# Increasing security and simplify operations with VXLAN EVPN in Campus and DC Networks

Patricio Cachaço, System Engineer

June 2023

# Agenda

What is VXLAN with BGP EVPN ?

BGP EVPN Fabric Drivers

Enterprise BGP EVPN Solution

Underlay and Overlay Networks

Secure Fabric and Microsegmentation

EVPN Fabric Automation

Q & A

# Why VXLAN?

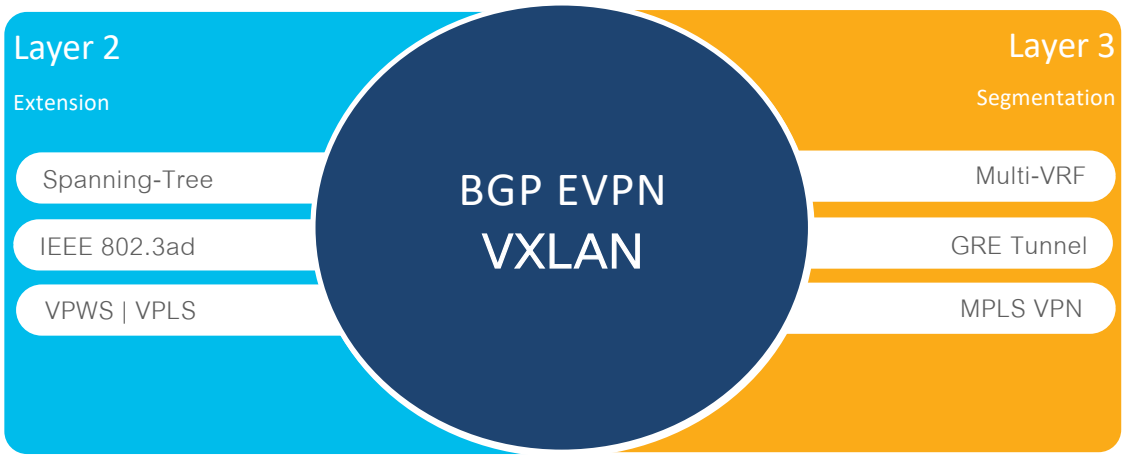
VXLAN provides a Network with  Segmentation, IP Mobility, and Scale

- “Standards” based Overlay (RFC 7348)
- Leverages Layer-3 ECMP – all links forwarding
- Increased Name-Space to 16M identifier
- Integration of Physical and Virtual
- It’s SDN ?

# What is VXLAN with BGP EVPN?

- Standards based Overlay (VXLAN) with Standards based Control-Plane (BGP)
- Layer-2 MAC and Layer-3 IP information distribution by Control-Plane (BGP)
- Forwarding decision based on Control-Plane (minimizes flooding)
- Integrated Routing/Bridging (IRB) for Optimized Forwarding in the Overlay
- Multi-Tenancy At Scale

# Traditional Network Transition




**EVPN Evolution**

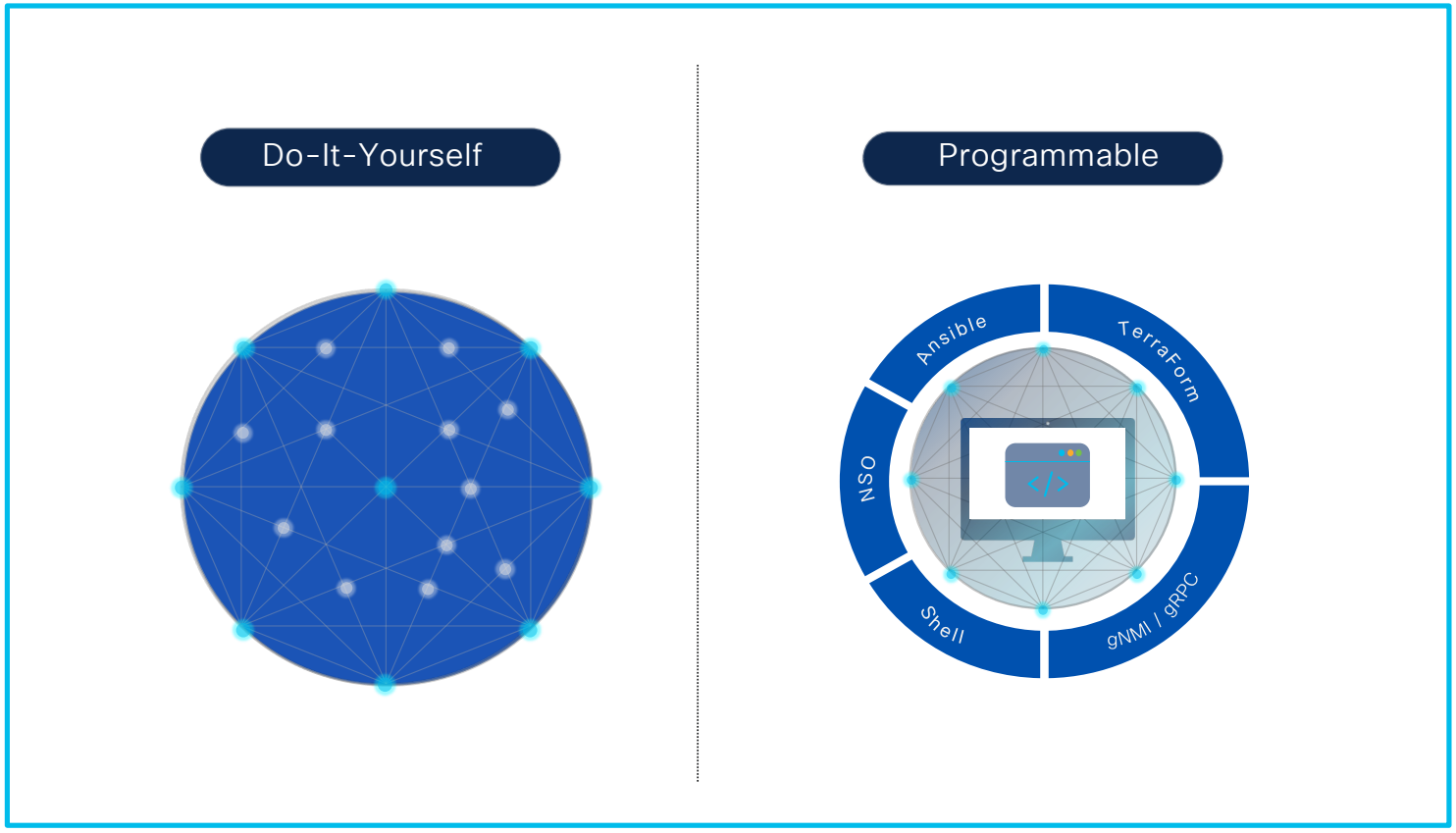
- Product transition drives architecture transitions
- Convergence of traditional L2 overlay to simplified and scalable fabric
- Transition classic L3 overlays to enterprise-grade scalable fabric
- Unified end-to-end common fabric architecture reducing cost and complexity

# BGP EVPN Fabric Drivers

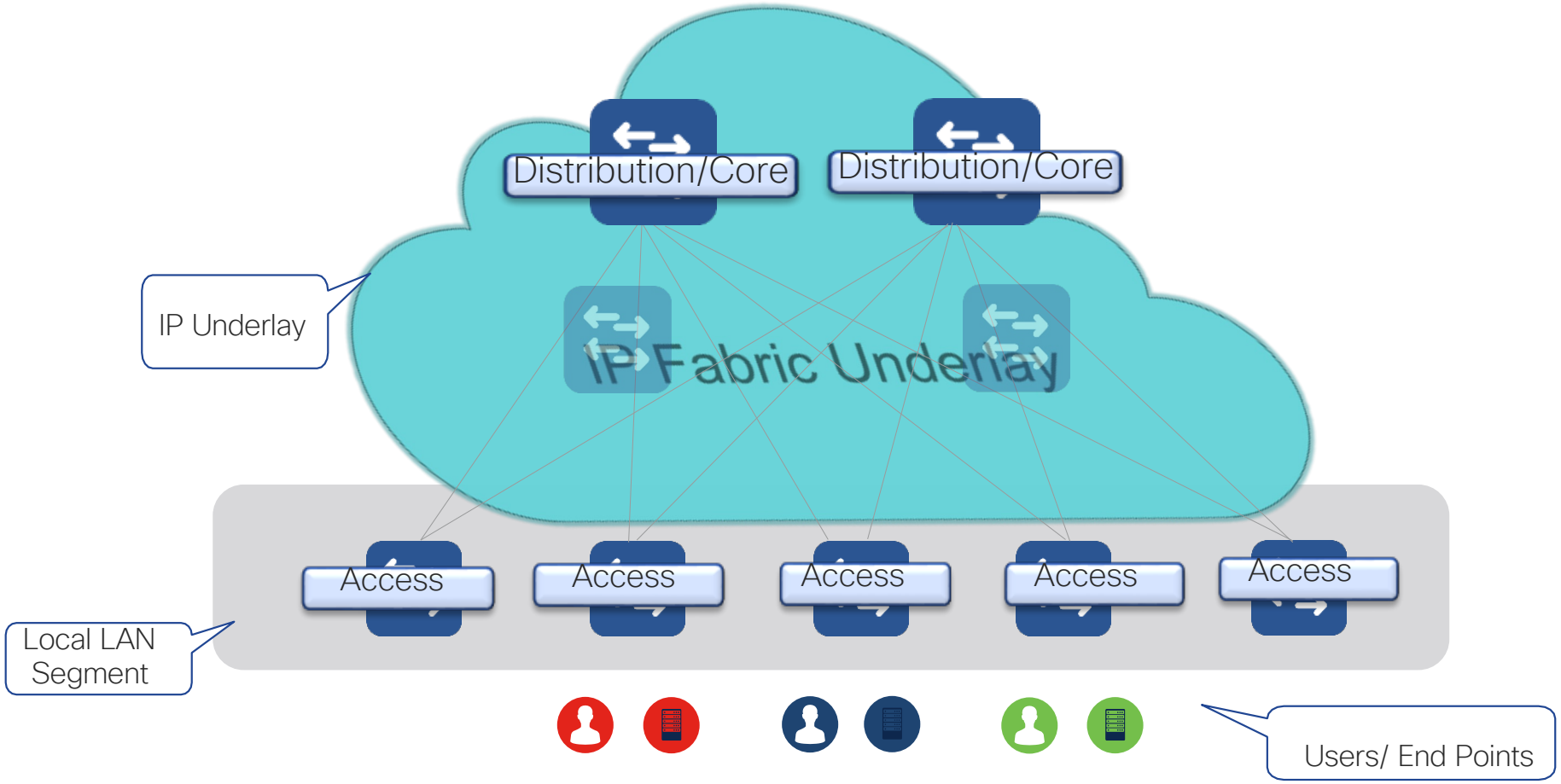
| | |
|---|---|
| Industry Standard | Multi-vendor IT strategy |
| One Fabric Architecture | Unified operation across – Campus \| DC \| WAN |
| Proven and Scalable | BGP Protocol History. Minimum new learning curve |
| Hierarchical Fabric Domain | Multi-tier Overlay network architecture |
| Flexible Overlay | Use-case driven customize Overlay networks Types and Topologies |

cisco

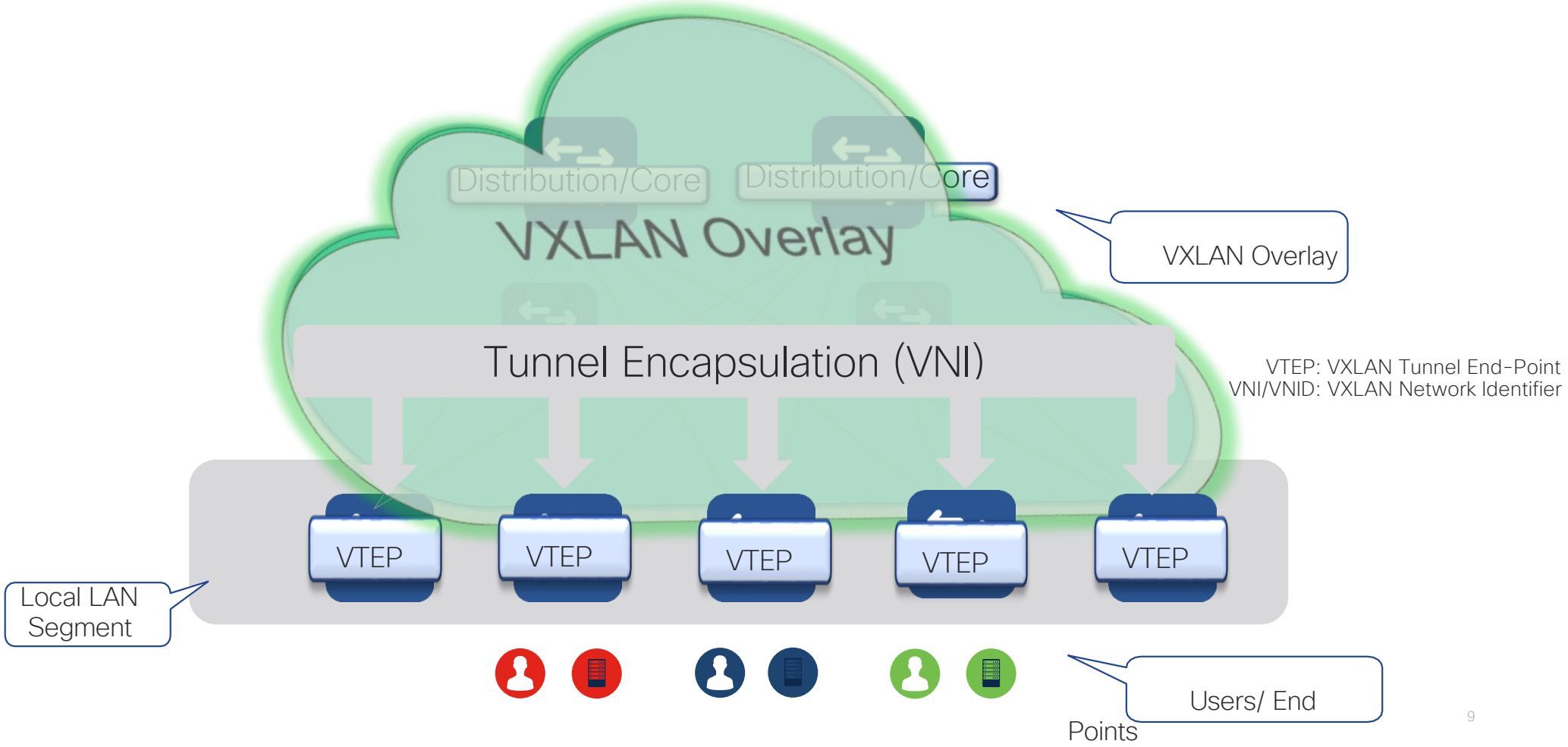
# Enterprise BGP EVPN Solution



# VXLAN Taxonomy – Underlay Network

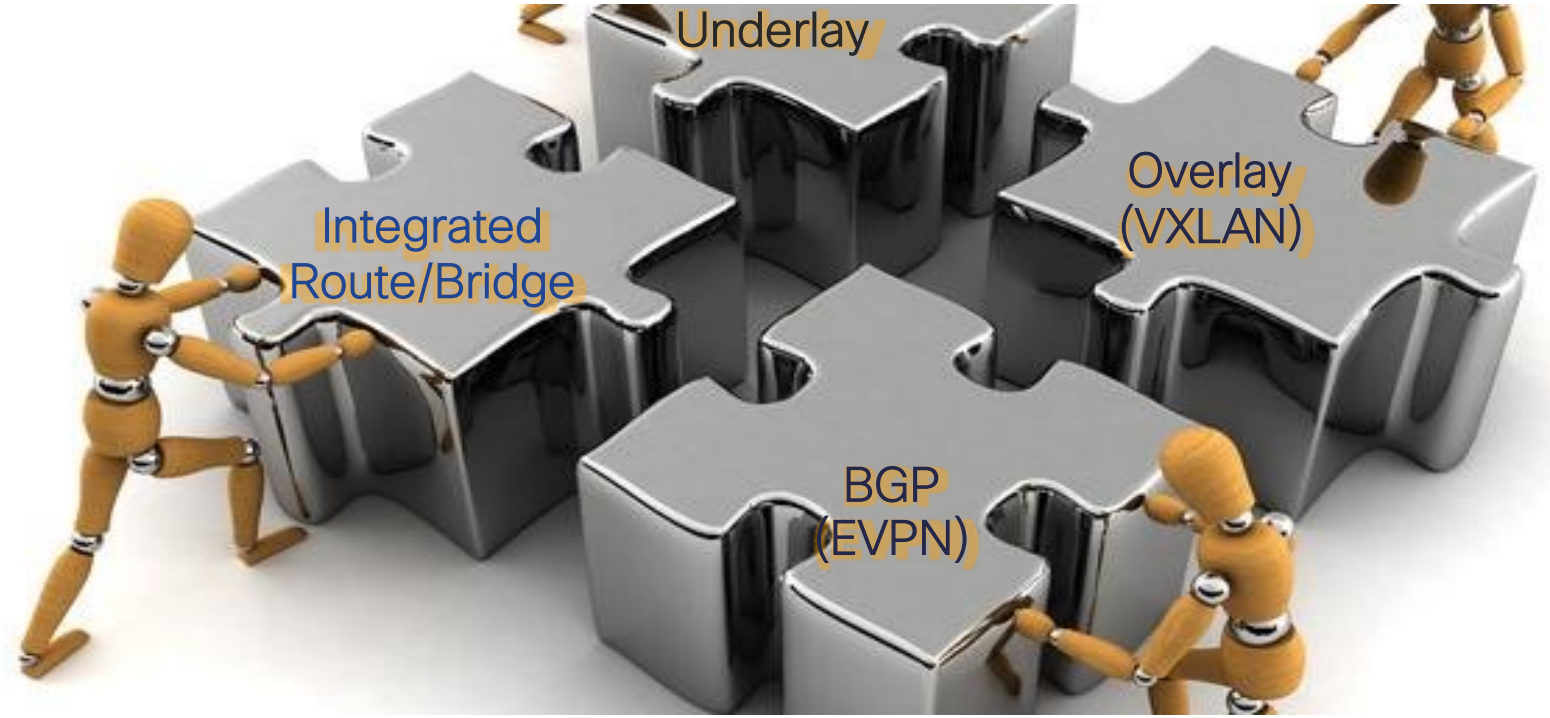


# VXLAN Taxonomy – Overlay Network



# Getting the Puzzle Together!

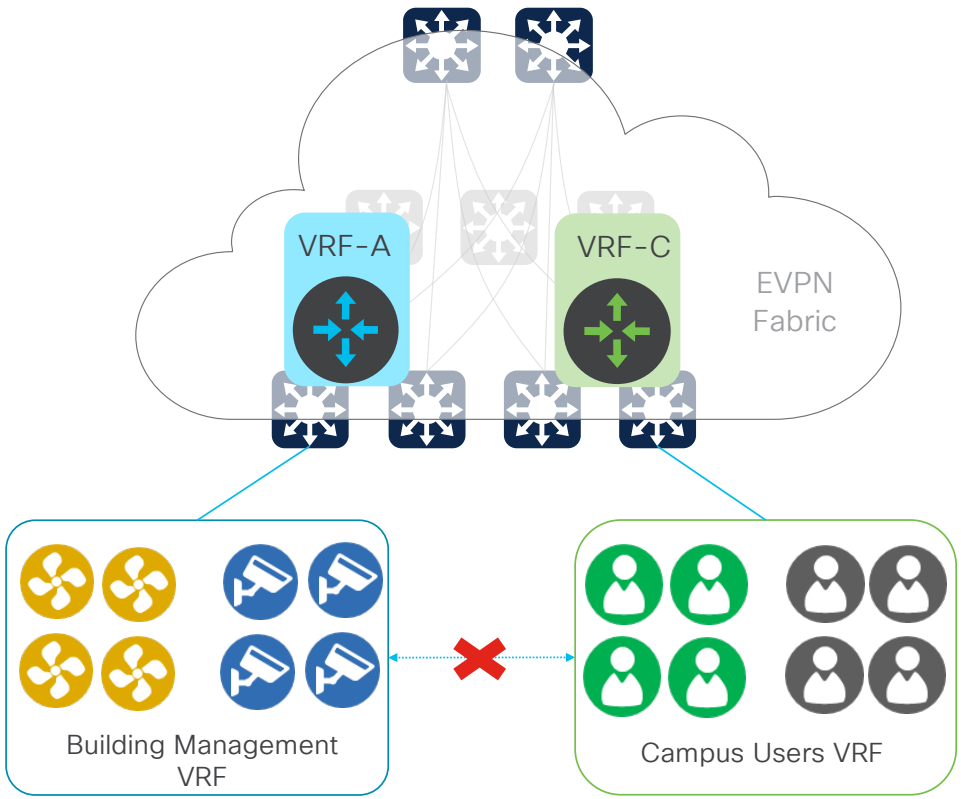
## Optimized Networks with VXLAN



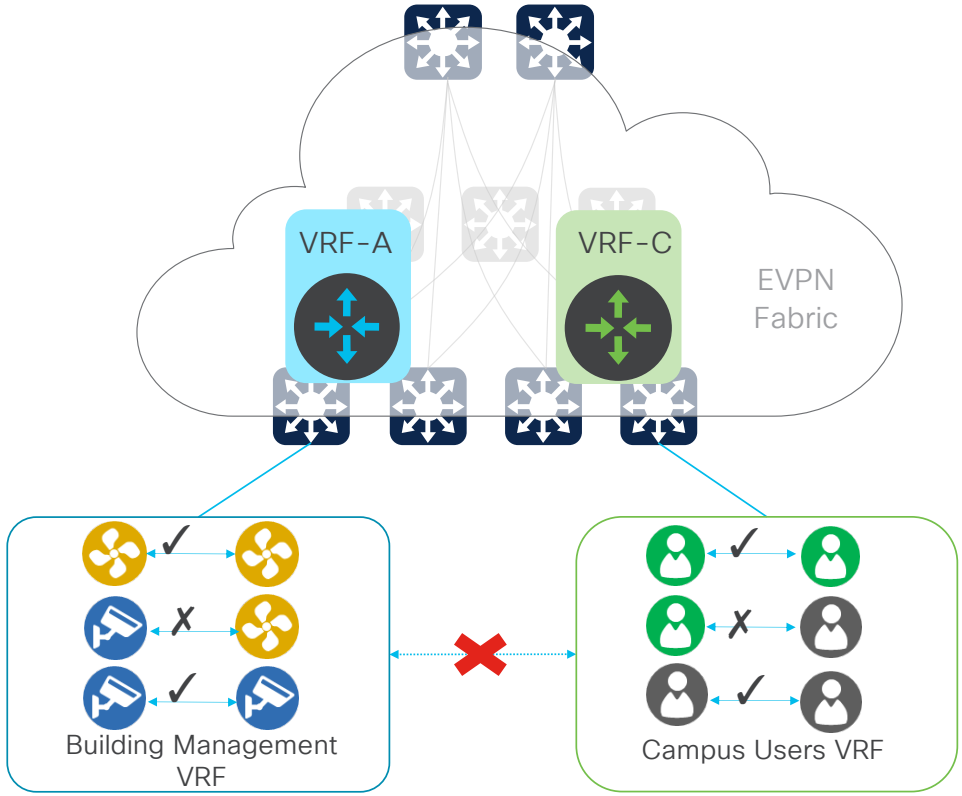
# Secure EVPN Fabric Microsegmentation

cisco

# Fabric Segmentation Options



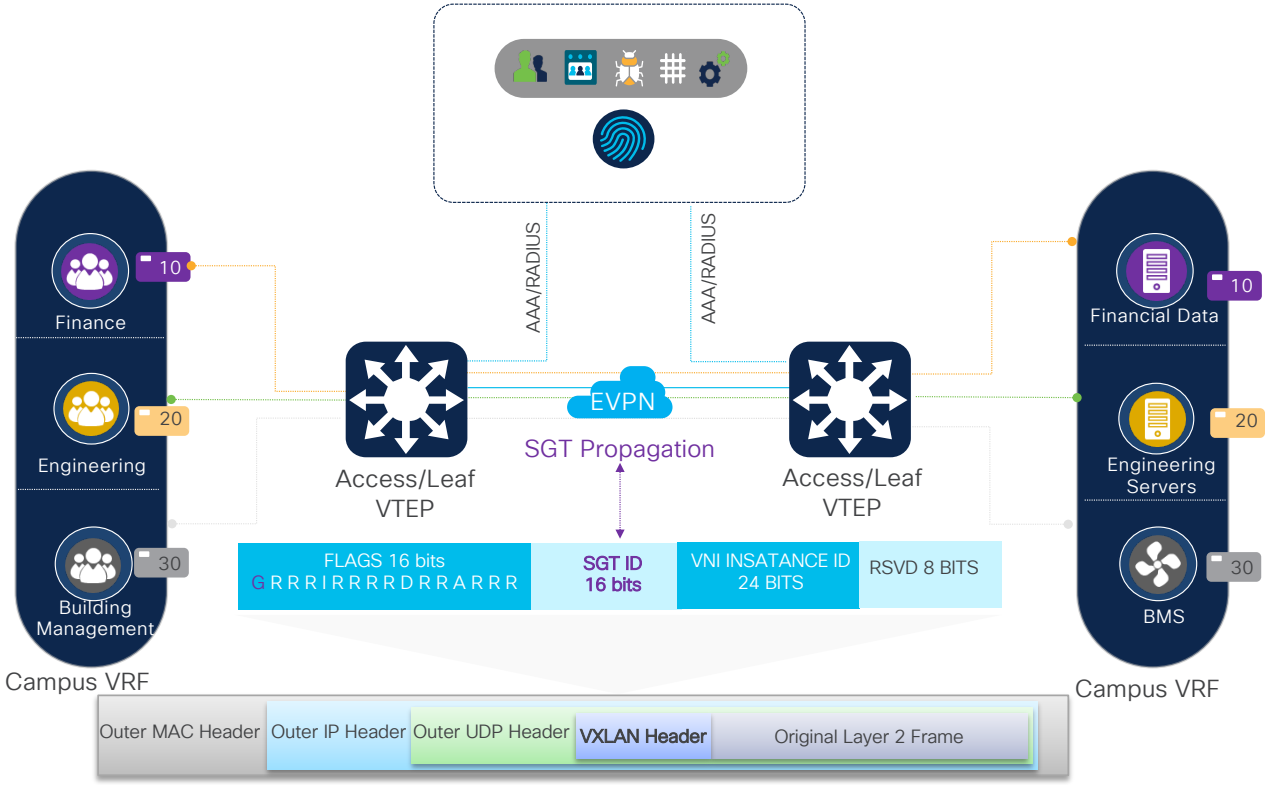

Macro Segmentation: No communication between VRF's




Micro Segmentation: Second level Segmentation between groups within a VRF

cisco

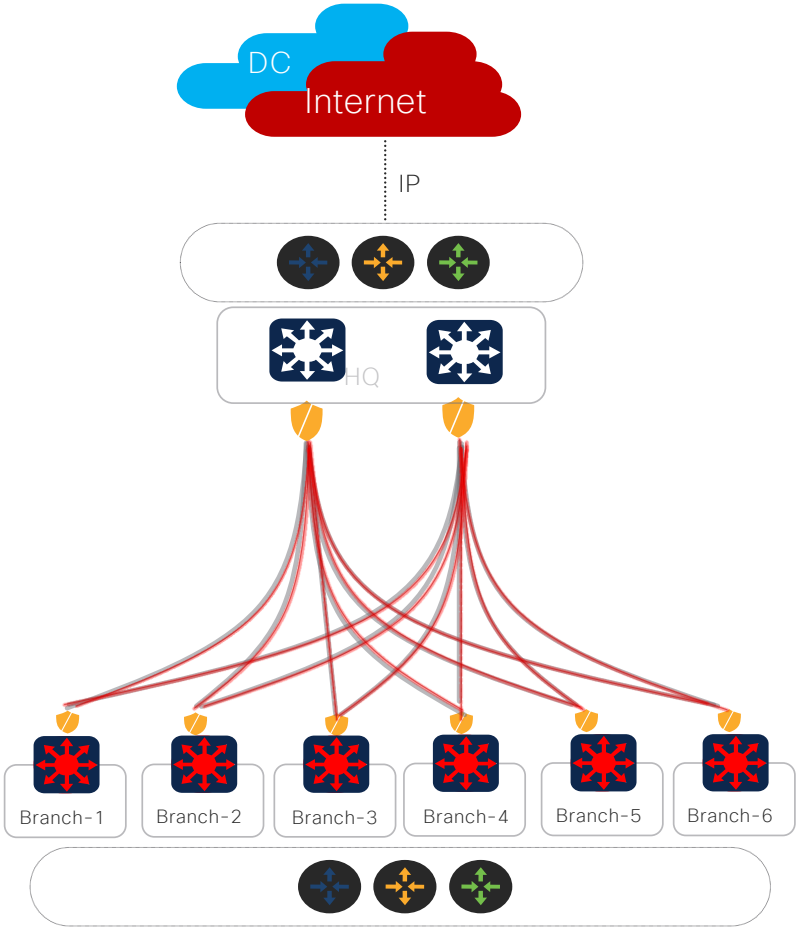
# BGP EVPN – Role based Access Control




- Role Based Access Control
- Scalable policy based on User role Dynamic or Static Policy enforcement
- Centralized Policy Management for Dynamic policy provisioning

# BGP EVPN over IPsec

## Secure Fabric




| | |
|---|---|
| Layer 2 Extension | BGP EVPN |
| Layer 3 Overlay | BGP EVPN |
| Underlay-1 | OSPF/BGP |
| Secure Overlay | IPsec |

**Key Benefits**

- Scalable Segmentation over IPSEC
- Secure End-to-End Fabric



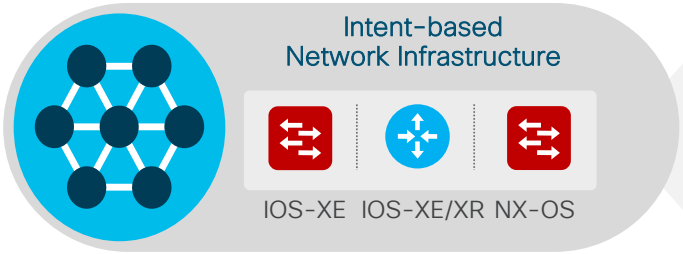
14

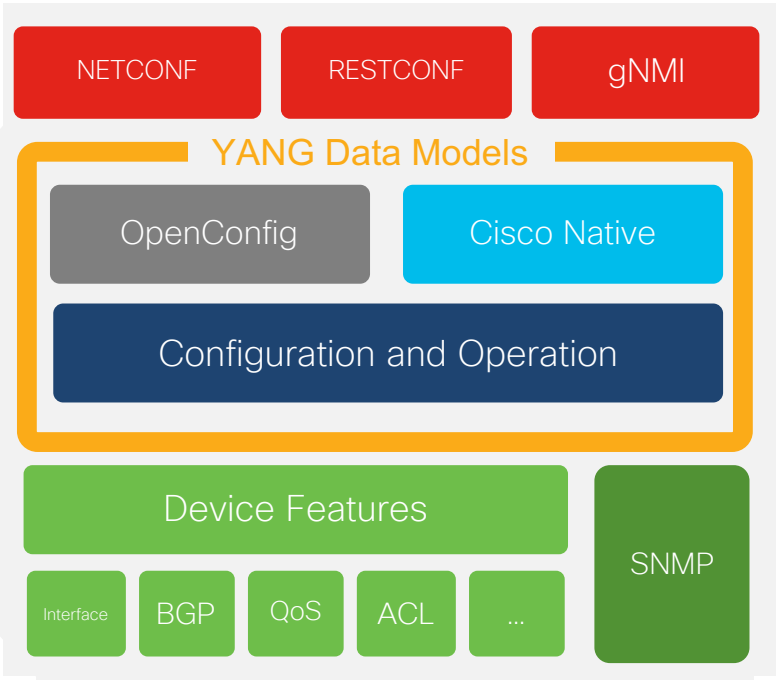
# EVPN Fabric Automation

# IOS XE Programmability

The NETCONF, RETCONF and gNMI are programmatic interfaces that provide additional methods for interfacing with the IOS XE device

YANG data models define the data that is available for configuration and streaming telemetry




IETF

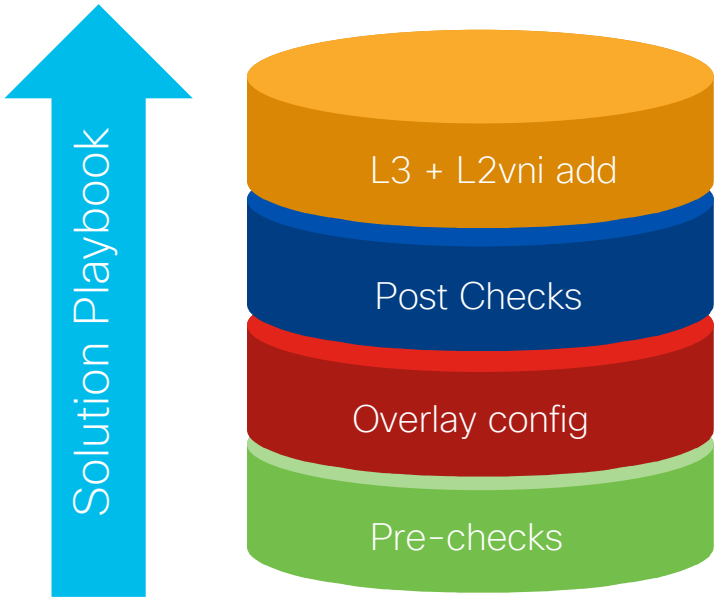
OPENCONFIG

IEEE

CISCO

cisco

# EVPN Ansible – Solution Playbook




**Same playbook to add L3/L2 VNI's**
Eg: Add one or multiple L3/l2vni using same playbook

**Framework for post-check**
Eg: BGP status up/down, overlay ping checks

**Solution level deployment**
Eg: Ipv4 + Ipv6 + TRM in a single playbook

**Framework for pre-checks**
Eg: License check, underlay reachability check

## Simple to Use

- Single playbook for complete solution
- Single inventory file to add Leaf/Spine variables

Get playbooks below

Ansible Playbooks

# EVPN Ansible – Feature level Playbook

| | |
|---|---|
| playbook_access_add_preview.yml | adding L2VNI and L3VNI |
| playbook_access_incremental_commit.yml | initial commit for release/2.x.x |
| playbook_access_incremental_preview.yml | initial commit for release/2.x.x |
| playbook_cleanup.yml | initial commit for release/2.x.x |
| playbook_dhcp_add_commit.yml | adding L2VNI and L3VNI |
| playbook_dhcp_add_preview.yml | adding L2VNI and L3VNI |
| playbook_dhcp_delete_commit.yml | dhcp incremental commit |
| playbook_dhcp_delete_preview.yml | dhcp incremental commit |
| playbook_output.yml | fix playbook_output |
| playbook_overlay_commit.yml | adding L2VNI and L3VNI |
| playbook_overlay_delete_commit.yml | ipv6_incremental |
| playbook_overlay_delete_generate.yml | initial commit for release/2.x.x |
| playbook_overlay_delete_ipv6_commit.yml | adding L2VNI and L3VNI |
| playbook_overlay_delete_ipv6_generate.yml | adding L2VNI and L3VNI |
| playbook_overlay_delete_ipv6_preview.yml | adding L2VNI and L3VNI |
| playbook_overlay_delete_preview.yml | initial commit for release/2.x.x |
| playbook_overlay_incremental_commit.yml | adding L2VNI and L3VNI |
| playbook_overlay_incremental_generate.yml | adding L2VNI and L3VNI |
| playbook_overlay_incremental_ipv6_commit.yml | adding L2VNI and L3VNI |
| playbook_overlay_incremental_ipv6_generate.yml | adding L2VNI and L3VNI |
| playbook_overlay_incremental_ipv6_preview.yml | ipv6_incremental |
| playbook_overlay_incremental_preview.yml | adding L2VNI and L3VNI |
| playbook_overlay_precheck.yml | initial commit for release/2.x.x |
| playbook_overlay_preview.yml | adding L2VNI and L3VNI |

Feature specific Playbooks

↓

Add/remove a feature

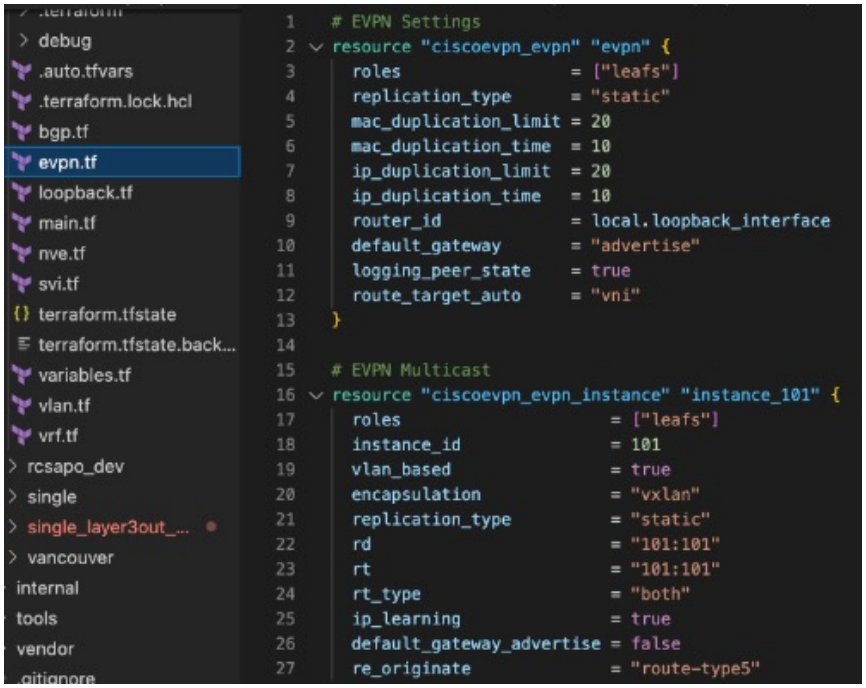
↓

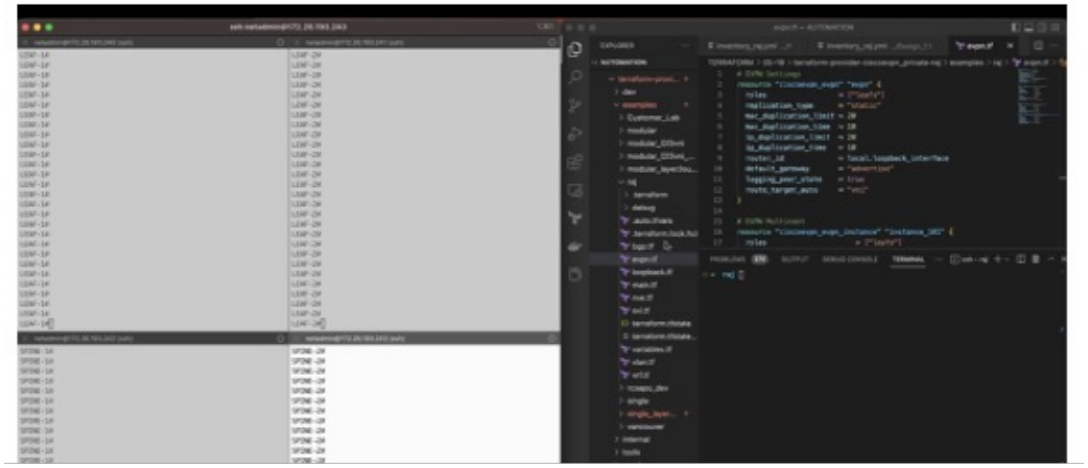
Get playbooks below

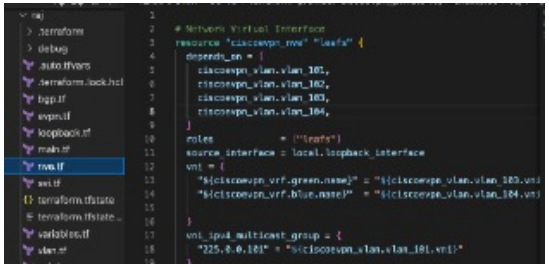
Ansible Playbooks

cisco

# EVPN **Automation** with Terraform





[Terraform Provider](#)
[Terraform Examples](#)

cisco


The bridge to possible