Securing the Modern Management Plane and other Lofty Goals

Scott Campbell

Helsinki, Finland

10 June, 2026

tnc26

Co-funded by the European Union

Problem Definition

How has the use of automation and orchestration changed the mental model required for securing the modern management plane?

Not surprisingly, I contend the answer is “it depends.”

Problem Definition

How has the use of automation and orchestration changed the mental model required for securing the modern management plane?

Not surprisingly, I contend the answer is "it depends."

Are you familiar with securing modern API driven complex applications?

Introducing Context to the Discussion

Why?

- No problem exists in isolation - related problems help build understanding
- No solution is perfect - every effort to solve a problem will create others

Introducing Context to the Discussion

Telemetry

High quality telemetry changes how we perceive the network (time resolution, data types)

Paradoxical - large benefit, new problems:

- Data maturity problem
- More data not always good - understanding
- More data not always good - resources/technical
- Issue magnifier: Pressure breaks things

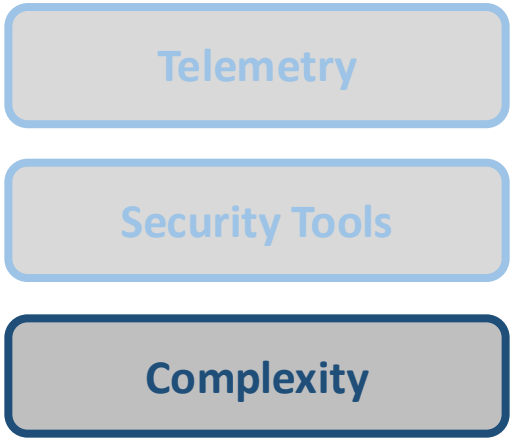
Introducing Context to the Discussion

Security tools create and enforce security boundaries:

- Firewalls
- VPNs
- Identity and authorization management
- Infrastructure management

- Huge incentive for attackers to find vulnerabilities
- Vendor issues with basic application safety
- Protect your network from your firewall?

Introducing Context to the Discussion



Complexity is both Technical and Social

Complex systems behavior is unpredictable, with strong coupling multiplying effects

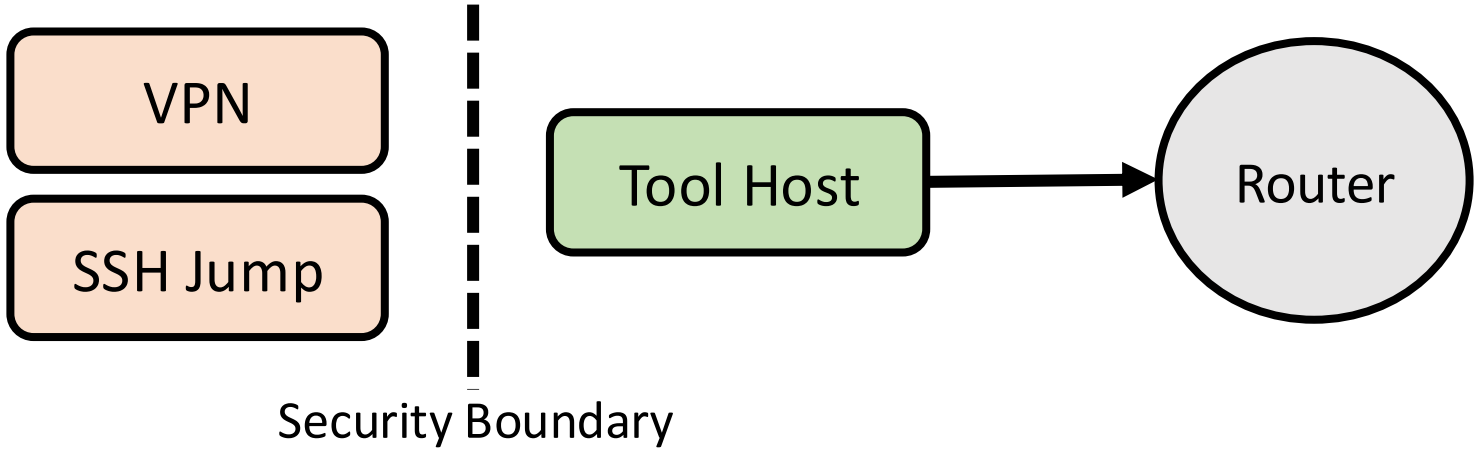
Exists and grows on many interacting layers:

- Hardware layer (smaller, faster)
- Router config and control
- Platform software control – think Kubernetes
- The number and interplay of humans (systems people, developers, and engineers)

“Historic” Management Plane Mechanics

Router configuration using a combination of SSH, Rancid, SNMP, and Netconf scripting/templating.

Focus has primarily been on automation.



Terminology

Automation:
- Atomic: do a single thing
- Device specific: (Nokia/Juniper)
- About the final artifact

Orchestration:
- Involves multiple systems
- Agnostic to the local device
- Represent the business logic

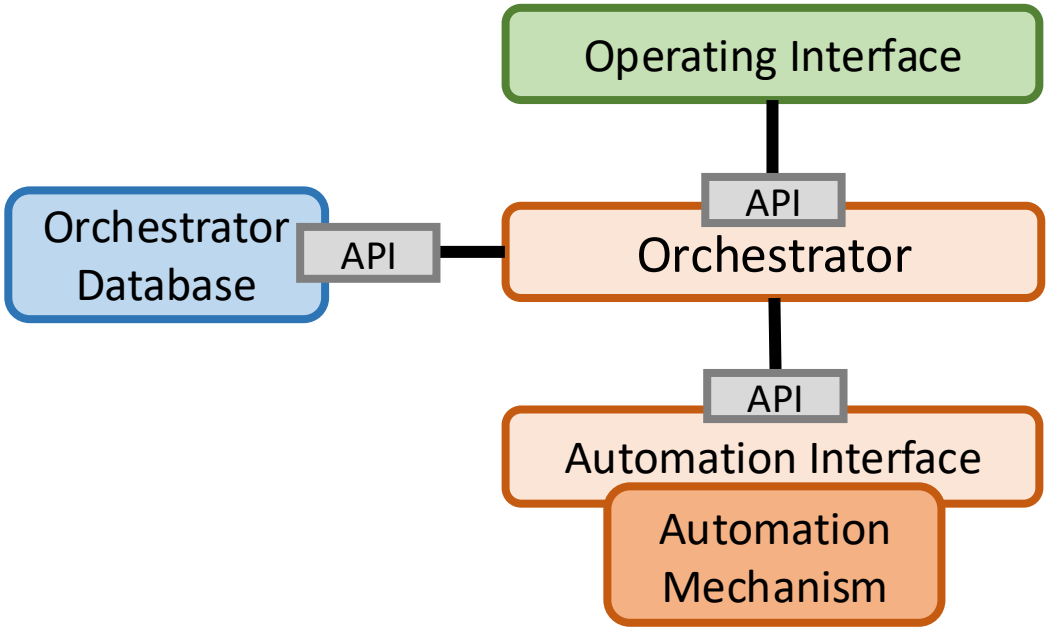
Orchestration means:
- No longer express discrete changes
- Describe desired changes in business logic
- Will focus mostly on that.

Terminology

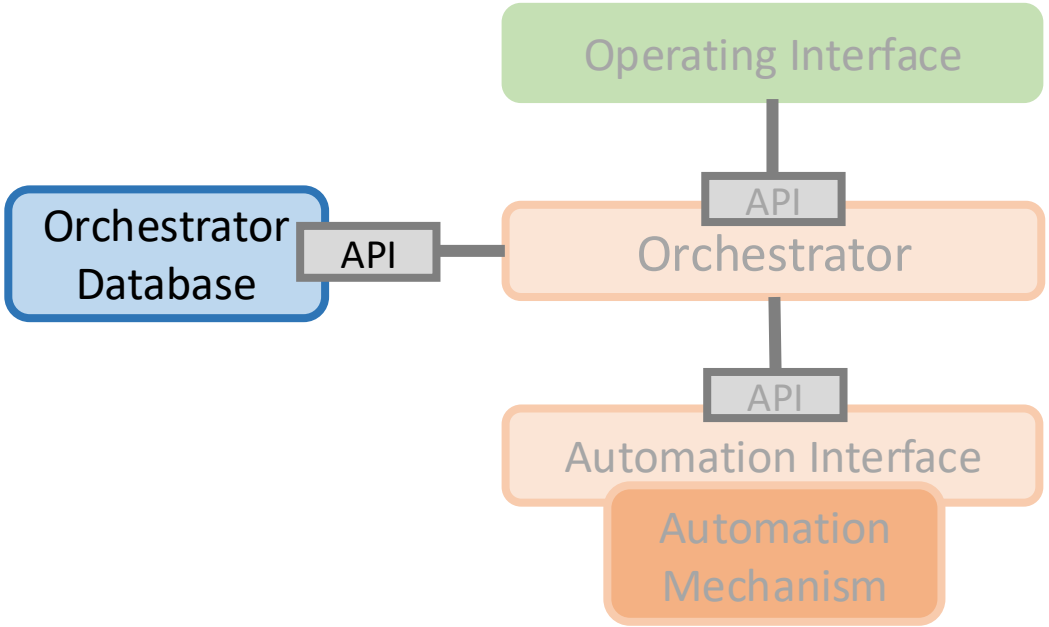
Control Plane:
- Network Protocols
- Establish topology
- What to do with packets via protocols

Management Plane:
- Administrative access
- State information
- Configuration

Simplified Orchestrator Stack



Simplified Orchestrator Stack: Database



State Reservoir

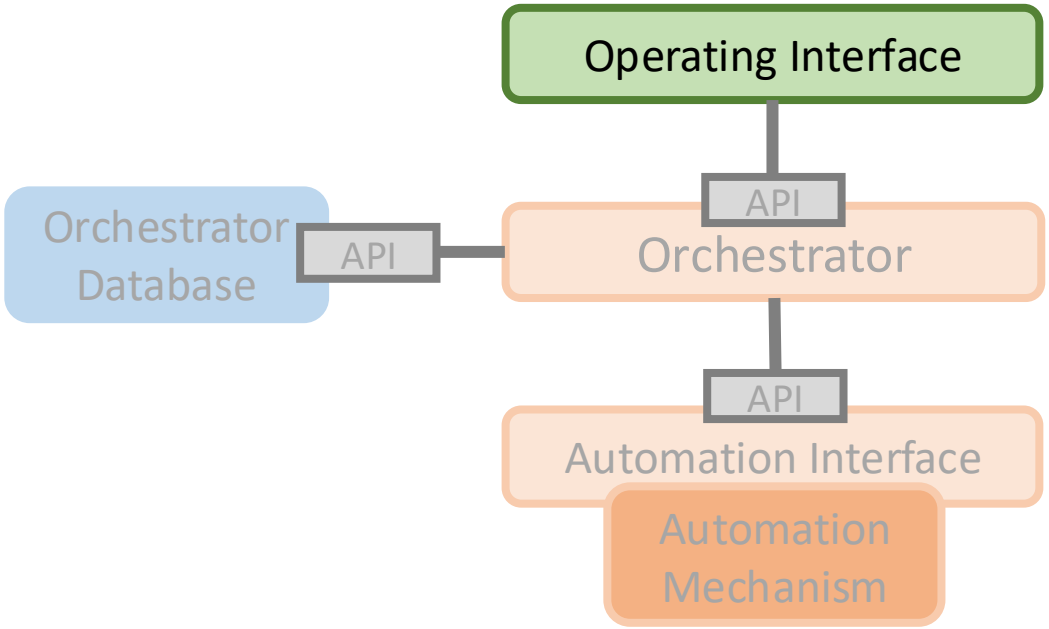
DB contains definitions for

- Abstract models for Services
- Routers/Interfaces
- Logic for automation

Network state and statistics

Long term state for the whole workflow process

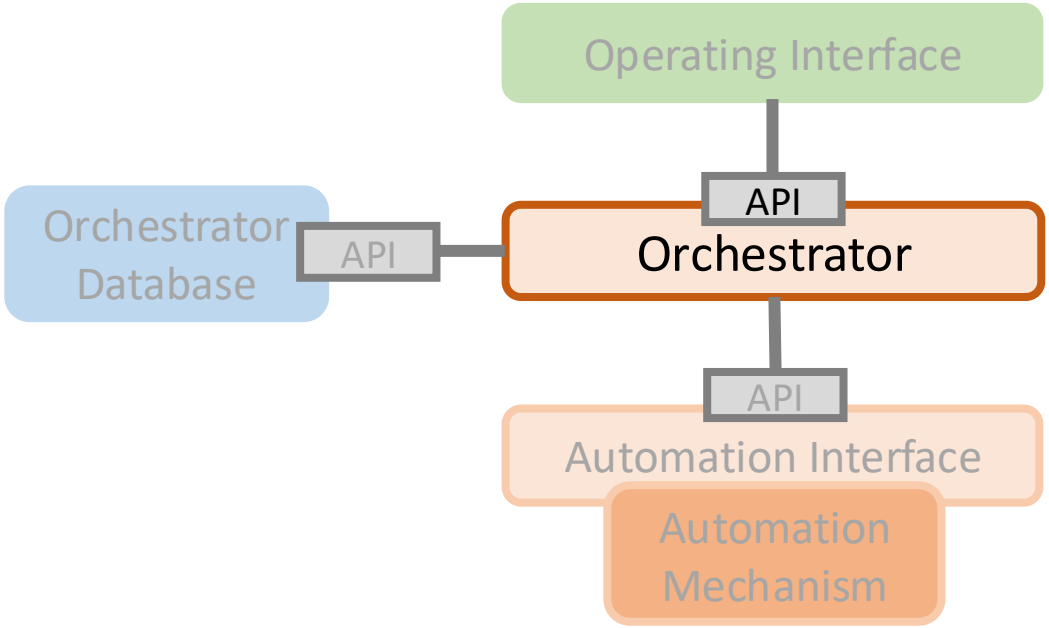
Simplified Orchestrator Stack: Interface



Interface for Orchestrator

- Web server and all related tools
- Redundancy
- Auth and access control

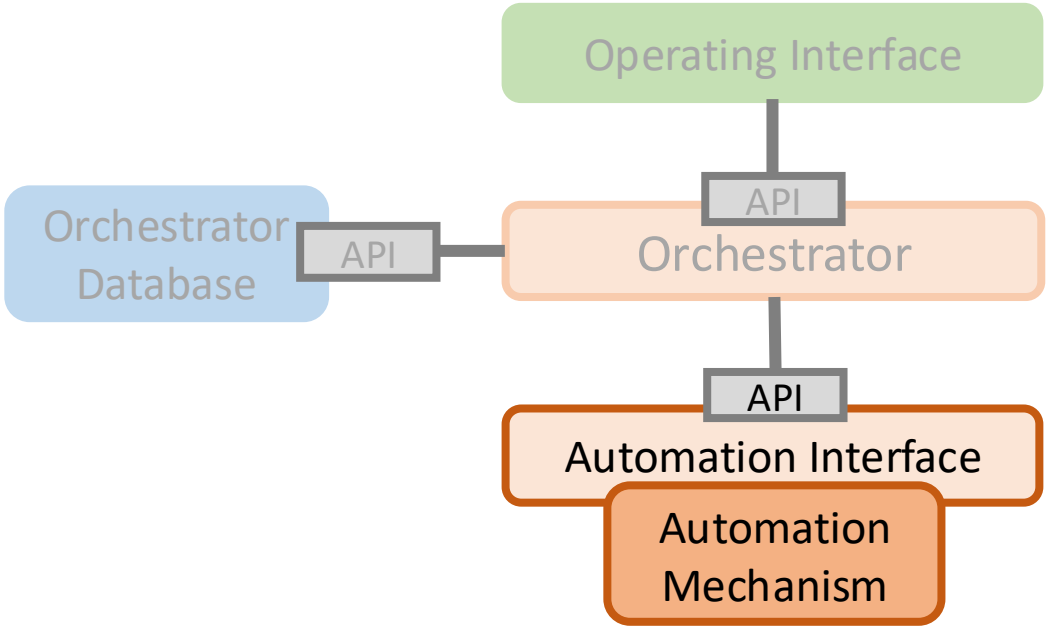
Simplified Orchestrator Stack: Orchestrator



Organizes Behavior

- Takes request from Interface
- Parse/process request into objects
- Get object definitions from DB
- Create automation request steps
- Interact with IP/DNS allocation
- Pass steps to Automation Interface

Simplified Orchestrator Stack: Automation



Performs Actions

- Takes automation requests from orchestrator
- Convert requests to configuration changes
- Apply changes to network device
- Gather operational information
- Return results

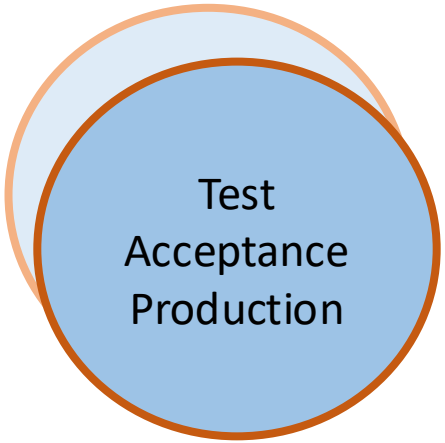
Security Complications

Security Complications

Application Logging and Security

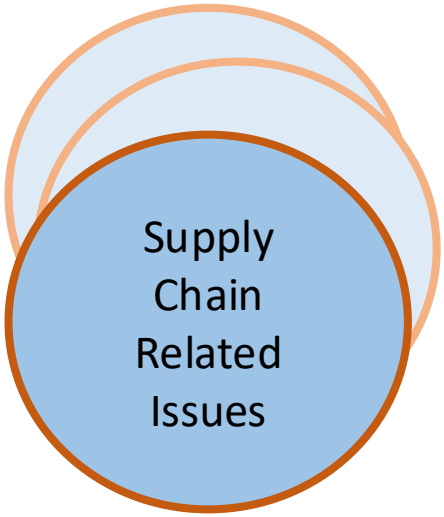
- Diversity and volume of new log types
- Realistically most logs are already ignored
- Analysis issue: Capacity and capability

Security Complications



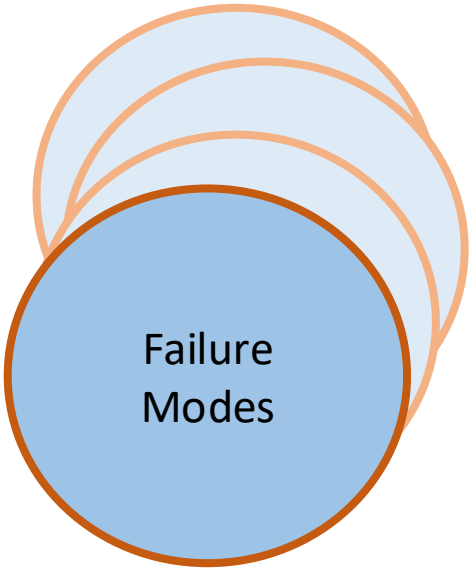
- Credential re-use across environments
- Attacker movement between: Implicit Trust
- Test/Acceptance often less tightly controlled
- Data consistency: critical yet risky

Security Complications



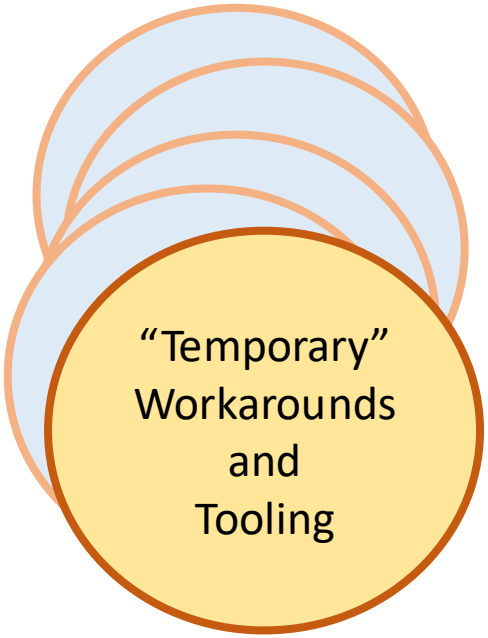
- Compromises in the ecosystem of packaged code and developer tooling
- Currently an unbound problem - example Visual Studio plugin leading to upstream compromise
- Risk with patching balanced against increased vulnerability velocity (from mechanized bug hunting)
- Very complex problem

Security Complications



- Complexity introduces new failure modes
- Boundary conditions for safe operation
- Risk is difficult to model
- Unknown unknowns …

Security Complications



- Not in your environment of course
- Byproduct of real production pressures
- Commonly seen in *other peoples’* organizations
- Good intentions – not malicious, just dangerous

Wrap Up

Automation is not a novel design and does not represent a huge change from a modern systems or application perspective.

- Every point in “context” also apply to modern systems and applications
- This is a question of securing a known design pattern

The mental model for securing the Orchestrator Stack is (mostly) understood.

Wrap Up

This design can be secured in terms of modern dev-ops and platform engineering:

Telemetry: New data sources, need to understand to monitor
Security Tools: No single tool will fix this, might make it worse
Complexity: Software, Hardware, Network Policy, Humans

- Very specific skill set for security
- The details are complex and require potentially new behaviors for security

From this perspective there are two paths:

Path I

You have experience securing modern API connected applications

You have experience securing modern software controlled infrastructure

The ability to apply what you know to network automation

You have a good framework for securing Management plane automation and orchestration. Focus on Basics:

- Integration
- Boundaries
- Communication

Path II

You lack experience in securing API connected application layer security.

- Design patterns for securing this type of application
- Well defined set of things to do
- Basics: Authentication, logging, analysis, communication
- Benefit your larger maturity and risk-reduction universe

Besides the usual Management Plane security controls, this is mostly a question of applying what the community has learned in application and systems automation.

The Last Slide!

Most organizations are between Path 1 (familiarity) and Path 2 (unfamiliar)

Automation is not a novel design and does not represent a huge change from a modern systems or application perspective.

Attempting to "solve" our problem requires something difficult: *cooperation amongst specialists*

The real mental model that needs addressing is the imposition of organizational boundaries: not "just a security problem"

Operating Interface | API | Orchestrator | API | Automation Interface

Thank you

Any questions?

scott.campbell@geant.org

Co-funded by the European Union