Ansible with Arista EOS + CloudVision

Fred Hsu
Carl Buchmann
Thomas Grimonet

ARISTA

Presenters

Fred Hsu

- Fred Hsu is a distinguished solutions engineer at Arista Networks.
- He leads technical marketing of partner solutions, NetDevOps, public cloud, and Kubernetes.
- Fred has worked in the networking industry for over 20 years and has a Master's degree in Computer Science from the University of Illinois Urbana-Champaign.
- He can be found on Twitter and Github at @fredhsu.

fredlhsu@arista.com

Carl Buchmann

- Carl Buchmann is a Systems Engineer at Arista Networks.
- Having worked in technology for the last 20 years, he's worn a lot of different hats!
- Here at Arista, Carl works on the customer engineering team and co-leads Ansible Working group, contributing to the development of Ansible modules and roles for Arista.
- He can be found on various Slack channels and Github at @carlbuchmann.

carl.buchmann@arista.com

Thomas Grimonet

- Thomas Grimonet is an Advanced Services Consultant at Arista Networks.
- Having worked in technology for the last 20 years, mostly in the network and open-source spaces!
- Here at Arista, Thomas works on the customer engineering team and co-leads Ansible Working group, contributing to the development of Ansible modules and roles for Arista.
- He can be found on Twitter and Github at @titom73.

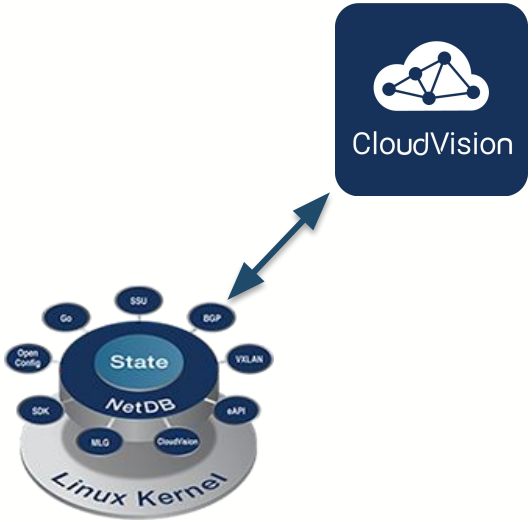
tgrimonet@arista.com

Arista Automation Overview

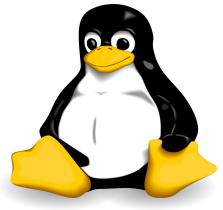
ARISTA

EOS: Consistent Software Driven Foundation

Single Image

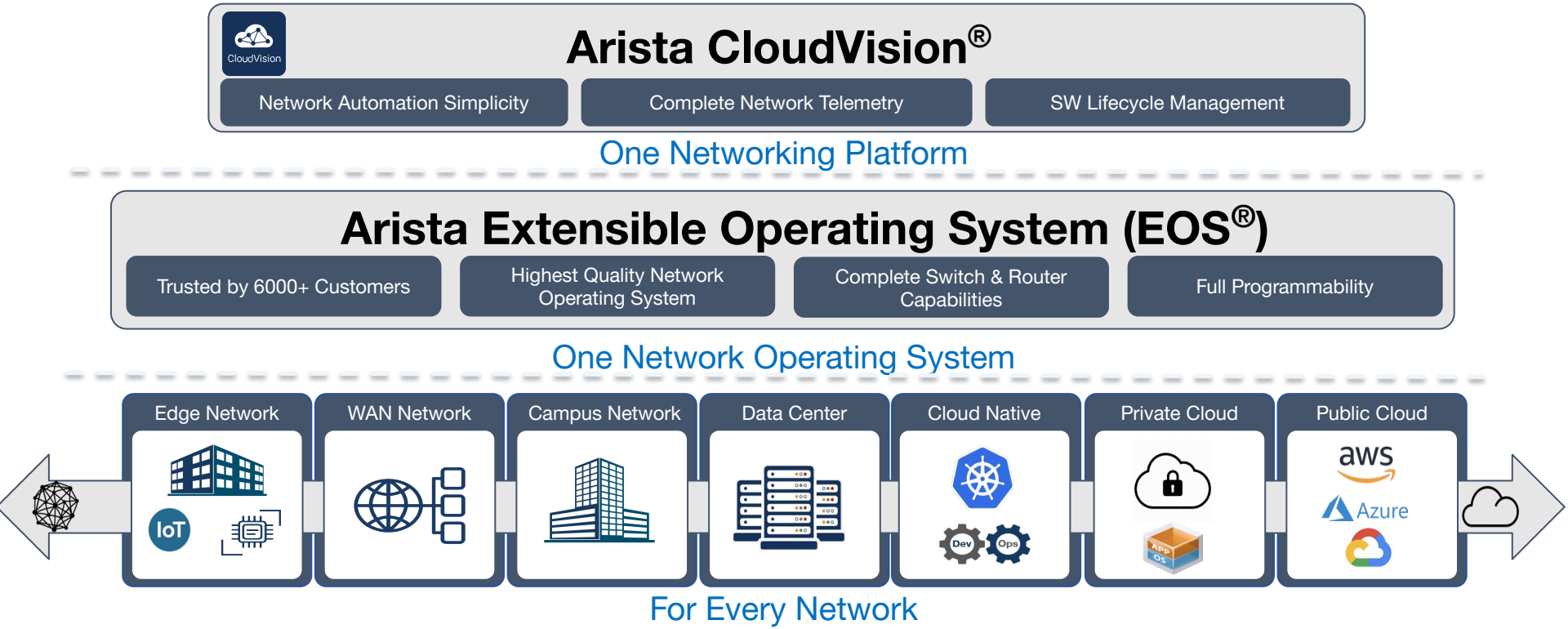


State Oriented



Programmable

Arista: The Software Driven Cloud Networking Company



Approaches to Network Automation

Turnkey NetOps

- Limited solutions today: Focus on legacy models
- Turnkey solution is needed for cloud-automation

ARISTA ☑

- CloudVision

DevOps

- Leveraging Existing tools
- Integration & Customization via scripts

ARISTA ☑

- EOS DevOps toolkits (Python / Ruby / GO)

D.I.Y

- Custom development integration
- Software development resources & approach

ARISTA ☑

- EOS SDK
- eAPI
- CloudVision

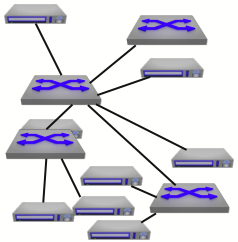
Network Operations Maturity Model

Cloud Experience

Cloud Operations

Cloud Architectures

Anti-Pattern



Patterns

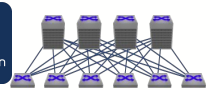
Repeatable design patterns

Runbooks/ITSM

Multi-Vendor/ Open Protocols

Process Automation

CloudVision



Monitoring / Telemetry / Observability

Read-Only Automation

System Automation

Deployment and Workflow Automation

Testing environments

Enforcement

Database driven Source of Truth

Integrated Testing and Toolchains

Configuration Automation

Autonomous

AI-driven Cognitive Ops

Predictive and Proactive Fault detection and correction

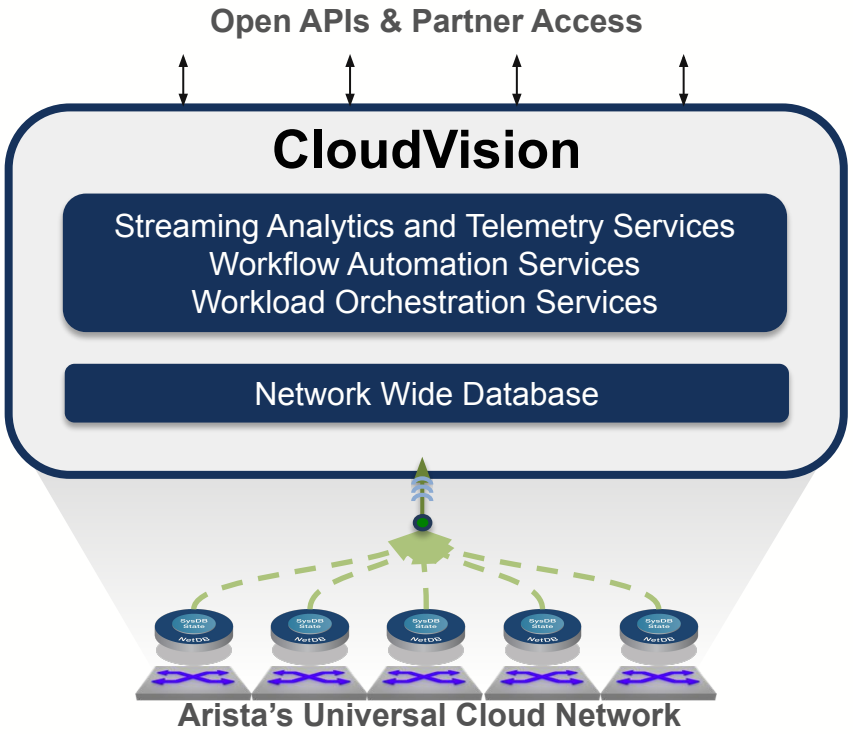
Network Simulation

Years | Quarters | Months | Weeks | Days | Hours

Frequency of Software Upgrades

ARISTA

Arista CloudVision



- **EOS Network Wide Services**
- **Network control point for 3rd party**
- **Turnkey workflow automation**
- **Streaming Analytics and Telemetry**
- **DC, Campus, WiFi, Any Cloud**

CloudVision: Multi-Function NetOps Platform

Automated Deployments

Zero Touch Provisioning,
Hierarchical Config, Extensibility

Real-time Telemetry

Granular state streaming for time-series metrics, flows, and events

Change Controls

Orchestrate network-wide upgrades, rollback and snapshots

CloudVision

Cognitive Analytics

Correlations, trend analysis, predictive algorithms across wired and wireless state, network-wide

Compliance / Risk

Continuously assess, report, and remediate deviations, vulnerabilities, bugs

Security Services

Security policy enforcement,
Policy server integration, Wireless IPS

Data Center, Campus Wired/WiFi, Public Cloud

ARISTA

CloudVision Integration and automation

MODEL #1

- CVP is the source of truth for every configuration
- No variables / Just static form
- Semi-automation with CVP Configbuilder
- Simplicity

MODEL #2

- CVP is the source of truth for every configuration
- Internal/external Repository (Source of Truth for data)
- Automation with CVP Configbuilder

MODEL #3

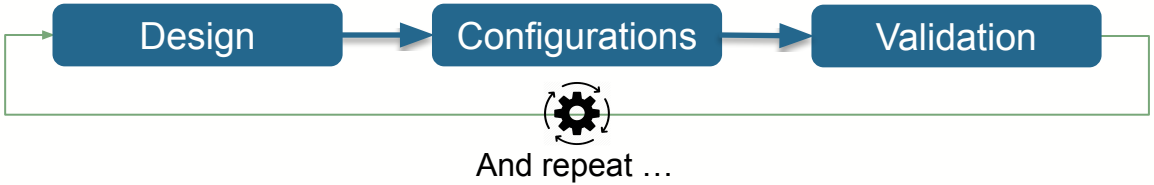
- CVP or an external repo could be the source of truth
- Internal/external Repository (Source of Truth for data)
- Automation with external tool
- Scale

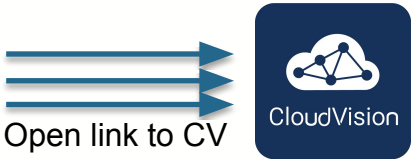
MODEL #4

- CI/CD approach
- CVP manage configuration deployment
- Internal/external Repository (Source of Truth for data)
- Automation with external tool
- Scale and automatic configuration/test deployment

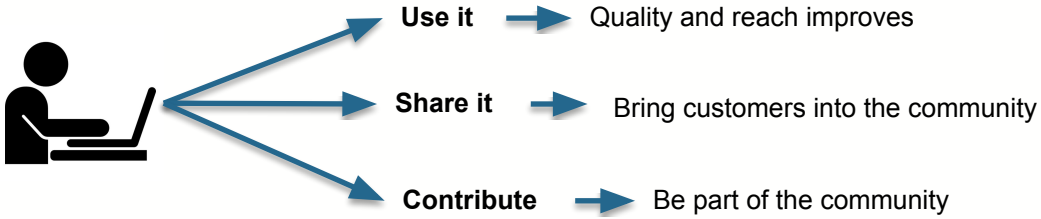
ARISTA

Automation strategy





Arista Customer Engineering



From a generic:

You can do all these things
EOS/CV automation capabilities

To a more concrete:

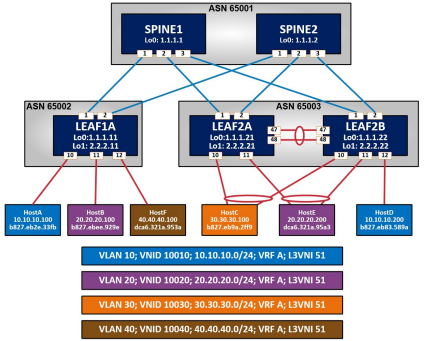
We have done this, and we share it
Providing an opinionated way
of doing automation...
Use/Reuse/Modify for your scenario

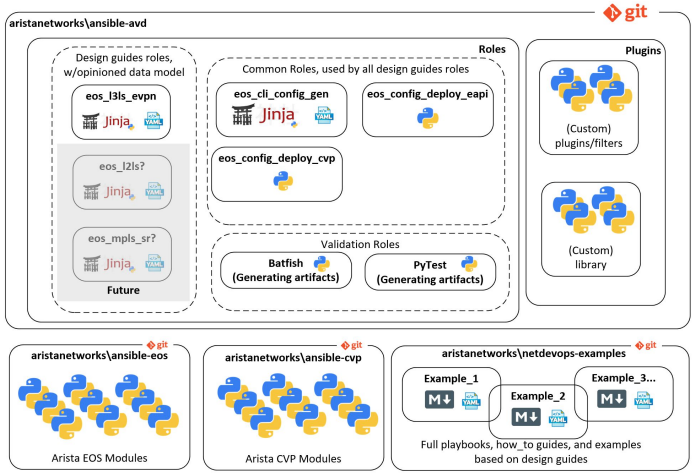
- Empower the customer to self sustain
- Arista intellectual property shared via GitHub under the Apache license

ARISTA

Ansible Collections and Public Git Repositories

- AVD stands for Arista Validated Design
- EVPN Deployment Guide [available here](#).
- Ansible collections:
  - EOS Modules (Foundational Modules)
    - https://github.com/ansible-collections/eos
  - CVP Modules (Foundational Modules & Roles)
    - https://github.com/aristanetworks/ansible-cvp
  - Arista Validated Design Roles (Opinionated Roles & Modules)
    - https://github.com/aristanetworks/ansible-avd
- Arista NetDevOps Community Repos:
  - https://aristanetworks.github.io/netdevops-examples/
  - https://github.com/arista-netdevops-community

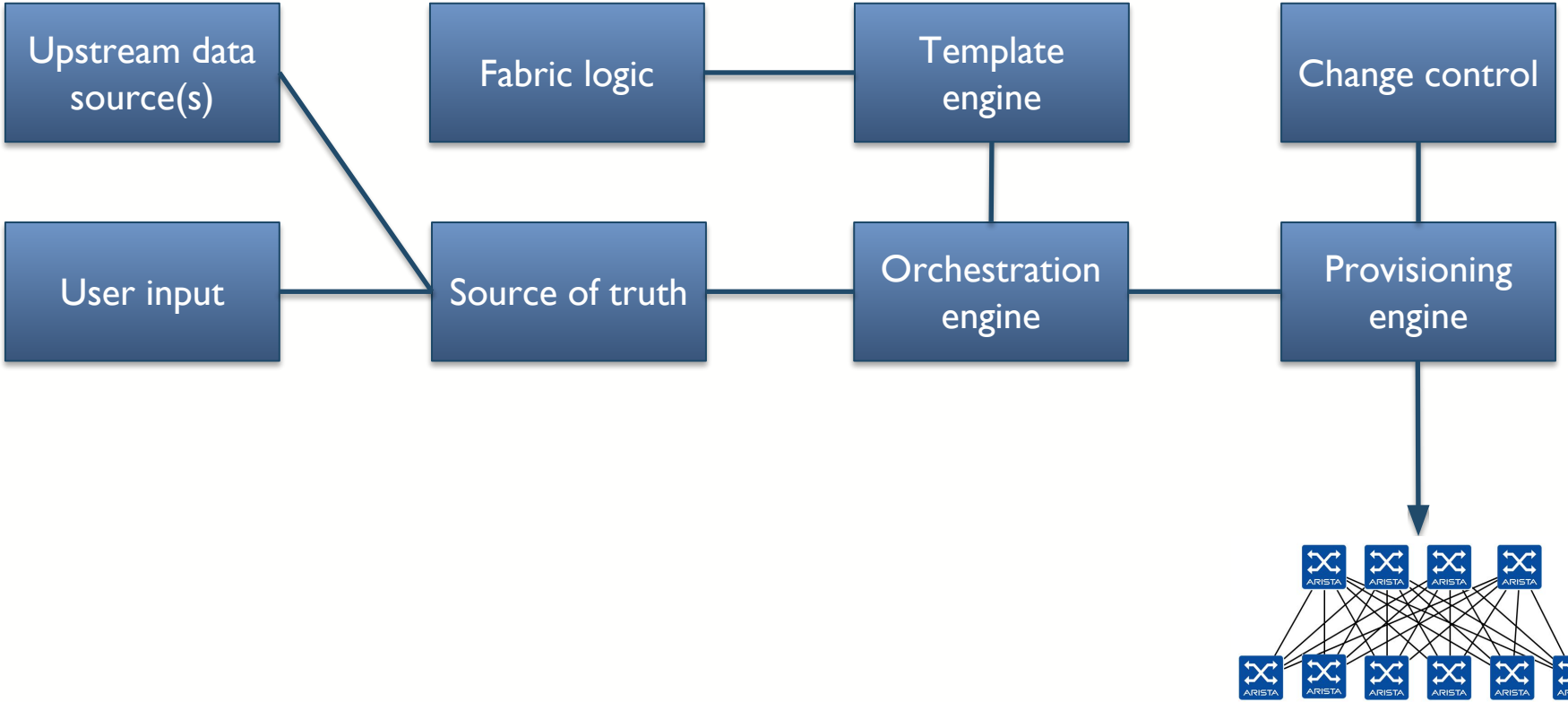




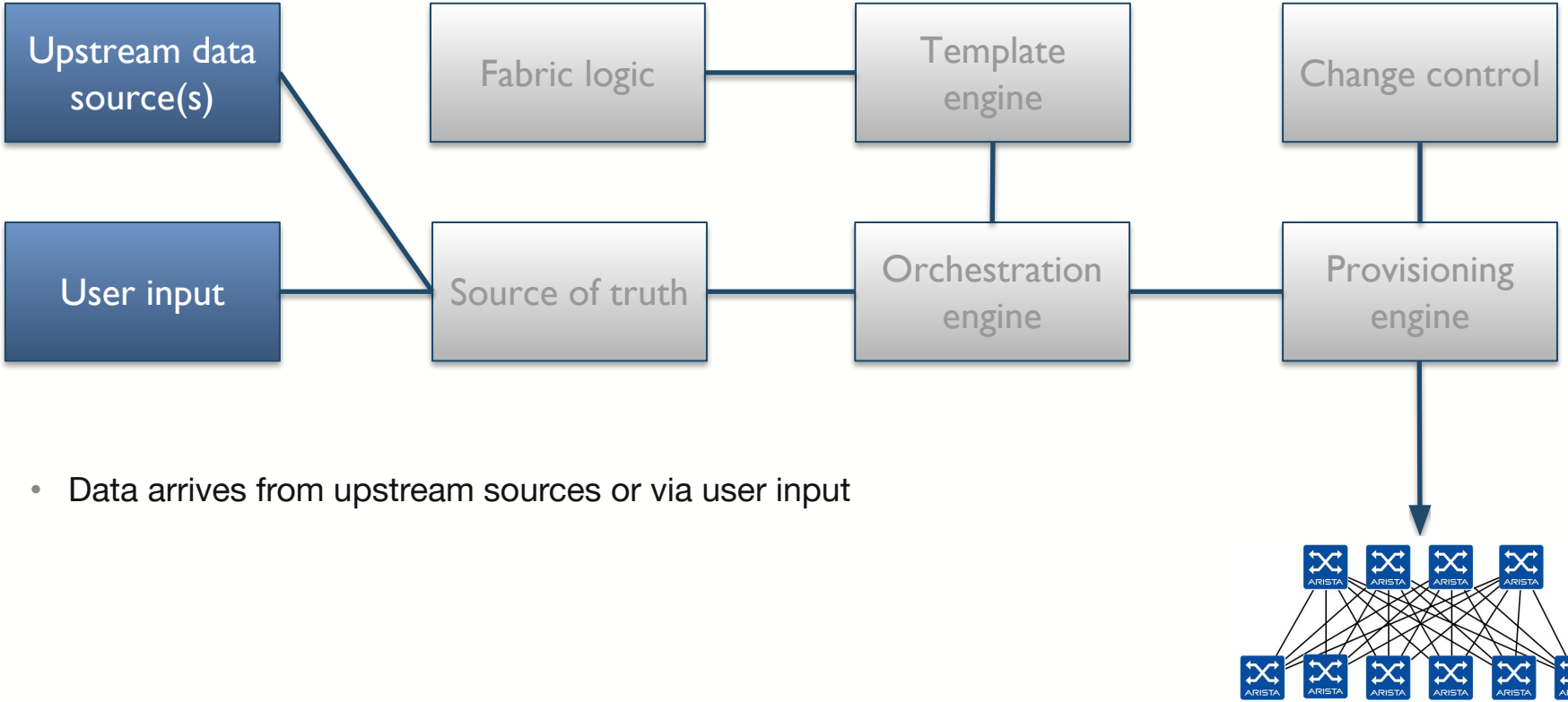
Automation Building Blocks

ARISTA

Provisioning Building Blocks

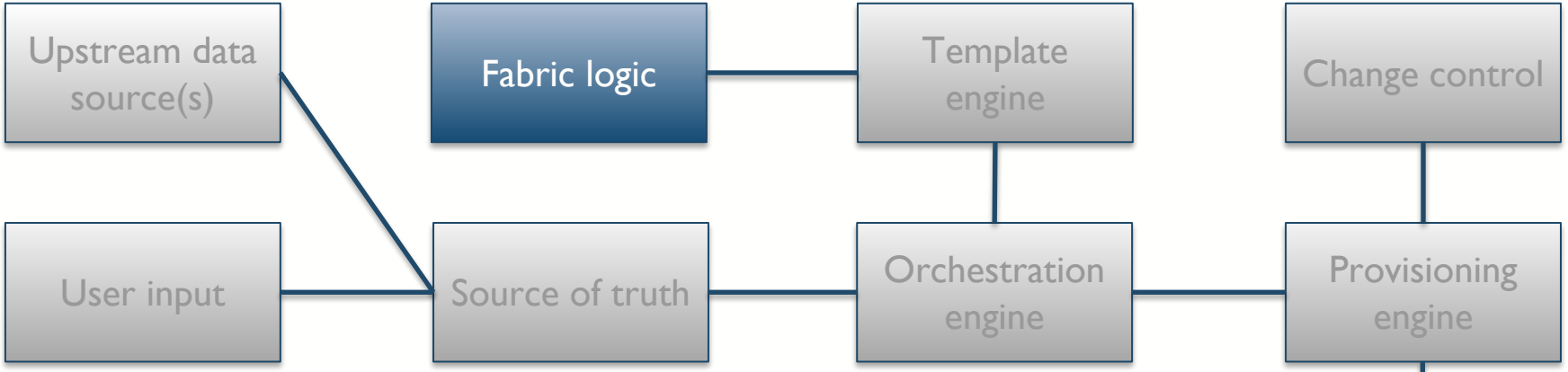


Provisioning Building Blocks: Input Data

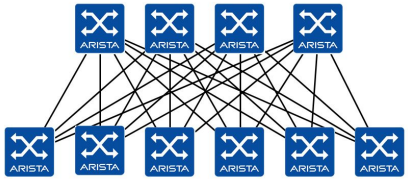


- Data arrives from upstream sources or via user input

Provisioning Building Blocks: Fabric Logic

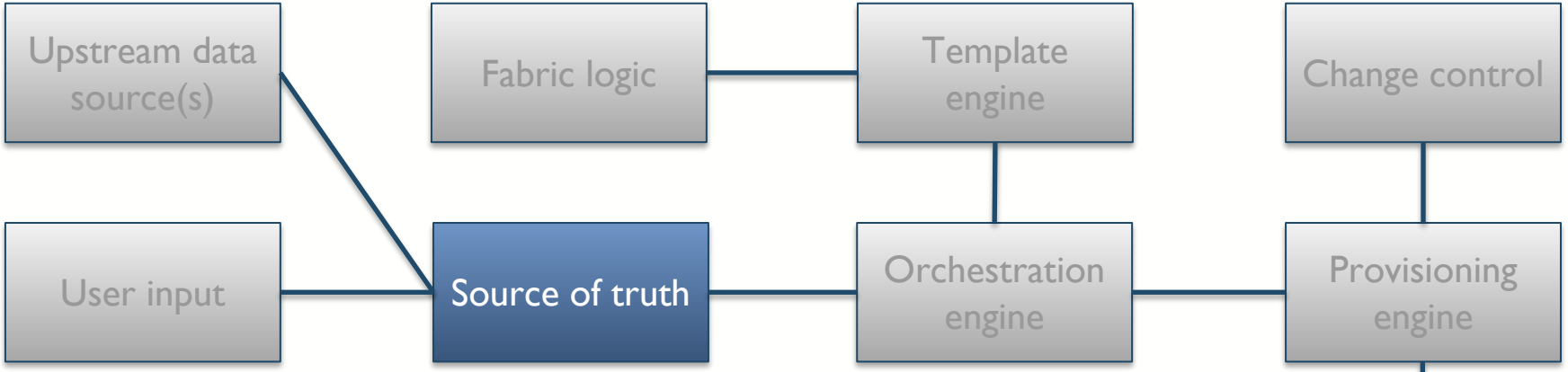


- Data is processed based on Fabric logic.
- Provides abstraction layer, establishing repeatable patterns.
- Example: Allocate P2P addresses from input subnet

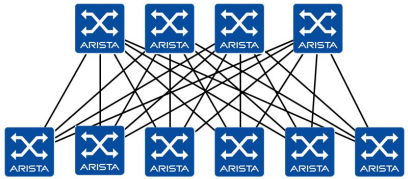


ARISTA

Provisioning Building Blocks: Source of truth

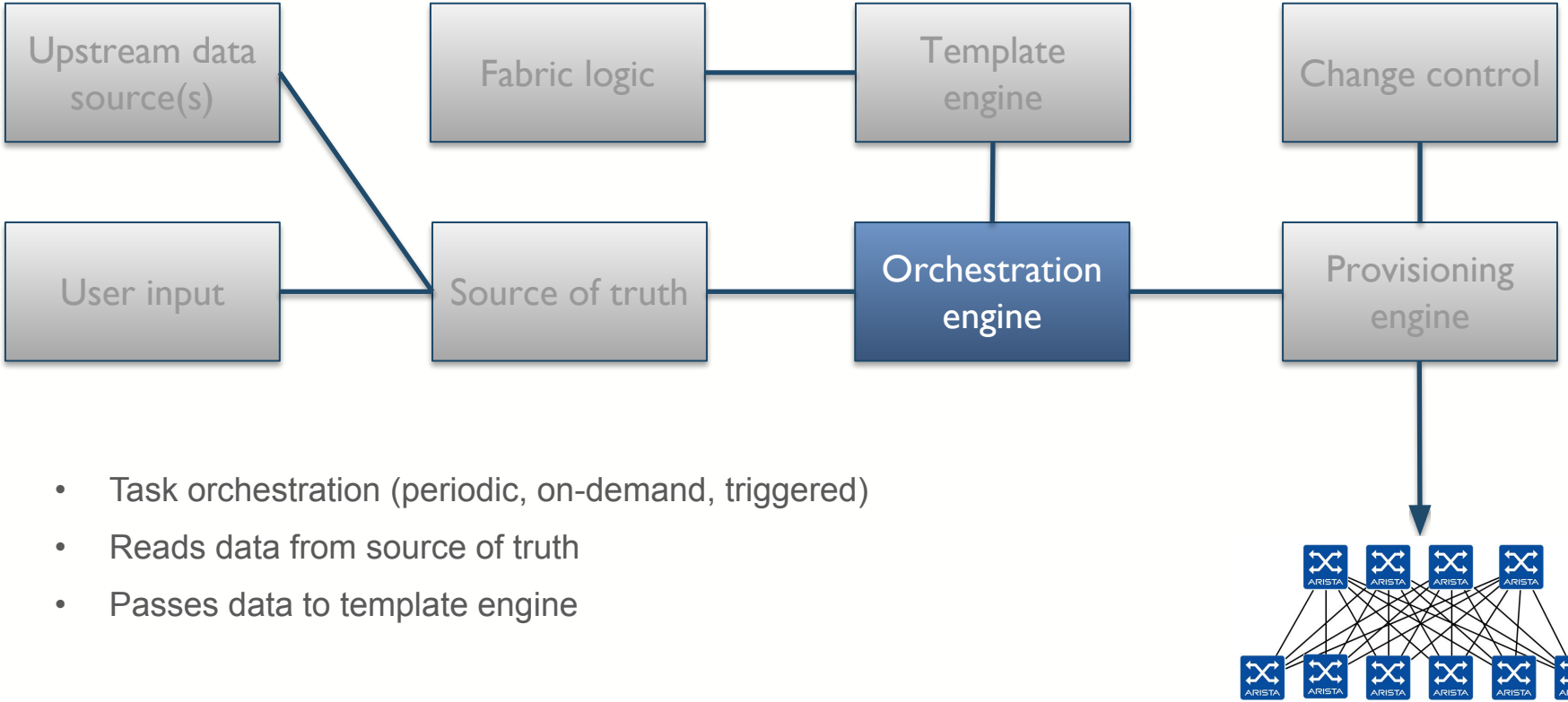


- Authoritative data source
- Input data validation
- Structured data model
- API for Create, Read, Update, and Delete (CRUD)
- Can be multiple sources of truth (one per domain)



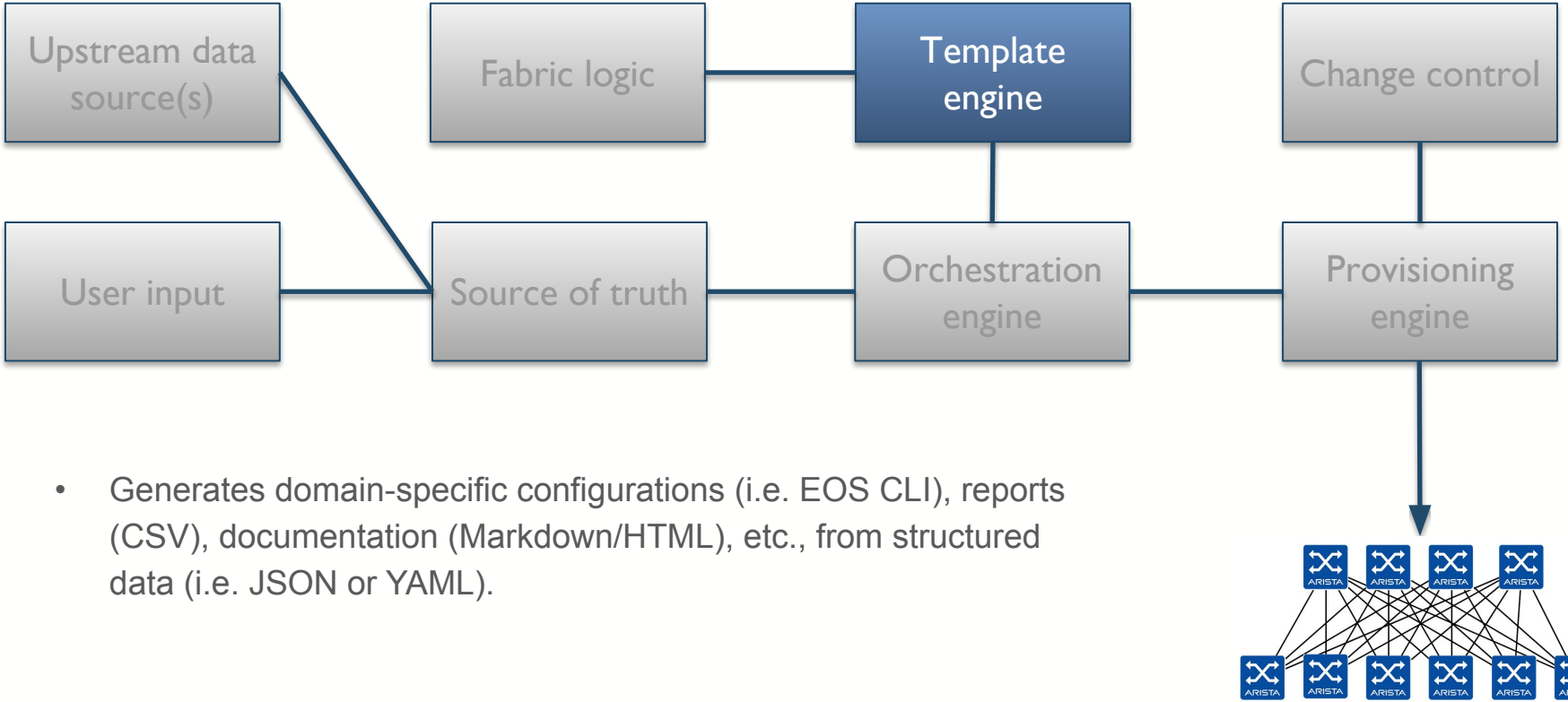
ARISTA

Provisioning Building Blocks: Orchestration Engine



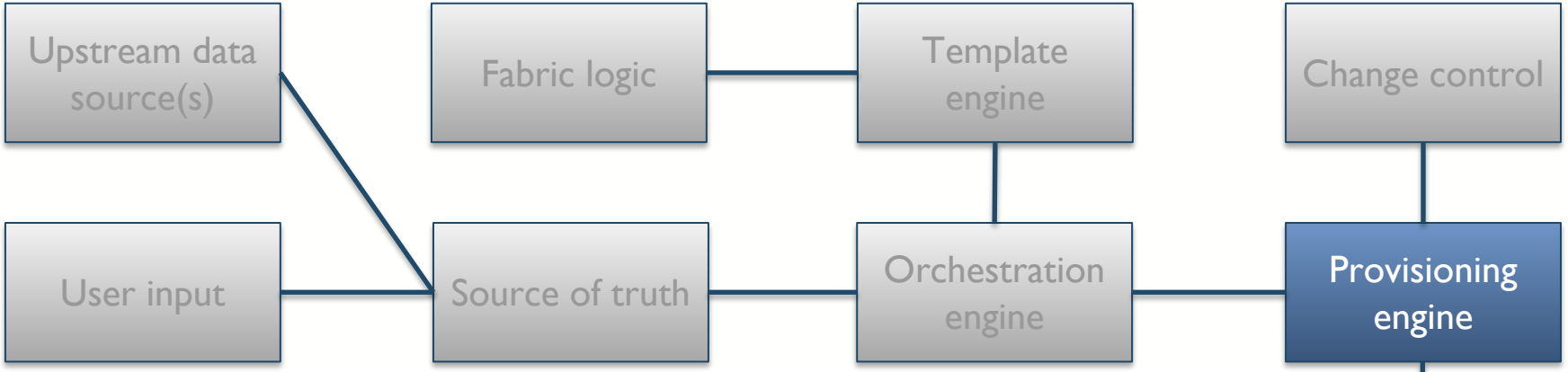
- Task orchestration (periodic, on-demand, triggered)
- Reads data from source of truth
- Passes data to template engine

Provisioning Building Blocks: Template Engine

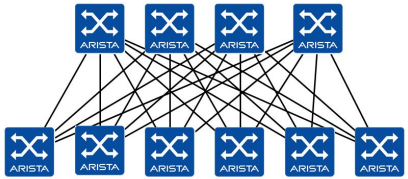


- Generates domain-specific configurations (i.e. EOS CLI), reports (CSV), documentation (Markdown/HTML), etc., from structured data (i.e. JSON or YAML).

Provisioning Building Blocks: Provisioning Engine

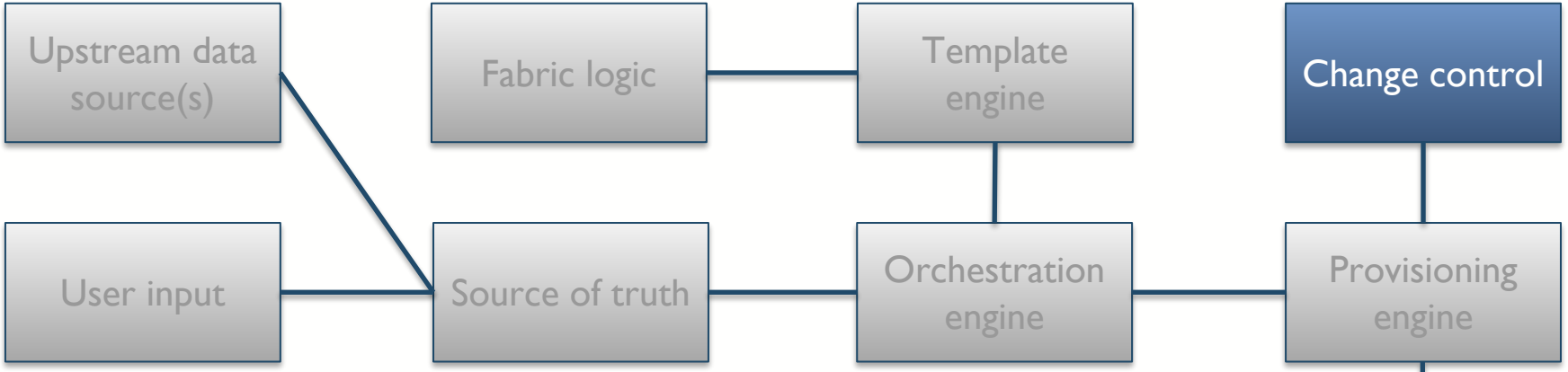


- Hierarchical provisioning model (containers and configlets)
- Configuration validation and compliance
- Communicates south-bound to Arista switches via eAPI

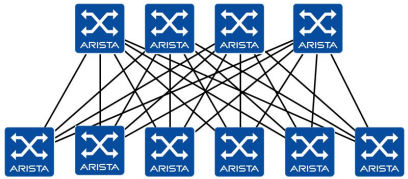


ARISTA

Provisioning Building Blocks: Provisioning Engine

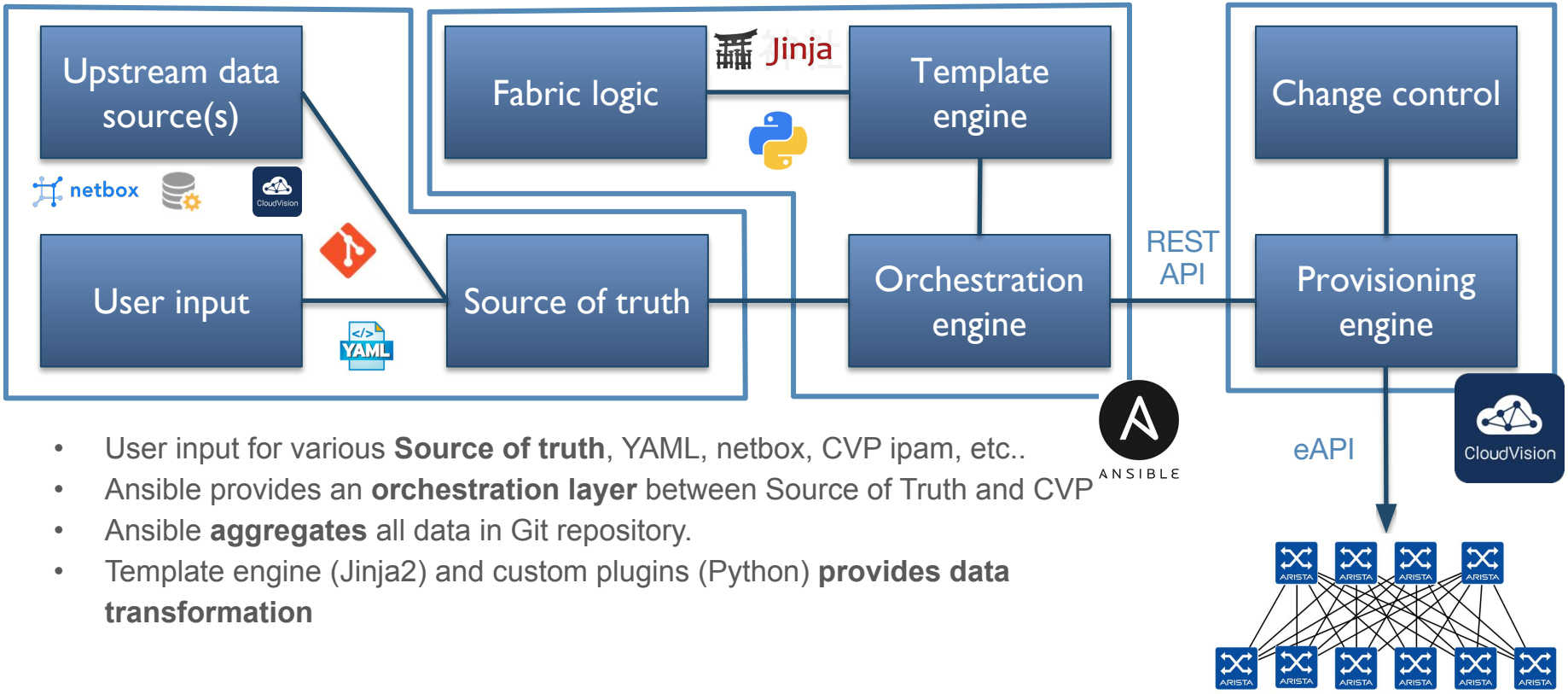


- Change review, approval, and execution
- API for integration with external systems (ex: ServiceNow)



ARISTA

Provisioning Building Blocks



- User input for various **Source of truth**, YAML, netbox, CVP ipam, etc..
- Ansible provides an **orchestration layer** between Source of Truth and CVP
- Ansible **aggregates** all data in Git repository.
- Template engine (Jinja2) and custom plugins (Python) **provides data transformation**

ARISTA

Ansible Collection: ansible.cvp

Ansible & CloudVision Integration

- CloudVision With Ansible:
  - Ansible collection to manage devices via CloudVision Platform.
  - Available on GitHub: https://cvp.avd.sh/
  - Support CVP from version 2018.2.x and onward.
  - Provide a set of foundational modules for CV:
    - **arista.cvp.cv_facts** - Collect CVP facts from server like list of containers, devices, configlet and tasks.
    - **arista.cvp.cv_configlet** - Manage configlet configured on CVP.
    - **arista.cvp.cv_container** - Manage container topology and attach configlet and devices to containers.
    - **arista.cvp.cv_device** - Manage devices configured on CVP
    - **arista.cvp.cv_task** - Run tasks created on CVP.

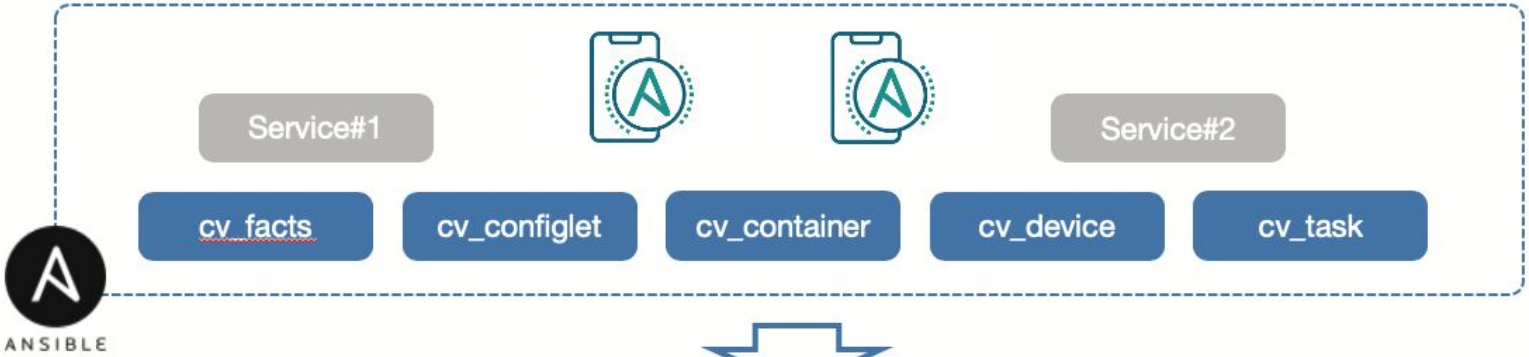
How Ansible and CloudVision work together!

Single pane for configuration generation
Configuration for all devices part of a service: Switches/FW's/ LB's

Idempotency model
running-configuration is based on desired state

Generate configlet content
Can import data from 3rd party tools
Modules feed information about the current status

Deploy configlets to CV
Modules apply configuration transparently to the CV API



Consistency
Pre/post snapshots
Push intended configuration with correct ordering

Control
Pushed configuration (optionally) requires approval before execution

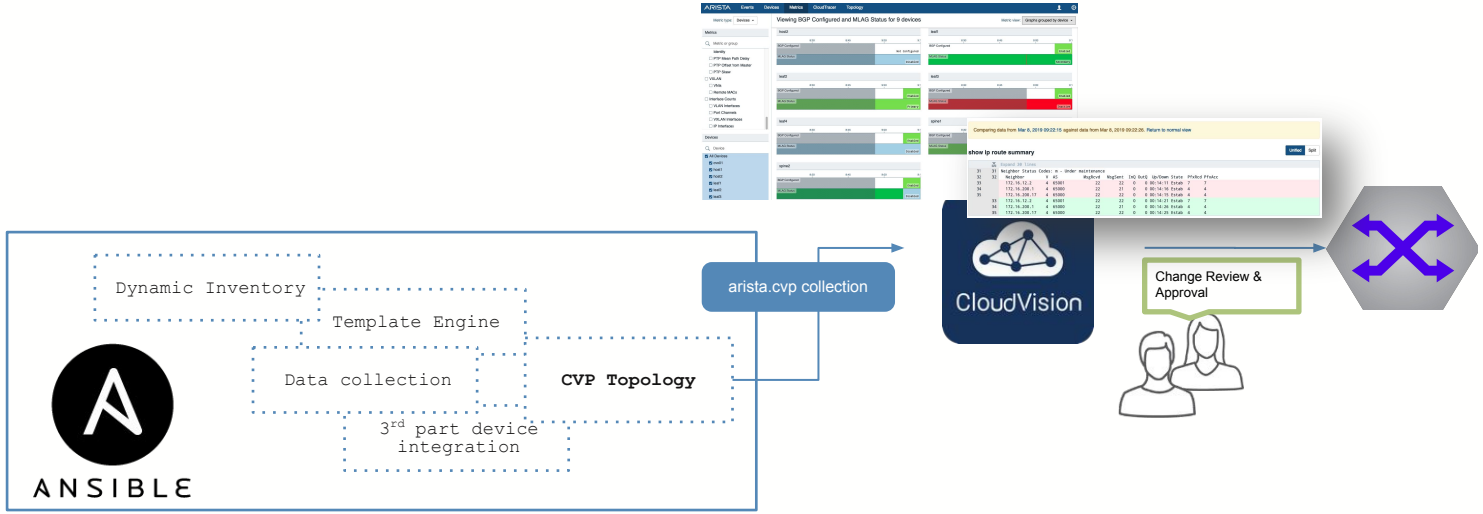
Full visibility
Trace changes with real time telemetry of all the variables

Synchronization
Synchronize configurations across multiple CV instances

ARISTA

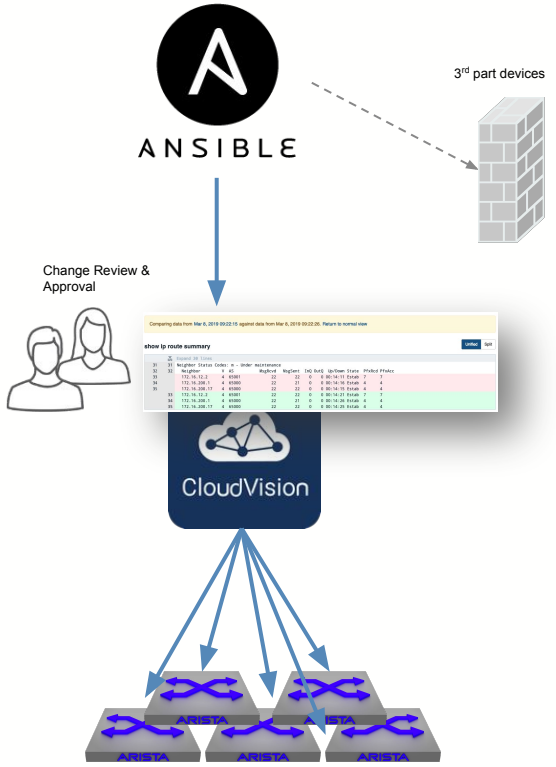
Ansible & CloudVision Integration

- CloudVision With Ansible:
  - Use a single pane for configuration deployment
    - Idempotency model: running-configuration is based on desired state
  - Use Telemetry for change management
  - Use CVP Change Management to trace changes and state comparisons



Ansible & CloudVision Integration

- CloudVision With Ansible:
  - Easy to fit with any existing workflow:
    - Use `cv_configlets` instead of `eos_config` to deploy configuration
    - Ansible can manage only a subset of configlets:
      - all your existing configlets can remain unchanged
    - Idempotency - you do not need to generate complete configuration: CVP will do
  - Better change-management and visibility
    - All changes are pushed to CVP
    - Change-Control helps for deploying changes
  - Easy to integrate in a complete workflow:
    - Not just EOS device configuration
    - Extendable to servers / firewalls / …



How Ansible and CloudVision work together!

Ansible	CloudVision
Builds configuration for all devices.	**Visualizes diff** between the current and the target configuration.
Pushes configlets to CloudVision.	**Delivers** EOS images, extensions and patches.
Generates **tasks** on CloudVision.	**Pushes** the changes to the network fabric with integrated **change control**
Orchestrates with pre and post validation.	**Monitoring and Troubleshooting** (compliance dashboard, historical tables, telemetry)
	Zero Touch Provisioning

Ansible Collection: ansible.avd

Ansible as Configuration Builder

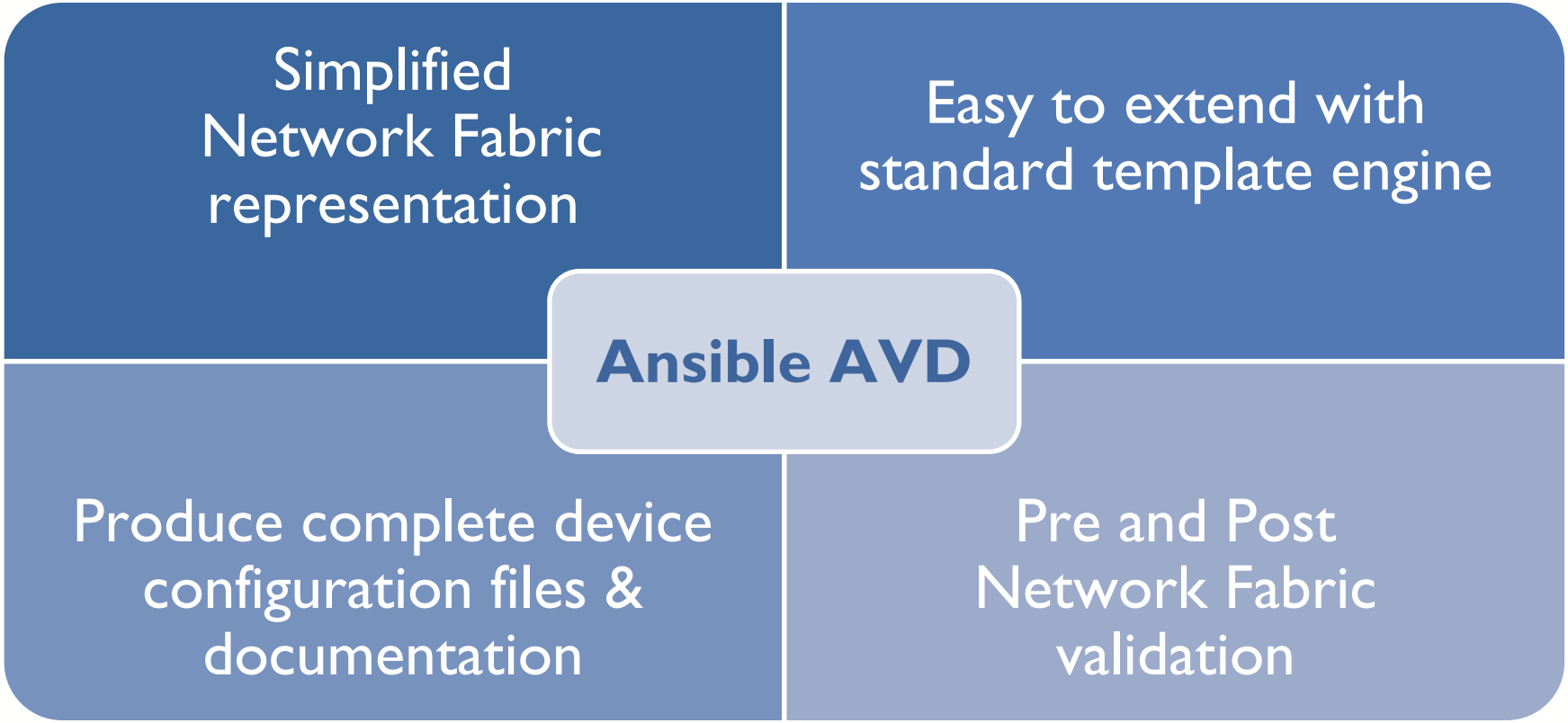
- Ansible Arista Validated Design (AVD)
  - Available on Github: https://github.com/aristanetworks/ansible-avd/
  - Documentation: https://www.avd.sh/
  - **Roles:**
    - **arista.avd.eos_l3ls_evpn** - *Opinionated Data model for deployment of L3 Leaf and Spine Fabric with VXLAN data-plane with an EVPN Control plane.*
    - **arista.avd.eos_cli_config_gen** - *Generate Arista EOS cli syntax and device documentation.*
    - **arista.avd.eos_config_deploy_cvp** - *Deploys intended configuration via CloudVision.*
    - **arista.avd.eos_config_deploy_eapi** - *Deploys intended configuration via eAPI.*
    - **arista.avd.cvp_configlet_upload** - *Uploads configlets from a local folder to CloudVision Server.*
    - **arista.avd.eos_validate_state** - *Validate operational states of Arista EOS devices*

Ansible as Configuration Builder

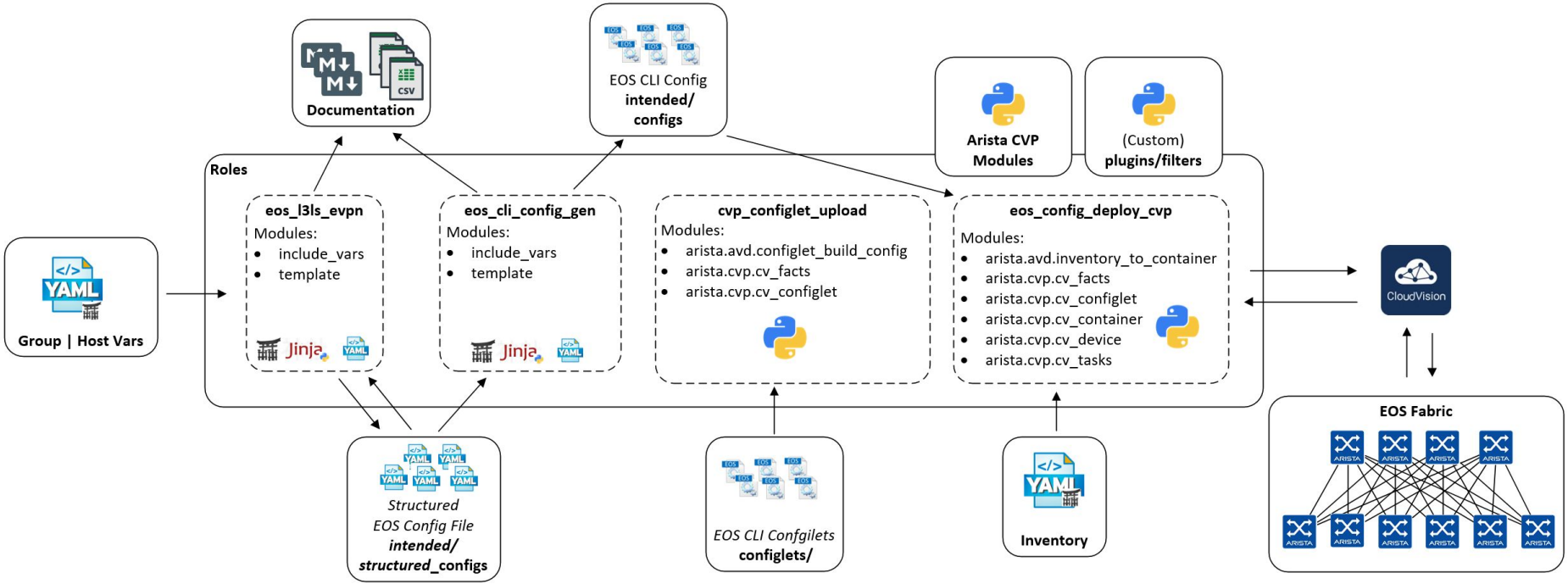
- **Filter Plugins:**
  - **arista.avd.list_compress** - *provides the capabilities to compress a list of integers and return as a string.*
  - **arista.avd.natural_sort** - *sort a list or a dictionary of integers and/or strings that contain alphanumeric characters naturally.*
  - **arista.avd.generate_esi** - *Transform 3 octets ESI to 5 Octet ESI*
  - **arista.avd.generate_lacp_id** - *Transform 3 octets ESI to lacp id*
  - **arista.avd.generate_route_target** - *Transform 3 octets ESI to route-target*

- **Modules:**
  - **arista.avd.configlet_build_config** - *Build configuration to publish configlets on* Cloudvision
  - **arista.avd.inventory_to_container** - *Transform information from ansible inventory to be able to provision CloudVision Platform using arista.cvp collection and its specific data structure.*

Ansible As Configuration Builder

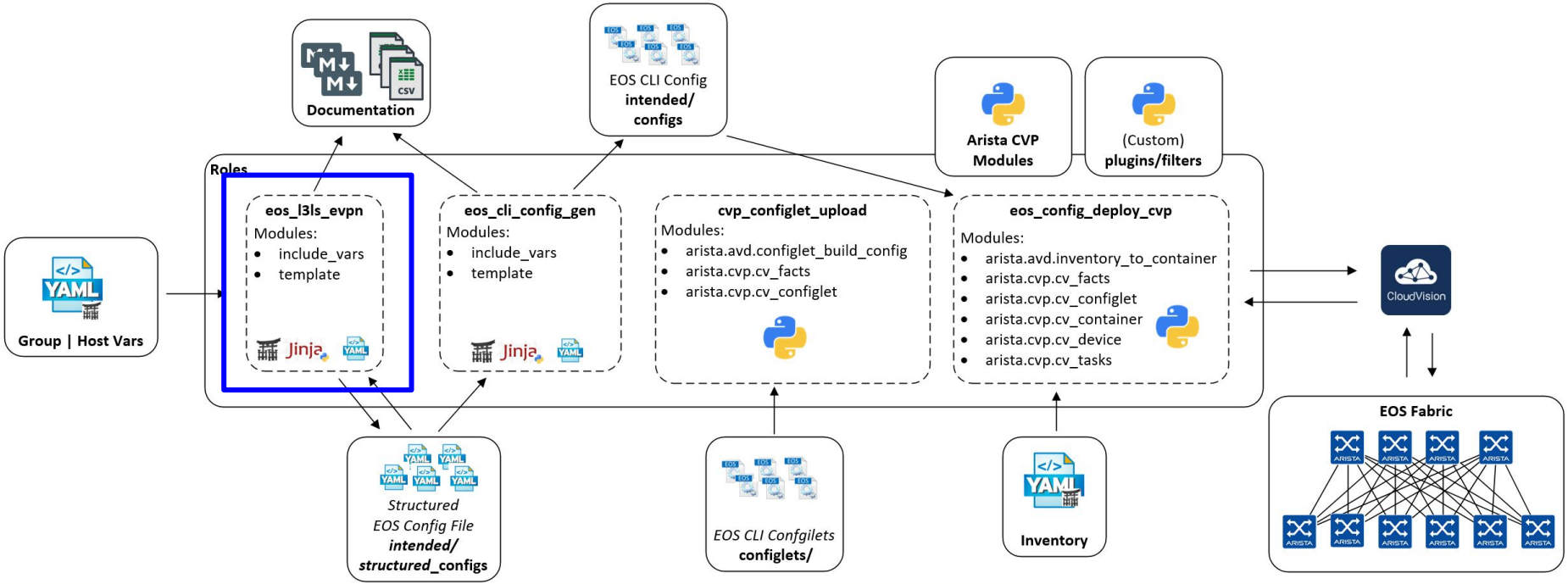


Arista AVD - CVP Deployment



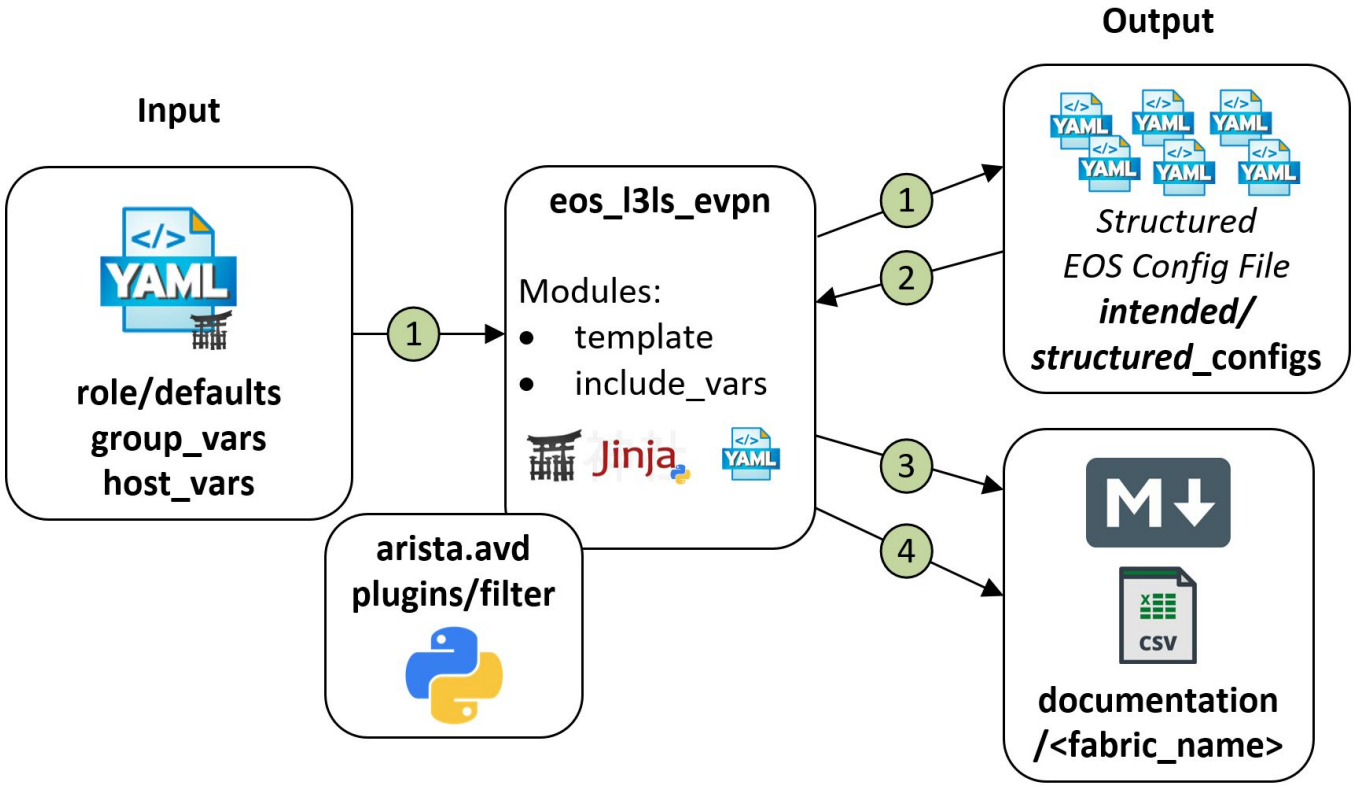
ARISTA

Arista AVD - CVP Deployment

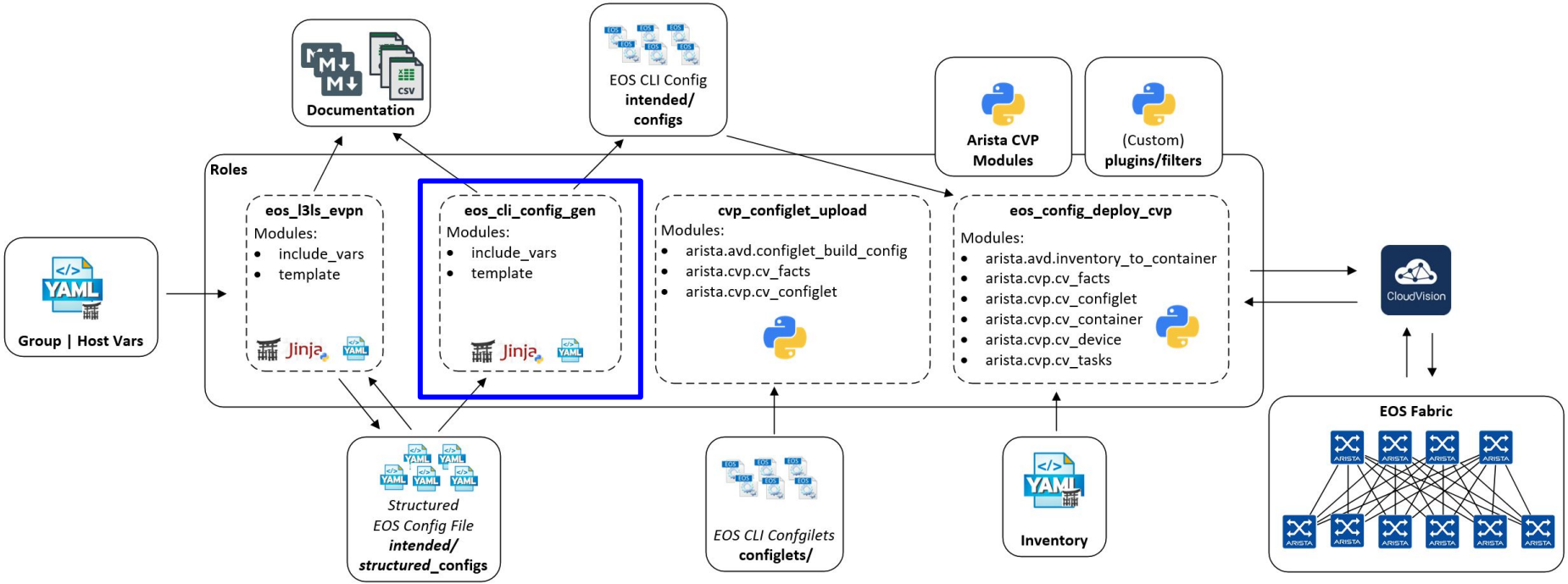


ARISTA

Role: eos_l3ls_evpn

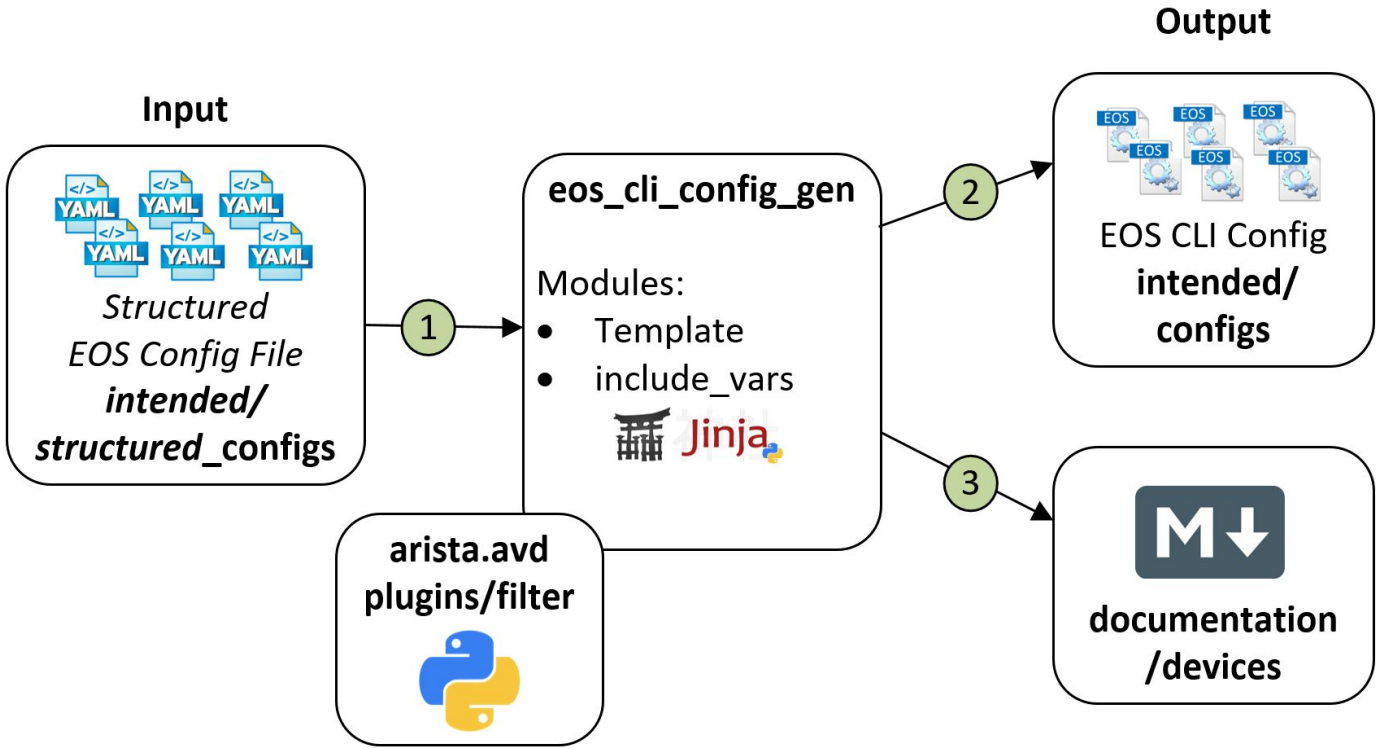


Arista AVD - CVP Deployment

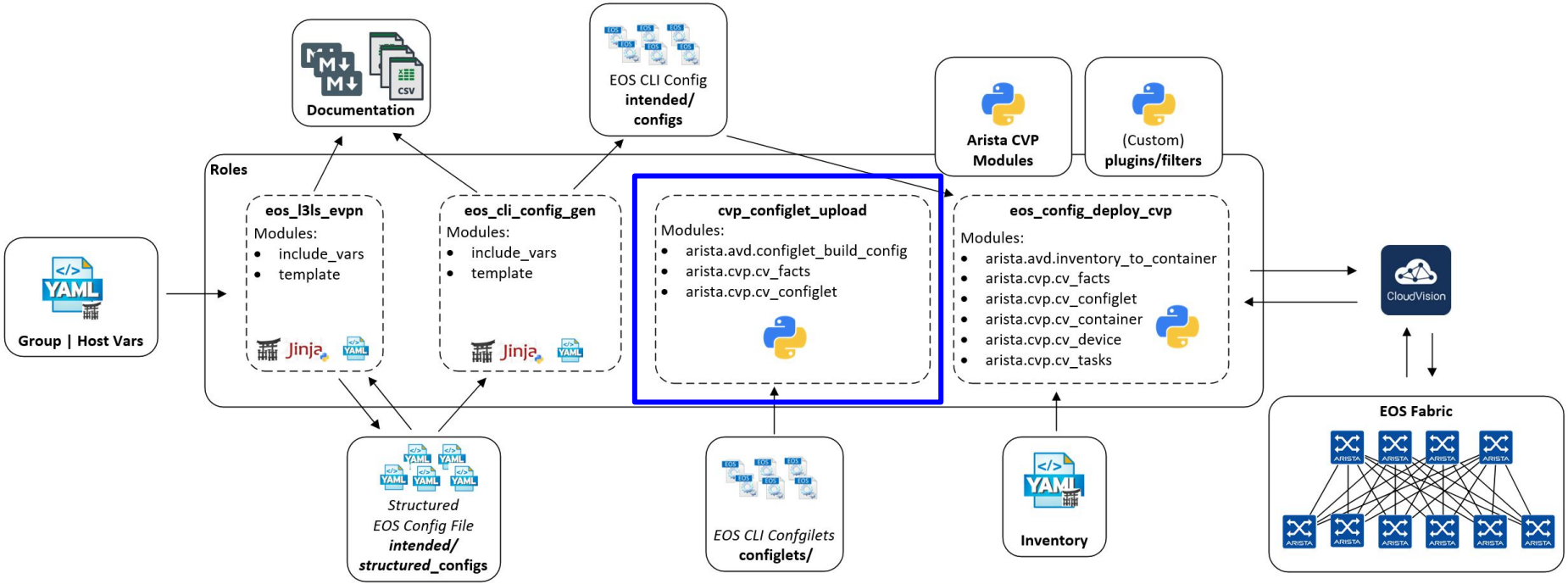


ARISTA

Role: eos_cli_config_gen

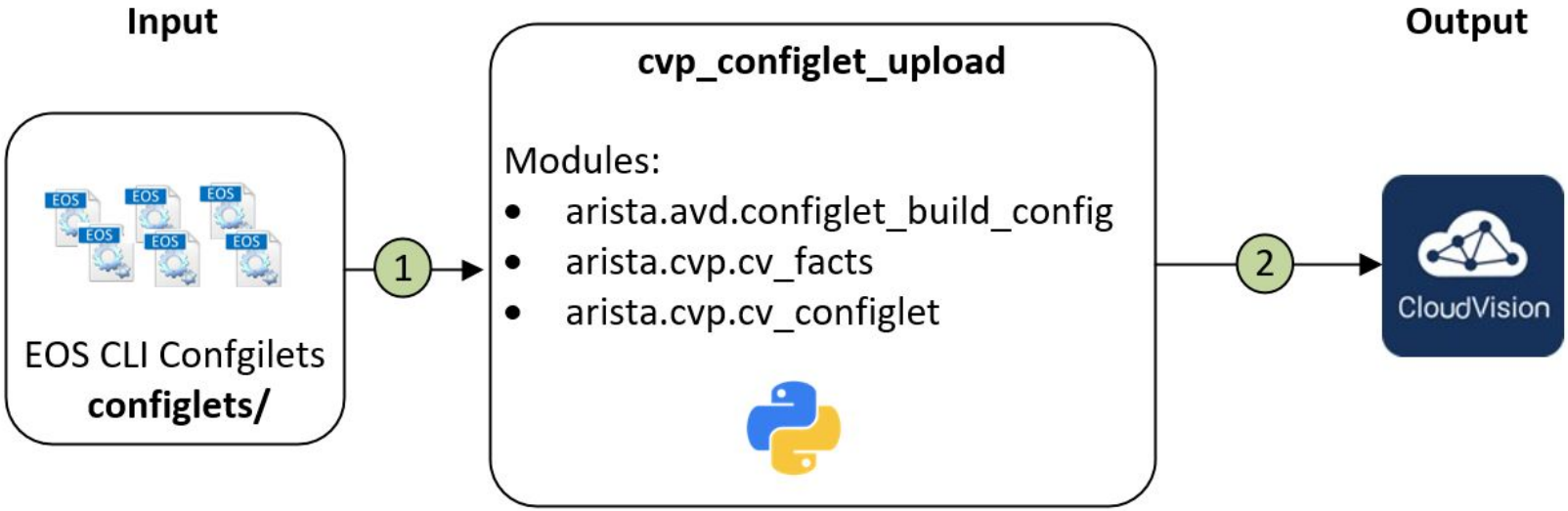


Arista AVD - CVP Deployment

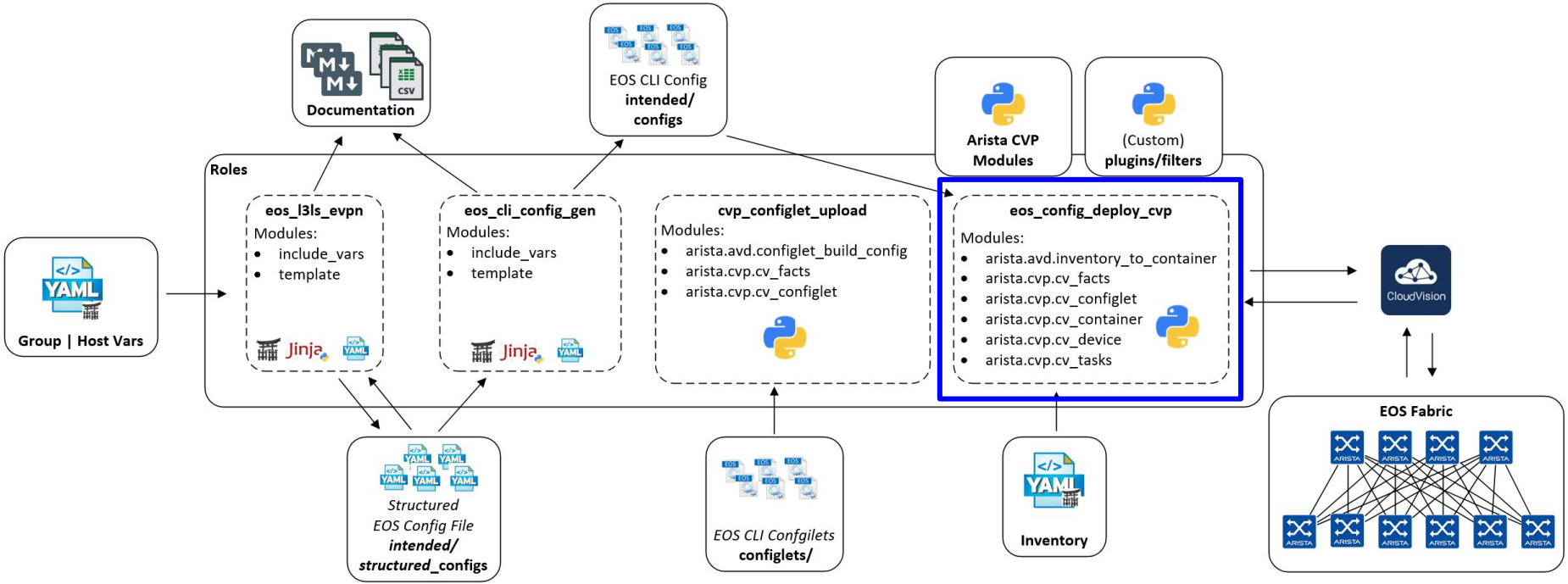


ARISTA

Role: cvp_configlet_upload

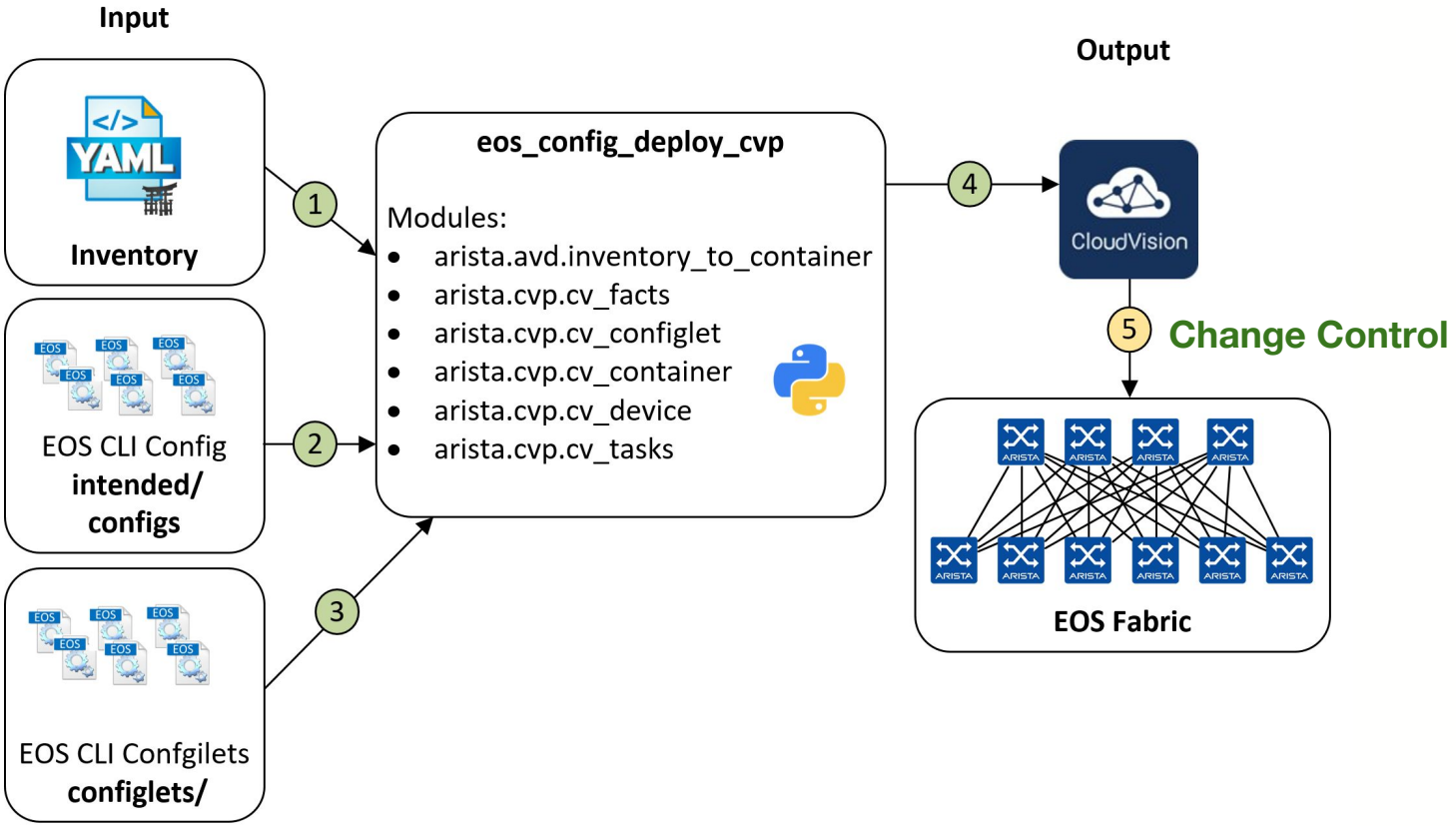


Arista AVD - CVP Deployment

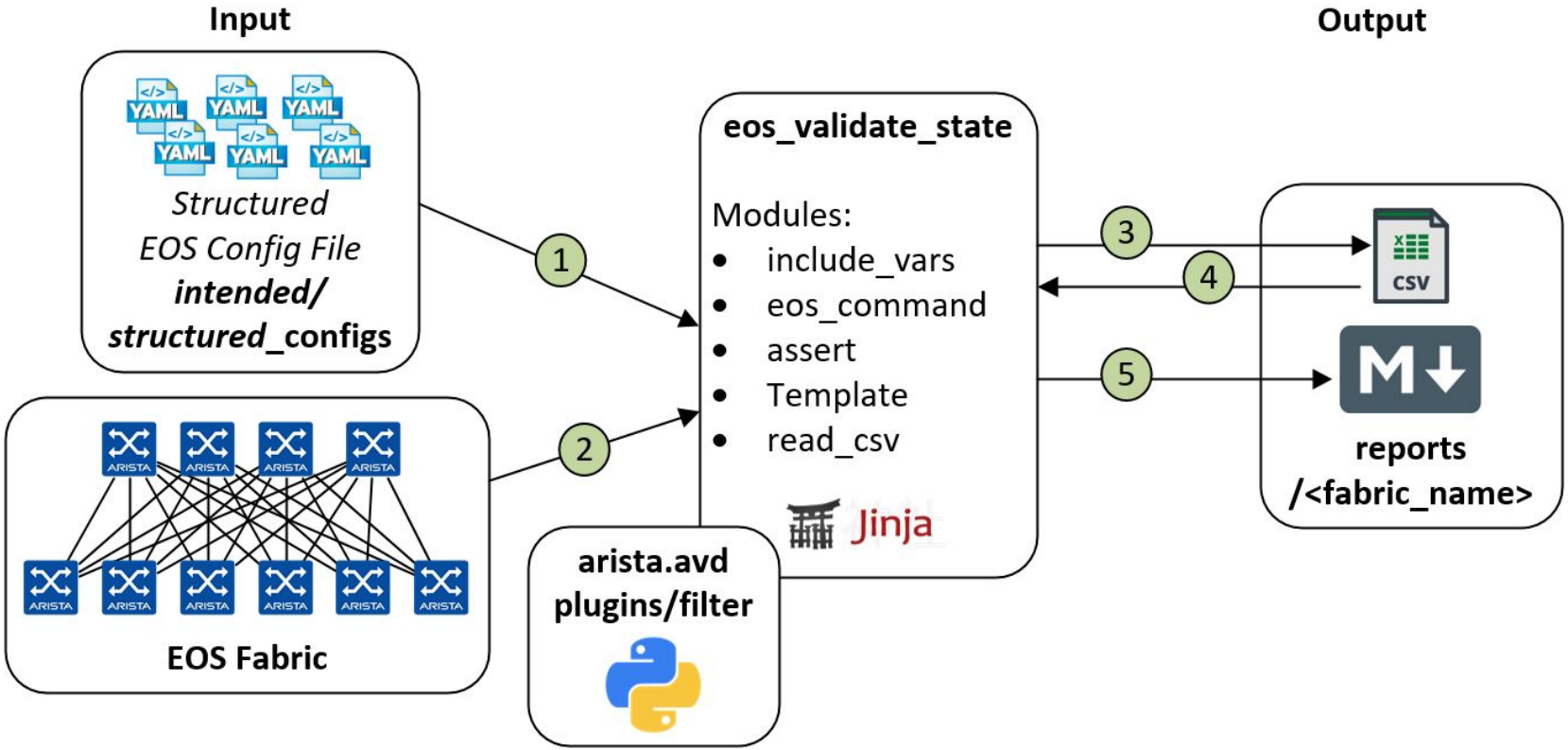


ARISTA

Role: eos_config_deploy_cvp



Role: eos_validate_state

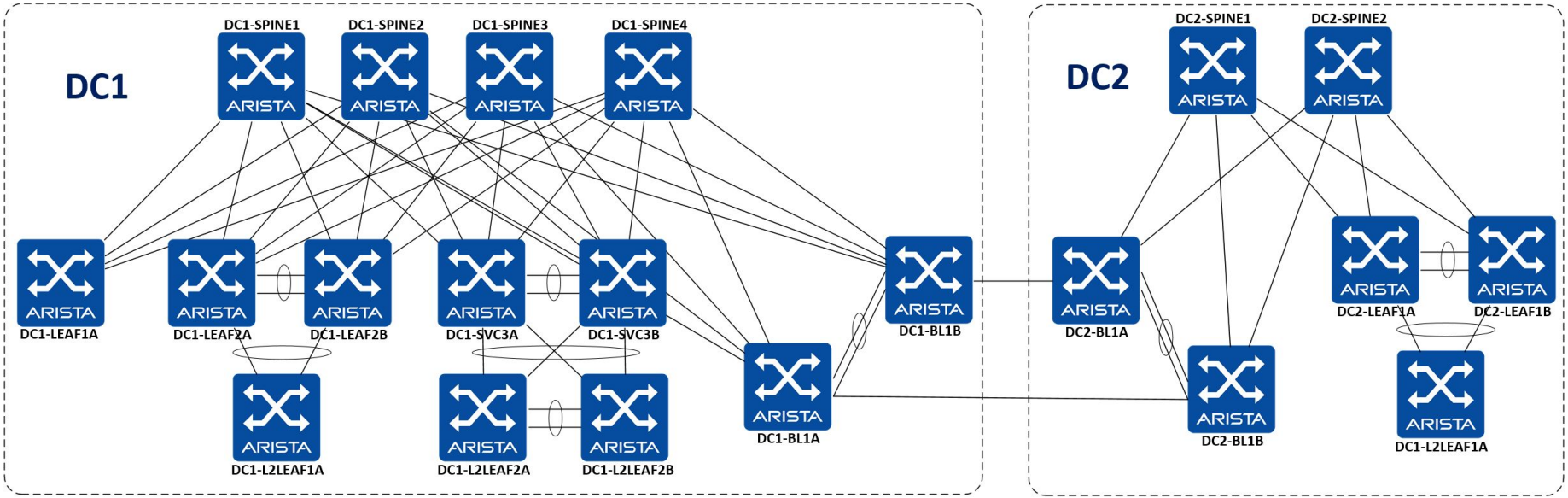


ARISTA

Ansible Collection: Demo
Overview

ARISTA

ipSpace Lab Topology



ARISTA

Getting Started with AVD

- Setting your Development Environment:
  - https://www.avd.sh/docs/installation/setup-environement/
- Leveraging AVD with Git Methodology:
  - Allows for customization of AVD templates, and contributing.
  - https://www.avd.sh/docs/installation/setup-git/
- Your First AVD Project
  - Work with ipSpace Webinar Demo:
    - https://github.com/arista-netdevops-community/ipspace-webinar-september15-2020
  - Build your own!
    - https://www.avd.sh/docs/how-to/first-project/

Ansible Collection: Demo
Provision Border Leafs in DC1 and DC2

ARISTA

Ansible Collection: Demo
Provision DCI

ARISTA

Ansible Collection: Demo
Validate State

ARISTA

Ansible Collection: Demo
Provision L2 and L3 Network Services

ARISTA

Ansible Collection: Demo
Validate State

ARISTA

Thank You