Software Defined Networking (SDN)

Realitätscheck: Eine Standortbestimmung

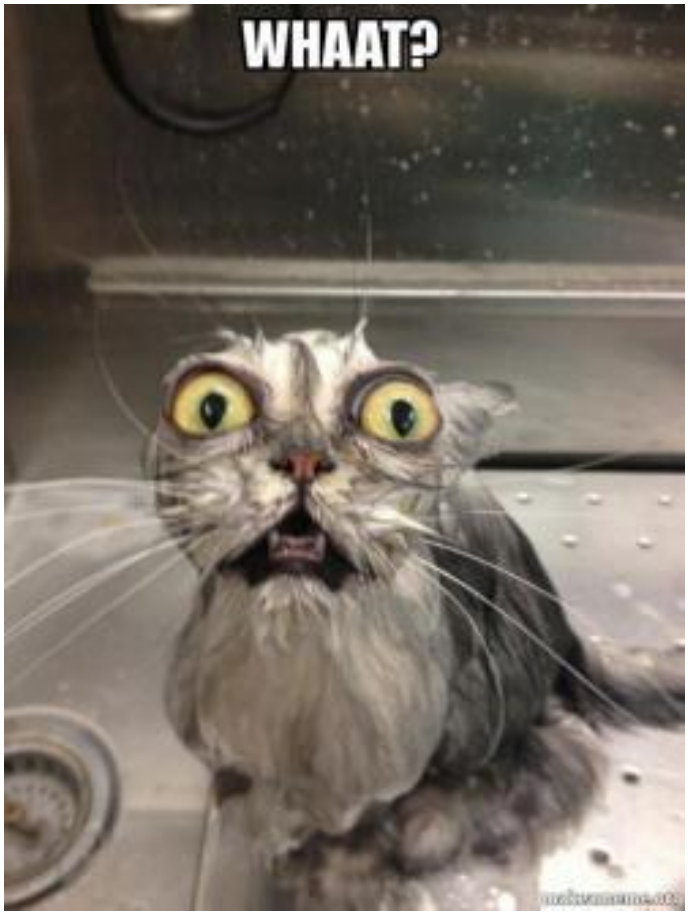
von Peter Infanger

Netzwerk- und Security-Spezialist,
CSS Versicherung & Hochschule Luzern – Informatik

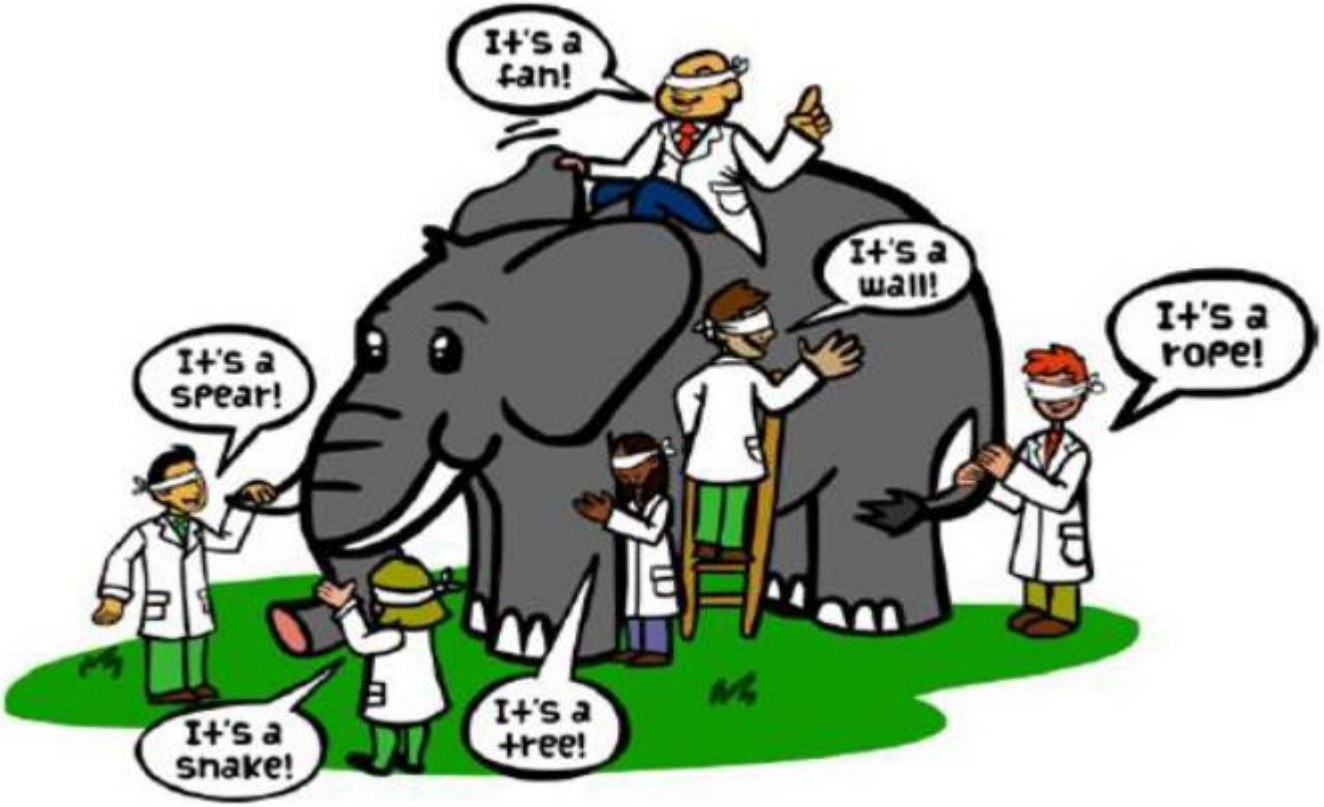
Zitat eines geschätzten Kollegen:

"Bis in ein paar Jahren braucht's keine Netzwerker mehr"

→ die Applikation bestimmt dann weitgehende selber wie sich das Netz zu verhalten hat!



"Schau Dir mal SDN an!"



Quelle: http://www.theblindelephant.com/the_blind_elephant_fable.html

Hintergrund – Ursprung von SDN

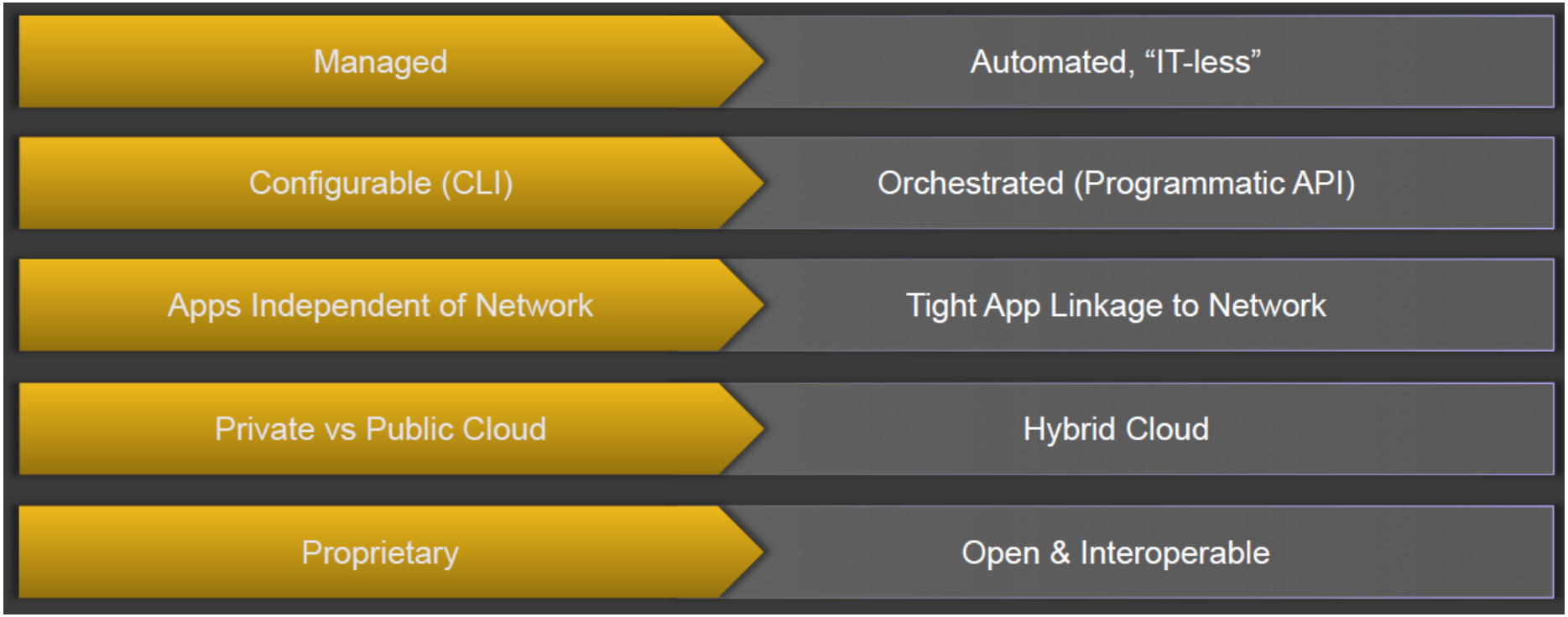
- Konfiguration, Betrieb und Unterhalt des Netzwerkes ist "zu" aufwendig, benötigt Spezialisten, ist wenig flexibel bzw. Veränderungen reinzubringen dauert vergleichsweise lange → zu wenig agil!

So zumindest die Ansicht verschiedener Stellen im Kontext von Etablieren neuer Services, Apps und Dienstleistungen

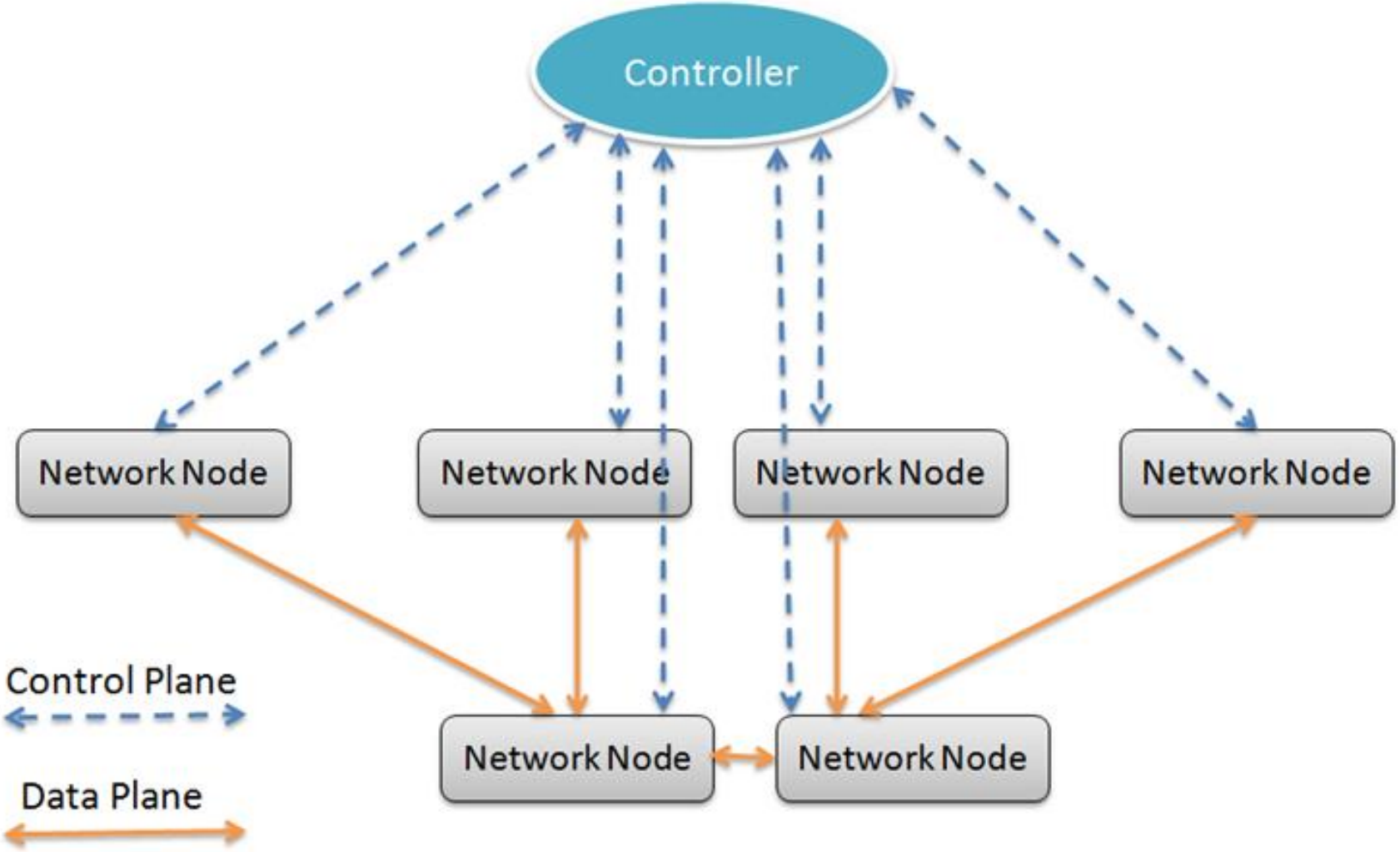
Idee für die Umsetzung

- Physisches Basisnetzwerk mit genügend Bandbreite zur Verfügung stellen, welches mittels einer **zentralen Stelle** je nach Bedürfnis konfiguriert werden kann
  - →Unabhängig davon wo der Serviceanbieter und wo der Servicekonsument sitzt
  - →Automatismen etablieren, welche den Traffic innerhalb des Netzwerkes optimal verteilen
  - →Schnittstellen zum Serviceentwickler anbieten damit der sein Servicemodell direkt ins Netzwerk reinbringen kann

angestrebte Ziele



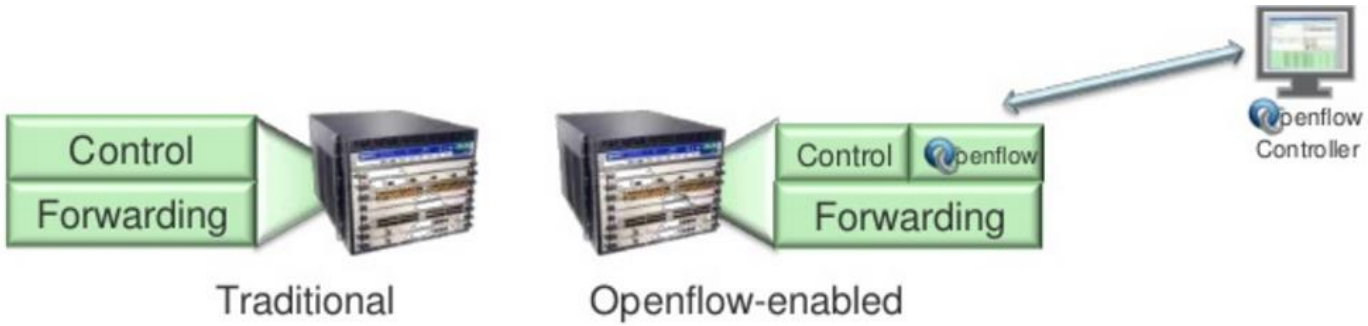
SDN Architektur



Source: http://www.aryaka.com/blog/why-sdn-concepts-need-to-extend-into-the-wan/

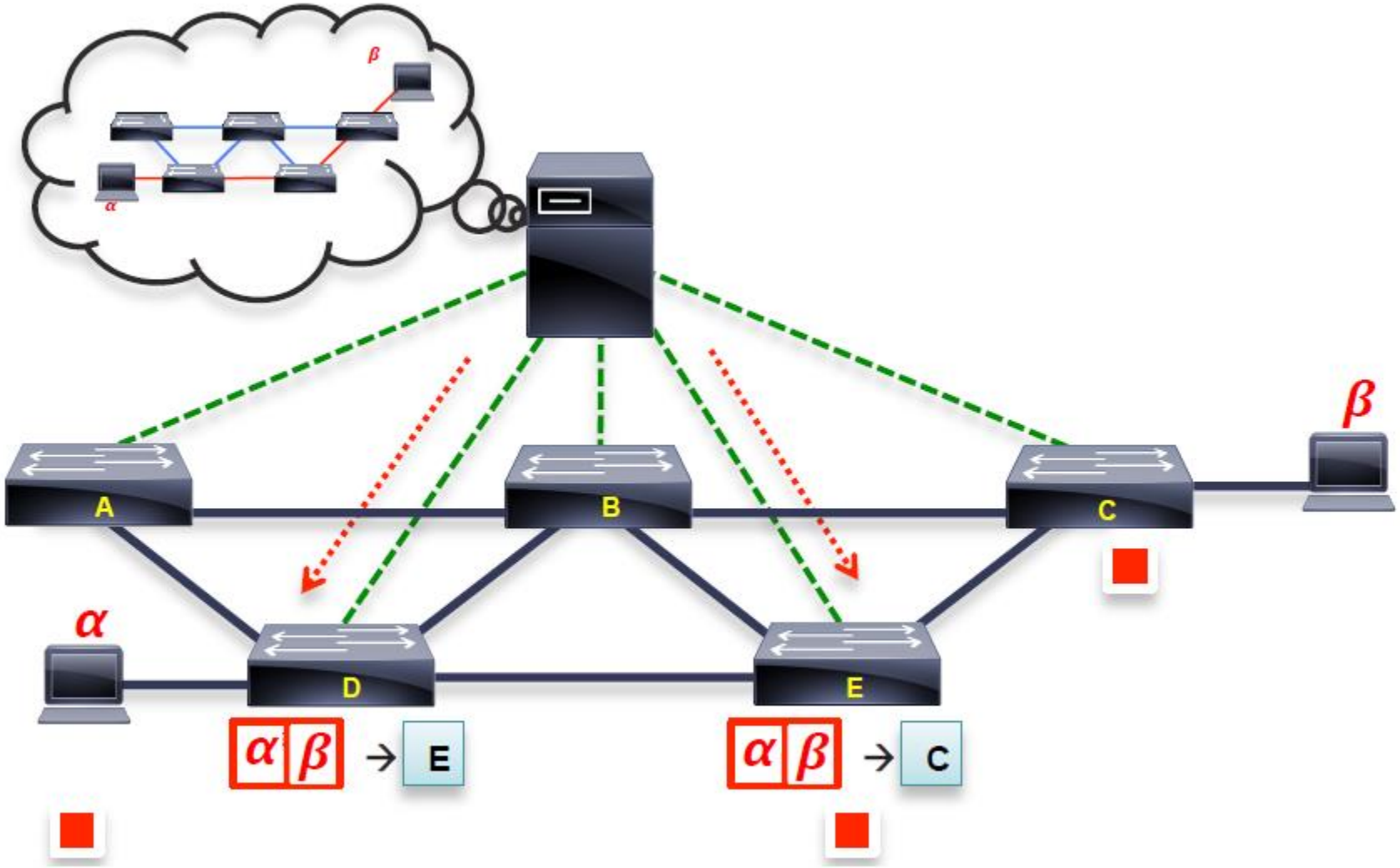
Eine konkrete Umsetzung: OpenFlow

- Mit OpenFlow wird sowohl eine Architektur als auch ein Protokoll bezeichnet



- In der OpenFlow Architektur wird auf dem Network Device eine Schnittstelle zur Verfügung gestellt, welche durch einen externen Controller bedient wird, der die "forwarding-Entscheidungen" trifft

"Per-Flow-Control" Konzept



Component Stack nach OpenFlow

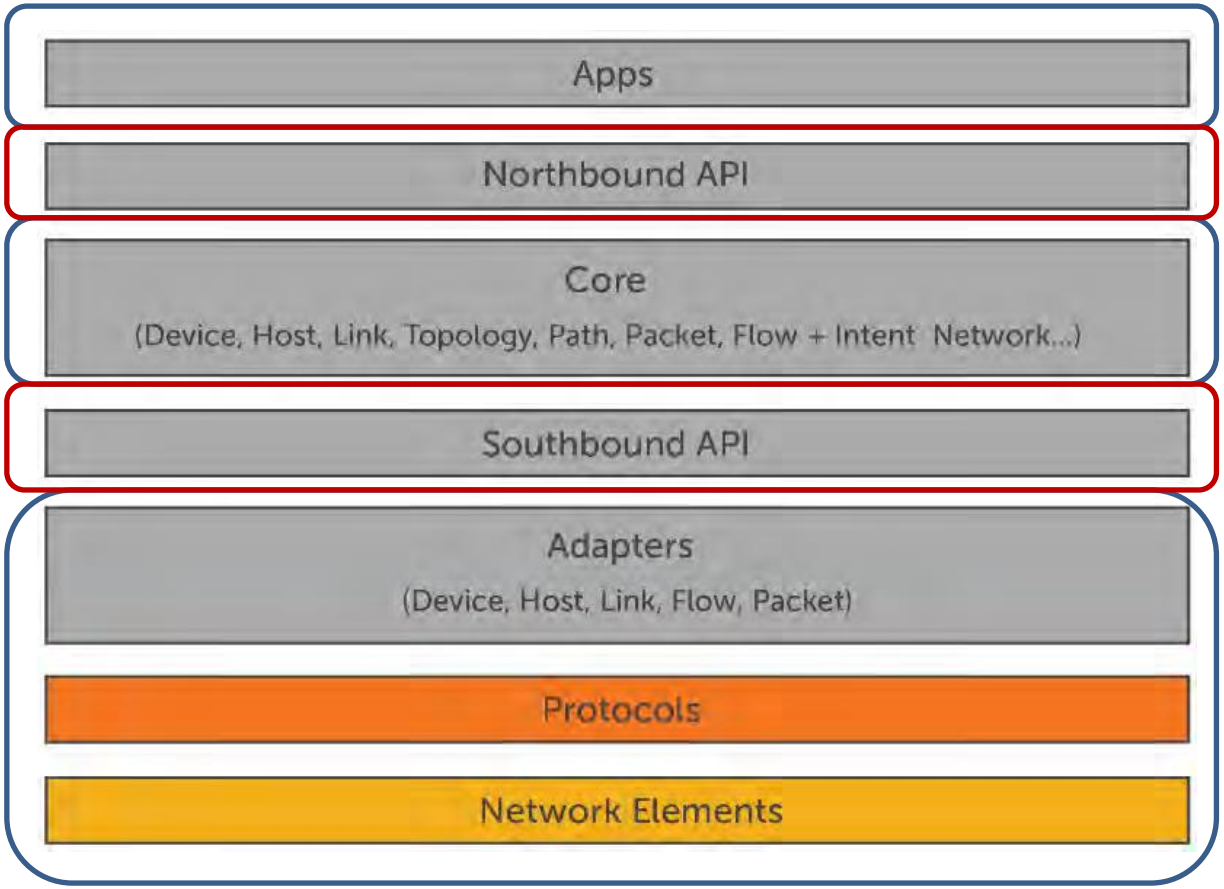
App / Entwicklungs-Umgebung

standardisierte Schnittstelle(n) ?

zentraler SDN Kontroller

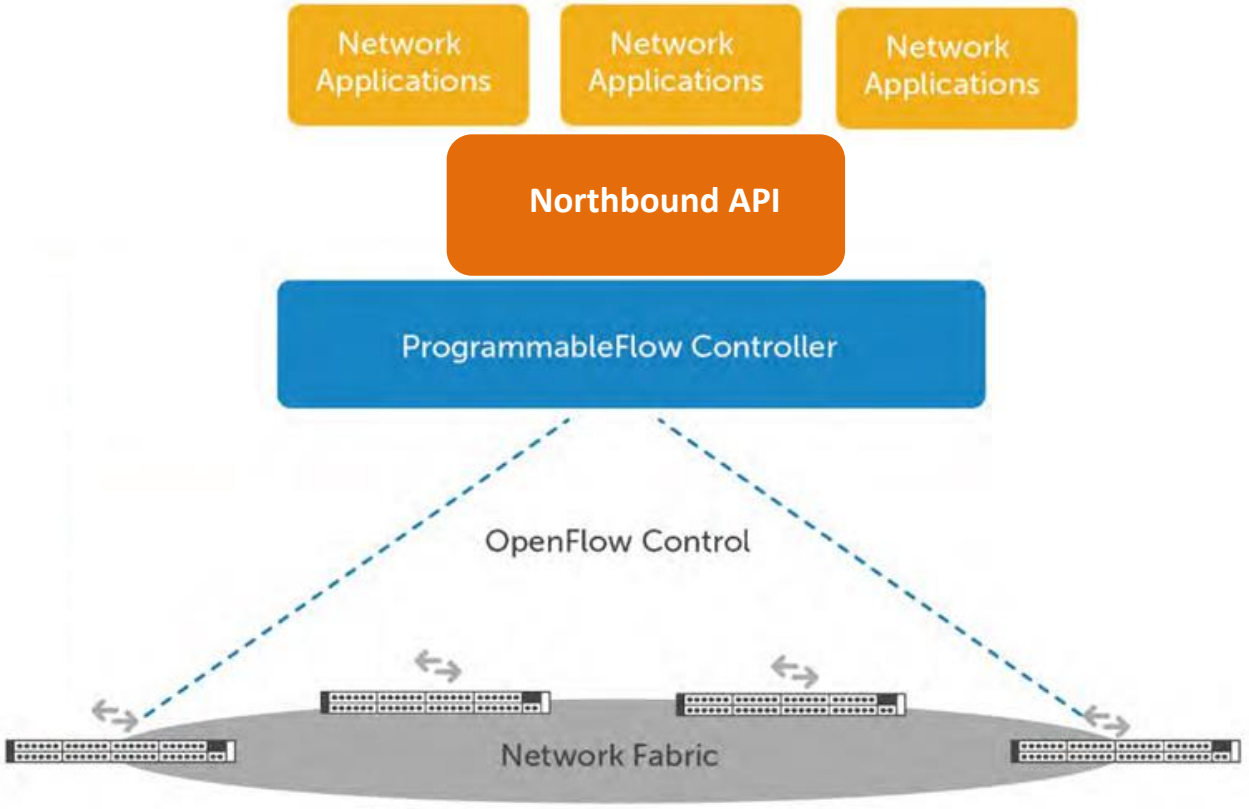
standardisierte Schnittstelle → OpenFlow Protokoll ?

"0815" Hardware (billig, beliebig austauschbar, leistungsfähig, ..)



Ergebnis

- Network Applications sagen dem Netzwerk, wie sie es gerade gerne verwenden möchten



Nun das tönt ja alles cool, aber …

- Was sind das nun denn für Network Applications?
  - Sind diese direkt in die eigentliche Anwendung integriert?
  - Oder müssen dieses zusätzlich zur Anwendung erzeugt werden?
  - Sind sie spezialisiert oder generalisiert?
  - Wie kommen die auf den Controller?
  - Und was ist mit den Basisservices (DNS, DHCP, …)?
  - Wie sieht es dabei mit Sicherheit aus (Zugriffsschutz, etc. …) und wer hat die unter seiner Kontrolle?

Und was macht die Industrie?

- Hersteller agieren unabhängig, d.h.:
  - Terminologie verschieden
  - unterschiedlicher Entwicklungsstand
  - nicht kompatible APIs + Protokolle

  →"Inflation" von Varianten
  - Verfügbarkeit Netzwerk-Komponenten die diese Protokolle unterstützen

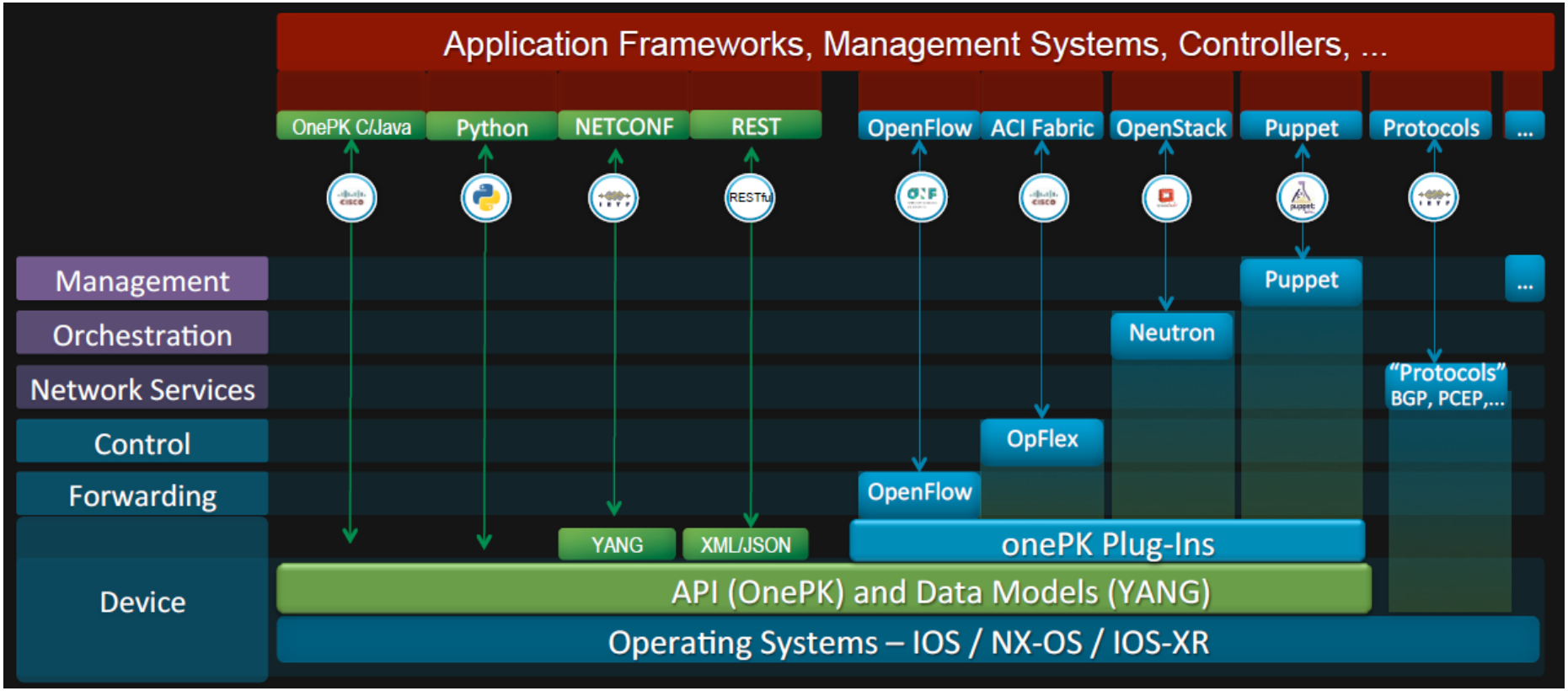
Siehe auch:

- VMware NSX
- Cisco ACI

vmware

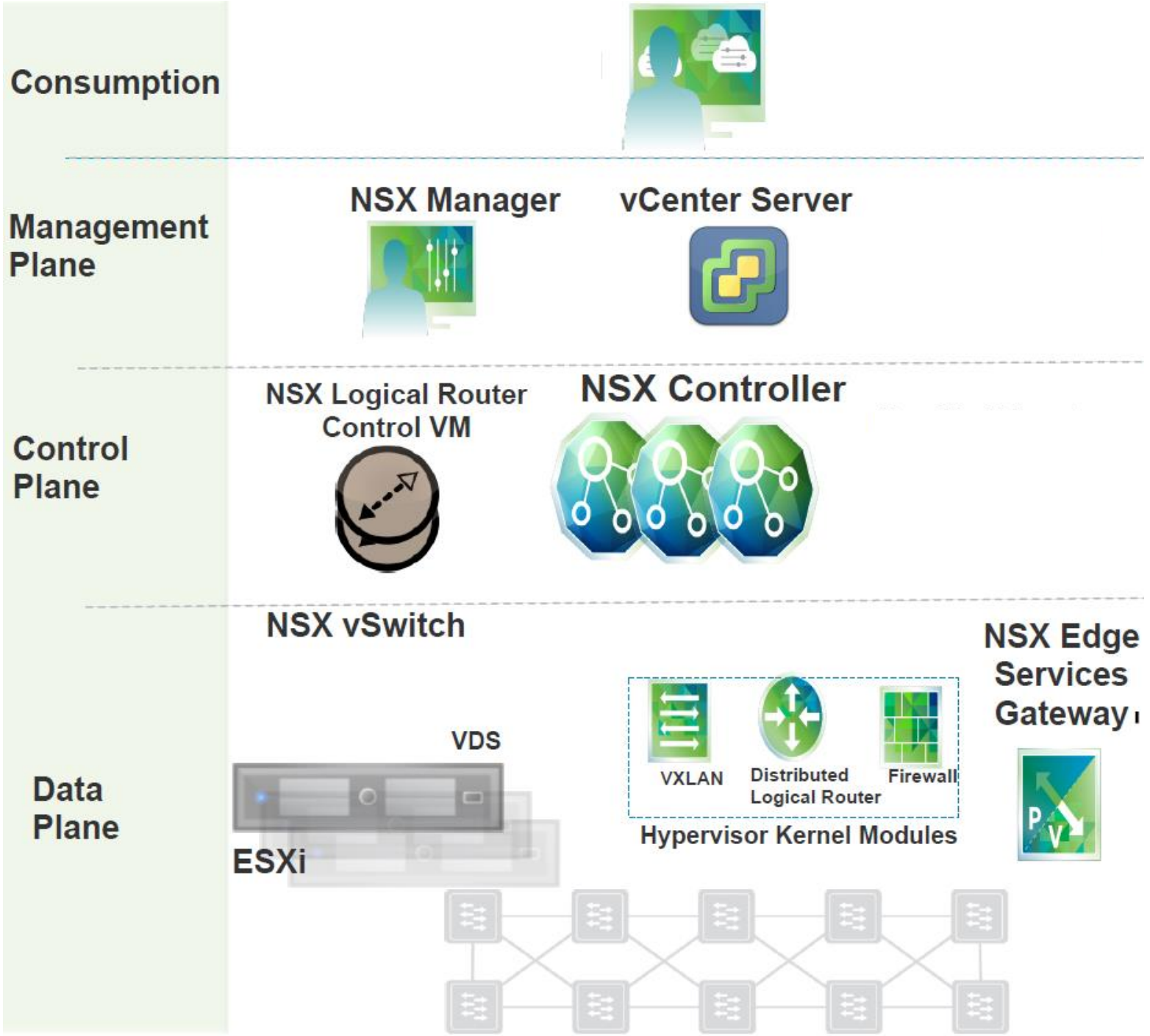
CISCO

"Inflation" – Varianten - Optionen



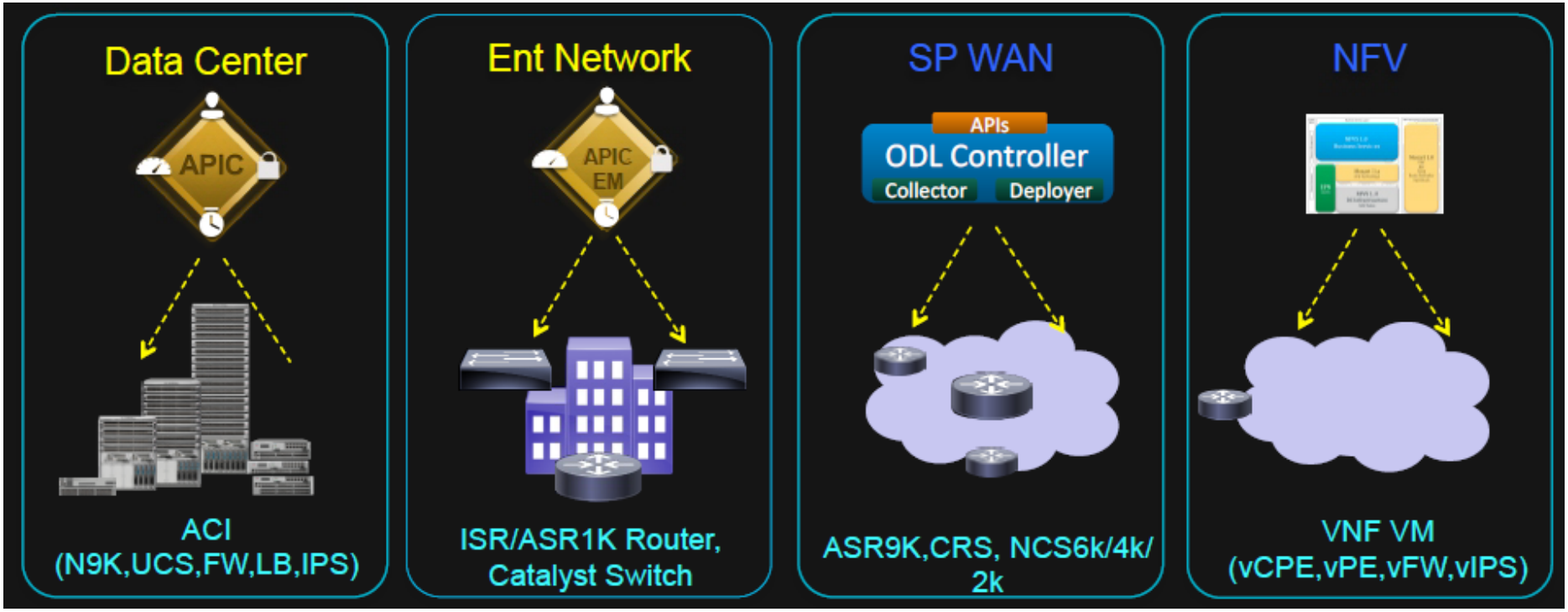
*The YANG data modeling language is used to model configuration and state data manipulated by NETCONF

VMware NSX


Consumption
Management Plane
NSX Manager
vCenter Server
Control Plane
NSX Logical Router Control VM
NSX Controller
Data Plane
NSX vSwitch
NSX Edge Services Gateway
VDS
VXLAN
Distributed Logical Router
Firewall
Hypervisor Kernel Modules
ESXi

Cisco ACI

- SDN Solution Platforms



Aber was bringt das jetzt einem KMU?

- Wie soll man beginnen, wo ist die niedrigste Hürde?
- Man kann nicht oder nur selten auf der grünen Wiese anfangen
  - Bereichsweiser Umbau des Netzes
- Migrationspfad und Parallelbetrieb?
  - Parallelbetrieb → wo spar ich mir den Netzwerker?
- Orchestrierung der Basisservices?
  - Tools fehlen oder unvollständig (oder dann halt klassisch machen?!)
- Entscheidung: OpenSource oder proprietär?

OpenSource Controller

- Verschiedene Projekte am laufen:
  - OpenDaylight
  - Onos
  - ryu
  - …
- Reife der Produkte und Unterstützung der Community sehr unterschiedlich
- Northbound-Schnittstellen wenig koordiniert → Kompatibilität fraglich

Kommerzielle Produkte

- VMware NSX
  - Komplettes, abgerundetes Paket
  - Nur SW → HW offen
- Cisco ACI
  - Einstieg teuer (HW + SW)
  - Hybrid Betrieb der HW nicht möglich
  - Fokussierung auf Bereiche: z.B. WAN, DC, …

Empfehlung: SDN Einsatz in Teilbereichen

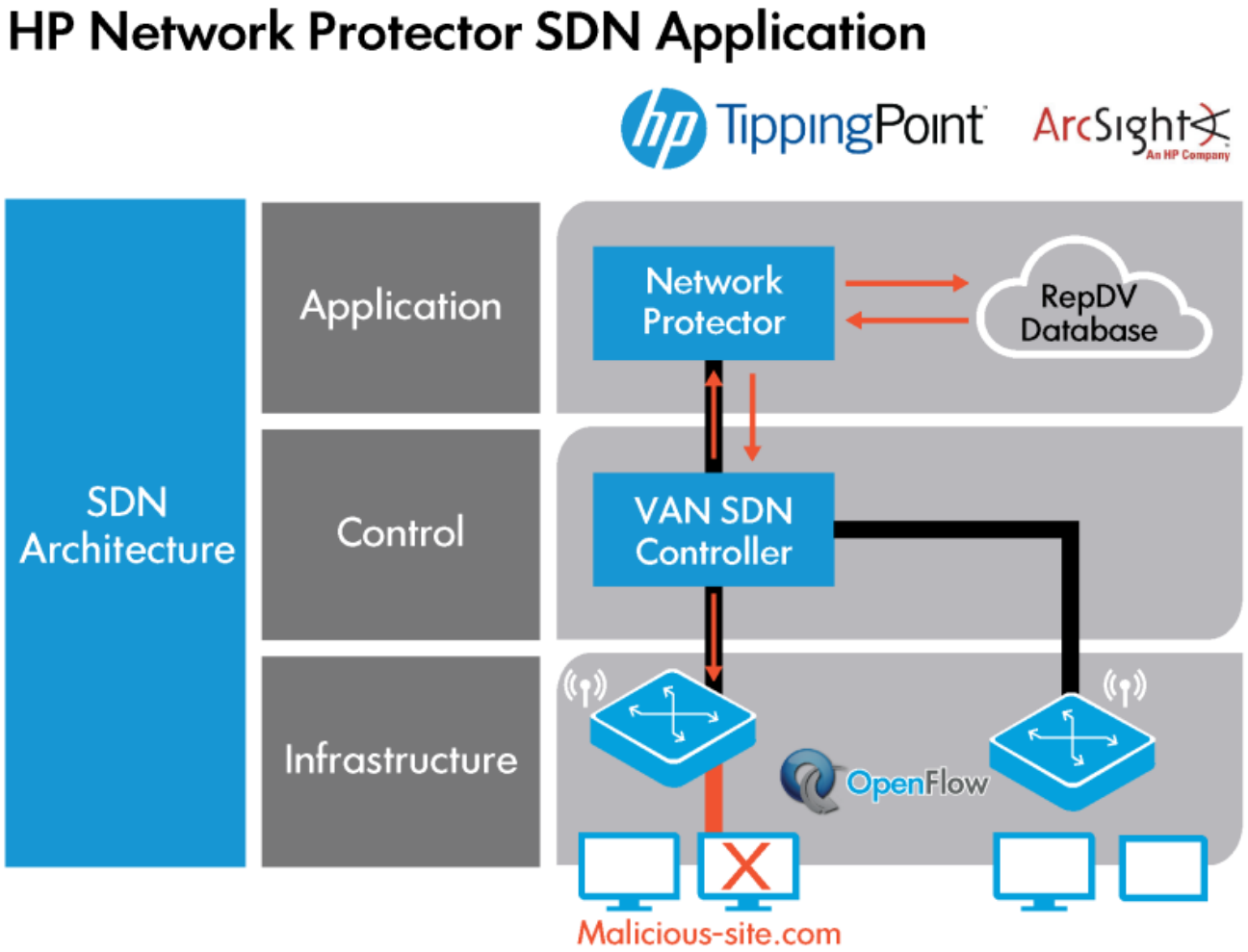
→ Vielversprechenste Weg

- Bei Neu-Beschaffungen von Infrastruktur entsprechende Produkte berücksichtigen (Kompatibilitätscheck nicht vergessen!)
- Klein anfangen, gut abgrenzen und Kriterien definieren, welche man erreichen möchte
- Auswahlkriterien der Bereiche: Agilität, Security, BW- Optimierung, …

Netzwerker miteinbeziehen und ggf. ausbilden!

Beispiel für einen abgegrenzten Einsatz

Jeder DNS-Request wird "abgefangen" und einem Reputation-Check unterzogen.



Weitere Empfehlungen für den SDN Einsatz

- OpenSource in Betracht ziehen
  → Einstiegshürde (Kosten) geringer!
  → Funktionalität muss darunter nicht leiden!
  → Aufwand dafür höher
- Hybride Netzwerkkomponenten verwenden
  → mehr Variabilität → Migration möglich
- Mit den Entwicklern zusammensitzen und deren Bedürfnisse abholen (z.B. wenn beim DevOps höhere Agilität gefragt ist)

SDN im real life

- Beispiele
  - QoS Steuerung bei Skype for Business (https://www.youtube.com/watch?v=T8TTB2hAOWw)
  - Google's SDN WAN (http://www.networkcomputing.com/networking/inside-googles-software-defined-network/512240144)
  - Microsoft SDN in Azure (https://docs.microsoft.com/en-us/windows-server/networking/sdn/software-defined-networking)
- Interessante Anwendung
  - HP Network Protector http://h20564.www2.hpe.com/hpsc/doc/public/display?docId=emr_na-c04626978)

Fazit

- SDN wird kommen – grosses Potential
- momentan Reife der Produkte (OpenSource) noch nicht gegeben
- Network Applications werden vielfältiger und werden Themen wie Security weitreichend abdecken
  → Standardisierung muss aber noch verbessert werden
- Bereich der Orchestrierung, besonders bei Open-Source, stark verbesserungswürdig!

- Es wird mich als Netzwerker noch länger brauchen!

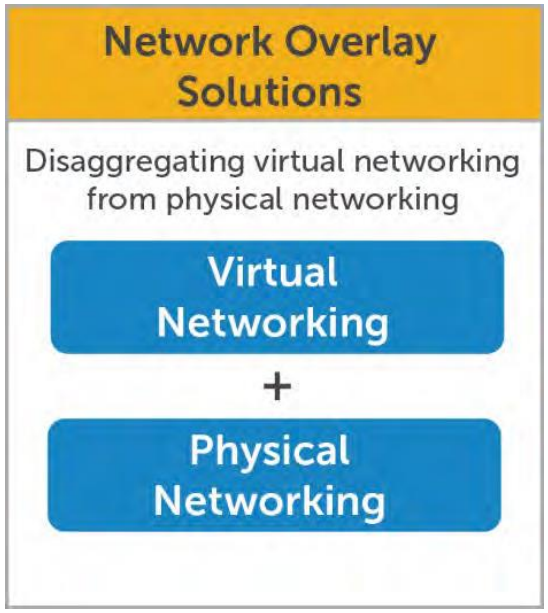
Backup Slides

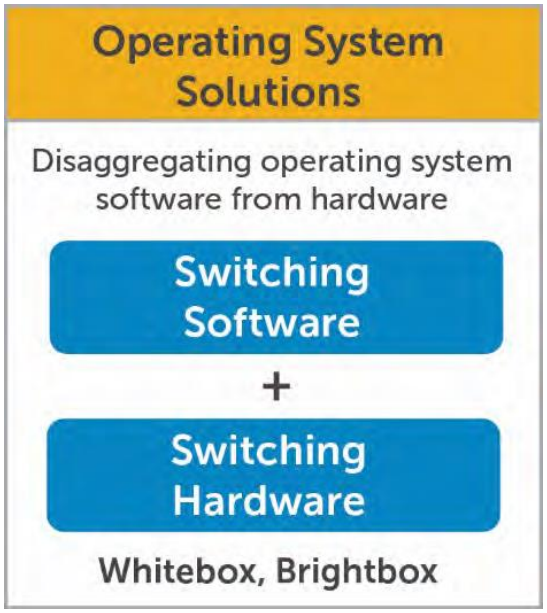
Was ist SDN?

- Software Defined Networking (SDN) ist ein neuartiger Ansatz für das Networking, bei dem die Netzwerksteuerung softwaremässig erfolgt und von der Hardware entkoppelt ist.

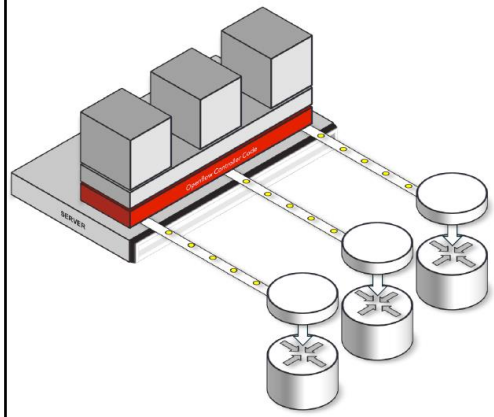
SDN und womit wir es da zu tun haben

- Ziele:
  - "Disaggregating Virtual Networking and Physical Networking"
  - "Disaggregating the operating system and hardware"
  - "Disaggregating the control plane and data plane"



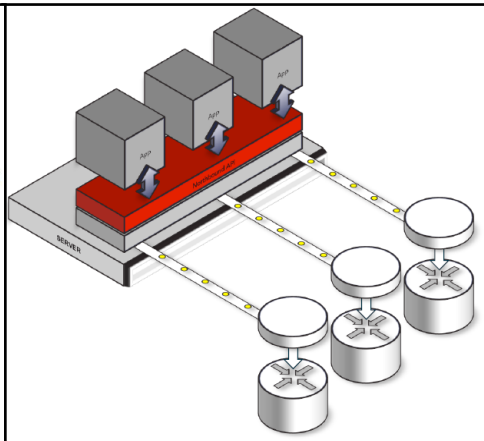


OpenFlow – 4 Komponenten



Openflow Controller

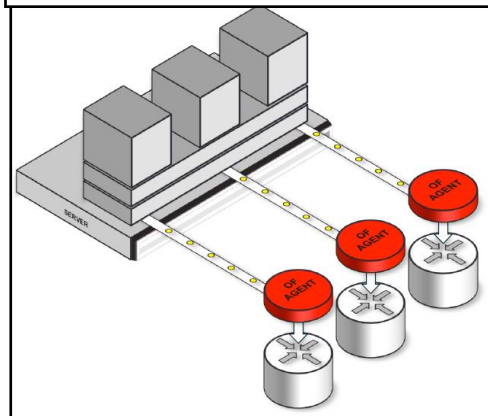
Central Administration and Operations point for Network Elements



Openflow Controller | Northbound API

Northbound API
Integral part of Controller

"Network enabled" application can make use of Northbound API to request services from the network…

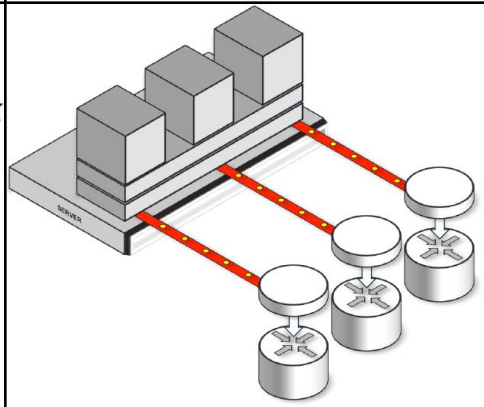


Openflow Device Agent

Agent runs on the network device

Agent receives instructions from Controller

Agent programs device tables

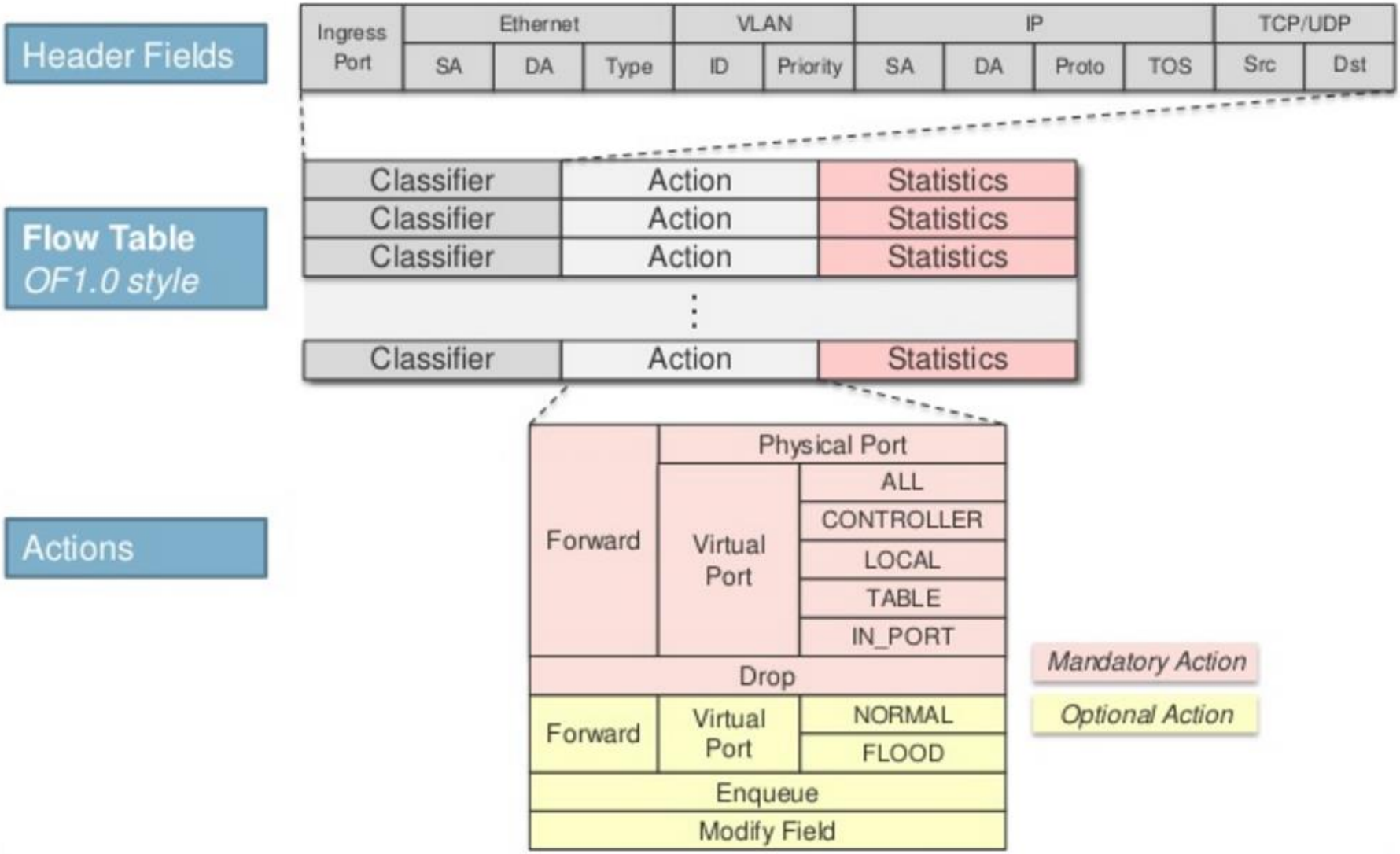


Openflow Protocol

Openflow Protocol is…

"A mechanism for the Openflow Controller to communicate with Openflow Agents…"

OpenFlow (1.0): Tabelle & Felder

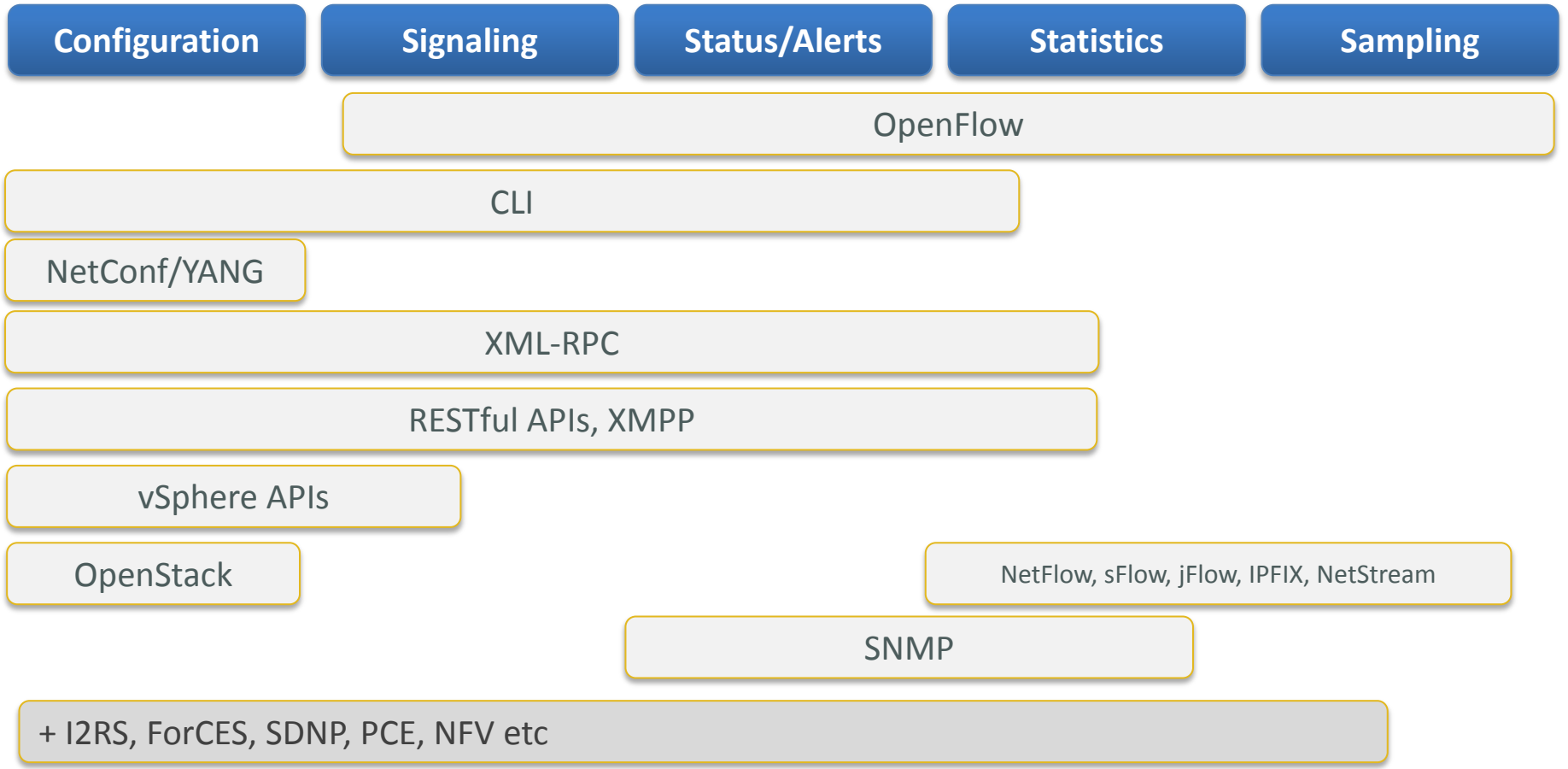


OpenFlow Basics

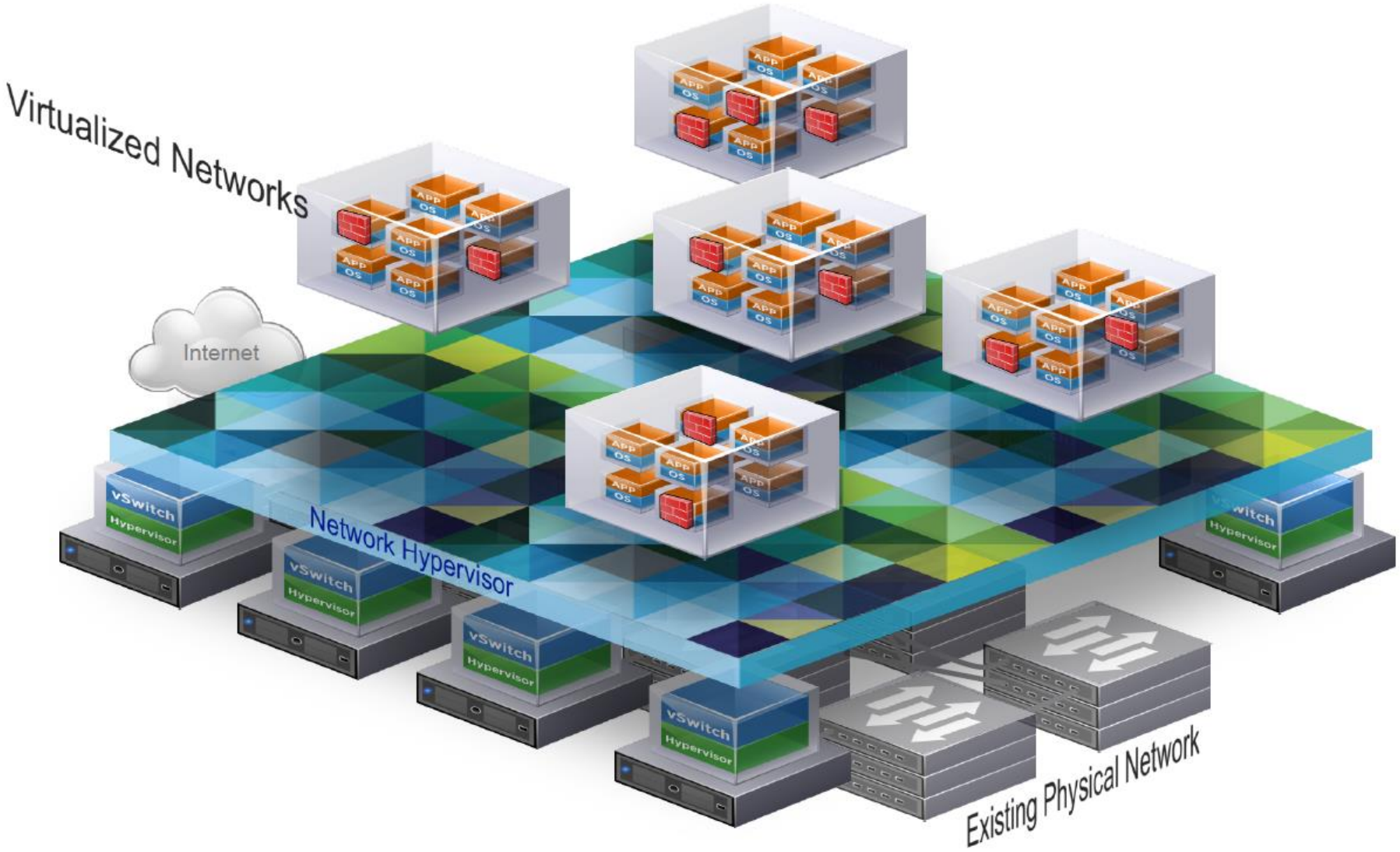
- Kontrolliert die FlowTabelle von Switches, Routers, ChipSets, etc.

OpenFlow & Rest der Welt

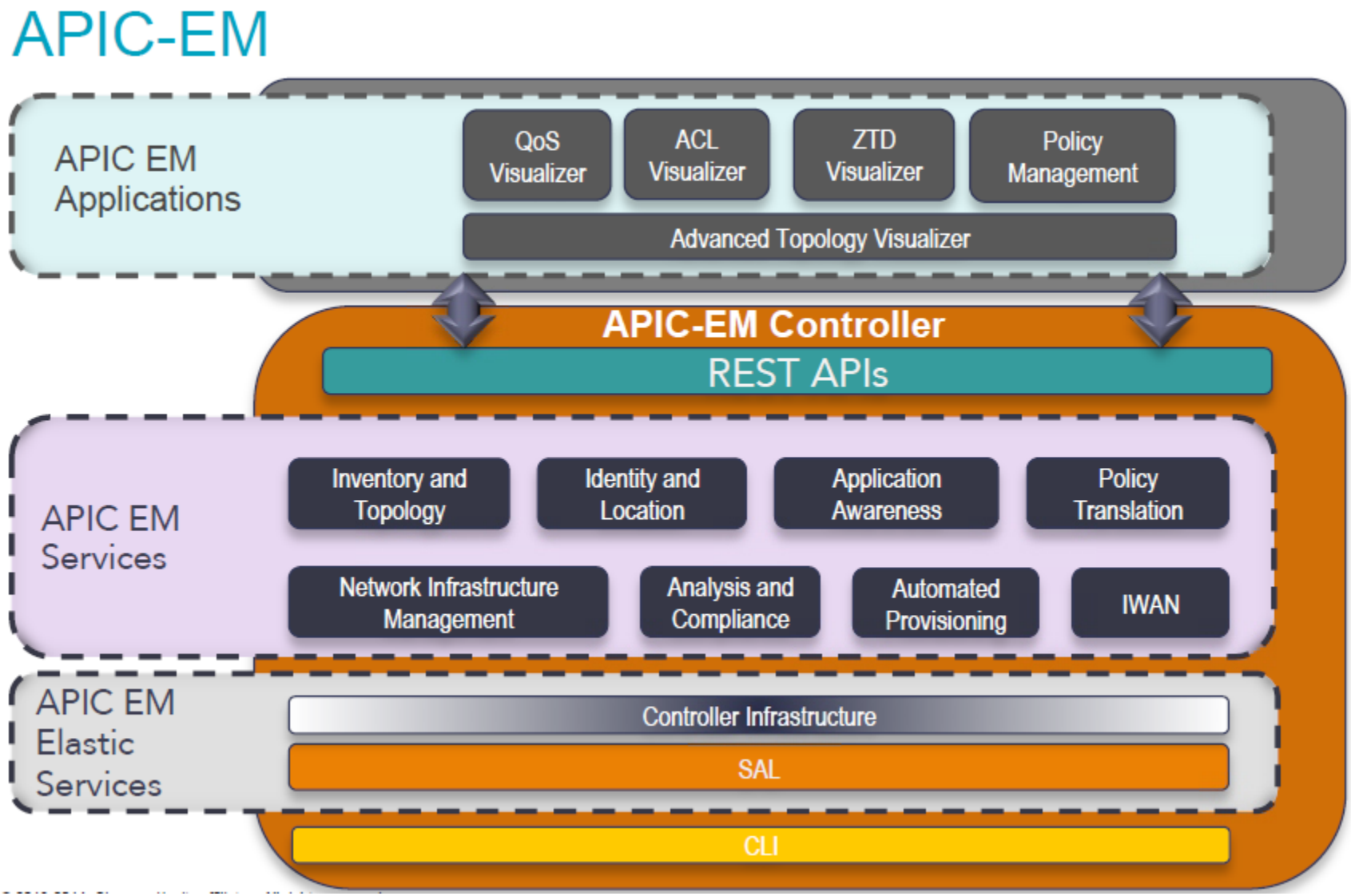
- Wer deckt was ab?



VMware NSX


Virtualized Networks
Internet
Network Hypervisor
vSwitch
Hypervisor
Existing Physical Network

ACI Stack


APIC-EM
APIC EM Applications
QoS Visualizer
ACL Visualizer
ZTD Visualizer
Policy Management
Advanced Topology Visualizer
APIC-EM Controller
REST APIs
APIC EM Services
Inventory and Topology
Identity and Location
Application Awareness
Policy Translation
Network Infrastructure Management
Analysis and Compliance
Automated Provisioning
IWAN
APIC EM Elastic Services
Controller Infrastructure
SAL
CLI

Integration in Entwicklungsprozess