

Health-Check Use Case

Introduction

The Health-check use case is being defined to ensure that O-RAN components are monitored, and their health being reported properly.

Background and Goal of the Use Case

As the O-RAN Alliance continues to advance the Open RAN architecture, it is important to define use cases that will focus implementations toward operational-readiness. This proposed Health-check use case is to enable implementations to build the initial set of capabilities for monitoring and reporting of the health of the Open RAN, consisting of the following elements:

- RIC and its common platform functions
- External RIC interfaces (O1, A1, E2),
- xAPPs deployed on RIC,
- RAN Managed Functions (i.e., O-CU[1], O-DU, O-RU).

The key goal is that the O-RAN elements can do the following:

- Perform self-health-checks
- Support on-demand health-checks and queries
- Report alarms, alerts and other data based on health-check results

The implementation of this use case will enable northbound clients – particularly SMO (Service Management and Orchestration), but also other platforms as well – to trigger Health-check requests and queries. The use case defines 4 separate flows to support these validations, which are explained in more detail in subsequent sections.

In addition, the use case documentation provides a mapping of the use case requirements to the EPICs defined for O-RAN Software Community (O-RAN SC) Bronze Release. The list below enumerates the currently-defined EPICs in the Bronze Release that pertain to the Health-Check use case per the [latest Bronze release planning excel](#) (also see Table 1 below):

- RIC dashboard shall be able to retrieve any defined alarms
- Support Health-Check Telemetry (FM, Heartbeat, PM)
- Non-RT RIC/SDN-R support for A1 messages
- Support xAPP Health-Check
- Provide E2E Health-check Test
- Support E2 Test Message Processing
- Support O1 Health-check Provisioning Command
- Support A1 Policy Test Message Generation
- Support A1 Policy Test Message Mediation
- Support A1 Policy Test Message Processing

JIRA-side issues mapped to healthcheck use case in near-RT RIC for Bronze: [filter](#)

[1] O-CU can be further subcategorized into O-CU-UP and O-CU-CP for specific health-checks.

Four End-To-End (E2E) flows are proposed to be implemented which will validate the health of the RIC, xAPPs, O-CU/O-DU/O-RU and the relevant interfaces among them:

- **Flow #1 – RIC Self-Health-Check** – RIC should conduct its own self-check, report the results and send out notifications on any alarm and alert conditions as determined by the health-check results. The self-check needs to cover fault management (FM), performance management (PM) and interface heart beats, all of which should be consistent with (per 3GPP 28.545 and 28.550). The flow covers the following scope:
 - O1 and A1 interface exposed by the RIC to NB clients in the form of heartbeats
 - RIC common functions
 - xAPP instances hosted on the RIC
 - E2 interface southbound to RAN resources
- **Flow #2 – O1 RIC Heartbeat, Health Retrieval and On-Demand Health-Check** – This Health-Check flow is similar to Flow #1's Self-Check, but it's triggered by NB clients to retrieve the latest RIC health conditions and/or initiate an on-demand health-check. The flow covers the following scope:
 - O1 interface Heartbeat check by NB clients
 - RIC common platform functions
 - xAPP instances hosted on the RIC

- E2 interface southbound to RAN resources
- **Flow #3 – A1 RIC Heartbeat and Policy Health-Check** – This Health-Check flow covers the following scope:
 - A1 interface Heartbeat Health-check by NB clients
 - Policy Status – A1 Retrieval of current active policies on the RIC
 - Policy Add and Delete – exercising of A1 interface and RIC Processing of a test policy creation and deletion
- **Flow #4 – O1 Managed Function Health-Check** – This Health-Check flow covers the following scope:
 - O1 interface to Managed Functions (i.e., O-CU, O-DU, and/or O-RU)
 - O-CU/O-DU/O-RU health

The implementation should focus on SMO being the NB client that triggers these Health-check flows, even though other external management platforms such as ONAP could also trigger the requests.

See O-RAN architecture page on <https://wiki.o-ran-sc.org/display/OAM/OAM+Architecture> for more information on RIC, O-CU/O-DU/O-RU and associated management interfaces.

Definitions and Assumptions

While Health-Checks are commonly implemented in many software systems, it is important to provide definition of Health-Check related terminology being used in the Use Case documentation.

- **Health-Checks** – Health-Checks are used to monitor and report the health of a system, including its interfaces. In the O-RAN context, a system can be the RIC, O-CU, O-DU or O-RU.
 - Each system is required to *self-monitor* its own health at configurable intervals, and results stored for status queries. A system with multiple subsystems or modules is required to have every subsystem self-monitored.
 - Each system is required to support *on-demand* Health-Checks requested by a client. A client can request for the latest stored/cached health results and/or an instantaneous health-check to be run.
 - Health-Checks can consist of simple *heartbeats* or more comprehensive
- **Heartbeats** – a heartbeat is just a simple message or signal sent by the system over a specific interface (e.g., heartbeats for O1, heartbeats for A1) to indicate that it is still alive or present. Subscribed clients will then know that the system is still available for communication over that interface. Heartbeats are an elementary way to determine whether an interface is up, but more checks are needed to ensure full operability.
- **Diagnostics** – A health-check request can specify a diagnostic to be run. Diagnostics are any sequence of checks that determine the overall health of a system (e.g., overall health-check of RIC). In most cases, a system running diagnostics will trigger multiple health-checks against all /most of its subsystem/modules (e.g., RIC platform modules, xAPPs, etc.). Any failures and anomalies identified will be mapped against pre-defined severity levels by the system, resulting in alarms (major severity) and alerts (minor severity) being declared and notifications being emitted. Similarly, any alarm/alert needs to be cleared and notified when the condition is cleared. The collective assessment of a system's failures/anomalies also determines the system's operability – for instance, whether the system is green (no conditions), yellow (minor anomalies), or red (major failures).
- **Multi-Hop Health-Checks within a System** – Health-Checks in general can be one-hop or multiple hop requests – with multiple hops involving the request recipient to invoke additional Health-Check requests downstream. The scenarios described above on health of a system are multi-hops (e.g., diagnostics), as individual components of that system have to be health-checked. However, the scope of this Use Case limits the Health-Check request to within a system (multi-hop to check components of that system) and not across systems (multi-hop to check multiple systems on one request is not in scope). Specifically, individual Health-Check requests should be defined for RIC, and for O-CU/O-DU/O-RU.
- **Managed Function Health-Check** – The Managed Function in this document refers to O-CU, O-DU and/or O-RU. Flow #4 addresses monitoring and reporting of the health of the RAN Managed Functions in general, and applies to O-CU, O-DU and O-RU. There are no separate flows defined for each.
- **SMO and Non-Real Time RIC** – Health-Checks related to SMO and Non-RT RIC are out of scope for this use case