CLAMP R4 - M3 Architecture Review

1. **Project Overview**
   CLAMP is the place where you manage the runtime of the Control loop (distributed by SDC) in ONAP:
   a. you can configure the parameters of the µS composing the Control Loop:
      i. this is achieved by creating/updating/deleting policies (configuration policies).
      ii. the above policies will be associated to the deployed µS
   b. you can configure the Operation to be taken by the Control Loop:
      i. this is achieved by creating/updating/deleting policies (operational policies).
   c. you can deploy/un-deploy Control Loop flow (blueprint) to DCAE.
   d. in separated/independent Docker containers, you have the "Control Loop Dashboard" which is an ELK stack dedicated to display DMAAP messages related to runtime control loop messages (from DCAE/Policies/APP).  

2. **New component capabilities for Dublin, i.e. the functional enhancements**
   a. introduction of the model driven approach for Control Loops
      i. add policy-model to the SDC distribution
      ii. automatic rendering of configuration policy UI based on the distributed policy-model
      iii. better rendering of the display of the Control Loop flow based on the distributed blueprint
   b. enhance internal model to better support model driven approach

3. **New or modified interfaces**
   a. User interface generation will stay the same even though its generation method will change.
   b. Some new internal interface will be created to support the new internal model; those new interface won’t be backward compatible since they will be new and they are internal anyway (so it won’t disturb any other project).

4. **Interface naming**
   CLAMP supports the following interfaces:
   a. **Control Loop Life Cycle Management** User interface (LCM UI) for:
      i. selecting the Control Loop flow.
      ii. entering configuration policy parameter.
      iii. entering operational policy parameter.
      iv. manage lifecycle of DCAE blueprint (Control Loop flow).
   b. **Control Loop dashboard** User Interface based on "Kibana" (ELK stack, completely independent from LCM UI).

5. **Reference to the interfaces**
   a. for the UI see readthedocs (obviously still under development for Dublin) : CLAMP latest user guide
   b. internal interface are available via swagger : clamp swagger pdf

6. **What are the system limits**
   a. none so far, CLAMP is not a in the path of runtime call, so it is not heavily hit by massive amount of call. So auto scaling is not really required but could be added.
   b. DB redundancy/HA relies on Kubernetes and persistent volume. MariaDB cluster can be a future improvement
   c. Application redundancy/HA relies on Kubernetes.

7. **Involved use cases, architectural capabilities or functional requirements**
   a. scaling use cases
   b. model driven Control Loop: Model driven Control Loop Design

8. **Listing of new or impacted models used by the project (for information only)**
   a. None.