Julien Danjou
jd__@Freenode // @juldanjou
julien@danjou.info
Nick Barcet
nijaba@Freenode // @nijaba
nick@enovance.com

Ceilometer

The OpenStack Measurement Project

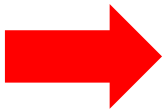
14 April 2013 @ ODS Havana

What About Billing?

Beginning of 2012:

- Billing has been left out of OpenStack core so far as it was not the primary problem and is not a trivial one...
- Yet almost every OpenStack deployment needs a way to track usage information

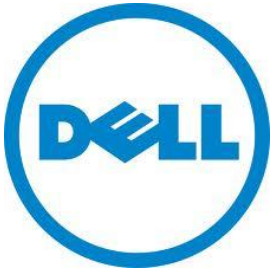
Billing: 3 Step Process



Metering	Collect usage data
Rating	Transform usage data into billable items and calculate costs
Billing	Create invoice, collect payment

Ceilometer Begins

- Started in May 2012



v0.1 Objectives (Folsom)

Focused only on metering for billing

Ceilometer aims to deliver a unique point of contact for billing systems to acquire all counters they need to establish customer billing, across all current and future OpenStack components.The delivery of counters must be traceable and auditable, the counters must be easily extensible to support new projects, and agents doing data collections should be independent of the overall system.

Grizzly objectives

Extended objective: cover measurement in general

The project aims to become the infrastructure to collect measurements within OpenStack so that no two agents would need to be written to collect the same data. It's primary targets are monitoring and metering, but the framework should be easily expandable to collect for other needs. To that effect, Ceilometer should be able to share collected data with a variety of consumers.

Grizzly objectives

Extended objective: cover measurement in general

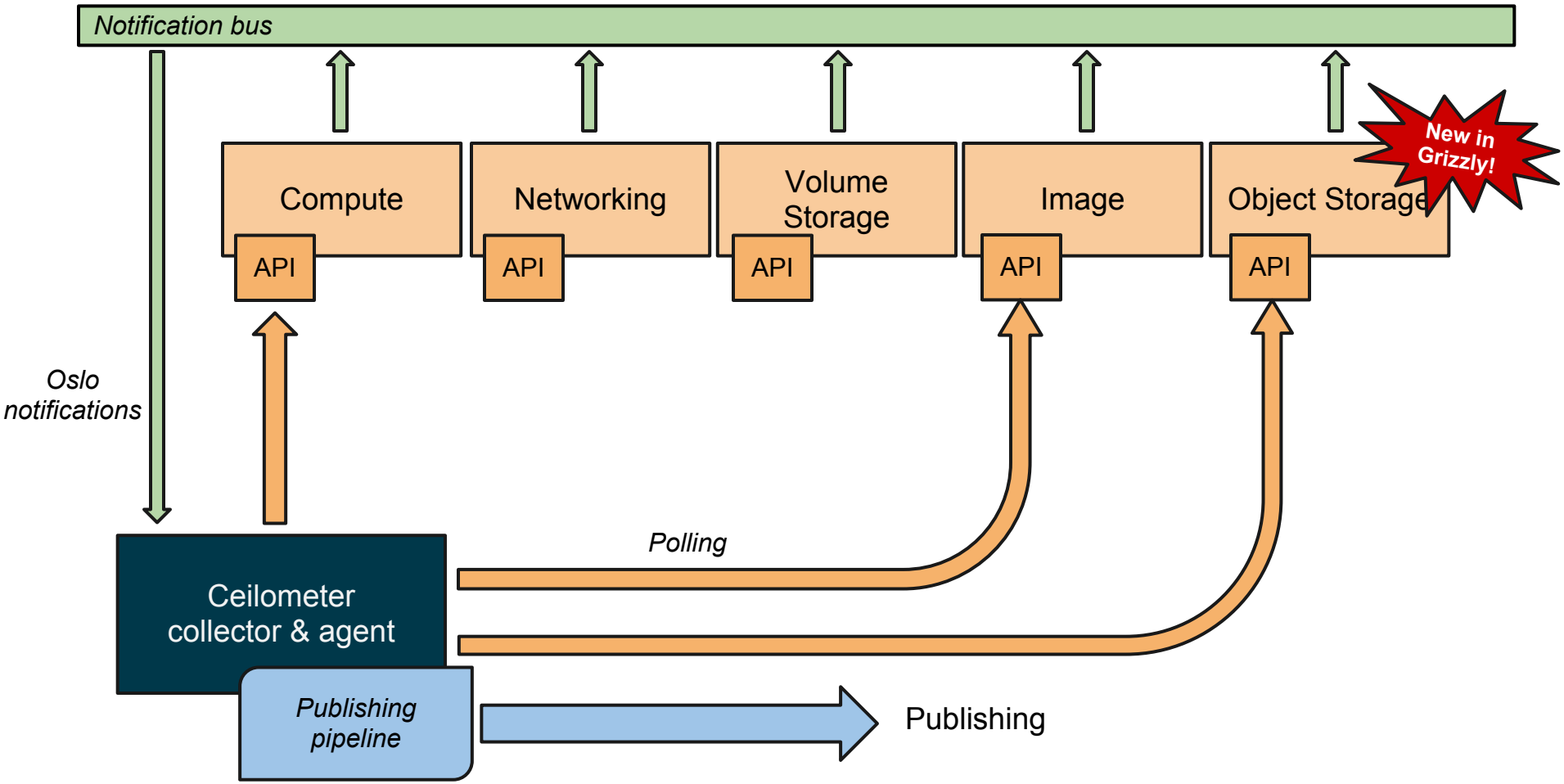
The project aims to become the infrastructure to collect measurements within OpenStack so that no two agents would need to be written to collect the same data. It's primary targets are monitoring and metering, but the framework should be easily expandable to collect for other needs. To that effect, Ceilometer should be able to share collected data with a variety of consumers.

Havana's objectives should remain unchanged!

Workflow

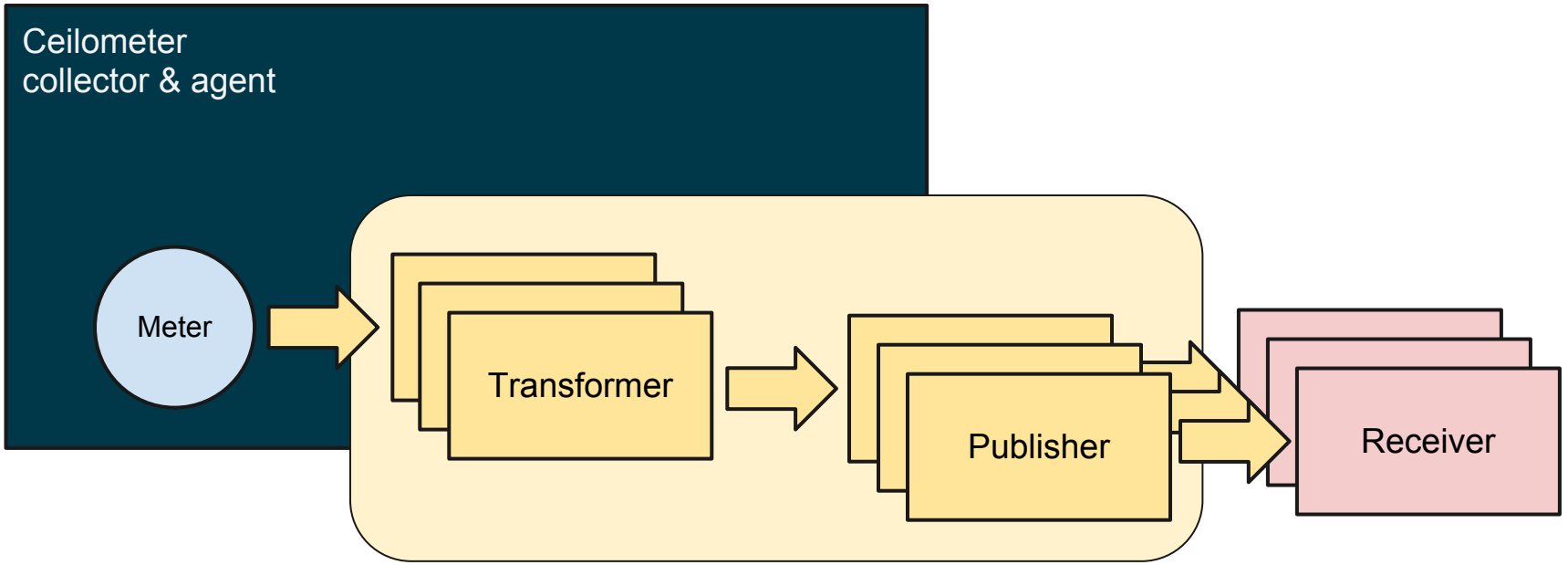
- Collect from OpenStack components
- Transform meters into other meters if necessary
- Publish meters to any destination (including Ceilometer itself)
- Store received meters and read them via the Ceilometer REST API

Collecting


Notification bus
Compute
Networking
Volume Storage
Image
Object Storage
New in Grizzly!
API
API
API
API
API
Oslo notifications
Polling
Ceilometer collector & agent
Publishing pipeline
Publishing

Pipeline

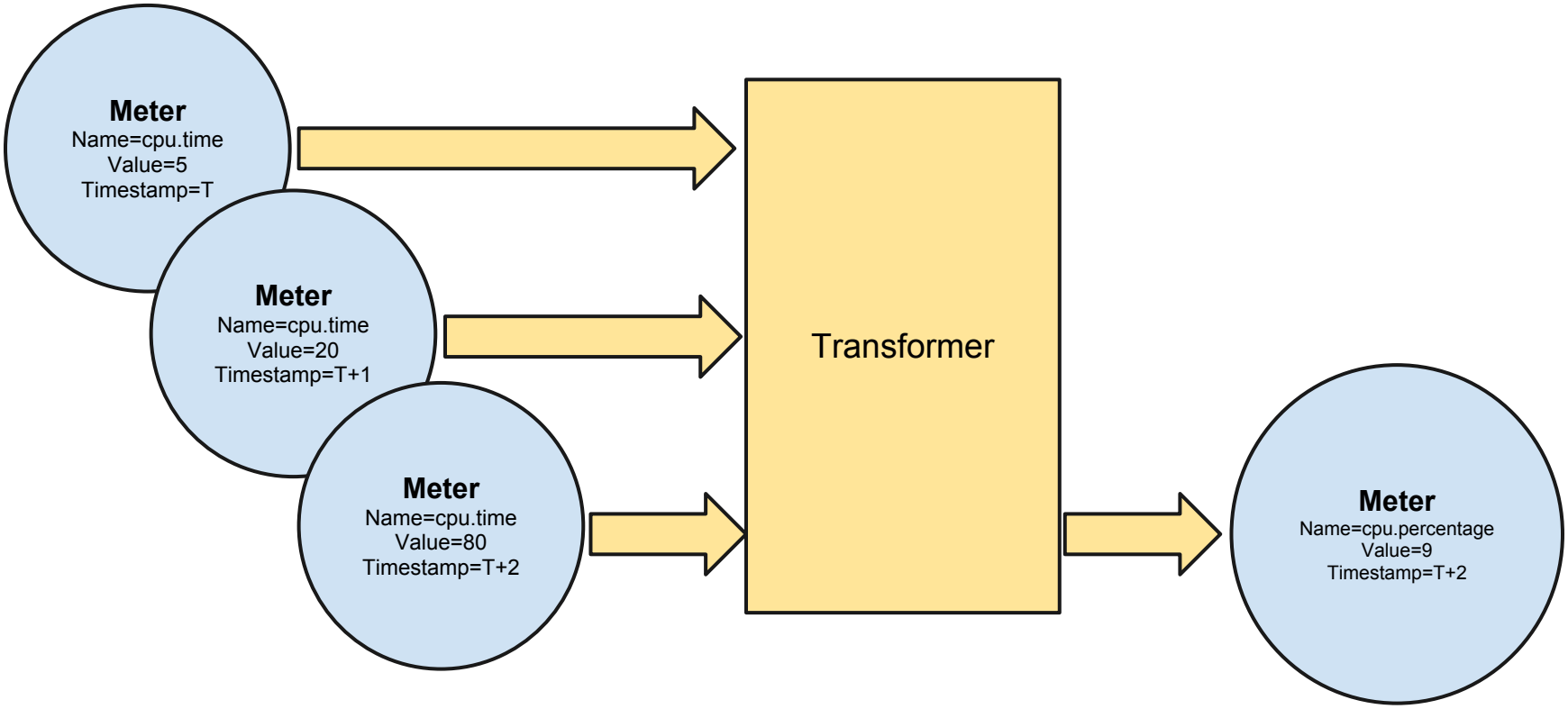
New in Grizzly!



Pipeline: a set of transformers mutating meters into something that publishers know how to send to external systems.

Transformer

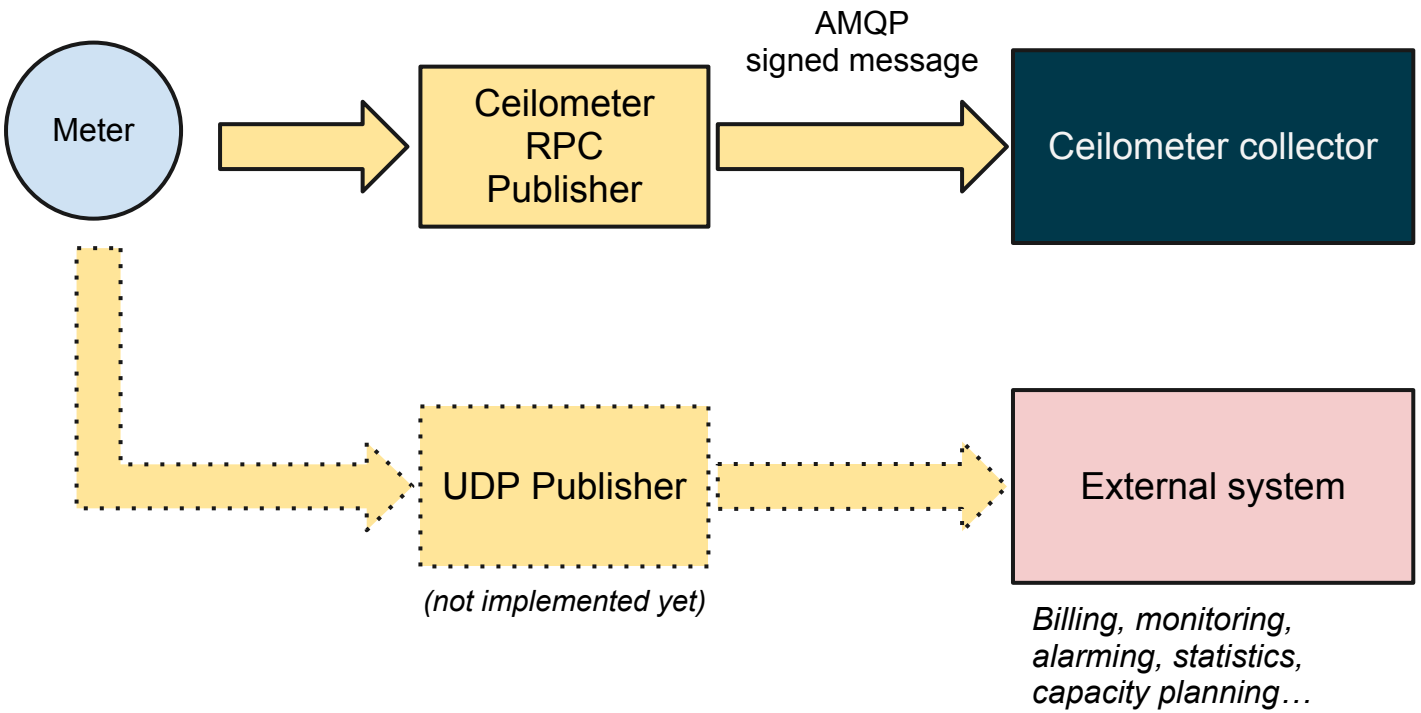
New in Grizzly!



Transform meters into new meters!

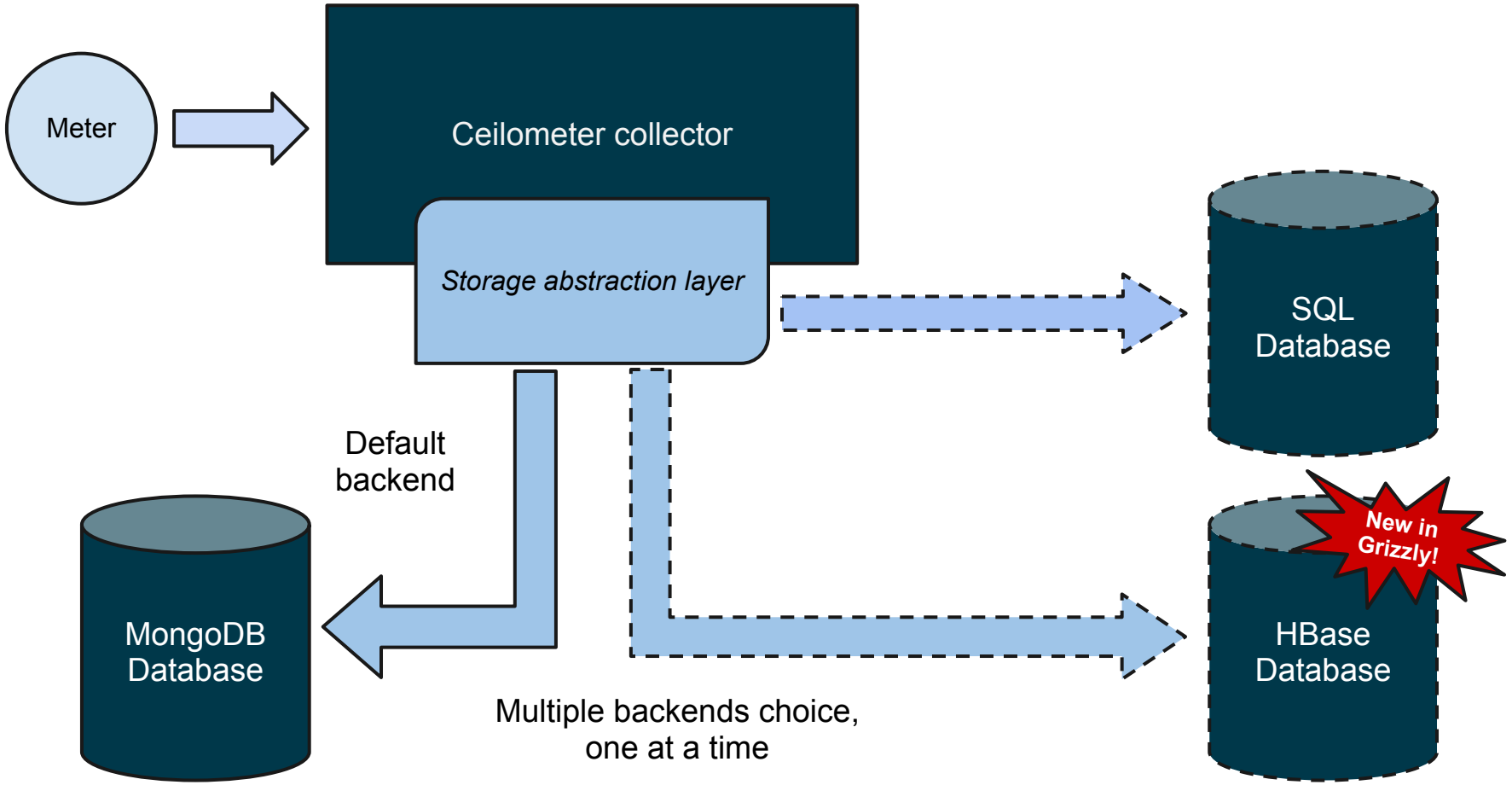
Publisher

New in Grizzly!

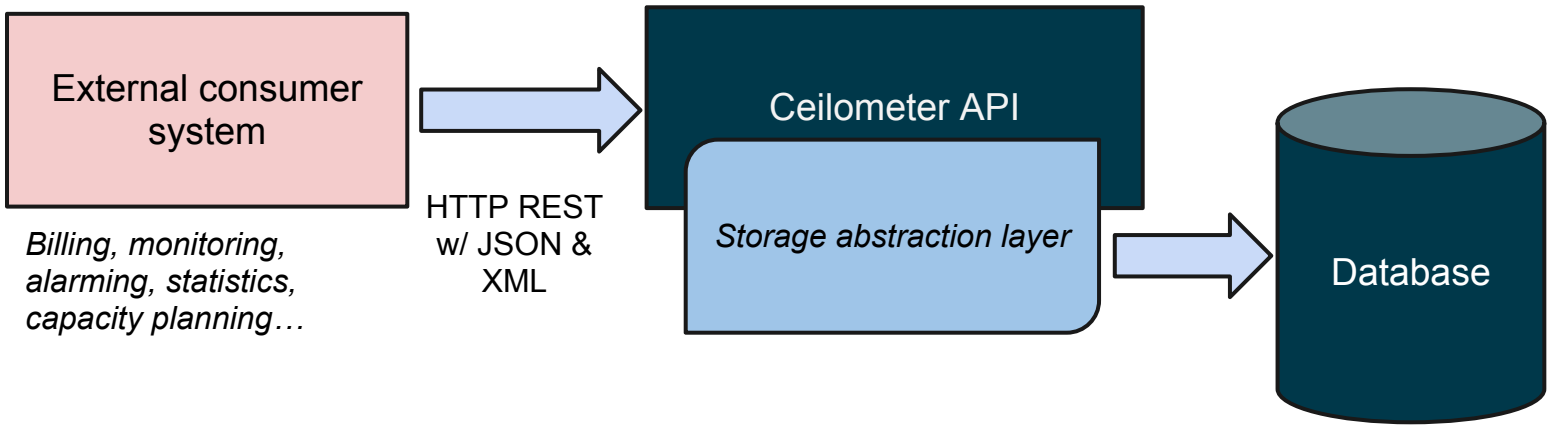


Yes, the frequency of the publication of each kind of meter is configurable for each publishing target.

Store: collector



Read: API



Raw events:
GET /v2/meters/cpu.time

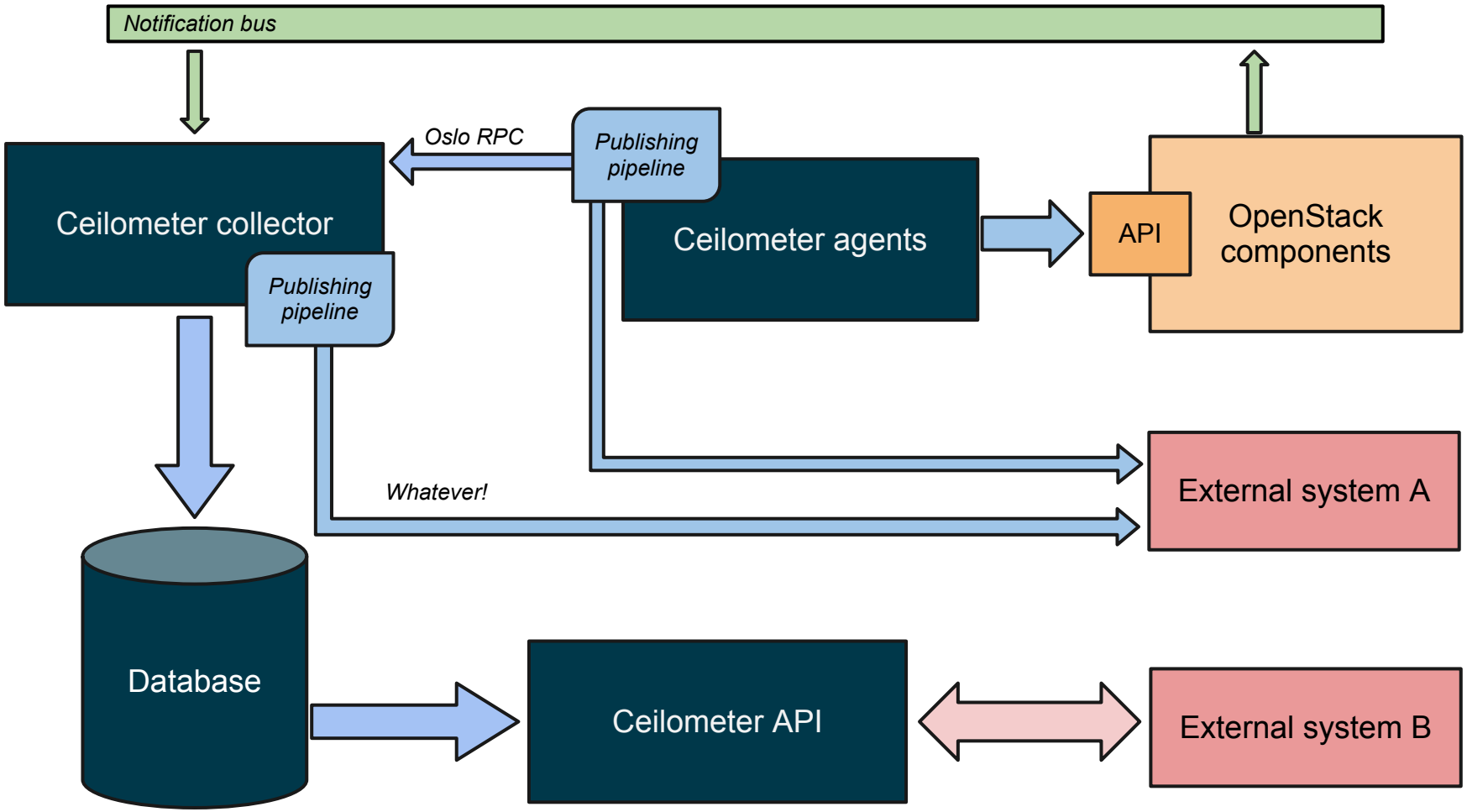
Statistics (sum, average, min, max…):
GET /v2/meters/cpu.time/statistics

New in Grizzly!

Filter:
GET /v2/meters/cpu.time?q.field=project_id&q.value=foobar&q.op=eq

Group statistics by period:
GET /v2/meters/cpu.time/statistics?period=3600

The big picture



Roadmap

Grizzly

- Incubated Project ✓
- Integration with Horizon □
- Agents for other components
  - Swift ✓
  - Ceph? □
  - Nicira? □
- SQLAlchemy storage driver ✓
- Multi-Publisher ✓
- API v2 ✓
  - User accessible API ✓
  - More aggregation ✓
  - Multi-dimension ✓

Havana

- Integrated Project ✓
- Integration with Horizon
- Publishing meters to other systems
- Enhance SQL driver
- Alarming
- Integration with Heat
- Deprecating APIv1
- Completing APIv2
- Move publishing part to Oslo and other projects
- Tighter integration with Nova
- Nova-scheduler integration?

I

- ?

Use cases

- Primary use-case is a rating/billing pipeline
- Analytics
  - capacity planning tools
  - adaptive scheduling algorithms
  - derive optimal pricing models
  - resource rationing with fuzzy quotas
- Realistic pre-prod simulations/loadtests
- Visualization
  - heat-maps/graphing to reveal usage patterns
- Monitoring
  - integration with diverse monitoring frameworks

Ceilometer @ design summit

Wednesday	16:30	Introduction to ceilometer architechture
	17:20	Feedback from Ceilometer users
Thursday	09:00	Incremental improvements grab-bag
	09:50	Double entry auditing of collected metrics
	11:00	API improvements
	11:50	Alarm Threshold Evaluation
	13:30	Time series data manipulation in nosql stores
	14:20	Simple messaging action for Alerting
	15:20	Alarm state and history management
	16:10	Ceilometer support for advanced billing models
	17:00	Supporting rich data types and aggregation

➡ See you in room B116

Questions?

http://launchpad.net/ceilometer
http://docs.openstack.org/developer/ceilometer
http://wiki.openstack.org/ceilometer

Freenode: #openstack-metering
Mailing List: openstack-dev [ceilometer]

Julien Danjou
jd__@Freenode // @juldanjou
julien@danjou.info

Nick Barcet
nijaba@Freenode // @nijaba
nick@enovance.com