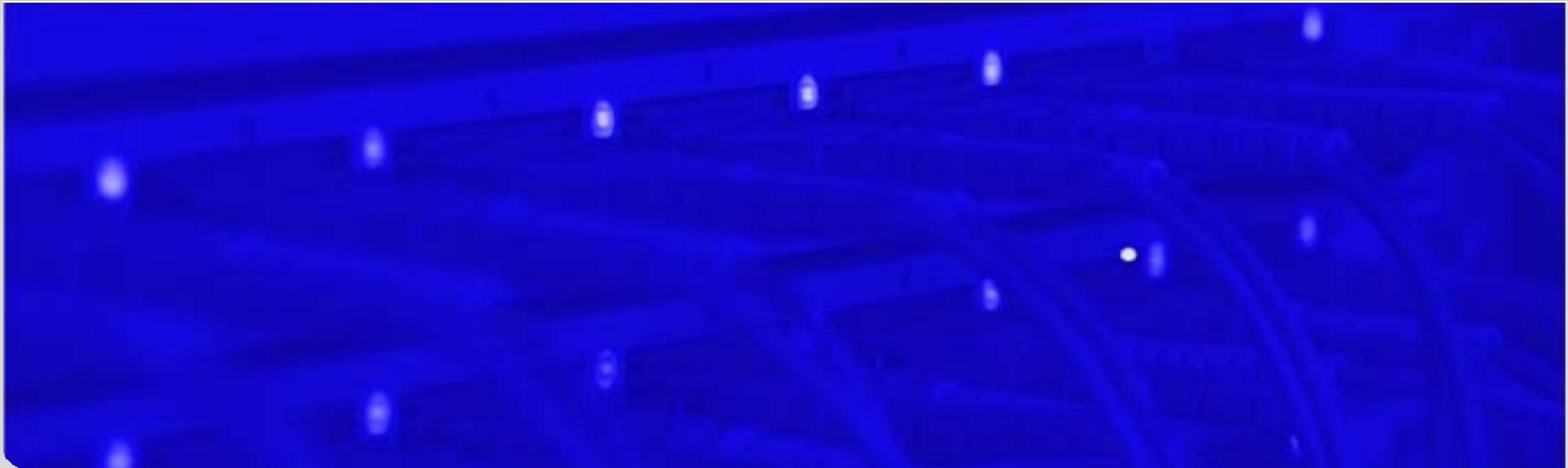


Next Generation Internet

9. Flexible Netze

INSTITUT FÜR TELEMATIK



Überblick Kapitel 9

I. Einführung

1. Einführung

II. Internet-Architektur

2. Internet-Architektur
3. NAT & IPv6
4. Dienstgüte

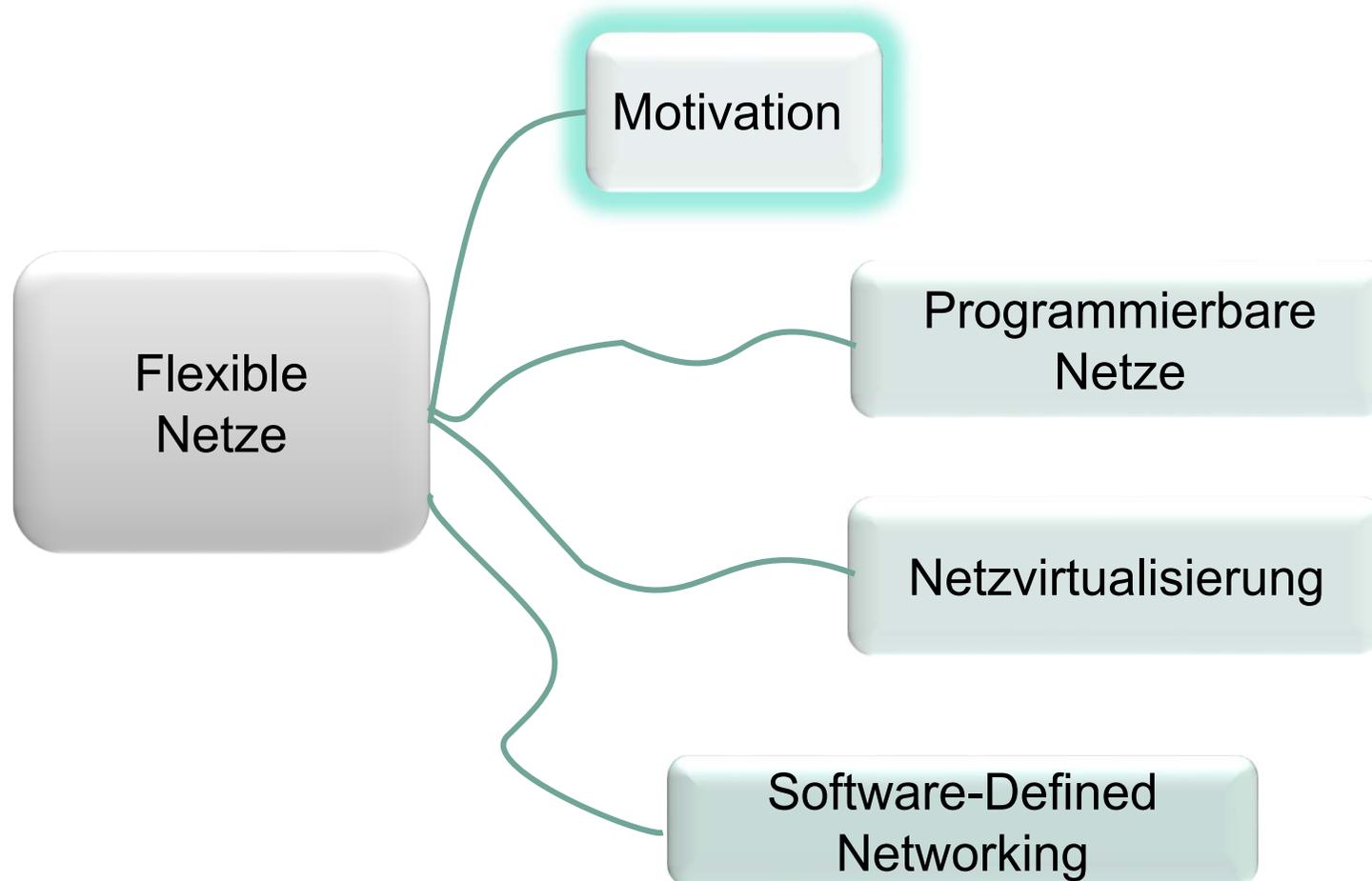
III. Multicast

5. Grundlagen
6. Multicast Routing
7. Multicast Transport

IV. Flexible Dienste und Selbstorganisation

8. Neuere Transportprotokolle
9. Flexible Netze
10. Peer-to-Peer

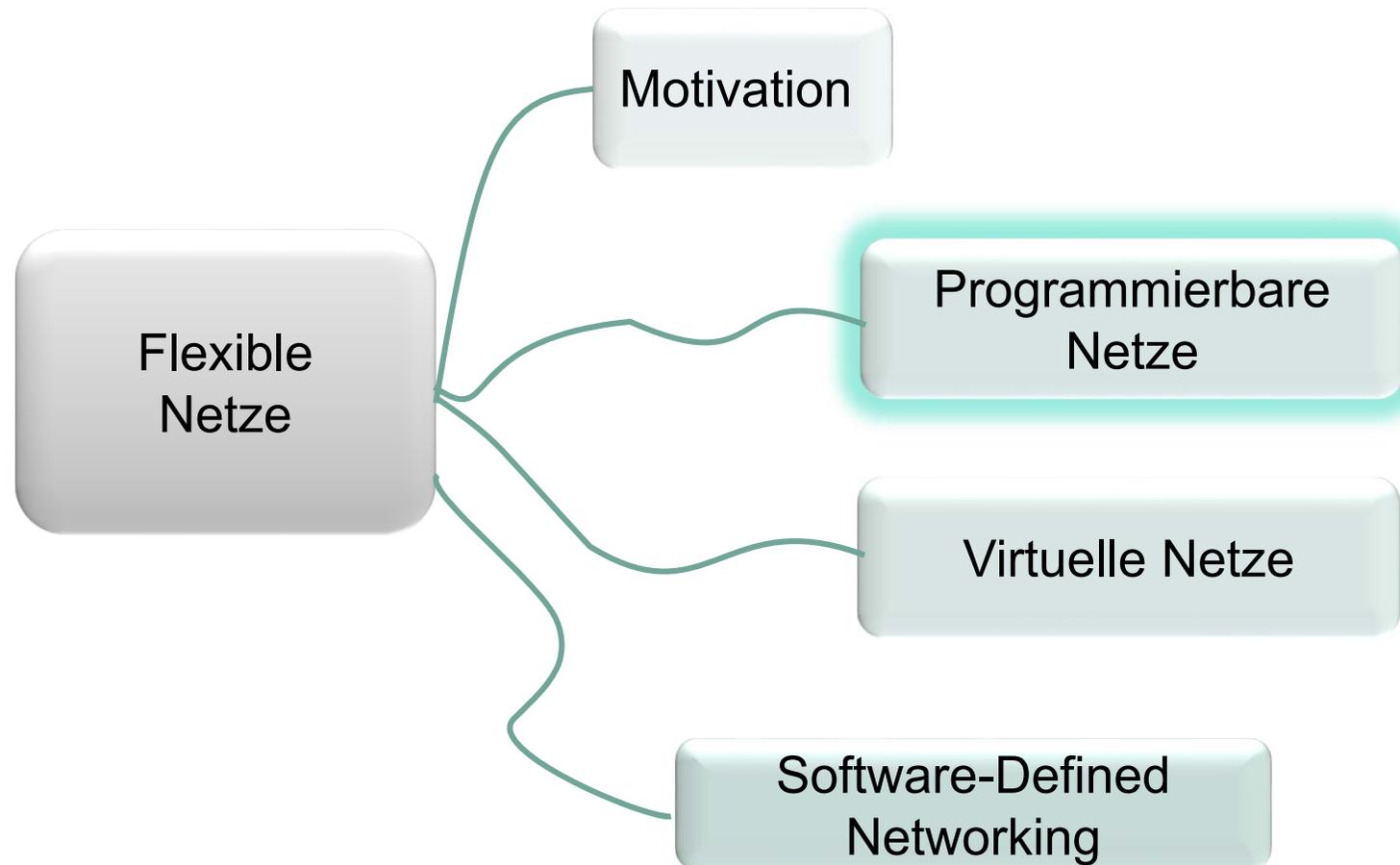
Überblick



Motivation

- Bisherige Paradigmen
 - ein Netz für alles
 - das Internet-Protokoll als Invariante
 - intelligente Endsysteme, „dummes Netz“ (nur Weiterleiten)
- Aber: Innovation im Netz schwierig
 - siehe IPv6, Multicast, Dienstgüte, AQM/ECN, ...
- **Flexibilisierung** der Funktionalität
 - erlaubt oft effizientere Lösungen
 - Überwinden von protokoll- oder implementierungsspezifischen Mängeln
 - im **Datenpfad** → Bsp. **Aktive/Programmierbare Netze**
 - im **Kontrollpfad** → **Software-Defined Networking**
 - des gesamten Netzes → **Netzvirtualisierung**

Überblick

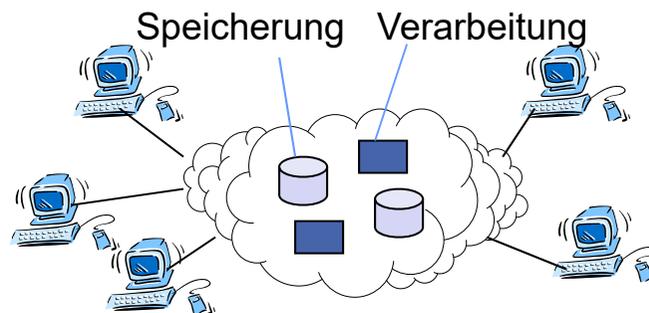


Aktive/Programmierbare Netze

- Große **Heterogenität**:
 - Vielzahl verschiedener Endgeräte
 - Unterschiedliche Anbindung (Übertragungskapazität)
- Mehr **Intelligenz im Netz**
 - nicht nur Weiterleiten, sondern auch **Verarbeiten/Speichern**
 - kann Endsysteme entlasten
- **Beispiele**
 - Aggregation von Messdaten im Rahmen von Sensornetzen „In-Network processing“
 - Transcodierung von Videoströmen

Rapid Service Creation

- Konvergenz von „Computing“ und „Communication“
 - Zusammenwachsen: Verarbeitung, Speicherung und Kommunikation
- Ziel
 - Flexible, schnelle Bereitstellung anwendungsspezifischer Dienste
- Programmierbarkeit
 - Von „einfacher“ Infrastruktur“ zu programmierbarer Infrastruktur mit Speicherung und Verarbeitung



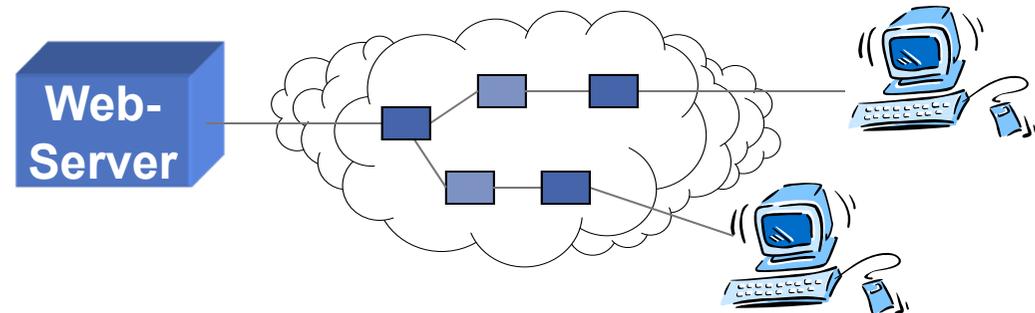
Aktive Netze

- Wie kann man Netze **flexibler** gestalten?
 - Idee: **Netzkomponenten** sind für Anwendungen individuell **programmierbar**
- Trennung von ISP und ASP
 - Infrastructure Service Provider (ISP)
 - Application Service Provider (ASP)
- **Schnelle Einführung neuer Dienste**
- Standardisiert werden müssen:
 - Schnittstellen zwischen den Diensten
- Widerspruch zum Ende-zu-Ende-Argument?
 - Eigentlich schon, denn anwendungsspezifische Funktionen sollten aus dem Netz fernbleiben
→ aber: mehr Flexibilität bei Platzierung
anwendungsnaher Funktionen im Netz

Beispielanwendungen

■ Web-Proxies

- Speicherung von Webseiten



■ Audio- und Video-Streaming

- Pufferung und „Shaping“ des Datenstroms

■ Netzmanagement

- Basiert meist auf zentralen Architekturen

■ Angriffserkennung

- Verkehrsanalyse im Netz zur frühzeitigen Erkennung von Angriffen

Individuelle Dienstgüte mit Videofiltern

- Videofilter, z.B. für MPE
- Beispiel

Original



Geringere Qualität



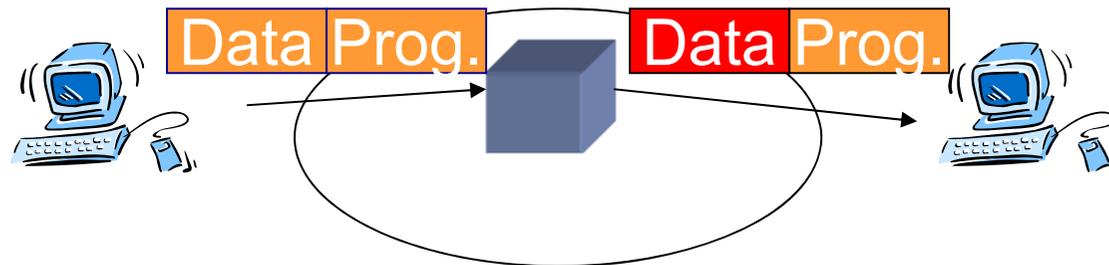
Schwarzweiß



Realisierungskonzepte (1)

■ Capsules

- Dateneinheiten führen das Verarbeitungsprogramm selbst mit



■ Vorteile

- Verarbeitungsprogramm kann sich per Dateneinheit ändern
- Keine Signalisierung/Installationsphase
- Keine Zustandshaltung

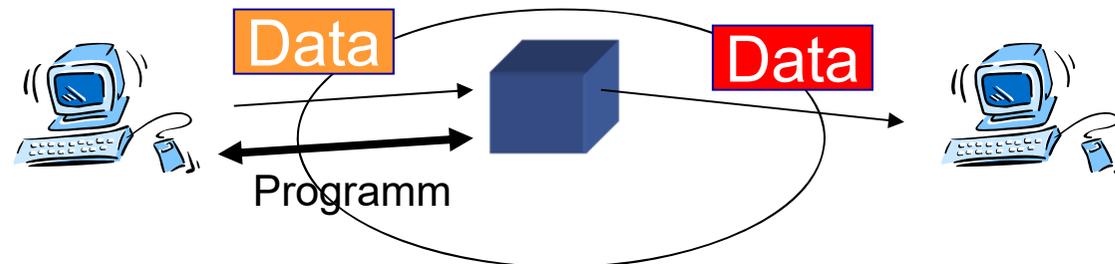
■ Nachteile

- Gefährdung der Sicherheit
- limitierte Programmgröße
- Kein Kontext auf dem Knoten nutzbar
- Hoher Mehraufwand während Weiterleitung

Realisierungskonzepte (2)

■ Programmierbare Knoten

- Programme werden per Signalisierung auf den Knoten geladen und werden anschließend auf Paketen ausgeführt



■ Vorteile

- Geringerer Mehraufwand während Weiterleitung
- Sicherheit „einfacher“ zu gewährleisten
- Kontext auf dem Knoten nutzbar

■ Nachteile

- Größere Infrastruktur erforderlich
- Zustand muss für jeden Flow verwaltet werden
- Hohe Reaktionszeit

ANTS



- Eines der ersten Projekte im Bereich der aktiven Netze

- Ursprünglicher Ansatz
 - Programm und Daten werden in Capsules verschickt
 - Aktiver Knoten stellt Basisfunktionalität zur Verfügung
 - Knoten überwacht Ressourcen- und Bandbreitenanforderungen eines Capsules

- Überarbeiteter Ansatz
 - Code-Loading on demand (vom vorherigen Knoten)

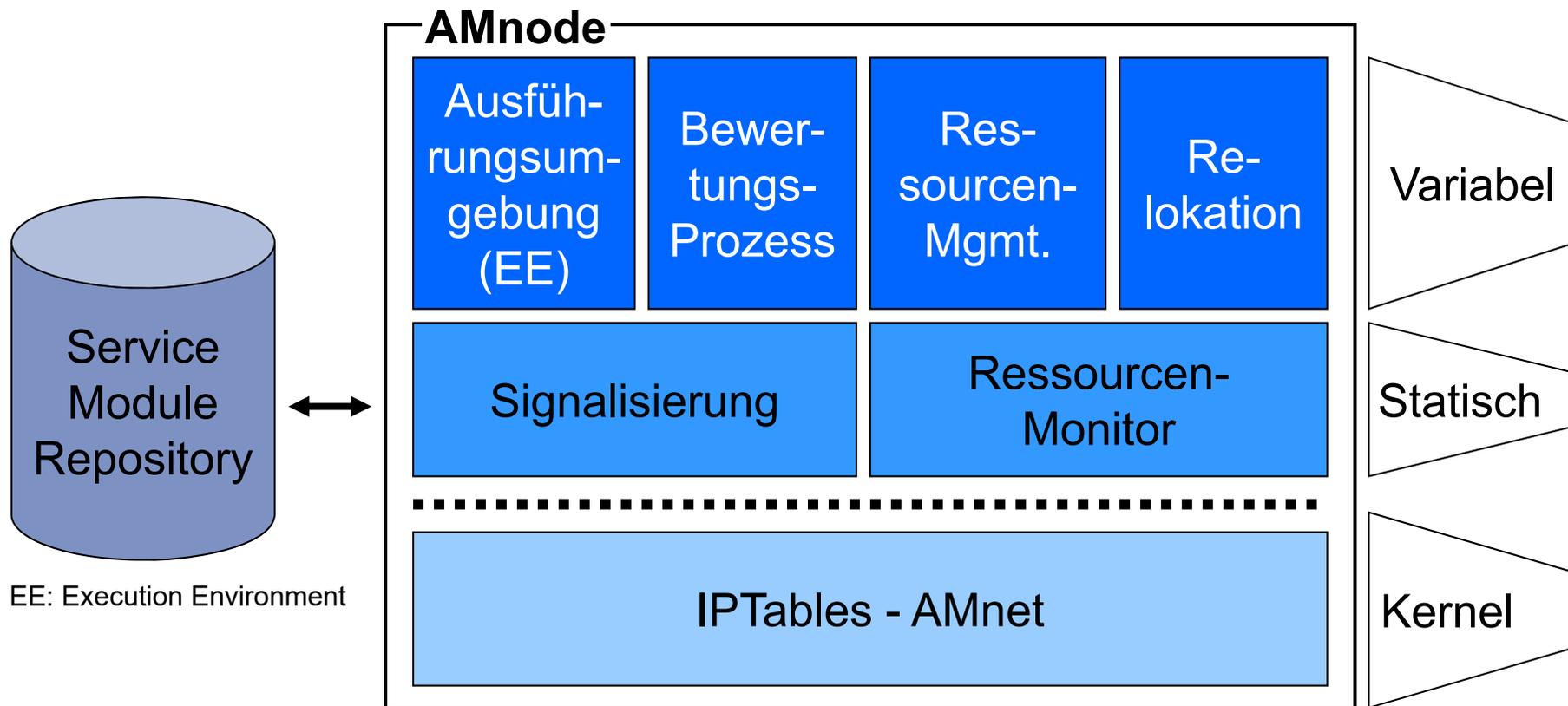
AMnet: Active Multicasting Network

- Rahmenwerk für programmierbare offene Dienstplattform
 - Basiert auf programmierbaren Knoten
 - Empfängerbasiertes Konzept, nutzt Multicast-Techniken
- Integration von Software und programmierbarer Hardware (FPGA)
 - Softwaremodule in C oder Java
 - Dedizierte Dienstmodule für programmierbare Hardware
- Beispielanwendungen in AMnet
 - Heterogene Gruppenkommunikation
 - Ankopplung mobiler Teilnehmer
 - Flexible Dienstebereitstellung in Ad-hoc-Netzen



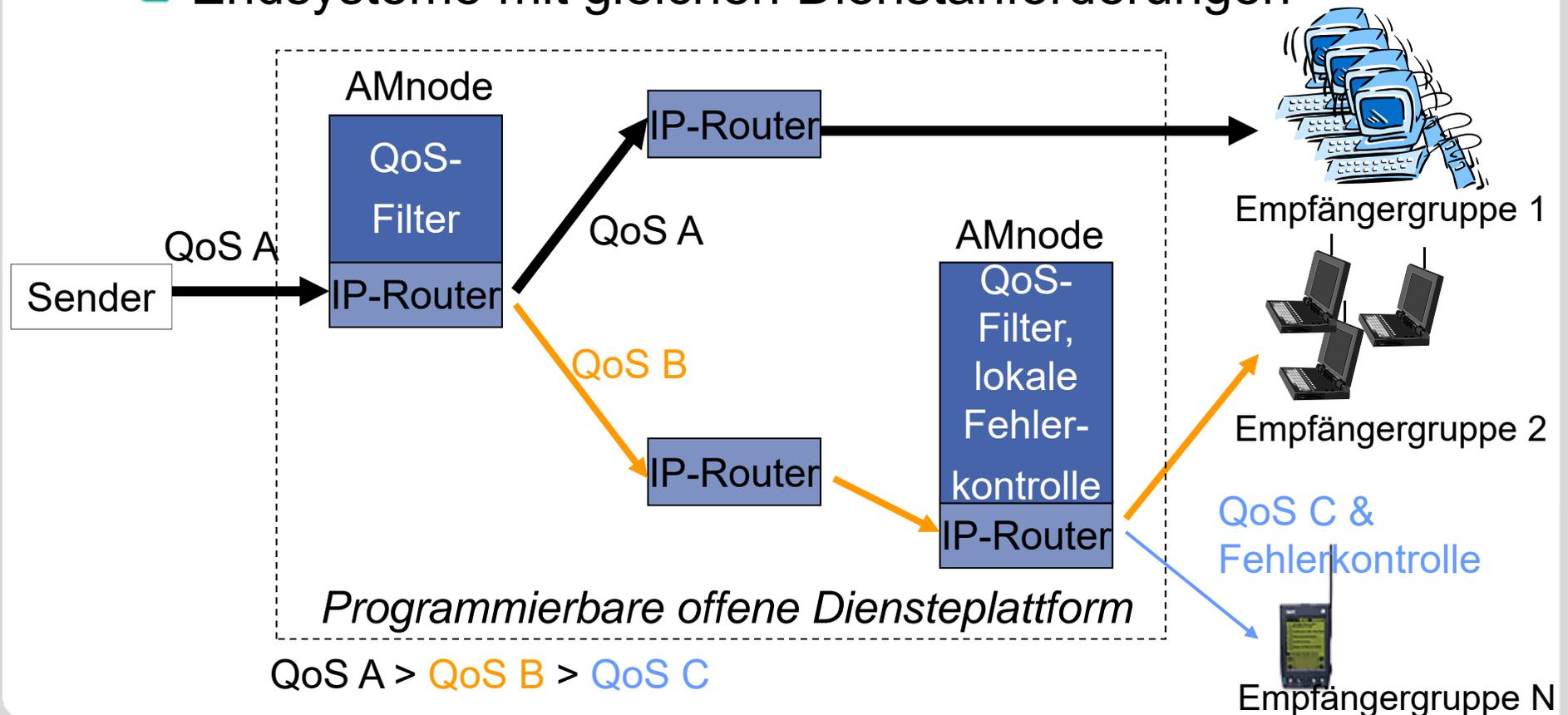
AMnet-Architektur

- **Statische** Komponenten werden einmal bei der Initialisierung des Knotens geladen
- **Variable** Komponenten können zur Laufzeit des Knoten ausgetauscht werden



Anwendungsszenario

- IP-Router: “Normales” Weiterleiten
- Empfängergruppe
 - Endsysteme mit gleichen Dienstanforderungen



Programmierbare Netze im Einsatz

- Richtig programmierbare Netze in der Praxis kaum zu finden, aber

- verschiedene kommerzielle Produkte, die Funktionen im Netz ausführen, u.a. diverse Middleboxes, vor allem auch Sicherheitsfunktionen, z.B. Virens Scanner etc.

- Cisco Application Oriented Networking (AON):

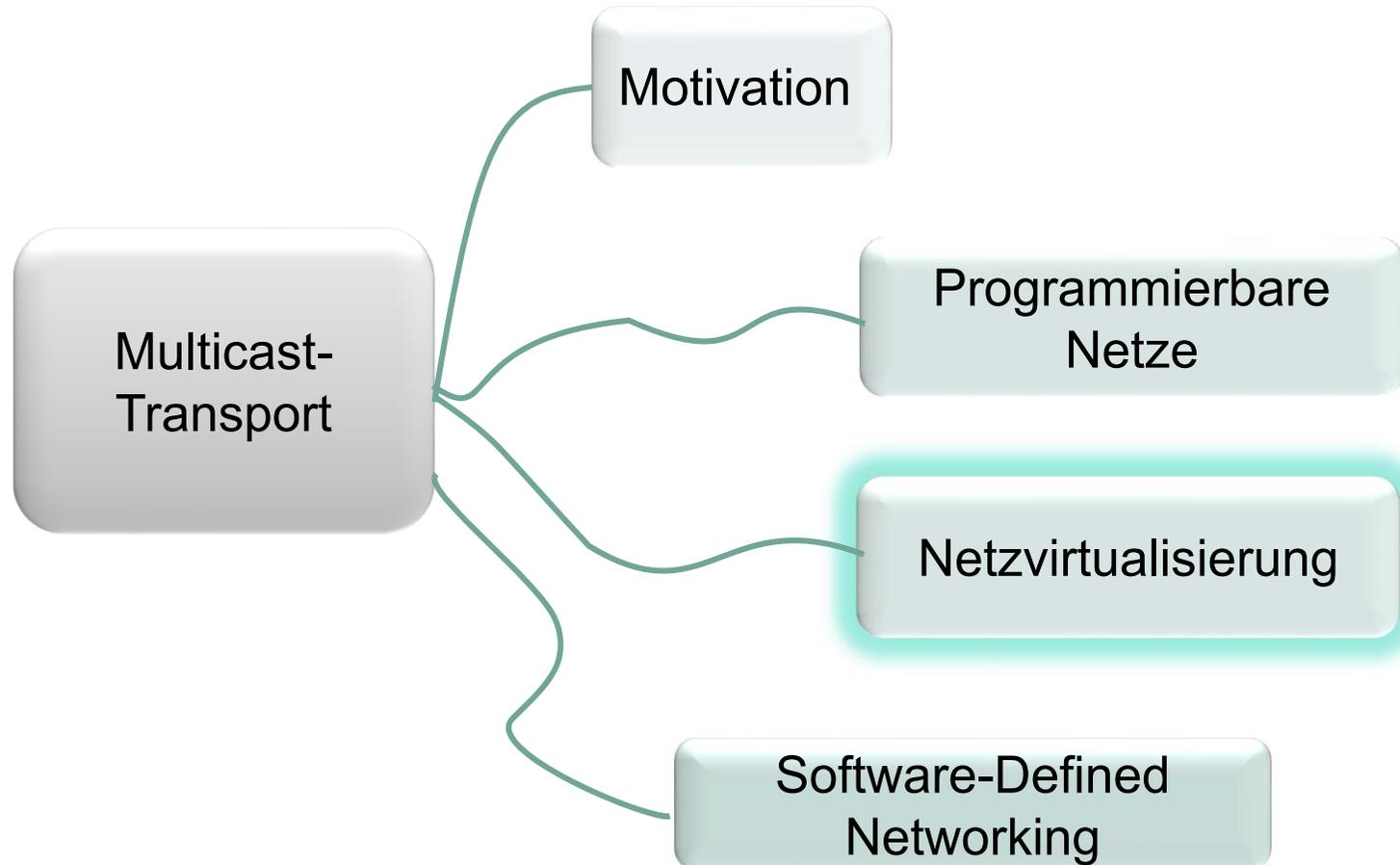
- “...is the foundation for a new class of network-embedded products and solutions that help converge intelligent networks with application infrastructure based on either service oriented or traditional architectures.”



[CISCOAON]

- Nachrichten-basiert → vollständige Paketanalyse

Überblick



Netzvirtualisierung

■ Virtuelles Netz

- virtuelle Knoten und virtuelle Links
- Leblose virtuelle Ressourcen: „Schicht-2-Abstraktion“
- mehrere virtuelle Knoten innerhalb eines physikalischen
- Isolation der virtuellen Netze

■ Vorteile

- Parallele Existenz unterschiedlicher Architekturen
- Netztopologie kann einfacher geändert werden
- bessere Ressourcenauslastung/-nutzung durch Multiplexing und Migration

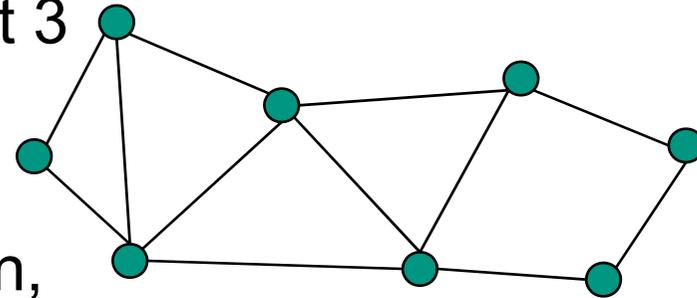
■ Nachteile

- Zusätzliche Indirektionsstufe erhöht Komplexität
- Gestiegener Verwaltungsaufwand durch viele virtuelle Ressourcen

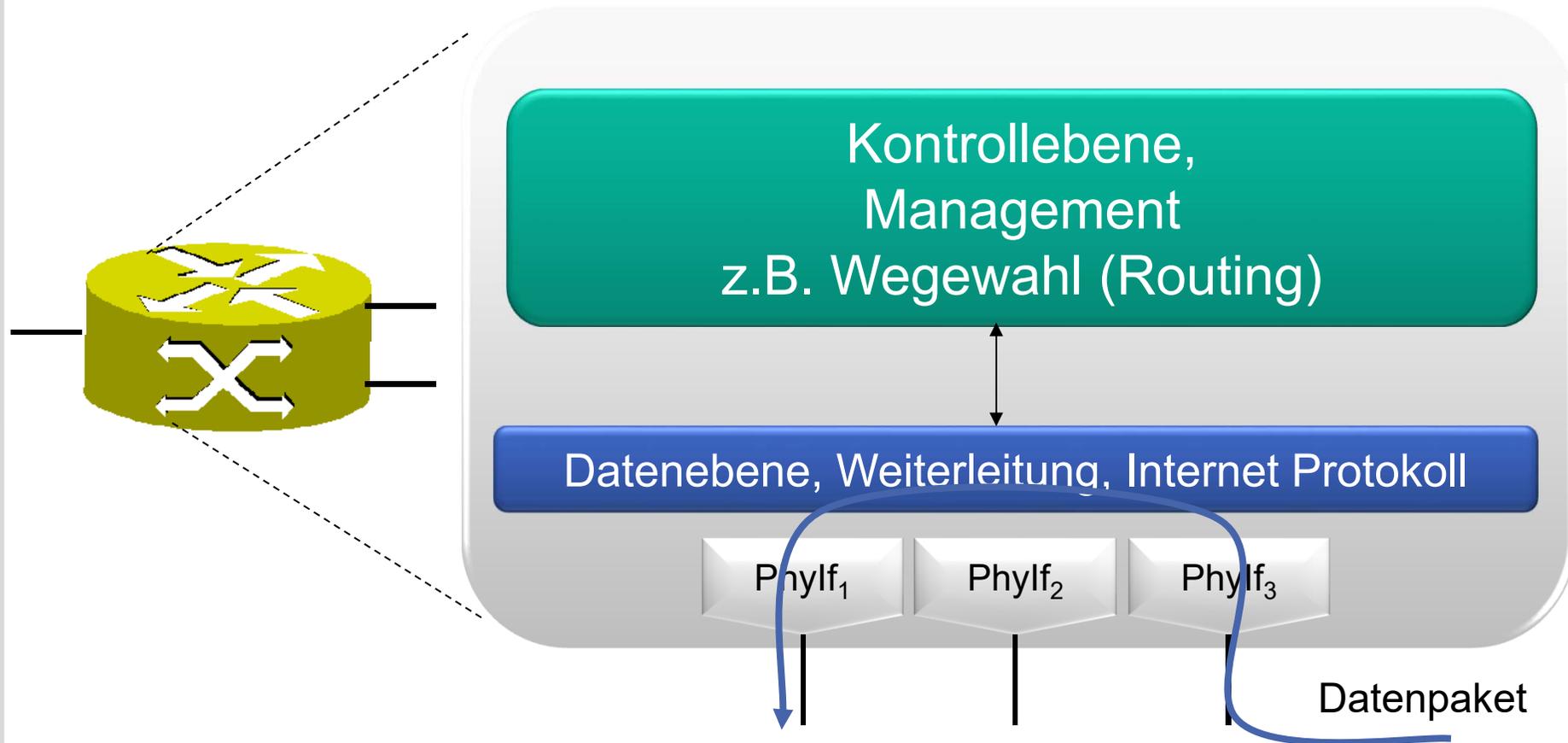
Virtuelle Netze im Kontext der Netzvirtualisierung

■ Virtuelles Netz (VNet)

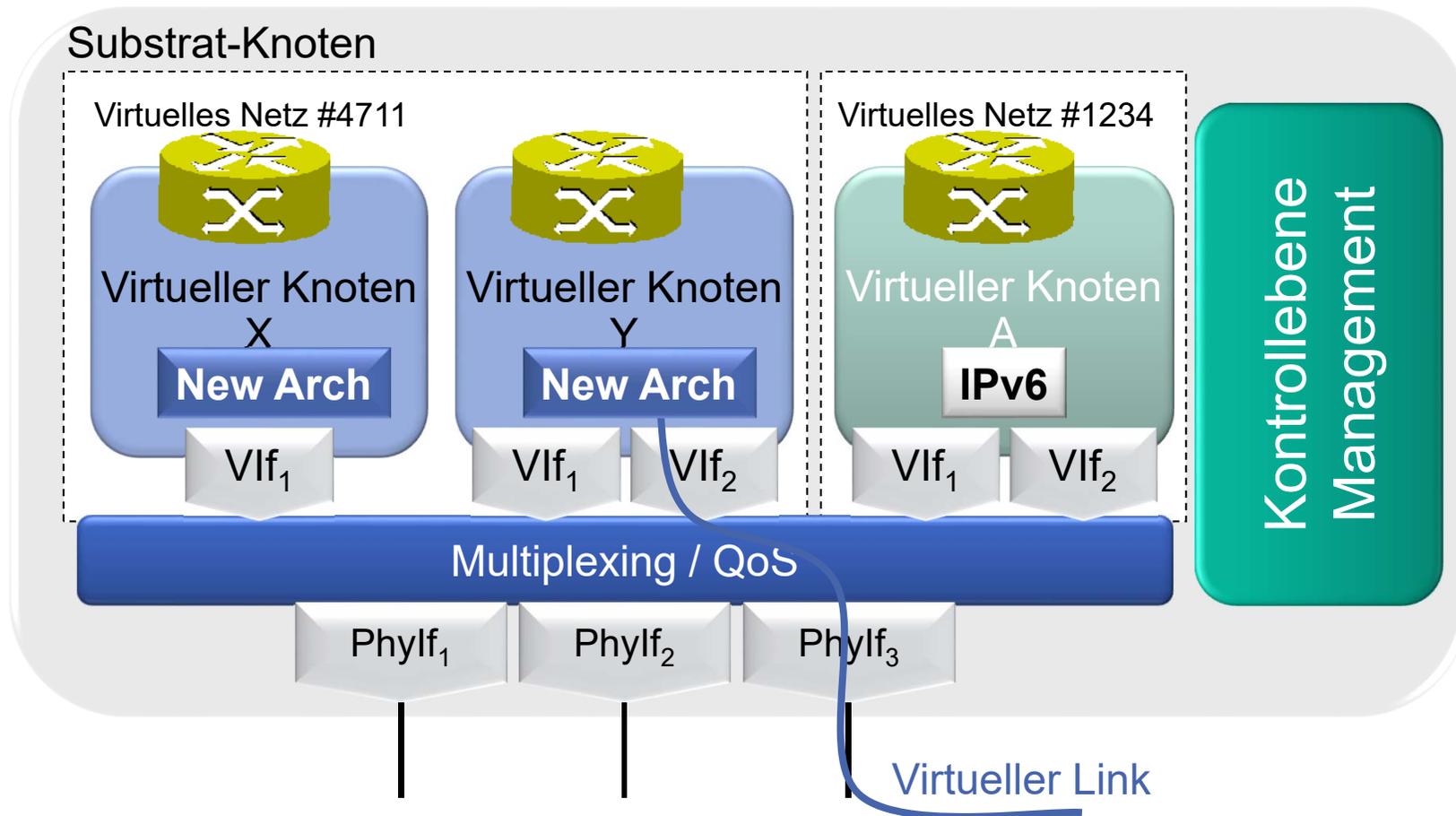
- Menge von (virtuellen) Knoten, direkt verbunden durch (virtuelle) Links (realisiert oberhalb einer Menge von physischen Ressourcen, das “Substrat”)
- „Nackte“ Topologie für Schicht 3
- Keine Annahmen über die Protokolle oder Architektur, die innerhalb des VNets laufen, z.B. nicht notwendigerweise IP
- Kann verschiedene Substrat-Techniken einsetzen, um virtuelle Links zu erzeugen, z.B. IP Tunnels, MPLS, Ethernet VLANs,...
- Geteilte oder aggregierte Ressourcennutzung möglich



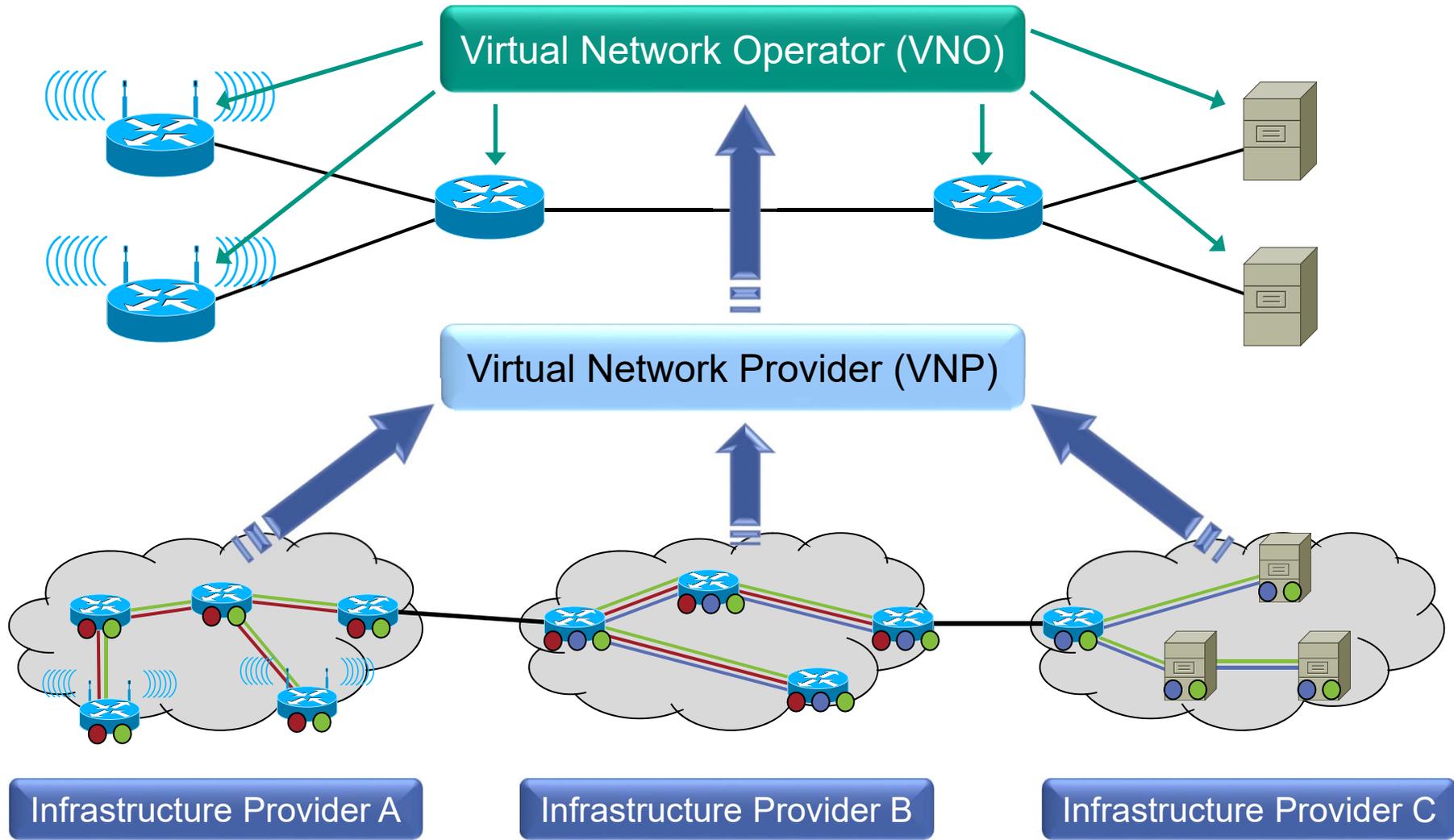
Normaler Router



Netzknoten mit Virtualisierungsunterstützung



4WARD Network Virtualization Business Model



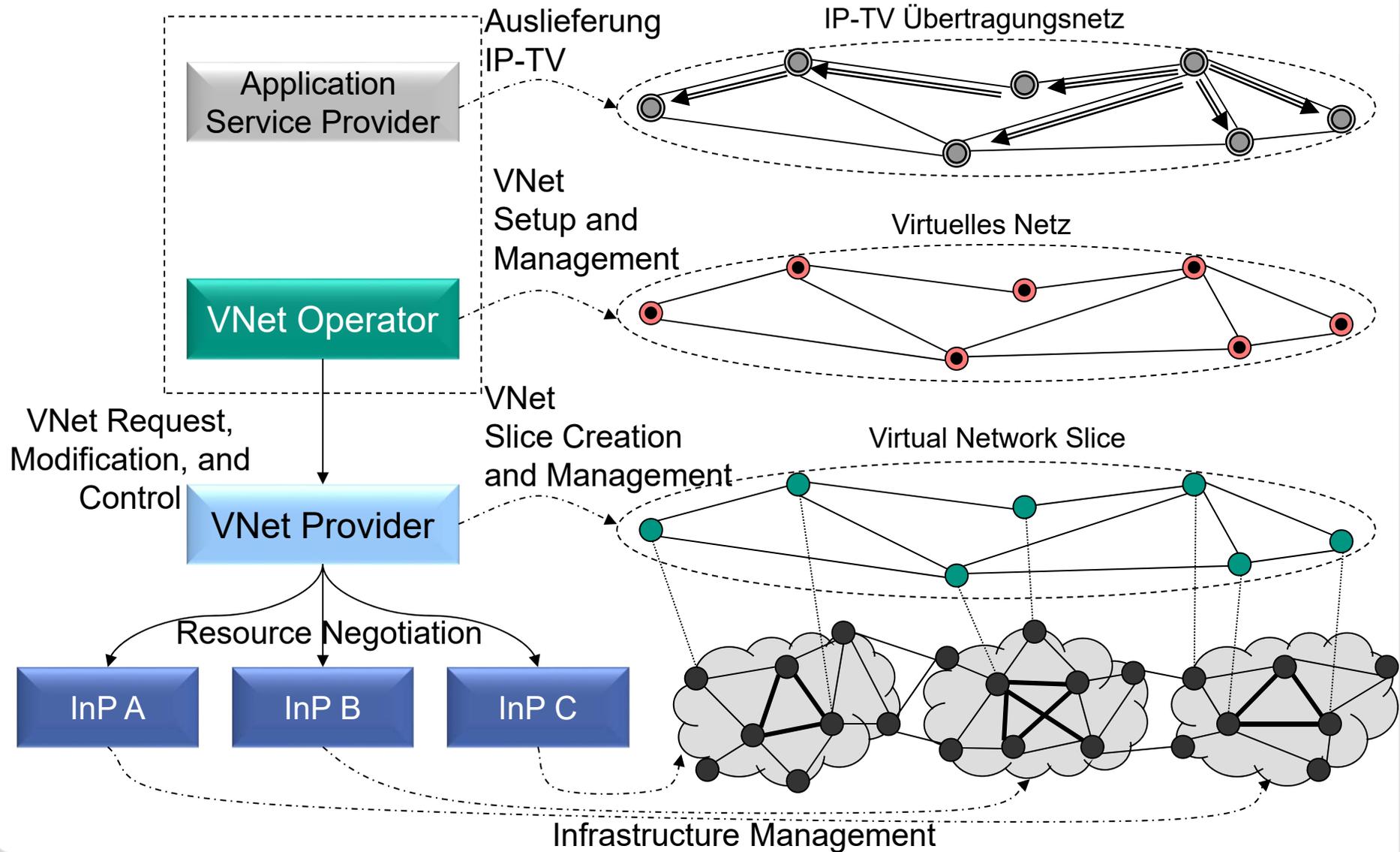
Beispiel-Szenario: Globaler IP-TV-Dienst

■ Ziel:

Ermöglicke Nutzern Zugriff auf HD-TV-Kanäle von überall

- Auf Dienst zugeschnittenes, bedarfsorientiertes virtuelles Netz
- Erzeuge Netz mit weltweiten virtuellen PoPs
- Nutzung von IP Multicast
 - Derzeit nicht verfügbar zwischen ISPs
 - Erforderlich, damit es überhaupt machbar ist (bis zu ~20MBit/s / Kanal)
- Netz kann wachsen und schrumpfen wie erfordert
→ Elastizität

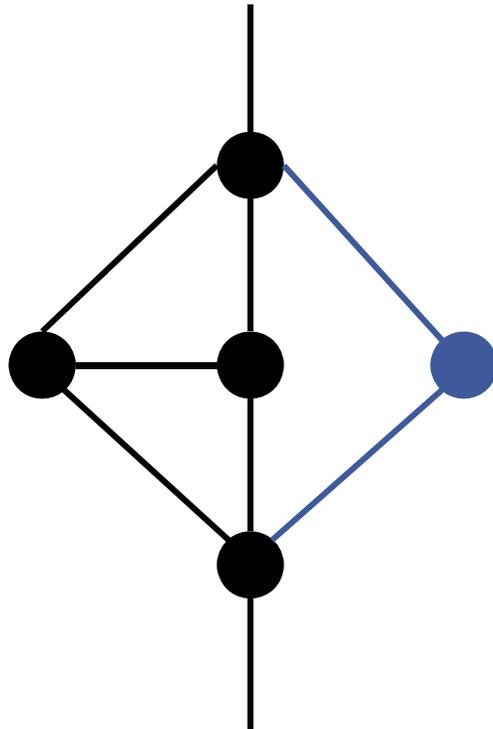
IP-TV Provider Beispiel



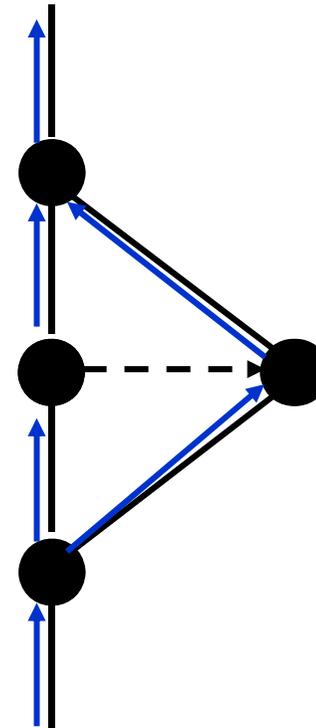
Kontrolle und Signalisierung

- Bedarfsgesteuertes Aufsetzen virtueller Knoten und virtueller Links erfordert Signalisierung zur Laufzeit
- Infrastrukturbetreiber brauchen interoperable Signalisierungslösung zum Aufsetzen virtueller Links über verschiedene Domänen hinweg
- Infrastrukturbetreiber kann virtuelle Knoten innerhalb seiner Infrastruktur verschieben
- Virtual Network Operator benötigt Zugang von außerhalb des VNets um Knoten zu installieren, neu zu starten, anzuhalten, und herunterzufahren
- Signalisierung muss gegen unautorisierten Zugriff auf echte oder virtuelle Ressourcen gesichert werden

Beispiele für Flexibilität durch Virtualisierung

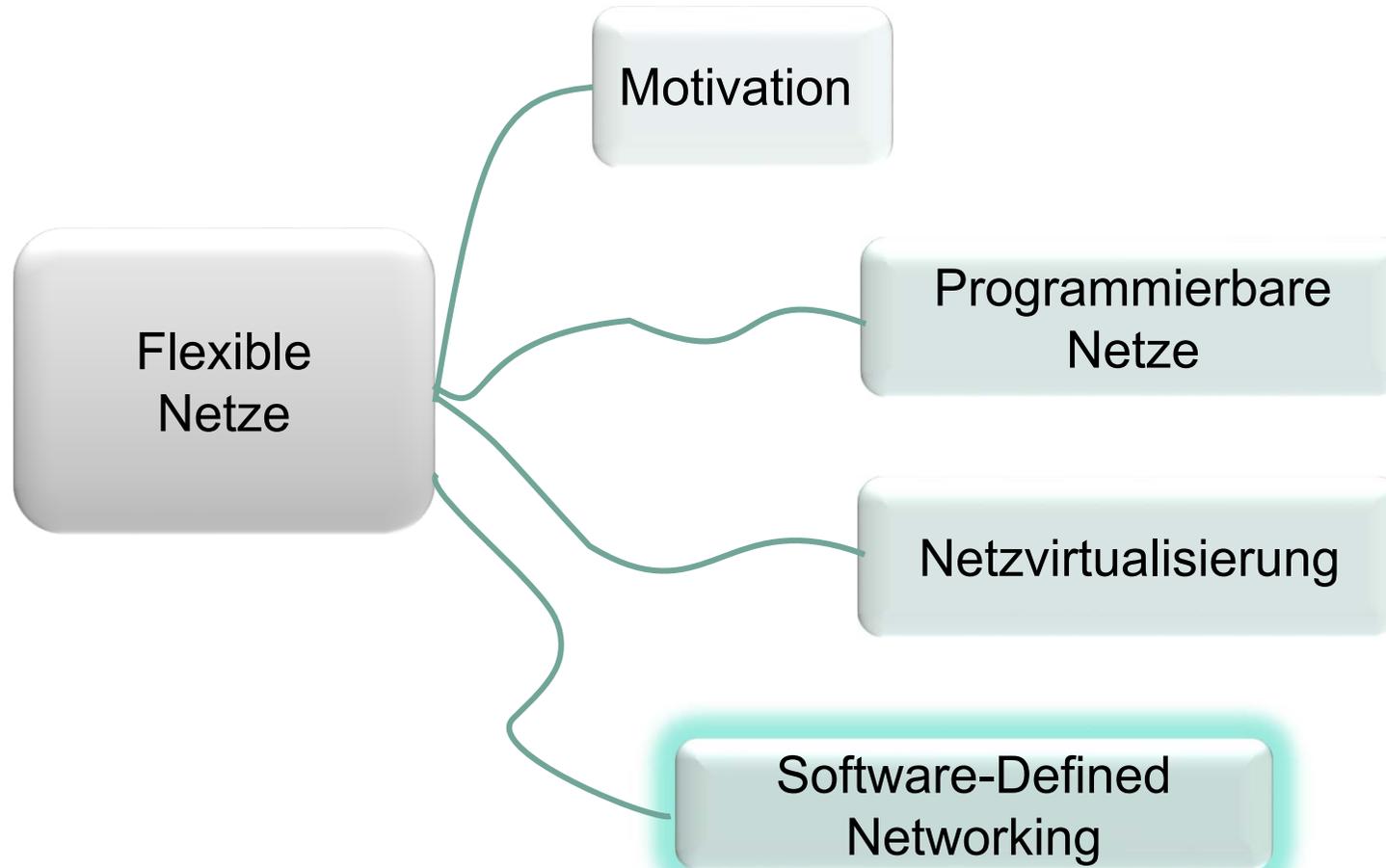


Expansion/Reduktion



Migration

Überblick



Software Defined Networking (SDN)

■ Generelle Zielsetzung

■ Flexibilisierung der Netze

- bezüglich Konfiguration und Funktionalität
- Beschränkungen häufig durch **Kontrollebene** vorgegeben (bestimmtes Routingprotokoll, bestimmte Implementierung des Herstellers)

■ Schnellere Anpassbarkeit

■ Schnelle Ausbringung (Rapid Deployment)

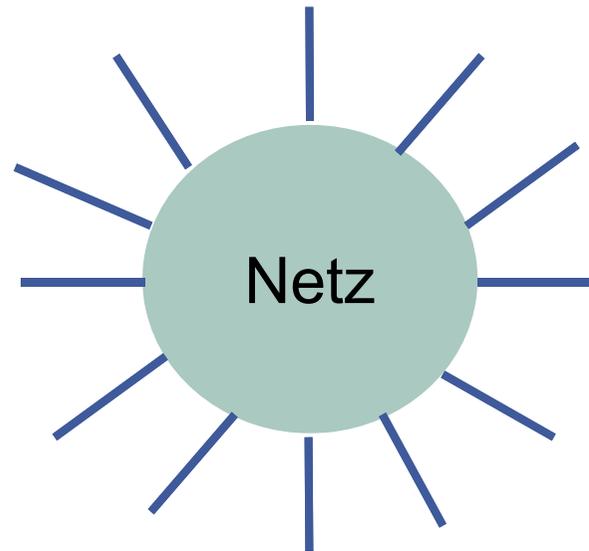
→ schnellere Innovationen

■ Bereitstellung einfacherer Abstraktionen

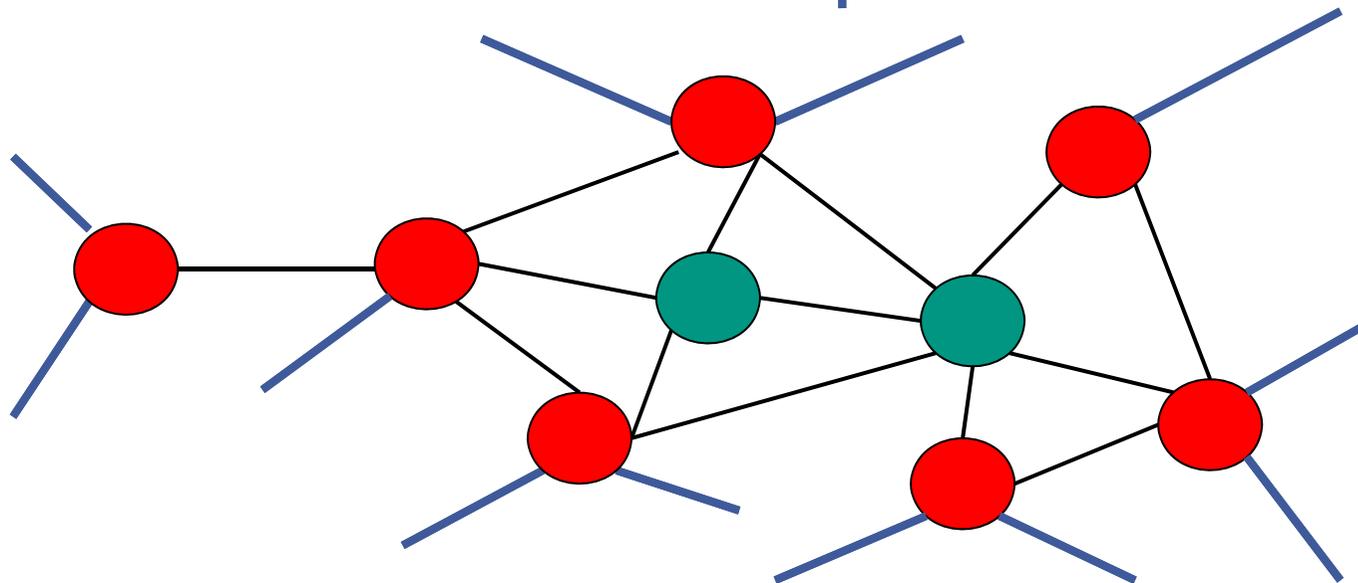
- Verteilte Kontrolle vs. logisch zentralisierter Kontrolle

Beispiel

- Richtlinien für Zugriff zu einem Netz



Abstrakte
Sicht



Globale
Sicht

Beispiel Google

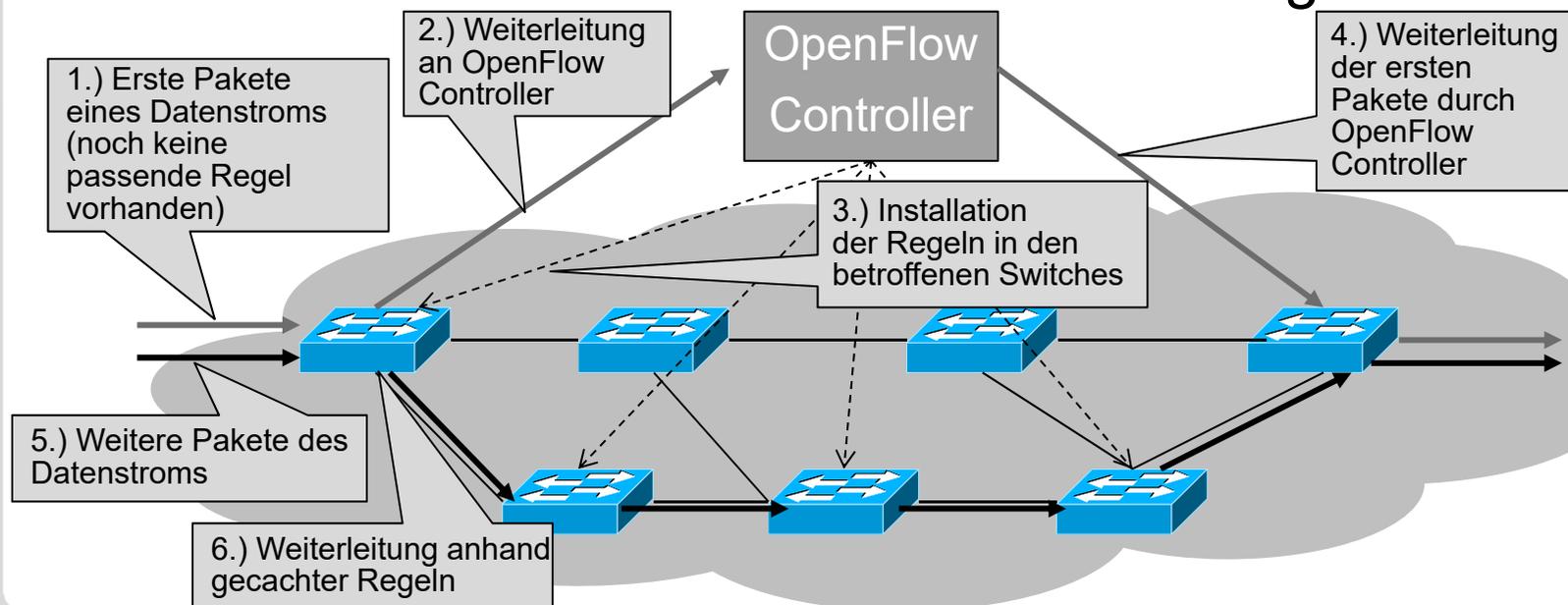
- (Internes) WAN zum Datenabgleich zwischen Datenzentren
 - Massendatentransfer, burst-artig
 - Ziel: bessere Auslastung der Übertragungsstrecken
 - existierende Ansätze nicht robust genug oder zu unflexibel
- Ansatz (OpenFlow-basiert)
 - zentrale Sicht und Steuerung ermöglicht bessere Optimierung (Internet-Sicht: verteilt, einzelne Knoten)
 - deterministisches Verhalten, schnellere Konvergenz auch im Fehlerfall
 - zeitabhängige Schaltung bestimmter Routen
 - schnelle Entwicklung und Testen möglich

OpenFlow

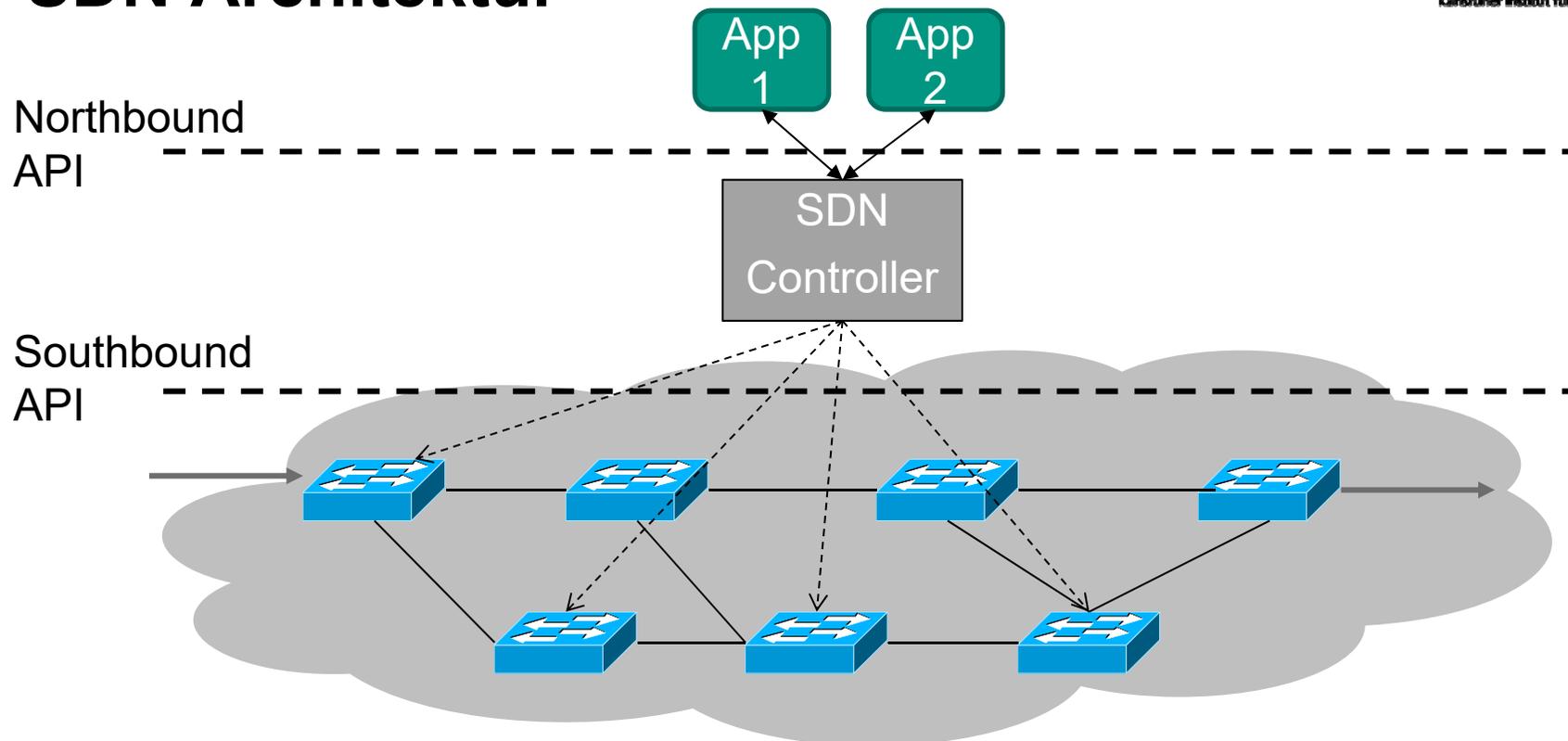
- Trennung **Kontrollebene/Datenebene**
 - OpenFlow: initiiert d. U Stanford, Industriekonsortium
 - Standardisierte Initiative der IETF ForCes WG (seit 2000)
- Aufgaben der Kontrollebene werden durch leistungsfähige Systeme erledigt
- Logisch zentraler Kontrollknoten: OpenFlow-Kontroller
- Datenebene
 - einfache Switches mit TCAMs
 - Anzahl der TCAM-Einträge begrenzt
 - Kontroller weist mit OpenFlow-Protokoll Regeln zu

OpenFlow

- Weiterleitung in Switches anhand einfacher Regeln
 - Regel: Flow-Klassifikator → Aktion
 - Proaktiv vs. reaktiv
 - OpenFlow-Controller konfiguriert Regeln dynamisch pro Datenstrom
- Flexiblere Kontrollebene schafft mehr Möglichkeiten



SDN Architektur



- Skalierbarkeit ggf. eingeschränkt
 - Begrenzte Anzahl der TCAM-Einträge
 - Frequenz der Interaktionen mit Controller
 - Proaktive + reaktive Regeln, ggf. aggregierte Flows

Übungen

- 9.1 Welche Ziele werden mit aktiven Netzen verfolgt?
- 9.2 Nennen Sie die wesentlichen Unterschiede zwischen traditionellen und aktiven Netzen.
- 9.3 Welche Basis-Konzepte gibt es zur Realisierung aktiver Netze?
- 9.4 Was versteht man unter „Relokation“?
- 9.5 Welche Kriterien müssen für eine Dienstausführung beachtet werden?
- 9.6 Nennen Sie wichtige Eigenschaften virtueller Netze im Kontext der Netzvirtualisierung

Literaturhinweise

- [BISa11] R. Bless, S. Perez Sanchez: *Network Design*. In: L. Correia, H. Abramowicz, M. Johnsson, K. Wünnel, Architecture and Design for the Future Internet, Chapter. 4, p. 59–87, Springer, ISBN 978-90-481-9345-5, 2011
- [BIWe09] R. Bless, C. Werle: *Control Plane Issues in the 4WARD Network Virtualization Architecture*, Proceedings of the KiVS Workshop on Overlay and Network Virtualization, Kassel, Germany, März 2009,
<http://journal.ub.tu-berlin.de/eceasst/article/view/225>, Invited Paper.
- [CiscoAON] Cisco AON: A Network Embedded Intelligent Message Routing System,
http://www.cisco.com/application/pdf/en/us/guest/products/ps6438/c1037/cdcco nt_0900aecd802c201b.pdf, <http://www.cisco.com/go/aon>
- [HSWZ01] T. Harbaum, A. Speer, R. Wittmann, M. Zitterbart; Providing Heterogeneous Multicast Services with AMnet; Journal of Communications and Networks, Vol. 3, No. 1, März 2001
- [JaSu13] 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://dx.doi.org/10.1109/MCOM.2013.6658648>

Literaturhinweise

[WeGT98] D. Wetherall, J. Guttag, D. Tennenhouse; ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols; Proceedings of the IEEE Openarch 1998, April 1998

[YeTG13] Yeganeh, S.H.; Tootoonchian, A; Ganjali, Y., "On scalability of software-defined networking," Communications Magazine, IEEE , vol.51, no.2, pp.136,141, February 2013, URL

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6461198&isnumber=6461169>