The only automation tool you need.

SOITRON aut

automation tool



Martin Kyrc
Soitron | Network and Automation System Engineer





Red Hat Innovate Together 2026, Prague, CZ

CHALLENGE

~ 2019

Number of services
DC: ~250 services, 3x geo-datacenter
DMZ: ~150 services, 2x geo-datacenter

The Goal: Migrate data center from "**legacy**" to "**software-defined**"



"Legacy" technologies:

- Cisco SW/RT
- Cisco ASA/FWSM(!)
- Cisco ACE(!!!)
- F5 BIG-IP (LTM)
- Cisco GSS (!!!)
- 10Gbps
- CLI based configuration
- No or limited app documentation
- Devices uptime 10y+, EoS, EoL…
- No/limited API

"Software-defined" technologies:

- Cisco ACI (APIC)
- Cisco FTD/FMC
- f5 Viprion (LTM, AFM, DNS, WAF)
- 40/100Gbps
- Open API

MISSION

Service migration
+
Change of mind

Primary goal

Service migration
with full-automation

Secondary goal

Change “Legacy thinking”
to “NetOps thinking”

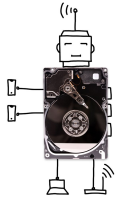


Why?

Migration accelerating
Reducing human error
Uniform configuration style
Move from traditional to modern NetOps thinking

DECISION

Looking for the right tool



There is no one perfect automation tool!

NETMIKO
ANSIBLE
puppet
POSTMAN
CISCO ACI
HashiCorp Terraform
NSO
SALTSTACK
CHEF
vmware NSX

DECISION

Final solution



Our custom on-premise solution based on...

python™ GitLab ANSIBLE

netbox HashiCorp Vault

What is it?



- On-Premise modular NetOps tool (Docker based, no GUI)
- Support for manual CLI or automated CI/CD workflow integration
- Very flexible Python-based wrapper supporting multiple orchestrators (e. g. Ansible) including direct API device connection
- One (or more) source of truth (GIT or/and NRM)
- Sensitive data stored on secure place (HashiCorp Vault)
- Support for (customized) workflow from user front end up to end device

Is neither a product nor a solution.
~~It is an approach.~~

Evolution

Soitronaut v1.0

- One Monolite
- Not very scalable
- But very useful!
- Device support:
  - Cisco ACI
  - Cisco FTD
  - F5 LTM, AFM, networking (imperative)

Soitronaut v2.0

- Redesign from scratch
- Modular concept
- Docker image
- Device support:
  - Cisco ACI
  - Cisco FTD
  - F5 LTM, DNS (declarative)
  - F5 NW (imperative)

New features

- We are ready for BIG-IP Next (declarative only)
- F5 AS3 improvement, app-based support
- CI/CD integration
- NRM/IPAM (NetBox), HashiCorp Vault
- Added support for:
  - F5 OS
  - Cisco DCNM
  - Juniper Junos
  - Cisco IOS-XR

Other features

- Other supported features:
  - Cloud (MS Azure)
  - Integration into existing CI/CD workflow
  - Input configuration generator
  - Validation tests
- Roadmap:
  - Cisco IOS-XE
  - Palo Alto
  - Ironport WSA, ESA
  - … (on demand)
- Our journey continues…

~2019/2020 → ~2022/23 → 2024+ → Today

Device support

Device	Available since	Configuration type	Validation support	Device configuration parity
Cisco ACI (APIC)	2019	Imperative	No	****
Cisco IOS-XR	2024	Imperative	Yes	****
Cisco IOS-XE	2026	Imperative	No	* (in progress)
Cisco DCNM	2025	Imperative	Yes	***
Juniper Junos (FW)	2025	Imperative	Yes	*****
F5 LTM	2019	Declarative	Yes	*****
F5 DNS	2022	Declarative	Yes	****
F5 Viprion	2019	Imperative	No	*** (focused to NW only)
F5 OS	2025	Imperative	No	*** (focused to NW only)
F5 NW	2019	Imperative	No	****
Ivanti ICS (VPN)	2026	Imperative	No	**

Roadmap

- PowerDNS
- Cisco Ironport WSA
- Cisco Ironport ESA
- Palo Alto

Support for additional devices on request

Imperative vs. Declarative configuration method

Imperative = HOW

- Focuses on actions
- Describes step-by-step process
- If something goes wrong half-way, state may be inconsistent
- Deployment time: “minutes” (lot of API calls)

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

Declarative = WHAT

- Focuses on outcomes
- You describe the desired end state, not how to get there
- Easier to make idempotent and retry safely
- Deployment time: “seconds” (one API call)

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





Imperative vs. Declarative configuration method

Imperative = HOW

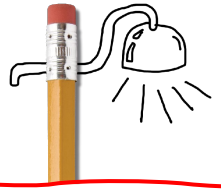
- Focuses on actions
- Describes step-by-step process
- If something goes wrong half-way, state may be inconsistent
- Deployment time: “minutes” (lot of API calls)



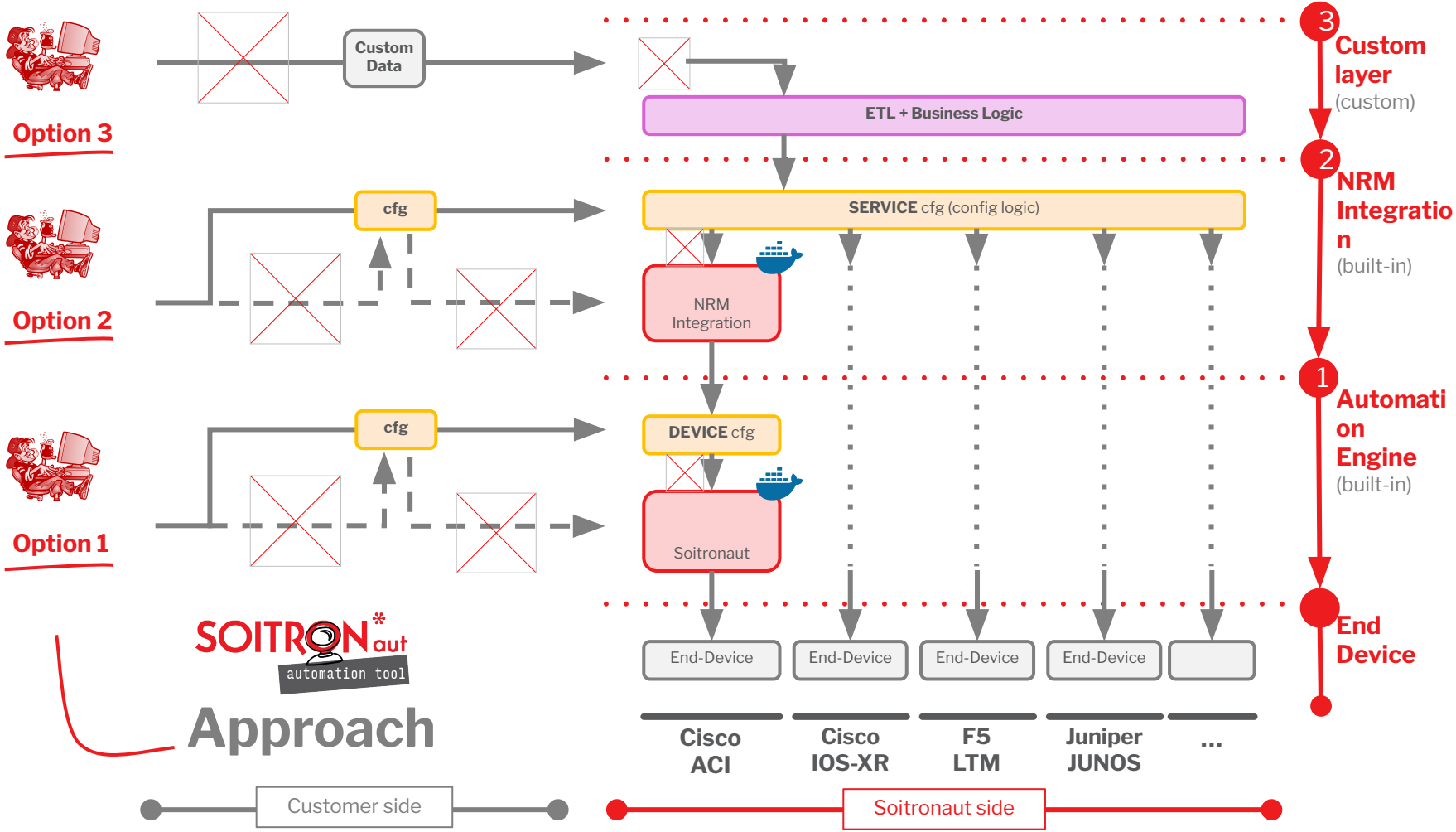
Declarative = WHAT

- Focuses on outcomes
- You describe the desired end state, not how to get there
- Easier to make idempotent and retry safely
- Deployment time: “seconds” (one API call)

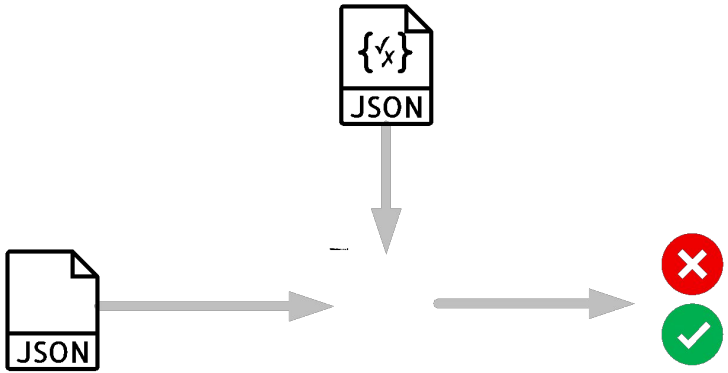


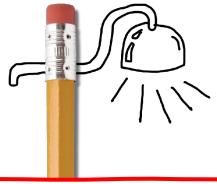


"In modern automation, declarative is generally preferred because it focuses on the desired end state rather than the specific steps required to reach it. However, both methods have their place depending on scale and complexity."


Option 3
Option 2
Option 1
Custom Data
cfg
cfg
ETL + Business Logic
SERVICE cfg (config logic)
NRM Integration
DEVICE cfg
Soitronaut
End-Device
End-Device
End-Device
End-Device
Cisco ACI
Cisco IOS-XR
F5 LTM
Juniper JUNOS
...
3 Custom layer (custom)
2 NRM Integration (built-in)
1 Automation Engine (built-in)
End Device
SOITRON aut automation tool
Approach
Customer side
Soitronaut side

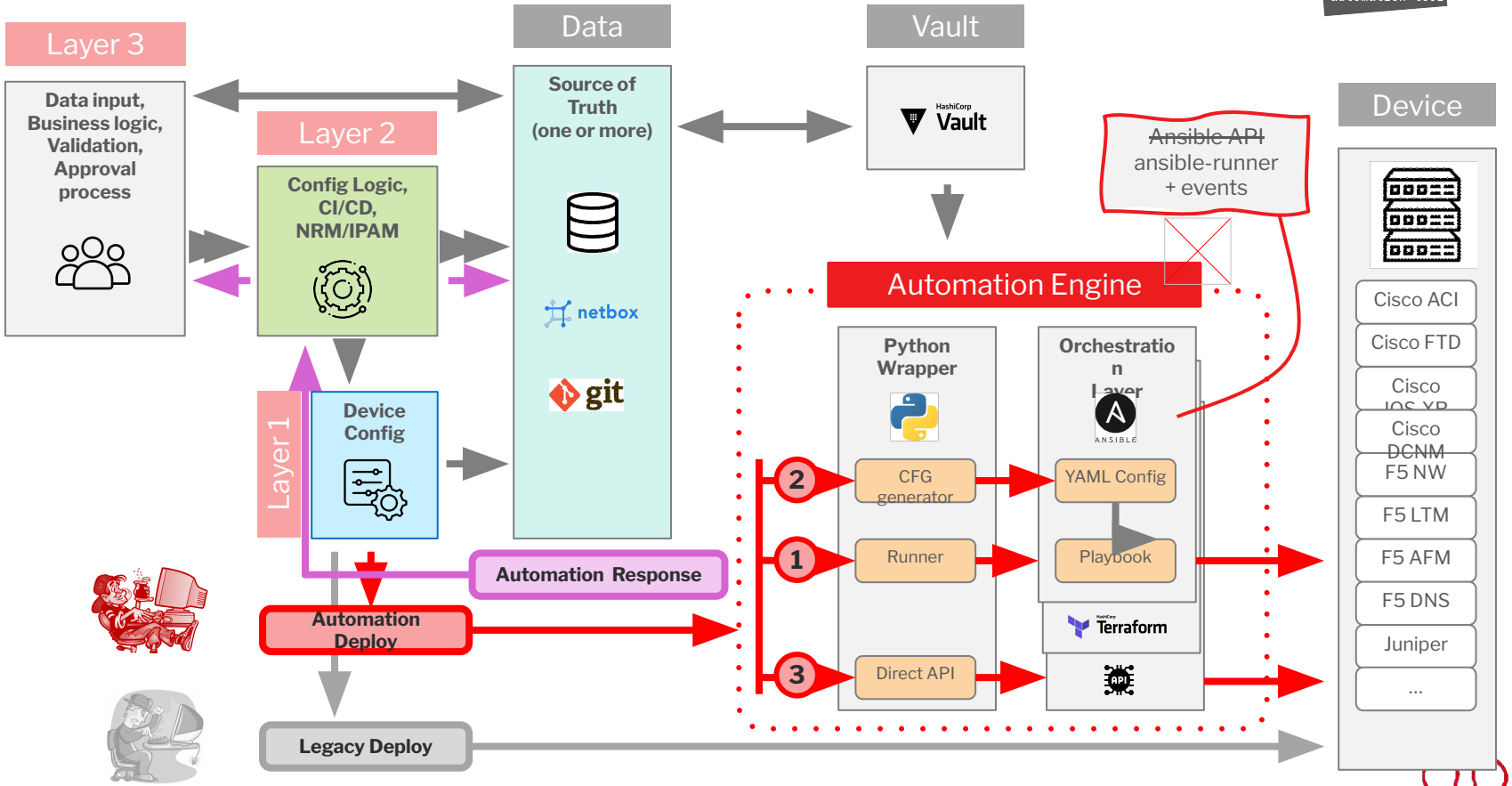
JSON Schema validation



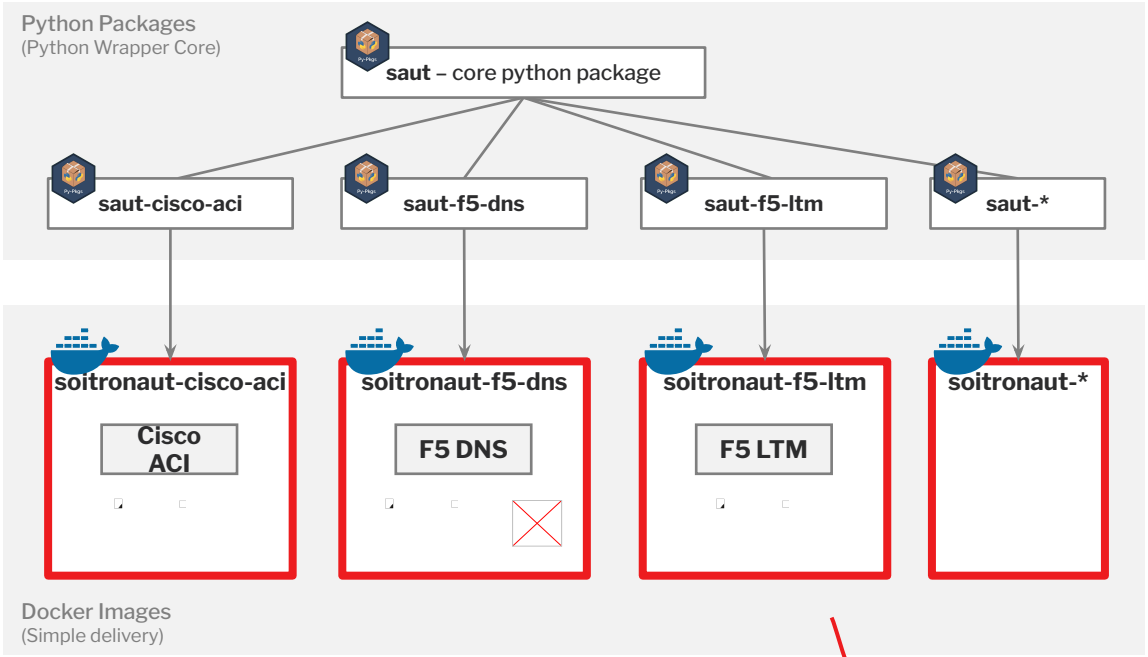


A JSON Schema is a definition for describing the data structure. It is commonly used to validate configuration files and data exchange formats. It is used by both people writing configurations and systems that process and validate the data.

Automation Engine: Configuration workflow



Automation Engine: Architecture



Provides common and core functionality

Extends core functionality with device-specific features

Docker image
delivered to the customer
(python pkgs, orchestration, other required functionality)

Modular and scalable solution
Easy to extend and support
Simple delivery to customer

Why NetOps?

Pros

- Faster provisioning (!!!)
- Scalability
- Human mistakes elimination
- Re-usable configuration
- Configuration versioning (history, audits, security)
- Coverage across the full service lifecycle
- Moving towards IaC (Infrastructure as Code)
- Full control over change process

Cons

- Complexity
- Less hands-on control
- Resistance from Traditional Network Teams :)



Requirements

- Mindset change from traditional networking to NetOps
- End-device API support
- More cooperation between teams (!!!)
  (NW, SEC, DEV, APP)

Automation isn’t just for experts.

It’s for anyone bold enough to start.

The only thing to fear is standing still.

The only automation tool you need.

Thank you for your attention

Feel free to stop by our booth to ask for more details and continue the discussion

SOITRON aut
automation tool

www.soitron.cz/soitronaut