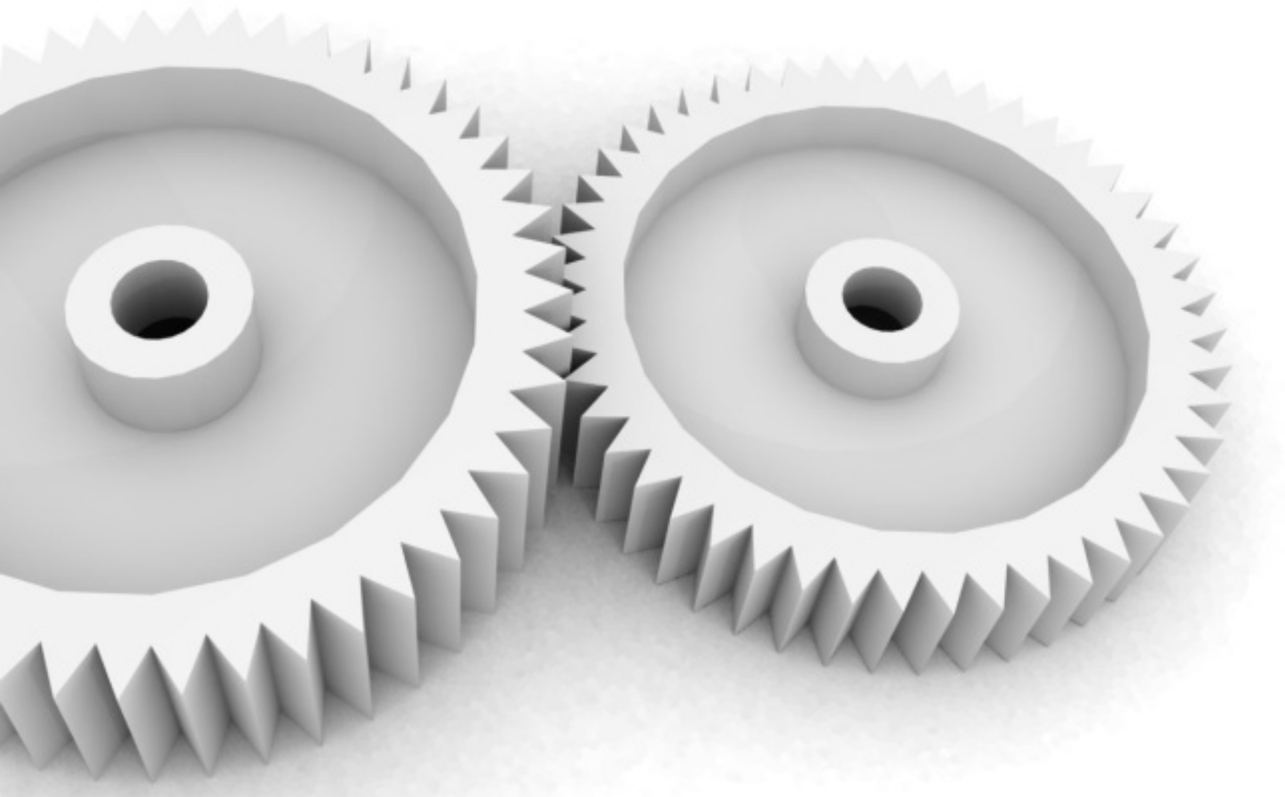
FloodLight

OpenFlow Controller

http://floodlight.openflowhub.org



Floodlight

Floodlight Controller

A great platform for OpenFlow

FloodLight

Apache

Research and commercial friendly ☺

Java

Jython

Easy to build, run, and develop

Toolchain

Rich set of build and debugging tools

big switch networks

OpenFlowHub BROADCASTING OPENFLOW

Community of OpenFlow experts, access to commercial upgrades, and frequent testing

Building Floodlight

Fast…and easy…

Download from Github

```
$ git clone git://github.com/floodlight/floodlight.git

$ sudo apt-get install build-essential default-jdk ant python-dev

$ cd floodlight; ant

$ java –jar target/floodlight.jar
```

Get the VM (including mininet)

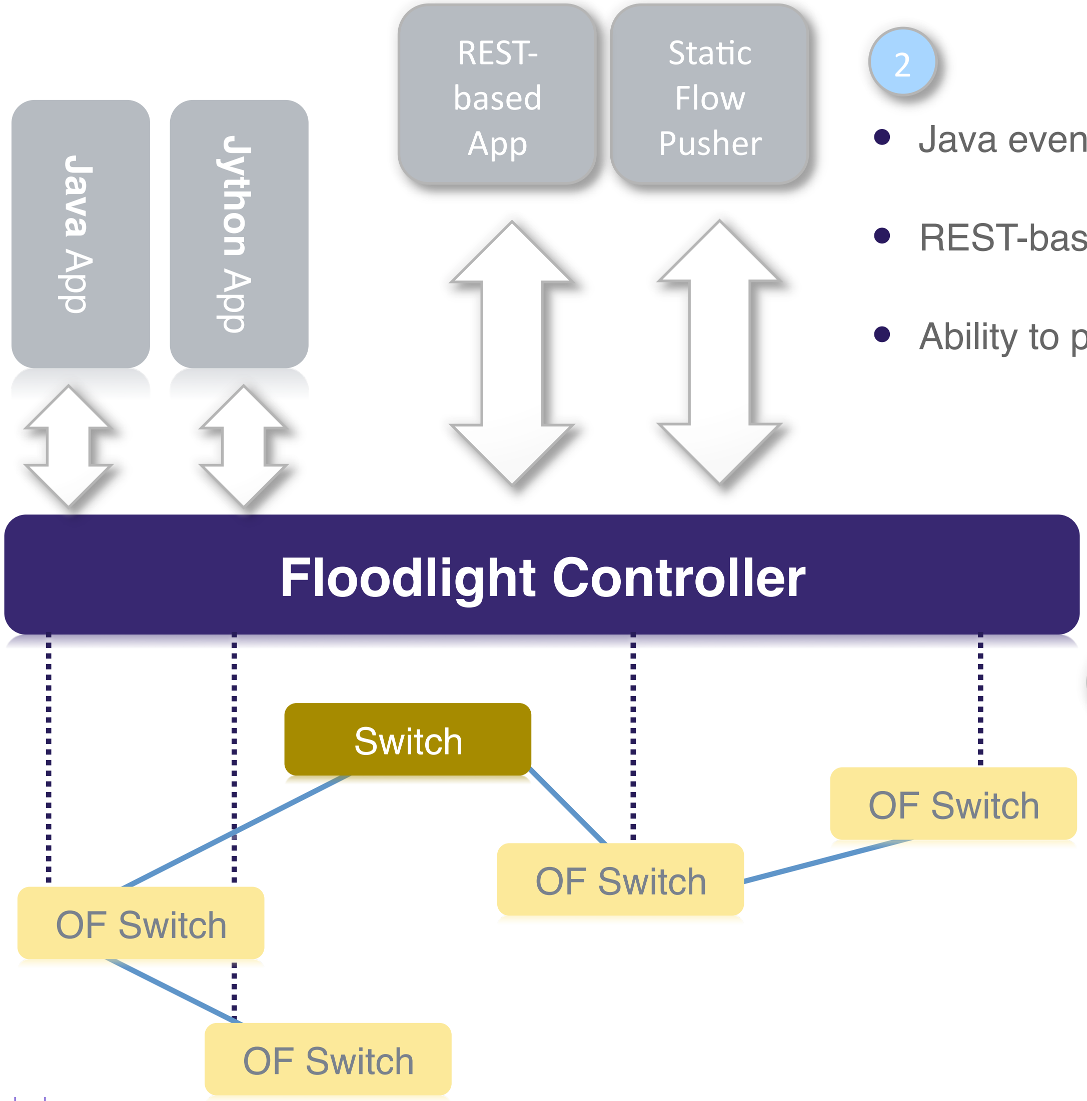
```
$ wget http://floodlight.openflowhub.org/files/floodlight-vm.zip

(login as “floodlight” user, no password)
```

Other Floodlight Highlights

1

- Active work in defining standard “Northbound” APIs

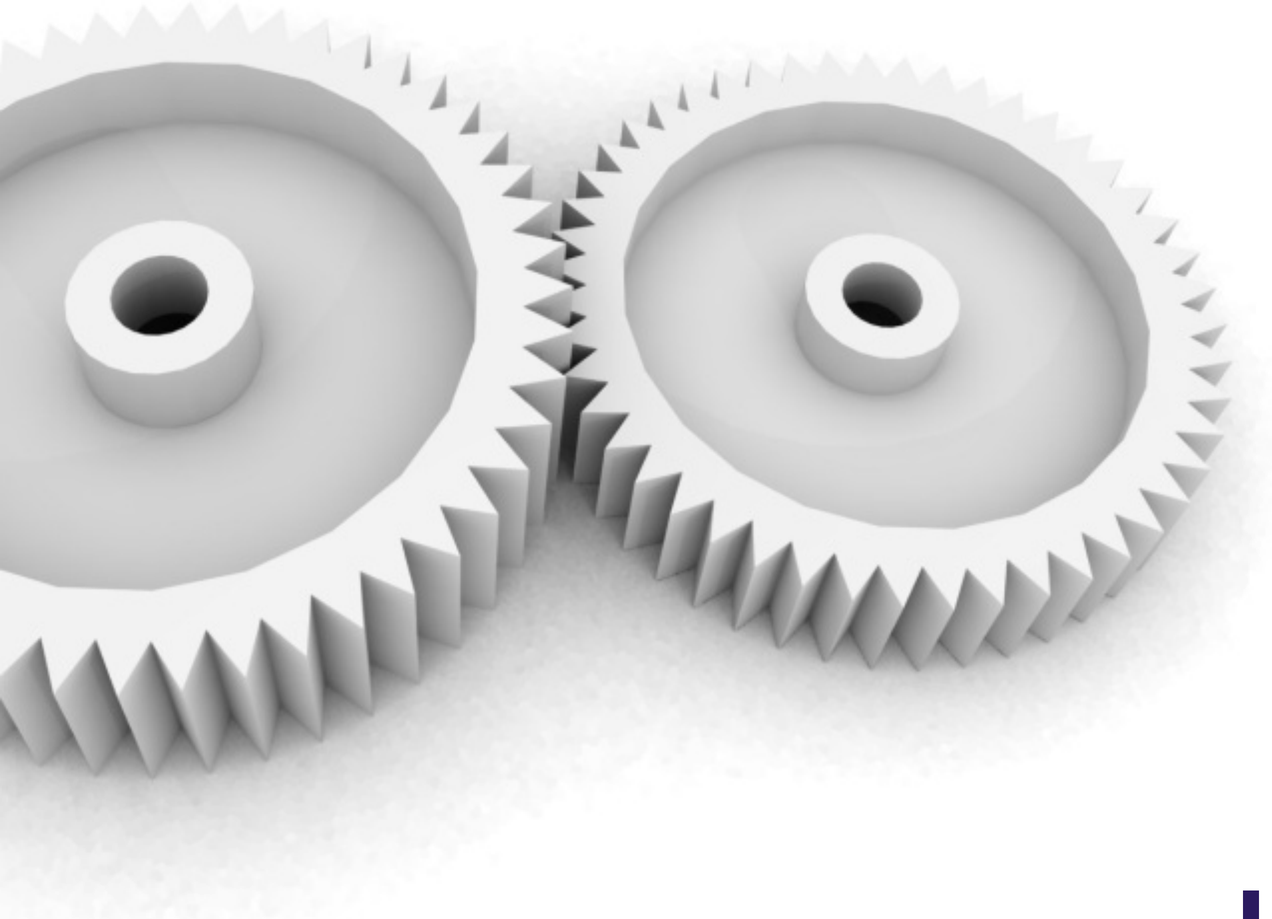


2

- Java event APIs
- REST-based APIs
- Ability to push flows

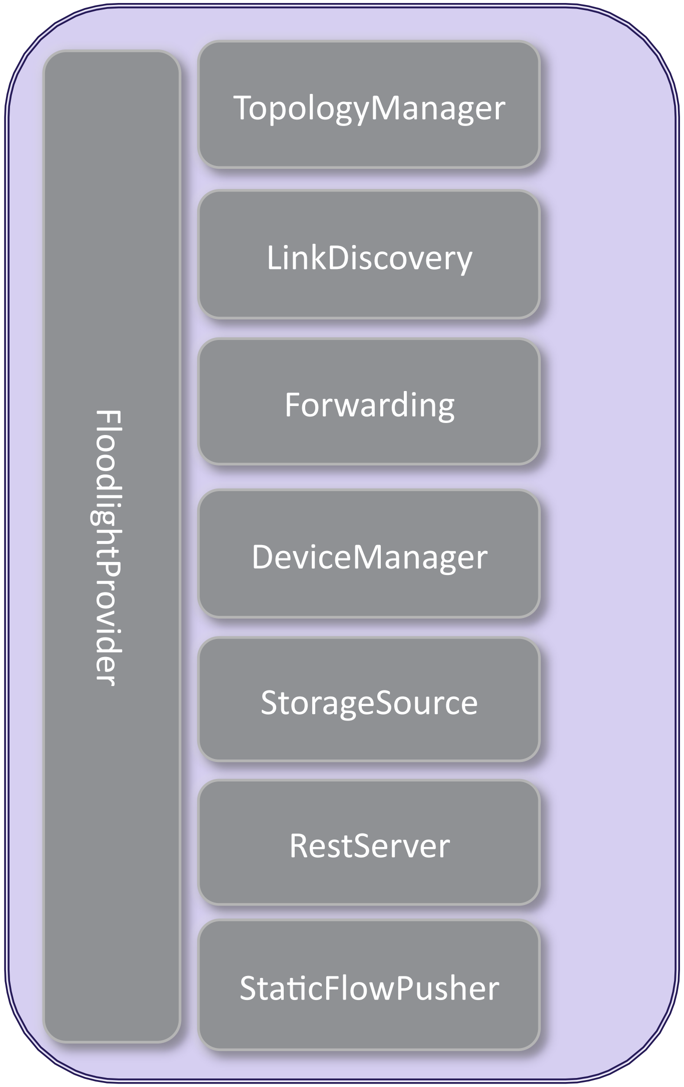
3

- Support for integrating with non-OpenFlow networks



Internals

Floodlight Architecture



- Modules exporting “services”
- All modules in Java, support for Jython as well
- Main module is FloodlightProvider
  - Manages I/O to switches
  - Translates OF messages to Floodlight events
  - Multi-threaded via Netty library (all modules must be thread-safe)
- Rich, extensible REST API

Module Descriptions

FloodlightProvider

TopologyManager

- Computes shortest path using Dijsktra
- Keeps switch to cluster mappings

LinkDiscovery

- Maintains state of links in network
- Sends out LLDPs

Forwarding

- Installs flow mods for end-to-end routing
- Handles island routing

DeviceManager

- Tracks hosts on the network
- MAC -> switch,port, MAC->IP, IP->MAC

StorageSource

- DB style storage (queries, etc)
- Modules can access all data and subscribe to changes

RestServer

- Implements via Restlets (restlet.org)
- Modules must implement RestletRoutable

StaticFlowPusher

- Supports the insertion and removal of static flows
- REST-based API

Controller Architecture is Modular

Every component is a loadable service

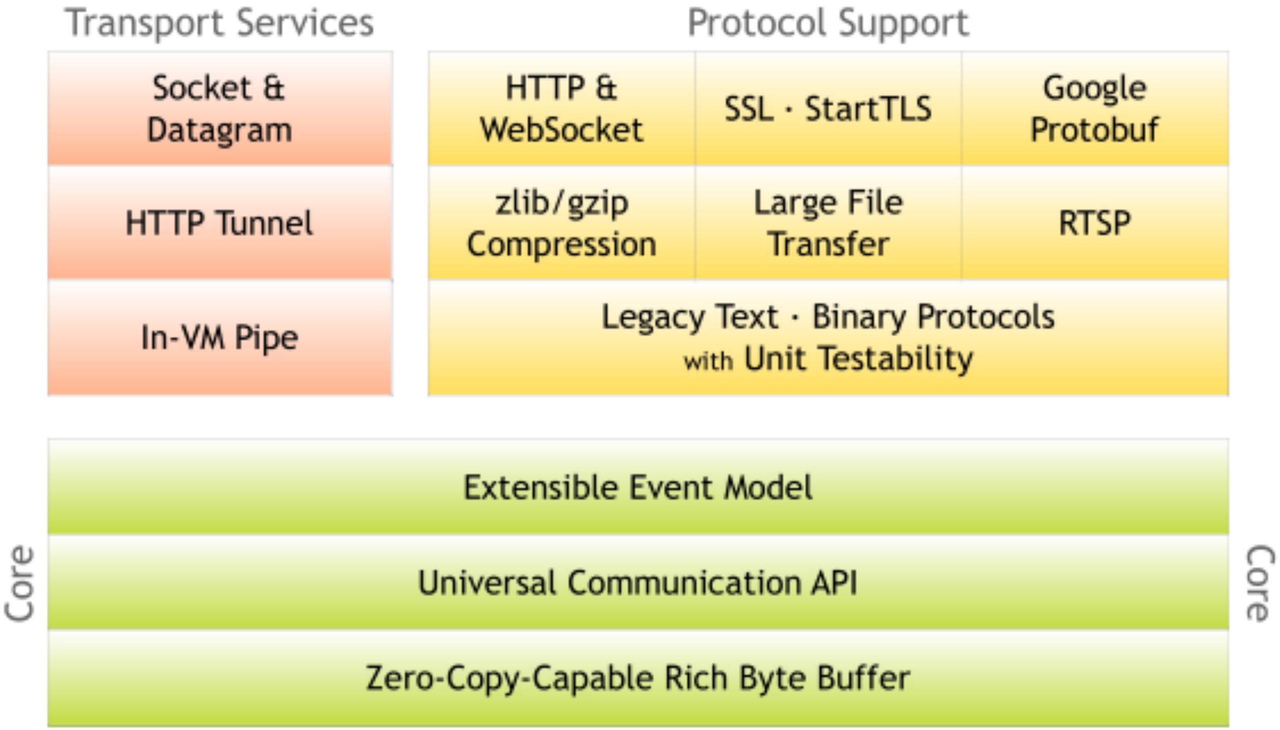
- A *service* is an interface that exports state and generates events
  - Consumers can GET/SET state and subscribe/ unsubscribe to events
  - Allows multiple implementations of the same service
  - Current implementation: Java Interface and REST (state export only)
- Each module:
  - *Consumes* zero or more service (dependencies)
  - *Provides* zero or more services
- Module loader resolves dependencies at load-time via config

IFloodlightModule Interface

Function	Description
getModuleDependencies()	What services does this module require?
getModuleServices()	Services does this module provide and how?
init(FloodlightModuleContext context)	Internal, before dependencies have init()’ed
startup(FloodlightModuleContext context)	External, with dependencies initialization

Netty

An asynchronous event-driven network application framework



Transport Services

Socket & Datagram

HTTP Tunnel

In-VM Pipe

Protocol Support

HTTP & WebSocket

SSL · StartTLS

Google Protobuf

zlib/gzip Compression

Large File Transfer

RTSP

Legacy Text · Binary Protocols with Unit Testability

Core

Extensible Event Model

Universal Communication API

Zero-Copy-Capable Rich Byte Buffer

Core

Threading Model

- All inter-module communication is through services
  - Inter-service calls need to be thread safe
- Event handling happens in publisher's thread context
  - Don't block, use a bottom half handler
- Thread pool executer service exists
  - Allows modules to share threads
- Number of shared data structures protected by locks
  - Any Java object can be an event
  - Standard locks apply : *synchronized*

Floodlight Provider Module

- Manages I/O from OF Switches
  - Tracks switch add/removes
  - Translates OF messages to Floodlight events
- IFloodlightProvider Service
  - addOFMessageListener(OFType type)
    - Ordering defined by caller with OFMessageListener iface
  - Map<dpid,Switches> getSwitches();
  - addOFSwitchListener();
  - injectOfMessage(IOFSwitch sw, OFMessage msg);
    - Used for recirculation-style hacks

Topology Module

Floodlight Automatically discover topologies

- OpenFlow and non-OF networks
  - SwitchClusters – managing OF-islands
- Controller sends active probes via packet out/in
  - Probes are formatted to look like LLDPs
- ITopologyService interface
  - getLinks()
  - addListeners()
  - inSameCluster(switch1, switch2)
  - Set<Switches> getSwitchesInCluster(switch1)

Device Manager Module

Host location tracking

- Tracks End-Host Locations in the network
  - Mac to (Switch, Port) mapping
  - Mac to IP
  - IP to Mac
- IDeviceManager Service
  - List<Device> getDevices()
  - addListener()
  - Device getDeviceByIPv4Address(ip)
  - Device getDeviceByDataLayerAddress(mac)
- TODO: Extend Device definition, include Vlan?

REST API Module

Any module can export via REST

- Implementation uses Restlets internally
  - www.restlets.org
- IRestAPI Service
  - addRestletRoutable(RestletRoutable rr)
- Your module implements RestletRoutable
  - String basePath()
    - “/rest/version1/myMod”
  - Restlet getRestlet(Context)
    - New Router(context).attach(“/switch/all/{statType}/json”, MyStatClass.class}
- MyStatClass extends org.restlet.resources.ServerResource

Getting Involved - OpenFlowHub

A community of open source OpenFlow developers

FloodLight

What it is:

1. A community of open source OpenFlow developers
2. An OpenFlow Blog (available for guest authors)
3. Free hosting, tools, and marketing for open source projects
   - Wiki, forums, bug tracking tools, logos, etc.

Get involved:

- **Submit a project**
- Write a blog post
- http://www.openflowhub.org
- Contact: mike.cohen@openflowhub.org

Projects:

OpenFlowHub BROADCASTING OPENFLOW

INDIGO PROJECT

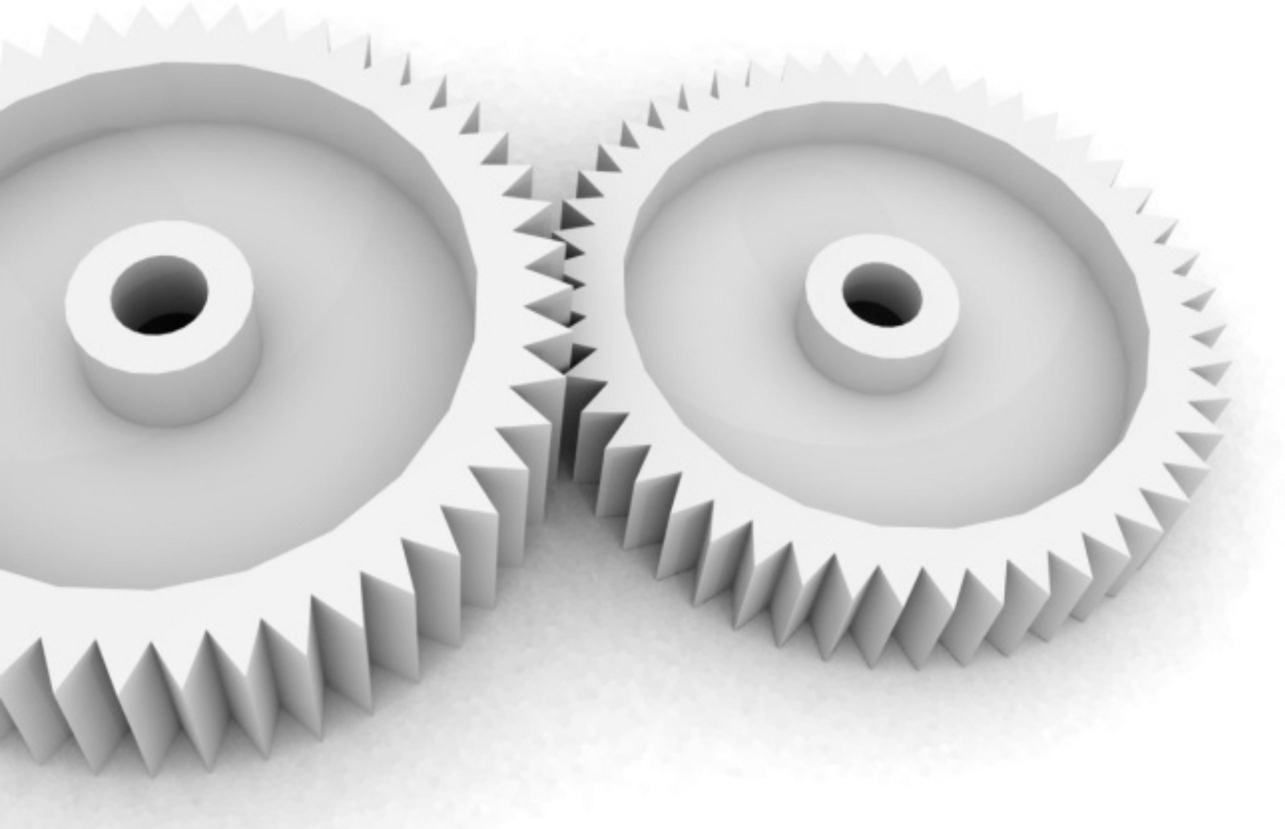
FlowScale

FloodLight

RouteFlow

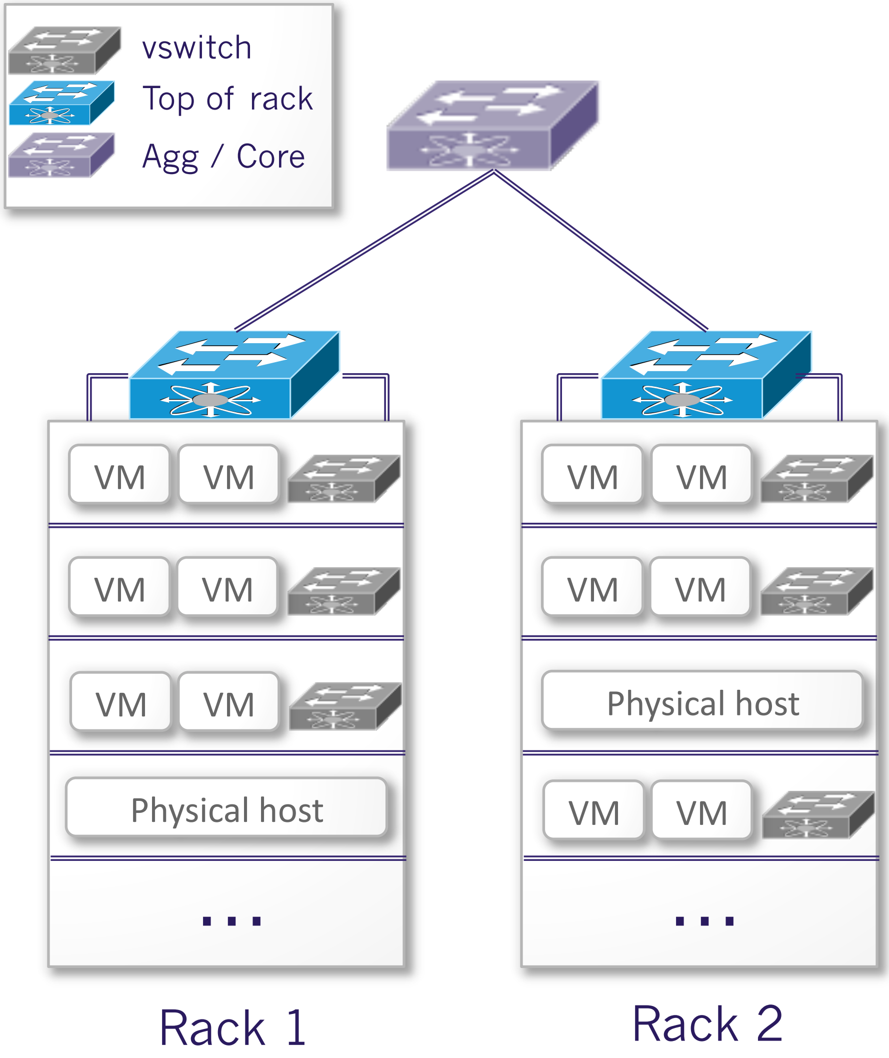
Interested in Learning More?

- Check out the website
  - http://floodlight.openflowhub.org
- Join the mailing list:
  - http://groups.google.com/a/openflowhub.org/group/floodlight-dev/topics
  - Or just email floodlight-dev@openflowhub.org
- Get the code:
  - http://floodlight.openflowhub.org/download



End

OpenFlow Topologies



Need OpenFlow at last hop in the network

- **For vms:**vswitch
- **For physical hosts:** Tor

Floodlight can manage multiple “islands” of OpenFlow switches

Demo

Problem:

Track the last N Packet-Ins seen by the controller and expose it via a REST API

What you will see:

1. Adding a new module
2. Creating a REST API
3. Running Floodlight