Toward a Distributed SDN Controller - ONOS

Jian Li

ONOS/CORD Ambassador Steering Team, Open Networking Foundation (ONF), US

ONOS/CORD Working Group, SDN/NFV Forum, Korea

PIRL, POSTECH, Korea

jian@opennetworking.org

APNOMS 2017

Agenda

- Background
- Introduction to ONOS
  - Distributed Core
  - Northbound
    - Dynamic Configuration
  - Southbound
  - Application
- ONOS Community
- DEMO

SDN Evolution and ONF

ON.LAB

- Non-profit, carrier and vendor neutral
- Provide technical shepherding, core team
- Build community
- Many organizations supports

Invention

2007 – Creation of SDN Concept

Platform Development

2007 – Ethane
2008 – OpenFlow
2009 – FlowVisor, Mininet, NOX
2010 – Beacon

Deployments

2009 – Stanford
2010 – GENI started and grew to 20 universities
2013 – 20 more camp uses to be added

Demonstrations

2008-2011 – SIGCOMM
2011 – Open Networking Summit, Interop

2012 –Define SDN research agenda for the coming years

2017 – ON.Lab and ONF merger

And Beyond

Why are Service Providers Interested in SDN?

Reduce CAPEX and OPEX

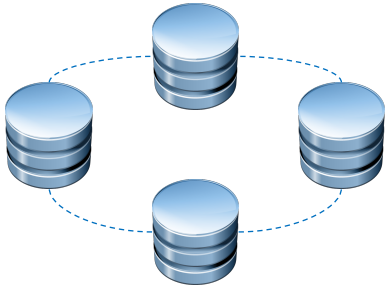
Bring cloud-style agility, flexibility, scalability to their networks

Roll out service rapidly

Reduce operational complexity, increase visibility

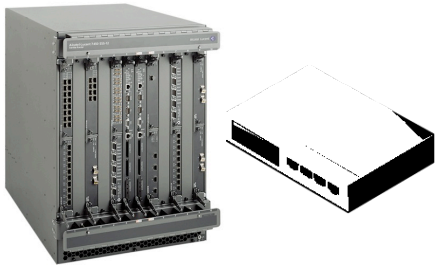
But Service Provider networks place stringent requirements on SDN control plane

Handle tens of millions of fixed and hundreds of millions wireless end points



Provide five nines availability, high performance, low latency

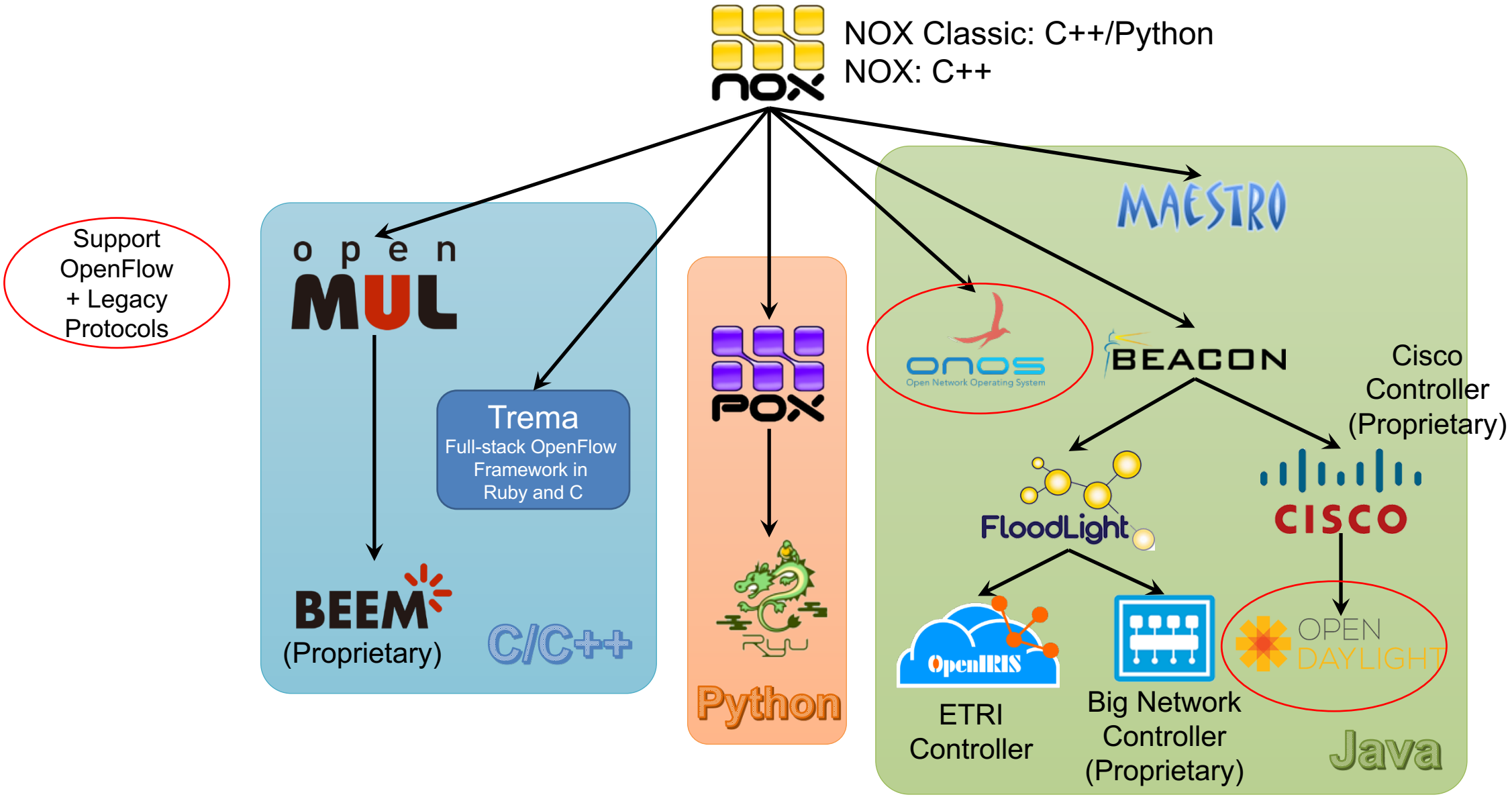
Need ease of use, service creation and delivery



Allow seamless migration of existing N/W while capitalizing on white boxes

ONOS is a SDN network operating system (control plane platform) designed for these stringent Service Provider requirements

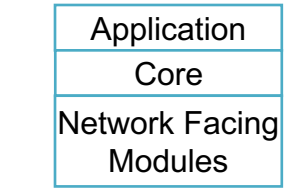
Pedigree Chart of OpenFlow Controllers

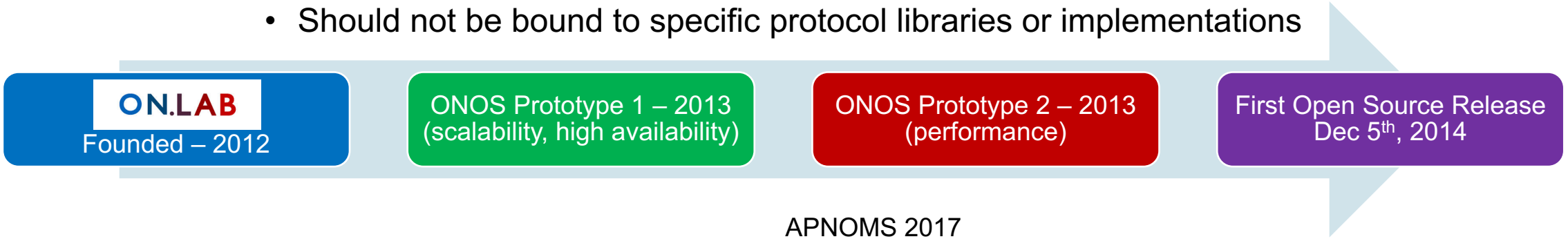


Introduction to ONOS

ONOS Overview

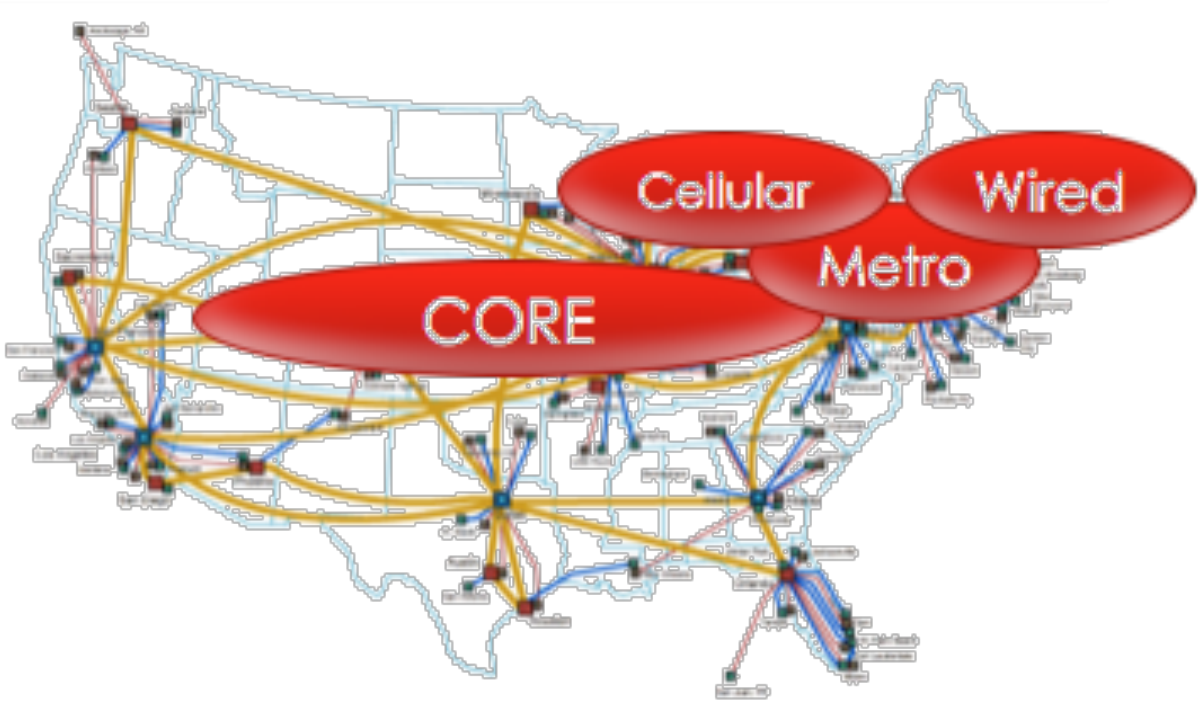
- ONOS: Open Network Operating System
  - Open source SDN network operating system
  - Objective: enable service providers to build real SDN/NFV solutions
  - Design goals
    - Code modularity: possible to introduce new functions as self-contained units
    - Configurability: possible to load and unload various features in runtime
    - Separation of Concern
      - **Protocol-aware network-facing modules** → interact with network
      - **Protocol-agnostic system core** → tracks and serves info on network state
      - **Application** → consumes and acts on the information provided by core
    - Protocol agnosticism
      - Should not be bound to specific protocol libraries or implementations





Service Provider Networks

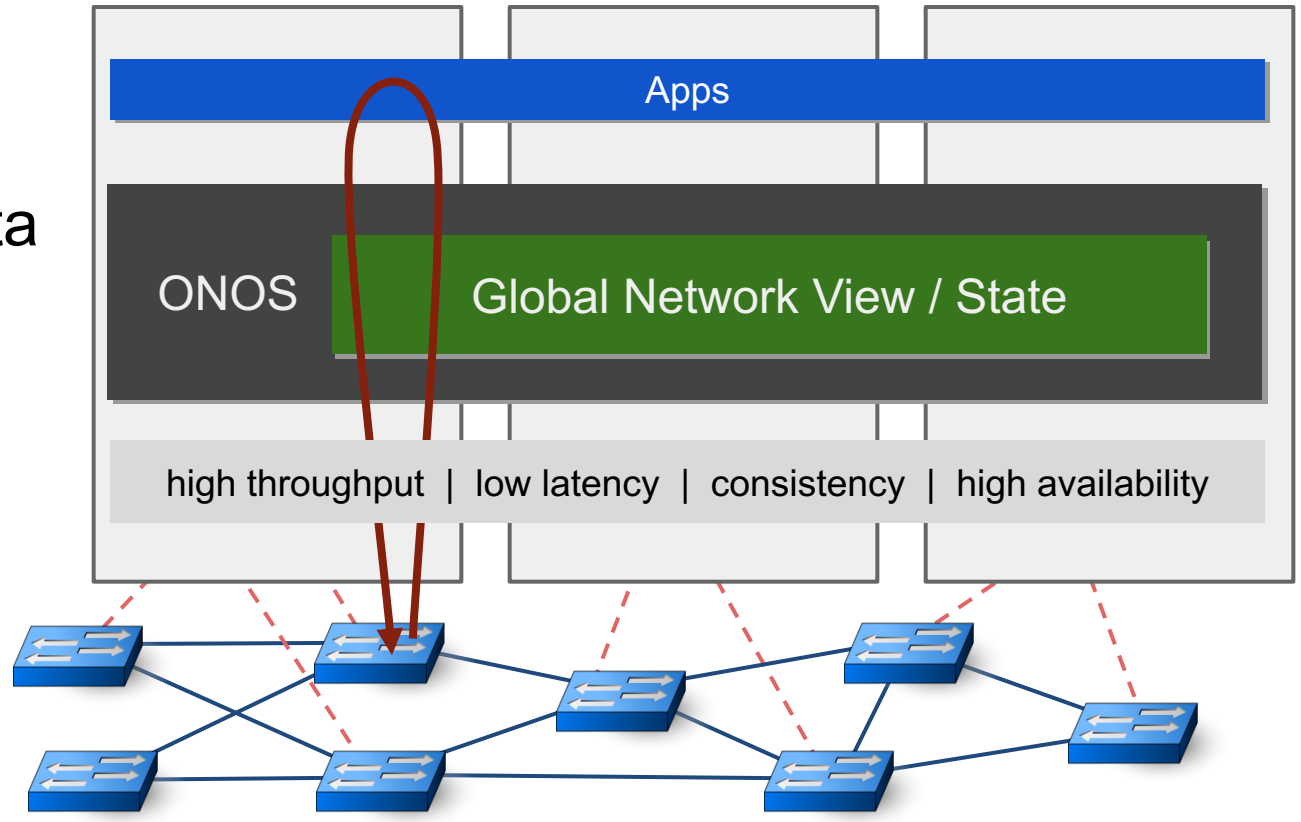
- WAN core backbone
  - Multi-protocol Label Switching (MPLS) with Traffic Engineering (TE)
  - 200-500 routers, 5-10K ports
- Metro Networks
  - Metro cores for access networks
  - 10-50K routers, 2-3M ports
- Cellular Access Networks
  - LTE for a metro area
  - 20-100K devices, 100K-100M ports
- Wired Access / Aggregation
  - Access network for homes
  - 10-50K devices, 100K-1M ports



Key Performance Requirements

- High Throughput
  - 500K - 1M paths setups/s
  - 3 - 6M network state operations/s
- High Volume
  - 500GB - 1TB of network state data

Challenging!



ONOS Architecture (1/2)

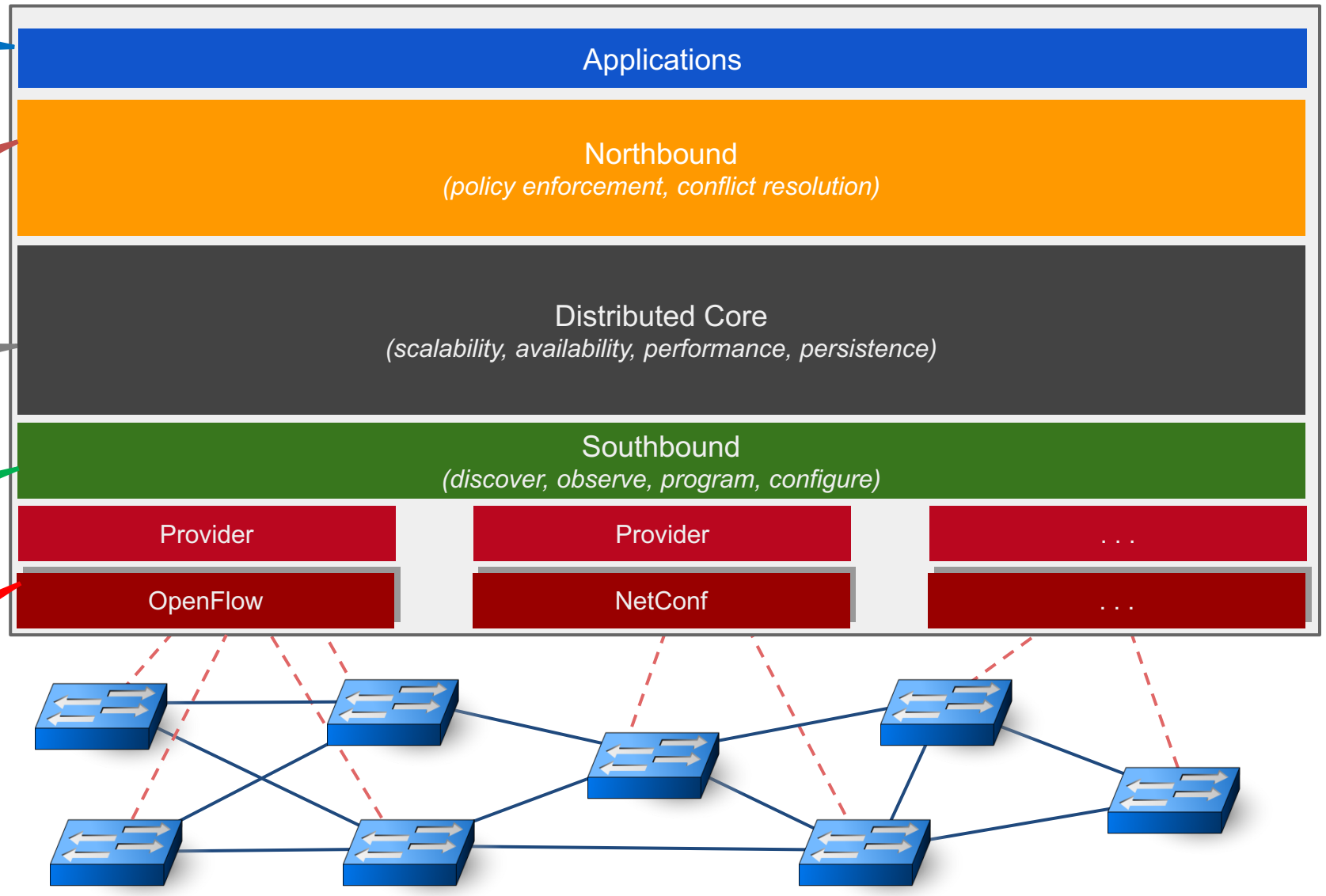
Contains user applications
E.g., reactive forwarding, ProxyARP, segment routing, SDN-IP, etc.

Transfer network info to app layer
Provide management interface for controlling lower layer component

Contains many core features
Provide distributed clustering func. for supporting HA and scalability

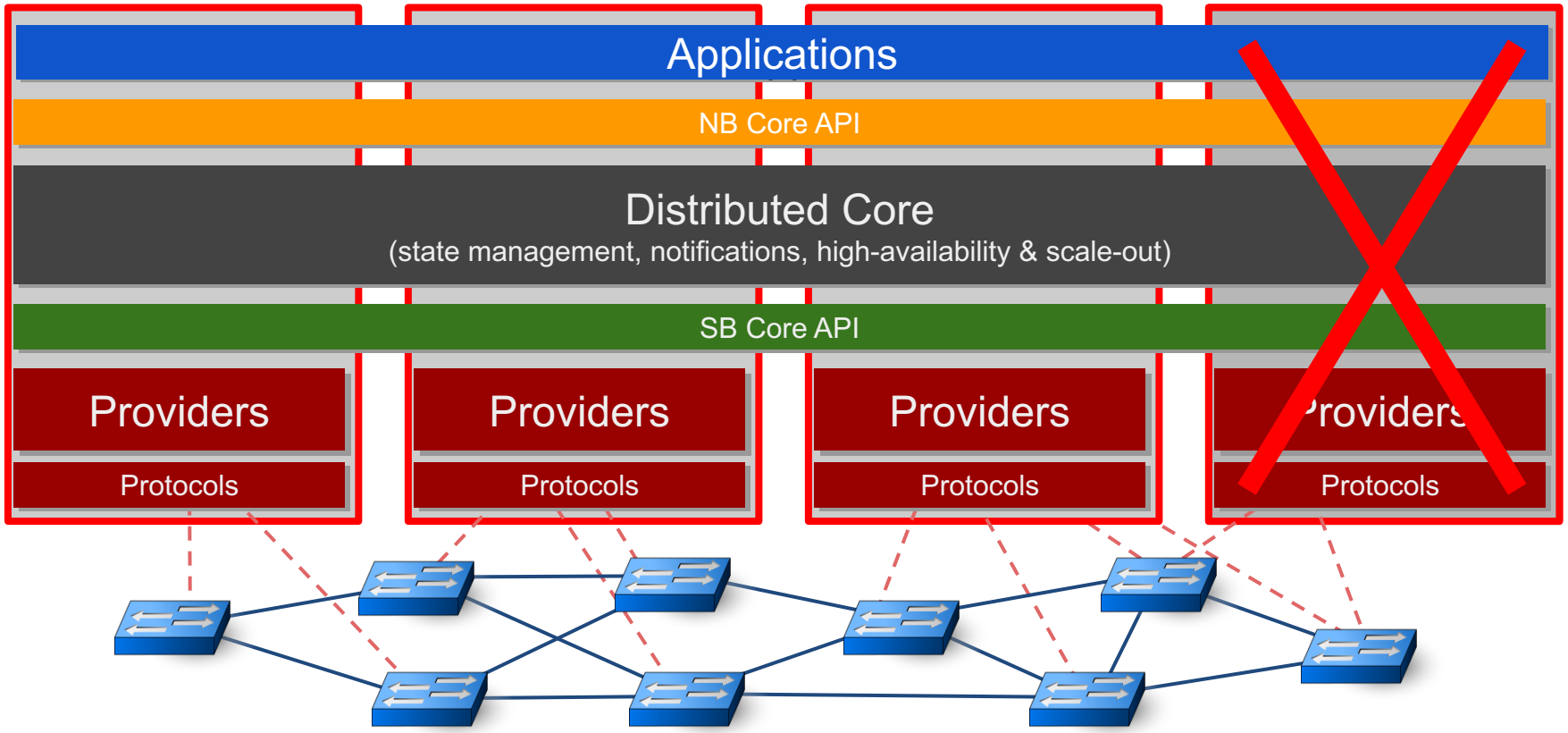
Provide an abstracted interface for controlling the network infrastructure

Network protocol implementation for managing network elements
E.g., OpenFlow, NetConf

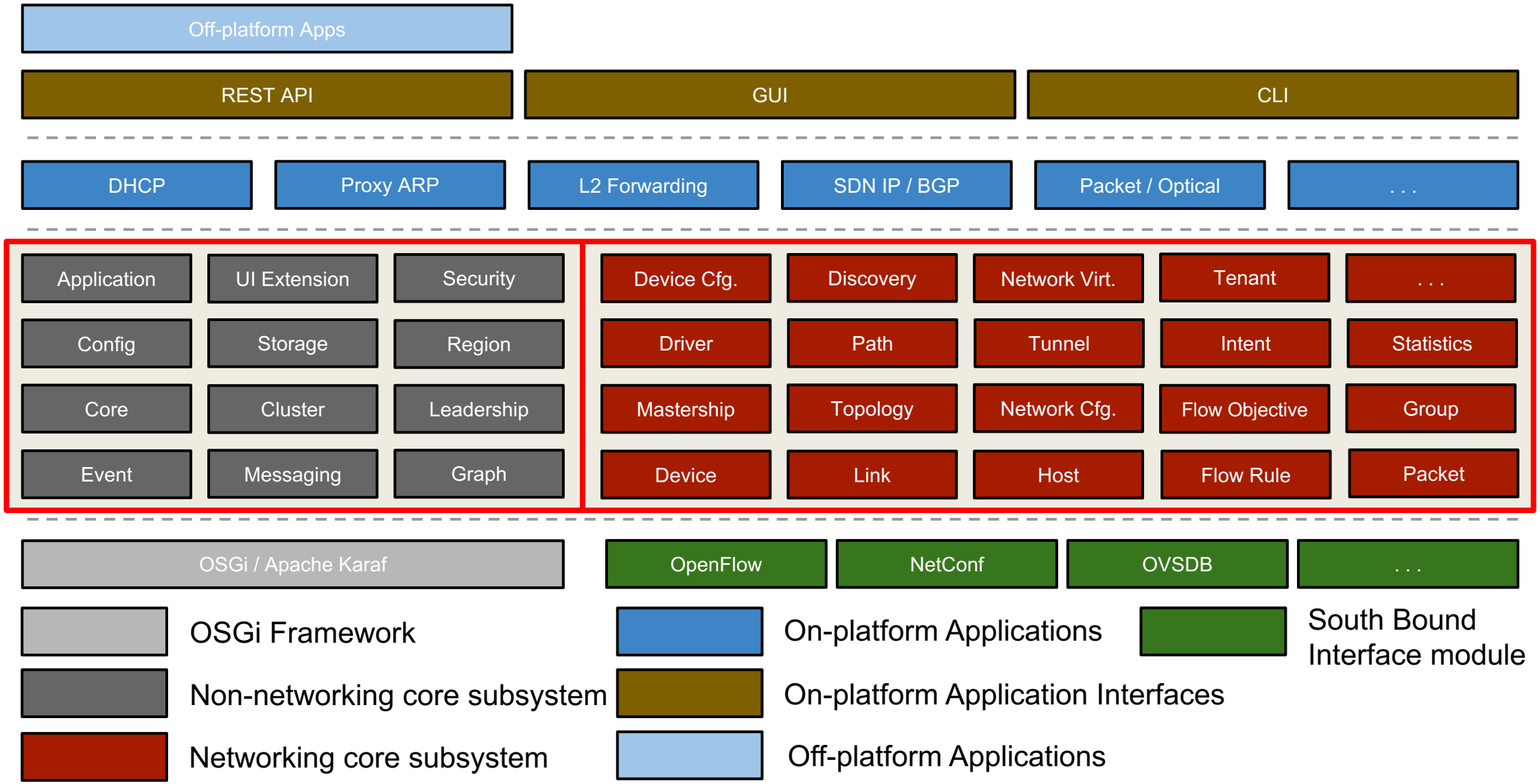


ONOS Architecture (2/2)

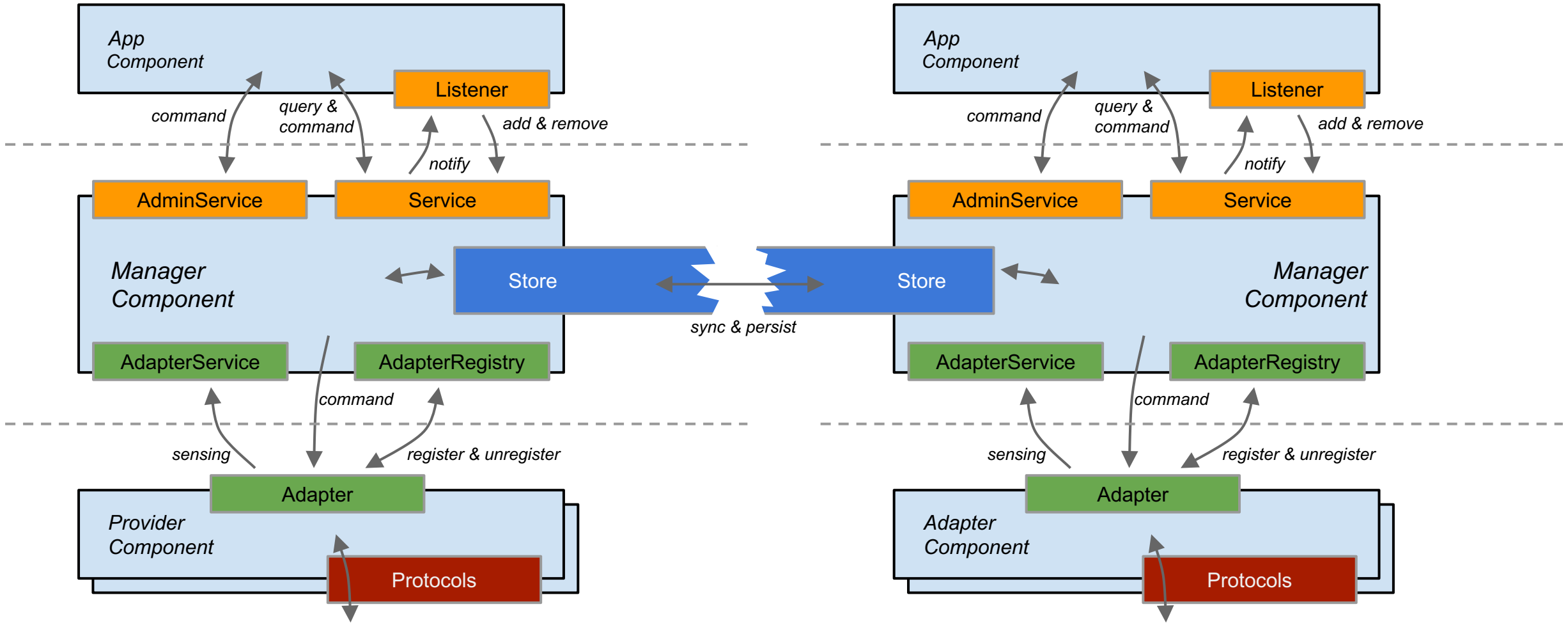
- Features
  - High Availability (HA)
  - Load Balancing (LB)



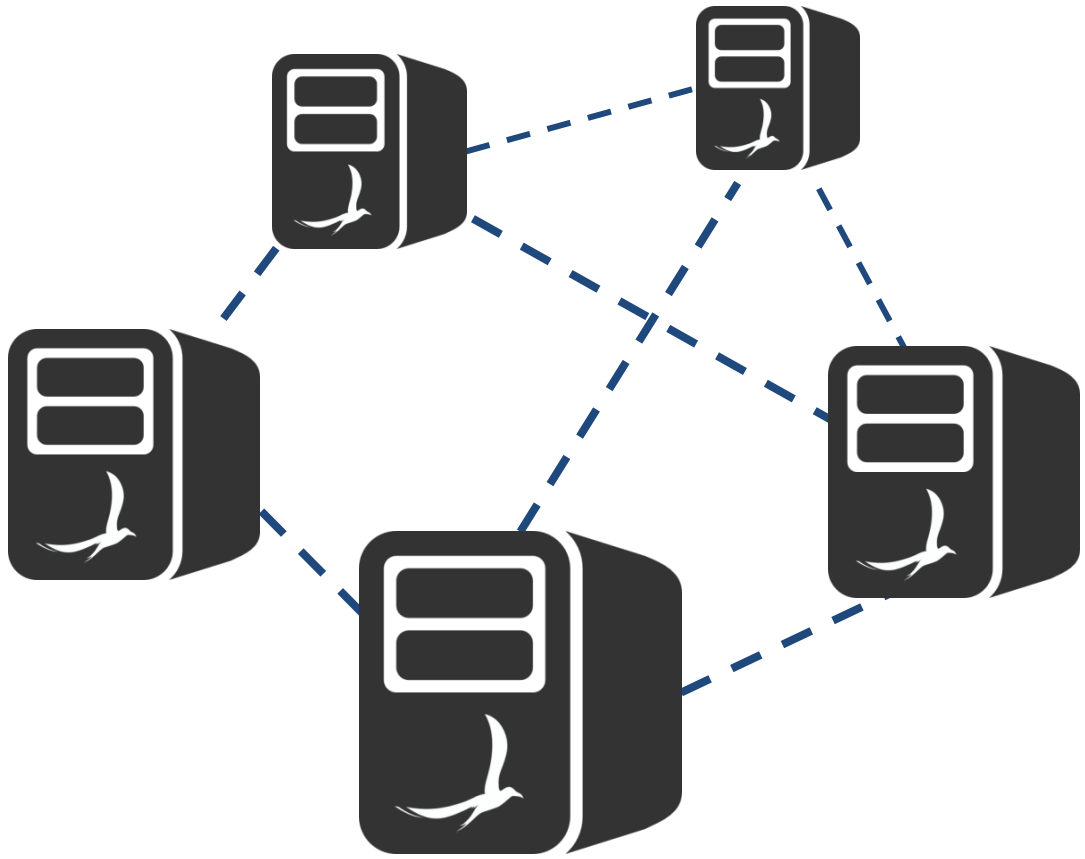
ONOS Subsystems (Services)



ONOS Core Subsystem Structure

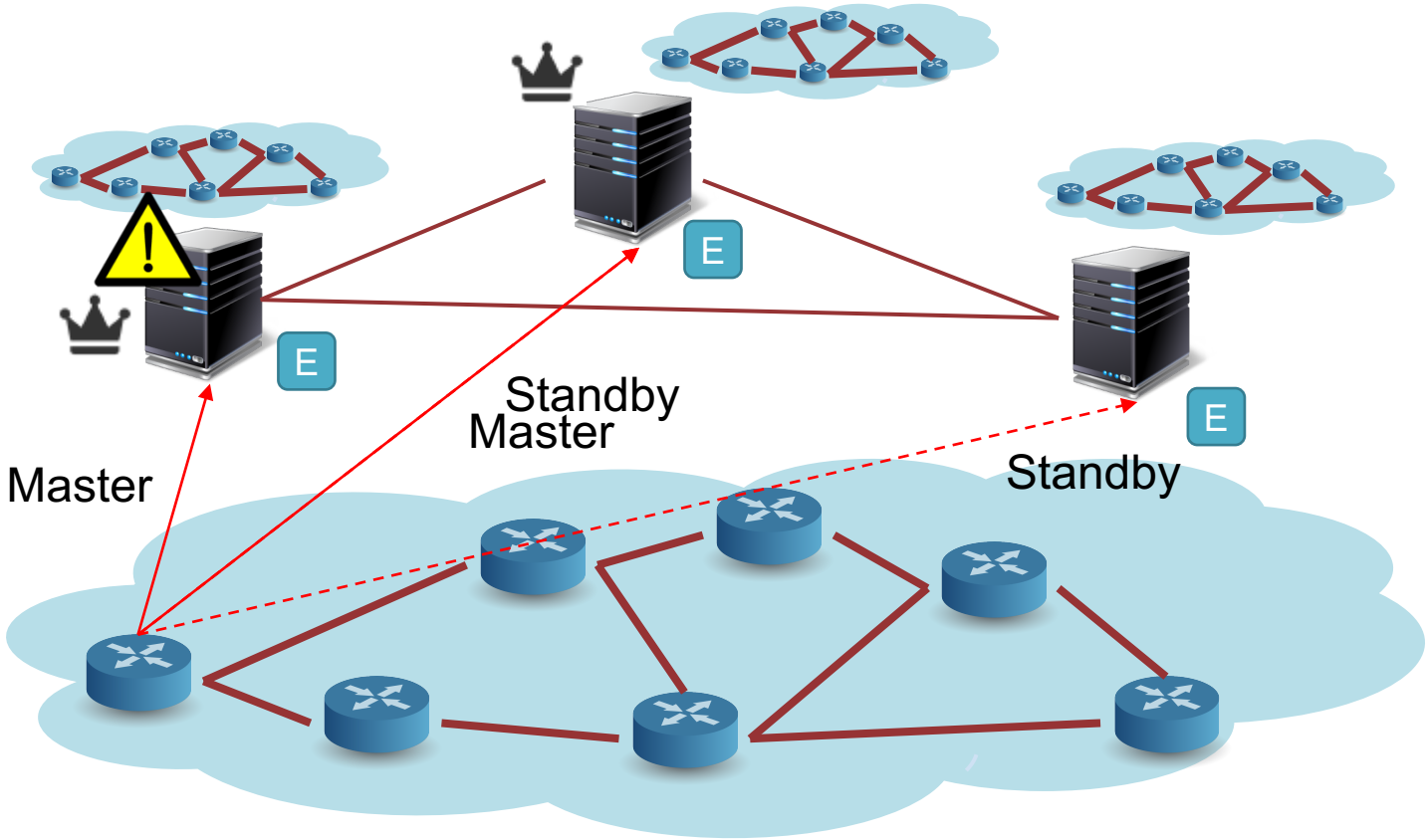


Distributed Core

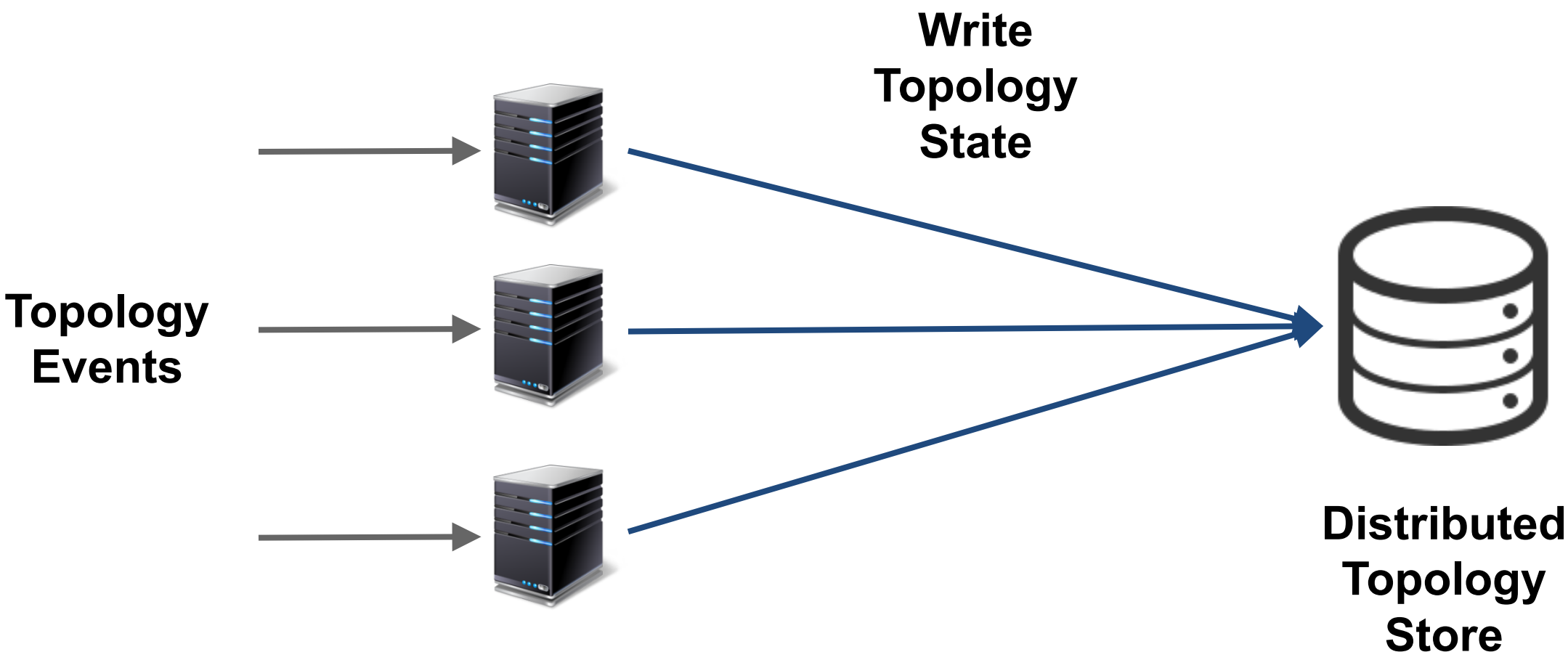


Distributed Architecture (1/5)

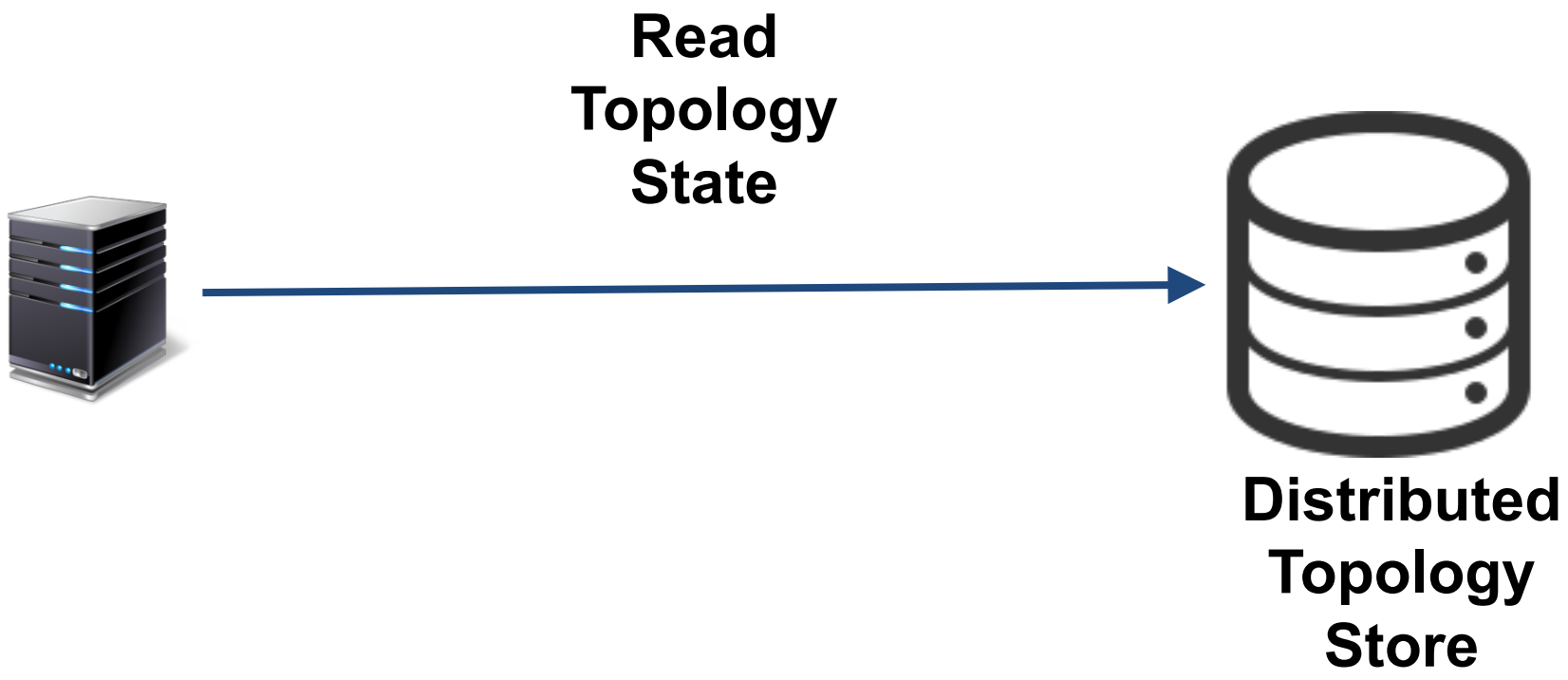
- Distributed
  - Setup as a cluster of instances
- Symmetric
  - Each instance runs identical software and configuration
- Fault-tolerant
  - Cluster remains operational in the face of node failures



Distributed Architecture (2/8)

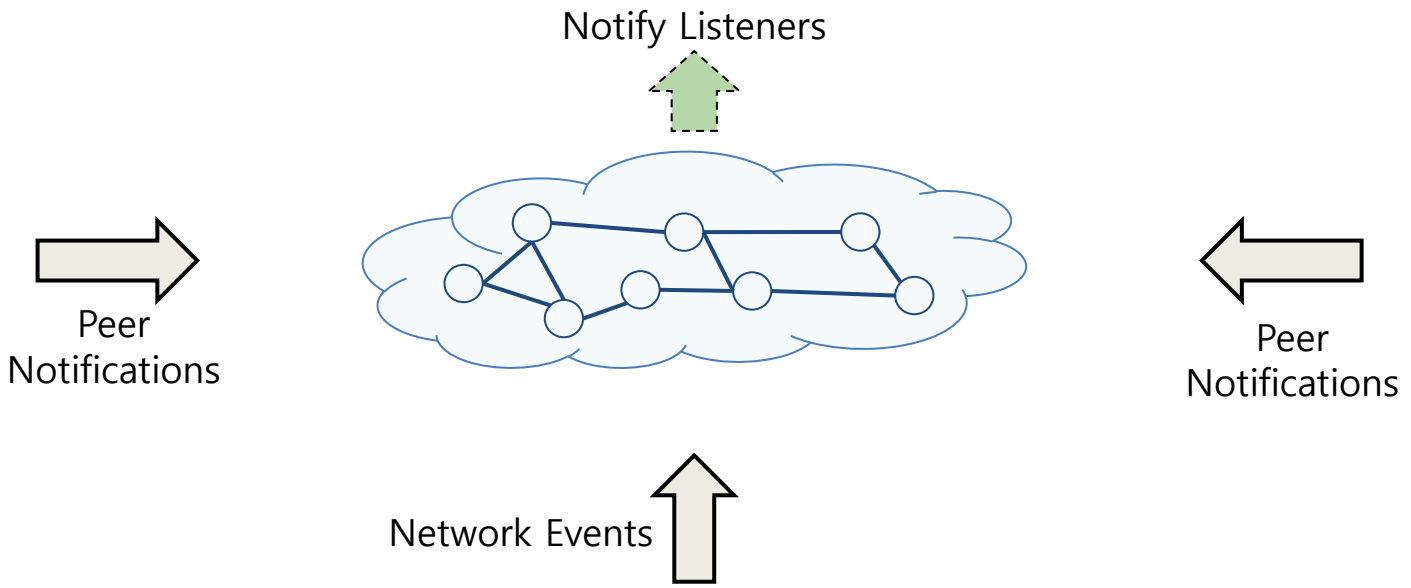


Distributed Architecture (3/5)



Distributed Architecture (4/5)

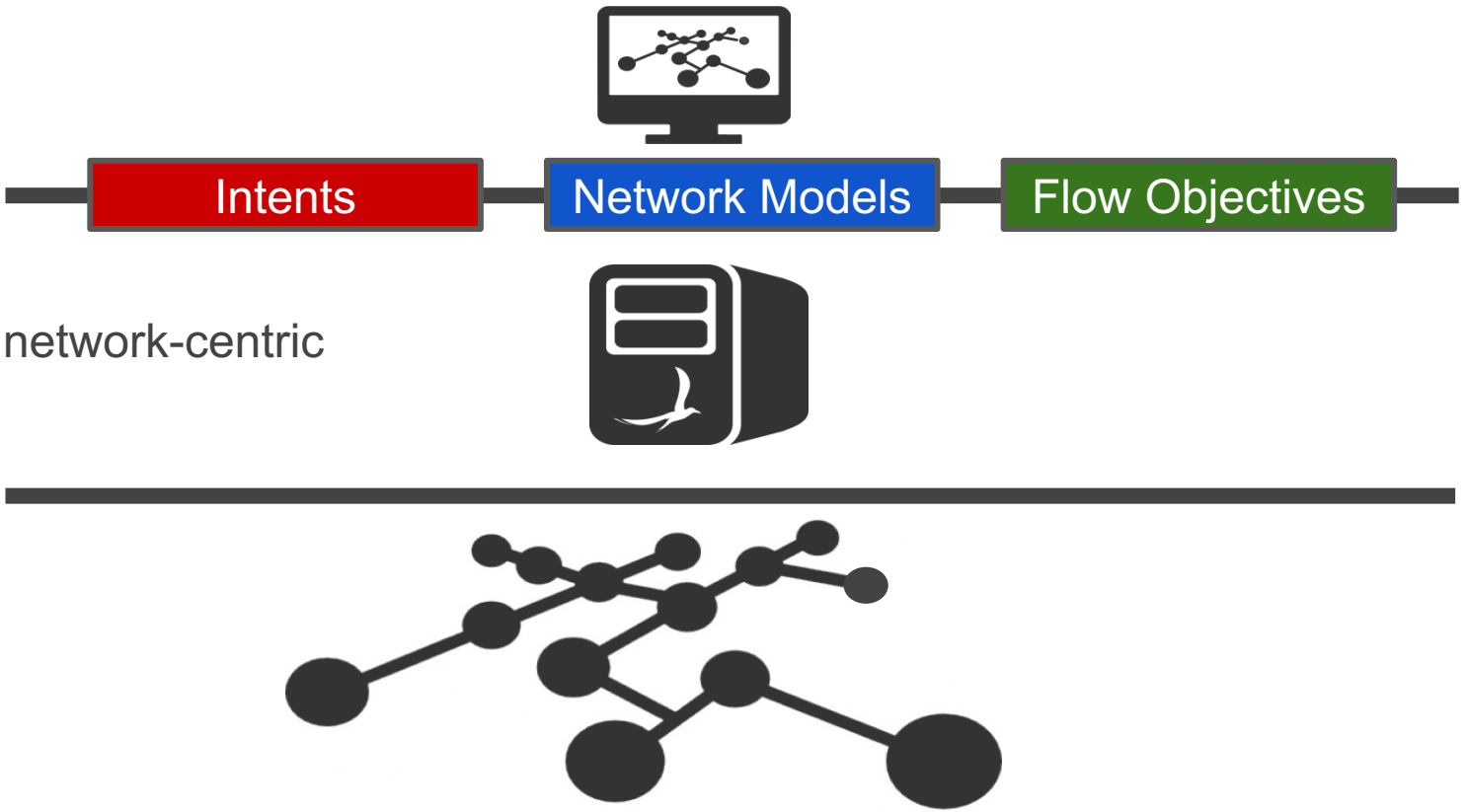
- Eventually Consistent
  - Reads are monotonically consistent
- Low Overhead Reads and Writes
- Gossip based Anti-Entropy Protocol Fixes Divergent Copies
- Generalized as **`EventuallyConsistentMap<K, V>`**



Distributed Architecture (5/5)

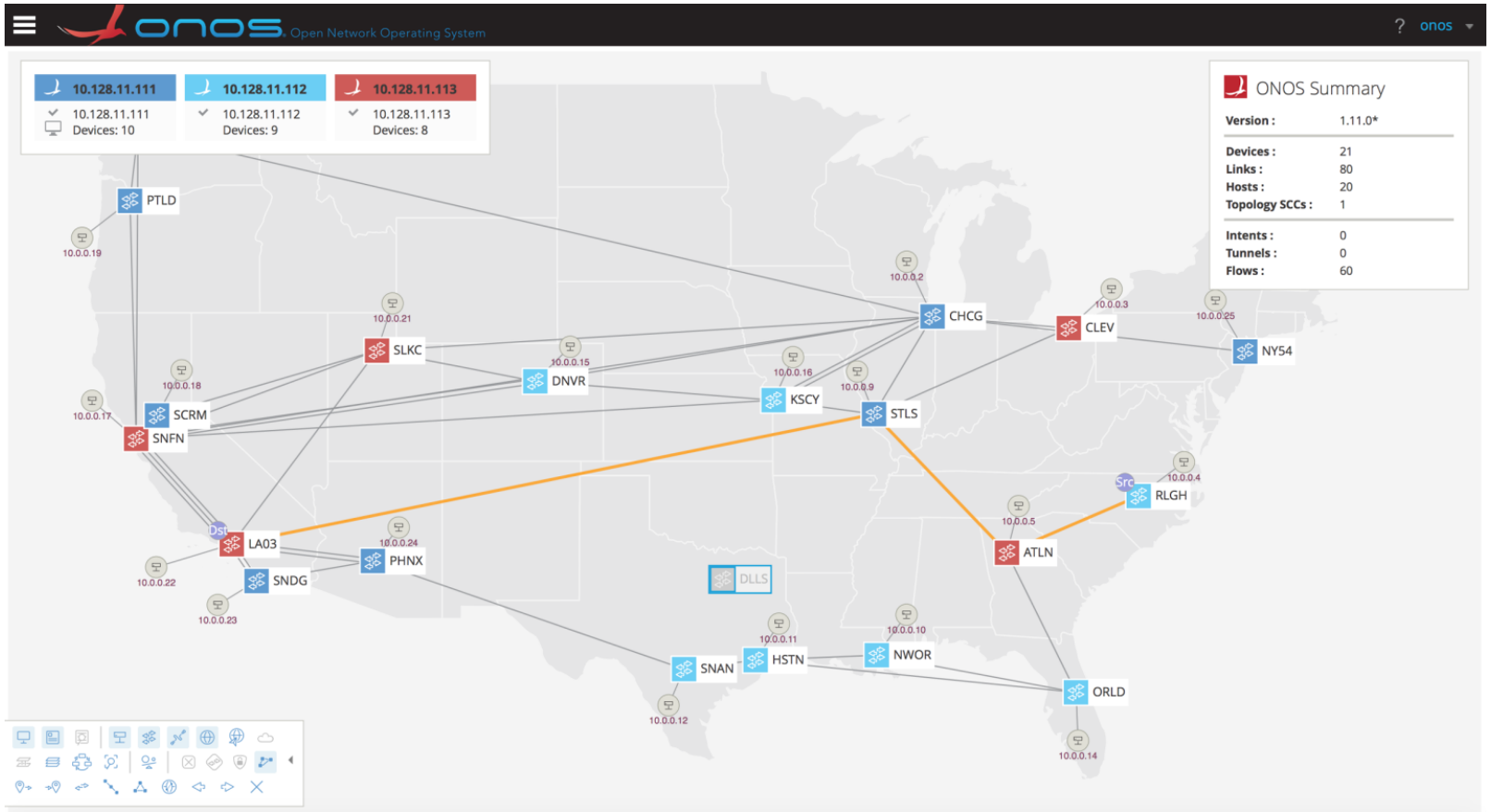
- State Management in ONOS
  - ONOS exposes a set of distributed primitives to cater different use cases
  - Primitives span the consistency continuum
- Distributed Primitives
  - `EventuallyConsistentMap<K, V>`
    - Map abstraction with eventual consistency guarantee
  - `ConsistentMap<K, V>`
    - Map abstraction with strong linearizable consistency
  - `DistributedQueue<E>`
    - Distributed FIFO queue with long poll support
  - `AtomicCounter`
    - Distributed version of Java AtomicLong
  - Etc.

Northbound



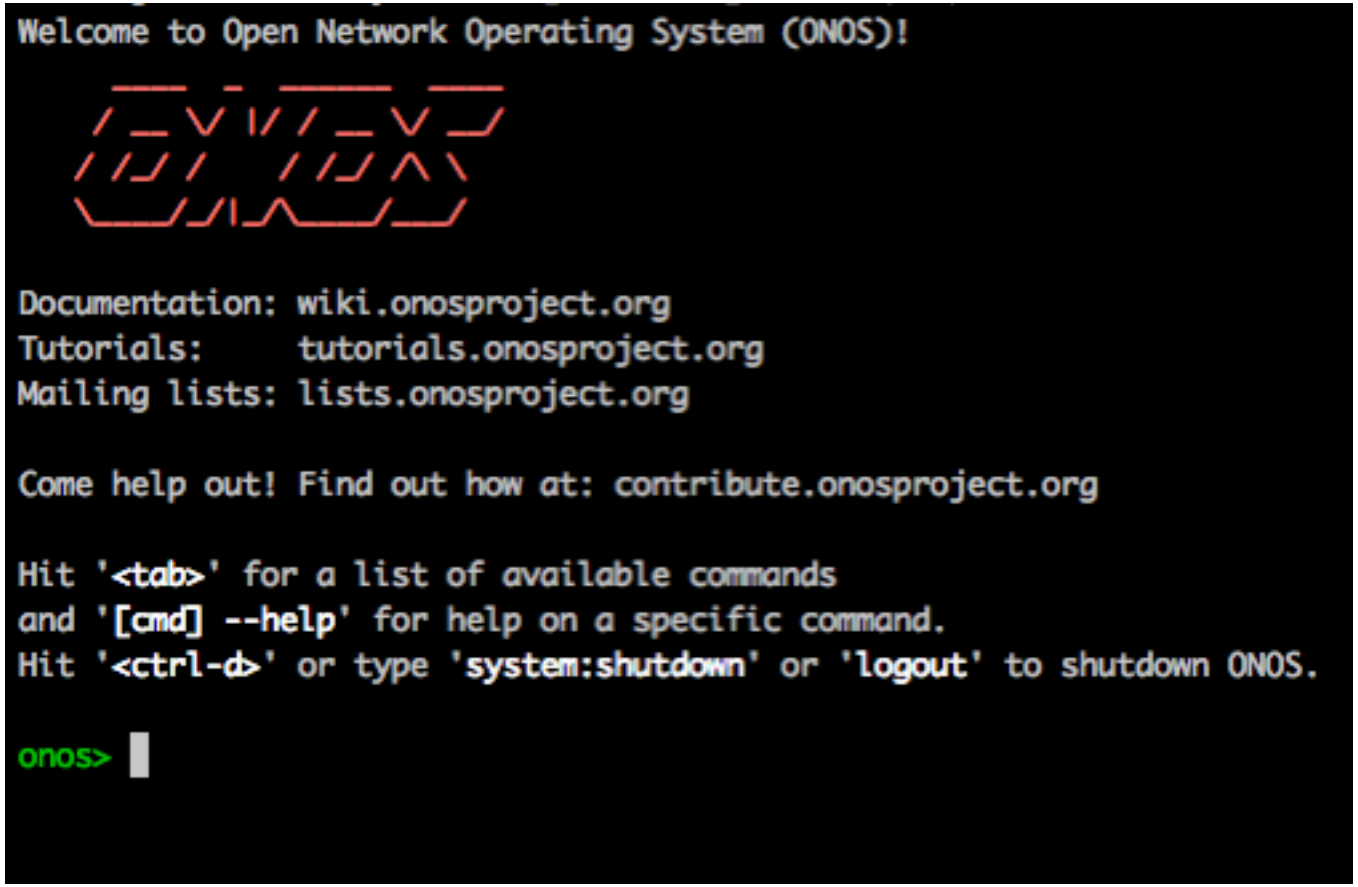
Interact with GUI

- ONOS Web GUI (http://<onos-ip>:8181/onos/ui)
  - A single-page web application



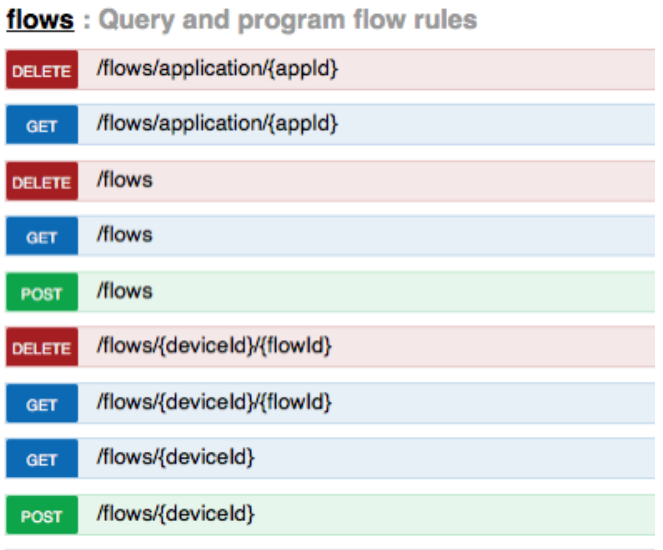
Interact with ONOS CLI

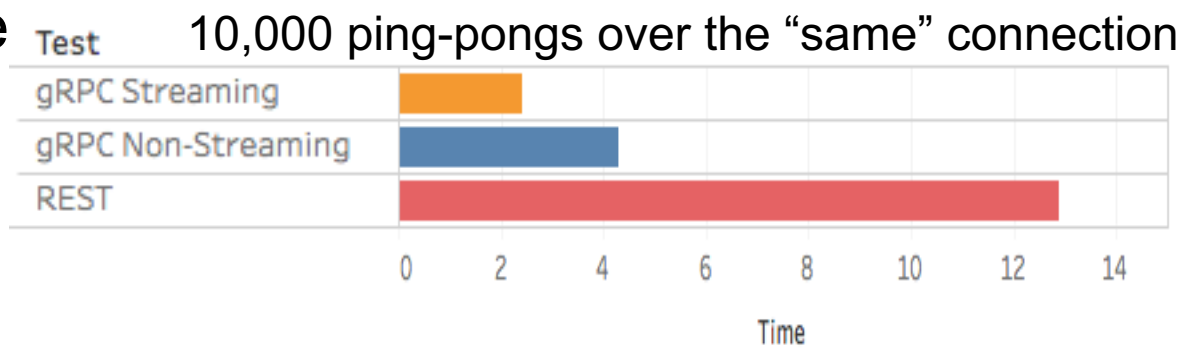
- ONOS CLI
  - An extension of Karaf's CLI
  - Leverage features such programmatic extensibility
- Access CLI
  - Local installation
    - Access with the command ***onos-karaf clean***
  - Remote installation
    - Deploy with ***onos-package*** and ***stc setup***
    - Access with the command ***onos***



Interact with ONOS REST and gRPC

- REST API
  - Provides a way to interact with off-platform applications
  - JSON, HTTP/1.1 based communication
  - Swagger based REST documents
- gRPC
  - Faster access than REST calls by using HTTP/2 connection multiplexing and bidirectional streaming
  - Significantly reduced data size with binary formatted data using protobuf model
  - Remote access to service interface similar to Java APIs



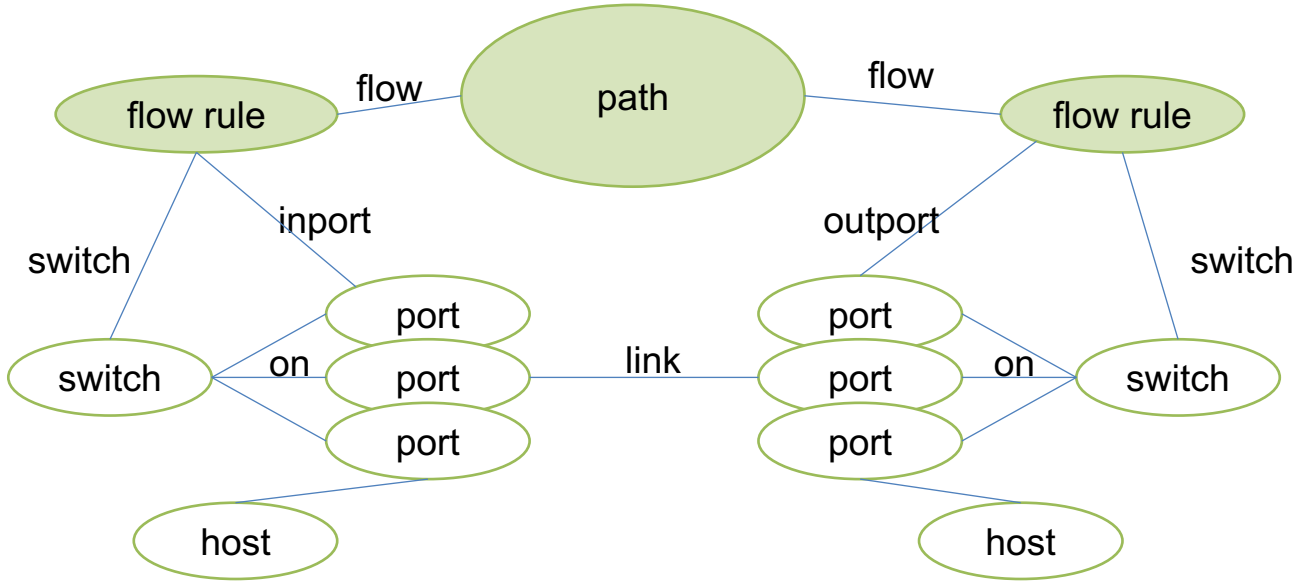


Key Northbound Abstractions

- Network Graph
  - Directed, cyclic graph comprising of infrastructure devices, infrastructure links and end-station hosts
- Flow Objective
  - Device-centric abstraction for programming data-plane flows in table pipeline-independent manner
- Intent
  - Network-centric abstraction for programming data-plane in topology-independent manner

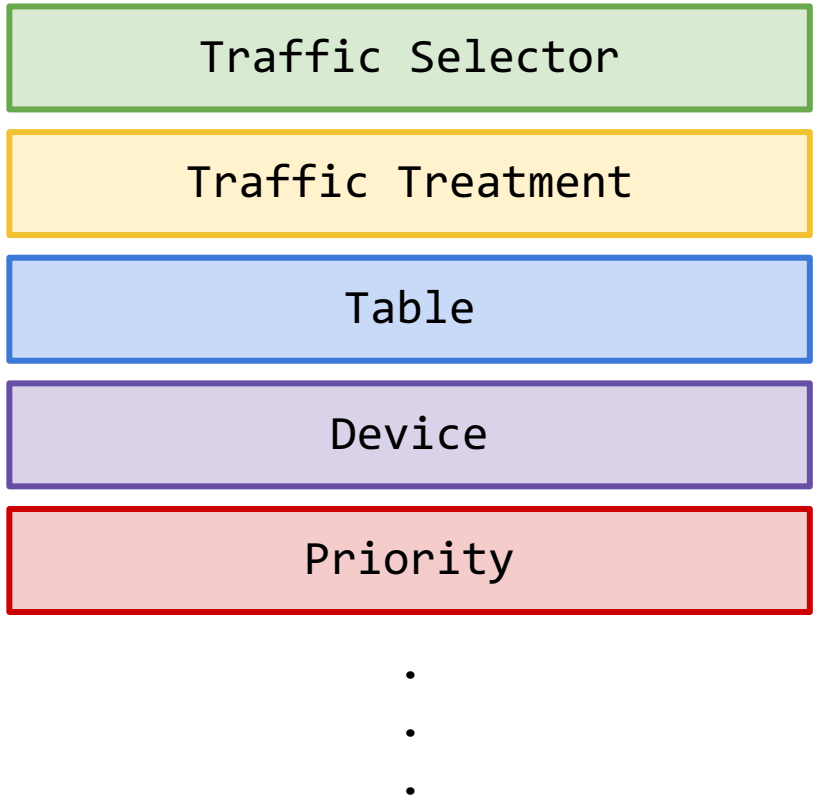
Network Graph

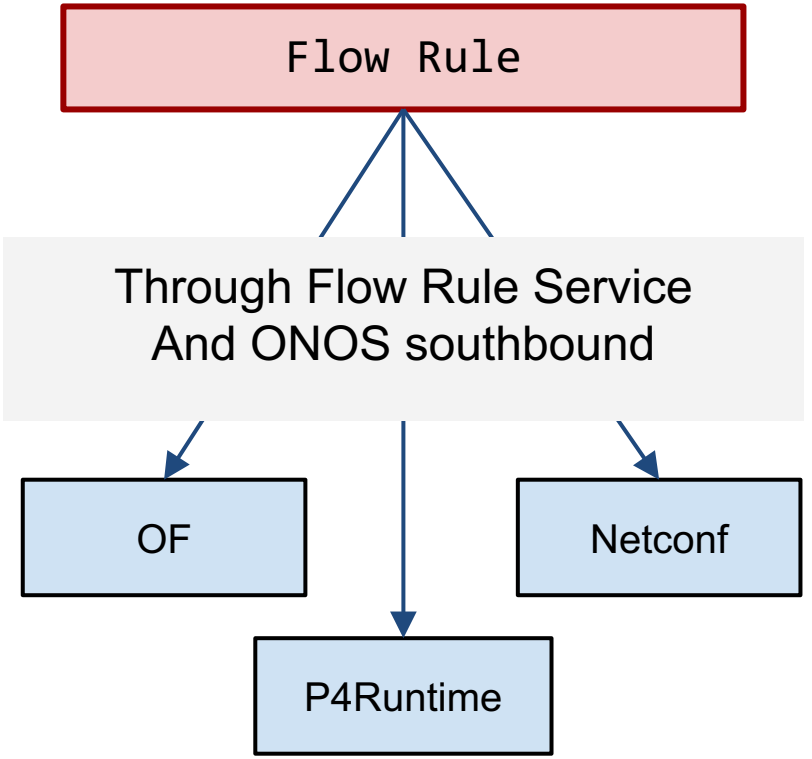
- Network Graph
  - Directed, cyclic graph comprising of infrastructure devices, infrastructure links and end-station hosts
  - Abstract the protocol-specific network element into protocol-agnostic network element (referred as **Model Objects**)
  - Applications are only exposed to Model Objects
  - Model Objects
    - Topology
      - Device, port, hosts, link, etc.
    - Control
      - Flow rule, role value, etc.
    - Packets
      - Outbound/inbound packet



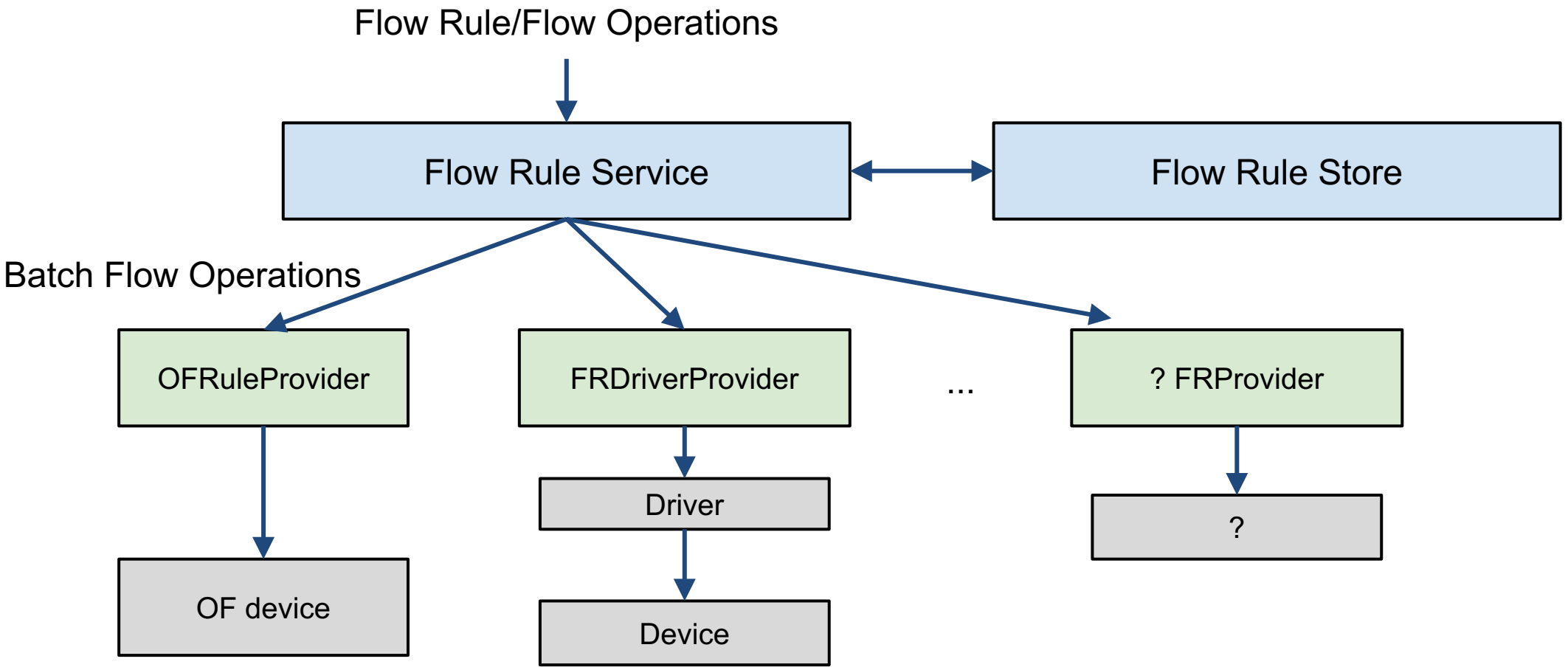
Flow Rule

- Protocol Independent, Pipeline Specific



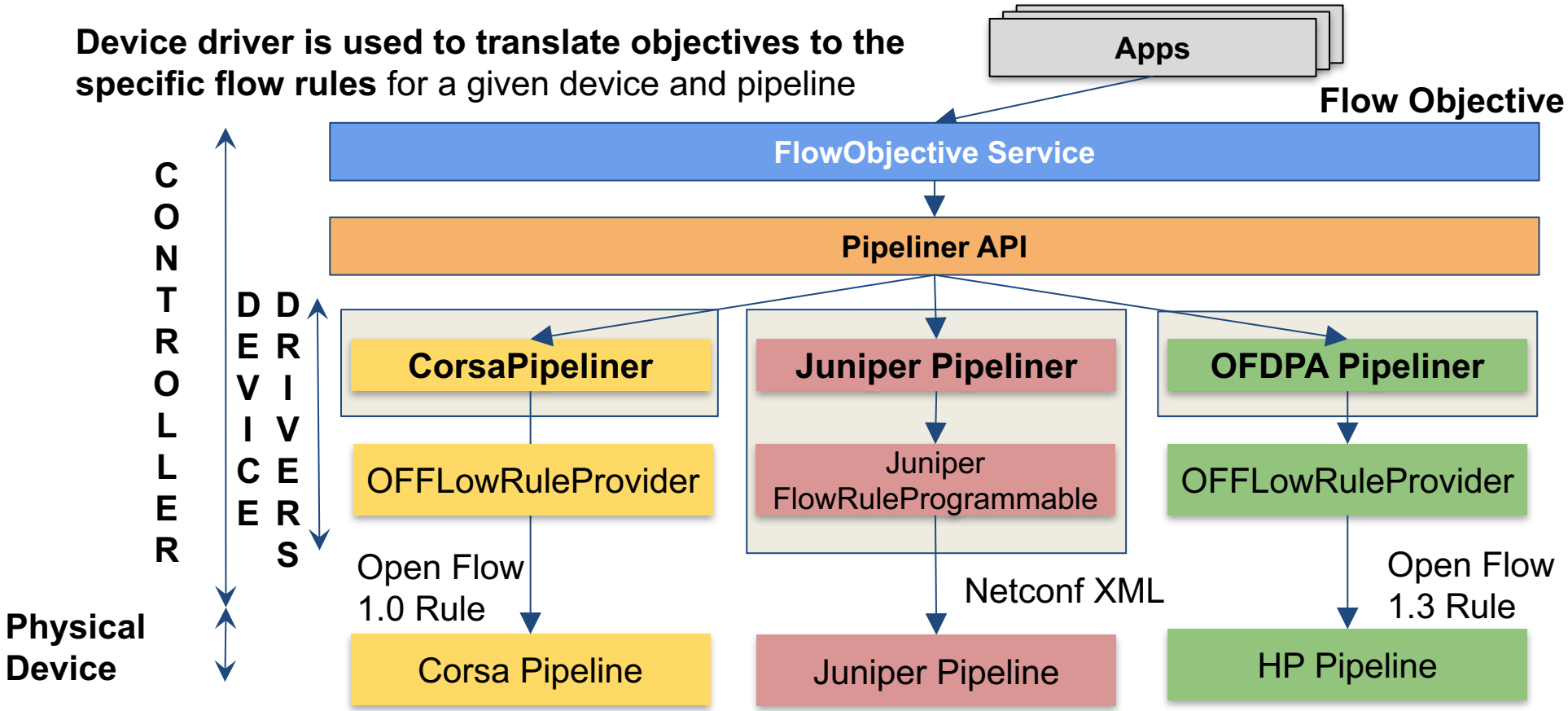


Flow Rule Service Architecture



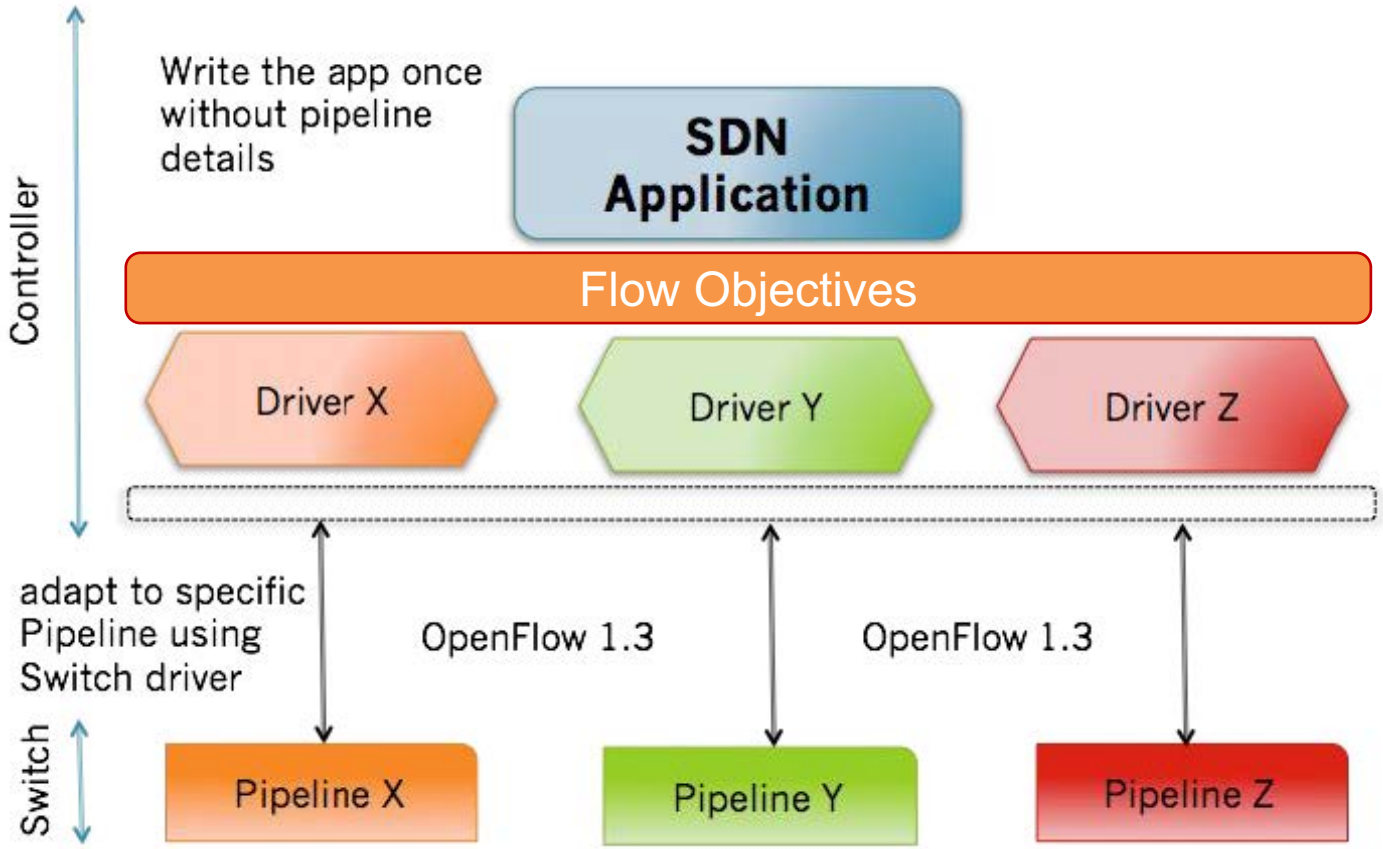
Flow Objective Subsystems (1/5)

- Problem
  - Applications must be pipeline aware, make them applicable to specific HW



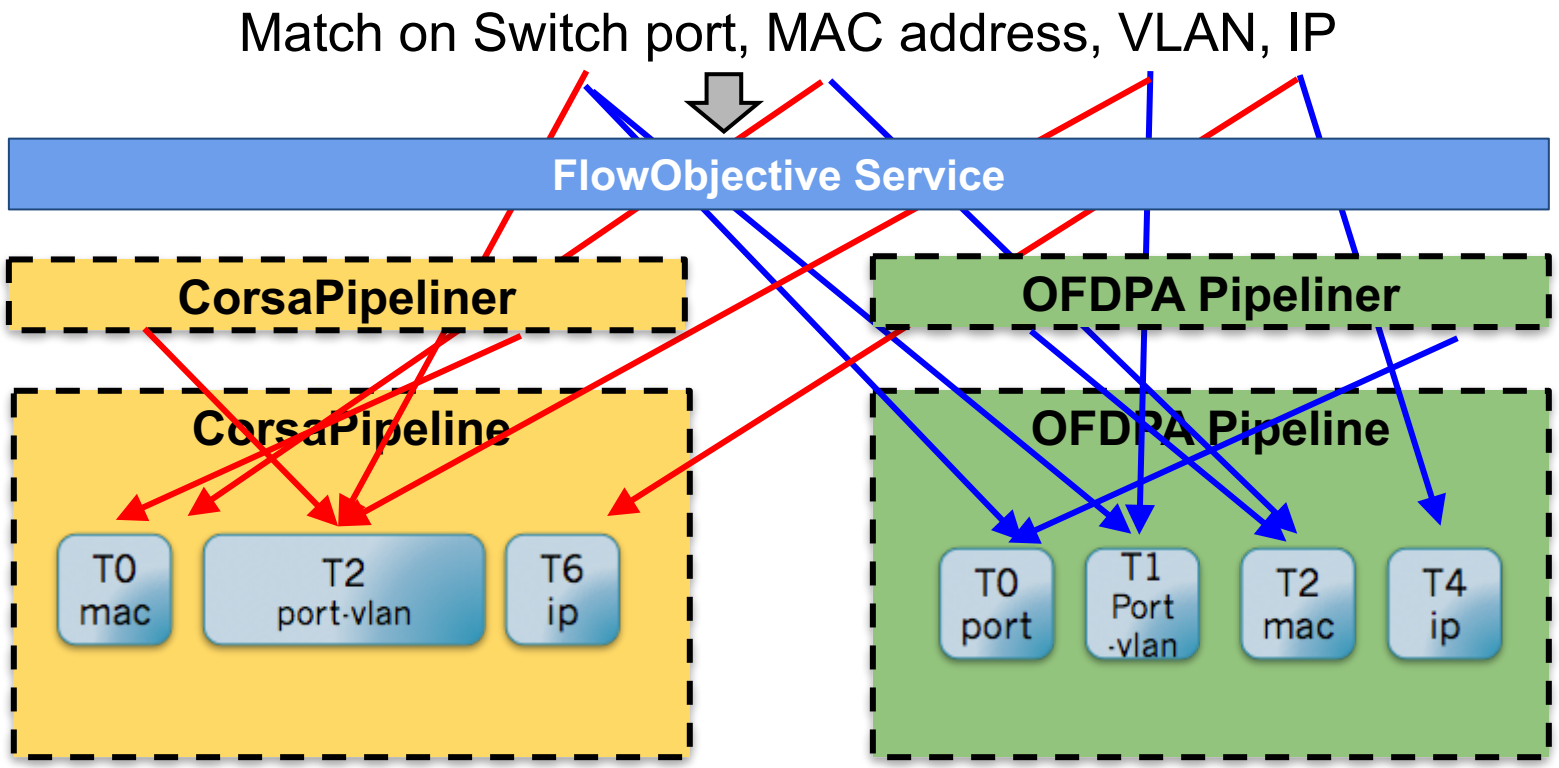
Flow Objective Subsystems (2/5)

- Flow Objective
  - Flow Objectives enable developers to write applications once for all pipelines
  - Flow Objectives describe a SDN application's objective behind a flow it is sending to a device
  - Three Types of Flow Objectives
    - Filtering objective
    - Forwarding objective
    - Next objective



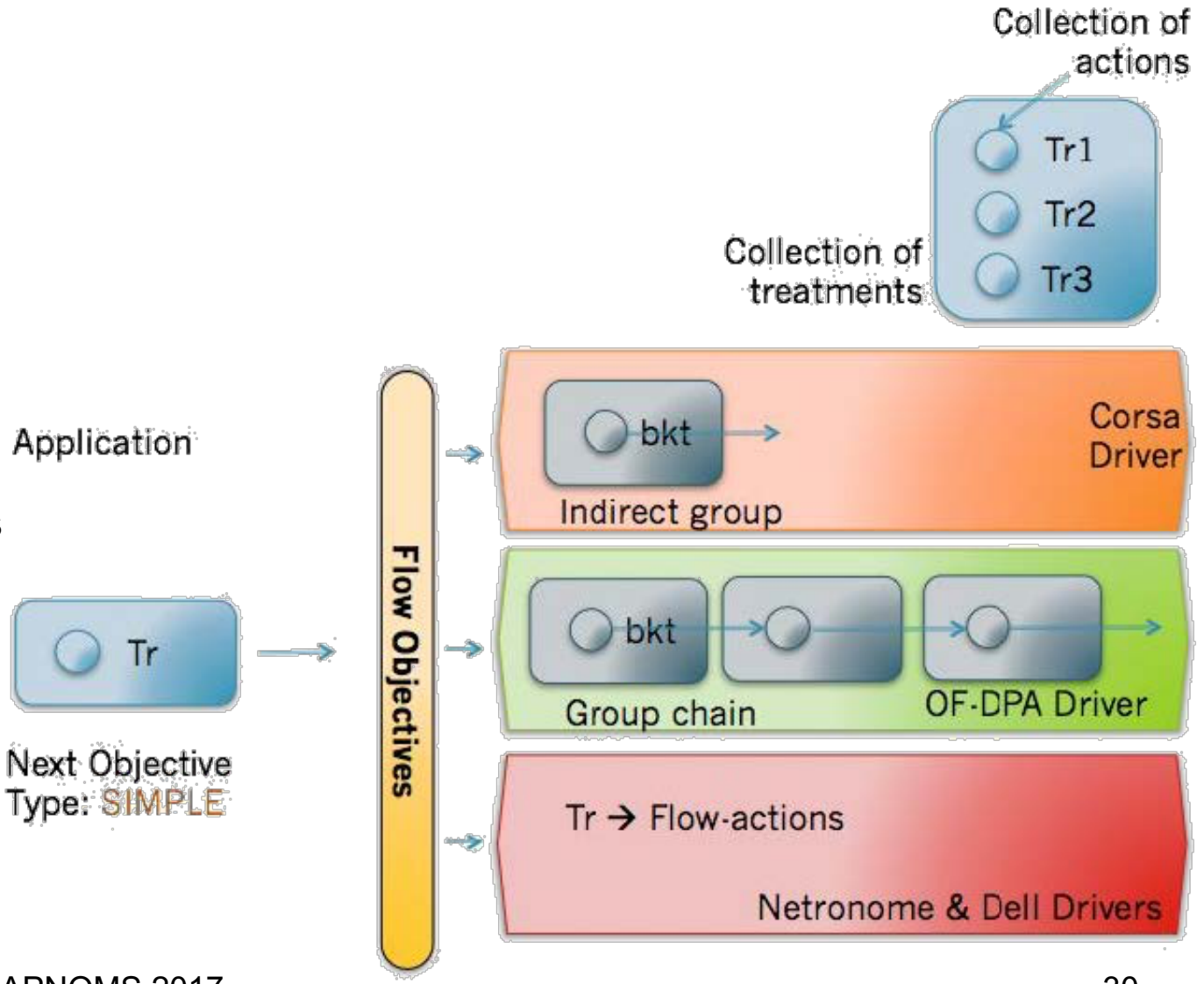
Flow Objective Subsystems (3/5)

- Filtering Objective
  - Only provides either **Permit** or **Deny** operations
  - On criteria (match fields in OpenFlow)



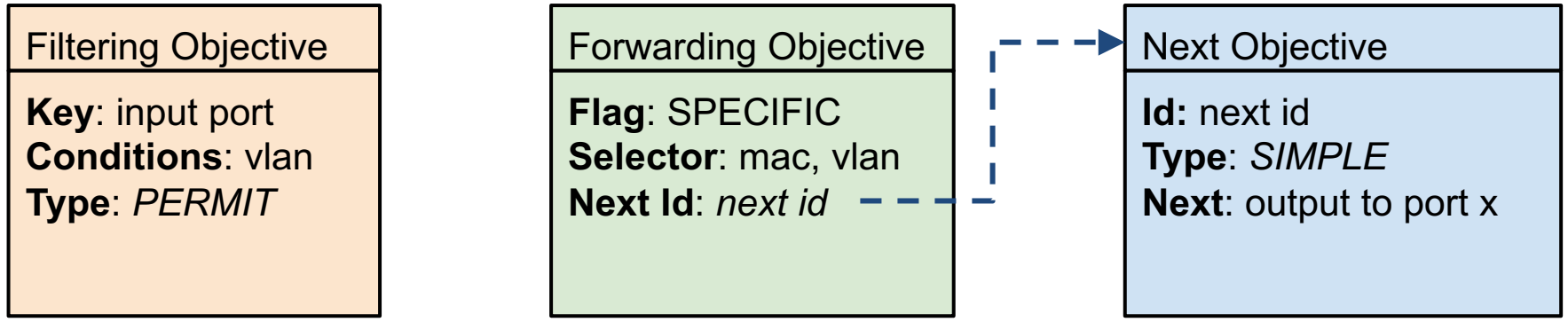
Flow Objective Subsystems (4/5)

- Next Objective
  - Next → next hop for forwarding
  - Keyed by a NextId used in Forwarding Objectives
- Forwarding Objective
  - Forwarding types
    - Specific
      - MAC, IP, MPLS forwarding tables
    - Versatile
      - ACL table

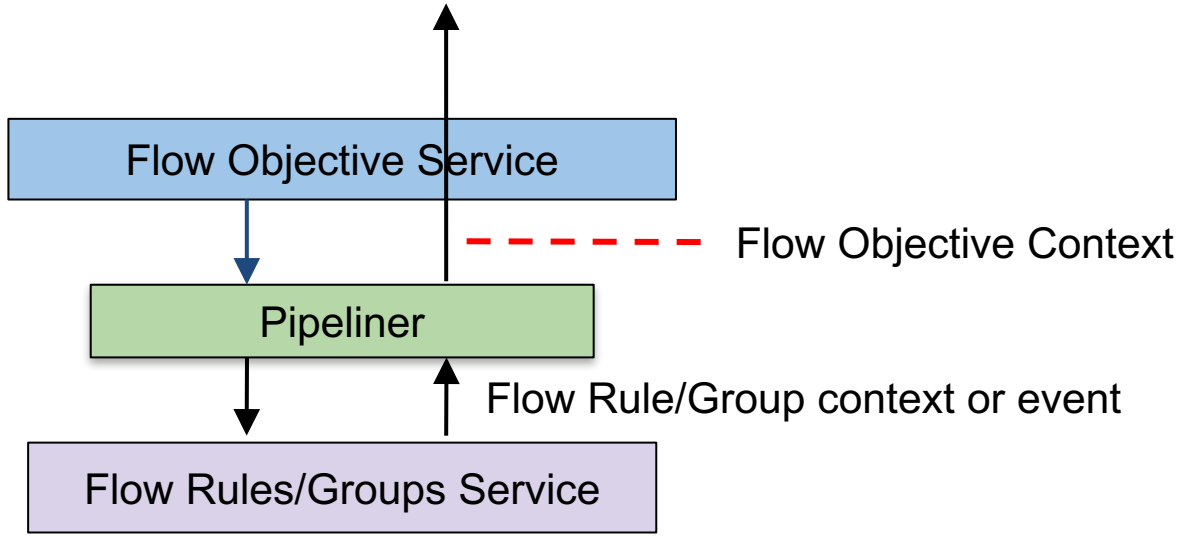


Flow Objective Subsystems (5/5)

- OF-DPA Pipeliner, L2 unicast

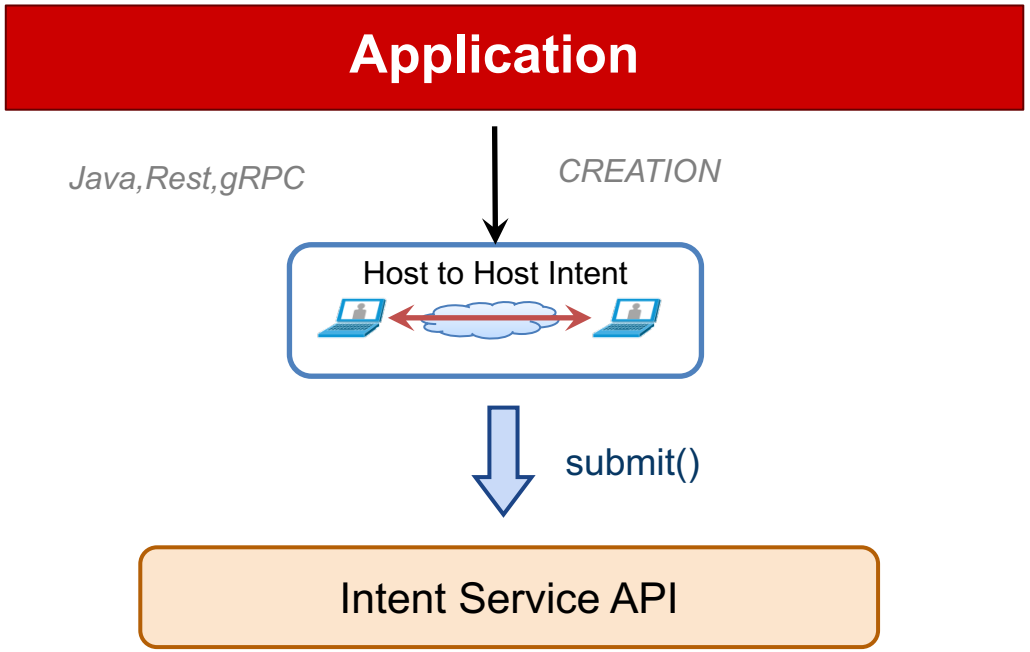


- After…
  - Generates specific **Flow Rules** and **Groups** and installed by *FlowRuleService* or *GroupService*
  - Reports the status by using Flow Objective Context



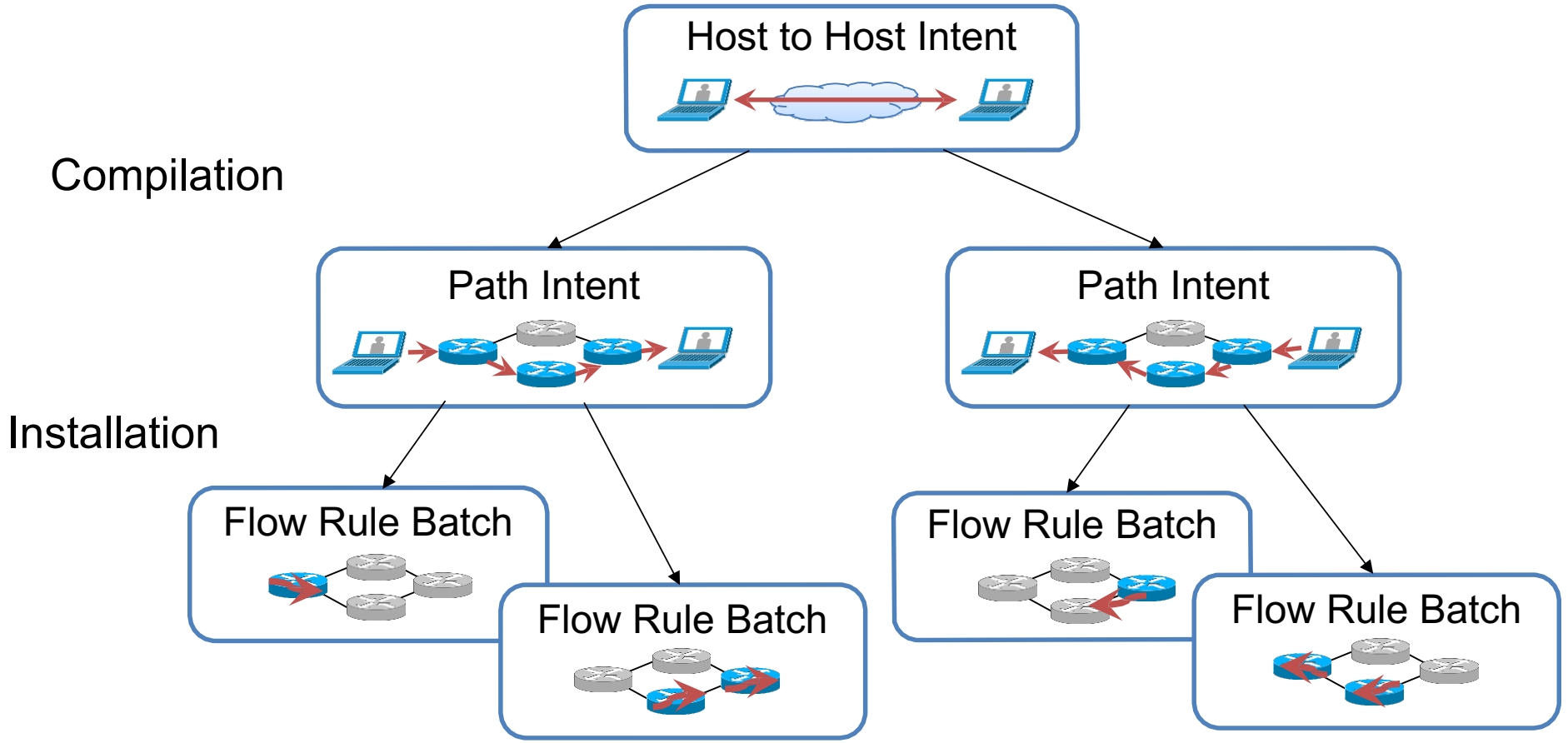
Intent Framework (1/2)

- Intent Framework
  - Providers high-level, network-centric interface that focuses on ***what*** should be done rather than ***how*** it is specifically programmed
  - Abstracts unnecessary network complexity from applications
  - Maintains Requested semantics as network changes

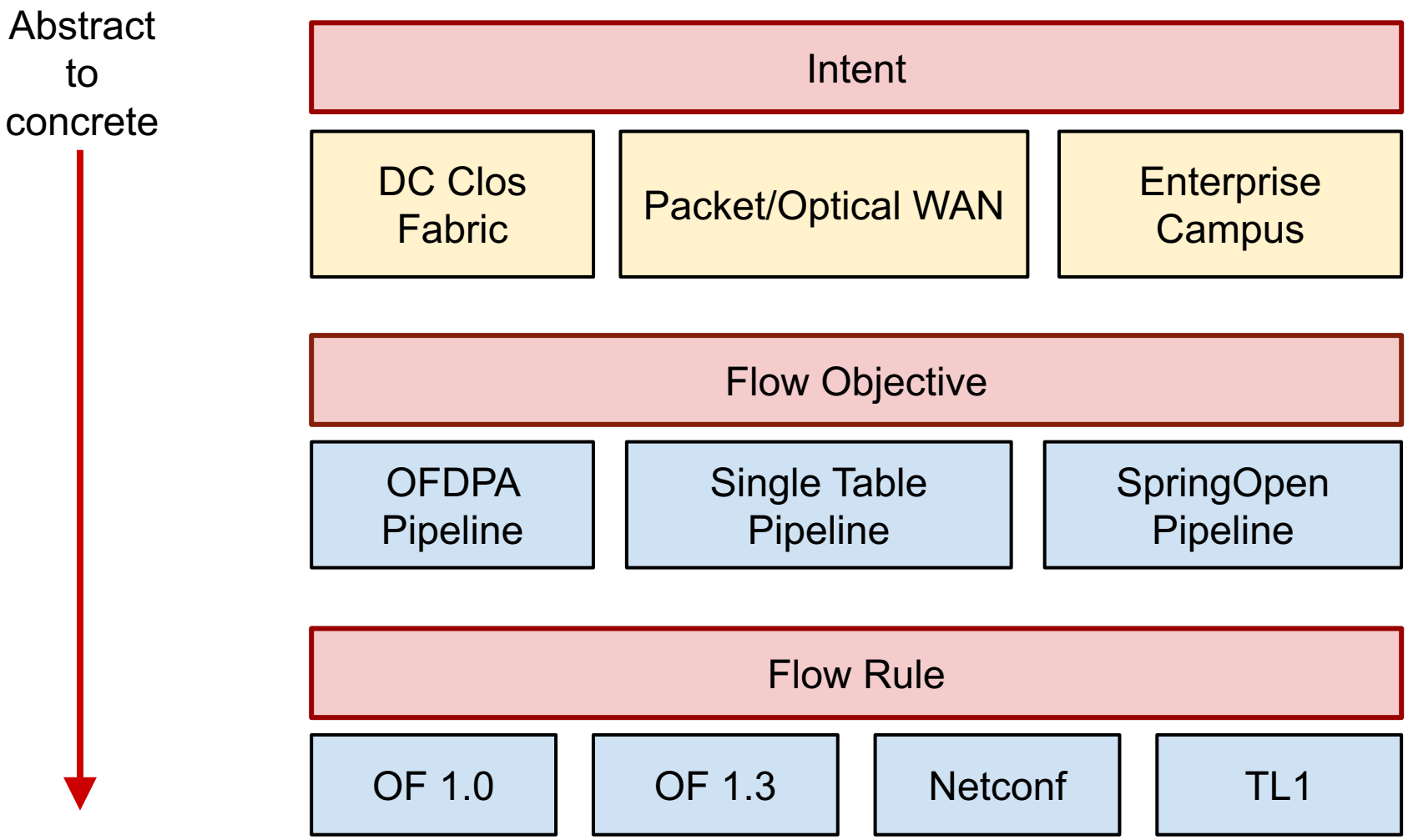


Intent Framework (2/2)

- Compiler & Installer
  - Compiler: produce more specific Intents given the environment
  - Installer: transform Intents into device commands



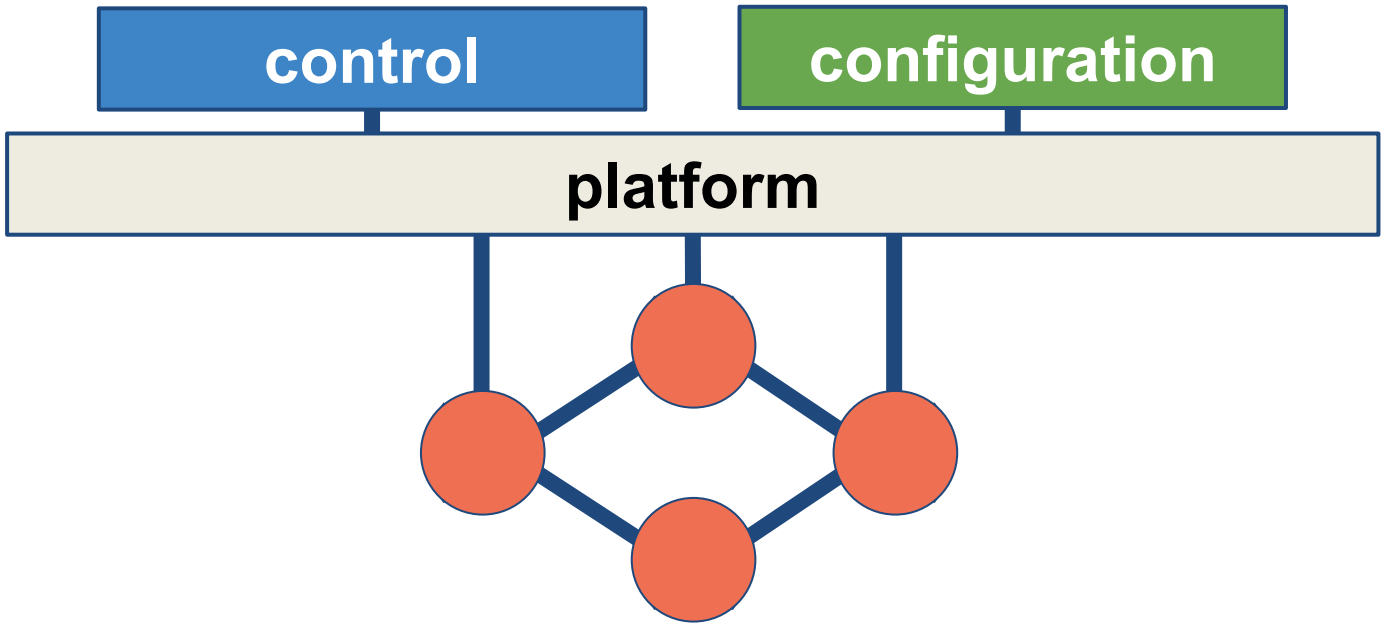
Network Programming



Configuration

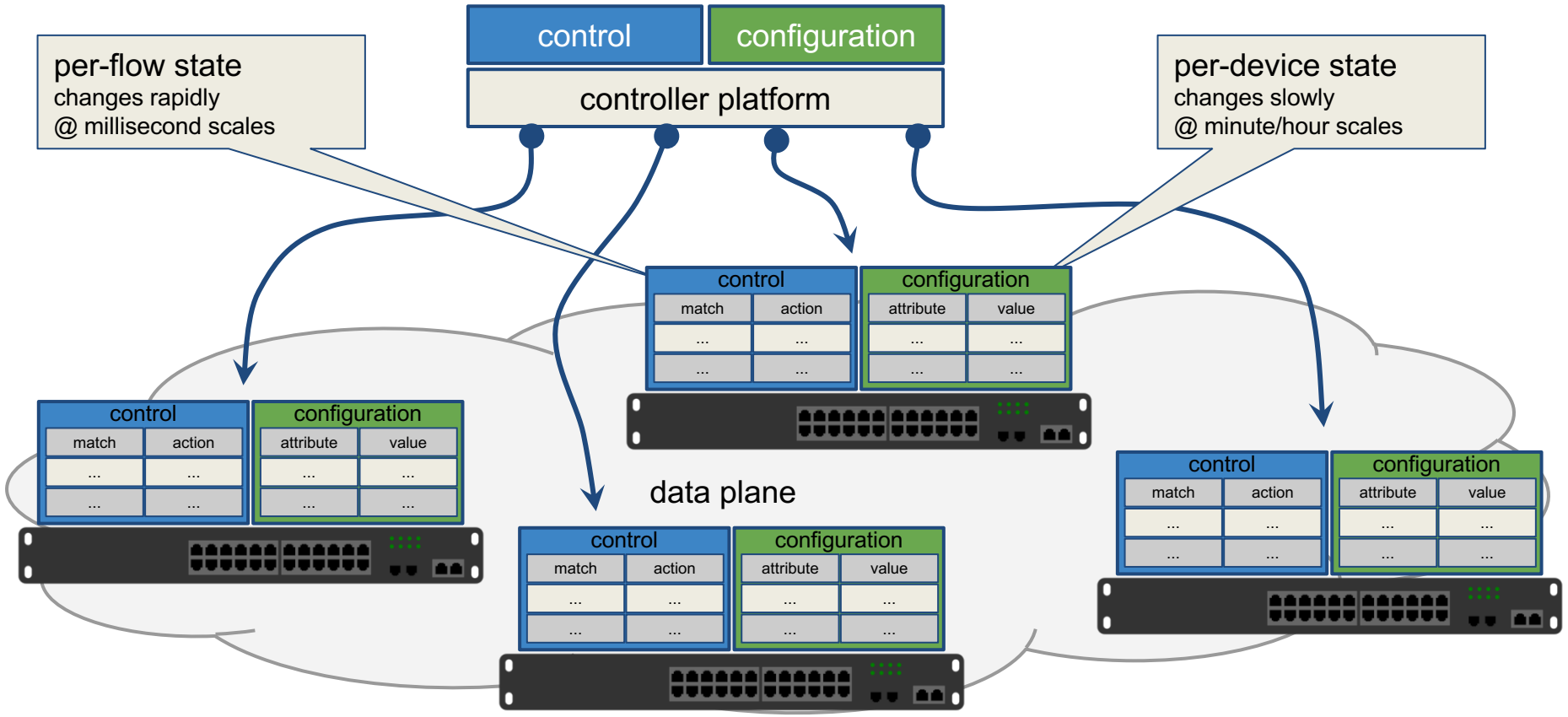
- Network Configuration (netcfg)
  - Provides mechanism for any service to register and receive configuration
- Device Configuration
  - Behaviors abstract the management and configuration aspects of a device
- Dynamic Configuration
  - Enable YANG-based service models to be introduced at runtime
  - Allow applications to implement dynamic services

Dynamic Configuration

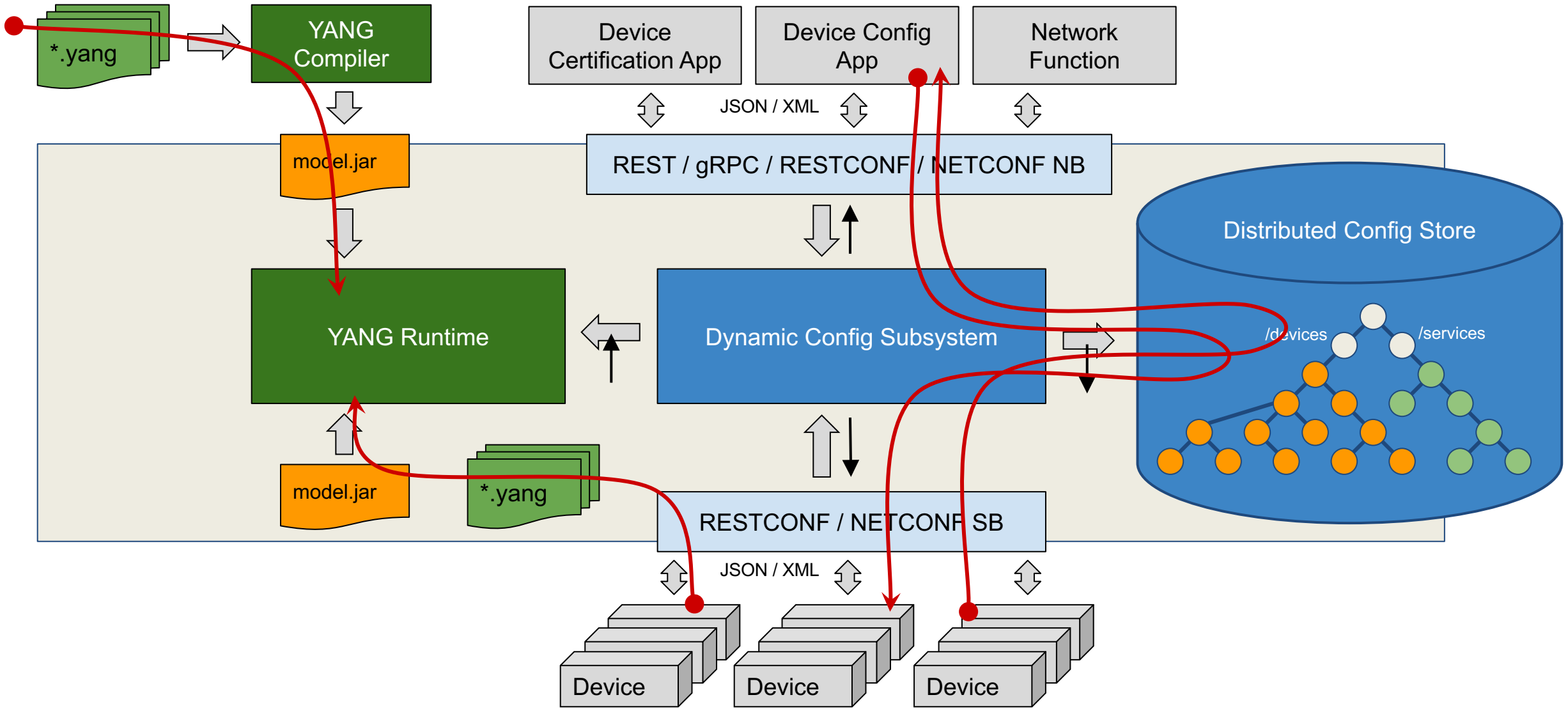


Control and Configuration

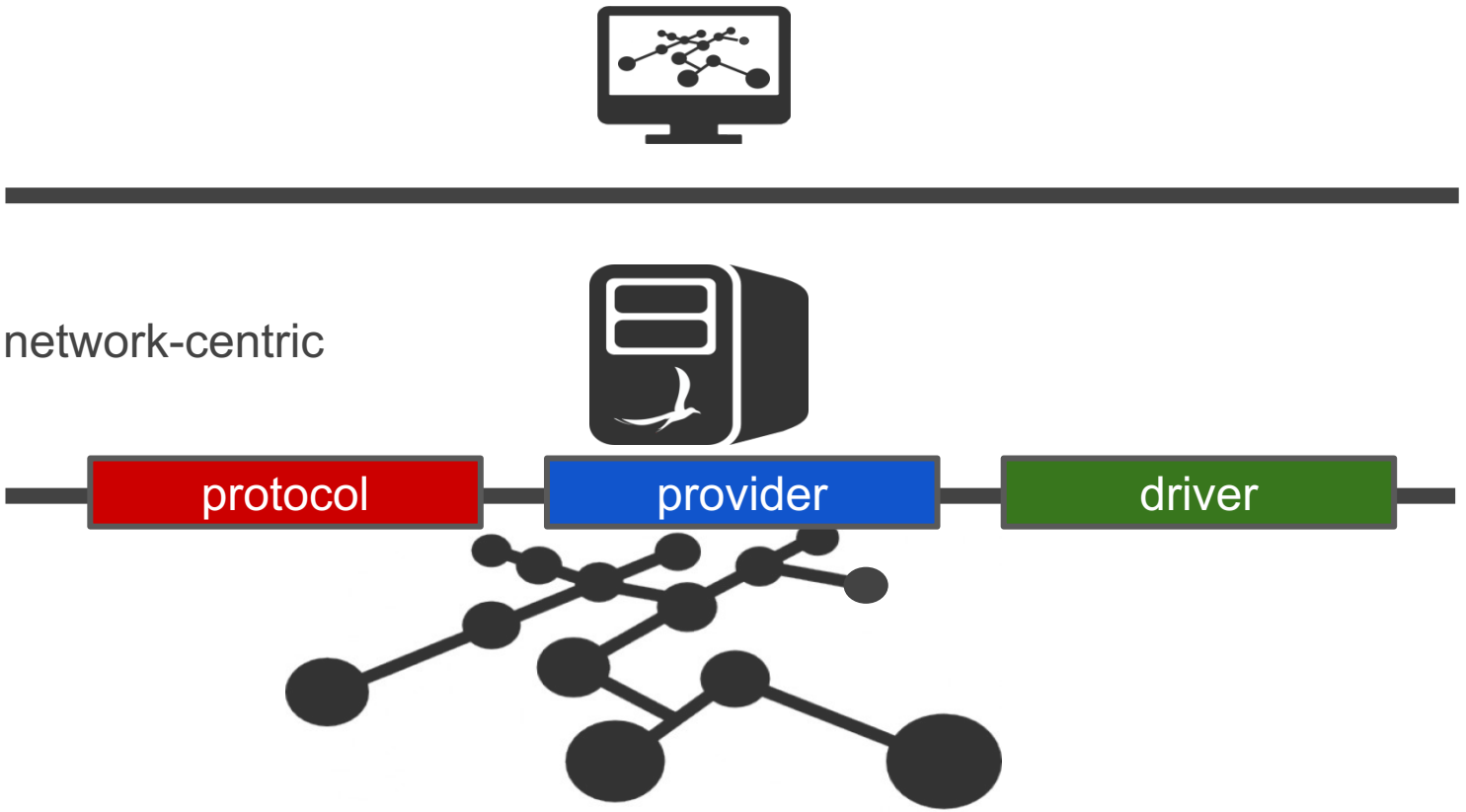
- Operators need a resilient and scalable platform capable of both control and configuration



Dynamic Configuration



Southbound

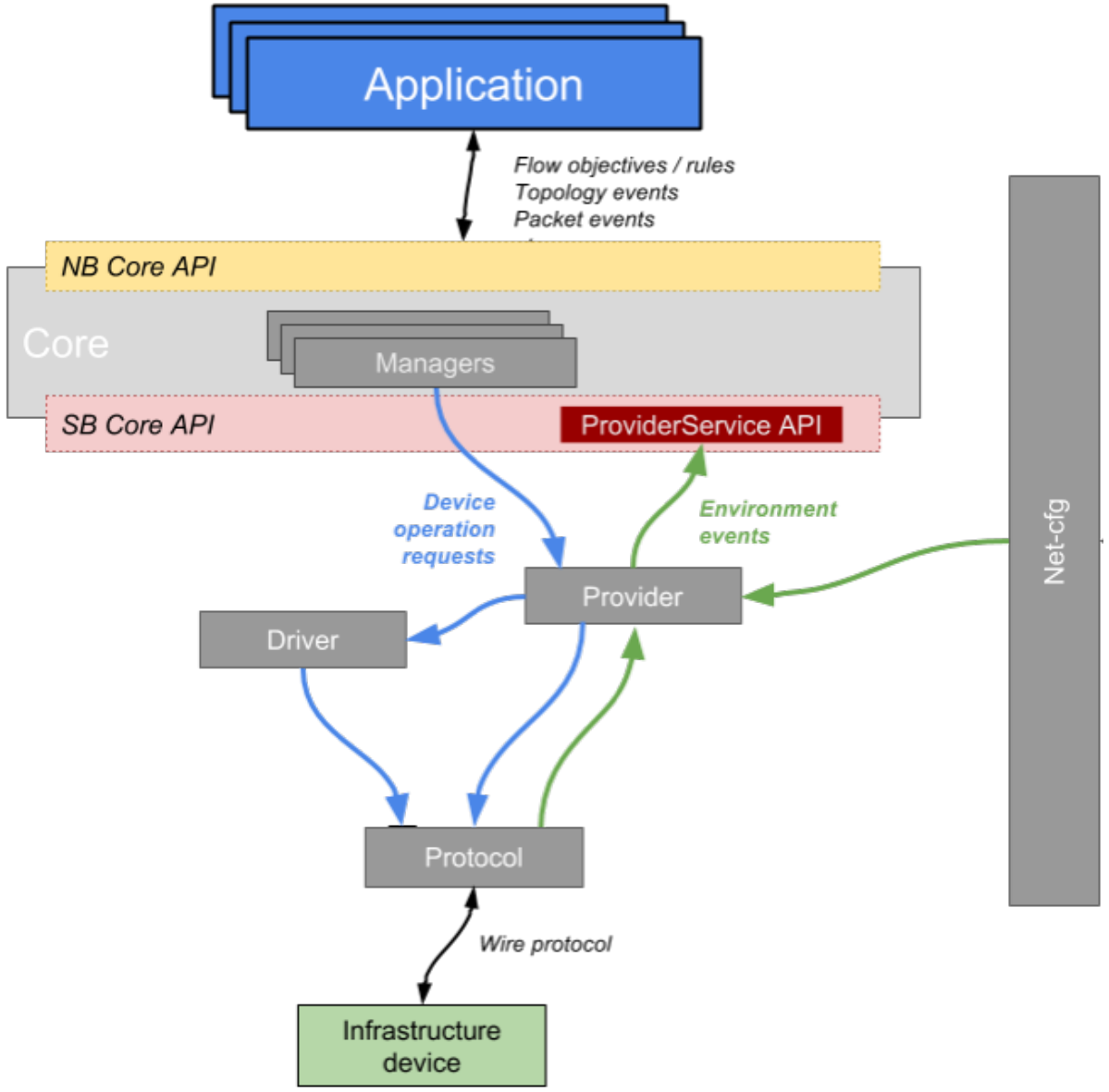


Southbound Overview (1/2)

- Southbound Protocols in 1.11.0
  - OpenFlow 1.3 + optical extension → 1.5 is work-in-progress
  - OVSDB
  - NETCONF + YANG → Yang tools and Yang management system (IETF)
  - SNMP → Simple Network Management Protocol (IETF)
  - P4 → thrift API for BMv2 softswitch
  - BGPLS, ISIS, OSPF → interoperability with legacy network
  - PCEP → Path Computation Element Protocol (IETF)
  - REST and RESTCONF
  - LISP → Locator/Identifier Separation Protocol (IETF)
  - TL1
  - gRPC

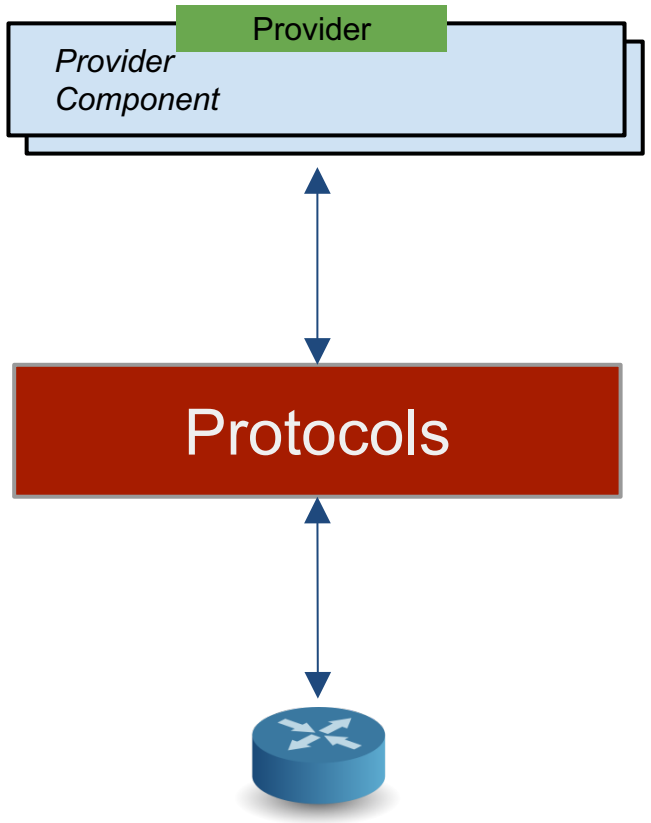
Southbound Overview (2/2)

- SBI Interactions
  - ONOS interacts with the underlying network with the help of its providers
  - Providers
    - Hide complexity from upper layers
  - Protocols
    - Features and modules to communicate with devices
  - Drivers
    - Define specific capabilities offered by the device



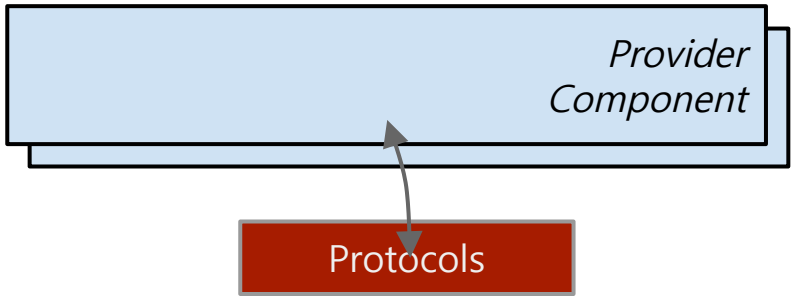
Southbound Protocols

- Features and modules to communicate with devices
- Expose the Standard set of APIs and Enabled Operations
  - OpenFlow
    - FlowMods, GroupMods
  - REST
    - Implements CURD operations
  - NETCONF
    - Open/close session, setConfiguration, getConfiguration
- Usually leverage 3rd party communication libraries
  - Openflowj, snmp4j, thrift, gRPC, netty, etc.



Southbound Providers

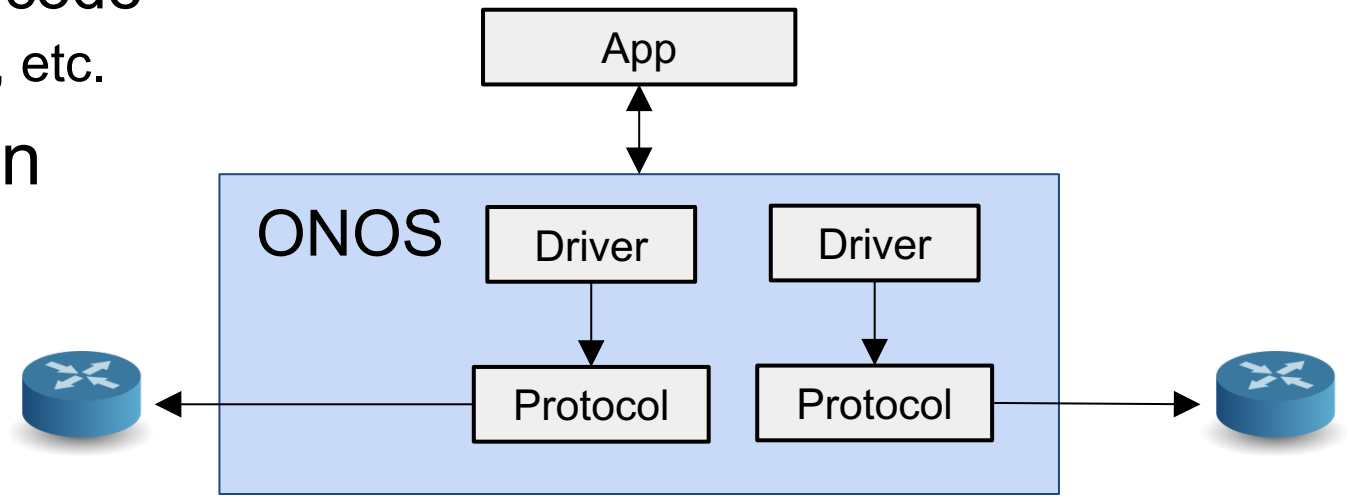
- Role of Providers
  - Translate to and from core abstractions into device specific commands
  - Providers are Used by the Core to (re)act on the Network
    - Up/down of device, links
    - Provisioning of rules, paths, tunnels
    - Receive notifications
  - Correct provider is chosen by the manager via deviceId
    - NETCONF → netconf:127.0.0.1:830
    - OpenFlow → of:00:00:00:00:00:00:00:01
    - LISP → lisp:192.168.1.1



Southbound Drivers

- Device Specific Driver
  - Collection of behaviors
  - On-demand activation
- Abstraction via Behaviors
  - Define specific capabilities offered by the device
  - Encapsulate specific logic and code
    - Port, controller, FlowRule, power, etc.
- Encapsulate Single Interaction
  - Protocol
  - Information

```
<driver name="default" manufacturer="ON.Lab"
    hwVersion="0.0.1" swVersion="0.0.1">
        <behaviour api=Tracking
                   impl=TrackingImpl />
</driver>
```

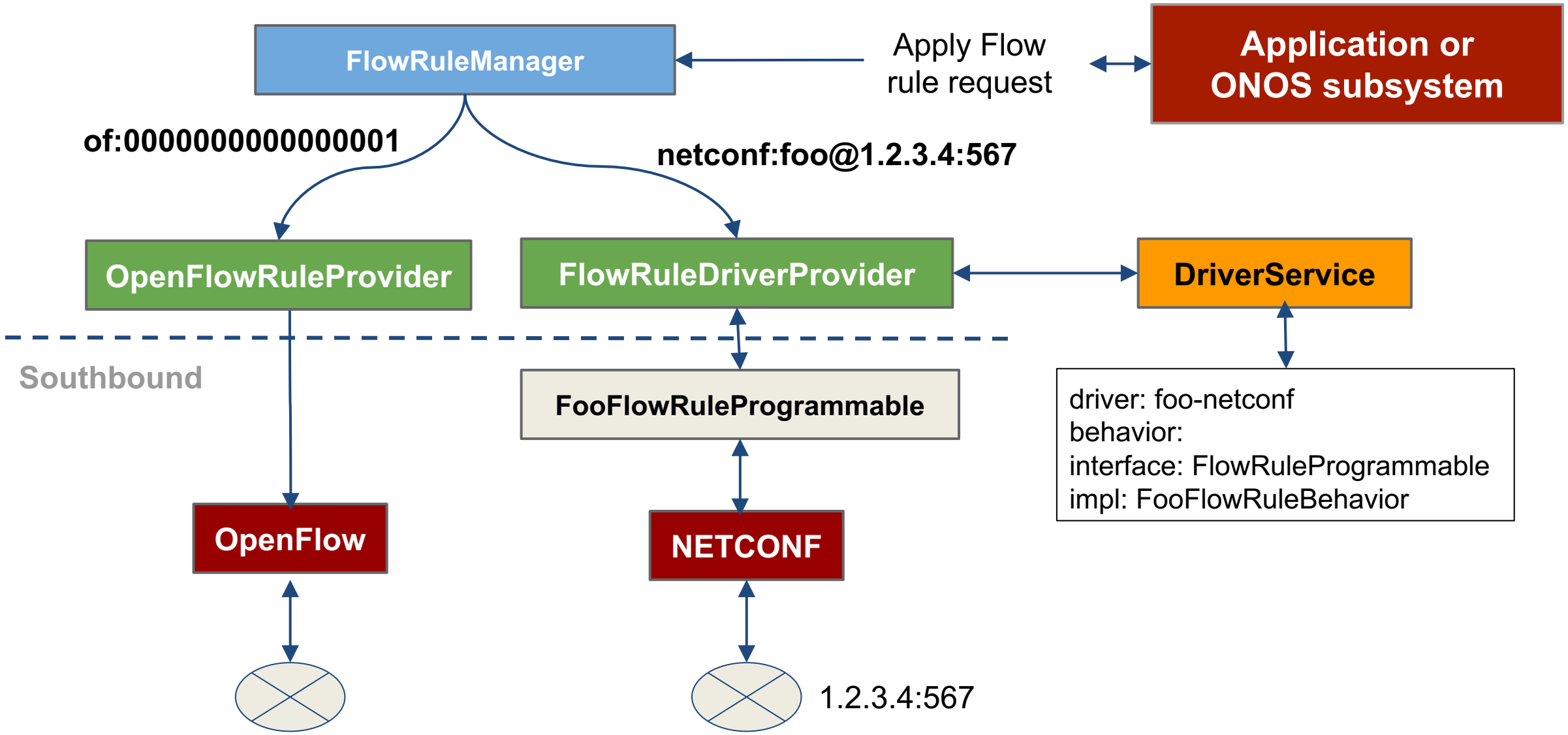


ONOS Core Behaviors

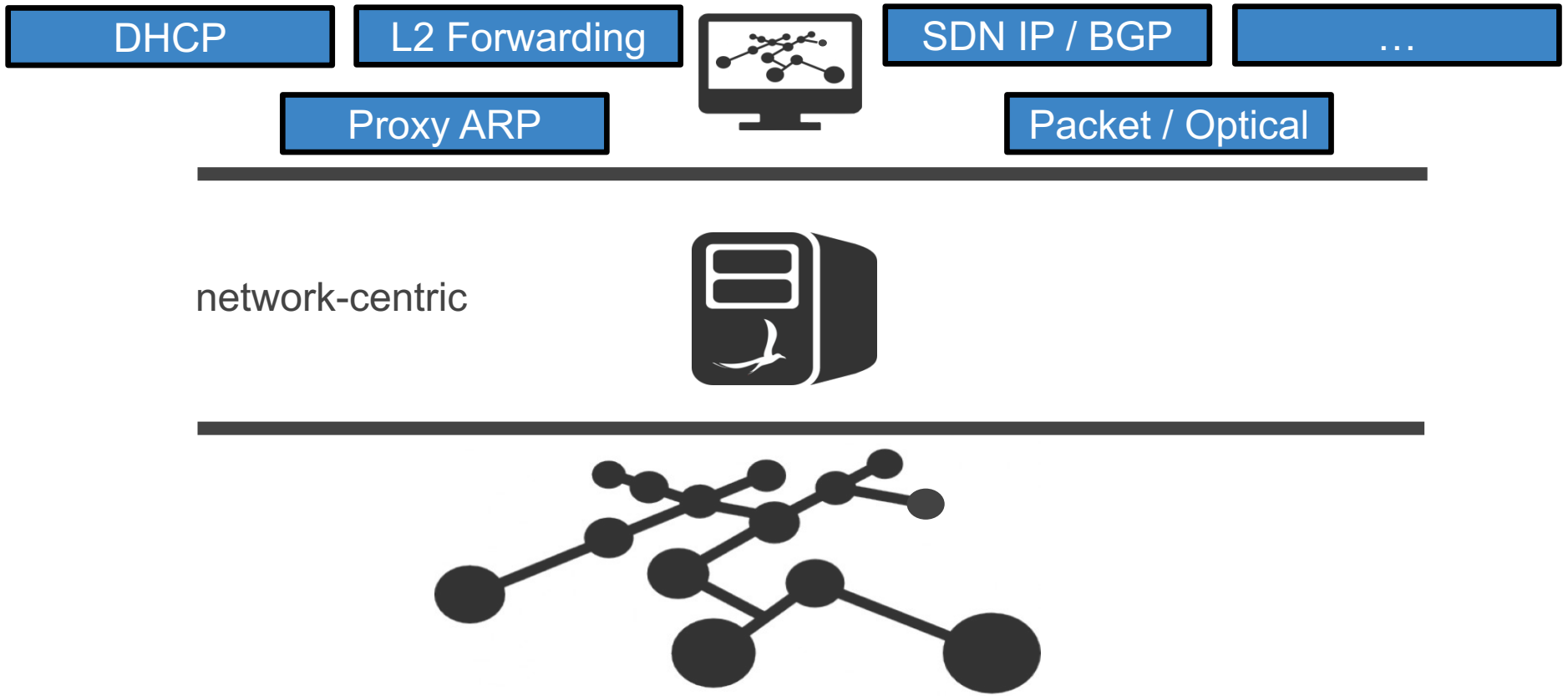
- APIs
  - DeviceDescriptionDiscovery → device description and ports
  - FlowRuleProgrammable → translates to and from ONOS core FR abstraction
  - PacketProgrammable → emit a given packet from a device
  - GroupProgrammable → translates to and from ONOS Groups
  - PortStatisticsDiscovery → statistics of Device Ports
  - Pipeliner → Pipeline abstraction, FlowObjectives to pipeline specific FR
- Recently Introduced
  - DeviceHandshaker → device handshake for GeneralDeviceProvider
  - PipelineProgrammable → installs a Programmable Pipeline on the Device

```
<behaviour api=InterfacePath
           impl=ImpementationPath />
```

Example: FlowRuleProgrammable

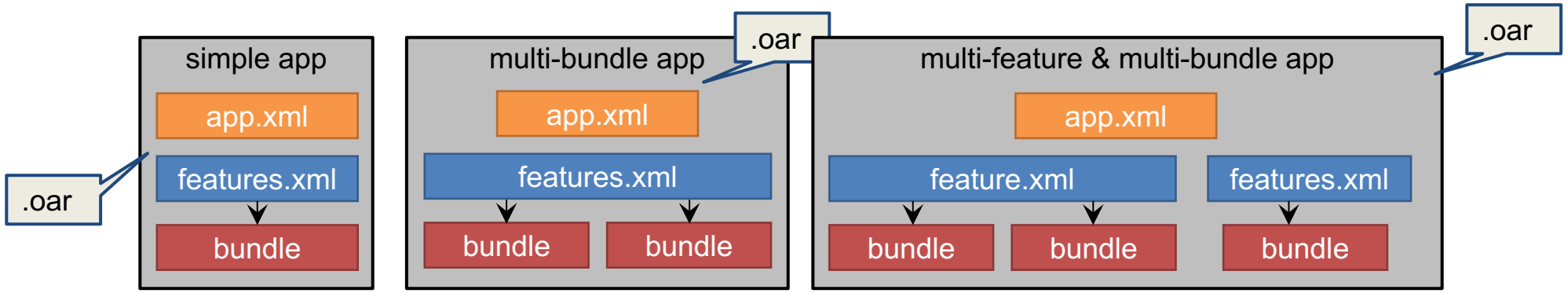


Applications



Application Overview (1/2)

- Overview
  - Interact with the northbound Java, REST, gRPC interface
  - Device and protocol agnostic
  - Augment ONOS through modularity
  - Provide GUI, REST, CLI and distributed stores
- Application Package
  - Applications can be packaged into a single .oar (ONOS Application aRchive)
    - .oar file is a JAR file contains all artifacts
      - E.g., app.xml, features.xml and a set of (OSGi) bundles



Application Overview (2/2)

- Application Types
  - Application as a mere Component
    - Offers no API, self-contained
  - Application with Service Interface
    - Offers API for other applications (e.g., CLI, REST API, web GUI)
  - Application may have its own state
    - Delegates responsibility for tracking state
- Example Applications
  - SDN-IP peering
  - Multi-level provisioning
  - Virtual Tenant Network (VTN)
  - Proxy ARP
  - Reactive forwarding

RESTful

Minimal ONOS Application Project

- 4 Steps of Creating an ONOS Application
  - Create minimal app project via onos-create-app tool
    - Internally use maven archetype to generate a template maven project

```
$ onos-create-app app org.oneping oneping 1.0-SNAPSHOT
```

  - Import into IDE and edit our app pom.xml file
  - Code your application
  - Build the app via maven and deploy it via onos-app tool

```
$ mvn clean install
$ onos-app ONOS_IP_ADDRESS install! target/oneping*.oar
```

ONOS Recap

OpenStack Integration

Segment Routing

SDN-IP

*CORD

DHCP Server

Control Plane Manager

Virtual Tenant Networks

Load Balancer

ProxyARP

Distributed DPI

Service Function Chaining

Virtual Private LAN

Fault Management

TL1

OSPF

REST

IS-IS

SNMP

P4

PCEP

LISP

BGP

NETCONF

OVSDB

OpenFlow

ONOS Release History

Q4/14 Avocet
Base Architecture

Q4/15 Emu
OPNFV
SONA
AARNET
KREONET-S

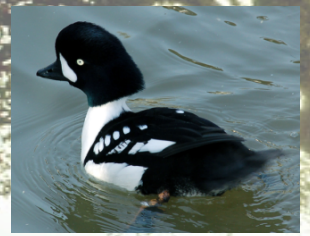
Q4/16 Ibis
BUCK Build Tool
Trellis Fabric enhancement
LISP SBI support, REST
Client, FatTree simulator

Q1/15 Blackbird
Performance

Q1/16 Falcon
ONS Use Cases
{A, E, M} CORD
Disaggregated ROADM
Global R&E Deployment

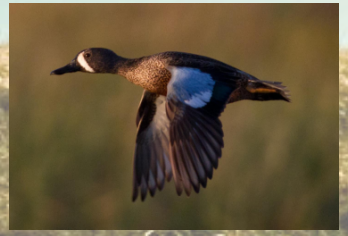
Q1/17 Junco
TL1 SBI support
Virtualization support
Regionalization support
Dynamic conf. enhancement

Q2/15 Cardinal
ONS Use Cases
SDN-IP
Packet Optical
R-CORD



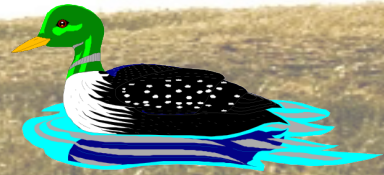
Q2/16 Goldeneye
CPMan Apps
Intents using Flow Objectives
P4 DEMO support
YANG tool chain

Q2/17 Kingfisher
YANG Tools 2.0
OpenFlow 1.4 support
Intent F/W imprement
vRouter, OpenROADM support



Q3/15 Drake
ONF ATRIUM
Secure Mode ONOS
VxLAN
Device Configuration

Q3/16 Hummingbird
RabbitMQ, Kafka Message buses
YANG NBI, SBI CODECs
ACTN Traffic Engineering
Distributed system primitives
SB - OSPF, ISIS



Q3/17 Loon
Coming soon…

ONOS Roadmap

Loon

Core
- *Atomix improvements*
- *Flow Rule store refactor*

Dynamic Configuration
- *OpenConfig models*
- *Live YANG compiler*
- *RESTCONF NB*
- *Config Synchronizer design*

ISSU
- *Portable Kryo serialization*
- *Upgrade Coordinator design*

P4
- *PI Framework*
- *P4Runtime*
- *BMv2 driver*

gRPC

Virtualization

GUI
- *Build & test enhancements*
- *Initial localization*

QA
- *Business performance paper*

Magpie

Dynamic Configuration
- *Config tree per device via RESTCONF*
- *Configuration Synchronizer*
- *Dynamic Store transactions design*
- *YANG 1.1 (partial support)*
- *Documentation*

ISSU
- *Upgrade Coordinator*
- *Core Stores upgradeability*

Core
- *Migrate distributed primitives to Atomix*
- *Dynamic scaling and repartitioning*

P4
- *P4Runtime enhancements*
  - *Groups*
  - *Device mastership*
- *gNMI*

gRPC
- *External app demonstration*
- *Core services*

GUI
- *Topo2 overlay migration*
- *Remaining localization*

QA
- *Document tree primitive tests*
- *Test framework for ISSU*
- *Technical Performance White Paper*

Build & Infrastructure
- *STC validation*
- *Bazel investigation*
- *Code disaggregation*

Incubation

P4
- *VNF offloading to HW (BNG)*
- *INT (in-band telemetry)*
- *P4 Fabric*
- *Dynamic program loading*

Virtualization
- *External connectivity*
- *OF Agent*
- *OpenStack integration*

N

Dynamic Configuration
- *Sharding & optimizations*

ISSU
- *App Stores upgradability*
- *Compatibility check tooling*

P4
- *gNMI enhancements*
- *INT*
- *P4 Fabric*
- *Dynamic program loading*

gRPC
- *Select app services*
- *Explore automated generation*

GUI
- *Search & filtering*
- *Additional views*
- *Accessibility*

Virtualization
- *Snapshotting & embedding*
- *Additional SB*
- *Resiliency*

QA

ONOS Community

ONOS Partners

Leading service providers make ONOS & SDN/NFV solutions relevant to them

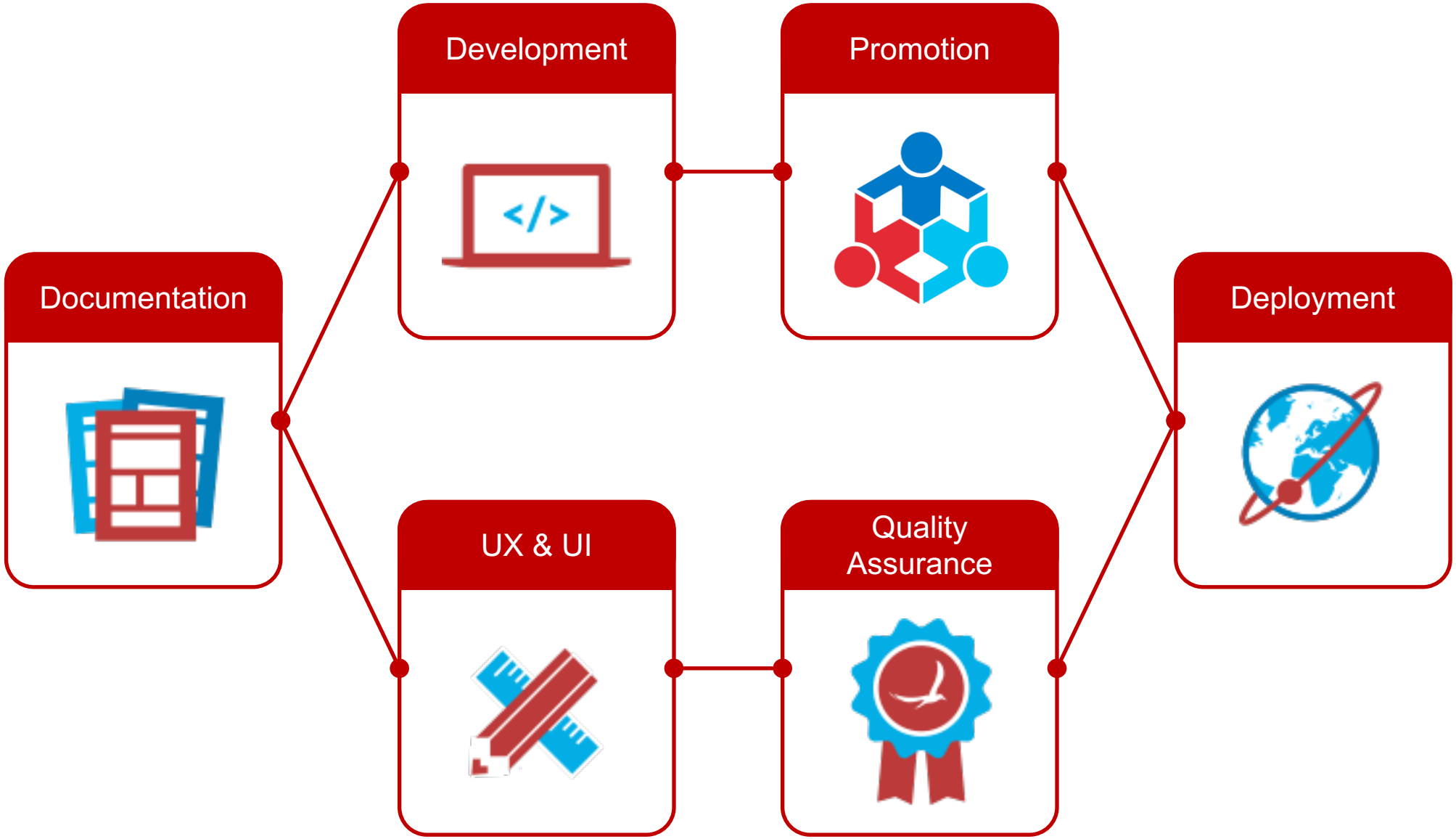
Leading vendors help make ONOS and SDN/NFV solutions real & ready for deployment

ONOS Collaborators

Collaborating organizations help grow the community and grow the impact

20 new collaborators so far in 2016 and more are joining every month

Contribution Opportunities



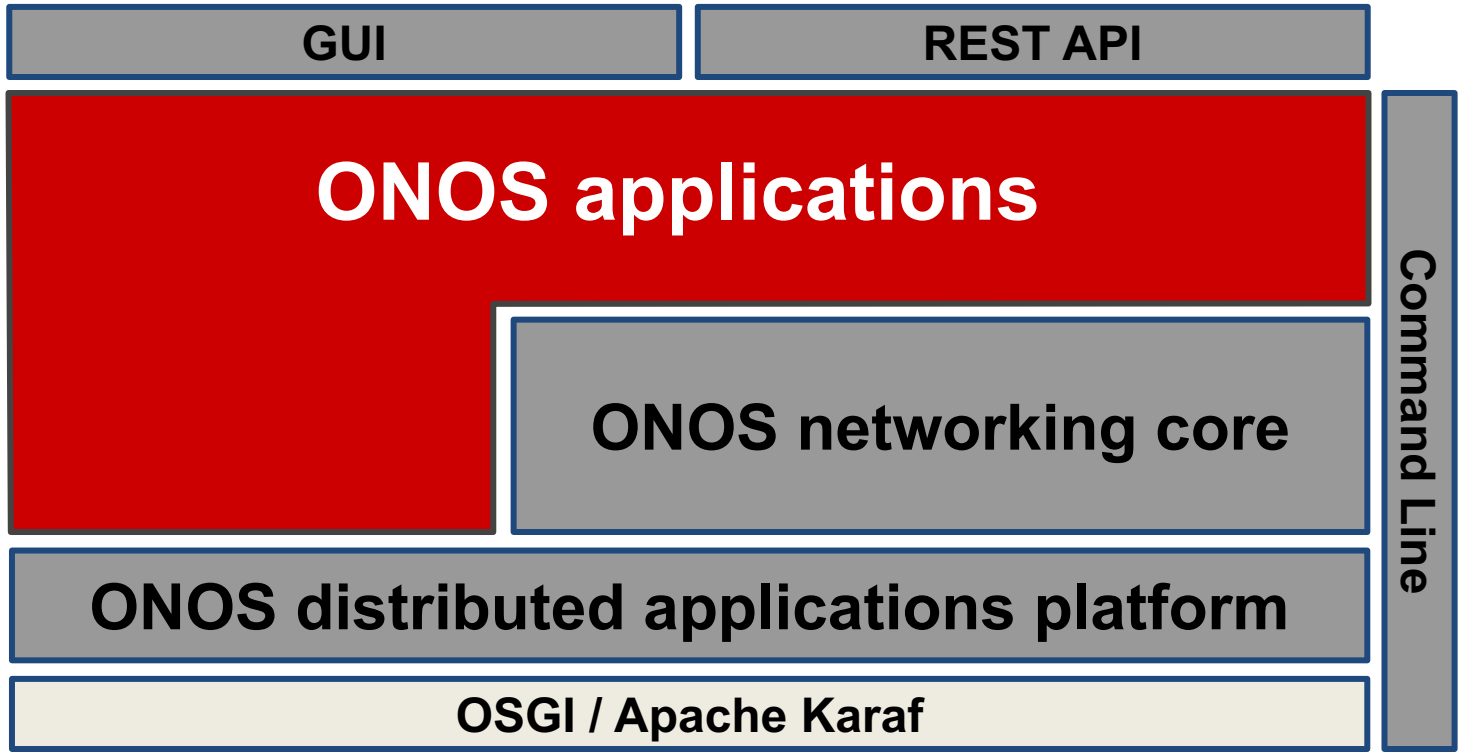
Development Contribution (1/5)

- Scratching Your Own Itch
  - For small fixes
    → submit a patch
  - For larger fixes or for new features
    → submit a proposal to the Technical Steering Team (TST)
    → submit a set of patches

- Helping Core Team with Roadmap
  - Get familiar with our processes by working on a starter bug
  - Start attending sprint planning and sprint DEMO meetings

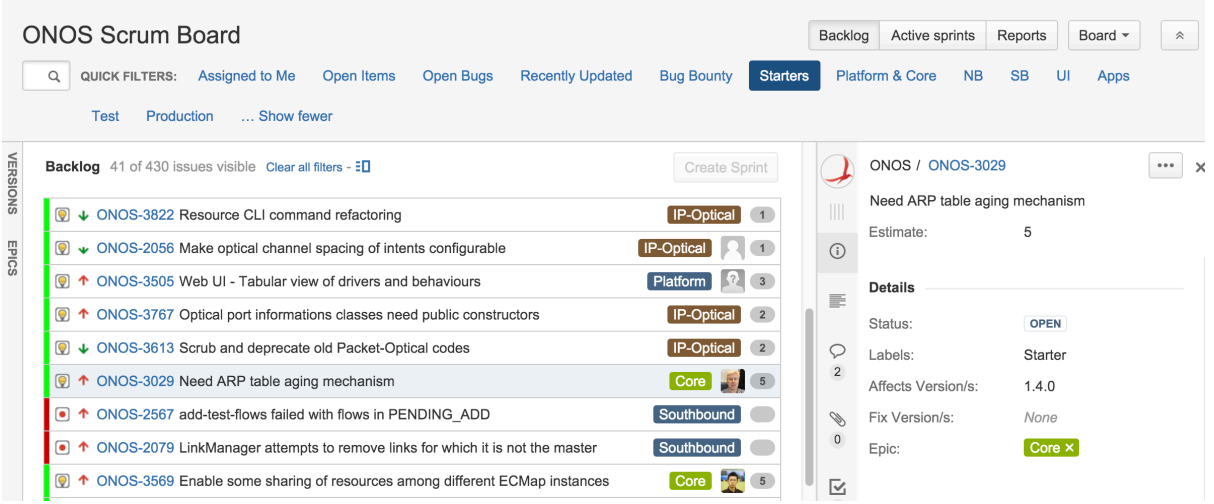
Development Contribution (2/5)

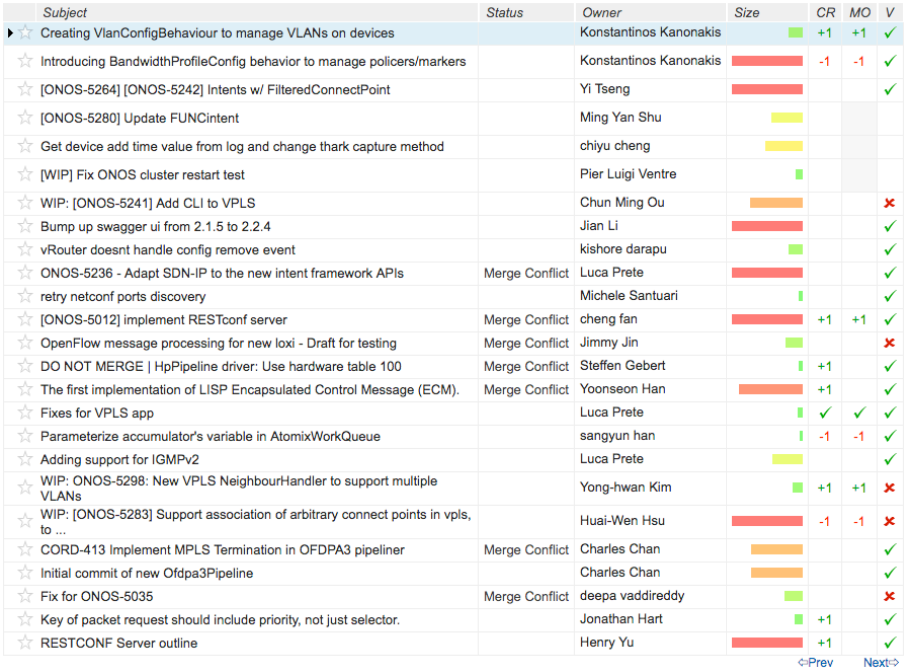
- Building Apps on top of ONOS
  - ONOS is an application platform that allows developers to dynamically extend the base capabilities



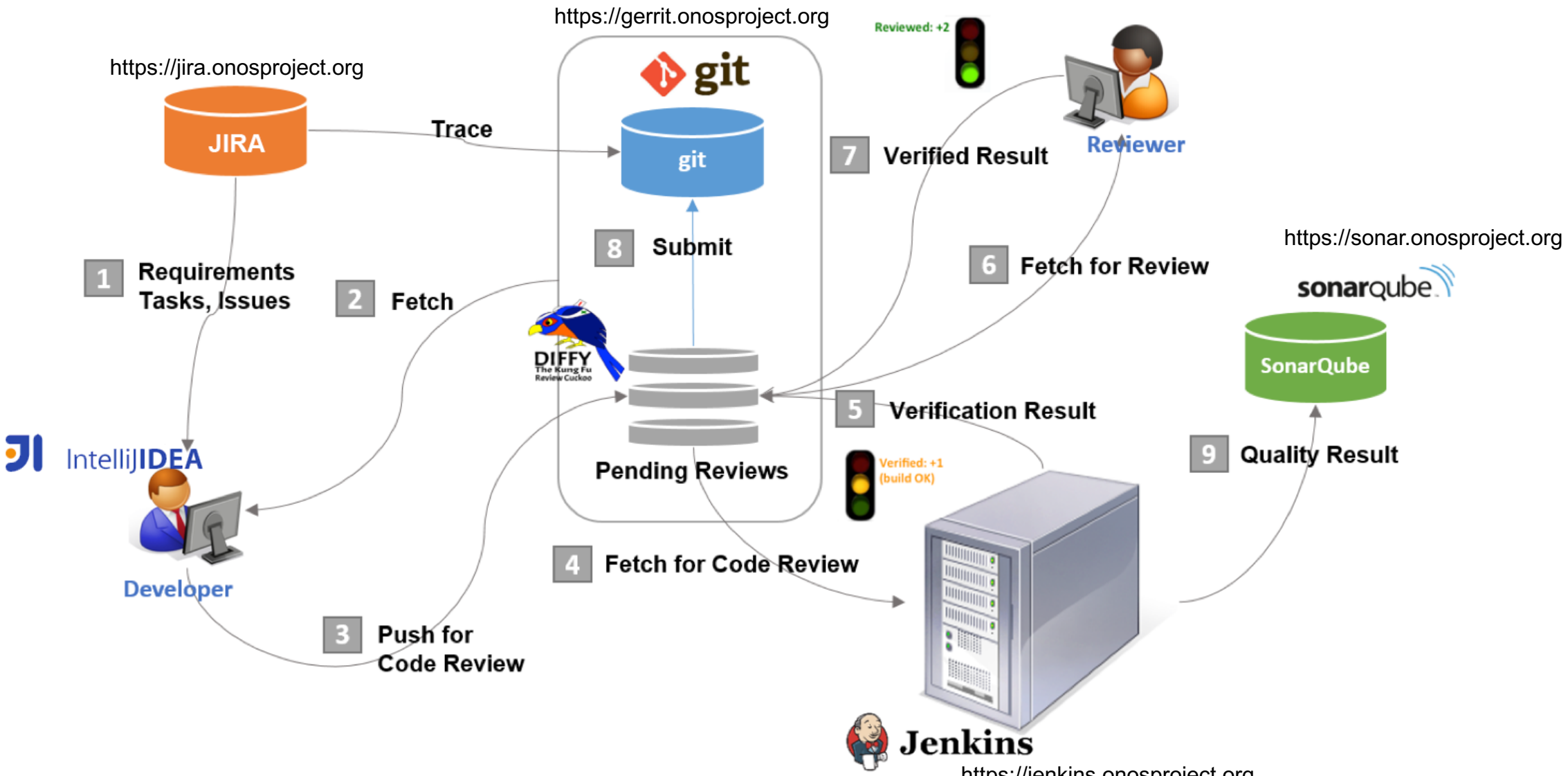
Development Contribution (3/5)

- Issue Tracking
  - https://jira.onosproject.org
  - Issue types
    - Epic, Story, Bug
  - Priority
    - Blocker, critical, minor
- Code Submission and Review
  - https://gerrit.onosproject.org
  - Roles
    - Contributor, reviewer, module owner, super module owner, project owner
  - Code submission
  - Code review
  - Scoring (-2, -1, 0, +1, +2)





Development Contribution (4/5)



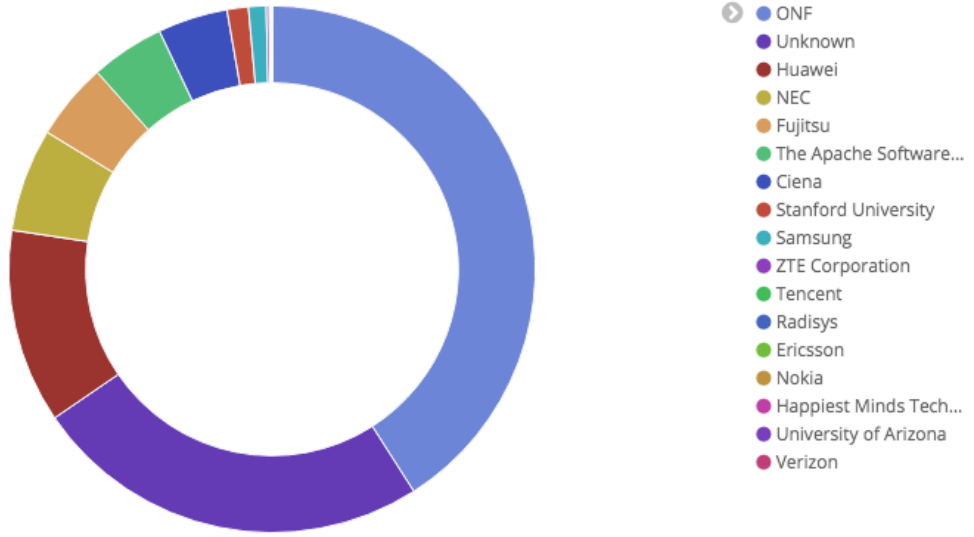
Development Contribution (5/5)

- ONOS Community Metrics – Last 2 Years

Top Authors

Author	Commits	Projects	Added Lines	Removed Lines	Avg. Files
HIGUCHI Yuta	620	1	45877	12456	4.984
Ray Milkey	587	1	65872	56684	65.206
Thomas Vachuska	484	1	131407	76207	9.11
Simon Hunt	455	1	55450	34340	6.207
Jian Li	444	1	149991	75644	6.919
Shimizu Sho	411	1	11566	13266	3.27
Jon Hall	410	1	41393	164569	5.127
Charles M.C. Chan	375	1	37500	20021	3.747
Brian O'Connor	345	1	50523	138089	91.878
Jonathan Hart	286	1	30669	18473	5.199

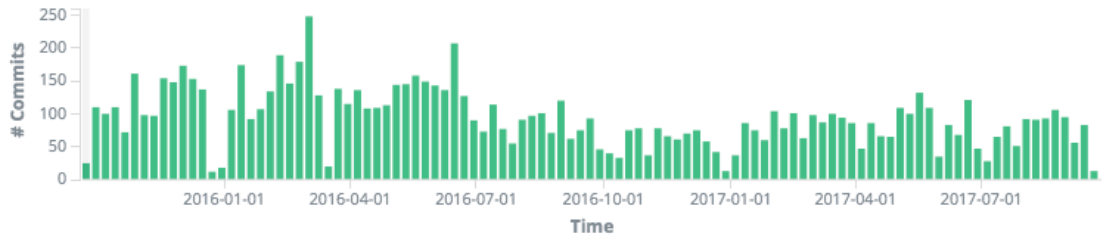
Organizations



Git

9,899	332	15
# Commits	# Authors	# Repositories

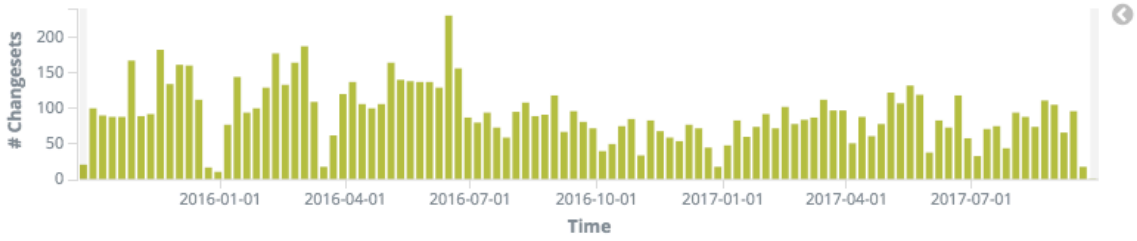
Commits



Gerrit

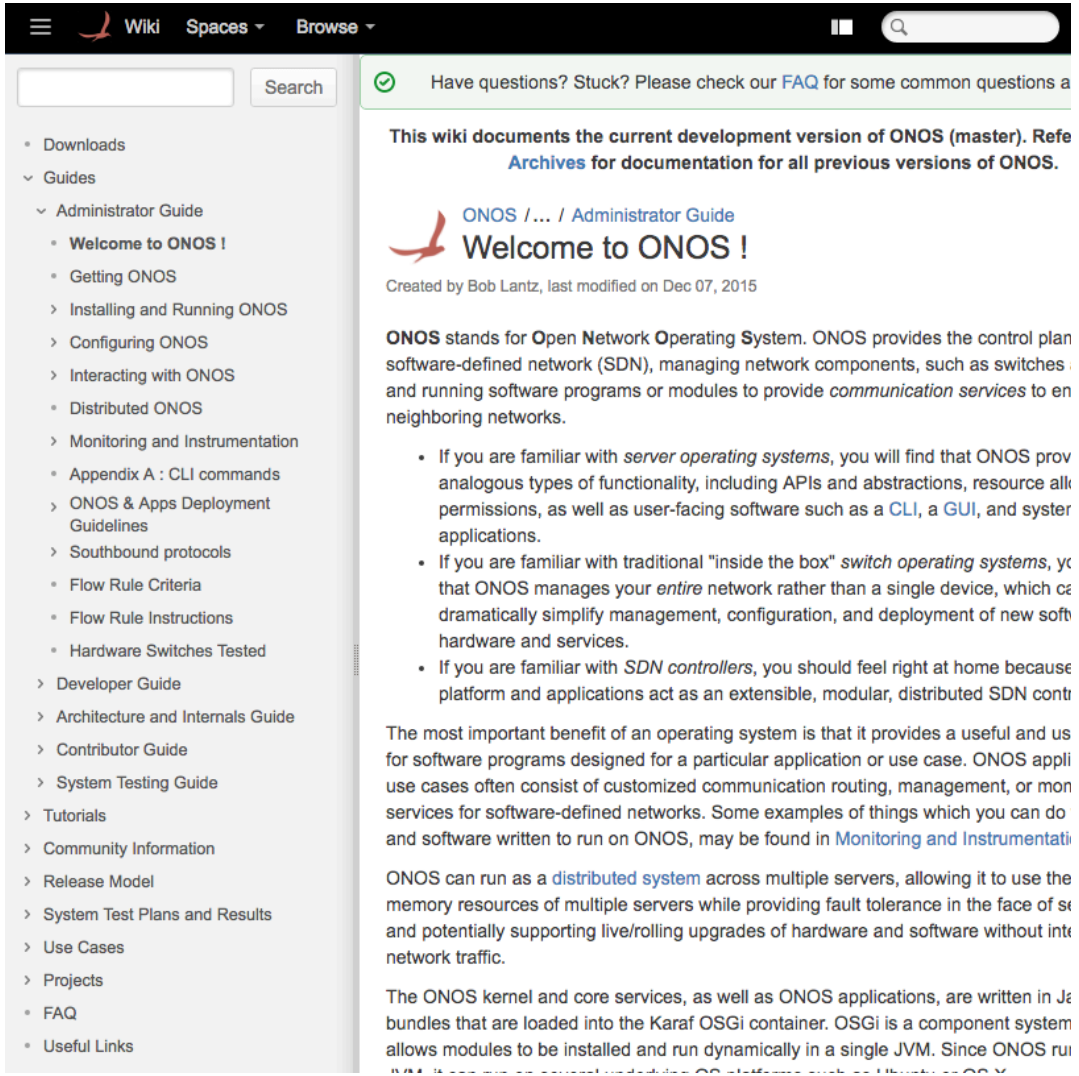
9,766	294	14
# Changesets	# Changeset Submitters	# Repositories

Changesets



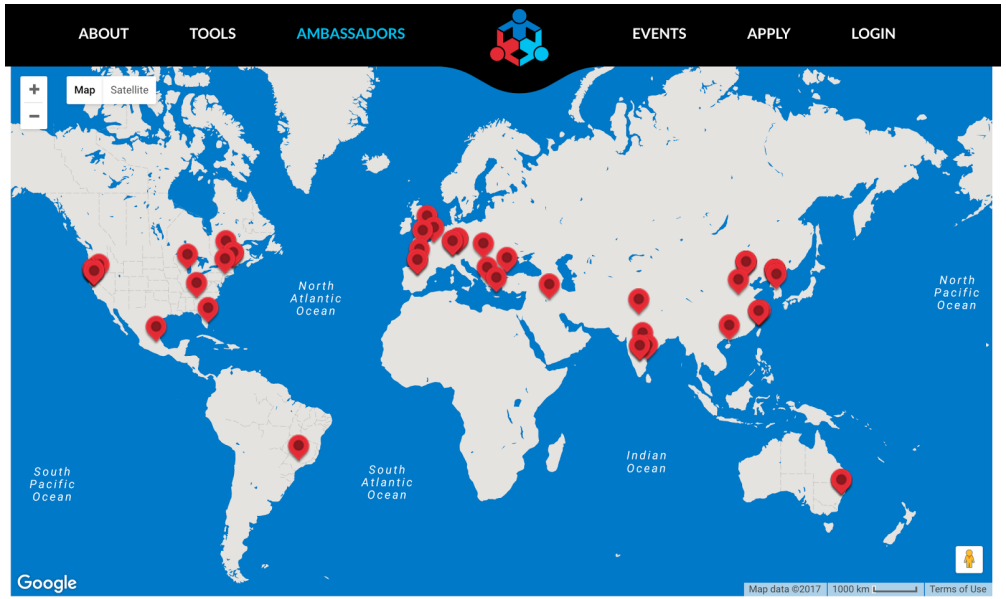
Documentation Contribution

- Documentation
  - https://wiki.onosproject.org
- Major Items
  - Tutorials, guides, use cases, projects
- Section Owners
  - Similar to *Module Owner* for ONOS codes
- Procedures
  - Check out JIRA to find any exist tickets
  - If exist, take ownership, proceed it!
  - If not, write up the detailed plan in JIRA and proceed documentation in wiki



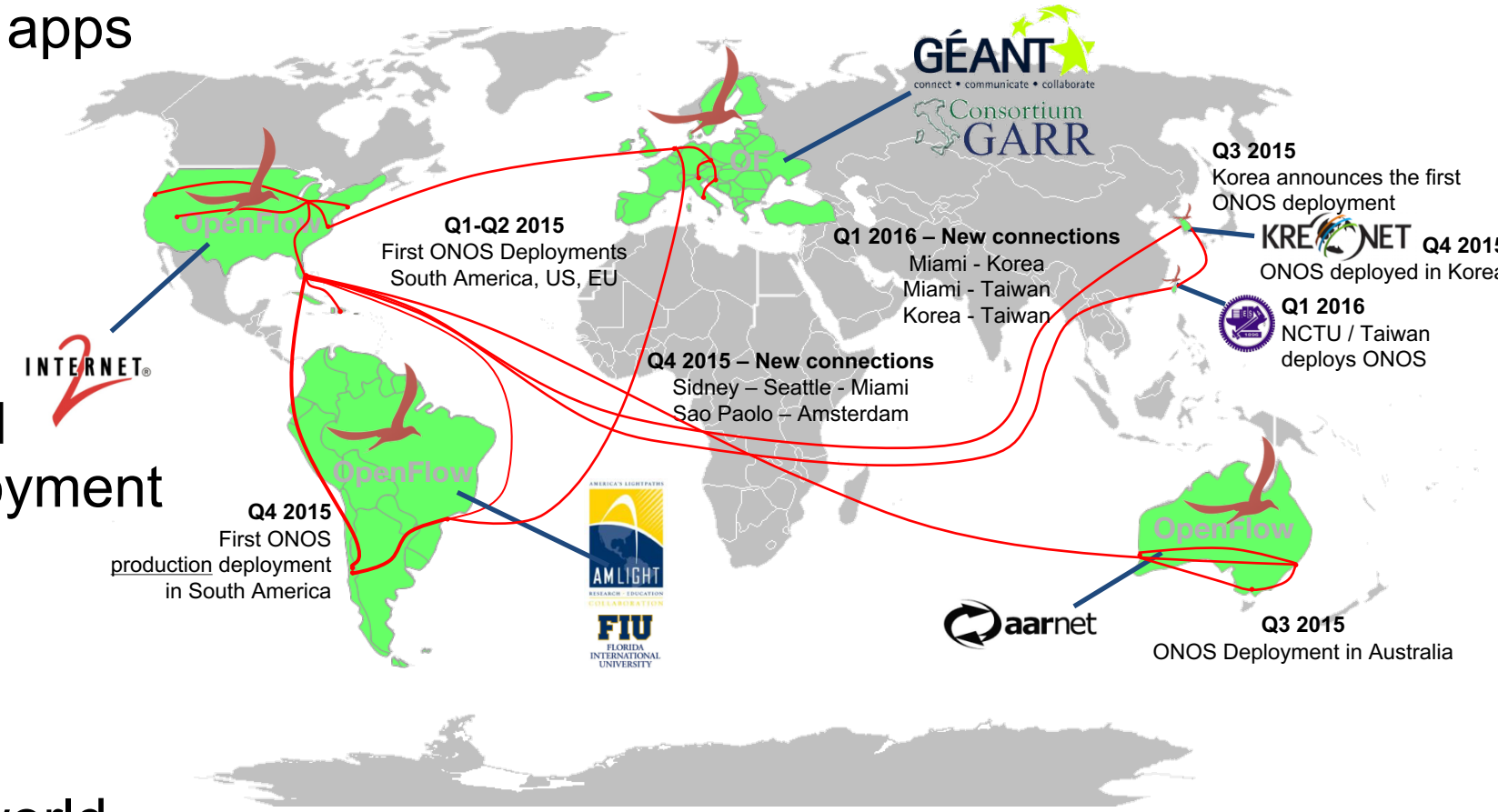
Spreading the Word Offline

- Share and subscribe (http://ambassadors.opennetworking.org/)
  - #ONOSProject, twitter, facebook, etc.
- Ambassador Program
  - Empower anyone who is passionate and knowledgeable about ONOS, who want to build a strong local community
  - Ways ONF supports Ambassadors
    - Provide guidance on organizing local events
    - Order and customize swag and business card
    - Provide relevant slides, templates, videos
    - Produce specific materials (posters, flyers, etc.)
  - Application steps
    1. Submit application form
    2. Interview with A-team member
    3. On-boarding!



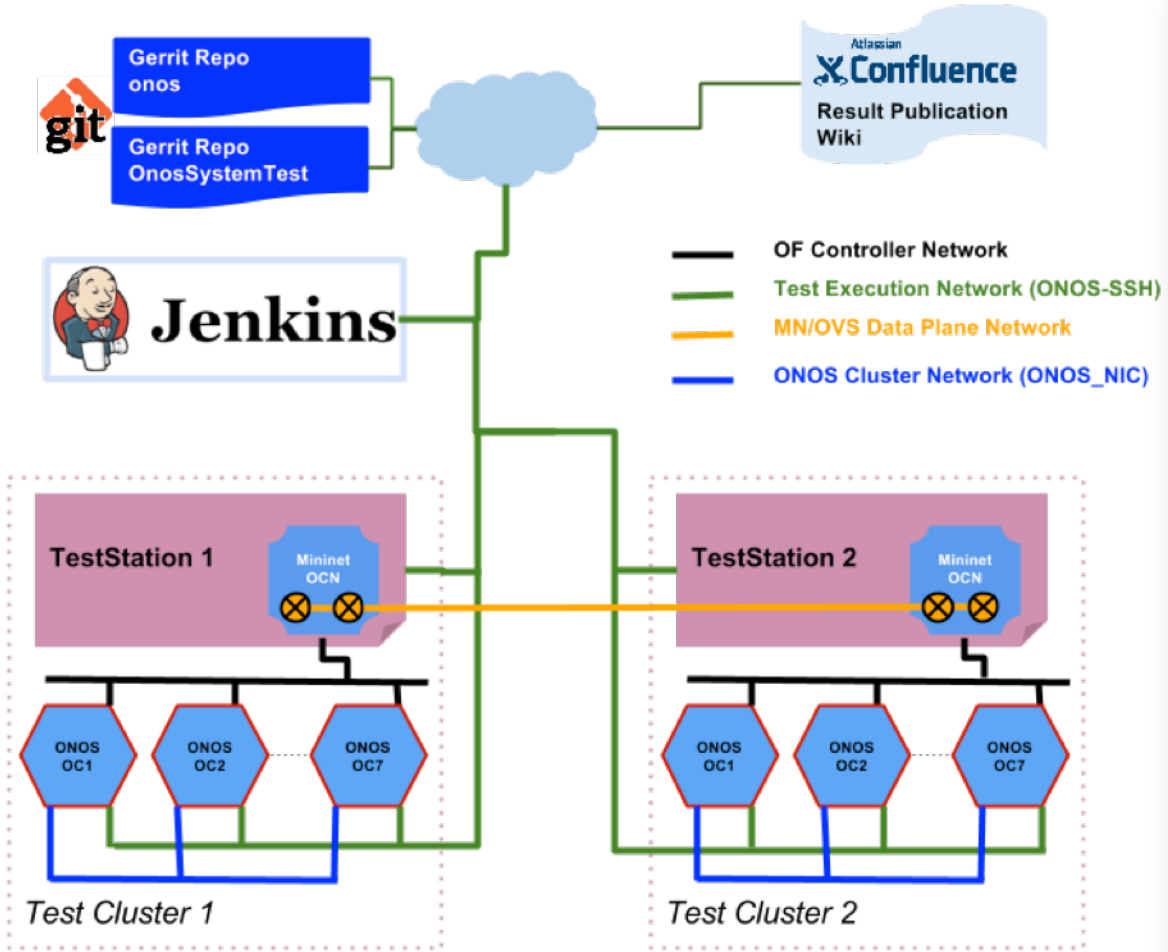
Deployment Contribution

- Expand
  - Deploy ONOS and its apps on your network
- Guide
  - ONF provides resources, guides and tips to help your deployment
- Share
  - ONF shares your deployment with the world

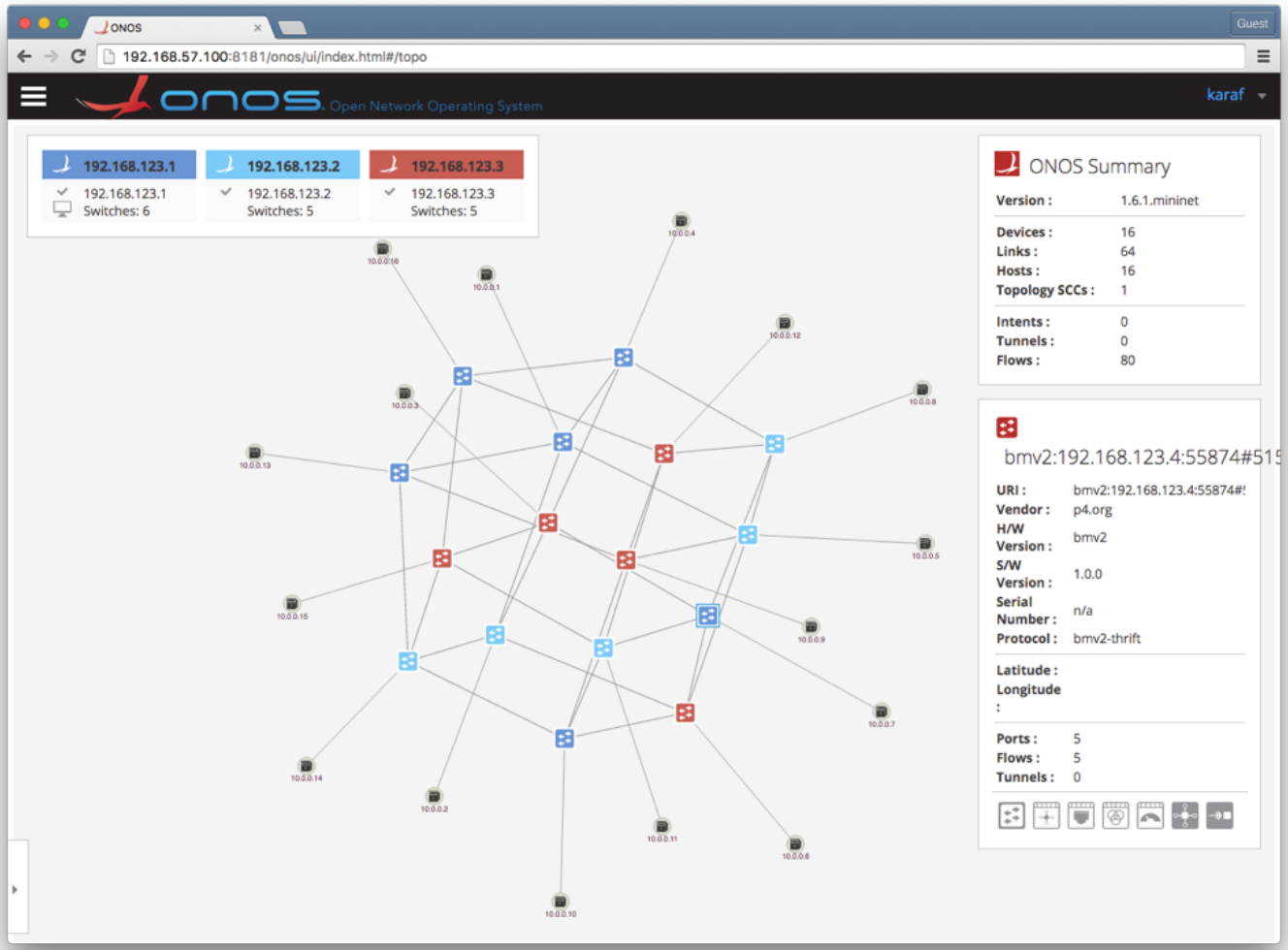


Other Contributions

- Quality Assurance Contribution

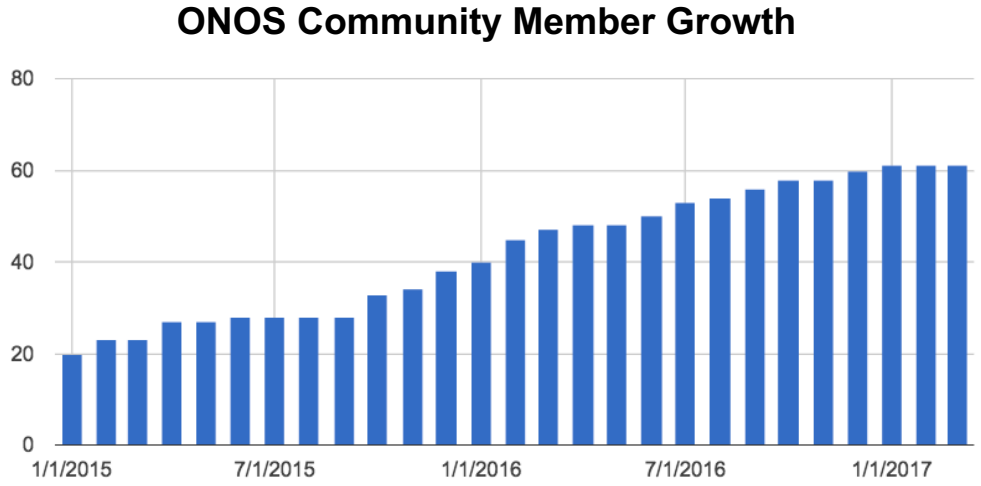


- UX & UI Contribution



Leverage the Power of Open Source Community (1/5)

- Brigade Model
  - Motivation
    - ONOS community continues to grow
    - Challenges of how to coordinate a large group to make sure we're all working toward a shared goal
  - Solution
    - Communicate clearly about ONOS vision
    - Invite people to work together on completing specific parts of the vision
  - Brigade Model
    - Create small teams around specific features that core team want to ship in upcoming version of ONOS



Leverage the Power of Open Source Community (2/5)

- Benefits of Joining a Brigade
  - Opportunity
    - Unique opportunity to work with the core engineering team
    - Participate in work onsite at Menlo Park
  - Recognition
    - Showcased widely with the community both online as well as at events
  - Experience
    - Get experience in network engineering
    - A great stepping stone to possibly work at ONF or other member organizations
  - Acceleration
    - Get work that you care about into an official ONOS release much more quickly
  - Funding
    - ONF provides budget for teams to work with the core engineering team

Leverage the Power of Open Source Community (3/5)

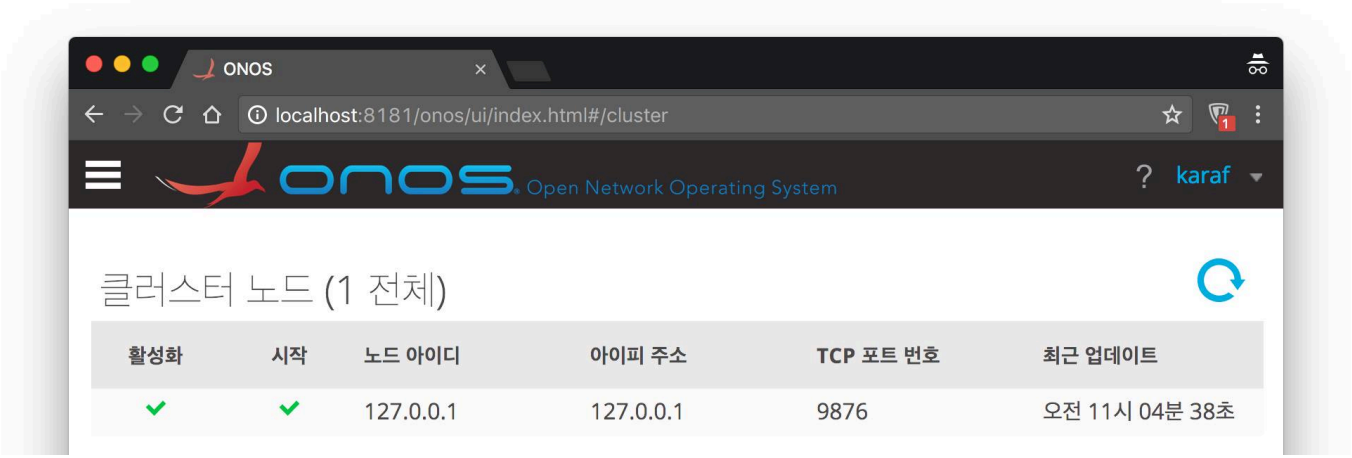
- ONOS Brigades in 2017
  - Intent subsystem 2.0
    - Offers Composable Network-centric Primitives
    - Joint organizations: Fujitsu, ONF
  - SDN/ONOS training
    - Provide and re-organize open source teaching materials
    - Joint organizations
      - DTU, Politecnico di Milano, UPMC, Universita di Pisa, Verizon, Strategic Virtualization, ONF, NCTU, Politecnico di Torino, etc.
  - Build and package infrastructure
    - Tools and processes for building ONOS and publishing the artifacts
    - Joint organizations
      - Verizon, Gigamon, etc.

Leverage the Power of Open Source Community (4/5)

- ONOS Brigades in 2017
  - gRPC northbound API
    - Allow high-performance interactions with off-platform applications
    - Joint organizations
      - ONF, ZTE, Inspur, etc.
  - Security and performance analysis
    - Assess controller robustness against network and system attacks
    - Compare ONOS controller to other equivalent controllers
    - Joint organizations
      - Orange Labs, Politecnico di Milano, UPMC, Nokia Bell Labs, ONF, QUB, CIT

Leverage the Power of Open Source Community (5/5)

- ONOS Brigades in 2017
  - P4
    - Support awareness of P4 programs including ability to deploy them
    - Joint organizations
      - ONF, ZTE, Inspur, POSTECH, UTS
  - localization (I10n) → LION
    - Develop a framework for localization of the GUI and produce a set of localized message bundles



ONOS Community is Growing

ONOS Build 2017
Samsung R&D Campus
Sept., 20-22

ONOS

Open Network Operating System

http://onosproject.org/

APNOMS 2017