Network Operational Simplicity with Zero Touch Deployment

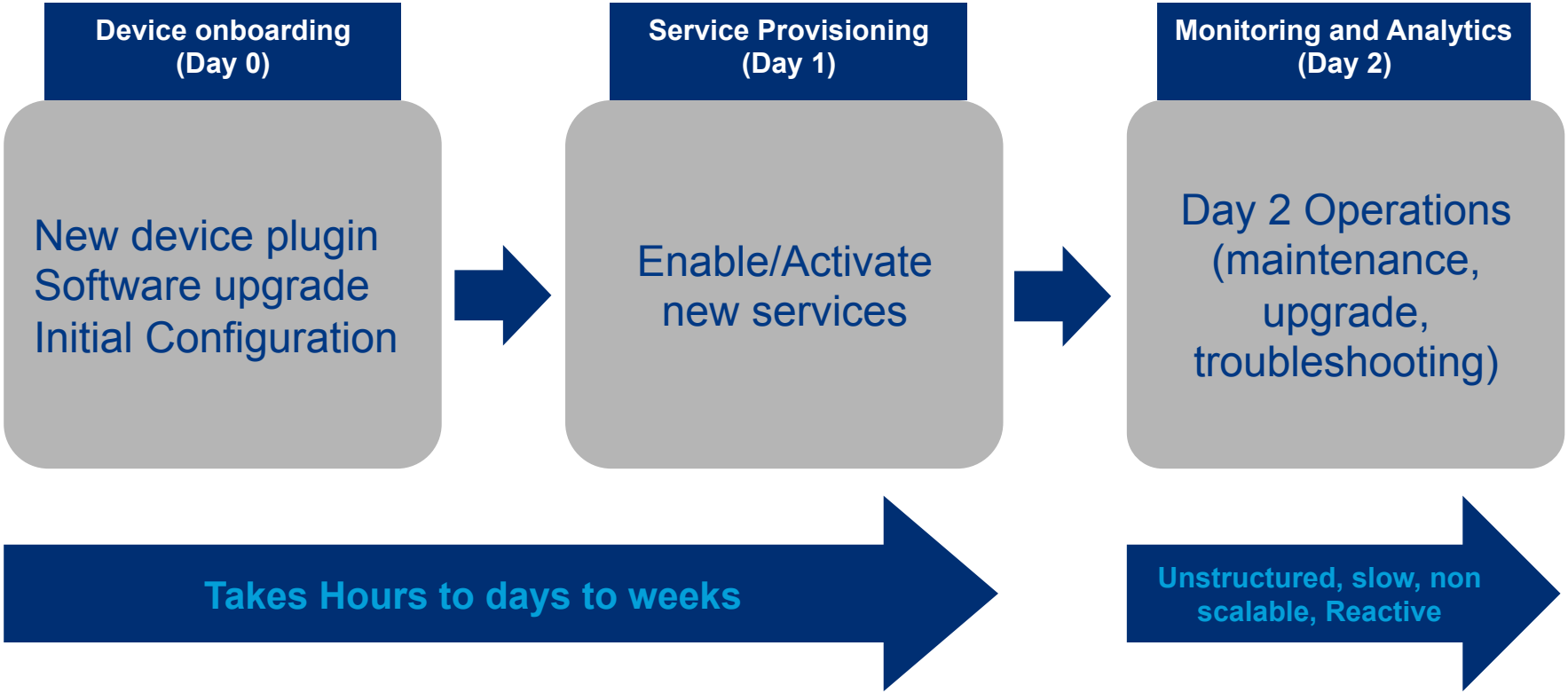
Ahmed Abeer, Technical Marketing Engineer, Cisco
Patrick Warichet, Technical Marketing Engineer, Cisco

Agenda

- What is Zero Touch Deployment (ZTD)
- ZTD Deployment Scenarios
- Solution Components
- Step by Step Flow
- Use Cases

Introduction

Life of a Device - Yesterday

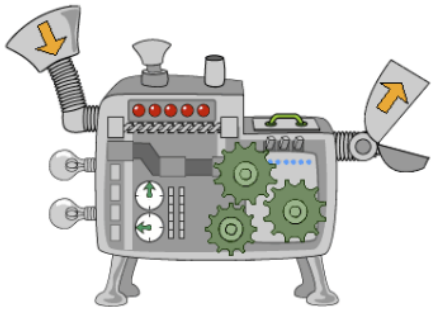


Modernizing the Life of a Device

	Device Onboarding (Day 0)	Service Provisioning (Day 1)	Monitoring & Analytics (Day 2)
Yesterday	Manual, Lengthy, Skilled Worker, Costly Device Bring-up	High Maintenance, Inflexible, Custom Scripts for Service Bring-up	Unstructured, Non scalable, Pull Mechanism, Non Machine Readable Format
Today	Orders of Magnitude Quicker, Automated Device Bring-up	Automation Friendly, Flexible, Predictable, Vendor Neutral Model Driven Service Bring-up	Push mechanism, consistent, machine readable, high performance, real time

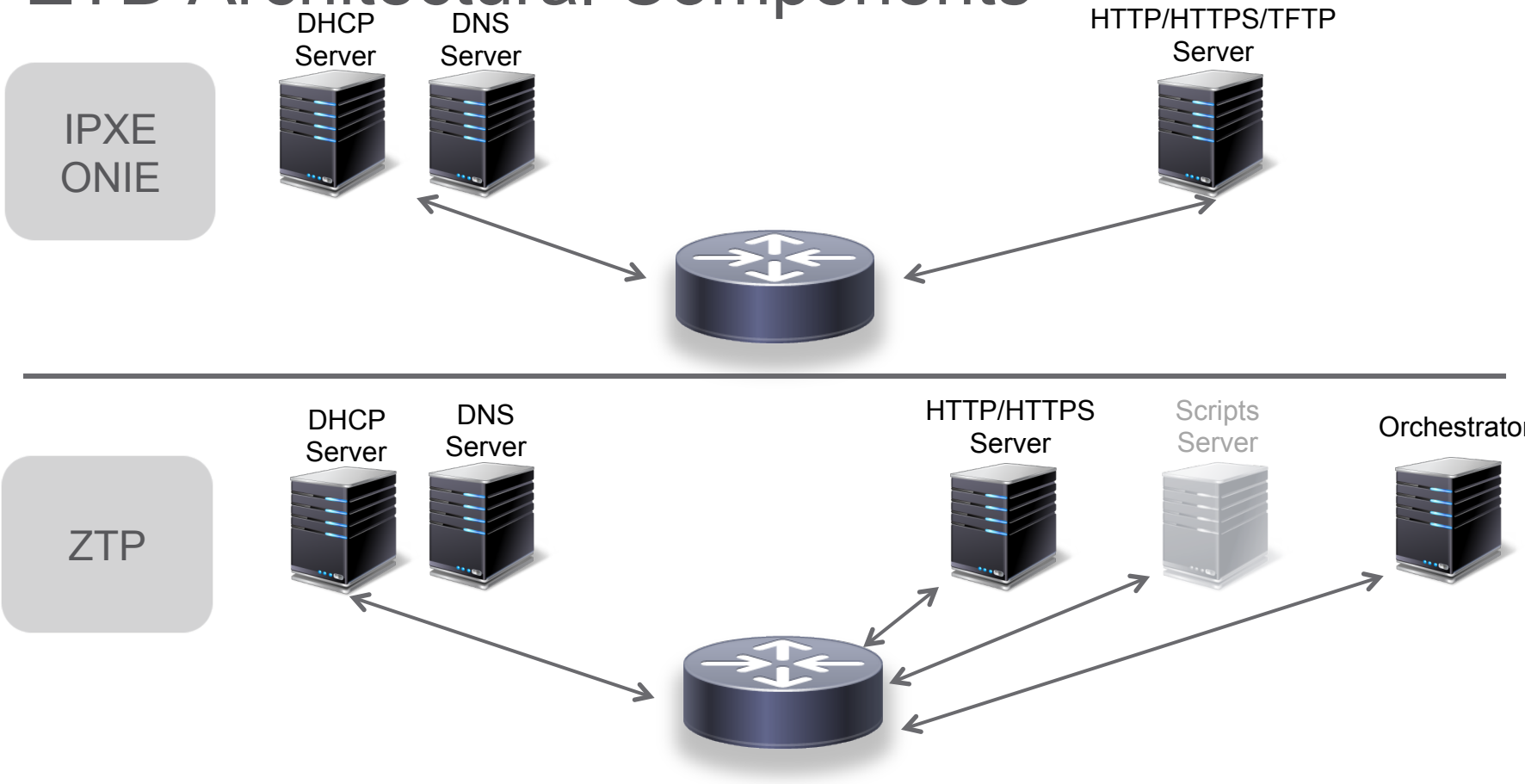
What is Zero Touch Deployment (ZTD)

- ZTD allows you to provision new devices in your network automatically, without manual intervention.
- ZTD Goals:
  - ✓ Bring Up the Device with Customer Certified Software or Image.
  - ✓ Download Day 0 Configuration or Script for Remote Connectivity.
  - ✓ Connect the Device to NMS, Controller or Orchestrator.
  - ✚ Secure Communication Channel Between Device and Server;
  - ✚ Multivendor Support



ZTD Components

ZTD Architectural Components



Boot Process - iPXE

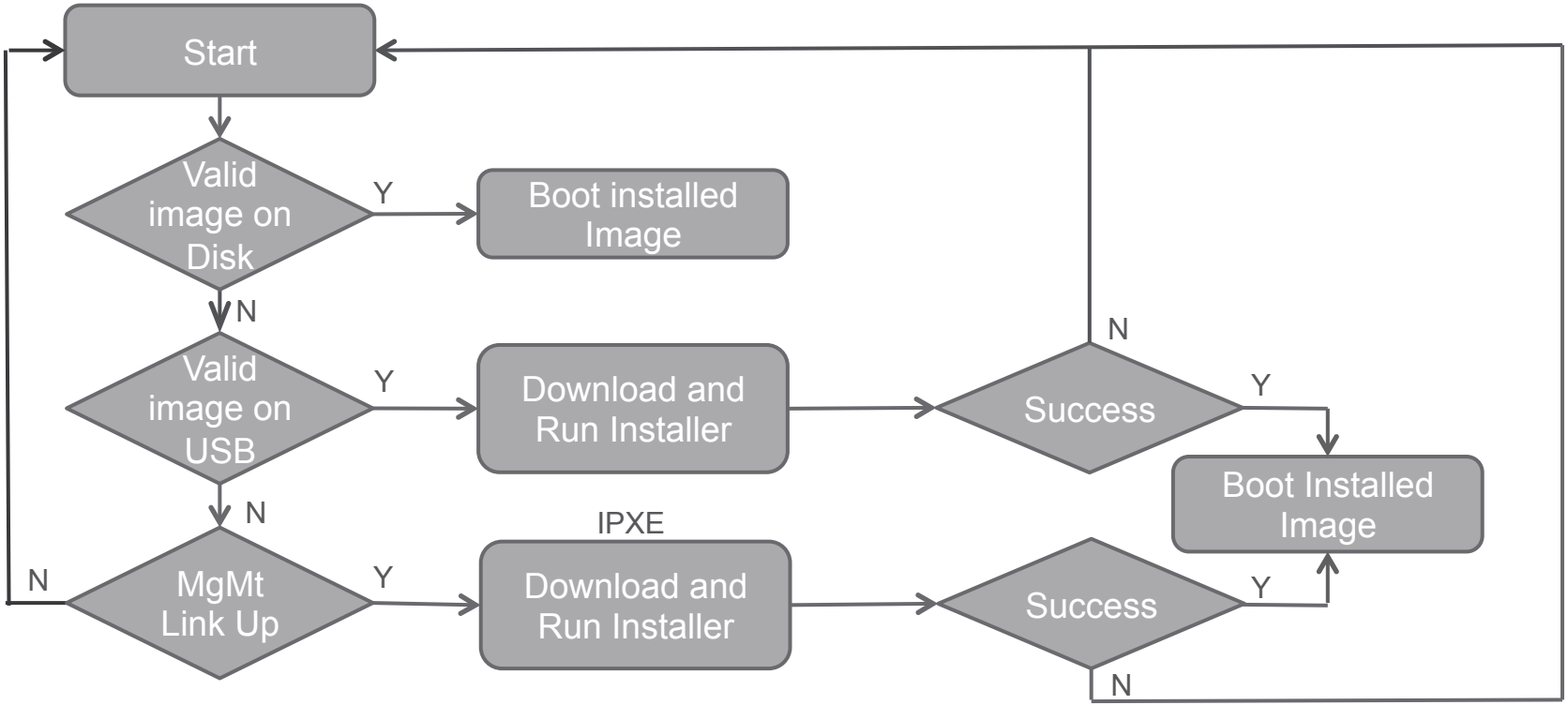
#!iPXE

- iPXE is an open source boot firmware.
- Provides an environment for installing any NOS
- iPXE is supported on the management interfaces (IPv4 and IPv6)
- Fully backward compatible with PXE with several enhancements.
  - Boot from a web server via HTTP
  - Boot from locally attached storage, (USB memory stick)
  - Control the boot process with scripts and menus.
  - DNS support.

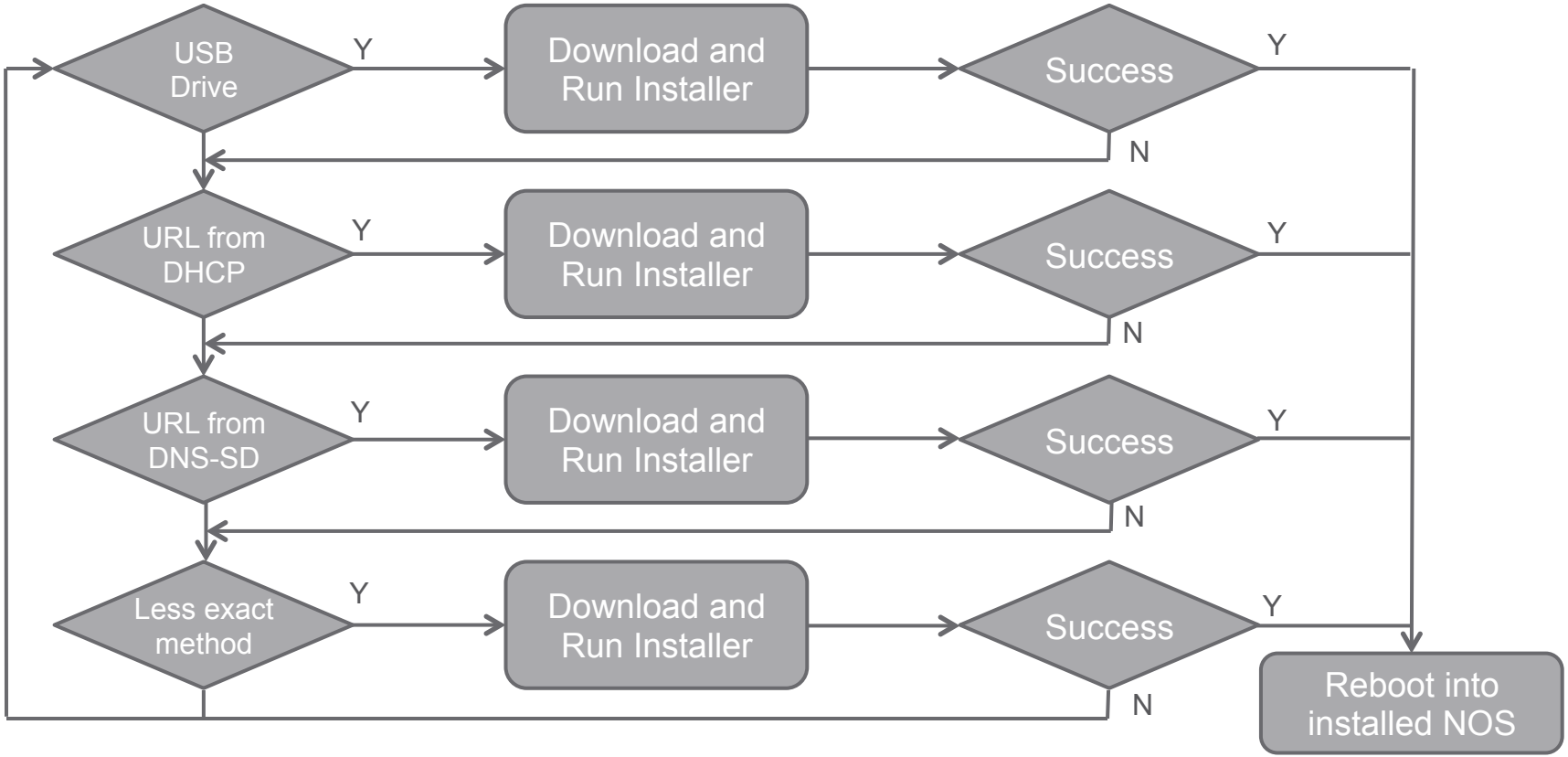
Boot Process - ONIE

- ONIE is a small operating system, pre-installed on bare metal network switches, that provides an environment for automated provisioning.
- Combines a boot loader with a Linux kernel and BusyBox.
- Provides an environment for installing any NOS.
- Supported on the management interfaces (IPv4 and IPv6).
- Aggressive Image discovery
  - Statically configured from the boot loader
  - Locally attached storage, (USB memory stick)
  - Bootfilename from DHCPv4 / DHCPv6
  - IPv4 / IPv6 link local neighbors
  - mDNS / DNS-SD
  - PXE-like TFTP and HTTP waterfalls

Boot Process with iPXE



Boot Process with ONIE



DHCP options make things easy

- Host identification can be done using mac-address.
- Including Serial Number in option 61 ”dhcp-client-identifier“ for example makes provisioning easy (S/N is written on the package)

```
######## Hosts #########
 host ncs-5001-b {
   option dhcp-client-identifier "FOC1947R144";
   if exists user-class and option user-class = "iPXE" {
     filename = "http://172.30.0.22/ncs5k-mini-2.iso";
   } else if exists user-class and option user-class = "exr-config" {
     filename = "http://172.30.0.22/scripts/ncs-ztp.sh";
}
   }
   fixed-address 172.30.12.52;
 }
```

Option 61

Option 77

Option 77

Dynamic URL makes things easy

- The URL provided by the DHCP server does not have to be a static. For example, you could direct iPXE to boot from the URL
- `http://172.30.0.22/boot.php?mac=${net0/mac}&product=${product:uristring}&serial=${serial:uristring}`
- Which would expand to a URL such as:
- http://172.30.0.22/boot.php?mac=c4:72:95:a7:ef:c0&product=NCS5001&serial=FOC1947R143
- The boot.php program running on the web server could dynamically generate a script based on the information provided in the URL.

```
<?php
 header ( "Content-type: text/plain" );
 echo "#!ipxe \n";
 echo "set myURL http://172.30.0.22/Cisco/NCS/NCS5001/FOC1947R143 \n";
 echo "boot myURL \n";
?>
```

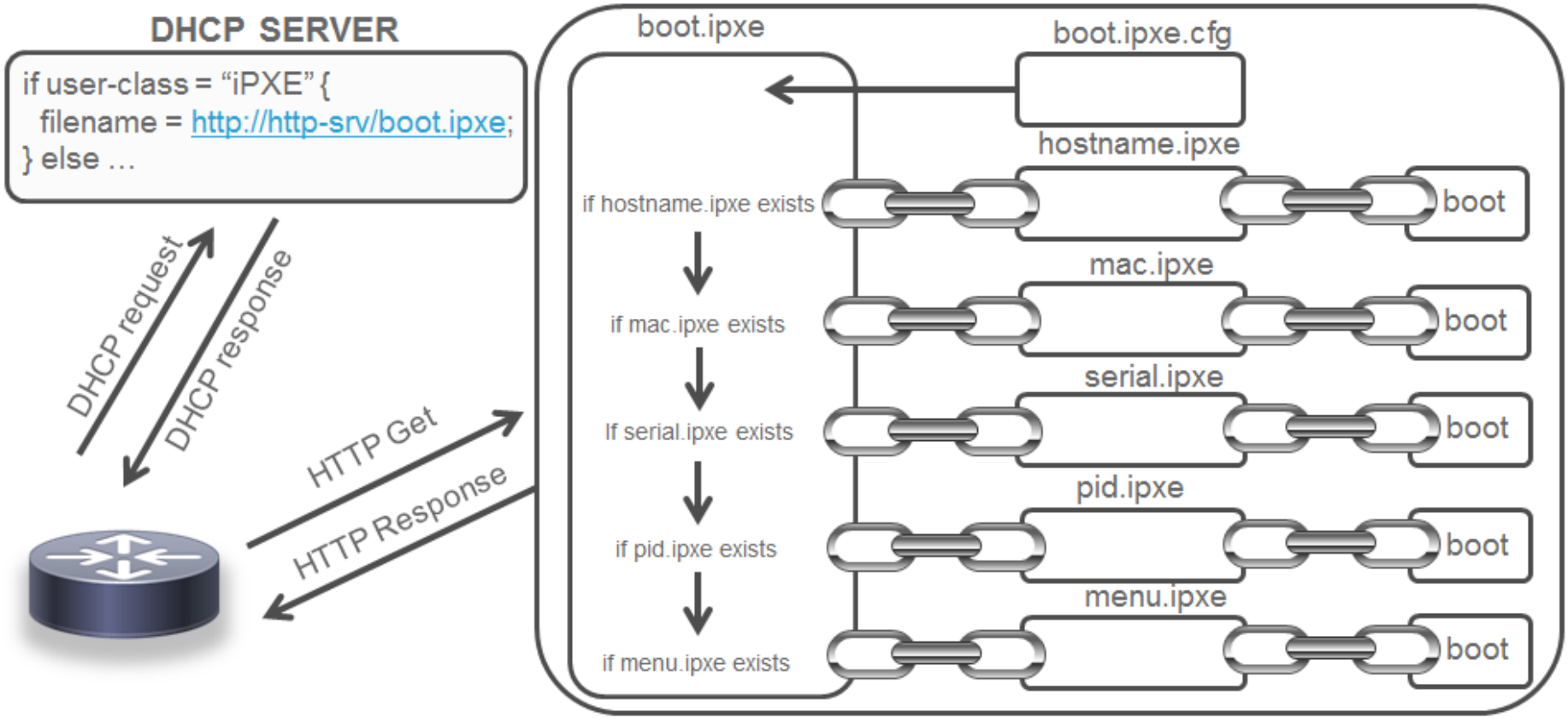
HTTP headers make things easy

HTTP Headers		
Header	Value	Example
ONIE-SERIAL-NUMBER:	Serial number	XYZ123004
ONIE-ETH-ADDR:	Management MAC address	08:9e:01:62:d1:93
ONIE-VENDOR-ID:	32-bit IANA Private Enterprise Number in decimal	12345
ONIE-MACHINE:	<vendor>_<machine>	VENDOR_MACHINE
ONIE-MACHINE-REV:	<machine_revision>	0
ONIE-ARCH:	CPU architecture	x86_64
ONIE-SECURITY-KEY:	Security key	d3b07384d-ac-6238ad5ff00
ONIE-OPERATION:	ONIE mode of operation	os-install or onie-update

iPXE Scripting and Chainloading

- Chainloading is the capability to jump from one boot statement to another.
- Using chainloading and the embedded scripting capability of iPXE we can have a very detail and complex selection mechanism for the boot image.
- Chainloading remove the need to create DHCP host definition
- Agnostic IPv4 or IPv6

Chainloading Example



Chainloading Example with IPv6

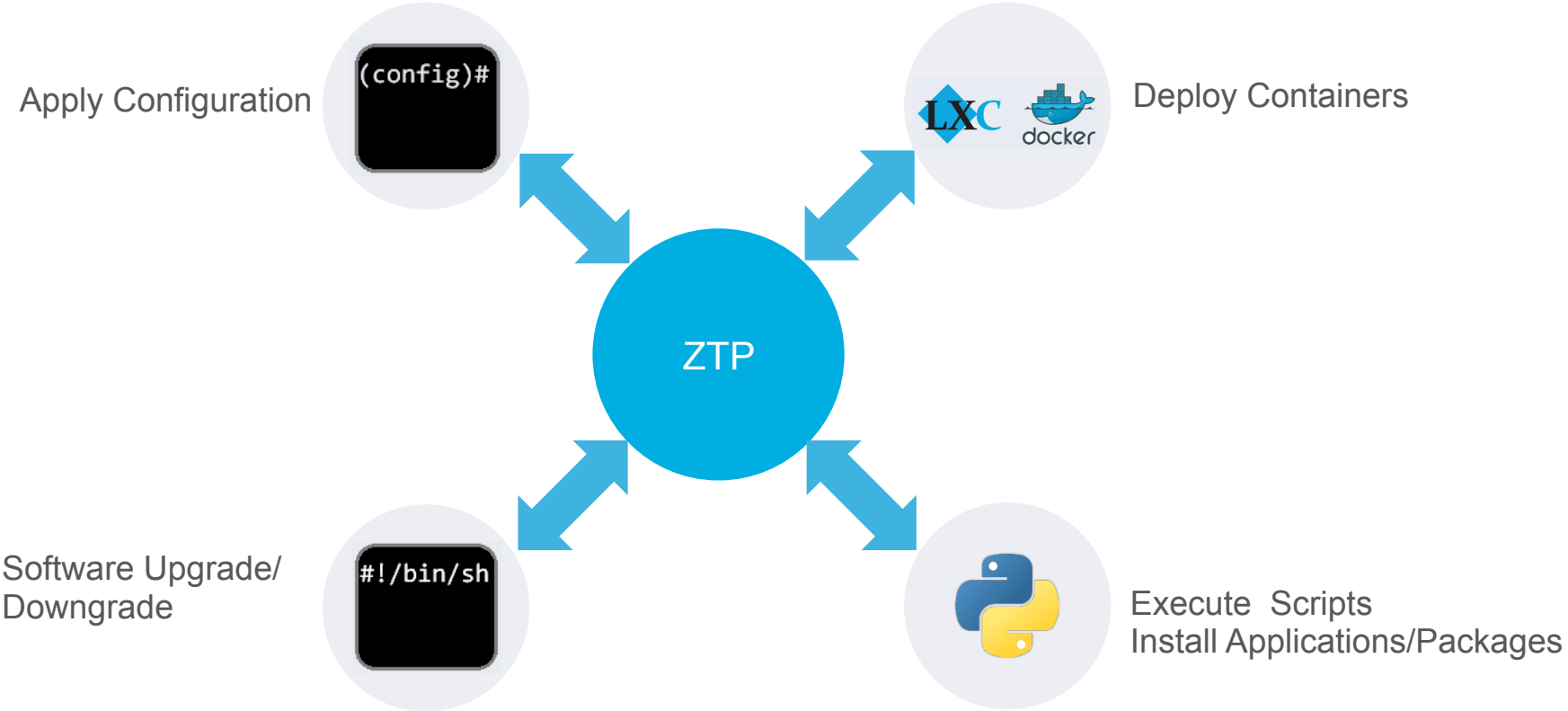
```
iPXE> autoboot net0 <- autoboot from the mgmt interface
net0: c4:72:95:a7:ef:c0 using dh8900cc on PCI01:00.1 (open)
[Link:up, TX:108 TXE:0 RX:5188624 RXE:5186887]
Configuring (net0 c4:72:95:a7:ef:c0)..... Ok
net0: fe80::c672:95ff:fea7:efc0/64
net0: fd:30:12::1124/64 gw fe80::fa72:eaff:fe8b:ce80 <- ipv6 stateful address assignment
Filename: http://[fd:30::172:30:0:22]/boot.ipxe <- ipv6 boot URI from DHCPv6 http://[fd:30::172:30:0:22]/
boot.ipxe... ok <- boot script is downloaded
/boot.ipxe.cfg... ok <- boot variable are chained
/ipxe/uuid-03000200-0400-0500-0006-000700080009.ipxe No such file or directory (http://ipxe.org/2d0c618e)
/ipxe/mac-c47295a7efc0.ipxe... No such file or directory (http://ipxe.org/2d0c618e)
/ipxe/serial-FOC1947R143.ipxe... No such file or directory (http://ipxe.org/2d0c618e)
/ipxe/pid-NCS-5001.ipxe... No such file or directory (http://ipxe.org/2d0c618e)
http://172.30.0.22/menu.ipxe... Network unreachable (http://ipxe.org/280a6090)
http://[fd:30::172:30:0:22]/menu.ipxe... ok <- boot menu is executed
```

Future Enhancement

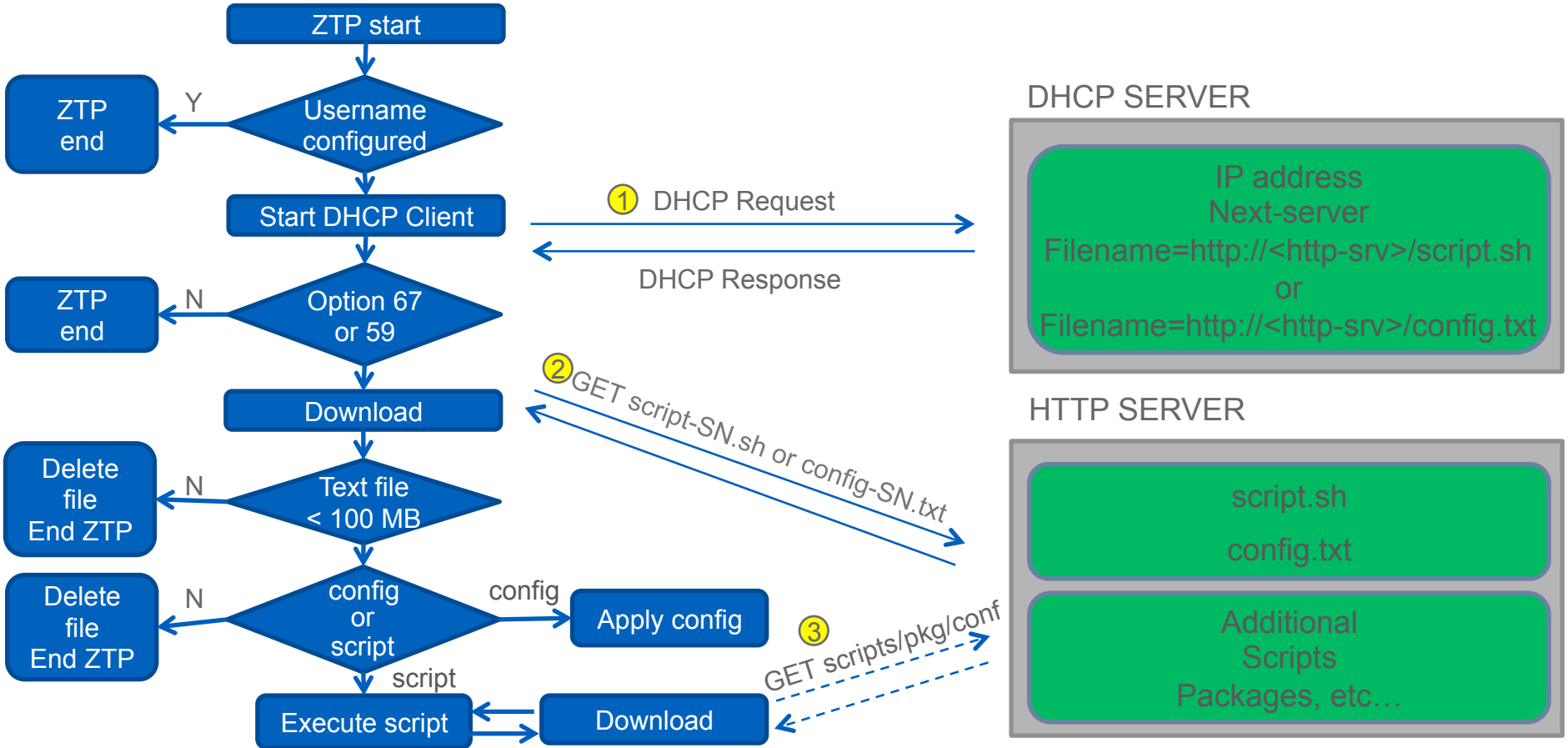
- Security
- Trusted Platform Module (TPM)
- UEFI Secure Boot

ZTP

ZTP Functions



ZTP Flow of Operations



ZTD via NETCONF/YANG

The Day 0 Challenges

5G

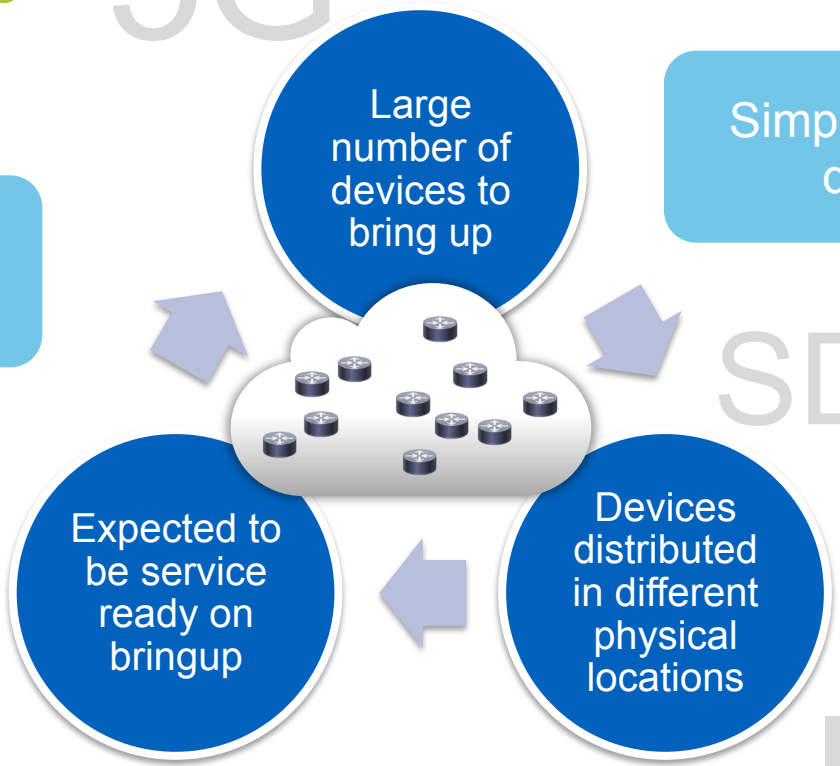
SD-WAN

IoT

Rapid Nodes and Service deployments

Simplify Day 0 device deployments

Service-Ready Infrastructure



ZTP Requirements

Baseline requirements across both deployment scenarios

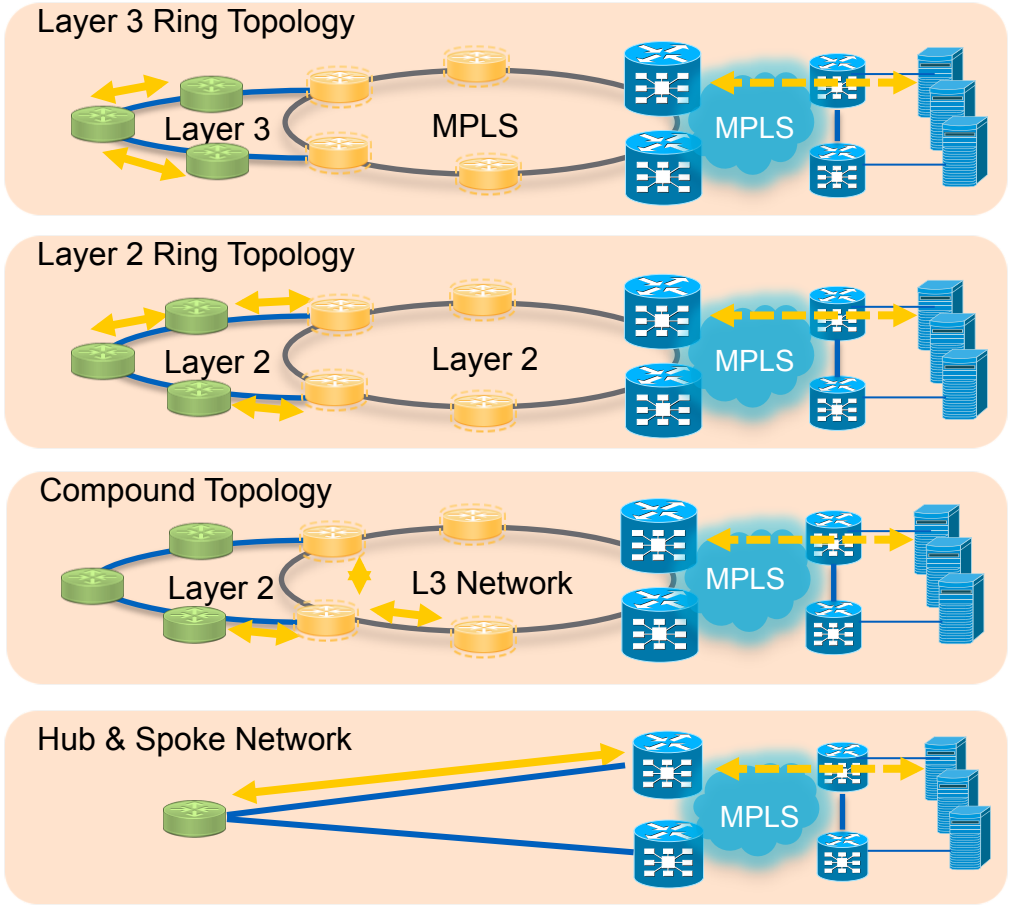
- No pre-staging required
- DHCP for management IP address
- Configuration download
- Image upgrade/downgrade
- Docker/LXC install
- Connection to the NMS

Baseline requirement for “in-band management” deployment scenario

- Auto L3 adjacency configuration in any topology
- L2 VLAN auto-discovery

Value added requirements

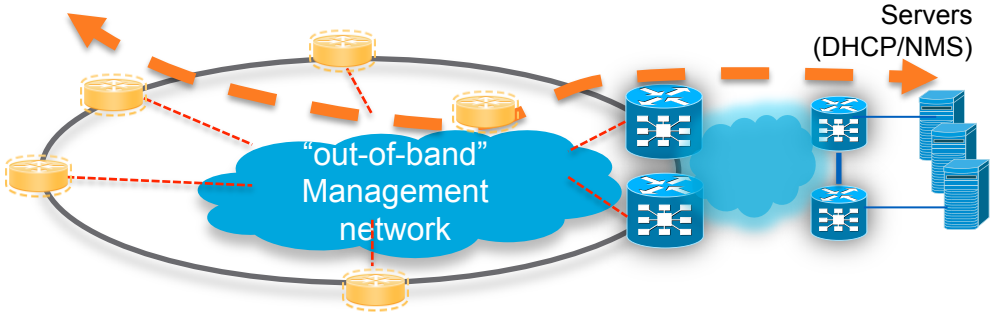
- Robust connection to NMS
- Secure
- Multi-vendor support
- Configuration template



ZTP – Two Different Deployment Scenarios

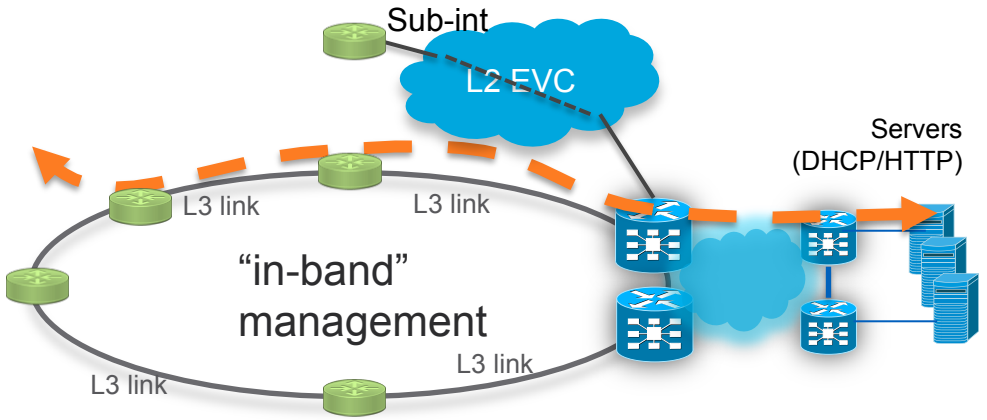
1

- Routers are connected to a management network via out-of-band management port
- Popular in Data Center, Enterprise, and Web customers



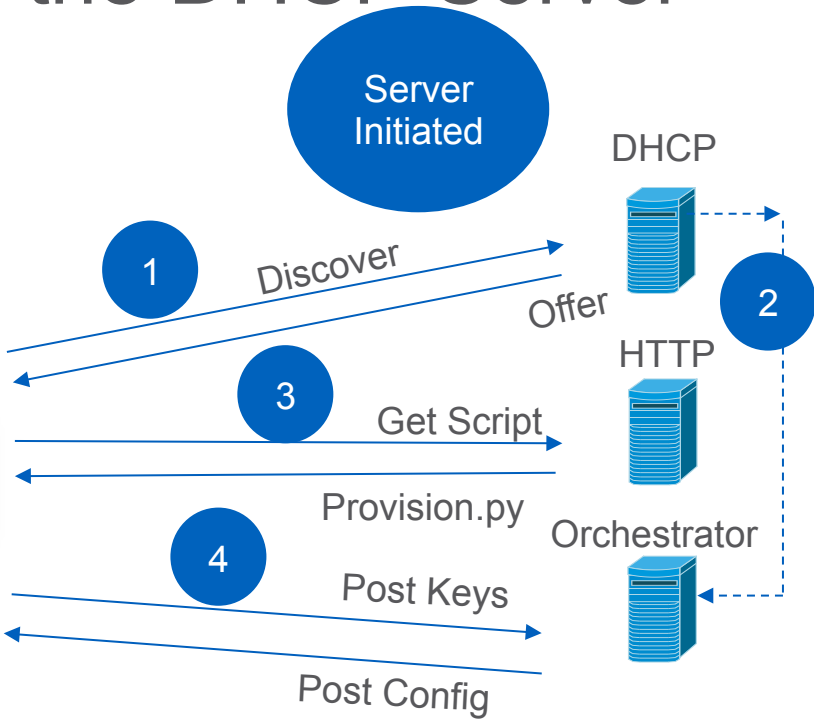
2

- There is no dedicated management network.
- Routers are managed via in-band, the same as user data network
- Typical deployment in the SP Access/ Metro



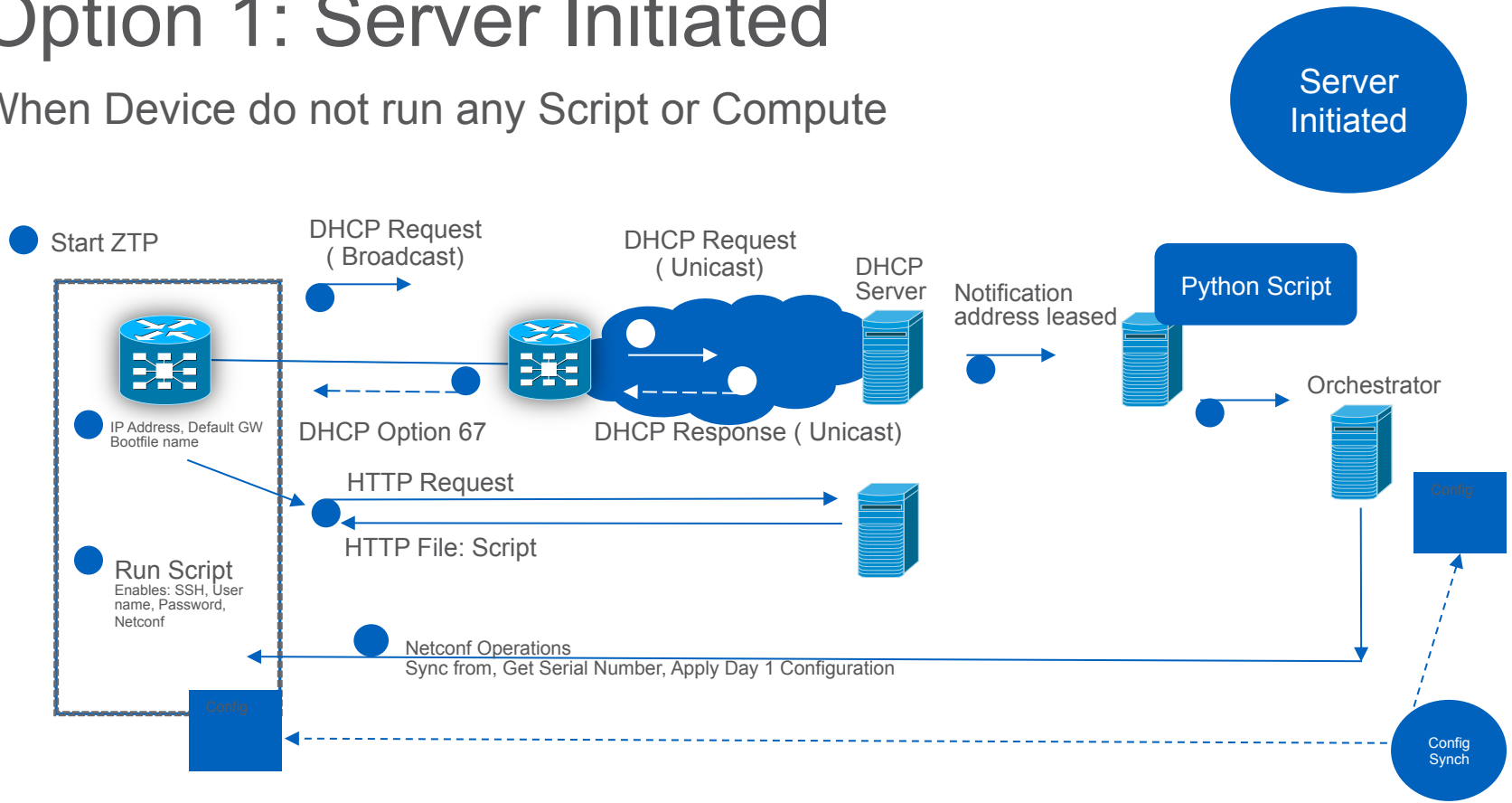
Option 1: Provisioning from the DHCP Server

1. Device boot up and initiate a **DHCP Discover**
   - DHCP server provides a script using “bootfilname” (option 67)
2. Upon commit DHCP server:
   - Registers device to Orchestrator using REST
   - Asks Orchestrator to retrieve RSA keys from device
3. Device Downloads scripts from HTTP server.
   - Scripts is executed on the device.
4. Once registered, the script perform a sync from the Orchestrator server



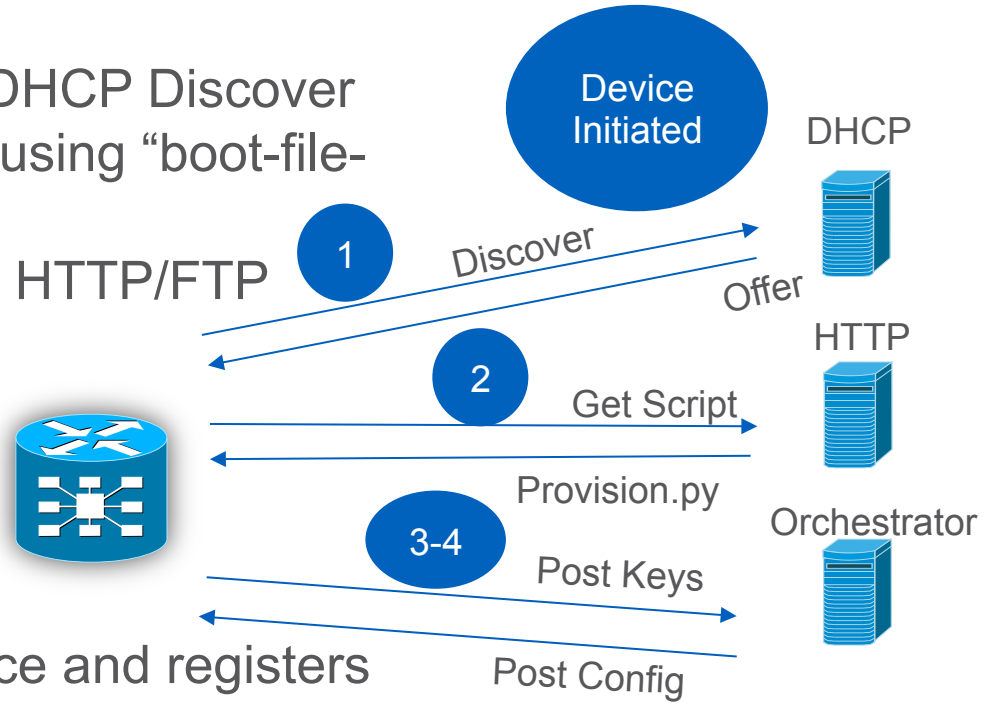
Option 1: Server Initiated

When Device do not run any Script or Compute



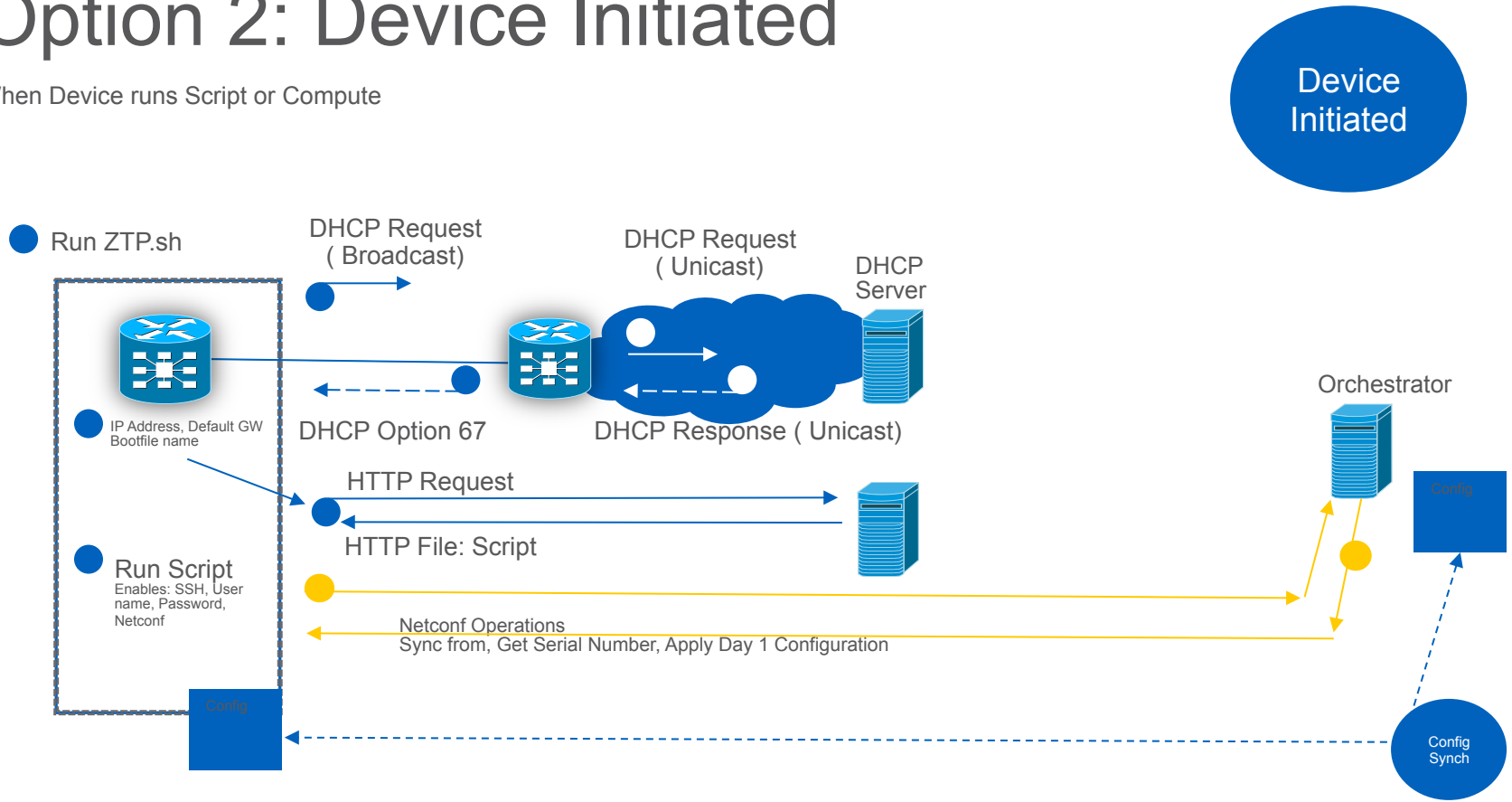
Option 2: Provisioning from the Device

1. Device boot up and initiates a DHCP Discover DHCP server provides a script using “boot-file-name” (option 67)
2. Device Downloads scripts from HTTP/FTP server
3. Scripts is executed on the device and registers to Orchestrator using REST/RESTCONF API
4. Once registered, the script perform a sync from the Orchestrator server



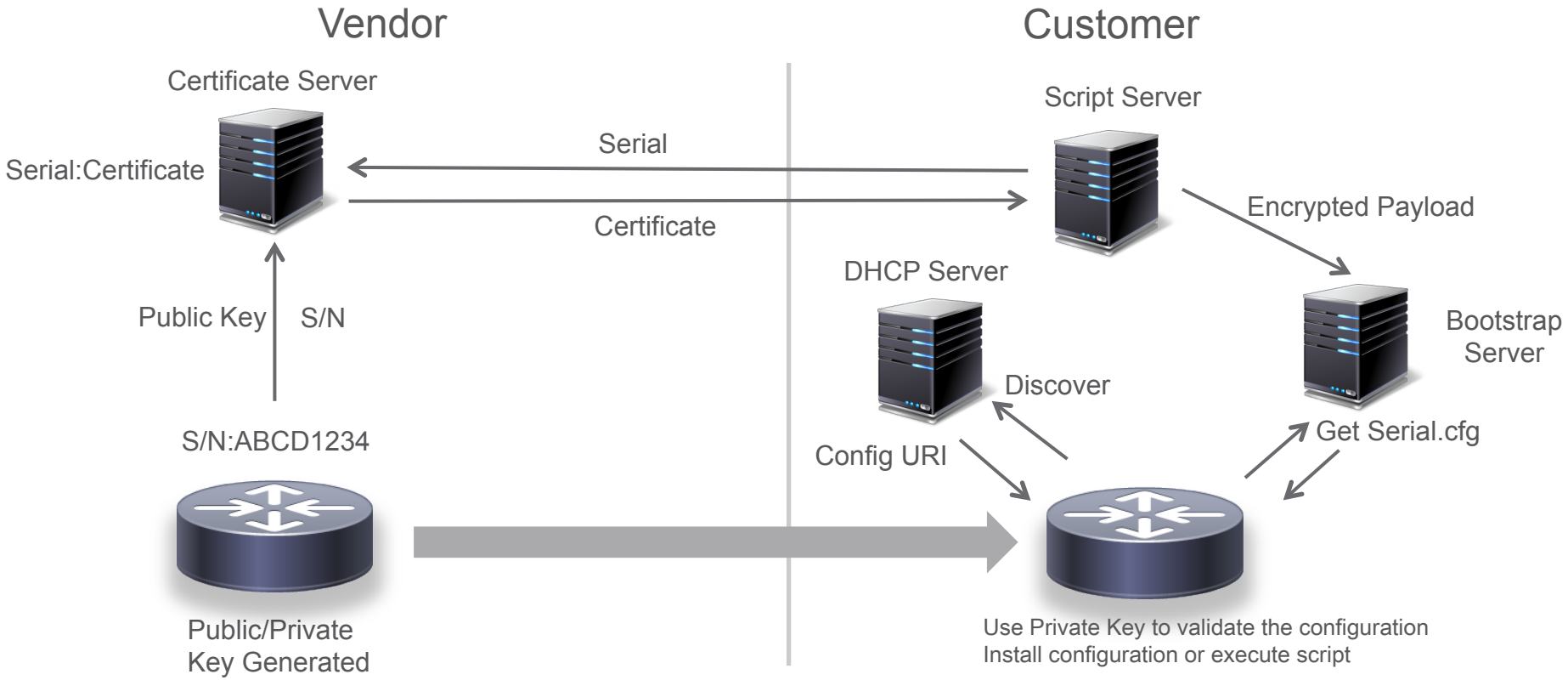
Option 2: Device Initiated

When Device runs Script or Compute



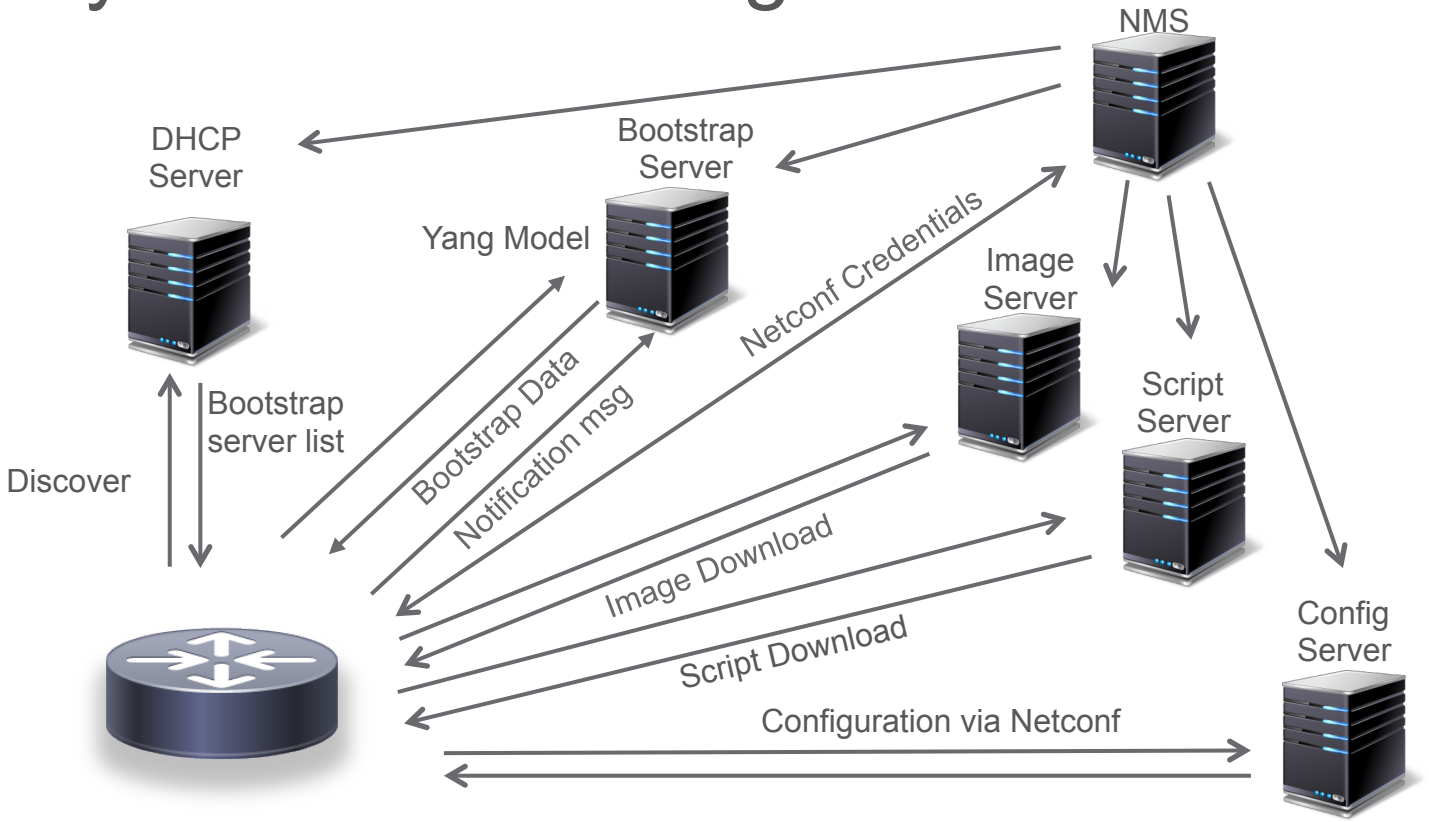
Enhancements

Simple Security



From https://www.ietf.org/archive/id/draft-wkumari-opsawg-sdi-01.txt

Day-0 Netconf Configuration Model



https://tools.ietf.org/html/draft-ietf-netconf-zerotouch-19

Day-0 Bootstrap Data using Yang Model

```
yang-data zerotouch-information:
      +---- (information-type)
      +--:(onboarding-information)
          +---- onboarding-information
            +---- boot-image
            |  +---- os-name              string
            |  +---- os-version           string
            |  +---- download-uri*        inet:uri
            |  +---- image-verification* [hash-algorithm]
            |     +---- hash-algorithm    identityref
            |     +---- hash-value?       yang:hex-string
            +---- configuration-handling?     enumeration
            +---- pre-configuration-script?   script
            +---- configuration?              <anydata>
            +---- post-configuration-script?  script
```

Name of the NOS required

Released version of NOS required

List of URIs providing bootstrapping data

e.g.: merge, replace

Agnostic of programing language
Executed before configuration is applied
Exit 0 : success
Exit > 0 : Soft error -> proceed
Exit < 0 : Hard error -> reset

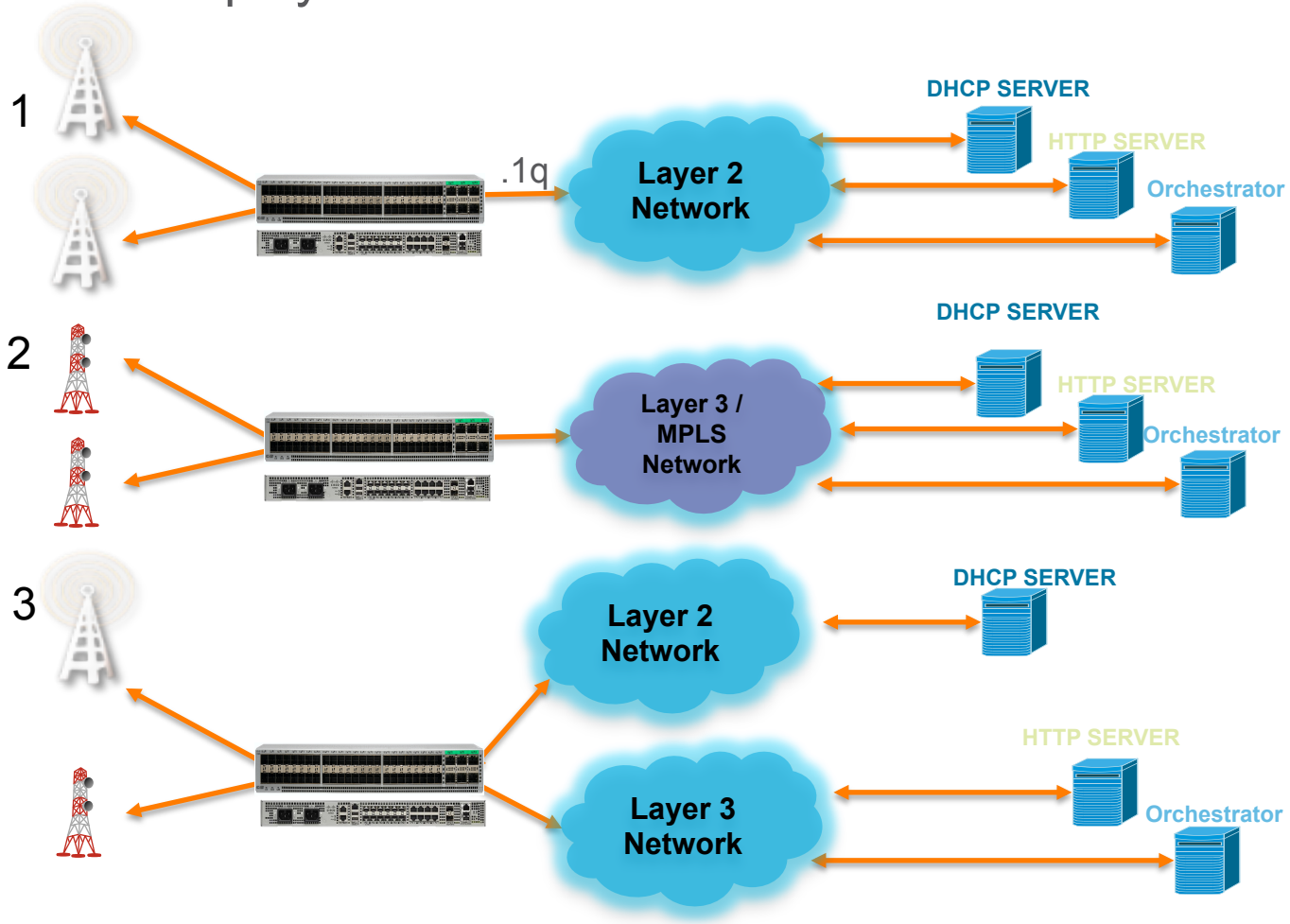
Any model supported by the device

Agnostic of programing language
Executed after configuration is applied
Exit 0 : success
Exit > 0 : Soft error -> proceed
Exit < 0 : Hard error -> reset

From draft-ietf-netconf-zerotouch-19

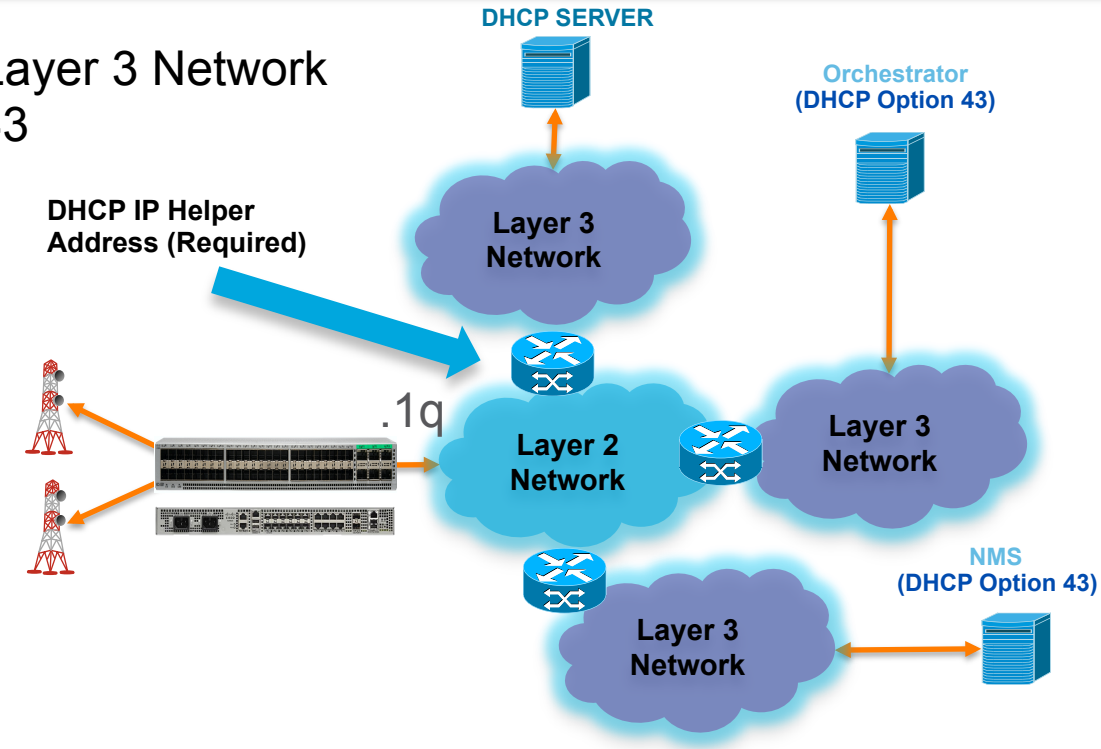
ZTD Use cases

ZTP Supported Deployment Scenarios



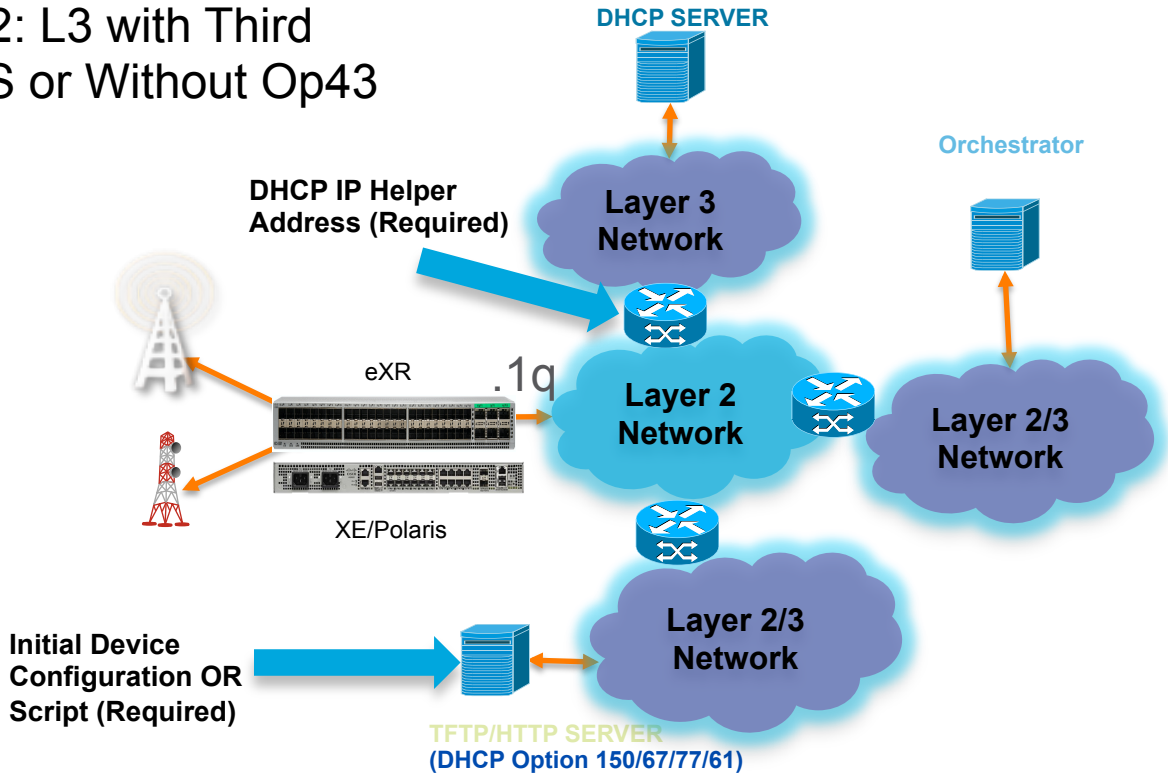
ZTP Supported Deployment Scenarios

Scenario 2: Layer 3 Network with Option 43



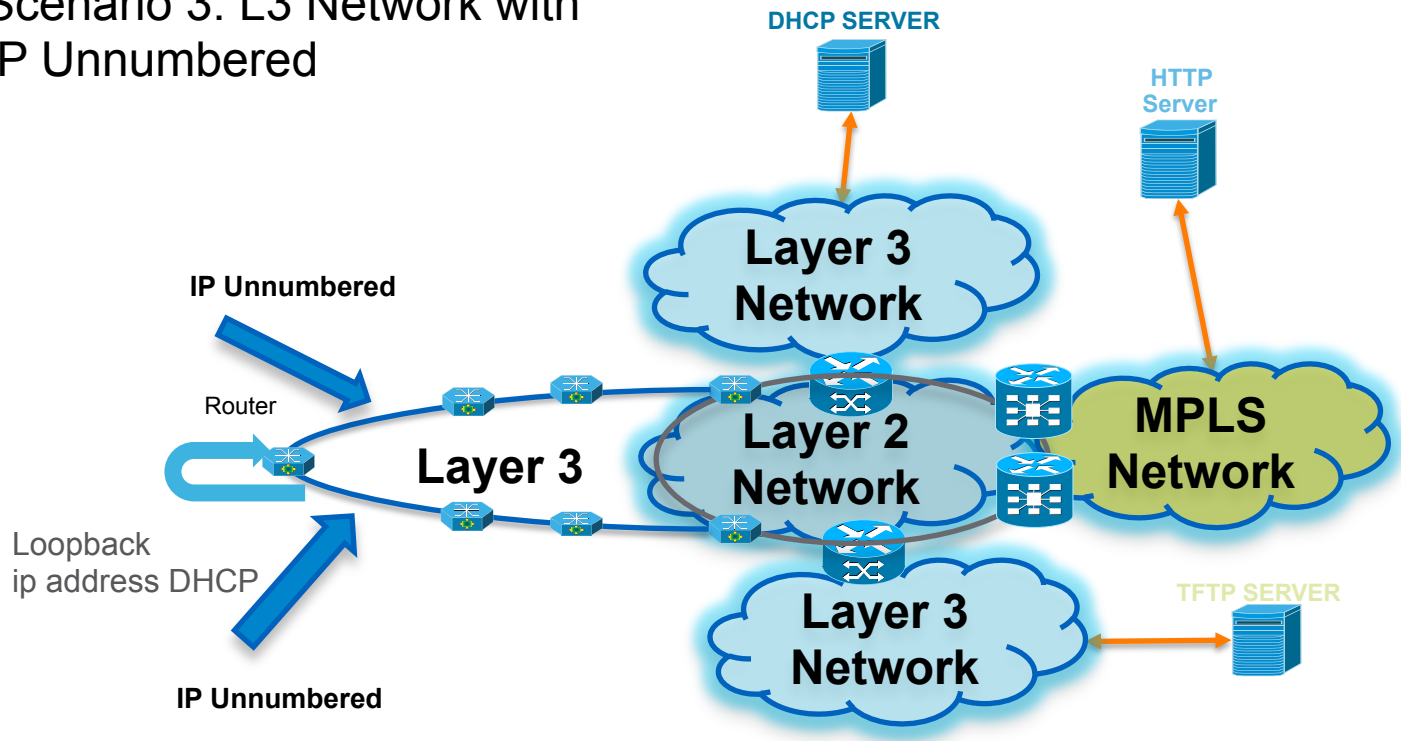
ZTP Supported Deployment Scenarios

Scenario 2: L3 with Third Party NMS or Without Op43



ZTP Supported Deployment Scenarios

Scenario 3: L3 Network with IP Unnumbered



Summary

- Parallel process, multiple system can be configured at the same time.
- Dramatically reduce deployment time
- Reduce the chance of configuration errors.
- Tutorials, Blogs, Examples
  - https://xrdocs.github.io/software-management/
  - https://github.com/ios-xr/iosxr-ztp-python
  - Network Field Day 17:
    - https://vimeo.com/253197077
    - https://vimeo.com/253197037
    - https://github.com/akshshar/nfd17

Thank You