Preparing Junos OS for the Future

An Architectural Update

Table of Contents

Executive Summary

The networking industry faces a number of challenges that the underlying technology must address. The operating system (OS) running on network elements such as routers, switches, and firewalls plays a key role in that ecosystem. Any OS incapable of evolving and adapting to customer needs will eventually fail. Meanwhile, network traffic levels continue to grow exponentially as new applications and devices proliferate, resulting in greater network complexity, higher operational costs, and slower time to revenue. Abstracting, simplifying, and obscuring this complexity will be a major challenge for the industry moving forward.

Juniper Networks® Junos® operating system was a major disruption to the industry when it was introduced 20 years ago. It was the first modular network OS, providing clear separation between control and data planes. It was also the first network OS to introduce machine-friendly interfaces based on XML, on-box automation, programmability, and open innovation with the Juniper Networks Junos SDK.

Juniper continues to invest in Junos OS, ensuring that it is ready to address any new challenges that may emerge in the future. From evolutionary architectural changes, deeper componentization, and a broader set of APIs, Junos OS continues to make great strides toward simplifying network operations. Moving forward, Juniper intends to maintain its disruptive behavior by introducing a network that is responsive, adaptive, and predictive—an autonomous Self-Driving Network that runs on Junos.

This white paper describes how the Junos OS architecture and infrastructure are evolving, providing details on key areas that have been enhanced such as software disaggregation, automation, programmability, openness, security, performance and scale, and virtualization. More architectural enhancements are inevitable as new business, customer, and industry challenges are identified and new technology trends are introduced.

Introduction: The Evolution of the Network OS

Network operating systems have traditionally had a proprietary and monolithic architecture that ran in a flat memory space, often directly from flash memory or ROM. These architectures often supported multiprocessing using a cooperative model in which each process would run to completion or voluntarily relinquish the CPU. While simpler to build, these systems were plagued with problems associated with resource management and fault isolation where, for instance, a single process could easily consume the entire processor or cause the entire system to fail.

The mid 1990s saw a significant increase in global data traffic, which quickly challenged the capacity of existing networks and routers. A breakthrough solution—the separation of the control and forwarding planes—soon became the norm after it debuted in the industry's first ASIC-driven routing platform: the Juniper Networks M40 3D Universal Edge Router. The network operating system that drove this innovation was Junos OS, and it allowed network operating systems to be more flexible, scalable, and resilient, providing a means of continuous operation with zero downtime.

The networking industry is changing rapidly; today's network OS strives to meet new market demands such as scalability, agility, high availability, security, and programmability. The ability to react to and adopt new architectures is now critical to a customer´s business success, and the network OS is the key enabler. New architecture models where software is disaggregated from hardware, and software components are disaggregated from each other, are proving to be better suited for today's business requirements. The distribution of components, scale-out models, and the integration of third-party components require that the OS be built to seamlessly support them. Junos OS continues to evolve around the key market trends and needs identified in Table 1.

Table 1: Market Trends and Junos OS Evolution

Business Requirements	Junos OS Evolution
Agility to adapt to new business models	Junos OS disaggregation
Continuous need for increased scale	Modularity, multi-threading, and symmetric multiprocessing (SMP)
High availability	Junos Continuity, unified in-service software upgrade (unified ISSU), Junos OS selective upgrade (JSU)
Programmability and automation	Support for YANG models (including OpenConfig, IETF, and custom), dynamic rendering of operational state data, Juniper Extension Toolkit APIs, multiple security options with Junos flex, and fast programmatic configuration
Security	Secure boot
Need for deeper, more scalable, more flexible network insight	Junos Telemetry Interface (JTI)

Agility to Adapt to New Business Models

Use Case: For communication service and cloud providers looking at new deployment models and new architectures that leverage components from multiple vendors.

Junos OS Disaggregation

Junos OS was the first network OS to introduce a clear separation between control and forwarding planes. Juniper has taken that innovation to the next logical step, further componentizing the Junos OS software so that new components can run and operate independently through well-defined APIs, as if they were blocks of Lego. This disaggregation allows users to build and run Junos OS with a set of different components, which may or may not be instantiated, depending on the type of device or type of deployment.

The key architectural change in Junos OS has been to cleanly separate any platform-dependent code from platform-independent code, in a truly compartmentalized model. This change produces a platform-independent code, the Junos OS control plane, and platform-dependent code such as platform and Packet Forwarding Engine (PFE) components. This architectural change increases agility while enabling faster platforms and hardware upgrades. At the same time, these components now run over an underlying Linux infrastructure, allowing users to leverage the benefits of the tools such an infrastructure can provide, as well as the open-source community of technology surrounding it. The disaggregated Junos OS architecture is the reference model that uses those clearly defined APIs and the components described.

The communication among components is entirely based on well-defined APIs that decouple the components from each other through clear abstractions. This ensures backward and forward compatibility, but also ensures that, as we make those APIs openly available, third-party software can interact with those components exactly as we do.

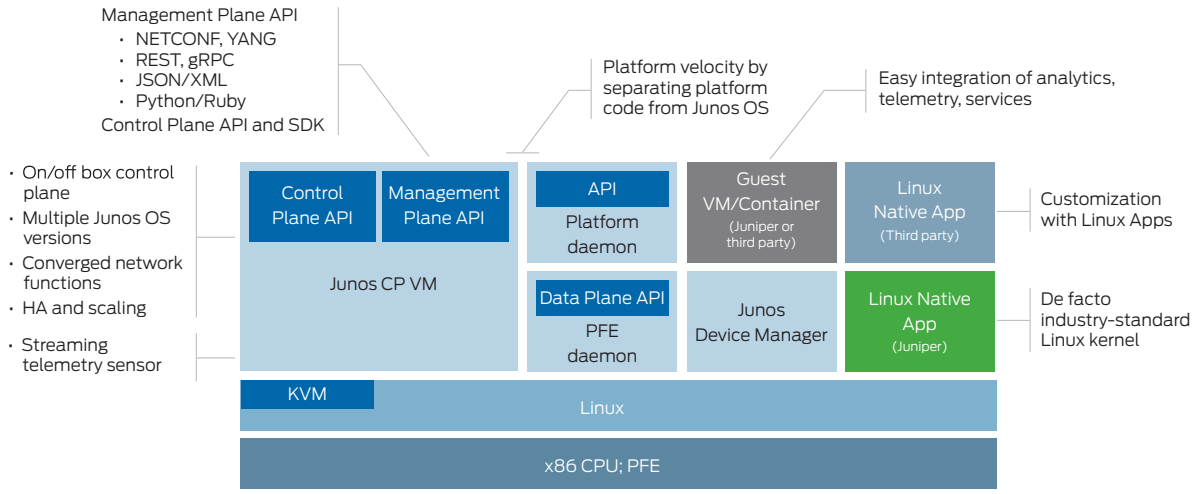

Figure 1: Disaggregated Junos OS software

The original disaggregated Junos OS architecture, running on the Linux infrastructure as a virtual machine (VM), was first introduced on Juniper Networks QFX5100 line of Ethernet switches and Next-Generation Routing Engines (NG-REs). The latest architectural evolution, introduced on the Juniper Networks QFX5200 Ethernet Switch, consisted of separating the control plane component from the platform and PFE-related software. Any new versions or modifications to the platform may remain abstracted from the rest of the system by means of APIs built on top. This new architecture enables running third-party software, either as native Linux processes or in the form of a VM or container.

A new process was required to simplify the management of such a set of loosely coupled components. This process, first introduced in the Juniper Networks NFX250 Network Services Platform, is called Junos Device Manager (JDM). JDM provides mechanisms and tools to manage the life cycle of various components, such as VMs, containers, and Linux processes, interfacing with the Linux virtualization toolkit library (libvrt) APIs. In this architecture, all components offer well-defined APIs that enable their independent evolution.

One interesting use case for this new architecture is the ability to run multiple instances of the Junos OS control plane, enabling use cases such as high availability on single Routing Engine (RE) systems, or running multiple system entities, such as with Node Virtualization[1]. It is now possible to support new deployment models where all components are not physically located on the in-chassis RE while maintaining the operational experience of traditional Junos OS systems.

[1] Node Virtualization refers to hardware virtualization on Junos OS platforms—a physical device will be partitioned into logical entities, each of which will have its own Junos

This is true for both third-party software agents that can run either on- or off-box. It is also now possible to run the Junos OS control plane off-box, such as with Node Virtualization, enabling scenarios that require higher scalability or a large number of control plane instances. This architecture is currently available on Juniper platforms such as the QFX5200 Ethernet Switch, NFX250 Network Services Platform, SRX1500 Services Gateway, PTX1000 Packet Transport Router, and ACX5000 Universal Access Router.

Key Advantages of Junos OS Disaggregation

- **Increased performance:** The new Junos OS architecture offers the ability to run some network components on external x86 servers or in a distributed scale-out model, improving control plane performance.
- **Better leverage of the Linux ecosystem:** Using Linux as the underlying OS, combined with deeper componentization, allows new runtime environments using VMs or even containers. Linux is now the de facto standard kernel, with a vast ecosystem of applications, drivers, toolkits, and features where no other OS or kernel comes close. Junos OS leverages this environment to offer customers all the benefits of the Linux ecosystem.
- **RPM-based package management:** The combination of deeper componentization and the Linux infrastructure allows Juniper to manage software packages in a more flexible and advanced manner, whether it is to track dependencies or upgrade components independently.
- **Programmability and third-party software:** One of the key aspects that disaggregation brings, by the nature of the architectural change to drive all communication between components through clearly defined APIs, is the programmability that customers or open-source communities can use to innovate. This is applicable to control plane APIs (JET, RPD programmability), data plane APIs (switch abstraction interfaces and others), or platform APIs (fans, optics, power supplies, and so on). Existing third-party applications or open-source toolkits may also be seamlessly integrated to add to the Junos OS software, creating customized and differentiated network offerings.

Continuous Need for Increased Scale

Use Case: For communication service providers who desire a network infrastructure that is more powerful, flexible, and scalable with multicore capabilities of the networking devices.

Several Junos OS devices incorporate the upgraded version of FreeBSD Kernel (version 10.0) to offer a host of key benefits:

- **Better use of available compute assets:** By supporting full symmetric multiprocessing (SMP), Junos OS enables the operating system to utilize all CPU cores on the Routing Engine. The scheduler places different daemons on different cores, allowing them to run in parallel. For example, management daemon (MGD) and routing protocol daemon (RPD) can run in parallel in different cores on an RE with two cores.
- **Performance improvement:** SMP support substantially improves overall system performance. In the uni-dimensional testing scenarios, an average improvement of 35% was observed in the major scaling parameters. In the bi-dimensional tests, Junos OS showed about a 60% improvement in the configuration load and commit. In the multi-dimensional case, Junos OS showed up to 25% improvement in local convergence time. When combined, these improvements deliver immense performance boosts across Junos OS devices. Apart from kernel upgrades, Junos OS is moving towards a more modular architecture via the separation of FreeBSD Kernel code from the Junos OS kernel code, ultimately making the Junos operating system modular and allowing the underlying OS to be easily upgraded.

High Availability

Use Case: For communication service providers requiring high SLAs with minimal downtime or disruption.

- **Junos Continuity:** A major cause of downtime is the need to upgrade the network OS whenever new hardware, such as a line card, is installed in a router. These upgrades and the subsequent router reboot always impact existing services. Junos Continuity software prevents these reboots by making the insertion of new line cards a non-service impacting event. Juniper is among the few vendors to support this plug-and-play approach, and Junos Continuity is now supported on the Juniper Networks MX Series 3D Universal Edge Routers product line starting with Junos OS Release 14.1.
- **Junos OS selective update:** The selective update feature provides the ability to apply a patch (hotfix) onto an installed version of Junos OS software running on a Juniper device—in many cases without any service disruption. This feature is especially useful for rapidly resolving critical issues that may be observed on a production router, thereby avoiding extended maintenance windows and prolonged operational impacts.

Programmability and Automation

Use Case: Networks have embraced automation and programmability to increase agility while reducing operational complexity. Automation and programmability also expedite time to market and improve CapEx and OpEx efficiency.

YANG Models Support

From day one, Junos OS has been internal modeling the configuration state using what was effectively the precursor of YANG[2], which is Juniper's Data Definition Language (DDL). Junos OS is the first network OS to use machine-friendly formats such as XML to simplify automation processes, and is also the first to introduce on-box automation programmability through SLAX[3] and Extensible Stylesheet Language Transformations (XSLT) scripts. What Junos OS started has now become mainstream and is standardized across the industry, where Network Configuration Protocol (NETCONF) is used as a transport mechanism, and YANG is used as a data-modeling language. Junos OS now has full support for both.

In addition to supporting YANG to model our configuration objects, Junos OS supports the standardized YANG models as defined by IETF and OpenConfig. Junos OS can easily incorporate any new model without requiring a software upgrade. It can also incorporate any custom YANG model that customers may be interested in, including built to define, and model intent-based interfaces from their operations support systems (OSS) towards the network infrastructure. Junos OS enables the translation from the service provider YANG model into the Junos OS-specific actions to activate such intent.

Dynamic Rendering of Operational State Data

In traditional Junos OS, daemons and utilities display their operational state via "show" commands based on the native Junos OS schema. In the world of programmatic access, the operational state values are retrieved in formats like XML and JSON, as well as plain text that can be based on different schemas such as OpenConfig or YANG. The evolving UI requirements, as well as emerging new schemas and encoding formats, require a dynamic rendering of operational state data.

Junos OS dynamic rendering is a framework that provides the ability to generate different views of operational state at runtime in a simple yet efficient manner. The translation logic is kept coresident with the source of data (the daemon) and the operational state is modeled into a tree of accessors and iterators. This tree describes to the outside world how to access any bits of operational state, thereby converting it into a pull-based model. This means that at runtime, Junos OS can emit XML, JSON, or plain text because the accessors and iterators are just returning data without formatting to a specific schema or encoding. This ability can be leveraged for operational simplicity by allowing a network operator to develop customized network element views using the encoding format of choice.

Juniper Extension Toolkit (JET) APIs

Juniper was one of the first vendors in the networking industry to open its OS to third-party programmability. Juniper introduced Juniper Networks Junos SDK for the control, management, and services data planes many years ago in order to enable third-party integration and to foster innovation.

With communication service providers requiring higher degrees of customization in their networks, and the emergence of open-source and third-party software designed to enhance an existing network OS, today's network operating systems require extreme programmability. JET API is a framework that makes Junos OS open and programmable in a simple and flexible fashion, providing developers with access to a set of high-speed APIs to change and retrieve state information for control and management plane services.

With JET, programmers can choose the development language and the OS for building applications. The JET APIs are consistent across different platforms, decoupled from Junos OS releases, and have binary compatibility across releases. APIs are also consistent, regardless of whether the application exercising those APIs runs on the Junos OS device or an external server attached to a Junos OS device.

Automation Frameworks

Junos OS now supports a wide range of automation frameworks such as Puppet, Chef, Ansible, and Salt, allowing seamless integration of Junos OS-powered network elements into the customer's overall IT systems management infrastructure.

Junos OS also supports PyEZ, which is a micro framework for Python developed by Juniper, which enables users to remotely manage and automate devices running the Junos operating system. PyEZ provides the same capabilities as the Junos OS CLI, in an environment built for automation tasks.

[2] YANG is a data modeling language used to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF), NETCONF remote procedure calls, and NETCONF notifications.

[3] Stylesheet Language Alternative Syntax (SLAX) is a language for writing Junos OS commit scripts, op scripts, event scripts, and SNMP scripts.

To complement the automation offerings of Junos OS, a separate Junos OS variant for enhanced automation, called Junos flex, is also available. This variant comes with VeriExe[4] disabled, allowing users to run unsigned applications on the device and giving them more flexibility.

Fast Programmatic Configuration

In the world of automation and software-defined networking (SDN), the rate at which configuration changes is very high. Such frequent changes are required by various network applications based on certain events to ensure that the network is dynamic and agile enough not to become a bottleneck, especially in the virtual world.

For example, when a VM moves from one server to another, an event is generated from VM management software, which is received by a network application or SDN controller that, in turn, configures VLAN changes on switches to ensure network connectivity. Another example is that of broadband subscribers joining or leaving a network. They may need to set up corresponding virtual private LAN service (VPLS) configurations followed by a number of filters. In a large network, this could mean thousands of frequently generated configuration transactions that need to be rapidly absorbed by network elements.

The "Fast Programmatic Configuration Database" is a special database intended for programmatic configuration transactions. It can be used with JET APIs or the NETCONF protocol to ensure that all network configurations are accurate, minimizing network downtime and virtually eliminating human errors.

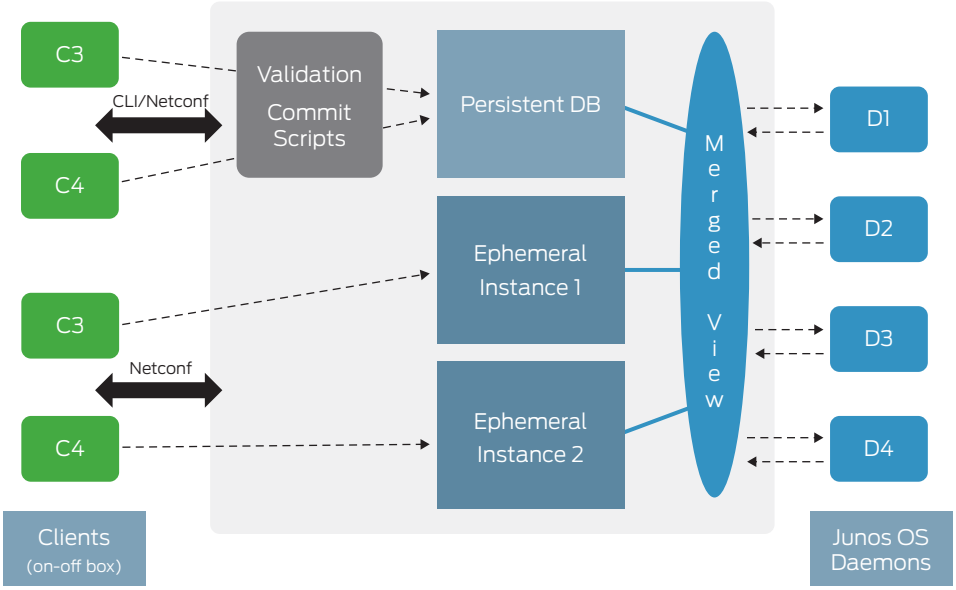

Figure 2: Fast Programmatic Configuration Database

Security

Use Case: Infrastructures demand an extra level of security to ensure that the networking devices boot up a genuine and untampered version of Junos OS.

As the Internet evolves and computer networks continue to scale up, security continues to be one of the most important considerations for network operators. Cyberattacks are becoming more complex, and Juniper continues to offer enhanced security features in Junos OS to minimize vulnerability.

Secure Boot allows a device to not only boot up with an authorized and signed version of the BIOS and Junos OS, but also ensures that the router boots an image that has not been tampered with. This is achieved by carefully controlling access to boot resources and checking signatures starting from BIOS to loader to kernel. This provides a secure and trusted chain of software development and compilation at Juniper, as well as at the network element at the customer site.

The first device with this capability was the Juniper Networks SRX1500 Services Gateway, which shipped in December 2015. Secure Boot is also enabled on the QFX10008 Switch, which shipped in January 2016. More Junos OS devices will ship with this powerful feature in the near future.

The latest upgrade to the FreeBSD kernel has brought numerous security updates to Junos OS, including Federal Information Processing Standards (FIPS) crypto compliance and common criteria compliance for FreeBSD kernel.

[4] VeriExec is a file-signing scheme for the NetBSD operating system.

Need for Deeper, More Scalable, and More Flexible Network Insights

Use Case: Network operators are looking at streaming statistics and event states from networking devices for real-time or post analysis, moving from a reactive to a self-healing approach.

Junos Telemetry Interface (JTI)

Increasingly, network change cycles are growing shorter and more frequent. As a result, network operators require enhanced visibility into the network to identify, isolate, and resolve potential issues as quickly as possible to keep their networks running smoothly. The once popular SNMP has limitations in addressing today's most pressing issues.

First, SNMP employs a "pull" model where servers initiate data collection on a set schedule or at predetermined intervals. However, network disruptions that occur between the set intervals, such as traffic drops caused by congestion, can be missed if polling is not done more frequently. A "push" model that reports notable events or statistics, triggered by preset thresholds, actions, or events, is preferable. Second, SNMP can no longer provide all the available information. Junos Telemetry Interface (JTI) can be used for low-level information export which may not be available in SNMP.

The ability to gather statistical data at scale, driven by events and in near real time, is enabled by JTI. JTI is a highly scalable distributed collection engine designed to help network operators stream statistics and event states to data collectors, network controllers, or similar devices for real-time or post analysis.

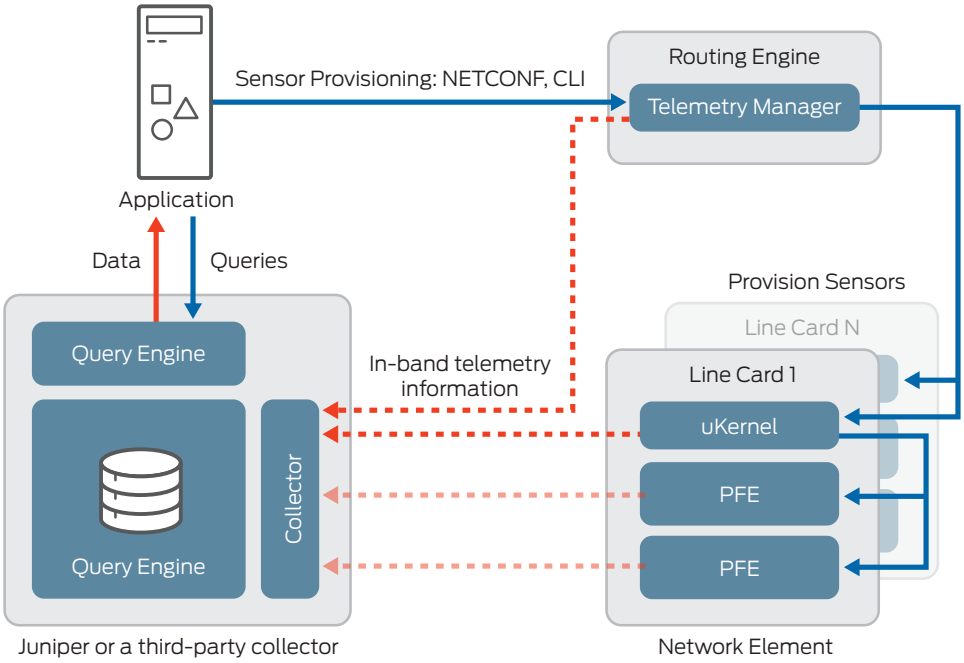

Figure 3: Streaming data and analytics

Scalable Message Communication Buses

Junos OS architecture has been enhanced to contain high-scale message communication buses like Message Queue Telemetry Transport (MQTT), which enable internal pub-sub models for telemetry and operational state. With this capability, new applications, new modules, or agents can subscribe to this bus and receive any desired telemetry or state for either local processing (on-box event correlation, for example) or to be exported to external collectors using multiple techniques.

Conclusion

More than ever, network operating systems play a fundamental role in rapidly evolving network technology. Junos OS is constantly transforming to ensure that customer requirements are being addressed, be they scale, modularity, resiliency, extensibility, programmability, automation, or telemetry. As new ways of building network infrastructures emerge, newer architectural evolutions are planned for Junos OS.

Junos OS disaggregation opens up a broad set of opportunities. Evolving beyond separating the control plane from the forwarding plane, communication service providers and enterprises alike can now leverage a new architecture enabling seamless separation of multiple software components that are logically coordinated and highly programmable. The result is a network infrastructure that is responsive, adaptive, and, ultimately, autonomous and self-driving.

About Juniper Networks

Juniper Networks challenges the status quo with products, solutions and services that transform the economics of networking. Our team co-innovates with customers and partners to deliver automated, scalable and secure networks with agility, performance and value. Additional information can be found at Juniper Networks or connect with Juniper on Twitter and Facebook.

Corporate and Sales Headquarters
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or +1.408.745.2000
Fax: +1.408.745.2100
www.juniper.net

APAC and EMEA Headquarters
Juniper Networks International B.V.
Boeing Avenue 240
1119 PZ Schiphol-Rijk
Amsterdam, The Netherlands
Phone: +31.0.207.125.700
Fax: +31.0.207.125.701

2000651-002-EN Sept 2017