# Integrating Network Management for Cloud Computing Services

Final Public Oral Exam
Peng Sun

# What is Cloud Computing

- Deliver applications as services over Internet
- Run applications in datacenters
- *Lower capital and operational expenses*

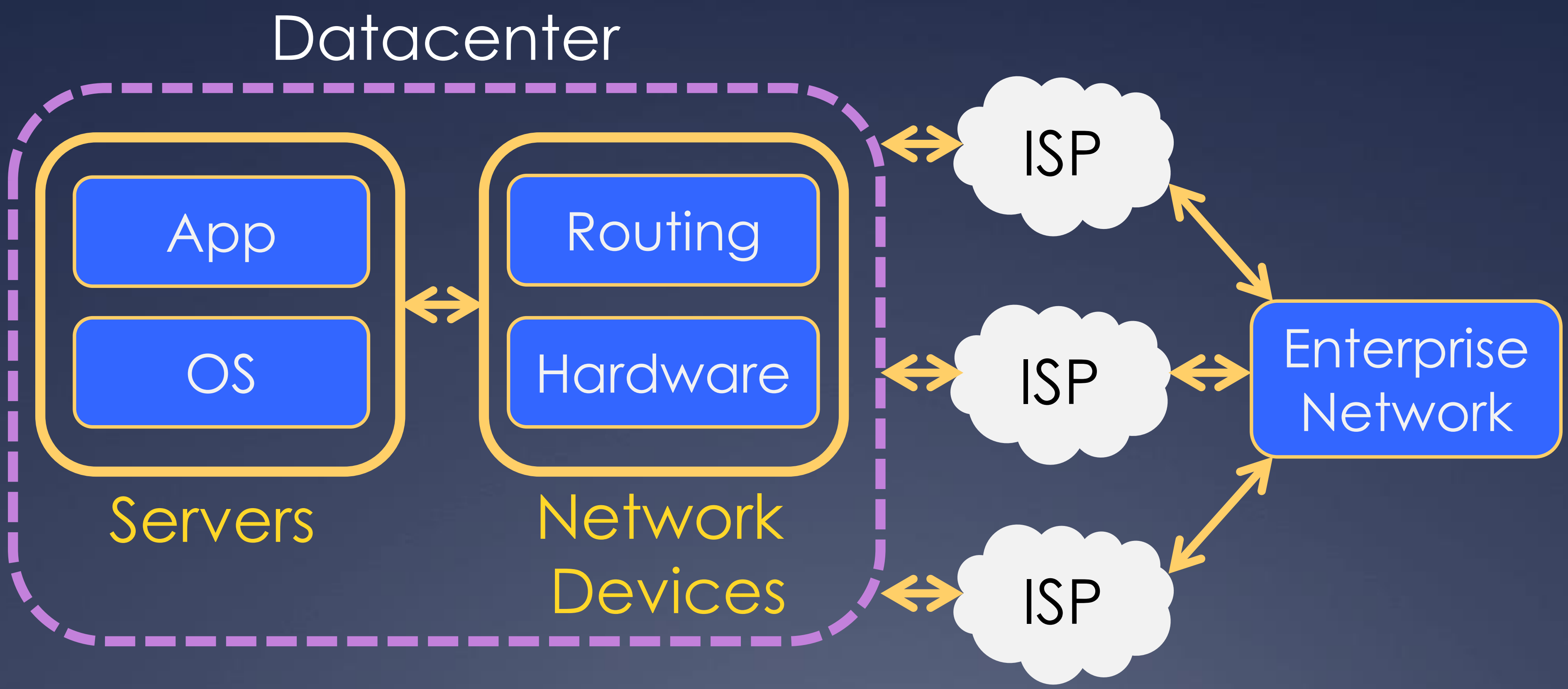
# Cloud Services are Growing

- Public cloud for consumer service
  - Amazon Web Services
  - Microsoft Azure
  - …

# Cloud Services are Growing

- Public cloud for consumer service
  - Amazon Web Services
  - Microsoft Azure
  - …
- Enterprises out-source IT service
  - Storage: Box, …
  - Analytics: Salesforce, …
  - Productivity: Office365, …

# Quality of Cloud Service Depends on Network Quality



# Improve Network Quality

- The old way cannot keep up with the growth of cloud services
  - *Deploying more devices with higher bandwidth*

# Improve Network Quality

- The old way cannot keep up with the growth of cloud services
  - *Deploying more devices with higher bandwidth*
- The key is proper management of network resources

# Examples of Network Management Solutions

# Examples of Network Management Solutions

Traffic Engineering

# Examples of Network Management Solutions

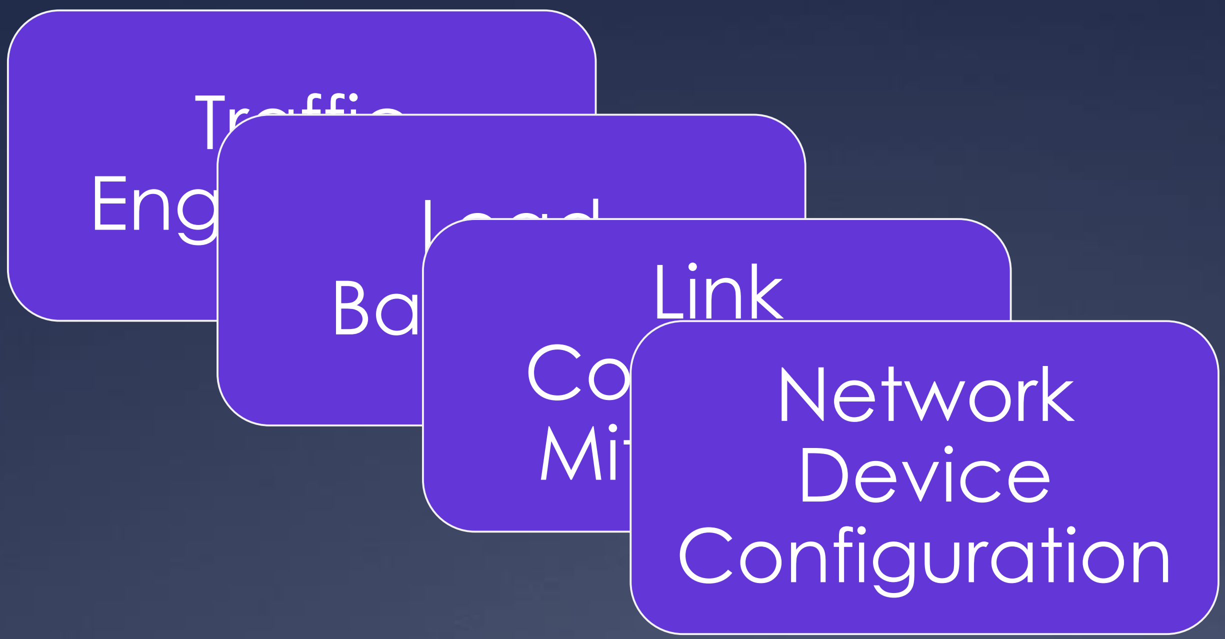
Traffic
Eng

Load Balancing

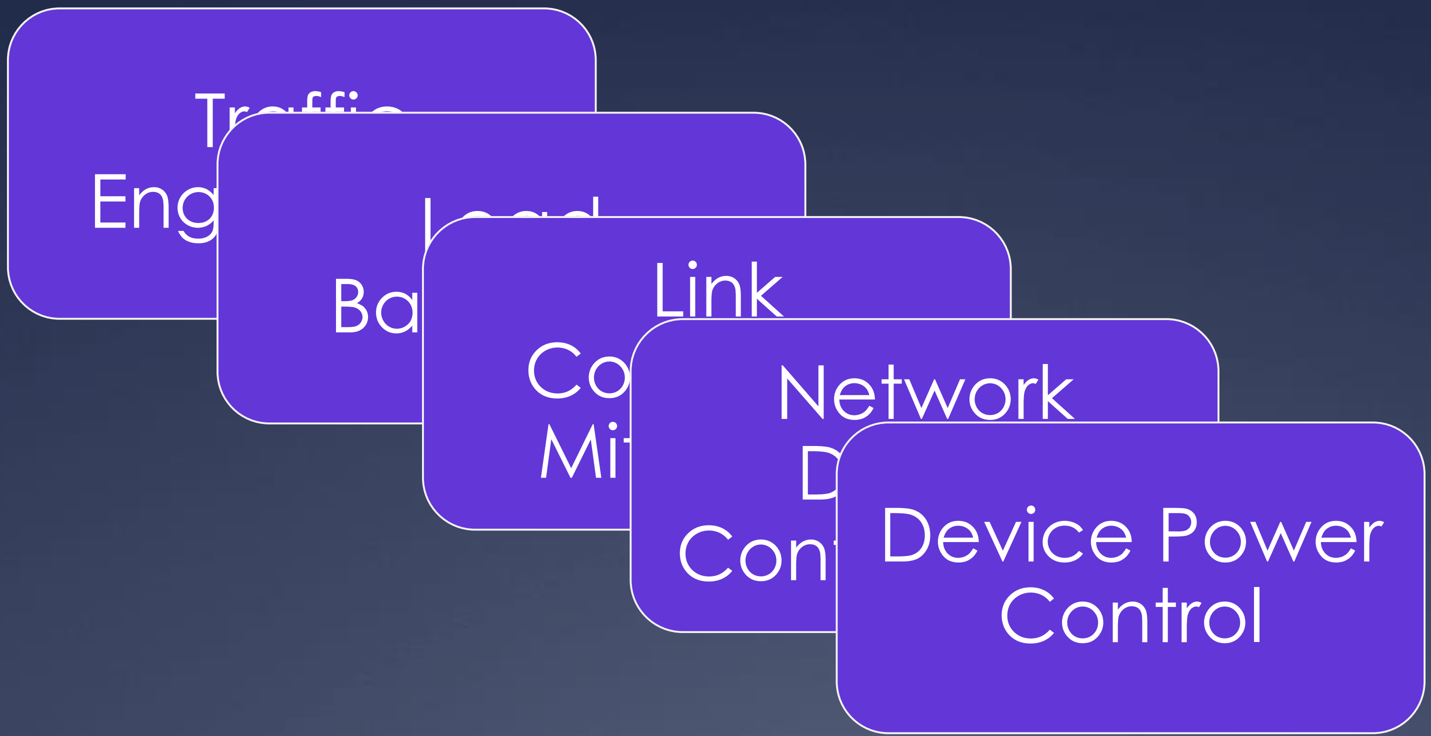
# Examples of Network Management Solutions

- Traffic Eng
- Load Ba
- Link Corruption Mitigation

# Examples of Network Management Solutions



# Examples of Network Management Solutions



# Examples of Network Management Solutions

- Traffic Eng
- Load Ba
- Link Co Mi
- Network D Con
- Device Power Control

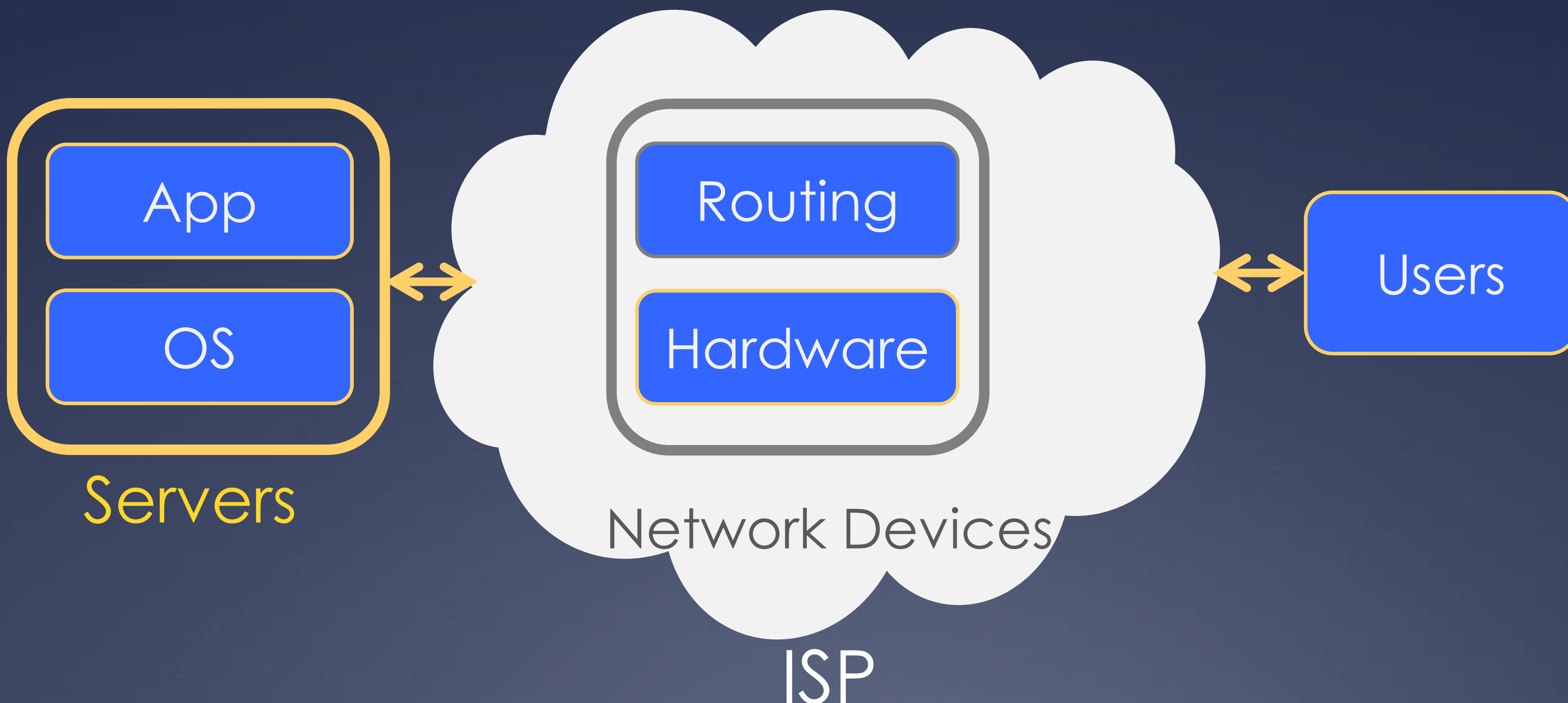
● ● ● ● ● ●

# Problems of Current Network Management

- Disjoint management of network components
- Low-level interfaces with network devices

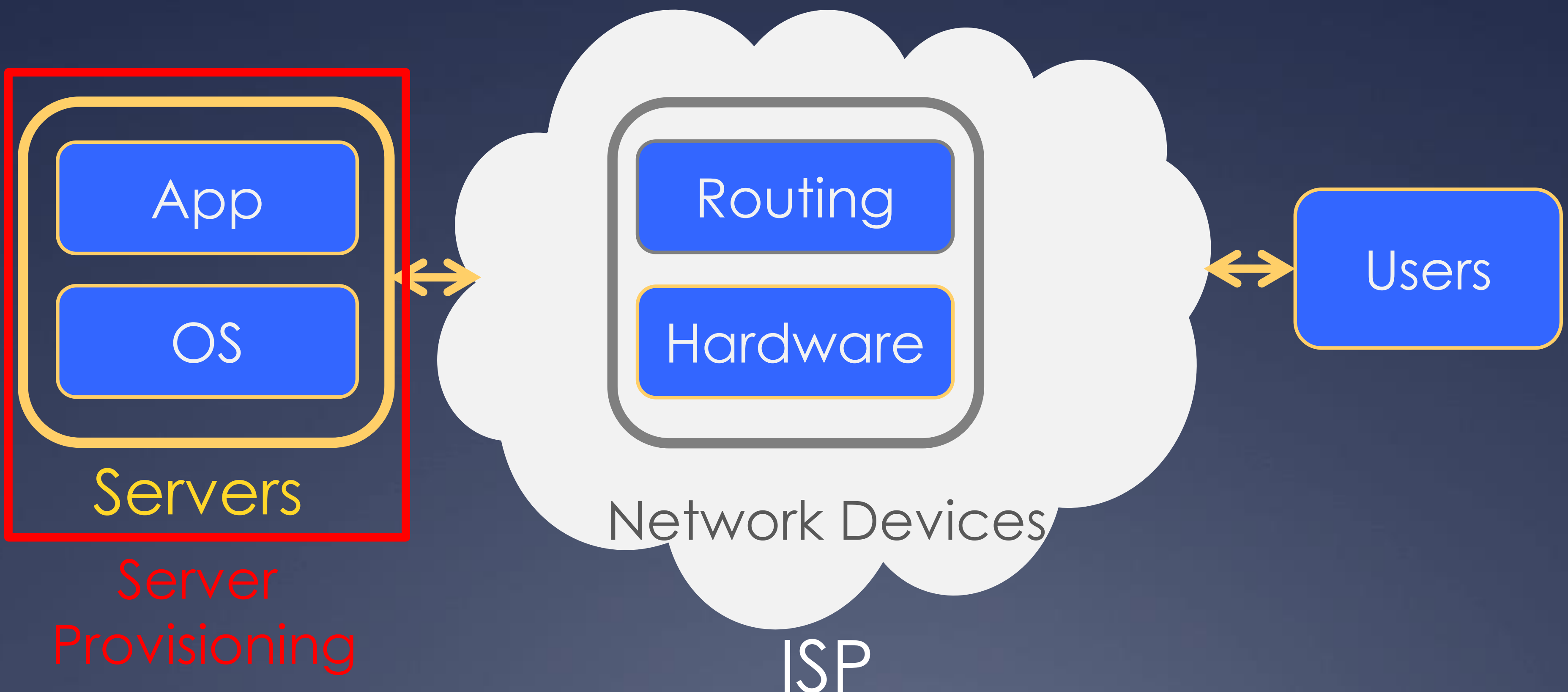
# Disjoint Management Systems

- Division of labor in pre-cloud era



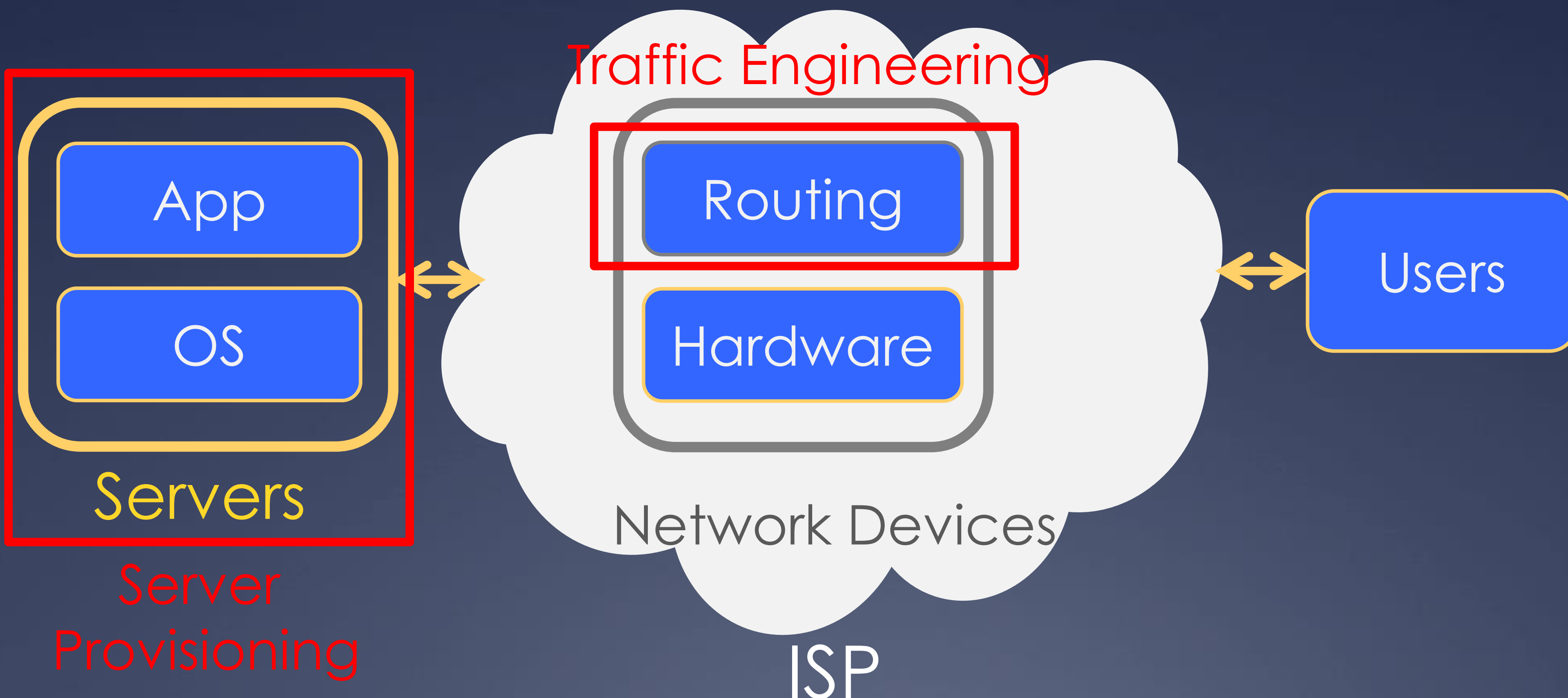
# Disjoint Management Systems

- Division of labor in pre-cloud era



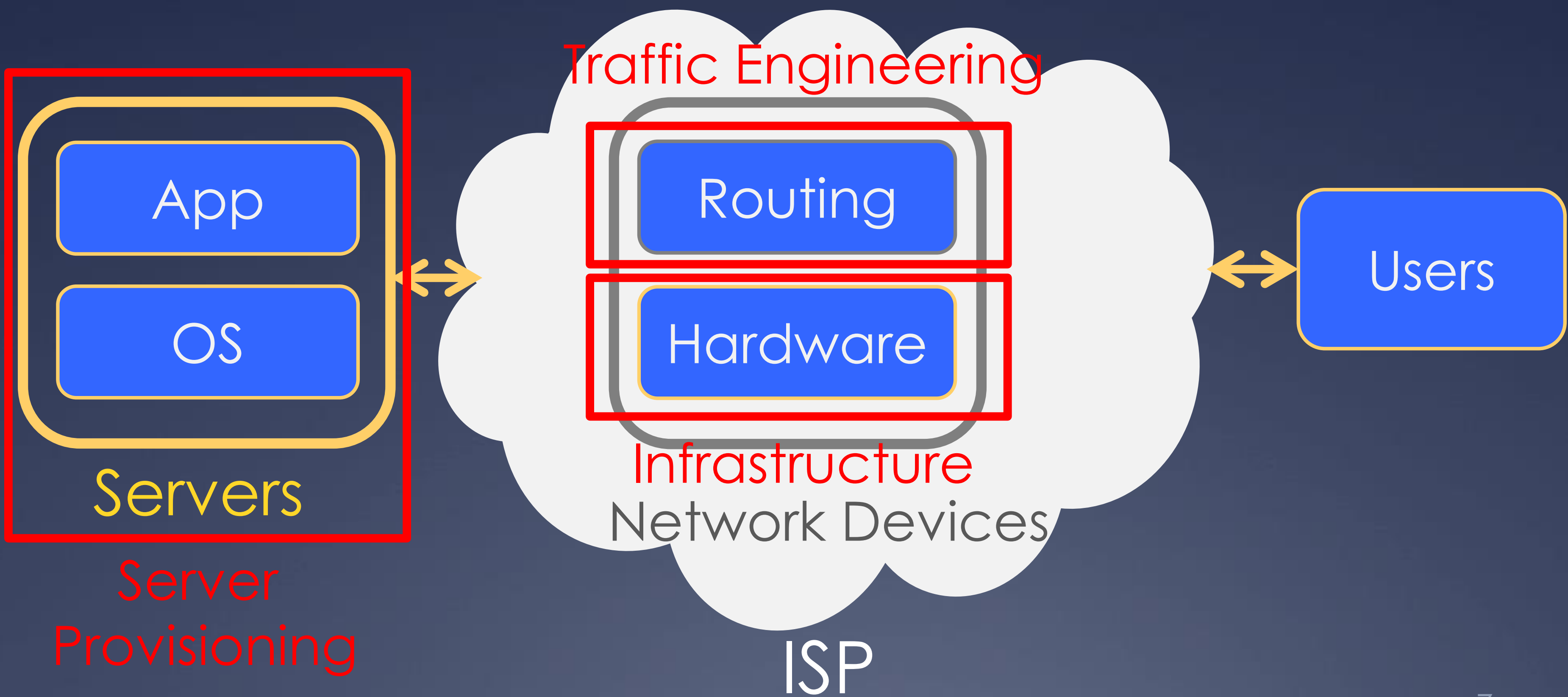
# Disjoint Management Systems

- Division of labor in pre-cloud era



# Disjoint Management Systems

- Division of labor in pre-cloud era



# Datacenter Breaks Balance

- Coordination of server & network:
  - More applications are built as multi-tier distributed systems
  - Intra-datacenter traffic is new majority

# Datacenter Breaks Balance

- Coordination of server & network:
  - More applications are built as multi-tier distributed systems
  - Intra-datacenter traffic is new majority
- Infrastructure evolution speeds up:
  - Network architecture changes (e.g., FatTree)

# Disjoint Mgmt. is Bottleneck

- Because:
  - Cloud service providers have stakes in all three management areas:
    - *Server, infrastructure, traffic*

# Disjoint Mgmt. is Bottleneck

- Because:
  - Cloud service providers have stakes in all three management areas:
    - *Server, infrastructure, traffic*
  - Great opportunity exists for consolidation
    - *Google and Microsoft on SoftWAN*

# Yet Another Problem: Low-level Device Interaction

- Hardware vendors differentiate with specialized devices
- Network operation:
  - intensively uses vendor-specific APIs
  - heavily depends on experiences

# Cloud Service is Different

- Much more devices in datacenters than traditional networks
  - *Automation is a must*

# Cloud Service is Different

- Much more devices in datacenters than traditional networks
  - *Automation is a must*
- Commodity devices instead of specialized hardware
  - *Homogeneity is preferred*

# Cloud Service is Different

- Much more devices in datacenters than traditional networks
  - *Automation is a must*
- Commodity devices instead of specialized hardware
  - *Homogeneity is preferred*
- *Lower vendor dependence pays off (MS & Amazon on SoftLB)*

# Promising Yet Limited SDN

- Great way of automating traffic management with high-level programming paradigms

- Yet literature focus on just ‘traffic’

# Summary of Problems

- This dissertation solves:
- *Disjoint management of server, infrastructure, and traffic*
- *Low-level device interaction for broader scope of network management*

# Practical Approach

# Practical Approach

Identify opportunity of integration

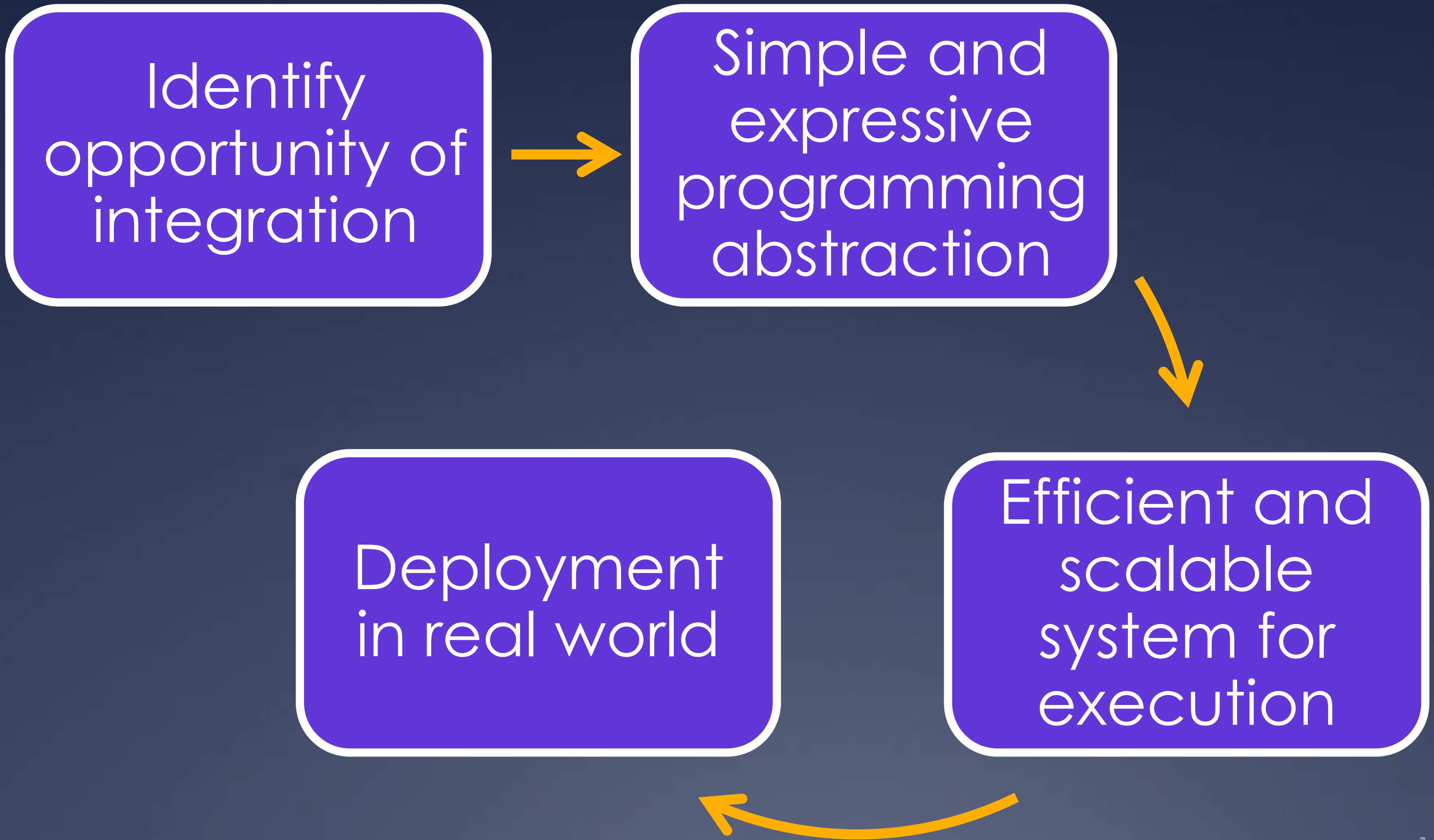
# Practical Approach

Identify opportunity of integration → Simple and expressive programming abstraction

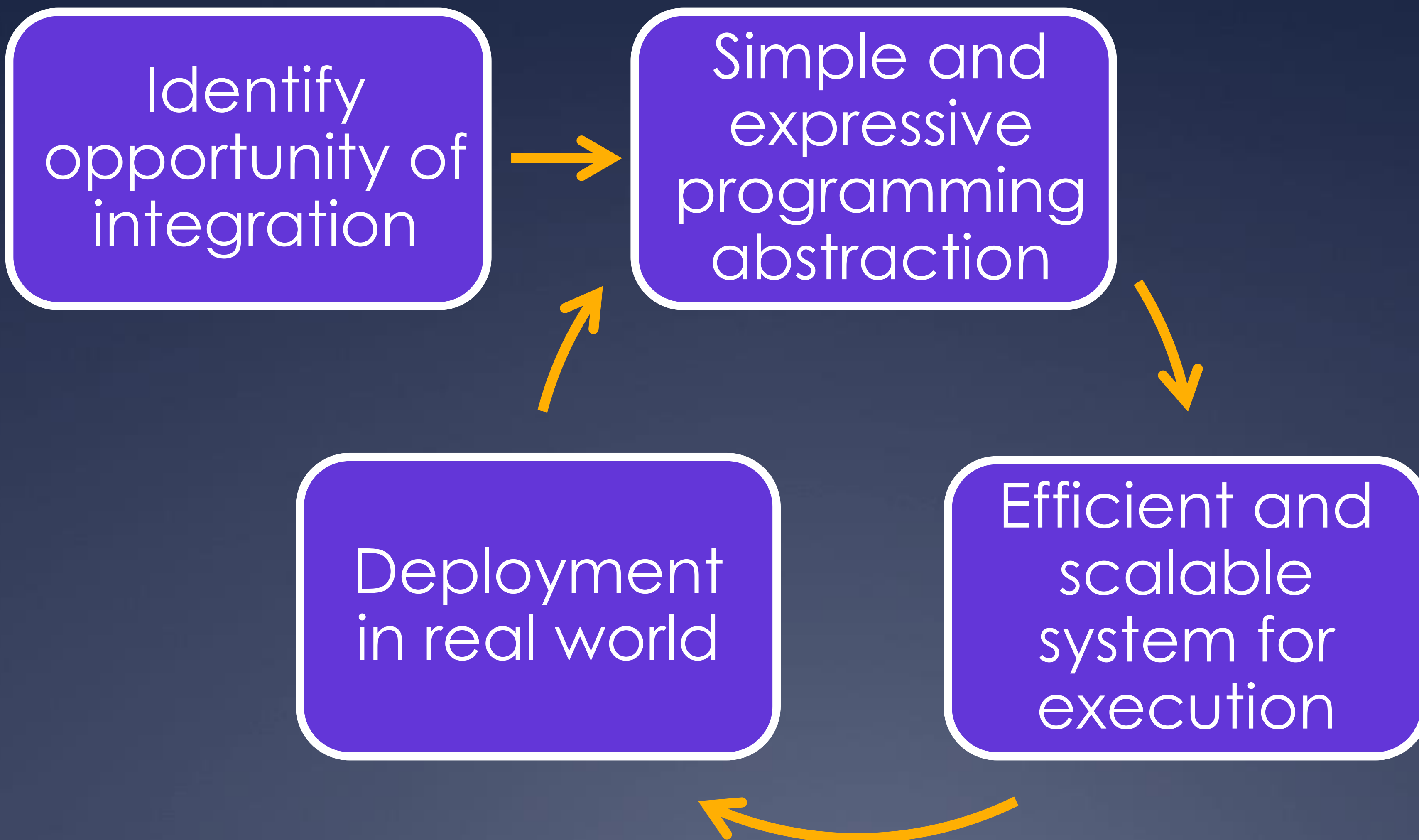
# Practical Approach

Identify opportunity of integration → Simple and expressive programming abstraction → Efficient and scalable system for execution

# Practical Approach



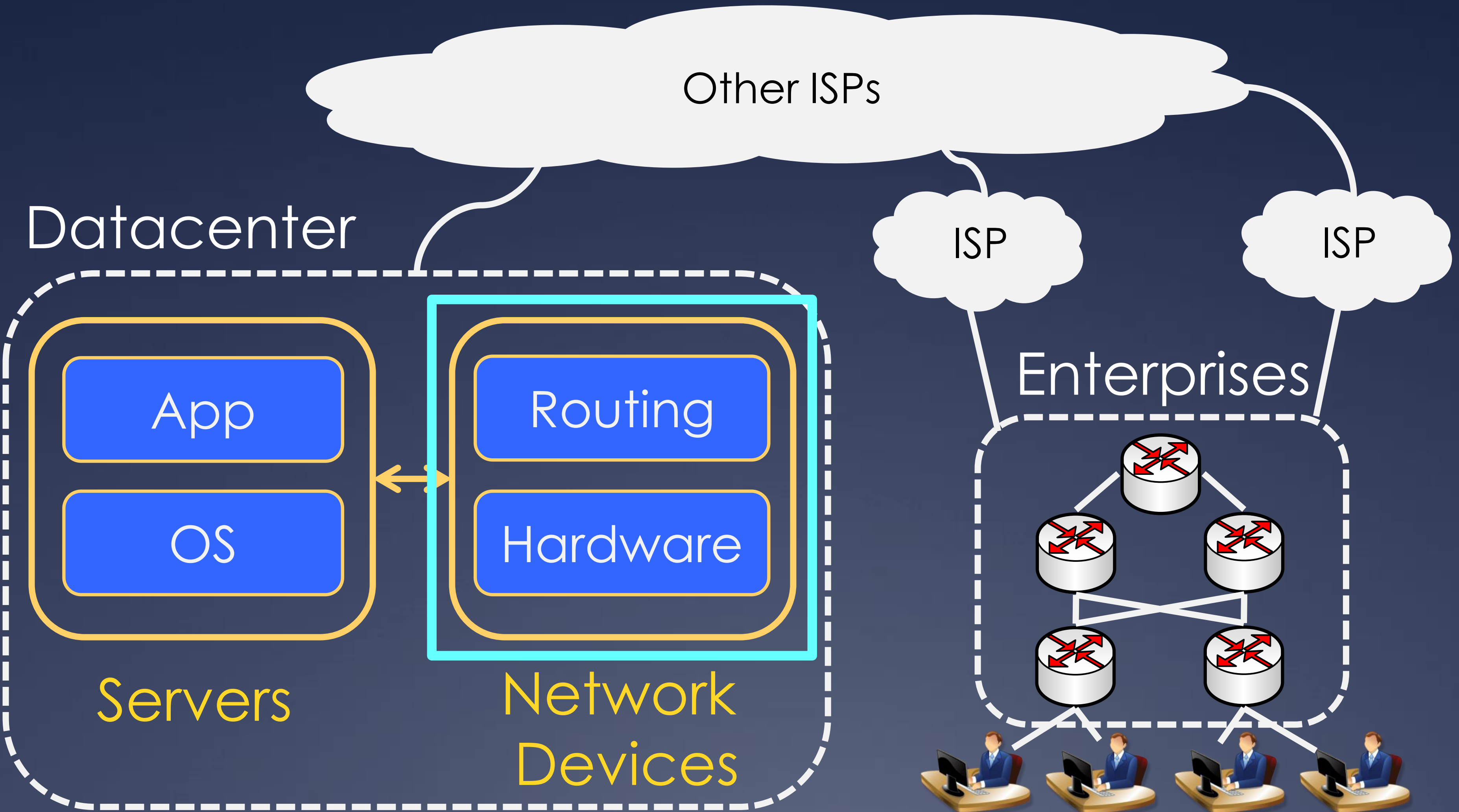
# Practical Approach



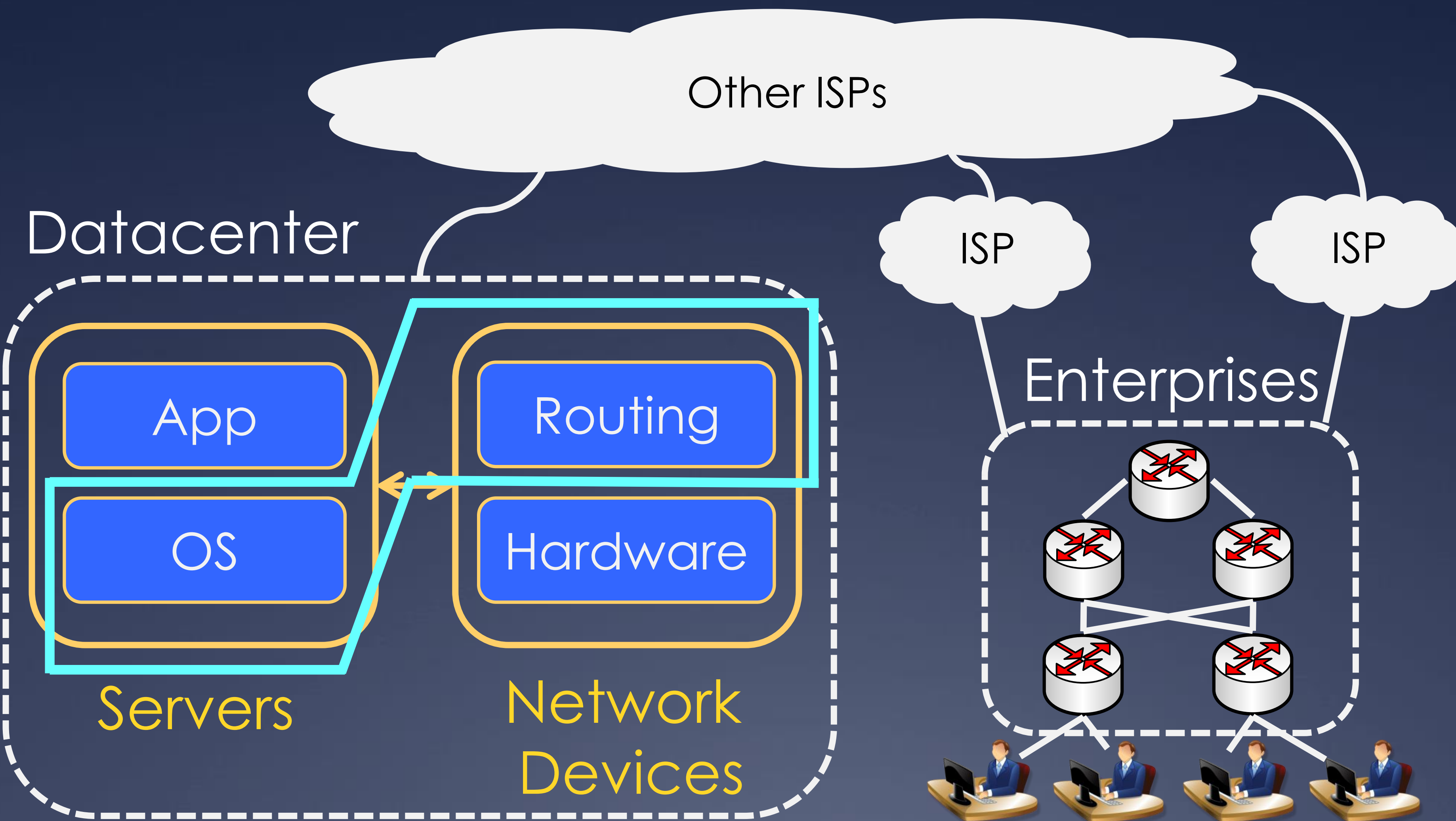
# Contributions

| Project | Publication | Deployment |
|---|---|---|
| Statesman | SIGCOMM'14 | Microsoft Azure |
| HONE | JNSM, Vol. 23, 2015 | Overture/ Verizon |
| Sprite | SOSR'15 | In process with OIT |

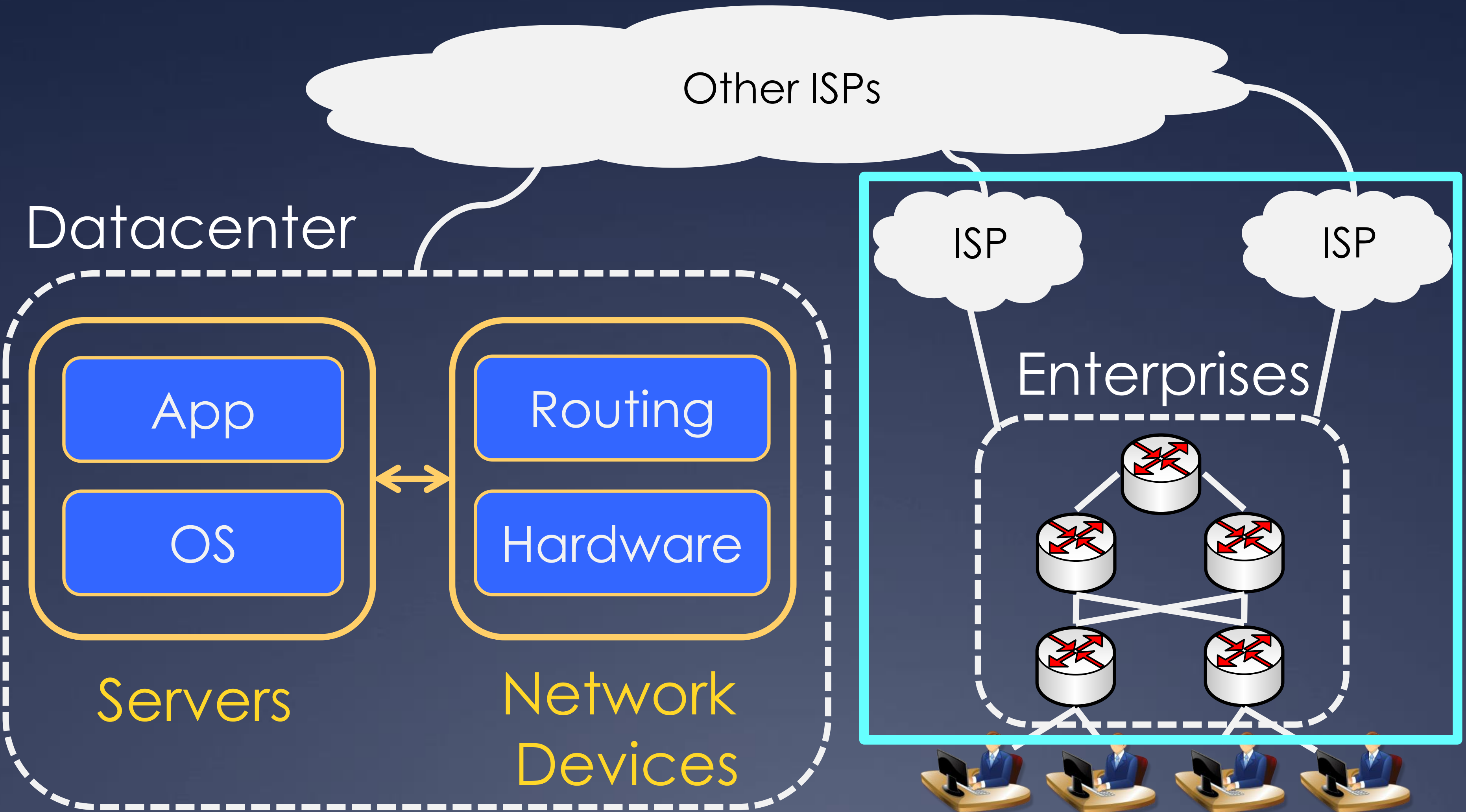
# Statesman: Safe Datacenter Traffic/Infrastructure Management



# Hone: End-host/Network Cooperative Traffic Management



# Sprite: Direct Control of Entrant ISP for Enterprise Traffic



# What Follows

- Brief explanation of each project
- Open issues
- Related work
- Q&A

# Statesman:

# Integrating Network Infrastructure Management

# Problem for Cloud Providers

- Multiple mgmt. solutions coexist
  - *for traffic and infrastructure mgmt.*

# Problem for Cloud Providers

- Multiple mgmt. solutions coexist
  - *for traffic and infrastructure mgmt.*
- Complexity is the main problem

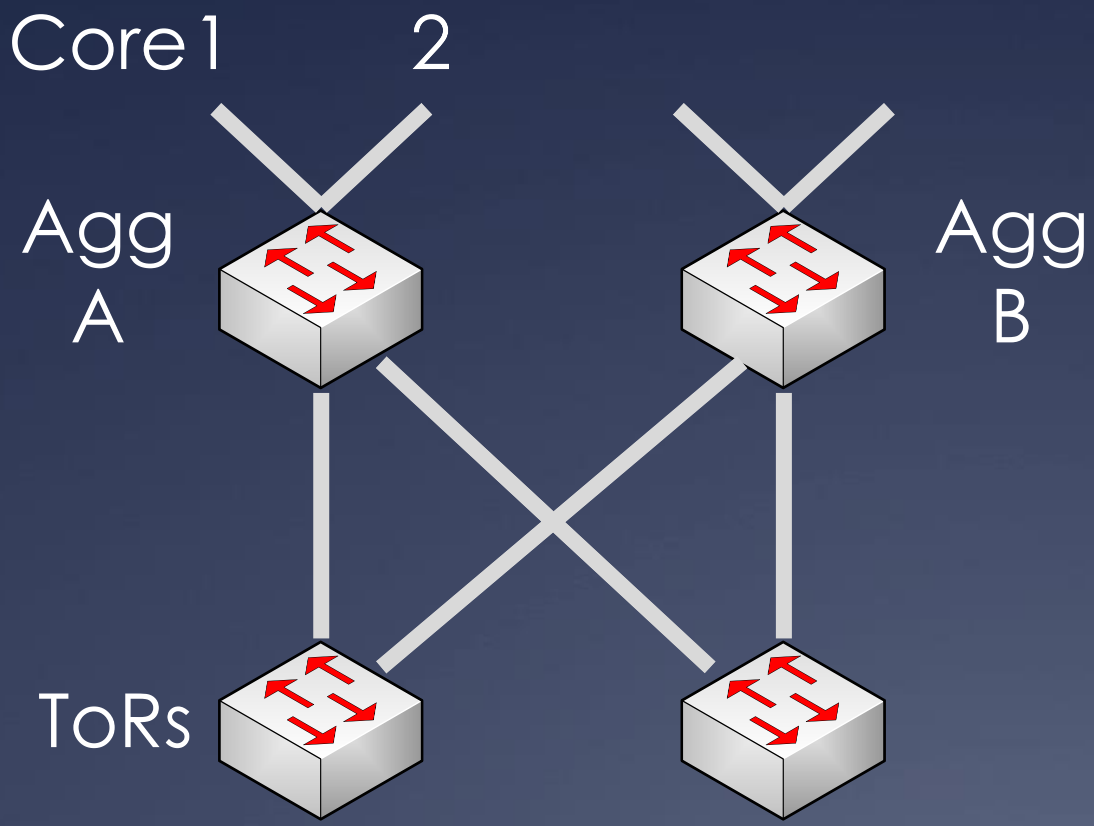
# Problem for Cloud Providers

- Multiple mgmt. solutions coexist
  - *for traffic and infrastructure mgmt.*
- Complexity is the main problem
  - Development
    - *Scale & heterogeneity of devices*

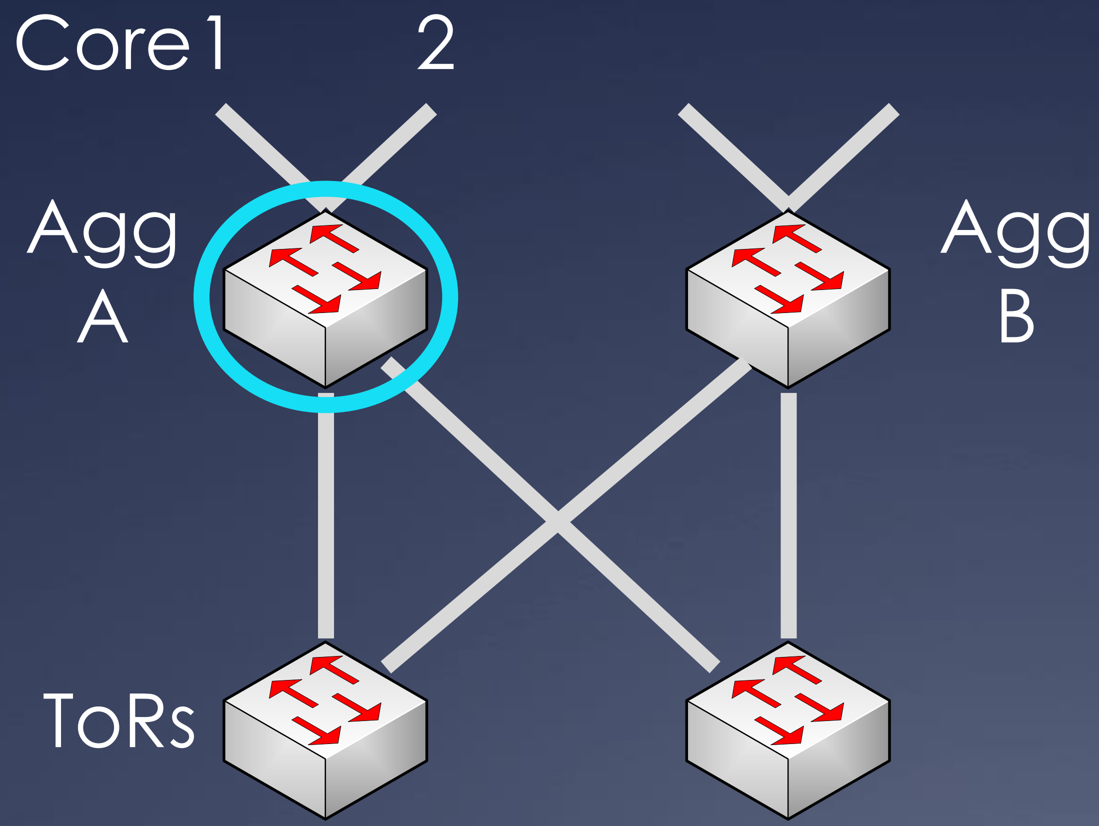
# Problem for Cloud Providers

- Multiple mgmt. solutions coexist
  - *for traffic and infrastructure mgmt.*
- Complexity is the main problem
  - Development
    - *Scale & heterogeneity of devices*
  - Coordination
    - *Conflicts and safety violations*

# Conflict

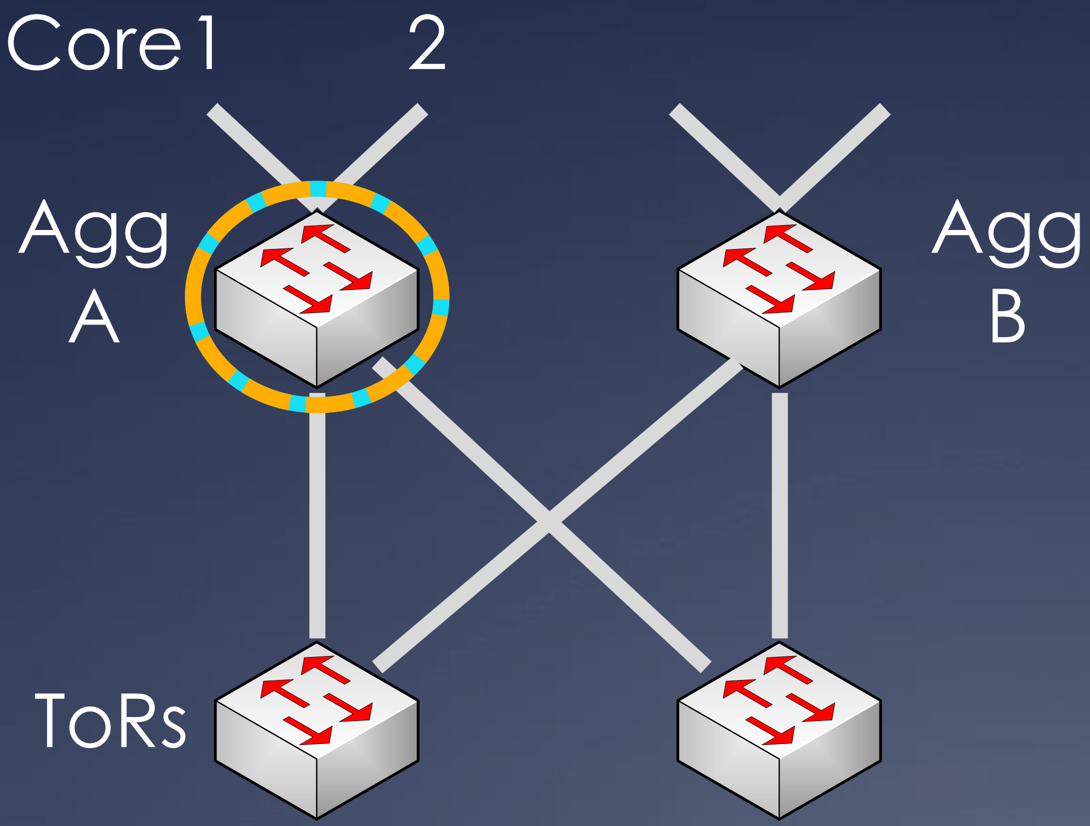


# Conflict




Link-corruption-mitigation adjusts traffic away from Core1

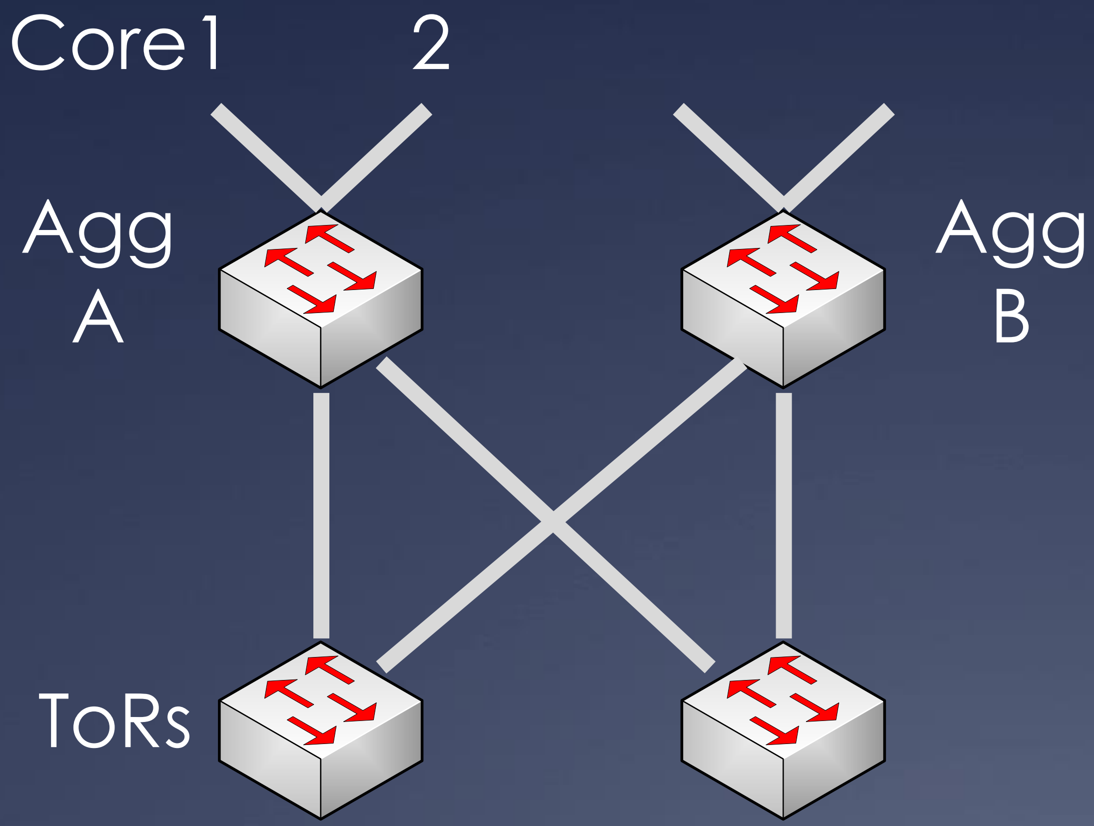
# Conflict



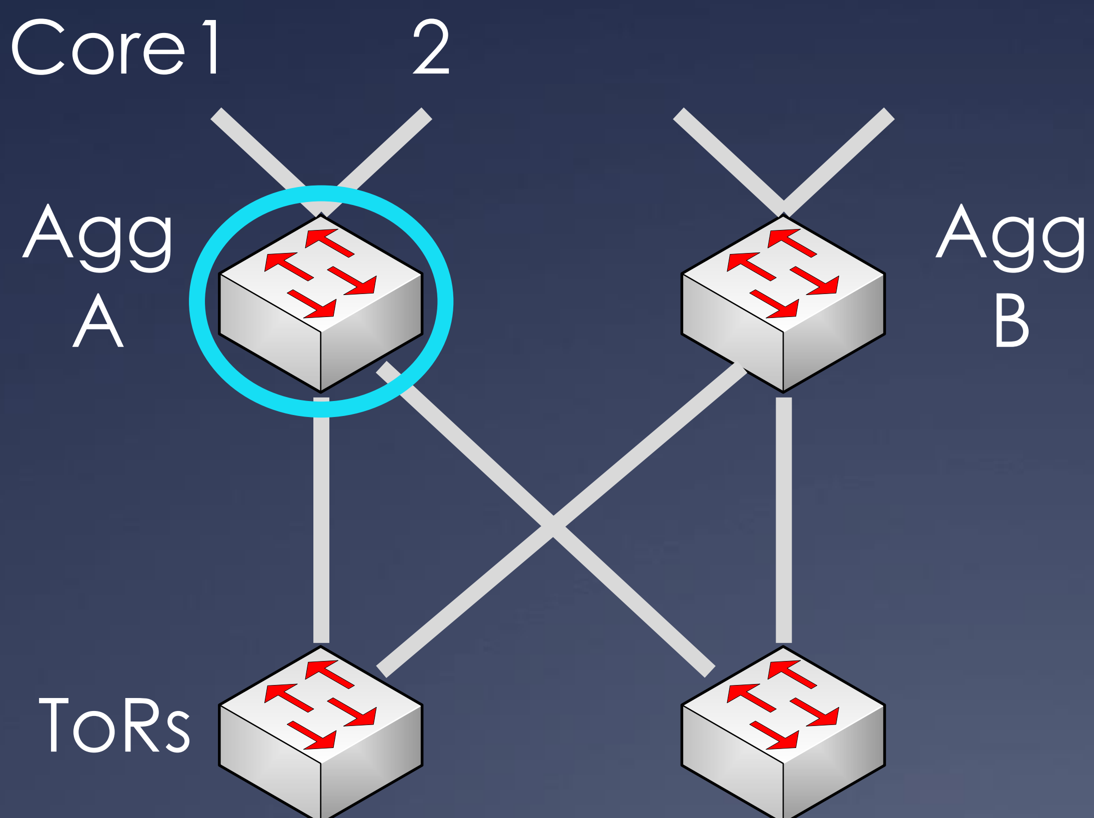

Link-corruption-mitigation adjusts traffic away from Core1

TE tunes traffic among links to Core1, 2

# Safety Violation

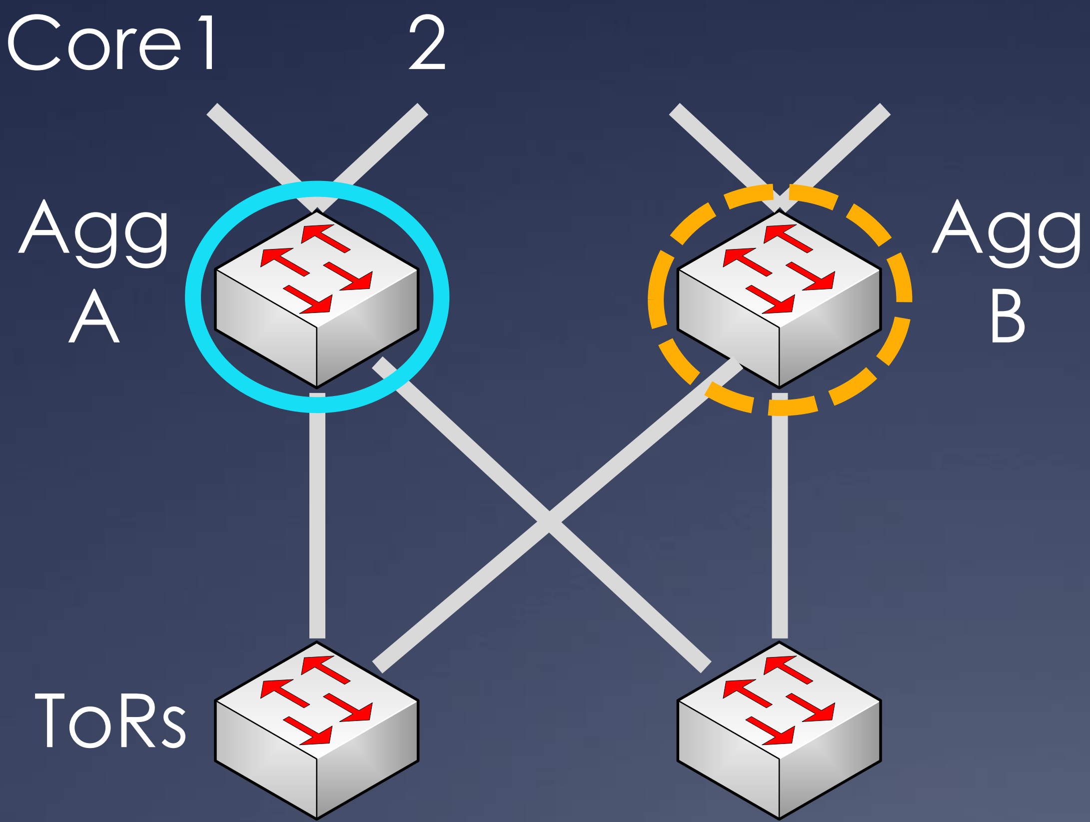


# Safety Violation




Link-corruption-mitigation shuts down faulty Agg A

# Safety Violation




Link-corruption-mitigation shuts down faulty Agg A

Firmware-upgrade schedules Agg B to upgrade

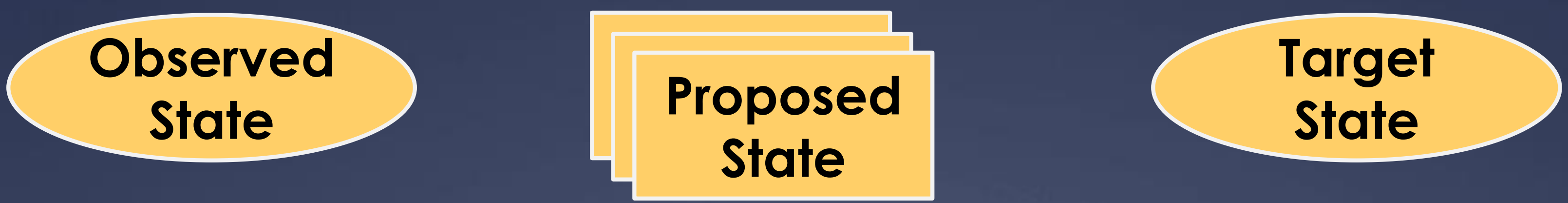
# The Statesman System

- Network operating system
  - Common layer to consolidate traffic and infrastructure management
  - Resolve conflicts & safety violations

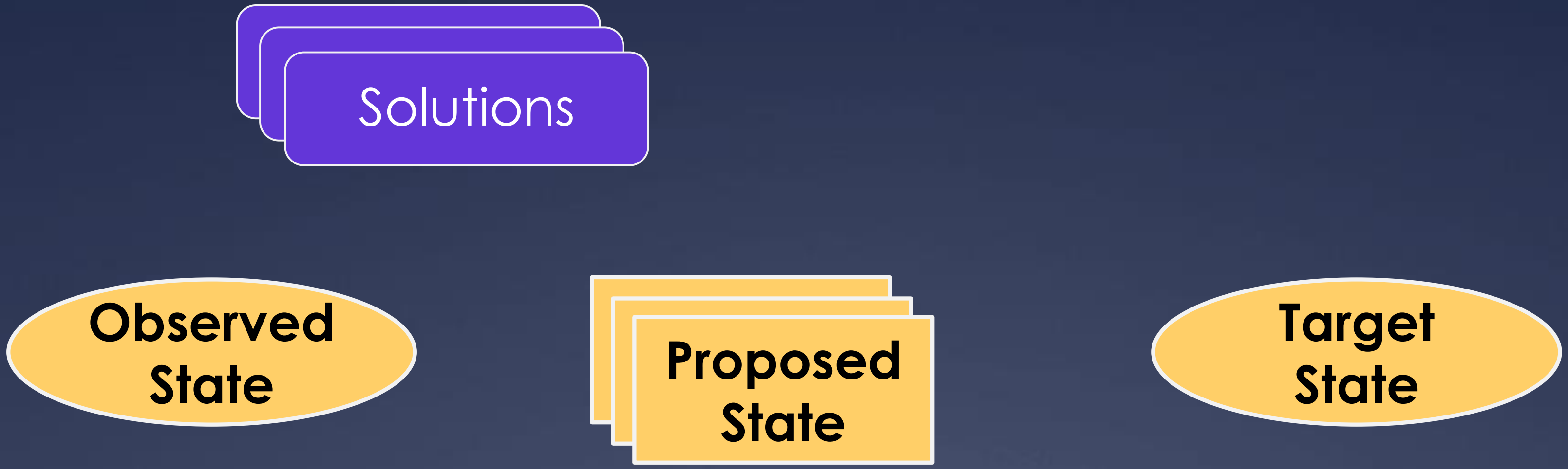
# The Statesman System

- Network operating system
  - Common layer to consolidate traffic and infrastructure management
  - Resolve conflicts & safety violations
- Core techniques:
  - Network state & three-view workflow
  - State dependency model
  - Scalable & robust system

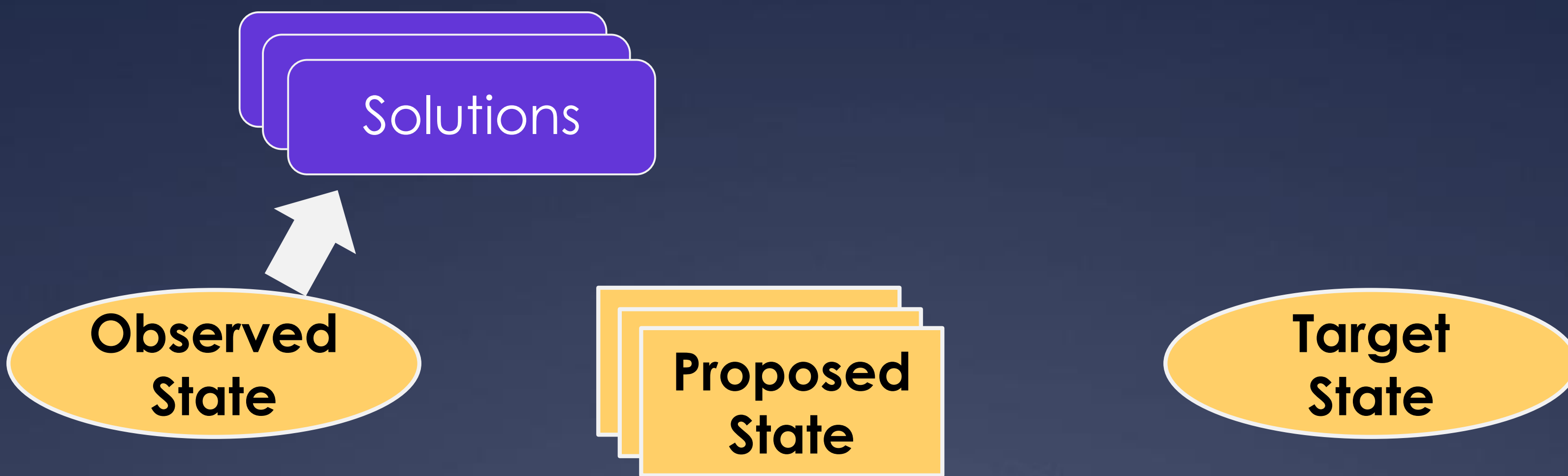
# Three-view Workflow with Network State



# Three-view Workflow with Network State

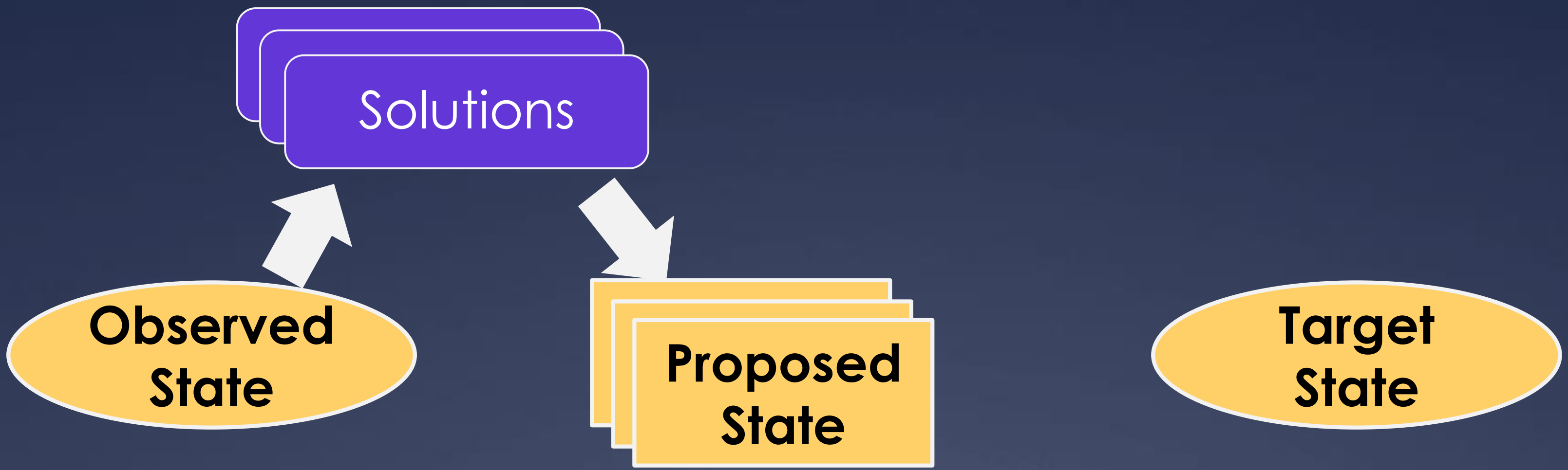


# Three-view Workflow with Network State




**What we see from the network**

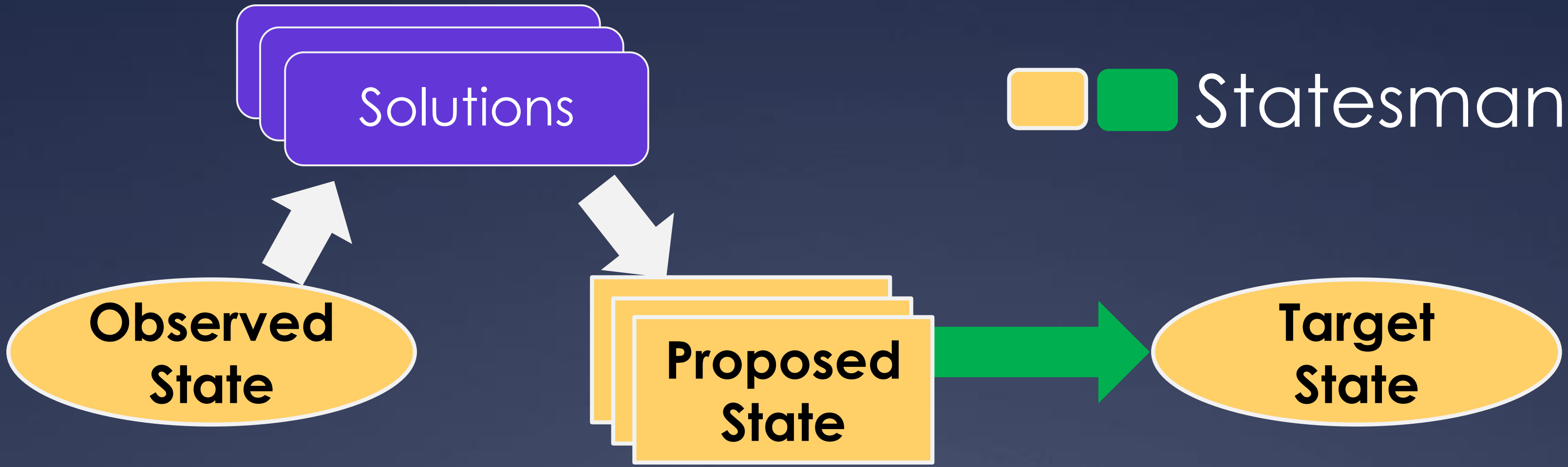
# Three-view Workflow with Network State




**What we see from the network**

**What we want the network to be**

# Three-view Workflow with Network State

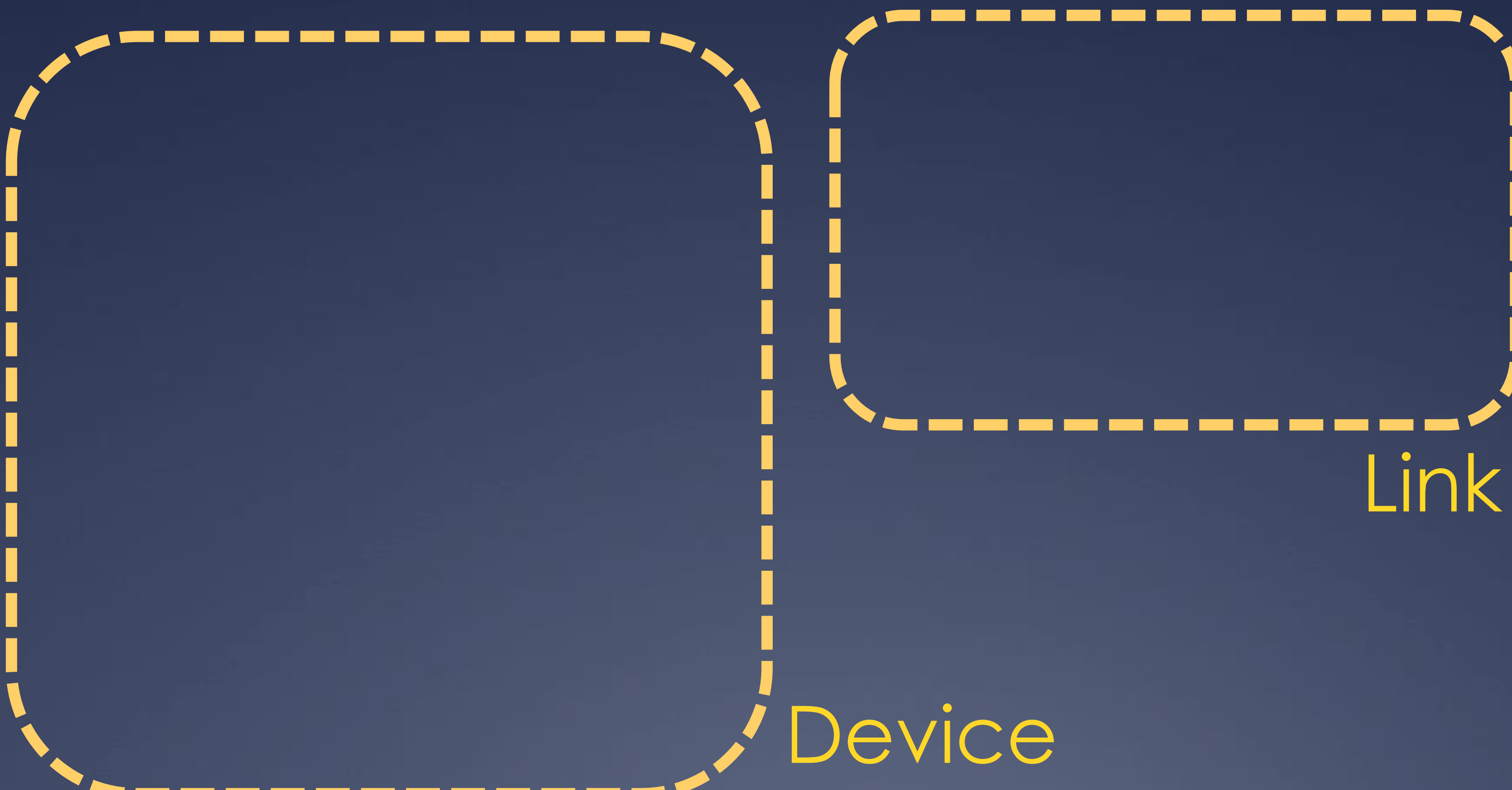



**What we see from the network**

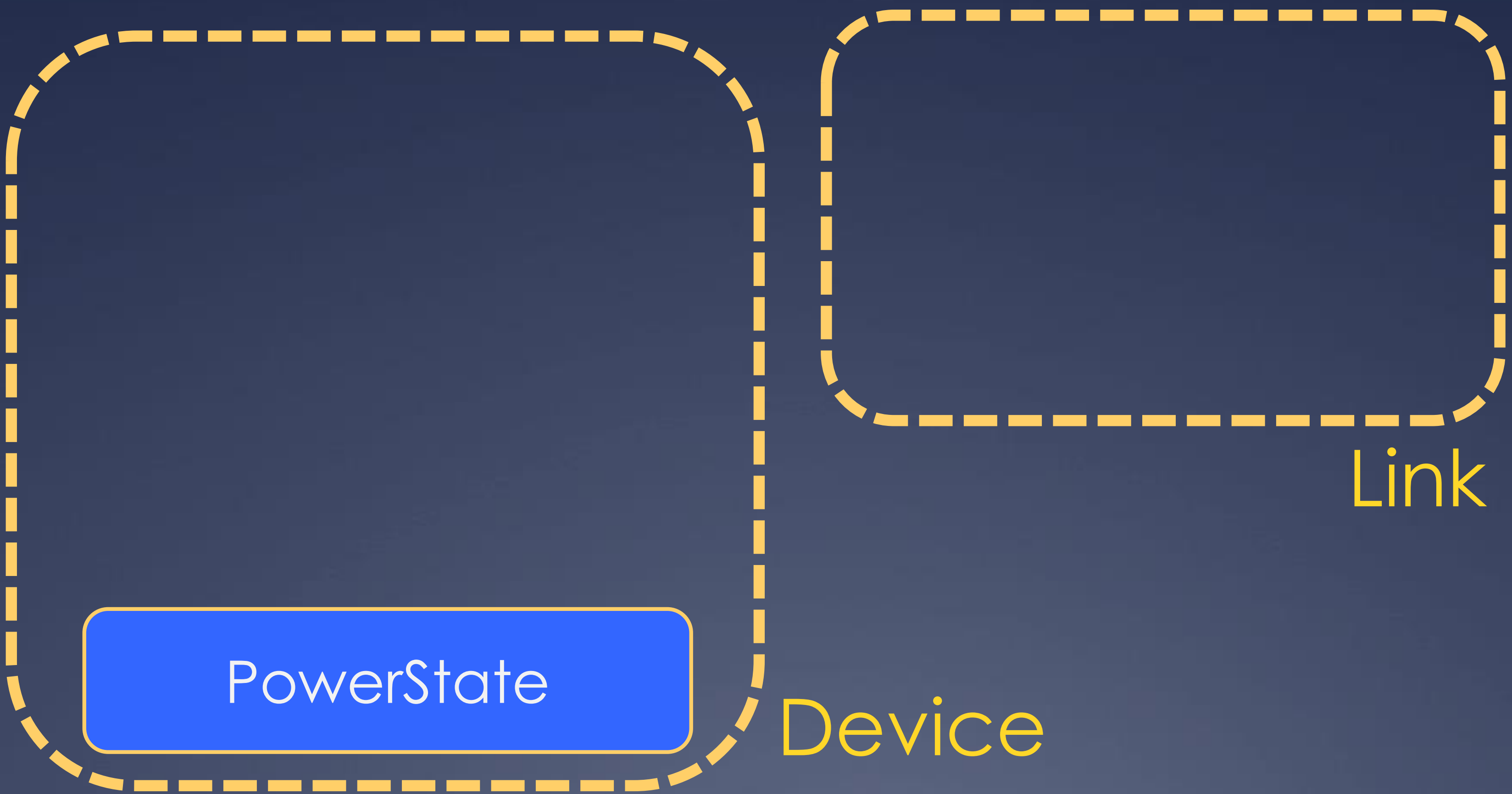
**What we want the network to be**

**What can be actually done on the network**

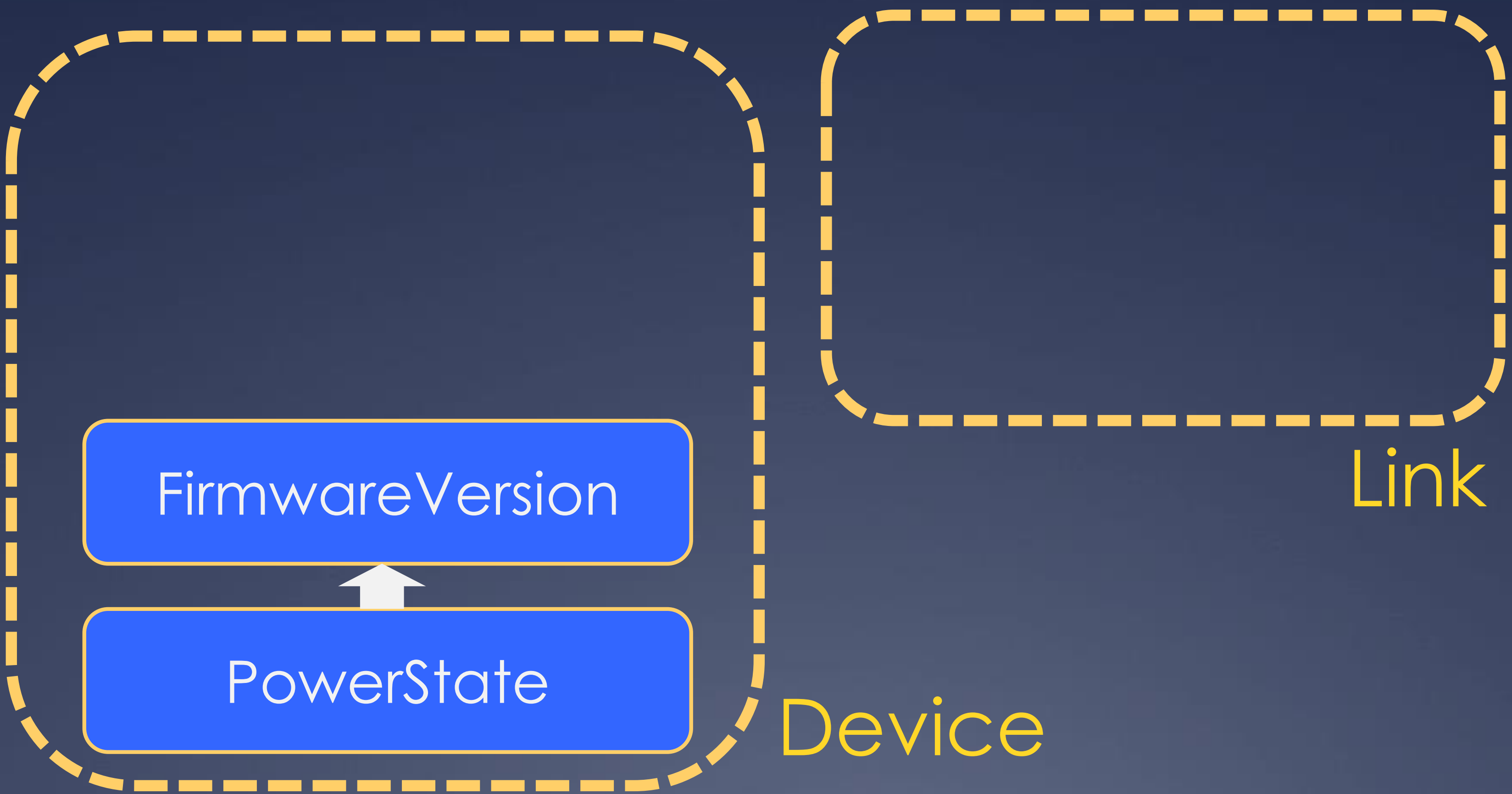
# State Dependency for Conflict Detection



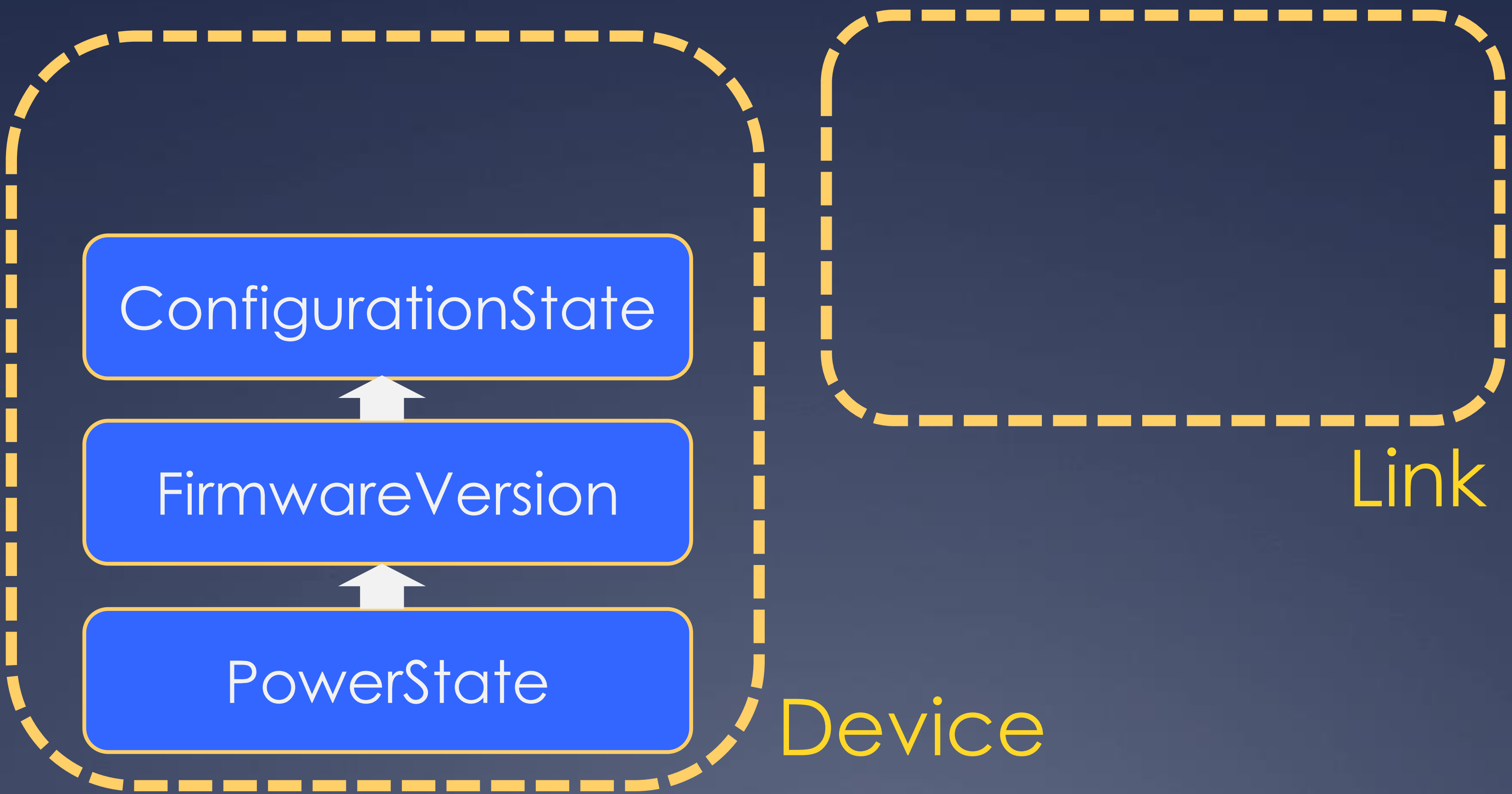
# State Dependency for Conflict Detection



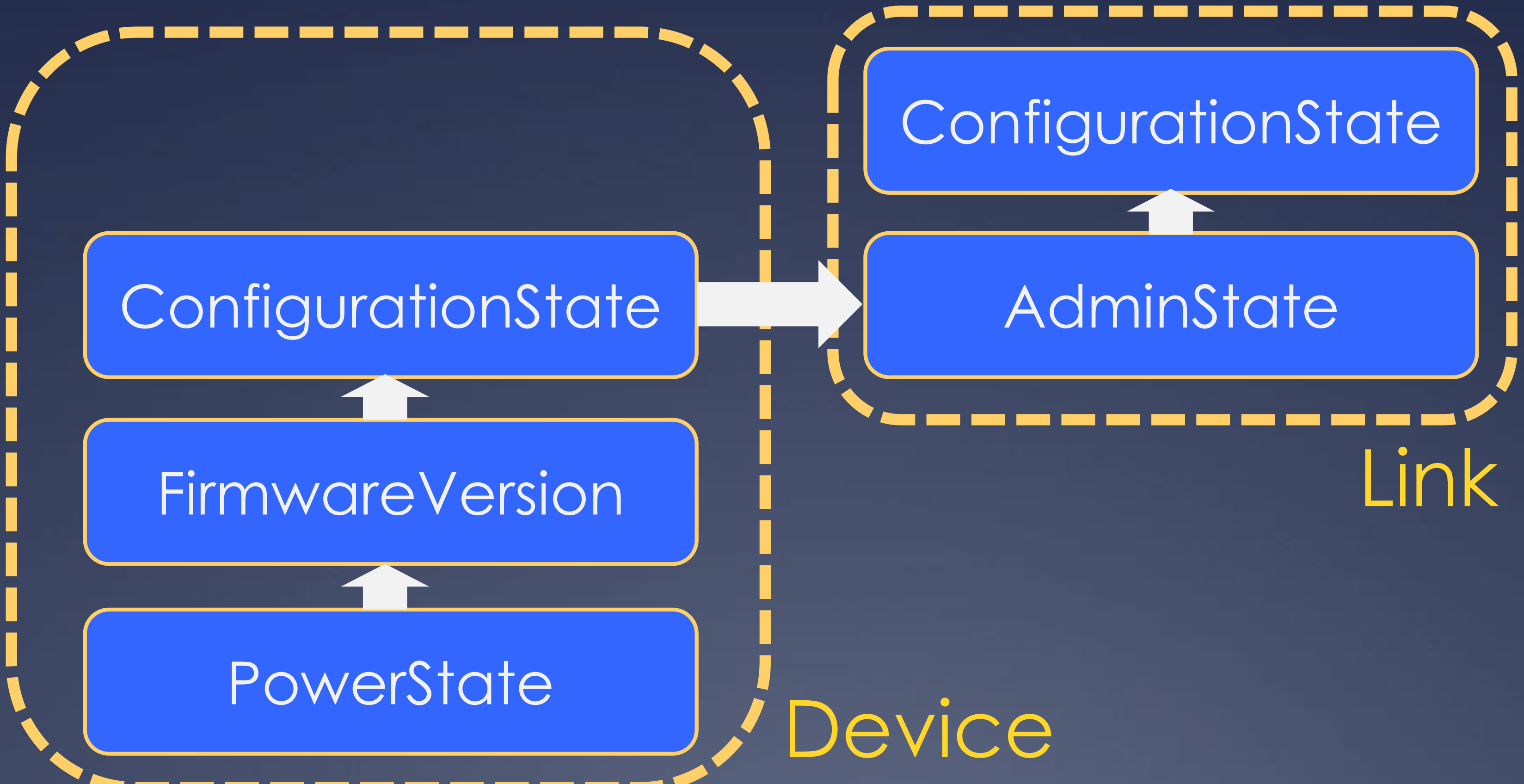
# State Dependency for Conflict Detection



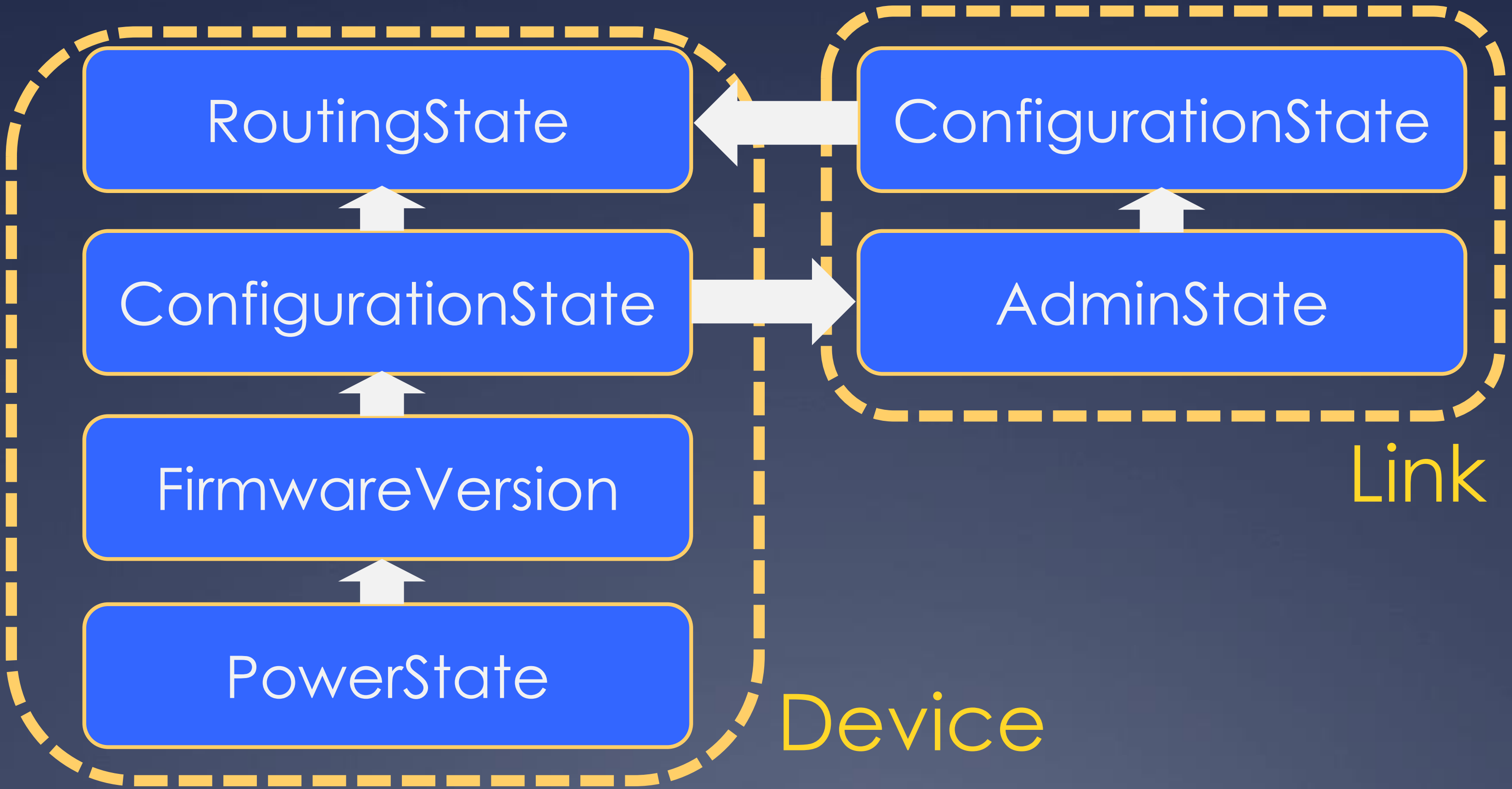
# State Dependency for Conflict Detection



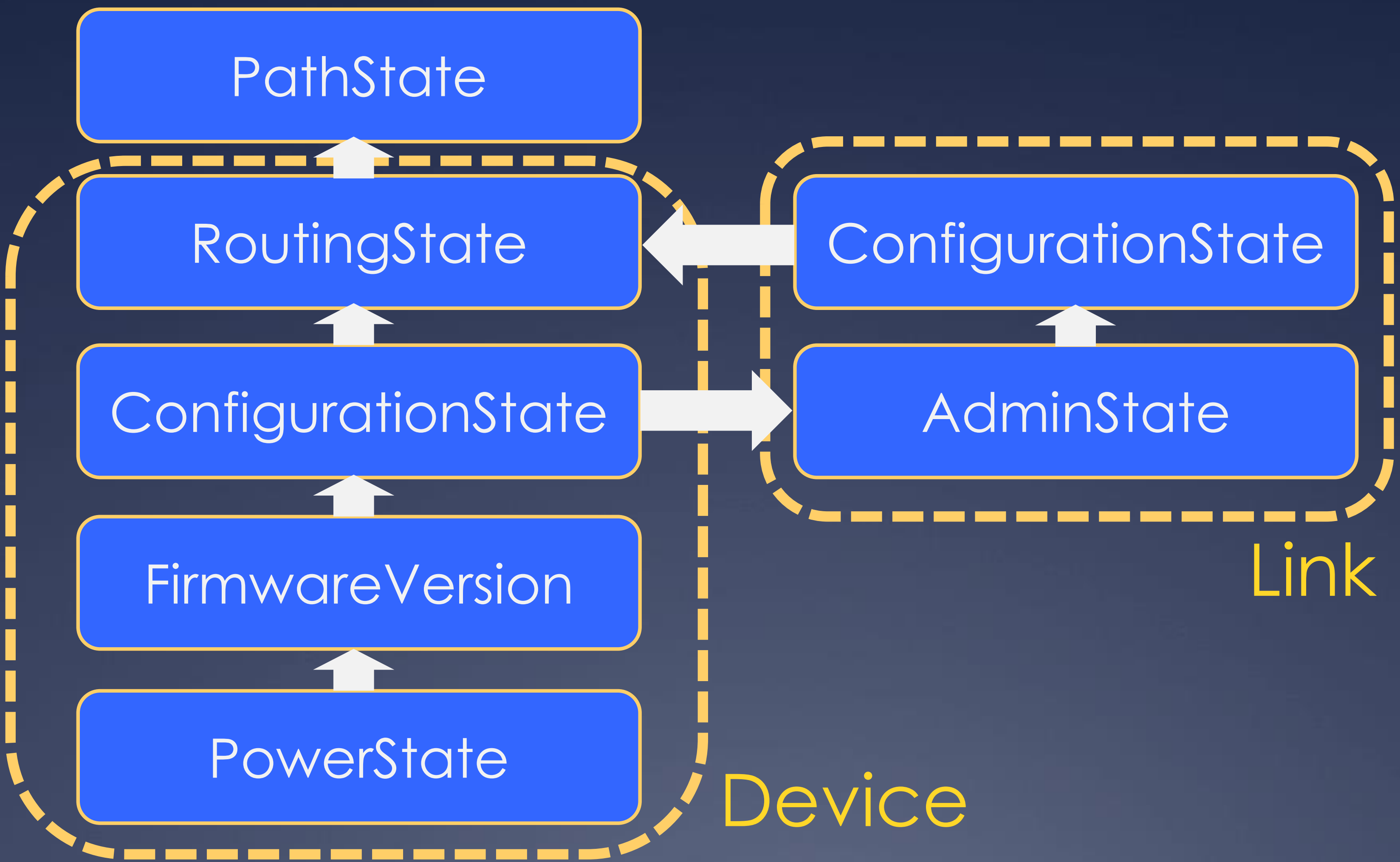
# State Dependency for Conflict Detection



# State Dependency for Conflict Detection



# State Dependency for Conflict Detection

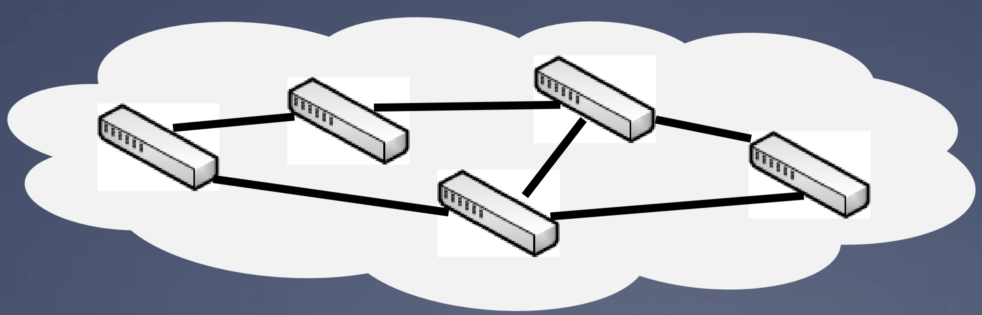


# Statesman System

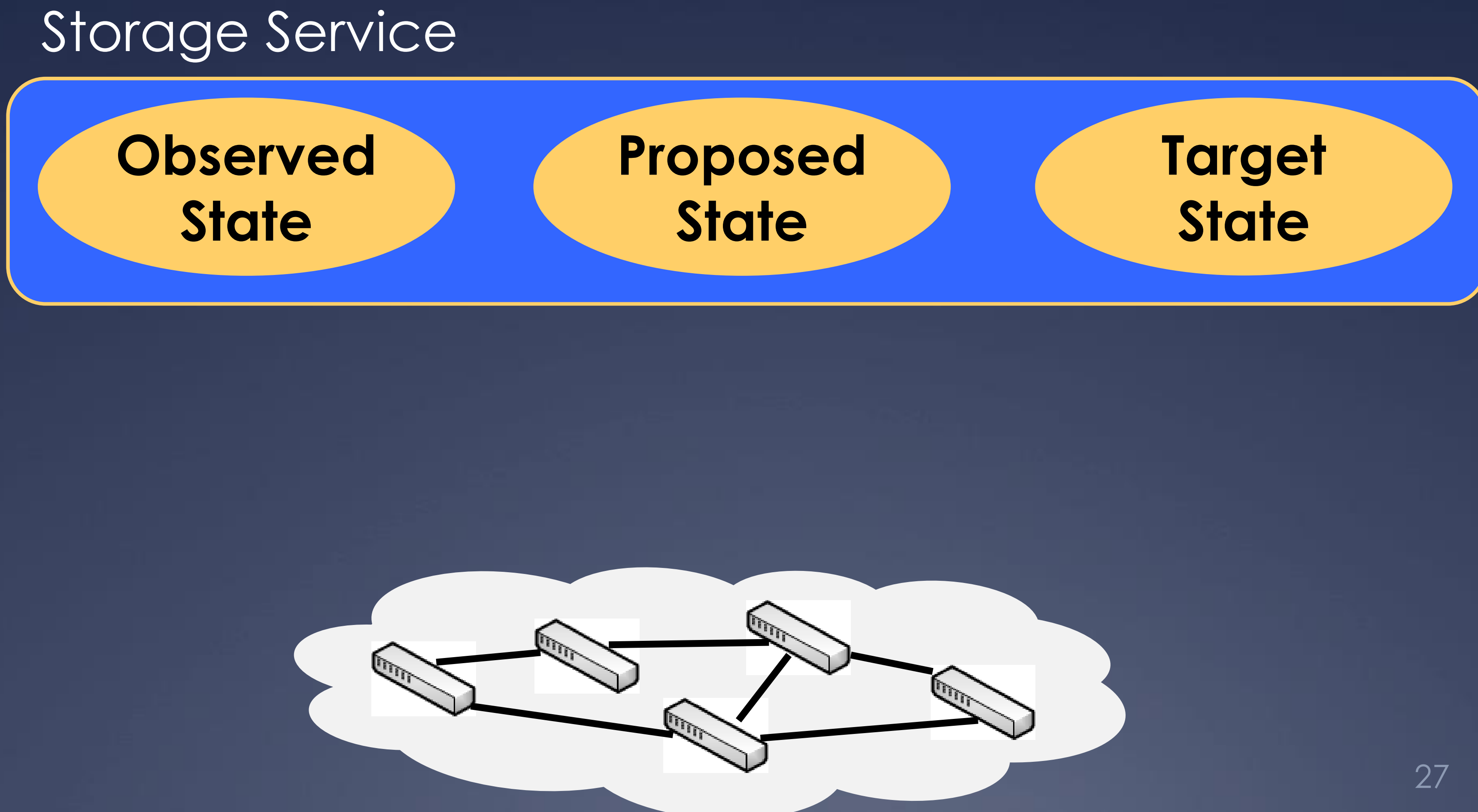
Observed State

Proposed State

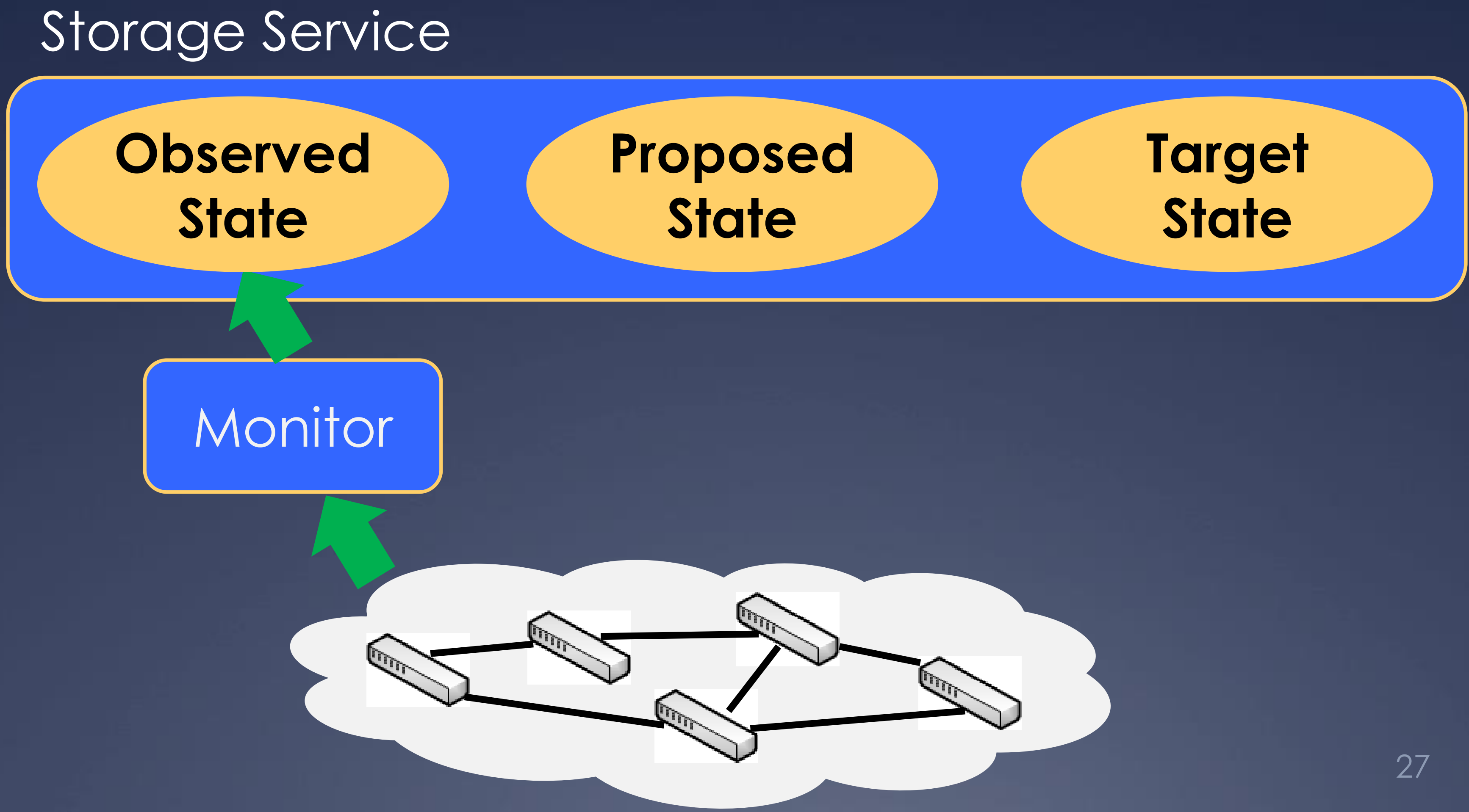
Target State



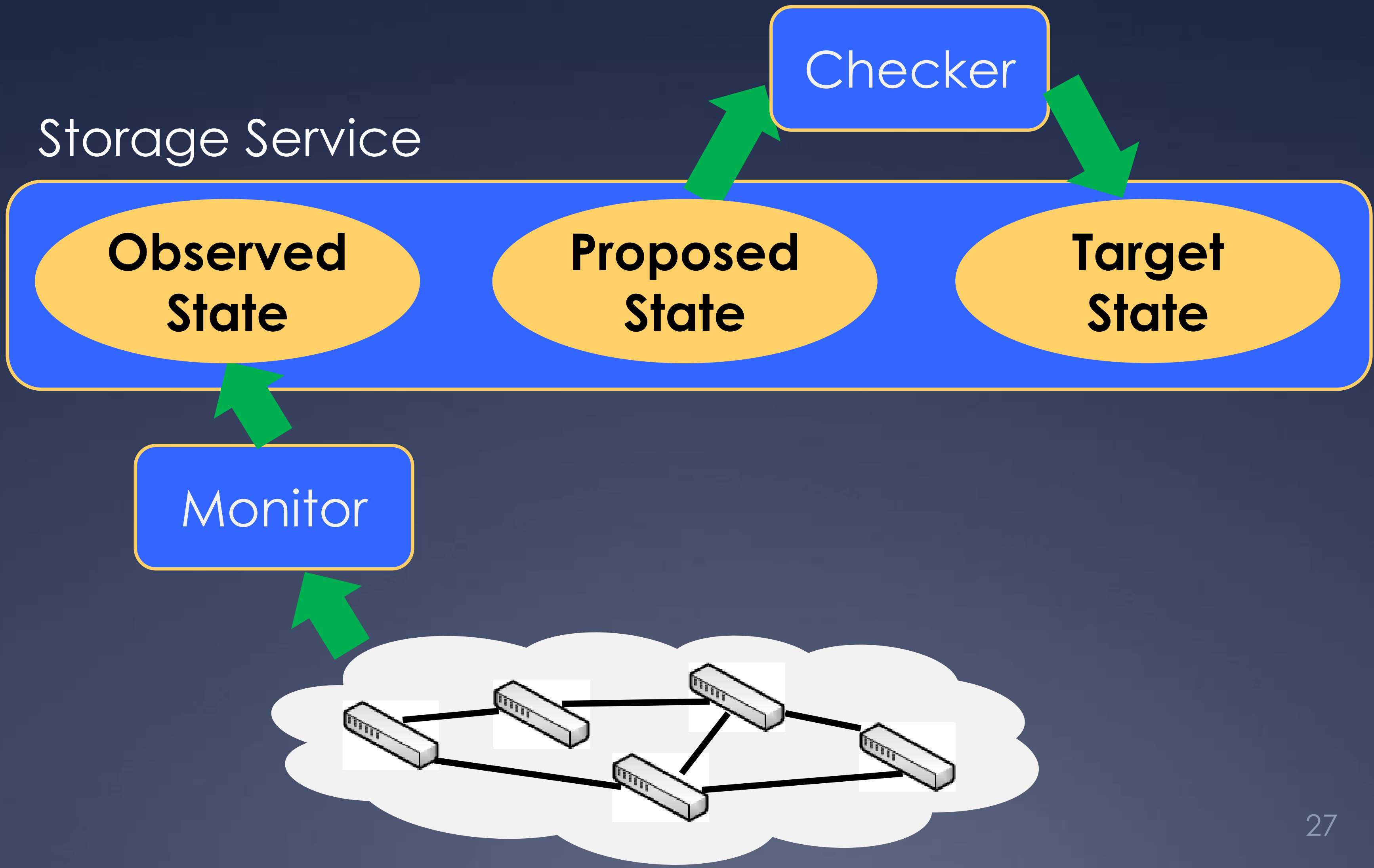
# Statesman System



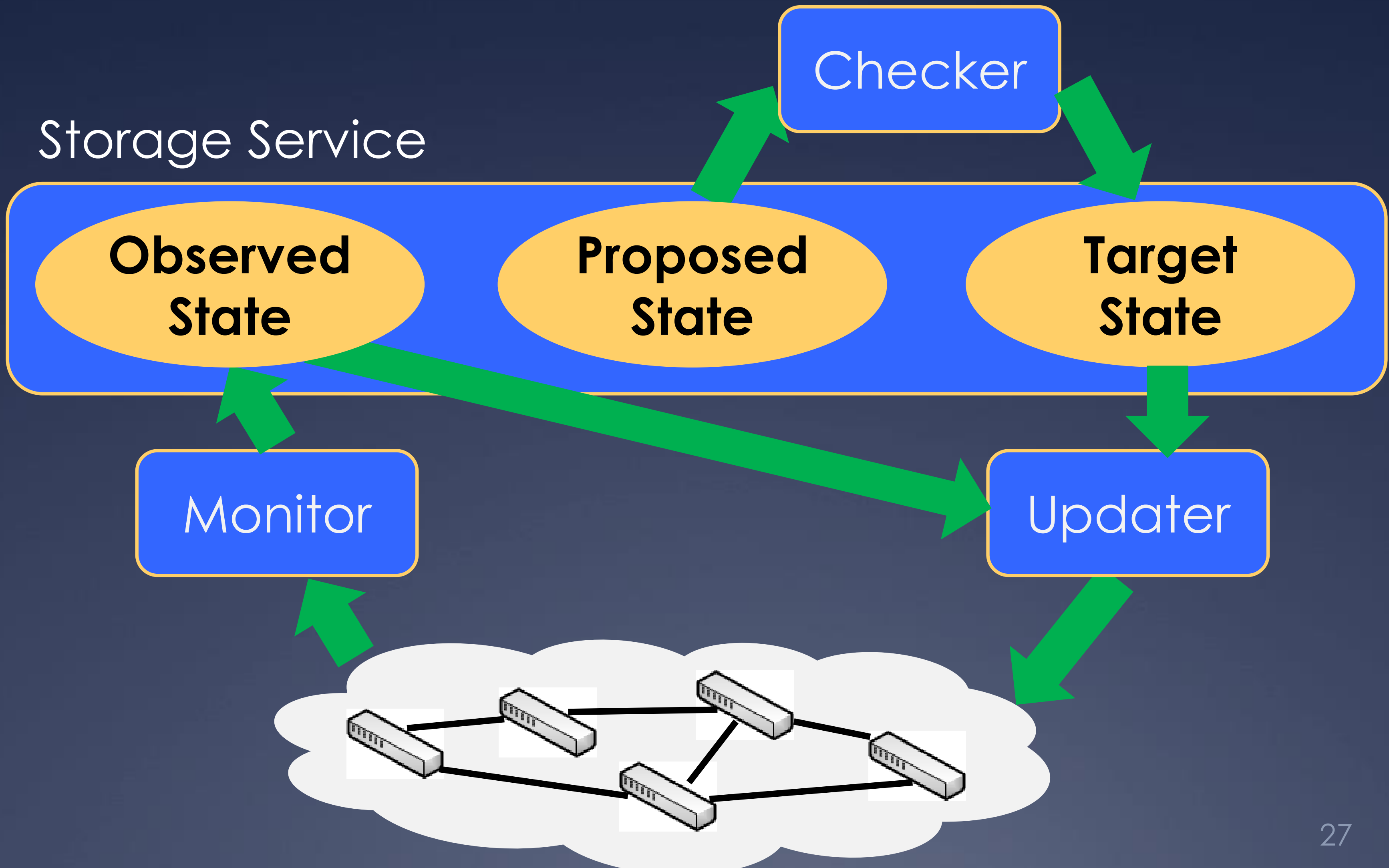
# Statesman System



# Statesman System



# Statesman System



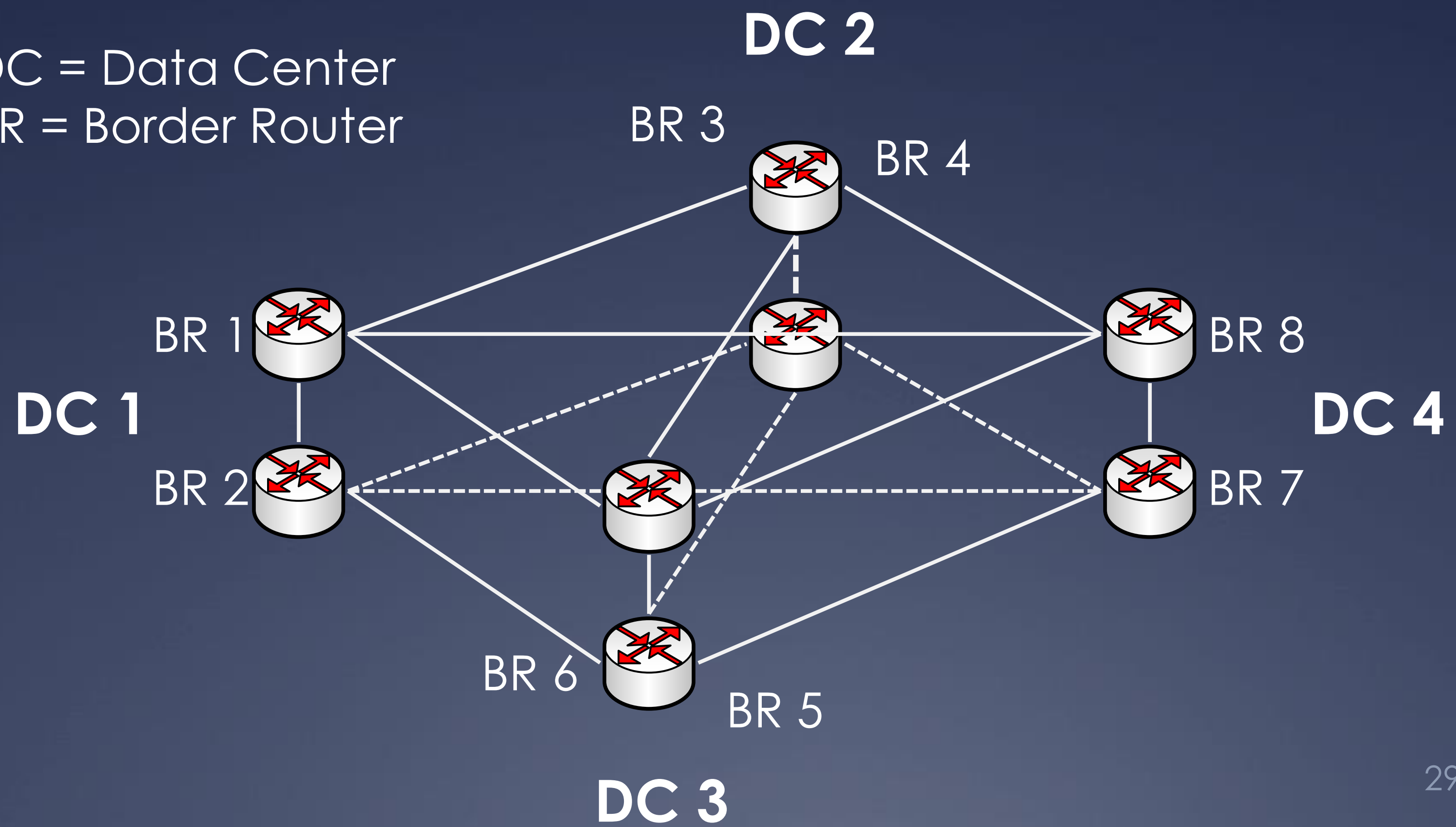
# Deployment Overview

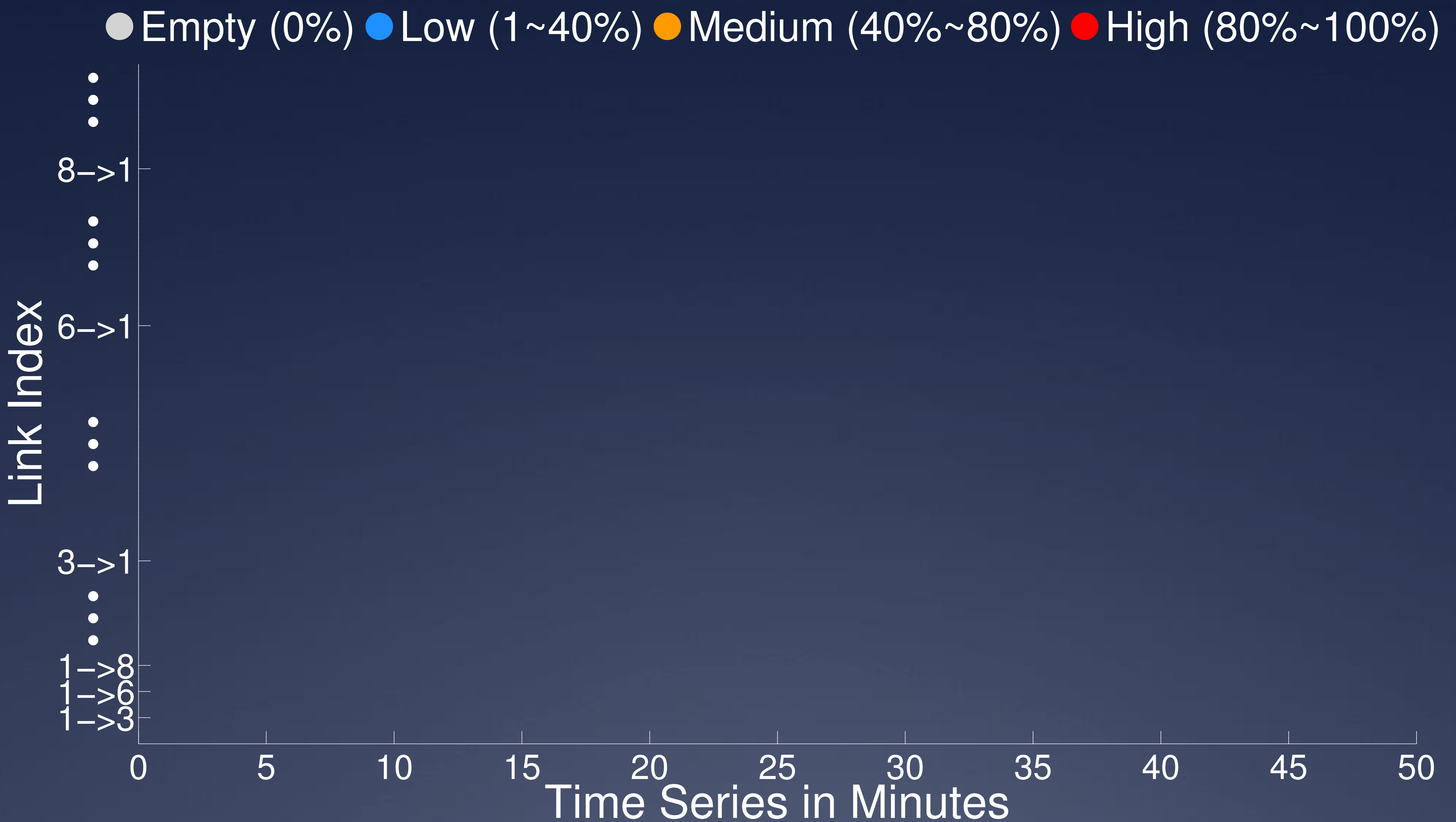
- Operational in Microsoft Azure since October 2013
- Cover 10 DCs of 20K devices
- 3 production management solutions built and running

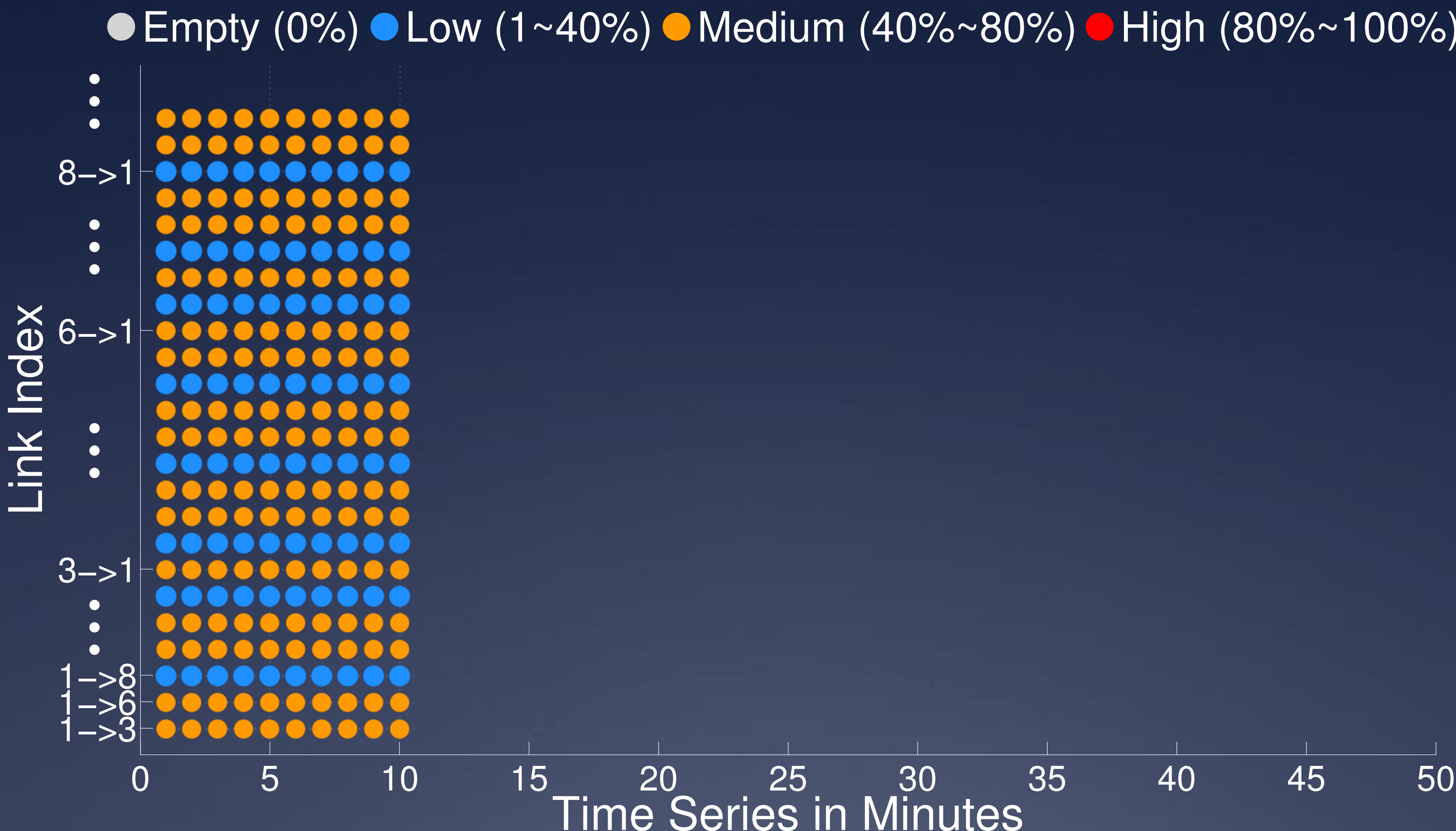
# Case: Resolve Conflict

## Inter-DC TE &
## Firmware-upgrade

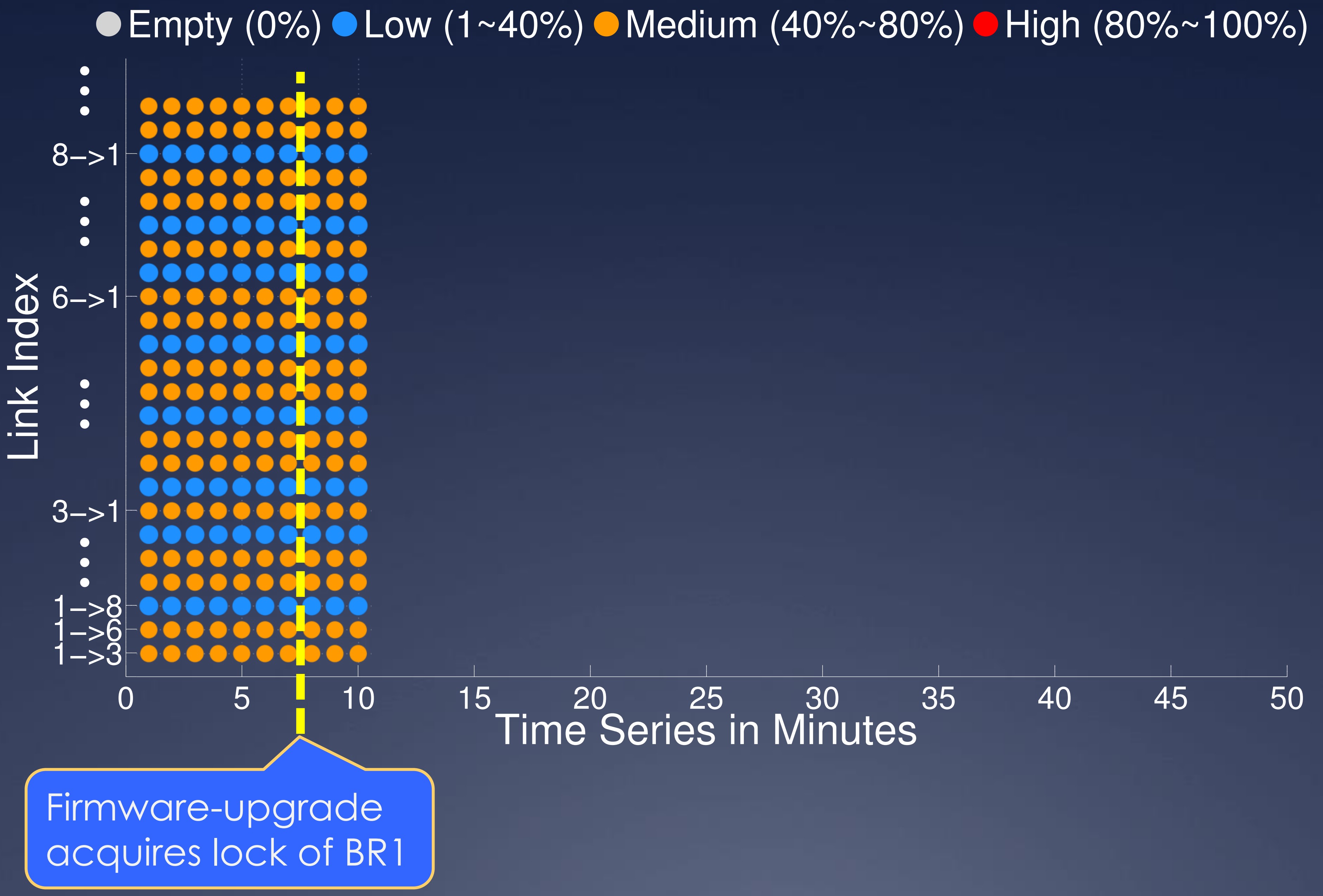
DC = Data Center
BR = Border Router



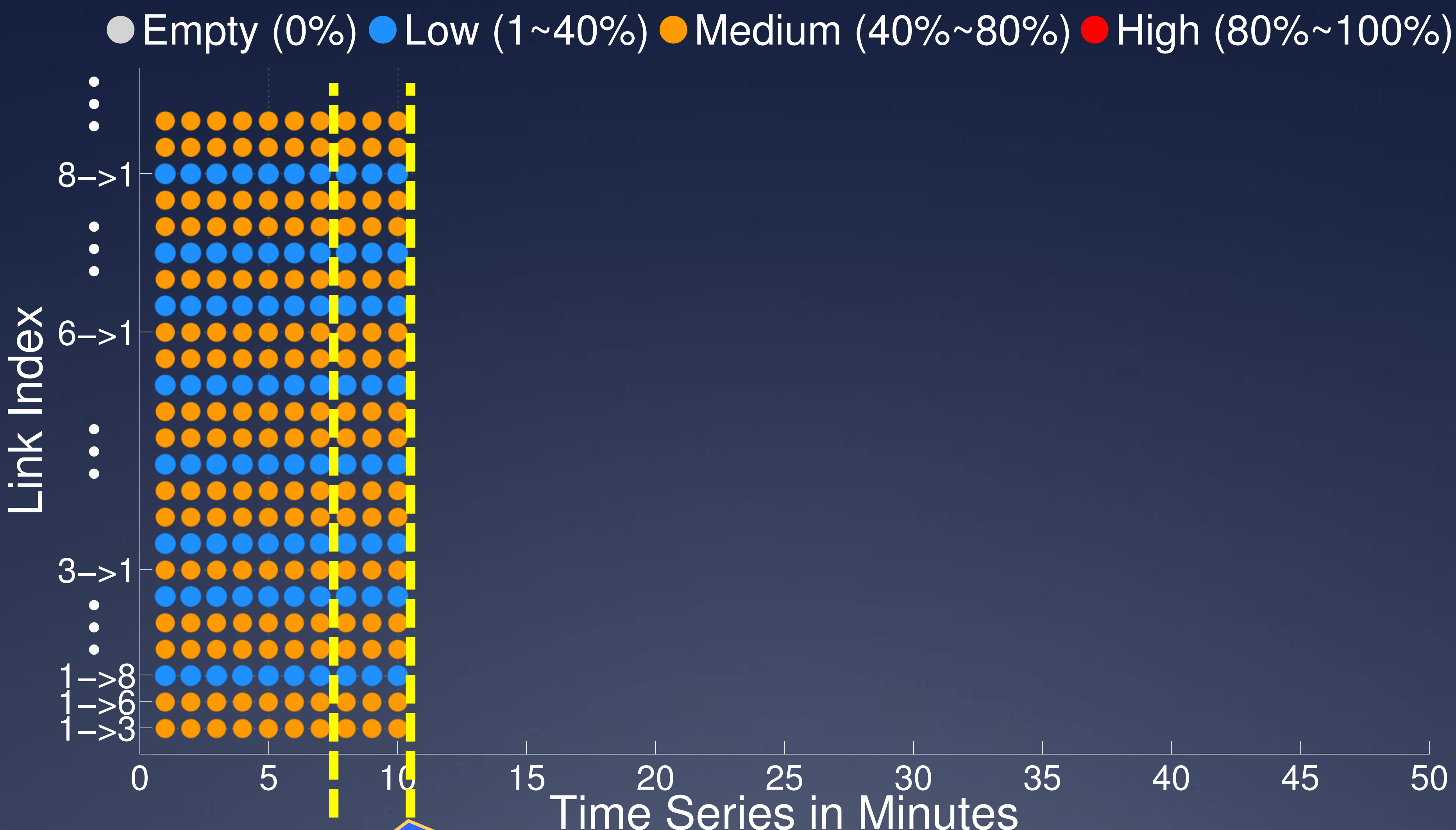

Empty (0%)
Low (1~40%)
Medium (40%~80%)
High (80%~100%)
Link Index
8–>1
6–>1
3–>1
1–>8
1–>6
1–>3
0
5
10
15
20
25
30
35
40
45
50
Time Series in Minutes



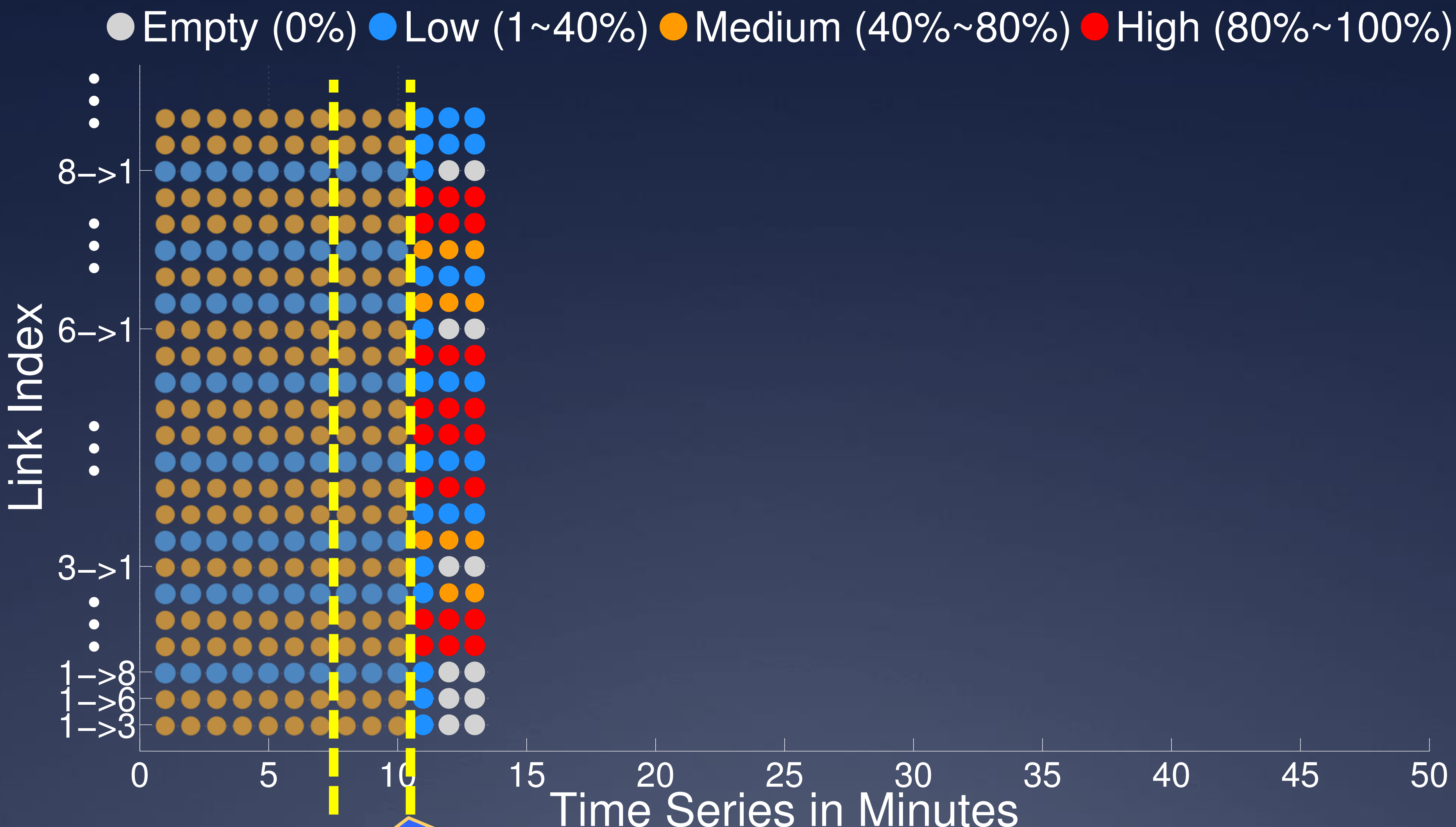
Empty (0%)
Low (1~40%)
Medium (40%~80%)
High (80%~100%)
Link Index
8−>1
6−>1
3−>1
1−>8
1−>6
1−>3
0
5
10
15
20
25
30
35
40
45
50
Time Series in Minutes



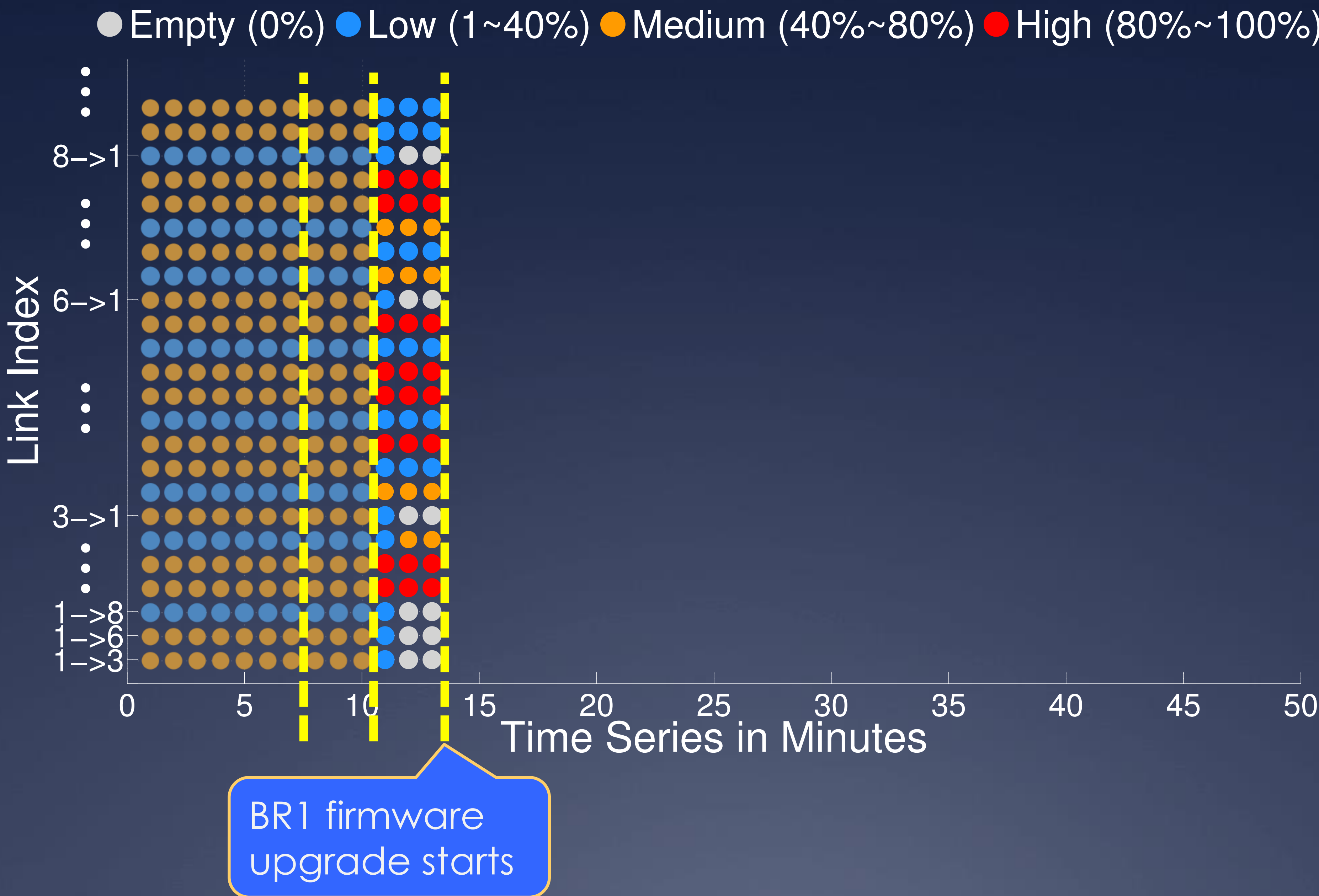
Empty (0%)
Low (1~40%)
Medium (40%~80%)
High (80%~100%)
Link Index
8–>1
6–>1
3–>1
1–>8
1–>6
1–>3
0
5
10
15
20
25
30
35
40
45
50
Time Series in Minutes
Firmware-upgrade acquires lock of BR1

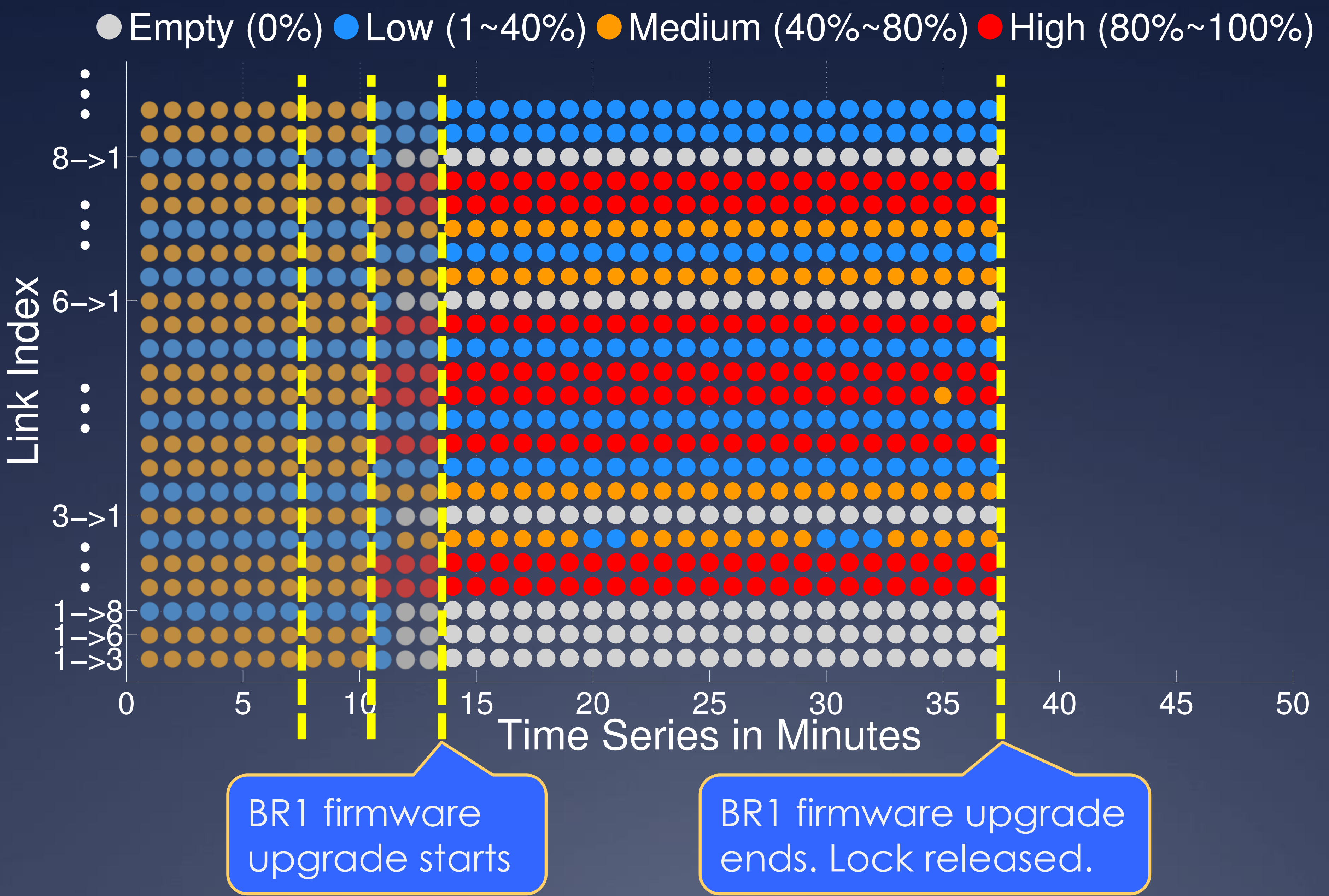


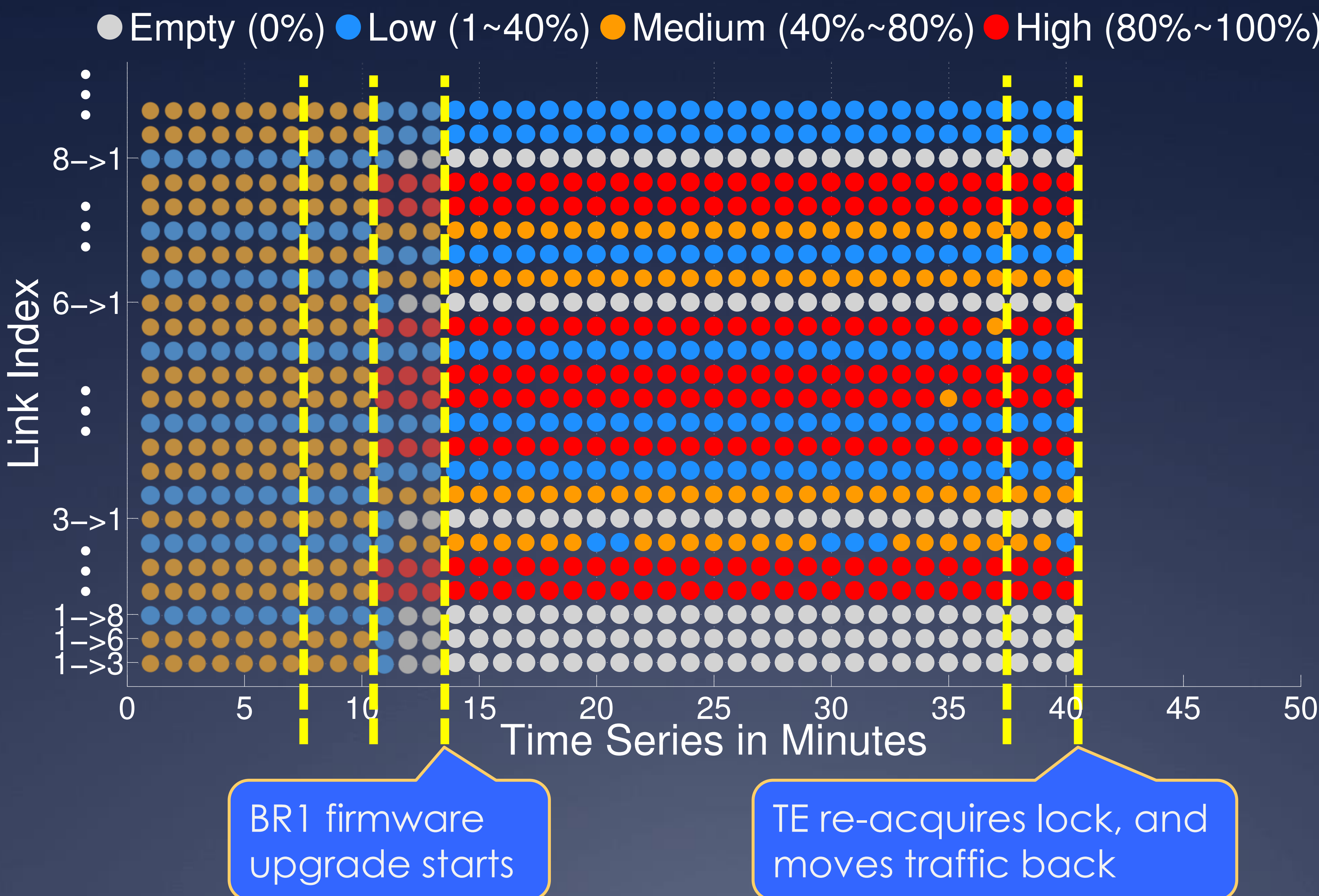

TE fails to acquire lock, and moves traffic away

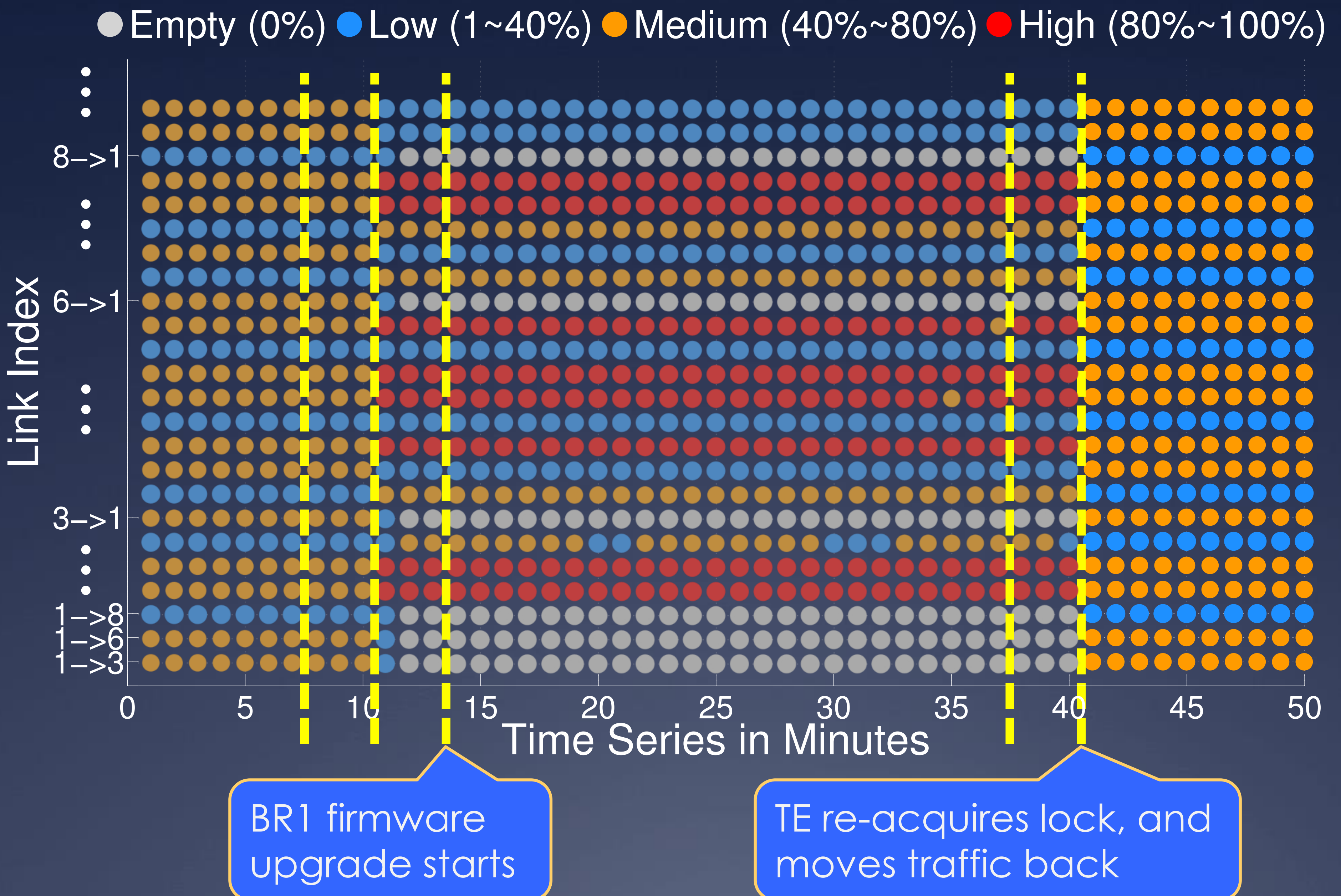



TE fails to acquire lock, and moves traffic away


Empty (0%)
Low (1~40%)
Medium (40%~80%)
High (80%~100%)
Link Index
8−>1
6−>1
3−>1
1−>8
1−>6
1−>3
0
5
10
15
20
25
30
35
40
45
50
Time Series in Minutes
BR1 firmware upgrade starts


Empty (0%)
Low (1~40%)
Medium (40%~80%)
High (80%~100%)
Link Index
8−>1
6−>1
3−>1
1−>8
1−>6
1−>3
0
5
10
15
20
25
30
35
40
45
50
Time Series in Minutes
BR1 firmware upgrade starts
BR1 firmware upgrade ends. Lock released.


Empty (0%)
Low (1~40%)
Medium (40%~80%)
High (80%~100%)
Link Index
8−>1
6−>1
3−>1
1−>8
1−>6
1−>3
0
5
10
15
20
25
30
35
40
45
50
Time Series in Minutes
BR1 firmware upgrade starts
TE re-acquires lock, and moves traffic back



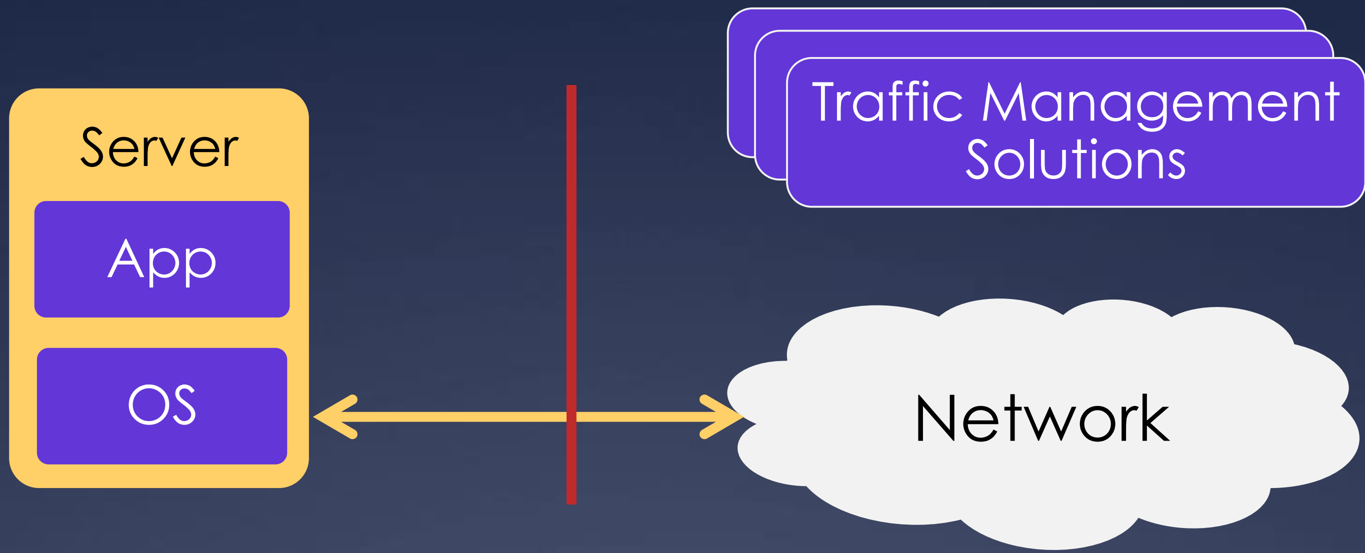
Empty (0%)
Low (1~40%)
Medium (40%~80%)
High (80%~100%)
Link Index
8–>1
6–>1
3–>1
1–>8
1–>6
1–>3
0
5
10
15
20
25
30
35
40
45
50
Time Series in Minutes
BR1 firmware upgrade starts
TE re-acquires lock, and moves traffic back

# Statesman Summary

- Programming abstraction
  - Three-view network state model
- Efficient and scalable system
  - Automatic and safe infrastructure management system
- Deployment
  - Operational in Microsoft Azure worldwide
- SIGCOMM 2014

# Hone:
# Combining End-host and Network for Traffic Management

# Problem




- Traffic management is limited by the scope of network devices
  - *Coarse granularity & limited view*

# The Hone System

- Bring end hosts into traffic mgmt.

# The Hone System

- Bring end hosts into traffic mgmt.
- Core techniques:
  - Access of socket & transport layers

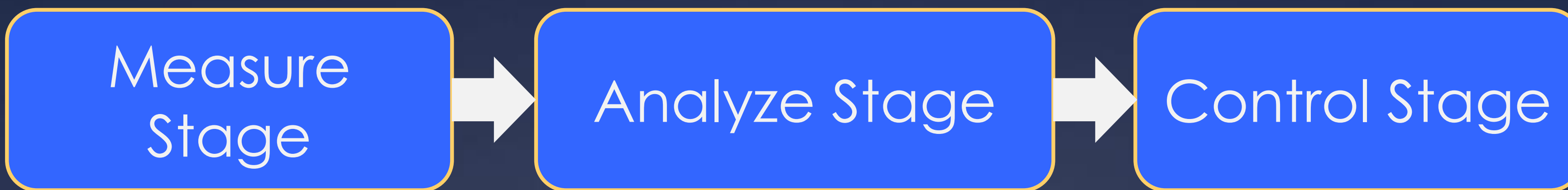
# The Hone System

- Bring end hosts into traffic mgmt.
- Core techniques:
  - Access of socket & transport layers
  - Expressive three-stage programming framework

# The Hone System

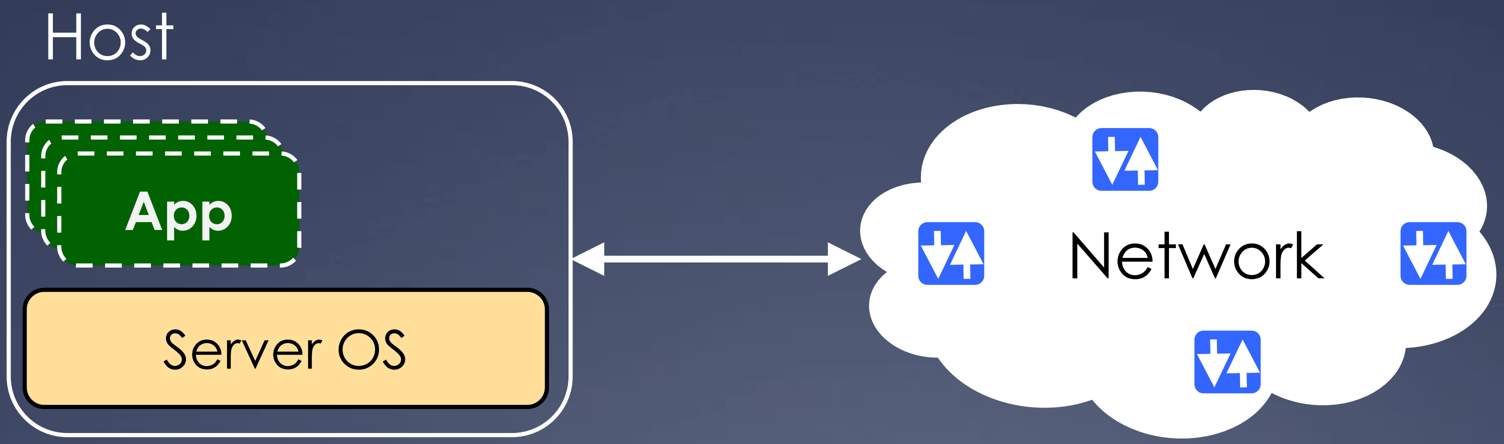
- Bring end hosts into traffic mgmt.
- Core techniques:
  - Access of socket & transport layers
  - Expressive three-stage programming framework
  - Efficient system with parallel execution of partitioned program

# Programming Model

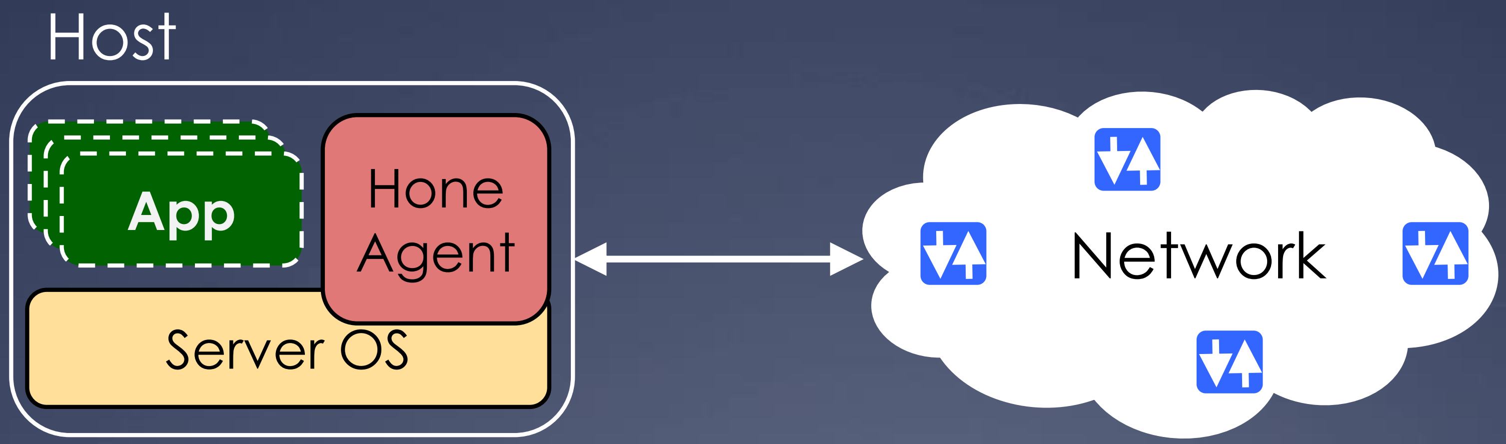



- Framework for each stage
- Programmable body of each stage
  - *Focus on measurement and analysis*

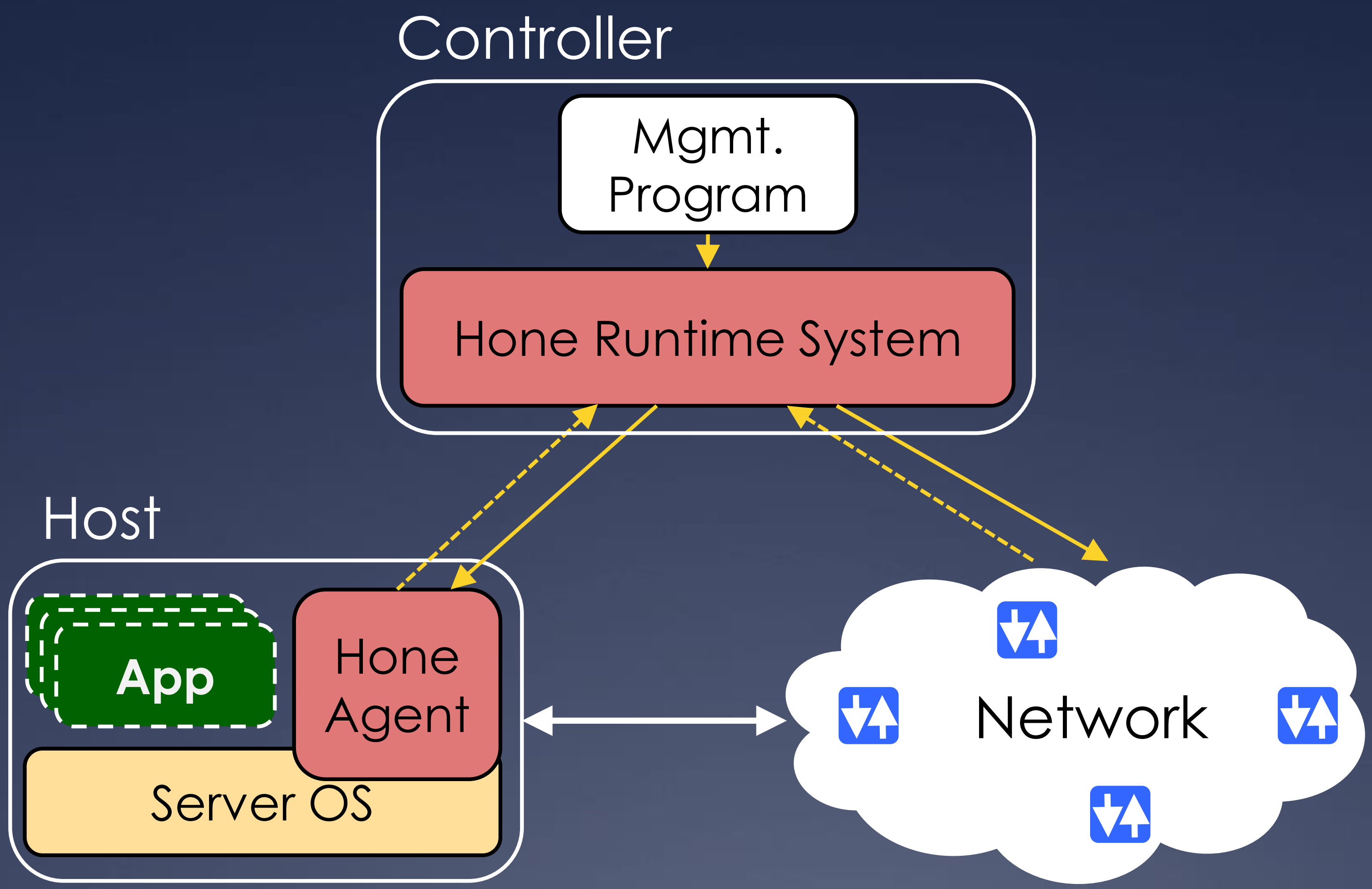
# Hone System



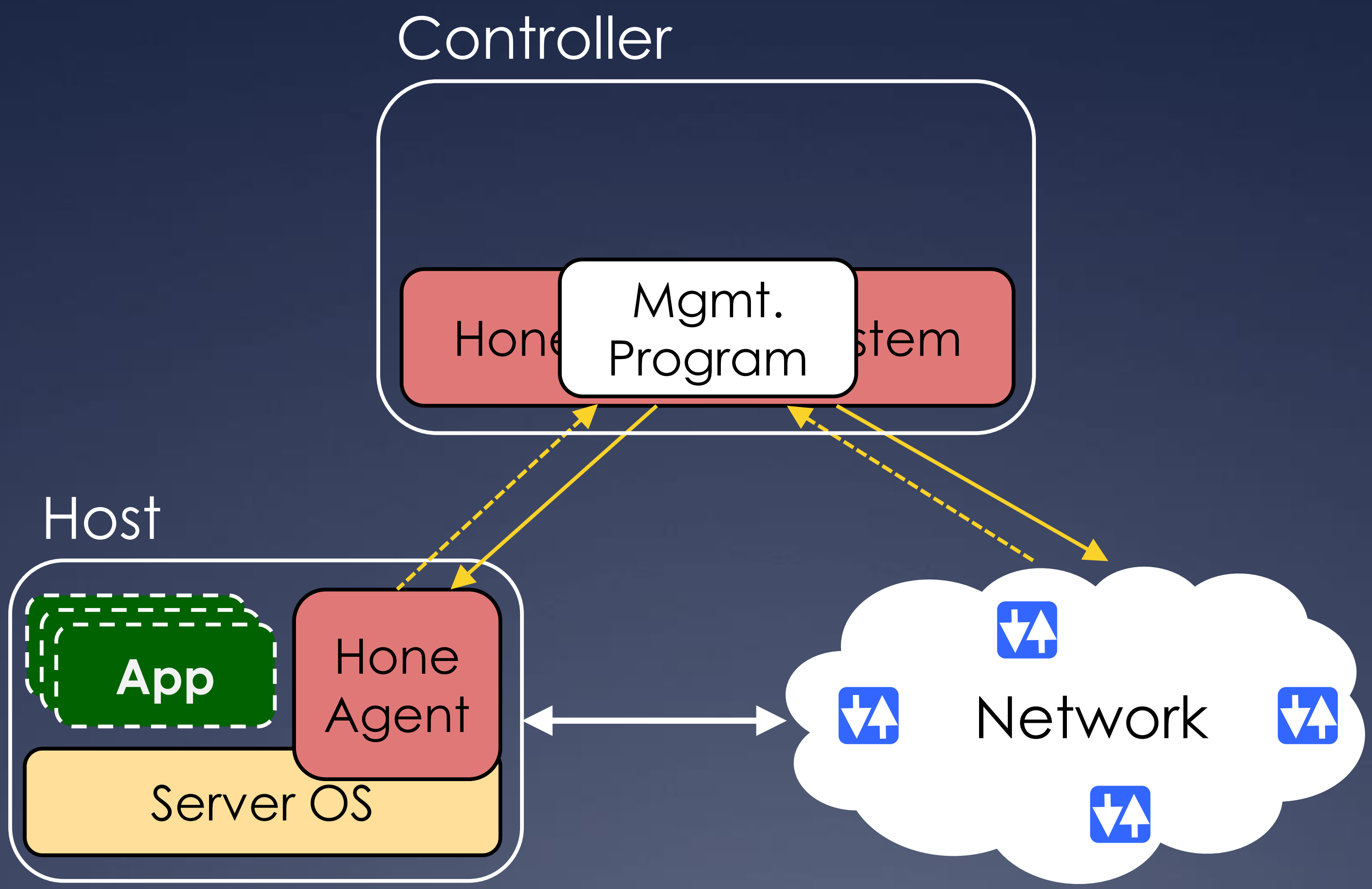
# Hone System



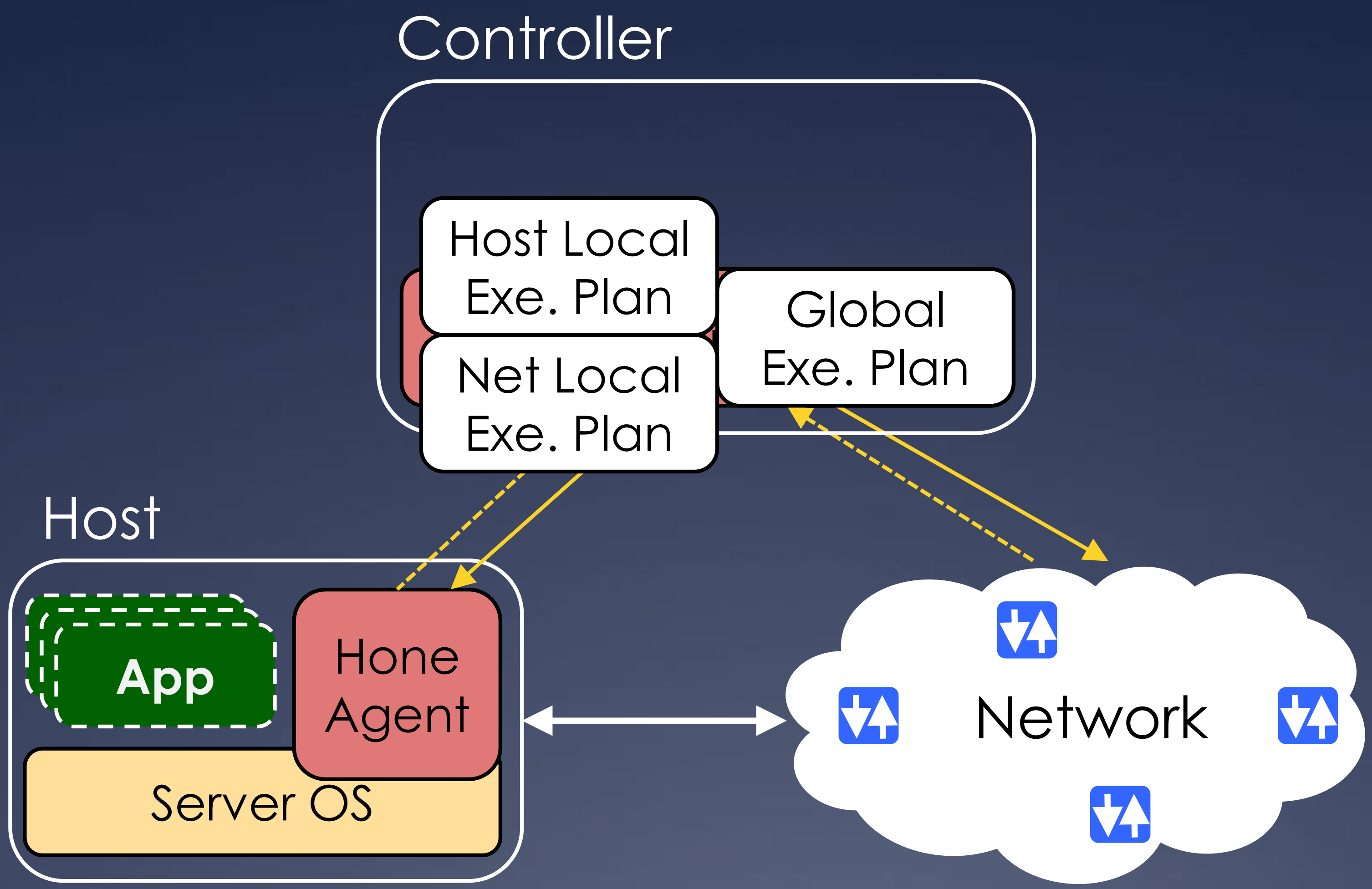
# Hone System


Controller
Mgmt. Program
Hone Runtime System
Host
App
Hone Agent
Server OS
Network

# Hone System

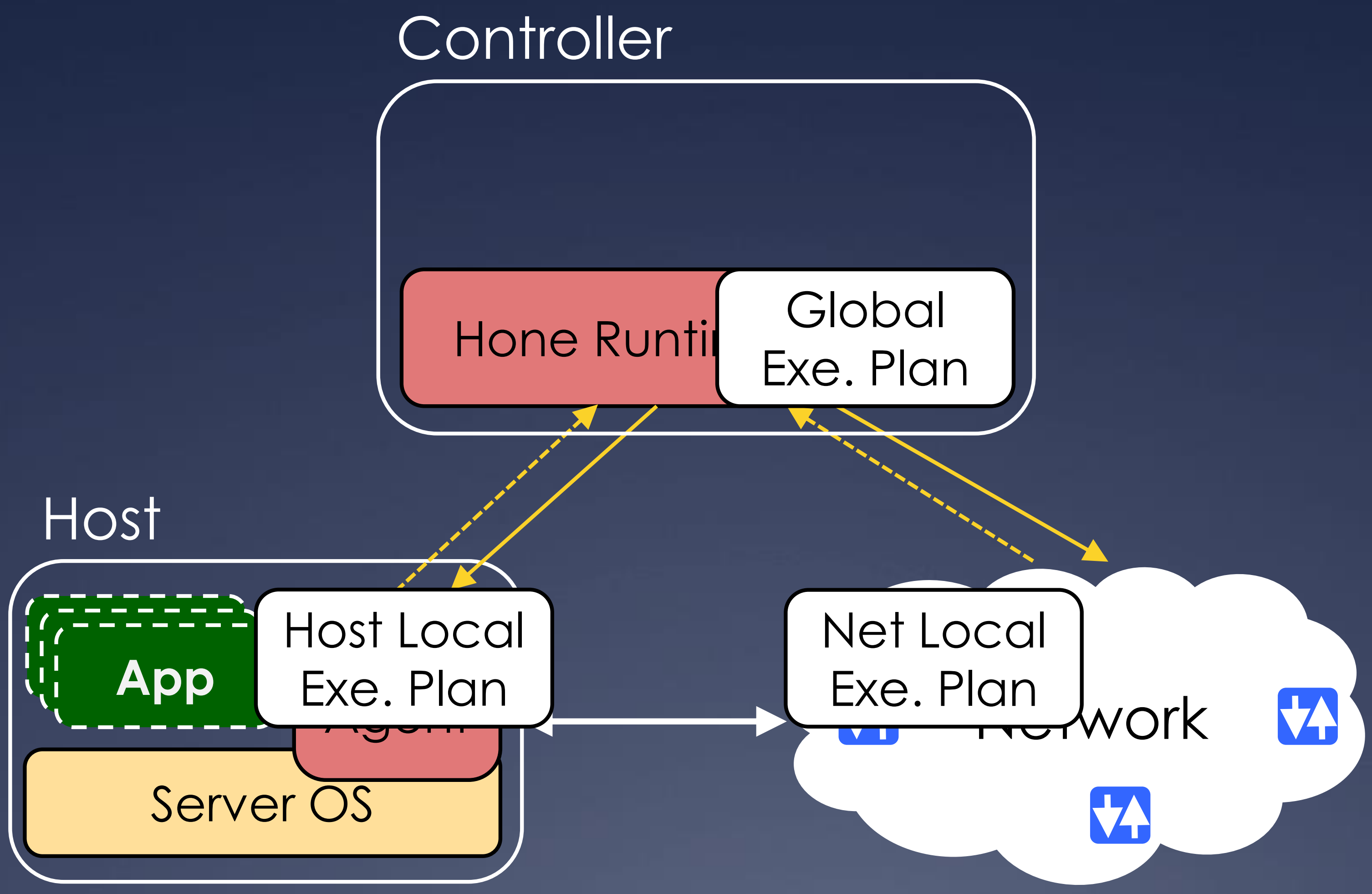

Controller
Hone
Mgmt.
Program
stem
Host
App
Hone
Agent
Server OS
Network

# Hone System



Controller
Host Local Exe. Plan
Net Local Exe. Plan
Global Exe. Plan
Host
App
Hone Agent
Server OS
Network

# Hone System



Controller
Hone Runti
Global
Exe. Plan
Host
App
Host Local
Exe. Plan
Server OS
Net Local
Exe. Plan

# Evaluation

- Built multiple traffic management solutions on Hone
- Show 'distributed rate limiting' here
  - *Limit total bandwidth across all instances of a tenant in a public cloud*

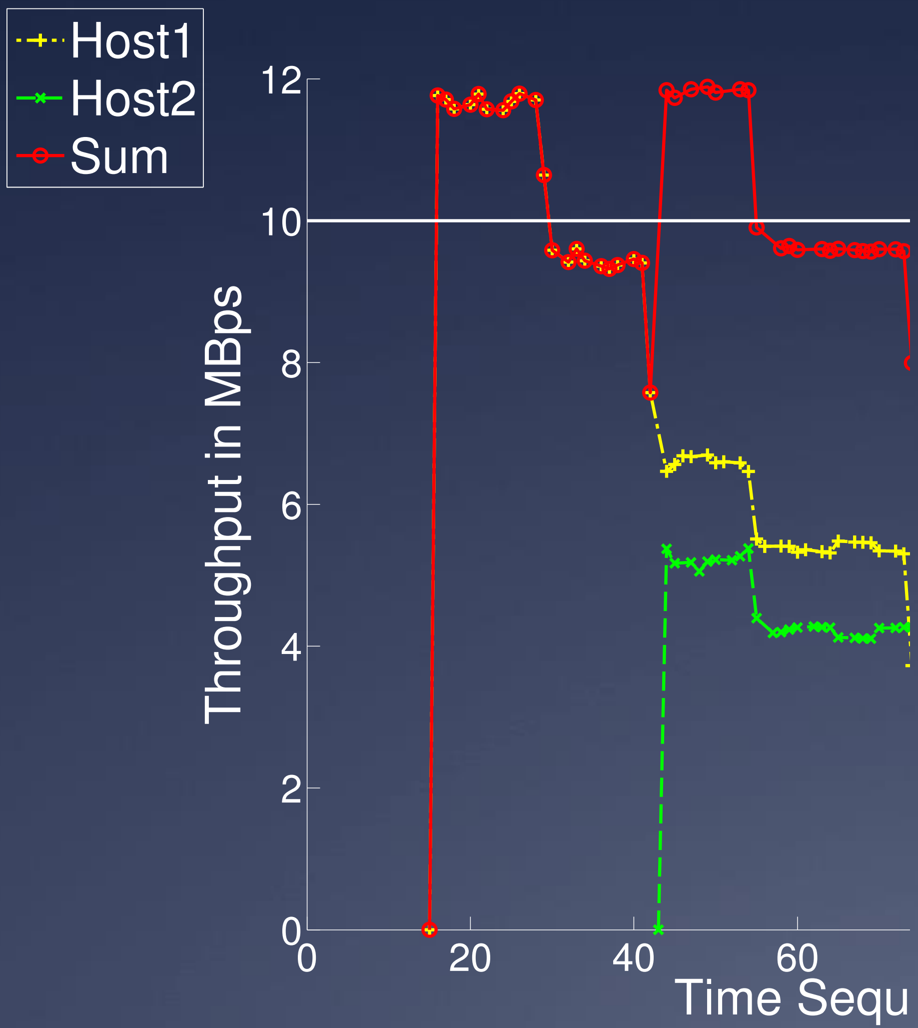
# DRL on HONE

- Host-side execution:
  - Measure
  - Calculate throughput
  - Aggregate among connections
- Controller-side execution:
  - Aggregate among hosts
  - Generate new rate-limit policies

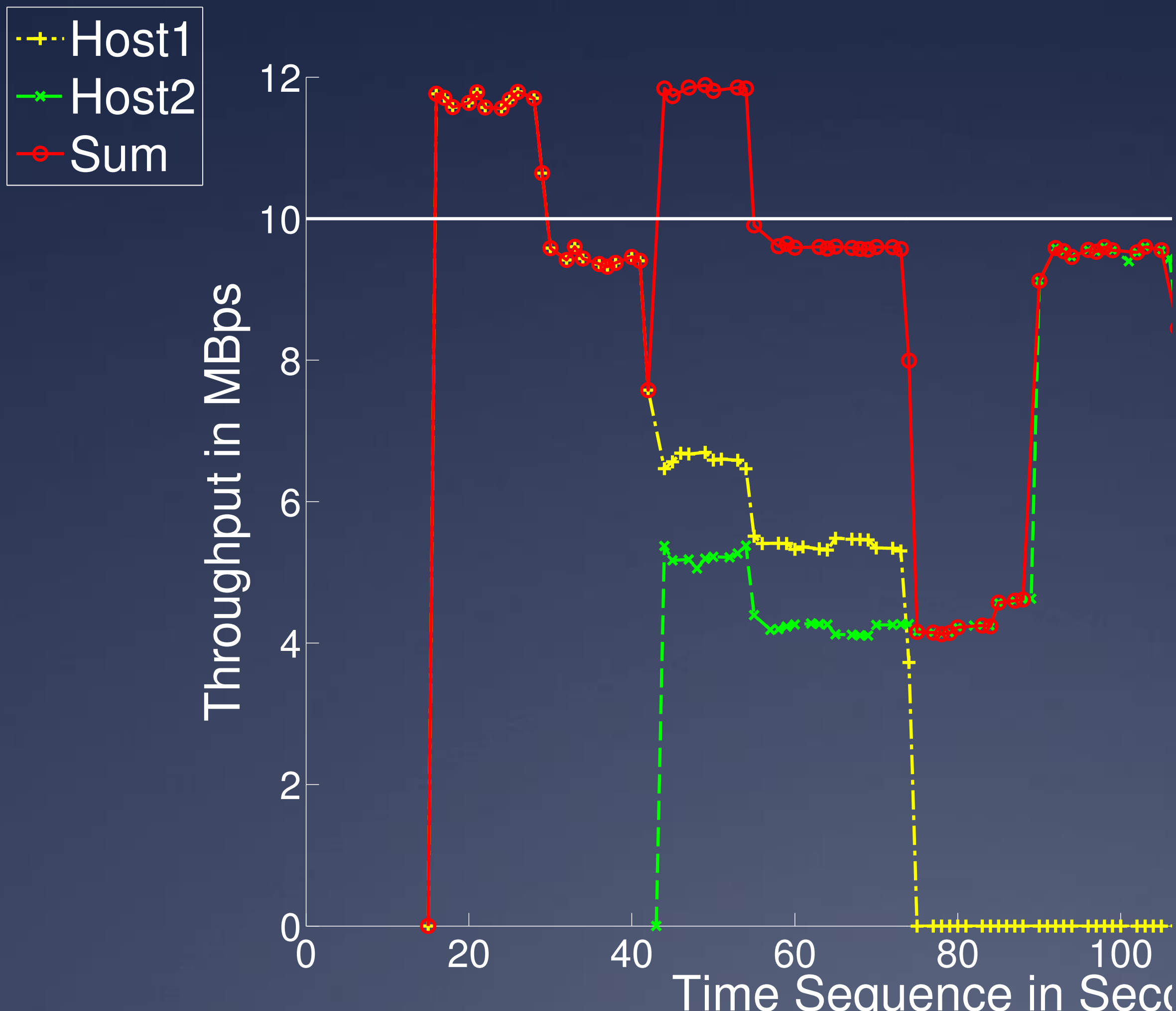
# Distributed Rate Limiting

# Distributed Rate Limiting

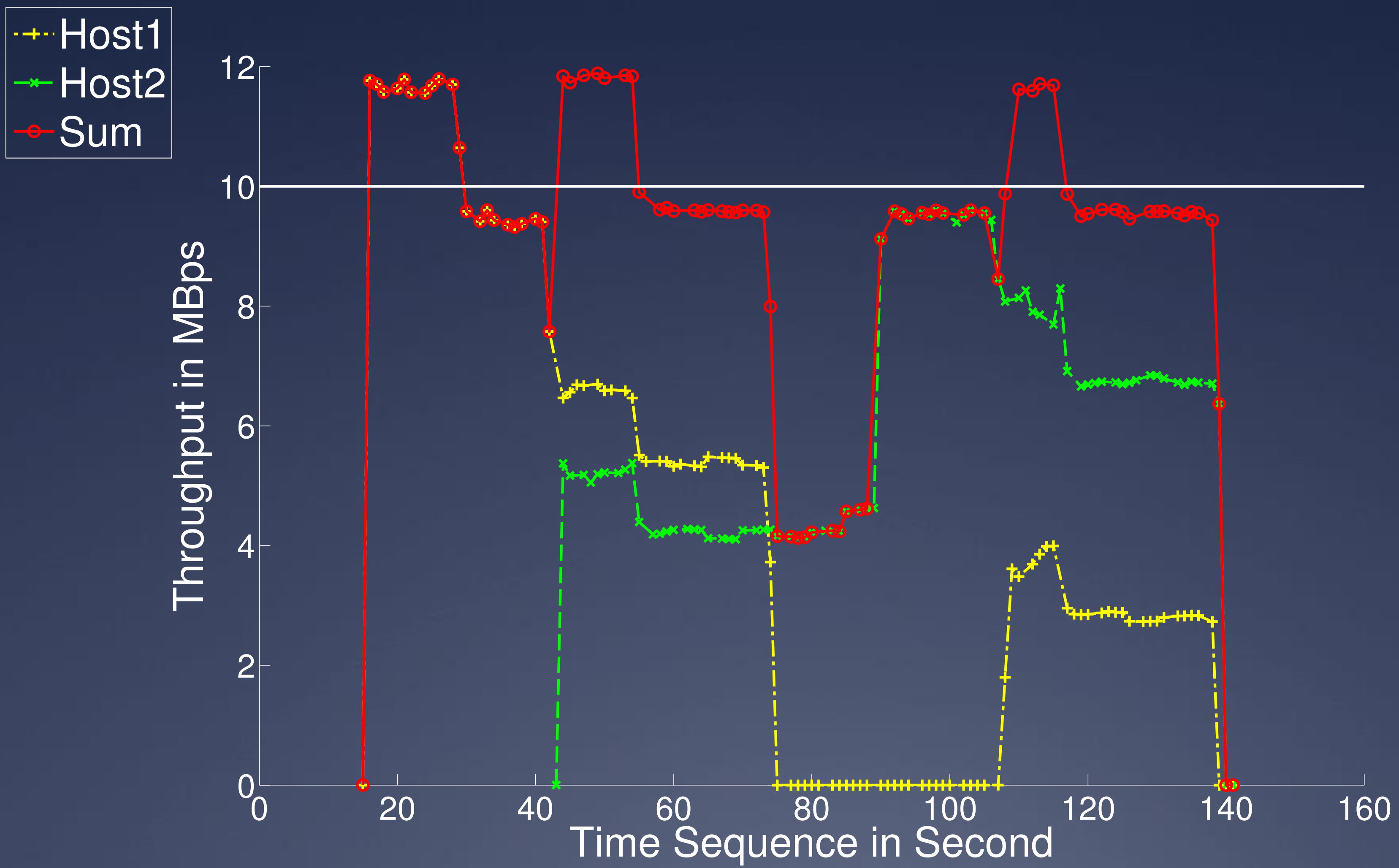
# Distributed Rate Limiting



# Distributed Rate Limiting



# Distributed Rate Limiting

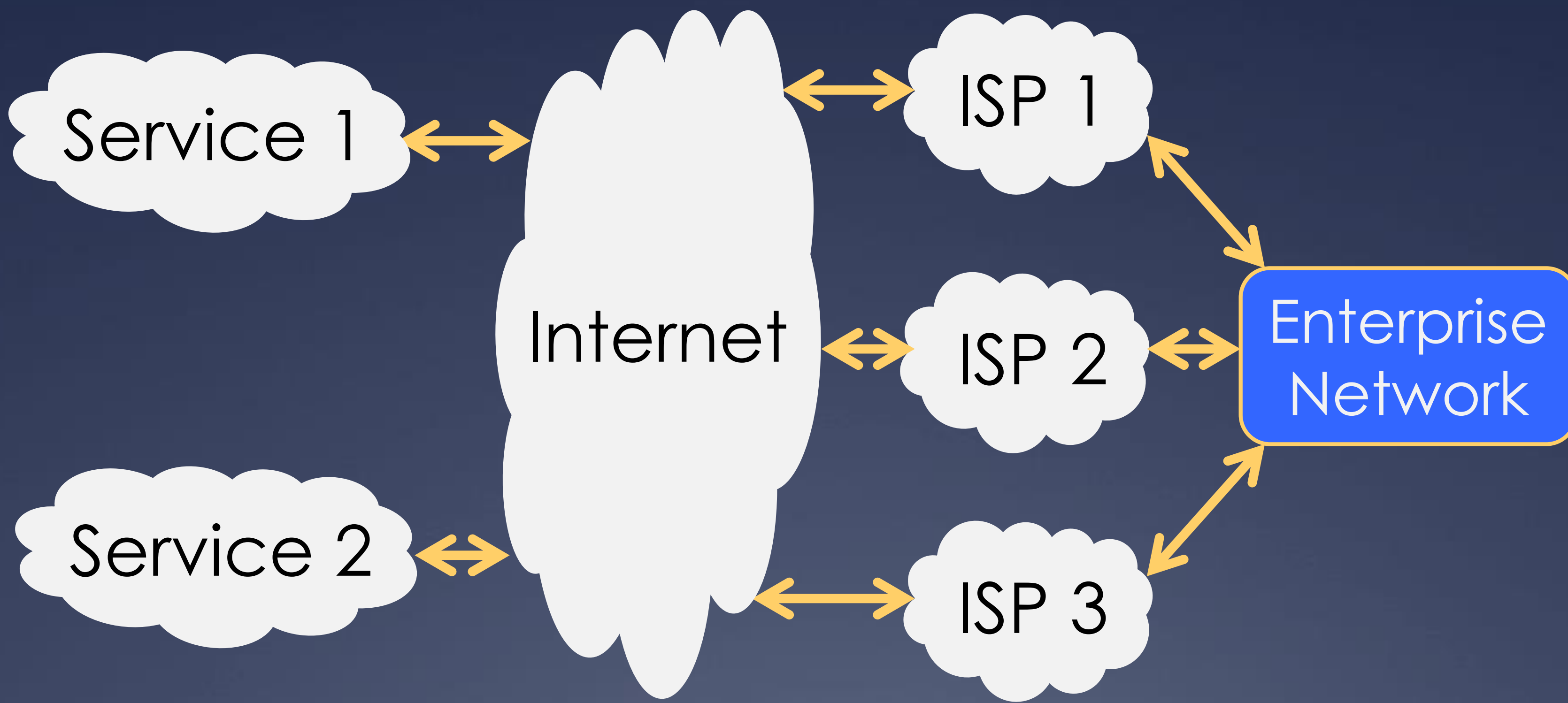


# Hone Summary

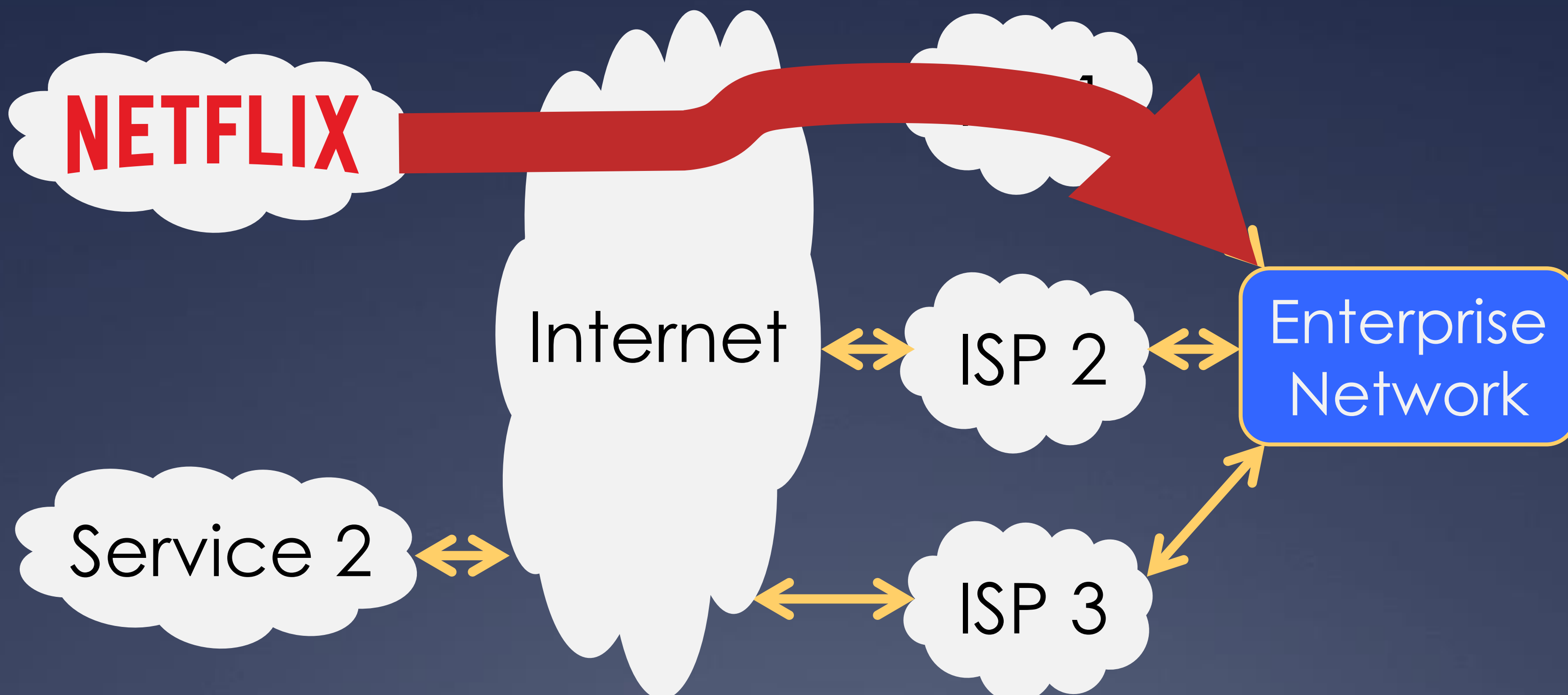
- Programming abstraction
  - Access to fine-grained data in servers
  - Three-stage framework
- Efficient and scalable runtime
- Deployment
  - Integrated into product of Overture for Verizon Business Cloud
- Springer JNSM Volume 23, 2015

# Sprite:
# Bridging Enterprise and ISP for Inbound Traffic Control

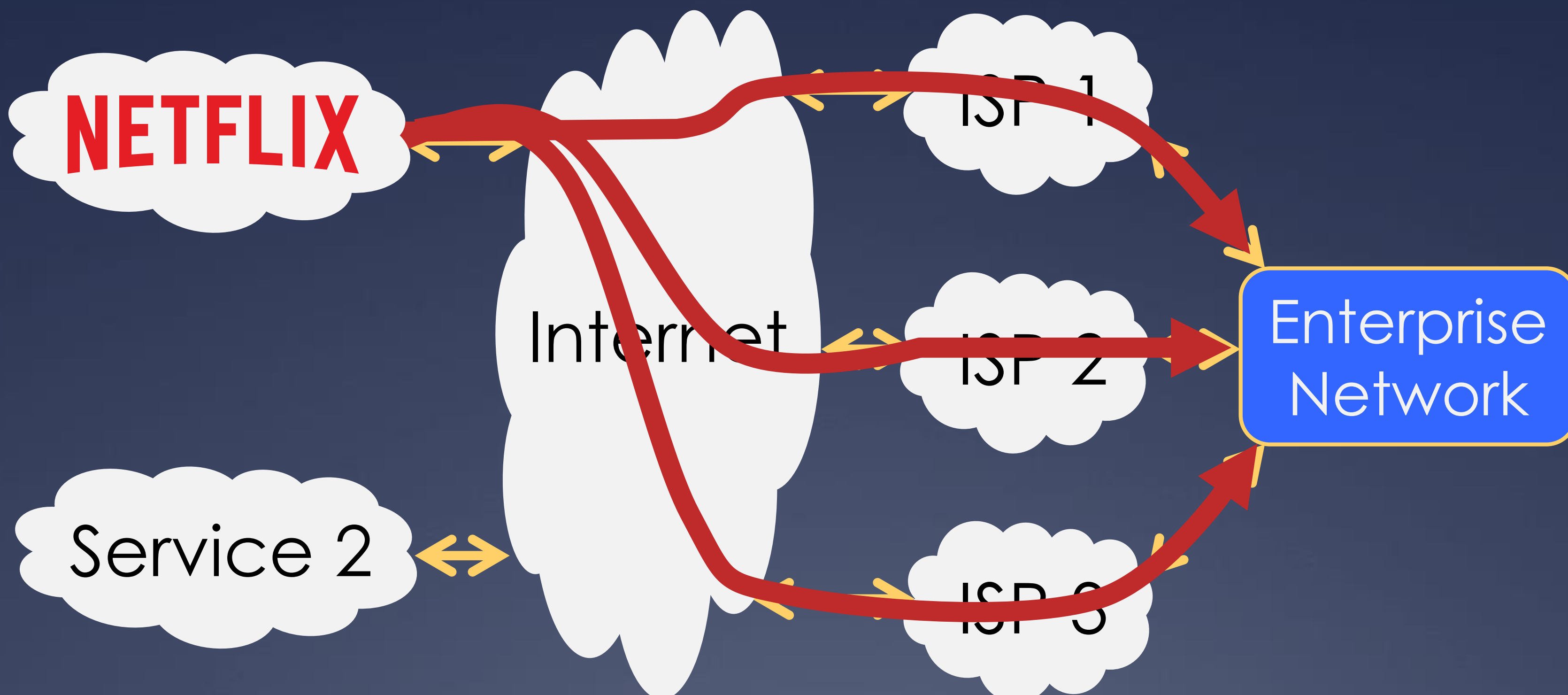
# Problem for Enterprises: Inbound Traffic Engineering



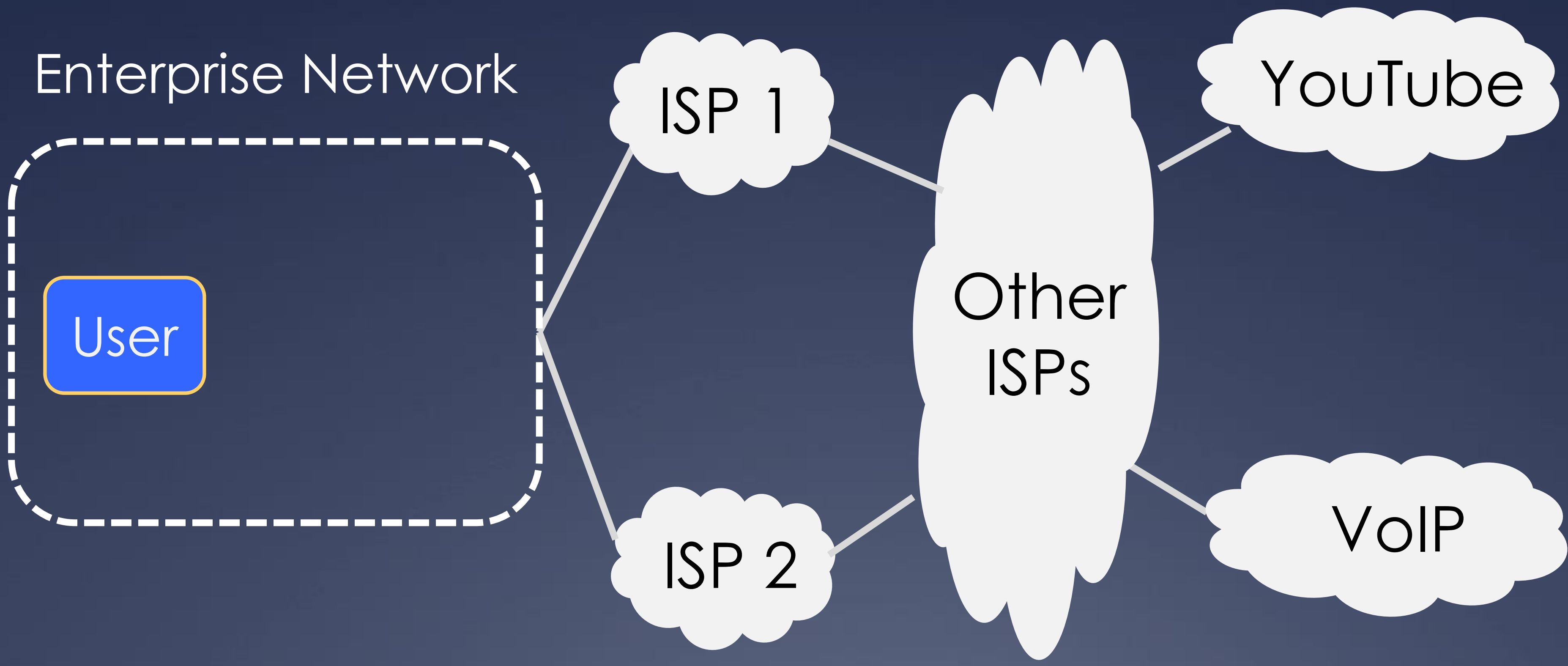
# Problem for Enterprises: Inbound Traffic Engineering



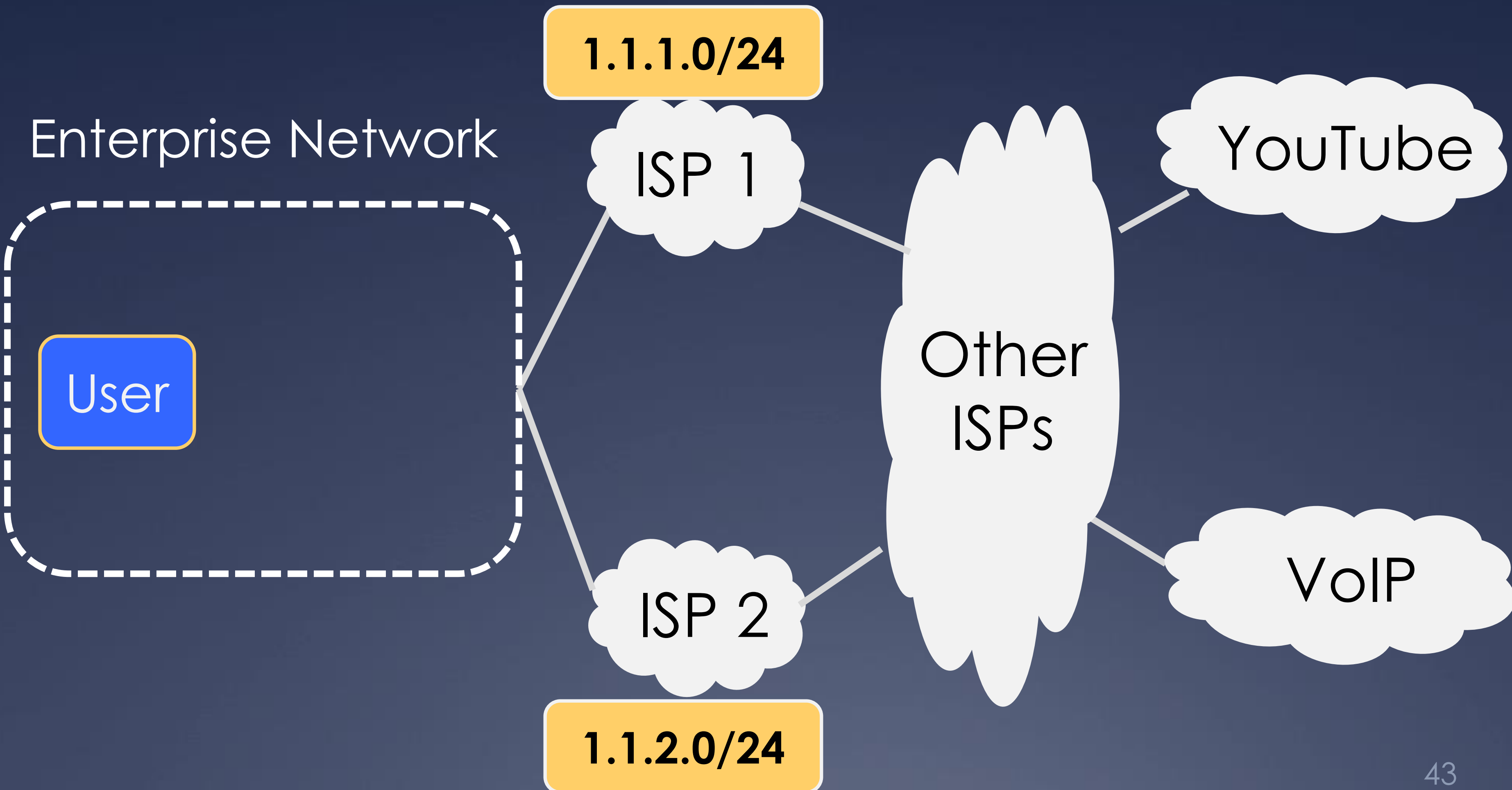
# Problem for Enterprises: Inbound Traffic Engineering



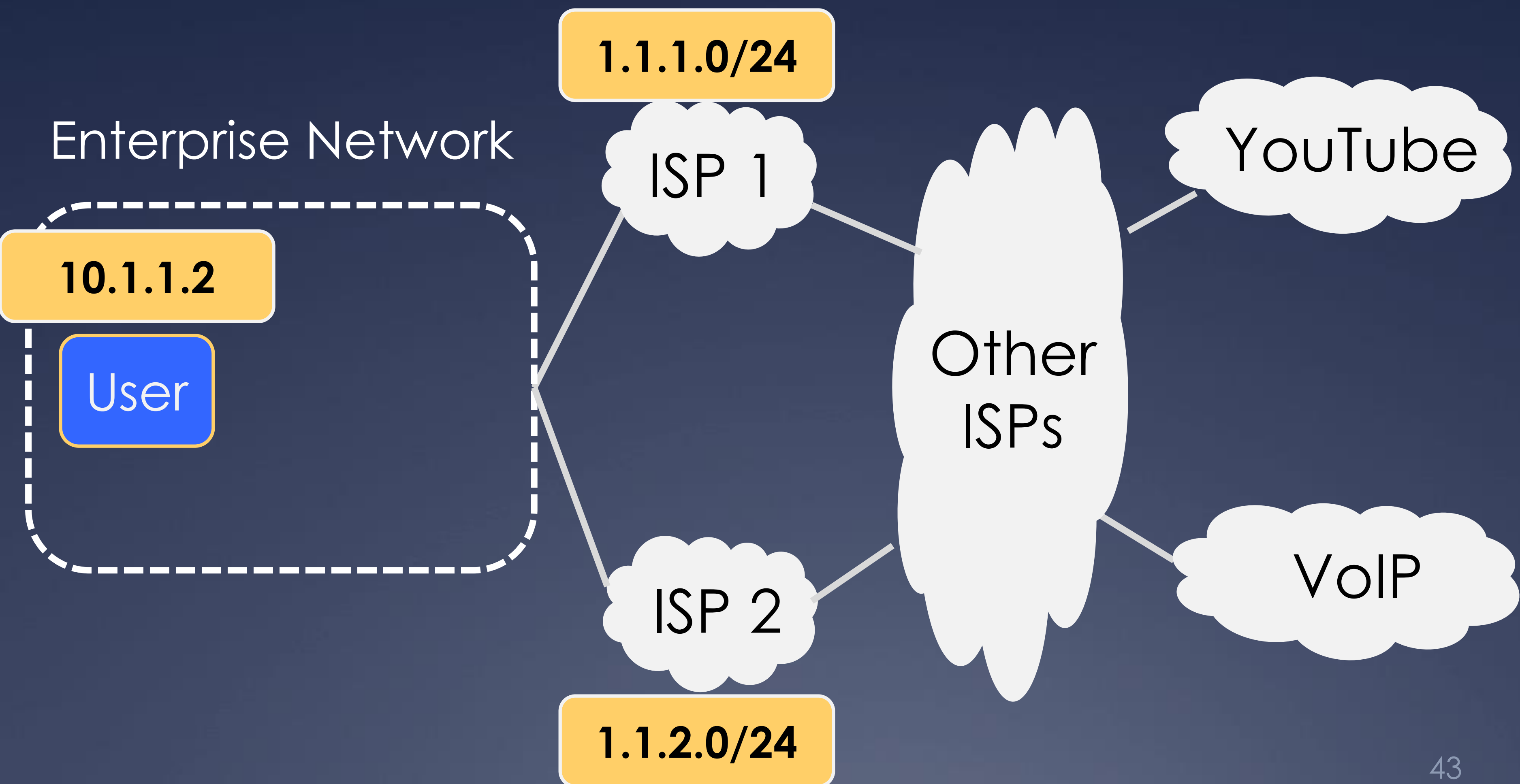
# Mechanism of Sprite



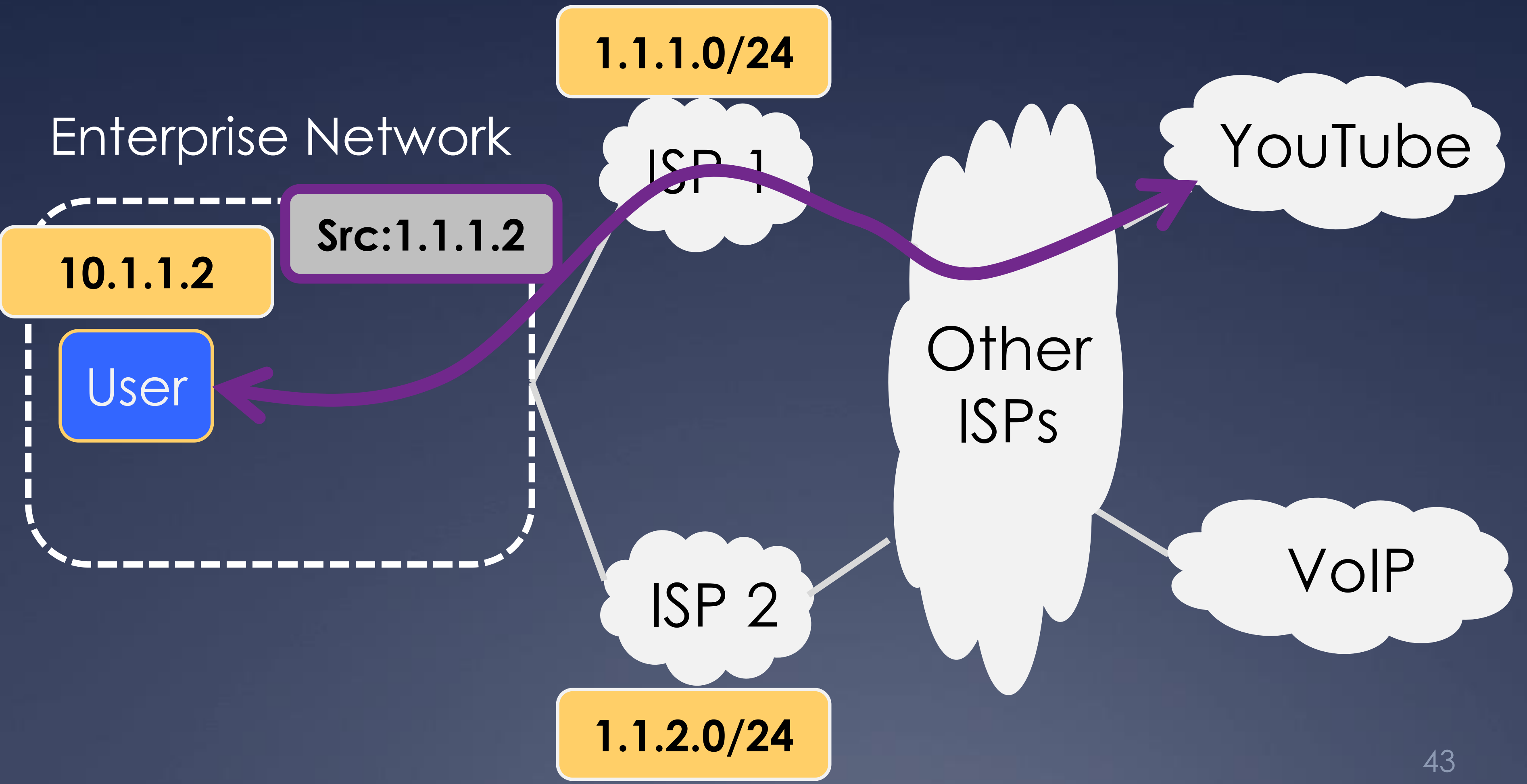
# Mechanism of Sprite



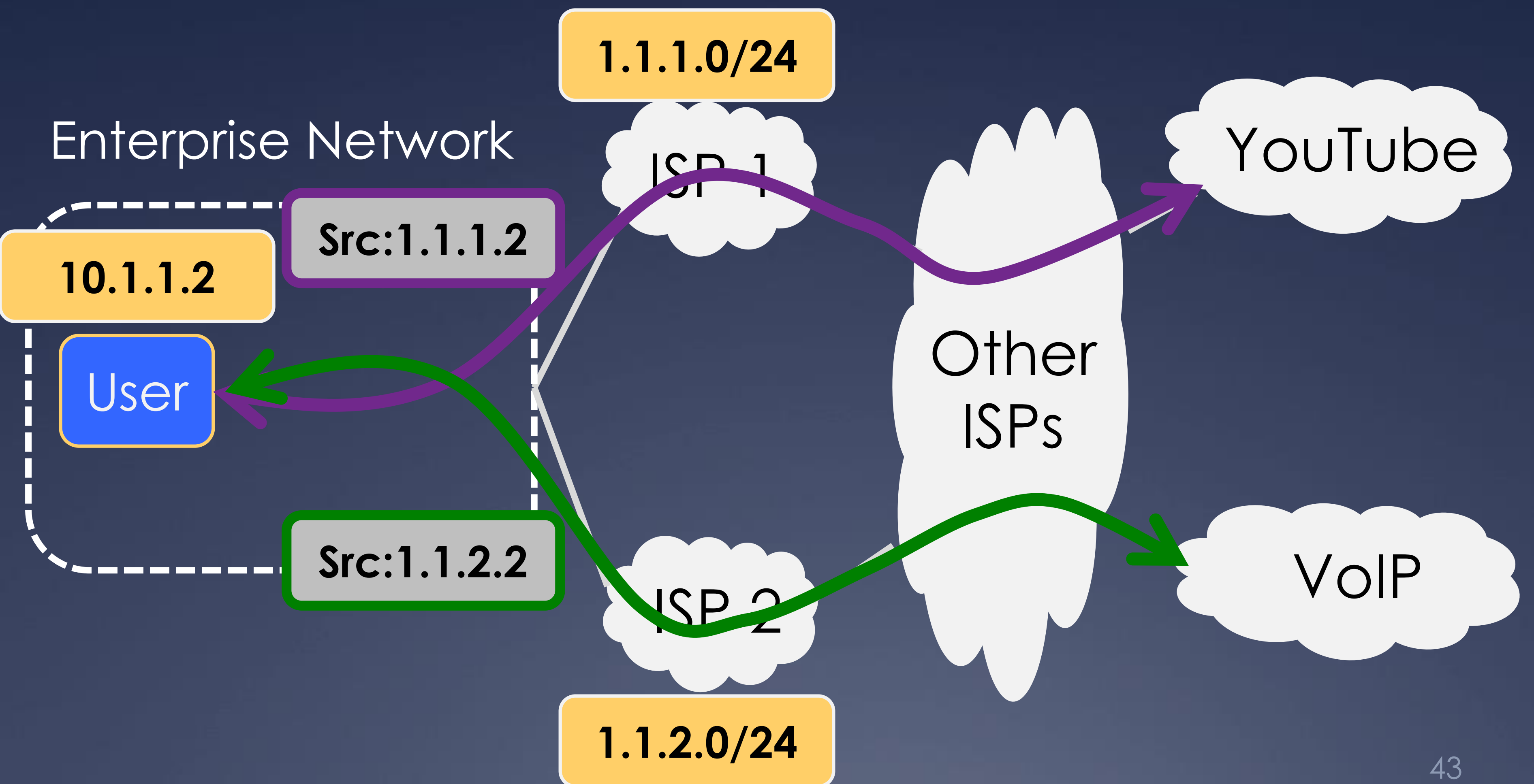
# Mechanism of Sprite



# Mechanism of Sprite



# Mechanism of Sprite



# Challenges

- Naïve solution
  - All done by the border router

# Challenges

- Naïve solution
  - All done by the border router
- Need a distributed solution

# Challenges

- Naïve solution
  - All done by the border router
- Need a distributed solution
  - Control-plane scaling

# Challenges

- Naïve solution
  - All done by the border router
- Need a distributed solution
  - Control-plane scaling
  - Data-plane scaling

# Challenges

- Naïve solution
  - All done by the border router
- Need a distributed solution
  - Control-plane scaling
  - Data-plane scaling
- Need a simple management interface

# Three Tiers of Abstractions

| ***Abstraction*** | ***Example*** |
|---|---|

# Three Tiers of Abstractions

| ***Abstraction*** | ***Example*** |
| --- | --- |
| High-level Policy | |

# Three Tiers of Abstractions

| ***Abstraction*** | ***Example*** |
| --- | --- |
| High-level Policy | <BioDept, Analytic> ➔BestLatency |

# Three Tiers of Abstractions

| ***Abstraction*** | ***Example*** |
|---|---|
| High-level Policy | <BioDept, Analytic> ➔BestLatency |
| ↓ | |
| Network-level Rule | |

# Three Tiers of Abstractions

| ***Abstraction*** | ***Example*** |
| --- | --- |
| High-level Policy | <BioDept, Analytic> ➔BestLatency |
| Network-level Rule | <10.1.0.0/16, 184.168.1.0/24> ➔ISP1 |

# Three Tiers of Abstractions

Control plane
Global sync

| ***Abstraction*** | ***Example*** |
|---|---|
| High-level Policy | <BioDept, Analytic> ➔BestLatency |
| Network-level Rule | <10.1.0.0/16, 184.168.1.0/24> ➔ISP1 |

# Three Tiers of Abstractions

| *Abstraction* | *Example* |
| --- | --- |
| High-level Policy | <BioDept, Analytic> ➔BestLatency |
| Network-level Rule | <10.1.0.0/16, 184.168.1.0/24> ➔ISP1 |
| Per-flow Rule | |

Control plane
Global sync

Data plane
Local only

# Three Tiers of Abstractions

| Abstraction | Example |
| --- | --- |
| High-level Policy | <BioDept, Analytic> ➔BestLatency |
| Network-level Rule | <10.1.0.0/16, 184.168.1.0/24> ➔ISP1 |
| Per-flow Rule | <10.1.1.42,184.168.1.17> SNAT to 1.1.1.2 |

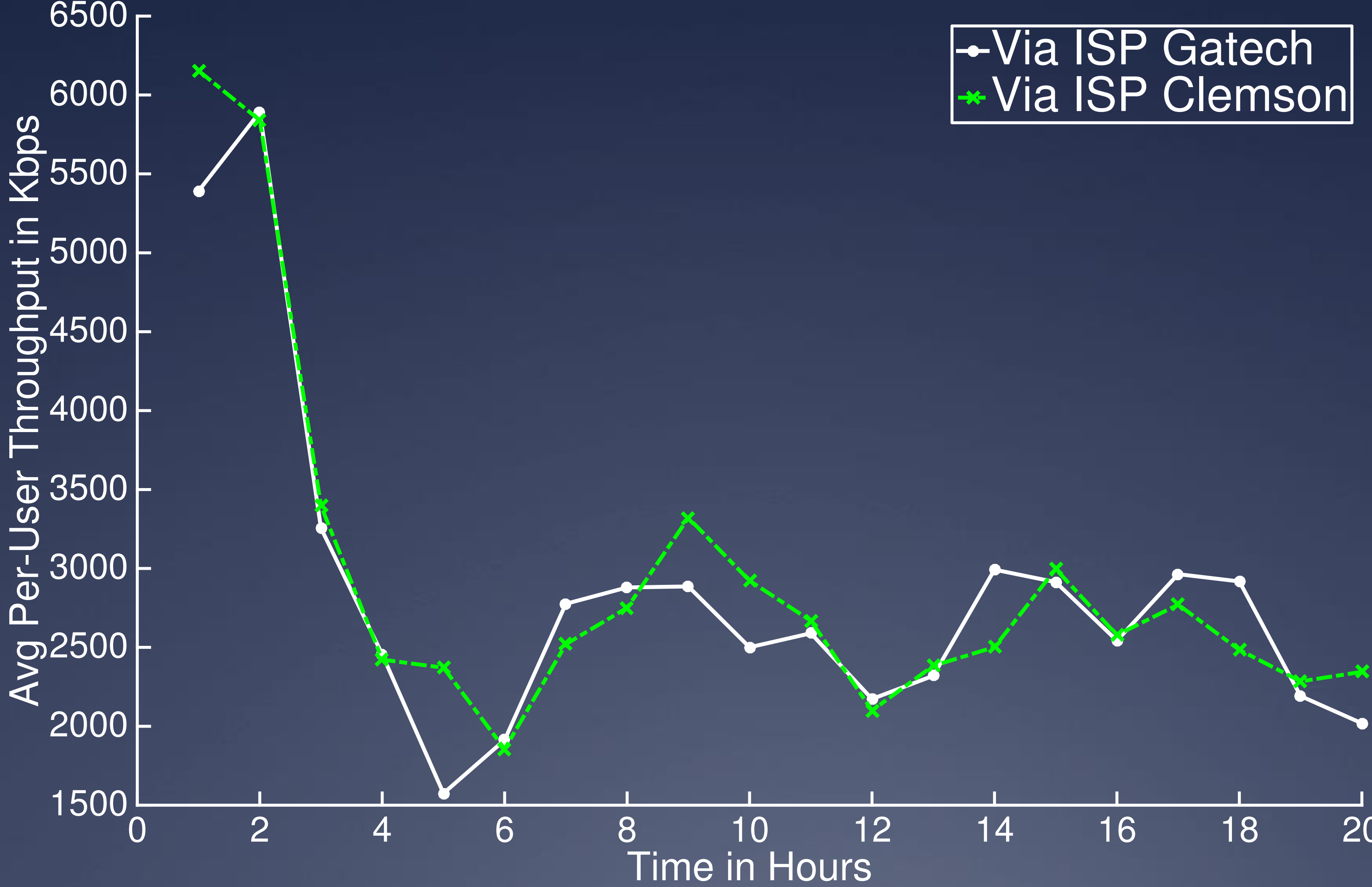
Control plane
Global sync

Data plane
Local only

# Case: Dynamic Perf-driven Balancing

- Move users' connections among ISPs for best per-user performance
- Live Internet experiment on AWS VPC and PEERING
  - 10 users watch movies on YouTube

# User YouTube Throughput



# Sprite Summary

- Direct and fine-grained inbound traffic control
- Scalable solution
  - Scaling control and data planes
- Evaluation
  - In collaboration with OIT
- SOSR'15

# Open Issue #1
# Statesman + Hone

- Merge Hone into Statesman
  - *Joint server, traffic, infrastructure mgmt.*
- Possible exploration
  - *State abstraction for server data*
  - *Dependency of server and network states*
  - *Safety invariants involving servers*

# Open Issue #2
# Transactional Statesman

- Transactional semantics for conflict resolution
- Possible exploration
  - *Grouping semantics*
  - *Condition semantics*

# Open Issue #3
# Hone for Multi-tenant Cloud

- Loosen the assumption of having access to the hosts' OS
- Possible exploration
  - *Infer stats in guest OS from data in hypervisor*

# Related work

# Statesman Related Work

| Work | What they do | Statesman |
|---|---|---|
| SDN works | Centralized control of flows | Wider spectrum of management applications |
| Onix | Single repository of network stats | Three-view network state model |
| Pyretic | Target at flow management | Wider spectrum of applications |
| Corybantic | Tight cross-application proposal evaluation | Loose coupling |
| FlowVisor | Virtual topology slicing | Network state model |

# Hone Related Work

| Work | What they do | Hone |
|---|---|---|
| Network Exception Handler | Use hosts only as software switches | Go deeper into socket layer |
| Gigascope | Extend SQL | Use functional language to construct |
| Chimera | Specific application (intrusion detection) | Support various management solutions |
| MapReduce | Naturally parallelizable data | Data inherently associated with collection point |

# Sprite Related Work

| Work | What they do | Sprite |
|---|---|---|
| BGP AS-path prepending | Tune BGP configurations to influence neighboring ASes | Direct control from inside enterprise networks |
| Works on Internet re-architecture | Clean-slate design in routing or hosts | Incrementally deployable |
| Tunnel-based works | Two ends cooperate to control traffic | Only need the client side to act |

# Thanks!

# Q&A