

# CORD Test Project in Okinawa Open Laboratory

Fukumasa Morifuji  
NTT Communications

Transform your business, transcend expectations with our technologically advanced solutions.

# Agenda

- **VxF platform in NTT Communications**
- **Expectation for CORD**
- **CORD Integration & Test Project in Okinawa Open Laboratory**
- **Current Status and Future Plan**

# VxF Platform in NTT Com

## ■ Goal

- Making platform that optimize network dynamically according to changes of application and traffic trend

## ■ Key factor of VxF Platform

- Controller/Orchestrator
- Underlay fabric
- NFV Service Pool

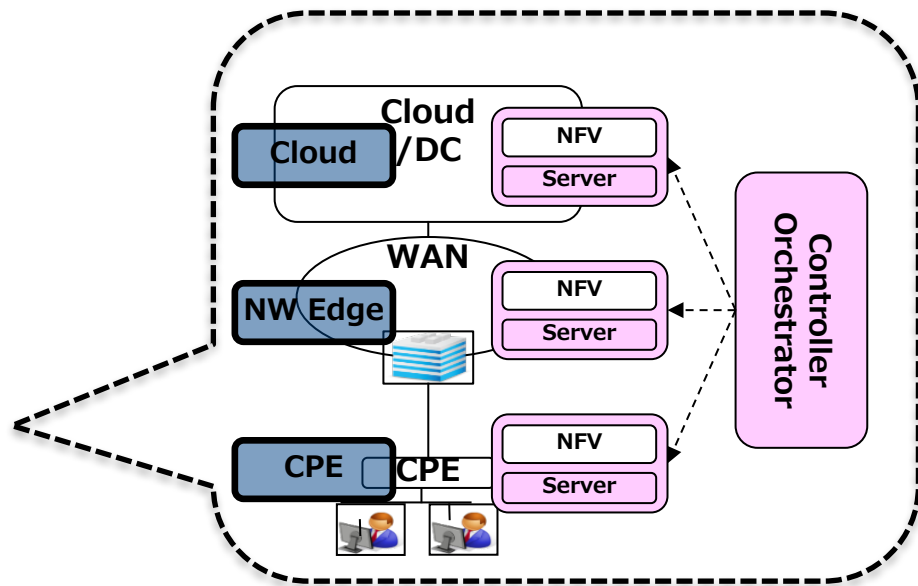
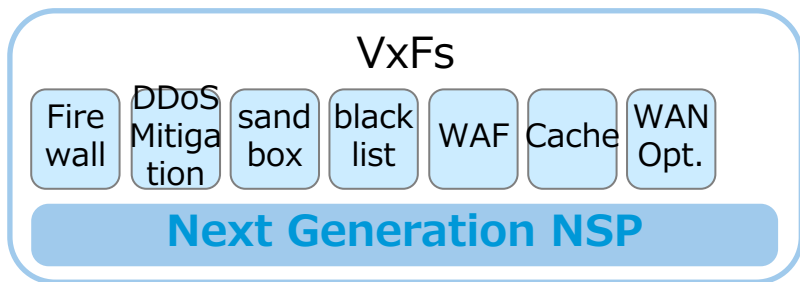
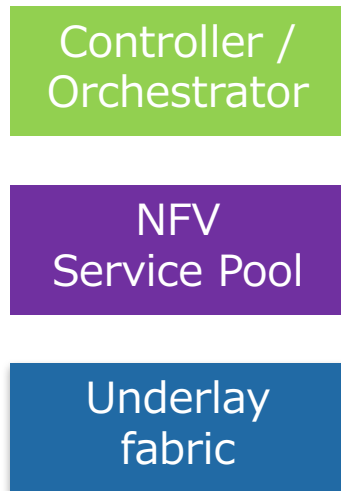


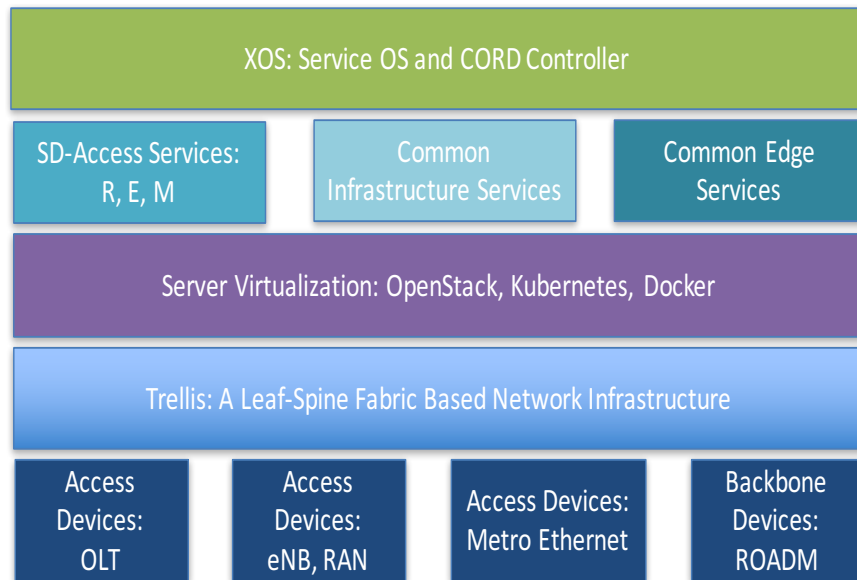
Fig. Overview of VxF Platform

# CORD as a reference architecture for VxF Platform

## VxF Platform



## CORD Architecture



# Evaluation topics from commercial deployment perspective

## Functional Requirement

- L2, L3 Forwarding
- Tenant Isolation
- Traffic Engineering
- etc...

## non-Functional Requirement

- Easy installment
- Document
- Usability
- Operability
- Availability
- etc...

# CORD Test Project in OOL

## ■ Motivation

- Integrate, test and use CORD Implementation in OOL from commercial deployment perspective
- Share knowledges and problems to be solved with the communities

## ■ Plan

- Define evaluation items from commercial deployment perspective (completed)
- Integrate CORD environment in OOL(completed)
- Test CORD Implementation(in progress)
- Summarize test results(not started)

# Okinawa Open Laboratory

## ■ Foundation

- Founded on May 8, 2013

## ■ Location

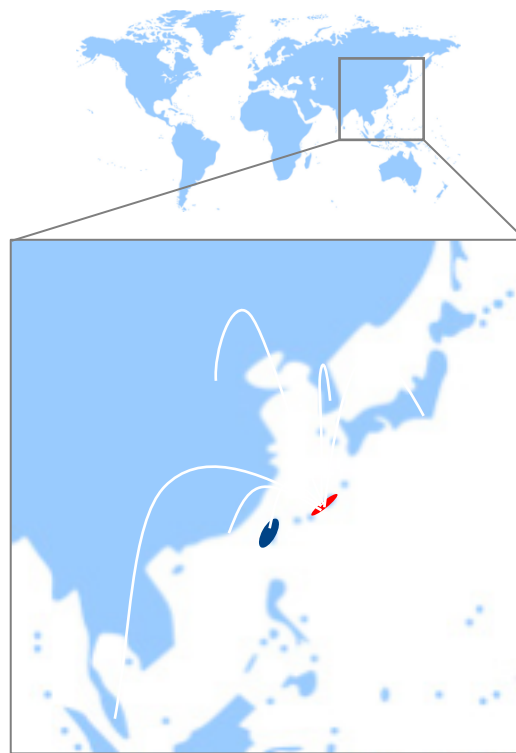
- Okinawa, Japan

## ■ Web

- <http://www.okinawaopenlab.org/en/>

## ■ Purposes

- Research and development activities for practical application and dissemination for SDN, NFV, cloud computing.



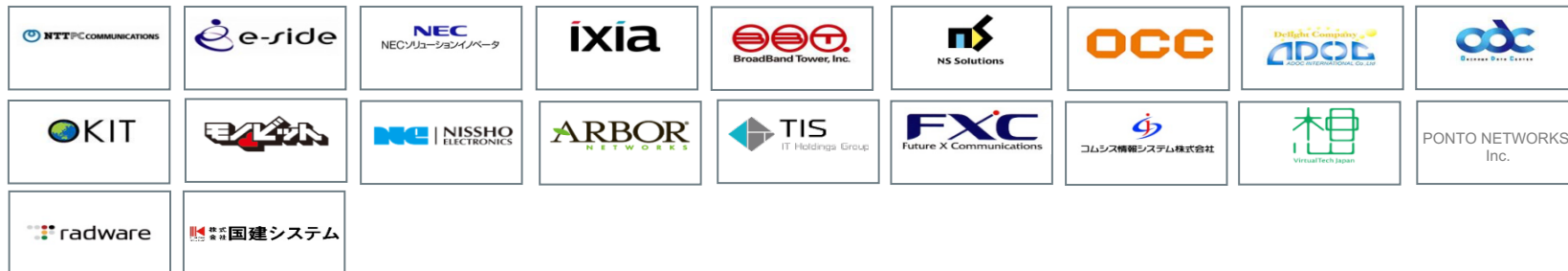
# Members

## Board Members (6)

August 2017



## General Members (20)



## Fellow Members (21)



# Evaluation Topics (Excerpt)

Category	Item	Application	Checkpoint			User information	Common	ユーザVLANを持ち込める事(VLAN変換できる事) ユーザがすでに使っているVLANIDを利用し続けることを想定
1 High availability	machine failure	Ctl	コントローラが全断(=ONOSクラスタが全部死ぬ場合)でも既存トラヒックが継続される事				Common	-ユーザmacアドレスを持ち込める事
		Ctl	コントローラが全断でも既存VMから既存VMへの新規トラヒックが生成される事				Common	ユーザIPアドレスを持ち込める事
		Ctl	コントローラはクラスタを組めること。冗長度は2ないしそれ以上である事(VM上に展開可だとより良い)				Common	ユーザにOpenStack環境をそのまま提供できる事
	Load	SW	OpenStackノードからLeafスイッチの冗長度は2ないしそれ以上である事(例:複数台でのMC-LAG)			Move detection	Common	ユーザにVMWare環境(vCD/vcenter)環境をそのまま提供できる事
		Ctl	冗長系のコントローラが片系故障した場合に切り替わること。				Common	VMotionによりmacが別IF/別筐体/別系に移動した場合に、移動が検知できテナントにマッピングできる事 (http://qiita.com/ukinau/items/1ebe20c8f5dad49a9b0b)
		Ctl	冗長系のコントローラが片系故障した場合に切り替わった際に既存・新規トラヒックに影響が無いこと。				Common	VMotionによりIPが別IF/別筐体/別系に移動した場合に、移動が検知できテナントにマッピングできる事
		SW	冗長系のSWの片系故障の場合に切り替わること			Provisioning	SW	VMotionによりIPが別IF/別筐体/別系に移動した場合に、Default GWを最速経路となるよう自動設定できる事
		SW	冗長系のSWの片系故障した場合に切り替わった際に既存・新規トラヒックに影響が無いこと。			Isolation	SW	ZTP(Zero Touch Provisioning)が可能なこと 1筐体に複数のIFより同一のVLAN/トラヒックが来た場合にIsolation出来ること (「ユーザVLANを持ち込める事」と同じ内容)
		Common	コントローラのノードの処理負荷が系全体に影響を与えない事(BCMC処理が特定のノードに集中しない事) BUMの処理はどうなっているのか?				SW	同一プレフィックスの別セグメントを分離できるか テナント単位でトラヒックがIsolation出来ること (「テナントにより管理単位を設定できる事」と同じ)
	VerUP	SW	特定のSWの処理負荷が系全体に影響を与えない事(GWにつながっているノードなど)			Traffic Engineering	Common	ECMPでトラフィックを分散処理できること
		Ctl	コントローラのバージョンアップ時にトラヒック/SOIに影響無い事(NSSU:Non Stop Software Upgrade)				Common	トラフィックエンジニアリングが可能なこと(1.RSVP/Petcoによる帯域リザーブ、2.Fast Reroute)
	Link detection Fault avoidance	SW	スイッチのバージョンアップ時にトラヒック/SOIに影響無い事(NSSU)				Common	パフォーマンス(遅延、帯域、パケロス)によって経路の選択ができること
2 Functionality	BUM	Common	系のトポロジ自動検知が出来ること			LAG	SW	①複数台でMC-LAGが組めること。 ②ハッシュのアルゴリズムなどにより特定のパスにトラヒックが偏らない事。 (Leaf側に外部機器を接続する場合など) (OpenStackノードからLeafスイッチの冗長度は2ないしそれ以上である事)
		Common	系のトポロジに異常(変化)(ポートダウン、電源断)が生じた場合に、検知できること					LDR(LinkDownRelay)のON/OFF設定が出来ること
		Common	系のトポロジ異常時にトラヒックを自動的に迂回できること・自動迂回時にパケロスが無い事(もしくは少ない事)					Leaf側のUpLinkがすべて死んだ際、DownLink側を積極的に落とすのか?
		Common	特定のリンクのトラヒック(機器交換時など)をオペレータが手動迂回できる事・手動迂回時にパケロスが無い事			LDR	SW	バケットキャプチャがLeaf側の任意のIFに対して行えること。バケットキャプチャ設定時にSWの再起動が不要なこと。既存トラヒックへ影響を与えない事。
		Common	ブロードキャストドメインを任意に分割できる事			キャプチャ	SW	バケットキャプチャが任意のIFに対して他筐体で行えること。
	Link detection Fault avoidance	Common	①VmotionなどでBUMツリーの再構築が行われるか?(BUMの処理はどうなっているのか?)				SW	任意のフローについて起点・終点及び経由しているノードを特定し可視化できる事。中継のノードも特定し可視化できること。
		Common	②その場合に後続の処理に影響が無い事(ARP処理や外部接続機器のVRPPのAdvertise開始など)			可視化	Common	Leaf側IF毎にL2/L3/L4(若しくは更に上位レイヤレベル)でQoS設定が出来ること(外部装置のIn/Outともに)unicast
		Common	Split Horizonなどによりループ回避が可能なこと				SW	IF毎にL2/L3/L4(若しくは更に上位レイヤレベル)でQoS設定が出来ること(系内部=Spine~Leaf間のIn/Outともに)
		SW	ループ検知が出来、ループガード設定が出来ること(ONOS自体にあるか?CORD制御アプリ側か?)				SW	外部からのパケットのToS/CoSにより系内部のQoSマッピングできる事unicast中心
		Common	同じセグメントのIPアドレスが異なるスイッチに設定できるか			QoS	Common	フロー単位で系内を通してQoS制御が可能なことunicast中心
		Common	外部機器の冗長制御トラヒック(VRRP/HSRP/PetcoのAdvertise)への影響を与えない事(L3のマルチホームなどで対応etc)				SW	B/U/Mで個別に帯域制御(QoS制御)が可能なこと
2 Functionality	Management	Common	テナントにより管理単位を設定できる事(Overlay側現状はQinQ)			ACL	SW	Leaf側外部接続側IF単位にACL機能(MAC/IP/Portによるバケットフィルタリング)が可能なこと
		Common	1ユーザに複数テナントを配置できる事				SW	Leaf側外部接続側IF単位にACLにより該当するバケットをカウンタリングできる事
	User information	Common	複数コントローラを跨いでテナントを設定できる事(まずはONOSクラスタでスイッチの管理上限はあるか?)			NB連携	Ctl	上位APIとしてRest/Netconf/CLI/Yang etcが提供できる事(仕様確認)
		Common	ユーザVLANを持ち込める事(VLAN変換できる事)				Ctl	上位コントローラでOpenStack(Neutron)/NSX(XCenter)/ODLとの連携ができる事
		Common	ユーザがすでに使っているVLANIDを利用し続けることを想定			VerUP	Common	ONOSとSWで複数世代(3代)間でのVerUPが可能なこと
		Common	-ユーザmacアドレスを持ち込める事				Ctl	VerUPが出来なかった場合に自動でロールバックし、既存/新規トラヒックに影響が無い事
		Common	ユーザIPアドレスを持ち込める事				SW	スケジュールで(自動で)VerUP出来ること
		Common	ユーザにOpenStack環境をそのまま提供できる事					

# Evaluation Topics (Availability)

## ■ ONOS Cluster

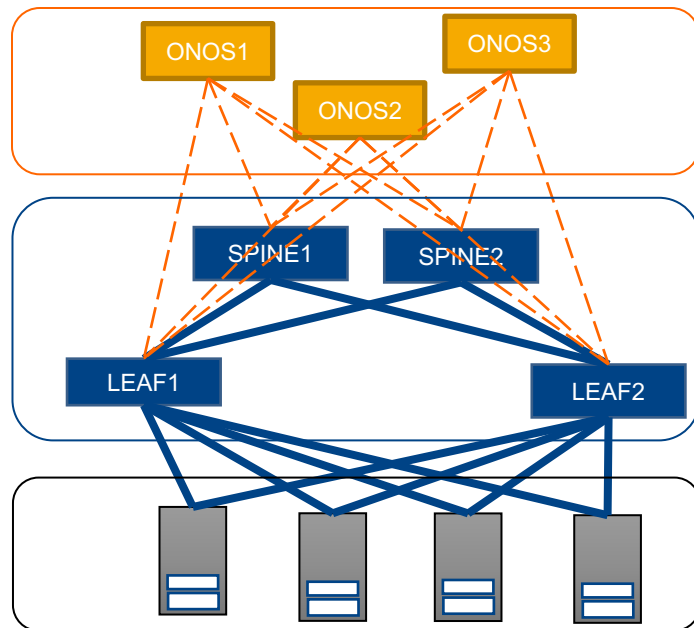
- Fabric controller can be clustered in CORD
- Traffic is not affected when one of the clustered controllers is down

## ■ Fabric Switches

- Traffic is not affected when one of the clustered controllers is down

## ■ Compute node

- Live migration of virtual machine is available



# Evaluation Topics (Scalability)

## ■ interconnectivity of multiple pods

- Operation multiple controller with single orchestrator

## ■ The number of SW and Compute node is easy to modify

- ZTP for SW and compute
- # of SW / Compute nodes

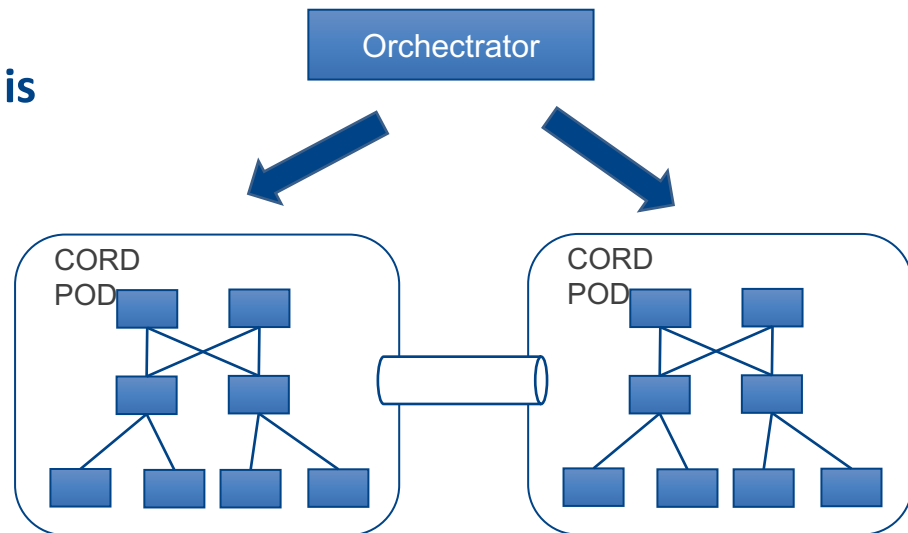


Fig. Interconnect multiple CORD POD

# Evaluation Topics (Usability / Operability)

- **Easy to install/operate**
- **Documentation**
- **Support by community**

# Test environment

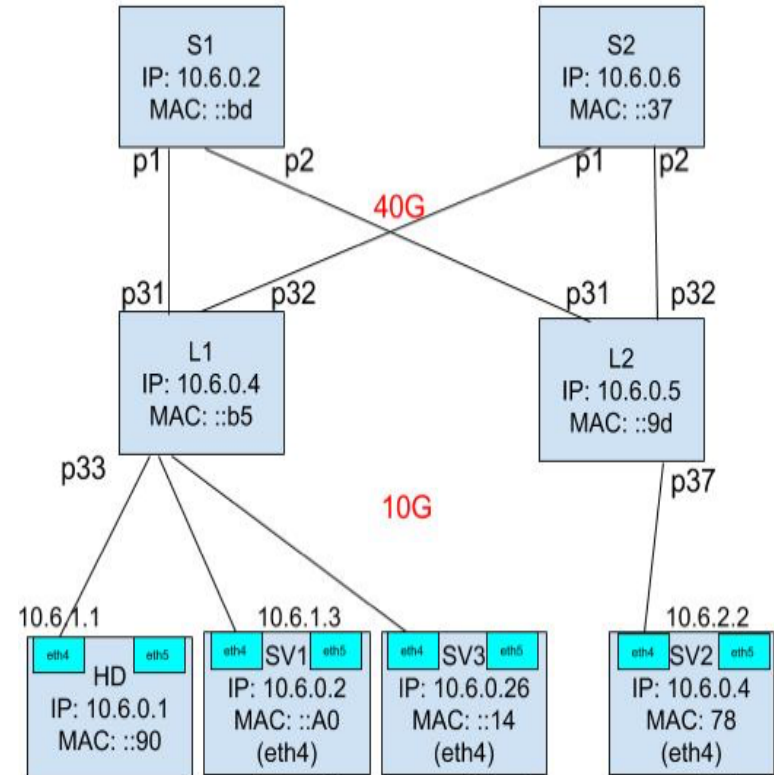
## Switches

- AS6712-32X
- 2-Spine,2-Leaf

## Servers

- 32GB+RAM
- 8+ CPU Core
- 200GB+ disk
- 10G NIC x 1, 1G NIC x4, IPMI/BMC Port x 1

✓ using 40G x 1 – 10G x 4 DAC cables



# Current status

## ■ Deployment completed

### ■ Connected to build system in ONF using VPN

- ✓ Successfully deployed CORD4.1
- ✓ Begin test from now on

# Summary

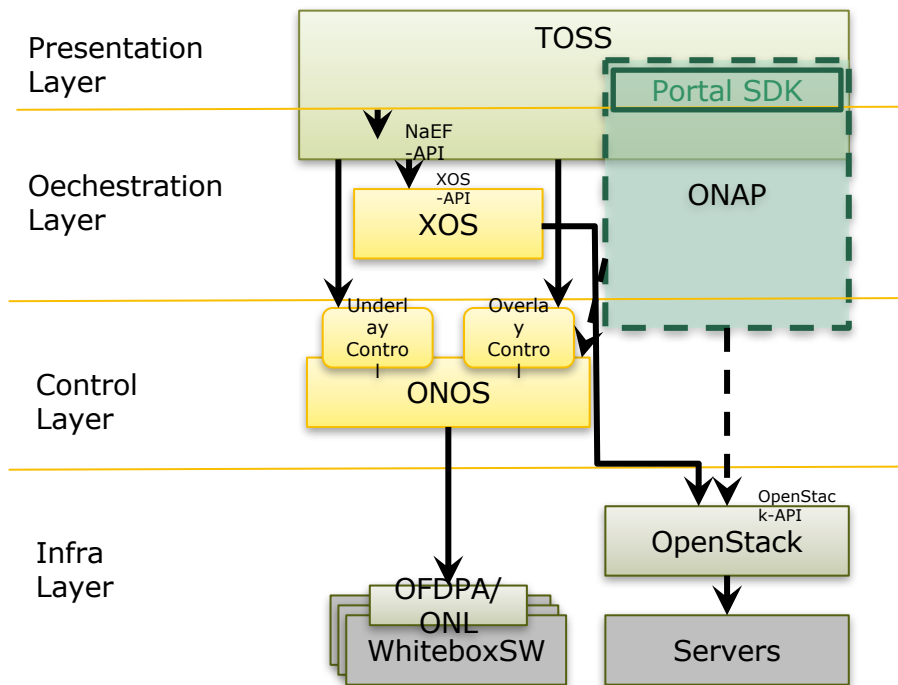
- **NTT Com aims to realize VxF Platform**
- **CORD is expected as reference architecture for VxF Platform**
- **Testing CORD in OOL**
- **We finally deployed test environment after a lot of trouble**
- **We will work on testing CORD from commercial deployment perspective**
  - **In addition, we evaluate CORD in other two projects.**

# Next step

- **Deploy CORD Implementation into OOL Testbed**
  - Evaluate usability, operability, quality and availability through long term CORD operation in OOL
- **E-CORD Field Trial between NCTU and OOL**
  - Evaluate multi CORD POD connection

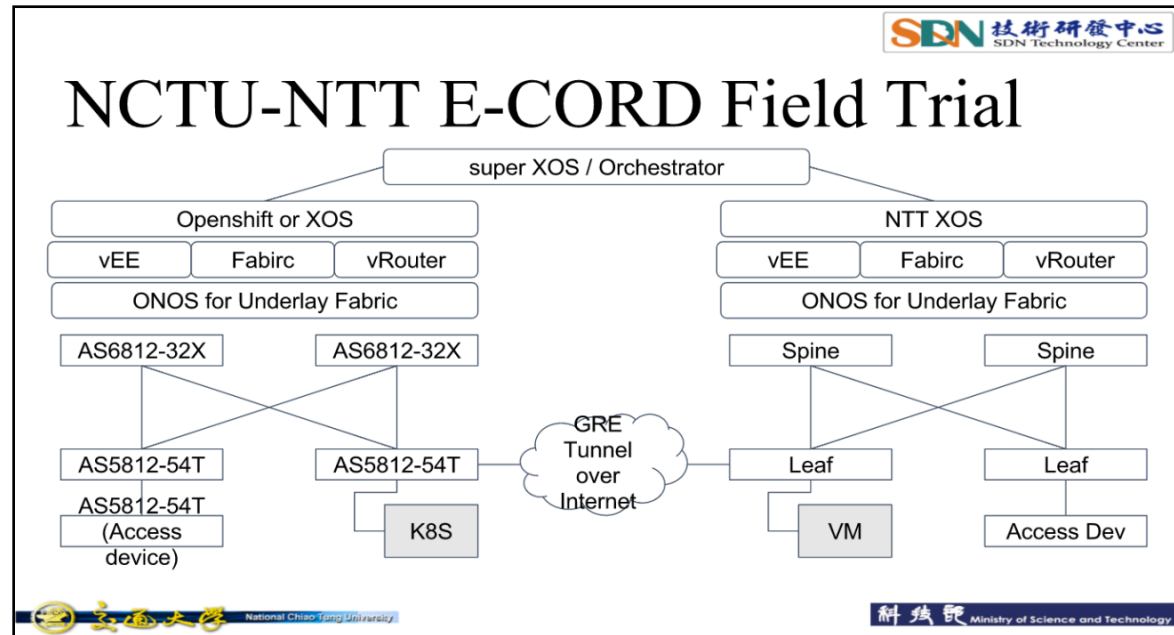
# Deploy CORD into OOL Testbed

- OOL Testbed is open testbed for OOL members so that they can easily try new technology.
- We evaluate usability, operability, quality and availability through long term CORD operation in OOL Testbed
- Integrate CORD with operation support system in OOL



# E-CORD Field Trial between NCTU and OOL

- Connect two CORD POD
- Investigate how to manage multiple CORD POD



# Toward Production Deployment

## ■ Requirements for production level

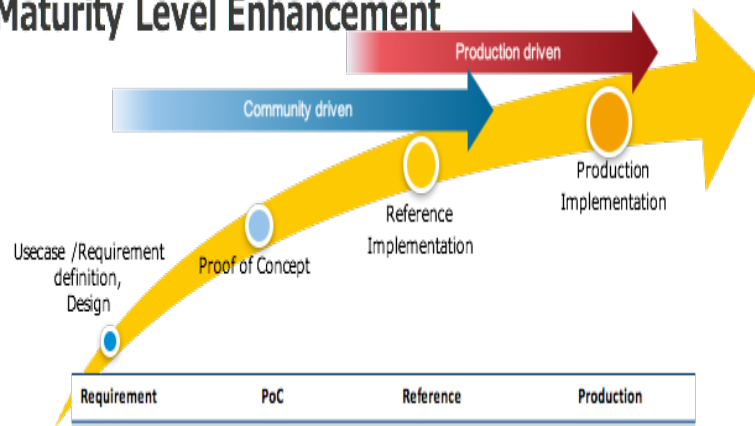
- Functions
- Easy installment, Documents
- Usability/Operability
- Availability, Performance

## ■ OOL will contribute to

- Testing CORD
- Making feedback to community
- Sharing knowledge
- Spreading to OOL members

### Expectations for ONOS/CORD

### Maturity Level Enhancement



Requirement	PoC	Reference	Production
Easy installment, Documents		👍👍	👍👍👍
Functions	👍👍	👍👍	👍👍👍
Usability/Operability	👍	👍👍	👍👍👍
Performance	👍	👍	👍👍👍
Quality/Availability	👍	👍	👍👍👍

Elegant 👍👍👍

Sufficient 👍👍

Minimum 👍

N/A

# Thank you!