VNF Meets the Cloud
Challenges and Solutions

Matti Hiltunen, Ph.D.

Principal Inventive Scientist
Cloud Technologies and Services Research
AT&T Labs – Research
Bedminster, NJ, USA

May 31 2016

Outline

- AT&T: NFV and SDN
- NFV Approach
- NFV Challenges
- Some solutions
  - CloudQoS Project at AT&T Labs - Research
    - Valet Holistic Placement Service
  - Looking forward
- Conclusions

AT&T: NFV and SDN

A Network Built In Software

Using Software to Put Customers in Control

AT&T reveals details of ECOMP SDN platform

AT&T Reveals Audacious SDN Plans

AT&T will launch SDN service in 63 countries simultaneously this year, de la Vega says

ONOS and AT&T Team Up to Deliver CORD

AT&T Software Defined Networking

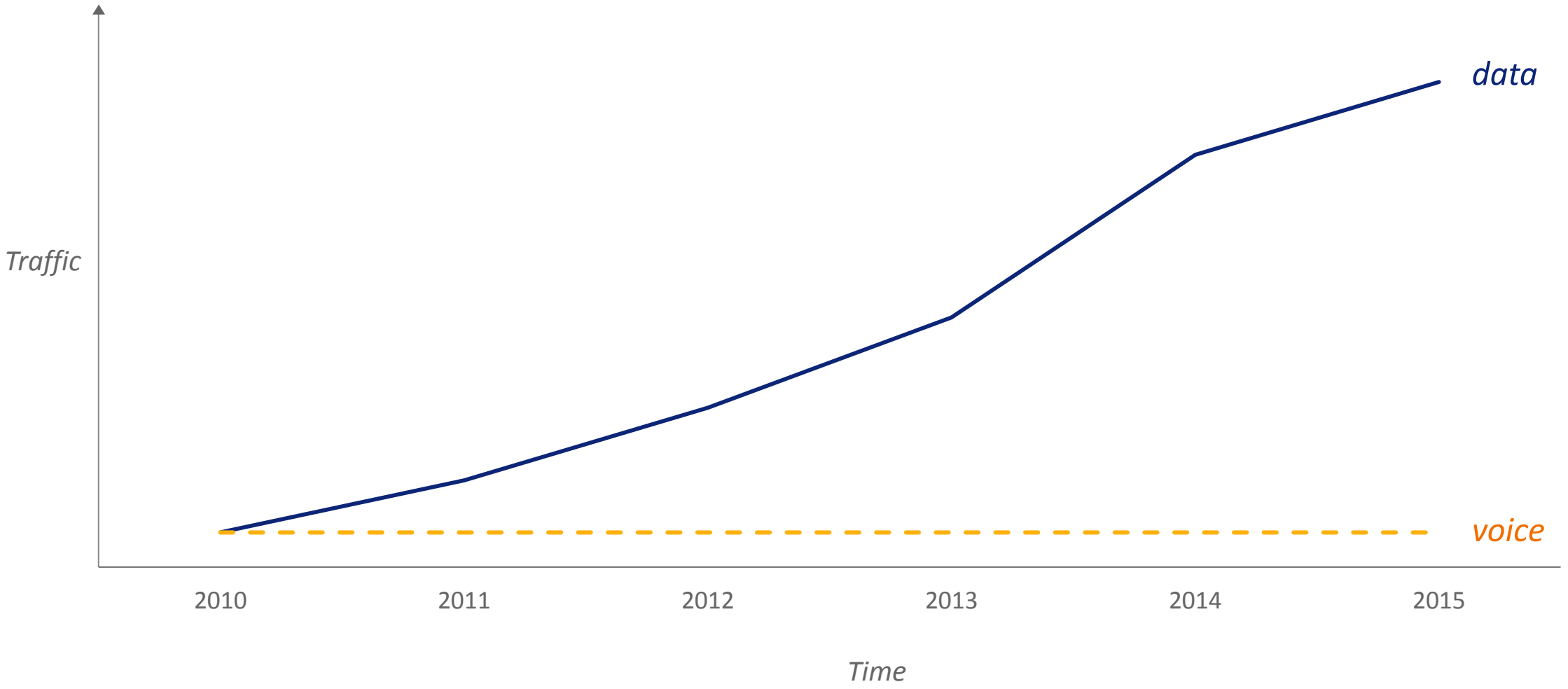
AT&T shows how serious it is about SDN, NFV and open source

AT&T Puts SDN/NFV in Driver's Seat

AT&T accelerates SDN vision by extending network on demand capability to 100 cities

AT&T Shares ECOMP Vision, Might Share Software

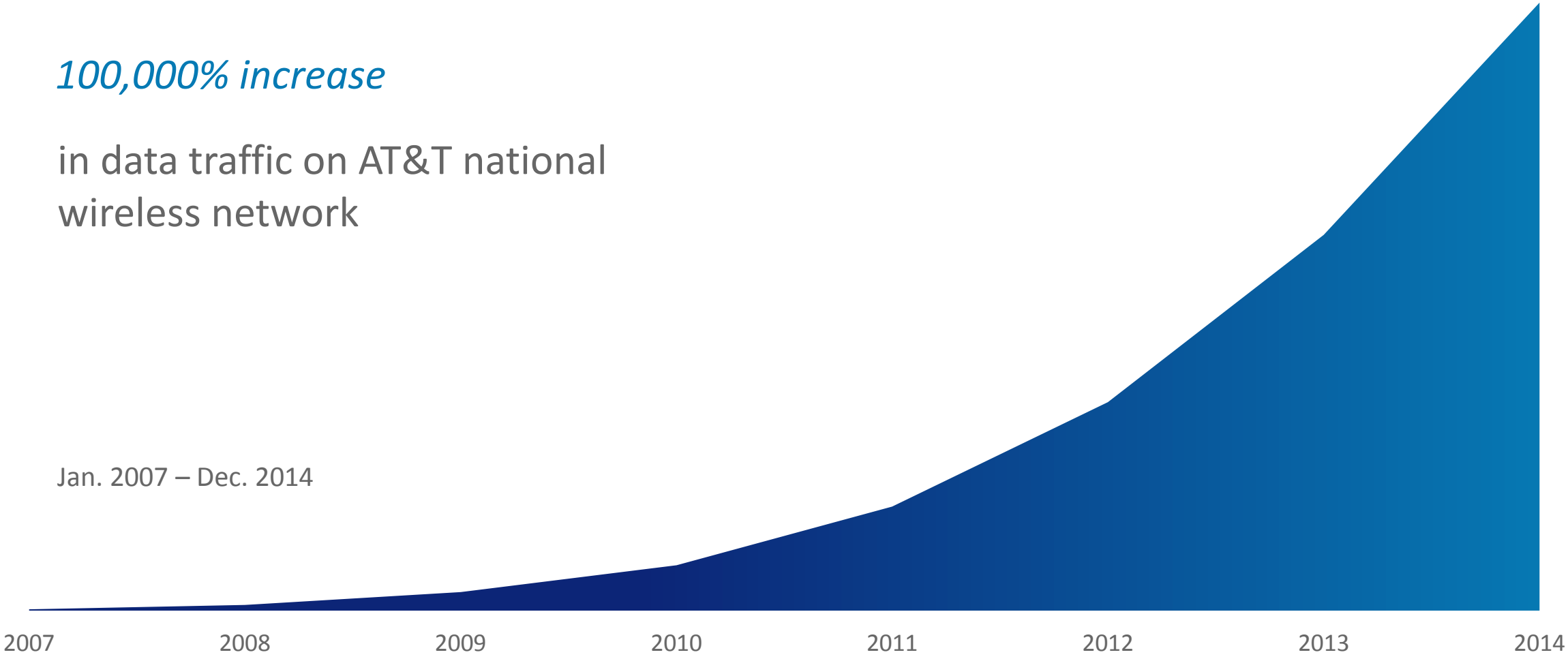
Why: Mobile vs. Data Growth



Why: Customer Demand

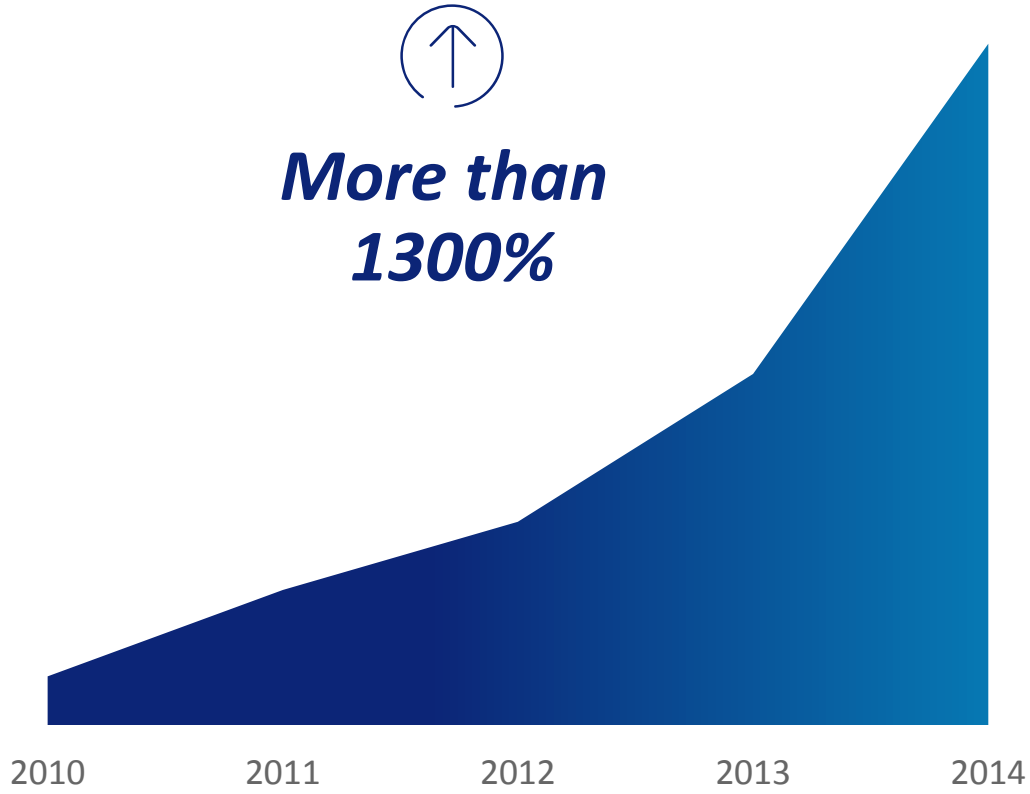
100,000% increase

in data traffic on AT&T national wireless network



Additional Growth Drivers

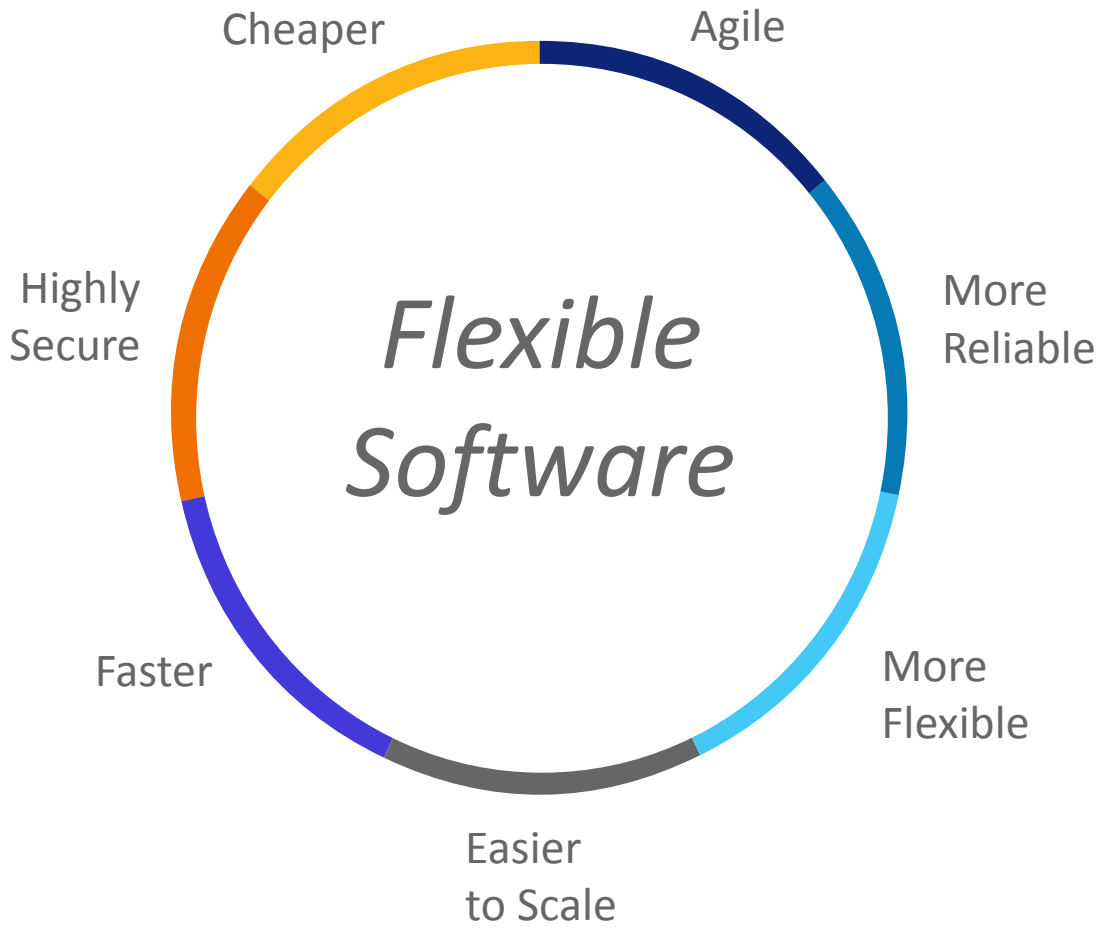
Mobile Video Traffic



Ethernet Ports

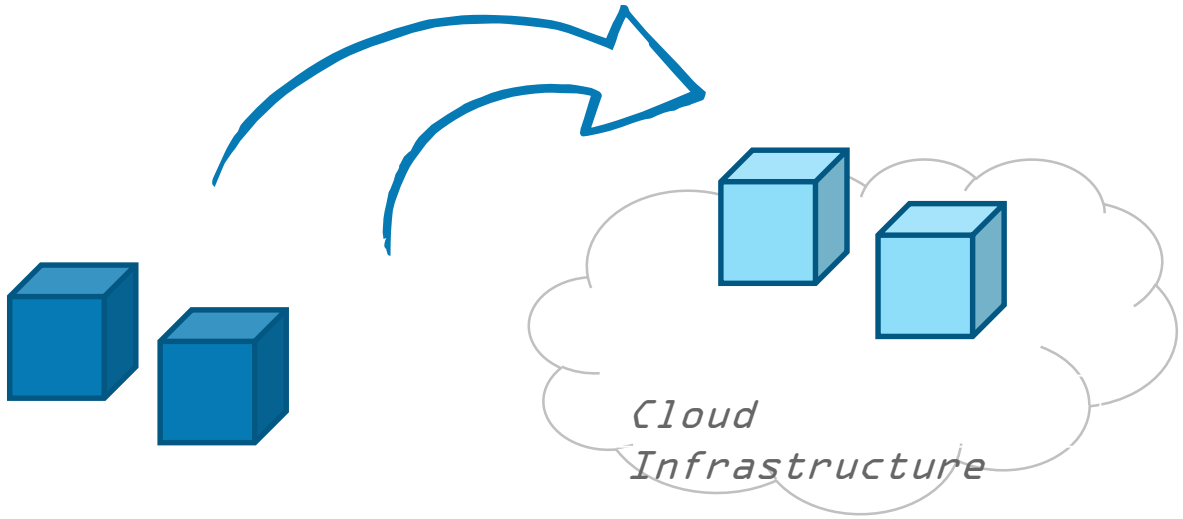
The Vision

Specialized Hardware



Solution

Network Function Virtualization



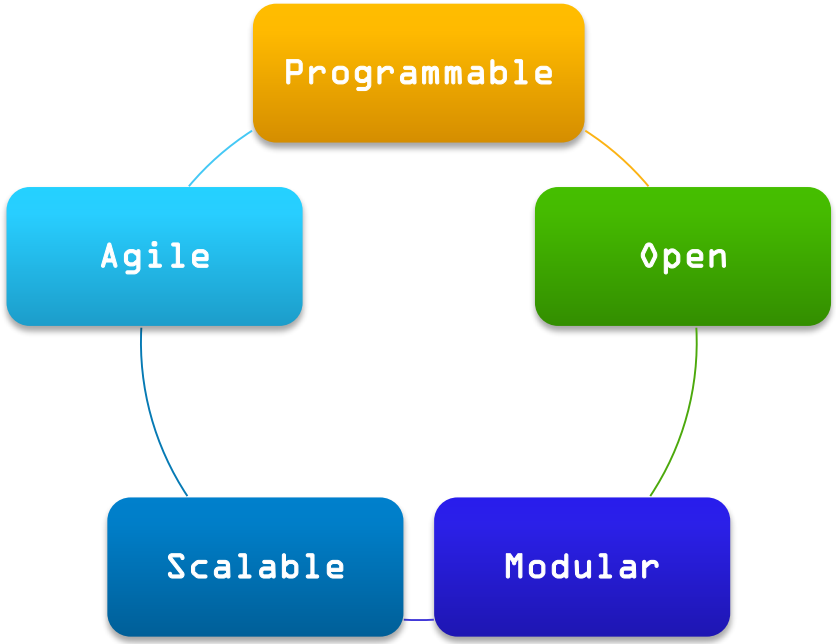
Physical functions become virtual functions

Applications evolve fast

Greater reuse & distribution

Elastic

Software Defined Networking

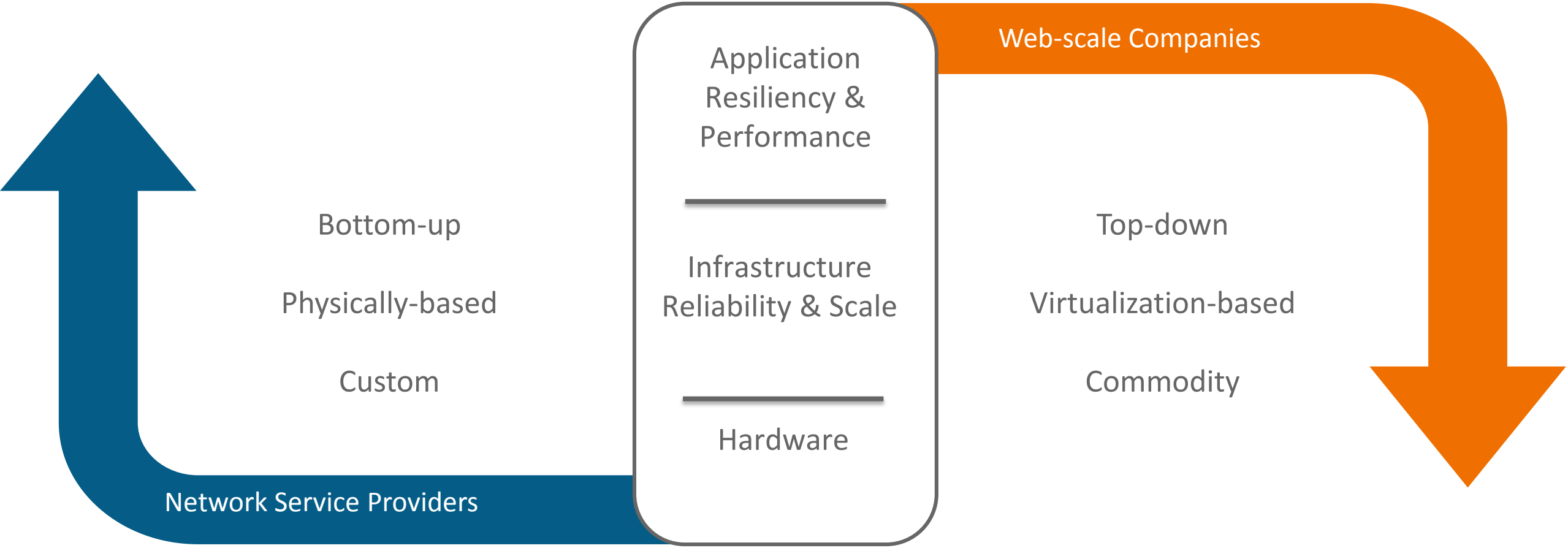


Intelligent – autonomous & automated

Reliable & secure – greater extensibility & control

Accelerates innovation – lower entry barrier, faster cycles

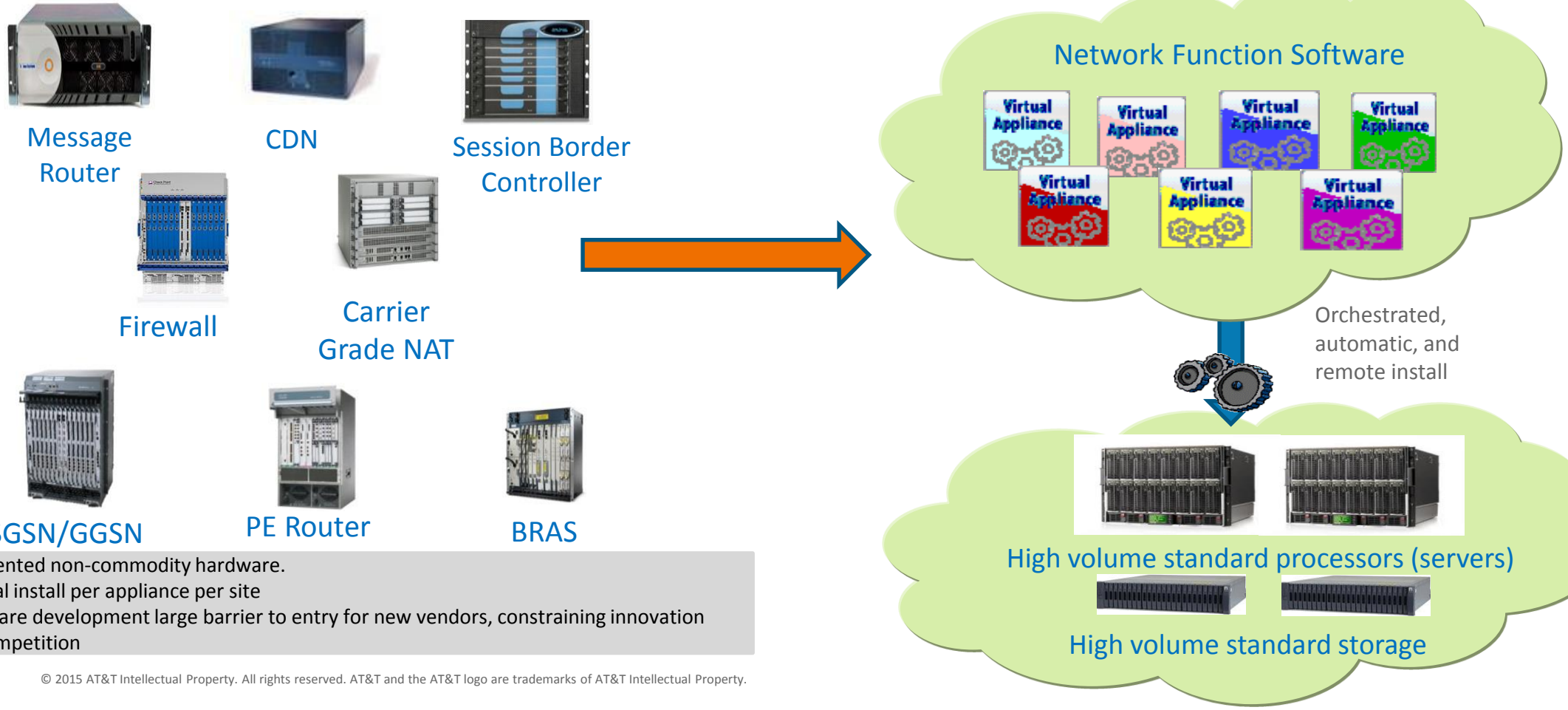
Shift: Adopt the Speed and Scale of the Web Platform Industry



NFV APPROACH

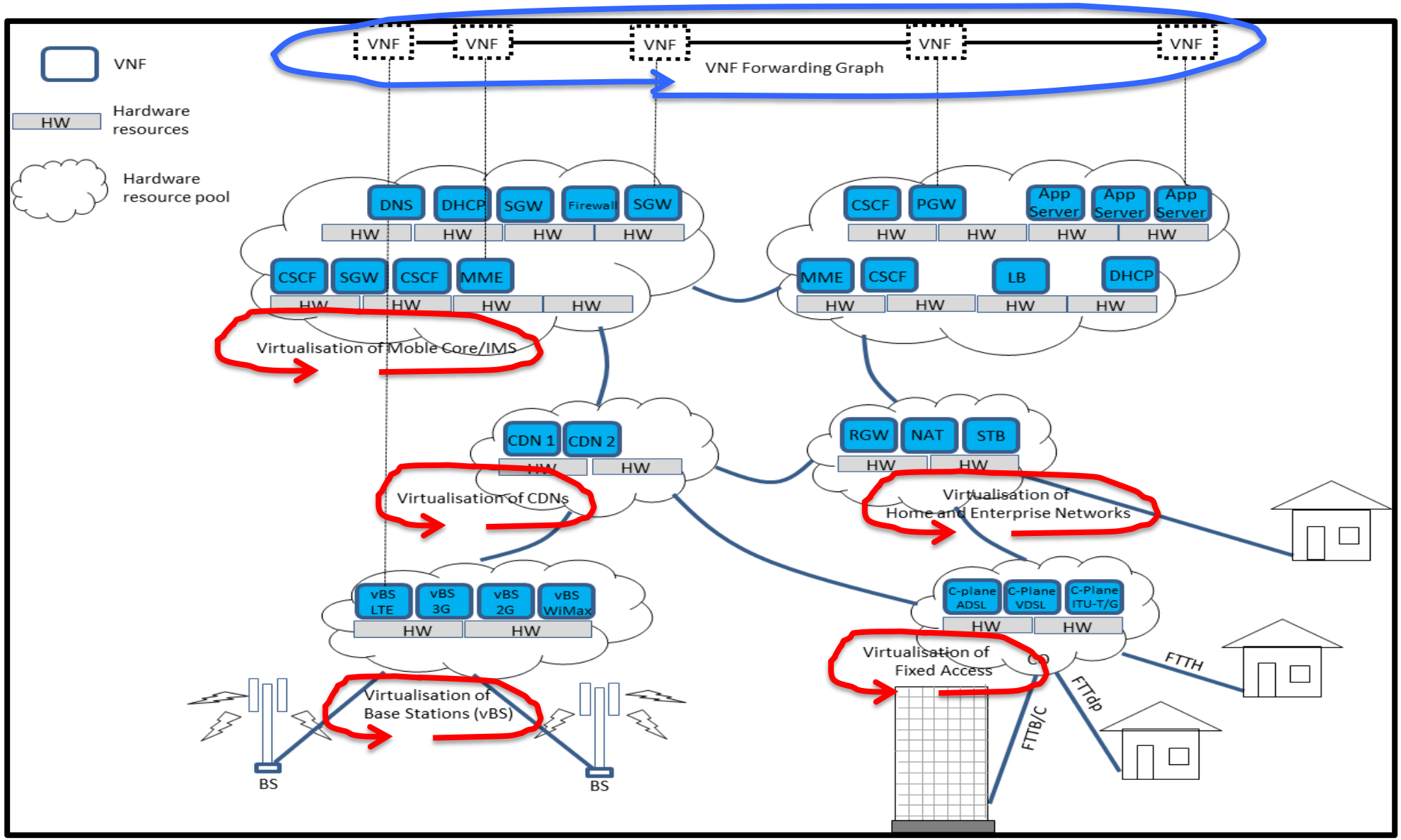
Network Virtualization Approach

European Telecommunications Standards Institute (ETSI) NFV Industry Specification Group (ISG), NFV White Paper (WP), Sept. 2014.



- Fragmented non-commodity hardware.
- Physical install per appliance per site
- Hardware development large barrier to entry for new vendors, constraining innovation and competition

NFV Use Cases



From ETSI NFV ISG WPU

Benefits of NFV (Examples)

Reduced equipment costs and power consumption through consolidation.

Improved operational efficiency thanks to higher uniformity of the physical platform.

Tailored services and connectivity for multiple users and applications on the same hardware.

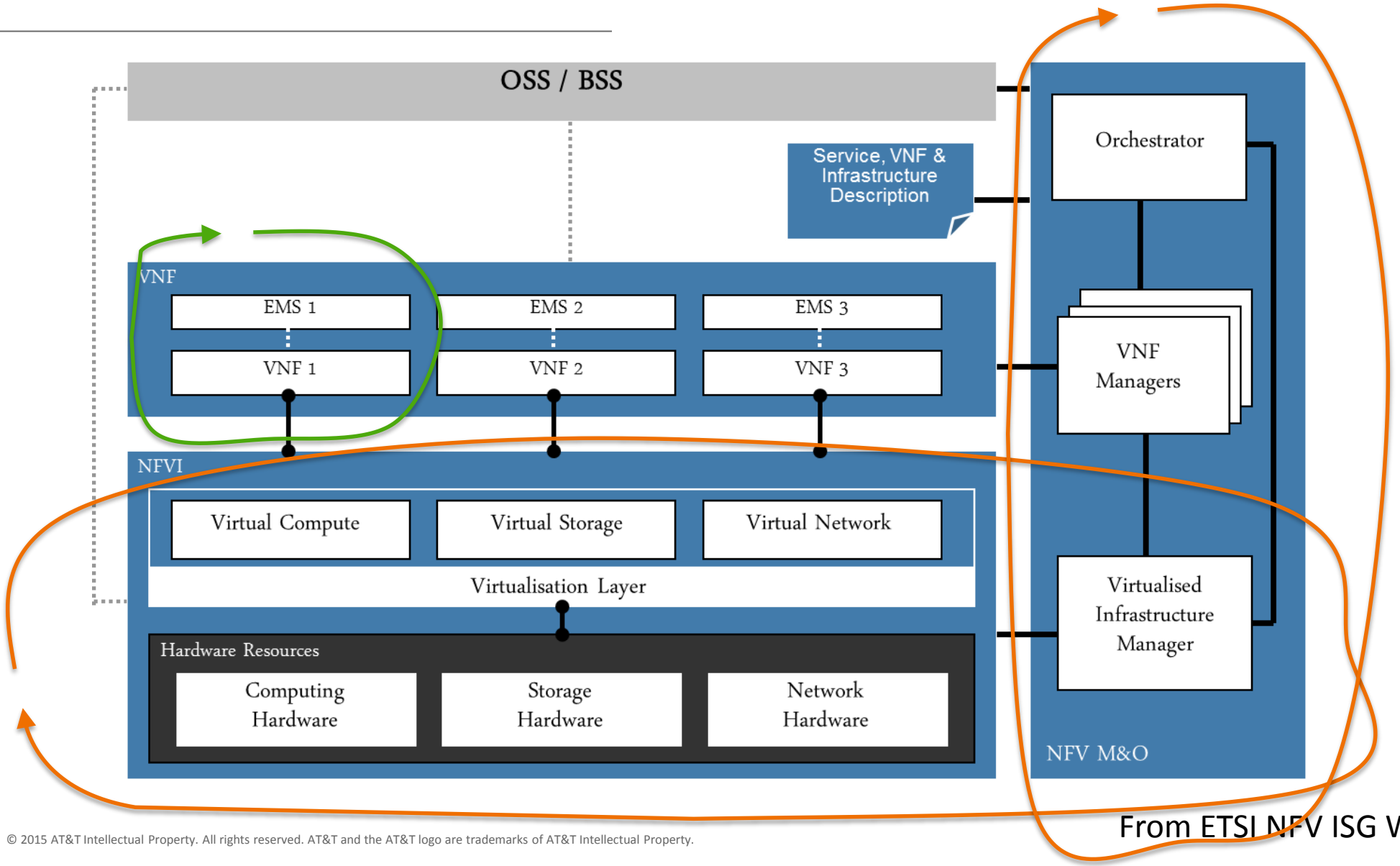
Targeted, very fast service introduction based on geography or customer sets.

Services can be rapidly scaled up/down as required.

Enabling a wide variety of eco-systems and encouraging openness.

From ETSI NFV ISG WP

Architectural Framework

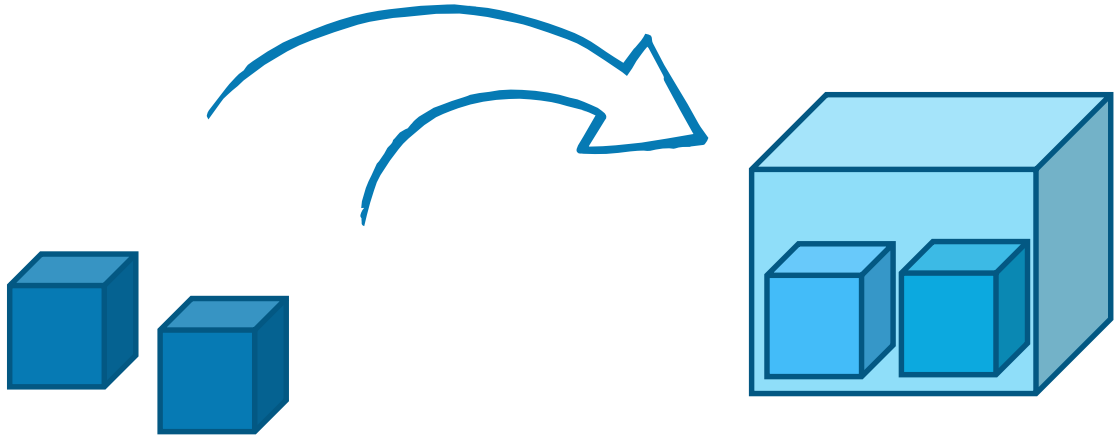


From ETSI NFV ISG WPU

NFV CHALLENGES

Challenges

1. Virtualization



VNF Provider:

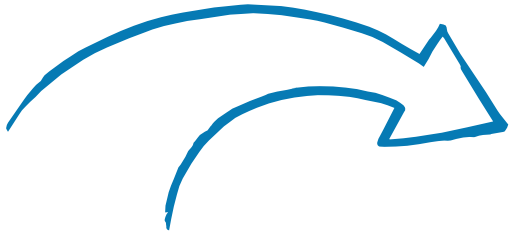
Commodity hardware and OS

Reliability

Virtualization overhead

Performance

2. “Cloudification”



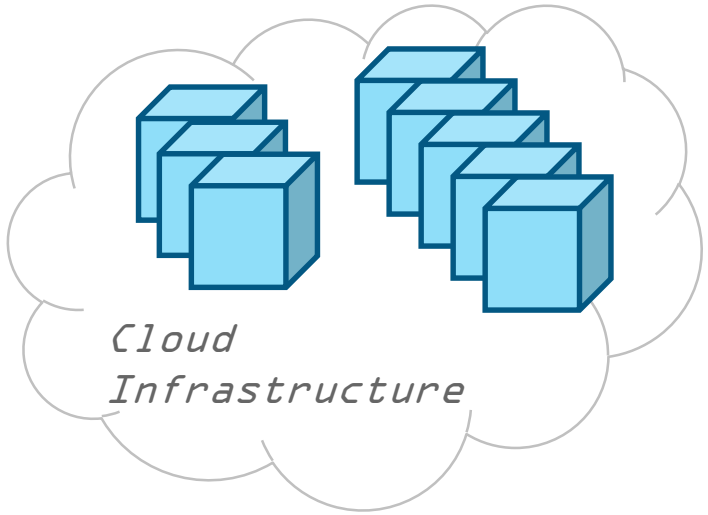
VNF Provider:

Unpredictability

Reliability

Security

Elasticity required



Cloud Provider:

Multi tenancy

QoS: Reliability, Performance, Security, ..

Dealing with scale

High utilization

Quality of Service (QoS)

A wide variety of non-functional requirements (e.g. throughout, latency, reliability, availability, security) that affect the quality of a service

Compute/Memory

Storage

Network

Requirements

ETSI GS NFV 004 V1.1.1 (2013-10)

Network Functions Virtualisation (NFV);
Virtualisation Requirements

Disclaimer

This document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Contents

Requirements

Contents

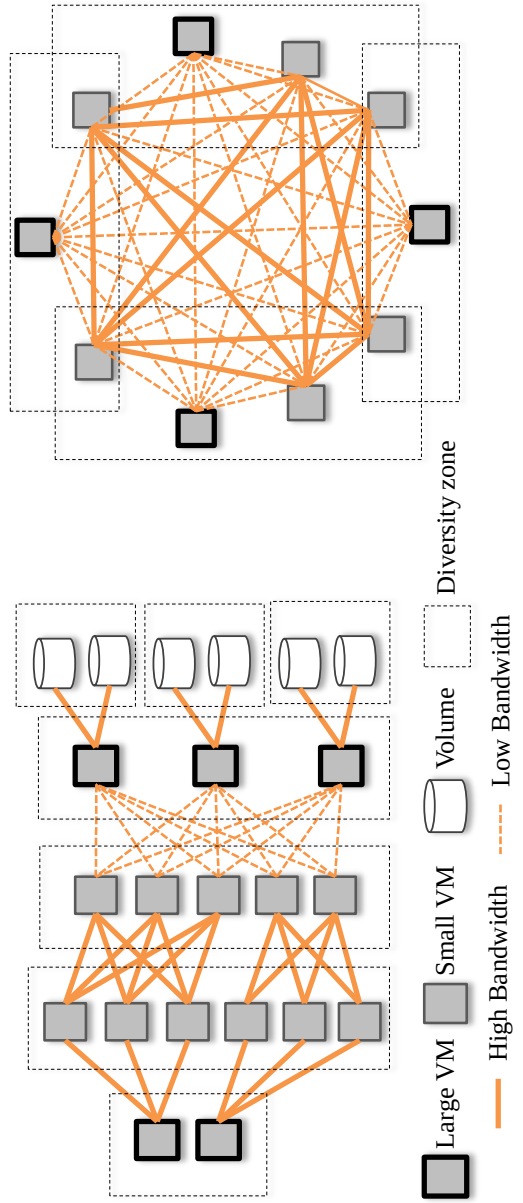
Requirements (Examples)

[Per.1] The NFV framework shall be able to instantiate and configure any given VNF over the underlying infrastructure so that the behavior…in terms of performance is conforming to the requirements expressed in the VNF information model….

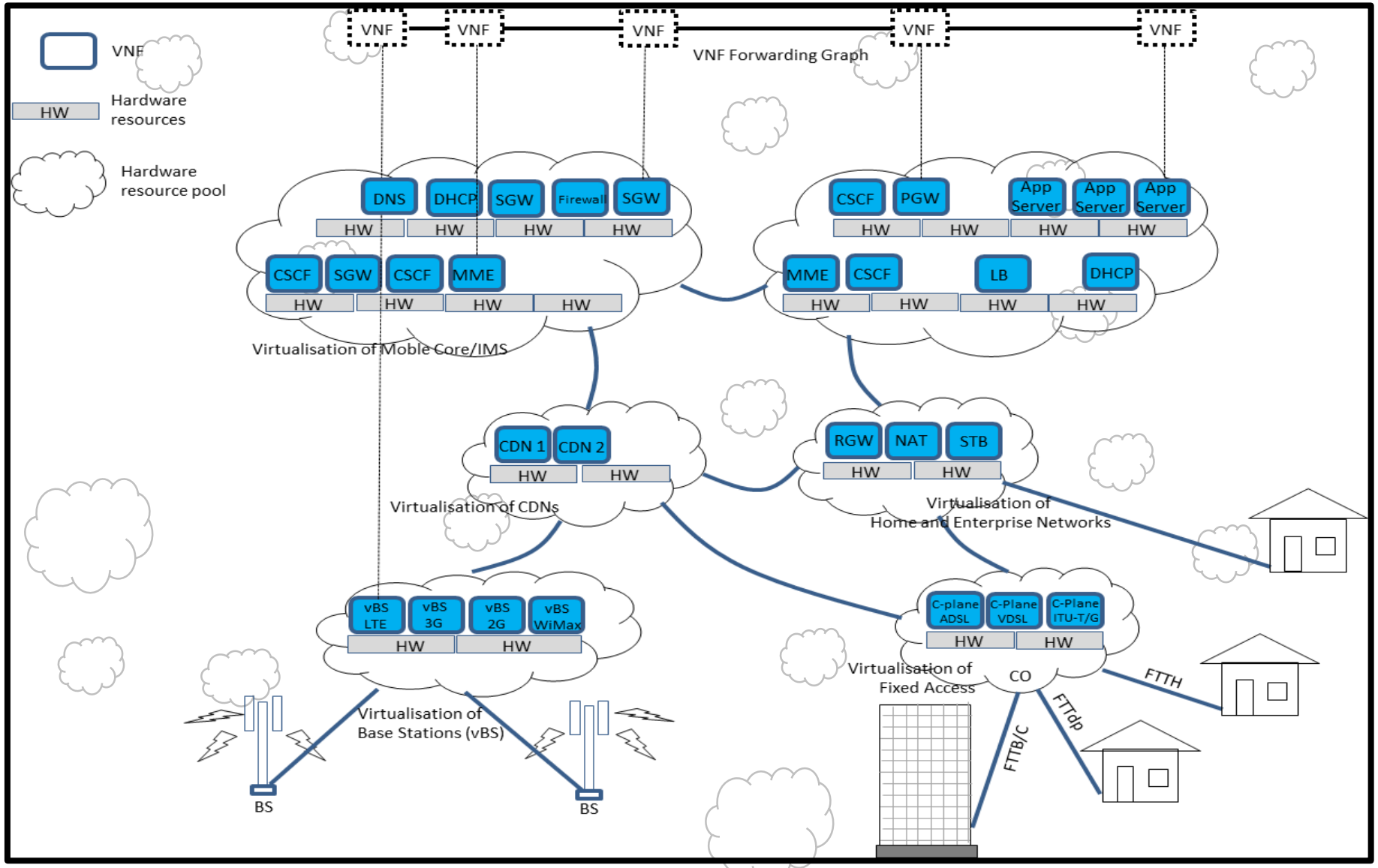
….

[OaM.14] The NFV framework shall be able to manage the assignment of NFVI resources to a VNF in a way that resources (compute hardware, storage, network) can be shared between VNFs.

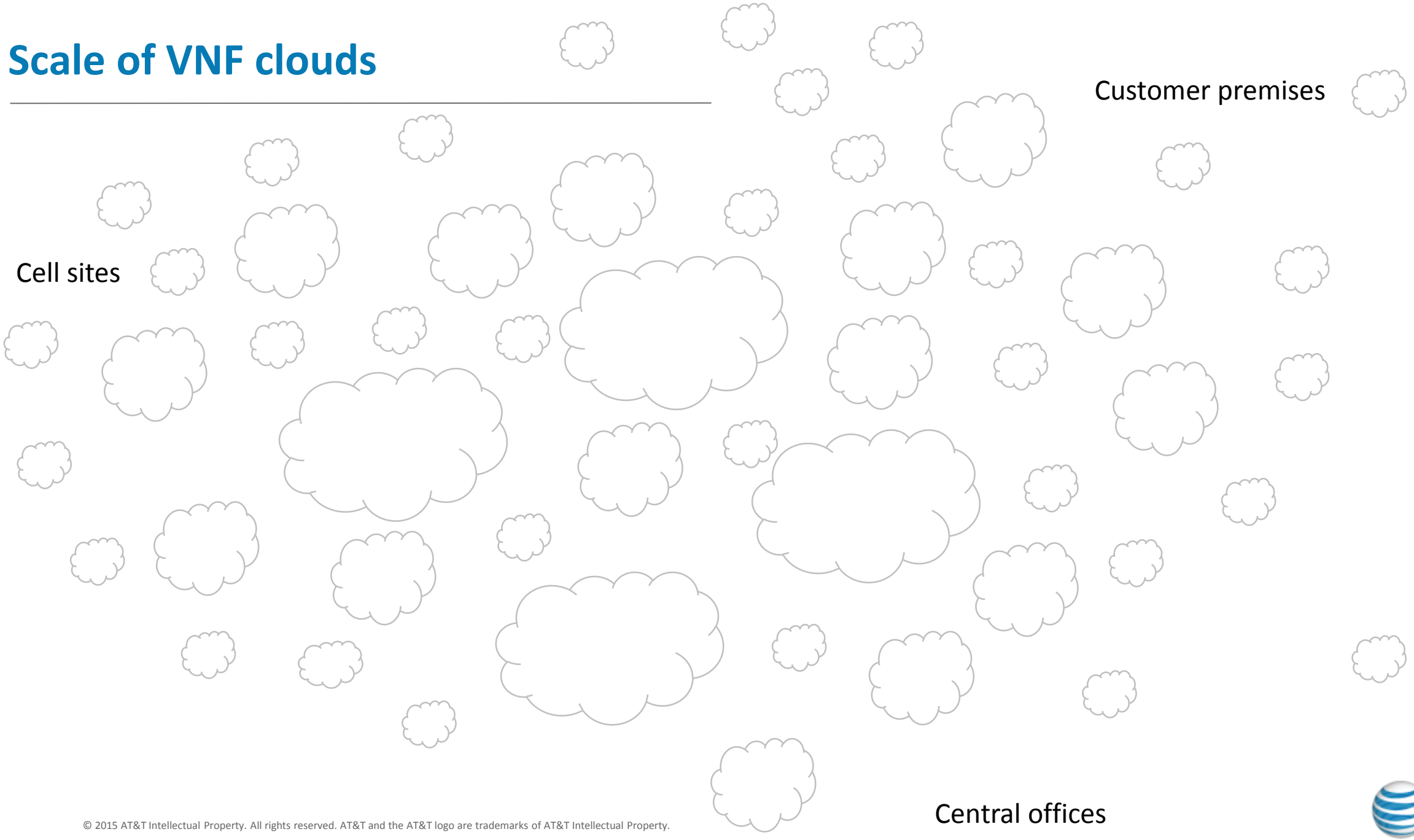
From ETSI NFV ISG WPU



Scale of VNF clouds



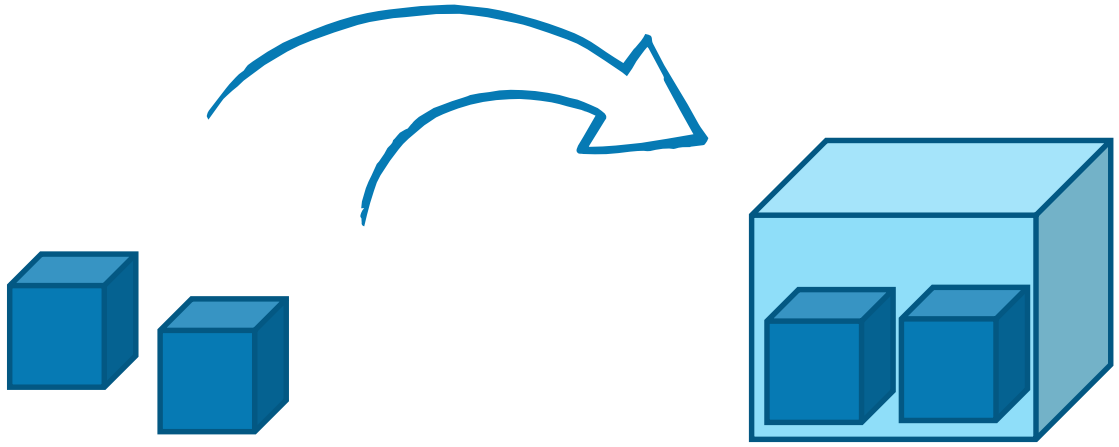
Scale of VNF clouds



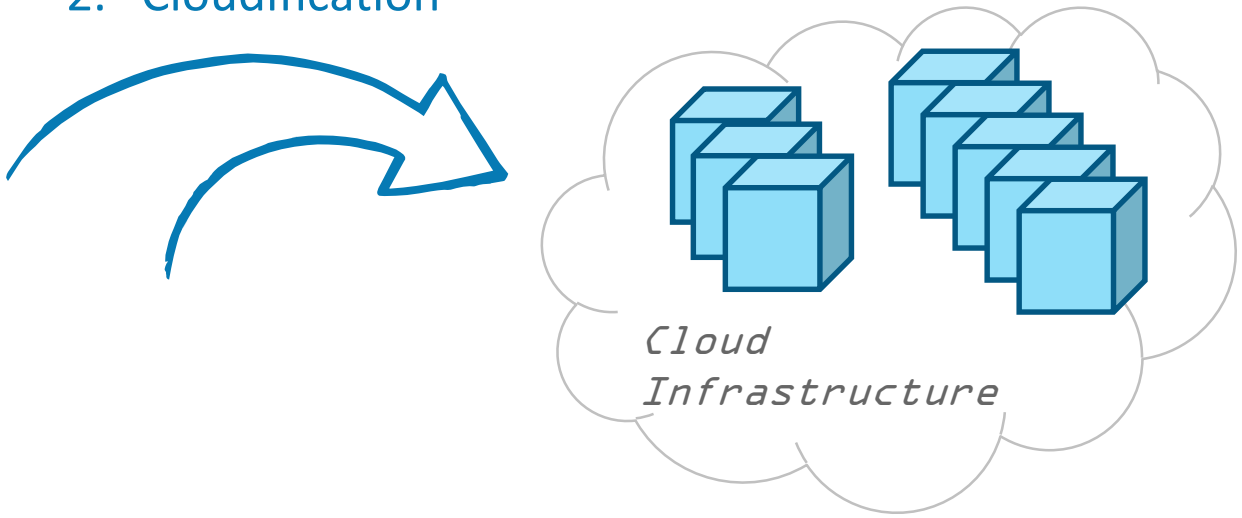
SOME SOLUTIONS

Tools

1. Virtualization



2. “Cloudification”



Containers, DPDK, SR-IOV

VM Replication, state replication

Failure detection

Leader election, consensus

Decoupling state from VNF replicas

Auto-scaling

Auto-repair

Auto-diagnosis

…

Cloud QoS

Automation

Scalable data collection

Scalable orchestration

Optimization

Cloud QoS Project (at AT&T Labs – Research)

Vision

- QoS *plus* all the advantages of a multi-tenant cloud infrastructure

Architecture

- Compute, network, storage (OpenStack++, SDN, Software-Defined Storage)

Techniques

- QoS specification
- Network control (Open vSwitch - rate limiting, packet classification)
- Hypervisor level controls (e.g., cgroups, interrupt steering)
- Admission control
- Optimization
- Holistic placement of application resources in OpenStack

VALET

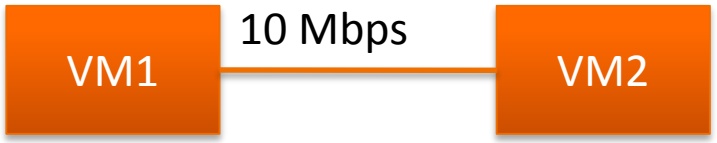
HOLISTIC PLACEMENT AND OPTIMIZATION SERVICE FOR OPENSTACK

Reminder: VNFs have complex requirements

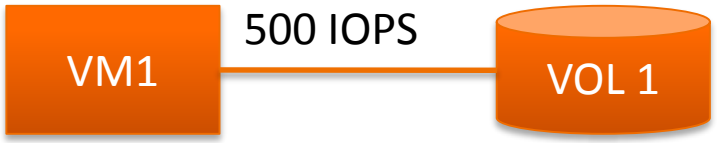
VM1 must have 4 vCPUs, 2 GiB of memory.
Must have assured access to the vCPUs (no overbooking).
Must run on a trusted host. Must support 10K packets-per-second, 1 Mbps
Require SR-IOV or DPDK. …

VOL 1 must be 100 GB, high performance (SSD), 1000 IOPS, 0.999 available, 0.9999 reliable, …

VMs 1, 2, and 3 must be in different racks (or hosts). Or on exclusive hosts (security).

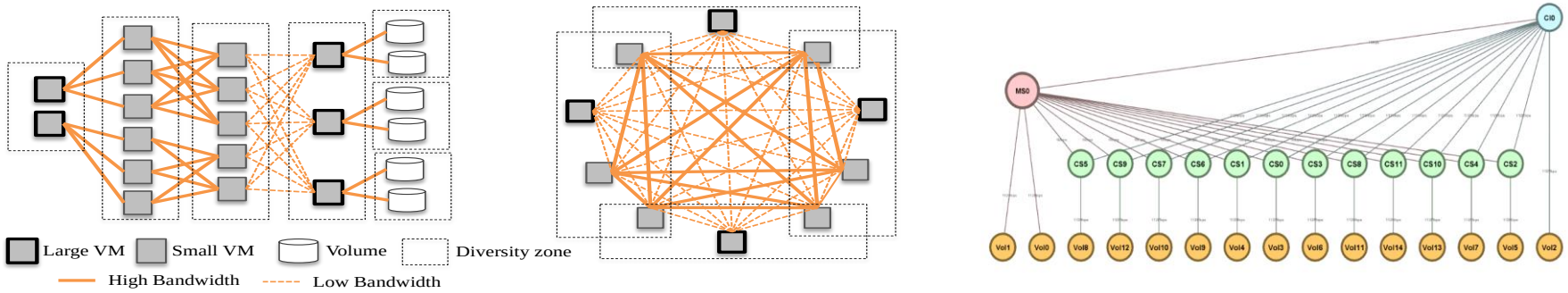


VMs VM1 and VM2 require a guaranteed network link of 10 Mbps.

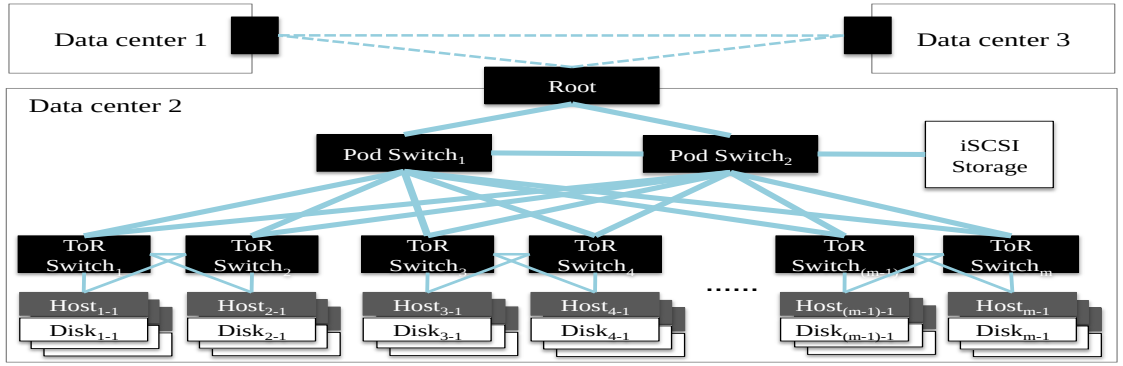


VM1 requires 500 IOPS connection to VOL1.

Valet's Mission



Placement



28

CONTEXT

Valet implemented as an extension to OpenStack.

Start with a historical perspective on OpenStack with focus on resource placement.

First there was Nova

OpenStack's Compute Service – allows the creation and management of virtual machines.

 # nova boot --image *imageID* --flavor *flavorID* --nic net-id=*nicID*

A number of constraints on on which host the VM can be and should be deployed:

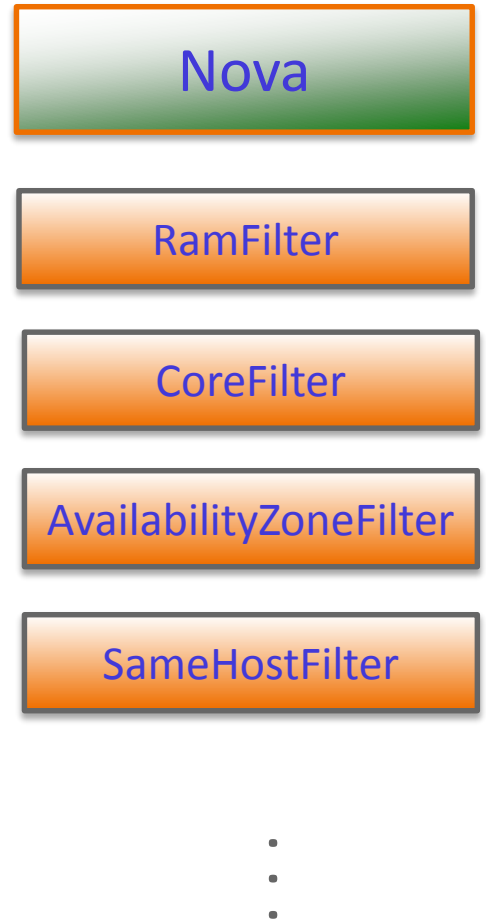
- Must have enough memory, CPU, local disk, ..
- Host must have special capabilities such as SSD, trusted, 1:1 CPU over-subscription ratio,..
- Must be on the same host/different host as some other VM, …
- ….

Nova scheduler filters

Each filter implements one scheduling constraint check.

A host that passes all filters is a candidate for VM placement.

More filters in each OpenStack release (34 in Kilo).

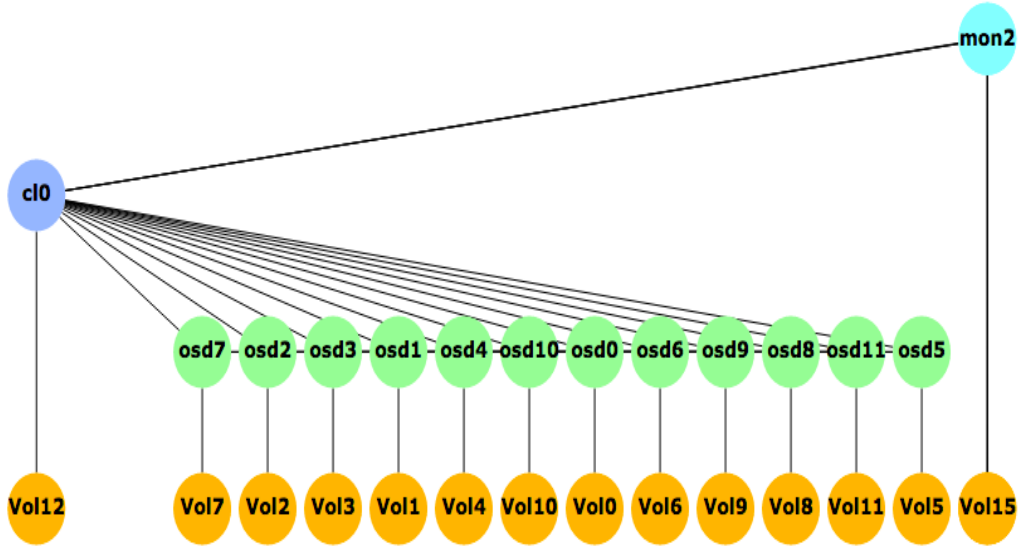


Cloud apps keep on getting more complicated

May consist of 10s of VMs and volumes.

VMs have dependencies, e.g., the software in one VM needs to know the IP address of another VM.

Creating an application one VM at a time becomes a tedious and error prone process.



Then came Heat

OpenStack's Orchestration Service – allows the creation and management of stacks, where each stack is described by a Heat template.

Heat template: specifies the cloud resources (VMs, volumes, networks, ports, etc), software configuration, even autoscaling of the application.

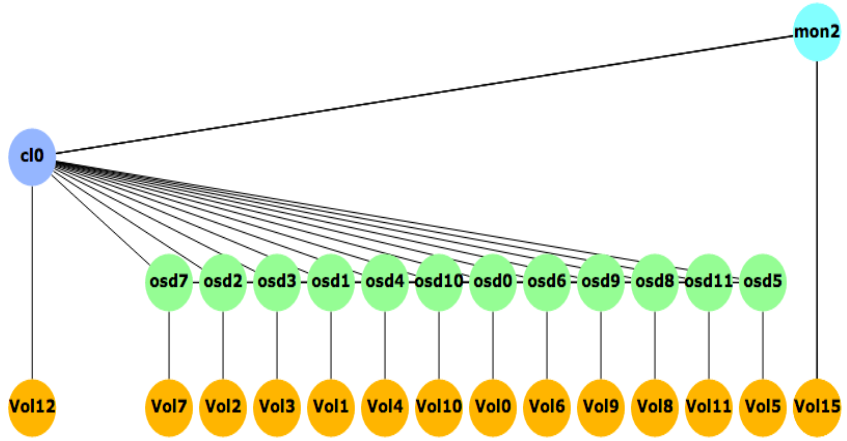
#heat stack-create my_first_stack –template_file=my_template.yaml

A complete application can be deployed, configured, and started with one command!

Cloud applications have requirements

Each VM/volume has its own requirements but the application often has requirements across a set of VMs/volumes.

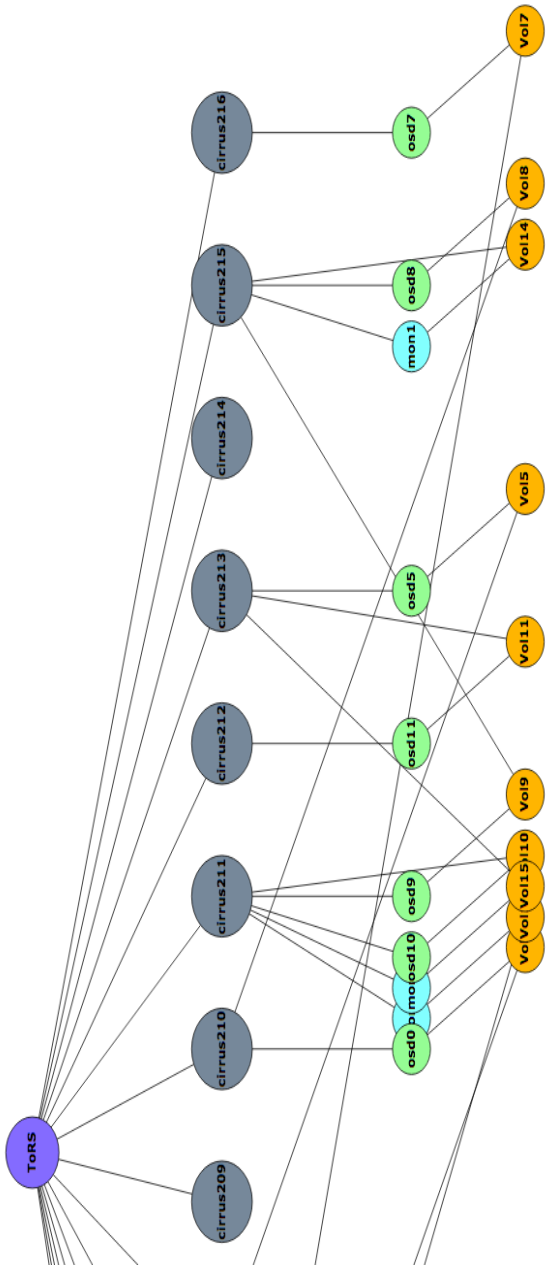
- A group of VMs replicating a service should be placed on different racks.
- Two VMs should be placed so that a 1 Gbps bandwidth can be guaranteed between them.
- VM and volume should be placed so that 1000 IOPS can be guaranteed.
- …



We have a problem ….

Nova and Cinder schedule each VM/volume independently without considering how they relate to other VMs/volumes in the same application.

- Makes it hard for applications to express and satisfy their constraints.
- Satisfying application constraints may require support from the cloud Ops (e.g., create a host aggregate, availability zone).
- Expensive, error prone, brittle, and leads into resource fragmentation.
- Poor application performance, inefficient use of cloud resources.



SOLUTION: Valet Placement Service

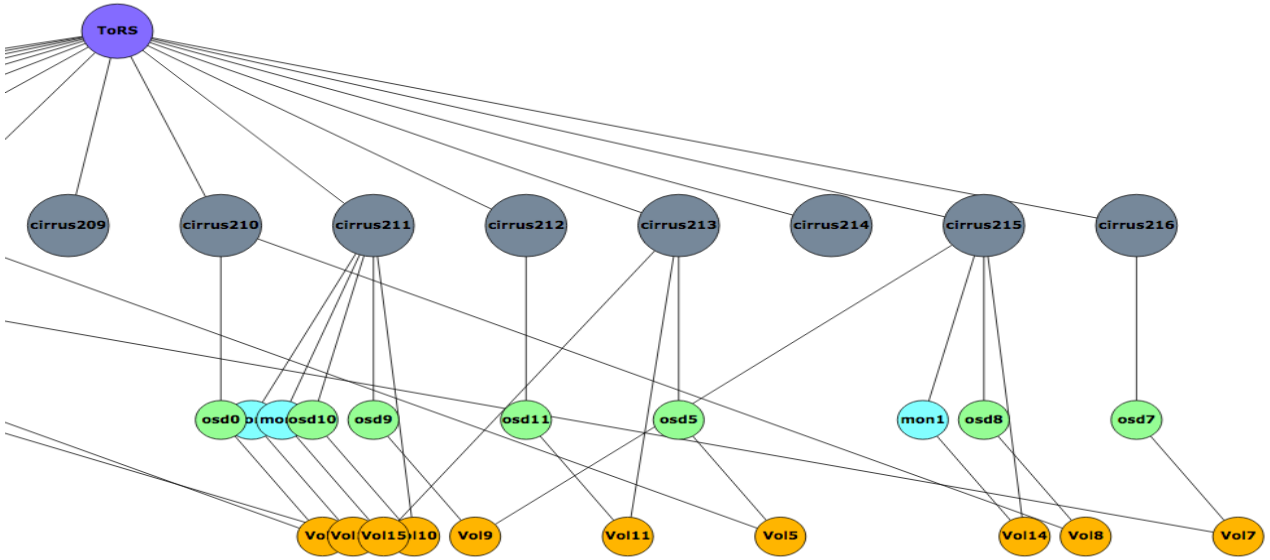
Valet is a holistic placement service for cloud resources within a data center.

It hides the underlying details of the hardware layer, and allows the application/VNF/service to express placement requirements at an abstract level.

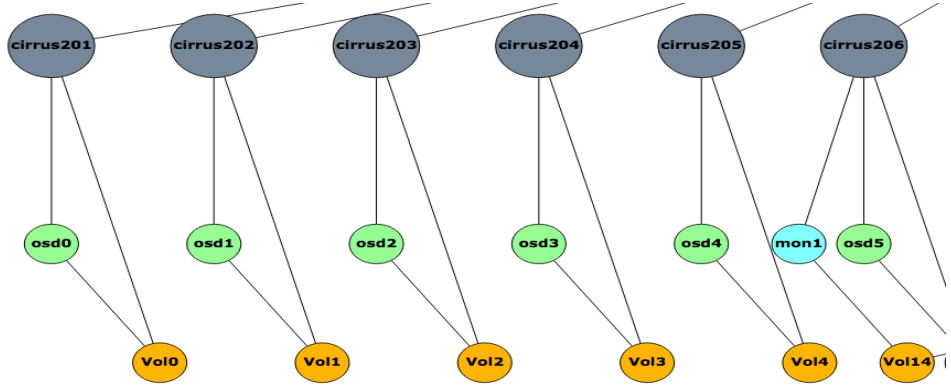
Valet maintains a detailed, up to date view of the data center's resources, and uses powerful constraint solving and optimization methods to satisfy all application requirements while attaining the best use of capital.

Impact of Valet

Without Valet



With Valet



SPECIFICATION OF APPLICATION PROPERTIES

Let’s extend Heat …

Application specification

Standard OpenStack: Specify VM flavor, volume size, possibly volume QoS, limited affinity/anti-affinity, availability zones.

Valet provides new Heat resource types. For example:

- ATT::Valet::Pipe (VM to VM bandwidth reservation)
- ATT::Valet::GroupAssignment (exclusivity, diversity and affinity groups)

Specifying bandwidth requirements

Application may require an end-to-end bandwidth guarantee between VMs.

Valet pick's hosts with enough bandwidth between them.

```
MS_CS1_Bandwidth_Reservation:
    type: ATT::Valet::Pipe
    properties:
            bandwidth: 5.0
            resources:
            - {get_resource: MetaServer}
            - {get_resource: ChunkServer}
```

Specifying affinity

Application may want VMs/volumes to be close to one another.

```
mon1-affinity:
  type: ATT::Valet::GroupAssignment
  properties:
    group_type: affinity
    level: host
    resources:
    - { get_resource: mon1-ceph }
    - { get_resource: vol-mon1-ceph }
```

Specifying diversity

Application may want VMs/volumes to be spread across multiple hosts, racks, (or even sites).

```
monitor-diversity:
  type: ATT::Valet::GroupAssignment
  properties:
    group_type: diversity
    level: rack
    resources:
    - { get_resource: mon1-affinity }
    - { get_resource: mon2-affinity }
    - { get_resource: mon3-affinity }
```

Specifying exclusivity

Application, or group of applications, may require exclusive placement for security reasons.

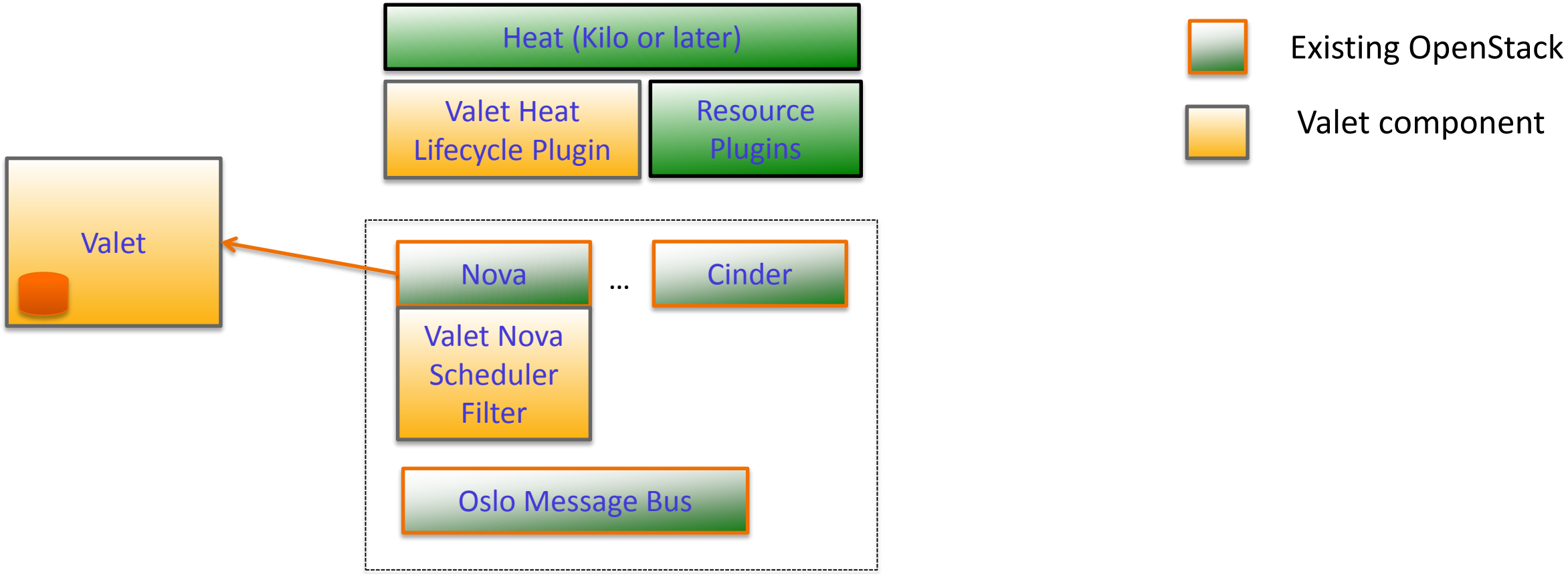
```
my_exclusive_group:
    type: ATT::Valet::GroupAssignment
    properties:
      group_name: Core_Infrastructure_VNFs
      group_type: exclusivity
      level: host
      resources:
      - { get_resource: <first_VM> }
      - { get_resource: <second_VM> }
```

How Valet operates in three simple steps

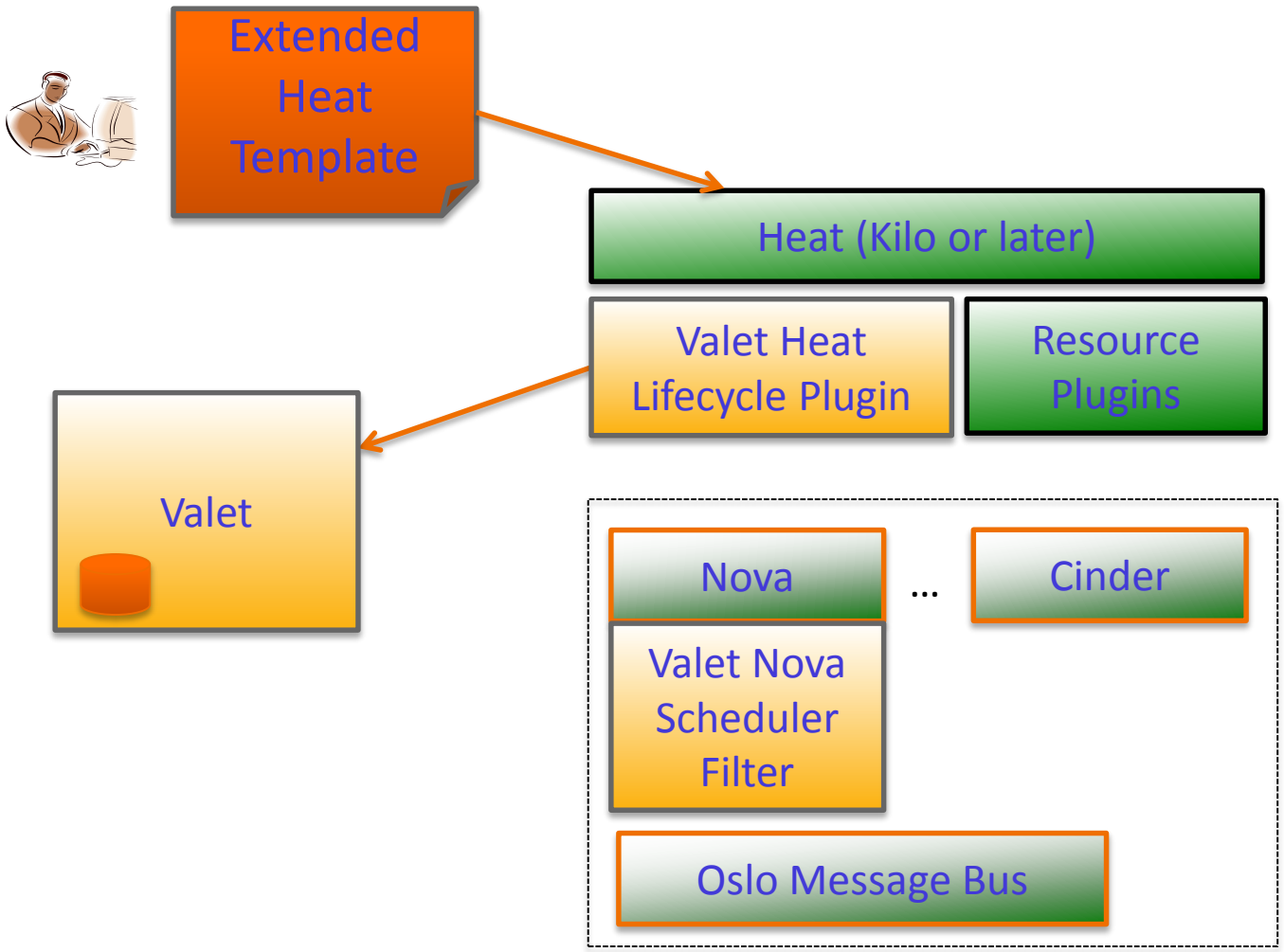
Well, one of them is pretty hard … actually NP-Hard.

Valet bootstrap

When Valet starts up it gets information from Nova (and other OpenStack services)



Valet placement planning



Tenant invokes Heat with template. Lifecycle plugin passes info to Valet.

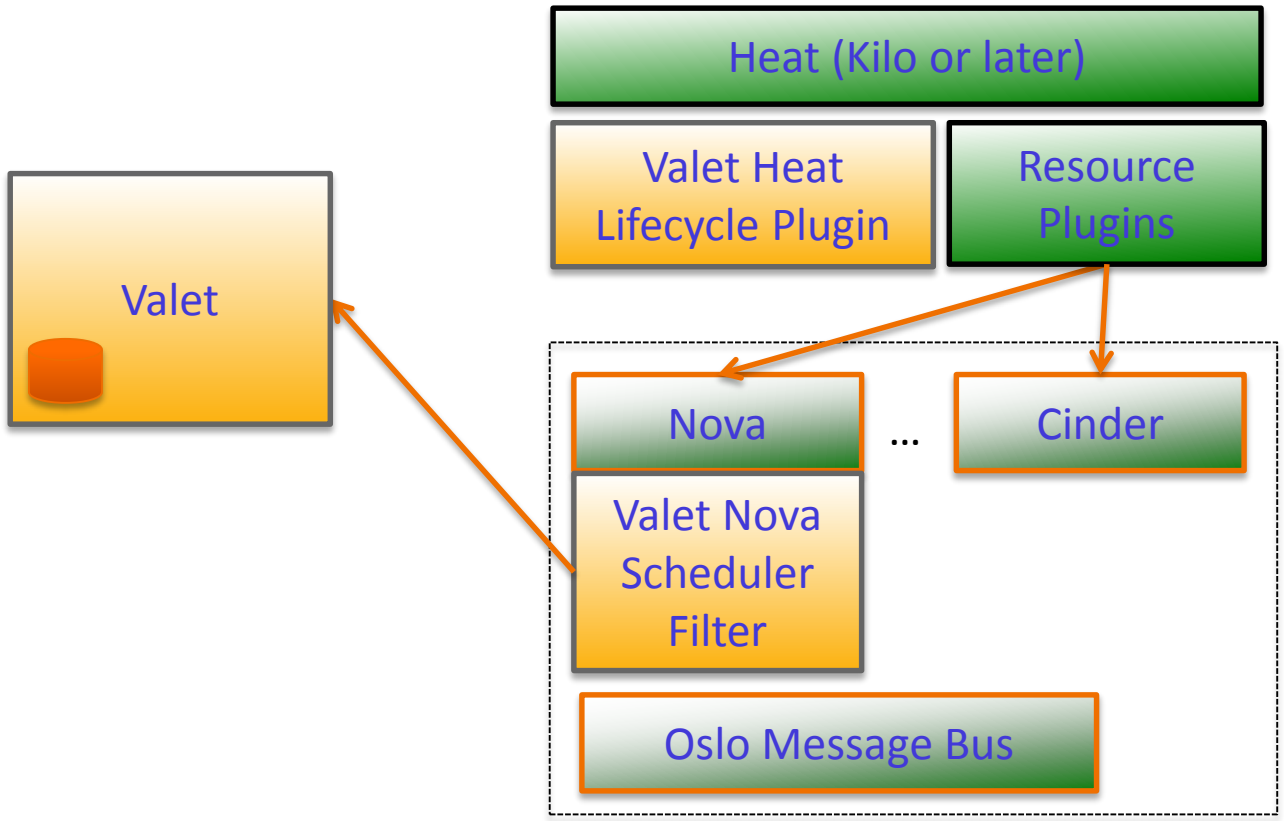
Valet's constraint-solving optimization algorithm finds a placement that satisfies all requirements while optimizing the resource usage of the OpenStack site.

Valet remembers the decision locally.

Existing OpenStack

Valet component

VM placement and creation



Heat calls Nova to create a VM. Valet Nova scheduler filter queries Valet for the stored decision; resources are placed according to the decision.

Existing OpenStack

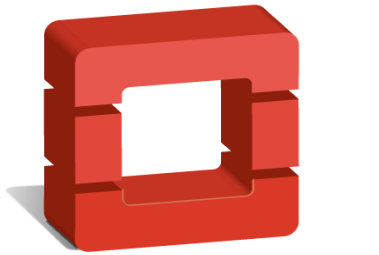
Valet component

Valet and OpenStack

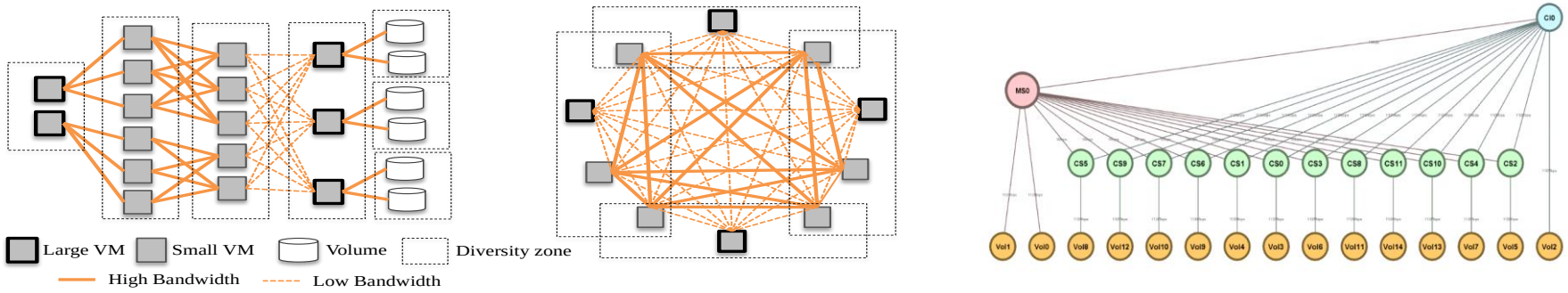
Valet is compatible with OpenStack Kilo. It does not require any changes in the code of existing OpenStack services.

Valet is in the process of being released as AT&T Open Source projects.

Valet was presented to the OpenStack community for the first time at the OpenStack Summit in April 2016.

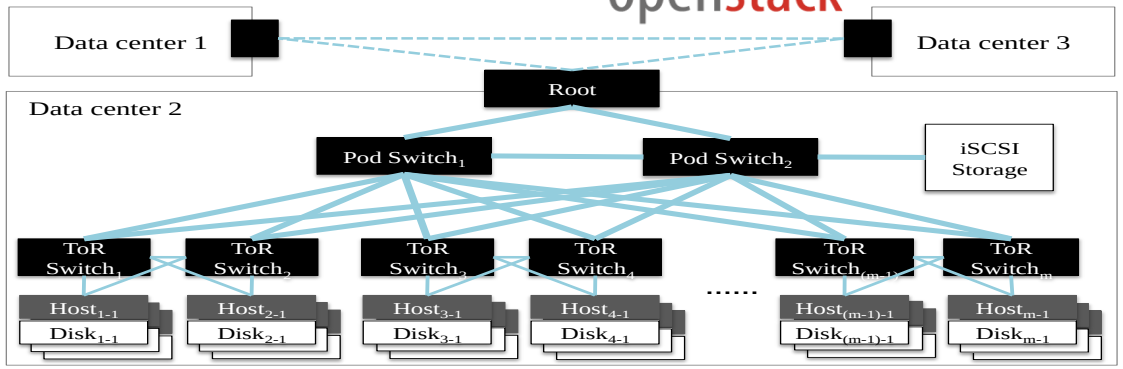


Valet Summary



Placement

openstack™



49

LOOKING FORWARD

VNFs in massively distributed clouds

Service owner

Cloud provider

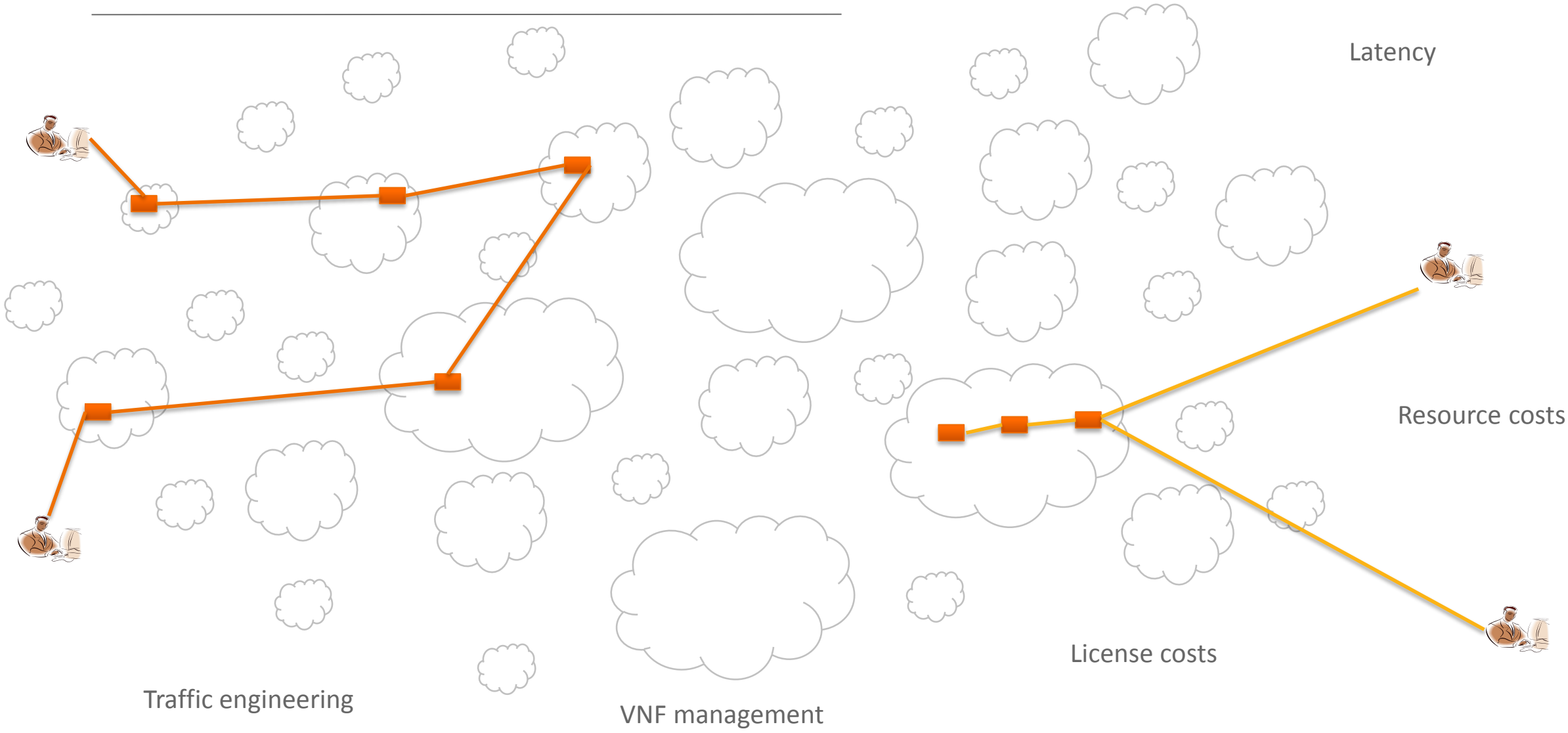
Optimization of resource usage.

Specification of service requirements

(proximity, latency, affinity, diversity, ..)

Scalable data collection and optimization.

Service Chaining in massively distributed clouds



IoT and cloud computing

CONCLUSIONS

Conclusions

VNFs, SDN, and cloud computing are key enabling technologies for telco services TODAY.

Many business opportunities and benefits, but also numerous challenges for the research community.

QoS in a multi-tenant cloud.

Scale of the massively distributed NFV cloud.

Cloud-aware VNFs.

AT&T Labs – Research is addressing many of the challenges.

Emphasis on using/extending open source software (OpenStack, Zookeeper, Cassandra, Ceph, etc) and contributing back to the open source community.

Questions?

Thank You