

Holmes (5/11/17)

Project Name:

- Proposed name for the project: `Holmes`
- Proposed name for the repository: `holmes`

Project description:

- Holmes project provides alarm correlation and analysis for Telecom cloud infrastructure and services, including hosts, vims, VNFs and NSs. Holmes aims to find the real reason which causes the failure or degradation of services by digging into the ocean of events collected from different levels of the Telecom cloud.
- Holmes provides docker(s) based fault correlation analysis system with APIs which could be called by external systems.
- DCAE supports Holmes to be deployed as an analytic application in the form of docker(s). Actual deployment options are flexible, which should be decided by the user/use cases, Holmes can be either deployed as a standalone alarm correlation application or be integrated into DCAE as an analytic application.



Differences between Policy and Holmes

- The business scope of Holmes is different from that of Policy

Both Holmes and Policy adopt Drools as the rules engine. The main difference between these two projects is that Holmes is mainly targeted at correlation analysis between different alarms while Policy is aimed to implement control loops by triggering a series of actions. Briefly speaking, Holmes is targeted at root cause analysis but policy is aimed for auto-healing/auto-scaling.

- Holmes is necessary for reducing the pressure caused by the large alarm quantity for Policy

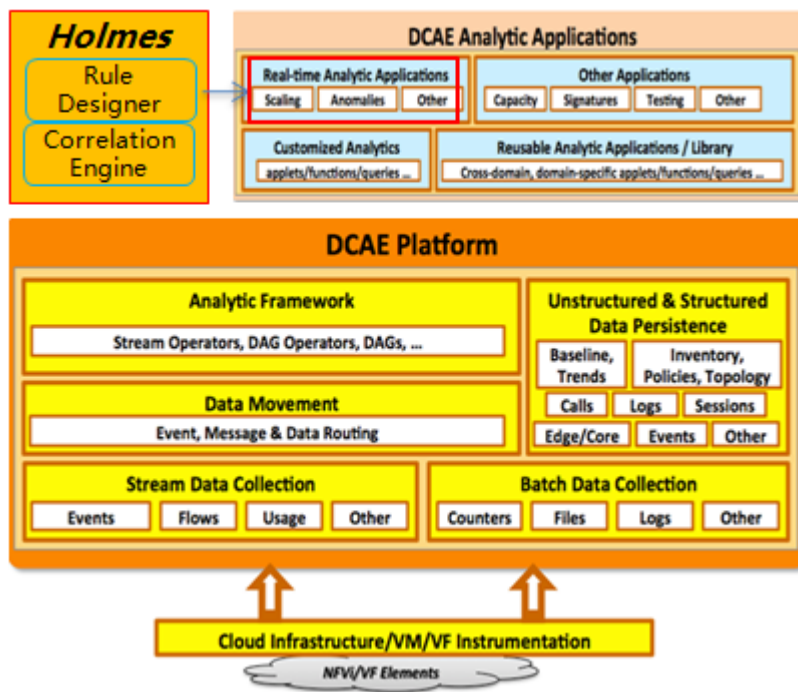
Policy does not need to face the original alarms directly with the help of Holmes. The root cause is picked out from all the original alarms by Holmes and then, the most suitable policy ID is selected and published accordingly. In this way, Policy is liberated from triggering similar or duplicated actions which are caused by the alarms with internal relations.

For example, if there are 3 events A, B and C which could lead to a power down fault, and B and C are caused by A. Without Holmes, all of these 3 events will be sent to Policy and 3 corresponding actions are going to be triggered. After we add Holmes to the close loop controller and make it the upstream system of Policy, only Event A will be sent to Policy and thus only one action will be triggered, which makes the close loop control more precise and efficient.

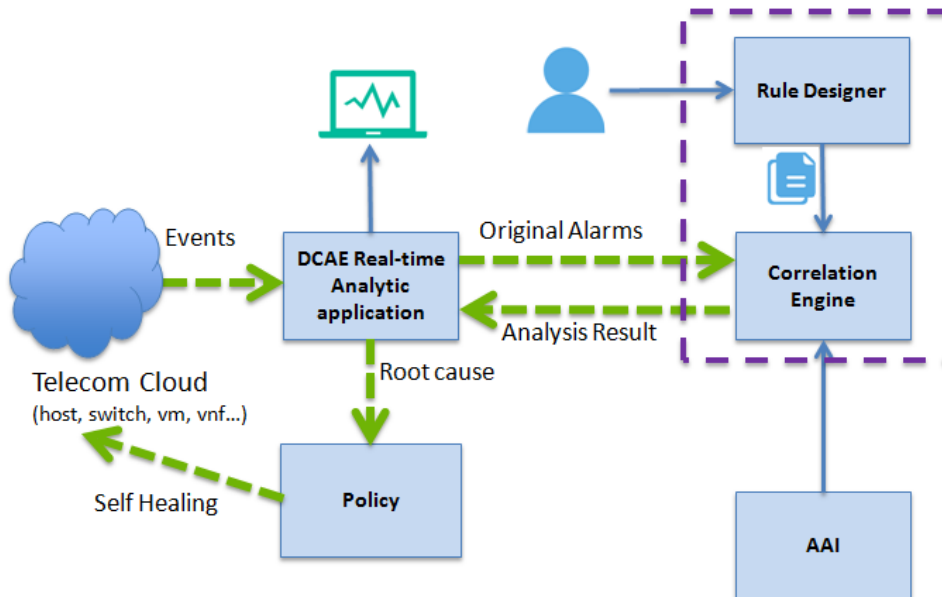
Scope:

- Alarm Correlation Rule Management
 - Holmes provides basic rule management functionalities which allow users to design, create or modify rules via a rule designer.
- Collect Alarms from Different Alarm Sources
 - Holmes supports different kinds of alarm sources, including NFV, SDN and any other legacy systems (as long as the corresponding interfaces of the source system are exposed).
- Alarm Analysis
 - Holmes can pick out the root cause from the ocean of alarms with the assistance of the topology information provided by other related systems.
- Persistence of the Results of Data Analyses
 - All analytic results are written to DB for persistence.
 - Holmes provides the functionality for users to view the statistical result of data analysis.
- Publish the Analytic Results to Subscribers
 - Besides result persistence, Holmes publishes the analytic results to a specific topic. Any potential users can subscribe to the topic to get the results in real time.

Architecture Alignment:



- Holmes is an application that processes events published by managed resources or other applications that detect specific conditions. Based on defined root cause analysis rules about the network, an application of this kind would determine root cause for various conditions and notify other interested applications.
- Holmes is designed in compliance with the VES standard. It can take event data from DMaaP, consume them and then send the correlation result back to DMaaP in the form of VES structure. Any other projects subscribe to the corresponding topic could fetch and use the result.



- Holmes consists of a rule designer and a correlation engine.
- Real-time Analytic application get the analysis result, which is the root cause can be used to drive the policy run-time for automation operation.
- Correlation Engine receives original alarms from monitor and outputs the result back to monitor after analysis based on rules and the resources relationships from A&AI.
- The Rule Designer provides a user-friendly GUI to design the Correlation rules. The GUI can be either run in standalone mode or integrated into other projects if needed.

Resources:

| Role | Name | Gerrit ID | Company | Email | TimeZone |
|------|------|-----------|---------|-------|----------|
|------|------|-----------|---------|-------|----------|

| | | | | | |
|-----------------|--------------|--|---------------|--|------------------------|
| Primary Contact | Guangrong Fu | | ZTE | fu.guangrong@zte.com.cn | Beijing, China. UTC +8 |
| Commiters | Guangrong Fu | | | | |
| | Peng Tang | | ZTE | tang.peng5@zte.com.cn | Beijing, China. UTC +8 |
| Contributors | Jiaqiang Du | | ZTE | du.jiaqiang@zte.com.cn | Beijing, China. UTC +8 |
| | Yi Li | | ZTE | li.yi101@zte.com.cn | Beijing, China. UTC +8 |
| | Youbo Wu | | ZTE | wu.youbo@zte.com.cn | Beijing, China. UTC +8 |
| | Liang Feng | | ZTE | feng.liang1@zte.com.cn | Beijing, China. UTC +8 |
| | Yuan Liu | | China Mobile | liuyuanjy@chinamobile.com | Beijing, China. UTC +8 |
| | Chengli Wang | | China Moblile | wangchengli@chinamobile.com | Beijing, China. UTC +8 |
| | Xin(Saw) Jin | | Huawei | saw.jin@huawei.com | Beijing, China. UTC +8 |

Other Information:

- link to seed code (if applicable)
git clone <https://gerrit.open-o.org/r/holmes-actions>
git clone <https://gerrit.open-o.org/r/holmes-engine-management>
git clone <https://gerrit.open-o.org/r/holmes-gui>
git clone <https://gerrit.open-o.org/r/holmes-rule-management>
- Vendor Neutral
 - if the proposal is coming from an existing proprietary codebase, have you ensured that all proprietary trademarks, logos, product names, etc., have been removed?
- Meets Board policy (including IPR)

TSC Comment Clarification

(Roberto Kung)

Holmes should be looked with Clamp or/and Policy, mainly policy (with introduction of engines and so on). May be a split is needed (analytics – alarm aggregation, filtering, correlation in DCAE analytics microservices / policy design RCA in policy). May not be high priority for R1 (not needed for our use cases). But it is useful to show intents for following releases

(Lingli Deng)

Just to clarify, cross-layer fault correlation is in scope for VoLTE usecase for auto-healing.

(Mazin Gilbert)

This project should be split and combine with DCAE (for the correlation engine), Policy engine (for Drools), and CLAMP (for designing the open loop).

(Lingli Deng)

What about the portal demonstrating the alarms gathered, and correleation made? Would DCAE be providing a portal for that?

(Unknown)

What's the relationship between CLAMP and Holmes?

(Guangrong Fu)

Holmes is essential for control loops so it should be somewhat provisioned by CLAMP. For instance, if possible, rules of Holmes can be deployed /un-deployed via CLAMP. But how to implement this is still a mystery because so far we haven't got any seed code or API docs about CLAMP, which prevents us from further analysis.

Key Project Facts

Project Name:

- JIRA project name: holmes
- JIRA project prefix: holmes

Repo name: holmes

Lifecycle State:

Primary Contact: Guangrong Fu (fu.guangrong@zte.com.cn)

Project Lead: Guangrong Fu (fu.guangrong@zte.com.cn)

mailing list tag [Should match Jira Project Prefix]

Committers:

Please refer to the table above.

*Link to TSC approval:

Link to approval of additional submitters: