Regarding the general behavior of CSP.LMC on Dishes ID we had many interchanges and two meetings.

From the two meetings some questions were reported but unfortunately, from our point of view, no clear answers were provided regarding:

a) how and who provides the map of dishID/VCCId to the Mid CBF.LMC. In our understanding this information should flow from TM to Mid CBF via the CSP.LMC.

b) where the map of dishID/VCCId is produced and maintained.

However we think the feature SP-3233 is completed according the description and the acceptance criteria. Of course there is ample space to improve and to extend, and we will be glad to implement it in due time.

We agree on the fact that some questions are still open. In the following we try to answer to your points and clarify the CSP.LMC behavior, from our point of view.

1. At the moment, the CSP.LMC has no information about the list of deployed dish IDs as well as the number of receptors, at deployment time.

   In one of our private discussion (03 April) you pointed out that CSP.LMC does not need to know which dishes are deployed, only which VCCs are deployed and the DISH ID for the dish that should be connected to the VCC. But there might be a a situation where more VCCs than DISHES are deployed, and a VCC does not receive input from a dish. Mid CBF.LMC should report that. How this will be done, is still open.

2. The CSP.LMC should release all the receptors (to have a clean start) and then receive a AssignResources with the new set of receptors. At the moment, the CSP.LMC only propagates this information to Mid CBF: it was decided that the CSP.LMC only validates this list against the ADR-32 dish format. All the deeper checks (for example VCC available, VCC-receptor connection status, VCC already assigned etc.) are demanded to the Mid CBF.LMC.

   The CSP.LMC extended check functionalities in this area, have been disabled as agreed.

3. CSP.LMC reports in its Capabilities all the relevant information provided by Mid CBF, including the DishId associated to each VCC.

Error handling

The behavior of CSP.LMC error handling is the following.

The JSON string received with the AssignResources command is validated against the telescope model. If this string is not validated, the CSP.LMC Subarray throws an exception.

After the JSON string validation, the CSP.LMC Subarray transitions to RESOURCING.

It the list specifies some invalid dish IDs:

- for each invalid dish id, a warning message (with the dish id name) is logged

- the invalid dish is removed from the list (i.e. it is not propagated to Mid CBF)
- the remain valid dishes are propagated to Mid CBF
- the CSP.LMC TANGO attribute assignedReceptors reports the receptors really assigned to Mid CBF subarray.
- If the command completes with success on Mid CBF Subarray, the CSP.LMC reports this command as TaskSTATUS=COMPLETED, ResultCode=OK and the observing state transitions to IDLE.

Below, we report the logs and the output of the command AssignResources when a set of valid and invalid dish IDs is provided with the command and the CSP Subarray initial obsState is EMPTY.


```
In [23]: json_string = '{"subarray_id": 1,"dish":{ "receptor_ids":["SKA001", "SKA500", "MKT003"]}}'

In [24]: result = csp_sub1.assignresources(json_string)

In [25]: csp_sub1.longrunningcommandstatus
Out[25]: ('1684926530.291849_169668790121707_AssignResources', 'COMPLETED')

In [26]: csp_sub1.longrunningcommandresult
Out[26]:
('1684926530.291849_169668790121707_AssignResources',
 '[0, "assign completed  1/1"]')

In [27]: csp_sub1.obsstate
Out[27]: <obsState.IDLE: 2>

In [28]: result
Out[28]:
[array([2], dtype=int32),
 ['1684926530.291849_169668790121707_AssignResources']]

In [29]: csp_sub1.assignedreceptors
Out[29]: ('MKT003',)

In [30]:
```

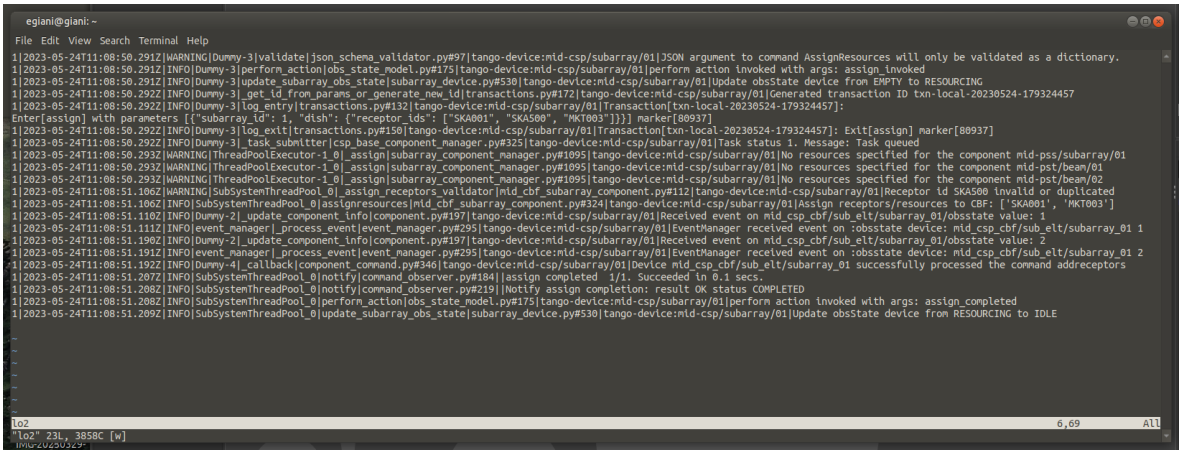
Note: SKA001 not assigned because not accepted by Mid CBF

In the case in which the list contains only invalid dish IDs:

- for each invalid dish id, a warning message (with the dish id name) is logged

- the invalid dish is removed from the list (i.e. it is not propagated to Mid CBF)
- since the list is empty the command is not propagated to Mid CBF Subarray
- the CSP.LMC reports to TaskStatus=COMPLETED and ResultCode=FAILED and the observing state transitions to EMPTY or IDLE depending on the initial state of the subarray.

Below, we report the logs and the output of the command AssignResources when a set of invalid dish IDs is provided with the command and the CSP Subarray initial obsState is EMPTY.



Note: ResultCode 3 = FAILED

Question 1

What if AssignResources specifies an invalid DISH ID (example: SKA456) ? How does CSP.LMC handle this?

a) Does CSP.LMC throw an exception, generate a log and/or report that the command is rejected?

Answer: CSP.LMC generates a log for each invalid dish ID. The command is not rejected: it is processed and returns COMPLETED with ResultCode OK or FAILED depending on the success of the operation on Mid CBF.

b) Does CSP.LMC allocate other dishes successfully (i.e. does it proceed with the command) or does it stop command execution. Does CSP.LMC return to EMPTY, stay in RESOURCING or transition to IDLE (if other dishes are assigned)?

Answer: If at least one dish ID is valid, the command is propagated to Mid CBF subarray and CSP.LMC transitions to IDLE when the other dishes are successfully assigned to Mid CBF subarray.

The CSP.LMC subarray transitions to its initial obsState (EMPTY or IDLE) when no receptor is assigned to the Mid CBF subarray.

c) When is this checked, immediately when the message is received or only after the command is removed from the input queue?

Answer:The check on the dish ID is done after the command is removed from the queue (when the CSP.LMC subarray obsState is RESOURCING). This means that the status of the command as well as its result code, are updated via the long running command (LRC) attributes events.

The only check that is done immediately (before the transition to RESOURCING) is the validation of the JSON string against the Telescope Model.

Question 2

What if AssignResources command specifies the DISH which has not been deployed (example: Dishes SKA001, SKA0064, SKA123, SKA133 have been deployed and the user wants to use SKA067). Please consider the same questions as for 1).

Answer:

For the moment the CSP.LMC is not in charge of performing this kind of checks (validation of dish id against the list of deployed dishes). As stated before, if the dish ID is ADR-32 compliant, the receptor name is sent to Mid CBF.

In our understanding, these checks are demanded to Mid CBF. On top of this CSP.LMC has no information about deployed dishes.

Question 3

What if Mid.CBF reports an invalid or unexpected DISH ID ? Does CSP.LMC check that the DISH IDs reported by Mid.CBF match what is expected?

Answer: At the moment CSP.LMC does not perform this check. CSP.LMC only monitors the information provided by Mid CBF on the VCCs (including the receptor id associated) via the VCC capability device.

Question 4

Does CSP.LMC collect information regarding the mapping between the VCC ID and DISH ID? If so, when does that occur? Does CSP.LMC store that information.

Answer: Right now this mapping is not provided to the CSP.LMC in advance. The information about VCCs are maintained into the VCC Capability. This device implements a TANGO attribute (vccJson) reporting also the dish Id associated to each VCC.

The mapping is obtained after successful connection to Mid CBF sub-system.

Question 5

Information related to DISH ID handling should be part of the CSP.LMC documentation. If this is already documented please provide the reference.

Answer: There is no documentation. We can use the present one and publish it in this Confluence page

(https://confluence.skatelescope.org/display/SE/CSP+LMC+Dish+ID+handling)

Question 6

TMC team reported that integration with CSP.LMC and DISH did not occur since the sub-systems are not ready. Is CSP.LMC ready for integration with TMC?

Answer: The CSP.LMC is ready to integrate with TMC because it has implemented the support for the ADR-32 in the way described above. CSP.LMC accepts all dishes (SKA and MKT) whose IDs are ADR-32 compliant but right now, Mid CBF considers invalid dish IDs different from [MKT000-MKT003].