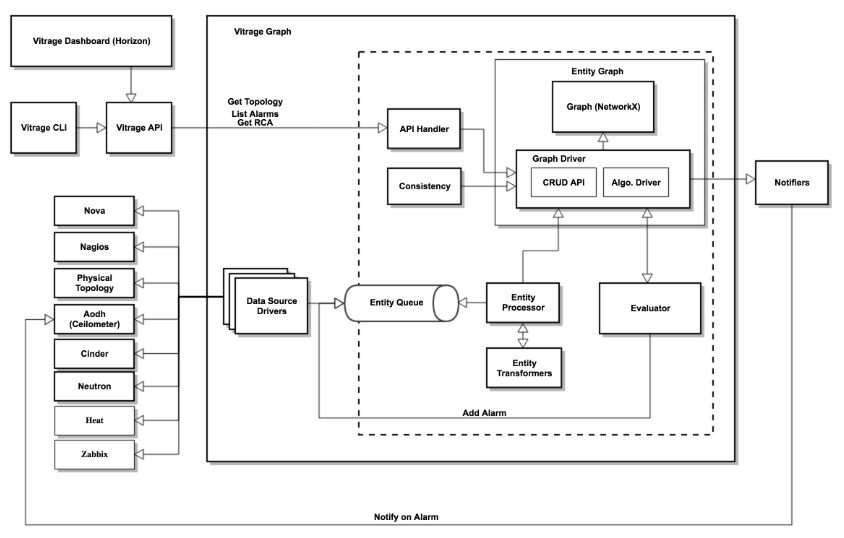
* Vai trò tổ chức: vitrage có thể biểu diễn mỗi quan hệ logic của các tài nguyên trong hệ thống thành grap+topology cho người xem, bao gồm cả thực thể virtual (instance) hay physical (host , switch). Để khi có một alarm xuất hiện từ một thành phần ta giải thích được nó sẽ ảnh hưởng tới thành phần khác như nào.
* Deduced alarm, state: mục đích nhắm tới là đưa ra cảnh báo về một thành phần hệ thống, ngay trước cả khi nó được phát hiện trực tiếp bởi hệ thống monitor. Ví dụ khi switch tới host gặp vấn đề, khiến ngay các instance nối với host đó cũng không monitor được, ta muốn dùng cơ chế suy luận của vitrage để đưa ra cảnh báo với insance và thay đổi trạng thái tương ứng sang error.
* **Root Cause Indicators:** Furthermore, we would like to be able to track this cause and effect – that the problem in the switch caused the problems experienced at the instances. Consider a single connection between the cause and effect for clarity – but all such links should be supported.
* Important Note: not all deduced alarms are caused by the trigger – the trigger might only be an indication of correlation, not causation. In the case we are examining, however, the trigger alarm is also the cause.

somekey: scenario repo

template load -> add vao scenario repo

co event den- > grep load scenario



Vitrage provides a root cause analysis service, which is used for analyzing the topology and alarms of the cloud, and providing insights about it.

The Root Cause Analysis service consists of the following components:

**vitrage-graph service**

The main process. It holds the in-memory entity graph, the template evaluator, and the different datasources.

**vitrage-notifier service**

Used for notifying external systems about Vitrage alarms/state changes. It only calls Nova force-down API and Simple Network Management Protocol (SNMP) in the Ocata release.

**vitrage-api service**

The API layer for Vitrage.

load template:

Regarding the root cause of problems and deducing the existence of problems before they are directly detected. Its ultimate goal is that

|  |
| --- |
| metadata:  name: <unique template identifier>  description: <what this template does>  definitions:  entities:  - entity: ...  - entity: ...  relationships:  - relationship: ...  - relationship: ...  scenarios:  - scenario:  condition: <if statement true do the action>  actions:  - action: ... |

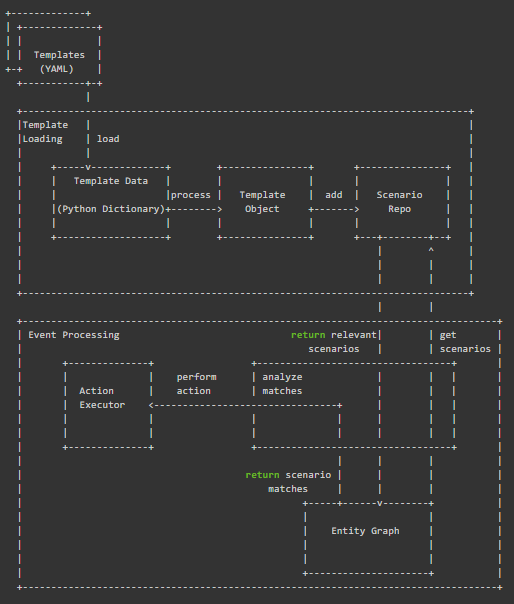
***definitions:* This section contains the atomic definitions referenced later on, for entities and relationships**

* *entities –* describes the resources and alarms which are relevant to the template scenario (conceptually, corresponds to a vertex in the entity graph)
* *relationships –* the relationships between the entities (conceptually, corresponds to an edge in the entity graph)

<https://github.com/openstack/vitrage/tree/master/etc/vitrage/templates.sample>

<https://github.com/openstack/vitrage/tree/master/vitrage/tests/resources/templates>

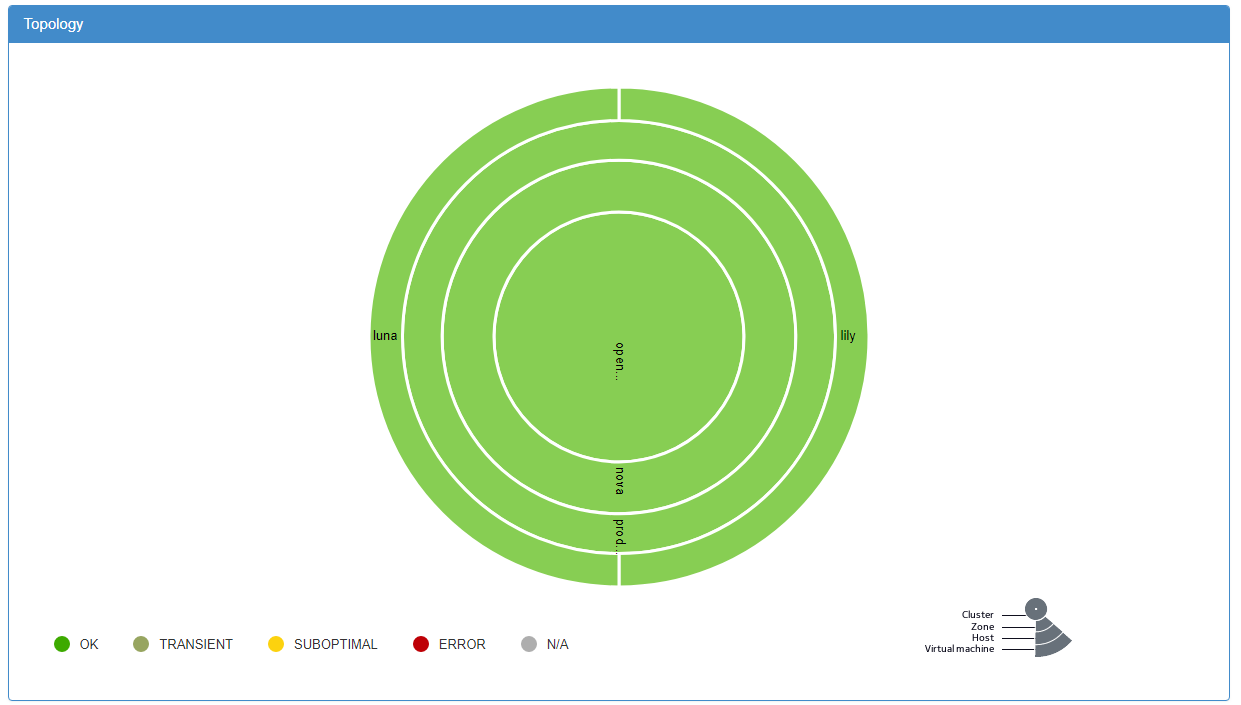
(mỗi template có thể dựng thành 1 use case mẫu, mà chưa dựng đc cái nào, cũng chưa hiểu cái nào?)



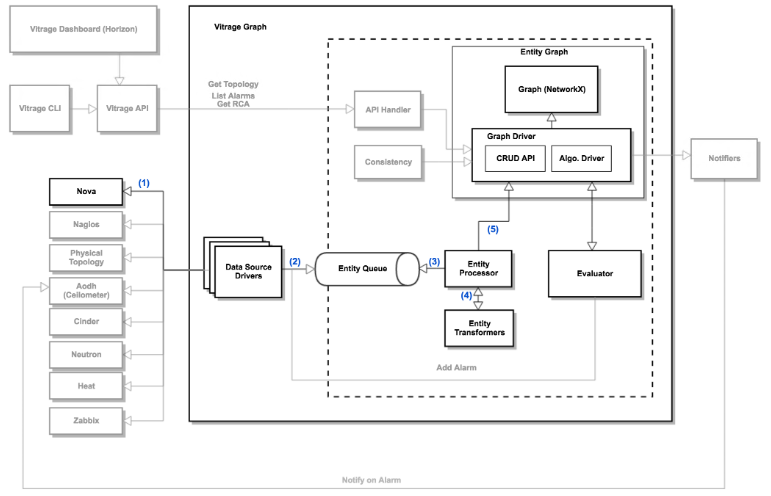
Interface:

1.topololy:

UI:

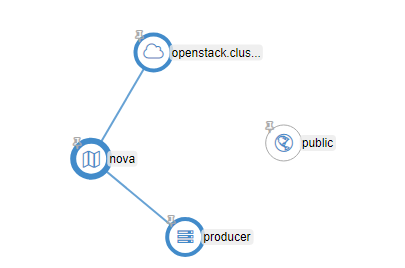


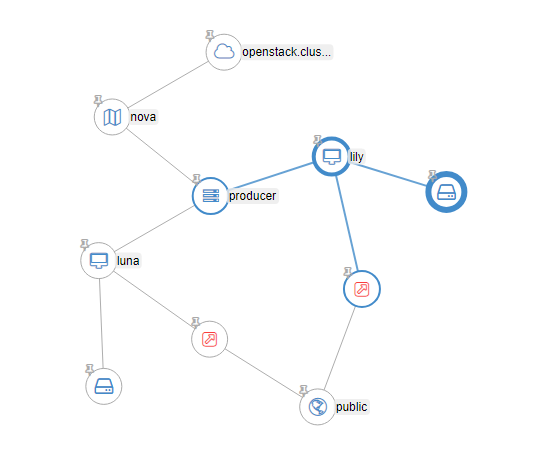
Hoạt dộng --- graph gan voi nhau  
(phần graph: chưa thể hiện rõ port nào, switch vật lý (chưa có cài đặt) như nào , nếu cài nhiều node???)



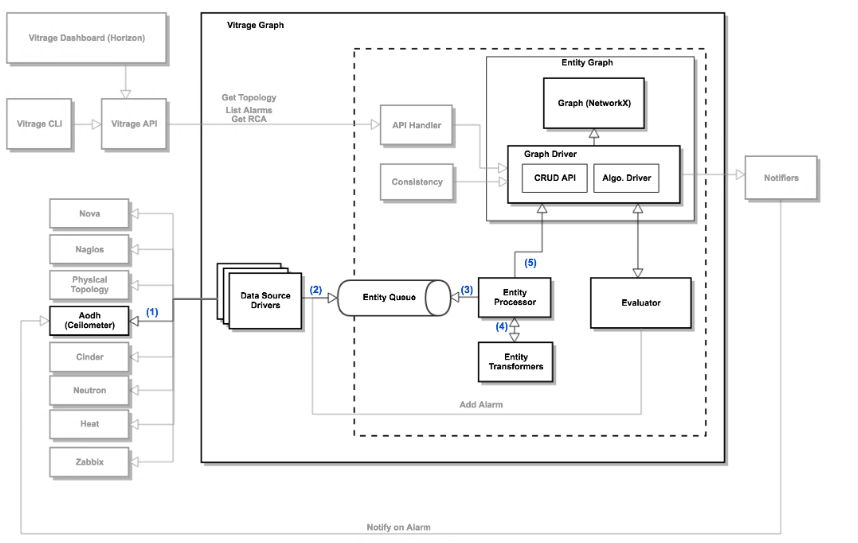
Khi nova thêm 1 instance:

<https://docs.openstack.org/vitrage/pike/contributor/vitrage-use-cases.html>



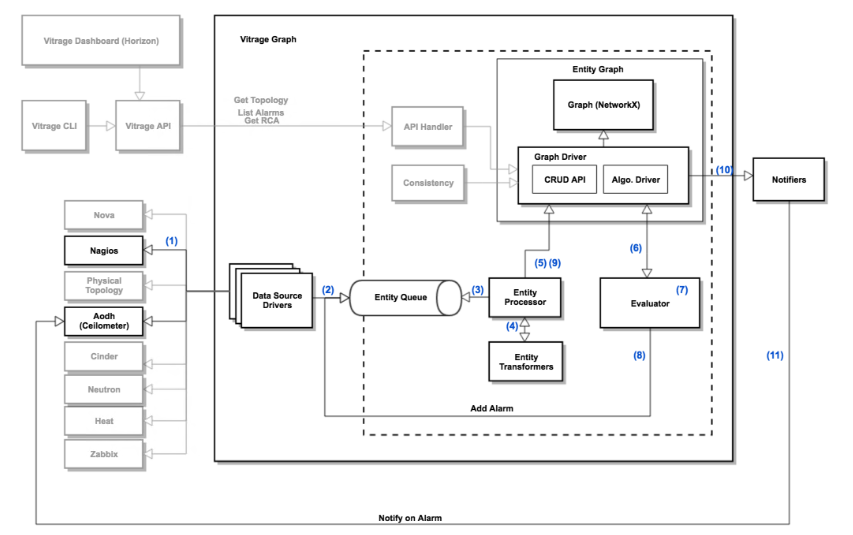


**Kết quả: zone nova có thêm 2 instance lily và luna, nó được cập nhật từ datasource nova, neutron. (Các hình mũi tên chỉ neutron port - hiện đang tắt)**



Khi aodh thêm alam:

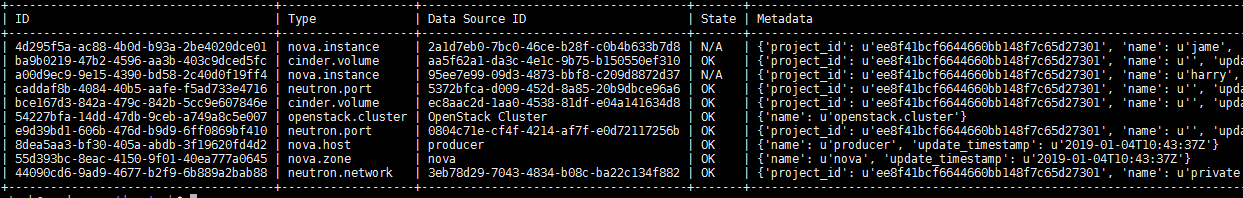
Khi có nagios - zambix alam:



Evaluator + template gan voi nhau

van deo viet dc template viet tha nao aa!!!!!! wtf man

Resource:



guilde cai dat:

<https://github.com/openstack/vitrage-dashboard>

<https://docs.openstack.org/vitrage/latest/install/install-rdo.html>