

Beyond NetBox : YANG, GitOps & the Future of Network Automation



Speaker

Eugène NGONTANG

Head of Engineering, Chief Research & Technology Officer @ SPITZKOP

Platform Engineering & DevNetOps Advocate

With contributions from:

Richard KLEIN,

Senior Network & Security Analyst @ SPITZKOP

Fabrice MBIDA,

Data Platform Engineer @ SPITZKOP

Younes Jellabi (Airbus)

Alexis Lameire (FRnOG community)





INTRODUCTION

This presentation delves into leveraging YANG and GitOps within a Platform Engineering framework from SPITZKOP Sovereign Network Automation Platform (SNAP), to revolutionize network automation, addressing current challenges, showcasing practical solutions, and exploring future AI integrations.



FRnOG

The logo for FRnOG is displayed in a large, bold, sans-serif font. The letters 'FR' and 'OG' are black, while the 'n' is red. The background features a complex network of blue and green lines connecting various nodes, creating a digital mesh effect.

YANG

The word 'YANG' is written in a large, bold, blue sans-serif font. It is positioned below the 'FRnOG' logo and is also set against the same network background.

GitOps

The logo for GitOps features the word 'Git' in a green sans-serif font, followed by 'Ops' in a blue sans-serif font. A green gear icon is placed to the left of 'Git', and a blue circular icon containing a code snippet is placed between 'Git' and 'Ops'.

SPITZKOP SNAP (Sovereign Network Automation Platform)

Transforms network operations from manual, error-prone processes into declarative, policy-driven automation.



Core Capabilities & Benefits

- Built on open standards (YANG, GitOps, NETCONF)
- Vendor-agnostic network automation
- Validates every configuration change
- Enforces compliance policies
- Delivers 90% error reduction
- Achieves 10x faster deployments



Differentiated Workflow

- Manages multi-vendor infrastructures
- Single, Git-based workflow
- Built-in validation and audit trails
- AI-assisted optimization
- Eliminates need for vendor-specific expertise



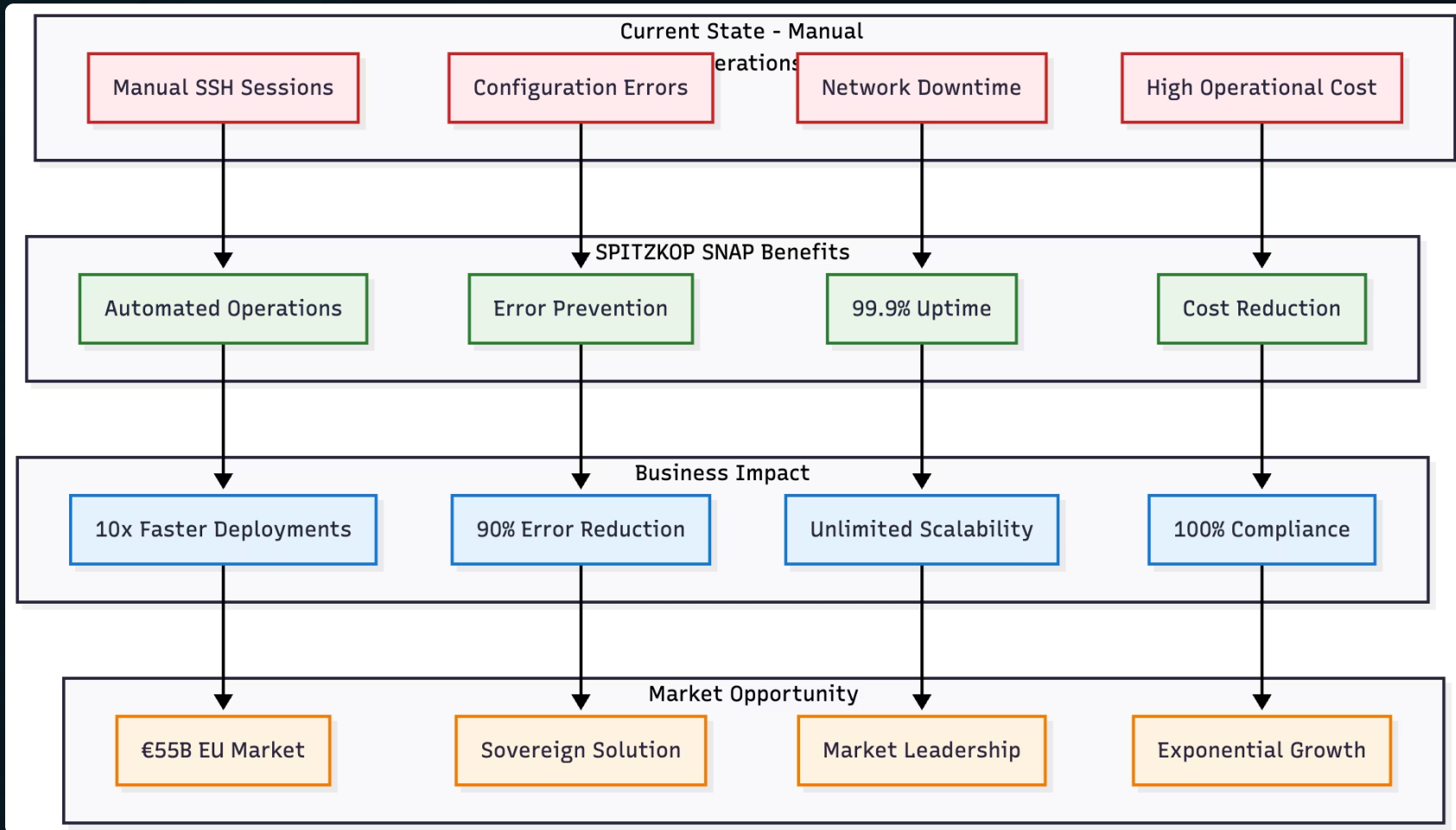
Sovereign & Market Focus

- Europe's first sovereign platform
- Addresses the €55 billion European market
- GDPR-compliant and data-resident solutions
- Eliminates vendor lock-in
- Provides enterprise-grade governance, security, and scalability

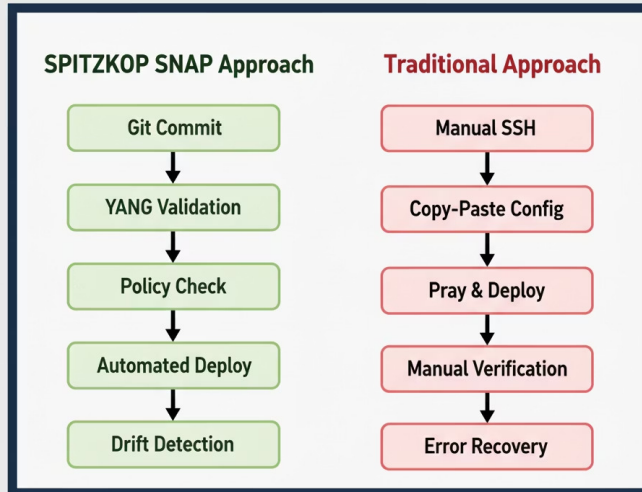
SNAP doesn't just automate networks — it transforms them into intelligent, self-validating infrastructure that adapts, learns, and evolves.



The €1.1M Problem: Why Network Operations Are Broken



From 'Pray & Deploy' to 'Validate & Automate'

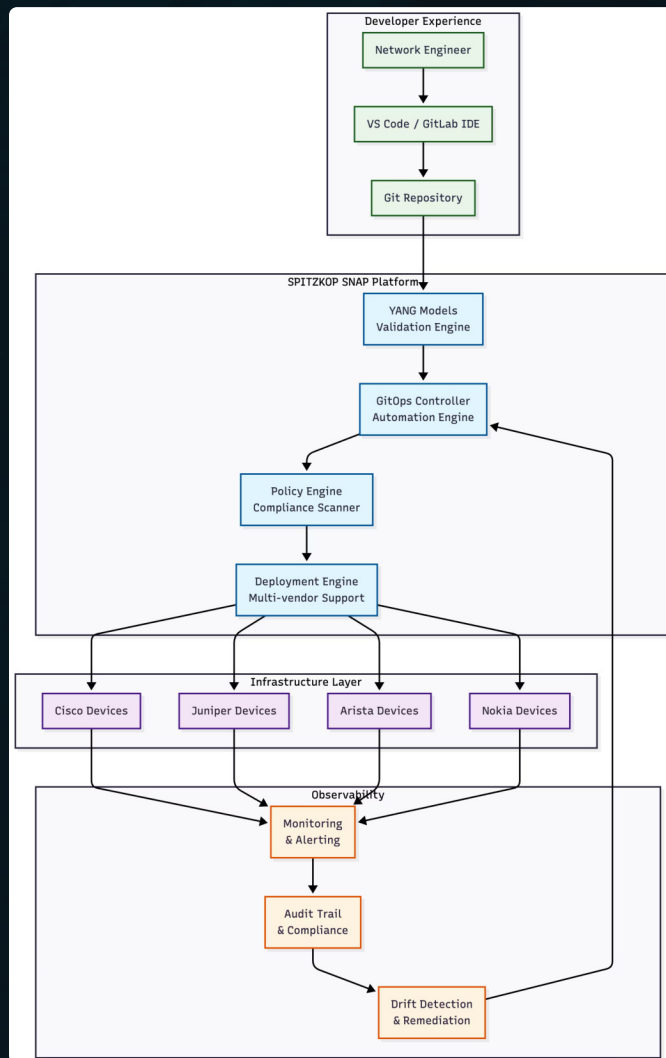


The Certainty of Validation-First Automation

The difference is profound. The Traditional approach asks: *"Will this work?"* In contrast, the SNAP approach declares with confidence: *"This will work."* That's the power and assurance that comes from validation-first automation.



Meet SNAP: The Platform That Makes Network Automation Actually Work



Bridging the Gap

SNAP unites developer experience with infrastructure reality.

Developer Experience

Network Engineers use VS Code, GitLab, and Git for a single source of truth.

SNAP Platform Core

YANG validation, GitOps automation, policy compliance, multi-vendor support.

Vendor Agnostic

Supports Cisco, Juniper, Arista, Nokia, and more – SNAP speaks their language.

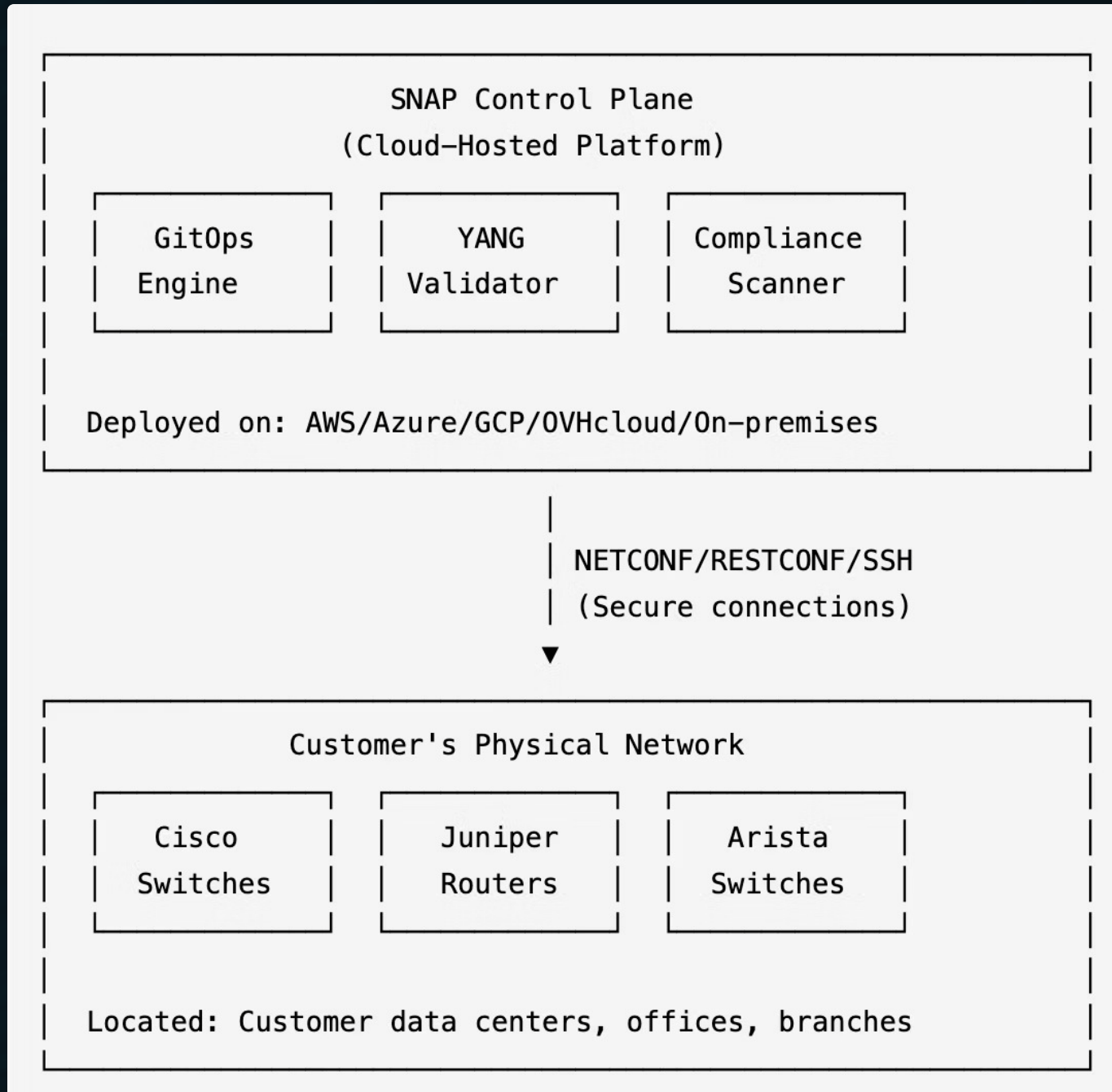
Continuous Feedback

Monitoring, audit trails, and drift detection enable intelligent adaptation.

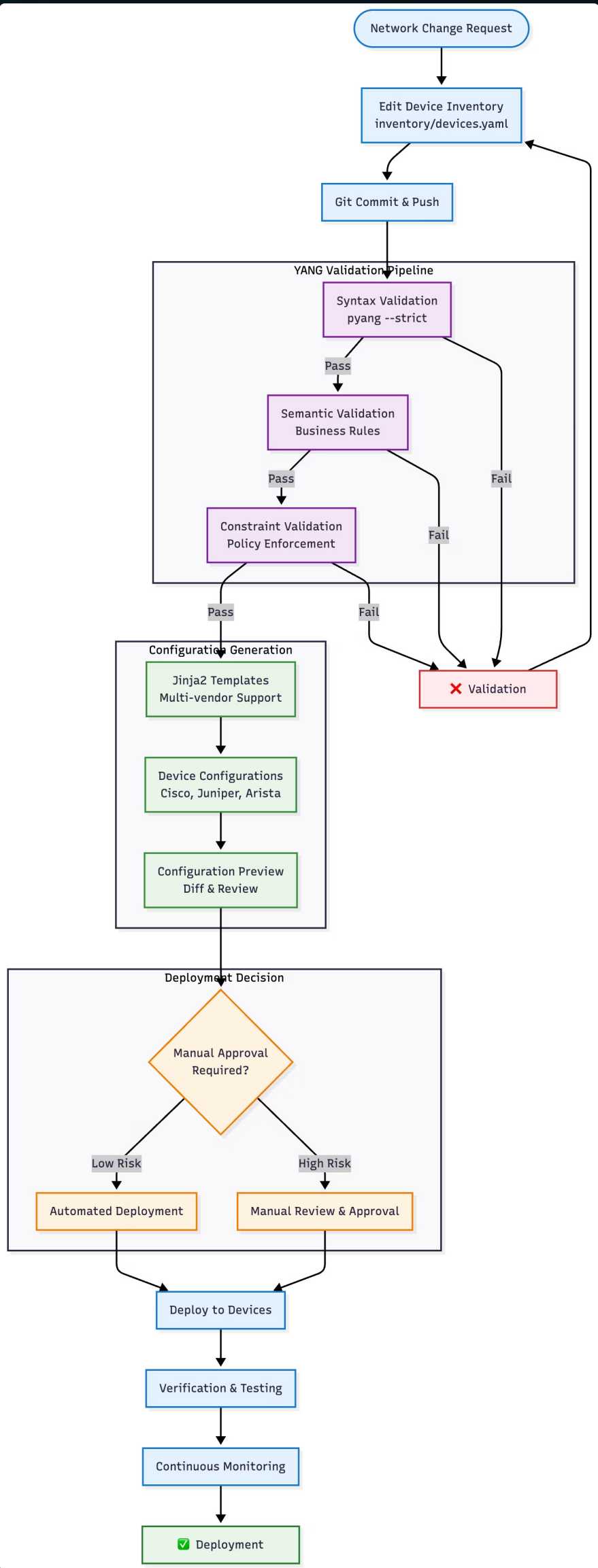
Empowered Engineers

Network engineers work their way with validated, automated, auditable changes.

SNAP High Level Design Architecture



How YANG Prevents the €300K Typo



The €300K Typo

SNAP prevents the '€300K typo' – that single character mistake that can bring down your network.



Engineer Initiates Change

A network engineer edits the device inventory file and commits to Git, initiating the process.



Automated Validation Pipeline

A robust validation pipeline immediately checks for errors, preventing them before deployment.

If any step fails, the change is rejected, safeguarding the network.



Configuration Generation & Preview

If validation passes, Jinja2 templates generate multi-vendor configurations. Engineers receive a preview to review the diff.



Deployment Decision

Low-risk changes deploy automatically, while high-risk changes require manual approval for an extra layer of control.



Deployment & Feedback Loop

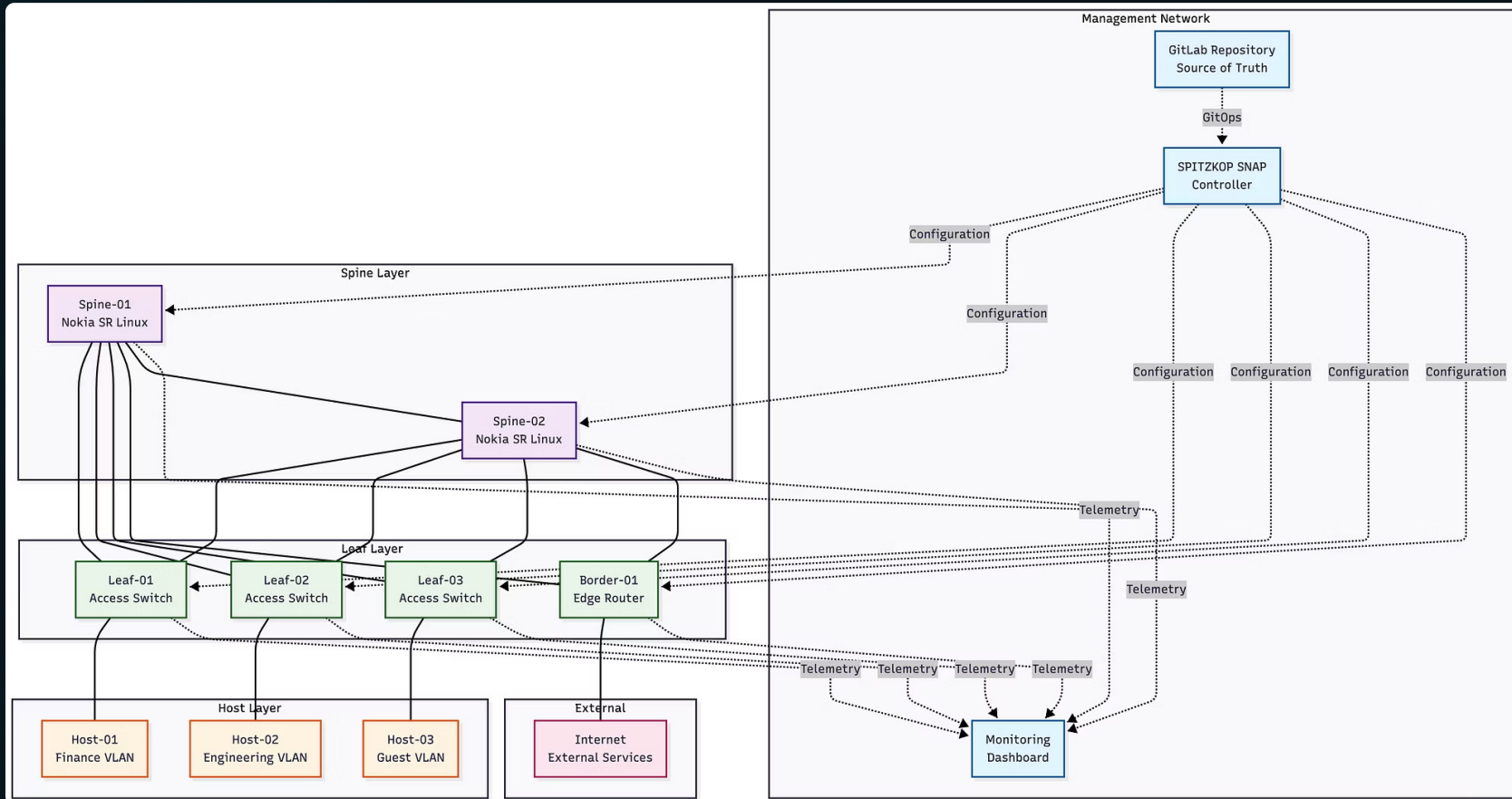
Changes are deployed to devices, followed by verification testing and continuous monitoring. This creates a critical feedback loop:



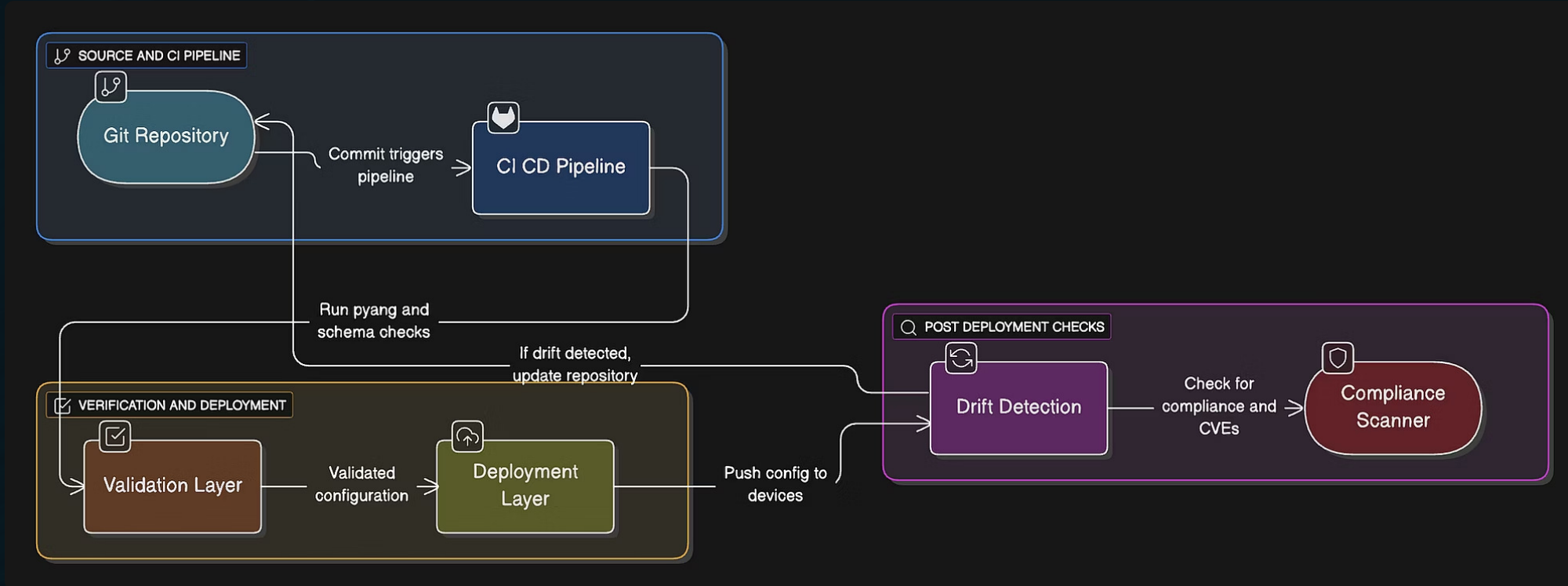
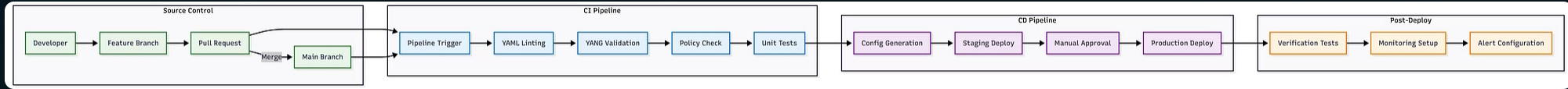
Proactive Automation

This process transforms network operations from reactive firefighting into proactive, predictable automation.

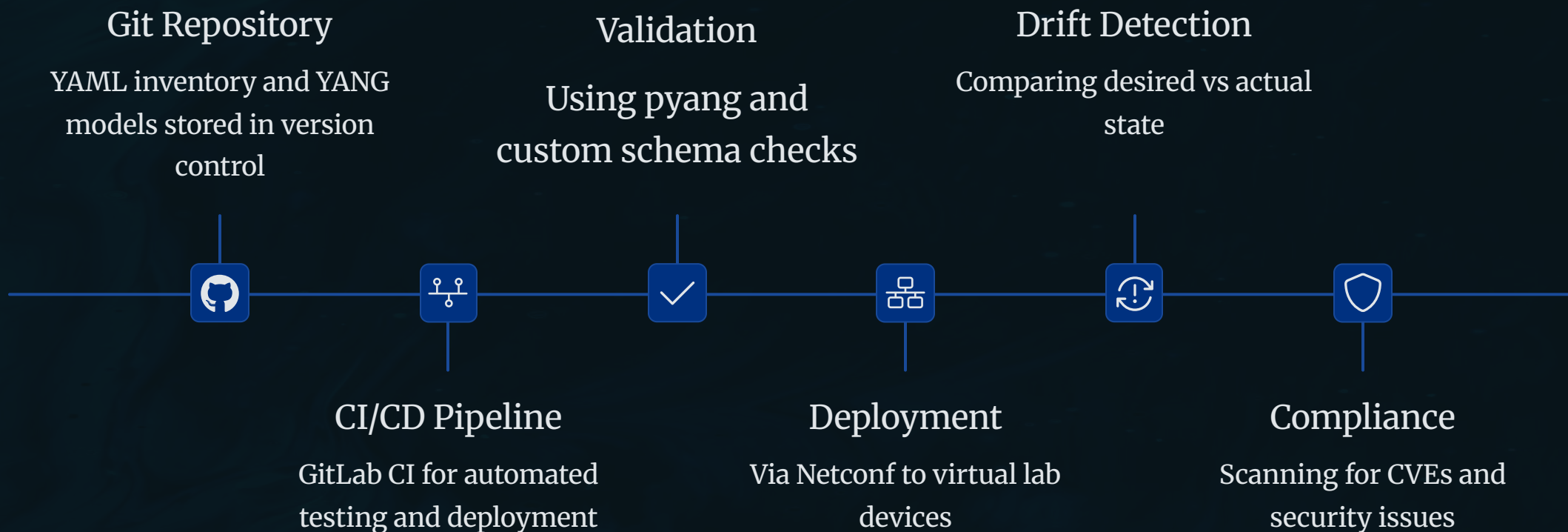
The Lab That Proves It Works: Our Live Demo Environment



GitOps for Networks: When Infrastructure Becomes Code



The GitOps Architecture



This architecture embodies Platform Engineering principles by abstracting complexity, enabling self-service, and providing reusable automation components.

Demo preview – NetBox Device Inventory

Search...

</

Demo preview – NetBox Device Configuration Interface

Search...

Devices / DM-Akron

spitzkop-snap-rtr01

Created 2020-12-20 00:00 · Updated 2025-10-19 15:53

+ Add Components

Bookmark

Subscribe

Clone

Edit

Delete

Device

Interfaces 14

Console Ports 2

Power Ports 1

Config Context

Render Config

Contacts

Images

Journal

Changelog

Device

RegionNorth America / United States / Ohio

SiteDM-Akron

Location—

RackComms closet

PositionU4 / Front

GPS Coordinates—

TenantCustomers / Dunder-Mifflin, Inc.

Device TypeCisco ISR 1111-8P (1U)

Description—

Airflow—

Serial Number—

Asset Tag—

Config Template—

Tags

No tags assigned

Comments

None

Virtual Device Contexts

NAME

IDENTIFIER

STATUS

TENANT

IP ADDRESS

No virtual device contexts found

Management

StatusActive

RoleRouter

PlatformCisco IOS

Primary IPv4—

Primary IPv6—

Out-of-band IP—

Application Services

NAME

PARENT

PROTOCOL

PORTS

DESCRIPTION

No application services found

Images

IMAGE

OBJECT

DESCRIPTION

DIMENSIONS

SIZE

No image attachments found

Dimensions

Height1.0U

Weight—

Front

Comms closet

Rear

12

11

10

9

8

7

6

5

4

48-Port Patch Panel

dmi01-akron-sw01

spitzkop-snap-rtr01

12

11

10

9

8

7

6

5

4

dmi01-akron-sw01

Demo preview – NetBox Configuration Template

nixmind
Admin

Help

Add a new export template

Create

Export Template

Name*

Object types*

Description

Template code

Data Source

Data source

File

☐ Auto sync enabled

Enable automatic synchronization of data when the data file is updated

Rendering

MIME type

Defaults to text/plain; charset=utf-8

File name

Filename to give to the rendered export file

File extension

Extension to append to the rendered filename

Cancel

Create

Create & Add Another

Demo preview – NetBox Change Tracking/Audit Log

Search...

nixmind Admin

Change Log

Export

Results 30Filters

Quick search

Configure Table

TIME	USERNAME	ACTION	TYPE	OBJECT	MESSAGE
2025-10-19 15:53	nixmind	Updated	Device	spitzkop-snap-rtr01	—
2025-10-19 15:52	nixmind	Updated	Device	spitzkop-essingan-pdu01	—
2025-10-19 15:41	demo1	Updated	Inventory Item	SFP_200 (test label1)	—
2025-10-19 14:12	bid	Created	ASN	AS65478	—
2025-10-19 13:28	testing	Created	Config Context	testConfigContext1	testing stuff 123
2025-10-19 12:13	demo1	Updated	Inventory Item	SFP_200 (test label)	—
2025-10-19 12:09	admin	Deleted	Site Group	<script>alert('XSS')</script>	—
2025-10-19 12:08	admin	Created	Site Group	<script>alert('XSS')</script>	<script>alert('XSS')</script>
2025-10-19 10:21	bid	Created	Inventory Item	SFP_200	—
2025-10-19 10:20	bid	Created	Inventory Item	SFP_200	—
2025-10-19 07:34	adm78	Updated	Export Template	Export-tmp	—
2025-10-19 07:33	adm78	Updated	Export Template	Trace multi-étapes	—
2025-10-19 07:29	adm78	Created	Export Template	Export-tmp	—
2025-10-19 07:25	adm78	Updated	Export Template	Trace multi-étapes	—
2025-10-19 07:19	adm78	Updated	Export Template	Trace multi-étapes	—
2025-10-19 07:14	adm78	Updated	Export Template	Trace multi-étapes	—
2025-10-19 07:10	adm78	Updated	Export Template	Trace multi-étapes	—
2025-10-19 07:06	adm78	Updated	Export Template	Trace multi-étapes	—
2025-10-19 06:55	adm78	Created	Export Template	Trace multi-etapes	—
2025-10-19 06:35	mario.rossi	Created	Power Port	PSU3	—
2025-10-19 06:35	mario.rossi	Created	Power Port	PSU2	—
2025-10-19 06:35	mario.rossi	Created	Power Port	PSU1	—
2025-10-19 06:35	mario.rossi	Created	Power Port	PSU0	—

Demo preview – SNAP Sample YANG Model

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
```

```
$ pyang -f tree yang-models/spitzkop-network.yang
```

```
module: spitzkop-network
```

```
+--rw devices
|   +--rw device* [name]
|       +--rw name          string
|       +--rw type          device-type
|       +--rw mgmt-ip       inet:ipv4-address
|       +--rw platform?    string
|       +--rw role          device-role
|       +--rw site?        string
|       +--rw config
|           +--rw hostname?  string
|           +--rw interfaces* [name]
|               +--rw name          string
|               +--rw description?  string
|               +--rw ip-address?   inet:ipv4-prefix
|               +--rw enabled?      boolean
|               +--rw mtu?          uint16
|           +--rw bgp
|               +--rw asn          asn
|               +--rw router-id    inet:ipv4-address
|               +--rw neighbors* [ip]
|                   +--rw ip          inet:ipv4-address
|                   +--rw remote-asn  asn
|                   +--rw description? string
|                   +--rw password?   string
+--rw global-config
|   +--rw ntp-servers*  inet:host
|   +--rw dns-servers*  inet:ipv4-address
|   +--rw snmp
|       +--rw community?  string
|       +--rw location?   string
|       +--rw contact?    string
+--rw compliance
|   +--rw security
|       +--rw required-policies*  enumeration
+--rw operational
|   +--rw required-policies*  enumeration
```

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
```

```
$
```

Demo preview – SNAP Ad Hoc Check CVE

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$ python3 compliance/scan-cve.py
Starting CVE vulnerability scan...
Scanning inventory/devices.yaml...
Scanning yang-models/spitzkop-network.yang...
CVE scan completed. 0 vulnerabilities found.
Security report saved to security-report.json
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$
```

	File: security-report.json
1	{
2	"scan_timestamp": "2025-10-19T17:09:35.912467",
3	"scanner_version": "1.0.0",
4	"total_files_scanned": 2,
5	"vulnerabilities_found": 0,
6	"vulnerabilities": [],
7	"security_score": "A+",
8	"recommendations": [
9	"All configurations follow security best practices",
10	"No known CVE vulnerabilities detected",
11	"YANG model validation prevents insecure configurations"
12	]
13	}

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$
```

Demo preview – SNAP Ad Hoc Check Policy Violation

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$ python3 compliance/policy-check.py
Starting security policy validation...
Checking: Password complexity requirements
Checking: SSH key authentication enforcement
Checking: VLAN isolation policies
Checking: Access control list validation
Checking: Encryption protocol compliance
Policy check completed. 0 violations found.
Compliance report saved to compliance-report.json
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$
```

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$ bat compliance-report.json
```

	File: compliance-report.json
1	{
2	"scan_timestamp": "2025-10-19T17:11:40.294446",
3	"policy_version": "2.1.0",
4	"total_policies_checked": 5,
5	"violations_found": 0,
6	"violations": [],
7	"compliance_score": 100,
8	"status": "COMPLIANT",
9	"recommendations": [
10	"All security policies are properly enforced",
11	"Configuration follows enterprise security standards",
12	"YANG model ensures policy compliance by design"
13	]
14	}

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$
```

Demo preview – SNAP Ad Hoc Check Configuration Drift

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$ python3 drift-detection/compare-state.py
Starting configuration drift detection...
Analyzing 3 devices for configuration drift
Checking device: spine-01 (arista_eos)
✓ No drift detected on spine-01
Checking device: leaf-01 (arista_eos)
✓ No drift detected on leaf-01
Checking device: leaf-02 (cisco_xr)
✓ No drift detected on leaf-02
Drift detection completed:
• Total devices: 3
• Devices with drift: 0
• Compliance: 100.0%
Drift report saved to drift-report.json
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$
```

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$ bat drift-report.json
```

	File: drift-report.json
1	{
2	"scan_timestamp": "2025-10-19T17:13:00.254904",
3	"scan_type": "manual",
4	"total_devices": 3,
5	"devices_with_drift": 0,
6	"compliance_percentage": 100.0,
7	"drift_results": [
8	{
9	"device": "spine-01",
10	"device_type": "arista_eos",
11	"drift_detected": false,
12	"drift_details": [],
13	"last_checked": "2025-10-19T17:13:00.254869",
14	"status": "compliant"
15	},
16	{
17	"device": "leaf-01",
18	"device_type": "arista_eos",
19	"drift_detected": false,
20	"drift_details": [],
21	"last_checked": "2025-10-19T17:13:00.254893",
22	"status": "compliant"
23	},
24	{
25	"device": "leaf-02",
26	"device_type": "cisco_xr",
27	"drift_detected": false,
28	"drift_details": [],
29	"last_checked": "2025-10-19T17:13:00.254899",
30	"status": "compliant"
31	}
32	],
33	"summary": "Drift detection completed: 0/3 devices have configuration drift",
34	"recommendations": [
35	"All devices are compliant with desired configuration",
36	"Continue regular drift monitoring",
37	"Maintain current configuration management practices"
38	]
39	}

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$
```

Demo preview – SNAP Ad Hoc Deploy to Staging

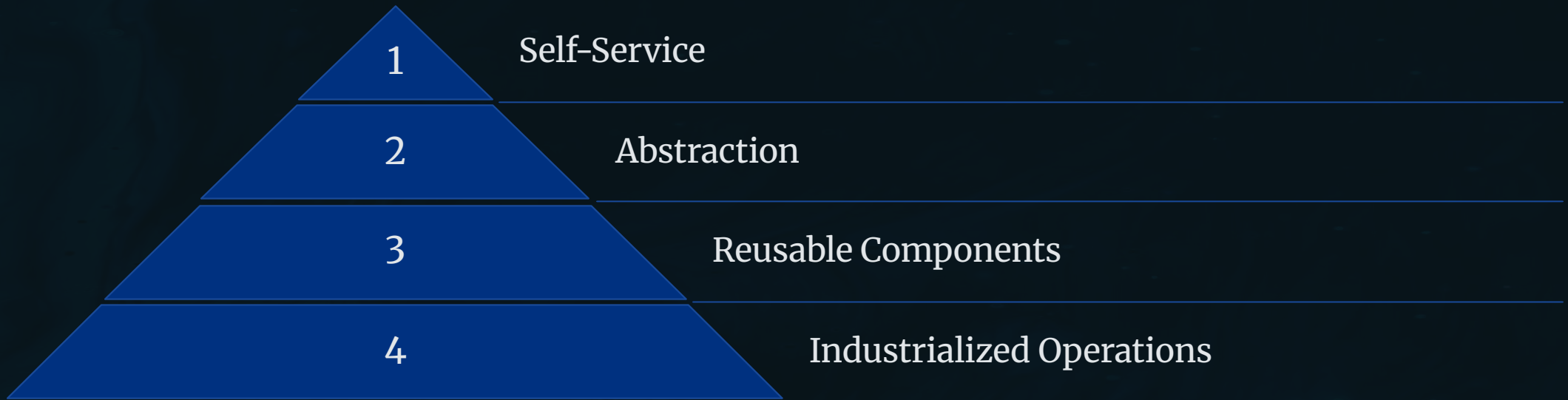
```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$ python3 deployment/push-config.py --environment staging --dry-run
Starting DRY RUN deployment to staging environment...
Found 3 devices in inventory
Processing device: spine-01 (arista_eos)
  ✓ Configuration validated for spine-01
Processing device: leaf-01 (arista_eos)
  ✓ Configuration validated for leaf-01
Processing device: leaf-02 (cisco_xr)
  ✓ Configuration validated for leaf-02
Deployment report saved to deployment-report-staging.json
  ✓ DRY RUN completed successfully - ready for live deployment
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$ |
```

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$ bat deployment-report-staging.json
```

	File: deployment-report-staging.json
1	{
2	"environment": "staging",
3	"deployment_type": "dry_run",
4	"timestamp": "2025-10-19T17:15:15.066144",
5	"total_devices": 3,
6	"successful_deployments": 3,
7	"failed_deployments": 0,
8	"results": [
9	{
10	"device": "spine-01",
11	"status": "validated",
12	"changes": [
13	"VLAN configuration updated",
14	"Interface settings validated"
15	],
16	"timestamp": "2025-10-19T17:15:15.066122"
17	},
18	{
19	"device": "leaf-01",
20	"status": "validated",
21	"changes": [
22	"VLAN configuration updated",
23	"Interface settings validated"
24	],
25	"timestamp": "2025-10-19T17:15:15.066136"
26	},
27	{
28	"device": "leaf-02",
29	"status": "validated",
30	"changes": [
31	"VLAN configuration updated",
32	"Interface settings validated"
33	],
34	"timestamp": "2025-10-19T17:15:15.066142"
35	}
36	],
37	"summary": "Successfully validated configurations to 3 devices"
38	}

```
engontangt001@FR_YW9YPJ6KQ9 > OSX ~/devspace/FRn0G-42/gitops-yang-demo
$
```

Platform Engineering Principles



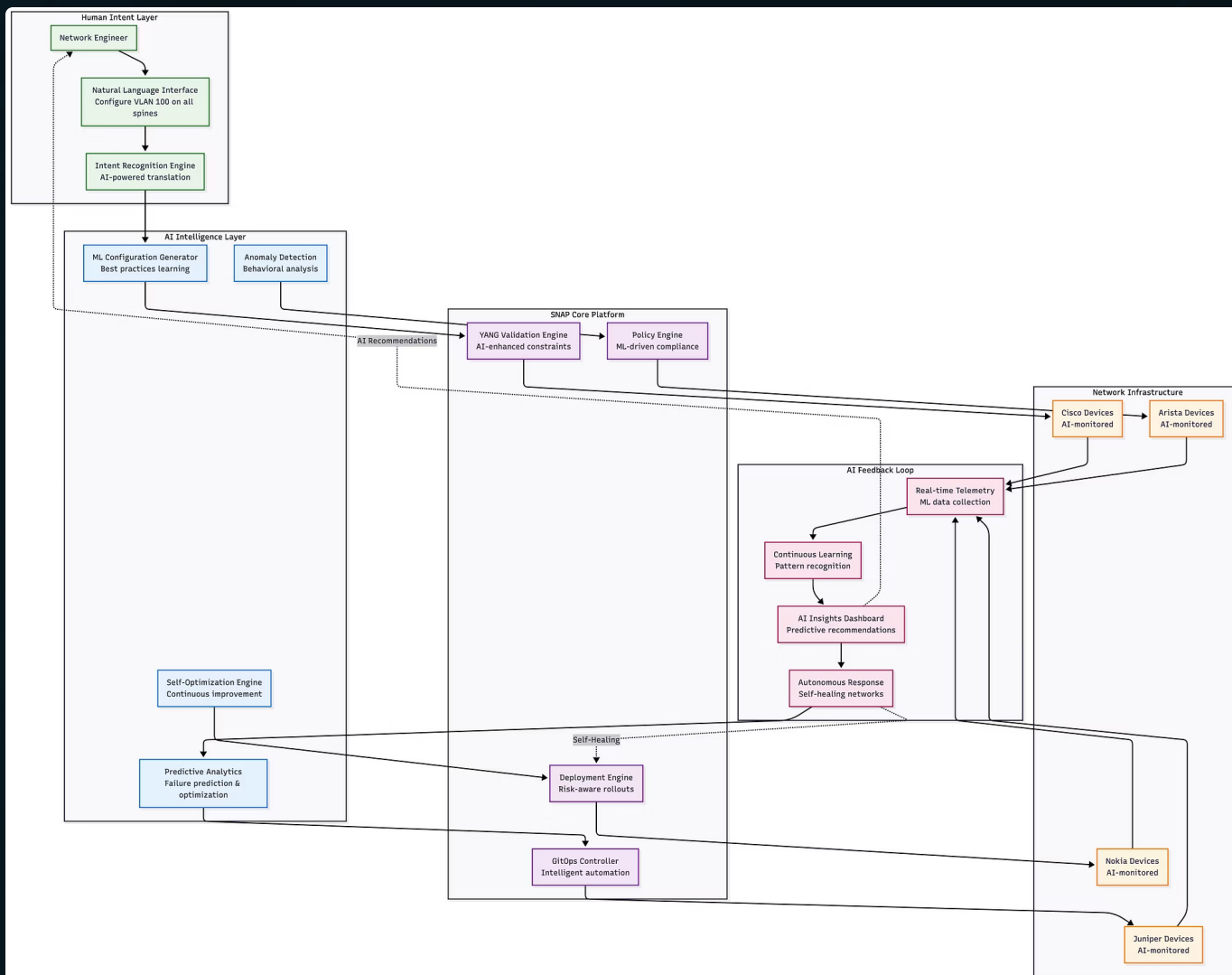
Applying Platform Engineering Principles

Our GitOps pipeline for network automation embodies these Platform Engineering principles, helping to bridge the gap between DevOps and NetOps teams while providing a scalable foundation for network operations.



*By applying these principles, we create a system that enables network engineers to focus on **intent** rather than **implementation details**.*

AI Vision – The Future of Intelligent Network Automation



Intent-Driven Automation

"Configure VLAN 100 on all spine switches for the new finance department."



Advanced AI Intelligence

The AI intelligence layer provides predictive analytics to warn of failures, anomaly detection for real.



Continuous Learning Loop

Every configuration change, performance metric, and incident feeds into a powerful feedback loop.



Roadmap to Autonomy

The future of networking is intelligent, and it starts now with SNAP.

Key Takeaways



YANG as Enabler

YANG can be a powerful enabler of network automation when integrated into GitOps pipelines



Platform Engineering

Platform Engineering principles help industrialize network operations



Complementary Tools

NetBox/Nautobot are useful but need complementary validation layers



AI Future

AI offers promising enhancements for future automation workflows

Thank You for Your Attention!

Questions & Discussion

Connect with Us:

Email: contact@spitzkop.io

LinkedIn: [Eugène NGONTANG](#) | [SPITZKOP](#)

Website: <https://www.spitzkop.io>

Document Access Link: <https://snap.spitzkop.io>

