NTT Software Innovation Center

NTT

# Introduction to Ryu SDN framework


FUJITA Tomonori
NTT

# Philosophy

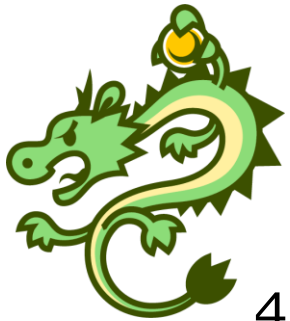
- **Agile**
  - Framework for SDN application development instead of all-purpose big monolithic ‘controller’.
- **Flexible**
  - Vendor-defined “Northbound” APIs are not enough to differentiate.

# Ryu: Component-based framework

- **Your application consists of component(s)**
  - Ryu provides a bunch of components useful for SDN applications.
  - You can modify the existing components and implement your new components.
  - Combines the components to build your application.

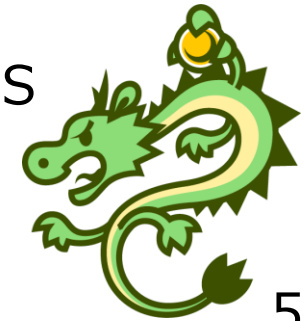
# What's 'component'?

- **Component is separation unit**
  - Provides interface for control and state and generates events.
  - Communicates by message passing instead of directly referenced.
  - Uses language-independent communication (for now uses python-specific messaging but soon move to such as JSON-RPC).

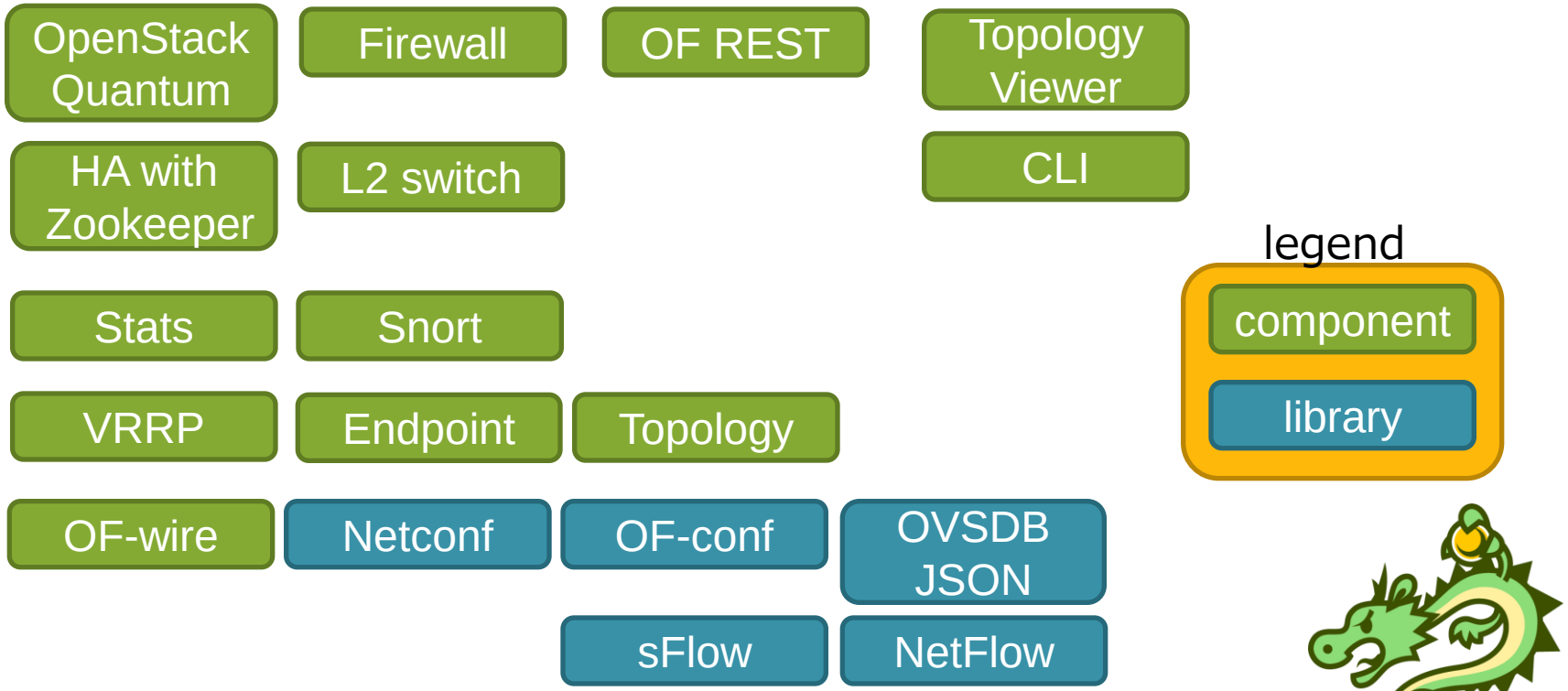


# Component implementation

- **Use favorite language**
  - Your component can work together with the existing components with Ryu's 'standard' messaging way.
  - You can run the existing software (such as routing protocol daemons) as Ryu component with some modification.
- **Components included in Ryu**
  - Implemented in Python.
  - A component consists of python thread(s) or OS process(s).



# Components and libraries included in Ryu



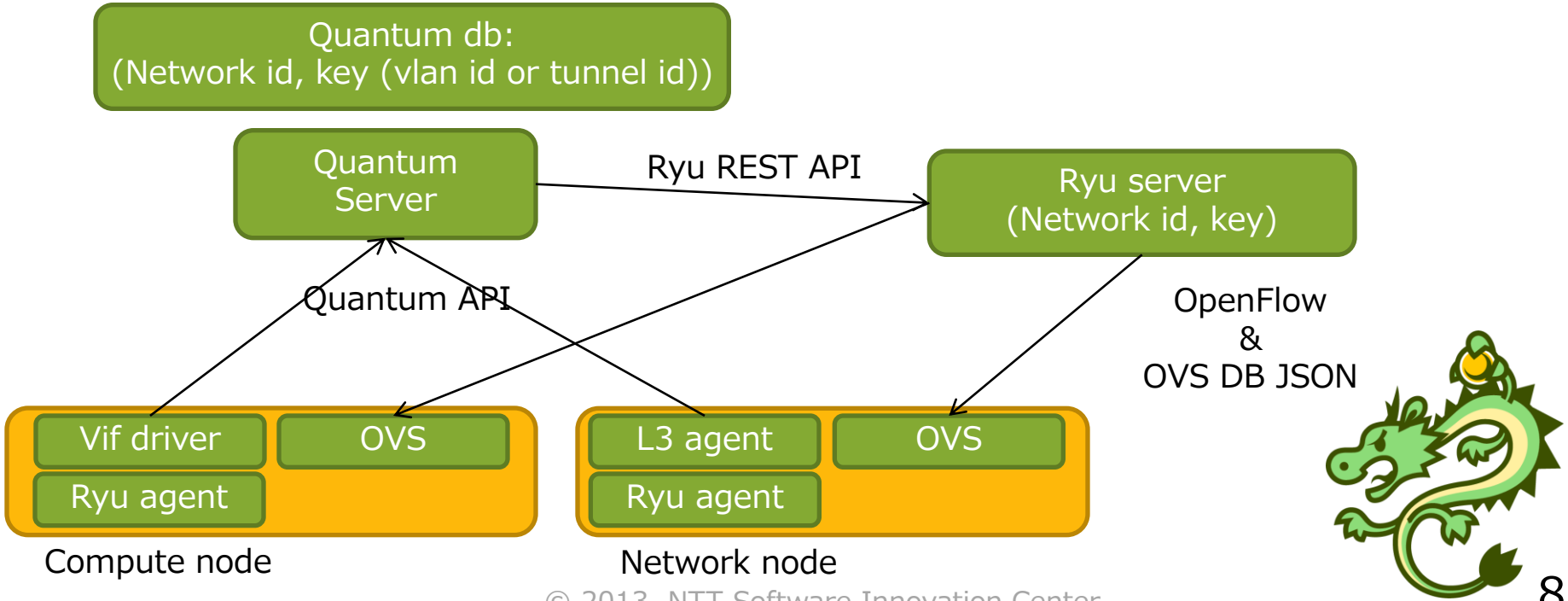

Libraries consist of functions called directly by components.

# Component description

- **OF-wire**
  - OpenFlow 1.0, 1.2, 1.3 and Nicira extensions.
- **Topology**
  - Builds topology and tracks links.
  - Path calculation feature will be supported soon.
- **VRRP**
  - Adds Virtual Router Redundancy Protocol (v3) support to OFS.
- **OF REST**
  - You can configure OF switches via REST APIs.

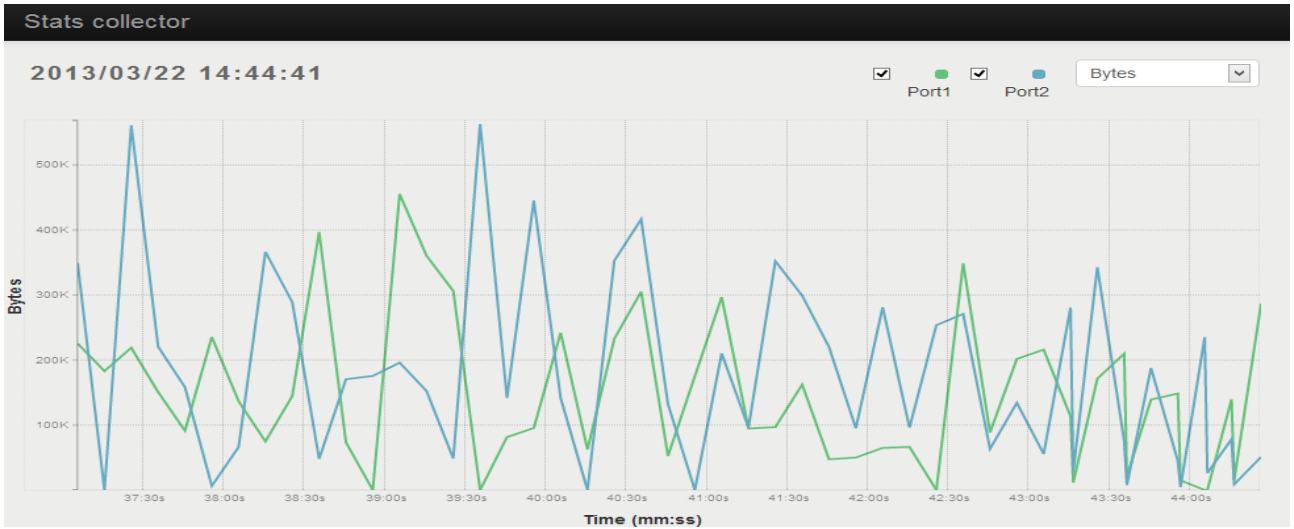
# OpenStack Quantum

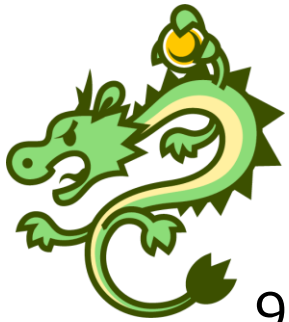
- **Support two ways to isolate tenants**
  - GRE tunnel and VLAN



# Stats component

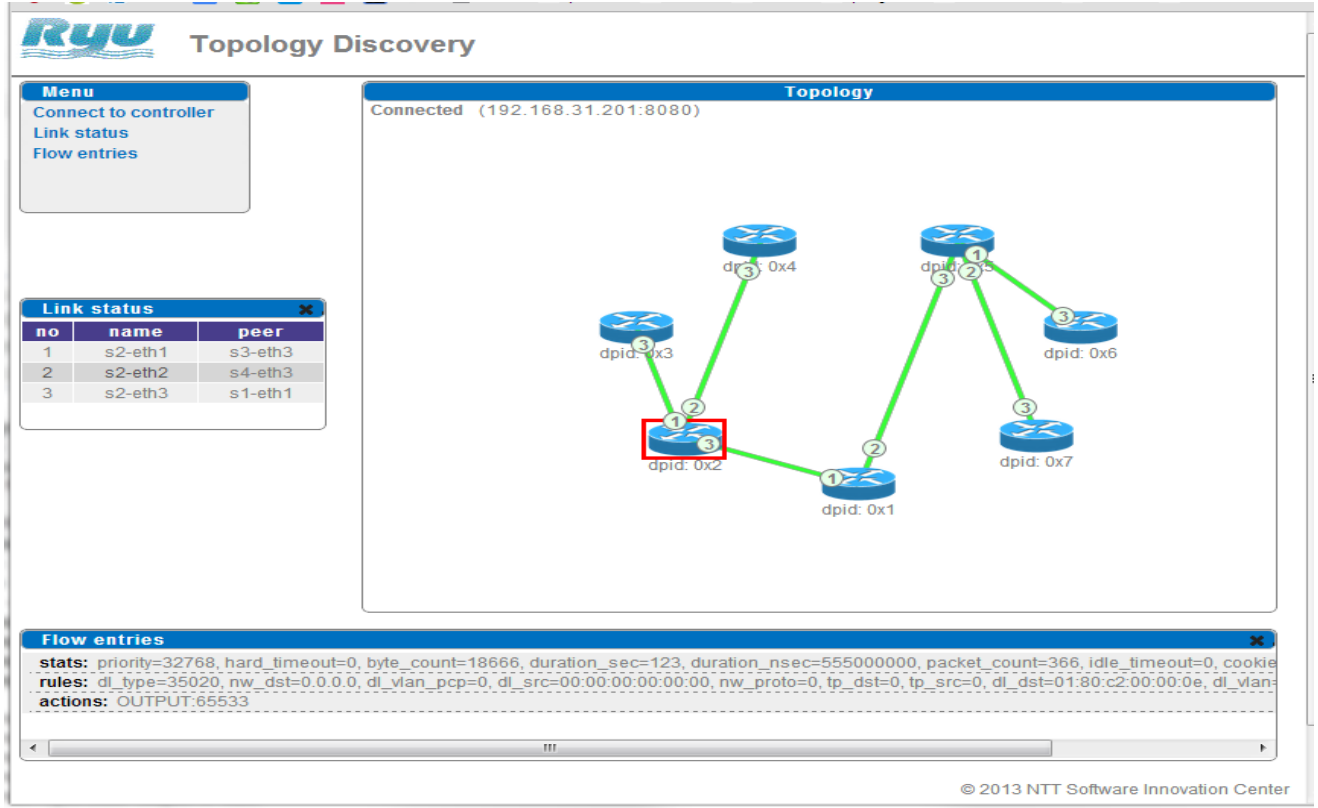
- **Stores stats to data store**
  - visualizes and analyzes.
  - The prototype stores switch stats to Hbase.



# Topology viewer

- Show topology and flows dynamically




# Future work

- **Make SDN development more agile**
  - Adds more components (protocols, IaaS, stats, security, etc).
  - Introducing network abstraction model (hide southbound difference, etc).
  - Improves distributed deployment component (cluster support).
  - New testing methods (Ryu has more than 15,000 lines test code).

# Ryu is an ongoing project

- **Ryu project needs more developers**
  - NTT team wants to make Ryu usable for many organizations.
  - The development is truly open and Ryu already has some code from non NTT developers.
  - NTT team would like to help you to use Ryu in production.

NTT Software Innovation Center

NTT

# Thanks!


Looking forward to your participation

http://osrg.github.com/ryu/

# Python Performance?

- **You need scalability probably**
  - Language runtime efficiency can’t solve scalability problem
  - Scalability about the whole system architecture.
- **Still need to improve runtime efficiency**
  - Pypy: anothr python runtime using JIT.
  - Using C for such components.