

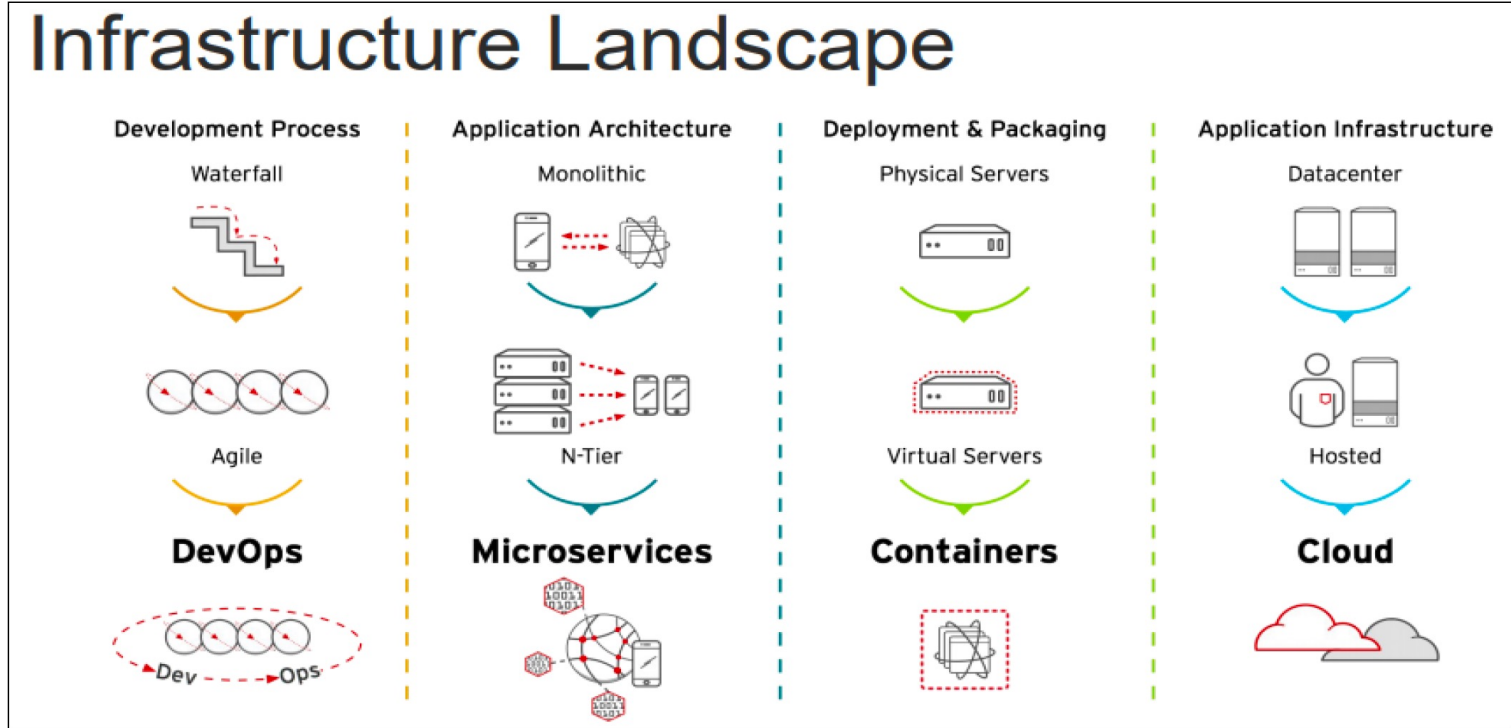
# Orchestrating Container Network Functions with Cisco NSO

Håkan Niska

Solutions Architect - EMEAR

2022/05/10

# Evolution of Applications development



# Network Functions Evolutions

A hybrid reality in real deployment environments

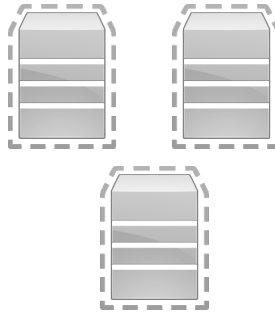
Physical

PNF



Virtual

VNF



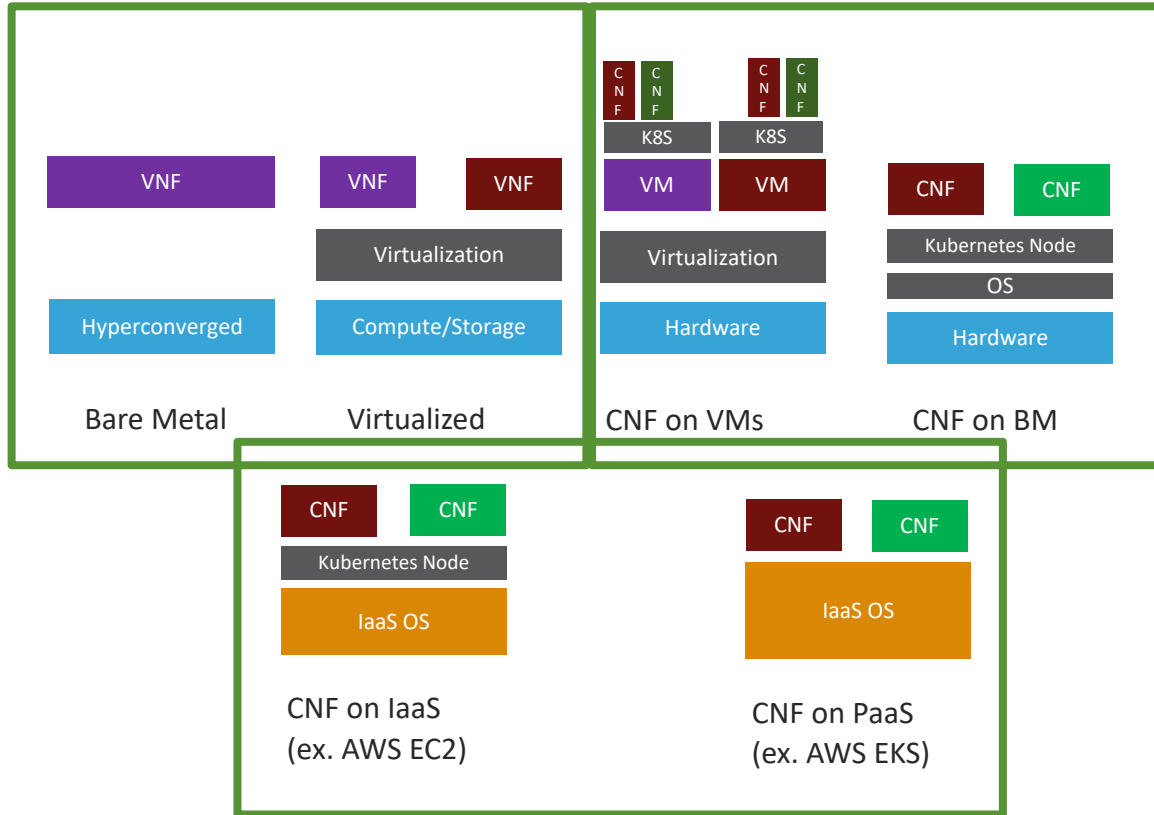
Containers

CNF



# Deployment Scenarios for VNF:s and CNF:s

In most cases Kubernetes is the chosen CNF infrastructure



- Some workloads may be best suited for specific scenarios
- An orchestration solution should be open for all different deployment solutions
- Customers may not have control over what deployment scenario to choose

# Not all CNFs are equal in adopting “cloud native” Kubernetes plugins, management and APIs

- Some CNFs are simple, single POD and some are complex applications running all Kubernetes features
- Some CNFs require dedicated Kubernetes controllers to manage its deployment and lifecycle
- Some CNFs require a particular CNI
- Many CNFs require stateful configuration lifecycle management via APIs
- Some CNFs may require management applications (Element Managers)

# Cisco NSO strategy for CNF Orchestration

## Long live Cisco's NFVO and VNFM

- Success Stories
  - Use Cases: Business Services, Packet core, ORAN, IT, etc.
  - ETSI compliancy:
    - SOL001, SOL003, SOL004, SOL005
  - Successful ETSI Plugtests since its conception
- Technologies incubated:
  - NSO Nano Services
  - NSO Kickers and Plans
  - NSO Resource Manager

### **Cisco MANO Differentiation:**

- Truly Generic VNFM and NFVO
- Integration with VNF configuration, DC and DCI environments for end-to-end automation and life-cycle management

# NSO evolution to CNF Orchestration objectives

## Keeping what NSO does best

1. Expand NSO resource life-cycle management to CNFs:
  - Adopt Helm for package management
  - Focus on Kubernetes integration but open for other platforms extensions in future
2. Facilitate insertion of CNFs into end-to-end services:
  - Make use of all NSO's orchestration features for CNFs: configuration management, service automation, kickers, recovery, etc.
  - Abstract implementation from definition...facilitate hybrid environments and migrations
- We will not focus on implementing cluster lifecycle

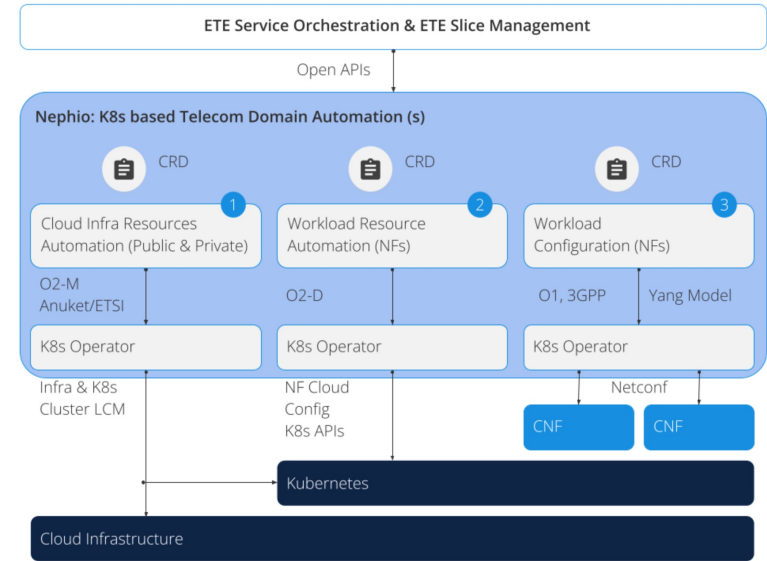


Figure 1: Configuration Layers

Nephio Linux foundation project Scope  
Cisco NSO aligned with points 2 and 3 but  
also facilitates ETE service in a single platform  
Source: <https://nephio.org/about/>

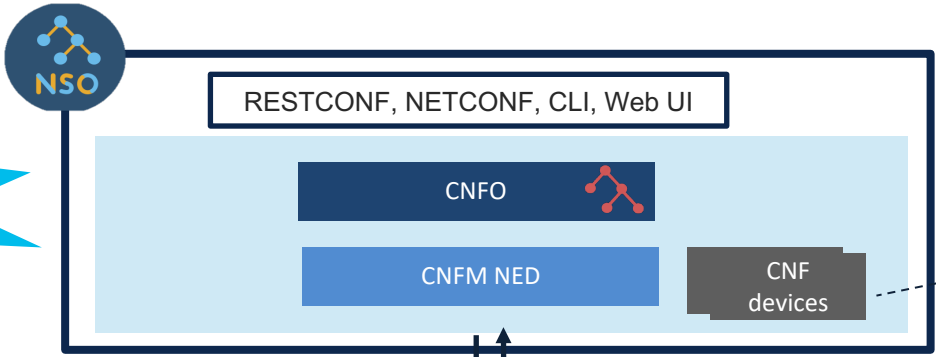
# Helm: Package Management for Kubernetes

- Collection of all artifacts for your application lifecycle in K8s
- Adopted by most CNF vendors
- Deals with versioning, upgrades and other complex operations
- **Some definitions:**
  - **Chart:** A package of pre-configured Kubernetes resources.
  - **Release:** A specific instance of a chart which has been deployed to the cluster using Helm.
  - **Repository:** A group of published charts which can be made available to others



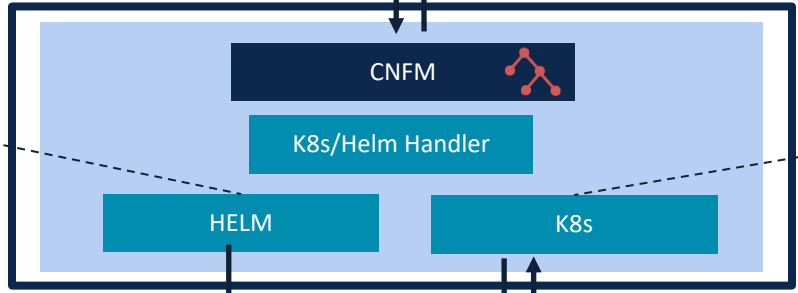
# CNF Orchestration Architecture

Available  
End of May  
2022



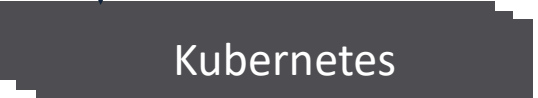
CNFs as NSO devices for CNF Configuration via NSO NEDs for direct configuration management or via EMS

CNFM Agent



- Install
- Uninstall
- Upgrade
- List

- Watch/List/Get
  - Pod
  - Service
  - Deployment
  - Replica-Set



Rest API

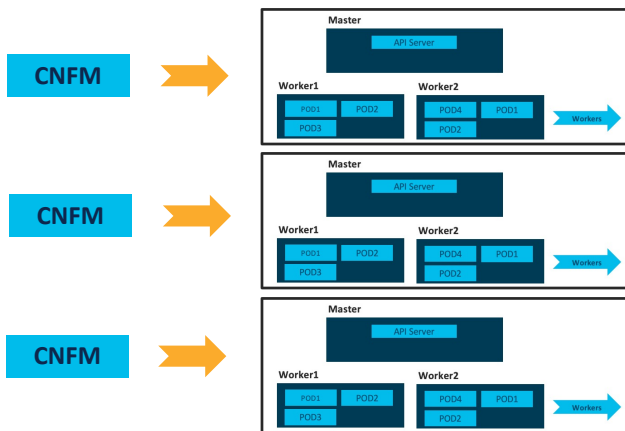
WebSocket

# Deployment Options

## Option 1: CNFM [1] -> K8s [1] (Supported)

CNFM managing Single Cluster (multiple namespaces)

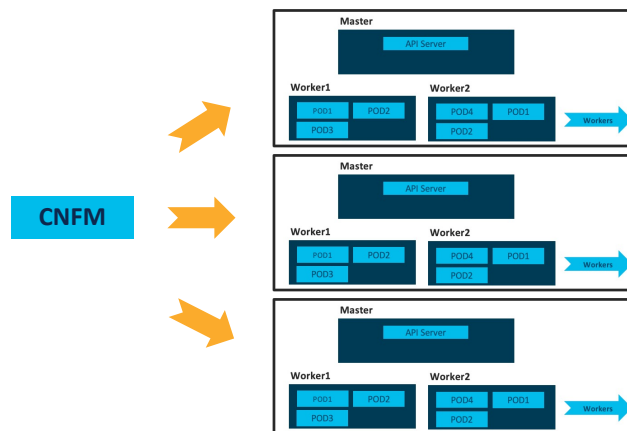
- Good model when number of K8s clusters is limited or distributed (edge) usecase



## Option 2: CNFM [1] -> K8s [n] (Future)

CNFM managing Multiple Clusters (multiple namespaces)

- Good for large number of K8s Clusters



# Cisco Hybrid orchestration

VNFs and CNFs End-to-End



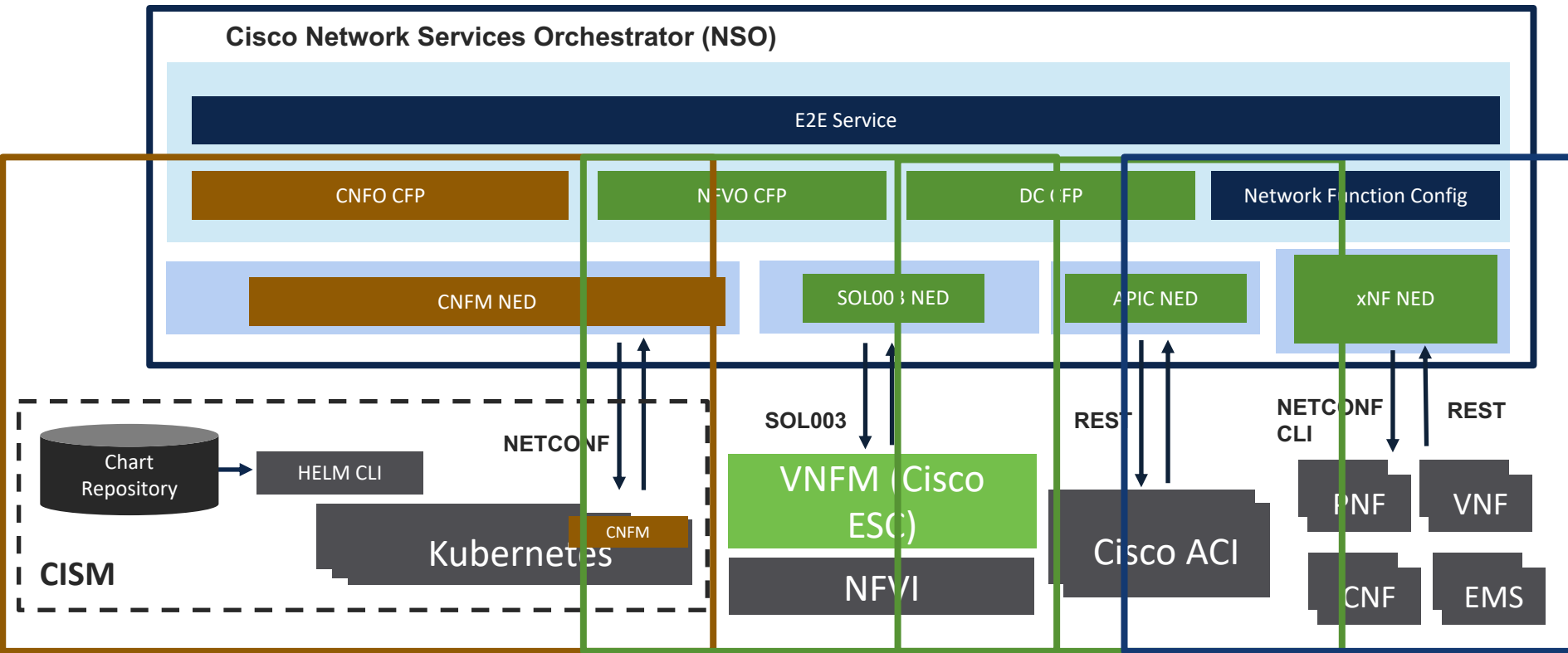
New assets



Customer assets



Existing assets from Cisco



# Conclusion

- We are expanding NSO MANO orchestration to CNFs and hybrid
- We will adopt Kubernetes and Helm but open to other platforms
- A CNFM will act as a NETCONF/YANG agent towards the clusters
- All existing NSO assets can be re-used towards CNFs (NEDs, kickers, plans, nano-services, etc.) for end-to-end orchestration



The bridge to possible