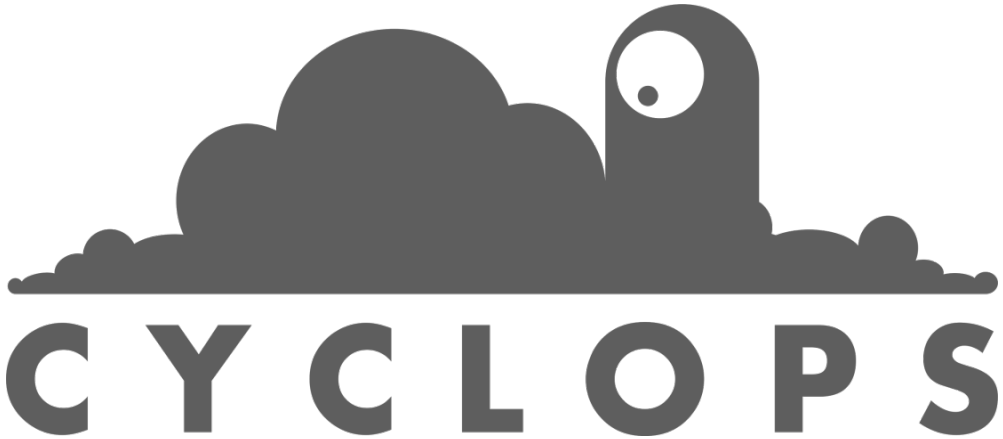


Versatile billing engine made for clouds and beyond..

Service Engineering Research Area
ICCLAB SPLAB

Dr. Piyush Harsh
http://piyush-harsh.info/

Senior Researcher, ICCLab
Zurich University of Applied Sciences

Talk Outline

What you will learn :)

- ★Quick primer
- ★Current Release Info
  - Architecture
  - Description of Micro services
  - Key innovations
  - Supported platforms
- ★Specific use-cases
  - EU FP7 Project: MCN
  - EU FP7 Project: T-Nova
  - Swiss National Project: SCALEUP
  - Swiss Transfer Project: SSC
- ★Short demo
- ★Future Plans
- ★Where to get this framework?

CYCLOPS

ICCLab @ ZHAW

We are a young lab - 4 Yrs old, mostly funded by EU and Swiss projects.

Rapid growth: from 3 founding members to ~25 researchers and still growing.

Highly diverse and international team

4 professors, 4 senior researchers, 6 researchers, 3 MS students, 1 Bachelor student and 5 interns.

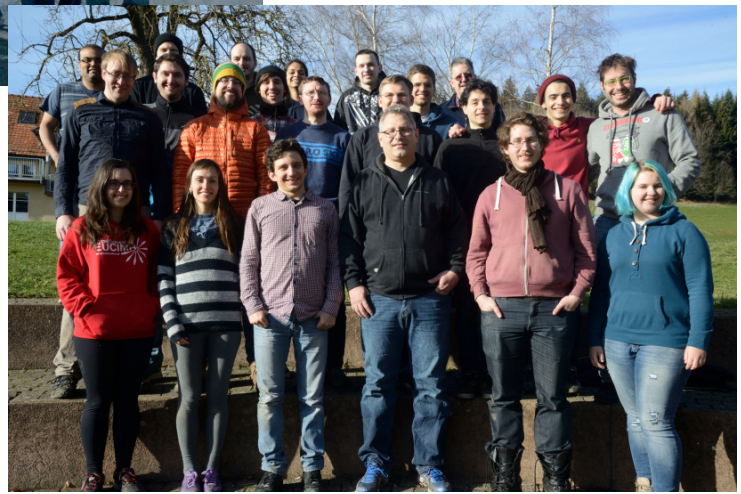
Find more info: **blog.zhaw.ch/icclab/**

We transformed from

ICC LAB

to ...

Service Engineering Research Area
ICCLAB SPLAB



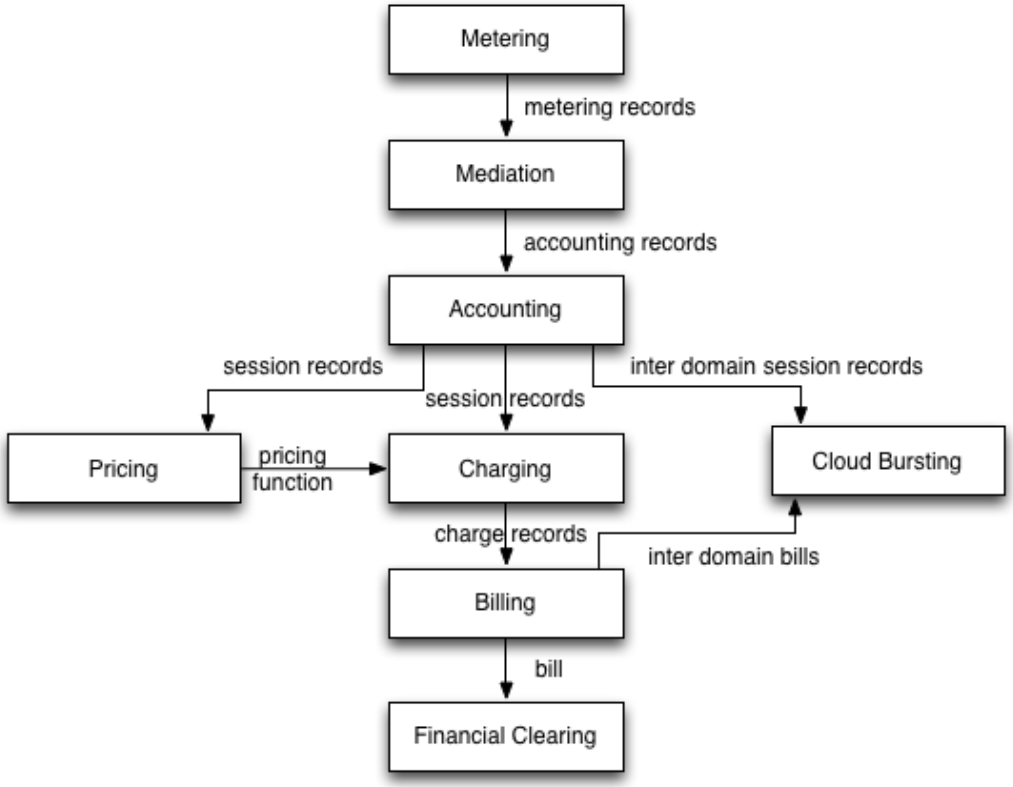
CYCLOPS

A quick primer ...

How does a typical accounting process look like?

Key phases -

- ➔Metering
- ➔Mediation
- ➔Accounting
- ➔Pricing
- ➔Charging
- ➔Billing
- ➔Financial Clearing



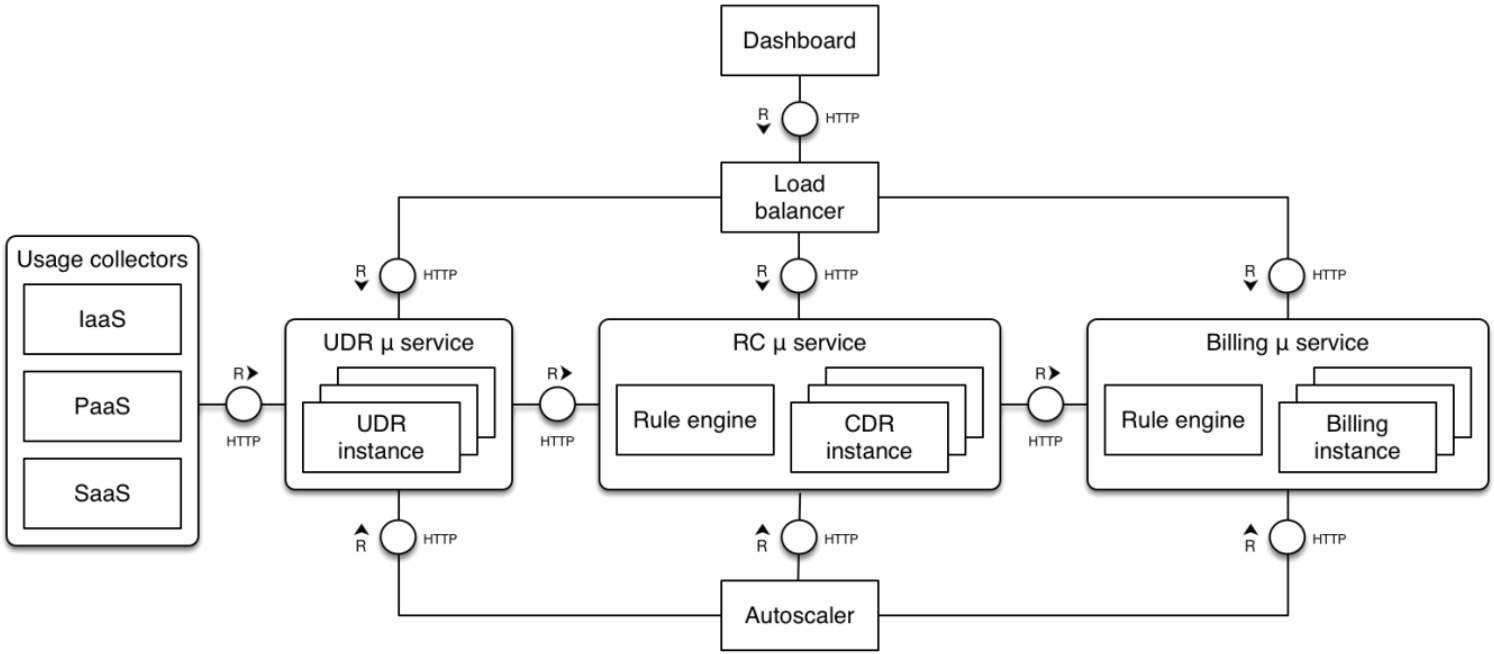
CYCLOPS

Cyclops: present release

What is Cyclops?

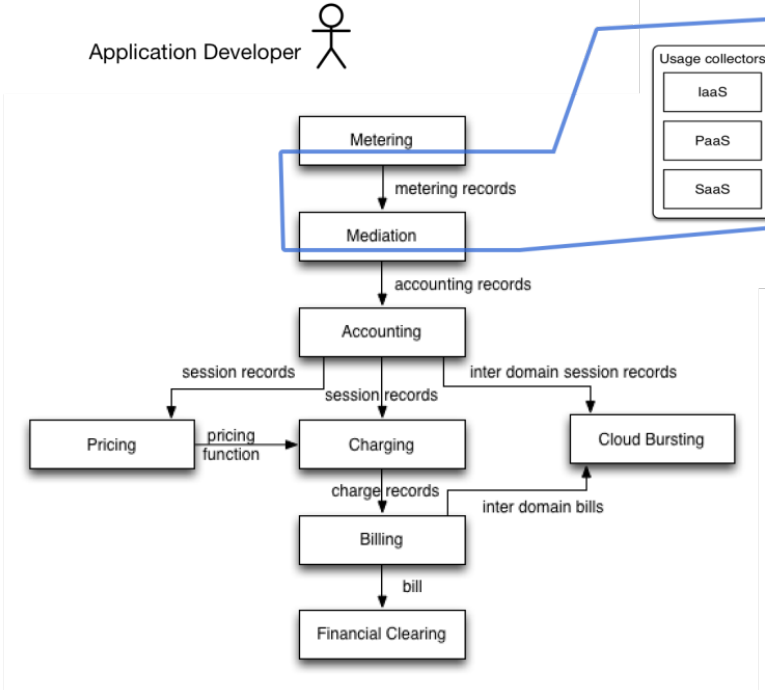
An open-source (ASL2.0), actively developed and maintained (by ICCLab), framework consisting of well defined micro-services that enables agile, model-based accounting, billing for cloud and cloud based services.

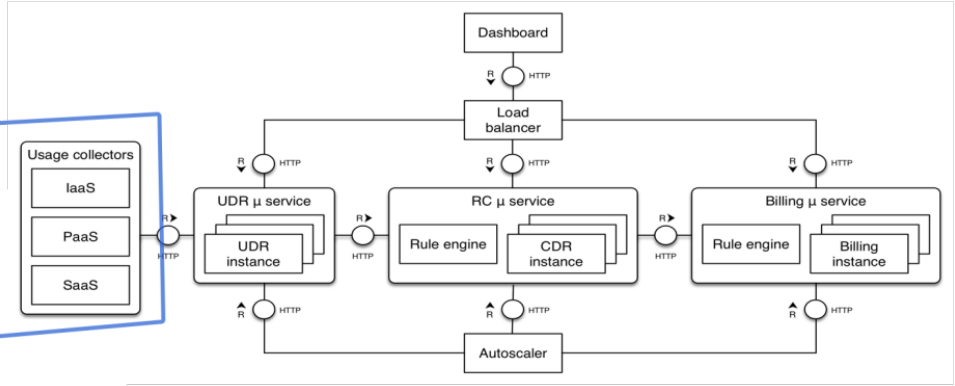
Architecture



CYCLOPS

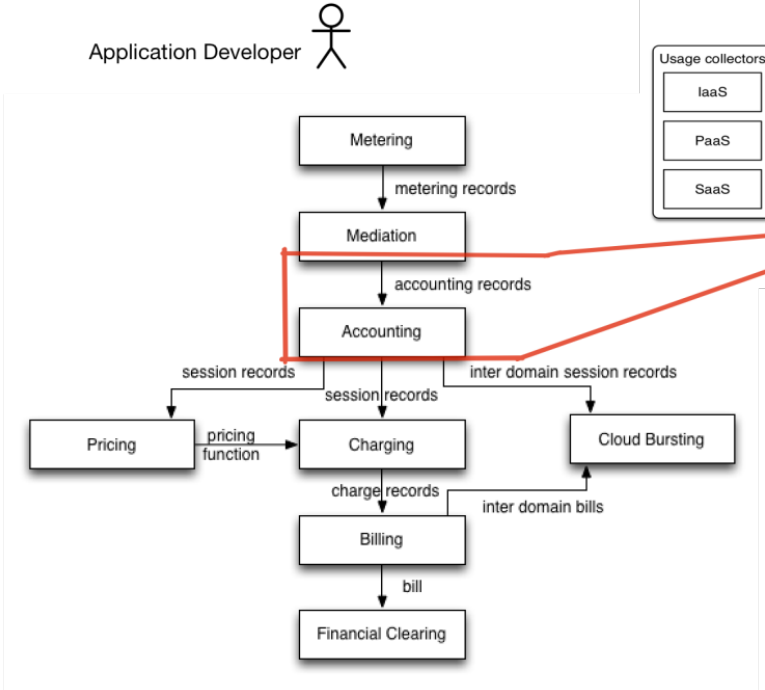
Mapping Cyclops to generic accounting process

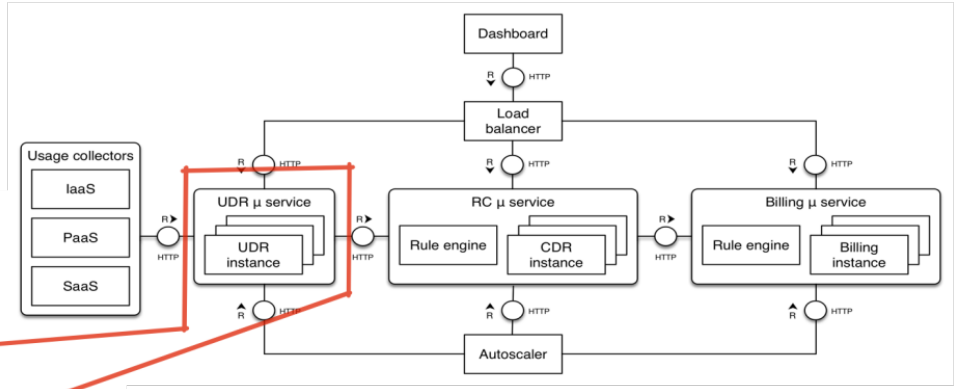




CYCLOPS

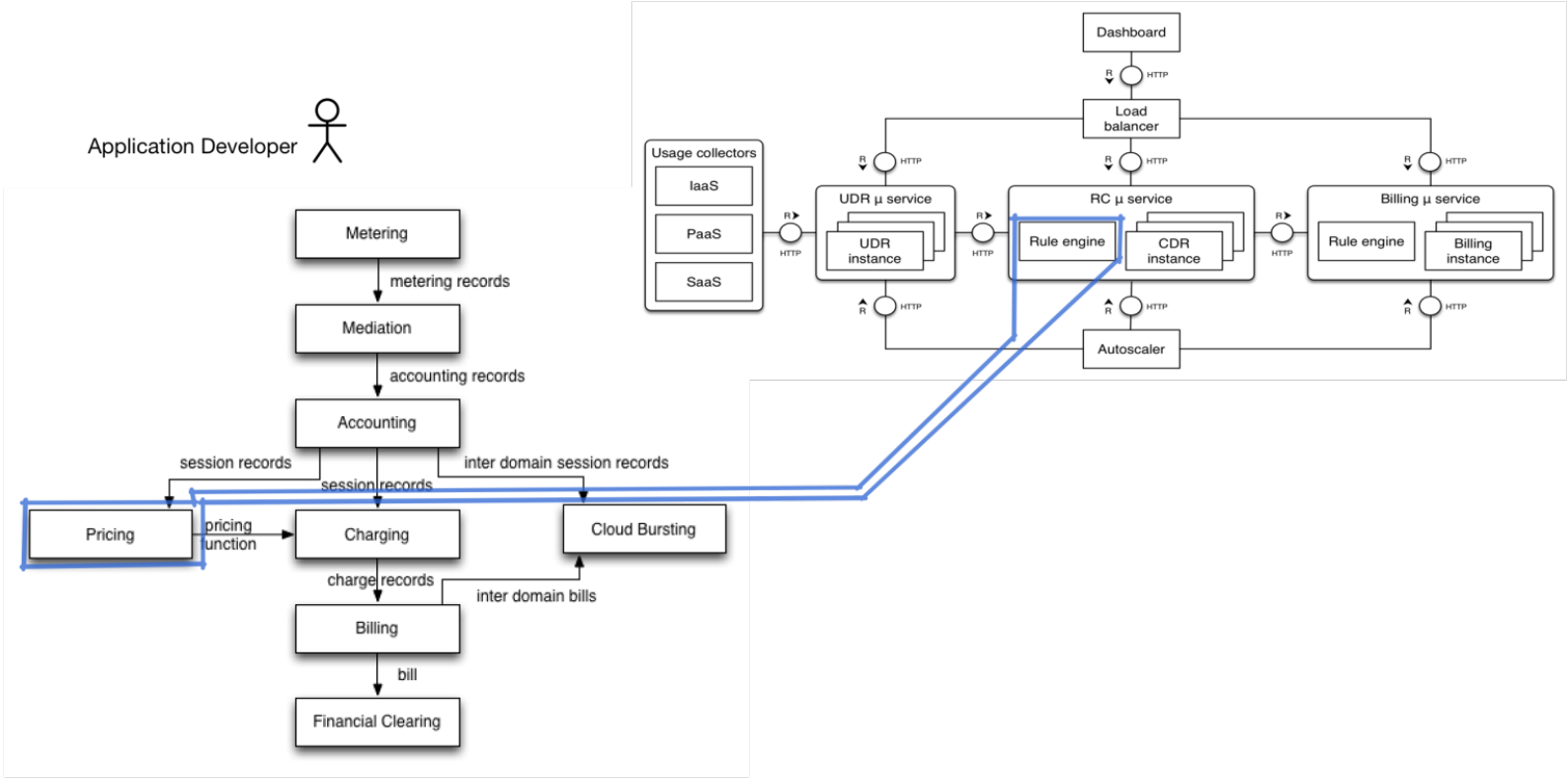
Mapping Cyclops to generic accounting process





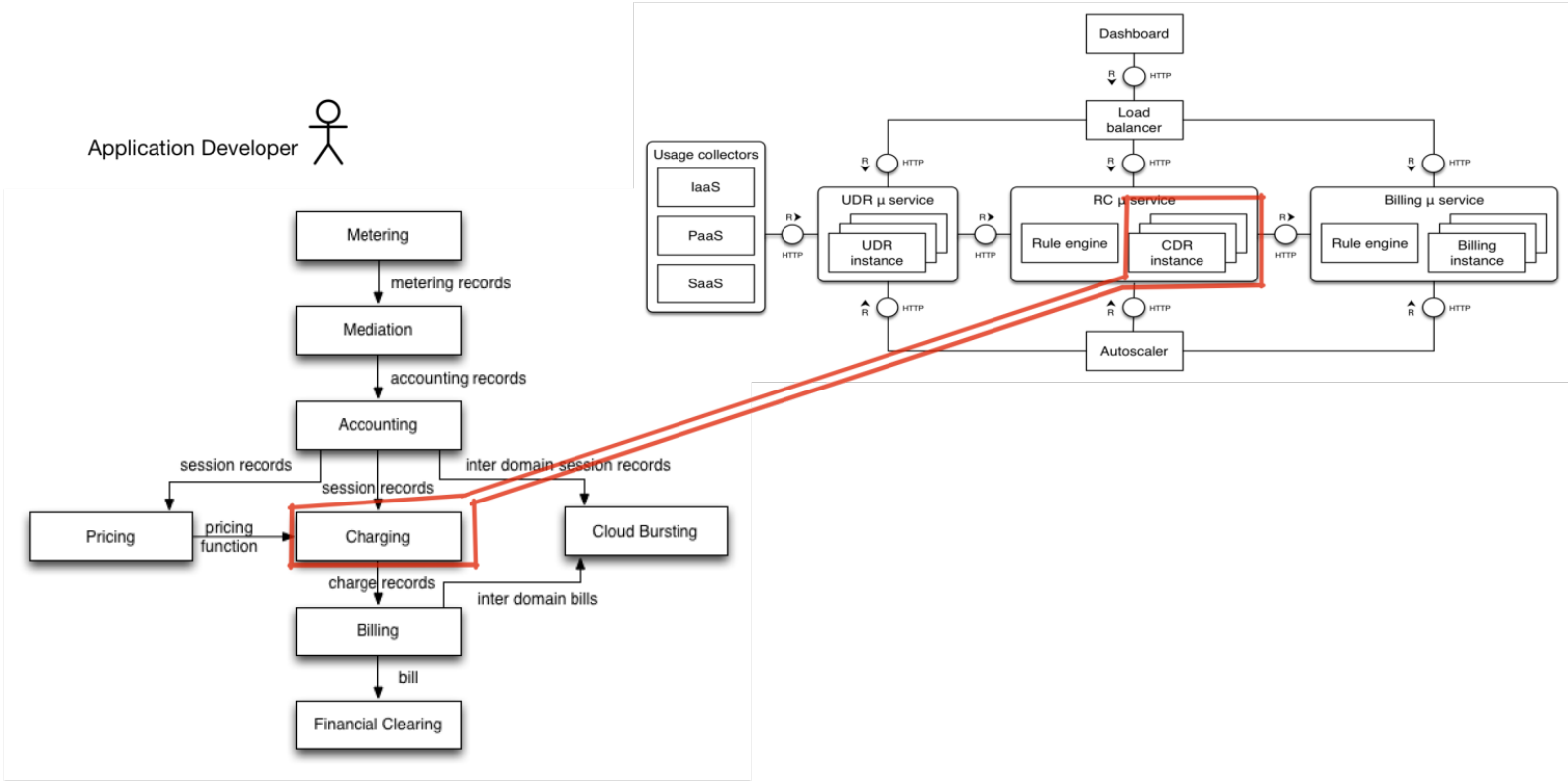
CYCLOPS

Mapping Cyclops to generic accounting process



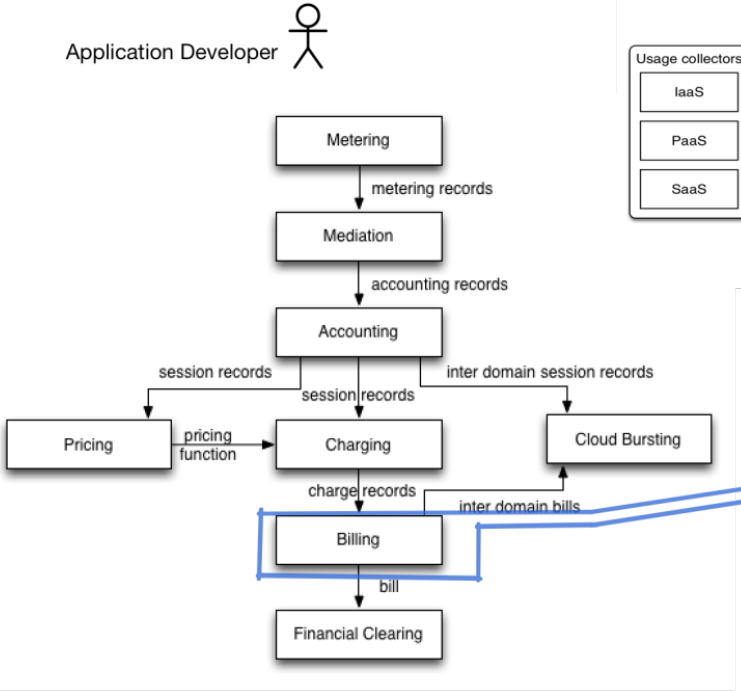
CYCLOPS

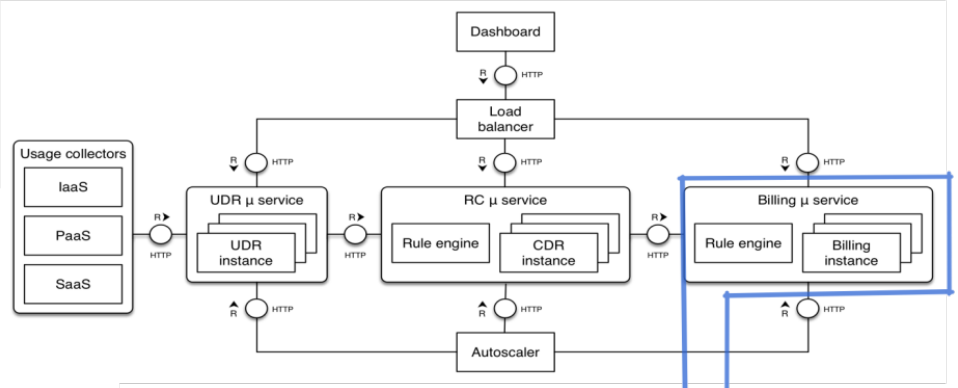
Mapping Cyclops to generic accounting process



CYCLOPS

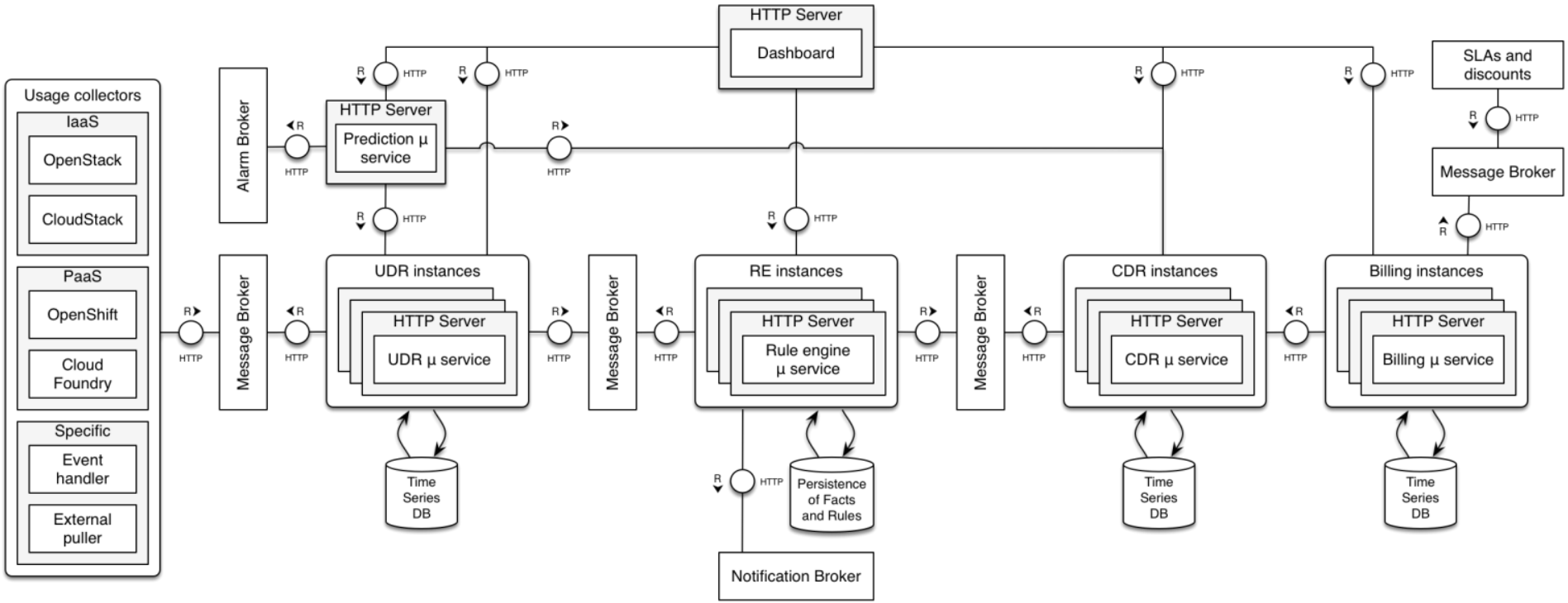
Mapping Cyclops to generic accounting process





CYCLOPS

Architecture (expanded) & workflows



CYCLOPS

Cyclops: micro-services

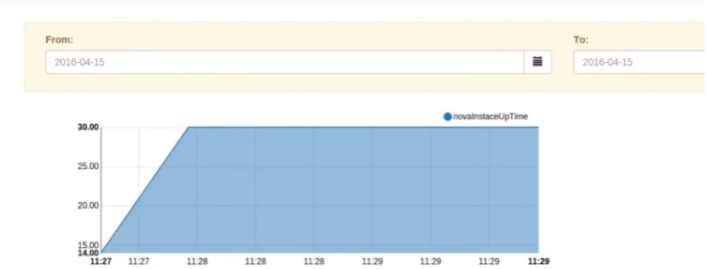
Microservices: Usage Data Records (UDR)

➔Harmonizes the resource consumption data coming from the collectors

➔Merges multiple data stream into a single output stream - udr-reports

➔Data APIs

- Usage data visualization in the dashboard
- Data prediction based on historic trends



CYCLOPS

Data transformation in UDR


```
{
  "metadata": {
    "project_id": "c49f7e6c177d44afb38fdaec62be8bc9",
    "resource_id": "935bec67-9f26-4be7-9a01-33c20e66bf00",
    "instance_name": "cyclops-demo"
  },
  "unit": "ns",
  "usage": 38190000000,
  "time": 1474488799,
  "_class": "OpenStackCeilometerCpu",
  "account": "8a3f30bb84a0495987393740264b8064"
}
```

```
{
  "metadata": {
    "project_id": "c49f7e6c177d44afb38fdaec62be8bc9",
    "resource_id": "935bec67-9f26-4be7-9a01-33c20e66bf00",
    "instance_name": "cyclops-demo"
  },
  "unit": "ns",
  "usage": 84980000000,
  "time": 1474537399,
  "_class": "OpenStackCeilometerCpu",
  "account": "8a3f30bb84a0495987393740264b8064"
}
```


Transforms into


```
{
  "measurement": "UDR",
  "data": [
    {
      "data": [
        {
          "metadata": {
            "instance_name": "cyclops-demo",
            "project_id": "c49f7e6c177d44afb38fdaec62be8bc9",
            "resource_id": "935bec67-9f26-4be7-9a01-33c20e66bf00"
          },
          "_class": "OpenStackCeilometerCpu",
          "usage": 123170000000,
          "account": "8a3f30bb84a0495987393740264b8064",
          "unit": "ns"
        }
      ],
      "time": 1474537464,
      "account": "8a3f30bb84a0495987393740264b8064",
      "from": 1474488799,
      "_class": "UDR",
      "to": 1474535599
    }
  ],
  "displayedRecords": 1,
  "totalRecords": 1,
  "pageNumber": 0,
  "pageSize": 500
}
```


CYCLOPS

Microservices: Rating & Charging (RC)

Made of rule-engine + data persistence and query interface

➔Converts UDR into charge records

- Pricing derived from business model encoded as rules
- Rule Engine maintains real time facts in memory (or in db as per configuration)
- Easy to configure tiered rates, and customized pricing for “special” customers

➔Rating and charging is truly generic

- You configure the rules, as per your business model
- Completely RESTful interface
- Highly programmable

➔CDRs are stored back for visualization (time series)

CYCLOPS

Data transformation in CDR


```
{
 "measurement": "UDR",
 "data": [
  {
   "data": [
    {
     "metadata": {
      "instance_name": "cyclops-demo",
      "project_id": "c49f7e6c177d44afb38fdaec62be8bc9",
      "resource_id": "935bec67-9f26-4be7-9a01-33c20e66bf00"
     },
     "_class": "OpenStackCeilometerCpu",
     "usage": 123170000000,
     "account": "8a3f30bb84a0495987393740264b8064",
     "unit": "ns"
    }
   ],
   "time": 1474537464,
   "account": "8a3f30bb84a0495987393740264b8064",
   "from": 1474488799,
   "_class": "UDR",
   "to": 1474535599
  }
 ],
 "displayedRecords": 1,
 "totalRecords": 1,
 "pageNumber": 0,
 "pageSize": 500
}
```



```
{
     “data": [
     {
      "metadata": {
       "instance_name": "cyclops-demo",
       "project_id": "c49f7e6c177d44afb38fdaec62be8bc9",
       "resource_id": "935bec67-9f26-4be7-9a01-33c20e66bf00"
      },
      "_class": "OpenStackCeilometerCpu",
      "usage": 123170000000,
      "account": "8a3f30bb84a0495987393740264b8064",
      "charge": 12
     }
    ],
    "time": 1474537469,
    "account": "018d1f191f6e4e27b726c84a51e0d248",
    "from": 1474488799,
    "_class": "CDR",
    "charge": 12,
    "stateful": false,
    "to": 1474535599
  }
```


CYCLOPS

Microservices: Billing

Made of rule-engine + data persistence and query interfaces

- ➔Handles customer retention policies (discounts, rebates, promotions etc.)
- ➔SLA violations and penalties
- ➔Geographical rules (VATs, sales-tax, currencies, etc.)
- ➔Federated billing (experimental)
  - Combines CDR records belonging to the list of users into same consolidated bill
  - Example: Input - List[{user-id$_{provider\text{-}A}$}, {user-id$_{provider\text{-}B}$}, ...], Output: Federated Bill Record

CYCLOPS

Microservices: Rule Engine

Allows template instantiation and truth maintenance over RESTful APIs & allows business developers to write their own business rules for data driven rule execution (advances cyclops generality).

The most prominent features are:

- ➔Supports rules with priorities
- ➔Stateless and stateful sessions
  - ◆ Live counters allow tiered rates, volume discounts, etc.
- ➔Ability to unpack incoming data stream and repack transformed data into output stream

CYCLOPS

Example rule insertion

```
curl -X "POST" "http://localhost:$1/ruleengine/rule" -H "Content-Type: text/plain" -d $'import
ch.icclab.cyclops.facts.model.rcb.usagedata.openstack.ceilometer.OpenStackCeilometerCpuUti
l;
import ch.icclab.cyclops.facts.model.rcb.chargedata.ChargeData;

rule "Rate Ceilometer CPU Util"
salience 10
when
   $usage: OpenStackCeilometerCpuUtil()
then
   ChargeData charge = new ChargeData($usage);
   charge.setCharge(20);
   insert(charge);
   retract($usage);
end'
```

- ➔ Give the rule a name!
- ➔ Set the weight, if multiple rules' entry conditions are satisfied then the one with higher "weight" is executed
- ➔ The micro-service unpacks the UDR, applies rules to each element and repacks and stores

CYCLOPS

Microservices: Gatekeeper

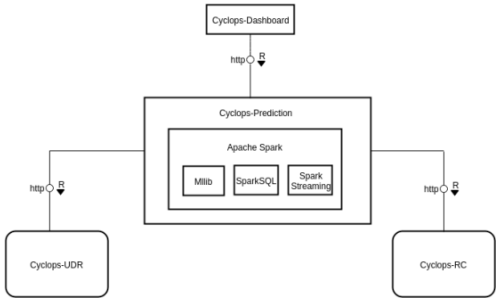
http://icclab.github.io/gatekeeper/

- RBAC microservice written in golang!
  - Supports 1000s of API calls / minute
  - User+process authentication and authorization
  - Token generation and management (validation, revocation, etc.)
  - Support for arbitrary set of roles, and fine-grained authorization granularity (per API endpoint, service, object, etc.)
  - Inter-services authorization support
  - RESTful APIs for ease of integration and programmability

Gatekeeper also features in these projects

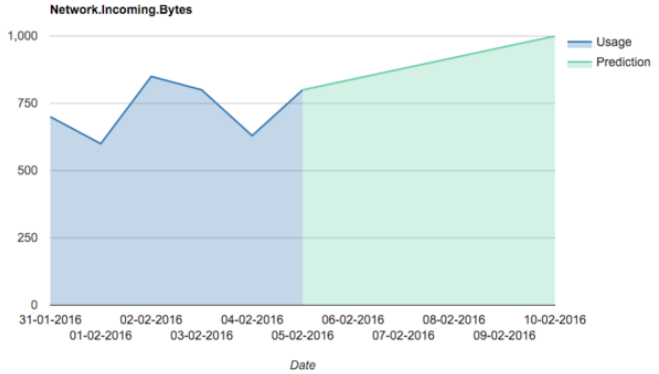
Microservices: Prediction Engine

Designed to predict values based on historical data but also capable of generic factual forecasting by encapsulating mllib and openforecast into a singular prediction engine.



- ➔ Mllib - linear regression
- ➔ OpenForecast - enables selection of best forecasting algorithm based on input data samples
  - ◆ Linear regression, exponential smoothing, naive bayes, etc.
- ➔ Provides usage forecasting to service end users

Planned: Resource planning tool for administrators, providers!



CYCLOPS

https://blog.zhaw.ch/icclab/prediction-engine-prototype-in-cyclops/

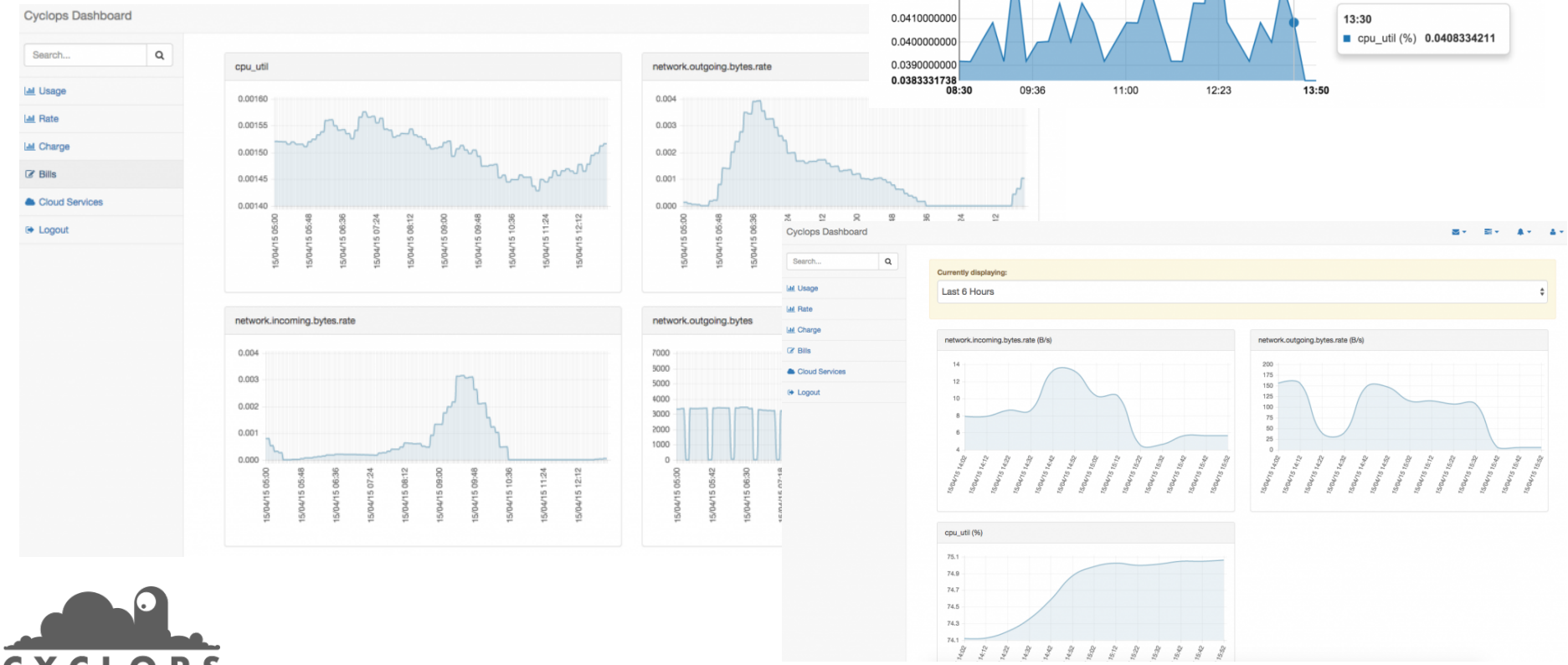
UI Tools: Dashboard

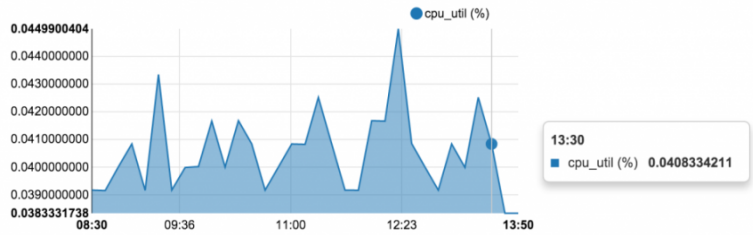
A fully function component allows all options of a billing process to be managed

- ➔Differentiated views: admin view + user view
- ➔Authenticated access (currently uses Keystone / Gatekeeper)
- ➔For end users
  - Allows access to usage and charge graphs
  - Usage and cost prediction
  - Current and past bills
- ➔For cloud providers (under development in dashboard 2.0 branch)
  - Selection and management of meters, both built-in ceilometer meters or external meters
  - Pricing control
  - Bill generation and release control (making a bill visible to the customer after validation)

CYCLOPS

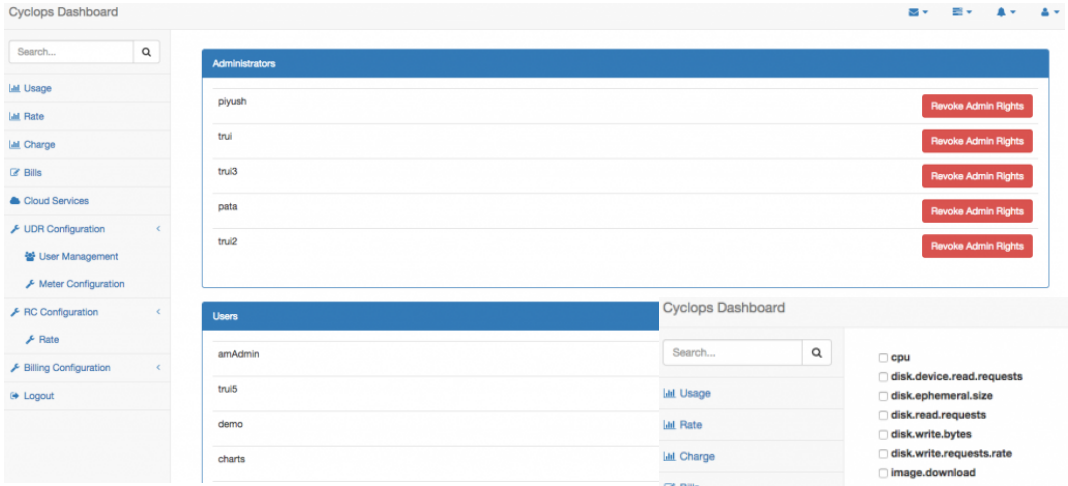
Cyclops Dashboard (then)

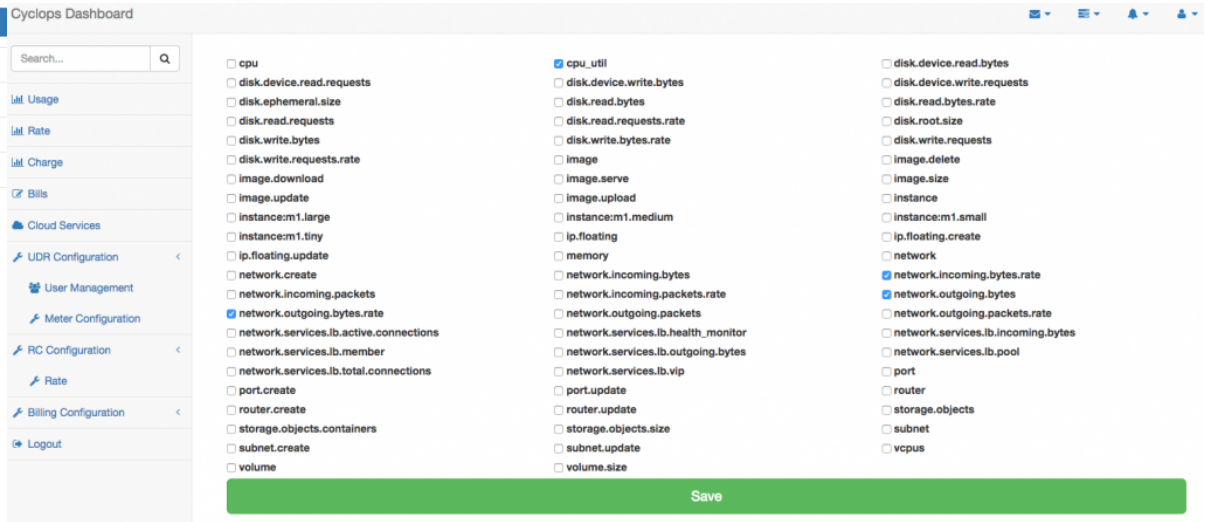




CYCLOPS

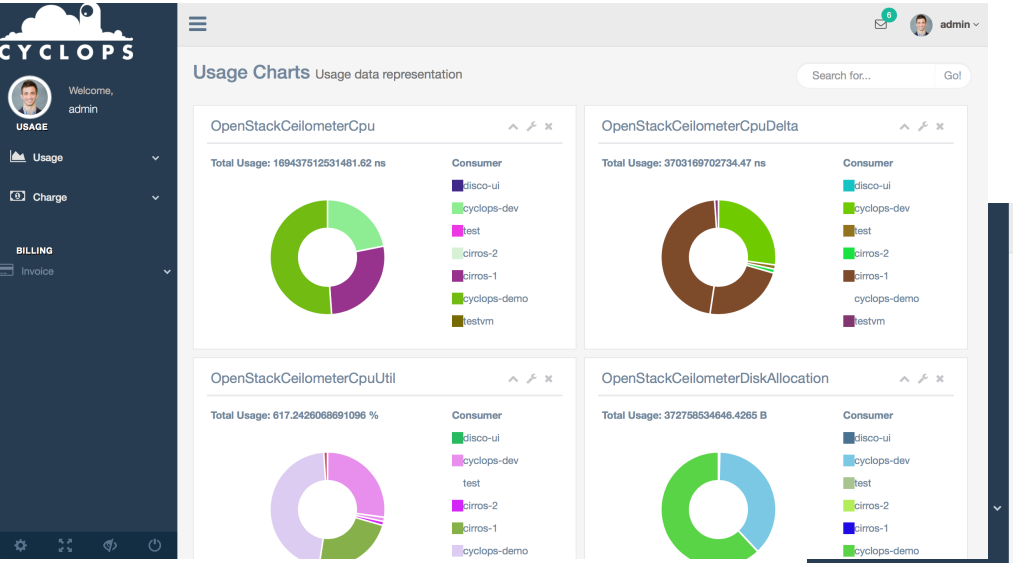
Cyclops Dashboard (then)

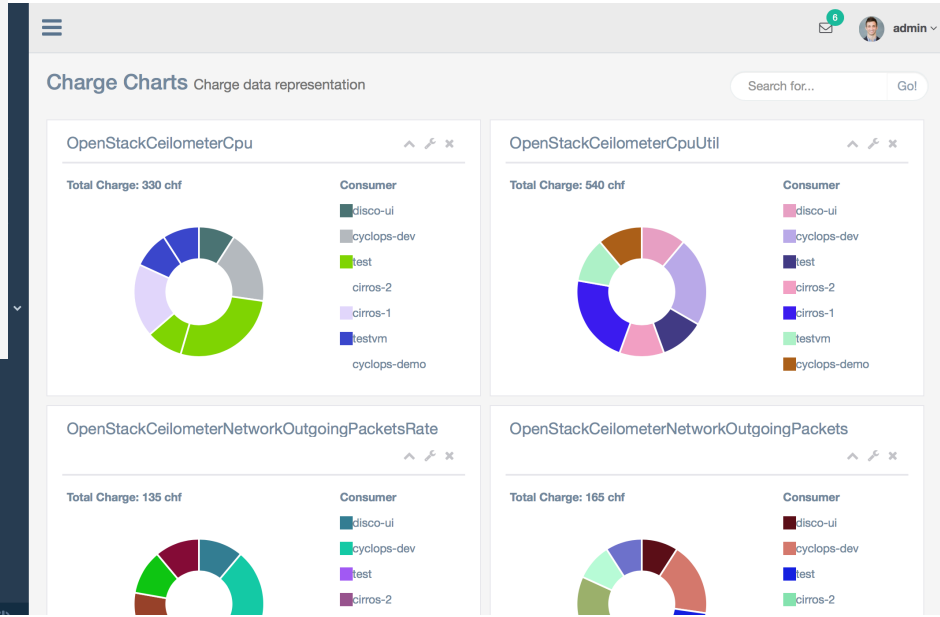




CYCLOPS

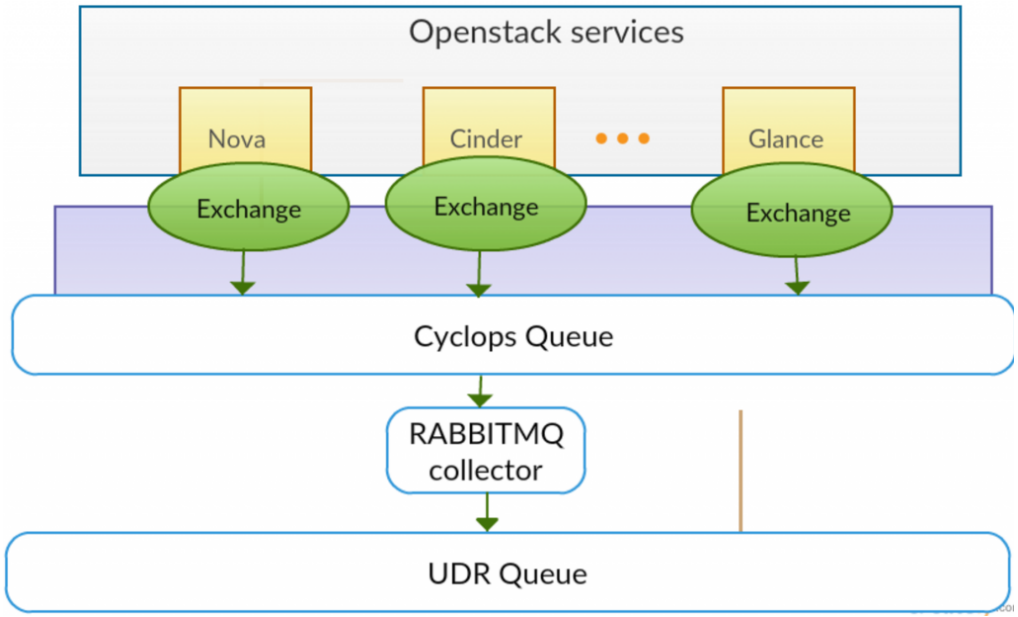
Cyclops Dashboard (now, under development)





CYCLOPS

Cyclops collector: OpenStack events



➔ Billing now possible based on key Nova events
- ◆ VM start
- ◆ VM terminated
- ◆ VM suspended
- ◆ VM resumed

➔ Key VM metadata is now processed
- ◆ VM flavor information

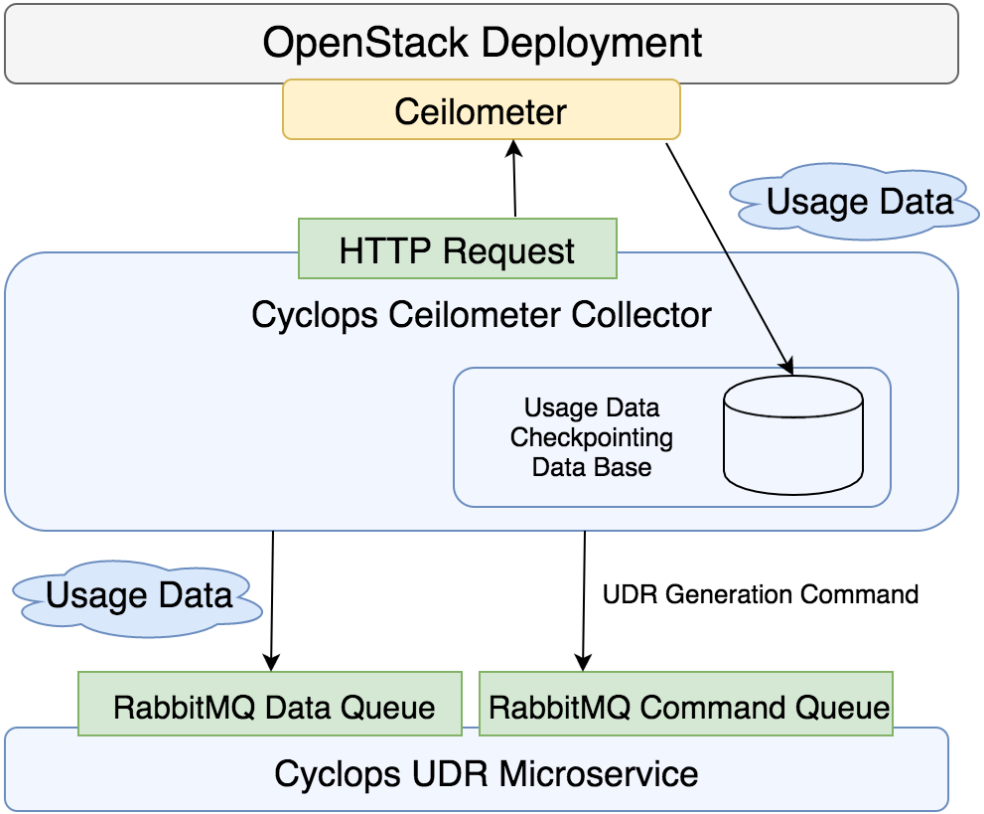
It is now possible define pricing rules like
- ❑ x.yy CHF / hr for m1.large VM
- ★ Extension of similar capabilities for events from Glance, Neutron are under development

CYCLOPS

https://blog.zhaw.ch/icclab/openstack-nova-events-are-now-billable-with-cyclops/

Cyclops collector: OpenStack Ceilometer

- ➔ Harmonizes all data sources
  - ◆ Proper handling of -
    - Cumulative
    - Delta
    - Gauge meters
- ➔ Computation of actual usage done within the collector itself
  - ◆ UDR micro-service treats all data sources as usage streams



CYCLOPS

List of supported platforms

- OpenStack
  - Ceilometer Collector
  - Events collector (taps into OpenStack messaging bus)
  - Tests with Mitaka release underway
- CloudStack
  - Usage collector supports all built-in meters

openstack™

cloudstack
open source cloud computing

PS: External Meters

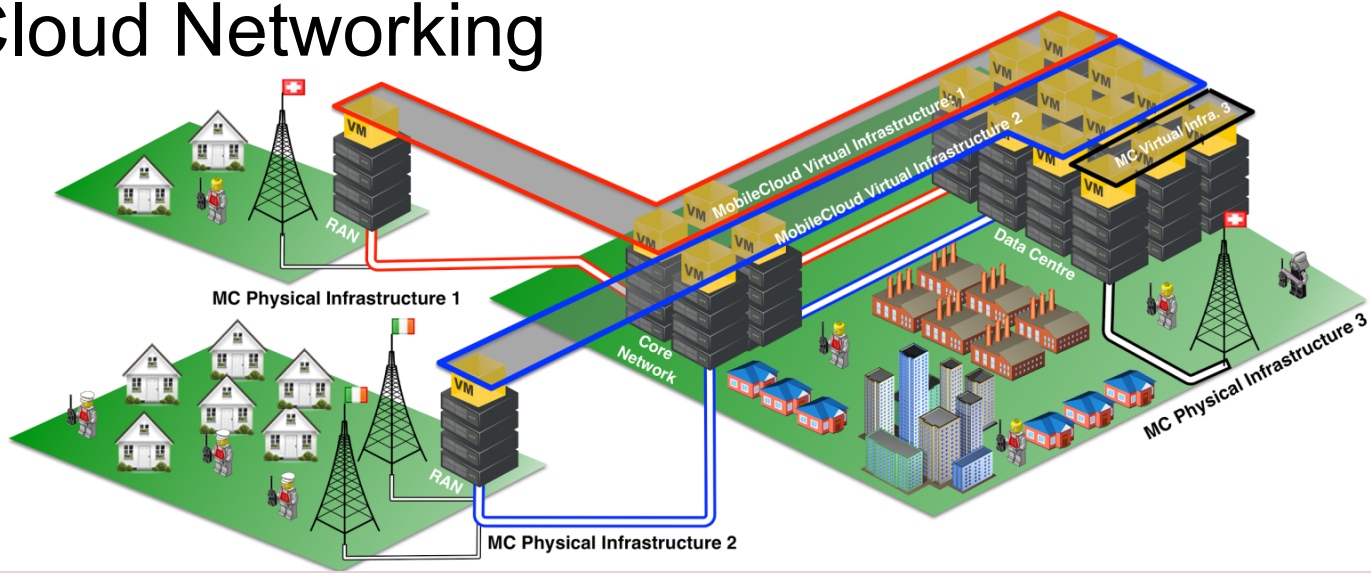
➔Any non-natively supported framework can be billed with Cyclops

- Use external meters: define a unique meter-name, implement the collector for the external framework, and modify the rules in the rule-engine. THAT'S IT!

CYCLOPS

Cyclops Demonstrable Use Cases

Mobile Cloud Networking



Rating-charging-billing for composed telco-functions deployed over OS

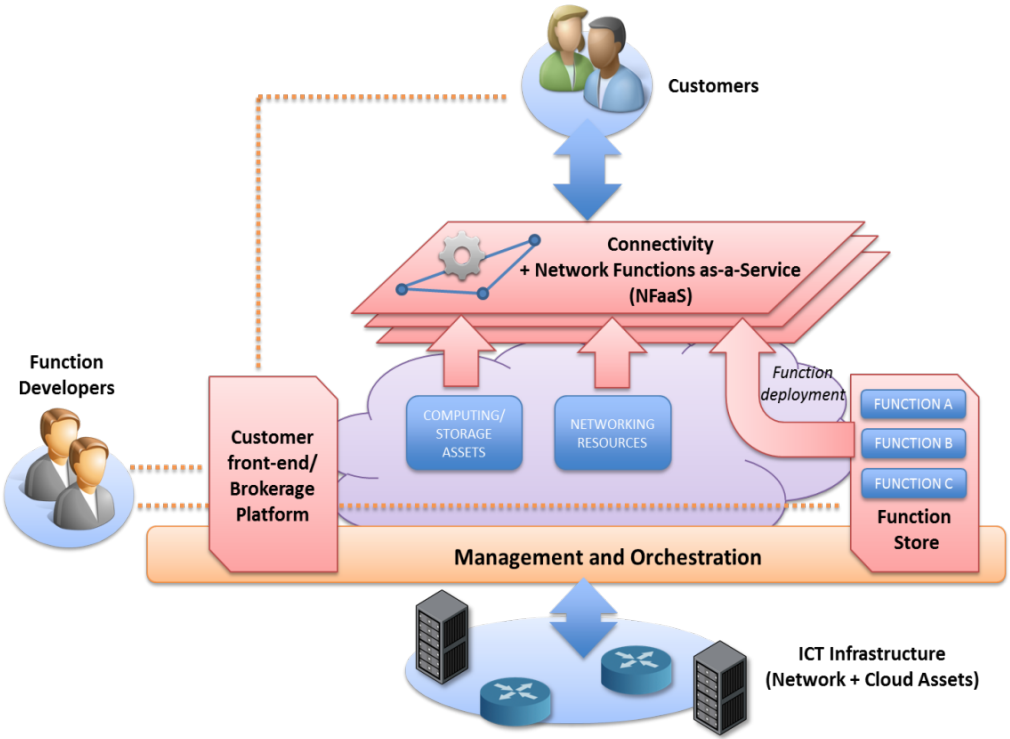
- ➔ {RAN + EPC + IMS + Support Services (mobility predictor, CDN, db, DNS, etc.)}aaS
- ➔ Rating, charging based on events (start, stop, resume, suspend) + OpenStack usage
  - ◆ Implemented using cyclops via external meters + built in Ceilometer collector!
  - ◆ Converged billing (OpenStack + on-demand provisioned telco services)

CYCLOPS

http://www.mobile-cloud-networking.eu/

T-NOVA

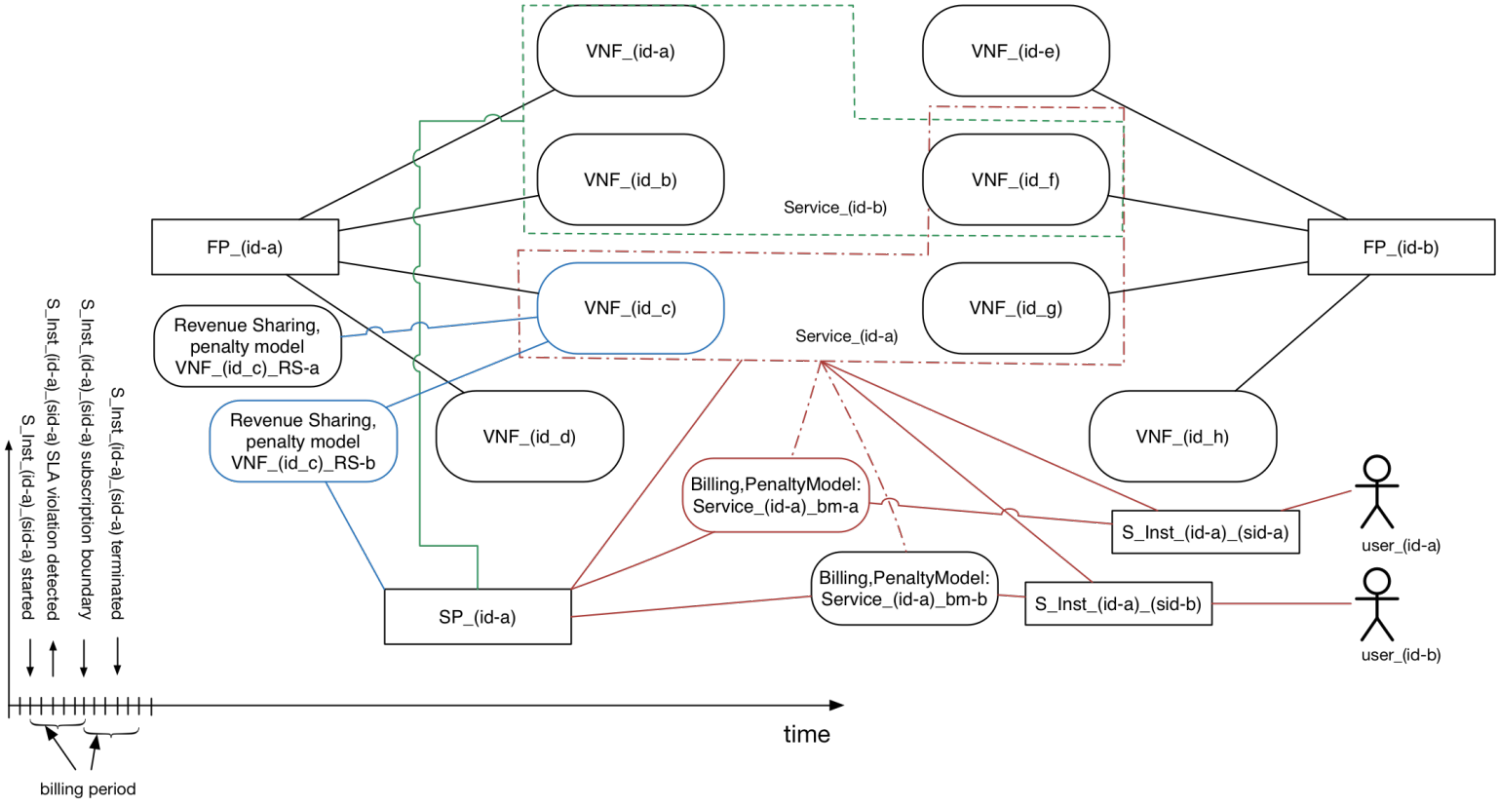
- Billing support for VNF marketplace
  - Multi-actor environment - service provider, enterprise customers, function developers
  - Negotiated revenue sharing models had to be supported
  - Penalty models based on SLA violations.
  - Billing models: subscription or PAYG
- Cyclops easily integrated with accounting & sla-m by ATOS
- Billing based on key lifecycle events



CYCLOPS

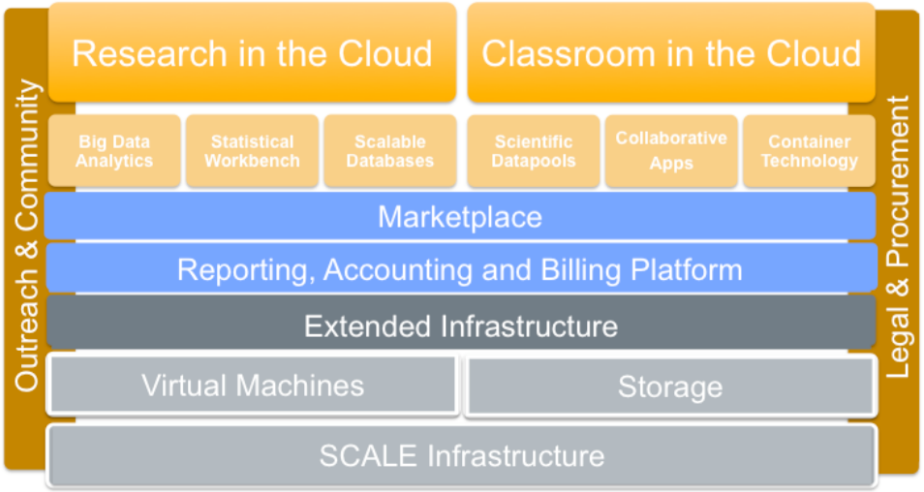
http://www.t-nova.eu/

T-NOVA - Billing Objects



CYCLOPS

SCALE-UP Project - Organizational Hierarchies



Rule engine easily encodes logic for organization hierarchies and users costs are automatically aggregated in the bill!

- Billing in academia
  - Organization level billing: university, research labs, academic units, users.
  - Cost trackable at sub-organization level
  - Tracking of resource usages per user
    - compute, storage, network, software licenses, support, etc.
- Supporting academic marketplace over OpenStack cloud
  - Revenue sharing, virtual credits, penalties upon sla violations, etc.
  - Developer friendly APIs
- Rating and charging of compute clusters - Hadoop, Spark, etc.

CYCLOPS

https://projects.switch.ch/scale-up/

SSC: Marketplace, product bundling

- ➔Ecosystem with ISVs, White label resellers and distributors
  - ◆ Revenue sharing models are to be supported
- ➔Marketplace
  - ◆ Individualized billing models for marketplace offered services and appliances
    - Custom meter creation, pricing rules, metrics collection and processing
  - ◆ Product bundles
    - New services and offerings to be created from existing appliances or products
    - Billing at the level of product bundles
- ➔Ability to track usage at elemental level of any given composition

Cyclops rule engine handles all these specific needs, and UDR reports are generated as long as collectors are written correctly.

CYCLOPS

Time for a quick demo

CYCLOPS

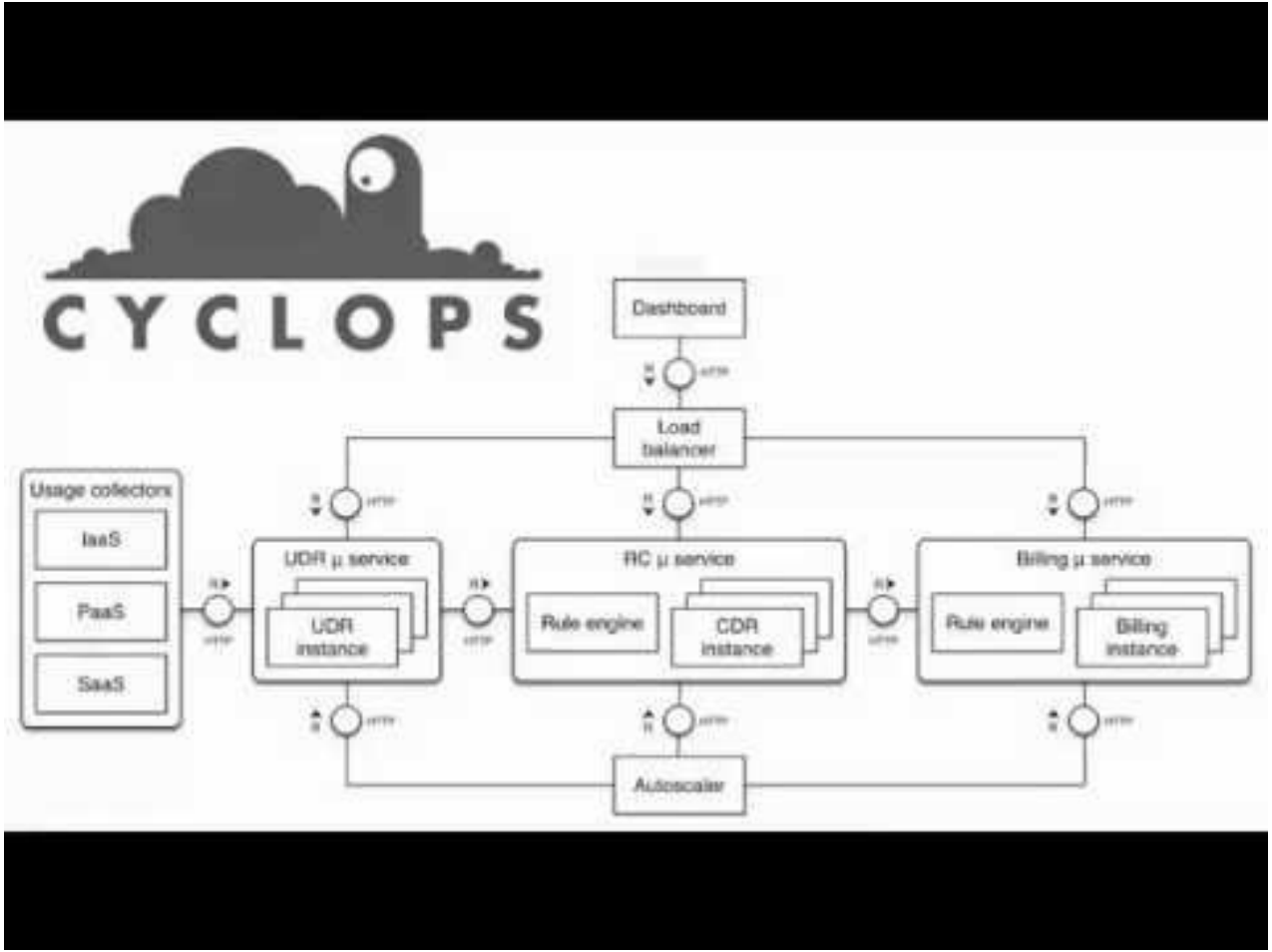
What you see in this video?

Rule management

- ➔ Starting of services
- ➔ Rule engine initialization
- ➔ Insertion of usage data
- ➔ Generation of consolidated bill

Dashboard

- ➔ Usage & Charge Views



CYCLOPS

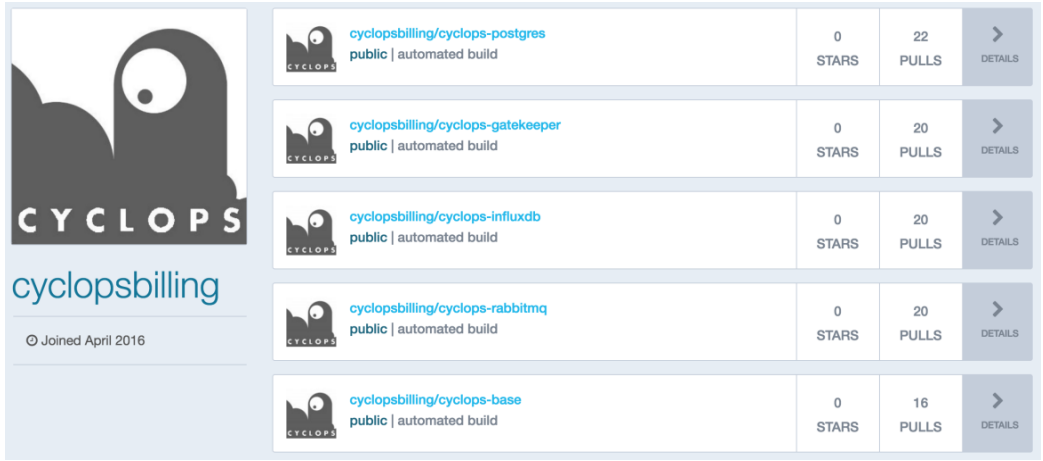
Cyclops: Dev & deployment support

Every microservice has been containerized.

➔Docker build images and docker-compose.yml files are already released as open source

➔Search for *cyclopsbilling* in hub.docker.com

- ➔ Extensive developer's guide to extend the framework has been released
- ➔ Automated build scripts for each microservice is available.



CYCLOPS

What next with Cyclops?

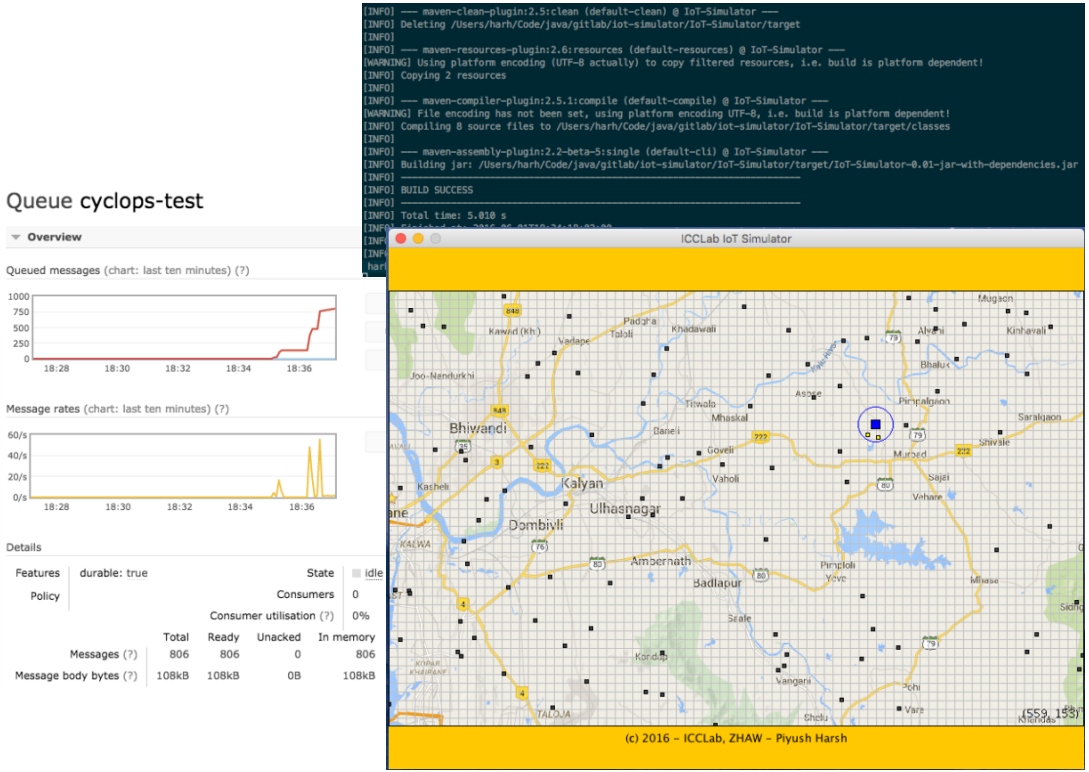
Billing for smart IoT deployments

➔Managing data-gaps in streams

- ◆ Devices may not be connected always for data offloading

➔Corrupt data identification, tagging & management

We wrote our own IoT deployment simulation tool!



http://icclab.github.io/iot-simulator/

CYCLOPS

Further information ...

Cyclops Page: http://icclab.github.io/cyclops/

Gatekeeper: http://icclab.github.io/gatekeeper/

ICCLab RCB initiative page: http://bit.ly/1WYC27l

@rcb_cyclops - follow us on twitter.

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
swissuniversities

TNOVA
NETWORK FUNCTIONS AS-A-SERVICE OVER VIRTUALISED INFRASTRUCTURES

SEVENTH FRAMEWORK PROGRAMME

MCN
Mobile Cloud Networking

CYCLOPS

Cyclops has been proudly funded by these organizations / projects