CPS-391Spike: Define and Agree NCMP REST Interface

CPS-391 - Getting issue details... STATUS

Guiding Principles

1. NCMP REST Interface will follow/be inspired by RESTConf interface for easy acceptance of and transition to this interface
2. Will follow ONAP's RESTful API Design Specification
3. The interface will include the concept of data-stores inspired by Network Management *Datastore* Architecture (*NMDA*) and as used in RESTConf
4. The application should be able to easily switch between 'pass-through' and other datastores (also identical rest endpoint and responses)

References

Follow principles/patterns of RESTCONF RFC-8040 https://datatracker.ietf.org/doc/html/rfc8040
Follow principles/patterns of yang-patch RFC-8072 https://datatracker.ietf.org/doc/html/rfc8040
Follow principles/patterns of RESTCONF NMDA RFC-8527 https://datatracker.ietf.org/doc/html/rfc8527

Requirements

Please note this section was added long after the implementation and focuses on characteristic only.

Issues & Decisions

	Issues	Notes	Decisions
1	KPI for De-registration of 100 CM-handles	This was mentioned. Was this ever agreed, is this a valid use case that needs to be covered together with the Registration ?	Not priority for now, but acceptable if we match the registration req. #2 for de-reg 21 May 2024 kieran mccarthy Kolawole Adebisi-Adeolokun
2	DMI delay	Could we get some feedback on DMI-delays for other use cases as not mentioned in FS document	Awaiting for ETH feedback 21 May 2024 AP On Kolawole Adebisi-Adeolokun and Csaba Kocsis

Characteristics - WIP

It is proposed that reported characteristics will be used as a baseline for NCMP when agreed and sign-off.

	Operation	Concurrent requests /parallel	DMI Delay	Response size	Performance Requirement (Blue Stone tablet KPI)	Notes	Sign-Off

1	Registration of 20,000 CM-handles (in batches of 100)	1 (requests are sequential)	100 ms to get module references 1,000 ms to get module resources	N/A	• 11 CM-Handles/second as per Stone Tablet E2E Wich include module conversion warm-up • NCMP Budget: 22 Cm Handles/second	1. Batch Size: 100 (per request) Not using Module Set Tags 2. Time measured start of first rest-call until all cm handle states READY 3. 1,000 unique module references. 4. Five different types of Nodes. So 5 requests for Module Resources. Avg 200 modules each.	21 May 2024 Csaba Kocsis Toine Siebelink
2	De-registration of 100 CM-handles	1 (requests are sequential)	No Module delays	N/A	• 11 NEs/second • NCMP should target 22 NEs/Second	De-registration is currently not mentioned in Stone Tablet KPI or FS, however we have agreed to match the performance of registration for now as de-reg is also not a priority at this point in time	21 May 2024 Csaba Kocsis Toine Siebelink Kolawole Adebisi-Adeolokun
3	CM-handle ID search with Module filter	2.5 Run in parallel with #4	N/A	20,000 CM Handles i.e. 100*20.000 = 2MB	TBD will be derived when testing is done in ETH envi ETH/Daniel Hanrahan		TDB
4	CM-handle search with Module filter	2.5 Run in parallel with #3	N/A	20,000 CM Handles i.e. 500*20.000 = 10MB	TBD will be derived when testing is done in ETH envi ETH/Daniel Hanrahan		TDB
5	Synchronous single CM-handle pass-through read	10 Run in parallel with #6	Awaiting input from eth Csaba Kocsis	5 KB	25 (parallel) request/sec	Read are done in parallel with Write	TDB
6	Synchronous single CM-handle pass-through write (CUD)	10 Run in parallel with #5	Awaiting input from eth Csaba Kocsis	5 KB	13 (parallel) request/sec	No response is expected	TDB

Notes

1. This is for mixed TCs
2. Single KPIs will be monitored in NCMP owned pipeline with our performance every day(2 hrs interval) - Performance

Synchronous single cm-handle pass-through (read) requests

Parameter	Expectation	Notes	Sign-Off
Average Response Size	5KB	Shall not exceed 5KB	06 Dec 2023 Kolawole Adebisi-Adeolokun
Concurrent request	12 clients requests toward 1 NCMP simultaneously DMI also support 12 simultaneous requests	40ms of overhead on top of DMI latency for each requests, at most for NCMP request. This shall remain within 40ms for 12 parallel requests. Given the DMI delay below; this means up to 240 request per second	06 Dec 2023 Kolawole Adebisi-Adeolokun
DMI Delay	10ms	This is not in control of CPS. So for performance testing our stub should simulate a 10ms delay Assume DMI is 1.25 seconds average DMI response time for high latency, low latency =10 ms, this should also work for DMI Plugin. I.e 40ms ontop of the DMI. 1.25seconds+40ms= 1.29 seconds	06 Dec 2023Kolawole Adebisi-Adeolokun
Test Environment	1. CPS and NCMP		06 Dec 2023 Kolawole Adebisi-Adeolokun

		requests: cpu: 2000m memory: 2Gi limits: memory: 3Gi cpu: 3000m		
	2. Postgres			

	requests: cpu: 4000m memory: 1Gi limits: memory: 3Gi cpu: 6000m		
Security	Disable Basic Authentication in Springboot	If configurable from application yaml, then it’s acceptable.	06 Dec 2023 Kolawole Adebisi-Adeolokun

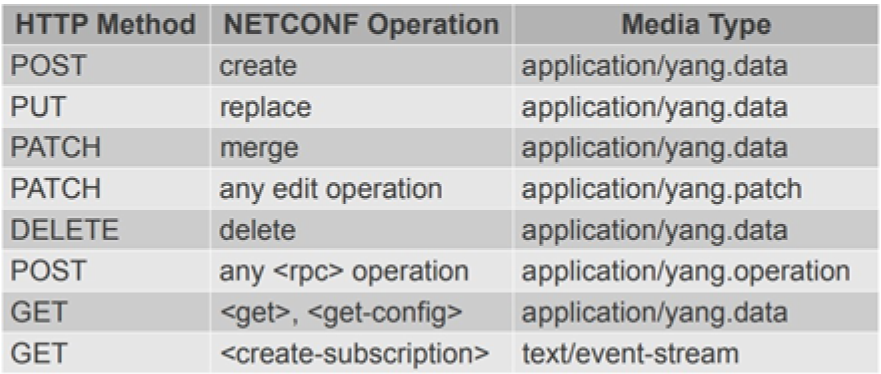
Open Issues & Decisions

	Description	Notes	Decision
1	Priority of async calls		In Istanbul, async calls are required only in pass-through cases. NCMP does not have to handle these requests
2	Will we use the data node wrapper on GET rest operations?	Currently, we wrap the response of GET operations using the data node wrapper.	we should mainly support yang-data/json controlled by "accept-header"

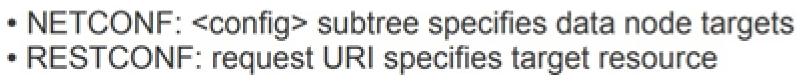
3	In the URI will we distinguish between data and operations (RFC calls) as part of the path?	e.g. http://localhost:8080/ncmp/v1/data http://localhost:8080/ncmp/v1/operation	This only applies to pass-through yes, we will distinguish between data and operation
4	Which query parameters will NCMP support?		Parent data resource identifier can handle any path using the same query parameter 1. cpsPath 2. RESTConf Path (pass-through) 3. Proprietary Path (pass-through)
5	Yml should include return types and examples of the payload		Legacy and new API documentation needs to include output examples. Task created, see CPS-401 - Getting issue details... STATUS
6	camel case or dash in URI		We will use a dash for param names e.g. cmHandle (although it has since been agreed we use 'ch' in this particular case) See no.3 https://restfulapi.net/resource-naming/
7	Insert /resourcePath in front of the resource path to prevent ambiguous paths	<OP>/ncmp/<v{vNumber}>/ch/<cmHandle>/<data|operations|{ncmp-operation}>/ds/{datastore}/[rp:]{resourcePath}?{query}	Optionally insert the resource path ('rp:') if it clashes with the current
8	Granularity of update scenarios (and priority)	1. Add child and its descendants (supported in cps core) 2. Add all list elements (supported in cps core) 3. Replace child and its descendants (supported in cps core) 4. Replace all list elements (pending in cps core) 5. Update single leaf (new) 6. Add list entry (new)	Priority is pass-through only so it depends on the RETSConf protocol that is supported. In Jakarta or if required by other projects more fine-grained 'operation' datastore update options can be implemented
9	Fallback option for datastore in release I		No, explicit datastore options will be used in Istanbul
10	fields is a rest conf option, investigate is it fully supported by onap		Supported in pass-through for ONAP DMI plugin but depending on the support by the actual target. The intention is to increase support 'fields' in future requirements following RFC-8040 for operational datastore etc.
11	Agree on URI syntax	Proposed syntax by CPS team <OP>/ncmp/<v{vNumber}>/ch/<cmHandle>/<data|operations|{ncmp-operation}>/ds/{datastore}/[rp:]{resourcePath}?{query}	review completed and proposed URI agreed
12	Will we combine query capabilities with update capabilities?		We have decided not to combine query capabilities with update capabilities
13	Description of header limitations		**HTTP Header Limitations** Some servers put size limitations on HTTP headers, making them unsuitable for storing cmHandle information. LIMITATION NOTE: server implementations put size limits on the headers meaning header contents should be designed carefully : Apache - 8K Nginx - 4K-8K IIS - 8K-16K Tomcat - 8K – 48K
14	Will NCMP support paths for pass-through:running		The plugin will not do transformation or validation of paths in the case of pass-through:running
15	Specification of path per cm handle		DMI Plugin can take cps paths or restconf paths and it needs to specify that per cm handle when cm handle is created
16	What is the default path for NCMP		In NCMP default will always be cps path and depending on the adapter we can change it as needed per cm handle
17	Fields parameter for ncmp /operational?		The fields parameter is ignored in ncmp/operational (in Istanbul release)
18	Is specifying the datastore mandatory?		Datastore is mandatory in Istanbul release
19	Register a DMI plugin with NCMP		DMI plugin is a part of cm handle registration. The rest endpoint on NCMP can be multiple calls

20	Retrieve list of modules (names) for a cmHandle		Retrieve a list of module names for cm handle - this will be used by ncmp to get the models. - assuming ncmp model discovery is complete and it is stored in cps core, this will come from cached information
21	Where will sync be implemented?		Implement sync in the dmi plugin and then have ncmp be able to pass on the request. This is not a bulk operation
22	Config-true only support (filter out config-false data)		Use datastore 'running' to select this but filtering not supported in I for cached data
23	Enable NCMP to convert cpsPath to mutliple options (RESTConf, netConf, leave as cpsPath)	When other DMI-Plugins are realized they might need a different conversion then the default from cspPath to RESTConf. This could be configured by using a known property for each cmHandle	Not required in Istanbul. But DB model can easily be updated to cater for this when needed
24	Datastore conversion in NCMP or DMI-Plugin	DMI-Plugin will know best how to convert. This will also reduce future impacts on NCMP for new options.	NCMP will do now conversion of datastore names
25	What datastore/s (name/s) is /are supported by NCMP to referred to the cached data. 'Operational' or 'running'	'operational' would imply RO and config=false data is included. 'To also support 'running' using the same data a filter would have to be applied	see supported datastore in I : Datastores
26	Consider fallback option when user specifies ncmp /operational but data is NOT synced		NCMP will forward requests for un-synced cmHandles to the DMI Plugin (Including required transformation of resource path etc.)
27	Support for &fields parameter when using cached data	1. not supported (ignored, not rejects, nice for future compatibility) 2. treat as 'no descendants' (low cost) 3. use to filter cached data	&Fields parameter will be ignored for 'cached' data in Istanbul timeframe long term expectation is to have support following RESTConf/ODL behavior as much as possible
28	Support for &fields parameter when forwarding to plugin for non-synced cmHandles	1. not supported (ignored, not rejects, nice for future compatibility) 2. treat as 'no descendants' (low cost) 3. translate (insert module names) and forward	A spike CPS-455 - Getting issue details... STATUS will be executed to determine the feasibility of option 3 and decide if it can make Istanbul scope
29	Response for Data Sync request (in Istanbul timeframe)	The action is blocking synchronous through whole stack (in I) so response could include the data returned by the node. However this seem incorrect for an 'action' so maybe the response should just be just an acknowledgment it is 'done'	No need to return data, just HTTP Code 200 (OK) will suffice

RESTCONF/NETCONF relationship



HTTP Method	NETCONF Operation	Media Type
POST	create	application/yang.data
PUT	replace	application/yang.data
PATCH	merge	application/yang.data
PATCH	any edit operation	application/yang.patch
DELETE	delete	application/yang.data
POST	any <rpc> operation	application/yang.operation
GET	<get>, <get-config>	application/yang.data
GET	<create-subscription>	text/event-stream



- NETCONF: <config> subtree specifies data node targets
- RESTCONF: request URI specifies target resource

NCMP URI

NCMP URI format to follow below pattern

<OP>/ncmp/<v{vNumber}>/ch/<cmHandle>/<data|operations|{ncmpAction}>/ds/{datastore}?[rp:]{resourcePath}&{options}

Below table shows the proposed interface, actual implementation might deviate from this but can be accessed from

- Gerrit Source
- Read-the-docs: https://docs.onap.org/projects/onap-cps/en/latest/design.html#offered-apis

URI		Mandatory or Optional
<OP>	the HTTP method	Mandatory
ncmp /	the ncmp root resource	Mandatory
<v{vNumber}>	version of the ncmp interface <path> is the target resource URI <query> is the query parameter list	Mandatory
ch/<cmHandle>	unique (string) identifier of a yang tree instance.	Mandatory
<data|operations|{ncmpAction}>	request category - yang data, rpc operation or a (non-modelled) ncmp api action. this could be data, operations or ncmpAction (e.g. 'sync-data')	Mandatory
ds/{datastore}		Mandatory
<resourceIdentifier>	the path expression identifying the resource that is being accessed by the operation. If this field is not present, then the target resource is the API itself.	Optional
<options>	Parameters with the familiar form of "name=value" pairs. Query parameters are optional to implement by the server and optional to use by the client. Each optional query parameter is identified by a URI	Optional DMI should be able to support (/pass through) ANY parameter associated with the RESTCONF message; see Section 3.4 of [RFC3986].

Datastores

New datastores are defined for ncmp to access the CPS 'running' or 'operational' datastore.
Alternatively, the request can be sent directly to the 'device' itself (bypassing CPS datastores) using one of the 'passthrough-*' datastores options as below

The new ncmp datastores required for ONAP Release I include :

CPS-333 Network Configuration Management (NCMP) scope for I release considerations

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

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 Forward request to plugin 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 Forward request to plugin Output received response	DMI-Plugin	config-only	
8	On	ncmp/operational	cpsPath	application/yang-data+json	• Not supported in Istanbul releases. • *Considered for Kohn Release*	Read from cache 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	• Not supported in Istanbul releases. • **Planned for Kohn Release**	Read from cache output: application/json	CPS-Core	config + non-config	Output will use DataNode wrapping (as is from CPS-Core) 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 Convert cpsPath to RESTConfPath* Forward request to plugin | Read from DMI plugin 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 Convert cpsPath to RESTConfPath* Forward request to plugin | Read from DMI plugin 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 Forward request to plugin Output received response (success or failure)	The DMI plugin may error if the RP or content type are not supported. 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 Forward request to plugin Output received response (success or failure)	The DMI plugin may error if the RP or content type are not supported. 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 Forward request to plugin Output received response (success or failure)	The DMI plugin may error if the RP or content type are not supported. 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 Forward request to plugin Output received response (success or failure)	The DMI plugin may error if the RP or content type are not supported. The DMI plugin may forward the request without processing too.
5	On | Off	Create	ncmp/running	cpsPath	application/yang-data+json	Resolve DMI plugin Convert cpsPath to RESTConfPath Forward request to plugin Output received response (success or failure)	
6	On | Off	Update	ncmp/running	cpsPath	application/yang-data+json	Resolve DMI plugin Convert cpsPath to RESTConfPath Forward request to plugin Output received response (success or failure)	
7	On | Off	Delete	ncmp/running	cpsPath	N/A	Resolve DMI plugin Convert cpsPath to RESTConfPath Forward request to plugin Output received response (success or failure)	
8	On | Off	Patch	ncmp/running	cpsPath	application/**yang-data**+json (*plain patch)	Resolve DMI plugin Convert cpsPath to RESTConfPath Forward request to plugin Output received response (success or failure)	
9	On | Off	Patch	ncmp/running	cpsPath	application/**yang-patch**+json	Resolve DMI plugin Convert cpsPath to RESTConfPath Forward request to plugin Output received response (success or failure)	

Write Example

Write Example

Sync & Model API

- Below table shows the proposed interface, actual implementation might deviate from this but can be accessed from
  - Gerrit Source
  - Read-the-docs: https://docs.onap.org/projects/onap-cps/en/latest/design.html#offered-apis

#	Req/usecase	REST Method	URI	Request/Response Example

1	DMI notifies NCMP of new , deleted or changed cmhandles DMI Plugin NCMP. Including initial registration	POST	{ncmpRoot}/ncmp/v1/ch/	**Scenario** : DMI notifies NCMP of new cmhandles
				Method : POST
				URI : {ncmpRoot}/ncmp/v1/ch/
				Header :
				    Content-Type: application/json
				Request Body
				Request Body : {
				      "dmiPlugin" : "onap.dmi.plugin",
				      "createdCmHandles" : [{
				"cmHandle" : "rf4er5454",
				"cmHandleProperties" :
				{
				"subSystemId" : "system-001" }
				}, {..}],
				      "updatedCmHandles" : [..],
				      "removedCmHandles" : ["node-1",
				"node-2" , ...]
				  }
				json attributes:
				• "dmiPlugin" resolvable servicename
				• "createdCmHandles" used for initial cm handle registrations or subsequent
				cmhandle creations
				• "updatedCmHandles"
				Used for updates to cmhandles. Same structure as for create handles
				• "removedCmHandles" array of cmhandles that have been deleted
				from the network (no additional properties

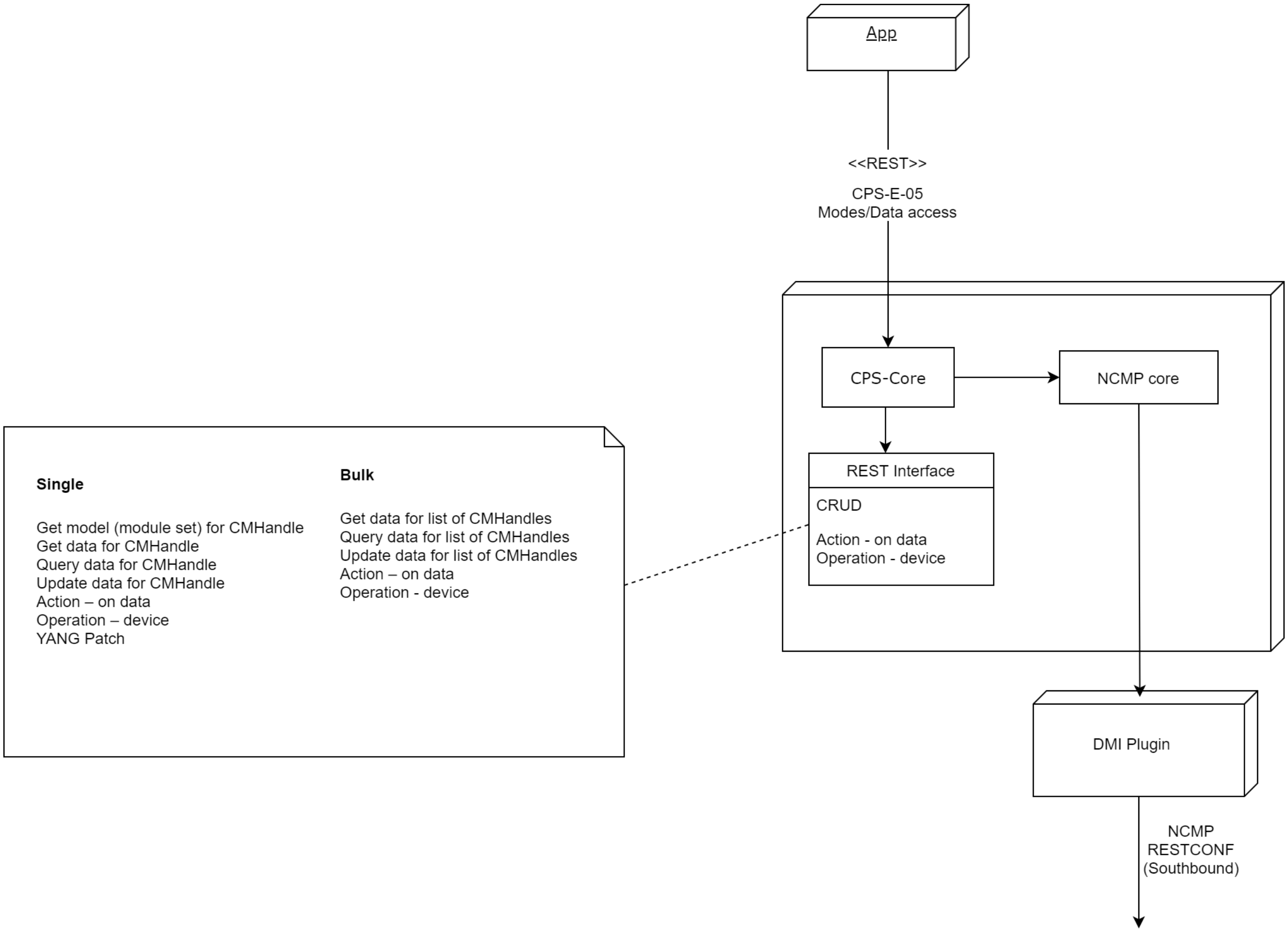
2	Get all cm handles that support all modules in a given list of modules	POST	{ncmpRoot}/ncmp/v1/ch/searches	**URI** : {ncmpRoot}/ncmp/v1/ch/searches
				The minimal requirement is if we provide the AND query impl then for OR query the client can send multiple requests
				Request Body
				Content: application/json
				Note: revision should be optional
				{
				  "modules": [
				    {
				      "moduleName": "", (Mandatory)
				      "revision": "" (Optional)
				    }
				  ]
				}
				Header :
				Accept: application/json
				Response:
				Should return an array of objects as we may add more data in the future
				{
				  "cmHandles": [
				   {
				     "cmHandleId": "xxx"
				   }
				]
				}

3	Request (trigger) Data Sync	POST	{ncmpRoot}/ncmp/v1/ch/<cmHandle>/syncData	**Scenario :** Client requests to sync a node **URI** : {ncmpRoot}/ncmp/v1/ch/node123/syncData **Response** : HTTP-Status code (only, no body)

4	Get model info for CMHandle	GET	{ncmpRoot}/ncmp/v1/ch/
			{cmHandle}/modules
				Scenario : Get the model data for CMHandle
				
				URI :{ncmpRoot}/ncmp/v1/ch/2334dedf/modules
				
				Header :
				Accept: application/json
				
				Response:
				
				[
				{
				"moduleName": "nc-notifications",
				"revision": "2008-07-14",
				},
				{
				"moduleName": "ietf-tls-server",
				"revision": "2016-11-02",
				},
				{
				"moduleName": "ietf-ssh-server",
				"revision": "2016-11-02",
				}
				]

5	Get all the registered cmhandles for a given plugin	GET	{ncmpRoot}/ncmp/v1
			/dmiPlugins/{pluginId}/ch
				Scenario : Get all cmhandles from NCMP for a given dmiPlugin. May be used
				for conciliation
				Method : GET
				URI : {ncmpRoot}/ncmp/v1/dmiPlugins/{dmiPlugin}/ch
				Header :
				Content-Type: application/json
				
				Response Body
				
				Success Response :
				HTTP/1.1 200 Ok
				Date: Thu, 26 Jan 2021 20:56:30 GMT
				Server: example-server
				{ "cmHandles" : [{
				"cmHandle" : "node-1",
				"cmHandleProperties " : {
				"subSystem" : "system-001" }
				}]
				}

NCMP / DMI Overview



Single

Get model (module set) for CMHandle
Get data for CMHandle
Query data for CMHandle
Update data for CMHandle
Action – on data
Operation – device
YANG Patch

Bulk

Get data for list of CMHandles
Query data for list of CMHandles
Update data for list of CMHandles
Action – on data
Operation - device