Power On Auto-Provisioning

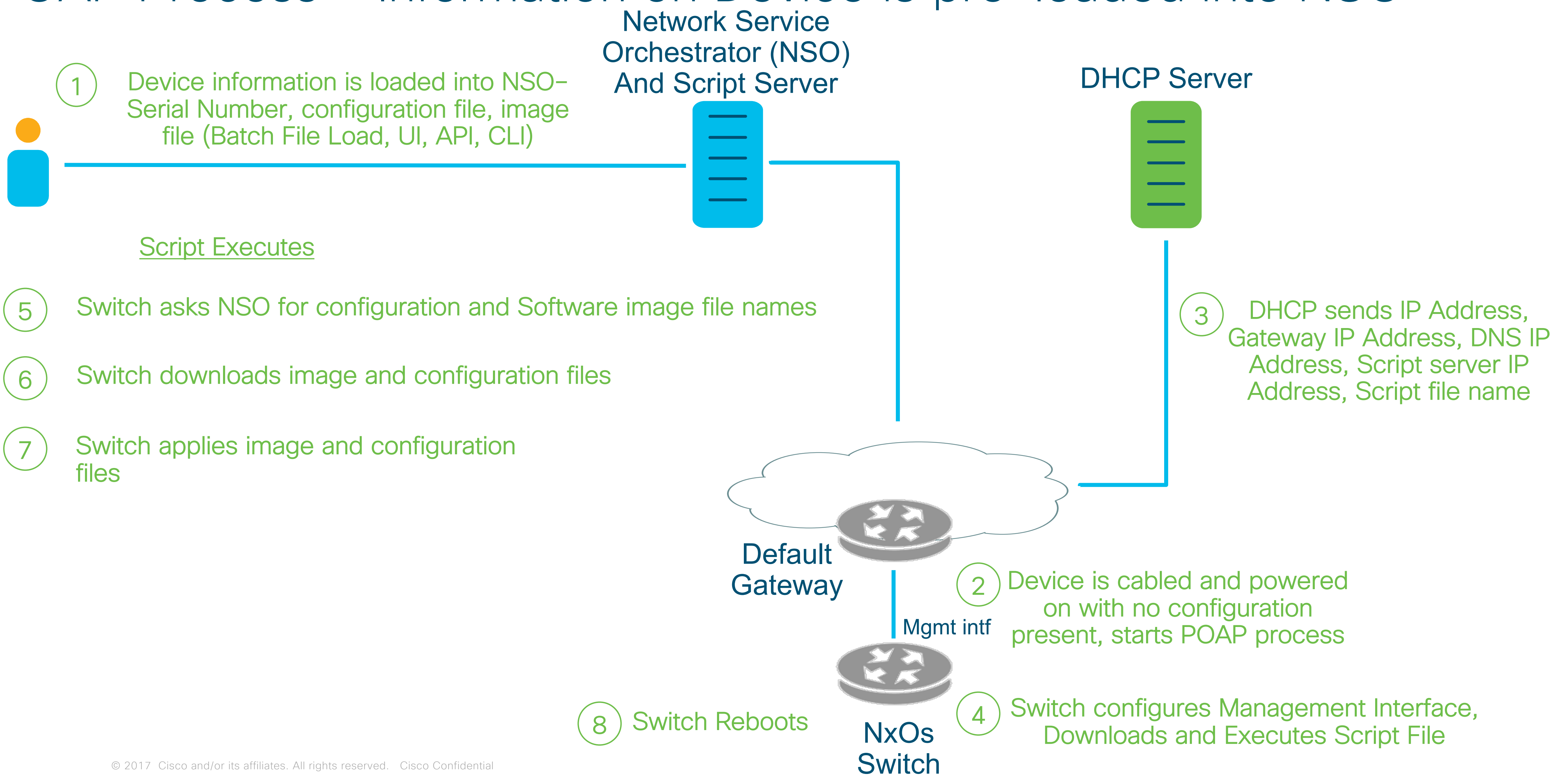
Utilize NSO as the Device Configuration Repository

Johan Nemitz

Agenda

- POAP Process overview
- POAP Setup
  - Network
  - DHCP Server
  - Script Server
  - NSO
  - POAP script file
- Demonstration

POAP Process – Information on Device is pre-loaded into NSO



1 Device information is loaded into NSO- Serial Number, configuration file, image file (Batch File Load, UI, API, CLI)

2 Device is cabled and powered on with no configuration present, starts POAP process

3 DHCP sends IP Address, Gateway IP Address, DNS IP Address, Script server IP Address, Script file name

4 Switch configures Management Interface, Downloads and Executes Script File

<u>Script Executes</u>

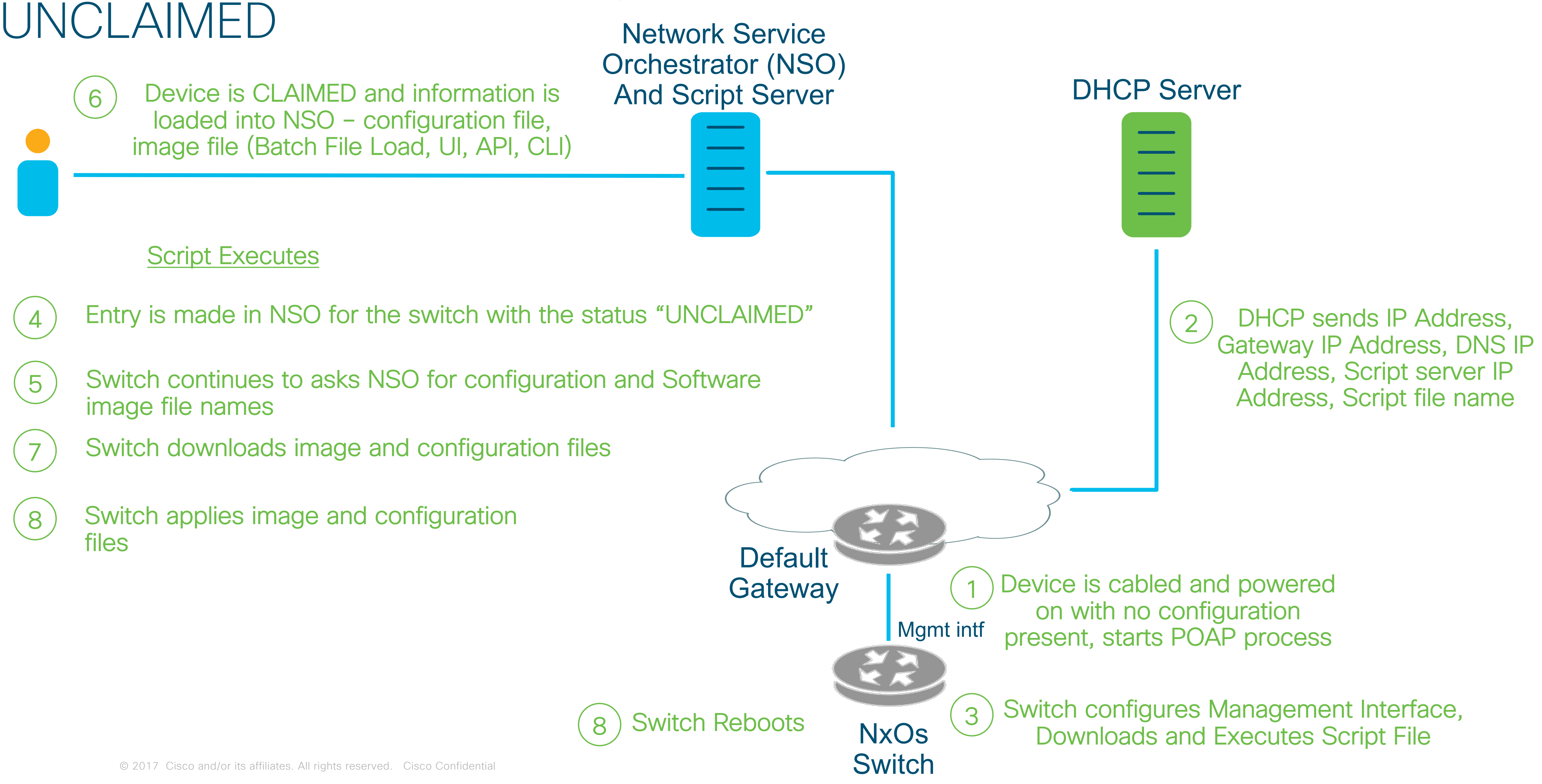
5 Switch asks NSO for configuration and Software image file names

6 Switch downloads image and configuration files

7 Switch applies image and configuration files

8 Switch Reboots

POAP Process – Device Registers itself first with NSO as UNCLAIMED



6 Device is CLAIMED and information is loaded into NSO – configuration file, image file (Batch File Load, UI, API, CLI)

Script Executes

4 Entry is made in NSO for the switch with the status “UNCLAIMED”

5 Switch continues to asks NSO for configuration and Software image file names

7 Switch downloads image and configuration files

8 Switch applies image and configuration files

2 DHCP sends IP Address, Gateway IP Address, DNS IP Address, Script server IP Address, Script file name

1 Device is cabled and powered on with no configuration present, starts POAP process

3 Switch configures Management Interface, Downloads and Executes Script File

8 Switch Reboots

POAP Setup - Network

- DHCP Server must be reachable from the Switch
  - Switch – UDP port 68
  - DHCP Server – UDP port 67

- Script Server must be reachable from the Switch
  - tftp, scp, https, etc. (Demo uses tftp for the script file download and http for the configuration and software image files)

- NSO Server must be reachable from the Switch
  - http or https for POAP communication (Demo uses http) –switch configuration management by NSO day 2 requires ssh or telnet access from/to NSO

POAP Setup – DHCP Server

- Management IP address pool
- DNS server IP Address
- Gateway IP Address
- Script server IP Address
- POAP script path/file name

IOS DHCP Example Config

```
subnet 10.255.0.0 netmask 255.255.255.0 {
    range 10.255.0.21 10.255.0.30;
    option domain-name-servers 171.70.168.183;
    option routers 10.255.0.10;
    option tftp-server-name “tftp://10.255.0.20/";
    option bootfile-name "poap.py";
}
```

Ubuntu DHCP Server Config Example

```
subnet 10.255.0.0 netmask 255.255.255.0 {
    range 10.255.0.21 10.255.0.30;
    option domain-name-servers 171.70.168.183;
    option routers 10.255.0.10;
    option tftp-server-name “tftp://10.255.0.20/";
    option bootfile-name "poap.py";
}
```

POAP Setup – Script File Server

- TFTP or HTTP Server must be used for POAP script retrieval

CentOS TFTP setup main steps

```
1. sudo yum install tftp tftp-server* xinetd*
2. edit /etc/xinetd.d/tftp
    set disable to no
    Set server_args to -s <tftp-file-direcory>
3. Start xinetd and tftp processes
```

- Method to be used for other downloads (configuration file, and software image file)
  - This is configured in the POAP script (one of the `copy uri` commands the switch accepts, eg. SCP/(S)FTP/HTTP(S)/TFTP)

POAP Setup – NSO Server

- Create NSO Service Package
- Register device(s)
  - UI – Individual Registrations
  - CLI – load from file (XML)
  - API – load from file (JSON or XML)
  - Custom Batch Program – convert custom formatted file to XML or JSON and load using an API call or CLI command

```
{
    "collection": {
        ”poap-service:device": [
            {
                "id": ”AD345FGD4",
                "state": ”CLAIMED",
                "image-file": "nxos.7.0.3.I6.1.bin",
                "configuration-file": " AD345FGD4-poap.cfg”
            },
        ]
    }
}
```

```
module poap {
...
  container poap {
    list device {
      key id;
      leaf id {
        type string;
      }
      leaf poap-state {
        type enumeration {
          enum "UNCLAIMED";
          enum "CLAIMED";
        }
      }
      leaf device-state {
        type enumeration {
          enum "UNKNOWN";
          enum "PROVISIONING";
          enum "PROVISIONED";
          enum "ONLINE";
          enum "OFFLINE";
        }
        default "UNKNOWN";
      }
      leaf image-file {
        type string;
      }
      leaf configuration-file {
        type string;
      }
      leaf management-ip-address {
        type inet:ipv4-address;
      }
    }
...
}
```

POAP Setup – POAP Script

Main Script Steps

1. Request Configuration filename and Software Image filename from NSO
2. If Device Serial number is not available in NSO, Create and entry in NSO with state UNCLAIMED
3. If the configuration filename of software image filename is unavailable report issue and restart the POAP process
4. Update the state of device in NSO to PROVISIONING
5. Verify storage space on the device
6. Copy the Configuration file from the Script Server
7. If the Software Image file does not exist on the switch download the file to the switch
8. Configure the switch to boot to the Software Image selected
9. Update the state of device in NSO to PROVISIONED
10. Reboot switch

POAP Setup – POAP Script

Code Example - Retrieve Information from NSO

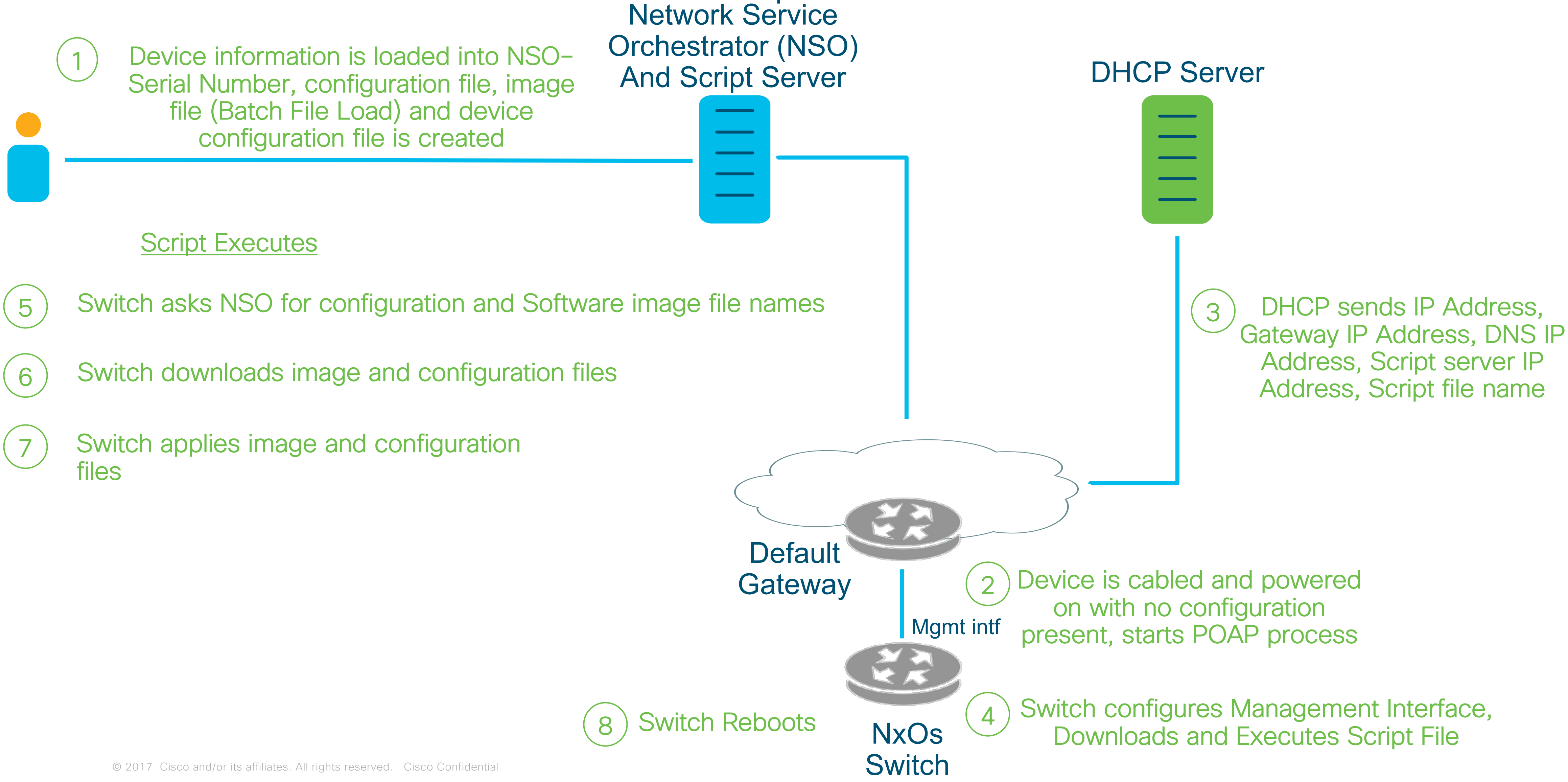
```
import httplib
import base64
import string
import json
from cli import *
nso_ip = "10.95.5.60"
nso_port = 8080
nso_username = "admin"
nso_password = "admin”
serial_number = cli('show inventory chassis | grep SN:').rstrip().split(" ")[-1]
request_path = "http://phx2-p5-nso:8080/api/running/poap/device/"+serial_number
url = "http://" + nso_ip + ":" + str(nso_port) + request_path
auth = base64.encodestring('%s:%s' % (nso_username, nso_password)).replace('\n', '')
headers = {"Accept": "application/vnd.yang.data+json", "Authorization": "Basic %s" % auth}

conn = httplib.HTTPConnection(host=nso_ip, port=nso_port)
conn.request(method="GET", url=request_path, headers=headers)
response_data = json.loads(conn.getresponse().read())
image_file = response_data['dns-service:device']['image-file']
configuration_file = response_data['dns-service:device']['configuration-file']

options["target_system_image"] = image_file
options["source_config_file"] = configuration_file
conn.close()
```

Demonstration

Demo 1– Information on Device is pre-loaded into NSO



1 Device information is loaded into NSO– Serial Number, configuration file, image file (Batch File Load) and device configuration file is created

Network Service Orchestrator (NSO) And Script Server

DHCP Server

Script Executes

5 Switch asks NSO for configuration and Software image file names

6 Switch downloads image and configuration files

7 Switch applies image and configuration files

3 DHCP sends IP Address, Gateway IP Address, DNS IP Address, Script server IP Address, Script file name

Default Gateway

2 Device is cabled and powered on with no configuration present, starts POAP process

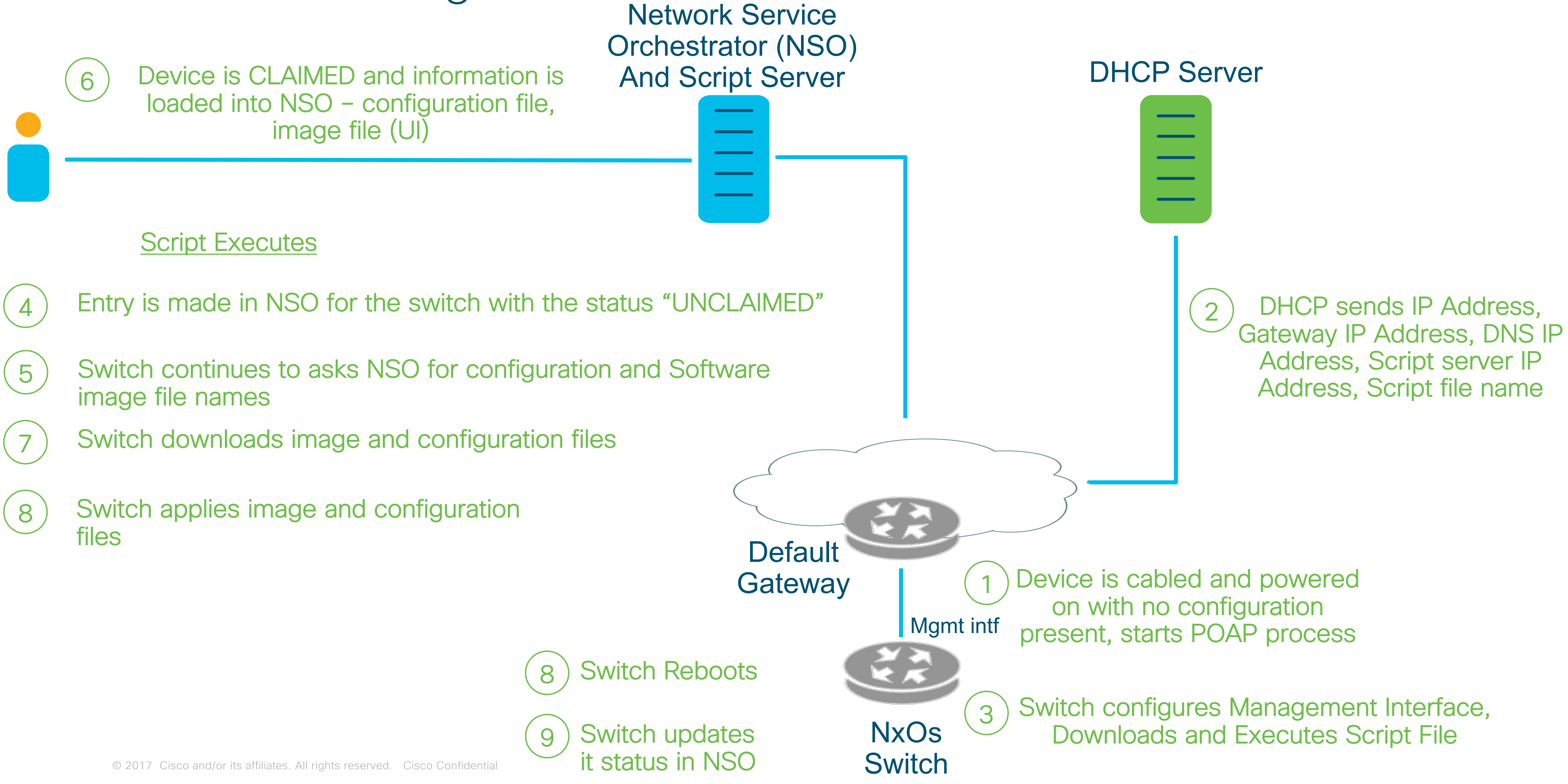
Mgmt intf

4 Switch configures Management Interface, Downloads and Executes Script File

8 Switch Reboots

NxOs Switch

Demo 2– Device Registers itself first with NSO as UNCLAIMED



6 Device is CLAIMED and information is loaded into NSO – configuration file, image file (UI)

Network Service Orchestrator (NSO) And Script Server

DHCP Server

Script Executes

4 Entry is made in NSO for the switch with the status “UNCLAIMED”

5 Switch continues to asks NSO for configuration and Software image file names

7 Switch downloads image and configuration files

8 Switch applies image and configuration files

2 DHCP sends IP Address, Gateway IP Address, DNS IP Address, Script server IP Address, Script file name

Default Gateway

1 Device is cabled and powered on with no configuration present, starts POAP process

Mgmt intf

8 Switch Reboots

9 Switch updates it status in NSO

NxOs Switch

3 Switch configures Management Interface, Downloads and Executes Script File