ESnet’s Orchestration Perspective

Marc Körner
Software Engineer
Orchestration and Core Data
Energy Sciences Network (ESnet)
Lawrence Berkeley National Laboratory

I2 TechEX
Minneapolis, MN
August 2023
ESnet in numbers:

- Thousands of miles of fiber cables, including transatlantic cables
- 380 locations with racks and equipment to track
- 346 Core links between routers
- 300 Customer facing Interfaces, 123 of which are 100G
- Multi Platform environment with lots of interoperability needs
What is orchestration?

Orchestration is defined as the automated arrangement, coordination, and management of computer systems, middleware, and services within the network.

-- Wikipedia

Workflow management implementing a repeatable pattern of steps for user-based interactions within our operations support system;

Automated service provisioning to programmatically change configuration across network, compute, and application resources.
Why is Orchestration important?

- Consistent and unified configurations
- Eliminate human errors
- Enhance network reliability, resiliency, and robustness
- Allows to manage a larger network
- Heterogeneous equipment
- Better use of human resources
- Planning safety
ESnet Network Orchestration and Automation
(1) Design network and import bulk data
(1) Assigned task

(2) Plan and populate service

(3) Instantiate service

ESDB
Orch
NSO

ESnet Network

Stardust

Joe the Planner

Eric the Engineer
(1) Fetch service data

(2) Apply network intent and create services

(3) Apply network configuration

(4) Update service data
NSO features used by ESnet

Architectural features:
- Device agnostic configuration
- Service centric abstraction
  - configure 1, 2, n devices

Other stuff which comes in handy:
- Data verification
- Service decommissioning and rollbacks
ESnet’s NSO implementation in numbers

Service models: 33

Services instantiated:

- Devices: 325
- System: 155
- Port: 1705
- BBL: 199
- Bridge: 401
- Host: 161
- LSP: 151
- VPLS: 1
- L3-Interface: 923
- L3-Customer: 277
- L3-Peer: 252
- L3-Transit: 369
Base-config ask

• Pushing base config to routers
• Combination of vlans, filters, etc.

Personal take on things:
• Pushing base config ahead of time => traditional thinking model
• Intent based networking => we deploy config when we need / use it
• Change of thinking is required IMHO
Service audit / architecture assessment

- Monolithic services with high complexity
- Validation complexity
  - SAPs are getting validated over multiple services
- Same functionality is getting defined in multiple services
  - IRB interfaces are getting defined in BBL and MPR baseconfig
  - prefix-lists are getting defined in L3, MPR baseconfig, DNS, NTP, syslog, etc.
- Cross dependencies between “higher layer” services
  - Port service has fields for defining its usage - link-type: BBL, VPLS, host, etc.
- Humongous services models and config templates for core services like BBL and L3
- High Test complexity
Investigate new architectural approach / model with the following objectives

- Clear functional separation and abstraction
- Reduce redundant validation and simplify code complexity
- Dependency hierarchy (dependencies are only pointing downwards)
- Object oriented programming based service approach
3-Tier service architecture

1. Services which are directly translate into configuration concepts and build the first device independent abstraction layer
2. Service which build basic service abstraction concepts out the composition of tier 1 services (might introduce additional config as glue between tier 1 services)
3. Services which are composed out of a combination of tier 1 & 2 services and build an operational foundation for ESnet’s network operation and customer facing portfolio
Tier 1 services

Services which are directly translate into configuration concepts and build the first device independent abstraction layer:

- object oriented concept of building blocks
  - leaf-list:devices
- Services must not be able to “function” independently but can
- Examples: Port, prefix-lists, SAPs, SDPs, etc.
- Services expose basically “every” configuration knob
- Namespace: config-services
General NSO Architecture Revision Concept

Tier 1
- interface
- vlan
- Prefix-lists

Tier 2
- SNMP
- NTP
- netflow
- DNS
- syslog
- SSH
- static-routing
Tier 2 services

Service which build basic service abstraction concepts out the composition of tier 1 services:

- Composed out of tier-1 services and should build the first layer of functional services which lead to a working service configuration
- Could introduce additional device configuration to glue together tier 1 services and transition them to an independent functional service
- Tier-2 services can be stacked / composed out of other tier-2 services
- Services expose a comprehensive amount of knobs in order to tweak a service configuration
- Namespace: /ncs:base-services
General NSO Architecture Revision Concept

Tier 1
- interface
- vlan
- Prefix-lists
- SNMP
- NTP
- netflow
- DNS
- syslog
- SSH
- static-routing

Tier 2
- system
- IRB
- Bridge
- routing-domain
- LAG
- ACLs
- DHCP
- Tier 1
- Tier 2
Tier 3 services

Services which are composed out of a combination of tier 1 & 2 services and build an operational foundation for ESnet’s network operation and customer facing portfolio:

- High level service constructs composed out of tier 1 & 2 services
- These services should not have any configuration templating and work only by composing services
- Examples: BBL, L3, MPR & CR6 base config, host, etc.
- Services expose a “minimal” amount of knobs for a rapid service deployment
- Namespace: /ncs:services or /ncs:esnet-services
General NSO Architecture Revision Concept

Tier 3
- MPR base config
- CR6 base config
- BBL
- L3-service
- host

Tier 2
- system
- IRB
- Bridge
- routing-domain
- LAG
- ACLs
- DHCP

Tier 1
- interface
- vlan
- Prefix-lists
- SNMP
- NTP
- netflow
- static-routing
- DNS
- syslog
- SSH
Conclusion

- Starting from scratch has too many risks especially since our services are widely used and also intertwined between multiple software layers
- Move tier 3 services into Workflow Orchestrator whenever possible
- Make overall service design a multi layer approach
- Refactor existing services following the tier 2 & 1 service paradigm whenever possible
- Further build out more granular multi-tier service design and architecture
Offloading “umbrella” services into the Workflow Orchestrator.

Joe the Planner

Eric the Engineer

ESnet Network

Stardust
Future directions

- ESnet is further increasing the amount of orchestrated services
- Use only NSO to manipulate config
  - Reduce inconsistencies
- Transition towards an Workflow Orchestrator only model
  - In a perfect world there won’t be any need to touch devices or NSO directly
Questions...
Orchestration Architecture

- Network Intent
- Network Discovery
- Network Topology

Provisioning GUI
- Workflow Management
- Automated Provisioning

Service Workflow GUI

Ticket Management
- Change Control
- Other (internal or external systems)

Open Line System
Packet Network
High-Touch Edge
NFV/Compute
Supporting Equipment
Anatomy of an NSO service

Web UI  |  RESTCONF  |  CLI
---|---|---
Service Model (YANG)
Business Logic (python)
Configuration Templates (XML)
Nokia NED  |  Juniper NED  |  Arista NED
---|---|---
CLI  |  NETCONF  |  CLI

ESnet Confidential -- Do Not Distribute
Object oriented service architecture design

Composition: The tier-2 service creates the tier-1 service instance
Example:

Aggregation: The tier-2 service references the tier-1 service
Example: multiple IRB (on multiple devices) reference the DNS prefix-list and their ACLs
Tier-1 to Tier-2 service composition example based on the IRB service
UML of the Base-configuration services
## ESDDB GUI

### Circuits

<table>
<thead>
<tr>
<th>ESnet Circuit ID</th>
<th>Endpoint A/Z</th>
<th>Orchestrator ID</th>
<th>Provider / ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESNET-Z00227</td>
<td>forestal-mpr1:ge-0/0/40</td>
<td>DOE-NNSA-FORR:Site_conn</td>
<td>DOE-IN Headquarters Forrestal building</td>
</tr>
<tr>
<td>In-service</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESNET-Z00330</td>
<td>ATLA-Ciena-Shelf2:ETH10G-2-7-1</td>
<td>ATLM-Ciena-Shelf1:ETH10G-1-7-1</td>
<td>Internet2</td>
</tr>
<tr>
<td>In-service</td>
<td></td>
<td></td>
<td>ESNET-ATLA-ATLM-10GE-22437</td>
</tr>
<tr>
<td>ESNET-Z00349</td>
<td>ATLA-Ciena-Shelf2:ETH10G-2-7-4</td>
<td>ATLM-Ciena-Shelf1:ETH10G-1-7-4</td>
<td>Internet2</td>
</tr>
<tr>
<td>In-service</td>
<td></td>
<td></td>
<td>ESNET-ATLA-ATLM-10GE-22440</td>
</tr>
<tr>
<td>ESNET-Z00355</td>
<td>ATLA-Ciena-Shelf2:ETH10G-2-7-3</td>
<td>ATLM-Ciena-Shelf1:ETH10G-1-7-3</td>
<td>Internet2</td>
</tr>
<tr>
<td>In-service</td>
<td></td>
<td></td>
<td>ESNET-ATLA-ATLM-10GE-22439</td>
</tr>
</tbody>
</table>
Orch GUI