

An abstract graphic on the left side of the slide, consisting of a network of light blue lines and small circles, resembling a circuit board or a data flow diagram. The lines are vertical and horizontal, with some diagonal connections, and the circles are placed at various points along these lines.

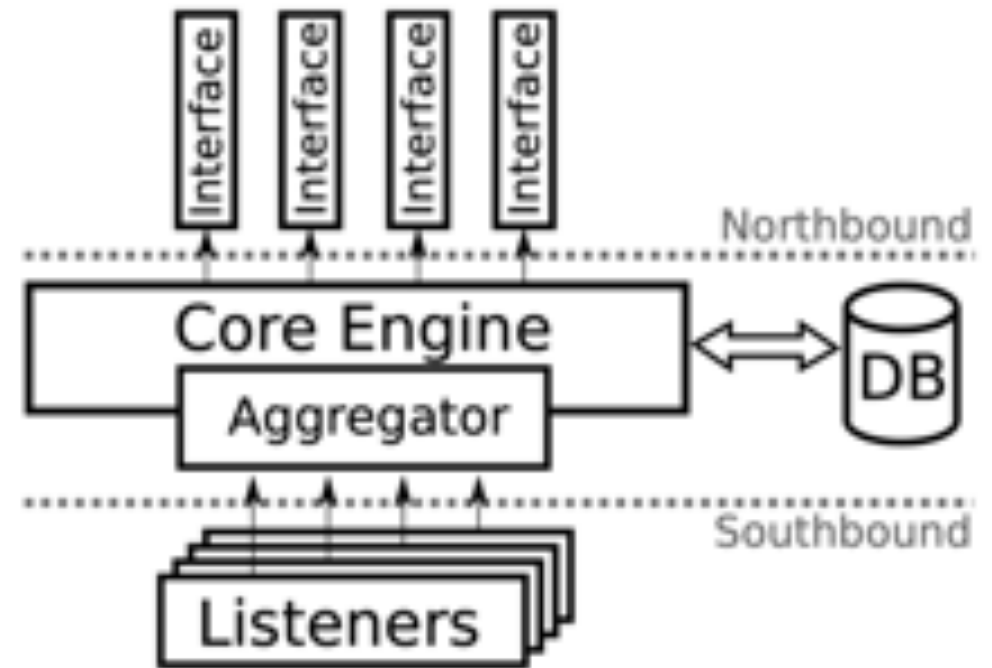
# FlowDirector

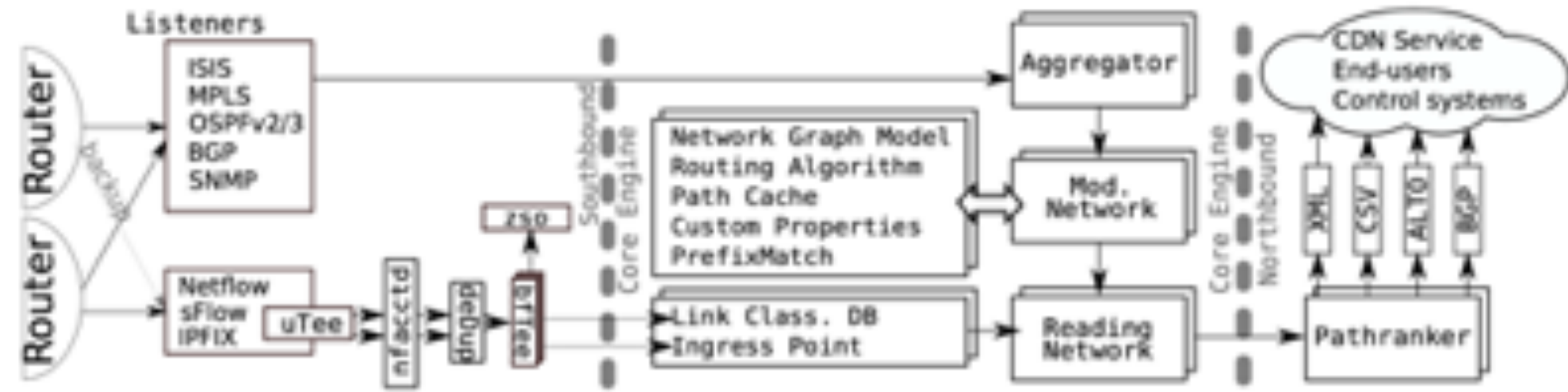
# Herausforderungen

- Sammeln von Netzwerkdaten um Mapping-Empfehlungen an hyper-giants geben zu können
- Dazu notwendig: Kenntnis der ISP-Topologie (alle Routen/Kosten von jeder HG-IP zu jeder Client-IP) → große Datenmenge
- ISP – Service → muss flexibel, skalierbar, gut integrierbar sein
- Soll generic sein: Plattform- und Markenunabhängig

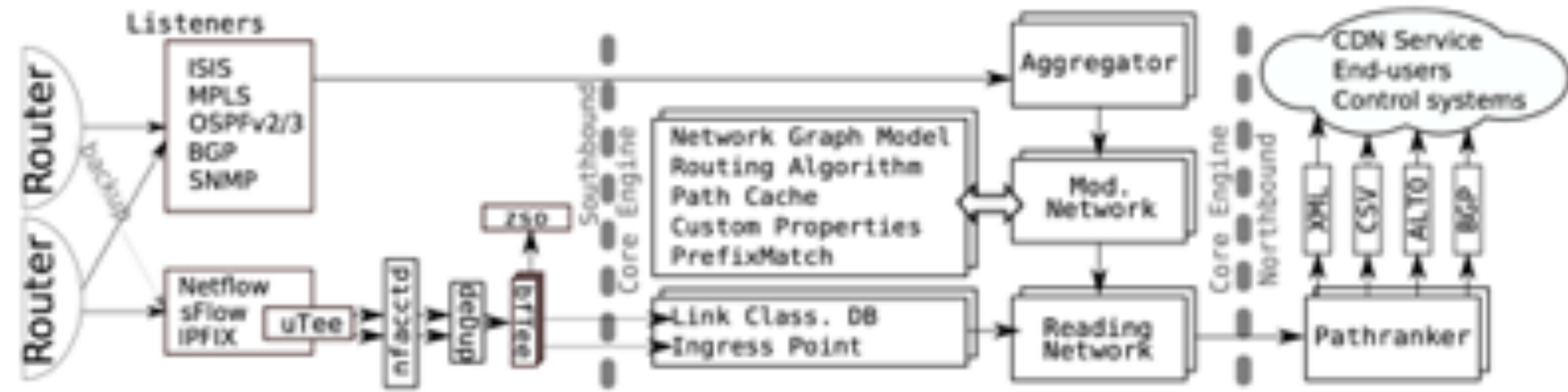
# AUFBAU/FUNKTIONSWEISE

- Modularer Aufbau
- Core Engine sammelt Netzwerkinformationen mit Listeners über Aggregator und speichert diese in ihrer Datenbank (Abbild d. Netzwerkes)
- Bei einem Trigger gibt Core Engine Informationen an Interfaces/Plugins weiter
- Listeners können ohne Anpassung in der Engine ausgetauscht werden (Modular)
- Interfaces geben/verarbeiten Daten der Engine weiter (Plugins, HG, ISP)

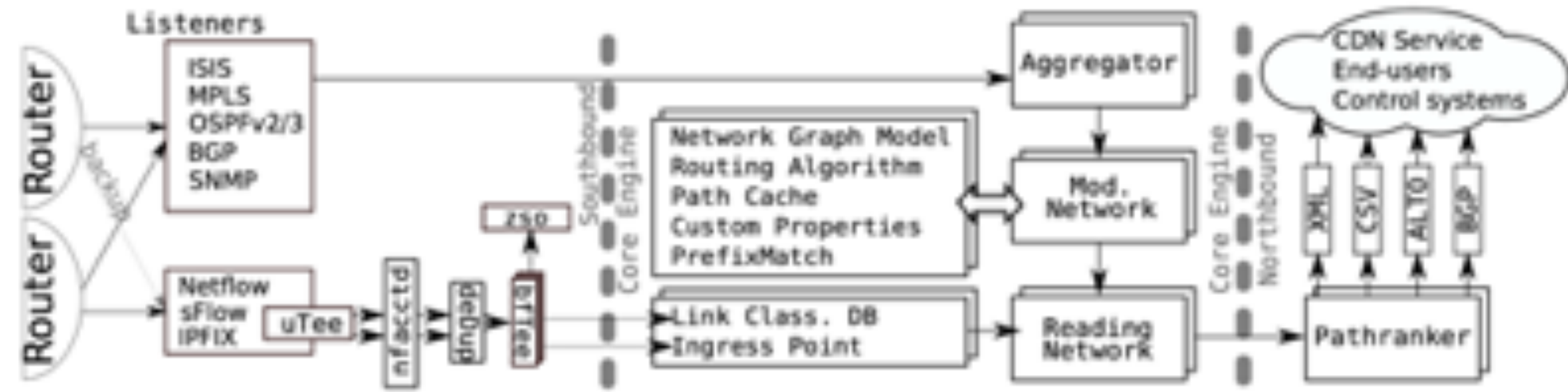




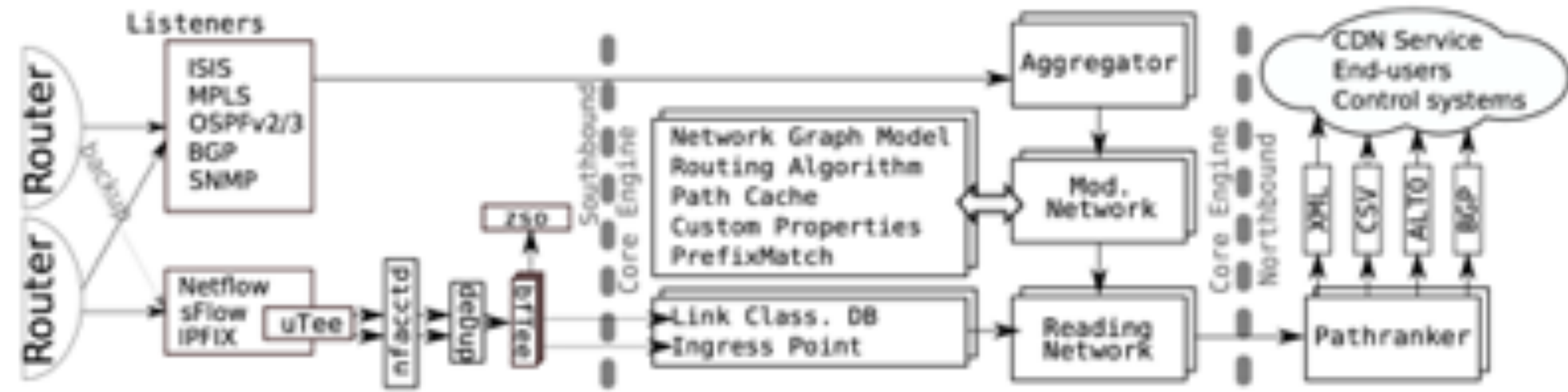
- Blöcke repräsentieren Klassen von Prozessen → Prozesse können verteilt ausgeführt werden
- Listeners: je ein Listener pro Protokoll, eigene Code-Base, kommunizieren nur mit dem Aggregator
  - FD BGP-Listener: empfängt gesamte FIB jedes einzelnen Routers → hoher Speicherbedarf (im Bereich:  $10^3$  GB RAM)
  - Traffic Sampling Listeners: Rohdaten müssen für Core Engine aufbereitet werden (Deduplizierung, Formatierung, Sortierung) → Toolchain



- Core Engine:
  - Link Classification DB: Enthält alle Links, klassifiziert in den Rollen: inter-AS, subscriber, backbone-transport-link
  - Ingress Point Detection: Bestimmung der Eintritts-Router-ID, zur Reduzierung des Speicherbedarfs werden die Flows mit einem Prefix versehen



- Core Engine:
  - Network Graph repräsentiert das Netzwerkmodell (Routen mit Kosten)
  - Custom Properties sind zusätzliche Informationen (vertragliche, cluster Kapazitäten, content Verfügbarkeit)
  - Routing Algorithmus berechnet Routen und speichert diese zusammen mit Custom Properties im Path Cache
  - Trennung von Modification und Reading Network → Aggregator Triggert Änderung des Reading Network



- Northbound Interface:
  - Pathranker: berechnet eine optimale Mapping-Empfehlung von jedem Eintrittspunkt
    - Ist für jeden HG unterschiedlich und hängt vom hop-count, Entfernung und den Custom Properties ab
    - Kann seine Empfehlung über verschiedene Interfaces weitergeben → Schnittstelle zum HG

## VOR-/NACHTEILE

- Mapping-Empfehlungen in Echtzeit
- Reduzierung des Long-Haul Traffics um 30%
- Empfehlungen nicht bindend
- Hohe Anforderungen an Ressourcen (Speicherbedarf)



## QUELLEN

- Enric Pujol, Ingmar Poesse, Johannes Zerwas, Georgios Smaragdakis, Anja Feldmann (2019) Steering hyper-giants' traffic at scale