

Network-as-Code in Practice: Safer Change Operations with CI/CD, Testing, and Data Models

Practical implementation approaches operators
can adopt today

Navin Suvarna – Principal Architect – Cisco Systems

01-JAN-2026

Agenda

- 01 Why
- 02 What
- 03 How
- 04 Recorded Demo

The background of the slide is a solid blue color with a complex, low-poly geometric pattern. This pattern is composed of numerous triangles of varying sizes and shades of blue, creating a textured, crystalline effect. The triangles are arranged in a way that they seem to flow and interlock across the entire surface.

Why ?

Network Operations Challenges – Real World



Manual Changes



Lack of Version Control



Configuration Drift



Lack of Testing & Validation



Best Practice Violations

Network-as-Code (NAC) – Operating Model Outcomes



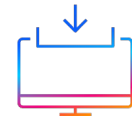
Simplify

Configuration
Consistency



Optimize

Fewer Rollbacks



Test & Validate

Faster Service
Onboarding

Clear Change
Ownership

Unified Automation

The image features a solid blue background with a complex, low-poly geometric pattern of various shades of blue. The pattern consists of numerous triangles and polygons of different sizes, creating a textured, crystalline effect. On the left side of the image, the text "What ?" is displayed in a bold, white, sans-serif font.

What ?

Terraform Primer – Declarative IAC Tool

Terraform is a Declarative Infrastructure as Code (IAC) Tool

```
provider "router" {  
  username = admin"  
  password = NANOG123"  
  url      = https://10.1.1.1"  
}
```

```
resource "router_vrf" "VRFs" {  
  name      = VRF-PROD"  
  rd        = 100:10.1.1.1"  
}
```

```
resource router_ospf "OSPF" {  
  process_id      = 10  
  router_id       = 10.1.1.1  
  passive_default = true  
}
```

NAC vs IAC Tools

Native Terraform – Define Variables

```
variable "vrfs" {  
  default = {  
    VRF1 = {  
      name = "VRF-PROD"  
    },  
    VRF2 = {  
      name = "VRF-DEV"  
    }  
  }  
}
```

```
variable "ospf" {  
  default = {  
    OSPF10 = {  
      process_id = 10  
    },  
    OSPF20 = {  
      process_id = 20  
    }  
  }  
}
```


NAC vs IAC Tools



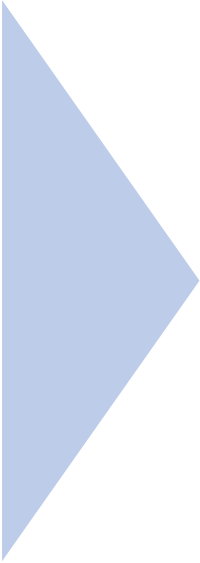
Native Terraform – Loops + Dependencies

```
resource "router_vrf" "VRFs" {  
  for_each = var.vrfs  
  name     = each.value.name  
}  
  
resource "router_ospf" "OSPF" {  
  depends_on = [router_vrf.VRFs]  
  for_each   = var.ospf  
  id         = each.value.process_id  
}
```



NAC vs IAC Tools

NAC – Simplified Config Intent



router:

devices:

- name: router-1

url: https://10.10.20.48

configuration:

vrfs:

- name: "VRF-PROD"
- name: "VRF-DEV"

routing:

ospf_processes:

- id: 10
vrf: "VRF-PROD"
- id: 20
vrf: "VRF-UAT"

NAC Separates Data from Code

 Input YAML - Data

```
router:
  devices:
    - name: router-1
      url: https://10.10.20.48
      configuration:
        vrfs:
          - name: "VRF-PROD"
          - name: "VRF-DEV"
```

```
    routing:
      ospf_processes:
        - id: 10
          vrf: "VRF-PROD"
        - id: 20
          vrf: "VRF-UAT"
```

NAC Separates Data from Code

 Schema YAML - Code

```
vrfs:
  name: str()
  description: str(required=False)
  rd: regex("^(\\d{1,5}:\\d{1,10})| ... required=False)

routing_ospf_processes:
  id: any(int(min=1, max=65535), required=False)
  vrf: str(required=False)
```

NAC Separates Data from Code

 Module TF - Code

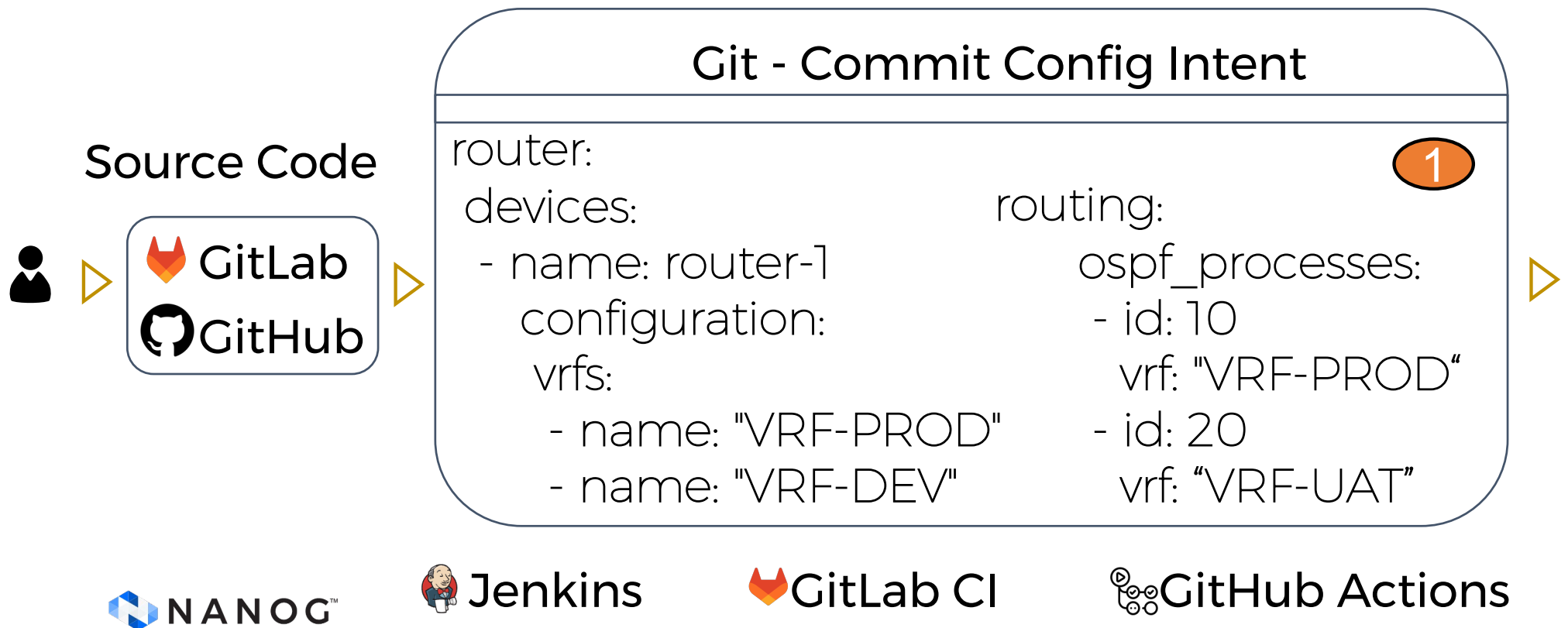
```
locals {  
  vrf = flatten([  
    for device in local.devices : [  
      for vrf in try(local.vrfs, []) : {}  
    ]  
  ])
```

```
resource "router_vrf" "vrfs" {  
  name      = each.value.name  
  description = each.value.desc  
  rd        = each.value.rd  
}
```

```
locals {  
  ospf = flatten([  
    for device in local.devices : [  
      for ospf in try(local.ospf, []) : {}  
    ]  
  ])
```

```
resource "router_ospf" "ospf" {  
  id          = each.value.process_id  
  vrf         = each.value.vrf  
  depends_on = [router_vrf.vrfs]  
}
```

Git-Driven Change Workflow – Validate, Review, Deploy, Notify



Git-Driven Change Workflow – Cont'd

Validate Config Intent

\$ terraform fmt
\$ nac-validate

2

router:

devices:

- name: router-1

configuration:

vrfs:

- name: "VRF-PROD" ✓

- name: "VRF-DEV" ✓

routing:

ospf_processes:

- id: 10 ✓

vrf: "VRF-PROD" ✓

- id: 20 ✓

vrf: "VRF-UAT" ✓

Plan and Deploy

3

\$ terraform init
\$ terraform plan
\$ terraform apply



Git-Driven Change Workflow – Cont'd

Test Operational State

4

VRF. Robot:
Verify VRF {{VRF.name}}
\${r} Get on session
Should be equal \${r.json}

OSPF. Robot:
Verify OSPF {{OSPF.id}}
\${r} Get on session
Should be equal \${r.json}

\$ nac-test

Notify

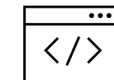
5



Chat

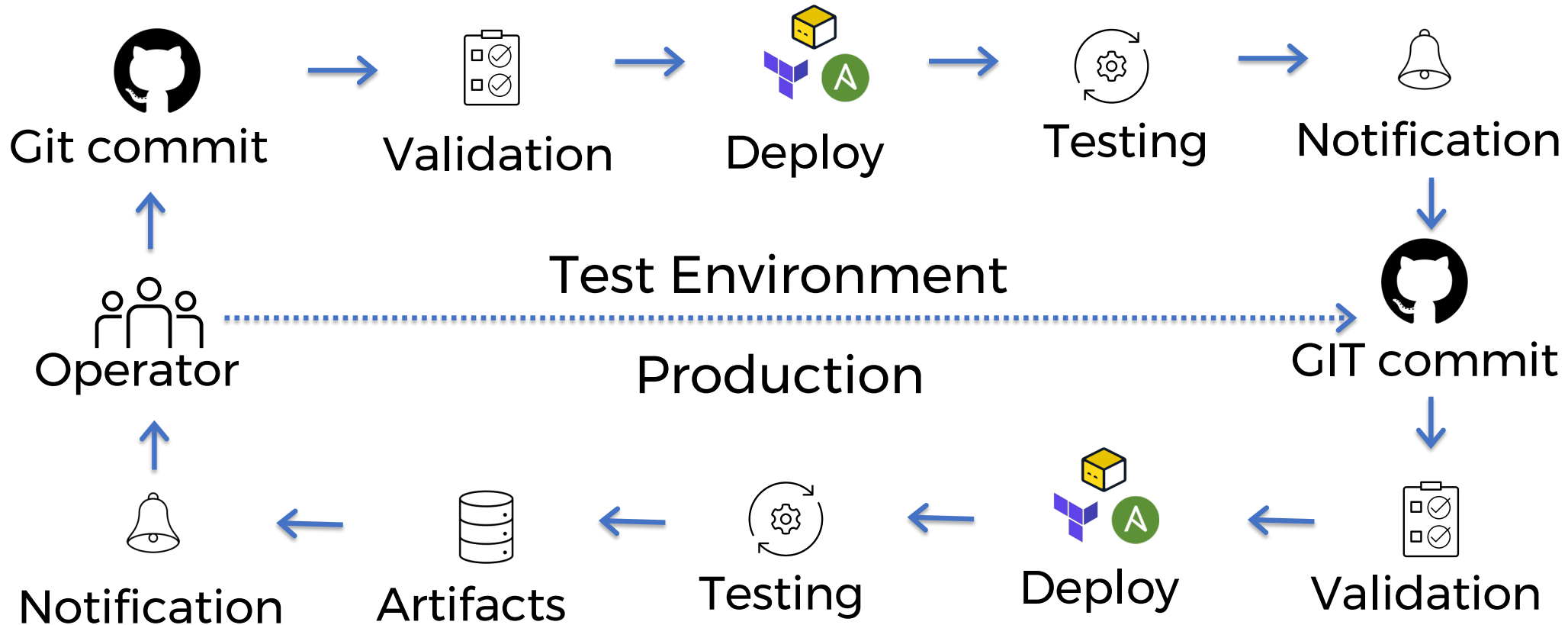


Emails

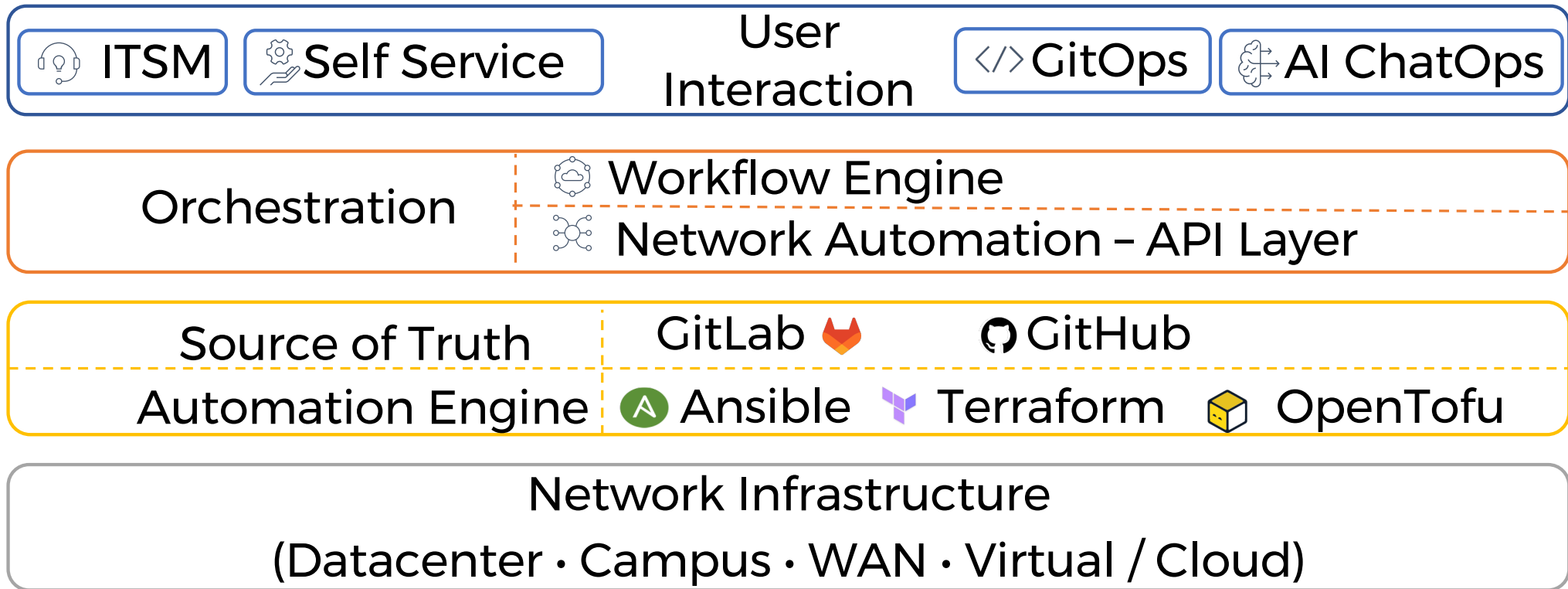


Webhooks

Automation Operational Workflow



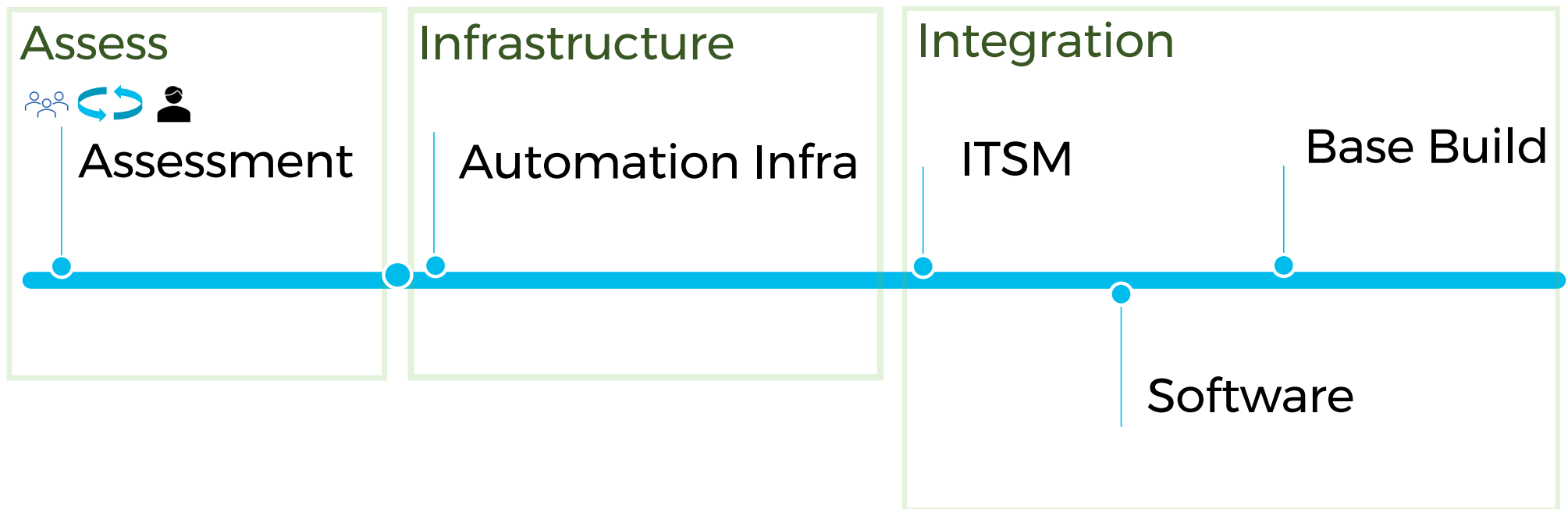
Reference Architecture



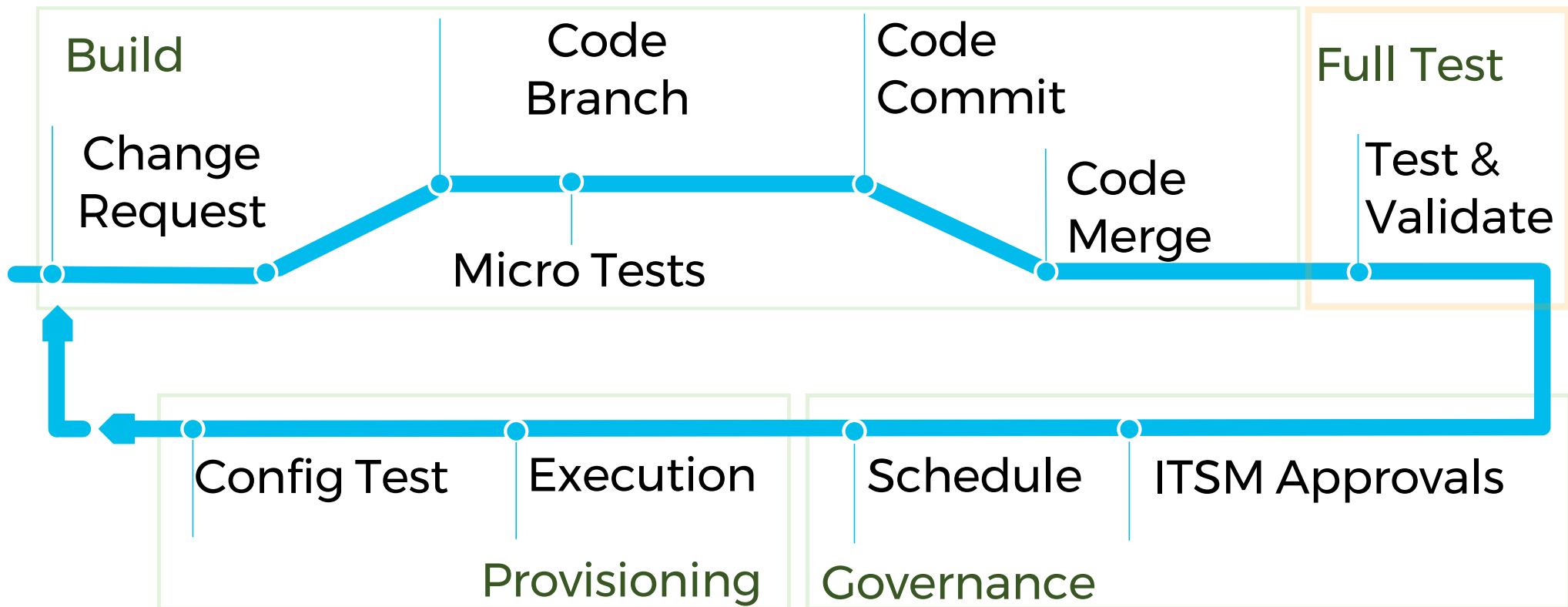
The background of the slide is a solid blue color with a complex, low-poly geometric pattern. This pattern is composed of numerous triangles of varying sizes and shades of blue, creating a textured, crystalline effect. The triangles are arranged in a way that they seem to flow across the entire surface.

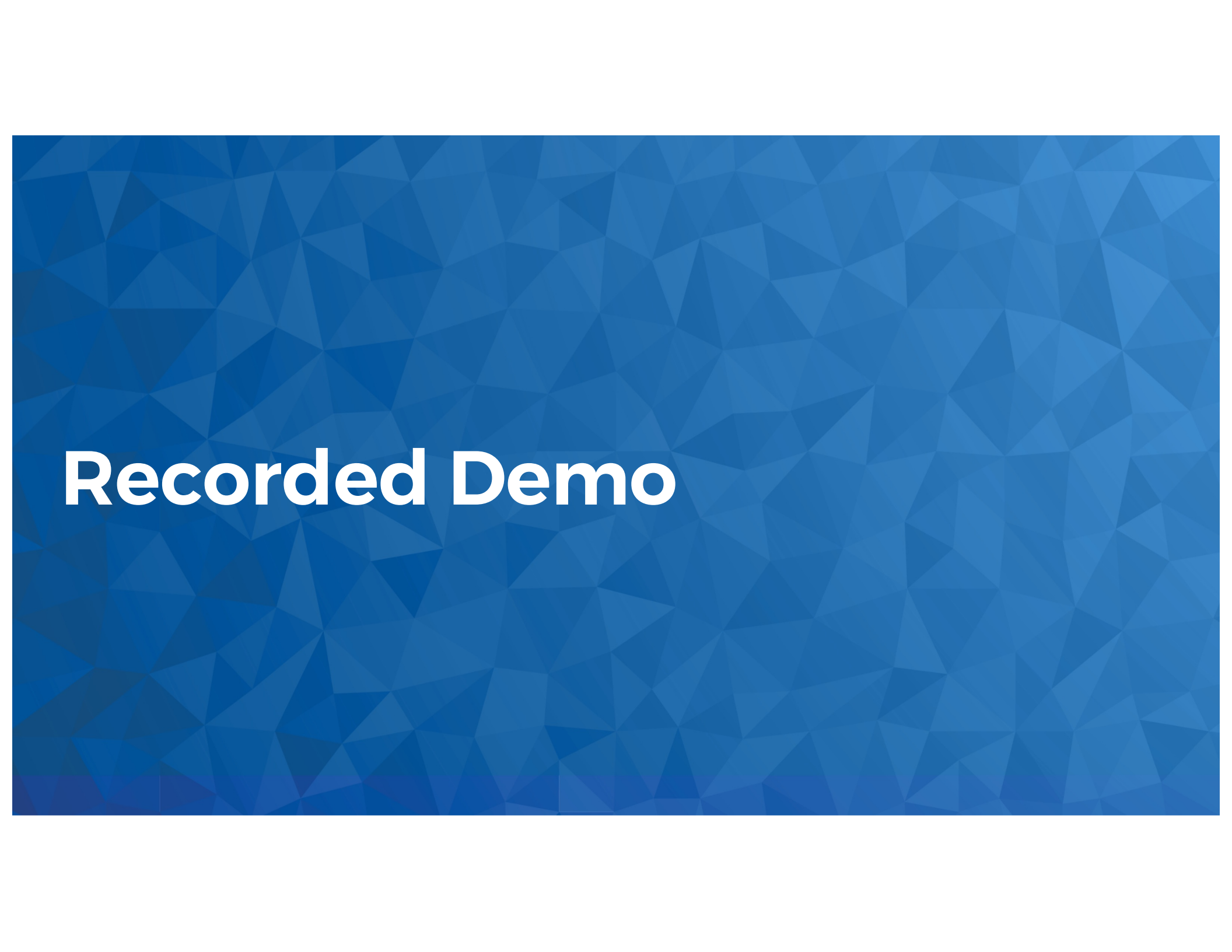
How ?

Incremental Adoption



Incremental Adoption



The background of the slide is a solid blue color with a complex, low-poly geometric pattern. It consists of numerous triangles of various sizes and shades of blue, creating a textured, crystalline effect.

Recorded Demo

File Edit Selection View Go Run ...

Search: nac-iosxe [SSH: p6-u-jumphost]

! access_switches.nac.yaml M X ! access-sw01.nac.yaml ! access-sw02.nac.yaml ! schema.yaml 102_trunk_validation.py

data > ! access_switches.nac.yaml > {} iosxe > {} global

2 iosxe:

4 # GLOBAL SECTION: Network-wide defaults for Access Layer

5 # =====

6 global:

7 variables:

8 # Global variables for access layer

9 company_name: "NANOG96"

10 domain_name: "nanog.org"

11 snmp_community: "public"

12

13 # Global configuration for all access switches

14 configuration:

SSH: p6-u-jumphost master* 0 0 0 Navin Suvarna (3 months ago) Ln 6, Col 10 Spaces: 2 UTF-8 LF {} YAML Network-as-Code Data Model



Thank you