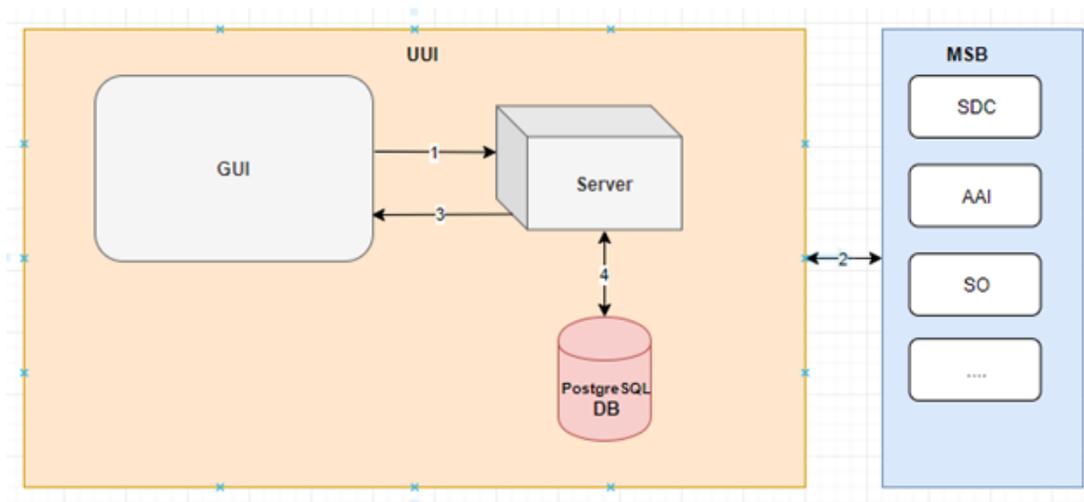


UII

- 1. Architecture
- 2. API Dependencies
- 3. Functional enhancements
- 4. Create MDONS instance
- 5. Get and Show MDON's Instance
- 6. Delete MDONS Instance

1. Architecture

1. The data which needs to be displayed like service details, service type details are requested to GUI (Angular project) to the UII server.
2. UII server is a dedicated server to UII front end which has controllers/services to call other components. On receiving the request appropriate service is called from UII server via MSB. MSB is where the other ONAP components like SO, AAI, etc. are registered with their API.
3. The response received from MSB is sent back to the UII and displayed accordingly to the User.
4. The information which are created in UII like Alarms, Service Instance details are stored in PostgreSQL DB which is specific to UII server.



2. API Dependencies

Below are the API/Dependency projects for UII.

No	API	Path	Comments	Impacts
1	Catalog API (SDC)	/api/sdc/V1/catalog/services	API to read the Network Service and VNF Catalogue and to fetch template details.	There is no change with the API. But will be adding MDON service in the list of service.
2	SO API	/api/so-serviceInstances/v3	API for Network Service instantiation and termination	There is no change with the API.
3	MSB API	Registered with MSB client	API for registration and use of micro-services bus	There is no change with the API.
4	A&AI API	/api/aa-externalSystem/V16/esr-nfvo-list	API for getting inventory and image management	There is no change with the API.

- **Catalog API:**

1. Request : Passing parameters E2E_SERVICE and DISTRIBUTED constants.

1. Response Sample: There is no change or impact in this API with MDOS use case.

```
[
  {
    "uuid": "5847f9cc-8425-4841-9abe-2e8aeba6dd39",
    "invariantUUID": "440484be-7b6e-4ca3-841a-a9d4661ec6fa",
    "name": "orange",
    "version": 2,
    "toscaModelURL": "<some>",
    "category": "second"
  },
  {
    "uuid": "5847f9cc-8425-4841-9abe-2e8aeba6dd39",
    "invariantUUID": "440484be-7b6e-4ca3-841a-a9d4661ec6fa",
    "name": "orange",
    "version": 2,
    "toscaModelURL": "<some>",
    "category": "second"
  },
  {
    "uuid": "5847f9cc-8425-4841-9abe-2e8aeba6dd39",
    "invariantUUID": "440484be-7b6e-4ca3-841a-a9d4661ec6fa",
    "name": "orange",
    "version": 2,
    "toscaModelURL": "<some>",
    "category": "second"
  }
]
```

- SO API:

1. Request Sample: This request sample is specific for MDON's use case.

```

{
  "service":{
    "name": "svc-Customer1-orange-001"
    "description": "svc-Customer1-orange-001",
    "serviceInvariantUuid": "440484be-7b6e-4ca3-841a-a9d4661ec6fa"
    "serviceUuid": "bf82273f-f7d1-4edb-b291-e2f5bb3a4674",
    "globalSubscriberId": "Customer1",
    "serviceType": "???",
    "parameters":{
      "locationConstraints":[
        ],
      "resources":[
        ],
      "requestInputs":{
        "name": "svc-Customer1-orange-001",
        "customer": "Customer1",
        "service_provider": "Orange",
        "due_date": "2020-09-06T18:00:00+09:00",
        "end_date": "2020-09-05T18:00:00+09:00",
        "uni_id": "5847f9cc-8425-4841-9abe-2e8aeba6dd39",
        "uni_client-proto": "Ethernet",
        "uni_coding-func": "10GBASE-R",
        "uni_optical-interface": "LR",
        "enni_id": "b55a1942-fcb7-4511-bb76-02bbe52c7c85"
      }
    }
  }
}

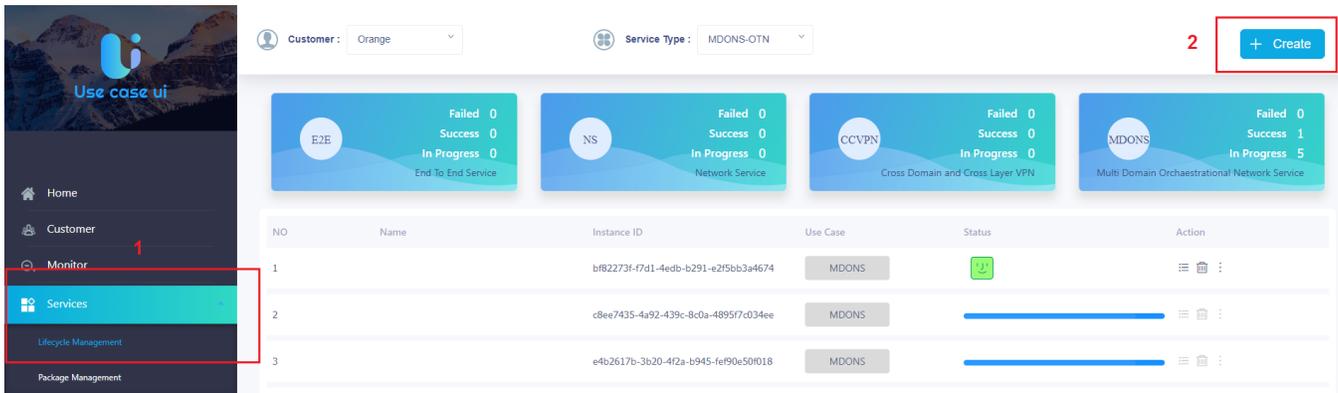
```

3. Functional enhancements

Below are the functionalities that need to be added in UI for MDONS use case.

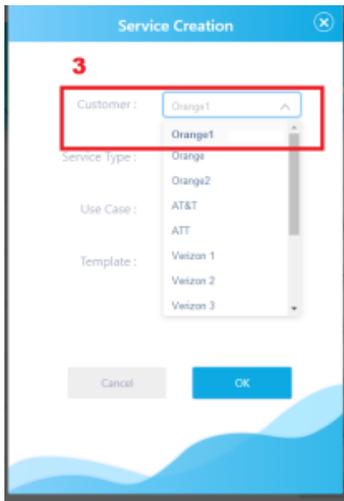
1. [Get and show MDONS instances](#)
2. [Create MDONS instances](#)
3. [Delete MDONS instances](#)

4. Create MDONS instance

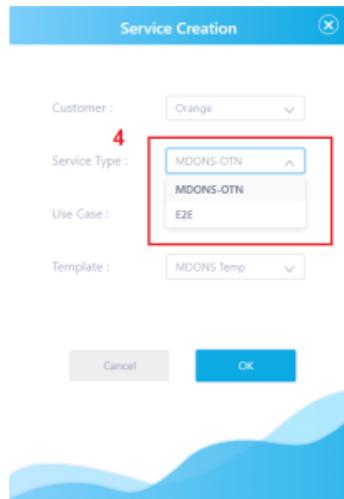


1. To create service go to Services tab and select Life cycle management. List of services created and their progress are listed.

2. Click on the create button on the top right corner of the screen. A pop up appears with the required fields for which service needs to be created.



3. Select the customer for the drop down list to which the service needs to be created,



4. Select MDONS as service type from the list displayed.

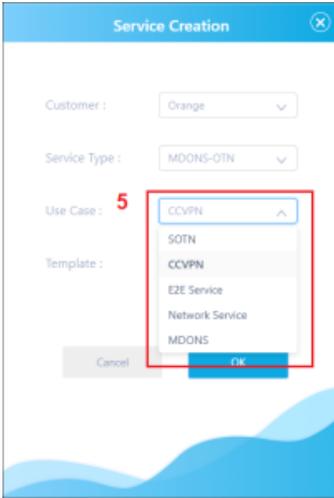
- Depending on the customer selected, service type is queried to the UUI server by the following path passing the customer id which is selected.

`/uui-lcm/customers/{customerId}/service-subscriptions`

- Then UUI server calls AAI to list the service subscriptions by the below path.

`/api/aai-business/v11/customers/customer/{global-customer-id}/service-subscriptions`

- The response which is a list of AAI Subscriptions is sent back to the UUI and populated in the Service Type drop down.



5. Use case type is the name or the type of the use case to which the service instance will be created. These service types are hardcoded values in the UUI. MDONS use case should be added to service types.

- On selecting the appropriate use case type, orchestrator list and available template list is fetched.
- For fetching the orchestrator list a call has been made to UUI server on below path

/uui-lcm/orchestrators

- Which in turn calls the AAI service to fetch back the list of AAI ESR Nfvo by calling the below path.

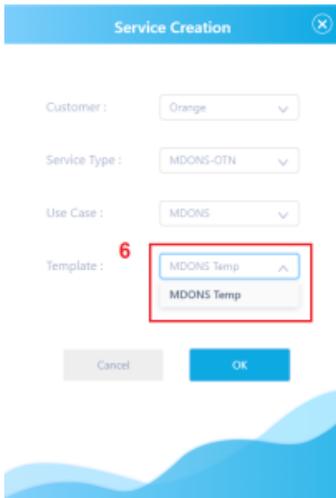
/api/aa-externalSystem/v16/esr-nfvo-list

- A call has been made to the UUI server on the below URL

/uui-lcm/service-templates

- Which in turn queries the SDC catalog service for fetching the list of SDCServiceTemplate objects.

/api/sdc/v1/catalog/services



6. Once the template is selected and click ok to decompose the selected template. Depending on the chosen template, template parameters are queried by passing the CSAR id & Tosca Model URL of the selected template from SDC. The CSAR file is downloaded the URL and the input fields are extracted which are the template parameters.

- On successful fetch of the template parameters, the parameter values are passed to their respective use case screen components.
- For the template parameters the input fields are extracted and a dynamic form is popped up with these fields as label and text box.
- The enni-id and uni-id will be queried from AAI and are populated as a list in the fields. The user has to choose the appropriate enni and uni id from the list.

7 Template Inputs

name:	<input type="text"/>	uni-id:	<input type="text"/>
enni_id:	<input type="text"/>	uni_coding-func:	<input type="text"/>
uni_client-protocol:	<input type="text"/>	due_date:	<input type="text"/>
end_date:	<input type="text"/>		

inputs:

name:

type: string

description: Service name required: false

uni_id:

type: string

description: Operator UNI identifier

required: true

enni_id:

type: string

description: Operator ENNI identifier

required: true

uni_coding-func:

type: string

description: Coding function

required: true

uni_client-protocol:

type: string

description: 'Client protocol. Must be one of: Ethernet, FibreChannel, SDH, or SONET'

required: true

due_date:

type: string

description: Due date

required: true

end_date:

type: string

description: End date

required: true

7. For MDONS use case, a form needs to be created populating the mandatory input fields from the template.

- Once the mandatory fields are filled and click ok to create the service instance call which is made to the SO-Service creation service.

/api/so-serviceInstances/v3

- Service Operation Response Object is received from SO with default properties Service ID and Operation ID.
- The Service Instance and Service Instance operation objects are created and saved in the PostgreSQL data base.
- In addition with these, in terms of MDONS use case another property "statusDetails" needs to be added, which holds the status message (Reason for failure) from SO.
- For this property a column needs to be added in service_instance_operations table with column name operation_status_details.
- An additional Column will be added in the UI service List page to show the status details as shown below.

NO	Name	Instance ID	Use Case	Status	Status Details	Action
1	MDONS Test 1	4bffe5f7-fbe1-4601-b4a5-fdcf196df50b	MDONS		Success	
2	MDONS Test 2	32cd906f-ee7a-4ecf-b1c3-2fce4d59f803	MDONS		Failed Due to Resource Un-Availale	
3	MDONS Test 3	32cd906f-ee7a-4ecf-b1c3-2fce4d59f8ad	MDONS	Creating 99% 		

NO	Name	Instance ID	Use Case	Status	Status Details	Action
1	MDONS Test 1	4bffe5f7-fbe1-4601-b4a5-fdcf196df50b	MDONS		Success	
2	MDONS Test 2	32cd906f-ee7a-4ecf-b1c3-2fce4d59f803	MDONS		Failed Due to Resource Un-Availale	
3	MDONS Test 3	32cd906f-ee7a-4ecf-b1c3-2fce4d59f8ad	MDONS	Creating 99% 		

8. After Service creation, the progress of the service creation is re-queried from SO component until the progress is 100% from below path mapping by passing service id and operation id. This API call returns back a response with the percentage of operational status. This API is queried until the operation status become 100%.

/api/so-serviceInstances/v3/{serviceId}/operations/{operationId}

Sample Response:

```
{
  "operation":{
    "serviceId":"bf82273f-f7d1-4edb-b291-e2f5bb3a4674",
    "operationId":"440484be-7b6e-4ca3-841a-a9d4661ec6fa",
    "serviceName":"svc-Customer1-orange-001",
    "operation":"Create",
    "result":"success",
    "reason":"",
    "userId":"2R65T",
    "operationContent":"",
    "progress":"100",
    "operateAt":"2020-09-06T18:00:00+09:00",
    "finishedAt":"2020-09-06T21:00:00+09:00"
  }
}
```

- The status Green – Created, Red – Failure/ Error
- Service status is decided by the status response code returned.
 1. 2001 – Available
 2. 2002 – Un-Available

9. The above MDONS create service screen needs to be designed.

5. Get and Show MDON's Instance

This screen shows the service detail and status of the particular service which is selected from the list.

The screenshot shows the 'Use case ui' dashboard. On the left is a navigation menu with options: Home, Customer, Monitor, Services (highlighted with a red box), Lifecycle Management, Package Management, and Network Topology. The main content area has two dropdowns: 'Customer' and 'Service Type'. Below these are two summary cards: 'E2E' (End To End Service) with 0 Failed, 1 Success, and 0 In Progress; and 'NS' (Network Service) with 1 Failed, 3 Success, and 0 In Progress. A table below lists instances:

NO	Name	Instance ID	Use Case
1	nsService44	4bffe5f7-fbe1-4601-b4a5-fdcf196df50b	Network Service
2	SOTNVPNIinfra	32cd906f-ee7a-4ecf-b1c3-2fce4d59f803	SOTN
3	SOTNVPNIinfra	32cd906f-ee7a-4ecf-b1c3-2fce4d59f8ad	SOTN
4	CCVPN_test	32cd906f-ee7a-4ecf-b1c3-2fce4d59f8e0	CCVPN

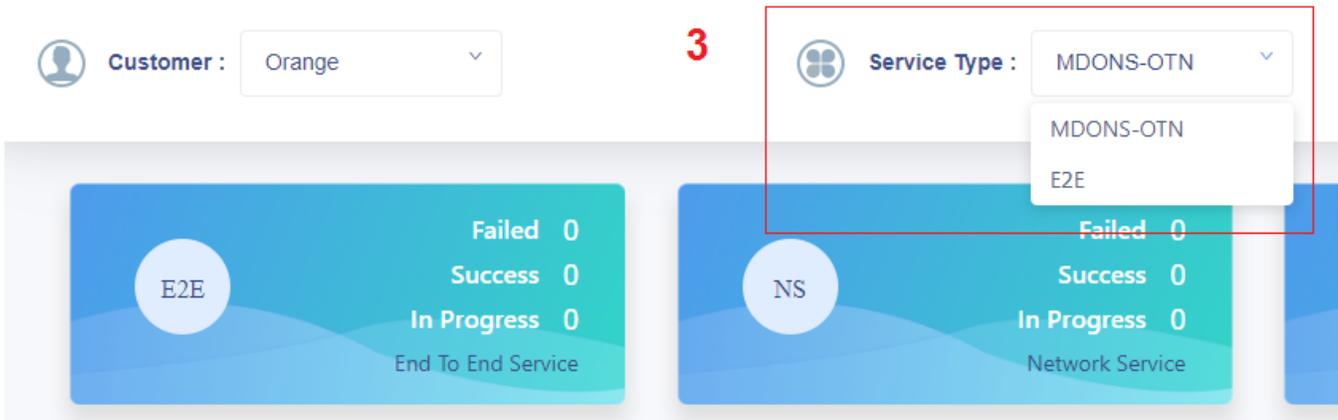
1. To view the detailed status of the service, go to Services tab and select Life cycle management. List of services created and their progress are listed as table in this page.

The screenshot shows the 'Use case ui' dashboard with the 'Customer' dropdown menu open, displaying a list of customers: Orange1, Orange, Orange 2, AT&T, Verizon, and Verizon2. The 'Service Type' dropdown is also visible. Below the dropdowns are two summary cards: 'E2E' (End To End Service) with 0 Failed, 1 Success, and 0 In Progress; and 'NS' (Network Service) with 1 Failed, 3 Success, and 0 In Progress. A table below lists instances with a 'Status' column:

NO	Name	Instance ID	Use Case	Status
1	nsService44	4bffe5f7-fbe1-4601-b4a5-fdcf196df50b	Network Service	🟢
2	SOTNVPNIinfra	32cd906f-ee7a-4ecf-b1c3-2fce4d59f803	SOTN	🔴

2. On the page init load Customer list of customers are fetched from AAI via UUI server by the following path.

/api/aa-business/v13/customers



3. On selecting the Customer, Service type is fetched by passing selected customer ID from AAI via UII server by following service call.

`/api/aa-business/v13/customers/customer/{global-customer-id}`

- Once the listed service type is selected from the drop down list, the service table data is fetched by passing customer id, service type, page index and page size to UII server. The Server in turn calls the AAI service for querying the available service instances for the specific customer id and service type.

`/api/aa-business/v13/customers/customer/{global-customer-id}/service-subscriptions/service-subscription/{service-type}/service-instances`

- The fetched service instance data is displayed as below.

NO	Name	Instance ID	Use Case	Status	Status Details	Action
1	MDONS Test 1	4bffe5f7-fbe1-4601-b4a5-fdcf196df50b	MDONS	✔	Success	☰ ☒ ⋮
2	MDONS Test 2	32cd906f-ee7a-4ecf-b1c3-2fce4d59f803	MDONS	✘	Failed Due to Resource Un-Availale	☰ ☒ ⋮
3	Mdns Test 3	32cd906f-ee7a-4ecf-b1c3-2fce4d59f8ad	MDONS	✔	Creating Successful	☰ ☒ ⋮
4	MDONS Test 4	32cd906f-ee7a-4ecf-b1c3-2fce4d59f8c0	MDONS	✔	Success	☰ ☒ ⋮

4. Click on the bars icon of the particular service to which you need to check the status and details.

NO.	Name	Description	eni-id	enni-id	Action
1	svc-Customer1-orange-001	svc-Customer1-orange-001	58479cc-8425-4841-9abe-2e8aeba6dd39	b55a1942-fcb7-4511-bb76-02bbe52c7c85	☰
2	svc-Customer1-orange-002	svc-Customer1-orange-002	8847s9c2-1455-831-dadbe-5epaebaaddm9	440484be-7b6e-4ca3-841a-a9d4661ec6fa	☰

5. The detail service view component is called and the created service template parameters are populated as shown below.

6. Delete MDONS Instance

The screenshot shows the 'Services' tab selected in the left sidebar. The main area displays two summary cards for 'E2E' and 'NS' services, and a table listing service instances.

NO	Name	Instance ID	Use Case
1	nsService44	4bffe5f7-fbe1-4601-b4a5-fdcf196df50b	Network Service
2	SOTNVPNinfra	32cd906f-ee7a-4ecf-b1c3-2fce4d59f803	SOTN
3	SOTNVPNinfra	32cd906f-ee7a-4ecf-b1c3-2fce4d59f8ad	SOTN
4	CCVPN_test	32cd906f-ee7a-4ecf-b1c3-2fce4d59f8c0	CCVPN

1. To delete the service, go to Services tab and select Life cycle management. Services which have been created will be listed.

NO	Name	Instance ID	Use Case	Status	Status Details	Action
1	MDONS Test 1	4bffe5f7-fbe1-4601-b4a5-fdcf196df50b	MDONS		Success	
2	MDONS Test 2	32cd906f-ee7a-4ecf-b1c3-2fce4d59f803	MDONS		Failed Due to Resource Un-Availbale	
3	MDONS Test 3	32cd906f-ee7a-4ecf-b1c3-2fce4d59f8ad	MDONS	Creating 99%		

2. Click on the Delete icon in the actions column of the particular service instance needs to be deleted.

Delete ✕

*** Are you sure delete this instance?**

Instance Name :

Instance ID :

3. A delete model pop up window appears with the specific instance name and instance id.

- Click on OK to proceed with the delete ACTION.
- A delete http call made to the UI server with Service ID.
- /uui-lcm/services/{serviceld}
- Again a call is made to SO service with service ID and request body which has global subscriber id and service type at /api/so-serviceInstances/v3/{serviceld}.
- A delete response is returned back from SO which has operation id.
- Using this operation id a delete service instance operation is created and updated in UI server local DB (PostgreSQL).
- The service progress loop is initiated which query the progress of the operation to SO at /api/so-serviceInstances/v3/{serviceld}/operations/{operationId}.
- The operation is status is also updated in the local DB with service instance operation object.

