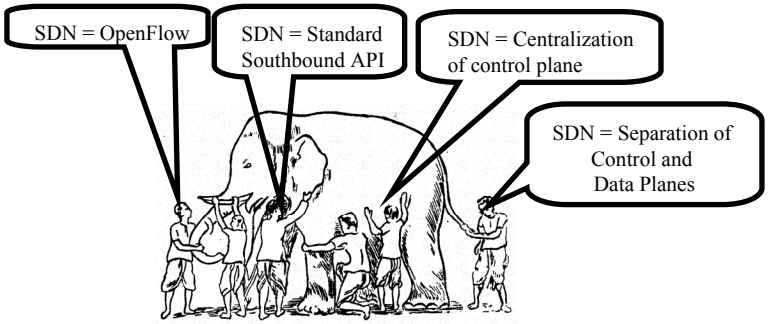
AppFabric: Application Deployment and Service Chaining in Future NFV Cloud WAN Environments



RAJ JAIN

Project Leader: Subharthi Paul
Washington University in Saint Louis
Saint Louis, MO 63130, Jain@cse.wustl.edu

CRC Tech Seminar, Cisco, May 15, 2014

These slides and audio/video recordings of this talk are at:
http://www.cse.wustl.edu/~jain/talks/apf_csc.htm

Overview

1. SDN 1.0 and SDN 2.0
2. Network Function Virtualization and Service Chaining
3. Function Virtualization and Service Chaining
4. Cloud of Clouds over WAN

Planes of Networking

- **Data Plane**: All activities involving as well as resulting from data packets sent by the end user, e.g.,
  - Forwarding
  - Fragmentation and reassembly
  - Replication for multicasting
- **Control Plane**: All activities that are necessary to perform data plane activities but do not involve end-user data packets
  - Making routing tables
  - Setting packet handling policies (e.g., security)

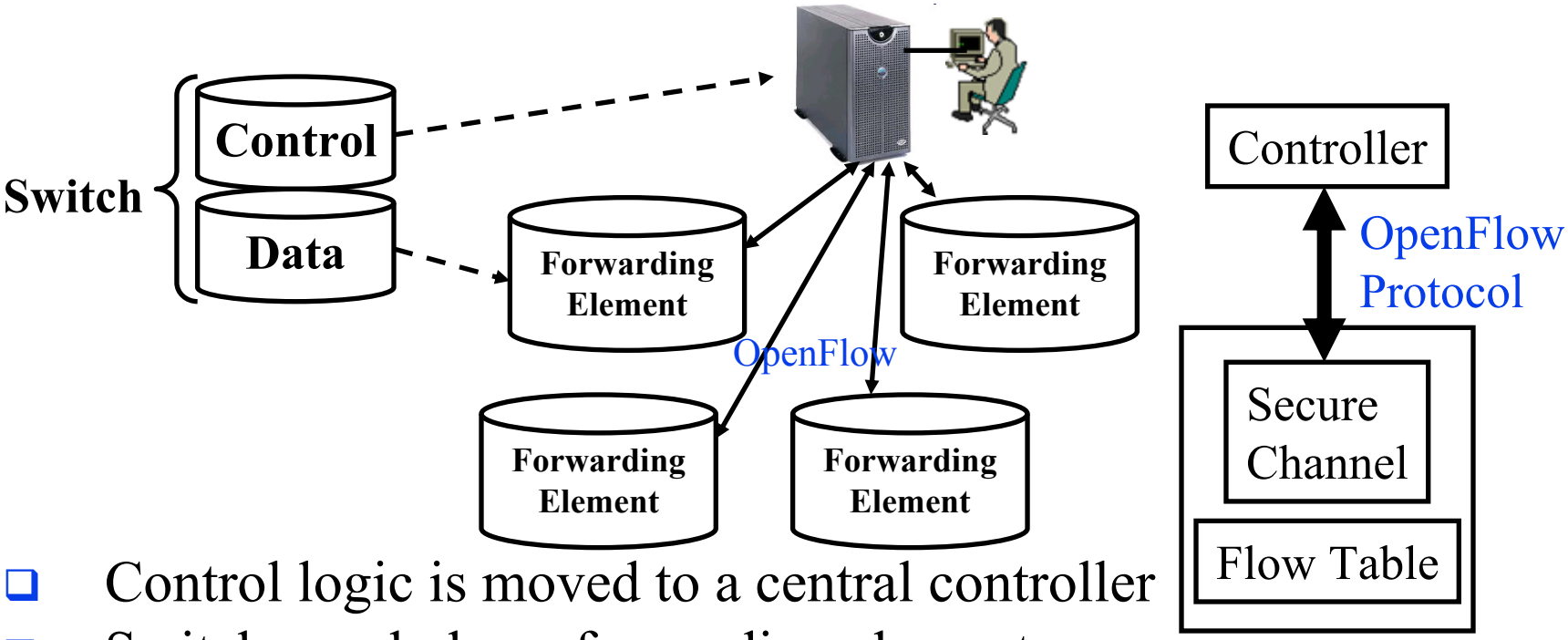
Dest.	Output Port	Next Hop

Ref: Open Data Center Alliance Usage Model: Software Defined Networking Rev 1.0,"
http://www.opendatacenteralliance.org/docs/Software_Defined_Networking_Master_Usage_Model_Rev1.0.pdf

Planes of Networking (Cont)

- **Management Plane**: All activities related to provisioning and monitoring of the networks
  - Fault, Configuration, Accounting, Performance and Security (**FCAPS**).
  - Instantiate new devices and protocols (Turn devices on/off)
  - Optional ⇒ May be handled manually for small networks.
- **Services Plane**: Middlebox services to improve performance or security, e.g.,
  - Load Balancers, Proxy Service, Intrusion Detection, Firewalls, SSL Off-loaders
  - Optional ⇒ Not required for small networks

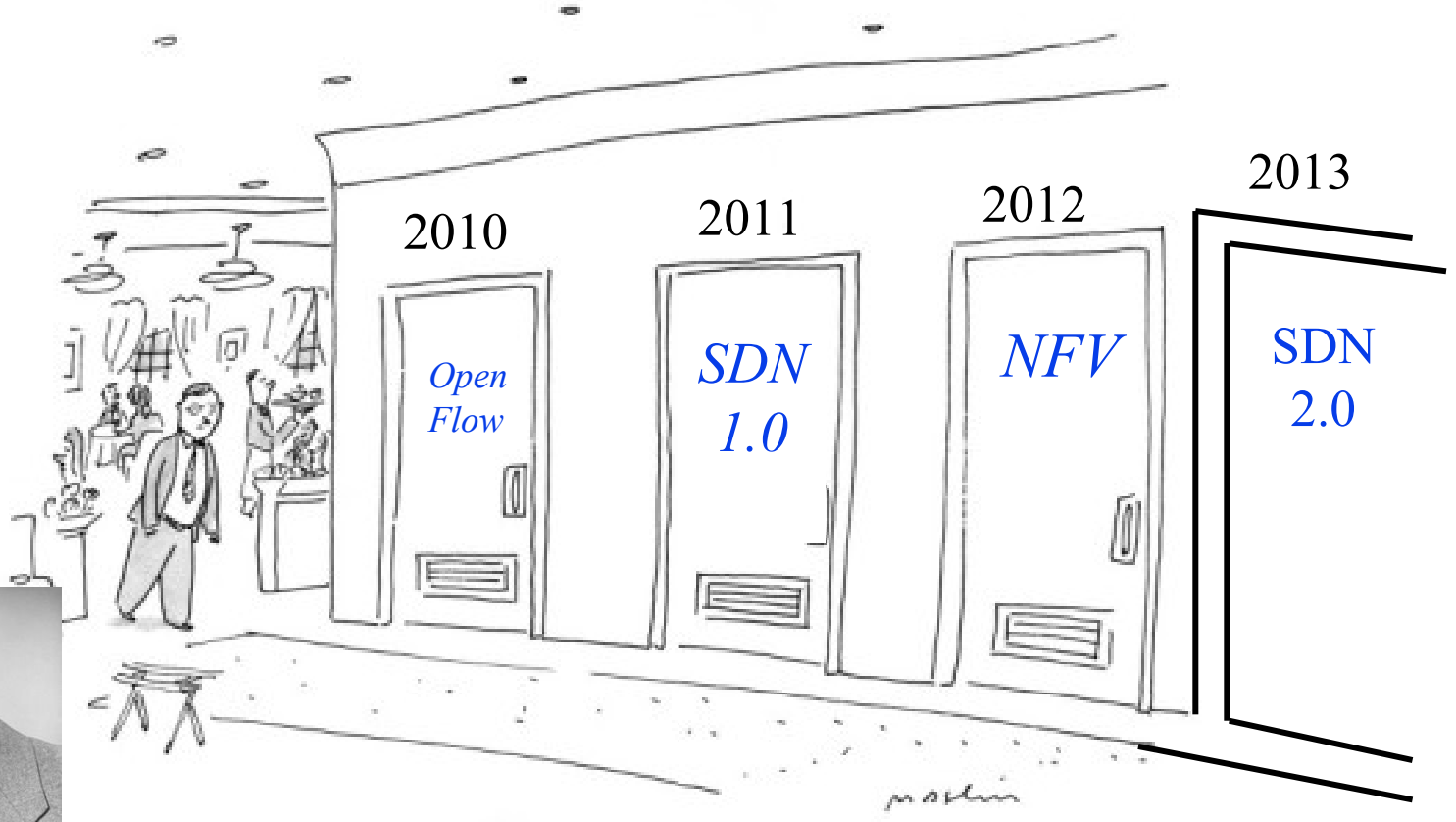
Separation of Control and Data Plane

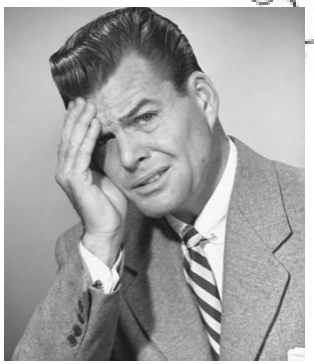


- Control logic is moved to a central controller
- Switches only have forwarding elements
- One expensive controller with a lot of cheap switches
- OpenFlow is the protocol to send/receive forwarding rules from controller to switches
- By programming the controller, we can quickly change the entire network behavior ⇒ **S**oftware **D**efined **N**etworking

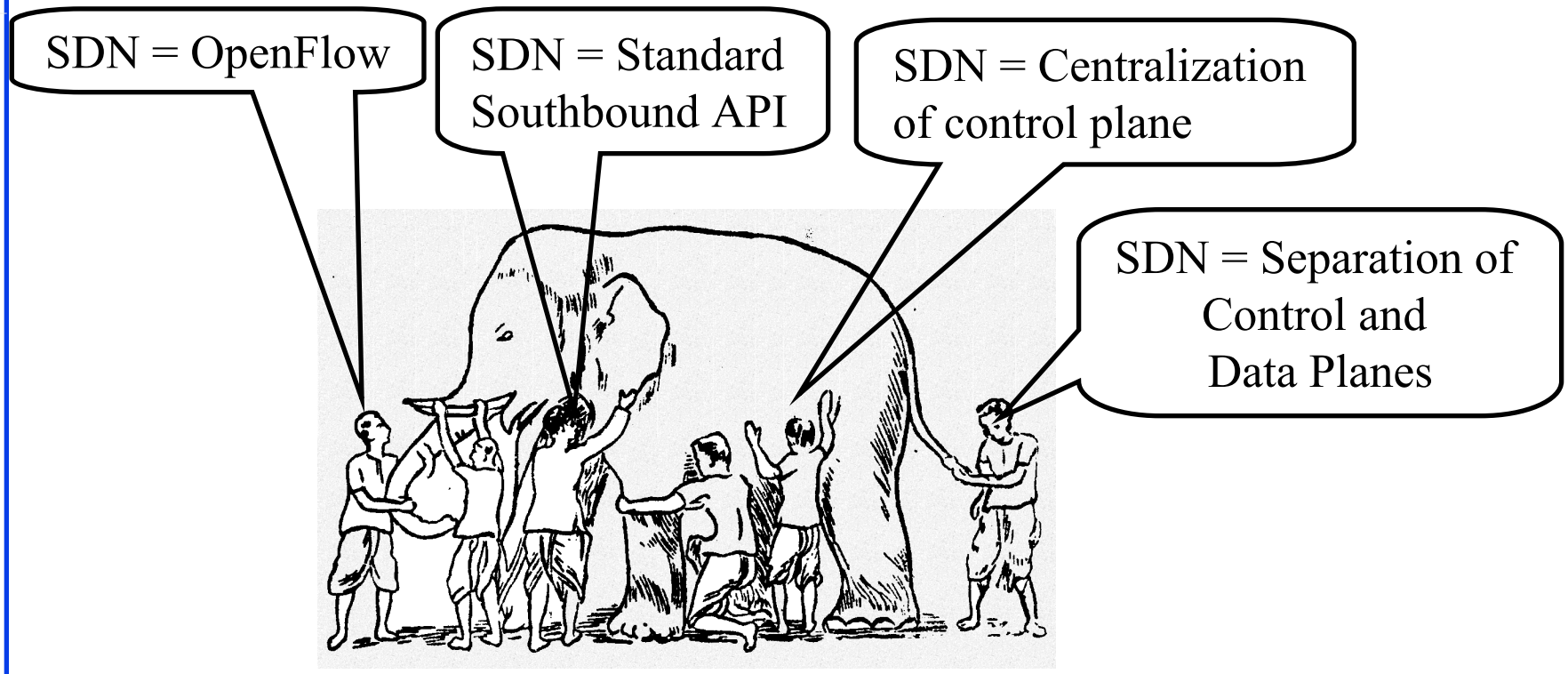
What have Telcos seen in the last 5 Years?

- A new future every year…





What is SDN?



- All of these are mechanisms.
- SDN is *not* about a mechanism.
- It is a framework to solve a set of problems ⇒ Many solutions

ONF Definition of SDN

"***What is SDN?***

The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices."

1. Directly programmable
2. Agile: *Abstracting control from forwarding*
3. Centrally managed
4. Programmatically configured
5. Open standards-based vendor neutral

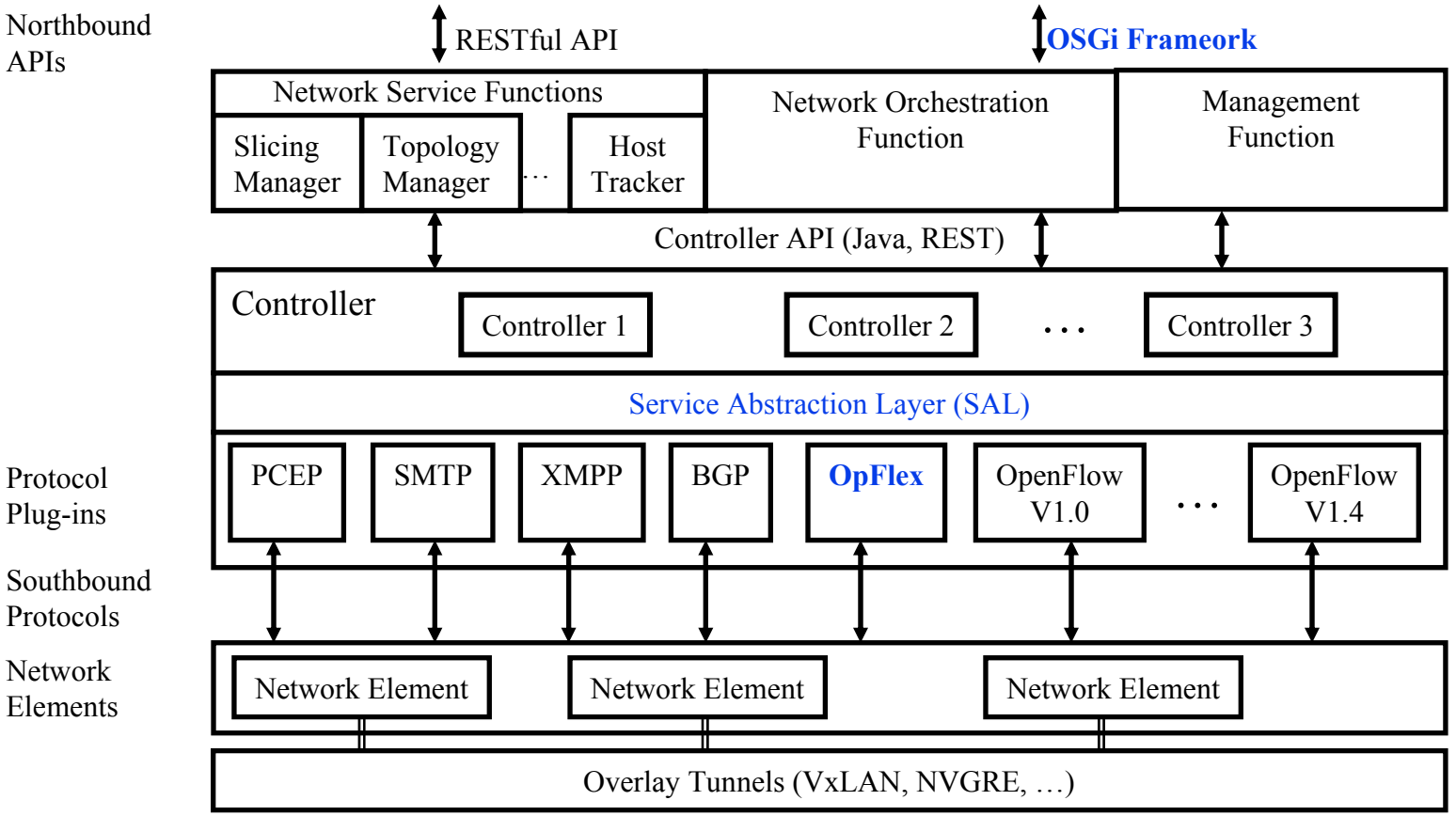
The above definition includes *How*.
Now many different opinions about ***How***.
⇒SDN has become more general.
Need to define by ***What***?

Ref: https://www.opennetworking.org/index.php?option=com_content&view=article&id=686&Itemid=272&lang=en

What do We need SDN for?

1. **Virtualization**: Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.
2. **Orchestration**: Manage thousands of devices
3. **Programmable**: Should be able to change behavior on the fly.
4. **Dynamic Scaling**: Should be able to change size, quantity
5. **Automation**: Lower OpEx
6. **Visibility**: Monitor resources, connectivity
7. **Performance**: Optimize network device utilization
8. **Multi-tenancy**: Sharing expensive infrastructure
9. **Service Integration**
10. **Openness**: Full choice of Modular plug-ins
11. **Unified management** of computing, networking, and storage

SDN 2.0: OpenDaylight Style SDN



- **NO-OpenFlow** (**Not Only** OpenFlow) Multi-Protocol
- New work in **IETF** XMPP, ALTO, I2RS, PCEP, ….
- Linux Foundation

Networking and Religion

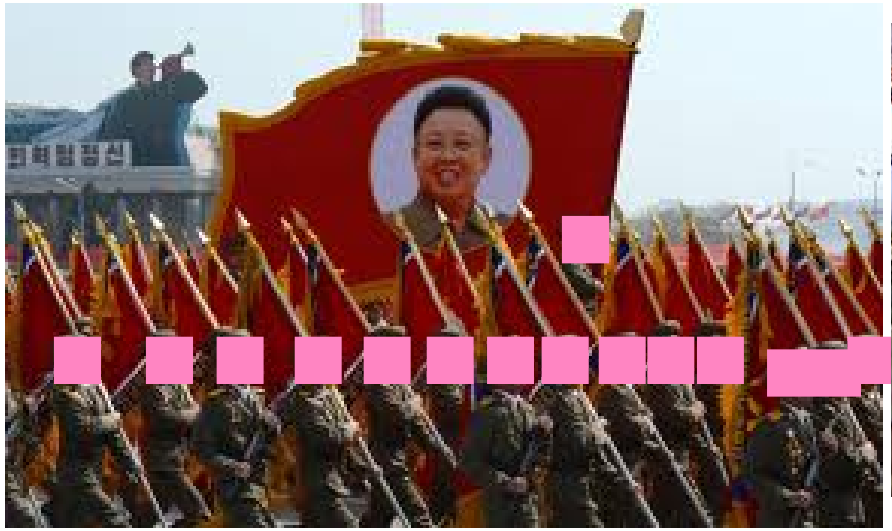
Both are based on a set of beliefs

Current SDN Debate: What vs. How?

- SDN is easy if control plane is centralized but not necessary. Distributed solutions may be required for legacy equipment and for fail-safe operation.
- Complete removal of control plane may be harmful. Exact division of control plane between centralized controller and distributed forwarders is yet to be worked out
- SDN is easy with a standard southbound protocol like OpenFlow but one protocol may not work/scale in all cases
  - Diversity of protocols is a fact of life.
  - There are no standard operating systems, processors, routers, or Ethernet switches.
- If industry finds an easier way to solve the same problems by another method, that method may win. E.g., ATM vs. MPLS.

Separation vs. Centralization

Separation of Control Plane

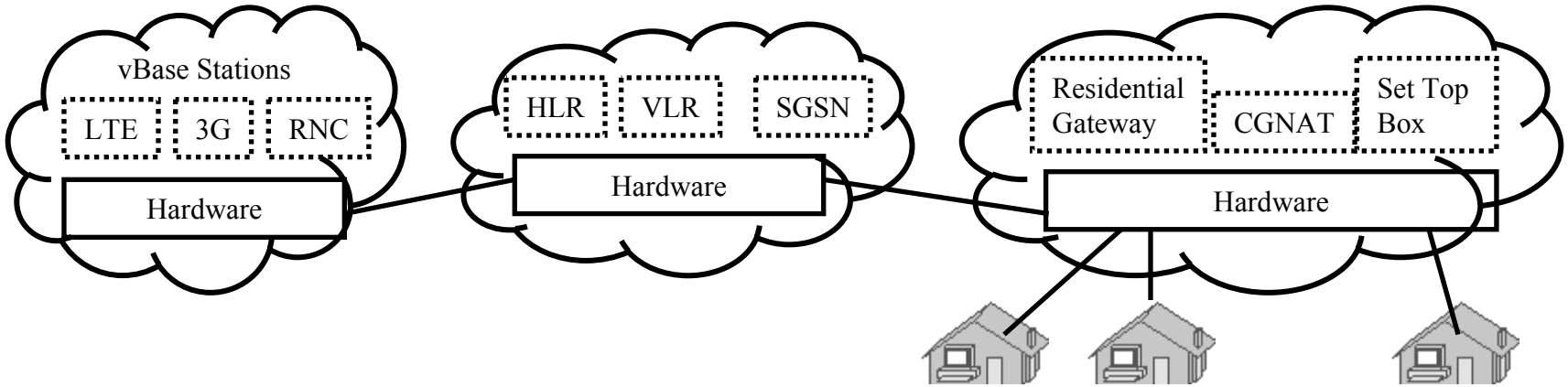


Centralization of Control

Network Function Virtualization (NFV)

1. Fast standard hardware ⇒ **Software based Devices**
   Routers, Firewalls, Broadband Remote Access Server (BRAS) ⇒ A.k.a. *white box* implementation

2. **Virtual Machine implementation**
   ⇒ Virtual appliances
   ⇒ All advantages of virtualization (quick provisioning, scalability, mobility, Reduced CapEx, Reduced OpEx, …)



Ref: ETSI, "NFV – Update White Paper," Oct 2013, http://www.tid.es/es/Documents/NFV_White_PaperV2.pdf (Must read)

What can NFV do?

1. **Virtualization**: Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.
2. **Orchestration**: Manage thousands of devices
3. **Programmable**: Should be able to change behavior on the fly.
4. **Dynamic Scaling**: Should be able to change size, quantity
5. **Automation**
6. **Visibility**: Monitor resources, connectivity
7. **Performance**: Optimize network device utilization
8. **Multi-tenancy**
9. **Service Integration**
10. **Openness**: Full choice of Modular plug-ins

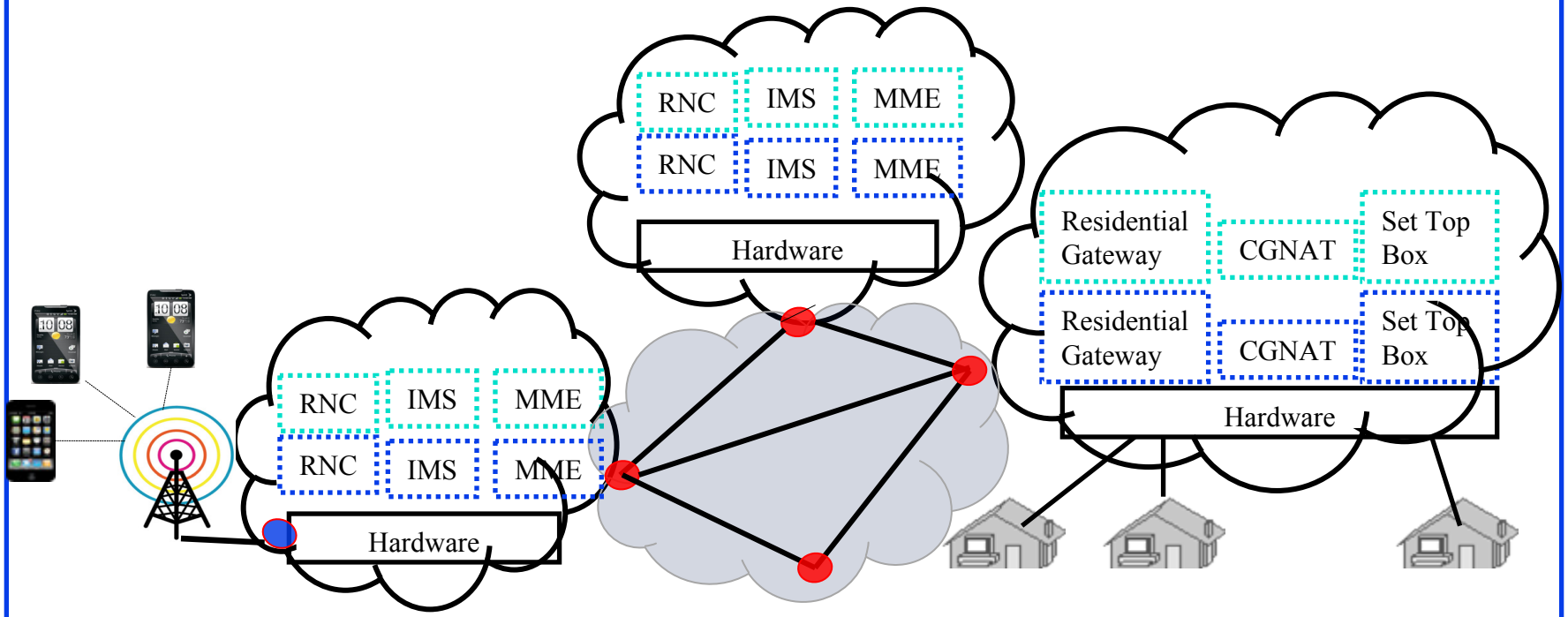
Note: These are exactly the **same** reasons why we need SDN.

Service-Infrastructure Separation

- With cloud computing, anyone can super-compute on demand.
  - Physical infrastructure is owned by Cloud Service Provider (CSP). Tenants get virtual infrastructure
  - **Win-Win** combination
- With virtualization, an ISP can set up all virtual resources on demand
  - Physical Infrastructure owned by NFV infrastructure service provider (NSP) and tenant ISPs get virtual NFVI services
  - **Win-Win** combination

Service Chaining in a Multi-Cloud Multi-Tenant Environment

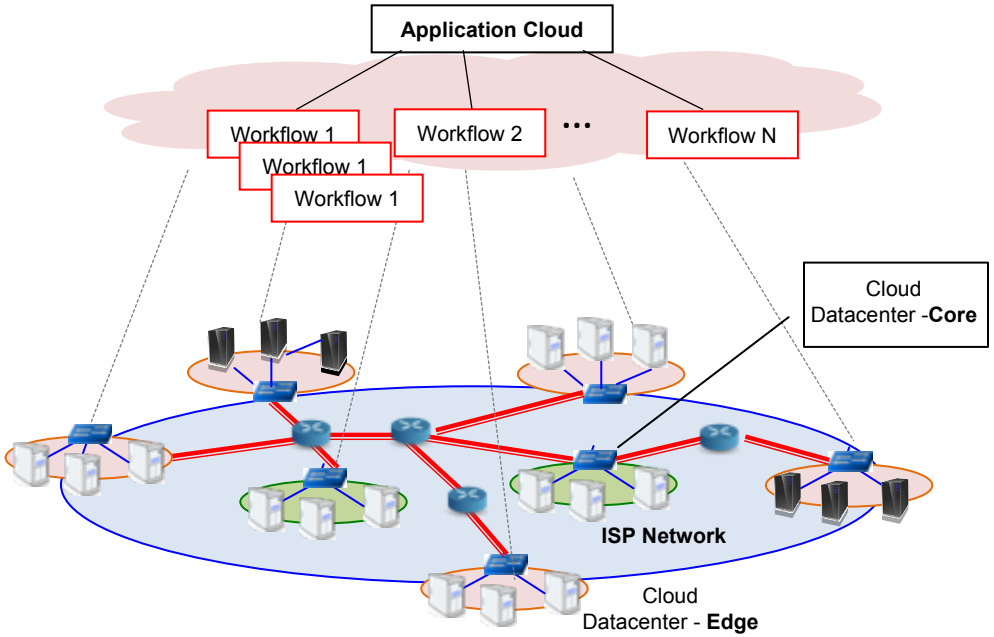
- VNFs (Virtual network fns) belong to tenants. Multiple tenants.
- Each Cloud belongs to a different Cloud Service Provider (CSP)
- Internet infrastructure belongs to an NFVI service provider (NSP)
- Service chain = Workflow



Any Function Virtualization (FV)

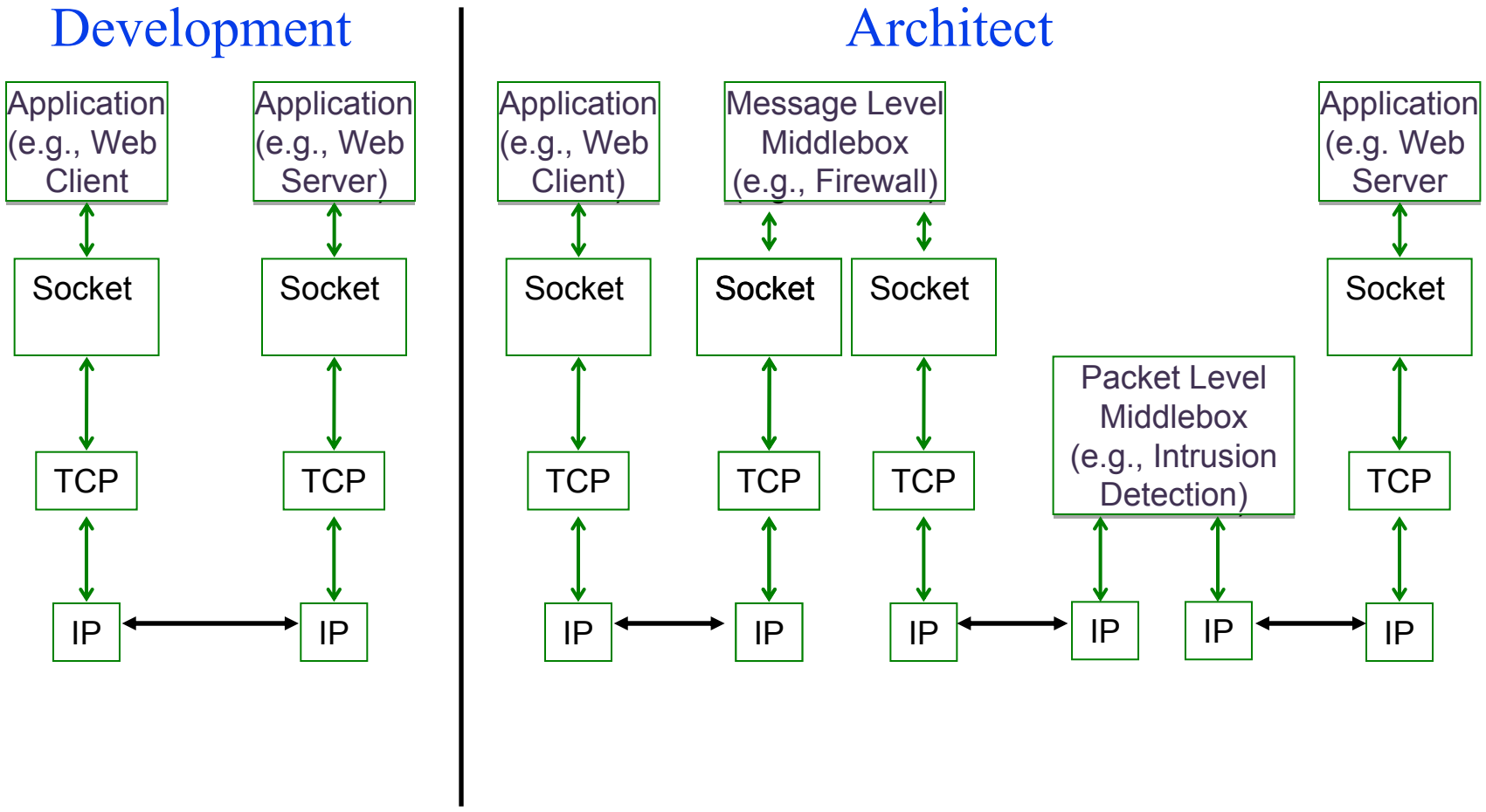
- Network function virtualization of interest to Network service providers
- But the same concept can be used by any other industry, e.g., financial industry, banks, stock brokers, retailers, mobile games, …
- Everyone can benefit from:
  - Functional decomposition of there industry
  - Virtualization of those functions
  - Service chaining those virtual functions (VFs)
    ⇒ A service provided by the next gen ISPs

Services in a Cloud of Clouds

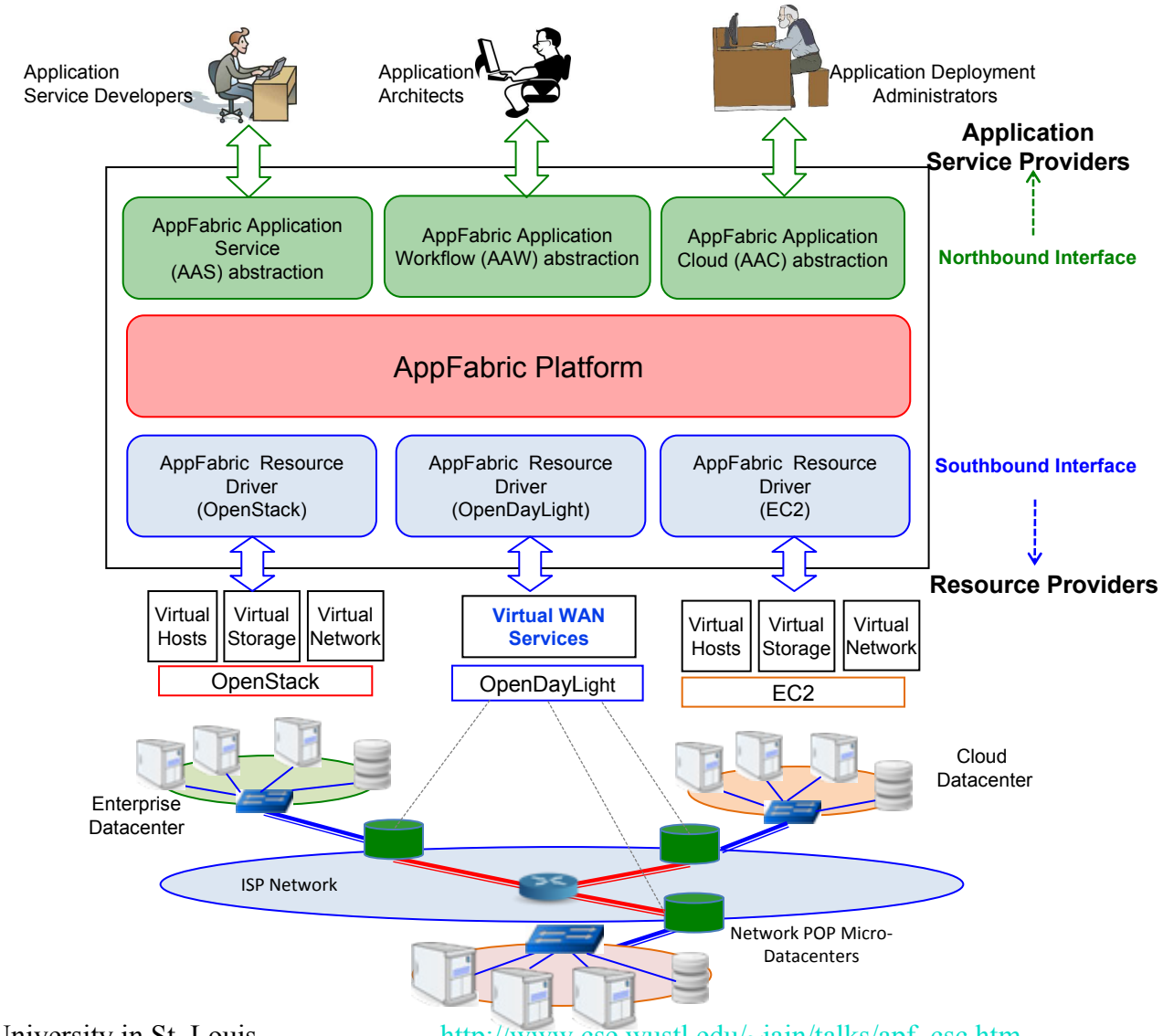


- Need to add/delete workflows as the load/locality changes
- Application **Developers** designing a workflow need not be aware of middleboxes, physical resources.
- Application **Architects** set guidelines for creation of new workflows including middleboxes
- **Deployment Administrators** set policies for quantity and location of resources inside various clouds.
- Also need to virtualize **wide area networks** just as it is done inside the clouds today

Workflow Example



Services in a Cloud of Clouds

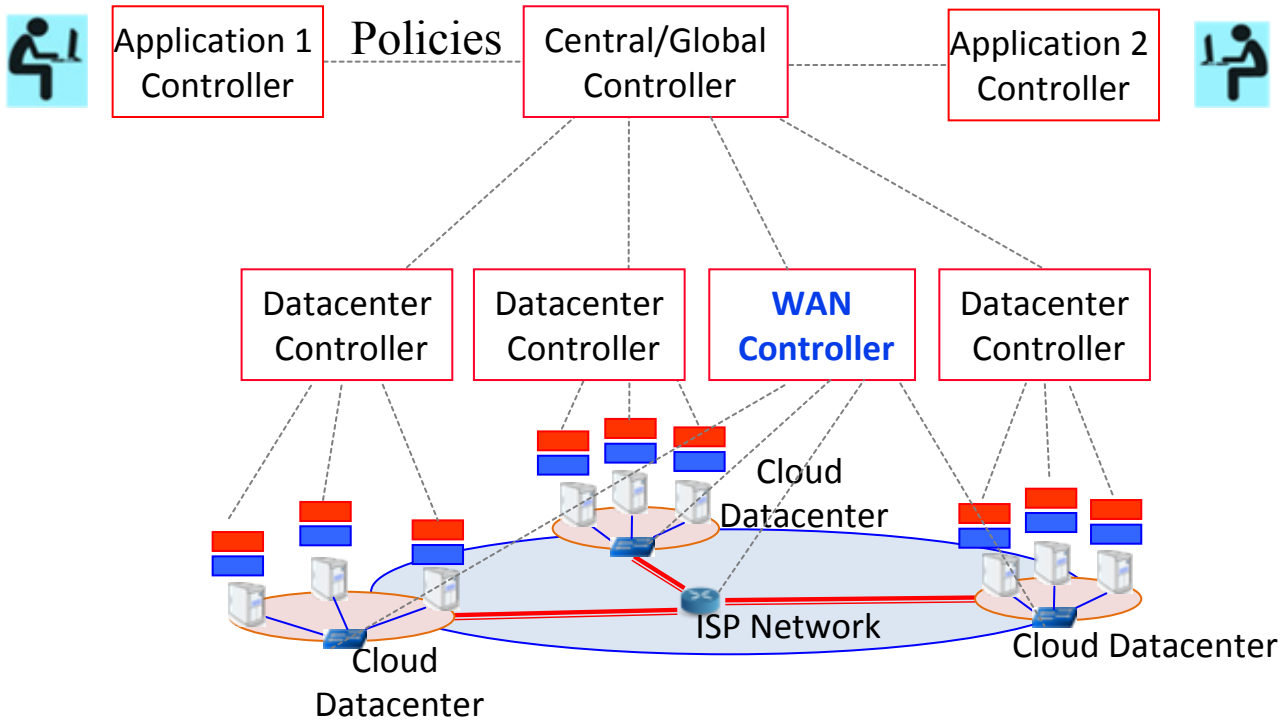


AppFabric Features

- Allows **application architects** to specify guidelines for creation of new workflows including middleboxes
- Allows **application developers** to specify their resource requirements and design their application without worrying about physical infrastructure
- Allows **Deployment Administrators** specify policies for quantity and location of resources inside various clouds.
- **Automates** the entire process of creating new workflows and installing them, managing them during runtime, uninstalling them as necessary
- Workflow creation includes virtual networks, computers, storage inside the clouds as well as the network between the clouds
- **WAN bandwidth** and latency is the key to placement. Allows manual approval and override.
- All interfaces initially XML based. GUI based in future.

Hierarchy of Controllers

- Controllers exchange and implement policies



Resource Control

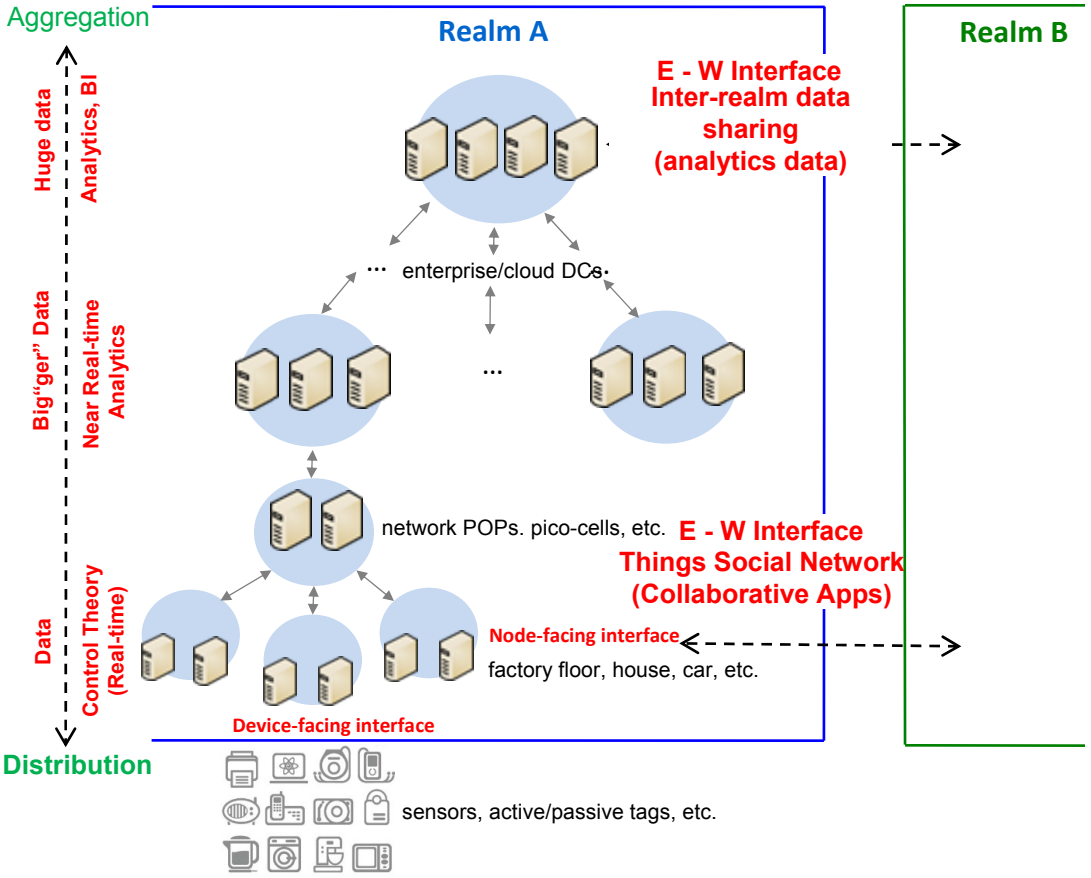
- Tenants keep complete control of their data.
  NSP does not have to look at the **application data** to enforce application level policies
- NSPs keep complete control of their equipment.
  tenants communicate their policies to NSP's control plane
- **VFs and Middle boxes** can be located anywhere on the global Internet
  (Of course, performance is best when they are close by)
- Tenants or NSPs can **own** OpenADN modules.
  NSPs can offer "Service Chaining" **service**

Challenges in Service Chaining

- **Dynamic**:
  - Forwarding changes with state of the servers, links, …
  - Cloud operators may want to move VMs themselves for security, reliability, performance, or in anticipation of load changes.
- **QoS vs. Cost**: Latency determined by link utilization. WAN links expensive. Need to keep the utilization high.
- **Content sensitive**:
  - Different for different types of videos, read-writes, …
- **Distributed Control**:
  - Equipment belongs to infrastructure provider
  - Data belongs to Tenants
- **Massive Scale**:
  - Billions of Users with different user context
- **Stateful Services**:
  - All packets of a flow should be sent to the same replica
    - Message level services (firewalls),
    - Packet level services (intrusion detection)

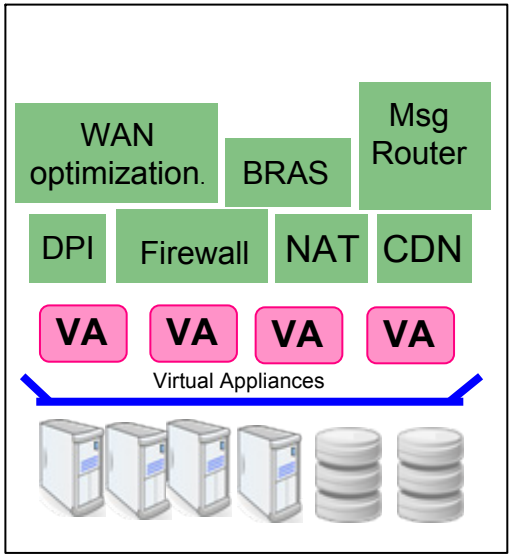
Use Case 1: Internet of Things

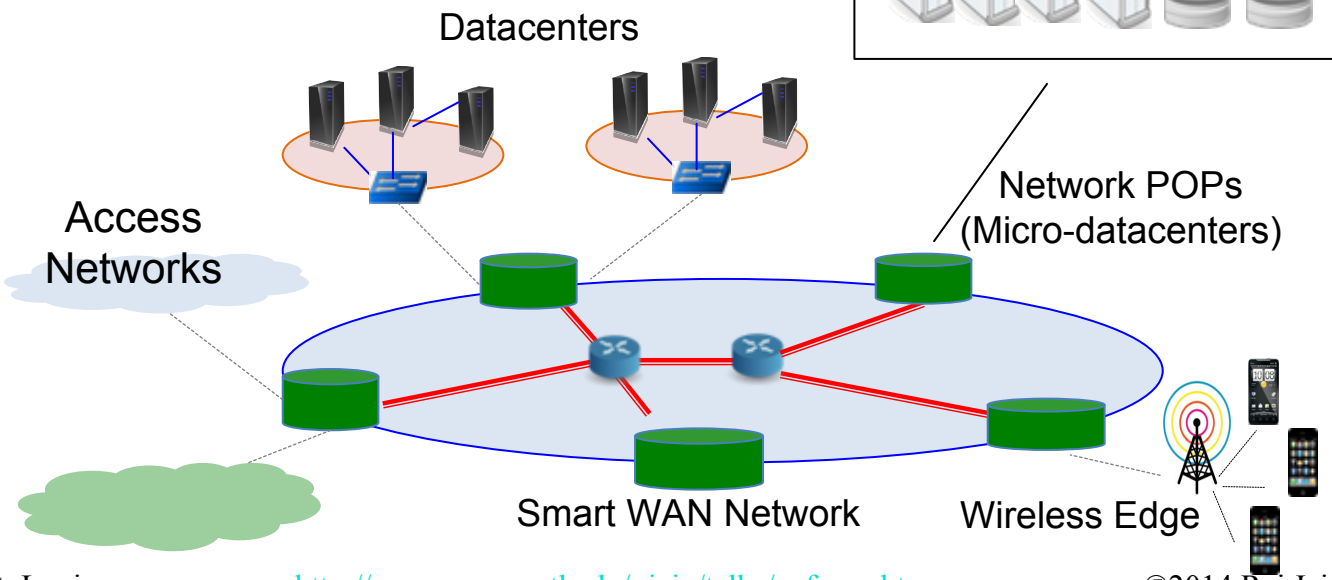
- Computing everywhere or Fog computing.
- Data aggregation and analytics at various levels



Use Case 2: Smart WANs

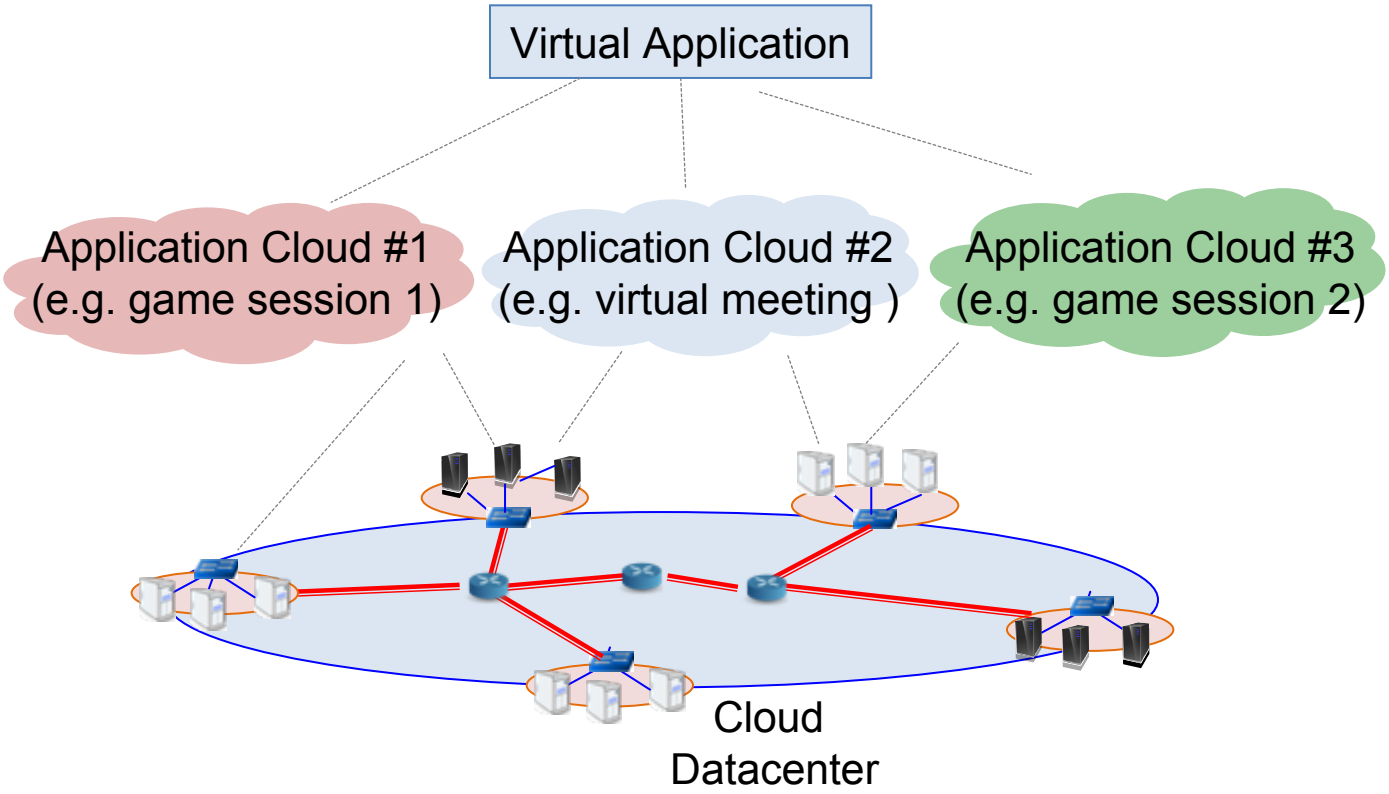
- Service Chaining
- Message-level Middleboxes
- Packet-level Middleboxes
- Dynamically place services at POPs based on application topology
- Provide differentiated transport
- Contextual mapping of messages to transport QoS

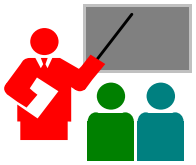




Use Case 3: Massively Distributed Apps

- Online games





Summary

1. Virtual Networking Functions (VNFs) will be replicated and deployed globally
   ⇒ Need **dynamic service chaining** based on user, network, and application context
2. Virtual functions useful not only for networking but also for **all other global enterprises** and games
   ⇒ New business opportunity for FV Infrastructure service
3. **AppFabric** allows customers to select **multiple clouds** from different providers and **share wide area network** infrastructure and specify their policies
4. **WAN link capacity, utilization, and latency** are key to the placement of VMs.
5. NSPs keep complete **control** over their resources.
   Tenants keep complete control over their traffic.

References

- Raj Jain and Subharthi Paul, "**Network Virtualization and Software Defined Networking for Cloud Computing - A Survey**," IEEE Communications Magazine, Nov 2013, pp. 24-31, http://www.cse.wustl.edu/~jain/papers/net_virt.htm