ONAP – Road to Complete Automation

Milind Jalwadi, Andreas Geissler

Tech Mahindra Ltd., Deutsche Telekom
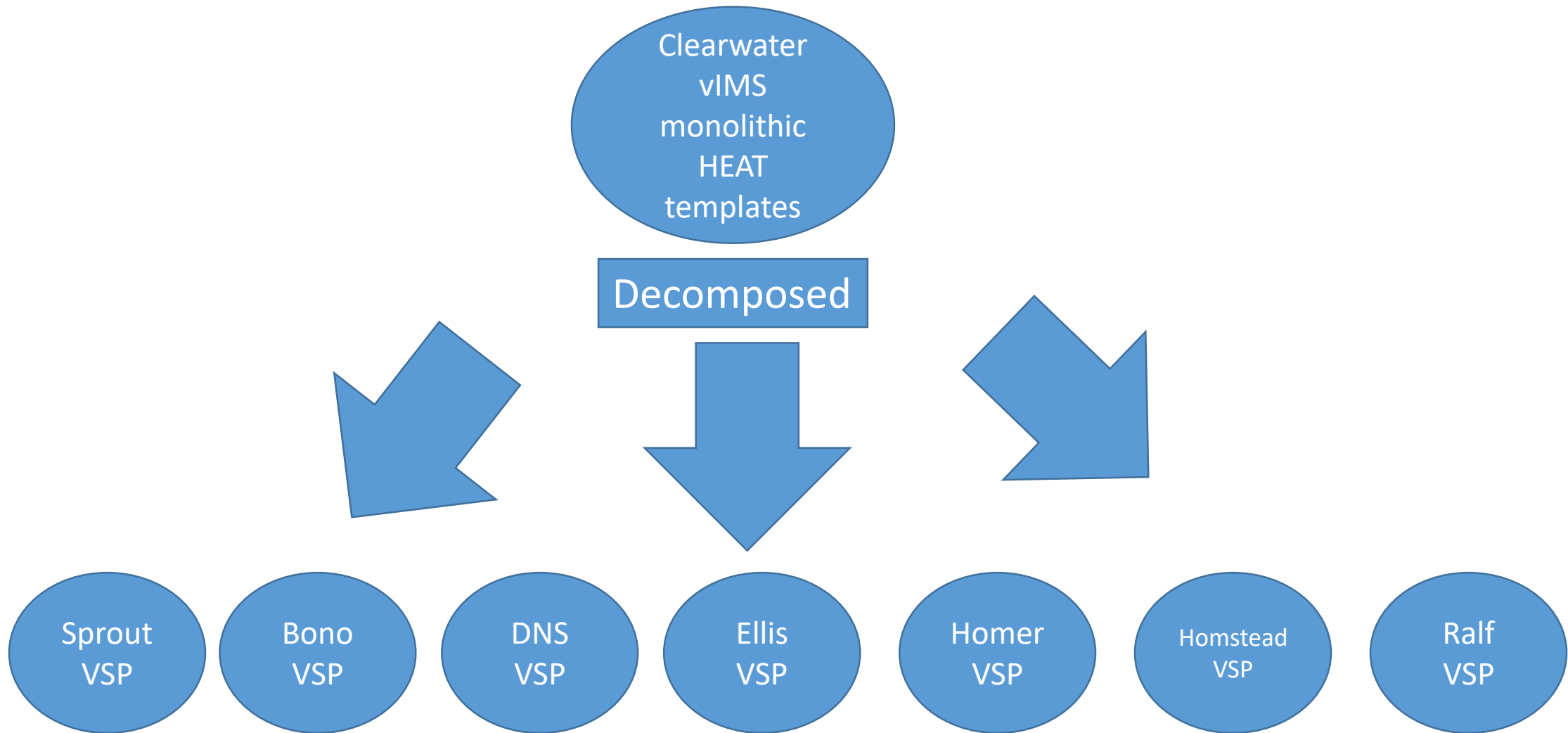
- 10th Jan, 2019

Can I?

- Onboard non ONAP use case VNF artifacts and that it would get on-boarded and configured seamlessly without any modifications to SO work flows?

- Instantiate VNFs without any "manual" steps?

- Perform seamless scale-out operations - again without many "manual" steps?

- On-board VNFs with complete TOSCA definitions and orchestrate the same using SO and configure it using APP-C / SDN-C?
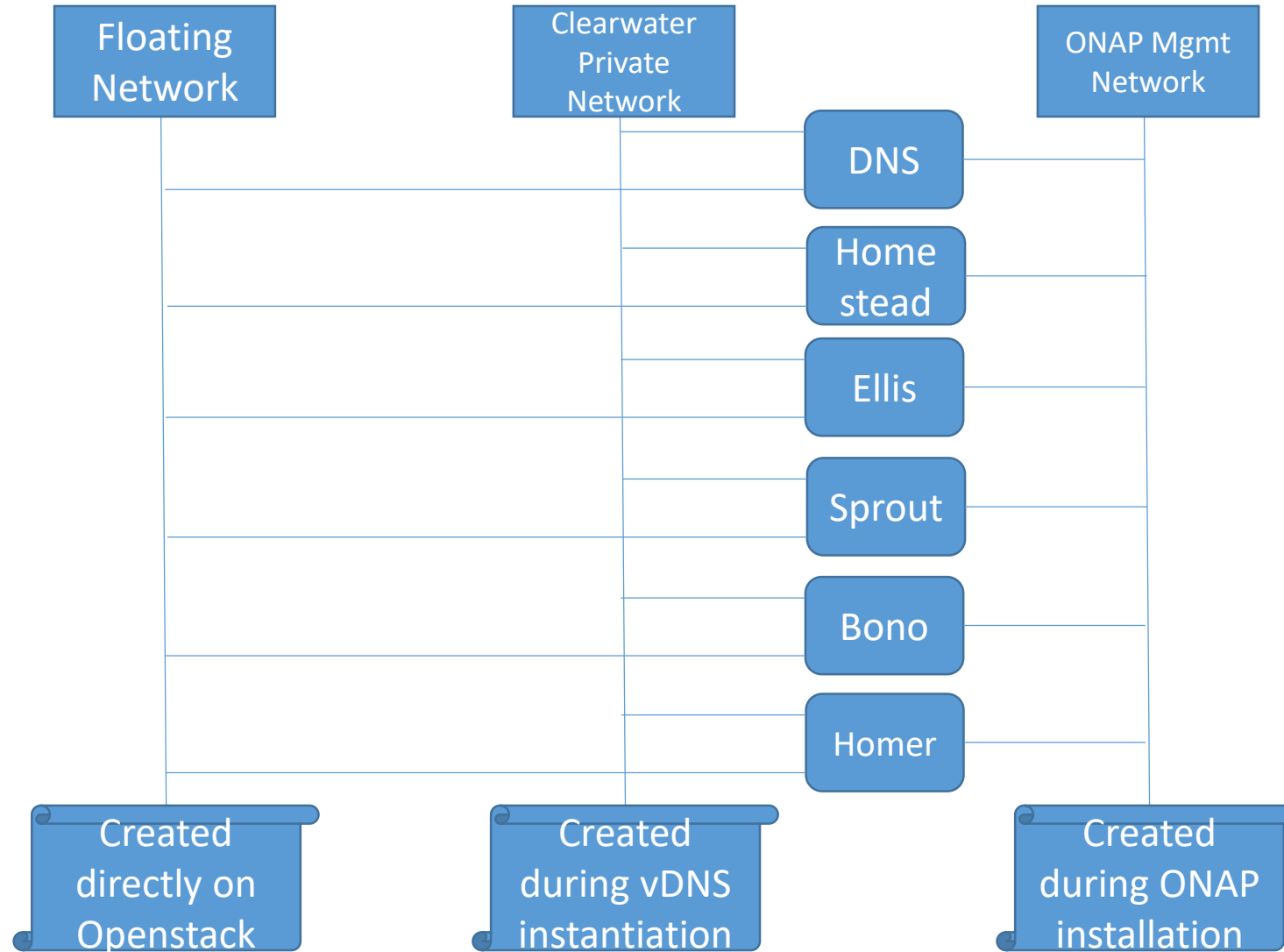
ONAP
OPEN NETWORK AUTOMATION PLATFORM

Use Case for Evaluation

❖ HEAT artifacts creation for individual Clearwater vIMS components like Sprout, Bono etc.

❖ Design and Instantiate vIMS service with all the individual components / VNFs

❖ Design day 0 and day 1 configurations using APP-C CDT (Ansible plug-in)

❖ Apply Day 0 configuration during instantiation

❖ Apply Day 0 and Day 1 configuration during a scale-out scenario

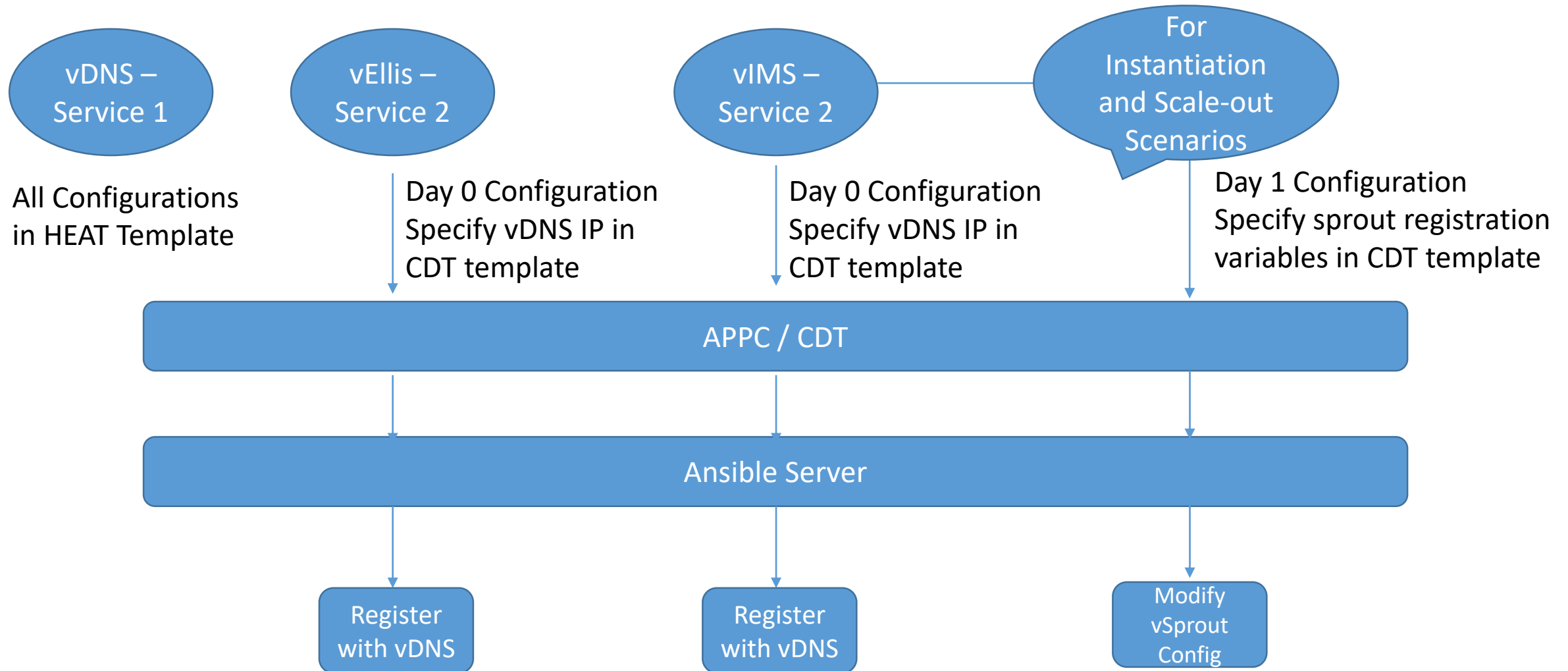ONAP
OPEN NETWORK AUTOMATION PLATFORM

vIMS Use Case - VSPs

Clearwater vIMS monolithic HEAT templates

Decomposed

Sprout VSP

Bono VSP

DNS VSP

Ellis VSP

Homer VSP

Homstead VSP

Ralf VSP

vIMS Network Layout

vIMS Service Deployment

vDNS – Service 1

vEllis – Service 2

vIMS – Service 2

For Instantiation and Scale-out Scenarios

All Configurations in HEAT Template

Day 0 Configuration Specify vDNS IP in CDT template

Day 0 Configuration Specify vDNS IP in CDT template

Day 1 Configuration Specify sprout registration variables in CDT template

APPC / CDT

Ansible Server

Register with vDNS

Register with vDNS

Modify vSprout Config

ONAP
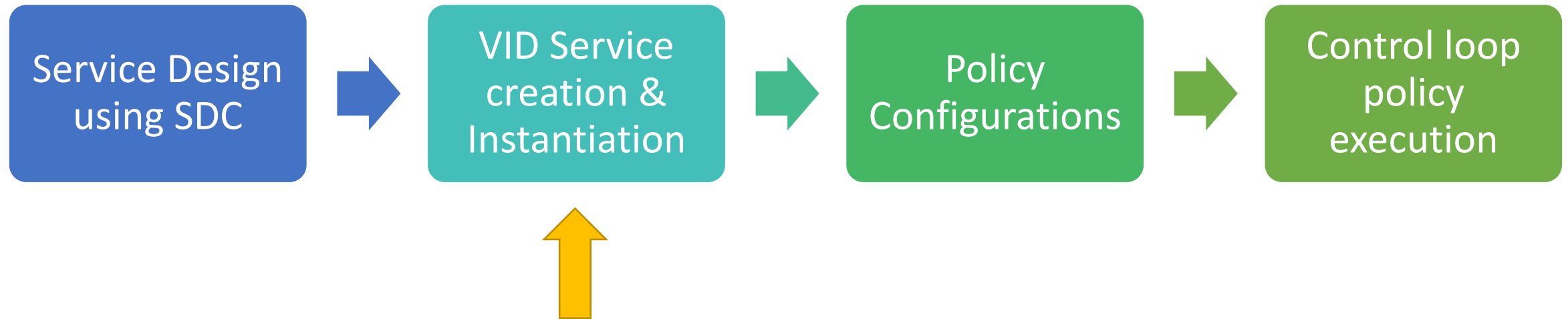OPEN NETWORK AUTOMATION PLATFORM

Contents

- Automation Objectives

- Onboarding, Instantiation, Configuration – Recap of the current process

- Enhancements recommended for Dublin and future releases
  - Support of SDC resources during instantiation
  - Single click instantiation
  - Heat bridge automation
  - Elimination of manual SDN-C Pre-load
  - Day 0 Configuration during instantiation
  - Automation requirements for scale-out

- Scale-in Readiness

- Managing ONAP from Operations perspective

- Q/A – Open Discussions
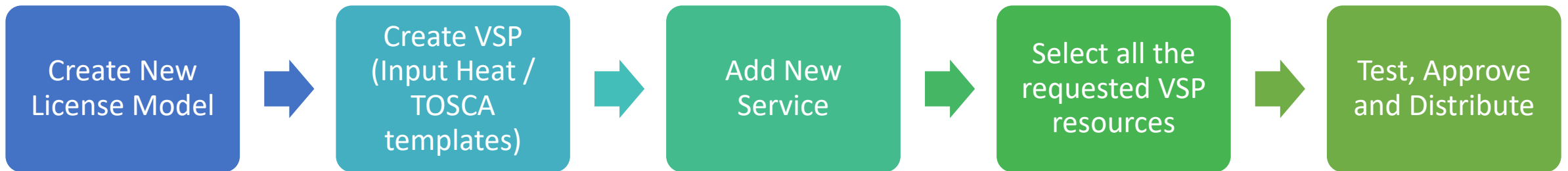
Automation Objectives

- Elimination of all the manual steps during instantiation

- Support generic functionality not tied to any specific use cases

- Enable day 0 configurations during instantiation

- Support SDC resources (Ex. Virtual Links, Connection Points) during instantiation

- Single Click Instantiation using the service TOSCA template

- Single pane of glass for managing ONAP

Onboarding, Instantiation, Configuration – Recap of the current process
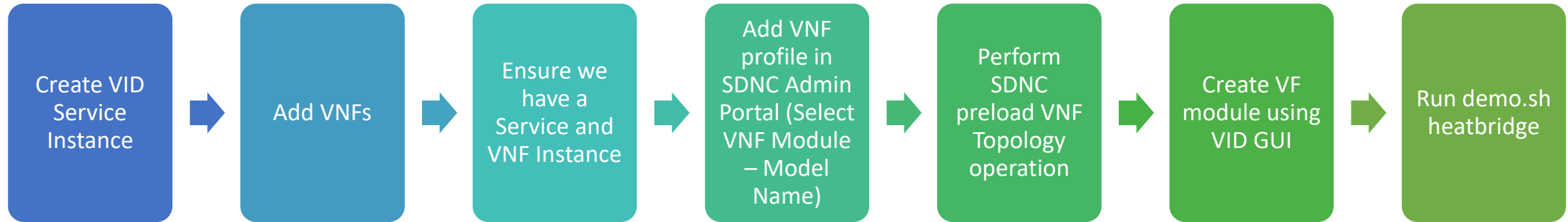
Service Design using SDC

VID Service creation & Instantiation

Policy Configurations

Control loop policy execution

Scope to automate / simplify steps during the VID Service creation and Instantiation

ONAP
OPEN NETWORK AUTOMATION PLATFORM

Service Design Steps – Steps

Create New License Model → Create VSP (Input Heat / TOSCA templates) → Add New Service → Select all the requested VSP resources → Test, Approve and Distribute

VID Service Creation (Base Module – Steps)

Create VID Service Instance → Add VNFs → Ensure we have a Service and VNF Instance → Add VNF profile in SDNC Admin Portal (Select VNF Module – Model Name) → Perform SDNC preload VNF Topology operation → Create VF module using VID GUI → Run demo.sh heatbridge
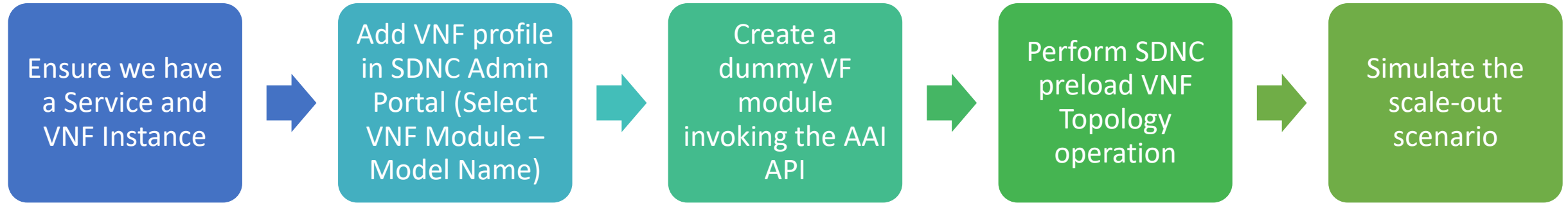
- All the steps in this phase need to be collapsed and provided with a single operation / click facility to instantiate a service

- Components Impacted – VID, SO

THE **LINUX** FOUNDATION

ONAP
OPEN NETWORK AUTOMATION PLATFORM

VID Service Creation (Scaling Module) – Steps

Ensure we have a Service and VNF Instance	Add VNF profile in SDNC Admin Portal (Select VNF Module – Model Name)	Create a dummy VF module invoking the AAI API	Perform SDNC preload VNF Topology operation	Simulate the scale-out scenario

- All the steps in this phase also need to be automated such that during base service instantiation, the required pre-requisites for "to-be" scaled out VNFs are in place. Ex. Homing policy, selection of cloud region, the existence of pre-load data, VNF naming policy etc.

- Components Impacted – SO

Enhancements recommended for Dublin Release

Support TOSCA Network Constructs

Limitations

- Currently, only the HEAT artifacts are supported for instantiations by SO.

Desired Features

- The capability to parse the TOSCA constructs for Connection Points and Virtual Links – and to create related entities in the openstack also needs to be supported.

Dublin Tracking

- Components Impacted – SO
- Jira tickets - SO-403 (Use End to End TOSCA workflows to support HPA)

ONAP
OPEN NETWORK AUTOMATION PLATFORM

JIRA SO-403

- **Title**: Use End to End TOSCA workflows to support HPA

- **Ticket Description**: Stretch Goal for R2. Create new workflows and use End to End TOSCA to support HPA where decomposition happens in SO. More detail TBD, but likely needs to have SO-22, as it needs to use TOSCA.

- **Additional Requirements**: Presume that this Jira item shall bring in the functionality to support orchestration of all the major TOSCA constructs such as VDU, Virtual Links, Connection Points etc. In effect SO should be able to orchestrate the current equivalent of vDNS, vFirewall HEAT templates available - but through the TOSCA artifacts - and be able to create multiple networks (Private, floating, management etc.) and attach the same to VM.

Single Click Instantiation

Limitations

- After Service is distributed there are multiple manual steps to instantiate the service.
- Individual VNFs and VF modules have to be created for all the Virtual Functions included in a service.
- If a service has many VFs – then this process becomes cumbersome and time consuming!

Desired Features

- Single click feature where all the minimal details are captured in a single screen and all the VNFs within a service are instantiated by a single GUI click / operation.
- Including the Heatbridge in the SO instantiation workflow

Dublin Tracking

- Components Impacted – VID, SO
- Jira tickets - SO-588, VID-380

Single Click Instantiation – JIRA Tickets

JIRA Ticket No	Title	Description / Comments
SO-588	Automate robot heatbridge manual step to add VF Module stack resources in AAI	After the instantiation there should be no need to
VID-380	Single click instantiation of a network service	VID to provide a GUI / API to that will eliminate the need to create every VNF (VF) and VF Module included in the service. Presumption: VID shall be calling the SO API – createE2EServiceInstance

THE **LINUX** FOUNDATION

ONAP
OPEN NETWORK AUTOMATION PLATFORM

Elimination of Manual SDN-C Preload

Limitations

- Before VNF module instantiation a SDN-C preload operation needs to be performed.
- For the scale-out VNF module instances too, this preload operation is required.
- Manual configurations in between the instantiation cycles (initial and subsequent) defeats the purpose of automating the operations.

Desired Features

- The preload should be implicit – Most of the inputs should be gathered from the service design artifacts and during the service instantiation.
- For all items where manual inputs are not provided, a related policy should be existing and a check should be performed before initiating the instantiation. Ex. VNF / VF Module naming policy, Placement policies, whether Scale-out is supported etc.

Dublin Tracking

- Components Impacted – SO
- Jira tickets – SO-1364, SO-1365

Elimination of Manual SDN-C Preload and related items – JIRA Tickets

JIRA Ticket No	Title	Description / Comments
SO-1364	With the new API the instantiation fails if SDN-C preload is not done	With the new API, the instantiation fails if SDN-C preload is not done. However the same scenario works for the old API. The issue needs to be fixed for the new API as well.
SO-1365	Eliminate the manual step of creating dummy VF module for scale-out scenario	To eliminate the manual step of creating a dummy non-base module during the scale-out scenario, some pre-configuration needs to be done during the instantiation of base vf module itself - such that scale-out can happen multiple number of times without any manual steps involved.

Enable Day 0 Configurations during Instantiation

Limitations

- The SO does not call the Configure API on APP-C currently
- Thus no configurations are applied after the service instantiation
- The APP-C configuration templates cannot be defined at service level. Hence if any service specific configurations have to executed, it would have to be bundled with some VNF configuration.

Desired Features

- The SO workflow during instantiation to be modified to call the APP-C Configure API
- Ability to pass the configuration parameters through API during instantiation workflow
- All the network interface details (Name, IP Address) of the instantiated VNFs to be populated in AAI
- Support for service level configurations

Dublin Tracking

- Components Impacted – SO, APP-C
- Jira tickets – APPC-1228, SO-1366, SO-1367

Enable Day 0 Configurations during Instantiation – JIRA Tickets

JIRA Ticket No	Title	Description / Comments
APPC-1228	Source as A&AI functionality not working	The CDT templates with parameters values as references to AAI fields should be fetched from AAI and substituted in the parameter values.
SO-1366	SO Workflow need to call configure API during instantiation	Ability to call configure API along with the configuration parameters passed in the API. The parameters could be absolute values, reference to AAI or reference to SDNC configuration.
APPC-1309, SO-1367	Support service level creation of CDT templates and corresponding APIs	The APP-C configuration templates cannot be defined at service level. Hence if any service specific configurations have to executed, it would have to be bundled with some VNF configuration. CDT should allow creation of service level configuration templates and APP-C should provide new APIs to apply the service configuration.

ONAP
OPEN NETWORK AUTOMATION PLATFORM

Automation Requirements for Scale-Out

Limitations

- Successive scale-outs cannot happen seamlessly – i.e. without any manual configuration
- The scale-out configurations to be passed are not generic and tied to ONAP vDNS use case
- Need to manually create a Dummy non-base VF module

Desired Features

- The mandatory VNF health check LCM to be removed and made generic such that it could enabled / disabled based on the service getting instantiated
- Ability to pass the scale-out configuration parameters supporting reference to SDN-C preload data, absolute values and reference to AAI fields
- Capability in SO to process the configuration parameters, fetch the details from SDN-C or AAI and then accordingly invoke the APP-C API
- Preparation for next scale-out after successful instantiation of the current scale-out operation

Dublin Tracking

- Components Impacted – SO, Policy
- Jira tickets – SO-1368, SO-1369, SO-1370, POLICY-1391

ONAP
OPEN NETWORK AUTOMATION PLATFORM

Automation Requirements for Scale-Out – JIRA Tickets

JIRA Ticket No	Title	Description / Comments
SO-1368	VNF Health check during scale-out to be made as a separate workflow	Currently during the scale-out, the VNF health check is done by default. Not every VNFs may support health check and neither there may be any requirement for mandatory health check before the scale-out. Hence this needs to be moved as a separate work-flow and as required through the configuration this may be included based on the VNF support / requirements.
SO-1369	Processing of configuration parameters during instantiation and scale-out	During the instantiation and scale-out operations, the SO should support processing of the configuration parameters with the below four variants - A. Reference to SDN-C preload data - Substitute the values from SDN-C preload B. Absolute values C. Reference to AAI fields D. No configuration data - Don't fail the request Currently when no configuration data is provided the SO returns an error.
SO-1370, POLICY-1391	Preparation for next scale-out after successful instantiation of the current scale-out operation	If max instances have not reached, refer to the naming policy to determine the name of the next scale-out instance Create the entry in AAI with the above name (Or revert to default) and with fields is-base-module as false and orchestration status as inactive

ONAP
OPEN NETWORK AUTOMATION PLATFORM

Scale-in Readiness

- Auto Scale-in left out from Dublin Release Goals

- Important architectural decisions to be taken –

  - VNF LCM API support to initiate the scale-in operation

  - Mechanisms for VNF to notify that VNF is ready to be scaled-in (After all the active traffic is handled or migrated)

  - The manual or policy based scale-in decisions based on telemetry data and DCAE analytics

- Affected Components – VID, SO, APPC, Policy

Managing ONAP from Operations perspective

- Goal - Single pane of glass for managing ONAP (Only Run Time)

- Dynamic map of the current ONAP services and VNF LCM Simplified GUI

- Ability to query and traverse the AAI Inventory

- Feature to select a VNF and invoke the VNF LCM APIs as required

- Components – UUI, VID