
>>> network .toCode()
NTC Reference Architecture
New Hire Onboarding

Agenda

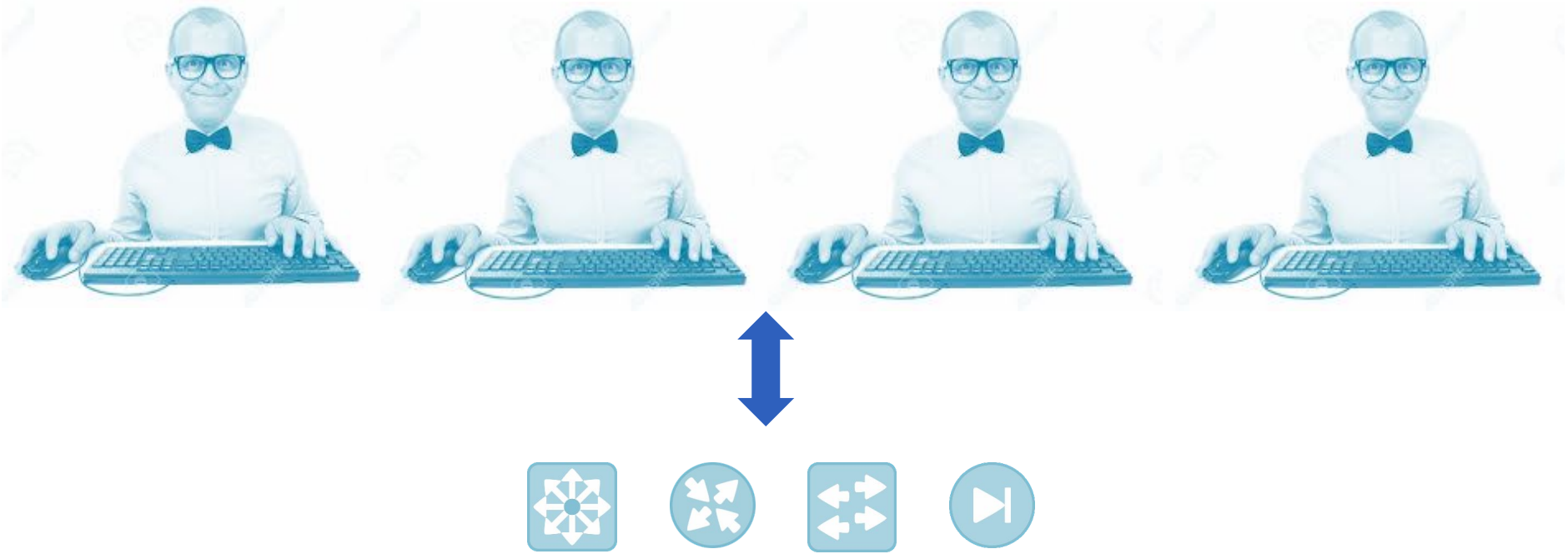
Introduction

Overview

Components

Introduction

>>> Traditional Network Operations



>>> [Common] Starting Point

Old school scripting

CLI & GUI

Python scripts
Ansible playbooks

- backups.pl
- vlans.pl
- MySQL
- Custom web app in PHP
- Saved on local disk

- get_neighbors.py
- vlan_check.py
- deploy.yml
- Git Push (force)

>>> Create a Strategy

- Nucleus for all network automation
- Extensible and modular
- Enterprise controls

- Makes tasks easy for admins
- Disparate scripts
- Limited to no data integration

>>> What is Network Automation?

Reference Architecture

Overview

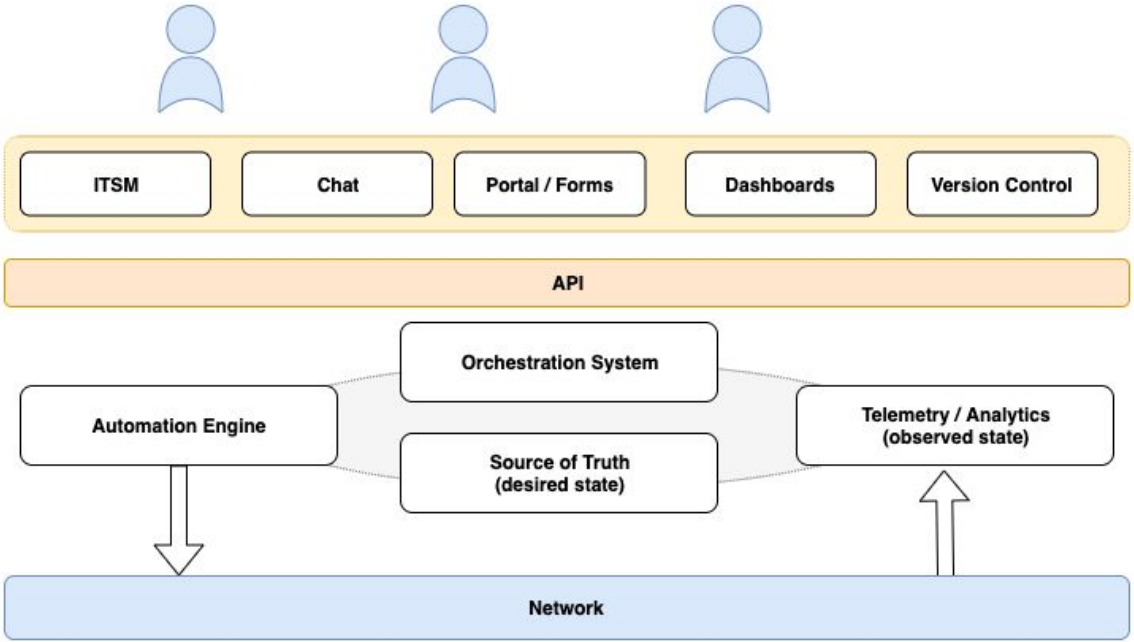
>>> Reference Architecture

Overview

Network to Code has developed a reference architecture, which is used as the foundation for the solutions and frameworks that it designs for its clients. This architecture includes the common functional elements and services required in any automation solution, broken into 6 sections, with multiple components in each.

- Source of Truth
- Automation Engine
- Telemetry and Analytics
- Orchestration System
- User Interaction
- External Integrations

>>> Network Automation Framework



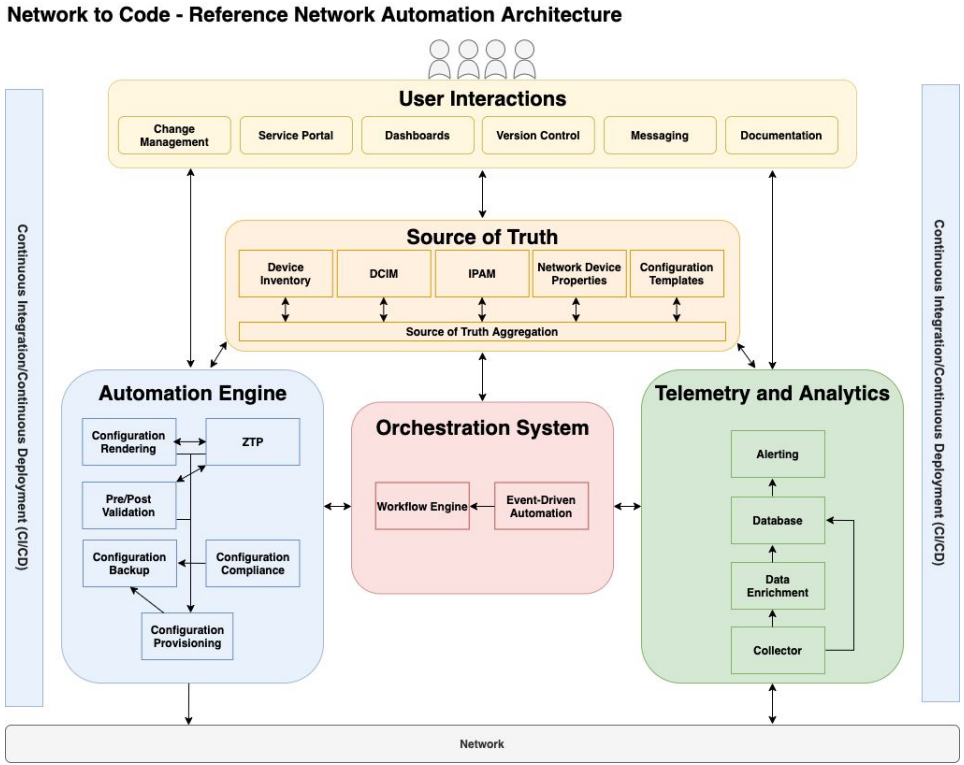
>>> Functional vs Implementation

It’s about the function, not the tool

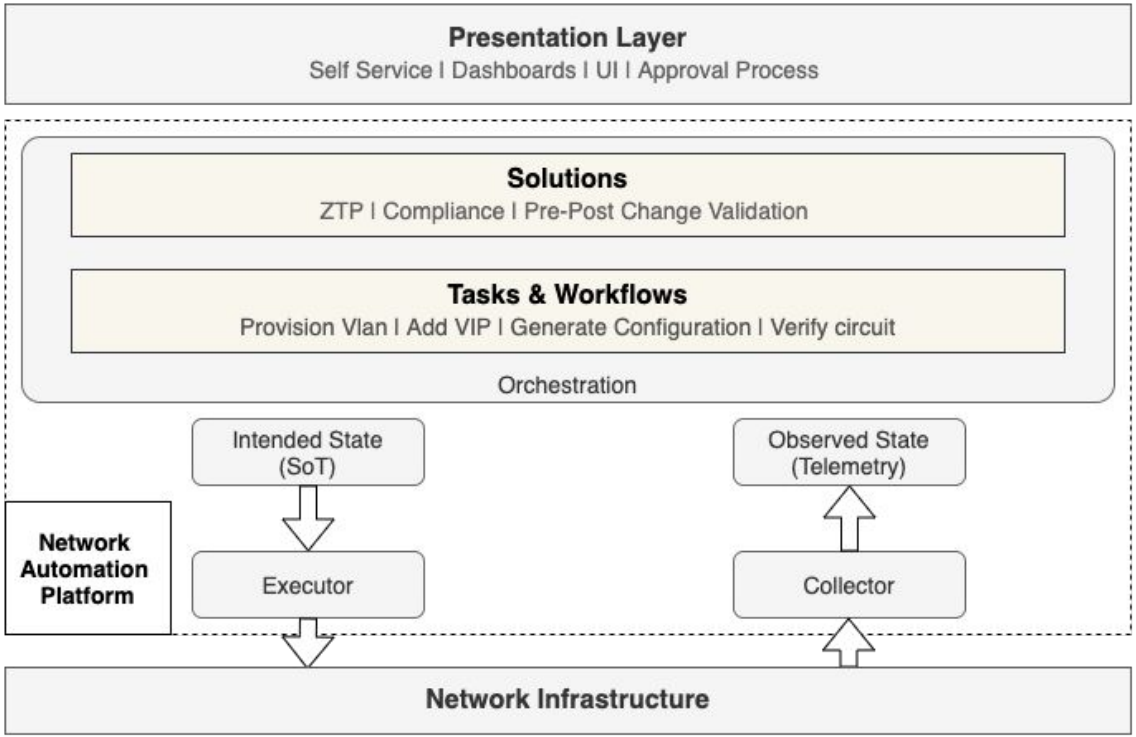
- Identify the requirements
- Identify the functional building blocks
- Define the main components
- Identify which tool(s) can be used for each requirements.

Especially important for system/product that provide multiple features (Nautobot, NetBox, ServiceNow ..)

>>> Network Automation Framework



>>> Network Automation Framework

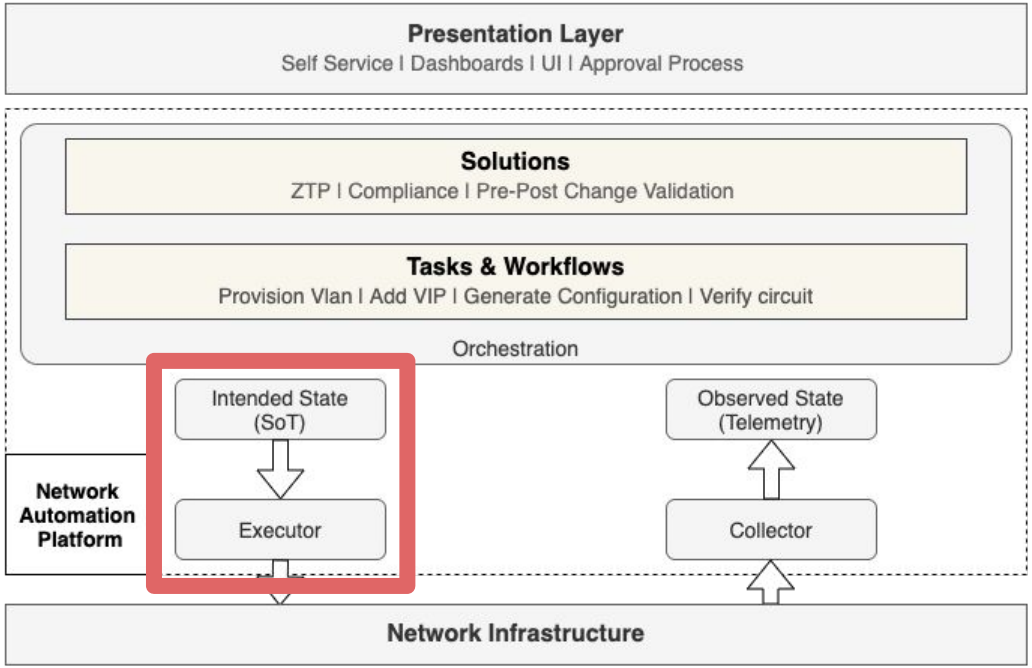


>>> Completed Assessments

- Roche
- University of Chicago
- Avast
- Marathon
- Baxter
- DTCC


Source of Truth

>>> Network Automation Framework



>>> Intended State/ Source of Truth

Role: Store and organize the intended/desired state of the network

Inventory	**Cabling**	**Routing Protocols**	**Configuration Templates**	**Applications Directory**
IP Addresses	**ACL**	**Circuits**	**Rack Physical**	**Software Standard**
Vlans	**Intended Operational State**	**Device Properties**	**Status and Roles**	**Servers**

>>> Intended State/ Source of Truth

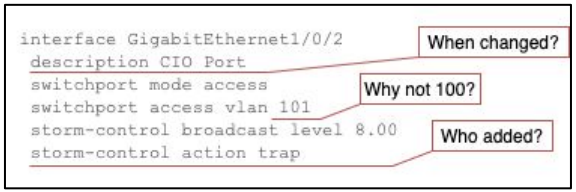
Role: Store and organize the intended/desired state of the network

Key Attributes:

- Guarantee accuracy of the data
- Provide traceability, history of the SoT
- Provide atomic changes
- Ease of access to the data
- Capture the relationship between the objects

>>> Why is SoT (data) Important?

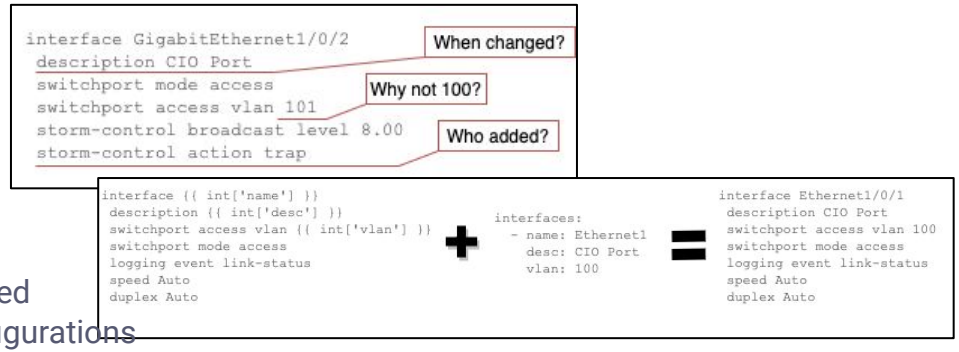
- Configurations do not provide **tracking**
  - Who created the configuration?
  - Why did they make the change?
  - When did they make the change?



>>> Why is SoT (data) Important?

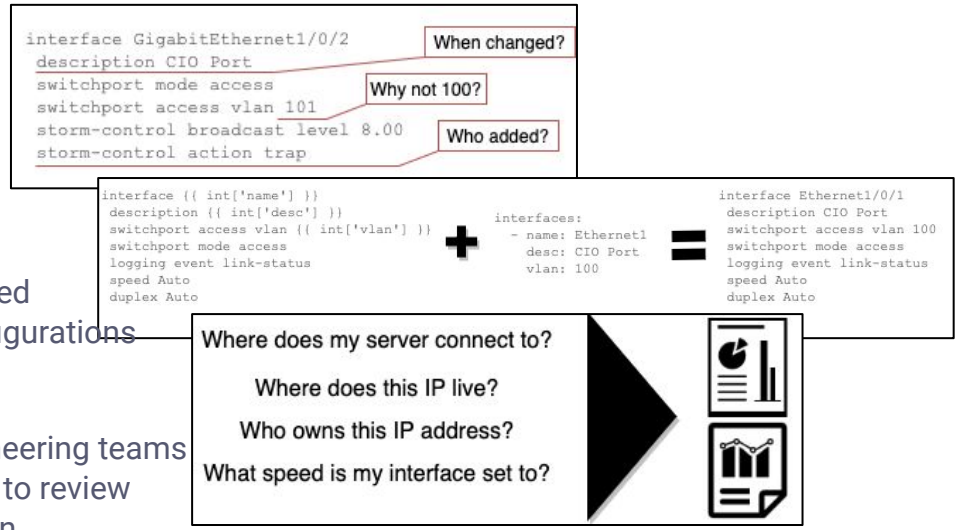
- Configurations do not provide **tracking**
  - Who created the configuration?
  - Why did they make the change?
  - When did they make the change?

- Ensures **consistency**, if data cannot fit in, can't be added
  - Allows ability to disaggregate data from the configurations



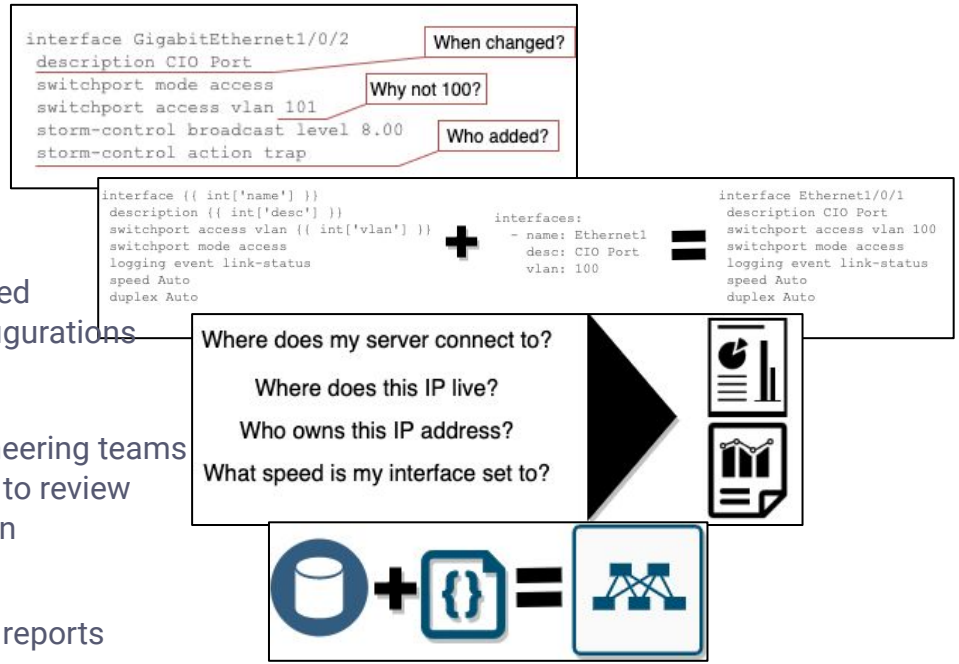
>>> Why is SoT (data) Important?

- Configurations do not provide **tracking**
  - Who created the configuration?
  - Why did they make the change?
  - When did they make the change?
- Ensures **consistency**, if data cannot fit in, can't be added
  - Allows ability to disaggregate data from the configurations
- Configurations are **not "queryable"** data is
  - The data is valuable outside of the network engineering teams
  - Only method is to take valuable engineering time to review
  - Data becomes auditable, rather than configuration



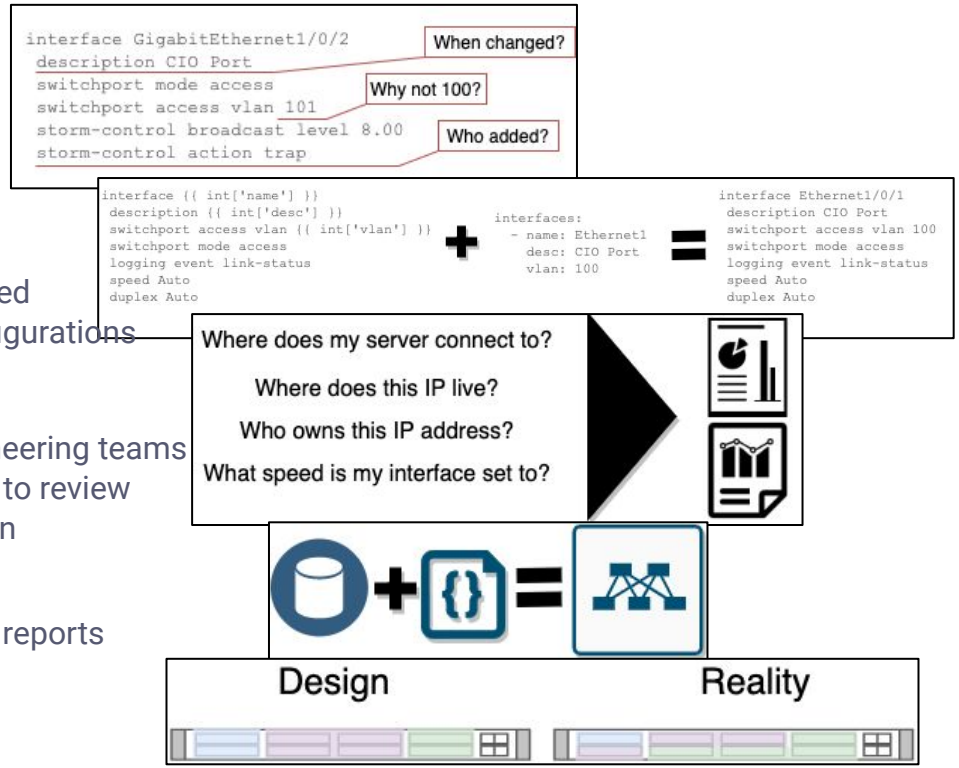
>>> Why is SoT (data) Important?

- Configurations do not provide **tracking**
  - Who created the configuration?
  - Why did they make the change?
  - When did they make the change?
- Ensures **consistency**, if data cannot fit in, can't be added
  - Allows ability to disaggregate data from the configurations
- Configurations are **not "queryable"** data is
  - The data is valuable outside of the network engineering teams
  - Only method is to take valuable engineering time to review
  - Data becomes auditable, rather than configuration
- Provides ability to document network
  - Documentation can be auto-generated visuals or reports



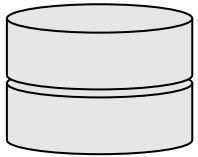
>>> Why is SoT (data) Important?

- Configurations do not provide **tracking**
  - Who created the configuration?
  - Why did they make the change?
  - When did they make the change?
- Ensures **consistency**, if data cannot fit in, can't be added
  - Allows ability to disaggregate data from the configurations
- Configurations are **not "queryable"** data is
  - The data is valuable outside of the network engineering teams
  - Only method is to take valuable engineering time to review
  - Data becomes auditable, rather than configuration
- Provides ability to document network
  - Documentation can be auto-generated visuals or reports
- Designs are intended to be consistent
  - Data allows a design to be flexible



>>> Git vs Database

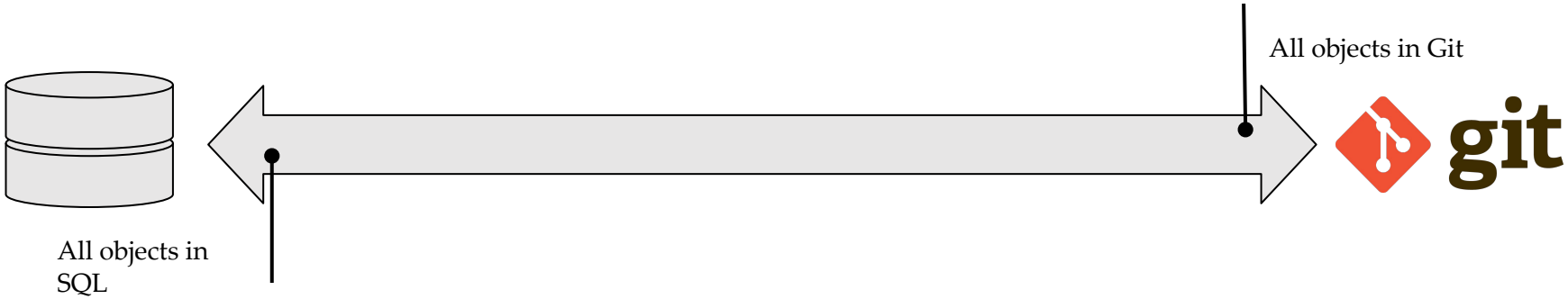
Source of Truth



	SQL Database	GIT
Store and organize intended state of the network	+++	+
Guarantee accuracy of the data	+	+++
Provide traceability, history of the SoT	+	+++
Provide atomic changes	++	+++
Ease of access to the data	+++	+

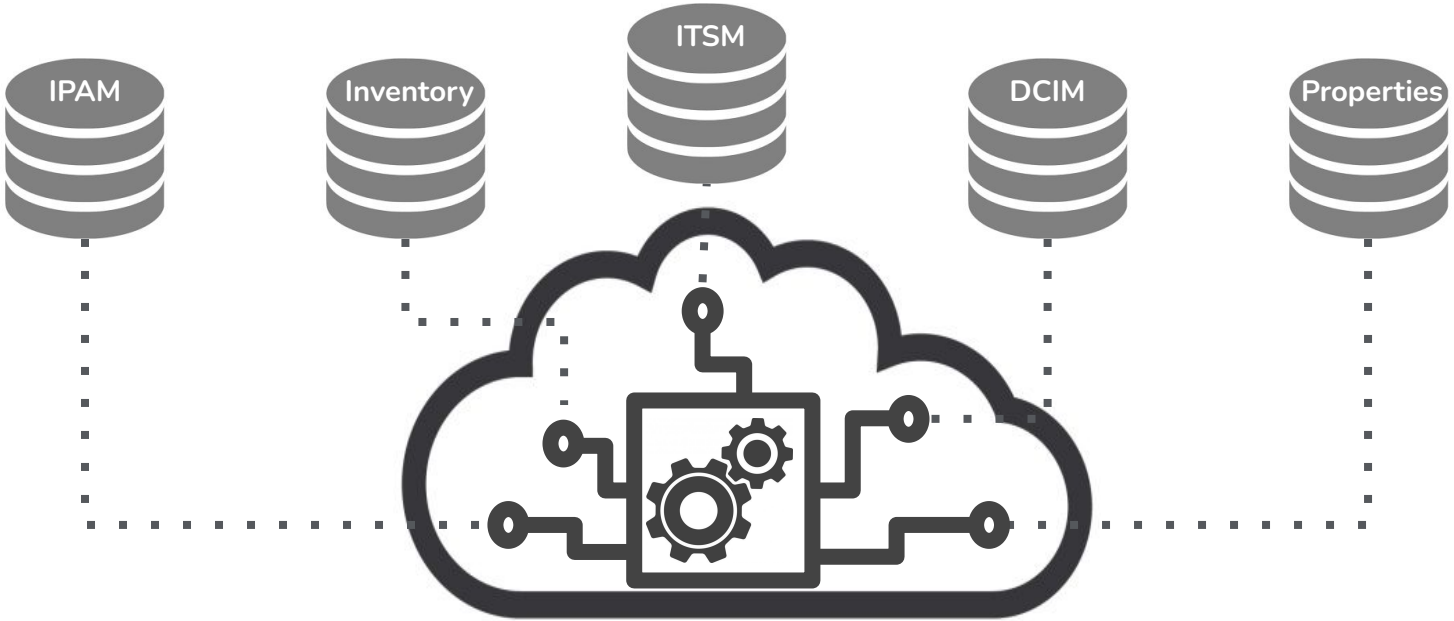
>>> Git vs Database

Source of Truth



>>> Aggregation

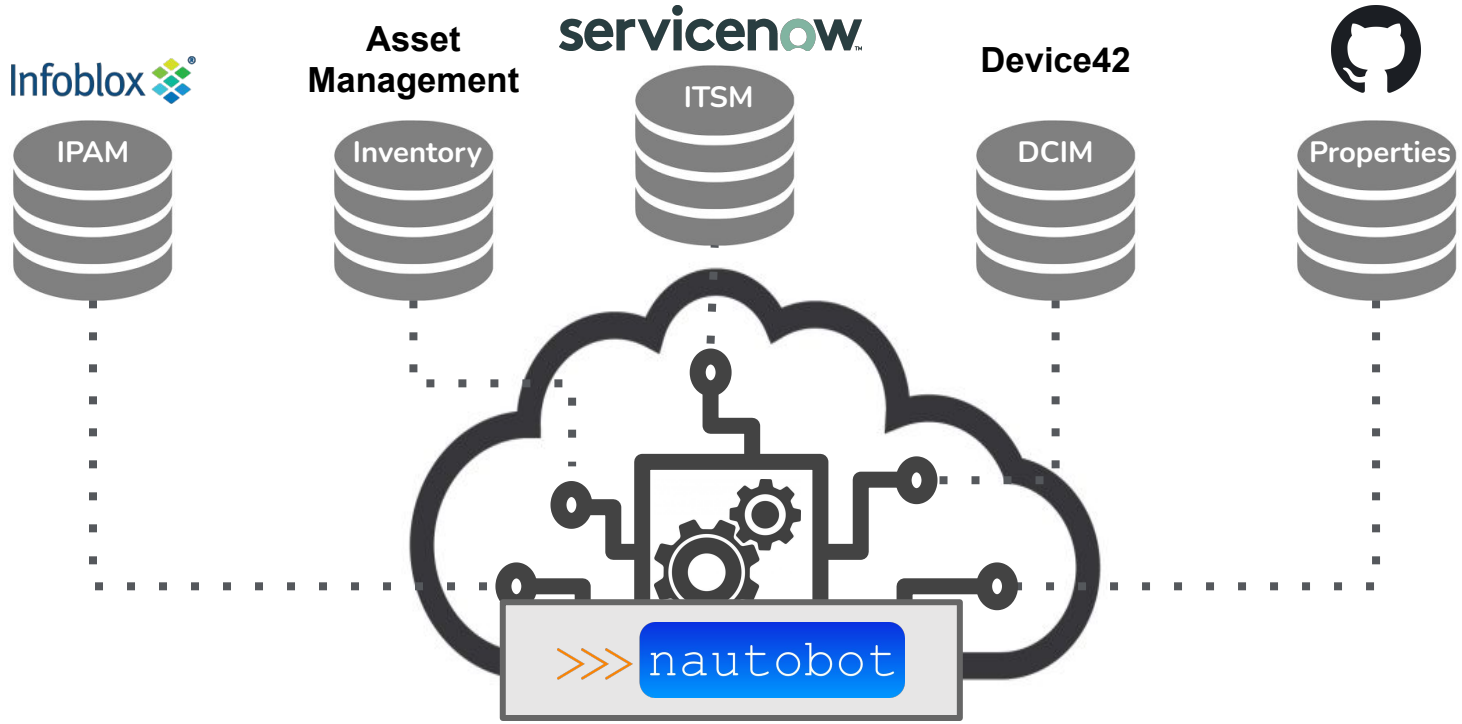
Source of Truth



Build a proxy which sits between data sources and the systems that consume them

>>> Aggregation

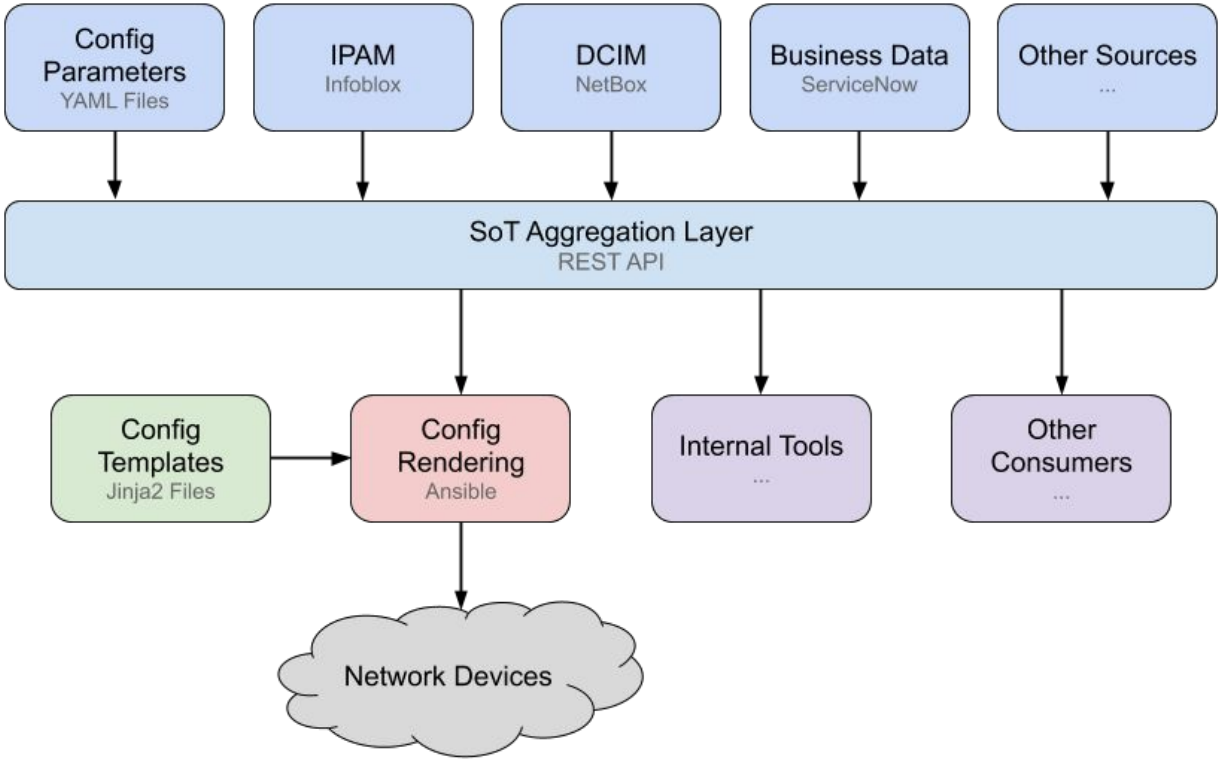
Source of Truth



Build a proxy which sits between data sources and the systems that consume them

>>> Aggregation

Source of Truth



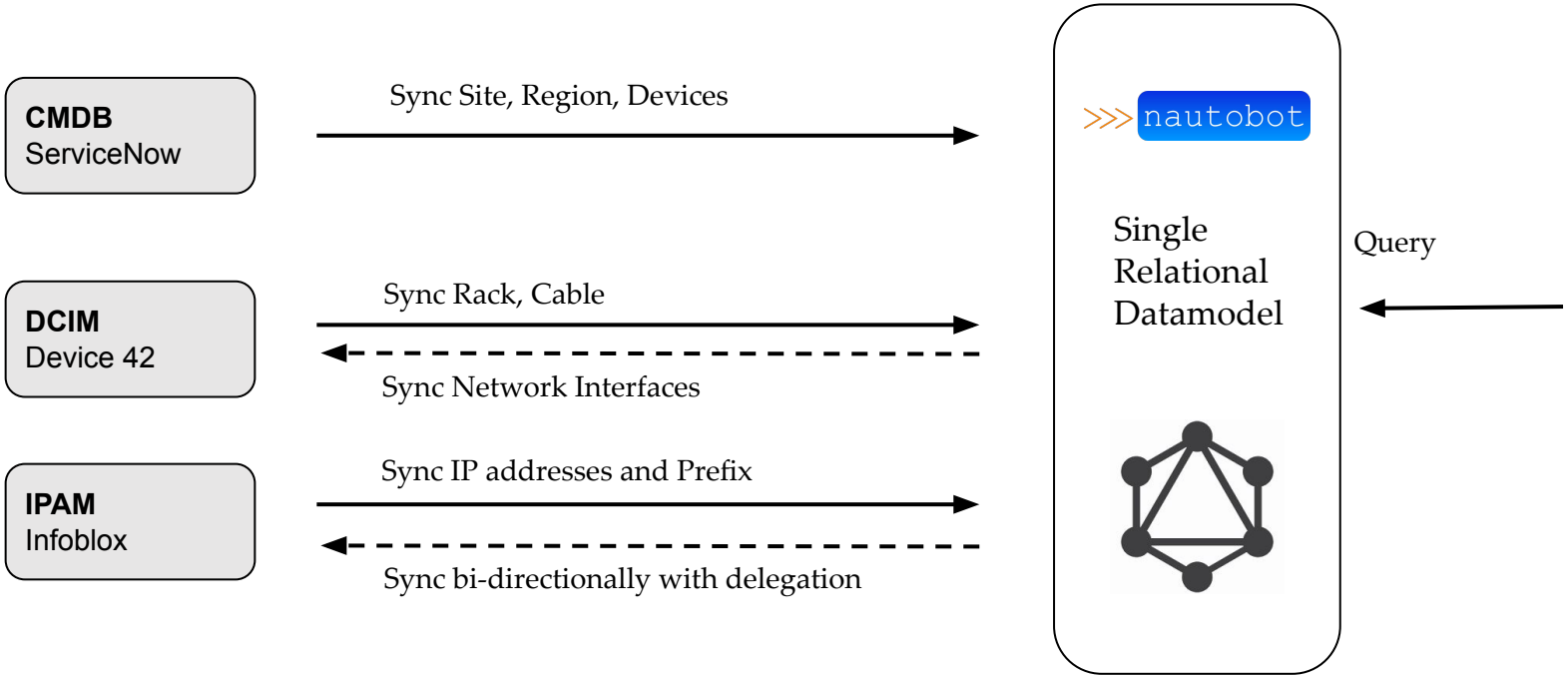
>>> SoT Replicas between system

- Still one official System of Record
- One or multiple replicas possible
  - Visibility
  - Integration
  - Consumption

Example : Infoblox is the office SOR for IP address,
data are replicated in a Network Source of Truth such as **Nautobot** to simplify consumption and integration with DCIM

>>> Aggregation

Source of Truth



>>> Aggregation

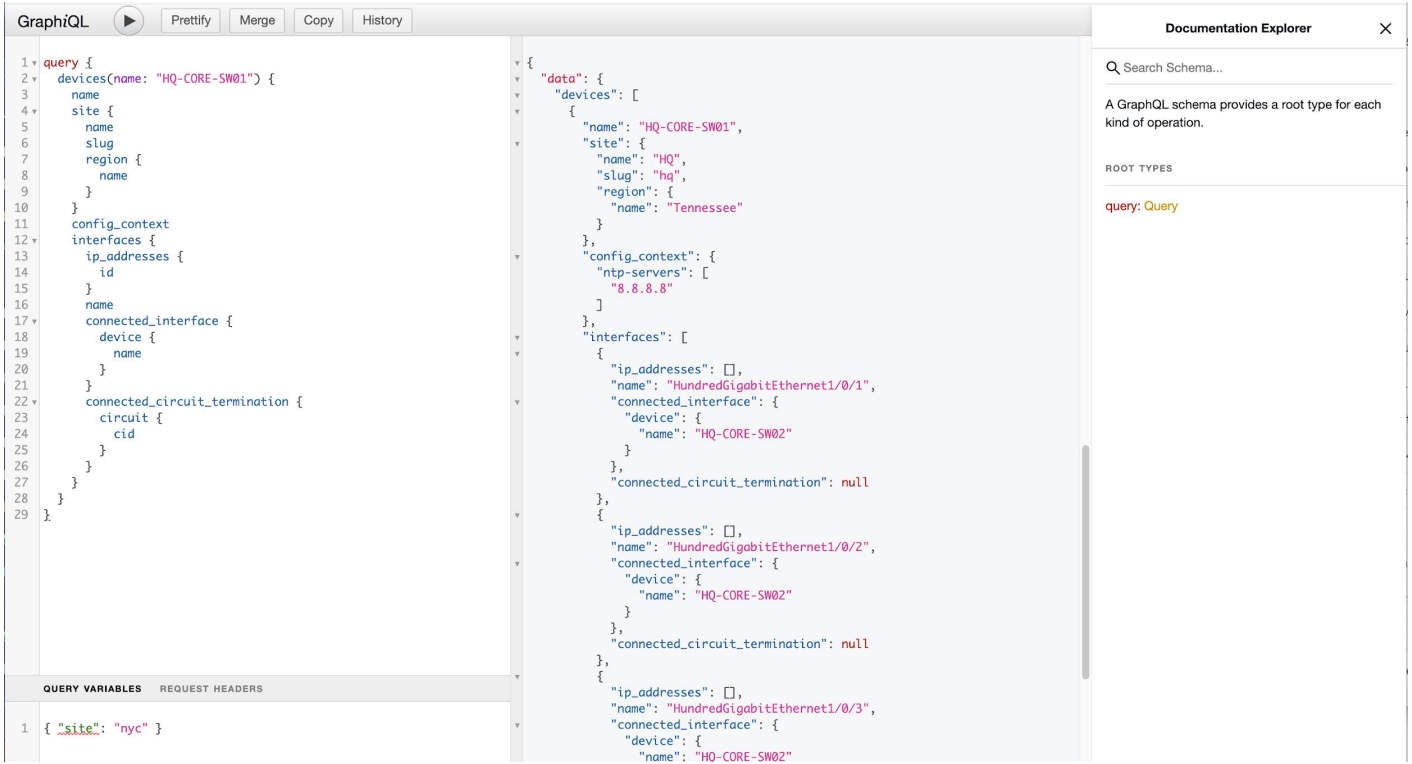
Source of Truth

GraphQL as the main interface to query data from the SOT

- Query Language for API
- Resources defined by a GraphQL Schema
- Client sends query,
- server orchestrates data

>>> Aggregation

Source of Truth



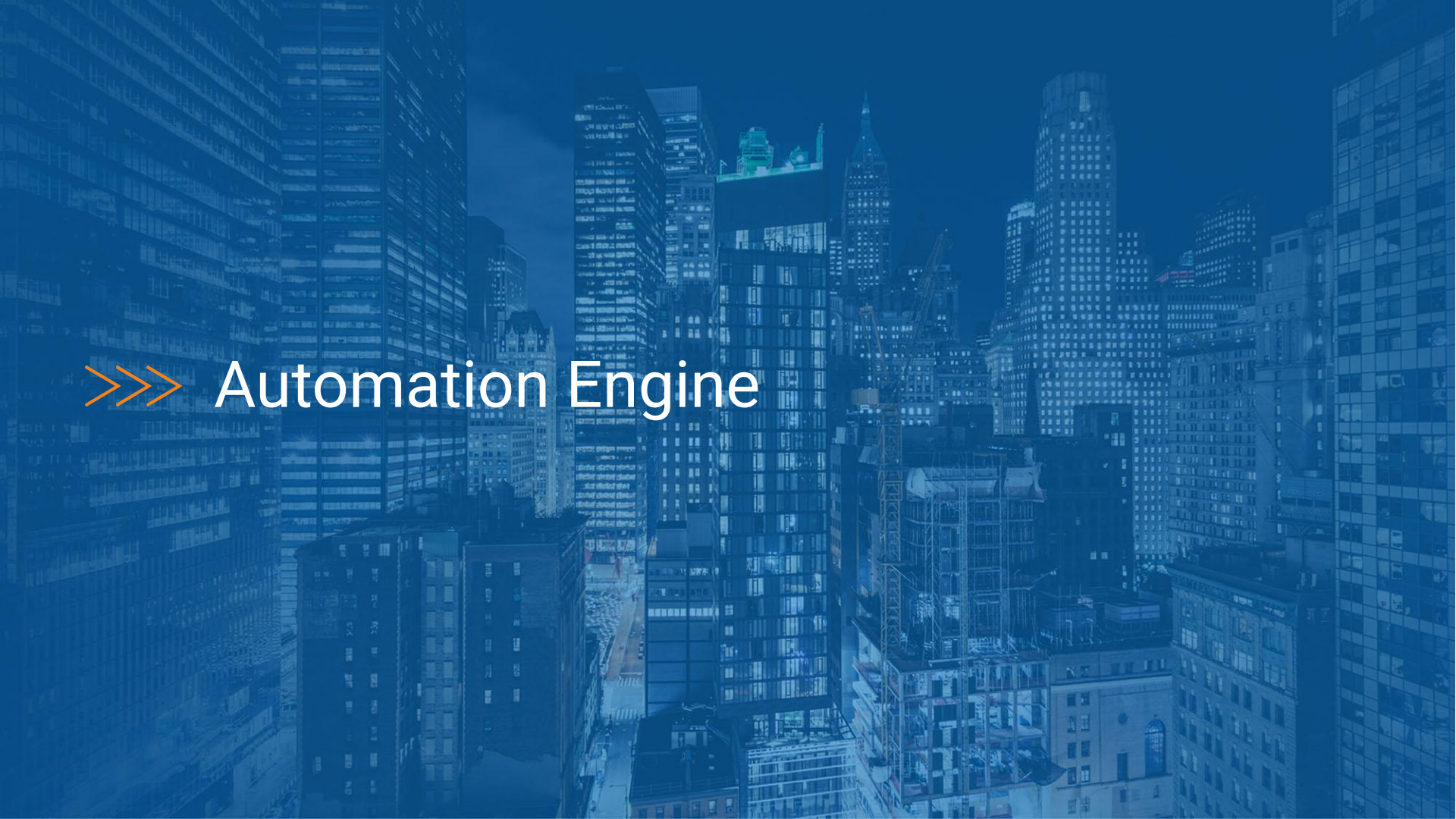
>>> Intended State/Source of Truth

Role: Store and organize the intended/desired state of the network

Examples:

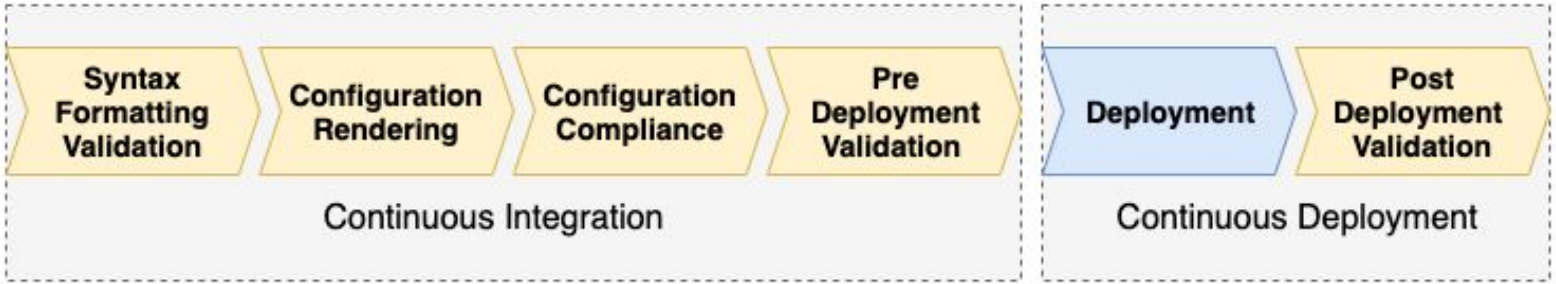
>>> nautobot

servicenow


Automation Engine

>>> Executor/Automation Engine

Role: Interact with the network, render and deploy configurations



>>> Executor/Automation Engine

Role: Interact with the network, render and deploy configurations

Key Attributes:

- Simplify interact with network devices
- Abstract vendor specific APIs/Interface
- Authentication
- Performance

>>> Executor/Automation Engine

Role: Interact with the network, render and deploy configurations

Examples

>>> Executor/Automation Engine

Role: Interact with the network, render and deploy configurations

Examples

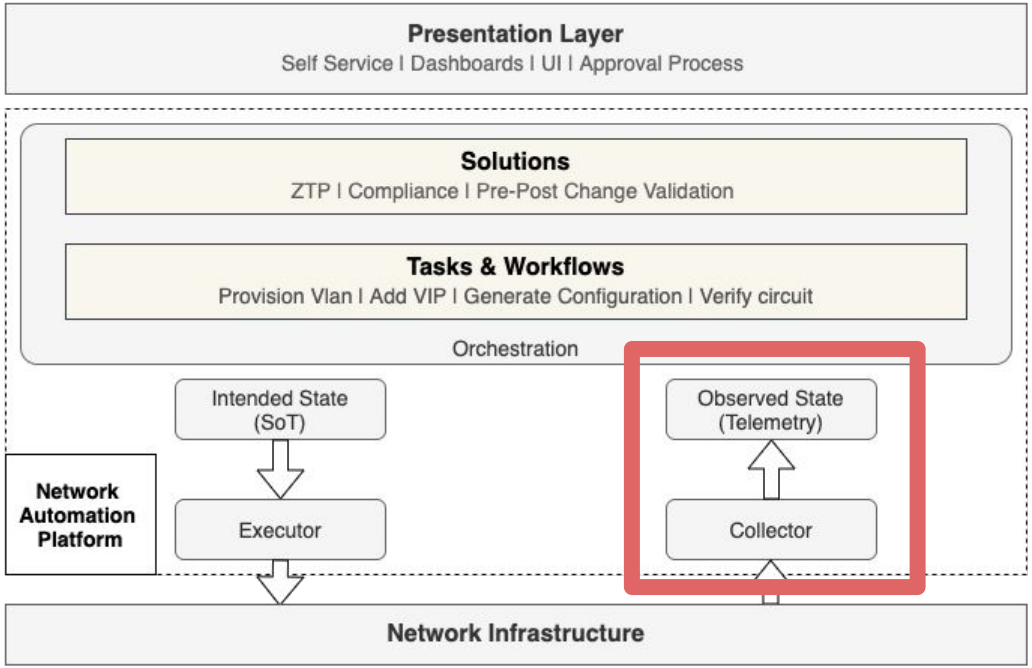
>>> Executor/Automation Engine

Feature	Ansible Tower/AWX	Nornir
Accessibility	✔✔✔	✔
Authentication	✔✔✔	✘
API	✔✔✔	✘
Complex Workflow	✔	✔✔✔
Commercial Support	✔✔✔	✘
Credential Management	✔✔✔	✘
Extensibility	✔✔	✔✔✔
RBAC	✔✔	✘
Network Device Support	✔✔✔	✔✔
Non-network Device Support	✔✔✔	✔
Scheduler	✔✔✔	✔
Speed	✔	✔✔✔
Traceability	✔✔✔	✘
User Interface	✔✔	✘
Workflow Deployment	✔✔✔	✔✔

Table - Automation Engine comparison

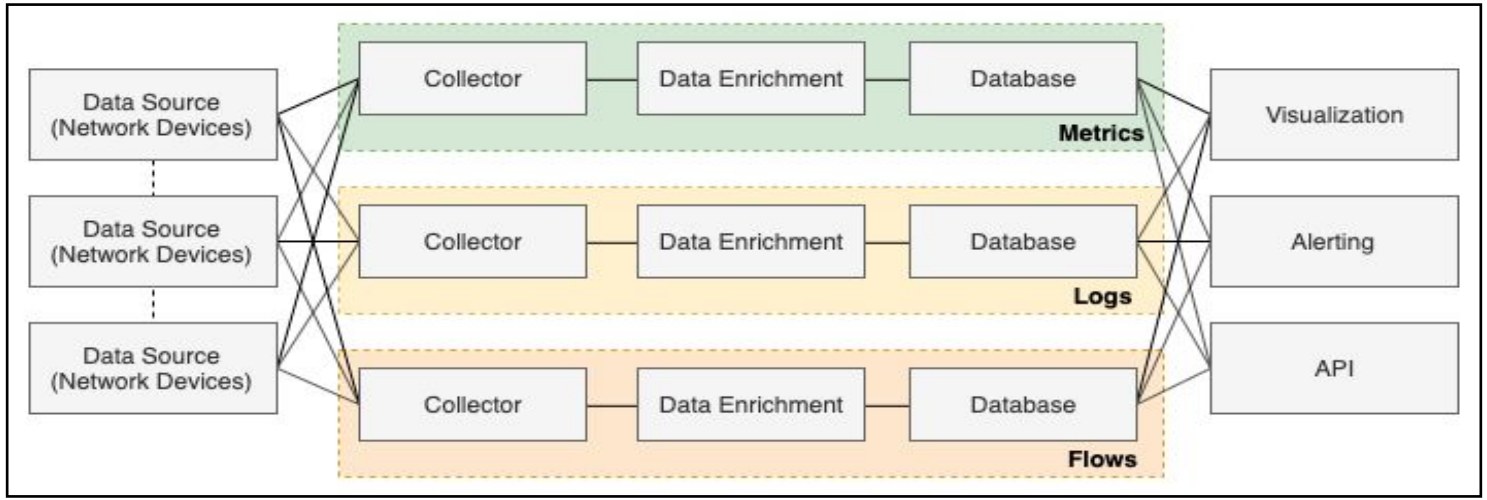

Telemetry & Analytics

>>> Network Automation Framework



>>> Telemetry and Analytics

Role: Provide visibility into the current and past state of the network, in a meaningful way.



>>> Collector & Enrichment Layer

Role: Extract the current state of the network

Key Attributes:

- Easily query all states from the network
- Easily add new information to the collection pipeline
- Insert business metadata to the data as needed
- Scalable and Resilient
- Support multiple types of data/interfaces (SNMP/ CLI / gNMI / Flows / Logs)

>>> Collector

Role: Extract the current state of the network

Examples

>>> Observed State

Role: Store the current and past state of the network

Key Attributes:

- Ease to query current and paste state of the network
- Ingest large volume of data
- Store business metadata
- Support all types of data (metrics, logs, structured ..)

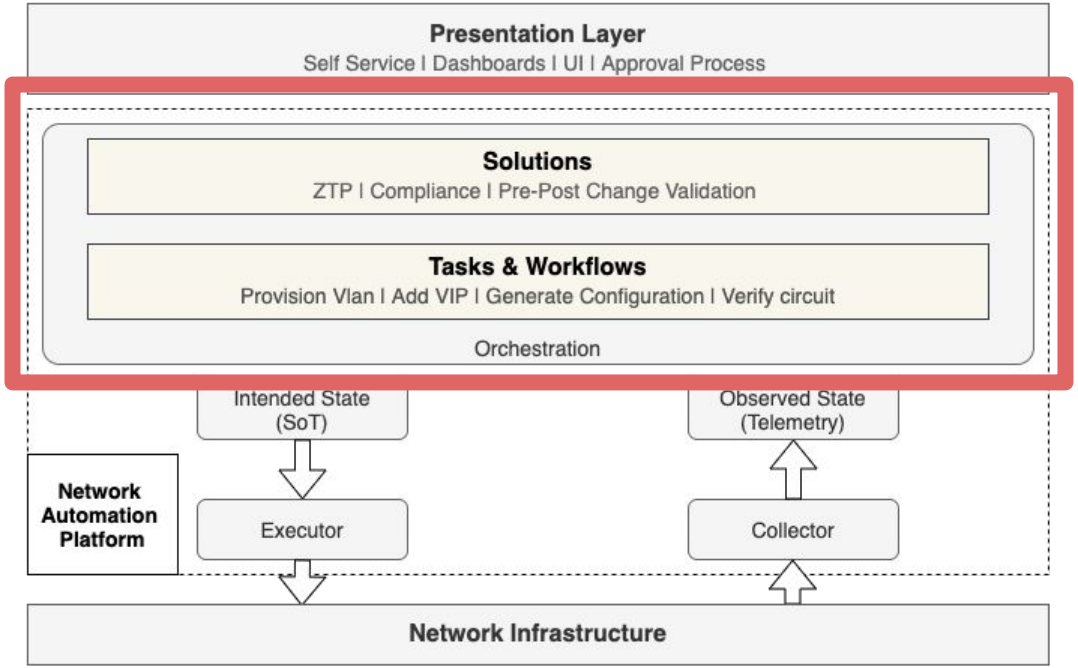
>>> Observed State

Role: Provide visibility into the current and past state of the network

Examples


Orchestration System

>>> Network Automation Framework



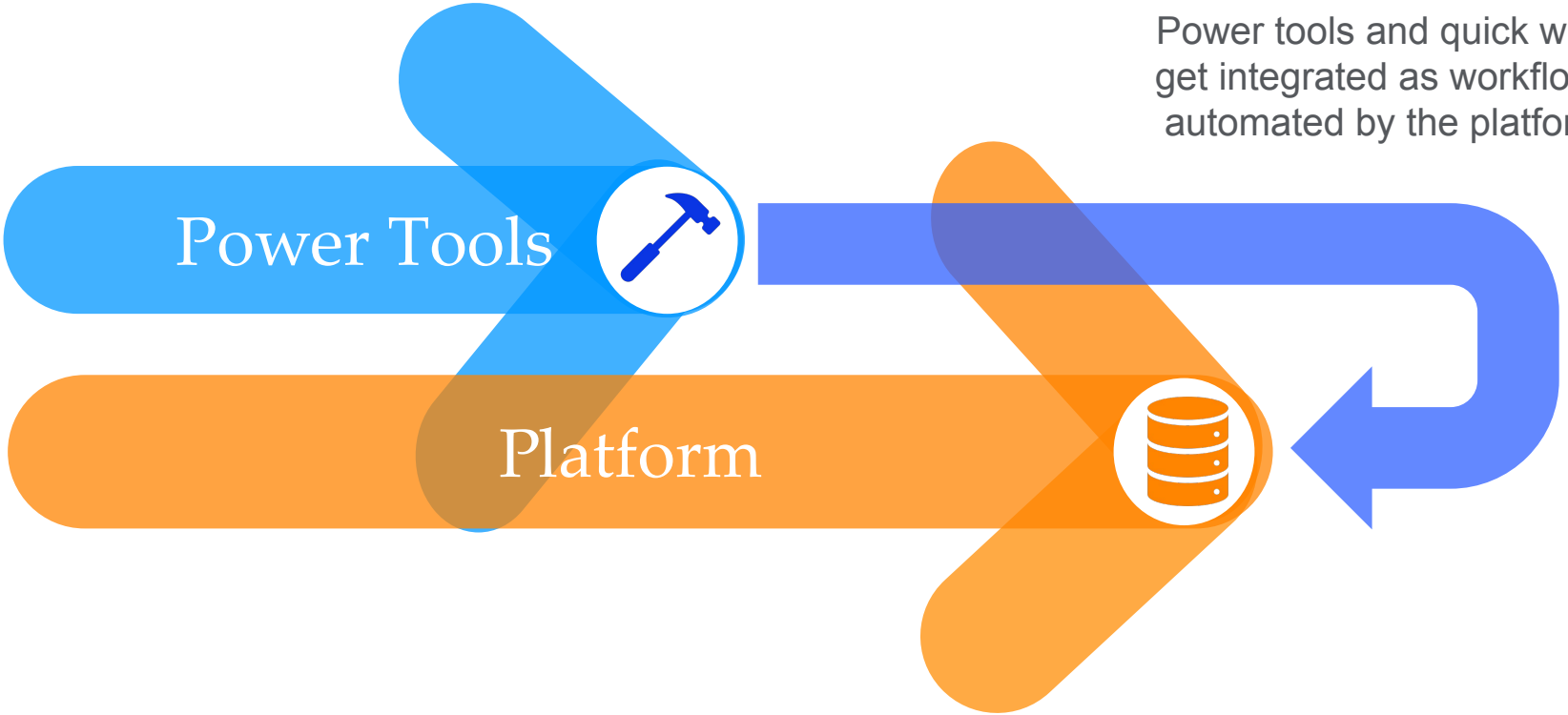
>>> Orchestration System

Role: Define & execute customer specific workflows

Key Attributes:

- Ease to define all workflows, simple and advanced
- Traceability & troubleshooting of past workflows execution
- Integration with multiple data sources

>>> Workflow is the new power tool

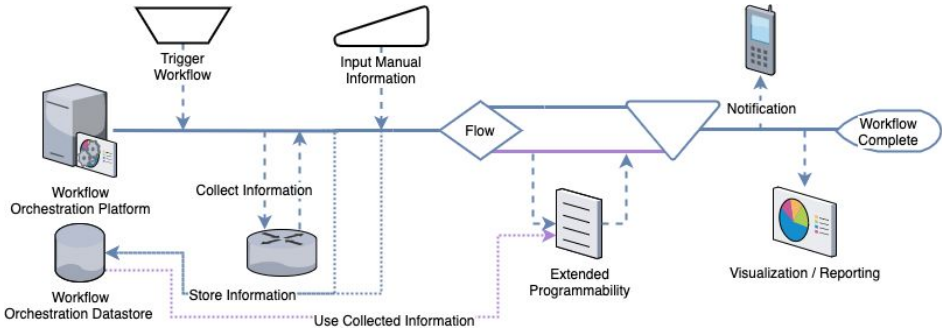


>>> Orchestration System

Workflow Engine

Characteristics considered when reviewing a Workflow Engine

- Inputs: How a user can input data for a single workflow run.
- Flow Control: How to control order of tasks, based on outcome of previous tasks.
- Data Store: Ability to have a temporary data store.
- Extended Programmability: Ability to add arbitrary code.
- Visualization: How to build and view job runs.
- Notification: Ability to send messages

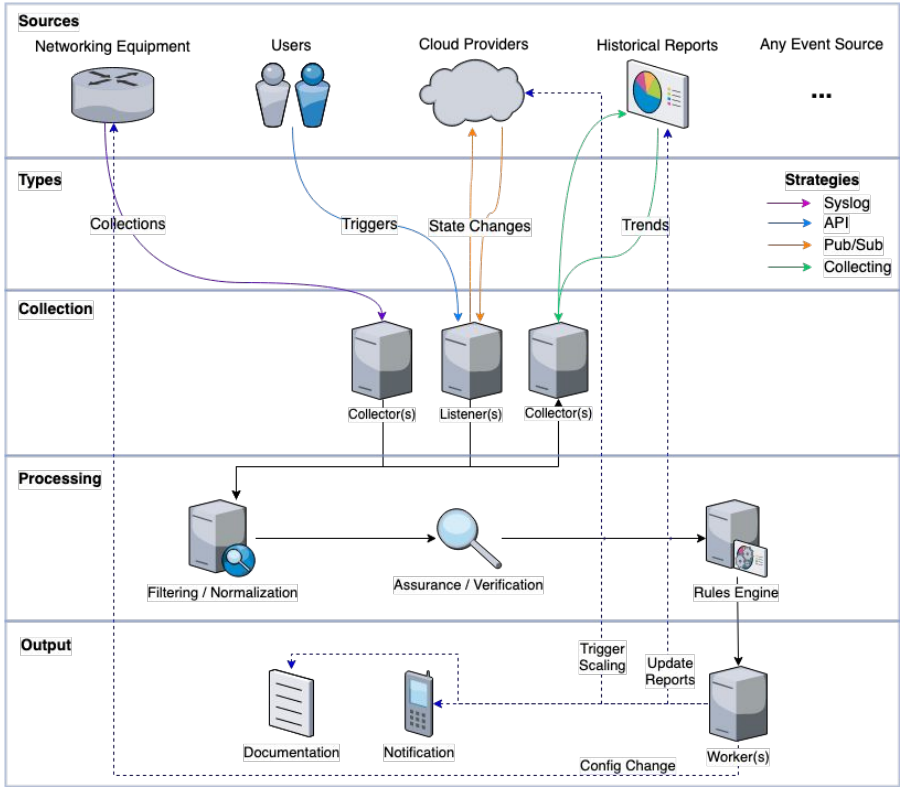


>>> Orchestration System

Event Driven Automation

Characteristics considered when reviewing a Event-Driven Automation tool

- Input Types: How data is presented and validated
- Ingest Filtering: How data is is filtered for “important information”
- Event Processing: How to control order of tasks, based on outcome of previous tasks.
- Output: Ability to have a temporary data store.
- Rule Engine: Ability to add arbitrary code.
- Even Correlation/Thresholds: How to build and view job runs.



>>> Orchestration System

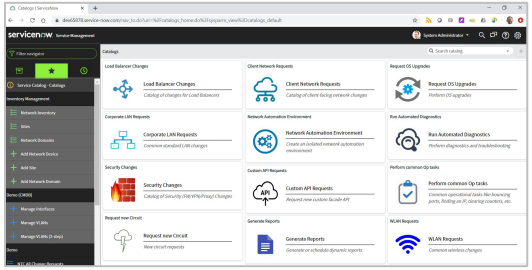
Role: Define & execute customer specific workflows

Examples:

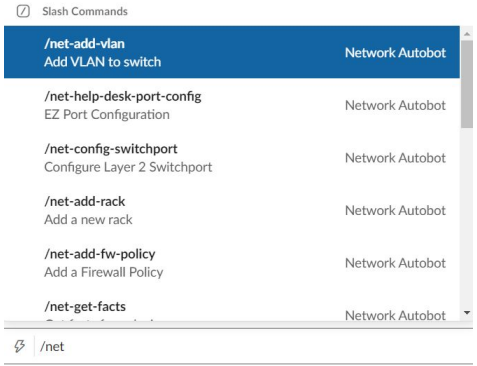
User Interaction

>>> User Interface / Presentation Layer

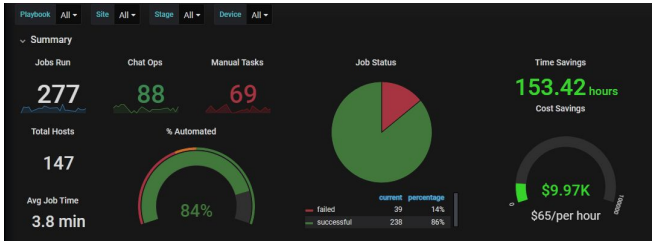
Role: Provide the best UI to the users and integrate with existing tools



ITSM



Chat



Dashboards

>>> One workflow, multiple interfaces

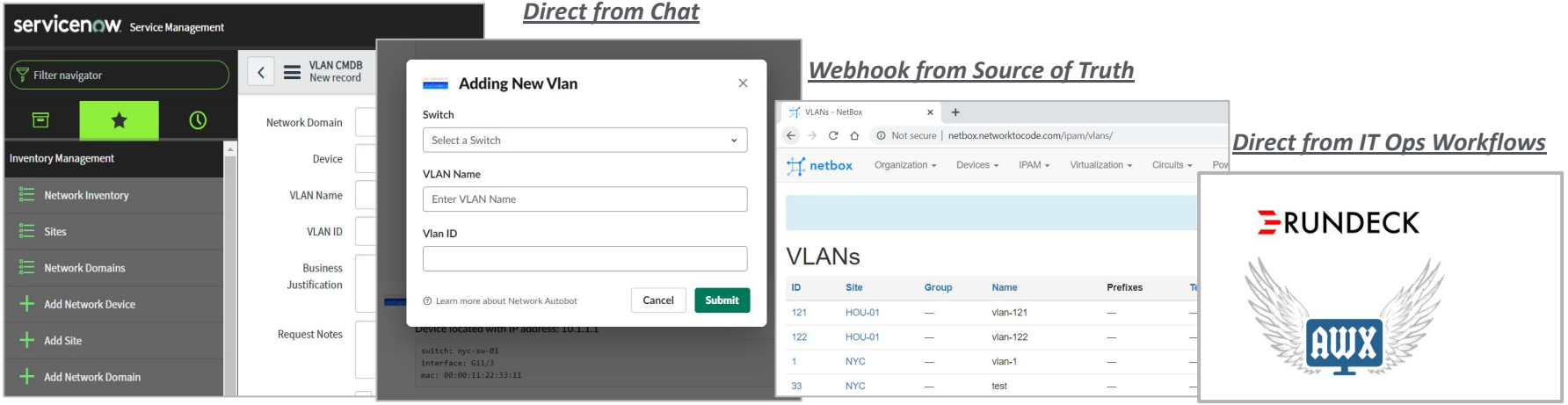
- Not all workflows have to be delivered the same way
- It's best to examine the target user and find the best presentation layer for them so it's an optimal user experience

ServiceNow Form

Direct from Chat

Webhook from Source of Truth

Direct from IT Ops Workflows



>>> User Interface / Presentation Layer

Role: Provide the best UI to the users and integrate with existing tools

Key Attributes:

- Integrate with ITSM Platform & CMDB
- Integrate with Chat system
- Allow creation & distribution of user defined dashboards
- Self-Service

>>> User Interface / Presentation Layer

Role: Provide the best UI to the users and integrate with existing tools

Examples

servicenow

slack

Confluence

bmc

Grafana

Webex Teams

kibana

External Integration

>>> External Integrations

Overview

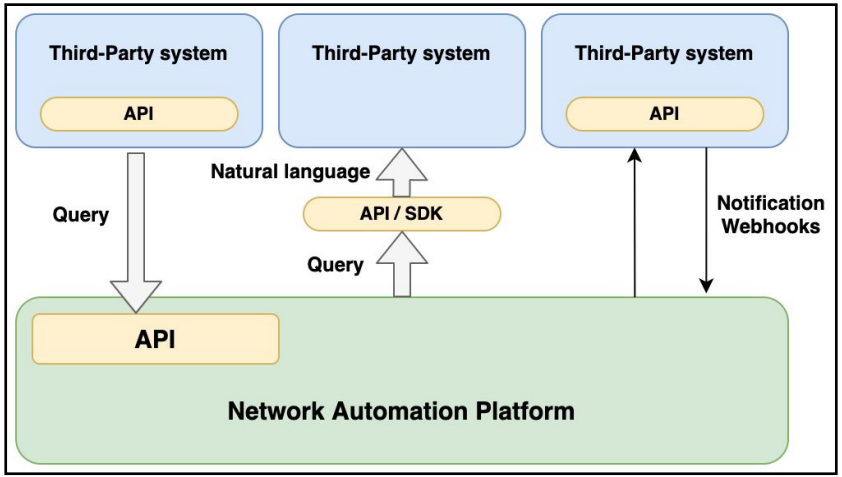
External integrations are meant to express the communication between systems, such as.

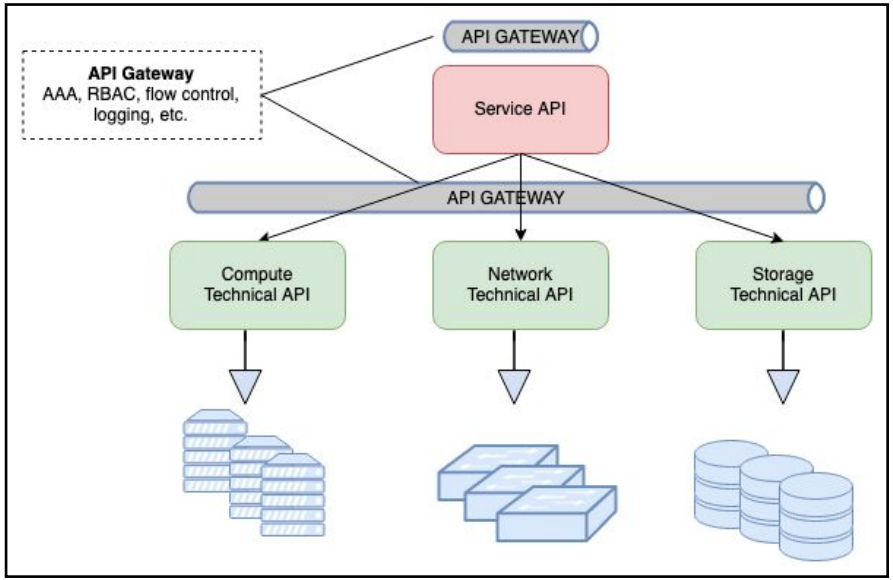
- Integrations between different IT teams and systems
- Integrations to external entities (such as Internet Service Provider)
- API control points between any two systems

Providing a common internal API gateway to control interactions, is a recommended way to interact with other systems and provide proper controls (RBAC, rate limiting, etc)

>>> External Integrations

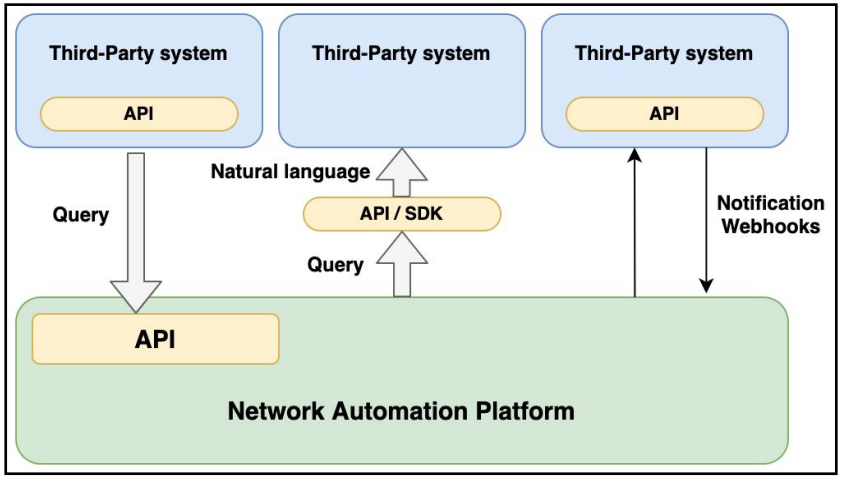
Diagrams





>>> External Integrations

High-Level Design

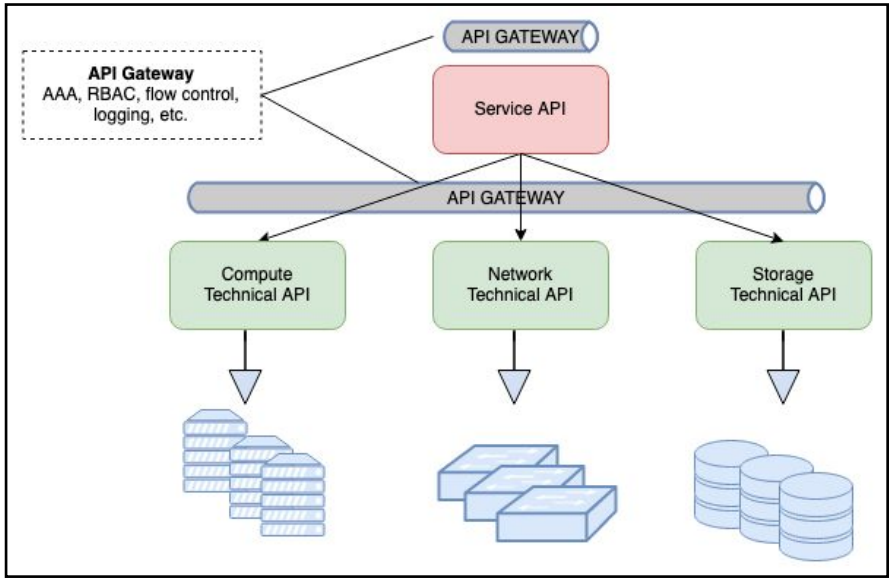


- Exposing API access to the Automation Platform allows external users controlled access
- Building API integrations to external systems (such as ISP) provides common method for Automation Platform to interact with
- Automated interactions via Webhooks to APIs allow for human-less automation

>>> External Integrations

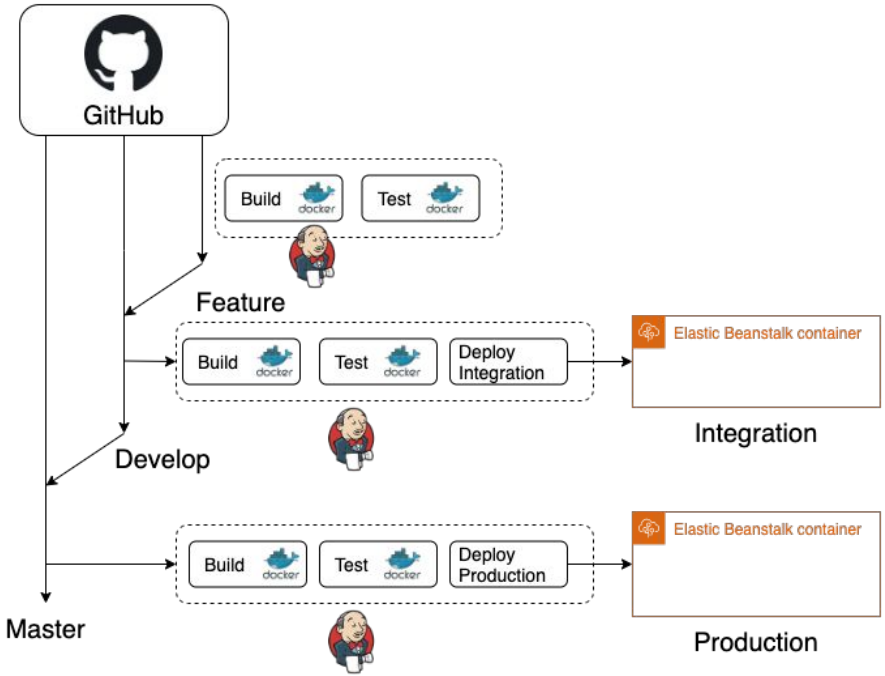
API Gateway Design

- Building an API gateway provides means for non-functional requirements
  - AAA, RBAC, logging, etc.
- Exposing Service APIs for higher level abstraction
  - /api/v1/create_server
    - Obtains IP
    - Configures port
    - Creates storage configuration
- Exposing Technical API's allow for low level features
  - /api/v1/{device}/add_vlan/{vlan}
  - /api/v1/{device}/config_port/{interface}

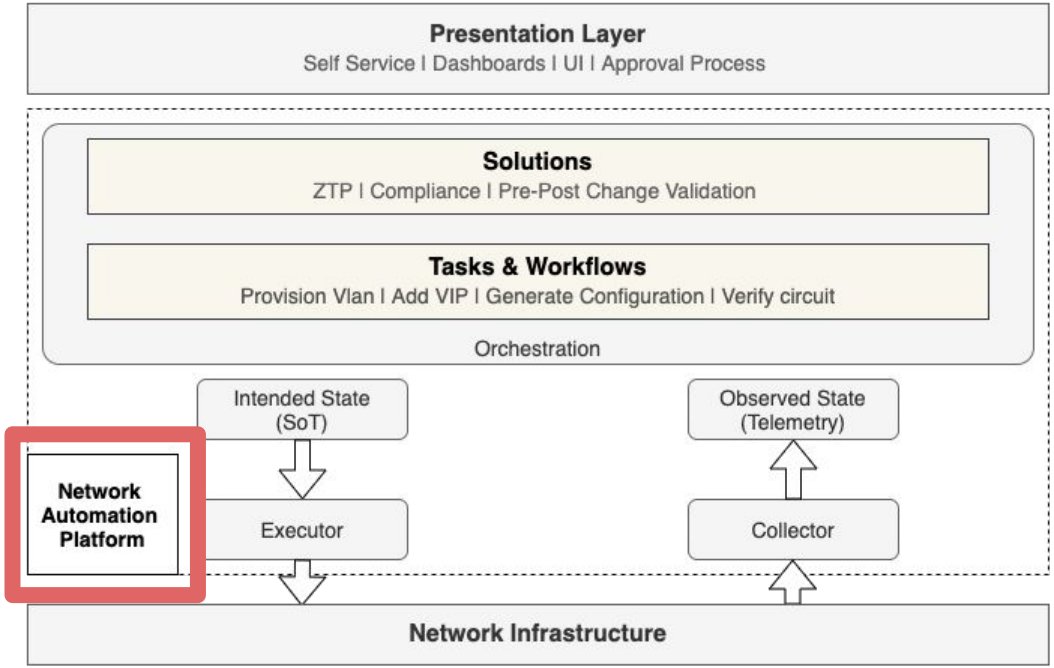


Platform

>>> Deployment pipeline



>>> Network Automation Framework



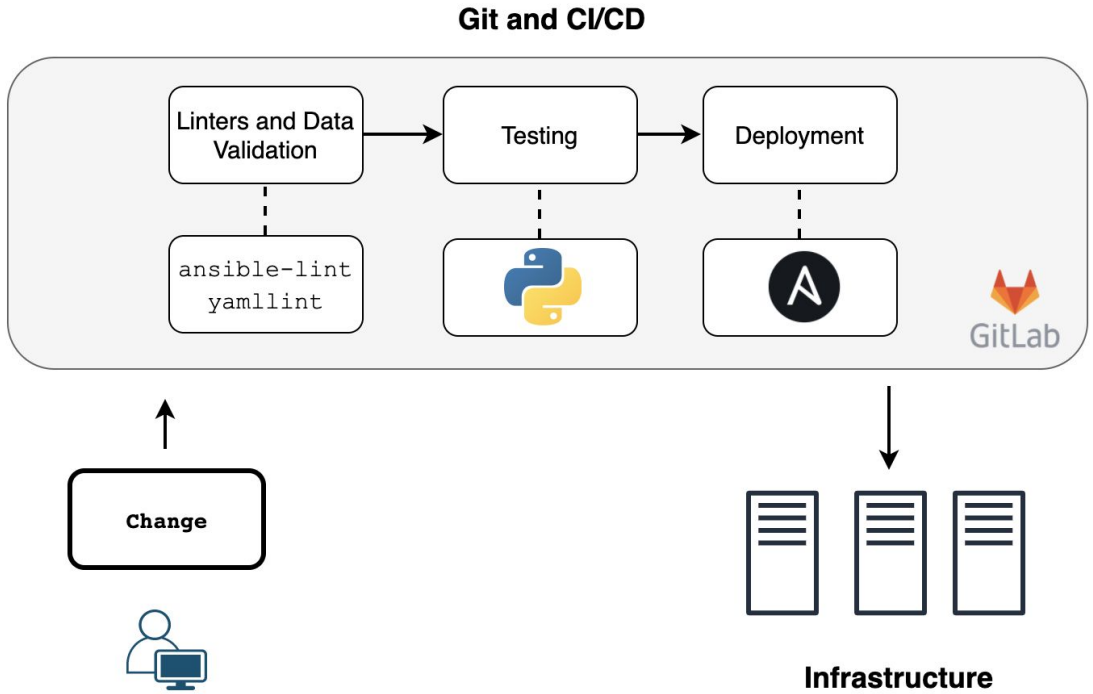
Role: Easily develop and maintain the components of the platform

Key Attributes:

- Version Control
- Continuous Integration / Unit Test
- Scale
- Security
- Logging/Maintainability

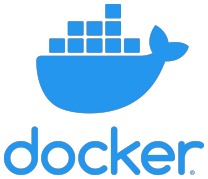
Development workflow

- Work done locally
- Changes submitted to Git branch/fork
- Linters and Data validation tools execute
- Testing executed
- PR/MR workflow for peer-review approval
- Deployment workflow executed



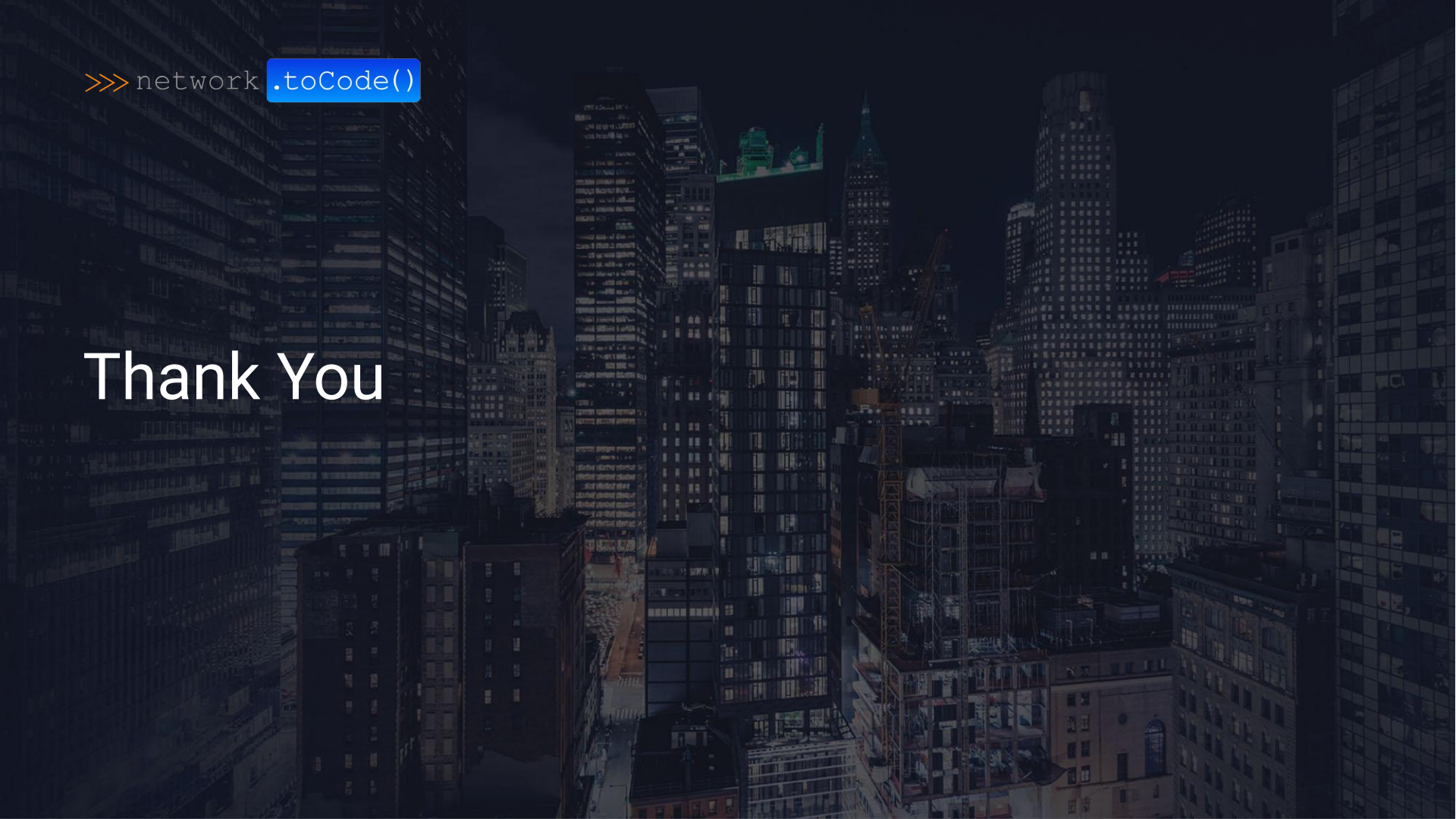
Role: Easily develop and maintain the components of the platform

Examples



>>> Summary

- Network Automation is about Workflows and Data (Intended and observed)
- A Network Automation Platform :
  - makes data easily accessible
  - is meant to be consumed by Network Engineers to define workflows
- A Network Automation Platform is composed of many components, assembled together.


>>> network .toCode()
Thank You