Modularity for Integration Usability
## Problem Statement

- ONAP is **too complex, too big** and hard to make changes.
- ONAP Components are **monolithic** (SDN-C, SO) and large, not sharing common utilities.
- Service providers might have a specific module already implemented and would like to **integrate** that module into ONAP.
  - External controllers (e.g. VNFM, SDN Controller), external orchestrators, collectors, analytic microservices.
- Service providers would like to deploy ONAP **incrementally**, whereas today ONAP supports **all-or-nothing approach**.
  - Core components of ONAP such as SDC, SO, and A&AI must be deployed.
  - Other components can be added on as needed basis, depending on the scope of use.
- Should ONAP modules migrate to **cloud-native microservices**?

*Can incorporate additional issues and/or more details if available*
Modularization - 1

- Aligned working assumption on terminology:
  - **Module**: Implements a business capability accessed through a defined set of APIs
    - E.g. A DCAE Data Collector microservice, A&AI data repository
  - **Component**: A collection of modules that are related in some form
    - E.g. SO, Controllers, A&AI, etc
  - **ONAP**: A collection of ONAP Components
  - **Microservice**: Small, single-capability focused, standalone services
    - E.g. IP address assignment, Tosca parser
  - **Cloud-Native**: Container-packaged, dynamically managed, microservices-oriented applications
    - E.g. Containerized microservices managed by Kubernetes
  - **Service Mesh**: Connective tissue between microservices
    - E.g. traffic control, resiliency, security, observability
    - Control plane (Istio, linkerd) and Data plane (Envoy, linkerd)
    - Note: This is different from service chaining

- Aligned working assumption on approach
  - Evolutionary approach
  - One component at a time
  - Start with SO and Controllers
Modularity - II

• SO Decomposition working assumption
  - API handler
  - Request DB
  - BPMN Infra
  - SDC controller
  - Catalog Adapter
  - Adapters for the controllers (SDNC/VFC/…) and
  - Cloud Adapter

• Controller decomposition working assumption:
  - Extract IP assignment from the controllers as a common microservice
  - Extract Tosca Parser from SO and make a common microservice

• Feedback welcome to mature to working assumption for Dublin.
  - Will discuss in Project meetings
  - Will share with PTLs in PTL meeting
Current progress of Modularity and Our question

<Our recognition of progress>

• As the above-cited slides, several issues of Modularization were discussed on Dublin Architecture Planning meeting.

• Modularity for incremental deployment is **very important for consumption**.

• **So, we’d like to see the current progress** of Modularity for Frankfurt release and further releases.

<Question>

• How is the **current progress on modularity (Microservice)** of each PT?
• **What is the future plan for modularity?**
Talk to us (on Modularity Questions)

Contact:

Naoki TATEISHI +81 80 2096 4681 (international mobile/voice only) naoki.tateishi.mz@hco.ntt.co.jp
Ken KANISHIMA +81 80 2096 6077 (international mobile/voice only) ken.kanishima.md@hco.ntt.co.jp

Team contact: onap-evaluation-p-ml@hco.ntt.co.jp