Challenges in Implementing Closed-Loop Standard PNF-Based services in ONAP – Lessons Learned

SEPTEMBER 24, 2019
ANTWERP, BELGIUM

Architecture, Technology, Systems & Innovation team – TATA Communications
Ali Fouladgar, Min Yoon & Ahmad Khalil
AGENDA

• Closed-Loop Standard PNF-Based services
  • Scope & Objectives
  • Alarm Correlation & Closed-Loop for Assurance - General Flow

• Closed-Loop in ONAP
  • ONAP Architecture – Analytics
  • Data Collection in ONAP – DCAE and VES Collector
  • Alarm Correlation with Holmes
  • Closed-Loop for Assurance with Policy

• ONAP Implementation - Analytics
  • Use Case Objectives and Overview
  • WebSocket Client
  • MCP alarm format to VES
  • A&AI schema design
  • Rule design and implementation for the use case
  • Action implementation in Policy

• Challenges
  • Rules development and deployment
  • Holmes Engine package development
  • Policy

• Demo
CLOSED-LOOP STANDARD PNF-BASED SERVICES

SCOPE & OBJECTIVES

- Designing, implementing and integrating Closed-Loop automation
  - PNF-based orchestrated services, e.g., implemented based on MEF standards and models
  - YAML-based TOSCA specification service templates, including closed-loop service template
  - Analyze alarms to identify faults’ root causes to provide automatic troubleshooting
  - “Automatic” policies/actions for self-healing, based on the detected root causes by alarm correlation component

- Why Alarm Correlation?
  - For orchestrated services, alarm correlation is critical to identify impacted services in real-time and reduce downtime and automate the operational support and ticketing and “close” the service lifecycle loop
  - With analytical alarm correlation, we can reduce the pressure caused by the large alarm quantity, hence efficiently decrease the computational resources needed for other closed-loop automation components

- Why Closed-Loop Assurance?
  - Automatically act on network issues to correct them based on monitoring and assessing the network events without human intervention
  - The solution that paves the way for self-healing networks
CLOSED-LOOP STANDARD PNF-BASED SERVICES

ALARM CORRELATION & CLOSED-LOOP FOR ASSURANCE - GENERAL FLOW

Telecom Cloud/Service -> Real-time Collection and Analytic Application

Events -> Rule Designer

Original Alarms -> Correlation Engine

Analysis Result -> Inventory

Root Cause -> Actions

Self Healing

Alarm Correlation

Closed-loop for Assurance
Design Time DCAE Holmes (engine) Control-loop Design & Deployment via Portal SDC/CLAMP or via REST call

Network SDN Controller DMAAP VES Collector & Mapper Holmes (rules) DCAE Holmes (engine) Topic 1 unauthenticated.SEC_FAULT_OUTPUT DMAAP Topic 2 unauthenticated.DCAE_CL_OUTPUT Policy AAI SO (Actor) Monitor

Alarms & Events Data Path

Internal ONAP Control/Management Path

SDC

CLAMP

AAI

Analysis Results

Query update

Alarms in VES format

Root Cause

Closed-loop for Assurance

CLOSED-LOOP IN ONAP

FLOW
CLOSED-LOOP IN ONAP

HOLMES RULE DESIGN & POLICY IMPLEMENTATION

Holmes Rule Design

- Engine Management Module: Based on Apache Drools (written in Java): is used as the reasoning engine of Holmes

- All enabled rules are distributed into Drools engine and then Drools takes over all the work and output the correlation results.

- Rule Management Module: is responsible for interactions with external systems for rule deployment. All rules get into Holmes and then are stored into the database and deployed into the Drools engine if the status of the rules is enabled.

Policy Design

- Maintains, distributes, and operates on the set of rules that underlie ONAP’s control, orchestration, and management functions.
- Policy Administration Point (PAP): offers interfaces for policy creation
- Policy Decision/Distribution Point (PDP)
  - PDP-X: based on XACML: An industry standard for authorization
  - PDP-D: based on Drools: A business process rules language

Policy vs. Holmes

- Holmes is targeted for root cause analysis, but policy is aimed for auto-healing/auto-scaling
- Holmes is mainly targeted at correlation analysis between different alarms while Policy is aimed to implement control loops by triggering a series of actions
- Holmes is necessary for reducing the pressure caused by the large alarm quantity for Policy
ONAP IMPLEMENTATION

USE CASE OVERVIEW

TATA Communications (Innovation Lab)

Internet

Customer Domain

Client

TATA Communications Video Server

PE Router (AS 3493)

End-to-end L3 Service (BGP Session)

End-to-end L4-L7 Service (Video Streaming/File Transfer)

Orchestrated EVPL Service

On Demand 

Blue Planet MCP

CONTANA (CUS/BUS)

Customer service ordering portal

BUS Application

LEGA TO (BUS:SOI)

Actions from Analytic Results

Raised/Cleared Alerts

TATA Communications

Internet

Customer Domain

Client

TATA Communications

Video Server

PE Router (AS 3493)

End-to-end L3 Service (BGP Session)

End-to-end L4-L7 Service (Video Streaming/File Transfer)

Orchestrated EVPL Service

On Demand 

Blue Planet MCP

CONTANA (CUS/BUS)

Customer service ordering portal

BUS Application

LEGA TO (BUS:SOI)

Actions from Analytic Results

Raised/Cleared Alerts
ONAP IMPLEMENTATION

Underlying Network – Ciena and MCP

- The orchestrated E-line services by ONAP on Ciena network using MCP as the 3rd party controller
- MCP collects all the alarms from the physical nodes (PNFs) which are 51XX and 39XX devices
- The alarm formats are in vendor-specific format
ONAP IMPLEMENTATION

- **Alarm Collection and WebSocket Client/Server**
  - MCP uses WebSocket server/client connection to hand in the alarms
  - WebSocket client connection is the only way to pass the received alarms from the network
  - As of now, there is no collector component in ONAP to collect alarm by making this connection
  - Needed to develop an application to collect and reformat the alarms into VES format
  - This in-house application is dockerized and integrated into ONAP in the DCAE standard way of deploying the applications (by deploying cloudify blueprints in DCAE-bootstrap pod)
ONAP IMPLEMENTATION

- **Websocket Client**
  - Alarms from 3rd party controller (MPC) is not in VES format
  - There is no mapping from “MCP format” to “VES format” in ONAP’s VES Mapper component.
  - This mapping is done in Websocket application
  - To fully integrate this application with ONAP, this functionality has to be moved/added to VES Mapper component

- **DMAAP**
  - Two topics are being used:
    - “unauthenticated.SEC_FAULT_OUTPUT”: to publish VES-formatted alarms. The subscriber of this topic is Holmes engine
    - “unauthenticated.DCAE_CL_OUTPUT”: to publish analysis results. The subscriber to this topic is Policy

- **Monitor**
  - Monitor the reformatted alarms and analysis results by Holmes
  - By calling RESTful requests to DMAAP for each specific topic
ONAP IMPLEMENTATION

Rule Design

- The rules are written in Drools
- Four rules designed along with adding functions to make all required mappings and analysis happen such as
  - query and update A&AI via REST calls
  - parse the Json responses
  - create policy messages
- Drools are deployed in Holmes rule management via Postman due to Casablanca issue of Holmes’s GUI

- Four rules designed:
  - **Update** the interface status in A&AI (for both cases of alarms “Raised” and alarms “Cleared”)
  - **Correlate Alarms** (for both cases of alarms “Raised” and alarms “Cleared”) for two purposes:
    - Alarm filtering to reduce the number of alarms to be processed
    - Analyze the alarms to list all the impacted services on a link failure
ONAP IMPLEMENTATION

• **A&AI Schema:**
  - Data schema for A&AI is designed based on MEF Network Resource Management (NRM) Information Model and ONF TAPI IM so that the network resources can be managed in A&AI following MEF definitions.
Holmes Analysis Result:
- Holmes publishes the results to DMAAP’s topic “unauthenticated.DCAE_CL_OUTPUT”
- Policy is subscribing this topic

Policy:
- Drool engine is deployed on policy component as well.
- The new policy is developed to react the event coming from Holmes.
- The policy deletes the existing service and creates new service by sending request to actor component.

Closed-loop for Assurance
Actor (SO)
- SO is deploying Camunda BPMN engine for orchestrating the workflow of service request process.
- When policy sends a request for service deletion, SO recreates the service instance from the inventory (A&AI) and sends request to SDNC for resource reprovisioning.

SDNC and 3rd Party Controller (MCP)
- SDNC is responsible for resource instance creation and resource provisioning.
- SDNC sends REST request to 3rd party controller for service provisioning.
- 3rd party controller deletes and provisions the service in the network domain upon the request.
ONAP IMPLEMENTATION

INFORMATION MODEL IMPLEMENTATIONS IN ONAP

Service Template
(MEF MSCM IM)

A&AI Instances
(ONF TAPI IM & MEF NRM IM)

- Connectivity-Service-End-Point
- Connectivity-service
- Connectivity-Service-End-Point
- Service-interface-point
- Connection
- Connection-End-Point
- Node edge-point
- Node
CHALLENGES & LESSONS-LEARNED

• Rule Deployment
  • In Casablanca, the Holmes’s GUI is not available to deploy the rules
  • Issue: The rule deployment and modification is not always working properly via API PUT/POST request calls
    • Engine and Rule Pods should be restarted to be able to deploy a new rule

• Available Use Cases and Documents
  • Every use case has its own highly hard-coded source codes, methods, classes and functions
  • It seems that general functionalities can be defined and integrated in ONAP so that for each use case, the only thing to design is the rules.

• Integration with A&AI
  • The A&AI should follow a standard schema (refer to A&AI implementation based on MEF standards)
  • Under /holmes-common-parent/holmes-actions/src/main/java/org/onap/holmes/common/aai/ path, the available codes in Casablanca are developed based on early A&AI schema which is not based on a standard
  • The implementation based on MEF standard can make the rule design and deployment much easier, efficient, scalable and use-case independent.

• Policy Development
  • Since built-in policies are designed for VNF use cases, most functionalities were focused on managing VNFs and VMs. Thus, we developed a new policy which recreates the service for our use case.
  • It seems that, similar to Holmes, general functionalities can be defined and integrated in ONAP to leave only the action and policy design to the use-case user.
DEMO VIDEO