



ONAP Third Party Operational Domain Manager

Telstra

Apr, 2019

Usecase#1: ONAP as “Third Party” Operational Domain Manager

Use Case Overview

- A standards-based approach that allows a service provider to have a network automation platform for composite or white labelled services managed by specific/Third Party “domain managers
- ONAP provides Operations Domain Management (ODM) and other complementary capabilities to ensure full automation of the E2E lifecycle management of the service via federation
- Services are exposed and consumed via Network as a Service (NaaS) which is an abstraction layer above the operational domains and exposes the services to BSS
- Consistent way of consuming 3rd party services for service providers like Telstra
- ONAP will facilitate service operations value chain across C2M, P2O, O2A, T2R for third party domain via federation
- Substitutes multiple handovers between parties/teams and applications to enable zero touch automation

Interaction Flow

- 3rd party catalog import onto SDC -> publish catalog -> submit an O2A order -> fulfil the order

Changes foreseen

- Service Provisioning
 - SDC – Import 3rd party catalog, Publish catalog
 - SO work for Order Activation
 - A&AI – add info on 3rd party domain inventory
 - Enhancements to Ext APIs
- The service lifecycle management action can trigger for:
 - CM: change management (capacity increase to meet scaling demands);
 - IM: incident management (problem identification and fix); and
 - VM: Any support (workflow/notification if any) for services associated with VNFs managed by “Third Party”
- Remediation actions are triggered automatically using ONAP as ODM
- Necessary references to service provided by “Third Party” domains will be made available to support C2M/P2O processes

Network as a Service (NaaS)

An abstraction layer above the operational domains which acts as a gateway; exposes the services to BSS

TMF Open APIs

ODM (ONAP)

Operational Domain Manager to orchestrate Third Party Services

TMF Open APIs

Domain Specific APIs

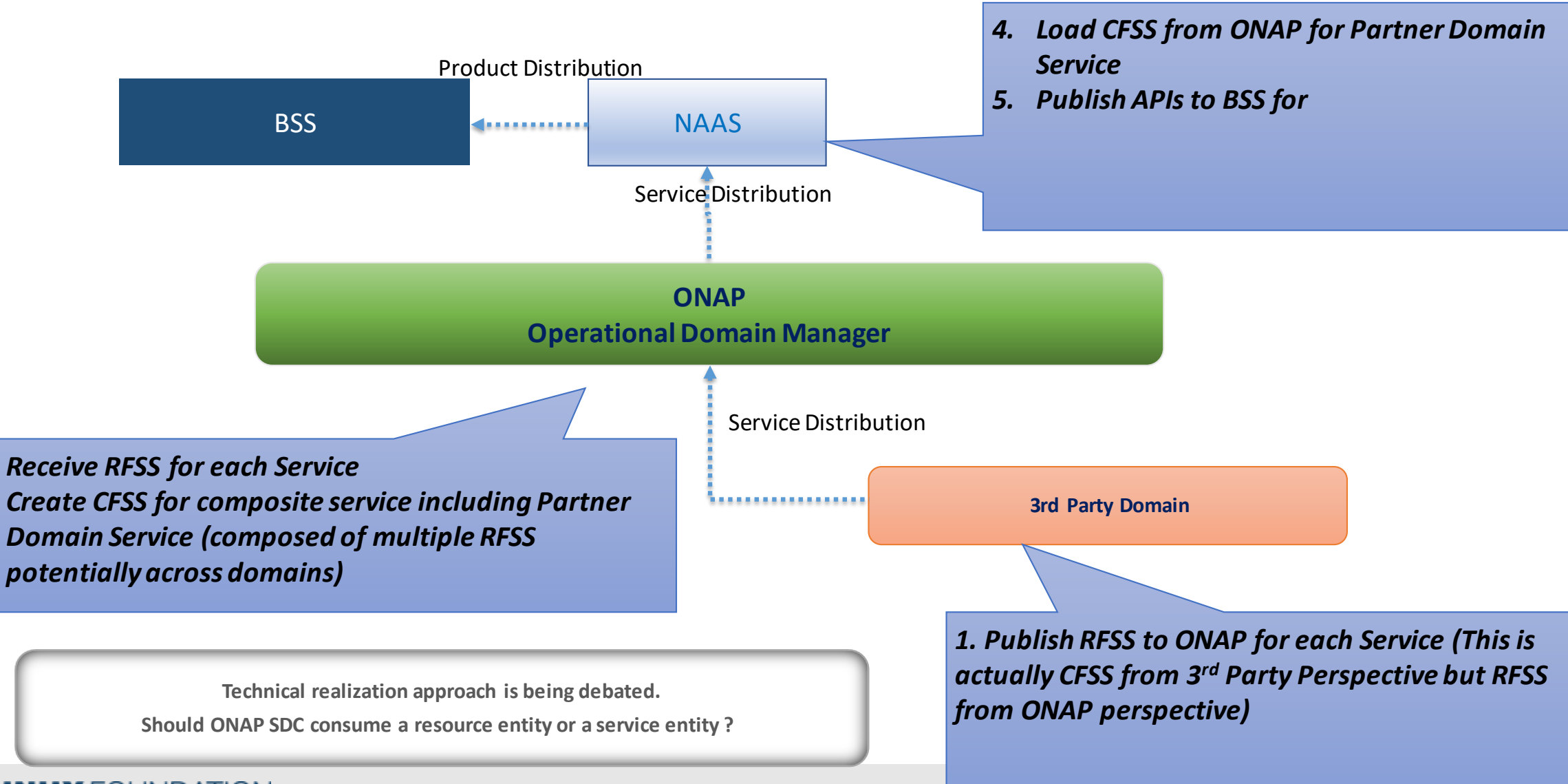
3rd Party Domain

Partner Domain – which can be another SP or another ONAP instance

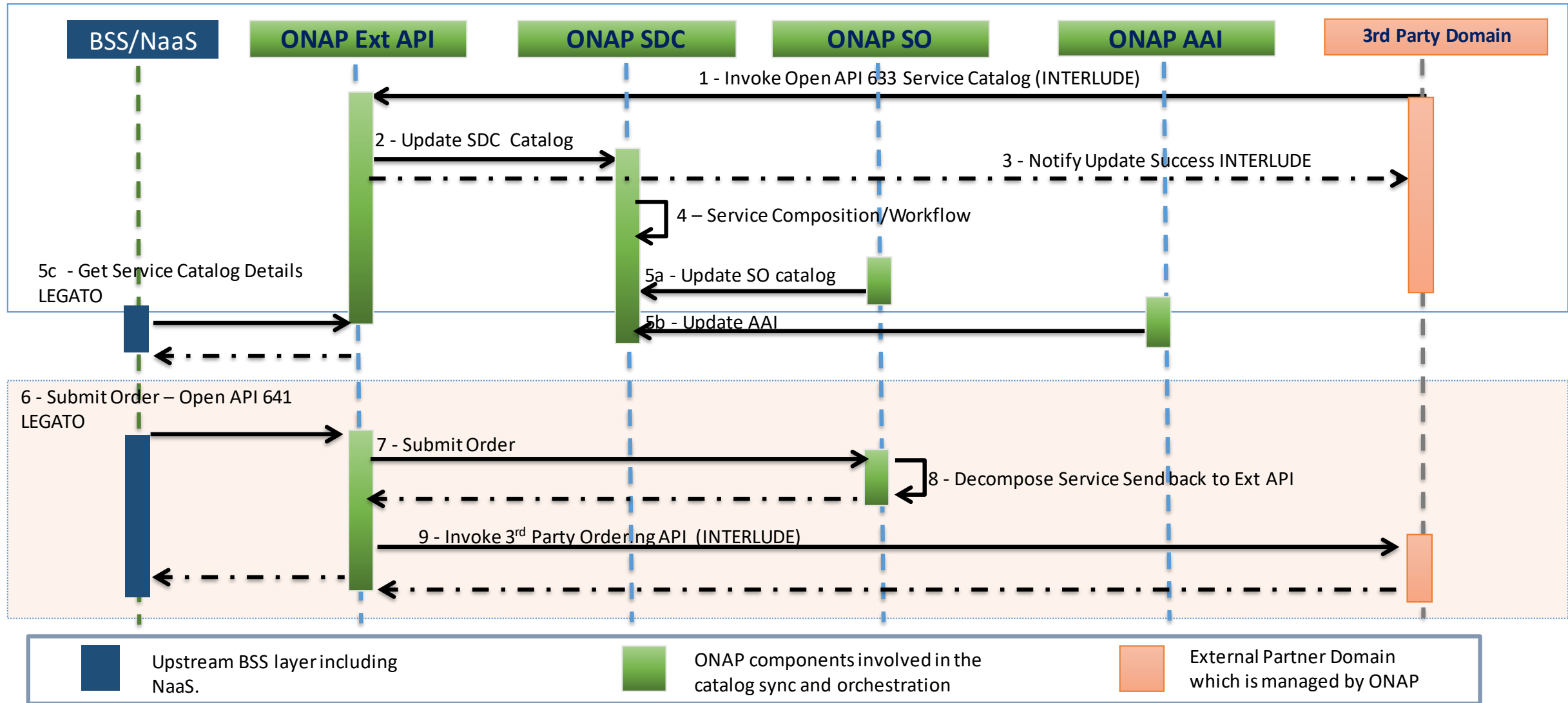
Changes mentioned in Grey are proposed for subsequent ONAP Releases beyond Frankfurt

Beneficiary: 3rd party providers and Service Providers like Telstra

Product Modelling & Distribution



Flow Diagram for 3rd Party Catalog Sync and Order Activation



The flow steps

Catalog Sync Summary

1 – External Third party domain exports its service catalog details to Telstra. Telstra orchestrator ONAP exposes TMF Open API 633 Service Catalog API via ONAP Ext API component. Third Party Domain leverages the API 633 to POST the Service Catalog payload.

POST nbi/api/v3/serviceSpecification

Request body –ONAP compatible Service CSAR / (json ??)

CSAR contents:

RFSS for Partner Domain Service

2 – ONAP Ext API updates SDC catalog by invoking internal SDC API

POST sdc/v2/catalog/services

3 – Ext API notifies Third party after successful update within ONAP

4 – Service Decomposition happens in SDC (any manual updates e.g. creating composite service)

SDC updates other ONAP components (which have registered with SDC DMaaP) with catalog details

5a – SO pulls SDC catalog details

5b – AAI pulls inventory details

Ext API also notifies northbound systems (BSS/NaaS) after successful import of the service catalog into ONAP.

5c - BSS retrieves catalog information from ONAP

Order Activation Summary

6 – BSS submit order using TMF 641 Service Ordering API, that is exposed by ONAP Ext API

7 – ONAP Ext API submits the request to ONAP SO

8 – ONAP SO decomposes the service and submits the request by invoking Ext API (This is similar to what is being proposed for CCVPN use case as well.

This maintains that only Ext API interacts with outside world and other ONAP components do not)

9 – ONAP Ext API invokes the Third Party Ordering API

Impact Analysis So Far for 3rd Party Catalog Sync

SDC

- Introduce POST function to SDC on-boarding functionality
- Reuse sdc-dao to update the Cassandra database and store the new service in SDC catalog
- Reuse SDC distribution functionality to distribute the new service to registered ONAP components (no change)
- Feedback from Ofir - Create ONAP compatible CSAR input to be consumed by the POST. The UUID creation functionality of SDC needs to be maintained, so that UUID is created on the fly in the same manner as today
- Last mile access service from 3rd party will be used for detailed analysis and reference implementation
- TOSCA based onboarding in work in progress in SDC, it supports heat based only. The TOSCA based work is ongoing separately in Modeling project. This dependency on Modeling project need to be looked into.

Ext API / NBI

- Introduce POST for TMF API 633 – Service Catalog API
- Realization of POST operation in Ext API will depend on decisions taken during SDC implementation.
- **Ext API changes to be planned for future release**

Possible Approaches for 3rd Party Catalog Sync

Entity Option 1: Resource

- Onboard the resource in ONAP SDC as a VSP, will require updates to VSP onboarding API

Entity Option 2: Service

- Onboard the service in ONAP SDC as a Service, will require updates to Service onboarding API

Legend : Pros , Cons, Neutral

Payload Option 1: JSON

- Leveraging existing approach for Ext API / NBI
- ExtAPI / NBI will need to convert JSON to YAML for SDC Consumption

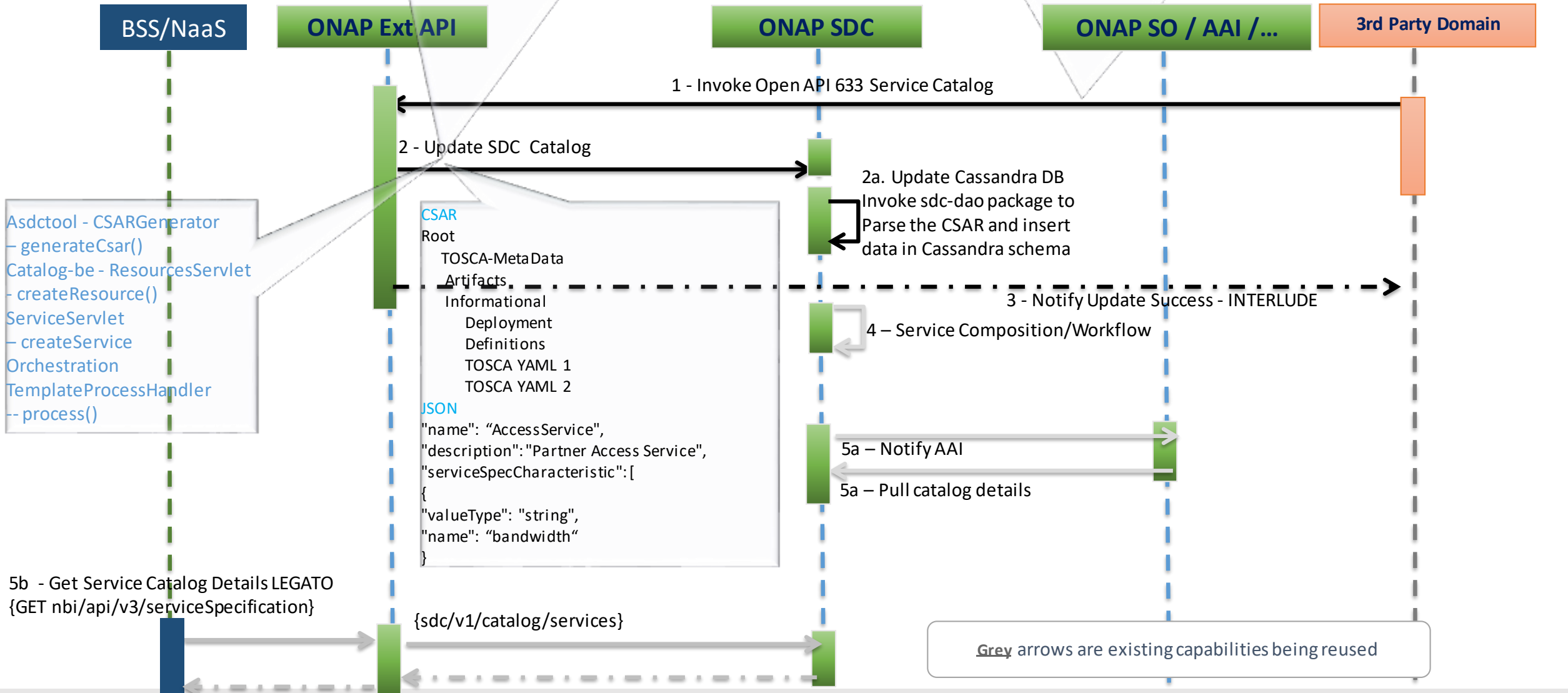
Payload Option 2: CSAR

- Potential reuse from TOSCA onboarding Project in SDC
- This might alter existing Ext API / NBI approach
- There would be additional implementation at Third Party end to generate higher level TOSCA

Flow Diagram for 3rd PARTY SDC Catalog Sync

Invoke onboarding API {POST sdc/v1/catalog/services} CSAR/JSON contents:
RFSS for partner service to be created as a Service in ONAP

Publish Resource or RFSS to ONAP using TOSCA Service CSAR or JSON
(INTERLUDE) {POST nbi/api/v3/serviceSpecification}



Asdctool - CSARGenerator
- generateCsar()
Catalog-be - ResourcesServlet
- createResource()
ServiceServlet
- createServiceOrchestration
TemplateProcessHandler
-- process()

```

CSAR
Root
  TOSCA-MetaData
  Artifacts
  Informational
  Deployment
  Definitions
  TOSCA YAML 1
  TOSCA YAML 2
JSON
{"name": "AccessService",
 "description": "Partner Access Service",
 "serviceSpecCharacteristic": [
 {
 "valueType": "string",
 "name": "bandwidth"
 }
 ]
 }
    
```

Grey arrows are existing capabilities being reused

Activities in Scope for SDC

- Impact Analysis
- Sprint Planning
- Design
- Implementation
- Architecture Documentation
- Integration ??
- Testing Based on ROBOT Framework
- Defect Management



ONAP

OPEN NETWORK AUTOMATION PLATFORM

Thank you