Rapid end to end Automation with Infrastructure as Code!

Turbocharge your services

Markus Harbeck
Principal Architect CX Germany

CCIE #8087
CCDE #20130015

Join at slido.com
#1136142

Cisco Webex App

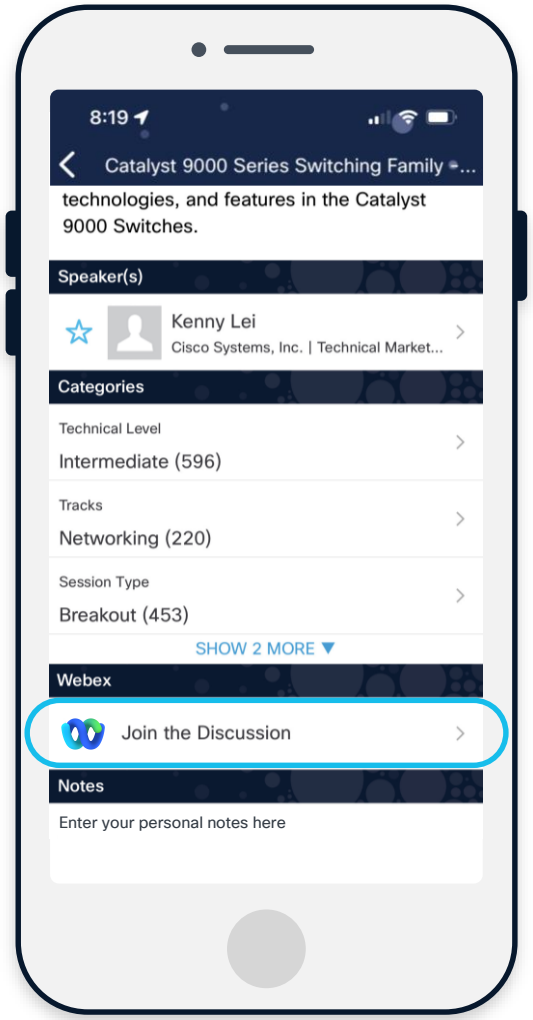
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 13, 2025.



Session Flow

- **Intro**
- **Automation in action – prepare the Demo**
- **Services as Code and Automation**
- Break
- **Network as Code domain automation**
  - Network as Code for ACI
  - Network as Code for SDWAN
  - Network as Code for Catalyst Center (SDA) with ISE
- **Next steps in automation**
- **Summary with Q&A**

Apply and automat
Break
Destroy
Talk

Short Hint:
“My English might be bad but although sexy”

Source: Henning Bornemann –
“Thank you for Deutsche Bahn”

Welcome to Cisco Live! Is this your first time to Cisco Live San Diego?

I am ...

...a network admin

And you ?

CCIE #8087 CCDE #20130015

Who is Markus Harbeck?

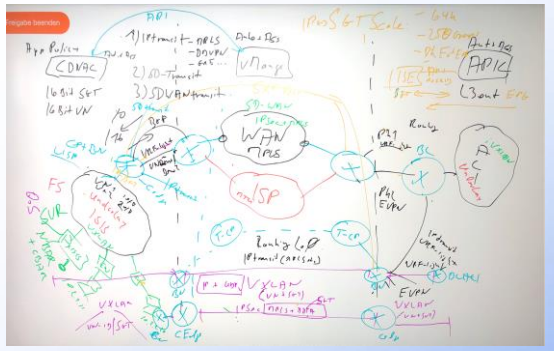
Personal

- Location: Eschborn, Germany (near Frankfurt) living in Bavaria
- Interests: My family, my two kids, my dog, horseback riding, motor cycling, hiking, flying drones for fawn rescuing

My Background

- CLI Junkie since 1996 for all Routing and Switching
- Joined Cisco October 2010
- Before: 12 years, operations, engineering, application engineering
- Orchestration and Cross Domain
- Analytics, assurance, automation and migration projects

→ first guy who compared horses and (SDN) intend



Copyright by Saskia

My kids view on cross domain network design

Copyright by Hanna

You can do everything yourself with infinite time and money

Unique design

Long build time

Questionable outcome

Extensive skillset

No warranty

No anticipated costs

Focus on build

Service as Code to let you focus

What are your top challenges?

My network as designed doesn't match reality and all networks look different

Network changes consume too many resources with too many errors take too long

I spend more of my time reacting to network needs

I’m questioning network and security compliance

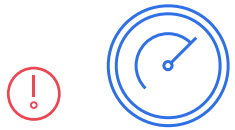
I’m challenged to apply automation

Skill shortages and low innovation

Network operations challenges

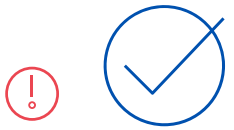
Lack of version control

- Difficult to track changes
- Potential configuration drifts
- Inconsistency
- Hard to audit and roll back
- Difficult troubleshooting



Lack of automation

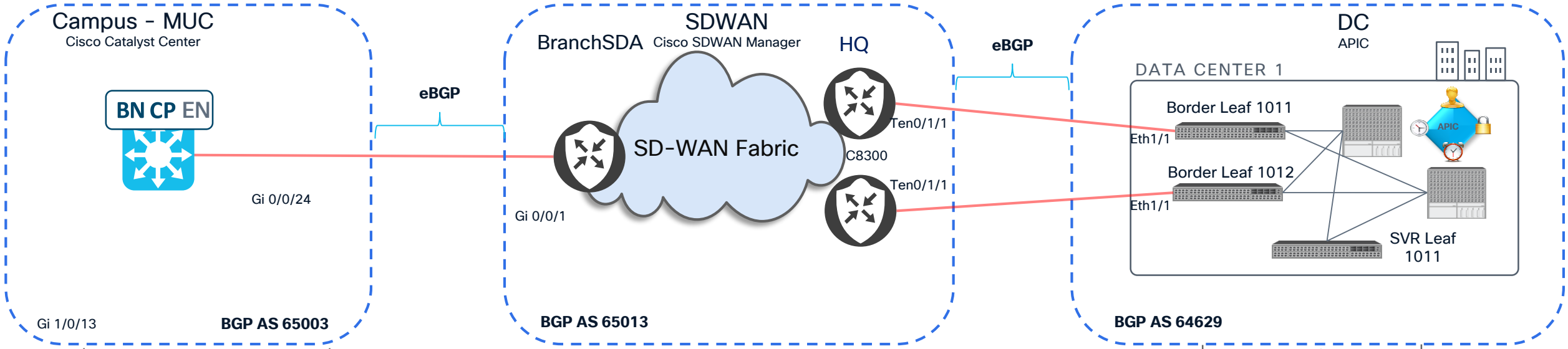
- Each deployment or provisioning becomes a unique manual process
- Automation is complex
- Each deployment or provisioning becomes a unique manual process



Lack of pre-production testing and validation

- changes made to the infrastructure and services are not thoroughly tested before being deployed in a production
- High risk off issues
- Unexpected downtimes

LAB Time → Yes let's start with a demo



Physical Demo Topology

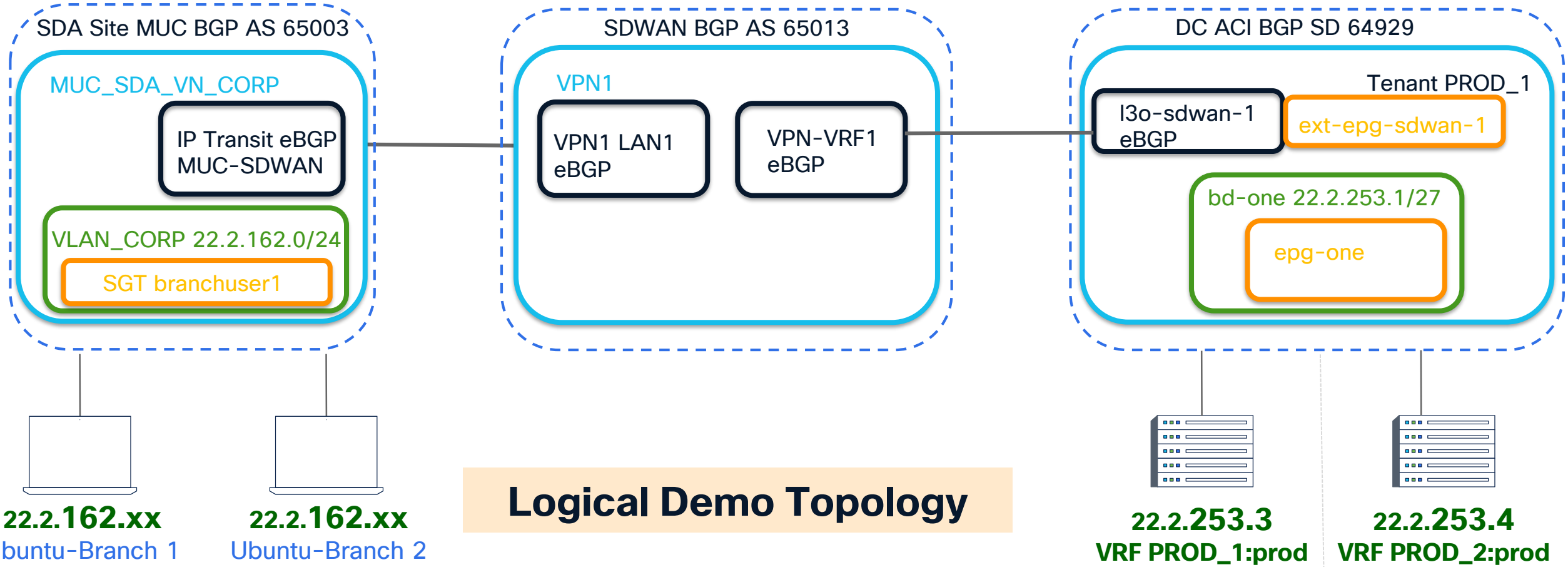
22.2.162.XX
Ubuntu-Branch 1

22.2.162.XX
Ubuntu-Branch 2

22.2.253.3
VRF PROD_1:prod
BLN14-Ubuntu-DC1

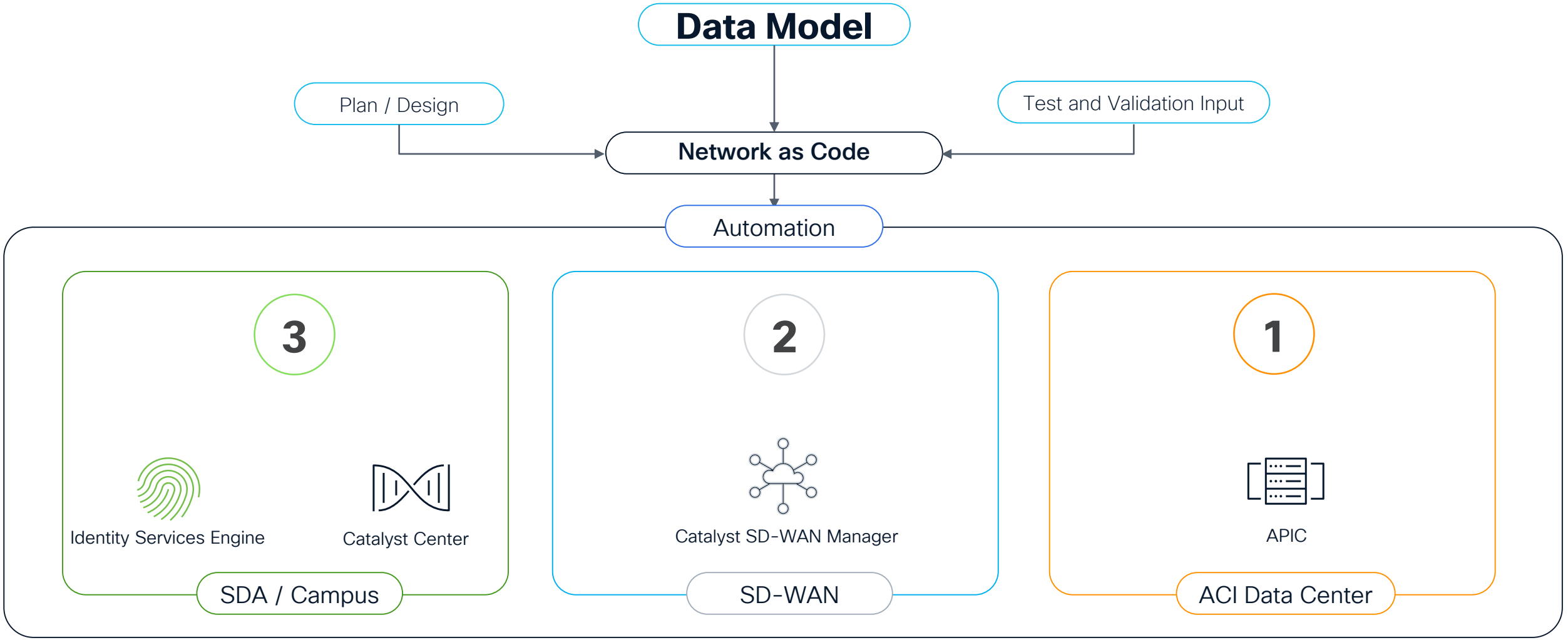
22.2.253.4
VRF PROD_2:prod
BLN14-Ubuntu-DC3

Demo Logical view



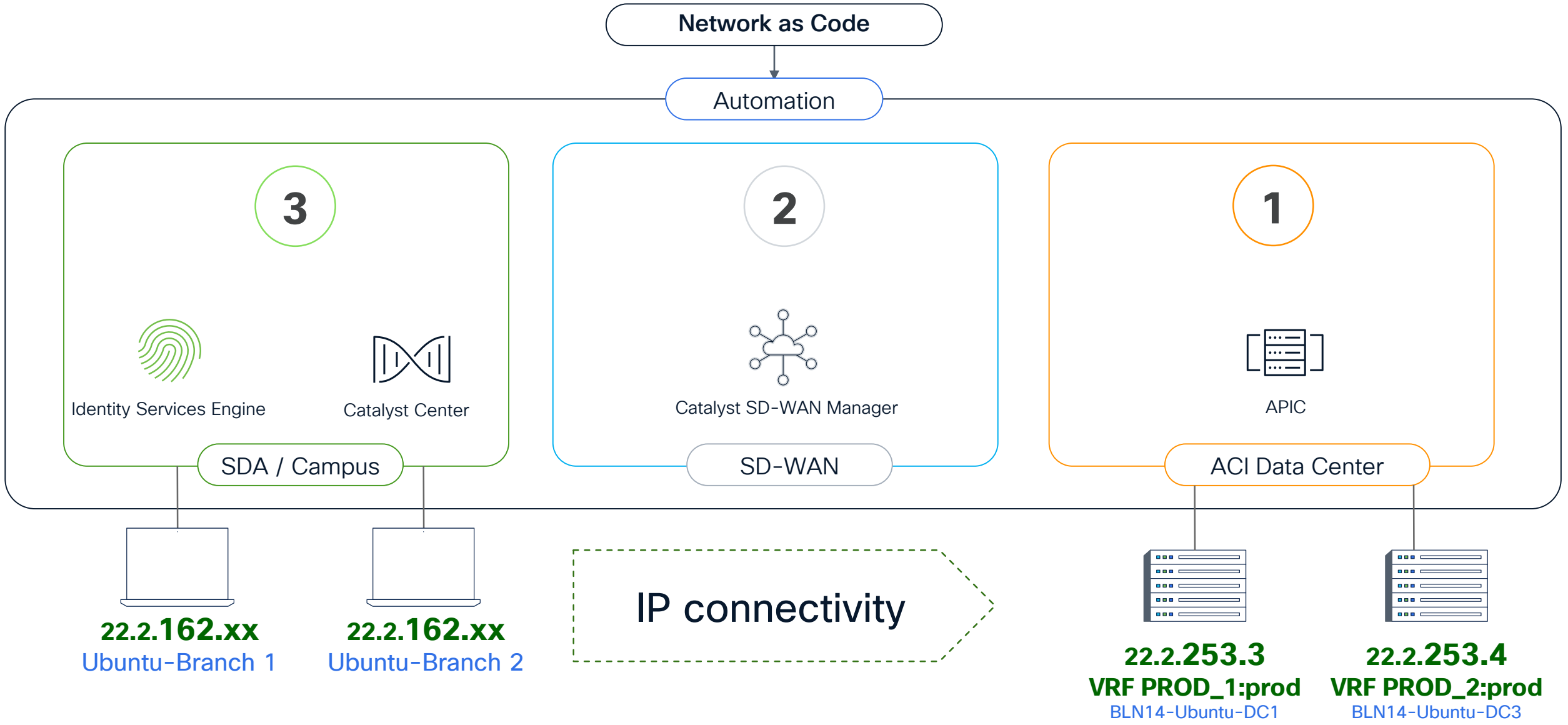
Logical Demo Topology

LAB / Demo Architecture view



Demo
The Demo environment – empty?

Expected Outcome – stay tuned we get there



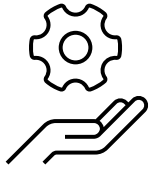
What is Services as Code? (SaC)

Network as Code

an introduction

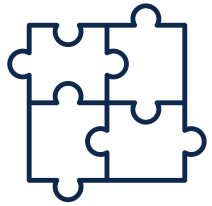
...as Code what?

Service as Code



The services around Network as Code
Asses, Plan, Design, Integrate, Document, guide and support

Automation

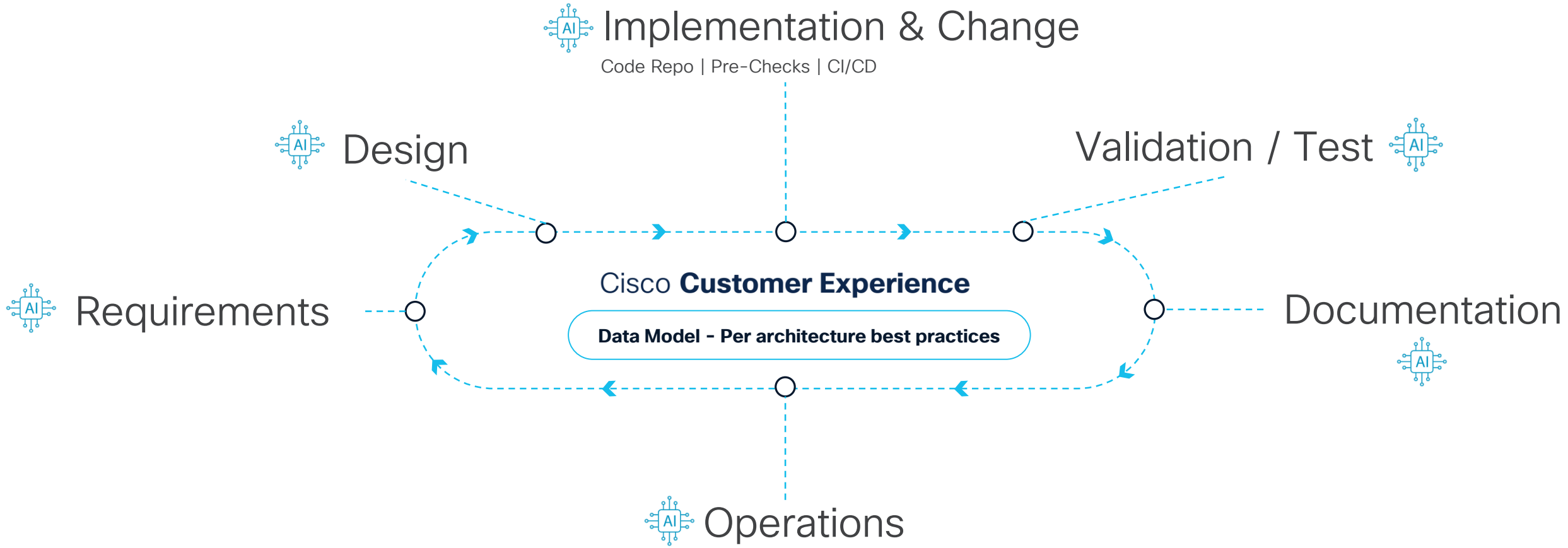


Life cycle

Network as Code

Based on Infrastructure as Code – enhanced by Cisco to simplify automation

Full digitization cycle



”Increase Efficiency with Service as Code for scalable, repeatable infrastructure automation."

Any idea ? Stay tuned – but its very Manual

Technical phases

Plan and Design

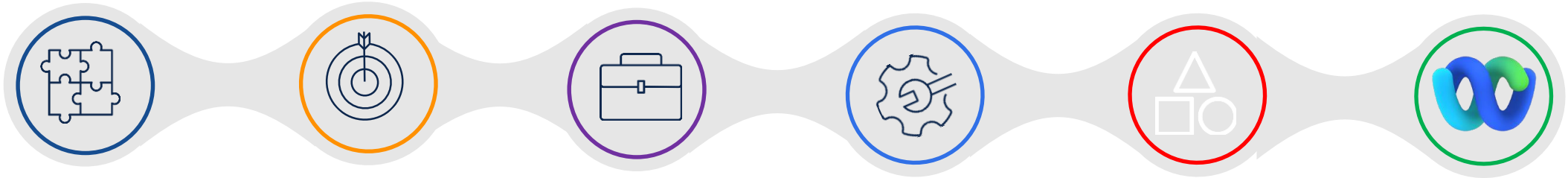
Use Blueprint
Declare requirements
Translate into Data

Implement Change in Production

Merge Test Branch into Main and configure the production network

Test Production Changes

Run Post checks



Branch and Test with pre Checks

Test the Configuration in a test branch, validate and ensure quality and function

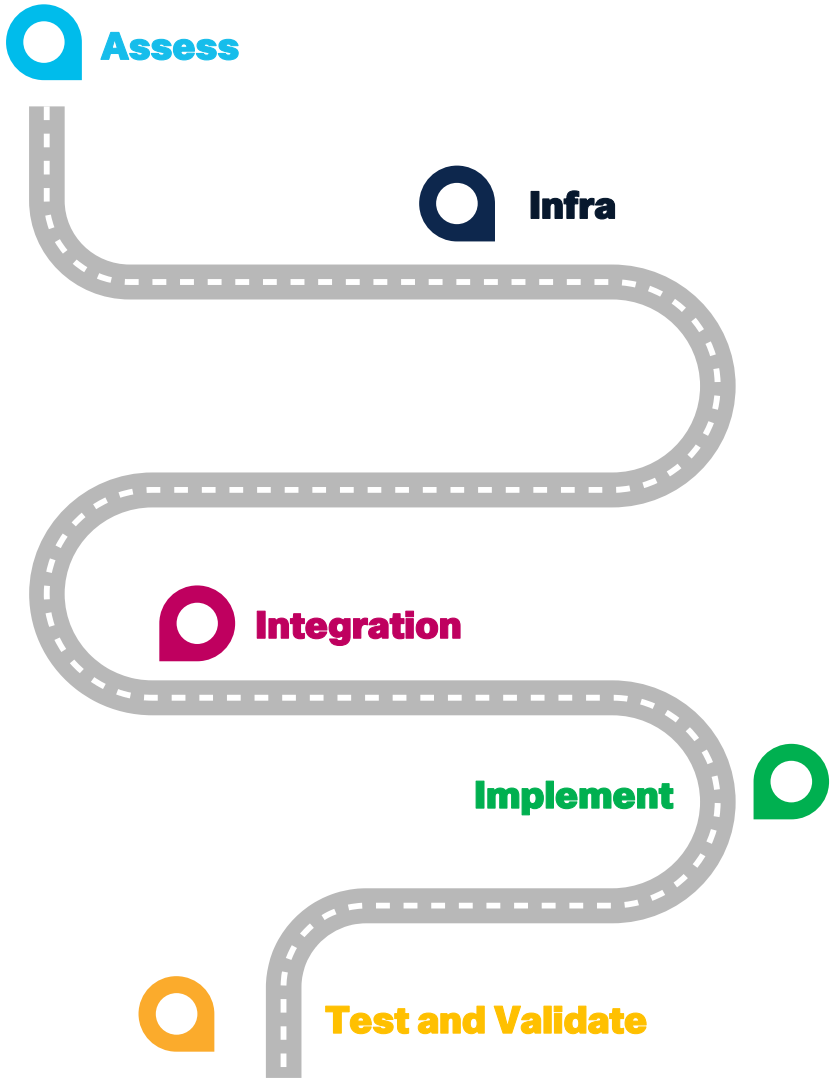
Run Post Validation

Validate applications and actual state of the network

Inform admin

Send information e.g. to Webex Teams

Our service journey



Assessment
of customer state to implement as Code capabilities

Infra
Existing DevOPS vs Cisco supported Framework definition

Integration
Into IPAM, ITSM, existing DevOPS tools

Implement
Build Main and Test branches (CICD)

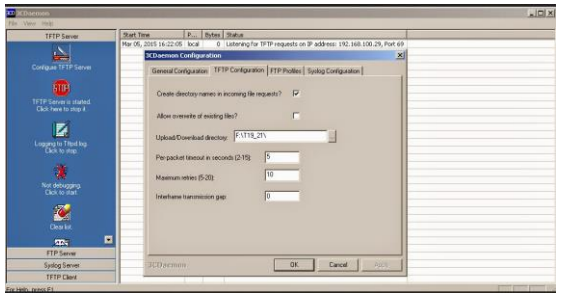
Test, Validate and Governance
Build Test and Validation setup for pre change and implementation phase

Challenges and optimization

NetOps Tools

```
(base) maharbec@MAHARBEC-M-92HK .ssh % ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=117 time=65.238 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=10.076 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 10.076/37.657/65.238/27.581 ms
(base) maharbec@MAHARBEC-M-92HK .ssh %
```

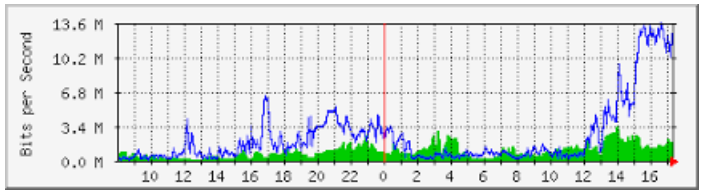
Ping



3com FTP Server

```
(base) maharbec@MAHARBEC-M-92HK .ssh % traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 64 hops max, 52 byte packets
 1  fritz.box (192.168.155.1)  71.241 ms  0.592 ms  0.301 ms
 2  p3e9bf3d7.dip0.t-ipconnect.de (62.155.243.215)  26.740 ms  4.764 ms  4.460 ms
 3  m-ef2-i.m.de.net.dtag.de (217.0.194.110)  31.444 ms  11.650 ms  11.049 ms
 4  m-ef2-i.m.de.net.dtag.de (217.0.194.110)  16.886 ms  10.678 ms  10.597 ms
 5  80.157.205.162 (80.157.205.162)  49.431 ms  11.656 ms  11.223 ms
^C
```

Traceroute



MRTG

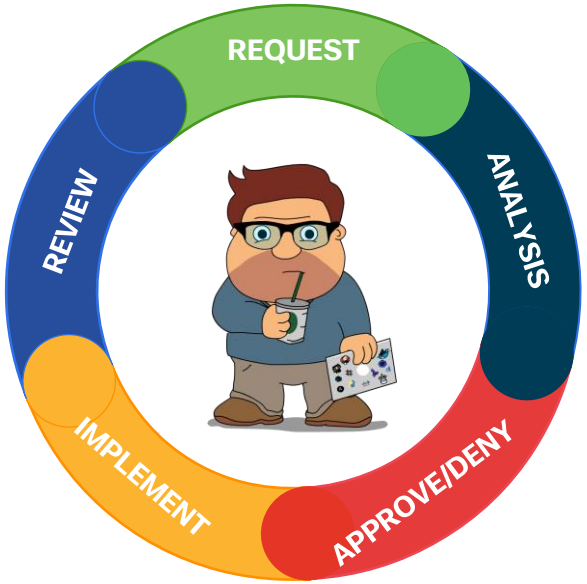
...Simple ?

DevOps tools periodic table

...Simpler ?

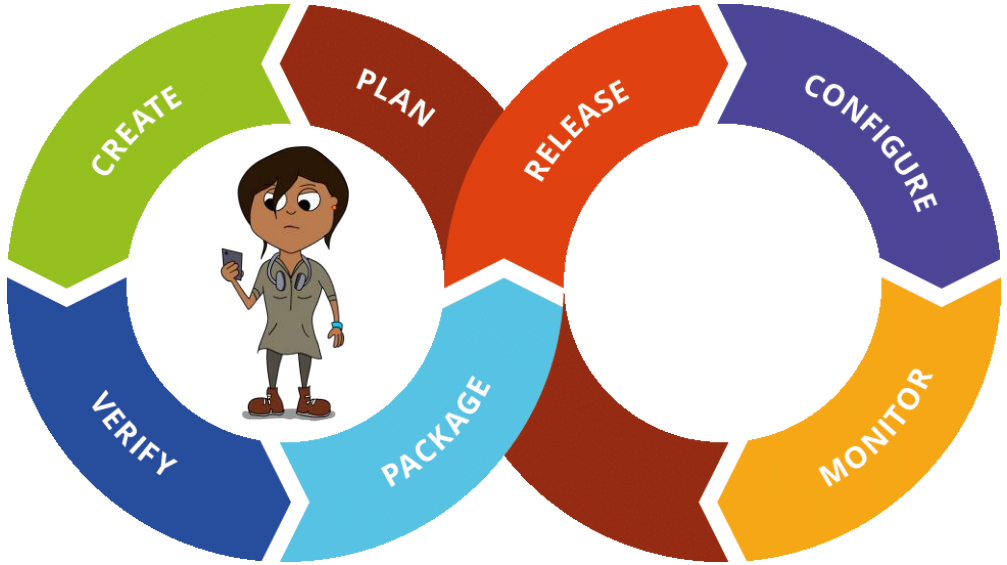
Source: https://xebialabs.com/periodic-table-of-devops-tools/
https://digital.ai/learn/devops-periodic-table/

Different mindsets



Change management mindset

Avoid failure – change is risky and complex – – empowered accountability – limited feedback systems – manual



DevOps Mindset

Embrace failure - change is good - active collaboration- empowered accountability - feedback systems - automation

Are you using Infrastructure as Code already?

Manual vs Automated

Evolution of the network administrator

Traditional Work

- Configure devices manually
- High invest of time
- Error-prone

Scripting

- Initial automation steps
- Often snowflakes done by individuals
- Risky and hard to maintain

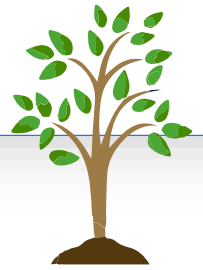
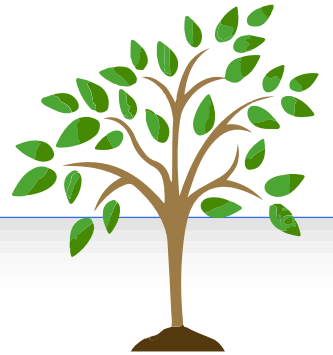
Network Management (NMS) or SDN Controller

- Intend automation
- Still many UI interactions

DevOPS network administration

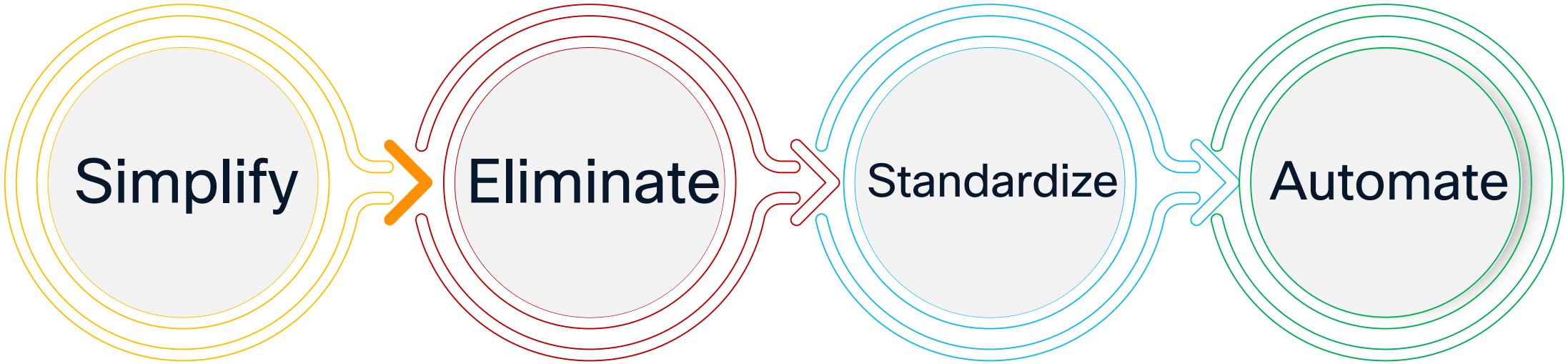
- Automaton
- CICD and versioning
- Rollback and validation



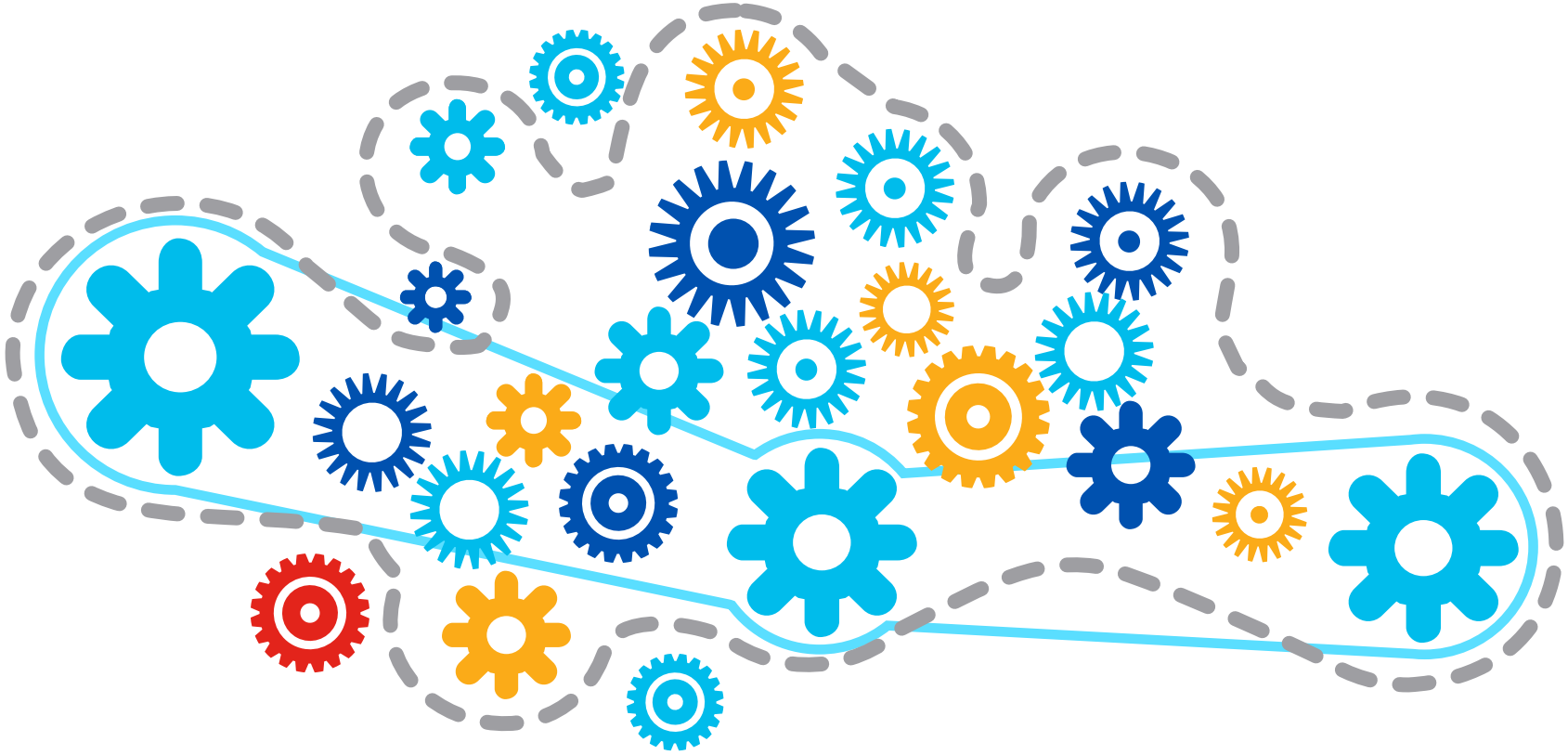



1980-2000 or today ☺ | 2000s - 2010 | 2010- | Just started

Support the journey



Tie Islands of automation together
End-to-end Automation delivers business value



Multiple operating systems and platforms (>10)

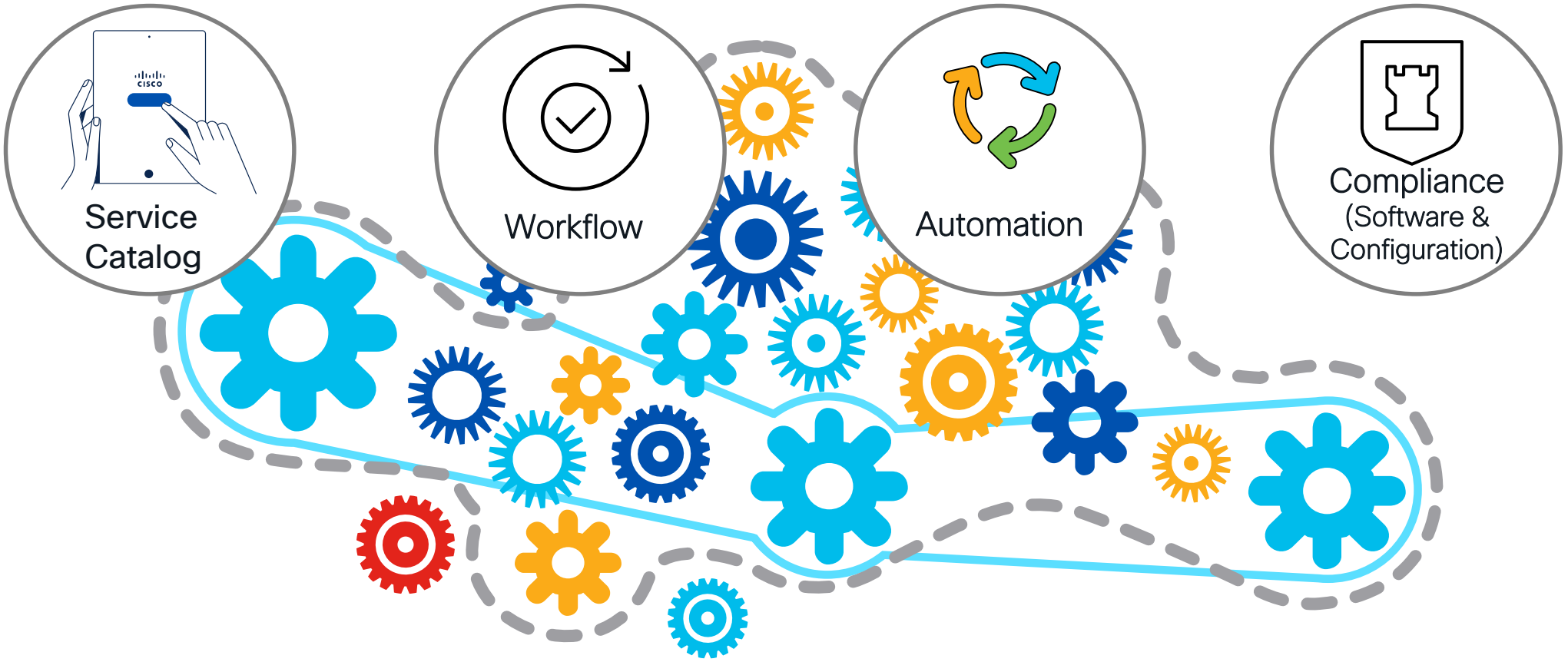
Different types of automation tools (>50)

APIs & Scripts (>10,000)

Manual activity (>1,000)

Orchestration software, cross-team collaboration

Reduce and use complexity



Start with "blue-prints"

Reusable Use cases

Business relevance

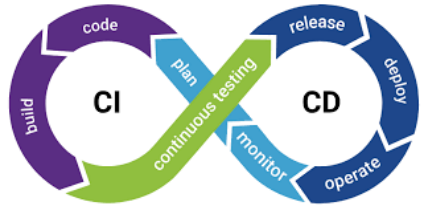
Best praxis's, defaults and reusability

New kind of CLI

Automate only what makes sense

Some DevOPS

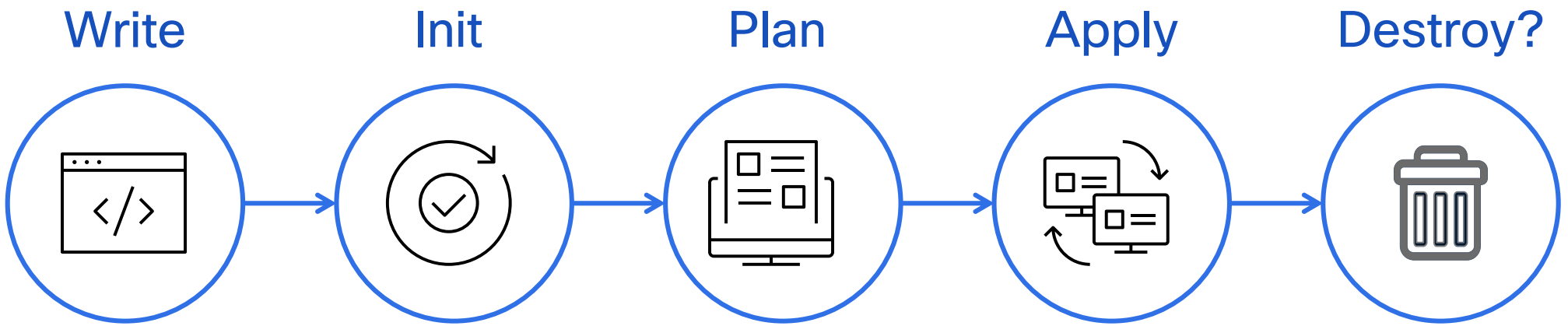
Three Pillars of DevOPS



GITlab

HashiCorp Terraform

Terraform Workflow Stages



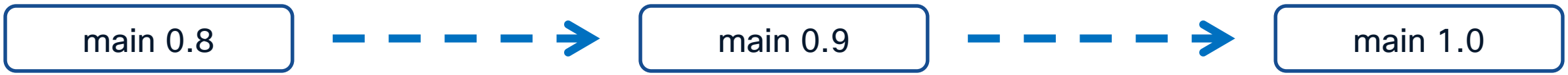
Write	Init	Plan	Apply	Destroy?
Write your code	Download and initialize environment (providers, modules, backend, ...)	Generate "terraform-statefile" by calculate required changes	Apply changes into real infrastructure	Remove all resources

GIT Branch Flow

GITlab

- GIT is not an acronym
- GIT is a "Version Control System"
- GIT can be used for source code, single files, or many YAML files

New features, parallel development, Test ”branches”

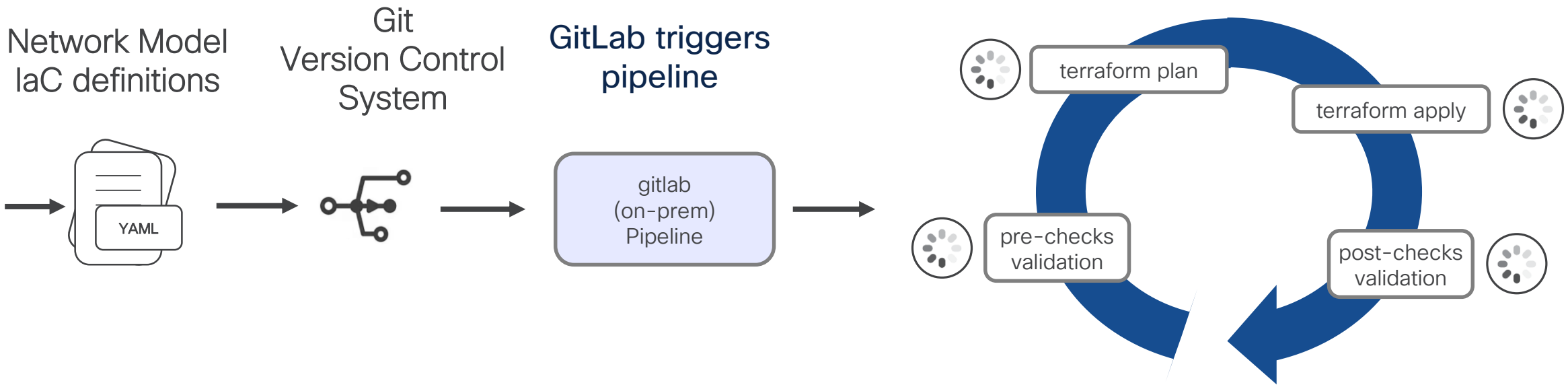


0.7 plus feature
"crystal ball"

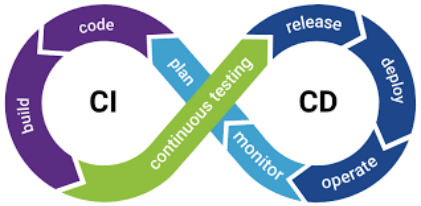
0.8 plus feature
"blue ball"

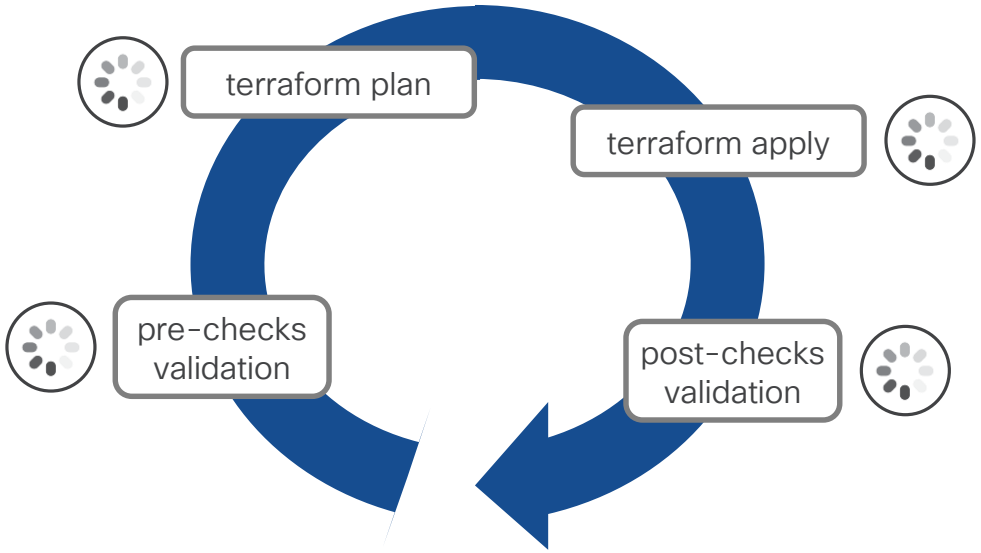
0.9 plus feature
"green ball"

Pipeline principals



Pipeline to CICD





Service as Code
The specialty

CX Data Model – example tenant config

Product Configuration Model API - native


```
{
    "totalCount": "1",
    "imdata": [
        {
            "fvTenant": {
                "attributes": {
                    "annotation": "",
                    "descr": "",
                    "dn": "uni/tn-CX",
                    "name": "CX",
                    "nameAlias": "",
                    "ownerKey": "",
                    "ownerTag": "",
                    "userdom": ":all:"
                },
                "children": [
                    {
                        "vnsSvcCont": {
                            "attributes": {
                                "annotation": "",
                                "userdom": ":all:"
                            }
                        }
                    },
                    {
                        "fvRsTenantMonPol": {
                            "attributes": {
                                "annotation": "",
                                "tnMonEPGPolName": "",
                                "userdom": ":all:"
                            }
                        }
                    },
                    {
                        "fvEpTags": {
                            "attributes": {
                                "annotation": "",
                                "userdom": ":all:"
                            }
                        }
                    },
                    {
                        "fvCtx": {
                            "attributes": {
                                "annotation": "",
                                "bdEnforcedEnable": "no",
                                "descr": "",
                                "ipDataPlaneLearning": "enabled",
                                "knwMcastAct": "permit",
                                "name": "VRF2",
                                "nameAlias": "",
...
```


+200 Lines of Configuration

TF-Provider

Infrastructure as Code Model TF provider - native

```
resource "aci_tenant" "tenant_CX" {
  name = "CAB"
}

variable "vrfs" {
  default = {
    VRF1 = {
      name = "VRF1"
    },
    VRF2 = {
      name = "VRF2"
    }
  }
}

resource "aci_vrf" "vrfs" {
  for_each  = var.vrfs
  tenant_dn = aci_tenant.tenant_CX.id
  name      = each.value.name
}
```

20 Lines of Configuration

TF-Module

CX Data Model

```
apic:
  tenants:
    - name: CX
      vrfs:
        - name: VRF1
        - name: VRF2
```

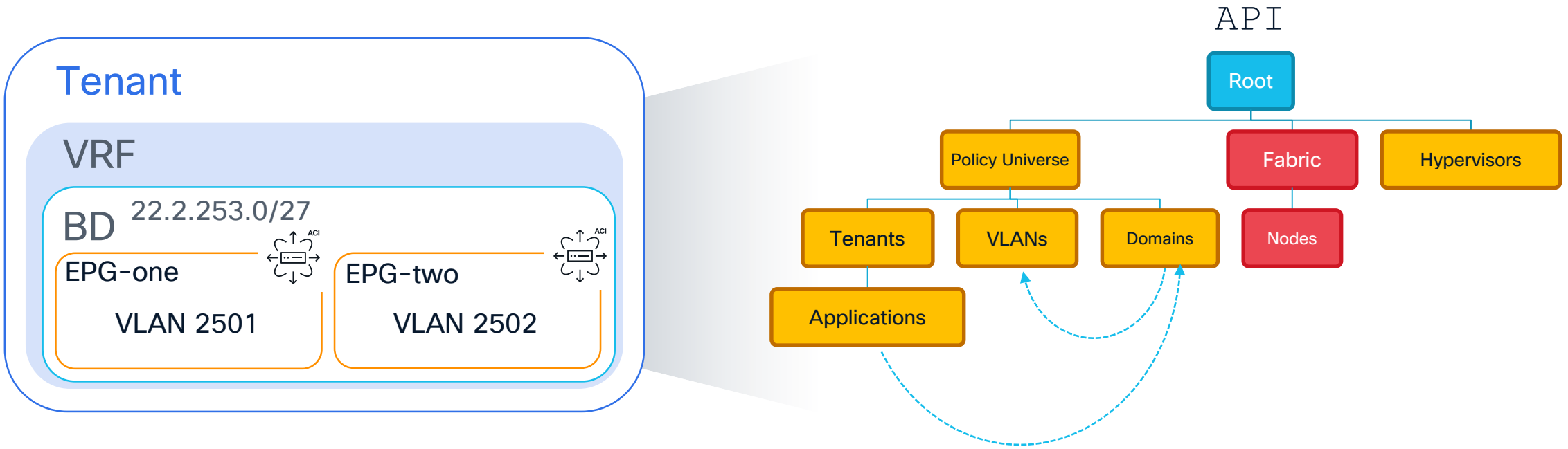
6 Lines of Data

- Drives all pit stops as single “source of truth”
- Same look & feel across architectures
- Highly simplified and abstracted
- Best Practices baked in
- Possibility to expand to 3rd parties
- Usable via NAC-API in ITSM “no Code”
- Toolset independent “Open Source”
- passive part of Digital Twin

The key element is data and abstraction

Example ACI

- Every Controller or every device configuration follows a data structure
- Data needs to be human and machine readable



Include important settings by default's

```
---
defaults:
  apic:
    fabric_policies:
      ep_loop_protection:
        admin_state: disabled
        detection_intervall: 60
        detection_multiplier: 4
        action: bd-learn-disable
    tenants:
      vrfs:
        name_suffix: '_VRF'
        data_plane_learning: enabled
        enforcement_direction: ingress
        enforcement_preference: enforced
```

- Lessons learned
- CVD and best practice
- Customer requirements
- Eliminate
- Standardize
- Simplify cases

”Increase Efficiency with Service as Code for scalable, repeatable infrastructure automation.”

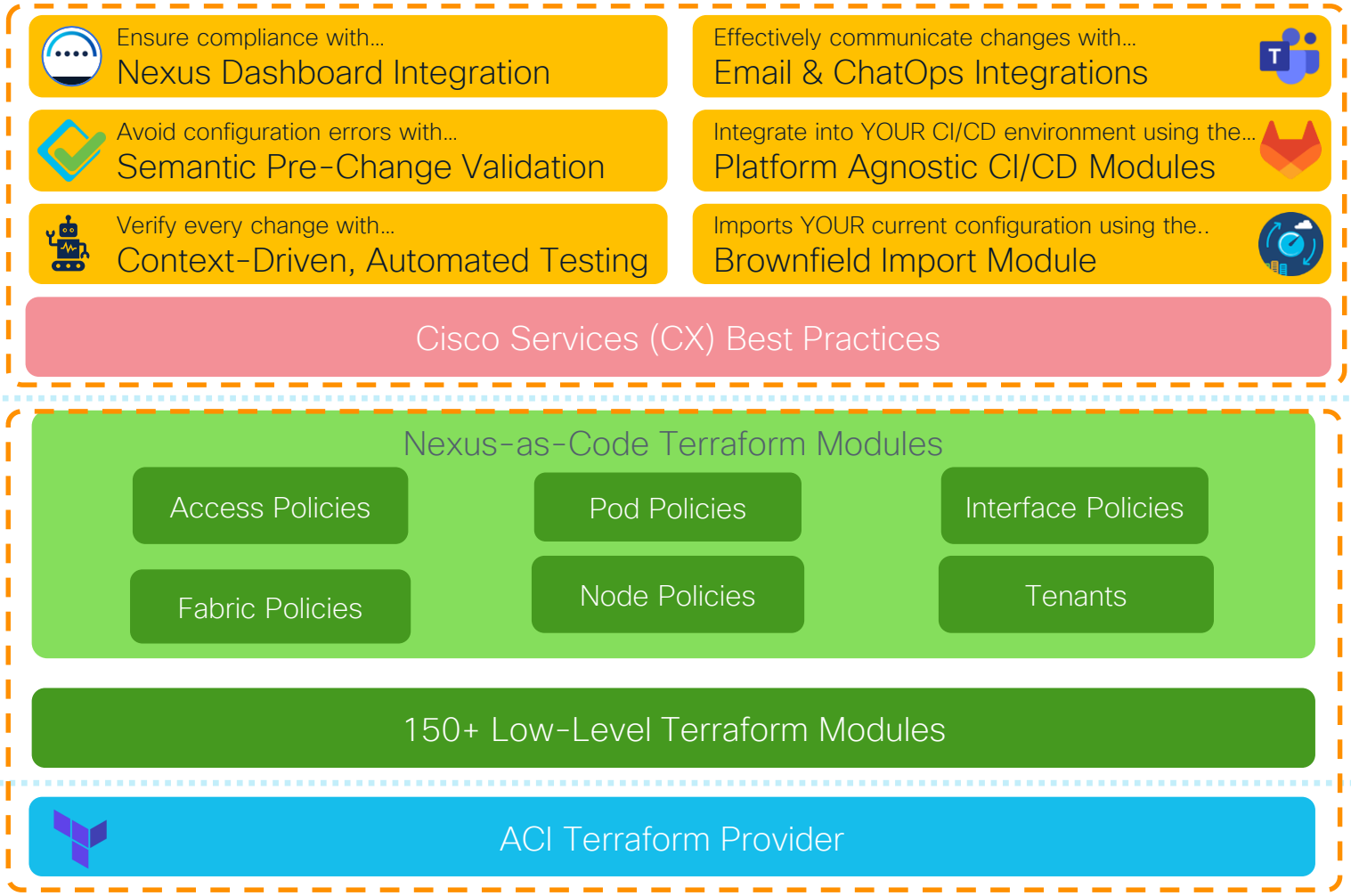
Planning, versioning and automation is key!

GITlab

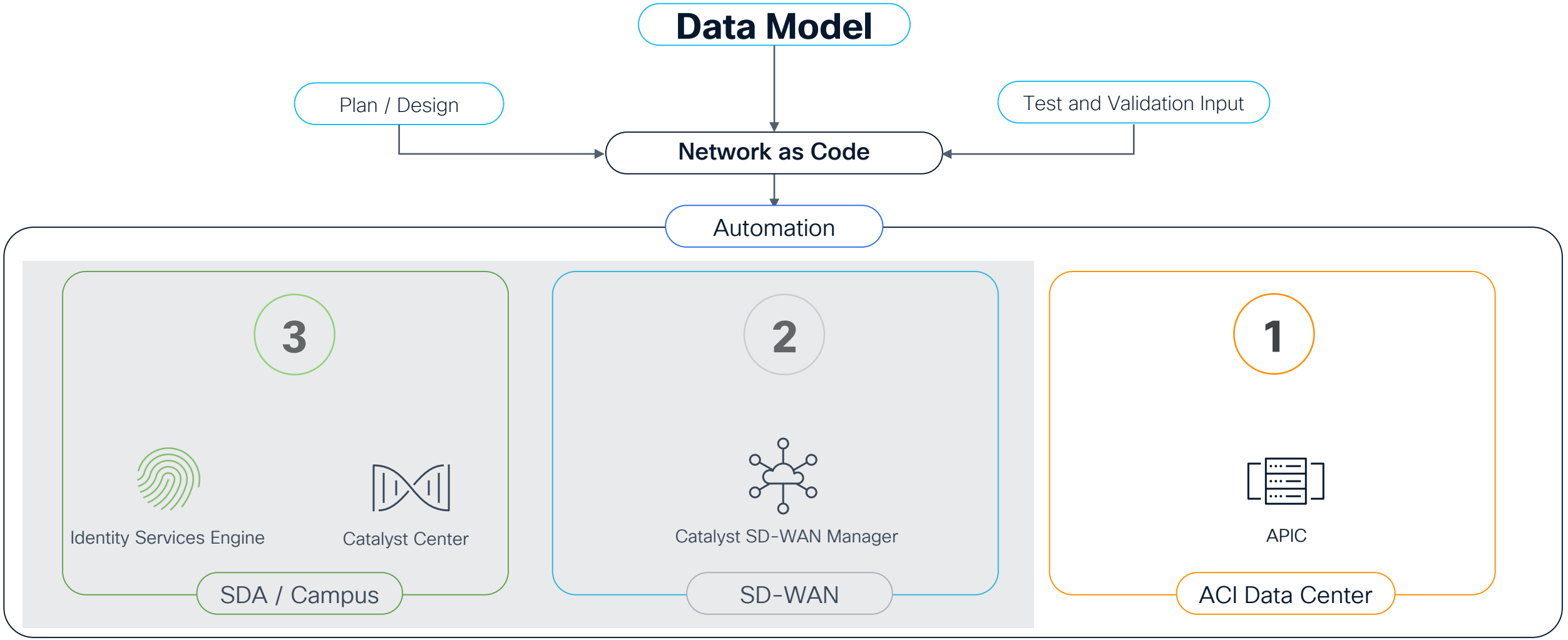
Automation “apply” via Pipeline

Network as Code for ACI

NW as Code for ACI Model as Blueprint for our IaC



Back to the concept



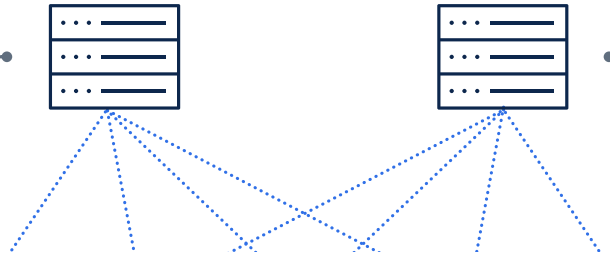
Application Centric Infrastructure building blocks

Built on Cisco Nexus 9000

Cisco APIC (3 or more)

Centralized policy model, network automation

Single open API for entire system – (Terraform, Ansible, Python, Etc)



Flexible: Modular and fixed spine options

Non-blocking 40/100/400G fabric, CLOS fabric

Integrated overlay | Distributed gateway

Built-in distributed stateless firewall, multi-tenant security

Physical, virtual and container workloads (VMW, HyperV, Hadoop, AIX, K8S etc)

WAN interconnect

IP storage (NVMEoF, iSCSI, NFS)

Network service appliances (FW, LB, IPS)

 Price

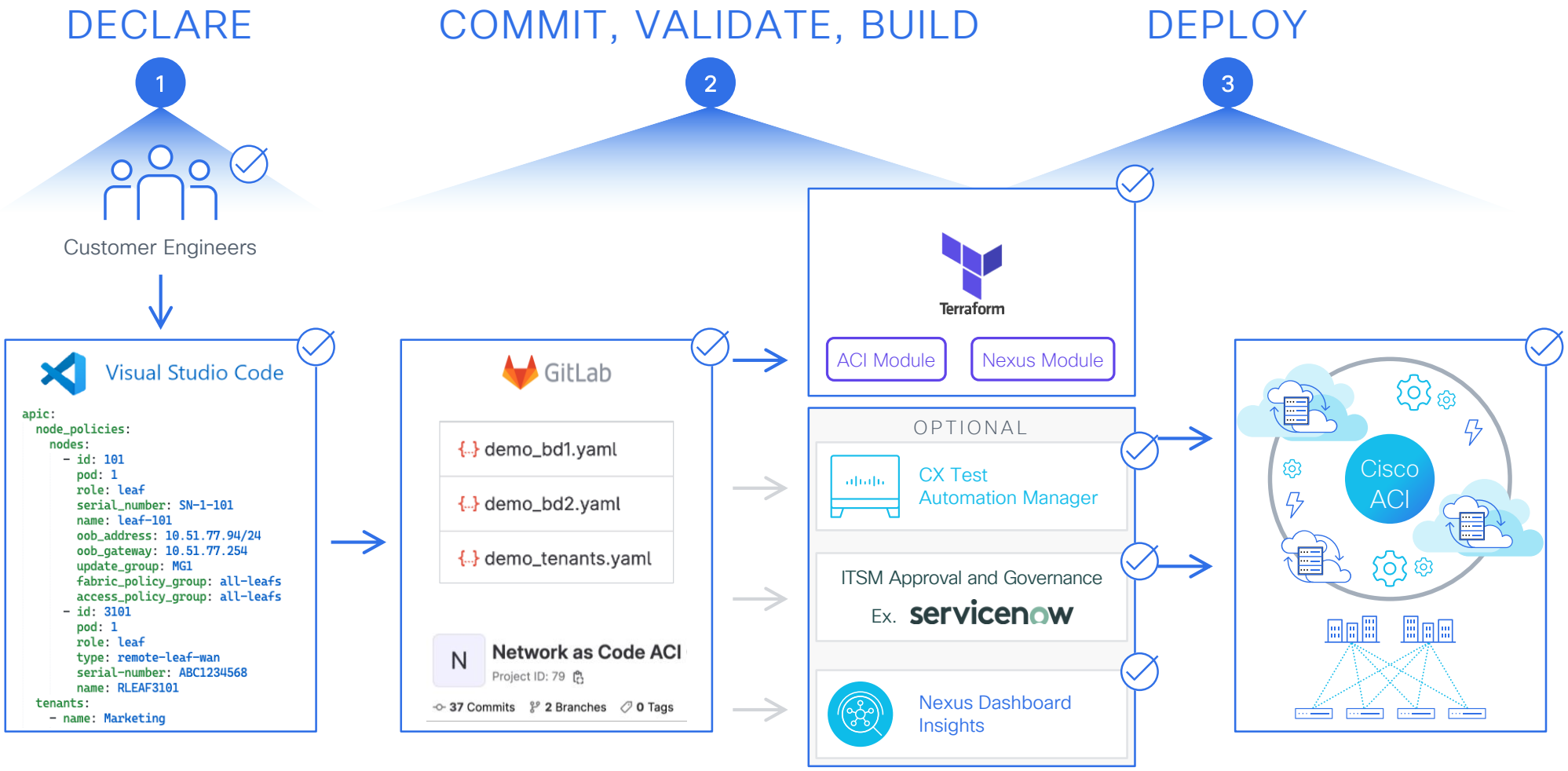
 Performance

 Port density

 Programmability

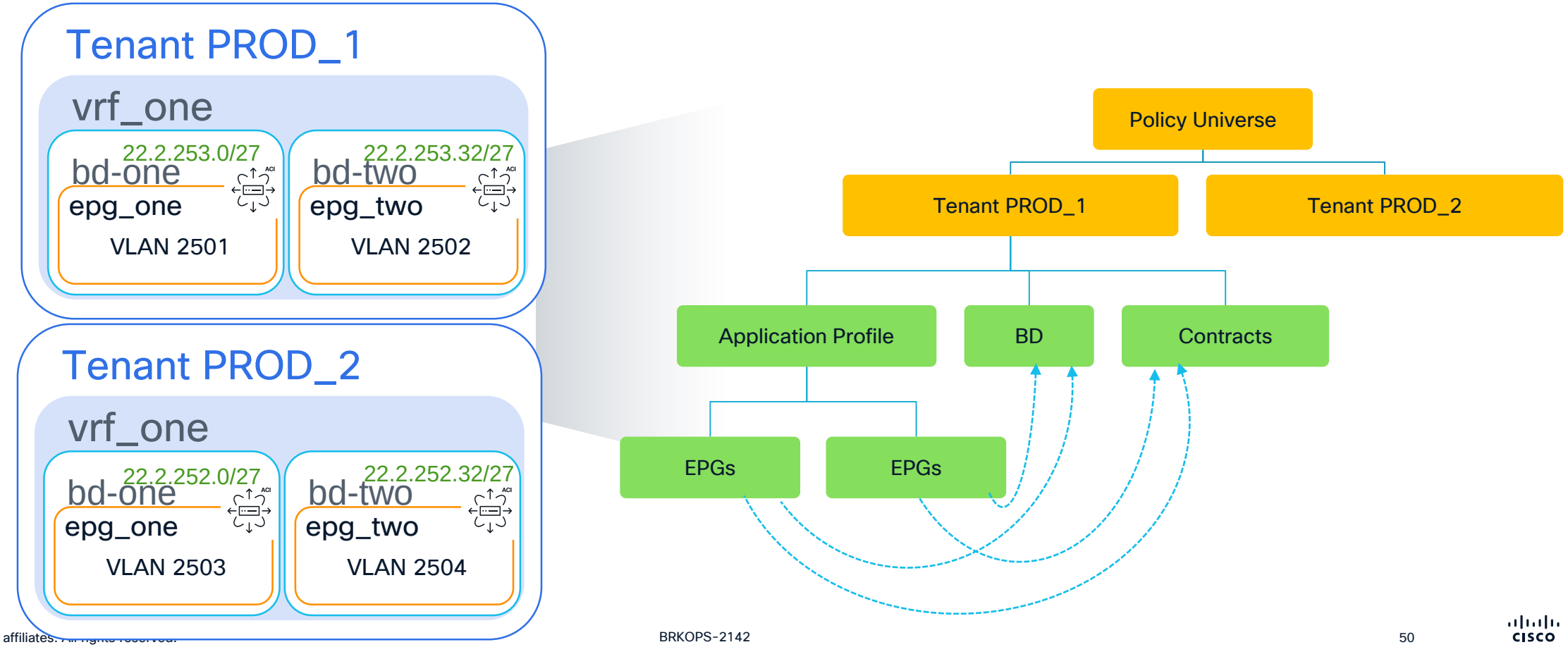
 Power efficiency

An Example: Network as Code for ACI flow

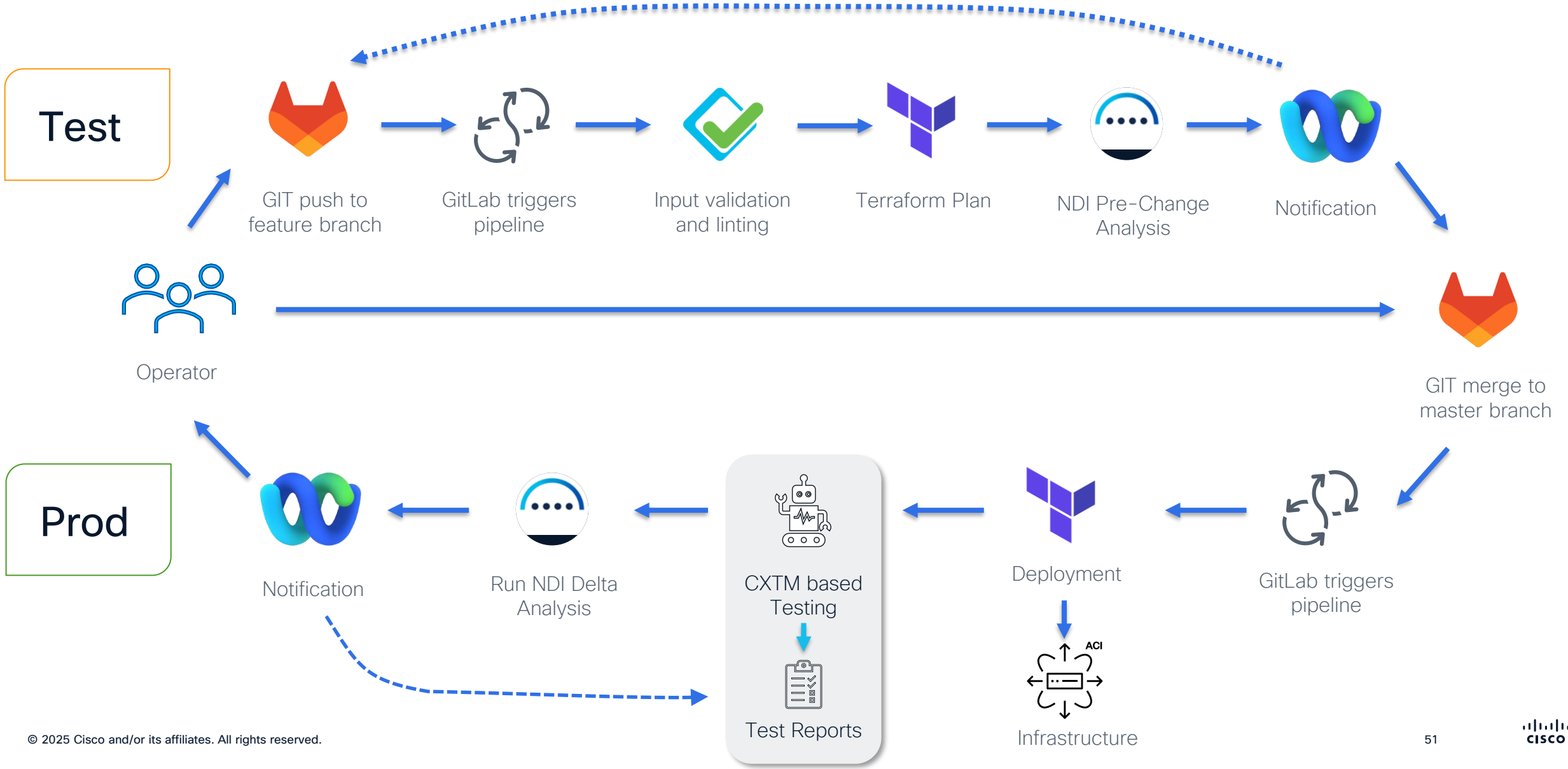


The key element is Data and abstraction

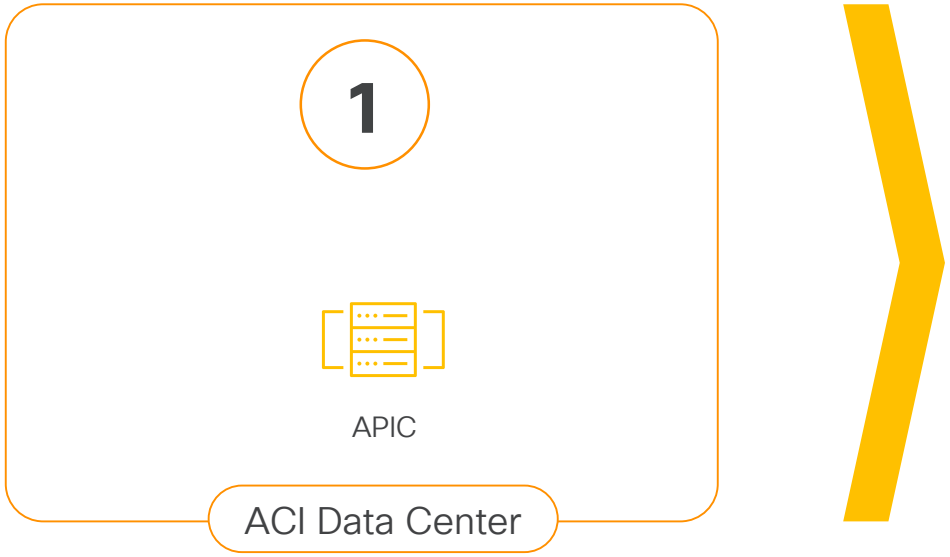
- Objects having dependencies to each other, as well sequence to create is important
- ACI-as-Code will take care



Example for Network as Code workflow



DEMO Time



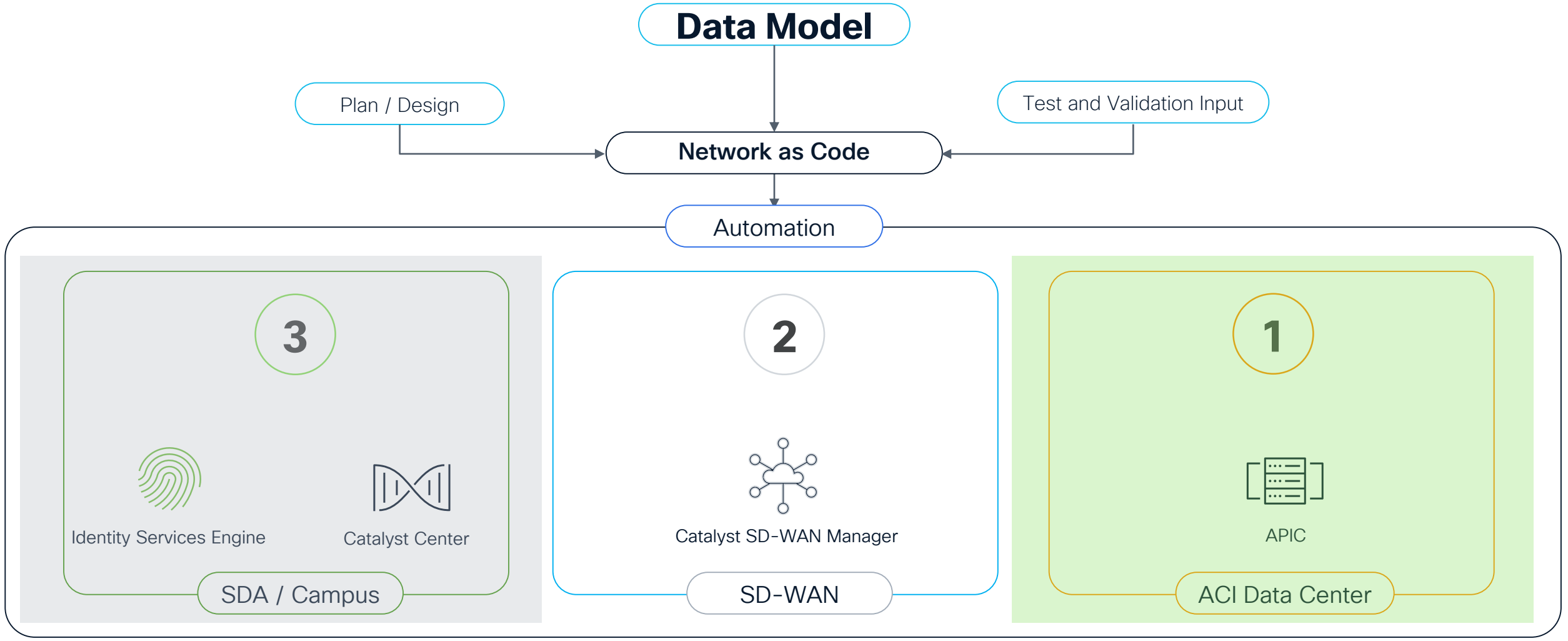
ACI.yaml

```yaml
---
apic:
  tenants:
    - name: PROD_1
      description: Cisco Live blue-print-1
      vrfs:
        - name: prod
          enforcement_direction: egress

      bridge_domains:

        - name: bd-one
          vrf: prod
          subnets:
            - ip: 22.2.253.1/27
      application_profiles:
        - name: production
          endpoint_groups:

            - name: epg-one
              bridge_domain: bd-one
              physical_domains:
                - DOM_PHY_SERVER
              static_ports:
                - node_id: 1011
                  pod_id: 1
                  port: 13
                  vlan: 2501
      l3outs:
# Transit to Services as Code Demo - SDA Munchen Hofbrauhaus
        - name: 'l3o-sdwan-1'
          vrf: prod
          domain: DOM_L3O_CORE
          description: Towards SDWAN BGP AS 64927

          node_profiles:   # Node profile and corrspondong link profile for IPv4
            - name: 'l3o_ipv4_NodeProfile'
              nodes:
                - node_id: 1001
                  pod_id: 1
                  router_id: 22.2.255.1
                - node_id: 1002
                  pod_id: 1
                  router_id: 22.2.255.2
```

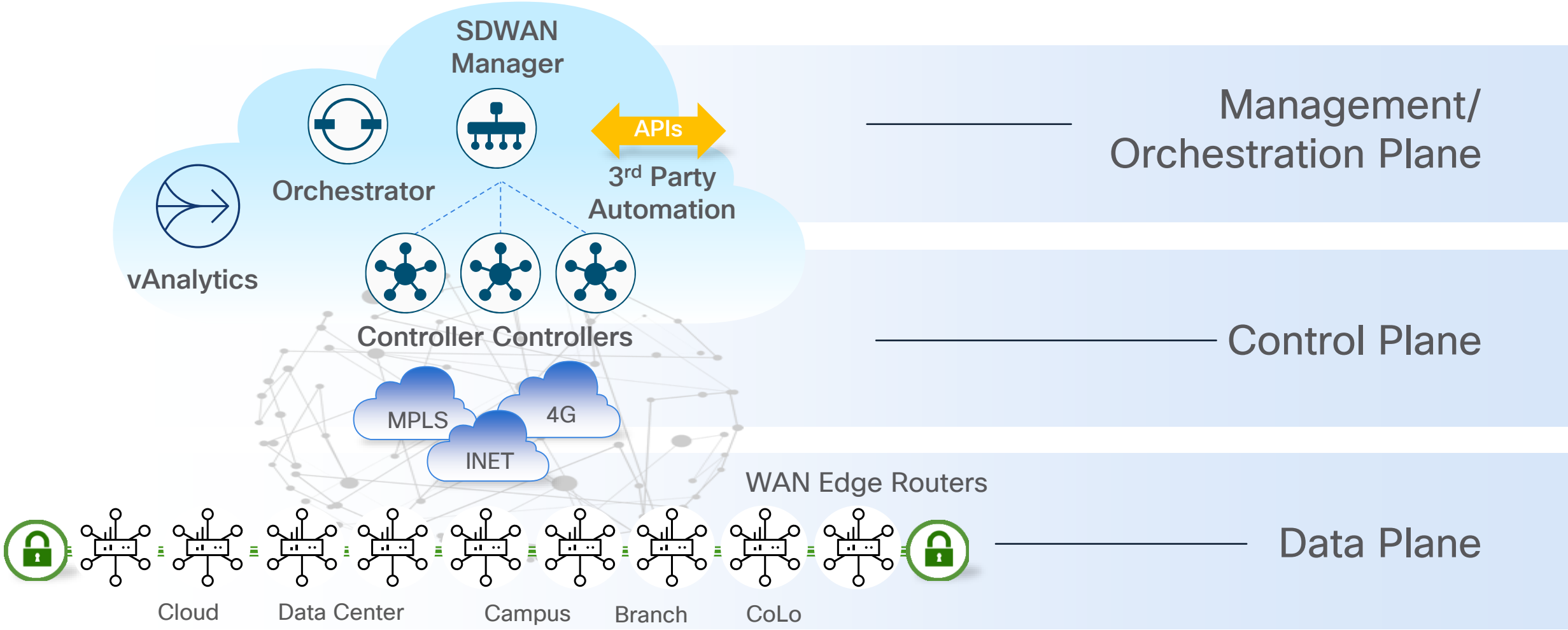
First run of Services as Code - whats your impression?

Network as Code for SDWAN

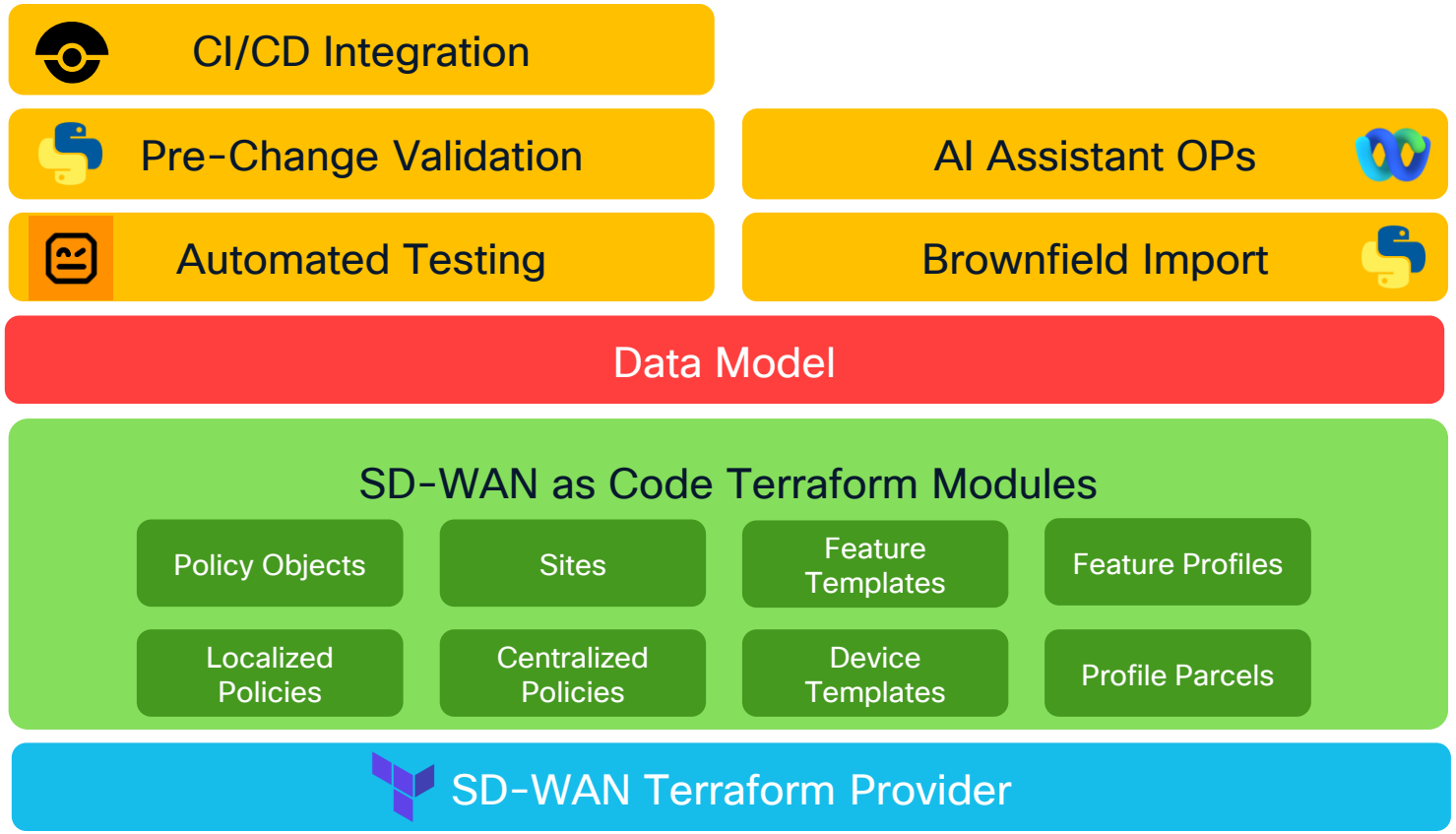
Network as Code for SDWAN



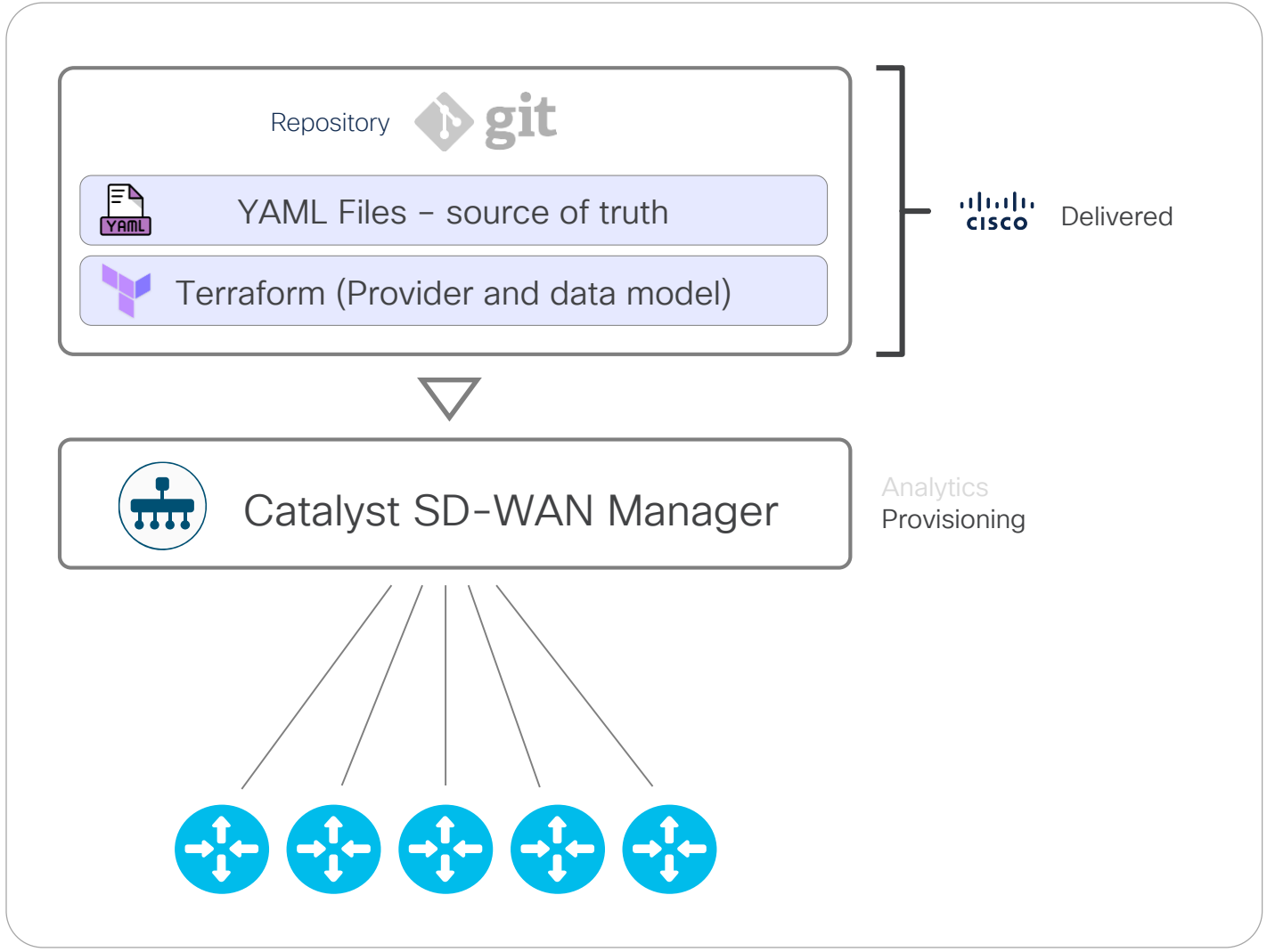
Cisco SD-WAN Solution Overview



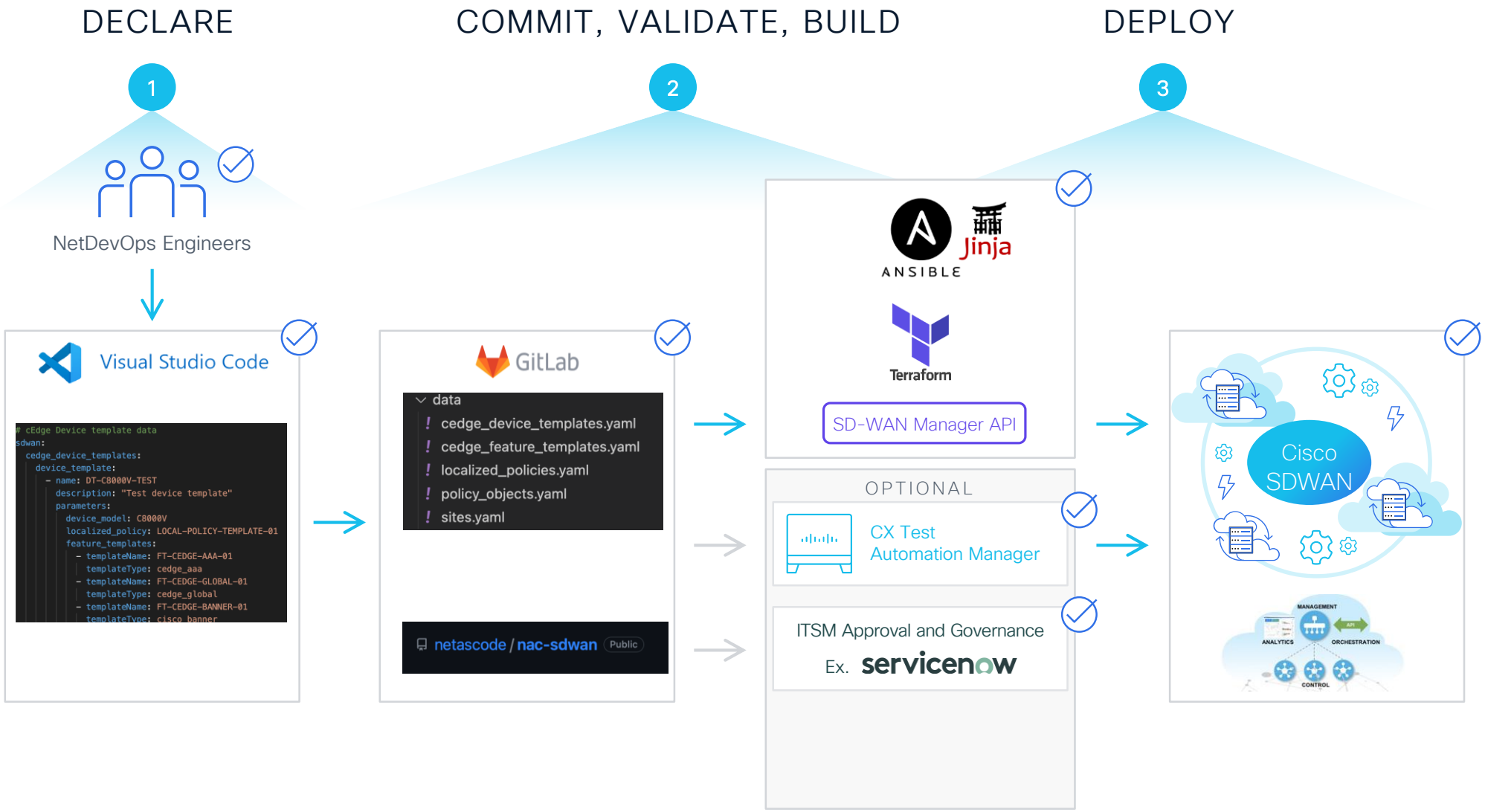
Network as Code for SDWAN overview



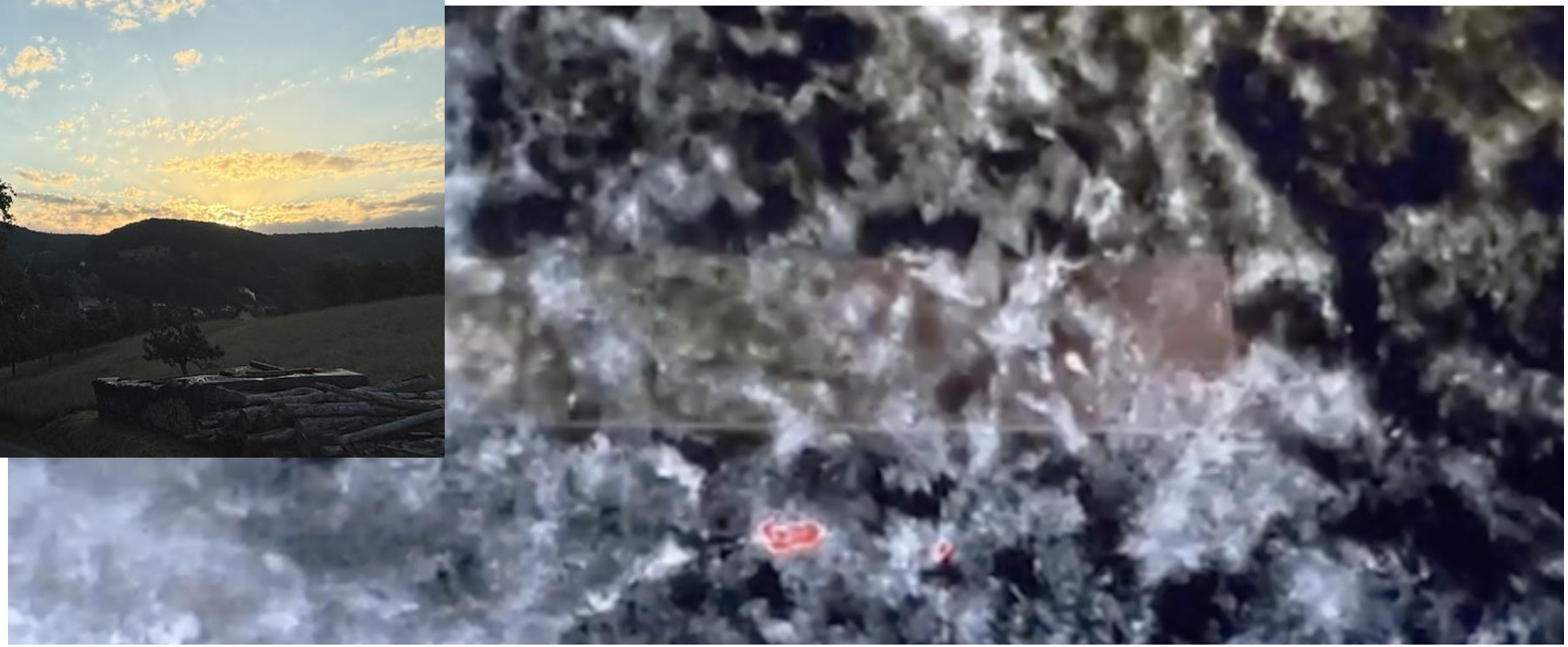
SDWAN as Code



Network as Code for SDWAN flow

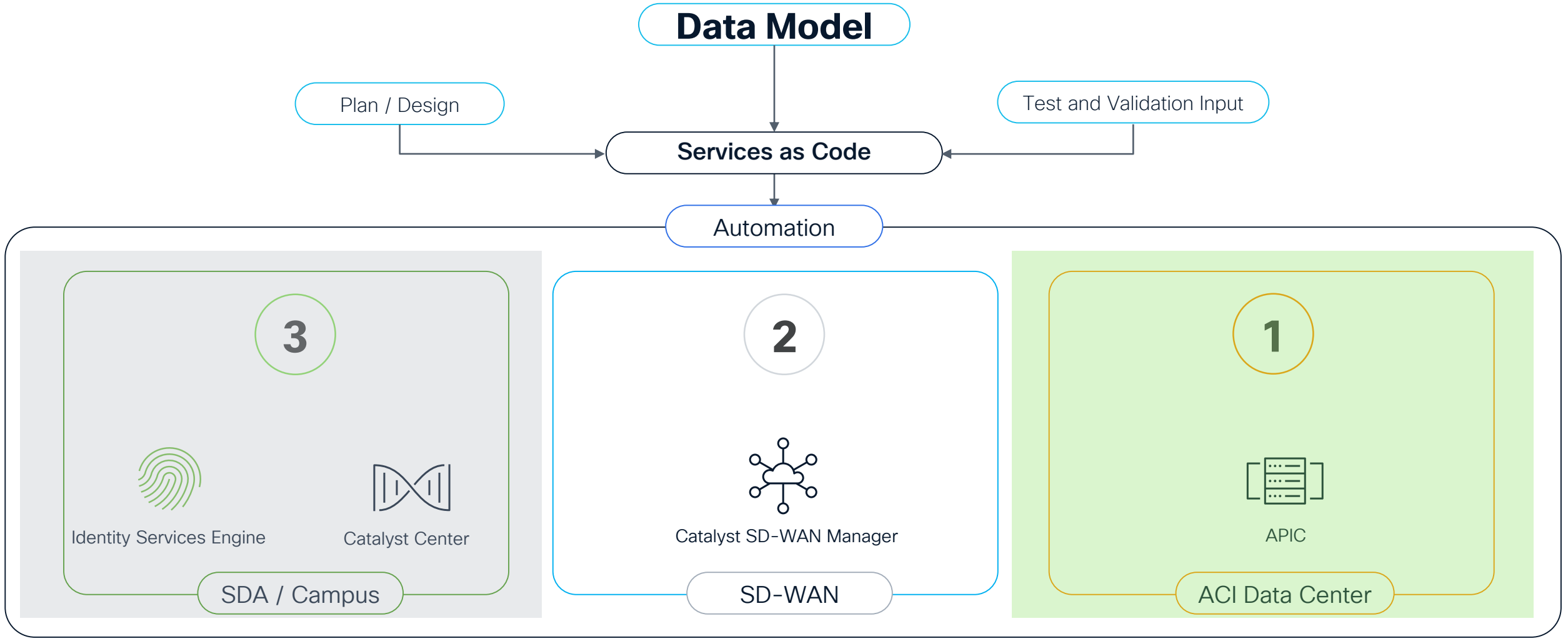


”Increase Efficiency with Service as Code for scalable, repeatable infrastructure automation."



...automated flight and crisp views

Demo Time → Network as Code for SDWAN



DEMO Time

2

Catalyst SD-WAN Manager

SD-WAN

SDWAN.yaml 1 ●

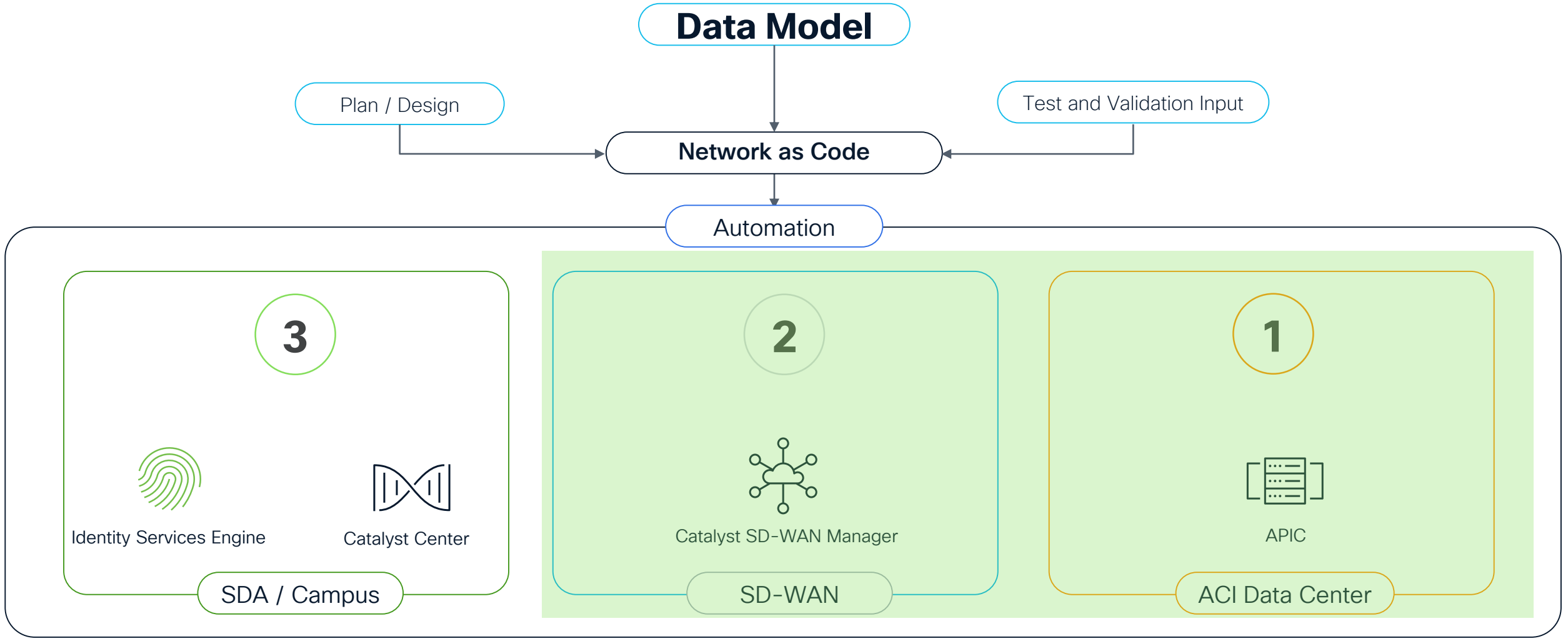
ers > maharbec > Documents > 06 - LAB > _SaC > 21 - AAC090 > SDWAN.yaml > {} sdwan > {} edge_feature_ter

```yaml
---
# SD-WAN Sites
sdwan:
  sites:
    - id: 221
      routers:
        - chassis_id: C8300-1N1S-4T2X-FDO2724M1K8
          model: C8300-1N1S-4T2X
          device_template: DT-C8300-1N1S-4T2X-SINGLECPE-Branch1
          device_variables:
            site_id: 221
            system_ip: 1.22.4.21
            system_hostname: CI83k-BER-HQ1
sdwan:
  edge_feature_templates:
    aaa_templates:
      - name: FT-CEDGE-AAA-01
        description: Local auth
        authentication_and_authorization_order:
          - local
    bgp_templates:
      - name: FT-CEDGE-BGP-VPN1-ACTIVE
        description: VPN 1 BGP
        ipv4_address_family:
          default_information_originate: true
          maximum_paths_variable: vpn1_bgp_ipv4_maximum_paths
          redistributes:
            - protocol: omp
              optional: false
              route_policy: RM-SITE-BGP-OUT-ACTIVE
          neighbors:
            - address_variable: vpn1_bgp_ipv4_neighbor1_address
              address_families:
                - family_type: ipv4-unicast
                  maximum_prefixes: 1000
              next_hop_self: false
    ntp_templates:
      - name: FT-CEDGE-NTP-01
        description: Base NTP template; no auth key; no ntp master
        servers:
          - hostname_ip: 10.49.216.10
            prefer: true
            source_interface_variable: ntp_server_source_interface
            vpn_id: 0
```


cisco

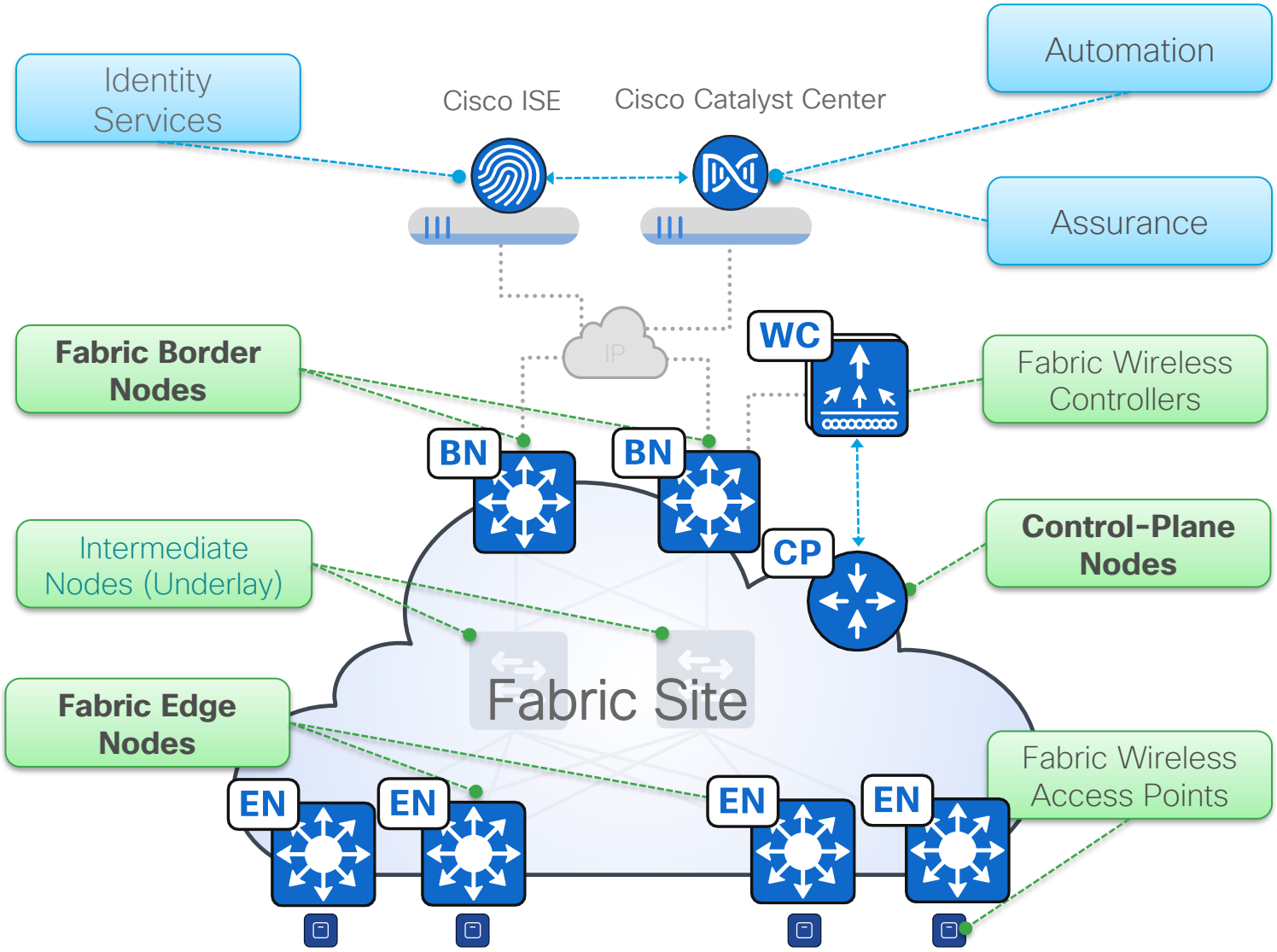
Network as Code for Catalyst Center (SDA) with ISE

Network as Code for Catalyst Center (SDA) with ISE



Cisco SD-Access

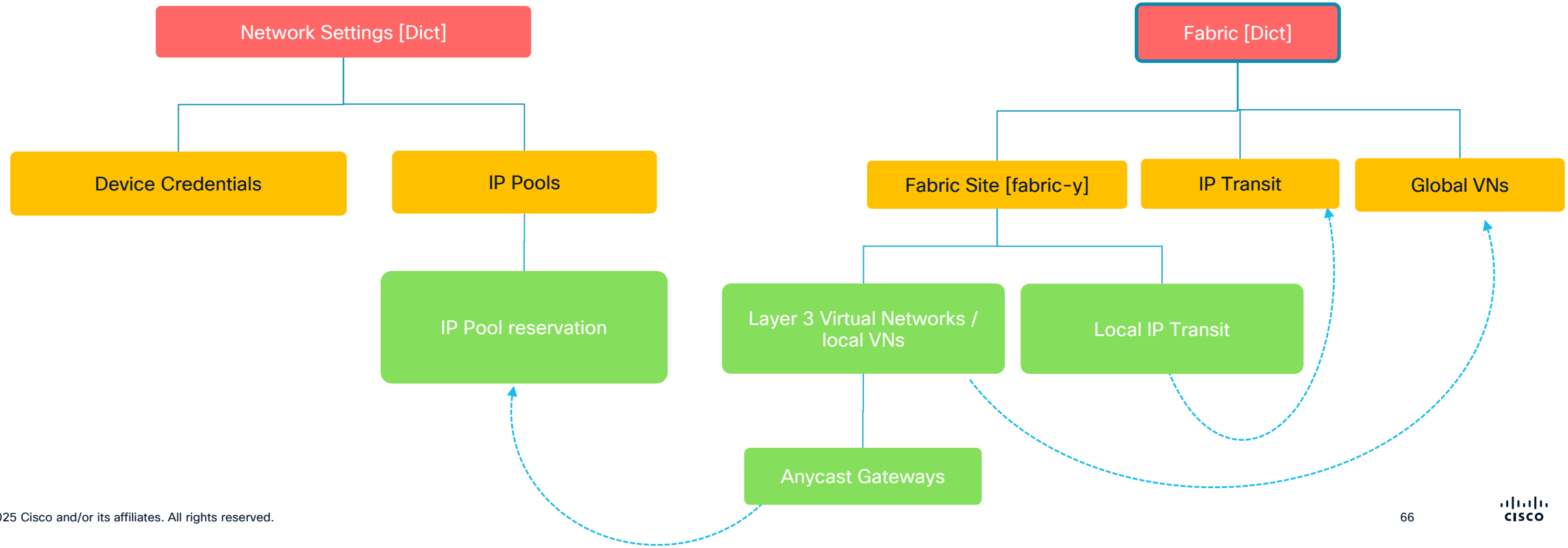
The ACI Policy Model – Add External Connection



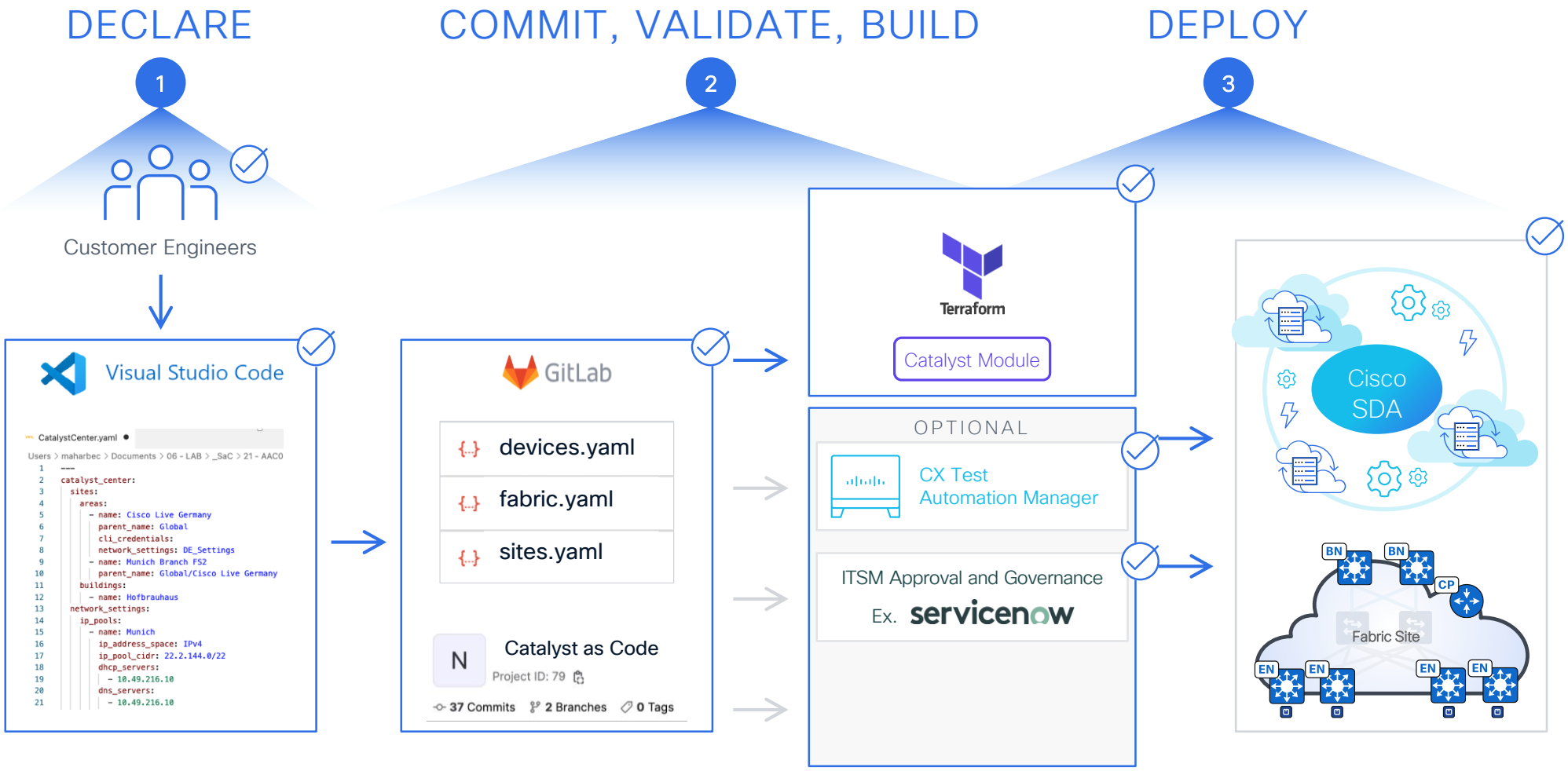
- **Network Automation** – Simple GUI and APIs for intent-based Automation of wired and wireless fabric devices
- **Network Assurance** – Data Collectors analyze Endpoint to Application flows and monitor fabric device status
- **Identity Services** – NAC & ID Services (e.g. ISE) for dynamic Endpoint to Group mapping and Policy definition
- **Control-Plane Nodes** – Map System that manages Endpoint to Device relationships
- **Fabric Border Nodes** – A fabric device (e.g. Core) that connects External L3 network(s) to the SD-Access fabric
- **Fabric Edge Nodes** – A fabric device (e.g. Access or Distribution) that connects Wired Endpoints to the SD-Access fabric
- **Fabric Wireless Controller** – A fabric device (WLC) that connects Fabric APs and Wireless Endpoints to the SD-Access fabric

Data abstraction for SDA fabric

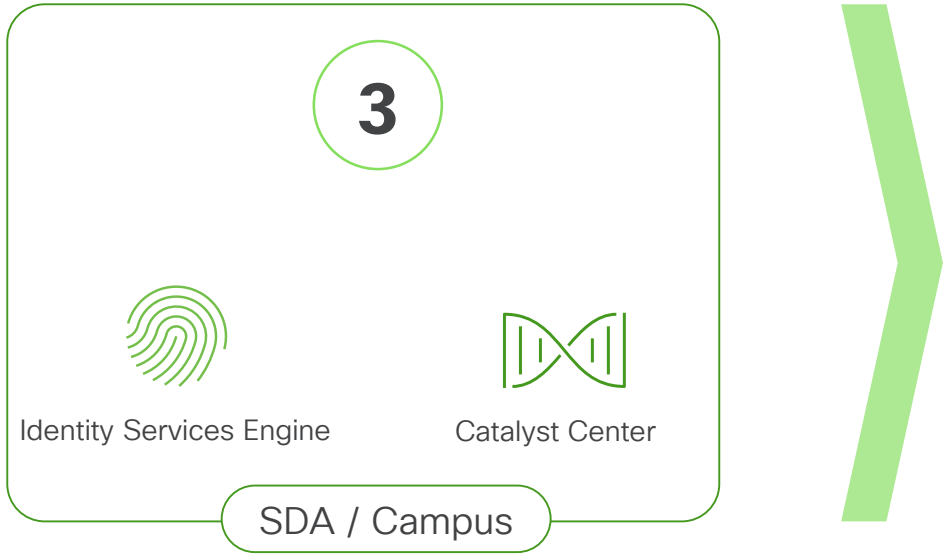
- Every Controller or every device configuration follows a data structure
- Objects having dependencies to each other, as well sequence to create is important
  - Catalyst Center as Code will take care



Network as Code for Catalyst Center flow



DEMO Time

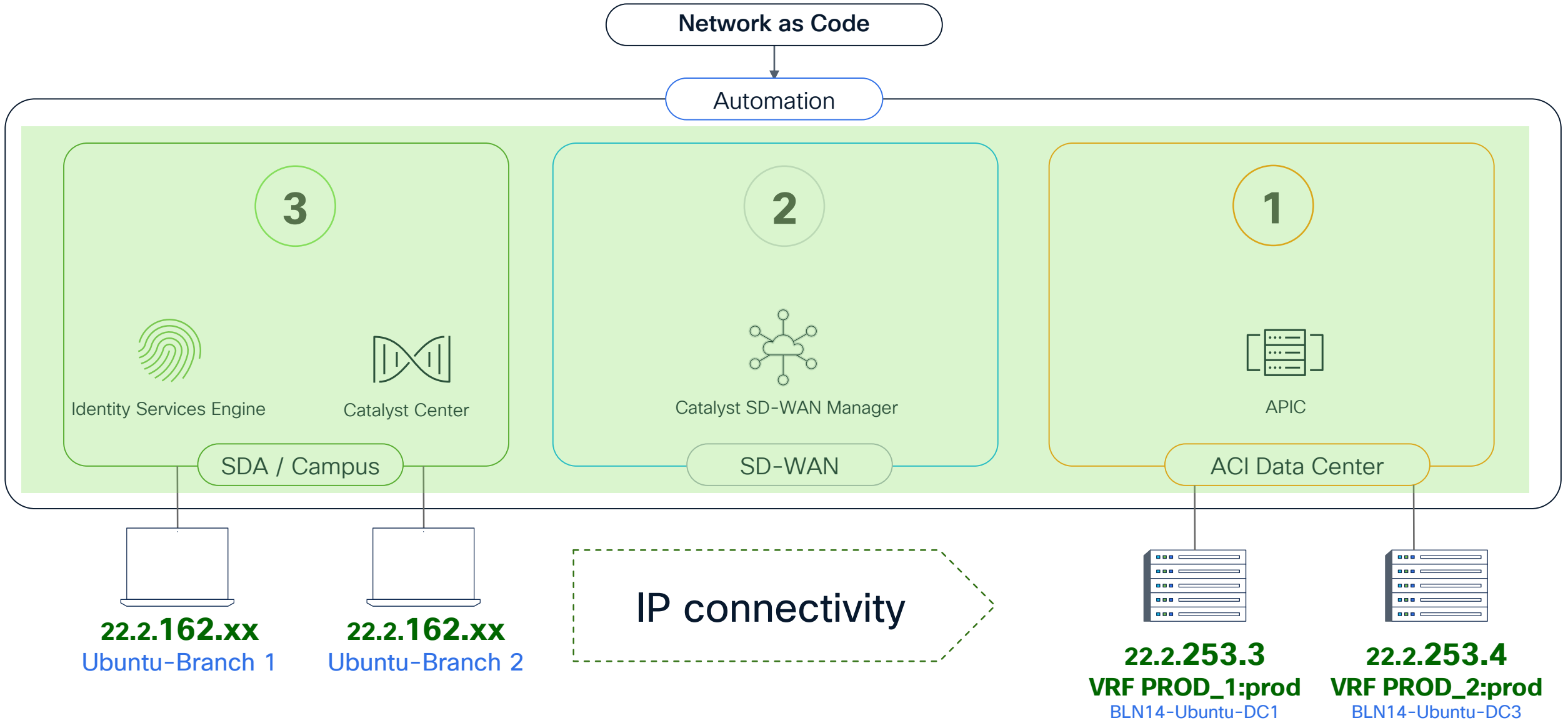


CatalystCenter.yaml

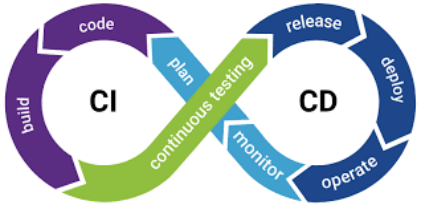
Users > maharbec > Documents > 06 - LAB > _SaC > 21 - AAC090 > CatalystCenter.yaml > { } catalyst_center > { }

```yaml
---
catalyst_center:
  sites:
    areas:
      - name: Cisco Live Germany
        parent_name: Global
        cli_credentials:
        network_settings: DE_Settings
      - name: Munich Branch FS2
        parent_name: Global/Cisco Live Germany
    buildings:
      - name: Hofbrauhaus
  network_settings:
    ip_pools:
      - name: Munich
        ip_address_space: IPv4
        ip_pool_cidr: 22.2.144.0/22
        dhcp_servers:
          - 10.49.216.10
        dns_servers:
          - 10.49.216.10
        ip_pools_reservations:
          - name: MUC_CORP
            prefix_length: 27
            subnet: 22.2.144.0
            type: Generic
  fabric:
    transits:
      - name: TRANSIT-MUC-SDWAN1
        type: ip_transit
        routing_protocol_name: BGP
        autonomous_system_number: 64927
    fabric_sites:
      - name: Global/Cisco Live Germany/Munich Branch FS2/Hofbrauhaus
        authentication_template_name: Closed Authentication
        fabric_type: FABRIC_SITE
        l3_virtual_networks:
          - name: MUC_SDA_VN_CORP
        anycast_gateways:
          - name: MUC_CORP
            vlan_name: VLAN_CORP
            vlan_id: 2021
            traffic_type: DATA
            wireless_pool: false
```

Demo Time → Outcome

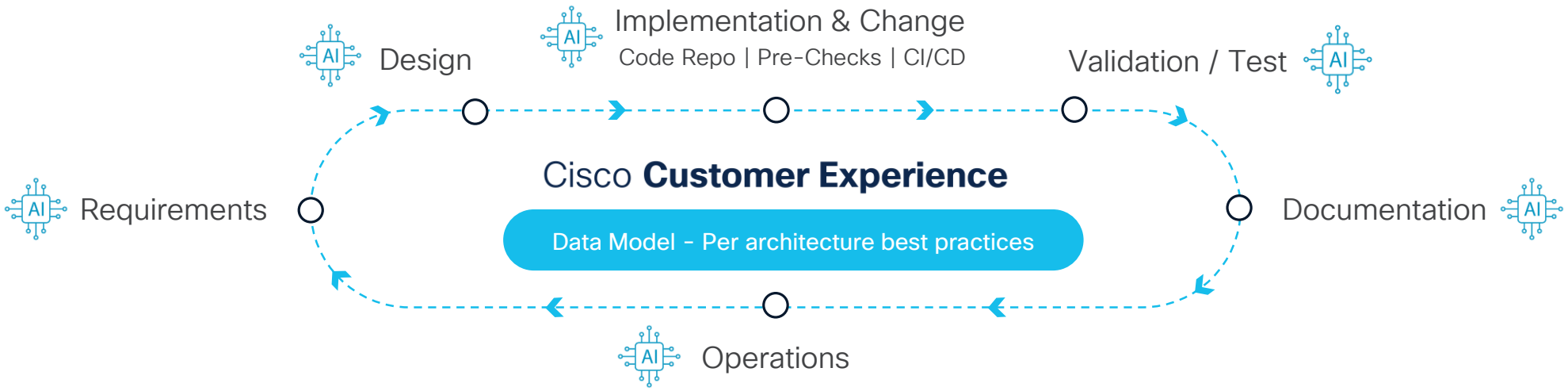


“Increase Efficiency with Service as Code for scalable, repeatable infrastructure automation.”



Next Steps in Automation

Network as Code



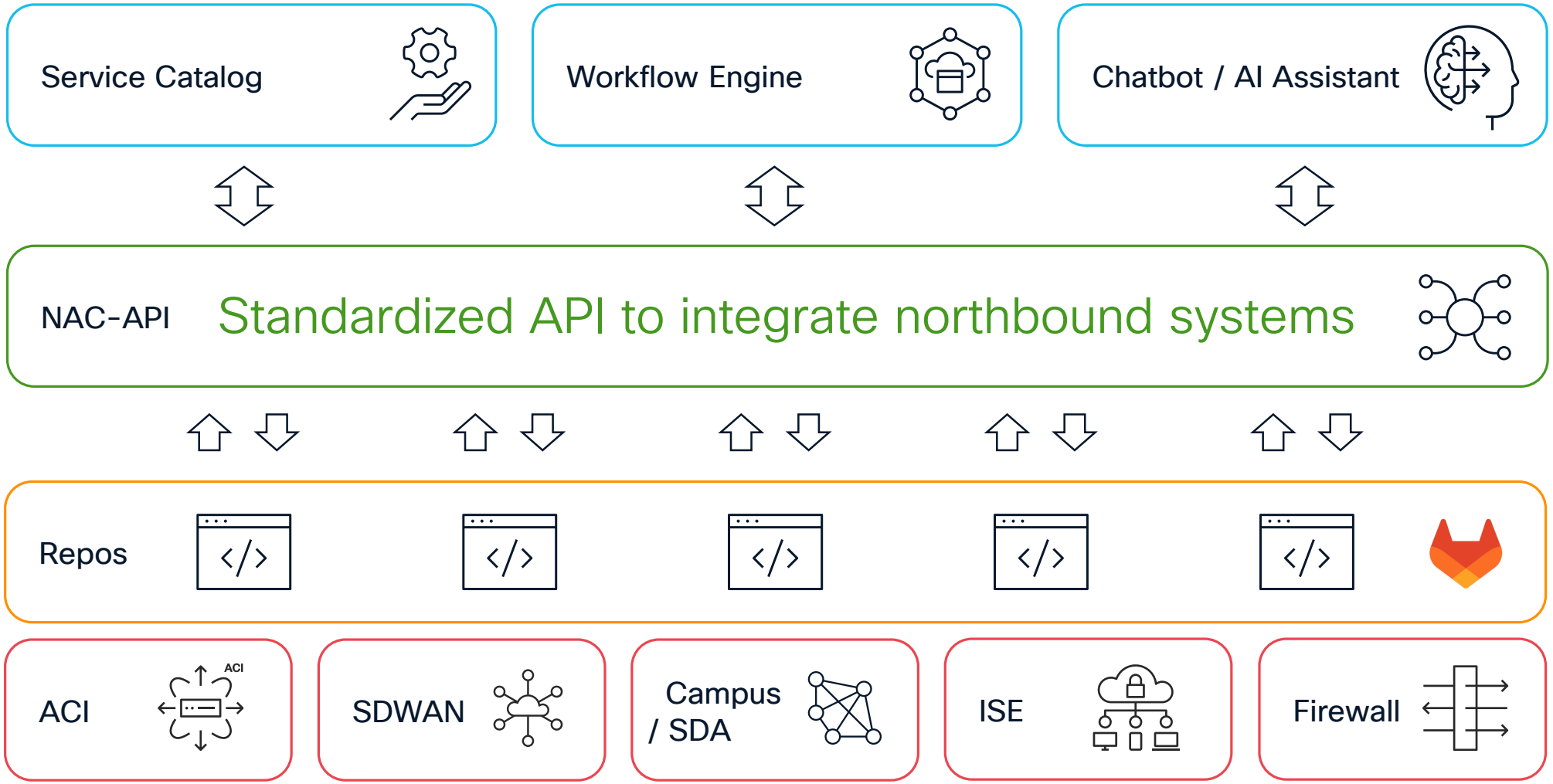
Easy consumption of many architectures in the same way

Digital Infrastructure interaction – API

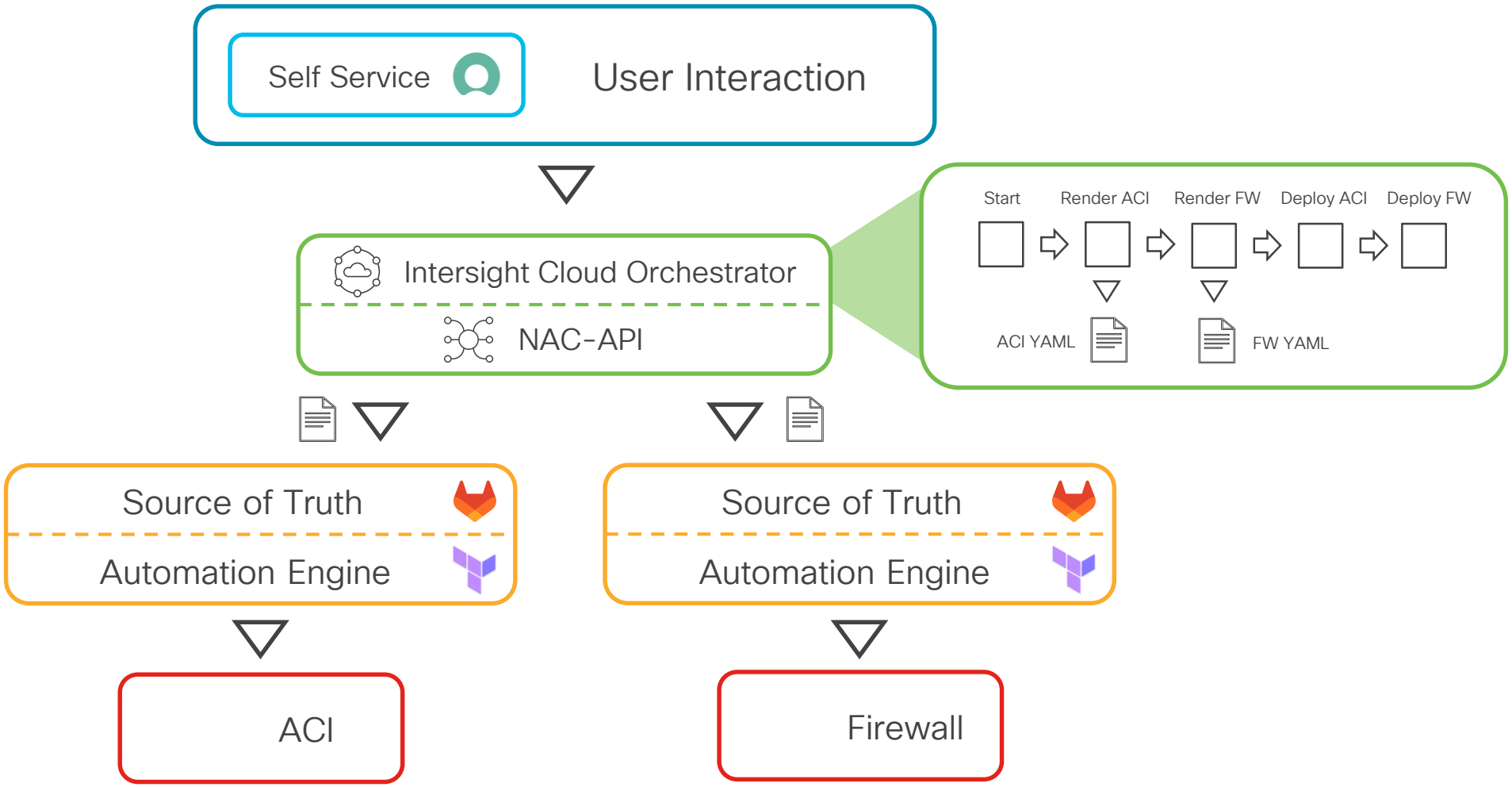
ACI | NDFC | SDWAN | ISE | Catalyst Center | Firewall | Meraki | NX-OS | IOS-XR | IOS-XE | Splunk | Thousand Eyes | 3rd parties

Shown in this presentation

Network as Code – Cross Domain Automation



Network as Code – Cross Domain Automation



Services as Code value insights (1)

My network as designed doesn't match reality and all networks look different

Network changes consume too many resources with too many errors

I spend more of my time reacting to network needs

The date model ensure constancy across your IT

Network changes are automated and simplified to release resources and avoid errors

Network / IT needs are abstracted and simplified

Services as Code value insights (2)

I’m questioning network and security compliance

I’m challenged to apply automation

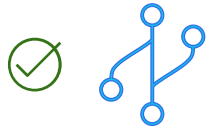
Skill shortages and low innovation

The services as code journey ensures compliance and enables easy audits

Cisco enables your automation journey for and with you!

Leveraging the power of NetDevOps let you enhance your skills and innovate

Services as Code benefits



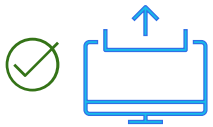
Version control

Configuration files can be versioned,

- track changes,
- collaborate,
- roll back to previous versions if needed.

Automation

- Code-based configuration allows for automation of provisioning, deployment, and scaling processes.
- Infrastructure can be created or modified programmatically, making it easier to manage complex and dynamic environments.

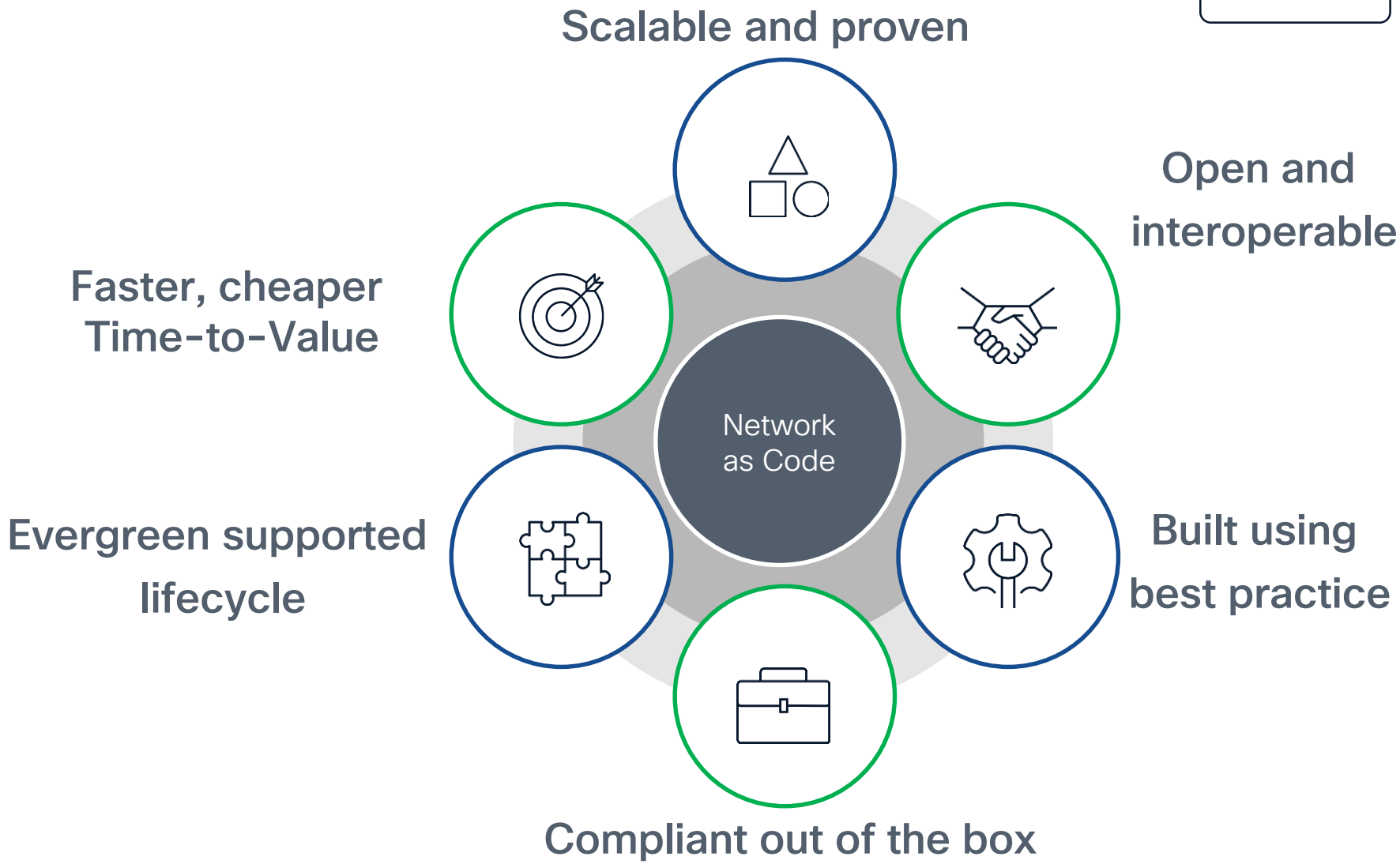


Testing and validation

- Configuration code can be tested, validated, and integrated into continuous integration and continuous deployment (CI/CD) pipelines,
- improving the reliability and quality of infrastructure changes.

Why buy Cisco Service as Code?

3 months to ... → ...less than 1 hour



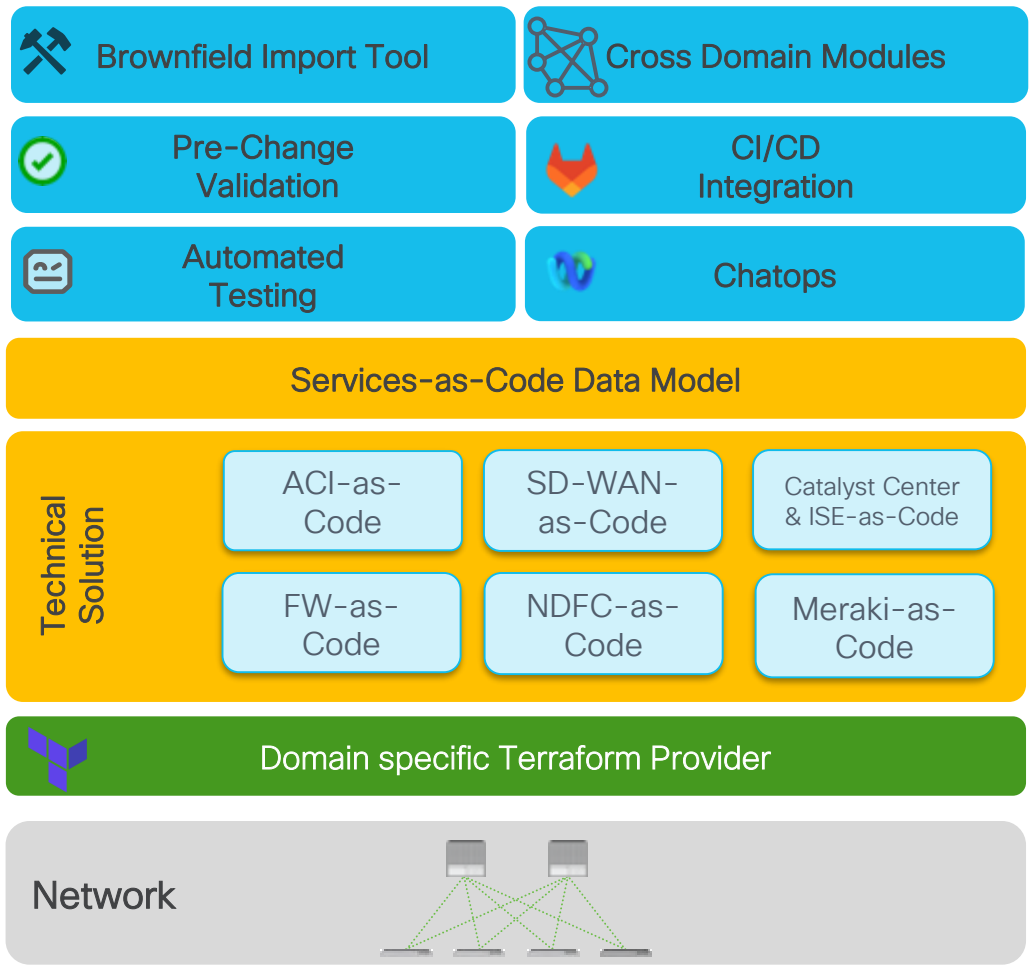
Summary and Conclusion

Optimized and aligned

- Unique design
- Long build time
- Questionable outcome
- Extensive skillset
- No warranty
- No anticipated costs
- Focus on build

- Reusable and proven design
- Time 2 Market optimized
- Defined outcomes
- Leverage existing skills
- Cisco CX Supported
- Cost efficiency & control
- Focus on USE

Building Blocks for Service-as-Code



CX Best Practices documented in scripts and templates and available via Cisco CX Service-as-Code offer. These are the key differentiator for Cisco's automation services. Available via:

- **AS-T for solution deployment**, best practises and on-boarding
- **LCS for continues support** operation, TACSW Lifecycle Management and support for complete tool chain

Solution Specific data model and specific tools, owned by CXPM.

Automation library maintained by Cisco and available as open-source. These are **available and customers are already using them** to build custom automation solutions.

Services as Code streamlines operation and enhanced business efficiency.

My volunteering (hobby) with and for sustainability

References

- ACI/Nexus-as-Code
  https://cisco.com/go/nexusascode
- Demo Repository
  https://github.com/netascode/BRKDCN-2673-Demo
- ACI Terraform Provider
  https://registry.terraform.io/providers/CiscoDevNet/aci/latest
- Pre-Change Validation Tool
  https://github.com/netascode/iac-validate
- Test Automation Tool
  https://github.com/netascode/iac-test
- NX-OS, IOS-XE, IOS-XR Terraform Providers
  https://registry.terraform.io/search/providers?q=netascode
- If you want to have more details, please contact us via session Webex room

- Public:
  - iac-validate: Used for pre-deployment validation
  - terraform-provider-sdwan: Terraform provider for SD-WAN
  - terraform-sdwan-nac-sdwan: Terraform modules for SD-WAN as a Code

Action items for you after this session

Relax its sounds more complex than it is
☺ Therefore:

Download Network as Code and start enjoying
- Play with the Data Model
- Apply against your LAB or simulator
- Send us feedback

As Angus said:
Have a Drink as Code on me

Complete Your Session Evaluations

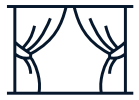
Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn exclusive prizes!

Complete your surveys in the Cisco Live mobile app.

Continue your education



Visit the Cisco Showcase for related demos

Book your one-on-one Meet the Engineer meeting

Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs

Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact me at: maharbec@cisco.com

Thank you

cisco Live !


CISCO