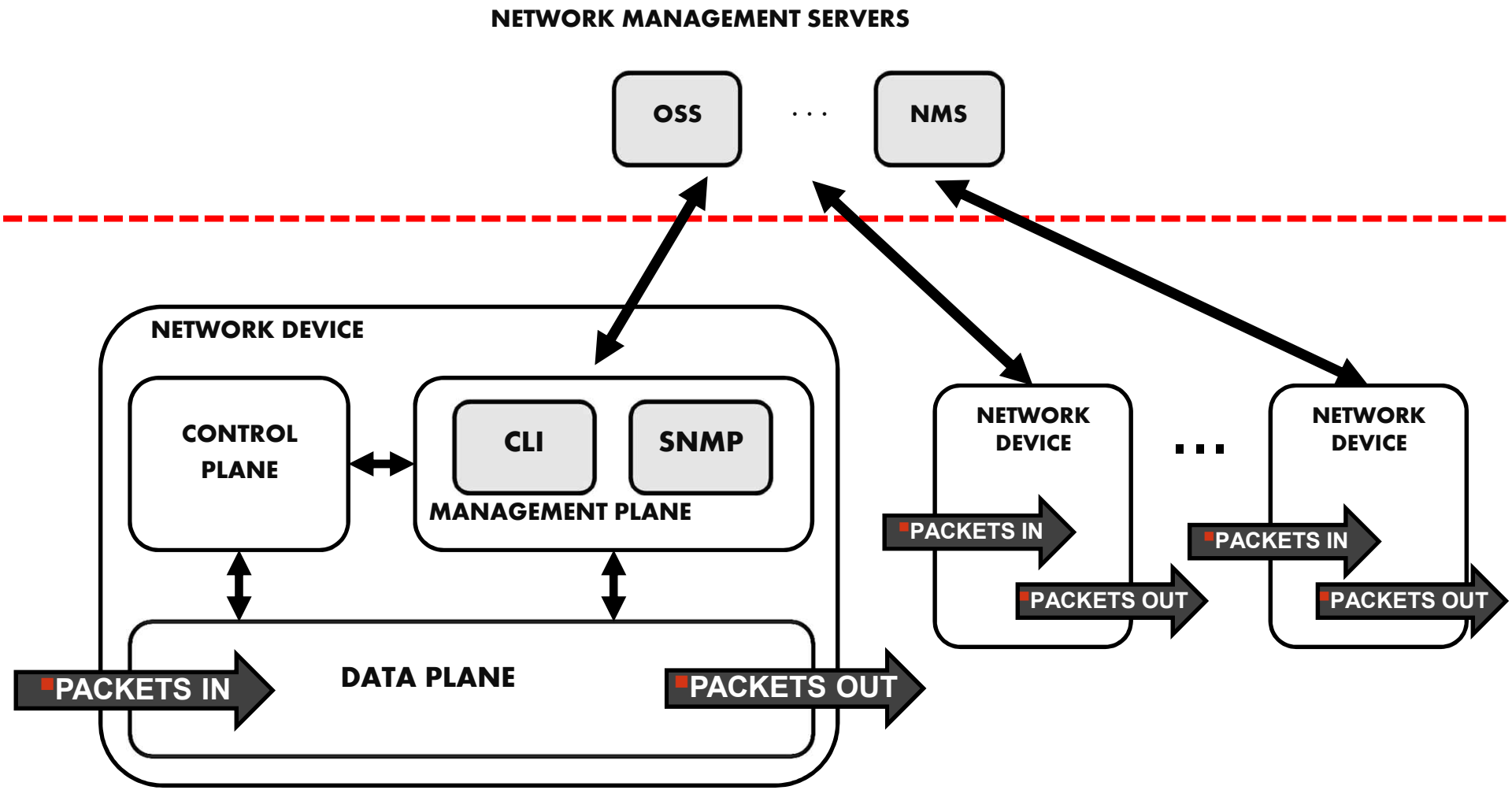
Scalable and Reliable control and Management for SDN-based Large-scale Networks

CJK Workshop @ CFI2014

2014. 06.18.

Taesang Choi
ETRI

Traditional Control & Network Management Architecture



Traditional Control & Network Management Architecture

- Network functionalities such as data, control, and management planes
  - → Distributed and embedded within the vendor specific networking devices
- Remote management through provisioning and configuration
- Closed, inflexible, complex, error-prone, and hard-to-manage production network problems

SDN has changed the way to manage Nets

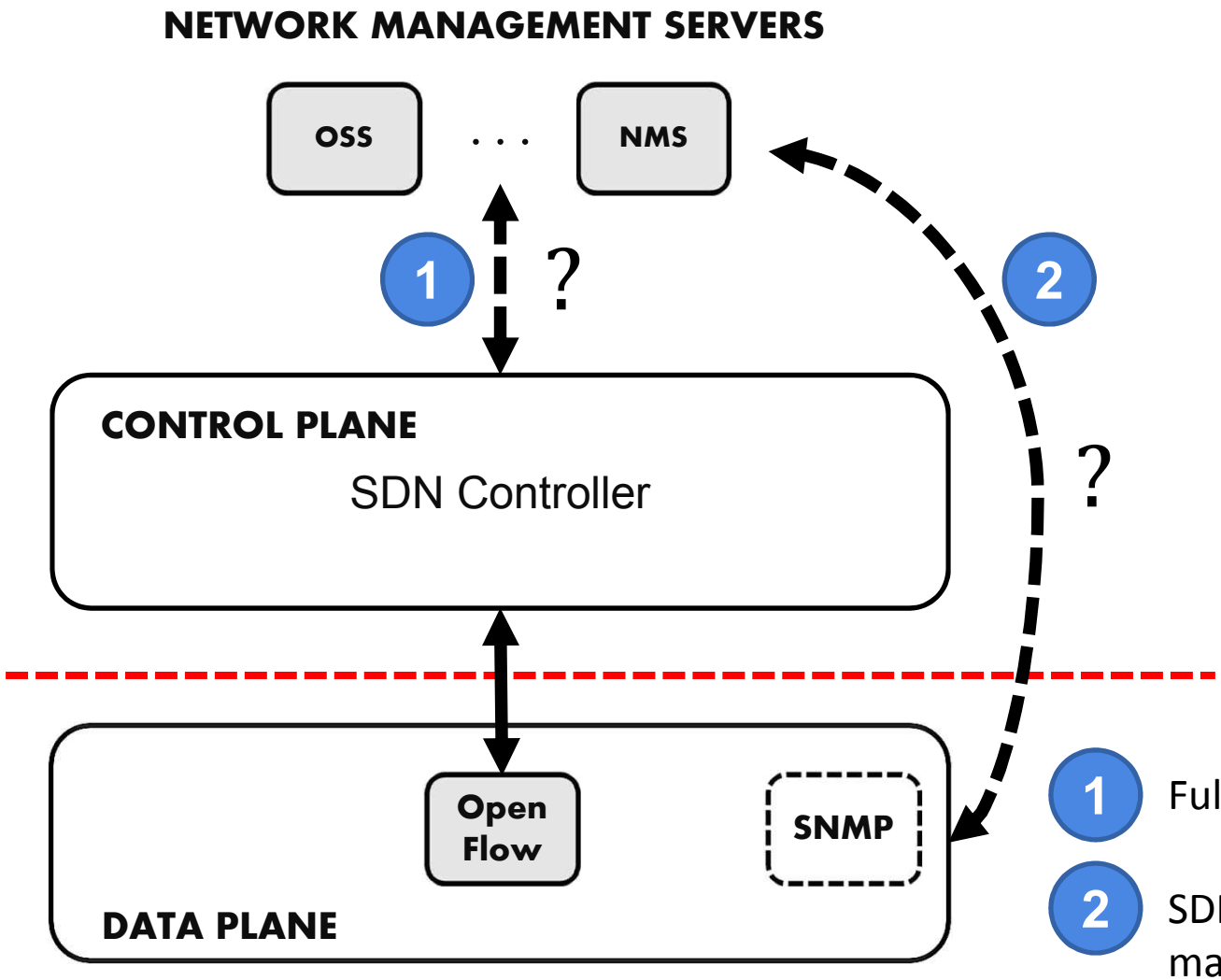
- **Enables direct programming and centralized management by deco upling the network control and forwarding functions**

- **Centralized control of multi-vendor environments (standard)**
  - Devices from any vendor, including switches and routers
  - Quick deployment, configuration and updating devices across the entire network

- **Reduced complexity through automation (deployment)**
  - Flexible network automation and management framework
  - Possibility of developing tools that automate many management tasks that are d one manually today

- **Higher rate of innovation (new network functionality)**
  - Network in real time to meet specific business needs and user requirements as t hey arise

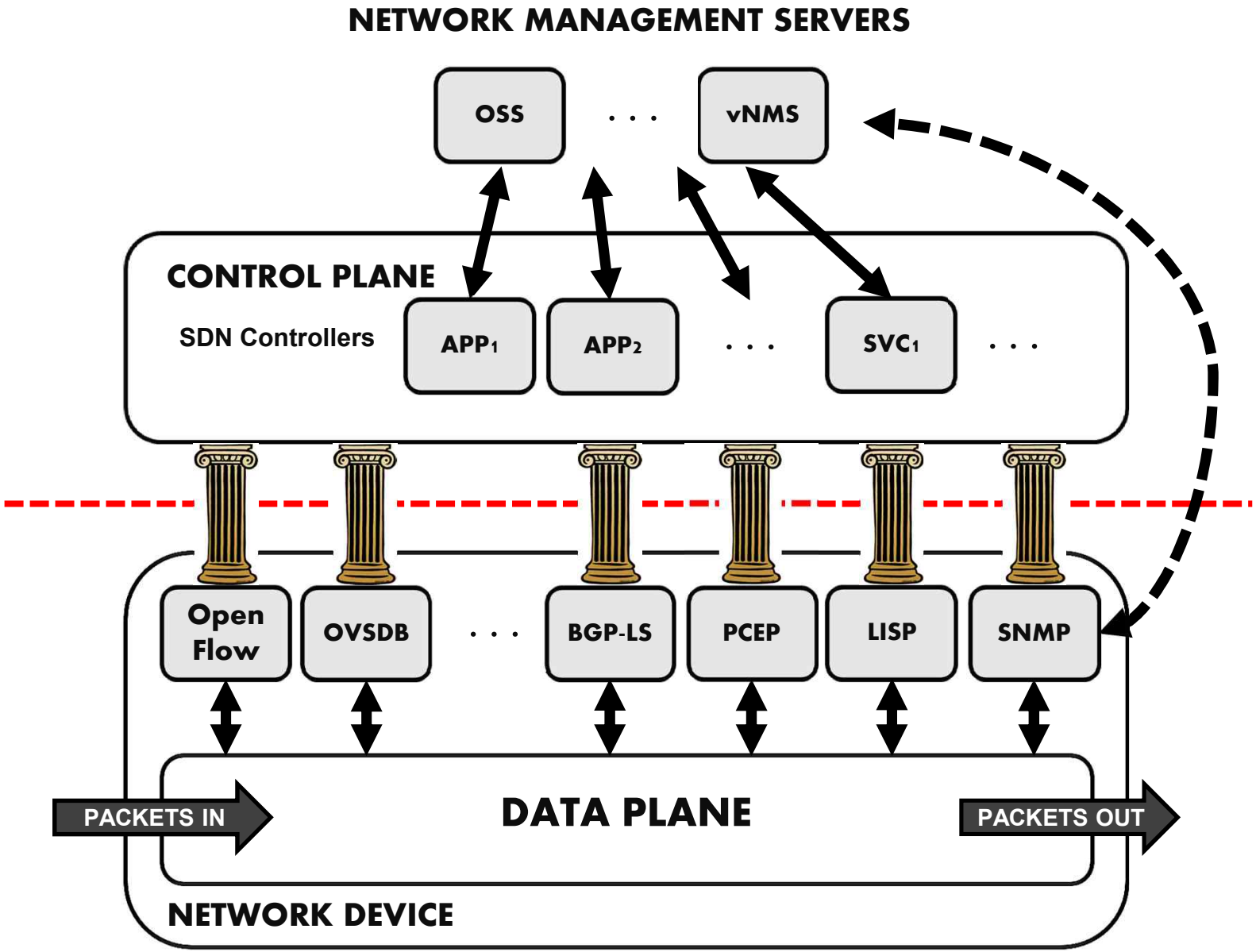
Ambiguous SDN Control & Management?



1 Fully integrated management

2 SDN controller agnostic management

Open Daylight Control & Management Arch.



Network Management in a Nutshell

IT R&D Global Leader ETRI

❖ Control and measurement

- Control: access control, routing, etc.
- Measurement:
  - Traffic engineering: flow size (elephant flows to route), traffic distribution (estimate rack-to-rack traffic matrix)
  - Accounting
    - ✓ **Billing based upon network usage**
  - Troubleshooting
    - ✓ **Find performance bottlenecks**
    - ✓ **Attacks**
    - ✓ **Failures**

❖ SDN focuses on control of traffic engineering (so far)

❖ Scalability, Availability, and Accuracy issues

Major Issues

❖ Scalability

- Control messages priority differentiate (annotation)
- Kind of best-efforts for control traffic
- Solutions so far: extension of SW(DevFlow, etc.) and Controller (clustering, hierachy)

❖ Availability

- 99.999% availability for carrier networks

❖ Inaccurate and Unreliable Management

- Management practice takes mainly remote approaches
- Network events should be inferred by the remote management systems

  → Potential network problems are often accumulated and enlarged

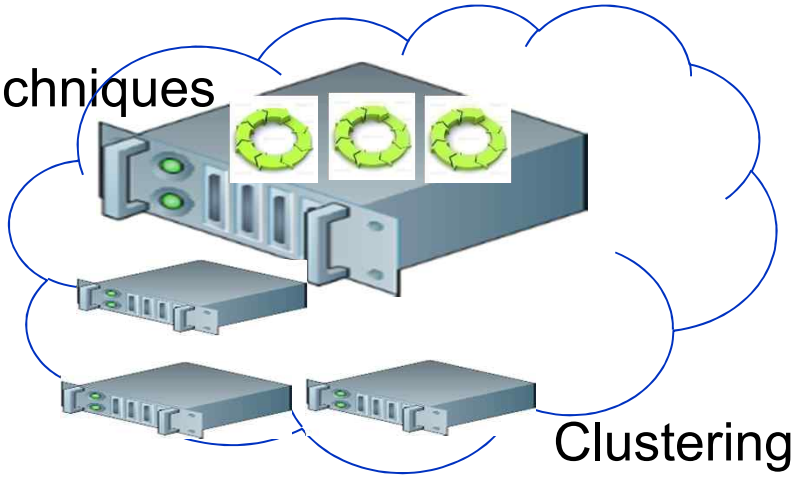
  → Diagnosis is delayed, inaccurate, unreliable, and not scalable
- SDN remote/centralized control tends to extend legacy network mgmt problems into the control plane

Existing SDN Scalability Solutions

Central SDN Controller

Enhancing controller techniques

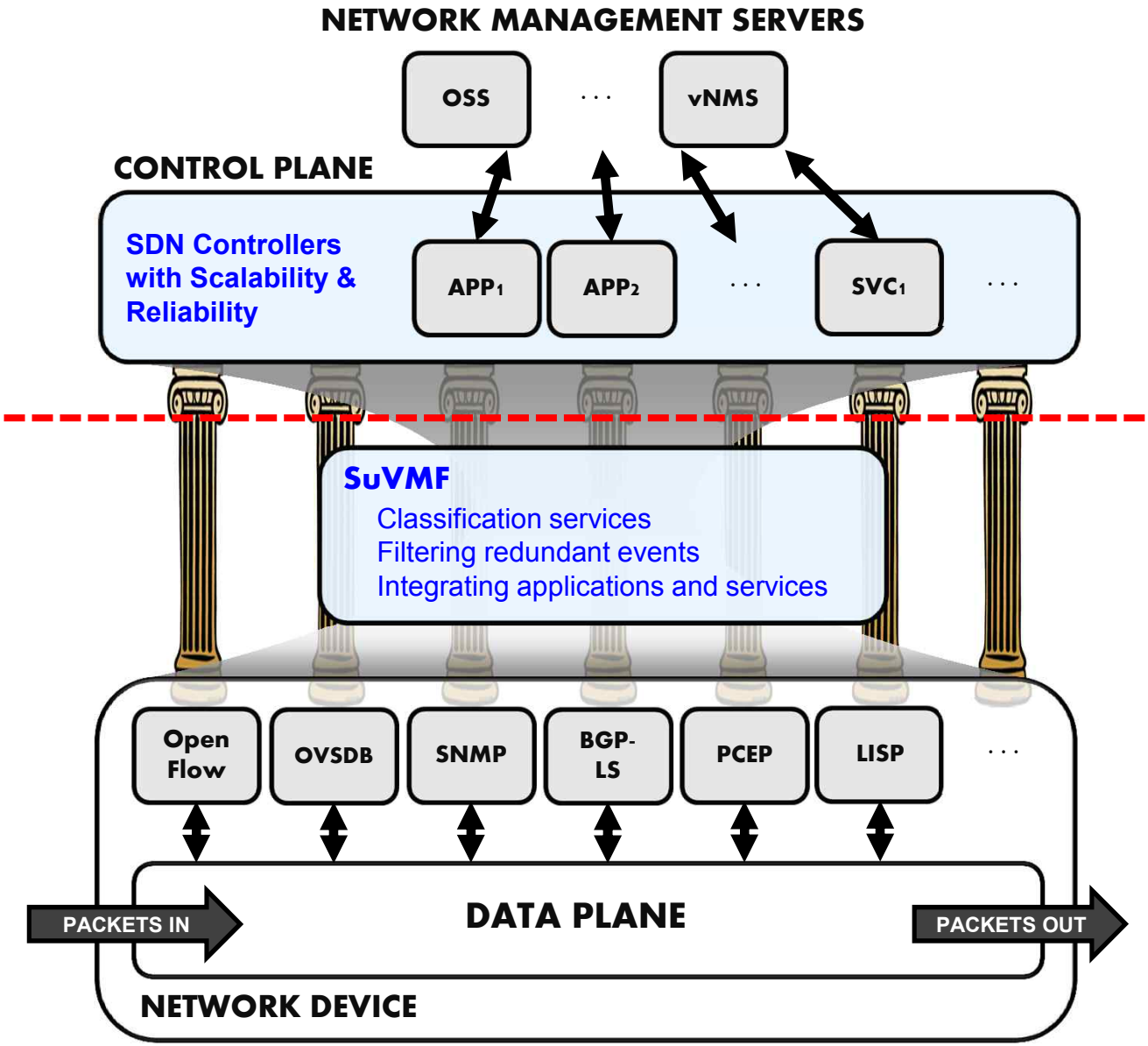
Recent controllers support methods of parallelism



Clustering multiple controllers (HyperFlow, ONIX).

Distributing some control functions to the switches (DevoFlow, DIFANE) to reduce control messages.

IRIS<CoMan> Architecture



IRIS-Controller

❖ A Spin-off project from Floodlight

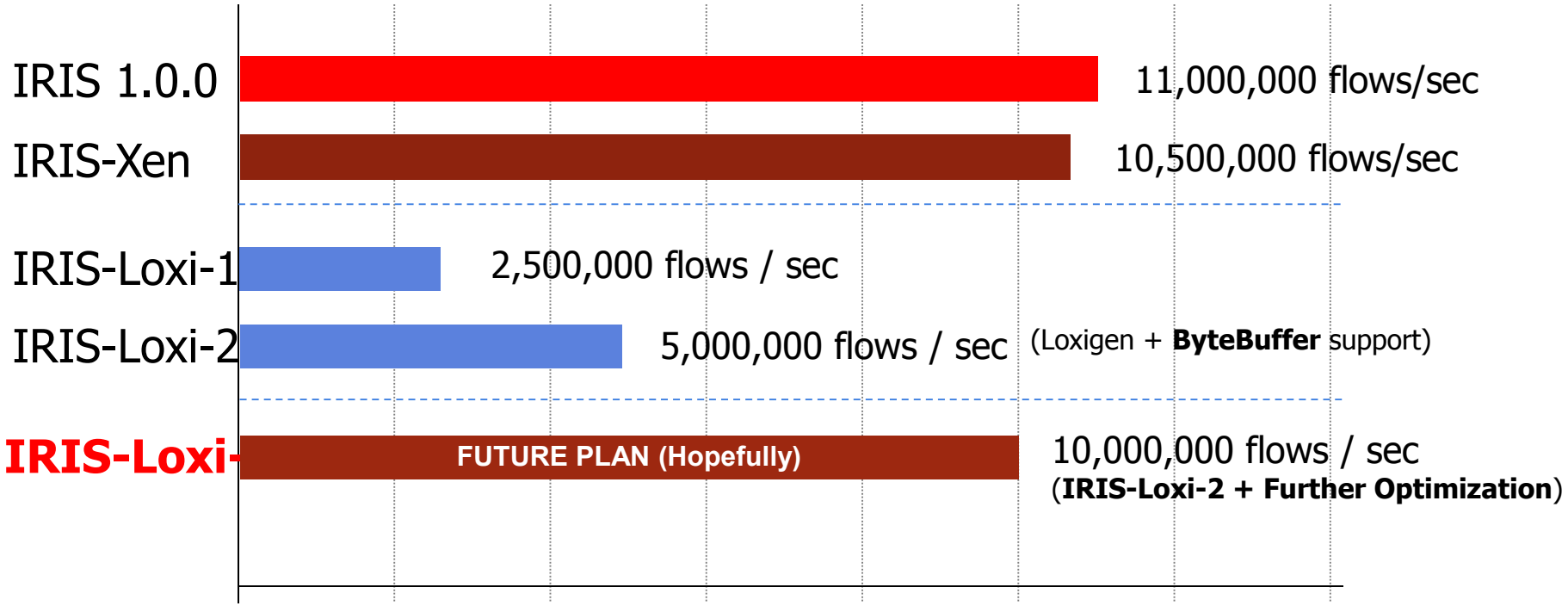
❖ Floodlight

- Openflow-based SDN Controller from BigSwitch (Open Source)
- Supports Openflow 1.0 (and soon will announce 1.3 support)
- Adopted widely by research communities

❖ IRIS (2013~)

- **Yet another Openflow-based SDN Controller from ETRI**
- With an IO engine implemented from scratch on top of Java NIO
- **Supports Openflow 1.0~1.3**
  - **Floodlight/Loxigen**-based Openflow API
- **Provides an Open-source version: OpenIRIS** (http://openiris.etri.re.kr)
- Provides a northbound API which is fully compliant with that of Floodlight
  (to support 3rd party applications from various research communities)
- **Focus on solving the scalability / availability issues of the centralized control**
- **Current release is v2.0.8**

IRIS-C: Single-box performance



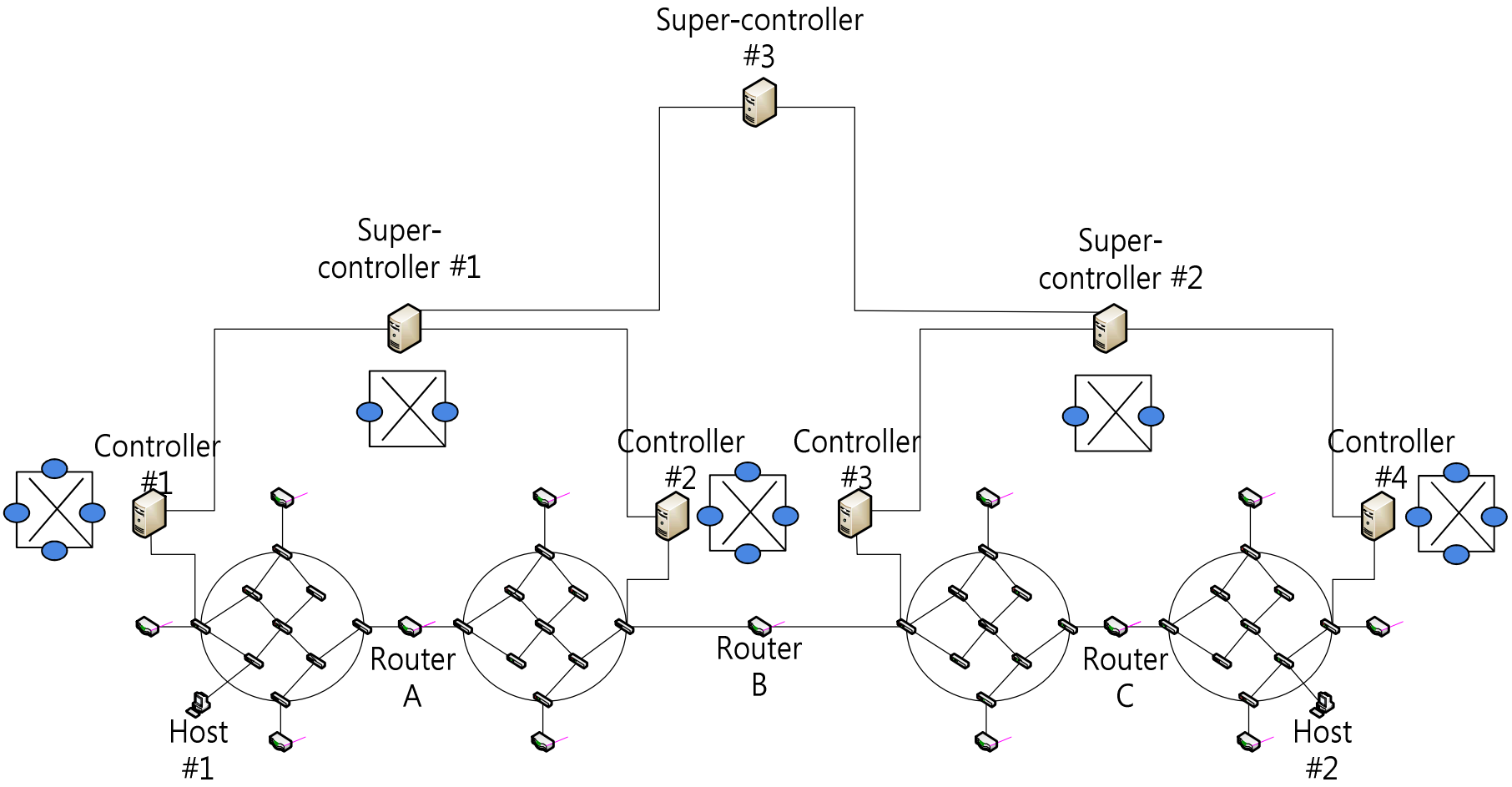
Ubuntu 12.04 LTS 64bit (Kernel : 3.5.0-23-generic)
CPU : Intel Xeon E5-2690 v2 3.00GHz (20 physical core)
RAM : 64G

Testing environment

Benchmarking

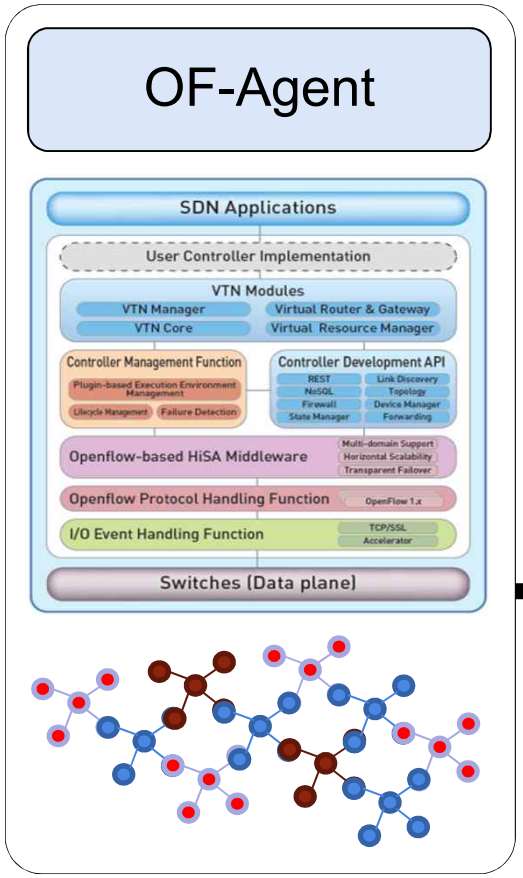
Cbench : 1 core
controller: max 19 core

Controller Architecture for Scalability



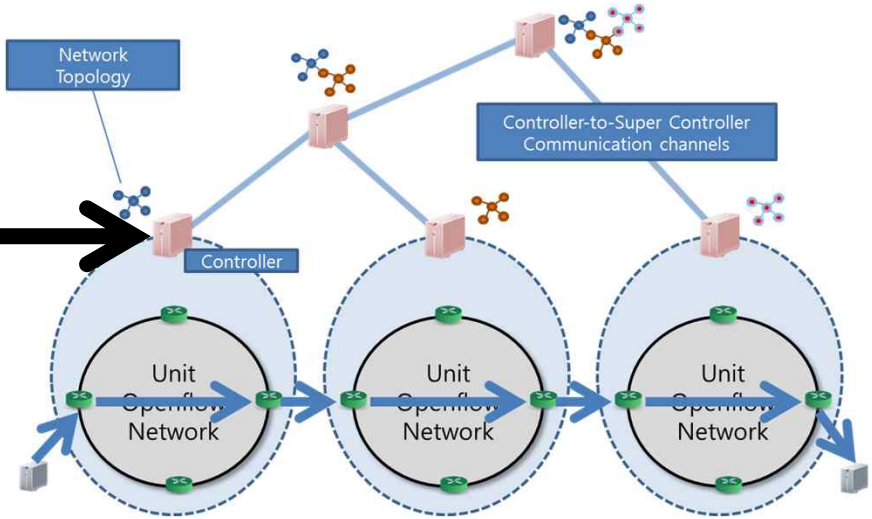
IRIS-RAON

RAON Architecture

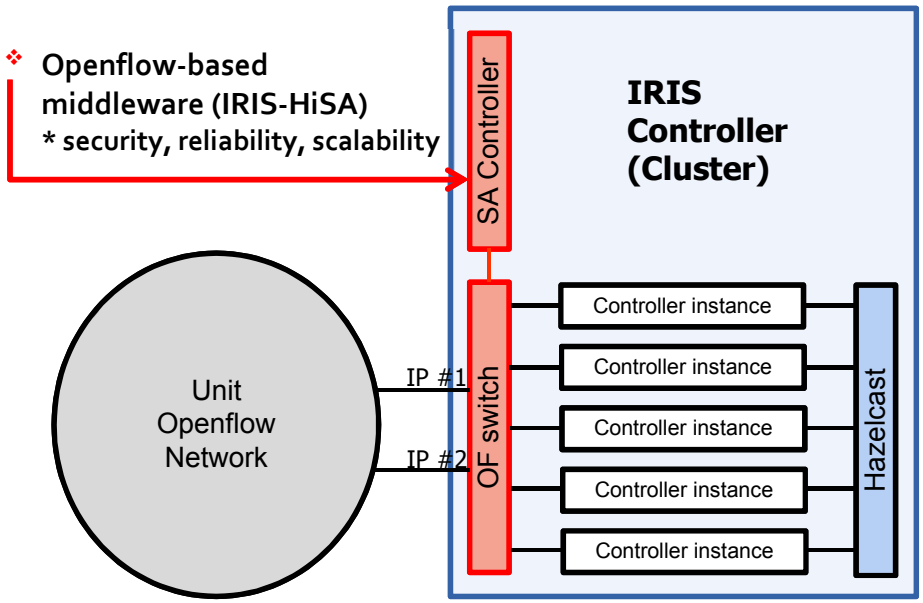


A Network as a "Big Switch" :-P

Recursive Abstraction of Large Network into a single switch with many ports



IRIS-HiSA for Availability



We believe OF-based brokering middleware will be one of the promising applications of Openflow

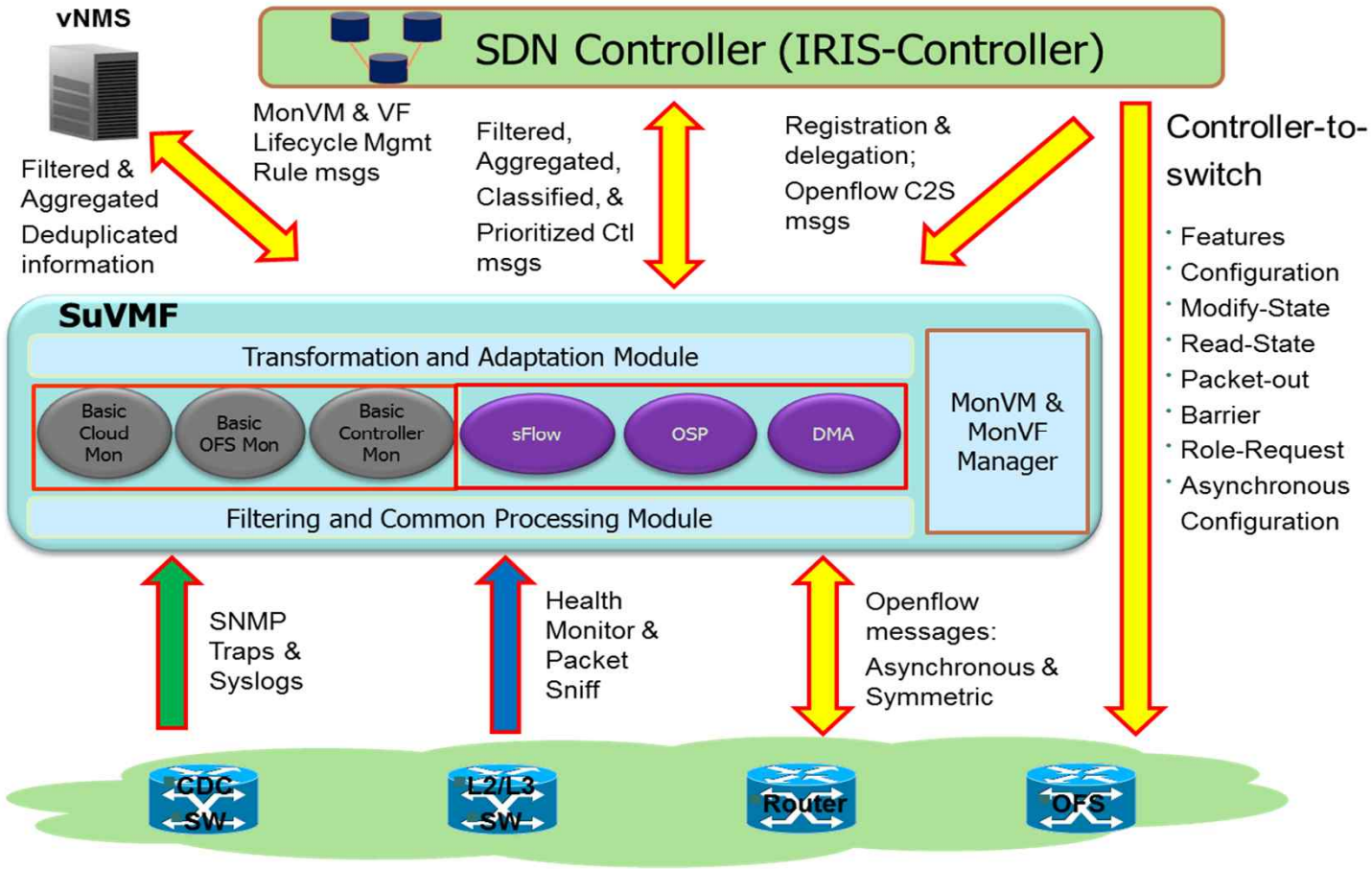
- Considerations
  - Addresses exposed to data plane
  - Transparency
  - Horizontal scalability
  - High availability
  - State sharing
- Functionalities
  - Load balancing among physical controller instances
  - Switch migration
    - For failed controller instances
    - For newer controller instances
  - Security
    - Immune to attack such as DDoS

Software-defined Unified Monitoring Agent for SDN

- A switch-side agent device providing control and management abstraction layer among SDN controllers, legacy NMS, and Openflow switches

- Functionalities:
  - Monitoring health of OpenFlow switches
  - Inspecting and verify the traffic (management sniffer)
  - Aggregation of verbose management information (syslog, SNMP, etc.)
  - Classification and prioritization of Openflow Asynchronous control messages
  - Filtering unnecessary messages
  - Identification of potential DoS attacks

SuVMF Architecture



User-defined Monitoring Service (UMS) Functions

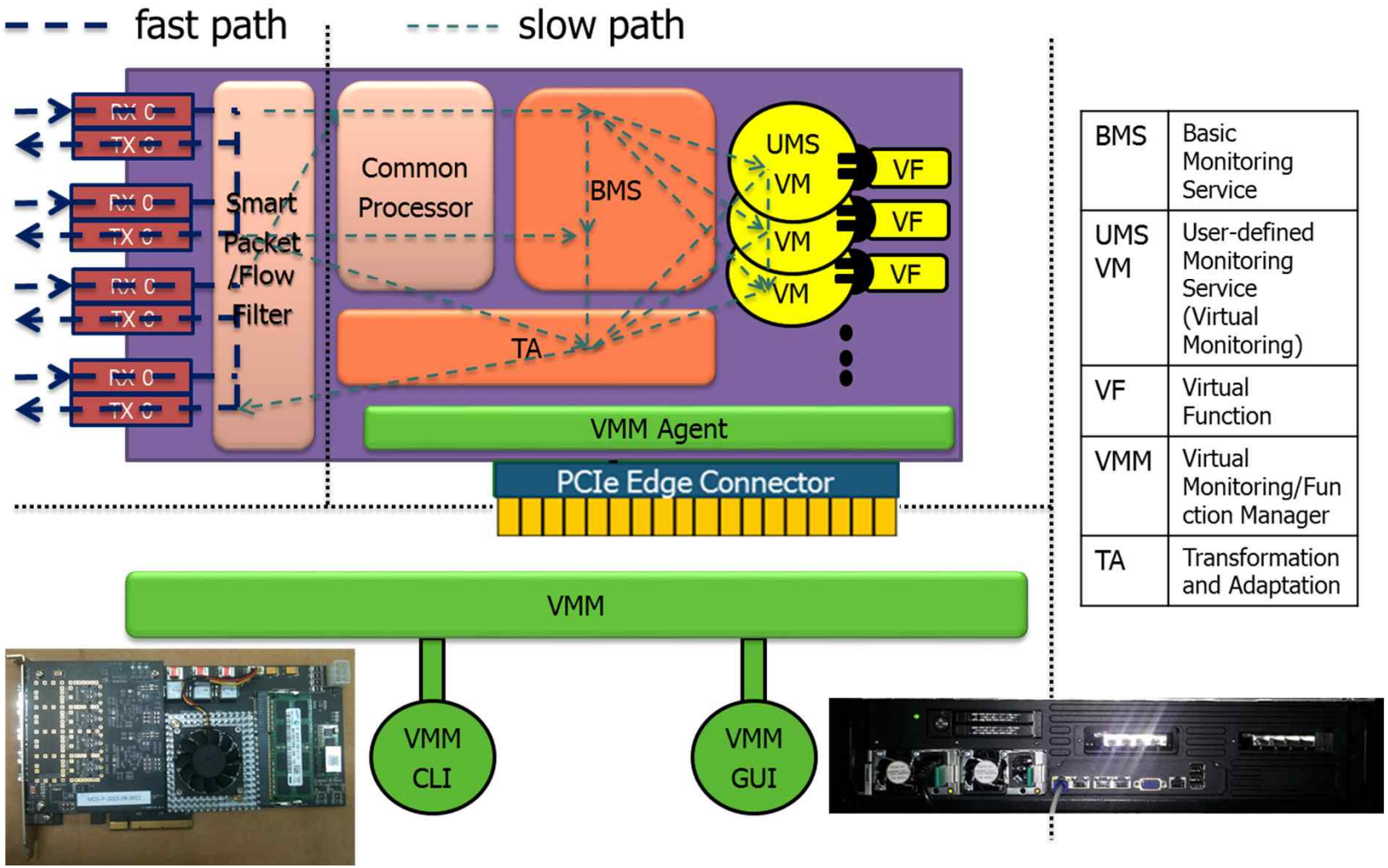
❖ MAC (Modify and Annotate Control)

- All the events should be inferred by a centralized remote SDN controller . As the underlying network is an inter-related complex system, it is not straightforward to identify a root cause of a problem or to chain policies.
- Provides algorithms, protocols, and facilities to modify and annotate con trol messages (e.g., adding sequence numbers in the control message) to assist remote network monitoring, control message differentiation, an d resource isolation.

❖ DMA (Detect and Mitigate Abnormality)

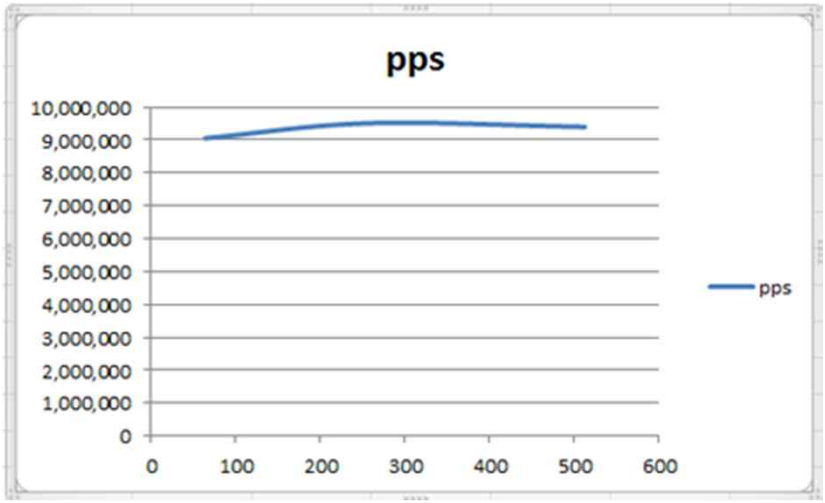
- Simple network status change (due to failure and attacks) may create v arious cascading critical network malfunctions.
- Detects and mitigates problems as near as the source of the problem.

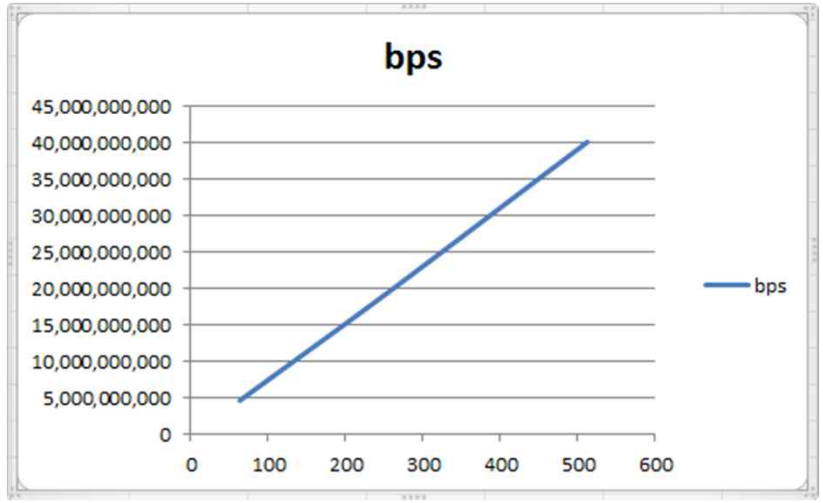
SuVMF Implementation Prototype



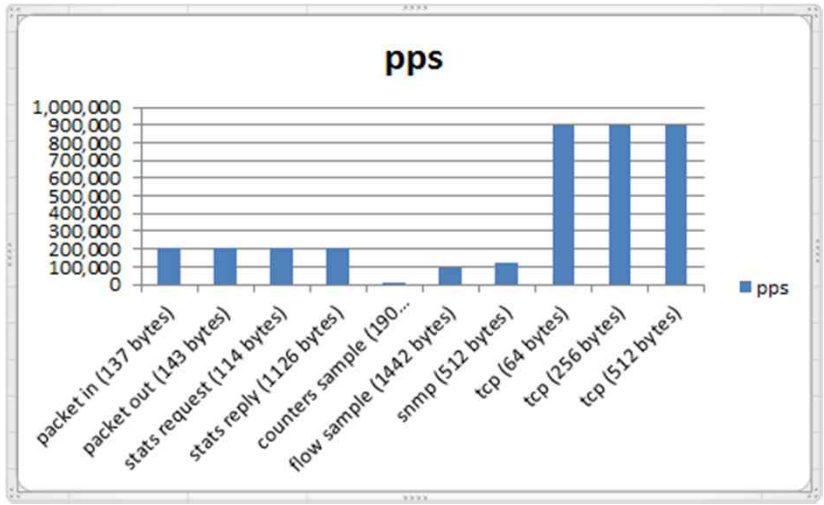
BMS	Basic Monitoring Service
UMS VM	User-defined Monitoring Service (Virtual Monitoring)
VF	Virtual Function
VMM	Virtual Monitoring/Function Manager
TA	Transformation and Adaptation

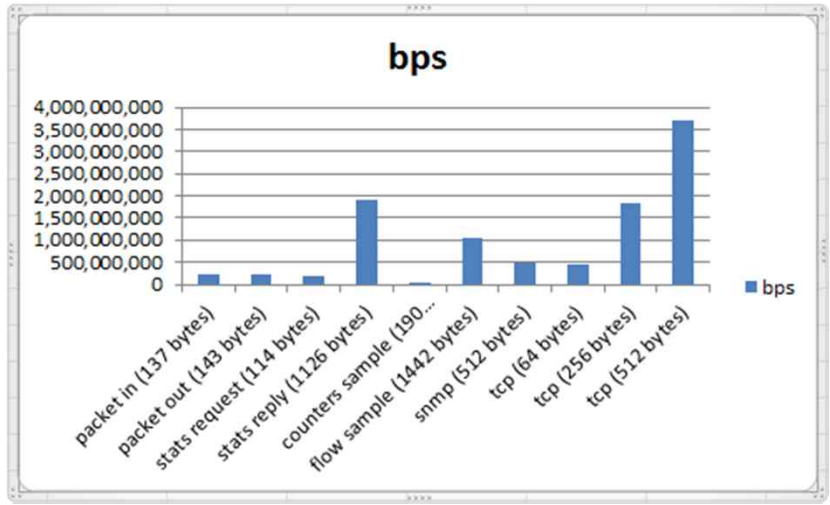
SuVMF Arch. for Intra-System Scalability





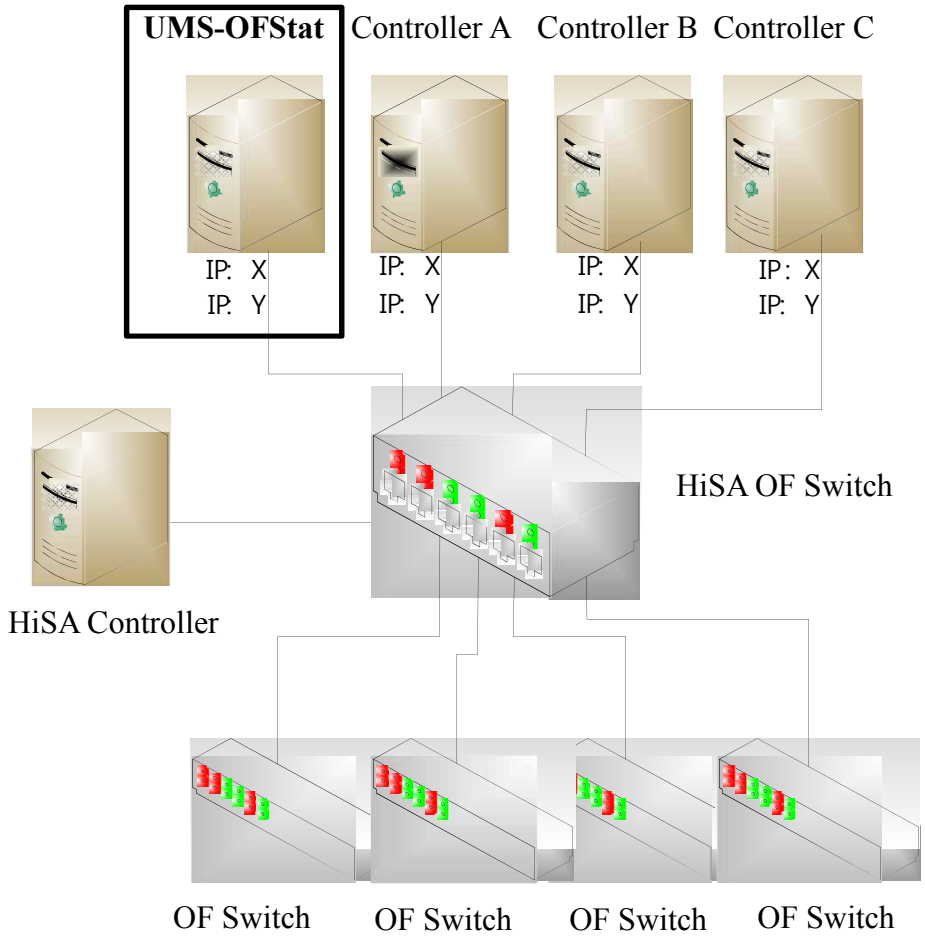
FCP Throughput: 9 mpps, bps 5Gbps ~ 40 Gbps at 64 ~ 512 bytes



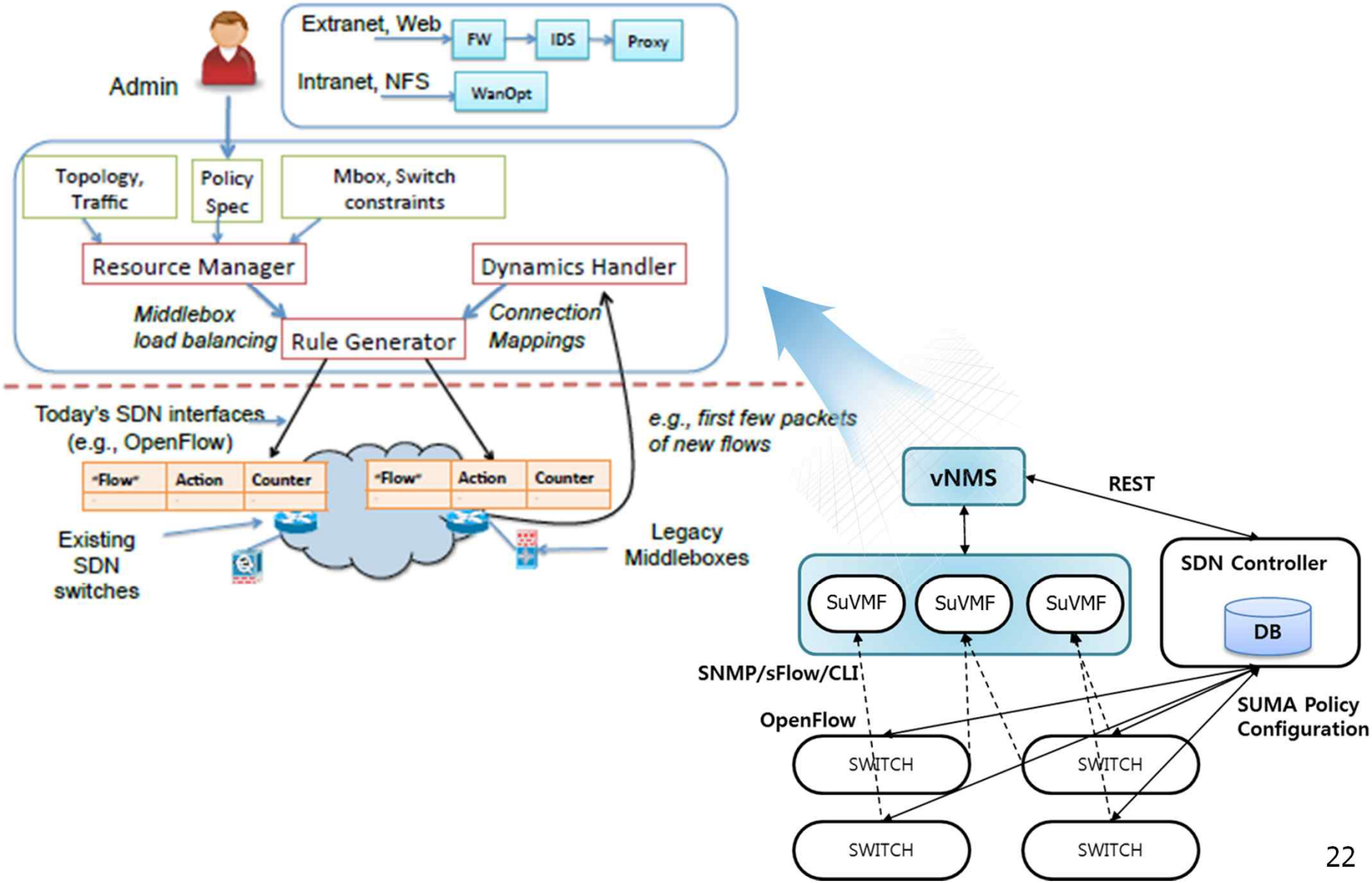


8 cores for FCP, 2x4 cores for BMS, 16 cores for UMS(sflow), 2 cores for VMM-agent, and 1 core for embedded Linux.
Test traffic is composed of 25% of 4 types OpenFlow msgs, 10% of sflow msgs, 5% of mgmt traffic, 60% of TCP dummy traffic in three different sizes (64, 256, and 512 bytes) out of a total of 10G traffic. The entire system throughput 3,296,000 pps/7.5 Gbps

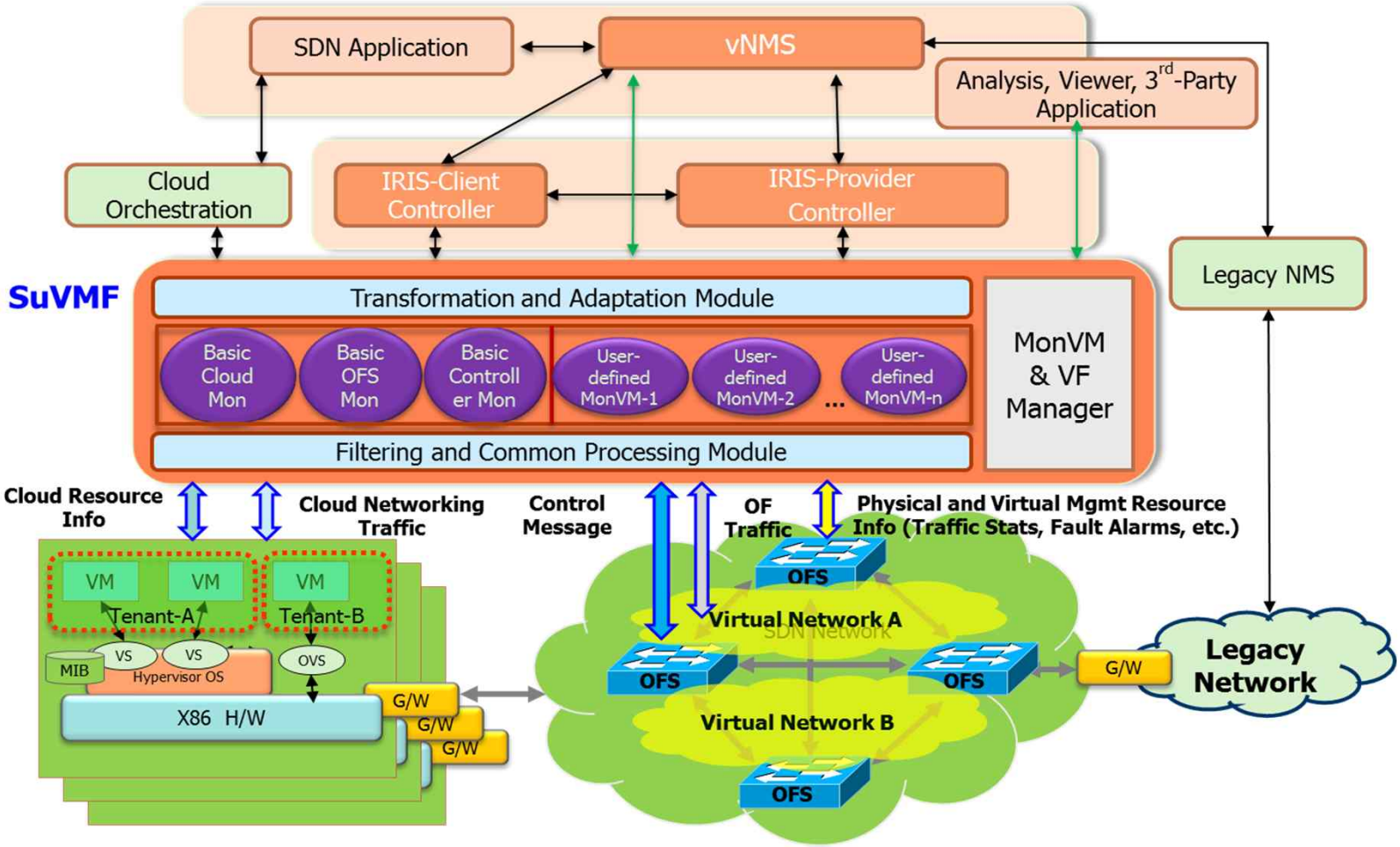
SuVMF Arch. for IRIS Controller Scalability



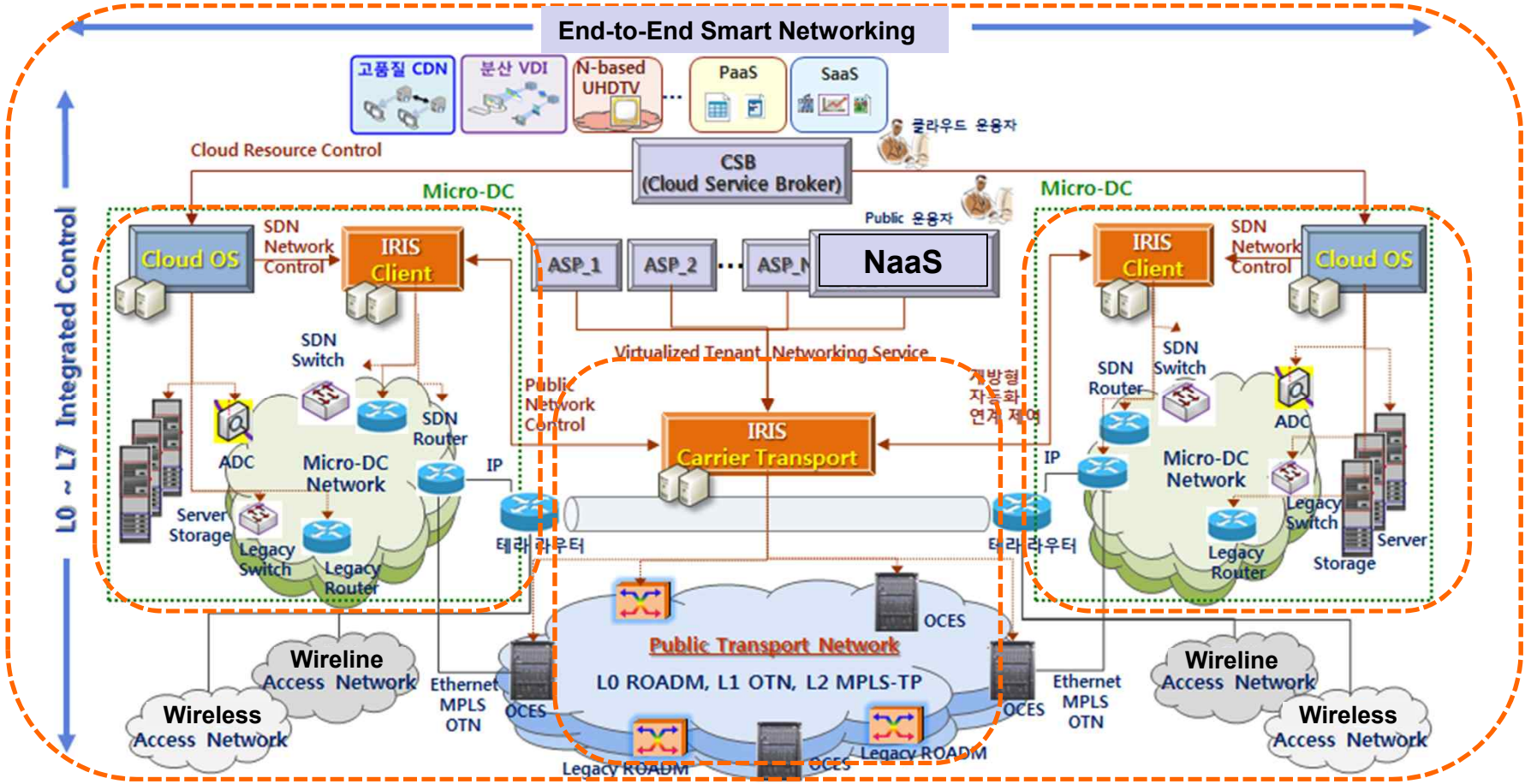
SuVMF Arch. for Inter-System Scalability



Overall Architecture: Putting Them Together



Overall Project Overview



Thank You

Q & A

choits@etri.re.kr