

PROGRAMMABILITY IN 5G NETWORKS






Contents

- 1. Executive Summary 4**

- 2. Introduction: 5G Brings Major Changes..... 4**
 - 2.1. Managing a More Dynamic Network..... 5
 - 2.2. Network Equipment Providers Must Adapt 5

- 3. 5G Architecture 6**
 - 3.1. Overview of 5G..... 6
 - 3.2. Inside a 5G Architecture 7
 - 3.3. Cloud-Native Technology..... 8
 - 3.4. Network Slicing10

- 4. Management, Orchestration,
and Automation in 5G Networks 12**
 - 4.1. 5G Brings New Requirements for Network Management12
 - 4.1.2. Programmable Interfaces in 5G Automation.....13
 - 4.2. Programmability in 5G Networks: Overview.....13
 - 4.3. Key Concepts in Programmability: Transactions.....14
 - 4.3.2. Transaction Support in Programmable Interfaces.....14
 - 4.4. Key Concepts in Programmability: Intent-Based Networking...15
 - 4.4.2. Service Orchestration for IBN.....16
 - 4.4.3. Implementing Intent-Based Networking.....16
 - 4.4.4. Standardized Data Models and Interfaces in IBN.....17
 - 4.5. Key Concepts in Programmability: Model-Driven Telemetry17



**5. Management and Orchestration
Across 5G Network Domains18**

- 5.1. Managing the Next-Generation RAN18
- 5.1.2. Virtualizing and Automating the RAN19
- 5.2. Managing 5G Core Networks..... 20
- 5.3. Managing 5G Transport Networks 20

6. Network Slice Orchestration.....21

- 6.1. A New Model for Managing Virtualized Networks.....21
- 6.2. Fundamentals of Network Slice Orchestration21
- 6.3. Intent-Based Slice Orchestration..... 22
 - 6.3.2. Inside a Slice Orchestrator 23
 - 6.3.3. Mapping Intent for 5G Slices..... 25
 - 6.3.4. Lifecycle Management of Network Slices 26

Conclusion 27

1. Executive Summary

Mobile network operators (MNOs) have long asked network equipment providers to support standards-based, data model-driven programmability in network elements. In previous-generation architectures, however, this request was not always an urgent requirement. Many MNOs have viewed model-driven programmability as a “nice-to-have” capability. They recognized that, in the future, programmable network elements would help them provision services more simply, quickly, and cost-effectively, and lay the foundation for end-to-end network automation. In the present though, it was still possible to manage their networks using vendor-specific CLI and basic scripting, and many did. Now, as operators roll out 5G networks, this approach is no longer a viable strategy for network management. Model-driven programmability is becoming essential.

Managing network elements for dynamic end-to-end 5G services, especially in core and access networks, is exponentially more complex than in previous architectures. There is simply no way to do the things operators need to do for 5G—dynamically scale network functions with demand, quickly provision services across multiple domains and vendors, configure “network slices” with different attributes over the same physical infrastructure—without automation. And automation, especially in multivendor environments, requires standardized, model-driven programmability.

When developing network functions for service provider environments then, it is more important than ever for NEPs to build model-driven programmability into their network functions (NFs). Ideally, they should do it via support for YANG data models and standardized NETCONF interfaces.

This paper provides an overview of programmability in 5G networks. It details the ways that 5G environments differ from legacy architectures and the capabilities operators must employ to manage and automate them. It illustrates how model-driven programmability enables many of the core capabilities that operators require to monetize their 5G network investments. Ultimately, it demonstrates why NEPs looking to sell products into service provider networks should make support for standardized data models and programmable interfaces a top priority.

2. Introduction: 5G Brings Major Changes

Each new generation of mobile network technology in the past—2G to 3G to 4G—has expanded operators’ capabilities and enabled new service experiences for their subscribers. In practice, however, while successive generations brought faster data rates and better performance, the underlying architecture of the mobile network remained essentially unchanged. 5G represents a significant departure, a network fundamentally different than anything that’s come before.

First, the technical capabilities of 5G networks (such as higher capacity, ultra-low latencies, vastly improved performance and efficiency in radio networks) enable transformative new use cases for consumers and enterprises. From fully automated factories, Smart Cities, and other Internet of Things (IoT) applications, to connected vehicles, mixed reality gaming, remote surgery and telemedicine, 5G enables network experiences that were not possible before.

Just as important for MNOs, 5G brings powerful new tools to differentiate services and launch new business models. With the concept of network slicing in particular (detailed later in this paper), operators can now build virtual networks tailored to the services running on them, enabling a wide range of new, industry-specific enterprise applications.

2.1. Managing a More Dynamic Network

By design, 5G networks must be open and adaptable to a continually expanding set of use cases. They are architected to be much more flexible and scalable than the “one-size-fits-all” mobile network models of the past. As a result, the management of 5G networks becomes orders of magnitude more complex than in previous-generation architectures. This complexity derives from:

- **Ubiquitous virtualization:** To enable on-demand scaling and repositioning of network resources, network elements in every domain (core, access, transport, etc.) must be virtualized and, increasingly, containerized.
- **Greater diversity in network services and slices:** Operators must now support multiple deployment scenarios using virtualized networks with attributes that are finely tuned for specific verticals and use cases.
- **Huge increase in data to be managed:** 5G networks support many more connections and devices (User Equipment, or “UE” in 3GPP terminology), with much more data flowing back and forth. To keep pace with exploding demand, 5G networks must dynamically spawn new NF instances for every element used in a given service. The need to continually copy data from one NF to another translates to a massive increase in the amount of session and state information that must be maintained.

There is no way to accommodate these requirements using the network element-centric management models of the past. Relying on CLI configuration and CLI-based automation (such as basic scripting) of network elements won't work. It's too slow, too error-prone, too vendor-specific. REST interfaces don't solve the problem either, as they are typically non-standardized and can be just as heterogeneous as CLIs. And, while IT automation tools like Chef, Puppet, and Ansible can be useful to automate certain tasks, they lack critical features that full programmability and automation require, such as transactions, configuration validation, rollback management, and service discovery. Ultimately, having a programmable, automation-ready network is now a core requirement for 5G success.

2.2. Network Equipment Providers Must Adapt

The 3GPP 5G standard introduces new concepts to facilitate manageability and automation in dynamic 5G network environments. In fact, unlike previous-generation architectures, 5G is designed from the ground up to employ virtualized NFs and cloud-native principles. For operators to succeed, however, they need network elements that support these newer methodologies for programmability and automation. It is therefore critical that developers of network functions (physical and virtual) understand what 5G automation entails, and that they build support for the model-driven programmability that makes it possible.

The following section provides an overview of a 5G architecture and the unique attributes that separate 5G from previous-generation networks. Next, we will explore manageability, orchestration, and automation in 5G architectures. We will discuss what programmability entails in this context and the role of newer methodologies such as intent-based networking. Finally, we will detail why support for YANG data models and NETCONF is the most effective way to enable the new management models that network operators now require.

3. 5G Architecture

3.1. Overview of 5G

5G is the fifth-generation wireless technology specification defined by 3GPP (Third-Generation Partnership Project), the successor to 4G and LTE mobile networks. While the [3GPP 5G specification](#) defines a wide range of new capabilities, the most significant in enabling new applications and MNO business models are:

- **Faster speeds:** MNOs are familiar with the challenge of trying to keep pace with their customers' continually growing bandwidth demands. New applications on the horizon though—things like 8K video streaming, virtual reality applications, connected vehicles—churn through bandwidth on a scale never previously approached, that legacy infrastructures can't accommodate. With new radio technologies and the ability to dynamically scale resources with demand, 5G makes these new application experiences possible.
- **Greater capacity and density:** The IoT began in earnest in the age of 4G/LTE, but today's networks are optimized for conventional mobile endpoints like smartphones. They are poorly suited to the needs of many IoT use cases, which can require low-cost, low-power connectivity at massive scales. As the number of connected devices exceeds [29.3 billion by 2023](#)—an increase of nearly 60 percent from just five years before—current networks, especially in densely populated areas, can't keep pace. New 5G radio technologies enable support for up to a million connected devices per square kilometer, a tenfold increase over 4G. 5G also allows operators to use virtualized network functions (VNFs) that can scale up resources on demand.
- **Lower latency:** Adding bandwidth allows for faster data rates, but that does not automatically translate to lower latencies. In fact, latency is a far more complex attribute to improve, as it requires the coordination of every element in the end-to-end path of a low-latency service. 5G includes new timing and synchronization specifications that allow for ultra-low latencies less than 1 millisecond. This will be essential for emerging real-time applications like remote surgery, self-driving cars, industrial automation, and others.

These capabilities, combined with new 5G radio innovations such as massive MIMO and beamforming, allow 5G networks to support current use cases more efficiently while enabling tomorrow’s more dynamic real-time services. At the same time, 5G introduces significant new management challenges for operators. Chiefly:

- They must be able to stitch network elements together seamlessly—across radio access networks (RAN), transport, core, and other domains—to provision end-to-end services and slices.
- They must support on-demand, automated scaling of virtualized resources to keep pace with demand, without requiring human intervention.
- They must support closed-loop service assurance, where the network automatically identifies problems or impending service-level agreement (SLA) violations and responds automatically, ultimately enabling self-healing networks.

5G relies on existing technologies such as software defined networking (SDN) and network functions virtualization (NFV) to help enable these capabilities. However, it also accelerates the adoption of newer cloud-native models in operator environments, including containers and container orchestration. All of these factors raise new considerations for the network functions that will participate in 5G services.

3.2. Inside a 5G Architecture

Figure 1 depicts a basic (non-roaming) 5G architecture encompassing a variety of common network functions. (For more details on the specific NFs depicted here, see the 3GPP 5G System Architecture Technical Specification [TS 123 501](#).)

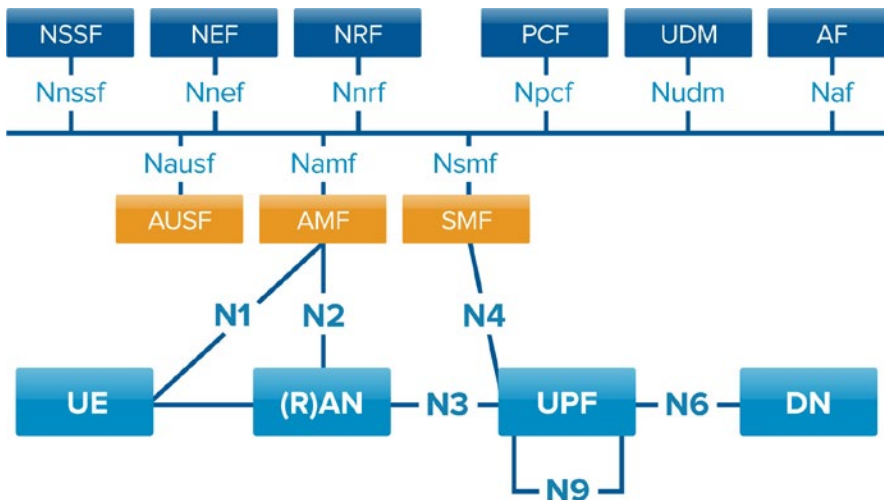


Figure 1. Basic (Non-Roaming) 3GPP 5G Architecture

With respect to network functions, this architecture has the following key attributes:

- **Open, service-based architecture (SBA):** Network functions in 5G systems are virtualized, self-contained, and run independently. However, each NF provides services to, and consumes services from, other NFs via RESTful APIs using a Service-Based Interface (SBI). This provides a modular framework for NFs from different providers to coexist and interoperate in the same network.
- **Control and user plane separation (CUPS):** The ability to distribute data plane functions, such as video streaming, out closer to users is an essential requirement to affordably scale network resources for new 5G services. With “CUPS” (as introduced in [3GPP Release 14](#)), operators can enable more flexible and distributed network models while maintaining centralized control. CUPS also helps facilitate network slicing, as detailed later in this paper.
- **Stateless network functions:** NF compute resources should be decoupled from storage to enable faster scaling and greater resiliency. For core networks in particular, the network should be able to instantiate NFs very quickly in response to real-time demand, or to failures or other events as part of SLA assurance. Stateless NFs allow for services to dynamically react to changes in the network and service states.

These architectural attributes do not necessarily imply a direct requirement for model-driven programmability. The ability to use consistent, standardized interfaces, however, makes it far easier for MNOs to assemble and reassemble NFs as part of an automated, end-to-end management framework.

3.3. Cloud-Native Technology

5G assumes virtualized, self-contained network functions communicating via a service-based interface. In many cases, however, legacy virtualization models cannot meet the requirements of 5G architectures and services.

The NFV models that operators have adopted in recent years brought significant benefits over prior networking approaches that used dedicated, purpose-built hardware. However, a reliance on virtual machines (VMs) means that current NFV approaches cannot deliver the ideal speed and agility that operators want in 5G networks. The long boot times and failure restarts associated with VMs, for instance, can negatively impact the availability of 5G networks and services. VMs also consume significant resources—a major issue in 5G networks, where the same physical infrastructure may be supporting multiple slices and applications simultaneously. Finally, a need for hypervisors introduces extra overhead that can impact overall system performance.

For all these reasons, 5G systems are designed to employ more modern, “cloud-native” models from the worlds of IT and hyperscale data centers. To be considered cloud-native, according to the [Cloud-Native Computing Foundation](#), a system or application must feature:

- **Microservices-oriented architecture:** In a cloud-native model, applications are decomposed into a loose collection of fine-grained services rather than a single, monolithic piece of code. Each distinct “microservice” implements business capabilities, runs in its own process, and communicates with other services and the cloud via HTTP APIs or messaging.
- **Containerization:** Resources should be packaged in a scalable format, such as Docker containers, rather than VMs. Containers encapsulate software with the minimal set of runtime resources it needs to perform its function. Unlike VMs, which encapsulate the full application platform and its dependencies, containers are built for lightweight microservices.
- **Dynamic orchestration:** Cloud-native application environments use orchestration software such as Kubernetes to automate the deployment, scaling, and management of containers and microservices across clusters of hosts.

These cloud-native principles offer significant advantages for 5G networks. First, they allow for improved flexibility, scalability, performance, and speed in deploying network resources. They optimize resource efficiency compared to more heavyweight VMs and hypervisors. They enable simplified, automated lifecycle management and orchestration of network functions. And, they provide native support for open environments. This allows operators to use a heterogenous ecosystem of network functions from multiple vendors, all working together to enable the core functionality of 5G.

When combined with standards-based programmable interfaces, cloud-native technology can be hugely beneficial for the management of 5G networks. Operators gain a framework for fast, automated orchestration of end-to-end services in multi-domain/multivendor 5G architectures.

Network Programmability and Cloud-Native NFV

The introduction of NFV brought major benefits to service providers, enabling a new generation of virtualized NFs implemented entirely in software, running on commercial off-the-shelf servers. Now, a new evolution in network elements is under way: the shift to cloud-native applications.

In a cloud-native world, decomposed container-based applications can more easily take advantage of the shared resources, speed, and agility of cloud environments. Since VNFs are, at their core, software applications, they too are now being decomposed into their constituent microservices to become “cloud-native.” However, deploying and managing VNFs, especially in complex multivendor operator environments, is different than running other types of applications in the cloud.

For details on the ways that cloud-native approaches impact programmable networks and the requirements for developing cloud-native VNFs, see the Tail-f white paper [Network Programmability in Cloud-Native NFV](#).

3.4. Network Slicing

In previous networks, a “one-size-fits-all” architectural approach was sufficient to meet most application requirements and customer needs. 5G introduces a more flexible model, where operators can create multiple virtual networks running over the same physical infrastructure, each finely tuned for a specific type of service, meeting specific application requirements (Figure 2). For example, operators could employ one slice for low-cost IoT connectivity, another for ultra-low-latency telemedicine, one for augmented reality gaming, one for 8K video streaming, etc. This is the concept of network slicing, and it’s a core component of how operators plan to monetize their 5G network investments and expand into new consumer and enterprise markets.

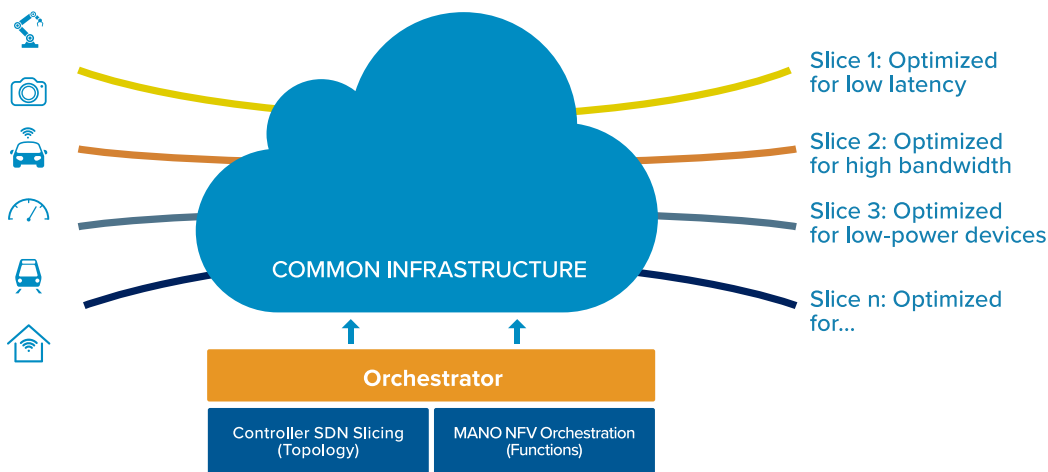


Figure 2. Conceptual View of Network Slicing

Fundamentally, network slicing is the ability to overlay multiple logical networks on a common physical/virtual infrastructure, across multiple domains, while maintaining separate SLAs, policy, charging, identity, monitoring, etc., for each service (Figure 3). Network slicing allows operators to tailor network services for specific subscriber experiences and enterprise applications, and guarantee services with a wide range of attributes under SLAs. This allows them to meet diverse demands on geographical coverage for access, density, speed, and latency, all using the same physical infrastructure.

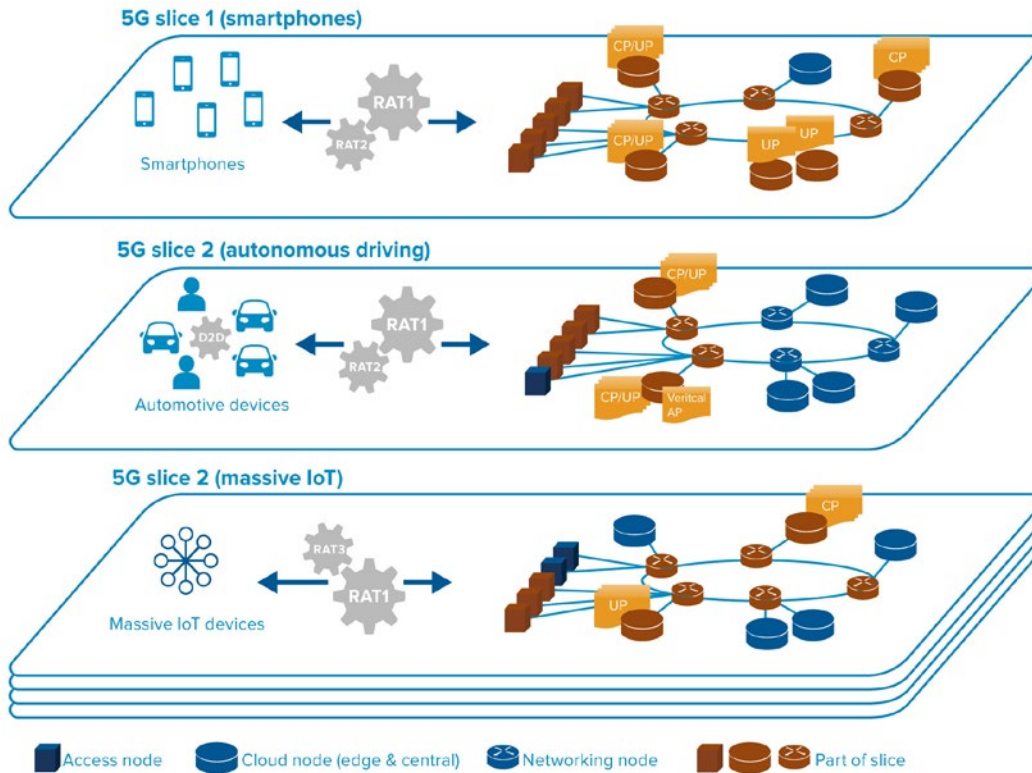


Figure 3. Network Slices Fulfill Different Use Cases While Sharing the Same Network Resources

To enable network slicing, operators must abstract away fine-grained management details for physical and virtual network resources for each domain via APIs. They must implement an intent-based architecture (as detailed in the following section). Most important, providing an application with a logical slice of the network, based on a negotiated SLA, requires end-to-end orchestration and automation of network configurations. Standardized, model-driven programmability of NFs is the most effective way to enable this.

4. Management, Orchestration, and Automation in 5G Networks

4.1. 5G Brings New Requirements for Network Management

5G introduces a far more dynamic service environment than in previous-generation architectures, where network functions—both physical and virtual—were largely fixed. This dynamism presents unique network management challenges that legacy approaches cannot address, and that NF developers must consider when building programmability for their products. NFs that will participate in 5G services must be designed for:

- **Autonomous real-time orchestration:** Much more than in previous-generation architectures, 5G NFs must be designed for flexibility, agility, and scale. To meet the requirements of applications and network slices, and maintain SLAs, NFs must be designed to be continually deployed, reconfigured, and scaled in and out—quickly and automatically.
- **Manageability within heterogeneous architectures:** In 5G architectures, operators will need to manage many more NFs operating across multiple domains—in some cases, even across multiple service provider networks. (For example, IoT transportation and logistics applications may need to support data roaming across different regions and operator footprints. Or, a vertical use case may encompass services from multiple service providers.) The multi-domain nature of 5G networks, spanning RAN, transport, core, and public and private clouds, calls for more flexible management. Operators must be able to integrate multiple domain-specific management solutions/orchestrators to provide an end-to-end service.
- **Closed-loop assurance:** Assuring services in dynamic 5G architectures is just as complex as provisioning them and demands the same degree of end-to-end automation. The network must be able to instantiate, change, and scale NFs automatically in response to failures or other network events or triggers, such as SLA violations, in a service or slice. That includes the ability to reconfigure services without having to turn off NFs.
- **Cloud-native orchestration:** As discussed, NFs that will run in 5G networks should be designed according to cloud-native principles. Operators should be able to decompose and manage an NF's constituent microservices via container-based orchestration tools like Kubernetes.

These requirements represent a significant departure from how operators have architected their networks in the past. And, they raise new considerations for NEPs developing physical and virtual NFs for 5G networks. First, while support for model-driven programmability might have been optional in the past, it is becoming an essential enabler of 5G network management. The 3GPP standard assumes a service-oriented management architecture, where each subnet/domain is programmable. It also assumes an open architecture where NFs from multiple vendors can interoperate as part of a service. The only viable way to integrate NFs in this model—with each other and with existing management systems and orchestrators—is via standards-based, machine-readable and programmable management interfaces.

4.1.2. Programmable Interfaces in 5G Automation

Network operators have neither the time nor skills to use a wide range of proprietary interfaces to manage all the vendors in their network. Nor do they want to deal with the complexity of using scripts for automation, which end up being tightly coupled with the management logic for each NF. Rather, operators want to minimize (ideally eliminate) the need for human-to-machine intervention and adopt machine-readable programmable interfaces across the network.

The most effective way to enable this is via NFs that support standardized YANG data models (defined in [RFC 7950](#)) and the NETCONF management protocol ([RFC 6241](#)). While REST interfaces can be used, they are not standardized, and therefore can become almost as complex as vendor-specific CLIs in multivendor networks. More significantly, REST does not support management features that will be critical in automated and self-healing networks, such as transactions, pre-commit validation of configurations, and rollbacks. (See the following sections for a detailed discussion of these features.)

For these reasons, traditional REST APIs cannot meet operators' requirements for enabling programmable networks. However, even REST's more modern, standardized heir, RESTCONF ([RFC 8040](#)), does not support the full range of features that operators will want for end-to-end programmability. Alternatively, NETCONF and YANG are designed specifically to enable standardized network programmability. NF developers will find that they offer a superior foundation for enabling end-to-end management and automation in their customers' networks.

The 5G specification also implies support for intent-based management. To manage complexity, operators should be able to express high-level intent and have the network automatically execute it. In this model, high-level orchestrators are not burdened with managing every procedural step for every NF, in every domain, contributing to an end-to-end service. Rather, the end-to-end orchestrator focuses on provisioning and assuring services according to high-level intent, while the granular procedural logic for configuring individual NFs gets pushed down to southbound orchestrators/controllers. (See the following sections for additional details.)

To meet these diverse management requirements, operators need NFs that can be configured via machine-readable data models such as YANG and standardized, model-driven management protocols like NETCONF.

4.2. Programmability in 5G Networks: Overview

Automation is essential to manage sprawling, heterogeneous multi-domain/multivendor networks efficiently. Operators therefore increasingly expect and demand support for standards-based programmable interfaces in their NFs, both physical and virtual. If an NF is not programmable via an automated, machine-readable management framework, it likely is not an option for a 5G network.

But what, specifically does programmability entail? First, the NF must provide a machine-to-machine management interface, not just a human-to-machine interface like CLI. It should expose data in a standardized format, such as YANG. And, it should define a set of standard operations (that is, a protocol), so that it can be configured via third-party controllers and orchestrators.

Beyond these basic principles, however, NFs that participate in 5G networks and services must meet several other programmability requirements, including support for transactions, intent-based networking,

Comparing NETCONF, RESTCONF, and REST

NETCONF was designed to provide a standardized programmable interface for configuring network devices. As the concept of programmability extended to the world of enterprise IT, however, enterprise network teams expressed a desire for alternate approaches based on REST APIs. Many had little background in NETCONF and incorrectly envisioned a steep learning curve to use NETCONF libraries and tools effectively. They were, however, intimately familiar with using REST APIs to programmatically access remote web services. Thus, RESTCONF ([RFC 8040](#)) was born.

RESTCONF standardizes the use of REST techniques to manipulate the data described in YANG data models—the same data used by NETCONF to configure network elements. Unlike NETCONF, however, RESTCONF runs over HTTP, using familiar HTTP commands to make changes to network elements. In this way, it allows enterprise IT programmers to begin using YANG data models to automate their environments using the REST-based tools and knowledge they have today.

4.3. Key Concepts in Programmability: Transactions

When configuring physical and virtual network elements in dynamic operator environments, many things can go wrong. Software or hardware could fail in the middle of a write operation. Applications could crash or unexpectedly get cut off from the database. Multiple clients attempting to write to the database simultaneously could overwrite each other, or a client could read partially updated data that doesn't make sense.

To deliver reliable 5G services under SLAs, the network must be able to deal with these and other faults and ensure they don't cause a catastrophic failure. For decades, transactions have been the developer's mechanism of choice to accomplish this. A transaction allows an application to group several reads and writes together into a logical unit. Conceptually, the application executes all reads and writes in the transaction as one operation. Either the entire transaction succeeds (commit) or it fails (abort, rollback). If the transaction fails, the application can safely retry.

The safety guarantees provided by transactions can be described by the acronym "ACID," which stands for:

- **Atomicity**, or the ability to abort a transaction on error and have all writes from that transaction discarded
- **Consistency**, meaning that to management systems, all actions within a transaction are viewed as instantaneous
- **Isolation**, meaning that concurrently executing transactions can't interfere with each other
- **Durability**, which implies that once a transaction has committed successfully, any data it has written will not be forgotten, even if there is a hardware fault or database crash

4.3.2. Transaction Support in Programmable Interfaces

Transactions make configuration management far more robust and coherent for network operators, while reducing out-of-sync errors, a major problem historically. Despite these advantages, however, many NF management interfaces still lack transaction support. Even rarer in NFs today: support for networkwide transactions. Networkwide transactions let operators apply a service configuration across multiple NFs in a single transaction. If any NF configuration fails, they automatically revert, greatly reducing the effort that would otherwise be needed to recover from such errors, which can span hundreds or thousands of network elements.

In designing products for multi-tiered 5G architectures, developers of physical and virtual NFs should consider where and how transactions will be used. Transactions may not seem important at the level of the overall orchestrator concerned with the end-to-end service. They are extremely valuable, however, for the domain-specific controllers underneath, such as those orchestrating NFs in the transport or radio network. There, the ability to use transactions offers major benefits for service-based management frameworks—especially for critical operations and scenarios where changes must be executed across multiple NFs with minimal risk of failure.

RESTCONF can provide a valuable option for implementing programmability in certain use cases, but it is not a NETCONF replacement. In fact, it lacks several key capabilities that service providers rely on to automate their networks. Network equipment vendors developing products applicable to a wide range of use cases should therefore make sure they understand what RESTCONF can and cannot do. For details, see the Tail-f white paper [Inside RESTCONF](#).

By using transactions, operators can substantially reduce the risk of downtime due to configuration errors. They can provision services much more quickly, accelerating service activation and time-to-revenues. Crucially for complex 5G services and slices, transactions also enable more robust, extensive automation. For these reasons, operators are increasingly asking NF providers to support transactional interfaces for managing configurations. In fact, this is one of the main reasons why many operators now ask network device vendors to support NETCONF, rather than only supporting REST or RESTCONF.

(For an in-depth discussion of transactions in programmable networks, see the Tail-f white paper [Managing Distributed Systems Using NETCONF and RESTCONF Transactions](#).)

4.4. Key Concepts in Programmability: Intent-Based Networking

As noted, 5G architectures assume support for the concept of intent-based networking, or IBN. But what, exactly, does IBN entail? Broadly, the term “intent” describes the end state of the system that a user wants to achieve. In the context of network programmability, this implies a configuration that has been tested and validated against specific business and technical requirements that the network service must fulfill.

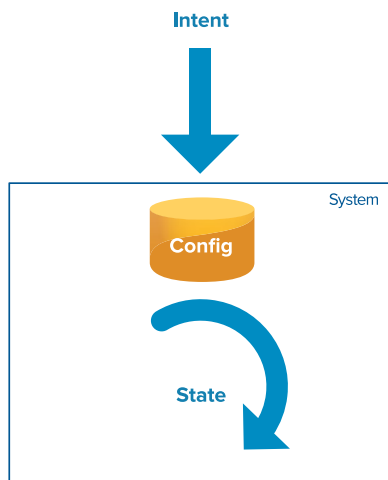


Figure 4. Conceptual View of Intent-Based Networking

With IBN, operators can use a declarative approach to automating network configurations. That is, the front-end user can focus only on the well-defined, high-level end state they wish to achieve, without worrying about the myriad procedural steps needed to configure each node or NF in the end-to-end service to achieve it. The network deals with all that complexity autonomously, abstracting it away from the front-end user and higher-level management systems.

The concept of IBN is extremely powerful for operators, especially in managing dynamic 5G networks. It can be applied anywhere in the environment where intent can be defined and where the network can be automated to fulfill that intent. And, it provides a crucial abstraction layer on top of complex 5G hardware and software infrastructure, as well as the domain-specific controllers participating in end-to-end services. This is especially important for network slicing, where the network may need to spawn many NFs across multiple domains and vendors as part of a slice. Without IBN, operators cannot automate this process—and cannot respond quickly to changing customer needs and consumption patterns.

4.4.2. Service Orchestration for IBN

Intent-based networking requires the ability to automate network state changes for services spanning multiple vendors and domains. As such, each physical or virtual network element must support a programmable interface towards northbound orchestrators or controllers. In earlier, simpler architectures, orchestration could be limited to individual domains or vendors. To manage dynamic services and slices in heterogeneous 5G networks, however, an overarching automation model is needed.

To enable this, operators will use a multi-tiered architecture, with a service orchestrator providing end-to-end lifecycle management and configuration. This top-level orchestrator coordinates management tasks among lower-level controllers and southbound NFs, bridging the different domains and vendors involved in a 5G service. It also plays a crucial role in network slicing, stitching together virtualized resources across subnets into an end-to-end slice and exposing management APIs to northbound systems.

The use of a top-level, end-to-end orchestrator is therefore a core enabler of IBN. It provides operators with a single front-end interface with a single API towards the entire network. It also facilitates closed-loop service assurance, ultimately enabling self-healing networks. For these reasons, 3GPP has introduced new management functions in the 5G specification to facilitate intent-based orchestration of NFs. (See the section “Network Slice Orchestration” later in this paper.)

4.4.3. Implementing Intent-Based Networking

Intent-based interfaces can be contrasted with action- or workflow-based interfaces, where configurations are mostly manual and proprietary. In these legacy models, the ordering logic and state of the system dictate how a network operator can interact with it. As a result, implementing a state change requires significant time and manual effort, resulting in frequent errors, downtime, and higher costs for operators.

This model will not work in 5G networks, where operators must be able to quickly test, deploy, and scale complex end-to-end services and slices. In the near future, operators will likely not use proprietary CLIs or traditional human-to-machine interfaces at all, except when troubleshooting special cases. Even then, they will seek to limit human interaction as much as possible, leaving any operation that requires configuration changes to the orchestrator.

To enable the IBN capabilities that 5G networks require, NF developers should provide intent-based interfaces that support:

- **Fast deployment:** The network should be able to receive intent from one location and quickly, automatically instantiate the service or change.
- **Transactions:** The network should be able to fulfill the desired intent across every NF in the service path using transactional mechanisms.
- **Idempotency:** Multiple requests with the same intent (that is, duplicate requests for the same configuration) should have no additional effect.
- **State-independence:** Management systems should be able to always execute intent, regardless of the state of the network.
- **Intent integrity:** The network should never modify the received intent.

4.4.4. Standardized Data Models and Interfaces in IBN

To accelerate delivery of new services and network changes, operators want their network teams, even their customers, to be able to define business-level intent and have the top-level orchestrator automatically drive those changes in the network. This is not possible if the service includes physical or virtual NFs that rely on legacy command-driven interfaces. This is because command-driven interfaces:

- Require explicit commands to move between states
- Create scenarios where the correct command depends on the current state of the network
- Require commands to be issued in sequence
- Require inefficient workflows or runbooks to automate tasks
- Do not describe the NF's data and semantics—a requirement for orchestrators interacting with multiple domains and vendors

Command-driven interfaces also prevent operators from implementing closed-loop automation in heterogeneous networks. For networks and services to become self-healing, an analytics engine must be able to detect problems or SLA violations, transmit this information to the service orchestrator, and have the system automatically take corrective action.

For all these reasons, operators implementing IBN increasingly require every NF in the network to support standardized machine-to-machine interfaces, ideally via NETCONF. NETCONF provides a mature, consistent interface for executing high-level intent across multiple vendors and domains. Unlike legacy interfaces, it also comes with an excellent standardized modeling language, YANG, to describe what matters in the network: data, including semantics.

4.5. Key Concepts in Programmability: Model-Driven Telemetry

Closed-loop automation requires the network to monitor the health and performance of network services and NFs, so it can automatically take corrective actions when needed. To enable this, every NF in a 5G architecture should support telemetry. That is, distributed NFs should continuously stream data about network statistics and events to a centralized analytics engine.

Historically, operators collected network statistics via SNMP polling. SNMP, however, does not provide the real-time visibility or granularity required in dynamic 5G environments. It provides only a subset of the data needed to monitor the health of 5G services and SLAs. SNMP also provides data intermittently, with long intervals between transmissions. As a result, it gives operators only a snapshot in time, not a real-time view into the health of NFs and services.

To enable closed-loop assurance for 5G services, operators are increasingly turning to model-driven telemetry. If NF providers want their products to participate in self-healing 5G networks, they should ensure that NFs can model telemetry data in YANG and stream it via NETCONF.

5. Management and Orchestration Across 5G Network Domains

5.1. Managing the Next-Generation RAN

The 5G specification for next-generation radio access networks (NG-RANs) introduces a new technology model for base stations—the network equipment that transmits and receives wireless communications between user devices and the mobile network. 3GPP defines this new base station as the gNB (or alternately, gNodeB). Figure 5 below shows the 5G NR interface with the core network.

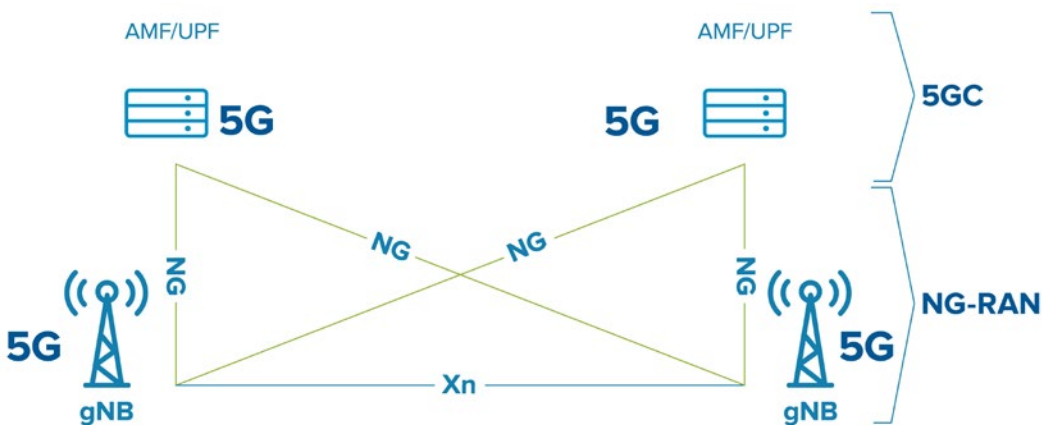


Figure 5. 5G New Radio Interfacing with Core Network

In previous-generation architectures, RAN base stations were tightly coupled to radios, typically provided by the same vendor. However, actual utilization of radio and base station resources often did not align. This created a number of inefficiencies for operators. First, they had to build out base station resources for peak capacity—even though the network only required those resources a fraction of the time. This was especially costly in dense areas, where many more base stations were needed. Additionally, previous-generation base stations were not always resource-efficient. As noted, this can create performance issues in 5G architectures, where the same base station might be supporting applications for multiple network slices simultaneously.

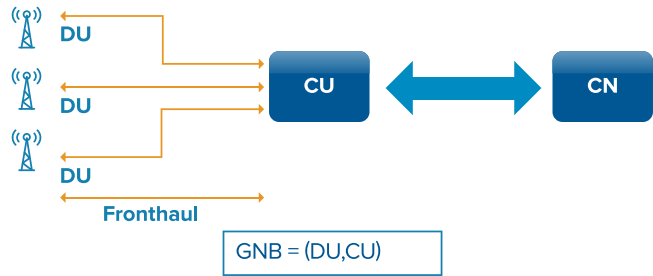
To address these inefficiencies, leading operators, NEPs, and other industry stakeholders came together to form the [O-RAN Alliance](#). This effort introduced a new model for more open, intelligent, and operator-defined RAN architectures and interfaces. The O-RAN architecture opens up RAN base stations to support heterogeneous multi-vendor deployments, virtualized resources, and support for standardized programmable interfaces.

This next-generation RAN architecture gives operators far more flexibility and efficiency—especially in dense environments and mass-scale IoT deployments. However, it also adds new layers of complexity to the management of RAN resources. As a result, NFs deployed in NG-RANs must now support open, programmable interfaces and dynamic management of virtualized resources.

5.1.2. Virtualizing and Automating the RAN

In a next-generation RAN, some base station functions for certain applications can be centralized (depicted in Figure 6 as centralized units, or CU), while in other scenarios, they can be distributed (DU). Operators have multiple options for splitting base station functions between centralized or distributed baseband units (BBUs), depending on the use case. Here, the gNB is a logical node that can be disaggregated into separate functional components, which can then be virtualized and deployed in different locations as required.

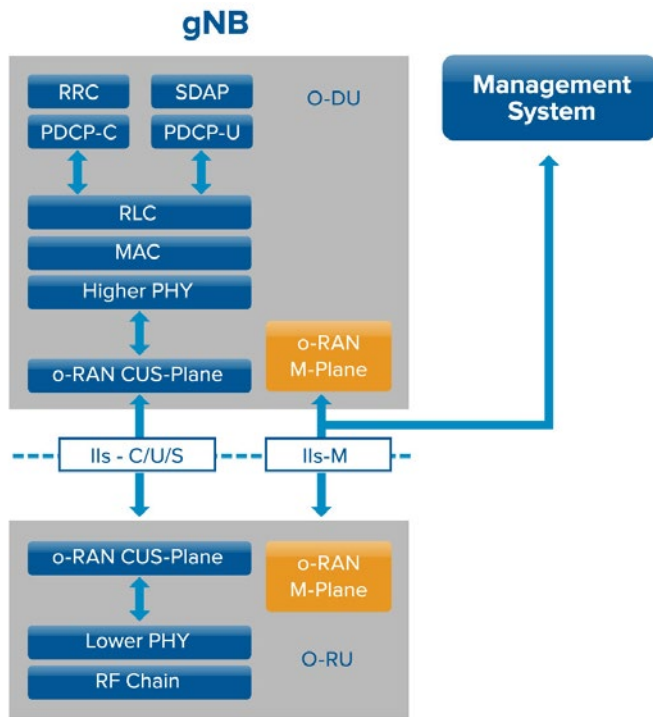
Figure 6. Distributing Virtualized Base Station Functions



This new RAN flexibility allows operators to incorporate equipment from multiple vendors into the radio network. It also enables them to support many new deployment scenarios, optimizing radio resources for the specific cost and/or scalability requirements of different locations and use cases. To take advantage of this flexibility, however, open RAN architectures require standardized, model-driven management. 5G radios and BBUs should be programmable, just like any other network element in an open multivendor network. Here again, the NETCONF management protocol and YANG data models are operators' preferred choices for enabling programmability in NG-RAN NFs.

Figure 7 shows an O-RAN architecture that enables separation of virtualized control and user plane functions, and management via standardized programmable interfaces.

Figure 7. High-Level View of O-RAN Architecture with CUPS



The presence of management intelligence (M-Plane) in both the distributed unit and the radio unit is what enables flexible management in the next-generation RAN. It allows operators to directly manage radio units—even from multiple vendors—using programmable NETCONF/YANG interfaces. (To review the YANG modules defined for the O-RAN Reference Architecture in the Fronthaul Interface Specification 1.0, see the [ONAP Developer Wiki](#).) Without support for standard management interfaces in RAN NFs, it would be very difficult for operators to integrate RAN products from multiple vendors with each other, as well as with the management and orchestration systems that automate end-to-end 5G services.

5.2. Managing 5G Core Networks

The 3GPP 5G specification introduces a core network model that looks very different from legacy architectures. To support more dynamic 5G services and slices, it defines a core where all core network functions are virtualized. Today, many operators are using the conventional ETSI NFV MANO architecture, with NFs running inside VMs, to accomplish this. Increasingly, however, they will adopt cloud-native NFs running in containers on bare metal, using container orchestration tools like Kubernetes. This will allow them to eliminate the resource inefficiencies and performance penalties associated with VMs and hypervisors, enabling greater flexibility, speed, and automation.

To support these efforts, NFs designed for 5G core networks must be programmable. Specifically, they must expose an API so that higher-level service orchestrators can push down configurations for services and slices, without requiring human intervention. Here again, operators prefer standards-based, model-driven programmable interfaces. Increasingly, they are choosing NFs that support NETCONF and YANG data models as the foundation for a more open, programmable 5G core.

5.3. Managing 5G Transport Networks

The 3GPP 5G specification focuses primarily on the NG-RAN and core, as those domains undergo the most significant architectural and management changes to support 5G services. However, NF developers should not take this to mean that 5G networks do not require standards-based programmability in the transport layer as well.

In fact, operators do require programmable transport networks to enable automated provisioning and closed-loop assurance for end-to-end services and slices. They are adopting new techniques to enable this, such as segment routing-based traffic engineering (SR-TE), which offers a more modern and efficient traffic management model compared to MPLS. The IETF defined the segment routing architecture that enables this in [RFC 8402](#).

An in-depth discussion of SR-TE is outside the scope of this paper. However, just as with other domains, developers creating NFs for the transport layer should ensure that their products expose a standardized programmable interface such as NETCONF towards northbound orchestrators. And, they should ensure that transport-layer NFs can be abstracted and automated via standardized YANG data models.

6. Network Slice Orchestration

6.1. A New Model for Managing Virtualized Networks

This paper has repeatedly discussed network slicing, both as a core enabler of new operator business models and a key driver of the need for end-to-end programmability in 5G networks. Developers of physical and virtual NFs that will participate in 5G network slices should understand what this process entails from the operator’s perspective, and how those requirements impact the design of their products.

From an architectural perspective, a 5G network slice is an end-to-end logical network that overlays the hardware and software infrastructure, spanning multiple network domains: core, access, transport, and cloud. Inherently then, provisioning and managing network slices requires the ability to control the many different resources within each domain. Operators must be able to orchestrate each domain-specific “slice subnet,” ultimately stitching all resources together to create an end-to-end slice.

To enable this degree of end-to-end programmability, operators are adopting a multi-tiered orchestration and control model. This model depends on all components at each layer of the architecture—every NF and orchestrator/controller within every domain—exposing standardized programmable interfaces.

6.2. Fundamentals of Network Slice Orchestration

In a multi-tiered 5G slicing architecture, a top-layer orchestrator provides a single point of control—and a single intent-driven API—towards the entire network. This top-layer orchestrator interacts with the domain-specific orchestrators/controllers in the layers underneath, which in turn interact with the network resources in that domain. It is the autonomous, real-time interaction between these layers and domains that enables orchestration of an end-to-end slice. Figure 8 provides a high-level illustration of this architecture.

Recursive Model

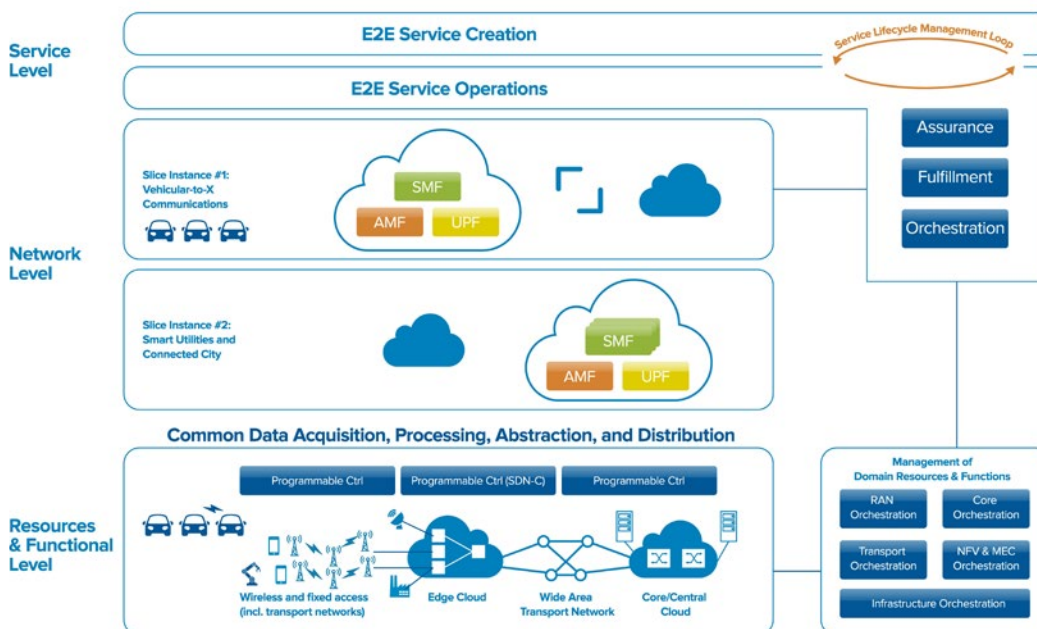


Figure 8. High-Level View of Network Slicing Architecture with Assurance

As noted previously, an end-to-end orchestrator must be able to address not only the multiple domains and resources within a single operator’s infrastructure, but in some circumstances, heterogeneous domains that include interconnected resources from multiple service providers. To enable SLAs for this kind of cross-provider network slice, the end-to-end orchestrator must be able to interact with external domain-specific orchestrators using standardized programmable interfaces. Therefore, each operator participating in the slice must use domain-specific orchestrators or controllers at each layer of the architecture and expose a service-based management API towards the physical and virtual resources it controls.

The best way to implement a service-based architecture for 5G slice orchestration is via standardized, model-driven APIs. Here, the network functions or controllers/orchestrators (termed “managed objects,” or MOs in the 3GPP specification) expose a programmable interface using standard protocols such as NETCONF. And, they use data models created in a standardized modeling language such as YANG. It is with these hierarchical building blocks that the top-layer orchestrator can dynamically stitch together end-to-end network slices, using intent-based networking and service modeling at every layer.

6.3. Intent-Based Slice Orchestration

Orchestration of end-to-end network slices uses an intent-based model. As noted, IBN effectively mandates the use of standardized data models to describe and abstract NFs in heterogeneous 5G networks. If NFs provide only a proprietary “description of operations,” they will require special handling—usually via human intervention. Which means they cannot be quickly or efficiently scaled in heterogenous environments, and they cannot participate in automated 5G network slices.

Figure 9 illustrates how standardized YANG data models can enable intent-based network slicing. Here, the YANG data model represents a contract that the managed objects (that is, southbound NFs or controllers/orchestrators) expose via a front-end programmable interface such as NETCONF. Intent parameters are modelled in YANG and exposed by the top-level service orchestrator, which manages service-level data, as well as the MOs managing NF- or domain-level data. Each domain exposes its resources (that is, its NFs or controllers) via a standards-based management API.

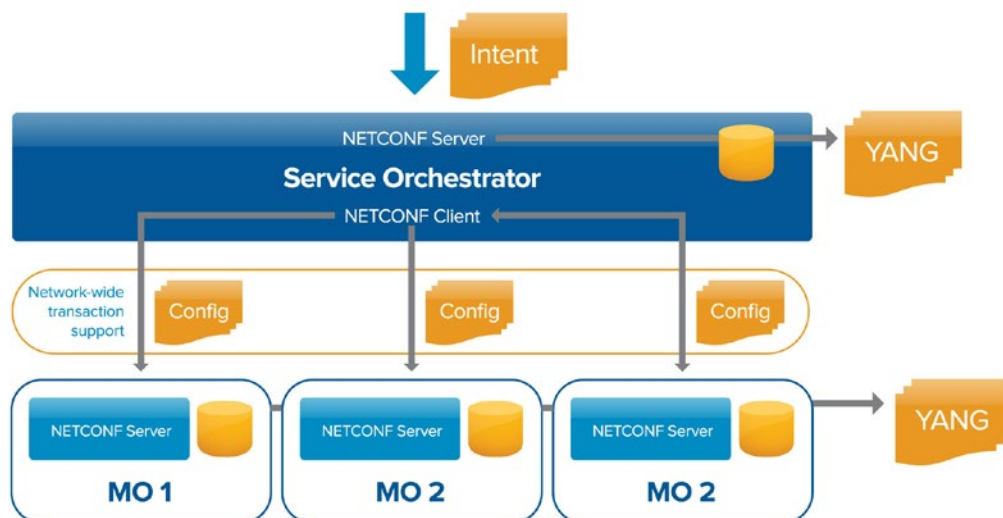


Figure 9. IBN Using YANG Data Models and NETCONF

Effectively, this model turns the management and orchestration of network slices into an entirely software-based operation, using open and standardized APIs. It gives operators the flexibility and speed needed to integrate the many heterogeneous resources participating in an end-to-end service and enables fully automated, end-to-end network slicing.

6.3.2. Inside a Slice Orchestrator

The fundamental advantage of network slicing is the ability to deliver virtual networks tuned for very different application requirements over the same physical infrastructure. Figure 10 details what this can look like in a 5G core.

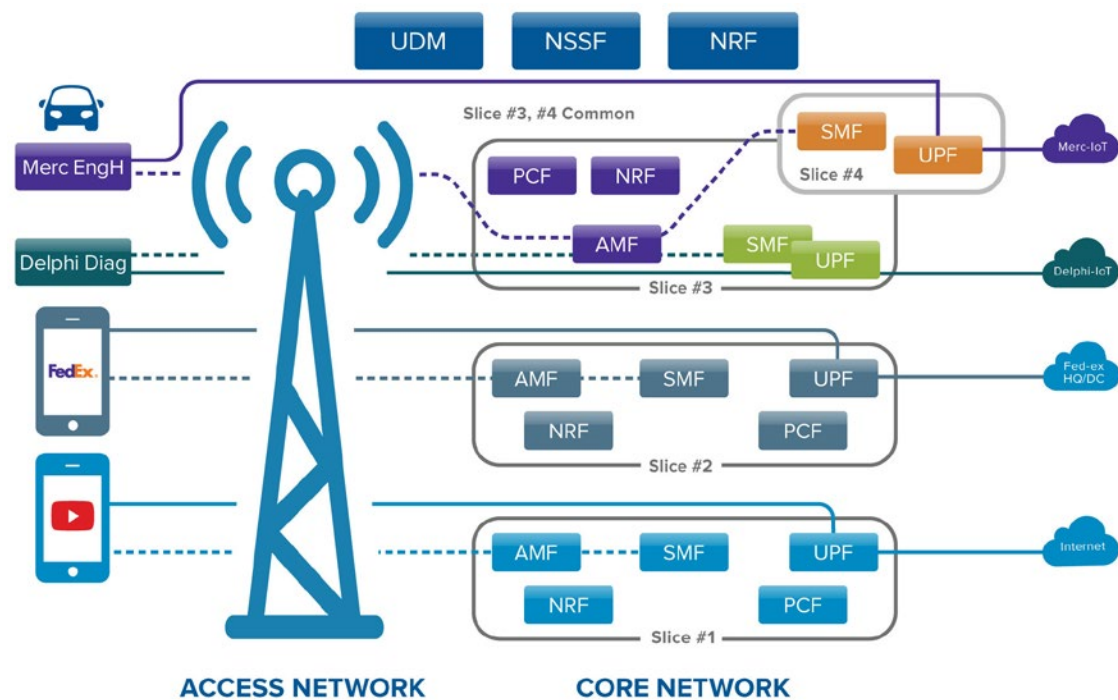


Figure 10. Network Slices Sharing Resources in a 5G Core Network

Here, the core network supports multiple slices for distinct use cases, even specific customers, while sharing underlying infrastructure resources. As depicted, the same Access and Mobility Management Function (AMF) resource can be shared among Slice #3 (green) and Slice #4 (purple). Meanwhile, each slice maintains its own separate Session Management Function (SMF) and User Plane Function (UPF). In this way, the operator can make more efficient use of resources when delivering network services tuned for IoT applications, while maintaining separate slices and SLAs for different customers.

To orchestrate and manage slices in this manner, operators must use a multi-tiered management architecture that employs standardized data models and exposes standard APIs at each layer. Figure 11 provides a high-level illustration of such an architecture.

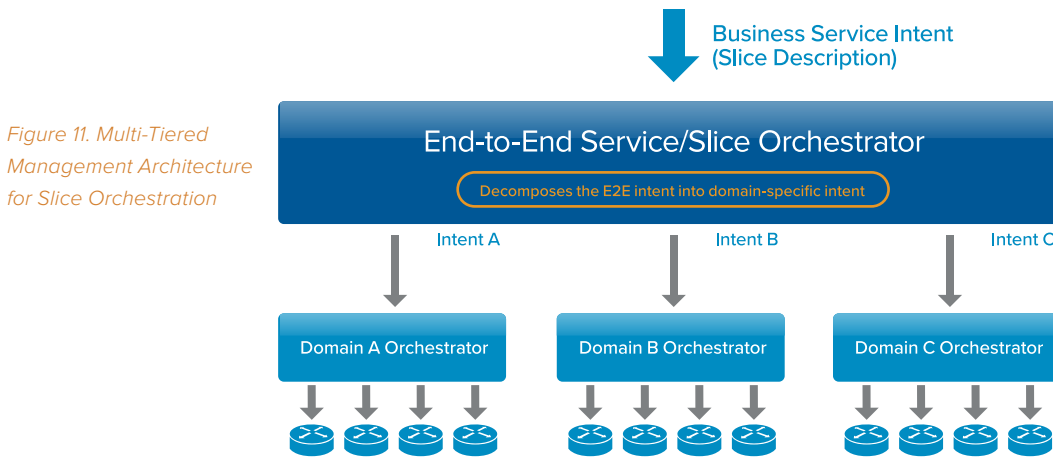


Figure 11. Multi-Tiered Management Architecture for Slice Orchestration

At the top is the BSS/OSS system that describes the highest-level service intent—that is, the business-level description of the service that the slice will provide. The OSS/BSS consumes APIs exposed by the end-to-end slice orchestrator below, which in turn consumes APIs exposed by the different domain-specific orchestrators and controllers southbound. In this way, the end-to-end slice orchestrator receives the business-level intent and decomposes it into configuration parameters (that is, domain-specific intents) for each domain.

When creating a slice, the end-to-end service orchestrator onboards each NF as a resource supporting the slice. However, the orchestrator’s responsibilities don’t end once the slice is activated. The orchestrator may push out updated configurations on an ongoing basis, at scale, to ensure that NFs and slices are performing in accordance with customer SLAs or to make changes to a running network function. Here, we see the beginning of true closed-loop automation (Figure 12).

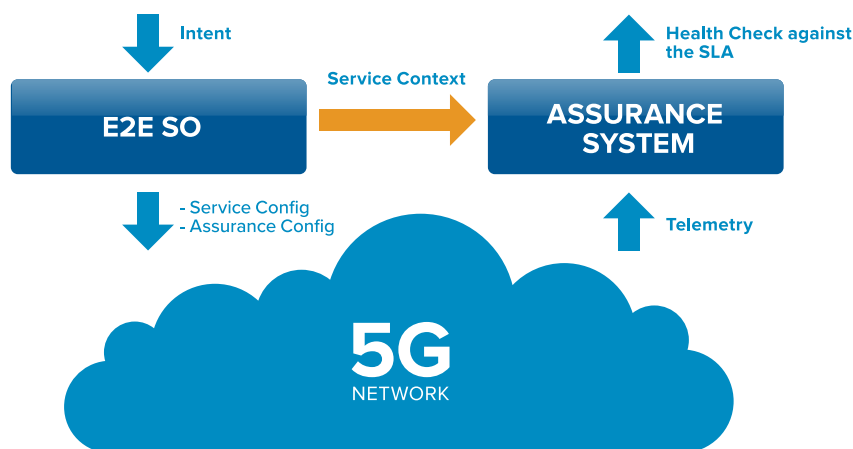


Figure 12. Autonomous Interaction Between the End-to-End Orchestrator and Assurance System

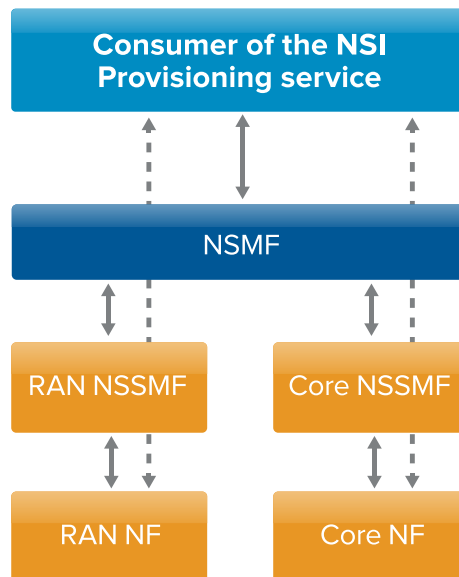
6.3.3. Mapping Intent for 5G Slices

3GPP introduced new network management functions in the 5G specification to make intent-based slice orchestration possible. These include:

- **Network Slice Management Function (NSMF):** NSMF is the 5G component that maps business intent to the end-to-end slice configuration. This includes the coordinating the multiple slice subnet configurations destined for each domain.
- **Network Slice Subnet Management Function (NSSMF):** This function, specific to each domain, is responsible for instantiating and activating configurations within each subnet slice.

These new functions facilitate the multi-tiered, intent-based architecture that the 3GPP 5G specification envisions, as illustrated in Figure 13.

Figure 13. Multi-Tiered, Intent-Based Architecture for 5G Network



At the RAN level, NSSMF sends instantiation requests for a subnet slice to an NFVO or cloud-native orchestrator. This domain-level orchestrator interfaces with the end-to-end slice orchestrator northbound and the radio network management plane southbound. In the coming years, operators will likely replace this framework with a cloud-native RAN architecture that relies on a container-based orchestrator, such as Kubernetes.

In core networks, cloud-native models may already be in use, with multiple virtualized network functions deployed via a microservices-based architecture. Here, the NSSMF coordinates with an NFVO (or increasingly, Kubernetes) to orchestrate subnet slices.

6.3.4. Lifecycle Management of Network Slices

In a 5G network, the end-to-end service orchestrator is responsible for managing and automating the full lifecycle of each slice and its constituent resources. This includes:

- **Mapping business intent to domain-specific configurations.** This includes coordinating the necessary network resources for the slice.
- **Instantiating the infrastructure resources.** The orchestrator must coordinate slice subnets and the Day-0 configuration of their constituent NFs, whether using existing NFs or instantiating new ones. It delegates this task to the NFVO/orchestrator/controller in each domain. For example, the core and access network will each have its own domain-specific controller/orchestrator to manage the lifecycle of its underlying network resources.
- **Provisioning.** The top-layer service orchestrator is ultimately responsible for stitching together the network resources that compose the slice, implementing their Day-1 configurations, and handling errors. To do this, the orchestrator coordinates provisioning tasks across the different domain-level controllers or directly with the instantiated NFs. (In some cases, the orchestrator may perform stitching/interconnection of slice subnets during the instantiation phase as well.)
- **Validation and activation.** Here, the orchestrator validates the configurations for the slice and, when applicable, activates the service. (Operators may validate network slices without activating them, so that they can be quickly activated in the future as needed.)
- **Performance monitoring and fault management.** The orchestrator coordinates monitoring of the slice, typically with the help of an end-to-end assurance system. This can entail aggregating data from multiple distributed network applications that are collecting performance data at the edge (for example, in Smart Cities or mass-scale IoT deployments). This data can then be pushed up to an assurance engine that handles service assurance for every slice in the network, enabling closed-loop assurance.
- **Decommission.** As needed, the end-to-end orchestrator deallocates resources that are no longer needed (preserving shared resources being used by other slices) and removes the network slice.

For an orchestrator to coordinate all these operations—that is, for an operator to implement end-to-end intent-based orchestration of network slices—every NF participating in the service must support IBN for full lifecycle management. Once again, best practices for programmability dictate that NFs expose a standardized API such as NETCONF and provide standardized data models that allow for full lifecycle management via an orchestrator. If an NF requires CLI or other proprietary models for some lifecycle functions, it cannot be automated as part of end-to-end slice orchestration.

7. Conclusion

5G promises amazing new capabilities for enterprise customers and lucrative new revenue streams for service providers. To fulfill this promise, however, operators need the help of the equipment providers developing NFs for their networks.

To enable the diverse range of applications that will run over 5G networks—and allow operators to automate and accelerate their delivery—the network must be built for end-to-end programmability. At every level of the network, NFs must support model-driven orchestration via standardized data models. And, every NF must expose a standard API. For providers of physical and virtual NFs that will participate in dynamic 5G networks, YANG data models and NETCONF are the most effective means to enable this.

By building support for standardized, model-driven management, NF providers can give operators the building blocks they need to automate the delivery and assurance of 5G services end to end. They can help operators become much faster and more efficient, so they can bring the full range of new 5G experiences to their customers.

For more information about programmability, NETCONF, and YANG in general as well as how our ConfD product can be used to enable this functionality in your network function, please, visit <https://www.tail-f.com>

tail-f a Cisco
company

www.tail-f.com
info@tail-f.com

Corporate Headquarters

Sveavagen 25
111 34 Stockholm
Sweden
+46 8 21 37 40