

Joris Claassen
Systems Engineer EMEA
NLNOG Summer BBQ

Network Architecture as Code

A glowing lightbulb is the central focus, with a complex network diagram of nodes and connections inside. The lightbulb is surrounded by a cloud of shattered glass fragments, suggesting a breakthrough or a new idea. The background is dark blue.

Network Architecture as Code

- To define a network using descriptive data models
- Or; model-based network-wide automation
- ~~Or; Software defined networking~~
 - ...
- Or; (big parts of) "Intent-based" networking
 - NLNOG Day 2022
 - >> "DIY Intent-based Networking - Robin Gilijamse"
- It always comes down to:
 - Setting domain-wide policies, not device-level configuration
 - >> The industry has been talking about this for a while now...

BUZZWORD				
B	I	N	G	O
Machine learning	Next-gen	Innovative	Mobile	Revenue sharing
Apps	Resilient	SaaS	Blockchain	Efficiency
Data-mining	IoT	★	Gamification	Sensors
Pilot	Savings	Dashboard	Scalable	Disruption
SoLoMo	Engagement	Cloud	Big data	Smart

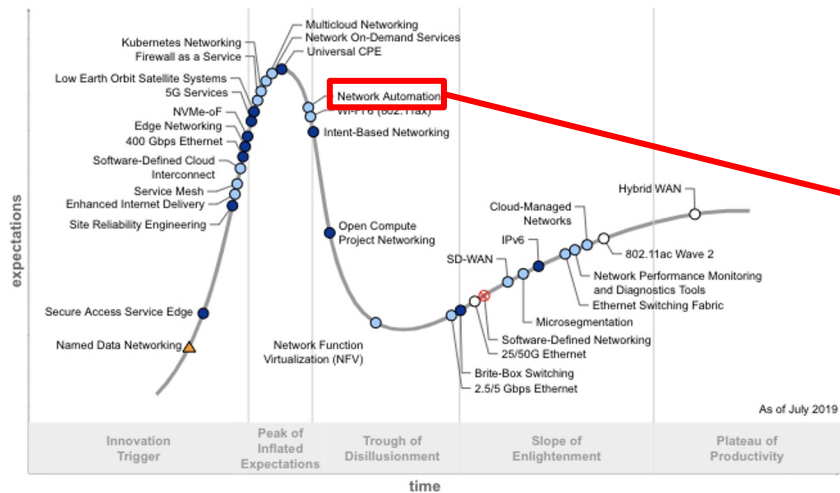
If it's not new...

Why should I care *now*?

Network Automation is Maturing

**Gartner Hype Cycle for Enterprise Networking
July, 2019**

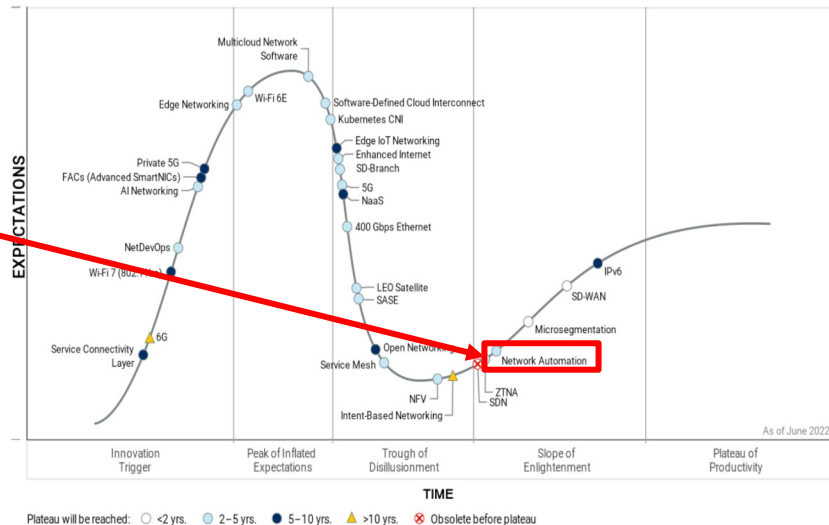
Interactive Hype Cycle



From 'Peak of Inflated Expectations'...

**Gartner Hype Cycle for Enterprise Networking
July, 2022**

Hype Cycle for Enterprise Networking, 2022



Source: Gartner (June 2022)

...to 'Slope of Enlightenment'

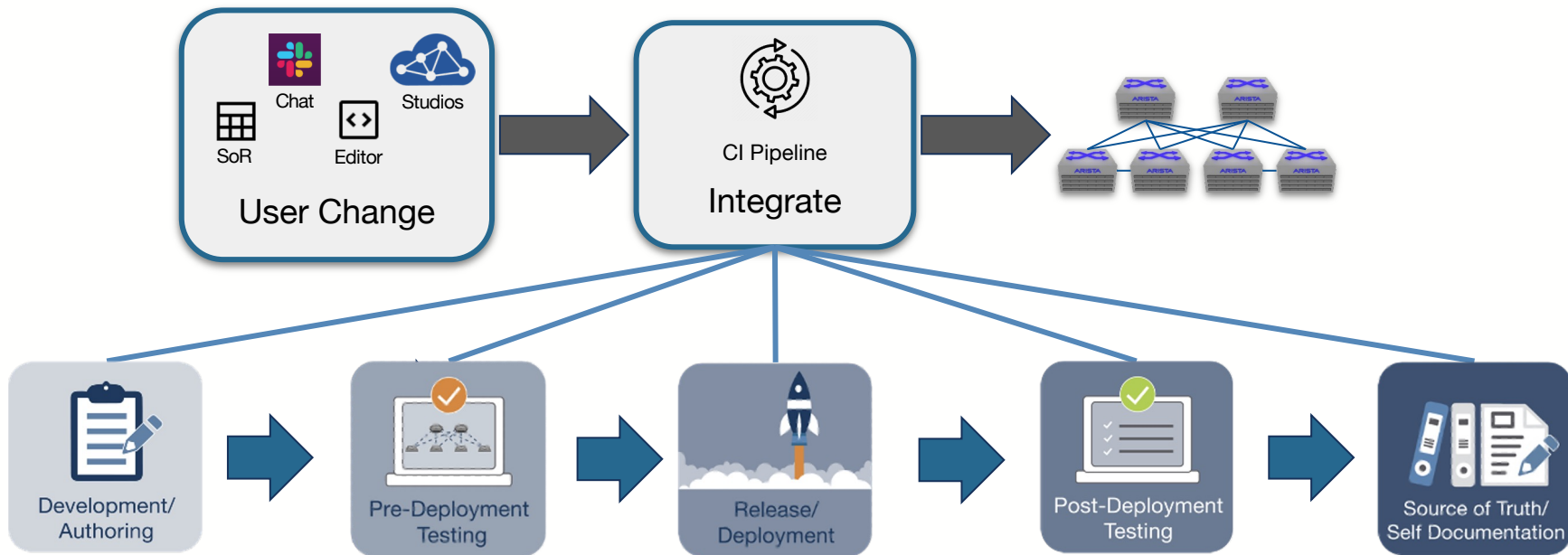
**Sorry, still not convinced.
Seen too many of those charts.
Why should I *really* care?**

Network Architecture as Code

- It can save you a **lot** of time *in the future*
 - Don't be scared to invest time - you reap what you sow
 - Automate your **design** using data models
 - >> Ideally these models should hold logic that abstract things away
 - Automate your **testing**
 - >> Test that the implemented logic worked well
 - Automate your **deployment**
 - >> To a level that feels comfortable for you
- **How to convince your manager you need to save time:**
 - You **can** do this **today** with just **open source tools**
 - **Validated** data models are **available**
 - >> No need to invent the wheel



Continuous Integration Pipeline Workflow



All changes are managed through a CI workflow

Authoring stage

- Data models can be "opinionated"
 - Example: Arista Validated Designs (AVD): modeled after UCN Design Guides
 - Less explicit variables that **need** to be set
 - » They still *can* be changed

- Variables to define

- Physical network topology
 - » DCs, PODs and how they are connected
- Logical network variables
 - » Single network or multiple interconnected networks
 - » To overlay or not to overlay
- Network services
 - » L3LS (multicast) EVPN; DCI ("multi-site"), L2LS, MPLS EVPN; IPv4-VPN, IPv6 VPN
- Connected endpoints

```
fabric_name: DUAL_DC_FABRIC

# Enable vlan aware bundles
evpn_vlan_aware_bundles: true

# Select rfc5549 or ospf, not both
underlay_routing_protocol: ebgp
# underlay_ipv6: true
# underlay_rfc5549: true
# overlay_mlag_rfc5549: true

# Optional, enable multicast features
# underlay_multicast: true
# evpn_multicast: true
```


AVD Ansible Collection

- Modular design
 - eos_designs → generates structured config
 - eos_cli_config_gen → generates CLI **configuration** and **documentation**
 - eos_config_deploy_eapi or eos_config_deploy_cvp
 - eos_validate_state → validate operational state of Arista EOS devices
 - eos_snapshot → collect CLI commands and generate reports
- Work is underway to decouple the data model from Ansible
 - Easier integration with other automation tools as well

Testing stages

- The most tricky to get right
 - So tricky that these are often skipped
 - >> Sometimes generating new job opportunities
- Ensure the configuration result of the design input is valid
 - ~~Push the network wide change to your physical staging network~~
 - Push the network wide change to a virtual copy of your network
- Test if the control plane converges well
 - “Network Ready For Use” (NRFU) tests
 - >> Based on the input given in the data model
- Test if the data plane does what you expect it to do
 - Can also be done in a virtual copy of your network

Pre-deployment vs post-deployment testing

- Often make use of the same toolkit
 - Snapshotting tools
 - Validation tools
- Consistency between your lab and production environment is key
 - Hardware (or *good* simulations)
 - >> Software
- Data plane testing **always** happens post-deployment ;-)
 - Always keep monitoring real-time telemetry
 - No alerts? All is *probably* well
 - >> Unless your change involved the ticketing system and phone circuit

Let's see this in action?

- If time and the demo-gods permit...
- Want to replicate this yourself?

https://github.com/jorisc90/nlnog_demo