From Ceilometer to Telemetry

Not so alarming!

A *Julien Danjou* & *Nick Barcet* presentation

for
OpenStack in action! 4
on the *5th December 2013*

eNovance

Speakers

Nick Barcet VP Products @ eNovance

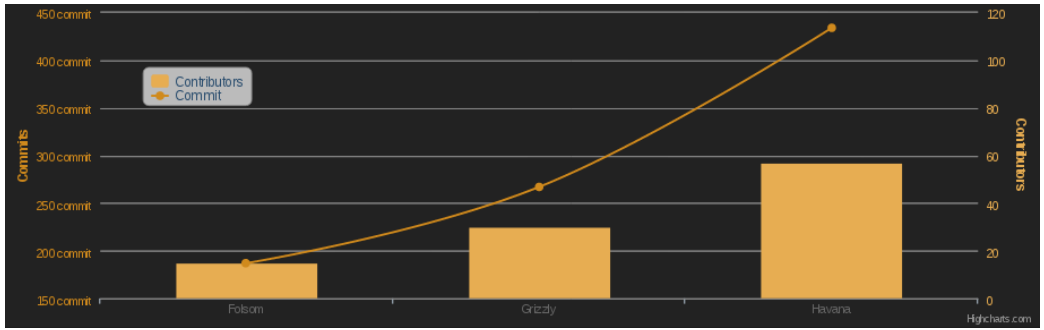
Co-founded the Ceilometer project at the Folsom summit and led the project through incubation

Julien Danjou Ceilometer Lead Dev @ eNovance

Has been a core Ceilometer contributor from the outset, taking over the PTL reins for Havana

State of the project

- Officially named **OpenStack Telemetry**
- *Havana* is the first integrated release
- Community growth
  - Grizzly: 30 contributors, 267 commits
  - Havana: 57 contributors, 434 commits



What was done during the *Havana* cycle?

UDP transport

- Faster, stateless
- Lighter (*msgpack* encoding)

but…

- No delivery guaranteed
- Not signed

▶ Use case: gathering metrics for alarms

Improved API

- Group samples by fields when requesting statistics (?groupby[]=user_id)
- Limit the number of items returned (?limit=42)
- Provides links to other resources in the API

Send your own samples

Users or operators can send samples

- ➔ Leverage the statistics
- ➔ Usable for alarming

```
POST /v2/meters/mymeter

[{
  "counter_type": "gauge",
  "counter_unit": "megabyte",
  "counter_volume": 142.0,
  "user_id": "efd87807-12d2-4b38-9c70-
5f5c2ac427ff",
  "project_id": "35b17138-b364-4e6a-a131-
8f3099c5be68",
  "resource_id": "bd9431c1-8d69-4ad3-803a-
8d4a6b89fd36",
  "resource_metadata": {
      "name1": "value1",
      "name2": "value2"
  },
  "source": "mypaasplatform",
  "timestamp": "2013-09-10T20:34:13.711330"
}]
```

New storage backends

Database TTL

Previously:

No way to purge data.

Ceilometer produces a lot of data (gigabytes per day)

Now:

ceilometer-expirer will drop data older than the configured time-to-live delay

Hyper-V

Windows Server®
Hyper-V™

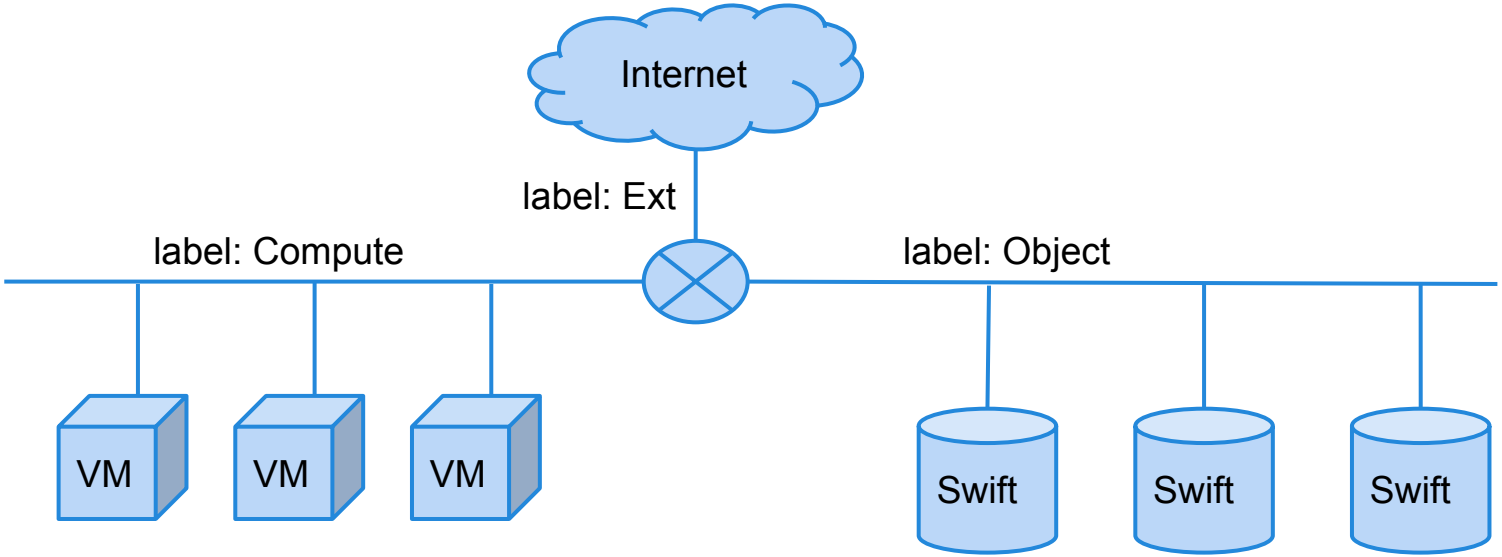
➔ Disk, network and CPU usage

eNovance

New meters

- API endpoints
  - Meters the requests made to API server (Neutron, Glance, Nova, Swift, etc)
- Neutron bandwidth
  - Meter the bandwidth consumed by each project
  - Traffic labeled as configured by operator (based on source/destination)

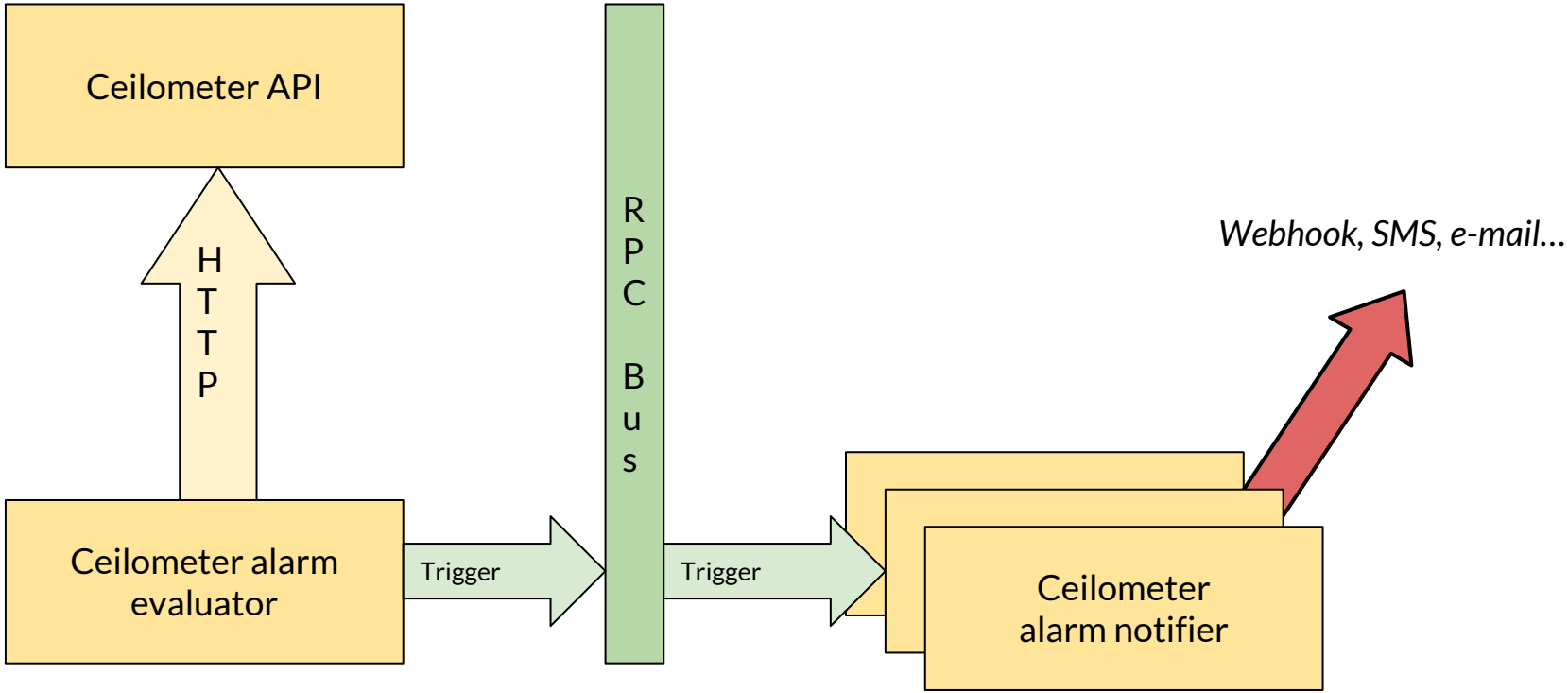
Neutron Traffic Labels



Alarms

Regularly watch for meters statistics values and triggers actions based on threshold crossings.

Alarms architecture



Alarm types

- ***Threshold* alarms**
  Triggered once a value crosses a threshold
  “Call a Webhook as soon as CPU usage goes above 80%”

- ***Combination* alarms**
  Triggered once all alarms in that alarm are triggered
  *“Call a Webhook as soon as alarm “foo” **and** alarm “bar” are triggered”*

Alarms API

POST /v2/alarms

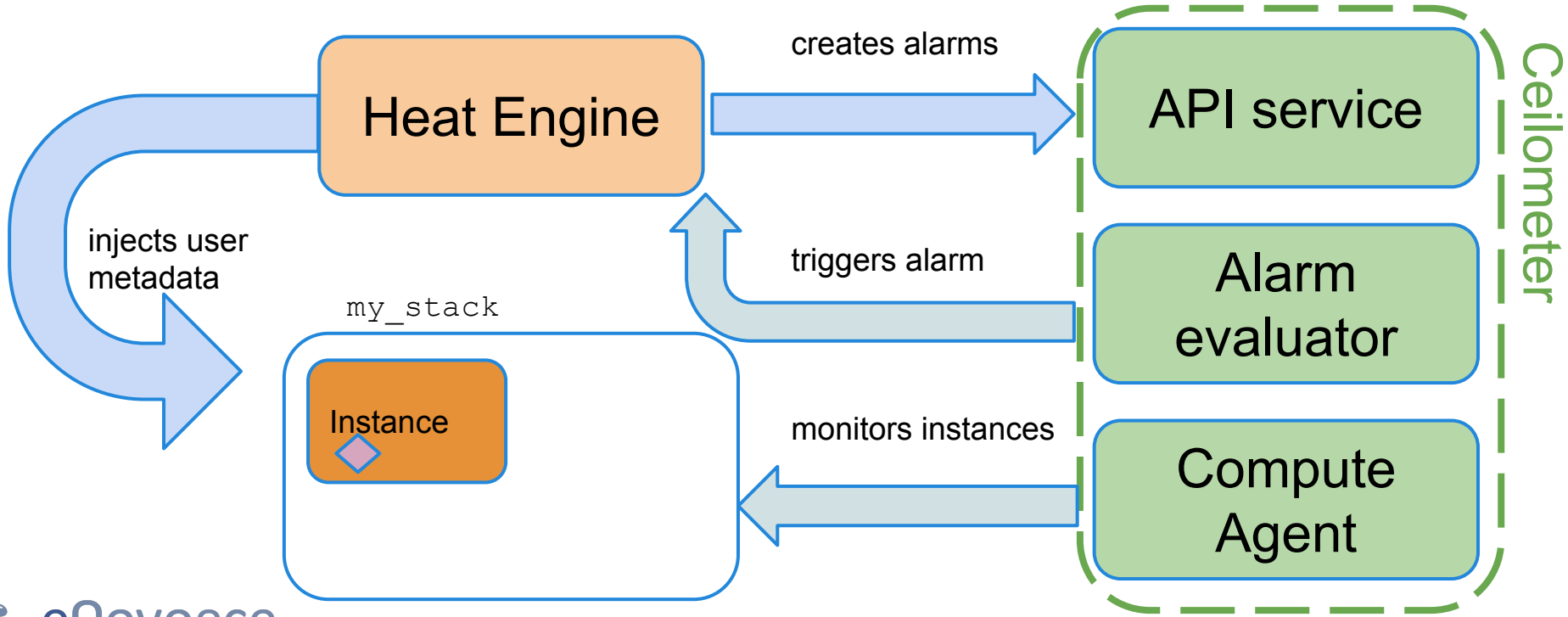
```
{
 "alarm_actions": [ "http://site:8000/alarm"],
 "insufficient_data_actions": ["http://site:8000/nodata"],
 "ok_actions": ["http://site:8000/ok"],
 "comparison_operator": "gt",
 "description": "An alarm",
 "evaluation_periods": 2,
 "matching_metadata": {"key_name": "key_value"},
 "meter_name": "storage.objects",
 "name": "SwiftObjectAlarm",
 "period": 240,
 "statistic": "avg",
 "threshold": 200.0
}
```

GET /v2/alarms/foobar

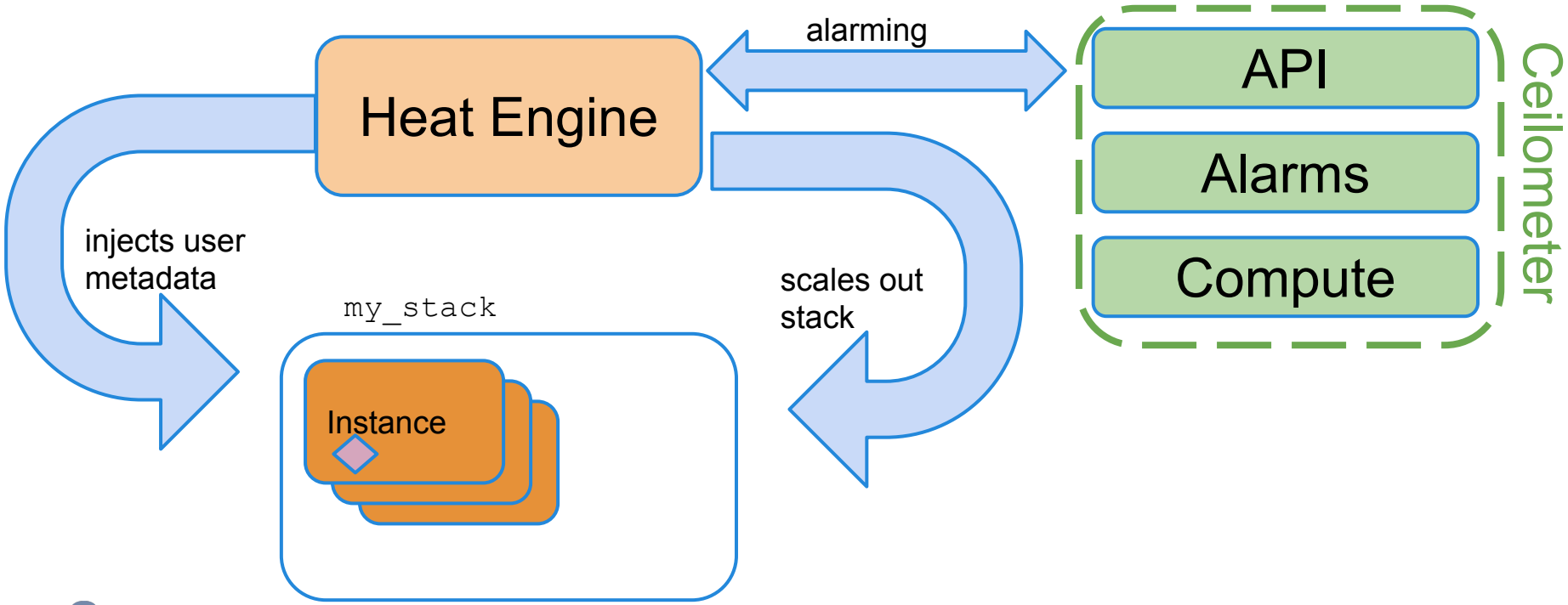
PUT /v2/alarms/foobar

DELETE /v2/alarms/foobar

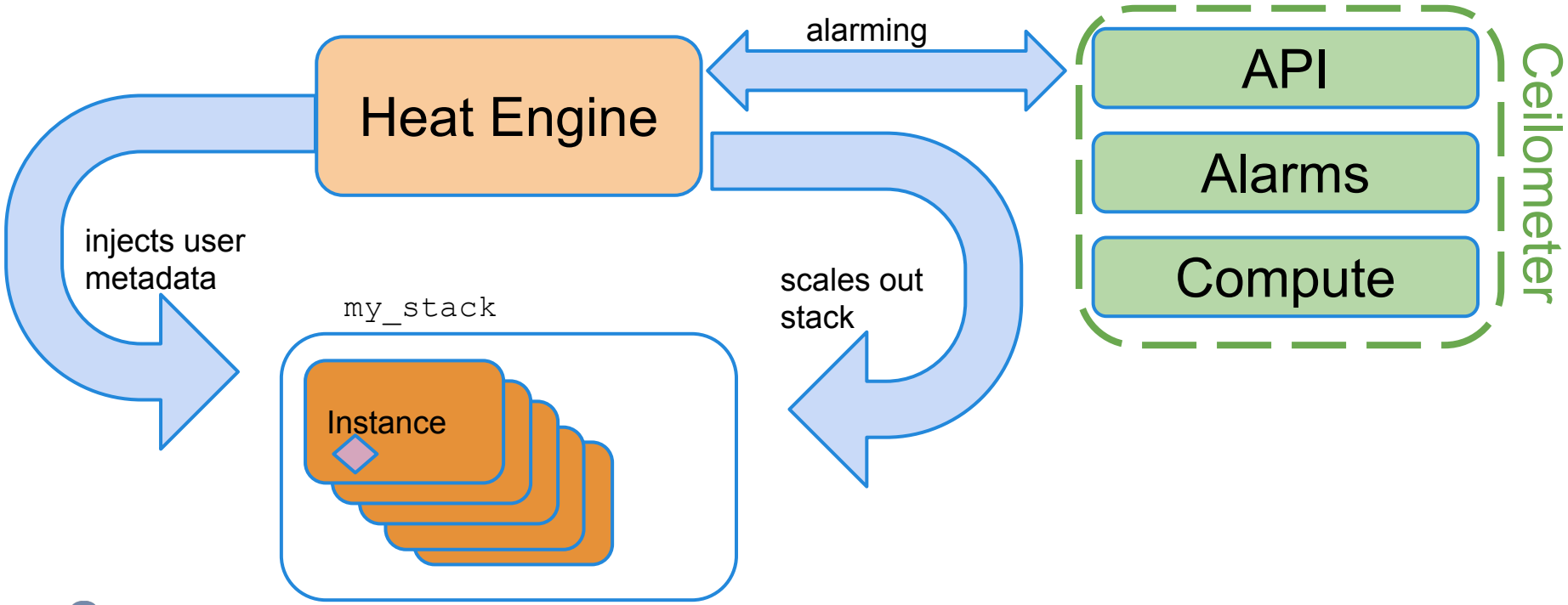
Heat & auto-scaling



Heat & auto-scaling



Heat & auto-scaling



Events storage

(Almost) all OpenStack components send notifications on events: let's store them.

- ➔ Useful to be able to re-generate samples
- ➔ Useful to generate new sample we did not think about
- ➔ Allow to have a double-entry accounting
- ➔ Audit ability

Not yet complete, to be continued in *Icehouse*

Exciting ideas for *Icehouse* we're going to hack on.

General improvements

- Split the collector in two logical pieces
- Rely on notification for samples rather than RPC
- Bring SQLAlchemy and MongoDB driver almost on parity
- Support for hardware polling
- Support Ironic

API improvements

- Complex filtering and query DSL
  `x OR y AND z`
- `/v2/samples`
  (a.k.a. `/v2/meter` without the meter)
- Return rate rather than absolute value
- More statistics functions (rate of change, moving-window averages…)
- Bulk requests

Alarming

- Exclude low sample counts
- Allow time constrained alarms

Distributed polling

Leveraging *Tooz* and *Taskflow* to distribute tasks among workers (agents).

★ Ability to distribute the polling

★ Replace alarm evaluator custom distributor

OpenStack Telemetry

Ceilometer

#openstack-ceilometer @ Freenode

The end.

eNovance

Backup slides

Heat & auto-scaling