ETSI GS NFV-IFA 027 V5.4.1 (2026-02)

GROUP SPECIFICATION

Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Performance Measurements Specification

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
RGS/NFV-IFA027ed541

Keywords
management, MANO, measurement, NFV,
performance

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

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

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights 6
Foreword 6
Modal verbs terminology 6
1 Scope 7
2 References 7
2.1 Normative references 7
2.2 Informative references 8
3 Definition of terms, symbols and abbreviations 9
3.1 Terms 9
3.2 Symbols 9
3.3 Abbreviations 9
4 Overview 9
4.1 Introduction 9
4.2 Use cases 10
4.2.1 Use case of virtual compute related measurements 10
4.2.2 Use case of network data volume related measurements 10
4.2.3 Use case of physical resource related measurements 10
4.3 Measurements received at VIM 10
4.4 Measurements received at PIM 11
5 Performance measurement definition template 11
6 Measured object types 13
6.1 Introduction 13
6.2 Measured object type definitions 13
6.2.1 Virtual compute 13
6.2.2 VNF 13
6.2.3 VNF component 14
6.2.4 Virtual network 14
6.2.5 Virtual storage 14
6.2.6 Internal connection point of a VNF instance 14
6.2.7 External connection point of a VNF instance 14
6.2.8 Service access point 14
6.2.9 CIS cluster node 15
6.2.10 MCIO-C 15
6.2.11 NS 15
6.2.12 Physical Resource 15
7 Performance measurements 16
7.1 Performance measurements associated to virtualised resources 16
7.1.1 Introduction 16
7.1.2 Mean virtual CPU usage 16
7.1.3 Peak virtual CPU usage 16
7.1.4 Mean memory usage 17
7.1.5 Peak memory usage 18
7.1.6 Mean disk usage 18
7.1.7 Peak disk usage 18
7.1.8 Number of incoming bytes on virtual compute 19
7.1.9 Number of outgoing bytes on virtual compute 19
7.1.10 Number of incoming packets on virtual compute 20
7.1.11 Number of outgoing packets on virtual compute 21
7.1.12 Number of incoming bytes of a virtual network 21
7.1.13 Number of outgoing bytes of a virtual network 22
7.1.14 Number of incoming packets of a virtual network 22
7.1.15 Number of outgoing packets of a virtual network 23

7.1.16 Mean usage of Virtualised storage resource 23
7.1.17 Peak usage of Virtualised storage resource....24
7.1.18 Energy consumption of a virtual compute 24
7.1.19 Mean energy consumption of a virtual compute....25
7.2 Performance measurements associated to VNF instances....25
7.2.1 Introduction....25
7.2.2 Mean virtual CPU usage of VNF/VNFC instance 26
7.2.3 Peak virtual CPU usage of VNF/VNFC instance 27
7.2.4 Mean memory usage of VNF/VNFC instance....28
7.2.5 Peak memory usage of VNF/VNFC instance 29
7.2.6 Mean disk usage of VNF/VNFC instance....30
7.2.7 Peak disk usage of VNF/VNFC instance....31
7.2.8 Number of incoming bytes of VNF internal CP 32
7.2.9 Number of outgoing bytes of VNF internal CP 33
7.2.10 Number of incoming packets of VNF internal CP....34
7.2.11 Number of outgoing packets of VNF internal CP....34
7.2.12 Number of incoming bytes of VNF external CP....35
7.2.13 Number of outgoing bytes of VNF external CP....36
7.2.14 Number of incoming packets of VNF external CP 37
7.2.15 Number of outgoing packets of VNF external CP....38
7.2.16 Energy consumption of VNFC instance 39
7.2.17 Mean energy consumption of VNFC instance....40
7.2.18 Energy consumption of VNF instance....41
7.2.19 Mean energy consumption of VNF instance....42
7.3 Performance measurements associated to NS instances....43
7.3.1 Introduction....43
7.3.2 Number of incoming bytes of SAP....43
7.3.3 Number of outgoing bytes of SAP....44
7.3.4 Number of incoming packets of SAP 45
7.3.5 Number of outgoing packets of SAP 46
7.3.6 Energy consumption of NS instance....46
7.3.7 Mean energy consumption of NS instance 47
7.4 Performance measurements associated to OS container workloads....48
7.4.1 Introduction....48
7.4.2 Mean CPU usage on OS container workload....48
7.4.3 Peak CPU usage on OS container workload....48
7.4.4 Mean memory usage on OS container workload 49
7.4.5 Peak memory usage on OS container workload 49
7.4.6 Number of incoming bytes on compute MCIO....50
7.4.7 Number of outgoing bytes on compute MCIO 50
7.4.8 Number of incoming packets on compute MCIO....51
7.4.9 Number of outgoing packets on compute MCIO....51
7.4.10 Mean usage of storage resource on OS container workload....52
7.4.11 Peak usage of storage resource on OS container workload 52
7.4.12 Energy consumption on OS container workload 53
7.4.13 Mean energy consumption on OS container workload....53
7.5 Performance measurements associated to CIS cluster....53
7.5.1 Introduction....53
7.5.2 Mean CPU usage on CIS cluster node....54
7.5.3 Peak CPU usage on CIS cluster node 54
7.5.4 Mean memory usage on CIS cluster node 55
7.5.5 Peak memory usage on CIS cluster node....55
7.5.6 Number of incoming bytes of a network on CIS cluster node....56
7.5.7 Number of outgoing bytes of a network on CIS cluster node....56
7.5.8 Number of incoming packets of a network on CIS cluster node 56
7.5.9 Number of outgoing packets of a network on CIS cluster node 57
7.5.10 Mean usage of storage resource on CIS cluster node 57
7.5.11 Peak usage of storage resource on CIS cluster node....58
7.6 Performance measurements associated to physical resources....58
7.6.1 Introduction....58
7.6.2 Mean physical CPU usage 58
7.6.3 Peak physical CPU usage 59

7.6.4 Mean memory usage 59
7.6.5 Peak memory usage 60
7.6.6 Mean disk usage 60
7.6.7 Peak disk usage 61
7.6.8 Number of incoming bytes on physical resource 61
7.6.9 Number of outgoing bytes on physical resource 62
7.6.10 Number of incoming packets on physical resource 63
7.6.11 Number of outgoing packets on physical resource 63
7.6.12 Mean usage of physical storage resource 64
7.6.13 Peak usage of physical storage resource 65
7.6.14 Energy consumption of a physical resource 65
7.6.15 Mean energy consumption of a physical resource 66

8 Key performance indicators 68
8.1 Key performance indicators associated to virtualised resources 68
8.1.1 Introduction 68
8.1.2 Energy efficiency of virtualised compute based on data volume 68
8.2 Key performance indicators associated to OS container workloads 69
8.2.1 Introduction 69
8.2.2 Energy efficiency of OS container workload based on data volume 69
8.3 Key performance indicators associated to VNF instances 70
8.3.1 Introduction 70
8.3.2 Energy efficiency of VNF based on data volume 70
8.3.3 Energy efficiency of VNFC based on data volume 70
8.4 Key performance indicators associated to NS instances 71
8.4.1 Introduction 71
8.4.2 Energy efficiency of NS based on data volume 71
8.5 Key performance indicators associated with physical resources 72
8.5.1 Introduction 72
8.5.2 Energy efficiency of a physical network resource based on data volume 72
8.5.3 Energy efficiency of a physical storage resource based on IOPS performance 73
8.5.4 Energy efficiency of a physical storage resource based on throughput performance 73
8.5.5 Energy efficiency of a physical storage resource based on storage capacity 74
8.5.6 Energy efficiency of a composed physical resource in active state 75

Annex A (informative): Mapping of ETSI GS NFV-TST 008 to OpenStack® measurements 76

Annex A1 (informative): Kubernetes® resource management and monitoring 77

A1.0 Introduction 77

A1.1 Resource Management for Pods and Containers 77
A1.1.1 Requests and limits 77
A1.1.2 Resource units in Kubernetes® 77

A1.2 Monitoring resource usage 78
A1.2.1 Measuring resource usage 78
A1.2.2 Monitoring and reporting resource usage 78
A1.2.3 Examples of getting resource usage 79

Annex A2 (informative): Mapping of PIM metrics to Redfish® and OpenStack® metrics 83

A2.1 PIM metrics and corresponding Redfish® metrics 83

A2.2 PIM metrics and corresponding OpenStack® measurements 84

Annex B (informative): Security and Regulatory Concerns 85

B.1 Risk analysis and assessment 85

Annex C (normative): Other definition templates 87

C.1 KPI definition template 87

Annex D (informative): Change history 88

History 90

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies the performance measurements associated to NFV-MANO and its managed objects. The present document introduces relevant use cases and identifies the usage of the performance measurements in descriptors and on interfaces, including Or-Vnfm reference point, Ve-Vnfm reference point, Vi-Vnfm reference point, Or-Vi reference point, Os-Ma-nfvo reference point, and service interfaces produced by Container Infrastructure Service Management (CISM) function, Container Cluster Management (CCM) function and Physical Infrastructure Management (PIM) function, based on the performance metrics collected at NFVI. Clause 5 specifies the template to be used for defining performance measurements.

The present document also specifies Key Performance Indicators (KPI) associated to NFV-MANO and its managed objects. Annex C specifies the template to be used for defining KPIs.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS NFV-IFA 005: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification".

[2] ETSI GS NFV-IFA 006: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification".

[3] ETSI GS NFV-IFA 008: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

[4] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; VNF Descriptor and Packaging Specification".

[5] Void.

[6] Void.

[7] Void.

[8] Void.

[9] Void.

[10] ETSI GS NFV-TST 008: "Network Functions Virtualisation (NFV) Release 3; Testing; NFVI Compute and Network Metrics Specification".

[11] ETSI GS NFV-IFA 040: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification".

[12] ETSI GS NFV-IFA 036: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements for service interfaces and object model for container cluster management and orchestration specification".

[13] ETSI GS NFV-IFA 053: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements and interface specification for Physical Infrastructure Management".

[14] ETSI EN 303 470 (V1.1.1): "Environmental Engineering (EE); Energy Efficiency measurement methodology and metrics for servers".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] Openstack® measurements.

NOTE 1: The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. ETSI is not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

NOTE 2: https://www.openstack.org/brand/openstack-trademark-policy/.

[i.2] ETSI GR NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.3] Kubernetes® document: "Resource Management for Pods and Containers".

[i.4] Kubernetes® document: "Resource metrics pipeline".

[i.5] Kubernetes® document: "Document stable metrics list".

[i.6] Kubernetes® document: "Tools for Monitoring Resources".

[i.7] Kubernetes® type definition: "Top-level container for holding NodeStats and PodStats".

[i.8] Linux® Networking Documentation: "Interface statistics".

NOTE: Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

[i.9] ETSI GS NFV-IFA 045: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Faults and alarms modelling specification".

[i.10] DMTF® DSP0268: "Redfish Data Model Specification".

[i.11] ETSI GS NFV-SEC 004: "Network Functions Virtualisation (NFV); NFV Security; Privacy and Regulation; Report on Lawful Interception Implications".

[i.12] ETSI GS NFV-SEC 006: "Network Functions Virtualisation (NFV); Security Guide; Report on Security Aspects and Regulatory Concerns".

[i.13] ETSI GS NFV-SEC 010: "Network Functions Virtualisation (NFV); NFV Security; Report on Retained Data problem statement and requirements".

[i.14] ETSI GS NFV-SEC 012: "Network Functions Virtualisation (NFV) Release 5; Security; System architecture specification for execution of sensitive NFV components".

[i.15] ETSI EN 303 804 (V1.1.1): "Environmental Engineering (EE); Energy efficiency metrics and measurement methods for data storage equipment".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GR NFV 003 [i.2] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR NFV 003 [i.2] and the following apply:

NOTE: An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in ETSI GR NFV 003 [i.2].

BMC	Baseboard Management Controller
IOPS	Input/Output operations Per Second
IPMI	Intelligent Platform Management Interface
OM	Object Mapping
SC	Status Counter
SERT	Server Efficiency Rating Tool
TF	Transparent Forwarding

4 Overview

4.1 Introduction

The present document specifies the performance measurements associated to NFV-MANO and its managed objects. The performance measurements are exposed on various NFV-MANO reference points (e.g. Or-Vnfm, Ve-Vnfm-em, Vi-Vnfm, Or-Vi and Os-Ma-nfvo) and produced by respective NFV-MANO functional blocks, and on service interfaces produced by other NFV-MANO functions (e.g. CISM and CCM). The performance measurements can also be produced by other management functions or services not referenced in the present document.

As an example, in the context of an NFV-MANO framework, ETSI GS NFV-TST 008 [10] specifies the Network Function Virtualisation Infrastructure (NFVI) performance metrics. These can be reported to Virtualised Infrastructure Manager (VIM). Then, the VIM processes the performance metrics received from NFVI to generate performance measurements to be sent to NFV Orchestrator (NFVO) and VNF Manager (VNFM). VNFM and NFVO conduct further processing to generate performance measurements. In a similar manner, CISM collects and processes the performance metrics about OS container workload received from CIS to generate performance measurements to be sent to NFVO and VNFM; and CCM collects and processes the performance metrics about CIS cluster to generate performance measurements to be sent to NFVO. Similarly, PIM receives performance metrics related to physical resources from the NFVI, processes them to generate performance measurements that can be made available to other management entities, e.g. VIM, NFVO, CCM.

NOTE: The present document does not specify the path (e.g. from which Kubernetes® function or time series database) by which the CISM can get performance metrics about OS container workload. In general, performance monitoring data flow can take the similar flow of monitoring and generation of alarms concerning container-based VNFC/VNF as described in Figure A.1.2.3-1 in ETSI GS NFV-IFA 045 [i.9].

4.2 Use cases

4.2.1 Use case of virtual compute related measurements

Virtual compute related measurements are used to monitor the loading conditions of virtual compute resources that include memory and CPU. Monitoring parameters associated with the compute related measurements and of interest for a VNF can be declared in the VNFD (see ETSI GS NFV-IFA 011 [4]). The measurements can be used as the triggers to the scale VNF operation (see clause 7.2.4 in ETSI GS NFV-IFA 008 [3]).

When the measurement indicated that the virtualised resources are overloaded, it can trigger the scale out operation. When the measurement indicated that the virtualised resources are underloaded, it can trigger the scale in operation.

4.2.2 Use case of network data volume related measurements

Network data volume related measurements are used to measure the data volume of networking interfaces (see clause 7.1 in ETSI GS NFV-TST 008 [10]) including incoming/outgoing of IP data packets and IP data octets. These measurements provide the traffic condition of the network interfaces. When the traffic condition is too high, then it may request the application to take appropriate action.

4.2.3 Use case of physical resource related measurements

Performance measurements related to physical resources in the NFVI include performance metrics related to physical compute, storage, and networking resources as well as power and energy consumption information associated with different kinds of physical resources in the NFVI.

Power consumption and energy related measurements of physical resources can lead to energy efficient design and planning of NFVI resources to be utilized for VNF/NS deployments. Collected measurements data can help determine the optimal placement of VNFs/NSs, delivering power consumption savings according to the energy efficiency policies.

4.3 Measurements received at VIM

Table 4.3-1 lists the measurements received at VIM to be used to define the performance measurements in clause 7. The measurements are derived from the performance metrics defined in ETSI GS NFV-TST 008 [10].

Table 4.3-1: Measurements received at VIM

Measurements received at VIM	Description
cpu_utilization	Measure the percentage of CPU utilization in the measurement interval that can be derived from Processor Utilization metric (see clause 6.6 in ETSI GS NFV-TST 008 [10]).
memory_utilization	Measure the percentage of memory utilization in the measurement interval that can be derived from the amount of memory used, which is the sum of Memory Buffered, Memory Cached, Memory Free, Memory Slab (see clause 8.6 in ETSI GS NFV-TST 008 [10]), and the amount of memory allocated.
disk_utilization	Measure the percentage of disk utilization. See Annex A for an example of the measurement mapping from disk.usage and disk.allocation measurements. No disk metric is defined in ETSI GS NFV-TST 008 [10], since the methods of measurement for storage systems vary widely and depend on the implementation (see clause 8.9 in ETSI GS NFV-TST 008 [10]).
num_of_incoming_packets	Measure the number of incoming packets in the measurement interval that can be derived from received Packet Count metric (see clause 7.6 in ETSI GS NFV-TST 008 [10]).
num_of_outgoing_packets	Measure the number of outgoing packets in the measurement interval that can be derived from transmitted Packet Count metric (see clause 7.6 in ETSI GS NFV-TST 008 [10]).
num_of_incoming_bytes	Measure the number of incoming octets in the measurement interval that can be derived from received Octet Count metric (see clause 7.6 in ETSI GS NFV-TST 008 [10]).

Measurements received at VIM	Description
num_of_outgoing_bytes	Measure the number of outgoing octets in the measurement interval that can be derived from transmitted Octet Count metric (see clause 7.6 in ETSI GS NFV-TST 008 [10]).

4.4 Measurements received at PIM

Table 4.4-1 lists the measurements and metrics received at PIM that are used to define the relevant performance measurements related to physical resources in clause 7. These measurements and metrics are derived from the performance metrics defined in ETSI GS NFV-TST 008 [10] and additional power consumption related metrics that the PIM can receive by interacting with the BMCs of relevant physical resources. Annex A2 provides a mapping of these PIM related metrics with corresponding metrics and measurements supported by Redfish® [i.10] and OpenStack® [i.1].

Table 4.4-1: Measurements received at PIM

Measurements received at PIM	Description
phy_cpu_utilization	Measure the percentage of CPU utilization in the measurement interval that can be derived from Processor Utilization metric (see clause 6.6 in ETSI GS NFV-TST 008 [10]). See note 1.
phy_memory_utilization	Measure the percentage of memory utilization in the measurement interval that can be derived from the amount of memory used, which is the sum of Memory Buffered, Memory Cached, Memory Free, Memory Slab (see clause 8.6 in ETSI GS NFV-TST 008 [10]), and the amount of total memory. See note 2.
phy_storage_utilization	Measure the percentage of system storage utilization in the measurement interval that can be derived from the amount of storage used, which is indicated by 'Used Space' metric (see clause 9.6 of ETSI GS NFV-TST 008 [10]) and the total storage size (as a sum of Used Space, Free Space and Reserved Space values). See note 3.
num_of_incoming_packets	Measure the number of incoming packets in the measurement interval that can be derived from received Packet Count metric (see clause 7.6 in ETSI GS NFV-TST 008 [10]). See note 4.
num_of_outgoing_packets	Measure the number of outgoing packets in the measurement interval that can be derived from transmitted Packet Count metric (see clause 7.6 in ETSI GS NFV-TST 008 [10]). See note 4.
num_of_incoming_bytes	Measure the number of incoming octets in the measurement interval that can be derived from received Octet Count metric (see clause 7.6 in ETSI GS NFV-TST 008 [10]). See note 4.
num_of_outgoing_bytes	Measure the number of outgoing octets in the measurement interval that can be derived from transmitted Octet Count metric (see clause 7.6 in ETSI GS NFV-TST 008 [10]). See note 4.
power_consumption	Measure the power consumption of a physical resource in the measurement interval.
energy_consumption	Measure the energy consumption of a physical resource in the measurement interval.
phy_storage_iops	Measure the total number of read and write input/output operations per second that are performed by a physical resource in a measurement interval. This measurement is the sum of 'Read IOPS' and 'Write IOPS' metrics (see clause 9.6 of ETSI GS NFV-TST 008 [10]). See note 3.
phy_storage_throughput	Measure the total number of bytes that are transferred per second by a physical resource in a measurement interval. This measurement is the sum of 'Read Throughput' and 'Write Throughput' metrics (see clause 9.6 of ETSI GS NFV-TST 008 [10]). See note 3.
NOTE 1: The scope of reported values indicates that the metrics are associated with physical processor. NOTE 2: The scope of reported values indicates that the metrics are associated with physical memory. NOTE 3: The scope of reported values indicates that the metrics are associated with physical storage device. NOTE 4: The scope of reported values indicates that the metrics are associated with physical network device. A physical network device can refer to e.g. a network adapter, network interface and/or a port.	

5 Performance measurement definition template

This clause defines the template to be used to describe the performance measurements.

a) **Description**

This clause contains the description of the performance measurement.

b) **Collection Method**

This clause contains the method in which this measurement is obtained:

- **Status Counter (SC):** The entity receives a metric at each predetermined interval. A measurement is generated from processing (e.g. arithmetic mean, peak) all of the samples received in the collection period (see clause 7.7.2.2 in ETSI GS NFV-IFA 006 [2]).
- **Transparent Forwarding (TF):** The entity maintains a measurement count that stores the content of the metric that it received.
- **Object Mapping (OM):** The entity receives a metric for measured object A in the collection period and maps the received metric from measured object A to measured object B. A measurement is generated for measured object B by processing the metric(s), which may be mapped from one or more measured object(s) A to a single measured object B. It is noted that:
  - The source metric for measured object A and the target measurement for measured object B may or may not contain subcounters. How the mapping is done for the case that either of the source metric and target measurement contain subcounters is to be defined case by case in the trigger of the measurement definition.
  - Multiple source metrics for measured object A may be mapped to a single target measurement for measured object B. How the mapping is done for this case is to be defined in the trigger of the measurement definition.

c) **Trigger**

This clause contains the trigger which causes the counter to be updated.

d) **Measurement Unit**

This clause contains the unit of the measurement value.

e) **Measurement Group**

This clause contains the group to which a measurement belongs.

f) **Measured Object Type**

This clause describes the object of a measurement. See clause 6 for the measured object types defined for the performance measurements specified in the present document.

g) **Measurement Name**

This clause describes the name of a measurement:

- The measurement name is used to identify a measurement. In case the sub-counter is used, the measurement is identified by <measurement type>.<sub-counter name>.
- The measurement name is used to identify the performanceMetric in the performance report (e.g. see clause 9.7.6 of ETSI GS NFV-IFA 008 [3]).
- In the create PM Job operation, the measurement type is the content of performanceMetric to identify the type of measurement(s) to be collected (e.g. see clause 7.4.2.2 of ETSI GS NFV-IFA 008 [3]). The PM Job is applicable to the sub-counters, if the measurement contains sub-counters.
- Examples of measurement names without sub-counters are:
  - VCpuUsageMean;
  - VCpuUsagePeak.

- Examples of valid measurement names with sub-counters are:
  - VCpuUsageMeanVnf.vCompute1:
    - wherein the "vCompute1" is the value of *computeId* of the *VirtualCompute* Instance (see clause 8.4.3.2.2 of ETSI GS NFV-IFA 006 [2]).
  - VCpuUsageMeanVnf.vCompute2:
    - wherein the "vCompute2" is the value of *computeId* of the *VirtualCompute* Instance (see clause 8.4.3.2.2 of ETSI GS NFV-IFA 006 [2]).

h) **Measurement Context**

This clause describes the context information of a measurement at the time that the measurement is generated, see ETSI GS NFV-TST 008 [10] for more information. The measurement context provides additional information (than performance name and value) which could facilitate the measurement consumer to do the tasks that rely on more detailed information associated with the performance measurement, such as trouble shooting, VNF/VNFC scaling, etc.

The measurement context is only provided in case the measurement producer has knowledge about the context information.

Each measurement may have its specific context, so the detailed measurement context is defined in each measurement definition.

6 Measured object types

6.1 Introduction

This clause defines the measured object types for the performance measurements specified in the present document. Depending on the measurement type, the related measured object can be structured or unstructured. A measurement either relates to an object identified by a combination of object instance identifier and sub-object instance identifier (in case of a structured measured object), or to an object identified by an object instance identifier (in case of an unstructured measured object).

6.2 Measured object type definitions

6.2.1 Virtual compute

The measured object type "VirtualCompute" is used to collect and report the performance measurements for one or more instances of the Virtualised compute resource.

The *objectType*, when used in PM job or performance report, is equal to "VirtualCompute".

The *objectInstanceId*, when used in PM job or performance report, corresponds to *computeId* (see clause 8.4.3.2.2 of ETSI GS NFV-IFA 006 [2] or clause 8.4.3.2.2 of ETSI GS NFV-IFA 005 [1]) of the measured Virtualised compute resource.

6.2.2 VNF

The measured object type "Vnf" is used to collect and report the performance measurements for one or more VNF instances.

The *objectType*, when used in PM job or performance report, is equal to "Vnf".

The *objectInstanceId*, when used in PM job or performance report, corresponds to *vnfInstanceId* that identifies the measured VNF instance.

6.2.3 VNF component

The measured object type "Vnfc" is used to collect and report the performance measurements for one or more VNFC instances.

The *objectType*, when used in PM job or performance report, is equal to "Vnfc".

The *objectInstanceId*, when used in PM job or performance report, corresponds to the *vnfInstanceId* of the VNF instance that contains the measured VNFC instance. The *subObjectInstanceId*, when used in PM job or performance report, corresponds to the *vnfcInstanceId* that identifies the measured VNFC instance.

6.2.4 Virtual network

The measured object type "VirtualNetwork" is used to collect and report the performance measurements for one or more instances of the virtual network.

The *objectType*, when used in PM job or performance report, is equal to "VirtualNetwork".

The *objectInstanceId*, when used in PM job or performance report, corresponds to *networkResourceId* (see clause 8.4.5.2 of ETSI GS NFV-IFA 006 [2] or clause 8.4.5.2 of ETSI GS NFV-IFA 005 [1]) of the measured virtual network instance.

6.2.5 Virtual storage

The measured object type "VirtualStorage" is used to collect and report the performance measurements for one or more instances of the Virtualised storage resource.

The *objectType*, when used in PM job or performance report, is equal to "VirtualStorage".

The *objectInstanceId*, when used in PM job or performance report, corresponds to *storageId* (see clause 8.4.7.2 of ETSI GS NFV-IFA 006 [2] or clause 8.4.7.2 of ETSI GS NFV-IFA 005 [1]) of the measured Virtualised storage resource.

6.2.6 Internal connection point of a VNF instance

The measured object type "VnfIntCp" is used to collect and report the performance measurements for one or more instances of the VNF internal CP.

The *objectType*, when used in PM job or performance report, is equal to "VnfIntCp".

The *objectInstanceId*, when used in PM job or performance report, corresponds to the *vnfInstanceId* of the VNF instance which the VNF internal CP belongs to. The *subObjectInstanceId*, when used in PM job or performance report, corresponds to the instance identifier of the measured VNF internal CP instance.

6.2.7 External connection point of a VNF instance

The measured object type "VnfExtCp" is used to collect and report the performance measurements for one or more instances of the VNF external CP.

The *objectType*, when used in PM job or performance report, is equal to "VnfExtCp".

The *objectInstanceId*, when used in PM job or performance report, corresponds to the *vnfInstanceId* of the VNF instance that exposes the measured VNF external CP instance. The *subObjectInstanceId*, when used in PM job or performance report, corresponds to the instance identifier of the measured VNF external CP instance.

6.2.8 Service access point

The measured object type "Sap" is used to collect and report the performance measurements for one or more SAP instances of an NS instance.

The *objectType*, when used in PM job or performance report, is equal to "Sap".

The *objectInstanceId*, when used in PM job or performance report, corresponds to the*NsInstanceId* of the NS instance that exposes the measured SAP. The *subObjectInstanceId*, when used in PM job or performance report, corresponds to the instance identifier of the measured SAP instance.

6.2.9 CIS cluster node

The measured object type "CisClusterNode" is used to collect and report the performance measurements for one or more CIS cluster nodes.

The objectType, when used in PM job or performance report, is equal to "CisClusterNode".

The objectInstanceId, when used in PM job or performance report, corresponds to cisClusterNodeId (see clause 4.2.4 of ETSI GS NFV-IFA 036 [12]) that identifies the measured CIS cluster node.

6.2.10 MCIO-C

The measured object type "Mcio-c" is used to collect and report the performance measurements for one or more instances of a Compute MCIO (see clause 5.2 of ETSI GS NFV-IFA 040 [11]) corresponding to a group of one or more OS containers.

NOTE: Pod is an example of Compute MCIO corresponding to a group of one or more OS containers in a Kubernetes® environment.

The objectType, when used in PM job or performance report, is equal to "Mcio-c".

The objectInstanceId, when used in PM job or performance report, corresponds to "ID of Compute MCIO" mapped to the resource handle's "resourceId" (see clause 9.4.7 of ETSI GS NFV-IFA 008 [3]) of the measured compute MCIO.

6.2.11 NS

The measured object type "Ns" is used to collect and report the performance measurements for one or more NS instances.

The objectType, when used in PM job or performance report, is equal to "Ns".

The objectInstanceId, when used in PM job or performance report, corresponds to the NsInstanceId that identifies the measured NS instance.

6.2.12 Physical Resource

The measured object type "PhysicalResource" is used to collect and report the performance measurements for one or more physical resources.

The objectType, when used in PM job or performance report, is equal to "PhysicalResource".

The objectInstanceId, when used in PM job or performance report, corresponds to the unique resource identifier that identifies the measured physical resource (see clause 8.3.5 of ETSI GS NFV-IFA 053 [13]).

7 Performance measurements

7.1 Performance measurements associated to virtualised resources

7.1.1 Introduction

The performance measurements defined in this clause 7.1 are applicable to virtualised resources and they can be exposed, among others, on interfaces defined on the following reference points:

- Vi-Vnfm;
- Or-Vi (for indirect mode of resource allocation, in case indirect mode is supported).

The measurement producer can be, among others, a VIM.

7.1.2 Mean virtual CPU usage

a) **Description:** This measurement provides the mean virtual CPU usage of the Virtualised compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the cpu_utilization measurement for the virtual compute instance from NFVI at the pre-defined interval, and then takes the arithmetic mean of the virtual CPU usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VCpuUsageMean.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last cpu_utilization measurement received from NFVI in the collection period.
- **TickInterval:** Indicates the tick interval (see ETSI GS NFV-TST 008 [10]) of the last cpu_utilization measurement received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of cpu_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the cpu_utilization measurements" received from NFVI in the collection period.
- **ExecutionContext:** Indicates the execution context (see ETSI GS NFV-TST 008 [10]) of the cpu_utilization measurements received from NFVI in the collection period. When a single "non-Idle" VCPU Utilization is required, the sum of the utilization of all the non-Idle execution contexts should be reported as the "active" execution context.

7.1.3 Peak virtual CPU usage

a) **Description:** This measurement provides the peak virtual CPU usage of the Virtualised compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the cpu_utilization measurement for the virtual compute instance from NFVI at the pre-defined interval, and then selects the maximum metric among the virtual CPU usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VCpuUsagePeak.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the cpu_utilization measurement with peak value received from NFVI in the collection period.
- **TickInterval:** Indicates the tick interval (see ETSI GS NFV-TST 008 [10]) of the cpu_utilization measurement with peak value received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of cpu_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the cpu_utilization measurements" received from NFVI in the collection period.
- **Execution Context:** Indicates the execution context (see ETSI GS NFV-TST 008 [10]) of the cpu_utilization measurement with peak value received from NFVI in the collection period. When a single "non-Idle" VCPU Utilization is required, the sum of the utilization of all the non-Idle execution contexts should be reported as the "active" execution context.

7.1.4 Mean memory usage

a) **Description:** This measurement provides the mean memory usage of the Virtualised compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the memory_utilization measurement for the virtual compute instance from NFVI at the pre-defined interval, and then takes the arithmetic mean of the memory usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VMemoryUsageMean.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last memory_utilization measurement received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of memory_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the memory_utilization measurements" received from NFVI in the collection period.
- **MeasurementSystemRam:** Indicates the system RAM (see ETSI GS NFV-TST 008 [10]) of the measured Virtualised compute resource.
- **MeasurementSystemSwapSpace:** Indicates the system SWAP space (see ETSI GS NFV-TST 008 [10]) of the measured Virtualised compute resource.

7.1.5 Peak memory usage

a) **Description:** This measurement provides the peak memory usage of the Virtualised compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the memory_utilization measurement for the virtual compute instance from NFVI at the pre-defined interval, and then selects the maximum metric among the memory usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VMemoryUsagePeak.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the memory_utilization measurement with peak value received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of memory_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the memory_utilization measurements" received from NFVI in the collection period.
- **MeasurementSystemRam:** Indicates the system RAM (see ETSI GS NFV-TST 008 [10]) of the measured Virtualised compute resource.
- **MeasurementSystemSwapSpace:** Indicates the system SWAP space (see ETSI GS NFV-TST 008 [10]) of the measured Virtualised compute resource.

7.1.6 Mean disk usage

a) **Description:** This measurement provides the mean disk usage of the Virtualised compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the disk_utilization measurement for the virtual compute instance from NFVI at the pre-defined interval, and then takes the arithmetic mean of the disk usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VDiskUsageMean.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last disk_utilization measurement received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of disk_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the disk_utilization measurements" received from NFVI in the collection period.

7.1.7 Peak disk usage

a) **Description:** This measurement provides the peak disk usage of the Virtualised Compute Resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the disk_utilization measurement for the virtual compute instance from NFVI at the pre-defined interval, and then selects the maximum metric among the disk usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VDiskUsagePeak.

h) **Measurement Context:**

- **MeasurementEndTime:** It indicates the end time (see ETSI GS NFV-TST 008 [10]) of the disk_utilization measurement with peak value received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of disk_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the disk_utilization measurements" received from NFVI in the collection period.

7.1.8 Number of incoming bytes on virtual compute

a) **Description:** This measurement provides the number of bytes received at the virtual compute. This measurement is split into subcounters per virtual network interface, which is the communication endpoint under an instantiated compute resource.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more num_of_incoming_bytes measurement for a virtual network interface from NFVI in the collection period, and maps the received num_of_incoming_bytes measurement(s) from the virtual network interface to the Virtual Compute instance. The measurement producer generates the measurement for the subject Virtual Compute instance by assigning the value of the received num_of_incoming_bytes measurement(s) to the sub-counter(s) per virtual network interface.

d) **Measurement Unit:** Each measurement is a real value.

e) **Measurement Group:** VirtualNetworkInterface.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VNetByteIncoming.*vNetItfId*, where *vNetItfId* is equal to the resourceId of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last num_of_incoming_bytes measurement received from NFVI for the measured virtual network interface in the collection period.
- **MeasurementInterval:** Equals the "number of num_of_incoming_bytes measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the num_of_incoming_bytes measurements" received from NFVI for the measured virtual network interface in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last num_of_incoming_bytes measurement received from NFVI for the measured virtual network interface in the collection period is generated.

7.1.9 Number of outgoing bytes on virtual compute

a) **Description:** This measurement provides the number of bytes transmitted at the virtual compute. This measurement is split into subcounters per virtual network interface, which is the communication endpoint under an instantiated compute resource.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more num_of_outgoing_bytes measurement for a virtual network interface from NFVI in the collection period, and maps the received num_of_outgoing_bytes measurement(s) from the virtual network interface to the Virtual Compute instance. The measurement producer generates the measurement for the subject Virtual Compute instance by assigning the value of the received num_of_outgoing_bytes measurement(s) to the sub-counter(s) per virtual network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VirtualNetworkInterface.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VNetByteOutgoing.*vNetItfId*, where *vNetItfId* is equal to the resourceId of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last num_of_outgoing_bytes measurement received from NFVI for the measured virtual network interface in the collection period.
- **MeasurementInterval:** Equals the "number of num_of_outgoing_bytes measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the num_of_outgoing_bytes measurements" received from NFVI for the measured virtual network interface in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last num_of_outgoing_bytes measurement received from NFVI for the measured virtual network interface in the collection period is generated.

7.1.10 Number of incoming packets on virtual compute

a) **Description:** This measurement provides the number of packets received at the virtual compute. This measurement is split into subcounters per virtual network interface, which is the communication endpoint under an instantiated compute resource.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more num_of_incoming_packets measurement for a virtual network interface from NFVI in the collection period, and maps the received num_of_incoming_packets measurement(s) from the virtual network interface to the Virtual Compute instance. The measurement producer generates the measurement for the subject Virtual Compute instance by assigning the value of the received num_of_incoming_packets measurement(s) to the sub-counter(s) per virtual network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VirtualNetworkInterface.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VNetPacketIncoming.*vNetItfId*, where *vNetItfId* is equal to the resourceId of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last num_of_incoming_packets measurement received from NFVI for the measured virtual network interface in the collection period.
- **MeasurementInterval:** Equals the "number of num_of_incoming_packets measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the num_of_incoming_packets measurements" received from NFVI for the measured virtual network interface in the collection period.

- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last num_of_incoming_packets measurement received from NFVI for the measured virtual network interface in the collection period is generated.

7.1.11 Number of outgoing packets on virtual compute

a) **Description:** This measurement provides the number of packets transmitted at the virtual compute. This measurement is split into subcounters per virtual network interface, which is the communication endpoint under an instantiated compute resource.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more num_of_outgoing_packets measurement for a virtual network interface from NFVI in the collection period, and maps the received num_of_outgoing_packets measurement(s) from the virtual network interface to the Virtual Compute instance. The measurement producer generates the measurement for the subject Virtual Compute instance by assigning the value of the received num_of_outgoing_packets measurement(s) to the sub-counter(s) per virtual network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VirtualNetworkInterface.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** VNetPacketOutgoing.*vNetItfId*, where *vNetItfId* is equal to the resourceId of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last num_of_outgoing_packets measurement received from NFVI for the measured virtual network interface in the collection period.

- **MeasurementInterval:** Equals the "number of num_of_outgoing_packets measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of num_of_outgoing_packets measurements" received from NFVI for the measured virtual network interface in the collection period.

- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last num_of_outgoing_packets measurement received from NFVI for the measured virtual network interface in the collection period is generated.

7.1.12 Number of incoming bytes of a virtual network

a) **Description:** This measurement provides the number of bytes received at a virtual network instance. This measurement is split into subcounters per virtual network port.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more measurements indicating the number of incoming bytes for a virtual network port of a virtual network instance from NFVI in the collection period, and maps the received measurement(s) from the virtual network port to the virtual network instance. The measurement producer generates the measurement for the subject virtual network instance by assigning the value, or summing up multiple values, of the received measurement(s) to the sub-counter(s) per virtual network port.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VirtualisedNetworkResource.

f) **Measured Object Type:** VirtualNetwork.

g) **Measurement Name:** ByteIncoming.*vNPort*, where *vNPort* is equal to the *resourceId* of the measured virtual network port.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last measurement indicating the number of incoming bytes of the measured virtual network port received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of measurements indicating the number of incoming bytes" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of measurements indicating the number of incoming bytes" received from NFVI for the measured virtual network port in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last measurement indicating the number of incoming bytes of the measured virtual network port received from NFVI in the collection period is generated.

7.1.13 Number of outgoing bytes of a virtual network

a) **Description:** This measurement provides the number of bytes transmitted at a virtual network instance. This measurement is split into subcounters per virtual network port.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more measurements indicating the number of outgoing bytes for a virtual network port of a virtual network instance from NFVI in the collection period, and maps the received measurement(s) from the virtual network port to the virtual network instance. The measurement producer generates the measurement for the subject virtual network instance by assigning the value, or summing up multiple values, of the received measurement(s) to the sub-counter(s) per virtual network port.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VirtualisedNetworkResource.

f) **Measured Object Type:** VirtualNetwork.

g) **Measurement Name:** ByteOutgoing.*vNPort*, where *vNPort* is equal to the *resourceId* of the measured virtual network port.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last measurement indicating the number of outgoing bytes of the measured virtual network port received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of measurements indicating the number of outgoing bytes" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of measurements indicating the number of outgoing bytes" received from NFVI for the measured virtual network port in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last measurement indicating the number of outgoing bytes of the measured virtual network port received from NFVI in the collection period is generated.

7.1.14 Number of incoming packets of a virtual network

a) **Description:** This measurement provides the number of packets received at a virtual network instance. This measurement is split into subcounters per virtual network port.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more measurements indicating the number of incoming packets for a virtual network port of a virtual network instance from NFVI in the collection period, and maps the received measurement(s) from the virtual network port to the virtual network instance. The measurement producer generates the measurement for the subject virtual network instance by assigning the value, or summing up multiple values, of the received measurement(s) to the sub-counter(s) per virtual network port.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VirtualisedNetworkResource.

f) **Measured Object Type:** VirtualNetwork.

g) **Measurement Name:** PacketIncoming.*vNPort*, where *vNPort* is equal to the *resourceId* of the measured virtual network port.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last measurement indicating the number of incoming packets of the measured virtual network port received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of measurements indicating the number of incoming packets" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of measurements indicating the number of incoming packets" received from NFVI for the measured virtual network port in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last measurement indicating the number of incoming packets of the measured virtual network port received from NFVI in the collection period is generated.

7.1.15 Number of outgoing packets of a virtual network

a) **Description:** This measurement provides the number of packets transmitted at a virtual network instance. This measurement is split into subcounters per virtual network port.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more measurements indicating the number of outgoing packets for a virtual network port of a virtual network instance from NFVI in the collection period, and maps the received measurement(s) from the virtual network port to the virtual network instance. The measurement producer generates the measurement for the subject virtual network instance by assigning the value, or summing up multiple values, of the received measurement(s) to the sub-counter(s) per virtual network port.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VirtualisedNetworkResource.

f) **Measured Object Type:** VirtualNetwork.

g) **Measurement Name:** PacketOutgoing.*vNPort*, where *vNPort* is equal to the *resourceId* of the measured virtual network port.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last measurement indicating the number of outgoing packets of the measured virtual network port received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of measurements indicating the number of outgoing packets" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of measurements indicating the number of outgoing packets" received from NFVI for the measured virtual network port in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last measurement indicating the number of outgoing packets of the measured virtual network port received from NFVI in the collection period is generated.

7.1.16 Mean usage of Virtualised storage resource

a) **Description:** This measurement provides the mean usage of a Virtualised storage resource.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more measurements indicating the usage of a Virtualised storage resource from NFVI in the collection period, and generates the measurement by averaging multiple values of the received measurement(s).

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedStorageResource.

f) **Measured Object Type:** VirtualStorage.

g) **Measurement Name:** UsageMeanVStorage.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last measurement indicating the usage of the virtualised storage resource received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of measurements indicating the usage of the Virtualised storage resource" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of measurements indicating the usage of the Virtualised storage resource" received from NFVI for the measured Virtualised storage resource in the collection period.

7.1.17 Peak usage of Virtualised storage resource

a) **Description:** This measurement provides the peak usage of a Virtualised storage resource.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more measurements indicating the usage of a Virtualised storage resource from NFVI in the collection period, and generates the measurement by taking the maximum of multiple values of the received measurement(s).

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedStorageResource.

f) **Measured Object Type:** VirtualStorage.

g) **Measurement Name:** UsagePeakVStorage.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the measurement indicating the usage of the virtualised storage resource received from NFVI in the collection period.
- **MeasurementInterval:** Equals the "number of measurements indicating the usage of the Virtualised storage resource" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of measurements indicating the usage of the Virtualised storage resource" received from NFVI for the measured Virtualised storage resource in the collection period.

7.1.18 Energy consumption of a virtual compute

a) **Description:** This measurement provides the total energy consumption of the virtualised compute resource over the collection period.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the energy measurements for the virtual compute instance from NFVI at the pre-defined interval, and then sums the energy measurement values received in the collection period.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** EnergyConsumptionVirtualCompute.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from NFVI in the collection period.
- **MeasurementInterval:** Equals the measurement interval utilized by the NFVI (e.g. 1 second) for providing energy consumption measurements in the collection period.

7.1.19 Mean energy consumption of a virtual compute

a) **Description:** This measurement provides the mean energy consumption of the virtualised compute resource over the collection period.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the energy measurements for the virtual compute instance from NFVI at the pre-defined interval, and then takes the arithmetic mean of the energy measurement values received in the collection period.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VirtualCompute.

g) **Measurement Name:** EnergyConsumptionMeanVirtualCompute.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from NFVI in the collection period.
- **MeasurementInterval:** Equals the measurement interval utilized by the NFVI (e.g. 1 second) for providing energy consumption measurements in the collection period.

7.2 Performance measurements associated to VNF instances

7.2.1 Introduction

The performance measurements defined in this clause 7.2 are applicable to VNF instances (and related sub-object types) and they can be exposed, among others, on interfaces defined on the following reference points:

- Ve-Vnfm-em.
- Or-Vnfm, with the exceptions that measurements do not apply which are related to aspects internal to the VNF instance, as follows:
  - the measured object type "VNFC" does not apply to this reference point;
  - the measured object type "VnfIntCp" do not apply to this reference point, unless the measured connection point is related to an externally-manged internal virtual link.

Measurements related to VNF/VNFC managed objects consider the two forms of deployment of VNF/VNFC supported by the NFV-MANO, i.e. VM-based and container-based deployments.

The measurement producer can be, among others, a VNFM.

7.2.2 Mean virtual CPU usage of VNF/VNFC instance

a) **Description:** This measurement provides the mean virtual CPU usage of the underlying Virtual Compute instance(s) or groups of one or more OS containers, related to a VNF instance, or to a VNFC instance in a VNF instance. This measurement is split into sub-counters per Virtual Compute instance or group of one or more OS containers.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VCpuUsageMean measurement(s) (see clause 7.1.2) for the VirtualCompute instance(s) in the collection period, and maps the received VCpuUsageMean measurement(s) from the VirtualCompute instance to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by assigning the value or averaging multiple values of the received VCpuUsageMean measurement(s) to the sub-counter(s) per VirtualCompute instance.

- For a container-based VNFC, the measurement producer receives one or more CpuUsageMeanMCIO.container measurements(s) see clause 7.4.2) for the group of one or more OS containers in the collection period, and maps the received CpuUsageMeanMCIO.container measurement(s) from the group of one or more OS containers to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by, firstly averaging multiple values of received CpuUsageMeanMCIO.container of the same group of one or more OS containers, and secondly assigning the value or averaging multiple values of the computed averaged values(s) to the sub-counter(s) per group of one or more OS containers.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VNF, VNFC.

g) **Measurement Name:** VCpuUsageMeanVnf.vComputeId, where *vComputeId* is equal to the *objectInstanceId* of the measured object of the mapped measurement, i.e. VirtualCompute in the case VM-based VNFC and Mcio-c in the case of container-based VNFC.

h) **Measurement Context:**

- For a VM-based VNFC:

  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VCpuUsageMean measurement received from NFVI for the measured Virtualised compute resource in the collection period.

  - **TickInterval:** Indicates the tick interval (see ETSI GS NFV-TST 008 [10]) of the last VCpuUsageMean measurement received from NFVI for the measured Virtualised compute resource in the collection period.

  - **MeasurementInterval:** Equals the "number of VCpuUsageMean measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VCpuUsageMean measurements" received from NFVI for the measured Virtualised compute resource in the collection period.

  - **ExecutionContext:** Indicates the execution context (see ETSI GS NFV-TST 008 [10]) of the VCpuUsageMean measurements received from NFVI for the measured Virtualised compute resource in the collection period. When a single "non-Idle" VCPU Utilization is required, the sum of the utilization of all the non-Idle execution contexts should be reported as the "active" execution context.

- For a container-based VNFC:

  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last cpu usage measurement received from the CIS in the collection period.

- **MeasurementInterval:** Equals to "window" (see Annex A1) of the cpu usage measurements received from the CIS in the collection period.
- **ResourcesCpuLimit:** Equals to the value of CPU resources limit.
- **ResourcesCpuRequest:** Equals to the value of CPU resources request.

7.2.3 Peak virtual CPU usage of VNF/VNFC instance

a) **Description:** This measurement provides the peak virtual CPU usage of the underlying Virtual Compute instance(s) or groups of one or more OS containers, related to a VNF instance or to a VNFC instance in a VNF instance. This measurement is split into sub-counters per Virtual Compute instance or group of one or more OS containers.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VCpuUsagePeak measurement(s) (see clause 7.1.3) for the *VirtualCompute* instance(s) in the collection period, and maps the received VCpuUsagePeak measurement(s) from the *VirtualCompute* instance to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by assigning the value, or taking the maximum of multiple values of the received VCpuUsagePeak measurement(s) to the sub-counter(s) per *VirtualCompute* instance.
- For a container-based VNFC, the measurement producer receives one or more CpuUsagePeakMCIO.container measurements(s) see clause 7.4.3) for the group of one or more OS containers in the collection period, and maps the received CpuUsagePeakMCIO.container measurement(s) from the group of one or more OS containers to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by, firstly selecting the maximum of multiple values of received CpuUsagePeakMCIO.container of the same group of one or more OS containers, and secondly assigning the value or taking the maximum of multiple values of the selected value(s) to the sub-counter(s) per group of one or more OS containers.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VNF, VNFC.

g) **Measurement Name:** VCpuUsagePeakVnf.*vComputeId*, where *vComputeId* is equal to the *objectInstanceId* of the measured object of the mapped measurement, i.e. VirtualCompute in the case VM-based VNFC and Mcio-c in the case of container-based VNFC.

h) **Measurement Context:**

- For a VM-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VCpuUsageMean measurement received from NFVI for the measured Virtualised compute resource in the collection period.
  - **TickInterval:** Indicates the tick interval (see ETSI GS NFV-TST 008 [10]) of the VCpuUsageMean measurement with peak value received from NFVI for the measured Virtualised compute resource in the collection period.
  - **MeasurementInterval:** Equals the "number of VCpuUsageMean measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VCpuUsageMean measurements" received from NFVI for the measured Virtualised compute resource in the collection period.

- **ExecutionContext:** Indicates the execution context (see ETSI GS NFV-TST 008 [10]) of the VCpuUsageMean measurements received from NFVI for the measured Virtualised compute resource in the collection period. When a single "non-Idle" VCPU Utilization is required, the sum of the utilization of all the non-Idle execution contexts should be reported as the "active" execution context.

- For a container-based VNFC:

  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last cpu usage measurement received from the CIS in the collection period.

  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the cpu usage measurements received from the CIS in the collection period.

  - **ResourcesCpuLimit:** Equals to the value of CPU resources limit.

  - **ResourcesCpuRequest:** Equals to the value of CPU resources request.

7.2.4 Mean memory usage of VNF/VNFC instance

a) **Description:** This measurement provides the mean memory usage of the underlying Virtual Compute instance(s) or groups of one or more OS containers related to a VNF instance or to a VNFC instance in a VNF instance. This measurement is split into sub-counters per Virtual Compute instance or group of one or more OS containers.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VMemoryUsageMean measurement(s) (see clause 7.1.4) for the *VirtualCompute* instance(s) in the collection period, and maps the received VMemoryUsageMean measurement(s) from the *VirtualCompute* instance to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by assigning the value or averaging the multiple values of the received VMemoryUsageMean measurement(s) to the sub-counter(s) per *VirtualCompute* instance.

- For a container-based VNFC, the measurement producer receives one or more MemoryUsageMeanMCIO.container measurement(s) (see clause 7.4.4) for the group of one or more OS containers in the collection period, and maps the received MemoryUsageMeanMCIO.container measurement(s) from the group of one or more OS containers to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by, firstly averaging multiple values of received MemoryUsageMeanMCIO.container of the same group of one or more OS containers, and secondly assigning the value or averaging the multiple values of the computed averaged values(s) to the sub-counters(s) per group of one or more OS containers.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VNF, VNFC.

g) **Measurement Name:** VMemoryUsageMeanVnf.*vComputeId*, where *vComputeId* is equal to the *objectInstanceId* of the measured object of the mapped measurement, i.e. VirtualCompute in the case VM-based VNFC and Mcio-c in the case of container-based VNFC.

h) **Measurement Context:**

- For a VM-based VNFC:

  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VMemoryUsageMean measurement received from NFVI for the measured Virtualised compute resource in the collection period.

- **MeasurementInterval:** Equals the "number of VMemoryUsageMean measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VMemoryUsageMean measurements" received from NFVI for the measured Virtualised compute resource in the collection period.
- **MeasurementSystemRam:** Indicates the system RAM (see ETSI GS NFV-TST 008 [10]) of the measured Virtualised compute resource.
- **MeasurementSystemSwapSpace:** Indicates the system SWAP space (see ETSI GS NFV-TST 008 [10]) of the measured Virtualised compute resource.

- For a container-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last memory usage measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the memory usage measurements received from the CIS in the collection period.
  - **ResourcesMemoryLimit:** Equals to the value of memory resources limit.
  - **ResourcesMemoryRequest:** Equals to the value of memory resources request.

7.2.5 Peak memory usage of VNF/VNFC instance

a) **Description:** This measurement provides the peak memory usage of the underlying Virtual Compute instance(s) or groups of one or more OS containers related to a VNF instance or to a VNFC instance in a VNF instance. This measurement is split into sub-counters per Virtual Compute instance or group of one or more OS containers.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VMemoryUsagePeak measurement(s) (see clause 7.1.5) for the *VirtualCompute* instance(s) in the collection period, and maps the received VMemoryUsagePeak measurement(s) from the *VirtualCompute* instance to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by assigning the value or taking the maximum of multiple values of the received VMemoryUsagePeak measurement(s) to the sub-counter(s) per *VirtualCompute* instance.
- For a container-based VNFC, the measurement producer receives one or more MemoryUsagePeakMCIO.container measurement(s) (see clause 7.4.5) for the group of one or more OS containers in the collection period, and maps the received MemoryUsagePeakMCIO.container measurement(s) from the group of one or more OS containers to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by, firstly selecting the maximum of multiple values of received MemoryUsagePeakMCIO.container of the same group of one or more OS containers, and secondly assigning the value or taking the maximum of multiple values of the selected value(s) to the sub-counters(s) per group of one or more OS containers.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VNF, VNFC.

g) **Measurement Name:** VMemoryUsagePeakVnf.*vComputeId*, where *vComputeId* is equal to the *objectInstanceId* of the measured object of the mapped measurement, i.e. VirtualCompute in the case VM-based VNFC and Mcio-c in the case of container-based VNFC.

h) **Measurement Context:**

- For a VM-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the VMemoryUsageMean measurement with peak value received from NFVI for the measured Virtualised compute resource in the collection period.
  - **MeasurementInterval:** Equals the "number of VMemoryUsageMean measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VMemoryUsageMean measurements" received from NFVI for the measured Virtualised compute resource in the collection period.
  - **MeasurementSystemRam:** Indicates the system RAM (see ETSI GS NFV-TST 008 [10]) of the measured Virtualised compute resource.
  - **MeasurementSystemSwapSpace:** Indicates the system SWAP space (see ETSI GS NFV-TST 008 [10]) of the measured Virtualised compute resource.
- For a container-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last memory usage measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the memory usage measurements received from the CIS in the collection period.
  - **ResourcesMemoryLimit:** Equals to the value of memory resources limit.
  - **ResourcesMemoryRequest:** Equals to the value of memory resources request.

7.2.6 Mean disk usage of VNF/VNFC instance

a) **Description:** This measurement provides the mean disk usage of the underlying Virtual Compute instance or group of one or more OS containers related to a VNF instance or to a VNFC instance in a VNF instance. This measurement is split into sub-counters per Virtual Compute instance or group of one or more OS containers.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VDiskUsageMean measurement(s) (see clause 7.1.6) for the *VirtualCompute* instance(s) in the collection period, and maps the received VDiskUsageMean measurement(s) from the *VirtualCompute* instance to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by assigning the value, or averaging the multiple values of the received VDiskUsageMean measurement(s) to the sub-counter(s) per *VirtualCompute* instance.
- For a container-based VNFC, the measurement producer receives one or more StorageUsageMeanMCIO.container measurement(s) (see clause 7.4.10) for the group of one or more OS containers in the collection period, and maps the received StorageUsageMeanMCIO.container measurement(s) from the group of one or more OS containers to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by, firstly averaging multiple values of received StorageUsageMeanMCIO.container of the same group of one or more OS containers, and secondly assigning the value or averaging multiple values of the computed averaged value(s) to the sub-counter(s) per group of one or more OS containers.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VNF, VNFC.

g) **Measurement Name:** VDiskUsageMeanVnf.*vComputeId*, where *vComputeId* is equal to the *objectInstanceId* of the measured object of the mapped measurement, i.e. VirtualCompute in the case VM-based VNFC and Mcio-c in the case of container-based VNFC.

h) **Measurement Context:**

- For a VM-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VDiskUsageMean measurement received from NFVI for the measured Virtualised compute resource in the collection period.
  - **MeasurementInterval:** Equals the "number of VDiskUsageMean measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VDiskUsageMean measurements" received from NFVI for the measured Virtualised compute resource in the collection period.
- For a container-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last storage resource usage measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the storage resource usage measurements received from the CIS in the collection period.

7.2.7 Peak disk usage of VNF/VNFC instance

a) **Description:** This measurement provides the peak disk usage of the underlying Virtual Compute instances or groups of one or more OS containers related to a VNF instance or to a VNFC instance in a VNF instance. This measurement is split into sub-counters per Virtual Compute instance or group of one or more OS containers.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VDiskUsagePeak measurement(s) (see clause 7.1.7) for the *VirtualCompute* instance(s) in the collection period, and maps the received VDiskUsagePeak measurement(s) from the *VirtualCompute* instance to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by assigning the value or taking the maximum of multiple values of the received VDiskUsagePeak measurement(s) to the sub-counter(s) per *VirtualCompute* instance.
- For a container-based VNFC, the measurement producer receives one or more StorageUsagePeakMCIO.container measurement(s) (see clause 7.4.11) for the group of one or more OS containers in the collection period, and maps the received StorageUsagePeakMCIO.container measurement(s) from the group of one or more OS containers to the VNF instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNF/VNFC instance by, firstly selecting the maximum of multiple values of received StorageUsagePeakMCIO.container of the same group of one or more OS containers, and secondly assigning the value or taking the maximum of multiple values of the selected value(s) to the sub-counters(s) per group of one or more OS containers.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** VirtualisedComputeResource.

f) **Measured Object Type:** VNF, VNFC.

g) **Measurement Name:** VDiskUsagePeakVnf.*vComputeId*, where *vComputeId* is equal to the *objectInstanceId* of the measured object of the mapped measurement, i.e. VirtualCompute in the case VM-based VNFC and Mcio-c in the case of container-based VNFC.

h) **Measurement Context:**

- For a VM-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the VDiskUsageMean measurement with peak value received from NFVI for the measured Virtualised compute resource in the collection period.
  - **MeasurementInterval:** Equals the "number of VDiskUsageMean measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VDiskUsageMean measurements" received from NFVI for the measured Virtualised compute resource in the collection period.
- For a container-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last storage resource usage measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the storage resource usage measurements received from the CIS in the collection period.

7.2.8 Number of incoming bytes of VNF internal CP

a) **Description:** This measurement provides the number of bytes received by a VNF internal Connection Point (CP).

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VNetByteIncoming.*vNetItfId* measurement(s) (see clause 7.1.8) for the Virtual Compute instance(s) in the collection period, and maps the received VNetByteIncoming.*vNetItfId* measurement(s) from the Virtual Compute instance to the VNF internal CP instance. The measurement producer generates the measurement for the subject VNF internal CP instance by assigning the value, or summing up multiple values, of the mapped VNetByteIncoming.*vNetItfId* subcounter(s).
- For a container-based VNFC, the measurement producer receives one or more NetByteIncomingMCIO.*vNetItfId* measurement(s) (see clause 7.4.6) for the group of one or more OS containers in the collection period, and maps the received NetByteIncomingMCIO.*vNetItfId* measurement(s) from the group of one or more OS containers to the VNF internal CP instance. The measurement producer generates the measurement for the subject VNF internal CP instance by assigning the value, or summing up multiple values, of the mapped NetByteIncoming.*vNetItfId* subcounter(s).

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VnfInternalCP.

f) **Measured Object Type:** VnfIntCP.

g) **Measurement Name:** ByteIncomingVnfIntCp.

h) **Measurement Context:**

- For a VM-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VNetByteIncoming.*vNetItfId* measurement received from NFVI for the measured CP in the collection period.
  - **MeasurementInterval:** Equals the "number of VNetByteIncoming.*vNetItfId* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VNetByteIncoming.*vNetItfId* measurements" received from NFVI for the measured CP in the collection period.

- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last VNetByteIncoming.*vNetItfId* measurement received from NFVI for the measured CP in the collection period is generated.

- For a container-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Rxbytes measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the Rxbytes measurements received from the CIS in the collection period.

7.2.9 Number of outgoing bytes of VNF internal CP

a) **Description:** This measurement provides the number of bytes transmitted by a VNF internal CP.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VNetByteOutgoing.*vNetItfId* measurement(s) (see clause 7.1.9) for the Virtual Compute instances in the collection period, and maps the received VNetByteOutgoing.*vNetItfId* measurement(s) from the Virtual Compute instance to the VNF internal CP instance. The measurement producer generates the measurement for the subject VNF internal CP instance by mapping the value of *vNetItfId* to the CP of the VNFC instance, and assigning the value, or summing up multiple values, of the mapped VNetByteOutgoing.*vNetItfId* subcounter(s).
- For a container-based VNFC, the measurement producer receives one or more NetByteOutgoingMCIO.*vNetItfId* measurement(s) (see clause 7.4.7) for the group of one or more OS containers in the collection period, and maps the received NetByteOutgoing.*vNetItfId* measurement(s) from the group of one or more OS containers to the VNF internal CP instance. The measurement producer generates the measurement for the subject VNF internal CP instance by mapping the value of *vNetItf* to the CP of the VNFC instance, and assigning the value, or summing up multiple values, of the mapped NetByteOutgoingMCIO.*vNetItfId* subcounter(s).

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VnfInternalCP.

f) **Measured Object Type:** VnfIntCP.

g) **Measurement Name:** ByteOutgoingVnfIntCp.

h) **Measurement Context:**

- For a VM-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VNetByteOutgoing.*vNetItfId* measurement received from NFVI for the measured CP in the collection period.
  - **MeasurementInterval:** Equals the "number of VNetByteOutgoing.*vNetItfId* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VNetByteOutgoing.*vNetItfId* measurements" received from NFVI for the measured CP in the collection period.
  - **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last VNetByteOutgoing.*vNetItfId* measurement received from NFVI for the measured CP in the collection period is generated.
- For a container-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last TxBytes measurement received from the CIS in the collection period.

- **MeasurementInterval:** Equals to "window" (see Annex A1) of the TxBytes measurements received from the CIS in the collection period.

7.2.10 Number of incoming packets of VNF internal CP

a) **Description:** This measurement provides the number of packets received by a VNF internal CP.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VNetPacketIncoming.*vNetItfId* measurement(s) (see clause 7.1.10) for the Virtual Compute instances in the collection period, and maps the received VNetPacketIncoming.*vNetItfId* measurement(s) from the Virtual Compute instance to the VNF internal CP instance. The measurement producer generates the measurement for the subject VNF internal CP instance by mapping the value of *vNetItfId* to the CP of the VNFC instance, and assigning the value, or summing up multiple values, of the mapped VNetPacketIncoming.*vNetItfId* subcounter(s).

- For a container-based VNFC, the measurement producer receives one or more NetPacketIncomingMCIO.*vNetItfId* measurement(s) (see clause 7.4.8) for the groups of one more OS containers in the collection period, and maps the received NetPacketIncomingMCIO.*vNetItfId* measurement(s) from the group of one or more OS containers to the VNF internal CP instance. The measurement producer generates the measurement for the subject VNF internal CP instance by mapping the value of *vNetItfId* to the CP of the VNFC instance, and assigning the value, or summing up multiple values, of the mapped NetPacketIncomingMCIO.*vNetItfId* subcounter(s).

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VnfInternalCP.

f) **Measured Object Type:** VnfIntCP.

g) **Measurement Name:** PacketIncomingVnfIntCp.

h) **Measurement Context:**

- For a VM-based VNFC:

  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VNetPacketIncoming.*vNetItfId* measurement received from NFVI for the measured CP in the collection period.

  - **MeasurementInterval:** Equals the "number of VNetPacketIncoming.*vNetItfId* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VNetPacketIncoming.*vNetItfId* measurements" received from NFVI for the measured CP in the collection period.

  - **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last VNetPacketIncoming.*vNetItfId* measurement received from NFVI for the measured CP in the collection period is generated.

- For a container-based VNFC:

  - **MeasurementEndTime:** Indicates the end time of the last Rx_Packets measurement received from the CIS in the collection period.

  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the Rx_Packets measurements received from the CIS in the collection period.

7.2.11 Number of outgoing packets of VNF internal CP

a) **Description:** This measurement provides the number of packets transmitted by a VNF internal CP.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more VNetPacketOutgoing.*vNetItfId* measurement(s) (see clause 7.1.11) for the Virtual Compute instances in the collection period, and maps the received VNetPacketOutgoing.*vNetItfId* measurement(s) from the Virtual Compute instance to the VNF internal CP instance. The measurement producer generates the measurement for the subject VNF internal CP instance by mapping the value of *vNetItfId* to the CP of the VNFC instance, and assigning the value, or summing up multiple values, of the mapped VNetPacketOutgoing.*vNetItfId* subcounter(s).
- For a container-based VNFC, the measurement producer receives one or more NetPacketOutgoingMCIO.*vNetItfId* measurement(s) (see clause 7.4.9) for the groups of one or more OS containers in the collection period, and maps the received NetPacketOutgoingMCIO.*vNetItfId* measurement(s) from the group of one or more OS containers to the VNF internal CP instance. The measurement producer generates the measurement for the subject VNF internal CP instance by mapping the value of *vNetItfId* to the CP of the VNFC instance, and assigning the value, or summing up multiple values, of the mapped NetPacketOutgoingMCIO.*vNetItfId* subcounter(s).

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VnfInternalCP.

f) **Measured Object Type:** VnfIntCP.

g) **Measurement Name:** PacketOutgoingVnfIntCp.

h) **Measurement Context:**

- For a VM-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VNetPacketOutgoing.*vNetItfId* measurement received from NFVI for the measured CP in the collection period.
  - **MeasurementInterval:** Equals the "number of VNetPacketOutgoing.*vNetItfId* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VNetPacketOutgoing.*vNetItfId* measurements" received from NFVI for the measured CP in the collection period.
  - **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last VNetPacketOutgoing.*vNetItfId* measurement received from NFVI for the measured CP in the collection period is generated.
- For a container-based VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Tx_Packets measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the Tx_Packets measurements received from the CIS in the collection period.

7.2.12 Number of incoming bytes of VNF external CP

a) **Description:** This measurement provides the number of incoming bytes received by a VNF external CP.

b) **Collection Method:** OM.

c) **Trigger:**

- The measurement is triggered based on the mapping of the measured VNF external CP (see clause 7.1.3 of ETSI GS NFV-IFA 011 [4]).

- In case the VNF external CP is mapped to a VNF internal CP (applicable to both VM-based and container-based VNF/VNFC):
  - The measurement producer generates the measurement by mapping the performance measurement VNetByteIncomingVnf.*vnfcInstanceId.cpId* (as defined in clause 7.2.8) from VNF internal CP to VNF external CP.
- In case the VNF external CP is mapped to an internal VL (applicable only to VM-based VNF/VNFC):
  - The measurement producer receives one or more ByteIncoming.*vNPort* measurement(s) (see clause 7.1.12) for the virtual network port(s) in the collection period, and maps the received ByteIncoming.*vNPort* subcounter(s) from the virtual network port firstly to the VNF internal VL and then to the VNF external CP (see note). The measurement producer generates the measurement for the subject VNF external CP by assigning the value, or summing up multiple values, of the mapped ByteIncoming.*vNPort* subcounter(s).

NOTE: The mapping between VNF internal VL and VNF external CP has dependency on the connectivity modelling between them. The exact mapping is out of scope of the present document.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VnfExternalCP.

f) **Measured Object Type:** VnfExtCP.

g) **Measurement Name:** ByteIncomingVnfExtCp.

h) **Measurement Context:**

- For the case of VM-based VNF/VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VNetByteIncoming.*vNetItfId* or ByteIncoming.*vNPort* measurement received from NFVI for the measured CP in the collection period.
  - **MeasurementInterval:** Equals the "number of VNetByteIncoming.*vNetItfId* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VNetByteIncoming.*vNetItfId* measurements" received from NFVI for the measured CP in the collection period, or the "number of ByteIncoming.*vNPort* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the ByteIncoming.*vNPort* measurements" received from NFVI for the measured CP in the collection period.
  - **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last VNetByteIncoming.*vNetItfId* or ByteIncoming.*vNPort* measurement received from NFVI for the measured CP in the collection period is generated.
- For a container-based VNF/VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Rxbytes measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the Rxbytes measurements received from the CIS in the collection period.

7.2.13 Number of outgoing bytes of VNF external CP

a) **Description:** This measurement provides the number of outgoing bytes transmitted by a VNF external CP.

b) **Collection Method:** OM.

c) **Trigger:**

- The measurement is triggered based on the mapping of the measured VNF external CP (see clause 7.1.3 of ETSI GS NFV-IFA 011 [4]).

- In case the VNF external CP is mapped to a VNF internal CP (applicable to both VM-based and container-based VNF/VNFC):
  - The measurement producer generates the measurement by mapping the performance measurement VNetByteOutgoingVnf.*vnfcInstanceId.cpId* (as defined in clause 7.2.9) from VNF internal CP to VNF external CP.
- In case the VNF external CP is mapped to an internal VL (applicable only to VM-based VNF/VNFC):
  - The measurement producer receives one or more ByteOutgoing.*vNPort* measurement(s) (see clause 7.1.13) for the virtual network port(s) in the collection period, and maps the received ByteOutgoing.*vNPort* subcounter(s) from the virtual network port firstly to the VNF internal VL and then to the VNF external CP (see note). The measurement producer generates the measurement for the subject VNF external CP by assigning the value, or summing up multiple values, of the mapped ByteOutgoing.*vNPort* subcounter(s).

NOTE: The mapping between VNF internal VL and VNF external CP has dependency on the connectivity modelling between them. The exact mapping is out of scope of the present document.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VnfExternalCP.

f) **Measured Object Type:** VnfExtCP.

g) **Measurement Name:** ByteOutgoingVnfExtCp.

h) **Measurement Context:**

- For the case of VM-based VNF/VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VNetByteOutgoing.*vNetItfId* or ByteOutgoing.*vNPort* measurement received from NFVI for the measured CP in the collection period.
  - **MeasurementInterval:** Equals the "number of VNetByteOutgoing.*vNetItfId* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VNetByteOutgoing.*vNetItfId* measurements" received from NFVI for the measured CP in the collection period, or the "number of ByteOutgoing.*vNPort* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the ByteOutgoing.*vNPort* measurements" received from NFVI for the measured CP in the collection period.
  - **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last VNetByteOutgoing.*vNetItfId* or ByteOutgoing.*vNPort* measurement received from NFVI for the measured CP in the collection period is generated.
- For a container-based VNF/VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last TxBytes measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the TxBytes measurements received from the CIS in the collection period.

7.2.14 Number of incoming packets of VNF external CP

a) **Description:** This measurement provides the number of incoming packets received by a VNF external CP.

b) **Collection Method:** OM.

c) **Trigger:**

- The measurement is triggered based on the mapping of the measured VNF external CP (see clause 7.1.3 of ETSI GS NFV-IFA 011 [4]).

- In case the VNF external CP is mapped to a VNF internal CP (applicable to both VM-based and container-based VNF/VNFC):
  - The measurement producer generates the measurement by mapping the performance measurement VNetPacketIncomingVnf.*vnfcInstanceId.cpId* (as defined in clause 7.2.10) from VNF internal CP to VNF external CP.
- In case the VNF external CP is mapped to an internal VL (applicable only to VM-based VNF/VNFC):
  - The measurement producer receives one or more PacketIncoming.*vNPort* measurement(s) (see clause 7.1.14) from for the virtual network port(s) in the collection period, and maps the received PacketIncoming.*vNPort* subcounter(s) from the virtual network port firstly to the VNF internal VL and then to the VNF external CP (see note). The measurement producer generates the measurement for the subject VNF external CP by assigning the value, or summing up multiple values, of the mapped PacketIncoming.*vNPort* subcounter(s).

NOTE: The mapping between VNF internal VL and VNF external CP has dependency on the connectivity modelling between them. The exact mapping is out of scope of the present document.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VnfExternalCP.

f) **Measured Object Type:** VnfExtCP.

g) **Measurement Name:** PacketIncomingVnfExtCp.

h) **Measurement Context:**

- For the case of VM-based VNF/VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VNetPacketIncoming.*vNetItfId* or PacketIncoming.*vNPort* measurement received from NFVI for the measured CP in the collection period.
  - **MeasurementInterval:** Equals the "number of VNetPacketIncoming.*vNetItfId* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VNetPacketIncoming.*vNetItfId* measurements" received from NFVI for the measured CP in the collection period, or the "number of PacketIncoming.*vNPort* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the PacketIncoming.*vNPort* measurements" received from NFVI for the measured CP in the collection period.
  - **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last VNetPacketIncoming.*vNetItfId* or PacketIncoming.*vNPort* measurement received from NFVI for the measured CP in the collection period is generated.
- For a container-based VNF/VNFC:
  - **MeasurementEndTime:** Indicates the end time of the last Rx_Packets measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to"window" (see Annex A1) of the Rx_Packets measurements received from the CIS in the collection period.

7.2.15 Number of outgoing packets of VNF external CP

a) **Description:** This measurement provides the number of outgoing packets transmitted by a VNF external CP.

b) **Collection Method:** OM.

c) **Trigger:**

- The measurement is triggered based on the mapping of the measured VNF external CP (see clause 7.1.3 of ETSI GS NFV-IFA 011 [4]).

- In case the VNF external CP is mapped to a VNF internal CP (applicable to both VM-based and container-based VNF/VNFC):
  - The measurement producer generates the measurement by mapping the performance measurement VNetPacketOutgoingVnf.*vnfcInstanceId.cpId* (as defined in clause 7.2.11) from VNF internal CP to VNF external CP.
- In case the VNF external CP is mapped to an internal VL (applicable only to VM-based VNF/VNFC):
  - The measurement producer receives one or more PacketOutgoing.*vNPort* measurement(s) (see clause 7.1.15) for the virtual network port(s) in the collection period, and maps the received PacketOutgoing.*vNPort* subcounter(s) from the virtual network port firstly to the VNF internal VL and then to the VNF external CP (see note). The measurement producer generates the measurement for the subject VNF external CP by assigning the value, or summing up multiple values, of the mapped PacketOutgoing.*vNPort* subcounter(s).

NOTE: The mapping between VNF internal VL and VNF external CP has dependency on the connectivity modelling between them. The exact mapping is out of scope of the present document.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** VnfExternalCP.

f) **Measured Object Type:** VnfExtCP.

g) **Measurement Name:** PacketOutgoingVnfExtCp.

h) **Measurement Context:**

- For the case of VM-based VNF/VNFC:
  - **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last VNetPacketOutgoing.*vNetItfId* or PacketOutgoing.*vNPort* measurement received from NFVI for the measured CP in the collection period.
  - **MeasurementInterval:** Equals the "number of VNetPacketOutgoing.*vNetItfId* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the VNetPacketOutgoing.*vNetItfId* measurements" received from NFVI for the measured CP in the collection period, or the "number of PacketOutgoing.*vNPort* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the PacketOutgoing.*vNPort* measurements" received from NFVI for the measured CP in the collection period.
  - **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last VNetPacketOutgoing.*vNetItfId* or PacketOutgoing.*vNPort* measurement received from NFVI for the measured CP in the collection period is generated.
- For a container-based VNF/VNFC:
  - **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Tx_Packets measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals to "window" (see Annex A1) of the Tx_Packets measurements received from the CIS in the collection period.

7.2.16 Energy consumption of VNFC instance

a) **Description:** This measurement provides the total energy consumption related to VNFC instance(s) in a VNF instance, based on the energy consumption of the underlying virtual compute instance(s) or groups of one or more OS containers, over the collection period. This measurement is split into sub-counters per virtual compute instance or group of one or more OS containers.

b) **Collection Method:** OM.

c) Trigger:

- For a VM-based VNFC, the measurement producer receives one or more virtual compute energy consumption measurement(s) (see clause 7.1.18) for the VirtualCompute instance(s) in the collection period, and maps the received EnergyConsumptionVirtualCompute measurement(s) from the VirtualCompute instance to the VNFC instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNFC instance by assigning the value, or summing up multiple values of the mapped EnergyConsumptionVirtualCompute measurements sub-counter(s) per VirtualCompute instance.

- For a container-based VNFC, the measurement producer receives one or more EnergyConsumptionMcio.total measurement(s) (see clause 7.4.12) for the group of one or more OS containers in the collection period, and maps the received EnergyConsumptionMcio.total measurement(s) from the group of one or more OS containers to the VNFC instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNFC instance by assigning the value, or summing up multiple values of the mapped EnergyConsumptionMcio sub-counters(s) per group of one or more OS containers.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** VnfComputeResource.

f) **Measured Object Type:** VNFC.

g) **Measurement Name:** EnergyConsumptionVnfc.vComputeId, where "vComputeId" is equal to the objectInstanceId of the measured object of the mapped measurement, i.e. VirtualCompute in the case of VM-based VNFC and Mcio-c in the case of container-based VNFC.

h) **Measurement Context:**

- For a VM-based VNFC:

  - **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from NFVI in the collection period.

  - **MeasurementInterval:** Equals the measurement interval utilized by the NFVI (e.g. 1 second) for providing energy consumption measurements in the collection period.

- For a container-based VNFC:

  - **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from the CIS in the collection period.

  - **MeasurementInterval:** Equals the measurement interval utilized by the CIS (e.g. 1 second) for providing energy consumption measurements in the collection period.

7.2.17 Mean energy consumption of VNFC instance

a) **Description:** This measurement provides the mean energy consumption related to VNFC instance(s) in a VNF instance, based on the energy consumption of the underlying virtual compute instance(s) or groups of one or more OS containers, over the collection period. This measurement is split into sub-counters per virtual compute instance or group of one or more OS containers.

b) **Collection Method:** SC.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more virtual compute energy consumption measurement(s) (see clause 7.1.18) for the VirtualCompute instance(s) in the collection period, and maps the received EnergyConsumptionVirtualCompute measurement(s) from the VirtualCompute instance to the VNFC instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNFC instance by taking the arithmetic mean of the mapped energy measurement values. The measurement producer assigns the computed measurement for the subject VNFC instance by assigning the value to the sub-counters(s) per VirtualCompute instance.

- For a container-based VNFC, the measurement producer receives one or more EnergyConsumptionMcio.total measurement(s) (see clause 7.4.12) for the group of one or more OS containers in the collection period, and maps the received EnergyConsumptionVirtualCompute measurement(s) from the group of one or more OS containers to the VNFC instance, or the combination of VNF instance and VNFC instance. The measurement producer generates the measurement for the subject VNFC instance by taking the arithmetic mean of the mapped energy measurement values. The measurement producer assigns the computed measurement for the subject VNF/VNFC instance by assigning the value to the sub-counters(s) per group of one or more OS containers.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** VnfComputeResource.

f) **Measured Object Type:** VNFC.

g) **Measurement Name:** EnergyConsumptionMeanVnfc.vComputeId, where "vComputeId" is equal to the objectInstanceId of the measured object of the computed measurement, i.e. VirtualCompute in the case of VM-based VNFC and Mcio-c in the case of container-based VNFC.

h) **Measurement Context:**

- For a VM-based VNFC:
  - **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from NFVI in the collection period.
  - **MeasurementInterval:** Equals the measurement interval utilized by the NFVI (e.g. 1 second) for providing energy consumption measurements in the collection period.
- For a container-based VNFC:
  - **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals the measurement interval utilized by the CIS (e.g. 1 second) for providing energy consumption measurements in the collection period.

7.2.18 Energy consumption of VNF instance

a) **Description:** This measurement provides the total energy consumption related to a VNF instance, based on the energy consumption of the underlying virtual compute instance(s) or groups of one or more OS containers, over the collection period.

b) **Collection Method:** OM.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more virtual compute energy consumption measurement(s) (see clause 7.1.18) for the VirtualCompute instance(s) in the collection period, and computes the "energy consumption of VNFC instance" per each VNFC instance comprising the VNF instance (see clause 7.2.16). Then, the measurement producer sums up the "energy consumption of VNFC instance" considering the whole set of VNFC instances comprising the VNF instance.
- For a container-based VNFC, the measurement producer receives one or more EnergyConsumptionMcio.total measurement(s) (see clause 7.4.12) for the group of one or more OS containers in the collection period, and computes the "energy consumption of VNFC instance" per each VNFC instance comprising the VNF instance (see clause 7.2.16). Then, the measurement producer sums up the "energy consumption of VNFC instance" considering the whole set of VNFC instances comprising the VNF instance.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** VnfComputeResource.

f) **Measured Object Type:** VNF.

g) **Measurement Name:** EnergyConsumptionVnf.

h) **Measurement Context:**

- For a VNF instance comprising VM-based VNFC instances:
  - **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from NFVI in the collection period.
  - **MeasurementInterval:** Equals the measurement interval utilized by the NFVI (e.g. 1 second) for providing energy consumption measurements in the collection period.
- For a VNF instance comprising container-based VNFC instances:
  - **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals the measurement interval utilized by the CIS (e.g. 1 second) for providing energy consumption measurements in the collection period.

NOTE: The MeasurementEndTime and MeasurementInterval for energy consumption of MCIO and virtual compute measurements are expected to be the same across all VNFC instances that constitute the VNF instance.

7.2.19 Mean energy consumption of VNF instance

a) **Description:** This measurement provides the mean energy consumption related to a VNF instance, based on the energy consumption of the underlying virtual compute instance(s) or groups of one or more OS containers, over the collection period.

b) **Collection Method:** SC.

c) **Trigger:**

- For a VM-based VNFC, the measurement producer receives one or more virtual compute energy consumption measurement(s) (see clause 7.1.18) for the VirtualCompute instance(s) in the collection period and computes the "mean energy consumption of VNFC instance" per each VNFC instance comprising the VNF instance (see clause 7.2.17). Then, the measurement producer sums up the "mean energy consumption of VNFC instance" considering the whole set of VNFC instances comprising the VNF instance.
- For a container-based VNFC, the measurement producer receives one or more EnergyConsumptionMcio.total measurement(s) (see clause 7.4.12) for the group of one or more OS containers in the collection period and computes the "mean energy consumption of VNFC instance" per each VNFC instance comprising the VNF instance (see clause 7.2.17). Then, the measurement producer sums up the "mean energy consumption of VNFC instance" considering the whole set of VNFC instances comprising the VNF instance.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** VnfComputeResource.

f) **Measured Object Type:** VNF.

g) **Measurement Name:** EnergyConsumptionMeanVnf.

h) **Measurement Context:**

- For a VNF instance comprising VM-based VNFC instances:
  - **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from NFVI in the collection period.
  - **MeasurementInterval:** Equals the measurement interval utilized by the NFVI (e.g. 1 second) for providing energy consumption measurements in the collection period.

- For a VNF instance comprising container-based VNFC instances:
  - **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from the CIS in the collection period.
  - **MeasurementInterval:** Equals the measurement interval utilized by the CIS (e.g. 1 second) for providing energy consumption measurements in the collection period.

NOTE: The MeasurementEndTime and MeasurementInterval for energy consumption of MCIO and virtual compute measurements are expected to be the same across all VNFC instances that constitute the VNF instance.

7.3 Performance measurements associated to NS instances

7.3.1 Introduction

The performance measurements defined in this clause 7.3 are applicable to NS instances (and related sub-object types) and they can be exposed, among others, on interfaces defined on the following reference points:

- Os-Ma-nfvo.

The measurement producer can be, among others, an NFVO.

7.3.2 Number of incoming bytes of SAP

a) **Description:** This measurement provides the number of incoming bytes received by an SAP of an NS instance.

b) **Collection Method:** OM.

c) **Trigger:**

- The measurement is triggered based on the mapping of the measured SAP.
- In case the SAP is mapped to a VNF external CP:
  - The measurement producer receives one or more ByteIncomingVnfExtCp measurement(s) (see clause 7.2.12) in the collection period, and maps the received ByteIncomingVnfExtCp measurement(s) from the VNF external CP to SAP. The measurement producer generates the measurement for the subject SAP by assigning the value, or summing up multiple values, of the mapped ByteIncomingVnfExtCp measurement(s).
- In case the SAP is mapped to an NS VL:
  - The measurement producer receives one or more ByteIncoming.*vNPort* measurement(s) (see clause 7.1.12) for the virtual network port(s) in the collection period, and maps the received ByteIncoming.*vNPort* subcounter(s) from the virtual network port firstly to the NS VL and then to the SAP (see note). The measurement producer generates the measurement for the subject SAP by assigning the value, or summing up multiple values, of the mapped ByteIncoming.*vNPort* subcounter(s).

NOTE: The mapping between NS VL and SAP has dependency on the connectivity modelling between them. The exact mapping is out of scope of the present document.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** ServiceAccessPoint.

f) **Measured Object Type:** Sap.

g) **Measurement Name:** ByteIncomingSap.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last ByteIncomingVnfExtCp or ByteIncoming.*vNPort* measurement received from NFVI for the measured SAP in the collection period.
- **MeasurementInterval:** Equals the "number of ByteIncomingVnfExtCp measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the ByteIncomingVnfExtCp measurements" received from NFVI for the measured SAP in the collection period, or the "number of ByteIncoming.*vNPort* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the ByteIncoming.*vNPort* measurements" received from NFVI for the measured SAP in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last ByteIncomingVnfExtCp or ByteIncoming.*vNPort* measurement received from NFVI for the measured SAP in the collection period is generated.

7.3.3 Number of outgoing bytes of SAP

a) **Description:** This measurement provides the number of outgoing bytes transmitted by an SAP of an NS instance.

b) **Collection Method:** OM.

c) **Trigger:**

- The measurement is triggered based on the mapping of the measured SAP.
- In case the SAP is mapped to a VNF external CP:
  - The measurement producer receives one or more ByteOutgoingVnfExtCp measurement(s) (see clause 7.2.13) in the collection period, and maps the received ByteOutgoingVnfExtCp measurement(s) from the VNF external CP to SAP. The measurement producer generates the measurement for the subject SAP by assigning the value, or summing up multiple values, of the mapped ByteOutgoingVnfExtCp measurement(s).
- In case the SAP is mapped to an NS VL:
  - The measurement producer receives one or more ByteOutgoing.*vNPort* measurement(s) (see clause 7.1.13) for the virtual network port(s) in the collection period, and maps the received ByteOutgoing.*vNPort* subcounter(s) from the virtual network port firstly to the NS VL and then to the SAP (see note). The measurement producer generates the measurement for the subject SAP by assigning the value, or summing up multiple values, of the mapped ByteOutgoing.*vNPort* subcounter(s).

NOTE: The mapping between NS VL and SAP has dependency on the connectivity modelling between them. The exact mapping is out of scope of the present document.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** ServiceAccessPoint.

f) **Measured Object Type:** Sap.

g) **Measurement Name:** ByteOutgoingSap.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last ByteOutgoingVnfExtCp or ByteOutgoing.*vNPort* measurement received from NFVI for the measured SAP in the collection period.

- **MeasurementInterval:** Equals the "number of ByteOutgoingVnfExtCp measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the ByteOutgoingVnfExtCp measurements" received from NFVI for the measured SAP in the collection period, or the "number of ByteOutgoing.*vNPort* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the ByteOutgoing.*vNPort* measurements" received from NFVI for the measured SAP in the collection period.

- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last ByteOutgoingVnfExtCp or ByteOutgoing.*vNPort* measurement received from NFVI for the measured SAP in the collection period is generated.

7.3.4 Number of incoming packets of SAP

a) **Description:** This measurement provides the number of incoming packets received by an SAP of an NS instance.

b) **Collection Method:** OM.

c) **Trigger:**

- The measurement is triggered based on the mapping of the measured SAP.

- In case the SAP is mapped to a VNF external CP:

  - The measurement producer receives one or more PacketIncomingVnfExtCp measurement(s) (see clause 7.2.14) in the collection period, and maps the received PacketIncomingVnfExtCp measurement(s) from the VNF external CP to SAP. The measurement producer generates the measurement for the subject SAP by assigning the value, or summing up multiple values, of the mapped PacketIncomingVnfExtCp measurement(s).

- In case the SAP is mapped to an NS VL:

  - The measurement producer receives one or more PacketIncoming.*vNPort* measurement(s) (see clause 7.1.14) for the virtual network port(s) in the collection period, and maps the received PacketIncoming.*vNPort* subcounter(s) from the virtual network port firstly to the NS VL and then to the SAP (see note). The measurement producer generates the measurement for the subject SAP by assigning the value, or summing up multiple values, of the mapped PacketIncoming.*vNPort* subcounter(s).

NOTE: The mapping between NS VL and SAP has dependency on the connectivity modelling between them. The exact mapping is out of scope of the present document.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** ServiceAccessPoint.

f) **Measured Object Type:** Sap.

g) **Measurement Name:** PacketIncomingSap.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last PacketIncomingVnfExtCp or PacketIncoming.*vNPort* measurement received from NFVI for the measured SAP in the collection period.

- **MeasurementInterval:** Equals the "number of PacketIncomingVnfExtCp measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the PacketIncomingVnfExtCp measurements" received from NFVI for the measured SAP in the collection period, or the "number of PacketIncoming.*vNPort* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the PacketIncoming.*vNPort* measurements" received from NFVI for the measured SAP in the collection period.

- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last PacketIncomingVnfExtCp or PacketIncoming.*vNPort* measurement received from NFVI for the measured SAP in the collection period is generated.

7.3.5 Number of outgoing packets of SAP

a) **Description:** This measurement provides the number of outgoing packets transmitted by an SAP of an NS instance.

b) **Collection Method:** OM.

c) **Trigger:**

- The measurement is triggered based on the mapping of the measured SAP.
- In case the SAP is mapped to a VNF external CP:
  - The measurement producer receives one or more PacketOutgoingVnfExtCp measurement(s) (see clause 7.2.15) in the collection period, and maps the received PacketOutgoingVnfExtCp measurement(s) from the VNF external CP to SAP. The measurement producer generates the measurement for the subject SAP by assigning the value, or summing up multiple values, of the mapped PacketOutgoingVnfExtCp measurement(s).
- In case the SAP is mapped to an NS VL:
  - The measurement producer receives one or more ByteOutgoing.*vNPort* measurement(s) (see clause 7.1.15) for the virtual network port(s) in the collection period, and maps the received ByteOutgoing.*vNPort* subcounter(s) from the virtual network port firstly to the NS VL and then to the SAP (see note). The measurement producer generates the measurement for the subject SAP by assigning the value, or summing up multiple values, of the mapped ByteOutgoing.*vNPort* subcounter(s).

NOTE: The mapping between NS VL and SAP has dependency on the connectivity modelling between them. The exact mapping is out of scope of the present document.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** ServiceAccessPoint.

f) **Measured Object Type:** Sap.

g) **Measurement Name:** PacketOutgoingSap.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last PacketOutgoingVnfExtCp or ByteOutgoing.*vNPort* measurement received from NFVI for the measured SAP in the collection period.
- **MeasurementInterval:** Equals the "number of PacketOutgoingVnfExtCp measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the PacketOutgoingVnfExtCp measurements" received from NFVI for the measured SAP in the collection period, or the "number of ByteOutgoing.*vNPort* measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the ByteOutgoing.*vNPort* measurements" received from NFVI for the measured SAP in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last PacketOutgoingVnfExtCp or ByteOutgoing.*vNPort* measurement received from NFVI for the measured SAP in the collection period is generated.

7.3.6 Energy consumption of NS instance

NOTE 1: In the present document version, this measurement only considers the computation of energy consumption based on the energy consumptions of VNF instances. Other NS constituents are not considered in the present document version.

a) **Description:** This measurement provides the total energy consumption related to an NS instance, based on the energy consumption of the VNF instances constituting the NS instance, over the collection period.

b) **Collection Method:** OM.

c) **Trigger:** the measurement producer receives one or more energy consumptions measurement(s) (see clause 7.2.18) for the VNF instances that constitute the NS instance in the collection period, and maps the received EnergyConsumptionVnf measurement(s) to the combination of NS instance and VNF instance. The measurement producer then generates the "total energy consumption of VNF instance over the collection" period by summing up the multiple values of mapped EnergyConsumptionVnf measurements per VNF instance. Finally, the measurement producer generates the measurement for the subject NS instance by summing up the "total energy consumption of VNF instance over the collection period" considering the whole set of VNF instances that constitute the NS instance.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** NetworkService.

f) **Measured Object Type:** Ns.

g) **Measurement Name:** EnergyConsumptionNs.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from the measurement producer of EnergyConsumptionVnf measurements.
- **MeasurementInterval:** Equals the measurement interval utilized by the measurement producer of EnergyConsumptionVnf (e.g. 1 second) for providing energy consumption measurements in the collection period.
- **MeasurementWindow:** Equals the "number of EnergyConsumptionVnf measurements" x "MeasurementInterval" utilized by the measurement producer of EnergyConsumptionVnf for providing energy consumption measurements in the collection period.

NOTE 2: The MeasurementEndTime, MeasurementInterval and MeasurementWindow for EnergyConsumptionVnf measurements are expected to be the same across all VNF instances that constitute the NS instance.

7.3.7 Mean energy consumption of NS instance

NOTE 1: In the present document version, this measurement only considers the computation of energy consumption based on the energy consumptions of VNF instances. Other NS constituents are not considered in the present document version.

a) **Description:** This measurement provides the mean energy consumption related to an NS instance, based on the energy consumption of the VNF instances constituting the NS instance, over the collection period.

b) **Collection Method:** SC.

c) **Trigger:** the measurement producer receives one or more mean energy consumptions measurement(s) (see clause 7.2.19) for the VNF instances that constitute the NS instance in the collection period, and maps the received EnergyConsumptionVnf measurement(s) to the combination of NS instance and VNF instance. The measurement producer then generates the "mean energy consumption of VNF instance over the collection" period by taking the arithmetic mean of mapped EnergyConsumptionMeanVnf measurements per VNF instance. Finally, the measurement producer generates the measurement for the subject NS instance by summing up the "mean energy consumption of VNF instance over the collection period" considering the whole set of VNF instances that constitute the NS instance.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** NetworkService.

f) **Measured Object Type:** Ns.

g) **Measurement Name:** EnergyConsumptionMeanNs.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from the measurement producer of EnergyConsumptionMeanVnf measurements.
- **MeasurementInterval:** Equals the measurement interval utilized by the measurement producer of EnergyConsumptionMeanVnf (e.g. 1 second) for providing energy consumption measurements in the collection period.
- **MeasurementWindow:** Equals the "number of EnergyConsumptionMeanVnf measurements" x "MeasurementInterval" utilized by the measurement producer of EnergyConsumptionMeanVnf for providing energy consumption measurements in the collection period.

NOTE 2: The MeasurementEndTime, MeasurementInterval and MeasurementWindow for EnergyConsumptionMeanVnf measurements are expected to be the same across all VNF instances that constitute the NS instance.

7.4 Performance measurements associated to OS container workloads

7.4.1 Introduction

The performance measurements defined in this clause 7.4 are applicable to the OS container workload.

The measurement producer can be, among others, a CISM.

7.4.2 Mean CPU usage on OS container workload

a) **Description:** This measurement provides the mean CPU usage on the OS container workload.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the CPU usage measurement for the OS container workload at the pre-defined interval, and then takes the arithmetic mean of the CPU usage metrics received in the collection period. The arithmetic mean shall be represented as a percentage with respect to the CPU resources request.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** McioComputeResource.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** CpuUsageMeanMcio.container, where container is equal to the resource name of the measured OS container.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last cpu usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the cpu usage measurements received by the measurement producer in the collection period.
- **ResourcesCpuLimit:** Equals to the value of CPU resources limit.
- **ResourcesCpuRequest:** Equals to the value of CPU resources request.

7.4.3 Peak CPU usage on OS container workload

a) **Description:** This measurement provides the peak CPU usage on the OS container workload.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the CPU usage measurement for the OS container workload at the pre-defined interval, and then selects the Maximum among the CPU usage metrics received in the collection period. The selected CPU usage shall be represented as a percentage with respect to the CPU resources request.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** McioComputeResource.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** CpuUsagePeakMcio.container, where container is equal to the resource name of the measured OS container.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last cpu usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the cpu usage measurements received by the measurement producer in the collection period.
- **ResourcesCpuLimit:** Equals to the value of CPU resources limit.
- **ResourcesCpuRequest:** Equals to the value of CPU resources request.

7.4.4 Mean memory usage on OS container workload

a) **Description:** This measurement provides the mean memory usage on the OS container workload.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the memory usage measurement for the OS container workload at the pre-defined interval, and then takes the arithmetic mean of the memory usage metrics received in the collection period. The arithmetic mean shall be represented as a percentage with respect to the memory resources request.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** McioComputerResource.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** MemoryUsageMeanMcio.container, where container is equal to the resource name of the measured OS container.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last memory usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the memory usage measurements received by the measurement producer in the collection period.
- **ResourcesMemoryLimit:** Equals to the value of memory resources limit.
- **ResourcesMemoryRequest:** Equals to the value of memory resources request.

7.4.5 Peak memory usage on OS container workload

a) **Description:** This measurement provides the peak memory usage on the OS container workload.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the memory usage measurement for the OS container workload at the pre-defined interval, and then selects the maximum among the memory usage metrics received in the collection period. The selected memory usage shall be represented as a percentage with respect to the memory resources request.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** McioComputeResource.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** MemoryUsagePeakMcio.container, where container is equal to the resource name of the measured OS container.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last memory usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the memory usage measurements received by the measurement producer in the collection period.
- **ResourcesMemoryLimit:** Equals to the value of memory resources limit.
- **ResourcesMemoryRequest:** Equals to the value of memory resources request.

7.4.6 Number of incoming bytes on compute MCIO

a) **Description:** This measurement provides the number of bytes received on the compute MCIO. This measurement is split into subcounters per virtual network interface.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more Rxbytes measurement for a virtual network interface in the collection period, and uses the last received value(s) to generate the measurement of the MCIO instance.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** McioNetworkInterface.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** NetByteIncomingMcio. *vNetItfId*, where *vNetItfId* is equal to the name of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Rxbytes measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the Rxbytes measurements received by the measurement producer in the collection period.

7.4.7 Number of outgoing bytes on compute MCIO

a) **Description:** This measurement provides the number of bytes transmitted on the compute MCIO. This measurement is split into subcounters per virtual network interface.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more TxBytes measurement for a virtual network interface in the collection period, and uses the last received value(s) to generate the measurement of the MCIO instance.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** McioNetworkInterface.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** NetByteOutgoingMcio. *vNetItfId,* where *vNetItfId* is equal to the name of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last TxBytes measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the TxBytes measurements received by the measurement producer in the collection period.

7.4.8 Number of incoming packets on compute MCIO

a) **Description:** This measurement provides the number of packets received on the compute MCIO. This measurement is split into subcounters per virtual network interface.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more RxPackets measurement for a virtual network interface in the collection period, and uses the last received value(s) to generate the measurement of the MCIO instance.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** McioNetworkInterface.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** NetPacketIncomingMcio. *vNetItfId,* where *vNetItfId* is equal to the name of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time of the last Rx_Packets measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to"window" (see Annex A1) of the Rx_Packets measurements received by the measurement producer in the collection period.

7.4.9 Number of outgoing packets on compute MCIO

a) **Description:** This measurement provides the number of packets transmitted on the compute MCIO. This measurement is split into subcounters per virtual network interface.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more TxPackets measurement for a virtual network interface in the collection period, and uses the last received value(s) to generate the measurement of the MCIO instance.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** McioNetworkInterface.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** NetPacketOutgoingMcio. *vNetItfId,* where *vNetItfId* is equal to the name of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Tx_Packets measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the Tx_Packets measurements received by the measurement producer in the collection period.

7.4.10 Mean usage of storage resource on OS container workload

a) **Description:** This measurement provides the mean usage of storage resource on the OS container workload.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the storage resource usage measurement for the OS container workload at the pre-defined interval, and then takes the arithmetic mean of the storage resource usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** McioStorageResource.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** StorageUsageMeanMcio.container, where container is equal to the resource name of the measured OS container.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last storage resource usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the storage resource usage measurements received by the measurement producer in the collection period.

7.4.11 Peak usage of storage resource on OS container workload

a) **Description:** This measurement provides the peak usage of storage resource on the OS container workload.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the storage resource usage measurement for the OS container workload at the pre-defined interval, and then takes the arithmetic peak of the storage resource usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** McioStorageResource.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** StorageUsagePeakMcio.container, where container is equal to the resource name of the measured OS container.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last storage resource usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the storage resource usage measurements received by the measurement producer in the collection period.

7.4.12 Energy consumption on OS container workload

a) **Description:** This measurement provides the total energy consumption of the OS container workload over the collection period.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the energy measurements for the OS container workload from CIS at the pre-defined interval, and then sums the energy measurement values received in the collection period.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** McioComputeResource.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** EnergyConsumptionMcio.[total,container], where "container" is equal to the resource name of the measured OS container, and "total" corresponds to the whole group of one or more OS containers conforming the OS container workload.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from the CIS in the collection period.
- **MeasurementInterval:** Equals the measurement interval utilized by the CIS (e.g. 1 second) for providing energy consumption measurements in the collection period.

7.4.13 Mean energy consumption on OS container workload

a) **Description:** This measurement provides the mean energy consumption of the OS container workload over the collection period.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the energy measurements for the OS container workload from CIS at the pre-defined interval, and then takes the arithmetic mean of the energy measurement values received in the collection period.

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** McioComputeResource.

f) **Measured Object Type:** Mcio-c.

g) **Measurement Name:** EnergyConsumptionMeanMcio.[total,container], where "container" is equal to the resource name of the measured OS container, and "total" corresponds to the whole group of one or more OS containers conforming the OS container workload.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from CIS in the collection period.
- **MeasurementInterval:** Equals the measurement interval utilized by the CIS (e.g. 1 second) for providing energy consumption measurements in the collection period.

7.5 Performance measurements associated to CIS cluster

7.5.1 Introduction

The performance measurements defined in this clause 7.5 are applicable to the CIS cluster.

The measurement producer can be, among others, a CCM.

7.5.2 Mean CPU usage on CIS cluster node

a) **Description:** This measurement provides the mean CPU usage on the CIS cluster node.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the CPU usage measurement for the CIS cluster node at the pre-defined interval, and then takes the arithmetic mean of the CPU usage metrics received in the collection period. The arithmetic mean shall be represented as a percentage with respect to the total CPU resources capacity of the CIS cluster node.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** CisNodeComputeResource.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** CpuUsageMeanCisNode.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last cpu usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the cpu usage measurements received by the measurement producer in the collection period.
- **ResourcesCpuCapacity:** Equals to the value of CPU resources capacity.
- **ResourcesCpuAllocated:** Equals to the value of allocated CPU resource.

7.5.3 Peak CPU usage on CIS cluster node

a) **Description:** This measurement provides the peak CPU usage on the CIS cluster node.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the CPU usage measurement for the CIS cluster node at the pre-defined interval, and then selects the Maximum among the CPU usage metrics received in the collection period. The selected CPU usage shall be represented as a percentage with respect to the total CPU resources capacity of the CIS cluster node.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** CisNodeComputeResource.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** CpuUsagePeakCisNode.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last cpu usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to"window" (see Annex A1) of the cpu usage measurements received by the measurement producer in the collection period.
- **ResourcesCpuCapacity:** Equals to the value of CPU resources capacity.
- **ResourcesCpuAllocated:** Equals to the value of allocated CPU resources.

7.5.4 Mean memory usage on CIS cluster node

a) **Description:** This measurement provides the mean memory usage on the CIS cluster node.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the memory usage measurement for the CIS cluster node at the pre-defined interval, and then takes the arithmetic mean of the memory usage metrics received in the collection period. The arithmetic mean shall be represented as a percentage with respect to the total memory resources capacity of the CIS cluster node.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** CisNodeMemoryResource.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** MemoryUsageMeanCisNode.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last memory usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the memory usage measurements received by the measurement producer in the collection period.
- **ResourcesMemoryCapacity:** Equals to the value of memory resources capacity.
- **ResourcesMemoryAllocated:** Equals to the value of allocated memory resources.

7.5.5 Peak memory usage on CIS cluster node

a) **Description:** This measurement provides the peak memory usage on the CIS cluster node.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the memory usage measurement for the CIS cluster node at the pre-defined interval, and then selects the maximum among the memory usage metrics received in the collection period. The selected memory usage shall be represented as a percentage with respect to the total memory resources Capacity of the CIS cluster node.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** CisNodeMemoryResource.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** MemoryUsagePeakCisNode.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last memory usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the memory usage measurements received by the measurement producer in the collection period.
- **ResourcesMemoryCapacity:** Equals to the value of memory resources capacity.
- **ResourcesMemoryAllocated:** Equals to the value of allocated memory resources.

7.5.6 Number of incoming bytes of a network on CIS cluster node

a) **Description:** This measurement provides the number of bytes received on the CIS cluster node. This measurement is split into subcounters per virtual network interface.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more Rxbytes measurement for virtual network interface on the CIS cluster node in the collection period, and generates the measurement by recording last received values of each virtual network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** CisNodeNetworkInterface.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** NetByteIncomingCisNode.*vNetItfId*, where *vNetItfId* is equal to the name of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Rxbytes measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the Rxbytes measurements received by the measurement producer in the collection period.

7.5.7 Number of outgoing bytes of a network on CIS cluster node

a) **Description:** This measurement provides the number of bytes transmitted on the CIS cluster node. This measurement is split into subcounters per virtual network interface.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more Txbytes measurement for virtual network interface on the CIS cluster node in the collection period, and generates the measurement by recording last received values of each virtual network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** CisNodeNetworkInterface.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** NetByteOutgoingCisNode.*vNetItfId*, where *vNetItfId* is equal to the name of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Txbytes measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the Txbytes measurements received by the measurement producer in the collection period.

7.5.8 Number of incoming packets of a network on CIS cluster node

a) **Description:** This measurement provides the number of packets received on the CIS cluster node. This measurement is split into subcounters per virtual network interface.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more RxPackets measurement for virtual network interface on the CIS cluster node in the collection period, and generates the measurement by recording last received values of each virtual network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** CisNodeNetworkInterface.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** NetRxPacketIncomingCisNode.*vNetItfId*, where *vNetItfId* is equal to the name of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Rx_Packets measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to"window" (see Annex A1) of the RxPackets measurements received by the measurement producer in the collection period.

7.5.9 Number of outgoing packets of a network on CIS cluster node

a) **Description:** This measurement provides the number of packets transmitted on the CIS cluster node. This measurement is split into subcounters per virtual network interface.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more TxPackets measurement for virtual network interface on the CIS cluster node in the collection period, and generates the measurement by recording last received values of each virtual network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** CisNodeNetworkInterface.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** NetPacketOutgoingCisNode.*vNetItfId*, where *vNetItfId* is equal to the name of the measured virtual network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last Tx_Packets measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the Tx_Packets measurements received by the measurement producer in the collection period.

7.5.10 Mean usage of storage resource on CIS cluster node

a) **Description:** This measurement provides the mean usage of storage resource on the CIS cluster node.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the storage resource usage measurement for the CIS cluster node at the pre-defined interval, and then takes the arithmetic mean of the storage resource usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** CisNodeStorageResource.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** StorageUsageMeanCisNode.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last storage resource usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the storage resource usage measurements received by the measurement producer in the collection period.

7.5.11 Peak usage of storage resource on CIS cluster node

a) **Description:** This measurement provides the mean peak of storage resource on the CIS cluster node.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the storage resource usage measurement for the CIS cluster node at the pre-defined interval, and generates the measurement by taking the maximum of multiple values of the received measurement(s).

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** CisNodeStorageResource.

f) **Measured Object Type:** CisClusterNode.

g) **Measurement Name:** StorageUsagePeakCisNode.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see Annex A1) of the last storage resource usage measurement received by the measurement producer in the collection period.
- **MeasurementInterval:** Equals to "window" (see Annex A1) of the storage resource usage measurements received by the measurement producer in the collection period.

7.6 Performance measurements associated to physical resources

7.6.1 Introduction

The performance measurements defined in this clause are applicable to physical resources. Physical resources can be of type compute, storage, network etc. The measurements defined in this clause apply accordingly to the corresponding type of physical resources.

The measurement producer can be, among others, a PIM.

7.6.2 Mean physical CPU usage

a) **Description:** This measurement provides the mean CPU usage of a physical compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the phy_cpu_utilization measurement for the physical resource at the pre-defined interval, and then takes the arithmetic mean of the CPU usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** PhysicalComputeResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** CpuUsageMean.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last phy_cpu_utilization measurement received in the collection period.
- **TickInterval:** Indicates the tick interval (see ETSI GS NFV-TST 008 [10]) of the last phy_cpu_utilization measurement received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_cpu_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the phy_cpu_utilization measurements" received in the collection period.
- **ExecutionContext:** Indicates the execution context (see ETSI GS NFV-TST 008 [10]) of the phy_cpu_utilization measurements received in the collection period. When a single "non-Idle" CPU Utilization is required, the sum of the utilization of all the non-Idle execution contexts should be reported as the "active" execution context.

7.6.3 Peak physical CPU usage

a) **Description:** This measurement provides the peak CPU usage of a physical compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the phy_cpu_utilization measurement for the physical resource at the pre-defined interval, and then selects the maximum metric among the CPU usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** PhysicalComputeResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** CpuUsagePeak.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the phy_cpu_utilization measurement with peak value received in the collection period.
- **TickInterval:** Indicates the tick interval (see ETSI GS NFV-TST 008 [10]) of the phy_cpu_utilization measurement with peak value received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_cpu_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the phy_cpu_utilization measurements" received in the collection period.

Execution Context: Indicates the execution context (see ETSI GS NFV-TST 008 [10]) of the phy_cpu_utilization measurement with peak value received in the collection period. When a single "non-Idle" CPU Utilization is required, the sum of the utilization of all the non-Idle execution contexts should be reported as the "active" execution context.

7.6.4 Mean memory usage

a) **Description:** This measurement provides the mean memory usage of a physical compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the phy_memory_utilization measurement for the physical resource at the pre-defined interval, and then takes the arithmetic mean of the memory usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** PhysicalComputeResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** PhyMemoryUsageMean.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last phy_memory_utilization measurement received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_memory_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the phy_memory_utilization measurements" received in the collection period.
- **MeasurementSystemRam:** Indicates the system RAM (see ETSI GS NFV-TST 008 [10]) of the measured physical resource.
- **MeasurementSystemSwapSpace:** Indicates the system SWAP space (see ETSI GS NFV-TST 008 [10]) of the measured physical resource.

7.6.5 Peak memory usage

a) **Description:** This measurement provides the peak memory usage of a physical compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the phy_memory_utilization measurement for the physical resource at the pre-defined interval, and then selects the maximum metric among the memory usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** PhysicalComputeResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** PhyMemoryUsagePeak.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the phy_memory_utilization measurement with peak value received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_memory_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the phy_memory_utilization measurements" received in the collection period.
- **MeasurementSystemRam:** Indicates the system RAM (see ETSI GS NFV-TST 008 [10]) of the measured physical resource.
- **MeasurementSystemSwapSpace:** Indicates the system SWAP space (see ETSI GS NFV-TST 008 [10]) of the measured physical resource.

7.6.6 Mean disk usage

a) **Description:** This measurement provides the mean disk usage of a physical compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the phy_storage_utilization measurement for the physical resource at the pre-defined interval, and then takes the arithmetic mean of the disk usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** PhysicalComputeResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** PhyDiskUsageMean.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last phy_storage_utilization measurement received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_storage_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the phy_storage_utilization measurements" received in the collection period.

7.6.7 Peak disk usage

a) **Description:** This measurement provides the peak disk usage of a physical compute resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the phy_storage_utilization measurement for the physical resource at the pre-defined interval, and then selects the maximum metric among the disk usage metrics received in the collection period.

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** PhysicalComputeResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** PhyDiskUsagePeak.

h) **Measurement Context:**

- **MeasurementEndTime:** It indicates the end time (see ETSI GS NFV-TST 008 [10]) of the phy_storage_utilization measurement with peak value received in the collection period.

MeasurementInterval: Equals the "number of phy_storage_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the phy_storage_utilization measurements" received in the collection period.

7.6.8 Number of incoming bytes on physical resource

a) **Description:** This measurement provides the number of bytes received at the physical network resource. This measurement is split into subcounters per network interface, which is the communication endpoint of a physical network resource.

NOTE: An example of physical network resource can be a Network Interface Card (NIC), which can further have multiple interfaces and ports. In that case, network related measurements, e.g. incoming/outgoing bytes and packets of the NIC physical resource also consist of the measurements related to its interfaces and ports, which are referred here as "network interface". See Annex A2.1 for more details on how network related measurements of networking components can be associated with each other in Redfish® data model.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more num_of_incoming_bytes measurement for a network interface in the collection period, and maps the received num_of_incoming_bytes measurement(s) from the network interface to the physical network resource. The measurement producer generates the measurement for the physical network resource by assigning the value of the received num_of_incoming_bytes measurement(s) to the sub-counter(s) per network interface.

d) **Measurement Unit:** Each measurement is a real value.

e) **Measurement Group:** PhysicalNetworkInterface.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** PhyNetByteIncoming.*NetItfId*, where *NetItfId* is equal to the resource identifier of the measured network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last num_of_incoming_bytes measurement received for the measured network interface in the collection period.
- **MeasurementInterval:** Equals the "number of num_of_incoming_bytes measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the num_of_incoming_bytes measurements" received for the measured network interface in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last num_of_incoming_bytes measurement received for the measured network interface in the collection period is generated.

7.6.9 Number of outgoing bytes on physical resource

a) **Description:** This measurement provides the number of bytes transmitted by the physical network resource. This measurement is split into subcounters per network interface, which is the communication endpoint of a physical network resource.

NOTE: An example of physical network resource can be a network interface card (NIC), which can further have multiple interfaces and ports. In that case, network related measurements, e.g. incoming/outgoing bytes and packets of the NIC physical resource also consist of the measurements related to its interfaces and ports, which are referred here as "network interface". See Annex A2.1 for more details on how network related measurements of networking components can be associated with each other in Redfish® data model.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more num_of_outgoing_bytes measurement for a network interface in the collection period, and maps the received num_of_outgoing_bytes measurement(s) from the network interface to the physical network resource. The measurement producer generates the measurement for the physical network resource by assigning the value of the received num_of_outgoing_bytes measurement(s) to the sub-counter(s) per network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** PhysicalNetworkInterface.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** PhyNetByteOutgoing.*NetItfId*, where *NetItfId* is equal to the resource identifier of the measured network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last num_of_outgoing_bytes measurement received for the measured network interface in the collection period.
- **MeasurementInterval:** Equals the "number of num_of_outgoing_bytes measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the num_of_outgoing_bytes measurements" received for the measured network interface in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last num_of_outgoing_bytes measurement received for the measured network interface in the collection period is generated.

7.6.10 Number of incoming packets on physical resource

a) **Description:** This measurement provides the number of packets received at the physical network resource. This measurement is split into subcounters per network interface, which is the communication endpoint of a physical network resource.

NOTE: An example of physical network resource can be a network interface card (NIC), which can further have multiple interfaces and ports. In that case, network related measurements, e.g. incoming/outgoing bytes and packets of the NIC physical resource also consist of the measurements related to its interfaces and ports, which are referred here as "network interface". See Annex A2.1 for more details on how network related measurements of networking components can be associated with each other in Redfish® data model.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more num_of_incoming_packets measurement for a network interface in the collection period, and maps the received num_of_incoming_packets measurement(s) from the network interface to the physical network resource. The measurement producer generates the measurement for the physical network resource by assigning the value of the received num_of_incoming_packets measurement(s) to the sub-counter(s) per network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** PhysicalNetworkInterface.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** PhyNetPacketIncoming.*NetItfId*, where *NetItfId* is equal to the resource identifier of the measured network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last num_of_incoming_packets measurement received for the measured network interface in the collection period.
- **MeasurementInterval:** Equals the "number of num_of_incoming_packets measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of the num_of_incoming_packets measurements" received for the measured network interface in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last num_of_incoming_packets measurement received for the measured network interface in the collection period is generated.

7.6.11 Number of outgoing packets on physical resource

a) **Description:** This measurement provides the number of packets transmitted by the physical network resource. This measurement is split into subcounters per network interface, which is the communication endpoint of a physical network resource.

NOTE: An example of physical network resource can be a network interface card (NIC), which can further have multiple interfaces and ports. In that case, network related measurements, e.g. incoming/outgoing bytes and packets of the NIC physical resource also consist of the measurements related to its interfaces and ports, which are referred here as "network interface". See Annex A2.1 for more details on how network related measurements of networking components can be associated with each other in Redfish® data model.

b) **Collection Method:** OM.

c) **Trigger:** The measurement producer receives one or more num_of_outgoing_packets measurement for a network interface in the collection period, and maps the received num_of_outgoing_packets measurement(s) from the network interface to the physical network resource. The measurement producer generates the measurement for the physical network resource by assigning the value of the received num_of_outgoing_packets measurement(s) to the sub-counter(s) per network interface.

d) **Measurement Unit:** Each measurement is an integer value.

e) **Measurement Group:** PhysicalNetworkInterface.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** PhyNetPacketOutgoing.*NetItfId*, where *NetItfId* is equal to the resource identifier of the measured network interface.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last num_of_outgoing_packets measurement received for the measured network interface in the collection period.
- **MeasurementInterval:** Equals the "number of num_of_outgoing_packets measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of num_of_outgoing_packets measurements" received for the measured network interface in the collection period.
- **MeasurementInterfaceStatus:** Indicates the interface status (see ETSI GS NFV-TST 008 [10]) when the last num_of_outgoing_packets measurement received for the measured network interface in the collection period is generated.

7.6.12 Mean usage of physical storage resource

a) **Description:** This measurement provides the mean usage of a physical storage resource.

NOTE: A physical storage resource can be storage subsystem, drive, logical volume etc., with possible association among these different resources. Storage utilization measurements for a desired resource can be produced by the measurement producer after processing relevant measurements related to associated storage resources. See Annex A2.1 for more details on how capacities and utilization can be associated with each other for different storage related resources in Redfish® data model.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more phy_storage_utilization measurements for a physical storage resource in the collection period, and generates the measurement by averaging multiple values of the received measurement(s).

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** PhysicalStorageResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** UsageMeanPhyStorage.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last phy_storage_utilization measurement for the physical storage resource received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_storage_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of phy_storage_utilization measurements" received for the physical storage resource in the collection period.

7.6.13 Peak usage of physical storage resource

a) **Description:** This measurement provides the peak usage of a physical resource.

NOTE: A physical storage resource can be a storage subsystem, drive, logical volume etc., with possible association among these different resources. Storage utilization measurements for a desired resource can be produced by the measurement producer after processing relevant measurements related to associated storage resources. See Annex A2.1 for more details on how capacities and utilization can be associated with each other for different storage related resources in Redfish® data model.

b) **Collection Method:** TF.

c) **Trigger:** The measurement producer receives one or more phy_storage_utilization measurements for a physical storage resource in the collection period, and generates the measurement by taking the maximum of multiple values of the received measurement(s).

d) **Measurement Unit:** Each measurement is a real value (Unit: %).

e) **Measurement Group:** PhysicalStorageResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** UsagePeakPhyStorage.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last phy_storage_utilization measurement for the physical storage resource received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_storage_utilization measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of phy_storage_utilization measurements" received for the physical storage resource in the collection period.

7.6.14 Energy consumption of a physical resource

a) **Description:** This measurement provides the total energy consumption of the physical resource over the collection period.

NOTE 1: Energy consumption measurement can be associated with different kinds of physical resources that consume energy and can report energy consumption metrics via corresponding sensors. Examples of these physical resources are chassis, CPU, GPU, memory, networking device, power supply, storage device, etc.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the energy_consumption measurement for the physical resource from NFVI at the pre-defined interval, and then sums the energy measurement values received in the collection period.

NOTE 2: Energy consumption measurement of a physical resource can also be determined from power_consumption metrics received over a specific time interval, based on the relation between power and energy (Energy = Power x Time).

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** PhysicalResourceEnergy.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** EnergyConsumptionPhysicalResource.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from NFVI in the collection period.

- **MeasurementInterval:** Equals the measurement interval utilized by the NFVI (e.g. 1 second) for providing energy consumption measurements in the collection period.

7.6.15 Mean energy consumption of a physical resource

a) **Description:** This measurement provides the mean energy consumption of the physical resource over the collection period.

NOTE 1: Energy consumption measurement can be associated with different kinds of physical resources that consume energy and can report energy consumption metrics via corresponding sensors. Examples of these physical resources are chassis, CPU, GPU, memory, networking device, power supply, storage device, etc.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives the energy measurements for the physical resource from NFVI at the pre-defined interval, and then takes the arithmetic mean of the energy measurement values received in the collection period.

NOTE 2: Energy consumption measurement of a physical resource can also be determined from power_consumption metrics received over a specific time interval, based on the relation between power and energy (Energy = Power x Time).

d) **Measurement Unit:** Each measurement is an integer value (Unit: J).

e) **Measurement Group:** PhysicalResourceEnergy.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** EnergyConsumptionMeanPhysicalResource.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time of the last energy consumption measurement received from NFVI in the collection period.

- **MeasurementInterval:** Equals the measurement interval utilized by the NFVI (e.g. 1 second) for providing energy consumption measurements in the collection period.

7.6.16 Mean IOPS of physical storage resource

a) **Description:** This measurement provides the mean IOPS of a physical storage resource.

NOTE: A physical storage resource can be storage subsystem, drive, logical volume etc., with possible association among these different resources. IOPS measurements for a desired physical storage resource can be produced by the measurement producer after processing relevant IOPS measurements related to storage resources that are associated with the desired physical storage resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives one or more phy_storage_iops measurements for a physical storage resource in the collection period, and generates the measurement by averaging multiple values of the received measurement(s).

d) **Measurement Unit:** Each measurement is a real value (Unit: operations per second).

e) **Measurement Group:** PhysicalStorageResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** IopsMeanPhyStorage.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last phy_storage_iops measurement for the physical storage resource received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_storage_iops measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of phy_storage_iops measurements" received for the physical storage resource in the collection period.

7.6.17 Peak IOPS of physical storage resource

a) **Description:** This measurement provides the peak IOPS of a physical resource.

NOTE: A physical storage resource can be storage subsystem, drive, logical volume etc., with possible association among these different resources. IOPS measurements for a desired physical storage resource can be produced by the measurement producer after processing relevant IOPS measurements related to storage resources that are associated with the desired physical storage resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives one or more phy_storage_iops measurements for a physical storage resource in the collection period, and generates the measurement by taking the maximum of multiple values of the received measurement(s).

d) **Measurement Unit:** Each measurement is a real value (Unit: operations per second).

e) **Measurement Group:** PhysicalStorageResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** IopsPeakPhyStorage.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last phy_storage_iops measurement for the physical storage resource received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_storage_iops measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of phy_storage_iops measurements" received for the physical storage resource in the collection period.

7.6.18 Mean throughput of physical storage resource

a) **Description:** This measurement provides the mean throughput of a physical storage resource.

NOTE: A physical storage resource can be storage subsystem, drive, logical volume etc., with possible association among these different resources. Throughput measurements for a desired physical storage resource can be produced by the measurement producer after processing relevant throughput measurements related to storage resources that are associated with the desired physical storage resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives one or more phy_storage_throughput measurements for a physical storage resource in the collection period, and generates the measurement by averaging multiple values of the received measurement(s).

d) **Measurement Unit:** Each measurement is a real value (Unit: MBs per second).

e) **Measurement Group:** PhysicalStorageResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** ThroughputMeanPhyStorage.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last phy_storage_throughput measurement for the physical storage resource received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_storage_throughput measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of phy_storage_throughput measurements" received for the physical storage resource in the collection period.

7.6.19 Peak throughput of physical storage resource

a) **Description:** This measurement provides the peak throughput of a physical storage resource.

NOTE: A physical storage resource can be storage subsystem, drive, logical volume etc., with possible association among these different resources. Throughput measurements for a desired physical storage resource can be produced by the measurement producer after processing relevant throughput measurements related to storage resources that are associated with the desired physical storage resource.

b) **Collection Method:** SC.

c) **Trigger:** The measurement producer receives one or more phy_storage_throughput measurements for a physical storage resource in the collection period, and generates the measurement by taking the maximum of multiple values of the received measurement(s).

d) **Measurement Unit:** Each measurement is a real value (Unit: MBs per second).

e) **Measurement Group:** PhysicalStorageResource.

f) **Measured Object Type:** PhysicalResource.

g) **Measurement Name:** ThroughputPeakPhyStorage.

h) **Measurement Context:**

- **MeasurementEndTime:** Indicates the end time (see ETSI GS NFV-TST 008 [10]) of the last phy_storage_throughput measurement for the physical storage resource received in the collection period.
- **MeasurementInterval:** Equals the "number of phy_storage_throughput measurements" × "measurement interval (see ETSI GS NFV-TST 008 [10]) of phy_storage_throughput measurements" received for the physical storage resource in the collection period.

8 Key performance indicators

8.1 Key performance indicators associated to virtualised resources

8.1.1 Introduction

The KPIs defined clause 8.1 are applicable to virtualised resources.

8.1.2 Energy efficiency of virtualised compute based on data volume

a) **Name:** EEVirtualComputeDataVolume.

b) **Description:**

- The KPI provides the energy efficiency of a virtualised compute based on the incoming and outgoing data volume transferred by the virtualised compute. The KPI is computed by dividing the total data volume on the virtualised compute virtual network interfaces over the energy consumption of the virtualised compute during the collection period of the respective measurements. The unit of this KPI is bytes/J. The greater the value of the KPI, the higher the energy efficiency of the virtualised compute.

c) **Formula definition:**

$$EEVirtualComputeDataVolume = \frac{\sum_{vNetItfId}(VNetByteIncoming.vNetItfId + VNetByteOutgoing.vNetItfId)}{EnergyConsumptionVirtualCompute}$$

where:

- VNetByteIncoming.vNetItfId represents the number of incoming bytes on the virtualised compute's virtual network interface identified by vNetItfId;
- VNetByteOutgoing.vNetItfId represents the number of outgoing bytes on the virtualised compute's virtual network interface identified by vNetItfId; and
- EnergyConsumptionVirtualCompute represents the energy consumption of the virtualised compute.

d) **Object:** VirtualCompute.

e) **Additional description:**

- In the present KPI, the sum of the incoming and outgoing data volume transferred represents the useful output or performance of the virtualised compute.

8.2 Key performance indicators associated to OS container workloads

8.2.1 Introduction

The KPIs defined clause 8.2 are applicable to OS container workloads.

8.2.2 Energy efficiency of OS container workload based on data volume

a) **Name:** EEMcioDataVolume.

b) **Description:**

- The KPI provides the energy efficiency of a compute MCIO of a group of one or more OS containers conforming the OS container workload based on the incoming and outgoing data volume transferred by the OS container workload. The KPI is computed by dividing the total data volume on the OS container workload virtual network interfaces over the energy consumption of the OS container workload during the collection period of the respective measurements. The unit of this KPI is bytes/J. The greater the value of the KPI, the higher the energy efficiency of the virtualised compute.

c) **Formula definition:**

$$EEMcioDataVolume = \frac{\sum_{vNetItfId}(NetByteIncomingMcio.vNetItfId + NetByteOutgoingMcio.vNetItfId)}{EnergyConsumptionMcio.total}$$

where:

- NetByteIncomingMcio.vNetItfId represents the number of incoming bytes on the compute MCIO's virtual network interface identified by vNetItfId;

- NetByteOutgoingMcio.vNetItfId represents the number of outgoing bytes on the compute MCIO's virtual network interface identified by vNetItfId; and
- EnergyConsumptionMcio.total represents the total energy consumption of the compute MCIO of a group of one or more OS containers conforming the OS container workload.

d) **Object:** Mcio-c.

e) **Additional description:**

- In the present KPI, the sum of the incoming and outgoing data volume transferred represents the useful output or performance of the OS container workload.

8.3 Key performance indicators associated to VNF instances

8.3.1 Introduction

The KPIs defined clause 8.3 are applicable to VNF instances and related sub-object types, like VNFC instances. KPIs related to VNF instances consider the two forms of deployment of VNF and VNFC instances, i.e. VM-based and container-based deployments.

8.3.2 Energy efficiency of VNF based on data volume

a) **Name:** EEVnfDataVolume.

b) **Description:**

The KPI provides the energy efficiency of a VNF instance based on the incoming and outgoing data volume transferred by the VNF instances over external CPs. The KPI is computed by dividing the total data volume on the VNF external CPs over the energy consumption of the VNF instance during the collection period of the respective measurements. The unit of this KPI is bytes/J. The greater the value of the KPI, the higher the energy efficiency of the virtualised compute.

c) **Formula definition:**

$$EEVnfDataVolume = \frac{\sum_{VnfExtCp}(ByteIncomingVnfExtCp + ByteOutgoingVnfExtCp)}{EnergyConsumptionVnf}$$

where:

- ByteIncomingVnfExtCp represents the number of incoming bytes of the VNF external CP;
- ByteOutgoingVnfExtCp represents the number of outgoing bytes of the VNF external CP; and
- EnergyConsumptionVnf represents the energy consumption of the VNF instance.

d) **Object:** Vnf.

e) **Additional description:**

- In the present KPI, the sum of the incoming and outgoing data volume transferred represents the useful output or performance of the VNF instance.

8.3.3 Energy efficiency of VNFC based on data volume

a) **Name:** EEVnfcDataVolume.

b) **Description:**

- The KPI provides the energy efficiency of a VNFC instance based on the incoming and outgoing data volume transferred by the VNFC instance. The KPI is computed by dividing the total data volume on the VNFC internal CPs over the energy consumption of the VNFC instance during the collection period of the respective measurements. The unit of this KPI is bytes/J. The greater the value of the KPI, the higher the energy efficiency of the virtualised compute.

c) **Formula definition:**

$$EEVnfcDataVolume = \frac{\sum_{VnfIntCp,Vnfc.vComputeId}(ByteIncomingVnfIntCp + ByteOutgoingVnfIntCp)}{EnergyConsumptionVnfc.vComputeId}$$

where:

- VnfIntCp,Vnfc.vComputeId represents a VNF internal CP that is part of the VNFC instance identified by Vnfc.vComputeId;
- ByteIncomingVnfIntCp represents the number of incoming bytes of the VNF internal CP;
- ByteOutgoingVnfIntCp represents the number of outgoings bytes of the VNF internal CP; and
- EnergyConsumptionVnfc.vComputeId represents the energy consumption of the VNFC identified by vComputeId.

d) **Object:** Vnfc.

e) **Additional description:**

- In the present KPI, the sum of the incoming and outgoing data volume transferred represents the useful output or performance of the VNFC instance.

8.4 Key performance indicators associated to NS instances

8.4.1 Introduction

The KPIs defined clause 8.4 are applicable to NS instances.

8.4.2 Energy efficiency of NS based on data volume

a) **Name:** EENsDataVolume.

b) **Description:**

- The KPI provides the energy efficiency of an NS instance based on the incoming and outgoing data volume transferred by the external CPs of all VNF instances that constitute the NS instance. The KPI is computed by dividing the total data volume on the VNF external CPs over the energy consumption of the NS instance during the collection period of the respective measurements. The unit of this KPI is bytes/J. The greater the value of the KPI, the higher the energy efficiency of the virtualised compute.

c) **Formula definition:**

$$EENsDataVolume = \frac{\sum_{Vnf,VnfExtCp}(ByteIncomingVnfExtCp + ByteOutgoingVnfExtCp)}{EnergyConsumptionNs}$$

where:

- Vnf,VnfExtCp represents the set of VNF external CPs of all VNF instances that constitute the NS instance;
- ByteIncomingVnfExtCp represents the number of incoming bytes of the VNF external CP;
- ByteOutgoingVnfExtCp represents the number of outgoing bytes of the VNF external CP; and

- EnergyConsumptionNs represents the energy consumption of the NS instance.

d) **Object:** Ns.

e) **Additional description:**

- In the present KPI, the sum of the incoming and outgoing data volume transferred represents the useful output or performance of the NS instance.

8.5 Key performance indicators associated with physical resources

8.5.1 Introduction

The KPIs defined clause 8.5 are applicable to physical resources in the NFVI.

8.5.2 Energy efficiency of a physical network resource based on data volume

a) **Name:** EEPhyResDataVolume.

b) **Description:**

- The KPI provides the energy efficiency of a physical network resource based on the incoming and outgoing data volume transferred by all the network interfaces of a physical network resource. The KPI is computed by dividing the total data volume over the energy consumption of the physical network resource during the collection period of the respective measurements. The unit of this KPI is Bytes/J. The greater the value of the KPI, the higher the energy efficiency of the physical network resource.

c) **Formula definition:**

$$EEPhyResDataVolume = \frac{\sum_{NetItf}(PhyNetByteIncoming.NetItfId + PhyNetByteOutgoing.NetItfId)}{EnergyConsumptionPhysicalResource}$$

where:

- NetItf represents the set of all network interfaces on the physical network resource;

- PhyNetByteIncoming.NetItfId represents the number of incoming bytes on the network interface of the physical network resource;

- PhyNetByteOutgoing.NetItfId represents the number of outgoing bytes on the network interface of the physical network resource; and

- EnergyConsumptionPhysicalResource represents the energy consumption of the physical network resource.

d) **Object:** PhysicalResource.

e) **Additional description:**

- In the present KPI, the sum of the incoming and outgoing data volume transferred represents the useful output or performance of the physical network resource.

8.5.3 Energy efficiency of a physical storage resource based on IOPS performance

a) **Name:** EEPhyResStorageIops.

b) **Description:**

- The KPI provides the energy efficiency of a physical storage resource based on its performance in terms of average read and write IOPS during active state. The KPI is computed by dividing the mean IOPS over the power consumption of the physical storage resource during the collection period of the respective measurements. The unit of this KPI is operations per second/W. The greater the value of the KPI, the higher the energy efficiency of the physical storage resource.

c) **Formula definition:**

$$EEPhyResStorageIops = \frac{IopsMeanPhyStorage}{PowerConsumptionPhysicalResource}$$

where:

- IopsMeanPhyStorage represents the average read and write IOPS of the physical storage resource during the collection period; and

- PowerConsumptionPhysicalResource represents the power consumption of the physical storage resource during the collection period while the physical storage resource is in active state. PowerConsumptionPhysicalResource measurement can be derived from the EnergyConsumptionPhysicalResource specified in clause 7.6.14 based on the relationship indicated in note 2 therein.

d) **Object:** PhysicalResource.

e) **Additional description:**

- In the present KPI, the mean read/write IOPS represents the useful output or performance of the physical storage resource.

- The present KPI is aligned with the 'Active State Periodic Energy Efficiency' metric specified in ETSI EN 303 804 [i.15].

- It is assumed that the producer of the present KPI is aware of the active states of the subject physical storage resource.

8.5.4 Energy efficiency of a physical storage resource based on throughput performance

a) **Name:** EEPhyResStorageThroughput.

b) **Description:**

- The KPI provides the energy efficiency of a physical storage resource based on its performance in terms of its data transfer rate during active state. The KPI is computed by dividing the mean throughput (data transfer rate) over the power consumption of the physical storage resource during the collection period of the respective measurements. The unit of this KPI is MBs per second/W. The greater the value of the KPI, the higher the energy efficiency of the physical storage resource.

c) **Formula definition:**

$$EEPhyResStorageThroughput = \frac{ThroughputMeanPhyStorage}{PowerConsumptionPhysicalResource}$$

where:

- ThroughputMeanPhyStorage represents the average read and write throughput of the physical storage resource during the collection period; and

- PowerConsumptionPhysicalResource represents the power consumption of the physical storage resource during the collection period while the physical storage resource is in active state. PowerConsumptionPhysicalResource measurement can be derived from the EnergyConsumptionPhysicalResource specified in clause 7.6.14 based on the relationship indicated in note 2 therein.

d) **Object:** PhysicalResource.

e) **Additional description:**

- In the present KPI, the mean read/write throughput represents the useful output or performance of the physical storage resource.

- The present KPI is aligned with the 'Active State Periodic Energy Efficiency' metric specified in ETSI EN 303 804 [i.15].

- It is assumed that the producer of the present KPI is aware of the active states of the subject physical storage resource.

8.5.5 Energy efficiency of a physical storage resource based on storage capacity

a) **Name:** EEPhyResStorageIdle.

b) **Description:**

- The KPI provides the energy efficiency of a physical storage resource during its idle state, i.e., when no operations are being performed by the resource. The KPI is computed by dividing the total capacity of the storage resource over the power consumption of the physical storage resource during the collection period of the respective measurements taken during the idle state of physical storage resource. The unit of this KPI is GBs/W. The greater the value of the KPI, the higher the energy efficiency of the physical storage resource.

c) **Formula definition:**

$$EEPhyResStorageIdle = \frac{TotalStorageCapacity}{PowerConsumptionPhysicalResource}$$

where:

- TotalStorageCapacity represents the total storage capacity (in GBs) of the physical storage resource; and

- PowerConsumptionPhysicalResource represents the power consumption of the physical storage resource during idle state.

d) **Object:** PhysicalResource.

e) **Additional description:**

- The present KPI is aligned with the 'Idle State Periodic Energy Efficiency' metric specified in ETSI EN 303 804 [i.15].

- It is assumed that the producer of the present KPI is aware of the idle states of the subject physical storage resource.

8.5.6 Energy efficiency of a composed physical resource in active state

a) **Name:** EEPhyResServActive.

b) **Description:**

- The KPI provides the energy efficiency of a composed physical resource, e.g., a physical server comprising of compute (CPU, memory) and storage resources during its active state. The KPI is computed by combining the performance to power efficiency of each component (CPU, memory, storage) of the composed physical resource during the collection period of the respective measurements. This KPI has no unit and is rather represented by a numerical value. The greater the value of the KPI, the higher the energy efficiency of the composed physical resource (physical server).

- The present KPI is based on the 'Active State Energy Efficiency' metric specified in ETSI EN 303 470 [14].

c) **Formula definition:**

$$Eff_{server} = Eff_{CPU}^{W_{CPU}} \times Eff_{Memory}^{W_{Memory}} \times Eff_{Storage}^{W_{Storage}}$$

where:

- Eff_{server} represents the energy efficiency of composed physical resource (physical server) denoted as EEPhyResServActive in the present document;

- Eff_{CPU}, Eff_{Memory} and $Eff_{Storage}$ represent the worklet efficiency scores of CPU, memory and storage workloads respectively; and

- W_{CPU}, W_{Memory} and $W_{Storage}$ are the weightings applied to the CPU, Memory and Storage worklets respectively.

d) **Object:** PhysicalResource.

e) **Additional description:**

- For more details regarding the different notations used in the formula and their values, refer to ETSI EN 303 470 [14].

- For the delivery of this KPI, the producer of the KPI is expected to possess the energy efficiency score of the composed physical resource computed by SERT®.

NOTE: SERT® is a registered trademark of Standard Performance Evaluation Corporation (SPEC).

Annex A (informative): Mapping of ETSI GS NFV-TST 008 to OpenStack® measurements

Table A.1-1 provides the mapping of ETSI GS NFV-TST 008 [10] metrics to OpenStack® compute measurements in Mitaka release [i.1].

Table A.1-1: ETSI GS NFV-TST 008 [10] to OpenStack® measurements mapping table

NFV-TST 008 metrics	OpenStack® Telemetry measurements						
	Name	**Type**	**Unit**	**Resource**	**Origin**	**Support**	**Note**
Processor Utilization	*cpu_util*	Gauge	%	instance ID	Pollster	vSphere®	Average CPU utilization
(see note 1)	*Memory* (see note 5)	Gauge	MB	instance ID	Notification	Libvirt, Hyper-V	Volume of RAM allocated to the instance
Memory Buffered, Memory Cached, Memory Free, Memory Slab (see note 2)	*memory.usage* (see note 5)	Gauge	MB	instance ID	Pollster	vSphere®	Volume of RAM used by the instance from the amount of its allocated memory
(see note 3)	*disk.allocation* (see note 6)	Gauge	B	instance ID	Pollster	Libvirt	The amount of disk occupied by the instance on the host machine
(see note 4)	*disk.usage* (see note 6)	Gauge	B	instance ID	Pollster	Libvirt	The physical size in bytes of the image container on the host
Octet Count received	*network.incoming.bytes*	Cumulative	B	interface ID	Pollster	Libvirt, Hyper-V	Number of incoming bytes
Octet Count transmitted	*network.outgoing.bytes*	Cumulative	B	interface ID	Pollster	Libvirt, Hyper-V	Number of outgoing bytes
Packet Count received	*network.incoming.packets*	Cumulative	packet	interface ID	Pollster	Libvirt, Hyper-V	Number of incoming packets
Packet Count transmitted	*network.outgoing.packets*	Cumulative	packet	interface ID	Pollster	Libvirt, Hyper-V	Number of outgoing packets

NOTE 1: The size of the memory allocated to an instance can be derived from VirtualMemory (see clause 8.4.3.5 in ETSI GS NFV-IFA 006 [2]).
NOTE 2: The size of the memory used by an instance can be computed from the sum of Memory Buffered, Memory Cached, Memory Free, and Memory Slab.
NOTE 3: The size of the disk allocated to an instance can be derived from VirtualStorageData (see clause 8.4.6.3 in ETSI GS NFV-IFA 006 [2]).
NOTE 4: No corresponding metric is defined in ETSI GS NFV-TST 008 [10].
NOTE 5: *memory.usage* and *memory* can be used to compute memory_utilization = 100 × *memory.usage*/*memory*.
NOTE 6: *disk.usage* and *disk.allocation* can be used to compute disk_utilization = 100 × *disk.usage*/*disk.allocation*.

Annex A1 (informative): Kubernetes® resource management and monitoring

A1.0 Introduction

This annex provides simplified information about resource management and monitoring, all the contents are simplified copy from Kubernetes® document and github.

A1.1 Resource Management for Pods and Containers

A1.1.1 Requests and limits

For each container, it is supported to specify resource limits and requests. For a particular resource, a Pod resource request/limit is the sum of the resource requests/limits of that type for each container in the Pod. Supported resource limits and requests includes following listed and other extended resources which is not in following list:

- spec.containers[].resources.limits.cpu
- spec.containers[].resources.limits.memory
- spec.containers[].resources.requests.cpu
- spec.containers[].resources.requests.memory
- spec.containers[].resources.limits.ephemeral-storage
- spec.containers[].resources.requests.ephemeral-storage
- spec.containers[].resources.requests.hugepages-<size>
- spec.containers[].resources.limits.hugepages-<size>

If the node where a Pod is running has enough of a resource available, it is possible (and allowed) for a container to use more resource than its request for that resource specifies. However, a container is not allowed to use more than its resource limit.

A1.1.2 Resource units in Kubernetes®

"CPU" and "memory" are each a resource type. A resource type has a base unit:

- CPU resource units

CPU resources are measured in **CPU** units. In Kubernetes®, **1 CPU unit** is equivalent to **1 physical CPU core**, or **1 virtual core**, depending on whether the node is a physical host or a virtual machine running inside a physical machine.

Fractional requests are allowed. Table A1.1.2-1 provides examples of CPU resource fractional requests.

Table A1.1.2-1: Examples of CPU resource fractional requests

```
Comparing to request 1 CPU: spec.containers[].resources.requests.cpu = 1

spec.containers[].resources.requests.cpu = 0.5.    It means half as much CPU time is
requested.

spec.containers[].resources.requests.cpu = 0.1 or 100m.    It means 1/10 as much CPU time
is requested. It can be read as "one hundred millicpu", or "one hundred millicores".
```

CPU resource is always specified as an absolute amount of resource, never as a relative amount. For example, 500 m CPU represents the roughly same amount of computing power whether that container runs on a single-core, dual-core, or 48-core machine.

- Memory resource units

Memory is measured in **bytes**. Memory can be expressed as a plain integer or as a fixed-point number using one of these quantity suffixes: E, P, T, G, M, k. The power-of-two equivalents can also be used: Ei, Pi, Ti, Gi, Mi, Ki.

A1.2 Monitoring resource usage

A1.2.1 Measuring resource usage

- CPU

CPU is reported as the average core usage measured in cpu units. This value is derived by taking a rate over a cumulative CPU counter provided by the kernel (in both Linux® and Windows kernels). The time window used to calculate CPU is shown under window field in Metrics API.

- Memory

Memory is reported as the working set, measured in bytes, at the instant the metric was collected. In an ideal world, the "working set" is the amount of memory in-use that cannot be freed under memory pressure.

A1.2.2 Monitoring and reporting resource usage

The kubelet gathers metric statistics at the node, volume, pod and container level, and emits this information in the Summary API. The kubelet reports the resource usage of a Pod as part of the Pod status. If optional tools for monitoring [i.6] are available in cluster, then Pod resource usage can be retrieved either from the Metrics API directly or from monitoring tools.

In CNCF Landscape, there are a number of monitoring projects that can work with Kubernetes® by scraping metric data and using that to help observe a cluster. In a production environment, Prometheus Server or some other metrics scraper can be configured to periodically gather metrics and make them available in some kind of time series database.

NOTE: The details of selection metric reporting solution and implementation is out of the present document.

A1.2.3 Examples of getting resource usage

Table A1.2.3-1 provides an example of using Metric API to retrieve metrics of a specific pod [i.4].

Table A1.2.3-1: Using Metric API to retrieve metrics of a specific pod

request:
curl http://localhost:8080/apis/metrics.k8s.io/v1beta1/namespaces/kube-system/pods/kube-scheduler-minikube

response:

```
{
  "kind": "PodMetrics",
  "apiVersion": "metrics.k8s.io/v1beta1",
  "metadata": {
    "name": "kube-scheduler-minikube",
    "namespace": "kube-system",
    "selfLink": "/apis/metrics.k8s.io/v1beta1/namespaces/kube-system/pods/kube-scheduler-
minikube",
    "creationTimestamp": "2022-01-27T19:25:00Z"
  },
  "timestamp": "2022-01-27T19:24:31Z",
  "window": "30s",
  "containers": [
    {
      "name": "kube-scheduler",
      "usage": {
        "cpu": "9559630n",
        "memory": "22244Ki"
      }
    }
  ]
}
```

Table A1.2.3-2 provides an example of using API to retrieve information of a specific cluster node [i.3].

Table A1.2.3-2: Using API to retrieve information of a specific cluster node

request:
kubectl describe nodes e2e-test-node-pool-4lw4

response:

```
Name:         e2e-test-node-pool-4lw4
[ ... lines removed for clarity ...]
Capacity:
 cpu:                    2
 memory:                 7679792Ki
 pods:                   110
Allocatable:
 cpu:                    1 800 m
 memory:                 7474992Ki
 pods:                   110
[ ... lines removed for clarity ...]
Non-terminated Pods:      (5 in total)
  Namespace      Name                          CPU Requests    CPU Limits     Memory Requests    Memory Limits
    ---------      ----                          ------------    ----------     ---------------    -------------
  kube-system    fluentd-gcp-v1.38-28bv1       100 m (5 %)     0 (0 %)        20 0Mi (2 %)       200 Mi (2 %)
  kube-system    kube-dns-3297075139-61lj3     260 m (13 %)    0 (0 %)        100 Mi (1 %)       170 Mi (2 %)
  kube-system    kube-proxy-e2e-test-...       100 m (5 %)     0 (0 %)        0 (0 %)            0 (0 %)
  kube-system    monitoring-influxdb-grafana   200 m (10 %)    200 m (10 %)   600 Mi (8 %)       600 Mi (8 %)
  kube-system    node-problem-detector-v0.1    20 m (1 %)      200 m (10 %)   20 Mi (0 %)        100 Mi (1 %)
Allocated resources:
  (Total limits can be over 100 percent, i.e. overcommitted.)
  CPU Requests    CPU Limits    Memory Requests      Memory Limits
  ------------    ----------    ---------------      -------------
  680 m (34 %)    400 m (20 %)  920 Mi (11 %)        1 070 Mi (13 %)
```

Table A1.2.3-3 provides examples of resource related metric definition from stable metrics list [i.5].

Table A1.2.3-3: Resource related metric definition from stable metrics list

Metric Name	Type	Help
container_cpu_usage_seconds_total	Custom	Cumulative cpu time consumed by the container in core-seconds.
container_memory_working_set_bytes	Custom	Current working set of the container in bytes.
pod_cpu_usage_seconds_total	Custom	Cumulative cpu time consumed by the pod in core-seconds.
pod_memory_working_set_bytes	Custom	Current working set of the pod in bytes.
kube_pod_resource_request	Custom	Resources requested by workloads on the cluster, broken down by pod. This shows the resource usage the scheduler and kubelet expect per pod for resources along with the unit for the resource if any.
kube_pod_resource_limit	Custom	Resources limit for workloads on the cluster, broken down by pod. This shows the resource usage the scheduler and kubelet expect per pod for resources along with the unit for the resource if any.
node_cpu_usage_seconds_total	Custom	Cumulative cpu time consumed by the node in core-seconds.
node_memory_working_set_bytes	Custom	Current working set of the node in bytes.

Table A1.2.3-4 provides an example for NodeStats and PodStats definition [i.7].

Table A1.2.3-4: NodeStats and PodStats definition

type NodeStats struct	{
	[... lines removed for clarity ...]
	// Stats pertaining to CPU resources.
	CPU *CPUStats `json:"cpu,omitempty"`
	
	// Stats pertaining to memory (RAM) resources.
	Memory *MemoryStats `json:"memory,omitempty"`
	
	// Stats pertaining to network resources.
	Network *NetworkStats `json:"network,omitempty"`
	
	// Stats pertaining to total usage of filesystem resources on the rootfs used by node k8s components.
	// NodeFs.Used is the total bytes used on the filesystem.
	Fs *FsStats `json:"fs,omitempty"`
	[... lines removed for clarity ...]
	}

type PodStats struct	{
	[... lines removed for clarity ...]
	
	// Stats of containers in the measured pod.
	Containers []ContainerStats `json:"containers" patchStrategy:"merge" patchMergeKey:"name"`
	// Stats pertaining to CPU resources consumed by pod cgroup (which includes all containers' resource usage and pod overhead).
	CPU *CPUStats `json:"cpu,omitempty"`
	
	// Stats pertaining to memory (RAM) resources consumed by pod cgroup (which includes all containers' resource usage and pod overhead).
	Memory *MemoryStats `json:"memory,omitempty"`
	
	// Stats pertaining to network resources.
	Network *NetworkStats `json:"network,omitempty"`
	
	// Stats pertaining to volume usage of filesystem resources.
	// VolumeStats.UsedBytes is the number of bytes used by the Volume
	VolumeStats []VolumeStats `json:"volume,omitempty" patchStrategy:"merge" patchMergeKey:"name"`
	// EphemeralStorage reports the total filesystem usage for the containers and emptyDir-backed volumes in the measured Pod.
	EphemeralStorage *FsStats `json:"ephemeral-storage,omitempty"`
	}

type ContainerStats struct	{
	[... lines removed for clarity ...]
	// Stats pertaining to CPU resources.
	CPU *CPUStats `json:"cpu,omitempty"`
	
	// Stats pertaining to memory (RAM) resources.
	Memory *MemoryStats `json:"memory,omitempty"`
	
	// Stats pertaining to container rootfs usage of filesystem resources.
	// Rootfs.UsedBytes is the number of bytes used for the container write layer.
	Rootfs *FsStats `json:"rootfs,omitempty"`
	// Stats pertaining to container logs usage of filesystem resources.
	// Logs.UsedBytes is the number of bytes used for the container logs.
	Logs *FsStats `json:"logs,omitempty"`
	
	// User defined metrics that are exposed by containers in the pod. Typically, only one container in the pod is expected to be exposing user defined metrics. In the event of multiple containers exposing metrics, they will be combined here.
	UserDefinedMetrics []UserDefinedMetric `json:"userDefinedMetrics,omitempty" patchStrategy:"merge" patchMergeKey:"name"`
	}

type CPUStats struct	// CPUStats contains data about CPU usage.
	{
	// The time at which these stats were updated.
	Time metav1.Time `json:"time"`
	// Total CPU usage (sum of all cores) averaged over the sample window.
	// The "core" unit can be interpreted as CPU core-nanoseconds per second.
	UsageNanoCores *uint64 `json:"usageNanoCores,omitempty"`
	// Cumulative CPU usage (sum of all cores) since object creation.
	UsageCoreNanoSeconds *uint64 `json:"usageCoreNanoSeconds,omitempty"`
	}

type MemoryStats struct	// MemoryStats contains data about memory usage.
	{
	// The time at which these stats were updated.
	Time metav1.Time `json:"time"`
	// Available memory for use. This is defined as the memory limit - workingSetBytes.
	// If memory limit is undefined, the available bytes is omitted.
	AvailableBytes *uint64 `json:"availableBytes,omitempty"`
	// Total memory in use. This includes all memory regardless of when it was accessed.
	UsageBytes *uint64 `json:"usageBytes,omitempty"`
	// The amount of working set memory. This includes recently accessed memory,
	// dirty memory, and kernel memory. WorkingSetBytes is <= UsageBytes
	WorkingSetBytes *uint64 `json:"workingSetBytes,omitempty"`
	
	// The amount of anonymous and swap cache memory (includes transparent
	// hugepages).
	RSSBytes *uint64 `json:"rssBytes,omitempty"`
	// Cumulative number of minor page faults.
	PageFaults *uint64 `json:"pageFaults,omitempty"`
	// Cumulative number of major page faults.
	MajorPageFaults *uint64 `json:"majorPageFaults,omitempty"`
	}

type NetworkStats struct	// NetworkStats contains data about network resources.
	{
	// The time at which these stats were updated.
	Time metav1.Time `json:"time"`
	
	// Stats for the default interface, if found
	InterfaceStats `json:",inline"`
	
	Interfaces []InterfaceStats `json:"interfaces,omitempty"`
	}

type InterfaceStats struct	// InterfaceStats contains resource value data about interface.
	{
	    // The name of the interface
	    Name string `json:"name"`
	    // Cumulative count of bytes received.
	    RxBytes *uint64 `json:"rxBytes,omitempty"`
	    // Cumulative count of receive errors encountered.
	    RxErrors *uint64 `json:"rxErrors,omitempty"`
	    // Cumulative count of bytes transmitted.
	    TxBytes *uint64 `json:"txBytes,omitempty"`
	    // Cumulative count of transmit errors encountered.
	    TxErrors *uint64 `json:"txErrors,omitempty"`
	}

Table A1.2.3-5 provides an example for network interface statistics definition in Linux® [i.8].

Table A1.2.3-5: Network interface statistics definition

struct rtnl_link_stats64	{
	  __u64 rx_packets;
	  __u64 tx_packets;
	  __u64 rx_bytes;
	  __u64 tx_bytes;
	  __u64 rx_errors;
	  __u64 tx_errors;
	  __u64 rx_dropped;
	  __u64 tx_dropped;
	  __u64 multicast;
	  __u64 collisions;
	  __u64 rx_length_errors;
	  __u64 rx_over_errors;
	  __u64 rx_crc_errors;
	  __u64 rx_frame_errors;
	  __u64 rx_fifo_errors;
	  __u64 rx_missed_errors;
	  __u64 tx_aborted_errors;
	  __u64 tx_carrier_errors;
	  __u64 tx_fifo_errors;
	  __u64 tx_heartbeat_errors;
	  __u64 tx_window_errors;
	  __u64 rx_compressed;
	  __u64 tx_compressed;
	  __u64 rx_nohandler;
	}

Annex A2 (informative): Mapping of PIM metrics to Redfish® and OpenStack® metrics

A2.1 PIM metrics and corresponding Redfish® metrics

Table A2.1-1 provides the mapping of the PIM related metrics to corresponding Redfish® metrics as specified in Redfish® data model specification [i.10]. The description of Redfish® metrics in Table A2.1-1 follows the format "<Redfish® schema object name>.<Property name>" according to the names of schema objects and corresponding properties defined in Redfish® data model specification [i.10].

Table A2.1-1: Mapping of PIM related metrics to Redfish® metrics

PIM Metric	Redfish metric	Unit	Redfish Resources	Note
Physical CPU Utilization	*ProcessorMetrics.BandwidthPercent*	%	Processor	CPU utilization of a processor as a percentage.
Physical Memory Utilization	*MemoryMetrics.CapacityUtilizationPercent*	%	Processor.Memory, Memory	Memory capacity utilization as a percentage.
Physical Storage Utilization	*Capacity.Data.AllocatedBytes* *Capacity.Data.ConsumedBytes (see note 1)*	Bytes	Volume	Indicates total and available capacity of a storage resource, particularly a volume (see note 2).
Number of incoming bytes	*NetworkAdapterMetrics.RXBytes* *NetworkDeviceFunctionMetrics.RXBytes* *PortMetrics.RXBytes*	Bytes	NetworkAdapter, NetworkDeviceFunction, Port	Total number of bytes received on a network adapter (NIC), and its logical interface and/or port (see note 3).
Number of outgoing bytes	*NetworkAdapterMetrics.TXBytes* *NetworkDeviceFunctionMetrics.TXBytes* *PortMetrics.TXBytes*	Bytes	NetworkAdapter, NetworkDeviceFunction, Port	Total number of bytes transmitted on a network adapter (NIC), and its logical interface and/or port (see note 3).
Number of incoming packets	No correspondence			
Number of outgoing packets	No correspondence			
Power consumption	*EnvironmentMetrics.PowerWatt*	W	See note 4.	Total power consumption of a Redfish® resource in Watts.
Energy consumption	*EnvironmentMetrics.EnergyJoules* *EnvironmentMetrics.EnergykWh*	J, kWh	See note 4.	Total energy consumption of a Redfish® resource in Joules or kWh (see note 5).

NOTE 1: Capacity utilization can be calculated from the values of consumed bytes and allocated bytes.
NOTE 2: In Redfish schema, a storage resource consists of one or multiple drives and each drive can further contain volumes either wholly or partially. The capacity utilization metrics are only accessible at the volume level granularity. The overall capacity utilization for a drive or storage device can be inferred from considering capacities and utilizations of all volumes associated with a drive and/or a storage resource.
NOTE 3: The NetworkAdapter, NetworkDeviceFunction and Port resources can be associated to each other, e.g. a NIC is represented by NetworkAdapter schema, and its physical ports and logical interfaces are represented by the Port and NetworkDeviceFunction schemas respectively.
NOTE 4: The EnvironmentMetrics schema can be linked to several Redfish resources and can report power and energy consumption information of those resources. These metrics report sensor data and the sensors can be associated with various physical devices and equipment, e.g. chassis, CPU, GPU, memory, networking device, power supply, storage device, etc.
NOTE 5: There are two separate properties of the EnvironmentMetrics schema that can provide reading of energy consumption both in Joules and kWh. The EnergyJoules property is used to report device-level energy consumption measurements whereas EnergykWh property is used for reporting large-scale consumption measurements.

A2.2 PIM metrics and corresponding OpenStack® measurements

Table A2.2-1 provides the mapping of the PIM related metrics with OpenStack® measurements supported in Mitaka release [i.1]. These measurements are collected by the OpenStack® Bare Metal service using the IPMI driver and then the Bare Metal service exposes them to the OpenStack® Telemetry service.

Table A2.2-1: Mapping of PIM related metrics to OpenStack® measurements

PIM Metric	Measurement Name	Type	Unit	Resource	Note
Physical CPU Utilization	*hardware.ipmi.node.cpu_util*	Gauge	%	host ID	CPU CUPS utilization of the system.
Physical Memory Utilization	*hardware.ipmi.node.mem_util*	Gauge	%	host ID	Memory CUPS utilization of the system.
Physical Storage Utilization	*hardware.ipmi.node.io_util*	Gauge	%	host ID	IO CUPS utilization of the system.
Number of incoming bytes	No correspondence.				
Number of outgoing bytes	No correspondence.				
Number of incoming packets	No correspondence.				
Number of incoming packets	No correspondence.				
Power consumption	*hardware.ipmi.node.power*	Gauge	W	host ID	Current power of the system.
Energy consumption	No correspondence.				

Annex B (informative): Security and Regulatory Concerns

B.1 Risk analysis and assessment

Table B.1-1 is the output of the Threat, Risk, and Vulnerability Analysis according to ETSI GS NFV-SEC 006 [i.12].

Table B.1-1: Threat, Risk, and Vulnerability Analysis

A Security Environment		
a.1	**Assumptions**	
a.1.1	A new set of management and orchestration functions in addition to existing EM and OSS functions is introduced by NFV. Such NFV-MANO functions have the role to manage the NFVI to control the collection of virtualised resources consumed by VMs.	See clause 4.1.
a.1.2	NFV-MANO management functions include performance Management that is capable of controlling and collecting performance metrics.	See clause 4.1.
a.1.3	An NFV-MANO service is one or more capabilities that are offered via NFV-MANO functional blocks (i.e. NFVO, VNFM, and VIM) and invoked using a defined interface.	See clause 4.1.
a.2	**Assets**	
a.2.1	NFVO: it is responsible for processing the VNFM and VIM performance metrics to generate the NS related performance metrics to be sent to OSS/BSS.	See clauses 4.1 and 7.3.
a.2.2	VNFM: it is responsible for processing the VIM performance metrics to generate the VNF related performance measurements to be sent to EM and NFVO.	See clauses 4.1 and 7.2.
a.2.3	VIM: it is responsible for processing the NFVI performance metrics to generate the Virtualised resource related performance measurements to be sent to NFVO and VNFM.	See clauses 4.1 and 7.1.
a.2.4	NFVI: it is responsible for collecting the NFVI performance metrics, and reporting them to VIM.	See clause 4.1.
a.2.5	Performance metrics: performance information that need to be reported/acquired.	See clauses 4 and 7.
a.3	**Threat agents**	
a.3.1	Unauthorized user of assets (e.g. reports, notifications, queries, fault information, resource information).	
a.3.2	(Industrial) espionage agent.	
a.3.3	Sabotage agent.	
a.3.4	Internal threat agent, e.g. corrupt employee.	
a.4	**Threats**	
a.4.1	Unauthorized read (viewing/copying/consuming of data and interfaces).	Refer to all threat agents a.3. Refer to all assets in a.2.
a.4.2	Unauthorized write action (Masquerade ("spoofing"), Forgery, Loss or corruption of information).	Refer to all threat agents a.3. Refer to all assets in a.2.
a.4.3	Unauthorized access.	Refer to threat agents a.3.1, a.3.2 and a.3.3. Refer to all assets in a.2.
a.4.4	Repudiation (end point and threat agents).	Refer to threat agent a.3.1, a.3.2, a.3.3, and a.3.4. Refer to all assets a.2.
a.4.5	Denial of service.	Refer to threat agents a.3.1, a.3.2 and a.3.3. Refer to all assets in a.2.

B Security Objectives		
b.1	**Security objectives for the asset**	
b.1.1	The system should ensure that only authorized and authenticated entities can access (read or write) the provided interfaces and that data is exchanged in a confidential manner. Therefore, requirements for access controls and communications security (see clauses 8.5 and 8.6 in ETSI GS NFV-SEC 012 [i.14]) should be followed.	
b.1.2	The system should ensure the authenticity and integrity of all data exchanged on the interfaces. Therefore, requirements for authentications controls (see clause 8.4 in ETSI GS NFV-SEC 012 [i.14]) should be followed.	
b.1.3	The system should prevent replay of any data. Therefore, requirements for authentications controls (see clause 8.4 in ETSI GS NFV-SEC 012 [i.14]) should be followed.	
b.1.4	The system should be accountable for the data provided, that is why the system should ensure collected data (e.g. performance metrics, timestamps) is authentic.	
b.1.5	The system should ensure that interception is possible where required to support regulatory requirements (such as Lawful Interception ETSI GS NFV-SEC 004 [i.11] and Retained Data ETSI GS NFV-SEC 010 [i.13]) and not possible otherwise.	
b.1.6	The system should provide means to detect and mitigate denial of service attacks.	

Annex C (normative): Other definition templates

C.1 KPI definition template

This clause defines the template to be used to describe KPIs.

a) **Name (mandatory):**

- This field contains the name of the KPI.

b) **Description (mandatory):**

- This field contains the description of the KPI. The description should describe if the KPI focuses on the perspective of the managed object or the perspective of the consumer of the managed object. This field should also describe the formula to derive the KPI. This field should also show the KPI unit (e.g. kbit/s, millisecond) and the KPI type (e.g. mean, ratio).

c) **Formula definition (optional):**

- This field should contain the KPI formula in mathematical form by referencing and using the names of measurements specified in the present document or in external references.

d) **Object (mandatory):**

- This field shall contain the name of the measured object type for which the KPI is applicable.

e) **Additional description (optional):**

- This field can be used for providing additional information needed for the KPI definition.

Annex D (informative): Change history

Date	Version	Information about changes
2016-12	0.1.0	Implemented NFVIFA-F2F#43 approved contributions; NFVIFA(16)0001456r2, NFVIFA(16)0001457r1.
2017-06	0.2.0	Implemented approved contributions: NFVIFA(17)000333r1, NFVIFA(17)000335r1, NFVIFA(17)000336r1, NFVIFA(17)000384r1, NFVIFA(17)000385r2, NFVIFA(17)000386r3, NFVIFA(17)000521r1.
2017-08	0.3.0	Implemented approved contributions: NFVIFA(17)000602r2, NFVIFA(17)000603r6.
2017-08	0.4.0	Implemented approved contributions: NFVIFA(17)000604r5, NFVIFA(17)000605r5, NFVIFA(17)000708.
2017-09	0.5.0	Implemented approved contributions: NFVIFA(17)000728r1, NFVIFA(17)000780r2, NFVIFA(17)000781r2, NFVIFA(17)000782r2.
2017-09	0.5.1	Implemented the changes of the approved contributions that were not implemented in 0.5.0: NFVIFA(17)000728r1, NFVIFA(17)000781r2, NFVIFA(17)000782r2.
2017-10	0.6.0	Implemented the changes of the approved contributions: NFVIFA(17)000606r7, NFVIFA(17)000857r1.
2017-11	0.7.0	Implemented the changes of the approved contributions: NFVIFA(17)001023, NFVIFA(17)001024, NFVIFA(17)001042, NFVIFA(17)001045.
2017-12	0.8.0	Implemented the changes of the approved contributions: NFVIFA(17)0001043r3, NFVIFA(17)0001044r3.
2018-02	0.9.0	Implemented the changes of the approved contribution: NFVIFA(18)0000062r2.
2018-04	0.9.1	Implemented some editorial changes, and leftover changes from NFVIFA(18)0000062r2, and the changes of the approved contributions: NFVIFA(18)000261, NFVIFA(18)000194r4, NFVIFA(18)000204r1, NFVIFA(18)000330r1.
2019-11	2.4.3	Implemented the changes of the approved contribution: NFVIFA(19)000720, NFVIFA(19)000721, NFVIFA(19)000722.
2020-02	3.0.1	Implemented the change #3 and #4 in NFVIFA(19)000721 Rapporteur action: aligning "VdiskUsage*" and "Measured Object Type: SAP" to naming conventions.
2020-11	4.0.2	Implemented the changes of the approved contribution: NFVIFA(20)000582r1.
2022-05	4.3.2	Starting version for 2H2022 specification work. Unmodified with respect to published version v4.3.1.
2022-08	4.3.3	Implemented the changes of the approved contribution: NFVIFA(22)00497.

Date	Version	Information about changes
2023-04	4.4.2	Starting version for 1H2023 specification work. Unmodified with respect to published version v4.4.1.
2023-05	4.4.3	Implemented the changes of the approved contribution: NFVIFA(23)000339r2, NFVIFA(23)000340r2, NFVIFA(23)000341r3, NFVIFA(23)000342r3.
2023-09	4.4.4	Implemented the changes of the approved contribution: NFVIFA(23)000630r2, NFVIFA(23)000631r1, NFVIFA(23)000632r1, NFVIFA(23)000633r1, NFVIFA(23)000634r1, NFVIFA(23)000635r2, NFVIFA(23)000636r2, NFVIFA(23)000637r1.
2023-11	5.0.1	Release 5 initial draft version. Unmodified with respect to version 4.4.4.
2024-01	5.0.2	Implemented the changes of the approved contribution NFVIFA(23)000830r1.
2024-03	5.0.3	Implemented the changes of the approved contribution: NFVIFA(24)000204r1, NFVIFA(24)000184. Rapporteur action: aligning styles in different sub-bullets; adding a missed sub-bullet name in clause 7.2.15 bullet h).
2024-04	5.0.4	Implemented the changes of the contribution NFVIFA(24)000245. Rapporteur action: deleting redundant blanks in clause 7.4.2 and 7.5.2, deleting redundant point in clause 7.4.11 and updating "Contents" part.
2024-07	5.1.2	Initial ed521 draft version created from published version 5.1.1.
2024-07	5.1.3	Implemented the changes of the contribution NFVIFA(24)000400r1.
2024-08	5.1.4	NFVIFA(24)000446r1 FEAT29 IFA027ed521 MegaCR VNF and NS level measurements.
2025-02	5.2.2	Initial ed531 draft version created from published version 5.2.1.
2025-04	5.2.3	NFVIFA(25)000080 FEAT29 IFA027ed531 MegaCR and NFVIFA(25)000061 FEAT33 IFA027ed531 MegaCR.
2025-05	5.2.4	Implemented the changes of the contribution NFVIFA(25)000124r1.
2025-09	5.3.2	Derived from v5.3.1, initial draft version in ed541 with including the following contributions until IFA#442 F2F meeting in NFV#51: NFVIFA(25)000212r1, NFVIFA(25)000213r1.

History

Version	Date	Status
V5.1.1	June 2024	Publication
V5.2.1	September 2024	Publication
V5.3.1	August 2025	Publication
V5.4.1	February 2026	Publication