Sharif University of Technology

Computer Engineering Department

Software-Defined Networking

Ali Movaghar

Mohammad Hosseini

Tools

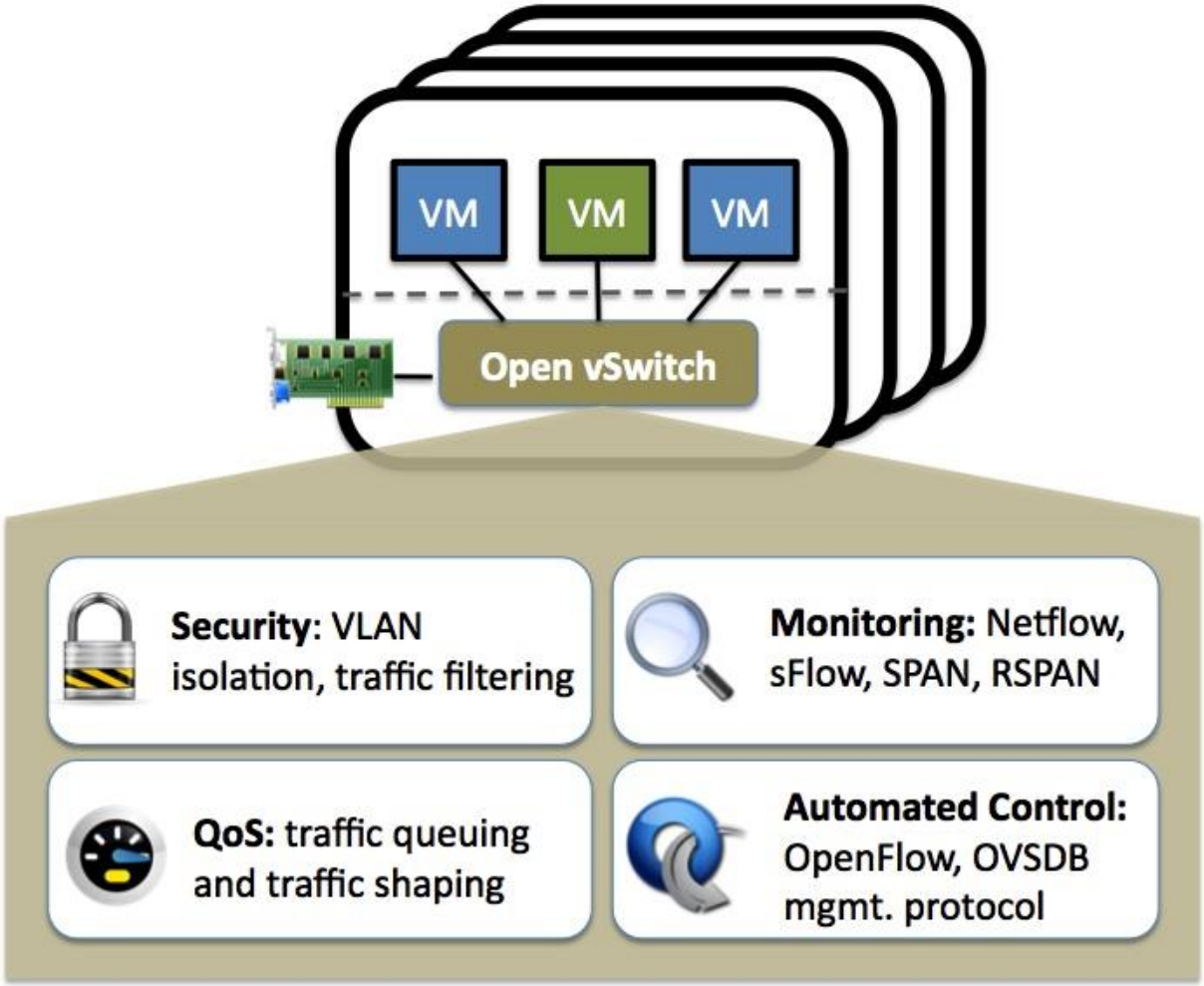
TA: Iman Rahmati & Farbod Shahinfar

What do we need to carry out OpenFlow-based SDN experiments?

- OpenFlow switch
- SDN controller (Network Operating System)
- A network of switches and hosts

Software switch

❖ **Open vSwitch** is a production quality, open source, multilayer virtual switch.



A network of emulated hosts and switches

A Network in a Laptop: Rapid Prototyping for Software-Defined Networks

Bob Lantz
Network Innovations Lab
DOCOMO USA Labs
Palo Alto, CA, USA
rlantz@cs.stanford.edu

Brandon Heller
Dept. of Computer Science,
Stanford University
Stanford, CA, USA
brandonh@stanford.edu

Nick McKeown
Dept. of Electrical Engineering
and Computer Science,
Stanford University
Stanford, CA, USA
nickm@stanford.edu

ABSTRACT

Mininet is a system for rapidly prototyping large networks on the constrained resources of a single laptop. The lightweight approach of using OS-level virtualization features, including processes and network namespaces, allows it to scale to hundreds of nodes. Experiences with our initial implementation suggest that the ability to run, poke, and debug in real time represents a qualitative change in workflow. We share supporting case studies culled from over 100 users, at 18 institutions, who have developed Software-Defined Networks (SDN). Ultimately, we think the greatest value of Mininet will be supporting collaborative network research, by enabling self-contained SDN prototypes which anyone with a PC can download, run, evaluate, ex-

1. INTRODUCTION

Inspiration hits late one night and you arrive at a world-changing idea: a new network architecture, address scheme, mobility protocol, or a feature to add to a router. With a paper deadline approaching, you have a laptop and three months. What prototyping environment should you use to evaluate your idea? With this question in mind, we set out to create a *prototyping workflow* with the following attributes:

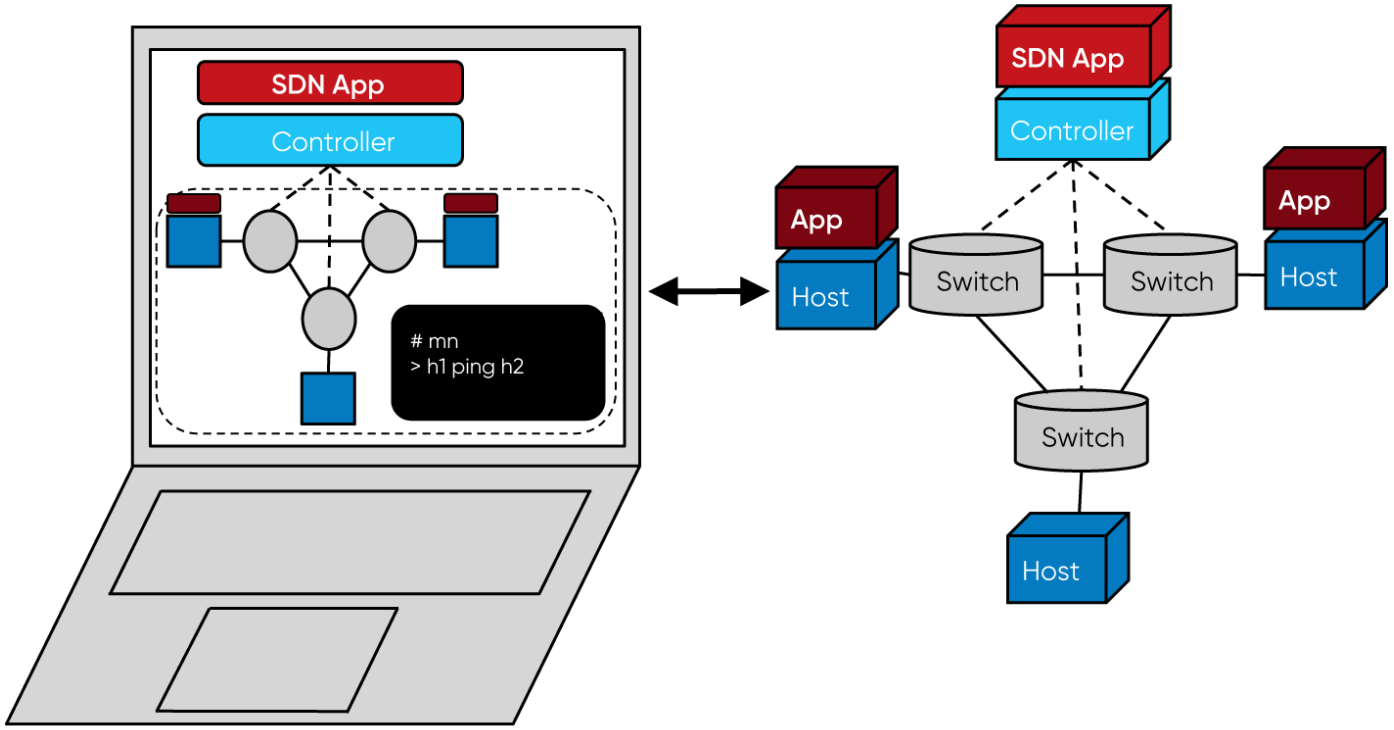
Flexible: new topologies and new functionality should be defined in software, using familiar languages and operating systems.

Deployable: deploying a functionally correct pro-

Mininet

❖ Mininet is a network emulator which creates a network of virtual hosts, switches, and links.

- ➢ Rapid prototyping of software-defined networks
- ➢ Complex topology testing
- ➢ Mininet networks run real code including standard Linux network applications as well as the real Linux kernel and network stack.
- ➢ Open source
- ➢ Python



Mininet

- **Isolated hosts**: A group of user-level processes moved into a network namespace that provide exclusive ownership of interfaces, ports and routing tables.

- **Emulated switches**: The default Linux Bridge or the Open vSwitch is used to switch packets across interfaces.

- **Emulated links**: Linux Traffic Control (tc) enforces the data rate of each link to shape traffic to a configured rate. Each emulated host has its own virtual Ethernet interface(s).

Mininet forks

- **Mininet-WiFi:** extends the functionality of Mininet by adding virtualized WiFi Stations and Access Points.
  - New classes has been added in order to support the addition of these wireless devices in a Mininet network scenario and to emulate the attributes of a mobile station such as position and movement relative to the access points.

- **Maxinet:** extends the Mininet emulation environment to span the emulation across several physical machines.
  - This allows to emulate very large software-defined networks.

Open Source SDN Controllers
(Network Operating System)

- NOX
- POX
- Beacon

- FloodLight
- RYU
- OpenDayLight
- ONOS

Controller Considerations

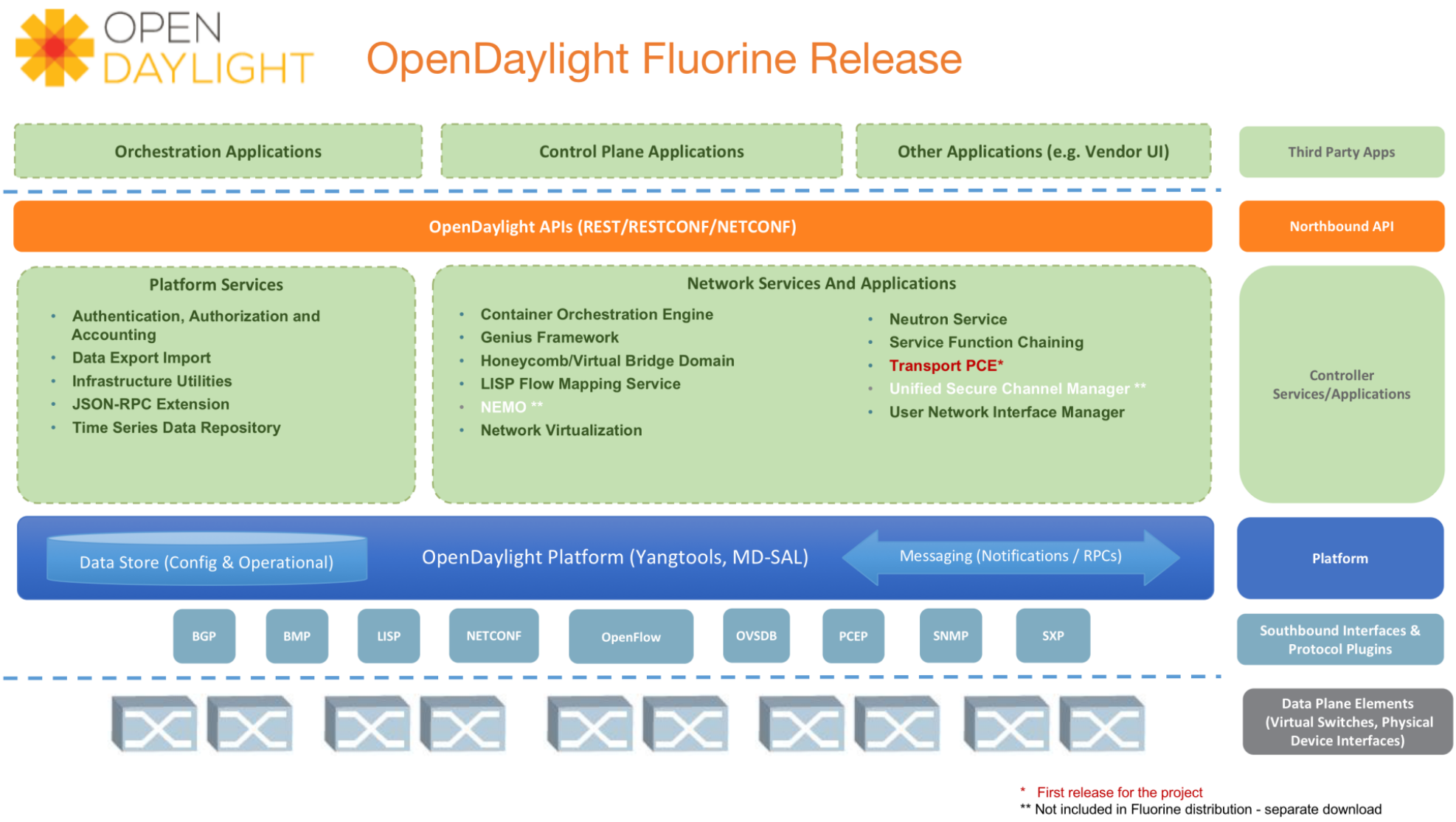
- Programming language (can affect performance)
- Learning curve
- User base and community support
- Focus in terms of:
  - Southbound API
  - Northbound API
  - Support for cloud platforms such as OpenStack
  - Education, Research, Production?

OpenDayLight (ODL)

- Heavy industry involvement and backing
  - Hosted by *Linux Foundation* and founded by *Big Switch Networks*, *Cisco*, *IBM*, *Juniper Networks*, *Microsoft*, *Red Hat*, *VMware, and etc*.

- Focused on having an open framework for building upon SDN/NFV innovations
  - not limited to OpenFlow innovations (supports many southbound APIs)

OpenDayLight (ODL)



* First release for the project
** Not included in Fluorine distribution - separate download

OpenDayLight (ODL)

- Written in Java
- Widespread industry acceptance
- Production-level performance and support

- Modular design based on OSGi.
  - Modules (bundles) can be dynamically loaded at runtime.
  - Allows registering dependencies and services exported
  - Exchanging information across bundles

- Southbound protocols: OpenFlow, P4, NETCONF, RESTCONF, SNMP, BGP, PCEP

- Northbound: REST, NETCONF, gRPC

OpenDayLight (ODL)

- Service Abstraction Layer (SAL) figures out how to fulfill the requested services irrespective of the underlying protocol used between the controller and the network devices.
- ODL contains internal functionality for maintaining a cluster
- ODL can be integrated with OpenStack cloud computing platform.
- Poor documentation

ONOS (Open Network Operating System)

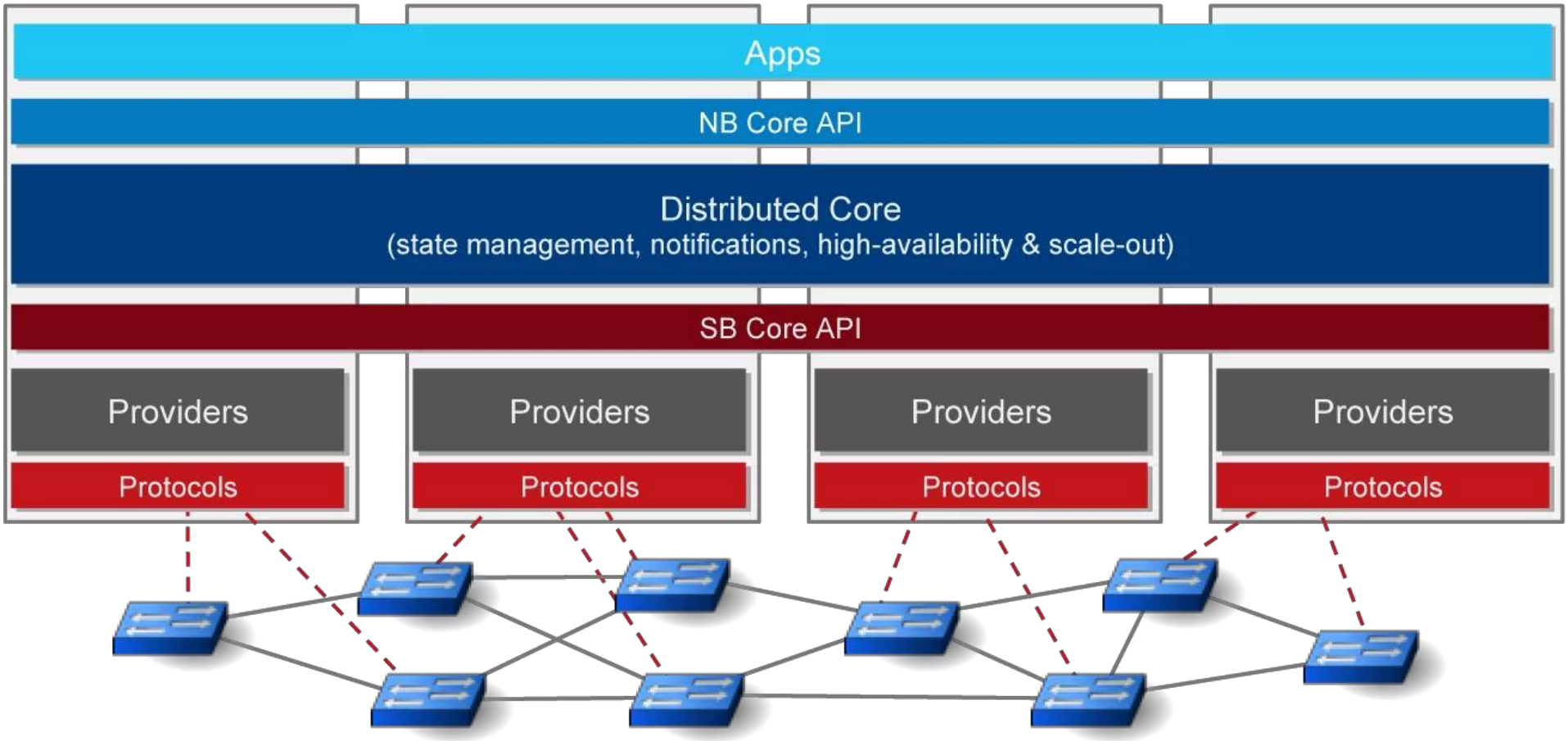
- Hosted by *Open Networking Foundation* and founded by *Cisco*, *Google*, *AT&T*, *Huawei*, etc.
- An SDN operating system for *communications service providers* that is designed for scalability, high performance and high availability.

- Written in Java
- Good documentation
- Modular design based on OSGi.

- Southbound protocols: OpenFlow, P4, NETCONF, RESTCONF, SNMP, BGP, PCEP
- Northbound: REST, NETCONF, gRPC

ONOS (Open Network Operating System)

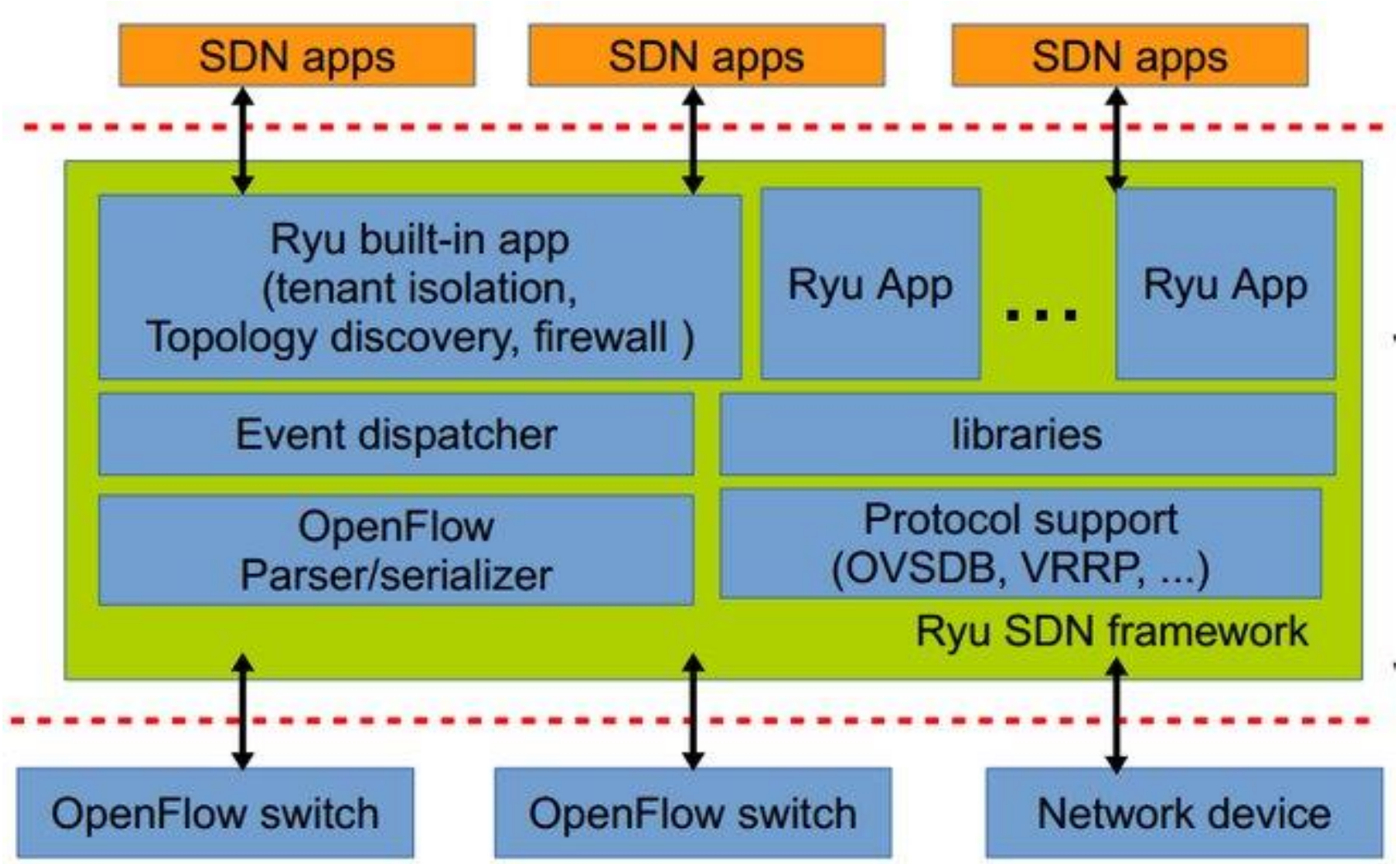
- ONOS is designed specifically to horizontally scale for performance and geo-redundancy. The cluster configuration is simple, with new controllers being able to join and leave dynamically. Each controller instance monitors and controls a subset of the physical switches in the network.



RYU

- Written in Python
- Component-based
- Southbound protocols: OpenFlow (1.0, 1.2, 1.3, 1.4, 1.5), NETCONF, OVSDB, OF-Config
- Northbound: RESTful APIs
- Easy to learn
- OpenStack integration
- Ryu does not have an inherent clustering ability.

RYU


SDN apps
SDN apps
SDN apps
Ryu built-in app
(tenant isolation,
Topology discovery, firewall)
Ryu App
...
Ryu App
Event dispatcher
libraries
OpenFlow
Parser/serializer
Protocol support
(OVSDB, VRRP, ...)
Ryu SDN framework
OpenFlow switch
OpenFlow switch
Network device

Installing Mininet

```
$ sudo apt update
$ sudo apt install mininet
```

- Also, automatically installs Open vSwitch

Installing RYU

```
$ sudo apt install git python3-setuptools
```

```
$ git clone https://github.com/osrg/ryu.git
$ sudo python3 ./setup.py install
```

Try running RYU:

```
$ ryu-manager ryu.app.simple_switch_13
```

Install the reported missing python packages and rerun RYU until it starts successfully. For example:

```
$ sudo apt install python3-webob
```

For some packages, a specific version may be needed. They can be installed by the "pip" command.

```
$ sudo apt install python3-pip
$ sudo pip instal tinyrpc==1.0.4
```