Introduction to Design Driven Automation

netboxlabs

Mark Coleman
CPSO
mcoleman@netboxlabs.com

- UK -> NL -> FR
- Computer Science with Maths
- Career started in games and has nosed dived to the deepest levels of infrastructure
- 3d graphics -> Backend Developer -> DevOps -> Cloud Native -> Infrastructure Automation

Everyone Uses NetBox

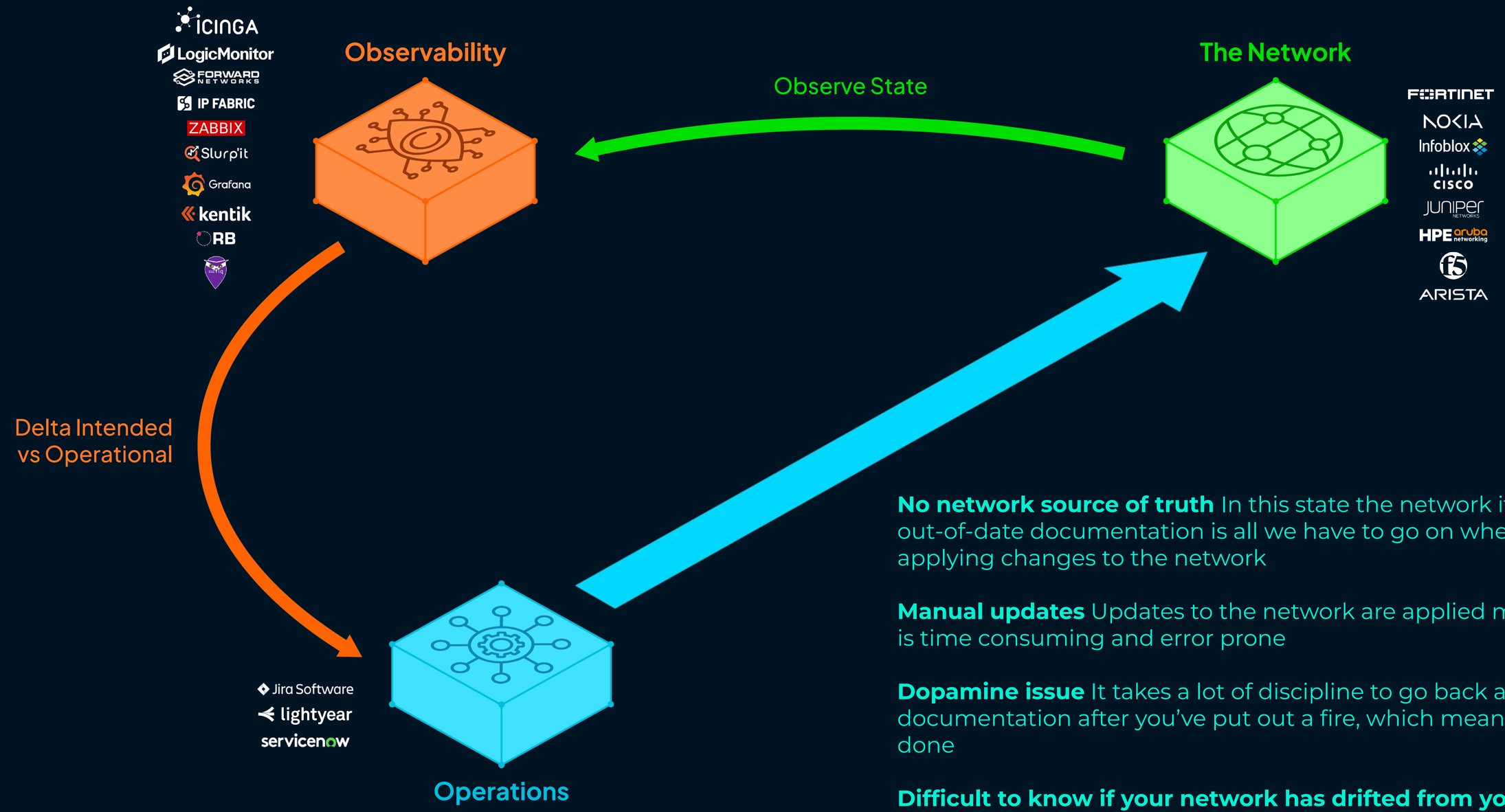
And 1000s more…

Intent Based Automation

First things first

Level 0

The infra is the truth and I don’t know what’s out there



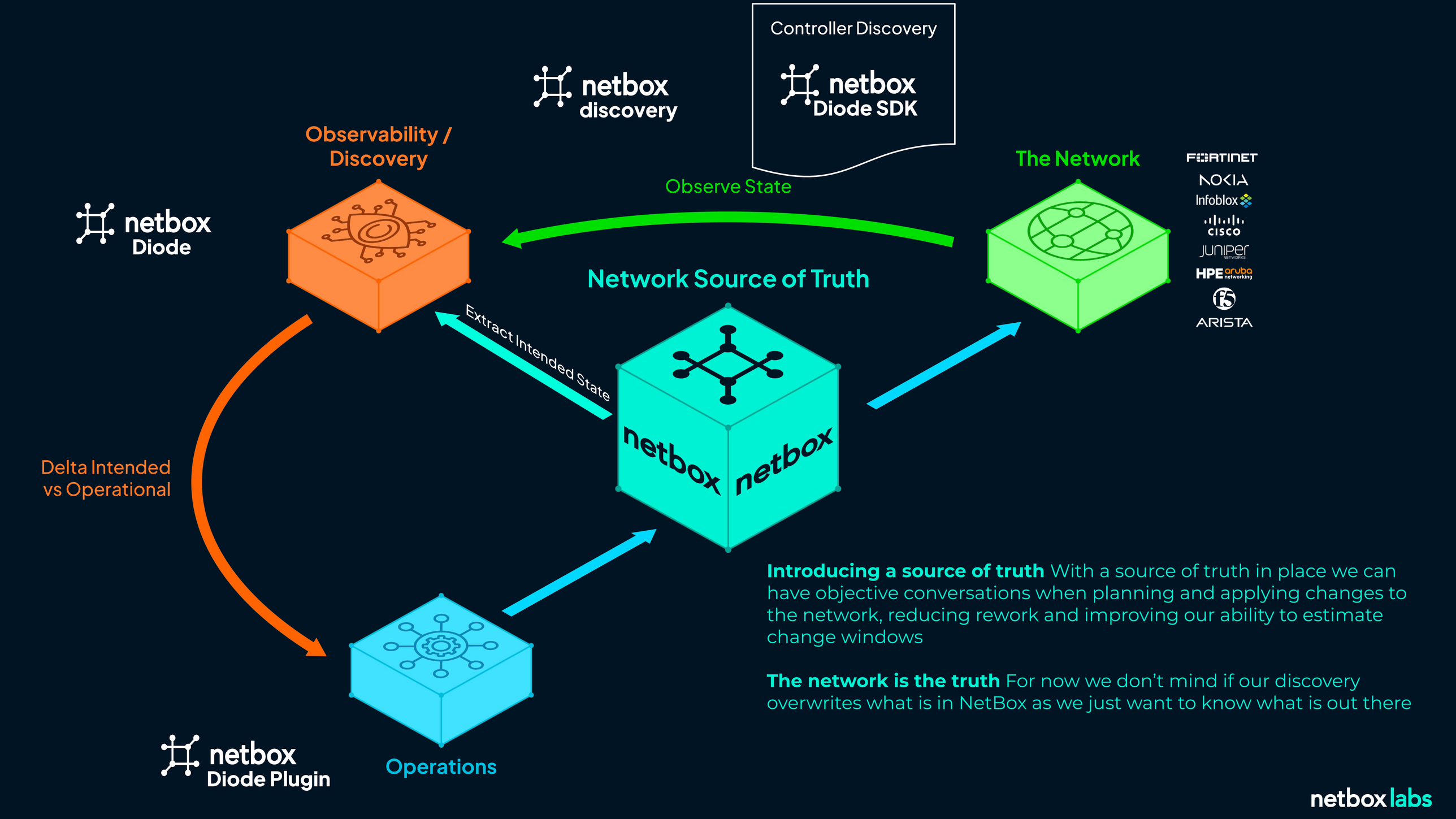
No network source of truth In this state the network itself, and often out-of-date documentation is all we have to go on when planning and applying changes to the network

Manual updates Updates to the network are applied manually which is time consuming and error prone

Dopamine issue It takes a lot of discipline to go back and update your documentation after you've put out a fire, which means it doesn't get done

Difficult to know if your network has drifted from your intent

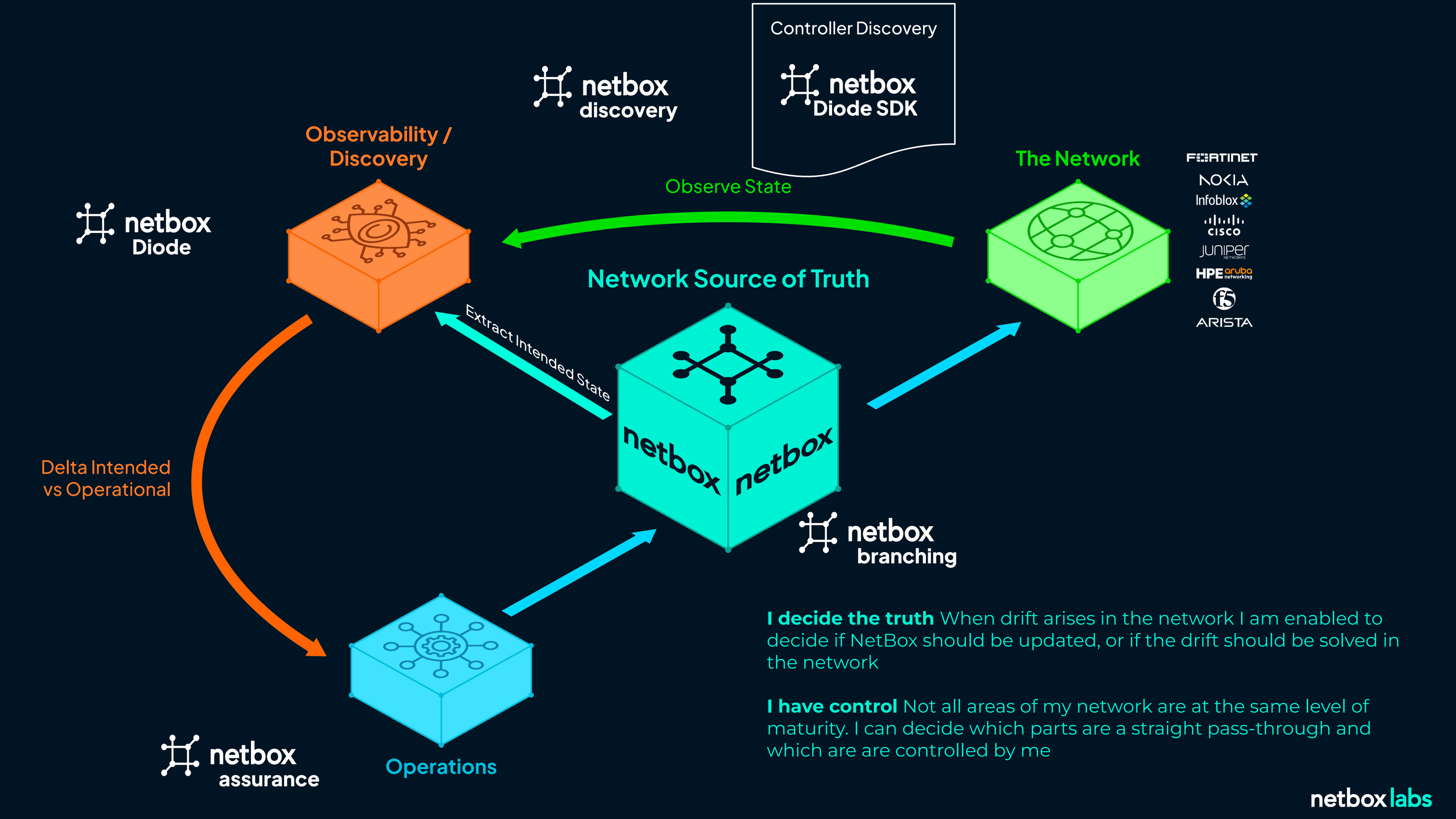
Level 1
The infra is the truth but I do know what is out there



Introducing a source of truth With a source of truth in place we can have objective conversations when planning and applying changes to the network, reducing rework and improving our ability to estimate change windows

The network is the truth For now we don't mind if our discovery overwrites what is in NetBox as we just want to know what is out there

Level 2
I have a baseline in my SoT that I must protect

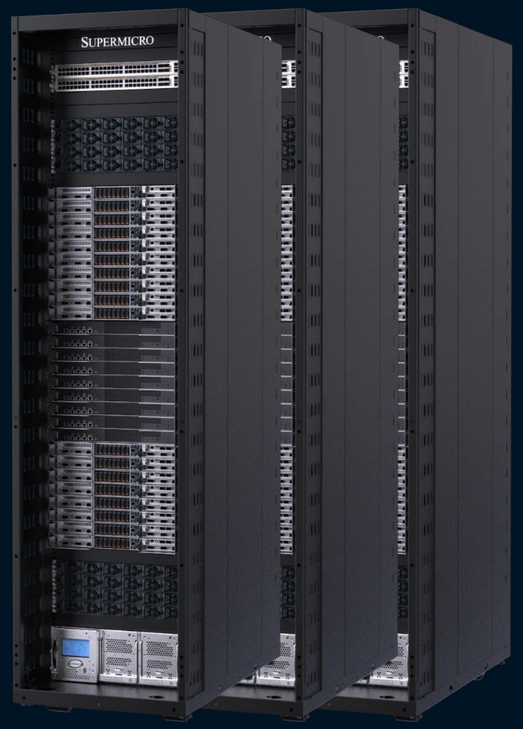


I decide the truth When drift arises in the network I am enabled to decide if NetBox should be updated, or if the drift should be solved in the network

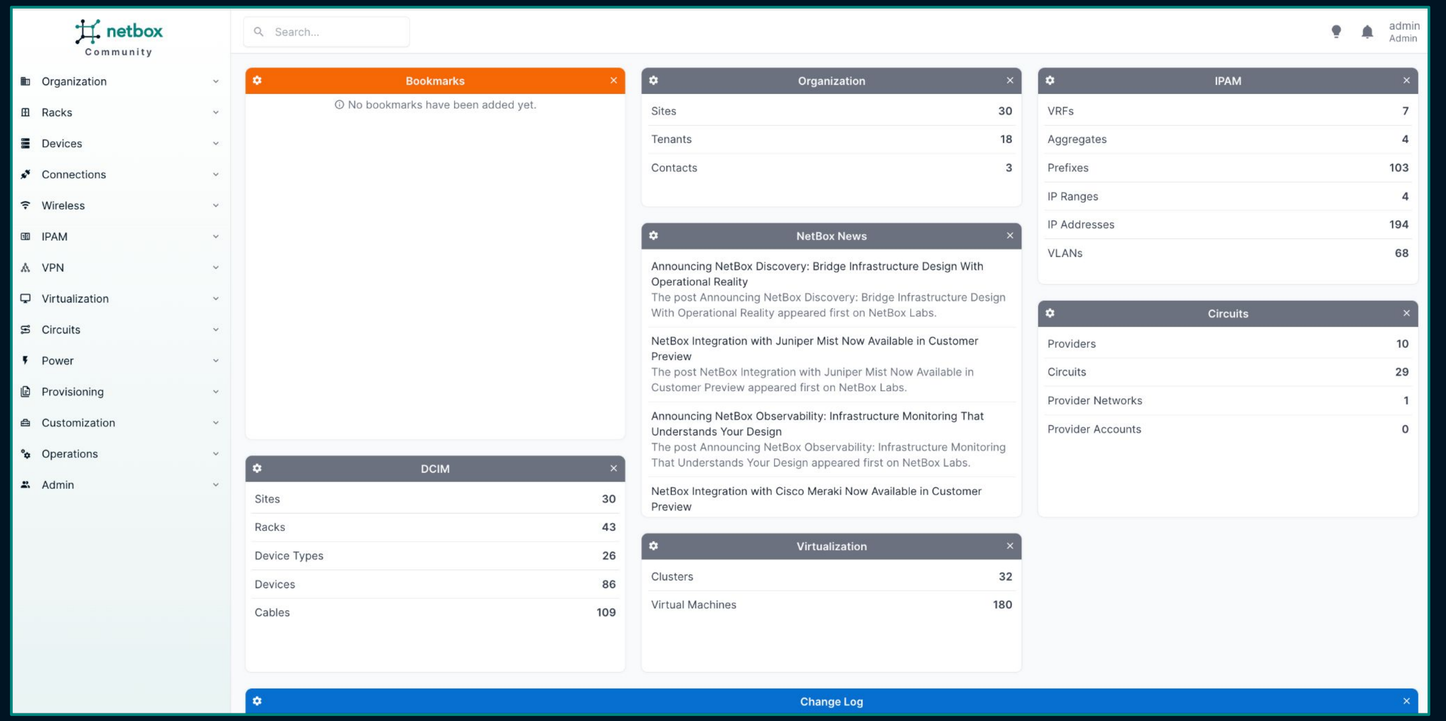
I have control Not all areas of my network are at the same level of maturity. I can decide which parts are a straight pass-through and which are are controlled by me

Design Based Automation
Scaling things up

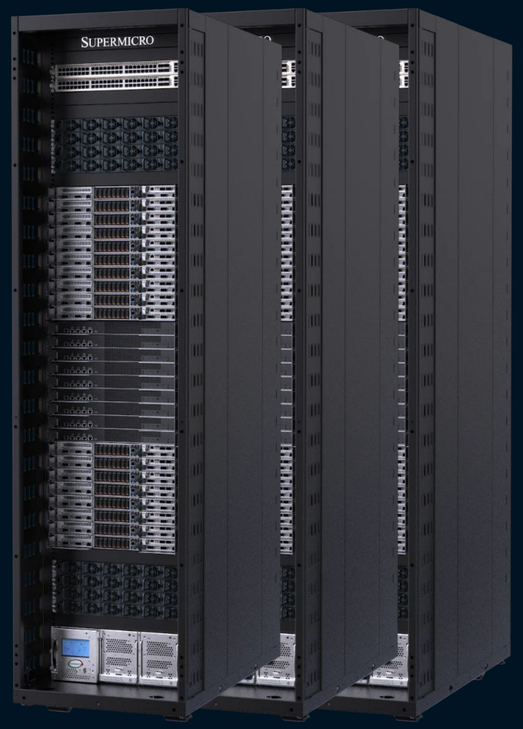
Being in control of intent is nice , but how do we express intent at scale?



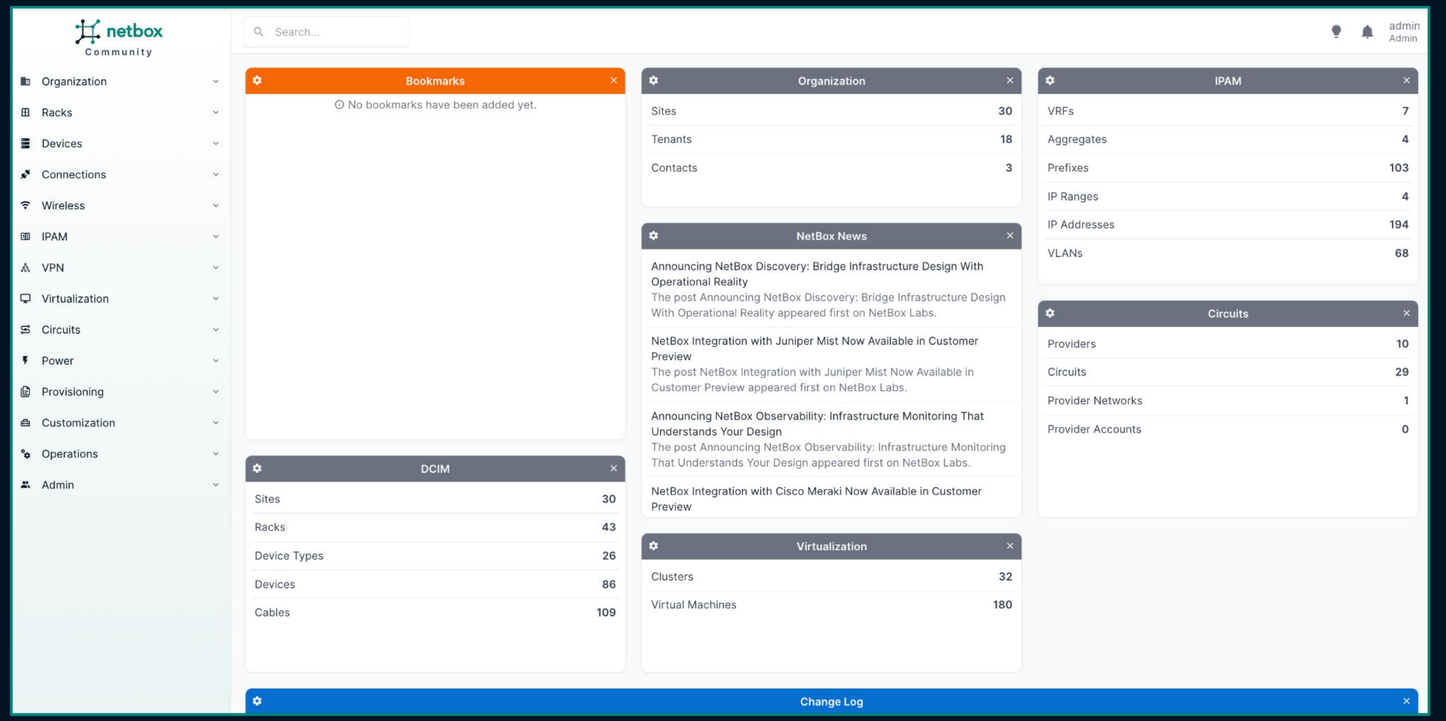
?



We already know how to think about this…



?



High-Level Design

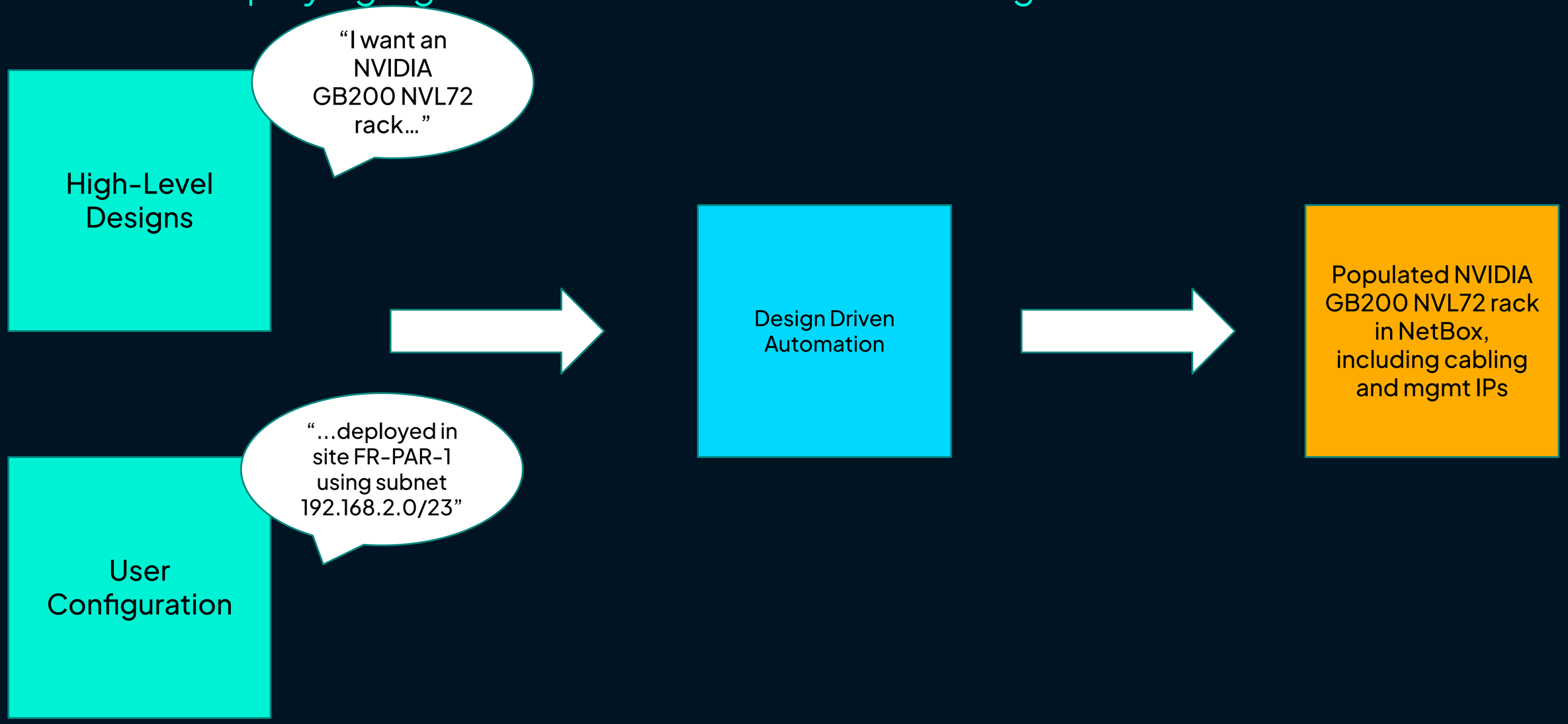
Low-Level Design

Everybody is building this themselves, but it is difficult to do correctly

- **Declarative** - Describe the High-Level Design and let the tooling figure out the plan
- **State Aware** - Don't want to hardcode IPs, ASNs etc. Designs should understand how to use rules to infer intent: next subnet from supernet, next IP from subnet, next ASN, etc
- **Composable** - Compose High-Level Designs from Design Components
- **Configurable** - Re-use the same High-Level Design across many projects, not hardcoded
- **Versioned** - Use GitOps style workflows for managing designs, track deployed designs against Low-Level Design state
- **Performant** - Users should be able to iterate on High-Level Designs in near real-time, very difficult to do without sprinkling some computer science on the solution

Real World Example

AI DCs are deploying a galactic amount of infrastructure right now



Design Composition

Device Type Library

Device Designs

Rack Designs

Pod Designs...

Device Types

Module Types

Rack Types

Cisco TOR w/ modules

Dell R640 with PCI module

Compute Rack

Pod Design

NetBox

Let’s see it already…
Demo time

Whole new world

Design Based Automation opens ups a lot of opportunities

High-Level Design intent makes much more possible

- **Out of the Box Test Suites** - Use HLD to derive Network Ready For Use (NRFU) and operational tests
    - Design includes cabling? Generate LLDP tests
    - Design includes BGP? Generate adjacency tests
- **Drive Asset Lifecycles from designs** - High-Level Design -> Bill of Materials -> POs -> Shipments -> Cutsheets-> Asset Reconciliation against LLD -> NRFU tests
- **Design Drift** - Concerned that people are making changes in the infrastructure that is deviating from design intent? Compare the HLD against the LLD again to get a drift report
- **Design Lifecycling** - Replacing all your TOR switches? Update the HLD and let it drive a deviation report

Thank you!

CONTACT US

Product Team
product@netboxlabs.com