**Software-Defined Networking Concepts**

SDN can be defined as:

* An approach and architecture in networking where control and data planes are decoupled and intelligence and state are logically centralized.
* Abstraction of the underlying network infrastructure from the applications (network virtualization).
* A concept that leverages programmatic interfaces to enable external systems to influence network provisioning, control, and operations.

**Traditional Network**

* Control plane learns and computes (apprend et calcule) forwarding decisions
* Data plane acts on the forwarding decisions
* Control plane and data plane reside within the physical device

**Classic SDN**

* Control plane becomes centralized
* Physical device retains data plane functions only

**Hybrid SDN**: a controller is centralized and separated from the physical device, but devices still retain localized control plane intelligence

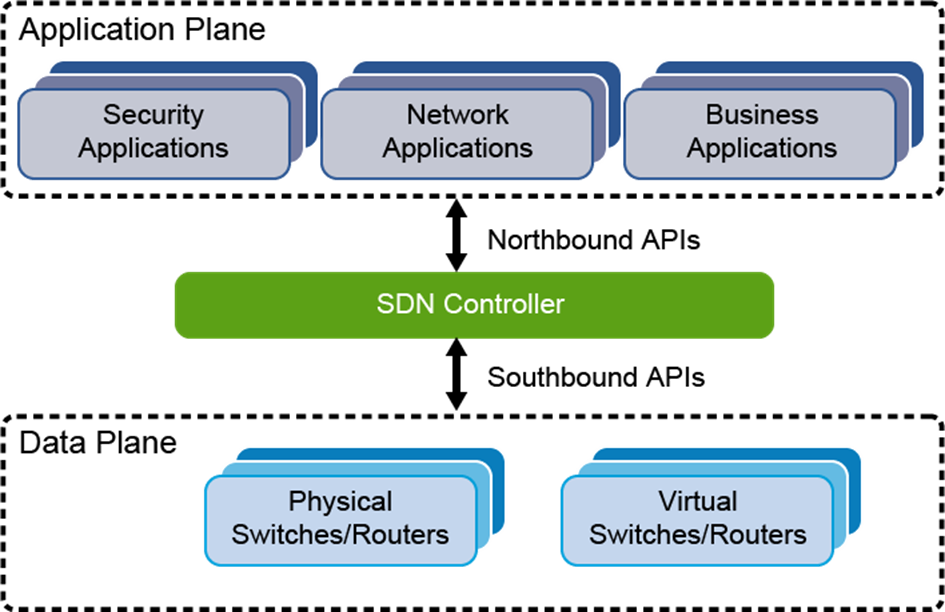
**SDN benefits**

* Fast and automated application deployment.
* On-demand application delivery and mobility at scale.
* Greater resource flexibility and utilization.
* Reduction of IT costs

**SDN Building blocks**

SDN offers a centralized, programmable network that consists of the following building blocks:

* **SDN controller**: a body of software that replaces, or enhances, the control functions that are traditionally part of the embedded operating system in a network device.
* **Northbound APIs:** used for communication with the applications and services running over the network (example: REST API)
* **Southbound APIs**: relay information to the switches and routers in network



**Open-Source Software**

Open source software is part of a community-driven trend to develop and promote open standards development particularly in the networking industry

* **OpenFlow:** A low-level protocol that is used to manipulate the forwarding table of a switch.
* **Open vSwitch (OVS):** A virtual switch that is normally used along with hypervisors to connect virtual machines within a host or hosts on different networks.
* **Contiv:** Several projects defining infrastructure operational policies for container-based application deployment.
* **OpenStack Neutron:** An SDN project that is focused on delivering network-as-a-service in multi-tenant virtual compute environments.
* **OpenDaylight:** A collaborative project seeking to promote and organize a community-driven approach to SDN adoption.
* **FD.io (fast data—input/output):** A collection of several projects and libraries to amplify the transformation that began with Data Plane Development Kit (DPDK) to support flexible, programmable, and composable services on a generic hardware platform.

**Programmable Infrastructure**

There are two forms of network programmability: “on-box” and “off-box

**Examples of on-box automation and scripting mechanisms are:**

* **TCL:** Cisco introduced embedded TCL into several of its operating systems over a decade ago and is a common method to perform on-box scripting of simple tasks.
* **EEM:** Another embedded automation mechanism on many Cisco platforms that are used to provide real-time event detection and scripting.
* **PowerOn Auto Provisioning:**
* **Smart Install:** A zero-touch method to provision a larger volume of network devices using on board mechanisms, the Cisco Discovery Protocol, a TFTP server, and a network device director.
* **Smart Port Macros:** Macros that are embedded in the switch software including some pre-tested Cisco-recommended templates for Catalyst switches.
* **Python:** Certain Cisco platforms support built-in interactive and non-interactive Python scripting modes which give programmatic access to the network device’s command line interface.

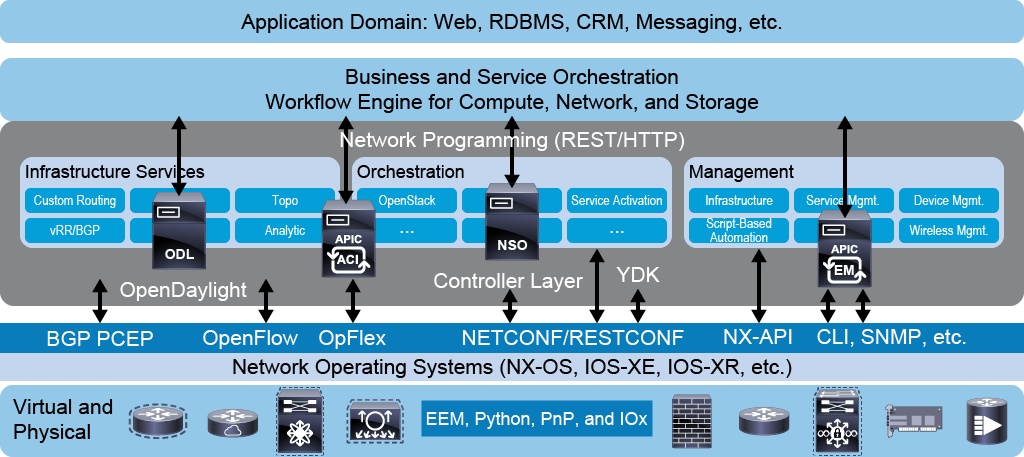
**Examples of programmatic APIs and mechanisms that enable off-box programmability to automate and script network devices are:**

* **REST APIs:** APIs using REST are stateless, client-server, and resource-based which means they use, in programming terms, nouns rather than verbs. REST APIs normally use HTTP as the transport mechanism.
* **NETCONF:** Provides mechanisms to install, manipulate, and delete the configuration of network devices
* **RESTCONF:** It is an HTTP-based protocol that provides a programmatic interface for accessing data that are defined in YANG and using the datastores that are defined in NETCONF.
* **Software development kit (SDK):** A set of tools that allows an end user to create their own custom applications for various purposes including managing hardware platforms.
* **DevOps tools:** The DevOps culture enables the continuous development and improvement of open, off-box mechanisms.
* **Linux**

**Cisco ACI**

It is an industry-leading SDN solution that facilitates application agility and data center automation. The Cisco Application Policy Infrastructure Controller (APIC) is the main architectural component of the Cisco ACI solution.

**Cisco SDN and Network Programmability Technologies**



**Cisco SDN Solutions**

Cisco portfolio has different SDN solutions:

* **Cisco Digital Network Architecture (DNA) Center:** Represents the network management and command center for Cisco DNA, the intent-based network for the enterprise. Cisco DNA is extensible, software-driven architecture that accelerates and simplifies your enterprise network operations, while lowering costs and reducing risks. Cisco DNA Center is an SDN controller, which enables you to provision and configure all your network devices in minutes
* **Cisco Software-Defined (SD) Access**: An intent-based networking solution for the Enterprise built on the principles of Cisco DNA.
* **Cisco SD-WAN:** A cloud-delivered architecture that enables you to securely connect any user and any application over WAN using a single management console
* **Cisco SD-Branch**: Enables you to deploy, monitor, and optimize network services with software and hardware that's purpose-built for branch deployments.

Network Automation

Uses

* **Device provisioning:** it is simply configuring network devices more efficiently, faster, and with fewer errors because human interaction with each network device is decreased.
* **Data collection and telemetry:** A common part of effectively maintaining a network is collecting data from network devices and telemetry on network behavior.
* **Compliance checks:** Network automation methods allow the unique ability to quickly audit large groups of network devices for configuration errors and automatically make the appropriate corrections with built-in regression tests.
* **Reporting:** Automation decreases the manual effort that is needed to extract information and coordinate data from disparate information sources in order to create meaningful and human readable reports.
* **Troubleshooting:** Network automation makes troubleshooting easier by making configuration analysis and real-time error checking very fast and simple even with many network devices.

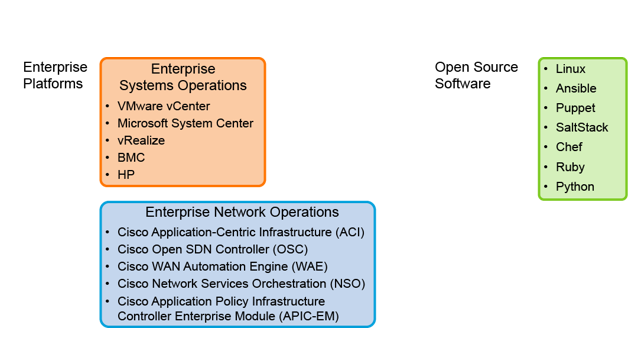
**Notes**

* The **API** is the mechanism by which an end user makes a request of a network device and the network device responds to the end user.
* This method provides increased functionality and scalability over traditional network management methods.
* In order to transmit information over the wire, the APIs require a transport mechanism, such as **SSH, HTTP, and HTTPS**.

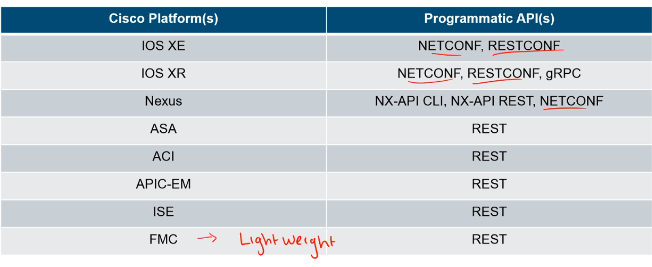
The programmatic methods for network device management utilize the APIs, such as:

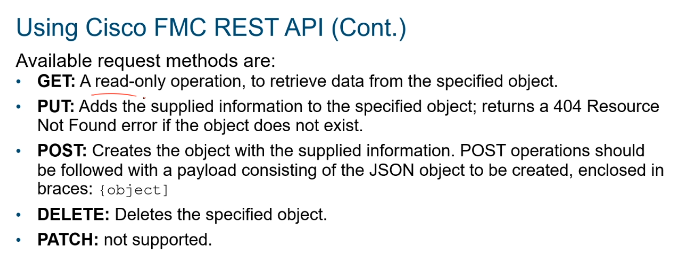
* NX-API
* REST
* NETCONF
* RESTCONF
* gRPC

**Network operations tool**



Cisco platforms and their corresponding APIs:





**Basic Python Scripts for Automation**

* Python is a simple and powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming, which makes it an ideal language for scripting and rapid application development
* Python is supported by the Cisco networking product line, you can use this language to edit and modify existing scripts, or write new ones, for efficient automation of Cisco networking tasks throughout the enterprise.
* You can also use Python scripts to interact with the exposed external APIs of different Cisco platforms (off-box network automation), such as NETCONF/RESTCONF, REST, etc

**YANG** is an API contract language used in many networking devices. In other words, you  
can use YANG to write a specification for what the interface between a client and network-  
ing device (server) should be on a particular topic.