3GPP TR 28.890 V16.0.0 (2019-03)

Technical Report

3rd Generation Partnership Project;

Technical Specification Group Services and System Aspects;

Management and orchestration;

Study on integration of Open Network Automation Platform (ONAP) and 3GPP management for 5G networks

(Release 16)

** 

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Report is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

ONAP, management services

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2019, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword [5](#__RefHeading___Toc2422188)

1 Scope [6](#__RefHeading___Toc2422189)

2 References [6](#__RefHeading___Toc2422190)

3 Definitions and abbreviations [8](#__RefHeading___Toc2422191)

3.1 Definitions [8](#__RefHeading___Toc2422192)

3.2 Abbreviations [8](#__RefHeading___Toc2422193)

4 Introduction [8](#__RefHeading___Toc2422194)

4.1 Architecture overview [8](#__RefHeading___Toc2422195)

4.1.1 3GPP 5G management framework [8](#__RefHeading___Toc2422196)

4.1.2 ONAP Architecture overview [9](#__RefHeading___Toc2422197)

5 Integration of ONAP/DCAE with 3GPP management services for FM and PM purposes [9](#__RefHeading___Toc2422198)

5.1 Positioning of ONAP/DCAE wrt. 3GPP management architecture framework [9](#__RefHeading___Toc2422199)

5.1.1 ONAP/DCAE architecture [9](#__RefHeading___Toc2422200)

5.1.2 Positioning [11](#__RefHeading___Toc2422201)

5.1.2.0 General [11](#__RefHeading___Toc2422202)

5.1.2.1 An example of ONAP DCAE Collection Framework utilizing the management services provided by 3GPP Management Services Provider [11](#__RefHeading___Toc2422203)

5.1.2.2 An example of 3GPP MnF using information provided by ONAP DCAE [11](#__RefHeading___Toc2422204)

5.2 Comparative analysis [12](#__RefHeading___Toc2422205)

5.2.1 Communication principles [12](#__RefHeading___Toc2422206)

5.2.1.1 ONAP DCAE [12](#__RefHeading___Toc2422207)

5.2.1.2 3GPP 5G management framework [15](#__RefHeading___Toc2422208)

5.2.1.3 Comparison [15](#__RefHeading___Toc2422209)

5.2.1.3.1 Events headers / notification parameters [15](#__RefHeading___Toc2422210)

5.2.1.3.2 FM/PM data collection [17](#__RefHeading___Toc2422211)

5.2.1.4 Comparison of interface technologies [18](#__RefHeading___Toc2422212)

5.2.1.5 Heartbeat [18](#__RefHeading___Toc2422213)

5.2.1.5.1 ONAP [18](#__RefHeading___Toc2422214)

5.2.1.5.2 3GPP [19](#__RefHeading___Toc2422215)

5.2.1.5.3 Proposed way forward [19](#__RefHeading___Toc2422216)

5.2.2 Handling of alarm / event notifications [19](#__RefHeading___Toc2422217)

5.2.2.1 ONAP DCAE VES Fault Field [19](#__RefHeading___Toc2422218)

5.2.2.2 3GPP alarm supervision [20](#__RefHeading___Toc2422219)

5.2.2.3 Comparison [21](#__RefHeading___Toc2422220)

5.2.2.3.1 Fault fields [21](#__RefHeading___Toc2422221)

5.2.2.3.2 Mapping 3GPP alarm notifications to ONAP R3 VES JSON API [21](#__RefHeading___Toc2422222)

5.2.2.4 Alarm Life Cycle management in ONAP [25](#__RefHeading___Toc2422223)

5.2.3 Handling of performance data collection [26](#__RefHeading___Toc2422224)

5.2.3.1 ONAP DCAE [26](#__RefHeading___Toc2422225)

5.2.3.2 3GPP Performance Management [26](#__RefHeading___Toc2422226)

5.2.3.3 Comparison [27](#__RefHeading___Toc2422227)

5.2.3.3.1 PM job handling [27](#__RefHeading___Toc2422228)

5.2.3.3.2 FM/PM data reporting [27](#__RefHeading___Toc2422229)

5.2.3.3.3 Reporting 3GPP Rel-15 PM data to DCAE R3 VES JSON Collector [29](#__RefHeading___Toc2422230)

5.2.3.3.4 3GPP Fault Supervision operations [31](#__RefHeading___Toc2422231)

5.3 Conclusions on FM/PM [31](#__RefHeading___Toc2422232)

6 Integration of ONAP with 3GPP management services for configuration management purposes [32](#__RefHeading___Toc2422233)

6.1 ONAP controllers and 3GPP provisioning service [32](#__RefHeading___Toc2422234)

6.1.1 ONAP controllers [32](#__RefHeading___Toc2422235)

6.1.1.1 Introduction [32](#__RefHeading___Toc2422236)

6.1.1.2 APPC [33](#__RefHeading___Toc2422237)

6.1.1.2.1 Overview [33](#__RefHeading___Toc2422238)

6.1.1.2.2 APPC Netconf Adapter [34](#__RefHeading___Toc2422239)

6.1.1.2.3 APPC Chef Adapter [34](#__RefHeading___Toc2422240)

6.1.1.2.4 APPC Ansible Adapter [34](#__RefHeading___Toc2422241)

6.1.1.3 SDNC [35](#__RefHeading___Toc2422242)

6.1.1.4 VF-C [35](#__RefHeading___Toc2422243)

6.1.2 3GPP provisioning service [35](#__RefHeading___Toc2422244)

6.1.3 Positioning [39](#__RefHeading___Toc2422245)

6.1.3.1 Introduction [39](#__RefHeading___Toc2422246)

6.1.3.2 An example of ONAP controller utilizing management services provided by 3GPP Management Services Provider [39](#__RefHeading___Toc2422247)

6.1.3.3 An example of 3GPP MnF using interface provided by ONAP VF-C [39](#__RefHeading___Toc2422248)

6.2 Comparative analysis [40](#__RefHeading___Toc2422249)

6.2.1 Introduction [40](#__RefHeading___Toc2422250)

6.2.2 Management principles [40](#__RefHeading___Toc2422251)

6.2.3 Communication protocols [40](#__RefHeading___Toc2422252)

6.2.3.1 NETCONF [40](#__RefHeading___Toc2422253)

6.2.3.2 Other protocol options [41](#__RefHeading___Toc2422254)

6.2.4 Handling of configuration management notifications [41](#__RefHeading___Toc2422255)

6.2.4.1 3GPP configuration management notifications [41](#__RefHeading___Toc2422256)

6.2.4.2 ONAP configuration notifications [41](#__RefHeading___Toc2422257)

6.2.4.3 Potential solution for enabling ONAP DCAE VES JSON Collector to consume 3GPP configuration management notifications [42](#__RefHeading___Toc2422258)

6.3 Conclusions on configuration management [43](#__RefHeading___Toc2422259)

Annex <A>: Change history [44](#__RefHeading___Toc2422260)

# Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

# 1 Scope

The present document studies compatibility of the ONAP management platform architecture and functionality with that of the 5G service-based management architecture being defined in 3GPP TS 28.53x, 28.54x and 28.55x series specifications. The present document is based on ONAP releases up to and including R3 ("Casablanca") and the corresponding APIs and related interfaces.

The scope of the present document includes:

- analysis and comparison of 3GPP and ONAP configuration management mechanisms.

- analysis of how ONAP configuration management mechanisms can be supported in 3GPP.

- analysis and comparison of 3GPP and ONAP alarm management and performance management mechanisms, concentrating on DCAE (Data Collection, Analytics and Events) / Collection Framework event stream and batch data collection on the ONAP side.

- study how notify mechanisms are handled by ONAP DCAE and 3GPP management services.

- study if any gaps exist between the semantics and format of data collected by DCAE and the semantics and format of data produced by both alarm supervision and performance management services.

- analysis of ONAP/3GPP protocol compatibility.

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 28.533: "Management and orchestration; Architecture framework".

[3] 3GPP TS 28.531: "Management and orchestration; Provisioning".

[4] 3GPP TS 28.541: "Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3".

[5] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: <https://wiki.onap.org/display/DW/Casablanca+Architecture> (v3.0.3).

[6] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: <https://onap.readthedocs.io/en/latest/submodules/dcaegen2.git/docs/sections/architecture.html>, accessed 10.10.2018

[7] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: ONAP Developer Wiki - <https://wiki.onap.org/> (v94), accessed 25.04.2018.

[8] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: <https://onap.readthedocs.io/en/latest/submodules/vfc/nfvo/lcm.git/docs/platform/architecture.html>, accessed 18.10.2018.

[9] 3GPP TS 28.550: "Management and orchestration; Performance assurance".

[10] 3GPP TS 28.532: "Management and orchestration; Generic management services".

[11] 3GPP TS 28.545: "Management and orchestration; Fault Supervision (FS)".

[12] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: [https://onap.readthedocs.io/en/latest/submodules/vnfsdk/model.git//docs/files/VESEventListener\_7\_0\_1.html?highlight=ves%20collector#datatype-commoneventheader](https://onap.readthedocs.io/en/latest/submodules/vnfsdk/model.git//docs/files/VESEventListener_7_0_1.html?highlight=ves collector" \l "datatype-commoneventheader), accessed 26.09.2018.

[13] Text Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the text: <https://onap.readthedocs.io/en/latest/submodules/vnfsdk/model.git/docs/files/VESEventListener_7_0_1.html>, accessed 10.10.2018.

[14] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: [http://onap.readthedocs.io/en/beijing/submodules/vnfrqts/requirements.git/docs/Chapter7.html#data-model-for-event-records](http://onap.readthedocs.io/en/beijing/submodules/vnfrqts/requirements.git/docs/Chapter7.html%23data-model-for-event-records), accessed 20.06.2018.

[15] 3GPP TS 28.552: "Management and orchestration; 5G performance measurements ".

[16] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: <https://onap.readthedocs.io/en/latest/submodules/vnfrqts/requirements.git/docs/Chapter7/Monitoring-And-Management.html>, accessed 19.10.2018.

[17] 3GPP TS 28.554: "Management and orchestration; 5G end to end Key Performance Indicators (KPI)".

[18] Table Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the table: <https://onap.readthedocs.io/en/latest/submodules/vnfsdk/model.git/docs/files/VESEventListener_7_0_1.html>, accessed 10.10.2018.

[19] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: <https://wiki.onap.org/display/DW/Controllers>, accessed 19.10.2018.

[20] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: [https://onap.readthedocs.io/en/latest/submodules/appc.git/docs/APPC%20User%20Guide/APPC%20User%20Guide.html](https://onap.readthedocs.io/en/latest/submodules/appc.git/docs/APPC User Guide/APPC User Guide.html), accessed 18.10.2018.

[21] Figure Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the figure: [https://onap.readthedocs.io/en/latest/submodules/appc/deployment.git/docs/APPC%20Ansible%20Adapter/APPC%20Ansible%20Adapter.html](https://onap.readthedocs.io/en/latest/submodules/appc/deployment.git/docs/APPC Ansible Adapter/APPC Ansible Adapter.html), accessed 18.10.2018.

[22] 3GPP TS 28.622: "Telecommunication management; Generic Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)".

[23] 3GPP TS 32.352: "Telecommunication management; Communication Surveillance (CS) Integration Reference Point (IRP); Information Service (IS)".

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

APPC Application Controller

CS Communication Surveillance

DCAE Data Collection, Analytics and Events

IRP Integration Reference Point

LCM Life Cycle Management

MD-SAL Model-Driven Service Abstraction Layer

MDAF Management Data Analytics Function

MDAS Management Data Analytics Service

MnF Management Function

MnS Management Service

NFV Network Function Virtualization

NFVO NFV Orchestrator

ONAP Open Network Automation Platform

PNF Physical Network Function

REST Representational State Transfer

VES VNF Event Stream

VF-C Virtual Function Controller

VNF Virtualized Network Function

VNFC VNF Component

xNF Network Function (x indicating either physical or virtual)

# 4 Introduction

## 4.1 Architecture overview

### 4.1.1 3GPP 5G management framework

The 3GPP 5G management framework (cf. TS 28.533 [2]) defines service-based management, where Management Services (MnSs) expose their capabilities to authorized consumers. MnSs are specified in:

- TS 28.531 [3] for provisioning,

- TS 28.545 [11] for fault supervision,

- TS 28.550 [9] for performance management,

- TS 28.532 [10] for generic management services.

### 4.1.2 ONAP Architecture overview

Figure 4.1.2.1 depicts the ONAP Release 3 overall architecture (see [5] - figure 1).

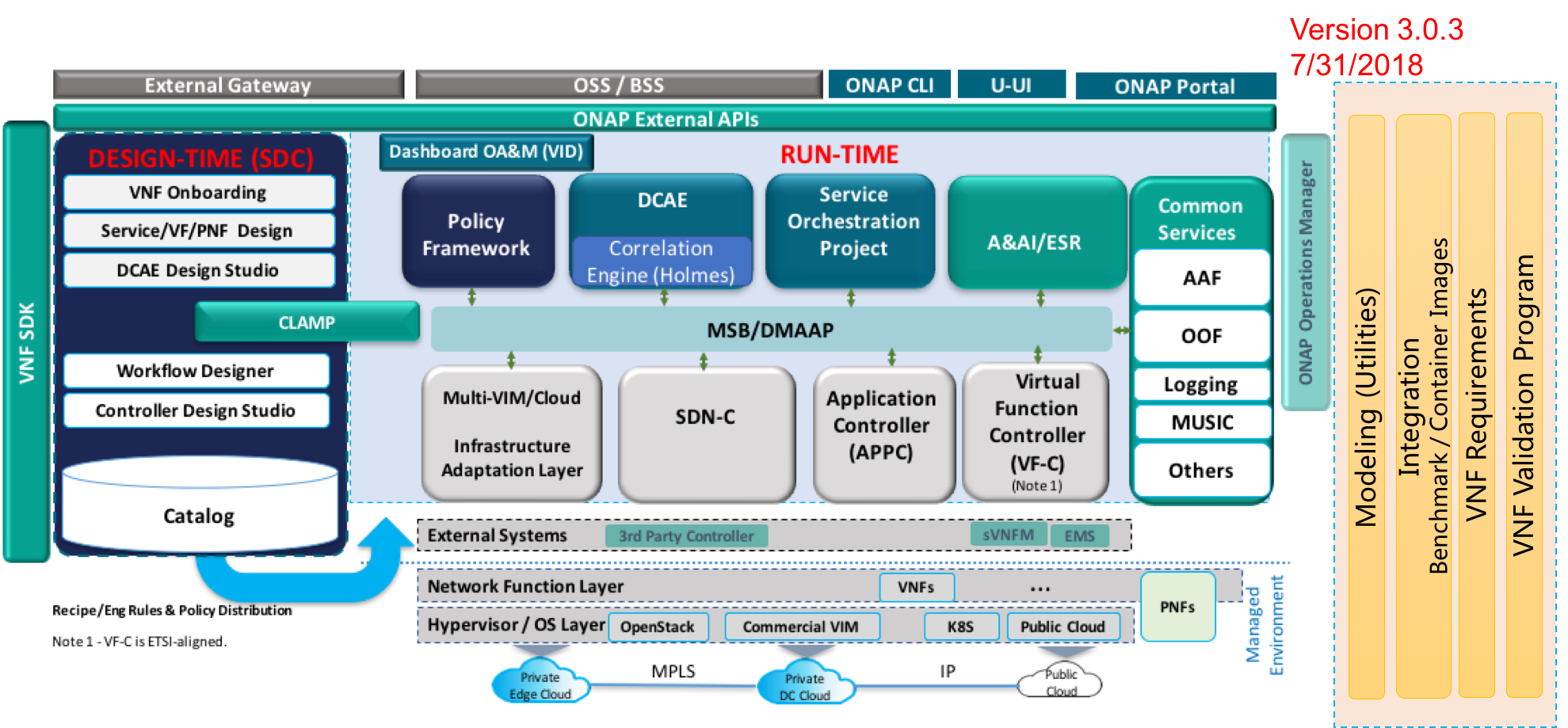


Figure 4.1.2.1: ONAP Release 3 overall architecture

NOTE: ONAP Release 3 overall architecture Figure attribution [5].

Within the ONAP architecture the DCAE (Data Collection, Analytics, and Events) platform collects, processes, and distributes performance, usage, configuration, and fault data from the managed environment, and is therefore of particular importance when comparing ONAP and 3GPP fault management and performance management functionality.

# 5 Integration of ONAP/DCAE with 3GPP management services for FM and PM purposes

## 5.1 Positioning of ONAP/DCAE wrt. 3GPP management architecture framework

### 5.1.1 ONAP/DCAE architecture

Figure 5.1.1.1 (see [6] - figure 1) provides a functional view of the DCAE platform architecture.

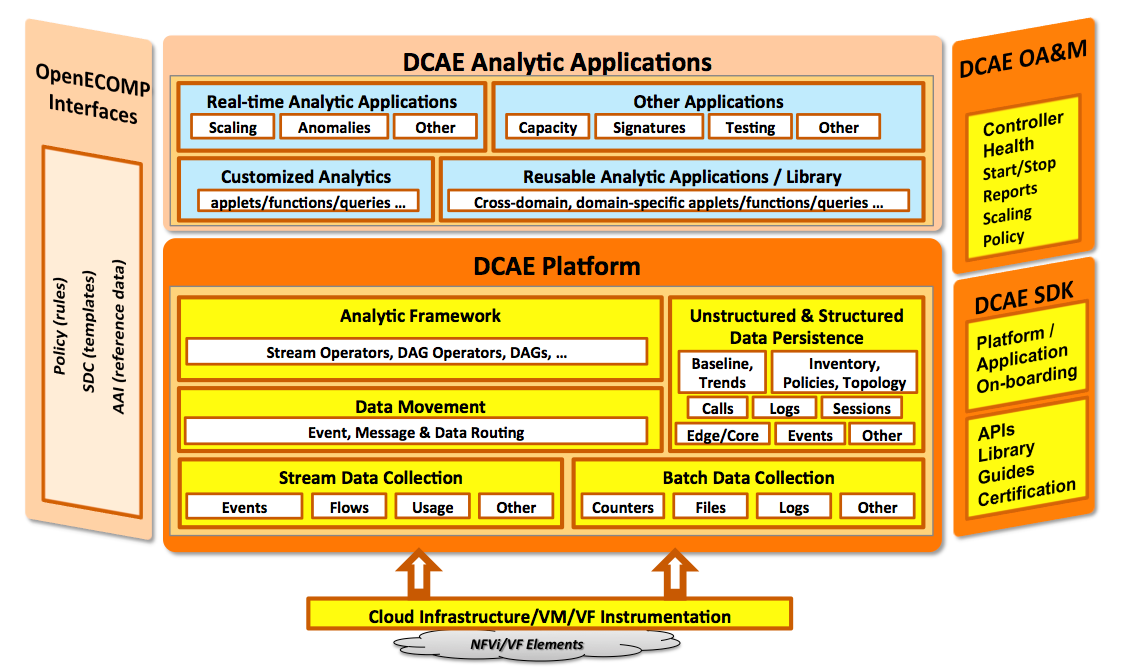


Figure 5.1.1.1: Functional view of the DCAE platform architecture

NOTE: Functional view of the DCAE platform architecture Figure attribution [6] – Figure 1.

The Data Collection, Analytics, and Events (DCAE) subsystem, in conjunction with other ONAP components, gathers performance, usage, and configuration data from the managed environment, which includes virtual network functions and their underlying infrastructure.

Figure 5.1.1.2 (see [6] - figure 3) shows the DCAE data flow and illustrates data flow issued by VNFs to push performance and usage data to the DCAE VES (VNF Event Stream) Collector, the DCAE Collection Framework component in charge of stream data collection.

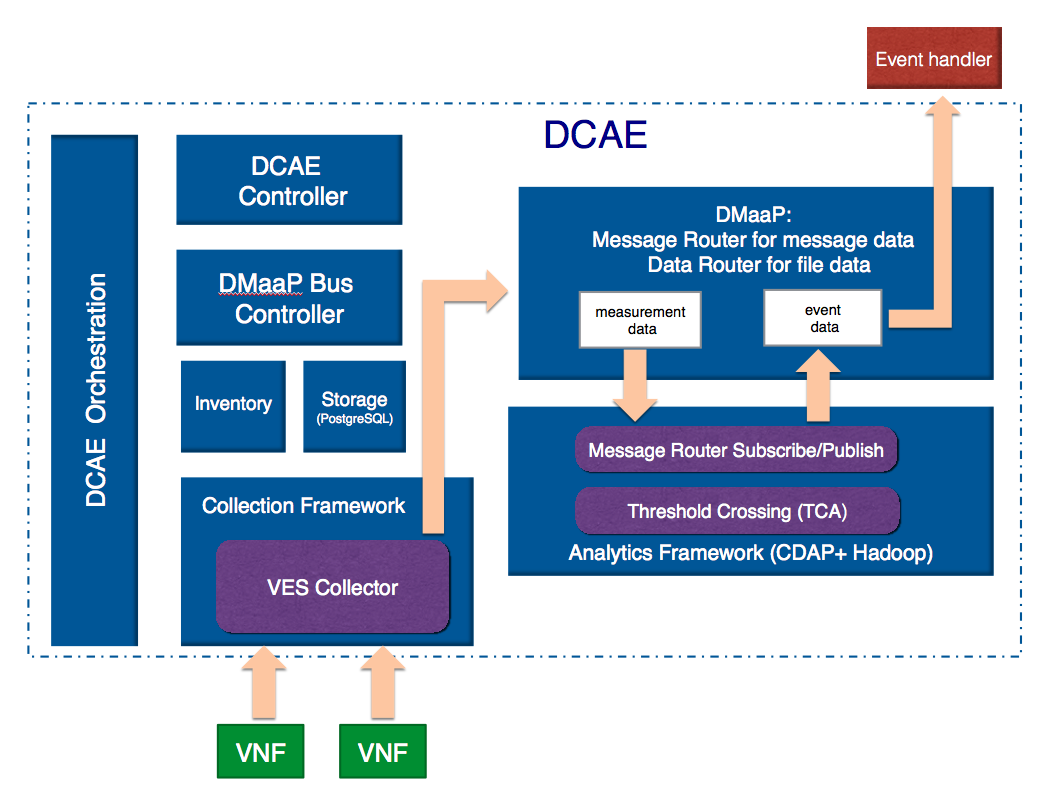


Figure 5.1.1.2: DCAE data flow (data plane)

NOTE: DCAE data flow (data plane) Figure attribution [6] – Figure 3.

### 5.1.2 Positioning

#### 5.1.2.0 General

If authorized, the ONAP / DCAE Collection Framework would be a consumer of PM and FM related notifications issued by 3GPP 5G management services, including, e.g.:

- fault management related notifications (cf. clause 6 of [11] and clause 6.1 of [10]),

- performance management related notifications (cf. clause 5.2 of [9] and clause 7.1 of [10]).

In subsequent sections, the focus is put on analysing how and to what extent various types of information produced by 3GPP 5G management services (alarm notifications, performance measurement results, configuration change notifications, etc.) can be consumed by the ONAP/DCAE Collection Framework.

#### 5.1.2.1 An example of ONAP DCAE Collection Framework utilizing the management services provided by 3GPP Management Services Provider

Figure 5.1.2.1.1 shows an example of ONAP DCAE Collection Framework utilizing the management services provided by 3GPP Data Report MnS provider. In this example, the ONAP DCAE Collection Framework, as part of the 3GPP MnS(s) Consumer, may utilize the management services (i.e., Performance Data Report MnS, Fault Supervision Data Report MnS) provided by 3GPP Management Service(s) Provider based on comparative analysis in the present document.

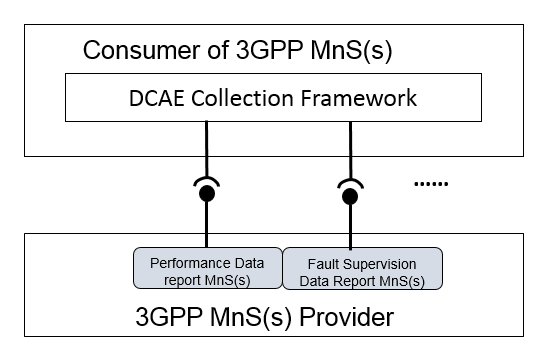


Figure 5.1.2.1.1: An example of ONAP DCAE Collection Framework utilizing the management services provided by 3GPP Data Report MnS(s) Provider

#### 5.1.2.2 An example of 3GPP MnF using information provided by ONAP DCAE

Figure 5.1.2.2.1 shows an example of 3GPP MnS Provider using information provided by ONAP DCAE. In this example, in order to support management data analytics, the 3GPP MDAF, which provides MDAS as an ONAP DCAE consumer, may use some information (e.g. FM data, PM data) provided by ONAP DCAE.



Figure 5.1.2.2.1: An example of 3GPP MnF using some information provided by ONAP DCAE

## 5.2 Comparative analysis

### 5.2.1 Communication principles

#### 5.2.1.1 ONAP DCAE

In ONAP (cf. [6]), VNFs use REST calls to push measurement data to the DCAE VES Collector. The VES Collector validates, filters, and packages the received measurement data and publishes the data to other components of the ONAP architecture. All VNF instances are provisioned with VES Collector address(es) so as to be able to send alarm notifications, event notifications, etc. to the VES Collector. ONAP does not offer a subscribe / notify mechanism between the DCAE / VES Collector and VNFs. Consequently, all possible notifications that can be issued by VNFs are to be sent out to the VES Collector.

Regardless of the type of notication, all notifications sent to the VES Collector will have a common header, whose schema is given in table 5.2.1.1.1.

NOTE: Table 5.2.1.1.1 is copied verbatim from ONAP (cf. [12]) and therefore the apparent normative language that appears therein does not imply a normative requirement within the context of the present document.

Table 5.2.1.1.1: ONAP DCAE VES Collector - common event header schema

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Required? | Description |
| domain | string | Yes | Event domain enumeration: 'fault', 'heartbeat', 'measurement', 'mobileFlow', 'notification', 'other', 'pnfRegistration', 'sipSignaling', 'stateChange', 'syslog', 'thresholdCross ingAlert', 'voiceQuality' |
| eventId | string | Yes | Event key that is unique to the event source. The key must be unique within notification life cycle similar to EventID from 3GPP. It could be a sequential number, or a composite key formed from the event fields, such as domain\_sequence. The eventId should not include whitespace. For fault events, eventId is the eventId of the initial alarm; if the same alarm is raised again for changed, acknowledged or cleared cases, eventId must be the same as the initial alarm (along with the same startEpochMicro sec but with a different sequence number). Note: see section 1.3 for eventId use case examples. |
| eventName | string | Yes | Unique event name (see section 1.2 for more information) |
| eventType | string | No | For example: 'applicationNf', 'guestOS', 'hostOS', 'platform' |
| internalHeaderFields | internalHeaderFields | No | Fields (not supplied by event sources) that the VES Event Listener service can use to enrich the event if needed for efficient internal processing. This is an empty object which is intended to be defined separately by each service provider (e.g., AT&T) implementing the VES Event Listener. |
| lastEpochMicrosec | number | Yes | the latest unix time aka epoch time associated with the event from any component–as microseconds elapsed since 1 Jan 1970 not including leap seconds |
| nfcamingCode | string | No | Network function component type: 3 characters (aligned with vfc naming standards) |
| nfNamingCode | string | No | Network function type: 4 characters (aligned with vnf and pnf naming standards) |
| nfVendorName | String | No | Network function vendor name |
| Priority | string | Yes | Processing priority enumeration: 'High', 'Medium', 'Normal', 'Low' |
| reportingEntityId | string | No | UUID identifying the entity reporting the event or detecting a problem in another vnf/vm or pnf which is experiencing the problem. (Note: the AT&T internal enrichment process shall ensure that this field is populated). The reportingEntity Id is an id for the reportingEntity Name. See 'reportingEntit yName' for more information.in |
| reportingEntityName | string | Yes | Name of the entity reporting the event or detecting a problem in another vnf/vm or pnf which is experiencing the problem. May be the same as the sourceName. For synthetic events generated by DCAE, it is the name of the app generating the event. |
| Sequence | integer | Yes | Ordering of events communicated by an event source instance (or 0 if not needed) |

|  |  |  |  |
| --- | --- | --- | --- |
| sourceId | string | No | UUID identifying the entity experiencing the event issue, which may be detected and reported by a separate reporting entity (note: the AT&T internal enrichment process shall ensure that this field is populated). The sourceId is an id for the sourceName. See 'sourceName' for more information. |
| sourceName | string | Yes | Name of the entity experiencing the event issue, which may be detected and reported by a separate reporting entity. The sourceName identifies the device for which data is collected. A valid sourceName must be inventoried in A&AI. If sourceName is a xNFC or VM, then the event must be reporting data for that particular xNFC or VM. If the sourceName is a xNF, comprised of multiple xNFCs, the data must be reported/aggreg ated at the xNF leveI. Data for individual xNFC must not be included in the xNF sourceName event. |
| startEpochMicro sec | number | Yes | the earliest unix time aka epoch time associated with the event from any component–as microseconds elapsed since 1 Jan 1970 not including leap seconds. For measurements and heartbeats, where events are collected over predefined intervals, startEpochMicro sec shall be rounded to the nearest interval boundary (e.g., the epoch equivalent of 3:00PM, 3:10PM, 3:20PM, etc…). For fault events, startEpochMicro sec is the timestamp of the initial alarm; if the same alarm is raised again for changed, acknowledged or cleared cases, startEpoch Microsec must be the same as the initial alarm (along with the same eventId and an incremental sequence number). For devices with no timing source (clock), the default value will be 0 and the VES collector will replace it with Collector time stamp (when the event is received) |
| timeZoneOffset | String | No | Offset to GMT to indicate local time zone for device formatted as 'UTC+/-hh.mm'; see <https://en.wikipedia.org/wiki/List_of_time_zone_abbreviations> for UTC offset examples |
| version | number | Yes | Version of the event header as "#.#" where # is a digit; see section 1 for the correct digits to use. |
| vesEventListenerVersion | String | Yes | Version of the ves event listener api spec that this event is compliant with (as "#" or "#.#" or "#.#.#" where # is a digit; see section 1 for the correct digits to use). |

NOTE: ONAP DCAE VES Collector - Common event header schema Figure attribution [12].

The VES Collector is a RESTful collector for processing JSON messages; an example of a notification header, in JSON, is given in [13] and reproduced here:

{

"event": {

"commonEventHeader": {

"version": "4.0.1",

"vesEventListenerVersion": "7.0.1",

"domain": "heartbeat",

"eventName": "Heartbeat\_vIsbcMmc",

"eventId": "heartbeat0000249",

"sequence": 0,

"priority": "Normal",

"reportingEntityId": "cc305d54-75b4-431b-adb2-eb6b9e541234",

"reportingEntityName": "ibcx0001vm002oam001",

"sourceId": "de305d54-75b4-431b-adb2-eb6b9e546014",

"sourceName": "ibcx0001vm002ssc001",

"nfVendorName": "Ericsson",

"nfNamingCode": "ibcx",

"nfcNamingCode": "ssc",

"startEpochMicrosec": 1413378172000000,

"lastEpochMicrosec": 1413378172000000,

"timeZoneOffset": "UTC-05:30"

}

}

}

NOTE: ONAP DCAE VES Collector - Common event header example in JSON Text attribution [13].

#### 5.2.1.2 3GPP 5G management framework

In the 3GPP 5G management framework, as specified in TS 28.533 [2], management services may generate notifications of events occurring within the network. Different kinds of events carry different kinds of information. For instance, a new alarm (as specified in TS 28.545 [11]) is one possible kind of event; an object creation (as specified in TS 28.531 [3]) is another possible kind of event.

Information of an event is carried in a notification. A management service producer may emit notifications. Authorized consumers receive notifications. Management service producers that are able to emit notifications offer the following capabilities:

- Notification mechanism subscription: management service producers provide authorized consumers with the capabilities to subscribe and unsubscribe to their notifications. A consumer can specify the types of notifications that the producer should emit to the consumer during subscription, to specify filtering criteria that can be applied by the notification mechanism;

- Subscription control: the management service provider may provide a consumer with capabilities to control its subscriptions;

- Notification control: in principle, notifications are forwarded to the consumer as soon as they are available;

#### 5.2.1.3 Comparison

##### 5.2.1.3.1 Events headers / notification parameters

The table below provides a high-level mapping between the ONAP common event header fields (in first column) and the 3GPP 5G management services notifications fields with similar functions (in second column).

|  |  |
| --- | --- |
| ONAP  Fields common to all events | Potential mapping to notification parameters in 3GPP TS 28.532 [10])  Notification parameters |
| **domain** (required): the eventing domain associated with the event. | Nothing similar in notification parameters. |
| **eventId** (required): event key that is unique to the event source. | Similar to notificationId. |
| **eventName** (required): unique event name | eventName could be mapped to notificationType.  Actual values of notificationType are specified on a per notification type basis. Example: in 3GPP TS 28.532 [10], the following actual notificationType values are specified: "notifyNewAlarm", "notifyChangedAlarm", "notifyFileReady", "notifyFilePreparationError", etc. |
| **eventType** (optional): for example: applicationVnf, guestOS, hostOS, platform | Nothing similar in notification parameters.  NFVI related event types are out of the scope of the present document. |
| **internalHeaderFields** (optional) | No definition of this field could be found. |
| **lastEpochMicrosec** (required): the latest unix time aka epoch time associated with the event from any component as microseconds elapsed since 1 Jan 1970 not including leap seconds | See eventTime, which does not distinguish the start time of the event from its end time. |
| **nfNamingCode** (optional): 4 character network function type, aligned with vnf naming standards | Nothing similar in notification parameters. |
| **nfcNamingCode** (optional): 3 character network function component type, aligned with vfc naming standards | Nothing similar in notification parameters. |
| **nfVendorName** (optional | Similar to vendorName |
| **priority** (required): processing priority | Nothing similar in notification parameters. |
| **reportingEntityId** (optional): UUID identifying the entity reporting the event, for example an OAM VM. | Similar to systemDN. |
| **reportingEntityName** (required): Name of the entity reporting the event, for example, an EMS name. May be the same as the sourceName. | Nothing similar in notification parameters. |
| **sequence** (required): Ordering of events communicated by an event source instance (or 0 if not needed) | Nothing similar in notification parameters. |
| **sourceId** (optional): UUID identifying the entity experiencing the event issue | Similar to objectInstance. |
| **sourceName** (required): Name of the entity experiencing the event issue | Nothing similar in notification parameters. |
| **startEpochMicrosec** (required): The earliest unix time aka epoch time associated with the event from any component as microseconds elapsed since 1 Jan 1970 not including leap seconds | See eventTime, which does not distinguish the start time of the event from its end time. |
| **timeZoneOffset** (optional) | Nothing similar in notification parameters. |
| **version** (required): version of the event header | Nothing similar in notification parameters. |
| **vesEventLinstnerVersion** (required) | Nothing similar in notification parameters. |

##### 5.2.1.3.2 FM/PM data collection

5.2.1.3.2.1 Existing gap

In ONAP, as previously described, VNF instances are provisioned with the address of the VES Collector and VNFs use REST calls to push FM and PM measurement data to the DCAE VES Collector. There is no Subscribe / Notify paradigm in place between the VES Collector and VNFs; all VNFs, by default, send out their 'performance, usage, and configuration data' to the DCAE / VES Collector as it becomes available. Therefore, it is necessary that VNFs know the address of the VES Collector (e.g. its URL or IP address) from the time of their creation, meaning that an input parameter is needed for the management operation which creates the VNFs, indicating the VES Collector address.

Within the existing 3GPP 5G management framework there is no mechanism for providing a VES Collector address during VNF instantiation, and thus this is an interoperability gap between 3GPP and ONAP. This can be solved by modifying the 3GPP Network Resource Model to add a new IOC capturing the attribute indicating the address of the recipient to which MOIs are to send their notifications.

5.2.1.3.2.2 Proposed modifications to 3GPP specifications

The following are proposed changes to 3GPP specification TS 28.541 (or TS 28.622 – FFS) to align the 3GPP 5G management framework with ONAP FM/PM data collection mechanisms:

The notificationsConsumerReference can be defined as a new attribute, possibly multi-valued. Whether this new attribute is added to an existing IOC or in a new IOC is FFS.

As a potential option in case a new IOC is introduced, a corresponding class diagram can be introduced. An example is given below:



Figure 5.2.1.3.2.2.1: NRM for support of FM/PM data consumer

In case a new IOC is introduced, it needs to be defined. The following is an example:

"

4.3.x NotificationConsumer

4.3.x.1 Definition

This IOC represents the consumer of notifications emitted by MOIs.

4.3.x.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| notificationsConsumerReference | M | T | T | F | T |

4.3.x.3 Attribute constraints

None.

4.3.x.4 Notifications

The common notifications defined in subclause 4.5 are valid for this IOC, without exceptions or additions.

".

#### 5.2.1.4 Comparison of interface technologies

The table below provides a high-level mapping between 3GPP and ONAP interface technologies.

Table 5.2.1.4-1: Comparison of 3GPP and ONAP interface technologies

|  |  |  |
| --- | --- | --- |
|  | 3GPP | ONAP |
| Management-application-layer- communication method | NETCONF (allowed protocol but not used in any SS)  SNMP  SOAP  REST | NETCONF  SNMP  VES  REST |
| Management-application-layer-protocols for bulk & file transfer | ftp  ftps  sftp  http  https | sftp  ftpes |
| Valid network layer protocols | IP | IP |
| Solution sets | SOAP (WSDL 1.1/SOAP 1.1) | JSON |
| Network Resource Model notations | UML  XSD | YANG  JSON |

#### 5.2.1.5 Heartbeat

##### 5.2.1.5.1 ONAP

Clause 7.4.1 - Data Model for Event Records - of [16] describes the data model for the collection of various types of data from xNFs, as illustrated below:

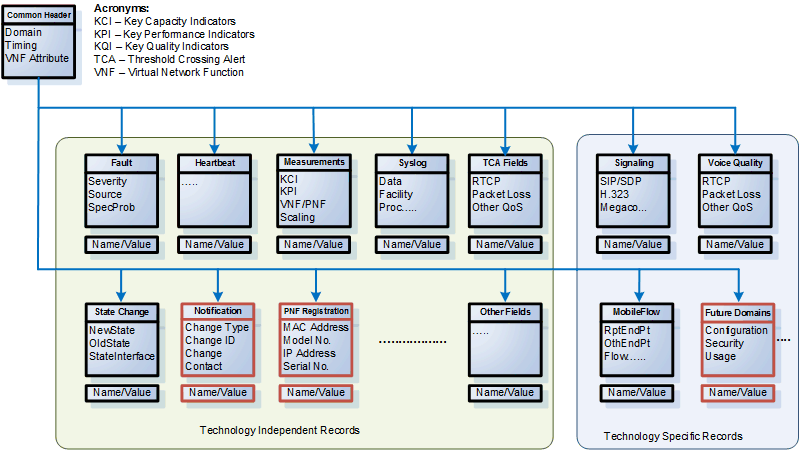


Figure 5.2.1.5.1. Data model for event records.

NOTE: ONAP data model of event records Figure attribution [16].

Clause 7.4.1 of [16] states:

*"*

*The Heartbeat Record provides an optional structure for communicating information about device health. Heartbeat records would only have the Common Event Header block. An optional heartbeat domain is available to specify information such as heartbeat interval and recommended action upon missing heartbeat interval. Heartbeat avoids the need to ping a device. A communication failure can be determined via missing heartbeat events being delivered to DCAE and appropriate action (e.g. restart VM, rebuild xNF or create ticket) can be taken by DCAE CLAMP.*

".

Bottom up, heartbeat messages are sent periodically by xNFs to the DCAE VES JSON Collector. These messages contain only the common header, in which the 'domain' field is equal to 'heartbeat'.

Top-down, heartbeat commands can be sent from an event collector, e.g. the DCAE VES JSON Collector, to xNFs.

##### 5.2.1.5.2 3GPP

For pre-5G technologies, 3GPP Communication Surveillance (CS) IRP provides functionalities similar to ONAP Heartbeat. 3GPP TS 32.352 [23] defines the following operations:

- getHeartbeatPeriod (), used by the CS IRPManager to obtain the current heartbeat period;

- setHeartbeatPeriod (),used by the CS IRPManager to set the heartbeat period;

- triggerHeartbeat (), used by the CS IRPManager to solicit a notifyHeartbeat notification

and the following notification:

- notifyHeartbeat (), used by the CS IRP Agent to notify the subscribed CS IRPManager instances that the resources supporting the communication path between the CS IRPAgent and the notification receiving CS IRPManagers are working.

##### 5.2.1.5.3 Proposed way forward

The concept of IRP has been replaced by the concept by Management Service (MnS) in the 5G management architecture framework [2]. However, there is no such Heartbeat or CS MnS defined in Rel-15. Consequently, it is proposed that such a MnS should be defined in Rel-16.

### 5.2.2 Handling of alarm / event notifications

#### 5.2.2.1 ONAP DCAE VES Fault Field

In ONAP, VES defines the fault domain for fault (alarm) notifications. Fault domain allows expansion, users can increase their own specific fault (alarm) related information.

VES fault fields schema is given in table 5.2.2.1.1 (cf. [18]).

Table 5.2.2.1-1: ONAP DCAE VES Collector - Fault fields schema

| Name | Required | Type | Format | Properties | Description |
| --- | --- | --- | --- | --- | --- |
| alarmAdditionalInformation | No | hashMap |  |  | Additional alarm information.  Note 1: for SNMP mapping to VES, for hash key use OID of varbind, for value use incoming data for that varbind).  Note 2: Alarm ID for 3GPP should be included (if applicable) in alarmAdditonalInformation as 'alarmId':'alarmIdValue'.  Could contain managed object instance as separate key:value; could add probable cause as separate key:value. |
| alarmCondition | Yes | string |  |  | Short name of the alarm condition/problem, such as a trap name. Should not have white space (e.g., tpLgCgiNotInCon fig, BfdSessionDown, linkDown, etc…) |
| alarmInterfaceA | No | string |  |  | card, port, channel or interface name of the device generating the alarm |
| eventCategory | No | string |  |  | Event category, for example: license, link, routing, security, signaling |
| eventSeverity | Yes | string |  | {'enum': ['CRITICAL', 'MAJOR', 'MINOR', 'WARNING', 'NORMAL']} | event severity |
| eventSourceType | Yes | string |  |  | type of event source; examples: card, host, other, port, portThreshold, router, slotThreshold, switch, virtualMachine, virtualNetworkFunction |
| faultFieldsVersion | Yes | number |  |  | version of the faultFields block |
| specificProblem | Yes | string |  |  | short description of the alarm or problem |
| vfStatus | Yes | string |  | {'enum': ['Active', 'Idle', 'Preparing to terminate', 'Ready to terminate', 'Requesting termination']} | virtual function status enumeration |

NOTE 1: ONAP DCAE VES Collector - Common fault fields schema Table attribution [18].

The VES Collector is a RESTful collector for processing JSON messages; an example of a fault fields, in JSON, is given below (cf. [13]):

"faultFields": {

            "faultFieldsVersion": 3.0,

            "alarmCondition": "PilotNumberPoolExhaustion",

            "eventSourceType": "other",

            "specificProblem": "Calls cannot complete - pilot numbers are unavailable",

            "eventSeverity": "CRITICAL",

            "vfStatus": "Active",

            "alarmAdditionalInformation": {

                "PilotNumberPoolSize": "1000"

            }

        }

NOTE 2: ONAP DCAE VES Collector - Common fault fields example in JSON Figure attribution [13].

#### 5.2.2.2 3GPP alarm supervision

The 3GPP fault supervision information model defines the AlarmInformation in TS 28.532 [10].

#### 5.2.2.3 Comparison

##### 5.2.2.3.1 Fault fields

A comparison of the ONAP fault event fields and 3GPP fault supervision AlarmInformation fields is given in Table 5.2.2.3.1-1.

Table 5.2.2.3.1-1: Comparison of ONAP and 3GPP fault fields

| ONAP  Fault events fields | Potential mapping to 3GPP TS 28.532 [10] – clause 6.2.1.3)  Generic fault supervision management service  AlarmInformation fields |
| --- | --- |
| **alarmAdditionalInformation** (optional): additional alarm information. | Similar to additionalText and/or additionalInformation. Probable cause would be included in this field. |
| **alarmCondition** (required): alarm condition reported by the device. | Nothing similar in Generic fault supervision management service AlarmInformation fields. Similar to alarmType |
| **alarmInterfaceA** (optional): card, port, channel or interface name of the device generating the alarm. | Nothing similar in Generic fault supervision management service AlarmInformation fields. |
| **eventCategory** (optional): Event category, for example: license, link, routing, security, signaling | Similar to eventType. |
| **eventSeverity** (required): event severity:CRITICAL, MAJOR, MINOR, WARNING, NORMAL | Similar to perceivedSeverity.  The legal values of perceivedSeverity include: CRITICAL, MAJOR, MINOR, WARNING, INDETERMINATE, CLEARED. |
| **eventSourceType** (required): type of event source; examples: card, host, other, port, portThreshold, router, slotThreshold, switch, virtualMachine, virtualNetworkFunction | Nothing similar in Generic fault supervision management service AlarmInformation fields. |
| **faultFieldsVersion**(required): version of the faultFields block | Nothing similar in Generic fault supervision management service AlarmInformation fields. |
| **specificProblem** (required): short description of the alarm or problem | Similar to specificProblem. |
| **vfStatus**(required): virtual function status enumeration | Nothing similar in Generic fault supervision management service AlarmInformation fields. |

##### 5.2.2.3.2 Mapping 3GPP alarm notifications to ONAP R3 VES JSON API

To allow mapping 3GPP alarm notifications to the ONAP R3 VES JSON API, 3GPP could request ONAP to add a new type of event to the existing types of events, i.e. a new possible value to the field 'domain', as shown in Table 5.2.2.3.2-1:

Table 5.2.2.3.2-1: ONAP DCAE VES common event header – domain field

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Required? | Description |
| domain | string | Yes | Event domain enumeration: 'fault', 'heartbeat', 'measurement', 'mobileFlow', 'notification', 'other', 'pnfRegistratio n', 'sipSignaling', 'stateChange', 'syslog', 'thresholdCross ingAlert', 'voiceQuality' |

NOTE: ONAP DCAE VES common event header - domain field Figure attribution [12].

It is proposed to add the following new possible value to the field 'domain' of ONAP VES JSON API:

- 3GPPFault, corresponding to 3GPP alarm notifications.

New types of fields would have to be added to the existing ones at the root level of the common event format of ONAP VES JSON API shown below:

Table 5.2.2.3.2-2: Existing types of headers for VES JSON Collector R3 ([12] – Datatype: event)

|  |  |  |
| --- | --- | --- |
| Name | Required | Type |
| commonEventHeader | Yes | commonEventHeader |
| faultFields | No | faultFields |
| heartbeatFields | No | heartbeatFields |
| measurementsForVfScalingFields | No | measurementsForVfScalingFields |
| mobileFlowFields | No | mobileFlowFields |
| otherFields | No | otherFields |
| sipSignalingFields | No | sipSignalingFields |
| stateChangeFields | No | stateChangeFields |
| syslogFields | No | syslogFields |
| thresholdCrossingAlertFields | No | thresholdCrossingAlertFields |
| voiceQualityFields | No | voiceQualityFields |

NOTE: ONAP DCAE VES JSON - Existing types of headers Figure attribution [12].

The following new type of fields would thus be added to ONAP DCAE VES JSON API header fields:

Table 5.2.2.3.2-3: Possible new ONAP DCAE VES JSON API header field

|  |  |  |
| --- | --- | --- |
| Name | Required | Type |
| Fault3GPPField | No | Fault3GPPFieldType |

where Fault3GPPField could use input parameter definitions of alarm notifications defined in 3GPP TS 28.532 [10]. An example, copied verbatim from clause 6.1.1.4.2 of 28.532 [10], valid for input parameters of notifyNewAlarm (), is shown in table 5.2.2.3.2-4.

Table 5.2.2.3.2-4: Example definitions of alarm notification fields

| Name | Required | Type | Format | Properties | Description |
| --- | --- | --- | --- | --- | --- |
| objectClass | M | MonitoredEntity.objectClass  It shall carry the MonitoredEntity class name. |  |  | The MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the new AlarmInformation. |
| objectInstance | M | MonitoredEntity.objectInstance  It shall carry the Distinguished Name (DN) of the instance of MonitoredEntity class. |  |  | The MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the new AlarmInformation. |
| notificationId | M | This is an identifier for the notification, which may be used to correlate notifications. The identifier of the notification shall be chosen to be unique across all notifications of a particular managed object throughout the time that correlation is significant, it uniquely identifies the notification from other notifications generated by the subject Information Object. |  |  |  |
| eventTime | M | AlarmInformation.alarmRaisedTime |  |  |  |
| systemDN | C | It shall carry the DN of service providers. |  |  |  |
| notificationType | M | "notifyNewAlarm". |  |  |  |
| probableCause | M | AlarmInformation.probableCause |  |  |  |
| perceivedSeverity | M | AlarmInformation.perceivedSeverity |  |  |  |
| rootCauseIndicator | O | It indicates that this AlarmInformation is the root cause of the events captured by the notifications whose identifiers are in the related CorrelatedNotification instances. |  |  | "Yes", "No" |
| alarmType | M | AlarmInformation.eventType |  |  | The notification structure of this table is applicable if this parameter indicates "Integrity Violation", "Operational Violation", "Physical Violation", "Security Service or Mechanism Violation", "Time Domain Violation". |
| specificProblem | O | AlarmInformation.specificProblem |  |  |  |
| correlatedNotifications | O | The set of CorrelatedNotification related to this AlarmInformation. |  |  |  |
| backedUpStatus | O | AlarmInformation.backedUpStatus |  |  |  |
| backUpObject | O | MonitoredEntity.objectInstance  It carries the DN of the back-up object. |  |  | This may contain no information if the identity of the service-user (requesting the service) is not known. |
| trendIndication | O | AlarmInformation.trendIndication |  |  | This shall always identify the service-provider receiving a service request, from serviceUser, that provokes the security alarm. |
| thresholdInfo | O | AlarmInformation.thresholdInfo |  |  | This may contain no information if the detector of the security alarm is the serviceProvider. |
| stateChangeDefinition | O | AlarmInformation.stateChangeDefinition |  |  |  |
| monitoredAttributes | O | AlarmInformation.monitoredAttributes |  |  |  |
| proposedRepairActions | O | AlarmInformaton.proposedRepairActions |  |  |  |
| additionalText | O | AlarmInformation.additionalText |  |  |  |
| additionalInformation | O | AlarmInformation.additionalInformation |  |  |  |
| alarmId | M | AlarmInformation.alarmId |  |  |  |

The following example in Table 5.2.2.3.2-5 illustrates the data type definition (described in clause 9.4.2.18) and JSON schema definition (described in Annex A.2) for notifyNewAlarm-NotifyType, which is valid forFault3GPPField:

**Table 5.2.2.3.2-5: Definition of type notifyNewAlarm-NotifType (defined in clause 9.4.2.18 in TS28.532)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Data type** | **Description** | **SQ** |
| header |  |  |  |
| > href | uri-Type | URI of the resource where the event (alarm) occurred | M |
| > notificationId | notificationId-Type | Notification identifier as defined in ITU-T Rec. X. 733 [4] | M |
| > notificationType | notificationType-Type | Notification type (notifyNewAlarm, etc.) | M |
| > eventTime | dateTime-Type | Event (alarm) occurrence time | M |
| > systemDN | systemDN-Type | System DN | C |
| body |  |  |  |
| > alarmId | alarmId-Type | Alarm identifier, see clause 6.2.1.5.1 | M |
| > alarmType | alarmType-Type | Alarm type as defined in ITU-T Rec. X. 733 [4] | M |
| > probableCause | probableCause-Type | Probable cause of an alarm as defined in ITU-T Rec. X.733 [4] | M |
| > specificProblem | specificProblem-Type | Identifies further refinements to the Probable cause of the alarm as defined in ITU-T Rec. X. 733 [4] | O |
| > perceivedSeverity | perceivedSeverity-Type | Perceived severity of an alarm as defined in ITU-T Rec. X. 733 [4] | M |
| > backedUpStatus | backedUpStatus-Type | Indicating if the object emitting the alarm has been backed up as defined in ITU-T Recommendation X. 733 [4] | O |
| > backUpObject | backUpObject-Type | Indicating the backup object of the alarmed object as defined in ITU-T Rec. X. 733 [4] | O |
| > trendIndication | trendIndication-Type | Severity trend of the alarmed object as defined in ITU-T Rec. X. 733 [4] | O |
| > thresholdInfo | thresholdInfo-Type | Provides additional information for threshold crossing alarms as defined in ITU-T Rec. X. 733 [4] | O |
| > correlatedNotifications | array(correlatedNotification-Type) | Set of all notifications to which this notification is considered to be correlated as defined in ITU-T Rec. X. 733 [4] | O |
| > stateChangeDefinition | array(attributeValueChange-Type) | Indicates a state transition associated to an alarm as defined in ITU-T Rec. X. 733 [4] | O |
| > monitoredAttributes | array(attributeNameValuePair-Type) | Defines one or more attributes of the alarmed manged object and their corresponding values at the time of the alarm as defined in ITU-T Rec. X. 733 [4]. | O |
| > proposedRepairActions | proposedRepairActions-Type | Used if the cause is known and the system being managed can suggest one or more solutions to fix the problem causing the alarm as defined in ITU-T Rec. X. 733 [4] | O |
| > additionalText | additionalText-Type | Allows a free form text description to be reported as defined in ITU-T Rec. X. 733 [4] | O |
| > additionalInformation | array(attributeNameValuePair-Type) | Allows the inclusion of a set of additional information in the event report as defined in ITU-T Rec. X. 733 [4] | O |
| > rootCauseIndicator | rootCauseIndicator-Type | Indicates if this event is the root cause of the events captured by the notifications whose identifiers are in the related correlatedNotifications attribute, see clause 6.2.1.5.1 | O |

The JSON schema definition (defined in Annex A.2 in TS 28.532) for "notifyNewAlarm-NotifType" is described as follows:

"notifyNewAlarm-NotifType": {

"type": "object",

"properties": {

"header": {

"$ref": "#/components/schemas/header-Type"

},

"body": {

"type": "object",

"properties": {

"alarmId": {

"$ref": "#/components/schemas/alarmId-Type"

},

"alarmType": {

"$ref": "#/components/schemas/alarmType-Type"

},

"probableCause": {

"$ref": "#/components/schemas/probableCause-Type"

},

"specificProblem": {

"$ref": "#/components/schemas/specificProblem-Type"

},

"perceivedSeverity": {

"$ref": "#/components/schemas/perceivedSeverity-Type"

},

"backedUpStatus": {

"$ref": "#/components/schemas/backedUpStatus-Type"

},

"backUpObject": {

"$ref": "#/components/schemas/backUpObject-Type"

},

"trendIndication": {

"$ref": "#/components/schemas/trendIndication-Type"

},

"thresholdInfo": {

"$ref": "#/components/schemas/thresholdInfo-Type"

},

"correlatedNotifications": {

"type": "array",

"items": {

"$ref": "#/components/schemas/correlatedNotification-Type"

}

},

"stateChangeDefinition": {

"type": "array",

"items": {

"$ref": "#/components/schemas/attributeValueChange-Type"

}

},

"monitoredAttributes": {

"type": "array",

"items": {

"$ref": "#/components/schemas/attributeNameValuePair-Type"

}

},

"proposedRepairActions": {

"$ref": "#/components/schemas/proposedRepairActions-Type"

},

"additionalText": {

"$ref": "#/components/schemas/additionalText-Type"

},

"additionalInformation": {

"type": "array",

"items": {

"$ref": "#/components/schemas/attributeNameValuePair-Type"

}

},

"rootCauseIndicator": {

"$ref": "#/components/schemas/rootCauseIndicator-Type"

}

}

}

}

},

#### 5.2.2.4 Alarm Life Cycle management in ONAP

The attributes of an alarm are defined in ONAP, example use cases exist, but there is no operation for acknowledgement of alarms, adding comments to alarms, or support for alarm correlation in ONAP.

### 5.2.3 Handling of performance data collection

#### 5.2.3.1 ONAP DCAE

In ONAP (cf. [6]), VNFs use REST calls to push measurement data to the DCAE VES Collector. The VES Collector validates, filters, and packages the received measurement data, and publishes the data to other components of the ONAP architecture. All VNF instances are provisioned with VES Collector address(es) so as to be able to send measurement data, in the form of event records, to the VES Collector.

The ONAP data model for event records consists of a Common Header, followed by records containing measurement data. In ONAP, measurement data may be of two kinds (cf. [14]):

- 'Technology independent' - these include Fault, Heartbeat, State Change, Syslog, Threshold Crossing Alerts, and measurements related to VNF scaling;

- 'Technology specific' - these include records related to Mobile Flow, Signalling and Voice Quality.

Both the Technology Independent and Technology Specific records are extensible.

Each technology-specific measurement data specification allows additional fields (name/value pairs) for extensibility.

In addition, technology specific measurement data can be extended to support other types of records (e.g. Network Fabric, Security records, etc.). In such cases, the VNF providers can use these VNF-specific additional fields to provide additional information that may be relevant to the managing systems. A placeholder for additional technology specific areas of interest to be defined in the future documents has been depicted.

This placeholder potentially leaves room for 3GPP to define performance measurement data for 3GPP technologies (GERAN, UTRAN, E-UTRAN).



Figure 5.2.3.1.1: ONAP R2 DCAE VES Collector – Technology-specific measurement data

NOTE: ONAP R2 DCAE VES Collector – Technology-specific measurement data Figure attribution [14].

#### 5.2.3.2 3GPP Performance Management

3GPP performance measurement data are specified on a per-technology basis, i.e. for GERAN, UTRAN, LTE and 5G technologies, and, for each technology, for the radio access network part and for the core network part. For 5G networks, all measurements are defined in TS 28.552 [15].

In 5G, performance management services (cf. 3GPP TS28.550 [9] and 28.532 [10]) enables a potential consumer to start / stop performance jobs on network functions of the 3GPP 5G network. When initiating a PM job, the consumer specifies the network functions on which the PM job is to be launched, the name of measurements to be collected, granularity period, collection period, etc. as well as threshold values, enabling the performance management service provider to issue notifications when actual performance measurement values reach or cross those threshold values.

Performance measurement data are made available to the consumer either via files or streaming (cf. [9]). Prior to this, potential consumers willing to be informed of the existence of such files or to receive streamed measurement data have to subscribe to the relevant notifications.

#### 5.2.3.3 Comparison

##### 5.2.3.3.1 PM job handling

5.2.3.3.1.1 Existing gap

In 3GPP 5G networks, performance measurement jobs are launched from an application service consumer on network functions. Performance measurement job parameters include start time, end time, identifiers of network function instances on which to collect the performance measurements, name of performance measurements to be collected, granularity period (i.e. the periodicity at which the measurements are collected), and reporting period (i.e. the periodicity at which the measurement results are made available to the consumer(s)). When measurement results are available, i.e. at the end of each reporting period, a notification ('Notify File Ready') is sent out to the consumer(s) that subscribed to such notifications, and which can then retrieve this measurement data via file transfer mechanisms.

In ONAP, there is no concept of Performance Management (PM) job. Instead, VNFs are configured with information indicating which performance measurements are to be collected, low and high thresholds assigned to these performance measurements, recommended actions to be initiated in case thresholds are crossed, etc. Virtualized network functions and services are designed within the ONAP Design Time Framework based on information supplied by the VNF provider about parameters, metrics, and KPIs exposed for resource and application management.

In ONAP Release 3, three collection frameworks are specified and VNF instances are configured, via configuration management, to enable them to collect specified measurements and send them to the specified DCAE collector.

In order to bridge this existing gap in the 3GPP 5G management framework, the Network Resource Model in TS 28.541 could be extended with IOCs and their attributes so as to represent PM job control capabilities, (e.g., the granularity period, reporting period, type of reporting (streaming, file-based), low and high thresholds assigned to these performance measurements, recommended actions to be initiated in case thresholds are crossed, etc.) and PM measurements which MOIs are to collect.

A potential integration of 3GPP 5G performance measurement data collection into ONAP/DCAE R3 would require the following:

- some ONAP component provides, at VNF instantiation time or later, a list of parameters corresponding to performance measurements to be collected during the life time of VNF instances, with their granularity period, reporting period, etc.;

- VNF instances have the capability to send their performance measurement data to the ONAP/DCAE Collector using the specified reporting method;

- the ONAP/DCAE VES Collector API is extended with the definition of 3GPP 5G technology-specific measurement data definitions defined in [15].

5.2.3.3.1.2 Proposed modifications to 3GPP specifications

A potential solution for the control of PM measurements in the 3GPP management framework, in line with ONAP, is to update TS 28.541 [4] with new IOCs and attributes, configurable via the provisioning management service (cf. [3] and [10]).

##### 5.2.3.3.2 FM/PM data reporting

5.2.3.3.2.1 Existing gap

In ONAP, the DCAE collection framework is aimed at collecting all sorts of events from VNFs.

In ONAP R1 and R2 architectures, the DCAE collection framework was comprised of:

- VES (VNF Event Stream) Collector, via REST / HTTPS / JSON API;

- SNMP Trap Collector, via SNMP.

In ONAP R3 architecture, the DCAE collection framework has been augmented with:

- File Collector, to support Bulk PM, to support 3GPP bulk data file collection over FTPES (SFTP);

- VES-HV (High-Volume) Collector, to support Real-Time Performance Measurement (RTPM), using Google Protocol Buffer (GPB) over TLS/TCP.

In ONAP R1 and R2, the following management requirements (still valid for R3) specified what xNFs (i.e. PNFs and VNFs) support with respect to how they deliver their alarms, performance measurements, etc.:

"

* *R-06924 The xNF MUST deliver asynchronous data as data becomes available, or according to the configured frequency. (cf.* [*https://onap.readthedocs.io/en/beijing/submodules/vnfrqts/requirements.git/docs/Chapter7.html*](https://onap.readthedocs.io/en/beijing/submodules/vnfrqts/requirements.git/docs/Chapter7.html) *- section 7.4.5.7)*
* *R-98191 The xNF MUST vary the frequency that asynchronous data is delivered based on the content and how data may be aggregated or grouped together. (cf.* [*https://onap.readthedocs.io/en/beijing/submodules/vnfrqts/requirements.git/docs/Chapter7.html*](https://onap.readthedocs.io/en/beijing/submodules/vnfrqts/requirements.git/docs/Chapter7.html) *- section 7.4.5.5)*

*Note:*

* *For example, alarms and alerts are expected to be delivered as soon as they appear. In contrast, other content, such as performance measurements, KPIs or reported network signaling may have various ways of packaging and delivering content. Some content should be streamed immediately; or content may be monitored over a time interval, then packaged as collection of records and delivered as block; or data may be collected until a package of a certain size has been collected; or content may be summarized statistically over a time interval, or computed as a KPI, with the summary or KPI being delivered.*
* *We expect the reporting frequency to be configurable depending on the virtual network function's needs for management. For example, Service Provider may choose to vary the frequency of collection between normal and trouble-shooting scenarios.*
* *Decisions about the frequency of data reporting will affect the size of delivered data sets, recommended delivery method, and how the data will be interpreted by ONAP. These considerations should not affect deserialization and decoding of the data, which will be guided by the accompanying JSON schema or GPB definition files.*

".

In ONAP R3, requirements for 'bulk performance measurement' have been introduced (cf. <https://onap.readthedocs.io/en/latest/submodules/vnfrqts/requirements.git/docs/Chapter9/index.html?highlight=ves>):

"

* *R-841740 - The xNF SHOULD support FileReady VES event for event-driven bulk transfer of monitoring data.*
* *R-440220 - The xNF SHOULD support File transferring protocol, such as FTPES or SFTP, when supporting the event-driven bulk transfer of monitoring data*
* *R-75943 - The xNF SHOULD support the data schema defined in 3GPP TS 32.435, when supporting the event-driven bulk transfer of monitoring data.*

".

In the 3GPP 5G management framework, the createMeasurementJob () operation (cf. draft TS 28.550 [1] – clause 6.1.1) supports, amongst others, the three following input parameters:

# reportingMethod

# granularityPeriod

# reportingPeriod

Note that, in draft TS 28.550 [1] – clause 6.1.1, it is stated that:

"

If the reportingMethod is performance data streaming:

- The value of granularityPeriod is an integer value in seconds (see Note 1 below).

Note 1: The granularityPeriod defines the measurement data production rate. The supported rates are dependent on the capacity of the producer involved (e.g. the processing power of the producer, number of measurements being measured by the producer at the time, the complexity of the measurement type involved etc) and therefore, it cannot be standardized for all producers involved. The supported rates can only reflect the negotiated agreement between producer and the consumer involved.

".

Table 5.2.3.3.2.1-1 gives an overview of reporting methods in 3GPP 5G management framework, supported values of Reporting Period, candidate collectors in DCAE (ONAP R3) and candidate solution sets in 3GPP Rel-16:

Table 5.2.3.3.2.1-1: Comparison of 3GPP and ONAP PM data reporting

|  |  |  |  |
| --- | --- | --- | --- |
| 3GPP performance data reporting method | Supported values of Reporting Period | Candidate collectors in ONAP R3 | Candidate 3GPP Rel-16 solution sets |
| Data file | 5 minutes, 15 minutes, 30 minutes, 1 hour, 12 hours and 24 hours | VES JSON Collector (for FileReady notification) | Protocol: REST / HTTPS  Payload: JSON |
| File Collector (for PM file upload) | Protocol: FTPES or SFTP  File content: XML |
| Data stream | No values defined in TS 28.550 2.0.0) | VES JSON Collector for low-medium volume PM | Protocol: REST / HTTPS  Payload: JSON |
| VES-HV Collector for real-time (less than 1 minute) high volume PM | Protocol: REST/HTTPS  Payload: ASN.1 Binary |

Table 5.2.3.3.2.1-2 applies for alarm notifications:

Table 5.2.3.3.2.1-2: Comparison of 3GPP and ONAP alarm event reporting

|  |  |  |  |
| --- | --- | --- | --- |
| 3GPP alarm reporting method | Number / Periodicity / Volume | Candidate collectors in ONAP R3 | Candidate 3GPP Rel-16 solution sets |
| Notification | Number: High (potentially)  Periodicity: unknown  Volume (of each notification): low | VES JSON Collector for alarm notifications under normal conditions | Protocol: REST / HTTPS  Payload: JSON |
| VES-HV Collector for alarm notifications under alarm flooding conditions | Protocol: REST / HTTPS  Payload: ASN.1 Binary |

5.2.3.3.2.2 Proposed modifications to 3GPP specifications

The following are proposed changes to 3GPP specifications TS 28.550 and TS 28.532 to align the 3GPP 5G management framework with ONAP for the reporting of FM/PM data:

- Specify in TS 28.550 (and in TS 28.532), in addition to existing file-based PM data reporting, how PM measurement data can be reported to a PM data consumer (e.g. ONAP DCAE Collection Framework), via:

# REST / HTTP and JSON for low-medium non-real-time reporting

# REST/HTTPS and ASN.1 Binary for high volume real-time reporting.

##### 5.2.3.3.3 Reporting 3GPP Rel-15 PM data to DCAE R3 VES JSON Collector

In [12] – clause 'Datatype: commonEventHeader', the 'domain' field indicates all types of events that can be received by the DCAE R3 VES JSON Collector:

Table 5.2.3.3.3-1: ONAP DCAE VES common event header domain field

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Required? | Description |
| domain | string | Yes | Event domain enumeration: 'fault', 'heartbeat', 'measurement', 'mobileFlow', 'notification', 'other', 'pnfRegistratio n', 'sipSignaling', 'stateChange', 'syslog', 'thresholdCross ingAlert', 'voiceQuality' |

NOTE: ONAP DCAE VES common event header - domain field Figure attribution [12].

As a potential solution for enabling the DCAE R3 VES JSON Collector to consume 3GPP Rel-15 PM data, 3GPP could collaborate with ONAP to add new possible values to the field 'domain' of ONAP VES JSON API:

- 3GPPNGRANMeasurement, corresponding to 3GPP TS 28.552 [15] – clause 5.1 (Performance measurements for gNB);

- 3GPP5GCMeasurement, corresponding to 3GPP TS 28.552 [15] – clauses 5.2 to 5.6;

- 3GPPNSMeasurement, corresponding to measurements for network slices defined 3GPP TS 28.552 [15] – clause 6;

- 3GPPE2EMeasurement, corresponding to measurements for networks defined 3GPP TS 28.552 [15] – clause 6.

New types of fields would have to be added to the existing ones at the root level of the common event format of ONAP VES JSON API shown in Table 5.2.3.3.3-2 below:

Table 5.2.3.3.3-2: Existing types of headers for VES JSON Collector R3 ([12] – Datatype: event)

|  |  |  |
| --- | --- | --- |
| Name | Required | Type |
| commonEventHeader | Yes | commonEventHeader |
| faultFields | No | faultFields |
| heartbeatFields | No | heartbeatFields |
| measurementsForVfScalingFields | No | measurementsForVfScalingFields |
| mobileFlowFields | No | mobileFlowFields |
| otherFields | No | otherFields |
| sipSignalingFields | No | sipSignalingFields |
| stateChangeFields | No | stateChangeFields |
| syslogFields | No | syslogFields |
| thresholdCrossingAlertFields | No | thresholdCrossingAlertFields |
| voiceQualityFields | No | voiceQualityFields |

NOTE: ONAP DCAE VES JSON - Existing types of headers Figure attribution [12].

The following new types of events would thus be added to ONAP DCAE VES JSON API header fields:

Table 5.2.3.3.3-3: Possible new types of headers for VES JSON Collector – Datatype: event

|  |  |  |
| --- | --- | --- |
| Name | Required | Type |
| 3GPPNGRANMeasurementFields | No | 3GPPNGRANMeasurementFields |
| 3GPP5GCMeasurementFields | No | 3GPP5GCMeasurementFields |
| 3GPPNSMeasurementFields | No | 3GPPNSMeasurementFields |
| 3GPPE2EMeasurementFields | No | 3GPPE2EMeasurementFields |

where 3GPPNGRANMeasurementFields, 3GPP5GCMeasurementFields, 3GPPNSMeasurementFields and 3GPPE2EMeasurementFields could use the input parameter definitions for performance data file content defined in 3GPP TS 28.532 [10] and names of measurements defined in TS 28.552[15] and TS 28.554[17].

##### 5.2.3.3.4 3GPP Fault Supervision operations

3GPP alarm notifications can be mapped to HTTP messages from xNFs to DCAE Collectors.

3GPP fault supervision data report management service for NF is also made up of the following operation (cf. [10] clause 6.1.1):

- getAlarmList ().

getAlarmList () may be used by a fault supervision data report management service consumer to deal with alarm loss. 3GPP TS 28.545 [11] does not specify conditions under which a management service consumer should invoke this operation. In the ONAP architecture, the fault supervision data report service consumer receiving alarm notifications from xNFs would typically be a DCAE Collector (VES/JSON or VES-HV), whereas the fault supervision data report service consumer issuing getAlarmList () operation requests to xNFs would be the APP-C.

3GPP fault supervision data control management service for NF is made up of the following operations (cf. [10] clause 6.1.2):

- acknowledgeAlarms ();

- unacknowledgeAlarms ();

- clearAlarms ().

The fault supervision data control management service consumer issuing acknowledgeAlarms (), unacknowledgeAlarms (), clearAlarms () operation requests to xNFs would be the ONAP APP-C.

## 5.3 Conclusions on FM/PM

The clause 5 of the present document studies the positioning of ONAP DCAE Collectors with regard to 3GPP management services for fault and performance management purposes. It also analyzes how ONAP DCAE Collectors can consume 3GPP management services, from various perspectives:

- General communication principles for all sorts of messages (FM and PM), based on ONAP DCAE common event header. It identifies a gap between ONAP DCAE Collectors which addresses, as recipient of notifications from 3GPP management service producers, are known by 3GPP management service producers (e.g. VNFs / PNFs), and 3GPP management framework which relies on the subscribe / notify paradigm;

- Fault management: possible mapping of 3GPP fault supervision alarm notification parameters to ONAP DCAE Collectors’ API, either by using existing technology-independent records or by defining a new 3GPP-specific fault management technology-specific record;

- Fault management operations, specified in 3GPP fault supervision management services, and not addressed by ONAP;

- Performance management: possible mapping of 3GPP performance measurement data reporting to ONAP DCAE Collectors’ API, both data file based and stream based. For the latter one, the study illustrates how the mapping can be done, either by using existing technology-independent records or by defining a new 3GPP-specific performance management technology-specific record. It aslo shows that 3GPP PM job control can be supported via configuration management in ONAP;

- Heartbeat mechanism, enabling DCAE Collectors to stay informed that VNFs / PNFs are alive.

The group recommends starting normative work on the following aspects:

- General communication aspects:

- Specifying a mapping between 3GPP notification mandatory input parameters and ONAP DCAE common event header required fields,

- Specifying a corresponding JSON schema,

- Specifying a solution for capturing the 3GPP notification recipients addresses when subscribing to notifications is not supported;

- Fault supervision:

- Definition of new use cases and requirements for fault supervision management services,

- Specifying a mapping from existing 3GPP alarm notification input parameters (except those addressed above) to a new ONAP VES JSON schema,

- Specifying JSON solution set definitions for fault supervision data report;

- Performance management:

- Definition of new use cases and requirements for performance management services,

- Specifying a mapping from existing 3GPP performance measurement data streaming parameters (except those addresseed above) to a new ONAP VES JSON schema

- Specifying an alternative to existing 3GPP performance management job control via configuration management based control of performance management,

- Specifying JSON and ASN.1 solution set definitions for performance measurement data streaming;

- Heartbeat:

- Definition of use cases and requirements for heartbeat,

- Specifying solutions for heartbeat, in case of 3GPP network functions.

# 6 Integration of ONAP with 3GPP management services for configuration management purposes

## 6.1 ONAP controllers and 3GPP provisioning service

### 6.1.1 ONAP controllers

#### 6.1.1.1 Introduction

The southbound interface of the ONAP platform for LCM and CM operations is based on a layer of controllers. Controllers are components in the ONAP platform that connect to cloud and network services, execute configuration and real-time policies, and control the state of distributed components and services. Rather than using a single monolithic control layer, operators may choose to use multiple distinct controller types to manage different resources in the execution environment, each corresponding to a distinct domain.

ONAP controllers expose a northbound API to other ONAP components that allow them to initiate activities (effectively commands) on VNFs and/or PNFs. ONAP controllers interact southbound with xNFs through network and application adapters to perform configuration and other lifecycle management activities towards xNFs.

In ONAP R3, the following controllers are available:

- Application Controller (APPC)

- Software Defined Networking Controller (SDNC)

- Virtual Function Controller (VF-C).

These controllers are further described in the following sections.

#### 6.1.1.2 APPC

##### 6.1.1.2.1 Overview

The Application Controller (APPC) is one of the components in the ONAP platform. Its main function is to control the lifecycle of VNFs and VNFCs (but not PNFs). APPC can use one of several adapters to connect to a VNF.

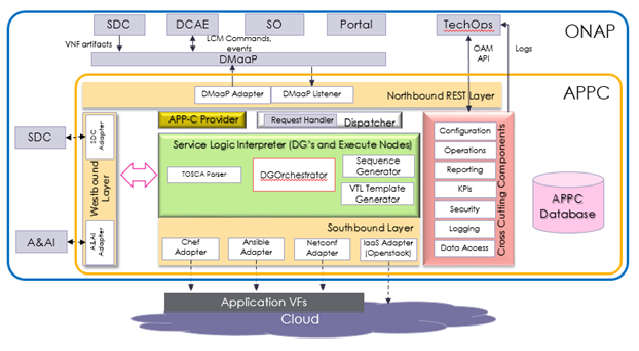


Figure 6.1.1.2.1-1: APPC High Level Architecture

NOTE: APPC High Level Architecture Figure attribution [20].

In ONAP R3, APPC contains adapters towards the following interface protocols:

- Netconf with uploadable Yang model (requires a Netconf server running on the VNF)

- Chef (requires a Chef client running on the VNF)

- Ansible (no requirements on the VNF software)

APPC interacts with VNFs through its Network and Application Adapters to perform configuration activities within NFV environment. These activities include service and resource configuration/reconfiguration, automated scaling of resources, service and resource removal to support runtime lifecycle management of VNFs and services. The Adapters employ a model driven approach along with standardized APIs provided by the VNF developers to configure resources and manage their runtime lifecycle.

The APPC architecture is designed to allow a VNF owner or vendor to enable a new VNF to support a set of LCM API actions that are designated as self-service. This procedure is known as VNF self-service onboarding. The VNF owner/vendor needs to create a template and parameter definitions for LCM actions that use the Netconf, Chef, or Ansible protocols.

The template is a text block in XML (if the protocol is Netconf) or JSON (for Chef or Ansible) format that defines the "payload" to be included in the request. This file contains the desired configuration parameter values to execute the LCM action and is sent, either directly to the VNF (for Netconf), or to the Chef/Ansible server that in turn uses these values to configure the VNF.

##### 6.1.1.2.2 APPC Netconf Adapter

The APPC Netconf Adapter is responsible for sending configuration tasks in XML format to VNFs that support the Netconf protocol. APPC includes a deployment of OpenDaylight and thus can consume NRMs expressed in the YANG modelling language.

##### 6.1.1.2.3 APPC Chef Adapter

Chef is an open-source VNF management framework. The APPC Chef adapter enables APPC to operate cookbooks that perform various LCM operations towards VNFs connected to an external Chef server. This server acts as a hub for configuration data and stores cookbooks, the policies that are applied to VNFs, and metadata that describes each registered VNF that is being managed by Chef. A Chef client is required on the VNFs that will be directly managed by the Chef server. The client exposes REST APIs used to manage the VNF.

A cookbook is the fundamental unit of configuration and policy distribution. A cookbook defines a scenario and contains recipes that specify the resources to use and the order in which they are to be applied. A recipe is expressed in the Ruby language and consists mainly of a collection of resources, defined using patterns (resource names, attribute-value pairs, and actions). Each recipe is required to be stored in a cookbook and may be included in another recipe or have a dependency on one (or more) recipes.

A cookbook also contains:

- Attribute values

- File distributions

- Templates

- Extensions to Chef, such as custom resources and libraries

##### 6.1.1.2.4 APPC Ansible Adapter

Ansible is an open-source VNF management framework. The APPC Ansible adapter enables APPC to operate playbooks that perform various LCM operations towards VNFs connected to an external Ansible server. This server can execute Ansible playbooks southbound towards the VNF and exposes a REST interface northbound towards APPC that is compliant with its requirements. The exact implementation of the Ansible Server is left open. Any action (e.g.,configure, restart, health check) can be executed on the VNF by constructing a playbook or set of playbooks that is executed by an Ansible agent on the VNF via SSH.

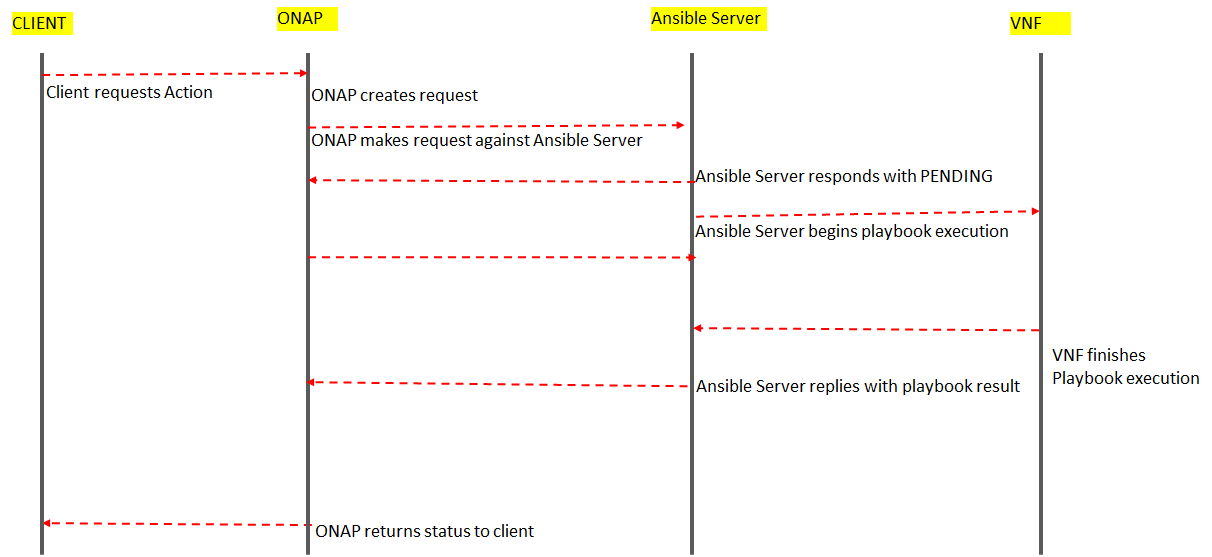


Figure 6.1.1.2.4-1: Ansible workflow envisioned when APPC receives an event

NOTE: Ansible workflow envisioned when APPC receives an event Figure attribution [21].

Playbooks are expressed in YAML format and model a configuration or process. Each playbook is composed of one or more 'plays' in a list. Each play contains of a list of tasks and indicates a set of host machines to target and a remote user for authentication. Tasks are executed in order, one at a time, against all machines matched by the host pattern, before moving on to the next task. Within a play, all hosts receive the same task directives. It is the purpose of a play to map a selection of hosts to tasks.

The goal of each task is to execute a single Ansible module with specific arguments. Ansible modules are reusable, standalone scripts that execute within an Ansible server and return information in JSON format. Variables can be used in arguments to modules. Each module has a dedicated use, such as administering users on a specific type of database or managing VLAN interfaces on a specific type of network device. Modules are for the most part vendor-specific.

#### 6.1.1.3 SDNC

The Software Defined Networking Controller supports management of transport network resources. Before ONAP R3, the ONAP controllers (APPC and SDNC) were classified as L4-L7 (APPC) and L0-L3 (SDNC) controllers. But this classification was removed in R3, which would mean that the scope of each controller becomes wider and potentially overlapping. Configuration management is not supported by SDNC in ONAP Rel-3.

#### 6.1.1.4 VF-C

The Virtual Function Controller (VF-C) is an implementation of an ETSI NFV-MANO stack and provides an LCM interface towards ETSI-compliant VNFs. VF-C also provides a GVNFM capability and can integrate with external VNFMs and VIMs as part of an NFV MANO stack.

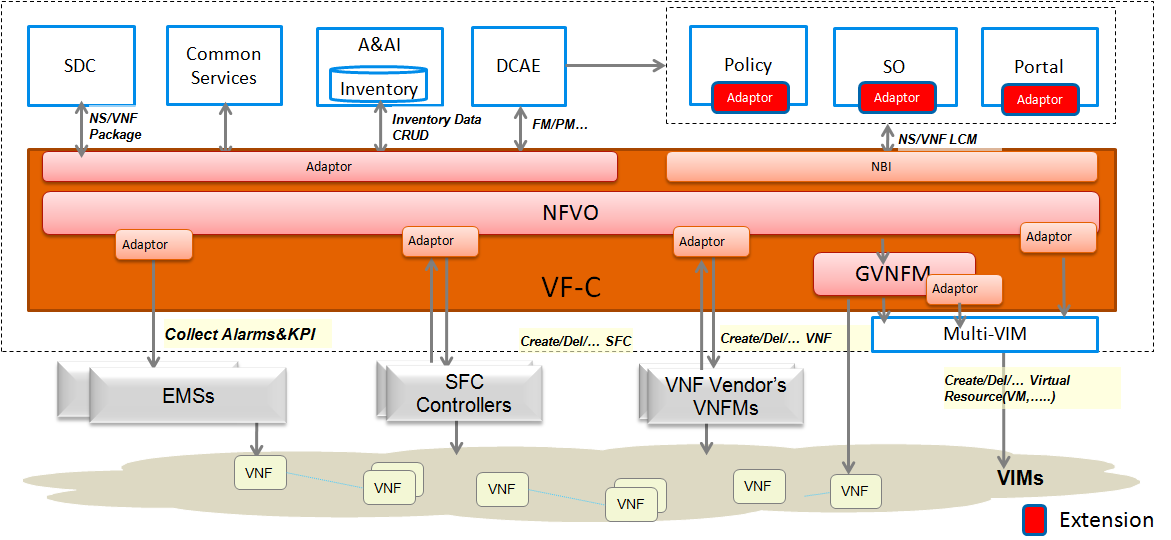


Figure 6.1.1.4-1: VF-C High Level Architecture

NOTE: VF-C High Level Architecture Figure attribution [8].

### 6.1.2 3GPP provisioning service

Management services for provisioning of networks and network slicing are defined in [3]. The management services for /NS/NSS/NF/network and subnetwork provisioning are listed in the following tables.

Table 6.1.2-1: Management services for network slice provisioning

|  |  |  |  |
| --- | --- | --- | --- |
| **MnS Name** | **MnS Component Type A (operations and notifications)** | **MnS Component Type B (information model)** | **Note** |
| Provisioning for NSI | Operations defined in clause 5 of TS 28.532 [10]:  - createMOI operation  - deleteMOI operation  - getMOIAttributes operation  - modifyMOIAttributes operation  Operations defined in clause 6.5:  - allocateNsi operation  - deallocateNsi operation | NSI information model defined in clause 6.3 of TS 28.541 [4] | This management service enables its consumer to request allocating, deallocating, or modifying an NSI.  The typical scenario is "Network Slices as NOP internals" model where this MnS is consumed by operators. |
| Provisioning data report for NSI | Notification defined in clause 5 of TS 28.532 [10]:  notifyProvisioning notification | NSI information model defined in clause 6.3 of TS 28.541 [4] | This management service enables its consumer to obtain notifications about NSI Information model data.  The typical scenario is "Network Slices as NOP internals" model where this MnS is consumed by operators |
| Provisioning exposure for NSI | Operations defined in clause 5 of TS 28.532 [10]:  - createMOI operation  - deleteMOI operation  - getMOIAttributes operation  - modifyMOIAttributes operation  Operations defined in clause 6.5:  - allocateNsi operation  - deallocateNsi operation | NSI information model defined in clause 6.3 of TS 28.541 [4] | This management service enables its consumer to request allocating, deallocating or modifying an NSI.  The typical scenario is NSaaS model where this MnS is consumed by vertical industry. |
| Provisioning data report exposure for NSI | Notifications defined in clause 5 of TS 28.532 [10]:  - notifyProvisioning notification | NSI information model defined in clause 6.3 of TS 28.541 [4] | This management service enables its consumer to obtain notifications about NSI Information model data.  The typical scenario is NSaaS model where this MnS is consumed by vertical industry. |

Table 6.1.2-2: Management services for NSS provisioning

|  |  |  |
| --- | --- | --- |
| **MnS Name** | **MnS Component of type A (Operations and notifications)** | **MnS Component of type B (information model)** |
| Provisioning for NSSI | Operations defined in clause 5 of TS 28.532 [10]:  - createMOI operation  - modifyMOIAttributes operation  - getMOIAttributes operation  - deleteMOI operation  Operations defined in clause 6.5:  - allocateNssi operation  - deallocateNssi operation | NSSI information model defined in clause 6.3 of TS 28.541 [4] |
| Provisioning data report for NSSI | Notifications defined in clause 5 of TS 28.532 [10]:  notifyProvisioning notification | NSSI information model defined in clause 6.3 of TS 28.541 [4] |

Table 6.1.2-3: Management services for NF provisioning

|  |  |  |
| --- | --- | --- |
| **MnS Name** | **MnS Component of type A (Operations and notifications)** | **MnS Component of type B (information model)** |
| Provisioning for NF | Operations defined in clause 5 of TS 28.532 [10]:  - createMOI operation  - modifyMOIAttributes operation  - getMOIAttributes operation  - deleteMOI operation | NF(s)/ME(s) information model defined in TS 28.541 [4] |
| Provisioning data report for NF | Notifications defined in clause 5 of TS 28.532 [10]:  notifyProvisioning notification | NF(s)/ME(s) information model defined in TS 28.541 [4] |

Table 6.1.2-4: Management services for network and sub-network provisioning

|  |  |  |
| --- | --- | --- |
| **MnS name** | **MnS Component of type A (Operations and notifications)** | **MnS Component of type B (information model)** |
| Provisioning for network and sub-networks | Operations defined in clause 5 of TS 28.532 [10]:  - createMOI operation  - modifyMOIAttributes operation  - getMOIAttributes operation  - deleteMOI operation  Operation defined in clause 6.5:  - AllocateNetwork operation | IOC(s) of sub-network, as defined in TS 28.541 [4] |
| Provisioning data report for sub-networks | Notifications defined in clause 5 of TS 28.532 [10]:  notifyProvisioning notification | IOC(s) of sub-network, as defined in TS 28.541 [4] |

### 6.1.3 Positioning

#### 6.1.3.1 Introduction

If authorized, the ONAP controller plays the role of the consumer of the provisioning management services produced by a 3GPP management service producer. If authorized, the 3GPP management function can play the role of the consumer of the LCM and CM interface produced by ONAP controller.

#### 6.1.3.2 An example of ONAP controller utilizing management services provided by 3GPP Management Services Provider

Figure 6.1.3.2-1 shows an example of ONAP Controller (e.g. APPC) utilizing part of provisioning management services provided by 3GPP Management Service Provider. In this example, the ONAP Controller as part of 3GPP MnS(s) Consumer may utilize part of provisioning management services (e.g. configuration related provisioning management service components) as follows provided by 3GPP Management Service Provider based on comparative analysis in the present document:

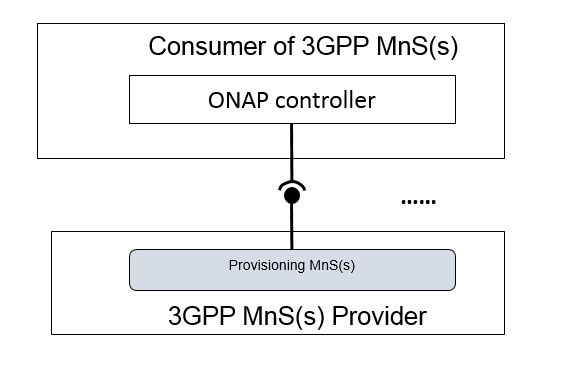


Figure 6.1.3.2-1: An example of ONAP Controller utilizing the management services provided by 3GPP MnS(s) Provider

#### 6.1.3.3 An example of 3GPP MnF using interface provided by ONAP VF-C

Figure 6.1.3.3-1 shows an example of 3GPP MnF using interfaces provided by OANP VF-C. In this example, in order to support provisioning MnS in the virtualization environment, the 3GPP MnF which provide provisioning MnS as ONAP VF-C consumer may use some interfaces (e.g. Network Service lifecycle management interface and VNF lifecycle management interface) provided by ONAP VF-C for the lifecycle management of Network Service(s) and VNF(s).

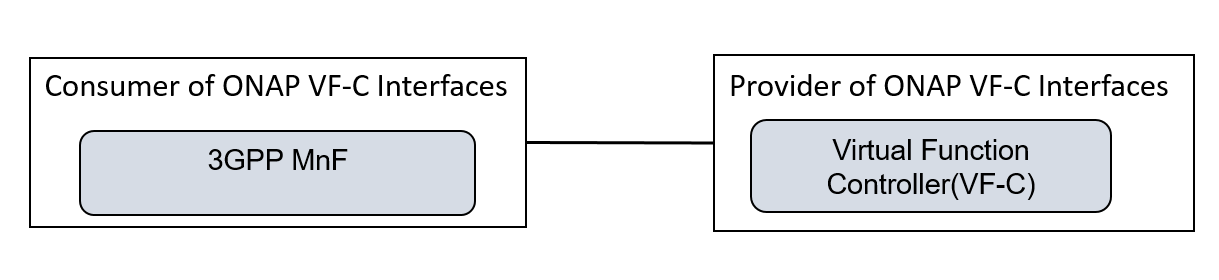


Figure 6.1.3.3-1: An example of 3GPP MnF using some interfaces provided by ONAP VF-C

NOTE: In this example, the 3GPP MnF is used to provide the provisioning MnS.

## 6.2 Comparative analysis

### 6.2.1 Introduction

This clause provides a comparison of configuration management capabilities between ONAP controllers and 3GPP 5G management services for provisioning based on the assumed positioning (cf. clause 6.1).

### 6.2.2 Management principles

APPC is designed to perform LCM and CM operations on VNFs. In ONAP R3, APPC does not officially support any operations towards PNFs.

VF-C is an implementation of an ETSI NFV-MANO stack and provides an LCM interface towards ETSI-compliant VNFs via the Os-Ma-nfvo, Ve-Vnfm-em and Ve-Vnfm-vnf reference points (cf. clause 5.2 of [2]). These reference points are outside of the domain of the 3GPP 5G management services. VF-C is not designed to perform application CM operations on VNFs.

A 3GPP 5G management service producer provides provisioning management services to perform LCM and CM operations and related notifications on a 5G network. The 5G network is modelled in the 5G NRM (model is specified in YANG, JSON and XML solution sets) and allows the consumer of the provisioning management service to create, delete, modify and get attributes of any object instances in the model (NRM) The consumer may additionally subscribe to receive notifications on creation and deletion of an object instance or an attributeValueChange of an object instance. The NRM fragments includes Subnetwork, NR, 5GC, NetworkSlice, NetworkSliceSubnet, ME, MF, VNF and PNF (VNF and PNF models are imported from ETSI NFV). For provisioning management of Subnetwork, NetworkSlice and NetworkSliceSubnet the 3GPP provisioning management service producer can in addition perform allocate and deallocate operations.

### 6.2.3 Communication protocols

#### 6.2.3.1 NETCONF

APPC allow the use of the NETCONF protocol towards the xNF. (As noted above, APPC does not officially have support for PNFs in ONAP R3). APPC and their Adapters utilize device YANG model and NETCONF APIs to make the required changes in the VNF state and configuration. APPC support the following operations which act directly upon the VNF. Most of these utilize the NETCONF interface.

The Table below is an example mapping between APPC configuration management action and NETCONF commands.

Table 6.2.3.1.1: Example mapping table between APPC configuration management action and NETCONF commands

|  |  |  |  |
| --- | --- | --- | --- |
| **Action** | **Description** | **VNF Action** | **NETCONF COMMANDs** |
| Audit | Compare active configuration against a configuration stored in ONAP’s configuration store. | Retrieve running configuration and device state information. Get-config updates the config tree which can then be compared to the stored current config in the ONAP database. | get-config |
| Configure | Configures the target VNF or VNFC. | The <edit-config> operation loads all or part of a specified configuration data set to the specified target VNF. | edit-config, commit |
| ConfigModify | The ConfigModify LCM action affects only a subset of the total configuration data of a VNF. It can be used to change specific parameters across a number of separate instances for the same VnfcType without changing instance specific values of each. It can also be used to make successive changes to a number of parameters where those changes are considered cumulative. Thus each ConfigModify invocation leaves previous values untouched and only edits the parameters which are sent to ONAP. | The <edit-config> operation loads only a part of the full set of configuration parameters to the specified target configuration without changing any existing parameters. | edit-config, commit |
| ConfigSave | Saves a VNF’s running configuration into the configuration store in ONAP, for later retrieval. | (optional) If copy-config to a local file is supported by the VNFC this command is used to store the running config locally in order to save time on any subsequent Reconfigure. To support this action, the VNF must allow <copy-config> to save to a local file and must support subsequent retrieval of the copied configuration back to the running configuration. If this capability is not supported, ONAP will still function, but updates will take longer. | copy-config, delete-config |
| ConfigRestore | Reconfigure a VNF to some previously stored baseline configuration stored by a previous ConfigSetBaseline. | If a previous config has been saved locally use quick restore (<copy-config> from file). If the restore fails, fallback to a process of changing the configuration value by value using <edit-config>. | edit-config or copy-config |

In 3GPP release 15, a YANG model of the 5G NRM is specified in TS 28.541. 3GPP SA5 has yet to decide if the required NETCONF solution should be introduced as Rel-16 new feature.

#### 6.2.3.2 Other protocol options

The Chef and Ansible protocols supported by ONAP controllers are primarily intended for LCM operations rather than CM operations. Furthermore, both protocols require the use of an open-source-based server external to ONAP. For these reasons, these protocols are not applicable to, or appropriate for, CM operations on the 5G NRM.

### 6.2.4 Handling of configuration management notifications

#### 6.2.4.1 3GPP configuration management notifications

3GPP configuration management notifications (i.e. notifyMOICreation, notifyMOIDeletion and notifyMOIAttributesValueChange) are described in clauses 5.1.5, 5.1.6 and 5.1.7 of TS 28.532 [10].

#### 6.2.4.2 ONAP configuration notifications

Currently, there is no existing domain field of the ONAP R3 VES JSON API that can be used to support or map to 3GPP configuration management notifications.

#### 6.2.4.3 Potential solution for enabling ONAP DCAE VES JSON Collector to consume 3GPP configuration management notifications

As a potential solution for enabling the DCAE VES JSON Collector to consume 3GPP configuration management notifications, 3GPP could collaborate with ONAP to add new possible values to the field 'domain' of ONAP VES JSON API:

- Prov3GPP, corresponding to 3GPP provisioning notifications.

The following new type of fields would thus be added to ONAP DCAE VES JSON API header fields:

Table 6.2.4.3-1: New domain ‘Prov3GPPFields’

|  |  |  |
| --- | --- | --- |
| Name | Required | Type |
| Prov3GPPFields | No | Prov3GPPFields |

whereProv3GPPFields could use input parameters definitions of provisioning related notifications defined in 3GPP TS 28.532 [10]. Following example illustrates the data type definition (described in clause 8.3.2.15 of TS 28.532 [10]) and JSON schema definition (described in Annex A.1) for notifyMOICreation-NotifyType, which is valid for Prov3GPPFields.

Table 6.2.4.3-2: Definition of type notifyMOICreation-NotifType (defined in clause 8.3.2.15 of TS 28.532 [10])

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Data type** | **Description** | **SQ** |
| header |  |  |  |
| > href | uri-Type | URI of the resource where the event (alarm) occurred | M |
| > notificationId | notificationId-Type | Notification identifier as defined in ITU-T Rec. X. 733 [4] | M |
| > notificationType | notificationType-Type | Notification type (notifyMOICreation) | M |
| > eventTime | dateTime-Type | Event (MOI creation) occurrence time | M |
| > systemDN | systemDN-Type | System DN | M |
| body |  |  |  |
| > correlatedNotifications | array(correlatedNotification-Type) | Set of all notifications to which this notification is considered to be correlated as defined in ITU-T Rec. X. 733 [4] | O |
| > additionalText | additionalText-Type | Allows a free form text description to be reported as defined in ITU-T Rec. X. 733 [4] | O |
| > sourceIndicator | sourceIndicator-Type | Indicates the source of the operation that led to the generation of this notification. | O |
| > attributeList | array(attributeNameValuePair-Type) | The attributes (name/value pairs) of the created MOI. | O |

The JSON schema definition (defined in Annex A.1 of TS 28.532 [10]) for "notifyMOICreation-NotifType" is described as follows:

"notifyMOICreation-NotifType": {

"type": "object",

"properties": {

"header": {

"$ref": "#/components/schemas/header-Type"

},

"body": {

"type": "object",

"properties": {

"correlatedNotifications": {

"type": "array",

"items": {

"$ref": "#/components/schemas/correlatedNotification-Type"

}

},

"additionalText": {

"$ref": "#/components/schemas/additionalText-Type"

},

"sourceIndicator": {

"$ref": "#/components/schemas/sourceIndicator-Type"

},

"attributeList": {

"type": "array",

"items": {

"$ref": "#/components/schemas/attributeNameValuePair-Type"

}

}

}

}

}

},

## 6.3 Conclusions on configuration management

The clause 6 of the present document studies the positioning of ONAP Controllers with regard to 3GPP management services for configuration management purposes. It also analyzes configuration management mechanisms in both ONAP Controllers (APPC) and 3GPP provisioning management services, from following perspectives:

- Configuration management: NETCONF protocol should be used for configuration management purposes. The Chef and Ansible protocols supported by ONAP controllers are primarily intended for LCM operations rather than CM operations and are not appropriate for configuration management on 5G network;

- Configuration management notification: is specified in 3GPP provisioning management services, but there is no existing domain field of the ONAP R3 VES JSON API that can be used to support or map to 3GPP configuration management notifications. One potential solution is enabling the DCAE R3 VES JSON Collector to consume 3GPP configuration management notifications.

It is recommended to start normative work on configuration management on the following aspects:

- Specifying NETCONF solution set support for configuration management purpose,

- Investigating potential solution for CM notification.

Annex <A>:  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2017-12 | SA5#116 | S5-176534  S5-176535 |  |  |  | Skeleton  Clause headers | 0.1.0 |
| 2018-01 |  |  |  |  |  | Editorial fixes (MCC) | 0.1.1 |
| 2018-02 | SA5#117 | S5-181518  S5-181519 |  |  |  | Positioning of ONAP/DCAE wrt. 3GPP reference management architecture  Communication principles for notifications | 0.2.0 |
| 2018-04 | SA5#118 | S5-182569 |  |  |  | Handling of performance data collection | 0.3.0 |
| 2018-05 | SA5#119 | S5-183175  S5-183598 |  |  |  | Re-introducing ONAP diagrams  Handling of alarm notifications | 0.4.0 |
| 2018-08 | SA5#120 | S5-185067  S5-185569  S5-185070  S5-185071  S5-185571  S5-185572  S5-185573  S5-185574  S5-185575  S5-185576  S5-185577 |  |  |  | Restructuring the TR to include CM aspects  Update clause 5 with 5G network management principles  Update clause 6.2 with 5G network management principles  Update clause 6.3 with 5G network management principles  Update clause 6.4 with 5G network management principles  Potential positioning of ONAP/DCAE wrt. for 3GPP management service  Correct ONAP architecture overview  Scope input  Update list of abbreviations  Add comparison of used technologies  Potential positioning of APPC wrt. for 3GPP management services | 0.5.0 |
| 2018-10 | SA5#121 | S5-186494  S5-186479  S5-186438  S5-186148  S5-186440  S5-186441  S5-186442  S5-186443  S5-186444 |  |  |  | pCR TR 28.900 No subscribe in ONAP  pCR TR 28.900 No PM job control in ONAP  pCR TR 28.900 FM-PM Data Reporting Methods  pCR TR 28.900 Correct the reference to configuration management related notifications  pCR 28.900 3GPP vs ONAP Interface technologies  pCR 28.900 Updated references  pCR 28.900 Updated architecture overview  pCR 28.900 ONAP DCAE VES Collector updates  pCR 28.900 Handling of alarm / event notifications | 0.6.0 |
| 2018-11 | SA5#122 | S5-187467  S5-187236  S5-187469  S5-187470  S5-187259  S5-187261  S5-187472  S5-187473  S5-187214  S5-187478  S5-187479 |  |  |  | Add example of 3GPP MnS Provider using interfaces provided by ONAP DCAE  Example of a notification header  Option for mapping 3GPP 5G alarm notifications on ONAP VES JSON Collector API  Mapping 3GPP 5G PM data reporting on ONAP VES JSON Collector API  ONAP Heartbeat and Event Throttling  3GPP Fault Supervision operations  Add example of 3GPP MnS Provider using interfaces provided by ONAP VF-C  ONAP controllers and 3GPP provisioning service for CM purposes  Update References for CM purposes  Add description for positioning  Comparative analysis for CM purposes | 0.7.0 |
| 2018-12 | SA#82 | SP-181062 |  |  |  | Presented for information | 1.0.0 |
| 2018-12 | SA#82 |  |  |  |  | EditHelp review | 1.0.1 |
| 2019-01 | SA#82 |  |  |  |  | Change of TR number (28.900 -> 28.890)  Change of TR title (“Telecommunication management; Study on integration of ONAP and 3GPP management for 5G networks” -> “Management and Orchestration; Study on integration of Open Network Automation Platform (ONAP) and 3GPP management for 5G networks” | 1.0.2 |
| 2019-01 | SA5#123 | S5-191486  S5-191436  S5-191125  S5-191433  S5-191434  S5-191129 |  |  |  | Editorial and Minor Technical Corrections  Conclusions on FM and PM  Align the term MnS and MnF  Update clause 5.2.3.3.3 Reporting 3GPP Rel-15 PM data to DCAE R3 VES JSON Collector  Alarm Life Cycle management in ONAP  Replace 3GPP managment system with 3GPP MnF | 1.1.0 |
| 2019-03 | SA5#124 | S5-192073  S5-192135  S5-192391  S5-192392  S5-192393  S5-192394  S5-192395  S5-192443 |  |  |  | Heartbeat and Communication Surveillance  Update clause 6.2.2 management principles  Add Handling of configuration notification  Minor Technical Corrections  Add the description of 3GPP provisioning management service  Add the description of SDNC  Configuration Management handling  Conclusion on CM | 1.2.0 |
| 2019-03 | SA#83 | SP-190136 |  |  |  | Presented for approval | 2.0.0 |
| 2019-03 | SA#83 |  |  |  |  | Upgrade to change control version | 16.0.0 |