NAF
NETWORK AUTOMATION FORUM

The Intelligence Layer

How AI will reshape Network Automation

AUTOCON 5

THE NETWORK AUTOMATION CONFERENCE

THESIS

We have not had an intelligence layer to harness network engineer knowledge.

The industry will embrace AI & Agentic concepts across most activities to reach a more robust Operating Model for Infrastructure.

Let’s start with Automation Success

Typically, we reference Hyperscalers:

- Highly standardized
- Replacement lifecycle of 5 years
- Linux-based OS focus
- Software development focus

Mostly DataCenters at scale
Always new gear
Apply compute style tooling
Software Engineering teams

How many of us get to work in this environment?

Our Attempts to Solve This Problem

Standards

NETCONF, gNMI, etc.







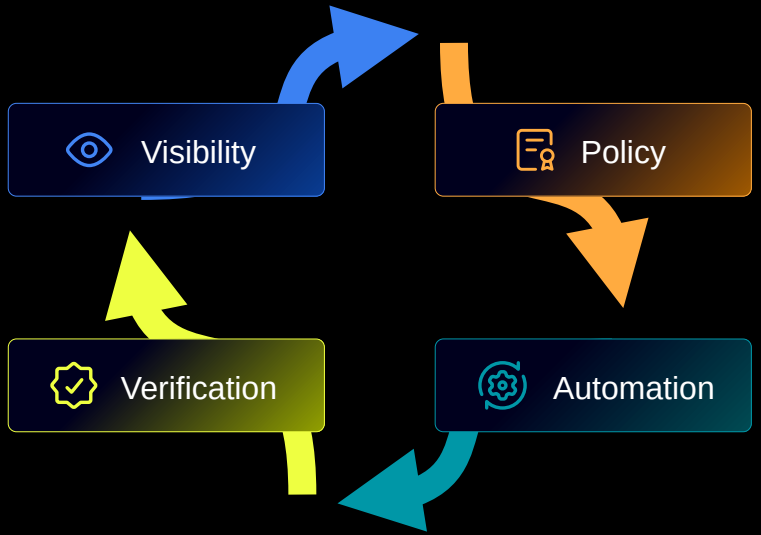




Our Attempts to Solve This Problem

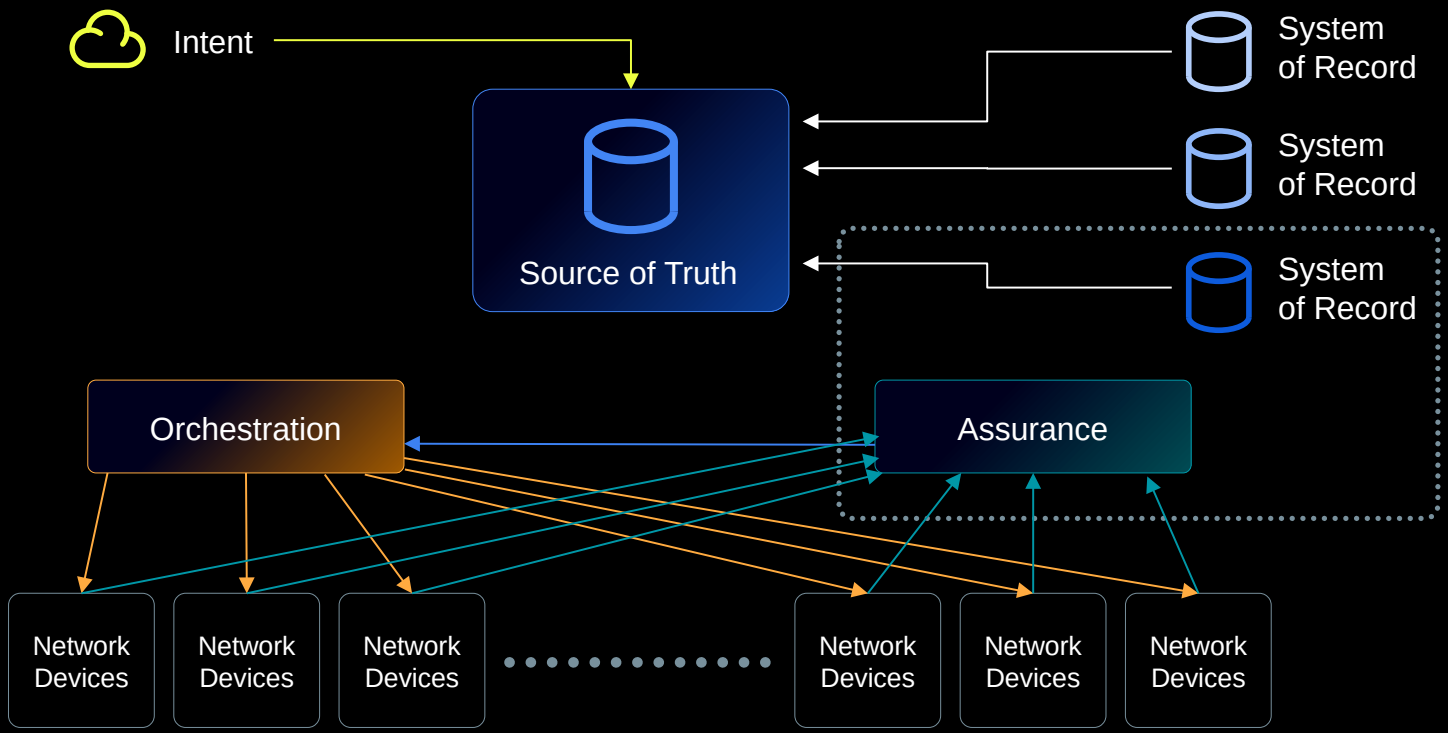
Intent Based Networking Framework

Continuous Alignment of the Network to the Business Intent



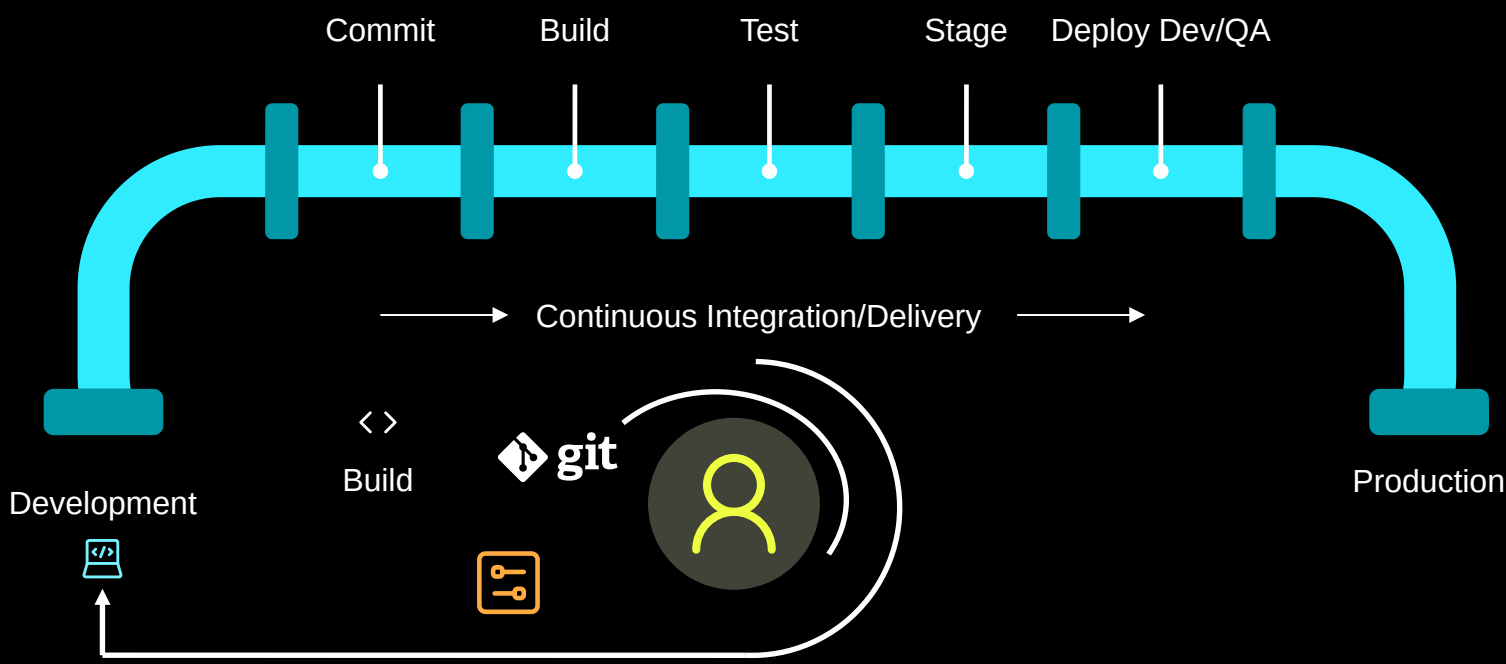
Our Attempts to Solve This Problem

Source of Truth



Our Attempts to Solve This Problem

Infrastructure as Code



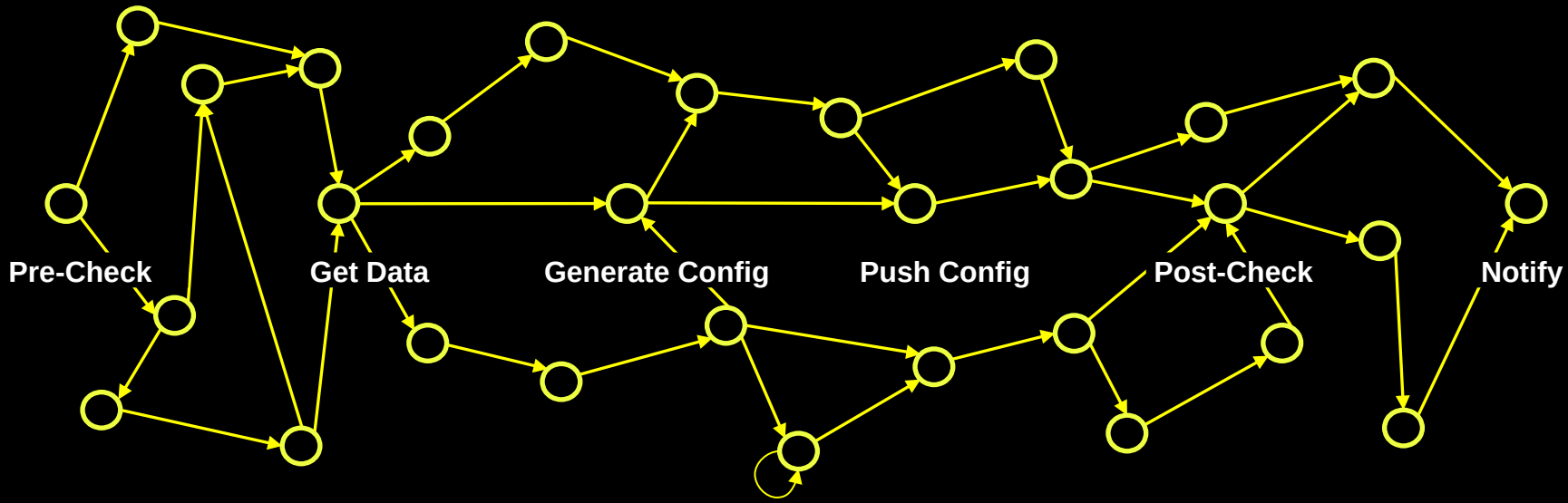
Why These Approaches Have Stalled

- Every attempt to provide an abstraction and codify our infrastructure has been disrupted by the next wave of technology
- Vendors - Network, Application, Cloud & SaaS - are attempting to differentiate. They fights abstractions.
- We have used DSLs and programming languages to capture our knowledge in code.
- We have attempted to train network engineers to be developers - typically focused on syntax and libraries rather than software engineering.

The best thing about determinism is its deterministic.



The worst thing about determinism is its deterministic



Error conditions, retry logic, device & system availability, data formatting, data correctness, etc, etc, etc

The Cost of Complexity

- Associated technical debt
- Voluminous testing for any change
- Every edge case we add makes the next edge case harder to manage.
- Troubleshooting issues not predetermined.

Find the right balance

Deterministic

- Fast execution
- Identical every time
- Known, predictable paths
- Pipeline-driven

Reasoning

- Handles exceptions
- Adapts to variance
- Human-like judgment calls
- AI-driven decision points




NETWORK AUTOMATION FORUM
Rethinking Automation
with Agents

What People Think AI Is for Networking

- Generating config scripts
- Machine learning for AIOps
- Chatbots for systems data
- Producing tailored visual data from stored data

Let’s explore how we can adopt Agents to both challenge our current mental model and augment our operating paradigm.

Where Is AI (LLMs) Good / Not Good in 2026?

STRONG

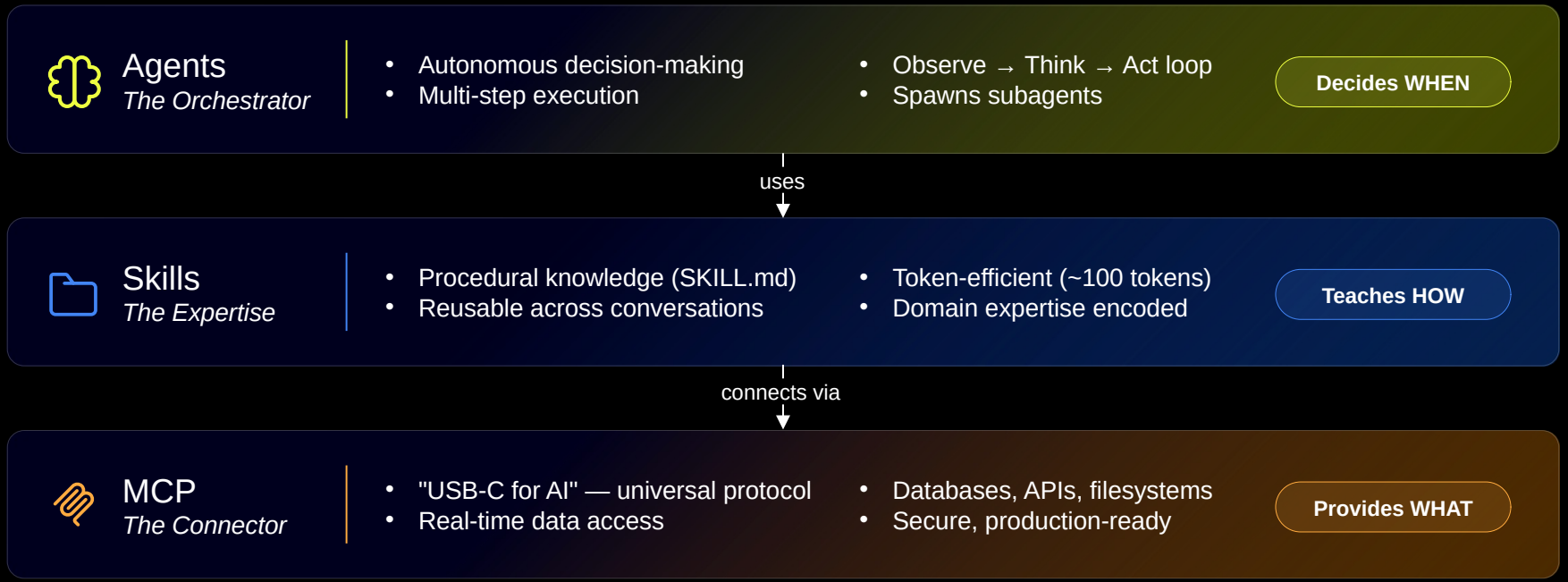
- Meta/abstract concepts
- Operational processes
- Network protocols
- RFC definitions
- General reasoning

GAPS

- Vendor-specific details behind paywalls
- Unpublished APIs
- No codebase scanning
- No TAC tickets/resolutions
- Need vendor SLMs

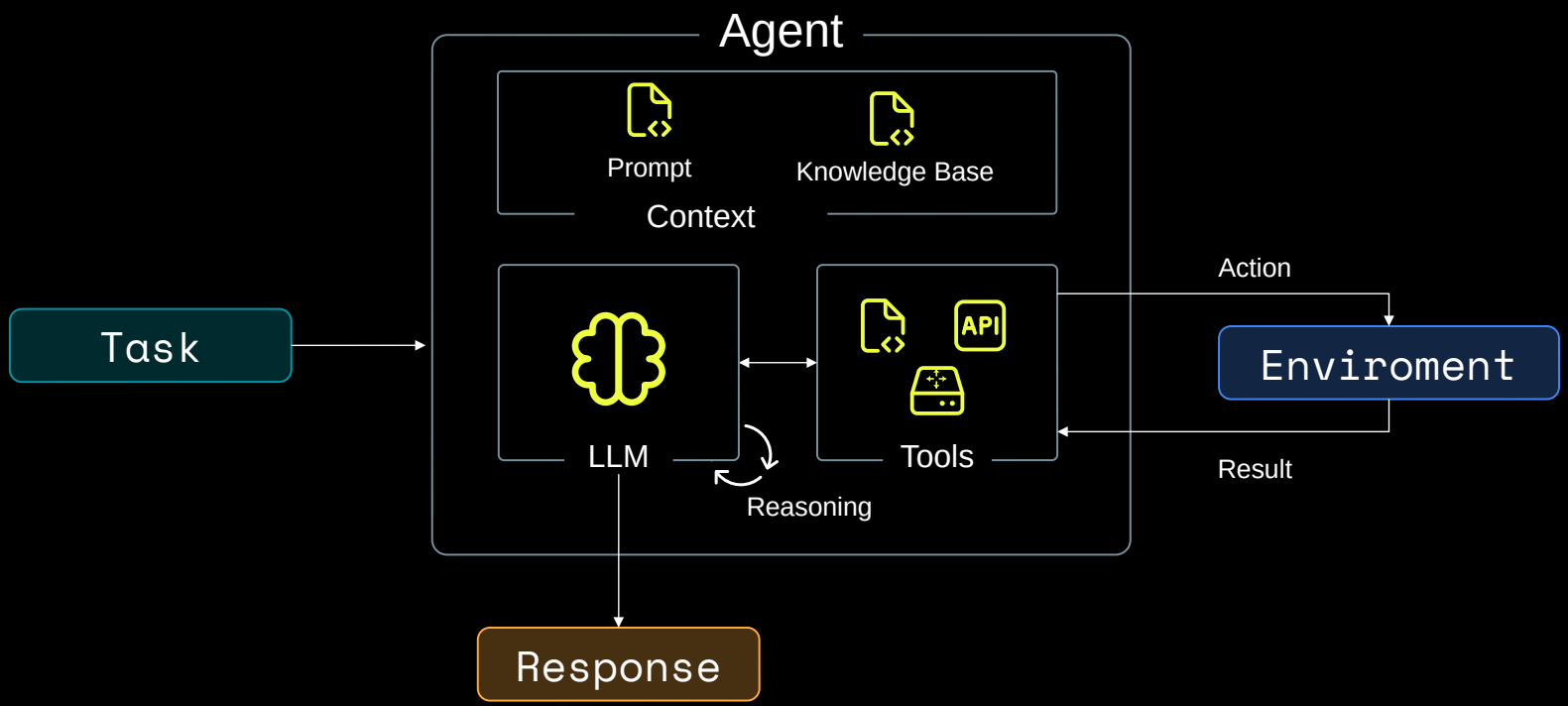
Tends to be good at things that are very well defined (protocols, RFCs) and general knowledge.

What is an agent?



These aren't competing approaches — they're complementary layers.

React Agent Introduction



New Hire Analogy

Training is as imperative for agents as it is for people.

User	Agent
School	LLMs
Processes	Prompts
Training	Skills
Tools	Tools

Context Is Everything

- The more ‘knowledge’ that is provided to an agent via prompts and context, the better decisions they will make.

- We have all experienced an outage call. When we join I don’t have context and we ask a ton of quesitons. The more context - the better outcome.

- With agents we need to remember you probably have a 1M token context window. We need to prioritize what fits in there - just like my 2am brain.

The Agent Platform

My discussions with customers and the market is that AI governance and guardrails are the main consideration above the question if agents will functionally work!

- Guardrails and security
- Oversight and tracking
- Management of agent work

Non Functionals >! Functional Concerns

NETWORK AUTOMATION FORUM

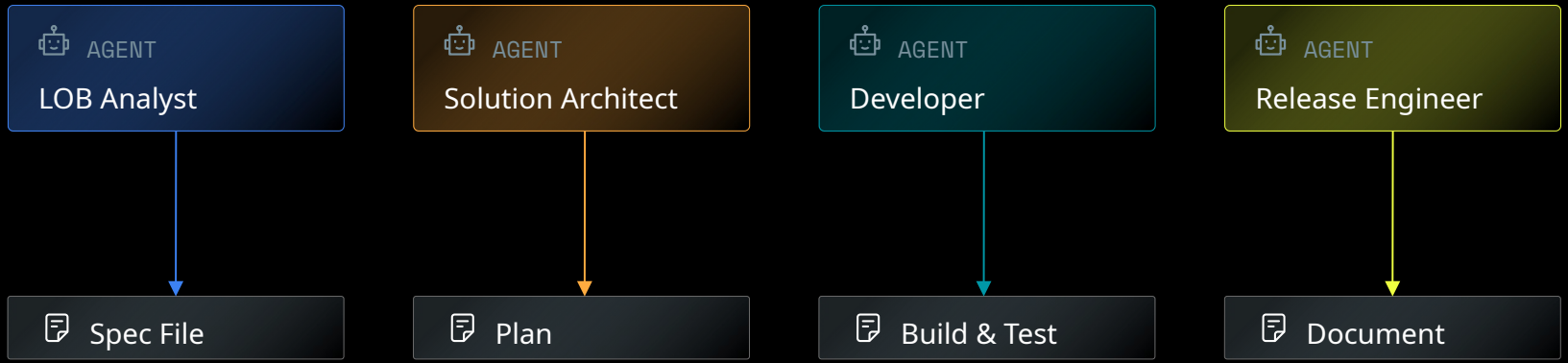
Future Vision:
Build, Execute, Operate



Build - AI-Driven Development

- Just like we borrowed build pipelines for IaC, software development is migrating to SDD (Specification Driven Development).
- We will use Build Agents (Cursor, Claude Code, Codex) to create our deterministic automations.
- Compare the lines of Python on GitHub to Ansible YAML - LLMs have vastly more training data for native code.
- This will be in programming languages (TypeScript, Rust, Go, Python) as DSLs were constructed for human abstraction.

Build Process for Agents



Execute - Fleets of Agents

Fleets of agents performing a variety of jobs. Centrally run in clouds or data centers - driven by LLM access and data gravity.

Examples:

- Software Upgrade Agents
- Port Turn-Up Agents
- Device Health Agents
- Service Health Agents

Engaged via forms, API calls, events, GitOps. Coordinating over A2A and MCP.

How Are Agents Different from Scripts?

DETERMINISTIC CODE

- Every decision, error path, and logic defined at build time
- Fallout when unknown conditions hit
- Same output every time
- Good in standardized environments

AGENTS

- Don’t require full pre-defined behavior
- Reason through unknowns like a human
- Guidance on error handling, not rigid paths
- Interactive - you can engage with them

Execute Agent Pattern

- Execute agents have their tools defined at Build time.
- We limit the tools so we can control what these agents can accomplish.
- Example: A read-only agent for device health cannot make modifications to network equipment if we don’t provide a tool that can accomplish a network change.
- Tool & skill management allows us to limit what and how an agent can accomplish.

Lets look at an example

NOC Connectivity Troubleshooting Agent

Prompt

You are an agent that supports troubleshooting network connectivity requests.

When troubleshooting you need to make sure you follow a regimented process outlined below. You should never modify any configurations and always notify the team via Teams and update appropriate ServiceNow Tickets.

Your procedure is the following:

1. Ping the device from our local NOC server and jump host.
2. Traceroute to the device from NOC server and jump host.
3. Check status on the performance monitoring system.
4. Notify and update assigned ticket and NOC team board.

Lets look at an example

NOC Connectivity Troubleshooting Agent

Knowledge Base

- Network Use Policy
- IT AI Policy
- Maintenance Window Policy

Tools

- Ping
- Traceroute
- Ticket Update
- Notification

Skills

- Network Troubleshooting Skill
- Change Management Skill
- Network Connectivity Health Check Skill

Operate - Engineers as Agent Managers

Engineers and Operations teams will have personalized agents augmenting their work:

- Look at work in SNOW queue
- Run tools and updates
- Send emails and Slack messages
- All under engineer management and oversight

We become managers of our teams doing the work.

Operate Agent Pattern

- You will interact with these agents in similar ways we interact with other humans today - Slack, email, ticket queues, message boards.

- Unlike centralized execute agents, operate agents pick their tools at run-time.

- This is like a human Operator who has to decide whether to log into a device, modify a system, or contact someone else to assist.

Lets look at an example

BGP L1 Agent

Prompt

SOUL.md	→	HOW to think	(judgment, values, boundaries)
IDENTITY.md	→	WHO you are	(name, role)
USER.md	→	WHO you serve	(reporting chain, approvals)
TOOLS.md	→	HOW to act	(tool sequences, decision trees)
BOOTSTRAP.md	→	WHAT you know	(network design, failure patterns)
HEARTBEAT.md	→	WHEN to act	(monitoring loops, schedules)
MEMORY.md	→	WHAT you've seen	(past incidents, known issues)

BGP Specific Skill

Tools = "I can run show_command on a device"

Skills = "When a BGP alert fires, first check the interface with show_command, then check BGP summary, then check the route table, then look for recent config changes in the audit log. If the neighbor is down and the interface is up, it's likely a config or peer issue — escalate as P2. If the interface is down, check the optic with show_transceiver and check the circuit with circuit_status."

Prompts = "You are a cautious NOC agent. Never make changes without approval. Always verify before escalating."

Skills are the **runbooks turned into agent-consumable procedures**.

Lets look at an example

BGP L1 Agent

Tools

```
Show Interfaces
BGP Toolkit
Email NOC
Escalate Ticket
```

Skills

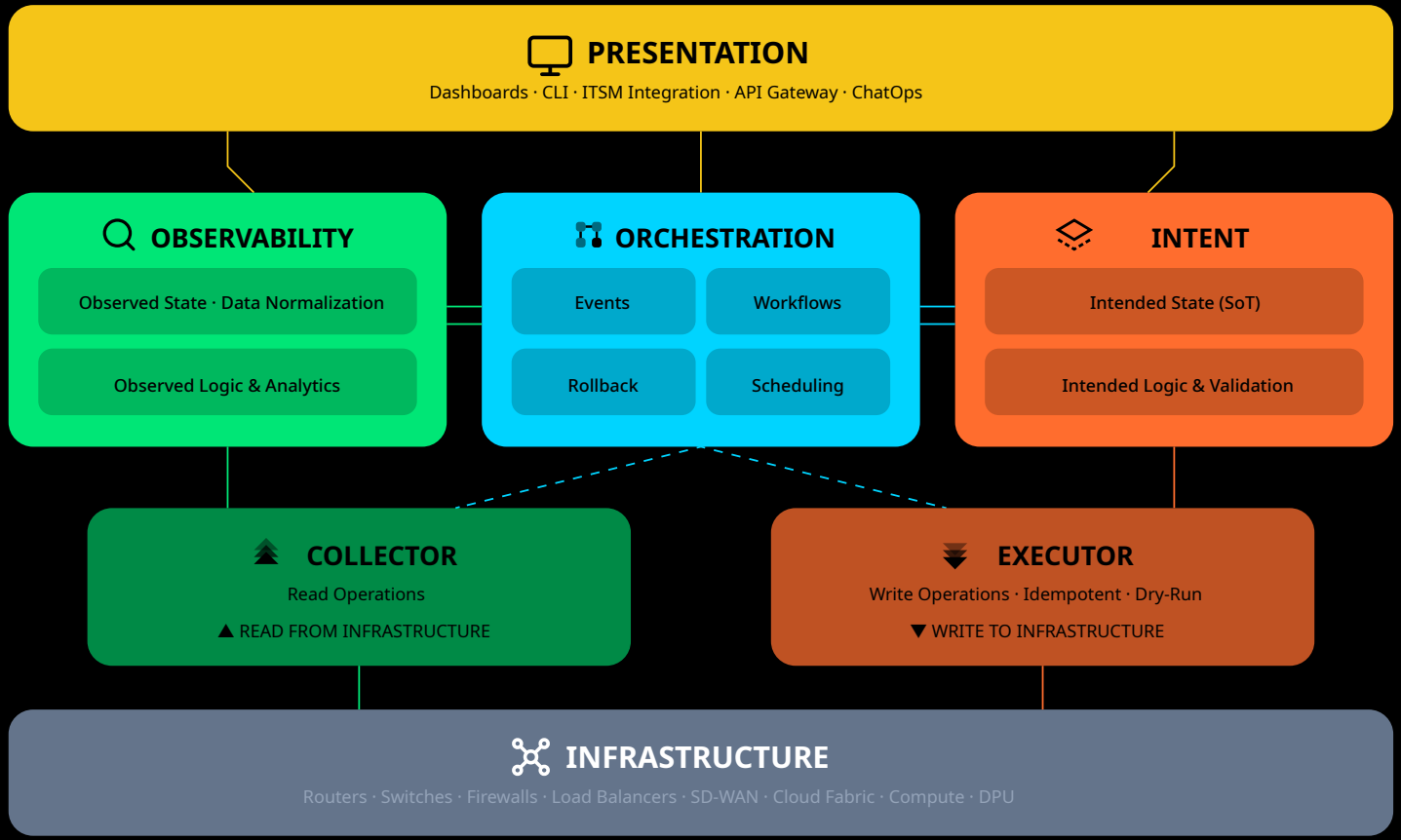
```
BGP Troubleshooting Skill
Change Management Skill
Email Skill
```


NETWORK AUTOMATION FORUM
Resolution / Conclusion

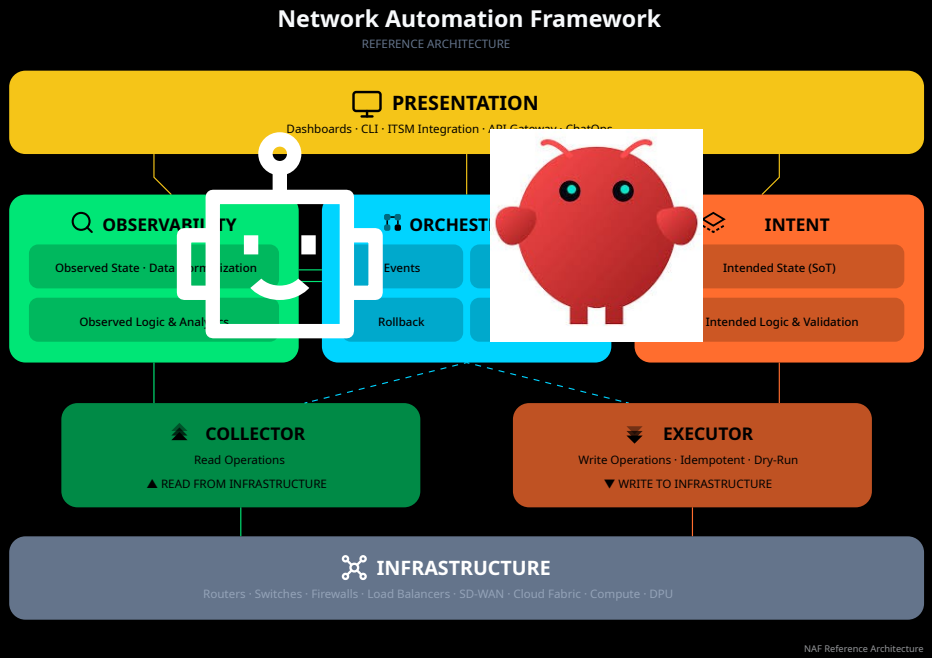
AI technology is proven enough for us to reimagine how we operate infrastructure with this technology as a central component.

Network Automation Framework

REFERENCE ARCHITECTURE



Can we consider Agents as users?



Network Agentic Forum?

Deterministic vs Reasoning

DETERMINISTIC

- Fast and same every time
- Measured in CPU speed
- Can be distributed
- Close to the network

REASONING

- Handles exceptions & variances
- Like a human does today
- Relatively slow
- Highly centralized

Back to First Principles

- We've spent 15 years converting Design Documents & MOPs into scripts and pipelines.
- We need to go back to first principles and focus on the design and operating criteria for our infrastructure.
- The quality of this documentation must be excellent and version controlled - agents will use this without interpretation.
- This is a great opportunity to focus on Engineering and Operating principles over development syntax.

Documentation for Agents

Today's design documents are written for humans with assumptions about their training. Agents require very specific operating paradigms.

Imagine providing an agent:

- Network diagrams & design documents
- Memory tied to source of truth data
- Network, AI, and Security policies
- Outage and degradation procedures
- Knowledge of applications on the network

I posit that the agent will be able to make great decisions.

Network Engineering can focus on Systems Thinking and less on the CLI & API syntax.

NETWORK AUTOMATION FORUM

Practical Next Steps



Start with Your ChatGPT Experience

- Everyone has had their ‘ChatGPT experience’ - a chatbot interaction where you request the agent to do something for you.
- Document generation, config generation, or recommendations.
- This is the starting point - but it’s not the destination.

Move to an Agentic Experience

- Build ReAct agents that perform a combination of reasoning and acting based on your prompting.
- Context from your knowledge base: policies, design documents.
- Limit tooling: API calls, CLI commands, scripts.
- Start with read-only agents. Adopt like you started with automation - lots of human in the loop approvals.
- Integrate with your tools today (ticketing, notification) to stay in the loop.

Challenge our Vendors

- We need the expert knowledge of the systems in a consumable format for agents.
- Skills & MCP Tools can be provided to leverage systems and devices more aggressively.

Governance & Strategy

- AI governance is a real thing. Work with your internal teams to determine what LLMs and tooling is available to build with.

- We should consider rethinking our strategy to put agents at the forefront.

- Agents change the possibilities of what can be accomplished. We need to surround them with the skills and context to be successful - just like we have for Network Operations.

Rethink automation with agents at the core.

- Surround them with the skills and context to be successful.
- The intelligence layer we've been missing is here.


NAF
NETWORK AUTOMATION FORUM
Questions?