

WHITEPAPER

# Routing-centric architecture for distributed AI infrastructure

Designing customer-specific routing domains across cloud, data center, and enterprise environments with Lumen Multi-Cloud Gateway

# Table of contents

Executive summary .....3

Architectural context .....4

    Service architecture .....5

    MCGW service constructs.....6

Operating the architecture with Lumen Connect .....9

Routing decision flow ..... 12

Routing policy and traffic control..... 14

MCGW and LCI architectural design roles ..... 16

    MCGW as a routing foundation ..... 17

    LCI as a cloud connectivity service ..... 17

    Independent and combined deployment models ..... 18

    Architectural comparison: native multi-cloud vs. centralized routing domains ..... 19

Distributed AI infrastructure reference architecture patterns..... 21

Deployment methodology ..... 25

Failure domain and resiliency considerations..... 29

Design considerations and best practices..... 33

Conclusion ..... 37

# Executive summary

Modern AI infrastructure increasingly operates across multiple infrastructure domains, including cloud providers, data centers, AI campuses, regional compute environments, research environments, and operational platforms. As these environments expand, organizations face the challenge of maintaining consistent routing, policy control, operational visibility, and connectivity management across a distributed infrastructure footprint. This paper examines how Lumen® Multi-Cloud Gateway (MCGW) and Lumen® Cloud Interconnect (LCI) support these requirements through centralized routing, private connectivity, and API-driven operations.

MCGW provides a centralized routing and policy framework that enables cloud, data center, and enterprise environments to participate in a common routing domain. LCI provides private cloud connectivity that can be deployed independently or alongside MCGW. Together, these services help infrastructure teams simplify route management, maintain operational consistency, and extend connectivity across multi-domain environments.

This paper is written for infrastructure architects, cloud network architects, platform engineering teams, and technical evaluators responsible for designing and operating large-scale distributed environments. It assumes a working knowledge of cloud networking, BGP routing, private cloud connectivity, distributed systems, and modern infrastructure operations. The objective is to examine routing architecture, connectivity models, operational workflows, and deployment patterns that can support increasingly distributed AI infrastructure environments.

The paper examines the architectural principles, operational model, and deployment approaches associated with MCGW and LCI. It explores how centralized routing domains, private connectivity services, policy-driven controls, and API-enabled operations can be combined to support distributed infrastructure environments at scale.

To illustrate these concepts, the paper presents a series of reference deployment patterns, including cloud-to-cloud connectivity, site-to-cloud connectivity, distributed infrastructure connectivity, and combined MCGW and LCI architectures. These examples demonstrate how routing domains, connectivity services, and operational workflows can be assembled to support real-world distributed infrastructure environments.

The operational challenge facing many infrastructure teams is not simply adding more connectivity. It is establishing a consistent way to manage routing, policy, visibility, and operations as infrastructure expands across cloud, data center, enterprise, and AI environments. By combining centralized routing domains, policy-driven controls, private connectivity options, and API-enabled operations, MCGW and LCI provide a framework for designing and operating distributed infrastructure architectures at scale.

## Distributed AI infrastructure connectivity pattern

Figure 1. Distributed AI Infrastructure Connectivity Pattern

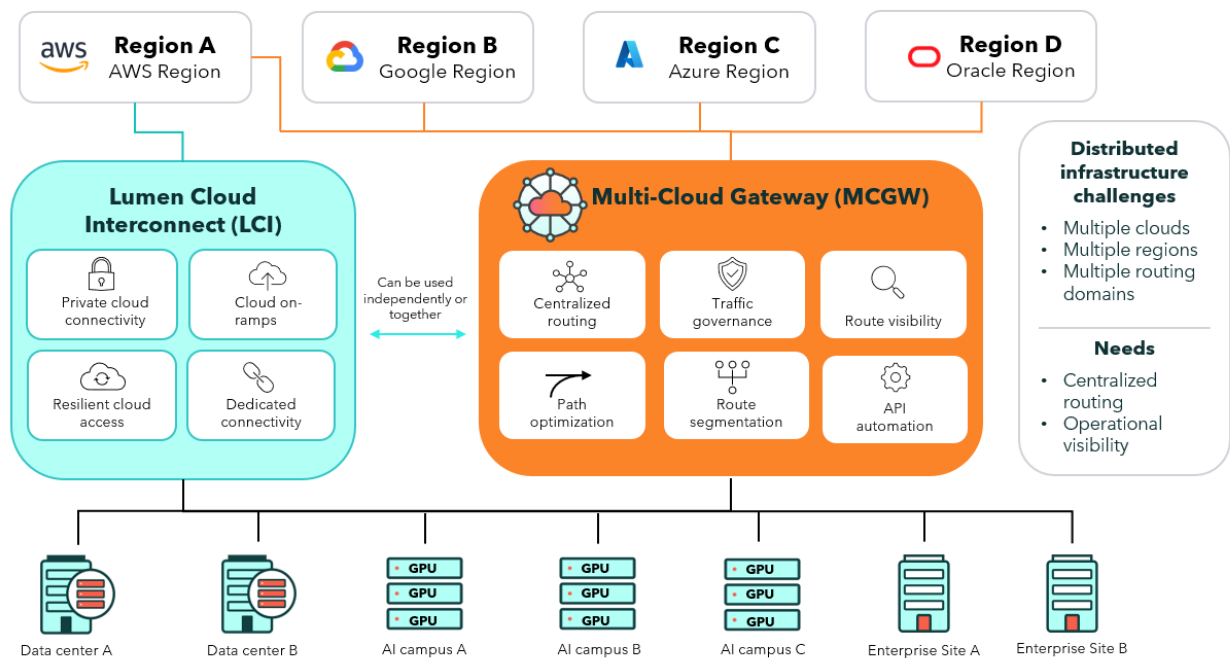


Figure 1. Distributed AI infrastructure connectivity pattern

MCGW provides a centralized routing and policy foundation across cloud, data center, AI campus and enterprise environments. LCI provides resilient private cloud connectivity that can be deployed independently or alongside MCGW to support distributed infrastructure operations.

The next section examines the architectural foundation of Multi-Cloud Gateway, including the routing-domain model, connectivity relationships, and service components used to support centralized routing across distributed infrastructure environments.

## Architectural context

The transition from isolated infrastructure environments to distributed architectures introduces new requirements for routing, policy management, and operational visibility. Rather than treating cloud, data center, AI campus, and enterprise environments as independent networking domains, many organizations are adopting architectures that provide a more consistent framework for connectivity and route governance across the entire infrastructure footprint.

Within this model, Multi-Cloud Gateway (MCGW) provides a customer-specific Layer 3 routing domain in which connected environments can exchange routes, apply routing policies, and operate through a common management framework. Lumen Cloud Interconnect (LCI) provides resilient private cloud connectivity and can be deployed independently or alongside MCGW. Together, these services establish a flexible architecture in which private connectivity and centralized route control can be adopted separately or combined to support distributed infrastructure requirements. Figures 1 through 3 introduce this architectural foundation before the paper examines the underlying service model and operational workflows.

## Service architecture

### Overview

Multi-Cloud Gateway should be understood as a service architecture, not as a device architecture. It provides a customer-specific Layer 3 routing domain on the Lumen Connectivity Fabric, where private connectivity from enterprise sites, data centers, AI campus locations, and Cloud On-Ramps can be brought together under a flexible and centralized routing and policy control model.

The following figures establish the MCGW service model before the paper examines operational workflows, routing behavior, policy controls, architectural design patterns, deployment methodology, and resiliency considerations. Figure 2 shows the architecture at a service level: access endpoints connect into MCGW through private Layer 2 attachments, and MCGW provides the virtual routing domain that controls reachability across those attachments. Figure 3 then decomposes that architecture into the service constructs an architect must understand in order to design, provision, and operate the service.

In this model, the access endpoint, Layer 2 connection, Layer 3 handoff, and routing domain are intentionally separate constructs. A Fabric Port or Cloud On-Ramp represents where traffic attaches to the Lumen Connectivity Fabric. A Fabric Connection represents the billable Layer 2 connection between known endpoints. Ethernet Fabric Connect is the customer-facing private Layer 2 connectivity used to deliver that connection construct. The MCGW Interface is the virtual Layer 3 endpoint where IP addressing, BGP or static routing, and route policy are configured. The MCGW Instance is the customer-specific global Virtual Routing and Forwarding (VRF) construct where routing decisions and policy control are applied.

This separation of functions is important for technical design. It allows architects to reason clearly about where traffic enters the service, where Layer 2 attachment ends, where Layer 3 routing begins, and where routing policy is enforced. It also supports a repeatable operating model: endpoints can be attached through Fabric Connections, routing can be configured at MCGW Interfaces, and reachability can be managed centrally within the MCGW Instance. Lumen abstracts the underlying network implementation while orchestrating the service constructs, attachment points, and connection resources needed to deliver the MCGW service.

Figure 2 introduces the MCGW service architecture from the perspective of an enterprise or partner architect evaluating how private connectivity is composed across sites, data centers, and public cloud providers.

The figure separates the architecture into three functional layers:

- **Access attachment:** Fabric Ports and Cloud On-Ramps provide the attachment points into the Lumen Connectivity Fabric.
- **Private Layer 2 attachment:** Ethernet Fabric Connect provides private Layer 2 connectivity through Fabric Connections, including Virtual Point-to-point (VP2P) and Hosted Cloud Connect patterns.
- **Layer 3 routing and policy:** MCGW Interfaces provide the Layer 3 handoff into the customer-specific MCGW routing domain.

## Multi-Cloud Gateway service architecture and functional layers

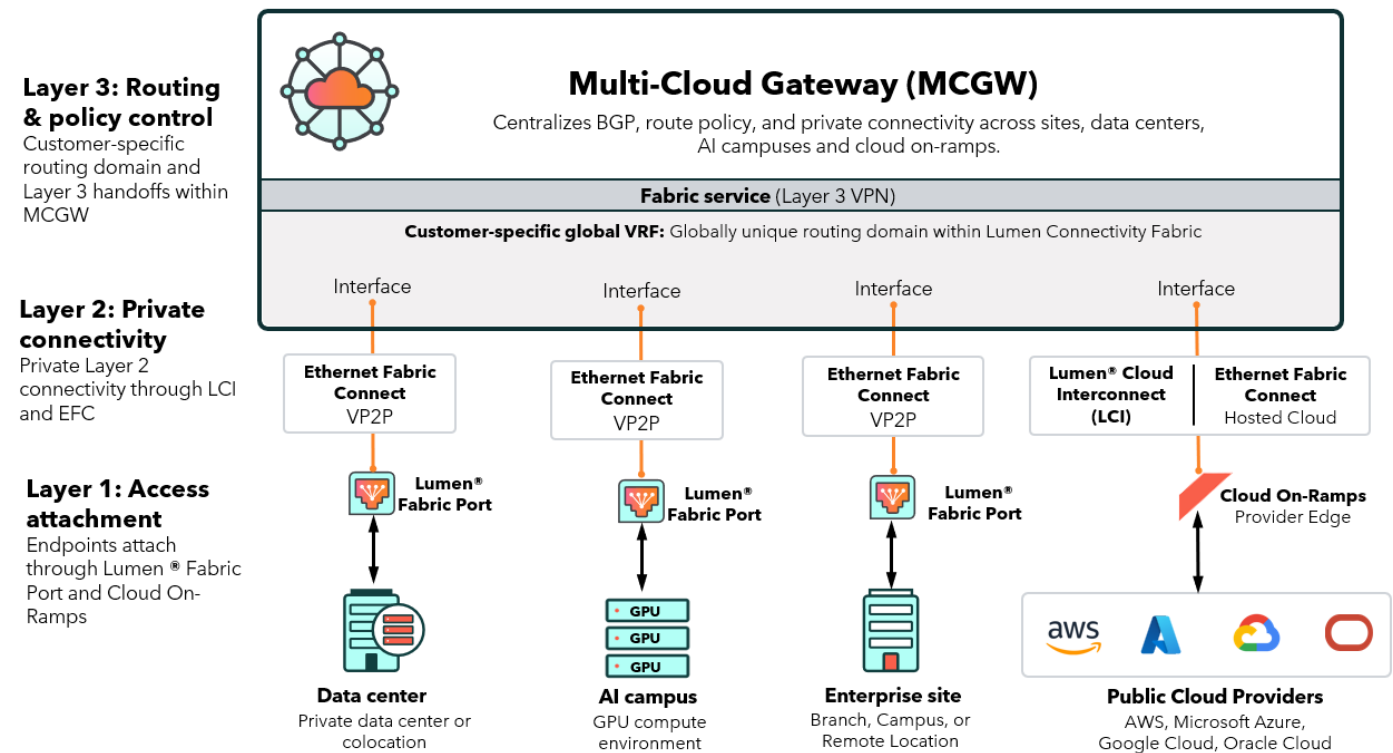


**Figure 2. Multi-Cloud Gateway service architecture and functional layers**

Multi-Cloud Gateway (MCGW) centralizes Layer 3 routing, BGP route exchange, and route policy across private connectivity attachments from enterprise sites, data centers, and cloud on-ramps. Fabric Ports and Cloud On-Ramps act as access endpoints, Ethernet Fabric Connect provides private Layer 2 attachment into MCGW, and MCGW Interfaces provide the virtual Layer 3 handoff into the customer-specific global VRF.

### Key architectural elements

- Customer-specific Global VRF
- Virtual Layer 3 handoffs through MCGW Interfaces
- Policy-based routing controls
- Centralized route exchange and route filtering
- Private connectivity from cloud, data center, and site endpoints
- Lumen Connect<sup>SM</sup> and API-driven operations

Figure 3 is intended to remove ambiguity between terms that can otherwise sound interchangeable. In practice, Fabric Port, Fabric Connection, Ethernet Fabric Connect, MCGW Interface, Fabric Service, and MCGW Instance each represent a different part of the service model. For design and operations, those distinctions matter.

### MCGW service constructs

The service architecture view shows how MCGW is assembled at a functional level. The service object model below maps those functions to customer-visible constructs, layer boundaries, and attachment points. The upper portion of the model represents Layer 3 routing and policy control. The lower portion represents Layer 2 attachment and access/provider endpoints.

## MCGW service object model and boundaries

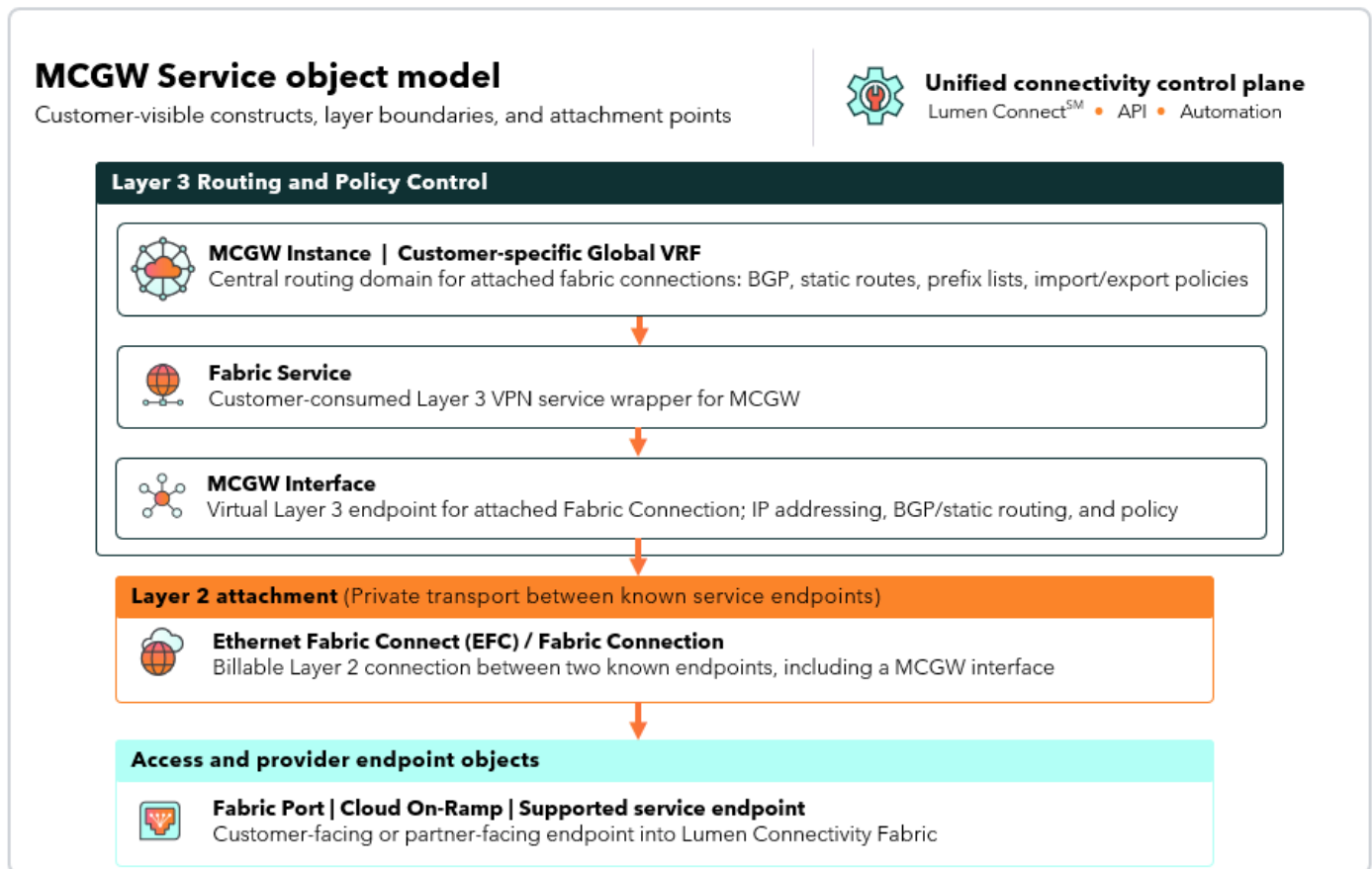


**Figure 3. MCGW service object model and boundaries**

The MCGW service model separates routing, interface, connection, and endpoint constructs. The MCGW Instance represents the customer-specific global VRF. The Fabric Service represents the service wrapper associated with the MCGW Layer 3 routing domain. MCGW Interfaces are virtual Layer 3 endpoints for attached Fabric Connections. Fabric Connections represent the billable Layer 2 connection objects, while Ethernet Fabric Connect is the customer-facing private Layer 2 service delivered using those constructs.

The following table defines each construct in the model using terms relevant to technical design, provisioning, and operations. The intent is not to provide a generic glossary, but to clarify where each construct sits in the architecture and which attributes matter when designing or operating an MCGW-based service.

## MCGW service object reference

| Construct                     | Layer / role                                                                                 | Key attributes                                                               |
|-------------------------------|----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| MCGW Instance                 | Customer-specific L3 routing domain and global VRF                                           | Gateway ID, name, ASN, bandwidth tier, route table                           |
| Fabric Service                | Service wrapper for the MCGW Layer 3 VPN construct                                           | Service type, service status                                                 |
| MCGW Interface                | Virtual L3 endpoint for an attached Fabric Connection                                        | IP addressing, BGP/static routing, BFD, prefix lists, policy                 |
| Fabric Connection             | System/service object representing a billable Layer 2 connection between two known endpoints | Connection ID, A/Z endpoints, bandwidth, CoS, VLAN, service type             |
| Ethernet Fabric Connect (EFC) | Customer-facing name for the private Layer 2 service that delivers the Fabric Connection     | VP2P / E-Service, Hosted Cloud Connect, private Layer 2 attachment into MCGW |
| Fabric Port                   | Physical provider-facing access attachment                                                   | Port speed, tagging, port type, admin/oper status                            |
| Cloud On-Ramp                 | Provider-facing attachment                                                                   | CSP, on-ramp, VLAN, hosted/dedicated model                                   |

**Table 1. MCGW service object reference**

This matrix defines the primary service constructs used in the MCGW architecture and identifies their role, layer, and key attributes.

These constructs are exposed to customers and partners at different levels of the service model. Some define how an endpoint attaches to Lumen. Others define how private Layer 2 connectivity is created, where routing begins, and how policy is applied. Lumen abstracts the underlying network implementation so customers and partners can work with service constructs rather than device-level implementation.

### Boundary notes

- MCGW is the customer-specific Layer 3 routing domain, not a physical appliance or customer-managed router.
- MCGW Interfaces are virtual Layer 3 endpoints, not physical ports. IP addressing, routing, and policy are configured at this boundary.
- Lumen abstracts the underlying network implementation while orchestrating the service constructs, attachment points, and connection resources required to establish and maintain private connectivity across the Lumen Connectivity Fabric.

For architects and engineering teams evaluating private multi-cloud connectivity, the value of this model is not simply that MCGW connects sites and clouds. The value is that the service separates access, attachment, routing handoff, and policy control into distinct constructs that can be designed, provisioned, and operated consistently. That separation gives architects a clearer way to build private multi-cloud connectivity, apply routing policy centrally, and expand connectivity without redesigning every cloud or site connection as a standalone network construct.

The following sections build on this foundation by showing how traffic flows, how routing is controlled, and how these constructs translate into deployable architectures.

# Operating the architecture with Lumen Connect

## API-driven operating model

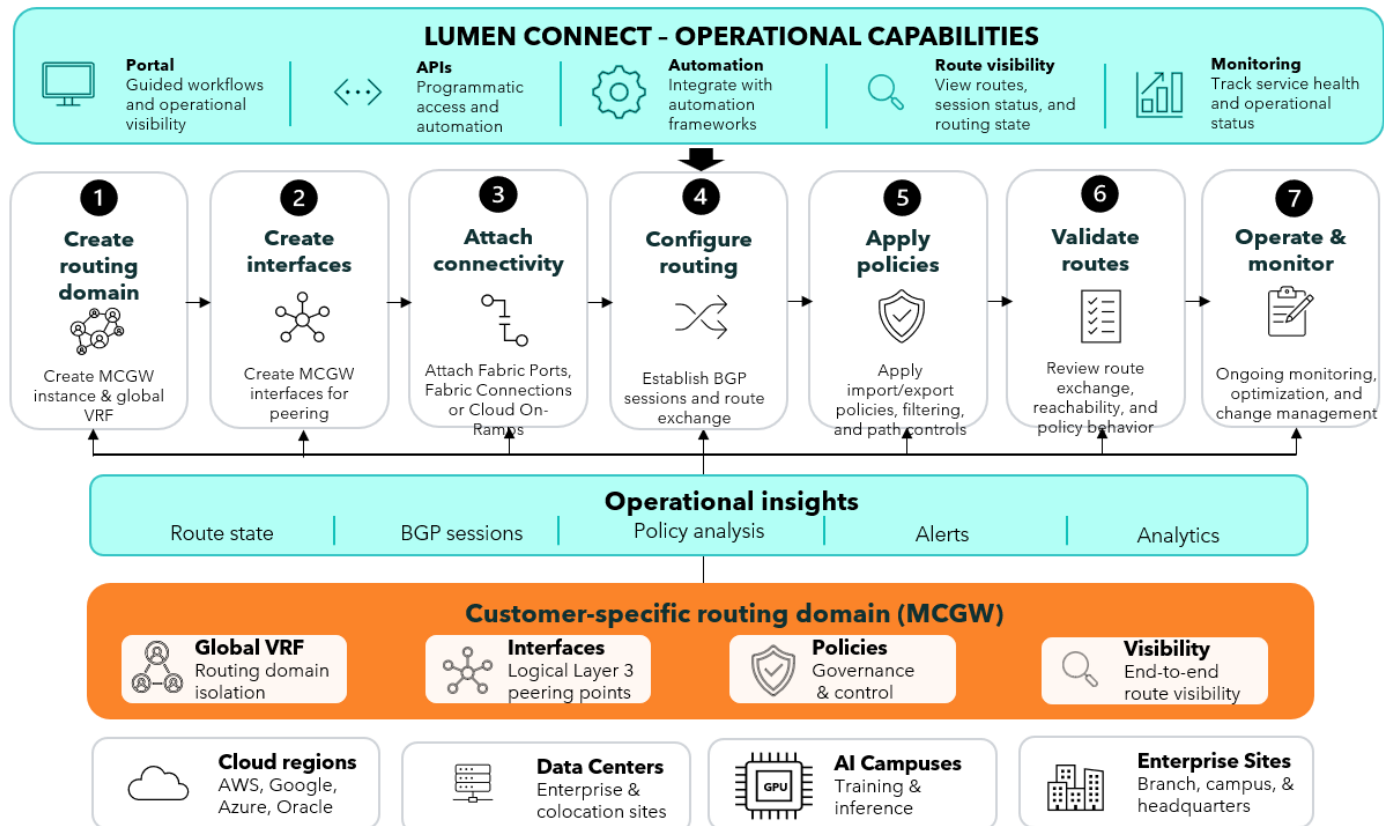


**Figure 4. API-driven operating model**

Lumen Connect<sup>SM</sup> provides the operational control capabilities for Multi-Cloud Gateway services. Through portal and API workflows, architects can create gateway instances, add interfaces, attach Fabric Connections, configure routing and policy controls, validate learned and advertised routes, and monitor ongoing service operations across distributed infrastructure environments.

## Overview

While the previous sections focused on the architecture and service objects that make up a Multi-Cloud Gateway deployment, operating a routing domain requires an additional set of capabilities. Infrastructure teams must be able to provision connectivity, define routing relationships, manage policy, validate route and session state, monitor operational status, and adjust network behavior as requirements evolve.

Lumen Connect<sup>SM</sup> serves as the operational interface for these activities. Through portal-based workflows and associated APIs, customers can deploy and manage the Lumen-side lifecycle of Multi-Cloud Gateway services. Rather than managing routing-domain operations through disconnected workflows, administrators use a common operational framework that supports the lifecycle of customer-specific MCGW routing domains.

## Operational lifecycle

At a high level, the lifecycle of a Multi-Cloud Gateway deployment follows a repeatable sequence:

Create MCGW Instance



Create Interfaces



Attach Connectivity



Establish Routing



Apply Policies



Validate Routes



Monitor and Operate

This model separates deployment activities from routing functions while providing a structured process for building and maintaining customer-specific routing domains.

### Creating the routing domain

The operational lifecycle begins with creation of an MCGW Instance, which represents the customer-specific routing construct that hosts the Global VRF and associated routing relationships. Once deployed, the routing domain becomes the centralized location through which attached infrastructure environments exchange route information. At this stage, the routing domain exists independently from any attached cloud, data center, or enterprise environment. Additional service objects must be introduced before route exchange can occur.

### Establishing connectivity

After the routing domain is created, access endpoints such as Fabric Ports and Cloud On-Ramps can be attached to MCGW Interfaces through Fabric Connections delivered across the Lumen Connectivity Fabric. This architecture intentionally separates transport responsibilities from routing responsibilities.

From an operational perspective, connectivity deployment focuses on:

- Creating or selecting attachment points
- Associating Layer 2 transport paths
- Mapping service objects to MCGW Interfaces
- Preparing routing interfaces for route exchange

At this stage, connectivity may be available, but routing policies and route relationships still determine how traffic is learned, accepted, preferred, and advertised.

## Defining routing relationships

Once interfaces are connected, routing relationships can be established. MCGW Interfaces act as virtual Layer 3 attachment points into the customer routing domain. Routing relationships may include BGP sessions, static routes, import policies, export policies, prefix lists, and route filtering controls.

The objective is to define how route information enters, moves through, and exits the customer-specific routing domain while maintaining consistent policy enforcement across attached environments.

## Policy and route management

Operational control extends beyond connectivity establishment. Routing behavior is governed through policy constructs that influence route admission, route visibility, route advertisement, and path selection. These controls enable architects to align routing behavior with the requirements of specific cloud, data center, or enterprise environments.

Examples of policy-controlled functions include:

- Route filtering
- Route advertisement
- Prefix acceptance
- Route preference and best path selection
- Route visibility

Subsequent sections examine these routing controls in greater detail.

## Visibility and operational monitoring

As infrastructure environments expand, operational visibility becomes increasingly important. Lumen Connect provides a centralized operational framework through which administrators can view deployed service constructs, review configuration state, validate route and session status, and monitor service health and operational status.

Rather than requiring operational teams to reconcile disconnected service views, MCGW operations can be evaluated through a common management framework aligned with the customer-specific routing domain.

## API-driven operations

Many large-scale infrastructure organizations prefer to manage networking services through automation frameworks rather than manual configuration processes. To support these operating models, Lumen Connect<sup>SM</sup> and associated APIs support workflows that can be integrated into broader infrastructure provisioning and operational processes.

This allows routing-domain deployment, connectivity provisioning, and service lifecycle management activities to be incorporated into standardized operational workflows, while still supporting interactive administration through the Lumen Connect<sup>SM</sup> portal.

The result is an operational model that supports both interactive administration through the portal and automated management through programmatic interfaces.

## Architectural takeaway

The Multi-Cloud Gateway operating model is designed around the lifecycle of a customer-specific routing domain. Lumen Connect provides the operational interface through which architects create service objects, establish connectivity, configure routing relationships, apply policy controls, validate route exchange, and monitor ongoing operations.

Together, these capabilities provide a common operational framework for managing distributed infrastructure environments, helping teams operate routing, policy, and visibility consistently as cloud, data center, and enterprise attachments expand.

# Routing decision flow

## How routes are learned, advertised, preferred, or blocked

Once enterprise sites, data centers, and Cloud On-Ramps are attached to MCGW, the architecture shifts from connectivity to control. The design question is no longer simply whether endpoints can connect, but how reachability is governed across those endpoints. Architects need to determine which routes are allowed into the routing domain, which routes are advertised out, which paths should be preferred, and which peers, sessions, or paths should be considered available for route exchange.

MCGW addresses this through a routing control model built around the customer-specific global VRF and the MCGW Interfaces attached to it. Routes can be learned dynamically through BGP or defined through static routing. Prefix lists and policy lists provide control over route eligibility and route handling. Accepted routes are evaluated within the MCGW routing domain, while export policy determines what route information is advertised to connected sites, data centers, cloud on-ramps, or service endpoints. Session and operational controls such as MD5 authentication where supported, BFD, BGP session state, and route visibility help protect session establishment, detect failures, track peer state, and validate route behavior.

For architects and engineering teams designing private multi-cloud connectivity, this model is important because routing intent can be expressed centrally rather than recreated independently for every site-to-cloud or cloud-to-cloud connection. A cloud on-ramp, enterprise site, or data center can attach through the same service model, while route policy determines which destinations are reachable, which paths are used, and what route information is exposed to other connected endpoints. This separation helps reduce design complexity and provides a more repeatable way to scale private connectivity across multiple environments.

The following figure presents the routing model as a route lifecycle. The first row shows how routes enter the MCGW routing domain: routes are sourced, filtered, governed by policy, and admitted into the MCGW Route Table. The second row shows how accepted routes are evaluated and advertised: MCGW applies route evaluation and export controls before selected routes are made visible to attached targets. The session and operational control layer supports the lifecycle by protecting BGP session establishment, detecting failures, tracking peer state, and exposing route state for operational validation.

## MCGW route lifecycle and control model

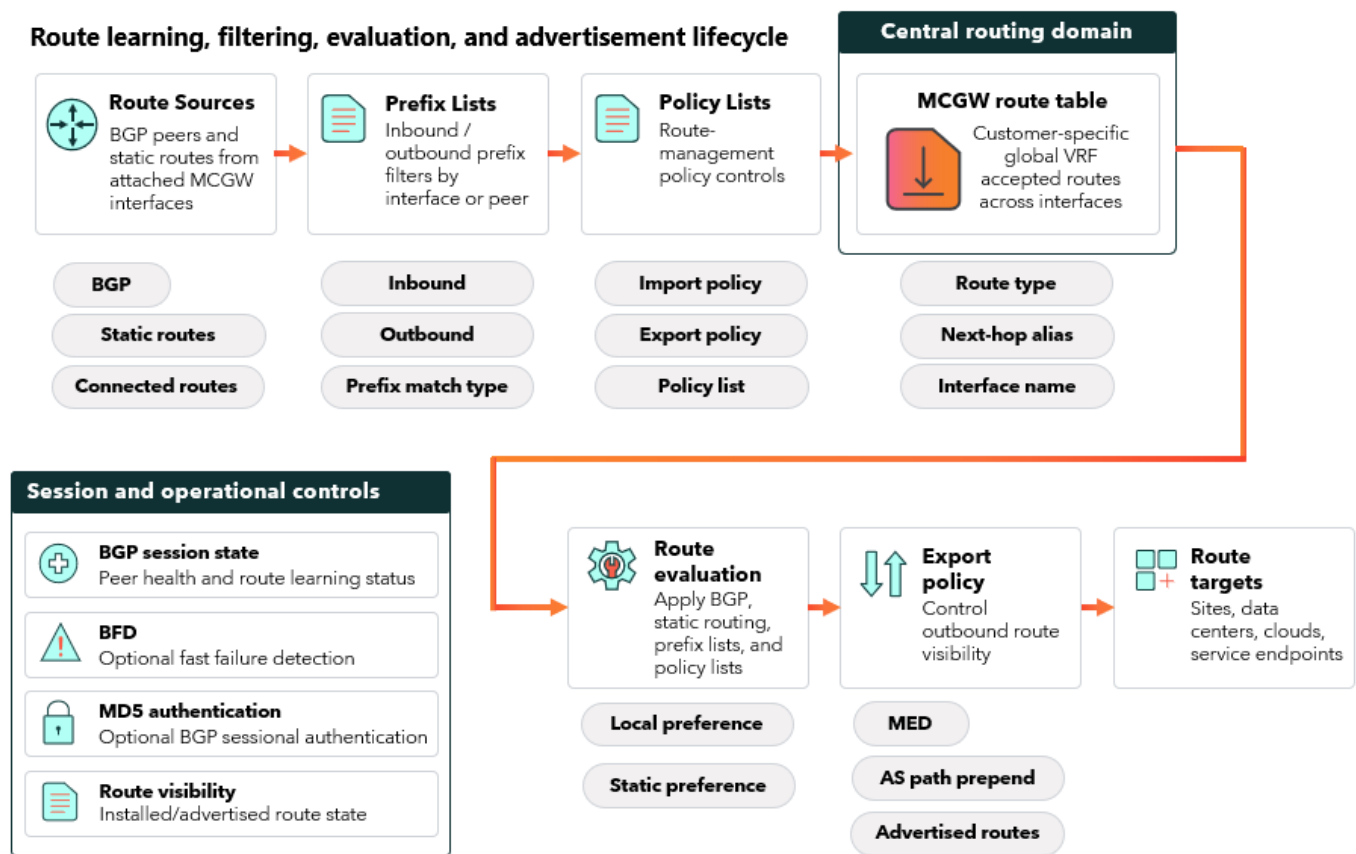


**Figure 5. MCGW route lifecycle and control model**

This diagram illustrates the route lifecycle through MCGW: how routes are learned from attached endpoints, filtered through prefix and policy controls, admitted into the customer-specific routing domain, evaluated for path selection, advertised to connected targets, and validated through session and operational controls.

The value of this model is that routing control is applied at distinct points in the service lifecycle. Route sources define how reachability enters the routing domain. Prefix lists and policy lists determine which route information is allowed or governed. Route evaluation determines how accepted routes are used inside the customer-specific global VRF. Export policy controls what connected endpoints learn from MCGW. Operational controls provide the status, integrity, and failure-detection signals needed to validate that routing behavior matches design intent. This distinction matters in multi-cloud designs because not every routing control applies at the same point in the route lifecycle.

Together, these controls give architects a structured way to express routing intent across heterogeneous endpoints. Instead of designing each connectivity path as a separate routing problem, MCGW provides a common routing domain where route learning, filtering, selection, advertisement, and operational visibility can be managed consistently as environments expand across cloud providers, data centers, enterprise sites, and service endpoints.

# Routing policy and traffic control

## Overview

The routing lifecycle described in the previous section determines how routes are learned, evaluated, selected, and advertised throughout the customer-specific Global VRF. While the route lifecycle provides the framework, routing policies and route-control mechanisms determine how individual routing decisions are made.

Multi-Cloud Gateway supports policy-driven route management through prefix controls, route filtering mechanisms, supported BGP attributes, and operational safeguards. These controls allow architects to influence route admission, route selection, route advertisement, failure detection, and routing-domain behavior across connected infrastructure environments.

## Routing policy and control mechanisms

| Control            | Primary purpose                                            | Route lifecycle stage               | Typical outcome                                |
|--------------------|------------------------------------------------------------|-------------------------------------|------------------------------------------------|
| Prefix lists       | Route filtering                                            | Route admission                     | Permit or deny specific prefixes               |
| Import policies    | Route admission control                                    | Import evaluation                   | Control which routes enter the routing domain  |
| Export policies    | Route advertisement control                                | Export evaluation                   | Control which routes are advertised            |
| Local preference   | Route preference within the routing domain                 | Best path selection                 | Prefer one available path over another         |
| MED                | Peer path influence                                        | Best path selection / advertisement | Influence path selection between routing peers |
| AS path prepend    | Route advertisement influence                              | Route advertisement                 | Make selected advertised paths less preferred  |
| BGP communities    | Route classification and policy signaling, where supported | Import / Export evaluation          | Apply consistent routing policy signals        |
| Static routes      | Explicit route definition                                  | Route installation                  | Create deterministic routing behavior          |
| BFD                | Failure detection                                          | Runtime operations                  | Accelerate path-failure detection              |
| MD5 authentication | Session protection                                         | BGP establishment                   | Help validate authorized routing relationships |

**Table 2. Routing policy and control mechanisms**

Routing behavior within MCGW is governed through policy-driven controls that influence route admission, route selection, route advertisement, and operational behavior throughout the route lifecycle. Availability of specific BGP attributes may depend on supported service capabilities.

## Route admission controls

Route admission controls determine which prefixes are permitted to enter the customer-specific Global VRF. Prefix lists and import policies evaluate candidate routes before those routes are accepted into the routing domain. These controls help reduce route leakage, limit unnecessary propagation, and align route visibility with the intended architecture.

Rather than accepting all advertised routes, policy controls allow architects to explicitly define which routing information participates in the routing domain.

## Route selection controls

When multiple candidate routes exist for the same destination, route-selection controls can influence forwarding behavior. Routing attributes such as Local Preference, MED, AS path prepend, and community-based policies where supported allow architects to express routing intent while maintaining centralized control through the customer-specific routing domain.

These controls do not create connectivity. They influence how available routes are evaluated, preferred, and advertised so traffic can follow the intended routing design where supported.

## Route advertisement controls

Export policies govern how routing information leaves the routing domain. By controlling route advertisement behavior, architects can determine which environments learn specific route information, which paths are visible to specific peers, and how routing information is distributed across attached cloud, data center, and enterprise environments.

This capability becomes increasingly important as infrastructure expands across multiple administrative and operational domains.

## Route security and stability controls

Routing policy extends beyond route selection and advertisement. Operational safeguards such as MD5 authentication and Bidirectional Forwarding Detection (BFD) support routing-session integrity and faster failure detection. MD5 authentication helps protect BGP session establishment where supported, while BFD can help detect path or session failures more quickly than standard routing timers.

These controls contribute to routing-domain stability while providing operational visibility into routing health and connectivity status.

## Routing and operations platform capabilities

| Capability                  | Function                                                     |
|-----------------------------|--------------------------------------------------------------|
| BGP                         | Dynamic route exchange between routing domains               |
| Static routes               | Explicit route definition and deterministic forwarding       |
| Prefix lists                | Prefix filtering and route admission control                 |
| Import policies             | Route acceptance and policy evaluation                       |
| Export policies             | Route advertisement control                                  |
| Lumen Connect <sup>SM</sup> | Operational management platform                              |
| APIs                        | Programmatic provisioning and automation                     |
| Monitoring                  | Visibility into routing status and service health            |
| Validation                  | Configuration, session, and route-state validation workflows |

**Table 3. Routing and operations platform capabilities**

MCGW combines routing protocols, route-policy constructs, operational tooling, APIs, and visibility capabilities to support lifecycle management of customer-specific routing domains.

### **Architectural takeaway**

Multi-Cloud Gateway is designed around policy-driven routing control rather than simple route transport. Prefix controls, routing policies, route-selection attributes, and operational safeguards collectively govern how route information enters, moves through, and exits the customer-specific Global VRF.

By centralizing these controls within a common routing domain, architects can apply consistent routing behavior across cloud providers, data centers, campuses, and enterprise environments while maintaining operational visibility and policy governance.

## **MCGW and LCI architectural design roles**

### **Overview**

Multi-Cloud Gateway (MCGW) and Lumen Cloud Interconnect (LCI) are complementary services that address different architectural requirements.

MCGW is the routing and policy foundation. It provides a customer-specific Layer 3 routing domain where connected environments can exchange routes, apply prefix and routing policies, and operate through Lumen Connect and associated APIs.

LCI is a private cloud connectivity service. It provides managed connectivity into supported cloud environments and can be used independently for private cloud access or alongside MCGW when the architecture requires both private cloud connectivity and centralized routing control.

The distinction matters because not every cloud connectivity requirement requires a centralized routing domain, and not every multi-cloud routing requirement is solved by a cloud access circuit alone.

## MCGW and LCI design roles

| Evaluation area               | Multi-Cloud Gateway (MCGW)                                                                    | Lumen Cloud Interconnect (LCI)                                                |
|-------------------------------|-----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Primary purpose               | Centralized routing and policy foundation                                                     | Private cloud connectivity and cloud access                                   |
| Core function                 | Route exchange and routing-domain management                                                  | Managed connectivity between defined endpoints                                |
| Routing domain                | Customer-specific Global VRF                                                                  | Abstracted connectivity model rather than a customer-managed routing domain   |
| Route control                 | Import/export policies, prefix controls, route management                                     | Connectivity-focused; routing details may be abstracted depending on use case |
| Best fit                      | Multiple sites, clouds, data centers, or infrastructure domains requiring centralized routing | Dedicated private connectivity into supported cloud environments              |
| Customer visibility           | Routing behavior, policy enforcement, route lifecycle                                         | Connectivity path, service status, and connection objects                     |
| Operational focus             | Routing and policy administration                                                             | Cloud attachment and connectivity management                                  |
| Typical architecture          | Cloud-to-cloud, cloud-to-data-center, multi-endpoint routing environments                     | Site-to-cloud or data-center-to-cloud connectivity                            |
| Relationship to other service | Can operate independently or with LCI                                                         | Can operate independently or with MCGW                                        |
| Architectural outcome         | Shared routing domain with centralized control                                                | Resilient private cloud connectivity                                          |

**Table 4. MCGW and LCI design roles**

Although MCGW and LCI are often deployed together, they address different architectural requirements. MCGW provides centralized route control, while LCI provides private cloud connectivity.

### MCGW as a routing foundation

MCGW is best viewed as the routing and policy foundation for multi-endpoint private connectivity.

The customer-specific Global VRF provides a centralized routing domain through which connected environments exchange route information. MCGW Interfaces, routing policies, route filtering controls, and route lifecycle management collectively establish a framework for controlling traffic movement across cloud, data center, and enterprise environments.

MCGW aligns well with architectures that require:

- Multi-cloud route exchange
- Cloud-to-cloud routing
- Site-to-cloud routing
- Data-center integration
- Centralized route governance
- Policy-driven routing behavior

### LCI as a cloud connectivity service

LCI is best viewed as a cloud connectivity service. Rather than focusing on customer-managed route-domain construction, LCI focuses on managed private connectivity into supported cloud environments.

These private cloud access models can simplify cloud attachment while providing an alternative to Internet-based connectivity approaches.

LCI aligns well with architectures that require:

- Private cloud access
- Dedicated cloud connectivity
- Single-cloud connectivity
- Direct connectivity into cloud provider environments

## Independent and combined deployment models

An important architectural consideration is that MCGW and LCI can be adopted independently or combined when both routing-domain control and private cloud connectivity are required.

An organization may deploy:

- **MCGW without LCI**

When the primary objective is route control, route exchange, centralized policy, and routing-domain management across multiple environments.

- **LCI without MCGW**

When the primary objective is private cloud connectivity rather than centralized routing-domain management.

This distinction helps prevent a common misconception that MCGW and LCI are inherently dependent on one another. The services address different requirements and can be adopted independently.

- **Combined deployment model**

Some organizations require both centralized routing control and private cloud connectivity. In these architectures:

- LCI can provide private cloud attachment.
- MCGW can provide route exchange and policy control.
- The customer-specific Global VRF can provide a shared routing domain for attached environments.

This approach allows cloud, site, and data-center environments to participate in a centralized routing architecture while leveraging private cloud connectivity paths where appropriate.

## When to use Hosted Cloud EFC vs LCI

While both Hosted Cloud EFC and LCI enable cloud attachment into an MCGW routing domain, LCI leverages provider-specific cloud interconnect architectures such as AWS Interconnect - Last Mile. These architectures provide built-in redundancy, automated failover, and abstraction of cloud-side connectivity and routing constructs that would otherwise require customer design and operational management. Hosted Cloud EFC provides cloud attachment into the routing domain, while LCI extends that attachment model with cloud-provider-specific resiliency architectures and simplified interconnect-side operations.

## Architectural takeaway

MCGW and LCI address different but complementary architectural requirements. MCGW provides the customer-specific routing domain, policy enforcement, and route lifecycle management needed for multi-endpoint private connectivity. LCI provides managed private cloud connectivity and cloud access consumption. Organizations can deploy either service independently or combine both when the architecture requires private cloud connectivity and centralized routing control within the same distributed infrastructure model.

## Architectural comparison: native multi-cloud vs. centralized routing domains

Before evaluating deployment patterns, it is useful to compare how routing, policy enforcement, and operational management are typically implemented in native multi-cloud environments versus a centralized routing-domain architecture. Many organizations begin with cloud-native networking constructs that evolve independently within each cloud provider, resulting in multiple routing domains, policy frameworks, operational workflows, and management tools.

As infrastructure expands across clouds, data centers, AI environments, and enterprise locations, maintaining consistent routing behavior and operational visibility can become increasingly complex. Multi-Cloud Gateway addresses this challenge by introducing a customer-specific Global VRF that centralizes route exchange, policy enforcement, routing visibility, and operational control across connected environments.

Figure 6 illustrates this common design challenge and contrasts it with an MCGW routing-domain model, where connected environments participate in a customer-specific Global VRF and common routing policy framework. Table 5 then expands on this comparison by highlighting the key architectural, operational, and policy-management differences between native multi-cloud networking approaches and a centralized routing-domain model.

## Native multicloud architecture vs. MCGW routing domain architecture

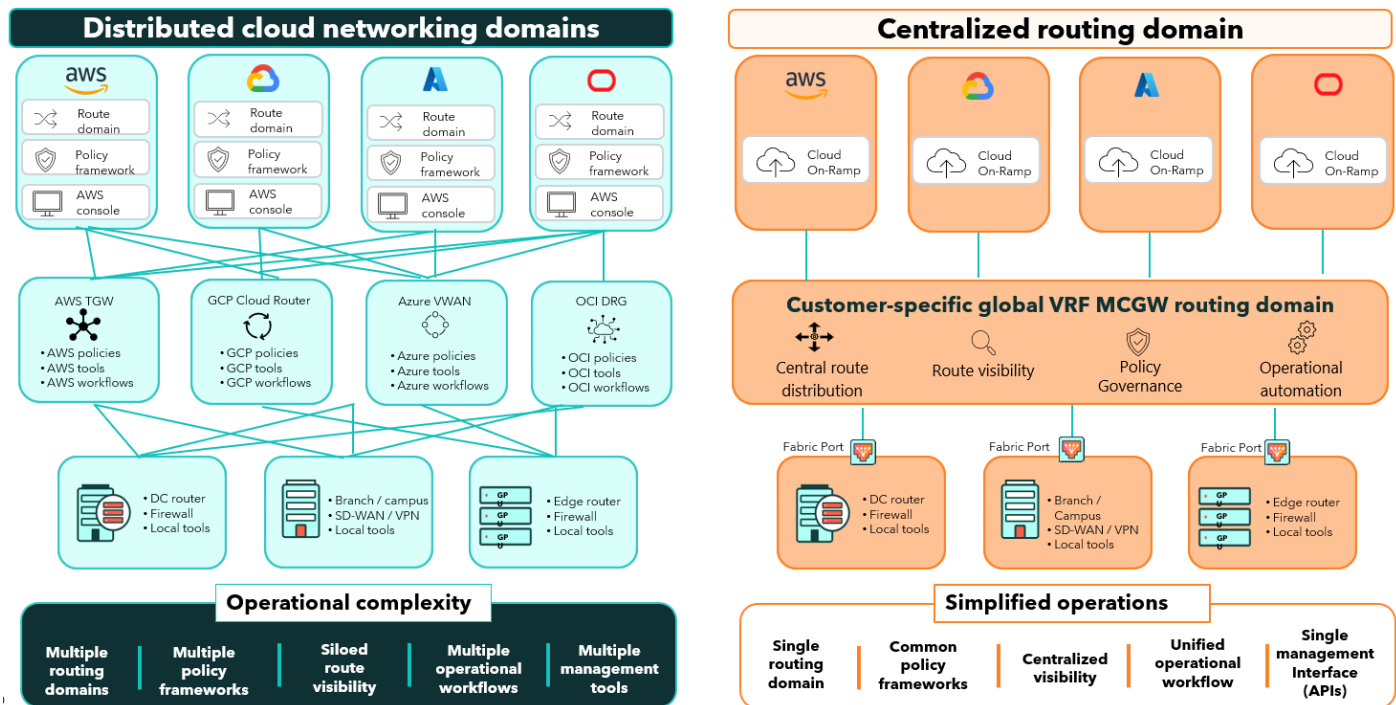


**Figure 6. Native multicloud architecture vs. MCGW routing domain architecture**

Native multi-cloud environments often distribute routing policy, route visibility, and operational workflows across multiple cloud-specific networking domains. MCGW introduces a customer-specific Global VRF that centralizes route exchange, policy enforcement, and routing visibility across connected cloud, data center, and enterprise environments. This model is designed to provide a more consistent framework for route governance, policy management, and infrastructure expansion.

## Routing domain design characteristics

| Design area                   | Native <u>multicloud</u> networking                                         | MCGW customer-specific routing domain                                      |
|-------------------------------|-----------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Routing domain                | Independent routing domains in each cloud and environment                   | Single customer-specific Global VRF provides centralized route exchange    |
| Route management              | Policies configured separately across providers and environments            | Routing behavior governed through a centralized policy framework           |
| Connectivity model            | Individual cloud constructs and connectivity services managed independently | Multiple environments connected through a common routing architecture      |
| BGP policy control            | Varies by provider and connectivity service                                 | Consistent import/export policy model across connections                   |
| Route visibility              | Visibility distributed across multiple tools and environments               | Routes visible within a centralized routing domain                         |
| Operational model             | Team manages multiple cloud networking constructs and workflows             | Unified operational model through Lumen Connect and APIs                   |
| Cloud expansion               | New environments often require additional provider-specific configuration   | New endpoints attached through existing routing framework                  |
| <u>Multicloud</u> consistency | Behavior influenced by differing cloud networking implementations           | Consistent routing and policy constructs across attached environments      |
| Policy enforcement            | Controls distributed across multiple administrative domains                 | Policies applied through centralized routing constructs                    |
| Failure response              | Troubleshooting often spans multiple cloud-native networking tools          | Route visibility and policy actions concentrated within the routing domain |
| Customer control              | Customer manages multiple environment-specific networking models            | Customer manages routing intent through a common architecture              |
| Architectural outcome         | Distributed connectivity with distributed control                           | Distributed infrastructure with centralized routing control                |

**Table 5. Routing domain design characteristics**

Native multi-cloud environments typically distribute routing policy, route visibility, and operational workflows across multiple networking domains. MCGW introduces a customer-specific Global VRF that centralizes route exchange, policy enforcement, and operational control across connected cloud, data center, and enterprise environments. This matrix illustrates the architectural differences that influence routing consistency and operational management.

# Distributed AI infrastructure reference architecture patterns

## Overview

The previous sections examined the architecture, service objects, operational model, routing lifecycle, and routing controls that make up a Multi-Cloud Gateway deployment. This section applies those concepts to representative deployment patterns that show how cloud providers, customer sites, data centers, and distributed infrastructure environments can participate in a common routing architecture.

These reference architectures are not intended to represent a specific customer topology. Instead, they illustrate how customer-specific routing domains, private connectivity services, and policy controls can be assembled to support common infrastructure requirements. In each pattern, the diagrams are intended to show service relationships and routing-domain participation rather than prescribe exact customer implementation details. Actual designs may vary based on cloud provider, region, access model, redundancy requirements, bandwidth, and routing policy.

## Cloud-to-Cloud connectivity pattern

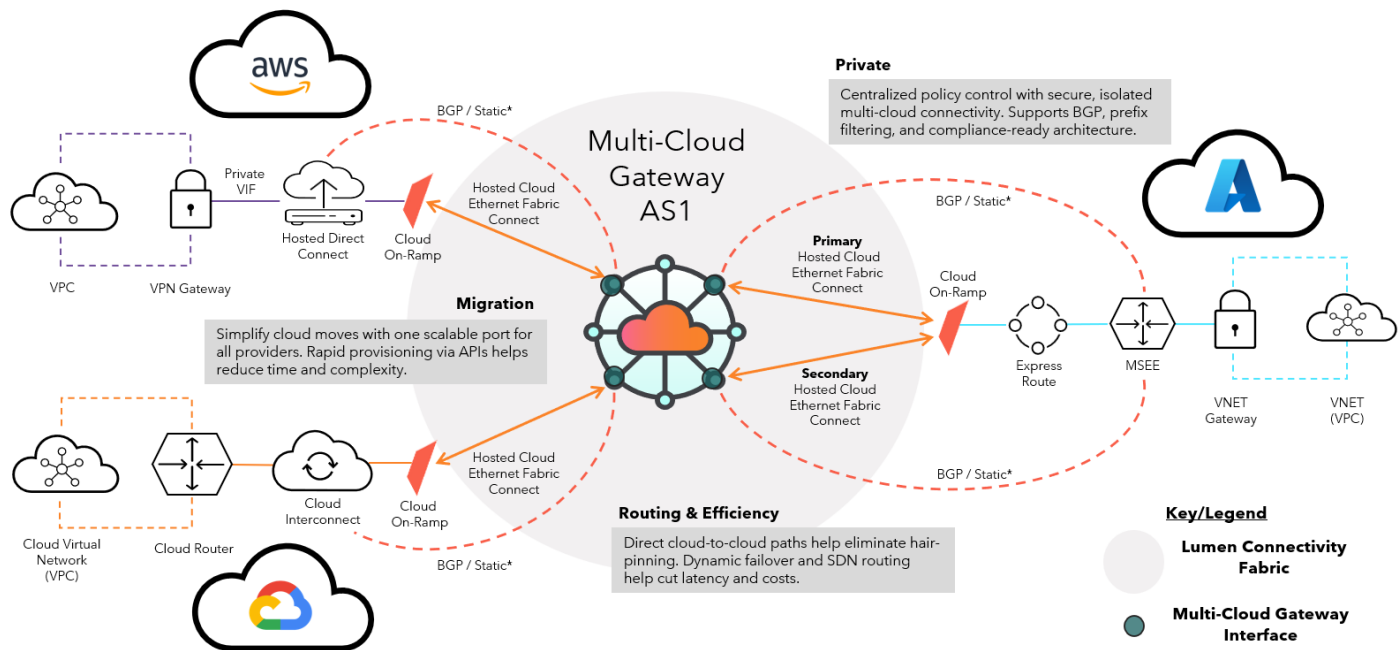


**Figure 7. Cloud-to-Cloud connectivity pattern**

MCGW provides a customer-specific routing domain for cloud-to-cloud route exchange. Cloud On-Ramps and hosted cloud connectivity services attach participating cloud environments, while the customer-specific Global VRF provides the common routing framework for route visibility, policy control, and route exchange.

Organizations increasingly operate workloads across multiple cloud providers. As cloud environments grow, maintaining consistent routing behavior across cloud networking constructs becomes increasingly challenging.

In this deployment model, Multi-Cloud Gateway provides a customer-specific routing domain through which cloud environments exchange route information. Cloud On-Ramps provide cloud attachment, MCGW Interfaces provide Layer 3 route exchange, and the customer-specific Global VRF becomes the centralized location where routing policies are applied.

Rather than relying only on provider-specific routing constructs in each cloud, customers can use MCGW as a common routing framework for route exchange and policy control across participating cloud environments, while preserving required cloud-specific networking constructs where appropriate.

## Site-to-Cloud connectivity pattern

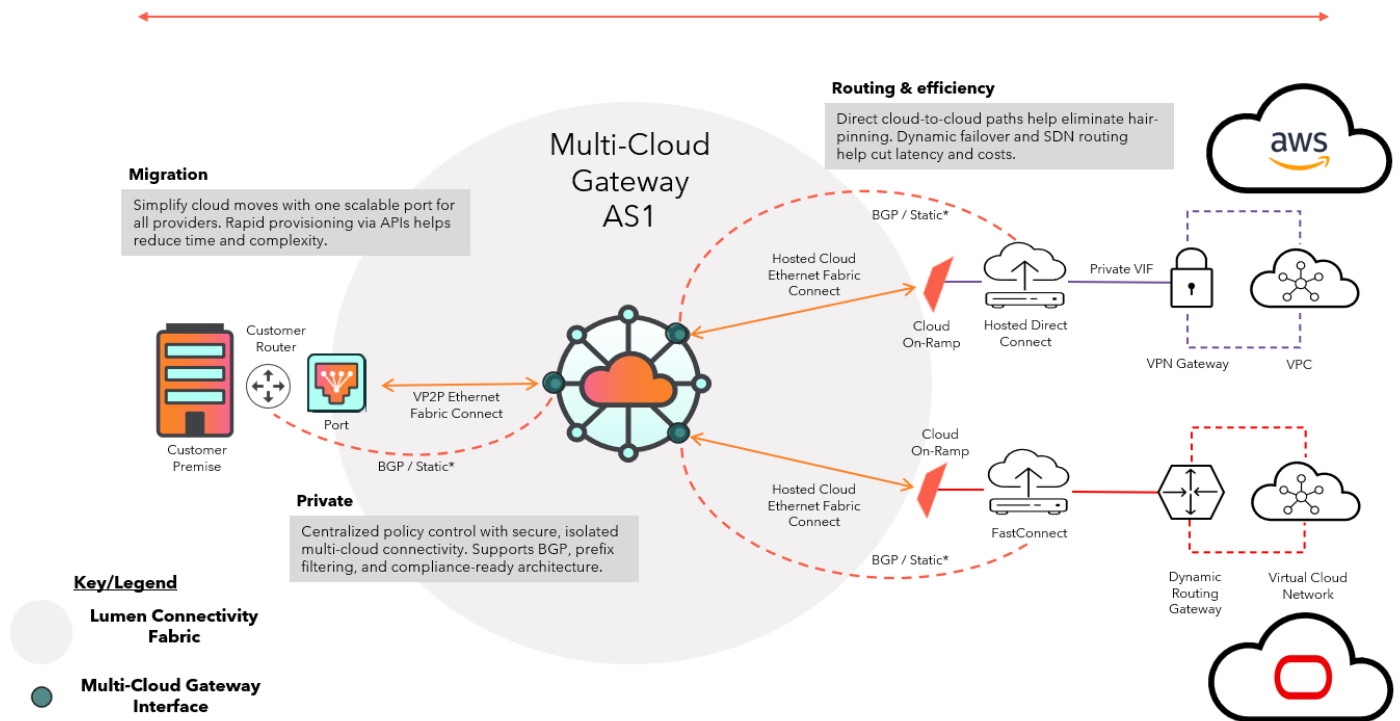


**Figure 8. Site-to-Cloud connectivity pattern**

Customer sites and data-center environments attach through Fabric Ports and connect to MCGW Interfaces using Ethernet Fabric Connect services. Cloud environments attach through Cloud On-Ramps, allowing routes to be exchanged through the customer-specific Global VRF with centralized policy control.

Site-to-cloud deployments typically require private connectivity, route visibility, and operational consistency between enterprise environments and cloud platforms.

In this pattern, customer environments attach through Fabric Ports and Ethernet Fabric Connect services. Cloud environments attach through Cloud On-Ramps. Multi-Cloud Gateway provides route exchange and policy control between attached environments, while the customer-specific Global VRF serves as the centralized routing domain.

This architecture enables centralized route management while maintaining private connectivity between participating infrastructure environments.

## Distributed infrastructure connectivity pattern

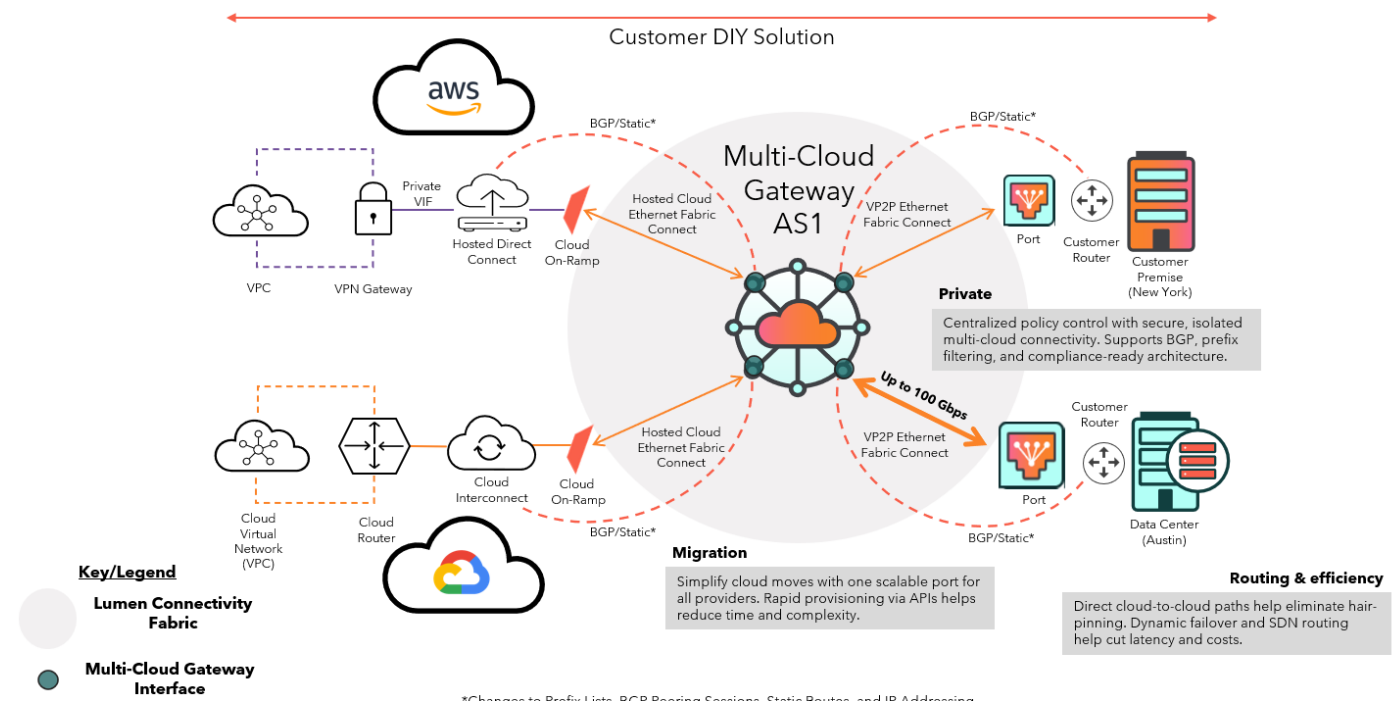


**Figure 9. Distributed infrastructure connectivity pattern**

MCGW acts as the routing and policy foundation for distributed infrastructure environments that span cloud providers, data centers, AI campuses, enterprise locations, and operational platforms. Routes are exchanged through a customer-specific Global VRF, enabling consistent route policy, visibility, and governance across connected domains.

Large-scale infrastructure environments often extend beyond traditional cloud deployments. Operational platforms, regional compute environments, research systems, AI campuses, enterprise locations, and data centers may need to participate within the same routing environment.

In this model, Multi-Cloud Gateway acts as a centralized routing and policy foundation that enables attached infrastructure domains to exchange route information through a common routing framework. Routing visibility, route filtering, and policy controls can be applied consistently across participating environments, regardless of physical location or cloud provider.

### Combined MCGW and LCI pattern

Some environments require both centralized routing control and private cloud connectivity. In this deployment pattern, LCI provides private cloud connectivity into cloud environments, while MCGW provides centralized route exchange, routing visibility, and policy control across connected sites, data centers, and cloud environments.

The services remain independent but can be combined when infrastructure requirements call for both centralized route control and private cloud access. This allows organizations to adopt either service independently or integrate both services within a broader distributed infrastructure architecture.

## Combined MCGW and LCI connectivity pattern

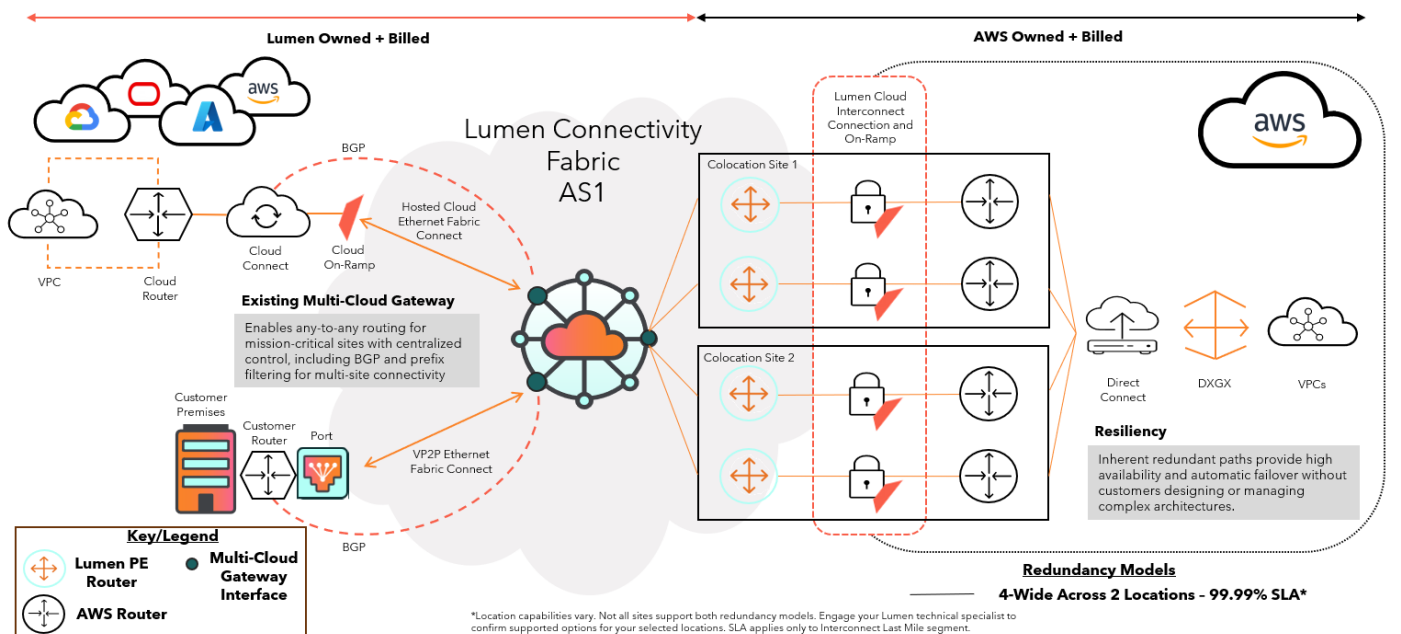


**Figure 10. Combined MCGW and LCI connectivity pattern**

MCGW and LCI can be deployed together when an architecture requires both centralized routing control and private cloud connectivity. LCI provides dedicated cloud access and private connectivity into supported cloud environments, while MCGW provides route exchange, routing visibility, and policy control through a customer-specific routing domain. The services remain architecturally distinct and can also be deployed independently.

# Deployment methodology

## Overview

The preceding sections examined service architecture, routing constructs, operational workflows, routing controls, and representative deployment patterns. Once an architectural model has been selected, infrastructure teams must translate that design into a deployable routing domain.

Multi-Cloud Gateway deployments typically follow a repeatable sequence that begins with endpoint identification and connectivity planning, progresses through routing-domain creation and policy definition, and concludes with route validation and ongoing operations. While implementation details vary by environment, the overall methodology remains consistent across cloud, data center, enterprise, and distributed infrastructure deployments.

The goal is to move from a conceptual architecture to a validated routing domain that can support production route exchange, policy control, and ongoing operations.

## MCGW deployment workflow

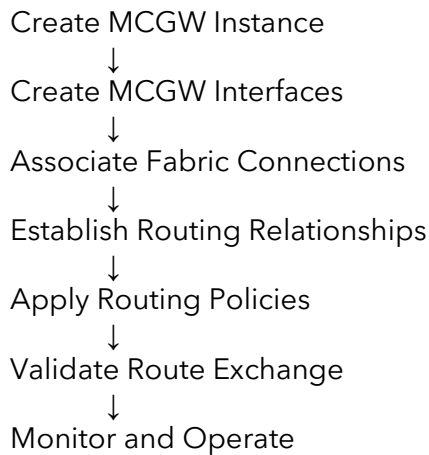
The typical Multi-Cloud Gateway deployment process follows the common sequence:

Identify Infrastructure Endpoints



Plan Connectivity Attachments





### **Step 1: Identify infrastructure endpoints**

The deployment process begins by identifying the environments that will participate in the routing architecture.

Typical environments include:

- Public cloud providers
- Enterprise sites
- Data centers
- Colocation environments
- Operational platforms
- Research environments
- Regional compute environments

Understanding participating endpoints establishes the foundation for attachment planning, route design, and connectivity requirements.

### **Step 2: Plan connectivity attachments**

Once participating environments have been identified, connectivity requirements can be evaluated.

This planning stage determines:

- Fabric Port requirements
- Cloud On-Ramp requirements
- Connectivity locations
- Access methods
- Required transport services
- Bandwidth considerations

Because routing and transport are intentionally separated within the MCGW architecture, this stage focuses on access and Layer 2 attachment planning before Layer 3 routing behavior is configured.

### Step 3: Create the routing domain

After connectivity planning has been completed, the customer-specific routing domain can be established. An MCGW Instance is created to host the customer Global VRF and associated routing relationships.

The routing domain becomes the centralized environment through which route exchange and routing-policy control are applied.

### Step 4: Create MCGW interfaces

MCGW Interfaces provide the Layer 3 attachment points through which connected environments participate in the routing domain.

Interfaces typically require:

- IP addressing
- Routing configuration
- BGP configuration (when applicable)
- Policy association
- Route-control configuration

Each interface serves as the logical boundary between the transport layer and the routing layer.

### Step 5: Attach Fabric Connections

After MCGW Interfaces are created, Fabric Connections attach the relevant endpoints to those interfaces through private Layer 2 connectivity. This step connects the planned access endpoints, such as Fabric Ports or Cloud On-Ramps, to the MCGW service model so routing can be established at the Layer 3 interface boundary.

This step is where the access and transport planning completed earlier is mapped to the MCGW Interfaces that participate in the customer-specific routing domain.

### Step 6: Establish routing relationships

Once interfaces have been created, route exchange can be established.

Depending on deployment requirements, this may include:

- BGP peering relationships
- Static routes
- Import policies
- Export policies
- Prefix lists
- Route filtering controls

The objective is to ensure that attached environments exchange route information according to defined routing policies and operational requirements.

### Step 7: Apply routing policies

Routing policies govern how routes are accepted, selected, and advertised within the customer routing domain.

Typical policy activities include:

- Prefix filtering
- Import-policy definition
- Export-policy definition
- Route visibility controls
- Route preference controls
- Advertisement controls

These policies establish the routing intent that governs traffic movement across connected infrastructure domains.

### Step 8: Validate route exchange

Before production traffic relies on the routing domain, route behavior should be validated.

Validation activities may include:

- Route acceptance verification
- Route advertisement verification
- Routing-policy validation
- Path-selection verification
- Connectivity validation
- Route visibility review

The goal is to confirm that routing behavior aligns with architectural expectations before operational workloads depend on the deployment.

### Step 9: Monitor and operate

Following deployment, the routing domain enters ongoing operational management. Lumen Connect provides visibility into deployed service objects, interface status, routing relationships, BGP session state, learned and advertised prefixes, and lifecycle management activities where available.

Operational teams can use portal and API-based workflows to manage ongoing routing-domain changes and infrastructure growth.

### Deployment considerations

The deployment methodology described in this section is intentionally architecture-agnostic. Whether implementing cloud-to-cloud routing, site-to-cloud connectivity, distributed infrastructure connectivity, or combined MCGW and LCI architectures, the same core workflow applies:

- Identify endpoints
- Establish connectivity

- Create routing constructs
- Define routing policies
- Validate route behavior
- Operate through a centralized control model

This consistency allows infrastructure teams to apply a common deployment framework across a wide range of distributed infrastructure architectures.

### **Architectural takeaway**

Successful MCGW deployments begin with connectivity planning but ultimately depend on the creation of a customer-specific routing domain that governs route exchange, policy control, and operational visibility. By separating connectivity functions from routing functions and following a structured deployment workflow, organizations can implement routing architectures that remain consistent as infrastructure expands across cloud, data center, and enterprise environments.

## **Failure domain and resiliency considerations**

### **Overview**

Distributed infrastructure environments often span multiple clouds, data centers, enterprise locations, cloud access points, and transport services. As the number of participating infrastructure domains increases, architects must consider how connectivity, routing, and operational dependencies are distributed throughout the architecture.

Failure-domain planning is not limited to individual links or devices. It includes understanding how routing relationships, connectivity attachments, access constructs, and operational dependencies interact within the shared customer-specific routing domain. This section examines architectural considerations that can help infrastructure teams identify, isolate where possible, and reduce the impact of potential failure domains within Multi-Cloud Gateway deployments.

## Failure domains and dependency boundaries in the MCGW routing architecture

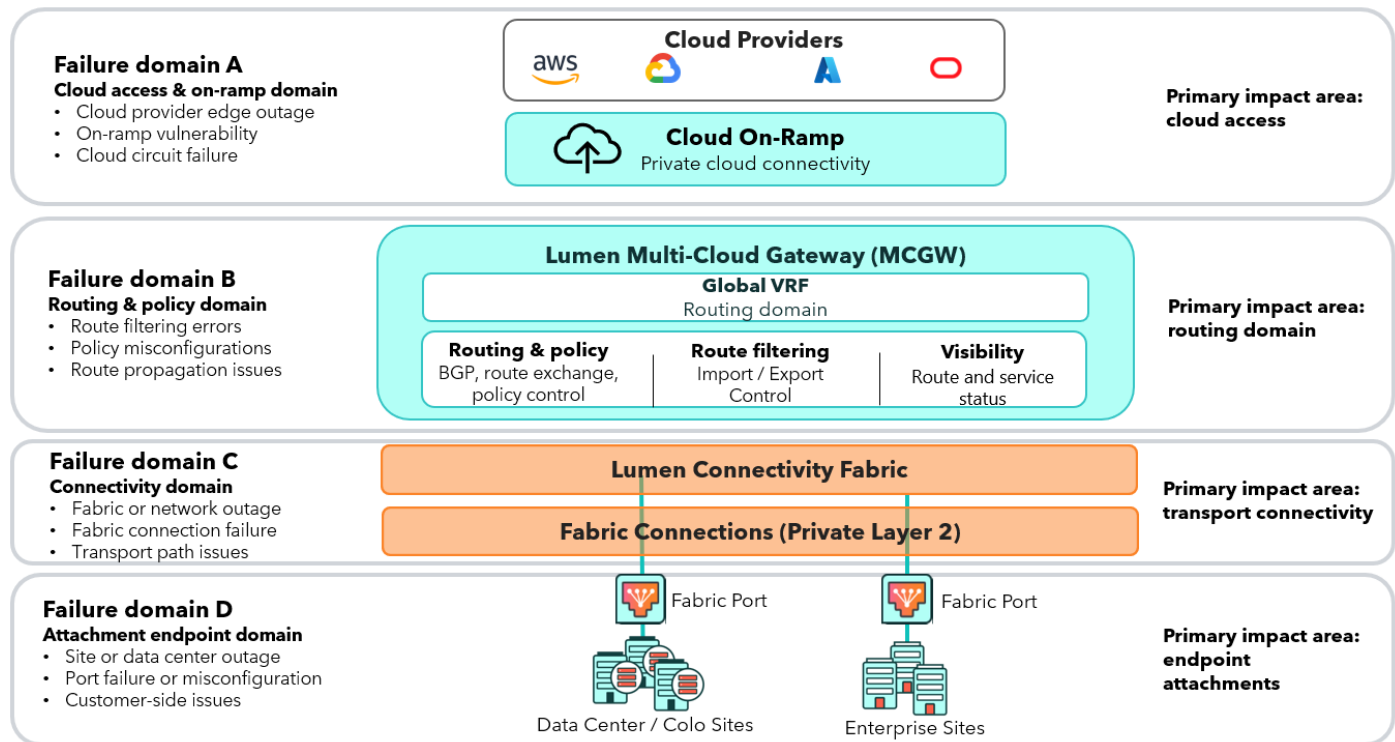


**Figure 11. Failure domains and dependency boundaries in the MCGW routing architecture**

The MCGW architecture can be evaluated as a collection of distinct failure domains spanning cloud access, routing and policy control, transport connectivity, and attachment endpoints. Understanding these dependency boundaries helps architects assess resiliency requirements, isolate operational impact, and design routing architectures that remain manageable as cloud, data-center, and enterprise environments expand.

### Routing domain considerations

The customer-specific Global VRF serves as the centralized routing domain through which attached environments exchange route information.

Because routing policies, route visibility, and route exchange decisions occur within this shared routing context, infrastructure teams should evaluate how routing dependencies are organized and governed across participating environments.

Architectural reviews should consider:

- Routing-domain scope
- Route propagation boundaries
- Routing-policy design
- Route filtering controls
- Administrative ownership
- Operational change processes

The objective is not simply route exchange, but controlled route exchange that remains operationally manageable as infrastructure grows.

## **Attachment and connectivity domains**

Connectivity attachments represent another important architectural consideration. Fabric Ports, Cloud On-Ramps, and Fabric Connections provide the access and transport layer through which infrastructure environments participate in the routing domain. These constructs do not make routing decisions, but they determine how environments attach to the service and which transport paths routing depends on.

Examples include:

- Enterprise access environments
- Data-center attachment points
- Cloud connectivity constructs
- Private transport services
- Interconnection points between infrastructure domains
- Access, region, or location diversity where applicable

Architects should understand which routing relationships depend on specific connectivity components and how those dependencies align with broader infrastructure objectives.

## **Policy dependency considerations**

Routing policies can influence not only route behavior but also the operational boundaries of the routing domain.

Import policies, export policies, prefix lists, route filtering controls, and route-preference controls collectively determine which routes become visible throughout the environment.

Poorly structured policies can introduce operational complexity, while consistent policy design can help simplify route governance across a distributed infrastructure architecture.

Considerations include:

- Policy consistency
- Route-filtering strategy
- Route-visibility requirements
- Route-advertisement boundaries
- Change-management procedures

These considerations become increasingly important as additional cloud providers, sites, and regional environments are integrated into the routing domain.

## **Operational visibility**

As infrastructure environments scale, operational visibility becomes an important component of resiliency planning. Lumen Connect<sup>SM</sup> provides a centralized operational framework for viewing deployed service constructs, routing relationships, configuration state, and routing-domain status where available.

Visibility into service objects, route behavior, and configuration state can help infrastructure teams evaluate operational health and support troubleshooting activities.

Visibility considerations may include:

- Routing-domain visibility
- Connectivity status visibility
- Route-validation workflows
- Policy-verification processes
- BGP session and route-state review where applicable
- Operational lifecycle management

Rather than treating connected environments independently, operational information can be evaluated through a common framework aligned to the customer routing domain.

### **Distributed infrastructure planning**

The distributed infrastructure patterns described earlier in this paper demonstrate how cloud providers, enterprise locations, data centers, and operational platforms can participate within a common routing architecture. As these environments grow, architects should evaluate both routing and operational dependencies across participating domains.

Areas frequently reviewed include:

- Routing-domain scope
- Attachment strategy
- Policy governance
- Change-control boundaries
- Operational monitoring
- Infrastructure growth planning

During design review, architects should be able to identify which endpoints depend on each attachment path, which routes are allowed to propagate between domains, which policies prevent unintended route visibility, and which operational signals will be used to validate route and session health.

These considerations help organizations maintain operational consistency as new environments are integrated into the architecture.

### **Architectural takeaway**

Resiliency planning within distributed infrastructure environments extends beyond connectivity alone. Routing domains, connectivity attachments, policy controls, and operational processes each represent architectural considerations that influence how infrastructure behaves as environments scale.

By evaluating these dependencies during design and deployment, organizations can create routing architectures that remain manageable, visible, and operationally consistent across cloud, data-center, and enterprise environments.

# Design considerations and best practices

## Overview

The preceding sections examined the architecture, service constructs, operational model, routing lifecycle, deployment methodology, and reference patterns that make up a Multi-Cloud Gateway deployment. Collectively, these elements establish a framework for connecting distributed infrastructure environments through centralized routing and policy control.

While deployment patterns may vary, several architectural principles consistently contribute to simpler operations, clearer routing visibility, and more predictable route management outcomes. The following considerations are intended to help architects plan and operate customer-specific routing domains across cloud, data-center, and enterprise environments.

## Design for routing-domain simplicity

One of the primary advantages of a customer-specific routing domain is the ability to centralize route exchange and routing policy management. As new environments are connected, architects should evaluate whether routing decisions can be governed through a common policy framework rather than introducing environment-specific routing models wherever possible.

Recommended considerations include:

- Establish clear routing-domain boundaries
- Standardize route policy design approaches
- Minimize unnecessary policy variation
- Maintain consistent route advertisement practices
- Document routing intent for connected environments

Simplifying routing-domain design can help reduce operational complexity as infrastructure scales across multiple environments.

## Separate connectivity from routing

A core architectural principle throughout this paper is the separation of transport functions from routing functions.

Fabric Ports, Cloud On-Ramps, and Fabric Connections provide connectivity and transport. MCGW Interfaces and the customer-specific Global VRF provide route exchange, route control, and routing policy control. This boundary is aligned with the service model where access endpoints and Layer 2 attachment constructs are distinct from the virtual Layer 3 routing and policy domain.

Maintaining this separation can simplify both deployment planning and operational troubleshooting.

Recommended considerations include:

- Evaluate connectivity requirements independently from routing requirements
- Design attachment models separately from routing-policy design
- Treat transport services and routing services as distinct architectural layers
- Maintain clear ownership of connectivity and routing responsibilities

## **Establish consistent routing policies**

Routing policies influence how routes enter, move through, and leave the customer-specific routing domain.

As distributed environments expand, inconsistent routing controls can increase operational complexity and make routing behavior more difficult to understand and troubleshoot. The routing lifecycle described earlier in this paper illustrates the importance of consistent policy application throughout route admission, route selection, and route advertisement workflows.

Recommended considerations include:

- Apply consistent import-policy design standards
- Apply consistent export-policy design standards
- Standardize prefix-filtering approaches
- Clearly define route-advertisement boundaries
- Establish documented policy review processes

## **Design for operational visibility**

Visibility becomes increasingly important as additional sites, clouds, and infrastructure domains are attached to the routing environment.

Operational teams should consider how routing status, policy application, route visibility, and service-object status will be reviewed and managed throughout the service lifecycle. Lumen Connect<sup>SM</sup> provides a centralized framework for managing deployed service objects, reviewing configuration state, and operating routing-domain workflows where available. The Lumen Connect<sup>SM</sup>/API/Automation model is also reflected in the MCGW service architecture materials.

Recommended considerations include:

- Establish route-validation procedures
- Review route visibility across connected environments
- Monitor routing relationships and service-object status
- Use consistent operational management processes
- Evaluate routing changes through controlled operational workflows

## **Plan interface and addressing strategies early**

MCGW Interfaces represent the virtual Layer 3 attachment points through which routing relationships are established. The source architecture materials define MCGW Interfaces as virtual Layer 3 endpoints where IP addressing, routing, and policy are configured.

Addressing, route exchange, and policy application are all influenced by interface design decisions. Interface planning is often most effective when performed before large-scale connectivity onboarding begins.

Recommended considerations include:

- Define interface requirements early in the design process

- Establish a consistent addressing strategy
- Align interfaces with routing-policy boundaries
- Document routing relationships associated with each interface
- Review operational ownership expectations before deployment

### **Evaluate cloud connectivity independently from routing requirements**

Cloud connectivity and routing architecture serve different functions and should be evaluated independently. Organizations may require private cloud connectivity, centralized routing domains, or both. The MCGW and LCI design-role discussion demonstrated that routing-domain requirements and cloud-connectivity requirements do not always overlap.

Recommended considerations include:

- Determine whether private cloud connectivity is required.
- Determine whether centralized route control is required.
- Evaluate cloud-access requirements independently from route-management requirements.
- Select architectures based on operational objectives rather than deployment tradition.
- Consider whether MCGW, LCI, or both best support the desired outcome.

### **Plan for growth**

Many routing architectures initially support a small number of environments but eventually expand to include additional clouds, regions, sites, operational platforms, and data centers.

Architects should consider how routing policies, route visibility practices, operational processes, and connectivity models will evolve as new environments are added to the routing domain.

Recommended considerations include:

- Design policies with future expansion in mind
- Establish repeatable onboarding processes
- Maintain consistent service-object design patterns
- Review routing-domain scope periodically
- Validate route behavior before adding additional environments
- Standardize deployment methodologies across environments

## Design considerations summary

| Design area            | Recommended practice                                                              |
|------------------------|-----------------------------------------------------------------------------------|
| Routing domains        | Establish clear routing-domain boundaries and centralized policy governance       |
| Connectivity design    | Separate transport planning from routing design                                   |
| Policy management      | Apply consistent import, export, and prefix-filtering controls                    |
| Operational visibility | Use centralized operational workflows and route validation processes              |
| Interface planning     | Establish addressing and interface strategies early                               |
| Cloud connectivity     | Evaluate connectivity requirements independently from routing-domain requirements |
| Growth planning        | Design routing architectures with future expansion in mind                        |

**Table 6. Design considerations summary**

Effective routing architecture combines centralized route governance, consistent policy application, operational visibility, and structured deployment practices to support distributed infrastructure environments as they scale.

### Architectural takeaway

Successful distributed infrastructure architectures rely on consistent routing-domain design, clear policy boundaries, repeatable operational workflows, and well-defined connectivity models. By separating transport from routing, centralizing route governance, maintaining operational visibility, and planning for growth, organizations can keep routing architectures manageable as infrastructure expands across cloud, data-center, and enterprise environments.

# Conclusion

Modern AI infrastructure increasingly extends across cloud providers, data centers, enterprise environments, operational platforms, research systems, and regional compute locations. As these environments grow, the challenge is not only establishing connectivity, but maintaining consistent routing behavior, policy governance, operational visibility, and lifecycle management across a distributed infrastructure footprint.

Throughout this paper, we examined how Lumen Multi-Cloud Gateway (MCGW) provides a customer-specific routing domain that enables cloud, data center, and enterprise environments to exchange routes through a centralized routing and policy framework. Rather than treating each infrastructure domain as a separate networking construct, MCGW introduces a common routing model built around customer-specific Global VRFs, MCGW Interfaces, routing policies, and route-management controls.

The paper also explored the customer-visible service constructs that underpin MCGW deployments, including MCGW Instances, MCGW Interfaces, Fabric Connections, Fabric Ports, Cloud On-Ramps, and Lumen Connect<sup>SM</sup>. Together, these constructs separate connectivity functions from routing functions, giving architects a clearer way to understand where traffic attaches, where Layer 2 transport ends, where Layer 3 routing begins, and where routing policy is applied.

Operationally, MCGW provides a lifecycle-oriented model for deploying and managing routing domains. Through Lumen Connect<sup>SM</sup> and associated APIs, infrastructure teams can provision service constructs, establish routing relationships, apply policies, validate route behavior, and manage ongoing operational activities through a centralized operating model. The routing lifecycle, policy framework, and operational workflows described throughout this paper show how route information can be governed as infrastructure expands across increasingly distributed environments.

The paper also distinguished between routing-domain requirements and cloud-connectivity requirements. Lumen Cloud Interconnect (LCI) provides private cloud connectivity and dedicated cloud-access capabilities, while MCGW provides centralized routing, route visibility, and policy control. Organizations can deploy either capability independently or combine both services when architectural requirements call for private cloud connectivity and centralized route governance within the same design.

The reference architecture patterns presented in this paper illustrated several approaches for applying these concepts across cloud-to-cloud routing, site-to-cloud connectivity, distributed infrastructure environments, and combined MCGW plus LCI architectures. While specific implementations vary, each pattern reinforces the same architectural principle: centralized routing and policy control can provide a more consistent framework for managing route exchange across diverse infrastructure domains.

Ultimately, the architectural challenge facing many infrastructure teams is not simply how to connect more environments. It is how to govern route exchange, policy control, operational visibility, and lifecycle management as those environments expand. By combining customer-specific routing domains, policy-driven controls, private connectivity options, and API-enabled operations, MCGW and LCI provide a

framework for designing and operating distributed infrastructure architectures across cloud, data center, and enterprise environments.

### **Key takeaway**

As infrastructure expands across multiple domains, consistent routing, policy control, visibility, and operations become increasingly important. MCGW provides centralized route exchange and policy control through a customer-specific routing domain, while LCI provides private cloud connectivity. Used independently or together, these services help infrastructure teams design and operate distributed architectures with clearer routing governance and a more consistent operational model.

### **Why Lumen?**

Lumen combines private connectivity, centralized routing, and API-driven operations within a common architectural framework. Multi-Cloud Gateway and Lumen Cloud Interconnect help infrastructure teams apply consistent routing, policy, and operational controls across cloud, data center, AI, and enterprise environments.

866-352-0291 | [lumen.com](https://lumen.com) | [info@lumen.com](mailto:info@lumen.com)

Services not available everywhere. Business customers only. Lumen may change, cancel or substitute products and services, or vary them by service area at its sole discretion without notice. ©2026 Lumen Technologies. All Rights Reserved.

**LUMEN**<sup>®</sup>