Batch System as a Service

The issue

Dynamically instantiate a computing cluster exploitable through the batch jobs submission

---

- Provide all the main batch system functionalities such as queues, resource management, fairshare priority, policies,…

- Provide resource for batch job execution in a dynamic way

BSaaS, first use case

I already have a batch system.

I just need to stretch/shrink computing capabilities

We have evaluated two solution

- CloudScheduler

- HTCondor rooster

CloudScheduler, one main limit

One can use a scheduling algorithm in either Condor Job Scheduler or Cloud Scheduler but one consequence of the current design is that Cloud Scheduler can impact the scheduling decision made by the Condor Job Scheduler(and vice versa). In the simplest case where Condor use FIFO algorithm it may be possible that job are not run in expected order under certain circumstances

HTCondor Rooster

condor_rooster is an optional daemon that may be added to the condor_master daemon's DAEMON_LIST. It is responsible for waking up hibernating machines when their UNHIBERNATE expression becomes True

HTCondor Rooster

UNHIBERNATE = TotalIdleJobs > 100

↓

ROOSTER_WAKEUP_CMD = /usr/share/submitEc2.py

↓

```
#!/usr/bin/env python
import boto
....
```

↓

```
root@wn-recas-uniba-18:~# nova  --os-tenant-name CondorCloud list
+--------------------------------------+----------------------------------------------+--------+------------+-------------+-----------------------------+
| ID                                   | Name                                         | Status | Task State | Power State | Networks                    |
+--------------------------------------+----------------------------------------------+--------+------------+-------------+-----------------------------+
| 1d77a66b-079b-43af-8d2d-05004cb05398 | Server 1d77a66b-079b-43af-8d2d-05004cb05398 | ACTIVE | None       | Running     | public-net=90.147.102.176 |
| 1ed4dbb1-e9c8-4fec-a190-f493aacd9728 | Server 1ed4dbb1-e9c8-4fec-a190-f493aacd9728 | ACTIVE | None       | Running     | public-net=90.147.102.174 |
| 213e6b26-ce10-4a4b-baa3-8ef7204e1f6e | Server 213e6b26-ce10-4a4b-baa3-8ef7204e1f6e | ACTIVE | None       | Running     | public-net=90.147.102.179 |
| 2869d080-3eca-4a00-b43a-a476a073d2ec | Server 2869d080-3eca-4a00-b43a-a476a073d2ec | ACTIVE | None       | Running     | public-net=90.147.102.177 |
| 8000656f-eb78-432e-9701-aff7543adafa | Server 8000656f-eb78-432e-9701-aff7543adafa | ACTIVE | None       | Running     | public-net=90.147.102.178 |
| 8b508a73-7ef7-4412-a8cf-673863bf610f | Server 8b508a73-7ef7-4412-a8cf-673863bf610f | ACTIVE | None       | Running     | public-net=90.147.102.170 |
| 8fbdb237-4eb2-4426-b60e-a61c7f20397b | Server 8fbdb237-4eb2-4426-b60e-a61c7f20397b | ACTIVE | None       | Running     | public-net=90.147.102.175 |
| e32b0288-1af3-45ca-8432-5ef445fb5881 | Server e32b0288-1af3-45ca-8432-5ef445fb5881 | ACTIVE | None       | Running     | public-net=90.147.102.173 |
| f1978b6f-c5bc-4008-95ae-e38bbfd8c567 | Server f1978b6f-c5bc-4008-95ae-e38bbfd8c567 | ACTIVE | None       | Running     | public-net=90.147.102.171 |
+--------------------------------------+----------------------------------------------+--------+------------+-------------+-----------------------------+
```

HTCondor Rooster, limitations

- *VMs instantiate with the same ec2 credentials*
- *A Idle VM can be shotted down at the most*

BSaaS, second use case

I need a batch facility just for five months

BSaaS, second use case

Requirements

- batch system service provisioning

- batch system configuration

- dynamic computing host provisioning

BaaS: Meet the requirements

- batch system service provisioning

- Heat orchestrates the batch system resource provisioning

- batch system configuration

- dynamic computing host provisioning

BaaS: Meet the requirements

- batch system service provisioning

- Heat can orchestrate the batch system resource provisioning

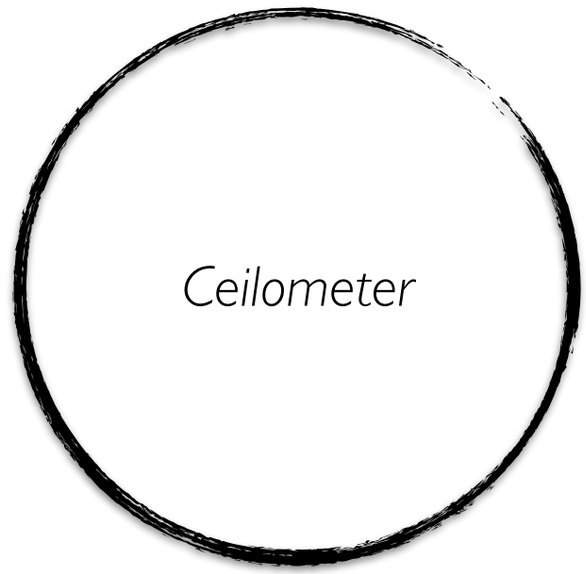
- batch system configuration

Puppet manages batch system installation and configuration

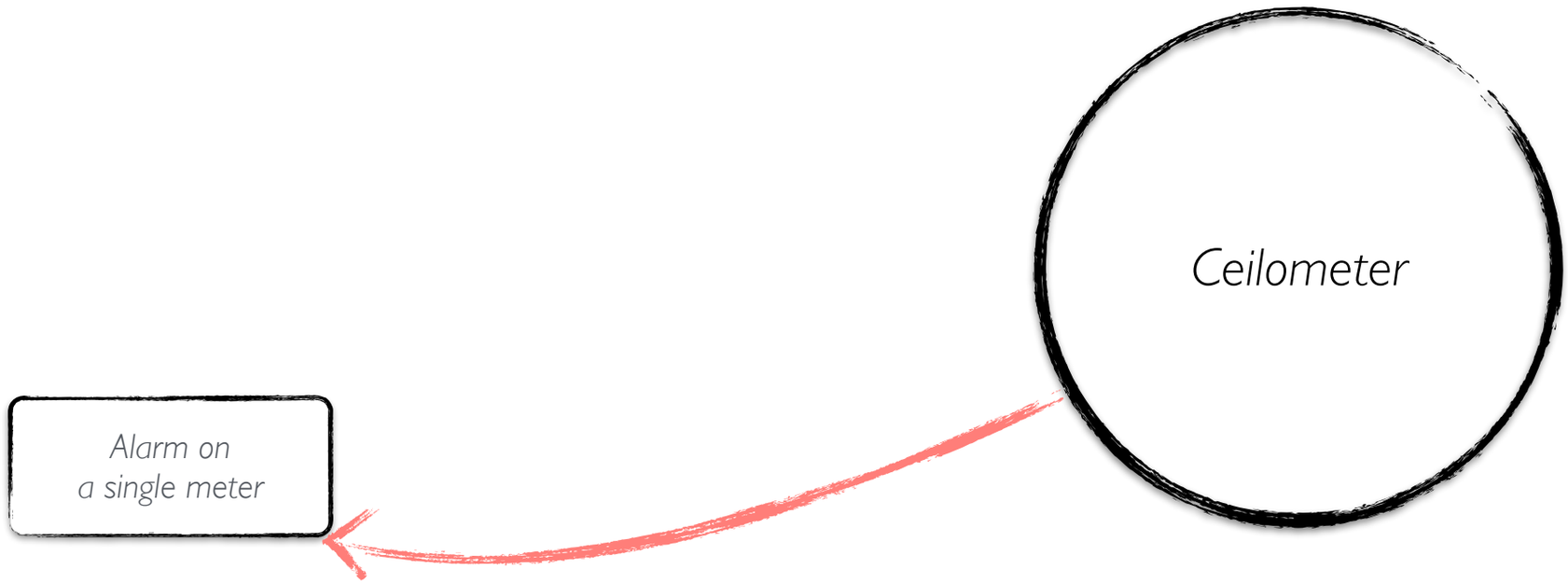
- dynamic computing host provisioning

Ceilometer Alarming

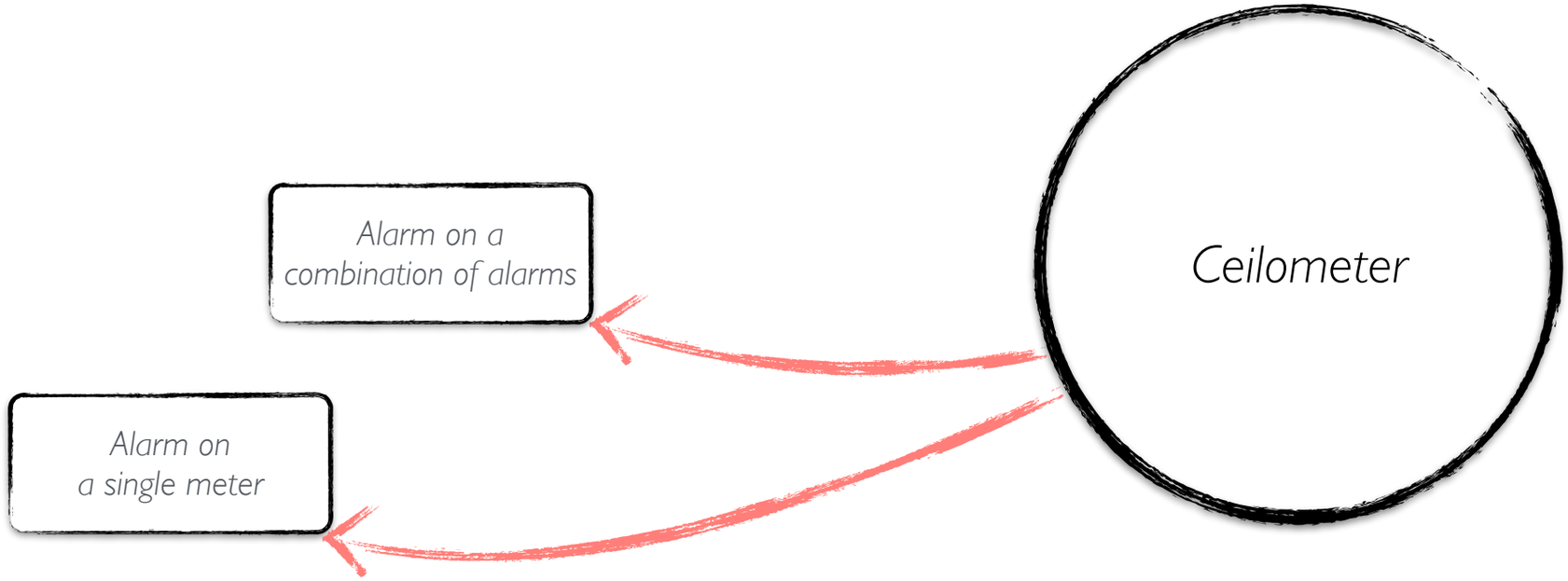
Ceilometer Alarming



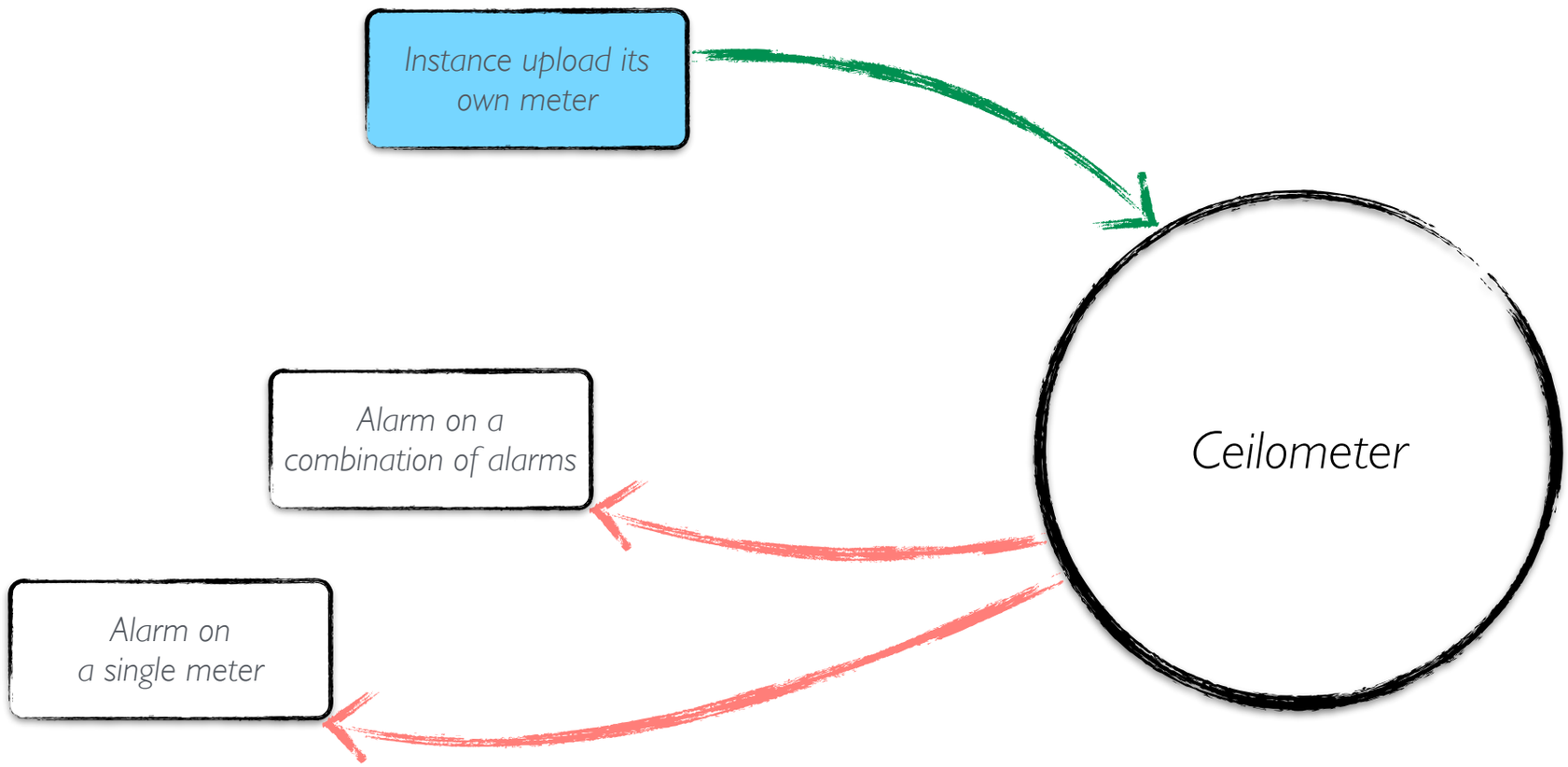
Ceilometer Alarming



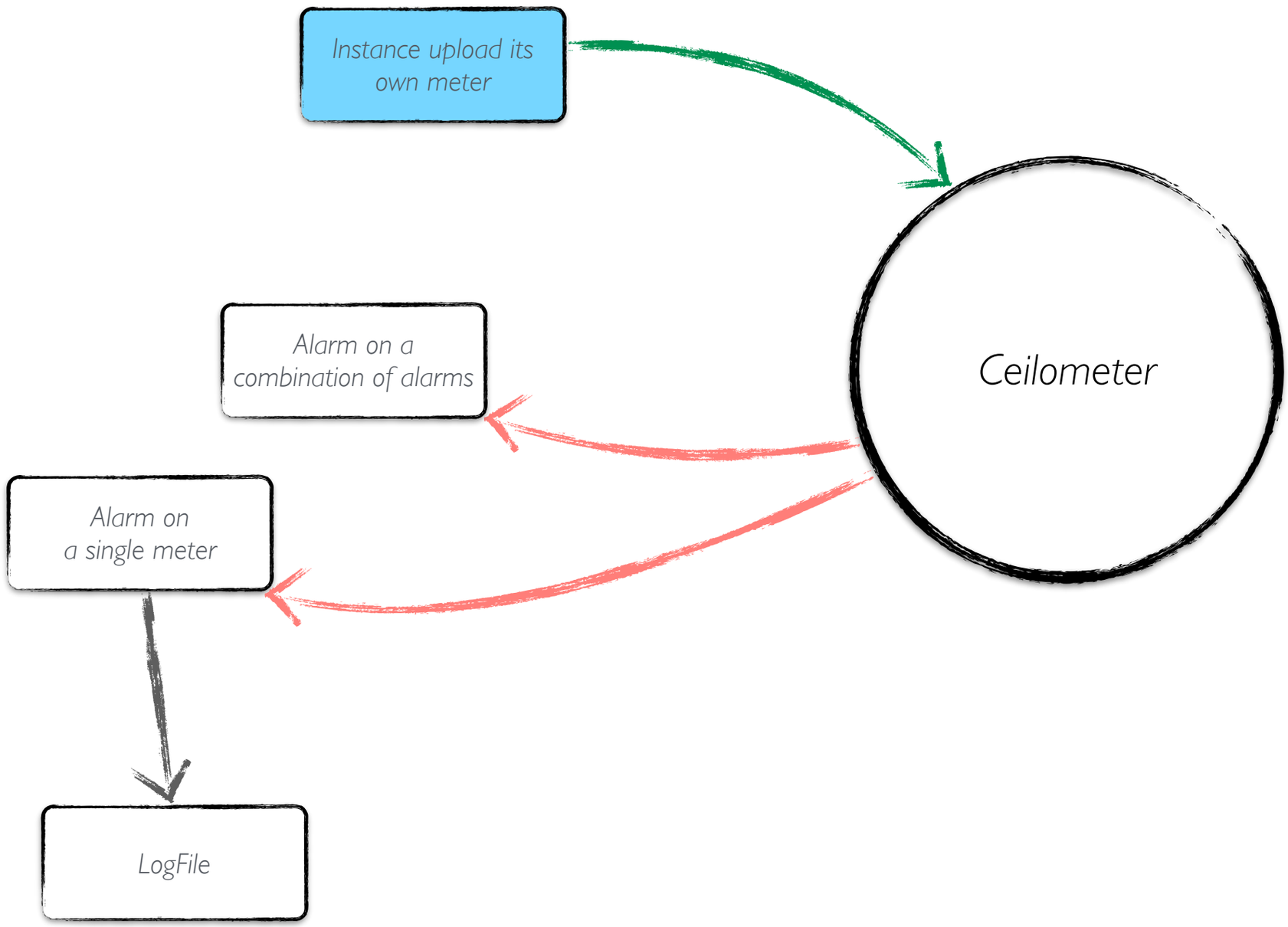
Ceilometer Alarming



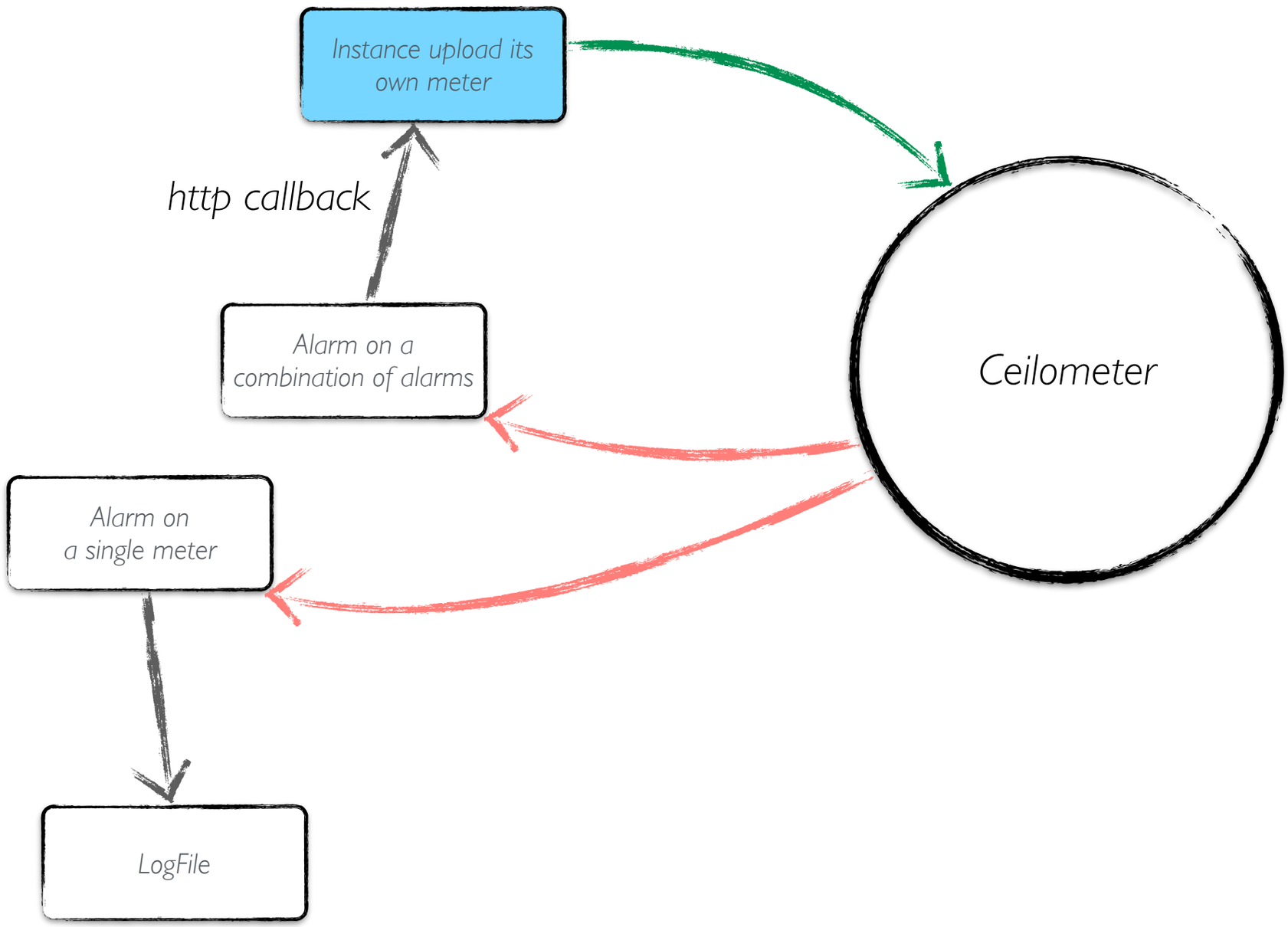
Ceilometer Alarming



Ceilometer Alarming



Ceilometer Alarming



BaaS: Meet the requirements

- batch system service provisioning

- Heat can orchestrate the batch system resource provisioning

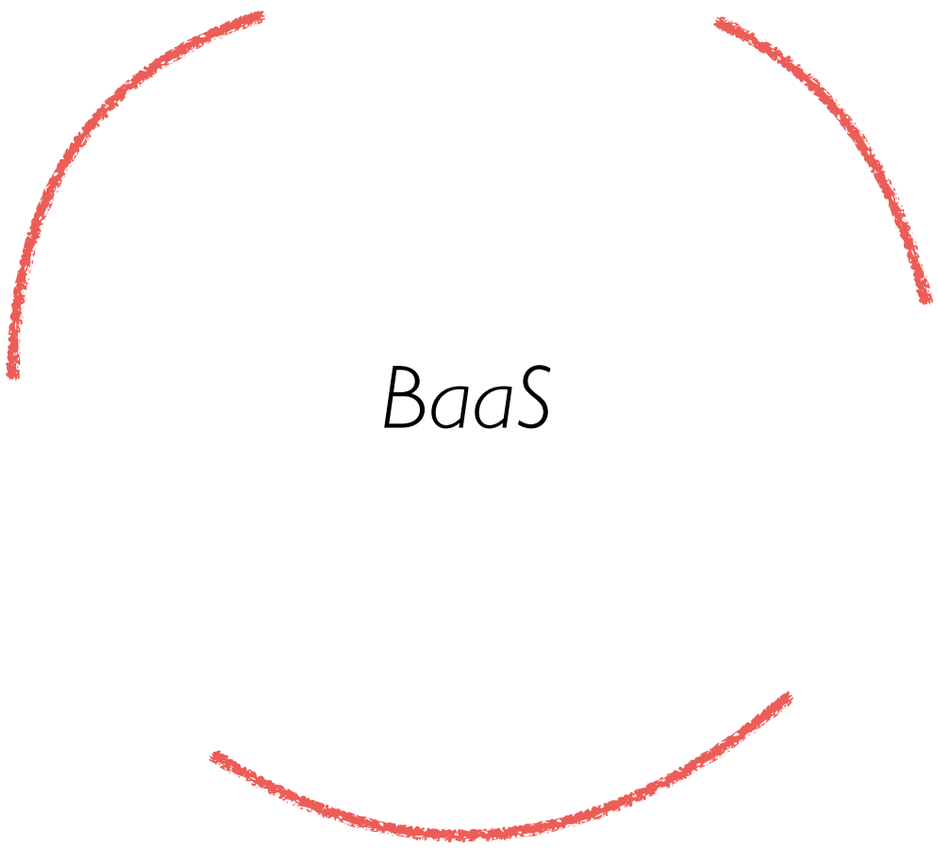
- batch system configuration

Puppet manages batch system installation and configuration

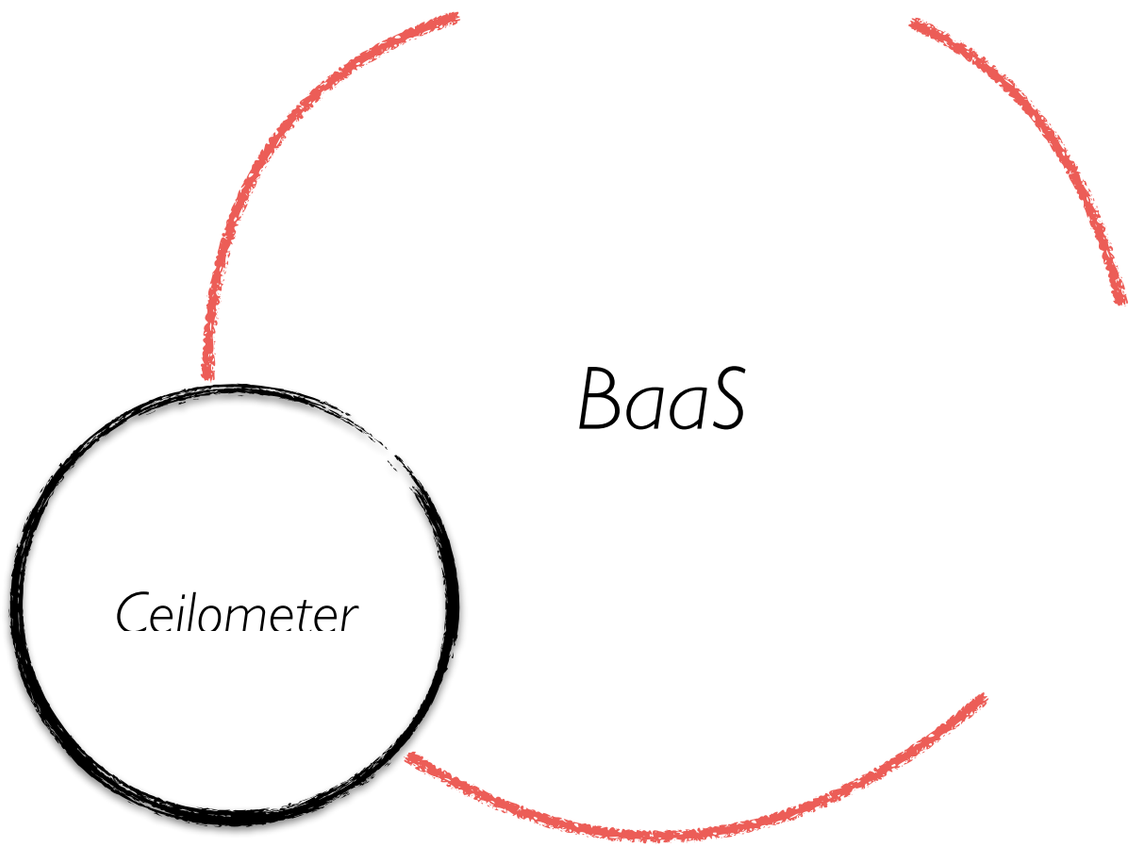
- dynamic computing host provisioning

Heat autoscaling based on IdleJobs alarm

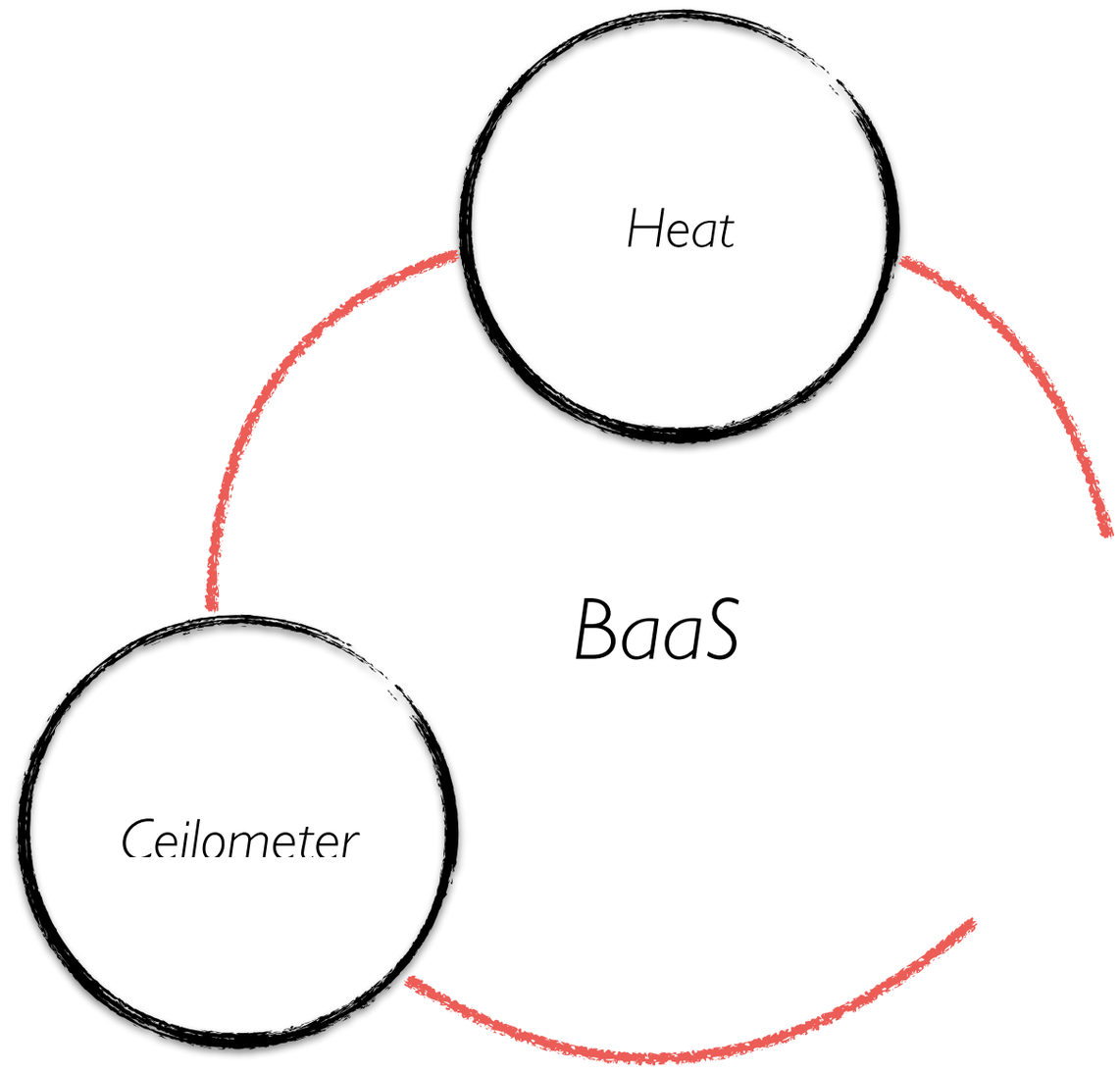
BaaS: Meet the requirements



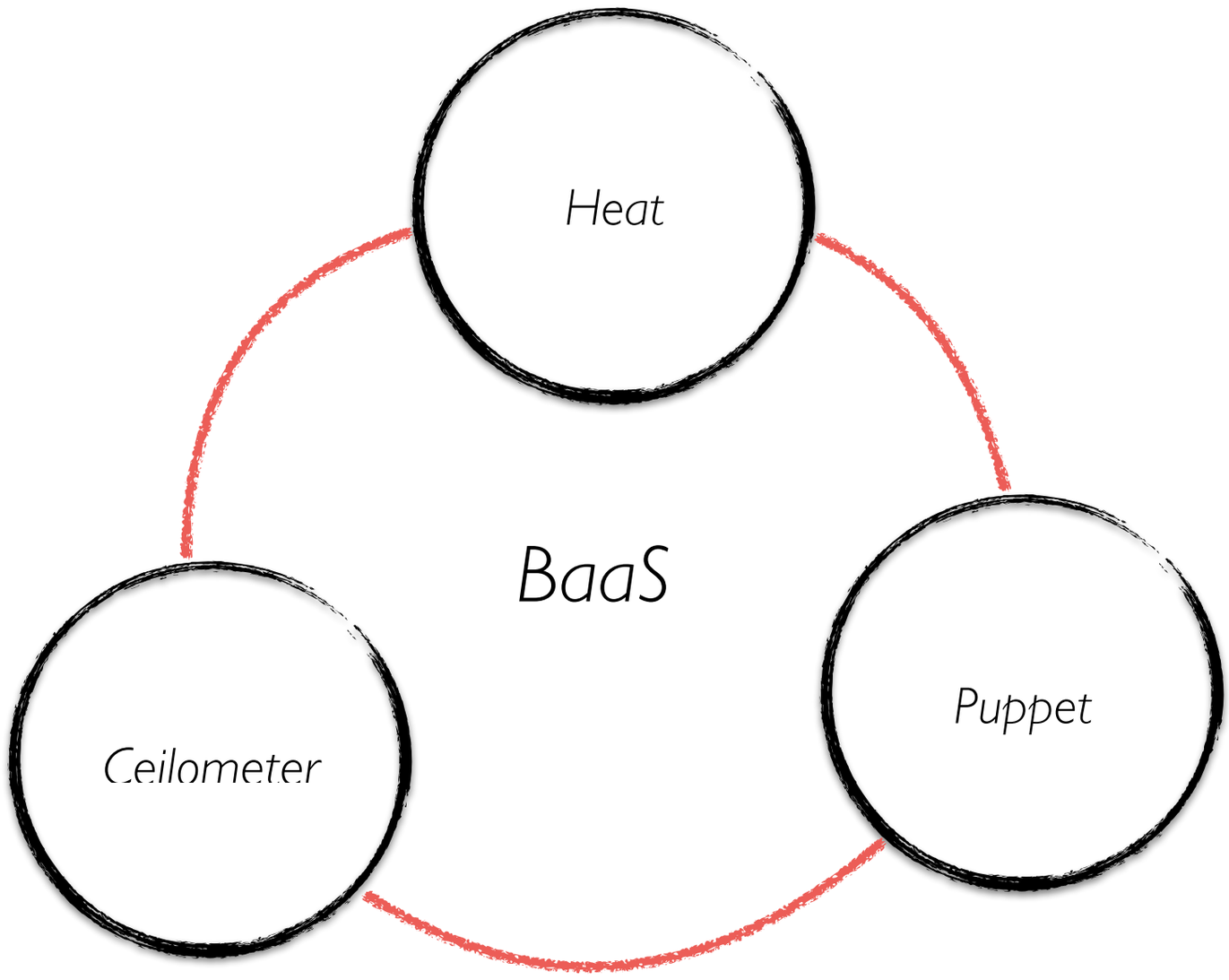
BaaS: Meet the requirements



BaaS: Meet the requirements



BaaS: Meet the requirements



PRISMA: HPC as a Service

Perché HPCaaS

- Uno degli use-case (eSeismic) di PRISMA richiede l'esecuzione di applicazioni che richiedono potenza di calcolo
  - E necessitano di brevi (certi) tempi di esecuzione
- Lo use-case di PRISMA può essere schematizzato come un SaaS.
  - Eseguire un software ben noto in modo semplice, affidabile, scalabile e ripetitivo.
- L'orchestatore di livello PaaS interagisce con i servizi esterni via REST API
  - Quindi dobbiamo fornire un'interfaccia di alto livello anche per HPCaaS.

Come è implementato

- Partire da oggetti già disponibili per ridurre il costo di sviluppo e di mantenimento:
  - **JST**: già usato in BioVeL per attività simili
    - Sviluppato e mantenuto da Bari
  - **Celery**: per distribuire le task all'interno degli host virtuali per una specifica applicazione di eSiesmic (OpenQuake)
- **HTCondor:** per le applicazioni in cui serve un batch system
- **Elastiq**: per la gestione delle macchine virtuali
  - Usato nella VAF di Alice, al momento interfacciato solo con HTCondor
    - Già aggiunto il supporto per celery e in fase di progettazione il supporto per JST

Schema logico

Orchestratore PaaS

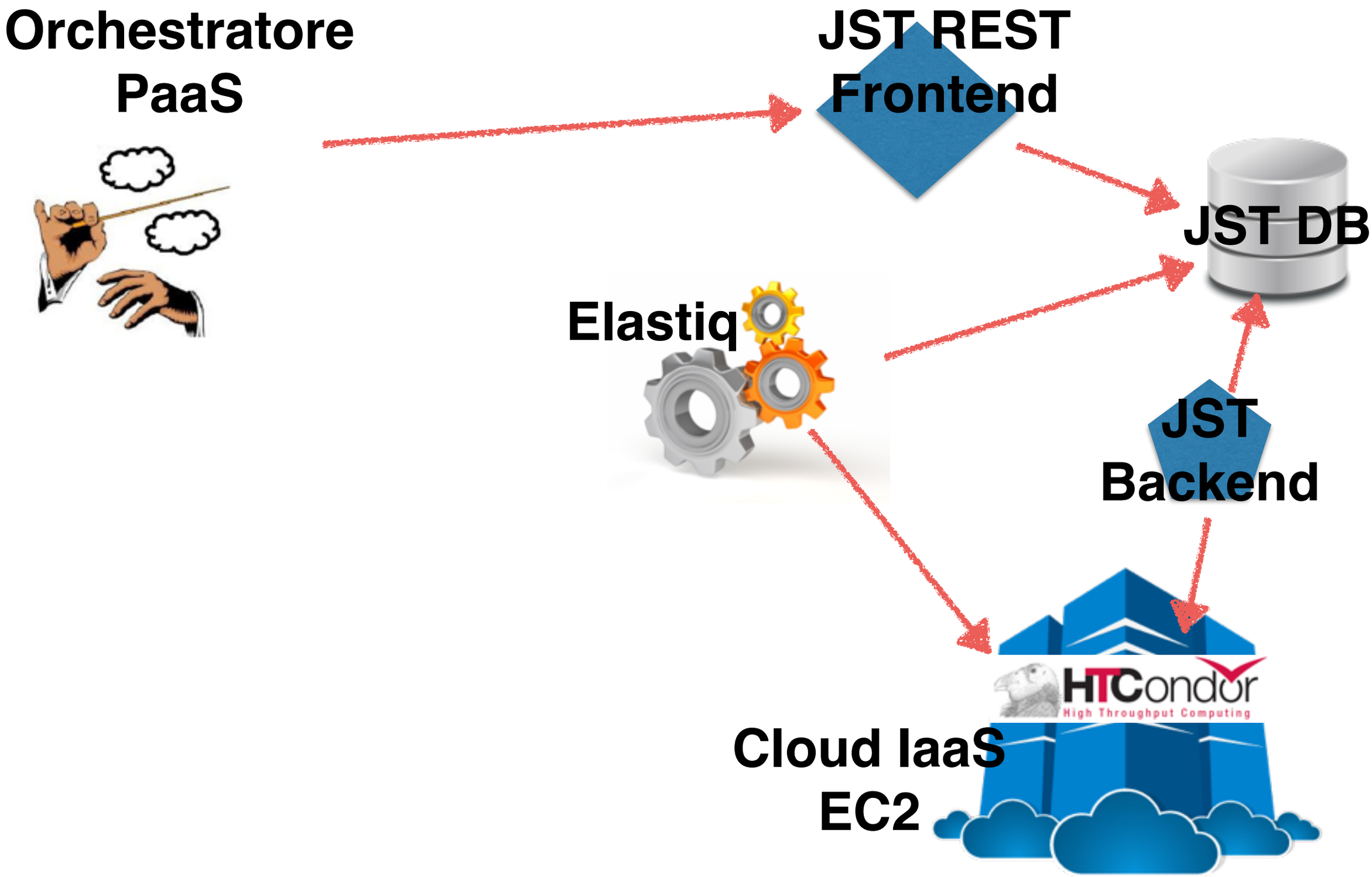
JST REST Frontend

JST DB

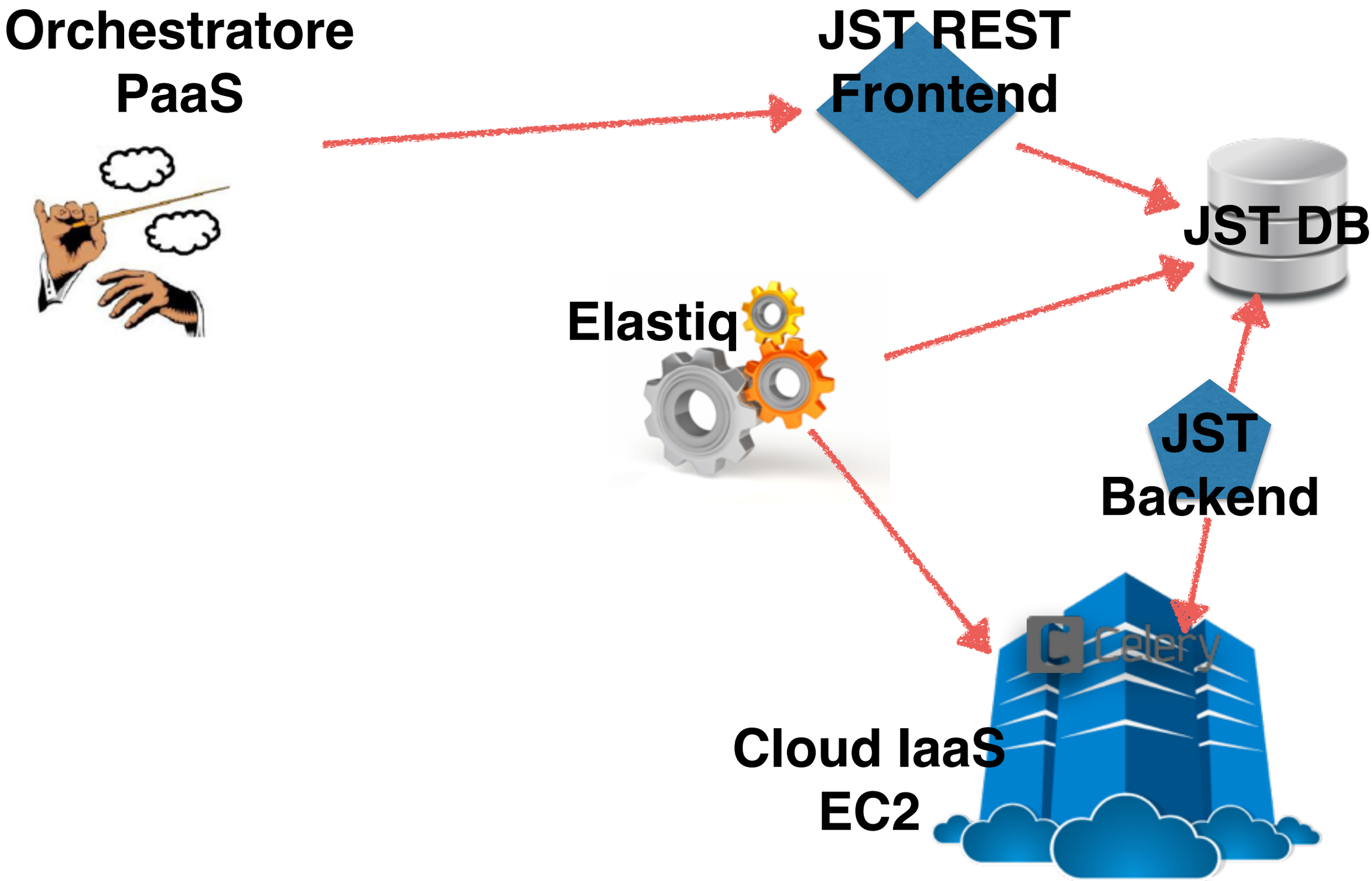
Elastiq

Cloud IaaS EC2

Schema logico



Schema logico



Referenze

- **Celery:**
  - http://www.celeryproject.org
  - https://github.com/celery/celery/wiki
- **Elastiq:**
  - https://github.com/dberzano/elastiq
  - https://agenda.infn.it/getFile.py/access?contribId=24&resId=0&materialId=slides&confId=8149