

The Network Automation Map

Navigate Your Journey with the NAF Framework

Damien Garros
OpsMill

SwiNOG 41

About Me

Damien Garros

Co-founder and CEO
OpsMill



Co-Founder and CEO of OpsMill
Creator of Infrahub, a next generation
Infrastructure data management
platform (Intent / Source of Truth)

Focused on Infrastructure as Code,
Automation & Observability for 12+
years

OpsMill 

 Infrahub



damiengarros



@dgarros

Introduction



Modern networks demand automation that teams can trust, understand, and safely evolve.



Yet, it's still overly complicated for most organizations

Building it is hard, maintaining it is almost impossible



The pattern we keep seeing

Rebuild from scratch

Teams keep recreating the same building blocks—data models, pipelines, integrations


Learn the hard way

Most “best practices” are discovered through outages, drift, and painful rewrites instead of repeatable patterns.

No shared blueprint for what ‘good’ looks like

Without a common map, it’s hard to assess maturity, compare architectures, or even agree on what problem to solve first.






Multiple people from the NAF community got together to find a solution

It quickly turned into a working group with weekly meetings





This doesn't
make any sense!

माइ क्वला हैं!

On ne peut
pas faire ça!

هذا لا يعمل

န s 75 နုဒါး!

No es correcto!

خطن امام بصر

It was hard at first
Everyone was biased by their own tools and habits
Ironically, we were missing a shared lexicon





Collector

Orchestration

Intent

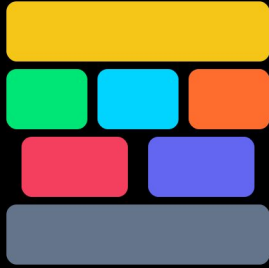
Executor


The solution was to stop focusing on tools and focus on functional building blocks

Quickly, everyone started talking the same language

Presentation



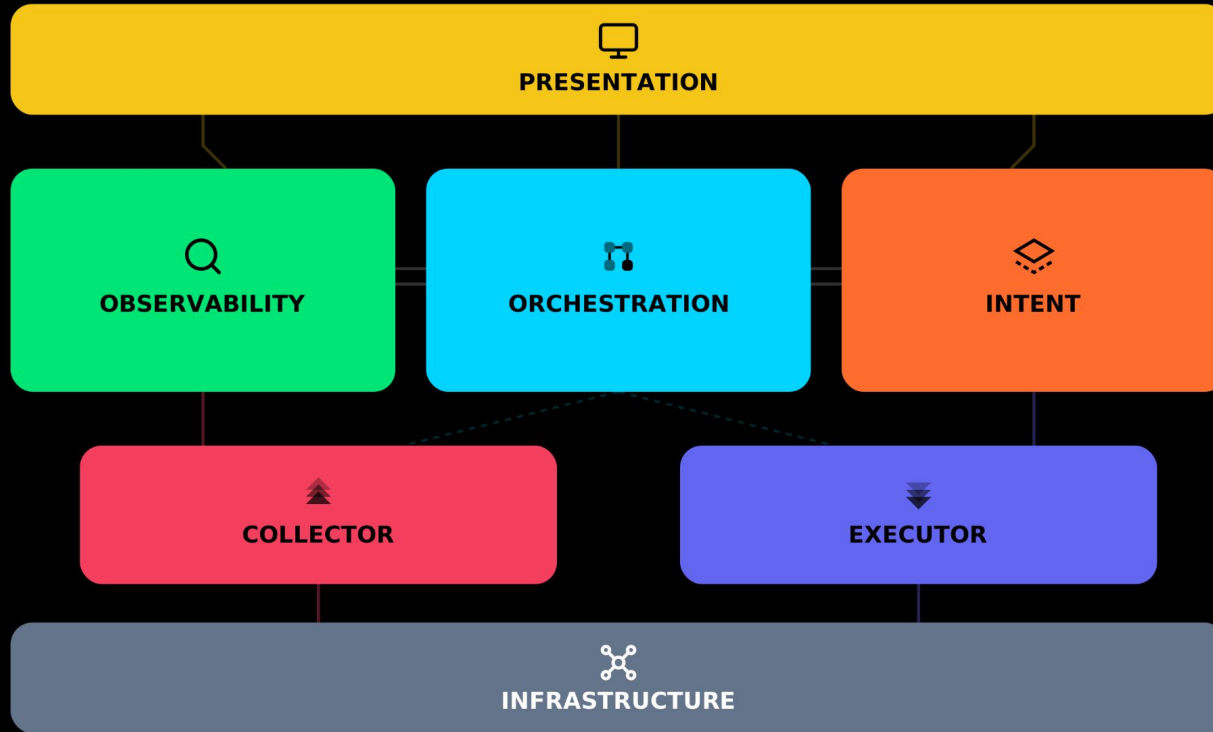


NA 
FRAMEWORK

Simple building blocks that
underpin any network
automation framework

Not a reference architecture
— it's a shared vocabulary +
responsibilities.

NAF Framework



NAF Framework Fundamentals

Functional Building Blocks

The NAF Framework is meant to be tool agnostic to avoid comparing apple and orange.

Each functional building block can be mapped to one or more tools and each tool can map to multiple functional blocks.

Living Document Not a Protocol

The NAF Framework is meant to evolve over time.

It's not a strict implementation that imposes a specific way of doing things and in many cases, not all components are required to get started.

Both for Beginners and Experts

The NAF Framework initiative is focusing initially on laying out the main building blocks to lay the foundation.

In the future the goal is to go deeper into each block.



A map to help navigate the terminology and understand how things are related to each other



Presentation



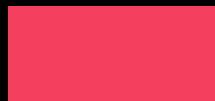
Observability



Orchestration



Intent



Collector

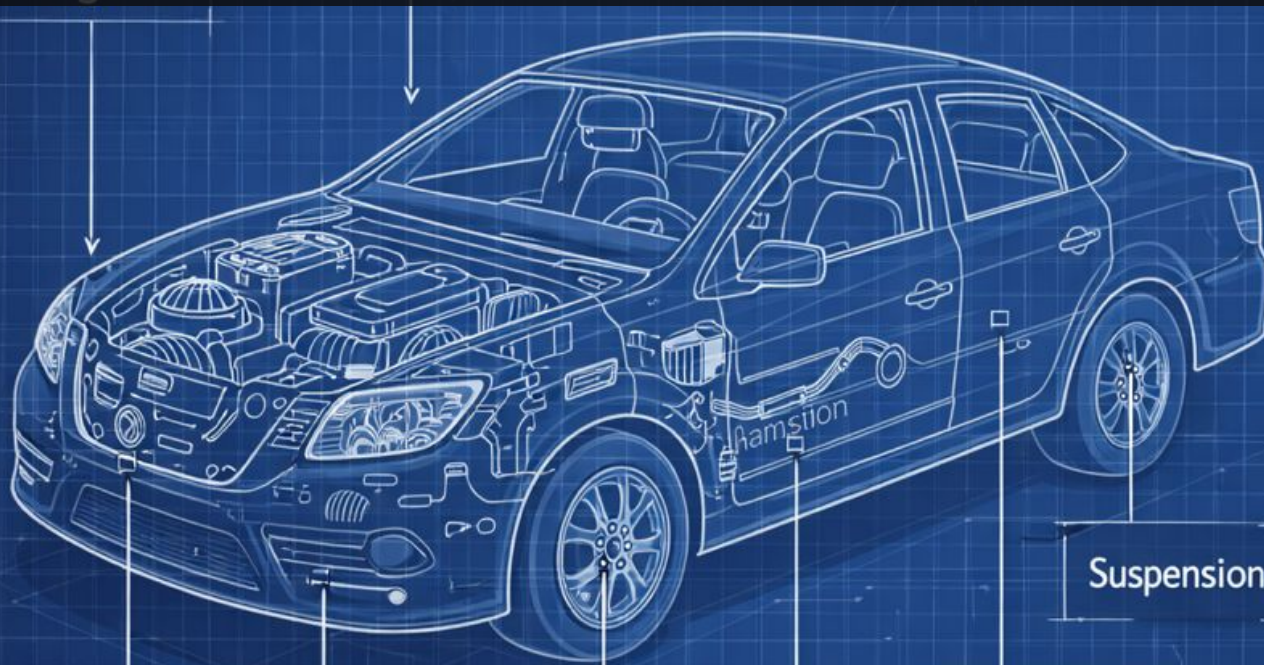


Executor



Infrastructure

A functional blueprint to help everyone understand and build Network Automation Stack (system thinking)



Battery

Chassis

Brakes

Suspension

Engine and transmission
that power the vehicle



Electrical System

Battery, wiring, and electronics



Safety

Airbags, seat belts, and crumple zones



Target Audience

Builders

Network Automation Engineer

Automation Architect

Software Engineer

Network Reliability Engineer

Users

Network Engineer

Network Ops

Network Architect

Engineering Manager





Wait, what about my tools ? Are they gone ?



Mapping between functional blocks and tools

	PRESENTATION	OBSERVABILITY	COLLECTOR
 gNMIc Telemetry collection	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 Prometheus Metrics + alerts	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 Grafana Compliance dashboard	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 SuzieQ State observability	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
 Kentik Flow observability	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Each tool / product / project can map
Completely or Partially
to
One or Multiple blocks



Where do I start?

Do I need to implement all blocks to get started?

No, you should only implement what is required for your use case(s)

Is there a specific order to get started?

No, it really depends on your use case(s).



Ryan Shaw



Dinesh Dutt



Christian Adell



Claudia de Luna



Srividya Iyer



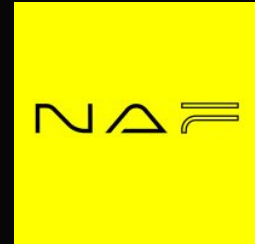
Wim Henderickx



Damien Garros



**The NAF
Community**



The Team behind the Working Group

Practical Example



Example- Roll out a new subnet across 50 branch sites

What we want to achieve, what the environment looks like, and the tools in play.

THE GOAL

Deploy a new subnet **10.100.0.0/24** to every branch, validate the config on **every device**, and automatically **detect and remediate** any site that drifts out of compliance.

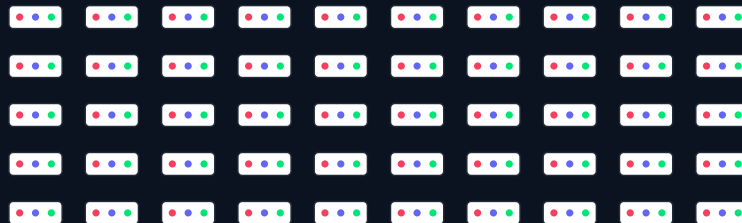
SUBNET	10.100.0.0/24	SITES	50 branches
DEVICES	1 router / site	VENDOR	Mixed (NETCONF)
VALIDATION	Interface + route table	SLO	< 5 min drift detect

THE ENVIRONMENT

HQ · DATA CENTER

Automation control plane

50 BRANCH SITES · 1 ROUTER EACH



S-01 ... S-50 · mixed-vendor, NETCONF-capable

OUR STACK · 8 TOOLS



Infrahub
INTENT · UI



AAP
ORCHESTRATION



Ansible
EXECUTOR



gNMIc
COLLECTOR



Prometheus
OBSERVABILITY



Grafana
PRESENTATION



SuzieQ
NET OBSERVABILITY



Kentik
FLOW OBSERVABILITY



Example - Mapping to the Framework

Rows are the 8 tools we run; columns are the 6 logic/transport blocks. Filled cell = tool covers that block. Infrastructure is excluded — it's the substrate, not a tool choice.

	PRESENTATION	OBSERVABILITY	ORCHESTRATION	INTENT	COLLECTOR	EXECUTOR
 Infrahub Source of truth + UI	●	●	●	●	●	●
 AAP Runbooks + web UI	●	●	●	●	●	●
 Ansible Config push (NETCONF)	●	●	●	●	●	●
 gNMIc Telemetry collection	●	●	●	●	●	●
 Prometheus Metrics + alerts	●	●	●	●	●	●
 Grafana Compliance dashboard	●	●	●	●	●	●
 SuzieQ State observability	●	●	●	●	●	●
 Kentik Flow observability	●	●	●	●	●	●

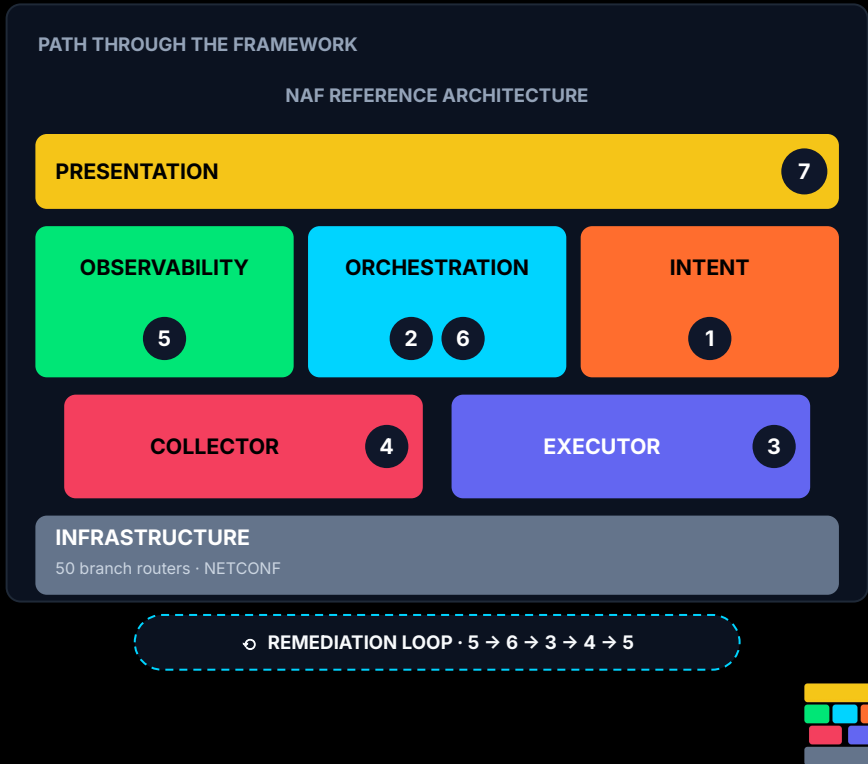
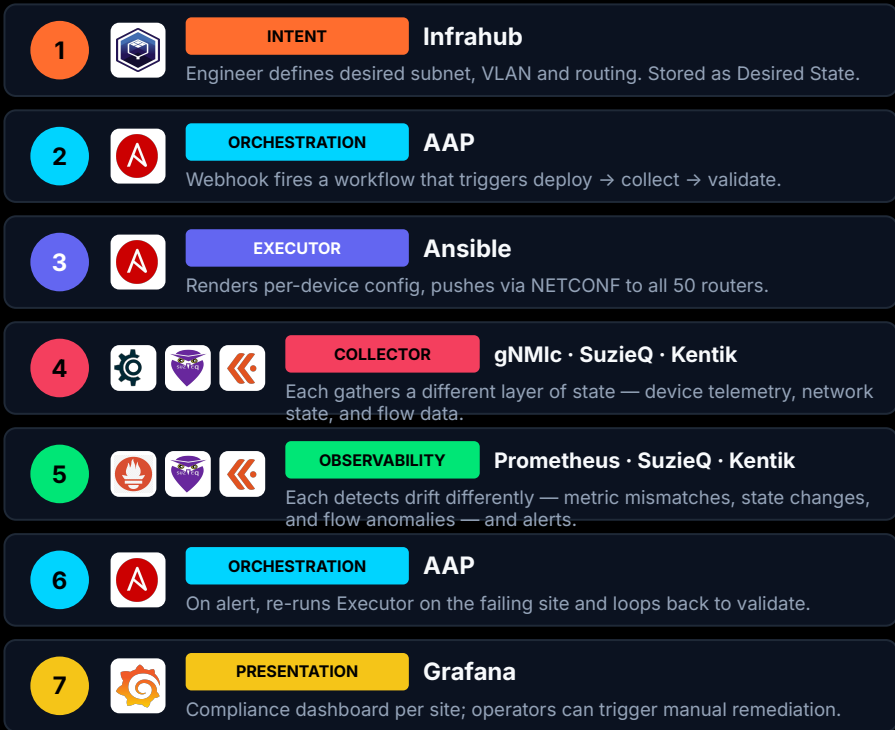
One tool can span multiple blocks. Infrahub = Intent + Presentation. AAP = Orchestration + Presentation. SuzieQ and Kentik each cover three blocks (Collector → Observability → Presentation) for different data types.

Multiple tools can share one block. Presentation is covered by 5 tools (Infrahub, AAP, Grafana, SuzieQ, Kentik). Collector is covered by 3 (gNMIc, SuzieQ, Kentik) — each gathering a different kind of state.



Example - Step by Step

The seven steps in reading order on the left; the path they trace through the NAF building blocks on the right. Each block shows the step numbers that touch it.



NAF Framework Deep Dive



What defines a functional building block

A **tool-agnostic function** in the automation system —

a

chunk of responsibilities with clear goals, inputs/outputs, and capabilities

Purpose / Goals

what outcome it exists to deliver

Capabilities

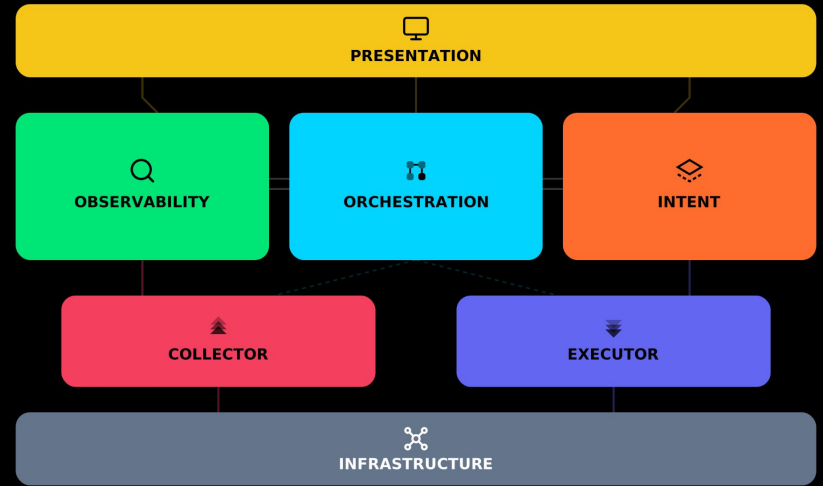
expressed as MUST / SHOULD / MAY (requirements, not products)

Boundaries

what it does and what it does not do (inputs → outputs)



Intent



The authoritative store of what your network should look like (config, topology, services, and operational expectations) expressed at business level and translated into device-level artifacts.

Intent - Goals

Capture everything you need

configs, assets, topology, services, IPs, circuits, and who owns what. Everything is data, describe it in a structured way

Keep the data clean and trustworthy, as it evolves

validate types, relationships, and uniqueness; track who changed what and when

Let operators think in business terms, not device syntax

abstract config artifacts details behind high-level intent

Let people work in parallel without stepping on each other

atomic change bundles, branching, and staging

Let people and machines access the data easily

REST/GraphQL APIs, web UI, CLI, etc. with fine-grained access control

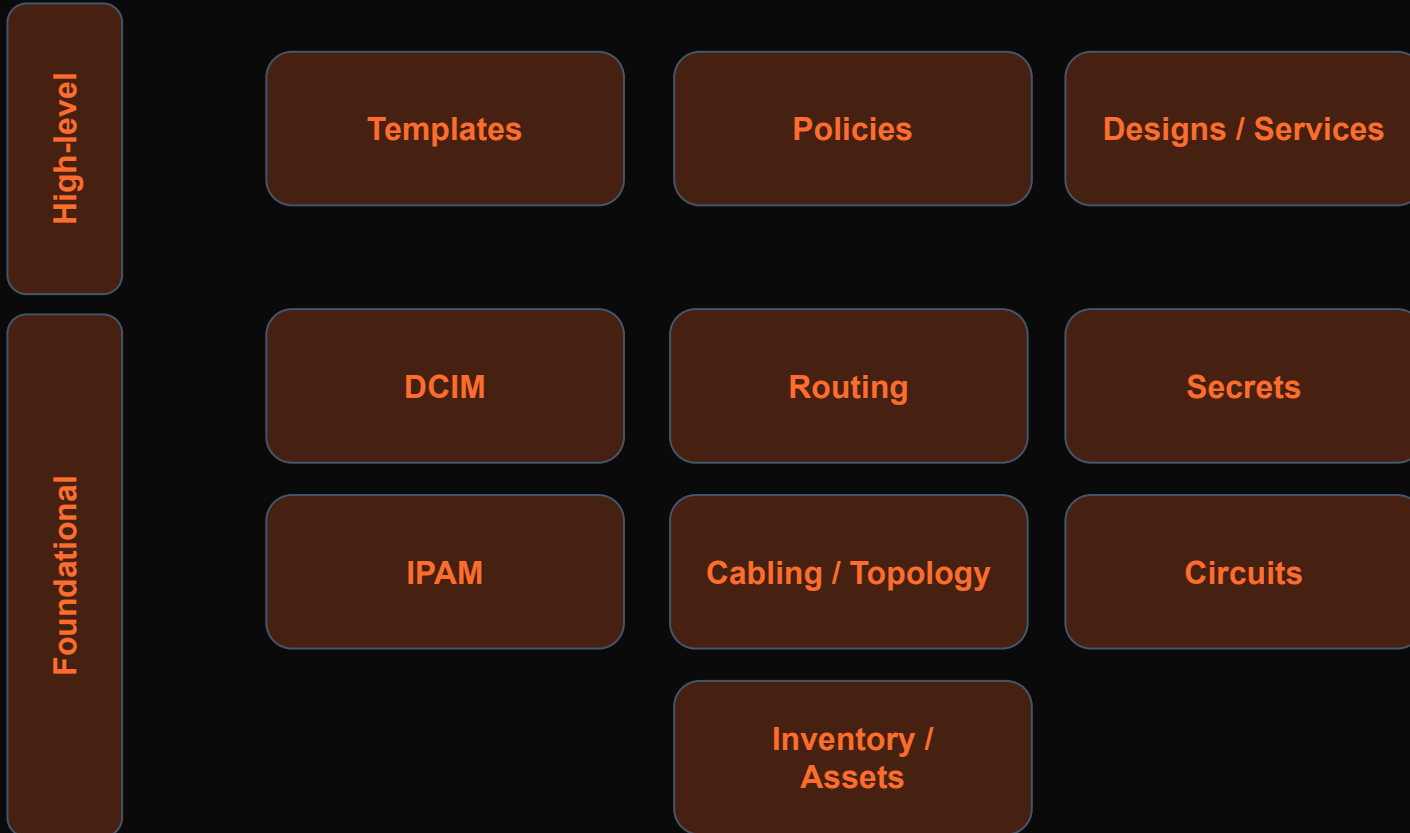
Bring data in from other systems

sync from CMDB, IPAM, asset management; define clear authority per domain

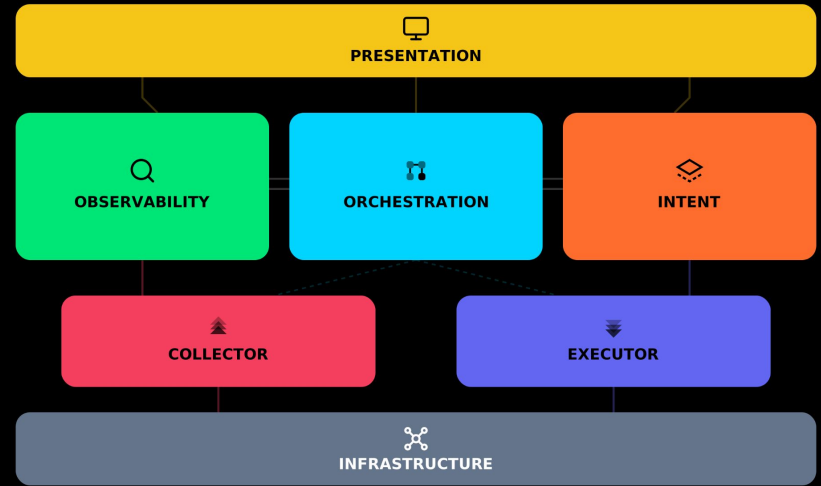


Intent - Data Modeling

What do you need to rebuild it
if the network was completely lost?



Executor



NMf Reference Architecture – Minimal

The block that directly interacts with network devices to apply changes, translating intended state into actual configuration pushes, operational commands, and verifiable outcomes.



Executor - Goals

Get the right data into place

pull inventory, credentials, and intended state from the Intent before executing.

Execute reliably at scale

parallelism, retries, rate limiting, dry-run, and partial-failure handling.

Start when needed, whether that's now or later

synchronous triggers (API, webhook) and asynchronous ones (scheduler, event)

Work with any network device or platform

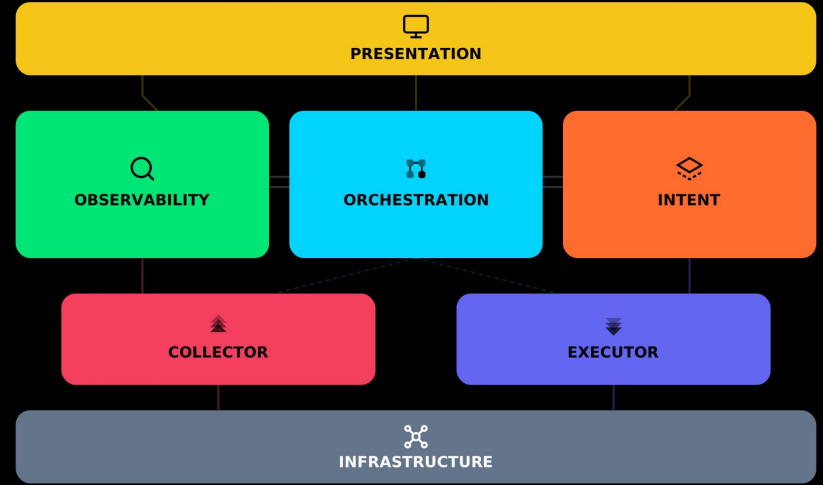
adapters for SSH/CLI, NETCONF, gNMI, REST APIs, and cloud platforms

Track state over time

record which devices succeeded, failed, or are pending; enable idempotency, rollback, and resume



Observability



NM Reference Architecture - Minimal

Collector

Observability stores the actual network state and defines the logic to process observed data. The Collector retrieves (reads) that state from devices using push or pull protocols.

Observability - Goals

Observe everything, automatically

New device in the SoT → monitoring starts.
Decommissioned → alerts stop. No manual lists, no stale inventories.

Handle heterogeneous environments

Network gear, cloud, containers, each speaks differently. Normalize into a common schema with near-real-time collection.

Correlate across layers

Connect metrics, logs, and flows from all planes.
Enrich with SoT context so symptoms map to root causes, not isolated signals.

Scale without degrading

Decouple the pipeline. Scope by domain. Use purpose-built storage. Design for a million metrics/sec from day one.

Detect and remediate automatically

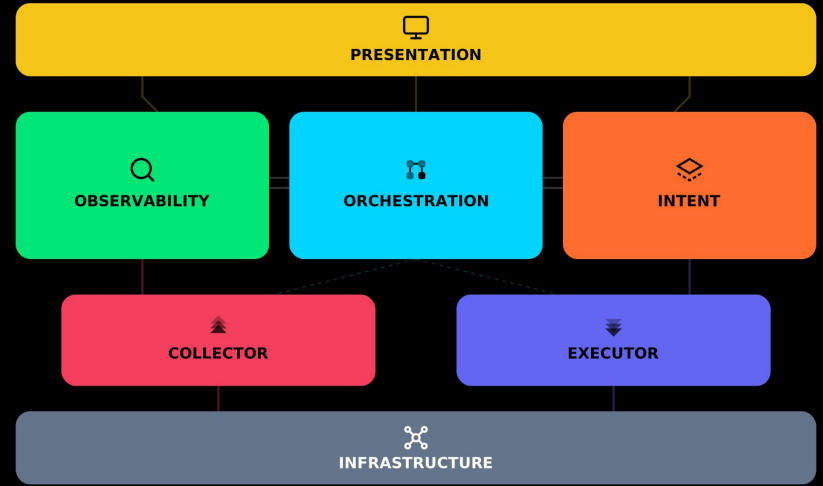
95% of alerts should trigger automation, not a human.
Dynamic baselines, correlated events, structured payloads the orchestrator can act on.

Give each audience the right view

Ops teams need real-time state. Engineers need query access to history. Business needs SLA summaries. One dataset, multiple lenses.



Orchestrator



NAP Reference Architecture - Minimal

The block that coordinates all the other blocks, deciding when each should act, sequencing multi-step workflows, handling failures gracefully, and providing tamper-evident visibility of every decision.



Orchestrator - Goals

Coordinate multi-block workflows end-to-end

hold the full sequence: validate SoT → pre-check → execute → validate → notify.

React to events automatically, with or without human initiation

Workflow approvals, Observability alerts, or scheduled scans.

Resilient and scalable execution

survive restarts, handle partial failures, complete reliably across hundreds of devices in parallel.

Provide tamper-evident visibility

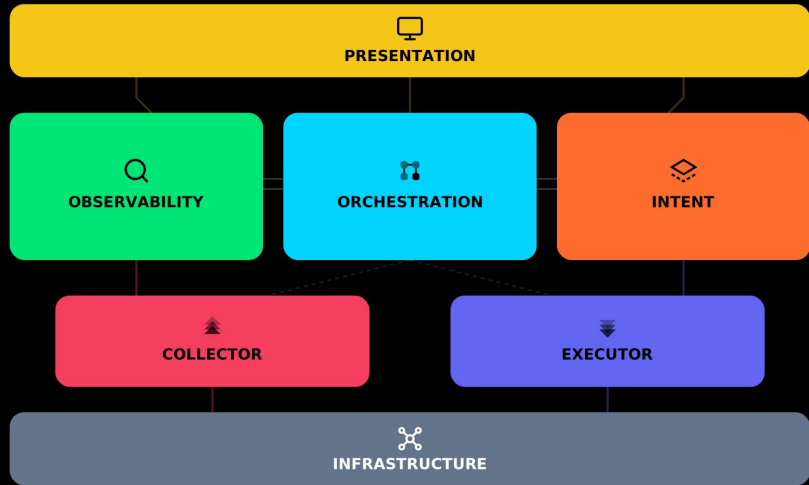
operators see what is running now; auditors see every decision with full traceability.

Enable safe automation through trust levels

auto-approve within guardrails; require human review only for genuinely risky changes.



Presentation



NAP Reference Architecture - Minimal

The outward-facing layer that makes the automation platform accessible to audiences who need not understand its internals: application teams, auditors, CI/CD pipelines, and external systems.



Presentation - Goals

Provide a stable, authenticated API with a consistent access model

versioned, access-controlled contract independent of internal changes.

Give every audience the interface they actually need

self-service portal, CLI for engineers, webhooks for CI/CD, dashboards for ops.

Ensure every action is attributable and auditable

no anonymous writes, no unlogged changes; full traceability from request to outcome.



Additional Resources



Additional Resources

Designing Network Automation at Scale

A comprehensive guide to designing scalable, reliable, and maintainable network automation systems

<https://designingnetworkautomation.com/>
By Christian Adell

NAF Network Automation Solution Wizard

This application helps you apply the Network Automation Forum's NAF to your automation projects.

<https://naf-naf-wizard.streamlit.app/>
By Claudia de Luna



Summary

- Not a standard, it's a guide
- You don't need everything to get started
- Think Automation as a system
- Focus on functional first, tools later
- Each tool can map to multiple building blocks
- Don't reinvent the wheel
- Map your existing workflow toward the framework
- Give us some honest feedback good and bad



How can you help

Use case document Github:

<https://github.com/Network-Automation-Forum/reference/blob/main/docs/Framework/Framework.md>

Tool mapping

Expand details per building block

NAF Slack channel:

<https://networkautomationfrm.slack.com/archives/C082G6VGZHP>



Questions ?



Thank You

