ONAP
OPEN NETWORK AUTOMATION PLATFORM

Acceleration Management

Lei Huang , Changjun Wang

January 08, 2018

Motivation

- CCVPN vision

A network hierarchy of at least two layering of DCs are assumed in this use case, including:

The core DC, where the ONAP central is running, and the edge DCs, where the virtual functions/applications that are deployed near the customer's sites are running.

For the intelligent surveillance application, which is a specific value-added service, once initialized,

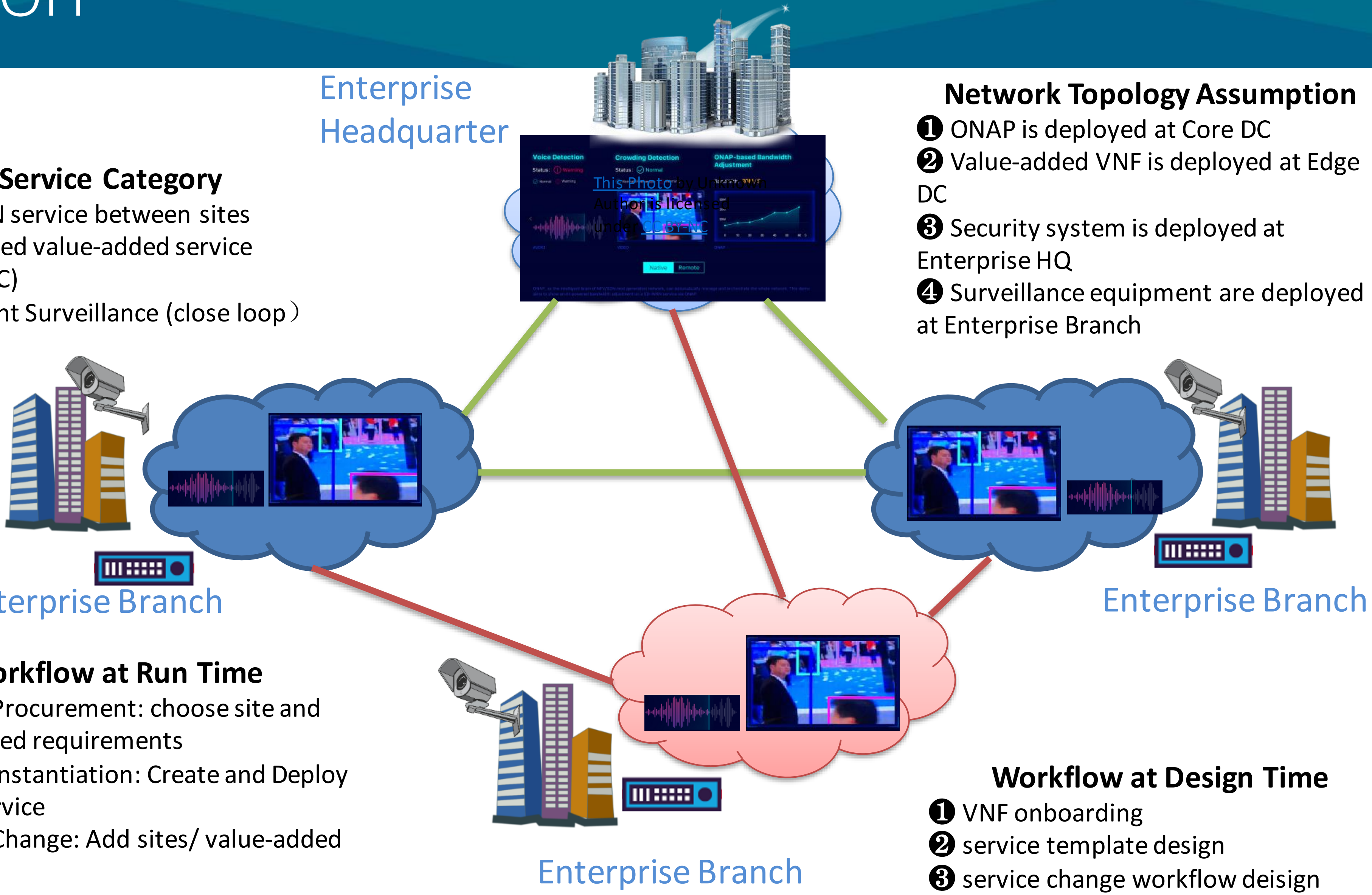
Its centralized monitoring portal is deployed at the edge DC near the enterprise HQ, and the AI applications for collecting both the voice/video monitoring and abnormally recognition, are deployed in a distributed fashion to the edge DCs that are near the specific site(s) under surveillance.

Motivation

SP Service Category

❶ Basic VPN service between sites
❷ Customized value-added service

- vFW (SFC)
- Intelligent Surveillance (close loop）



This Photo by Unknown Author is licensed under CC BY-NC

Network Topology Assumption

❶ ONAP is deployed at Core DC
❷ Value-added VNF is deployed at Edge DC
❸ Security system is deployed at Enterprise HQ
❹ Surveillance equipment are deployed at Enterprise Branch

Workflow at Run Time

❶ Service Procurement: choose site and set cutomized requirements
❷ Service Instantiation: Create and Deploy the VPN service
❸ Service Change: Add sites/ value-added service
❹ Intelligent Surveillance: Anormaly recognition and adjust bandwidth in close loop

Workflow at Design Time

❶ VNF onboarding
❷ service template design
❸ service change workflow deisign
❹ bandwidth adjustment policy design

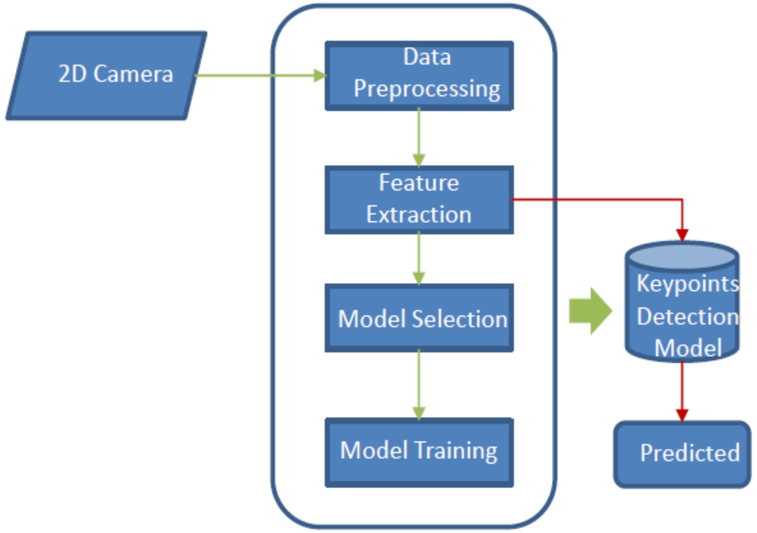
Requirements from CCVPN

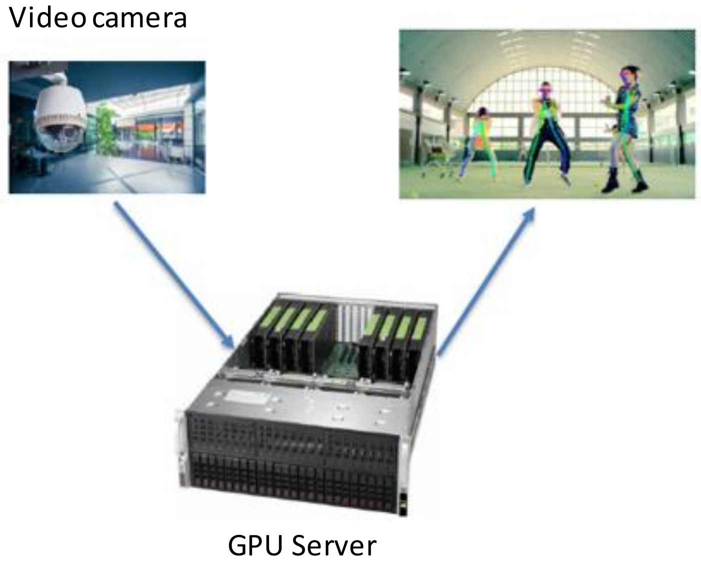
1. Existing implementation ways

- Data collection of video streaming： Video camera collects data and processes 30 frames of data per second；
- Centralized processing of video streams： Upload to cloud center to centrally handle video streaming, including data preprocessing, image feature extraction, algorithm model selection, model training, etc.

2.Reqs

- Rational allocation of GPU resources；
- Distributed processing of video data flow；
- Assign suitable acceleration technology according to the monitoring situation of video stream





Requirements from CCVPN

- 1. CCVPN currently needs acceleration technologies , such as GPU or FPGA in intelligent surveillance system. In the case of limited GPU resources, it is necessary to be able to realize the discovery process of proper acceleration resources and allocate resources reasonably in ONAP.
- 2. At present, the amount of video streaming data in intelligent security system is huge, by deploying the video stream through the edge cloud, and adding the monitoring and analysis mechanism in the edge cloud can alleviate the rate of large data flow processing of video stream.
- 3. When deploying GPU and FPGA resources, we need to increase the management capabilities of existing ONAP for GPU resources;
- 4. Users need to monitor and analyze the performance status of the GPU and the situation of the video stream

Why we need acceleration management?

- Why we need acceleration management?

✓ Higher performance required by CCVPN

- Lower latency / jitter
- High real-time media, complex computing requirements
- Support for high efficiency in AI scenarios

✓ Requirements for efficient deployment of video streams in Edge clouds

- Lightweight video streaming transmission
- Allocate GPU resources appropriately to realize fast operation requirements of edge side computing equipment

✓ Make up gaps inside the current HPA framework and enhance this part

What we can support

- What we can support

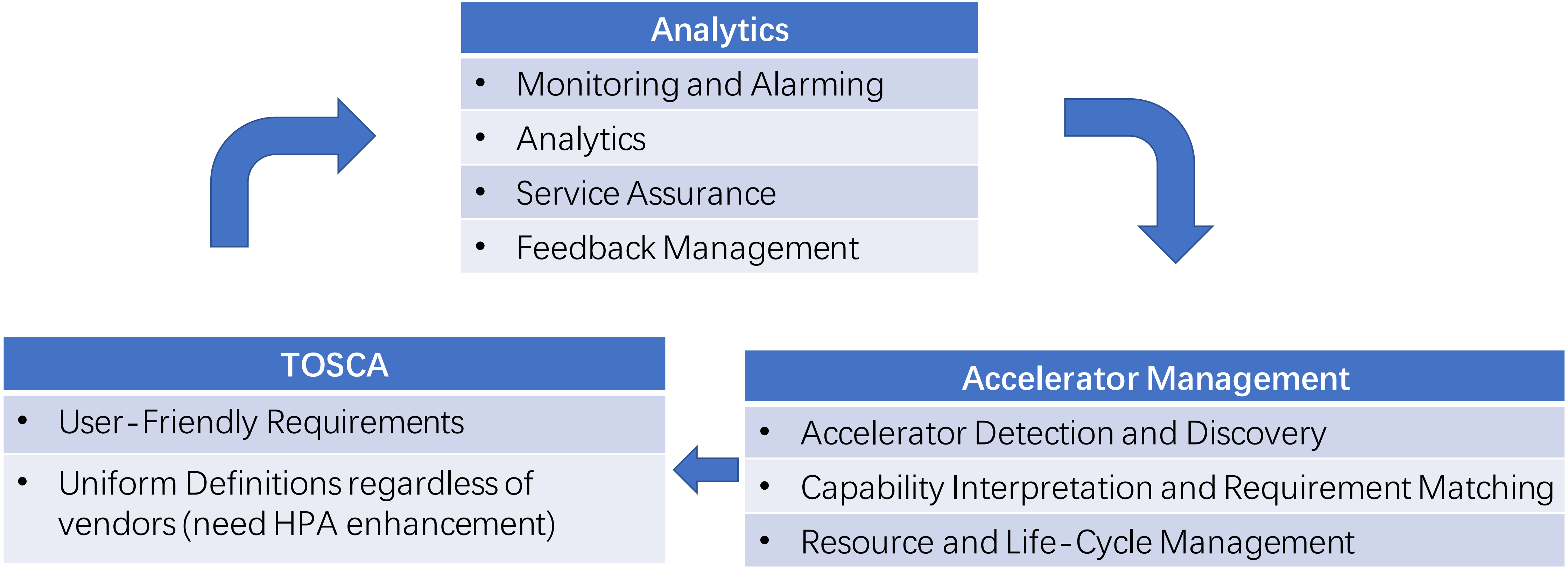
✓ Deploy video stream in Edge cloud, allocate GPU resources reasonably, reduce the transmission pressure of video stream, and realize the optimal allocation of limited GPU resources

✓ Monitor the state of hardware devices at run time and allocate hardware resources reasonably according to analytics of collected monitoring data

✓ Support hardware resources that are not currently reflected in HPA :

1. GPU;
2. FPGA;
3. ……

Acceleration Management



Analytics

- Monitoring and Alarming
- Analytics
- Service Assurance
- Feedback Management

Accelerator Management

- Accelerator Detection and Discovery
- Capability Interpretation and Requirement Matching
- Resource and Life-Cycle Management

TOSCA

- User-Friendly Requirements
- Uniform Definitions regardless of vendors (need HPA enhancement)

- Acceleration Management doesn’t handle any beyond acceleration tech, such as memory size or cpu number.
- It is an optional extension, working with the existing HPA and may need HPA’s enhancement to support some specific acceleration technologies in use cases.

Dublin and beyond goals

- From users’ perspective

1.Discovery and registration of acceleration resources

2.Instantiation and assignment of acceleration resources (Reuse existing process in HPA)

3.The termination of the use of acceleration resources and release process.

4.Adjustment of suitable acceleration resources according to the data AM monitored and analyzed. – Whether to put it in D release or beyond depends on different target of different cases, seen as stretch goal.

1.Discovery and registration of acceleration resources

- In this process, we want to achieve following goals:
- 1. Get the latest acceleration capabilities reported by OpenStack, in CCVPN we need the report of GPU or FPGA resources;
- 2. Monitoring and analysis of the running state of the accelerators;

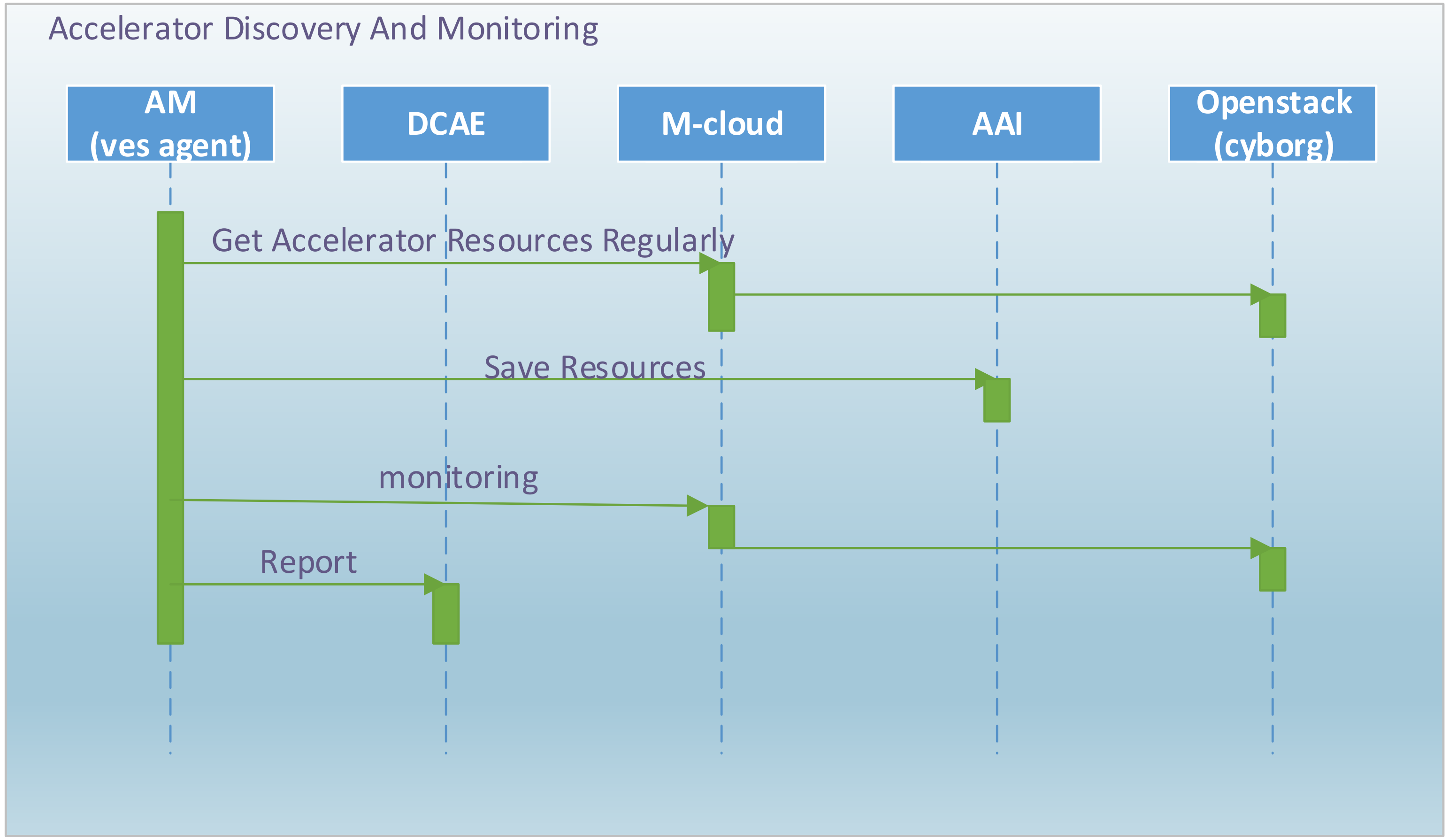
- Specific bandwidth, latency requirements, etc.

-Accelerator status monitoring, whether machines down or health status of accelerators can be regular reported to AM;

-Management of acceleration technologies with similar performance;

-Many others , we need your input and contribute to serve your cases demands

1.Discovery and registration of acceleration resources

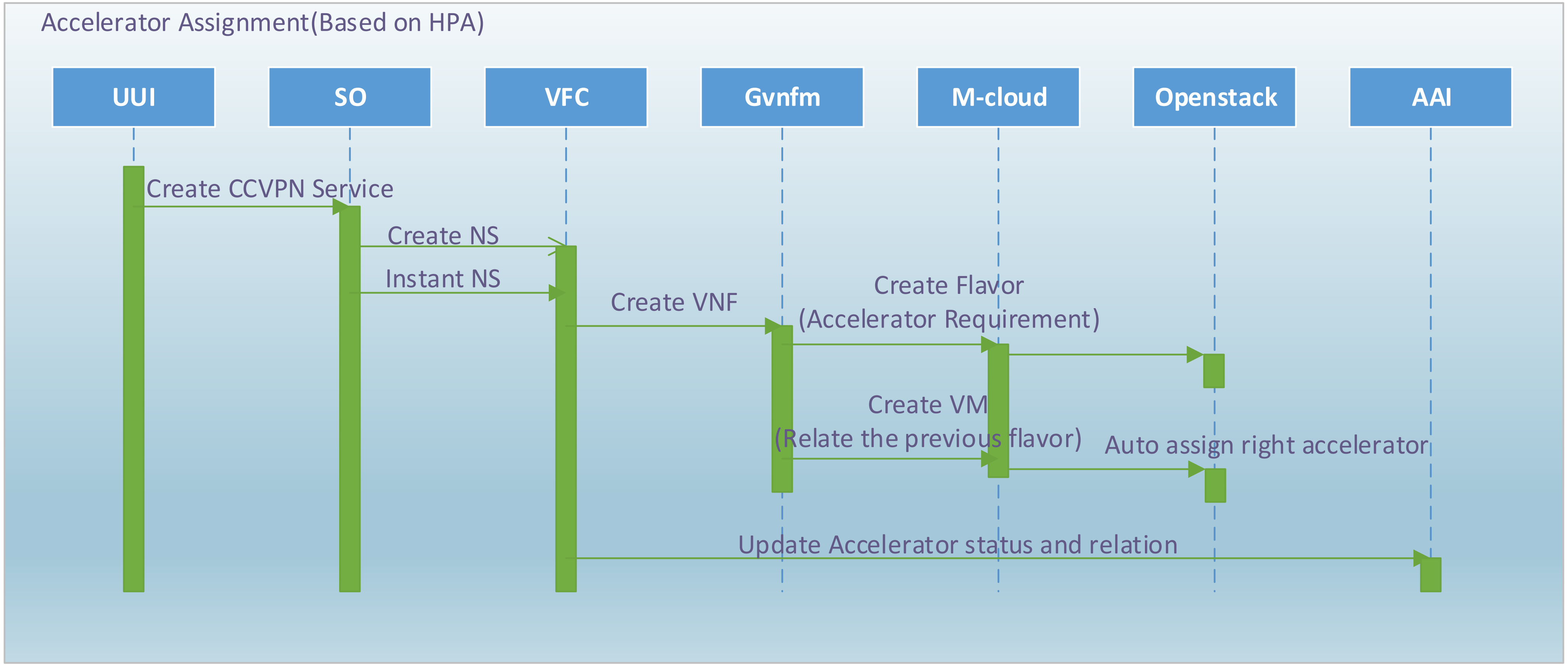


AM: Acceleration Management

HPA upload all the accelerator informations when registering the vim to ESR. But if some accelerators become abnormal, AAI doesn't know it

2.Instantiation and assignment of acceleration resources

- Work with HPA and need its enhancement



Accelerator Assignment(Based on HPA)
UUI
SO
VFC
Gvnfm
M-cloud
Openstack
AAI
Create CCVPN Service
Create NS
Instant NS
Create VNF
Create Flavor
(Accelerator Requirement)
Create VM
(Relate the previous flavor)
Auto assign right accelerator
Update Accelerator status and relation

Acceleration Technologies

Abbreviation	Description	Capability
FPGA	FPGA is an integrated circuit that can be programmed in the field after manufacture. FPGAs are similar in principle to, but have vastly wider potential application than, programmable read-only memory (PROM) chips.	Offload workloads from CPU, for e.g. AI, deep-learning, etc.
SmartNIC	A SmartNIC based on ASIC, FPGA or SOC goes beyond simple connectivity, and implements network traffic processing on the NIC that would necessarily be performed by the CPU in the case of a foundational NIC.	Offload networking process for fast or slow path
GPU	A GPU is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device.	Accelerate parallel computing, image processing, media transcoding, etc.

Acceleration Technologies

Abbreviation	Description	Capability
QAT	Intel QuickAssist Technology is a hardware accelerator for cryptographic and compression algorithms.	Accelerate cryptographic tasks and data compression.
SR-IOV	SR-IOV is a specification that allows the isolation of the PCI Express resources for manageability and performance reasons. A single physical PCI Express can be shared on a virtual environment using the SR-IOV specification.	Usually for networking acceleration by offering VFs to VMs on a NIC.
NVMe	NVMe is a host controller interface and storage protocol created to accelerate the transfer of data between enterprise and client systems and SSDs over a computer's high-speed PCIe bus.	Accelerate data reading and writing on SSDs

Acceleration Technologies

Abbreviation	Description	Capability
PCIe Pass-thru	PCI passthrough allows to assign a PCI device (NIC, disk controller, HBA, USB controller, firewire controller, soundcard, etc) to a VM, giving it full and direct access to the PCI device	Accelerate those applications running on the PCI devices
NUMA	NUMA is a method of configuring a cluster of microprocessor in a multiprocessing system so that they can share memory locally, improving performance and the ability of the system to be expanded.	Accelerate for tasks which run on the CPUs sharing the same memory NUMA.
CPU pinning/affinity	Pin the task to specific physical CPUs	Ensure the performance of the tasks running on those CPUs
RDT, RDMA, and more…We need your input		

Gaps between HPA framework & new requirements on accelerators

Current HPA Features

- ✓ PCI SR-IOV
- ✓ CPU pinning
- ✓ OVS+DPDK
- ✓ Host CPU capability request
- ✓ TXT and TCP
- ✓ FPGA
- ✓NUMA support
- ✓ Huge page support
- ✓RDT(CMT,CAT,CDP,MBM,MBA)
- ✓ IPMI/PTAS for monitor
- ✓GPU
- ✓and many others

Gaps between HPA framework & new requirements on accelerators

- Current Data Modeling for HPA (Sample)

```
logical_node:
      logical_node_requirements:
        localNumaMemorySize: '{"schema-version": "0", "schema-location": "", "platform-id": "generic", "mandatory": true, "configuration-value": "256 MB"}'
    virtual_memory:
      virtual_mem_size: 4096 MB
      #TODO HPA
      vdu_memory_requirements:
        memoryPageSize: '{"schema-version": "0", "schema-location": "", "platform-id": "generic", "mandatory": true, "configuration-value": "2 MB"}'
        numberOfPages: '{"schema-version": "0","schema-location": "","platform-id": "generic","mandatory": true,"configuration-value": 1024}'
    virtual_cpu:
      num_virtual_cpu: 2
      cpu_architecture: generic
      vdu_cpu_requirements:
        simultaneousMultiThreading: '{"schema-version": "0","schema-location": "","platform-id": "generic","mandatory": false,"configuration-value": "enabled"}'
```

Gaps between HPA framework & new requirements on accelerators

- Current Data Modeling for HPA (Sample)

```
virtual_network_interface_requirements:
    - name: sriov
      support_mandatory: true
      #TODO HPA
      network_interface_requirements:
        interfaceType: '{"schema-version": "0","schema-location": "","platform-id": "generic","mandatory": true, "configuration-value": "SR-IOV"}'
      nic_io_requirements:
        logical_node_requirements:
          pciVendorId: '{"schema-version": "0","schema-location": "","platform-id": "generic", "mandatory": true, "configuration-value": "8086"}'
          pciDeviceId: '{"schema-version": "0","schema-location": "","platform-id": "generic", "mandatory": true, "configuration-value": "10ed"}'
    - name: dpdk
      support_mandatory: true
      #TODO HPA
      network_interface_requirements:
        dataProcessingAccelerationLibrary: '{"schema-version": "0","schema-location": "","platform-id": "generic","mandatory": true,"configuration-value": "dpdk"}'
```

Gaps between HPA framework & new requirements on accelerators

- Problem Statement

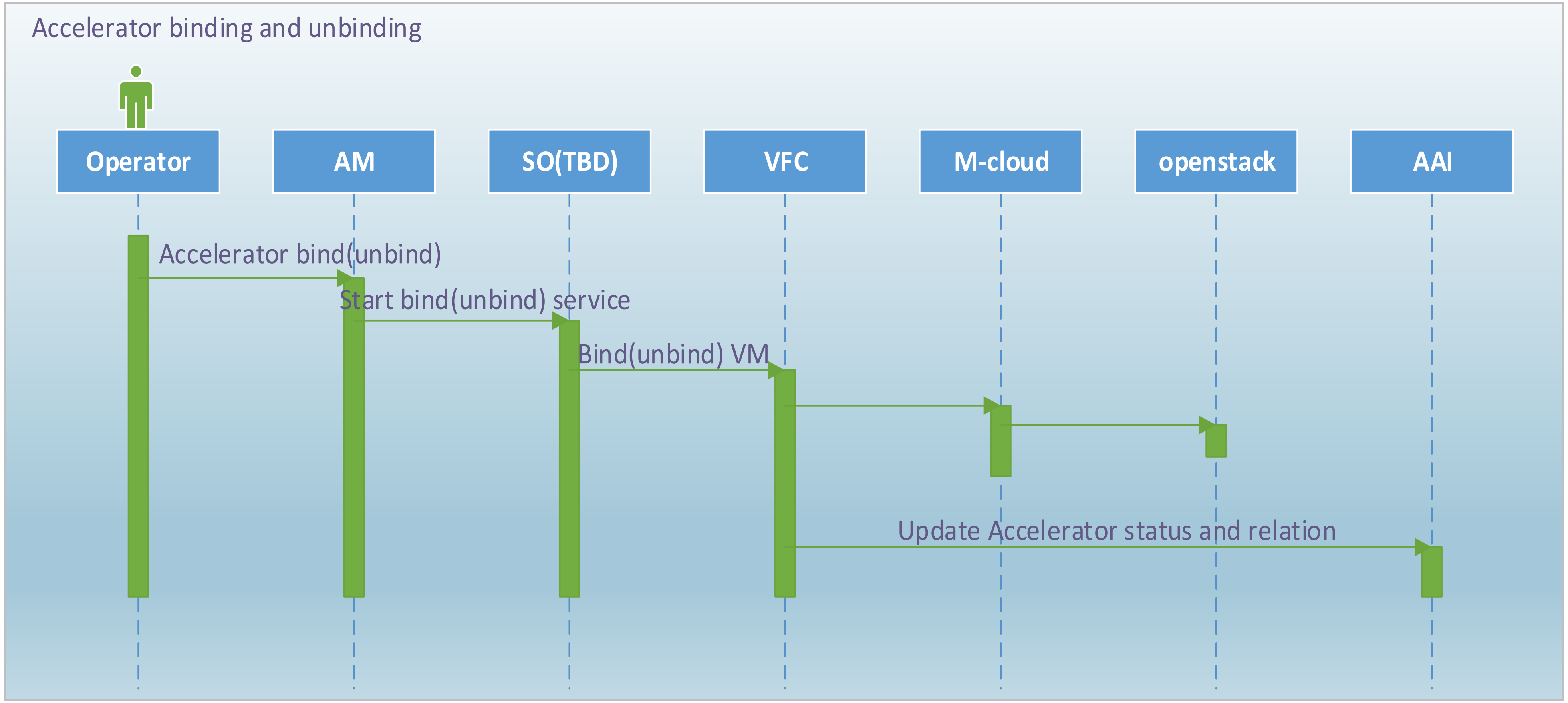
1. Besides basic hardware settings and configurations, request VNF implementers specify requirements about HPA for performance purpose
2. Lack of mechanism to track and manage performance of VNFs once onboarding for users
3. Request users have knowledge on different detailed technologies and acceleration solutions
4. Acceleration might differ to different hardware vendors, lack of uniform definitions
5. Lack of comprehensive and user-friendly definitions in modeling languages

Users should specify WHAT is going to achieve only instead of HOW to achieve

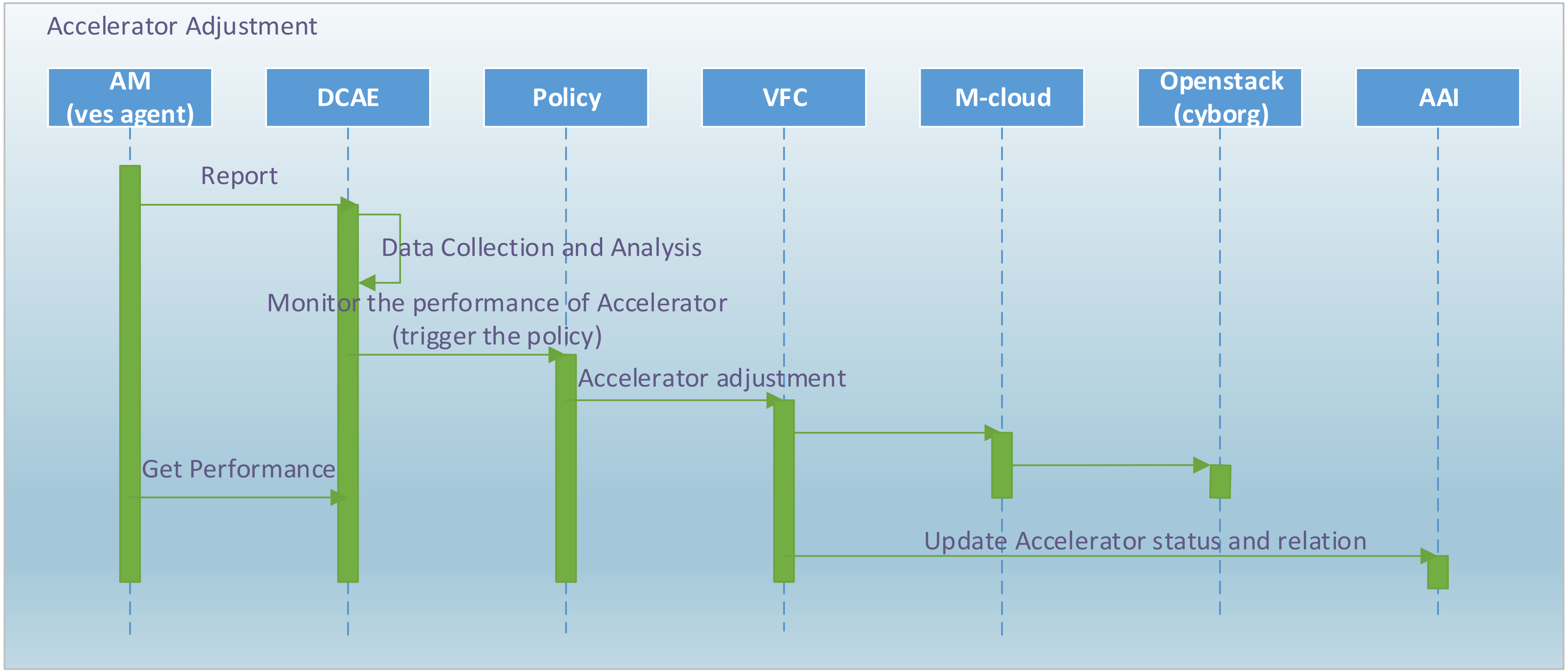
Relationship between AM & HPA

- AM complements the life cycle management of accelerator resources based on HPA solutions.
- AM will reuse binding process of HPA during instantiation, and will help promote and work with HPA to support newly acceleration capabilities which is not exist in current HPA, such as GPU, FPGA, etc.
- AM will enhance and supplement the resource discovery process, to ensure that ONAP can aware the change of acceleration resources from OpenStack side immediately by regularly updating the accelerator resource status.
- AM is a newly proposed micro service, it allows users to manually binding and decoupling acceleration resources after instantiation, enabling manual redistribution of acceleration after instantiation.
- AM complements closed-loop control for accelerators, which automatically triggers the allocation and redistribution of acceleration resources when their performance reaches a bottleneck or an exception occurs.

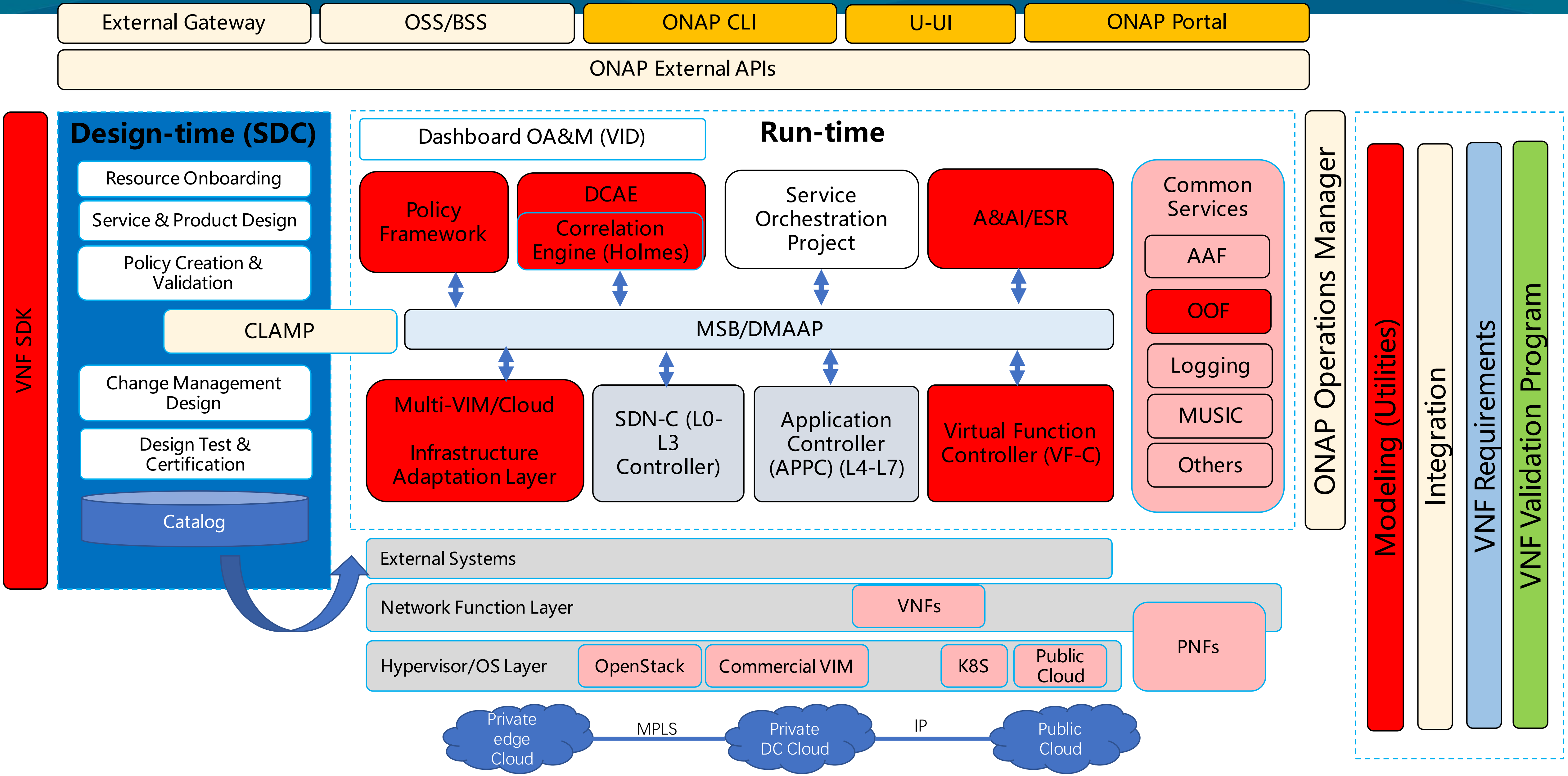
3.Termination of the use of acceleration resources and release process


Accelerator binding and unbinding
Operator
AM
SO(TBD)
VFC
M-cloud
openstack
AAI
Accelerator bind(unbind)
Start bind(unbind) service
Bind(unbind) VM
Update Accelerator status and relation

4. Adjustment of acceleration resources


Accelerator Adjustment
AM (ves agent)
DCAE
Policy
VFC
M-cloud
Openstack (cyborg)
AAI
Report
Data Collection and Analysis
Monitor the performance of Accelerator (trigger the policy)
Accelerator adjustment
Get Performance
Update Accelerator status and relation

Components affected by AM(highlight with red)



New Model for Req and Cap

New definition extension for various requirements

- General Desc: networking acceleration, transcoding acceleration, etc.
  - networking_interface_requirements: networking_acceleration: true
- Specific Desc:
  - networking_interface_requirements: min_network_bandwidth: “100MB”, max_network_latency: “5ms”
  - storage_interface_requirements: min_storage_iops: “64Mbps”, min_storage_bandwidth: “128MB”
- Complicated Desc to be defined, e.g.,
  - networking_interface_requirements: network_sriov: true, network_preferred_vendor: “intel”…
  - storage_interface_requirements: storage_media: “SSD”

New language or schema should be defined to satisfy the needs in TOSCA

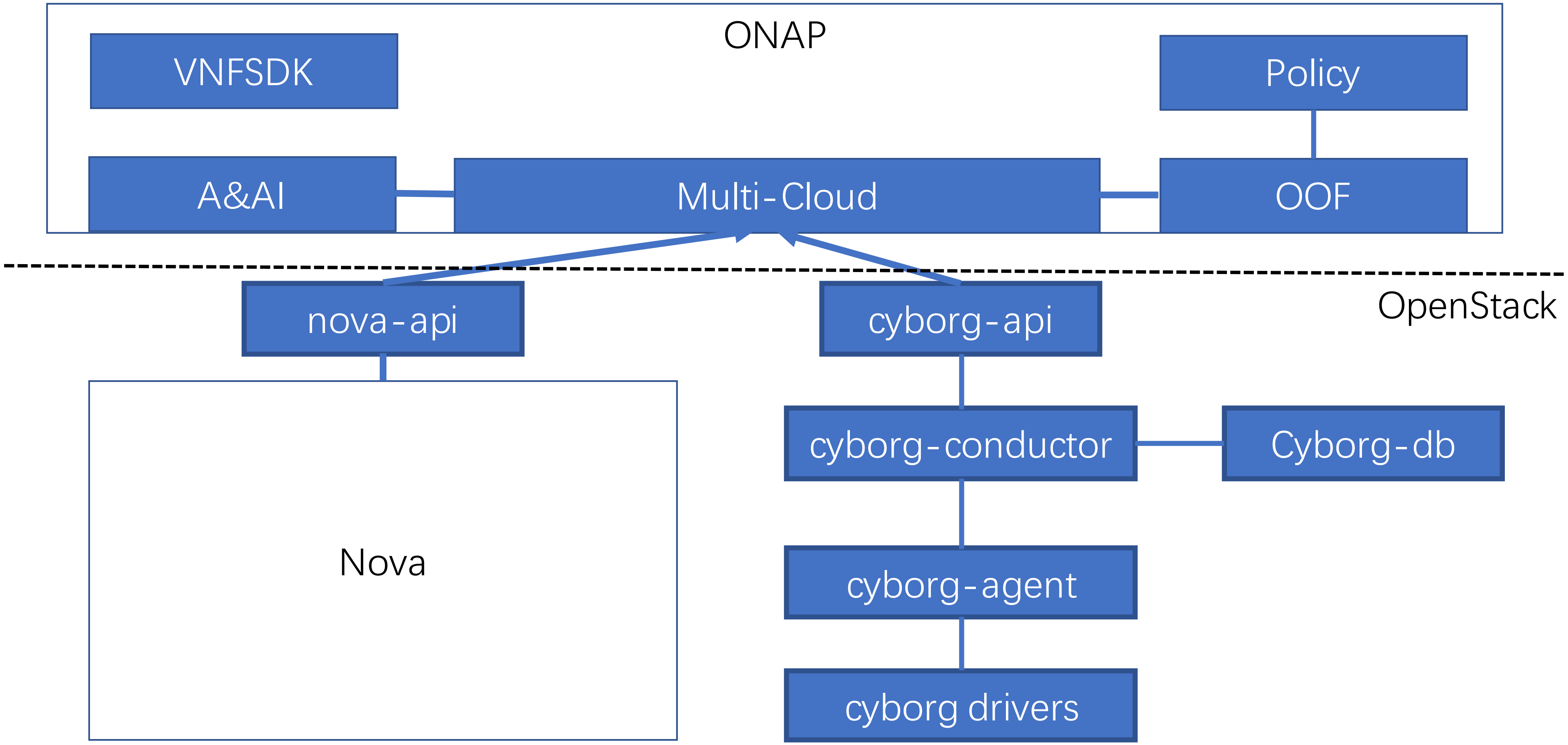
Generic capability of accelerator is discovered or detected by OpenStack

Accelerator vendors use modeling language to describe special or detailed capability of accelerators

ONAP retrieves the information of accelerator capability thru OpenStack Cyborg and Nova APIs, register it into A&AI

OOF helps to match the requirements interpreted by Policy and the capabilities

Registration from OpenStack



Resource Management

- Leverage and count on cyborg to do resource management for accelerators, including resource allocation, release, resource state tracking and life-cycle management (READ-ONLY)

- Cyborg won’t call to retrieve any data from A&AI but expose more APIs to support resource queries, and mapping between feedbacks and resources in A&AI

- Cyborg won’t save any data about accelerator capability which is written by accelerator vendor in the config file, or monitored by Analytics

Contact info

- Welcome to join in AM thread, send your email with ‘Commit ’ or ‘Contribute’ in subject line.

- Contact info:

✓ Name: Lei Huang

✓ Email address: 18350830036@163.com

✓ Wiki : https://wiki.onap.org/display/DW/Acceleration+Management+Interest+Group+Participants

Thank You!