**3GPP TSG-SA5 Meeting #155 *S5-*** ***242985d1***

Jeju, Korea (Republic Of), 27th May 2024 - 31st May 2024

**Source: Ericsson Hungary Ltd.**

**Title: Rel-19 pCR TR 28.872 Planned configuration requirements and updates**

**Document for: Approval**

**Agenda Item: 6.19.9**

# 1 Decision/action requested

***Approve the proposal.***

# 2 References

[1] 3GPP TS 28.872: "Study on Management of planned configurations" v0.1.0

# 3 Rationale

Some additional requirements and a terminology change are proposed for planned configurations.

The term for planned configuration should be re-considered. Immediate activation of configuration changes is a requirement. In case of immediate activation, the set of changes are sent to the producer, activated immediately and then removed after activation. In this case the change-set is not necessarily stored in any persistent way. Calling something a "plan" that exists only in an ephemeral form would be misleading.

It is proposed to use the term "cmwritejob" when the changes are activated immediately and use the term "planned configuration" when the activation is on a separate request.

# 4 Detailed proposal

**First change**

## 5.1 Use case #1: Managing cmwritejobs and planned configurations

### 5.1.1 Description

The information architecture specified in SA5 knows only one type of data node tree on a MnS producer. This data node tree includes configuration data nodes and state data nodes. When writing to configuration data nodes on a MnS producer it is normally expected that the new values are applied to the managed system. The data node tree represents hence the best information that the management system has about the current configuration really running in the managed system and the state of the managed system.

Many management problems would benefit from the possibility of creating planned configurations that are not active yet and that can be manipulated without changing the current configuration of the managed system. When ready, the planned configurations can be activated.

Planned configurations do not contain any state data, only configuration data.

In many cases it is needed to collect a large set of required changes/updates to the configuration and activate these immediately as a single cmwritejob. The immediate activation of a cmwritejob offers an efficient means of applying large volume changes directly to the network in an asynchronous way without having to go through an unnecessary multi-step process. This also provides the benefit of handling the changes as a "set" where result and progress can be provided both for the cmwritejob and also for the individual changes. This also allows the efficient handling of big configuration changes that might take a longer time to activate.

The mechanisms for planned configuration and cmwritejobs are mostly common.

1) sending a set of changes to the producer

2) activating the changes

3) retrieving progress and result information

A planned configuration is an extension of a cmwritejob, adding the capabilities for storing and updating the set of changes, and allowing a separate activation step. The difference between a cmwritejob and a planned configuration is that for the cmwritejob activation is immediate while a planned configuration is activated with a separate request. This also means that planned configurations may be edited by the consumer before activation.

Beside creating the set of changes requested, the cmwritejob/planned configuration has other data associated with it:

1) Set of changes requested

2) Metadata about the cmwritejob/planned configuration (name, description, activation-mode, etc.)

3) Status Information

4) Progress information

5) Result information

### 5.1.2 Potential requirements

Req-1: The 3GPP management system should support creating planned configurations/cmwritejobs for managed systems. The 3GPP management system should support reading, updating, and deleting planned configurations for managed systems. Support for bot cmwritejobs and planned configurations is optional, but at least one of them should be supported.

Req-2: The 3GPP management system should support creating multiple planned configurations/cmwritejobs for managed systems. The 3GPP management system should support reading, updating, and deleting multiple alternative planned configurations for managed systems.

Req-3: The 3GPP management system shall support creating, reading metadata (data about the set of changes) for cmwritejobs and planned configurations. The 3GPP management system shall support modifying, deleting metadata for planned configurations. Potential examples of metadata include plan-name, userLabel, revision-information/date, activation mode, operationId for the individual changes.

### 5.1.3 Potential solutions

#### 5.1.3.1 Potential solution #1: NRM and CRUD operations

A new capability is introduced that allows to create, update, and delete cmwritejobs and planned configurations. Planned configurations can be manipulated without impacting the current configuration of the managed system. They contain only information relating to configuration data nodes. Information relating to state data nodes is omitted.

A planned configuration/cmwritejob may include only the configuration for the part of the managed system that shall be reconfigured. For example, a planned configuration/cmwritejob may contain the configuration of a complete BTS or only the configuration of a cell supported by a BTS.

A planned configuration/cmwritejob is represented by a data node tree. The data node tree is compliant to the NRM schema that describes also the data node tree of the corresponding current configuration. The values in the data node tree are those values with which the managed system shall be reconfigured.

Note that NRM schemas currently specified in 3GPP SA5 are OpenAPI definitions and YANG definitions.

The normal CRUD operations specified in TS 28.532 [2] are used for manipulating the data node trees of planned configurations. Note: For further study as this depends on the format of the planned configuration/cmwritejob.

Multiple alternative configurations/cmwritejobs may be instantiated.

*Editor's note: It is ffs how to combine the current configuration with related planned configurations in a common tree.*

## 5.a Use case #2: Activating cmwritejobs, planned configurations

### 5.a.1 Description

After a cmwritejob/planned configuration is created (and potentially updated) it needs to be activated: that is the configuration changes included in the cmwritejob/planned configuration need to be pushed into the addressed managed entities. There is a need for different ways of executing this activation.

### 5.a.2 Potential requirements

Req-act-1: The MnS producer shall make it possible for the consumer to **activate** a cmwritejob/planned configuration: that is execute the changes included in the planned configuration in the addressed managed entities.

Req-act-2: The MnS producer shall **check** the consistency, syntax and semantic of the requested changes before activation to ensure that the configuration changes can be implemented in the network. Failed checks shall block activation. The producer shall provide the results of failed checks.

Req-act-3: The MnS producer shall make it possible for the consumer to **activate** a configuration **immediately upon creation (cmwritejob) or in a separate request (planned configuration)**. Both activation modes (immediate and separate request) shall be optional, but at least one of them shall be supported by the MnS producer. If both modes are supported, the consumer shall be able to select the mode of activation.

Req-act-4: The MnS producer shall make it possible for the consumer to activate the requested changes either using an **all-or-nothing or a best-effort** mechanism. Both activation modes (all-or-nothing and best-effort) shall be optional, but at least one of them shall be supported by the MnS producer. If both mechanisms are supported, the consumer shall be able to select the mechanism for activation.

Req-act-5: The MnS producer shall make it possible for the consumer to activate the requested changes either using a **minimum-time or a minimum-impact** impact-mode. Both impact-modes (minimum-time and minimum-impact) shall be optional, but at least one of them shall be supported by the MnS producer. If both impact-modes are supported, the consumer shall be able to select the impact-modes for activation. Using the minimum-impact mode the activation should be executed in a manner that results in the least network impact, network disturbance e.g. node-by-node activation. Using the minimum-time mode the activation should be executed in the shortest amount of time e.g. activation on all nodes at once. The exact method to implement the impact-modes is vendor specific.

Req-act-6: The MnS producer shall **provide progress information** about the activation.

Req-act-7: The MnS producer shall make it possible for the consumer to **read progress information** about the activation.

Req-act-8: The MnS producer shall support **notifications** to indicate **progress and completion** of an activation.

Req-act-9: The MnS producer shall provide **result information** about the activation. The result shall be provided both for the activation as a whole and for the individual included changes.

Req-act-10: The MnS producer shall make it possible for the consumer to **cancel** an activation. Result information according to Req-act-9 should be provided after the cancellation.

### 5.a.3 Potential solutions

#### 5.a.3.1 Possible Configuration Write Job States and State Transitions

A cmwritejob/planned configuration may have a number of different states depending on what phase it is at.

**1. Draft**

* Description: Initial state when the job is created but not yet finalized. The cmwritejob may be edited when in this state.
* Transitions: Can move to **Validated**, **Canceled**, or **Scheduled**.

**2. Validated**

* Description: The job configuration has been validated successfully.
* Transitions: Can move to **Scheduled**, **Activation Pending**, or **Canceled**.

**3. Validation Failed**

* Description: The job configuration failed validation.
* Transitions: Can move back to **Draft** for corrections or to **Canceled**.

**4. Scheduled**

* Description: The job is scheduled to be executed at a future time.
* Transitions: Can move to **Activation Pending** when the scheduled time arrives or to **Canceled**.

**5. Activating**

* Description: The job is in the process of being activated on the network.
* Transitions: Can move to **Activated**, **Activation Failed**, or **Canceled**.

**6. Activated**

* Description: The job has been successfully activated on the network.
* Transitions: This is typically a terminal state but can be reverted to **Draft** for modifications in some cases.

**7. Activation Failed**

* Description: The job failed to activate on the network.
* Transitions: Can move back to **Draft** for corrections or to **Canceled**.

**8. Executing**

* Description: A Scheduled periodic job is executing.
* Transitions: Can move to **Executed, Execution Failed or Canceled**

**9. Executed**

* Description: A running/executing Scheduled periodic job has completed execution.
* Transitions: Can move back to **Scheduled or Completed if the Scheduler has expired**

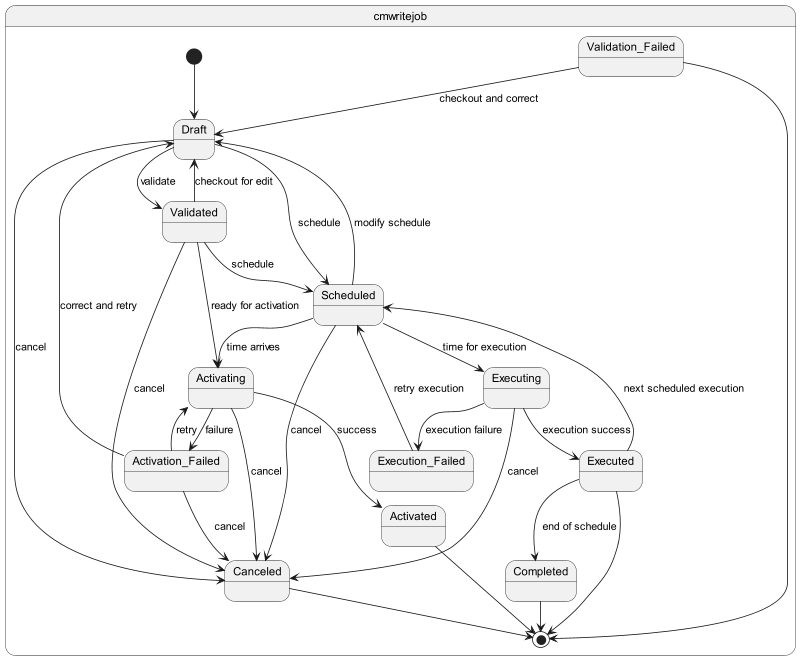
**9. Execution Failed**

* Description: A Scheduled periodic job has failed to execute successfully.
* Transitions: Can move back to **Scheduled** or may be **Canceled**

**10. Canceled**

* Description: The job has been canceled and will not proceed further.
* Transitions: This is a terminal state.

The exact states are FFS.



#### 5.a.3.1 Possible methods for handling cmwritejobs/planned configurations

The consumer can handle plans/CmwriteJobs in multiple steps. Edit/write, validate, activate, schedule, cancel. Sometimes multiple steps are required e.g. for immediate activation cmwritejob needs to be written, validated and activated. For scheduling a plan the plan should be written, validated and scheduled. These steps can be excuted in 2 ways:

1) Specifying in the POST or PATCH message a

“targetState” : “ACTIVATED”

2) Sending multiple POST messages for each individual step, write, validate, schedule

### 5.a.4 Evaluation of potential solutions

TBD

**End of change**