



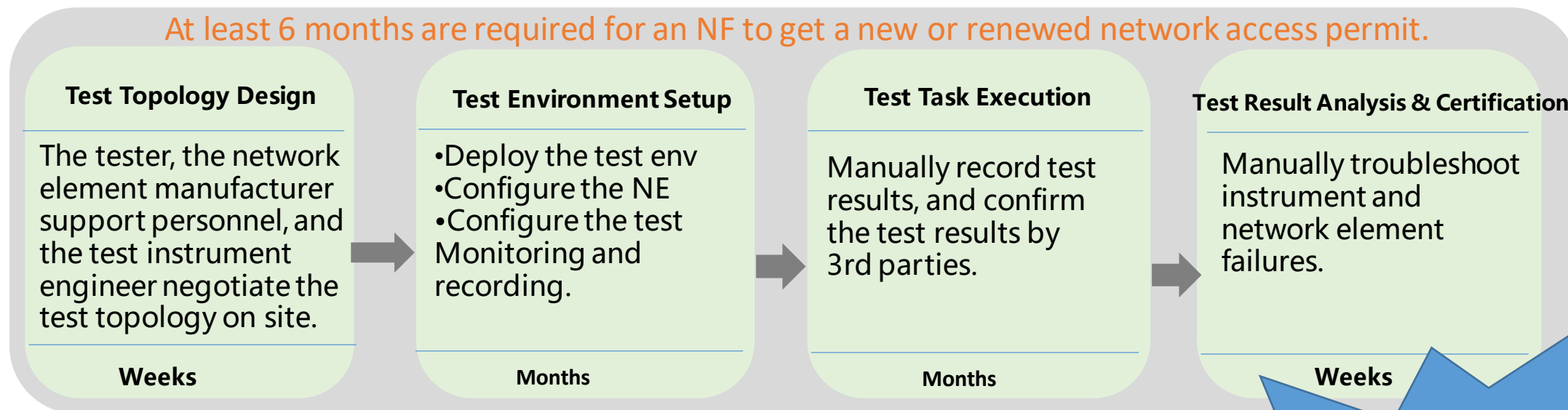
Support for Test Topology Auto Design – SDC Enhancement

ONAP G release

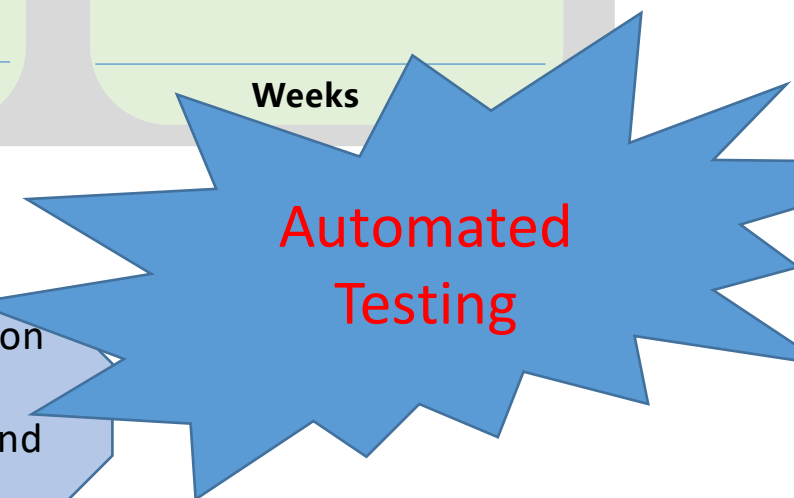
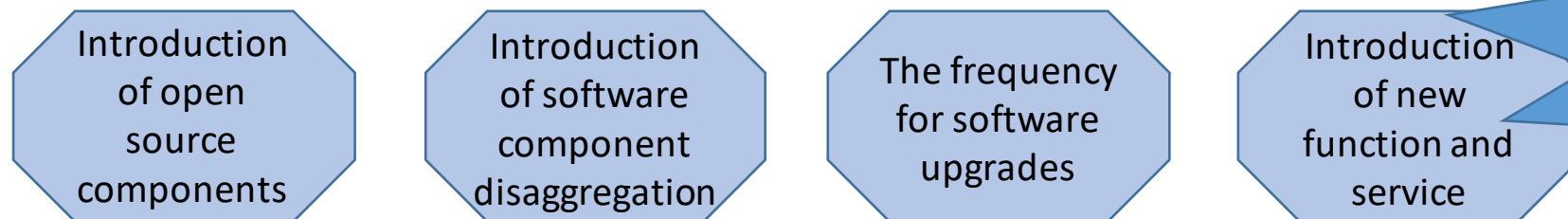
Yan Yang, Lei Huang, Keguang He (China Mobile)

NE Testing: Reality

- The NE testing process is usually divided into four steps: test topology design, test environment setup, task execution and result analysis and certification.



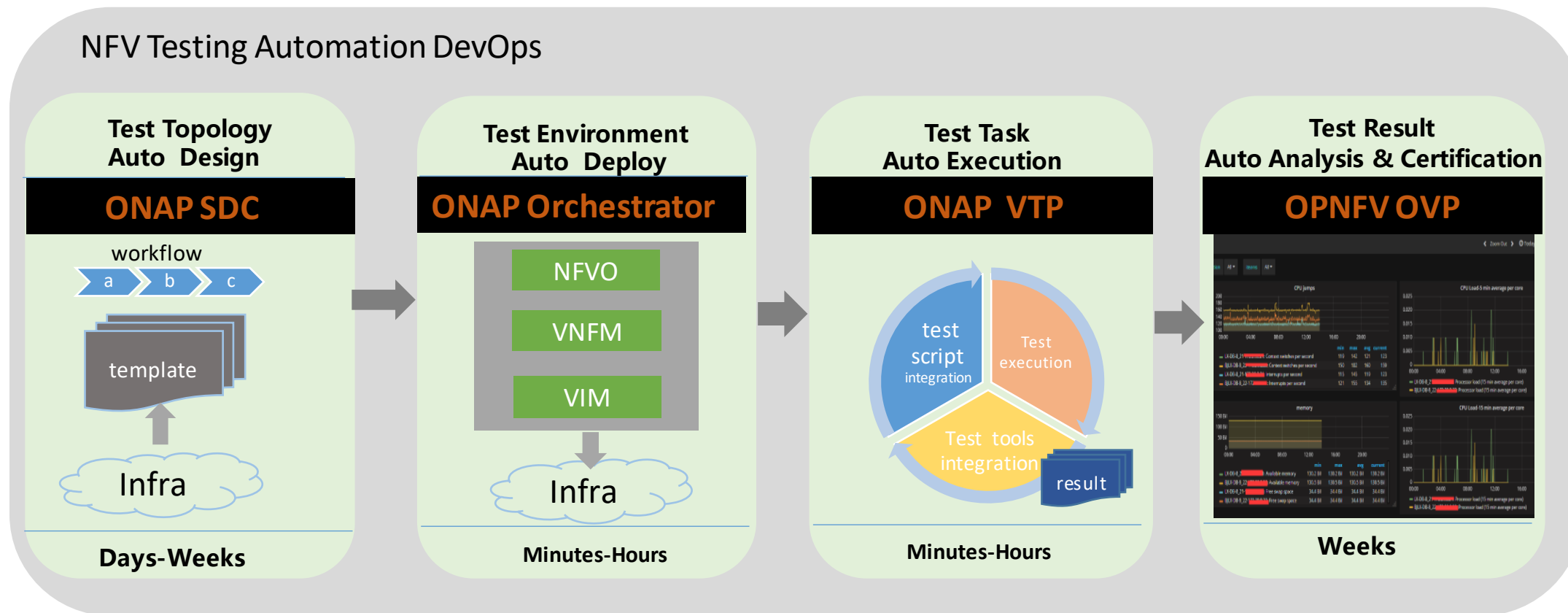
- Changes after the introduction of NFV



NFV Testing Automation with OVP+ONAP

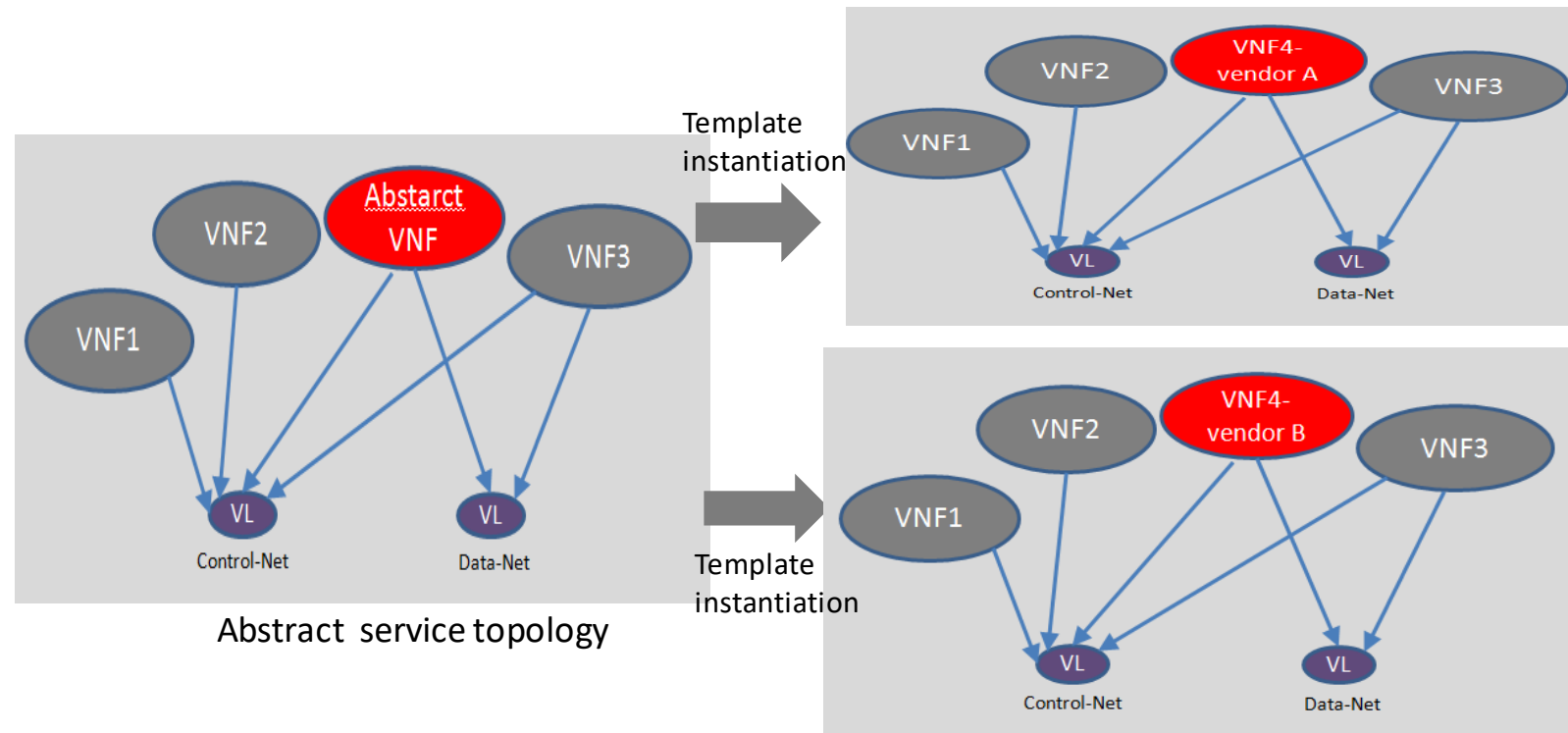
Function mapping with ONAP components

- **Test Topology Design - ONAP SDC**
- Test Environment Deploy - ONAP Orchestrator(SO, VF-C, APPC , A&AI, etc)
- Test Task Execution - ONAP VTP(VNFSDK+CLI)
- Test Result Certificate - OPNFV OVP

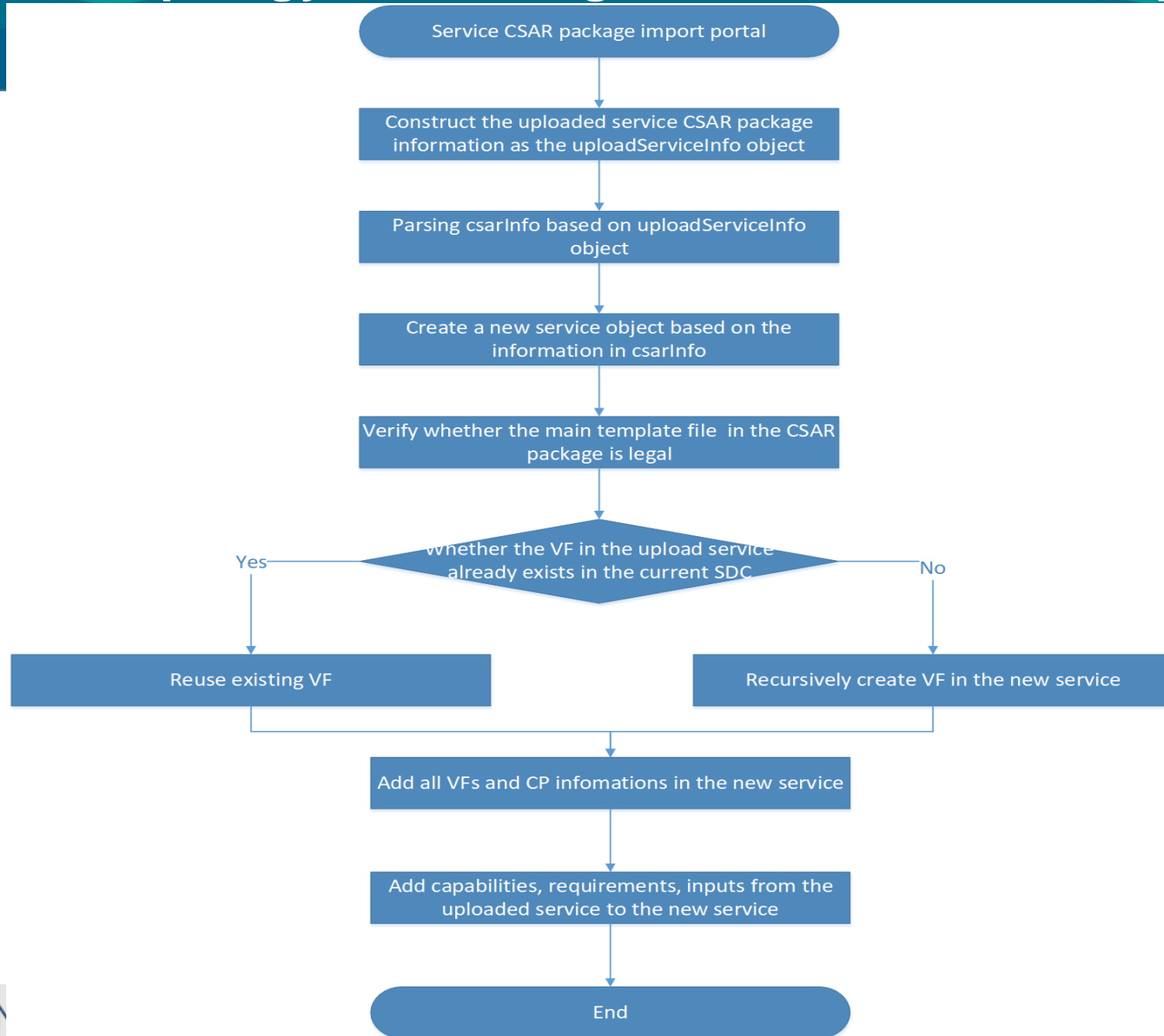


Support for Test Topology Auto Design

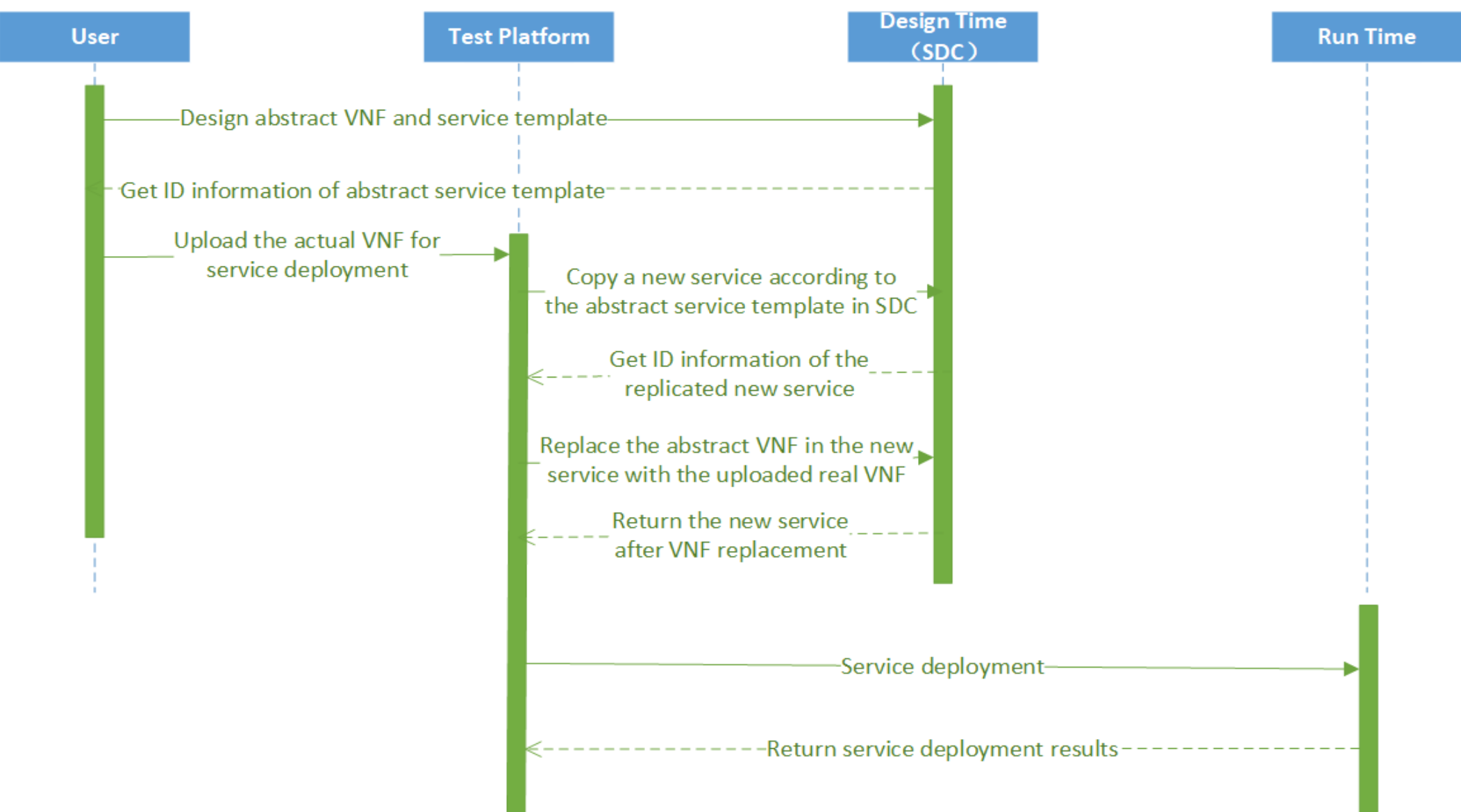
- **Goal:** To achieve a more flexible NFV automated test platform, ONAP-based NFV automatic testing platform provides a quick test service(topology) design composed of tested VNF and test environment, as well as supports the import and reuse test service(topology) between different test environments.
- **Current situation:** At present, the service design need to be repeated for each VNF / test vendor, resulting in duplication, complexity, and also reduce efficiency.
- **Possible solution:** Define abstract testing service (topology) template for each type of VNF
- **Enhance SDC to support:**
 1. For the service designed, can be imported into SDC for modification or enhancement, or the test template can be reused for different test environments (the SDC needs to support service import);
 2. Quickly design a test service (topology) composed with tested VNF and test environment (One way is to introduce the concept of abstract template to simplify the process of repeated design of test service(topology)).



Support for Test Topology Auto Design- Possible Workflow (Service import)

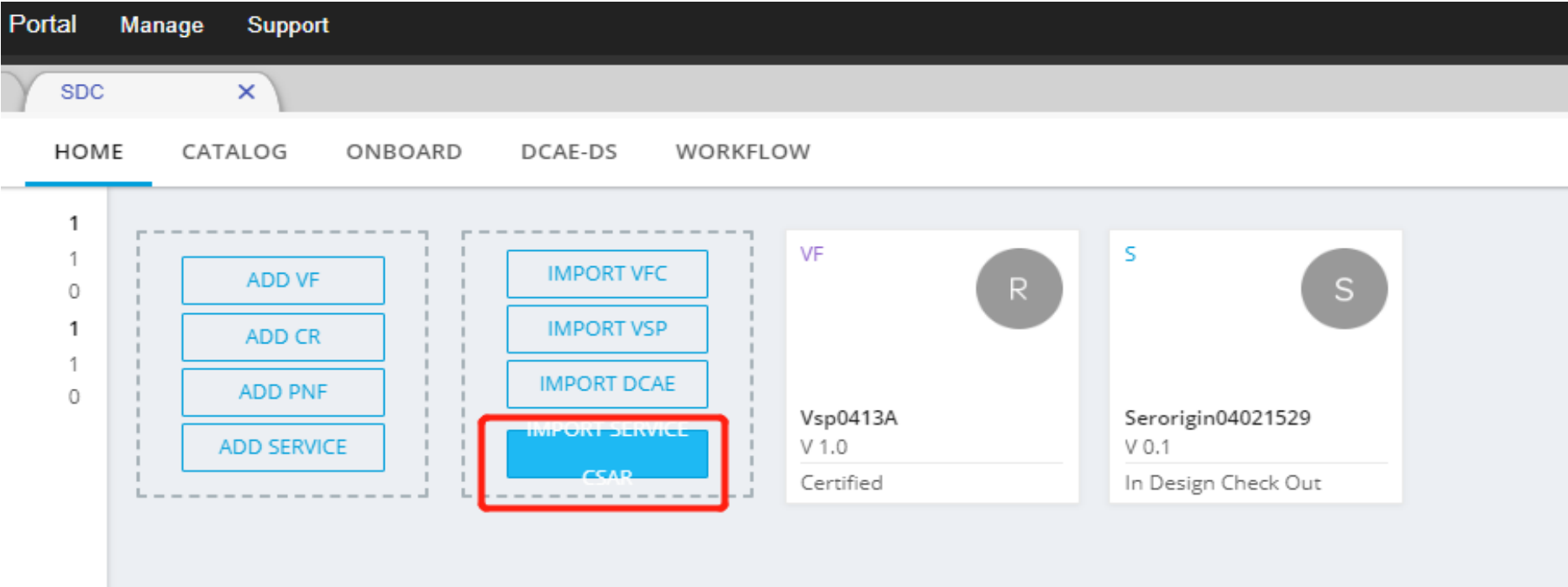


Support for Test Topology Auto Design- Possible Workflow (Abstract service template)



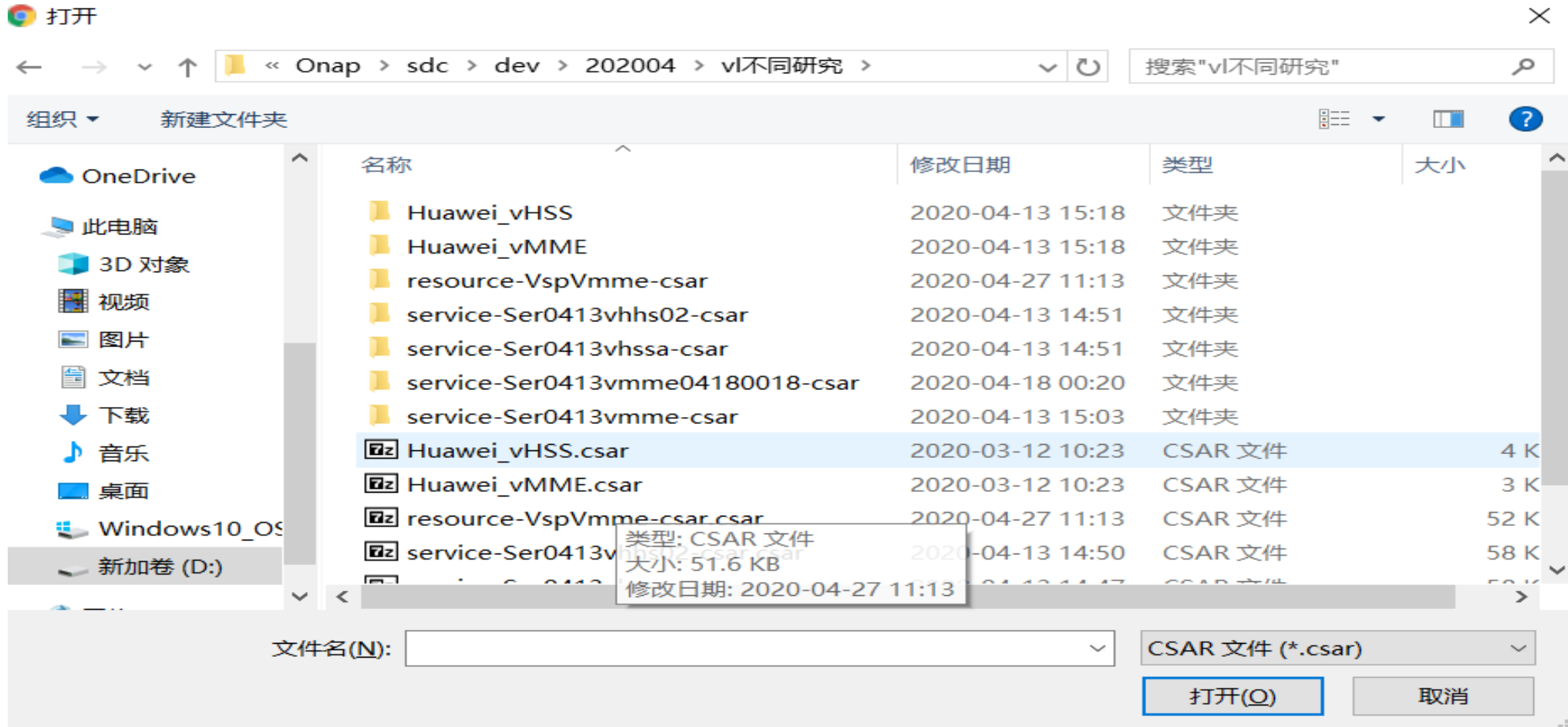
Service import - Changes in SDC Portal

- Add a button “IMPORT SERVICE CSAR” to perform service CSAR import.



Service import - Changes in SDC Portal

- When clicking the “IMPORT SERVICE CSAR” button on the portal, a window will pop up to select the service CSAR file to be imported.



Service import - Changes in SDC Portal

- After selecting the service CSAR file to be imported, it will switch to the general information input page for creating the service.

The screenshot shows the SDC Portal interface for creating a service. The browser tab is labeled 'SDC' and the search bar contains 'What are you looking for?'. The navigation menu includes 'HOME', 'Create new service', and 'General'. The main content area is titled 'IN DESIGN' and contains a 'Create' button. The 'General' section is divided into two columns. The left column includes a circular icon with the letter 'S', a text input field for '* Name', a dropdown menu for '* Category' with the text 'Select category', a text area for 'Tags', and a text area for '* Description'. The right column includes a 'Save' button, a text input field for '* VSP' with the value 'service-Ser0413vmme-csar.csar' and a 'Browse' button, a text input field for '* Contact ID:' with the value 'cs0008', a dropdown menu for 'Generated Naming' with the value 'true', a text input field for 'Naming Policy', a text input field for 'Service Type' with the value 'Service', a text input field for 'Service Role', a text input field for 'Service Function', a dropdown menu for 'Environment Context' with the value 'General_Revenue-Bearing', and a dropdown menu for 'Instantiation Type' with the value 'A-la-carte'.

Service import- Changes in SDC Portal

- After filling in all the required fields, you can click the "create" button to create a new service.

The screenshot shows the 'General' configuration page for a service named 'MMETest'. The page is titled 'SERVICE: MMETest > General' and has a status of 'IN DESIGN'. A 'Create' button is highlighted with a red box. The form contains the following fields:

- Name:** MMETest
- Category:** E2E Service
- Tags:** MMETest
- Description:** 2020.05.22
- VSP:** service-Ser0413vmme-csar.csar
- Contact ID:** cs0008
- Generated Naming:** true
- Naming Policy:** (empty)
- Service Type:** Service

Buttons for 'Save' and 'Create' are visible. The 'Create' button is highlighted with a red box.

Service import - Changes in SDC BE

- ServiceServlet.java

Add a new API for the request of importing service CSAR.

Post:

/v1/catalog/services/importService

Define the UploadServiceInfo class to receive the service CSAR package uploaded by the portal and the general information filled in from the portal.

Service import - Changes in SDC BE

- ServiceImportBusinessLogic.java

Including all processing logic codes for creating service.

- Analyze and verify the uploaded CSAR file.
- Parsing Tosca template file to get the corresponding contents of inputs, node_templates, instances, groups and so on respectively.
- Create a new service based on the above parts.

- Add support for service type in the following related code.

CsarArtifactsAndGroupsBusinessLogic.java YamlTemplateParsingHandler.java

ComponentBusinessLogic.java

ServiceBusinessLogic.java

CsarUtils.java ToscaOperationFacade.java

Service import - Result

- A new service is generated in SDC.

87 Active Elements found

The screenshot displays three service elements in a grid. Each element has a blue 'S' icon and a grey circle with a white 'S'. The third element is highlighted with a red border.

| Service ID | Version | Status |
|---------------------|---------|---------------------|
| Ser05211622 | V 0.1 | In Design Check Out |
| Ser05211059 | V 0.1 | In Design Check Out |
| Ser0413vmme04180018 | V 0.3 | In Design Check Out |

Abstract service template - Changes in SDC Portal

- On the general page of VF, add a IS_ABSTRACT_RESOURCE selection box, which is false by default. If it is an abstract VNF, select true manually.

ME > VF: vsp0413A > General >

V1.0 CERTIFIED Switch to the latest version Upgrade Services Check Out ×

General

Save

*** Name**
vsp0413A

*** Category**
Configuration

Tags
vsp0413A

*** Description**
ok

*** VSP**
vsp0413A (1.0) Browse

*** Contact ID:**
cs0008

*** Vendor**
lic0413

*** Vendor Release**
1.0

Vendor Model Number

IS_ABSTRACT_RESOURCE
false

Created:
04/13/2020, Carlos Santana

Abstract service template - Changes in SDC BE

- Add three APIs to handle the corresponding requests of abstract service template.
 - Return whether the service is a abstract service:
GET /v1/catalog/abstract/service/serviceUUID/{uuid}/status
 - Copy a new service based on the existing service:
POST /v1/catalog/abstract/service/copy
 - Replace the abstract VNF in the abstract service template with the actual VNF:
PUT /v1/catalog/abstract/service/replaceVNF

We will use automated interfaces to replace the original manual implementation process.

Summary- SDC changes in automatic testing requirement (Portal)

- The requirements and related changes are relatively independent from existing functions in SDC, they will not affect other functions in SDC, just enhancement and complement.
- **Changes in SDC Portal**
 - **Abstract service template**

On the general page of VF, add a IS_ABSTRACT_RESOURCE selection box, which is false by default. If it is an abstract VNF, select true manually.
 - **Service import**
 1. Add a button "IMPORT SERVICE CSAR" to perform service CSAR import.
 2. When clicking the "IMPORT SERVICE CSAR" button on the portal, a window will pop up to select the service CSAR file to be imported.
 3. After selecting the service CSAR file to be imported, it will switch to the general information input page for creating the service.
 4. After filling in all the required fields, you can click the "create" button to create a new service.

- **Changes in SDC BE**

- **Abstract service template**

Add three APIs to handle the corresponding requests of abstract service template.

(1) Return whether the service is a abstract service:

GET /v1/catalog/abstract/service/serviceUUID/{uuid}/status

(2) Copy a new service based on the existing service:

POST /v1/catalog/abstract/service/copy

(3) Replace the abstract VNF in the abstract service template with the actual VNF:

PUT /v1/catalog/abstract/service/replaceVNF

We will use automated interfaces to replace the original manual implementation process.

- **Service import**

1. Add a new API for the request of importing service CSAR.

Post: /v1/catalog/services/importService

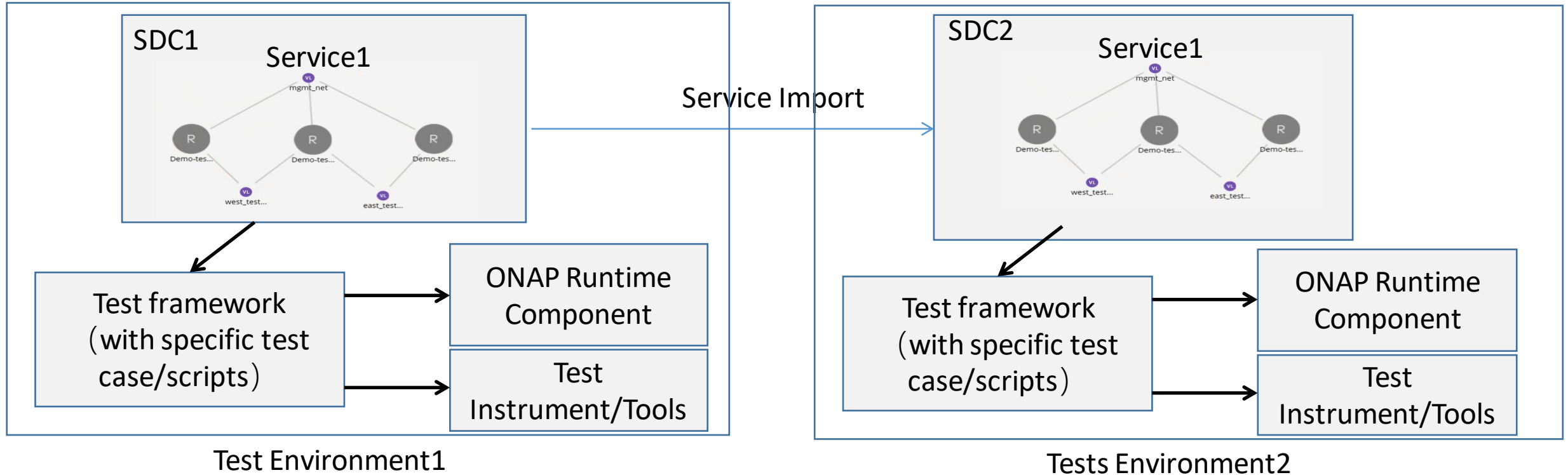
Define the UploadServiceInfo class to receive the service CSAR package uploaded by the portal and the general information filled in from the portal.

2. Including all processing logic codes for creating service.

3. Add support for service type in the following related code.

Conclusion of service import discussion- part1(use case)

- Support for service import: Service csar package exported from one SDC can be re-imported to another SDC



- Process:
 - Step 1: Create and design Service1 in SDC1
 - Step 2 : Export Service1 csar package created in SDC1
 - Step 3: Import Service1 csar package to SDC2

Conclusion of service import discussion- part2 (illustration)

- Premise: The data type of different SDC environments are keep in sync.
- Support for features:
 - VNF features;
 - Contents of inputs, node_templates, instances, groups , etc.
 - Substantiation mapping (tentative support)
- Verify if the service imported in different SDC environment are equal:
 - Consider comparing services manually (e.g. use Beyond Compare, etc.);
 - OR create automatic comparison process (in terms of SDC's suggestion).

Remaining issues- solution plan

- Solution for datatype challenge:
 - A: We've explained the usecase of service import from one SDC to another, the data type of different SDC environments are keep in sync, so the data type can be supported in SDC.
- Can abstract template definition reuse category field:
 - A: We think it's OK to reuse the Abstract category.

Remaining issues- solution plan

- Select which field to judge whether VF exists:
 - A: Now we use the field type, we also can combine the fields such as type and category to judge

```
Product111 0:  
  type: org.openecomp.resource.vf.Product111  
  metadata:  
    invariantUUID: b891c828-093c-4fa3-9d41-b03cf85fb7b0  
    UUID: 17ac3ba8-4974-4636-b896-3baf7b68172f  
    customizationUUID: f5a6a7d1-9656-43e1-a0d9-09c4dc4be7a7  
    version: '0.1'  
    name: Product111  
    description: ok  
    type: VF  
    category: Generic  
    subcategory: Abstract  
    resourceVendor: he  
    resourceVendorRelease: '1.0'  
    resourceVendorModelNumber: ''  
  properties:
```