

# CPS-333 Network Configuration Management (NCMP)

## scope for I release considerations

- [Reference Material](#)
- [Open Issues & Decisions](#)
- [Istanbul Roadmap](#)
- [Priorities & Scope for Istanbul Release](#)
- [API Changes](#)
  - [Datastore, Paths and Format Combinations for Read Operations](#)
    - [Read Example](#)
  - [Datastore, Paths and Format Combinations for Write Operations](#)
    - [Write Example](#)
- [Epics Summary](#)
- [Data Stores](#)
  - [Datastore Mapping in ONAP DMI Plugin impl.](#)
  - [Component description diagram](#)
- [Interfaces](#)
- [Sequence diagrams](#)

## Reference Material

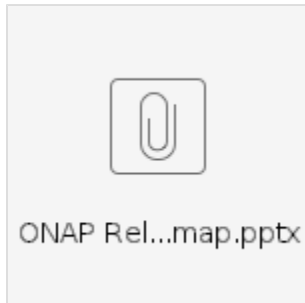
- [CPS-333](#) - Getting issue details... STATUS
- [REQ-717](#) - Getting issue details... STATUS
- [Release Planning: Istanbul](#)
- <https://wiki.onap.org/download/attachments/93006150/CPS-NCMP-Design.pptx?version=1&modificationDate=1614348138000&api=v2>

## Open Issues & Decisions

| # | Description   | Notes  | Decision   |
|---|---|--|--|
| 1 | How to deal with hardcoded E2E NW Slicing model, spike  | 23 Apr 2021 Discussons started with E2E NW Slicing Team, to be continued next week   |  |
| 2 | Which Simulator to use? NetConf Simulator Netopeer v2 <ol style="list-style-type: none"> <li>1. Nokia</li> <li>2. Ericsson</li> <li>3. Honeycomb(?)</li> <li>4. <a href="#">lighty netconf simulator</a></li> </ol> | <a href="#">Rahul Tyagi</a> Ericsson more lightweight, more experience.<br>Available CSIT standard setup<br><br><a href="#">Swaminathan Seetharaman</a> mentioned 'Honeycomb' simulator they used before. Need to investigate if it supports get-schema method over RESTConf/NetConf<br><br>Need to check if we can use it with E2E NW Slicing model   | For E2E use case Honeycomb simulator will be used. If needs be for certain non E2E functionality where this simulator has limitations the team can still decide to test/develop against NeToPeer simulator |
| 3 | Should (ONAP) dmi-instance be as separate (springboot) application<br>Could be part of DMI Manager  | DMI-Instance interface should be an open standard  | <a href="#">Tony Finnerty</a> and Team to allow easy integration of future 3rd part DMI-Plugin instances a separate application with its own REST Interface is required                                    |
| 4 | What datastores are supported /recognized by NCMP Interface in Istanbul Release   | Suggested, see <a href="#">Datastore paths and format combinations</a> section below <ol style="list-style-type: none"> <li>1. ncmp-datastores:operational (will use cached data when sync=On)</li> <li>2. ncmp-datastores:running (will forward to DMI-Plugin in Istanbul release)</li> <li>3. ncmp-datastores:passthrough-operational (always forward to DMI-Plugin)</li> <li>4. ncmp-datastores:passthrough-running (always forward to DMI-Plugin)</li> </ol> | Suggestions in previous column were agreed in CPS Weekly meeting 04 Jun 2021   |
| 5 | What datastores are supported /recognized by (ONAP) DMI-Plugin Interface in Istanbul Release  | Suggested exact same as NCMP supported options <ol style="list-style-type: none"> <li>1. ncmp-datastores:operational (for cmhandles with sync=Off)</li> <li>2. ncmp-datastores:running</li> <li>3. ncmp-datastores:passthrough-operational</li> <li>4. ncmp-datastores:passthrough-running</li> </ol>  | Suggestions in previous column were agreed in CPS Weekly meeting 04 Jun 2021   |

|   |  |  |   |
|---|--|--|---|
| 6 | How to handle any other (custom) datastore value         |  | Agreed in CPS Weekly meeting 04 Jun 2021 : Both NCMP and DMI-Plugin will not support any other datastore value in the Istanbul timeframe                          |
| 7 | What to do with old CM-Proxy interface implemented in H. | Methods never really used by E2E Network Slicing team. | Methods will be 'deprecated' using Open API option for that.. Replace and refactor other layers as needs when new methods using the new API are being implemented |

## Istanbul Roadmap



## Priorities & Scope for Istanbul Release

| Priority | Description  | Notes  | Istanbul commitment /delivery | Jira(s)  |
|----------|--|--|-------------------------------|--|
| 1        | Publish and Share NCMP Rest interface proposal   | detailed for Istanbul scope, general structure only for functionality related to later releases  | Committed & Delivered         | <a href="#">CPS-391</a> - Getting issue details... <span>STATUS</span>   |
| 2        | Publish and Share DMI Plugin Rest interface proposal                                   | detailed for Istanbul scope, general structure only for functionality related to later releases  | Committed & Delivered         | <a href="#">CPS-390</a> - Getting issue details... <span>STATUS</span>   |
| 3        | Register a DMI Plugin with NCMP  | DMI plugin registration is implicit part of CM handle registration (inventory update), REST endpoint on NCMP (can be multiple calls)   | Committed & Delivered         | <a href="#">CPS-350</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-384</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-352</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-353</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-389</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-442</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-403</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-404</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-405</a> - Getting issue details... <span>STATUS</span> |
| 4        | Support method to add initial inventory (e.g. hidden REST endpoint on ONAP DMI plugin) | <p>Additional properties for ONAP plugin may need ODL mountpoint, stored by NCMP and later provided to DMI plugin for each request. The mountpoint is created outside of CPS and DMI plugin.</p> <p>Note 1: This method (hidden REST endpoint) is expected to be a temporary solution, additional options are part of stretch goal (#13).</p> <p>Note 2: Adding the initial inventory should also trigger a model-sync for each node in that inventory (step #8)</p> | Committed & Delivered         | <a href="#">CPS-445</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-385</a> - Getting issue details... <span>STATUS</span><br><a href="#">CPS-406</a> - Getting issue details... <span>STATUS</span>   |

|    |   |   |  |  |
|----|---|---|--|--|
| 5  | Support <code>ncmp-datastores: passthrough-operational</code> for reading data (single CM handle, synchronous only)         | See <a href="#">CPS-391</a> page for details about supported operations and combinations. Note: There can be some overlap between work items for #5, #6, #11 and #12.<br>RestConf: content=all<br>NO conversion. NO schema's in OpenAPI Yml (just String) | Committed & Delivered  | <a href="#">CPS-388</a> - Getting issue details... <input type="button" value="STATUS"/><br><a href="#">CPS-394</a> - Getting issue details... <input type="button" value="STATUS"/><br><br><a href="#">CPS-486</a> - Getting issue details... <input type="button" value="STATUS"/><br><a href="#">CPS-487</a> - Getting issue details... <input type="button" value="STATUS"/>   |
| 6  | Support <code>ncmp-datastores: passthrough-running</code> for reading and writing data (single CM handle, synchronous only) | See <a href="#">CPS-391</a> page for details about supported operations and combinations. Note: There can be some overlap between work items for #5, #6, #11 and #12.<br>Almost identical to step #5 above just RestConf: content=config                  | Committed but Partly Delivered: <ul style="list-style-type: none"> <li>'create' delivered</li> <li>'update' &amp; 'delete' delayed until Jakarta. See <a href="#">R10 NCMP Requirements</a></li> </ul> | <a href="#">CPS-395</a> - Getting issue details... <input type="button" value="STATUS"/><br><a href="#">CPS-562</a> - Getting issue details... <input type="button" value="STATUS"/><br><br><a href="#">CPS-532</a> - Getting issue details... <input type="button" value="STATUS"/><br><a href="#">CPS-575</a> - Getting issue details... <input type="button" value="STATUS"/>   |
| 7  | Inventory Changes: define and implement interface in NCMP to allow DMI plugin to add, update & remove CM handles            | Overlap with #3, used by #4 and #13   | Committed & Delivered  | <a href="#">CPS-443</a> - Getting issue details... <input type="button" value="STATUS"/><br><a href="#">CPS-444</a> - Getting issue details... <input type="button" value="STATUS"/>   |
| 8  | Model discovery (automatic retrieval and caching of model information in NCMP for a CM handle)                              | NCMP queries DMI plugin that provides information, for ONAP retrieved via SDNC/ODL  | Committed & Delivered  | <a href="#">CPS-386</a> - Getting issue details... <input type="button" value="STATUS"/><br><br><a href="#">CPS-483</a> - Getting issue details... <input type="button" value="STATUS"/><br><a href="#">CPS-484</a> - Getting issue details... <input type="button" value="STATUS"/><br><a href="#">CPS-505</a> - Getting issue details... <input type="button" value="STATUS"/><br><a href="#">CPS-506</a> - Getting issue details... <input type="button" value="STATUS"/><br><br><a href="#">CPS-508</a> - Getting issue details... <input type="button" value="STATUS"/><br><br><a href="#">CPS-531</a> - Getting issue details... <input type="button" value="STATUS"/> |
| 9  | Retrieve list of modules (names) for a CM handle  | Used by applications to get cached information from NCMP about models   | Committed & Delivered  | <a href="#">CPS-485</a> - Getting issue details... <input type="button" value="STATUS"/>   |
| 10 | Explicit (initial) <b>data-sync</b> for a CM handle (extend model-sync started in step #8)                                  | Triggered by client using REST endpoint on NCMP   | Delayed until Jakarta  | <a href="#">CPS-507</a> - Getting issue details... <input type="button" value="STATUS"/>   |
| 11 | Support <code>ncmp-datastores: operational</code> for reading data (single CM handle, synchronous only)                     | See <a href="#">CPS-391</a> page for details about supported operations and combinations. Note: There can be some overlap between work items for #5, #6, #11 and #12.   | Delayed until Jakarta  | <a href="#">CPS-382</a> - Getting issue details... <input type="button" value="STATUS"/>   |
| 12 | Support <code>ncmp-datastores: running</code> for reading and writing data (single CM handle, synchronous only)             | See <a href="#">CPS-391</a> page for details about supported operations and combinations. Note: There can be some overlap between work items for #5, #6, #11 and #12.   | Delayed until Jakarta  |  |
| 13 | Support dynamic inventory changes (ONAP DMI Plugin)   | React to events from AAI sent over DMaaP, in turn using API in #7 for updates<br><br>Possible in a generic way o it can also listen to similar events sent by SDN-R (as suggested by <a href="#">Ahila P</a> )  | Delayed until Jakarta  | <a href="#">CPS-392</a> - Getting issue details... <input type="button" value="STATUS"/><br><a href="#">CPS-393</a> - Getting issue details... <input type="button" value="STATUS"/>   |
| 14 | Retrieve list of cm-handles that have a given module  |   | Delayed until Jakarta  |  |
| 15 | Automatic (optional) Data Sync  | Metadata (per cmHandle) controls whether this will happen or not  | Out of Scope   |  |
| 16 | Implement -async option for CRUD and Patch operations   |   | Out of Scope   |  |
| 17 | Support multiple cmHandles in a single call (bulk)  |   | Out of Scope   |  |
| 18 | Support all cmHandles in a single call?   |   | Out of Scope   |  |

|    |                                    |  |              |  |
|----|------------------------------------|--|--------------|--|
| 19 | Invoke YANG modelled RPC or action |  | Out of Scope |  |
|----|------------------------------------|--|--------------|--|

## API Changes

To support the added functionality in NCMP, changes are expected in the CPS external interface **CPS-E-05**. (For overview of CPS interfaces, see the [CPS architecture page](#).) Changes are planned in the following areas, matching items in the scope/priority table above:

- New APIs for reading data for CM handle
- New APIs for writing data for CM handle
- New API to request data sync for CM handle
- New API to retrieve model information for CM handle
- (stretch goal) New API to retrieve list of CM handles supporting a given module

Some existing CPS-E-05 APIs may be removed or deprecated if not currently used. See Decision #7

In addition to external APIs, extensions may be made to the CPS Core library to support some NCMP use cases. Plugin APIs will also be added to support communication between NCMP and DMI plugins.

Excerpt from [CPS-391Spike: Define and Agree NCMP REST Interface](#) :

## Datastore, Paths and Format Combinations for Read Operations

| # | State     | Input                         |                              |                             |   | Behavior  | Data        |                             | Notes   |
|---|-----------|-------------------------------|------------------------------|-----------------------------|---|---|-------------|-----------------------------|---|
|   | Data-Sync | Datastore parameter           | Expected resourcePath format | Accept-Header               | Fields (filter)   |   | Data Source | Included DataNodes (config) |   |
| 1 | On        | Not Specified                 | cpsPath                      | application /yang-data+json | N/A   | Not supported   | N/A         | N/A                         |   |
| 2 | On        | Not Specified                 | cpsPath                      | application /json           | N/A   | Not supported   | N/A         | N/A                         |   |
| 3 | Off       | Not Specified                 | cpsPath                      | application /yang-data+json | N/A   | Not supported   | N/A         | N/A                         |   |
| 4 | Off       | Not Specified                 | cpsPath                      | N/A                         | N/A   | Not supported   | N/A         | N/A                         | there are NO DataNode objects in CPS to output as JSON)   |
| 5 | Off       | Not Specified                 | other then cpsPath           | N/A                         | N/A   | Not supported   | N/A         | N/A                         | Not supported Since NCMP can only convert cpsPaths  |
| 6 | On   Off  | ncmp /passthrough-operational | NCMP does not parse          | NCMP does not parse         | depends on DMI-Plugin (supported in ONAP)   | Resolve DMI plugin<br><br>Forward request to plugin<br>Output received response | DMI-Plugin  | config + non-config         | The DMI plugin may error if the RP or accept header are not supported. The DMI plugin may forward the request without processing too.                     |
| 7 | On   Off  | ncmp /passthrough-running     | NCMP does not parse          | NCMP does not parse         | depends on DMI-Plugin (supported in ONAP)   | Resolve DMI plugin<br><br>Forward request to plugin<br>Output received response | DMI-Plugin  | config-only                 |   |
| 8 | On        | ncmp /operational             | cpsPath                      | application /yang-data+json | <ul style="list-style-type: none"> <li>• Not supported in Istanbul releases.</li> <li>• <i>Considered for Kohn Release</i></li> </ul> | Read from cache<br><br>output: application /yang-data+json                      | CPS-Core    | config + non-config         | NCMP/CPS-Core needs to remove DataNode wrapping   |
| 9 | On        | ncmp /operational             | cpsPath                      | application /json           | <ul style="list-style-type: none"> <li>• Not supported in Istanbul releases.</li> <li>• <b>Planned for Kohn Release</b></li> </ul>    | Read from cache<br><br>output: application/json                                 | CPS-Core    | config + non-config         | Output will use DataNode wrapping (as is from CPS-Core)<br><br>For forwarding (cached config off) dmi-reposne need to be wrapped explicitly in 'DataNode' |

|    |          |                  |         |                            |  |   |            |                     |  |
|----|----------|------------------|---------|----------------------------|--|---|------------|---------------------|--|
| 10 | Off      | ncmp/operational | cpsPath | application/yang-data+json | to be determined in spike, see issue #28 | Resolve DMI plugin<br><br>Convert cpsPath to RESTConfPath*<br><br>Forward request to plugin   Read from DMI plugin<br><br>Output application/yang-data+json | DMI-Plugin | config + non-config |  |
| 11 | On   Off | ncmp/running     | cpsPath | application/yang-data+json | to be determined in spike, see issue #28 | Resolve DMI plugin<br><br>Convert cpsPath to RESTConfPath*<br><br>Forward request to plugin   Read from DMI plugin<br><br>Output application/yang-data+json | DMI-Plugin | config-only         |  |

**\*Note** Convert cpsPath to RESTConfPath wil only support 'absolute' cpsPath for conversion no query-type paths

## Read Example

### Read with fields

```
{ncmpRoot}/ncmp/v1/ch/<cmHandle>/data/ds/<datastore>/{dataResourceIdentifier}?fields={fieldsExpression}
```

```
URI : {ncmpRoot}/ncmp/v1/ch/node123/data/ds/ncmp-datastores:operational/TopElement[@id=1]/SomeFunction[@id=1]?fields=cell-model:Cell/attributes(attr1;attr2)
```

```
Header :
    Accept : application/yang-data+json
```

```
Response :
    200 OK
{
  "function-model:SomeFunction": [
    {
      "id": "1",
      "cell-model:Cell": [
        {
          "id": "Cell-001",
          "attributes": {
            "attr1": "value1",
            "attr2": "value2"
          }
        },
        {
          "id": "Cell-002",
          "attributes": {
            "attr3": "value3",
            "attr4": "value4"
          }
        }
      ]
    }
  ]
}
```

Works Items for above.

| # | Description                           | Component | Enables |
|---|---------------------------------------|-----------|---------|
| 1 | Forward request from NCMP to CPS-Core | NCMP      | 8,9     |

|   |   |               |                     |
|---|---|---------------|---------------------|
| 2 | Forward request from NCMP to DMI-Plugin   | NCMP          | 6,7                 |
| 3 | Convert json (dataNode) to yang-data+json | CPS-Core/NCMP | 8                   |
| 4 | Convert cpsPath to RESTConf Path          | NCMP          | 10,11               |
| 5 | Enhance &fields parameter where needed    | NCMP          | 10,11+fields option |
| 6 | NOT Supported                             | N/A           | 1,2,3,4,5           |

## Datastore, Paths and Format Combinations for Write Operations

- Write operations are only supported on the *ncmp-datastores:running* and *ncmp-datastores:passthrough-running* datastores
- The Data Target for all write operation is DMI-Plugin
- Write operations are only supported for config=true data
- Fields and similar parameters are not supported for write operations

|   | State     | Input     |                          |                              |  | Behavior  | Notes   |
|---|-----------|-----------|--------------------------|------------------------------|--|---|---|
| # | Data-Sync | Operation | Datastore parameter      | Expected resourcePath format | Content-Type   |   |   |
| 1 | On   Off  | Create    | ncmp/passthrough-running | NCMP does not parse          | NCMP only checks it is valid JSON, then embeds the data in a larger JSON structure (see CPS-390 page)              | Resolve DMI plugin<br><br>Forward request to plugin<br>Output received response (success or failure)  | The DMI plugin may error if the RP or content type are not supported.<br>The DMI plugin may forward the request without processing too. |
| 2 | On   Off  | Replace   | ncmp/passthrough-running | NCMP does not parse          | NCMP only checks it is valid JSON, then embeds the data in a larger JSON structure (see CPS-390 page)              | Resolve DMI plugin<br><br>Forward request to plugin<br>Output received response (success or failure)  | The DMI plugin may error if the RP or content type are not supported.<br>The DMI plugin may forward the request without processing too. |
| 3 | On   Off  | Delete    | ncmp/passthrough-running | NCMP does not parse          | NCMP doesn't expect any input data from application, will create request body to DMI plugin without embedded data. | Resolve DMI plugin<br><br>Forward request to plugin<br>Output received response (success or failure)  | The DMI plugin may error if the RP or content type are not supported.<br>The DMI plugin may forward the request without processing too. |
| 4 | On   Off  | Patch     | ncmp/passthrough-running | NCMP does not parse          | NCMP only checks it is valid JSON, then embeds the data in a larger JSON structure (see CPS-390 page)              | Resolve DMI plugin<br><br>Forward request to plugin<br>Output received response (success or failure)  | The DMI plugin may error if the RP or content type are not supported.<br>The DMI plugin may forward the request without processing too. |
| 5 | On   Off  | Create    | ncmp/running             | cpsPath                      | application/yang-data+json   | Resolve DMI plugin<br><br>Convert cpsPath to RESTConfPath<br><br>Forward request to plugin<br><br>Output received response (success or failure) |   |
| 6 | On   Off  | Update    | ncmp/running             | cpsPath                      | application/yang-data+json   | Resolve DMI plugin<br><br>Convert cpsPath to RESTConfPath<br><br>Forward request to plugin<br><br>Output received response (success or failure) |   |
| 7 | On   Off  | Delete    | ncmp/running             | cpsPath                      | N/A  | Resolve DMI plugin<br><br>Convert cpsPath to RESTConfPath<br><br>Forward request to plugin<br><br>Output received response (success or failure) |   |
| 8 | On   Off  | Patch     | ncmp/running             | cpsPath                      | application/yang-data+json<br><br>(*plain patch)   | Resolve DMI plugin<br><br>Convert cpsPath to RESTConfPath<br><br>Forward request to plugin<br><br>Output received response (success or failure) |   |

|   |          |       |              |         |                                      |   |  |
|---|----------|-------|--------------|---------|--------------------------------------|---|--|
| 9 | On   Off | Patch | ncmp/running | cpsPath | application/ <b>yang-patch</b> +json | Resolve DMI plugin<br>Convert cpsPath to RESTConfPath<br>Forward request to plugin<br>Output received response (success or failure) |  |
|---|----------|-------|--------------|---------|--------------------------------------|---|--|

## Write Example

Write Example

## Epics Summary

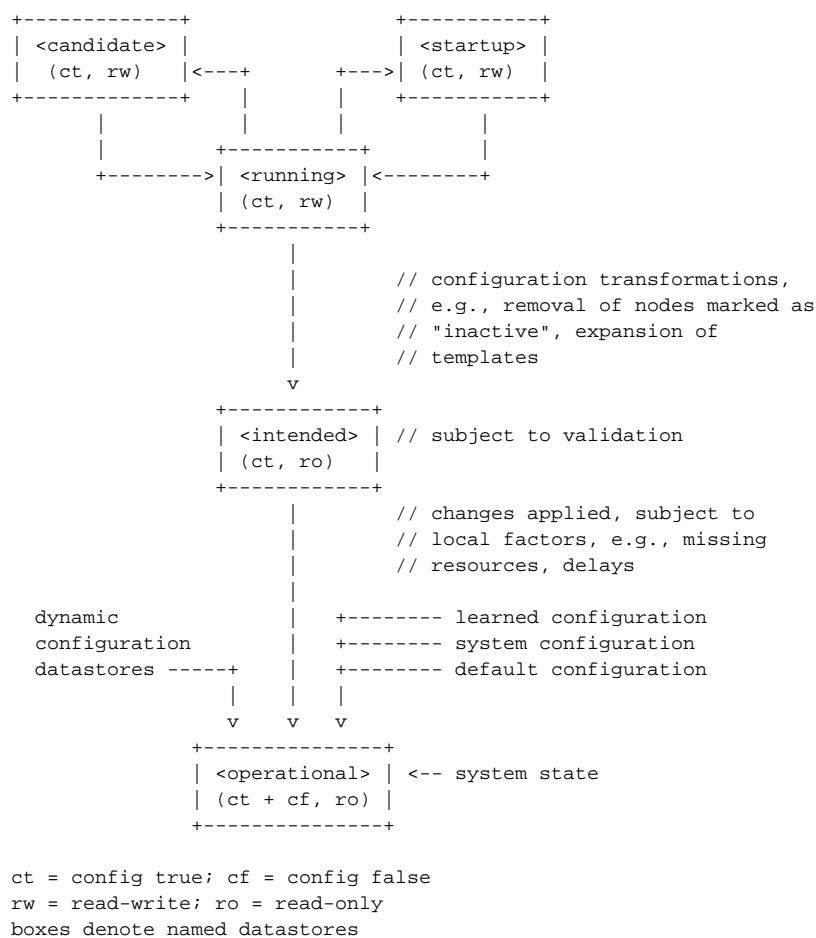
Jira filter all user stories in all epics: [0 issues](#)

| # | Epic   | Project /Component      | Description  | Comments   |
|---|--|-------------------------|--|--|
| 1 | E2E Network Slicing<br><br><a href="#">CPS-382</a> - Getting issue details... <b>STATUS</b>  | CPS/NCMP                | create rest interface on northbound for GET/POST /PUT/DELETE to access yang data.<br>Note. Currently driven by "E2E Network Slicing" Use Case mix of CPS-CORE and NCMP Interface   | 1. interfaces for yang data are: operational, intended, pass-through (selection can be done by parameter)<br>2. check with netconf-sim if current nw slicing yang model can be imported including its imports. |
| 2 | DMI-Registry<br><br><a href="#">CPS-350</a> - Getting issue details... <b>STATUS</b>         | (NCMP) DMI-Registry     | Create dmi-registry. Data Model Inventory Interface (DMI)<br><br>'Internal service' no public Java or REST API<br><br>responsibilities are<br><br>1. Store & retrieve cm-handle/dmi-plugin mapping in yang modelled CPS data table.<br>a. include additional data (key-value pars as provided by dmi-Plugin)<br>2. Dmi manager is to select the dmi instance at runtime for cm handle for a northbound (write) request |  |
| 3 | DMI-Plugin<br><br><a href="#">CPS-384</a> - Getting issue details... <b>STATUS</b>           | DMI-Plugin (ONAP impl.) | Common standard (REST and Java?) interface for extensibility<br><br>create dmi plugin for onap. Its responsibility of admin to provide instance of dmi.  | 1. extract dmi-plugin into separate container<br>consensus on rest interface is needed.  |
| 4 | Inventory<br><br><a href="#">CPS-385</a> - Getting issue details... <b>STATUS</b>            | DMI-Plugin (ONAP impl.) | <ul style="list-style-type: none"> <li>Capability to get initial Inventory</li> <li>Process Inventory updates</li> </ul>   |  |
| 5 | Model & Data Sync<br><br><a href="#">CPS-386</a> - Getting issue details... <b>STATUS</b>    | DMI-Plugin (ONAP impl.) |  |  |
| 6 | Pass-Through Support<br><br><a href="#">CPS-388</a> - Getting issue details... <b>STATUS</b> | DMI-Plugin (ONAP impl.) | 1. dmi plugin instance (onap-dmi) have sdnc connector to access sdnc.<br>2. call get/post/put/delete towards restconf interface (forward northbound write requests)  | Possibly overlaps with #1 E2E Network Slicing  |

|   |   |     |  |
|---|---|-----|--|
| 7 | <p>E2E Integration test for CPS /NCMP</p> <div> <p><b>INT-1902</b> - Getting issue details...</p> <p><b>STATUS</b></p> </div> | All |  |
|---|---|-----|--|

## Data Stores

IETF Datastores Architecture (source <https://datatracker.ietf.org/doc/html/rfc8342#page-11>)

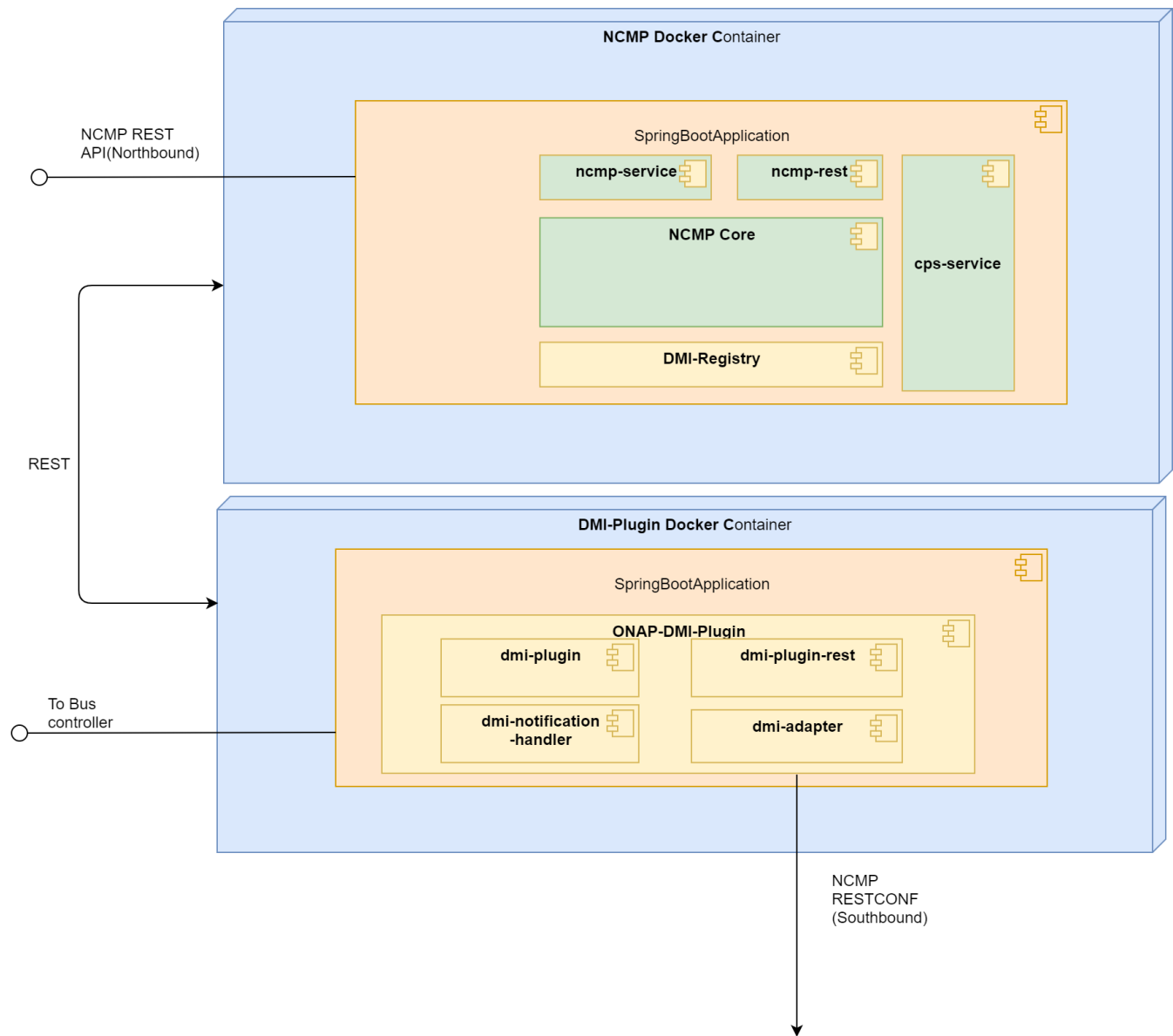


## Datastore Mapping in ONAP DMI Plugin impl.

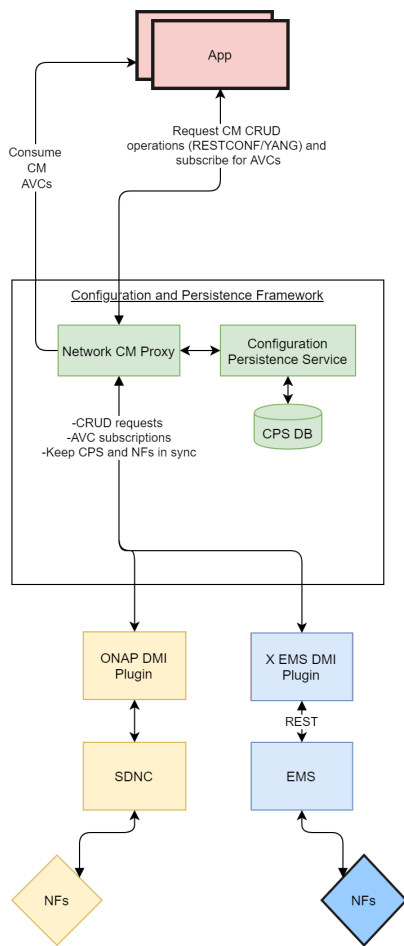
| # | Incoming DS value (NCMP & DMI Rest interfaces) | Outgoing (non-NMDA RestConf controller) | Notes         |
|---|--|---|---------------|
| 1 | /ds/ncmp-datastores:operational                | content=all                             | CT + CF, RO   |
| 2 | /ds/ncmp-datastores:running                    | content=config                          | CT, RW        |
| 3 | /ds/ncmp-datastores:passthrough-operational    | content=all                             | CT + CF, RO   |
| 4 | /ds/ncmp-datastores:passthrough-running        | content=config                          | CT, RW        |
| 5 | /ds/ <anything-else>                           | N/A                                     | Not supported |



# Component description diagram



## Interfaces



Sequence diagrams

## register dmi-plugin

This sequence diagram shows registry of dmi-plugin, so that it can be discovered at the time of cm-handle operations.

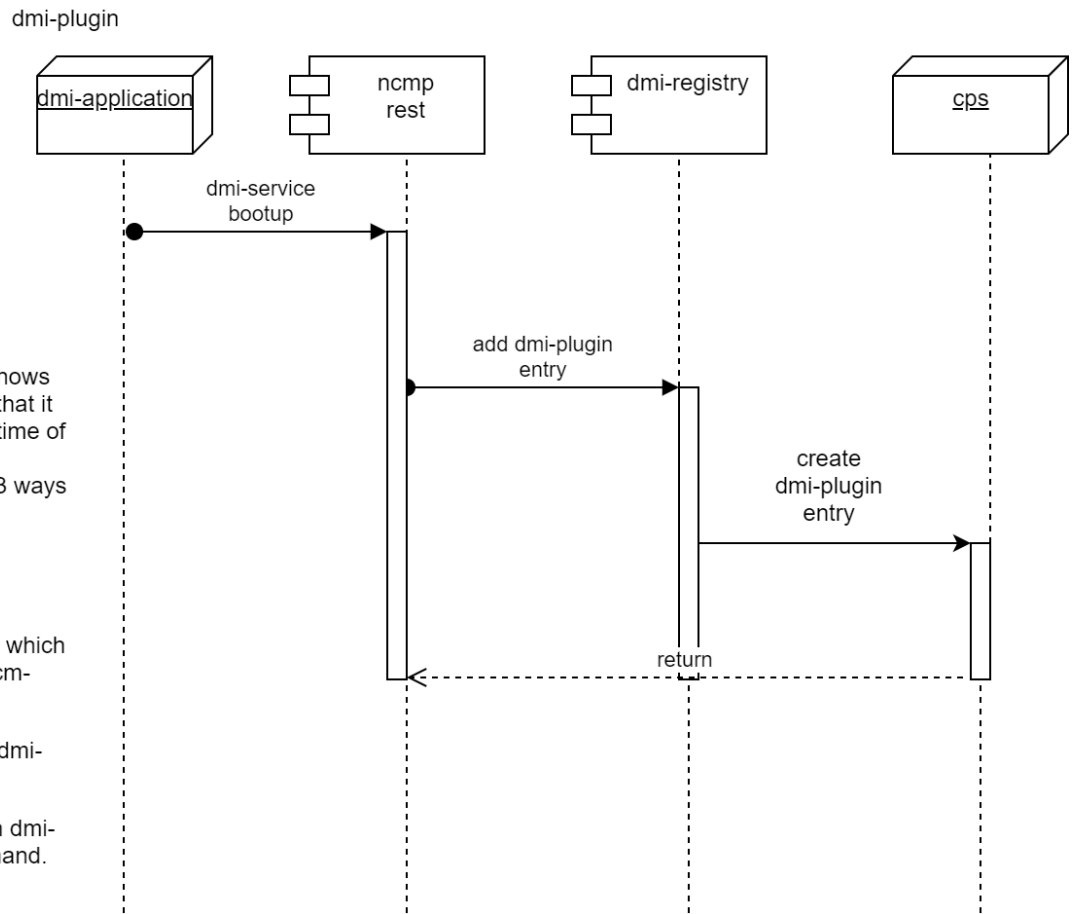
Registration can happen 3 ways

1. lazy
2. early
3. hardcoded

1. lazy: registration is one which registers itself when first cm-handle event happened.

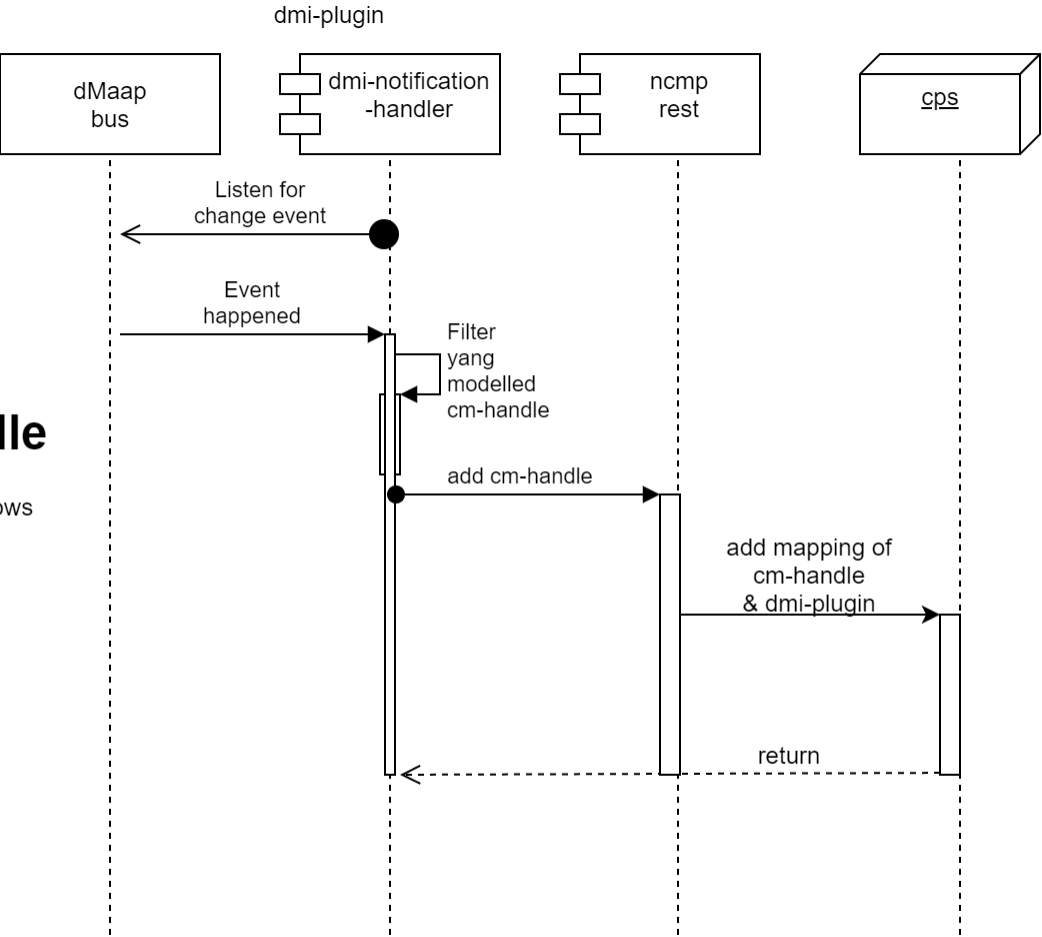
2. early: register as soon dmi-plugin comes up.

3 hardcoded: put a known dmi-pluginentry in cps beforehand.



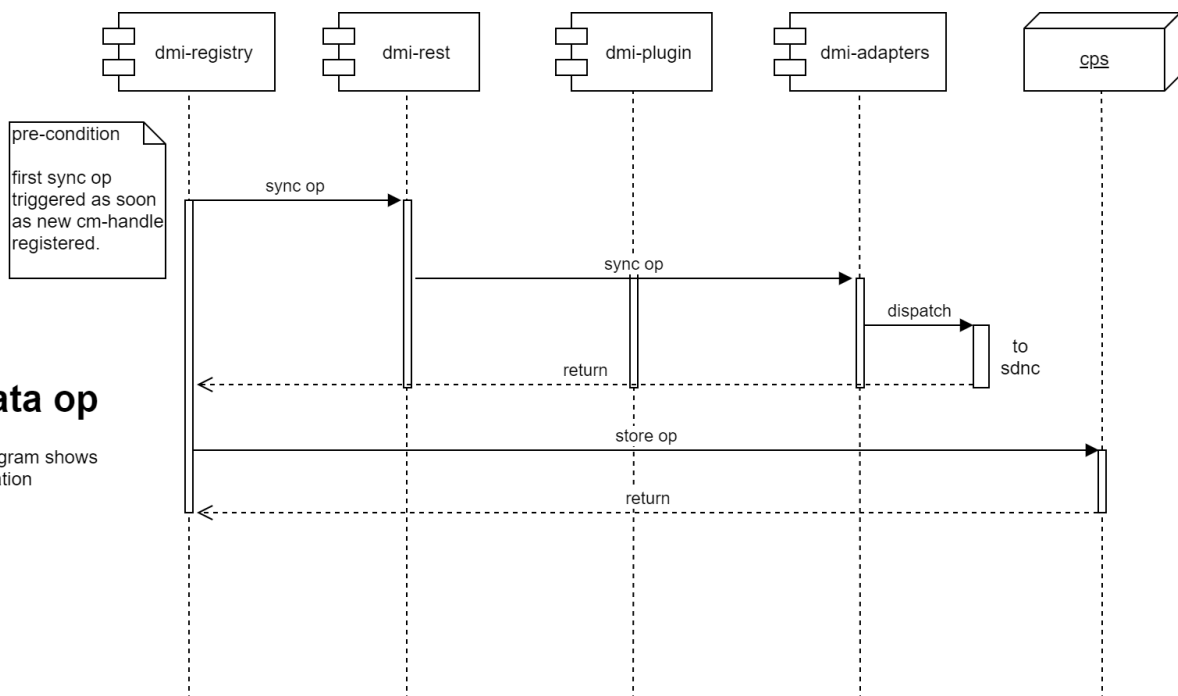
# add cm-handle

This sequence diagram shows flow for adding cm-handle



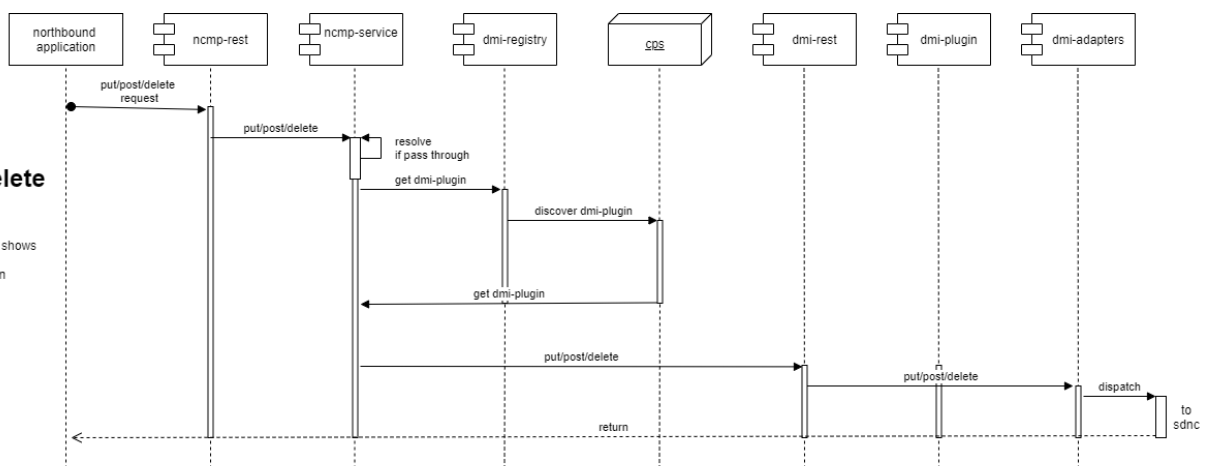
## sync model/data op

This sequence diagram shows  
flow for sync operation



## put/post/delete op

This sequence diagram shows  
flow for pass through  
put/post/delete operation



## get op

This sequence diagram shows flow for get operation

