



# ONAP Deployment: Possible Scenarios, Problems/Showstoppers, Solution Proposal

Abinash Vishwakarma, Netcracker

# Going on production environment with ONAP : Context Setup

*The real challenges of ONAP production deployment is still need to be collected and might vary based on business plans, size etc. This presentation is an attempt to identify what CSPs might expect in a production scenario and how we can address those expectations*

## ONAP deployment Scenarios (using Green Field and Brown Field terminology):

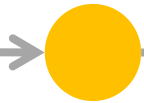
- *Green Field : No existing virtualization orchestration function, ONAP is deployed for the first time.*
- *Brown Field : There are other orchestrators already deployed and ONAP is added as another components.*

# Deployment Stages : ONAP for green/brown field

Plan to adopt SDN/NFV    First ONAP Deployment



Green Field



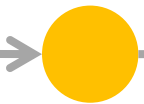
Able to integrate ONAP with existing **OSS/BSS applications.**

Plan to deploy ONAP

First ONAP Deployment



Brown Field



Able to integrate ONAP with existing **OSS/BSS applications and Other SDN/NFV functional components.**

## Driven by Business plans

Expand by adding another ONAP Instance – **Geographic location demand , edge computing**

Add or Upgrade component of ONAP, like SO, APC etc. - Enhance performance

Remove an ONAP instance or component of ONAP, like SO, APC etc.- closing a data center in a region; or for edge site management)

Add another orchestrator from Vendor or Opensource - **Business demand**

- Integrate the new ONAP instance with the OSS/BSS.
- Integrate the new ONAP instance (for new geographical area or business unit) with existing ONAP instance.
- Add/Upgrade a functional component to an ONAP instance , less impacting the overall deployment.
- Add another orchestrator from Vendor or Opensource.
- Remove an ONAP instance without impacting the other ONAP deployment.

**Optimized architecture for fully automated processes, continuously available and support agile/DevOps mode of operation**

- Architectural Flexibility
- Optimal utilization of resources and reuse
  - Scalable on demand
  - Minimize integration effort

# Challenges and Approach

Implementing ONAP as  
Green Field

Implementing ONAP as  
Brown Field

## Challenges

Adapt to SPs business plans

Maintaining the business continuity

Integration with existing applications

Continuously meeting the performance requirements

Minimizing integration and migration cost

Technology and Vendor agnostic

## Approach

Continuous delivery  
demand

The deployment need to adapt to SPs business demands. Enabled by inherent ONAP architecture.

ONAP as a box Vs ONAP  
components/applications

ONAP must support horizontal /vertical scaling and upgrade of components based on SPs business plans .

Support various  
Deployment options

Use case Driven  
Platform Driven  
Capability Driven

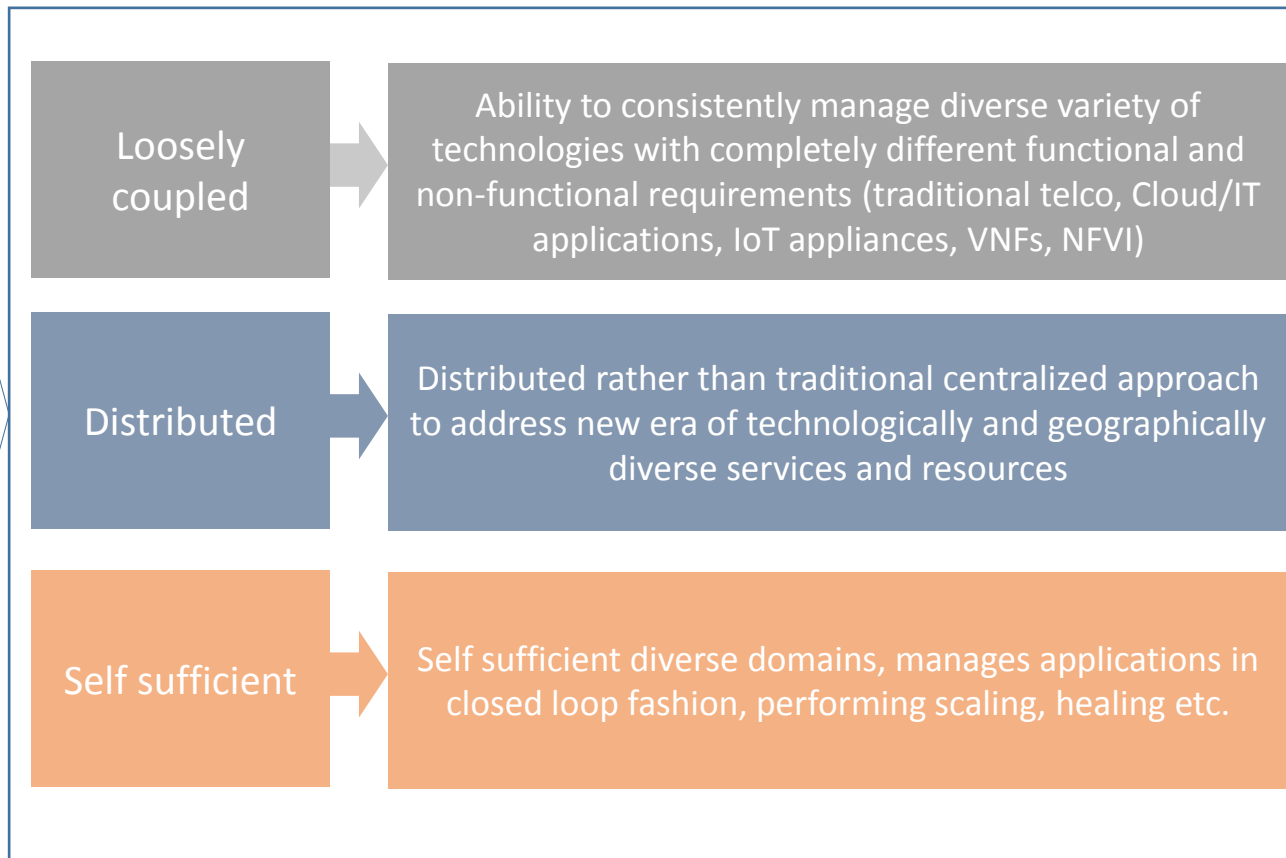
# Challenge and Approach

## Deployment Approach

One time Vs  
Continuous activity

ONAP as a box Vs  
ONAP  
components/applications

Support various  
Deployment options



## Deployment Challenges

Integration with existing applications

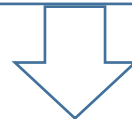
Adopt to SPs business plans

Technology and Vendor agnostic

Maintaining the business continuity

Minimizing migration

Continuously meeting the performance requirements



1. Domain Orchestration (DO)
2. Componentize ONAP for future transformation

# What is Domain Orchestrator?



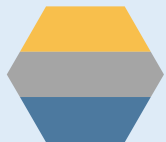
Self-sufficient distributed architecture building block fully covering Domain services fulfillment, assurance and control



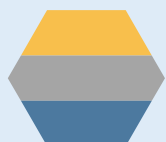
## What is “domain” here?



Technological (e.g. NFVI, SDWAN, IP/MPLS)



Service (e.g. L3 VPN, VoIP)



Geographical



## Key attributes

- **Closed-Control Loop** for domain services
- Service **intent** driven interactions
- Fully **programmable**
- Open APIs
- **Cloud** native

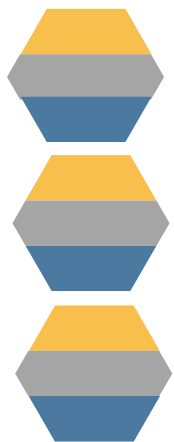
## Benefits:

Minimized Time-to-Market of new Services (changes are localized within single DO borders)

Optimized for Agile/DevOps work (one DO ~ one DevOps team)

Higher availability and resiliency due to no “single point of failure”

# ONAP as configurable Domain with its components



Service (e.g. L3 VPN, VoIP)

Service Domain Specific ONAP Instance

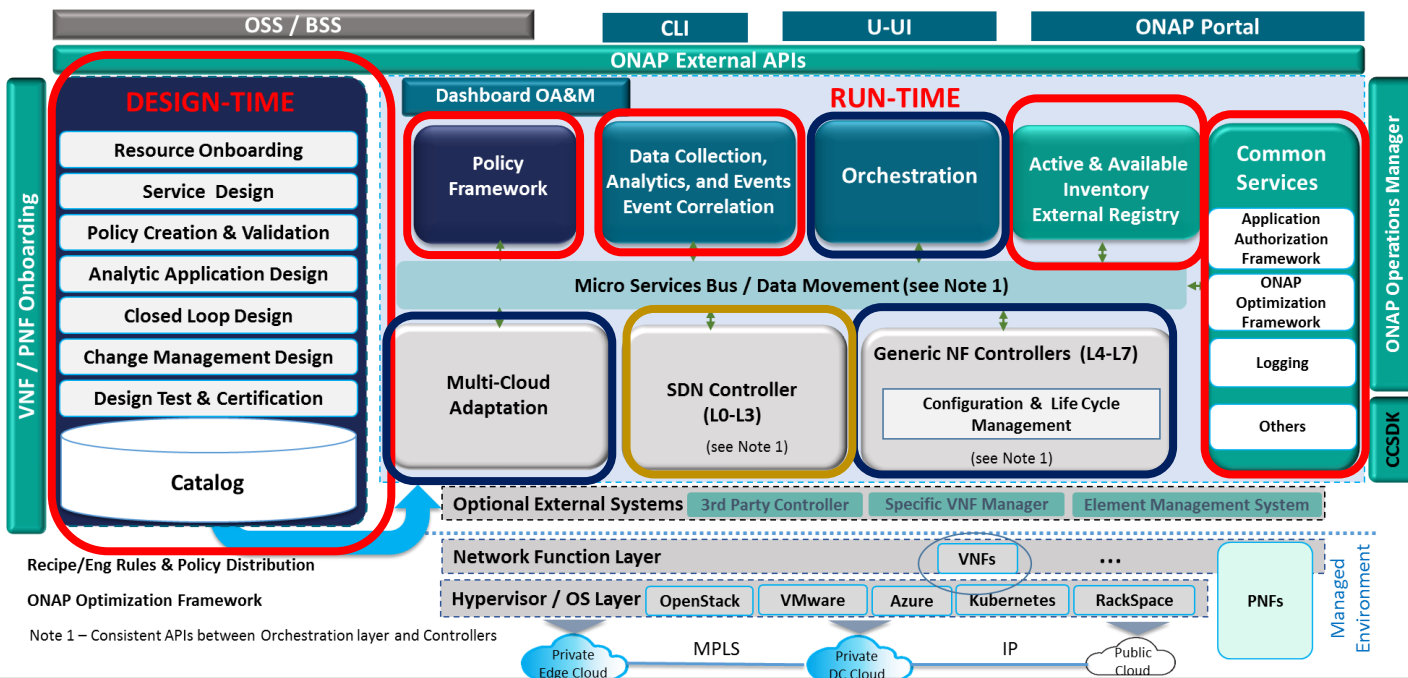
Technological  
(e.g. NFV,RAN,  
IP/MPLS)

Technology Domain Specific ONAP instance

Geographical

ONAP Instance for each geography

- ONAP itself can be DO
- Considering complete ONAP deployment to achieve Domain might not be viable solution for all use cases

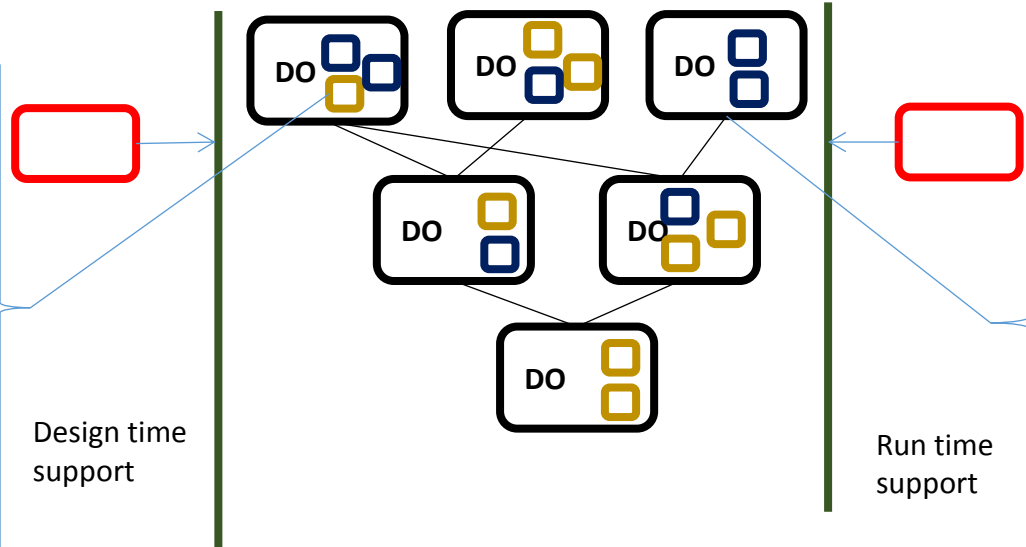


- Can be considered as Operational Management Support applications. Provides or collects information from other functional components of ONAP.
- Can support more than one instance of ONAP.
- Service level orchestration.
- Can support more than one technology specific domains, or specific for a domain
- The functional component can serve more than one services too.
- Reusing functional components for more than one service is deployment choice
- Technology dependent.
- Can directly be managed by Service level DOs.

# Bringing together DO and ONAP

## ONAP Functional Components

- Loosely coupled, modular ONAP functional blocks.
- Each ONAP application can be Scaled without impacting the other applications. Scaling of ONAP application is transparent to other ONAP functional block.
- Exposes its capabilities via **open API, ensuring that APIs are aligned to standards.**
- Able to manage itself and lifecycle of components orchestrated by it.
- Implemented as micro-services
- Full automation of micro-services lifecycle including auto-healing/scaling



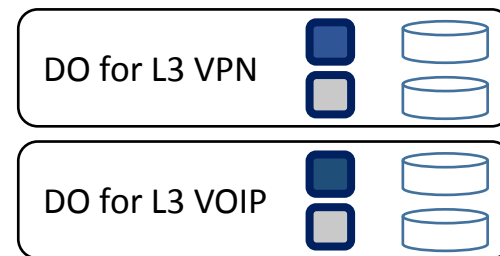
## Domain Orchestrator

- Enabler for adapting ONAP to SP business plans.
- Composition of ONAP and Non ONAP functional components.
- Self sufficient to handle the domain specific services needs fulfillment, assurance and have knowledge of immediate sub domain/s.
- Can share ONAP functional components if required.
- Implementation specific architecture flexibility to share the resources such as inventories, orchestration functional blocks etc..
- Can directly or indirectly communicates with Domain Support components.
- DO <-> DO communication is “**service intent**” based
- DO behaviour/models should be fully “**programmable**” via standard APIs

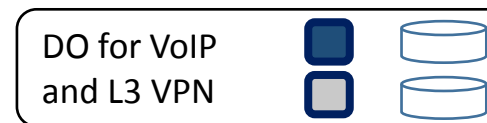
## Visualizing ONAP for VOIP and VPN Domains



### Separate DO for VOIP and VPN



### Combined DO for VOIP and VPN



Implementation as per SP Plan

Run time support  
(Catalog, inventory, policy, Analytics)



Operational Management Support can support DO implementation as per SP plan

Operational Management Support can support DO implementation as per SP plan



## **Self-sufficient and loosely coupled modular ONAP functional block**

- Loosely coupled, modular functional blocks of ONAP will ensure that each components can be deployed as per business needs.
- This will ensure that ONAP when deployed will have no impact or less impact on other existing applications.

## **DO can be composed of components based on business needs**

- Possibility of having ONAP and non ONAP applications as part of one DO.
- Component of DO can be configured and scaled individually without impacting the other ONAP or non ONAP component.

## DO's Management and Operational support functions

- Can support ONAP and non ONAP applications.
- Can be reused across more than one DOs.
- Exposes APIs to interface with DOs

*For example SDC can be used for design of VOIP and VPN Domains*

## **Standardized API**

- Standardized API (For example External API) for building bigger solution of multiple DOs
- Interface between DOs and other Non-ONAP systems like legacy OSS or WAN controllers to achieve seamless operability.
- Interoperability at ONAP component level.