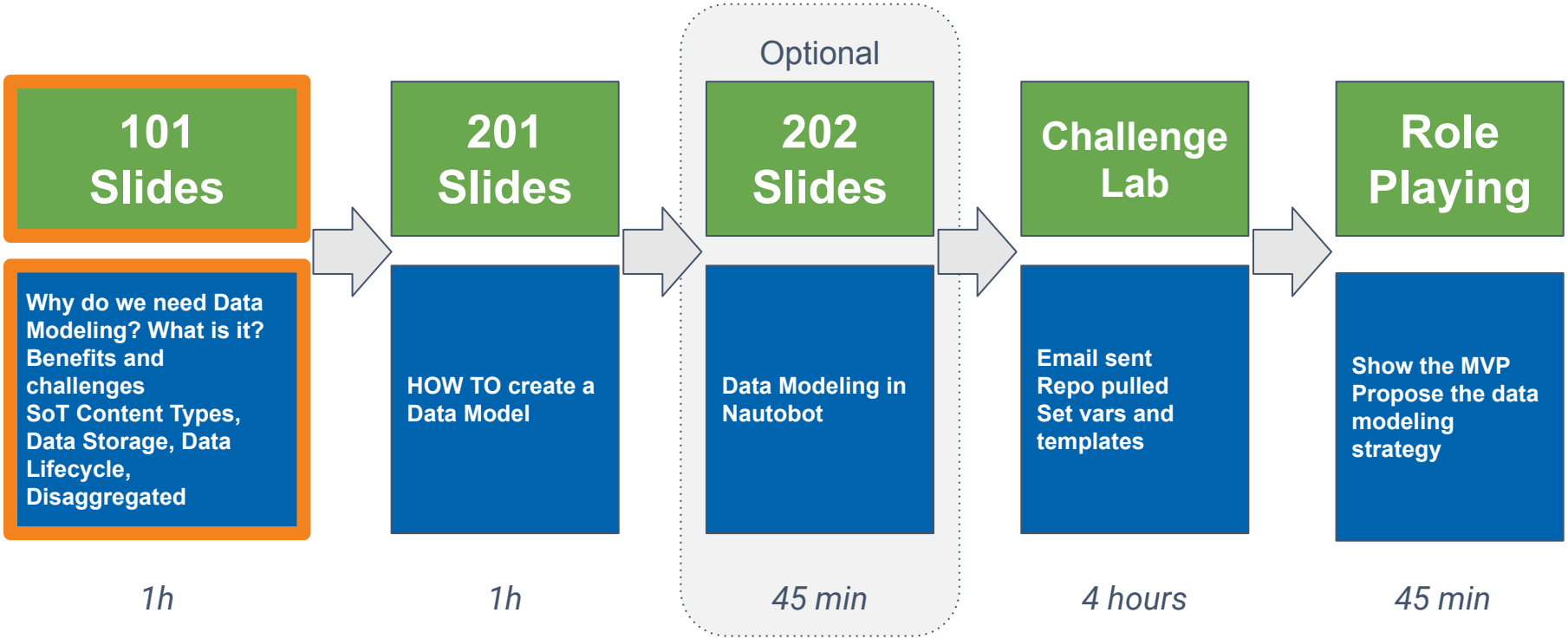

network .toCode()
101 - Data Modeling
August, 2022

Data Modeling Training Plan

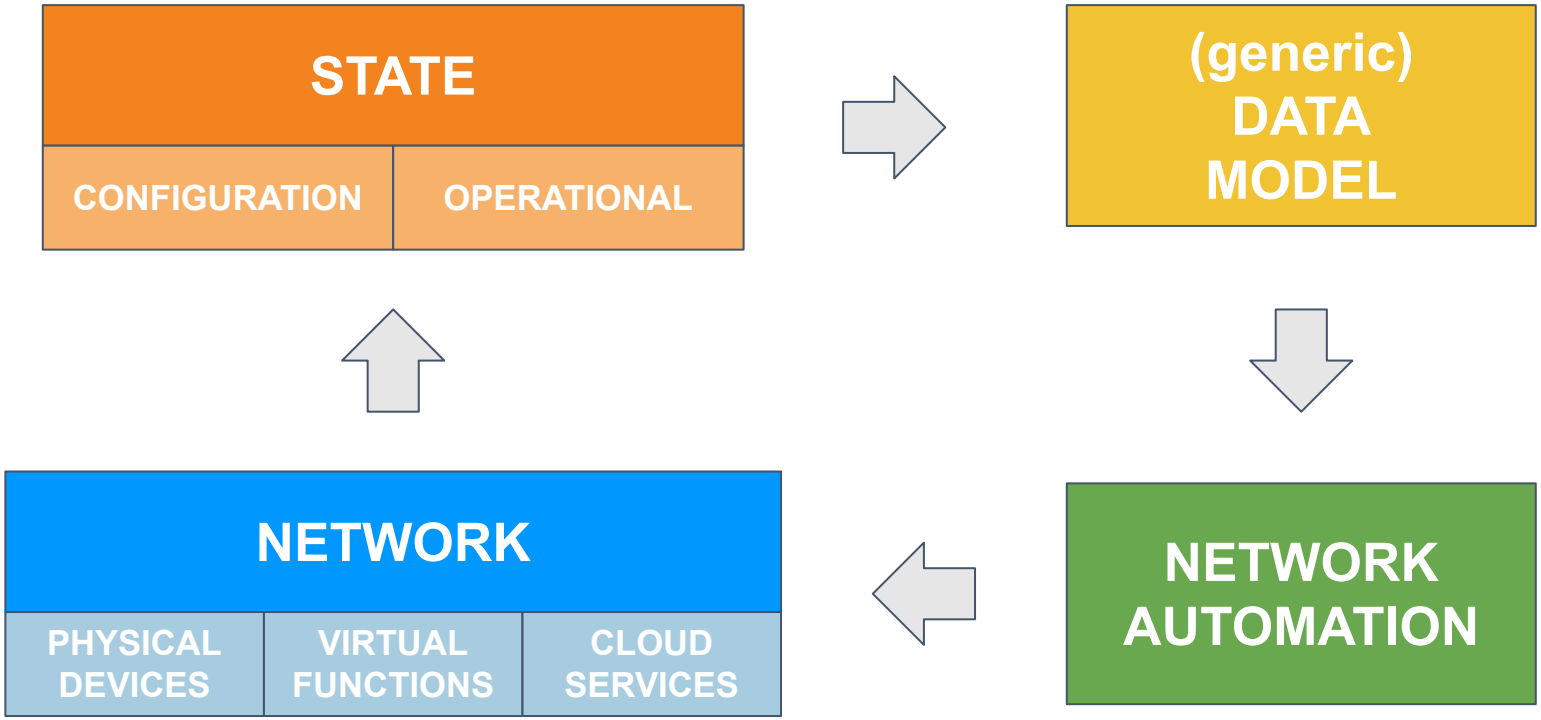


>>> Agenda

- Why do we need Data Modeling?
  - Source of Truth
- What is Data Modeling?
- Benefit of Data Modeling for Network Automation
- Challenges of Data Modeling

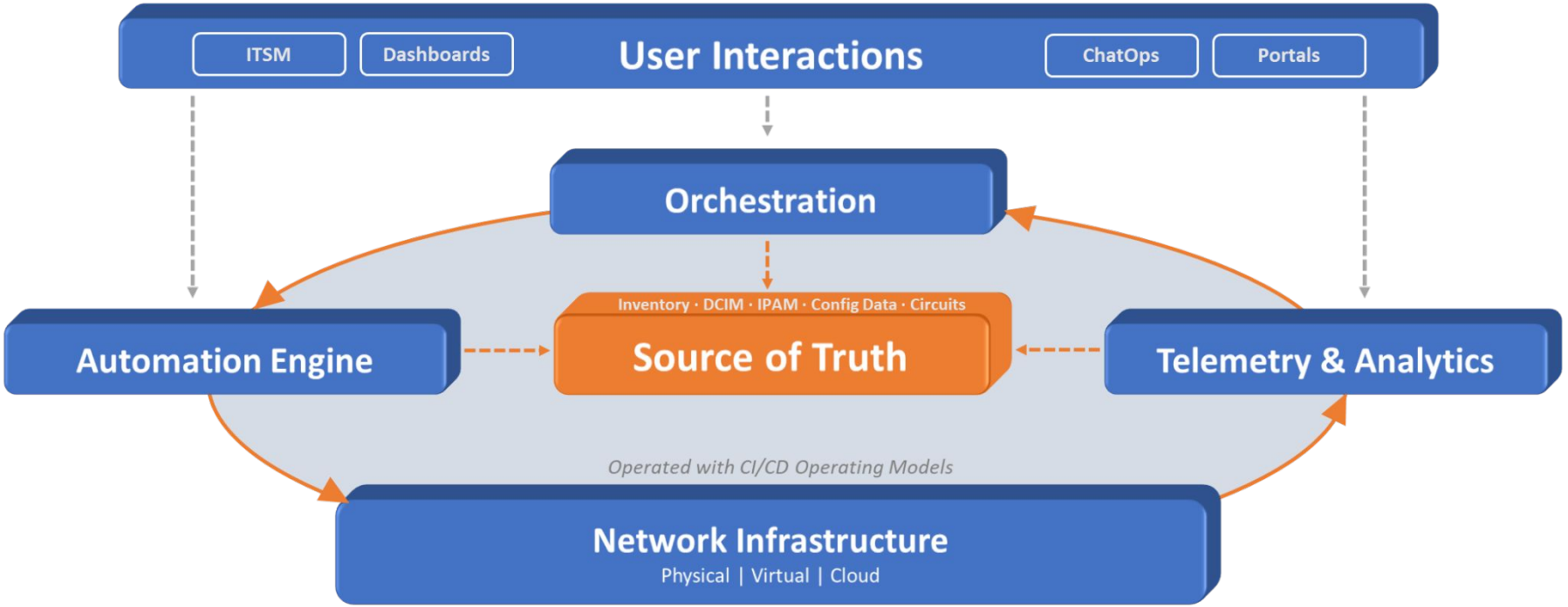
Why do we need Data Modeling?

>>> Why do we need Data Modeling?



>>> Our Network Data Model is stored in the Source of Truth

The role of Source of Truth as the driver and foundation

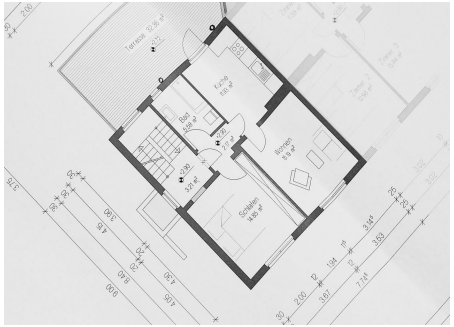


>>> Understanding the Source of Truth

Actual State

Your intended goal

Intended State In the Source of Truth



Source of Truth is all about **intentions** and planning, the expected state

All images are from Pixabay

>>> Which information could be stored in the SoT?

Inventory

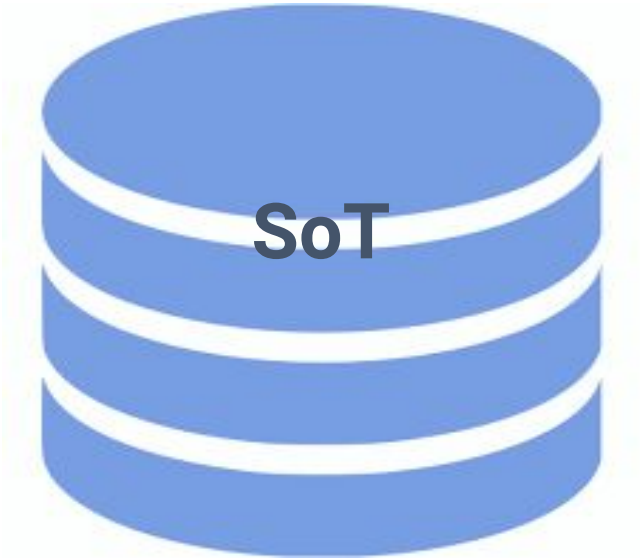
- Device List
- Software Version
- Hardware
- License

DCIM

- Location
- Racks
- Cables
- Power

IPAM

- IP Addresses
- IP Prefixes
- VLANs
- VRFs



STATE

CONFIGURATION | **OPERATIONAL**

Config Mgmt

- Templates
- Programmatic Interfaces

Circuits

- Circuits
- Providers
- Maintenances

Cloud Infrastructure

- Network Services

Network Properties

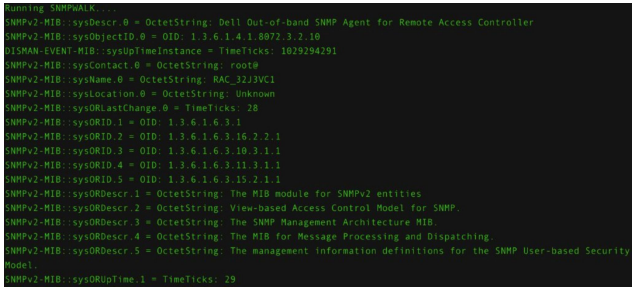
- NTP
- AAA
- DNS
- Routing
- ACLs

>>> How the network data can be represented?

CLI

```
interface Ethernet1
   description CIO Port
   switchport access vlan 100
   switchport mode access
   speed auto
   duplex auto
```

SNMP



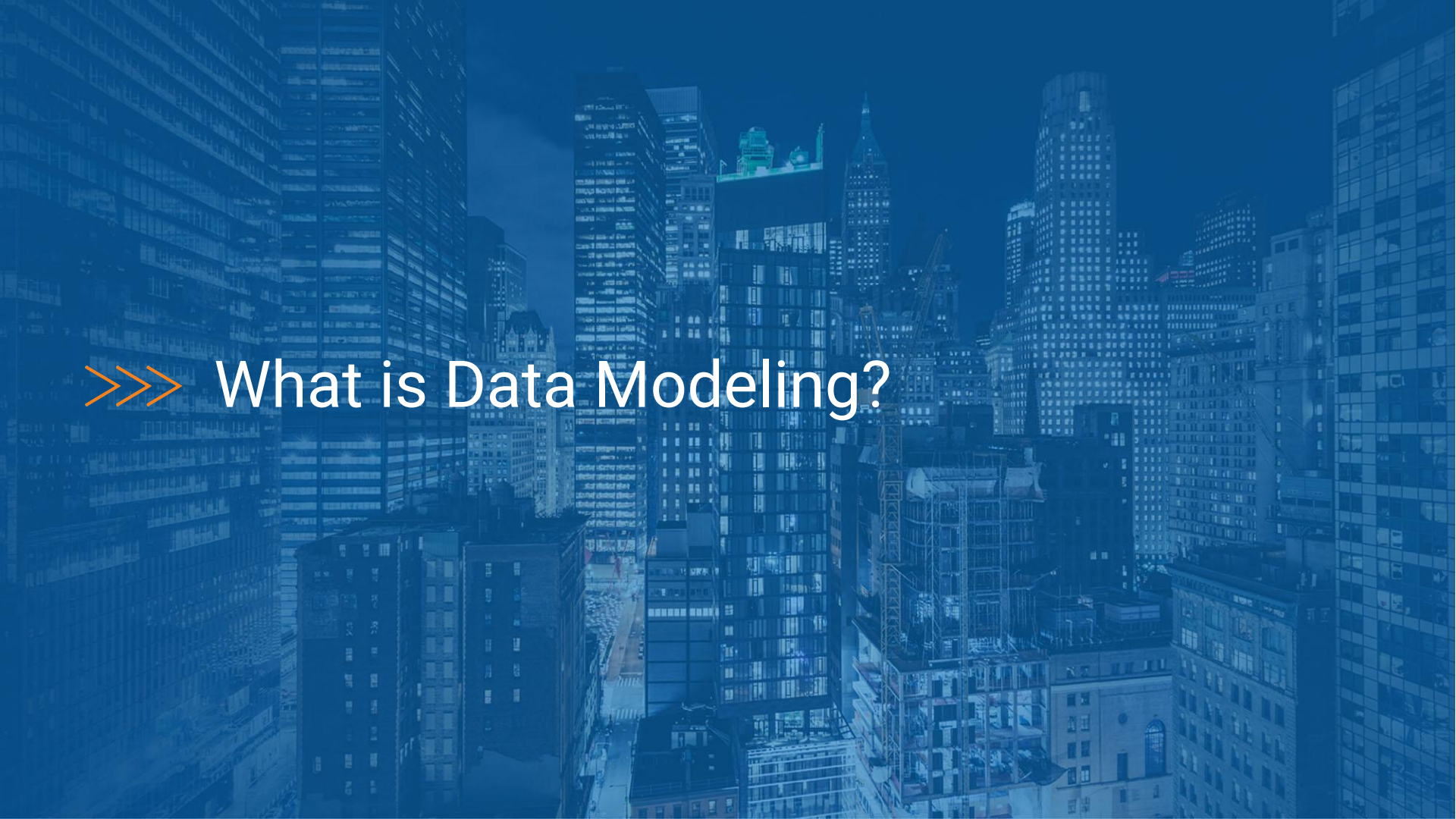
YANG

Custom APIs

```
https://ec2.amazonaws.com/?Action=CreateVpc
&CidrBlock=10.32.0.0/16
&InstanceTenancy=dedicated
&AUTHPARAMS
```

LINUX

```
[root@localhost ~]# clear
[root@localhost ~]# route add default gw 192.168.1.1
[root@localhost ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     *               255.255.255.0   U     0      0        0 eth0
default         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
[root@localhost ~]#
```



What is Data Modeling?

>>> Define Objects in a Structured and Abstract Way

What is Data Modeling?

*Which data attributes are **common** to all the objects, and **relevant**?*

Vehicle

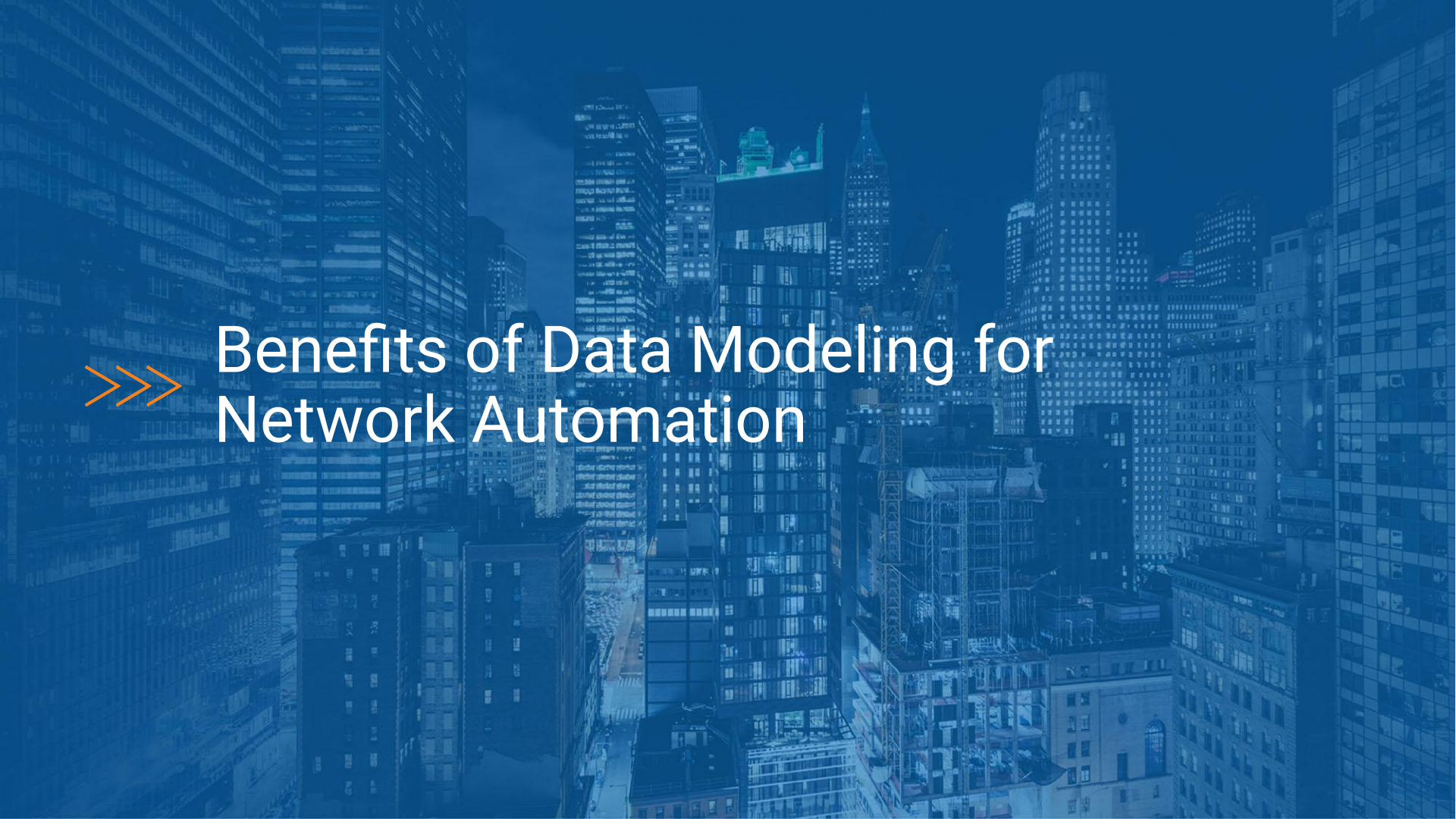
```
---
wheels: 4
headlights: 2
taillights: 2
doors: 2
make: Gio
model: Metro
fast: false
```

```
---
wheels: 2
headlights: 1
taillights: 1
doors: 0
make: Triumph
model: Tiger 1200
fast: true
```

>>> Key aspects of Data Modeling

ABSTRACT	A data model represents could represent different objects, with different characteristics, but with common attributes
IDENTIFIABLE	It should include a unique way to identify one object member of the model. I.e. the “name”, or the combination of “make” and “model”
COMPLETE	The model should provide all the information necessary to describe the objects. If not included directly in the model, it could come via extensibility.
CONCISE	Only the necessary attributes should be included to reduce complexity. Do not add information that is not going to be used.
STRUCTURED	The data model representation of an object should be able to be exchanged as structured data

There are multiple ways to represent a data model: JSON, XML, YAML, SQL DB, and others. From them, the most human friendly, especially for experimentation, it’s YAML



Benefits of Data Modeling for Network Automation

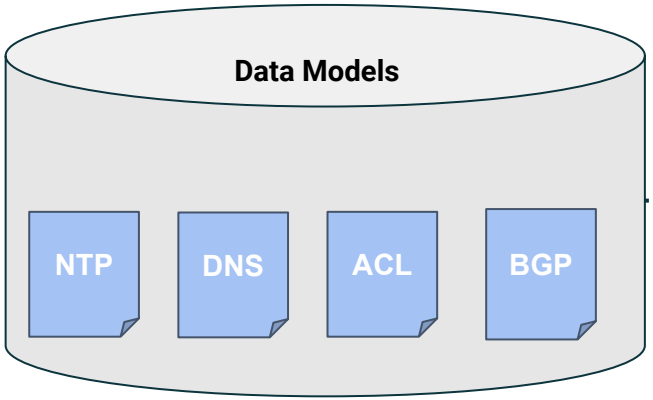
>>> Benefits of Data Modeling for Network Automation

1	**Vendor/Interface Agnostic**	• Portability between vendors. • Vendor implementations are handled within templates. • It helps to compare data due normalization
2	**Data Compression**	• Abstract use cases away from the data model. • Greater adoption by removing complexity.
3	**Data Validation**	• Agreed upon model for network features. • Ensures good inputs result in good outputs. • Extraneous data cannot impact configuration output.

>>> Vendor Agnostic/Interface Models

Benefits of Data Modeling

Using a common Data Model, we can get particular configuration for different vendors, and interfaces



CISCO

```
ntp server 203.0.113.44 prefer
ntp server 203.0.113.172
ntp source Loopback0
```

ARISTA

```
ntp server 203.0.113.44 prefer
ntp server 203.0.113.172
ntp local-interface loopback0
```

JUNIPER NETWORKS

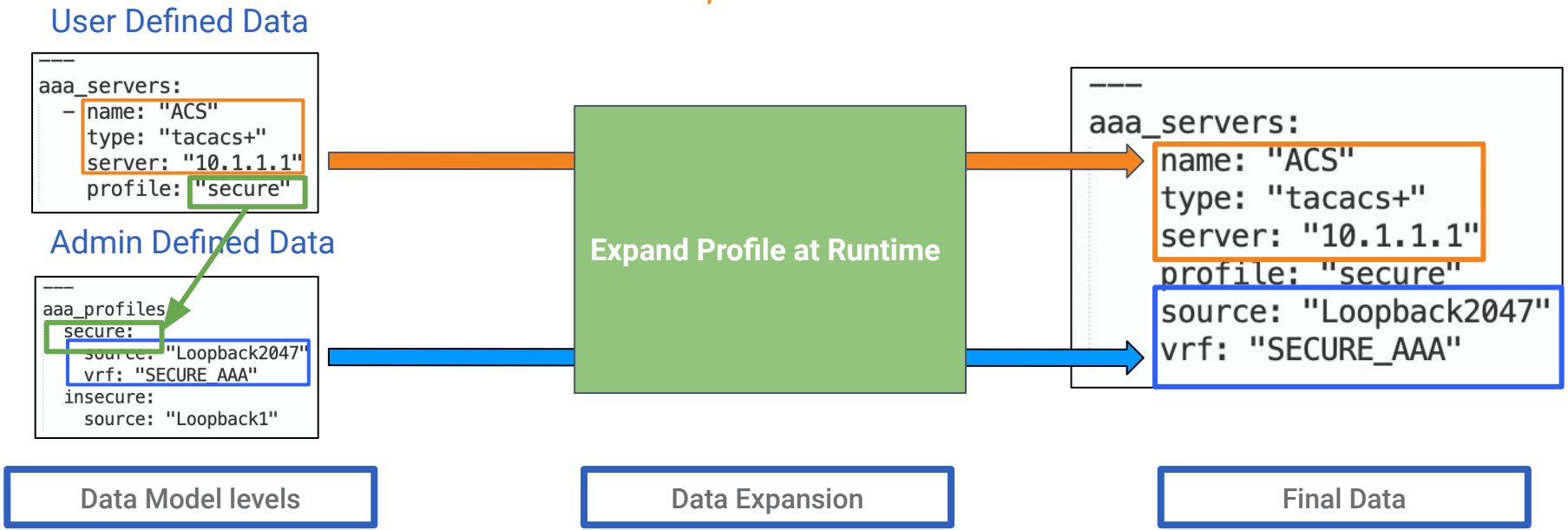
```
set system ntp server 203.0.113.44
prefer
set system ntp server 203.0.113.172
set system ntp source-address 192.0.2.1
```

Network Device Data | **Template Engine** | **Final artifact**

>>> Data Compression

Benefits of Data Modeling

Using different "levels" of intent, we can reuse Data Model references to simply them with extensibility options



>>> Data Validation

Benefits of Data Modeling

```yaml
---
$schema: "http://json-schema.org/draft-07/schema#"
$id: "schemas/aaa"
description: "AAA Configuration schema."
type: "object"
properties:
  aaa_servers:
    type: "array"
    items:
      type: "object"
      properties:
        name:
          type: "string"
        type:
          type: "string"
          enum: [“tacacs+”, “radius”]
        profile:
          type: "string"
          enum: [“secure”, “insecure”]
      required:
          - "name"
          - “type”
    uniqueItems: true
additionalProperties: false
```

Validation Logic
I.e. Schema Validation

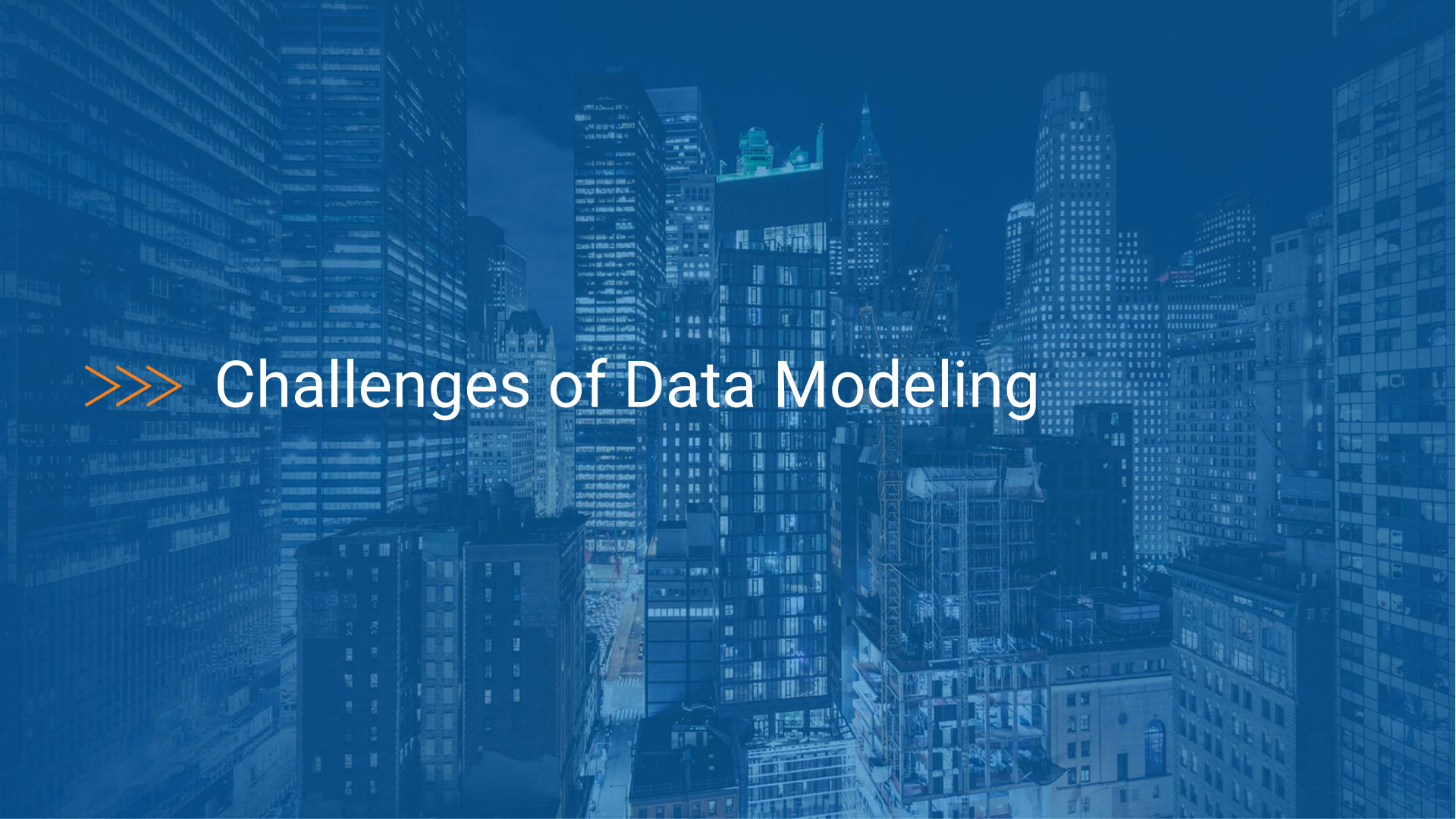
```yaml
# jsonschema: schemas/aaa
---
aaa_servers:
  - name: ACS
    type: tacacs+
    profile: secure
```

```yaml
# jsonschema: schemas/aaa
---
aaa_servers:
  - name: ACS
    type: tacacs+
    profile: not-secure
```

Data Objects

Enforcing a Data Validation criteria ensures data correctness.

Validation Result



Challenges of Data Modeling

>>> Challenges to create useful Data Models for our SoT

NORMALIZED & VALIDATED	Data should be normalized (it could represent different objects) and validation must be enforced
COMPRESSION	Defining the levels of intent across the design of a network based on roles, etc., in order to expose the right level of complexity
USABILITY	Providing the correct amount of abstraction to increase adoption of the data model, and reusability
STORAGE	Depending on the underlying technology to store the data models, there would be different pros and cons, and this will impact how to interact with this data
CONSISTENCY	Data should be consistent, even in distributed environments where multiple System of Records exist, to provide only one valid interpretation
LACK OF STANDARDS	Not having a good reference of the outcome, it will add complexity managing exceptions

>>> Lack of Configuration Standard

```
interface GigabitEthernet1/0/1
  description To ESX-01-p01
  switchport mode trunk
  channel-group 1 mode passive
  switchport trunk encapsulation dot1q
  switchport trunk native vlan 10
  switchport trunk allowed vlan 10,20,30,40,55-60
interface GigabitEthernet1/0/2
  description To ESX-01-p02
  switchport mode trunk
  channel-group 1 mode active
  switchport trunk native vlan 20
  switchport trunk allowed vlan 10,20,30,40,300
  spanning-tree portfast trunk
```

- From all the challenges, the most important one to tackle, if possible, is the lack of a consistent network configuration.
- Without it, the effort to **model the network** state will always be a pain, and the outcome would be **suboptimal**.
- Sometimes it can't be changed, but any effort in this direction will ease the network automation process.

>>> Disaggregating Configuration from Data Model

Complexities of Data Modeling

```
interface Ethernet1
  switchport access vlan 150
  switchport mode access
  authentication open
  authentication order dot1x mab
  authentication priority dot1x mab
  authentication port-control auto
  authentication periodic
  authentication timer reauthenticate server
  authentication violation replace
  mab
  dot1x pae authenticator
  dot1x timeout tx-period 8
  spanning-tree portfast
```

```
interface:
 - name: Ethernet1
   vlan: 150
   switchport_mode: access
   authentication: open
   authentication_order: [dot1x, mab]
   authentication_priority: [dot1x, mab]
   authentication_port-control: auto
   authentication_periodic: true
   authentication_timer_reauthenticate: server
   authentication_violation: replace
   mab: True
   dot1x_pae: authenticator
   dot1x_timeout: tx-period 8
   spanning-tree_portfast: True
```

```
interface:
 - name: Ethernet1
   vlan: 150
   dot1x: True
   spanning-tree_portfast: True
```

Don't map one-for-one configuration, provide the intention in the data of the configuration


>>> network .toCode()
Characteristics of the Source of Truth


SoT Content Types

>>> Intended State, stored in a **Source of Truth**

Inventory

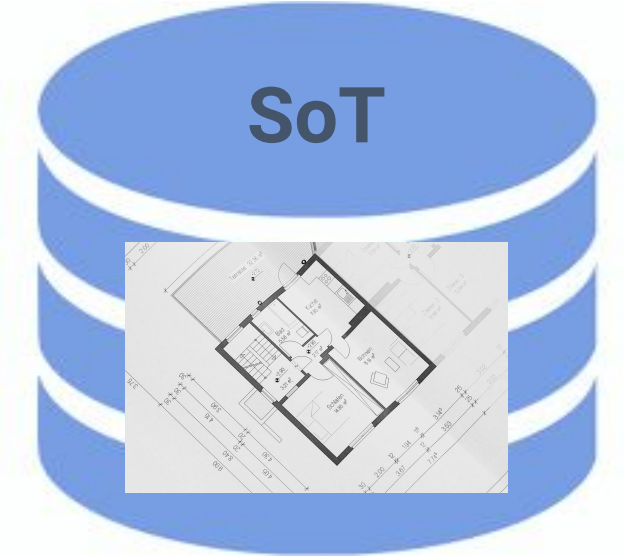
- Device List
- Software Version
- Hardware
- License

DCIM

- Location
- Racks
- Cables
- Power

IPAM

- IP Addresses
- IP Prefixes
- VLANs
- VRFs



Config Mgmt

- Templates
- Programmatic Interfaces

Circuits

- Circuits
- Providers
- Maintenances

Cloud Infrastructure

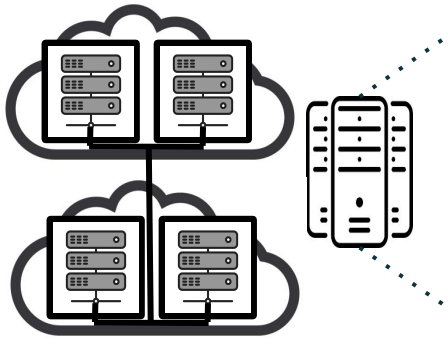
- Network Services

Network Properties

- NTP
- AAA
- DNS
- Routing
- ACLs

>>> SoT - Device Inventory

- **Network device inventory is the main foundation of any network automation platform**
- Allows automation based on:
  - Device type, location, service, internal grouping, etc.

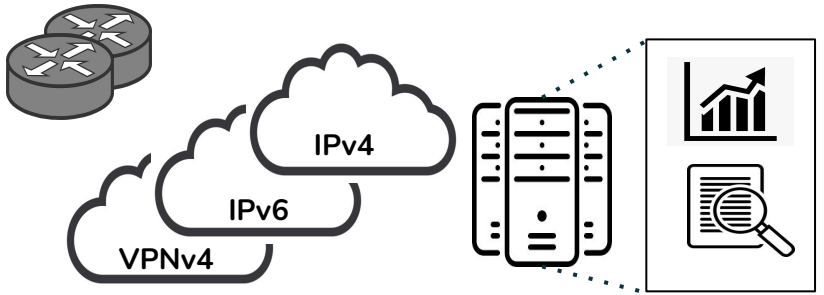


- Geographic Location
- Site Name
- Suite/Room ID
- Rack ID
- Rack Unit Position
- Cable Type
- Connectors
- Circuit Reference
- Floor Plan
- Power Feed
- Console Cables

Inventory Field	Description
Name	Device name, ideally in simple naming convention
Location	Device location including region, site, location (room, aisle, rack)
Platform	Device operating system - this determines the connection driver/protocol
IP address/DNS name	Management IP address and/or DNS name of the device
Status	Device status, e.g. active, offline, maintenance
Role	Function of the device within the network, e.g. leaf, spine, core aggregation
Type	Device hardware type, e.g. Cisco 3750 switch
Components	Components of an inventory elements, e.g. Card XYZ

>>> SoT - IPAM

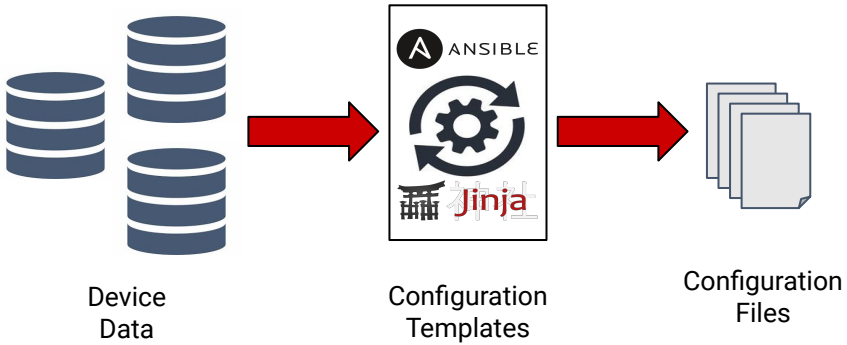
- IPAM
  - IP address management
  - Prefix and VLAN management



IPAM Models	Description
IP Address	IP address, along with related Device, Interface, Circuit
Prefix	IP Prefix information, along with related Routing Protocols Information
VRF	VRF information, along with tenant
IP address/DNS name	Management IP address and/or DNS name of the device
VLAN	Vlan information

>>> SoT - Configuration templates

- Configuration templates:
  - Contain references to the data values rendered
  - Contain discrete data values forming the configuration
- Configuration templates should consider the proper organizational grouping

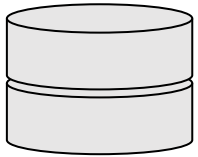


Element	Description
Vendor, Manufacturer	Configurations differ between vendors
Device Platform	Configurations differ between different platforms (e.g. IOS vs IOS-XE)
Device Role	Configurations differ between device roles (e.g. leaf and spine)
Device Type	Configurations differ between different hardware (e.g. stack provisioning)

Note: More to come in section “Levels of Intent”


SoT Data Storage

>>> Data Storage - Comparison

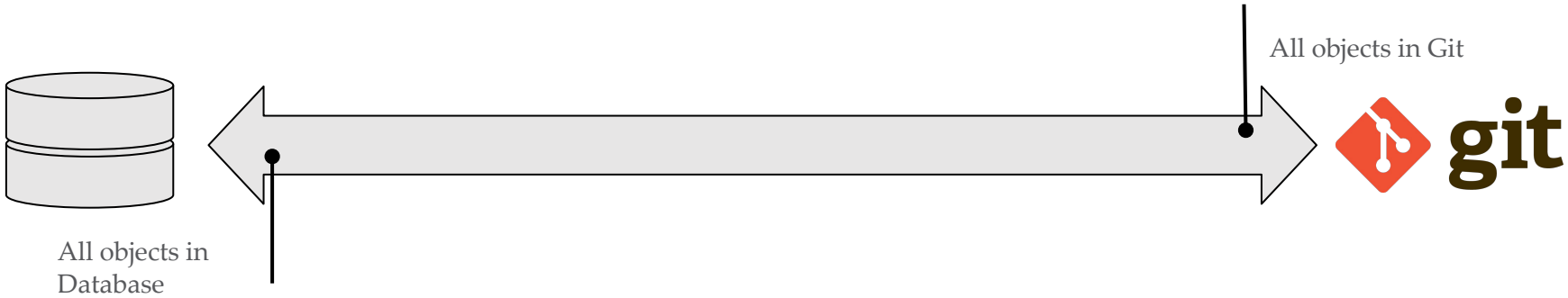


	Database	GIT
Store and organize intended state of the network	+++	+
Support ACID properties, consistency & atomicity	+++	+++
Provide traceability, history of the SoT	+	+++
Ease of access to the data	+++	+
Scalability of data	+++	+
Collaboration	+	+++

>>> Data storage - Volatility of Data

Type	Example	Location
Slow Moving Data	Global Config	Git
Large Quantity of Data	Interfaces, VLANs, Inventory	DB
Complex Data	Routing Protocols	DB/Git

>>> Git vs Database



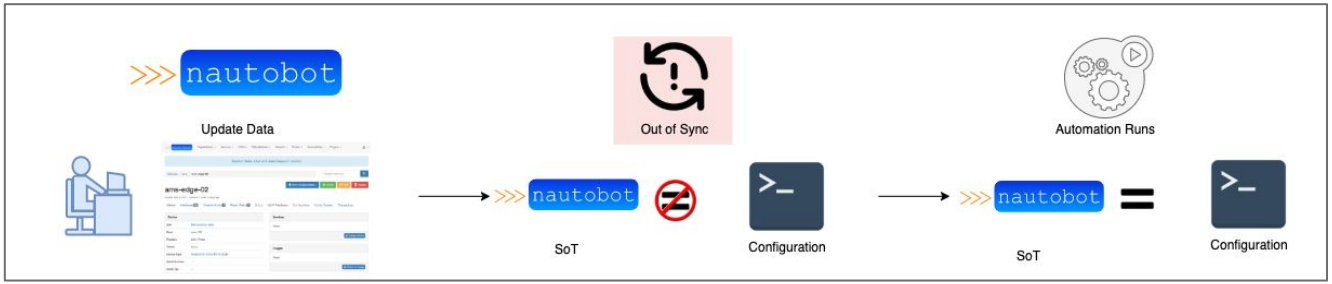
At the extremes, neither are likely to best for the customer, so consider what makes sense **for each SoR**

Some day, we won't need to choose one or another, and we could get a Git experience using DB objects. E.g. Dolt Database. But, we are not still 100% production ready, right now.

SoT Data Lifecycle

Data Lifecycle

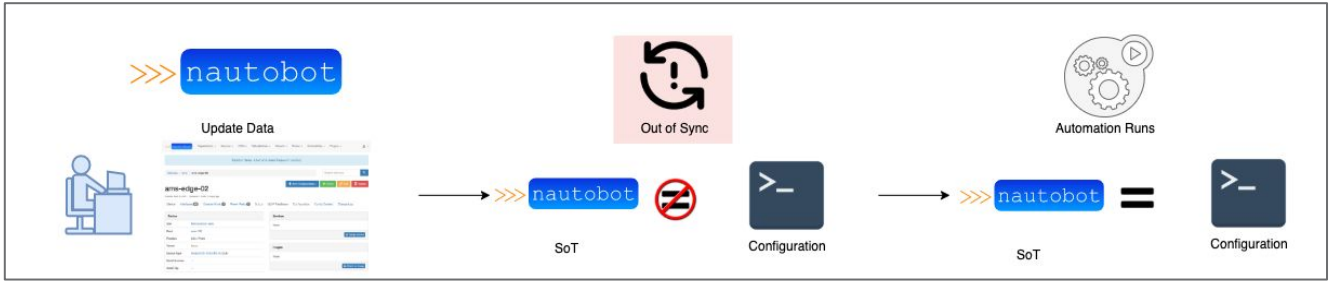
Before



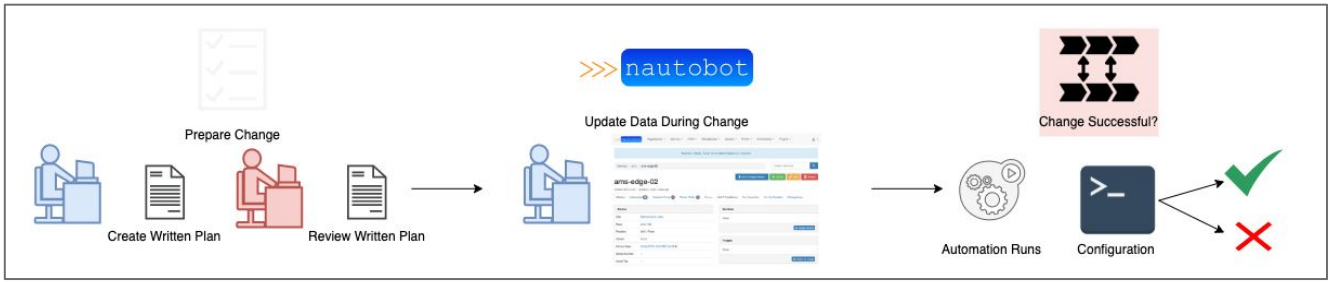
When you update the data **before the change**, you can test the expected behavior in advance, but the data is out of sync with the actual state for a while, leading to **mistrust in the data** if the procedure is not deterministic, and short, enough

Data Lifecycle

Before



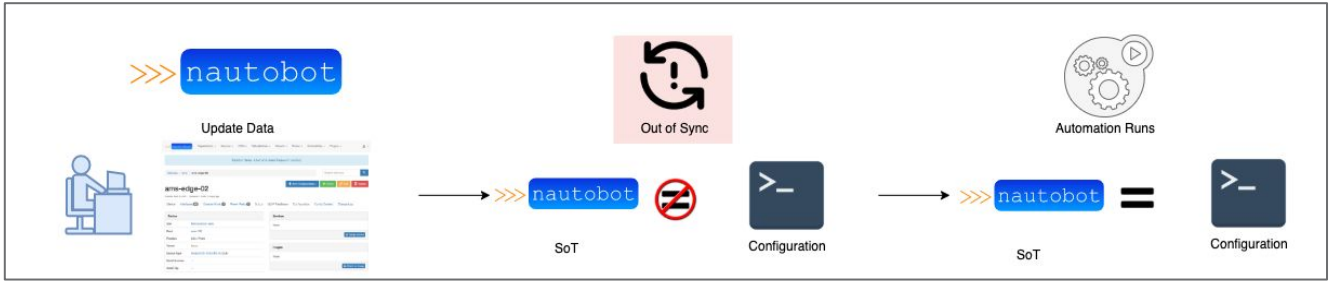
During



When you update the data **during the change**, you risk not testing the data beforehand, and **do not have confidence in change success**

Data Lifecycle

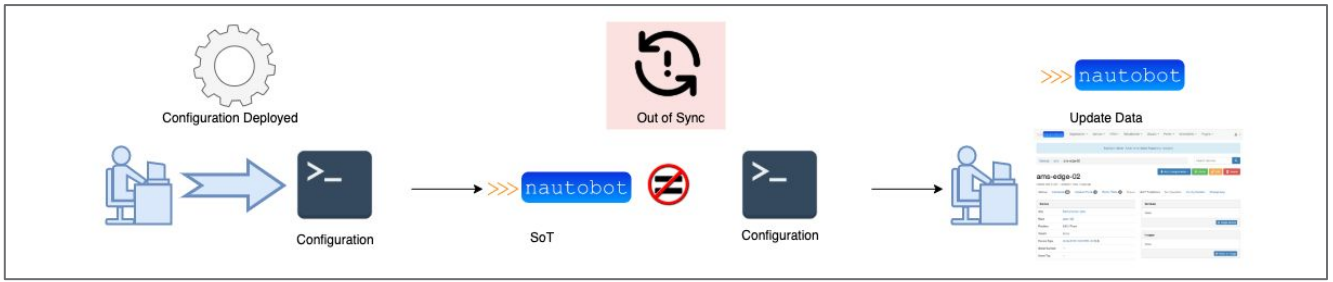
Before



During

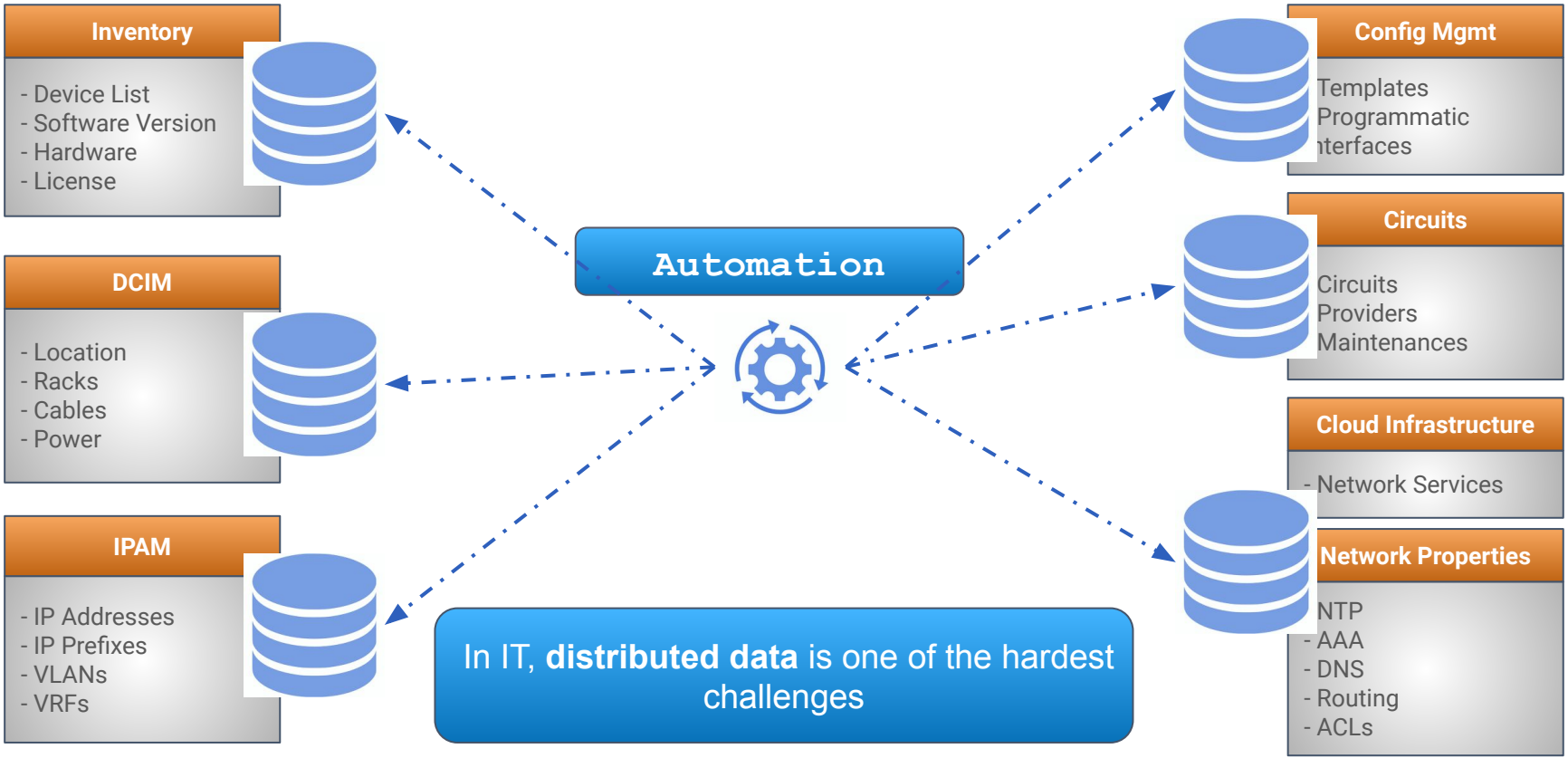
When you update the data **after the change**, the data is now a **reporting tool** and likely to be out of sync, leading to **mistrust in the data**

After




Multiple Sources of Truth

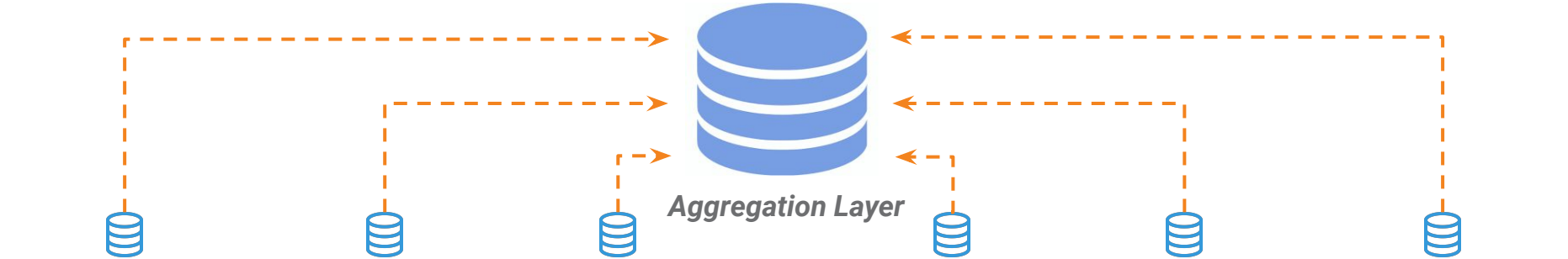
System of Records



>>> Aggregation

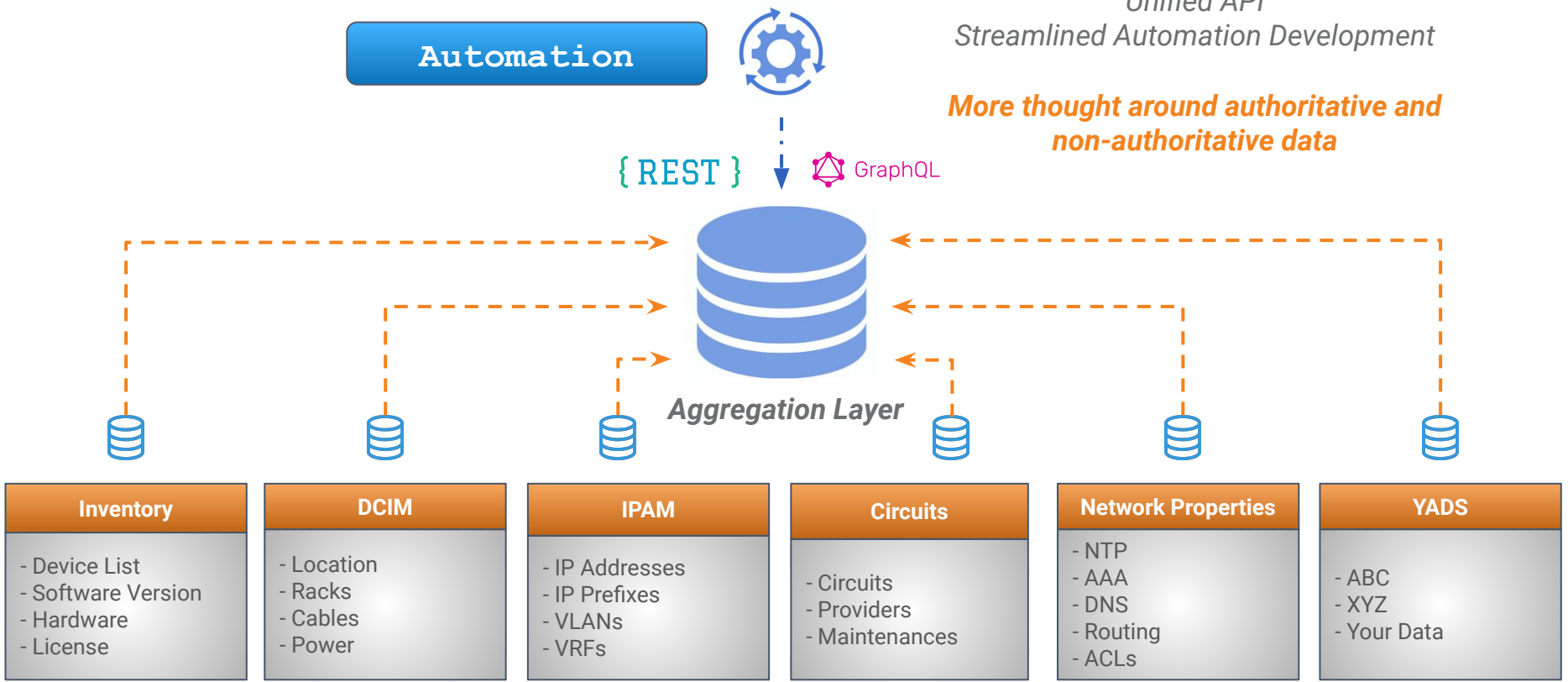
What if each data type was its own database?

What is the trusted data source?
What is AUTHORITATIVE?

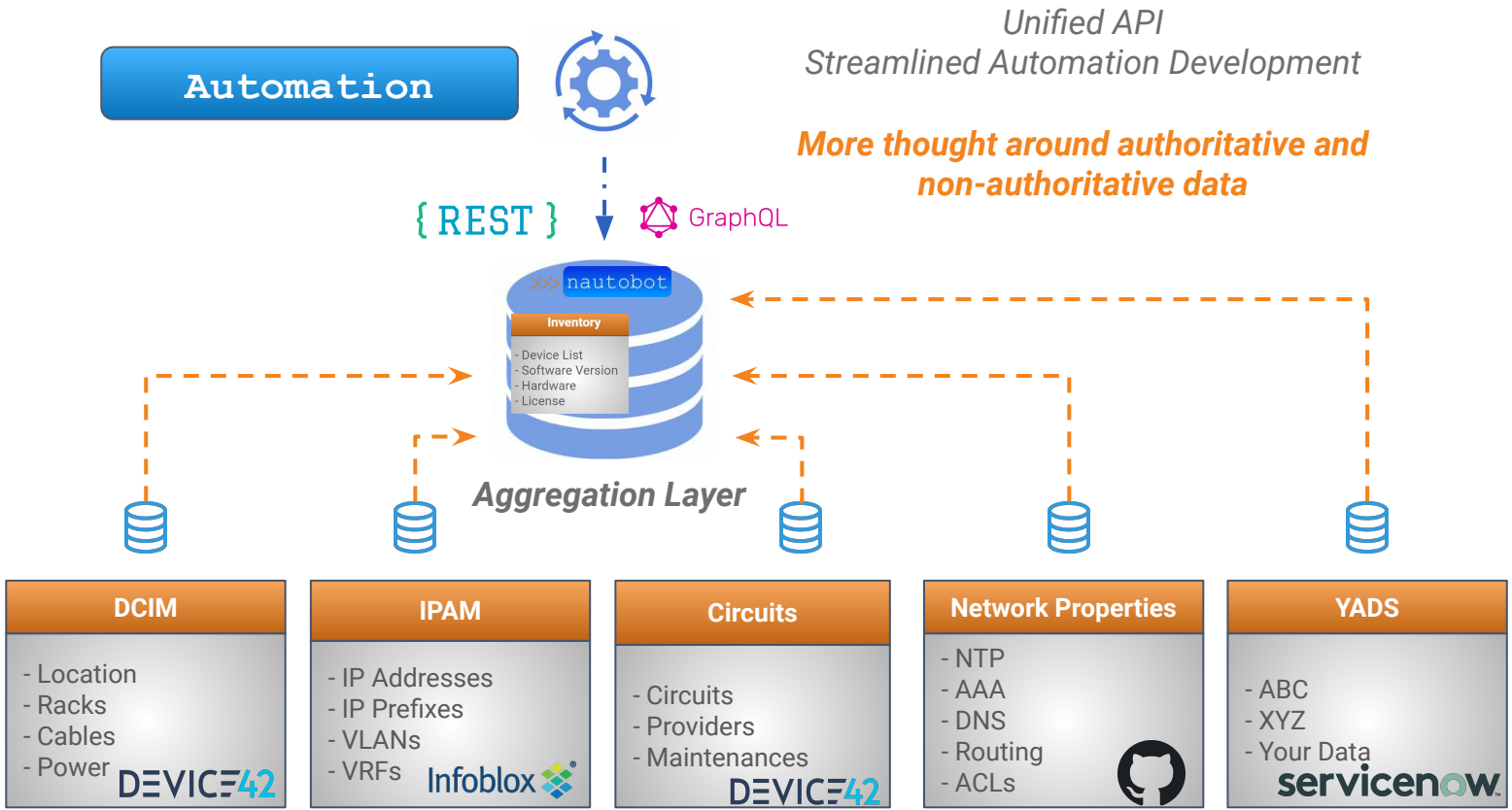


Inventory	DCIM	IPAM	Circuits	Network Properties	YADS
- Device List - Software Version - Hardware - License	- Location - Racks - Cables - Power	- IP Addresses - IP Prefixes - VLANs - VRFs	- Circuits - Providers - Maintenances	- NTP - AAA - DNS - Routing - ACLs	- ABC - XYZ - Your Data

>>> Why Aggregate?



Why Aggregate?



>>> Source of Truth - Aggregation & Synchronization

Why	What
Humans copying data is **error prone**	**Aggregate** multiple Systems of Record
Present a **consistent view** of the data to the automation tools	**Abstract** the different databases and interfaces
Clear on where data should be edited - **fail fast**	Copy data **programmatically**

>>> Aggregation

Source of Truth

01	**Aggregate multiple Systems of Record (SoR) for their specific data type.**	• Collect data from multiple sources
02	**Abstract the different databases and interfaces.**	• Normalize the data collected from multiple sources • Enrich and abstract the data
03	**Present a consistent view of the data to the automation tools.**	• Present a consistent view of the data sourced from multiple systems • Allow to access for data in one place
04	**Delegation and Ownership of data is challenging**	• Multiple data instances (data duplication) • Data governance and ownership

>>> Which are the challenges?

SUBNET
IPAM A

cidr 2001:DB8::/32
family 6
vrf vrf-blue
vlan VLAN123
customer_id abc

How can I load the data?

What is the difference?

How could we compare vlan name and vlan id?

How can I synchronize the data?

PREFIX
IPAM B

network 2001:DB8::
prefix_length 32
vrf vrf-blue
vlan_id 123
tenant abc

>>> Which are the challenges? - Examples

What is a cable?

Ethernet == 1 physical cable
Fiber == tx and rx port?
OR
Fiber_1 == tx_port?
Fiber_2 == rx_port?

What do you apply OSPF to?

A network?
An interface?
Different vendors may have alternate opinions

How do you store an IP?

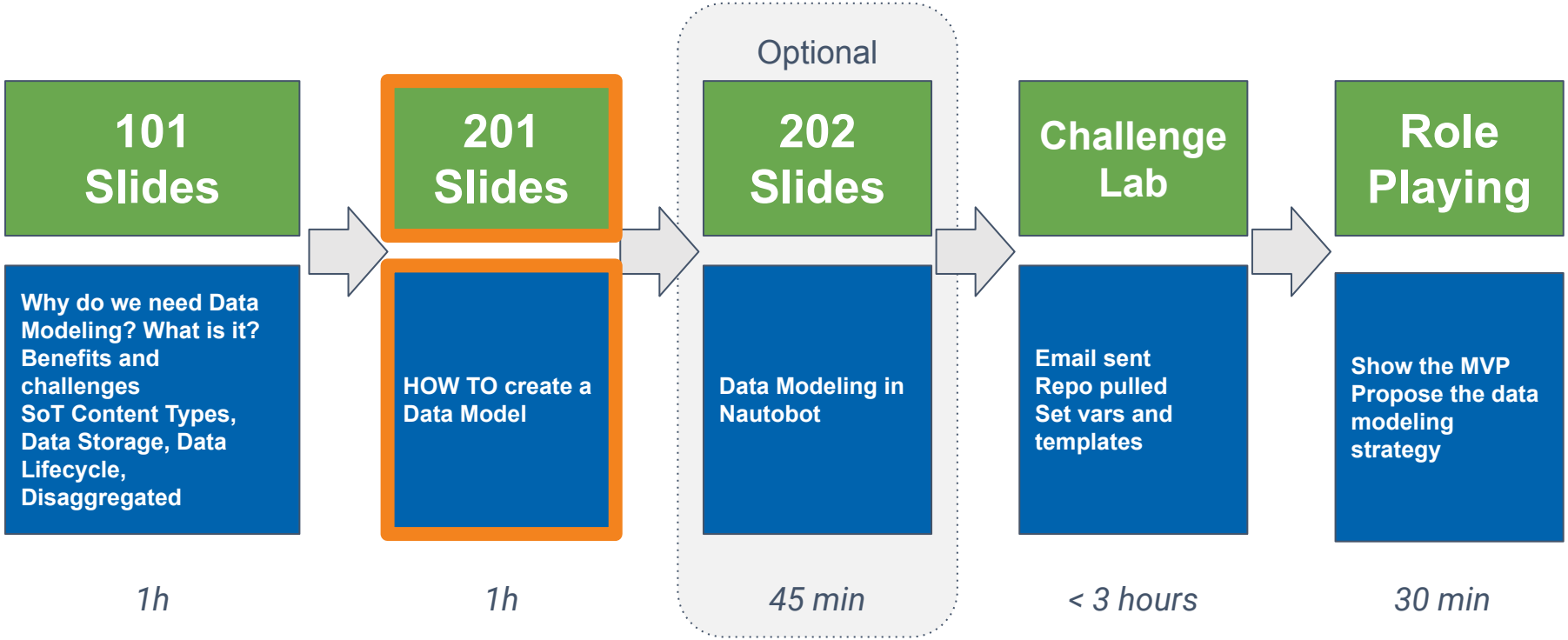
As an integer?
As a string?
With cidr?
With subnet mask?

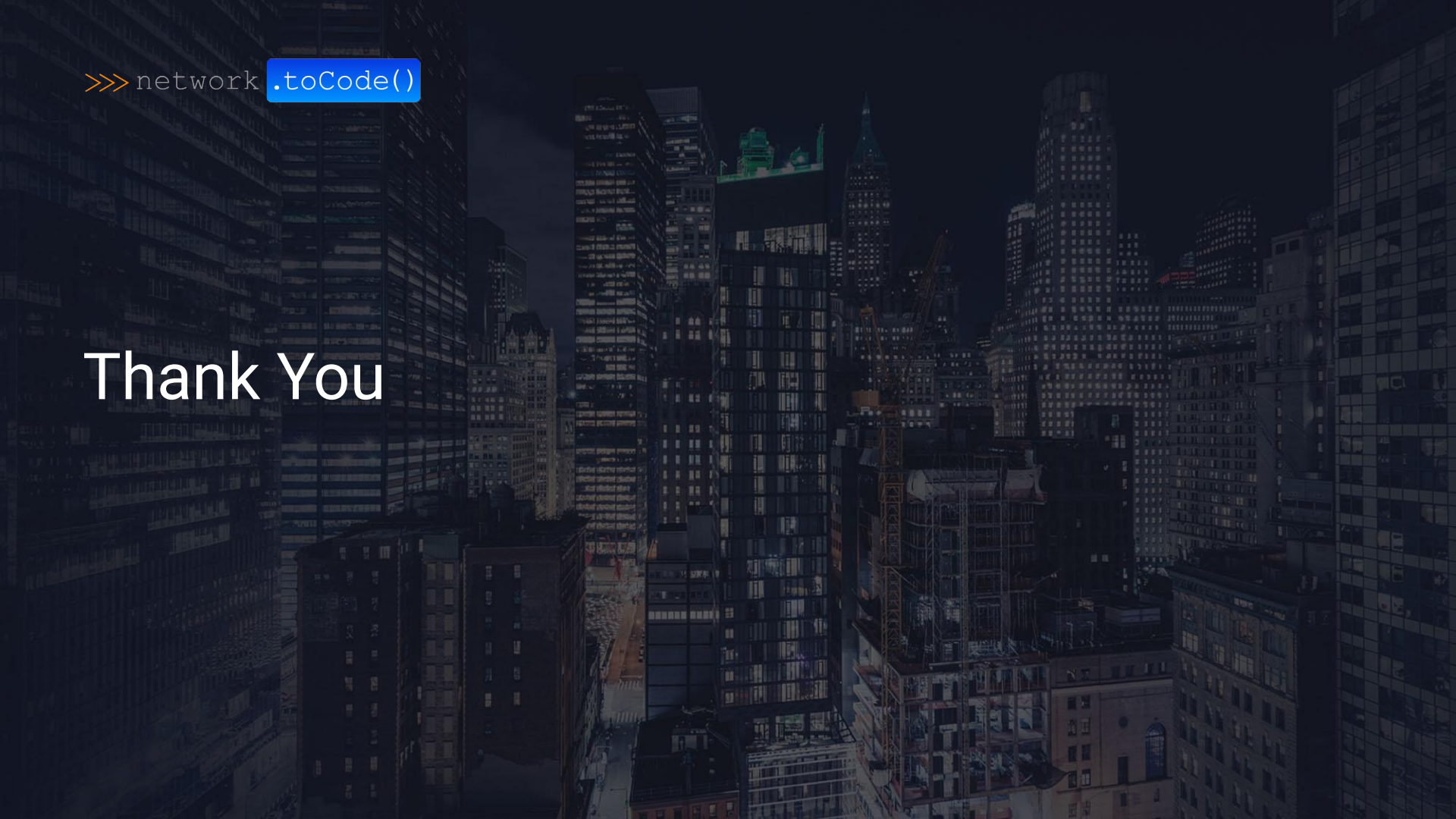
What is a device?

When it is a chassis?
When Cisco Stackwise?
When a Fabric Extender?
When a Virtual Port Channel?

Next Step: Data Modeling 201

Data Modeling Training Plan




>>> network .toCode()
Thank You