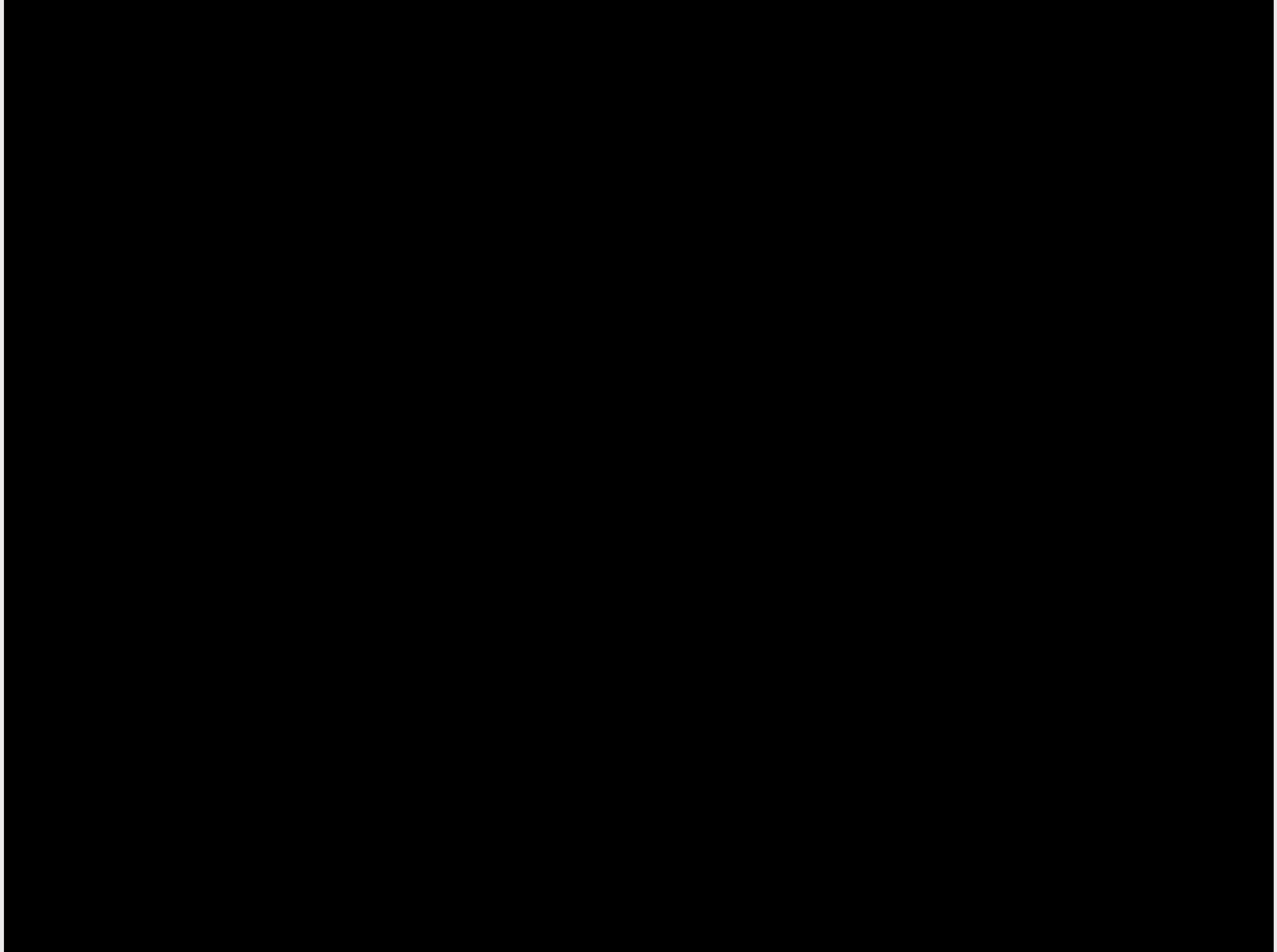
facebook connectivity | magma

Introduction to Magma

Dec 10, 2019



The problem of bringing the next billion onto a faster internet is a problem of heterogeniety

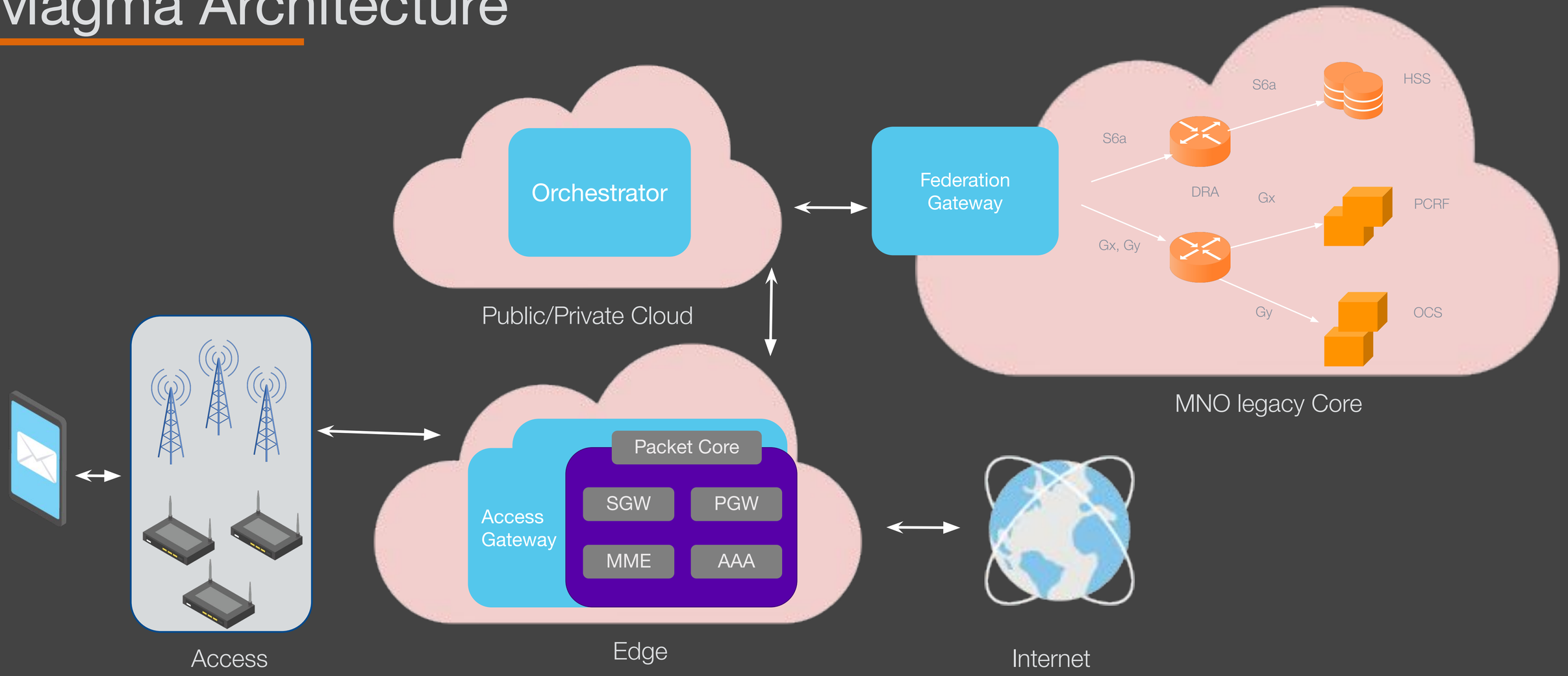
Heterogeneity in access, backhaul, scale and business models

Design Principles

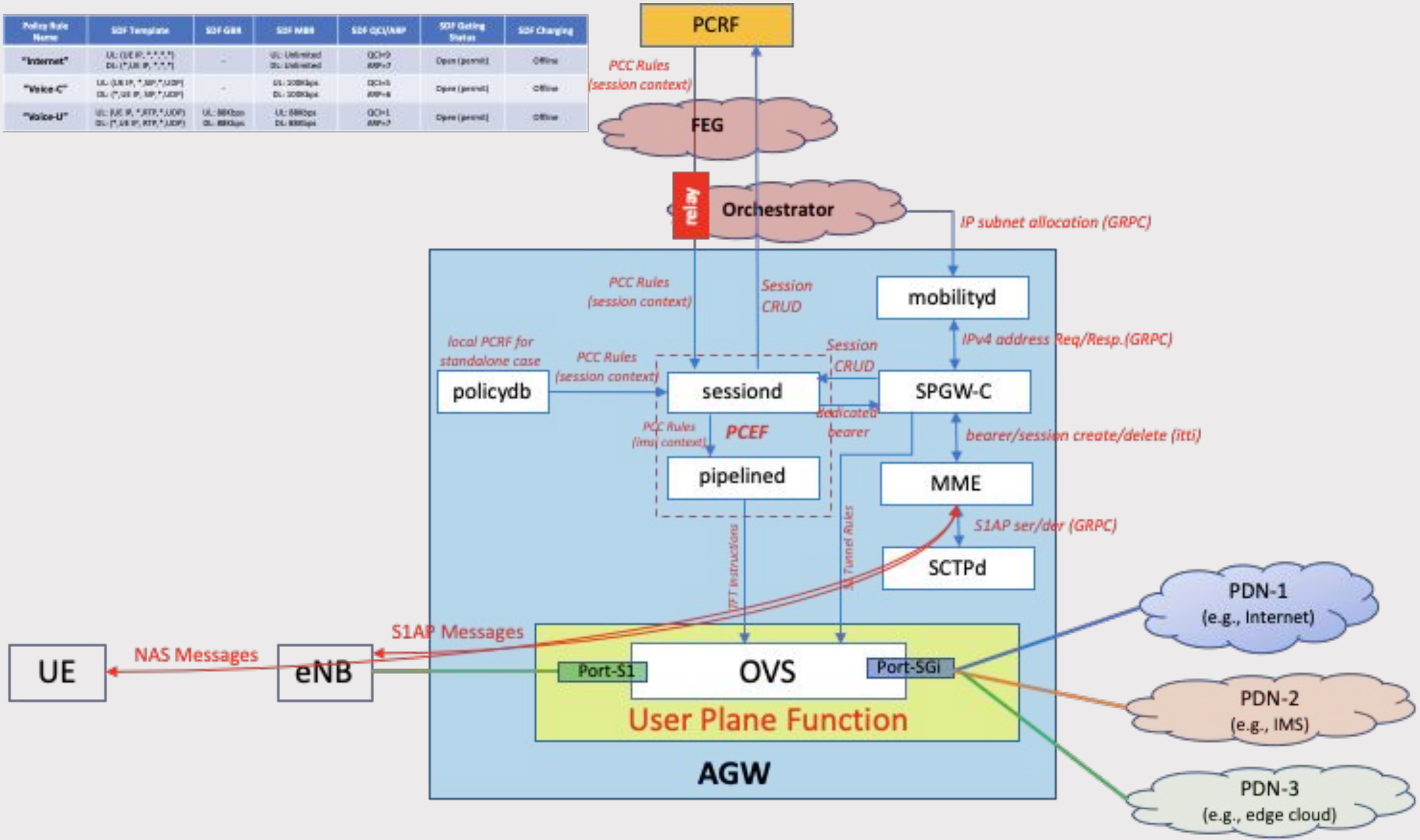
Why Magma? Why Facebook?

1. Encapsulation of state and the fabric
2. State in control planes
3. Software release and fault domains

Magma Architecture

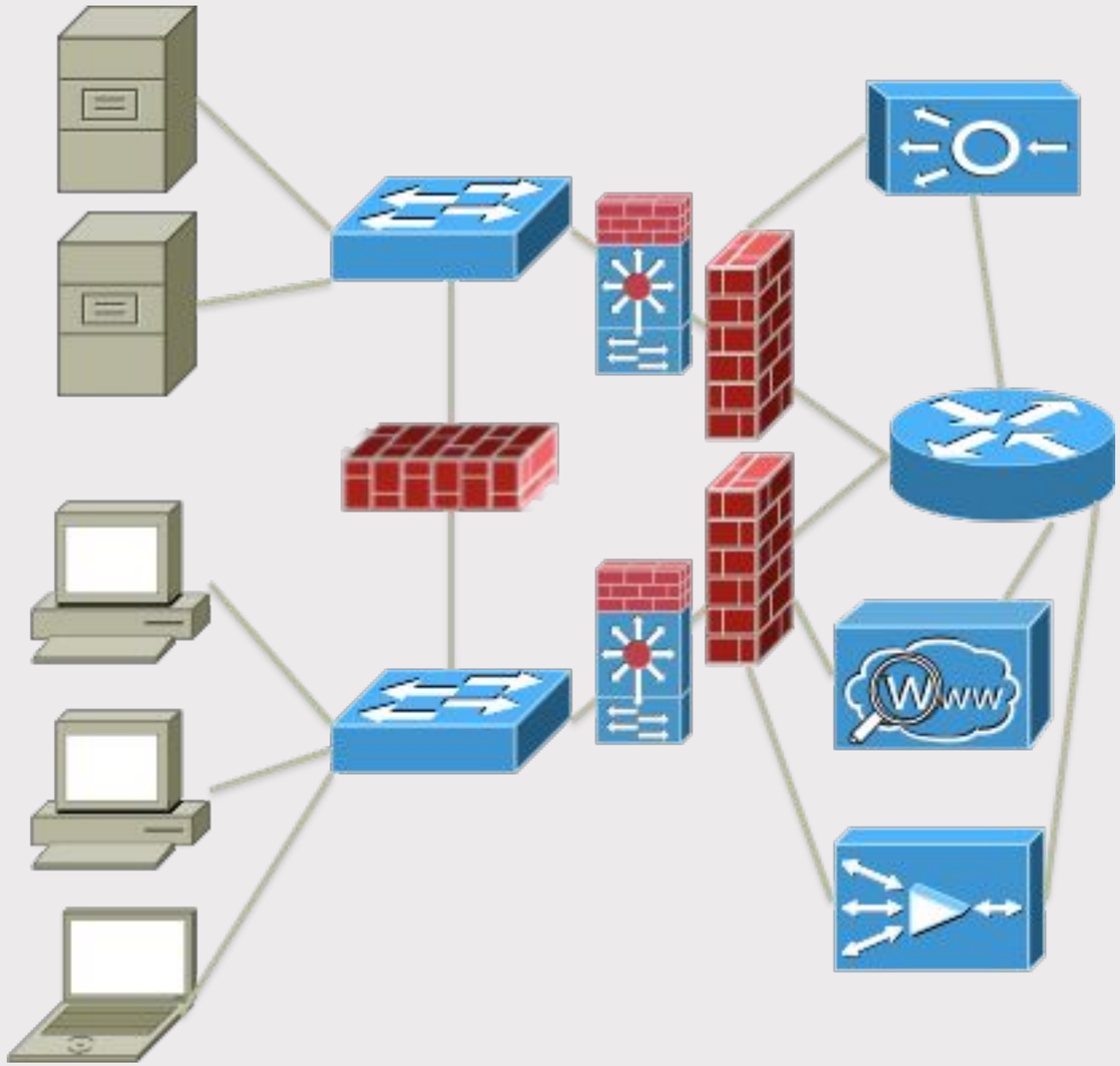


Policy Rule Name	SDF Template	SDF GBR	SDF MBR	SDF QCI/ARP	SDF Gating Status	SDF Charging
"Internet"	UL: (UE IP, *, *, *, *) DL: (*,UE IP, *, *, *)	-	UL: Unlimited DL: Unlimited	QCI=9 ARP=7	Open (permit)	Offline
"Voice-C"	UL: (UE IP, *, SIP, *, UDP) DL: (*, UE IP, SIP, *, UDP)	-	UL: 100Kbps DL: 100Kbps	QCI=5 ARP=8	Open (permit)	Offline
"Voice-U"	UL: (UE IP, *, RTP, *, UDP) DL: (*, UE IP, RTP, *, UDP)	UL: 88Kbps DL: 88Kbps	UL: 88Kbps DL: 88Kbps	QCI=1 ARP=7	Open (permit)	Offline



Encapsulation and the Fabric

Traditional datacenters: State throughout the network

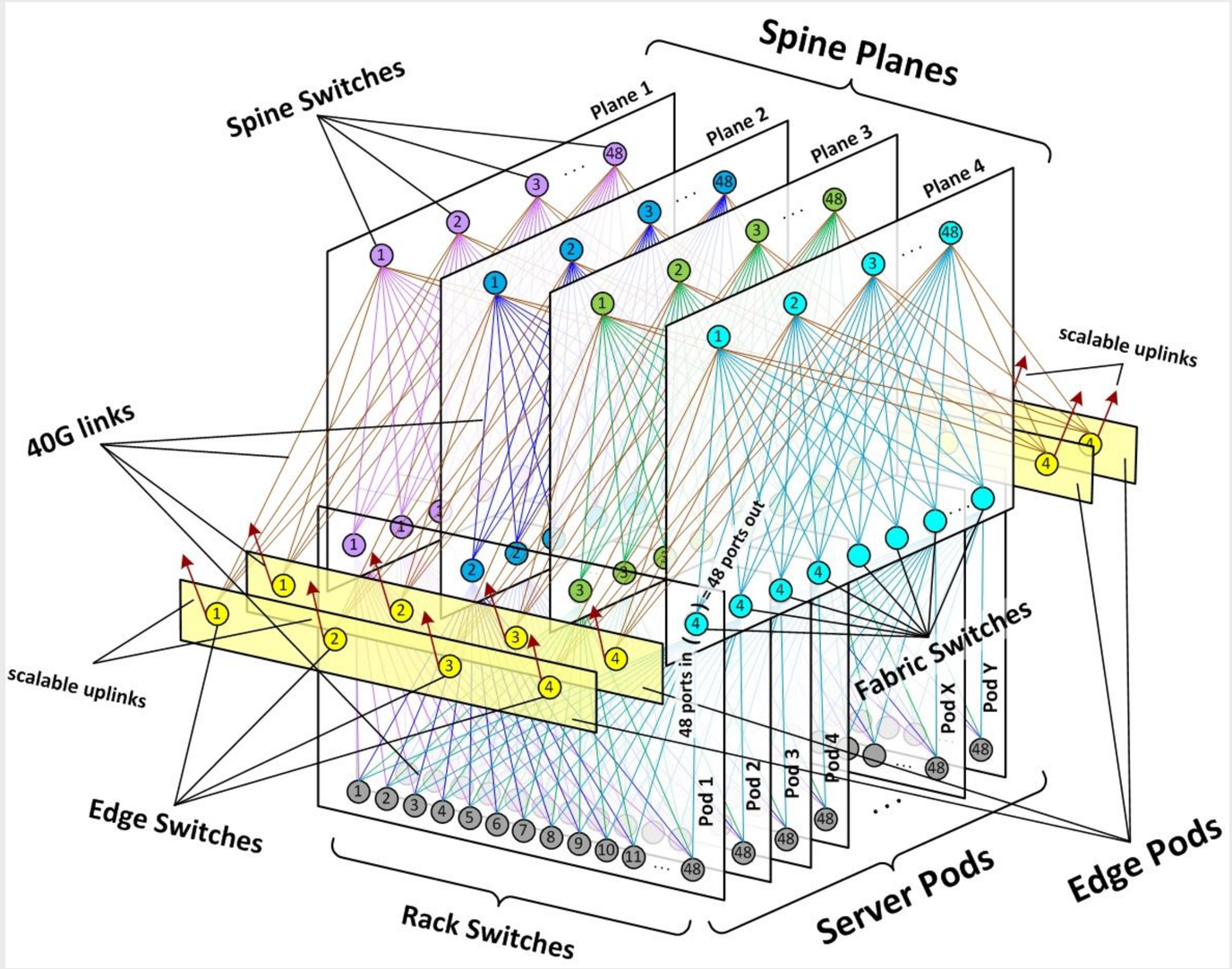


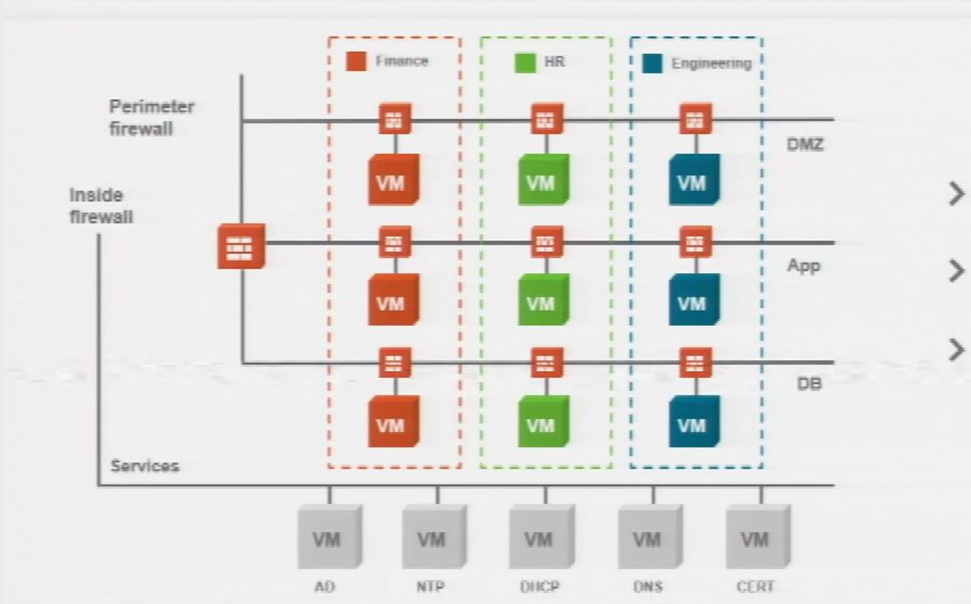
Each middlebox has state and policy associated with workloads

- State needs to be in sync across services (config + runtime)
- Policy needs to be enforced at high packet rates
- Independently solved scaleout + high availability
- Hard to adapt to dynamic workloads (tasks/VMs lifecycle/moving)

magma

Modern datacenters: Fabric and policy rich edge



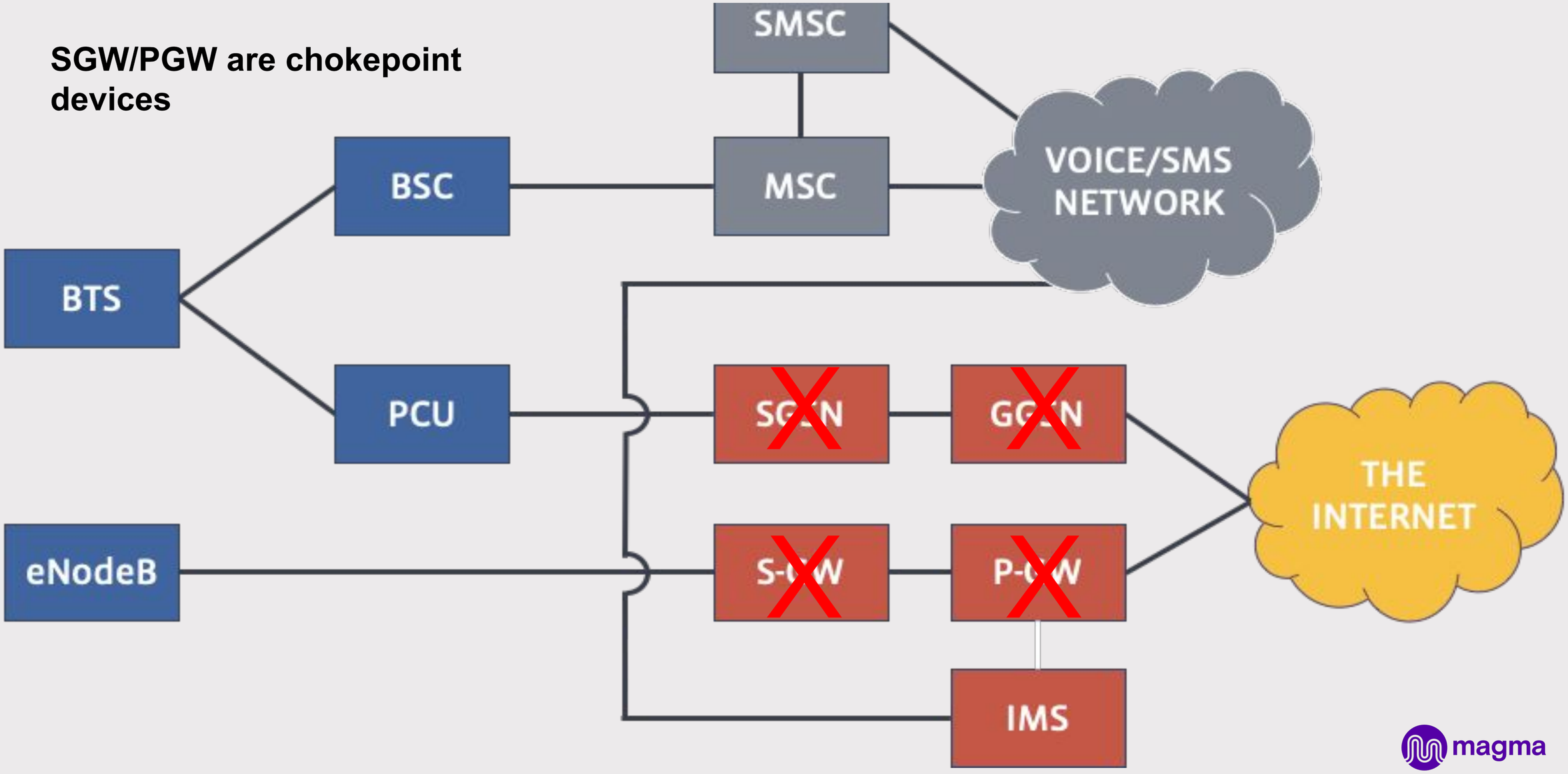


Modularize the network: Fabric responsible for moving packets faster. Distributed edge responsible for rich policy enforcement and state

magma

Today’s GSM/LTE architecture

SGW/PGW are chokepoint devices

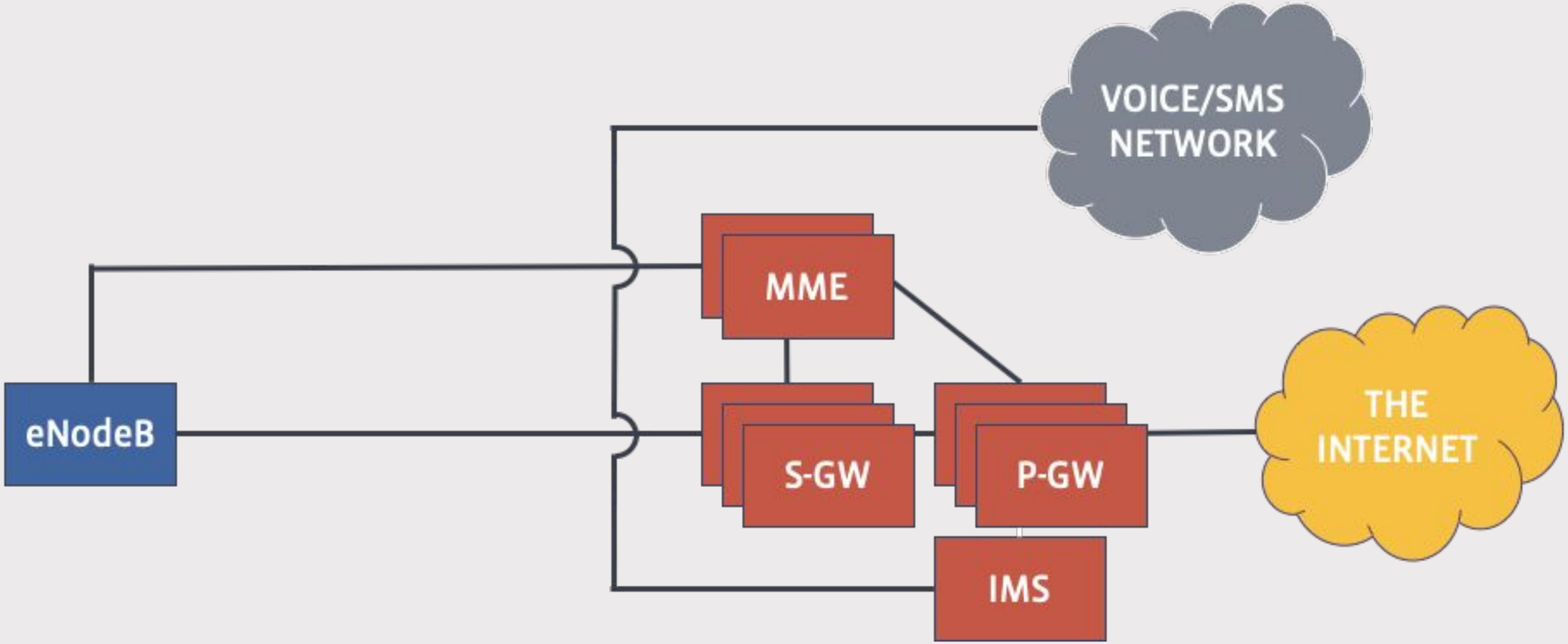


magma

Encapsulation in traditional LTE networks

UE state exists in all nodes

- No clear device abstractions
- Air interface specifics leak through the network



magma

Magma takeaway: Encapsulation and the fabric

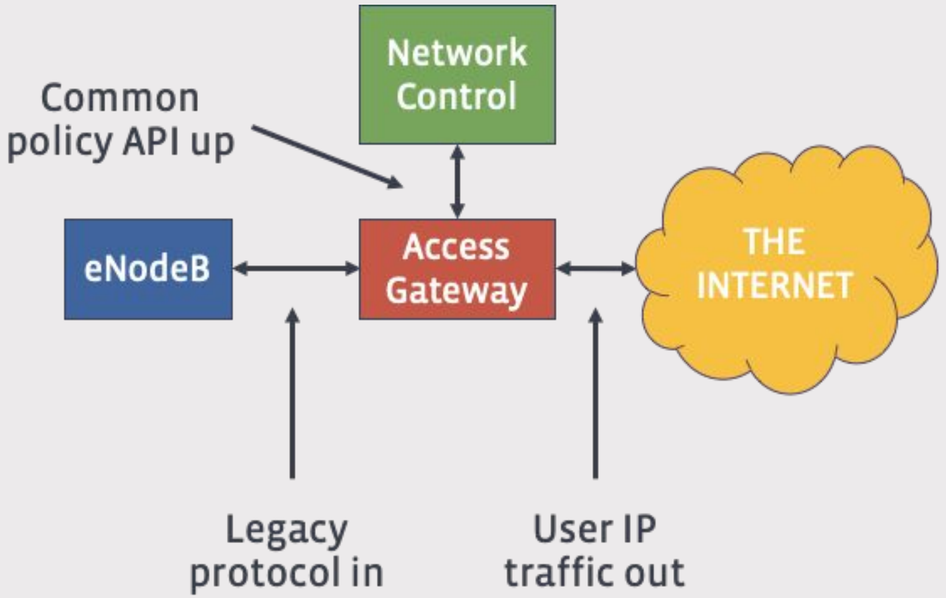
- Encapsulate the UE state
  - Config maintained in a central location
  - Runtime encapsulated at the edge
- Distribute policy enforcement point
  - Let the ideal topology decide the policy enforcement point
- Keep core network simple
  - Allows for easy scale up/down
  - Cheap: Core network only needs to move packets fast
- Abstract away radio specific technology
- Focus on **operationalizing** the network

Mobility is complicated but solved in the datacenter (IP is both identity and location)

State in the control plane

Magma takeaway 2: State in control plane

- Desired state model: Centralized through APIs
  - User inputs intent, control plane enforces it
- Control logic completely decoupled from datapath
  - Programmable APIs exposed by datapath
  - Independent evolution of control + datapath
- Use modern distributed systems to propagate state
  - HTTP2, Protobuf, K/V store

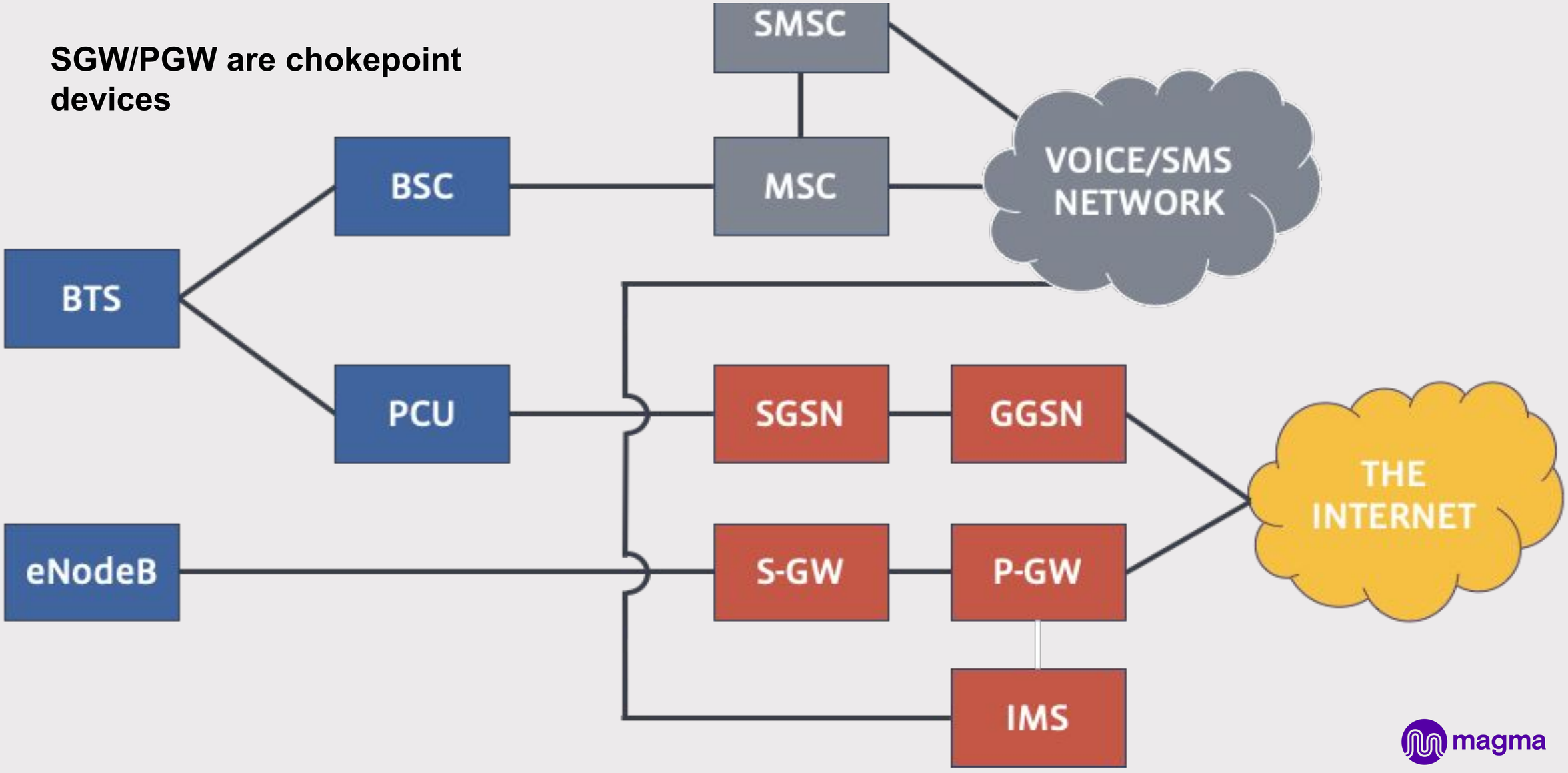


Software release

The need for fault domains

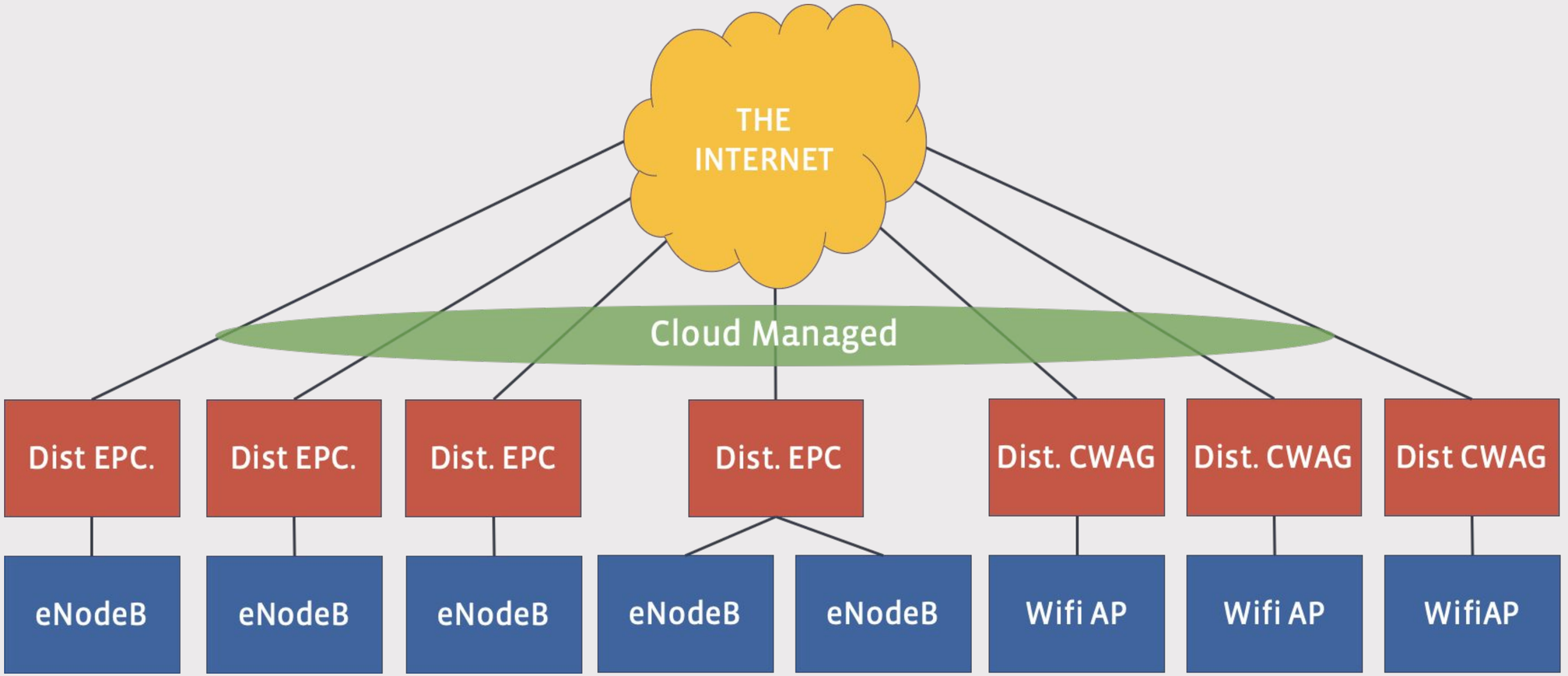
Software delivery: Too big to fail

SGW/PGW are chokepoint devices



magma

Software delivery: Fault domain



magma

Magma takeaway 3: Software upgrades

Design for localized fault domains

- Small upgrade domains
  - Each node is independently upgradable
  - Gradual rollout is baked into the platform
- Control plane independent from dataplane operations
  - Existing traffic not affected by control plane outage

Summarizing: Solving for heterogeneity

- **Flexibility:** Modularize the network into a fast fabric and a policy rich edge
- **Scalability:** Encapsulate UE state and use proven distribution techniques
- **Any spectrum (4G/wifi/5G):** Localize air interface specifics to the edge
- **Programmability:** Desired state store model with a centralized controller
- **Agility:** Design for upgrades by minimizing fault domains

Ovs collaboration oppurtunities

- **MSS clamping:** Many phones don’t respect MTU settings
- **Routing support:** Add support for route action:
  - Action managed by local agent or controller
  - Also add support for route advertisement.
- **GTP support in kernel**
- **IPFIX extension:** Add support for custom fields

Thank You