Connecting the World:
A look inside Facebook's Networking Infrastructure

Arun Moorthy
arunm@fb.com
https://fb.me/arun.moorthy

about:me

- B. Tech CSE – Indian Institute of Technology, 1997
- MSCS, University of North Carolina, 1999
- Current: Facebook Inc, (2012 –)
  - Engg. Mgr on Network Team: Load-balancing, Transport Security
- Previous: Intel Corp, Nexsi Systems, Microsoft Corp, 1999 – 2011

Agenda

- Challenges
- Facebook's Global Network
- Software Load Balancing
- POP Network Architecture
- Data Center Networking
- Concluding Remarks

The Challenges

- 1.4+ Billion users
- 1+ Tb/s egress
- 4B+ video views/day
- 1+ Million Requests/sec
- Worldwide user-base (80+% users outside US and Canada)
- Highly available + reliable

Did I mention “Highly Available”?


Datacenter

Datacenter

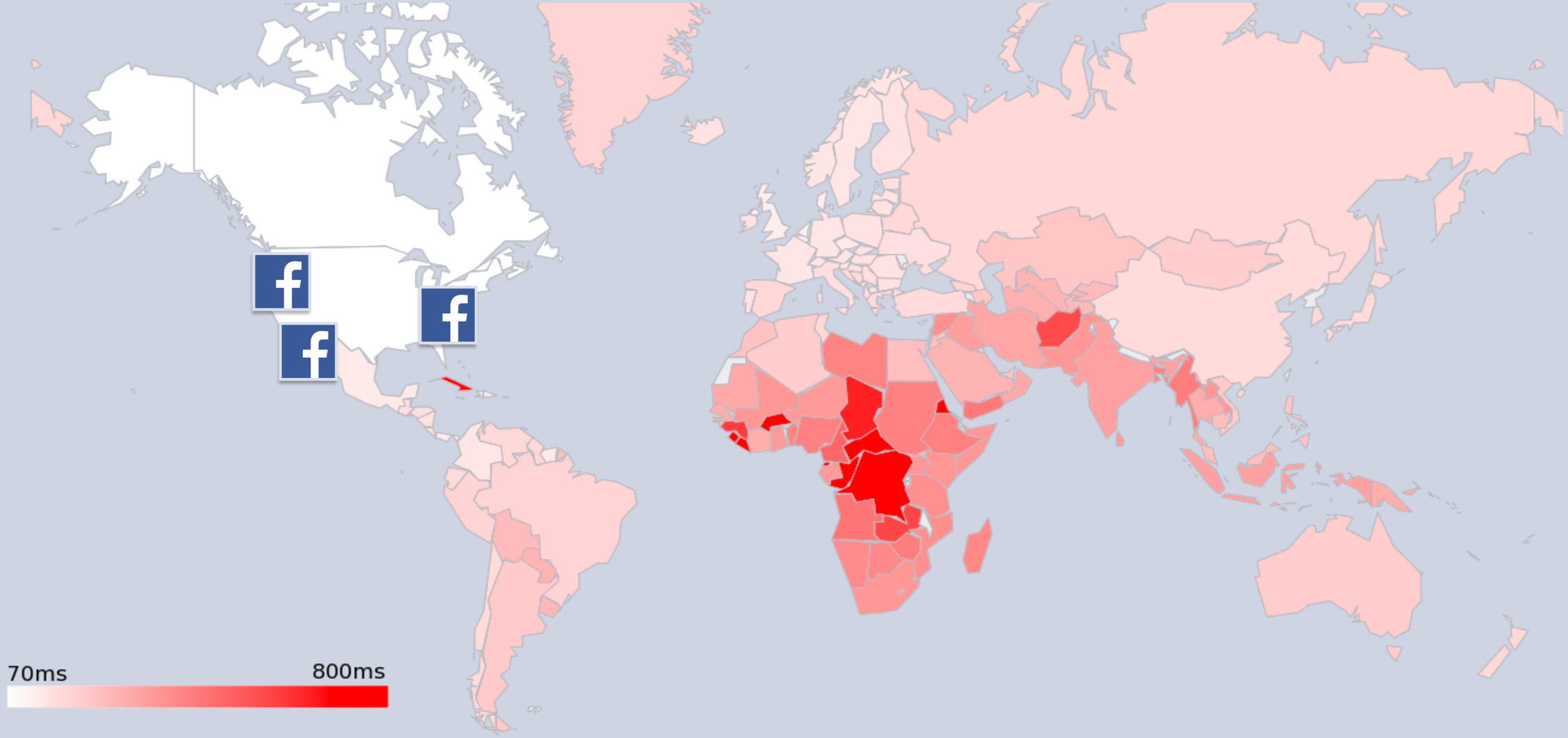

Datacenter
Edge PoP

Datacenter

Edge PoP

International RTT

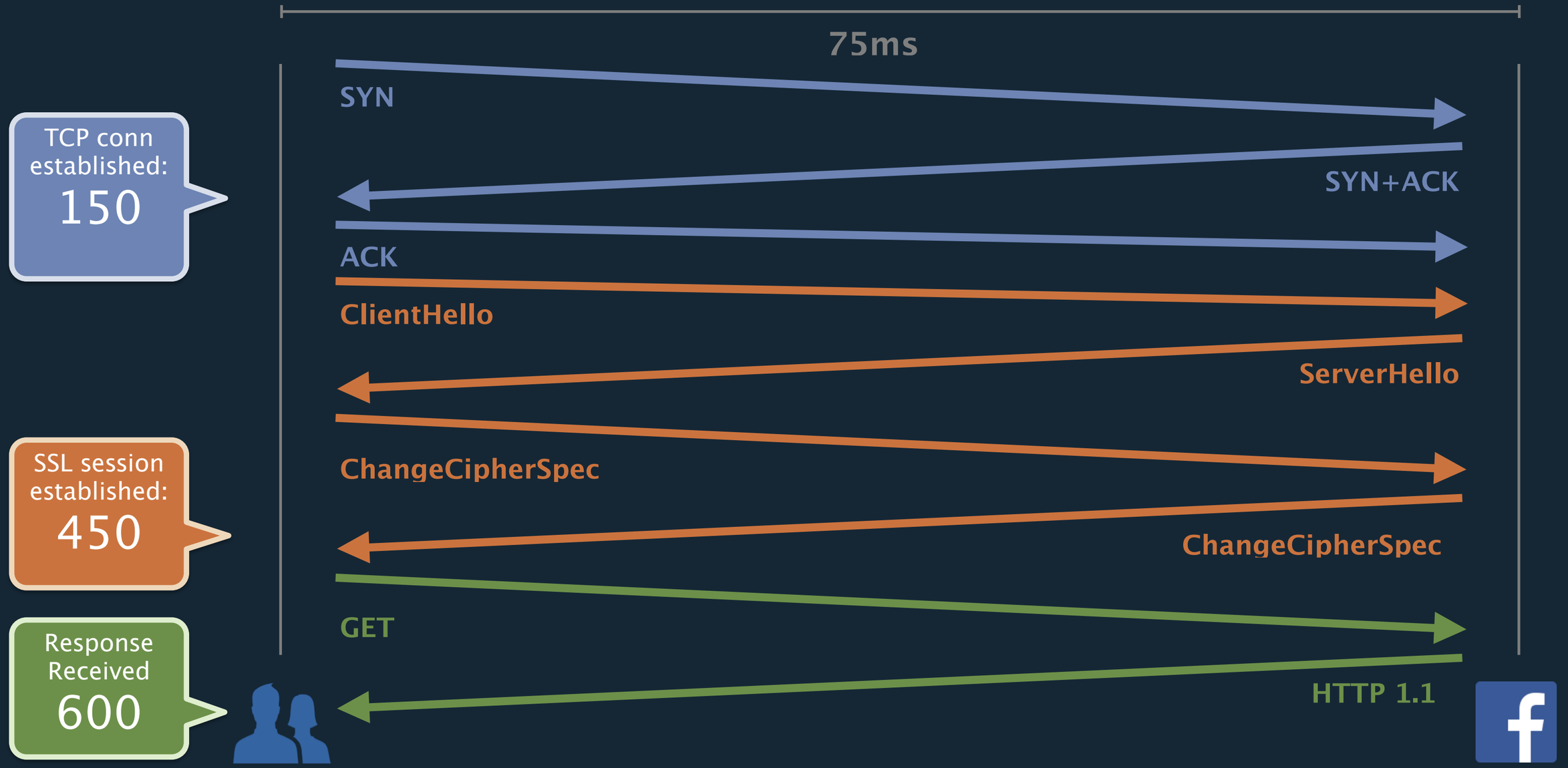
circa 11/2011



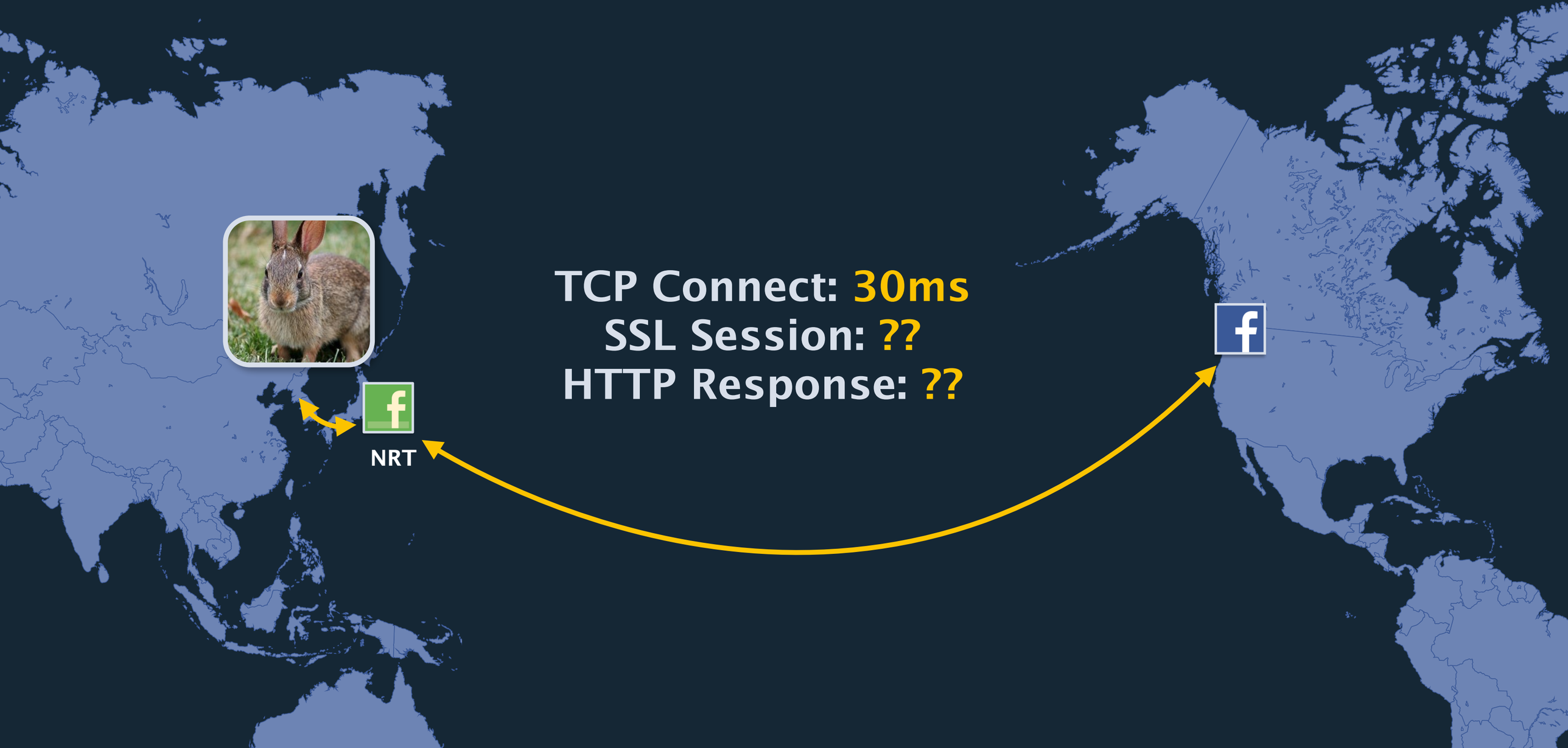
Seoul -> Oregon

TCP Connect: 150ms

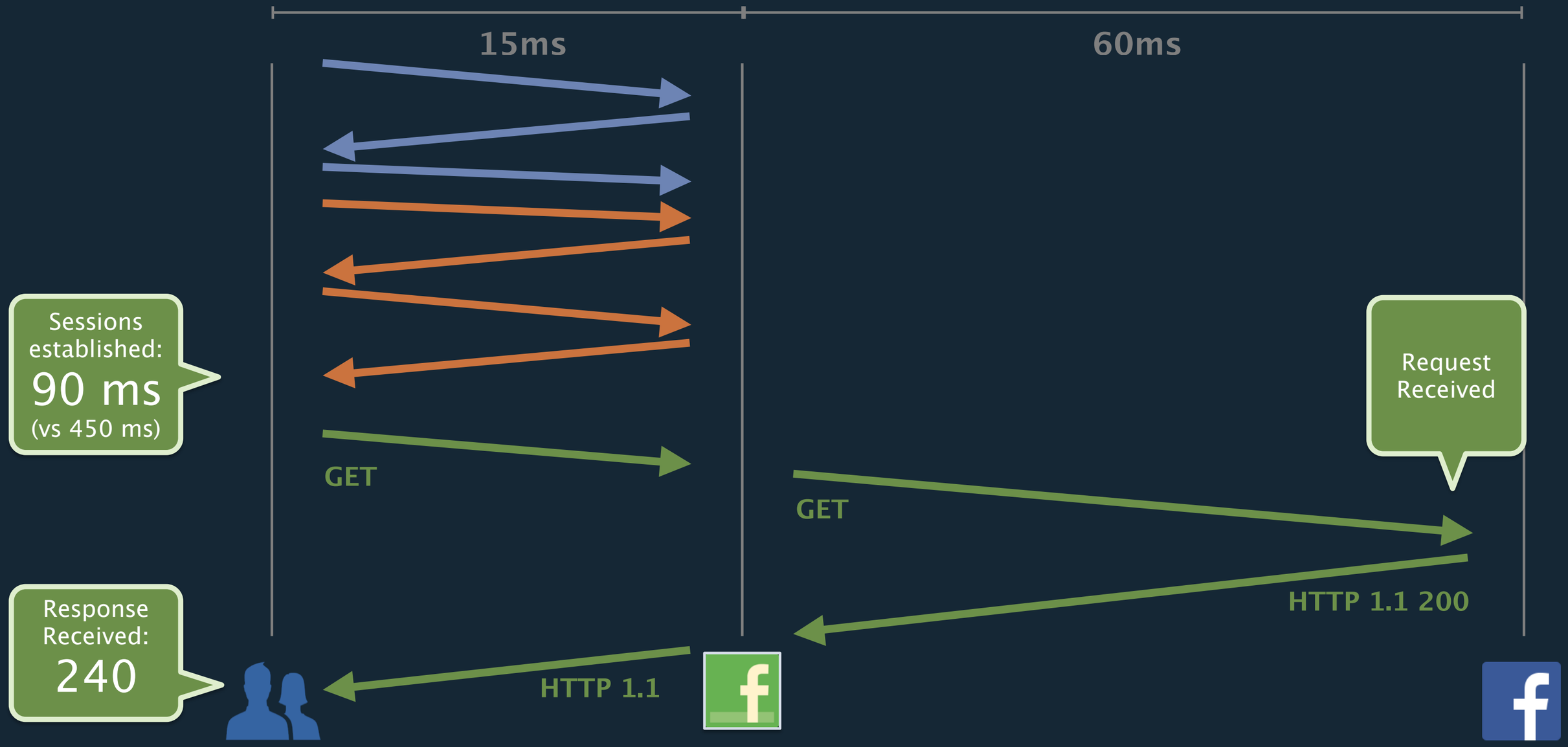
HTTPS Seoul -> Oregon



Seoul -> Tokyo -> Oregon



HTTPS Seoul->Tokyo->Oregon

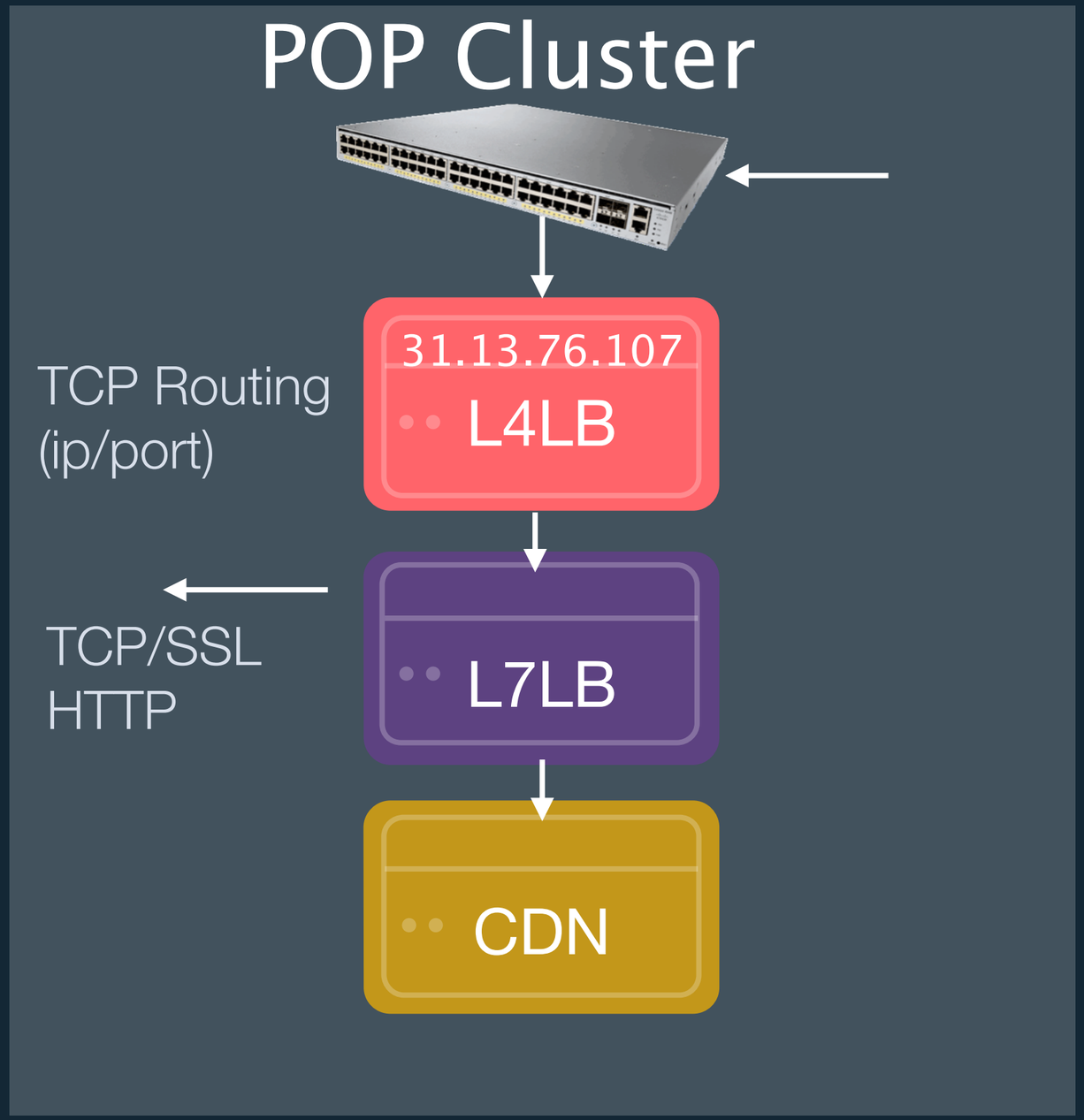


Seoul -> Oregon

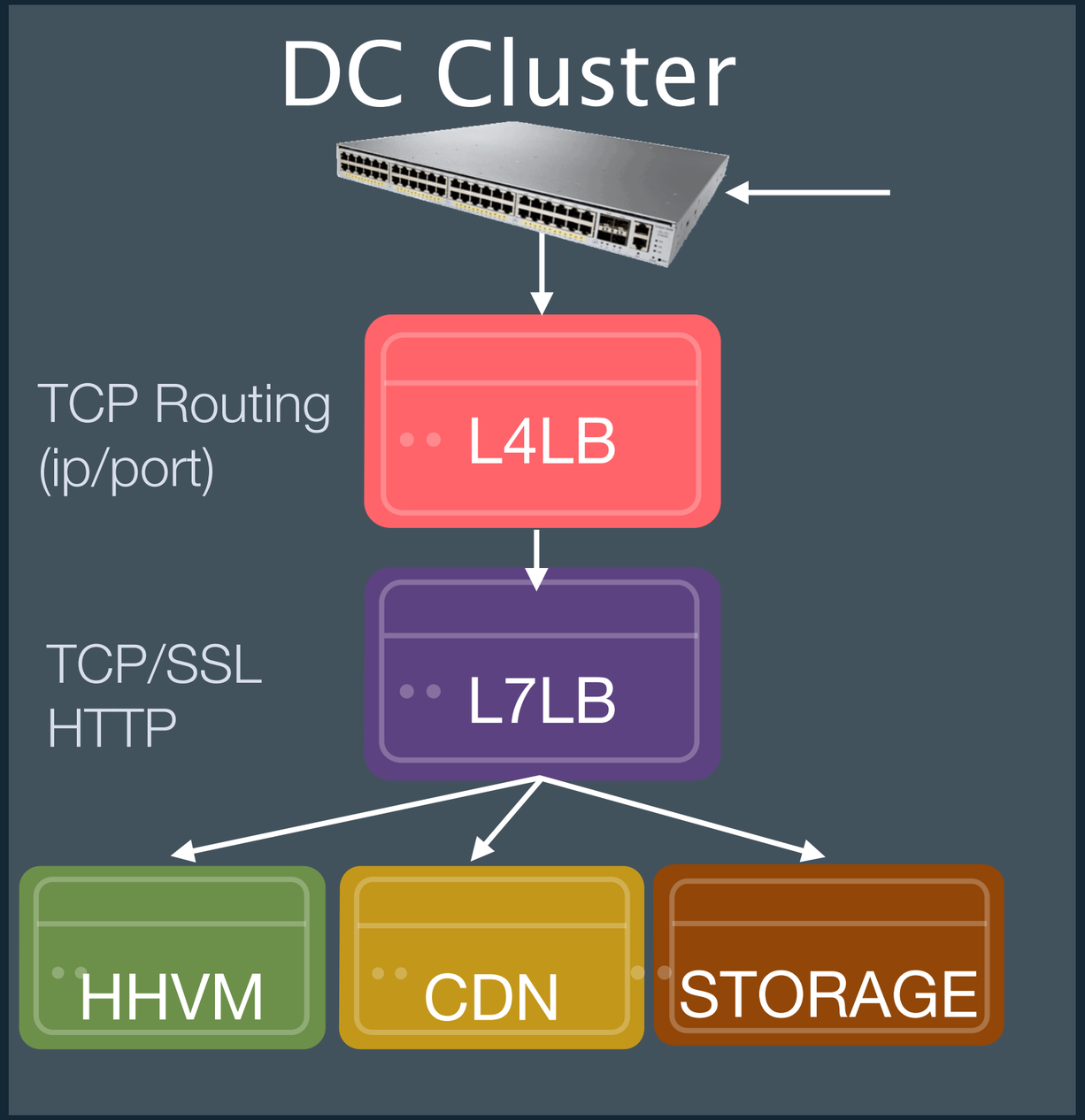
NRT

TCP Connect: ~~150ms~~ 30ms
SSL Session: ~~450ms~~ 90ms
HTTP Response: ~~600ms~~ 240ms


POP


POP Cluster
TCP Routing
(ip/port)
31.13.76.107
L4LB
L7LB
TCP/SSL
HTTP
CDN

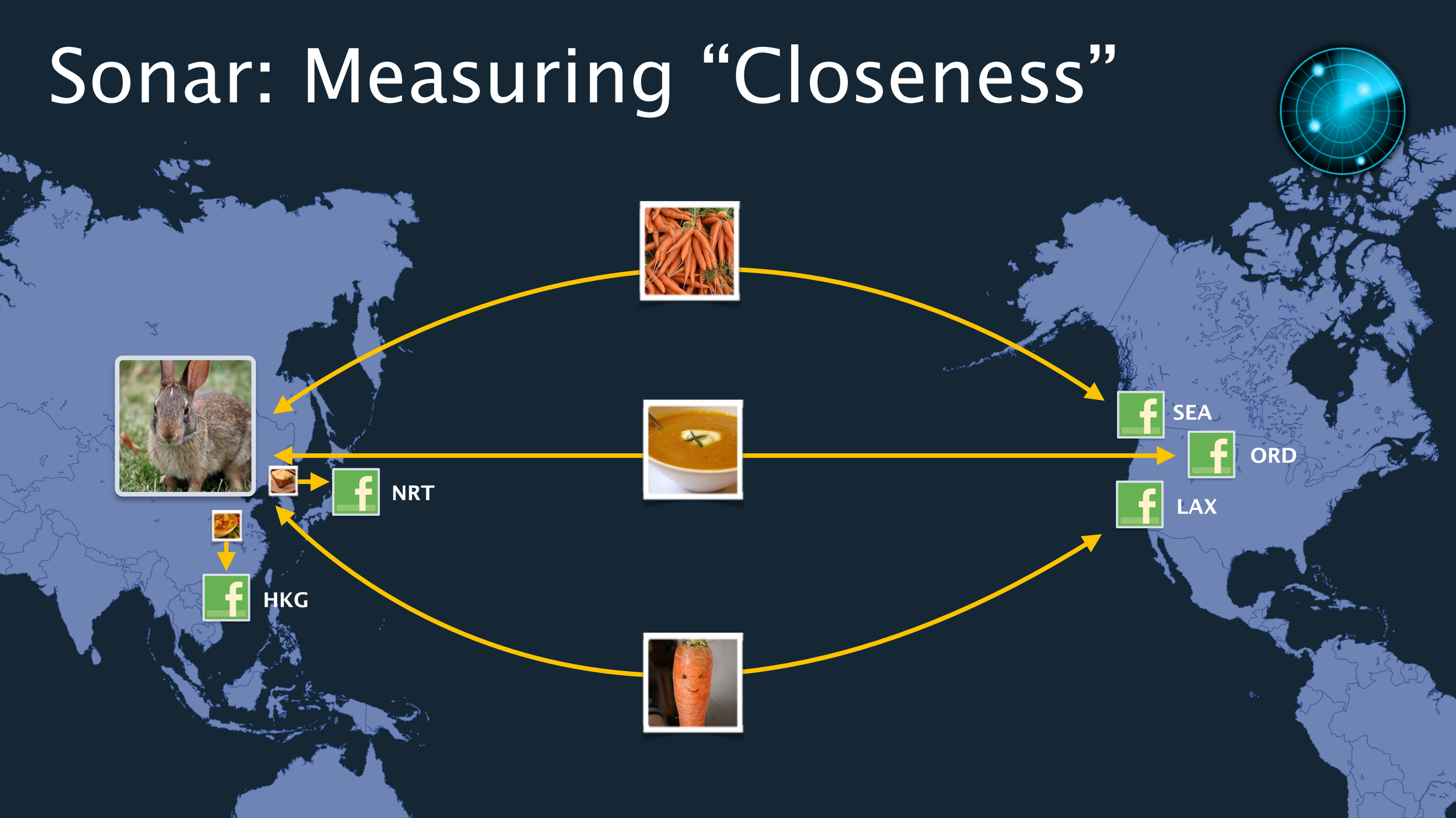

DC
POP


DC Cluster
TCP Routing
(ip/port)
L4LB
TCP/SSL
HTTP
L7LB
HHVM
CDN
STORAGE

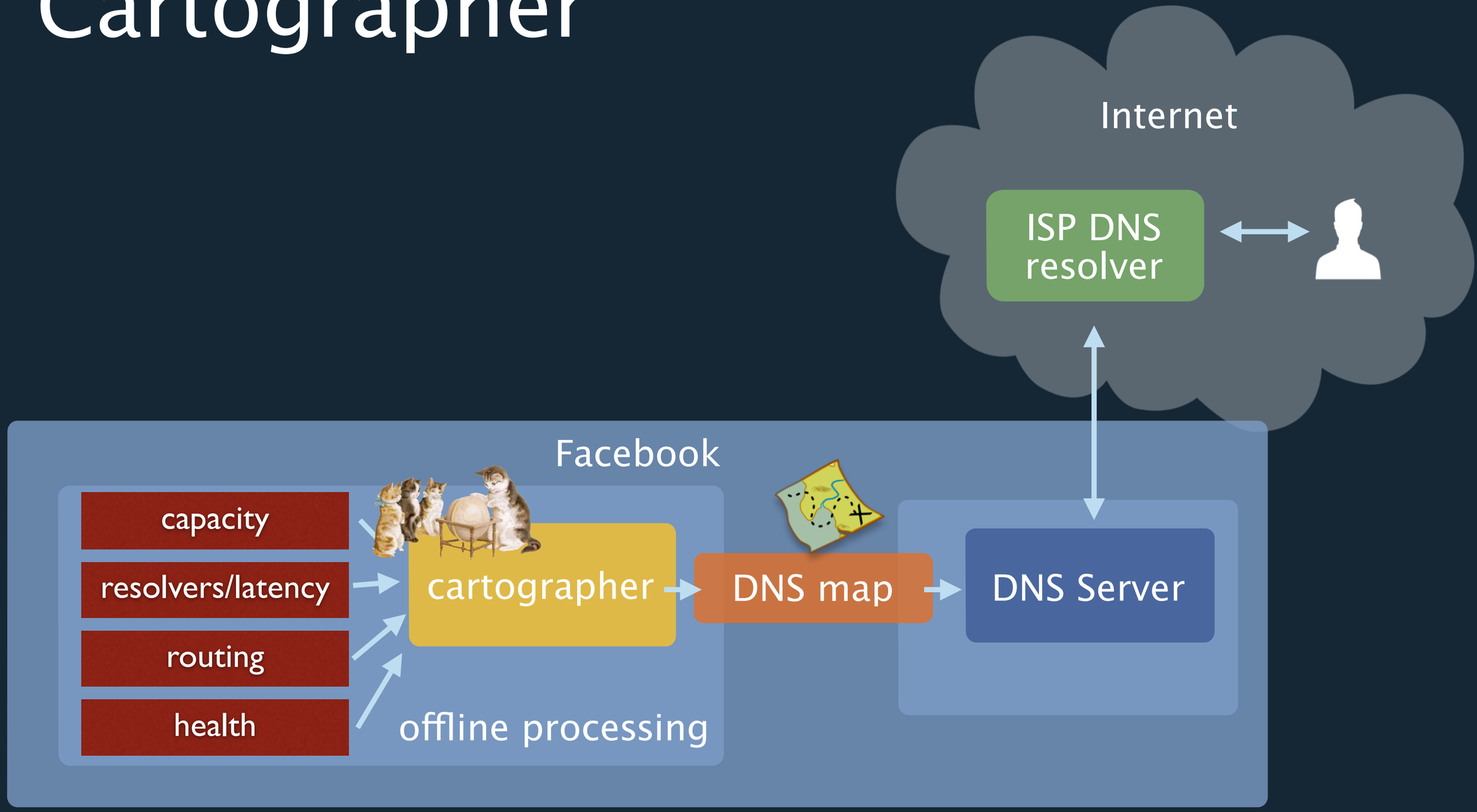
Benefits of the “Edge”

- Reduced latencies (Edge Termination)
- Caching static content (CDN)

Sonar: Measuring “Closeness”



Cartographer



Proxygen

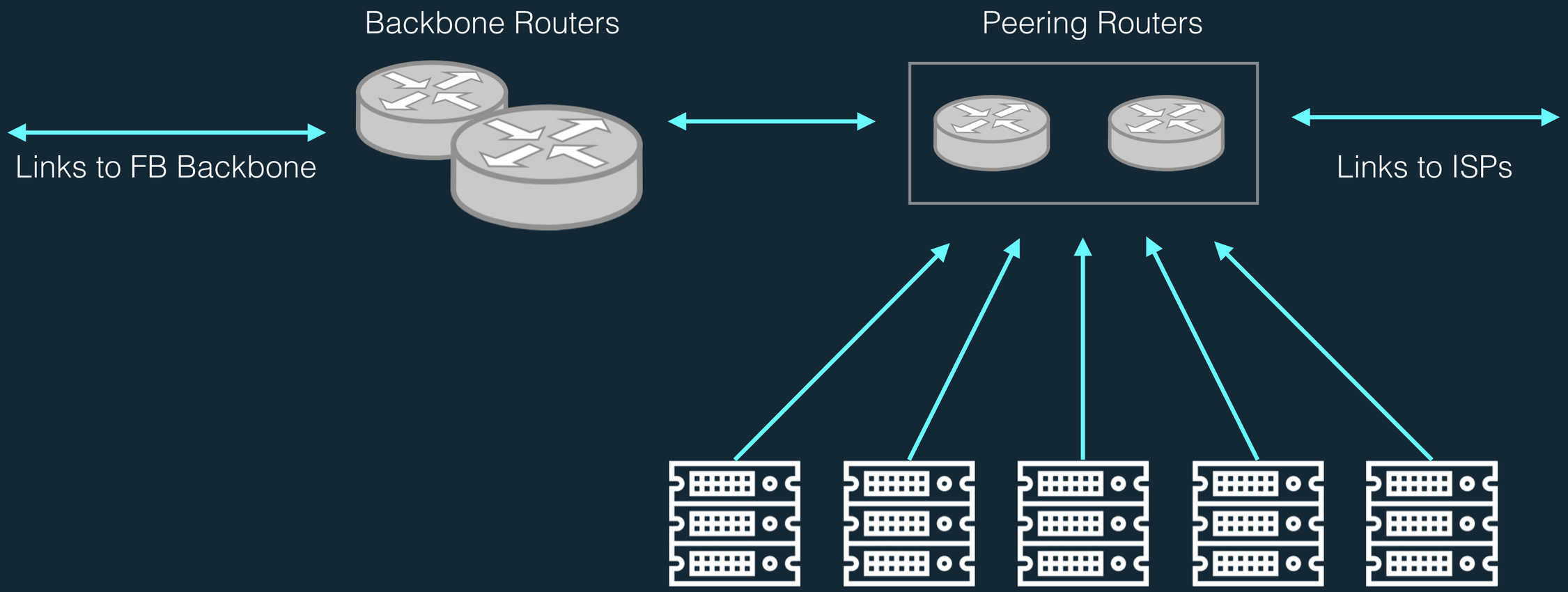
HTTP Framework

- High-performance C++ Server & Client
- Customizable Forward/Reverse Proxy
- Open-source
- Mobile Proxygen:
  - Cross-platform
  - Deep instrumentation
  - Modular components: DNS TCP TLS HTTP SPDY

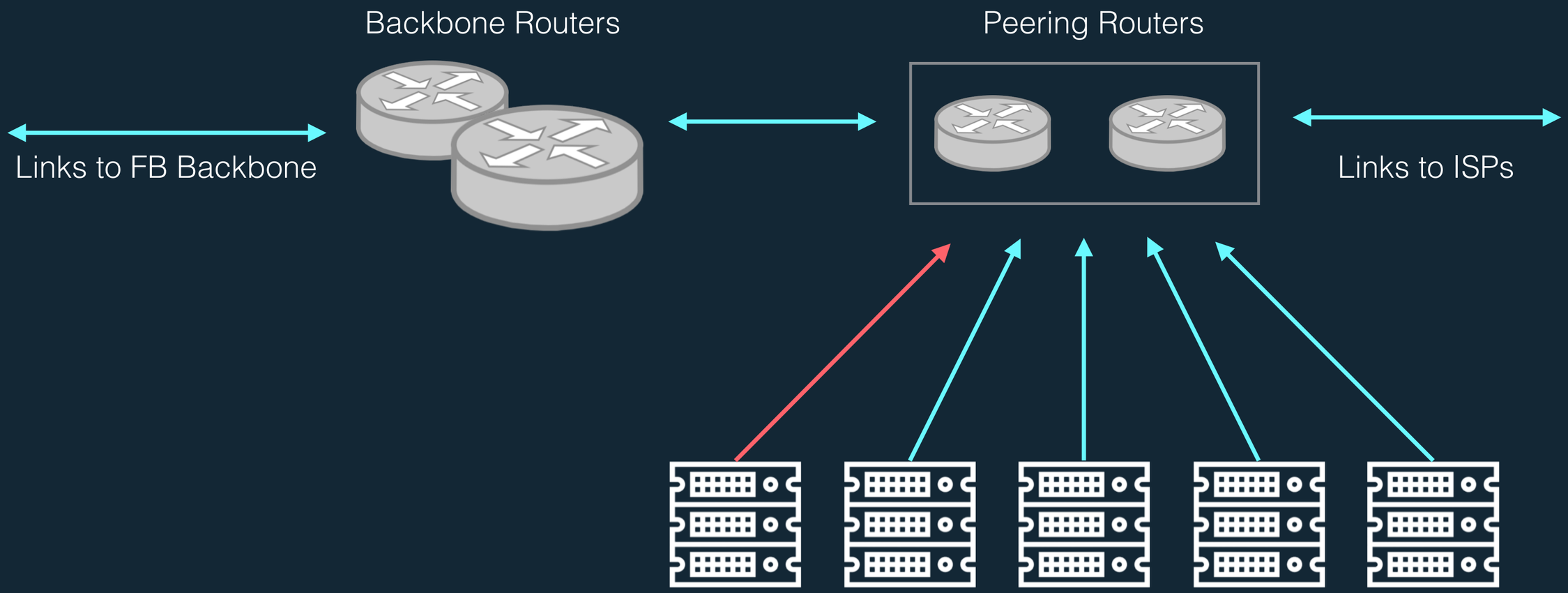
And More

- Shiv: L4 Load-balancer based on IPVS + python
- Edge-Fabric: Intelligent Interface utilization in POPs
- FB CDN: BigCache

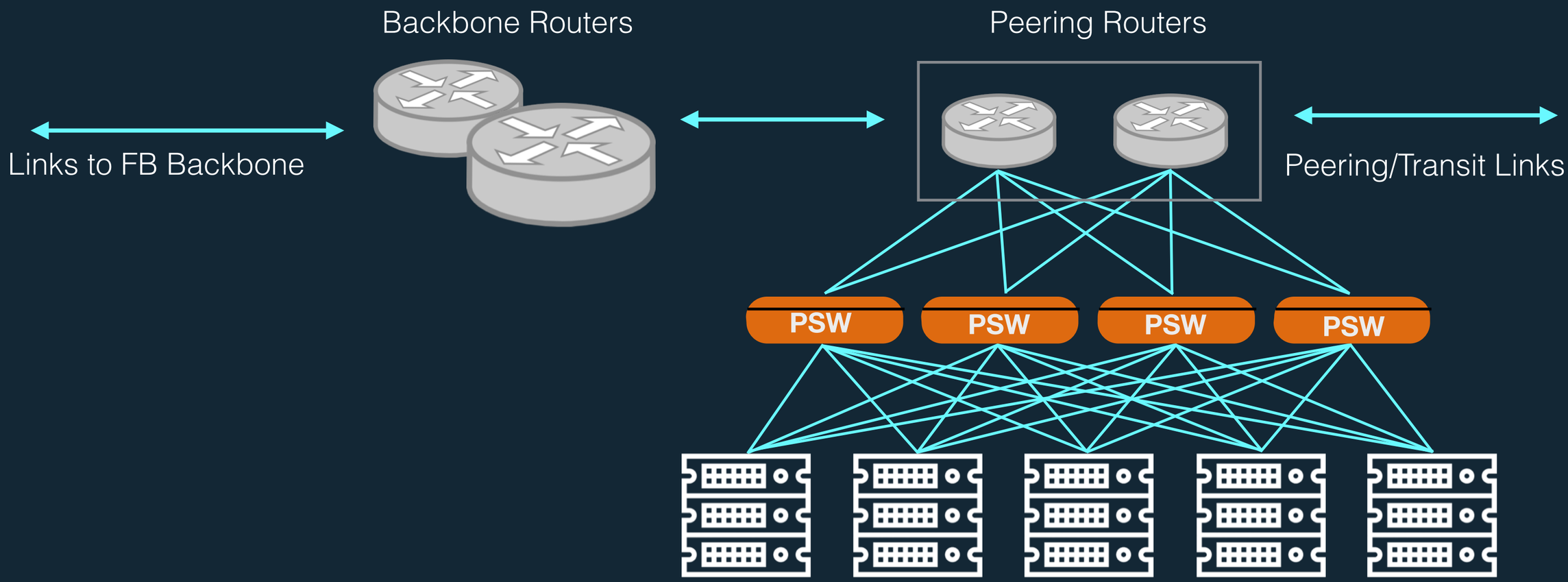
Previous PoP Architecture



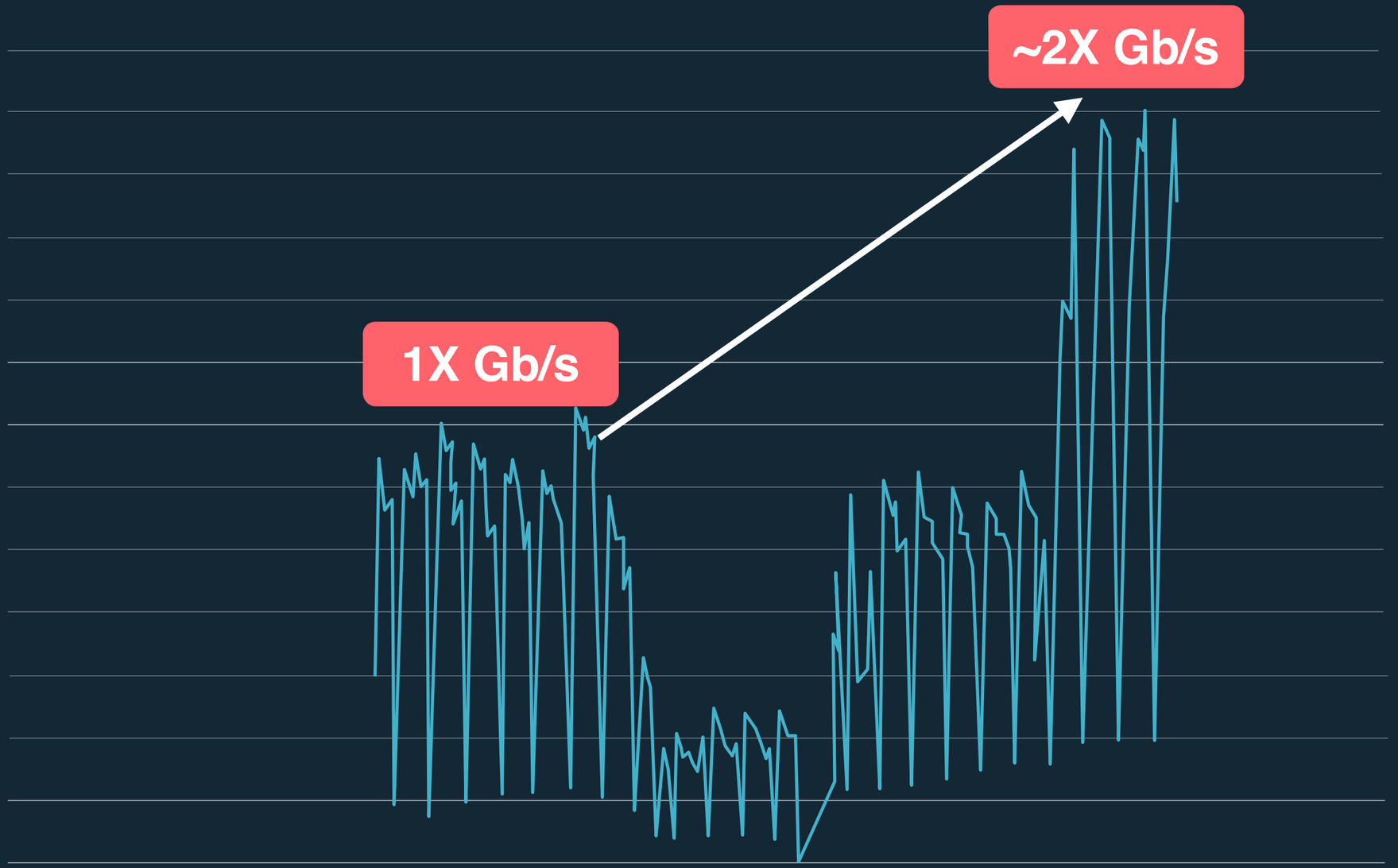
Previous PoP Architecture



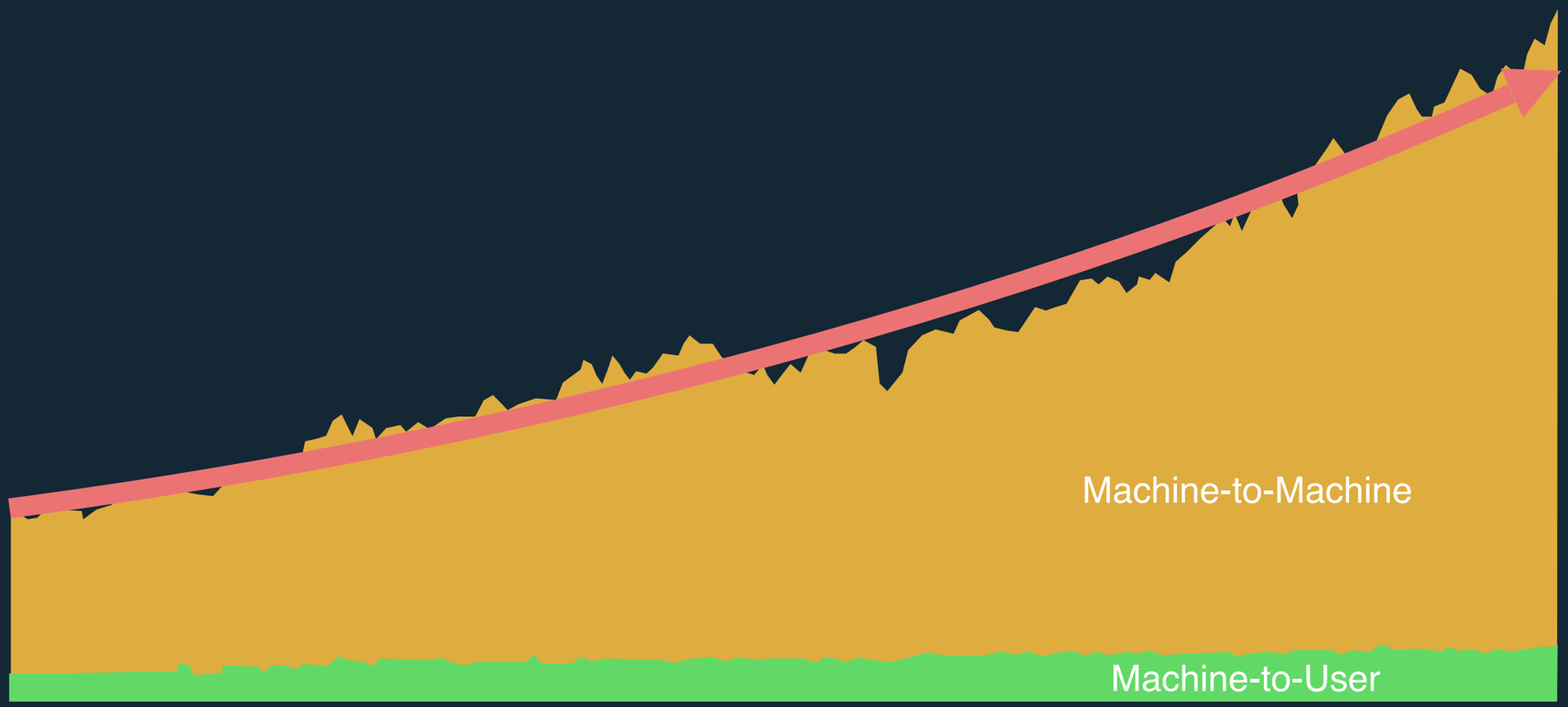
Fabric-style PoP Architecture



Edge Cluster Upgrade

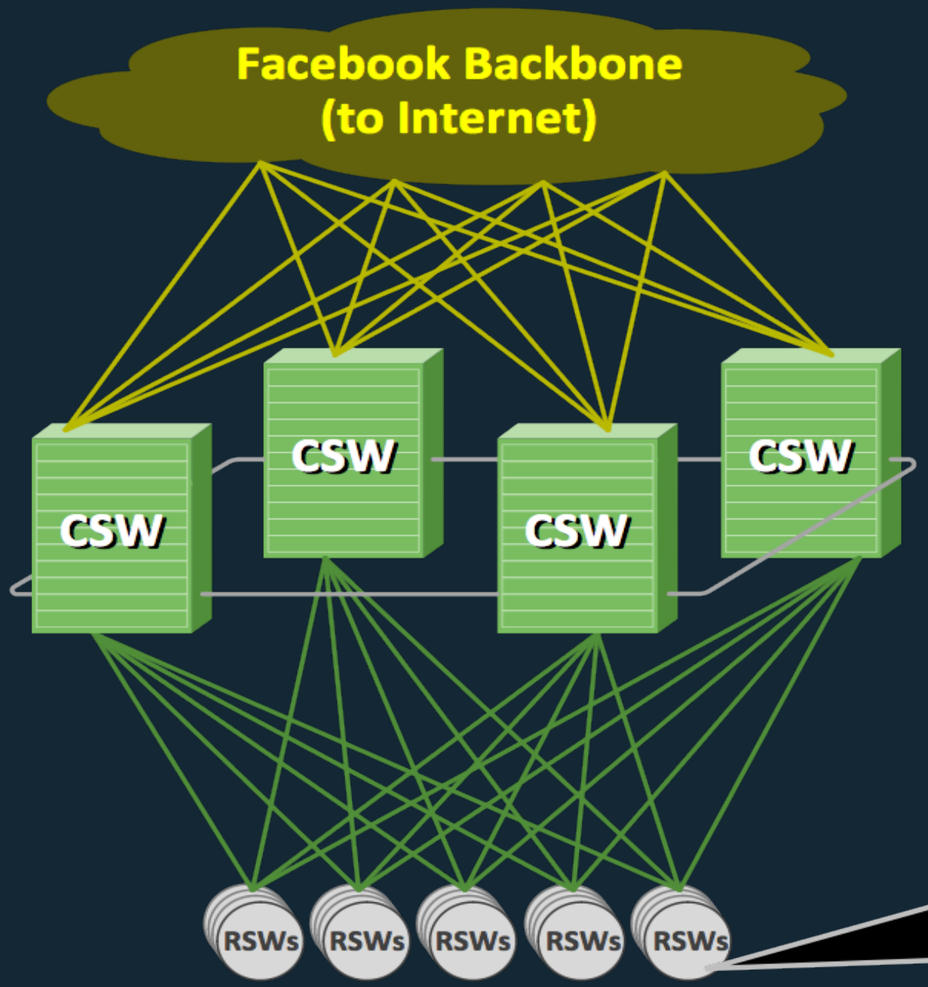


Rise of the @scale data center network

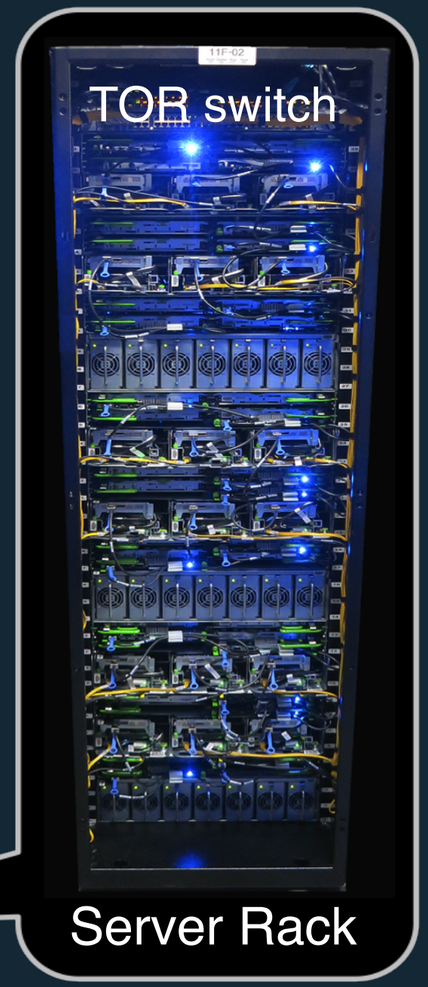


The 4-post cluster - our old design

4 x Large Cluster
Core Switches

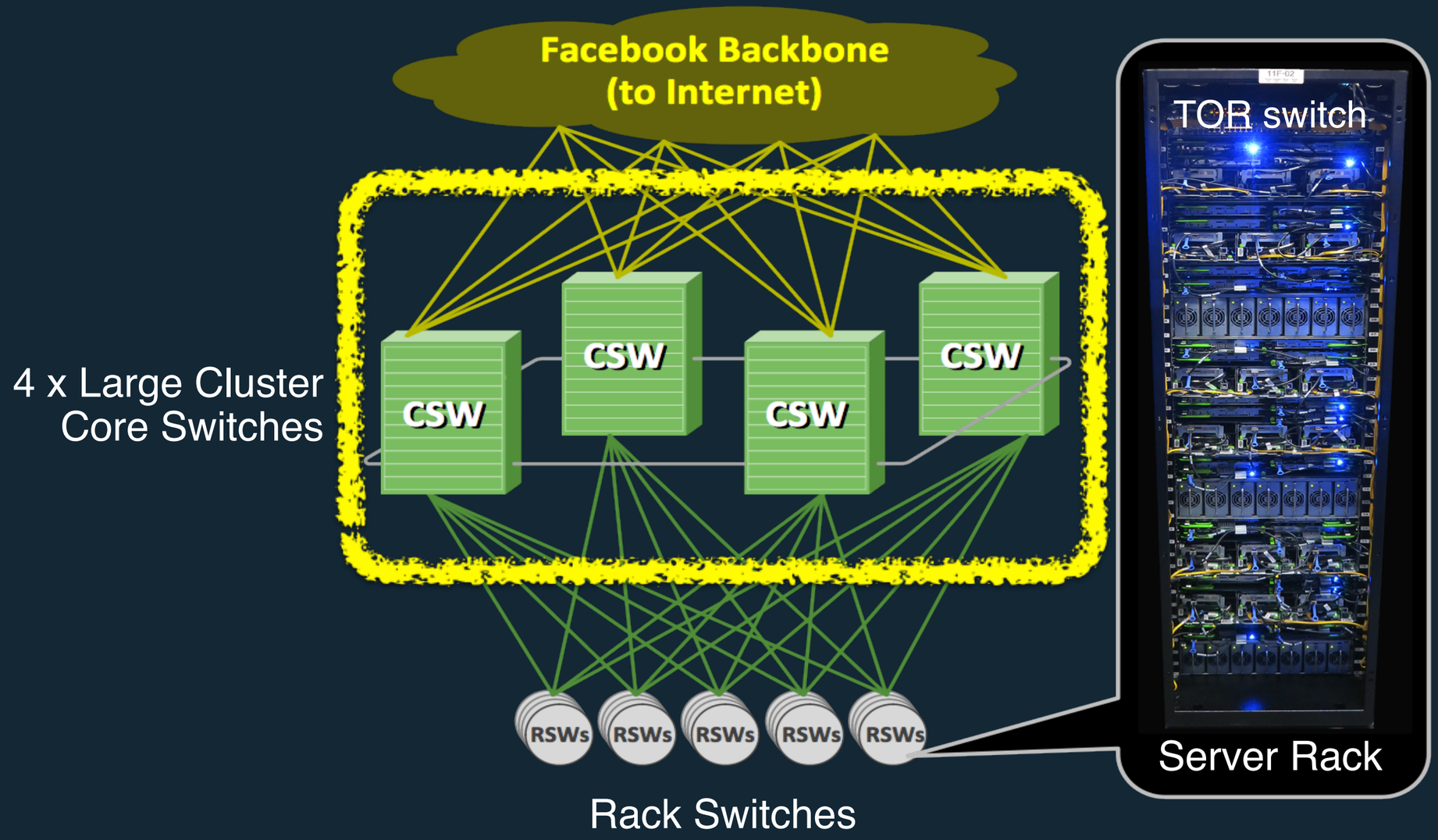


Rack Switches

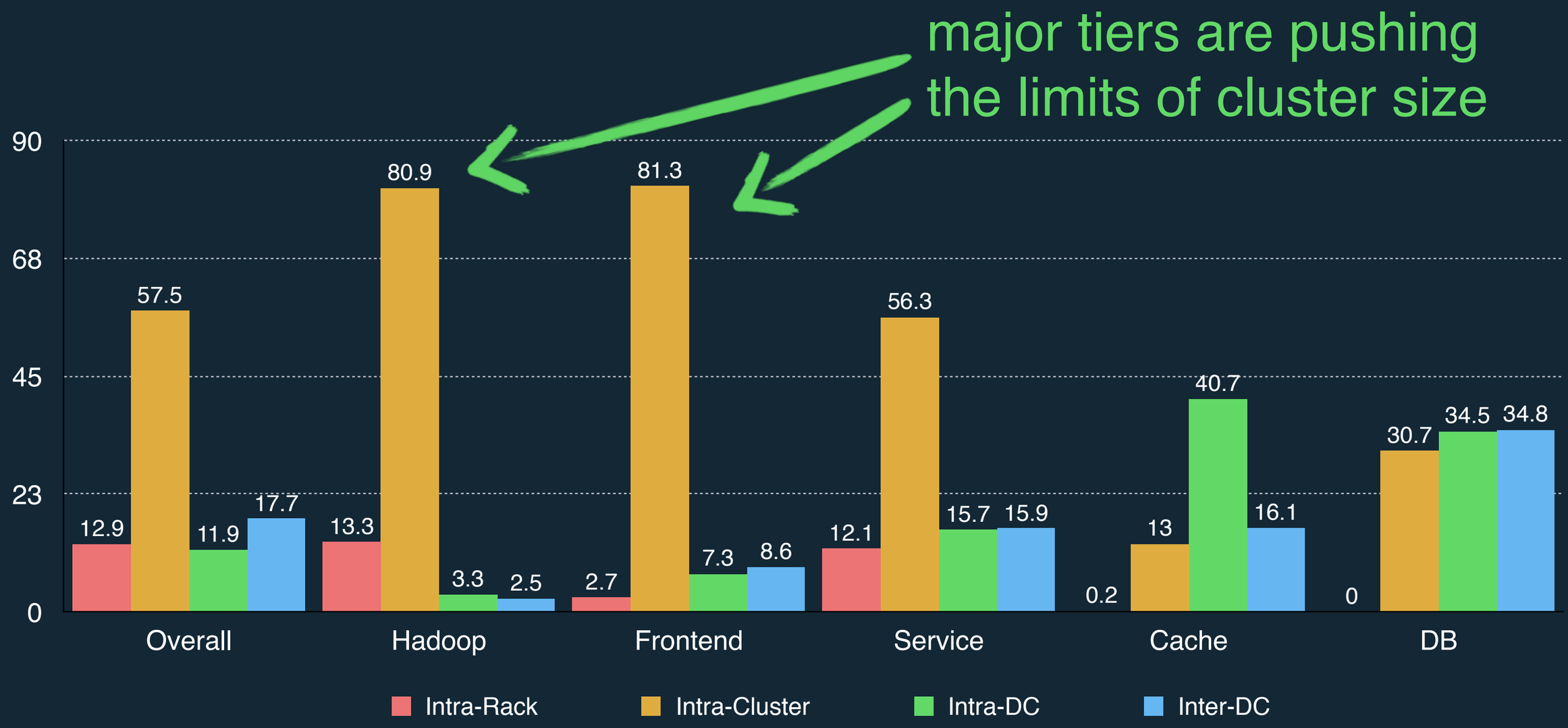


Server Rack

Box size limited cluster size



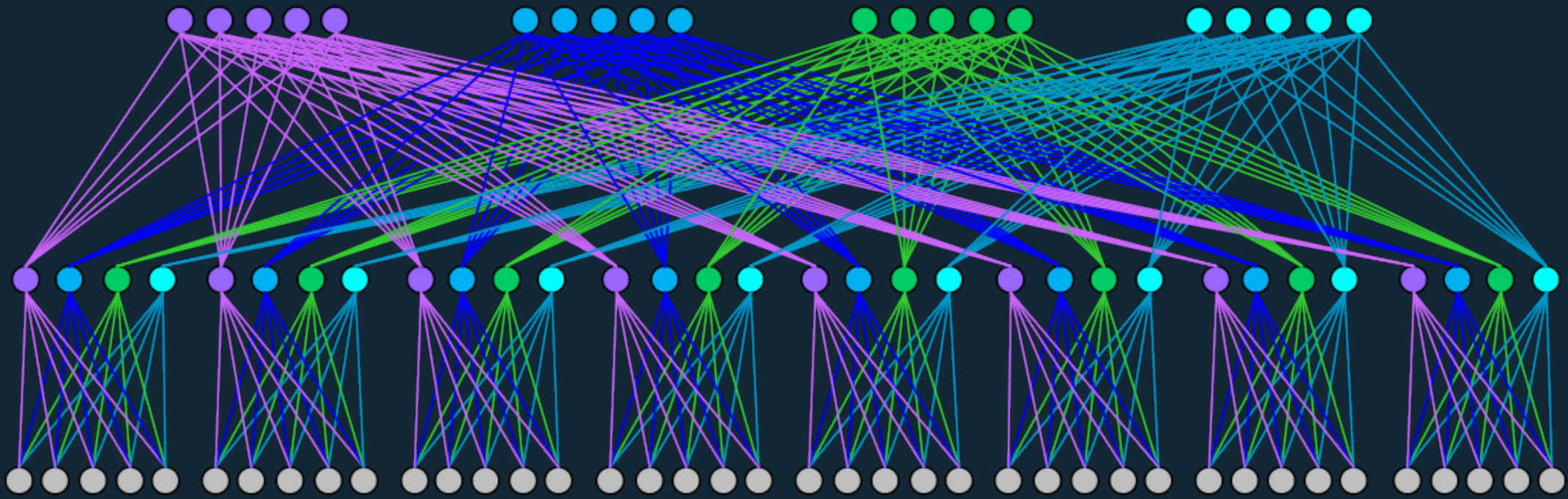
Cluster size limited application size



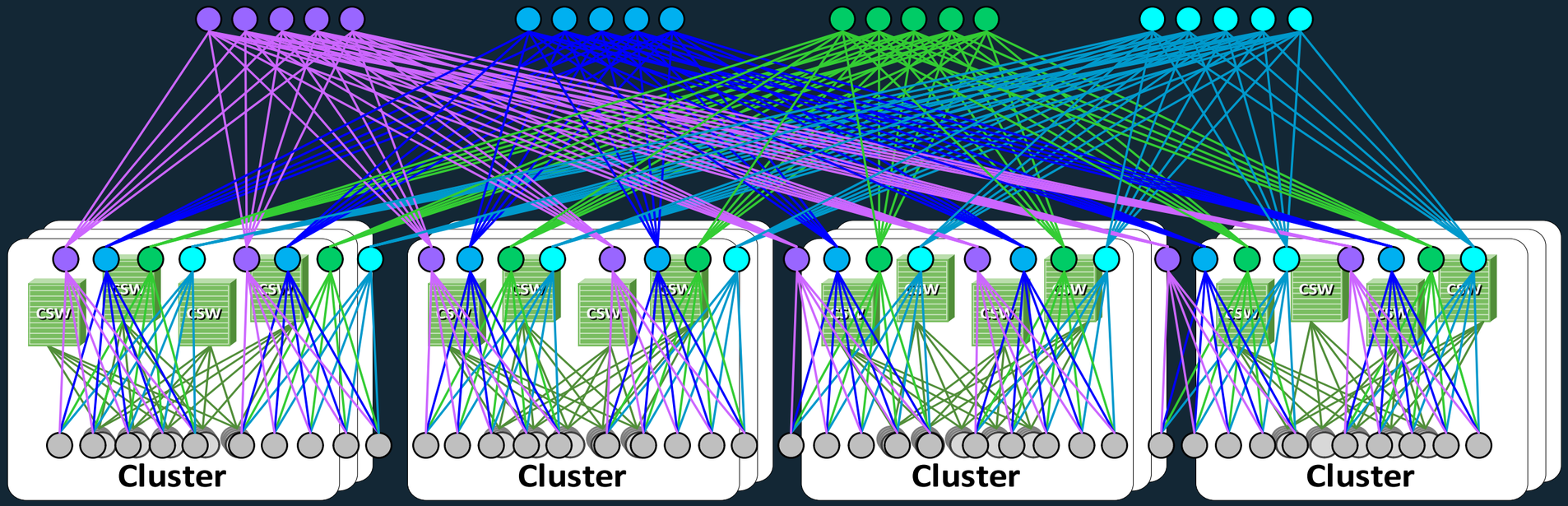
The Vision: The Whole Data Center, Redone.

The topology

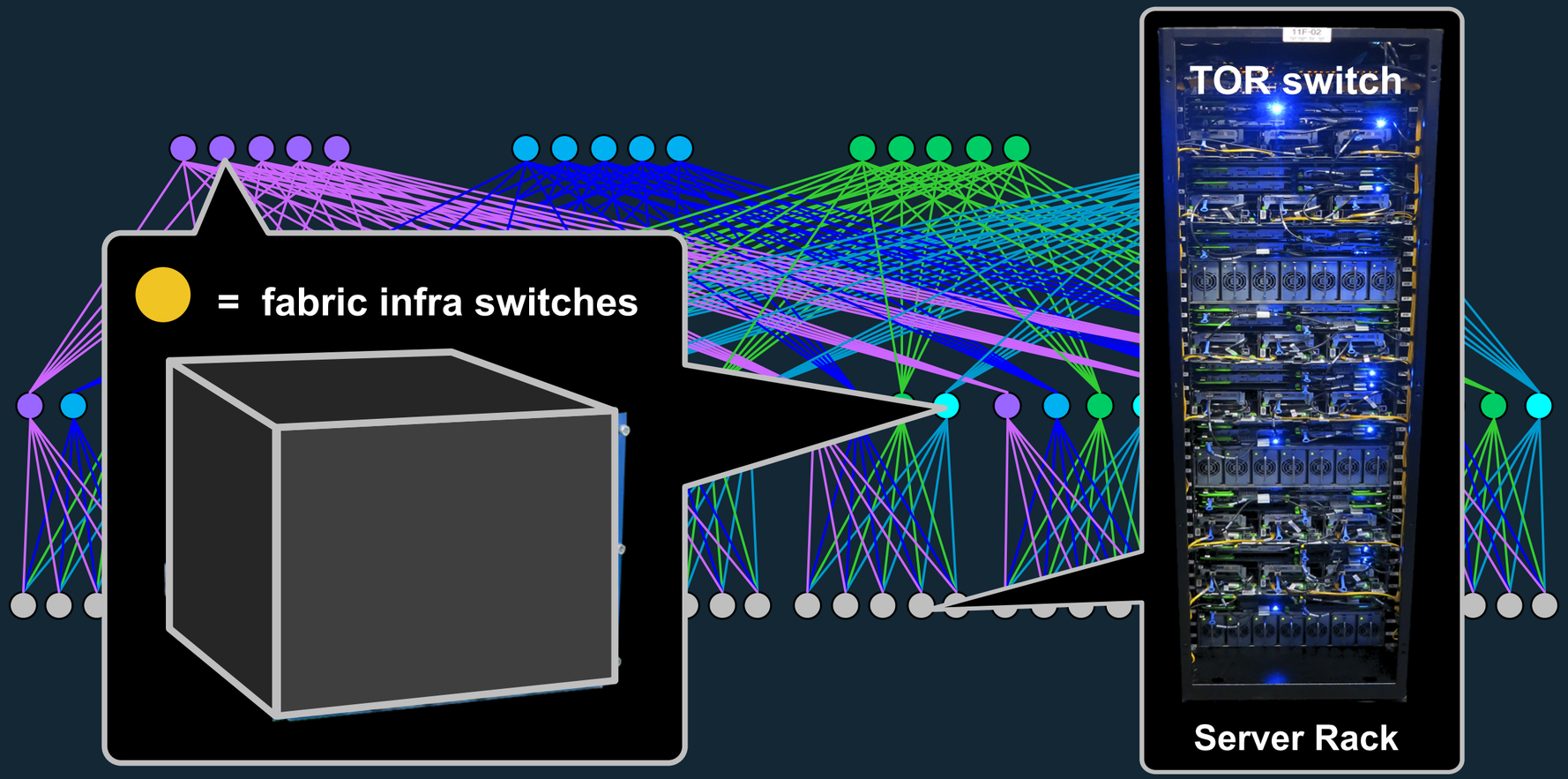
Facebook Fabric:
an innovative network
topology for data centers



The Fabric: one datacenter-wide network



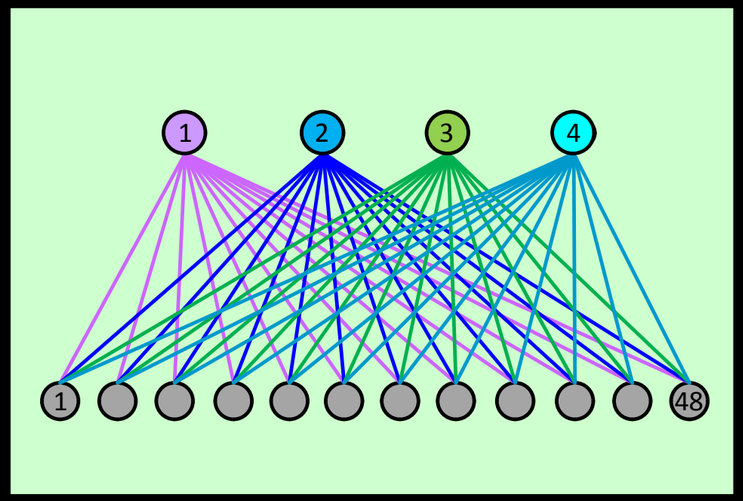
The Fabric: one datacenter-wide network



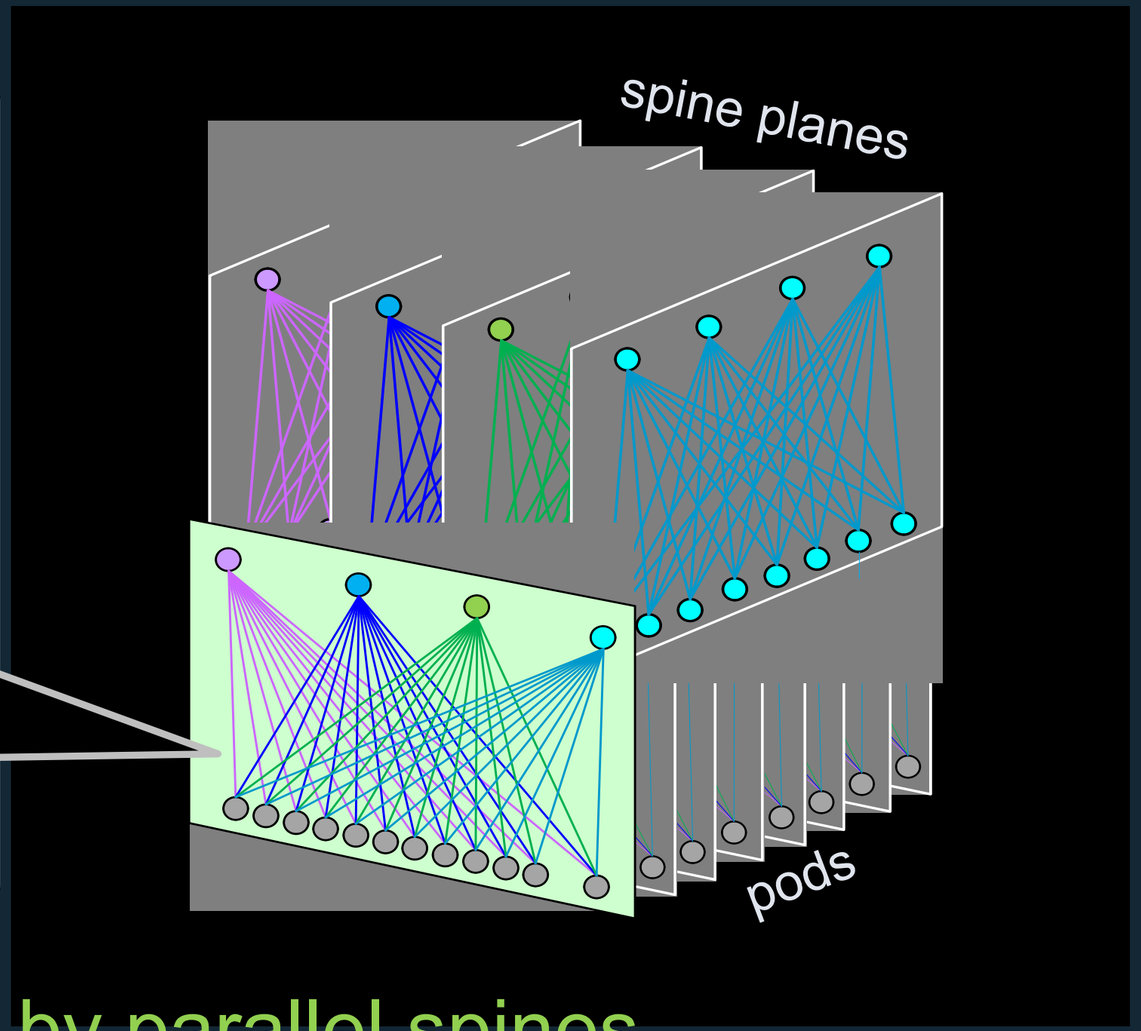
small & simple boxes

Server Pod: a [small] unit of deployment

4 fabric switches

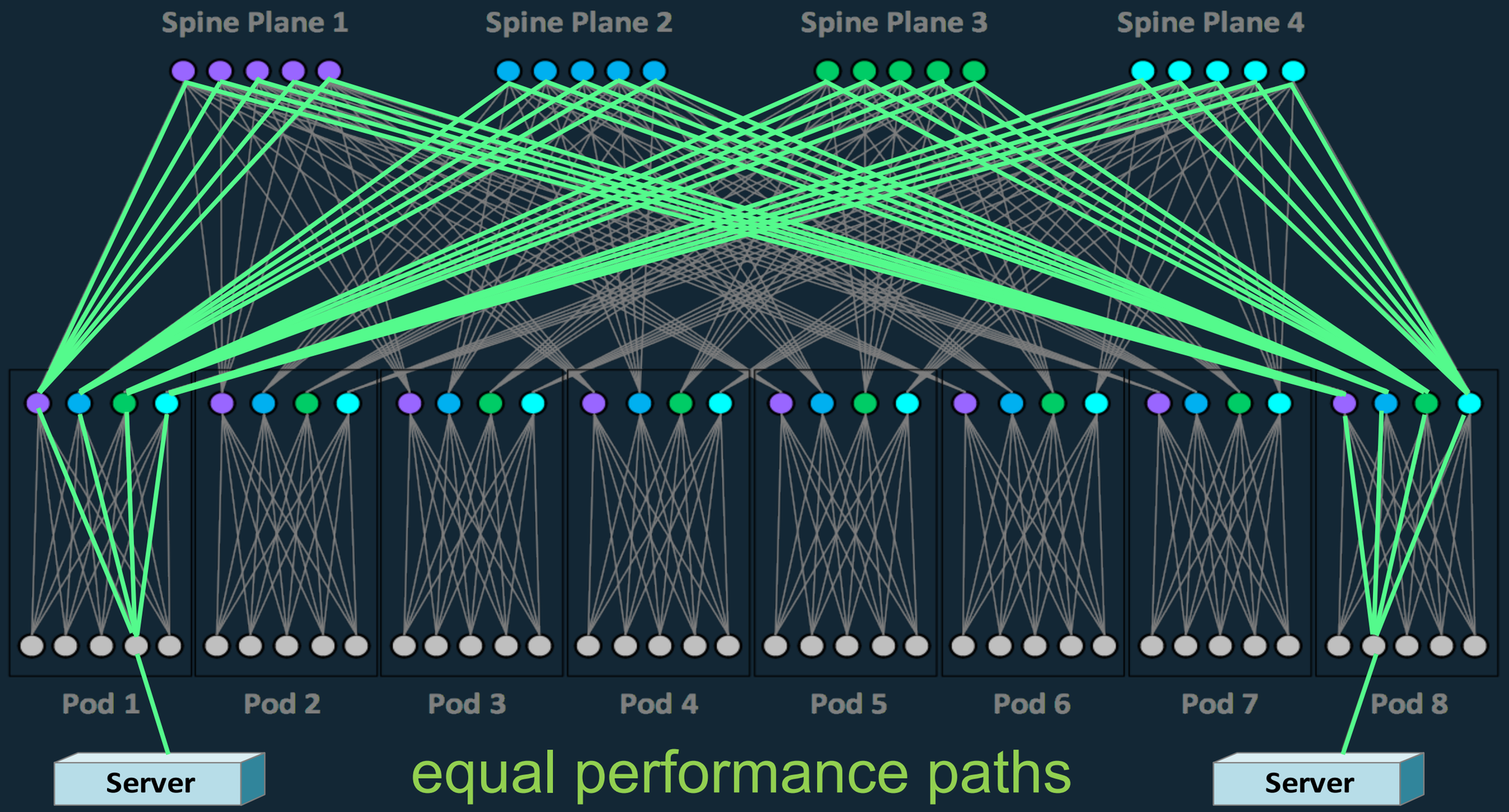


48 rack switches



pods interconnected by parallel spines

Many paths between servers



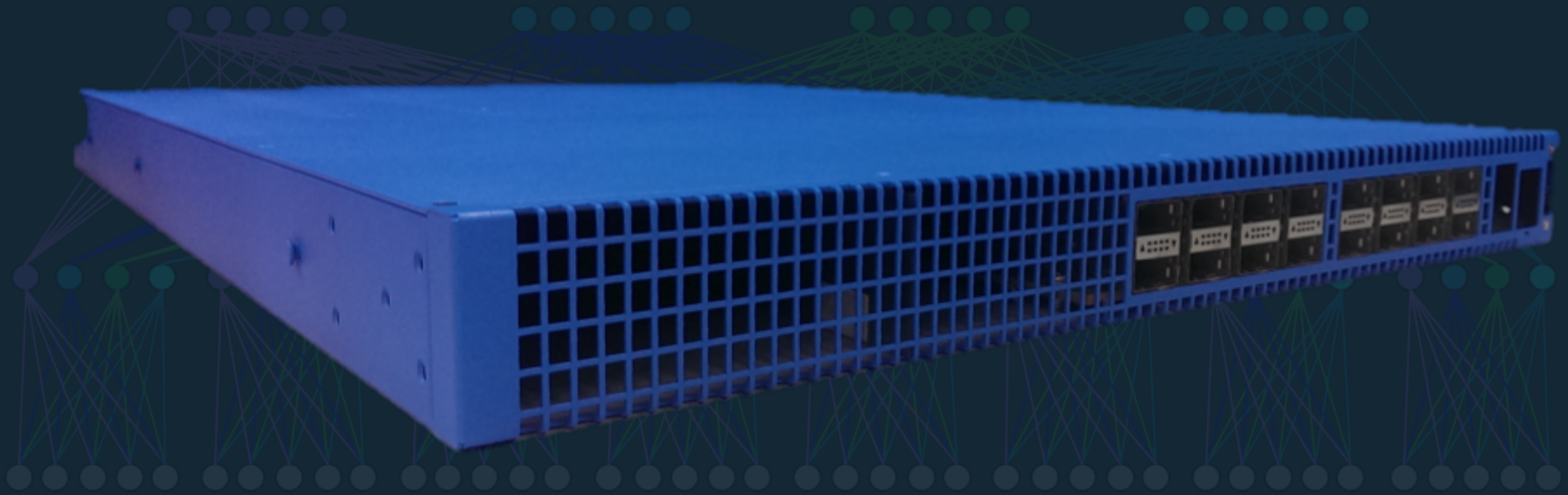
equal performance paths

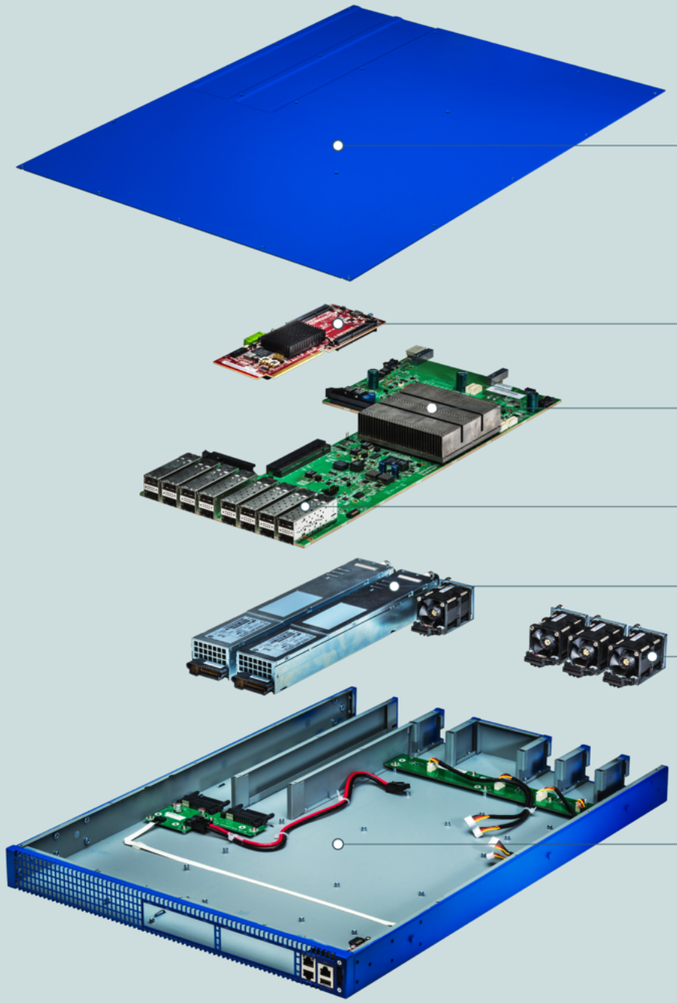
Advantages of Fabric

- Modular/scalable building block
- More bandwidth capacity - future proof
- Distributed load
- Resilient to failures
  - Individual devices and links are not important

The top-of-rack switch

Facebook Wedge





Wedge Hardware Design

Chassis

Open Compute “Group Hug”
Micro Server

40Gb switching ASIC
Commercially available

Sixteen 40Gb network ports
spaced for optimal airflow

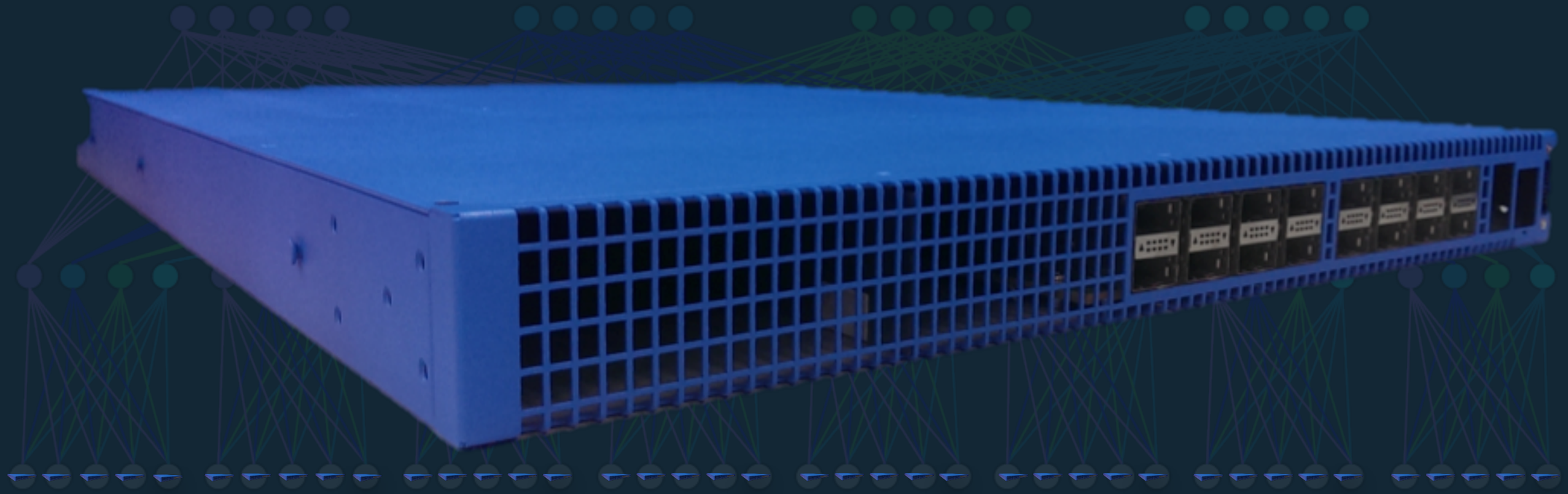
Dual power supplies
with AC and DC options

Fans

Simple enclosure
optimized for efficient cooling

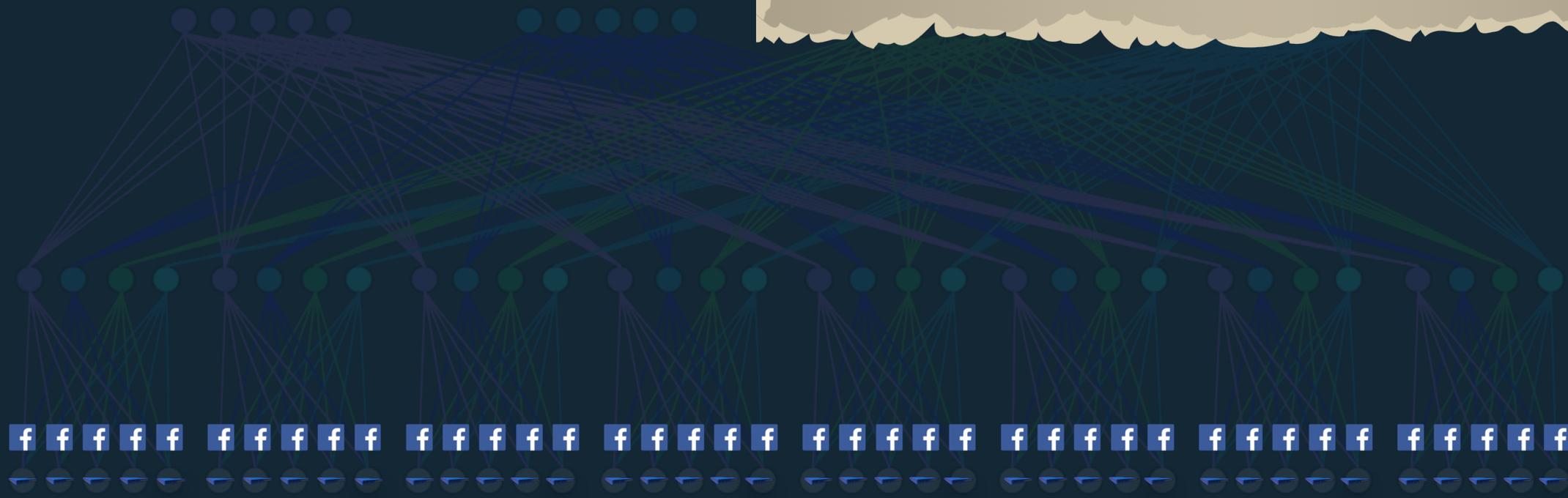
The top-of-rack switch

Facebook Wedge

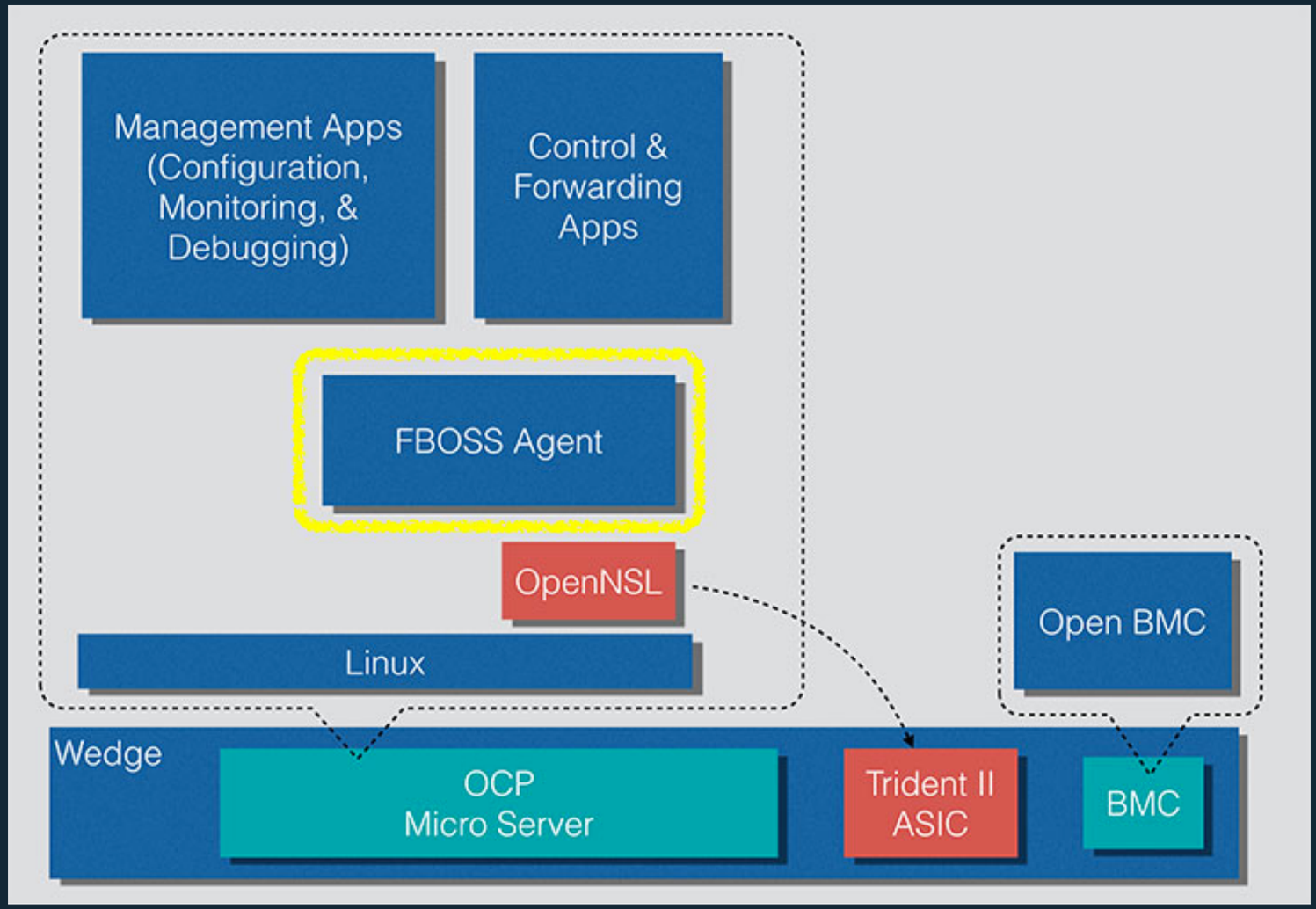


The software

FBOSS:Facebook Open Switching System



FBOSS

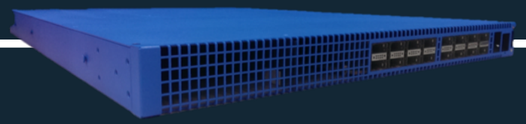

Management Apps (Configuration, Monitoring, & Debugging)
Control & Forwarding Apps
FBOSS Agent
OpenNSL
Linux
Open BMC
Wedge
OCP Micro Server
Trident II ASIC
BMC

6-pack - Core/Spine Switch

6-pack Switch

- First **open** hardware **modular switching** platform
- 128x40GE non-blocking switch
- Runs FBOSS over Linux
- Modular
  - 12 independent Wedges
  - 4 fabric, 8 front-panel
- 100G ready

Data Center Networking Summary



FBOSS
& BMC

OPEN
Compute Project

1	2	3	4	5
From Wedge	We built 6-pack	FBOSS & BMC	OCP based eco system	Open hardware & software

Summary

- Facebook's network infrastructure
  - Datacenters, Edge, CDN, Load-balancers,
  - @Scale Data Center Networking Redefined
  - Software & Hardware
- Open
- Modular
- Ready for experimentation!

For more information…

- https://code.facebook.com/posts/networking
- http://www.opencompute.org/
- https://github.com/facebook
- Email: arunm@fb.com

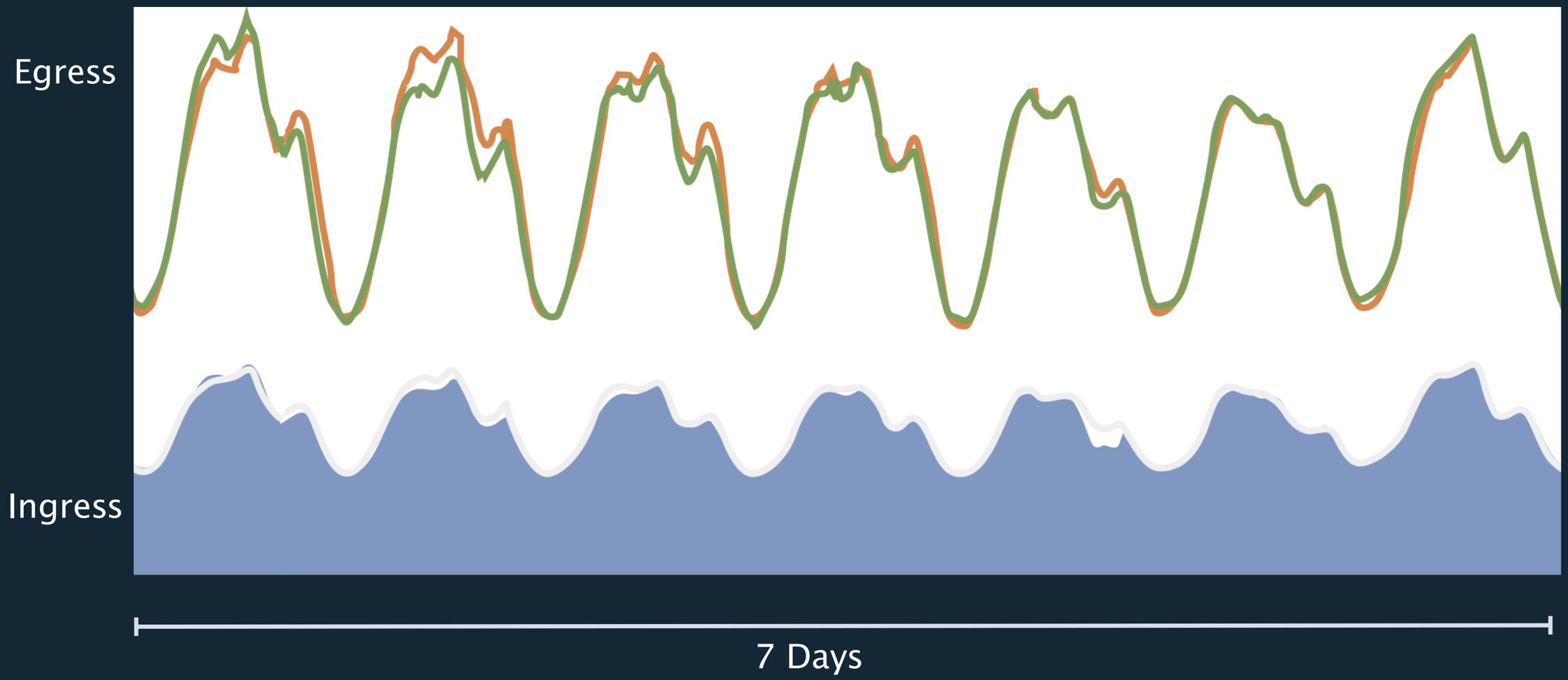

TRUE, OPEN NETWORK SW
ECOSYSTEM


facebook
INFRASTRUCTURE

Backup

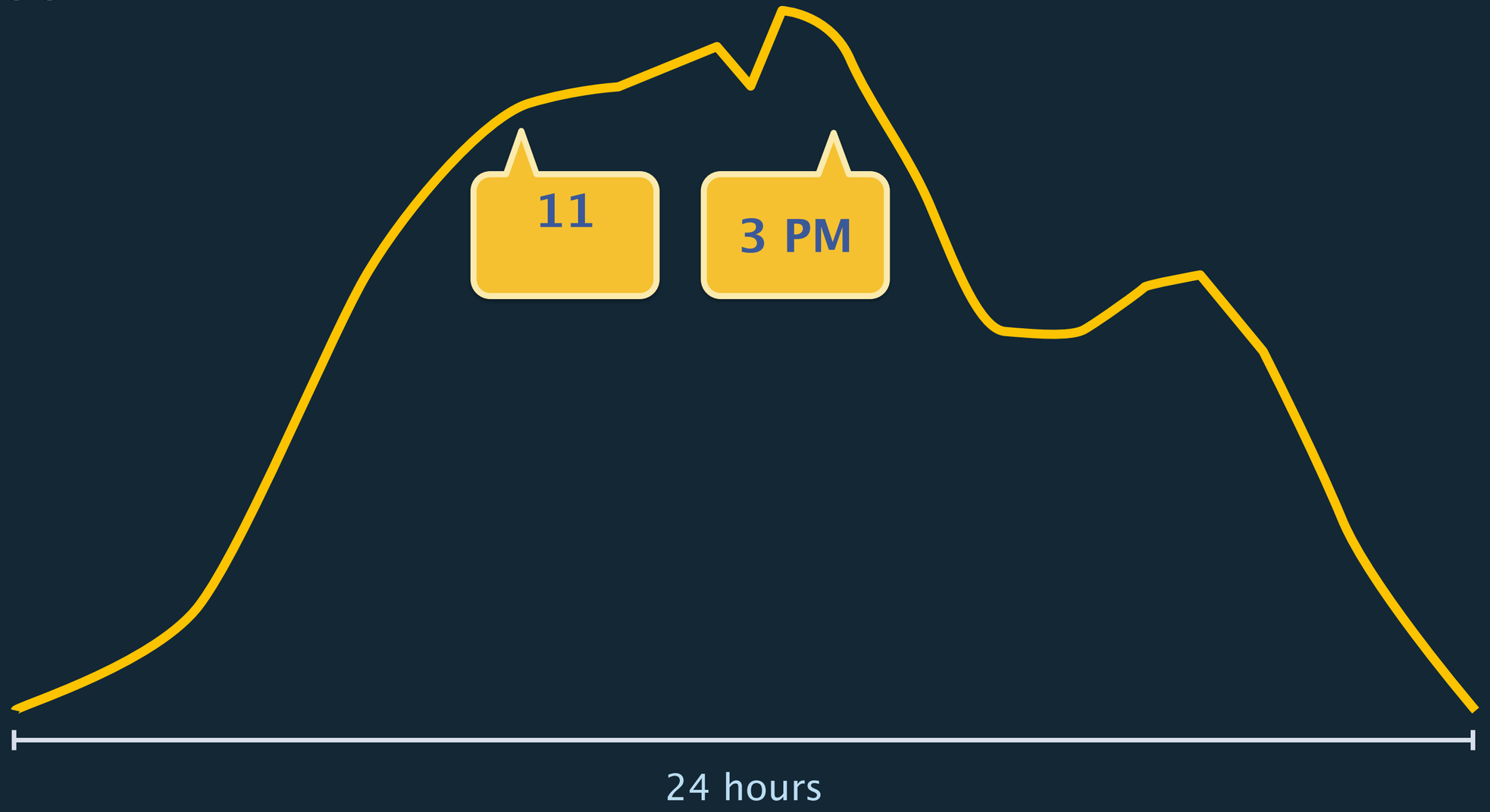
Traffic Characteristics

Weekly Cycle

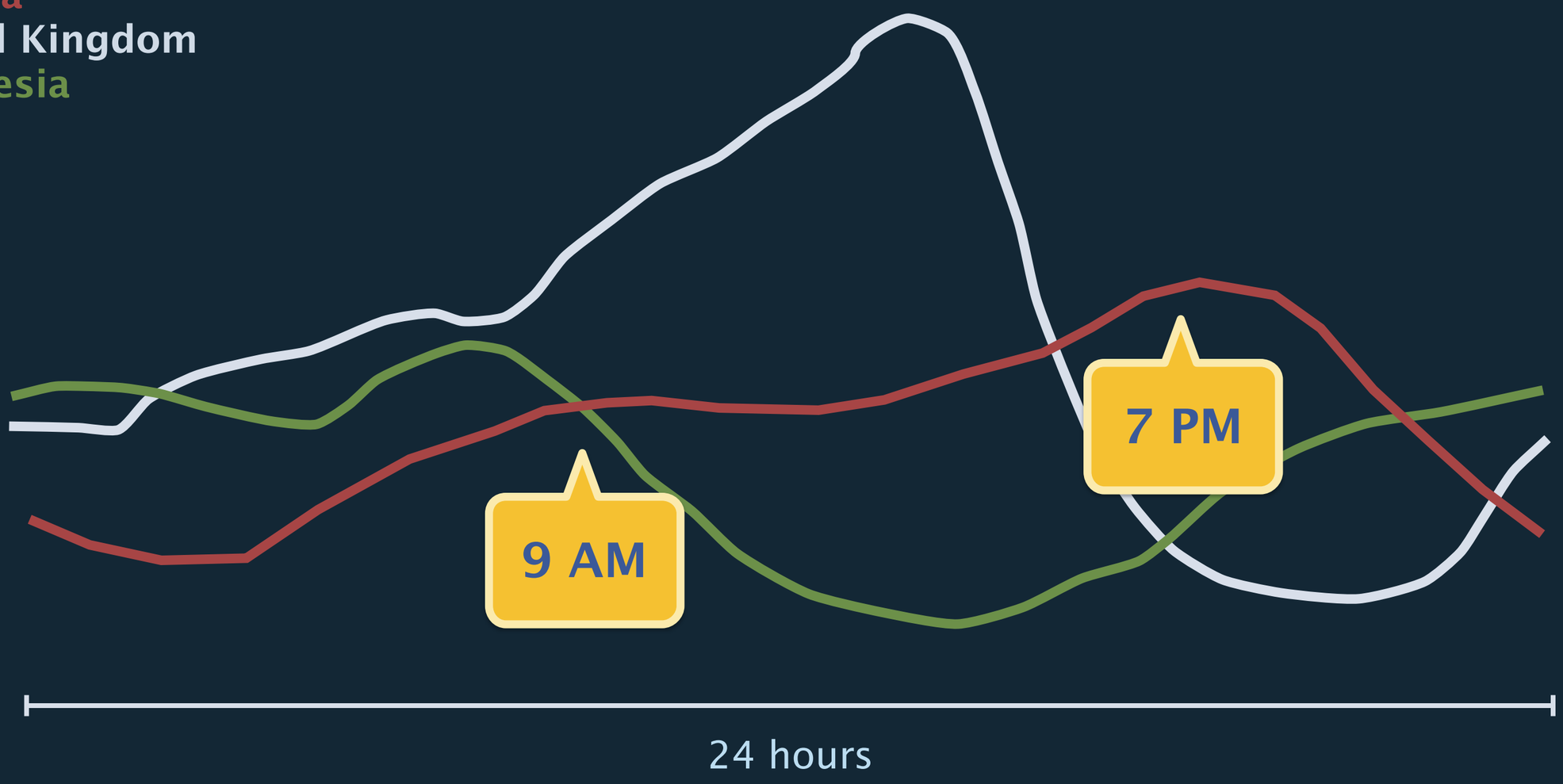


Daily Cycle

Egress



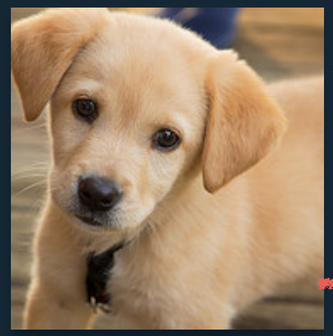
Sum of timezones



Load-balancing: Deep Dive

FB Request -- one web server

rps = requests per second



www?

A 1.2.3.4

DNS

www ?

GET /

<html>...

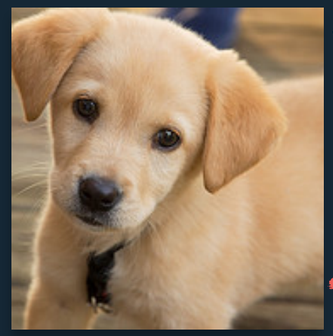
HHVM

low rps

how do we get more rps?!

Add a load balancer!

rps = requests per second



www?

A 1.2.3.4

DNS

GET /

<html>...

L7LB
(proxygen)

GET /

<html>...

www ?

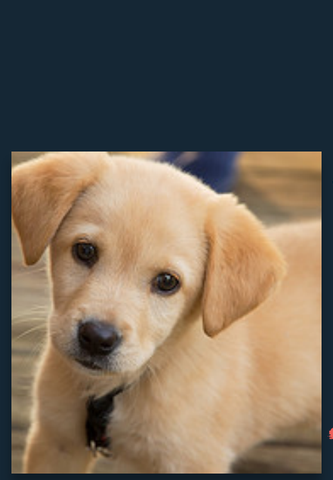
HHVM

lots more rps

low rps

how do we get more rps?!

Add another load balancer!



www?

A 1.2.3.4

DNS

www ?

GET /

<html>...

L4LB (shiv)

GET /

<html>...

L7LB (proxygen

GET /

<html>...

HHVM

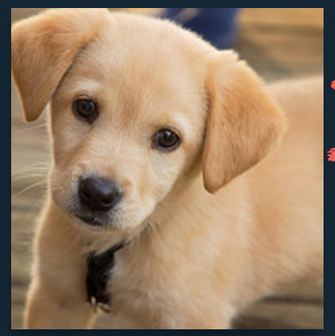
network bound

lots more rps

low rps

how do we get more rps?!

Add another load balancer!



www?

A 1.2.3.4

DNS

ECMP

GET /

<html>

L4LB (shiv)

network bound

GET /

<html>...

L7LB (proxygen

lots more rps

GET /

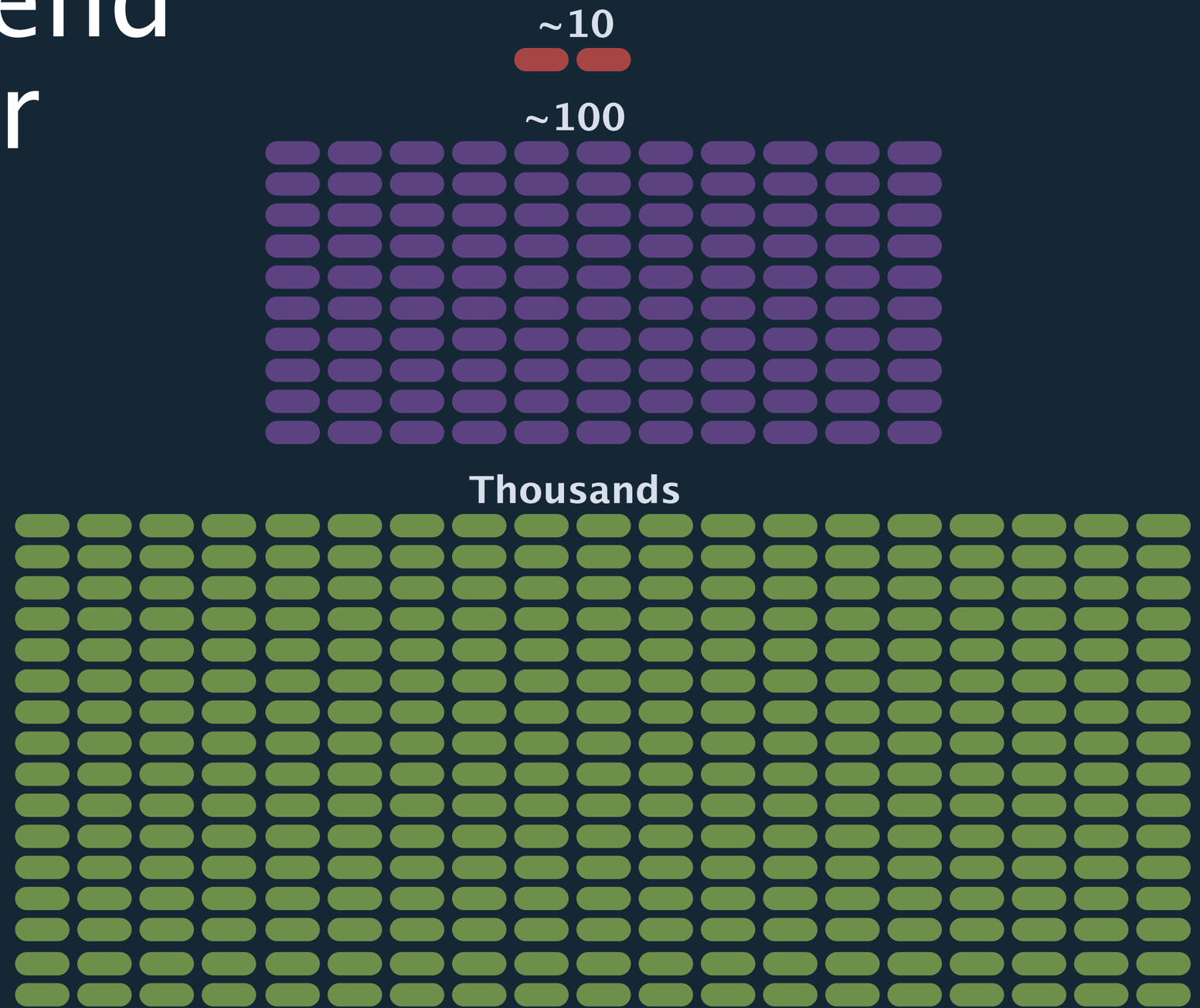
<html>...

HHVM

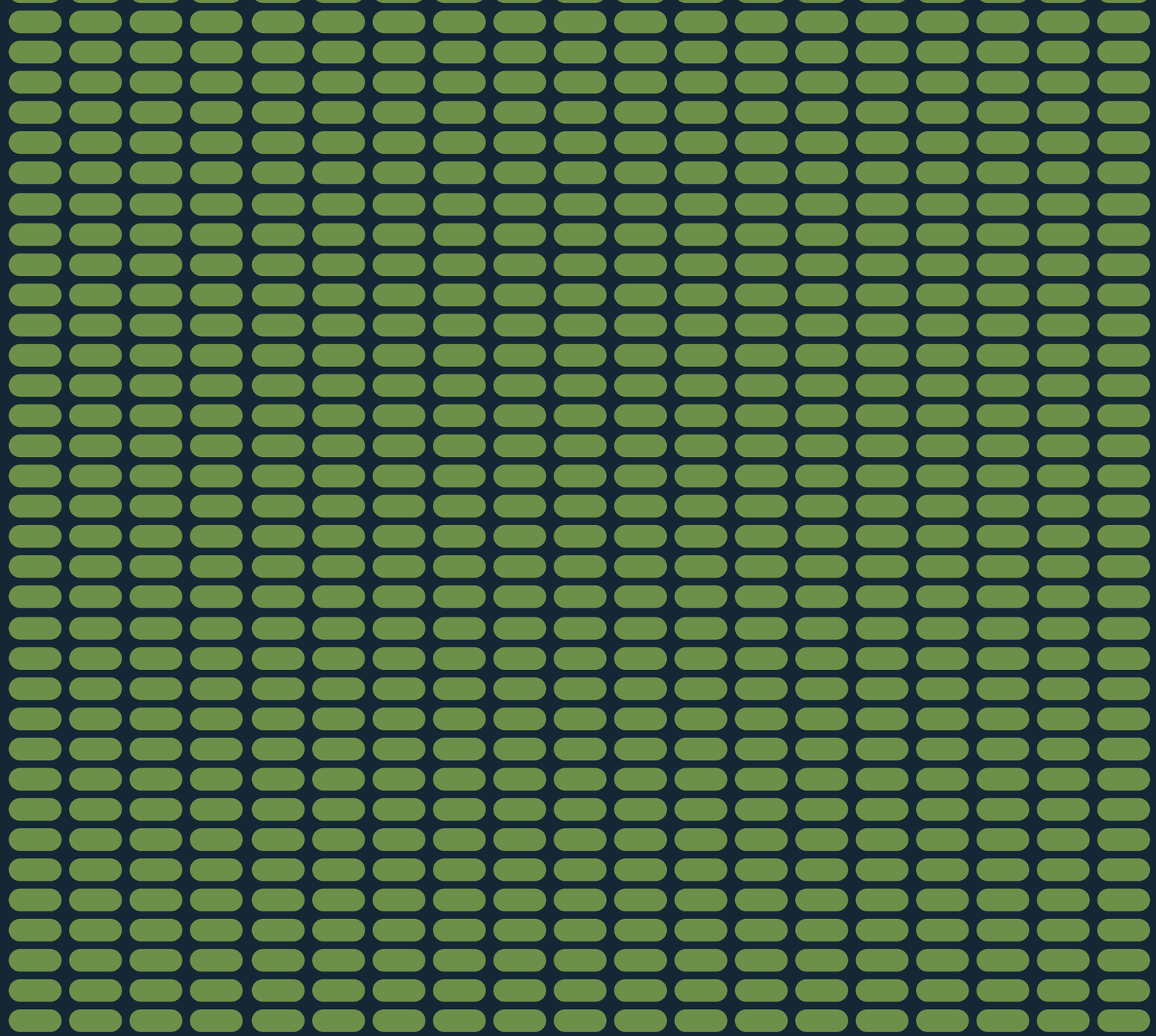
HHVM

low rps

Front end Cluster

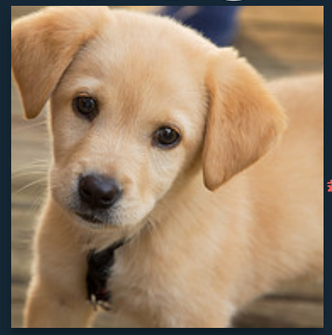


cont.



x 10 or more

More RPS? Add another cluster!



www?

A 1.2.3.4

DNS

GET /

<html>

L4LB (shiv)

GET /

<html>...

L7LB (proxygen

GET /

<html>...

HHVM

GET /

<html>

L4LB (shiv)

GET /

<html>...

L7LB (proxygen

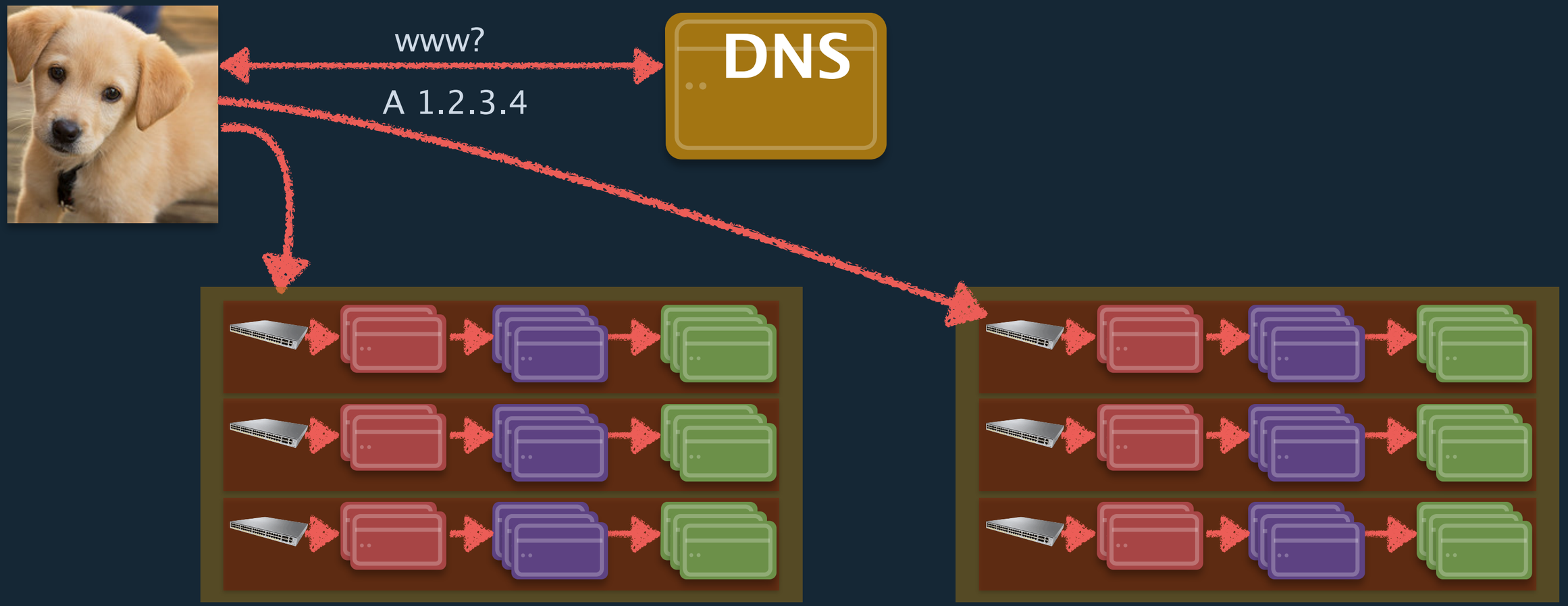
GET /

<html>...

HHVM

how do we get more rps?!

Add another datacenter!



L4LB

BGP

ECMP Hash

State Table + Hash

1.2.3.4

L4LB

1.2.3.4

L4LB

1.2.3.4

L7LB

1.2.3.4

L7LB

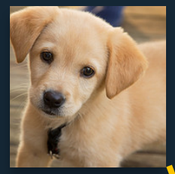
1.2.3.4

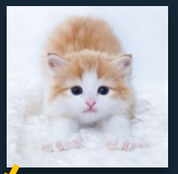
L7LB

1.2.3.4

L7LB

L4LB Routing





ECMP Hash

State Table + Hash

A

1.2.3.4

L4LB

1.2.3.4

L4LB

D

1.2.3.4

L7LB

1.2.3.4

L7LB

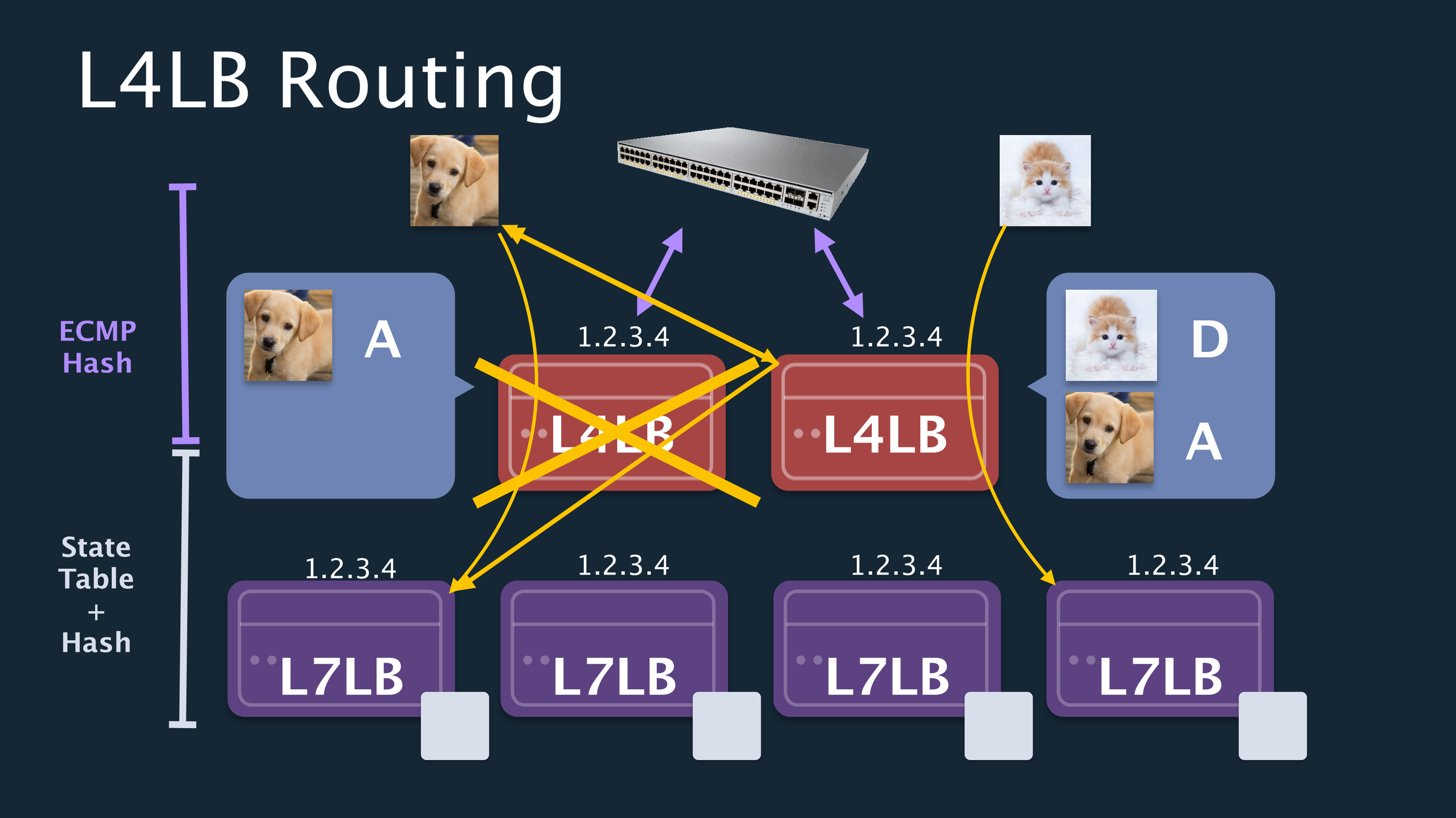
1.2.3.4

L7LB

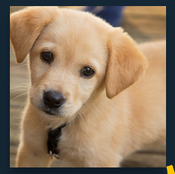
1.2.3.4

L7LB

L4LB Routing



L4LB Routing



ECMP Hash

State Table + Hash

B

1.2.3.4

L4LB

1.2.3.4

L4LB

1.2.3.4

1.2.3.4

L7LB

1.2.3.4

L7LB

1.2.3.4

L7LB

L4LB Routing

ECMP Hash

State Table + Hash

B

1.2.3.4

L4LB

1.2.3.4

L4LB

1.2.3.4

L7LB

1.2.3.4

L7LB

1.2.3.4

L7LB

1.2.3.4

L7LB