

The Network Automation Map: Navigate Your Journey with the NAF Automation Framework

Wim Henderickx, Nokia
Damien Garros, OpsMill



About Us

Wim Henderickx



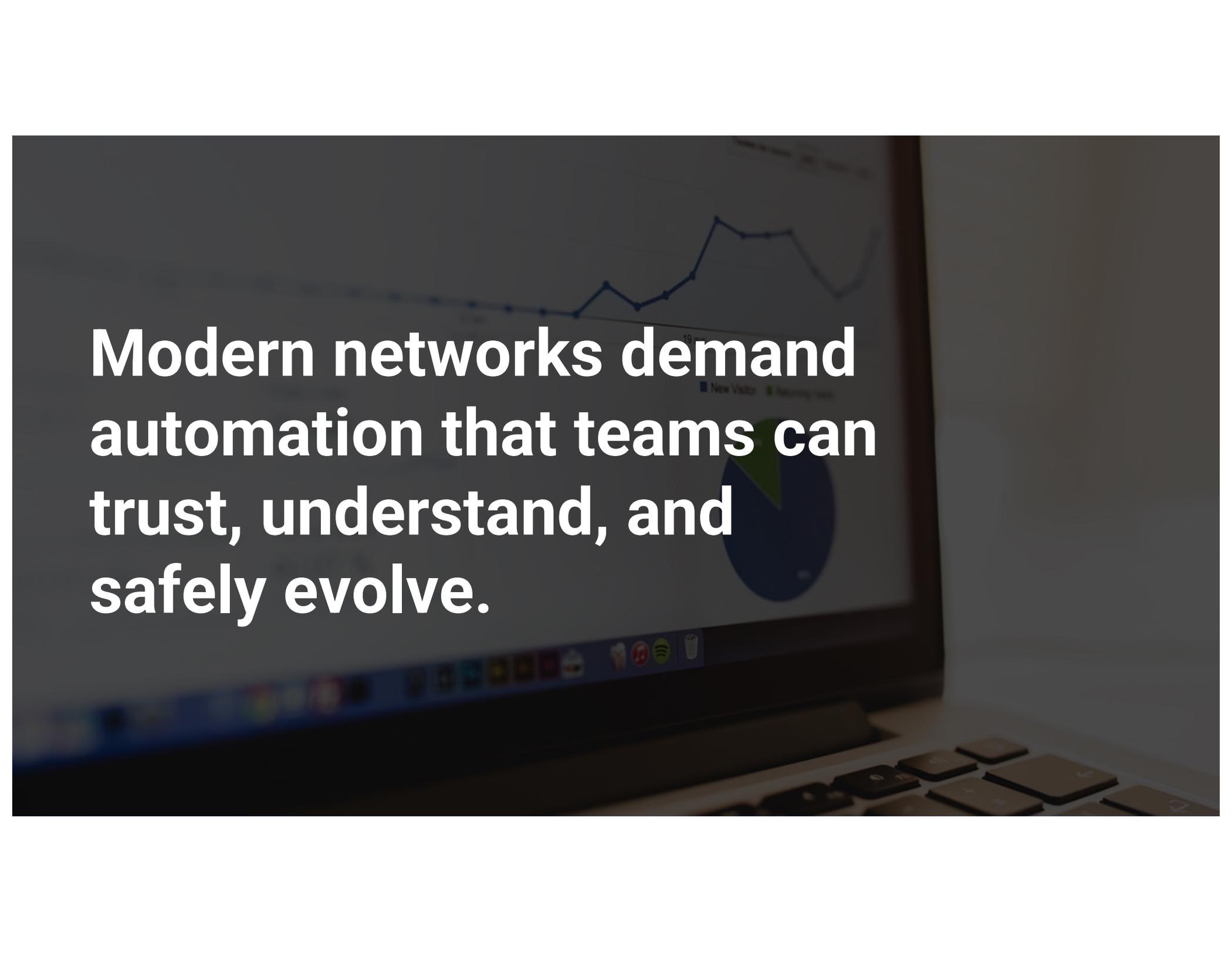
**CTO IP Division
Nokia**

Damien Garros



**Co-founder and CEO
OpsMill**

Introduction

A laptop screen is shown in a dark, dimly lit environment. The screen displays a data visualization interface. At the top, there is a line graph with a blue line showing an upward trend. Below the graph, a pie chart is visible, with a legend that includes the text 'New Visitor'. The bottom of the screen shows a taskbar with several application icons. Overlaid on the screen in large, white, bold, sans-serif font is the text: 'Modern networks demand automation that teams can trust, understand, and safely evolve.'

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.



SALT



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 कवाला हैं!

On ne peut
pas faire ça!

هذا لا يعمل

✂ s 75 ✂ たい!

No es correcto!

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

خطن امام ب...



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



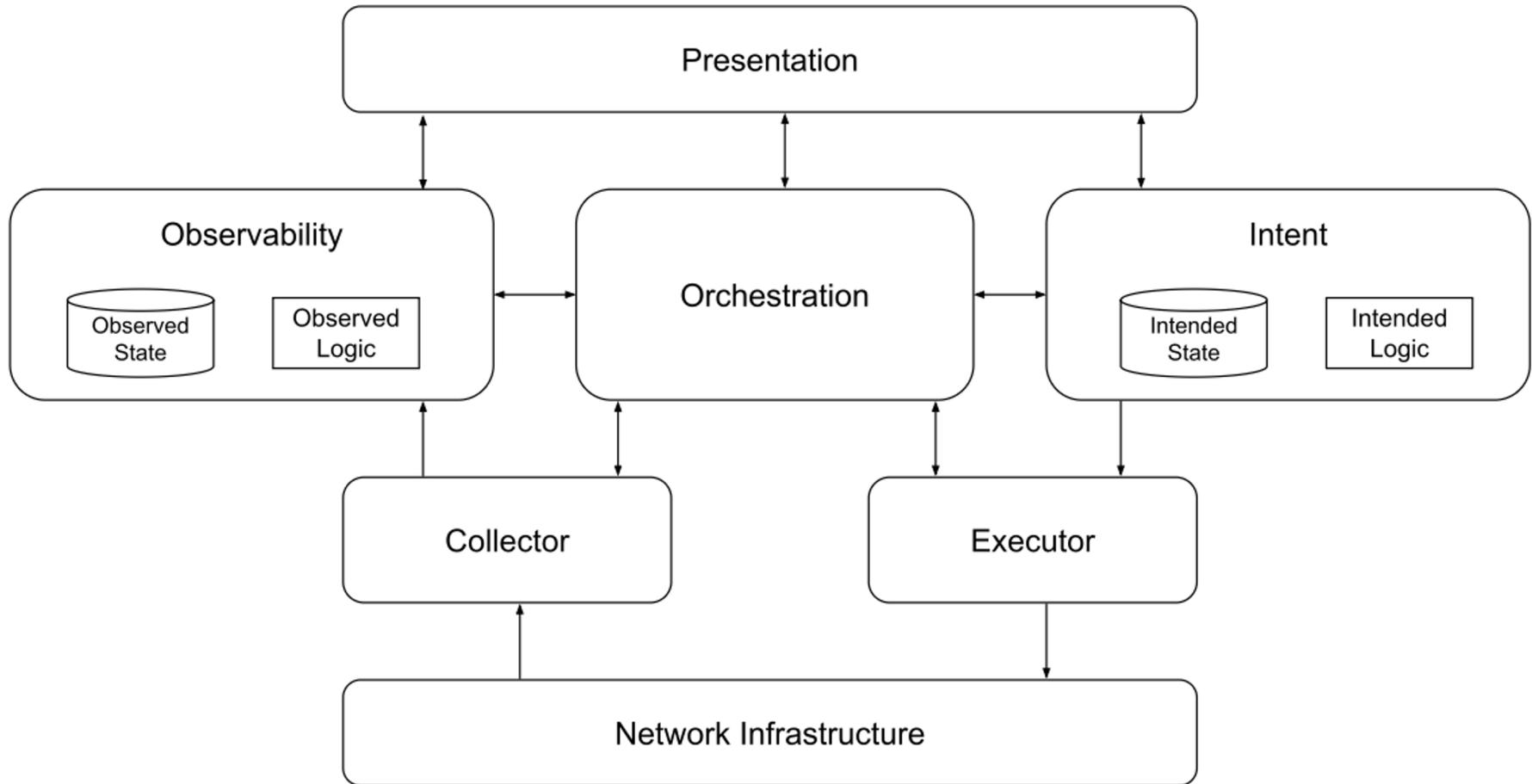
NAF Automation Framework



Simple building blocks
that underpin any network
automation framework

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

NAF Automation Framework



NAF Automation Framework Fundamentals

Functional Building blocks

The NAF automation framework is meant to be tools agnostic to avoid comparing apple and orange.

Each functional building block can be map to one or multiple tool and each tool can map to multiple functional blocks

Living Document Not a protocol

The NAF automation framework is meant to evolve over time.

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

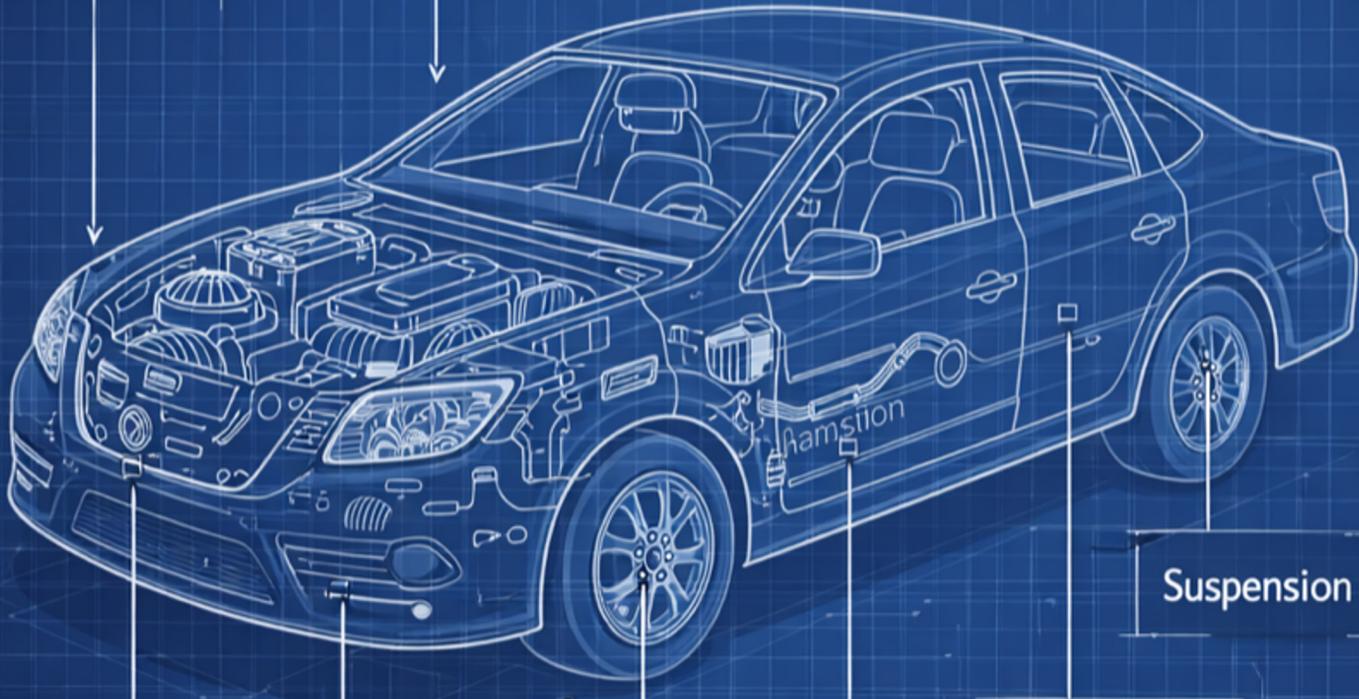
Both for the beginners and the experts

The NAF automation 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



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



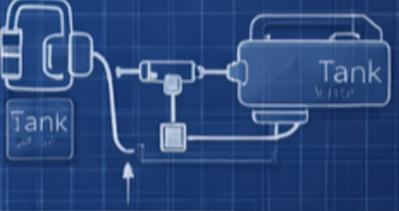
Engine and transmission that power the vehicle

Electrical System

Battery, wiring, and electronics

Safety

Airbags, seat belts, and crumple zones



Battery

Chassis

Brakes

Suspension

Target Audience

Builders

Network Automation Engineer

Automation Architect

Software Engineer

Network Reliability Engineer

Users

Network Engineer

Network Ops

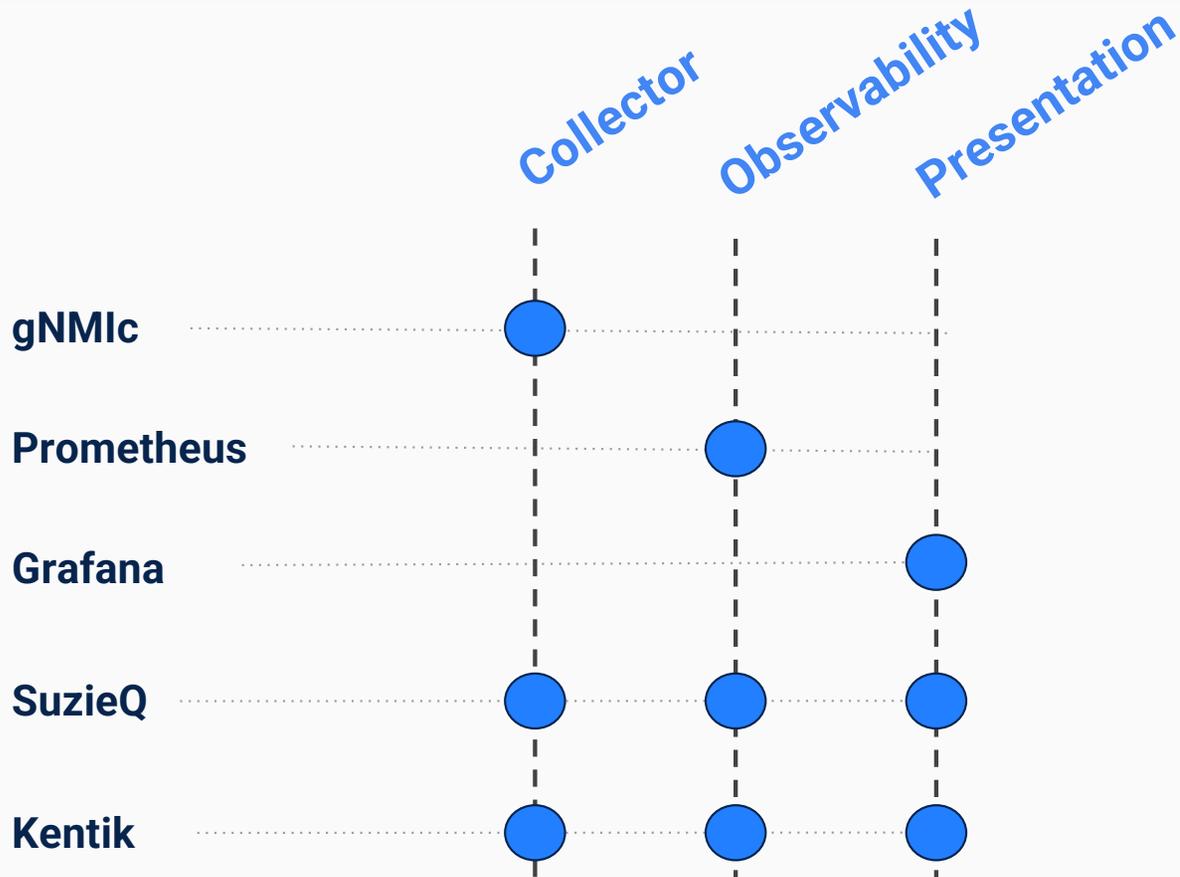
Network Architect

Engineering Manager



Wait, what about my tools ? Are they gone ?

Mapping between functional blocks and tools



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

Where do I start ?

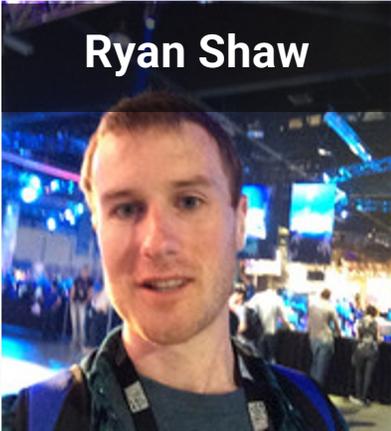
Do I need to implement all blocks to get started ?

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



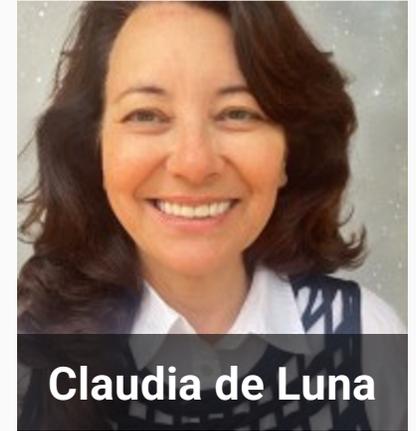
Dinesh Dutt



Christian Adell



Claudia de Luna



Srividya Iyer



Wim Henderickx



Damien Garros



**The NAF
Community**



The Team behind the Working Group

Main functional Building blocks

What defines a functional building block

A **tool-agnostic function** in the automation system —
a **chunk of responsibility** with clear goals, inputs/outputs, and capabilities

Purpose / Goals

what outcome it exists to deliver

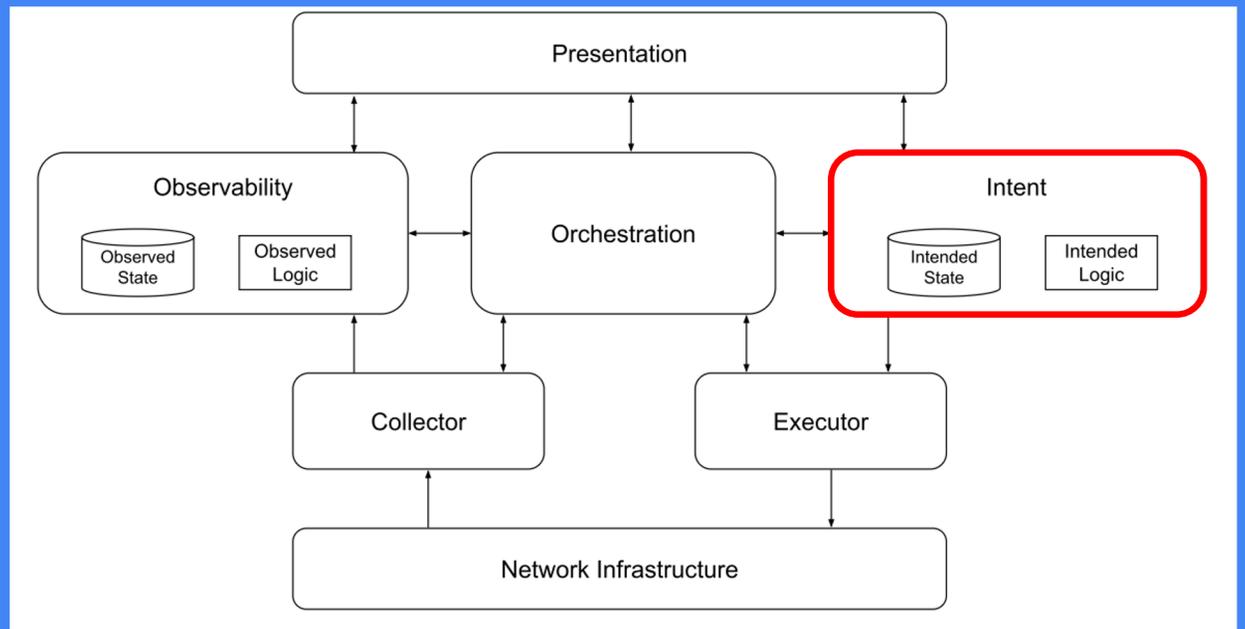
Capabilities

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

Boundaries

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

Intent



Intent

Defines the logic to handle and the persistence layer to store the desired state of the network, including both configuration and operational expectations.

Source of Truth



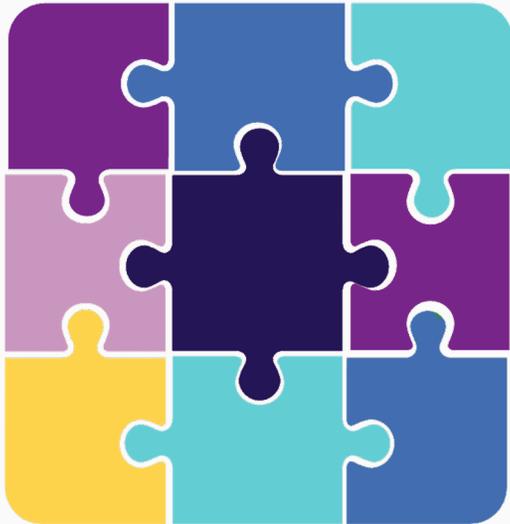
Source of Truth has been the root cause of many misunderstandings within our industry.

Intent



Intent is meant to provide a more accurate definition

Intent - Introduction



Inventory

**Cabling /
Topology**

DCIM

Circuit

IPAM

Roles/Statuses

**Configuration
Templates**

**Routing
Information**

Services

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

Why do we need the Intent



Declarative Vs Imperative

```
configure terminal
interface GigabitEthernet0/1
switchport access vlan 10
exit
Exit
write memory
```

Imperative - HOW

- Manually describe the step-by-step recipe.
- If something goes wrong halfway, state may be inconsistent.

```
interface:
  name: GigabitEthernet0/1
  vlan: 10
```

Declarative - WHAT

- You describe the desired end state, not how to get there.
- Easier to make idempotent and retry safely.
- Vendor neutral

Differences between Cloud and Networking

	Cloud	Networking
Declarative		   
Imperative	Optional	Optional

Intent - Requirements

MUST be capable of representing, in a structured form, any network-related aspect.

MUST support create, read, update, and delete operations

MUST be exposed through a standardized, well-documented API

SHOULD provide a consistent and unified view of the desired state

SHOULD use a neutral representation that will be derived into vendor-specific configuration artifacts

SHOULD include metadata that supports effective data governance

SHOULD be transactional, and provide a versioned access to data.

MAY include all the logic related to intended state management

Version Control

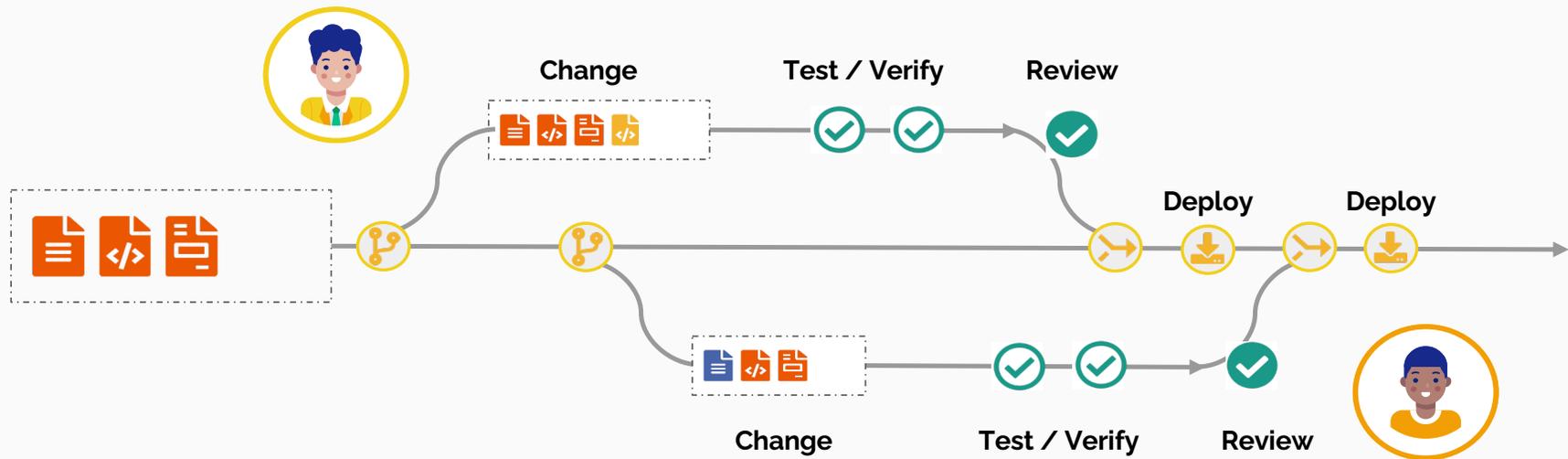
SHOULD be transactional, and provide a versioned access to data.

Version control allows changes to be:

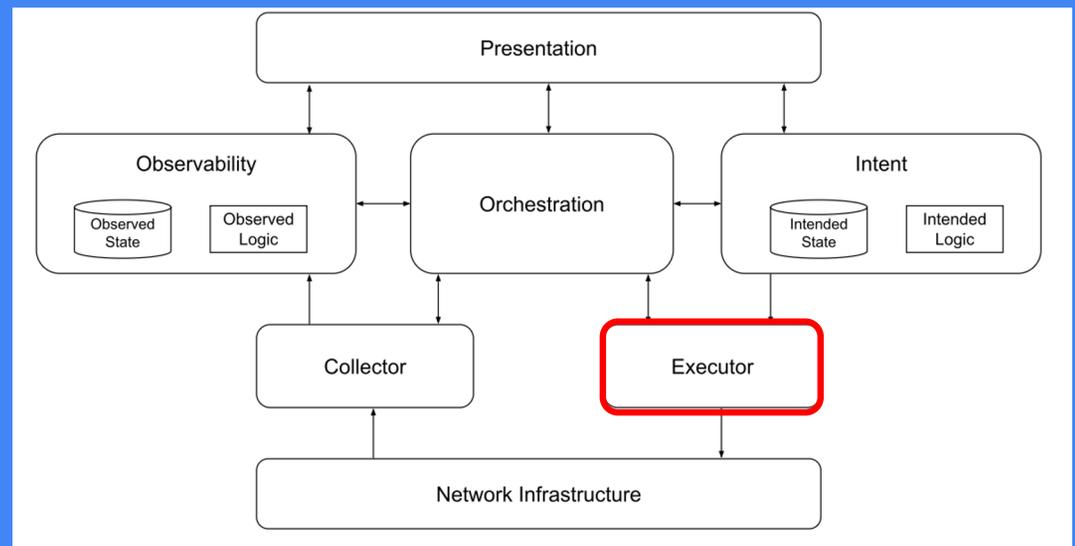
- Prepared in isolation
- Safely validated
- Reviewed

and only then integrated into the main automation environment.

Version Control Workflow



Executor



Executor

Encompasses the actual tasks applied to the network to drive changes (e.g., updating configuration) as guided by the intended state.

Executor - Requirements

MUST be capable of interacting with any of the supported network write interfaces

SHOULD provide a dry-run operation to check the expected result of the execution

SHOULD come from the intended state or be derived from it

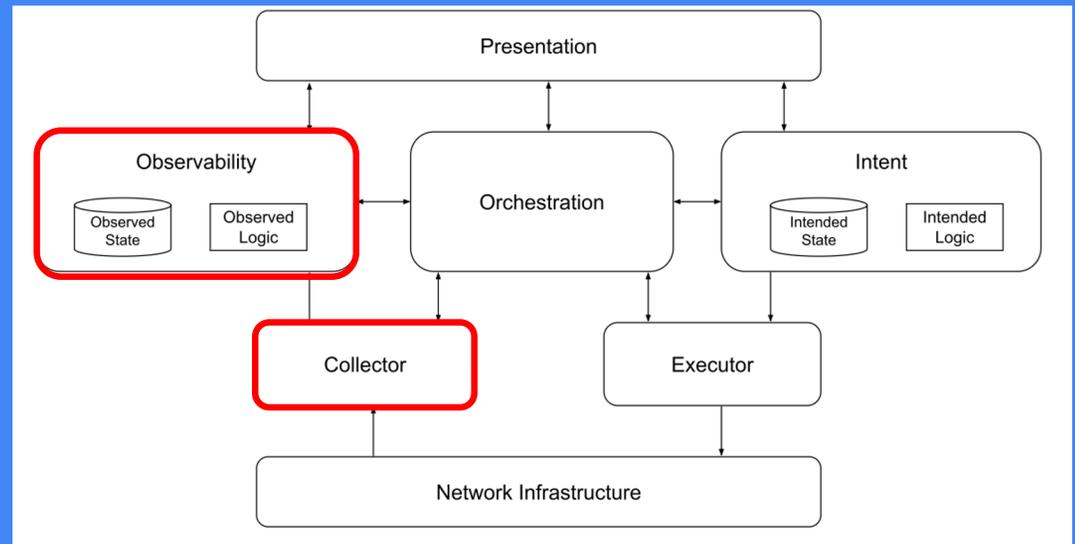
SHOULD support transactional execution of the changes

SHOULD support any network operation that alters the network state

MAY support both imperative and declarative approaches

Observability

Collector



Observability

Persists the actual network state, and defines the logic to process the observed data.

Collector

Focuses on retrieving (i.e., reading) the actual state of the network.

Observability Sources

Logs

Metrics

Traces

Flows

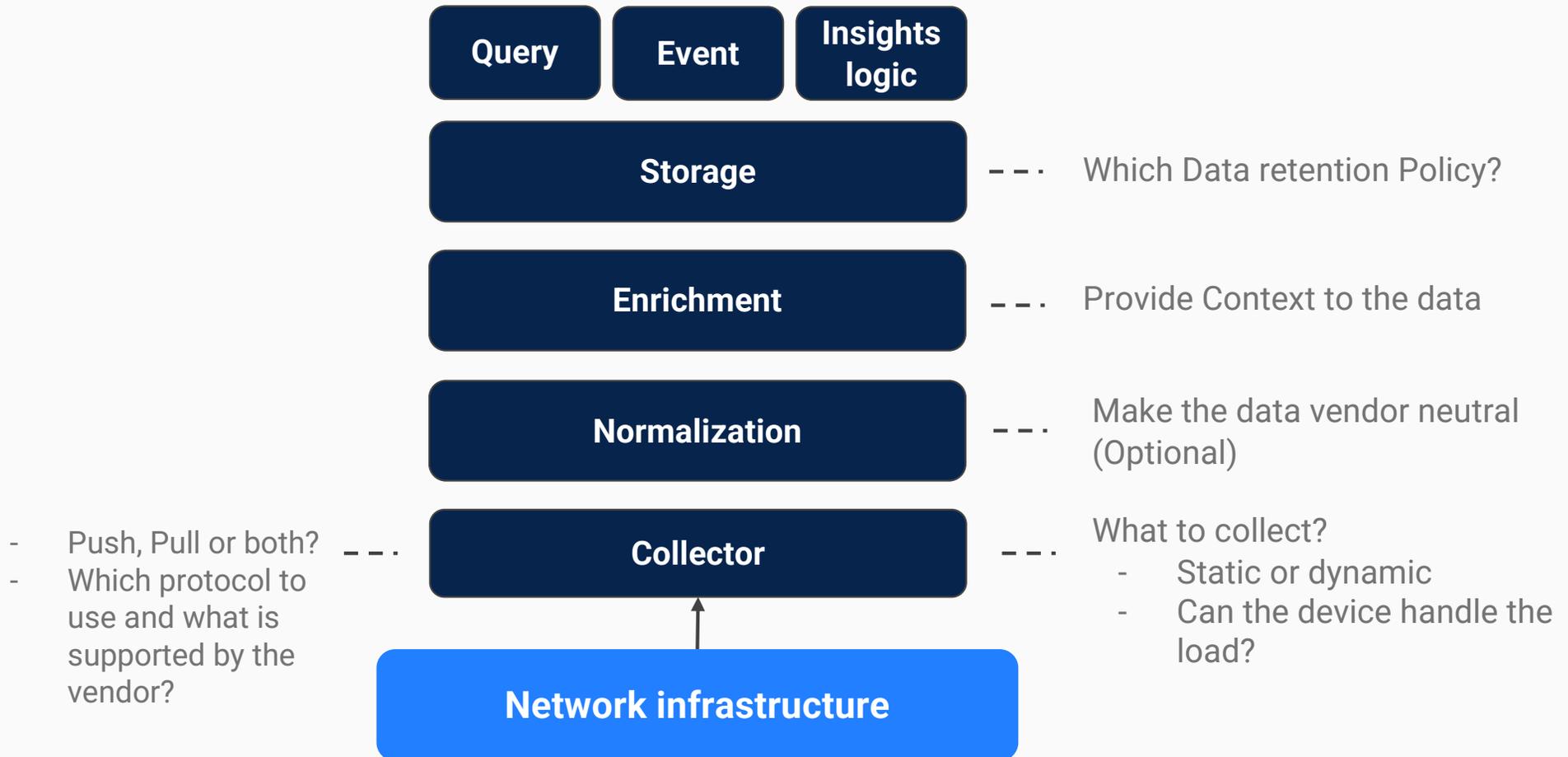
**Packet
Captures**

State

Config

Probes

Observability Stack: some questions to ask



Observability - Requirements

MUST support historical data persistence

MUST offer programmatic access to this data

SHOULD offer a capable query language to extract the data.

SHOULD expose relevant insights into the current network state

SHOULD automatically generate events when discrepancies are detected

data SHOULD be normalized into vendor agnostic data model

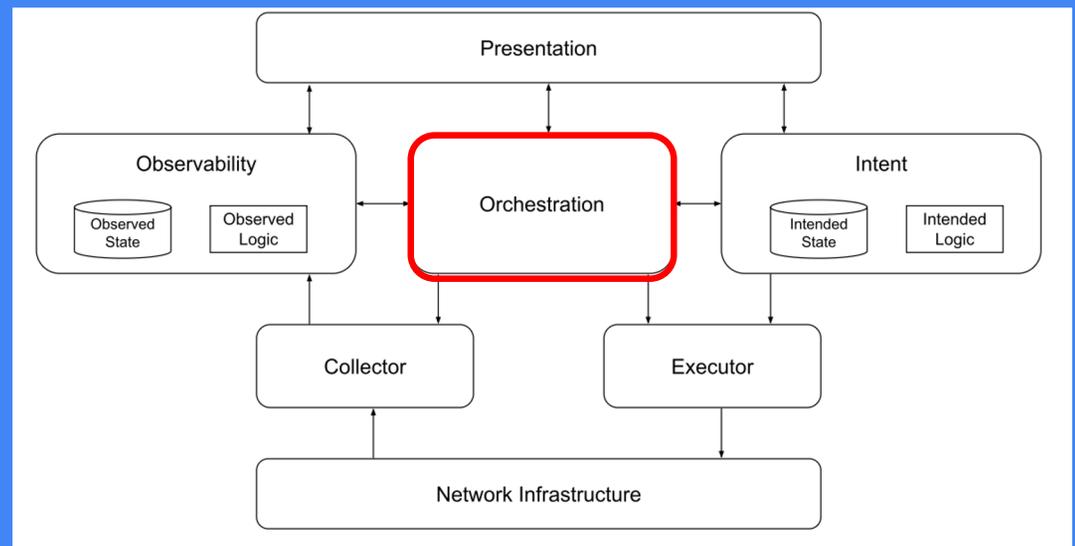
events MAY be processed by humans or connected to the Orchestration

MAY be enriched with contextual information from the intended state

Collector - Requirements

MUST include capabilities for retrieving live data from the network using read interfaces (push, pull)

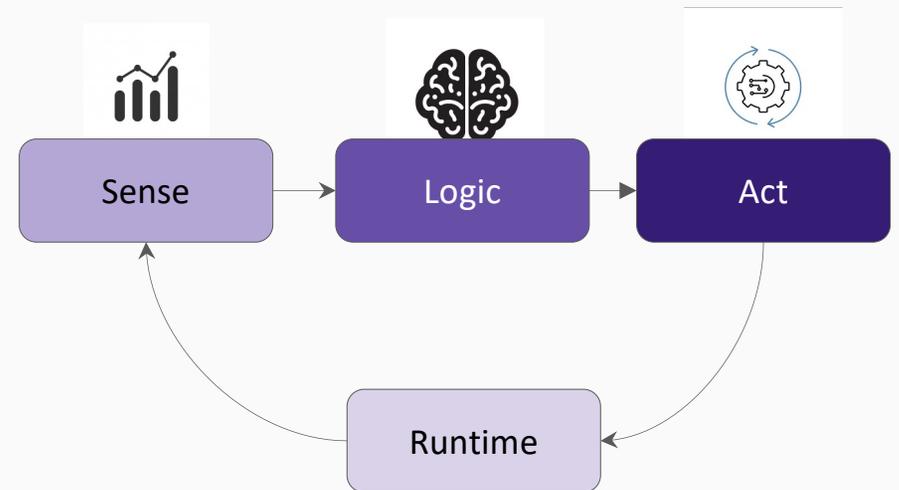
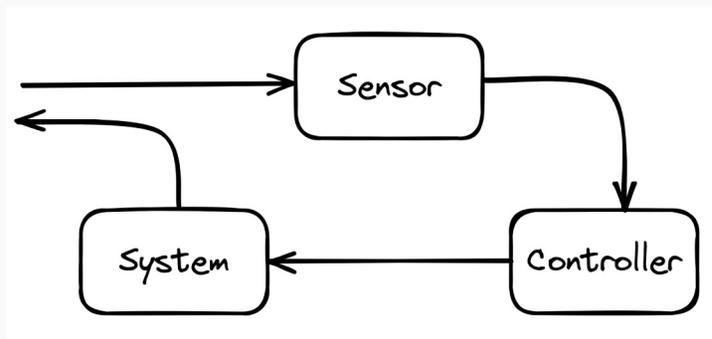
Orchestrator



Orchestrator

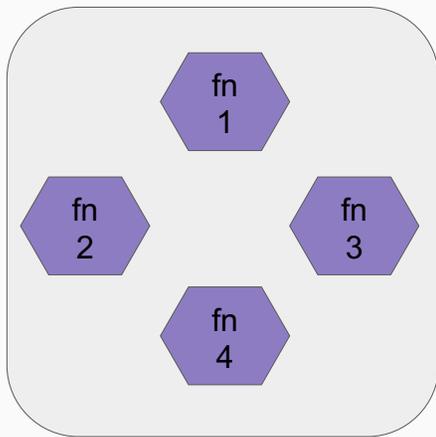
Coordinates activities across various building blocks

Event Driven Control Loop: Sense - Logic (Reason) - Act



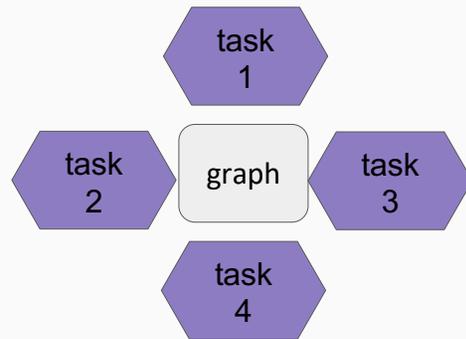
Orchestration implementation approaches

Monolith



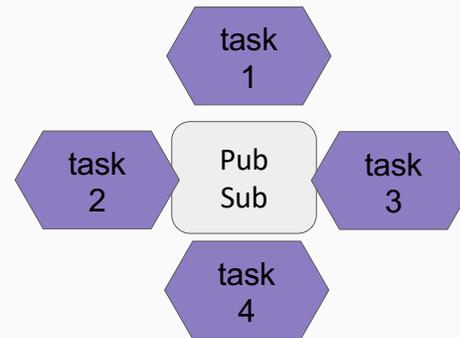
E.g. a python program

Workflow



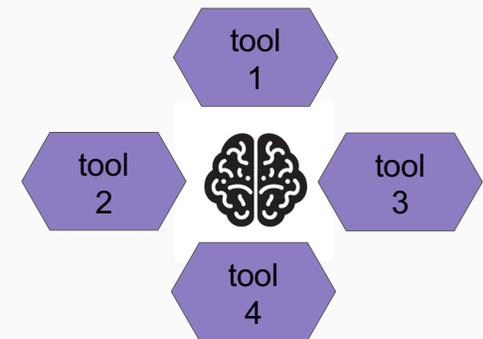
- Ansible (predefined workflow)

Choreography



- Event driven and distributed

Agentic



- Agentic

Gaining Trust: Dry Run - Rollback - Diff

DRY-RUN

Try the workflow without applying it to the network

Diff:

- What changed?
- Review the changes before applying to the network

BEFORE

AFTER

Rollback:

- Something bad happened, move back to a well known good state

Orchestrator - Requirements

MUST enable to coordinate processes across the various building blocks

Process execution SHOULD follow an event-driven approach

Execution logic SHOULD allow for reverse/compensating actions

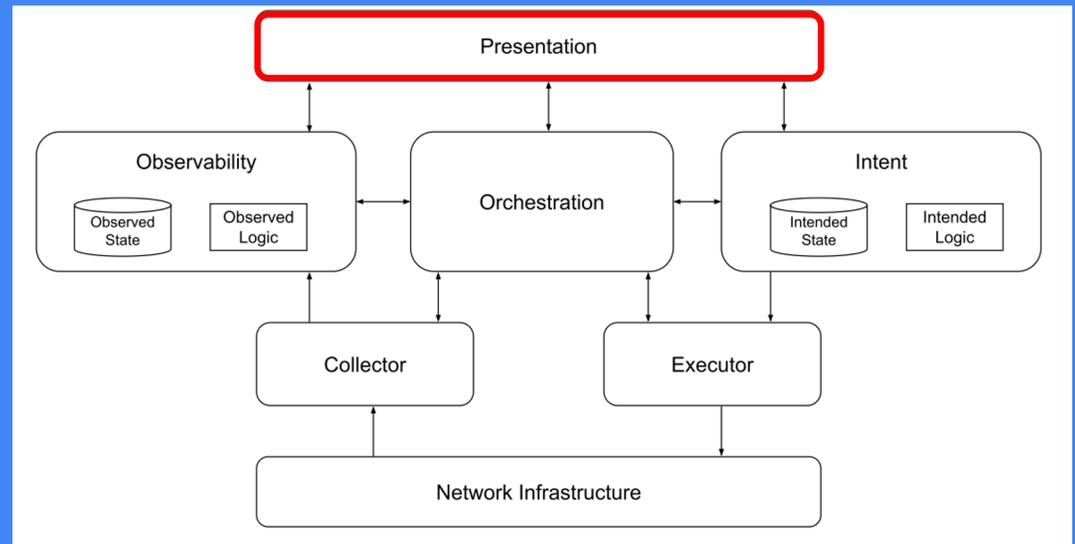
SHOULD provide the ability to schedule the execution of a workflow

SHOULD provide a dry-run operation to check the expected result of the workflow

SHOULD provide logging and traceability of past and current workflows

MAY include logic to correlate multiple events, infer relationships

Presentation



Presentation

Provides the interfaces through which users interact with the system, including dashboards, graphical user interfaces

Presentation

It's important to separate the presentation layer to avoid building workflows that are specific to one tool / interface

It is designed to interface with any of the other building blocks as required, serving as the primary point of contact between humans and the automation system, but this does not imply the need for a single pane of glass.

Presentation - Requirements

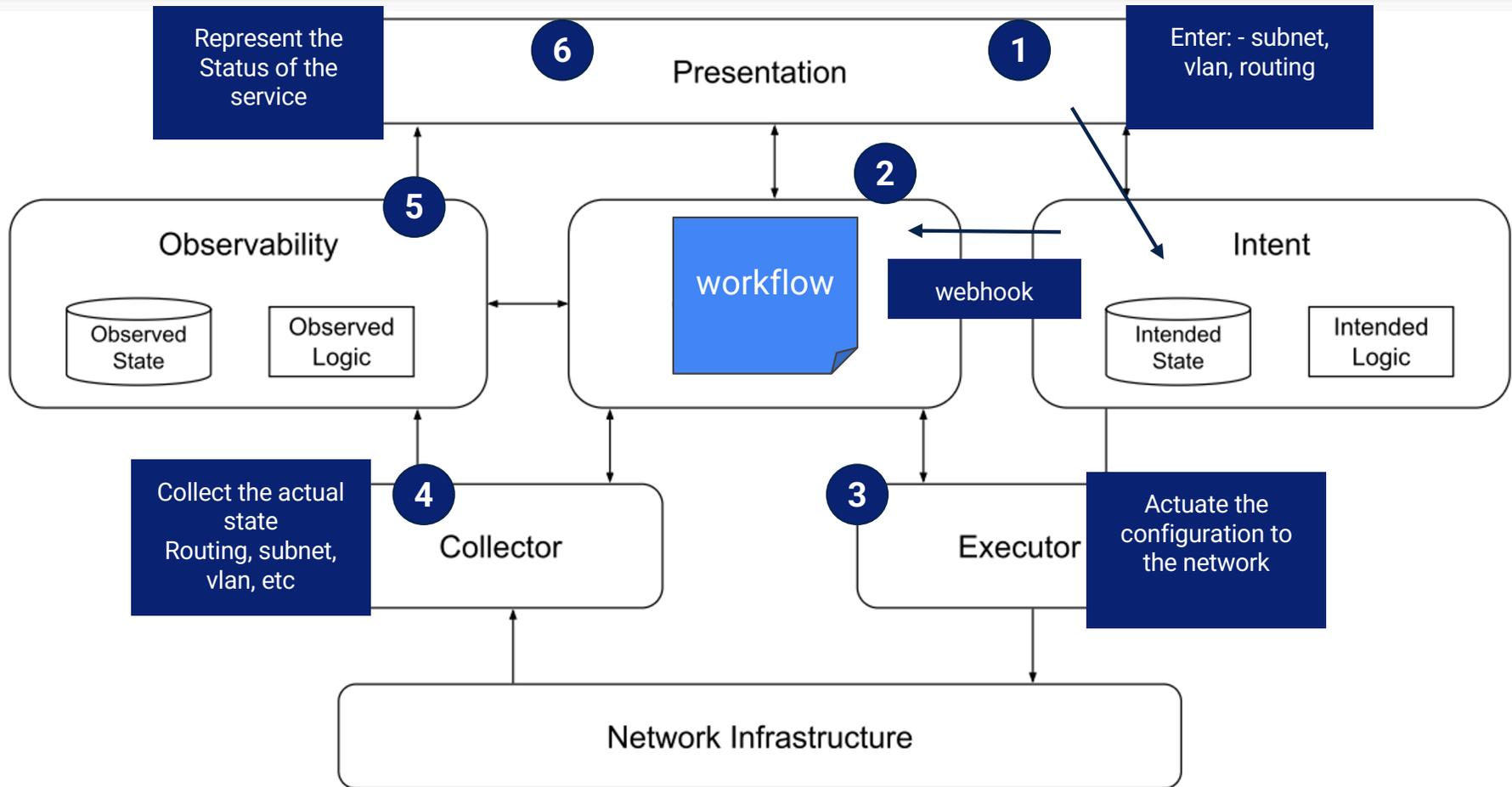
MUST provide robust authentication and authorization capabilities.

MAY take various forms depending on the needs of the end user

MAY support both read and write interactions

Practical Example

Use case: Rollout subnet 10.100.0.0/24 across 50 sites



Additional Resources

Additional Resources

The image shows the cover and table of contents for the book "Designing Network Automation at Scale" by Christian Adell. The cover features a blue header with the title "Designing Network Automation at Scale" and a subtitle "A comprehensive guide to designing scalable, reliable, and maintainable network automation systems". A yellow disclaimer box states: "Disclaimer: This book is a work in progress. The content, examples, and recommendations are continuously evolving as new insights emerge. Feedback and contributions are warmly welcomed to help refine and expand the material. The author plans to publish new chapters regularly and may later release the complete work in additional formats." Below the disclaimer is a blue button labeled "Table of Contents". The table of contents on the left lists the following parts and chapters:

- Preface
- Part 1: Rethinking Networking with Automation
 - 01 - The Automation Imperative
 - 02 - Design Principles
 - 03 - Architectural Thinking
- Part 2: Architectural Building Blocks
 - 04 - Source of Truth
 - 05 - Execution
 - 06 - Observability
 - 07 - Orchestration
 - 08 - Presentation
 - 09 - The Network
- Part 3: Designing for Scale and Reliability
 - 10 - Platform Engineering and CI/CD
 - 11 - Scaling and Reliability
 - 12 - Security, Governance and Compliance
- Part 4: Human and Organizational Dimension
 - 13 - Cultural Shift
 - 14 - Automation as a Product
- Part 5: Patterns and Use Cases
 - 15 - Closed Loop Automation

<https://designingnetworkautomation.com/>

By Christian Adell

The image shows the interface for the "Network Automation Forum (NAF) Network Automation Framework (NAF) Solution Wizard". The title is "Network Automation Forum (NAF) Network Automation Framework (NAF) Solution Wizard". Below the title, it states: "This application helps you apply the Network Automation Forum's Network Automation Framework (NAF) to your automation projects. Use the Solution Wizard to describe how you plan on designing your automation solution: streamlitApp". The main heading is "Design Your Automation Solution" with a double-headed arrow icon. Below this, it says: "The Solution Wizard guides you through each NAF component as well as additional considerations for your automation solution:". A list of components follows:

- **Initiative:** Define the problem, scope, expected use, and deployment strategy
- **Stakeholders:** Identify who is supporting the project
- **My Role:** Specify your skills and development approach
- **Dependencies:** List required infrastructure and systems
- **Timeline:** Plan staffing, milestones, and delivery schedule

NAF Components:

- **Presentation:** Define user types, interaction modes, and presentation tools
- **Intent:** Specify development approaches and provided formats
- **Observability:** Plan monitoring, go/no-go criteria, and tools
- **Orchestration:** Design workflow automation
- **Collector:** Plan data collection methods and tools
- **Executor:** Define execution methods

The wizard generates a complete solution design document (JSON + Markdown + timeline) that you can share with:

- Team members who will design or build the automation
- Stakeholders who need to understand what the automation will do

<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>

An aerial photograph of New York City at dusk. The sky is a mix of dark blue and purple, with some clouds. The city lights are visible, and the Empire State Building is prominent in the center, illuminated with red and green lights. The word "Questions?" is written in large, white, sans-serif font across the middle of the image.

Questions ?