GE VERNOVA

DIGITAL TRANSFORMATION OF ELECTRIC GRID OPERATIONS TECHNOLOGY DRIVERS

An analysis of
five critical technologies
that power the digital
transformations
of electric utilities.

GridOS® Orchestration Software

CONTENTS

CONTENTS

EMBRACING DIGITAL TRANSFORMATION IN THE ELECTRIC SECTOR

Since the introduction of the first electric grids in the 1880s, the industry has undergone a remarkable journey of technological and regulatory evolution (Figure 1). This journey has shaped the way energy is produced, distributed, consumed, and transacted worldwide. More recently, driven by the growing urgency of climate change, the world is stepping into a more significant energy transition – an entirely different reality, characterized by an exponential rate of change. As a result, the electric sector is at the precipice of unprecedent opportunities and challenges.

The urgency to reduce carbon emissions continues to spawn a wide range of new energy, consumer, and adaptive management technologies, thus increasing the complexity and diversity of devices connected to the grid. In turn this dramatically alters its dynamics, especially in terms of voltage, frequency, and angle stability.

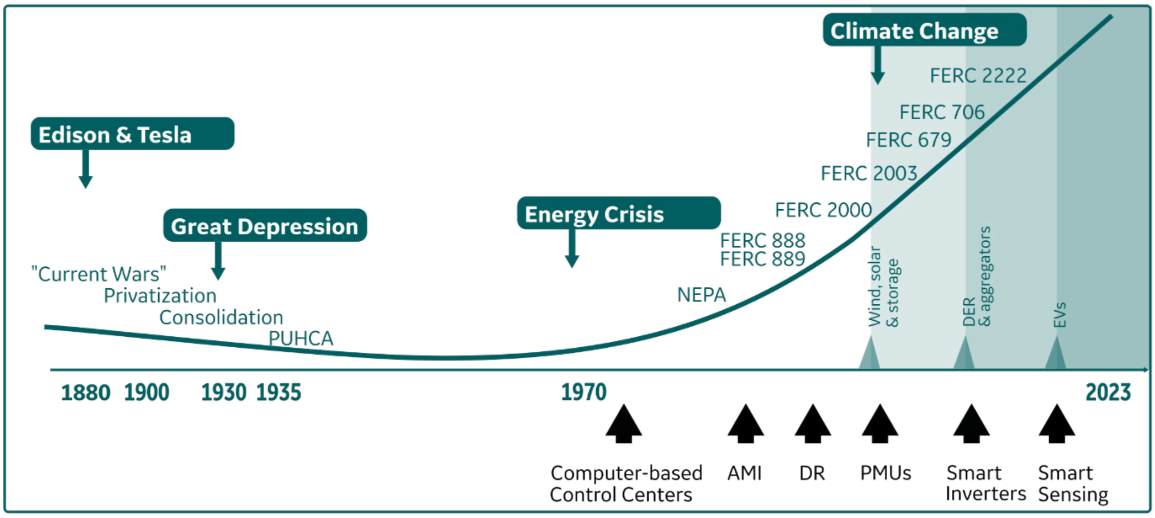
Putting some of the required headline transformations into numbers, the world should deploy 1,000GW worth of renewable power every year, increase the share of green energy in the global mix to 77%, and invest around US$103 trillion[1] to achieve Net Zero greenhouse gas (GHG) emissions by 2050. Regrettably, these numbers continue to increase as the targets are often missed (for example, only 300GW worth of renewable generation were deployed in 2022).

This unprecedented transformation of the grid cannot be achieved without an accompanying digital transformation. Embracing digital transformation will be the foundation to operating the clean energy grid in a reliable and affordable way. In 2022, investments in digital-related grid efficiency grew to a new high of US$63 billion, including US$1.5 billion in analytics for grid operations and asset management, a trend reinforced by the European Union's Digitalizing the Energy System Plan, and countrywide plans such as the United Kingdom's Energy Digitalization Taskforce[2].

The unique combination of massive infrastructure investments, technology advancements, new market frameworks, , innovation ecosystems, and even talent is pushing the traditional business model to its limits and creating astonishing opportunities for a new and digital clean energy industry.

The drive to facilitate and accelerate this journey is fundamentally shifting how grids are managed. Beyond merely adapting to the current challenges, the right digital strategy should empower companies to thrive, explore new business models, and become pioneers in the sustainable energy future, moving from reactive grid management to proactive and policy-guided grid orchestration.

Figure 1: Expansion of North American Regulatory and Technology Complexity



[1] World Energy Transitions Outlook 2023: 1.5C Pathway. International Renewable Energy Agency (IRENA), 2023.
[2] International Energy Agency (IEA), Decarbonization Enablers. July 2023.

GRID ORCHESTRATION

In the face of the ever-growing complexity of modern power systems, an essential concept emerges: grid orchestration. This pivotal approach involves the intelligent coordination and optimization of diverse elements within the electricity grid, allowing it to seamlessly function and adapt to variable and intermittent conditions in real-time. By integrating cutting-edge technologies, harnessing data-driven insights, and leveraging advanced control systems, grid orchestration collaboratively optimizes grid operations, ensuring a steadfast, efficient, and secure electricity supply.

At the heart of grid orchestration lies data – vast amounts of real-time information collected from diverse grid assets and energy resources. Data is redefining the way utilities embed technologies across their business to improve efficiency and empower grid operators to make informed decisions. It is paving the way to build the tools and flexibility required to address the core component of the current transition: uncertainty.

By leveraging advanced data analytics and artificial intelligence algorithms, grid orchestration enables predictive modeling of electricity demand and generation patterns. This foresight empowers grid operators to anticipate fluctuations and ensure the seamless integration of renewable energy sources, such as wind and solar, into the grid. Additionally, data-driven insights optimize energy storage technologies, like batteries, ensuring efficient storage and release of excess energy during peak demand periods, thereby enhancing grid stability.

However, as data becomes increasingly central to grid orchestration, ensuring grid security is paramount. Cybersecurity measures play a critical role in safeguarding the grid's critical infrastructure from potential threats and attacks. As data transmission and communication between grid assets intensify, implementing robust cybersecurity measures becomes imperative to prevent disruptions and potential breaches.

New technology advancements constantly reshape the landscape. The integration of Internet of Things (IoT) devices, smart sensors, and advanced communication and control systems enhances real-time monitoring and coordination. These technologies create a dynamic and diverse data portfolio that will be necessary to support rapid adaptation to changing conditions, continually optimizing grid performance. While a portion of such data will be collected using established SCADA protocols, the majority will be acquired through new, dedicated, purpose-built interfaces based on industry standards such as IEEE 2030.5.

As utilities and grid operators develop and implement comprehensive data-driven strategies, they can increasingly better harness the full potential of their assets and infrastructure. This can be supported with strategic collaborations with external partners, startups, research institutes, utilities, and other ecosystem players to further enrich knowledge, drive progress, and deliver an attractive environment to new talents. At the same time, collaboration provides access to outsourced, commoditized, and non-critical knowledge.

The digital utility model marks a paradigm shift in the approach to developing and implementing energy management systems. The traditional approach of turnkey projects, with fixed and lengthy timelines, is rapidly becoming outdated. Instead, the industry is moving towards a more agile and incremental approach, driven by the need to stay adaptable in a dynamic energy landscape.

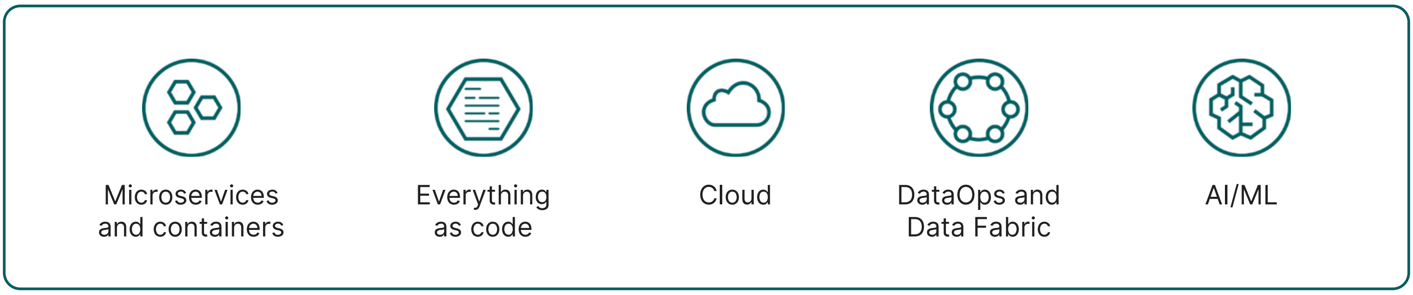
In this new era, the focus moves from a technology- and functional-driven perspective to a value-driven investment one. The goal is to deliver software solutions that precisely cater to specific needs and objectives, at the right scale and precisely when necessary. This means moving away from one-size-fits-all solutions and embracing modular applications that align with the unique requirements of a given utility.

Creating a robust technology platform is crucial for enabling this transition. This platform must exhibit embedded scalability, flexibility, and cyber-security to optimize time-to-value and reduce the overall cost of ownership. The platform serves as the foundation upon which utilities can build and customize data-driven grid orchestration and management systems, seamlessly adapting to evolving challenges and demands. Key technologies that will underpin this platform have individually transformed other industry sectors. Combining and applying these same technologies will provide the basis of the digital transformation in the energy sector.

KEY TRANSFORMATIVE TECHNOLOGIES

The digital transformation that unlocks and enables modern grid orchestration must rely on more than technology alone, and include fundamental shifts in organizational structure, business models, and culture. Nonetheless, technology will still play a critical role. Over the next two to five years, five key technologies will have a profound impact on the way electric utilities build, deploy and maintain grid orchestration systems (Figure 2). These five technologies enable key transformations both in unison and on their own:

Figure 2: Key Transformative Technologies



MICROSERVICES AND CONTAINERS

Breaking down applications into smaller, manageable microservices helps navigate the increasing complexity of systems as the number of functions and their interactions grow. Microservices – and the container technologies currently used to implement them – bring modularity, scalability, and mechanisms to manage the complexity of the expanding interplay of functions of large software systems. When combined with efficient messaging, container orchestration, and modular cybersecurity, microservices form the basis for delivering extensible, event driven, Zero Trust-secured architectures.

EVERYTHING AS CODE (EAC)

EaC is an approach and mindset that advocates representing all aspects of an application's lifecycle, infrastructure, and configuration as code – human-readable and version-controlled files that can be executed by machines. Embracing a code-centric approach to infrastructure and configuration management simplifies processes, enhances collaboration, and accelerates development and deployment cycles. Infrastructure as code (IaC), application composition as code, workflows and configuration as code, and more, all allow modern version control systems and deployment mechanisms to dramatically simplify deployment automation, rollout, rollback, and change control in a modern software solution. This enables a secure and fast evolution of software functions, with significant time-to-value benefits.

CLOUD

Leveraging the power of cloud computing liberates utilities from physical infrastructure constraints and opens the door to resource elasticity, capacity on demand, and time of use-based billing. Moreover, adoption of cloud-native technologies can unlock some of the advantages of cloud architectures, like flexible resource utilization, to isolated, on-premise deployments. Cloud-native architectures allow the cloud to be readily adopted when the cost, resource, regulatory, and security dimensions make sense.

DATAOPS AND DATA FABRIC

Streamlining data operations and implementing a DataOps approach ensures that data flows seamlessly across the entire organization. This in turn fosters the data-driven decision making that empowers utilities to harvest valuable insights and optimize their strategies. A data fabric affords the technologies that allow on-demand access to data across the organization. It provides the ability to easily find, integrate with, and combine disparate data sources to transform data into actionable intelligence. It provides the infrastructure, governance, and life-cycle management for data-centric analytics, simplifying the process of extracting value from data and supporting increasingly diverse data landscapes.

]

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (AI/ML)

The integration of AI/ML capabilities revolutionizes utilities' ability to process and analyze vast amounts of data, uncover patterns, and predict future trends. This intelligence alongside physics-based models enables predictive actions, improved risk management, and forecasting (e.g., load and generation), all of which maximize efficiency and reliability. The ease of access to well-managed data enables AI/ML analytics to be developed and deployed alongside physics-based analytics. Securely delivering AI/ML in operational environments will enable leveraging progressively intelligent decision support and policy-guided smart automation.

> Far from being mutually exclusive, these technologies are highly complementary with impacts that are not only augment one another but, in some cases, amplify one another. **Table 1** summarizes the primary (blue ✓) and subordinate non-functional advantages of each.

Benefit	Micro-services	Everything as Code	Cloud*	Data Fabric	AI/ML
Improved resource utilization	✔	✔	✔		
Time to value / speed	✔	✓	✔	✔	✓
Extensibility	✓			✓	✔
Data discovery and federation				✓	
Automation enablement		✔	✔	✔	✓
Horizontal scalability	✓		✓		
Reduced admin overhead	✔	✓	✔	✔	
Improved modularity	✓			✔	
Improved maintainability	✔	✓		✔	✔
Capacity on demand			✓	✔	
Pay only for what you need			✓		
Availability of managed services			✓		

*Benefits achievable will vary depending on cloud type

The following sections introduce and explore each of these technologies and discuss their specific impacts in more detail.

MICROSERVICES, CONTAINERS, AND BEYOND

Microservices can best be thought of as a small collection of closely related processes that collectively operate to form a single, independent application. Each microservice generally focuses on a specific business capability, providing independent lifecycle management and efficient scaling. To better visualize the benefits of microservices, consider monitoring the massive expansion of distributed energy resource (DER) devices. Some of the most valuable microservice functions for this context include:

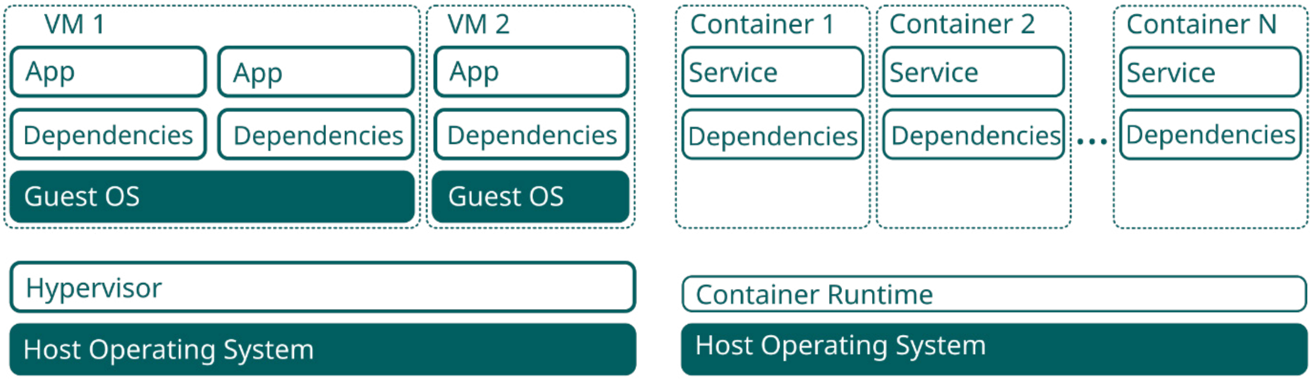
- Collecting data from a large and growing population of DERs and aggregators; for example, using the IEEE 2030.5 internet-based protocol
- Consolidating the DER data into a physical model of the distribution network to identify constraints and perform optimizations
- Generation of alarms identifying equipment and constraint risks requiring operator attention

While a solution could, in theory, be architected as a single, colossal application, microservices provide a better and more granular approach. By breaking up a given application into separate microservices, one could have a larger pool of microservices focused on receiving and processing millions of DERs' publishing data, a small number of microservices independently computing power flow for different segments of the network, and one or two alarm-processing services. Each of the services are capable of scaling dynamically based on actual demand, providing deterministic response while minimizing the required resources.

CONTAINERS

A container is a standard unit of software that packages up code for an application and all its dependencies. As such, the resulting process runs quickly and reliably and is portable between different computing environments. Containers enable an application to bring along its unique dependencies independently of the host operating system. The resulting containers may include one or more processes and all their associated dependencies including executables, libraries, and configuration files – but unlike virtual machines, they do not carry a copy of the guest operating system. This makes them more atomic, portable, and lightweight. While different, container and virtual machine (VM) technologies are not mutually exclusive – containers can run on virtual machines in addition to bare metal or cloud. A visual comparison of hardware virtualization and containerization is provided in Figure 3.

Figure 3: Comparison of virtual machines and containers



Containers can easily run on a computer cluster – a set of computers that work together so they can be viewed as a single system. Clusters are usually deployed to improve performance and availability over that of a single computer in a cost-effective manner. Container runtimes are software components that run containers on the individual host operating systems. They are responsible for loading container images from a repository, monitoring local system resources, isolating system resources for use of a container, and managing the lifecycles of individual containers running on every compute node in the cluster. Container runtimes work together with container orchestrators to provide management of the cluster and its associated containers through the standard container runtime interface (CRI).

Current container technology based on the Open Container Initiative (OCI) standards is fast evolving. Although future containers will include a diversity of technologies, such as Web Assembly (WASM), the principles of portability, isolation, and orchestration (see below) will not just continue, but also improve.

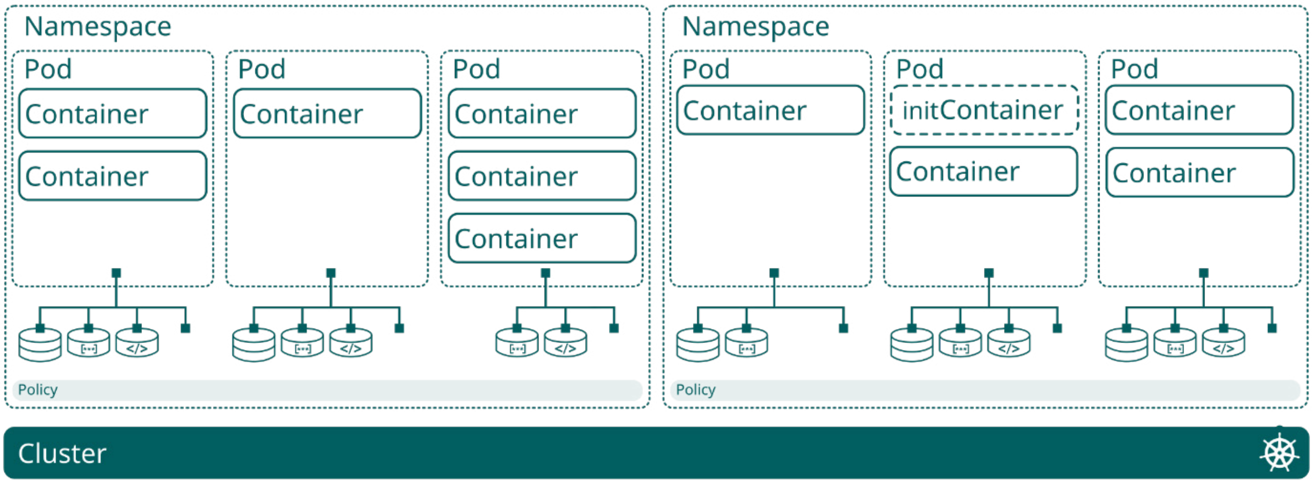
CONTAINER ORCHESTRATION

Container orchestration provides management of containerized workloads for a cluster. It executes secured and automated rollouts and rollbacks, quality of service enforcement based on available hardware, secrets management, modular security, configuration management, self-healing, and horizontal scaling. An orchestrator's actions keep an application, which may consist of multiple containers, in a desired state based on its defined constraints. This design feature of orchestrators leads to better system resiliency against variable and adverse conditions.

While there are many different commercial and open-source container orchestration solutions available today, almost all derive from a single common upstream source – Kubernetes. Kubernetes was originally designed by Google and the name originates from a Greek word meaning "helmsman" or "pilot." If you have ever shopped on eBay, searched via Google, used Twitter, or managed files in Box folders, then you have used Kubernetes.

Each container runs within a logical construct known as a pod. Collectively, the containerized processes within a pod form a microservice and pods are the basic unit of replication within an orchestrated cluster. A cluster will usually contain many pods, each of which has its own allocation of resources. Pods can be further grouped using additional logical constructs such as deployments, which define more high-level capabilities like how microservices interact, dynamically scale, and are resourced. Grouping pods into segregated sets (called "namespaces") enables policies and high-level resource management to be easily and widely applied. The following figure depicts the relationship between the containers, pods, namespaces, and the cluster.

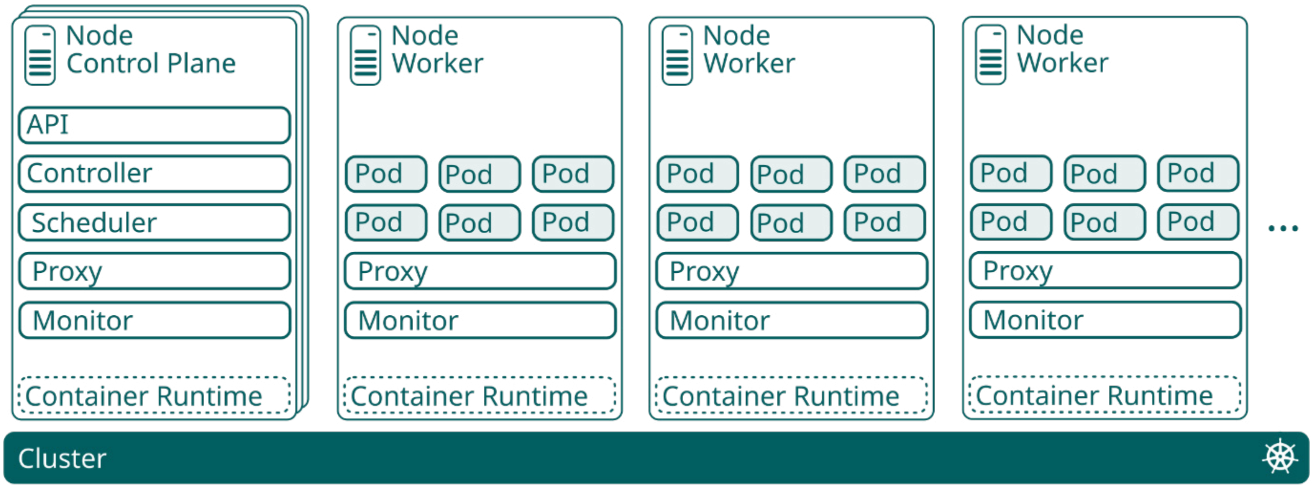
Figure 4: Cluster, Namespaces, Pods and Containers



A Kubernetes cluster consists of two main components:

1. The control plane, where management microservices detect and respond to cluster events.
2. The worker nodes that host the functional containers, pods, deployments, and namespaces described above.

Figure 5: Kubernetes Control Plane and Workers



Both the control plane and the worker nodes operate using the container runtime, with Kubernetes components providing container monitoring and securely proxying communications. Nodes can be either physical or virtual, and residing on one or more physical servers.

Unlike the rigid architectures that were historically employed for grid management and rely on primary and backup servers for critical applications, Kubernetes utilizes an approach based on distributed consensus. Kubernetes can scale functions and applications, both dynamically and massively, while achieving the critical high availability. A minimum of three control plane nodes are utilized to provide fault-tolerance and high availability. The number of worker nodes is proportional to the availability and computational requirements of the deployed applications and may far outnumber control plane nodes. Figure 5 depicts the relationship between the control plane and the worker nodes.

In summary, an orchestrated microservice architecture built on Kubernetes provides complete lifecycle management for containerized workloads, including:

- Automated and secured rollouts and rollbacks
- Modular security and enforced application of security policies
- Quality of service (QOS) of domain functions based on available hardware
- Secret, configuration, and application management
- Self-healing
- Horizontal scaling

EVERYTHING AS CODE (EAC)

EaC is an approach and mindset that advocates representing an application's lifecycle, infrastructure, solution architecture, and configuration as code – human-readable files that can be processed by machines. EaC enables automation for deployments and updates, reproducibility for disaster recovery and simulation, and strong change control for governance. Increasingly, all aspects of solutions are being described by code, from infrastructure to solution and architecture.

Kubernetes is not only the orchestration engine for microservice architectures – it is also one of the key enablers for software defined architecture (SDA). SDA allows the solution architecture to be completely software defined. An example would be human-readable code that fully specifies the desired state of the microservices comprising a solution, in terms of the specific versions of containers to be deployed, their mapping to pods, deployments, and services, and the replicas and scaling of their domain functions.

Consider the DER monitoring example in Section 2. The composition of each of the three types of microservices may vary between those performing the DER data collection, those performing power flow, and those providing alarm management. Moreover, new functions may be added, such as services performing the construction of hosting capacity maps. Defining the entire architecture as code allows the enormous flexibility of the architecture to be well managed with change control, automated updates, and rollbacks.

Through simple, human-readable code, a complete functional architecture can be defined to deliver the services, analytics, controls, and visualizations required for grid operation and orchestration. Moreover, how data flows through the components can also be captured as code (see Section 5.2). As the solution architecture is defined as code itself, it can easily be modified to allow new services and capabilities to be added to the system as grid requirements change. This is one of the key enablers for composable software solutions – modular software that can be combined and "composed" to meet the constantly evolving requirements of the grid (see Section 7.1).

The concepts of infrastructure as code (IaC) and SDA are being applied throughout modern software solutions. For example, configuration as code enables configurations to reside and be managed in the same, strong change control systems, rolling up and back as required. The principle of everything as code (EaC) enables all aspects of a defined deployment to be managed in the same way as code, with strong change control and simple update management.

DevOps AND DEPLOYMENT AUTOMATION

DevOps is defined as a set of practices that combine software development (Dev) and IT operations (Ops) to shorten the development lifecycle for software and provide automation that implements best practices to improve and ensure software quality. DevOps enables continuous integration (CI) of software – where developers frequently merge code changes into a central repository triggering pipelines of automatic builds, code analysis, testing, and packaging/containerization.

Continuous delivery (CD) extends CI by automatically deploying the incremental changes into production environments after end-to-end testing and validation, thus enabling faster and more frequent releases of new features. DevOps is complementary with Agile software development methodology and relies heavily on the use of EaC and automation to achieve those outcomes.

In theory, with CD, software can be deployed and updated daily, weekly, fortnightly, or however often suits the business requirements. However, to truly reap the full benefits of CD, software must be deployed to production as early as possible. This both ensures the latest security updates for a production system, and also releases features and changes in small batches that are easy to troubleshoot and adopt.

CI/CD is a powerful capability that enables frequent delivery of features, security updates, and apps to customers through automation.

GitOps AND CHANGE CONTROL

Regularly applying software updates to multiple production and non-production environments is a complex and time-consuming task. As a result, updates are often deferred, delaying their benefits for operations and slowing their overall time to value. Given the amount of software involved and the subtle differences in configuration between such environments, it is also often challenging to accurately describe the state of each environment and to validate that they are properly provisioned.

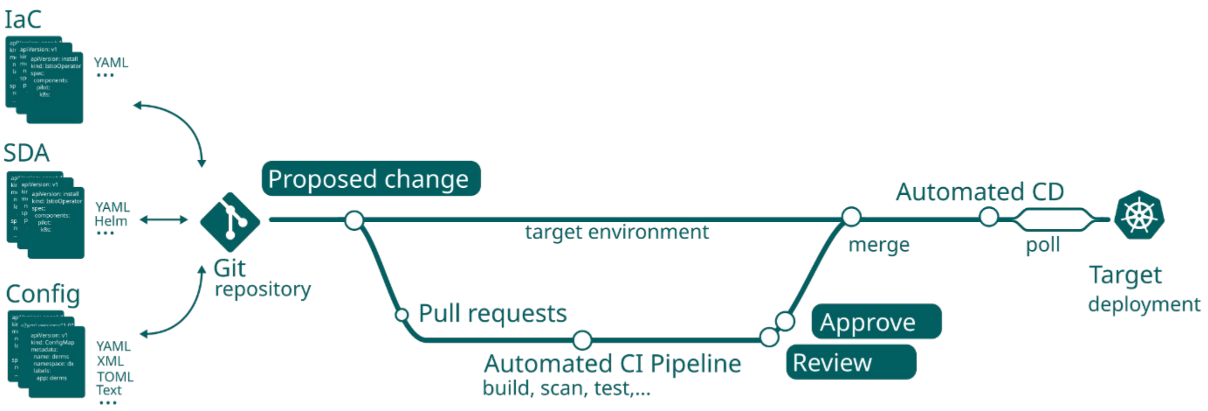
Git is a version control system that tracks changes in any set of electronic files. Git is not only the predominant version control system used in modern software development, but it equally can serve as the single source of truth for the policy that describes everything as code. As described above, “everything” in this context is the complete desired state of the target environment, including the underlying infrastructure, solution architecture, and application configuration.

GitOps is a policy-based deployment automation that is an alternative to manual methods for building and maintaining complex systems. GitOps provides tooling and a framework to take DevOps’ best practices and apply them to infrastructure automation and application deployment. GitOps also eliminates the need for administrator access to operational environments for deployment and updates. Instead, it enables workflows built around strong version control to manage change requests and their approvals by a limited group of individuals with appropriate permissions.

While there are many ways to define EaC and policies, an increasingly popular approach is using a human-readable data serialization language called YAML. In Kubernetes, YAML is used to define the infrastructure and deployment architecture for a solution. Related YAML files may be generated from templates using package managers such as Helm to make their construction and life-cycle management easier. Everything is defined using code such as YAML, and all code is stored in the Git version control system. Collectively, these templates and their generated code provide a description of the needed policy that sufficiently defines an entire deployed solution, from infrastructure to architecture, to configuration. The code completely defines the desired state of the target software and can be used to automate both a new installation and the continued evolution of the solution through updates.

Rather than manually applying the code stored in Git to deploy or update running solutions, GitOps automates the updates. With this approach, approved changes in the policies held within Git automatically update the running system. For example, by changing the YAML, one can alter the desired state of the target environment (for example, adding a new service or changing the version of a particular service). The continuous deployment tooling monitors Git for any changes and when detected, applies them to Kubernetes for implementation. As a result, the state of the target environment is updated to reflect the new desired state, as expressed in the revised policy. Should someone attempt to circumvent the above process and directly make changes to a target environment, the GitOps machinery would detect the inconsistency between (1) the desired state described by the policy held in Git and (2) the actual state. Then GitOps would automatically revert the changes.

Figure 5: Kubernetes Control Plane and Workers



Key outcomes delivered by describing EaC and leveraging GitOps include:

- Ability to verify the provenance and integrity of the entire deployed solution (from software artifacts to configuration and architecture)
- Ability to confirm the actual states of target environments match their corresponding desired states
- Improved consistency and repeatability of deployments across different environments
- Strong change control and full-versioned lifecycle management of infrastructure, solution architecture, and configuration

TEST AUTOMATION

The ability to validate changes using manual testing alone is limited, both in terms of the time available for such testing as well as the domain knowledge of the individual performing the tests. Accordingly, the achievable level of testing is often limited, increasing the likelihood of errors being propagated to the production environment. Like deployment automation, test automation is considered essential for consuming smaller, more frequent, and incremental software updates and more quickly realizing operational benefits from new features.

Test automation executes a defined sequence of tests on a specified target environment. The aim is to assess correct configuration and proper behavior of the installed software. There are different types of test automation, applied at different stages in the CI/CD pipelines, including unit testing for validating key functions and preventing regressions, and system and solution testing that assesses the combinations of functions participating in the whole solution. In addition, security, API, and UI testing should all be included in overall test automation.

Key outcomes delivered by test automation include:

- Improved software and solution quality
- Reduction in regressions
- Reduction in the level of labor-intensive manual testing
- Improved consistency in terms of test results due to elimination of tester-induced variation
- More frequent and thorough testing, given time constraints surrounding urgent changes

CLOUD

Strictly speaking, the term 'cloud' simply implies a particular computing style based on the use of scalable and elastic capabilities to deliver defined services using internet technologies. However, most people automatically tend to associate it with services that use the public internet. In reality, there are three different types of clouds (public, private, and hybrid), each of which abstract, pool, and share scalable computing resources across a network.

Every cloud type is supported by a unique mix of technologies, which almost always includes a management platform and application programming interfaces (APIs) to manage and control infrastructure resources. All typical security best practices for non-cloud environments also apply to any cloud environment, be it public, private, or hybrid. A cloud environment may even allow further enhancement of the existing on-premises security controls. The type of cloud does dictate the level of direct control the tenant has in these areas.

Containers, microservices, and automation software can also be added to every kind of cloud for additional capabilities or increased efficiencies. A cloud platform enhances the effectiveness of various project activities such as development, software deployment, and testing. In a private or hybrid cloud environment, the tenant may assume most of that responsibility, as opposed to the cloud service provider (CSP).

> The five general categories of service offerings that further define responsibilities between the CSP and the tenant are described in **Table 2.**

Service type	What the CSP provides	What the tenant provides
IaaS Infrastructure as a Service	Configuration, deployment, and management of base infrastructure only (data center, networking, firewalls / security, servers, and storage)	Configuration, deployment, and management of operating systems, development tools, database management, and applications
PaaS Platform as a Service	Configuration, deployment, and management of base infrastructure (data center, networking, firewalls / security, servers, and storage), operating systems, development tools and databases	Configuration, deployment, and management of the applications
SaaS Software as a Service	Configuration, deployment, and management of base infrastructure (data center, networking, firewalls / security, servers, and storage), operating systems, development tools, databases, and applications	Nothing – you simply use the provided applications
CaaS Container as a Service	Container-based virtualization in which container engines, orchestration, and the underlying compute resources are delivered to users as a service from a cloud provider	Configuration, deployment, and management of containerized apps… upload, organize, start, stop, scale, and otherwise manage containers, applications, and clusters.
FaaS Function as a Service	A platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the event-based infrastructure. A "serverless" architecture in the sense that services can be "on demand" and instantly scalable	Function code that focuses primarily on business logic. Examples of such functions include but are not limited to Extract, Transform and Load (ETL), mobile apps, web apps, etc. Unlike other service models such as PaaS, users pay only for the resources consumed by their functions and do not incur idle time charges

> **In the following sections we explore each of these cloud types further to understand how they differ and the unique outcomes they can potentially deliver.**

PUBLIC CLOUD

Public clouds are the most common type of cloud computing deployment. The cloud resources (like servers and storage) are owned and operated by a third-party cloud service provider and delivered over the internet. With a public cloud, all hardware, software, and other supporting infrastructure are owned and managed by the cloud provider. As of the date of this writing, the three main providers of public cloud services are Microsoft Azure, Amazon Web Services, and Google Cloud Platform.

In a public cloud, one usually shares the same hardware, storage, and network devices with other organizations or cloud "tenants." One also accesses services and manages an account using a web browser or publicly available API. Public cloud deployments are frequently used to provide computing and storage services for testing and development environments.

A key aspect of public cloud is the shared responsibility model that defines what the CSP is responsible for and what the owner of the workload is responsible for. In general, the CSP is responsible for the platform's security and maintenance (the underlying virtualization platform and control infrastructure), while the workload owner is responsible for securing the operating systems and applications running on top of the CSP platform.

Multi-cloud solutions are software solutions that are portable across multiple cloud providers. They are usually built using cloud-native technologies, such as Kubernetes, that are supported by all cloud providers. Multi-cloud solutions allow the most appropriate CSP to be used (for example, based on cost or availability of local/regional data centers). In addition, they generally support multi-cloud deployments, where cloud services from more than one CSP are used at the same time. This can offer the advantages of cloud cost optimization and improved resilience to cloud outages and downtime.

Public clouds increasingly provide dedicated services and hardware tuned for AI/ML (Section 6). For example, dedicated graphics processing unit (GPU) and tensor processing unit (TPU) resources together with MLOps pipeline services can provide scalable resources for AI/ML model training that can accelerate AI/ML development and reduce investment costs.

Advantages of public clouds:
Lower upfront costs – You pay for what you consume, when you use it. In addition, most services are available immediately, compared to waiting three to six months for physical hardware
No hardware maintenance – your service provider is responsible for maintaining the hardware
Near-unlimited scalability – on-demand resources are available to meet your business needs, including dedicated hardware for specific computational loads like AI/ML
High reliability – in addition to the significant level of hardware redundancy within a given data center, applications may be designed to take advantage of equipment from multiple data centers

Disadvantages of public clouds:
Less control – the CSP will maintain the platform and will not be scheduling their maintenance around your business requirements
Potential for reduced availability – while public clouds are highly available, their tenants are, by definition, remotely located. The interconnection to such a cloud is a potential source of failure that may involve multiple hops and communication providers. It should be noted that this is not unlike some traditional control center architectures today, wherein the operators are remotely located from the servers and, as such, exhibit a similar potential for loss of interconnecting communications
Cost – only certain workloads and use bases benefit from cloud efficiencies, and cost needs to be carefully managed

PRIVATE CLOUD

A private cloud consists of cloud computing resources used exclusively by one business or organization. The private cloud can be physically located at your organization's onsite datacenter, or it can be hosted by a third-party service provider. But in a private cloud, the services and infrastructure are always maintained on a private network and the hardware and software are dedicated solely to your organization. There are numerous providers of private cloud technology and services including but not limited to Hewlett Packard, VMWare, Dell, Red Hat, Microsoft, and Amazon.

In this way, a private cloud can make it easier for an organization to customize its resources to meet specific IT requirements. Private clouds are often used by government agencies, financial institutions, and other mid- to large-size organizations with business-critical operations seeking enhanced control over their environments.

Advantages of private clouds:
More flexibility – your organization can customize its cloud environment to meet specific business needs
More control – resources are not shared with others*, so higher levels of control and privacy are possible
More availability – private clouds, when co-located with the tenant's end users, eliminate a potential weakness of public cloud (and some existing control center architectures) as the interconnection availability is no longer a concern

Disadvantages of private clouds:
Higher upfront cost – as private cloud infrastructure is dedicated to a single tenant, that tenant bears the full cost of its operation and time-of-use billing is generally unavailable

*Among the potential concerns with hosted private cloud are that the hosting service's administrative personnel may have or be able to gain access to the customer's confidential information or that the cloud is hosted in a foreign country increasing the risk of data exfiltration and/or security breaches. Both concerns can be effectively mitigated by proper due diligence during CSP selection.

HYBRID CLOUD

A hybrid cloud is a type of cloud computing environment that combines on-premises infrastructure (or a private cloud) with a public cloud. Hybrid clouds allow data and apps to move between the two environments.

Many organizations choose a hybrid cloud approach due to business imperatives such as meeting regulatory and data sovereignty requirements, taking full advantage of on-premises technology investment, or addressing low latency issues. Both Microsoft's Azure Stack and Amazon's Outposts are examples of IaaS offerings for hybrid distributed cloud that provide options for tethered and untethered operation of services from the public cloud at private locations.

Advantages of hybrid clouds:
Control – your organization can maintain a private infrastructure for sensitive assets or workloads that require low latency while leveraging public cloud where appropriate
Flexibility – you can take advantage of additional resources in the public cloud as appropriate when you need them. Moving to the cloud does not have to be overwhelming because you can migrate gradually, phasing in workloads over time
Balance – by offering the strengths of both on-premises/private cloud and public cloud, hybrid cloud mitigates concerns about availability, security, and cost associated with public or private-only implementations

Disadvantages of hybrid clouds:
Complexity – because of its hybrid nature, hybrid cloud is typically more challenging to implement and manage

DATA OPERATIONS AND FABRIC

Data and its derived value will play a critical role in supporting the radical changes occurring in electricity grids. Data is now the fourth strategic business asset, alongside time, talent, and money. The amount of data being collected in grids is exponentially increasing, and the diversity of data available to drive analytics, automation, and intelligent decision support has vastly expanded. DataOps built around a modern data fabric provides the suite of technologies, processes and practices that enables and accelerates the transformation of data into real value.

DataOps is a confluence of DevOps and agile engineering practices that, together with enabling technologies, provides the processes to develop and continually improve effective and predictable data analytics. DataOps encompasses the entire life cycle of data, from collection and governance to data analysis and quality improvement. The result of effective DataOps is to shorten the cycle time of analytics development and significantly improve time-to-value of data-driven outcomes.

One of the key enabling technologies of DataOps is a data fabric. A data fabric provides an integration layer, or fabric, that connects data to processes. It simplifies data discovery, access, data enrichment, and exploration, and it forms the foundation for knowledge discovery, data analysis, and intelligent data-driven decision support.

DATAOps

Digital transformation requires the ability to connect to data across many different systems via many different technologies. Traditional software applications, regulatory and security segregations, and the business organization itself, create data silos, which are among the biggest impediments to the digital transformation. DataOps provides the processes, practices, and machinery to securely free data from these traditionally isolated data repositories. It is at the intersection of Agile,

DevOps, and Lean practices, using automation to accelerate time-to-value, time-to-insight, promote collaboration, enable and expediate new data applications and analytics, and improve the reliability of operational data.

Modern DataOps best practices include:

- **Building on open standards and technologies**

  Allowing data to be accessed using open API technologies and standard data formats.

- **Decentralized ownership of application data**

  Keeping data in its original, distributed stores and adopting centralized metadata and governance (see below) delivers the advantages of decentralized data such as using technologies and teams best placed to curate the data, while retaining the advantages of centralization. Leveraging decentralized data also avoids complex and fragile transformations commonly required in centralized data repositories and minimizes data duplication.

- **Centralized cataloging of metadata**

  The benefits of centralization, such as easier data discovery and change management, can still be leveraged by centralizing the metadata – the data about the data, such as schemas, quality metrics, etc. Automation is applied to the creation and maintenance of metadata to ensure it is up to date and to trigger workflows and processes on data change.

- **Implementing metadata-driven collaborative governance**

  Centralizing the metadata enables a single, coherent data governance, ranging from ownership and access controls to tracking data lineage (providing a clear understanding of where derived data originates and the impacts of data changes).

- **Enriching the metadata and building knowledge graphs**

  Automation can enrich metadata with additional insights such as data quality, availability, and usage information. Building a knowledge graph allows semantics to be added to data, clarifying data types, standardizing definitions, and further enhancing data discovery.

- **Virtualizing data to simplify data queries across sources**

  Data virtualization techniques allow data from distributed and disparate sources to be accessed and combined as if it were a single, coherent, central data source. This simplifies data queries and access.

- **Provide self-service data preparation tools for curating high value datasets**

  Technologies like data virtualization facilitate easy self-service capabilities for preparing data for use in high-value analytics. This further simplifies the analytic development and dramatically improves the time-to-value.

- **Providing self-service tools for turning datasets into actionable insights**

  Data discovery, together with data governance and data virtualization, empowers individuals to leverage the data to improve decision making and drive insights. Creating a safe and up-to-date citizen-oriented data environment ensures that data constantly brings value to the whole organization.

- **Using statistical process control to enable a self-healing architecture**

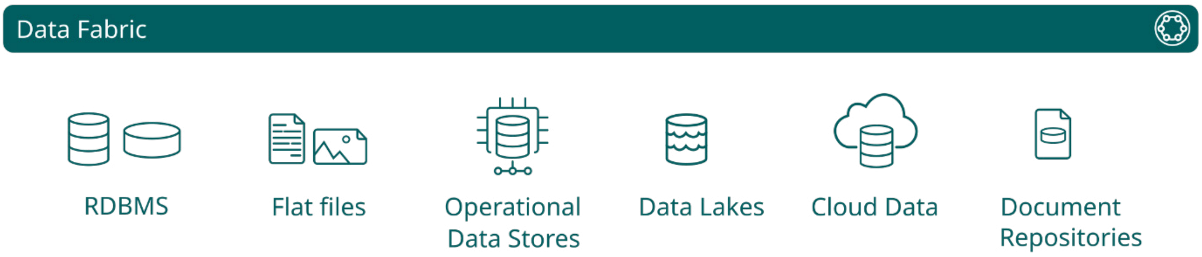
  A centralized, enriched metadata catalog allows data transformations and pipelines to leverage the metadata to adapt to disruptions and operational changes. For example, data transformation patterns can be automatically optimized based on data usage.

These principles and practices enable an organization to easily integrate new data sources and quickly gain value from the data through the fast development of new analytics and the self-service insights it can provide. DataOps promotes the move to a data-centric design, putting data and its life cycle management at the center of a system. Analytics, analysis, reports, self-service decision support, and automation all feed from data.

DATA FABRIC

The data fabric provides the suite of tools and technologies that enables effective DataOps. It provides a “fabric” that connects data to processes, mapping the data residing in disparate locations and making it available for exploration and analysis.

Figure 7: Schematic illustration of a Data Fabric and diverse data sources



The key technology components of a data fabric include:

- **Operational Data Store**

  Providing a highly performant and centralized data store for snapshots of the latest, real-time, operational data allows applications to access and combine key data easily and efficiently. It is crucial for supporting business intelligence and analytics driven from the latest operational information. Moreover, the operational data store (ODS) supports operational user interfaces (UIs) and dashboards that combine different data types and sources.

- **Augmented metadata catalog**

  Provides the technology to collect and curate the metadata associated with disparate data stores. It enables the data discovery and data governance capabilities of DataOps. Moreover, advanced catalogs allow automatic enhancement of the metadata; for example, the active analysis of data usage patterns. The metadata catalog is a key enabler of DataOps.

- **Enriched knowledge graph**

  Further augmenting the metadata with data lineage, recency, tags, and glossaries, an enriched knowledge graph improves data discovery and curation. Moreover, it enables semantic information to be added and, in advanced systems, automatically inferred.

- **Data preparation and self-service user interfaces**

  A user experience to easily enable data discovery and exploration. The self-service user interface should allow data to be quickly identified and viewed to enable a user to easily build displays that support data-driven decisions. In addition, the ability to build virtual and curated datasets and pipelines is key in preparing data for use in analytics. Data preparation can also be integrated into the processes and technologies enabling the development of AL/ML based applications (see Section 6).

- **Data delivery and exchange machinery**

  Retrieving and exporting data from different systems can involve a myriad of possible protocols and technologies. Being able to securely retrieve, interface, and deliver data across a wide range of technologies is a key capability of an effective data fabric. In grid environments, supporting very specific protocols and integration mechanisms is highly beneficial (see Section 5.4).

- **Data orchestration systems**

  Robust data delivery mechanisms can be extended to full data orchestration. Configurable workflows and data integration patterns provide the flexibility to adapt how data is processed and exchanged inside and outside the solution. This includes allowing data validations to be easily added to pipelines, or the automatic creation, deployment, and maintenance of resilient integrations based on metadata. Adoption of EaC allows the data flows to be captured, modularized, and managed as part of the entire solution (Section 3).

This is not an exhaustive list, nor are the functional boundaries of a data fabric universally defined. For example, data fabrics may also include operational historians, where historical data that drives operational analytics is maintained. At its core, the technologies and tools that comprise a data fabric combine to support full DataOps capabilities.

Even if an organization does not fully leverage DataOps business processes, the data fabric tools form the basis for building composable software solutions. A single, flexible method of accessing data and orchestrating data flows and interactions of different modular applications is one of the building blocks of composability (see Section 7.1 for more details).

DATA-CENTRIC DESIGN

Most software development is currently application-centric, with each application constructing its own data models highly tailored to its function and persisting data in tightly specific schemas. As the number of applications increases and the amount and types of data exponentially grows, the data becomes sprawling and complex. The same data can often become duplicated and stored in application-specific silos in multiple formats at multiple times. Should that happen, it becomes difficult to identify the ideal data source when constructing new data sets or analytics. As the need for a larger and more diverse suite of applications to support the increasingly complex grid operation increases, this application-centric approach becomes unsustainable.

Although a data fabric provides the tools that can manage this complexity, it also offers an alternate approach: data-centric design. In data-centric design, application developers leverage the metadata and the semantics within the enriched knowledge graph to dynamically identify and select the data that best supports their function. This places the data as the central asset, rather than the applications.

Allowing applications to easily use the data that is available via the fabric – versus copying the data into an application specific silo – is the tenet of data-centric design. Through data fabric technologies, it is now possible to have a single instantiation of core data types that support a wider range of applications. This simplifies application development and reduces data duplication. It can even improve availability for applications that support multiple alternate data sources in their functions depending on metadata (e.g. quality and availability). For example, applications requiring raw kilovolt telemetry at specific locations may dynamically select SCADA or synchrophasor measurements depending on data availability.

A corollary to the data-centric design is that applications are increasingly ephemeral. The data is the important and enduring asset. In turn, applications start evolving faster, and deliver value to support the changing grid.

Adopting data-centric design is not mandated by a data fabric; rather, the latter is one of the many advantages of adopting the former. Furthermore, the modern development of AI/ML based applications increasingly follows data-centric patterns (see Section 6).

A GRID DATA FABRIC

The data fabric, through delivering DataOps capabilities and data centric application design, provides a key technology to support the effective orchestration required for rapidly evolving electric grids. It enables significantly improved lifecycle management of data and time-to-value for analytics and automation, further increasing its value over time. The fabric simplifies incorporating new data types and data integrations, empowering uses with dynamic and self-service data insights improving data-driven decisions and delivering actionable intelligence.

Although this technology will have a fundamentally positive impact on modern grids and their operation, as mentioned in Section 1, electricity grids have a long history with entrenched technologies. In practice, the data generated and used within grids can have very specific characteristics; for example, extensive quality metadata that can determine when and how data should be used. Data can be grouped and mapped to geographical areas and electrical bands, corresponding to areas of responsibility (AOR) which can change how it is accessed and optimized. Moreover, data from electrical grids has specific and enduring standards like the

Common Information Model (CIM) that is used to identify and model key domain data. The specific use and heritage of data in grids can make standard data fabric tools and machinery less effective without appropriate and targeted adaptation.

The protocols and data formats built and used within the industry over decades creates a very specific and unique data environment for a data fabric's delivery and integration mechanisms.
A data fabric must have grid-specific integration capabilities and awareness in order to provide immediate value and accelerate the benefits it affords. It must also be both tuned to and optimized for the grid (for more discussion, see Section 7.1).

AI/ML FROM INSIGHTS TO AUTOMATION

AI/ML are at the early stages of fundamentally transforming all industries.

Although ML models demonstrating increasingly complex and human-like capabilities dominate the media headlines – even in conservative industries like the power sector – widely validated and well-established approaches are being applied. In the energy industry, both the targeted application of validated models and the leveraging of more cutting-edge ML in non-mission critical areas are being increasingly adopted. An example would be using advanced forecasting to factor weather information and other types of data into generation and load forecasts. Other examples of applying ML in less-mission critical areas include machine vision and object detection for directing and optimizing vegetation management around transmission and distribution assets.

At present, the targeted application of ML is primarily around producing intelligent data insights – in other words, providing the operators, planners, and engineers with information that leads to more optimal activity and outcomes. As more and more data becomes available for decision making, the use of AI/ML-based analytics to transform the data into digestible and actionable intelligence is increasingly required. A human operator can no longer consider all the different raw data sources that may be pertinent to a decision.

The use of advanced algorithms to process raw data into useful information is not new in the grid industry. Physics-based approaches, such as power system state estimation using SCADA telemetry, or extraction of electro-mechanical and other oscillations from synchrophasor measurements, are normal practice. These approaches are highly beneficial, with their results easy to interpret and explain from the underlying physics.

However, physics approaches do not readily allow new data types to be combined and leveraged to improve or enhance the results. They are slow to change and require highly experienced power engineers to improve or modify the algorithms. Current AI/ML approaches can operate alongside physics-based algorithms, often leveraging their results and, as a consequence, beneficially incorporating the physics into their learning. Running physics-based systems in parallel with AI/ML systems will endure for some time, allowing the core physics approaches to provide operator confidence and yield physics-based training data for further enhancing the ML models (see Section 6.5 for how this may evolve in the future).

We are at the beginning of an explosion of AI/ML based insights, building on and incorporating physics-based algorithms. As this explosion happens, it is imperative that these insights are both easy to validate and, importantly, easy to integrate into grid workflows and solutions. Sections 6.1 and 6.2 detail the technologies that unlock the development and validation of AI/ML insights and the efficient and timely integration into grid solutions.

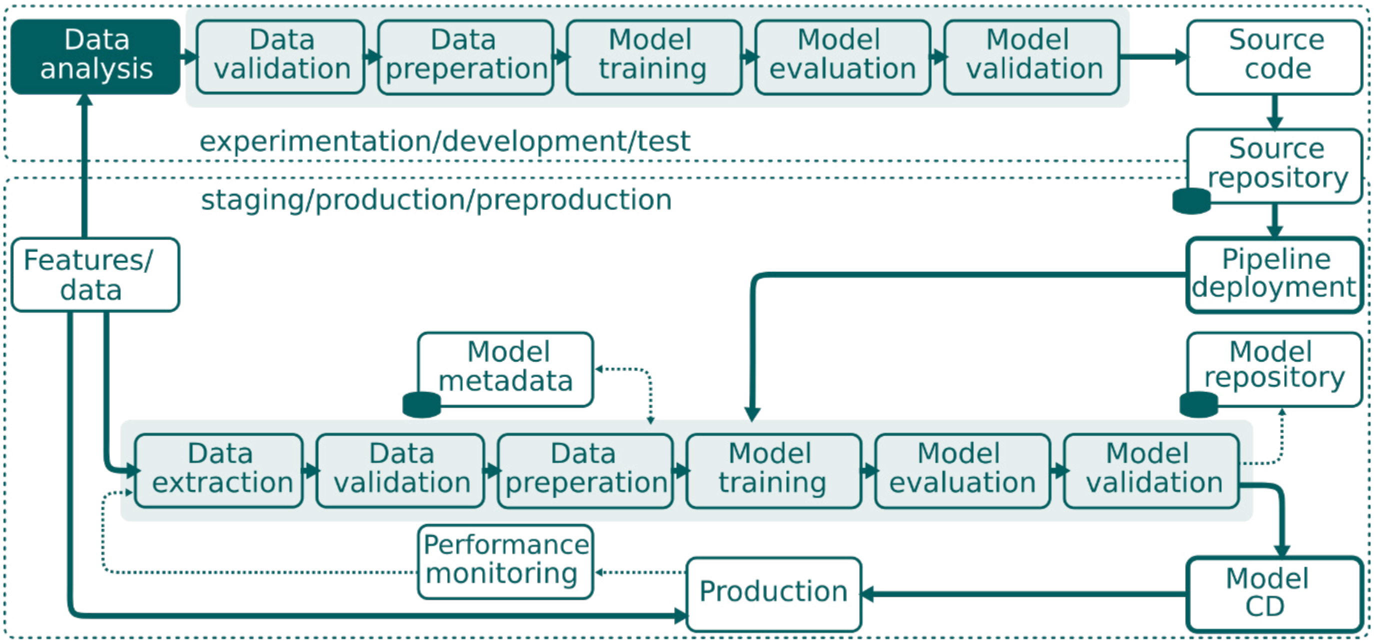
Although the use of AI/ML in automation of grid operation is currently limited, the rapid adoption of intelligent data insights, developing key ML models designed for grid data, and the application of AI/ML to enhance current automation systems will be the beginning of its rapid adoption. Section 6.4 describes the technologies and processes that will enable and drive AI/ML based automation.

The technologies underlying the electricity grid span from the late Industrial Age through the Information Age, and it is now entering and adapting to the Age of Intelligence.
This is incredibly timely, as both the energy transformation and the resulting complexity of supporting it with grid orchestration will require the tools and transformations of the Age of Intelligence.

AI/ML LIFECYCLE

Creating an efficient lifecycle for AI/ML-based models from data cleansing and preparation through to deployment into production and governance is called MLOps. It expands on DevOps and DataOps, bringing together tools and processes into standardized pipelines for aiding and automating AI/ML model development through to the operational use of the model. As data changes over time the operation of an AI/ML model should be constantly monitored, validated, and frequently updated with a re-trained and refined model. This cycle also allows new data sources to be added as they become available, enabling AI/ML models to be easily updated to include new feature sources. Like DevOps and DataOps, MLOps leverages automation to optimize model development and standardize model operation.

Figure 8: Typical MLOps pipelines



Development of AI/ML models, and MLOps dovetails into DataOps and data-centric design. It's all about the data. At the conception of an AI/ML-based product, data exploration and analysis are key enablers for starting the data science development process. Tools that provide access to all available metadata enable self-service data discovery, exploration, and transformation to create feature processing and pipelines. DataOps and a data fabric are critical for ensuring MLOps efficiency.

Figure 8 illustrates typical MLOps pipelines and high-level stages including:

- **Data science, development, and experimentation**

  This includes data extraction, validation, preparation, labeling, model development and training, model evaluation, and model testing. It's important to note that in mature MLOps, model development (the original data science activity) is only one high-level stage in the lifecycle. This stage is generally performed by data scientists and ML researchers. Key tools include the data fabric, which unlocks self-service data discovery and exploration and the metadata catalog. In addition, interactive computing dashboards, ML toolkits, feature stores, and model development automation (hyperparameter tuning, etc.) will generally be leveraged during model development.

- **Continuous Integration**

  Like DevOps, MLOps encompass continuous integration – essentially, the build, validation, and testing of the ML model should follow a continuous integration pipeline. Moreover, MLOps continuous integration should include data testing and validating. MLOps CI runs in parallel to DevOps software CI, with the software CI providing the software artifacts, like microservices that execute the AI/ML models. Key tools include a model registry, integration with the metadata catalog, and model pipeline orchestration.

- **Continuous Training**

  Continuous training involves automatically re-training ML models with the latest available data, adding the validated model to the model registry. The automatic training may be engaged based on new data availability, time or externally triggered (see Continuous Monitoring below).

  As with CI, key tools include a model registry, integration with the metadata catalog, and model pipeline orchestration.

- **Continuous Delivery**

  Automation of the productization of new and updated ML models, including delivery, should also follow standard CD practices. MLOps model continuous delivery runs in parallel with DevOps software CD. The software CD pipeline generates the deployable artifacts, like microservices, that execute the model. The model CD pipeline allows for new model parameters held in the model registry to be deployed, thus maintaining and improving the production model.

  Key tools include model pipeline orchestration and model service API. The model service provides an API for easily accessing models in the model registry.

- **Continuous Monitoring**

  The performance of AI/ML models operating in production should be constantly monitored. This includes metrics that indicate the source of production data in addition to the performance of the model itself. A drop in model performance (e.g., predictive performance) can trigger new model training and delivery as long as the source data quality is maintained.

  Key tools include metadata catalog and model pipeline orchestration. The machinery for assessing model performance can depend on the type of AI/ML model and the resulting performance metrics should be made available, like other standard metrics, in the deployed software artifact.

One of the key requirements for successful MLOps is extensive, automated testing and validation integrated to all stages of MLOps. AI/ML models add a new level of testing that significantly expands traditional software code testing (Section 3.3). AI/ML models degrade rather than report errors; for example, when source data or environments are not as expected. Testing and calculating metrics on the data, schemas, and environmental differences between training and production can be just as useful as measuring the predictive power of the AI/ML model itself in assessing model performance and appropriate actions.

In general, two of the most challenging parts of delivering effective and beneficial AI/ML in production systems are:

- Inconsistent delivery pipelines of AI artifacts
- Integration with existing software and solutions

Leveraging DevOps, DataOps and MLOps best practices can significantly address the first of the above. The following section will look at the technologies and methods that can be used to resolve the second.

SIMPLIFYING AI/ML INTEGRATION

As mentioned above, running AI/ML-based models will initially augment the physics-based solutions already in production in grid software systems. It is immensely challenging to integrate AI/ML approaches into the complex myriad of traditional algorithms and sub-systems that comprise enterprise grid management, planning, and markets software.

At a macro level, integrating entirely new analytics based on AI/ML is no different than integrating those based on physics or engineering approaches. Building on technologies that enable composable solutions (see Section 7.1) is a key requirement. However, AI/ML based applications need stricter monitoring and validation, including monitoring of the input data itself, and require more frequent AI/ML model updates (following MLOps practices, described above). Therefore, a sufficient complement of MLOps machinery, such as access to the model registry, must be available in production environments. Depending on the zones in which recent training data can be accessed, continuous training may also need to be implemented in production security zones.

Integrating MLOps into the data fabric is one approach to simplify this macro level integration.

Availability of AI/ML models recorded in the metadata catalog, using the active metadata to monitor operational data driving the ML inference, and integrating model and data orchestration mechanisms to deploy new models can all simplify application of MLOps. Moreover, adding the latest results into an operational data store simplifies integration into UIs and dashboards, including the self-service decision support dashboards that comprise part of the data fabric itself (see Section 5.2).

Integration at a lower level can be more complex. For example, the preparation and transformation of raw data into data frames optimized for input into AI/ML models can be computationally expensive, requiring dedicated transformation pipelines. This part of MLOps machinery can also be integrated into the data fabric. For example, key data fabric components, like the ODS, can be built around technologies that allow direct integration of operational data into AI/ML input data-frames, with minimal-at-most transformation. Increasingly, the DataOps and MLOps will blend to provide a single set of tools and processes for automating the lifecycle of data and the analytics that use it (see Section 6.3).

Another integration complexity of AI/ML analytics that is generally more convoluted than traditional physics-based analytics is the management and application of data governance. This is particularly the case as the analytical output becomes used as an input to other analytics (e.g. physics-based analytics as input to AI/ML analytics described above). Data governance capabilities will increasingly require data lineage management; specifically, tracking and visualizing the flow of data through different analytics and usage pipelines. Moreover, centralized management and application of access controls based on permissions and AORs must be strictly adhered to. The permissions and AORs of derived data also need to be explicitly managed at all stages of the data lineage.

Like other AI/ML integration complexities described above, DataOps processes can also be leveraged in data governance. A data fabric that provides a metadata catalog which allows data lineage to be monitored and managed can help simplify and standardize the governance and integration of AI/ML analytics. Moreover, a data fabric that centralizes the application and administration of permissions and AORs provides a single point of application across all analytics.

XOps

The merging of DevOps, DataOps and MLOps (and the further automation and processes of ModelOps, PlatformOps, and AIOps, not discussed here) is leading to the concept of "XOps." XOps is an umbrella term encompassing all the processes and automation machinery that provide the agile solutions that allow an organization to quickly adapt to an evolving data landscape. Together with EaC (Section 3), XOps provides the processes and mechanisms to shorten the cycles of developing, deploying, and operationalizing all types of advanced analytics.

THE PATH TO INTELLIGENT AUTOMATION

The use of AI/ML analytics in the automation of grid operations is currently very limited. However, automation of grid operations is increasing, albeit based on traditional approaches. Examples include automated frequency restoration reserve (aFRR), driven by proportional integral controllers; or automated power restoration tools based on power flow calculations together with rules and heuristics.

Drivers of the expanding use of automation in grid operations include:

- The growing complexity of grids and their composition. These call for more automation and moves operator decisions to higher levels of abstraction. This drives automation at all levels of a grid, from within substations, to regional/zonal, to grid-wide.
- Improving grid security and efficiency, particularly while the grid complexity grows, will also drive automation. An example would be enabling fast frequency response systems in low inertia environments.
- Regulation is also acting as a driver for automation, from the automation of report generation to inter-grid data exchanges.
- Workforce and skill availability constraints are another driver for automation, such as progressively automating common tasks to allow workers to focus on more complex activities and decisions.

The energy transition – and its effects on all aspects of grid operation, energy markets and grid planning – will ultimately drive a massive expansion of automation over the next decade. Traditional approaches, from engineered and rule-based solutions to building on physics-based analytics, are slow to develop and sensitive to data changes. In addition, the complexity of such traditional approaches dramatically increases as more data types are added. It's not unheard of for the number of decision points to potentially explode.

AI/ML models and MLOps will provide an increasing means of building automation into grid systems. As the use of AI/ML increases in data intelligence and decision support, there will be training datasets and even models with proven value that can be adapted and expanded to drive automation. In addition, the current engineered automation systems, which are often operating with highly constrained inputs, can provide training data to build AI/ML automation with more diverse inputs. This empowers the resulting AI/ML models to quickly incorporate new data sources that improve automation efficiency and outcomes.

Improvements in grid simulation – in particular, simulation of grid dynamics -- will also drive the development of AI/ML based automation. Training and exercising in sufficiently accurate and simulated environments will build the core models and training datasets that can be further refined for real environments.

The same approach of MLOps, which will drive the operationalization of AI/ML models in data intelligence and decision support, will also drive their use in automation. MLOps tooling and processes will improve and drive innovation momentum across the board, enabling and accelerating AI/ML model adoptions across all areas of grid operation.

PREPARING FOR THE FUTURE

Running physics-based and AI/ML systems in parallel is currently advantageous, both in terms of operator confidence and ease of explanation enabled by the physics algorithms and the physics constrained data the process can provide for ML training. Eventually, AI/ML models may become sufficiently accurate to completely replace physics-based analytics. This has already happened in other domains; for example, where ML models of protein-folding out-perform the physics-based solutions due to the complexity and combinatorial explosion of sub-atomic interactions. For example, if a deep neural network model provides more robust error correction and data estimation over traditional state estimation, it will eventually become more valuable. Moreover, if this deep model does not require an accurate physics (digital twin) network model of the grid to be maintained, it becomes even more advantageous.

Although we are currently a long way from this scenario, it is the MLOps and DataOps technologies and practices that will provide the path toward this.

A PATH TO THE FUTURE

All of the technologies presented in the sections above will play prominent roles in the grid software solutions of the future. The agility, extensibility, scalability, and improved time-to-value they collectively offer will fundamentally change how future grids are operated.

Software solutions that meet the ever-changing requirements of modern grids must be composable – in other words, allowing new functions to be quickly incorporated and old functions to be easily retired. For example, to maintain grid balance, utilities will need to implement software solutions to manage active demand response programs, smart grid technologies, and battery energy storage solutions (BESS) while replacing traditional load forecasting models and automatic generation control (AGC) with modern algorithms that incorporate market signals and AI/ML models. Section 7.1 describes how these technologies come together to support composable energy solutions.

The future of grid software will need to be dynamically scalable to adapt to the constant growth of new data sources like DERs and EVs, and the new analytics that will consume that data. Such scalability will also help unlock many new and extended business capabilities, from performing hyper-scale contingency and constraint analysis (to better manage the complexity of modern grid risks) to transiently scaling services (for managing severe disruptive events). Section 7.2 describes how these technologies can deliver on scalability.

Grid software will need to provide an environment that empowers users and operators with data. This environment will facilitate a data-driven culture in comprehending the current and future states of the grid and supporting the decisions it will drive. Data will increasingly be used to solve problems that were traditionally solved physically.

An example is using high density weather data and line geometry to estimate dynamic line parameters, rather than deploying sensors. Section 7.3 describes how the core technologies will provide the data environment that simplifies the development, deployment, and lifecycle of advanced data-driven analytics and data self-service.

Driving innovation by freeing up data is certainly important for unlocking grid orchestration – but it is only a single step in the right direction. No single organization will ever hold the secret ingredient to the energy transition. Instead, developing the solutions to orchestrate the grid of the future will be highly collaborative. Section 7.4 describes how the core technologies can democratize the development of data-driven solutions and applications.

The expanding scope of cybersecurity threats (which, unsurprisingly, rises in unison with the increasing complexity of modern grids) requires a paradigm shift in how we secure the grid from cyber threats. For example, the expanding diversity of connected devices, both behind and in front of the meter, dramatically increases the attack service area. Section 7.5 describes how the core technologies support this paradigm shift in how we secure grid software.

COMPOSABLE ENERGY SOLUTIONS

A composable architecture provides the foundation for energy software solutions that can easily adapt to the rapidly changing environment of modern grids. It allows new capabilities and functions to be added independently of other functions, without disruption. Functions can be removed, scaled, and adapted to support rapid changes as well as adapt to long-term trends.

Composable grid software should enable modular functions through:

- Easy integration into the network and data models and a consistent approach to the evolution of these models.
- Simple access to data and feature sources.
- Consistent and automatic management of application results and output, making them discoverable, easily visualized, and easily integrated into actions.
- Consistent operation in compliance with data permissions, AORs, and data governance and policies.
- Compliance with security standards, including verifiable compliance for all components, ensuring system-wide security requirements.
- Harmonious solution-wide manageability and debuggability.
- Integration into UIs and workflows with a consistent user experience.
- Independent removal of old functions without disruption.

This level of composability sets a high bar, but contains a core set of requirements that must be met to ensure a viable, composable architecture within the grid domain. Although the core technologies described in this document will play a key part in achieving a composable grid software solution, they will need to be brought together within a design that is specifically engineered to deliver composability. This design will augment the technologies with services, software standards, and patterns that enable a composable solution.

Nonetheless, the following sections highlight how the core technologies provide the scaffolding for a composable grid software architecture.

MODULARITY

The core modularity of composable solutions can emerge from orchestrated microservice architectures (Section 2). Containerization provides a mechanism to package, distribute, and manage the individual modules as isolated software artifacts. Container images can be scanned, signed, versioned, and distributed in full compliance with grid standards (Section 2.1). They are also portable, allowing the same modules to run on premise, in the cloud, or on appropriate edge hardware. Container orchestration (Section 2.2), through orchestrators like Kubernetes, provides the software machinery to run and integrate multiple container workloads. Kubernetes also allows the core deployments, services, and architecture to be defined as code (Section 3). This enables strong version and change control to be applied as modular capabilities are added or removed from a system. As more functions operate in parallel, the integration into EaC is critical to manage the complexity in a reproducible and auditable way.

INTEROPERABILITY

Allowing these independent functional modules access to grid network and data models, and the range of associated data and features sources, can be provided by a data fabric (Section 5.2). Through the metadata catalog, modules can discover and identify the data they need to perform their function. This could be dynamic or, more likely, pre-engineered using the data fabric's self-service and discovery capabilities. If needed, new virtual datasets can be created using the self-service tools, designed specifically for the new functions. New virtual datasets are automatically registered with the metadata catalog, integrating into the data discovery, lineage, and governance processes.

Similarly, the outputs and data results of new modular functions, both real-time and historical, should be registered in the metadata catalog. The output then becomes available for discovery, active metadata analysis, and integration into the knowledge graph. Integration into the knowledge graph allows an organization to apply semantics to data, further facilitating its discovery and reuse.

Integration with the data orchestration capabilities of the data fabric, provides the tools for integrating new functional modules into dataflows and workflows following standard enterprise integration patterns. Importantly, the integrations should be specified as code, allowing them to be managed using the same EaC version control machinery mentioned above. Integrations could be with the wider system, such as transformation into external protocols, or with internal event processing and services like alarm management.

PATTERNS AND SHARED SERVICES

As mentioned above, a complete grid software solution consisting of composable applications requires dedicated, shared services performing common functions across the modular analytics and applications.

There will be many types of shared services, including:

- **Security**
  Providing the core authentication, authorization, secrets and certificate management, permissions and AOR infrastructure, user sessions, and session lifecycle management. All integrated into the cluster level security, such as service meshes and data ingress/egress controls (see Section 7.5).
- **UI Services**
  Providing the delivery of a consistent user experience (UX) across the different applications. Utilization of a design system governed by a common design language with reusable components.
- **Observability**
  Extensive and consistent metric and log collection and monitoring across the applications to provide near-real time and historical information reporting on the state of the various services and applications that comprise the solution.
- **Data Persistence**
  The data persistence layer provides managed services for various types of data stores integrated into the data fabric. Each store will be based on technologies optimized for holding specific types of data, such as time-series, relational, or data blobs.
- **Model Management**
  Providing access to the network, data, and ML models; again, tightly integrated into the data fabric. Model services are designed to provide access to grid network and data model information that is consistent and optimized for different app use-cases. This includes model change machinery for managing the evolution of network and data models.
- **Alarm Management**
  Alarm services and sub-systems for allowing composable applications to emit events and alarms in a consistent manner. Similarly to the other shared services above, it is fully integrated into the data fabric, allowing, for example, the alarms, alarm types, and associated schemas to be managed using the same data fabric tooling.

The shared services and the data fabric will be tuned to support many application "patterns" – typical flows and ways for applications to integrate with the system to perform their function. Patterns may range from pure event-driven apps, to streaming applications, to long-running stateful analytics. In a composable solution, patterns facilitate optimized mechanisms that improve performance and simplify common behaviors such as high availability and compliance to security policies. Building composable apps that do not follow patterns is still possible, but said patterns may not integrate or perform optimally. Patterns are a key part of composable solutions.

DATA GOVERNANCE AND DATA SECURITY

As mentioned in Section 5.1, collaborative, yet central, data governance is core to coherent data management and consistent policy-driven data security in DataOps. As the data fabric provides the mechanisms to access data and the machinery to perform data integration patterns, it can provide the middleware in which governance, permissions, and access control are applied.

Data lifecycle management is also facilitated by the data fabric. Consider one of the more complex, yet typical, data changes in grid environments – an update to the network model, such as a new line or feeder. In addition to the mechanisms and workflows to bring the physical equipment online, there is an associated data process by which applications that use the network model need to be notified and model updates synchronized across the composable solution. This is also the case for data model changes, such as new sensors or RTUs integrated into the system. The data governance capabilities of the data fabric, and in particular, the metadata catalog, together with the model management shared services, play a key role in providing data lifecycle management in a composable grid solution.

It's important to note that the data fabric is not the sole mechanism for data permissions and security. It plays a part in a wider and more comprehensive cybersecurity design (see Section 7.5).

SCALABILITY AND PERFORMANCE

Microservice architectures enable modular functions to be designed to scale horizontally and independently of each other (see Section 7.2). This provides a method of independently scaling capabilities and modules within a composable architecture; for example, to easily adapt to the rapid growth in data. However, a disadvantage of microservice architectures is the overhead of efficiently getting data to the functions, UIs, and systems that need it. A data fabric optimized for grid use cases (see Section 5.4) needs to operate within the performance profiles required for grid orchestration.

This too is where a grid-optimized data fabric plays a crucial role. For example, operational data, where performance is often both mandated and critical, will reside in the ODS. A modern ODS should be designed for analytical performance, while providing the transactional requirements of consistent grid data. The balance of analytical and transactional requirements, a normal contention in generic hybrid analytical/transactional processing (HTAP) databases, must be tuned for the grid use cases in a high performance ODS.

Moreover, the ODS must provide the associated data transport mechanisms to ship the operational data into apps with minimum latency. This requires the confluence of designing the data in representations that can be directly consumed by apps, and technologies that enable high performance transport.

A COMPOSABLE APP

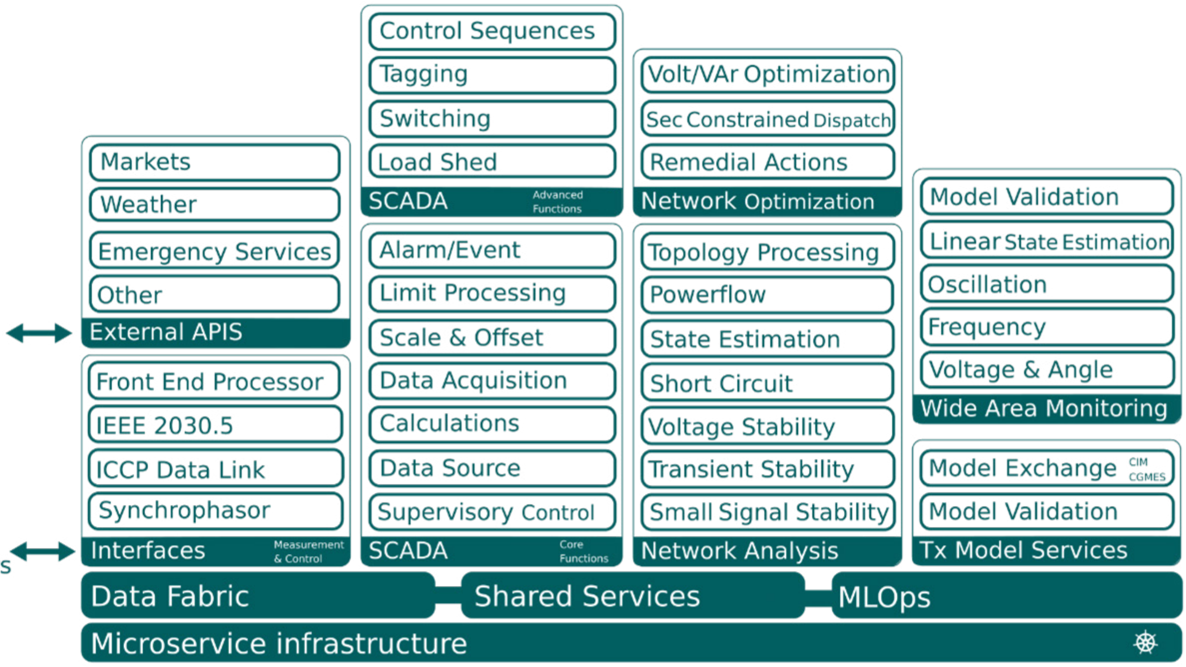
As described above, the architecture of a fully composable solution requires a detailed and consistent design, beyond the core technologies discussed in this document.Nonetheless, the technologies that enable independent microservices and their orchestration, plus a data fabric with the middleware for secure data access and lifecycle (both configured and specified using code, together provide the scaffolding for a composable grid solution.

Applications designed to fully operate in this type of composable architecture must:

- Be built using agreed-upon container technologies with minimum standards and security compliance (see Section 7.5).
- Be designed around the data fabric for accessing, processing, and contributing to data.
  - Fully participate in the metadata management, governance, and lifecycle the data fabric provides. This includes AI/ML model management and MLOps for AI/ML apps.
  - Fully participate in the data and permissions modelling provided by the fabric's modelling services.
- Follow the design patterns optimized for composable apps, and the common standards defined for the whole grid solution. This includes the patterns required for security, maintainability, scalability, and high availability.
- Utilize the common services for alarms, UI, simulation modes, etc. to fully integrate into the coherent whole.
- Provide the templates for all the configuration code required to deploy, configure, scale, and integrate into the system.

Although the technologies are critical for a composable architecture, the solution as a whole consists of services, patterns, and standards that must be well documented and enforced. Once in place, bundling the application container images and EaC templates together provides a composable unit, that can be easily deployed and integrated into a solution.

Figure 8: Example composable modular components configured for transmission functions on the shared infrastructure



Composable grid architectures can provide highly flexible, scalable, and resilient solutions that can support the dynamic requirements inherent in current grid operations. They allow a fluid composition of apps, enabling solutions shaped for transmission, distribution, generation, and energy market capabilities by simply layering on additional modular applications (Figure 9 illustrates a composable example for transmission). More importantly, composable grid architectures allow for continuums that cross the boundaries of traditional domains, expanding functionality to meet targeted needs. They also provide the ability to independently patch and upgrade individual functions. They offer the pliable scalability both for quickly adapting to transient events, and also for supporting long-term growth and change.

SCALABLE ENERGY SOLUTIONS

Application microservices can be designed to collectively operate so that modular functions can horizontally scale through the addition of new resources. Recall the simple example of DER growth in Section 2. Being able to flexibly scale modular components, optimize resources, and accommodate both transient and long-term demand on analytics are all significant advantages of orchestrated microservice architectures.

Performance metrics and analysis capabilities available through the observability sub-systems (see Section 7.1) allows the performance of the system to be effectively and appropriately tuned. This allows modules to be scaled both vertically (altering the resources individual modules use) and, where designed, horizontally (in which additional module replicas are deployed to take advantage of new hardware or resources and share the associated workload).

Modifying the architecture by increasing and decreasing module replicas may be automated; for example, through horizontal pod autoscaling (HPA) mechanisms in Kubernetes, or applied manually by changing the appropriate microservices' number of instances. Both automated and manual approaches integrate into EaC, ensuring the full architectural composition of the system is managed via code and strong version control.

Significantly greater IT resource scalability comes with cloud services. Consider, for example, engaging on demand in-depth cascading contingencies, or performing "what if" simulations, including transient dynamics, or scaling beyond locally available hardware resources during emergencies. The flexibility and horizontal scalability that cloud services provide can offer new capabilities that traditional on-premises resources cannot. A composable architecture based on the modular, EaC data fabric technologies described here provides the basis for building cloud-hybrid solutions that balance the enormous scalability of cloud with the on-premises system security. Moreover, the architecture allows this balance to be appropriately shifted for different applications and different needs.

DATA DRIVEN GRID ORCHESTRATION

Grid software of the future will provide an environment which empowers users and operators with data, facilitating a data-driven culture in comprehending the current and future states of the grid and supporting the decisions this will drive. Being able to access and interpret data easily and accurately, is one of the goals of DataOps and the data fabric (Section 5). The self-service capabilities allow users with different skills/personas to dynamically construct dashboards and visualizations that can transform data into effective decision support.

Data will be able to drive new capabilities that dedicated hardware traditionally supported . This is important as deploying new hardware is expensive and time-consuming. Examples could be using synchrophasor telemetry to provide effective measurements of inertia instead of deploying physical probes, or using high-density weather data and line geometry to estimate dynamic line parameters, rather than deploying sensors. Providing a data environment that brings together DataOps and MLOps within a composable architecture allows these types of data-driven alternatives to be developed, validated, and operationalized.

COLLEGIATE GRID SOFTWARE

No one company, nor a single organization, will be able to provide the complete suite of innovative and evolving software required for orchestrating the grid through the energy transition. A composable solution (Section 7.1), that allows customers, independent vendors, and even start-ups to safely develop, test, and operationalize modules will be required. A solution that (1) uses a data fabric, (2) standardizes data exchange using open APIs, and (3) centralizes data governance can provide the foundation for a community of applications that work consistently together.

The current ways of adapting grid software to specific needs or adding new features often involves building dedicated customs or providing the source code so utilities can modify the software themselves. These approaches to adapting software in response to specific, changing needs is fragile, leading to multiple branches, poor testing, and slow upgrades. A composable architecture allows the system to be easily extended using new modules.

Examples include:

- Integrating entirely new features and analytics into the system.
- Adding modules that modify steps in current workflows, such as data pre-processing or validation.
- Adding new authorization steps in certain actions, like the four-eyes principle.

The flexibility of a comprehensive and composable solution (Section 7.1) built on open standards allows the system to be expanded, changed, and tailored. As a result, the individual software modules can follow best practices in their development lifecycles, testing, and validation.

A NEW CYBER SECURITY APPROACH

As the complexity of the modern grid increases, so does the surface area of cybersecurity threats. These two happenings call for a major change in securing the grid from cyberthreats. For example, the expanding diversity of connected devices, both behind and in front of the meter, dramatically increases the attack surface area. Bringing in more types of data – for example, insight-rich advanced metering infrastructure (AMI) or EV data – can greatly help optimize grid orchestration, but can also create more points for accessing critical systems.. The increasing application of AI/ML and data-driven analytics also yields new cyberthreat avenues, such as polluting the source data to achieve designed outcomes. As discussed in Section 7.2, increasing use of cloud resources also adds significantly to the threat surface area. Traditional, isolation-based security is no longer sufficient to manage the magnitude of changes in cybersecurity threats.

A paradigm shift in how we secure grid software is required to safely use the modular, flexible, and data-driven technologies and architectures and the advantages they bring. Fortunately, other highly regulated and secure industries have successfully made the transition to many of the technologies and approaches that have been discussed. The core principles that have been developed and successfully operated to secure such systems is called Zero Trust.

Zero Trust, as the name suggests, is about removing trust from all parts of the system. For example, rather than trusting communications within a secured isolation zone, no communication whatsoever is trusted. Every communication requires mutual authentication and encryption. This includes communications within the microservice architecture and between individual microservices. There are standards that now define how Zero Trust principles are applied and architected, such as NIST 800-207. Zero Trust fundamentally changes all parts of a solution and generally cannot be bolted onto architectures not designed for it. Consequently, grid software that follows cybersecurity best practices in the complex and expanding threat environment must be designed from the ground up with Zero Trust grid security principles.

MANAGING THE TRANSFORMATION

Grid software of the future will be fundamentally different in its architecture and core technologies compared to today's software. **There are several best practices that can simplify the digital transformation and allow some of the advantages of the new software paradigm to be leveraged earlier and with minimal disruption.**

ENABLE EXTENSIBILITY

A transition pathway is fundamental to the success of the digital transformation. This allows modular apps and solutions to be employed early and integrated with traditional/established products. Existing grid software solutions can integrate with new modular applications to allow the addition of new capabilities while providing trusted and established classic solutions that operate in parallel. Ideally, integration should happen directly via the data fabric, enabling the established products to fully participate and engage in the evolving data landscape.

However, integration can still be accomplished via several routes:

- Integration into a data fabric. This may be partial, depending on available integration points
- Direct integration with an OSB
- Dedicated open APIs
- Microservices providing integration capabilities for older proprietary APIs
- Common information modelling (CIM) tools Common and integrated authorization standards

By using these extension capabilities, new modular applications and functions can be operated alongside existing grid management solutions, both enhancing and extending functionality (see below).

INCREMENTALLY ADD CAPABILITIES

Given the magnitude of the projected changes to today's operational environments, an incremental approach to progress based on consideration of derived benefits and associated costs, difficulty, and time should be adopted. While specifics will vary from one utility to the next, the following general areas are believed to hold significant opportunities for most.

- **Addition of new containerized applications to extend existing functionality**

  As introduced in Section 8.1, being able to run composable architectures side-by-side and integrated with established systems is highly advantageous when scalability, modularity, and reductions in time-to-value are goals. Applications that exhibit extreme performance requirements or are likely to evolve quickly due to changes in utility communication, control technologies, and/or regulatory frameworks are excellent candidates for this. In addition, applications that can be loosely coupled to existing grid management platforms are also targets for early migration.

- **Cloud hosting of non-production environments or study functions**

  As non-production environments and study functions tend to exhibit lower utilization levels, requiring less stringent levels of availability, and are in general deemed less sensitive as compared to production environments, they are typically excellent targets for transitioning to cloud. Primary advantages of such a transition are speed when it comes to provisioning and the ability to create and destroy such environments on demand. The capability to dynamically scale capacity as needed in support of increased training or test needs is also a potentially significant advantage. In addition, where its use is feasible, the public cloud can offer significant cost savings over traditional on-premise deployments. This is because dedicated capacity is not required and pay-for-what-you-use billing is generally available.. Lastly, transition of non-production and study mode environments offers an opportunity to learn cloud technologies and providers in a limited, gradual way and to better assess the potential applicability to other use cases.

INVESTMENT IN TRAINING

The underlying technologies present in the grid management systems of the future are likely to be very different from those used for traditional Windows and Linux administration. Understanding how these technologies work at a conceptual level is essential to understanding their potential applicability and, ultimately, to achieving the desired outcomes. In addition, the willingness and flexibility to re-examine existing business processes considering such technological changes is key to extracting the full benefit. Thankfully, there are many sources of training available today for most of the core technologies described in this paper.

ADOPT AND APPLY IT BEST PRACTICE

While their underlying technologies are quite different, most, if not all, of the practices applied to traditional bare-metal or VM-based architectures are also applicable to both microservices, container and cloud. In particular, the best practices focus on adoption of IaC, application of network access control, identity access management, encryption of data in motion or at rest, general system hardening, and backup and restoration are all appropriate.

ENGAGEMENT AND COLLABORATION

When developing a digitization strategy for operations, it is important to connect with potential partners to understand their roadmaps and perspectives on market trends. Given the pace of technological changes, it is also important to establish a regular rhythm for updating said roadmaps. CSPs can often offer invaluable insights into how other industries have approached or are currently approaching similar challenges.

REGULATORY BODIES

Existing cybersecurity regulations governing electric utility operations were developed based on the historical usage of on-premises data centers, which were typically deployed on bare metal. Accordingly, they often do not explicitly consider the use of modern technologies including hardware virtualization, containers/microservices and cloud, leading some to conclude that the use of these technologies is prohibited when in fact they have yet to be fully evaluated.

Regulatory bodies prefer to work with their industries' members when developing or revising standards (rather than simply dictating policy). This provides an opportunity for early adopters to work collaboratively with their regulators to inform and adapt such regulations for the good of their industries and consumers. For example, the North American Electric Reliability Corp (NERC), formed the Security Integration and Technology Enablement Subcommittee (SITES) to identify, assess, recommend, and support the integration of technologies on the bulk power system (BPS) in a secure, reliable, and effective manner. SITES activities are intended to help the electric utility industry adopt emerging technologies in a secure, reliable, and resilient manner to ensure reliability, security, and resilience of the BPS. The SITES team also may develop a guidance document outlining how to take that path forward without stifling innovation.

LIST OF ABBREVIATIONS

AC	Alternating Current
aFRR	Automated Frequency Restoration Reserve
AI	Artificial Intelligence
AMI	Advanced Metering Infrastructure
AOR	Area Of Responsibility
API	Application Programming Interface
BESS	Battery Energy Storage Systems
BPS	Bulk Power System
CaaS	Container as a Service
CD	Continuous Delivery
CET	Consumer Energy Technology
CI	Continuous Integration
CIM	Common Information Model
CRI	Container Runtime Interface
CSP	Cloud Service Provider
CVE	Common Vulnerabilities and Exposures
DER	Distributed Energy Resource
DPMU	Distribution Network PMU
DR	Distributed Resource
EaC	Everything as Code
ET	Energy Technology
ETL	Extract, Transform, Load
EV	Electric Vehicle
FaaS	Function as a Service
FERC	Federal Energy Regulatory Commission
GHG	Greenhouse Gas
GPU	Graphics Processing Unit
GW	Gigawatt
HPA	Horizontal Pod Autoscaling
HTAP	Hybrid Analytical/ Transactional Processing

IaaS	Infrastructure as a Service
IaC	Infrastructure as Code
IED	Intelligent Electronic Device
IoT	Internet of Things
IT	Information Technology
LLM	Large Language Model
ML	Machine Learning
NEPA	National Environmental Policy Act
NERC	North American Electric Reliability Corporation
NIST	National Institute of Standards and Technology
OCI	Open Container Initiative
ODS	Operational Data Store
OSB	Operational Service Bus
OT	Operational Technology
PaaS	Platform as a Service
PMU	Phasor Measurement Unit
PUHCA	Public Utility Holding Company Act
QOS	Quality of Service
RTU	Remote Terminal Unit
SaaS	Software as a Service
SBOM	Software Bill of Materials
SCADA	Supervisory Control and Data Acquisition
SDA	Software Defined Architecture
SITES	Security Integration and Technology Enablement Subcommittee
TPU	Tensor Processing Unit
UI	User Interface
WASM	Web Assembly
YAML	Yet Another Markup Language

GLOSSARY

AIOps	The application of AI/ML technologies, including NLP models, to enhance an organization's full range of IT operations. This includes automating IT services and ticket management, through to intelligently analyzing the large amounts of data generated through DevOps, DataOps, MLOps etc.
DevOps	The combination of business processes, tools that promote automation, and culture that increases an organization's ability to deliver applications and services through software development and infrastructure management. See Section 3.1
DataOps	The combination of business processes and technologies that provides a process-oriented approach to data and promotes a culture of continuous improvement in data analytics. See Section 5.1
Enterprise Integration Patterns	Standard message and data integration patterns that have emerged as effective and widely used designs over the last few decades. Collected and published in a book by Gregor Hophe and Bobby Woolf.
MLOps	The combination of business processes, technologies and automation that increases an organization's ability to deliver, operationalize and continuously improve the use of machine learning analytics and capabilities. See Section 6.1
ModelOps	The combination of business processes and technologies that provide governance and life-cycle management for operationalized AI and ML models.
SecOps	Combining internal information security and IT operations practices to automate and improve cyber security processes and risk management.
XOps	See Section 6.3

CONCLUSION

Modularity through microservices and containers, EaC, cloud, DataOps with data fabric tooling, and MLOps delivering AI/ML applications are all established technologies that, when combined, can have a dramatic and multiplicative impact. They collectively deliver the ability to scale on demand while reducing cost. They also unlock rapid time-to-value for adapting solutions, both with new features and evolving solutions through a composable architecture. Their presence in utility operations, while limited today, will be essential in achieving the digital transformation required to orchestrate the grid of the future. A successful strategy for achieving such a transformation will include establishing clear outcomes, investing in education, early experimentation, willingness to reconsider existing business processes, regular communication and collaboration with technology and application providers, and the ability to pivot as technology matures.

MEET OUR AUTHORS:

Dr Andrew Gillies

BSc. MSc. PhD. – GridOS® Applications CTO

Andrew led the early validation of the modern transformative technologies that now forms the core of the GridOS® strategy. Bringing together orchestrated microservice architectures, containers, Kubernetes, Kafka, and other cloud-native technologies, he demonstrated a full suite of real-time WAMS in 2019.

Jens-Martin Grønne

MSc.– GridOS® Platform CTO

Jens-Martin joined GE Vernova through the recent acquisition of Greenbird Integration Technologies. He was a cofounder and led the development of the Utilihive platform. Utilihive is a cloud-native, highly scalable enterprise iPaaS purpose built for the utility industry.

Michael Unum

Principal Digital Product Manager

Mr. Unum is responsible for establishing the strategic direction for next generation solution for orchestrating the sustainable energy grid. He joined Harris Controls in 1983 and joined GE Vernova by acquisition in 2000. He holds a BSEE degree from the Georgia Institute of Technology.

Renan Leites

GridOS® Solution Architect

Renan graduated as a BSEE with a Master`s degree in Power System Engineering from the Universidad Federal de Santa Catarina. He is currently focused on connecting business opportunities and technology advancements to provide a flexible and agile adaptation to the current global energy transition.

GE VERNOVA

GridOS® Orchestration Software

ORCHESTRATING A MORE SUSTAINABLE ENERGY GRID

Get in touch

☎ 1-833-690-5552