



3. NFV - Network Function Virtualization

2023-2024

Ecole Supérieure Polytechnique (ESP) / Département Génie Informatique (DGI)

DIC3 TR (Télécommunications et Réseaux) & M2 SRT (Systèmes Réseaux et Télécommunications)

Cours : SDN et NFV

Enseignant : Dr Ousmane SADIO



3. NFV - Network Function Virtualization

La NFV (Network Function Virtualization) ou la virtualisation de fonctions réseau est née de discussions entre les principaux opérateurs de réseau sur la manière d'améliorer l'exploitation des réseaux à l'ère du multimédia à haut débit.

Ce groupe a indiqué que l'objectif global de la NFV est de tirer parti de la technologie de virtualisation pour consolider de nombreux types d'équipements réseau sur des serveurs, des commutateurs et des systèmes de stockage à haut volume, conformes aux normes industrielles, qui pourraient être situés dans des datacenters, des nœuds de réseau...

L'approche NFV s'éloigne de la dépendance à l'égard d'une variété de plateformes matérielles pour privilégier l'utilisation de techniques de virtualisation pour fournir la fonctionnalité réseau nécessaire.

De cette manière, le logiciel et le matériel sont découplés, et la capacité de chaque application dépend de l'ajout ou la réduction de ressources virtuelles.



3. NFV - Network Function Virtualization

□ NFV : concept

NFV découple les fonctions de réseau, telles que le NAT, le pare-feu, l'IPS/IDS, le DNS et la mise en cache, de matériel propriétaire afin qu'elles puissent être exécutées de façon logiciel sur les VM.

NFV s'appuie sur les technologies VM standard, en étendant leur utilisation dans le domaine de la mise en réseau.

La technologie des machines virtuelles, permet de faire migrer les équipements réseau dédiés vers des serveurs x86 / x64 du commerce. Ces équipement sont :

- **Dispositifs de fonction réseau** : tels que les commutateurs, les routeurs, les points d'accès, les inspections profondes de paquets...
- **Dispositifs informatiques liés au réseau** : tels que les pare-feu, les systèmes de détection des intrusions....
- **NAS (Network-Attached Storage)**: serveurs de fichiers et de bases de données connectés au réseau.

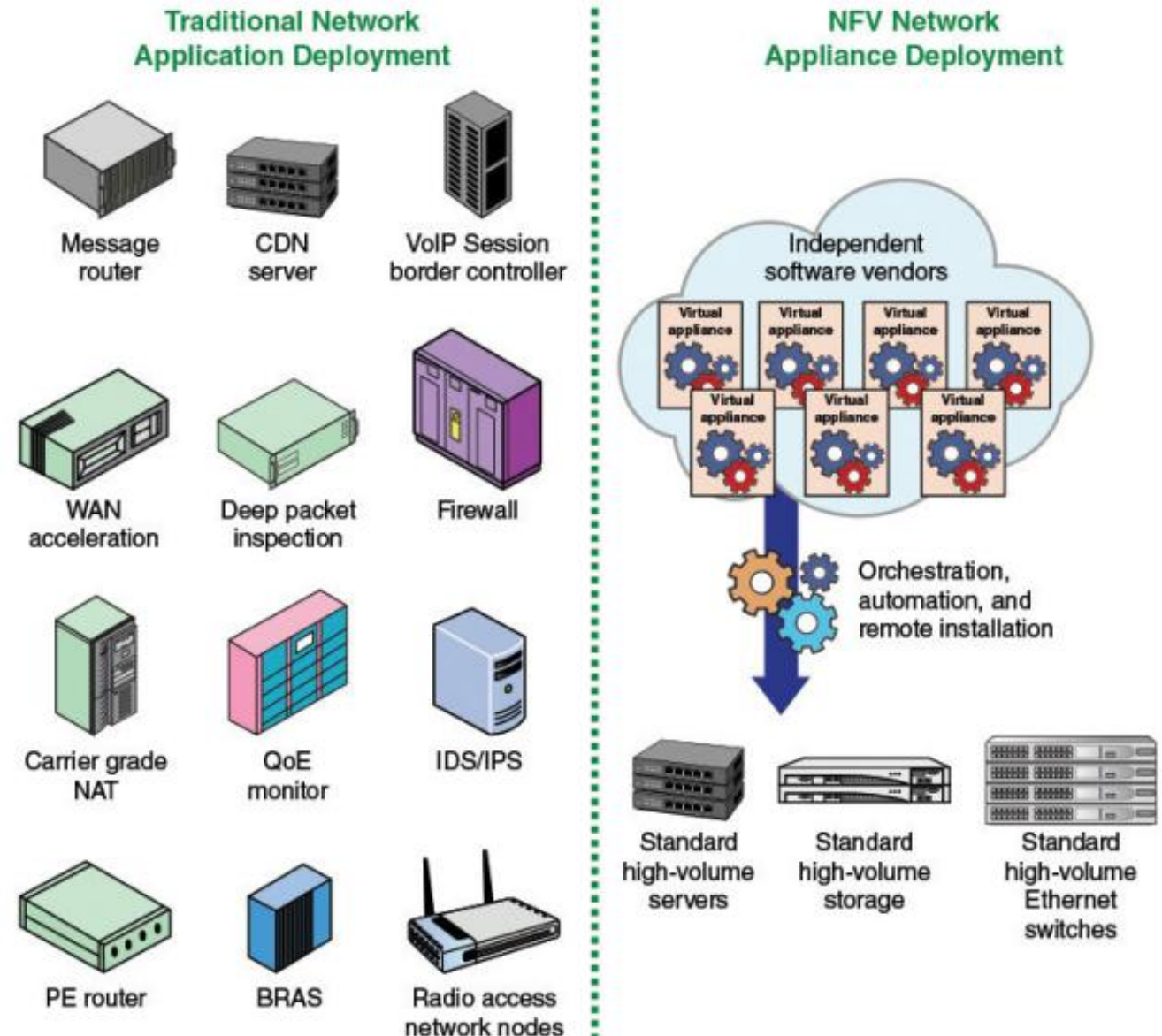
3. NFV - Network Function Virtualization

□ NFV : concept

- **Réseaux traditionnels** : tous les dispositifs sont déployés sur des plateformes propriétaires / fermées. Pour augmenter la capacité réseau, il faut du matériel supplémentaire. Mais ce matériel est inutilisé lorsque le système fonctionne en dessous de sa capacité.
- **Réseaux NFV** : les éléments de réseau sont des applications qui sont déployées de manière flexible sur une plateforme unifiée comprenant des serveurs, des dispositifs de stockage et des commutateurs standard.

CDN = content delivery network
WAN = wide area network
NAT = network address translation
QoE = quality of experience
VoIP = voice over Internet Protocol

IDS = intrusion detection system
IPS = intrusion prevention system
PE = provider edge router
BRAS = broadband remote access server

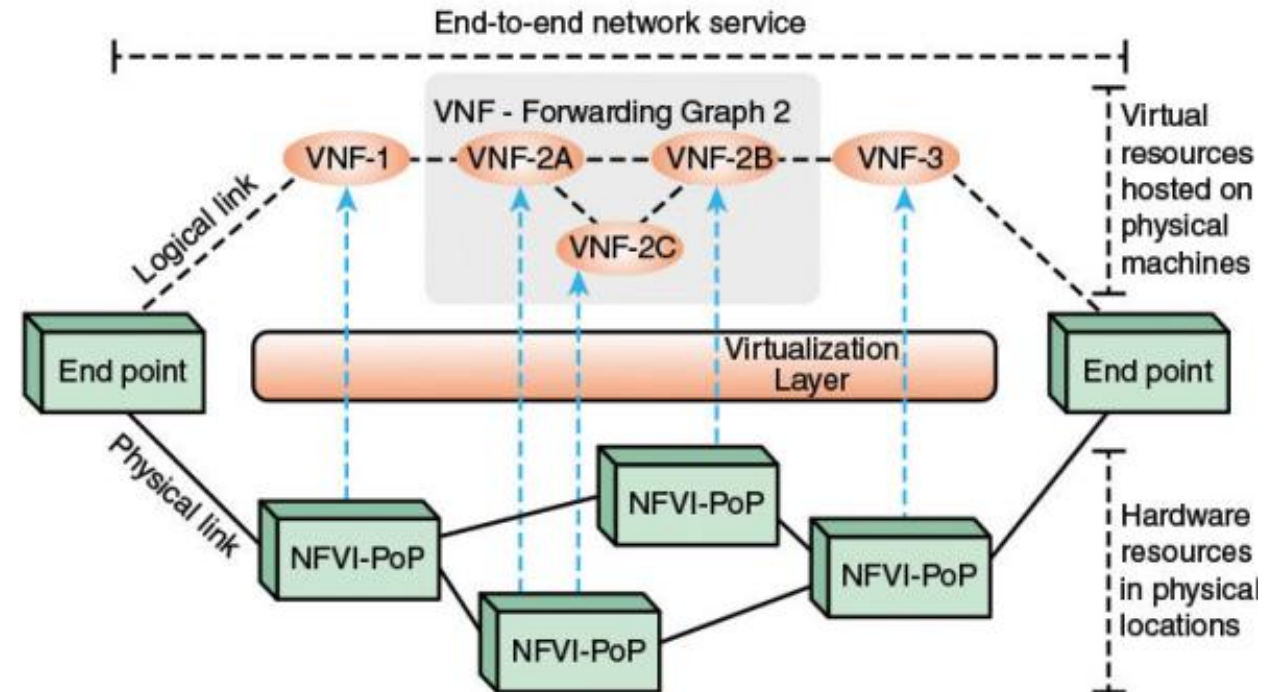


3. NFV - Network Function Virtualization

□ NFV : principes

Les VNF (Virtualized Network Function) sont les éléments constitutifs utilisés pour créer des services de réseau de bout en bout. Trois principes NFV clés sont impliqués dans la création de services de réseau pratiques :

- **Chainage de services** : les VNF sont modulaires et chaque VNF fournit une fonctionnalité limitée. Pour un flux de trafic donné, plusieurs VNF sont utilisés pour obtenir la fonctionnalité de réseau souhaitée.
- **Management and orchestration (MANO)** : implique le déploiement et la gestion du cycle de vie des instances VNF. MANO gère également les NFV Infrastructure (NFVI). Un NFVI-PoP (Point of Presence) est une plateforme dans laquelle une VNF peut être déployé.
- **Architecture distribuée** : une VNF peut être constitué de VNF Components (VNFC) dont chacun implémente un sous-ensemble de la VNF. Ces instances peuvent être déployées sur des hôtes distincts, distribués pour assurer l'évolutivité et la redondance



□ NFV : standardisation

Le groupe de normalisation ISG NFV, une branche de l'ETSI, a été créé en 2012 par sept grands opérateurs de réseaux de télécommunications. Ses membres se sont depuis élargis à d'autres acteurs de l'industrie.

ETSI : European Telecommunications Standards Institute

ISG : Industry Standards Group

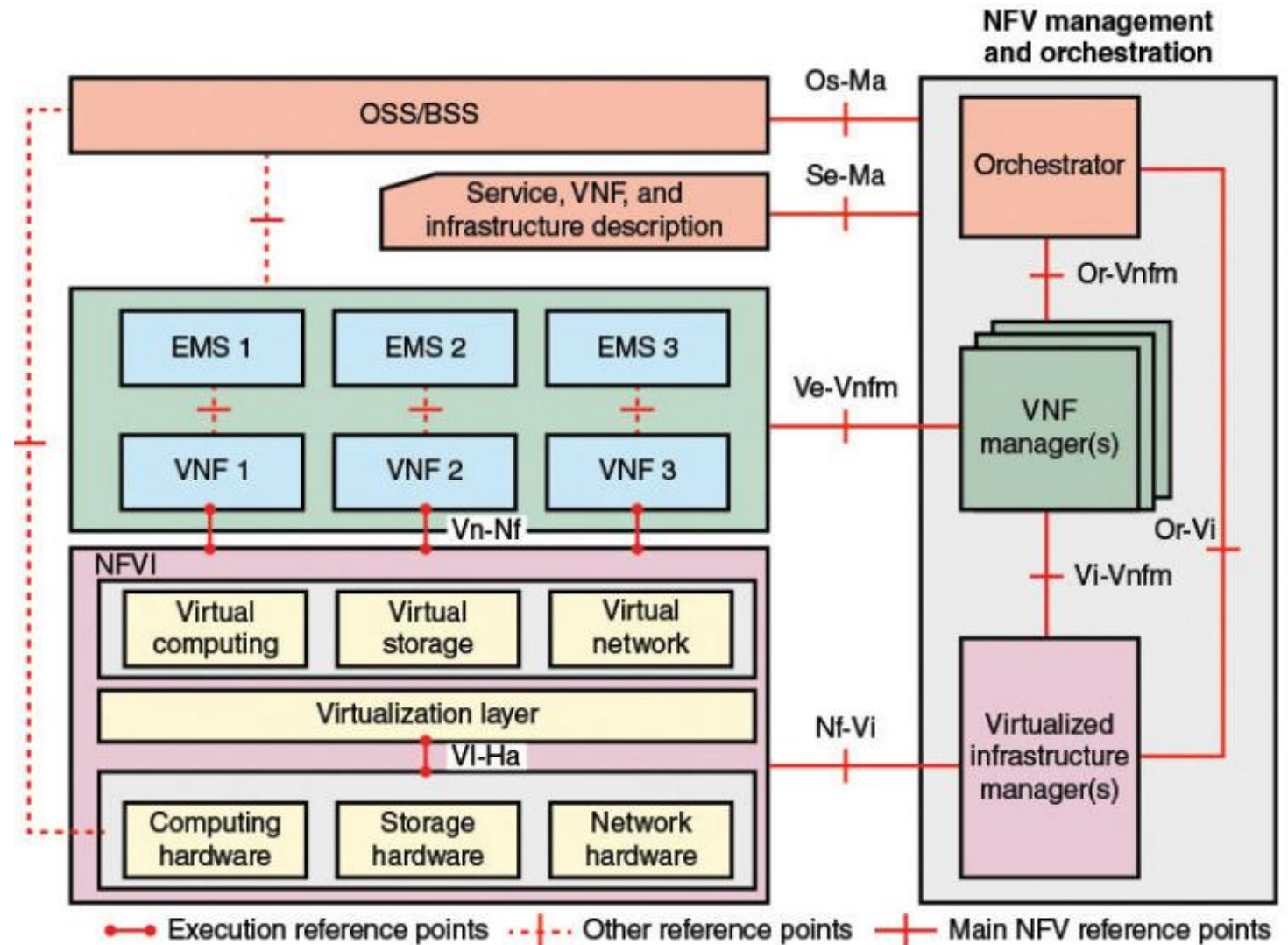
Le groupe ISG NFV joue un rôle central dans la normalisation NFV.

- **NFV Release 4 (2019-2020)** : met l'accent sur la prise en charge des technologies de conteneurisation, de l'optimisation l'abstraction de la NFVI pour réduire la dépendance des VNF à l'infrastructure, et optimiser l'intégration réseau dans l'infrastructure afin de faciliter la connectivité des VNF et des NS (Network Service).
- **NFV Release 3 (2017-2018)** : se focalise sur l'enrichissement du cadre architectural NFV afin de la rendre opérationnelle pour un déploiement et des opérations à l'échelle mondiale. Prise en charge des dernières technologies de réseau, telles que le Edge Computing et le Network Slicing.
- **NFV Release 2 (2015-2016)**: la nécessité de produire des spécifications normatives a constitué une partie importante de cette phase. Le travail a couvert les étapes de spécification des exigences, de l'architecture, des interfaces, des modèles d'information et des protocoles...

3. NFV - Network Function Virtualization

□ NFV : architecture de référence

- **NFV Infrastructure (NFVI)** : comprend les ressources matérielles et logicielles qui créent l'environnement dans lequel les VNF sont déployés.
- **VNF/EMS**: ensemble de VNF implémentés dans un logiciel pour s'exécuter sur des ressources virtuelles ainsi qu'un ensemble de EMS (Element Management Systems) qui gèrent les VNF.
- **NFV management and orchestration (NFV-MANO)**: framework pour la gestion et l'orchestration de toutes les ressources dans l'environnement NFV.
- **OSS/BSS** : systèmes de support opérationnel et business implémentés par le fournisseur de services VNF.

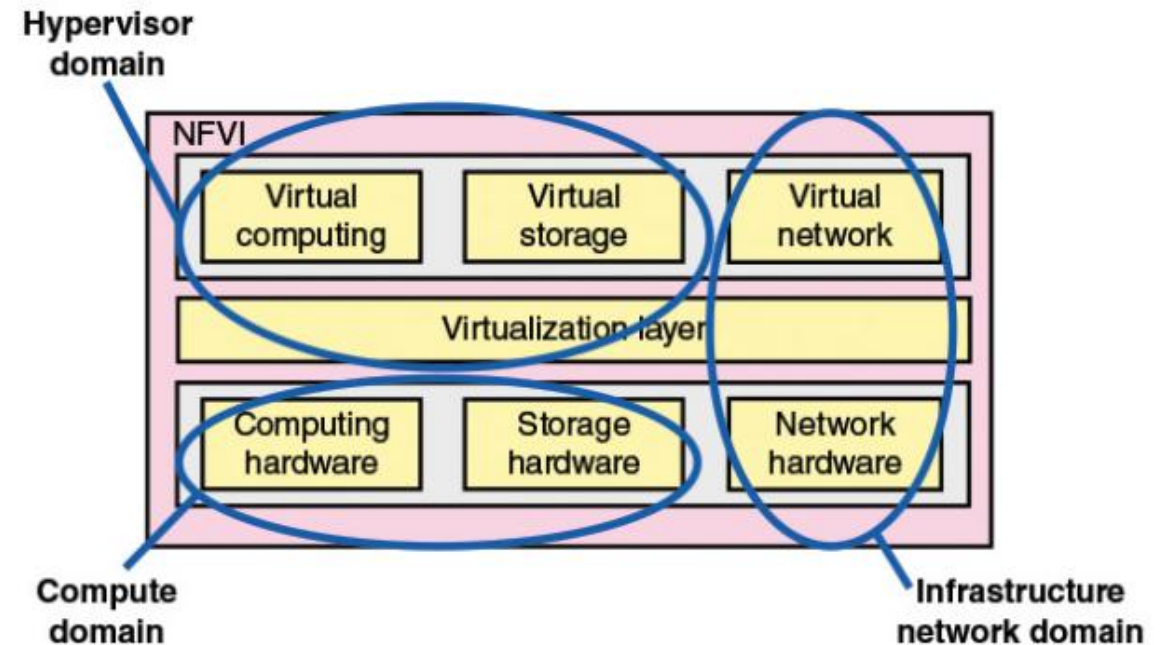


3. NFV - Network Function Virtualization

❑ NFV Infrastructure (NFVI)

Le cœur de l'architecture NFV est un ensemble de ressources et de fonctions appelé NFV infrastructure (NFVI). Le NFVI englobe trois domaines :

- **Compute domain** : fournit des serveurs et des systèmes de stockage standard à haute capacité.
- **Hypervisor domain** : est un environnement logiciel qui fait abstraction du matériel et implémente des services, tels que le démarrage et l'arrêt d'une VM, l'application de politiques, la mise à l'échelle, la live migration et la haute disponibilité.
- **Infrastructure Network Domain (IND)** : comprend tous les commutateurs génériques à haute capacité interconnectés dans un réseau qui peut être configuré pour fournir des services de réseau d'infrastructure.

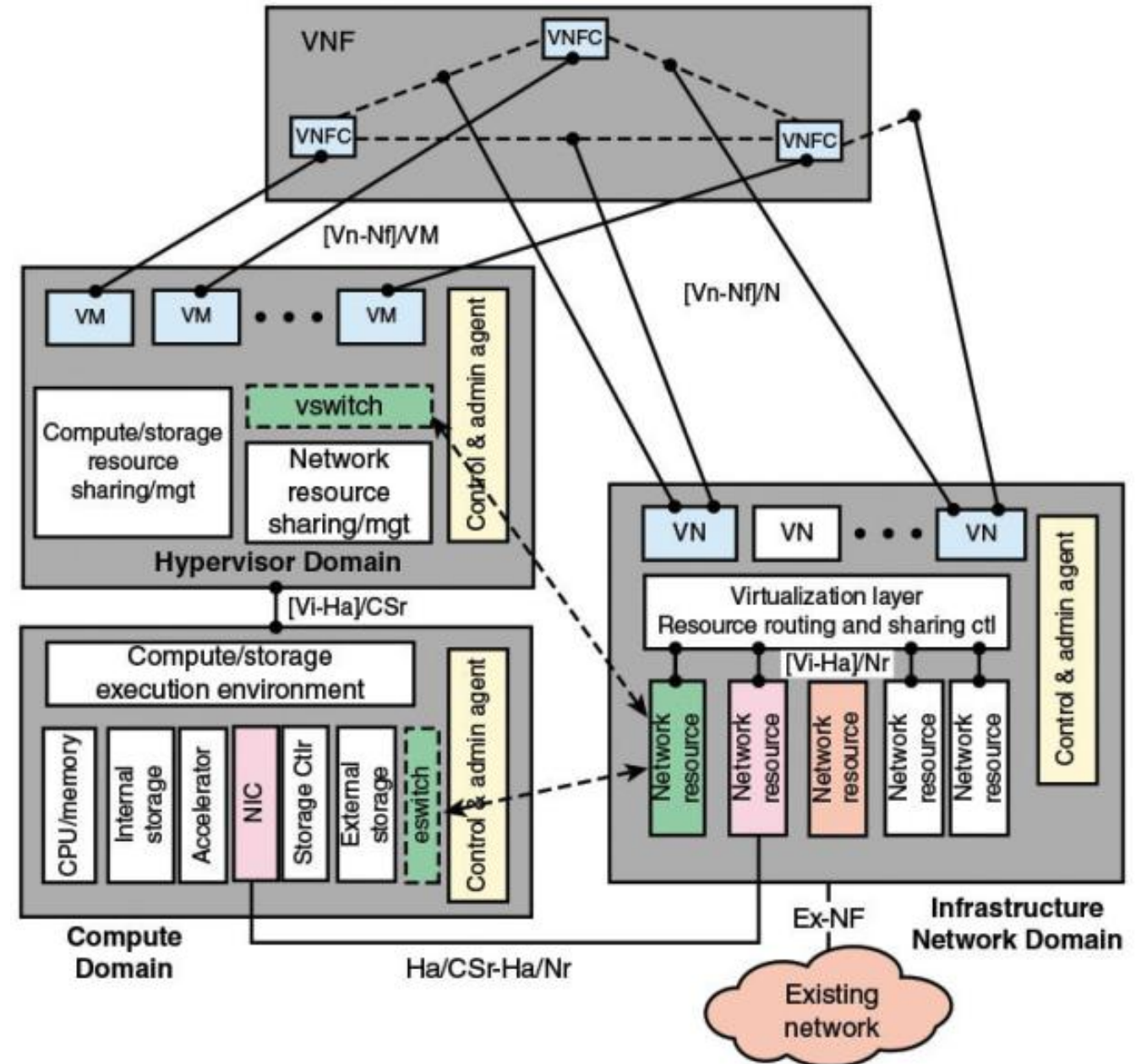


3. NFV - Network Function Virtualization

□ NFV Infrastructure (NFVI)

■ Structure logique des domaines

Cette structure fournit un framework pour permettre le développement des domaines NFVI ainsi que l'identification des interfaces entre les principaux composants.





3. NFV - Network Function Virtualization

❑ Virtualized Network Functions (VNF)

Un VNF est une implémentation virtualisée d'une fonction réseau traditionnelle. Le tableau suivant indique quelques exemples de fonctions qui pourraient être virtualisées.

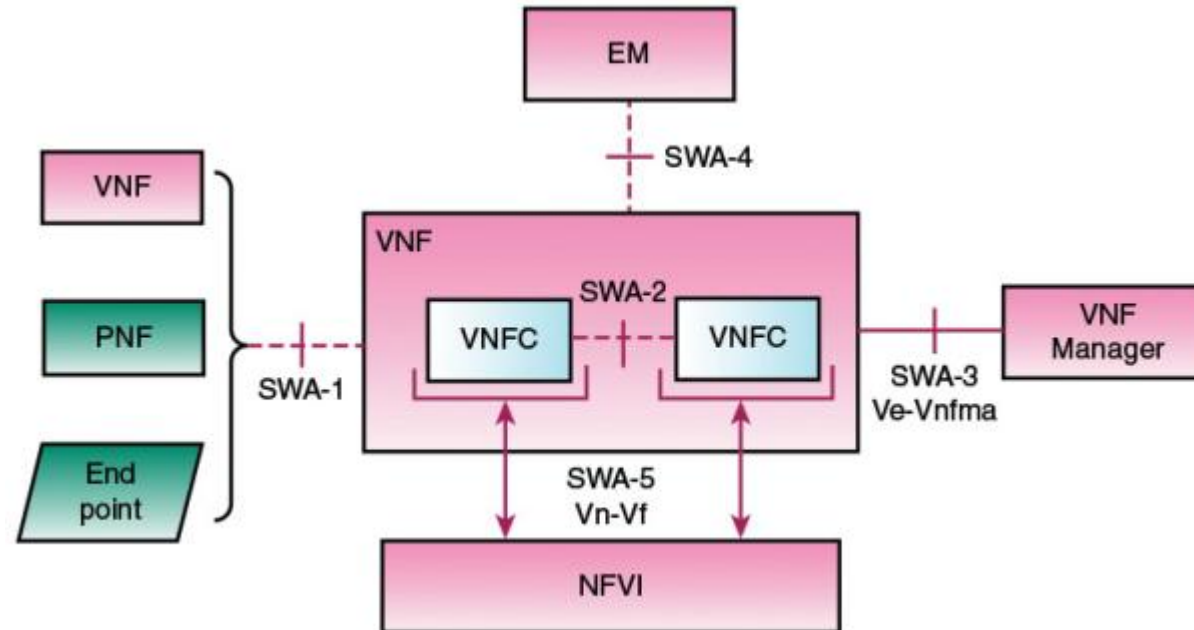
Éléments réseau	Fonctions
Élément de commutation	BAS (Broadband Access Server), NAT, commutateur, routeur
Nœuds réseau mobile	HLR/HSS, passerelle, contrôleur d'accès radio, fonctions Node B
Passerelle de tunneling	Passerelle VPN IPsec/SSL
Analyse trafic	DPI (Deep Packet Inspection), mesure de la QoE
Signalisation	SBC (Session Border Controller), composants IMS
Fonctions d'accès	Serveur AAA, serveur DHCP
Optimisation d'application	CDN (Content Delivery Network), serveurs cache, équilibreur de charge

3. NFV - Network Function Virtualization

❑ Virtualized Network Functions (VNF)

■ Interfaces

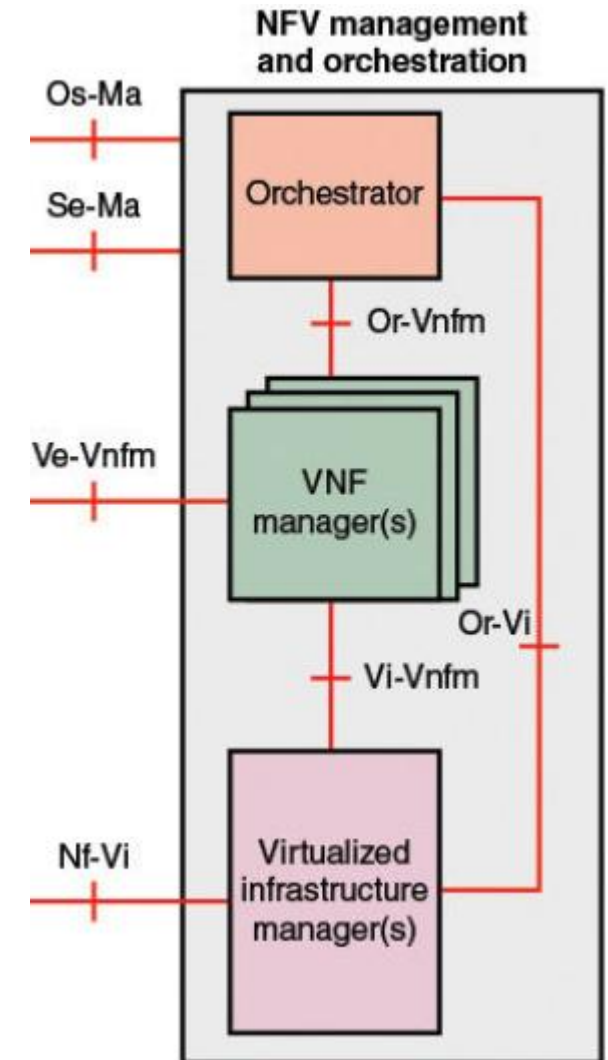
Un VNF est constitué d'un ou plusieurs VNF Components (VNFC). La figure suivante indique l'ensemble des interfaces utilisées au sein d'un VNF.



❑ Management and Orchestration (MANO)

L'installation MANO comprend les éléments de blocs fonctionnels suivants :

- **NFV Orchestrator (NFVO):** responsable de l'orchestration des ressources et l'orchestration des services réseau.
 - *Orchestration des ressources* : gère et coordonne les ressources sous la responsabilité des VIM : coordonne, autorise, libère et engage des ressources NFVI entre différents PoP.
 - *Orchestration des services réseau* : gère/coordonne la création d'un service de bout en bout qui implique des VNF de différents domaines de VNFM.
- **VNF Manager (VNFM):** supervise la gestion du cycle de vie des instances VNF. Plusieurs VNFM peuvent être déployés: un VNFM peut être déployé pour chaque VNF, ou un VNFM peut servir plusieurs VNF.
- **Virtualized Infrastructure Manager (VIM)** : contrôle et gère l'interaction d'une VNF avec les ressources de calcul, de stockage et de réseau sous son autorité, en plus de leur virtualisation.



❑ Usage des OSS/BSS

Les OSS/BSS doivent passer d'une gestion traditionnelle centrée sur les ressources à une gestion centrée sur les services, les avantages promis étant l'efficacité des opérations et l'agilité des services.

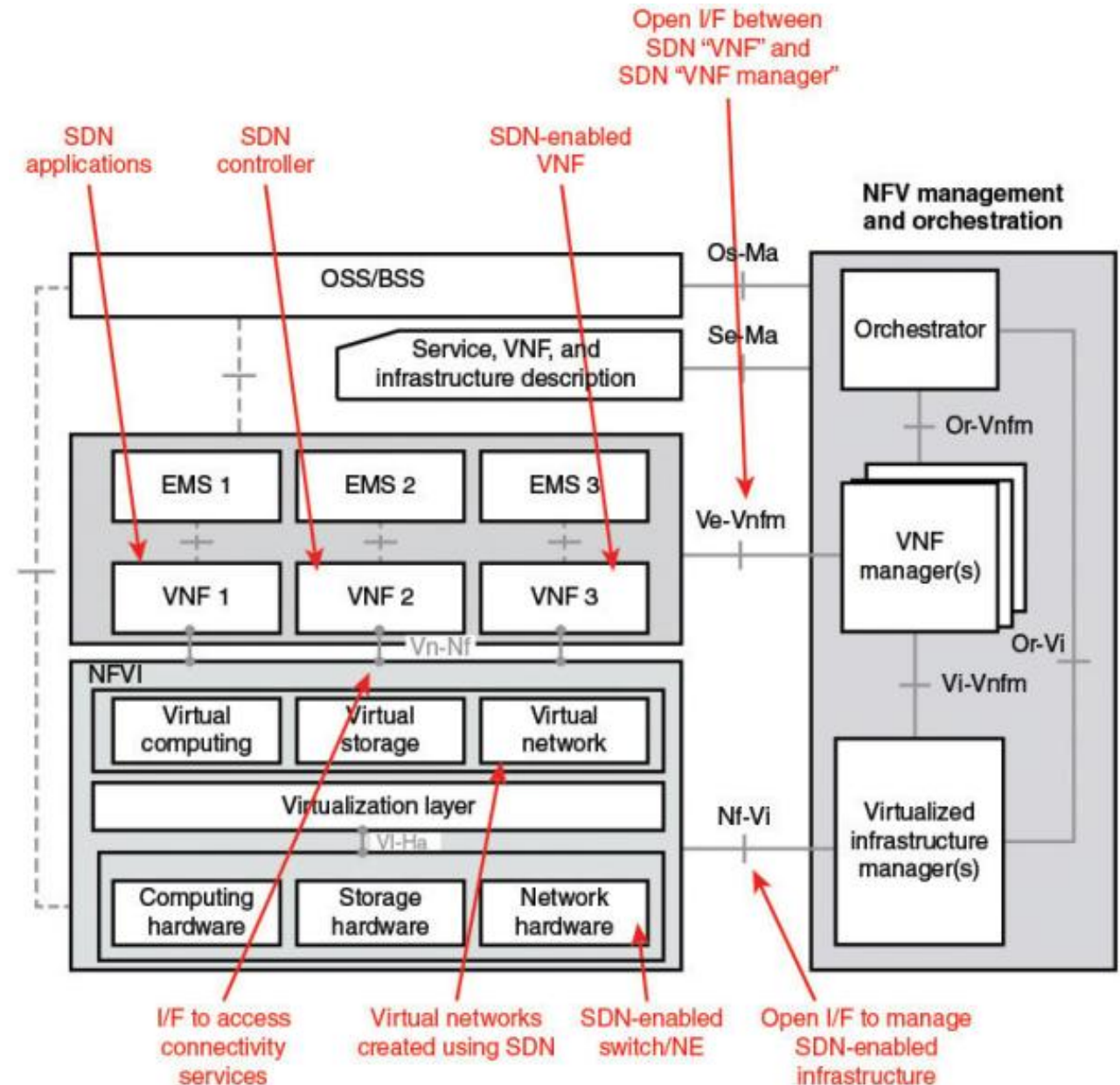
- **Support système OSS (Operation Support System)** : suite logicielle permettant d'administrer le réseau opérateur et de superviser les ressources. Le support OSS maintient un inventaire des entités réseaux, provisionne des services, configure les entités et récupère les éléments de supervision de chaque entité réseau.
- **Support commercial (Business Support System)**: gère le déploiement de services à la demande des clients. Il offre ainsi les outils logiciels pour gérer les commandes jusqu'à la mise en paiement des services. Les fournisseurs ne se contentent plus de fournir des services de stockage et de calcul, mais proposent également des VNF tels que des commutateurs, des équilibresurs de charge, des pare-feu...

3. NFV - Network Function Virtualization

□ Relation entre SDN et NFV

La NFV est très complémentaire du SDN, mais n'en dépend pas. La NFV est capable de supporter le SDN en fournissant l'infrastructure sur laquelle le logiciel SDN pourra être exécuté. Ainsi, le contrôleur SDN pourrait être implémenté comme une VNF :

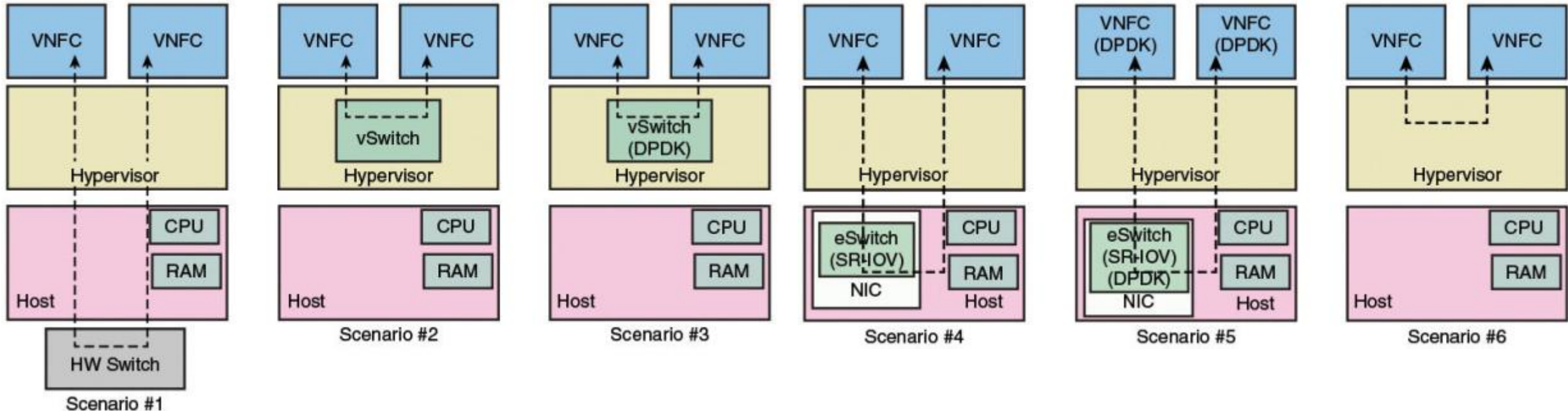
- Pour programmer l'infrastructure réseau virtuel afin de définir les règles de routages et de sous-réseaux pouvant être utilisées pour interconnecter des fonctions réseaux virtualisés (VNF) : SDN Overlay.
- Ex : l'un des avantages du SDN/NFV est le chaînage de service dynamique virtuel (traffic steering and chaining service) défini par une politique de flux. Lorsque l'utilisateur souhaite accéder à un service, le contrôleur SDN définit un ensemble de fonction réseaux à déployer. Le rôle du NFV est alors de gérer les VMs à mettre en œuvre et le contrôleur SDN gère le routage des flux.



❑ Communication VNFC à VNFC

Un VNF apparaît comme un système fonctionnel unique dans le réseau. Toutefois, la connectivité interne entre VNFC au sein d'un même VNF ou entre des VNF co-localisés doit être spécifiée par le fournisseur de VNF.

Le document *Architecture VNF* décrit un certain nombre de modèles de conception d'architecture destinés à fournir les performances et la QoS souhaitées. L'un des plus importants de ces modèles de conception concerne la communication entre les VNFC..

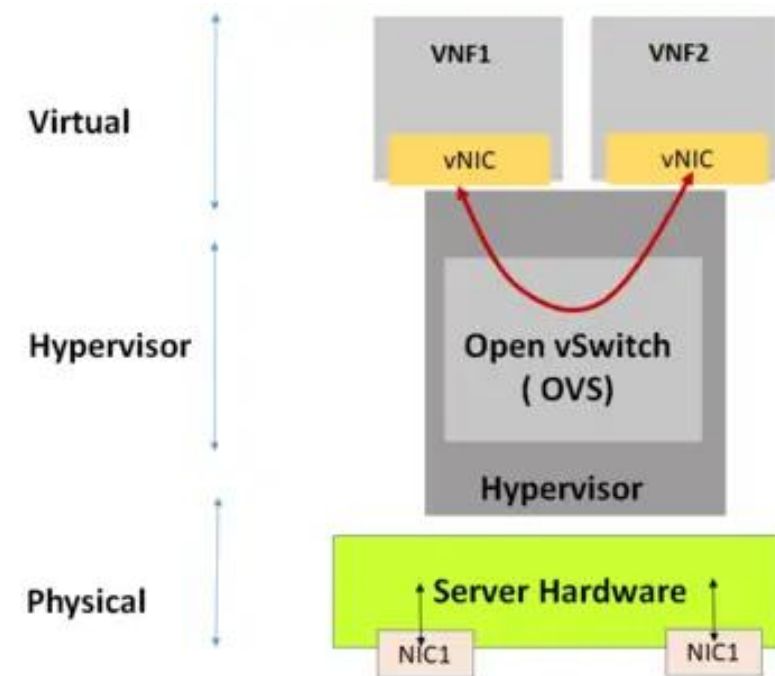


❑ Communication VNFC à VNFC : optimisation

Par défaut, Linux utilise le noyau pour traiter les paquets. Avec l'augmentation de la vitesse des cartes réseau (NIC), le noyau est de plus en plus sollicité pour un traitement encore plus rapide des paquets.

De nombreuses techniques (DMDK, SR-IOV...) ont été mises au point pour contourner le noyau afin d'assurer l'efficacité de traitement des paquets.

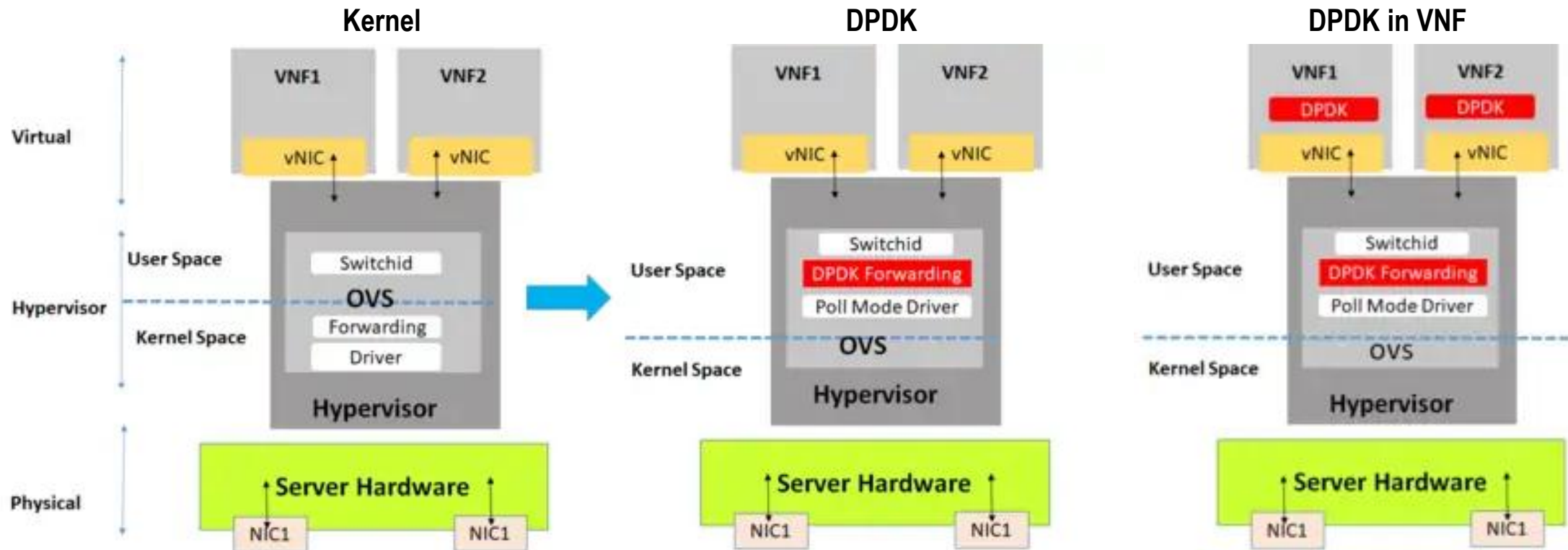
Cependant, bien que ces technologies soient utilisées pour augmenter les performances de traitement des paquets dans les serveurs, la décision quant à la meilleure d'entre elles relève davantage de la conception que des technologies elles-mêmes.



❑ Communication VNFC à VNFC : optimisation

▪ DPDK (Data Plane Development Kit)

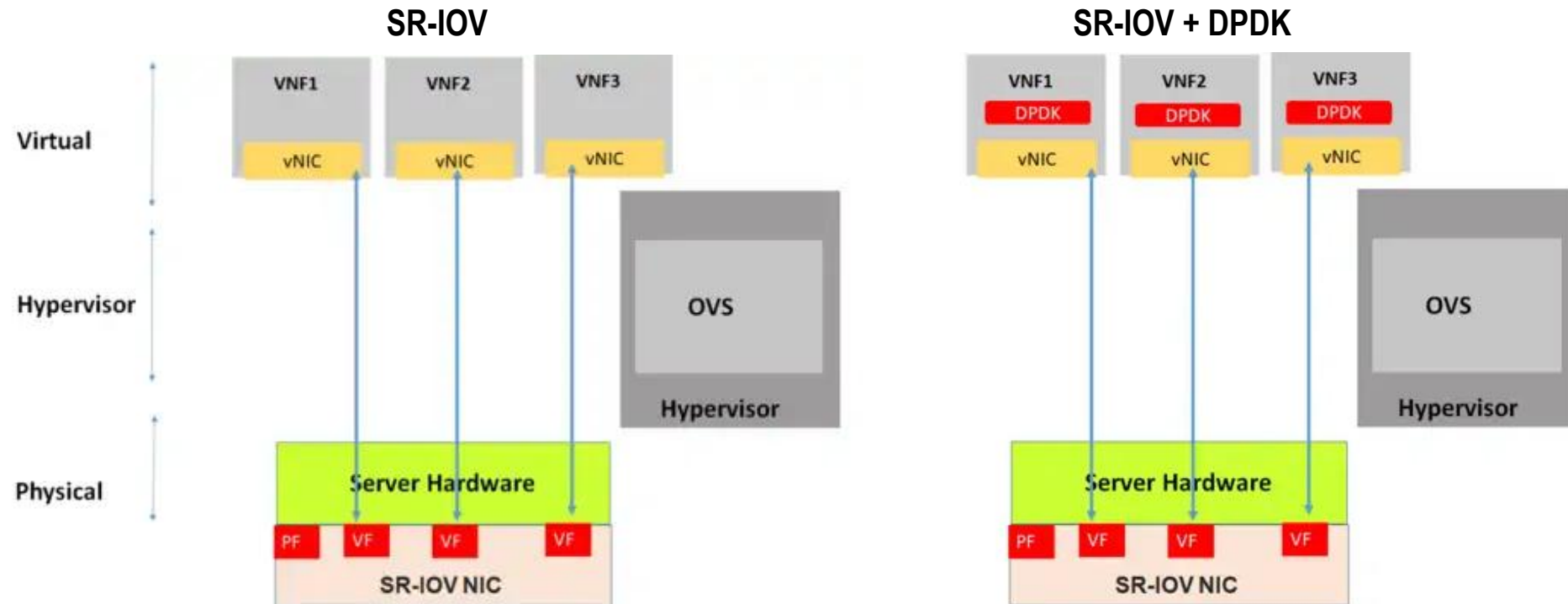
DPDK permet décharger le traitement des paquets du noyau (kernel space) de l'hyperviseur vers les processus s'exécutant dans l'espace utilisateur (user space). Il est constitué d'un ensemble de bibliothèques permettant d'implémenter des pilotes d'espace utilisateur. Il est également possible d'exécuter DPDK dans la VM ou VNF.



❑ Communication VNFC à VNFC : optimisation

▪ SR-IOV (Single Root I/O Virtualization)

SR-IOV définit un mécanisme standardisé permettant de virtualiser une interface Ethernet PCIe pour qu'il apparaisse comme plusieurs interfaces Ethernet. Cela permet à la VM ou VNF d'accéder directement à la carte physique sans passer par l'hyperviseur. On obtient ainsi une meilleure amélioration les performances du réseau par rapport à la virtualisation logicielle. Pour de meilleurs performances, SR-IOV peut être couplé à DPDK.

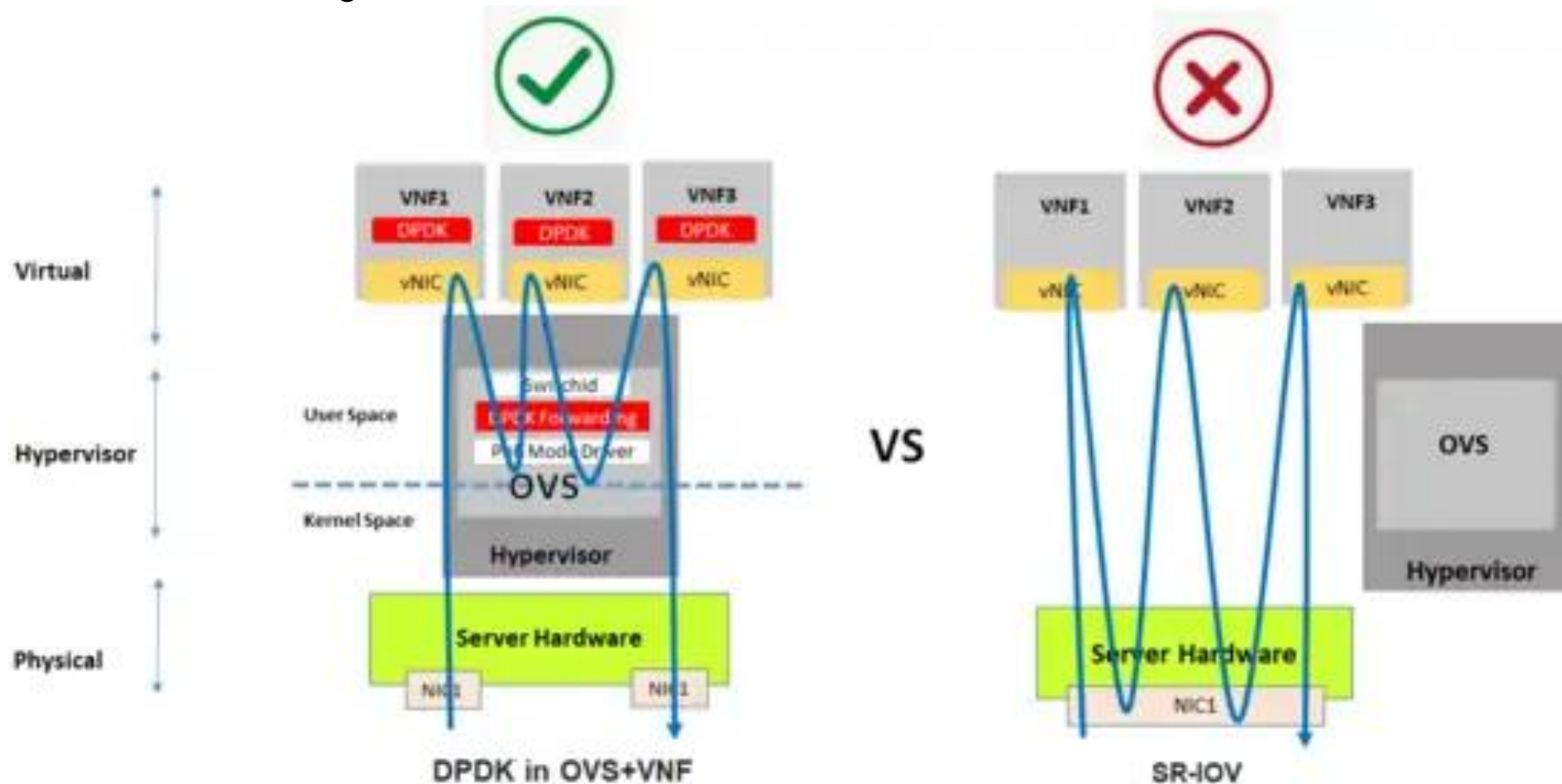


3. NFV - Network Function Virtualization

❑ Communication VNFC à VNFC : optimisation

■ Trafic Est-West

Dans une situation où le trafic Est-Ouest au sein du même serveur, DPDK est plus performant que SR-IOV. La situation est illustrée dans le diagramme ci-dessous.

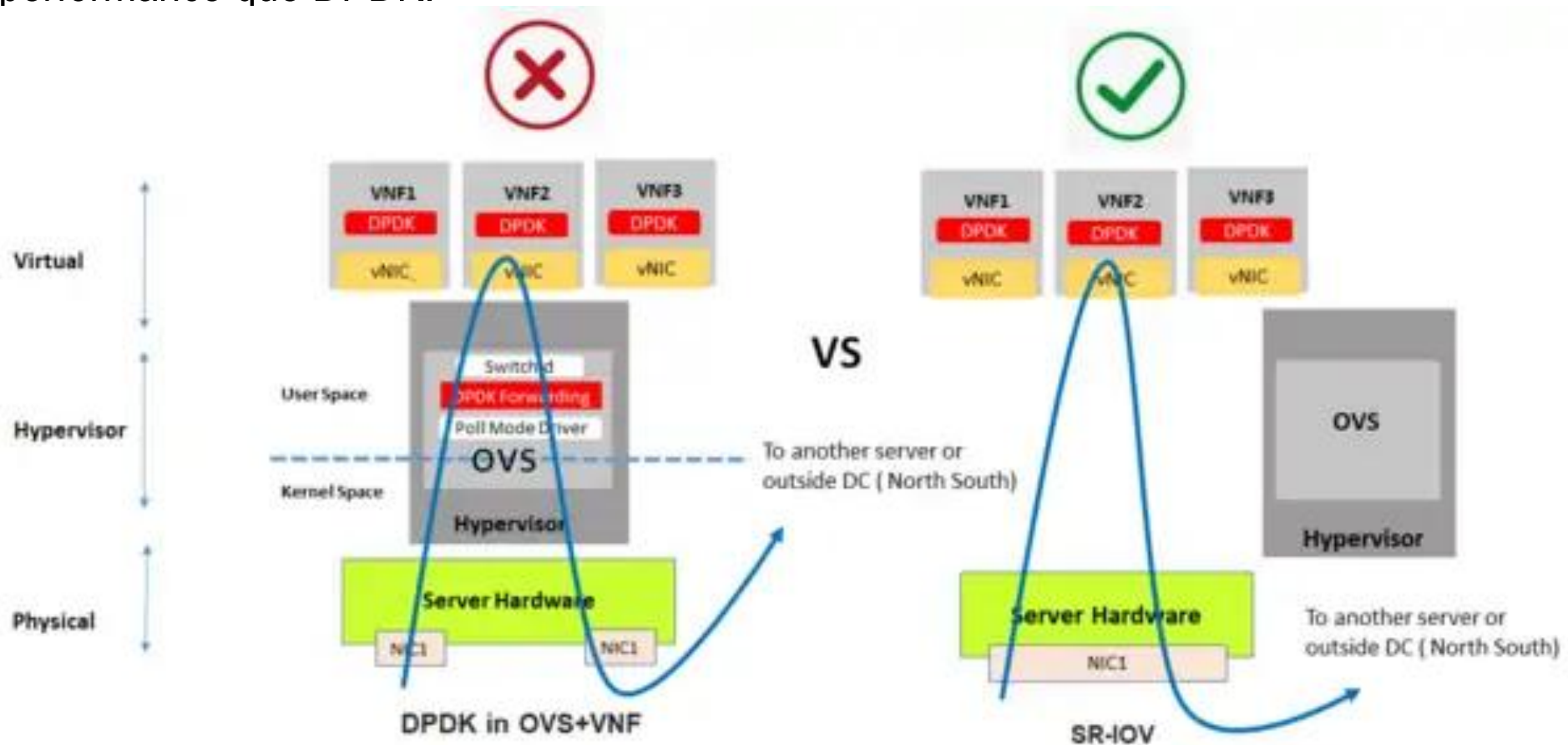


3. NFV - Network Function Virtualization

❑ Communication VNFC à VNFC : optimisation

▪ Trafic North-South

Dans une situation où le trafic est Nord-Sud (y compris le trafic Est-Ouest mais d'un serveur à un autre), SR-IOV devient plus performant que DPDK.

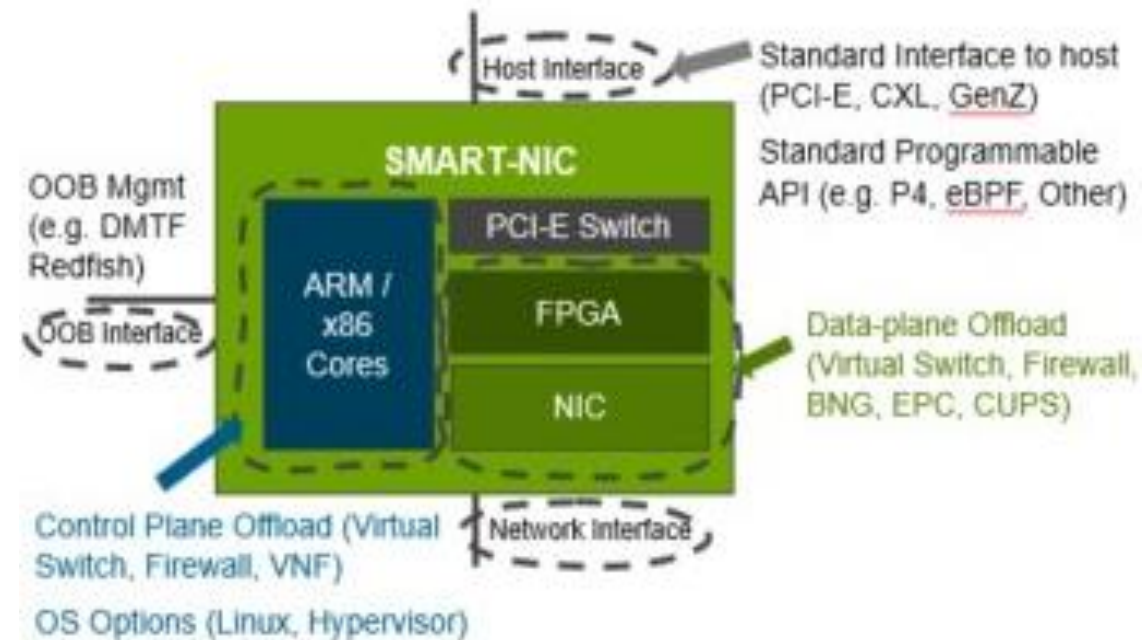


❑ Communication VNFC à VNFC : optimisation

▪ SmartNIC

Une SmartNIC (Smart Network Interface Card) est un accélérateur matériel dédié qui peut être utilisé pour accélérer les fonctions de réseau, de stockage et de sécurité. Elles peuvent également effectuer la virtualisation, l'équilibrage des charges et l'optimisation des chemins de données.

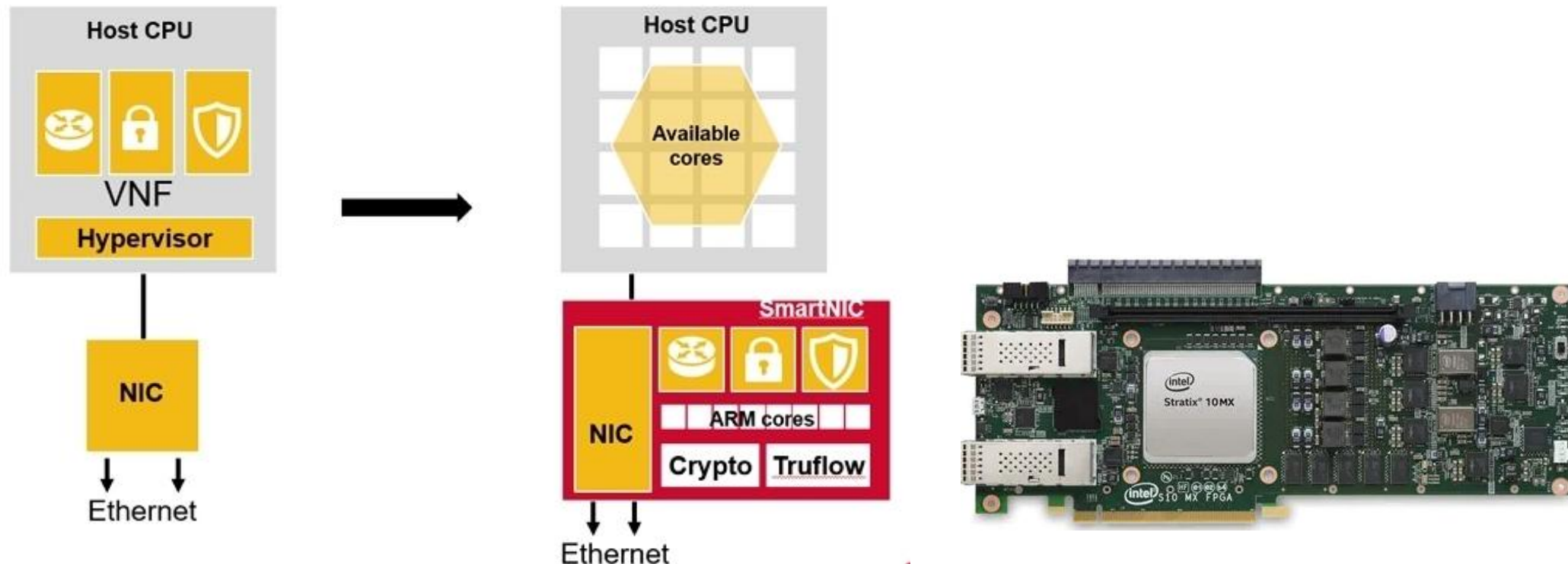
- Les SmartNIC permettent aux opérateurs de tirer davantage de performances du matériel existant, car ils peuvent décharger certaines des fonctions réseau, de stockage et de sécurité de la CPU vers la SmartNIC.
- Une SmartNIC fonctionne comme un serveur à l'intérieur d'un serveur, où la NIC peut communiquer directement avec la VM sans avoir à gérer les interruptions dans la couche du noyau.
- Les protocoles qui peuvent être déchargés de la CPU incluent le VXLAN, NVGRE, Geneve... Et aussi les fonctions réseau suivantes : inspection des paquets, traitement des tables de flux, cryptage, VXLAN overlay...



❑ Communication VNFC à VNFC : optimisation

▪ SmartNIC

Les SmartNIC, avec leurs cœurs ARM haute performance et leurs accélérateurs matériels pour la mise en réseau, constituent une solution idéale pour décharger les VNF de l'hôte. Les VNF s'exécutant sur la SmartNIC libéreront des cœurs sur la CPU de l'hôte et offriront des performances réseau inégalées.



□ Conteneurisation

La conteneurisation est une méthode qui permet de virtualiser les ressources matérielles (systèmes de fichiers, réseau, processeur, mémoire RAM...) nécessaires à l'exécution d'une application.

- **Conteneur** : processus auquel on donne une certaine vision du système de façon qu'il ait l'impression d'être le seul processus du système sur lequel il tourne. Ainsi un conteneur a :

Javascript code

Node_modules

Node.js runtime

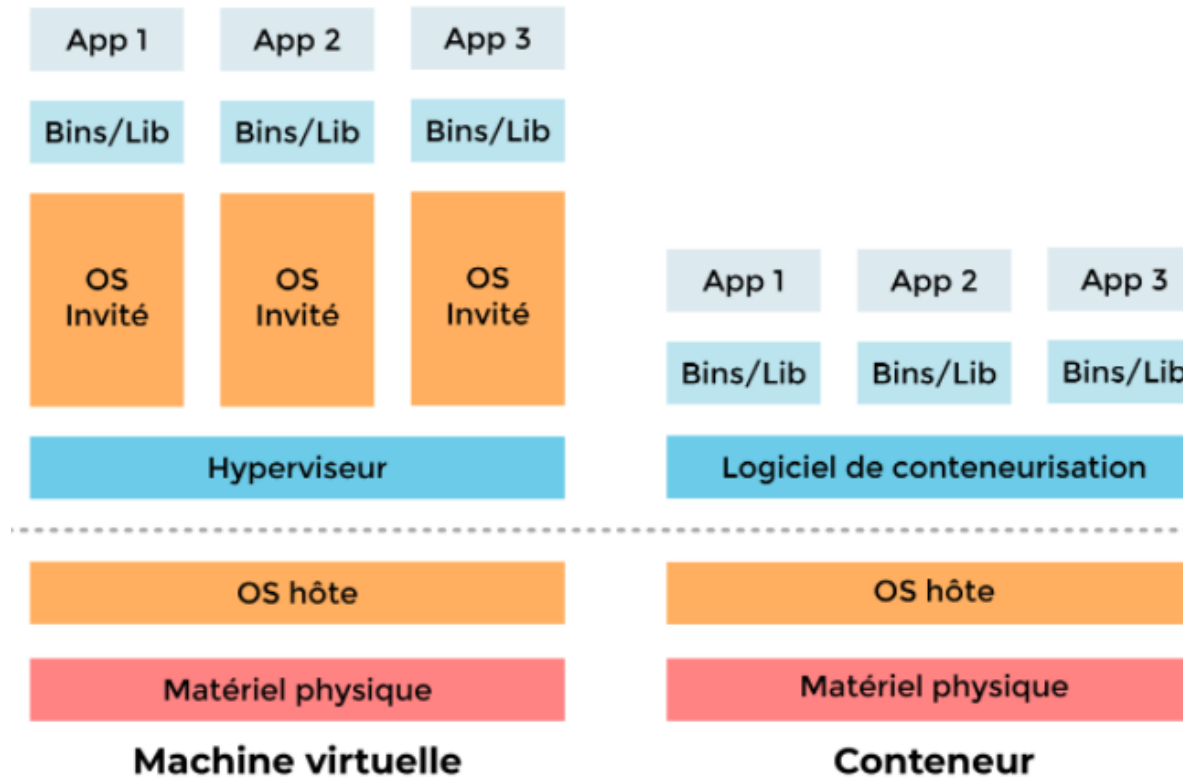
Debian

Exemple de rootfs

- Son propre système de fichier : on donne au conteneur son propre rootfs (système de fichier Linux)
 - JeOS (Just Enough Operating System) prononcer "Juice" : système d'exploitation Linux léger et hautement personnalisé pour une application spécifique.
 - Librairies et binaires
 - Dépendances applicatives
 - Environnement d'exécution
 - Code applicatif
- Son propre stack réseau
- Vision limité du système

□ Conteneurisation vs Virtualisation

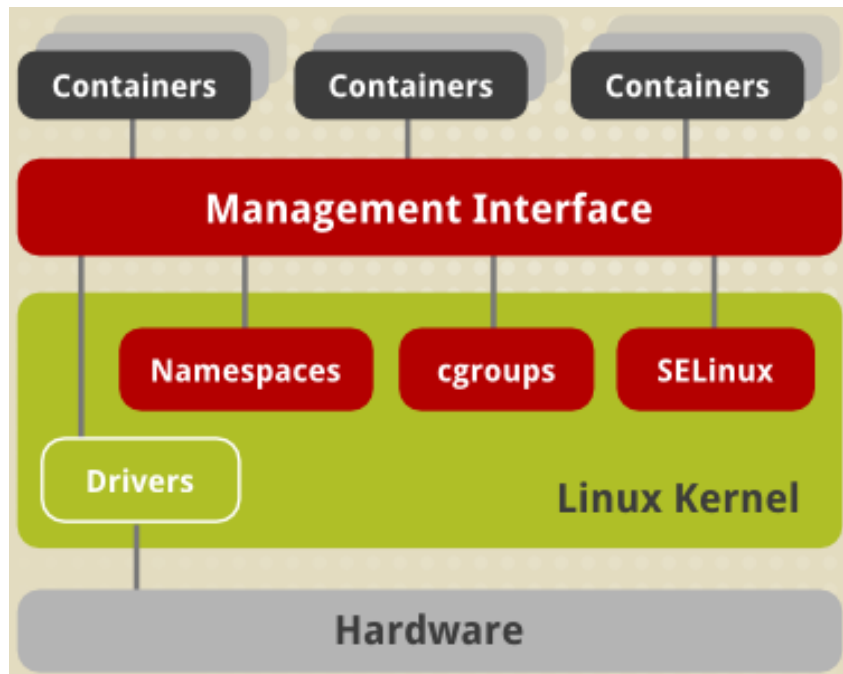
- **Virtualisation** : l'isolation des VMs se fait au niveau matériel (CPU, RAM, disque) avec un accès virtuel aux ressources de l'hôte via un hyperviseur.
- **Conteneurisation** : l'isolation se fait au niveau de l'OS et va partager le noyau de la machine hôte avec d'autres conteneurs. Ne prenant pas plus de mémoire que tout autre exécutable, ce qui le rend léger.



3. NFV - Network Function Virtualization

❑ Conteneurisation : conteneur Linux

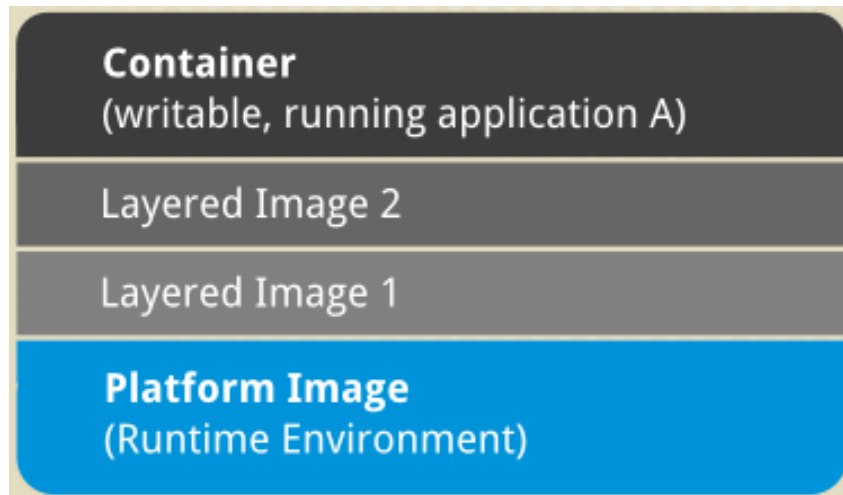
Plusieurs composants sont nécessaires pour que les conteneurs fonctionnent correctement, la plupart d'entre eux sont fournis par le noyau Linux.



- **Namespaces** : assure l'isolation des processus en créant des espaces de noms distincts pour les conteneurs. Il existe plusieurs types d'espaces de noms : PID, User, Mount, IPC, Net...
- **Control Groups (cgroups)** : limitation sur l'utilisation des ressources en permettant de spécifier ce que peut faire un conteneur. Les cgroups allouent le temps processeur, la mémoire système, la bande passante...
- **SELinux** : fournit une séparation sécurisée des conteneurs en appliquant la politique et les labels SELinux.
- **Management Interface** : forme une couche supérieure qui interagit avec les composants du noyau susmentionnés et fournit des outils pour la construction et la gestion des conteneurs.

❑ Conteneurisation : images

Les conteneurs basés sur des images conditionnent les applications avec des piles d'exécution individuelles, ce qui rend les conteneurs résultants indépendants du système d'exploitation hôte.

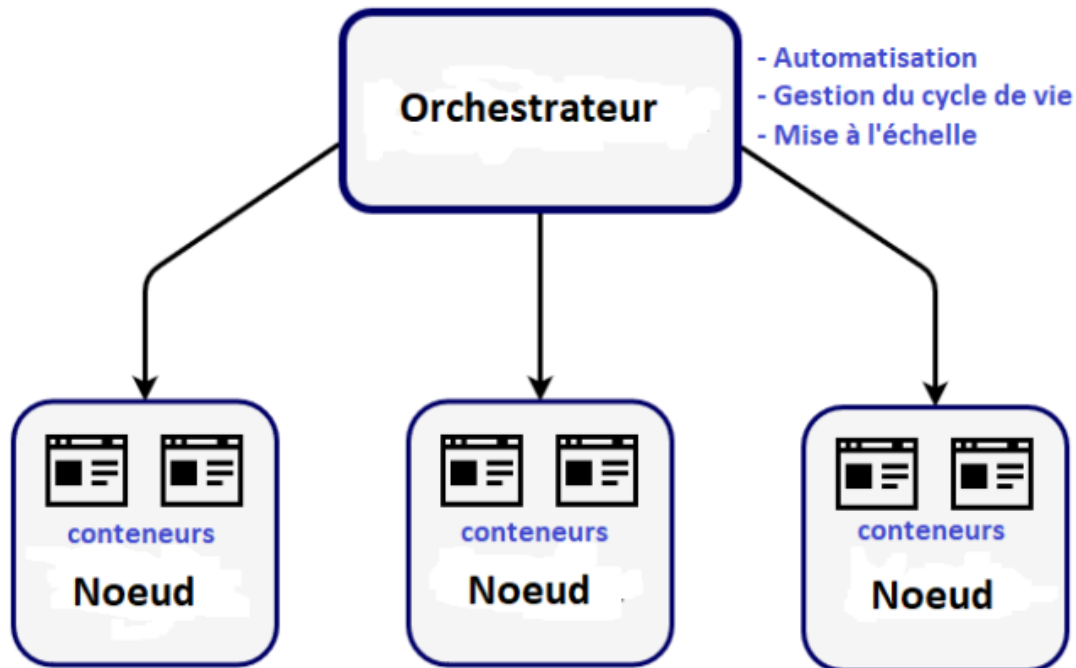


- **Image** : un snapshot de la configuration des conteneurs. L'image est une couche en lecture seule (*immutable*). Toutes les modifications sont effectuées sur la couche inscriptible la plus élevée (*volatile*). Chaque image dépend d'une ou plusieurs images parentes.
- **Platform Image** : une image qui n'a pas de parent. Les images de plate-forme définissent l'environnement d'exécution, les paquets et les utilitaires nécessaires à l'exécution de l'application conteneurisée.
- **Conteneur** : est basé sur une image qui contient les données de configuration nécessaires. Lorsque vous lancez un conteneur à partir d'une image, une couche inscriptible est ajoutée par-dessus cette image.

3. NFV - Network Function Virtualization

❑ Orchestration

Il devient très difficile de gérer le cycle de vie du conteneur et sa gestion lorsque le nombre augmente dynamiquement avec la demande. L'orchestration de conteneurs résout le problème en automatisant la planification, le déploiement, l'évolutivité, l'équilibrage de charge, la disponibilité et la mise en réseau des conteneurs. La gestion du cycle de vie des conteneurs avec l'orchestration aide également les équipes DevOps qui l'intègrent dans les workflows CI/CD.



- Provisionnement et déploiement
- Configuration et planification
- Allocation des ressources
- Disponibilité des conteneurs
- Mise à l'échelle (scaling-up) ou suppression de conteneurs (scaling-down)
- Équilibrage de la charge et routage du trafic
- Surveillance de l'intégrité des conteneurs

❑ Applications Cloud Native

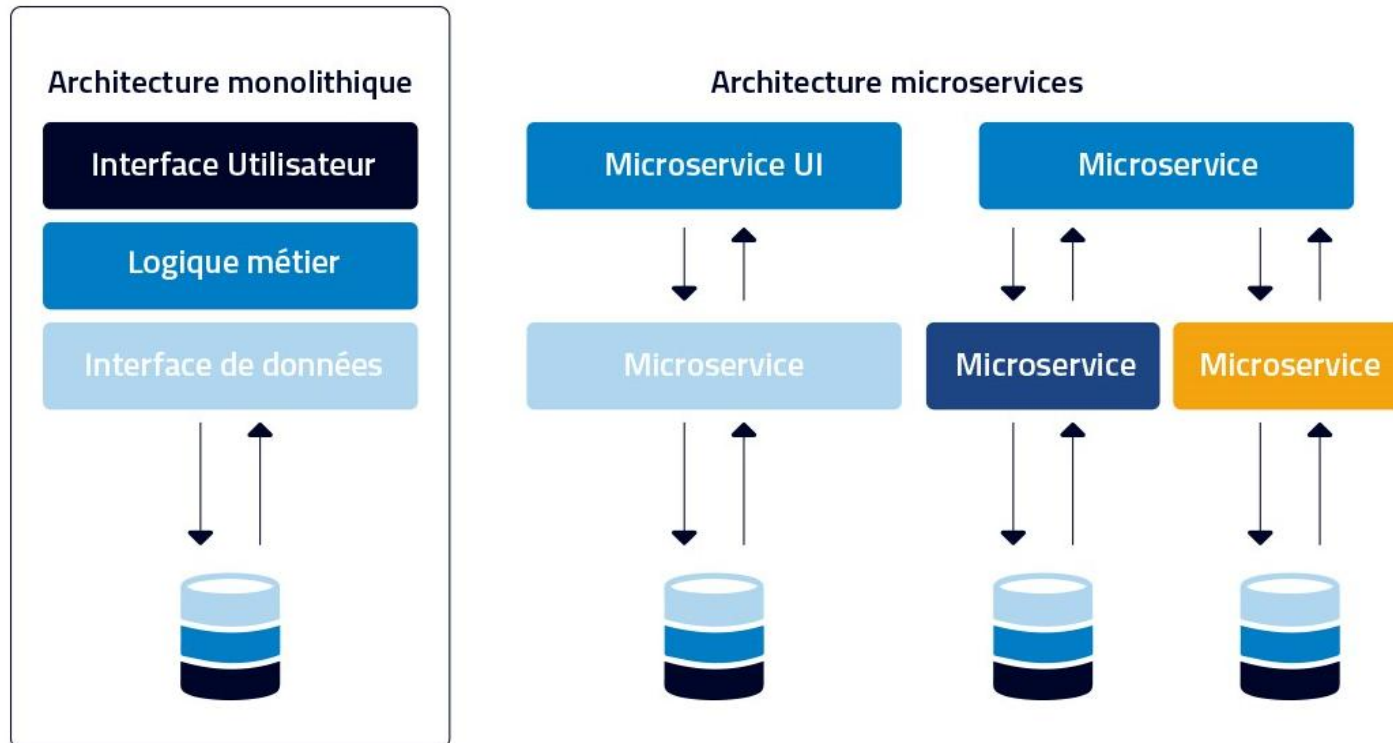
Lorsque l'on dit d'une application qu'elle est « cloud-native », cela signifie qu'elle a été conçue spécialement pour offrir une expérience cohérente de développement et de gestion automatisée dans le Cloud.

	Applications traditionnelles	Applications cloud natives
Axe	Longévité et stabilité	Commercialisation rapide
Méthode de Développement	En cascade, semi-agile	Agile, pratiques DevOps
Equipes	Cloisonnement des équipes de développement, d'exploitation, de contrôle qualité et de sécurité	Collaboration des équipes DevOps
Cycles de Distribution	Longs	Courts et continus
Architecture d'applications	Couplage fort Monolithique	Couplage faible Basée sur des services; utilisation d'API
Infrastructures	Basée sur des serveurs Conçue pour les déploiements sur site Crée une dépendance Mise à l'échelle verticale Capacité maximale pré approvisionnée	Basée sur des conteneurs Conçue pour les déploiements sur site et dans le cloud Portabilité des applications Mise à l'échelle horizontale Capacité à la demande

3. NFV - Network Function Virtualization

❑ Microservices

Une architecture de microservices est un nouveau style d'architecture dans laquelle une application est conçue à partir de petites unités autonomes, appelées microservices, qui interagissent via des API indépendantes du langage de programmation utilisé.

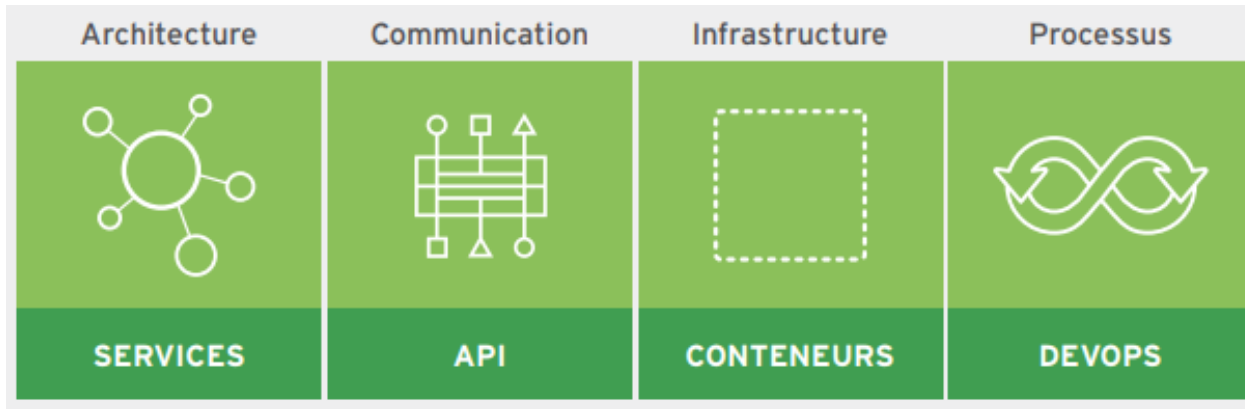


- Chaque service a une portée limitée, se concentre sur une seule tâche et est totalement indépendant.
- Une architecture de microservices se différencie d'une approche monolithique classique par le fait qu'elle décompose une application pour en isoler les fonctions clés.

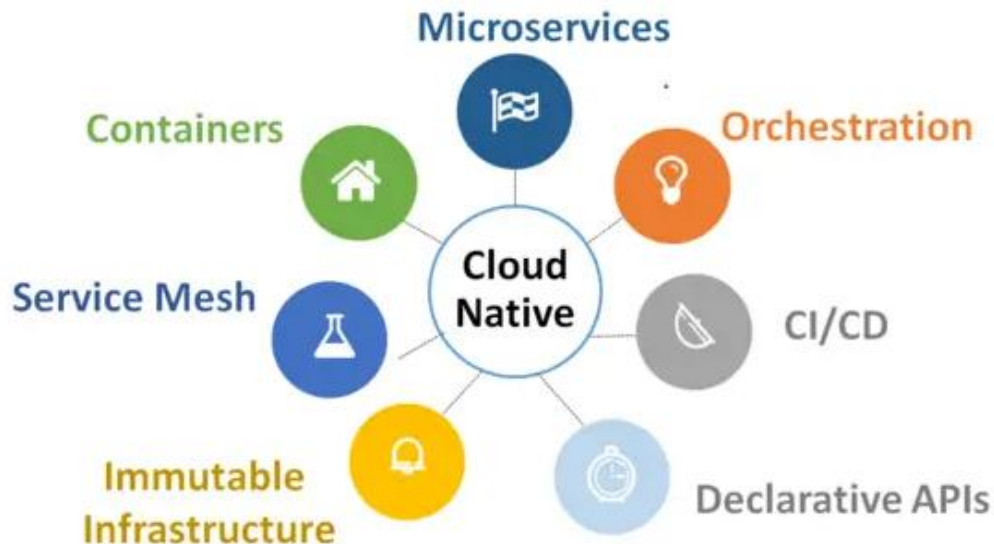
3. NFV - Network Function Virtualization

☐ Applications Cloud Native

Une application cloud native repose sur plusieurs notions :



- Les microservices qui constituent autant de briques de l'application cloud native.
- Ils communiquent grâce à un système d'API.
- Les conteneurs qui améliorent la cohérence de l'environnement et en accélèrent le déploiement et l'évolutivité.
- La culture DevOps qui permet la collaboration entre développeurs de logiciels et administrateurs système.

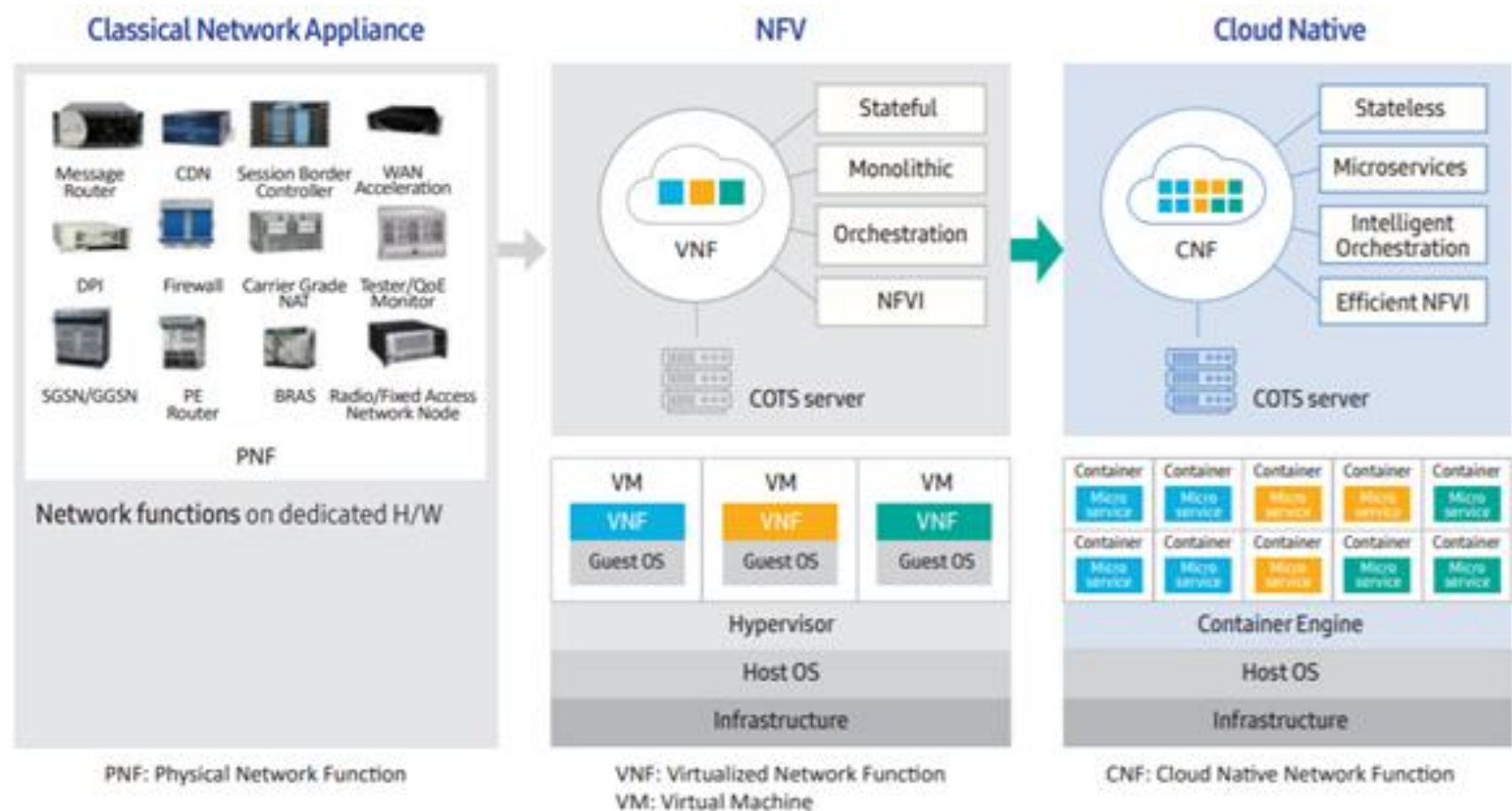


❑ Cloud Native Network Function (CNF)

Les anciens VNF sont monolithiques avec un code volumineux et fonctionnent dans certains cas comme une seule VM. La création de VNFC via l'approche Cloud Native suit les étapes suivantes :

(1) Fractionnement des VNF en microservices VNFC

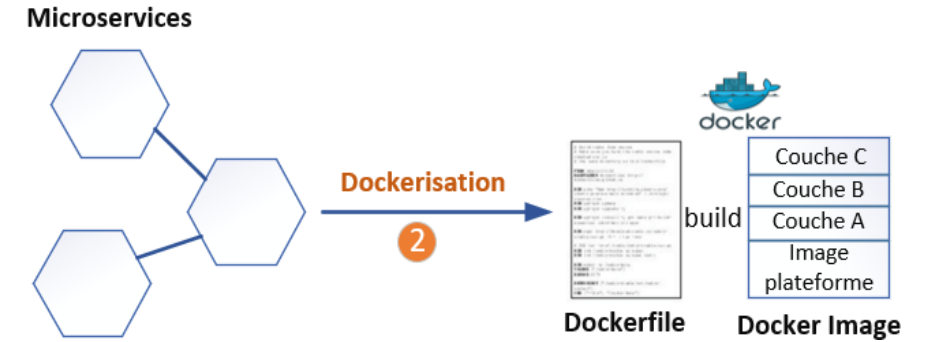
Les VNF peuvent être assez complexes, comme la 5G Core, et aucun opérateur ne pourra déployer un VNF "tout en un" sur une seule VM. Ainsi, il s'agira de créer plusieurs petits VNFC, également appelés microservices, qui seront déployés et formeront ensemble une fonctionnalité du VNF.



❑ Cloud Native Network Function (CNF)

(2) Dockersisation des VNFC

Ce processus consiste à transformer les microservices en images Docker. Plus concrètement, cela consiste à définir dans un fichier nommé DOCKERFILE les restrictions qui vont permettre de créer une image à partir du code source du VNFC.

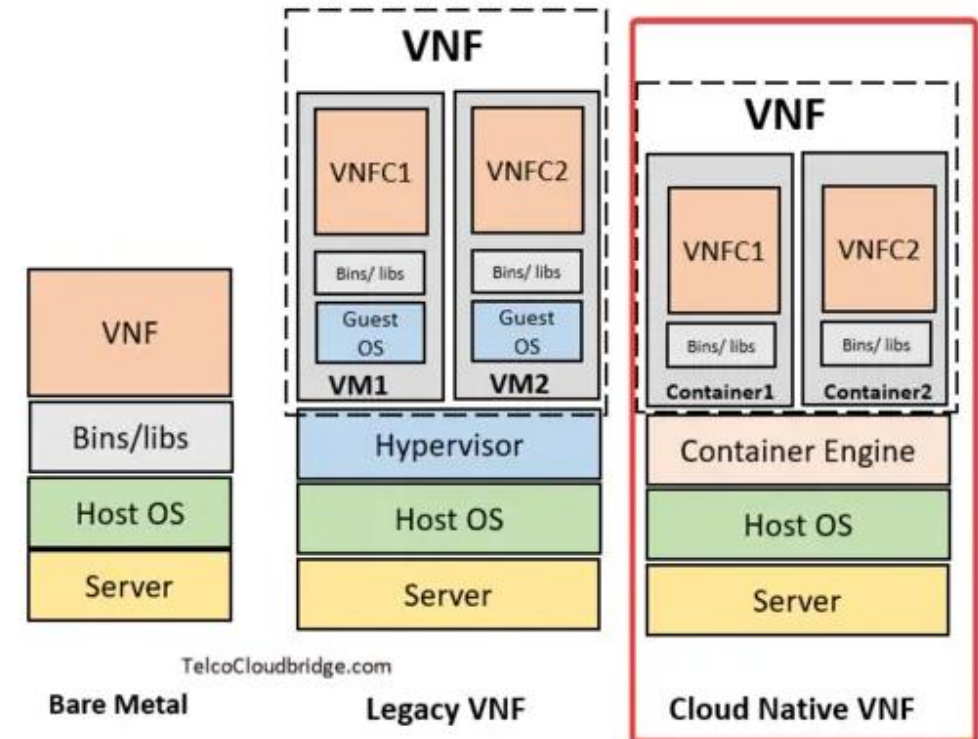


(3) Conteneurisation des VNFC

Il est possible d'exécuter, à partir de leurs images, chaque VNFC comme un conteneur directement sur le système d'exploitation hôte. Un moteur de conteneur (Docker, containerd) sera utilisé pour faire tourner les conteneurs.

(4) Orchestration des VNFC

Permet de gérer le cycle de vie des conteneurs ainsi que leurs mise en réseau pour permettre la communication entre VNFC et autres microservices.



3. NFV - Network Function Virtualization

❑ Projets NVF

Avec la tendance croissante à la softwarisation et à la virtualisation des fonctions et systèmes de réseau, des solutions de gestion et d'orchestration NFV (MANO) sont en cours de développement pour répondre aux exigences de gestion agile et flexible des services de réseau virtualisés à l'ère de la 5G et au-delà.

A cet égard, l'ETSI ISG NFV a spécifié un système MANO standard qui est utilisé comme référence par les fournisseurs ainsi que par les projets MANO open source.

Nous présenterons dans la suite 3 projets MANO :



❑ Open Source MANO (OSM)

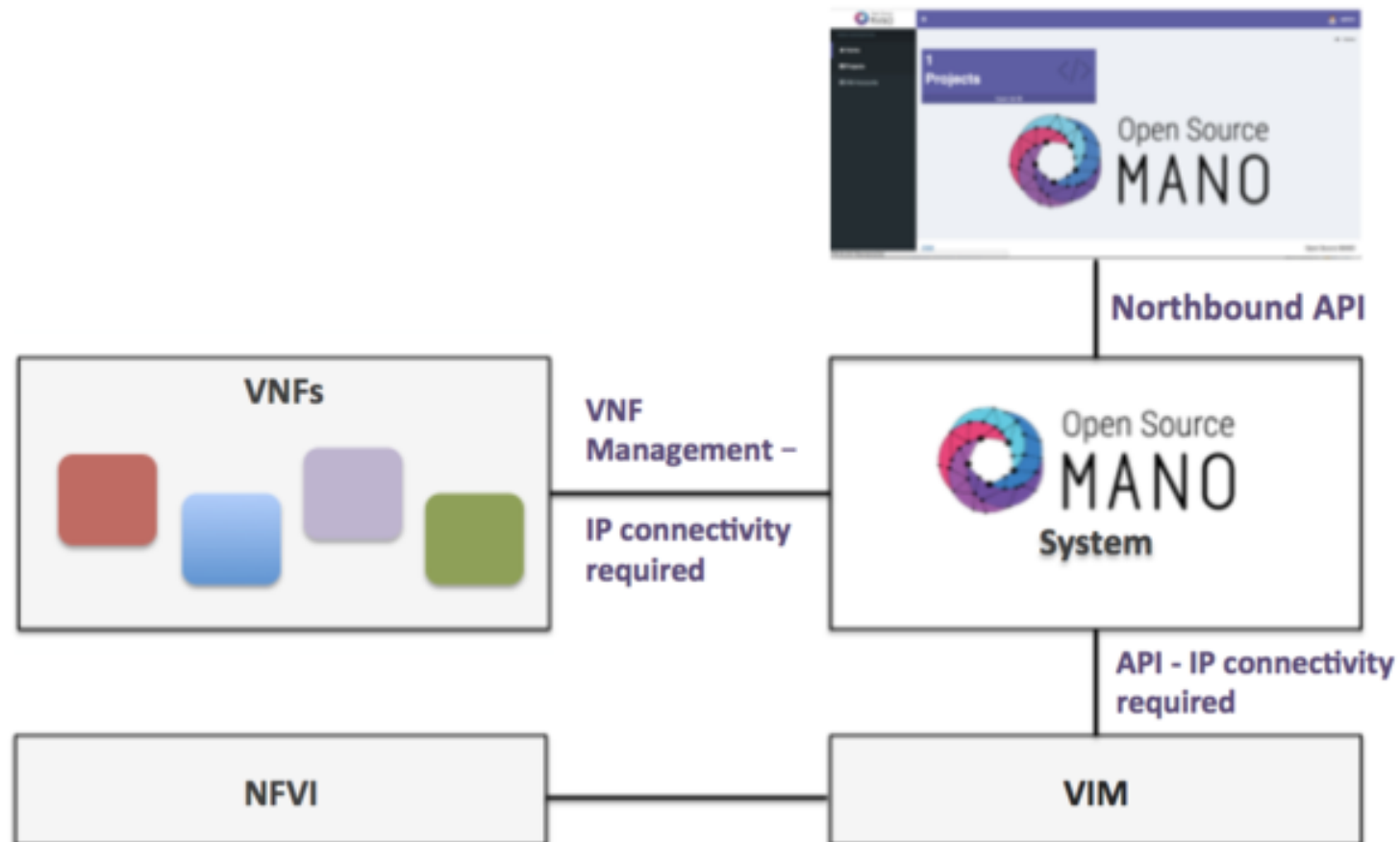
OSM est une implémentation open-source de la pile MANO de l'ETSI NFV. Il a été fondé par l'ETSI et est entièrement alignée sur les modèles d'information ETSI NFV. OSM est de qualité production qui répond aux exigences des fournisseurs de services de télécommunications pour les déploiements NFV commerciaux. Au fur des releases, OSM est doté d'un ensemble de nouvelles fonctionnalités :

- Slicing réseau pour la 5G
- Prise en charge des VNF, PNF et HNF (Hybrid NF)
- Cloud Native Services : fonctions réseau basées sur Kubernetes (KNF) ou des fonctions réseau conteneurisées (CNF) dans un cluster Kubernetes. Orchestration du cycle de vie des KNF.
- Interopérabilité avec les clouds publics/privés : AWS, MS Azure, Google Cloud, OpenStack
- Améliorations du framework de monitoring et de politique fournissant des opérations en boucle fermée.



❑ Open Source MANO (OSM) : architecture

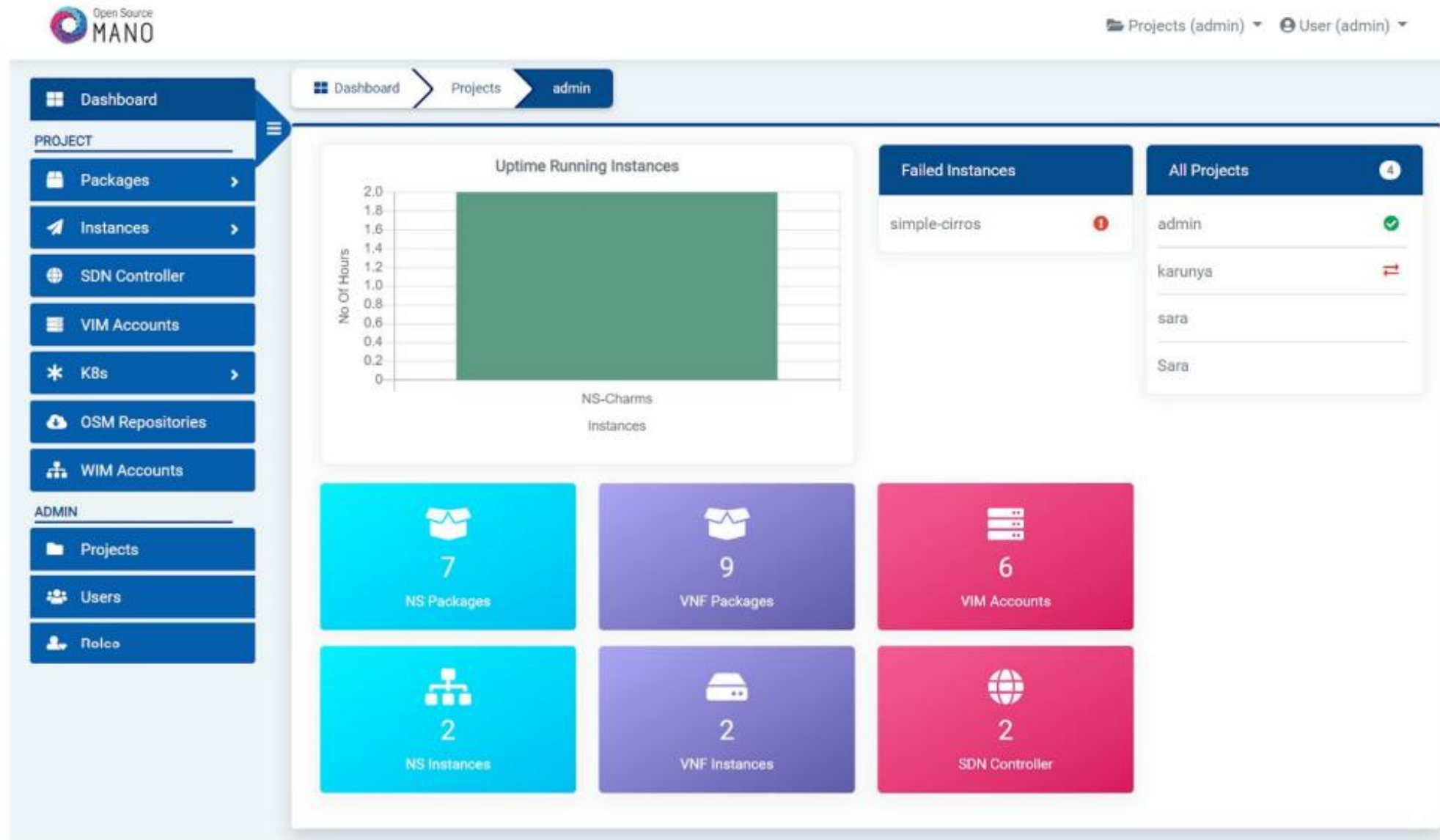
- OSM communique avec le VIM pour le déploiement des VNF et des VL (Virtual Link) qui les connectent.
- OSM communique avec les VNFs déployés dans un VIM pour exécuter les primitives de configuration.





3. NFV - Network Function Virtualization

❑ Open source MANO (OSM) : interface d'administration



❑ Charmed OSM

Charmed OSM est une distribution basée sur OSM, développée et maintenue par Canonical, qui utilise charmed Juju pour simplifier les déploiements et les opérations. Charmed OSM permet aux opérateurs télécoms de déployer facilement OSM en amont dans des clusters hautement disponibles, de qualité production et évolutifs.

- Distribution OSM pure *upstream* : les opérateurs télécoms sont ainsi assurés d'un rythme prévisible de publications et de mises à jour.
- S'inscrit dans le programme de support *Ubuntu Advantage* : mises à jour et correctifs de sécurité critiques, assistance 24/7, accord de niveau de service (SLA)...
- Distribution OSM stable et sécurisée, conçue pour accompagner l'évolution des opérateurs vers la 5G.
- En tant que collection Juju charms, Charmed OSM peut être facilement intégré à d'autres applications charms. Cela permet une coopération native entre Charmed OSM et d'autres applications et se traduit par un gain de temps, et donc par une réduction des coûts opérationnels.

❑ Charmed OSM : mode de déploiement

Charmed OSM utilise Juju comme gestionnaire générique de fonctions réseau, MikroK8s comme cluster Kubernetes pour accueillir l'installation d'OSM et MicroStack pour déployer un cluster OpenStack et l'utiliser comme VIM.



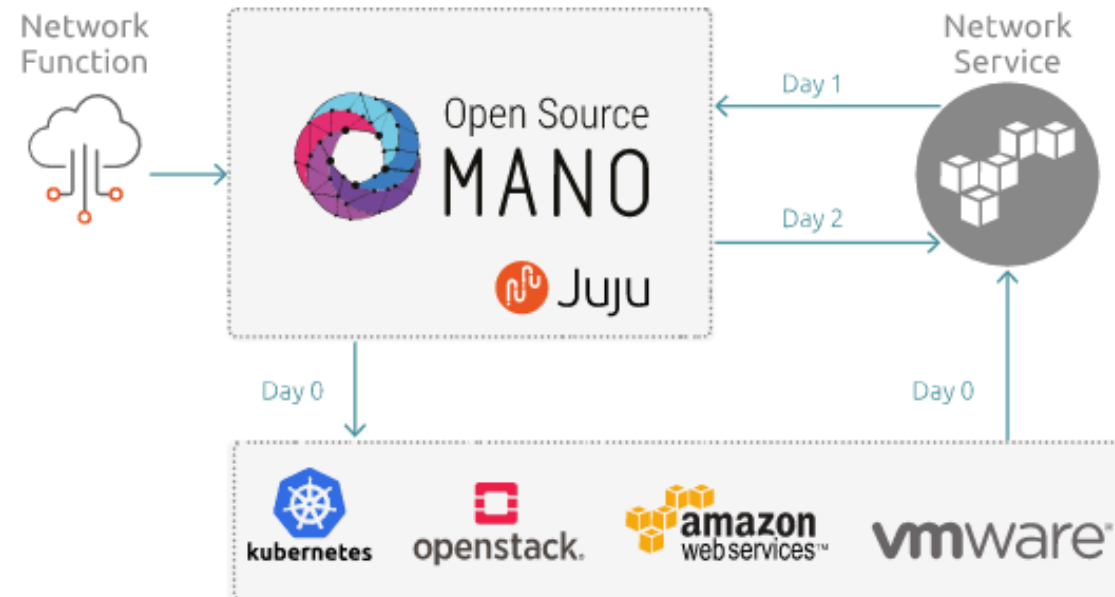
■ Juju

C'est un framework open source qui utilise les Charmed Operators, ou Charm, pour déployer des infrastructures et des applications Cloud et gérer leurs opérations. Juju est utilisé pour installer, maintenir, mettre à niveau et intégrer des applications sur des clusters Kubernetes, des conteneurs, des machines virtuelles et des machines physiques, sur des Clouds publics ou privés.

- **Charm** : c'est une expansion et une généralisation de la notion d'opérateur dans Kubernetes. C'est un conteneur qui pilote la gestion du cycle de vie, la configuration, l'intégration et les actions d'une application.

❑ Charmed OSM : intégration des fonctions réseau

Charmed OSM prend en charge l'intégration des VNF, des CNF et des PNF : ce qui réduit la complexité des opérations au *Day 0* et *Day 1*.

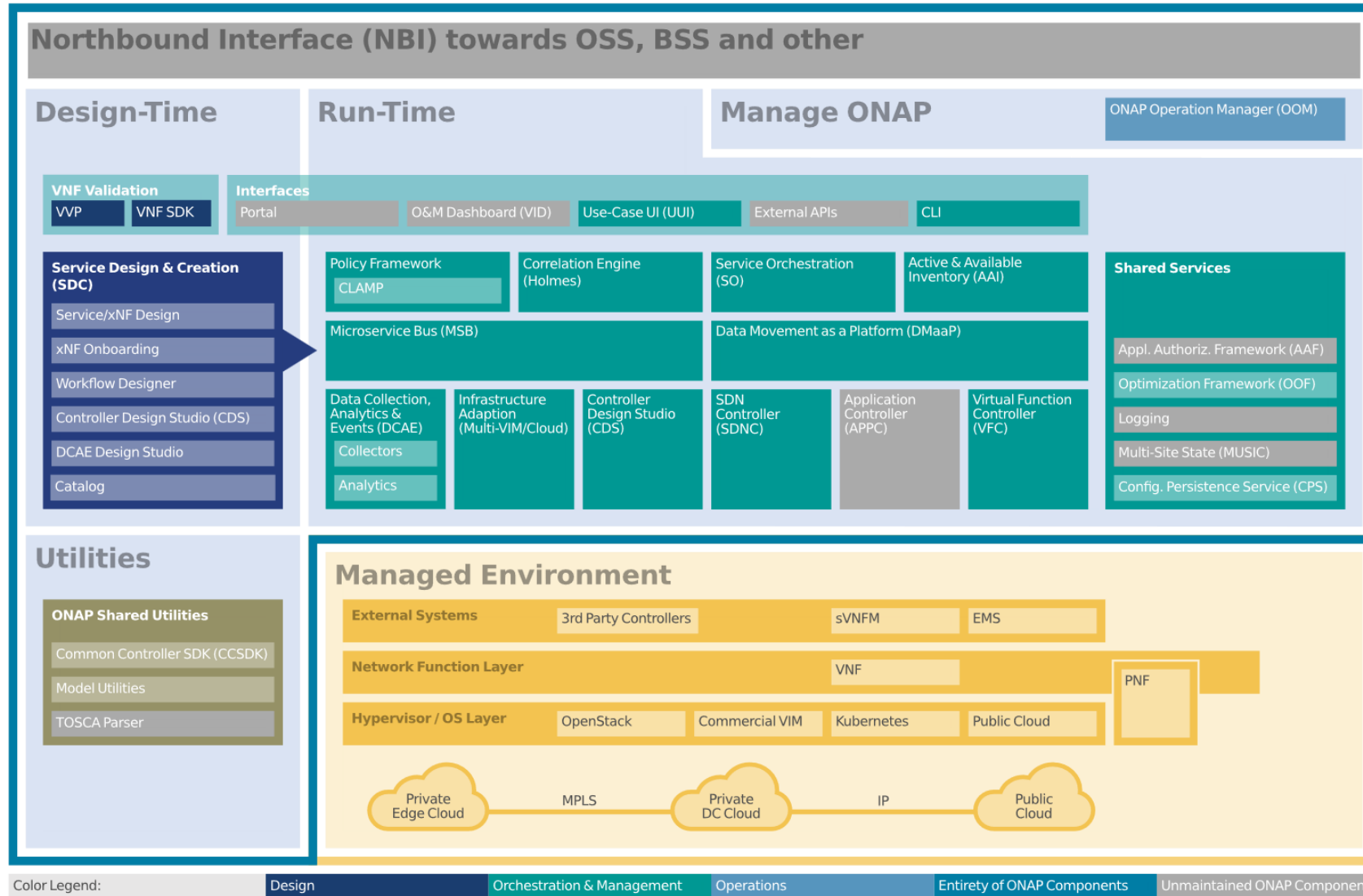


❑ Open Network Automation Platform (ONAP)

ONAP est un projet open source placé sous la gouvernance de la Fondation Linux et fondé par AT&T et China Mobile. ONAP implémente une plateforme complète d'orchestration, de gestion et d'automatisation des services de réseau et de edge computing pour les opérateurs de réseau, les fournisseurs de cloud et les entreprises. ONAP est basé sur les politiques des PNF et VNF permettant ainsi une automatisation rapide des nouveaux services et une gestion complète du cycle de vie, critique pour les réseaux 5G et de nouvelle génération. Les fonctionnalités de base :

- Introduire dynamiquement l'orchestration du cycle de vie complet
- Approche Cloud Native
- Fonctions d'optimisation SON (Self-Organizing Network) et de slicing réseau pour la 5G RAN
- Prise en charge de : 5G,
- Framework multicloud : MS Azure, OpenStack, Vmware VIO, Kubernetes

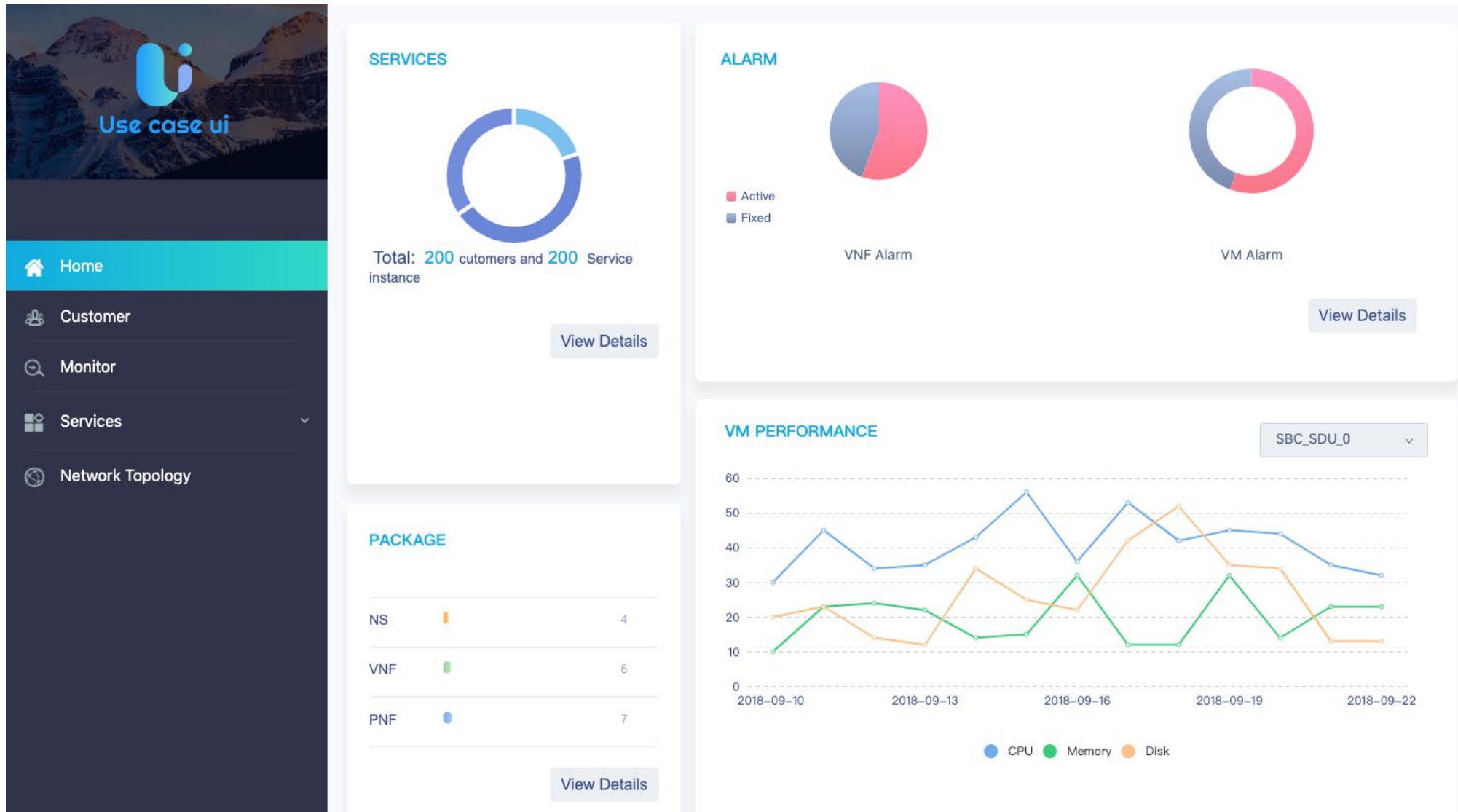
❑ Open Network Automation Platform (ONAP) : architecture





3. NFV - Network Function Virtualization

❑ Open Network Automation Platform (ONAP) : interface d'administration



❏ Open Baton

Open Baton est un framework open source extensible et personnalisable conforme à la norme NFV MANO. Open Baton a considérablement augmenté la liste des fonctionnalités fournies.

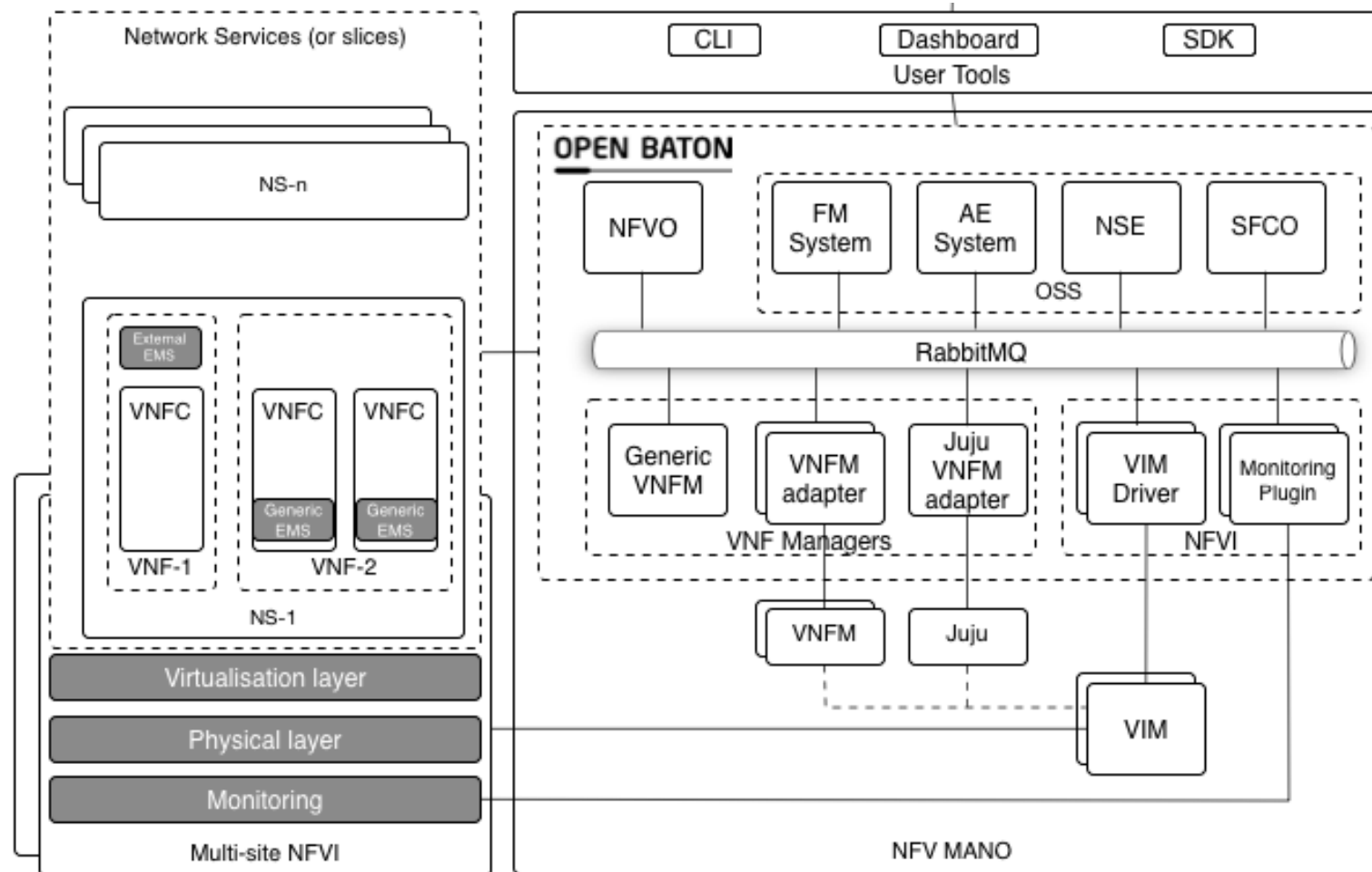
- Peut gérer un écosystème diversifié de VNF, par l'intermédiaire de son EMS générique et de son VNFM générique, en les composant à l'exécution dans tout type de services de réseau.
- S'intègre aux VNFM existants via un modèle plug-and-play, en exposant des API AMQP et RESTful, ainsi que des SDK dans différents langages de programmation (Java, Python, Go).
- Gère un NFVI multisite prenant en charge des technologies hétérogènes de virtualisation et de cloud : OpenStack, AWS, Docker.
- Prend en charge le slicing réseau, au niveau de l'infrastructure en utilisant les technologies SDN pour assurer l'isolation entre plusieurs services réseau partageant les mêmes ressources physiques.
- Prend en charge plusieurs VNF telles que les VNFs OpenIMSCore et Clearwater IMS.

OPEN BATON

3. NFV - Network Function Virtualization

❑ Open Baton : architecture

Open Baton comprend un ensemble cohérent de fonctionnalités permettant le déploiement de VNF à travers des infrastructures NFV, conformément à la spécification MANO de l'ETSI.





❑ Open Baton : interface d'administration

