

# **MOC Monitoring (MOCMon) Platform v2.0 Project Proposal**

## **1. Vision and Goals Of The Project:**

The MOC Monitoring (MOCMon) Platform v2.0 will be an improvement upon the original MOC Monitoring platform. We will change the existing monitoring platform of the MOC to Monasca, which is an open source alternative. Monasca is an actively developed multi-tenant, highly scalable, performant, fault tolerant monitoring as a service solution.

The current platform uses three different tools for monitoring, Celiometer, Logstash and Sensu. Reducing them to a single platform will make the system more user-friendly, and Monasca is a capable system that is being developed. Some advantages that we will gain by changing to monasca include an alert system, as well as any benefits of using an open-source platform.

There are some difficulties in using Monasca as well, since Monasca in its current state doesn't support some key metrics that MOCMon collects, such as power usage, switch data and fan speed. Therefore, a collection mechanism for these metrics would have to be developed to make a full transition to Monasca.

### Users/Personas Of The Project:

MOCMon will be used by the users of MOC, the administrators of MOC and any developers that develop for Cloud platforms using MOC. Although, some of these users may only use MOCMon as an intermediate step, and might not deal directly with MOCMon. For example, when a Cloud user is billed, the amount to be paid is determined by the usage amount of the VMs that the user has, which is determined by the data that goes through MOCMon.

The administrators of the system can use the metrics from Monasca to see how the system is being used, and to make a better informed decision on future system changes.

The developers can see how much system resources their VMs are using, and use these metrics to calibrate their software and see the performance of their software on the system.

## **2. Scope and Features Of The Project:**

The Monitoring (MOCMon) Platform v2.0 aims to perform the transition of MOCMon to the open source OpenStack monitoring project Monasca. Hence the project has many similarities with the original MOCMon platform, as well as a few new features.

- Composed of 4 function layers: Data Collection, Data Storage, Services and Applications.
  1. Data Collection: Virtual layer utilization information, the syslogs of the individual compute servers, the logs of the OpenStack services, and the physical server resource utilization information will be collected.
  2. Data Storage: the data gathered from the 1st layer will be stored in time-series DB like influxDB.
  3. Services: provides with IaaS services such as alerting and metering, as well as private API services if possible.
  4. Applications: provides with a wide variety of extra value-added services that utilize the monitoring data.
- Various safety levels with access permission of different user group, eg. some amateur users are only able to access certain high-level applications designed for them in application level.
- Ability to modify collection mode or parameters in data collection layer, eg. only collect one hour's data in every two hours in order to save storage and reduce cost.
- Well-organized user interface in data storage layer for displaying, eg. user can modify charts display appearance or compare 2 group of charts.
- Highly customized services options, eg. set alert content.
- Establish a market space or offer access in application layer for third-party software developers to provide with diversified applications for different user and market.

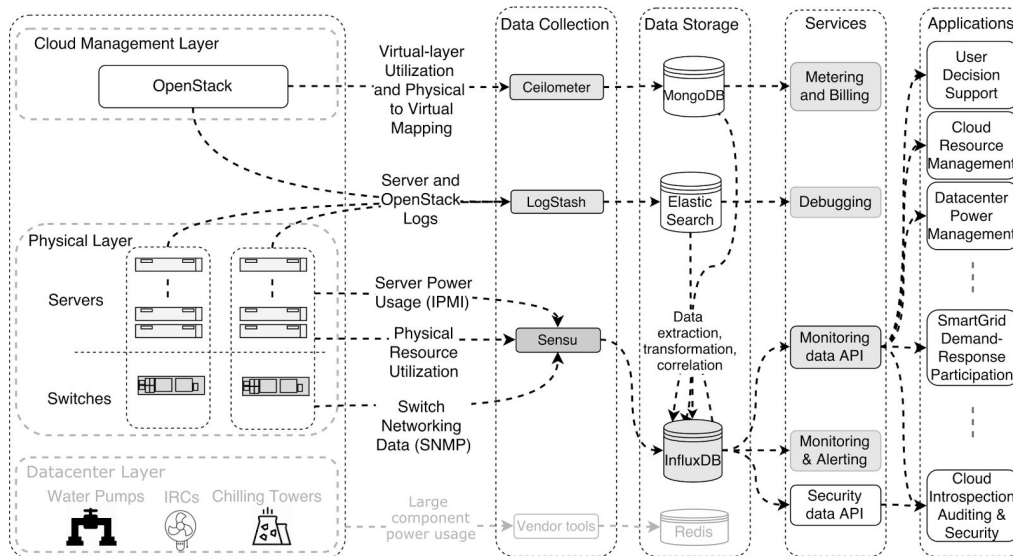
### **3. Solution Concept**

This section provides a high-level outline of the solution.

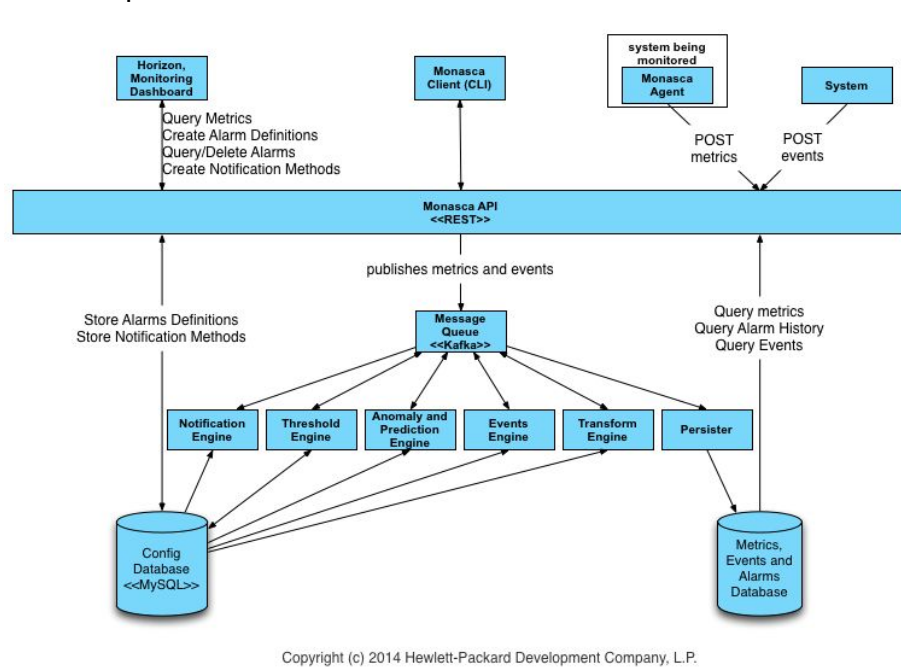
#### **Global Architectural Structure Of the Project:**

This section provides a high-level architecture or a conceptual diagram showing the scope of the solution.

- MOCMon:Massachusetts Open Cloud Monitoring Platform is the original implementation for the recording of the metrics of the MOC. This needs to be altered to work with Monasca and its open-source nature.



- Sensu: Current implementation in MOCMon for collecting physical data from the server and switches.
- InfluxDB: MOCMon's current storage implementation and backend for monitoring data.
- OpenStack: Open source cloud solution that MOCMon and Monasca use as a basis
- Ceilometer: Collects, normalise and transform data produced by OpenStack services
- Monasca: Open-source Monitoring-as-a-Service solution that provides metrics for Openstack.



### Design Implications and Discussion:

This section discusses the implications and reasons of the design decisions made during the global architecture design.

- Open-source nature of Monasca: Migrating MOCMon to Monasca will allow more monitor plugins to be designed and iterated on for the system by a larger amount of people.
- Move MOCMon to a more manageable and scalable infrastructure: Because of Monasca, more can be added to MOCMon easily, such as alarm notifications and other plugins.
- Allow MOCMon to take advantage of the OpenStack community: Having Monasca on MOCMon will allow all the open-source work created for OpenStack.

### **4. Acceptance criteria**

Our minimum acceptance criteria is to run Monasca on MOC, and to implement support for power usage metrics. Our stretch goals are to create support for a variety of metric plugins, including fan speed and switch metrics.

### **5. Release Planning:**

Release #1 (Due Feb.4) : Project Initialization

In the initialization of the project, the basic monitoring model will be set up through Monasca, simple data need to be successfully collected from MOC, data collection layer need to be accomplished.

Team members will firstly set up and learn to use Monasca in VM on their computer, will then finish getting familiar with OpenStack, will then get access into MOC from mentors and then figure out how to collect and display some basic data.

Release #2 (Due Feb.18) : DB Choosing and Managing

Release #3 (Due March.17) : IaaS Services and API providing

Release #4 (Due March.31) : Additional apps providing

Final Release (Due Apr.26) : Debug, Modify and Improve UI