovations. https://www.rti.com/products/what-is-a-databus.

Sharma, Poorva, Artis Riepnieks, Andrew Reiman, Alka Singh, Ronald Melton, James Ogle, and Craig Allwardt. 2022. "Model-Based Interface Design for Smart Field-Device Integration." *2022 IEEE Rural Electric Power Conference (REPC).* Cleveland, OH: IEEE. 31-37. doi:10.1109/repec55671.2022.00014.

Singh, Alka, Andrew Fisher, Craig Allwardt, and Ronald B Melton. 2020. "A Data Exchange Interface for a Standards Based Data Integration Platform." *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT).* Washington, D.C: IEEE. 1-5. doi:10.1109/ISGT45199.2020.9087646.

Taft, J D. 2017. *Sensor Network Issues for Power Grid.* PNNL-27309, Richland, WA: Pacific Northwest National Laboratory. https://gridarchitecture.pnnl.gov/media/advanced/Sensing_for_Advanced_Grids_v1_7_GMLC.pdf.

Taft, J D, and A Becker-Dippmann. 2015. *Grid Architecture.* PNNL-24044, Richland, WA: Pacific Northwest National Laboratory. doi:10.2172/1176825.

Taft, J D, Z Li, and E Stewart. 2019. *Extended Grid State Definition.* PNNL-SA-141027, V3.3, Richland, WA: Pacific Northwest National Laboratory. doi:https://gmlc.doe.gov/resources/extended-grid-state-definition.

UCAIug. 2022. *Open Field Message Bus (OpenFMB).* UCA International Users Group. https://openfmb.ucaiug.org/.

Xue, Yaosuo, Guodong Liu, Jeffrey D Taft, and Mallikarjun Shankar. 2022. *Emerging Trends and Systemic Influencing Today's U.S. Electric Grid.* GMLC-1.2.1; ORNL/TM-2022/2372, Oak Ridge, TN: Oak Ridge National Laboratory. https://info.ornl.gov/sites/publications/Files/Pub173902.pdf.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov