



# Disaggregated Datacenter Networks

Auf dem Weg zu neuen Vendors

**” Dieser Talk gibt einen Überblick  
über die neue Welt des  
Disaggregated Networkings“**

**” Es werden einige Konzepte  
diskutiert. Jedoch wird nicht in  
die Tiefe eingestiegen! “**

# **Eine Geschichte: Die Uni Ulm braucht ein neues Datacenter!**

# Aus alt mach neu!

## Datacenter der Universität Ulm kam in die Jahre!

1. Alte Infrastruktur.
2. Viel verworrener Spanning-Tree.
3. Kopplung an Campusnetzwerk mit großer Failure-Domain.
4. Mangelnde Flexibilität bei der Anbindung redundanten Anbindung von Diensten.
5. Basiert auf "Legacy-Technologien": Spanning-Tree, Active-Passive-Bonds usw.

Das muss weg 🔥



# Design-Prinzipien

## Möglichst kein Vendor Lock-In:

- Offene Standards.
- Kein Vendor NMS.
- Offene Schnittstellen.



## Maximale Flexibilität:

- Routing/Switching in einem.
- Möglichkeit, große Anzahl an Nutzerkontexten zu bewältigen.
- Redundancy by design.
- “Everything, everywhere, all of the time!”



## Leichte Erweiterbarkeit:

- Ankopplung an bestehendes Campusnetzwerk.
- “Pluggable Architektur”



**Wir sind offen an die Sache ran  
gegangen!**

# Klassische Vendors: Case Studies



## #1 MLAG-based Architecture (Cisco VPC, Aruba, Arista).

👍 Bessere Redundanz.  
👍 Relativ günstig.

👎 Einzelne Switches nicht independent.  
👎 Asymmetrische Routingzustände möglich.  
👎 Proprietär  
👎 L2-zentrisch.



## #2 Vollintegrierte Lösung (Cisco ACI, ...)

👎 Funktioniert (irgendwie).

👎 Hochgradig Proprietär.  
👎 Kann Universitäts-usecases nicht.  
💰 Sehr teuer.



## #3 EVPN/VXLAN

👍 Sehr flexibel. (vgl IB!)  
👍 "One Overlay to rule them all".  
👍 Prinzipiell offener Standard.

👎 Verschiedenste Feature-Portfolios.  
👎 Bugs, Bugs, Bugs.  
💰 Sehr teuer!

# Here's what we found:

## Klassische Vendors bieten:

- Vendorspezifische Lösungen (ACI usw.).
- Aber auch offene Lösungen (EVPN/VXLAN).
- Automatisierung oft schwierig.
- Lizenzmodelle aus der Hölle!
- Sehr teure Hard- und Softwarepakete!
- Verbuggte Software.
- Teurer Support.

 **Lock-In sein Urgroßvater!**

**Worst offender:** Keine perpetual Licenses, Zwang zu Lizenzserver.

**” Klassische Vendors wollen  
unseren Markt nicht bedienen.**

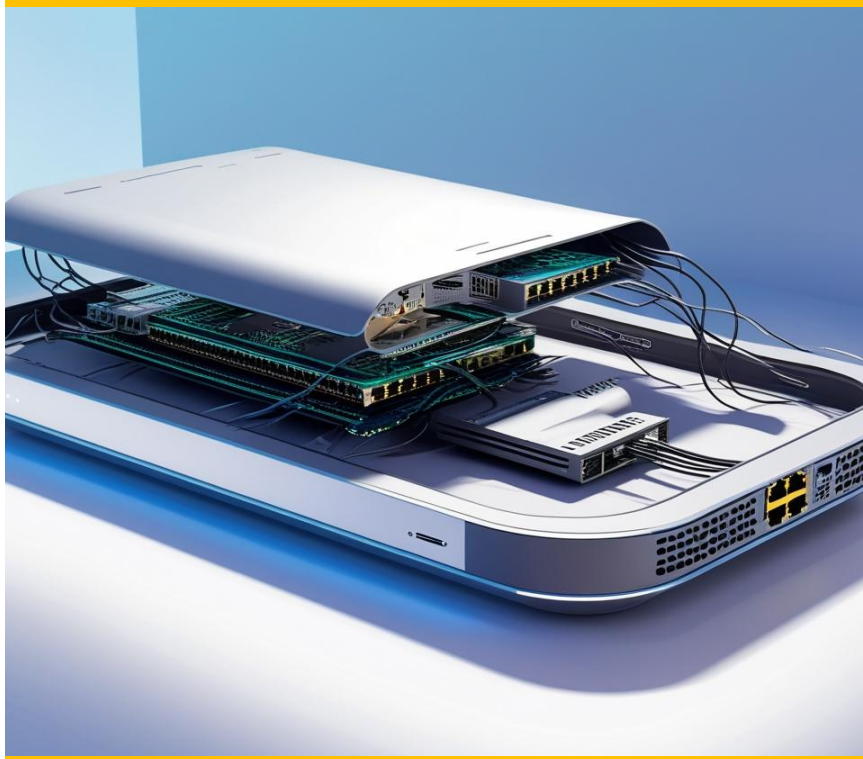
**“**

# Vendor Lock-In quasi unvermeidbar!



**Was gibts denn dann noch?**





## Disaggregated - zerfallen!

Trennung von:

- Hardware und Software!
- Dataplane und Controlplane!
- Herstellern/Vendors!

# Klassische Netzwerkkomponenten



## **Hardware Platform:**

Entweder eigenes Silicon/ASIC oder zugekauft von Broadcom



## **System Software:**

Proprietäres Betriebssystem (NXOS, IOS, EOS, CX...).



## **Network Management System (NMS)**

Meist auch proprietär.

# Disaggregated Networking



## Hardware Platform:

“Whitebox Router”



## System Software:

Auswahl aus verschiedenen Betriebssystemen (OcNOS, Cumulus, Sonic, PicOS)



## Network Management System (NMS)

Offene Standards (Restconf, Netconf)

# Open Compute Project

## “Sharing and Defining Data Center Architecture Components”

- Getrieben von den Hyperscalern.
- Offene Standards für alles im DC-Bereich.
- Auch: Networking Standards!

**Ursprünglich:** Facebook will eigenes OS auf Switching Hardware.

**Heute:** Offener Standard für Hardware-Plattform.  
Viele verschiedene Softwarepakete verfügbar!



**Ist das ein Traum?  
Jein.**

Wen gibts denn da so (Software)?

PIC7A8

 BROADCOM®



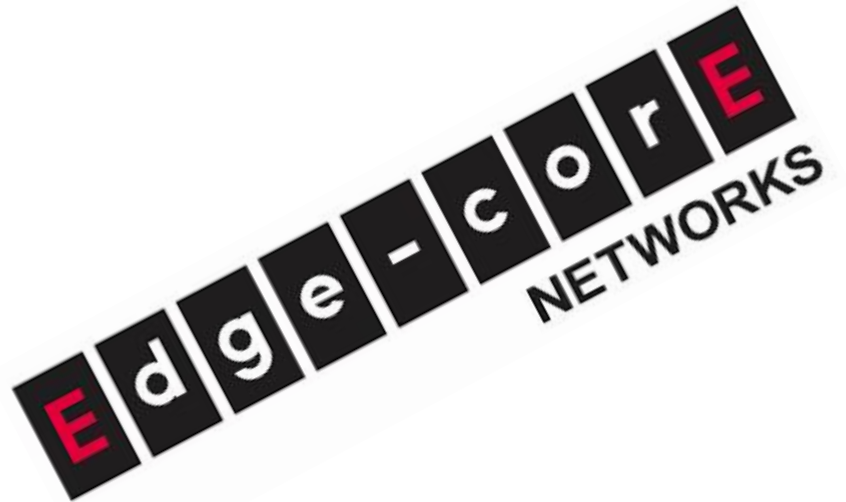
NVIDIA®

ipinfusion™

Wen gibts denn da so (Hardware)?



**ufiSpace**



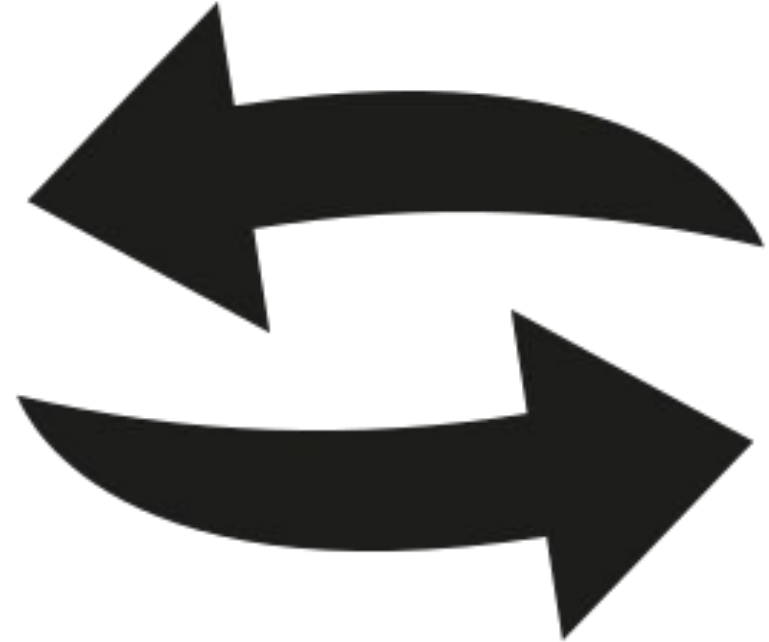
# Disaggregated Networking bietet viele Vorteile:

- Viel Auswahl an Hardware und Software verschiedener Hersteller.
- Möglichkeit, Software auszutauschen!
- Preislich oft sehr attraktiv.
- *Guter* Support!

## Aber:

- Nicht jede Software läuft auf jeder Hardware.
- Sehr starker Fokus auf wenige, aber innovative Technologien.
- Weniger breites Feature-Set im Vergleich zu klassischen Vendors.
- Doku teils mau.

**Daher:** Netzwerk OS legt ein Stück weit die Architektur fest.



# Gemeinsamkeiten

## Hardware:

- Oft basierend auf Broadcom Trident/Qumran/Tomahawk. (Vgl Nexus 3000)
- In ähnlicher Konfiguration von verschiedenen Herstellern verfügbar (Edgecore, UfiSpace...)
- Verschiedenste Portkonfigurationen verfügbar.

## Software:

- Meist ausgerichtet auf große Fabrics mit MPLS/EVPN.
- Einfache automatisierbarkeit.



## Datacenter mit UfiSpace und IPInfusion Ocnos

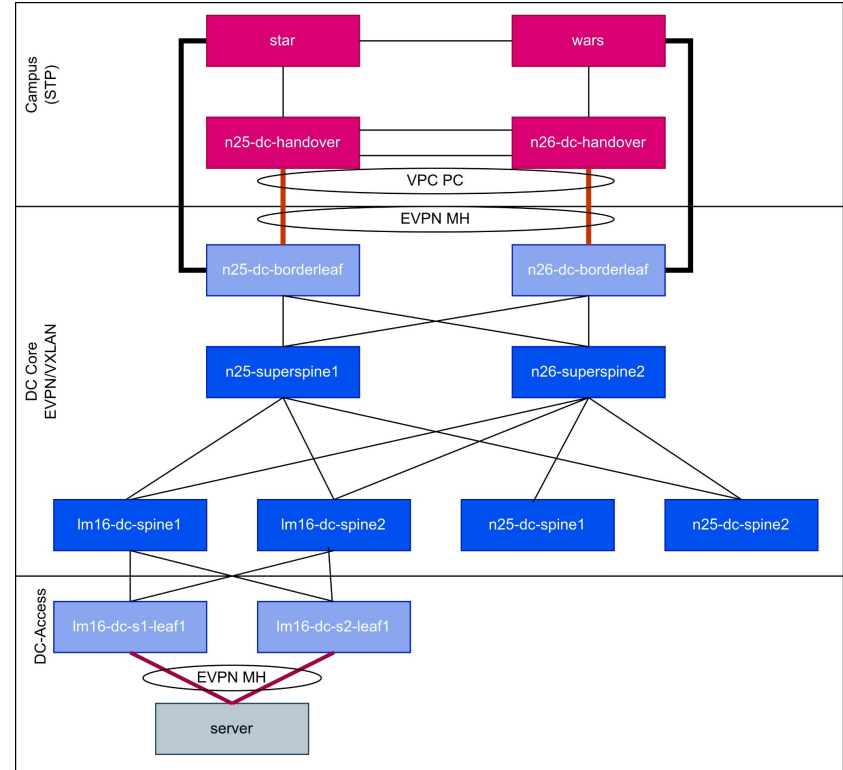
Ein Erfahrungsbericht

# DC Architektur mit UfiSpace + OcNOS

## Grundsätzlich:

- CLOS-Fabric.
- EVPN/VXLAN Overlay Netzwerk.
- Loadbalancing auf allen Links.
- EVPN Multihoming. Kein proprietäres MLAG.
- Handover Campus:
  - Routed für IP
  - EVPN MH

## Standard Datacenter Architecture!



**Früher: SNMP für Metriken.  
Für Config?**

# Automatisierung und CI

## Früher:

- Ausrollen von “Config-Files”.
- Config-Replace meist eher mäßig gut.
- Kontext-sensitiv.
- Unstrukturierte Files, schwer zu parsen.

## Zwischendurch:

- Schlimme, vendorspezifische REST-APIs.
- Keine Standardisierung.

**Jetzt: Netconf/YANG**

**(Kommt bei fast allen Disaggregated OS mit!!)**



# Automatisierung und CI

## Datenmodell: YANG

- Vergleiche “MIBs” bei SNMP.
- Abbild aller Stateinformationen eines Switches.
- Nützlich für Monitoring und Konfiguration.

## Datenformat: XML

- Formulierung von “RPC-Calls” mit Anweisungen.
- Strukturierte Übermittlung von Daten.

## Transport: SSH.

- Einfach, kennt jeder.

**Endlich ein offener Standard!**  
**(Und kann Config Replace!)**



# Streaming Telemetry

**SNMP: Pull-Protokoll!**

**Was aber, wenn ich lückenlose Daten haben  
will?**

**Streaming Telemetry!  
(Offener Standard)**



**” Netconf ist kein Afterthought im  
Disaggregated Networking.**

**“**

```
superspine1#sh xml running-config
<netconf-server xmlns="http://www.ipinfusion.com/yang/ocnos/ipi-management-server">
  <vrfs>
    <vrf>
      <vrf-name>default</vrf-name>
      <config>
        <vrf-name>default</vrf-name>
      </config>
      <netconf-ssh-config>
        <config>
          <feature-netconf-ssh>>false</feature-netconf-ssh>
        </config>
      </netconf-ssh-config>
      <netconf-tls-config>
        <config>
          <feature-netconf-tls>>false</feature-netconf-tls>
        </config>
      </netconf-tls-config>
    </vrf>
    <vrf>
      <vrf-name>management</vrf-name>
      <config>
        <vrf-name>management</vrf-name>
```

# Mit Netconf automatisieren?

## Zwei Möglichkeiten:

- 1) **YANG Modell nutzen.**
  - a) Kompletten Switch als Datenstruktur.
  - b) Befüllen der Datenstruktur aus SoT.
  - c) Serialisieren als XML/JSON.
  - d) Deployment auf Switch.
  
- 2) **“Poor Man's(m/w/d) Solution”**
  - a) Deploy Feature von Hand.
  - b) `sh xml run`
  - c) Baue Template (jj2)
  - d) Generiere mit Ansible/whatever.

**Man kommt sehr schnell zu was Sinnvollem.**



# Die Sache mit dem Support

## Support ist Mixed Bag:

- FS.com war sehr bemüht.
- Broadcom hatte hauptsächlich shiny slides.
- nVidia sprach nicht mit uns.
- EPS-Global/IPI: Außerordentlich gut!

**Für uns ausschlaggebend!**



# Licensing - Burning Bridges of the Bay

**Kurz zusammengefasst:**

**Bei allen Anbietern sehr fair.**

**Hardware Service:**

**Lieber Geräte auf Halde kaufen!**



**” Man fühlt sich wie in einer  
neuen Welt. “**

# Key Takeaways

## Was gibt einem Disaggregated Networking?

- Viel Flexibilität.
- Modernes, aber nicht allumfassendes Feature-Set.
- Faires Licensing.

## Was braucht man?

- Know How!

**Disaggregated Networking ist eine echte Alternative.**

**Danke für die Aufmerksamkeit!**

**Mail: [johannes@jaydi.de](mailto:johannes@jaydi.de)**