

Holmes Release Planning for Casablanca

- 1 [Overview](#)
- 2 [Scope](#)
 - 2.1 [What is this release trying to address?](#)
 - 2.2 [Use Cases](#)
 - 2.3 [Minimum Viable Product](#)
 - 2.4 [Functionalities](#)
 - 2.4.1 [Epics](#)
 - 2.4.2 [Stories](#)
 - 2.5 [Longer term roadmap](#)
- 3 [Architecture](#)
- 4 [Architecture](#)
 - 4.1 [High level architecture diagram](#)
 - 4.2 [Platform Maturity](#)
 - 4.3 [API Incoming Dependencies](#)
 - 4.4 [API Outgoing Dependencies](#)
 - 4.5 [Third Party Products Dependencies](#)
- 5 [Testing and Integration Plans](#)
- 6 [Gaps](#)
- 7 [Known Defects and Issues](#)
- 8 [Risks](#)
- 9 [Resources](#)
- 10 [Release Milestone](#)
- 11 [Team Internal Milestone](#)
- 12 [Documentation, Training](#)
- 13 [Other Information](#)
 - 13.1 [Vendor Neutral](#)
 - 13.2 [Free and Open Source Software](#)

Overview

Project Name	Enter the name of the project
Target Release Name	Casablanca
Project Lifecycle State	Incubation
Participating Company	ZTE, HUAWEI, China Mobile

Scope

What is this release trying to address?

- S3P requirements approved by TSC
- Optimizing the integration with Kubernetes based DCAE
- Supporting the use cases which need correlation analysis

Use Cases

- [Use Case: VoLTE\(approved\)](#)







Minimum Viable Product

- Scalable engine management component - the actual running place for Holmes rules.
- Scalable rule management component - responsible for the CRUD operations on Holmes rules.
- Rule templates/entities for specific use cases

Functionalities

List the functionalities that this release is committing to deliver by providing a link to JIRA Epics and Stories. In the JIRA Priority field, specify the priority (either High, Medium, Low). The priority will be used in case de-scoping is required. Don't assign High priority to all functionalities.

Epics

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
HOLMES-139	Platform Maturity Requirements for Casablanca		Jun 26, 2018	Jan 18, 2019		Unassigned	None		CLOSED	Done
HOLMES-138	Integration with Kubernetes-Based DCAE		Jun 26, 2018	Jan 18, 2019		Unassigned	None		CLOSED	Done
HOLMES-95	Documentation Maintenance		Jan 13, 2018	Jan 18, 2019		Unassigned	None		CLOSED	Done
3 issues										

Stories

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
HOLME S-182	Update Release Notes		Nov 23, 2018	Dec 07, 2018		Unassigned	None		CLOSED	Done
HOLME S-181	Update the Frontend to HTTPS		Nov 05, 2018	Nov 07, 2018		Unassigned	None		CLOSED	Done
HOLME S-180	Fix the License Info for the Frontend		Nov 05, 2018	Nov 06, 2018		Unassigned	None		CLOSED	Done
HOLME S-179	Adjust the Docker Version to Abey the Tagging Convention		Oct 31, 2018	Nov 01, 2018		Unassigned	None		CLOSED	Done
HOLME S-177	Fix the AAI API Version fo VoLTE		Oct 25, 2018	Nov 07, 2018		Unassigned	None		CLOSED	Done
HOLME S-176	HealthCheck for Holmes		Oct 15, 2018	Aug 12, 2023		Unassigned	None		CLOSED	Done
HOLME S-174	Fix ActiveMQ Vulnerability Issue		Oct 15, 2018	Oct 22, 2018		Unassigned	None		CLOSED	Done
HOLME S-173	Release Holmes Dockers		Oct 15, 2018	Oct 18, 2018		Unassigned	None		CLOSED	Done
HOLME S-171	Update the Version of Homles Common Lib to 1.2.0		Sep 25, 2018	Aug 12, 2023		Unassigned	None		CLOSED	Done
HOLME S-169	Update Holmes Docs on Readthedocs		Sep 17, 2018	Dec 30, 2018		Unassigned	None		CLOSED	Done
HOLME S-168	Update the swagger file.		Sep 17, 2018	Sep 17, 2018		Unassigned	None		CLOSED	Done
HOLME S-167	Upload the VoLTE Rule to Gerrit Repo		Sep 13, 2018	Sep 13, 2018		Unassigned	None		CLOSED	Done
HOLME S-165	Fix the Vulnerability Issues Leveled Critical and High		Sep 10, 2018	Sep 15, 2018		Unassigned	None		CLOSED	Done
HOLME S-162	Finish the CCVPN Rule		Aug 23, 2018	Aug 12, 2023		Unassigned	None		CLOSED	Done
HOLME S-161	Update Holmes Component Versions		Aug 22, 2018	Aug 30, 2018		Unassigned	None		CLOSED	Done
HOLME S-160	Implement the Query Tools for CCVPN Use Case		Aug 17, 2018	Aug 12, 2023		Unassigned	None		CLOSED	Done
HOLME S-159	Improve the UT Coverage to above 50%		Aug 17, 2018	Sep 11, 2018		Unassigned	None		CLOSED	Done

HOLME S-151	Sonar reported issue fix	✓	Jul 30, 2018	Aug 31, 2018	Unassigned	None	=	CLOSED	Done
HOLME S-150	Add a header and a sidebar to the fontpage of Holmes	📖	Jul 28, 2018	Aug 08, 2018	Unassigned	None	=	CLOSED	Done
HOLME S-149	Provide Holmes UI with a Framework (Hor. Bar & Menu)	📖	Jul 25, 2018	Aug 08, 2018	Unassigned	None	=	CLOSED	Done

Showing 20 out of 30 issues

Longer term roadmap

- A user friendly GUI for the sake of rule management
- AI supportive

Release Deliverables

Deliverable Name	Deliverable Description
API description	A brief introduction of the APIs of Holmes. Both external and internal users (systems) could implement alarm analyses using these APIs.
Documentation	Installation manual, user guide, etc.
Release Note	Release note of the release
Source Code	The source code of the sub-components listed below.

Sub-Components

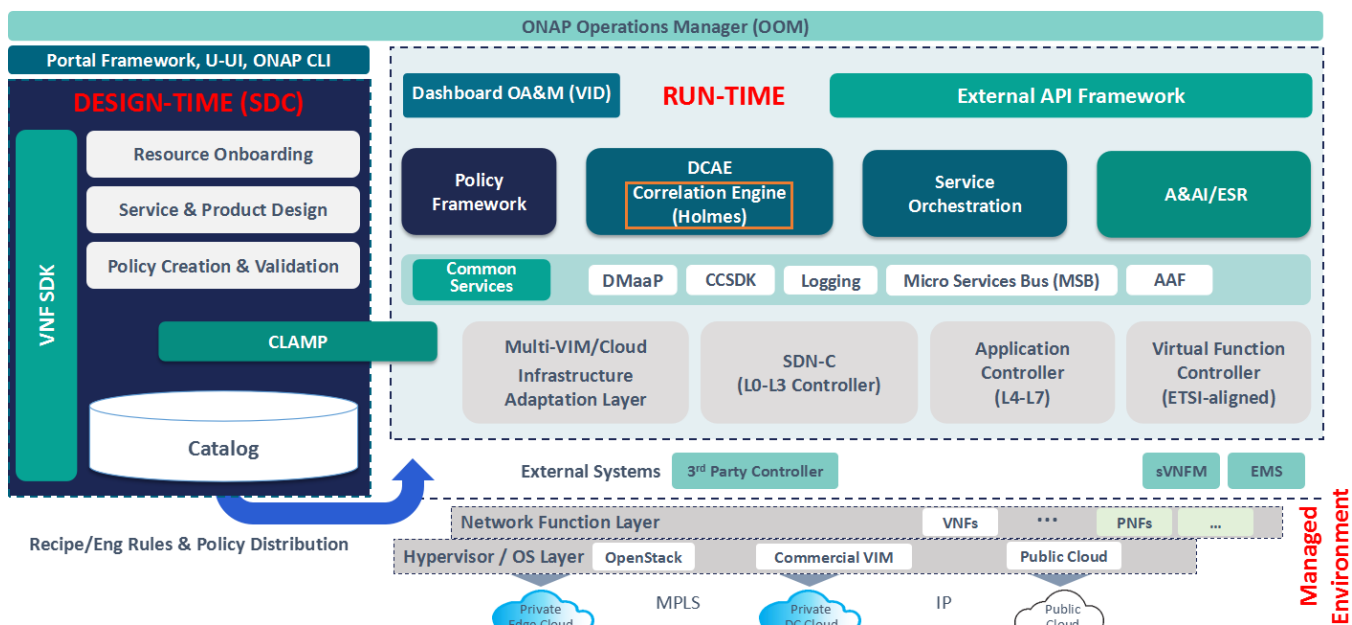
Please refer to the [Resources and Repositories](#) page.

Architecture

Architecture

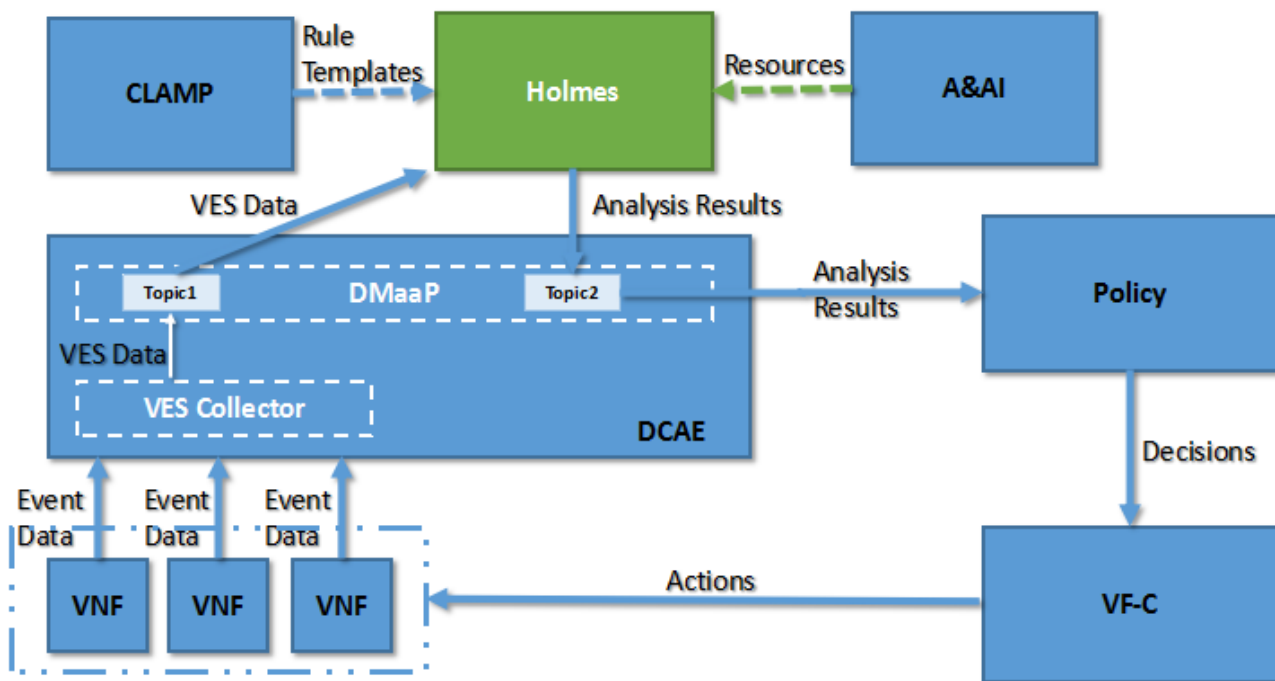
High level architecture diagram

Holmes is architecturally an analytics application reside within DCAE.

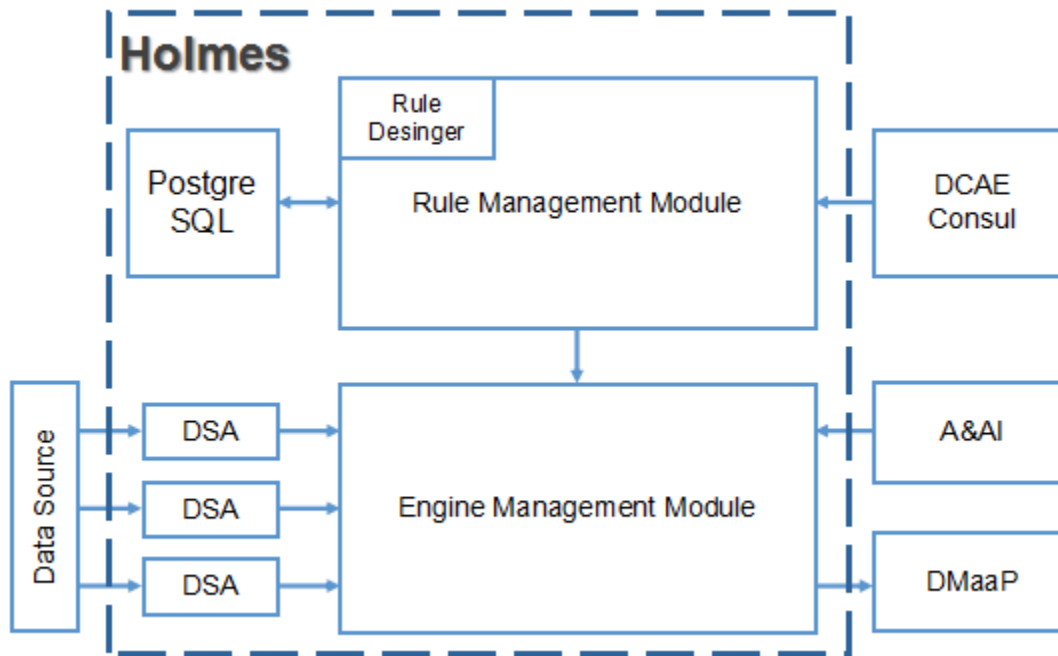


Normally, it is deployed by DCAE. But if users want to use Holmes independently (without DCAE), it could also be deployed in a standalone mode in the form of ordinary docker containers.

The interaction diagram between Holmes and its relative components is depicted below:



Holmes itself consists of three main sub-modules: the rule management module, the engine management module and the data source adapter. The rule management module is mainly responsible for the CRUD operations of Holmes rules and persisting the rules into a database. The engine management module uses the Drools engine as its core component to support correlation analysis among alarms. The data source adapter is used as the data format converter between Holmes and other components. The module diagram is like below:



Platform Maturity

Referring to [CII Badging Security Program](#) and [Platform Maturity Requirements](#), fill out the table below by indicating the actual level , the targeted level for the current release and the evidences on how you plan to achieve the targeted level.

Area	Actual Level	Targeted Level for current Release	How, Evidences	Comments
Performance	1	1	HOLMES-142 - Getting issue details... <input type="button" value="STATUS"/>	<ul style="list-style-type: none"> 0 -- none 1 – baseline performance criteria identified and measured 2 & 3 – performance improvement plans created & implemented
Stability	1	1	HOLMES-143 - Getting issue details... <input type="button" value="STATUS"/>	<ul style="list-style-type: none"> 0 – none 1 – 72 hours component level soak w/random transactions 2 – 72 hours platform level soak w/random transactions 3 – 6 months track record of reduced defect rate
Resiliency	1	2	HOLMES-107 - Getting issue details... <input type="button" value="STATUS"/> HOLMES-108 - Getting issue details... <input type="button" value="STATUS"/> HOLMES-140 - Getting issue details... <input type="button" value="STATUS"/>	<ul style="list-style-type: none"> 0 – none 1 – manual failure and recovery (< 30 minutes) 2 – automated detection and recovery (single site) 3 – automated detection and recovery (geo redundancy)
Security	1	2 (stretched goal)	NA	<ul style="list-style-type: none"> 0 – none 1 – CII Passing badge + 50% Test Coverage 2 – CII Silver badge; internal communication encrypted; role-based access control and authorization for all calls 3 – CII Gold
Scalability	1	1	NA	<ul style="list-style-type: none"> 0 – no ability to scale 1 – single site horizontal scaling 2 – geographic scaling 3 – scaling across multiple ONAP instances

Manage ability	1	1	HOLMES-140 - Getting issue details... <div>STATUS</div>	<ul style="list-style-type: none"> 1 – single logging system across components; instantiation in < 1 hour 2 – ability to upgrade a single component; tracing across components; externalized configuration management
Usability	1	2 (stretched goal)	NA	<ul style="list-style-type: none"> 1 – user guide; deployment documentation; API documentation 2 – UI consistency; usability testing; tutorial documentation

API Incoming Dependencies

List the API this project is expecting from other projects.

Prior to Release Planning review, Team Leads must agreed on the date by which the API will be fully defined. The API Delivery date must not be later than the [release API Freeze date](#).

Prior to the delivery date, it is a good practice to organize an API review with the API consumers.

API Name	API Description	API Definition Date	API Delivery date	API Definition link (i.e. swagger)
Data Movement as a Platform APIs	DMaaP message sub/pub related APIs which will be used by Holmes to collect the data from and publish data to DMaaP topics.			Data Movement as a Platform Message Router DMaaP Message Router API
Resource Query	Query different resource information from A&AI. All A&AI operations are implemented in the form of RESTful APIs. I'm using "Resource Query" as a general name for the APIs in case there will be too many APIs listed here.			AAI API
DCAE APIs	APIs used for service registration and discovery.			DCAE API Documentation
Service Registration /Un-registration Service Discovery	The APIs used to register/un-register a micro-service to/from MSB . The APIs used to discover another micro-service via MSB.			Microservice Bus API Documentation

API Outgoing Dependencies

API this project is delivering to other projects.

API Name	API Description	API Definition Date	API Delivery date	API Definition link (i.e. swagger)
Rule Creating	This API is intended for creating a rule in the database.	28 Jun 2017	August, 24th, 2017	Rule Management - Beijing
Rule Modifying	This API is intended for modifying a rule in the database.	28 Jun 2017	August, 24th, 2017	Rule Management - Beijing
Rule Deleting	This API is intended for deleting a rule from the database.	28 Jun 2017	August, 24th, 2017	Rule Management - Beijing
Rule Query	This API is intended for querying rules from the database.	28 Jun 2017	August, 24th, 2017	Rule Management - Beijing
Health Check	This API is used by other components to check whether Holmes is working.	17 Aug 2017	August, 24th, 2017	Health Check - Beijing

Third Party Products Dependencies

Third Party Products mean products that are mandatory to provide services for your components. Development of new functionality in third party product may or not be expected.
List the Third Party Products (OpenStack, ODL, RabbitMQ, ElasticSearch, Crystal Reports, ...).

Name	Description	Version
Drools (JBoss Rules)	Drools is a Business Rules Management System (BRMS) solution. It provides a core Business Rules Engine (BRE), a web authoring and rules management application (Drools Workbench) and an Eclipse IDE plugin for core development.	7.5.0
PostgreSQL	PostgreSQL is used for the sake of data (holmes rules) persistence.	9.5.0

Testing and Integration Plans

- For unit tests, we are going to keep the line coverage of each module to be above 50% or even higher.
- For the functional tests, because there will be few functional requirements for Holmes, we will main reuse the current auto testing scripts to promise that the basic functions of Holmes work well. As for the GUI part, we will add some new test cases onto the wiki page and attach the corresponding testing report to it.
- For the non-functional requirements, we will set up a set of testing env and get them tested by ourselves. Meanwhile, we'll be collaborating with the integration team to get some advice on how to get all those platform maturity requirement tested.

Gaps

This section is used to document a limitation on a functionality or platform support. We are currently aware of this limitation and it will be delivered in a future Release.
List identified release gaps (if any), and its impact.

Gaps identified	Impact
None	None

Known Defects and Issues

Provide a link toward the list of all known project bugs.

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
HOLM ES-156	Rules can not be deployed after they've been added /removed from then engine.		Aug 14, 2018	Aug 23, 2018		Unassigned	None		<div>CLOSED</div>	Done
HOLM ES-145	Swaggerresource in rule-management handle with try-with-resources		Jul 20, 2018	Aug 08, 2018		Unassigned	None		<div>CLOSED</div>	Done
HOLM ES-133	Don't rely on key word 'import' when extracting package name from rule		May 02, 2018	Aug 08, 2018		Unassigned	None		<div>CLOSED</div>	Done
HOLM ES-130	Holmes can not be successfully registered to MSB when trying to register itself with a health check parameter.		Apr 10, 2018	Sep 11, 2018		Unassigned	None		<div>CLOSED</div>	Done

4 issues

Risks

List the risks identified for this release along with the plan to prevent the risk to occur (mitigation) and the plan of action in the case the risk would materialized (contingency).

Risk identified	Mitigation Plan	Contingency Plan
To fill out	To fill out	To fill out

Resources

Fill out [the Resources Committed to the Release](#) centralized page.

Release Milestone

The milestones are defined at the [Release Level](#) and all the supporting project agreed to comply with these dates.

Team Internal Milestone

This section is optional and may be used to document internal milestones within a project team or multiple project teams. For instance, in the case the team has made agreement with other team to deliver some artifacts on a certain date that are not in the release milestone, it is recommended to provide these agreements and dates in this section.

It is not expected to have a detailed project plan.

Date	Project	Deliverable
To fill out	To fill out	To fill out

Documentation, Training

- Highlight the team contributions to the specific document related to the project (Config guide, installation guide...).
- Highlight the team contributions to the overall Release Documentation and training asset
- High level list of documentation, training and tutorials necessary to understand the release capabilities, configuration and operation.
- Documentation includes items such as:
 - Installation instructions
 - Configuration instructions
 - Developer guide
 - End User guide
 - Admin guide
 - ...



Note

The Documentation project will provide the Documentation Tool Chain to edit, configure, store and publish all Documentation asset.

Other Information

Vendor Neutral

If this project is coming from an existing proprietary codebase, ensure that all proprietary trademarks, logos, product names, etc. have been removed. All ONAP deliverables must comply with this rule and be agnostic of any proprietary symbols.

Free and Open Source Software

FOSS activities are critical to the delivery of the whole ONAP initiative. The information may not be fully available at Release Planning, however to avoid late refactoring, it is critical to accomplish this task as early as possible.

List all third party Free and Open Source Software used within the release and provide License type (BSD, MIT, Apache, GNU GPL,...).

In the case non Apache License are found inform immediately the TSC and the Release Manager and document your reasoning on why you believe we can use a non Apache version 2 license.

Each project must edit its project table available at [Project FOSS](#).

Charter Compliance

The project team comply with the [ONAP Charter](#).