NeMo - Network Modeling for Applications

---- An Application API for Intent Driven Networking

Topics

- Why NeMo?
- Status
- State machine
- Demo Description

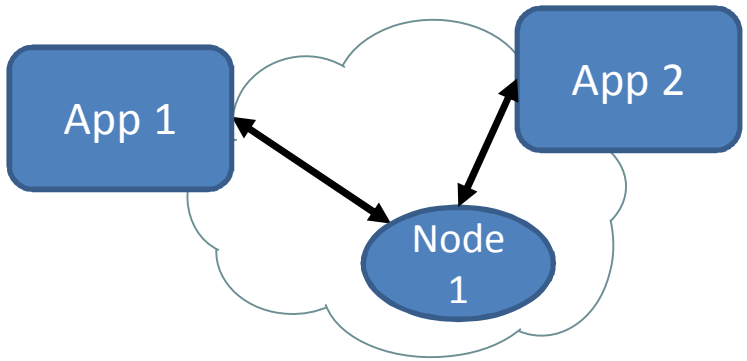
Why Intent-Driven NeMo?

Application needs Intent-Driven not prescriptive Control

- Application to state:
  - A connection between two sites with flows
  - A service flow with SLA
  - A customer network service chain
- Intent Driven: What I want not how to do it
  - Let network layers figure out how to accomplish intent
- High level
  - Yang is low-level specific to device

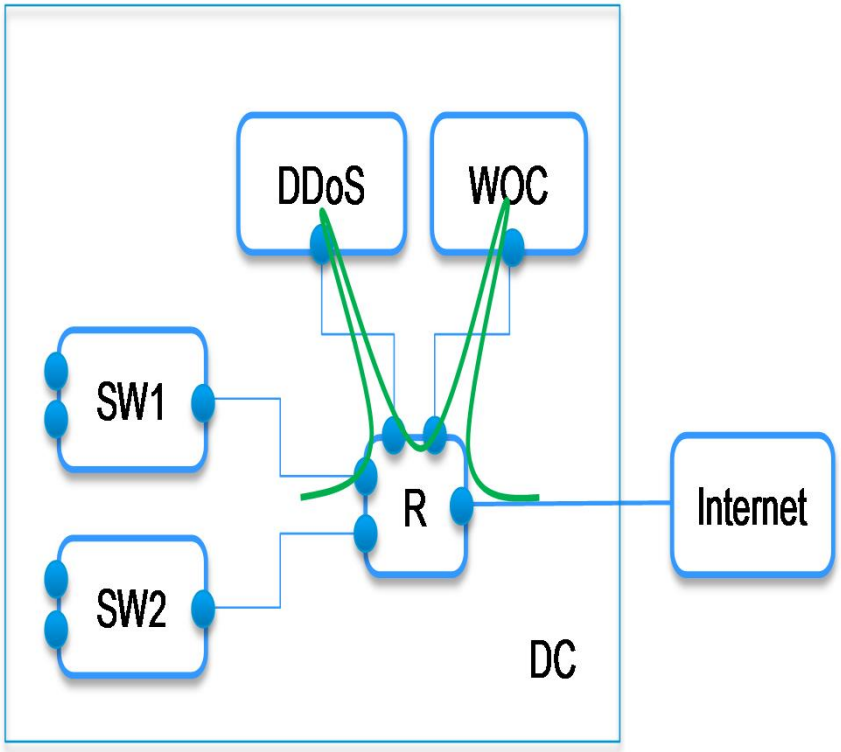
Applications need a Simple API

- Request virtual networks through specific nodes with network services at flow rate,
- When applications can aid control of network, storage, compute – can reach 95% utilization of net, storage, compute
- **NeMo** has 3 primitive groups, 15 sentences, and 36 key words

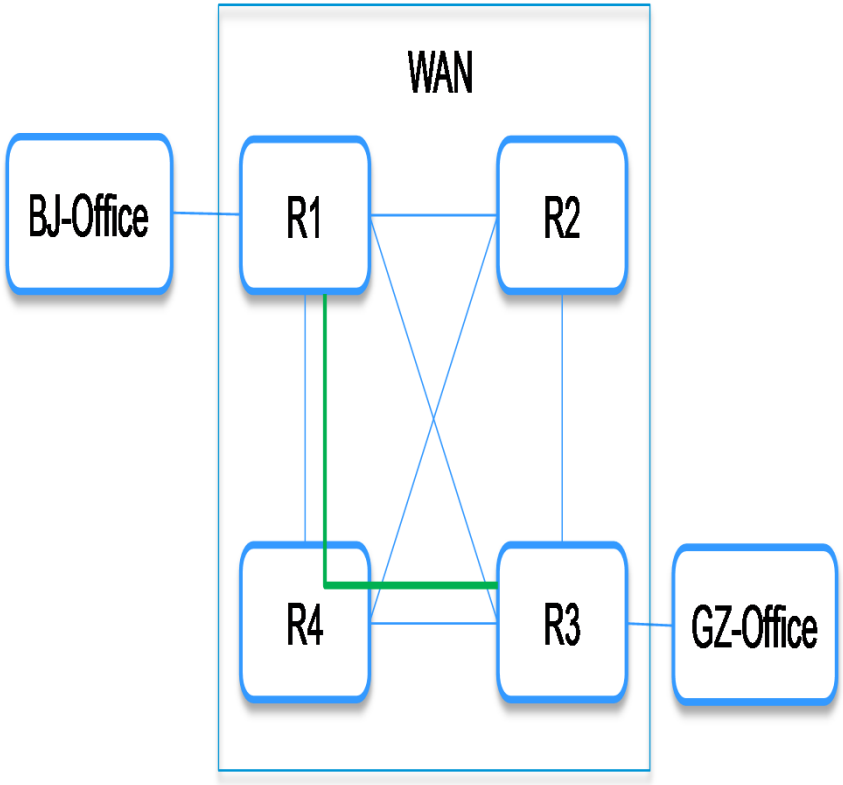


Use Cases Supported

Service Chaining



Virtual WAN with TE

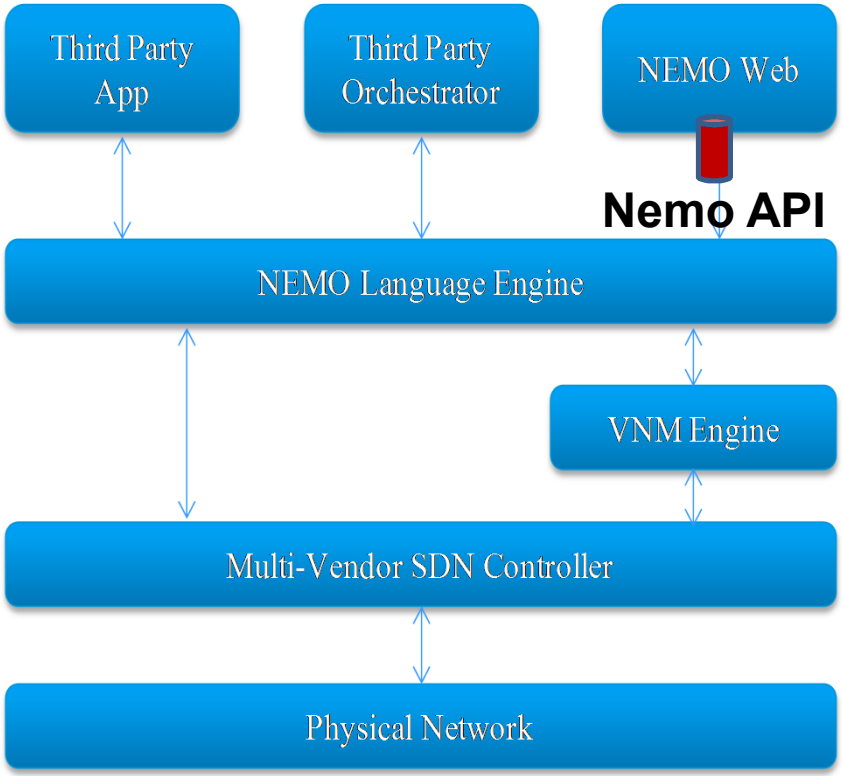


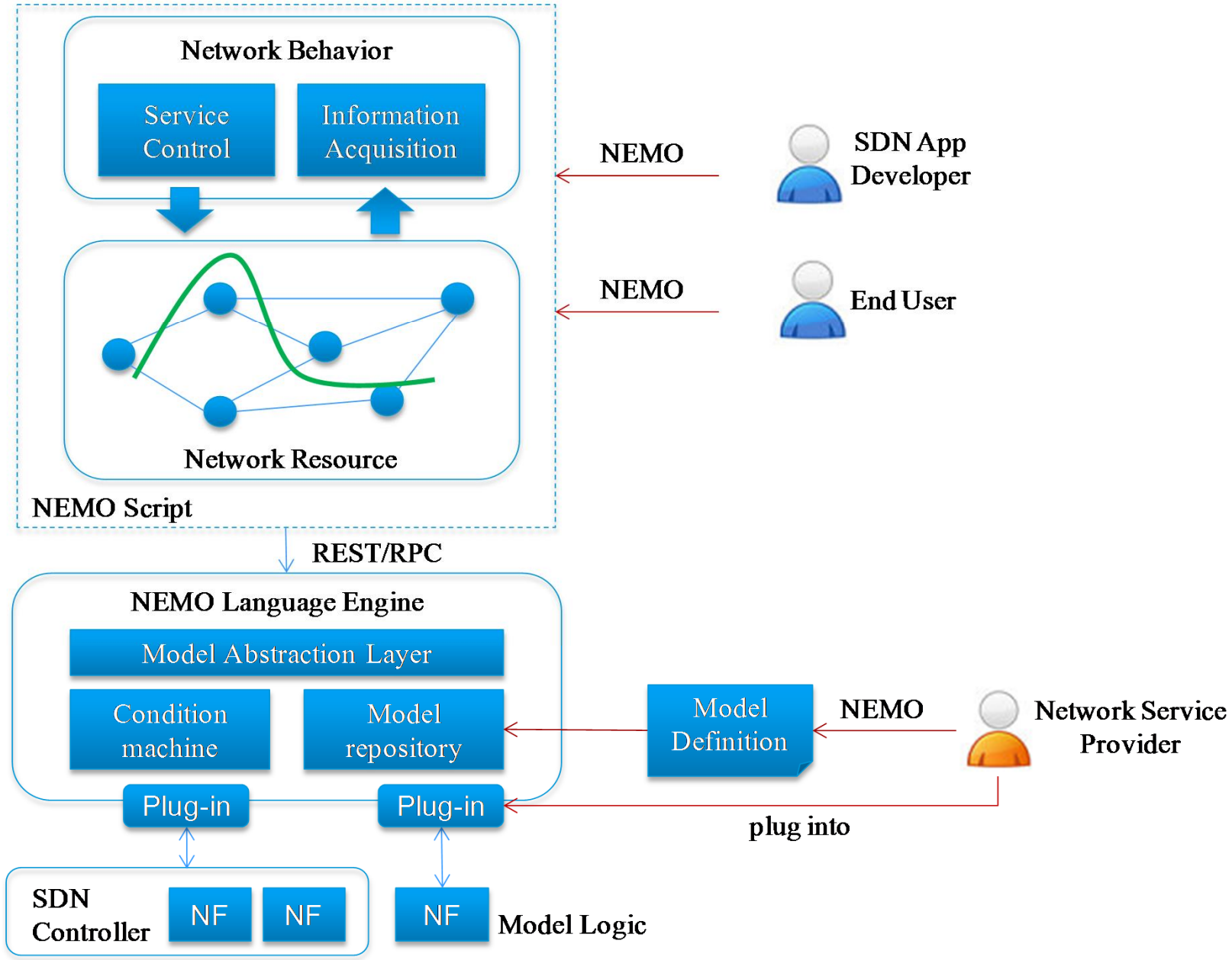
NeMo can enable Multi-service SDN Controller

Multi-Service SDNcontroller

- **Problem**
  - It's hard to support multiple, independently developed SDN applications or services without **resource conflicts**
- **State of the Art**
  - ODL Helium has not solved this problem which prevents competing flow writers that can't be run simultaneously.
  - It is not possible to run e.g. NetVirt and SFC services in the same controller domain.
  - Commercial controllers have not solved this problem either

NeMo's API uses REST/RPC to talk to Nemo Language Engine



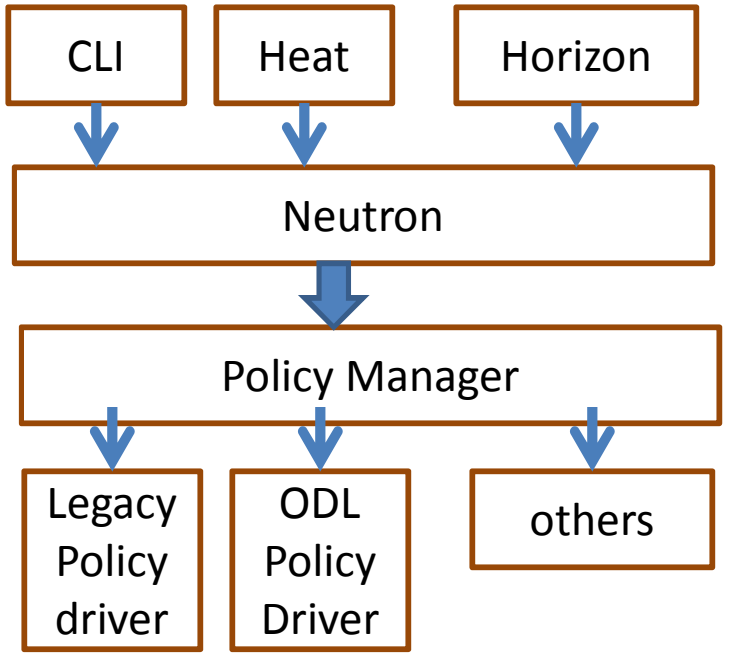

Network Behavior
Service Control
Information Acquisition
Network Resource
NEMO Script
NEMO
SDN App Developer
NEMO
End User
REST/RPC
NEMO Language Engine
Model Abstraction Layer
Condition machine
Model repository
Model Definition
NEMO
Network Service Provider
Plug-in
Plug-in
plug into
SDN Controller
NF
NF
NF
Model Logic

NeMo API at App layer rather than ODL Policy Groups

OPL Group Policy

- **Purpose:**
  - "higher" than neutron policy storage and control
- **Benefits:**
  - Intent based
  - Use PCIM concepts (RFC3060, 3460, 3644) that combine policy rules into policy groups (aka contracts)
- **Problem:**
  - Only Flow behavior, no create node or specify network service so cannot handle NFV devices or TE channels
  - Need Network flows, NFV, SFC, TE **plus** compute and storage placement

Policy Groups architecture



Status

Completed: (July – Nov)

- API presented at network forums
- IETF drafts + technical Manual specify language State Machine +
- Proof of Concept demo created

Possible Next Steps:

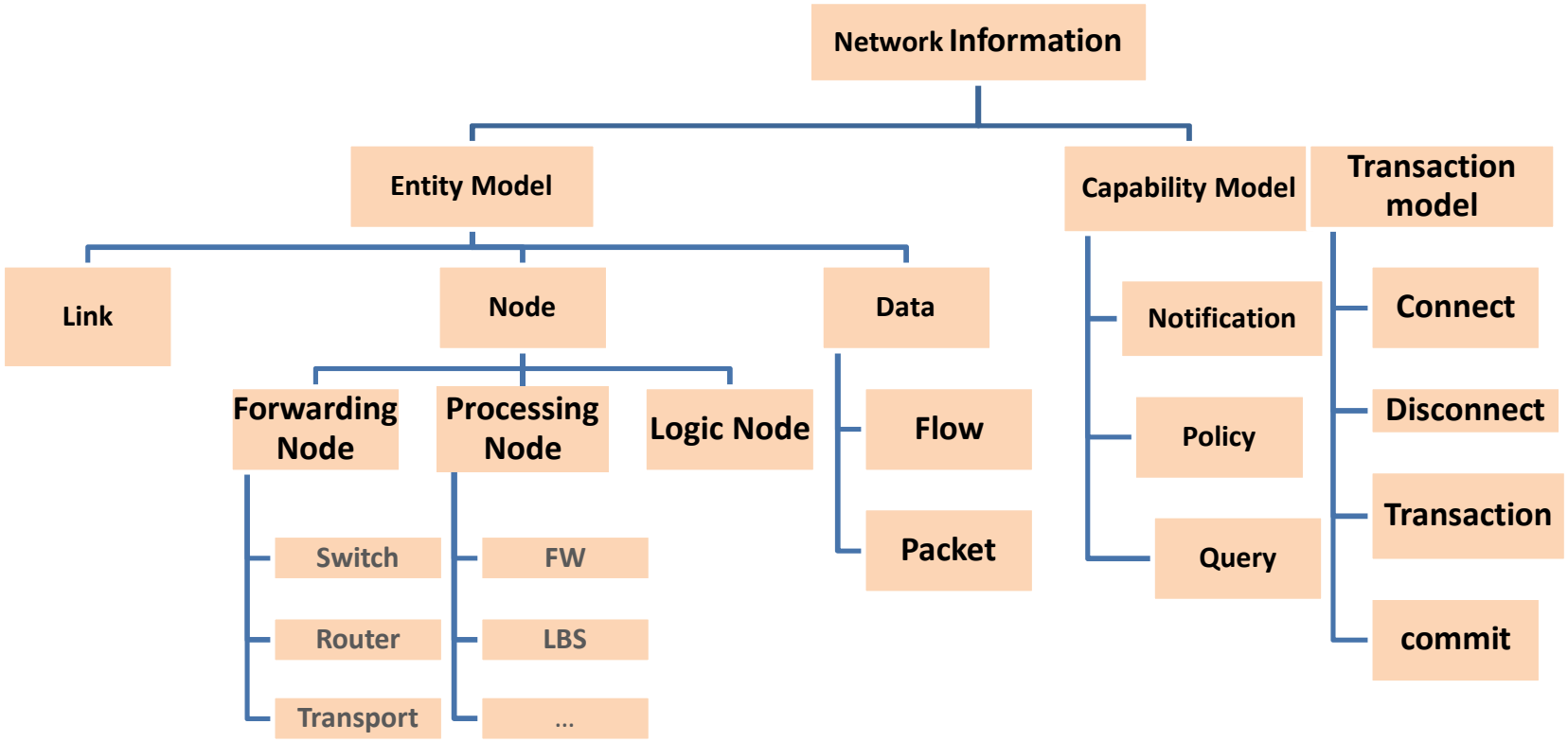
- Work with Partners on API
- Open Daylight project to integrate NB API + Nemo Engine running over
  - Open Flow with SFC and SFC chaining,
  - I2RS yang modules ,

We welcome feedback on NEMO, proof of concept demo, and our next steps.

NeMo State Machine

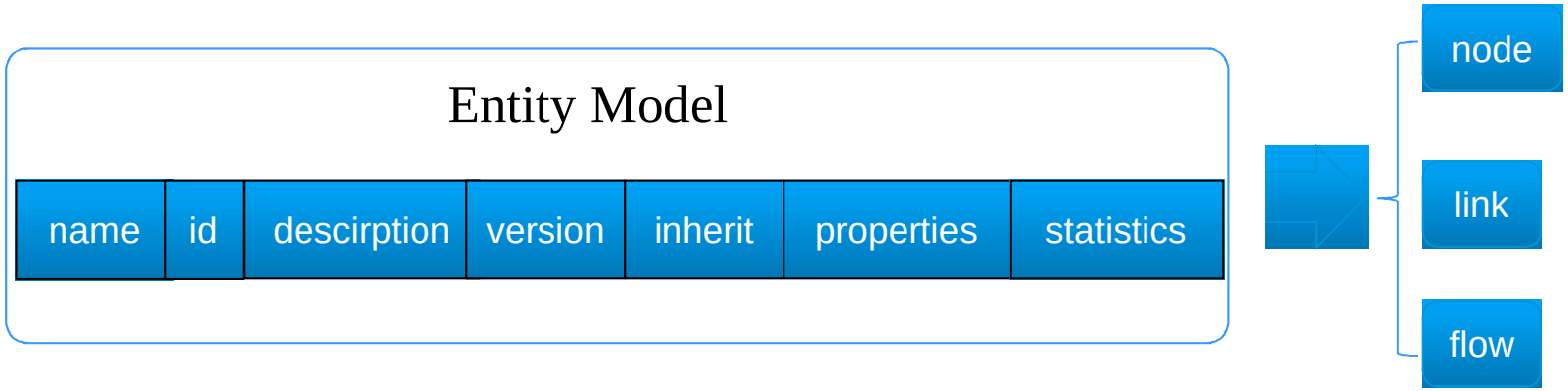
- Top down view
- Entity Model
- Capability Model
- Language primitives: 3 groups, 15 sentences, 36 key words)

Top-Down design
Network Abstraction Model



Entity Model

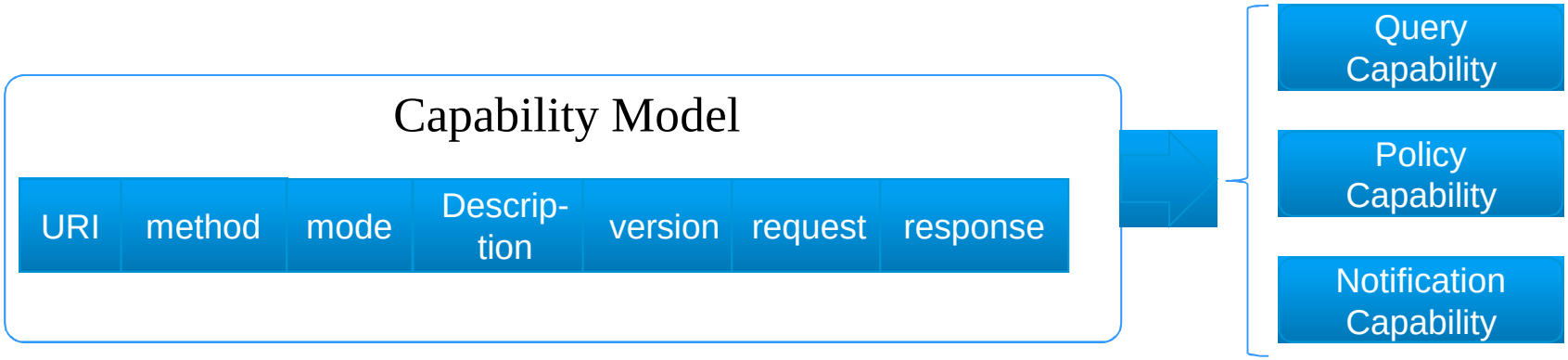
- The entity model provides a fundamental abstraction for both basic network objects (such as basic network element, link, and flow) and extended objects (such as firewall, load-balancer, and DPI).



Entities derived from the entity model

Capability Model

- Capability model describes a set of network functions and operations that is opened to the user
- Two operation modes are defined in the capability model:
  - Synchronous mode: e.g. a creation of virtual network.
  - Asynchronous mode: e.g. port failure notification.



Capabilities derived from the capabilities model

NEMO Language: Concise and Flexible

Resource Access

Entity Model	**node**	Node/UnNode entity_id Type {FN|PN|LN} Owner node_id Properties key1 ,value1
	link	Link/UnLink entity_id Endnodes (node1_id,node2_id) SLA key,value Properties key1 ,value1 ….
	flow	Flow/UnFlow entity_id Match/UnMatch key1, value1|Range(value, value) |Mask(value, value) Properties key1 ,value1

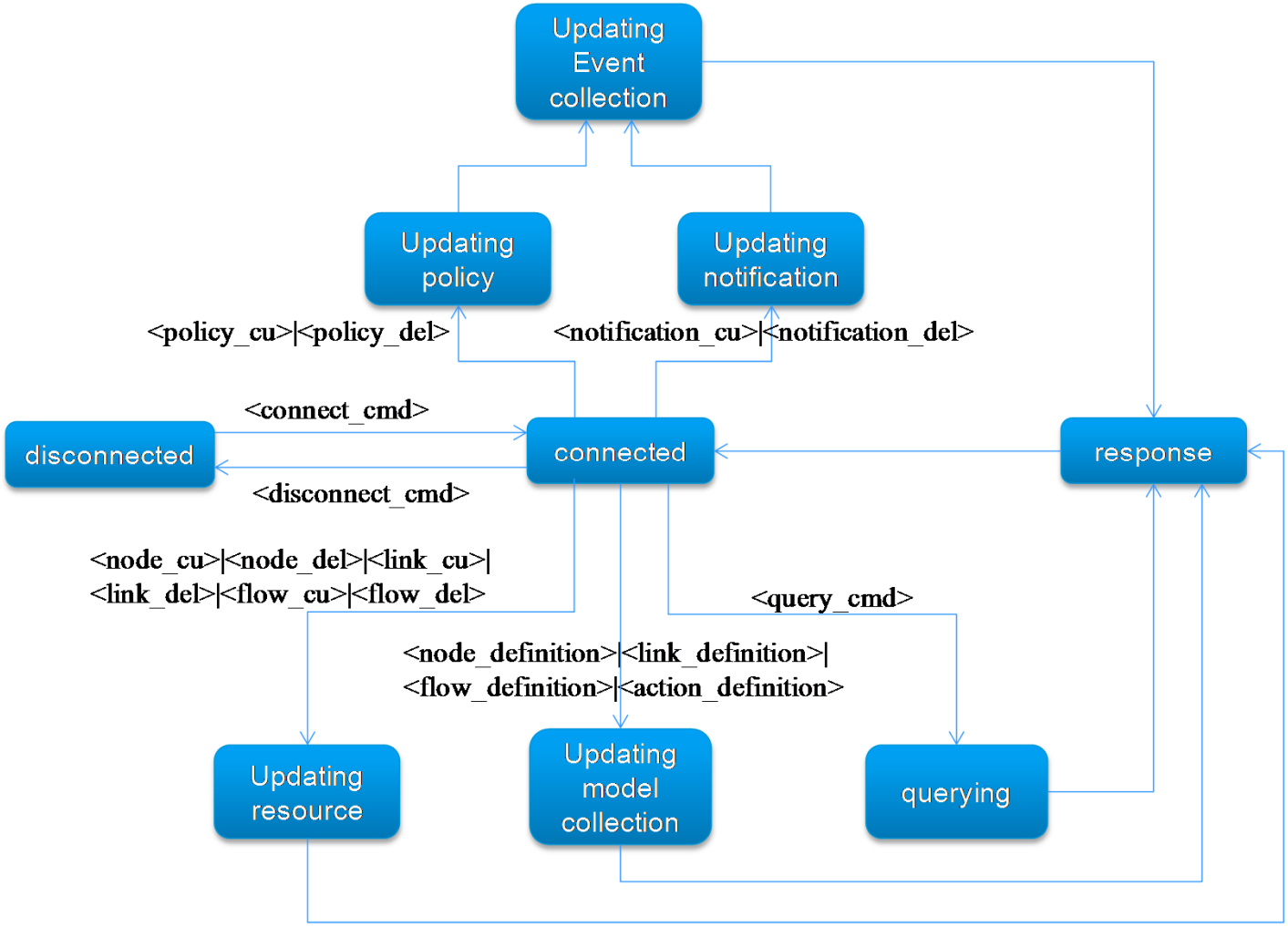
Policy and Event Handling

Capability Model	**Query**	Query key Value {value} From entity_id
	Policy	Policy/UnPolicy policy_id Appliesto entity_id Condition {expression} Action { "forwardto" | "drop" | "gothrough" | "bypass" | "guaranteeSLA" | "Set" |"Packetout"| Node | UnNode | Link | Unlink} Commit / Withdraw
	Notifica-tion	Notification entity_id On key Every period RegisterListener callbackfunc

Model Definition and Transactions Control

Connect	**Connect <conn-id> Address <ip-prefix> Port <integer>**
	Disconnect <conn_id>
Transaction	**Transaction …. Commit**
Node definition	**NodeModel <node_type> Property { <data_type> : <property_name> }**
Link definition	**LinkModel <Link_type> Property { <data_type> : <property_name> }**
Action definition	**ActionModel <Action_Name> parameter { <data_type> : <property_name> }**

NeMo Language Engine



Demos and Documents

Demos – After SDNRG and NFVRG

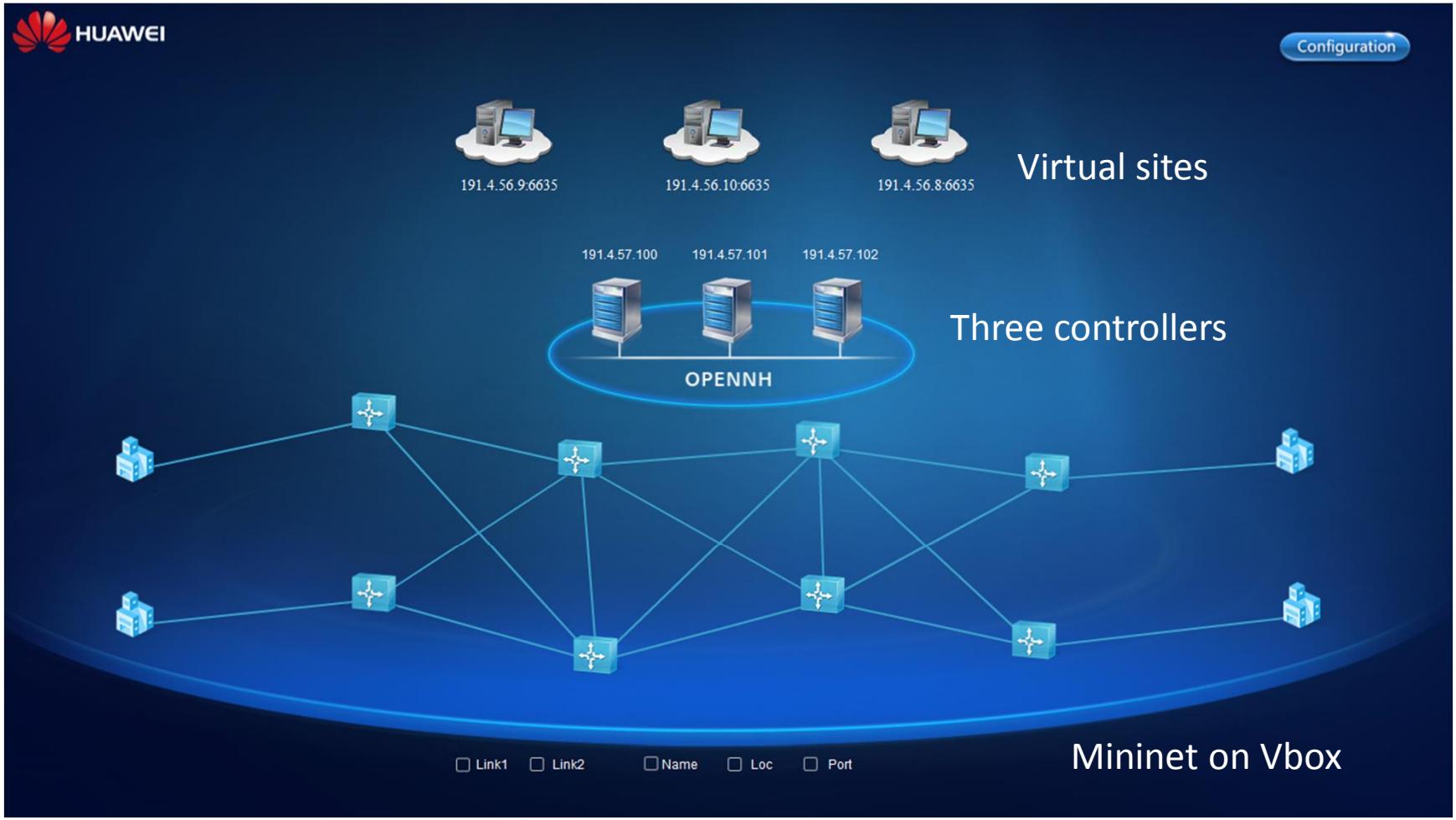
IETF Drafts:

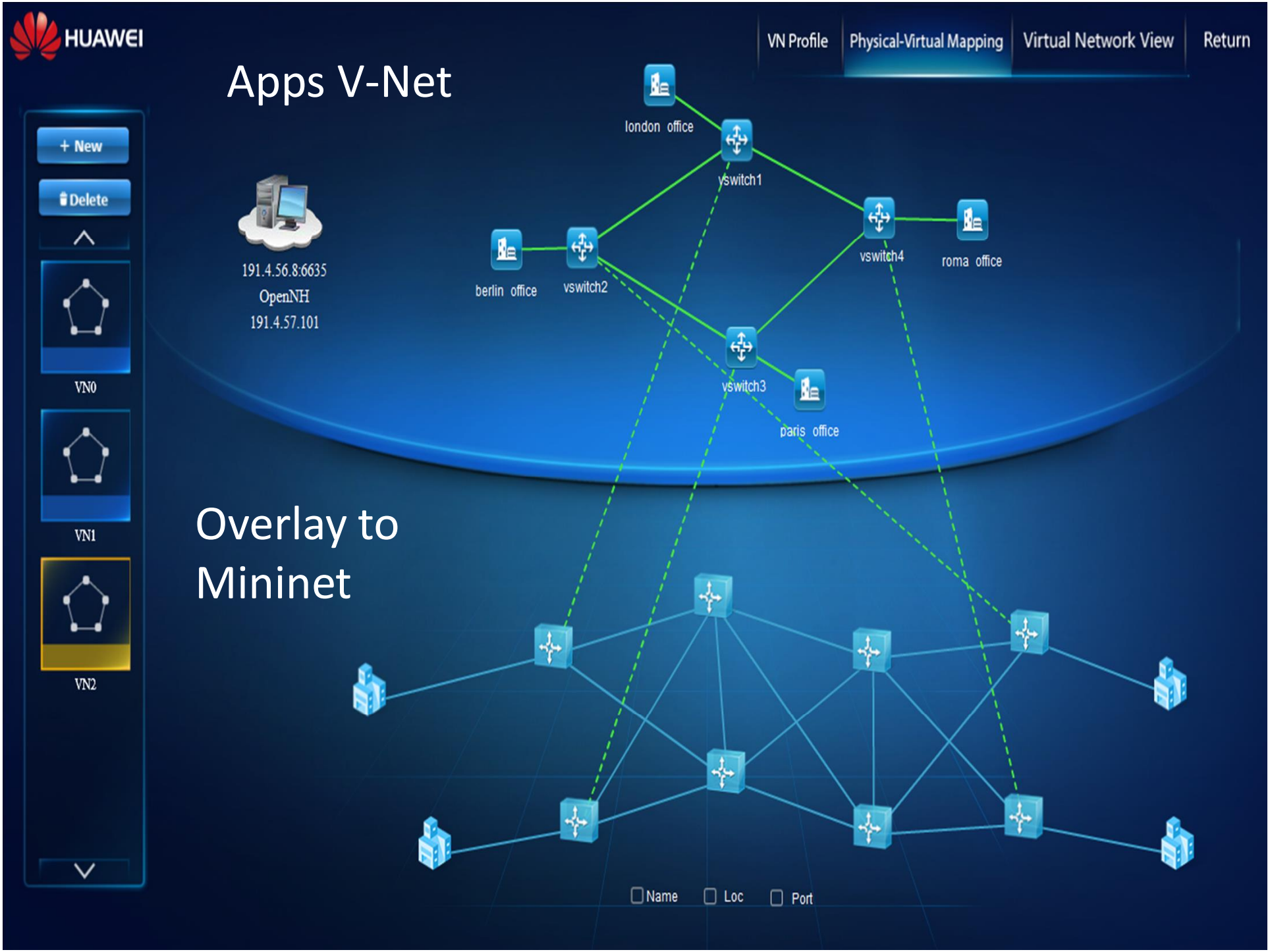
- draft-xia-sdnrg-service-description-language-01
- draft-xia-sdnrg-nemo-language-01

All Project documentation

- Technical Reference
- 5 page summary
- Status of Code
- Presentations

Demo


HUAWEI
Configuration
191.4.56.9:6635
191.4.56.10:6635
191.4.56.8:6635
Virtual sites
191.4.57.100
191.4.57.101
191.4.57.102
Three controllers
OPENNH
Link1
Link2
Name
Loc
Port
Mininet on Vbox


HUAWEI
VN Profile
Physical-Virtual Mapping
Virtual Network View
Return
Apps V-Net
+ New
Delete
VN0
VN1
VN2
191.4.56.8:6635
OpenNH
191.4.57.101
london office
vswitch1
berlin office
vswitch2
vswitch4
roma office
vswitch3
paris office
Overlay to
Mininet
Name
Loc
Port

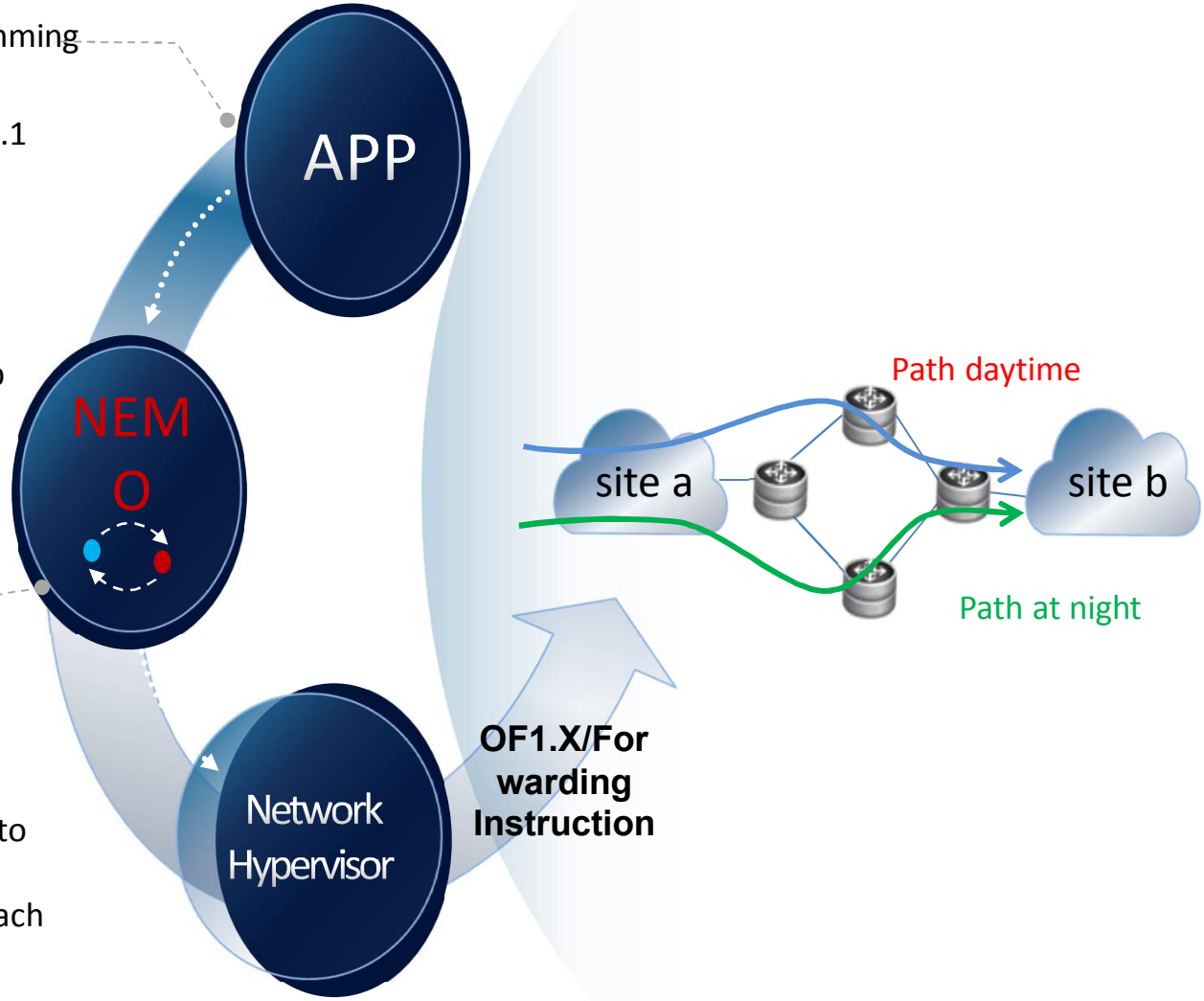
Example of Service Programming by NEMO

App use NEMO language to programming their service:

Flow sitea2siteb **Match srcip**:10.0.0.1 **dstip:**10.0.1.1;

Policy day **applyto flow** sitea2siteb **condition** 0800<**time**<2000 **action gothrough** {R1,R2,R4};

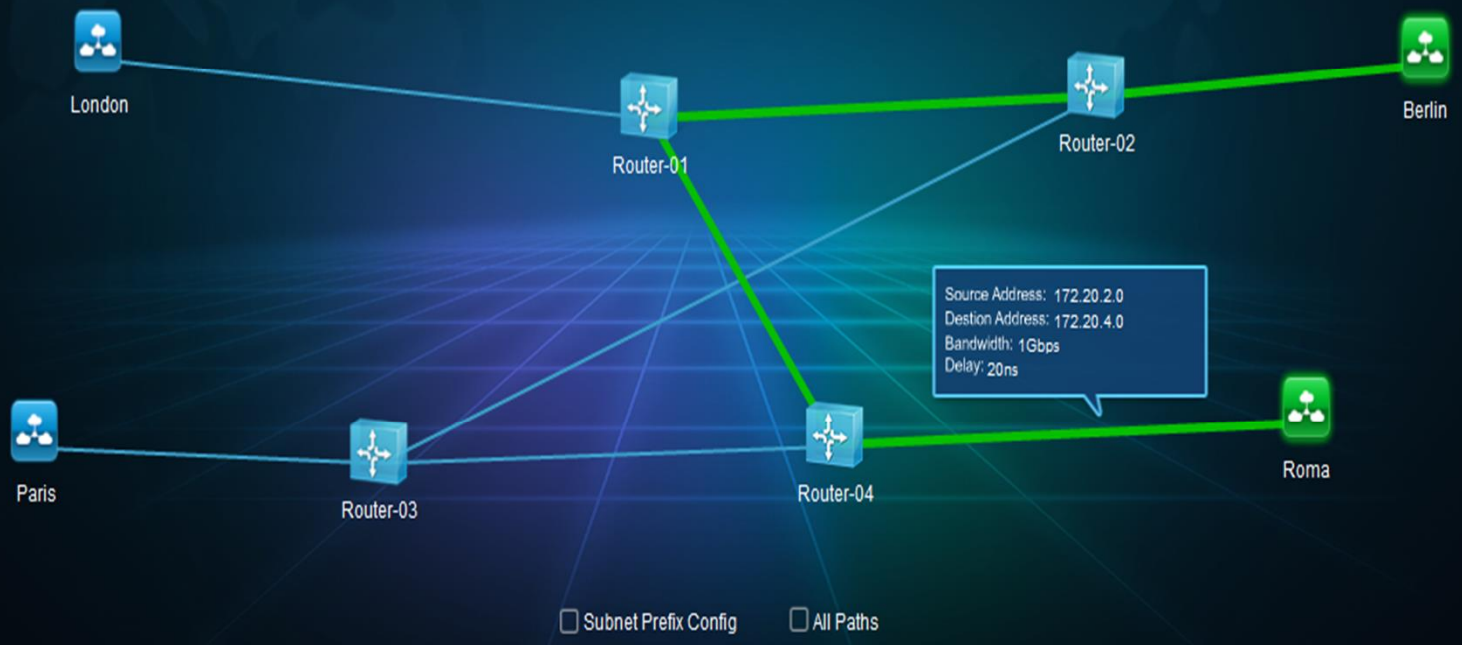
Policy night **applyto flow** sitea2siteb **condition** 2000<time<0800 **action gothrough** {R1,R3,R4};



Compiler resolver NEMO code to southbound instruction and maintain a state machine for each app.

At daytime go through path1;

Flows in Apps Virtual Network



Q & A