Cisco Digital Network Architecture

Intent-based Networking for the Enterprise

Tim Szigeti
David Zacks
Matthias Falkner
Simone Arena
Foreword by **Scott Harrell**

ciscopress.com

Cisco Digital Network Architecture

Intent-based Networking in the Enterprise

Tim Szigeti, CCIE No. 9794
David Zacks
Matthias Falkner
Simone Arena

Cisco Press

While examples certainly exist of "cross-overs" between these three broad categories—a common example being the WAN backbone of a large, multi-departmental enterprise organization that may actually begin to resemble more of a service provider deployment model—the preceding major categories define three distinct and unique sets of use cases, each corresponding to a particular associated set of network requirements around (among other variables) speeds and feeds, resiliency and fault tolerance, expansion requirements, Layer 2 and Layer 3 traffic handling, and more.

Accordingly, Cisco's network operating system software strategy evolved over the years to address these three unique markets as follows:

- **Enterprise:** The Enterprise market is largely served by Cisco IOS, which has now evolved into Cisco Open IOS XE (more details following). Open IOS XE (often abbreviated to simply IOS XE) is the direct evolution of Cisco IOS, building on this legacy and set of capabilities while evolving to support a next generation of enterprise-focused network designs, solutions, and outcomes. Given the focus of this text on examining Cisco Digital Network Architecture (DNA) and its impact on the evolution of the enterprise network, special attention is paid here to Cisco IOS XE and its capabilities and evolution. Open IOS XE is called "Open" because it embraces, and enables within the enterprise network, several attributes of open networking, including YANG data models, open industry-standard interfaces such as NETCONF and RESTCONF, and the ability to deploy containerized applications onto distributed network devices.
- **Service Provider (SP):** While Cisco IOS (and now, IOS XE) is certainly used in various places in SP networks, Cisco's SP-focused network operating system software is IOS XR. Designed with the unique scale, performance, and functionality needs of SPs in mind, IOS XR powers some of the largest and most complex SP networks in the world. Built from the ground up to provide the capabilities that SPs demand most, IOS XR sheds many of the diverse protocols and capabilities that need to be supported in enterprise environments for a more focused set of capabilities that are designed to service the world's largest networks. While some very large enterprises (especially those resembling SPs in some aspects, as previously mentioned) also deploy IOS XR–based platforms, the primary focus of IOS XR remains the core SP market.
- **Data Center (DC):** The demanding environment of the modern data center drives several unique requirements. These include (but are not limited to) widespread use of Layer 2 for server-to-server intra-DC connectivity, very rapid network failover and recovery requirements, and the need to accommodate very high (and ever-growing) performance needs, typically pushing the boundaries of what is possible for network connections. To meet these demands, Cisco created NX-OS, focused on the needs of the modern—and ever-evolving—data center. Because both enterprises and service providers deploy their own DCs, or host DCs for others, NX-OS is used in both types of deployments, within the data center and focused on the outcomes that data centers require.

The evolution of these various operating systems—IOS XE, IOS XR, and NX-OS—is illustrated in Figure 8-2.

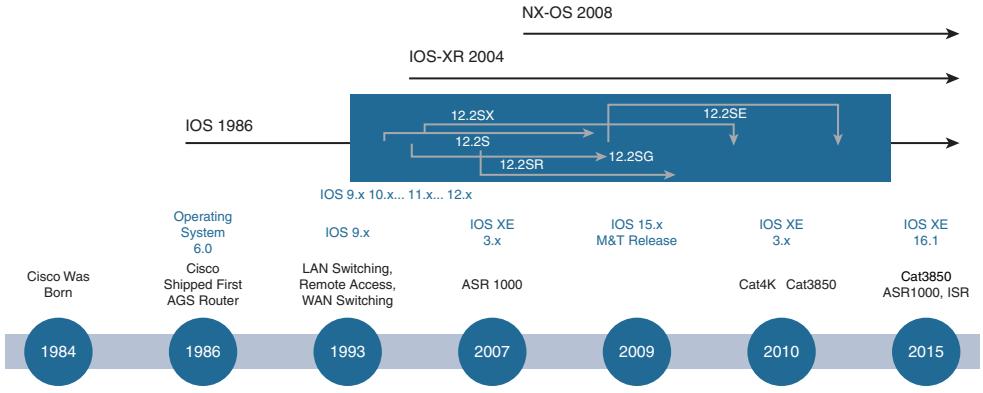

Figure 8-2 *Network Operating Systems—Origins and Evolution*

The Evolution of Cisco IOS to IOS XE

While Cisco IOS has served many tens of thousands of customers worldwide well for over three decades, an evolution of the basic architecture of Cisco IOS was undertaken several years ago to help IOS continue to adapt to an every-changing world, and an ever-evolving set of network requirements.

What factors came into play as the evolution of Cisco IOS was undertaken?

First, the need to provide a level of modularity for IOS was desirable. Traditionally, Cisco IOS was implemented and operated as a single, monolithic code base, with direct access to the underlying hardware of the platform on which it ran. While this model evolved over the years (for example, to address the separation of data plane and control plane detailed previously in this chapter) the basic structure of Cisco IOS was maintained—a single software code base, running on a device CPU, with direct access to the underlying network hardware.

For most enterprise network tasks, this worked very well over the years. Nevertheless, changes in network and device operational practices made some modifications in Cisco IOS desirable. As well, the ever-changing "state of the art of what's possible" made such changes more practical to realize than they were in the past. As well, the continuing evolution of network designs made alterations to what IOS needed to be capable of more urgent than had previously been the case.

Taken together, all of these elements created the necessity for Cisco IOS to evolve. IOS XE is the result of that evolution.

Cisco IOS XE in a Nutshell

Fundamental shifts have occurred in the enterprise networking landscape. These include the move from networks that only had to concern themselves with moving data, voice, and video, to more modern networks that have to encompass mobility, the deployment of ever-increasing types and numbers of IoT devices and applications, and the accommodation of cloud capabilities, all while also addressing the manifold security requirements of the modern enterprise. Taken together, these constitute a new era of networking, as illustrated in Figure 8-3.

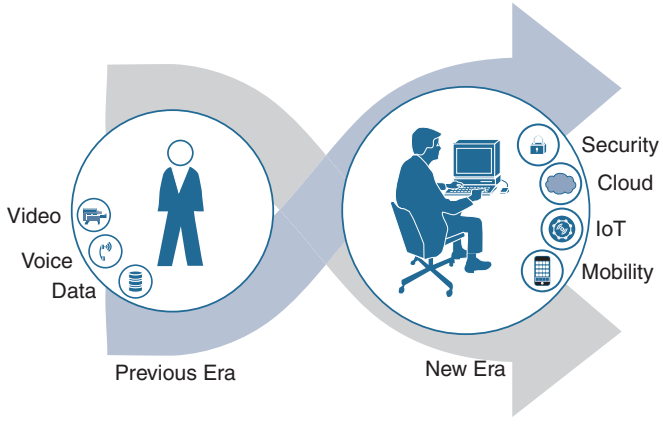

Figure 8-3 *New Era of Networking*

This is the era, and the trends, that Cisco IOS XE was created to address.

IOS XE set out to evolve the IOS infrastructure. The fact that this is an evolution is critical. There are tens of thousands of network professionals that know Cisco IOS inside and out—and many more networks that they support that depend on Cisco IOS, and the features and capabilities it provides, every day. Thus, one of the first goals of IOS XE was to provide an evolutionary path both for these networks and for the network managers and operators that implement and support them, while allowing the introduction of network technologies, new capabilities, and new network deployment modes and operating paradigms.

To evolve IOS to the next level of functionality and capability, it was important to set IOS XE on a strong foundation. IOS XE is implemented on top of a Linux kernel, providing a robust foundation that allows for extensibility, supports the use of multicore CPU processing, and sets the stage for hosting of additional applications and containers on the same platform that operates the core IOS XE set of processes and functions.

On top of that foundation, the core IOS XE functions operate within the context of IOSd, the IOS daemon that runs the core set of networking capabilities that Cisco IOS users and network managers depend on. Over the many years during which Cisco IOS evolved, IOS implemented hundreds of different capabilities, functions, and protocols that network managers use to create, define, operate, and manage their networks. You name a function provided by IOS, and somewhere in the world there is a network that leverages it, and a network manager that depends on it.

While not every function provided by traditional Cisco IOS is necessarily supported in IOS XE, most functions and capabilities are. This provides an important bridge in terms of functionality that network managers require to support business continuity and maintain operational discipline and control in their network infrastructures.

IOS XE is also designed to leverage the evolving hardware platforms that are in use in today's—and tomorrow's—network infrastructures. IOS XE is flexible enough to be implemented on CPU-only architectures (as often found in lower-end routers), as well as to leverage the heavily ASIC-based, silicon forwarding infrastructures found in many higher-end routing platforms (and virtually all switching platforms), including such advanced capabilities as multiswitch stacking, in which multiple switches are connected together to operate as a single, unified system, such as with StackWise-480 on the Catalyst 3850, 3650, and 9300 switch platforms and StackWise Virtual on selected versions of the latest Cisco Catalyst 9000 Series switch platforms.

IOS XE is also designed to be able to leverage the multicore CPU architectures and capabilities that are often employed in platforms today, distinct from the single-core CPU platforms of yesteryear. Leveraging such multicore CPU capabilities, where needed, required a fundamental rework of many Cisco IOS software capabilities and processes. The move to Cisco IOS XE provided this capability.

Overall, IOS XE set out to ultimately be a single OS that drives consistency in enterprise network deployments and, at the same time, drives simplified network administration, enabling an accelerated software lifecycle management capability with a similar CLI to older IOS variants, but enabling a whole new set of additional capabilities with modern interfaces such as NETCONF, RESTCONF, and Python scripting capabilities—thus moving IOS device and network management into the 21st century, and beyond.

Figure 8-4 illustrates this consistency. Cisco IOS XE enables the rapid deployment of new solutions, allows for simplified network administration, and provides streamlined software lifecycle management capabilities and ease of scripting, all while providing a familiar IOS CLI and allowing for a rapid adoption and learning curve.

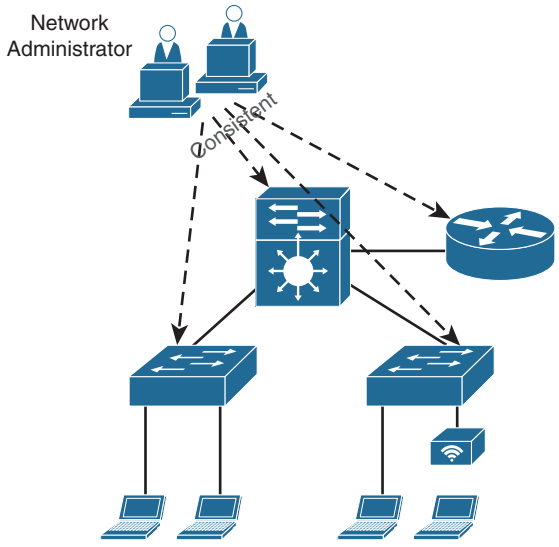

Figure 8-4 *Cisco IOS XE Across the Enterprise Network*

Cisco IOS XE: Delving Deeper

To deliver on some of the key goals of IOS XE, an evolution of the IOS architecture was required. This actually took place over several IOS variants over a period of a few years, as illustrated in Figure 8-5.

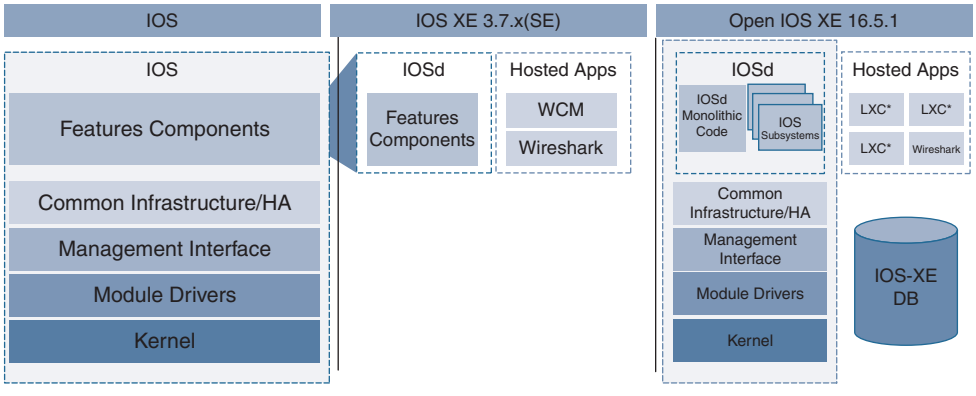

Figure 8-5 *Evolution of IOS to IOS XE*

Starting with the basics of Cisco IOS operating as a single monolithic code base, an evolution began with the first variants of IOS XE (the 3.6.*x* and 3.7.*x* train), which separated

out the various feature components of IOS into IOSd, and allowed for the co-hosting on the IOS XE platform of other applications, such as Wireless Control Module (WCM) and Wireshark.

Continuing and accelerating this evolution, the IOS XE 16.*x* train introduced several key new capabilities for IOS XE. These included the separation of several key functions from the IOS XE monolithic code base (allowing these to operate as separate threads of execution on top of the IOS XE underlying kernel) and providing an in-built database function against which IOS XE processes could checkpoint their various data structures as they were created and used.

In addition, IOS XE expands the application hosting ecosystem to include a framework for support of container-based applications to operate and be co-hosted on the IOS XE platform involved.

Figure 8-6 outlines these three key architectural enhancements enabled with IOS XE 16.*x*—namely, IOS subsystems, the IOS XE Database, and support for containers for application hosting.

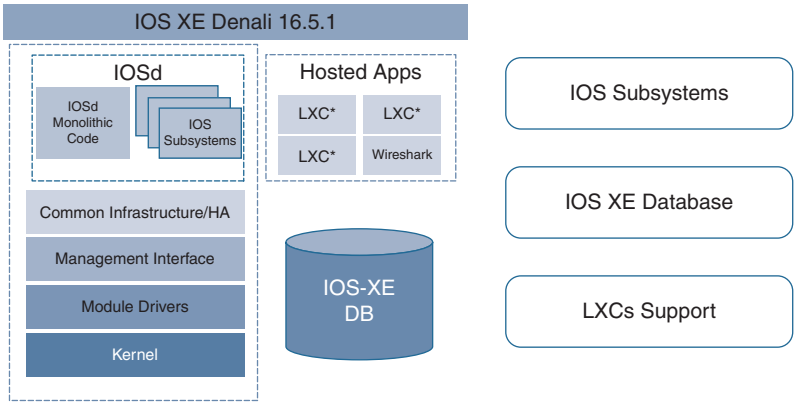

Figure 8-6 *IOS XE—Key Architectural Enhancements*

Let's examine each of these architectural enhancements provided by IOS XE.

IOS XE Subsystems

By separating out the code from the overall IOSd monolithic code base for key functions within Cisco IOS XE, several important benefits are realized. Perhaps the most critical among these from an operational perspective is the ability to provide greater process-level resiliency.

Once portions of the IOS XE code are modularized into separate subsystems, a failure of one of the subsystems leaves the rest of the system, and the other portions of code executing on it, intact. This is illustrated in Figure 8-7.