

Test v0.1 (Current)

Version 0.1

[Publication Date]

This is a Non-binding Permanent Reference Document of the GSMA

Security Classification: Non-confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is confidential to the Association and is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Copyright © 2020 GSM Association

Disclaimer

The GSM Association (“Association”) makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Antitrust Notice

The information contain herein is in full compliance with the GSM Association’s antitrust compliance policy.

Table of Contents

[1 Introduction 3](#_Toc333330446)

[1.1 Overview 3](#_Toc333330447)

[1.2 Scope 3](#_Toc333330448)

[1.3 Definitions 3](#_Toc333330449)

[1.4 Abbreviations 3](#_Toc333330450)

[1.5 References 3](#_Toc333330451)

[1.6 Conventions 3](#_Toc333330452)

[2 Body of Document – Heading 1 3](#_Toc333330453)

[2.1 Heading 2 3](#_Toc333330454)

[2.1.1 Heading 3 3](#_Toc333330455)

[3 ASN.1 Sample 3](#_Toc333330456)

[4 Landscape Section 3](#_Toc333330457)

[5 XML Sample 3](#_Toc333330458)

[Legal Agreement Styles Section 3](#_Toc333330459)

[1 Legal clause level 1 style 3](#_Toc333330460)

[Annex A Annex Style 3](#_Toc333330461)

[A.1 Annex-heading 1 Style 3](#_Toc333330462)

[A.1.1 Annex-heading 2 Style 3](#_Toc333330463)

[Annex B Document Management 3](#_Toc333330464)

[B.1 Document History 3](#_Toc333330465)

[B.2 Other Information 3](#_Toc333330466)

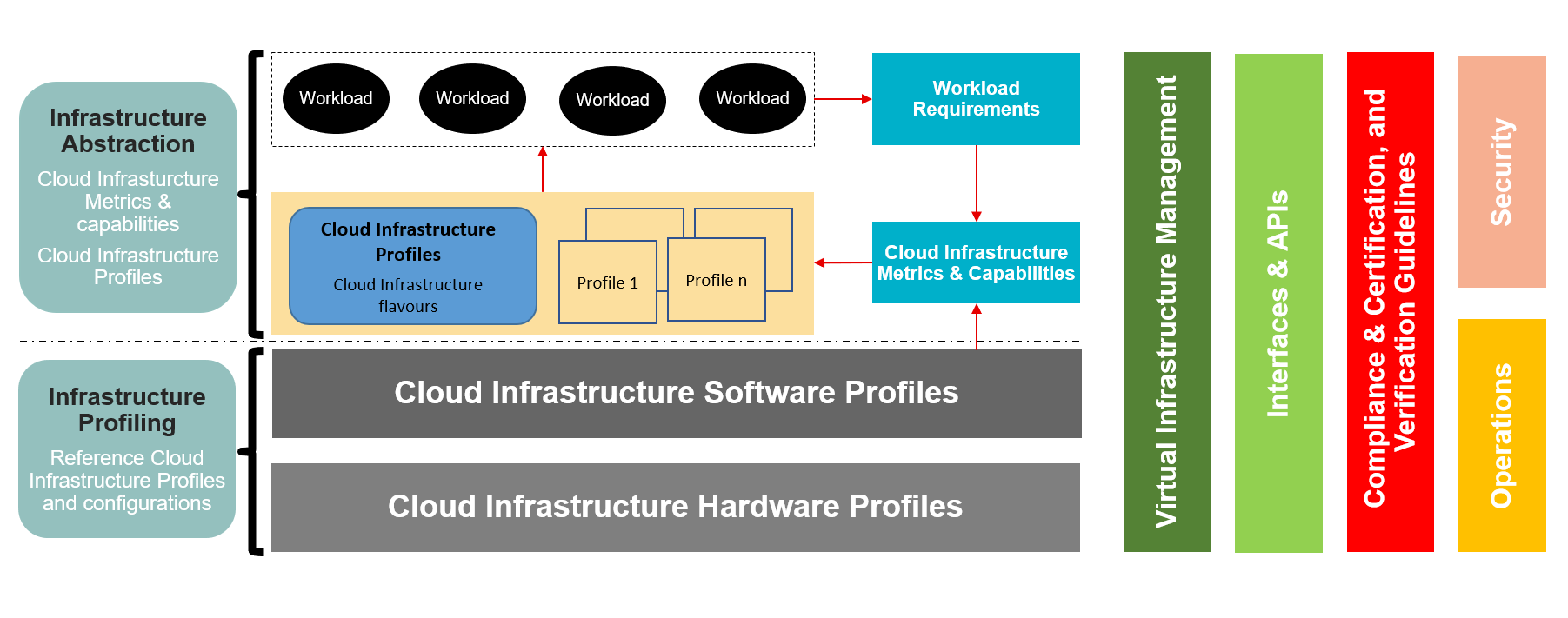
# Introduction

## Overview

The Reference Model (RM) specifies the infrastructure abstraction and the exposure of a set of capabilities, resources, and interfaces to workloads. The aim of the Reference Model is to be virtualisation technology agnostic (VM-based and container-based) and act as a "catalogue" of the exposed infrastructure capabilities, resources, and interfaces needed to develop the workloads.

## Scope

This document focuses on the documenting the higher level concepts that are needed to identify **Reference Model**. Figure 1 below highlights its scope in more details.



1. : Scope of Reference Model

This document specifies:

* **Cloud Infrastructure abstraction**: in context with how it interacts with the other components required to build a complete system that supports workloads deployed in Virtual Machines (VNF) or containers (CNF).
  + **Cloud Infrastructure metrics & capabilities**: A set of metrics and capabilities for the cloud infrastructure which workloads require to perform telco scale network functions.
  + **Infrastructure profiles catalogue**: A catalogue of standard profiles needed in order to abstract the infrastructure from workloads. With a limited and well-defined set of profiles with well understood characteristics, workload compatibility and performance predictability can be achieved.
* **Cloud Infrastructure Software and Hardware profiling**
  + **Cloud Infrastructure software profiles and configurations**: These are software profiles and configurations that map directly to the infrastructure profiles within the infrastructure profiles catalogue.
  + **Cloud Infrastructure hardware profiles and configurations**: These are hardware profiles and configurations which are suitable for the defined cloud infrastructure software profiles & configurations.
* **Conformance and verification**
  + **Conformance programs**: This defines the requirements for certification and validation programs for both workloads and cloud infrastructure.
  + **Test framework:** Provide test suites to allow conformance, certification, and verification of workloads and cloud infrastructure against the defined set of profiles.

## Audience

The document starts from the abstract and as it progresses it increasingly gets into more details. It follows the traditional design process where you start from core principles, progress to abstract concepts and models, then finish with operational considerations, such as security and lifecycle management.

* **Chapter 01 - Introduction**: Overall scope of the Reference Model document including the goals and objectives of the project.  
  **Audience**: This chapter is written for a general technical audience with interest in this topic.
* **Chapter 02 - Workload requirements & Analysis**: High level requirements and core principles needed to understand how the model was developed. Addresses the thinking behind the decisions that were made.  
  **Audience**: This chapter is written for architects and others with an interest in how the decisions were made.
* **Chapter 03 - Modelling**: The high-level cloud infrastructure model itself.  
  **Audience**: This chapter is written for architects and others who wants to gain a quick high-level understanding of the model.
* **Chapter 04 - Infrastructure Capabilities, Metrics, and Catalogue**: Details about the capabilities needed to support the various types of workloads and how the capabilities are applied to the model. The details regarding T-shirt sizes and other considerations are found in this section.  
  **Audience**: This chapter is written for architects, developers and others who need to deploy infrastructure or develop applications.
* **Chapter 05 - Featureset and Requirements from Infrastructure**: This chapter goes into more details on what needs to be part of the cloud infrastructure. It describes the software and hardware capabilities and configurations recommended for the different types of cloud infrastructure profiles.  
  **Audience**: This chapter is written for architects, developers and others who need to deploy infrastructure or develop applications.
* **Chapter 06 - External Interfaces**: This chapter covers APIs and any actual interfaces needed to communicate with the workloads and any other external components.  
  **Audience**: This chapter is written for architects, developers and others who need to develop APIs or develop applications that use the APIs.
* **Chapter 07 - Security Guidelines**: This chapter identifies the security components that need to be taken into consideration when designing and implementing a cloud infrastructure environment. It does not cover details related to company specific requirements to meet regulatory requirements.  
  **Audience**: This chapter is written for security professional, architects, developers and others who need to understand the role of security in the cloud infrastructure environment.
* **Chapter 08 - Conformance, Verification, and Certification**: This chapter details the requirements for developing test suites for a verification and validation program for developers and vendors to validate that their software and applications meet the requirements for the cloud infrastructure architectures.  
  **Audience**: This chapter is written for QA testers, developers and others who need to deploy infrastructure or develop applications.
* **Chapter 09 - Life Cycle Management**: This chapter focuses on the operational aspects of the cloud infrastructure. Discussions include deployment considerations, on-going management, upgrades and other lifecycle concerns and requirements. It does not cover details related to company specific operational requirements, nor does it go into how the cloud infrastructure will interface with existing BSS/OSS systems.  
  **Audience**: This chapter is written for lifecycle managers, operational support teams and others who need to support the infrastructure or the applications.
* **Chapter 10 - Challenges and Gaps**: Opportunities for future developments as technology changes over time.  
  **Audience**: This chapter is written for a general technical audience with interest in this topic.
* **Appendix A - Guidelines For Application Vendors**: More details related to how the applications will interface with the cloud infrastructure through APIs (including Cloud Infrastructure Manager and CaaS).  
  **Audience**: This chapter is written for architects, developers and others who need to deploy infrastructure or develop applications.

## Principles

The Reference Model specification conform with the principles defined in [here](file:///C:\\Users\\tpelt\\Documents\\Infrastructure\\cNTT\\RM\\tech" \l "2.0).

## Definitions/Terminology

To help guide the reader, a glossary [Reference Model Terminology](file:///C:\Users\tpelt\Documents\Infrastructure\cNTT\RM\tech\glossary.md) provides an introduction to the main terms used within this document and throughout the project in general. These definitions are, with a few exceptions, based on the [ETSI GR NFV 003 V1.5.1 (2020-01)](https://www.etsi.org/deliver/etsi_gr/NFV/001_099/003/01.05.01_60/gr_NFV003v010501p.pdf) definitions. In a few cases, they have been modified to avoid deployment technology dependencies only when it seems necessary to avoid confusion.

## Abbreviations

| Term | Description |
| --- | --- |
|  | <This is an optional section giving a list of the abbreviations necessary for the understanding of the GSMA document. This list shall contain all abbreviations and their corresponding full terms which are used within the document, in alphabetical order. If there are no entries needed in an abbreviations table, this section may be deleted> |
| <e.g. PRD> | Permanent Reference Document |

## References

| Ref | Doc Number | Title |
| --- | --- | --- |
|  | <e.g. PRD AA.34> | <PRD or document title e.g. “Policy and Procedures for Official Documents”. For non-binding documents with no reference entries, this section may be deleted > |
|  | RFC 2119 | “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997. Available at <http://www.ietf.org/rfc/rfc2119.txt> |

## Conventions

“The key words “must”, “must not”, “required”, “shall”, “shall not”, “should”, “should not”, “recommended”, “may”, and “optional” in this document are to be interpreted as described in RFC2119 [2].”

# Workload Requirements & Analysis

The Cloud Infrastructure is the totality of all hardware and software components which build up the environment in which VNFs/CNFs (workloads) are deployed, managed and executed. It is, therefore, inevitable that different workloads would require different capabilities and have different expectations from it.

One of the main targets of the CNTT is to define an agnostic cloud infrastructure, to remove any dependencies between workloads and the deployed cloud infrastructure, and offer infrastructure resources to workloads in an abstracted way with defined capabilities and metrics.

This means, operators will be able to host their Telco workloads (VNFs/CNFs) with different traffic types, behaviour and from any vendor on a unified consistent cloud infrastructure.

Additionally, a well defined cloud infrastructure is also needed for other type of workloads such as IT, Machine learning, Artificial Intelligence, etc.

In this chapter we try to analyse various workload types used in telco and examine their requirements. We will also highlight some of the cloud infrastructure parameters needed to achieve the desired performance expected by various workloads.

## Workloads Collateral

There are different ways that workloads can be classified, for example:

* **By function type:**
  + Data Plane (a.k.a., User Plane, Media Plane, Forwarding Plane)
  + Control Plane (a.k.a, Signalling Plane)
  + Management Plane

**Note**: Data plane workloads also include control and management plane functions ; control plane workloads also include management plane functions.

* **By service offered:**
  + Mobile broadband service
  + Fixed broadband Service
  + Voice Service
  + Value-Added-Services
* **By technology:** 2G, 3G, 4G, 5G, IMS, FTTx, Wi-Fi...

Below is a list of Network Functions that covers almost 95% of the Telco workload (and the most likely to be virtualised). They are gathered by network segment and function type.

* **Radio Access Network (RAN)**
  + Data Plane
    - BBU: BaseBand Unit
    - CU: Centralised Unit
    - DU: Distributed Unit
* **2G/3G/4G mobile core network**
  + Control Plane
    - MME: Mobility Management Entity
    - 3GPP AAA: Authentication, Authorisation, and Accounting
    - PCRF: Policy and Charging Rules Function
    - OCS: Online Charging system
    - OFCS: Offline Charging System
    - HSS: Home Subscriber Server
    - DRA: Diameter Routing Agent
    - HLR: Home Location Register
    - SGW-C: Serving GateWay Control plane
    - PGW-C: Packet data network GateWay Control plane
  + Data Plane
    - SGW: Serving GateWay
    - SGW-U: Serving GateWay User plane
    - PGW: Packet data network GateWay
    - PGW-U: Packet data network GateWay User plane
    - ePDG: Evolved Packet Data GateWay
    - MSC: Mobile Switching Center
    - SGSN: Serving GPRS Support Node
    - GGSN: Gateway GPRS Support Node
    - SMSC : SMS Center
* **5G core network** 5G core nodes are virtualisable by design and strong candidate to be onboarded onto Telco Cloud as "cloud native application"
  + Data Plane
    - UPF: User Plane Function
  + Control Plane
    - AMF: Access and Mobility management Function
    - SMF: Session Management Function
    - PCF: Policy Control Function
    - AUSF: Authentication Server Function
    - NSSF: Network Slice Selection Function
    - UDM: Unified Data Management
    - UDR: Unified Data Repository
    - NRF: Network Repository Function
    - NEF: Network Exposure Function

**Note:** for Service-based Architecture (SBA) all Network Functions are stateless (store all sessions/ state on unified data repository UDR)

* **IP Multimedia Subsystem (IMS)**
  + Data Plane
    - MGW: Media GateWay
    - SBC: Session Border Controller
    - MRF: Media Resource Function
  + Control Plane
    - CSCF: Call Session Control Function
    - MTAS: Mobile Telephony Application Server
    - BGCF: Border Gateway Control Function
    - MGCF: Media Gateway Control Function
* **Fixed network**
  + Data Plane
    - MSAN: MultiService Access Node
    - OLT: Optical Line Termination
    - WLC: WLAN Controller
    - BNG: Border Network Gateway
    - BRAS: Broadband Remote Access Server
    - RGW: Residential GateWay
    - CPE: Customer Premises Equipment
  + Control Plane
    - AAA: Authentication, Authorisation, and Accounting
* **Other network functions**
  + Data Plane
    - LSR: Label Switching Router
    - DPI: Deep Packet Inspection
    - CG-NAT: Carrier-Grade Network Address Translation
    - ADC: Application Delivery Controller
    - FW: FireWall
    - Sec-GW: Security GateWay
    - CDN: Content Delivery Network
  + Control plane
    - RR: Route Reflector
    - DNS: Domain Name System
  + Management Plane
    - NMS: Network Management System

## Analysis

Studying various requirements of workloads helps understanding what expectation they will have from the underlying cloud infrastructure. Following are *some* of the requirement types on which various workloads might have different expectation levels:

* Computing
  + Speed (e.g., CPU clock and physical cores number)
  + Predictability (e.g., CPU and RAM sharing level)
  + Specific processing (e.g., cryptography, transcoding)
* Networking
  + Throughput (i.e., bit rate and/or packet rate)
  + Latency
  + Connection points / interfaces number (i.e., vNIC and VLAN)
  + Specific traffic control (e.g., firewalling, NAT, cyphering)
  + Specific external network connectivity (e.g., MPLS, VXLAN)
* Storage
  + IOPS (i.e., input/output rate and/or byte rate)
  + Volume
  + Ephemeral or Persistent
  + Specific features (e.g., object storage, local storage)

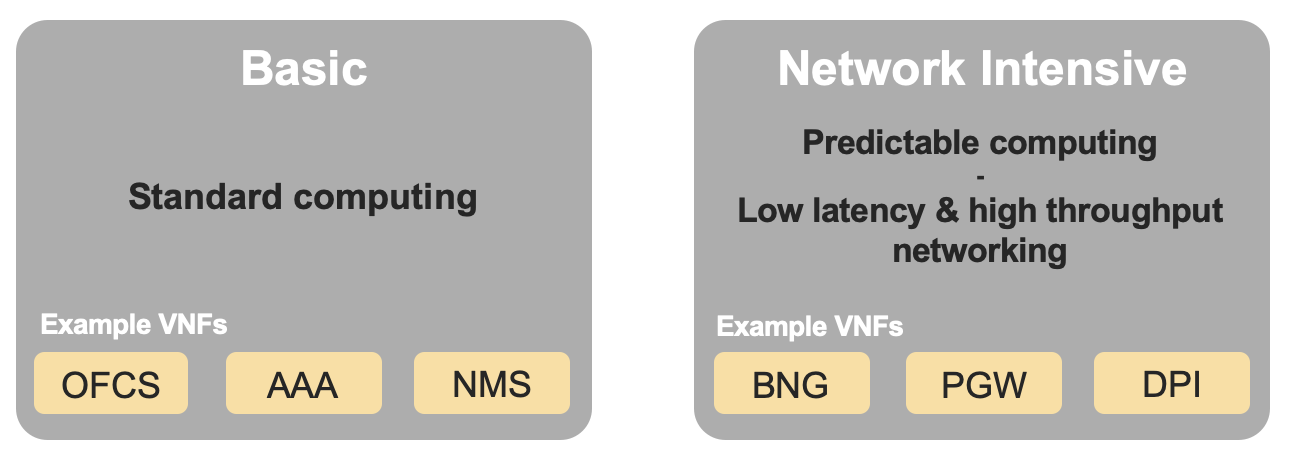
By trying to sort workloads into different categories based on the requirements observed, below are the different profiles concluded, which are mainly driven by expected performance levels:

* **Profile One**
  + Workload types
    - Control plane functions without specific need, and management plane functions
    - *Examples: OFCS, AAA, NMS*
  + No specific requirement
* **Profile Two**
  + Workload types
    - Data plane functions (i.e., functions with specific networking and computing needs)
    - *Examples: BNG, S/PGW, UPF, Sec-GW, DPI, CDN, SBC, MME, AMF, IMS-CSCF, UDR*
  + Requirements
    - Predictable computing
    - High network throughput
    - Low network latency

## Cloud Infrastructure Profiles

Based on the above analysis, following cloud infrastructure profiles are proposed (also shown in Figure 2 below)

* **Basic**: for Workloads that can tolerate resource over-subscription and variable latency.
* **Network Intensive**: for Workloads that require predictable computing performance, high network throughput and low network latency.



1. : Infrastructure profiles proposed based on VNFs categorisation.

On **Chapter 4** later in the document, these infrastructure profiles will be offered to workloads: **B (Basic)** and **N (Network intensive)** respectively.

**Note:** This is an initial set of proposed profiles and it is expected that more profiles will be added as more requirements are gathered and as technology enhances and matures. For instance, the following profiles may be added in future releases:

* **Compute Intensive**: for Workloads that require predictable computing performance and low network latency.
* ***Storage Intensive****: for Workloads that require low storage latency and/or high storage IOPS.*
* ***Enhanced Compute Intensive****: for compute intensive Workloads that require higher computing performance and/or specific compute resource (e.g., GPU).*
* ***Enhanced Network Intensive****: for network intensive Workloads that require higher network performance and/or specific network resource (e.g., crypto acceleration).*

# Modelling

There is the necessity to clearly define which kind of infrastructure resources a shared network function virtualisation infrastructure (NFVI) will provide for hosting workloads including virtual network functions (VNFs) and/or cloud-native network functions (CNF), so that the requirements of the workloads match the capabilities of the NFVI.

The lack of a common understanding of which resources and corresponding capabilities a suitable NFVI should provide may lead to several issues which could negatively impact the time and the cost for on-boarding and maintaining these solutions on top of a virtualised infrastructure. For Example:

* Supporting any kind of workload specific requirements (e.g. regarding network acceleration or API access) might result in having to establish different silo of NFVIs for each workload type.
* Synchronising the release cycles of a large set of different technologies will sooner or later lead to situations in which required upgrades cannot be applied easily due to incompatibilities.

The abstraction model presented in this chapter specifies a common set of virtual infrastructure resources which NFVI will need to provide to be able to host most of the typical VNF/CNF workloads required by the operator community.

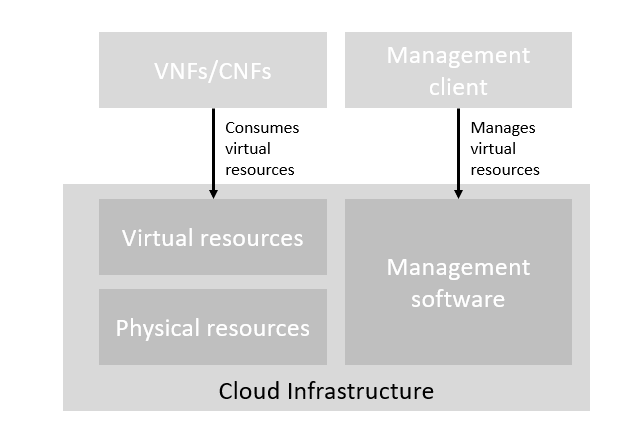
Although a couple of explicit and implicit abstraction models (e.g. in the context of ETSI NFV) are already available, they fall short when addressing the following design principles:

* **Scope:** the model should describe the most relevant virtualised infrastructure resources (incl. acceleration technologies) an NFVI needs to provide for hosting Telco workloads
* **Separation of Concern:** the model should support a clear distinction between the responsibilities related to maintaining the network function virtualisation infrastructure and the responsibilities related to managing the various VNF workloads
* **Simplicity:** the amount of different types of resources (including their attributes and relationships amongst one another) should be kept to a minimum to reduce the configuration spectrum which needs to be considered
* **Declarative**: the model should allow for the description of the intended state and configuration of the NFVI resources for automated life cycle management
* **Explicit:** the model needs to be rich enough to allow for a direct mapping towards the APIs of NFVIs for the instantiation of virtual infrastructure elements without requiring any additional parameters
* **Lifecycle:** the model must distinguish between resources which have independent lifecycles but should group together those resources which share a common lifecycle
* **Aligned:** the model should clearly highlight the dependencies between the elements to allow for a well-defined and simplified synchronisation of independent automation tasks.

**To summarise:** the abstraction model presented in this document will build upon existing modelling concepts and simplify and streamline them to the needs of telco operators who intend to distinguish between infrastructure related and workload related responsibilities.

## Model

The abstraction model for the NFVI makes use of the following layers (only the virtual infrastructure layer will be directly exposed to workloads (VNFs/CNFs)):



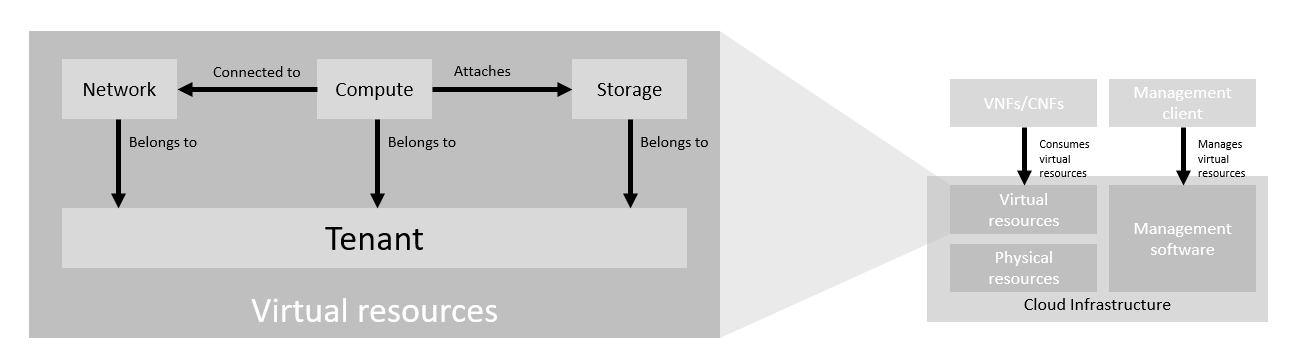
1. : NFVI Model Overview.

The functionalities of each layer are as follows:

* **Virtual Infrastructure Resources:** These are all the infrastructure resources (compute, storage and networks) which the NFVI provides to the workloads such as VNFs/CNFs. These virtual resources can be managed by the tenants and tenant workloads directly or indirectly via an application programming interface (API).
* **NFVI Management Software:** This consists of the software components that manage the physical and virtual resources and make those management capabilities accessible via one or more APIs. The host Operating System (OS) is responsible for managing the physical infrastructure resources and abstracting them from processes running within the OS. Virtualisation / containerisation technology dynamically allocates these abstracted hardware components and exposes them as virtual resources. Additional software is responsible for the management of logical constructs such as tenants, tenant workloads, resource catalogues, identities, access controls, security policies, etc.
* **Physical Infrastructure Resources:** These consist of physical hardware components such as servers, (including random access memory, local storage, network ports, and hardware acceleration devices), storage devices, network devices, and the basic input output system (BIOS).
* **Workloads (VNFs/CNFs):** These consist of workloads such as virtualized and/or containerized network functions that run on top of a VM or as a Container.

## Virtual Resources

The virtual infrastructure resources provided by the Cloud Infrastructure can be grouped into four categories as shown in the diagram below:



1. : Virtual Infrastructure Resources provides virtual compute, storage and networks in a tenant context.

* **Tenants:** represent an isolated and independently manageable elastic pool of compute, storage and network resources
* **Compute resources:** represent virtualised computes for workloads and other systems as necessary
* **Storage resources:** represent virtualised resources for persisting data
* **Network resources:** represent virtual resources providing layer 2 and layer 3 connectivity

The virtualised infrastructure resources related to these categories are listed below.

### Tenant

A network function virtualisation infrastructure (NFVI) needs to be capable of supporting multiple tenants and has to isolate sets of infrastructure resources dedicated to specific workloads (VNF/CNF) from one another. Tenants represent an independently manageable logical pool of compute, storage and network resources abstracted from physical hardware.

***Example****: a tenant within an OpenStack environment or a Kubernetes cluster.*

|  |  |
| --- | --- |
| Attribute | Description |
| name | name of the logical resource pool |
| type | type of tenant (e.g. OpenStack tenant, Kubernetes cluster, …) |
| vcpus | max. number of virtual CPUs |
| ram | max. size of random access memory in GB |
| disk | max. size of ephemeral disk in GB |
| networks | description of external networks required for inter-domain connectivity |
| metadata | key/value pairs for selection of the appropriate physical context (e.g. location, availability zone, …) |

1. : Attributes of a tenant

### Compute

A virtual machine or a container/pod belonging to a tenant capable of hosting the application components of workloads (VNFs). A virtual compute therefore requires a tenant context and since it will need to communicate with other communication partners it is assumed that the networks have been provisioned in advance.

***Example****: a virtual compute descriptor as defined in TOSCA Simple Profile for NFV.*

|  |  |
| --- | --- |
| Attribute | Description |
| name | name of the virtual host |
| vcpus | number of virtual cpus |
| ram | size of random access memory in GB |
| disc | size of root disc in GB |
| nics | sorted list of network interfaces connecting the host to the virtual networks |
| acceleration | key/value pairs for selection of the appropriate acceleration technology |
| metadata | key/value pairs for selection of the appropriate redundancy domain |

1. : Attributes of compute resources

### Storage

A block device of a certain size for persisting information which can be created and dynamically attached to/detached from a virtual compute. A storage device resides in a tenant context and exists independently from any compute host.

***Example****: an OpenStack cinder volume.*

|  |  |
| --- | --- |
| Attribute | Description |
| name | name of storage resources |
| size | size of disc in GB |
| attachments | list of compute hosts to which the device is currently attached |
| acceleration | key/value pairs for selection of the appropriate acceleration technology |
| metadata | key/value pairs for selection of the appropriate redundancy domain |

1. : Attributes of storage resources

***Comments****: we need to be more specific regarding acceleration and metadata.*

### Availability Zone

An Availability Zone is a logical pool of physical resources (e.g. compute, block storage, network). These logical pools segment the physical resources of a cloud based on factors chosen by the cloud operator. The cloud operator may create availability zones based on location (rack, datacenter), or indirect failure domain dependencies like power sources. Workloads can leverage availability zones to utilise multiple locations or avoid sharing failure domains for a workload, and thus increase its fault-tolerance.

As a logical group with operator-specified criteria, the only mandatory attribute for an Availability Zone is the name.

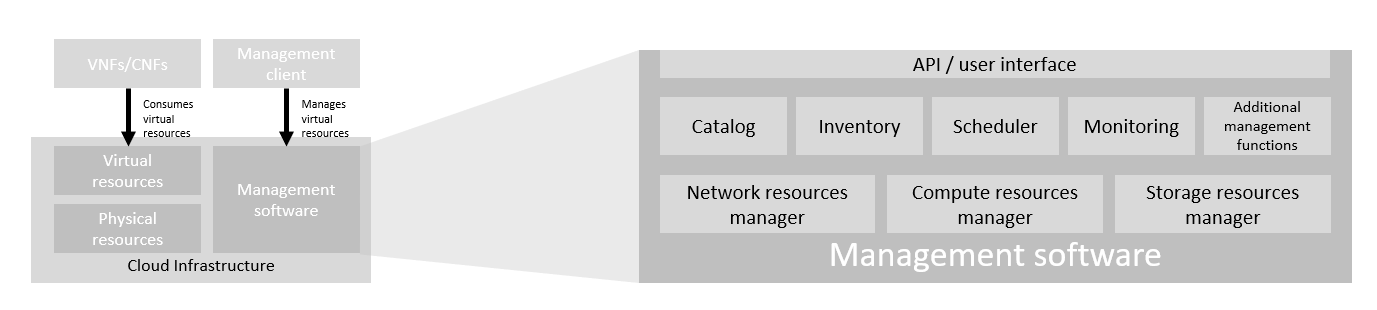
|  |  |
| --- | --- |
| Attribute | Description |
| name | name of the availability zone |

1. : Attributes of availability zones

## NFVI Management Software

Network Function Virtualisation Infrastructure provides the capability to manage physical and virtual resources via Application Programmable Interfaces or graphical user interfaces. The management software allows to:

* setup, manage and delete tenants,
* setup, manage and delete user- and service-accounts,
* manage access privileges and
* provision, manage, monitor and delete virtual resources.



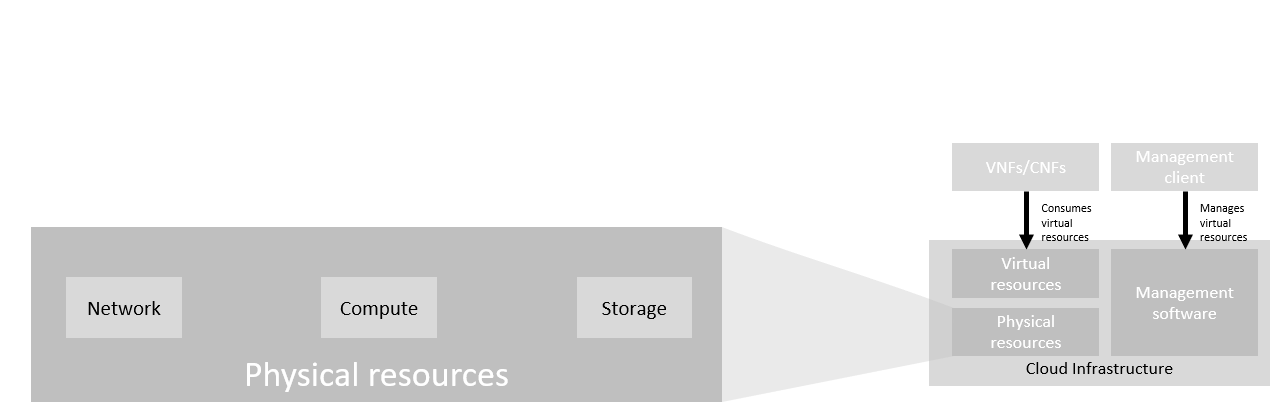
1. : NFVI Management Software.

The management software needs to support following functional aspects:

* **API/UI** : an application programming interface / user interface providing access to the NFVI management functions
* **Catalogue** : manages the collection of available templates for virtual resource the NFVI can provide
* **Inventory** : manages the information related to all the physical and virtual resources of a NFVI
* **Scheduler** : receives requests via API/UI, provisions and manages virtual resources by coordinating the activities of the compute-, storage- and network resources managers
* **Monitoring** : monitors and collects information on all events and the current state of all physical and virtual resources
* **Additional Management Functions** : include identity management, access management, policy management (e.g. to enforce security policies), etc.
* **Compute Resources Manager** : provides a mechanism to provision virtual resources with the help of physical compute resources
* **Storage Resources Manager** : provides a mechanism to provision virtual resources with the help of physical storage resources
* **Network Resources Manager** : provides a mechanism to provision virtual resources with the help of physical network resources

## Physical Resources

The physical compute, storage and network resources serve as the foundation of the network function virtualisation infrastructure. They are as such not directly exposed to the workloads (VNFs/CNFs).



1. : NFVI Physical Resources

## Network

Networking, alongside Compute and Storage, is an integral part of the Cloud Infrastructure (Network Function Virtualisation Infrastructure). The general function of networking in this context is to provide the connectivity between various virtual and physical resources required for the delivery of a network service. Such connectivity may manifest itself as a virtualised network between VMs and/or containers (e.g. overlay networks managed by SDN controllers, and/or programmable network fabrics) or as an integration into the infrastructure hardware level for offloading some of the network service functionality.

Normalization of the integration reference points between different layers of the Cloud Infrastructure architecture is one of the main concerns. In the networking context the primary focus is directed on the packet flow and control flow interfaces between the virtual resources (referred to as Software (SW) Virtualisation Layer) and physical resources (referred to as Hardware (HW) Infrastructure Layer), as well as on related integration into the various MANO reference points (hardware/network infrastructure management, orchestration). The identification of these two different layers (SW Virtualisation Layer and HW Infrastructure Layer) remains in alignment with the separation of resources into virtual and physical resources, generally used in this document, see e.g. Figure 3. The importance of understanding the seperation of concerns between SW Virtualisation Layer and HW Infrastructure Layer is important because without it, the cardinality of having multiple CaaS and IaaS instances executing on their own private resources from the single shared HW Infrastructure Layer cannot be expressed into separate administrative domains.

# Infrastructure Capabilities, Measurements and Catalogue

## Capabilities and Performance Measurements

This section describes and uniquely identifies the Capabilities provided directly by the Infrastructure, as well as Performance Measurements (PMs) generated directly by the Infrastructure (i.e. without the use of external instrumentation).

The Capability and PM identifiers conform to the following schema:

**a.b.c** (Ex. "e.pm.001")  
a = Scope <(e)xternal | (i)nternal | (t)hird\_party\_instrumentation>  
b = Type <(cap)ability | (pm)>  
c = Serial Number

A spreadsheet in the artefact repository maintains the list of assigned identifiers, along with their respective descriptions and the next available identifier, globally across all chapters of the RM.

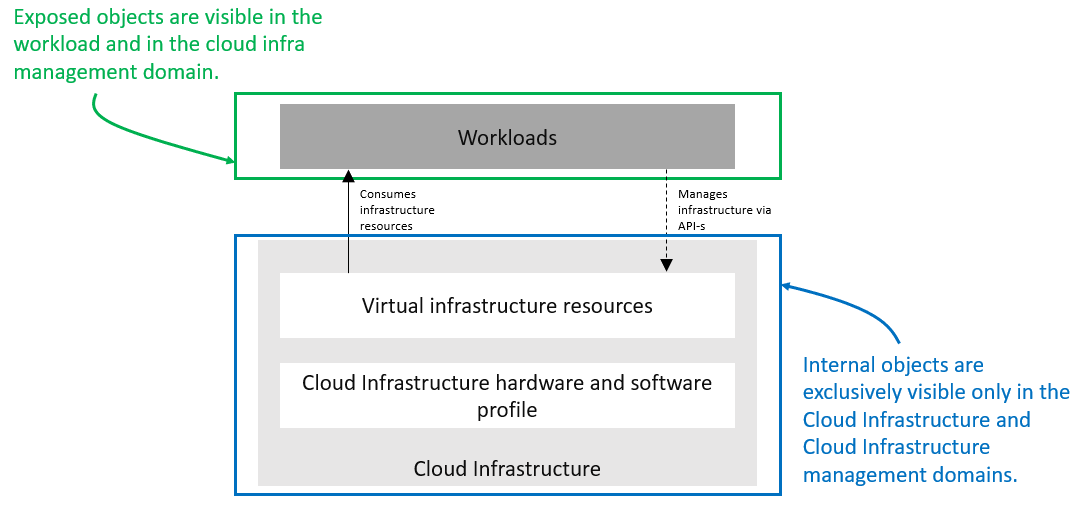
**To-Do**: Create tracking ss and add link here

### Exposed vs Internal

The following pertains to the context of Cloud Infrastructure Resources, Capabilities and Performance Measurements (PMs) as discussed within this chapter.

**Exposed**: Refers to any object (e.g., resource discovery/configuration/consumption, platform telemetry, Interface, etc.) that exists in or pertains to, the domain of the Cloud Infrastructure and is made visible (aka “Exposed”) to a workload. When an object is exposed to a given workload, the scope of visibility within a given workload is at the discretion of the specific workload’s designer. From an Infra perspective, the Infra-resident object is simply being exposed to one or more virtual environments (i.e. Workloads). It is then the responsibility of the kernel or supervisor/executive within the VM to control how, when and where the object is further exposed within the VM, with regard to permissions, security, etc. As the object(s) originate with the Infra, they are by definition visible within that domain.

**Internal**: Effectively the opposite of Exposed; objects Internal to the Cloud Infrastructure, which are exclusively available for use by the Cloud Infrastructure and components within the Cloud Infrastructure.



1. : Exposed vs. Internal Scope

As illustrated in the figure above, objects designated as "Internal" are only visible within the area inside the blue oval (the Cloud Infrastructure), and only when the entity accessing the object has the appropriate permissions. Whereas objects designated as "Exposed" are potentially visible from both the area within the green oval (the Workload), as well as from within the Cloud Infrastructure, again provided the entity accessing the object has appropriate permissions.

**Note**: The figure above indicates the areas from where the objects are visible. It is not intended to indicate where the objects are instantiated. For example, the virtual resources are instantiated within the Cloud Infrastructure (the blue area), but are Exposed, and therefore are visible to the Workload, within the green area.

### Exposed Infrastructure Capabilities

This section describes a set of explicit Cloud Infrastructure capabilities and performance measurements that define a Cloud Infrastructure. These capabilities and PMs are well known to workloads as they provide capabilities which workloads rely on.

**Note**: It is expected that Cloud Infrastructure capabilities and measurements will expand over time as more capabilities are added and technology enhances and matures.

#### Exposed Resource Capabilities

Table 5 below shows resource capabilities of Cloud Infrastructure. Those indicate resources offered to workloads by Cloud Infrastructure.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure Capability | Unit | Definition/Notes |
| e.cap.001 | # vCPU | number | Max number of vCPU that can be assigned to a single VM or Pod 1) |
| e.cap.002 | RAM Size | MB | Max memory in MB that can be assigned to a single VM or Pod by the Cloud Infrastructure 2) |
| e.cap.003 | Total per-instance (ephemeral) storage | GB | Max storage in GB that can be assigned to a single VM or Pod by the Cloud Infrastructure |
| e.cap.004 | # Connection points | number | Max number of connection points that can be assigned to a single VM or Pod by the Cloud Infrastructure |
| e.cap.005 | Total external (persistent) storage | GB | Max storage in GB that can be attached / mounted to VM or Pod by the Cloud Infrastructure |

1. : Exposed Resource Capabilities of Cloud Infrastructure

**Notes: 1)** In a Kubernetes based environment this means the CPU limit of a pod.

**2)** In a Kubernetes based environment this means the memory limit of a pod.

#### Exposed Performance Optimisation Capabilities

Table 6 shows possible performance optimisation capabilities that can be provided by Cloud Infrastructure. These indicate capabilities exposed to workloads. These capabilities are to be consumed by workloads in a standard way.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure Capability | Unit | Definition/Notes |
| e.cap.006 | CPU core pinning support | Yes/No | Indicates if Cloud Infrastructure supports CPU core pinning |
| e.cap.007 | NUMA support | Yes/No | Indicates if Cloud Infrastructure supports NUMA |
| e.cap.008 | IPSec Acceleration | Yes/No | IPSec Acceleration |
| e.cap.009 | Crypto Acceleration | Yes/No | Crypto Acceleration |
| e.cap.010 | Transcoding Acceleration | Yes/No | Transcoding Acceleration |
| e.cap.011 | Programmable Acceleration | Yes/No | Programmable Acceleration |
| e.cap.012 | Enhanced Cache Management 1) | Yes/No | If supported, L=Lean; E=Equal; X=eXpanded |
| e.cap.013 | SR-IOV over PCI-PT 2) | Yes/No | Traditional SR-IOV |
| e.cap.014 | GPU/NPU 2) | Yes/No | Hardware coprocessor |
| e.cap.015 | SmartNIC 2)3) | Yes/No | Network Acceleration |
| e.cap.016 | FPGA/other Acceleration H/W 2) | Yes/No | Non-specific hardware |

1. : Exposed Performance Optimisation Capabilities of Cloud Infrastructure

**Notes**: 1) L and X cache policies require CPU pinning to be active.

2) These Capabilities generally require hardware-dependent drivers be injected into workloads, which is prohibited by CNTT principles. As such, use of these features shall be governed by the applicable CNTT policy. Please consult the RM Appendix for the usage policy relevant to any needed hardware Capability of this type.

3) SmartNICs that do not utilise PCI-PT are not subject to the Abstraction Principle, nor any related policies or prohibitions.

Enhanced Cache Management is a compute performance enhancer that applies a cache management policy to the socket hosting a given virtual compute instance, provided the associated physical CPU microarchitecture supports it. Cache management policy can be used to specify the static allocation of cache resources to cores within a socket. The "Equal" policy distributes the available cache resources equally across all of the physical cores in the socket. The "eXpanded" policy provides additional resources to the core pinned to a workload that has the "X" attribute applied. The "Lean" attribute can be applied to workloads which do not realize significant benefit from a marginal cache size increase and are hence willing to relinquish unneeded resources.

In addition to static allocation, an advanced Reference Architecture implementation can implement dynamic cache management control policies, operating with tight (~ms) or standard (10s of seconds) control loop response times, thereby achieving higher overall performance for the socket.

#### Exposed Monitoring Capabilities

Monitoring capabilities are used for the passive observation of workload-specific traffic traversing the Cloud Infrastructure. As with all capabilities, Monitoring may be unavailable or intentionally disabled for security reasons in a given Cloud Infrastructure deployment. If this functionality is enabled, it must be subject to strict security policies. Refer to the Reference Model Security chapter for additional details.

Table 7 shows possible monitoring capabilities available from the Cloud Infrastructure for workloads.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure Capability | Unit | Definition/Notes |
| e.cap.017 | Monitoring of L2-7 data | Yes/No | Ability to monitor L2-L7 data from workload |

1. : Exposed Monitoring Capabilities of Cloud Infrastructure

### Exposed Infrastructure Performance Measurements

The intent of the following PMs is to be available for and well known to wokloads.

#### Exposed Performance Measurements

The following table of exposed Performance Measurements shows PMs per VM or Pod, vNIC or vCPU. Network test setups are aligned with ETSI GS NFV-TST 009 [2]. Specifically exposed PMs use a single workload (PVP) dataplane test setup in a single host.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure Measurement | Unit | Definition/Notes |
| e.pm.xxx | Place Holder | Units | Concise description |

1. : Exposed Performance Measurements of Cloud Infrastructure

### Internal Infrastructure Capabilities

This section covers a list of implicit Cloud Infrastructure capabilities and measurements that define an Cloud Infrastructure. These capabilities and metrics determine how the Cloud Infrastructure behaves internally. They are hidden from workloads (i.e. workloads may not know about them) but they will impact the overall performance and capabilities of a given Cloud Infrastructure solution.

**Note**: It is expected that implicit Cloud Infrastructure capabilities and metrics will evolve with time as more capabilities are added as technology enhances and matures.

#### Internal Resource Capabilities

Table 9 shows resource capabilities of Cloud Infrastructure. These include capabilities offered to workloads and resources consumed internally by Cloud Infrastructure.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure Capability | Unit | Definition/Notes |
| i.cap.014 | CPU cores consumed by the Cloud Infrastructure overhead in a compute node | % (of total available) | Indicates the percentage of cores consumed by the Cloud Infrastructure components (including host OS) in a compute node |
| i.cap.015 | Memory consumed by the Cloud Infrastructure overhead in a compute node | % (of total available) | Indicates the percentage of memory consumed by the Cloud Infrastructure components (including host OS) in a compute node |

1. : Internal Resource Capabilities of Cloud Infrastructure

#### Internal SLA capabilities

Table 10 below shows SLA (Service Level Agreement) capabilities of Cloud Infrastructure. These include Cloud Infrastructure capabilities required by workloads as well as required internal to Cloud Infrastructure. Application of these capabilities to a given workload is determined by its Cloud Infrastructure Profile.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure capability | Unit | Definition/Notes |
| i.cap.016 | CPU allocation ratio | N:1 | Number of virtual cores per physical core; also known as CPU overbooking ratio |
| i.cap.017 | Connection point QoS | Yes/No | QoS enablement of the connection point (vNIC or interface) |

1. : Internal SLA capabilities to Cloud Infrastructure

#### Internal Performance Optimisation Capabilities

Table 11 below shows possible performance optimisation capabilities that can be provided by Cloud Infrastructure. These include capabilities exposed to workloads as well as internal capabilities to Cloud Infrastructure. These capabilities will be determined by the Cloud Infrastructure Profile used by the Cloud Infrastructure.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure capability | Unit | Definition/Notes |
| i.cap.018 | Huge page support | Yes/No | Indicates if the Cloud Infrastructure supports huge pages |

1. : Internal performance optimisation capabilities of Cloud Infrastructure

#### Internal Performance Measurement Capabilities

Table 12 shows possible performance measurement capabilities available by Cloud Infrastructure. The availability of these capabilities will be determined by the Cloud Infrastructure Profile used by the workloads.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure Measurement | Unit | Definition/Notes |
| i.pm.001 | Host CPU usage | nanoseconds | Per Compute node. It maps to [ETSI GS NFV-TST 008 V3.2.1](https://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/008/03.02.01_60/gs_NFV-TST008v030201p.pdf) clause 6, processor usage metric (Cloud Infrastructure internal). |
| i.pm.002 | Virtual compute resource CPU usage | nanoseconds | Per VM or Pod. It maps to [ETSI GS NFV-IFA 027 v2.4.1](https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/027/02.04.01_60/gs_nfv-ifa027v020401p.pdf) Mean Virtual CPU usage and Peak Virtual CPU usage (Cloud Infrastructure external). |
| i.pm.003 | Host CPU utilization | % | Per Compute node. It maps to [ETSI GS NFV-TST 008 V3.2.1](https://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/008/03.02.01_60/gs_NFV-TST008v030201p.pdf) clause 6, processor usage metric (Cloud Infrastructure internal). |
| i.pm.004 | Virtual compute resource CPU utilization | % | Per VM or Pod. It maps to [ETSI GS NFV-IFA 027 v2.4.1](https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/027/02.04.01_60/gs_nfv-ifa027v020401p.pdf) Mean Virtual CPU usage and Peak Virtual CPU usage (Cloud Infrastructure external). |
| i.pm.005 | Measurement of external storage IOPs | Yes/No |  |
| i.pm.006 | Measurement of external storage throughput | Yes/No |  |
| i.pm.007 | Available external storage capacity | Yes/No |  |

1. : Internal Measurement Capabilities of Cloud Infrastructure

### Cloud Infrastructure management Capabilities

Cloud Infrastructure management is responsible for controlling and managing the Cloud Infrastructure compute, storage, and network resources. Resources allocation is dynamically set up upon workloads requirements. This section covers the list of capabilities offered by the VIM to workloads or service orchestrator.

Table 13 shows capabilities related to resources allocation

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure management Capability | Unit | Definition/Notes |
| e.man.001 | Virtual Compute allocation | Yes/No | Capability to allocate virtual compute resources to a workload |
| e.man.002 | Virtual Storage allocation | Yes/No | Capability to allocate virtual storage resources to a workload |
| e.man.003 | Virtual Networking resources allocation | Yes/No | Capability to allocate virtual networking resources to a workload |
| e.man.004 | Multi-tenant isolation | Yes/No | Capability to isolate resources between tenants |
| e.man.005 | Images management | Yes/No | Capability to manage workload software images |
| e.man.010 | Compute Availability Zones | list of strings | The names of each Compute Availability Zone that was defined to separate failure domains |
| e.man.011 | Storage Availability Zones | list of strings | The names of each Storage Availability Zone that was defined to separate failure domains |

1. : Cloud Infrastructure management Resource Allocation Capabilities

Table 14 Shows performance measurement capabilities

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure management Capability | Unit | Definition/Notes |
| e.man.006 | Virtual resources inventory per tenant | Yes/No | Capability to provide information related to allocated virtualised resources per tenant |
| e.man.007 | Resources Monitoring | Yes/No | Capability to notify state changes of allocated resources |
| e.man.008 | Virtual resources Performance | Yes/No | Capability to collect and expose performance information on virtualised resources allocated |
| e.man.009 | Virtual resources Fault information | Yes/No | Capability to collect and notify fault information on virtualised resources |

1. : Cloud Infrastructure management Performance Measurement Capabilities

### Cloud Infrastructure management Performance Measurements

#### Resources Management Measurements

Table 15 shows resource management measurements of VIM as aligned with ETSI GS NFV TST-012 [3].

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Cloud Infrastructure management Measurement | Unit | Definition/Notes |
| e.man-pm.001 | Time to create Virtual Compute resources for a given workload | Max ms |  |
| e.man-pm.002 | Time to delete Virtual Compute resources of a given workload | Max ms |  |
| e.man-pm.003 | Time to start Virtual Compute resources of a given workload | Max ms |  |
| e.man-pm.004 | Time to stop Virtual Compute resources of a given workload | Max ms | 1) |
| e.man-pm.005 | Time to pause Virtual Compute resources of a given workload | Max ms | 2) |
| e.man-pm.006 | Time to create internal virtual network | Max ms |  |
| e.man-pm.007 | Time to delete internal virtual network | Max ms |  |
| e.man-pm.008 | Time to update internal virtual network | Max ms |  |
| e.man-pm.009 | Time to create external virtual network | Max ms |  |
| e.man-pm.010 | Time to delete external virtual network | Max ms |  |
| e.man-pm.011 | Time to update external virtual network | Max ms |  |
| e.man-pm.012 | Time to create external storage ready for use by workload | Max ms |  |

1. : Cloud Infrastructure management Resource Management Measurements

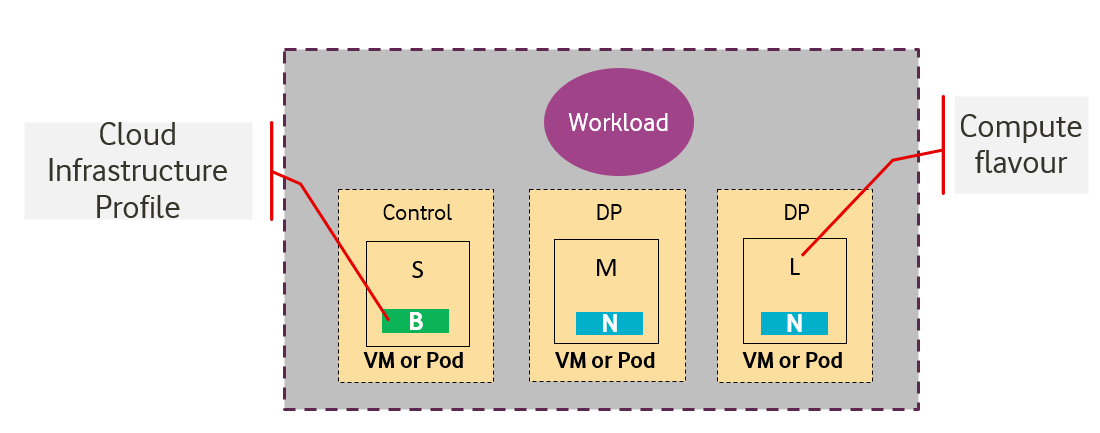
**Notes: 1)** In case of containers there is no stop operation.

**2)** In case of containers there is no pause operation.

## Infrastructure Profiles Catalogue

Infrastructure exposes compute Flavours with options, virtual interface options, storage extensions, and acceleration extensions to workloads. These Cloud Infrastructure Profiles are offered to workloads with their corresponding options and extensions.

The idea of the Cloud Infrastructure profiles is to have a predefined set of infrastructure capabilities with a predefined set of compute Flavours which workload vendors use to build their workloads. Each workload can use several Flavours from different Cloud Infrastructure Profiles to build its overall functionality as illustrated in Figure 8.



1. : Workloads built against Cloud Infrastructure Profiles and compute Flavours.

### Compute Flavours

Compute Flavours represent the compute, memory, storage, and management network resource templates that are used by VMs on the compute hosts. Each VM is given a compute Flavour (resource template), which determines the VMs compute, memory and storage characteristics.

Compute Flavours can also specify secondary ephemeral storage, swap disk, etc. A compute Flavour geometry consists of the following elements:

|  |  |
| --- | --- |
| Element | Description |
| Compute Flavour Name | A descriptive name |
| Virtual compute resources (aka vCPUs) | Number of virtual compute resources (vCPUs) presented to the VM instance. |
| Memory | Virtual compute instance memory in megabytes. |
| Ephemeral/Local Disk | Specifies the size of an ephemeral data disk that exists only for the life of the instance. Default value is 0.The ephemeral disk may be partitioned into boot (base image) and swap space disks. |
| Management Interface | Specifies the bandwidth of management interface/s |

1. : Compute Flavour Geometry Specification.

#### Predefined Compute Flavours

The intent of the following Flavours list is to be comprehensive and yet effective to cover both IT and NFV workloads. The compute Flavours are specified relative to the “large” Flavour. The “large” Flavour configuration consists of 4 vCPUs, 8 GB of RAM and 80 GB of local disk, and the resulting virtual compute instance will have a management interface of 1 Gbps. The “medium” Flavour is half the size of a large and small is half the size of medium. The tiny Flavour is a special sized Flavour.

**Note***:* Customised (Parameterized) Flavours can be used in concession by operators and, if needed, are created using TOSCA, HEAT templates, and/or VIM APIs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .conf | vCPU ("c") 2) | RAM ("r") 2) | Local Disk ("d") | Management Interface |
| .tiny | 1 | 512 MB | 1 GB | 1 Gbps |
| .small | 1 | 2 GB | 20 GB | 1 Gbps |
| .medium | 2 | 4 GB | 40 GB | 1 Gbps |
| .large | 4 | 8 GB | 80 GB | 1 Gbps |
| .2xlarge 1) | 8 | 16 GB | 160 GB | 1 Gbps |
| .4xlarge 1) | 16 | 32 GB | 320 GB | 1 Gbps |
| .8xlarge 1) | 32 | 64 GB | 640 GB | 1 Gbps |

1. : Predefined Compute Flavours.

**Notes**: 1) These compute Flavours are intended to be used for transitional purposes and workload vendors are expected to consume smaller Flavours and adopt microservices-based designs for their workloads.

2) In Kubernetes based environments these are the resource requests of the containers in the pods. To get guaranteed resources the resource requests should be set to the same values as the resource limits, to get burstable resources the resource limits should be higher than the resource requests while to get best effort resources none of resource requests of resource limits should be set.

### Virtual Network Interface Specifications

The virtual network interface specifications extend a Flavour customization with network interface(s), with an associated bandwidth, and are identified by the literal, “n”, followed by the interface bandwidth (in Gbps). Multiple network interfaces can be specified by repeating the “n” option.

Virtual interfaces may be of an Access type, and thereby untagged, or may be of a Trunk type, with one or more 802.1Q tagged logical interfaces.

**Note**: Tagged interfaces are encapsulated by the Overlay, such that tenant isolation (i.e. security) is maintained, irrespective of the tag value(s) applied by the workload.

Note, the number of virtual network interfaces, aka vNICs, associated with a virtual compute instance, is directly related to the number of vNIC extensions declared for the environment. The vNIC extension is not part of the base Flavour.

<network interface bandwidth option> :: <”n”><number (bandwidth in Gbps)>

|  |  |
| --- | --- |
| Virtual Network Interface Option | Interface Bandwidth |
| n1, n2, n3, n4, n5, n6 | 1, 2, 3, 4, 5, 6 Gbps |
| n10, n20, n30, n40, n50, n60 | 10, 20, 30, 40, 50, 60 Gbps |
| n25, n50, n75, n100, n125, n150 | 25, 50, 75, 100, 125, 150 Gbps |
| n50, n100, n150, n200, n250, n300 | 50, 100, 150, 200, 250, 300 Gbps |
| n100, n200, n300, n400, n500, n600 | 100, 200, 300, 400, 500, 600 Gbps |

1. : Virtual Network Interface Specification Examples

### Storage Extensions

Persistent storage is associated with workloads via Storage Extensions. The size of an extension can be specified explicitly in increments of 100GB, ranging from a minimum of 100GB to a maximum of 16TB. Extensions are configured with the required performance category, as per Table 19. Multiple persistent Storage Extensions can be attached to virtual compute instances.

**Note**: CNTT documentation uses GB and GiB to refer to a Gibibyte (230 bytes), except where explicitly stated otherwise.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| .conf | Read IO/s | Write IO/s | Read Throughput (MB/s) | Write Throughput (MB/s) | Max Ext Size |
| .bronze | Up to 3K | Up to 1.5K | Up to 180 | Up to 120 | 16TB |
| .silver | Up to 60K | Up to 30K | Up to 1200 | Up to 400 | 1TB |
| .gold | Up to 680K | Up to 360K | Up to 2650 | Up to 1400 | 1TB |

1. : Storage Performance Profiles

**Note**: performance is based on a block size of 256KB or larger.

1. : Reserved

### Cloud Infrastructure Profiles

#### Basic Profile

This Cloud Infrastructure Profile is intended to be used for both IT workloads as well as NFV workloads. It has limited IO capabilities (up to 10Gbps Network interface).

#### Network Intensive Profile

This Cloud Infrastructure Profile is intended to be used for those applications that has high network throughput requirements (up to 50Gbps).

##### Network Acceleration Extensions

Network Intensive Profile can come with Network Acceleration extensions to assist workloads offloading some of their network intensive operations to hardware. The list below is preliminary and is expected to grow as more network acceleration resources are developed and standardized.

Interface types are aligned with [ETSI GS NFV-IFA 002](https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/002/02.01.01_60/gs_NFV-IFA002v020101p.pdf).

|  |  |  |
| --- | --- | --- |
| .conf | Interface type | Description |
| .il-ipsec | virtio-ipsec\* | In-line IPSec acceleration. |
| .la-crypto | virtio-crypto | Look-Aside encryption/decryption engine. |

1. : Acceleration Extensions for Network Intensive Profile

\*Need to work with relevant open source communities to create missing interfaces.

### Cloud Infrastructure Profile Capabilities Mapping

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Basic | Network Intensive | Notes |
| e.cap.001(#vCPU cores) | Per selected <Flavour> | Per selected <Flavour> | Exposed resource capabilities as per [Table 5](#Table4-1) |
| e.cap.002(Amount of RAM (MB)) | Per selected <Flavour> | Per selected <Flavour> |  |
| e.cap.003(Total instance (ephemeral) storage (GB)) | Per selected <Flavour> | Per selected <Flavour> |  |
| e.cap.004(# vNICs) | Per selected | Per selected |  |
| e.cap.005(Total instance (persistent) storage (GB)) | Per selected | Per selected |  |
| e.cap.006(CPU pinning support) | No | Yes | Exposed performance capabilities as per [Table 6](#Table4-2) |
| e.cap.007(NUMA support) | No | Yes |  |
| e.cap.008(IPSec Acceleration) | No | Yes (if offered) |  |
| e.cap.009(Crypto Acceleration) | No | Yes (if offered) |  |
| e.cap.010(Transcoding Acceleration) | No | No |  |
| e.cap.011(Programmable Acceleration) | No | No |  |
| e.cap.012(Enhanced Cache Management) | E | E |  |
| e.cap.013(SR-IOV over PCI-PT) | Yes | No |  |
| e.cap.014(GPU/NPU) | No | No |  |
| e.cap.015(SmartNIC) | Yes (if offered) | No |  |
| e.cap.016(FPGA/other Acceleration H/W) | Yes (if offered) | No |  |
| i.cap.014(CPU cores consumed by the Cloud Infrastructure on the worker nodes) | any | any |  |
| i.cap.015(Memory consumed by Cloud Infrastructure on the worker nodes) | any | any |  |
| i.cap.016(CPU allocation ratio) | 4:1 | 1:1 | Internal SLA capabilities as per [Table 10](#Table4-6)  ***Note****:* This is set to 1:1 for the Basic profile to enable predictable and consistent performance during benchmarking and certification. Operators may choose to modify this for actual deployments if they are willing to accept the risk of performance impact to workloads using the basic profile. |
| i.cap.017(Connection point QoS) | No | Yes |  |
| i.cap.018(Huge page support) | No | Yes | Internal performance capabilities as per [Table 11](#Table4-7) |
| i.pm.001(Host CPU usage) | Yes | Yes | Internal monitoring capabilities as per [Table 12](#Table4-8) |
| i.pm.002(Virtual compute CPU usage) | Yes | Yes |  |
| i.pm.003(Host CPU utilization) | Yes | Yes |  |
| i.pm.004(Virtual compute CPU utilization) | Yes | Yes |  |
| i.pm.005(Measurement of external storage IOPs) | Yes | Yes |  |
| i.pm.006(Measurement of external storage throughput) | Yes | Yes |  |
| i.pm.007(Available external storage capacity) | Yes | Yes |  |

1. : Mapping of Capabilities to Cloud Infrastructure Profiles

### Cloud Infrastructure Profile Performance Measurement Mapping

**Comment:** To be worked on.

### One stop shop

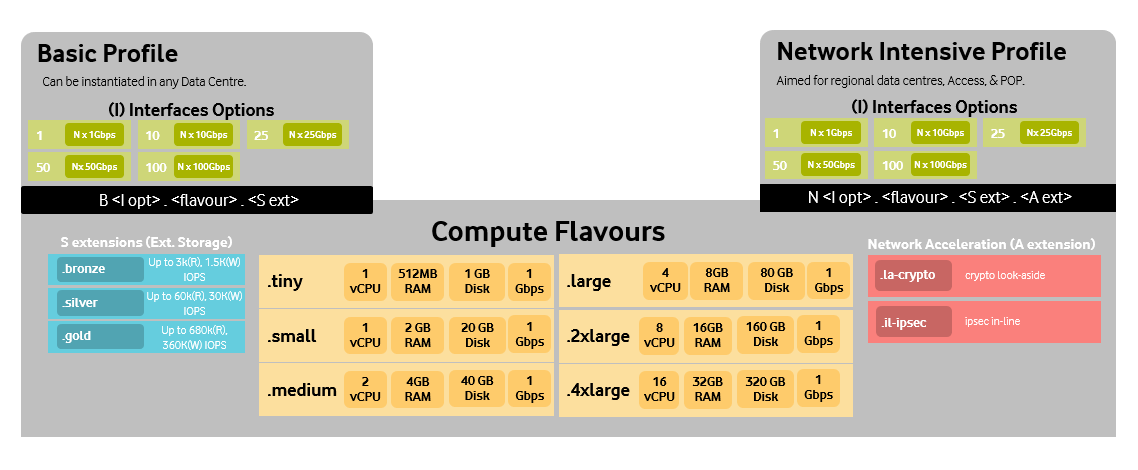
#### Naming convention

An entry in the infrastructure profile catalogue can be referenced using the following naming convention.

B/N <I opt> . <Flavour> . <S ext> . <A ext>

Whereas:

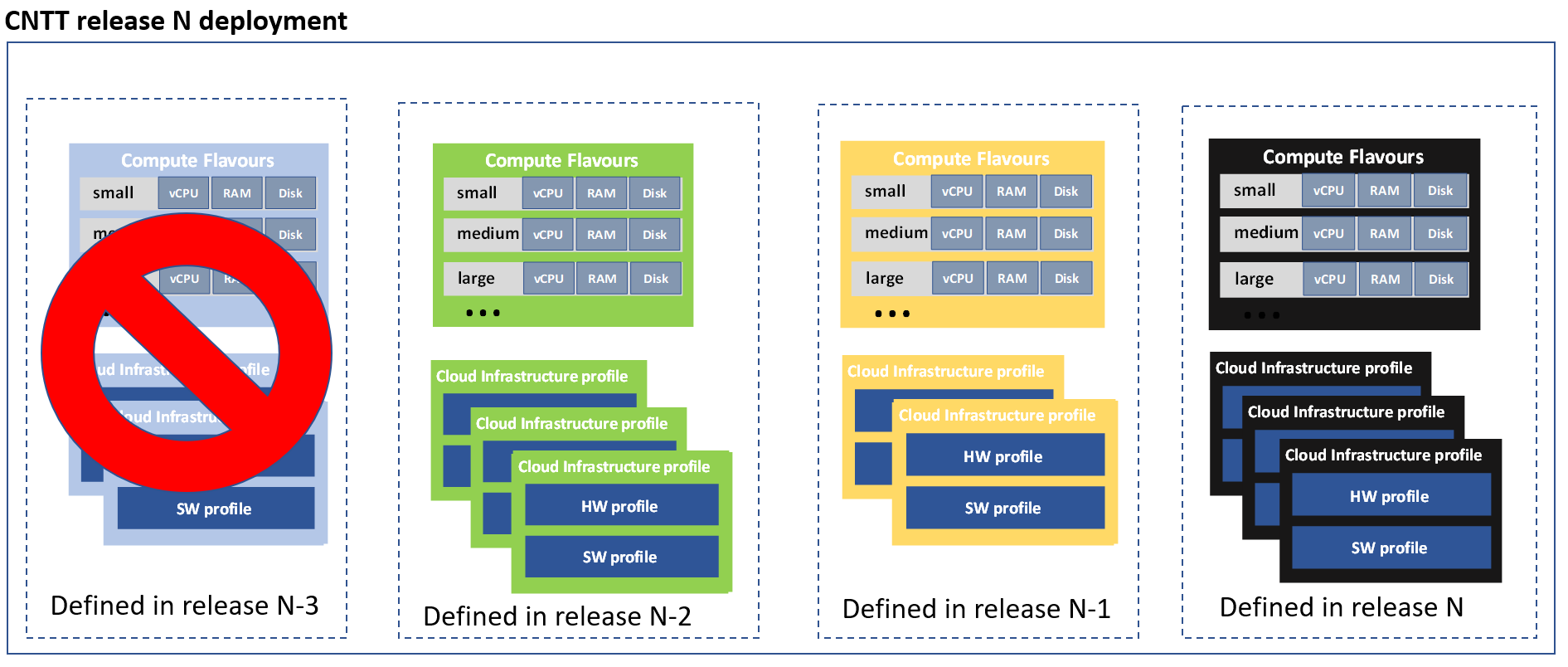
* **B/N**: specifies the Cloud Infrastructure Profile (Basic or Network Intensive)
* **<I opt>**: specifies the interface option.
* **<Flavour>**: specifies the compute Flavour.
* **<S ext>**: specifies an optional storage extension.
* **<A ext>**: specifies an optional acceleration extension for the Network Intensive Cloud Infrastructure Profile.



1. : Infrastructure Profiles Catalogue

#### Backwards Compatibility

The Reference Model (RM) specification describes an infrastructure abstraction including a set of cloud infrastructure hardware and software profiles and compute flavours offered to workloads. The set of defined profiles and flavours will evolve along the releases but at the same time the existing workloads need to be supported. This means that any CNTT deployed cloud should be backwards compatible and support profiles and flavours from the latest three CNTT releases (N-2, N-1, N) as presented in Figure 10.



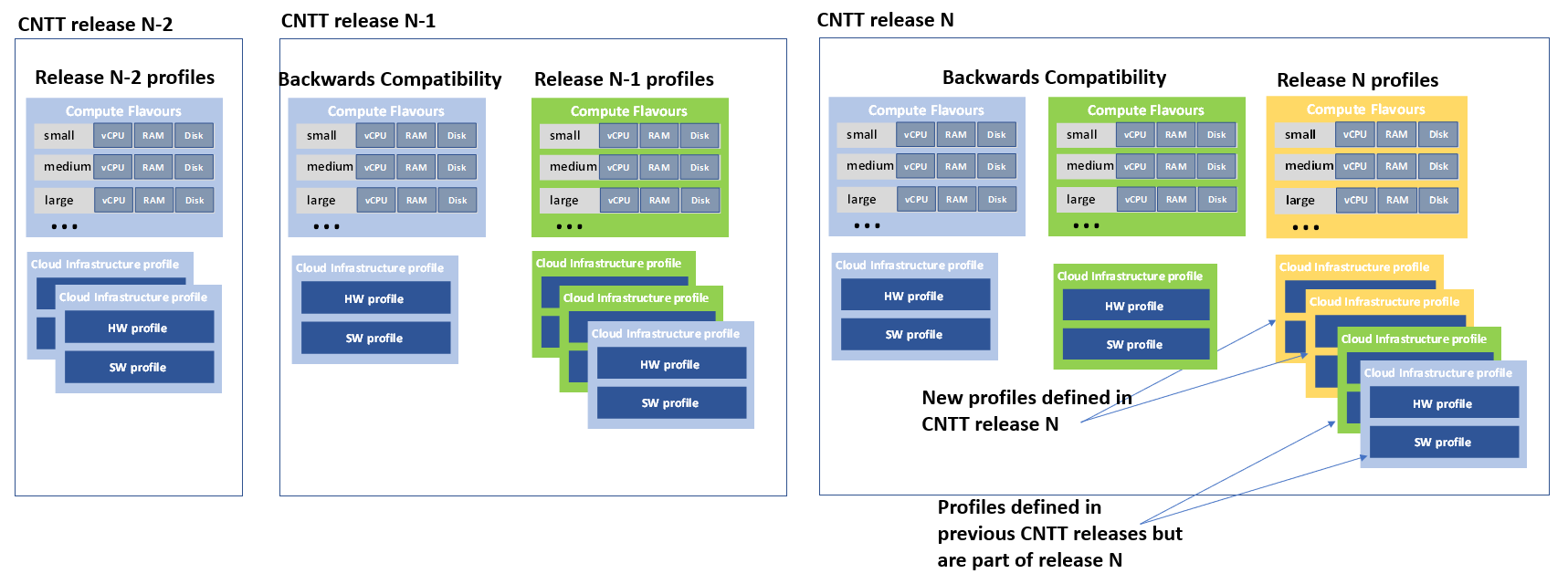
1. : Backwards Compatibility

Cloud Infrastructure profiles that are available in CNTT release N deployment can be divided into two categories:

1. Cloud infrastructure profiles that are part of CNTT release N. These can be either
   * new profiles defined in release N or
   * existing profiles from earlier releases that are incorporated for backward compatibility reasons in release N
2. Cloud infrastructure profiles from releases N-1 and N-2 that are deployed only because of backwards compatibility, these profiles are not part of CNTT release N definition.

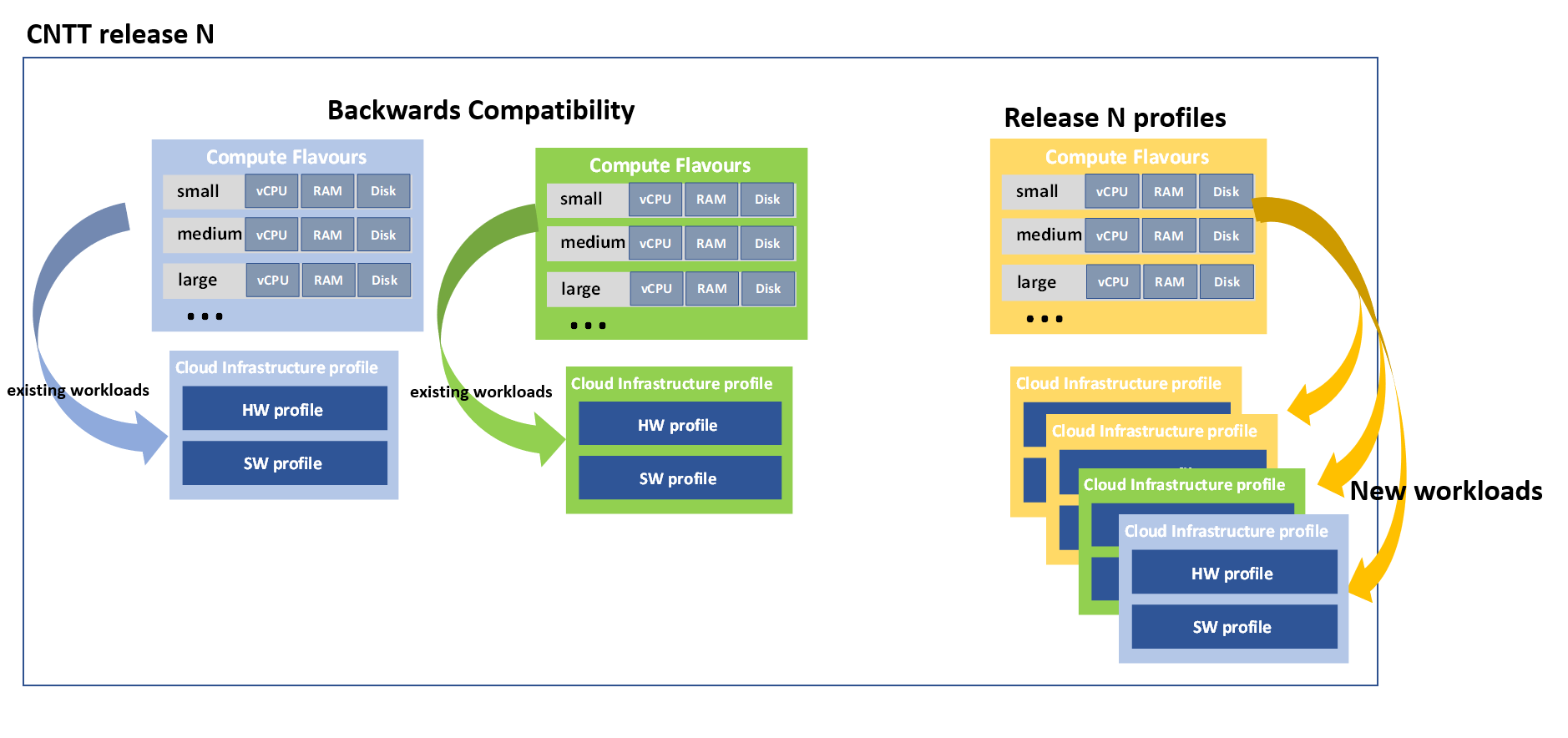
**Note:** a profile defined in previous releases that is modified in release N is considered to be a new profile

Different profile categories described above are presented in Figure 11. In this example profiles that are part of CNTT release N consist of two new profiles (yellow), one profile that is originally defined in release N-1 (green) and one defined in release N-2 (blue). Profiles that were defined in earlier releases but are also supported in release N will be referred to by several names. Existing workloads continue using the profile names from previous releases. New workloads will use release N naming.



1. : Cloud Infrastructure profiles in CNTT release N

Like predefined cloud infrastructure profiles, predefined compute flavours are also specified per CNTT release. CNTT release N flavours are used when new workloads are deployed into profiles that are part of the CNTT N release. Existing workloads continue using the flavours from previous releases. The difference in flavours can be for example, that newer flavours defined in release N may not have extra-large flavours that are earlier defined for transitional purposes. Workloads that use backwards compatible profiles will use the flavours from the older release (Figure 12).



1. : New workloads in Release N would use only Release N profiles

As discussed above backwards compatibility is the reason why cloud infrastructure profiles and flavours from several CNTT releases are configured and used in one CNTT deployment. Therefore, CNTT release number need to be added to each profile:

B/N<”\_Gen”><release #>. <Flavour>

Flavours are unique only when combined with a profile. For example, CNTT release N small flavour in basic profile has the naming:

B\_GenN.small

#### Forward compatibility

CNTT provides a framework for exceptions described in [9.2.3 Transition Framework](file:///C:\Users\tpelt\Documents\Infrastructure\cNTT\RM\gov\chapters\chapter09.md#9.2). The exceptions of a given CNTT release are listed in [A.3](file:///C:\Users\tpelt\Documents\Infrastructure\cNTT\RM\ref_model\output\appendix-a.md#a3-exception-list) Exception List. The exceptions are not part of any Cloud Infrastructure profile defined in CNTT. If a flavour needs to be defined to support one or more exceptions its name should contain the identifyer of the exception. If needed several exceptions can be combined into the same flavour.

The naming scheme for flavours with exceptions should be B\_GenN.small.ExceptionIds where the exception is is generated from the numerical part of the exception identifier prefixed with ex.

For example B\_Gen4.small.ex001 refers to rm.exc.001 defined in the [Exception List](file:///C:\Users\tpelt\Documents\Infrastructure\cNTT\RM\ref_model\output\appendix-a.md#a3-exception-list). These flavors, similarly to other flavours, should be supported for three CNTT releases after the exception was removed from the CNTT release.

## Networking

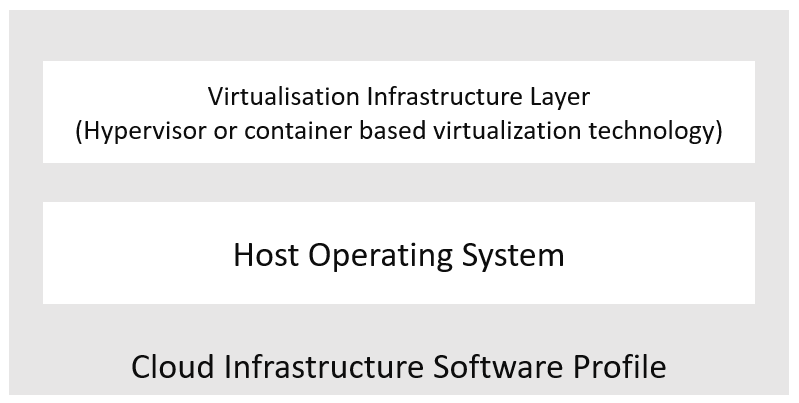
This is a placeholder for Cloud Infrastructure infrastructure networking information that is common to all Reference Architectures.

# Feature set and Requirements from Infrastructure

## Cloud Infrastructure Software profile description

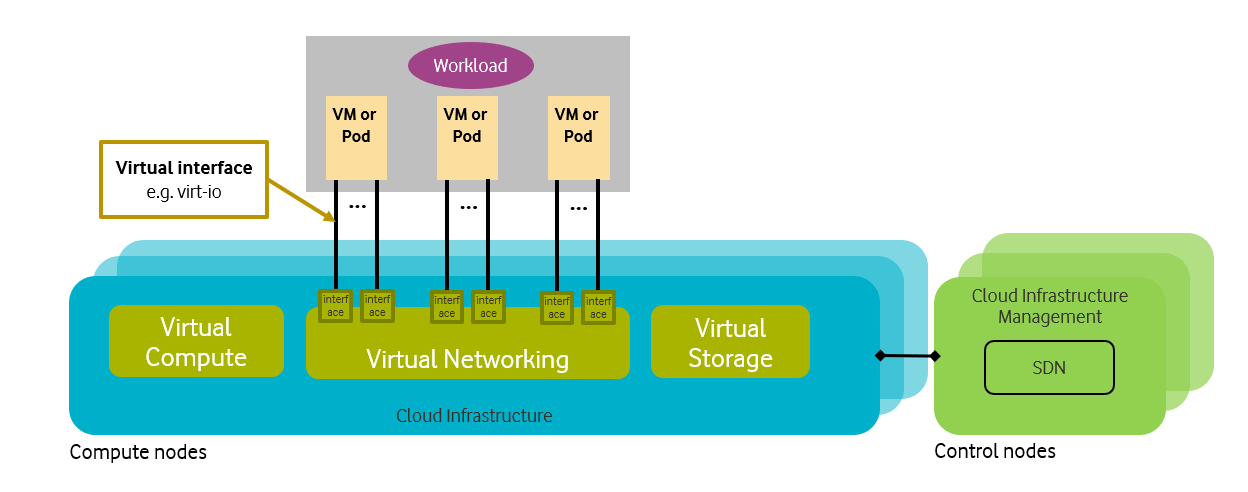
Cloud Infrastructure Software layer is composed of 2 layers, Figure 13:

* + The virtualisation Infrastructure layer, which is based on hypervisor virtualisation technology or container-based virtualisation technology. Container virtualisation can be nested in hypervisor-based virtualisation
  + The host OS layer



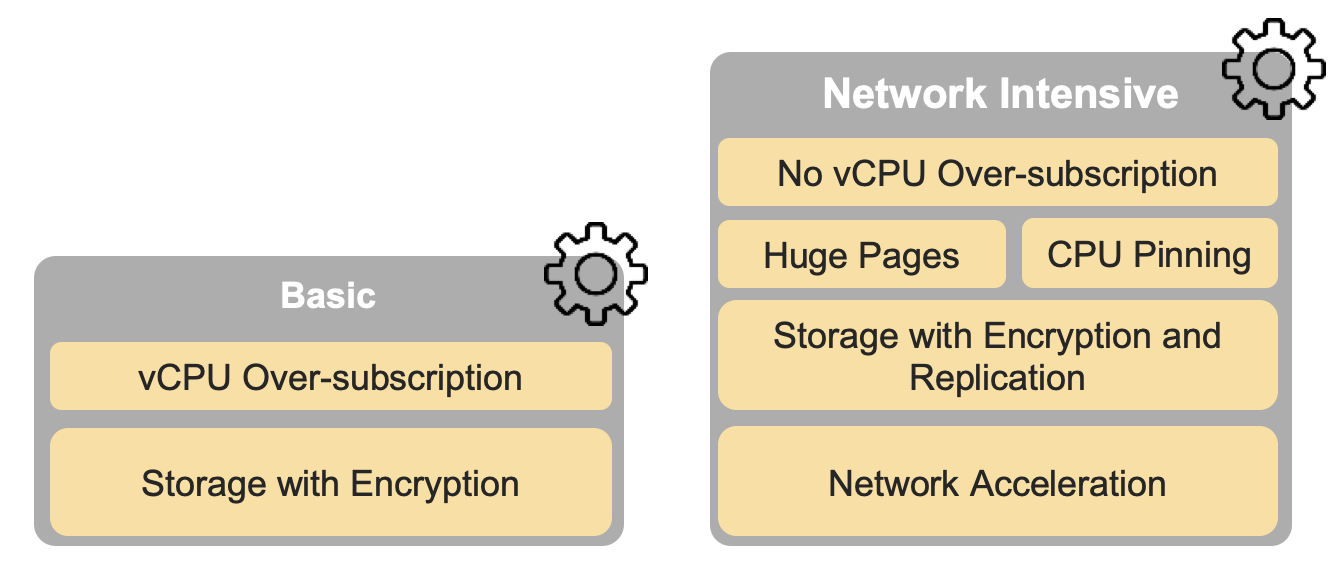
1. : Cloud Infrastructure software layers

For a host (compute node or physical server), the virtualisation layer is an abstraction layer between hardware components (compute, storage, and network resources) and virtual resources allocated to a VM or a Pod. Figure 14 represents the virtual resources (virtual compute, virtual network, and virtual storage) allocated to a VM or a Pod and managed by the Cloud Infrastructure Manager.



1. : Cloud Infrastructure Virtual resources

Depending on the requirements of the workloadss, a VM or a Pod will be deployed with a Cloud Infrastructure Profile and an appropriate compute flavour. A Cloud Infrastructure Profile is defined by a Cloud Infrastructure Software Profile and a Cloud Infrastructure Hardware Profile. A Cloud Infrastructure Software Profile is a set of features, capabilities, and metrics offered by an Cloud Infrastructure software layer. Figure 15 depicts a high level view of the Basic and Network Intensive Cloud Infrastructure Profiles.



1. : Cloud Infrastructure Profiles

The following sections detail the Cloud Infrastructure Software Profile features per type of virtual resource. The list of these features will evolve over time.

### Virtual Compute

Table 23 and Table 24 depict the features related to virtual compute.

|  |  |  |  |
| --- | --- | --- | --- |
| .conf | Feature | Type | Description |
| infra.com.cfg.001 | CPU allocation ratio | Value | Number of virtual cores per physical core |
| infra.com.cfg.002 | NUMA awareness | Yes/No | Support of NUMA at the virtualization layer |
| infra.com.cfg.003 | CPU pinning capability | Yes/No | Binds a process/vCPU to a physical core or SMT thread |
| infra.com.cfg.004 | Huge Pages | Yes/No | Ability to manage huge pages of memory |

1. : Virtual Compute features.

|  |  |  |  |
| --- | --- | --- | --- |
| .conf | Feature | Type | Description |
| infra.com.acc.cfg.001 | ***Editor Note:*** *To be worked on* |  |  |

1. : Virtual Compute Acceleration features.

### Virtual Storage

Table 25 and Table 26 depict the features related to virtual storage.

|  |  |  |  |
| --- | --- | --- | --- |
| .conf | Feature | Type | Description |
| infra.stg.cfg.001 | Storage Types | Yes/No | Support of Storage types described in the catalogue |
| infra.stg.cfg.002 | Storage Block | Yes/No |  |
| infra.stg.cfg.003 | Storage with replication | Yes/No |  |
| infra.stg.cfg.004 | Storage with encryption | Yes/No |  |

1. : Virtual Storage features.

|  |  |  |  |
| --- | --- | --- | --- |
| .conf | Feature | Type | Description |
| infra.stg.acc.cfg.001 | Storage IOPS oriented | Yes/No |  |
| infra.stg.acc.cfg.002 | Storage capacity oriented | Yes/No |  |

1. : Virtual Storage Acceleration features.

### Virtual Networking

Table 27 and Table 28 depict the features related to virtual networking.

|  |  |  |  |
| --- | --- | --- | --- |
| .conf | Feature | Type | Description |
| infra.net.cfg.001 | vNIC interface | IO virtualisation | e.g. virtio1.1 |
| infra.net.cfg.002 | Overlay protocol | Protocols | The overlay network encapsulation protocol needs to enable ECMP in the underlay to take advantage of the scale-out features of the network fabric. |
| infra.net.cfg.003 | NAT | Yes/No | Support of Network Address Translation |
| infra.net.cfg.004 | Security Groups | Yes/No | Set of rules managing incoming and outgoing network traffic |
| infra.net.cfg.005 | SFC | Yes/No | Support of Service Function Chaining |
| infra.net.cfg.006 | Traffic patterns symmetry | Yes/No | Traffic patterns should be optimal, in terms of packet flow. North-south traffic shall not be concentrated in specific elements in the architecture, making those critical choke-points, unless strictly necessary (i.e. when NAT 1:many is required). |

1. : Virtual Networking features.

|  |  |  |  |
| --- | --- | --- | --- |
| .conf | Feature | Type | Description |
| infra.net.acc.cfg.001 | vSwitch optimisation | Yes/No and SW Optimisation | e.g. DPDK. |
| infra.net.acc.cfg.002 | Support of HW offload | Yes/No | e.g. support of SmartNic. |
| infra.net.acc.cfg.003 | Crypto acceleration | Yes/No |  |
| infra.net.acc.cfg.004 | Crypto Acceleration Interface | Yes/No |  |

1. : Virtual Networking Acceleration features.

### Security

**Comment:** To be worked on.

### Platform Services

This section details the services that may be made available to workloads by the Cloud Infrastructure.

|  |  |  |  |
| --- | --- | --- | --- |
| .conf | Feature | Type | Description |
| infra.svc.stg.001 | Object Storage | Yes/No | Object Storage Service (e.g S3-compatible) |

1. : Cloud Infrastructure Platform services.

| Minimum requirements | Example | |----------------------|--------------------------------------------| |Database as a service | Cassandra | |Queue | Rabbit MQ | |LB and HA Proxy | |

## Cloud Infrastructure Software Profiles features and requirements

This section will detail Cloud Infrastructure Software Profiles and associated configurations for the 2 types of Cloud Infrastructure Profiles: Basic and Network intensive.

### Virtual Compute

Table 30 depicts the features and configurations related to virtual compute for the 2 types of Cloud Infrastructure Profiles.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| .conf | Feature | Type | Basic | Network Intensive | Notes |
| infra.com.cfg.001 | CPU allocation ratio | value | 1:1 | 1:1 | **Note**: This is set to 1:1 for the Basic profile to enable predictable and consistent performance during benchmarking and certification. Operators may choose to modify this for actual deployments if they are willing to accept the risk of performance impact to workloads using the basic profile. |
| infra.com.cfg.002 | NUMA awareness | Yes/No | N | Y |  |
| infra.com.cfg.003 | CPU pinning capability | Yes/No | N | Y |  |
| infra.com.cfg.004 | Huge Pages | Yes/No | N | Y |  |

1. : Virtual Compute features and configuration for the 2 types of Cloud Infrastructure Profiles.

**Note**: Capability nfvi.com.cfg.001 is set to 1:1 for the Basic profile to enable predictable and consistent performance during benchmarking, certification, and deployment. Operators may choose to modify this for actual deployments if they are willing to accept the risk of performance impact to these workloads.

Table 31 will gather virtual compute acceleration features. It will be filled over time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .conf | Feature | Type | Basic | Network Intensive |
| infra.com.acc.cfg.001 | ***Editor Note:*** *To be worked on* |  |  |  |

1. : Virtual Compute Acceleration features.

### Virtual Storage

Table 32 and Table 33 depict the features and configurations related to virtual storage for the 2 types of Cloud Infrastructure Profiles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .conf | Feature | Type | Basic | Network Intensive |
| infra.stg.cfg.001 | Catalogue storage Types | Yes/No | Y | Y |
| infra.stg.cfg.002 | Storage Block | Yes/No | Y | Y |
| infra.stg.cfg.003 | Storage with replication | Yes/No | N | Y |
| infra.stg.cfg.004 | Storage with encryption | Yes/No | Y | Y |

1. : Virtual Storage features and configuration for the 2 types of SW profiles.

Table 33 depicts the features related to Virtual storage Acceleration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .conf | Feature | Type | Basic | Network Intensive |
| infra.stg.acc.cfg.001 | Storage IOPS oriented | Yes/No | N | Y |
| infra.stg.acc.cfg.002 | Storage capacity oriented | Yes/No | N | N |

1. : Virtual Storage Acceleration features.

### Virtual Networking

Table 34 and Table 35 depict the features and configurations related to virtual networking for the 2 types of Cloud Infrastructure Profiles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .conf | Feature | Type | Basic | Network Intensive |
| infra.net.cfg.001 | vNIC interface | IO virtualisation | virtio1.1 | virtio1.1\* |
| infra.net.cfg.002 | Overlay protocol | Protocols | VXLAN, MPLSoUDP, GENEVE, other |  |
| infra.net.cfg.003 | NAT | Yes/No | Y | Y |
| infra.net.cfg.004 | Security Group | Yes/No | Y | Y |
| infra.net.cfg.005 | SFC support | Yes/No | N | Y |
| infra.net.cfg.006 | Traffic patterns symmetry | Yes/No | Y | Y |

Notes: \*[Workload Transtion Guidelines.](file:///C:\\Users\\tpelt\\Documents\\Infrastructure\\cNTT\\RM\\ref_model\\chapters\\appendix-a.md) might have other interfaces (such as SR-IOV VFs to be directly passed to a VM or a Pod) or NIC-specific drivers on guest machines transiently allowed until mature enough solutions are available with a similar efficiency level (for example regarding CPU and energy consumption).

1. : Virtual Networking features and configuration for the 2 types of SW profiles.

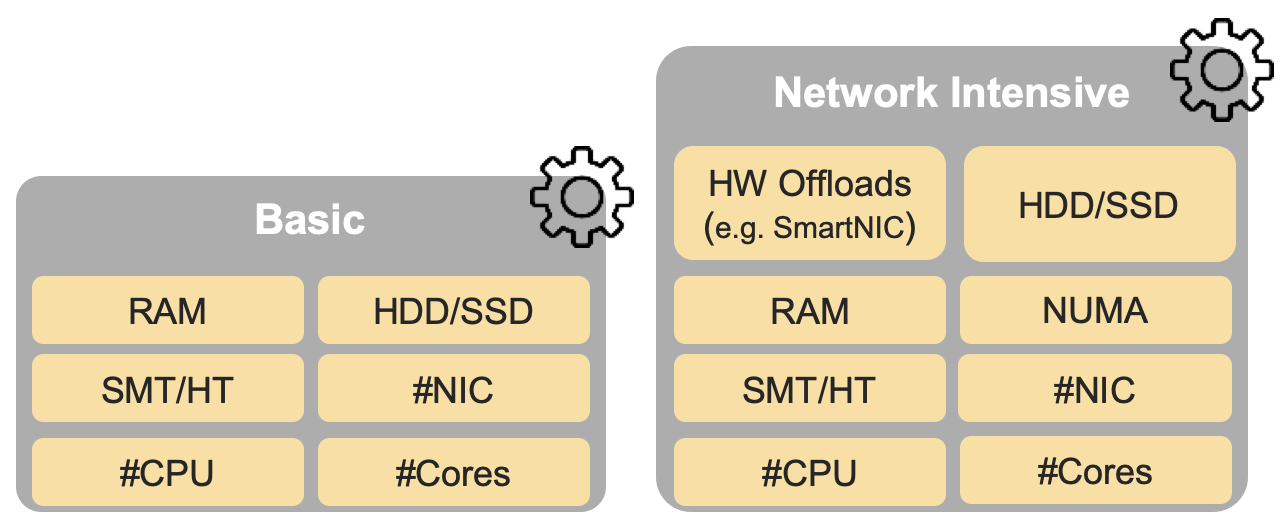
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .conf | Feature | Type | Basic | Network Intensive |
| infra.net.acc.cfg.001 | vSwitch optimisation | Yes/No and SW Optimisation | N | Y, DPDK |
| infra.net.acc.cfg.002 | Support of HW offload | Yes/No | N | Y, support of SmartNic |
| infra.net.acc.cfg.003 | Crypto acceleration | Yes/No | N | Y |
| infra.net.acc.cfg.004 | Crypto Acceleration Interface | Yes/No | N | Y |

1. : Virtual Networking Acceleration features.

## Cloud Infrastructure Hardware Profile description

The support of a variety of different workload types, each with different (sometimes conflicting) compute, storage, and network characteristics, including accelerations and optimizations, drives the need to aggregate these characteristics as a hardware (host) profile and capabilities. A host profile is essentially a “personality” assigned to a compute host (physical server, also known as compute host, host, node, or pServer). The host profiles and related capabilities consist of the intrinsic compute host capabilities (such as #CPUs (sockets), # of cores/CPU, RAM, local disks and their capacity, etc.), and capabilities enabled in hardware/BIOS, specialised hardware (such as accelerators), the underlay networking, and storage.

This chapter defines a simplified host, host profile and related capabilities model associated with each of the different Cloud Infrastructure Hardware Profile and related capabilities; some of these profiles and capability parameters are shown in Figure 16.



1. : Cloud Infrastructure Hardware Profiles and host associated capabilities.

The host profile model and configuration parameters (hereafter for simplicity simply "host profile") will be used in the **Reference Architecture** to define different hardware profiles. The host profiles can be considered to be the set of EPA-related (Enhanced Performance Awareness) configurations on Cloud Infrastructure resources.

Please note that in this chapter we shall not list all of the EPA-related configuration parameters.

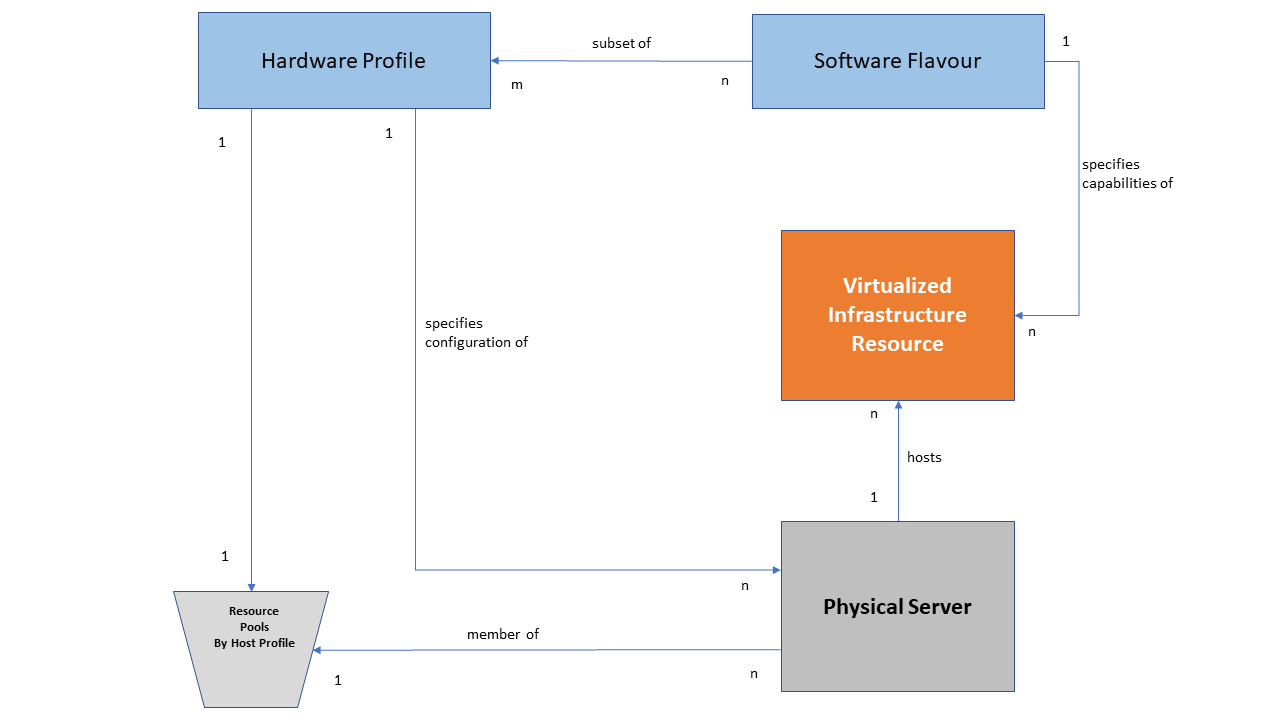
A software profile (see **Chapter 4**, **5.1 and 5.2**) defines the characteristics of Cloud Infrastructure SW of which Virtual Machines or Containers will be deployed on. A many to many relationship exists between software profiles and host profiles. A given host can only be assigned a single host profile; a host profile can be assigned to multiple hosts. Different Cloud Service Providers (CSP) may use different naming standards for their host profiles.

The following naming convention is used in this document:

<host profile name>:: <”hp”><numeral host profile sequence #>

When a software profile is associated with a host profile, a qualified name can be used as specified below. ***For Example:*** *for software profile “n” (network intensive) the above host profile name would be “n-hp1”*.

<qualified host profile>:: <software profile><”-“><”hp”><numeral host profile sequence #>



1. : Generic Hardware Profile, Software Flavour, Physical server relationship.

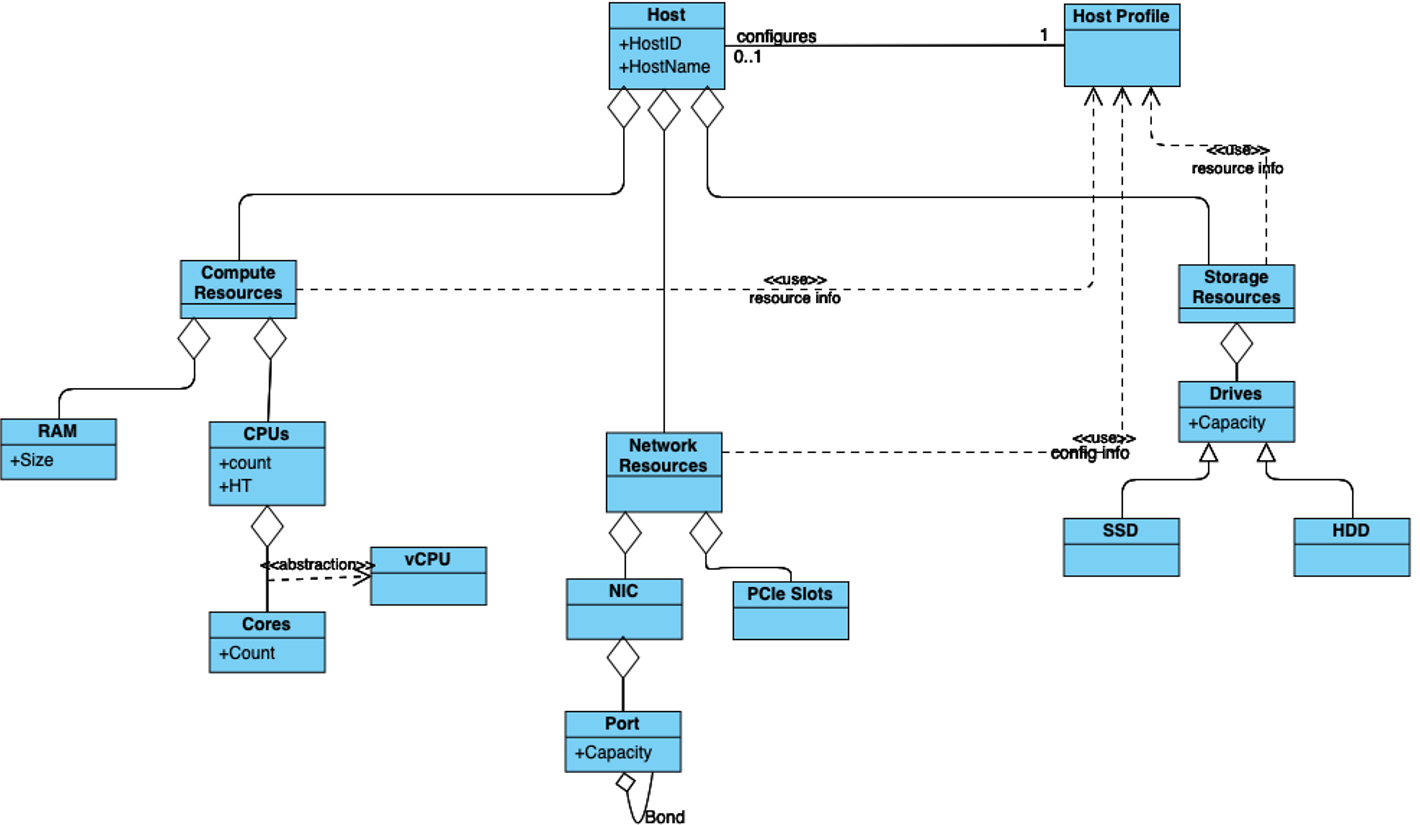
Figure 17 shows a simplistic depiction of the relationship between Hardware profile, Software Profile, Physical server, and virtual compute. In the diagram the resource pool, a logical construct, depicts all physical hosts that have been configured as per a given host profile; there is one resource pool for each hardware profile.

**Note**: resource pools are not OpenStack host aggregates.

The host profile and capabilities include:

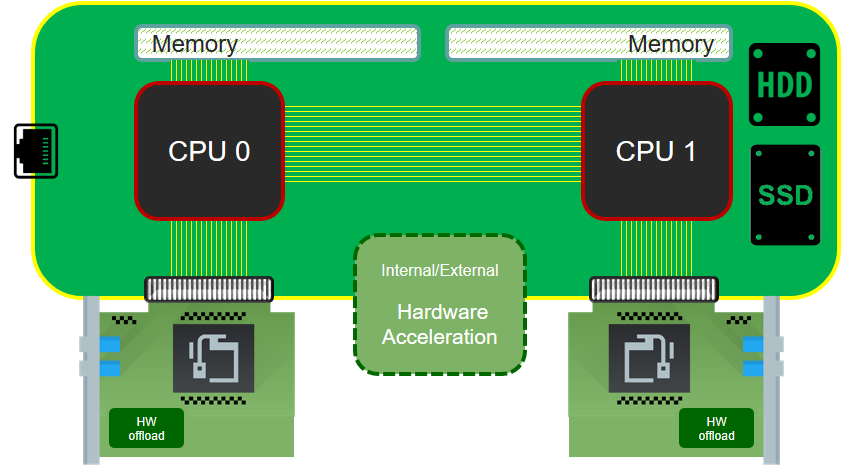
1. **# of CPUs (sockets)**: is the #of CPUs installed on the physical server.
2. **# of cores/CPU**: is the number of cores on each of the CPUs of the physical server.
3. **RAM (GB)**: is the amount of RAM installed on the pysical server.
4. **Local Disk Capacity**: is the # of local disks and teh capacity of the disks installed on the physical server.
5. **SMT/HT (SMT: Simultaneous Multithreading/ HT: Hyper Threading)**: Enabled on all physical servers. Gets multiple threads per physical core. Always ON. Configured in the host.
6. **NUMA (Non-Uniform Memory Access)**: Indicates that vCPU will be on a Socket that is aligned with the associated NIC card and memory. Important for performance optimized workloads. Configured in the host.
7. **SR-IOV (Single-Root Input/Output Virtualisation)**: Configure PCIe ports to enable SR-IOV.
8. **smartNIC (aka Intelligent Server Adaptors)**: Accelerated virtual switch using smartNIC
9. **Cryptography Accelerators**: such as AES-NI, SIMD/AVX, QAT.
10. **Security features**: such as TRusted Platform Module (TPM).

The following model, Figure 18, depicts the essential characteristics of a host that are of interest in specifying a host profile. The host (physical server) is composed of compute, network, and storage resources. The compute resources are composed of physical CPUs (aka CPU sockets or sockets) and memory (RAM). The network resources and storage resources are similarly modelled.



1. : Generic model of a compute host for use in Host Profile configurations.

The hardware (host) profile properties are specified in the following sub-sections. The following diagram (Figure 19) pictorially represents a high-level abstraction of a physical server (host).



1. : Generic model of a compute host for use in Host Profile configurations.

## Cloud Infrastructure Hardware Profiles features and requirements.

The configurations specified in here will be used in specifying the actual hardware profile configurations for each of the Cloud Infrastructure Hardware Profiles depicted in Figure 16.

### Compute Resources

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Feature | Description | Basic Type | Network Intensive |
| infra.hw.cpu.cfg.001 | Number of CPU (Sockets) | This determines the minimum number of CPU sockets within each host | 2 | 2 |
| infra.hw.cpu.cfg.002 | Number of Cores per CPU | This determines the number of cores needed per CPU. | 20 | 20 |
| infra.hw.cpu.cfg.003 | NUMA | NUMA support and BIOS configured to enable NUMA | N | Y |
| infra.hw.cpu.cfg.004 | Simultaneous Multithreading/Hyperthreading (SMT/HT) | This allows a CPU to work multiple streams of data simultaneously | Y | Y |

1. : Minimum Compute resources configuration parameters.

#### Compute Acceleration Hardware Specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Feature | Description | Basic Type | Network Intensive |
| infra.hw.cac.cfg.001 | GPU | GPU | N | N |

1. : Compute acceleration configuration specifications.

### Storage Configurations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Feature | Description | Basic Type | Network Intensive |
| infra.hw.stg.hdd.cfg.001\* | Local Storage HDD | Hard Disk Drive |  |  |
| infra.hw.stg.ssd.cfg.002\* | Local Storage SSD | Solid State Drive | Recommended | Recommended |

1. : Storage configuration specification.

*\*This specified local storage configurations including # and capacity of storage drives.*

### Network Resources

#### NIC configurations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Feature | Description | Basic Type | Network Intensive |
| infra.hw.nic.cfg.001 | NIC Ports | Total Number of NIC Ports available in the host | 4 | 4 |
| infra.hw.nic.cfg.002 | Port Speed | Port speed specified in Gbps (minimum values) | 10 | 25 |

1. : Minimum NIC configuration specification.

#### PCIe Configurations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Feature | Description | Basic Type | Network Intensive |
| infra.hw.pci.cfg.001 | PCIe slots | Number of PCIe slots available in the host | 8 | 8 |
| infra.hw.pci.cfg.002 | PCIe speed |  | Gen 3 | Gen 3 |
| infra.hw.pci.cfg.003 | PCIe Lanes |  | 8 | 8 |

1. : PCIe configuration specification.

#### Network Acceleration Configurations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Feature | Description | Basic Type | Network Intensive |
| infra.hw.nac.cfg.001 | Cryptographic Acceleration | IPSec, Crypto | N | Optional |
| infra.hw.nac.cfg.002 | SmartNIC | A SmartNIC that is used to offload network functionality to hardware | N | Optional |
| infra.hw.nac.cfg.003 | Compression |  |  |  |

1. : Network acceleration configuration specification.

# External Interfaces

## Introduction

In this document’s earlier chapters, the various resources and capabilities of the Cloud Infrastructure have been catalogued and the workloads have been profiled with respect to those capabilities. The intent behind this chapter and an “API Layer” is to similarly provide a single place to catalogue and thereby codify, a common set of open APIs to access (i.e. request, consume, control, etc.) the aforementioned resources, be them directly exposed to the workloads, or purely internal to the Cloud Infrastructure.

It is a further intent of this chapter and this document to ensure the APIs adopted for CNTT Cloud Infrastructure implementations are open and not proprietary, in support of compatibility, component substitution, and ability to realize maximum value from existing and future test heads and harnesses.

While it is the intent of this chapter, when included in a Reference Architecture, to catalogue the APIs, it is not the intent of this chapter to reprint the APIs, as this would make maintenance of the chapter impractical and the length of the chapter disproportionate within the Reference Model document. Instead, the APIs selected for CNTT Cloud Infrastructure implementations and specified in this chapter, will be incorporated by reference and URLs for the latest, authoritative versions of the APIs, provided in the References section of this document.

Although the document does not attempt to reprint the APIs themselves, where appropriate and generally where the mapping of resources and capabilities within the Cloud Infrastructure to objects in APIs would be otherwise ambiguous, this chapter shall provide explicit identification and mapping.

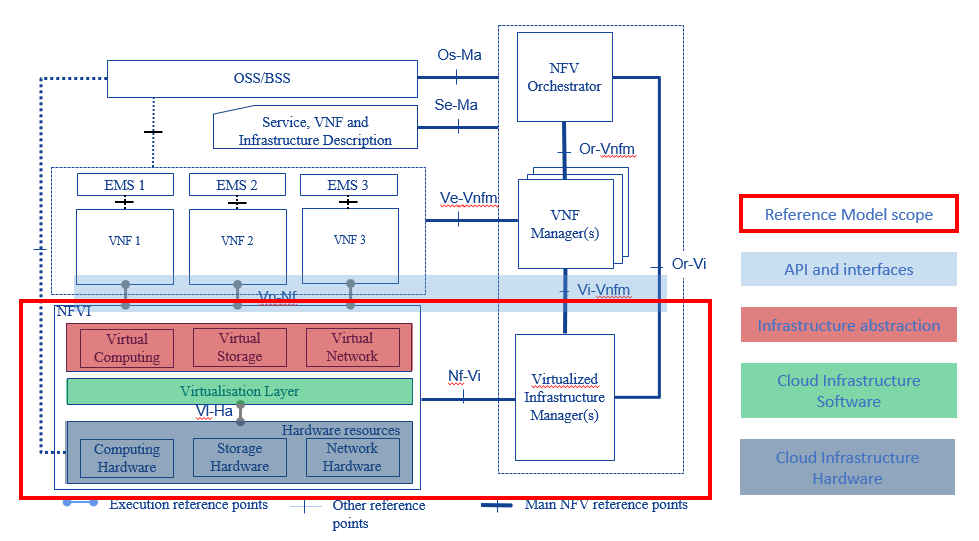
In addition to the raw or base-level Cloud Infrastructure functionality to API and object mapping, it is further the intent to specify an explicit, normalized set of APIs and mappings to control the logical interconnections and relationships between these objects, notably, but not limited to, support of SFC (Service Function Chaining) and other networking and network management functionality.

Chapter 3 introduced a model of the Cloud Infrastructure. Figure 3 shows an overview of the Cloud Infrastructure model including the external application programming interface (API)/ user interface (UI) for providing access to the Cloud Infrastructure management functions. Section 3.3 lists the actions supported by the Cloud Infrastructure Management Software. This chapter specifies the abstract interfaces (API, CLI, etc.) supported by the Cloud Infrastructure Reference Model. The purpose of this chapter is to define and catalogue a common set of open (not proprietary) APIs, of the following types:

* Cloud Infrastructure APIs: These APIs are provided to the workloads (i.e. exposed), by the infrastructure in order for workloads to access (i.e. request, consume, control, etc.) Cloud Infrastructure resources.
* Intra-Cloud Infrastructure APIs: These APIs are provided and consumed directly by the infrastructure. These APIs are purely internal to the Cloud Infrastructure and are not exposed to the workloads.
* Enabler Services APIs: These APIs are provided by non-Cloud Infrastructure services and provide capabilities that are required for a majority of workloads, e.g. DHCP, DNS, NTP, DBaaS, etc.

## Cloud Infrastructure APIs

The Cloud Infrastructure APIs consist of set of APIs that are externally and internally visible. The externally visible APIs are made available for orchestration and management of the execution environments that host workloads while the internally visible APIs support actions on the hypervisor and the physical resources. The ETSI NFV Reference MANO Architecture (Figure 20) shows a number of Interface points where specific or sets of APIs are supported. For the scope of the reference model the relevant interface points are shown in Table 42.



1. : ETSI NFV architecture mapping

|  |  |  |  |
| --- | --- | --- | --- |
| Interface Point | Cloud Infrastructure Exposure | Interface Between | Description |
| Vi-Ha | Internal NFVI | Software Layer and Hardware Resources | 1. Discover/collect resources and their configuration information 2. Create execution environment (e.g., VM) for workloads (VNF) |
| Vn-Nf | External | NFVI and VM (VNF) | Here VNF represents the execution environment. The interface is used to specify interactions between the VNF and abstract NFVI accelerators. The interfaces can be used to discover, configure, and manage these acceleartors and for the VNF to register/deregister for receiving acceleartor events and data. |
| NF-Vi | External | NFVI and VIM | 1. Discover/collect physical/virtual resources and their configuration information2. Manage (create, resize, (un) suspend, reboot, etc.) physical/virtualised resources3. Physical/Virtual resources configuration changes4. Physical/Virtual resource configuration. |
| Or-Vi | External | VNF Orchestrator and VIM | See below |
| Vi-Vnfm | External | VNF Manager and VIM | See below |

1. : NFVI and VIM Interfaces with Other System Components in the ETSI NFV architecture

The Or-Vi and Vi-VNfm are both specifying interfaces provided by the VIM and therefore are related. The Or-Vi reference point is used for exchanges between NFV Orchestrator and VIM, and supports the following interfaces; virtualised resources refers to virtualised compute, storage, and network resources:

* Software Image Management
* Virtualised Resources Information Management
* Virtualised Resources Capacity Management (only VNF Orchestrator and VIM (Or-Vi))
* Virtualised Resources Management
* Virtualised Resources Change Management
* Virtualised Resources Reservation Management
* Virtualised Resources Quota Management
* Virtualised Resources Performance Management
* Virtualised Resources Fault Management
* Policy Management
* Network Forwarding Path (NFP) Management (only VNF Orchestrator and VIM (Or-Vi))

### Tenant Level APIs

In the abstraction model of the Cloud Infrastructure (**Chapter 3**) a conceptual model of a Tenant (Figure 4) represents the slice of a cloud zone dedicated to a workload. This slice, the Tenant, is composed of virtual resources being utilized by workloads within that Tenant. The Tenant has an assigned quota of virtual resources, a set of users can perform operations as per their assigned roles, and the Tenant exists within a Cloud Zone. The APIs will specify the allowed operations on the Tenant including its component virtual resources and the different APIs can only be executed by users with the appropriate roles. For example, a Tenant may only be allowed to be created and deleted by Cloud Zone administrators while virtual compute resources could be allowed to be created and deleted by Tenant administrators.

For a workload to be created in a Tenant also requires APIs for the management (creation, deletion, and operation) of the Tenant, software flavours (Chapter 5), Operating System and workload images (“Images”), Identity and Authorization (“Identity”), virtual resources, security, and the workload application (“stack”).

A virtual compute resource is created as per the flavour template (specifies the compute, memory, and local storage capacity) and is launched using an image with access and security credentials; once launched, it is referred to as a virtual compute instance or just “Instance”). Instances can be launched by specifying the compute, memory, and local storage capacity parameters instead of an existing flavour; reference to flavours covers the situation where the capacity parameters are specified. IP addresses and storage volumes can be attached to a running Instance.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Resource | Create | List | Attach | Detach | Delete | Notes |
| Flavour | + | + |  |  | + |  |
| Image | + | + |  |  | + | Create/delete by appropriate administrators |
| Key pairs | + | + |  |  | + |  |
| Privileges |  |  |  |  |  | Created and managed by Cloud Service Provider(CSP) administrators |
| Role | + | + |  |  | + | Create/delete by authorized administrators where roles are assigned privileges and mapped to users in scope |
| Security Groups | + | + |  |  | + | Create and delete only by VDC administrators |
| Stack | + | + |  |  | + | Create/delete by VDC users with appropriate role |
| Virtual Storage | + | + | + | + | + | Create/delete by VDC users with appropriate role |
| User | + | + |  | + | + | Create/delete only by VDC administrators |
| Tenant | + | + |  | + | + | Create/delete only by Cloud Zone administrators |
| Virtual compute | + | + |  | + | + | Create/delete by VDC users with appropriate role. Additional operations would include suspend/unsuspend |
| Virtual network | + | + | + | + | + | Create/delete by VDC users with appropriate role |

1. : API types for a minimal set of resources.

Table 43 specifies a minimal set of operations for a minimal set of resources that are needed to orchestrate workloads. The actual APIs for the listed operations will be specified in the Reference Architectures; each listed operation could have a number of associated APIs with a different set of parameters. For example, create virtual resource using an image or a device.

### Hardware Acceleration Interfaces

**Acceleration Interface Specifications** ETSI GS NFV-IFA 002 defines a technology and implementation independent virtual accelerator, the accelerator interface requirements and specifications that would allow a workload to leverage a Virtual Accelerator. The virtual accelerator is modeled on extensible para-virtualised devices (EDP). ETSI GS NFV-IFA 002 specifies the architectural model in Chapter 4 and the abstract interfaces for management, configuration, monitoring, and Data exchange in Chapter 7.

ETSI ETSI (Ref: NFV IFA 019 v03101p) has defined a set of technology independent interfaces for acceleration resource life cycle management. These operations allow: allocation, release, and querying of acceleration resource, get and reset statistics, subscribe/unsubscribe (terminate) to fault notifications, notify (only used by NFVI), and get alarm information.

These acceleration interfaces are summarized here in Table 6.3 only for convenience.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Request | Response | From, To | Type | Parameter | Description |
| InitAccRequest | InitAccResponse | VNF → NFVI | Input | accFilter | the accelartor sub-system(s) to initialize and retrieve their capabilities. |
| Filter | accAttributeSelector | attribute names of accelerator capabilities |
| Output | accCapabilities | acceleration sub-system capabilities |
| RegisterForAccEventRequest | RegisterForAccEventResponse | VNF → NFVI | Input | accEvent | event the VNF is interested in |
| Input | vnfEventHandlerId | the handler for NFVI to use when notifying the VNF of the event |
| AccEventNotificationRequest | AccEventNotificationResponse | NFVI → VNF | Input | vnfEventHandlerId | Handler used by VNF registering |
| for this event |  |  |  |  |  |
|  |  |  | Input | accEventMetaData |  |
| DeRegisterForAccEventRequest | DeRegisterForAccEventResponse | VNF → NFVI | Input | accEvent | Event VNF is deregistering from |
| ReleaseAccRequest | ReleaseAccResponse | VNF → NFVI |  |  |  |
| ModifyAccConfigurationRequest | ModifyAccConfigurationResponse | VNF → NFVI | Input | accConfigurationData | Config data for accelerator |
|  |  |  | Input | accSubSysConfigurationData | Config data for accelerator sub-system |
| GetAccConfigsRequest | GetAccConfigsResponse | VNF → NFVI | Input | accFilter | Filter for subsystems from which config data requested |
|  |  |  | Input | accConfigSelector | attributes of config types |
|  |  |  | Output | accComfigs | Config info (only for the specified attributes) for specified subsystems |
| ResetAccConfigsRequest | ResetAccConfigsResponse | VNF → NFVI | Input | accFilter | Filter for subsystems for which config is to be reset |
|  |  |  | Input | accConfigSelector | attributes of config types whose values will be reset |
| AccDataRequest | AccDataResponse | VNF → NFVI | Input | accData | Data (metadata) sent too accelerator |
|  |  |  | Input | accChannel | Channel data is to be sent to |
|  |  |  | Output | accData | Data from accelerator |
| AccSendDataRequest | AccSendDataResponse | VNF → NFVI | Input | accData | Data (metadata) sent too accelerator |
|  |  |  | Input | accChannel | Channel data is to be sent to |
| AccReceiveDataRequest | AccReceiveDataResponse | VNF → NFVI | Input | maxNumberOfDataItems | Max number of data items to be received |
|  |  |  | Input | accChannel | Channel data is requested from |
|  |  |  | Output | accData | Data received form Accelerator |
| RegisterForAccDataAvailableEventRequest | RegisterForAccDataAvailableEventResponse | VNF → NFVI | Input | regHandlerId | Registration Identifier |
|  |  |  | Input | accChannel | Channel where event is requested for |
| AccDataAvailableEventNotificationRequest | AccDataAvailableEventNotificationResponse | NFVI → VNF | Input | regHandlerId | Reference used by VNF when registering for the event |
| DeRegisterForAccDataAvailableEventRequest | DeRegisterForAccDataAvailableEventResponse | VNF → NFVI | Input | accChannel | Channel related to the event |
| AllocateAccResourceRequest | AllocateAccResourceResponse | VIM → NFVI | Input | attachTargetInfo | the resource the |
| accelerator is to be attached to (e.g., VM) |  |  |  |  |  |
|  |  |  | Input | accResourceInfo | Accelerator Information |
|  |  |  | Output | accResourceId | Id if successful |
| ReleaseAccResourceRequest | ReleaseAccResourceResponse | VIM → NFVI | Input | accResourceId | Id of resource to be released |
| QueryAccResourceRequest | QueryAccResourceResponse | VIM → NFVI | Input | hostId | Id of specified host |
|  |  |  | Input | Filter | Specifies the accelerators for which query applies |
|  |  |  | Output | accQueryResult | Details of the accelerators matching the input filter located in the selected host. |
| GetAccStatisticsRequest | GetAccStatisticsResponse | VIM → NFVI | Input | accFilter | Accelerator subsystems from which data is requested |
|  |  |  | Input | accStatSelector | attributes of AccStatistics whose data will be returned |
|  |  |  | Output | accStatistics | Statistics data of the accelerators |
| matching the input filter located in the |  |  |  |  |  |
| selected host. |  |  |  |  |  |
| ResetAccStatisticsRequest | ResetAccStatisticsResponse | VIM → NFVI | Input | accFilter | Accelerator subsystems for which data is to be reset |
|  |  |  | Input | accStatSelector | attributes of AccStatistics whose data will be reset |
| SubscribeRequest | SubscribeResponse | VIM → NFVI | Input | hostId | Id of specified host |
|  |  |  | Input | Filter | Specifies the accelerators and related alarmsThe filter could include accelerator information, severity of the alarm, etc. |
|  |  |  | Output | SubscriptionId | Identifier of the successfully created |
| subscription. |  |  |  |  |  |
| UnsubscribeRequest | UnsubscribeResponse | VIM → NFVI | Input | hostId | Id of specified host |
|  |  |  | Input | SubscriptionId | Identifier of the subscription to be |
| unsubscribed. |  |  |  |  |  |
| Notify |  | NFVI → VIM |  |  | NFVI notifies an alarm to VIM |
| GetAlarmInfoRequest | GetAlarmInfoResponse | VIM → NFVI | Input | hostId | Id of specified host |
|  |  |  | Input | Filter | Specifies the accelerators and related alarmsThe filter could include accelerator information, severity of the alarm, etc. |
|  |  |  | Output | Alarm | Information about the alarms if filter matches an alarm. |
|  |  |  |  |  |  |
| AccResourcesDiscoveryRequest | AccResourcesDiscoveryResponse | VIM → NFVI | Input | hostId | Id of specified host |
|  |  |  | Output | discoveredAccResourceInfo | Information on the acceleration resources discovered within the NFVI. |
| OnloadAccImageRequest | OnloadAccImageResponse | VIM → NFVI | Input | accResourceId | Identifier of the chosen accelerator in the NFVI. |
|  |  |  | Input | accImageInfo | Information about the acceleration image. |
|  |  |  | Input | accImage | The binary file of acceleration image. |

1. : Hardware Acceleration Interfaces in the ETSI NFV architecture

## Intra-Cloud Infrastructure Interfaces

### Hypervisor Hardware Interface

Table 42 lists a number of NFVI and VIM interfaces, including the internal VI-Ha interface. The VI-Ha interface allows the hypervisor to control the physical infrastructure; the hypervisor acts under VIM control. The VIM issues all requests and responses using the NF-VI interface; requests and responses include commands, configuration requests, policies, updates, alerts, and response to infrastructure results. The hypervisor also provides information about the health of the physical infrastructure resources to the VM. All these activities, on behalf of the VIM, are performed by the hypervisor using the VI-Ha interface. While no abstract APIs have yet been defined for this internal VI-Ha interface, ETSI GS NFV-INF 004 defines a set of requirements and details of the information that is required by the VIM from the physical infrastructure resources. Hypervisors utilize various programs to get this data including BIOS, IPMI, PCI, I/O Adapters/Drivers, etc.

## Enabler Services Interfaces

An operational cloud needs a set of standard services to function. Services such as NTP for time synchronization, DHCP for IP address allocation, DNS for obtaining IP addresses for domain names, and LBaaS (version 2) to distribute incoming requests amongst a pool of designated resources.

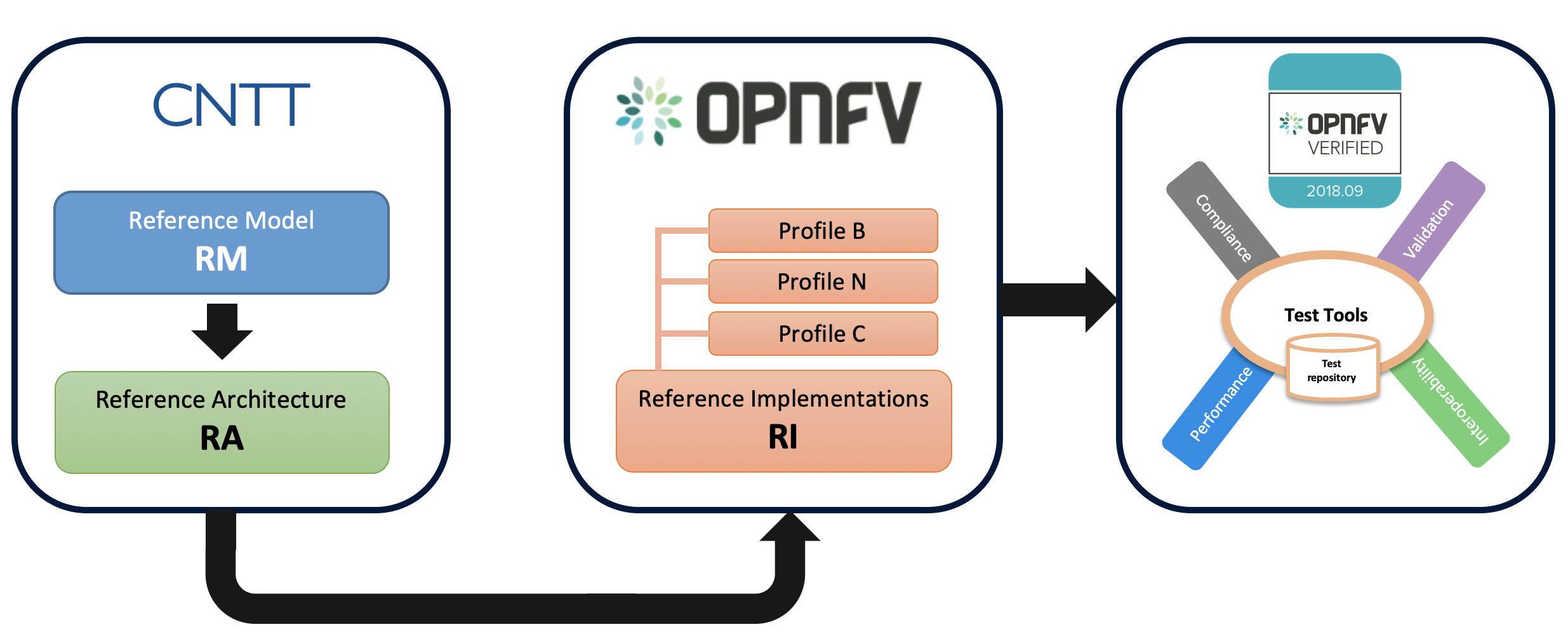
## References

Network Functions Virtualisation (NFV); Infrastructure; Hypervisor Domain. ETSI GS NFV-INF 004 Network Functions Virtualisation (NFV); Acceleration Technologies; VNF Interfaces SpecificationETSI. GS NFV-IFA 002 v2.4.1 Network Functions Virtualisation (NFV); Acceleration Technologies; Acceleration Resource Management Interface Specification; NFV IFA 019 v03101p Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification; ETSI GS NFV-IFA 005 V3.2.1

# Security

## Introduction

This document includes process flow, logistics, and requirements which must be satisfied to ensure workloads meet the design, feature, and capability expectations of workload consumers to deliver cloud promoting the use and scalability of SDN capabilities. This chapter captures the core fundamentals and steps needed to conformance test workloads on target Cloud Infrastructure frameworks and architectures which drives more work into the community, resulting in pre-certified workloads on core capabilities ultimately reducing the amount of time and cost it takes each operator to on-board and maintain vendor provided workloads.



1. : CNTT relation to LFN OVP

## Principles and Guidelines

The objectives of the Security verification program are to deliver certified reference architectures which match application-developer specifications, leveraging the OVP ecosystem as the vehicle for deliverying security validated Cloud Infrastructure.

These core principles will guide Cloud Infrastructure verification deliverables

### Overarching Objectives and Goals

1. Deliver security certified reference architecture which matches application-developer specifications
2. All accomplished with augmentation to the current OVP ecosystem.
3. Certified applications will on-board and function first shot

### Verification Methodologies

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

### Governance

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

## Common standards

Security vulnerabilities and attack vectors are everywhere. The telecom industry and its cloud infrastructures are even more vulnerable to potential attacks due to the ubiquitous nature of the infrastructures and services combined with the vital role Telecommunications play in the modern world. The attack vectors are many and varied, ranging from the potential for exposure of sensitive data, both personal and corporate, to weaponized disruption to the global Telecommunications networks. The threats can take the form of a physical attack on the locations the infrastructure hardware is housed, to network attacks such as denial of service and targeted corruption of the network service applications themselves. Whatever the source, any Cloud Infrastructure built needs to be able to withstand attacks in whatever form they take.

With that in mind, the Cloud Infrastructure reference model and the supporting architectures are not only required to optimally support networking functions, but they must be designed with common security principles and standards from inception. These best practices must be applied at all layers of the infrastructure stack and across all points of interconnections with outside networks, APIs and contact points with the NFV network functions overlaying or interacting with that infrastructure. Standards organizations with recommendations and best practices, and certifications that need to be taken into consideration include the following examples. However this is by no means an exhaustive list, just some of the more important standards in current use.

* Center for Internet Security - <https://www.cisecurity.org/>
* Cloud Security Alliance - <https://cloudsecurityalliance.org/>
* Open Web Application Security Project <https://www.owasp.org>
* The National Institute of Standards and Technology (NIST) (US Only)
* FedRAMP Certification <https://www.fedramp.gov/> (US Only)
* ETSI Cyber Security Technical Committee (TC CYBER) - <https://www.etsi.org/committee/cyber>
* ETSI Industry Specification Group Network Functions Virtualisation (ISG NFV) - <https://www.etsi.org/technologies/nfv>
  + ETSI NFV ISG [SEC WG specifications](https://www.etsi.org/standards-search#page=1&search=NFV-SEC&title=0&etsiNumber=1&content=0&version=1&onApproval=0&published=1&historical=0&startDate=1988-01-15&endDate=2020-02-27&harmonized=0&keyword=&TB=&stdType=&frequency=&mandate=&collection=&sort=1)
* ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) - [www.iso.org](http://www.iso.org). The following ISO standards are of particular interest for NFVI
  + ISO/IEC 27002:2013 - ISO/IEC 27001 is the international Standard for best-practice information security management systems (ISMSs).
  + ISO/IEC 27032 - ISO/IEC 27032is the international Standard focusing explicitly on cybersecurity.
  + ISO/IEC 27035 - ISO/IEC 27035 is the international Standard for incident management. Incident management
  + ISO/IEC 27031 - ISO/IEC 27031 is the international Standard for ICT readiness for business continuity.

A good place to start to understand the requirements is to use the widely accepted definitions developed by the OWASP – Open Web Application Security Project. These include the following core principles:

* Confidentiality – Only allow access to data for which the user is permitted
* Integrity – Ensure data is not tampered with or altered by unauthorized users
* Availability – ensure systems and data are available to authorized users when they need it

Additional Cloud Infrastructure security principles that need to be incorporated:

* Authenticity – The ability to confirm the users are in fact valid users with the correct rights to access the systems or data.

### Potential attack vectors

Previously attacks designed to place and migrate workload outside the legal boundaries were not possible using traditional infrastructure, due to the closed nature of these systems. However, using Cloud Infrastructure, violation of regulatory policies and laws becomes possible by actors diverting or moving an application from an authenticated and legal location to another potentially illegal location. The consequences of violating regulatory policies may take the form of a complete banning of service and/or an exertion of a financial penalty by a governmental agency or through SLA enforcement. Such vectors of attack may well be the original intention of the attacker in an effort to harm the service provider. One possible attack scenario can be when an attacker exploits the insecure VNF API to dump the records of personal data from the database in an attempt to violate user privacy. Cloud Infrastructure operators should ensure that the applications APIs are secure, accessible over a secure network (TLS) under very strict set of security best practices, and RBAC policies to limit exposure of this vulnerability.

### Testing demarcation points

It is not enough to just secure all potential points of entry and hope for the best, any Cloud Infrastructure architecture must be able to be tested and validated that it is in fact protected from attack as much as possible. The ability to test the infrastructure for vulnerabilities on a continuous basis is critical for maintaining the highest level of security possible. Testing needs to be done both from the inside and outside of the systems and networks. Below is a small sample of some of the testing methodologies and frameworks available.

* + OWASP testing guide
  + PCI Penetration testing guide
  + Penetration Testing Execution Standard
  + NIST 800-115
  + *VULCAN: Vulnerability Assessment Framework for Cloud Computing (NIST)*
  + Penetration Testing Framework
  + Information Systems Security Assessment Framework (ISSAF)
  + Open Source Security Testing Methodology Manual (“OSSTMM”)
  + FedRAMP Penetration Test Guidance (US Only)
  + CREST Penetration Testing Guide

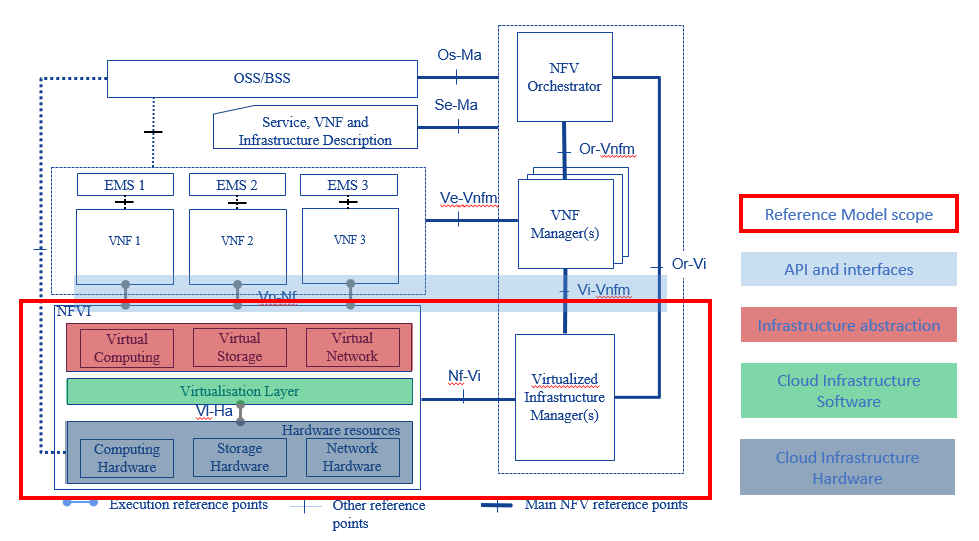
Insuring that the security standards and best practices are incorporated into the Cloud Infrastructure and architectures must be a shared responsibility, among the Telecommunications operators interested in building and maintaining the infrastructures in support of their services, the application vendors developing the network services that will be consumed by the operators, and the Cloud Infrastructure vendors creating the infrastructures for their Telecommunications customers. All of the parties need to incorporate security and testing components, and maintain operational processes and procedures to address any security threats or incidents in an appropriate manner. Each of the stakeholders need to contribute their part to create effective security for the Cloud Infrastructure.

## Security Scope

### In-scope and Out-of-Scope definition

The scope of the security controls requirements maps to the scope of the Reference Model architecture.

The Reference Model scope is shown below (as outlined in chapter 1 of the reference model):



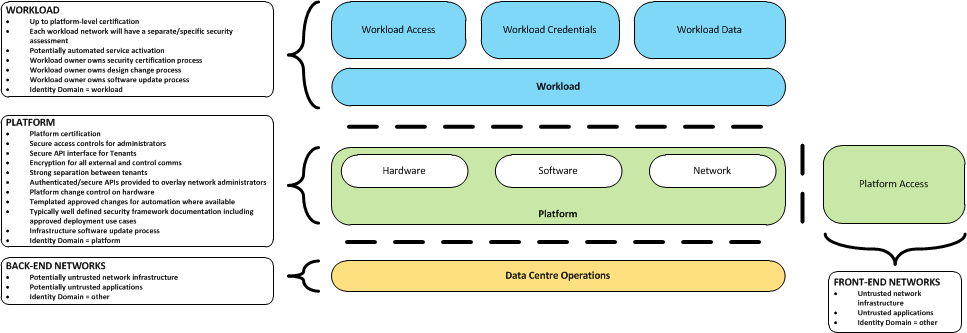
1. : ETSI NFV architecture mapping

This means that the security of the Reference Model requirements must cover the virtual resources (including the virtualisation layer), the hardware resources, and the VIM (Virtualised Infrastructure Manager).

There will be a different set of security requirements for each Cloud Infrastructure reference architecture. In this case, the first reference architecture is OpenStack.

### Security Requirements

The following diagram shows the different security domains that impact the Reference Model:



1. : Reference Model Security Domains

### Platform security requirements

At a high level, the following areas/requirements cover platform security for a particular deployment:

* Platform certification
* Secure access controls for administrators
* Secure API interface for Tenants
* Encryption for all external and control comms
* Strong separation between tenants - ensuring network, data, and runtime process isolation between tenants
* Authenticated/secure APIs provided to overlay network administrators
* Platform change control on hardware
* Templated approved changes for automation where available
* Typically well defined security framework documentation including approved deployment use cases
* Infrastructure software update process
* Identity Domain = platform

### Workload security requirements

At a high level, the following areas/requirements cover workload security for a particular deployment:

* Up to platform-level certification
* Each workload network will need to undertake it own security self-assessment and accreditation, and not inherit a security accreditation from the platform
* Potentially automated service activation
* Workload owner owns workload security certification process
* Workload owner owns workload design change process
* Workload owner owns workload software update process
* Identity Domain = workload

### Certification/validation requirements

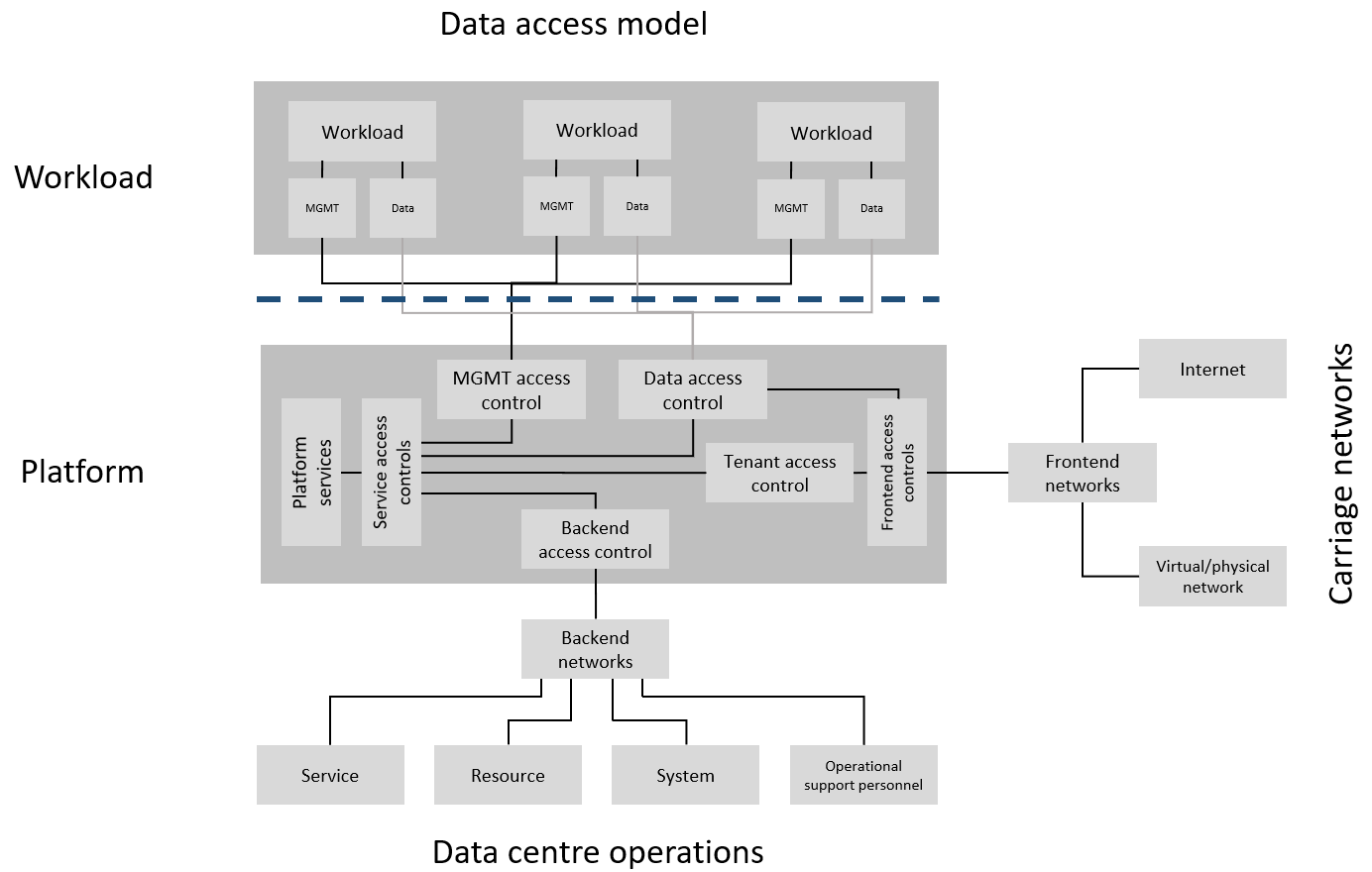
\*(An overview/introduction to workload certification requirements and  
incl types of workloads covered)\*

## Platform Security

### General Platform Security

The security certification of the platform will typically need to be the same, or higher, than the workload or VNF requirements.

The platform supports the workload, and in effect controls access to the workload from and to external endpoints such as carriage networks used by workloads, or by Data Centre Operations staff supporting the workload, or by tenants accessing workloads. From an access security perspective, the following diagram shows where different access controls will operate within the platform to provide access controls throughout the platform:



1. : Reference Model Access Controls

#### The high-level functions of these different access controls

* **MGNT ACCESS CONTROLS** - Platform access to workloads for service management. Typically all management and control-plane traffic is encrypted.
* **DATA ACCESS CONTROLS** - Control of east-west traffic between workloads, and control of north-south traffic between the VNF and other platform services such as front-end carriage networks and platform services. Inherently strong separation between tenants is mandatory.
* **SERVICES ACCESS CONTROLS** - Protects platform services from any platform access
* **BACK-END ACCESS CONTROLS** - Data Centre Operations access to the platform, and subsequently, workloads. Typically stronger authentication requirements such as (Two-Factor Authentication) 2FA, and using technologies such as Role-Based Access Control (RBAC) and encryption. Application Programming Interface (API) gateways may be required for automated/script-driven processes.
* **FRONT-END ACCESS CONTROLS** - Protects the platform from malicious carriage network access, and provides connectivity for specific workloads to specific carriage networks. Carriage networks being those that are provided as public networks and operated by carriers, and in this case with interfaces that are usually sub, or virtual networks.
* **TENANT ACCESS CONTROLS** - Provides apropriate tenant access controls to specific platform services, and tenant workloads - including Role-Based Access Control (RBAC), authentication controls as approriate for the access arrangement, and Application Programming Interface (API) gateways for automated/script-driven processes.

#### The following general security requirements apply to the platform

* Restrict traffic to (and from) the workload to only traffic that is necessary, and deny all other traffic.
* Provide protections between the Internet and any workloads including web and volumetrics attack preventions.
* Support zoning within a tenant workload - using application-level filtering.
* All host to host communications within the cloud provider network are to be cryptographically protected in transit.
* Not expose tenant IP address details to another tenant.
* Use cryptographically-protected protocols for administrative access to the platform.
* Data Centre Operations staff and systems must use management protocols that limit security risk such as SNMPv3, SSH v2, ICMP, NTP, syslog, and TLS.
* A platform change management process that is documented, well communicated to staff and tenants, and rigourously followed.
* A process to check change management adherence that is implemented, and rigourously followed.
* Processes for managing platform access control filters that are documented, followed, and monitored.
* No login to root on any platform systems (platform systems are those that are associated with the platform and include systems that directly or indirectly affect the viability of the platform).
* Role-Based Access Control (RBAC) must apply for all platform systems access.
* An approved system or process for last resort access must exist for the platform.
* All API access must use TLS protocol.
* All production workloads must be separated from all non-production workloads including separation between non-hosted non-production external networks.
* Where there are multiple hosting facilities used in provision of the service, network communications between facilities for the purpose of backup, management, and application communication are cryptographically protected in transit between data centre facilities.
* Continuous cloud security compliance is mandatory.
* All data persisted to primary, replica, or backup storage is to be encrypted.
* All platform security logs are to be time synchronised.
* Logs are to be regularly scanned for events of interest.
* An incident response plan must exist for the platform.
* The cloud services must be regularly vulnerability and penetration tested.

### Platform ‘back-end’ access security

* Restrict traffic to only traffic that is necessary, and deny all other traffic.
* Use cryptographically-protected protocols for administrative access to the platform.
* Data Centre Operations staff and systems must use management protocols that limit security risk such as SNMPv3, SSH v2, ICMP, NTP, syslog, and TLS.
* A platform change management process that is documented, well communicated to staff and tenants, and rigourously followed.
* A process to check change management adherence that is implemented, and rigourously followed.
* Processes for managing platform access control filters that are documented, followed, and monitored.
* No login to root on any platform systems.
* Role-Based Access Control (RBAC) must apply for all systems access.
* An approved system or process for last resort access must exist for the platform.
* All back-end API access must use TLS.

### Platform ‘front-end’ access security

* Front-end network security at the application level will be the responsibility of the workload, however the platform must ensure the isolation and integrity of tenant connectivity to front-end networks.
* The front-end network may provide (Distributed Denial Of Service) DDOS support.

## Workload Security - Vendor Responsibility

### Software Hardening

* No hard-coded credentials/ clear text passwords
* Software should be independent of the infrastructure platform (no OS point release dependencies to patch)
* Software is code signed and all individual sub-components are assessed and verified for EULA violations
* Software should have a process for discovery, classification, communication, and timely resolution of security vulnerabilities (i.e.; bug bounty, Penetration testing/scan findings, etc)

### Port Protection

* Unused software and unused network ports should be disabled by default

### Software Code Quality

* Vendors should use industry recognized software testing suites
  + Static and dynamic scanning
  + Automated static code review with remediation of Medium/High/Critical security issues. The tool used for static code analysis and analysis of code being released must be shared.
  + Dynamic security tests with remediation of Medium/High/Critical security issues. The tool used for Dynamic security analysis of code being released must be shared
  + Penetration tests (pen tests) with remediation of Medium/High/Critical security issues.
  + Methodology for ensuring security is included in the Agile/DevOps delivery lifecycle for ongoing feature enhancement/maintenance.

### Alerting and monitoring

* Security event logging (All security events should be logged, including informational)
* Privilege escalation detection

### Logging

* (Logging output should support customizable Log retention and Log rotation)

### VNF images

* Image integrity – fingerprinting/validation
* Container Images
  + Container Management
  + Immutability

### Identity and Access Management

### CVEs and Vulnerability Management

* Security defect reporting
* Cadence with Cloud Infrastructure vendors (OSSA for OpenStack)

### Encryption suite support

* Software should support recognized encryption standards and encryption should be decoupled from software

### Password complexity support

* Software should support configurable, or industry standard, password complexity rules

### Banner

* Software should have support for configurable banners to display authorized use criteria/policy

## Workload Security - Operator Responsibility.

The Operator’s responsibility is to not only make sure that security is included in all the vendor supplied infrastructure and NFV components, but it is also responsible for the maintenance of the security functions from an operational and management perspective. This includes but is not limited to securing the following elements:

* Maintaining standard security operational management methods and processes
* Monitoring and reporting functions
* Processes to address regulatory compliance failure
* Support for appropriate incident response and reporting
* Methods to support appropriate remote attestation certification of the validity of the security components, architectures, and methodologies used

### Remote Attestation/openCIT

Cloud Infrastructure operators must ensure that remote attestation methods are used to remotely verify the trust status of a given Cloud Infrastructure platform. The basic concept is based on boot integrity measurements leveraging the TPM built into the underlying hardware. Remote attestation can be provided as a service, and may be used by either the platform owner or a consumer/customer to verify that the platform has booted in a trusted manner. Practical implementations of the remote attestation service include the open cloud integrity tool (Open CIT). Open CIT provides ‘Trust’ visibility of the cloud infrastructure and enables compliance in cloud datacenters by establishing the root of trust and builds the chain of trust across hardware, operating system, hypervisor, VM, and container. It includes asset tagging for location and boundary control. The platform trust and asset tag attestation information is used by Orchestrators and/or Policy Compliance management to ensure workloads are launched on trusted and location/boundary compliant platforms. They provide the needed visibility and auditability of infrastructure in both public and private cloud environments.

Insert diagram here: <https://01.org/sites/default/files/users/u26957/32_architecture.png>

### Workload Image Scanning / Signing

It is easy to tamper with workload images. It requires only a few seconds to insert some malware into a workload image file while it is being uploaded to an image database or being transferred from an image database to a compute node. To guard against this possibility, workload images can be cryptographically signed and verified during launch time. This can be achieved by setting up a signing authority and modifying the hypervisor configuration to verify an image’s signature before they are launched. To implement image security, the VNF operator must test the image and supplementary components verifying that everything conforms to security policies and best practices.

## Application Vendors responsibility

The application vendors need to incorporate security elements to support the highest level of security of the networks they support. This includes but is not limited to securing the following elements:

* Operating system or container
* Application
* Network interfaces
* Management and controller systems used to support the VNFs directly, examples include a SIEM system or a SD WAN policy manager
* Regulatory compliance failure as it relates to the application itself only

Image from <https://www.networkworld.com/article/2840273/sdn-security-attack-vectors-and-sdn-hardening.html> Will replace with a better image when I create it in the future.

## Cloud Infrastructure and Cloud Infrastructure Manager Vendors responsibility

The Cloud Infrastructure vendors need to incorporate security elements to support the highest level of security of the infrastructure they support. This includes but is not limited to securing the following elements:

* Hypervisor
* VM/container management system
* APIs
* Network interfaces
* Networking security zoning
* Platform patching mechanisms
* Regulatory compliance Failure

### Networking Security Zoning

Network segmentation is important to ensure that VMs can only communicate with the VMs they are supposed to. To prevent a VM from impacting other VMs or hosts, it is a good practice to separate VM traffic and management traffic. This will prevent attacks by VMs breaking into the management infrastructure. It is also best to separate the VLAN traffic into appropriate groups and disable all other VLANs that are not in use. Likewise, VMs of similar functionalities can be grouped into specific zones and their traffic isolated. Each zone can be protected using access control policies and a dedicated firewall based on the needed security level.

Recommended practice to set network security policies following the principle of least privileged, only allowing approved protocol flows. For example, set 'default deny' inbound and add approved policies required for the functionality of the application running on the NFVI infrastructure.

### Encryption

Virtual volume disks associated with workloads may contain sensitive data. Therefore, they need to be protected. Best practice is to secure the workload volumes by encrypting them and storing the cryptographic keys at safe locations. Be aware that the decision to encrypt the volumes might cause reduced performance, so the decision to encrypt needs to be dependent on the requirements of the given infrastructure. The TPM module can also be used to securely store these keys. In addition, the hypervisor should be configured to securely erase the virtual volume disks in the event of application crashes or is intentionally destroyed to prevent it from unauthorized access.

* Composition analysis: New vulnerabilities are discovered in common open source libraries every week. As such, mechanisms to validate components of the VNF application stack by checking libraries and supporting code against the Common Vulnerabilities and Exposures (CVE) databases to determine whether the code contains any known vulnerabilities must be embedded into the NFVI architecture itself. Some of the components required include:
* Tools for checking common libraries against CVE databases integrated into the deployment and orchestration pipelines.
* The use of Image scanners such as OpenSCAP to determine security vulnerabilities

### Platform Patching

Cloud Infrastructure operators should ensure that the platform including the components (hypervisors, VMs, etc.) are kept up to date with the latest patch.

### Boot Integrity Measurement (TPM)

Using a trusted platform module (TPM) as a hardware root of trust, the measurement of system sensitive components, such as platform firmware, bootloader, OS kernel, static filesystem, and other system components can be securely stored and verified. NFVI Operators should ensure that the TPM support is enabled in the platform firmware, so that platform measurements are correctly recorded during boot time.

Additionally, NFVI Operators should ensure that OS kernel measurements can be recorded by using a TPM-aware bootloader (e.g. [tboot](https://sourceforge.net/projects/tboot/) or [shim](https://github.com/rhboot/shim)), which can extend the root of trust up to the kernel level. The validation of the platform measurements can be performed by TPM’s launch control policy (LCP) or through the remote attestation server.

### Runtime Integrity Measurement (TPM)

If a remote attestation server is used to monitor platform integrity, the operators should ensure that attestation is performed periodically or in a timely manner. Additionally, platform measurements may be extended to monitor the integrity of the static filesystem at run-time by using a TPM aware kernel module, such as [Linux IMA (Integrity Measurement Architecture)](https://sourceforge.net/p/linux-ima/wiki/Home/) for linux platforms, or by using the [trust policies](https://github.com/opencit/opencit/wiki/Open-CIT-3.2-Product-Guide#88-trust-policies) functionality of OpenCIT. The static filesystem includes a set of important files and folders which do not change between reboots during the lifecycle of the platform. This allows the attestation server to detect any tampering with the static filesystem during the runtime of the platform.

### Cloud Infrastructure & Cloud Infrastructure Manager

Resources management is essential. Requests coming from a higher orchestration layer to the Cloud Infrastructure manager must validated and the integrity of these requets must be verified.

#### Internal security capabilities

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | NFVI capability | Unit | Definition/Notes |
| i.sec.cap.001 | VNF-C<->VNF-C memory isolation | Yes/No | Are VNF-C memories isolated from each other by hardware support |
| i.sec.cap.002 | VNF-C -> Host | Yes/No | Can VNF-C access host memory |
| i.sec.cap.003 | Host -> VNF-C | Yes/No | Can Host access VNF-C memory |
| i.sec.cap.004 | External storage at-rest encryption | Yes/No | Is external storage encrypted at-rest |

1. : Cloud Infrastructure internal security capabilities

Table 46 shows security capabilities

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | VIM capability | Unit | Definition/Notes |
| e.sec.cap.001 | Resources management requests verification | Yes/No | Capability to validate and verify the integrity of a resources management requests coming from NFVO or VNFM |

1. : Cloud Infrastructure Manager capabilities related to security

## Certification requirements (Just ideas)

* Security test cases executed and test case results
* Industry standard compliance achieved (NIST, ISO, PCI, FedRAMP Moderate etc.)
* Output and analysis from automated static code review, dynamic tests, and penetration tests with remediation of Medium/High/Critical security issues. Tools used for security testing of software being released must be shared.
* Details on un-remediated low severity security issues must be shared.
* Threat models performed during design phase. Including remediation summary to mitigate threats identified.
* Details on un-remediated low severity security issues.
* Any additional Security and Privacy requirements implemented in the software deliverable beyond the default rules used security analysis tools
* Resiliency tests run (such as hardware failures or power failure tests)

## Consolidated Security Requirements

### System Hardening

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Requirement | Unit | Definition |
| sec.gen.001 | The Platform **must** maintain the state to what it is specified to be and does not change unless through change management process. |  |  |
| sec.gen.002 | All systems part of Cloud Infrastructure **must** support password hardening (strength and rules for updates (process), storage and transmission, etc.) |  | Hardening: NIST SP 800-63B |
| sec.gen.003 | All servers part of Cloud Infrastructure **must** support a root of trust and secure boot |  |  |
| sec.gen.004 | The Operating Systems of all the servers part of Cloud Infrastructure **must** be hardened |  | NIST SP 800-123 |
| sec.gen.005 | The Platform **must** support Operating System level access control |  | Details on OS |
| sec.gen.006 | The Platform **must** support Secure logging |  | Details |
| sec.gen.007 | All servers part of Cloud Infrastructure **must** be Time synchronized with authenticated Time service |  |  |
| sec.gen.008 | All servers part of Cloud Infrastructure **must** be regularly updated to address security vulnerabilities |  |  |
| sec.gen.009 | The Platform **must** support Software integrity protection and verification |  |  |
| sec.gen.010 | The Cloud Infrastructure **must** support Secure storage (all types) |  | Expand/Delete based on other requirements |
| sec.gen.011 | The Cloud Infrastructure **should** support Read and Write only storage partitions (write only permission to one or more authorized actors) |  |  |
| sec.gen.012 | The Operator **must** ensure that only authorized actors have physical access to the underlying infrastructure. |  |  |
| sec.gen.013 | The Platform **must** ensure that only authorized actors have logical access to the underlying infrastructure. |  |  |
| sec.gen.014 | All servers part of Cloud Infrastructure **should** support measured boot and an attestation server that monitors the measurements of the servers. |  |  |

### Platform and Access

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Requirement | Unit | Definition |
| sec.sys.001 | The Platform **must** support authenticated and secure APIs, API endpoints |  |  |
|  | The Platform **must** implement authenticated and secure access to GUI |  |  |
| sec.sys.002 | The Platform **must** support Traffic Filtering for workloads (for example, Fire Wall) |  |  |
| sec.sys.003 | The Platform **must** support Secure and encrypted communications, and confidentiality and integrity of network traffic |  |  |
| sec.sys.004 | The Cloud Infrastructure **must** support Secure network channels |  | A secure channel enables transferring of data that is resistant to overhearing and tampering |
| sec.sys.005 | The Cloud Infrastructure **must** segregate the underlay and overlay networks |  |  |
| sec.sys.006 | The Cloud Infrastructure must be able to utilize the Cloud Infrastructure Manager identity management capabilities |  |  |
| sec.sys.007 | The Platform **must** implement controls enforcing separation of duties and privileges, least privilege use and least common mechanism (Role-Based Access Control) |  |  |
| sec.sys.008 | The Platform **must** be able to assign the Entities that comprise the tenant networks to different trust domains. |  | Communication between different trust domains is not allowed, by default. |
| sec.sys.009 | The Platform **must** support creation of Trust Relationships between trust domains |  | These maybe uni-directional relationships where the trusting domain trusts anther domain (the “trusted domain”) to authenticate users for them or to allow access to its resources from the trusted domain. In a bidirectional relationship both domain are “trusting” and “trusted” |
| sec.sys.010 | For two or more domains without existing trust relationships, the Platform **must not** allow the effect of an attack on one domain to impact the other domains either directly or indirectly |  |  |
| sec.sys.011 | The Platform **must not** reuse the same authentication key-pair (for example, on different hosts, for different services) |  |  |
| sec.sys.012 | The Platform **must** only use secrets encrypted using strong encryption techniques, and stored externally from the component |  | e.g., Barbican (OpenStack) |
| sec.sys.013 | The Platform **must** provide secrets dynamically as and when needed |  |  |

### Confidentiality and Integrity

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Requirement | Unit | Definition |
| sec.ci.001 | The Platform **must** support Confidentiality and Integrity of data at rest and in-transit |  |  |
| sec.ci.002 | The Platform **should** support self-encrypting storage devices |  |  |
| sec.ci.003 | The Platform **must** support Confidentiality and Integrity of data related metadata |  |  |
| sec.ci.004 | The Platform **must** support Confidentiality of processes and restrict information sharing with only the process owner (e.g., tenant). |  |  |
| sec.ci.005 | The Platform **must** support Confidentiality and Integrity of process-related metadata and restrict information sharing with only the process owner (e.g., tenant). |  |  |
| sec.ci.006 | The Platform **must** support Confidentiality and Integrity of workload resource utilization (RAM, CPU, Storage, Network I/O, cache, hardware offload) and restrict information sharing with only the workload owner (e.g., tenant). |  |  |
| sec.ci.007 | The Platform **must not** allow Memory Inspection by any actor other than the authorized actors for the Entity to which Memory is assigned (e.g., tenants owning the workload), for Lawful Inspection, and by secure monitoring services |  | Admin access must be carefully regulated |
| sec.ci.008 | The Cloud Infrastructure **must** support tenant networks segregation |  |  |

### Workload Security

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Requirement | Unit | Definition |
| sec.wl.001 | The Platform **must** support Workload placement policy |  |  |
| sec.wl.002 | The Platform **must** support operational security |  |  |
| sec.wl.003 | The Platform **must** support secure provisioning of workloads |  |  |
| sec.wl.004 | The Platform **must** support Location assertion (for mandated in-country or location requirements) |  |  |
| sec.wl.005 | Production workloads **must** be separated from non-production workloads |  |  |
| sec.wl.006 | Workloads **must** be separable by their categorisation (for example, payment card information, healthcare, etc.) |  |  |
| sec.wl.007 | The Operator **should** implement processes and tools to verify VNF authenticity and integrity. |  |  |

### Image Security

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Requirement | Unit | Definition |
| sec.img.001 | Images from untrusted sources **must not** be used |  |  |
| sec.img.002 | Images **must** be maintained to be free from known vulnerabilities |  |  |
| sec.img.003 | Images **must not** be configured to run with privileges higher than the privileges of the actor authorized to run them |  |  |
| sec.img.004 | Images **must** only be accessible to authorized actors |  |  |
| sec.img.005 | Image Registries **must** only be accessible to authorized actors |  |  |
| sec.img.006 | Image Registries **must** only be accessible over secure networks |  |  |
| sec.img.007 | Image registries **must** be clear of vulnerable and stale (out of date) versions |  |  |

### Security LCM

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Requirement | Unit | Definition |
| sec.lcm.001 | The Platform **must** support Secure Provisioning, Maintaining availability, Deprovisioning (secure Clean-Up) of workload resources |  | Secure clean-up: tear-down, defending against virus or other attacks, or observing of cryptographic or user service data |
| sec.lcm.002 | Operational **must** use management protocols limiting security risk such as SNMPv3, SSH v2, ICMP, NTP, syslog and TLS |  |  |
| sec.lcm.003 | The Cloud Operator **must** implement change management for Cloud Infrastructure, Cloud Infrastructure Manager and other components of the cloud |  | Platform change control on hardware |
| sec.lcm.004 | The Cloud Operator **should** support automated templated approved changes |  | Templated approved changes for automation where available |
| sec.lcm.005 | Platform **must** provide logs and these logs must be regularly scanned |  |  |
| sec.lcm.006 | The Platform **must** verify the integrity of all Resource management requests | Yes/No |  |
| sec.lcm.007 | The Platform **must** be able to update newly instantiated, suspended, hibernated, migrated and restarted images with current time information |  |  |
| sec.lcm.008 | The Platform **must** be able to update newly instantiated, suspended, hibernated, migrated and restarted images with relevant DNS information. |  |  |
| sec.lcm.009 | The Platform **must** be able to update the tag of newly instantiated, suspended, hibernated, migrated and restarted images with relevant geolocation (geographical) information |  |  |
| sec.lcm.010 | The Platform **must** log all changes to geolocation along with the mechanisms and sources of location information (i.e. GPS, IP block, and timing). |  |  |
| sec.lcm.011 | The Platform **must** implement Security life cycle management processes including proactively update and patch all deployed Cloud Infrastructure software. |  |  |

### Monitoring and Security Audit

The Platform is assumed to provide configurable alerting and notification capability and the operator is assumed to have automated systems, policies and procedures to act on alerts and notifications in a timely fashion. In the following the monitoring and logging capabilities can trigger alerts and notifications for appropriate action.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Requirement | Unit | Definition |
| sec.mon.001 | Platform must provide logs and these logs must be regularly scanned for events of interest |  |  |
| sec.mon.002 | Security logs must be time synchronised |  |  |
| sec.mon.003 | The Platform must log all changes to time server source, time, date and time zones |  |  |
| sec.mon.004 | The Platform must secure and protect Audit logs (contain sensitive information) both in-transit and at rest |  |  |
| sec.mon.005 | The Platform must Monitor and Audit various behaviours of connection and login attempts to detect access attacks and potential access attempts and take corrective actions accordingly |  |  |
| sec.mon.006 | The Platform must Monitor and Audit operations by authorized account access after login to detect malicious operational activity and take corrective actions accordingly |  |  |
| sec.mon.007 | The Platform must Monitor and Audit security parameter configurations for compliance with defined security policies |  |  |
| sec.mon.008 | The Platform must Monitor and Audit externally exposed interfaces for illegal access (attacks) and take corrective security hardening measures |  |  |
| sec.mon.009 | The Platform must Monitor and Audit service handling for various attacks (malformed messages, signalling flooding and replaying, etc.) and take corrective actions accordingly |  |  |
| sec.mon.010 | The Platform must Monitor and Audit running processes to detect unexpected or unauthorized processes and take corrective actions accordingly |  |  |
| sec.mon.011 | The Platform must Monitor and Audit logs from infrastructure elements and workloads to detected anomalies in the system components and take corrective actions accordingly |  |  |
| sec.mon.012 | The Platform must Monitor and Audit Traffic patterns and volumes to prevent malware download attempts |  |  |
| sec.mon.013 | The monitoring system must not affect the security (integrity and confidentiality) of the infrastructure, workloads, or the user data (through back door entries). |  |  |
| sec.mon.014 | The Monitoring systems should not impact IAAS, PAAS, and SAAS SLAs including availability SLAs |  |  |
| sec.mon.015 | The Platform must ensure that the Monitoring systems are never starved of resources |  |  |
| sec.mon.016 | The Platform Monitoring components should follow security best practices for auditing, including secure logging and tracing |  |  |
| sec.lcm.017 | The Platform must Audit systems for any missing security patches and take appropriate actions |  |  |

### Compliance with Standards

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Requirement | Unit | Definition |
| sec.std.001 | The Cloud Operator **should** comply with Center for Internet Security CIS Controls (<https://www.cisecurity.org/>) |  | Center for Internet Security - <https://www.cisecurity.org/> |
|  | [Q: Are we going to verify compliance w Controls? If not, then why a “must” – but making it a “should” implies only guidance and not a control.] |  |  |
| sec.std.002 | The Cloud Operator, Platform and Workloads **should** follow the guidance in the CSA Security Guidance for Critical Areas of Focus in Cloud Computing (latest version) <https://cloudsecurityalliance.org/> |  | Cloud Security Alliance - <https://cloudsecurityalliance.org/> |
| sec.std.003 | The Platform and Workloads **should** follow the guidance in the OWASP Cheat Sheet Series (OCSS) <https://github.com/OWASP/CheatSheetSeries> |  | Open Web Application Security Project <https://www.owasp.org> |
| sec.std.004 | The Cloud Operator, Platform and Workloads **should** ensure that their code is not vulnerable to the OWASP Top Ten Security Risks <https://owasp.org/www-project-top-ten/> |  |  |
| sec.std.005 | The Cloud Operator, Platform and Workloads **should** strive to improve their maturity on the OWASP Software Maturity Model (SAMM) <https://owaspsamm.org/blog/2019/12/20/version2-community-release/> |  |  |
|  | The Cloud Operator, Platform and Workloads **should** utilize the OWASP Web Security Testing Guide <https://github.com/OWASP/wstg/tree/master/document> |  |  |
| sec.std.013 | The Cloud Operator, and Platform **should** satisfy the requirements for Information Management Systems specified in ISO/IEC 27001 <https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-2:v1:en> |  | ISO/IEC 27002:2013 - ISO/IEC 27001 is the international Standard for best-practice information security management systems (ISMSs) |
| sec.std.014 | The Cloud Operator, and Platform **should** implement the Code of practice for Security Controls specified ISO/IEC 27002:2013 (or latest) <https://www.iso.org/obp/ui/#iso:std:iso-iec:27002:ed-2:v1:en> |  |  |
| sec.std.015 | The Cloud Operator, and Platform **should** implement the ISO/IEC 27032:2012 (or latest) Guidelines for Cybersecurity techniques <https://www.iso.org/obp/ui/#iso:std:iso-iec:27032:ed-1:v1:en> |  | ISO/IEC 27032 - ISO/IEC 27032is the international Standard focusing explicitly on cybersecurity |
| sec.std.016 | The Cloud Operator **should** conform to the ISO/IEC 27035 standard for incidence management |  | ISO/IEC 27035 - ISO/IEC 27035 is the international Standard for incident management |
| sec.std.017 | The Cloud Operator **should** conform to the ISO/IEC 27031 standard for business continuity |  | ISO/IEC 27031 - ISO/IEC 27031 is the international Standard for ICT readiness for business continuity |
| sec.std.018 | The Public Cloud Operator **must**, and the Private Cloud Operator **may** be certified to be compliant with the International Standard on Awareness Engagements (ISAE) 3402 (in the US: SSAE 16) |  | International Standard on Awareness Engagements (ISAE) 3402. US Equivalent: SSAE16 |

### References

Network Functions Virtualisation (NFV);NFV Security; Problem Statement, ETSI GS NFV-SEC 001 V1.1.1 (2014-10)

Network Functions Virtualisation (NFV);NFV Security; Security and Trust Guidance, ETSI GS NFV-SEC 003 V1.1.1 (2014-12)

Network Functions Virtualisation (NFV) Release 3; Security; Security Management and Monitoring specification, ETSI GS NFV-SEC 013 V3.1.1 (2017-02)

Network Functions Virtualisation (NFV) Release 3; NFV Security; Security Specification for MANO Components and Reference points, ETSI GS NFV-SEC 014 V3.1.1 (2018-04)

Network Functions Virtualisation (NFV) Release 2; Security; VNF Package Security Specification, ETSI GS NFV-SEC 021 V2.6.1 (2019-06)

ETSI Industry Specification Group Network Functions Virtualisation (ISG NFV) - <https://www.etsi.org/committee/1427-nfv>

ETSI Cyber Security Technical Committee (TC CYBER) - <https://www.etsi.org/committee/cyber>

**NIST Documents**

NIST SP 800-53 Security and Privacy Controls for Federal Information Systems and Organizations <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf>

NIST SP 800-53A Assessing Security and Privacy Controls in Federal Information Systems and Organizations: Building Effective Assessment Plans <https://www.serdp-estcp.org/content/download/47513/453118/file/NIST%20SP%20800-53A%20Rev%204%202013.pdf>

NIST SP 800-63B Digital Identity Guidelines <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf>

NIST SP 800-115 Technical Guide to Information Security Testing and Assessment <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf>

NIST SP 800-123 Guide to General Server Security <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-123.pdf>

NIST SP 800-125 Guide to Security for Full Virtualization Technologies <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-125.pdf>

NIST SP 800-125a Security Recommendations for Server-based Hypervisor Platforms <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-125Ar1.pdf>

NIST SP 800-125b Secure Virtual Network Configuration for Virtual Machine (VM) Protection <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-125B.pdf>

NIST SP 800-137 Information Security Continuous Monitoring for Federal Information Systems and Organizations <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-137.pdf>

NIST SP 800-145 The NIST Definition of Cloud Computing <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

NIST SP 800-190 Application Container Security Guide <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf>

# Conformance, Verification, and Certification

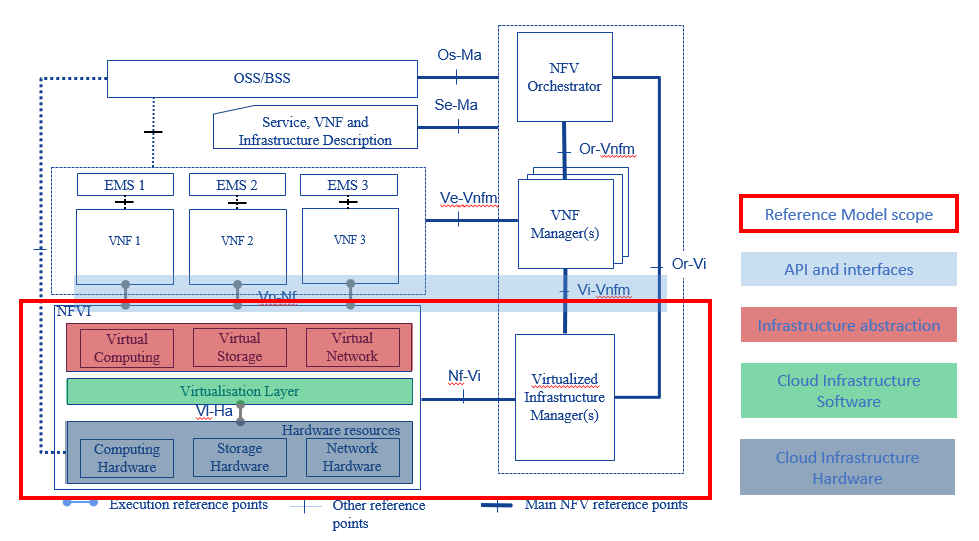
## Introduction

# Infrastructure Operations and Lifecycle Management

## Introduction

The purpose of this chapter is to define the capabilities required of the infrastructure to ensure it is effectively supported, maintained and otherwise lifecycle-managed by Operations teams. This includes requirements relating to the need to be able to maintain infrastructure services "in-service" without impacting the applications and VNFs, whilst minimising human labour. It shall also capture any exceptions and related assumptions.

According to the scope laid out in chapter 1, this chapter will include any requirements of the Cloud Infrastructure and the Cloud Infrastructure Management capabilities. This is reflected in Figure 25 below - the main area of interest for this chapter being the reference points between the reference model scope (in red) and the OSS/BSS block at the top.



1. : Mapping of the reference model scope to the ETSI NFV architecture

**Note**: this may seem like a large overlap with the topics in Chapter 6, however that chapter focusses primarily on the interfaces provided by the Cloud Infrastructure Management and Cloud Infrastructure to application management and applications, not the interfaces used to manage the Cloud Infrastructure and Cloud Infrastructure Management themselves.

There are two main business operating frameworks that are commonly known and used across the Telecommunications industry related to the topics in this chapter:

* FCAPS (ISO model for network management)
* eTOM (TM Forum Business Process Framework (eTOM))

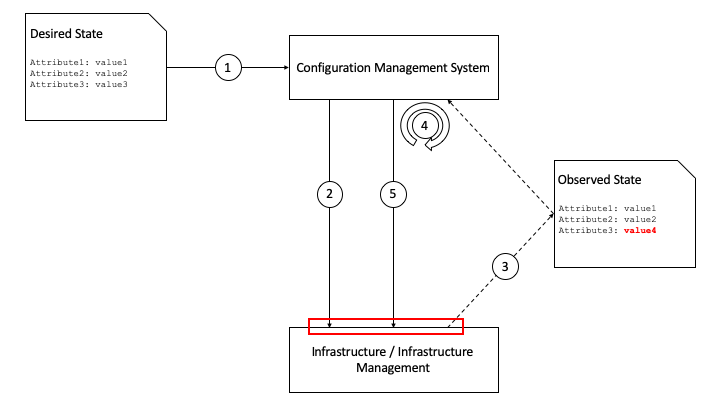
The chapters below roughly map to these frameworks as follows:

|  |  |  |
| --- | --- | --- |
| Chapter Name | FCAPS | eTOM |
| Configuration and Lifecycle Management | Configuration | Fulfilment |
| Assurance | Performance | Assurance |
| Fault |
| Capacity Management | Configuration | Fulfilment |

1. : Operating Frameworks

## Configuration and Lifecycle Management

Configuration management is concerned with defining the configuration of infrastructure and its components, and tracking (observing) the running configuration of that infrastructure, and any changes that take place. Modern configuration management practices such as desired state configuration management also mean that any changes from the desired state that are observed (aka the delta) are rectified by an orchestration / fulfilment component of the configuration management system. This "closed loop" mitigates against configuration drift in the infrastructure and its components. Our recommendation is to keep these closed loops as small as possible to reduce complexity and risk of error. Figure 26 shows the configuration management "loop" and how this relates to lifecycle management.



1. : Configuration and Lifecycle Management

The initial desired state might be for 10 hosts with a particular set of configuration attributes, including the version of the hypervisor and any management agents. The configuration management system will take that as input (1) and configure the infrastructure as required (2). It will then observe the current state periodically over time (3) and in the case of a difference between the desired state and the observed state it will calculate the delta (4) and re-configure the infrastructure (5). For each lifecycle stage (create, update, delete) this loop takes place - for example if an update to the hypervisor version is defined in the desired state, the configuration management system will calculate the delta (e.g. v1 --> v2) and re-configure the infrastructure as required.

However, the key requirements for the infrastructure and infrastructure management are those interfaces and reference points in the red box - where configuration is **set**, and where it is **observed**. Table 48 lists the main components and capabilities required in order to manage the configuration and lifecycle of those components.

|  |  |  |  |
| --- | --- | --- | --- |
| Component | set / observe | Capability | Example |
| Cloud Infrastructure Management Software | Set | Target software / firmware version | Software: v1.2.1 |
| Desired configuration attribute | dhcp\_lease\_time: 86400 |
| Desired component quantities | # hypervisor hosts: 10 |
| Observe | Observed software / firmware version | Software: v1.2.1 |
| Observed configuration attribute | dhcp\_lease\_time: 86400 |
| Observed component quantities | # hypervisor hosts: 10 |
| Cloud Infrastructure Software | Set | Target software version | Hypervisor software: v3.4.1 |
| Desired configuration attribute | management\_int: eth0 |
| Desired component quantities | # NICs for data: 6 |
| Observe | Observed software / firmware version | Hypervisor software: v3.4.1 |
| Observed configuration attribute | management\_int: eth0 |
| Observed component quantities | # NICs for data: 6 |
| Infrastructure Hardware | Set | Target software / firmware version | Storage controller firmware: v10.3.4 |
| Desired configuration attribute | Virtual disk 1: RAID1 [HDD1, HDD2] |
| Observe | Observed software / firmware version | Storage controller firmware: v10.3.4 |
| Observed configuration attribute | Virtual disk 1: RAID1 [HDD1, HDD2] |

1. : Configuration and Lifecycle Management Capabilities

This leads to the following table (Table 49) which defines the standard interfaces that should be made available by the infrastructure and Cloud Infrastructure Management components to allow for successful Configuration Management.

|  |  |  |
| --- | --- | --- |
| Component | Interface Standard | Link |
| Infrastructure Management | ... | ... |
| Infrastructure Software | ... | ... |
| Infrastructure Hardware | Redfish API | [DMTF Specification](https://www.dmtf.org/standards/redfish) |

1. : Interface Standards for Configuration Management

## Assurance

Assurance is concerned with:

* The proactive and reactive maintenance activities that are required to ensure infrastructure services are available as per defined performance and availability levels.
* Continuous monitoring of the status and performance of individual components and of the service as a whole.
* Collection and analysis of performance data, which is used to identify potential issues including the ability to resolve the issue with no customer impact.

There are the following requirement types:

1. Data collection from all components, e.g.
   * The ability to collect data relating to events (transactions, security events, physical interface up/down events, warning events, error events, etc.)
   * The ability to collect data relating to component status (up/down, physical temperature, disk speed, etc.)
   * The ability to collect data relating to component performance (used CPU resources, storage throughput, network bandwidth in/out, API transactions, transaction response times, etc.)
2. Capabilities of the Infrastructure Management Software to allow for in-service maintenance of the Infrastructure Software and Hardware under its management, e.g.
   * The ability to mark a physical compute node as being in some sort of "maintenance mode" and for the Infrastructure Management Software to ensure all running workloads are moved off or rescheduled on to other available nodes (after checking that there is sufficient capacity) before marking the node as being ready for whatever maintenance activity needs to be performed
   * The ability to co-ordinate, automate, and allow the declarative input of in-service software component upgrades - such as internal orchestration and scheduler components in the Infrastructure Management Software

Note that the above only refers to components - it is expected that any "service" level assurance doesn't add any further requirements onto the infrastructure, but rather takes the data extracted and builds service models based on the knowledge it has of the services being offered.

## Capacity Management

Capacity Management is a potentially wide ranging process that includes taking demand across lines of business, analysing data about the infrastructure that is running, and calculating when additional infrastructure might be required, or when infrastructure might need to be decommissioned.

As such the requirements for Capacity Management on the infrastructure are covered by the Assurance and Configuration and Lifecycle Management sections above. The Assurance section deals with the collection of data - there is no reason to consider that this would be done by a different mechanism for Capacity Management as it is for Assurance - and the Configuration and Lifecycle Management section deals with the changes being made to the infrastructure hardware, software, and management components (e.g. changing of number of hypervisor hosts from 10 to 12).

# Challenges and Gaps

## Introduction [Draft Language]

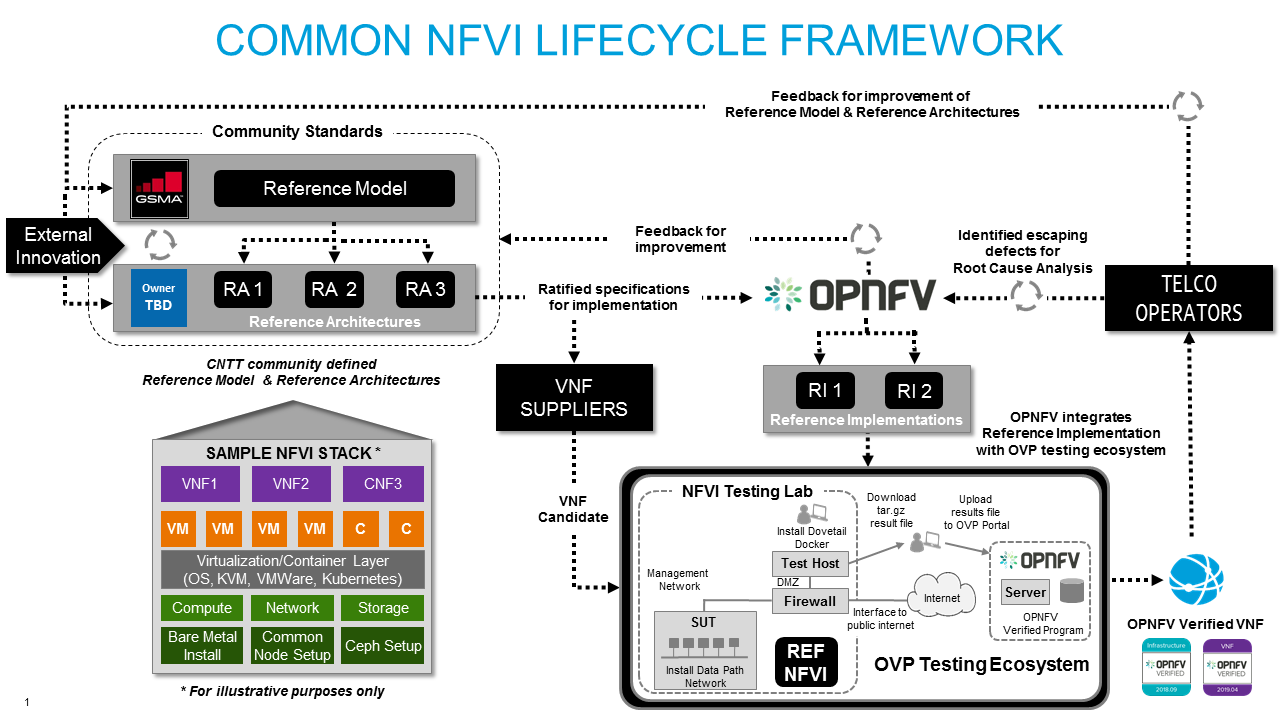
The development of this Reference Model is in its initialization stage and it will go through several iteration cycles before it is adding the full value intended. The reference model is also the first component in the establishment of a Common Cloud Infrastructure Lifecycle Framework--see also diagram below:

Key Components of The Common Cloud Infrastructure Lifecycle Framework

* Reference Model
* Reference Architecture
* Reference Implementation
* Verification & Validation
* Lifecycle Feedback Loops

As the first component, the reference model must also be structured and iterated so that it plays its part as the foundation on which the other five components of the ecosystem are built.

Going forward, this chapter shall be dedicated to identifying the challenges and gaps found in the course the development--to ensure the reference model is adding the strategic and tactical value intended over time. Should a challenge or gap not be identified and it is not already addressed in the model itself, the community may assume it will remain an unknown and, therefore, the community is welcomed to engage with and raise an issue with the appropriate working group(s) to close the gap. In this manner, the Reference Model can continuously improve.

For Reference - Common Cloud Infrastructure Lifecycle Framework Diagram (Figure 27) 

1. : Common Cloud Infrastructure Lifecycle Framework Diagram

## Challenges [Initial Language]

Challenge 1 The initial challenge was launching the Reference Model as an MVP and then gathering and incorporating necessary feedback from the community on the initial assumptions, and then iterating foundational constructs of the model with that feedback. This release includes the results of a significant effort to incorporate the feedback. Of course, alignment efforts will always continue.

Challenge 2 The next challenge ahead is getting to a stable working version from which all stakeholders in the application value-chain can begin to extract the intended value of a Common Cloud Infrastructure. The next maturity level is reached when the Reference Model has stabilized enough for stakeholders to begin to intake the model into their own application development and deployment cycles.

[Diagram-TBD]

## Gaps [Initial Language]

Gaps

This section includes a backlog table of all existing open issues in the development of the Reference Model, Reference Architecture, Reference Implementation of the Common Cloud Infrastructure Lifecycle Framework. Each issue is captured in the CNTT document repository and is viewable there as a reference and a work in progress item.

Backlogs [https://github.com/cntt-n/CNTT/issues]

* Reference Model
* Reference Architecture
* Reference Implementation

1. Guidelines For Application Vendors
   1. Goals

This Appendix has two goals:

1. Provide guidance to VNF or more generally Application vendors on how to consume CNTT Reference Model and Architectures
2. Provide usable definitions of maturity levels for VNF software architecture between Physical-to-Virtual migration and “Cloud Native”.

The goal is not to be prescriptive on how to re-architect existing or architect new applications but rather staying within scope of focusing on interface and interaction between applications and platform.

* 1. Intro and Terminology

Taking advantage of RM and RA environments with common capabilities, applications can be developed and deployed more rapidly, providing more service agility and easier operations. The extent to which this can be achieved will depend on levels of decoupling between application and infrastructure or platform underneath the application:

1. Infrastructure:
   1. Application functionality or application control requires infrastructure components beyond RM profiles or infrastructure configuration changes beyond RA exposed APIs. Generally, such an application is tightly coupled with the infrastructure which results in an Appliance deployment model.
   2. Application control using RA APIs finds node (already configured in support of the profiles) with required infrastructure component(s), and in that node using RA APIs configures infrastructure components that make application work. Example is application that to achieve latency requirements needs certain acceleration adapter available in RM profile and is exposed through RA APIs.
   3. Application control using RA APIs finds node (already configured in support of the profiles) with optional infrastructure component(s), and in that node using RA APIs configures infrastructure component(s) that make application work better (like more performant) than without that infrastructure component. Example is application that would have better TCO with certain acceleration adapter but can also work without it.
   4. d. Application control using RA APIs finds general profile node without any specific infrastructure component.
2. Platform Services
   1. Application functionality or application control can work only with its own components instead of using RA-defined Platform Services.
   2. With custom integration effort, application can be made to use RA-defined Platform Services.
   3. Application is designed and can be configured for running with RA-defined Platform Services.
3. Application Resiliency
   1. Application was designed and tested to run only on Carrier Grade platform with predictable infrastructure availability and performance.
   2. Application was designed and tested for full failures of infrastructure HW and SW components, but not for infrastructure impairment as the Application still needs predictable infrastructure performance (like CPU cycles and network latencies).
   3. Application was designed to run on shared Cloud platforms and tested for resilience to infrastructure impairments.

Relevant for sizing infrastructure and application operations (which often is another telco organizational unit or external 3rd party) is also how much is application decomposed from:

1. **Other application functionality** (decomposition and manageability for scaling, availability and upgrades):
   1. Application consists of huge monolithic components including algorithms that have different scaling (for example depending on type of traffic) and/or availability requirements.
   2. Application consists of smaller tightly coupled components.
   3. Decomposed application with loosely- or decoupled components.
   4. Availability like N+K or 1+1 is defined during application design and not configurable at deployment time.
   5. Mutable or immutable instances of application components.
   6. Exception List

As Part of the [Transition Plan](file:///C:\\Users\\tpelt\\Documents\\Infrastructure\\cNTT\\RM\\gov\\chapters\\chapter09.md" \l "9.2) described in the adoption strategy, following table explains the exceptions allowed in this RM release. The list of Exceptions described here are considered to be against CNTT principles and will be removed in future releases as soon as an alternative technology that is aligned with CNTT principles develops and matures.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Type | Name | Description |
| rm.exc.001 | Technology | SRIOV | This exception allows workloads (VNF/CNF) to rely on SR-IOV over PCI-PassThrough technology. |
| rm.exc.002 |  |  |  |
| rm.exc.003 |  |  |  |

1. : Exceptions allowed in this RM release
   1. Links
   2. Hardware-Dependent Coding Policies

As described in the Principles sections of RM Chapter 1, features that require hardware-dependent code in the workload are prohibited in CNTT compliant implementations. This principle is henceforth referred to as the "Abstraction Principle". Note, this prohibition does not apply to the Compute node host software (e.g., host OS). Within the Infra, hosts are expected to have software that is customized for the specific hardware equipped. However, the intent is that these software drivers and higher layers will abstract the Capabilities they enable, thereby exposing them with an open API. An example of exposing capabilities in this manner is implemented in the Virtio family of APIs. This requirement is in support of VNF abstraction and portability of VNFs across the Infra landscape.

CNTT realizes there are implications and limitations to the ability to live by the Abstraction Principle. A textbook example of a Capability that transgresses this principle, is SR-IOV over PCI-PT 1). Other, less notable, yet very important examples include GPU and other acceleration hardware, such as FPGA. A less obvious, yet critically important example is the VNF program(s) itself. As workloads are coded in native microarchitecture opcodes, microarchitectures' instruction sets effectively constitute ABIs (Application Binary Interface). Additionally, the VNF programs may or may not, attempt to execute vendor-specific extensions to base instruction sets, such as de facto x86 or ARM.

Solving the problems associated with implementing the Abstraction Principle is a work in progress. CNTT has not solved all of the associated problems, nor has the industry. As technology evolves and more designs incorporate cloud native concepts, these problems will be addressed. This appendix is specifically intended to provide CNTT policies to manage these situations as they exist today, and their exceptions and transitions, as the technology around and supporting the Abstraction Principle matures.

1) For a detailed description of the mechanisms underlying PCI-PT (PCI-PassThrough), refer to [Section 8.1](https://github.com/cntt-n/CNTT/blob/master/doc/tech/technologies.md" \l "8.1) of Relevant Technologies.

Several specific technology areas have been identified by CNTT as using an ABI impacted by the Abstraction Principle, as follows:

* SR-IOV over PCI-PT
* GPU/NPU
* FPGA/Other Acceleration
* CPU Instruction Sets and Extensions

The preceding list is not exhaustive; technologies will be added as required.

Current CNTT Policies:

**SR-IOV over PCI-PT:**

CNTT recognizes that today, SR-IOV over PCI-PT provides a critical Capability for increased throughput over network interfaces at an economical cost. As such, the CNTT approach to SR-IOV over PCI-PT is to (detailed policy language under development; to be tied in with VNF Evolution).

Without arguing for or against SR-IOV over PCI-PT, CNTT provides the following anecdotes which have been raised in discussions over SR-IOV:

* SR-IOV over PCI-PT mitigates the need for duplicated servicing of interrupts from unbuffered (i.e. small buffer) NICs, however it does not reduce the number of frame reception driven interrupts which much be serviced.
* SR-IOV over PCI-PT increases the Fabric management complexity, as encapsulation must be applied by the ToR/Leaf interface and the encapsulation must be managed as VNFs and/or networks are added/deleted from the Tenant. Therefore, performance isn't the only factor; fabric touch points and Service Chaining must also be considered.
* Indications are that technologies such as DPDK, VPP, FD.io and others offer comparable throughput, today.(Citations Needed)

**GPU/NPU:**

**FPGA/Other Acceleration:**

**CPU Instruction Sets and Extensions:**

End of policies.

* 1. VNF Design Guidelines

A number of software design guidelines (industry best practices) have been developed over the years including micro-services, cohesion and coupling. In addition to the industry best-practices, there are additonal guidelines and requirements specified by ONAP in "[VNF or PNF Requirements Documentation](https://onap.readthedocs.io/en/latest/submodules/vnfrqts/requirements.git/docs/index.html)." This section does not supplant these well-known guidelines and practices. The content here only draws attention to some other design consideration that VNF Developers need to incorporate in their practices. Please note that some of these guidelines may be incorporated by operators in their contracts with VNF Vendors.

These guidelines are written in an informal style and any resemblance to requirements is incidental. The VNF Developer **should** ensure that their software and the resultant VNF image:

1. does not contain malicious code (e.g., malware, logic bombs, etc.).
2. does not contain code such as daemons that exposes them to risk.
3. does not contain clear text secrets.
4. are only created with content and files from trusted sources.
5. are only packaged with files that have been found free of malware and vulnerabilities.

Additionally, in the design and implementation of their software, the VNF Developer **should** follow the guidance in the:

1. [CSA Security Guidance for Critical Areas of Focus in Cloud Computing (latest version)](https://cloudsecurityalliance.org).
2. [OWASP Cheat Sheet Series (OCSS)](https://github.com/OWASP/CheatSheetSeries) from the [Open Web Application Security Project](https://www.owasp.org).

The VNF Developer **should** ensure that their code is not vulnerable to the [OWASP Top Ten Security Risks](https://owasp.org/www-project-top-ten/) created by the [Open Web Application Security Project](https://www.owasp.org).

* 1. Miscellaneous
     1. VNF Network Monitoring Capabilities - UseCase.

Network Monitoring capabilities exposed by NFVI Platform are used for the passive observation of VNF-specific traffic traversing the NFVI when:

* Performance issues and/or packet drops reported in VNF
* Determining performance bottle necks at VNF level
* Doing anomaly detection and network forensics

**Note:** It is responsibility of NFVI Platform to expose capability to create virtual interface having mirrored traffic from monitored VNF. This port can be attached to Monitoring VNF so that all traffic from Monitored VNF would be available for troubleshooting/debugging purpose.

1. Document Management
   1. Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Brief Description of Change | Approval Authority | Editor / Company |
| 1.0 |  | New PRD (WG Doc nn/nnn). | WG #nn  PSMC #nn |  |
| 1.1 |  | Minor CR nnn (WG Doc nn/nnn).<description of change><reason for change> | WG #nn |  |
| 2.0 |  | Major CR nnn (WG Doc nn/nnn).<description of change><reason for change> | eVote  PSMC #nn |  |

* 1. Other Information

|  |  |
| --- | --- |
| Type | Description |
| Document Owner | <Working Group/Project> |
| Editor / Company |  |

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at [prd@gsma.com](mailto:prd@gsma.com)

Your comments or suggestions & questions are always welcome.