Government Enterprise Architecture Reference

The Government Enterprise Architecture Reference (GEAR) is not just a tool but a pivotal instrument for digital transformation in government organizations. Its structured approach is the backbone for implementing digital transformation visions, underscoring its crucial role.

Authors

Dr. Darren Pulsipher
Dr. Anna Scott

Table of Contents

Introduction

Digital transformation is the application of computing, networking, storage, and analytic technologies to data to enable organizations to modify, enhance, improve, or deploy more efficient, more compelling, and more timely services to their constituents and organizations while meeting mission outcomes. It's the opportunity to reimagine organizational practices and capabilities in light of technological innovations. New technology can enable the digital transformation of government, but determining where to start, what provides value, how new capabilities work with existing systems, and what is economically feasible can be daunting. Many governments and government organizations have a vision for digital transformation. However, mapping their vision to innovative technology and existing systems can be complex. This paper presents an approach to help governments implement their visions while taking advantage of current systems (no need to rip and replace) and ensuring that the new capabilities are interoperable, manageable, future-proof, and scalable.

Digital Transformation Methodology

Only some organizations have the luxury of starting from scratch when embracing digital transformation. Almost always, an organization has very well-developed systems that have existed for years or decades. These systems also represent substantial investment and significant institutional knowledge. Many of these systems work well enough for the businesses' needs and do not need to be altered. However, they may have valuable data that – if it were available – could create additional value when looked at at an organizational level. Alternatively, existing systems may no longer be able to provide the level of service needed for the organization to be competitive. Matching current systems to the future vision and performing a gap analysis provides a straightforward methodology for getting started with digital transformation.

The following steps can be taken to identify gaps and find solutions:

1. The first step in the methodology is to define the problem(s) to be solved and the vision of the future state. This strategic planning is crucial for the success of digital transformation.
2. Define the current state (including data sources and existing software and hardware).
3. Map your current capabilities to the Government Enterprise Architecture Reference (GEAR).
4. Assess gaps.
5. Develop a plan for how to fill these gaps using the reference architecture as a guide for an interoperable future state.
6. Implement projects to fulfill your vision based on organizational priorities.

This oversimplifies but provides a starting point for making high-level decisions and tackling a complex problem. In this context, the GEAR offers a basis for understanding what new technology can provide and how it works together. It helps organizations "see" the art of the possible to decide quickly where technology can bring organizational value. Edge-to-cloud architectures are inherently complex and can rarely be solved by a single company. This means that implementation of the reference architecture, especially when integrated with existing systems, will require multiple partners and some customization.

GEAR is designed to provide a data management structure for large organizations with complex requirements. If this approach is followed, an organization can add new capabilities that will integrate existing systems in an open and interoperable fashion. Even if an organization starts with only a few specific problems to solve in the short term, it's essential to plan for existing changes in the future. This will minimize complexity, cost, and maintenance as an organization continues its digital transformation journey. Digital transformation is a journey that can take years to complete, so this approach is an iterative process applied as the organization grows and evolves.

GEAR Overview

This paper proposes a Government Enterprise Architecture Reference (GEAR) suitable for roughly 80% of the government use cases relevant to digital transformation. GEAR is designed to provide a foundational structure supporting a wide range of applications and services using all data available across an organization. This basic architecture will need customization because of the wide variety of use cases and existing systems. Additionally, there will be many GEAR instances because multiple suppliers and software vendors can provide each logical architectural component.

GEAR is based on open, interoperable systems since proprietary ones are constrained to single usage and can add cost and complexity if done at scale. GEAR is designed to work with existing systems where possible and incorporate new solutions where valuable.

We explain the GEAR using two diagrams:

1. Physical – Diagram to show how to compute maps to applications, networks, and physical locations.
2. Logical – Diagram to show the underlying software stack that manages data, storage, orchestration, applications, and services and how these are supported by hardware (HW) and networks.

Physical GEAR

A single, high-level diagram (see Figure 1) showing how the primary data sources and their physical locations can benefit digital transformation. For example, this mapping indicates the relevance of networks in the overall architecture. In enterprise, we take it for granted that we have high reliability and bandwidth connectivity. For many operational data sources, this is different. Data may be needed by a critical function (related to health and human safety), meaning that a typical cellular network does not have the required ultra-high reliability.

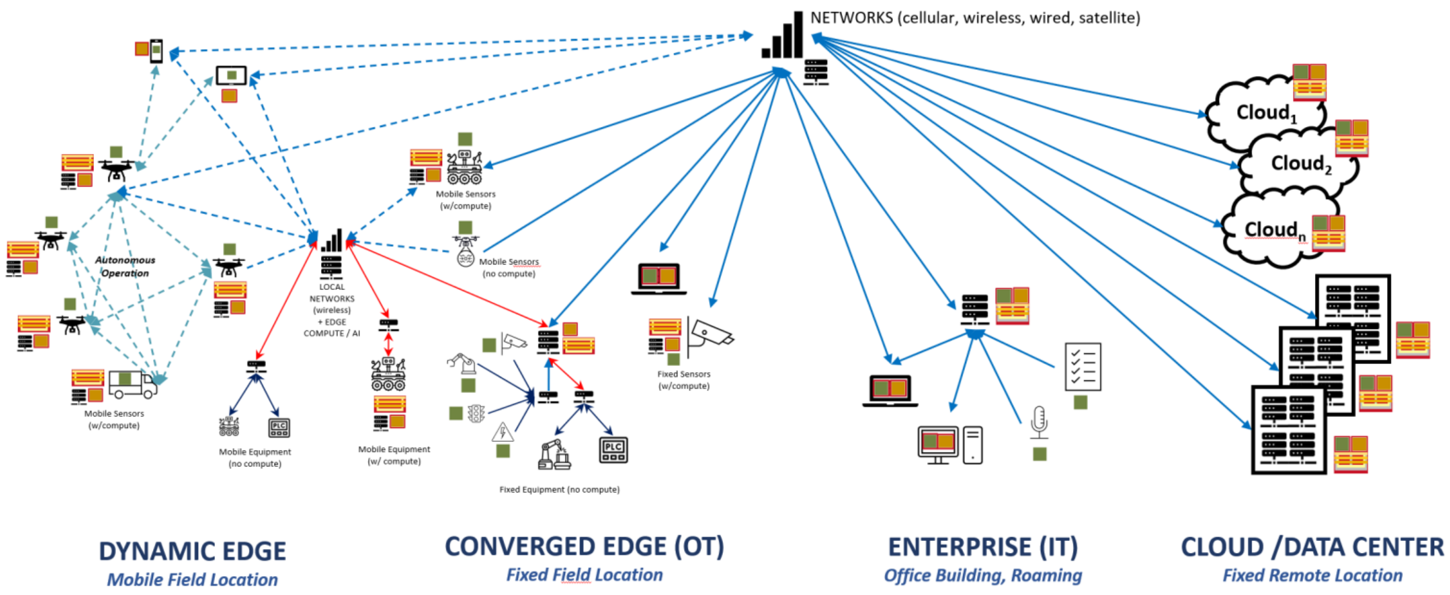

NETWORKS (cellular, wireless, wired, satellite)
Cloud$_1$
Cloud$_2$
Cloud$_n$
Autonomous Operation
Mobile Sensors (w/compute)
Mobile Sensors (w/compute)
Mobile Sensors (no compute)
LOCAL NETWORKS (wireless) + EDGE COMPUTE / AI
Mobile Equipment (no compute)
Mobile Equipment (w/ compute)
Fixed Equipment (no compute)
Fixed Sensors (w/compute)
PLC
PLC
DYNAMIC EDGE
Mobile Field Location
CONVERGED EDGE (OT)
Fixed Field Location
ENTERPRISE (IT)
Office Building, Roaming
CLOUD /DATA CENTER
Fixed Remote Location

Figure 1 Physical Representation of the Government Enterprise Architecture Reference

🟧	Secured Applications	↔	Network (cellular, wireless, wired, satellite – TCP/IP)
🟧	Applications	⇠⇢	DDIL* Network
🟩	Secured Data	⇠⇢	DDIL* Mesh Network
🟩	Data	↔	Proprietary Protocols (Modbus, CANbus, Profibus, OPC-UA, etc)
▤	SW Stack	↔	Control / Critical Operation
▤	HW + SW Stack		

Figure 1 illustrates how data can be collected, analyzed, shared, and stored from edge to cloud using a reference architecture built on a standard software (SW) stack that can operate on almost any type of hardware. The GEAR's design is highly flexible, empowering organizations to move applications and data based on business needs and the economics of data transport, storage, and computing costs.

This physical diagram grossly oversimplifies almost infinite use cases and existing applications. It is intended to help organizations map where their data sources, computers, and networks can reside and easily visualize how these elements come together to form a workable architecture. This is especially important for the dynamic and converged edge environments. Existing systems in these areas are often designed and installed as OT (operations technology) that is usually hardwired and heavily reliant on isolation for security. Bringing data into a converged architecture from these data sources poses unique challenges (protocol language, security, update frequency, etc).

Logical GEAR

A cloud or edge-only architecture can solve a few of today's data problems. Resources from the edge to the cloud are often needed to optimize the solution for effectiveness (e.g., real-time) and cost (e.g., total TCO). As a result, we have developed a conceptual architecture that targets multi-hybrid cloud and edge computing strategies, where data and applications can be moved as needed to optimize results and deliver on the mission.

To enable a future-proof and expandable system, it is essential to understand how different parts of the system relate to each other and establish isolation layers (through standard interfaces or abstractions). This isolation allows the various systems in the solution to "grow" in parallel with minimal effect on each other.

It also allows multiple vendors to be used within each layer, making the system flexible and adaptable with no vendor lock. Establishing standard interfaces between the sub-systems further allows the easy adoption of new features for hardware or software based on use case needs. This is the key to GEAR's flexibility and interoperability.

The logical enterprise architecture (Figure 2) shows how the different subsystems (layers) fit together.

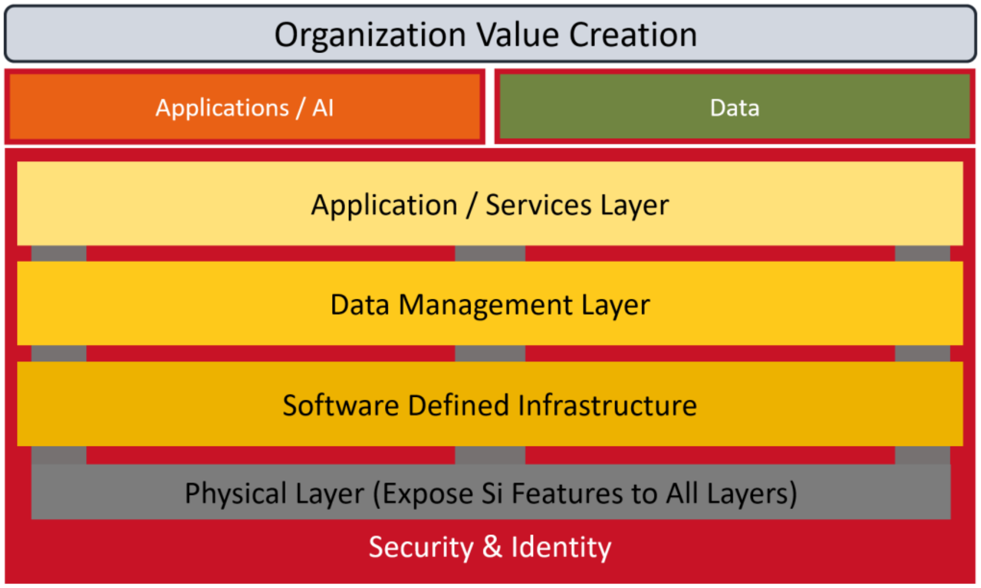

Organization Value Creation
Applications / AI
Data
Application / Services Layer
Data Management Layer
Software Defined Infrastructure
Physical Layer (Expose Si Features to All Layers)
Security & Identity

Figure 2: Logical Representation of the GEAR

Because the purpose of changing existing systems and adopting new technology is to improve outcomes and better meet mission/organizational needs, we place "organization value creation" at the top of the stack. It's critical not to lose sight of the purpose of digital transformation and to ensure that innovations are employed to provide clear benefits to the organization and its constituents. Value can take many forms, but some common examples are:

- High-level situational awareness for management (common operating picture).
- Improved constituent services through real-time data access and analysis.
- Planned downtimes enabled by predictive maintenance.
- Improved organizational efficiency.
- Lower cloud, storage, and compute costs.

Data and applications are undervalued because they form the foundation for value creation. Data may be the new oil, but how data is used by applications, analytics, and AI creates value. Since data and analytics do not have to be collocated, we have called them separate entities in the reference architecture.

The other layers in the stack form the structure that allows data to be collected and analyzed. We will first briefly overview each layer and then cover each in depth.

- Application / Services Layer—Development, testing, deployment, monitoring, and provisioning of services and applications in the solution space. This is the primary interface to Organizational Value Creation.
- Data Management Layer—Manages (curation, governance, lifecycle management, and tagging) data across a heterogeneous infrastructure (Cloud, Data Center, Edge, and Client).
- Software-defined Infrastructure—Responsible for managing the physical layer's solution (deploying, monitoring, and provisioning).

- Physical Layer—This layer is Responsible for commanding, controlling, and monitoring the solution's physical devices (Compute, Storage, Network, and Accelerators).
- Security Aspect – Gives a standard security model across the subsystems of the solution.
- Identity Aspect—The ability to uniquely identify and attest the identities of users, hardware, applications, services, and virtual resources.

Many organizations already have many of these subsystems in their toolbox of solutions, so developing a solution from scratch is rarely necessary. The recommended starting point is to understand what you are currently using, how those tools fit together, and how they interface. It is often possible to build in state-of-the-art features (use of AI models) by expanding current capabilities. No one wants to rip and replace systems that are in place and working. Instead, the goal is to utilize the current devices as a foundation to build for the future goal. A roadmap of technology and process changes shows how the foundation can be built upon to achieve long-term architecture.

Application / Services Layer

The Application / Services Layer can be broken into 2 component layers (see Figure 3):

- The Application Management Layer (AML)
- The Services Management Layer (SML)

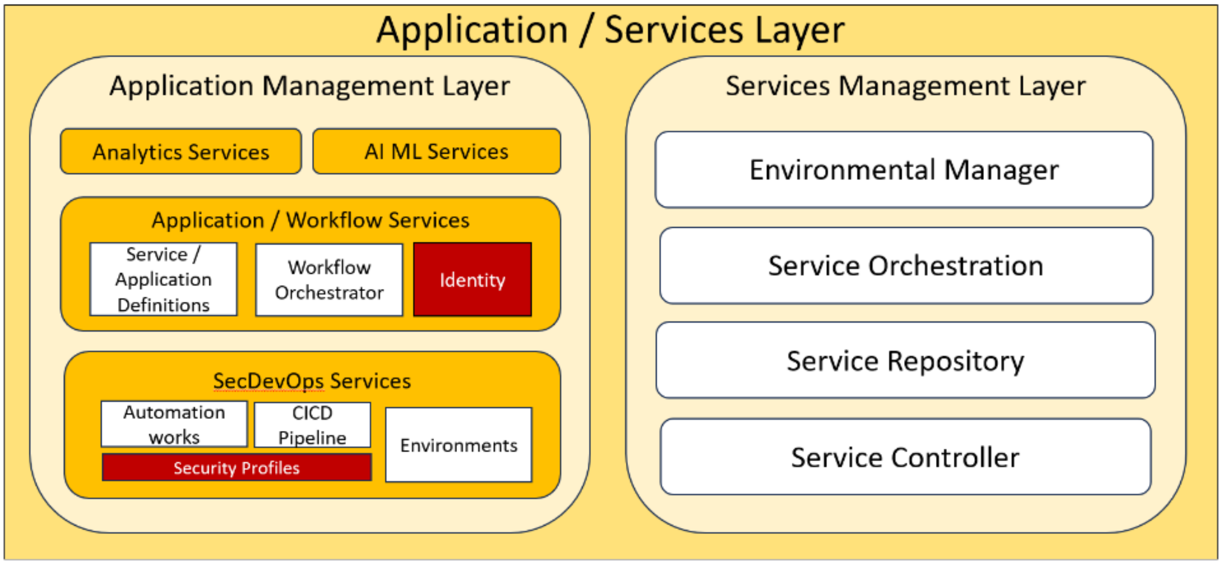


Figure 3 Application Services Layer

Application Management Layer

The AML manages applications and workflows and the development, testing, deployment, and updates of those applications and workloads. The AML contains abstractions that help App Development, DevOps, and IT Operations manage complex workflows and applications through the application development lifecycle. The AML sits at the top of the system stack and communicates directly with the Data Management Layer and the Service Management Layer. It also leverages the Identity Aspect and Security Aspect layers. Because certain applications operate most efficiently on specific hardware types, the Physical Layer and its capabilities are exposed to the AML. For example, training a large language model runs on Intel Habana hardware most efficiently.

The AML contains sub-packages that group common-off-the-shelf tools together. In the Analytics Services package, standard tools for data analytics can be found, including business data understanding, modeling, and simulation. AI/ML Services represent various tools and services focusing on Artificial Intelligence and Machine Learning algorithms and solutions. These two packages take advantage of the application and workflow services that allow these solutions to be orchestrated at the highest level of integration by providing a standard definition framework to show how these applications and workflow interact.

Several tool suites have been built to aid DevSecOps. These tools are grouped in the DevSecOps Services and include Automation Frameworks (Salt, Chef, puppet, ansible), CICD tools (Gitlab, GitHub, Jenkins, etc.), and Environment Management.

Service Management Layer

The Service Management Layer (SML) subsystem manages services, stacks, environments, and multi-clouds (see Figure 4). It is a middleware layer in the architecture responsible for orchestrating and managing services across multiple clouds (public and private) and the edge. The SML coordinates with the Data Management Layer and the Software Defined Infrastructure. It takes requests from the AML to deploy services that makeup applications and workflows.

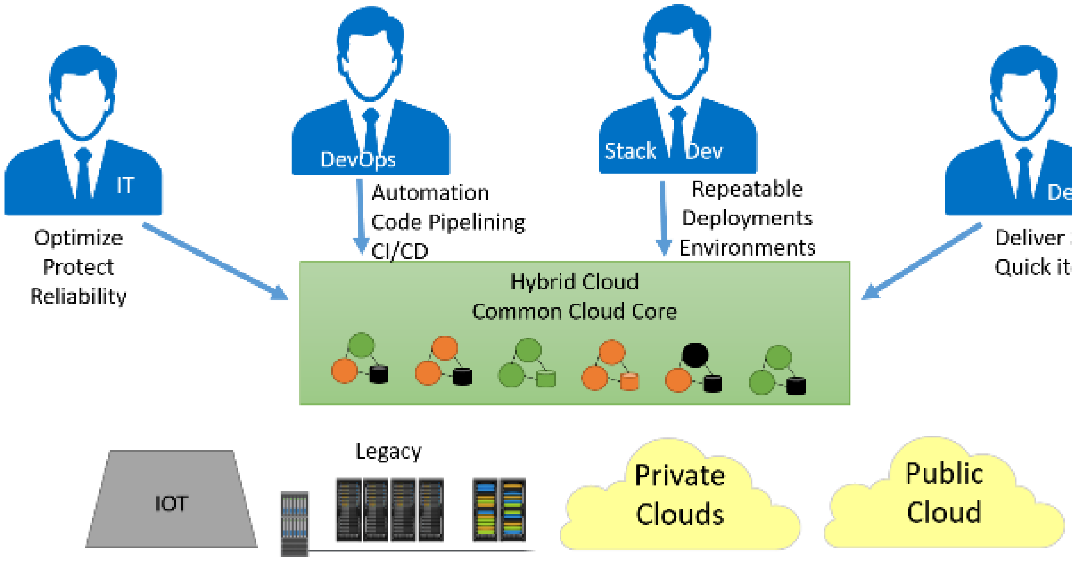


Figure 4: Actors of the Service Management Layer

The main goal of the SML is to provide the Application Developer with a simple, repeatable, robust mechanism to deploy services into the multi-cloud and edge ecosystem. It must also offer IT Operations mechanisms to enforce cost, reliability, and security policies. Applications and Services are deployed to cloud assets based on these policies and can be run across cloud/edge boundaries as dictated by the IT policies enforced in the system. All communications between services should follow secure communication protocols as the IT policies dictate. The key is that a single portal or gateway should be used so that applications and services are deployed and managed automatically without human interaction. Decisions about where services should land should be automatic based on the IT policies established, not determined by the Application Developer or IT Operations Engineer.

The SML has several actors who work with the subsystem. Each one of these has a different motivation for using the system. Even though some of the methods used are the same, their reasons for using the system are very different.

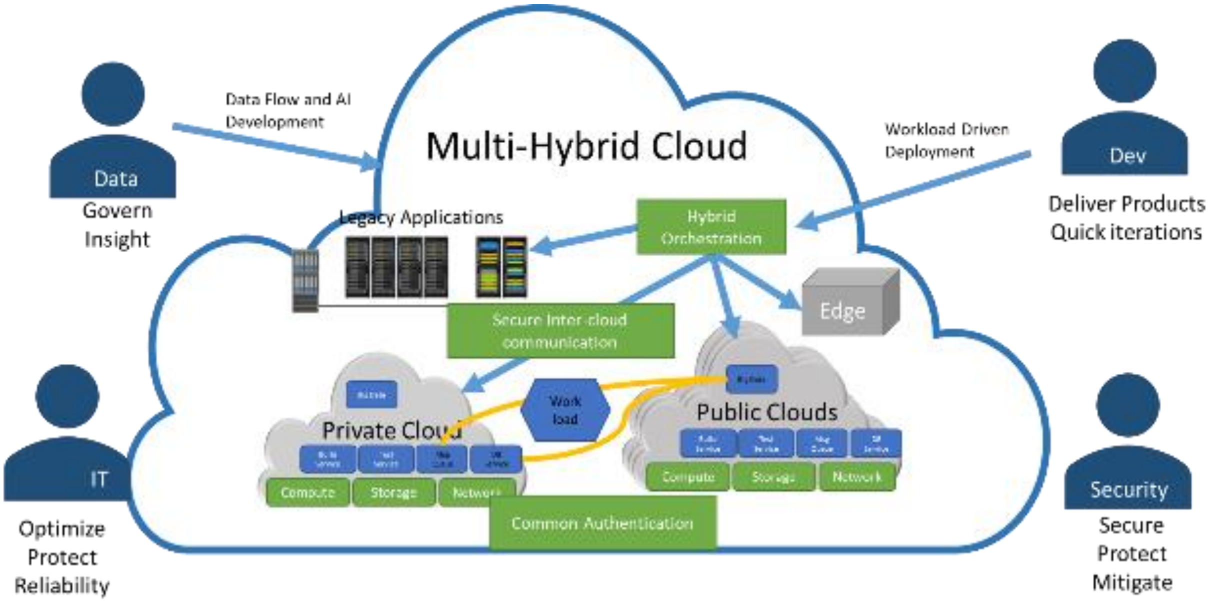


Figure 5: Motivators of Actors of Multi-Hybrid Cloud

- <u>IT Operations Motivators</u>: optimizing infrastructure for cost, protecting infrastructure and IP, and increasing reliability and resiliency.
- <u>DevOps Engineer Motivators</u>: automating everything, streamlining code pipelining, and managing build and deployment with CI/CD.
- <u>Application Developer Motivators</u>: repeatable and reusable service stacks, deploying services across cloud and environments.
- <u>Stack Developer Motivators</u>: delivering solutions in quick iterations, concise break, fix, and deploy cycles.

The SML spaces are full of tools that can be integrated to deliver the use cases demanded by the actors in this space, which can be categorized as follows:

- Cloud Management Platforms - built for IT Operations focusing on multi-cloud support and management of infrastructure profiles across multi-clouds.
- Automation Frameworks - built for DevOps and Stack Development, focusing on providing and deploying software in a repeatable/reusable manner.
- Platform As A Service—built for Application Developers, it focuses on reusing services and decreasing the complexity of using those services to build applications.

These tool sets have been developed by and for specific actors. Integrating these tools helps to fill gaps in the individual devices.

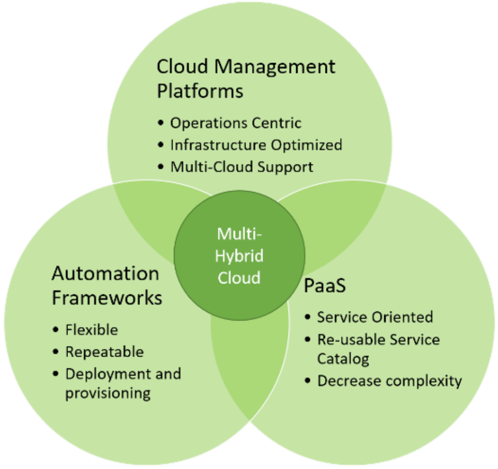


Figure 6: Convergence of tools

The SML has sub-systems as part of the multi-cloud architecture (for private, public, and micro clouds), including:

- On-demand self-service portal

- Environment Management - manages environments (dev, test, prod) across multiple clouds.
- Service Orchestrator - orchestrates the services in different environments.
- Service Registry – provides a centralized repository of service definitions.
- Data Coordinator - works with the DML to orchestrate data and services.
- Security - works with the security policies and tools to ensure applications and services communicate securely.
- Provision Engine - provisions software stacks and services on infrastructure.
- Cloud Broker - manages the clouds (which can manage which request).

Data Management Layer

The DML is the newest architectural element in enterprise architecture and is crucial in providing the flexibility needed for today's demanding computing environments. It was developed to handle the complexity of managing data across multiple data centers, clouds, and edge devices. The Data Management Layer (DML) manages data across the ecosystem, including data lifecycle management, data security and governance, storage infrastructure, analytics, data sources, and application data usage. Across all architectures, there are three standard components:

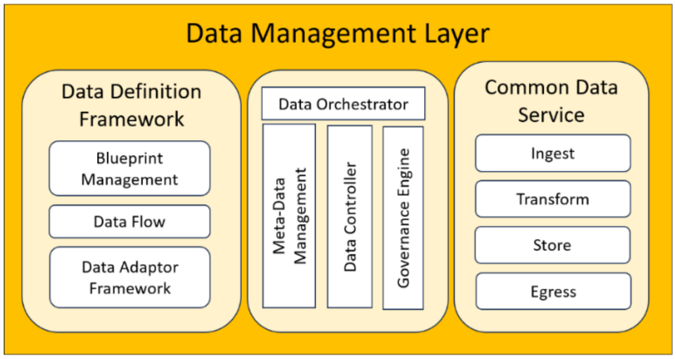


Figure 7: Data Management Layer

1. The Data Management component orchestrates the ecosystem's data movement, lifecycle management, and governance.
2. Data Definition Framework that defines data pipelines, categorization of data, and their generating or storage sources.
3. Common Data Services like ingress, transform, store, and egress.

Almost all architecture actors have some input into the DML:

- Data Officer – set data policy and strategy.
- Data Steward – manage data and policies.
- IT Operations – manage infrastructure.
- Application Developer – develop Apps.
- Data Scientist – analyze data and derive intelligence.
- Data Engineer – manages sources, blueprints, and procedures.

The DML subsystem supports multiple data architectures at the same time. This allows the Data Engineer to quickly build repeatable blueprints for different data architectural approaches based on the most efficient for a specific problem. As a result, the same system or solution can utilize various operating data models. Data models can be categorized into two architecture types:

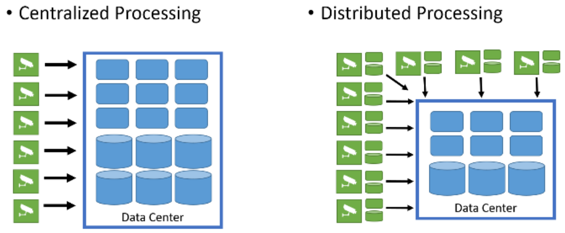


Figure 8: Data Architecture Types

- Centralized – Data Warehouse and Data Lake
- Distributed – Data Mesh, Data Exchange, Data Fabric, Data Mart, and Data Streams

The centralized processing approach utilizes data architecture to benefit the end users. This paradigm is good for some data use cases but not all. Many centralized data architectures fall apart as more systems move outside the traditional data center walls. This is where distributed processing architectures become essential. The distributed architecture is flexible enough to handle data processing modes from edge to cloud. Since these architectures are more numerous and less well-known, we will discuss them in more detail below.

Distributed Operating Data Models

Data Mesh

In the Data Mesh architecture, applications can be moved close to the data or the data close to the applications. Data processing is done on edge devices, and results are pushed to the data center/cloud to be linked. This contrasts with traditional Data Warehouses and Data Lakes, where data is stored in a centralized location.

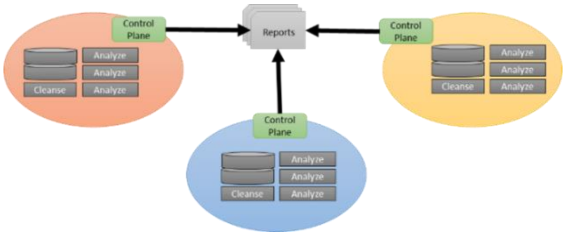


Figure 9: Data Mesh

Data Exchange

Another mode of operation is Data Exchange. This takes the Data Mesh concept and extends it to different classifications or owners of data. This mode limits the movement of data and who has access to it, making it ideal for Government and Healthcare, where privacy and classification regulations restrict data access. Data Exchange architecture allows policy gates to limit the data that can be passed back to the application requesting the data. It also provides analytics/services for the geo-fenced data site.

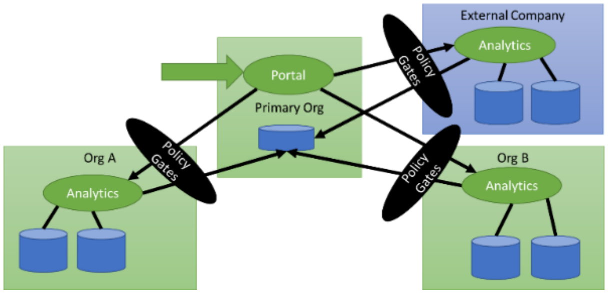


Figure 10: Data Exchange

Data Fabric

Data Fabrics are the natural next architectural evolution to emerge since they resolve some of the problems with Data Lake architectures caused by centralizing all of the data. Data Fabrics process data on the edge where the source generates the data. This distributed architecture follows much of the same path that cloud technology did in the early 2000s and includes centralized control, orchestration, and management of the data.

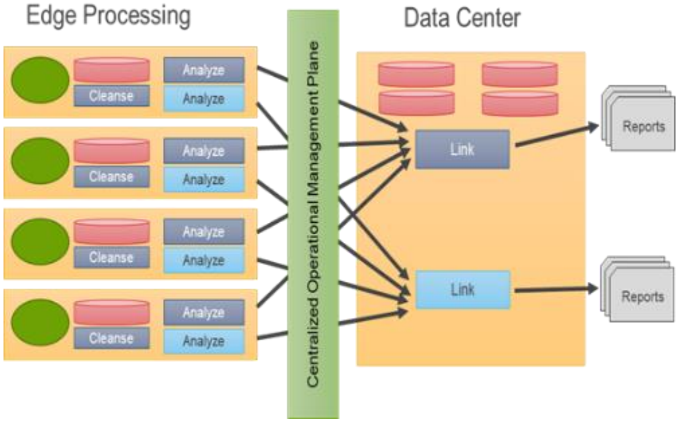
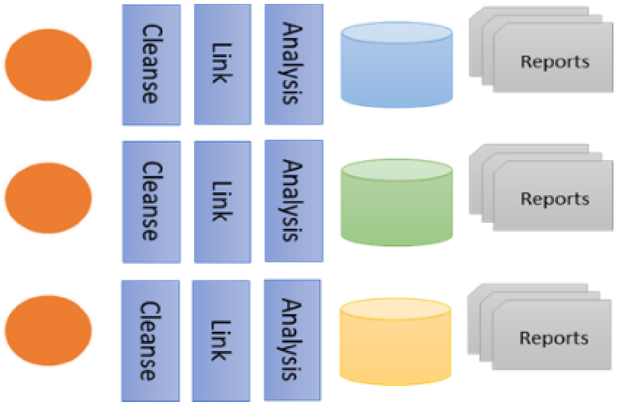


Figure 11: Data Fabric

Data Mart

A Data Mart is a small data repository for structured data specific to a department. Tailored to the detailed problem statements, they contain copies of data from limited sources and typically a smaller data set. Data Marts usually limit access to the data and report to one organization or a small group of users in one organization. Data scientists leverage Data Marts to build complex analytical models, generate timely periodic reports that require highly predictable performance, and work with sensitive data and resulting reports.



.

Figure 12: Data Mart

Data Streams

Data Stream architectures allow data analytics to be processed in the data stream. Each Data Stream manipulates the data as it is ingested and egressed to another application, report, or data stream. Data Streams allow for data to be used anywhere in the ecosystem, including on the edge devices, in the data center, in the cloud, and even in transit between the different types of infrastructure. Analytical reports can be generated parallel across multiple devices by combining data transforms through data streams.

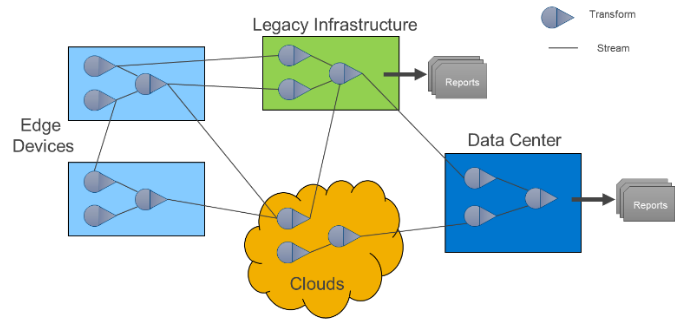


Figure 13: Data Streaming

Software Defined Infrastructure

Software Defined Infrastructure (SDI) contains the abstractions for private and public clouds. The SDI layer is a familiar standard interface for all cloud resources-virtual, container, and bare metal. The Software Defined Infrastructure Layer (SDI) is a middleware layer in the architecture. It primarily manages Infrastructure as a Service (IaaS) operations and management. SDI architectural elements are well-known and established in the industry, with commercial and open-source product offerings available (VMWare, OpenStack, Nutanix, etc.). The critical elements of an SDI layer are Orchestration and Control, Infrastructure elements (Storage, Network, Compute/Accelerators, and Security), and a Common Infrastructure API Gateway.

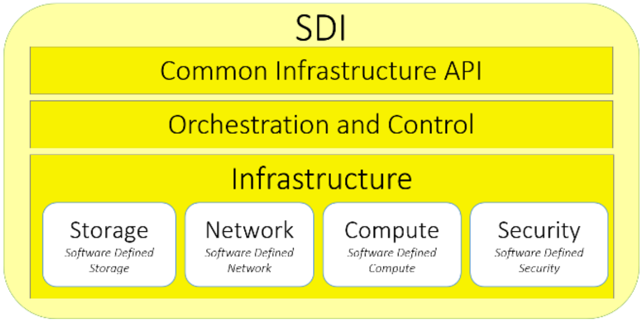


Figure 14: Software Defined Infrastructure

These key architectural elements are minimal viable features for a standard interface to IaaS solutions used in a Physical Layer. The ability to interact with a standard API interface regardless of the type of Cloud is essential for interoperability between private and public cloud offerings. To include Edge Devices into the ecosystem, the concept of a micro cloud was developed with the same minimal Common Infrastructure API.

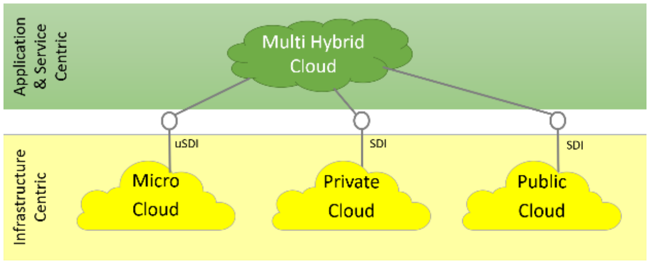


Figure 15: Multi-Hybrid Cloud

This concept extends the boundaries of the cloud to the edge and allows for the management of infrastructure and applications across a traditionally tricky border. The shared Common Infrastructure API allows the Multi-Cloud Orchestrator from the Service Management Layer to request infrastructure (Bare metal, Virtual, or containerized) to deploy complex applications across several cloud offerings.

Common Physical Layer

The Physical Layer (PL) contains abstractions allowing better management across an ecosystem inside the data center, cloud, and edge devices. These abstractions give the ability to manage highly variable hardware configurations by describing the common operating and taxonomy of the devices. This architectural layer has the goal of addressing the following characteristics:

- Common Taxonomy
- Portability and Interoperability
- Security and Root of Trust
- Common Management Control Plane
- Performance Optimization
- Stability and Reliability
- Flexibility and Agility

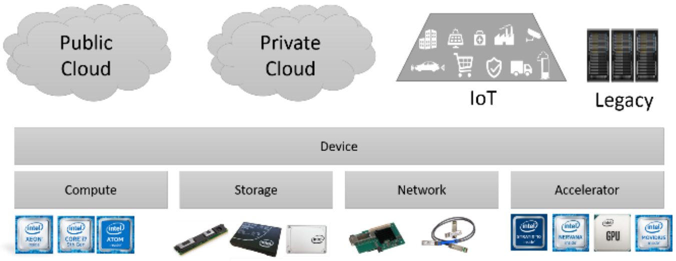


Figure 16: Physical Layer Representation

The PL sits at the bottom of the Architectural stack but interacts with all other layers and aspects. It relies on Security and Identity aspects to establish the hardware root of trust, identity, and data encryption at the lowest levels. Figure 16 shows the abstract layers established across Edge, Legacy, Data Center, and Cloud physical resources.

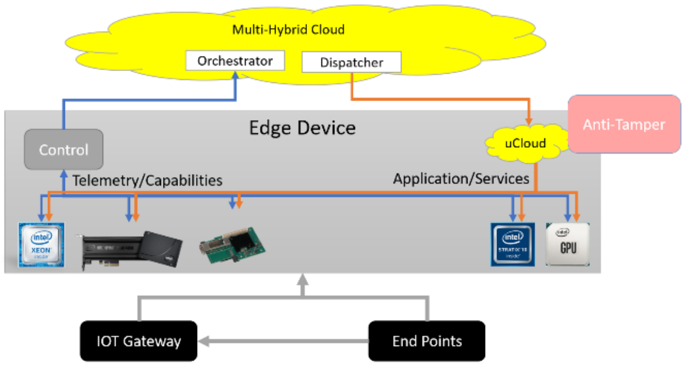


Figure 17: Edge Device

The critical element in this layer is called a Device. A Device contains one or more hardware elements, including processors, memory, accelerators, storage, and network capabilities. Each Device has a snapshot of its capabilities, hardware, and currently available resources in a "Profile" abstraction. The Device has a simple interface for control and telemetry through the Device Profile, which allows the Software-Defined infrastructure layer (SDI) to deploy and provision applications and services to take advantage of the Device's specialized hardware.

With the explosion of edge computing and sensors, the complexity of managing the devices in conjunction with the cloud and the data center has dramatically expanded. Managing 10s to 10,000s of devices is overwhelming for IT operations engineers, and many management and control architectures cannot scale appropriately. For this reason, the enterprise architect has created the Aggregated Device that allows the grouping of devices into collections that can be managed and controlled more easily. Aggregated devices can contain devices or other aggregated devices, giving the ability to have infinite layers in the hierarchy of devices.

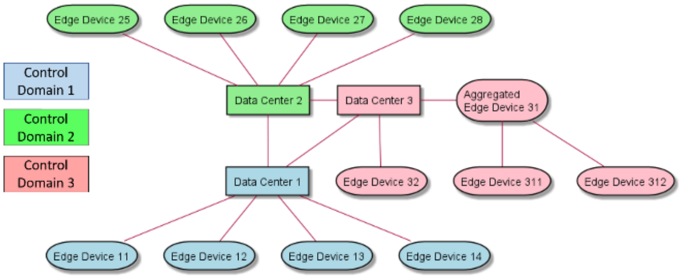


Figure 18: Physical Aggregate Layout

Often, organizations combine the physical management and the logical management of devices. Combining the physical and logical approaches is problematic as they create highly coupled, rigid, and fragile solutions that cannot adapt to change.

The enterprise architecture separates the physical and logical topologies, allowing for flexible architecture in business and operating environments. Additionally, the two topologies give the flexibility to establish an authentic edge to cloud architectures, including setting up a cloud that spans resources in prem data centers, public clouds, and edge devices. It allows scheduling and managing applications and services across traditional boundaries.

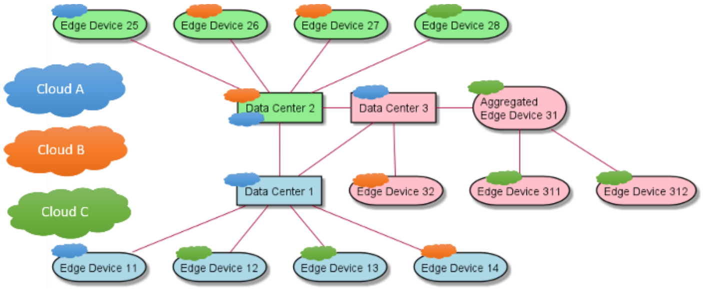


Figure 19: Logical and Physical Separation

Figure 19: Logical and Physical Separation This figure shows three clouds that share devices and span the control topology established for optimized IT operations. This flexibility allows clouds (logical devices) to adapt to changing environments. These changes can include cyber threats, physical disasters, partial connectivity of edge devices, or even someone tripping over a network connection in the data center.

Identity and Security

The Identity provides identity through all layers of the enterprise architecture. This aspect is responsible for the trusted identity of users, devices in the data center, the cloud, the edge, services and applications, and data. Having a standard identity management systemis critical to having consistency in the system. This identity must be trusted so that the data, applications, and hardware can deliver solutions that can be used confidently.

Identity has critical sub-systems that help manage identity: Access, Authorization, Authentication, and Key Management. Authenticating an entity in the system is the first step in identity management. Keys are used to certify and attest to the authentication of an entity, human or machine. Once an entity is authenticated, it is given authority to access other resources in the system. By providing identity to every element in the design, mishaps in security can be mitigated and better controlled. Given the new focus on Zero Trust Architectures, we constantly re-verify identity, and authorization is revoked if credentials do not match those expectations.

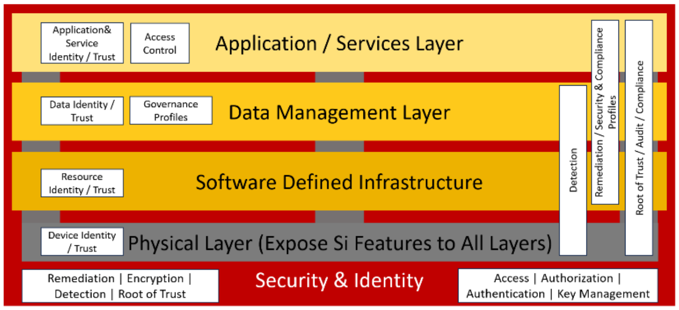

Application& Service Identity / Trust
Access Control
Application / Services Layer
Data Identity / Trust
Governance Profiles
Data Management Layer
Resource Identity / Trust
Software Defined Infrastructure
Device Identity / Trust
Physical Layer (Expose Si Features to All Layers)
Detection
Remediation / Security & Compliance Profiles
Root of Trust / Audit / Compliance
Remediation | Encryption | Detection | Root of Trust
Security & Identity
Access | Authorization | Authentication | Key Management

Figure 20: Identity and Security

Security

Security contains security tools and subsystems used throughout the architecture. It is used in every layer and was developed to provide a standard mechanism for performing common security use cases, such as encryption, detection, remediation, and root of trust.

Conclusion

With the complexity and ever-changing ecosystem of technology, business process innovation, and operating environments, we have developed the GEAR to show how all the elements needed to deliver digital solutions come together in a consistent and manageable framework. This facilitates bringing new capabilities to existing architectures by clarifying their interactions and dependencies and enabling organizations to transform rationally and cost-effectively digitally.

[1] All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.
Intel technology's features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration.
No computer system can be secure.