SK telecom

Networking Intelligence with SDN

Sangho Shin, Ph.D

Corporate R&D Center

Outline

1. SDN to SDN
2. Use Cases in CORD
   - R/M/E CORD
   - A-CORD
3. Use Cases in SKT
   - Operation Intelligence: T-CORE
   - Network Intelligence: SONA & TINA

SDN to SDN

SDN to Self Driving Network

Reference: Prof. Guru Parulkar's talk @ Standard University Workshop

SDN to SDN

Software Defined Networks

So developers can write programs to control, configure, and sense networks

Self Driving Networks

Leverage ML to auto-generate programs to control and configure networks

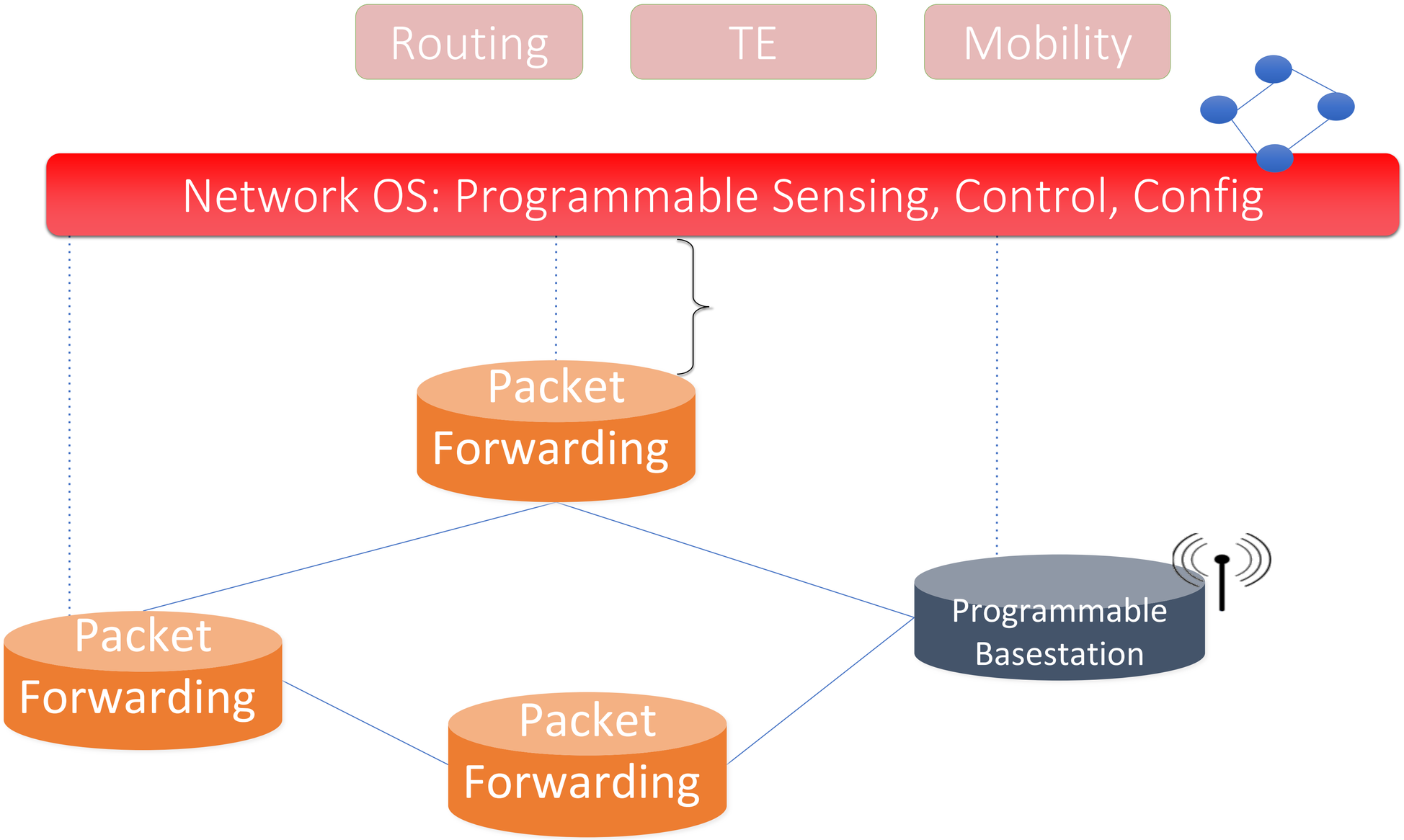
Benefits

Much faster and accurate response to network conditions

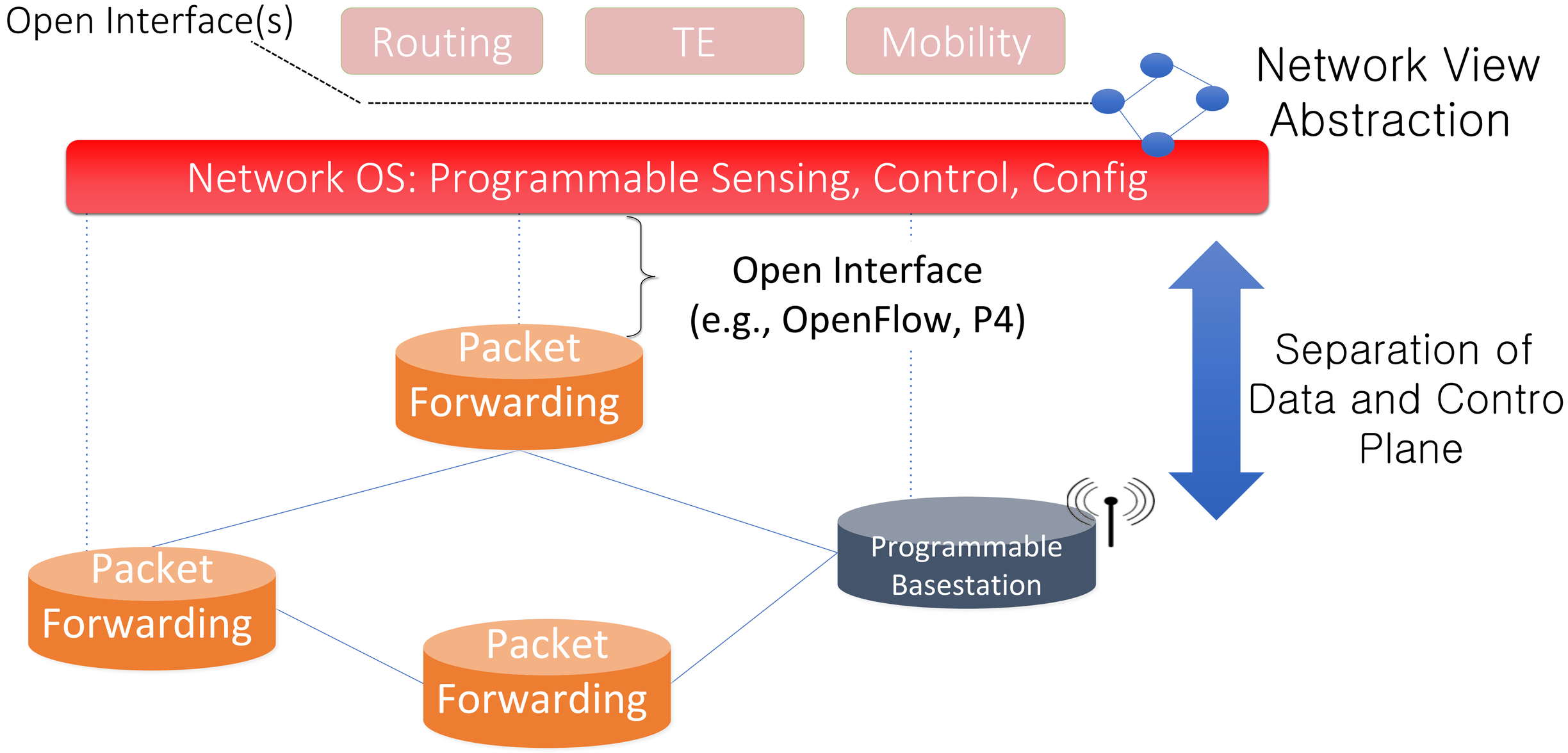
Customization for subscribers, apps, flows
Maximize resource utilization
Faster response to failures

…

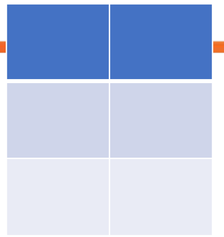
What is SDN?


Routing
TE
Mobility
Network OS: Programmable Sensing, Control, Config
Packet Forwarding
Packet Forwarding
Packet Forwarding
Programmable Basestation

What is SDN?


Open Interface(s)
Routing
TE
Mobility
Network View Abstraction
Network OS: Programmable Sensing, Control, Config
Open Interface
(e.g., OpenFlow, P4)
Separation of Data and Control Plane
Packet Forwarding
Packet Forwarding
Packet Forwarding
Programmable Basestation

What is SDN?

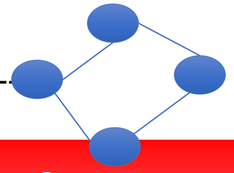


Open Interface(s)

Routing

TE

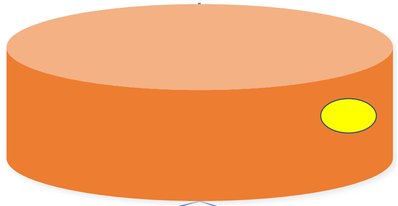
Mobility



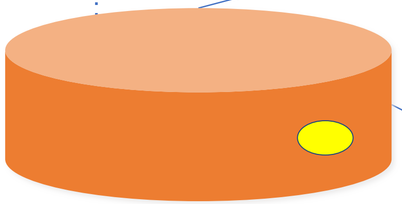
Network View Abstraction

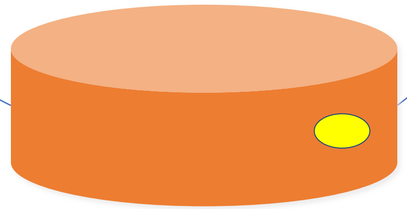
Network OS: Programmable Sensing, Control, Config

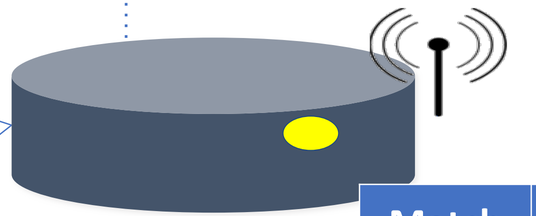
OpenFlow, P4



Programmable Probes



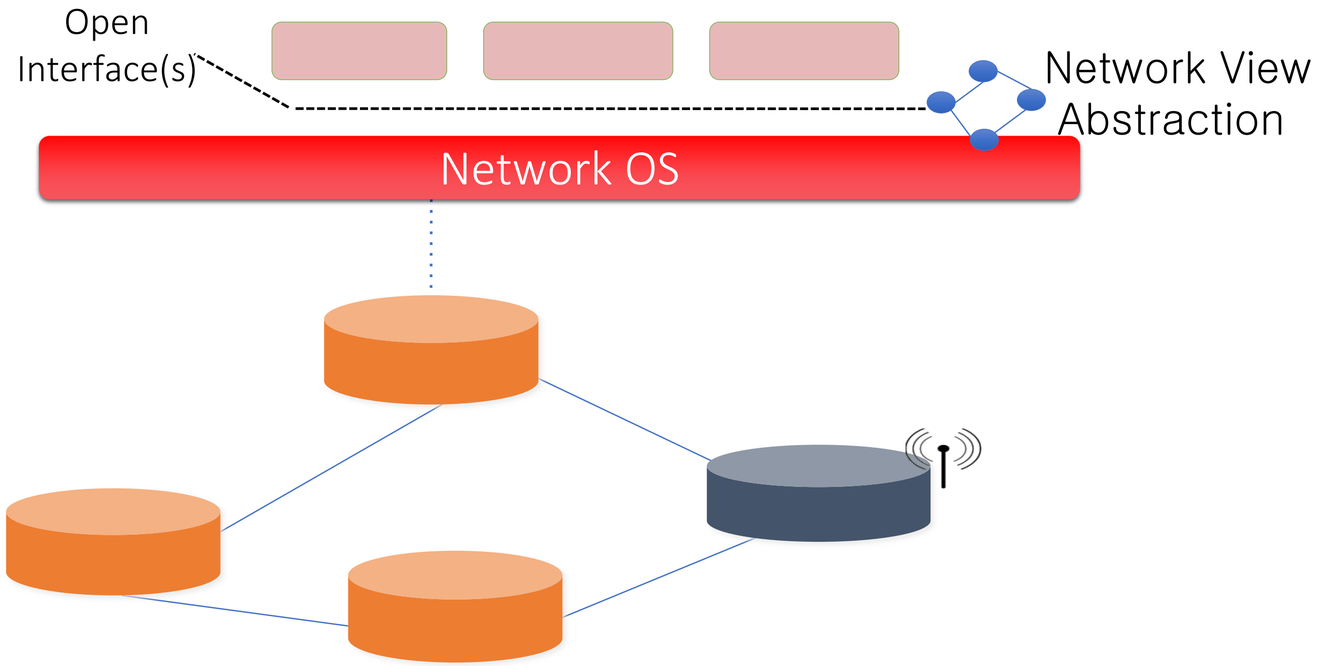




Match	Action

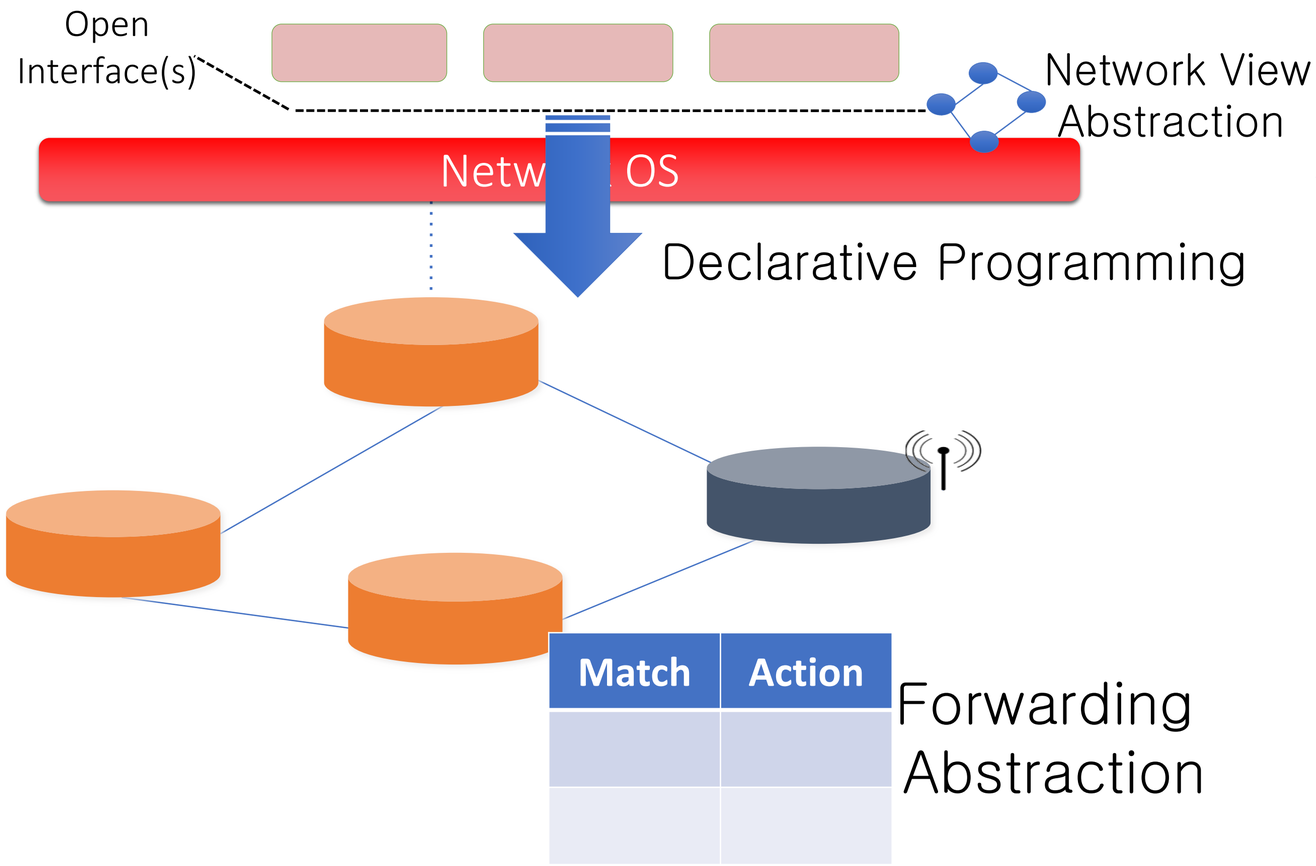
Forwarding Abstraction

How SDN helps?



Makes it easy to write control, config, sensing apps: locally manipulate graph

How SDN helps?

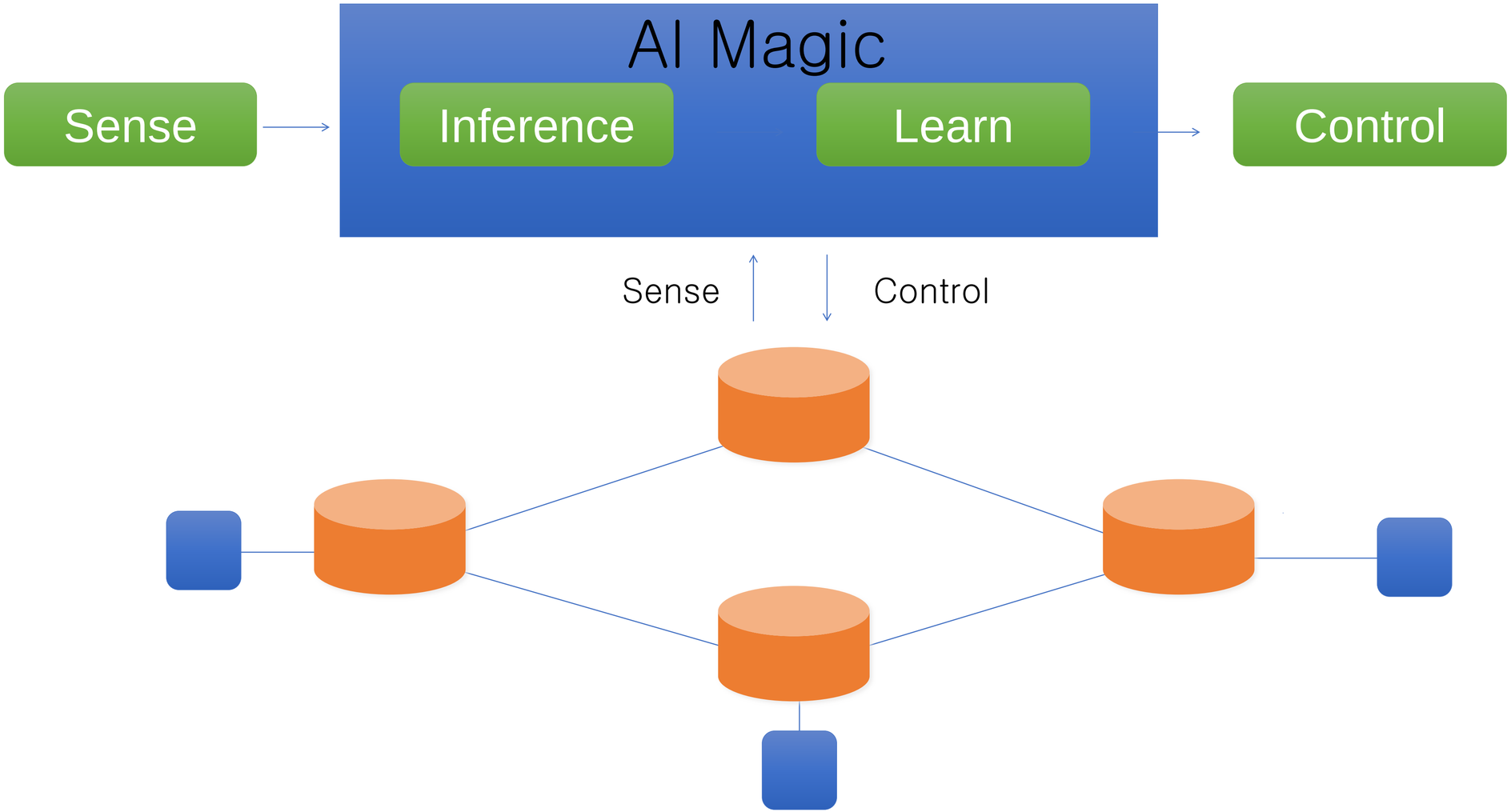


Makes it easy to write control, config, sensing apps: locally manipulate graph

Makes it easy to reason about network and control programs

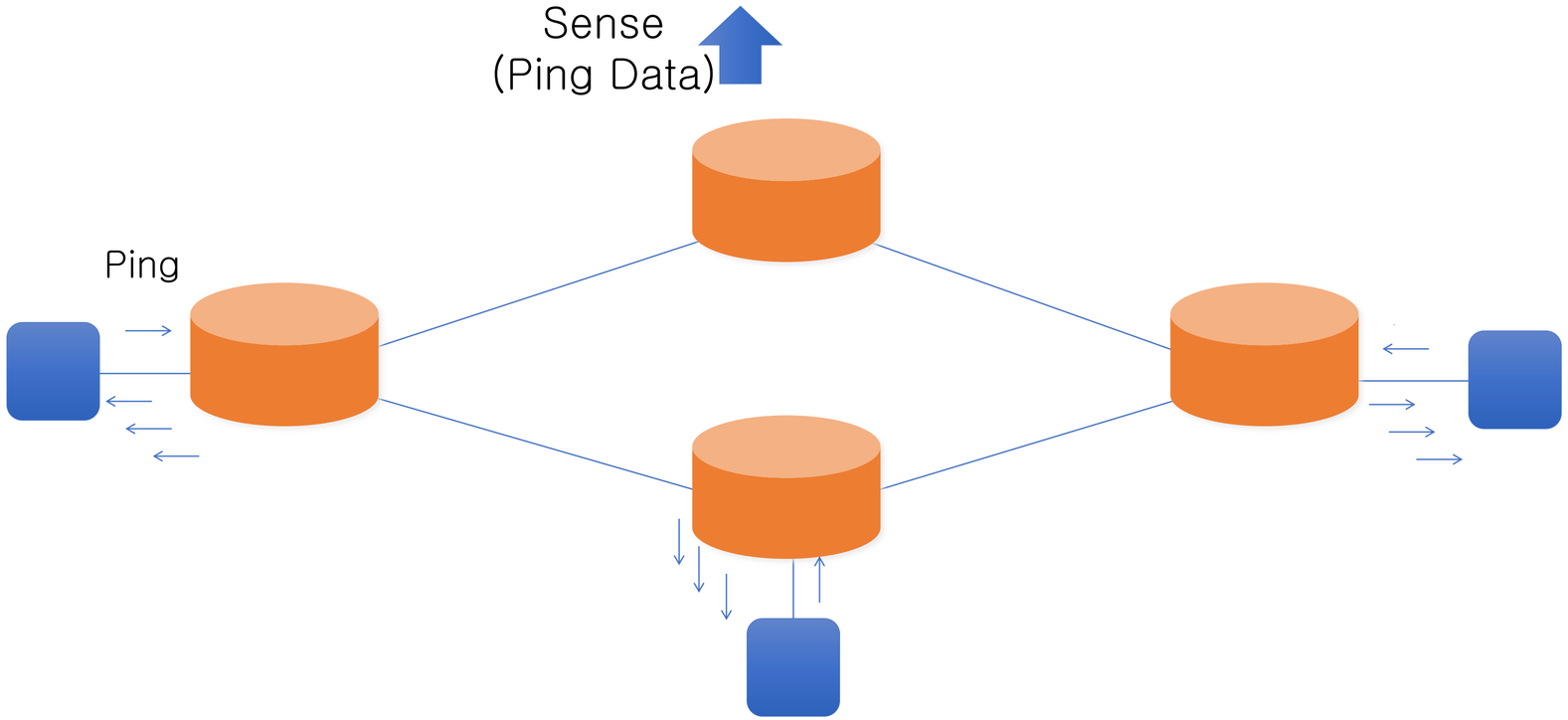
Balaji's Self Driving Networks:

Networking + Machine Learning (AI)



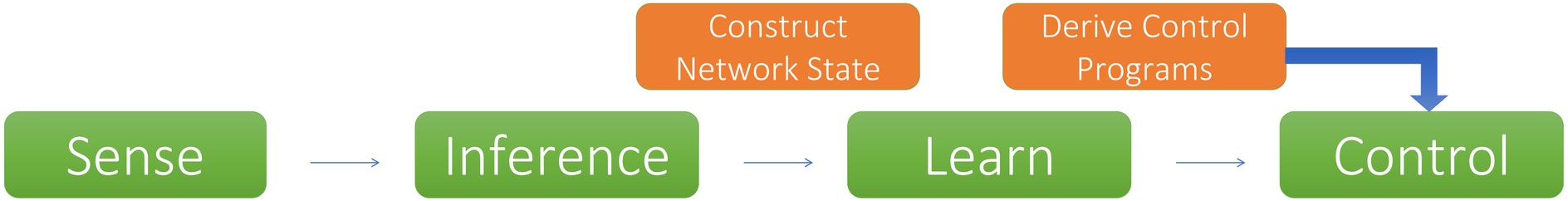
Balaji's Self Driving Networks:

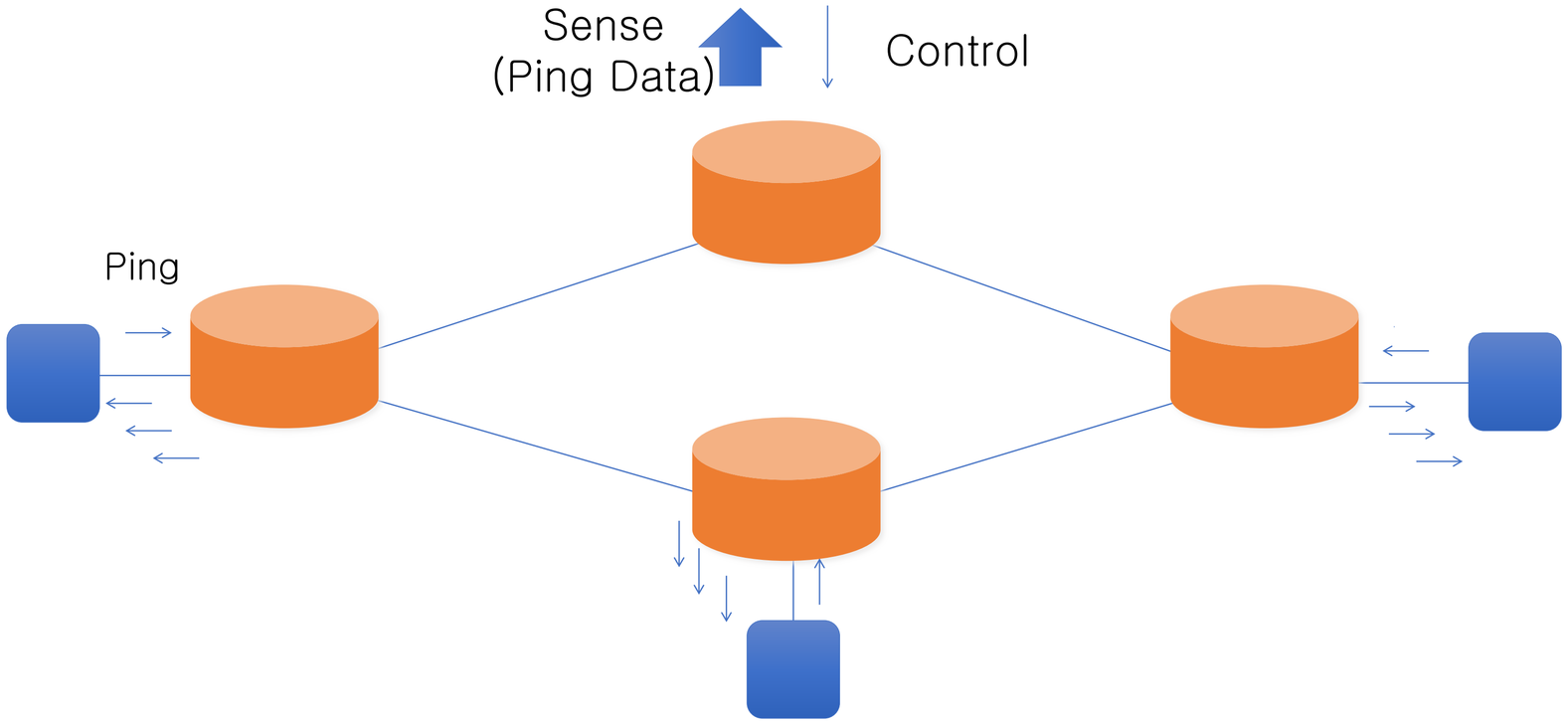
Networking + Machine Learning (AI)



Balaji's Self Driving Networks:

Networking + Machine Learning (AI)



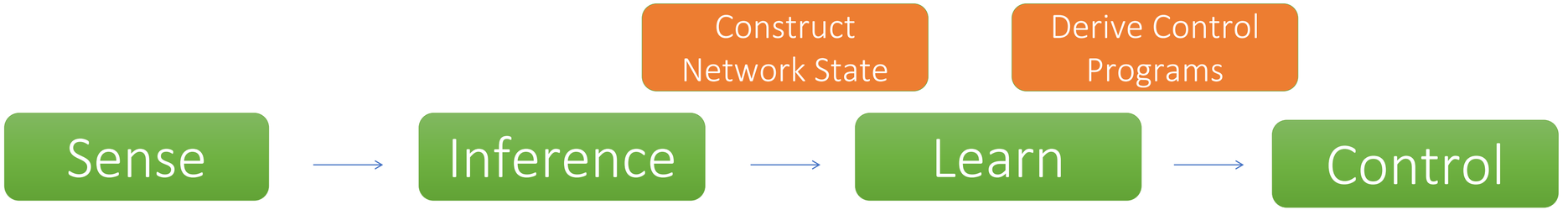


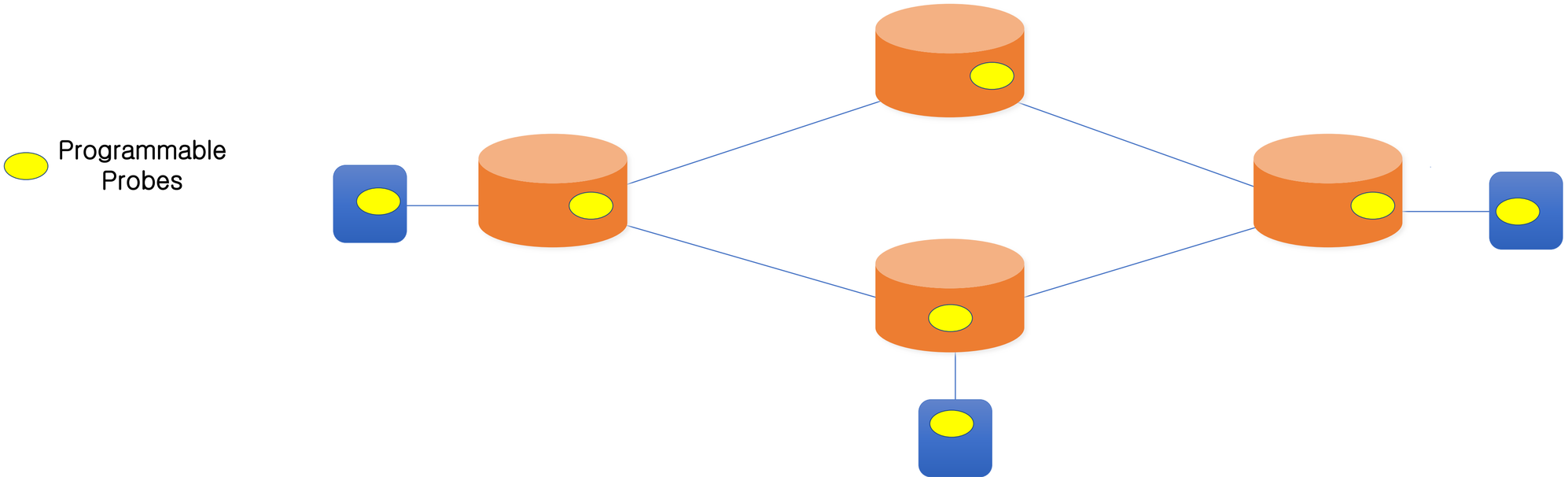
Self Driving Networks Enabled by Software Defined Networks

Software defined networks make it much easier and efficient to do sensing and control

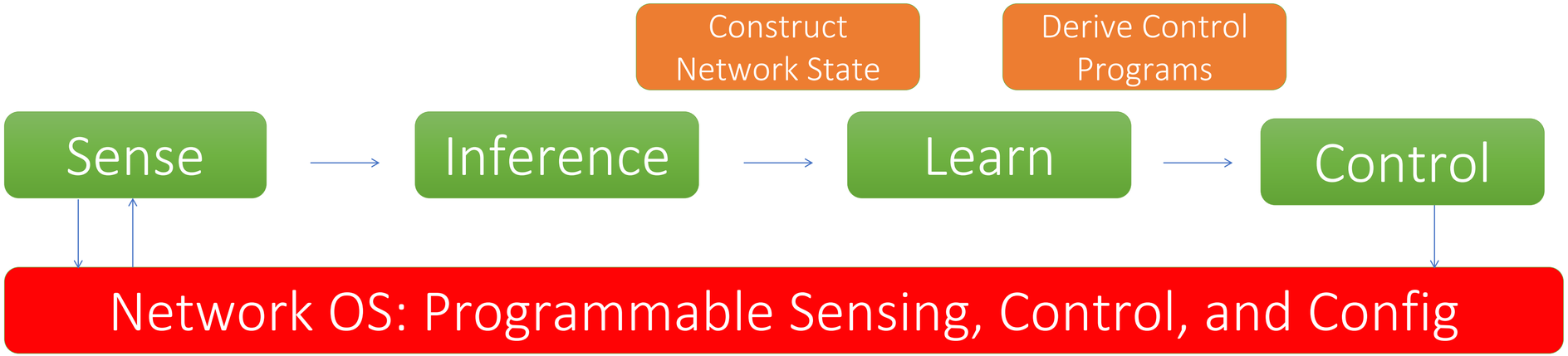
Two critical parts of the self driving networks

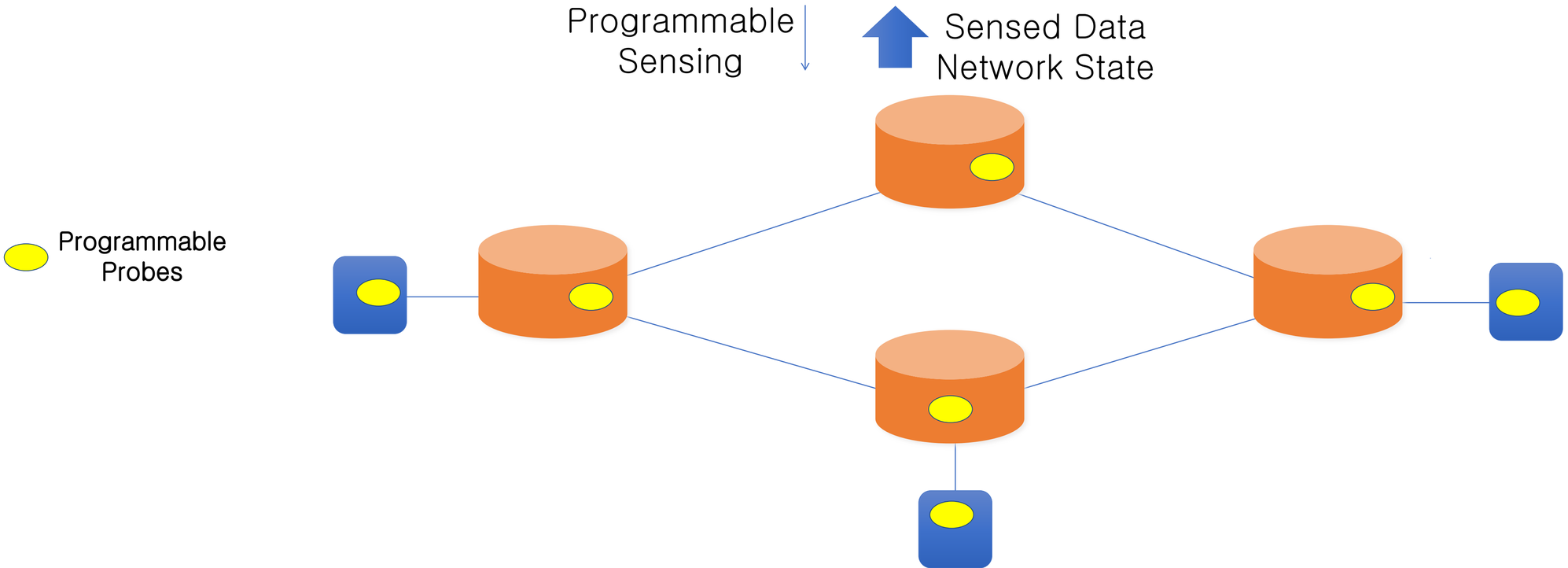
Self Driving Networks Enabled by Software Defined Networks



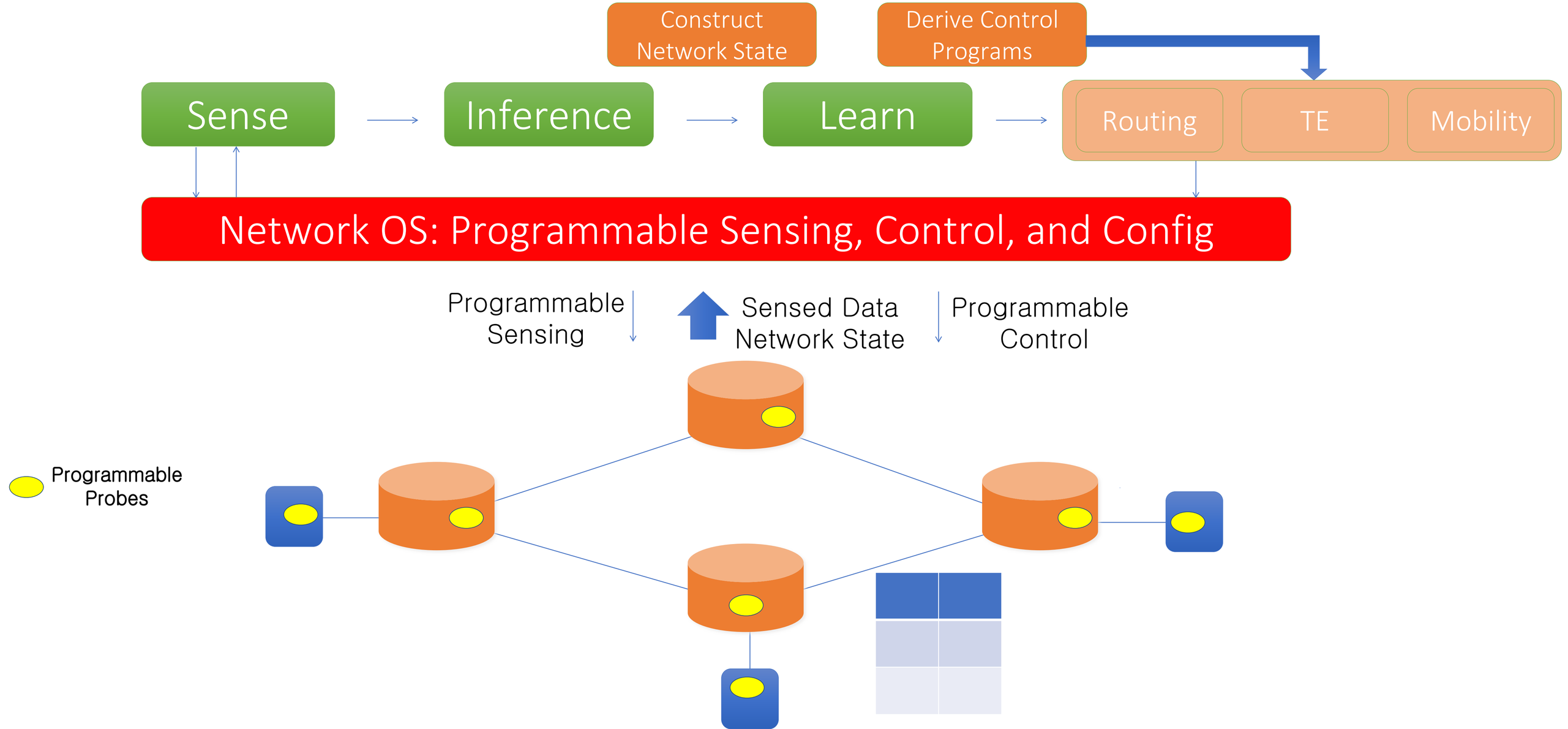


Self Driving Networks Enabled by Software Defined Networks

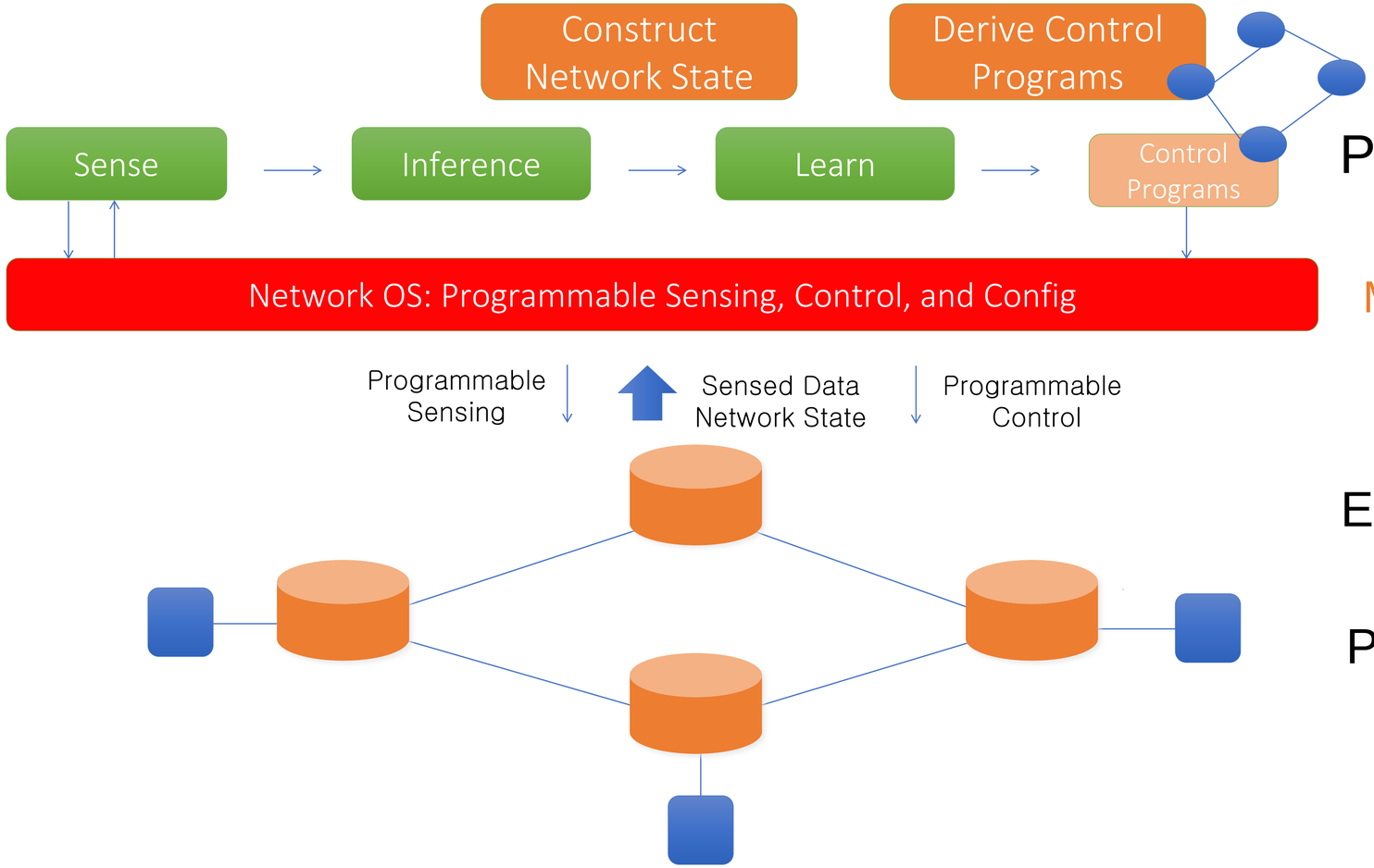




Self Driving Networks Enabled by Software Defined Networks



Self Driving Networks Enabled by Software Defined Networks



Hypothesis

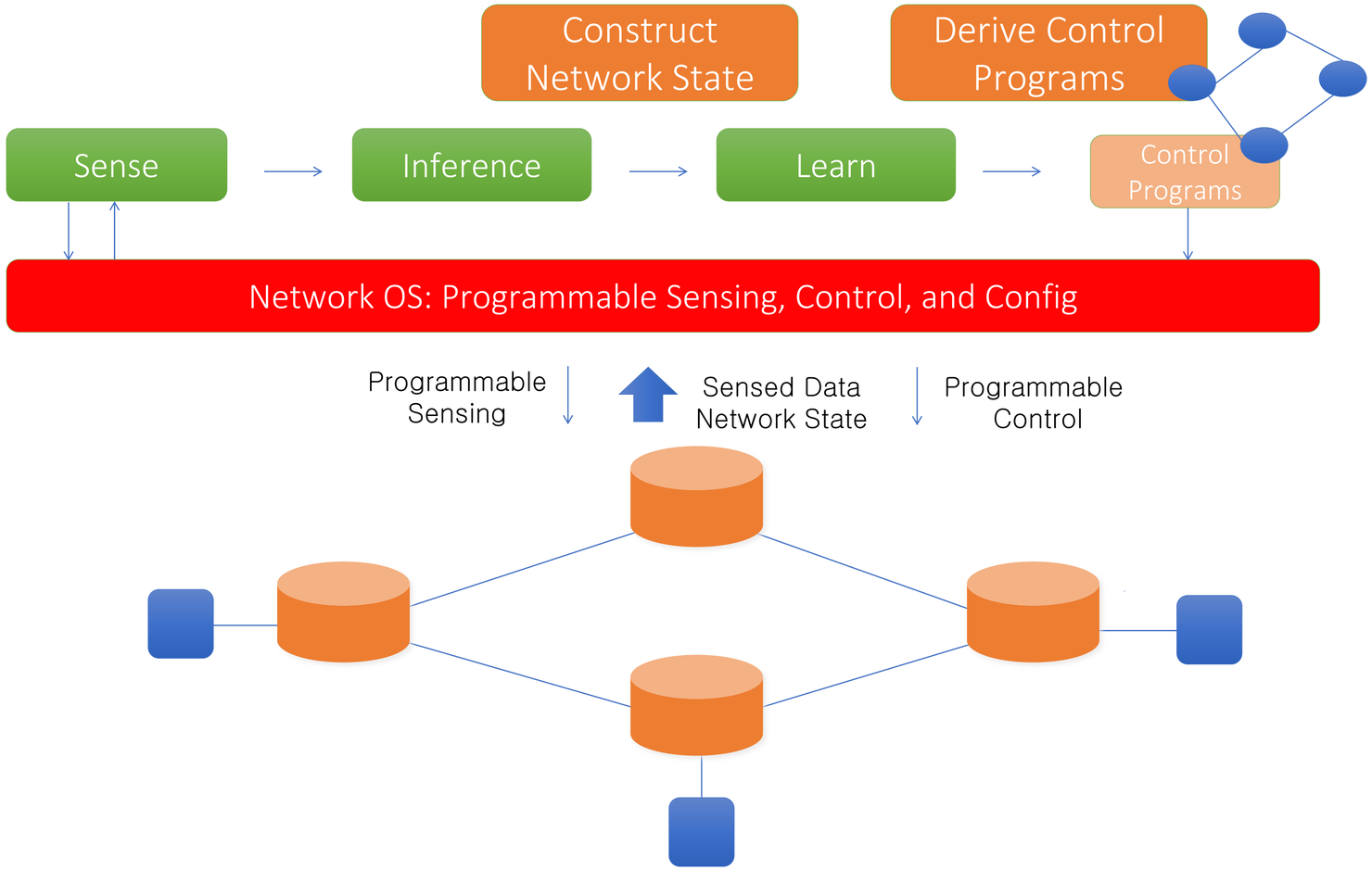
Programmable sensing makes reconstruction easier & faster

More/customized data helps ML algorithms to be more effective

E.g. ping data + queue statistics is better than just ping data

Ping data + queue statistics per network slice is much better than just ping data

Self Driving Networks Enabled by Software Defined Networks



Hypothesis

Deriving network control for SDN is much much easier & faster than for legacy networks

E.g. deriving Dijkstra's algorithm for routing on a network graph vs OSPF on a legacy network

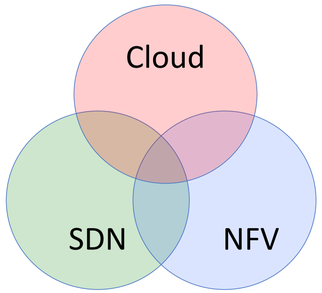
(Half) Use Case in CORD

CORD (Central Office Re-architectured as Data center)

Large number of COs

Evolved over 40-50 years

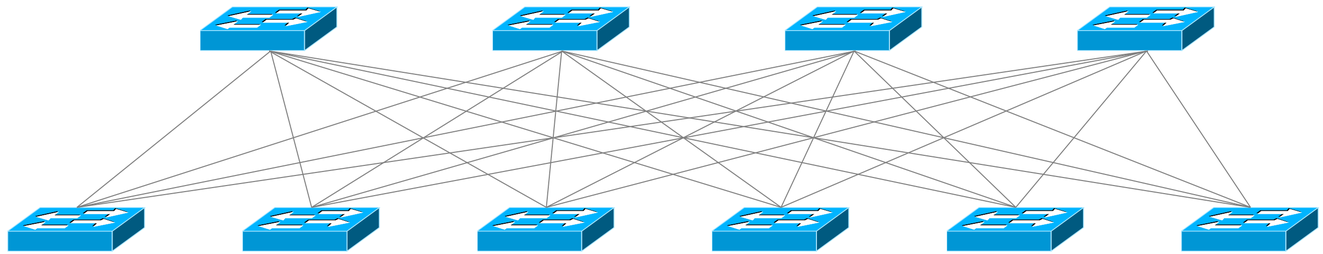
300+ Types of equipment
Huge source of CAPEX/OPEX



SDN Control Plane - ONOS

VNF/Services Control Plane – XOS

SDN enabled Central Office Leaf-Spine Fabric



Commodity servers, switches, and network access

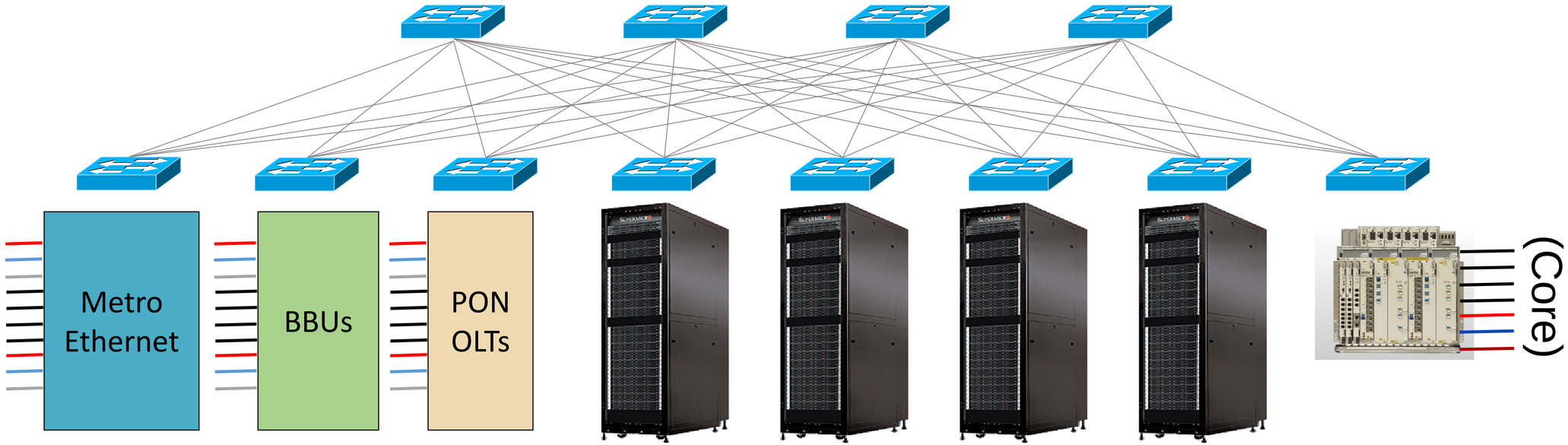
CORD domains of use

Residential
vOLT, vSG,
vRouter, vCDN

Mobile
vBBU. vMME,
vSGW, vPGW,
vCDN

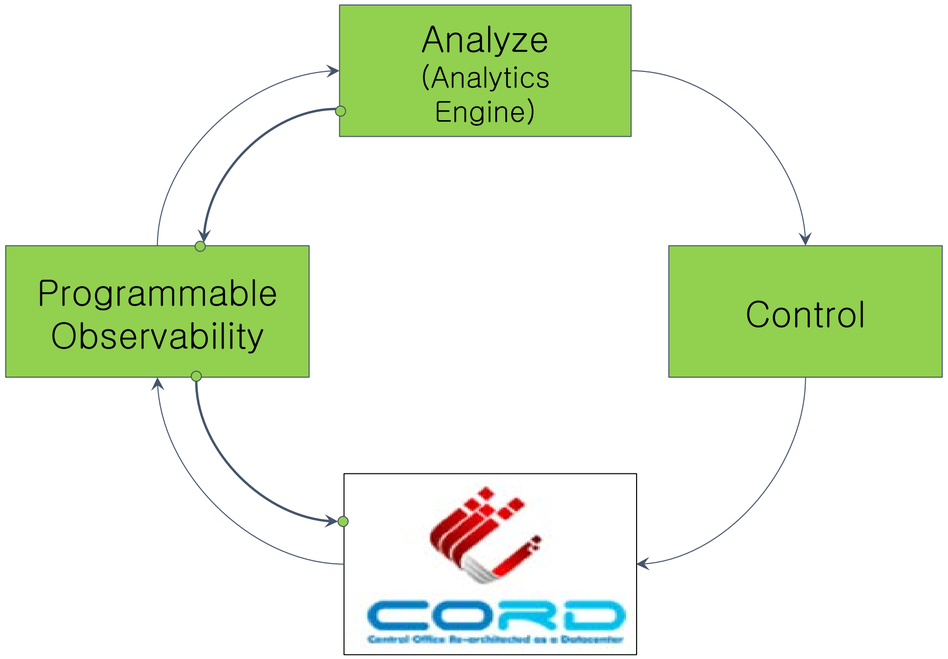
Enterprise
vCarrierEthernet,
vOAM, vWanEx,
vIDS

CORD Platform



CORD Goals

- Enable programmable observability & closed loop control based on analytics



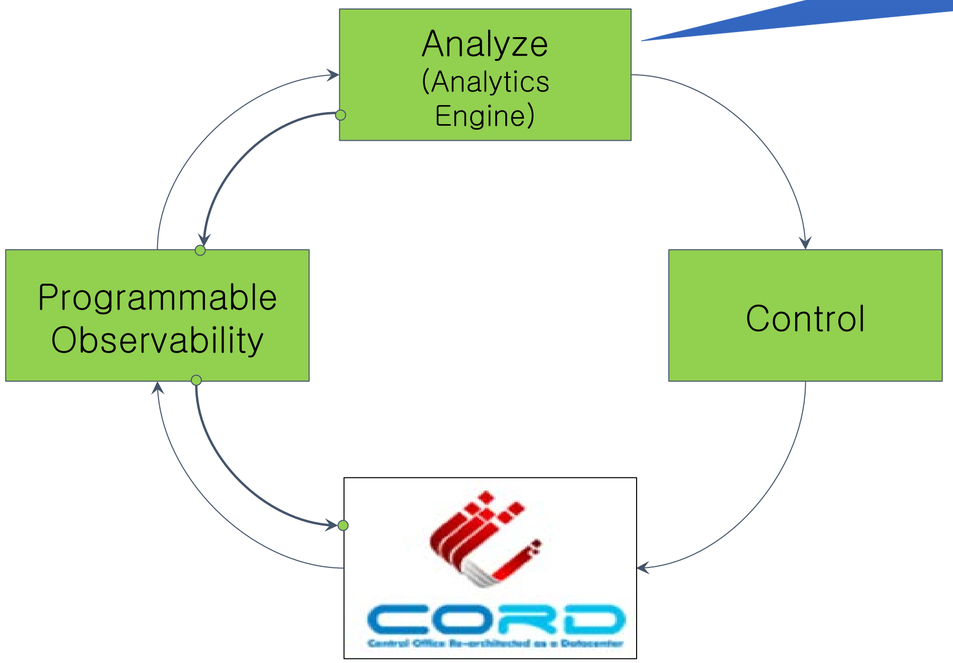
CORD Goals

- Enable programmable observability & closed loop control based on analytics

Examples:

- Fault correlation and Root cause analysis
- Fault prediction, Resource utilization prediction
- Anomaly Detection
- E2E SLA Violation Detection

Could we replace it with ML algorithms?



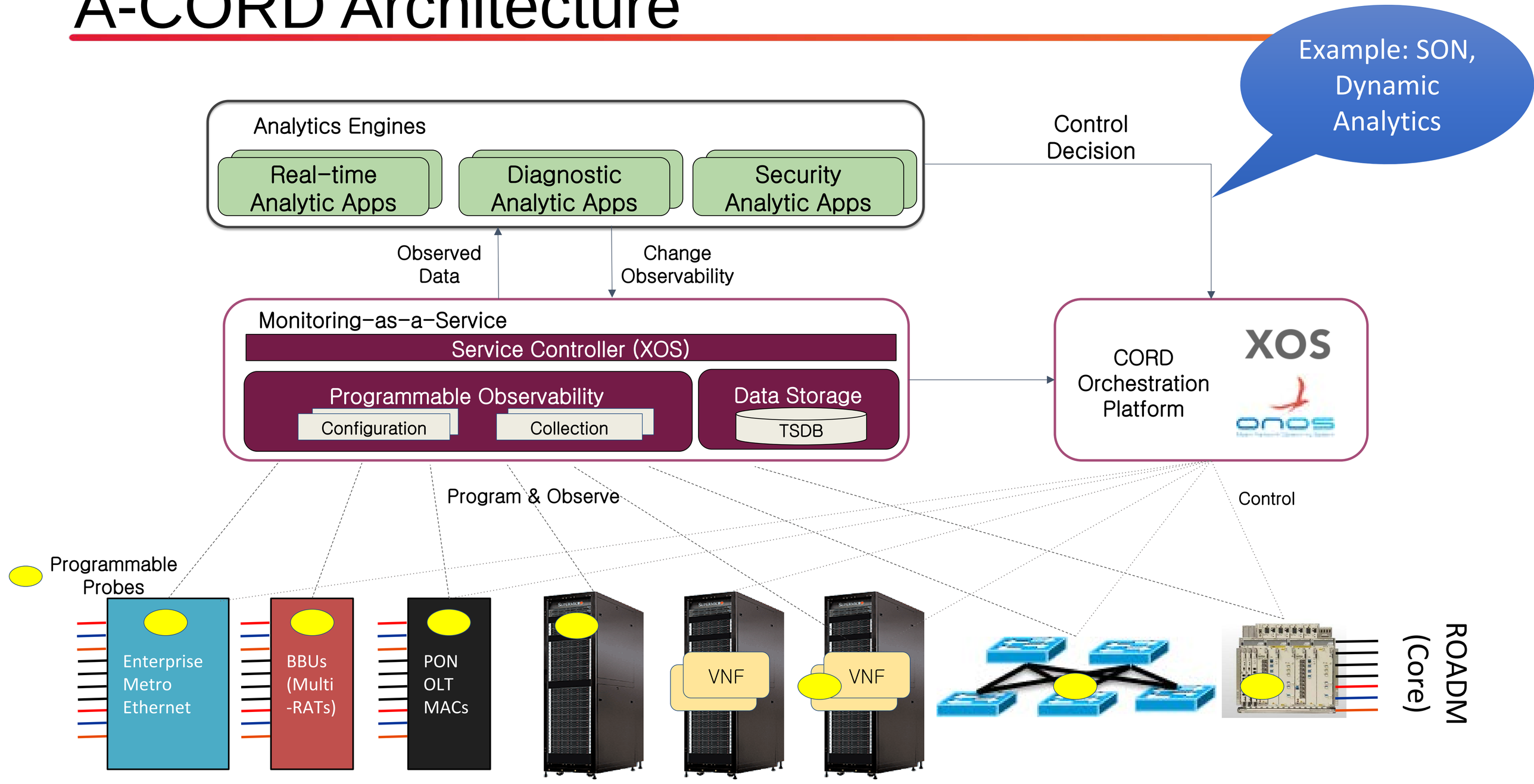
Examples:

- Observe packet ‘inter-arrival-time’ for a given time window and notify results
- Activate smart vProbe & Observe VNF for a target subscribers for a given time and notify events

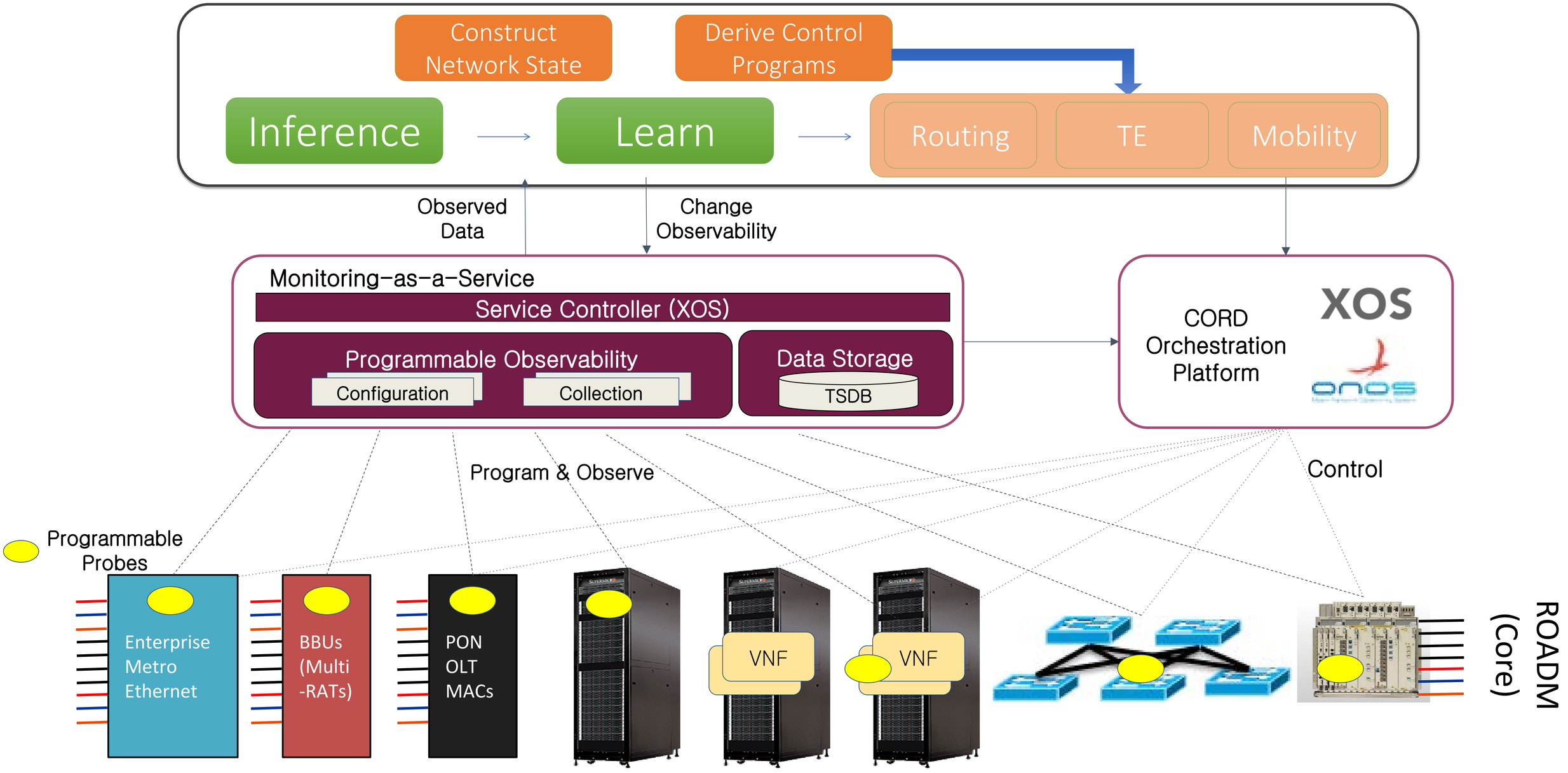
Examples:

- Resource scaling & load balancing
- SDN Flow Rerouting (Traffic Engineering)
- Blacklisting Subscriber Traffic

A-CORD Architecture

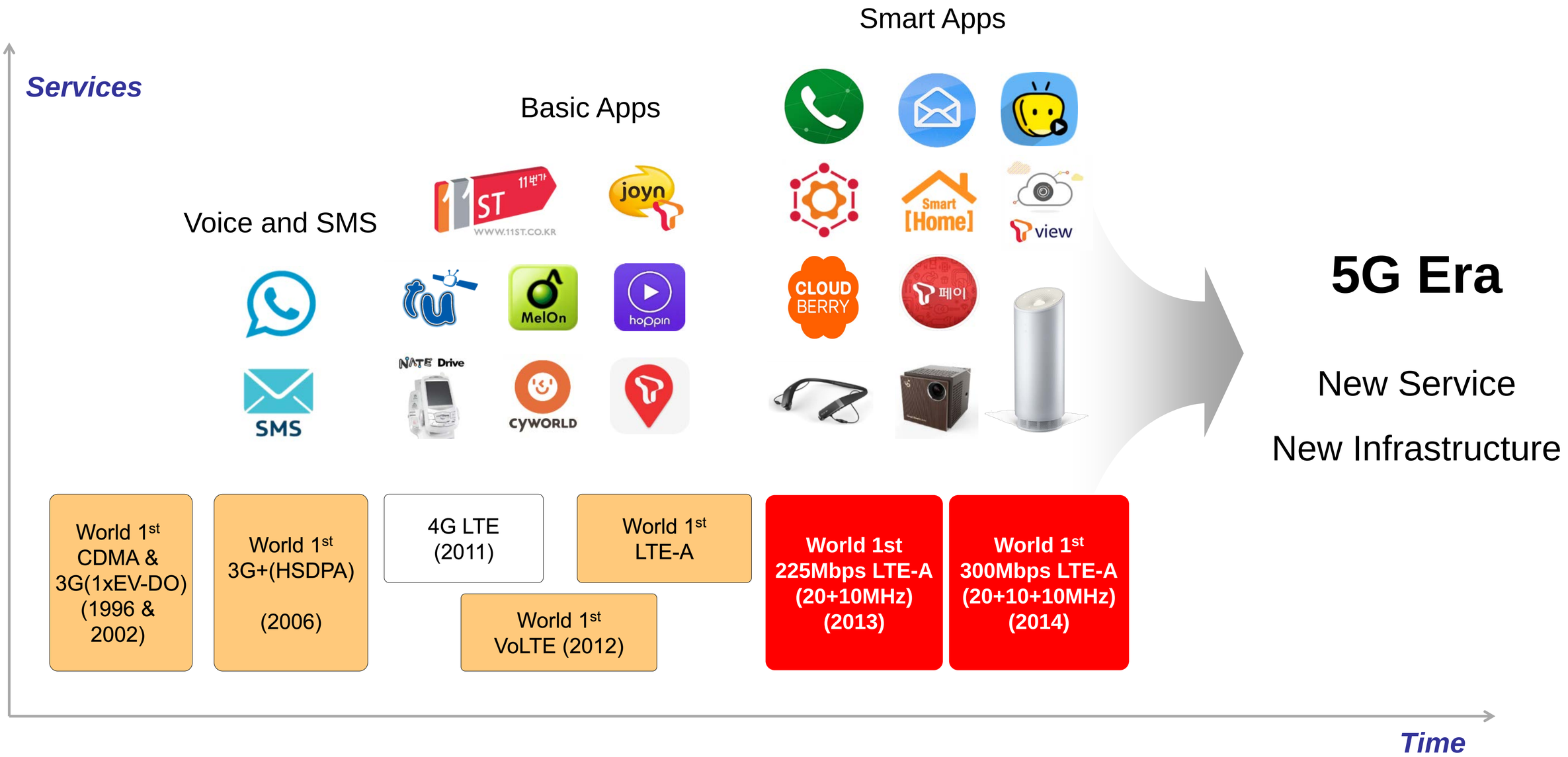


A-CORD + Self-Driving ?



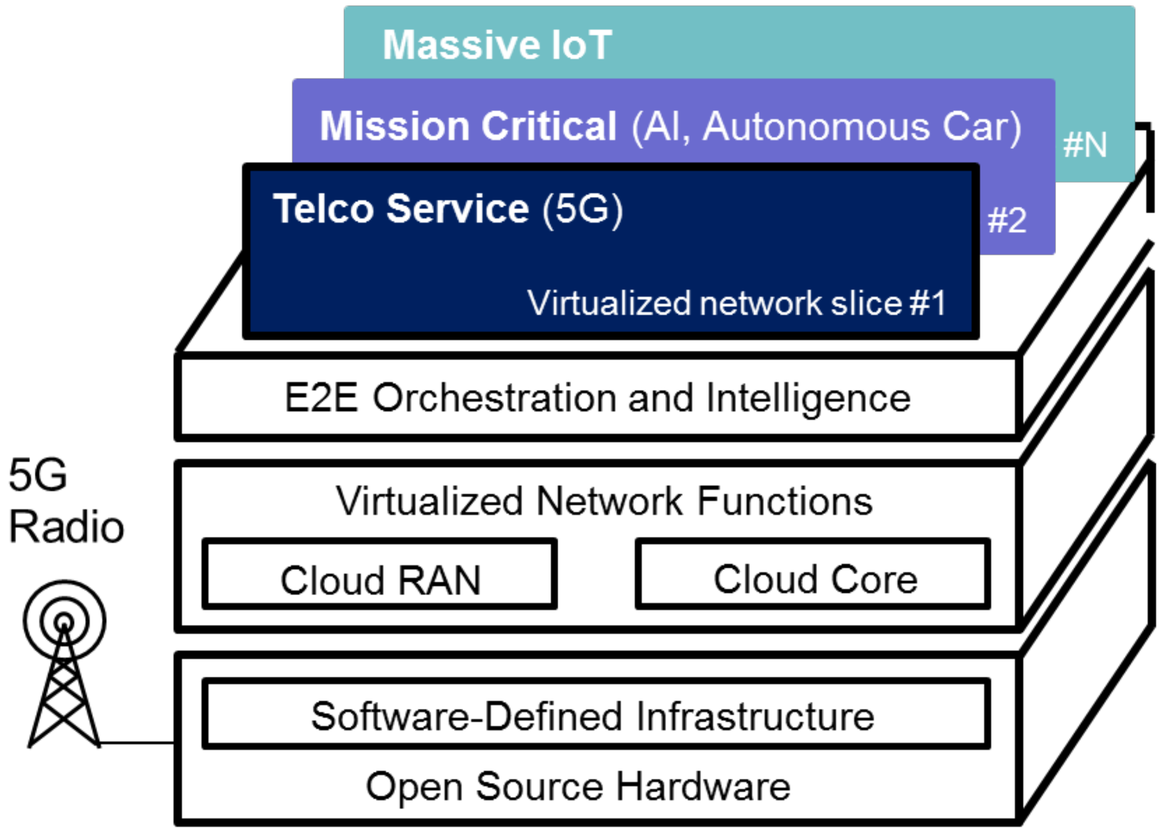
Use Case in SKT

SKT over years



Innovative Services of 5G

5G Evolution to Open Infrastructure



SKT Services

- Ultra High Data Rate
- Mission Critical Service
- Massive Connectivity

ATSCALE

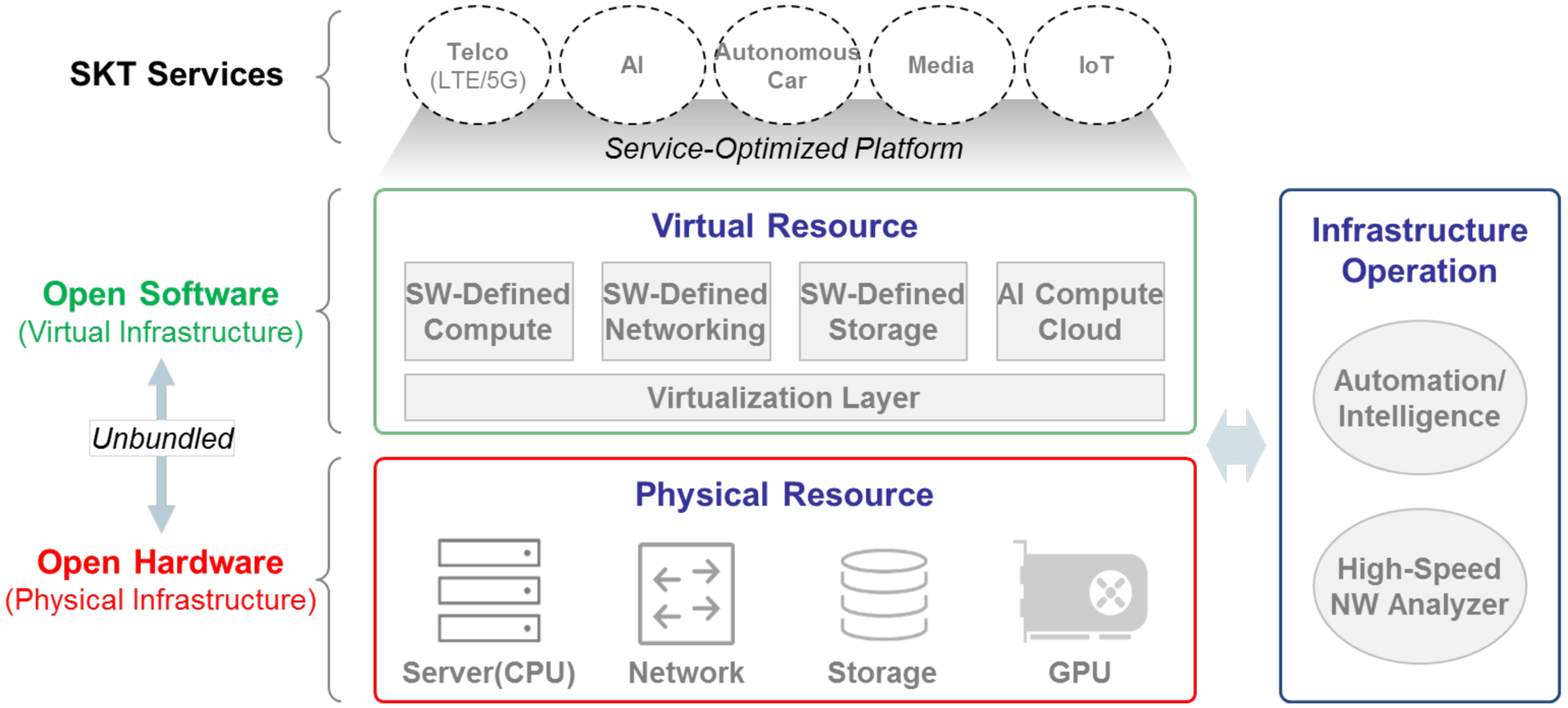
- Virtualized Network Functions
- Network & Service Slicing
- Next-Generation OSS (TANGO)

COSMOS

- Software-Defined Infrastructure
- Open Hardware and Software
- Telco & Mission Critical Services

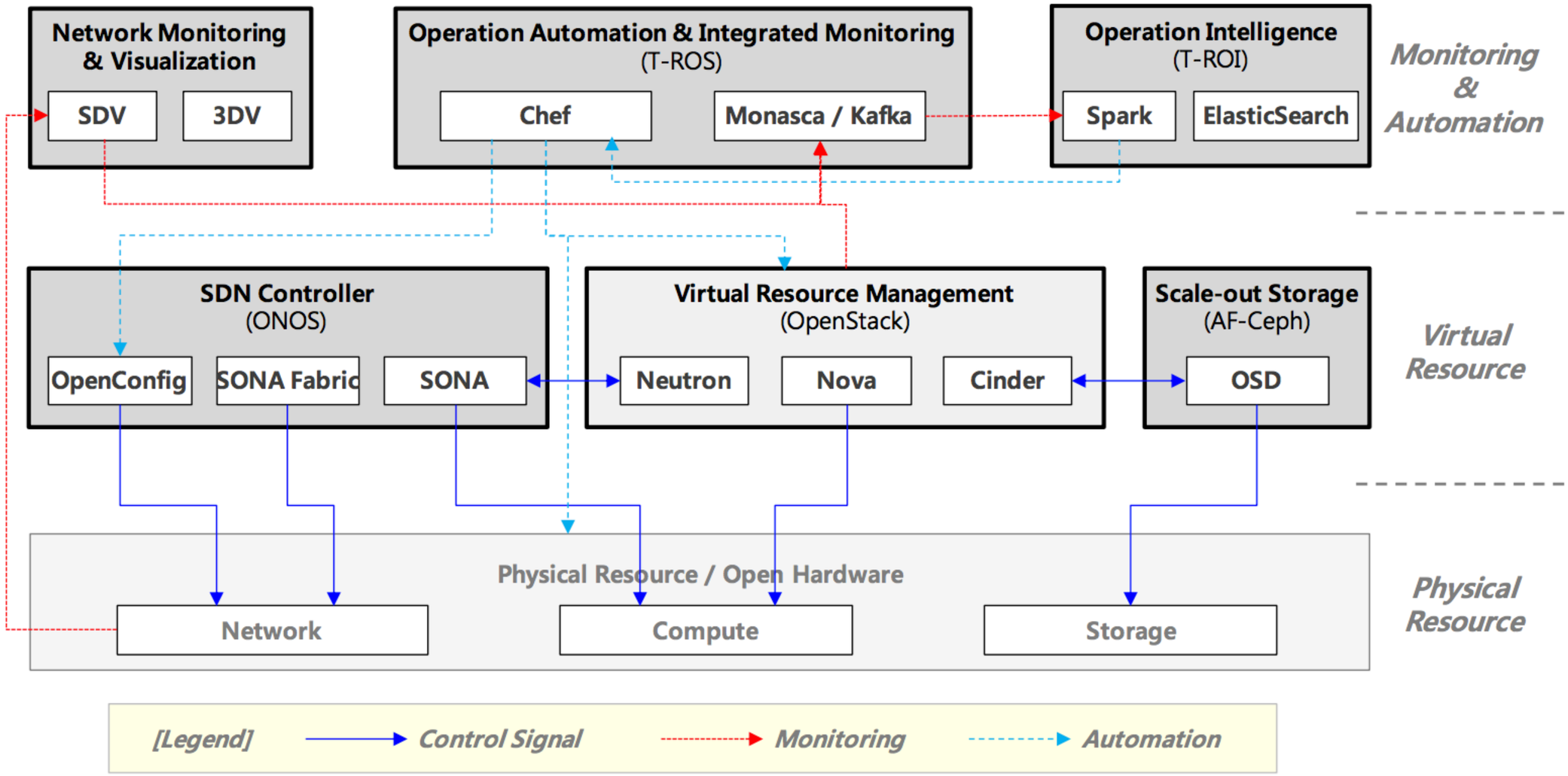
COSMOS Vision

- **Composable, Open, Scalable with Open Software and Hardware**
- **Mission Critical Services (5G, AI, Autonomous Car, etc.)**



COSMOS

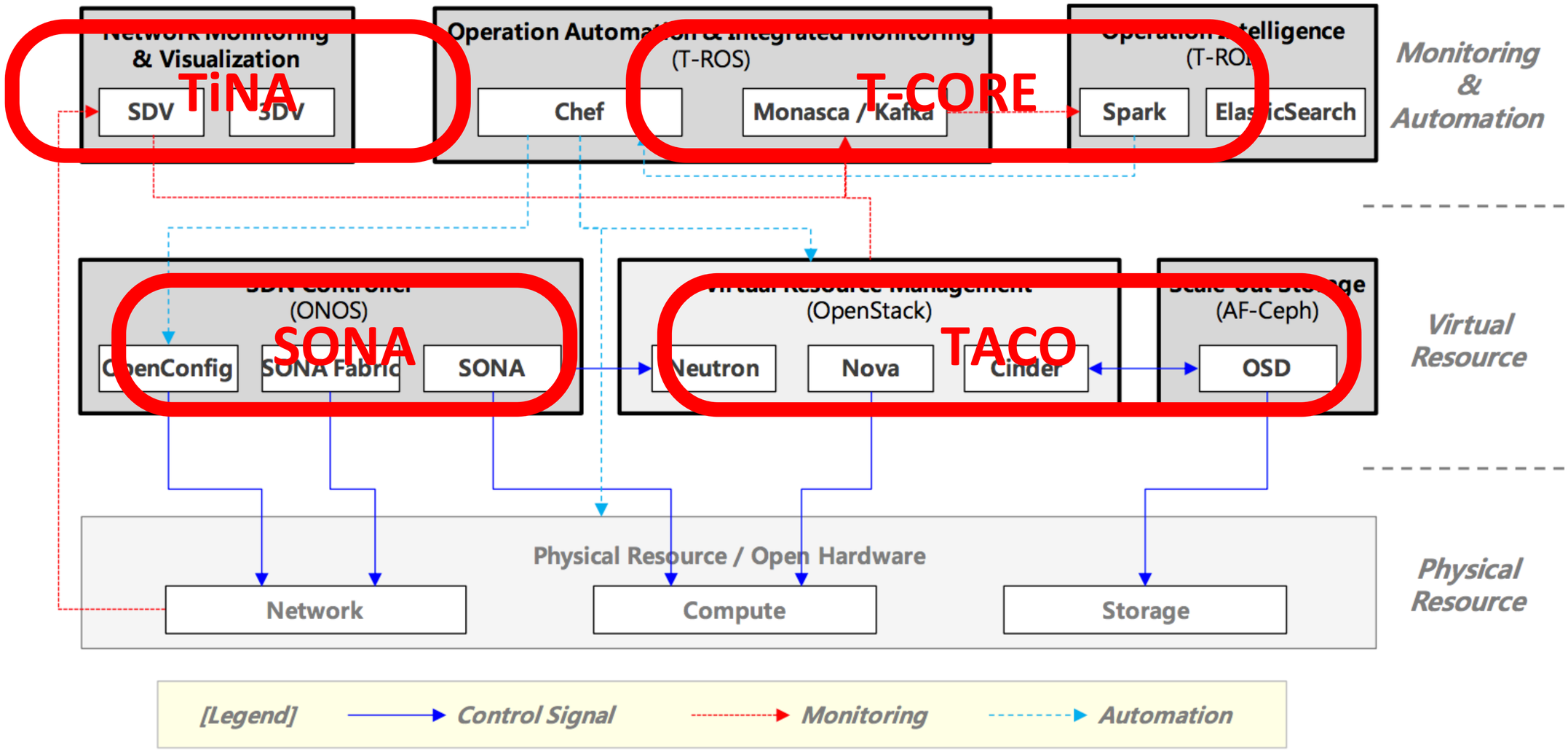
Composable Open Scalable Mission Critical Service



Ref: SKT COSMOS WhitePaper

COSMOS

Composable Open Scalable Mission Critical Service

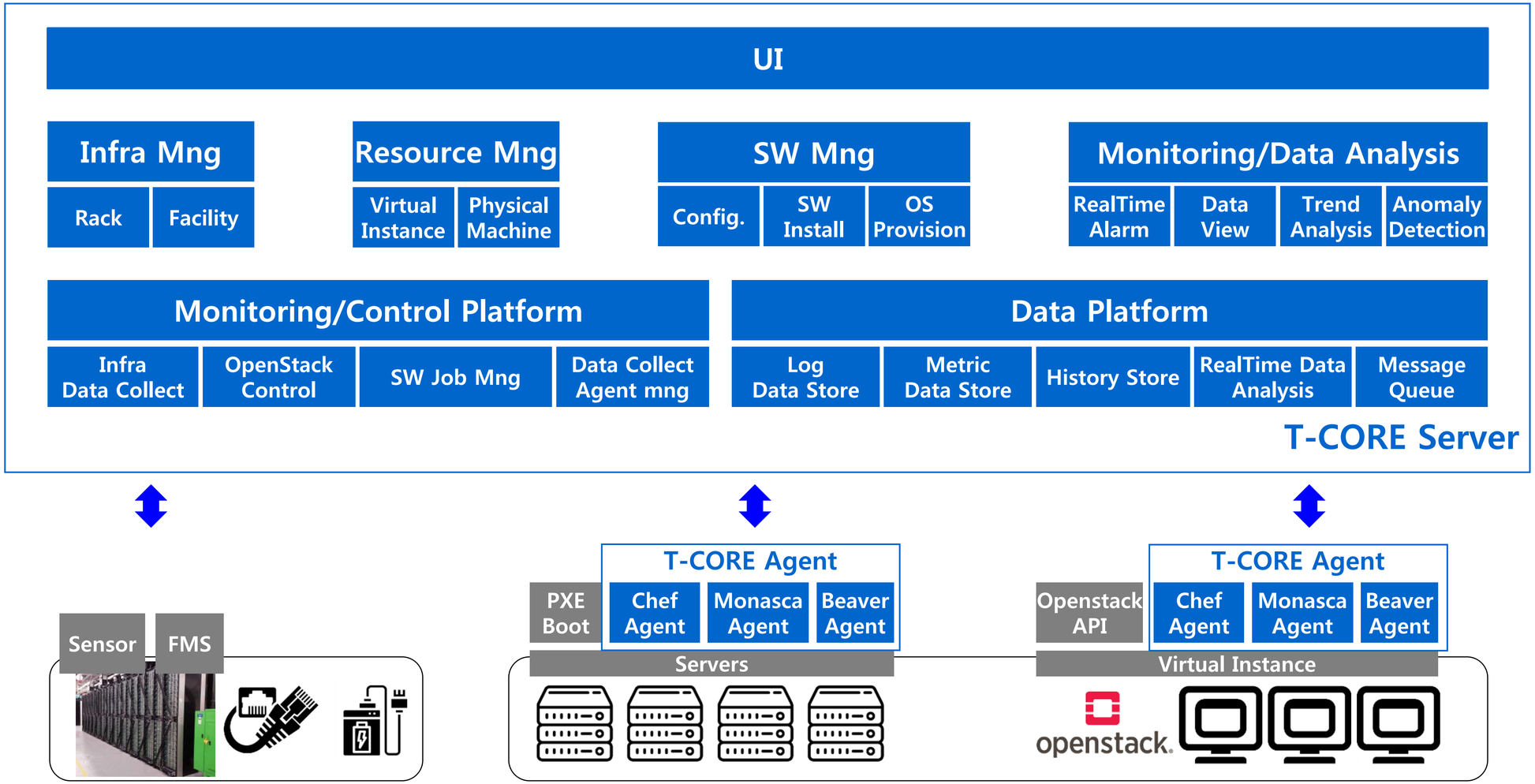


Ref: SKT COSMOS WhitePaper

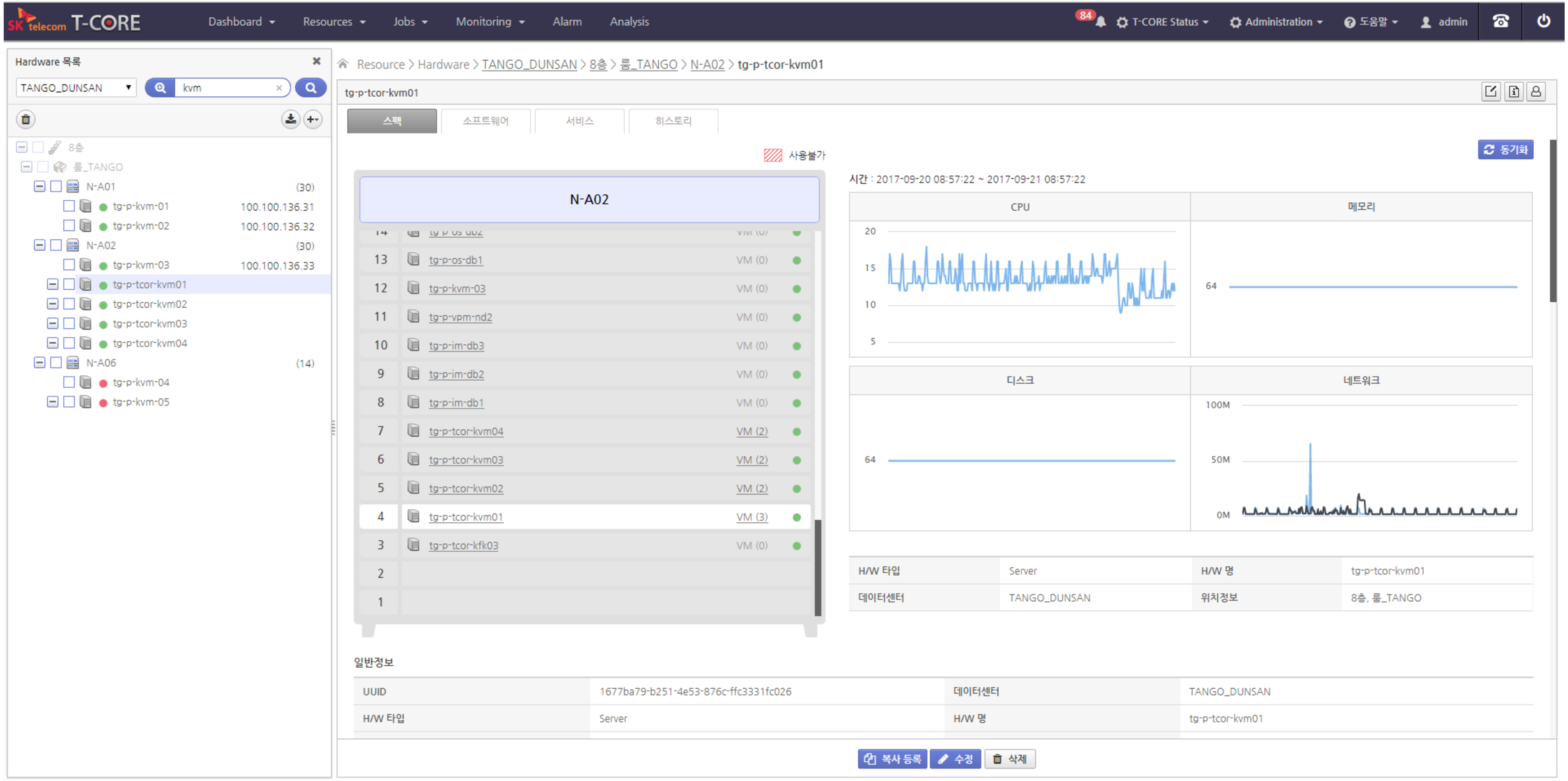
Operation Intelligence: T-CORE

T-CORE: Overview

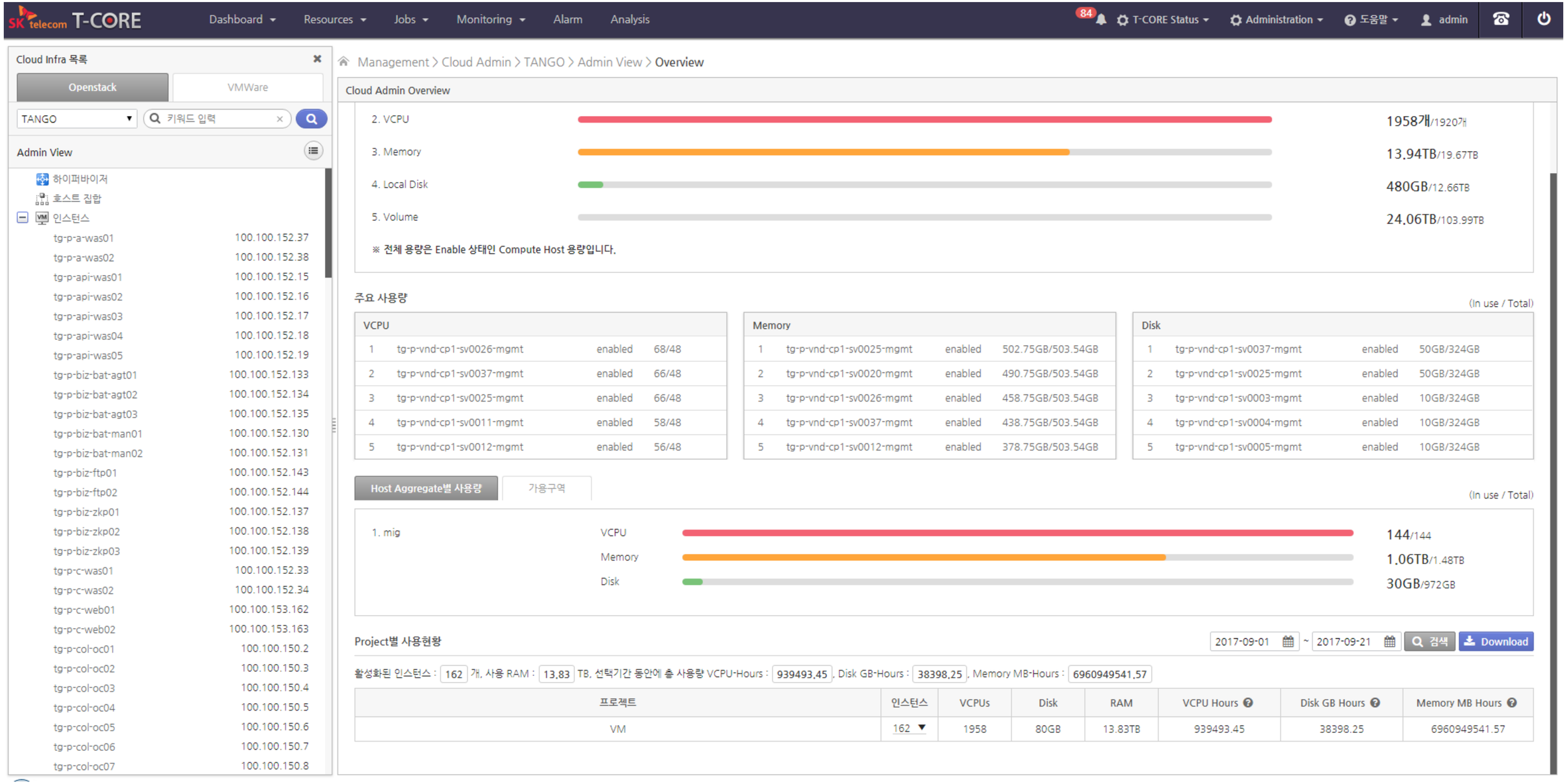
Data Center Operation Intelligence Solution through Big Data and API



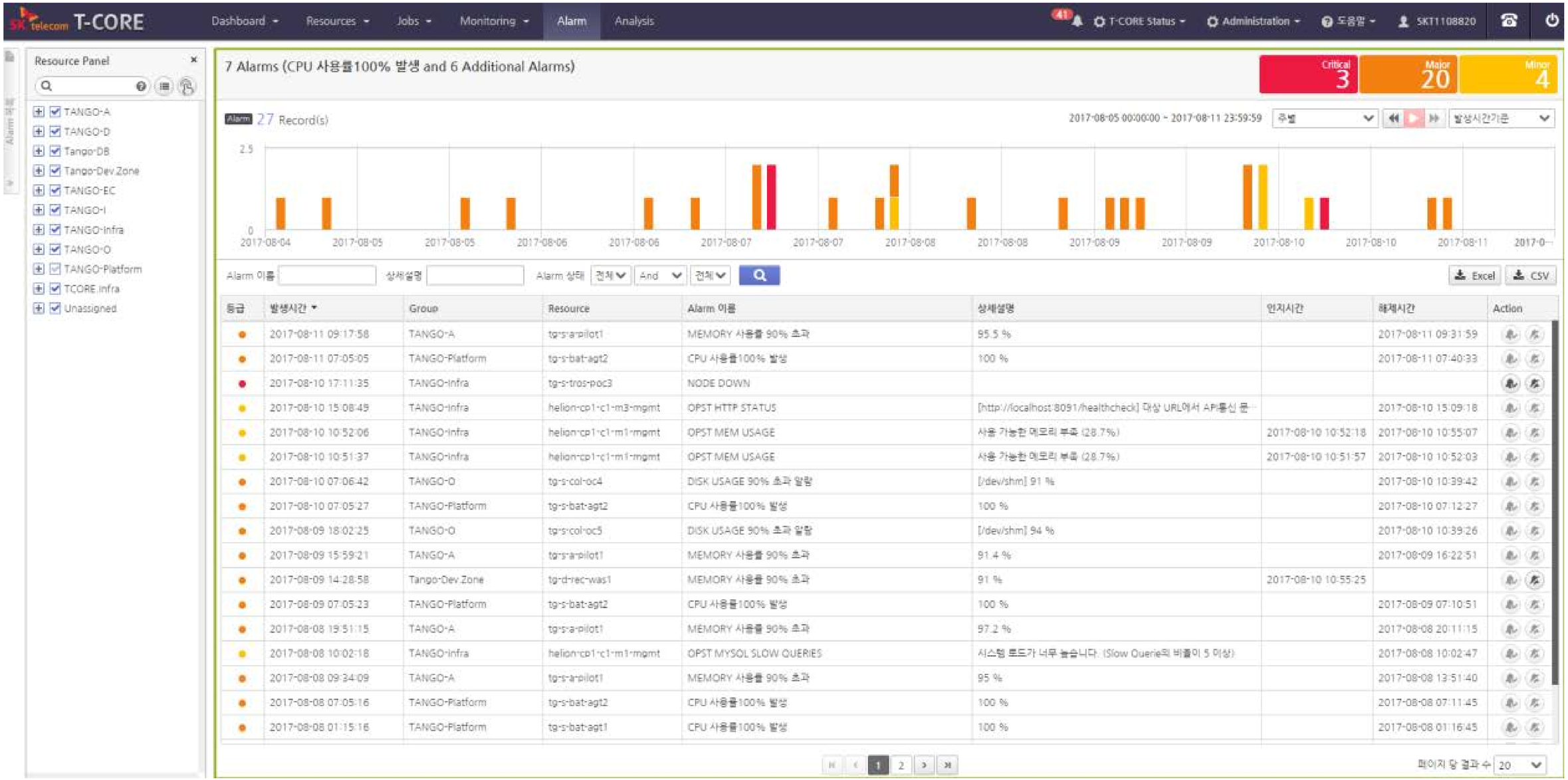
T-CORE: Hardware Resource Management



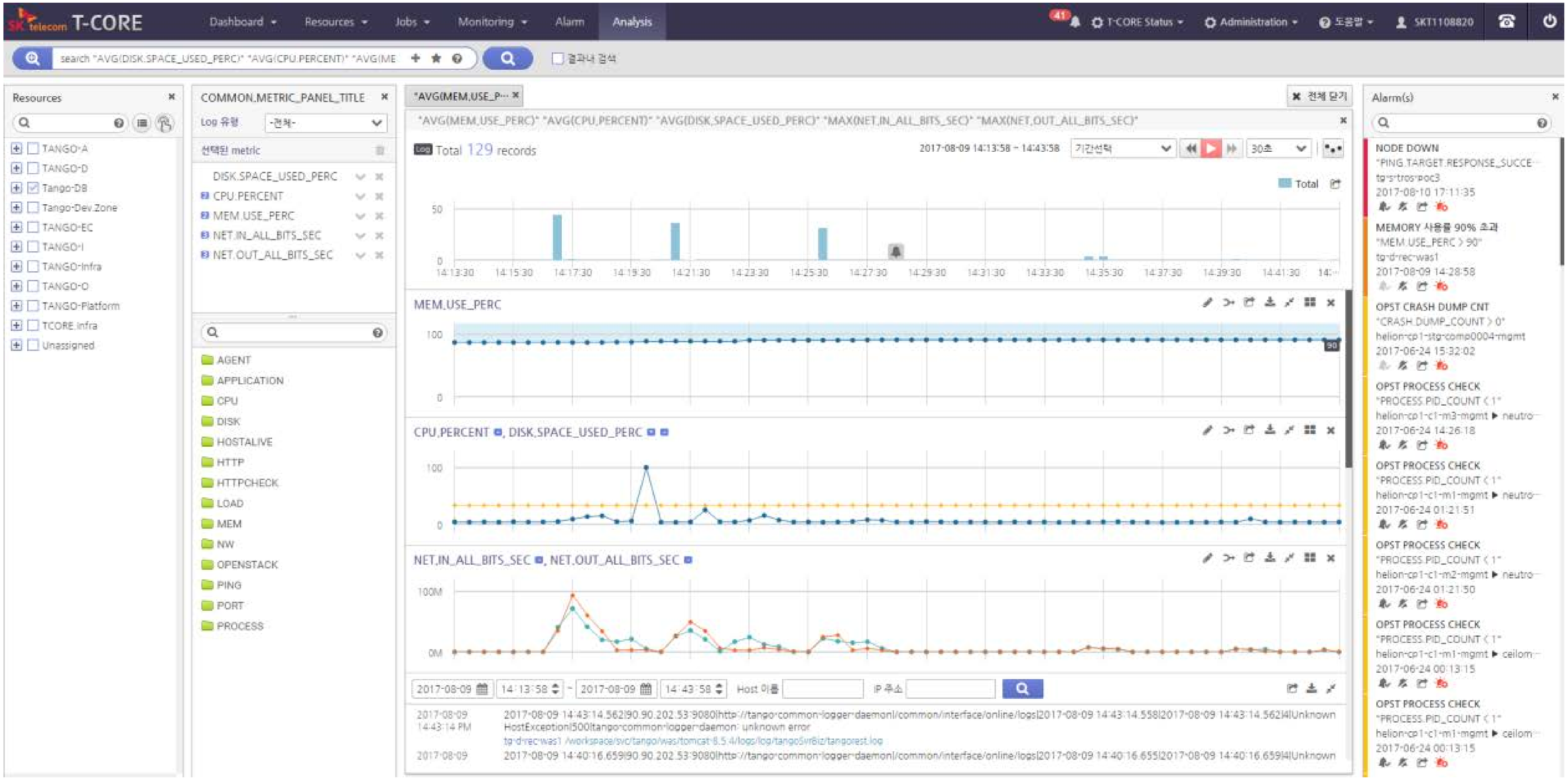
T-CORE: OpenStack Management



T-CORE: Alarm Statistics, Log Analysis



T-CORE: Log, Metric, Alarm Integrated Analysis



T-CORE: Deployment

Monitoring Infrastructure

- SKT’ OSS (Operation Support System)
- 400 servers (→ 1000 servers)
- 200 VM(HOS) Instance
- ‘17. 3.27

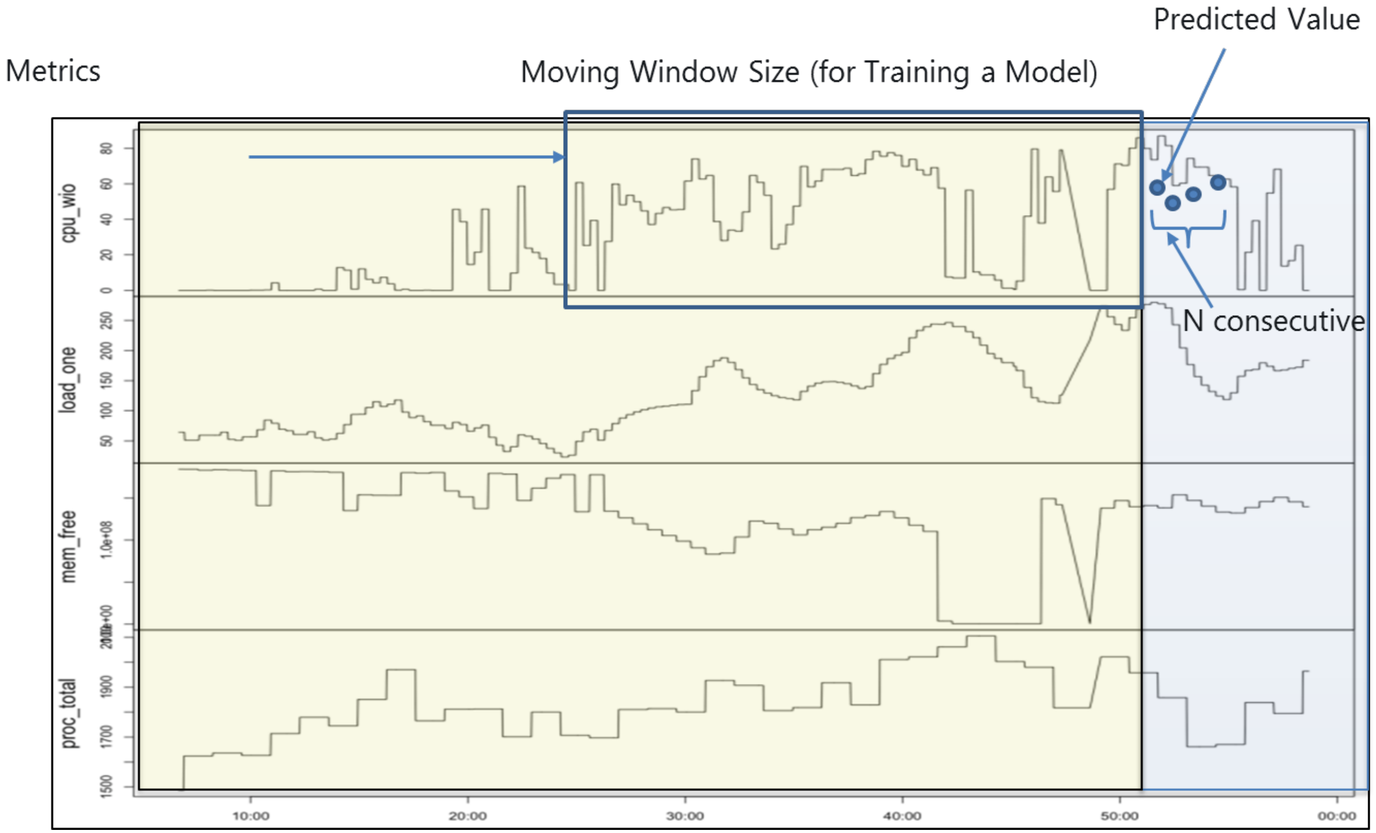
T-CORE: Deployment

Monitoring Infrastructure

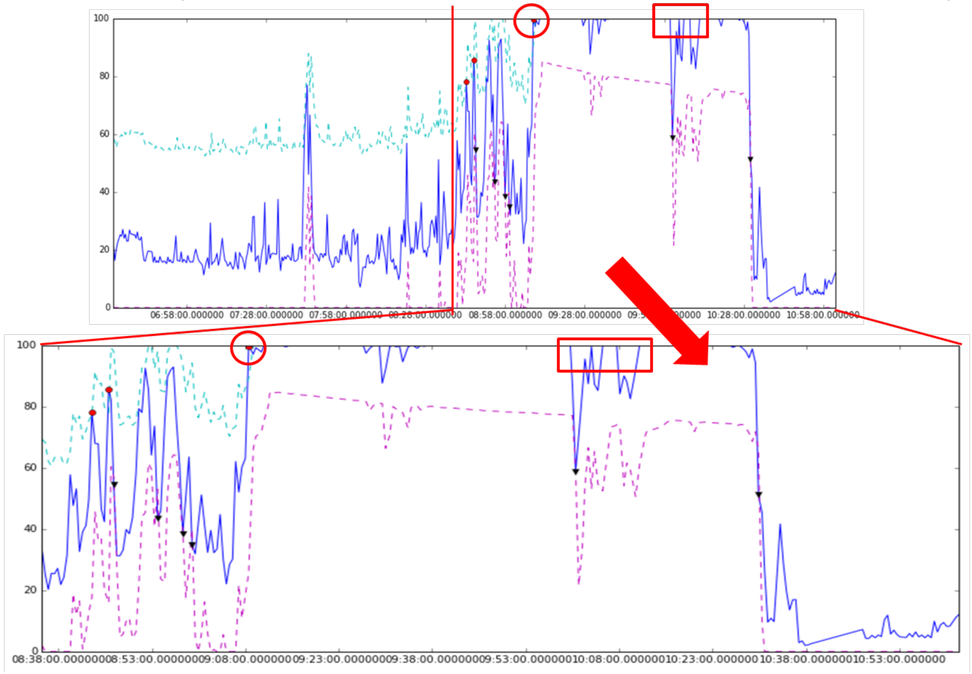
- SKT’ OSS (Operation Support System)
- 400 servers (→ 1000 serve
- 200 VM(HOS) Instance
- ‘17. 3.27

	Metric	Log
Items	70 and more	20 and more
Interval	Default: 30 secs Health check: 5 secs	Real-time
# of Alarms	30 and more	To be
Data per day	Collection: 500GB Store: 250GB	Collection: 200~600GB
Keep	3 months	

T-CORE: Metric Analysis



- CPU.PERCENT (Threshold = Prediction ± α·σ, with α = 4 - 3· Prediction / 100)



Anomaly Detection using dynamic threshold

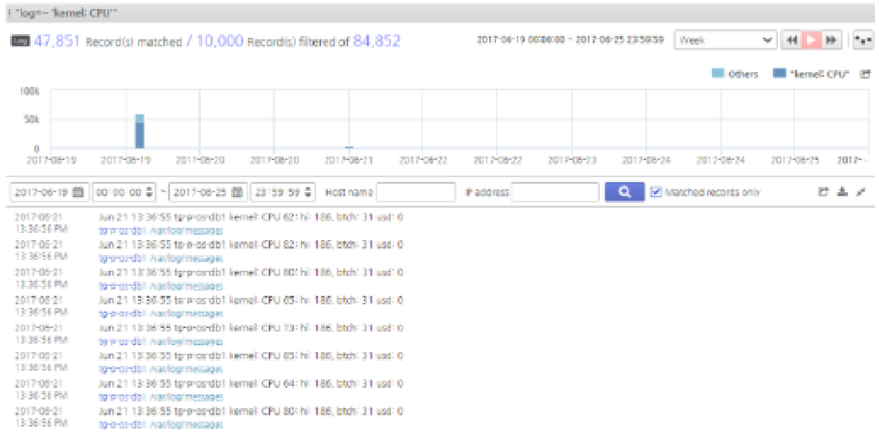
T-CORE: Log Analysis

/var/log/messages

해당 기간에 발생한 로그 81688건 중 81365건에 해당하는 message에 쌓인 로그는 로그량이 굉장히 많음.

Number of Logs	Regexed Log
47851	Jun ** 1*:**:** tg-p-os-db1 kernel: CPU **: hi: ***, btch: *1 usd: 0
17578	Jun ** **:**:** tg-p-os-db1 sshd[**5**]: pam_namespace(sshd:session)
2062	Jun 19 19:5*:** tg-p-os-db1 kernel: [2****] 54327 2**** ***6**** ****0**
2061	Jun 19 19:5*:** tg-p-os-db1 kernel: [*****] 5432* ***** 577*9*** ********
772	Jun ** 1*:**:** tg-p-os-db1 kernel: [<ffffffff814***3^>] ? *******_skb+0x**
558	Jun ** **:01:01 tg-p-os-db1 run-parts(/etc/cron.hourly)[*****]: ******** 0
536	Jun ** 1*:**:** tg-p-os-db1 kernel: lowmem_reserve[]: 0 0 0 0
524	Jun ** 1*:**:** tg-p-os-db1 kernel: Node 1 Normal per-cpu:
446	Jun ** 1*:**:** tg-p-os-db1 kernel: [<ffffffff8117f5bf>] ? cache_grow+0x
324	Jun ** **:**:*6 tg-p-os-db1 auditd[18***]: Audit daemon rotating log file
320	Jun ** 1*:**:** tg-p-os-db1 kernel: [<ffffffff811454f1>] ? lru_cache_add_
272	Jun ** 1*:**:*5 tg-p-os-db1 kernel: [<ffffffff81******>] ? ****_*e**********
256	Jun ** 1*:**:** tg-p-os-db1 kernel: [<ffffffff81466066>] ? __sys_sendms

일단 가장 많이 발생한 첫번째 로그 Jun ** 1*:**:** tg-p-os-db1 kernel: CPU **: hi: ***, btch: *1 usd: 0를 살펴보면, 해당 로그는 공교롭게도 19일과 21일에만 발생하였으며 해당일에 발생한 CPU 사용률 90% 초과 알람이 주변에 분포하고 있음.



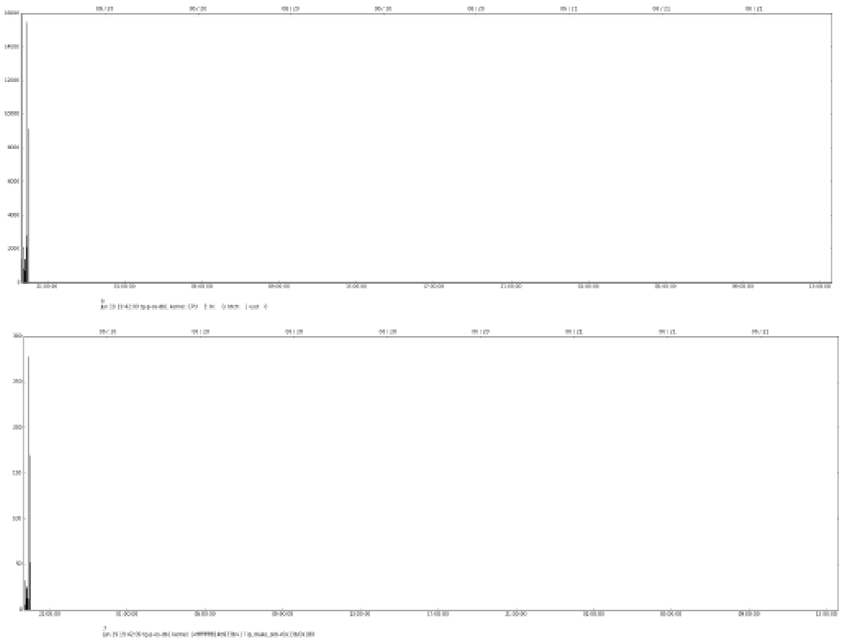
하기를 보면(알람발생), **6/19 19:41:04~19:55:29 (약 14분)**까지만 해당 로그가 쌓였는데, 해당 날짜의 CPU 90% 초과 알람은 **19:20:58**과 **19:55:41**에 발생하였고, 두번째 초과 알람이 발생하기 **14분전부터 12초 전**까지만 로그가 발생하였음. 또한, **6/21 13:36:45~13:36:55(약 1분)**간 로그가 발생하였는데 해당 일자의 CPU 90% 초과 알람은 **13:53:47**에 발생하였으므로 90% 초과 알람이 발생하기 **약 17분전에만 발생하였음.** **따라서, CPU 사용률 90% 알람과 깊은 관련이 있는 로그로 보이며 CPU 사용률과 관련하여 false-positive 알람(불필요한 초과 알람)을 줄이며 사전에 알람을 제공하는 키워드가 될 수 있지 않을까 함.**

Auto Log filtering & cross analysis with metrics

T-CORE: Log Analysis

로그 요약(패턴)	로그수
Jun ** 1*:**:** tg-p-os-db1 kernel: CPU **: hi: ***, btch: *1 usd: 0	47851
Jun ** **:**:** tg-p-os-db1 sshd[**5**]: pam_namespace(sshd:session): Unknown user xguest in configuration	17578
Jun 19 19:5*:** tg-p-os-db1 kernel: [2****] 54327 2**** ***6**** ****0** 1* 0 0 *********	2062
Jun 19 19:5*:** tg-p-os-db1 kernel: [*****] 5432* ***** 577*9*** ******** ** 0 0 ora*l*********	2061
Jun ** 1*:**:** tg-p-os-db1 kernel: [<ffffffff814***3*>] ? *******_skb+0x********	772
Jun ** **:01:01 tg-p-os-db1 run-parts(/etc/cron.hourly)[*****]: ******** 0anacron	558
Jun ** 1*:**:** tg-p-os-db1 kernel: lowmem_reserve[]: 0 0 0 0	536
Jun ** 1*:**:** tg-p-os-db1 kernel: Node 1 Normal per-cpu:	524
Jun ** 1*:**:** tg-p-os-db1 kernel: [<ffffffff8117f5bf>] ? cache_grow+0x2cf/0x320	446
Jun ** **:**:*6 tg-p-os-db1 auditd[18***]: Audit daemon rotating log files	324
Jun ** 1*:**:** tg-p-os-db1 kernel: [<ffffffff811454f1>] ? lru_cache_add_lru+0x21/0x40	320

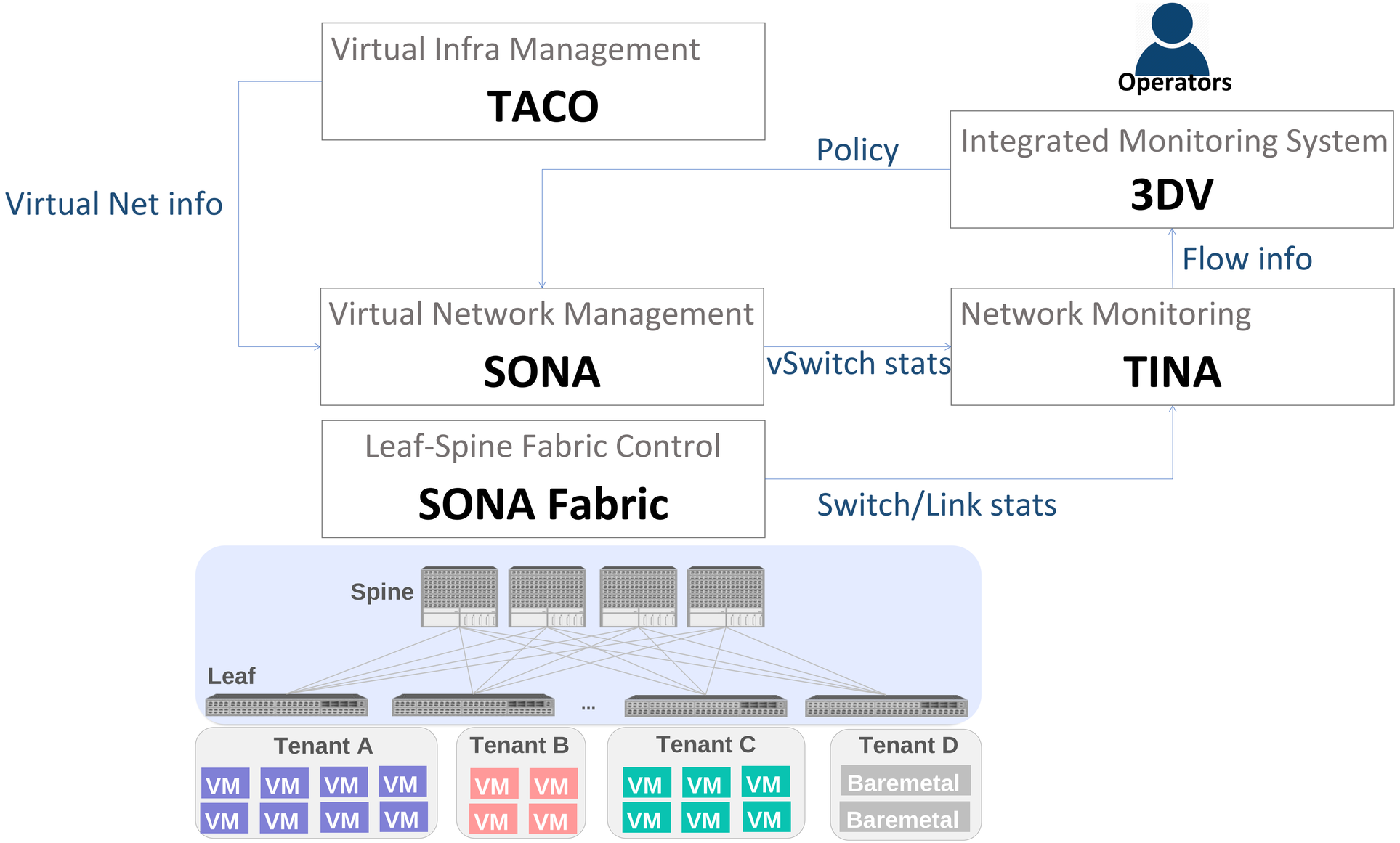
Level	Occured Time	Group	Resource	Alarm Name
●	2017-06-19 19:20:58	Tango-O	tg-p-os-db1	CPU 사용률 90% 초과
●	2017-06-19 19:55:41	Tango-O	tg-p-os-db1	CPU 사용률 90% 초과
●	2017-06-19 20:01:13	Tango-O	tg-p-os-db1	NODE DOWN
●	2017-06-19 20:04:49	Tango-O	tg-p-os-db1	ORACLE SERIVCE DOWN
●	2017-06-19 20:04:49	Tango-O	tg-p-os-db1	ORACLE SERIVCE DOWN
●	2017-06-20 13:53:47	Tango-O	tg-p-os-db1	CPU 사용률 90% 초과
●	2017-06-20 14:15:54	Tango-O	tg-p-os-db1	ORACLE SERIVCE DOWN
●	2017-06-20 14:17:25	Tango-O	tg-p-os-db1	ORACLE SERIVCE DOWN
●	2017-06-21 13:14:38	Tango-O	tg-p-os-db1	CPU 사용률 90% 초과
●	2017-06-21 13:43:06	Tango-O	tg-p-os-db1	ORACLE SERIVCE DOWN
●	2017-06-21 13:51:48	Tango-O	tg-p-os-db1	ORACLE SERIVCE DOWN
●	2017-06-22 21:11:45	Tango-O	tg-p-os-db1	ORACLE SERIVCE DOWN



Log distribution analysis and cross analysis with alarms

Network Intelligence: SONA-TINA

Open Networking in COSMOS



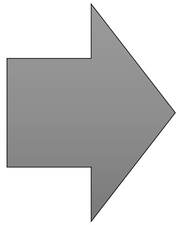
TACO (SKT All Container OpenStack)

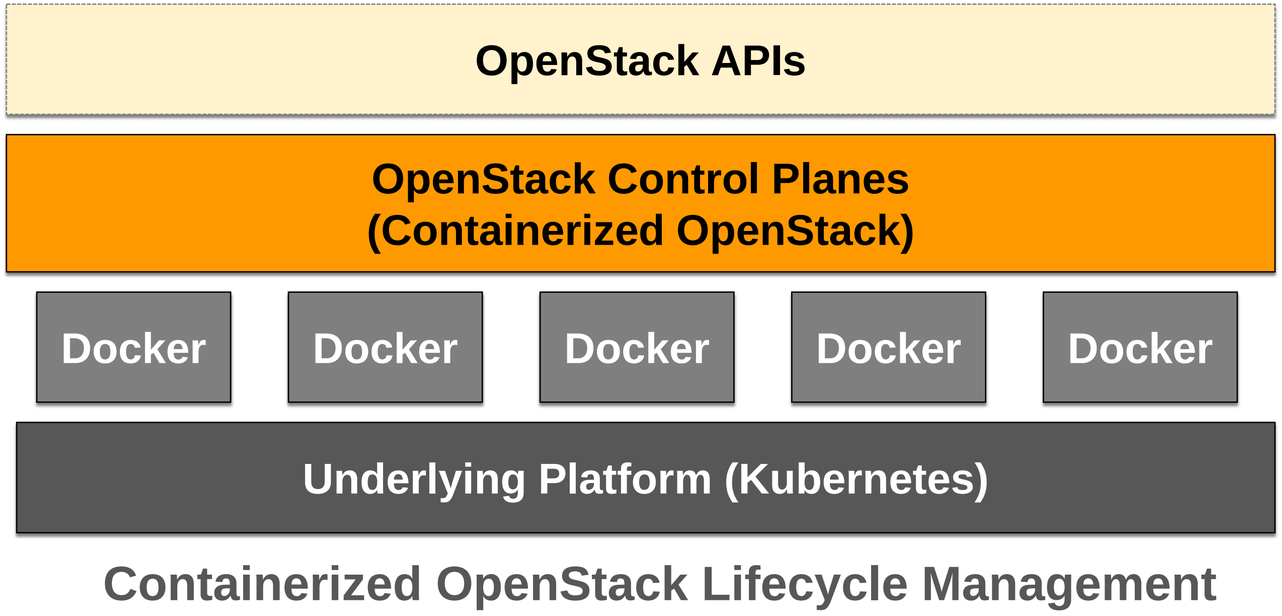
- Developed with **Container and Kubernetes**
- **Community Version** with **Continuous Integration / Delivery** System
- **Self-Healing, Upgrade w/o Service Interruption, Simple and Easy Deployment**

Container Orchestration
(Kubernetes, OpenStack-Helm)

OpenStack Containerization
(Docker, Kolla)

CI/CD & Testing
(Jenkins, Rally/Tempest, Chaos Monkey)

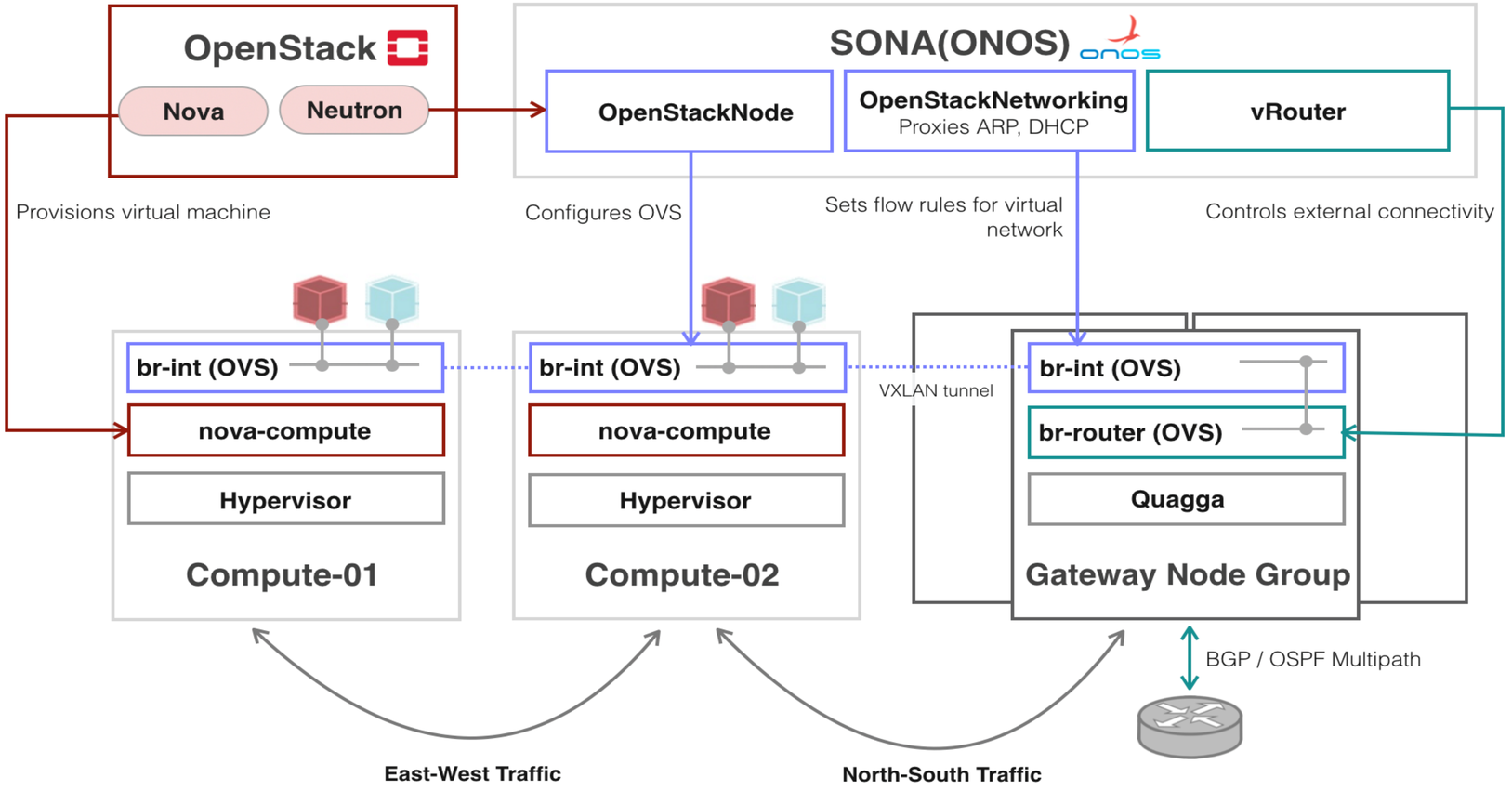




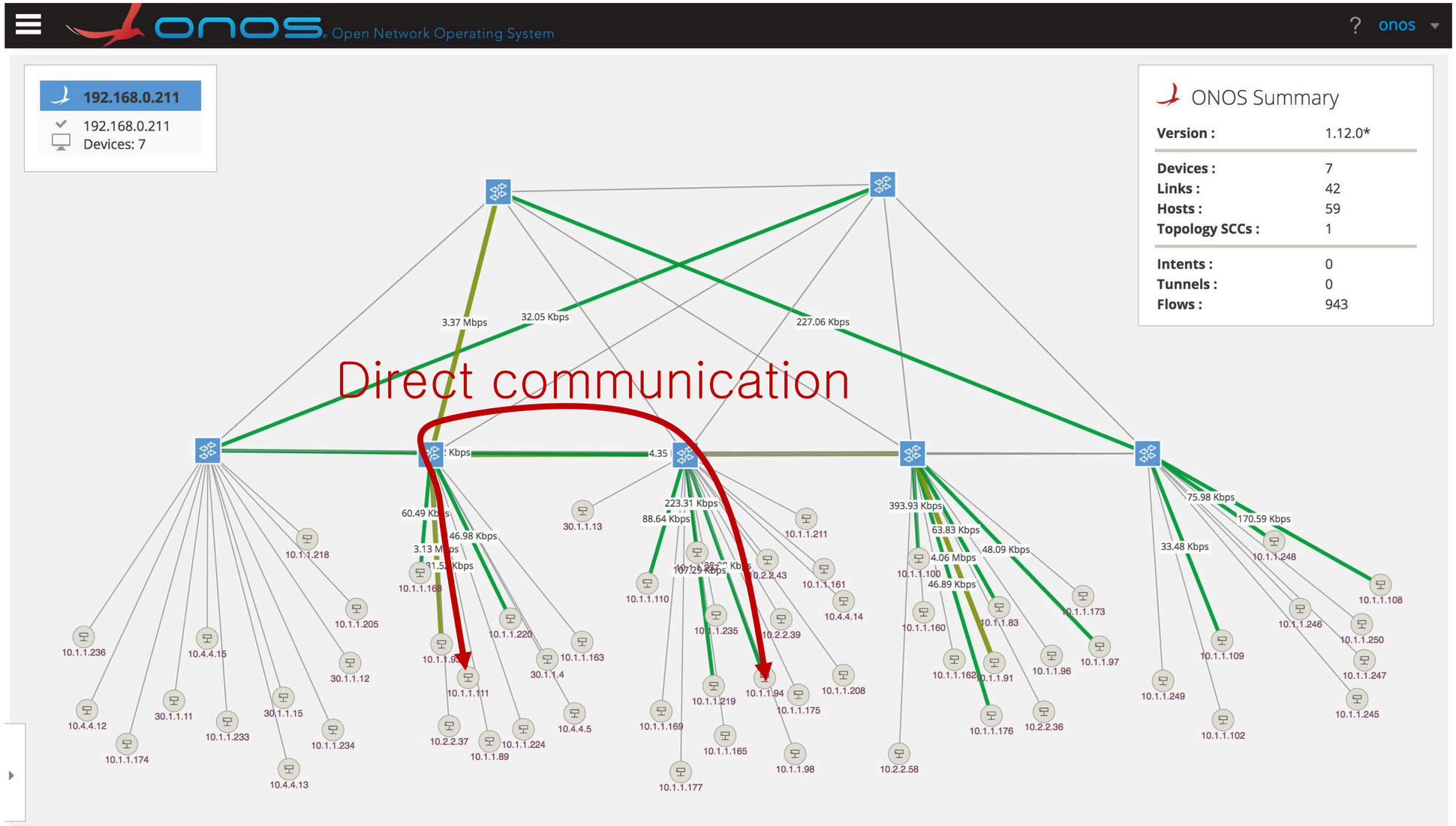
Containerized OpenStack Lifecycle Management

SONA (Simplified Overlay Networking Architecture)

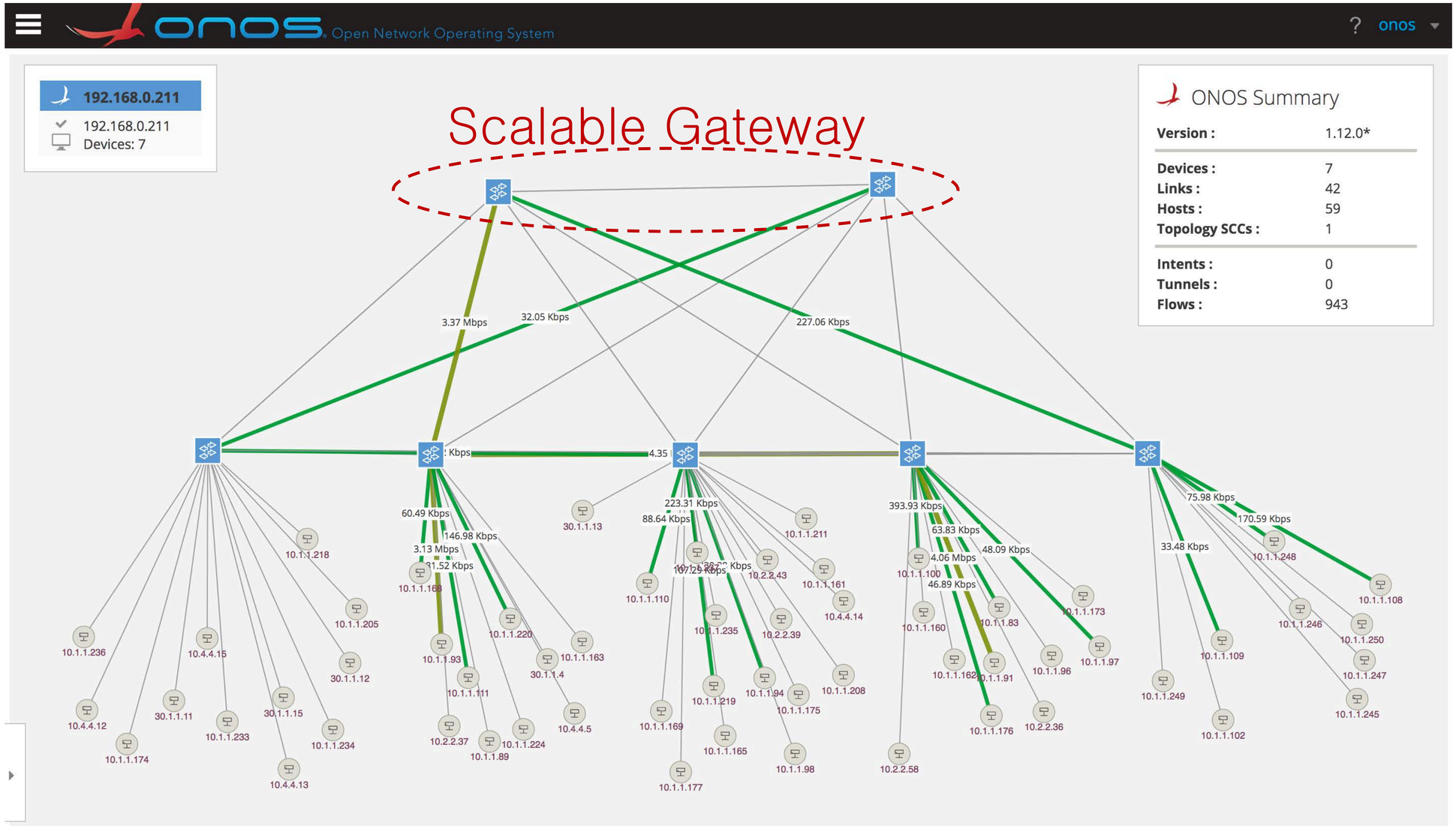
- ONOS based **Virtual Network Management** solution
- **Multi-Tenancy** support with **VxLAN** (and VLAN also)
- **Scalable Gateway,** Full Compatible with **OpenStack**



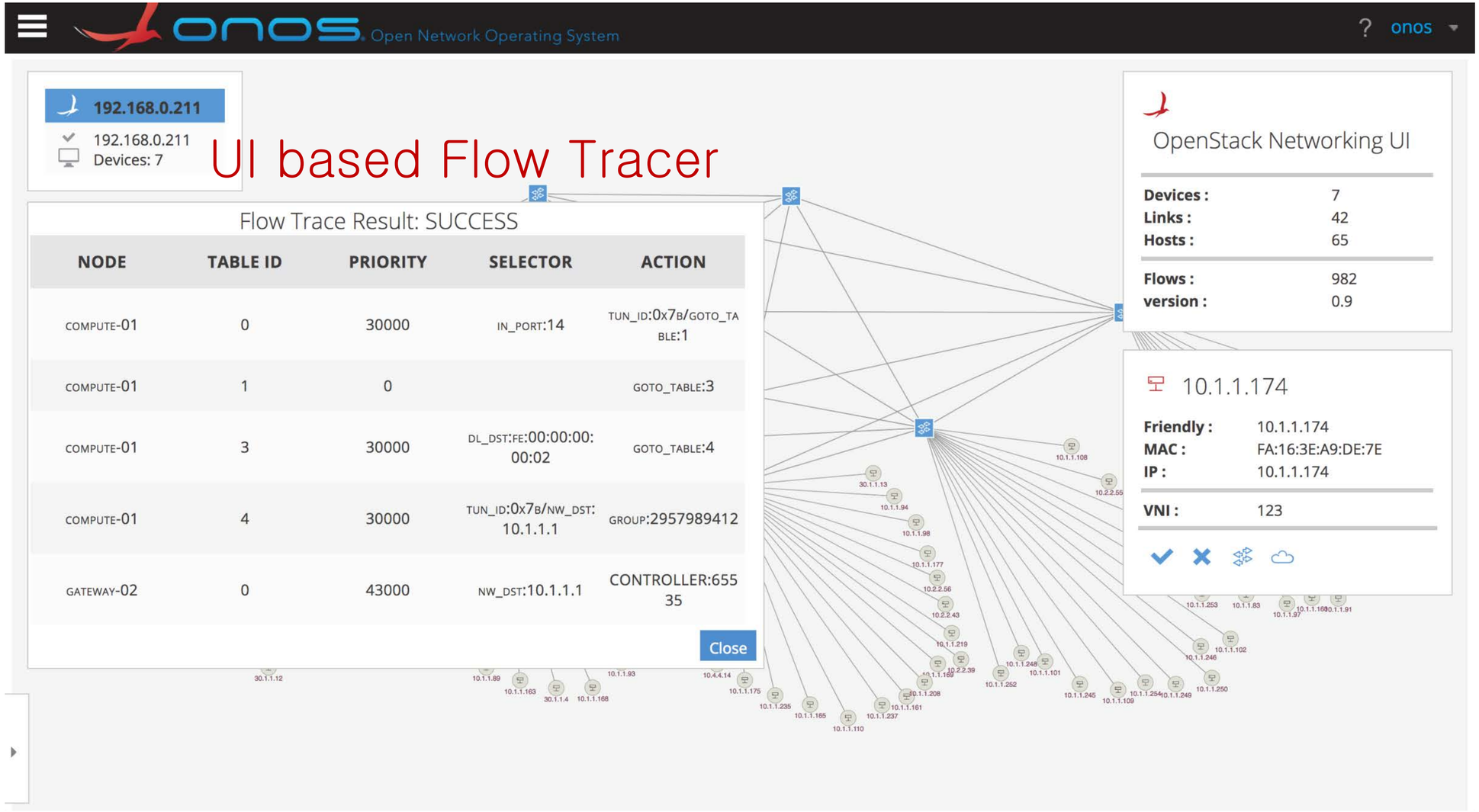
SONA (Simplified Overlay Networking Architecture)



SONA (Simplified Overlay Networking Architecture)

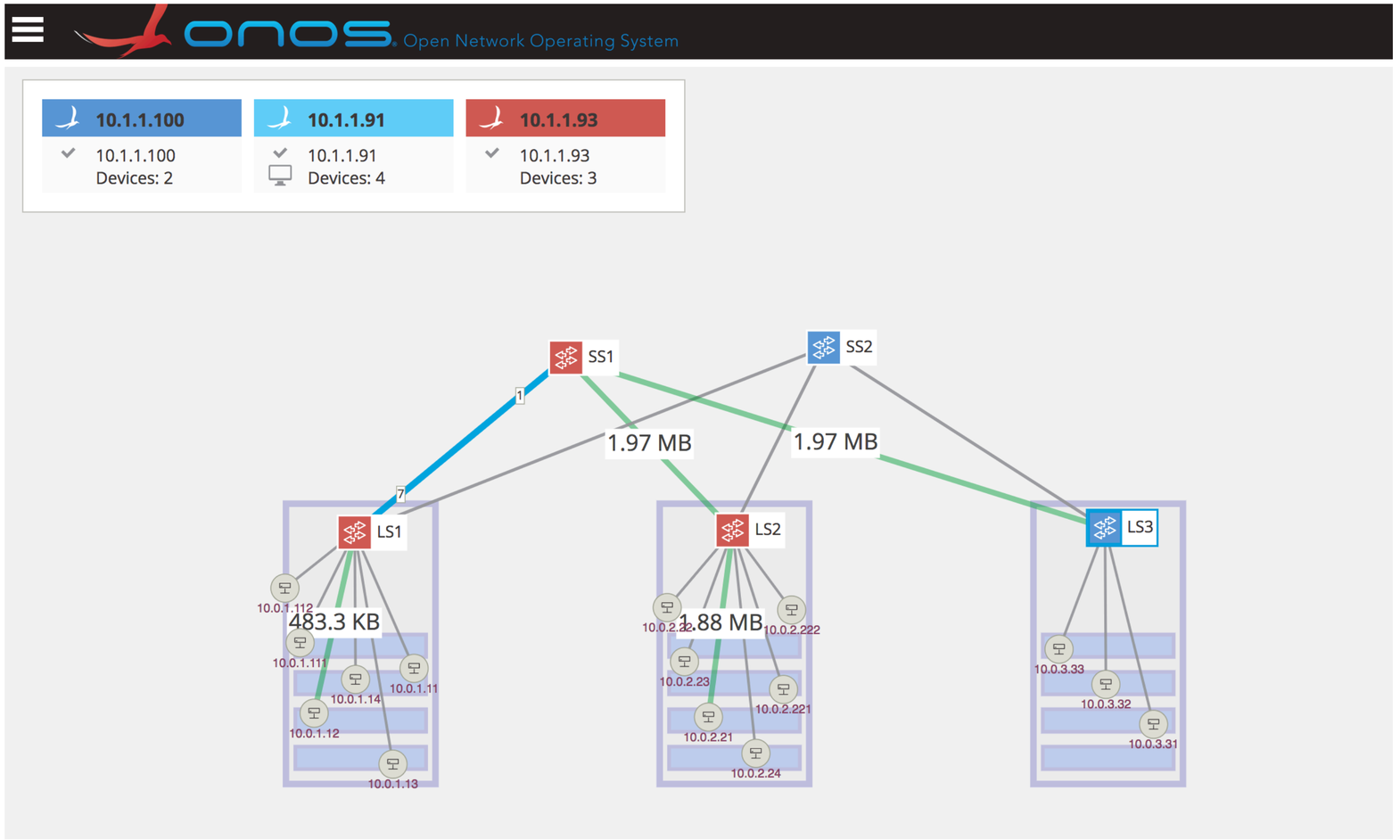


SONA (Simplified Overlay Networking Architecture)



SONA Fabric

- Pure OpenFlow based Leaf-Spine Fabric Solution
- Supports ECMP, Failure detection & auto recovery
- Physical + Virtual Network Integration



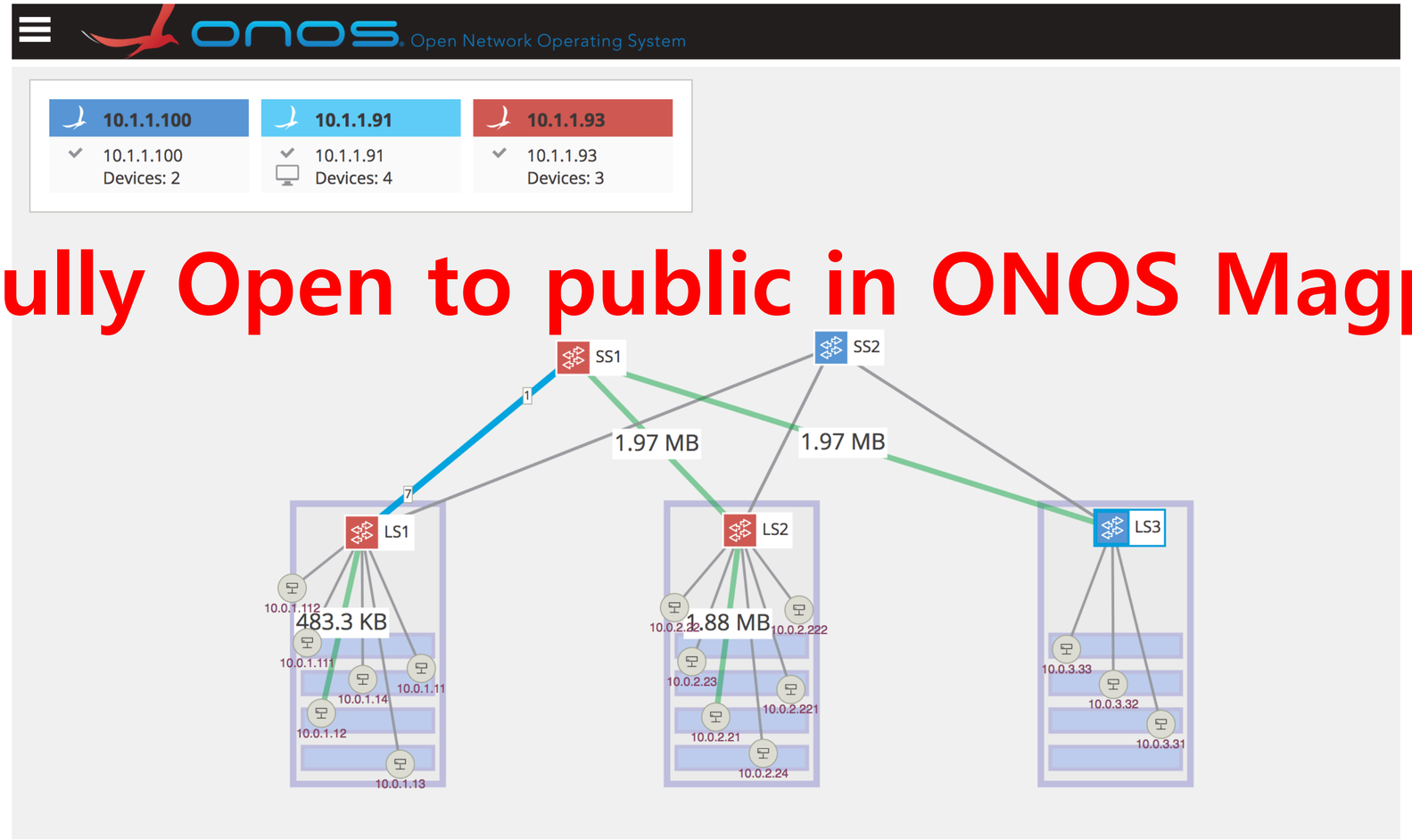
ARISTA

CISCO

Edge-corE NETWORKS

SONA Fabric

- Pure OpenFlow based Leaf-Spine Fabric Solution
- Supports ECMP, Failure detection & auto recovery
- Physical + Virtual Network Integration



Fully Open to public in ONOS Magpie

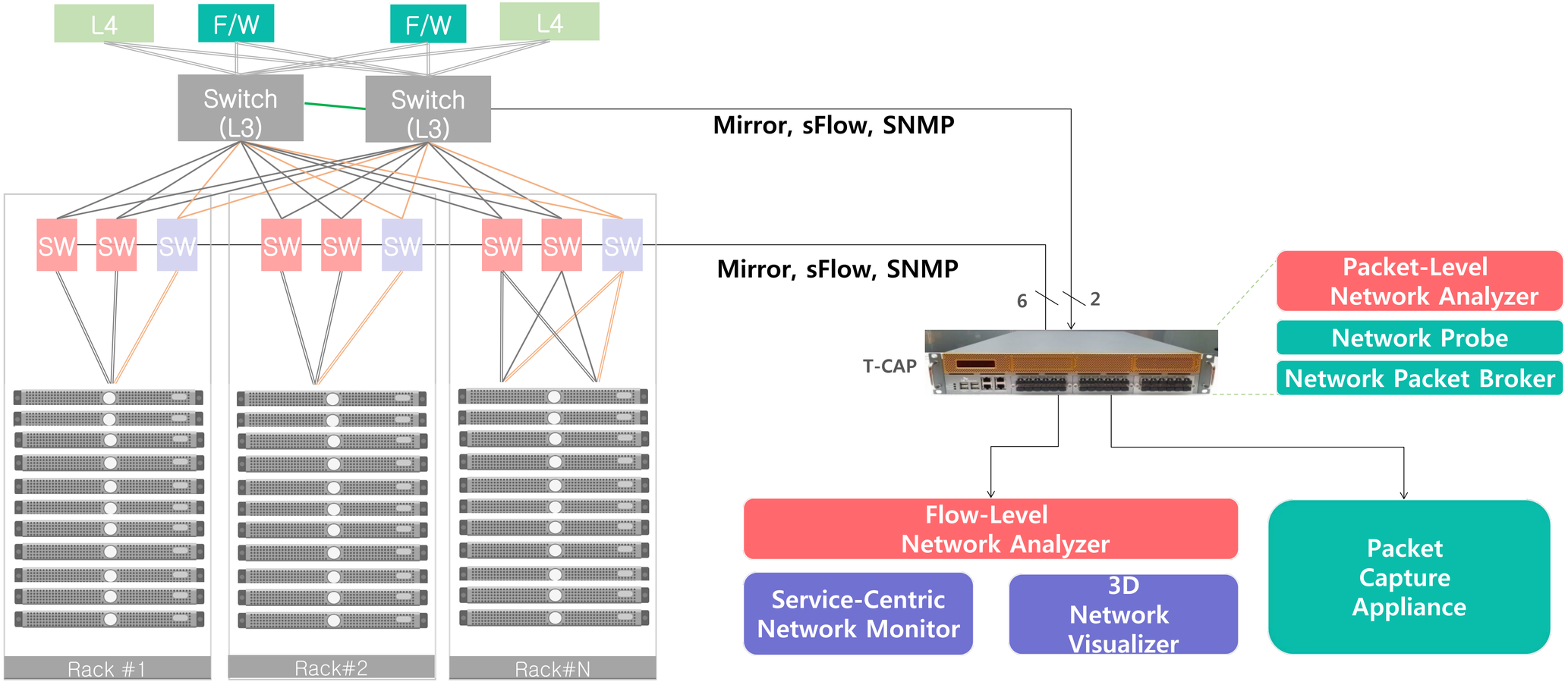
ARISTA

CISCO

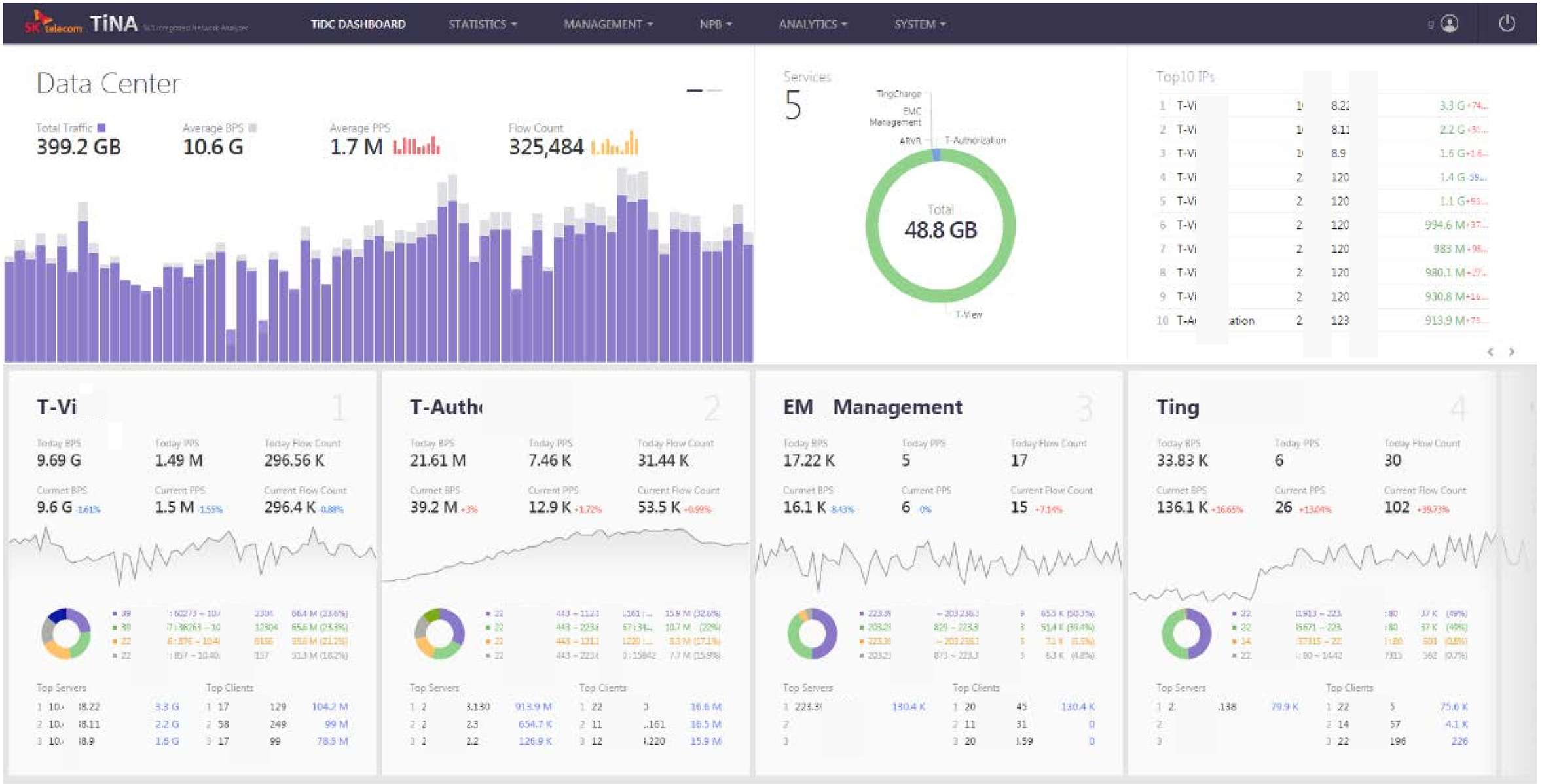
Edge-corE NETWORKS

TINA (SKT Integrated Network Analyzer)

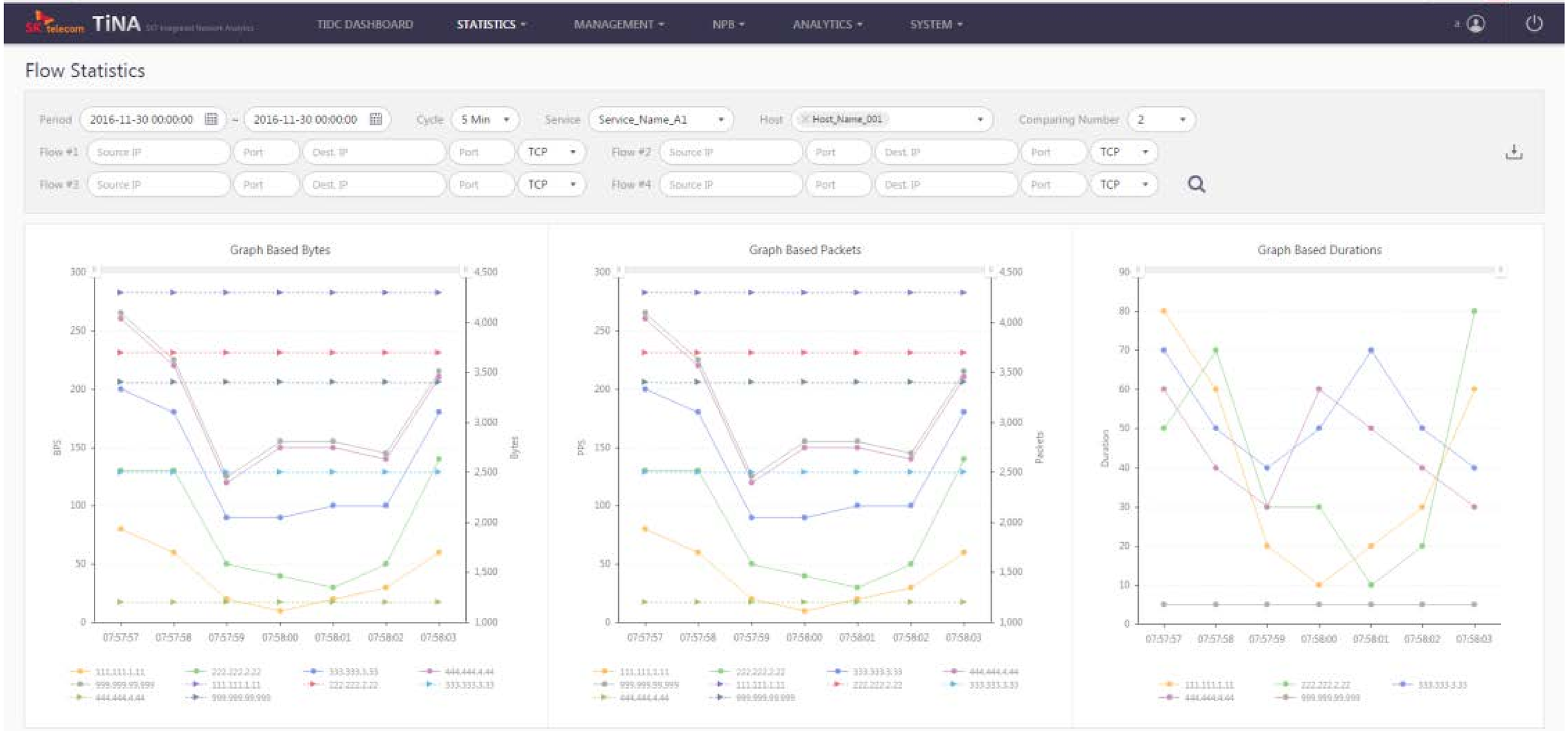
- Unified network monitoring and operating solution
- Essential systems and tools for network health monitoring, traffic monitoring
- Supports packet analysis, session analysis, and troubleshooting.



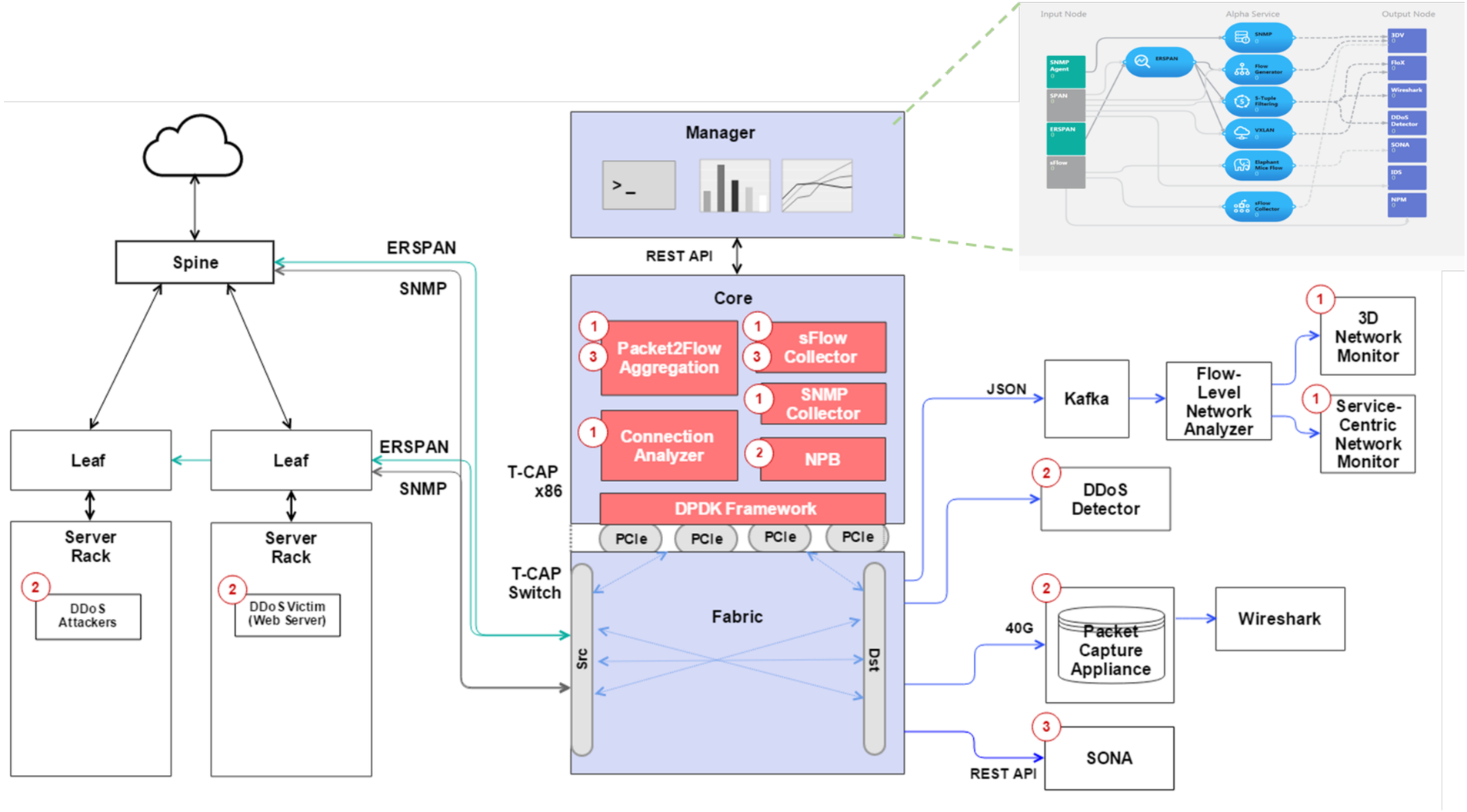
TINA (SKT Integrated Network Analyzer)



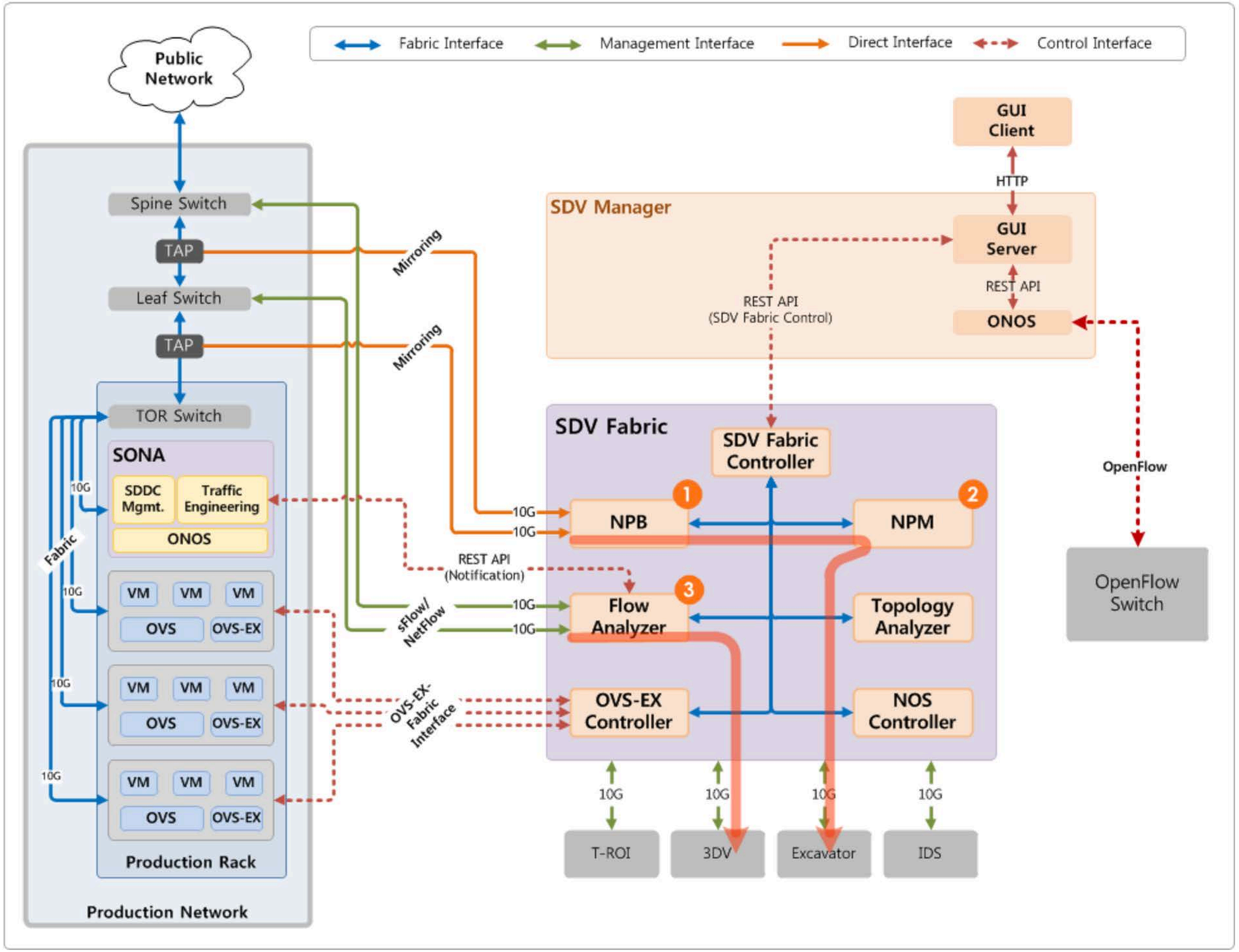
TINA (SKT Integrated Network Analyzer)



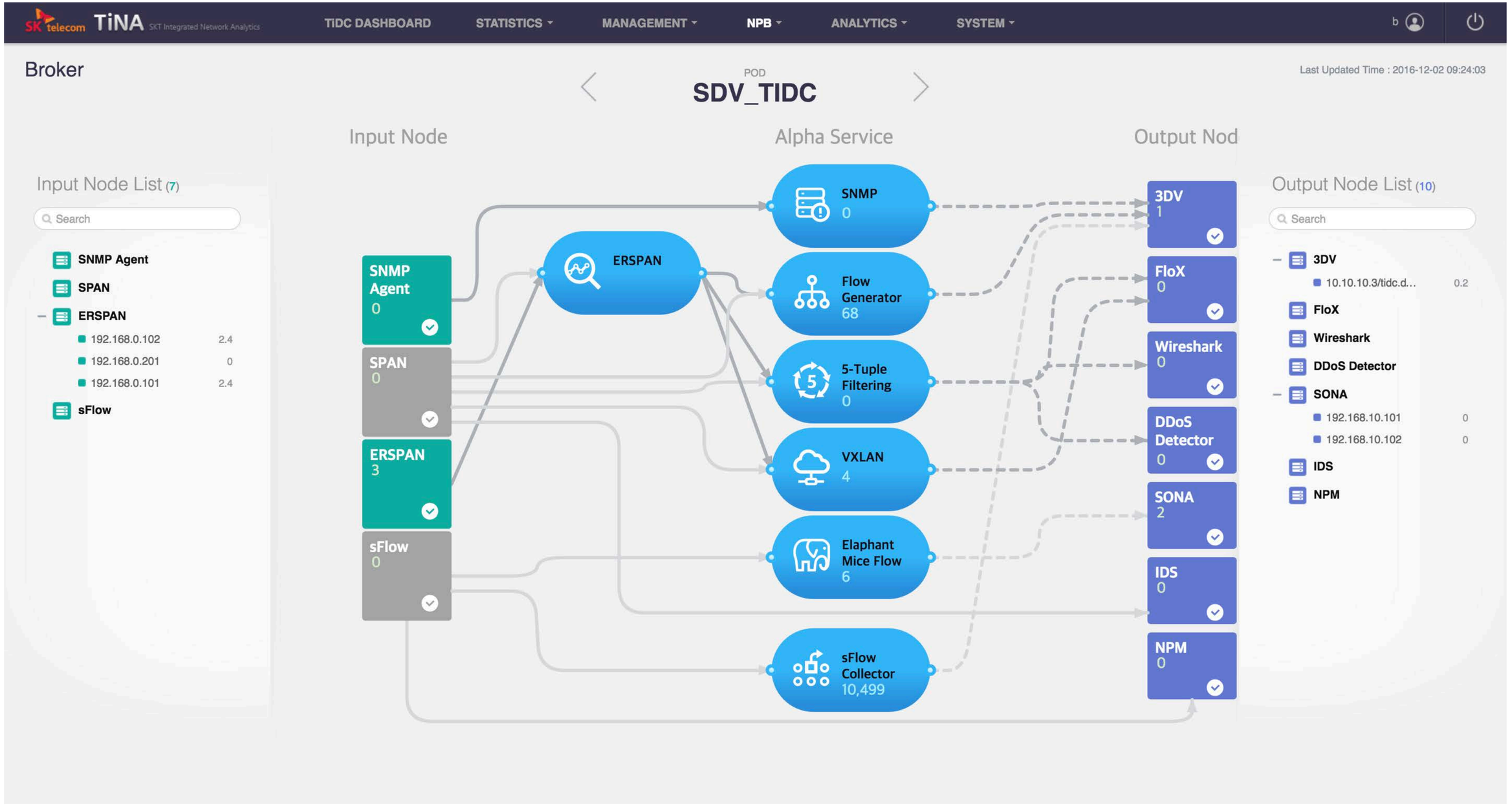
TINA (SKT Integrated Network Analyzer)



TINA (SKT Integrated Network Analyzer)

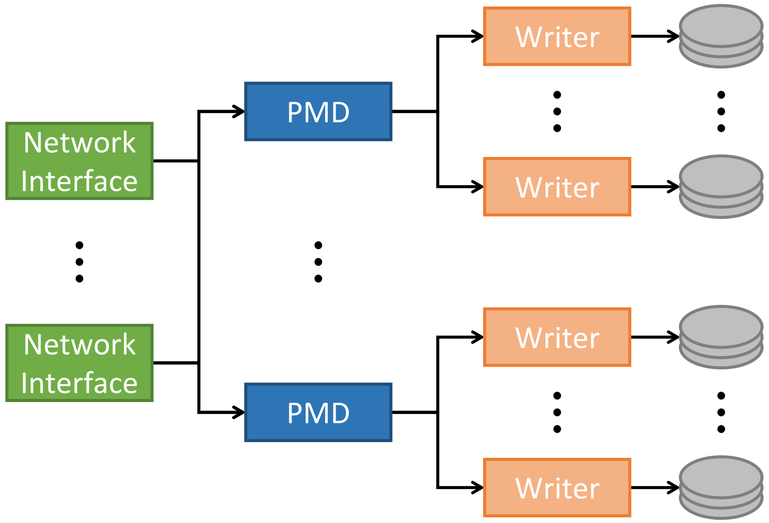


TINA (SKT Integrated Network Analyzer)

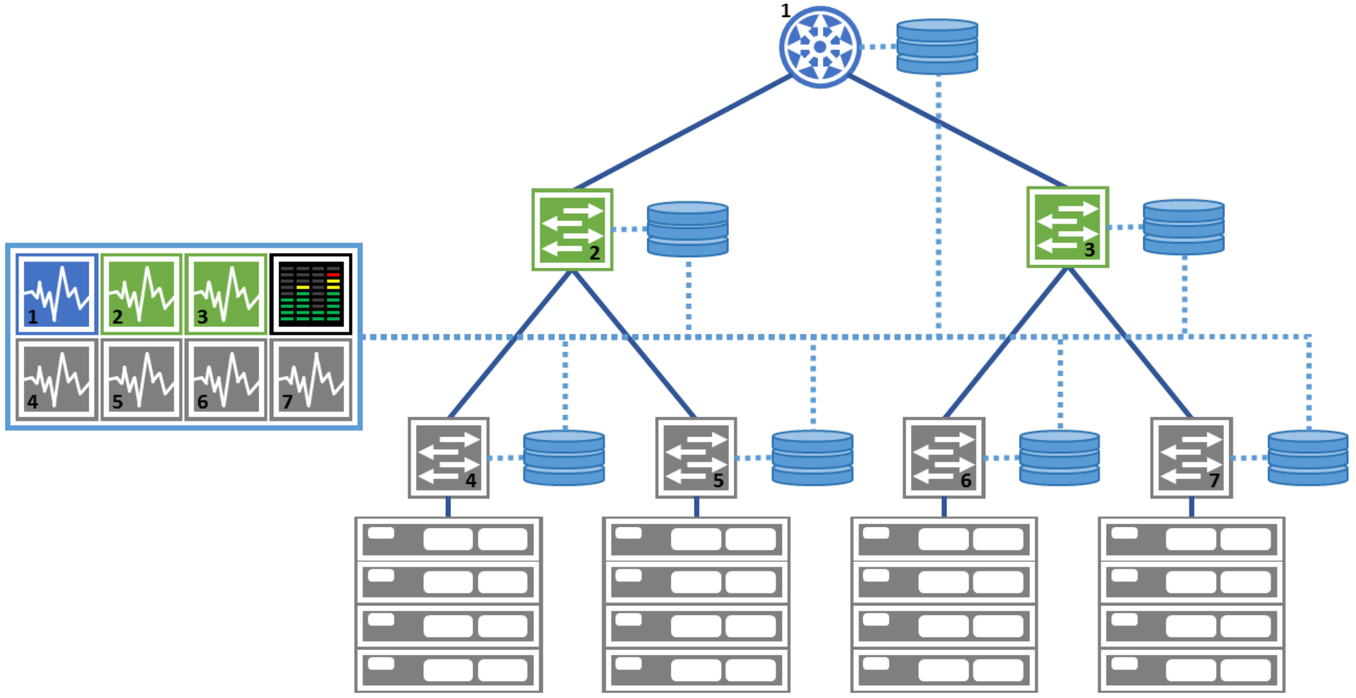


FloX (Flow eXcavator)

- Flow-based packet high-speed packet storage (up to 40Gbs)
- Fast flow search through flow-based indexing, parallel processing and pipelining
- Cluster-mode will be supported for scalability

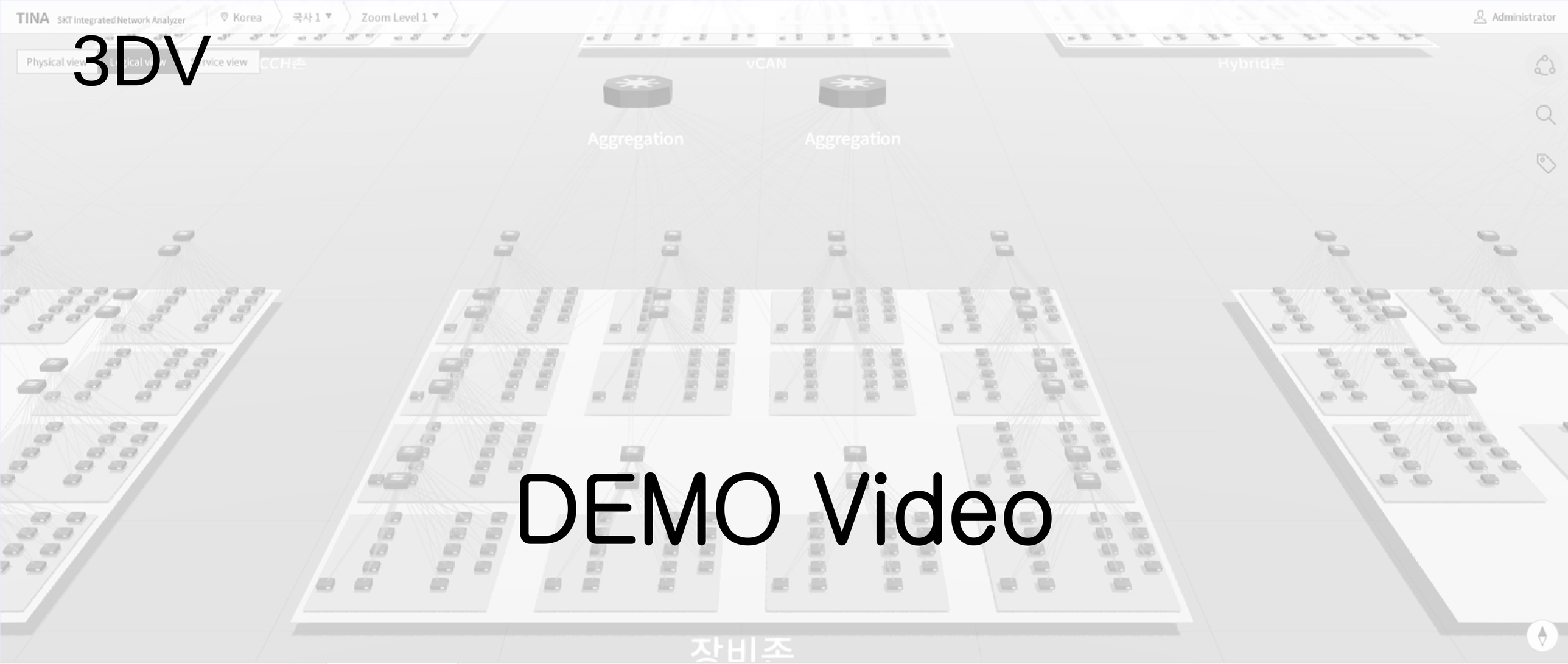


[System Architecture]



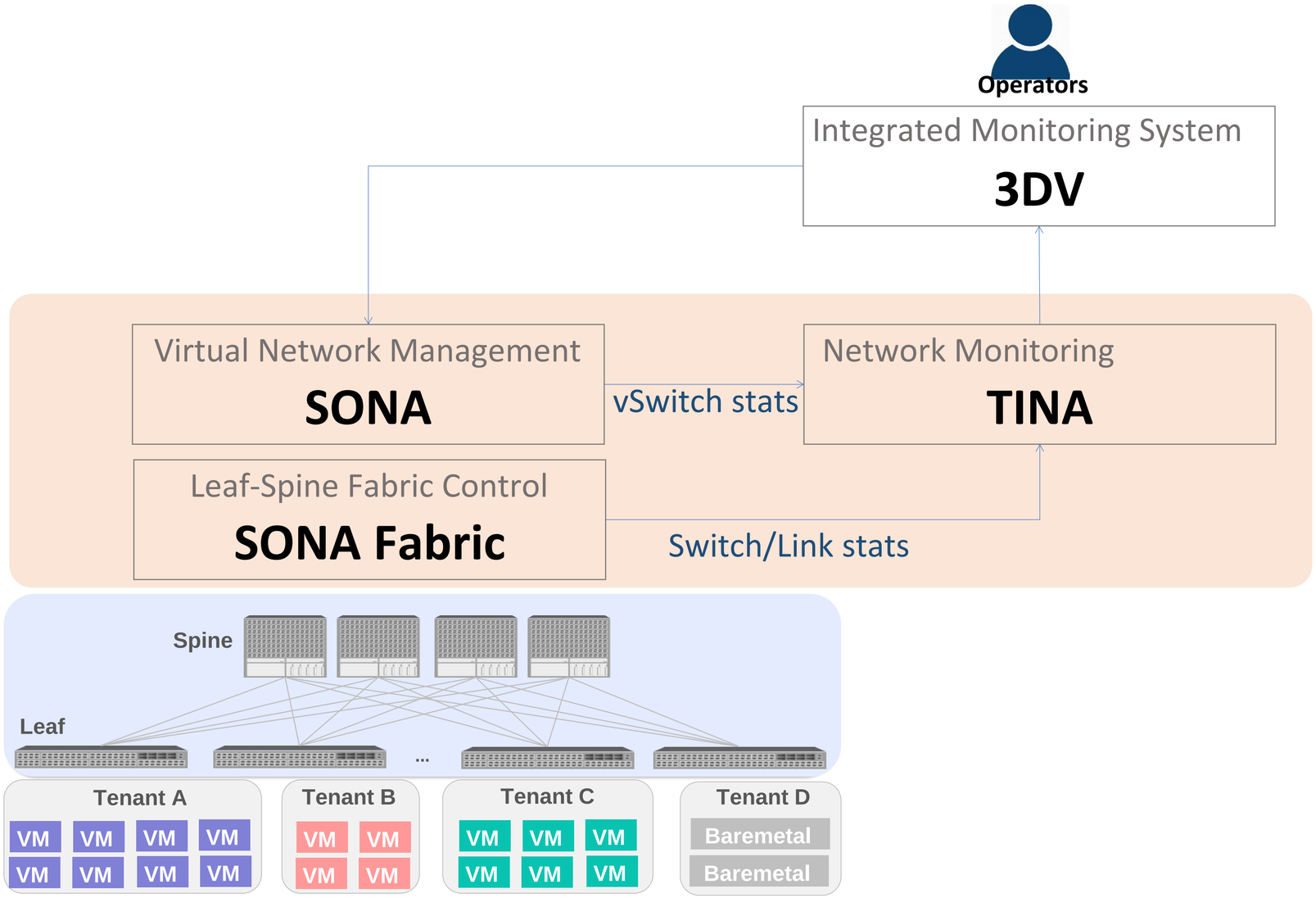
[Deployment Architecture]

3DV



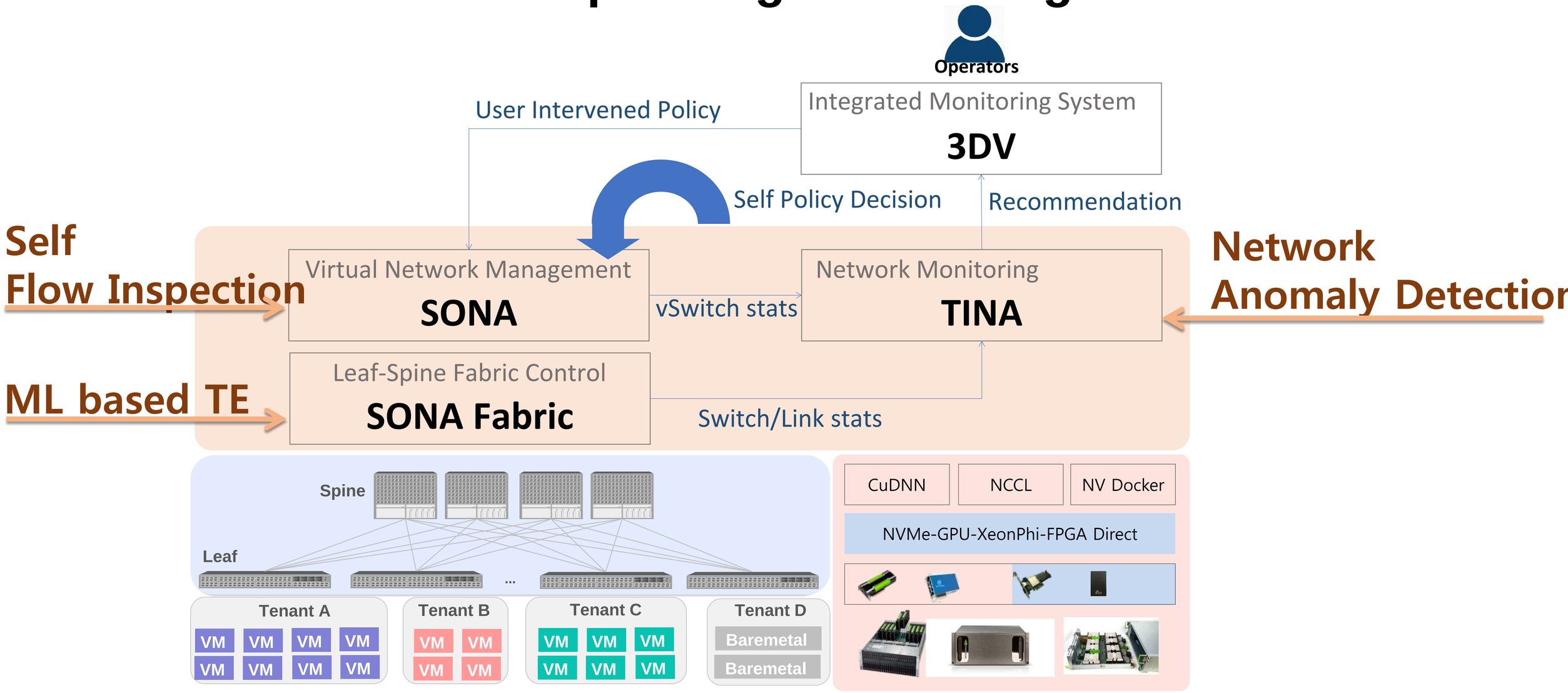
DEMO Video

SDN to SDN: Self Driving Network



SDN to SDN: Self Driving Network

SONA to SONA : Self Operating Networking Architecture



Summary

- SDN to SDN : SDN helps SDN
- Use Case in CORD : A-CORD + SDN
- Use Cases in COSMOS
  - T-CORE
  - TINA
  - SONA

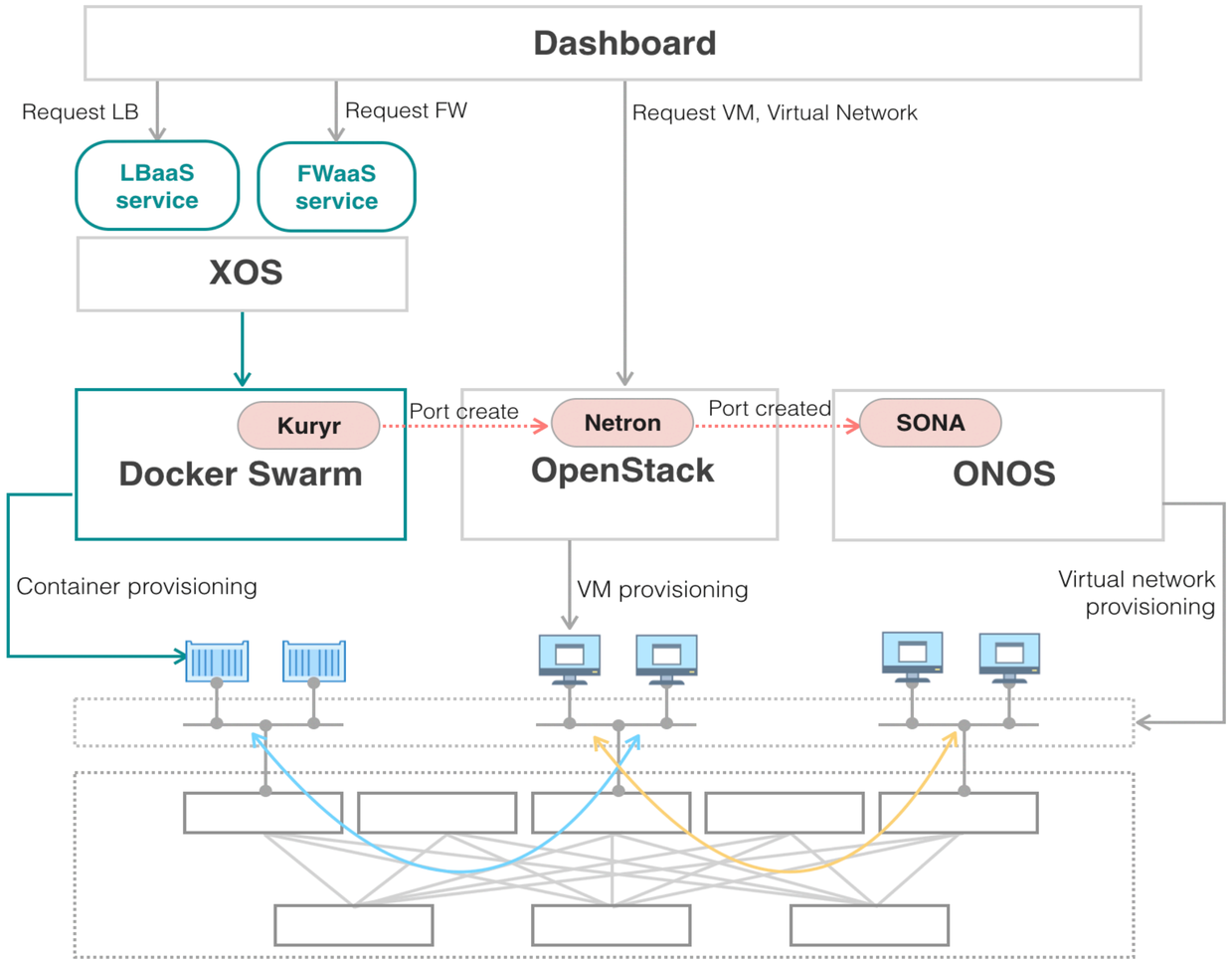
Summary

- SDN to SDN : SDN helps SDN
- Use Case in CORD : A-CORD + SDN
- Use Cases in COSMOS
  - T-CORE
  - TINA
  - SONA
- SONA to **S**elf **O**perating **N**etworking **A**rchitecture

Thank You!

XOS based NFV Framework

1. SONA Fabric
2. NFV F/W with XOS
   - XOS based General VNF Framework
   - Various VNF integration to SONA



Event

NOTIFICATION | HISTORY

Level | EventTime | Target | Code | 완료처리

No Items

Total

≡ 99

DEVICE

Switch	Server	Link	VM
3/0	2/1158	0/2723	0/1155

TRAFFIC

Outbound		Inbound	
0	0%	0	0%

OUTBOUND TRAFFIC TOP 5

google.com	heardontv.com	stylechaja.com	couponjaba.com	coreefrance.com
216.58.199.14	92.243.2.213	112.171.173.159	103.224.212.187에	211.239.150.200

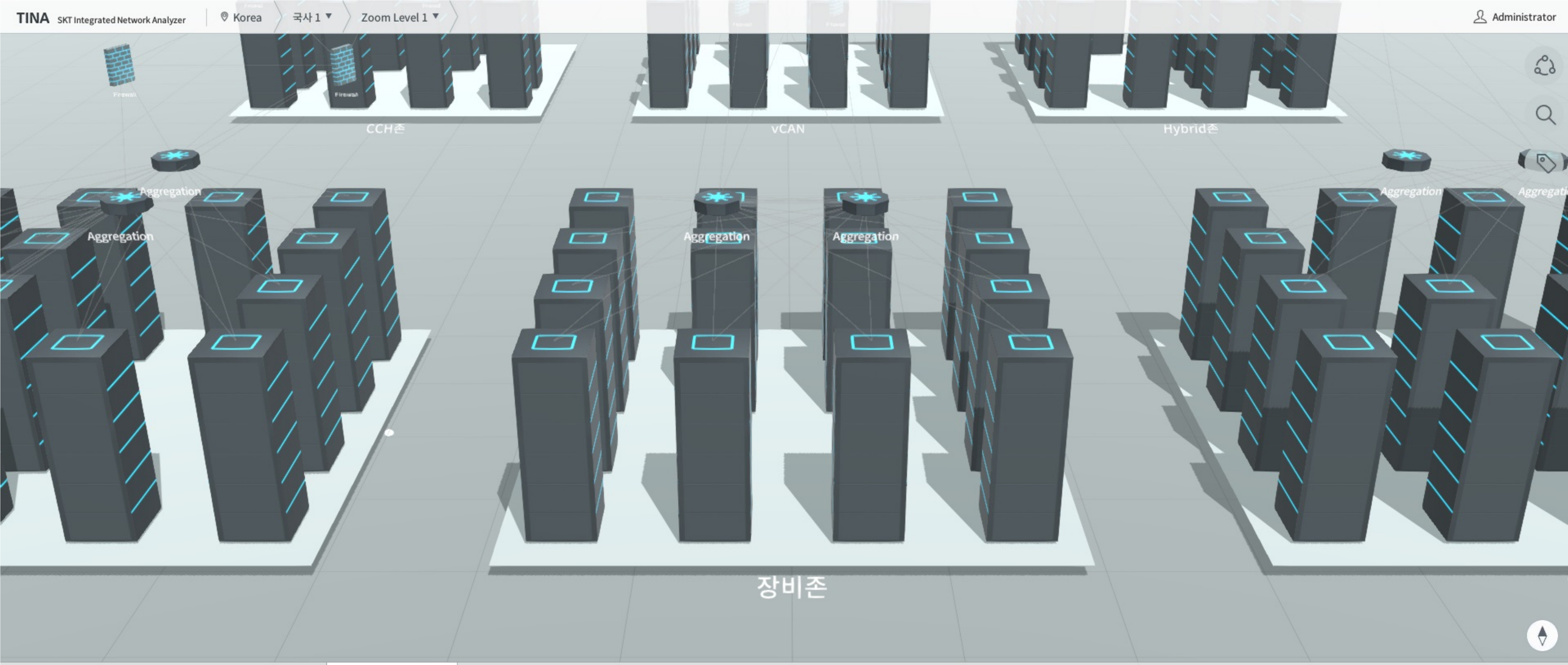
PACKET | THROUGHPUT

Flow Top5

src IP : Port	dst IP : Port	bps
172.168.3.121 : 55898	172.168.3.110 : 5010	453.03M
172.168.3.121 : 55884	172.168.3.110 : 5010	449.84M
172.168.3.121 : 55918	172.168.3.110 : 5010	436.89M
172.168.3.121 : 55882	172.168.3.110 : 5010	414.73M
10.0.92.6 : 20184	10.0.92.14 : 80	130.77M

Host Top 5

IP	I/O	pps
172.168.3.121	OUT	3.82G
172.168.3.110	IN	3.82G
10.0.92.6	OUT	142.96M
10.0.92.14	IN	142.96M
10.0.92.14	OUT	124.61M



Event

NOTIFICATION | HISTORY

Level | EventTime | Target | Code | 완료처리

No Items

Total

≡ 99

DEVICE

Switch	Server	Link	VM
3/0	2/1158	0/2723	0/1155

TRAFFIC

Outbound		Inbound	
0	0%	0	0%

OUTBOUND TRAFFIC TOP 5

google.com	heardontv.com	stylechaja.com	couponjaba.com	coreefrance.com
216.58.199.14	92.243.2.213	112.171.173.159	103.224.212.187에	211.239.150.200

PACKET | THROUGHPUT

Flow Top5

src IP : Port	dst IP : Port	bps
172.168.3.121 : 55898	172.168.3.110 : 5010	454.0M
172.168.3.121 : 55882	172.168.3.110 : 5010	449.75M
172.168.3.121 : 55918	172.168.3.110 : 5010	445.34M
172.168.3.121 : 55884	172.168.3.110 : 5010	436.24M
10.0.89.3 : 22	10.0.89.9 : 7193	139.63M

Host Top 5

IP	I/O	pps
172.168.3.121	OUT	3.86G
172.168.3.110	IN	3.86G
10.0.92.6	OUT	201.48M
10.0.92.14	IN	201.48M
10.0.92.14	OUT	155.77M