Project NF's Cloud Native OAM Module

Introduction

After researching on NFs from different vendors, no matter NFs are PNF, VNF or CNF, they are designed as distributed applications, which usually contains functional modules including interface module, data storage module, operation and management module, NF logic processing modules.

Functional modules' description are as follows. Different NFs from different vendors may have difference on design and implementation, but they almost follow this similar design paradigm.

- **Interface Module:** module in charge of South-North communication between different NFs, which is usually implemented as load balancer.
- **NF logic processing modules:** modules processing NF service logics. For example AMF contains functions of access management, security management, session management etc., and these functions can be implemented as different NF logic processing modules.
- **Data storage module:** module stores NF states, user data and service processing context. Cloud native NF requires network logic processing module is stateless and supports scaling in /out horizontally, which usually results in that the NF data are stored in independent DB.
- **OAM(operation and management) module:** module in charge of NF management, configuration, monitoring, etc.
- **Internal communication module:** module support internal communication between differrent modules within a NF.

Among all these functional modules, **OAM module is one of the most commonly used module within different NFs, which makes it possible to be shared among NFs as a PaaS service**. Providing OAM module as a platform capability can relief the developers from spending too much effort on non-NF logic design and development. So **in XGVela release 1, cloud native OAM for NF is planned to be delivered**.
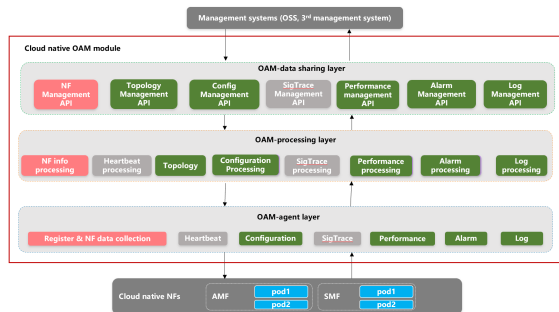
Draft High-level architecture

Figure 1: Draft High-level Architecture of cloud native OAM module

Cloud native OAM is a group of operation and management functionalities, which is treated as a group of tightly related PaaS capabilities running as containers. This module can provide local O&M functionalities for cloud native NFs on edge as well as O&M data service capablity.

Cloud native OAM is planned to be designed as three logical layers: agent layer, processing layer and data sharing layer.

- **Agent layer** connects with network functions southbound. It will be in charge of the collection of NF's O&M data and operation delivery. Cloud native network functions can use the methods defined by Agent layer to get connected with and use functions of cloud native OAM. Functionalities of this layer is usually deployed in the same environment together with NFs.
- **Processing layer** is the moddile and core processing layer of cloud native OAM. It will processing the data collected by Agent layer, the processing operation includes but not limited to data storage, data processing, policy checking, etc. This layer will also transfer data and instructions among Agent layer and Data sharing layer. Functionalities in this layer can be deployed together with Agent layer as well as in different container environment.
- **Data sharing layer** is a group of northbound interfaces, which provide O&M data subscription and NF management interfaces. The management systems like OSS or 3rd party management systems could manage the NF and get datas through these interfaces.

Relationship clarification

Cloud native OAM module planned to be designed as a Telco PaaS capability. Newly developed code and existing software integration will both exist.

Potential software integration will include Prometheus as performance metrics management backend, EFK as log management backend, Kafka as message queue, MySQL as NF data storage backend. XGVela TMaaS will also be integrated to provide NF topology.
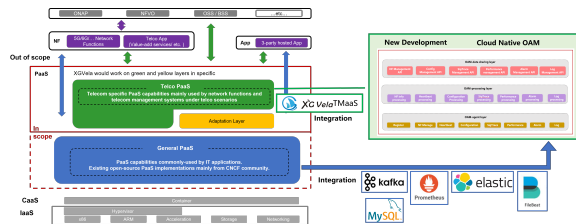
Release Planning & Release Notes

Contents to be delivered in Release 12.2023

1. ⚠ XGVELA-35 - Jira project doesn't exist or you don't have permission to view it.

Cloud native OAM Design Doc update: the current layered functional architecture figure for cloud native oam is independent from xgvela architecture. Functional type (Telco PaaS, General PaaS, Adaptation layer) and its relationship with xgvela should be reflected.

2. ⚠ XGVELA-36 - Jira project doesn't exist or you don't have permission to view it.

Complete a doc for cloud native oam southbound interface: interfaces or interaction methods between cloud native oam and sounthbound NFs should be defined.

Functional features to be delivered in Release 11.2022:

1. ⚠ **XGVELA-34 - Jira project doesn't exist or you don't have permission to view it.**

Topology management (Mavenir code): A function that provides the topology of NF, NF microservices and related deployment.

2. ⚠ **XGVELA-14 - Jira project doesn't exist or you don't have permission to view it.**

NF configuration management (new): A function that manages that configuration of NFs. It supports getting configuration files, version management of configuration, configuration file storage, and configuring NFs. The northbound API provided supports northbound management ystems to get historical configuration list, get in-use configuration of target NF, and make configuration update.

3.
> ⚠️ **XGVELA-23 - Jira project doesn't exist or you don't have permission to view it.**

NF performance management (Prometheus integration): A function that manages the performace data of NFs. It supports collecting performance data from NFs, provide data subscription functionality through northbound API. As promrtheus is commonly used for perfomance metrics data management, it will be integrated to implement this function.

4.
> ⚠️ **XGVELA-27 - Jira project doesn't exist or you don't have permission to view it.**

NF alarm management (new): A function that collects and store alarm of NFs, and report them to management systems. This function will also processing the alarm data by alarm data supplement, duplicate alarm remove, etc.

5.
> ⚠️ **XGVELA-31 - Jira project doesn't exist or you don't have permission to view it.**

NF log management (EFK integration): A function that collects and store logs of NFs. EFK integratoin prefered.

6. Other supporting functions and tasks are:

> ⚠️ XGVELA-12 - Jira project doesn't exist or you don't have permission to view it.

> ⚠️ XGVELA-21 - Jira project doesn't exist or you don't have permission to view it.

> ⚠️ XGVELA-22 - Jira project doesn't exist or you don't have permission to view it.

NOTEGreen blocks in Figure 1 are OAM functions plans to be delivered in Release 11.2022. Pink blocks are necessary for other OAM functions to operate propoerly but whose source code will not be open sources. Gray blocks are within OAM scope which will be delivered in future release.

Link to features in Project candidate list: No 10, 11, 12, 13

Development Language: Go, Java

Test plan:

> ⚠ XGVELA-32 - Jira project doesn't exist or you don't have permission to view it.

Documents to be delivered:

> ⚠ XGVELA-33 - Jira project doesn't exist or you don't have permission to view it.

Planned milestones:

Action	Planned time
Project proposal and detailed plans	July
Functional & Interface Design	July
Development plan and start coding	July/August
Code delivery	Late September/ early October
Test	Middle October
Document delivery	Late October/ Early November

Project Working Documents

Task Report

Task report

Looking good, no incomplete tasks.

Project Files

File	Modified

No files shared here yet.