Software Defined Networks

Virtualization and Telecom Clouds

Dr. Walter Häffner - 8. KuVS Fachgespräch NGSDP, Hirschburg, 17 April 2013

Software Defined Networks
Virtualization and Telco Clouds – Topics

Software Defined Networks – SDN

- idea is not new but by now it hypes
- all major network suppliers jump in
- Vodafone just began some research
- What is the basic idea behind SDN?

A Telco Cloud for Internal Services

- Virtualization in Computing not new
- Promises simplification and savings
- Vodafone started Telco Cloud trials
- VF-D going to be first with NGN/IMS

Software Defined Networks
Mastering Complexity by Brute Force

No-one can really master the complexity of our IP networks anymore and it **costs us more than a fortune to keep our telco business alive**

Well, that‘s the price you‘re gonna pay for Internet, IPTV, VoIP and that App stuff. Fact is, we need **more powerful boxes and more headcount**

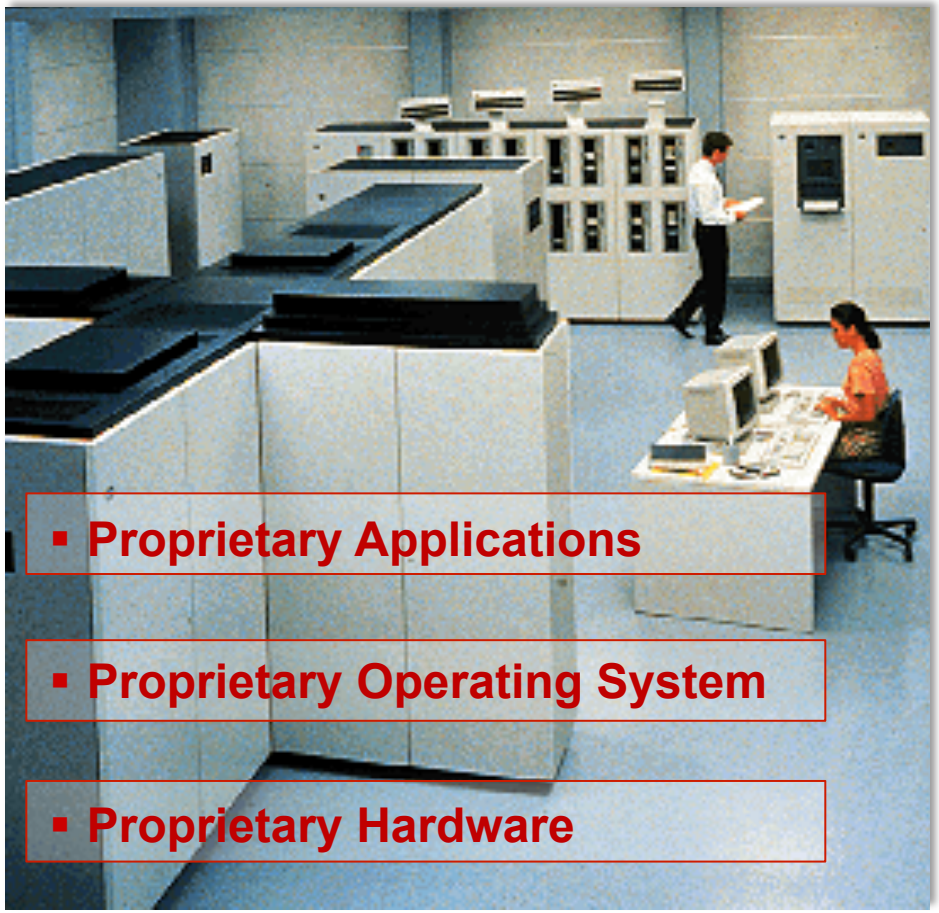
Software Defined Networks
Mastering Complexity: In Search of Creative Answers

I hear this year by year.
We have to stop this way of working.
Any innovative ideas?
Other diciplines do better.

Well, indeed our IT colleagues have some advantages. **IT ystems are much more modular** and in **IT innovation is faster**.

Software Defined Networks
Is there any Magic behind IT Innovation Speed?

The Golden 60th: IBM System 360



In the 70th IT became modular

ORACLE SAP Wolfram Mathematica 8 OpenOffice.org

Open Interfaces

Windows 7 Linux Mac OS

Open Interfaces

Software Defined Networks

In 2013 Telco Boxes are still Monolithic Proprietary Solutions

Situation 2013

A Dream Scenario beyond 2013

Policy Compiler

VPN Creator

Other 3rd Party Apps

Open Interfaces

Control Plane

Control Plane

vmware

nicira

Still academic and industrial research

Open Interfaces

Standardized Switch/Router Module

Software Defined Networks

People, Organizations, Business behind this Approach

Martin Casado

Nick McKeown

Scott Shenker

founded in 2007

nicira networks

in 2012

vmware

founded 1998

USD 1.26 Billion

Most known buzzword in this context: OpenFlow

Software Defined Networks
Basic Idea: Separating Control and Forwarding Planes

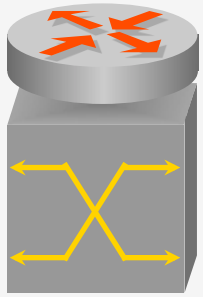
Traditional Router

OSPF Distributed System

IS-IS Distributed System

Operating System

Custom Hardware



CPU: intelligent and slow

ASIC: dumb and fast

<u>Routing is distributed computing:</u>

RFC 2328 OSPF is 244 pages,
Thereof ~100 pages on consistency

Dijkstra Algorithm for shortest path can be described on 4 pages

Software Defined Network

Centralized OS somewhere in the Telco Cloud

OSPF Centralized System

IS-IS Centralized System

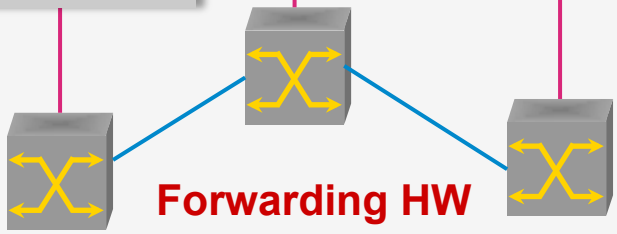
Global Network View

Network Operating System

OpenFlow

OpenFlow

OpenFlow



Forwarding HW

Software Defined Networks
It's not (only) Separation of Layers, it's about Abstraction

Information Technology Example: Transitions in Programming

- **Assembler: no abstractions**

  Mastering complexity was crucial

- **Higher level languages: OS and other abstractions**

  File systems, virtual memory, abstract data types, …

- **Modern languages: even more abstractions**

  Object orientation, garbage collection, …

- **Lessons learned: Abstractions are key to extracting simplicity**
- **Design of large systems based on abstractions as foundation of proper subsystem decompositions**

Software Defined Networks
Abstraction: Inheritance in C++ (Modern Language?)

```
class Polygon
{
   protected int width, height;
   public void setParameters (int w,int h)
   {
      width = w;
      height = h;
   }
};

class Rectangle : public Polygon
{
   public int calculateArea ()
   {
      return (width*height);
   }
};

class Triangle : public Polygon
{
   public int calculateArea ()
   {
      return( (width*height)/2);
   }
};
```

rectangle inherits width and height from polygon

triangle inherits width and height from polygon

just for completeness: the area calculation

```
int main ()
{
   Rectangle rect;
   Triangle trgl;
   rect. setParameters (4,5);
   trgl. setParameters (4,5);
   cout << rect. calculateArea () << endl;
   cout << trgl. calculateArea () << endl;
   return 0;
 }
……..
……..
```


Software Defined Networks
It's not (only) Separation of Layers, it's about Abstraction

Network Example: There's Light and Darkness

- **OSI (or other) layer models are great abstractions**

  But this deals only with the data plane

- **Currently no powerful control plane abstractions in place**
  - Generate configuration for each physical device,

    e.g. ACLs, forwarding tables, policies, …
  - Operate without communication guarantees

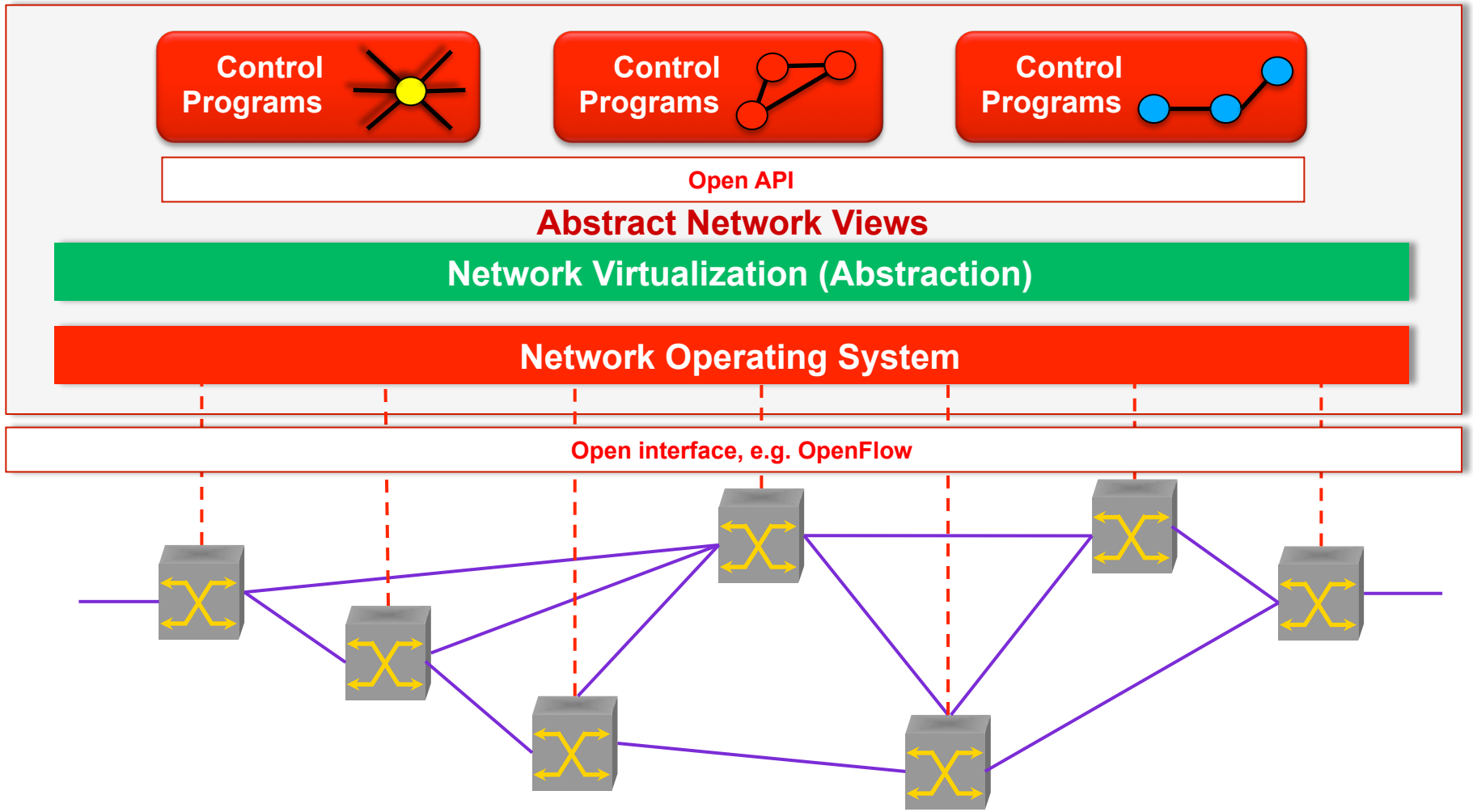
    No automatic consistency check, prone for men made errors

    From time to time there are misconfigured boxes out there
  - Only people who love with complexity find this situation reasonable
  - Software engineers would immediately start to abstract

Software Defined Networks

Essential Objective is an Abstraction Layer above Network OS

Centralized in a Telco Cloud

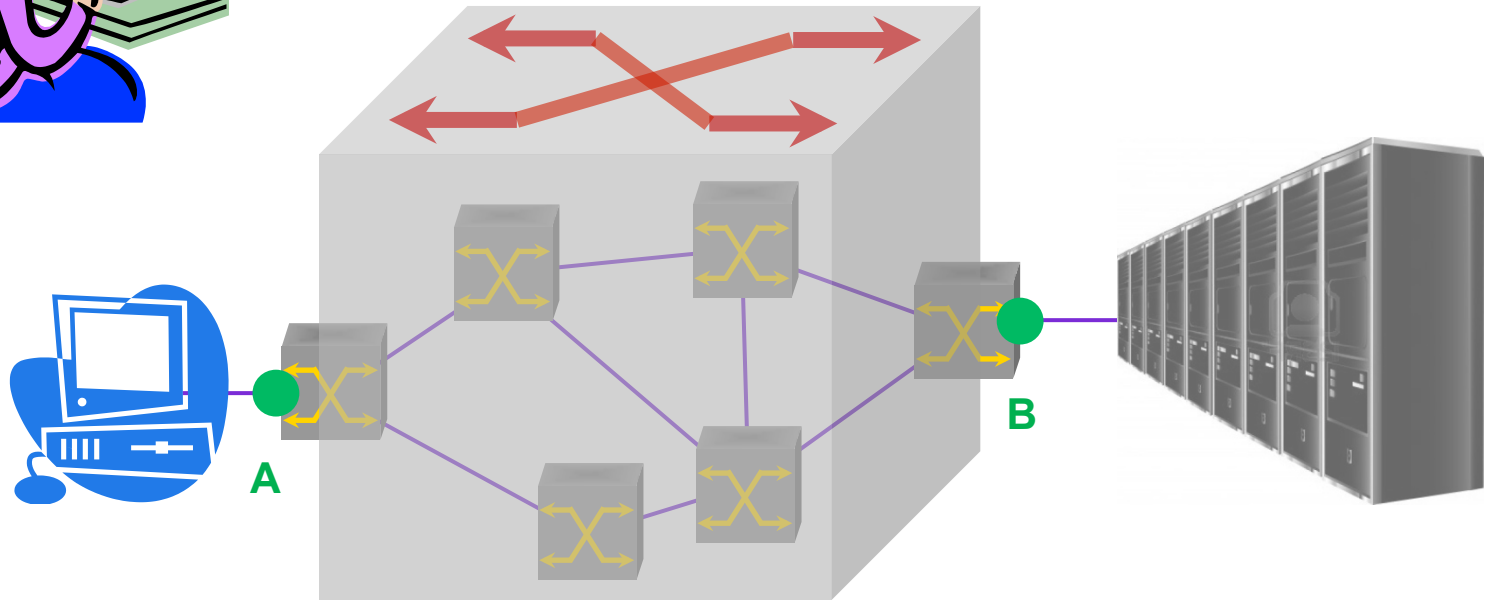


Software Defined Networks

The Network looks like one big Switching Fabric

From A to B I like to have

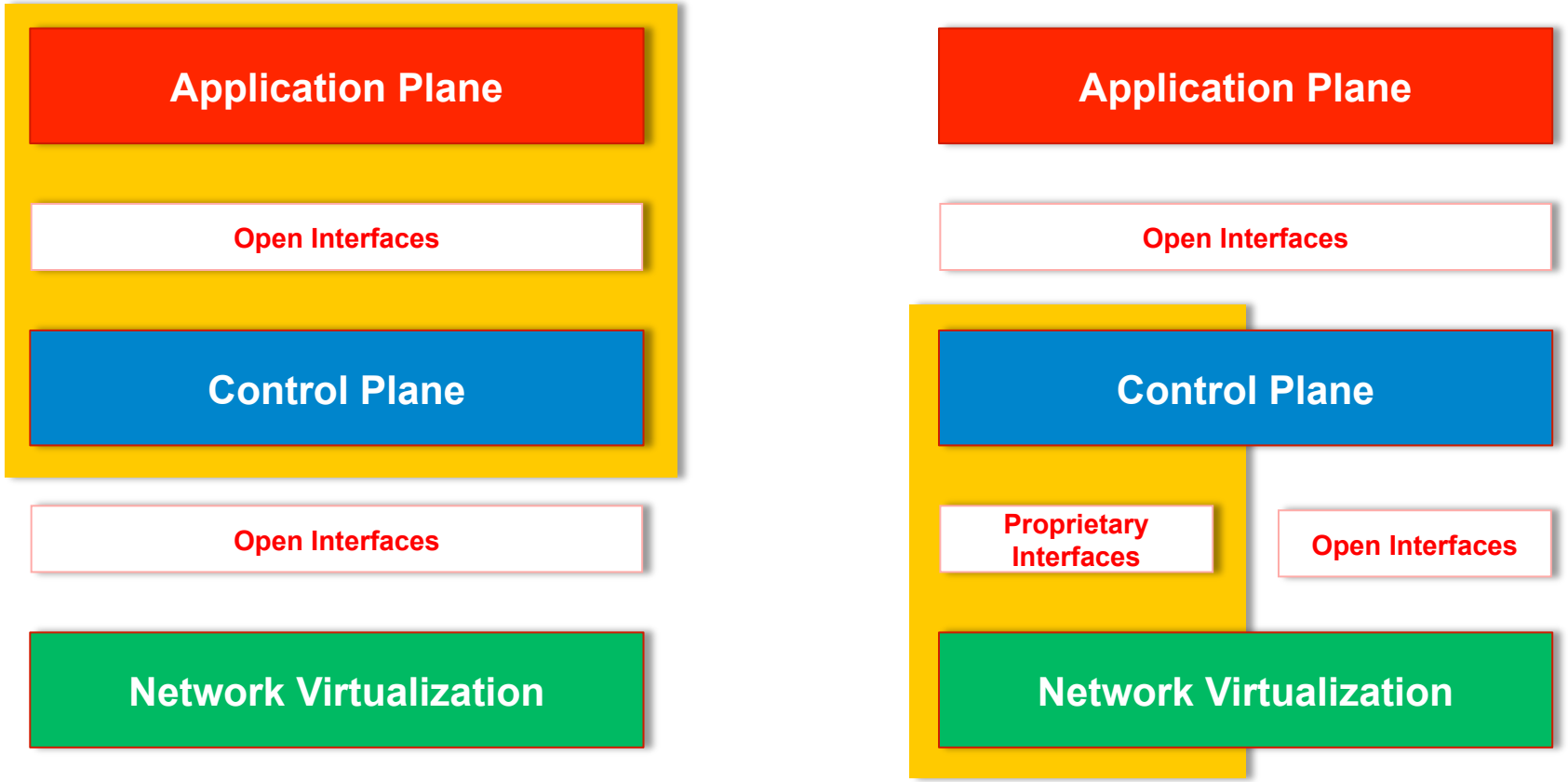
- **just one best effort connection <OR>**
- **a single connection for VoIP traffic class <OR>**
- **a connection for video with lowest jitter <OR>**
- **a SAP connection with a back-up path**



Software Defined Networks

Some vendors currently try to keep decentralized Control Plane

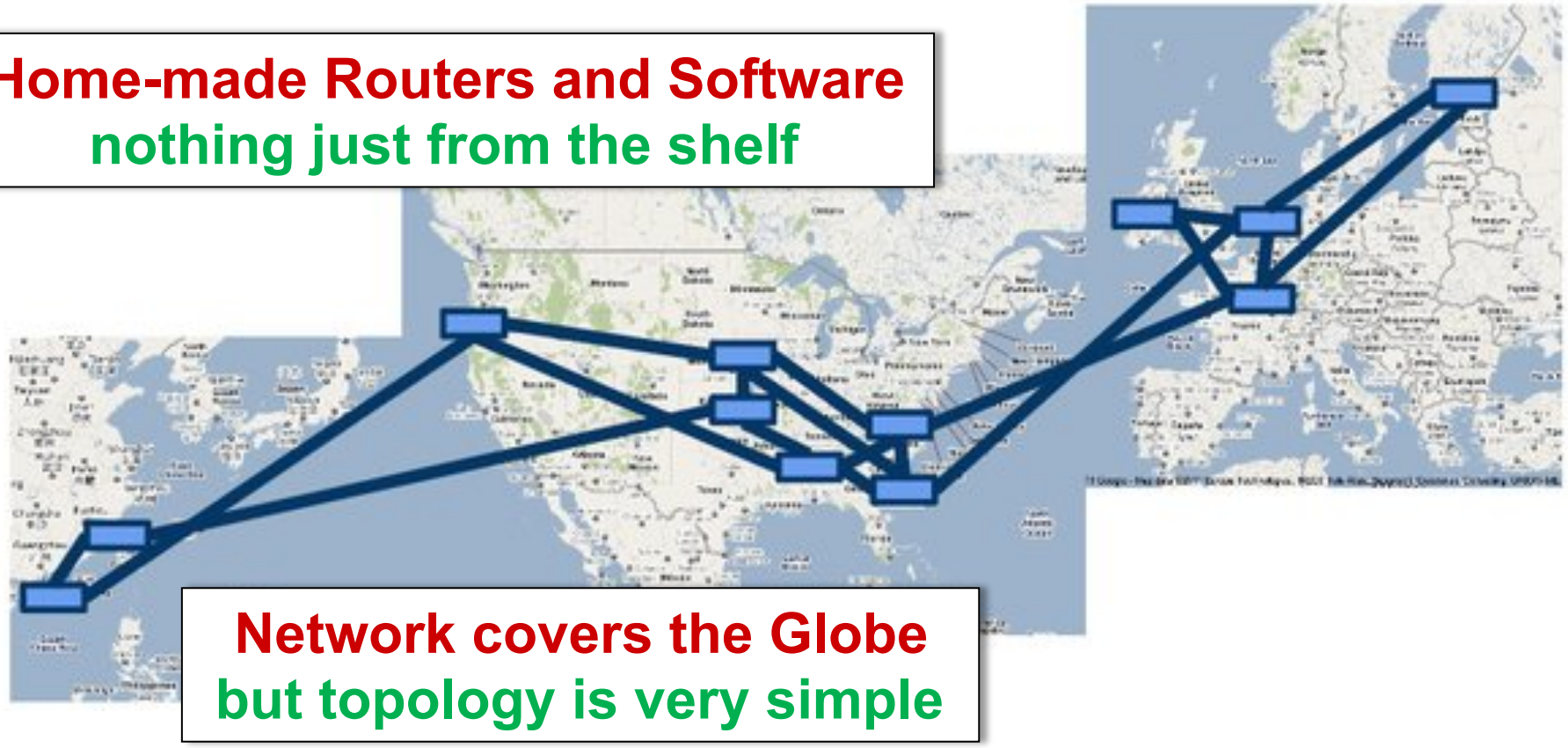
Spectrum: From fully centralized Control Planes to more traditional Systems with open APIs



Software Defined Networks

Google is using SDN Technology for their Data Distribution Network

Home-made Routers and Software
nothing just from the shelf



Network covers the Globe
but topology is very simple

OpenFlow applicable
not that much features required

Software Defined Networks

Down to Earth – SDN is some Years in the Future

.. Sounds great – but frankly, it‘s science fiction.

SDN seems to be 5 to 10 years away. Forget it!

- Well, to some extend we can go this way
- In NGN a **SIP server is a control plane** and the **IP network is the data plane**. It‘s **just another level of abstraction**
- And we **save you a lot of money** when we throw control functions whenever possible onto **cheap standardized servers**
- And it‘s **even more economical** when we put everything into **Virtual Machines**.
- Virtual machines are software and therefore easy to move

Software Defined Networks
Virtualization - Your Personal Use Case

You are proud of your MacBook and like your Mountain Lion

But sometimes you wish you had your good old bloody Windows

- Buy a piece of emulation software – friends recommend VMware or Parallels
- Install it like any other program
- Follow the instructions to install Win7 or 8

- If done properly you can run a Windows PC including disk space like any other program on your MacBook with the Mountain Lion OS
- Your new PC is a Virtual Maschine running on top of VMwares Virtualization Layer software
- You even could install more than one WinXP
- You have more Laptops for the price of one

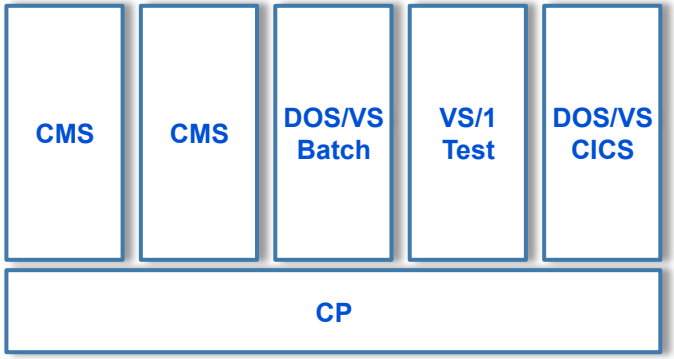
Software Defined Networks
In the 70th: IBM VM/370 already supported Virtual Machines

VM/370 - a study of multiplicity and usefulness
by L. H. Seawright and R. A. MacKinnon
IBM SYST J VOL 18 NO 1 1979

"The **productivity** of data processing professionals and other professionals **can be enhanced** through the use of **interactive and time-sharing systems**. Similarly, system programmers can benefit from the use of system testing tools. A systems solution to both areas can be the **virtual machine concept**, which provides multiple software replicas of real computing systems on one real processor. Each **virtual machine** has a full complement of input/ output devices and provides functions **similar to those of a real machine**. One system that implements virtual machines is **IBM'S Virtual Machine Facility1370 (VM1370)."**

VMMs, Virtual Machine Monitors, also known as **hypervisors**, first reached prominence in the early 1970s and achieved commercial success with the **IBM 370** mainframe series. Virtualization allowed mainframes to run **multiple operating systems simultaneously**, thus making it possible **to time-share expensive hardware** without requiring software modifications to legacy systems, including single-user operating systems.

CP/CMS: Control Program/Cambridge Monitor System
CICS: Customer Information Control System
DOS/VS: Disc Operation System/Virtual Storage



Software Defined Networks

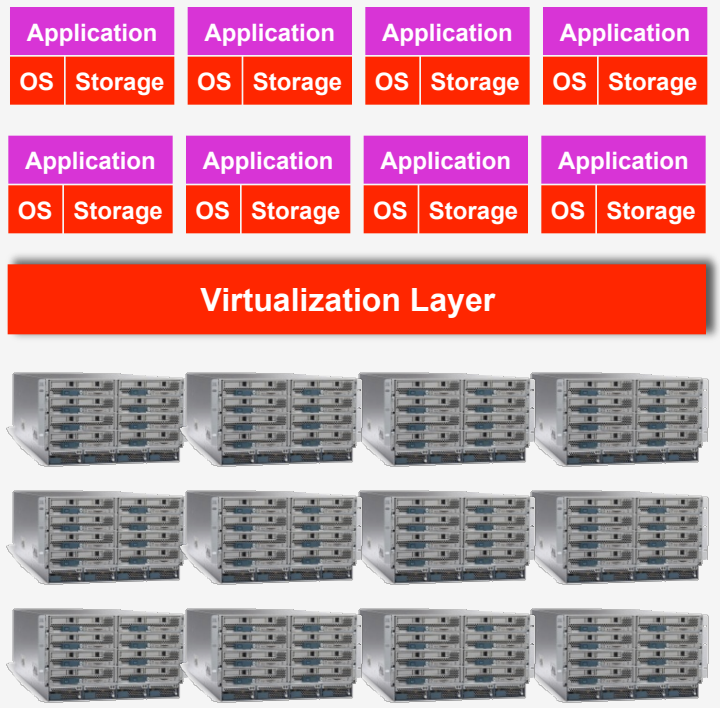
Network Infrastructures became inhomogeneous over the Decades

- **Carrier Network typically is multi-vendor**
  - Mix of modern and legacy equipment
  - Ongoing harmonization towards All-IP
  - Phase-out SDH and ATM networks
  - Phase-out legacy services like ISDN
- **IP brought many server-based functions**
  - Often applications like AAA, DNS, IMS, NGN, Web, typically reside on supplier preferred specific server platforms
  - Results finally in a zoo of hardware
  - Some servers close to idle
  - Some servers close to overload
  - Often capacity provisioning takes too long
  - Infrastructure costs going to explode

Software Defined Networks

Benefits of Virtualization: Economics of Scale, Simplification, Speed

Much more Virtual Machines than Physical Servers

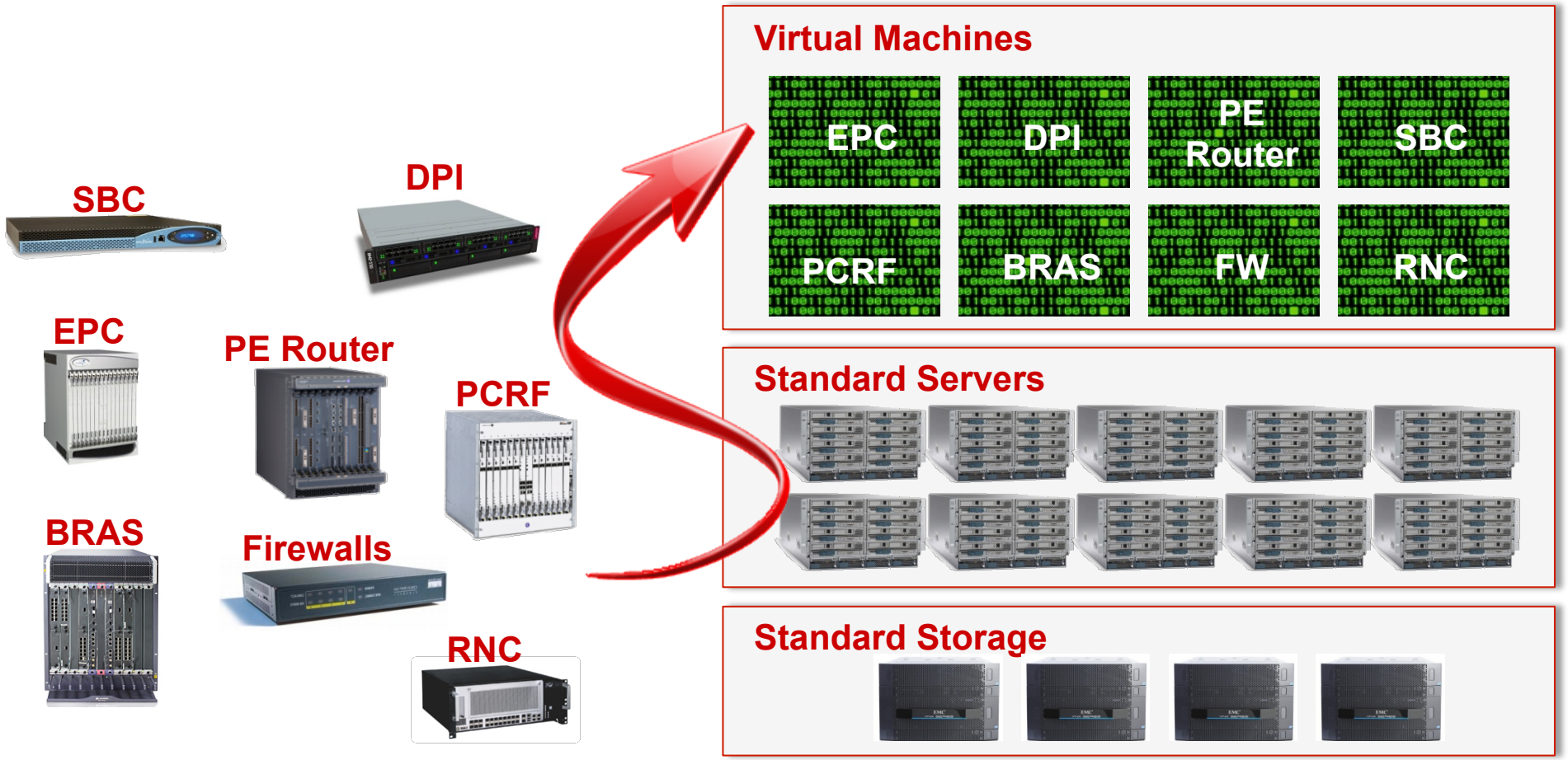


Larger Computer Center based on Standard Blade Servers

- **Virtualization of (legacy) servers**
  - Reduced hardware support costs
  - Reduced range of platforms supported
  - Reduced power and space requirements
  - Standardized configurations
- **Abstract application from hardware**
  - Changes to configuration will be less intrusive and time consuming
  - Dynamically change configurations according actual meet business needs
- **Improve delivery through faster build**
- **Snapshot and roll back opportunities**

Software Defined Networks

Benefits of Virtualization: Economics of Scale, Simplification, Speed



Software Defined Networks

Economical Benefits of Virtualization (Source VMware)

- Reduce the varieties of server models and configurations
- Reduce the range of spare parts to be kept on site in case of failure
- Reduce the complexity of the computing center
- Improve the performance per watt ratio of the computing center
- Reduce hardware and operating costs by as much as 50% and energy costs by 80%, saving more than $3,000 per year for every server workload virtualized
- Reduce the time it takes to provision new servers by up to 70%
- Decrease downtime and improve reliability with business continuity and built-in data disaster recovery
- Deliver IT services on-demand now and in the future, independent of hardware, OS, application or infrastructure providers

Software Defined Networks

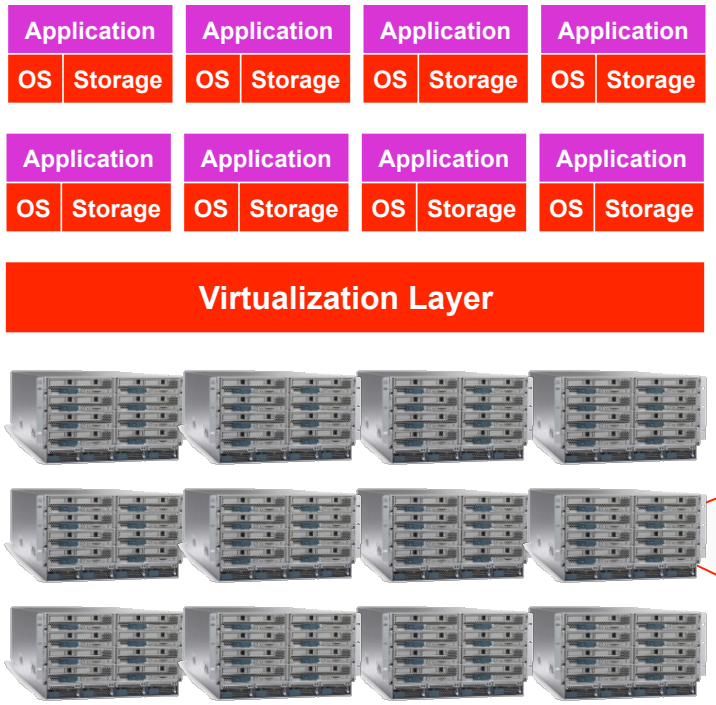
First Steps: Migrate VoIP Services into a Telco Cloud

- **Over the last 20 years a variety of voice service platforms appeared**
  TDM (GSM, UMTS, ISDN), IP Packet (fixed NGN, mobile IMS)
- **Life cycle costs and operational complexity of voice increased rapidly**
- **Older service platforms like ISDN switches are end of life until 2020**
- **Target is one VoIP platform in place for fixed and mobile services**
- **Start migration into internal Telco Cloud with server-based platforms**
  SIP proxies, DNS, AAA, media gateway control functions
- **Drive development to enable migration of systems with proprietary designs**
  SBC, PCRF, BRAS/BNG, …. Often complete new vendor re-designs required

Standard Blade Server Hardware as Platform for Virtual Machines

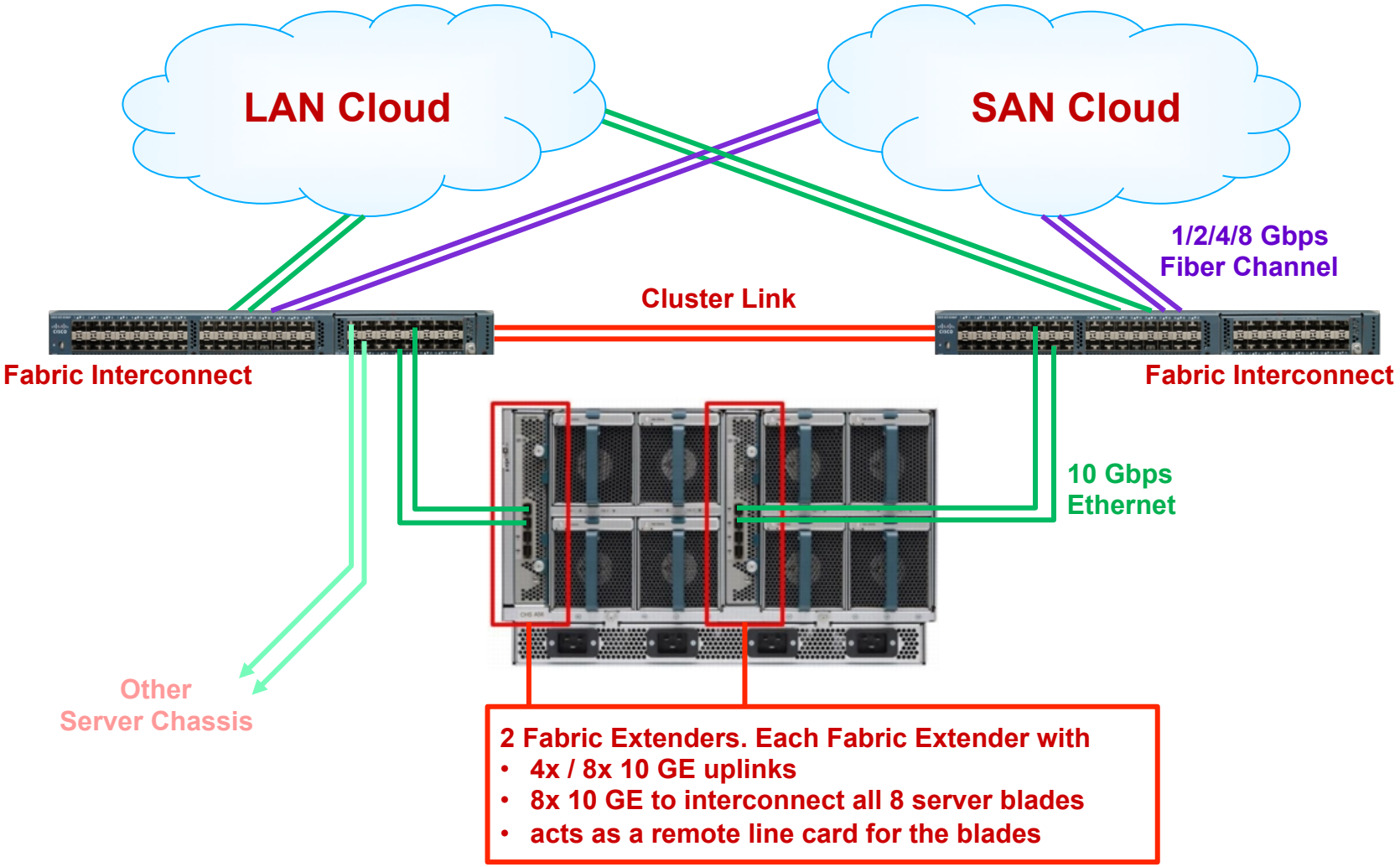
Software Defined Networks
Virtualization – On a larger Scale

Much more Virtual Machines than Physical Servers



Larger Computer Center based on Standard Blade Servers

Software Defined Networks
Redundant Blade Server Integration



Software Defined Networks

Data Center Building Blocks – Example: Cisco Computing Pot UCS-C1

UCS-C1

- Max. 48 x B230 M1 Blade Server @ 128GB RAM
- Dual Xeon X6550 130W processors/blade
- 2 x Cisco virtual interface card/blade
- 6 x 5108 blade enclosures
- 6140 fabric interconnect
- 60+ VM/server possible

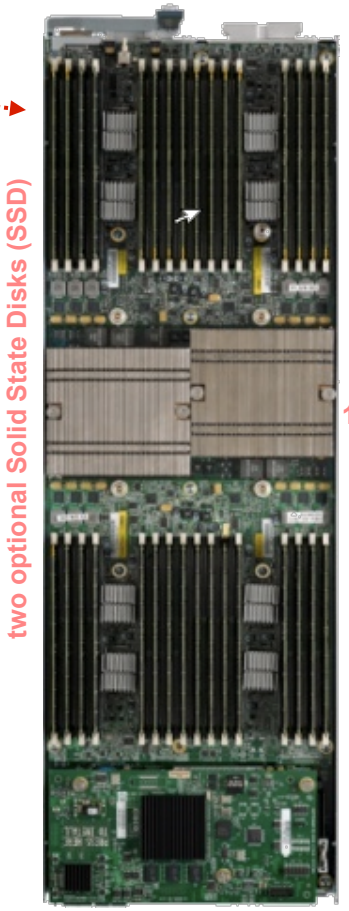
UCS-C1 POD properties

- 12 x 10GE Interfaces to aggregation layer
- 12 x 8G Fiber Channel Interfaces to SAN
- 1 rack space used
- 12.500 W (typical)

UCS Capacity (this example)

- 768 physical CPU cores. physical memory 6.144 TB
- 1.536 Advanced VMs @4GB/VM av. (4 vCore/VM) or
- 3.072 Standard VMs @2GB/VM av. (2 vCore/VM) or
- 6.144 Basic VMs @1GB/VM av. (1 vCore/VM)
- Calculation @ 8:1 vCore/pCore

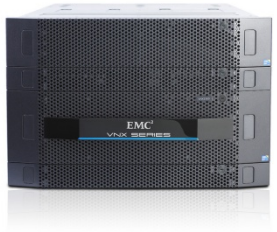
B230 M1 Blade



Software Defined Networks

Data Center Building Blocks – Example: EMC2 VNX Storage Family

VNX5100
max. 75 Drives



VNX5300
max. 125 Drives

VNX5500
max. 250 Drives

VNX5700
max. 500 Drives

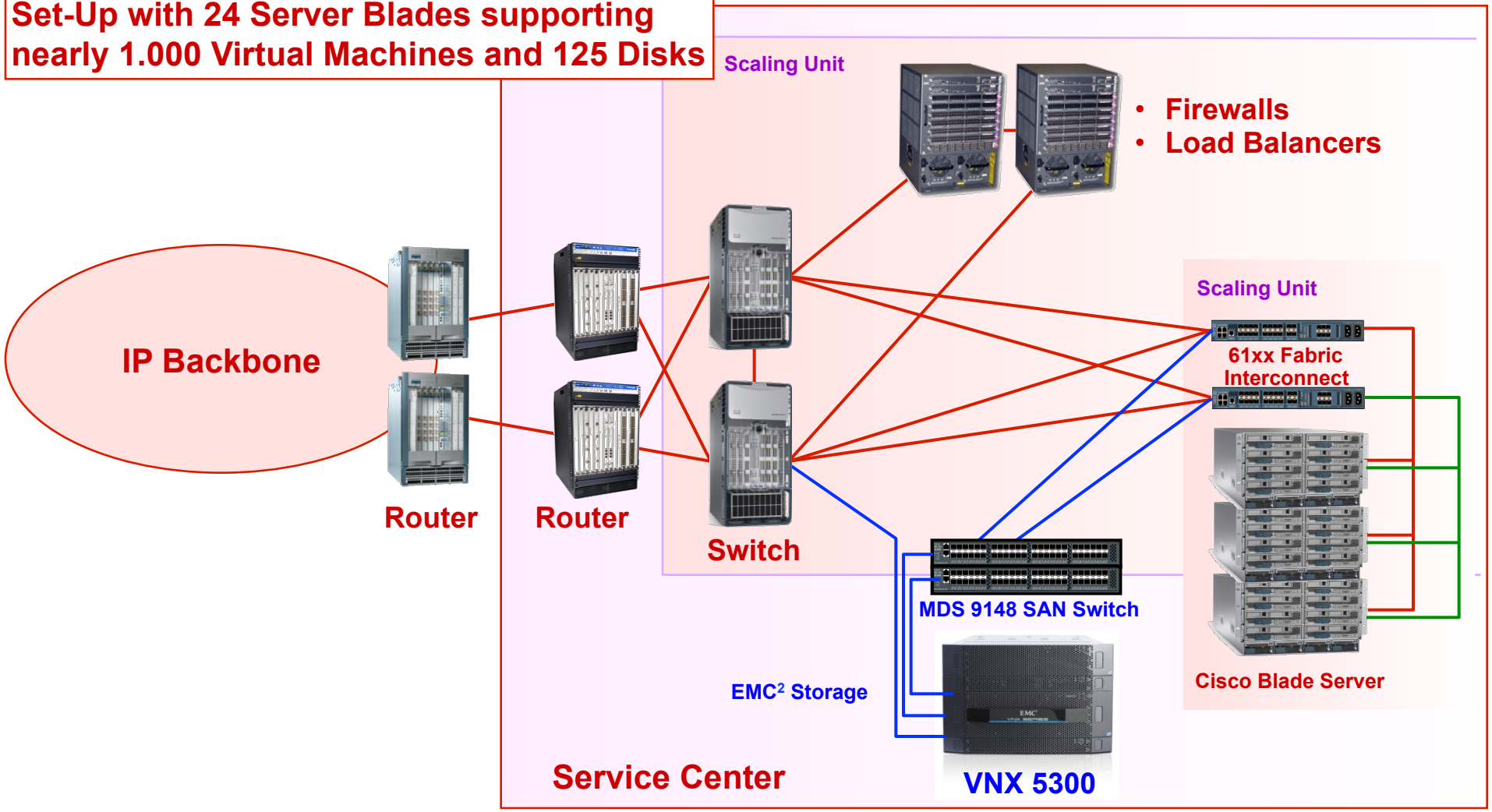
VNX7500
max. 1.000 Drives

- Disk Types: SAS, NL-SAS, Flash
- Protocols: FC, iSCSI, FCoE

Software Defined Networks

Putting all together – Example for a typical Sevice Center Layout

Set-Up with 24 Server Blades supporting nearly 1.000 Virtual Machines and 125 Disks



Software Defined Networks
Summary

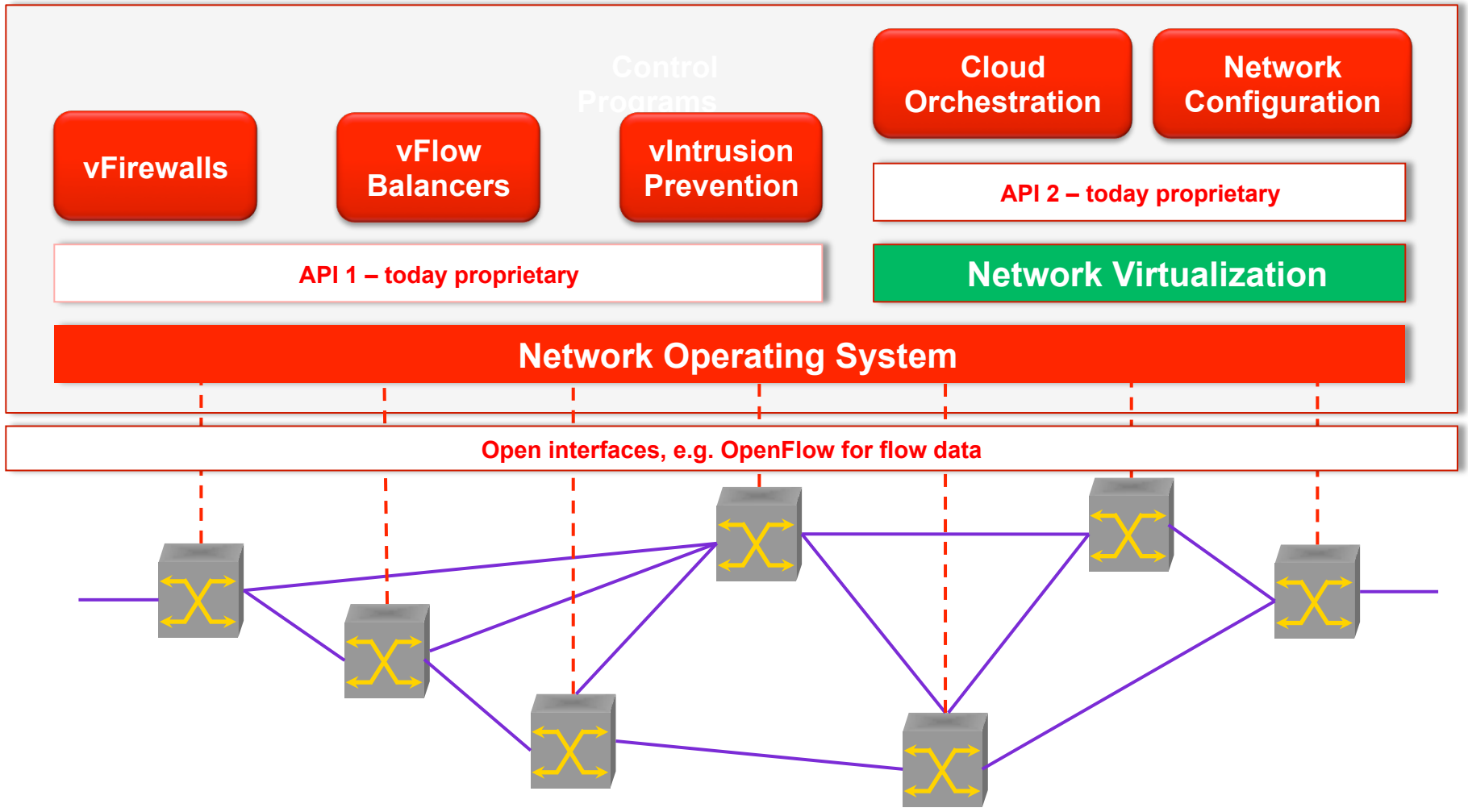
- Even if we already observe some sporadic promising early SDN applications (Google), for most of us SDN is some years away.
- Currently equipment suppliers try to position themselfes with different interpretations of what SDN means. Related Standards on SDN are not yet stable. It‘s still a moving target in the industry.
- SDN in ist original meaning seems to be a bonanza for start-up companies while established telecom vendors like to protect their business by their interpretation of SDN.
- Meanwhile operators focus more on Telco Clouds (network function virtualization) to keep operational costs low while improving and accelerating deployment of new services.


thank you for listening
walter.haeffner@vodafone.com

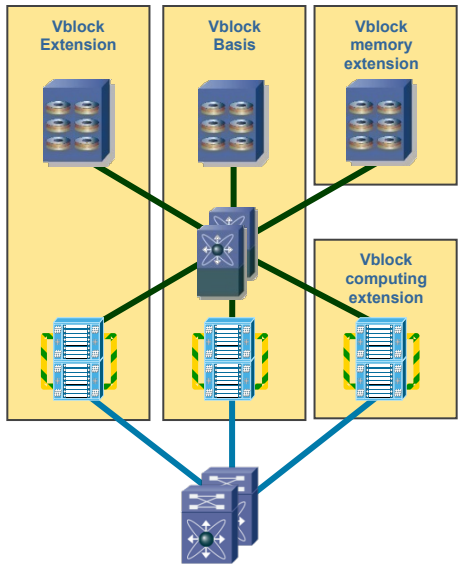
Software Defined Networks

Today Controller and Apps are glued together in a proprietary System

Control plane centralized in a Telco Cloud



Software Defined Networks
Cisco‘s Customizable Cloud Infrastructure - VBLOCKs



- Simple and scalable extensions
- additional memory and CPU according market demand
- fits seamlessly into Vodafone‘s CompanyNet Central Service architecture
- In-service upgrades possible
- Separate storage area network architecture (SANs) with known and standardized platforms
- supports fiber channel over IP (FCoIP), storage and encryption services as well as virtualization services

- **Vblock 2** (from 3.000 VMs to more than 6.000 VMs) for High End Configurations
  most ambitious IT requirements for ERP or CRM systems
- **Vblock 1** (from 800 VMs to more than 3.000 VMs) as Mid Size Configuration
  for many apps, shared services – email, print-shares, virtual desktops
- **Vblock 0** (from 90 VMs to about 800 VMs) as a typical smaller entry configuration
  preferable for software development platforms or test platforms