



Enterprise CORD Hands-On

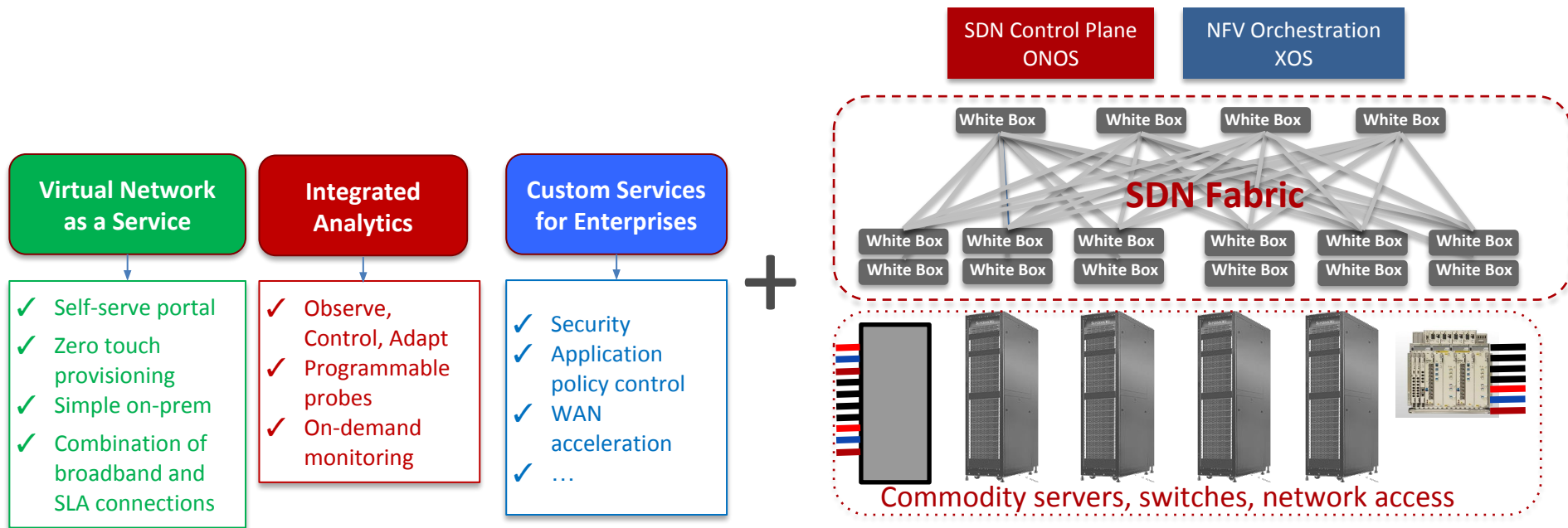
Marc De Leenheer, Andrea Campanella, ONF

CORD Build, OCT headquarters, San Jose
November 10, 2017

Outline

- E-CORD recap
- Building E-CORD
 - Wiring diagram
 - Configuration
- Developing for E-CORD
 - Dev environment
 - Carrier Ethernet
- Continuous Integration / Deployment
 - Jenkins
 - E-CORD Test Subscriber
- E-CORD Guide
- Q&A

E-CORD Value Proposition



Carrier-grade Network as a Service

Built on an open platform

Bring data center economy and cloud agility

Building E-CORD POD

Start with [bill of materials](#)

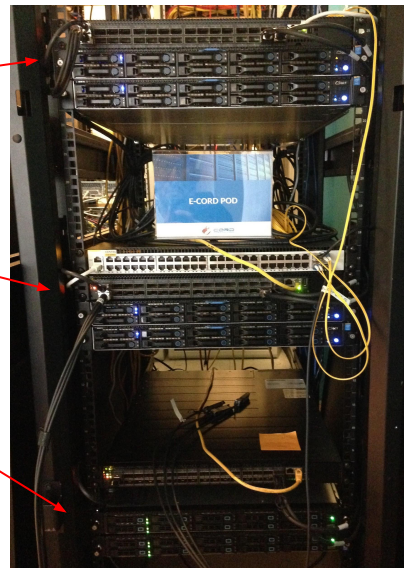
Follow the wiring diagram

Configure the PODs

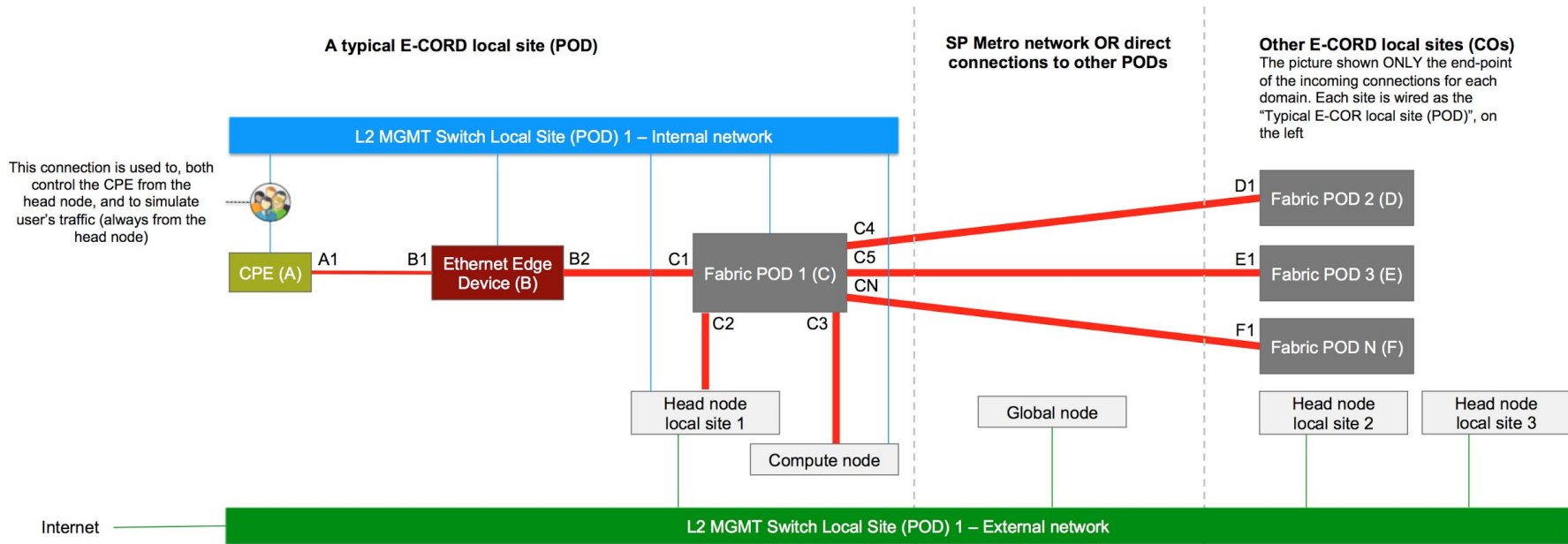
Service provisioning

Bill of Materials (ONF pod)

- **Used for dev and data plane testing**
- **3 local PODs**, each one is
 - CPE: 1x Microsemi EA1000
 - Ethernet Edge: 1x Centec V350
 - Fabric: 1x Edgecore AS6712-32X
 - Head node and compute node
 - 2x Quanta (QCT) QuantaGrid D51B-1U
- **1 global node:**
 - 1 x Quanta (QCT): QuantaGrid D51B-1U



Wiring Diagram



Local POD Configuration

JSON Files

CPE info
Links
Global endpoint config

Fabric cross connect
Global endpoint config

<http://YOUR-LOCAL-POD-IP:8182/onos/v1/network/configuration>

<http://YOUR-LOCAL-POD-IP:8181/onos/v1/network/configuration>

Local CORD POD

Access ONOS

Fabric ONOS

CPE

EE

Fabric

Global Node Configuration

YAML

Bandwidth profiles
ONOS endpoint
UNI information

JSON

Local sites location

Global CORD Node

```
python run_tosca.py 9000  
xosadmin@opencord.org YOUR_XOS_PASSWORD  
PATH_TO_YOUR_TOSCA FILE
```

XOS

```
http://YOUR-LOCAL-POD-IP:8182/onos/v1/ne  
twork/configuration
```

Global ONOS

Developing for E-CORD

CORD relies heavily on git/repo for version control and gerrit for code review

- <https://gerrit.opencord.org>

Setting up dev environment can be done in hardware or software

Critical software piece is Carrier Ethernet app for orchestration and provisioning

Development Environment Overview

E-CORD services are defined in the profile manifests

- Global node: [ecord-global.yaml](#)
- Local pod: [ecord.yaml](#)

These services are deployed in either software or hardware environment

Software: CORD-in-a-Box (CiaB) or developer's machine

Hardware: BOM

Software Development Environment

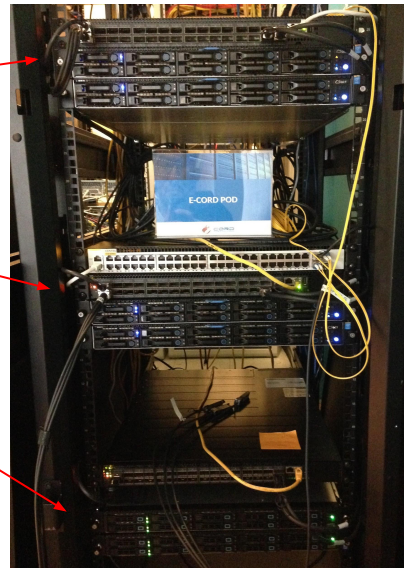
Environment choice depends on required feature set

Software deployments uses CORD-in-a-Box (CiaB) or your local machine

- `ecord-global/local-single.yaml` → your machine
- `ecord-global/local-mock.yaml` → your machine frontend/API only
- `ecord-local-virtual.yaml` → CiaB with OpenStack

Hardware Dev Environment (ONF)

- **Used for dev and data plane testing**
- **3 Local PODs**, each one is
 - CPE: 1x Microsemi EA1000
 - Ethernet Edge: 1x Centec V350
 - Fabric: 1x Edgecore AS6712-32X
 - Head node and compute node
 - 2x Quanta (QCT) QuantaGrid D51B-1U
- **1 Global Node:**
 - 1 x Quanta (QCT): QuantaGrid D51B-1U



Hardware Dev Environment (QCT)

- **Used for build, deployment and API tests only**
- **1 Local POD**
 - Fabric: Edgecore AS6712-32X
 - Head node and compute node: 2x Quanta (QCT): QuantaGrid D51B-1U
- **1 Global Node**
 - 1 x Quanta (QCT): QuantaGrid D51B-1U

Carrier Ethernet Application

- **Carrier Ethernet is the app we use for global and local orchestration**
- **Common APIs** `org.opendaylight.ce-api`
- **Global** `org.opendaylight.ce.global`
- **Local** `org.opendaylight.ce.local.bigswitch,org.opendaylight.ce.local.channel.http`
 - Fabric** `org.opendaylight.ce.local.fabric`
 - Cord** `org.opendaylight.ce.local.vee`
- **OAM/CFM functionality included in ONOS & driver behaviour (covered this afternoon)**

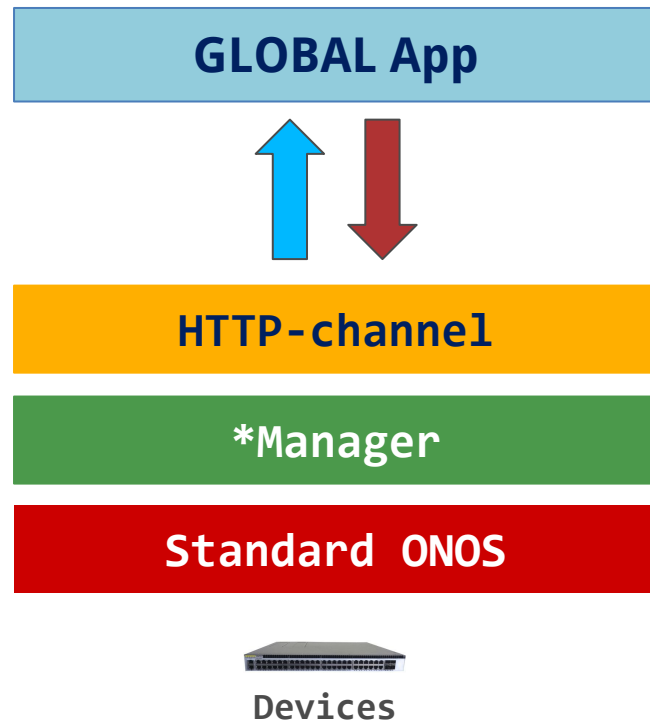
Carrier Ethernet Application

*Manager class translates forwarding constructs (FC) into hardware specific configuration.

Access VeeManager → FC to Metering in EE and vlan tagging into CPE

Fabric FabricManager → FC to segment routing config

Transport TransportManager → FC into OpticalConnectivityIntent



Adapt to your network through a new manager

Jenkins: Continuous Deployment and Testing

- **Automated Nightly Deployment**

- All ONF Pods and global node
- All QCT Pods and global node

- **Automated Nightly Testing**

- XOS and Service installation
- ONOS applications present and configured
- Fabric, Head-Nodes, Compute-Nodes Provisioning

- 1 local POD and one global node **builds and test always happen nightly**



Jenkins



Constant feedback on the
status of the software



No regressions
Functionality maintained

TEST E-CORD Subscriber

Test the E-CORD local service chain:

- *cd /opt/cord/build/platform-install*
- *ansible-playbook -i inventory/head-localhost --extra-vars
"@/opt/cord_profile/genconfig/config.yml" ecord-test-subscriber-playbook.yml*

E-CORD Guide

Overview:

<https://guide.opencord.org/profiles/ecord/overview.html>

Installation and Demo:

https://guide.opencord.org/profiles/ecord/installation_guide.html

Partners and Collaborators



Partners



Future work and collaboration opportunities

Services, services, services

- E.g., firewall, WAN accelerator, encryption, ...
- Open and closed source versions

Device Integration

- CPE
- Ethernet Edge

Multi-Access CORD

- {R,E,M}-CORD service chains co-existing in the same pod

ONAP Integration

Further Reading

CORD website:

<http://opencord.org>

Tutorials, documentation and general reading at:

<https://wiki.opencord.org/> and <https://guide.opencord.org>

CORD is on Github at:

<https://github.com/opencord>

ONOS Transport wiki:

<https://goo.gl/UiMauo>

Mailing List:

cord-dev@opencord.org

cord-discuss@opencord.org

By email:

marc@opennetworking.org

andrea@opennetworking.org

Questions

