A Highly Available Generic Billing Architecture for Heterogeneous Mobile Cloud Services

Piyush Harsh
Zurich University of Applied Sciences

SEVENTH FRAMEWORK PROGRAMME

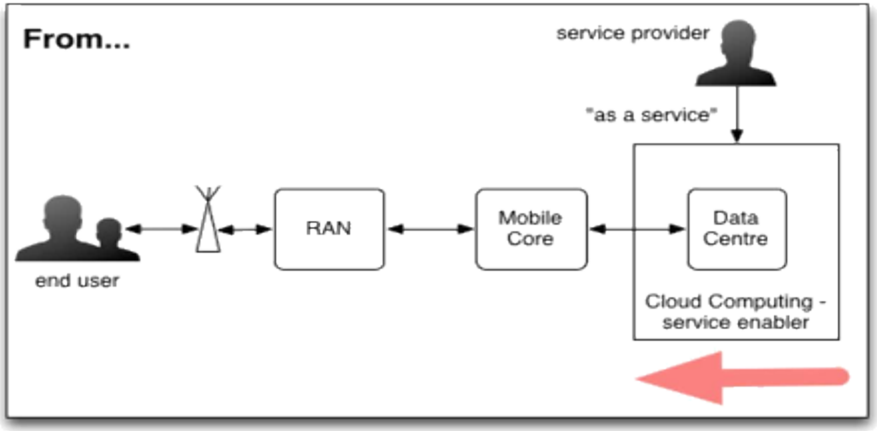
MCN
Mobile Cloud Networking

Talk Outline

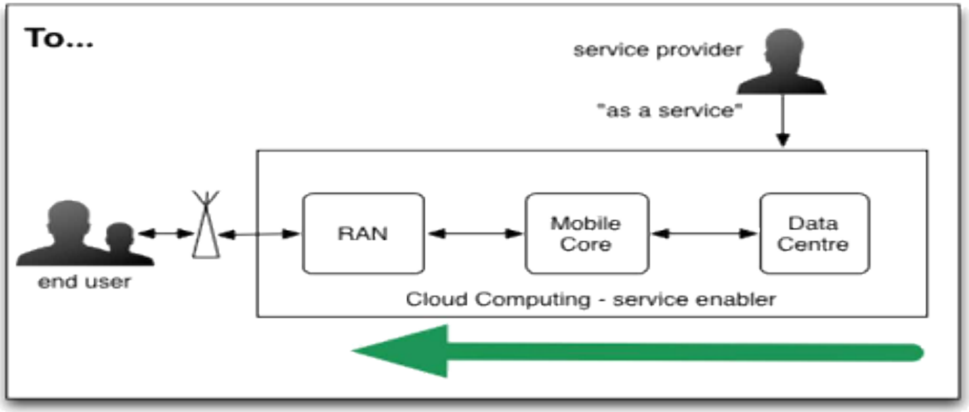
1. Introduction to Mobile Cloud Networking project
2. Accounting Process
3. Platform Architectures
4. Current Implementation Status
5. Future Roadmap & Conclusion

Mobile Cloud Networking

The motivation



Cloud computing principles beyond datacenters …



… towards the mobile end users.

Mobile Connectivity
Decentralized Computing
Smart Storage

Offered as a single end-to-end service

- On-demand and self service
- Elastic
- Multi-tenant
- Pay-as-you-go

Mobile Cloud Networking

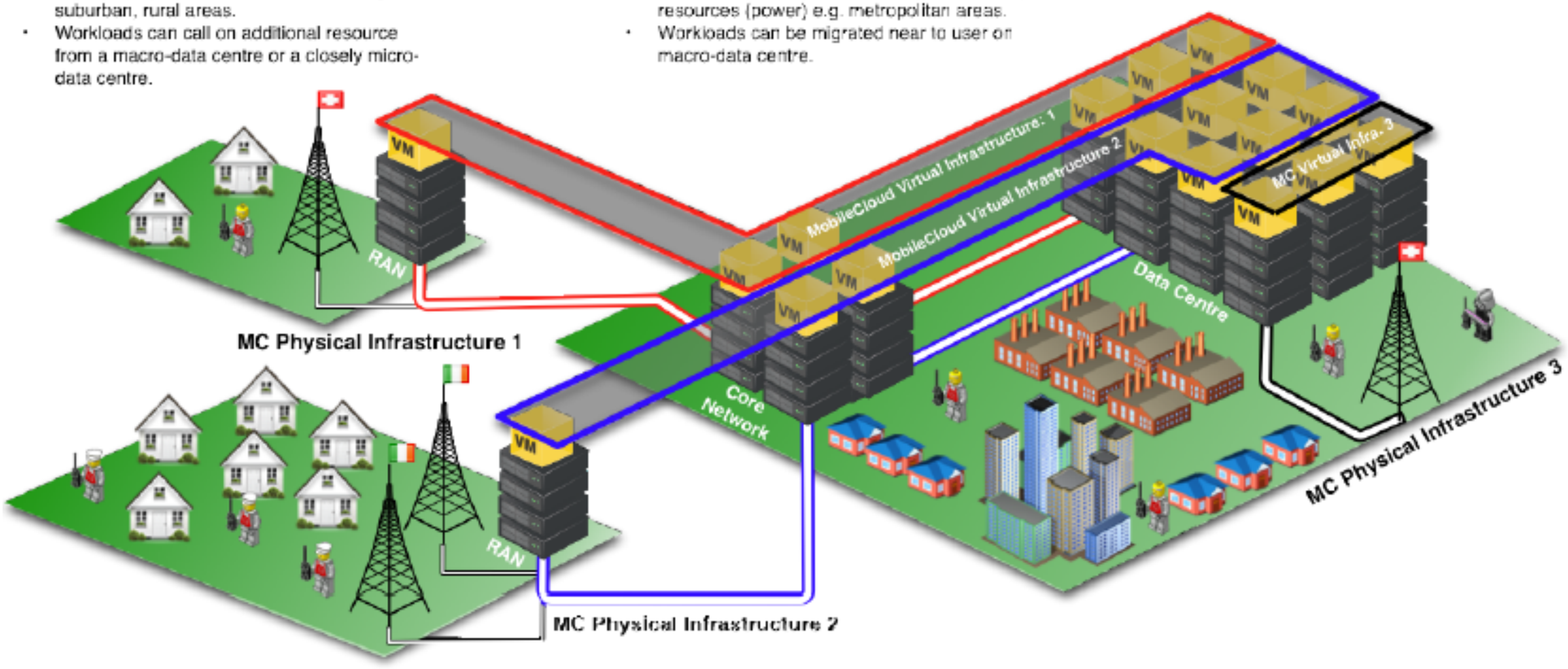
http://mobile-cloud-networking.eu/

Micro-Data Centre Deployments

- Local deployments limited resources e.g. suburban, rural areas.
- Workloads can call on additional resource from a macro-data centre or a closely micro-data centre.

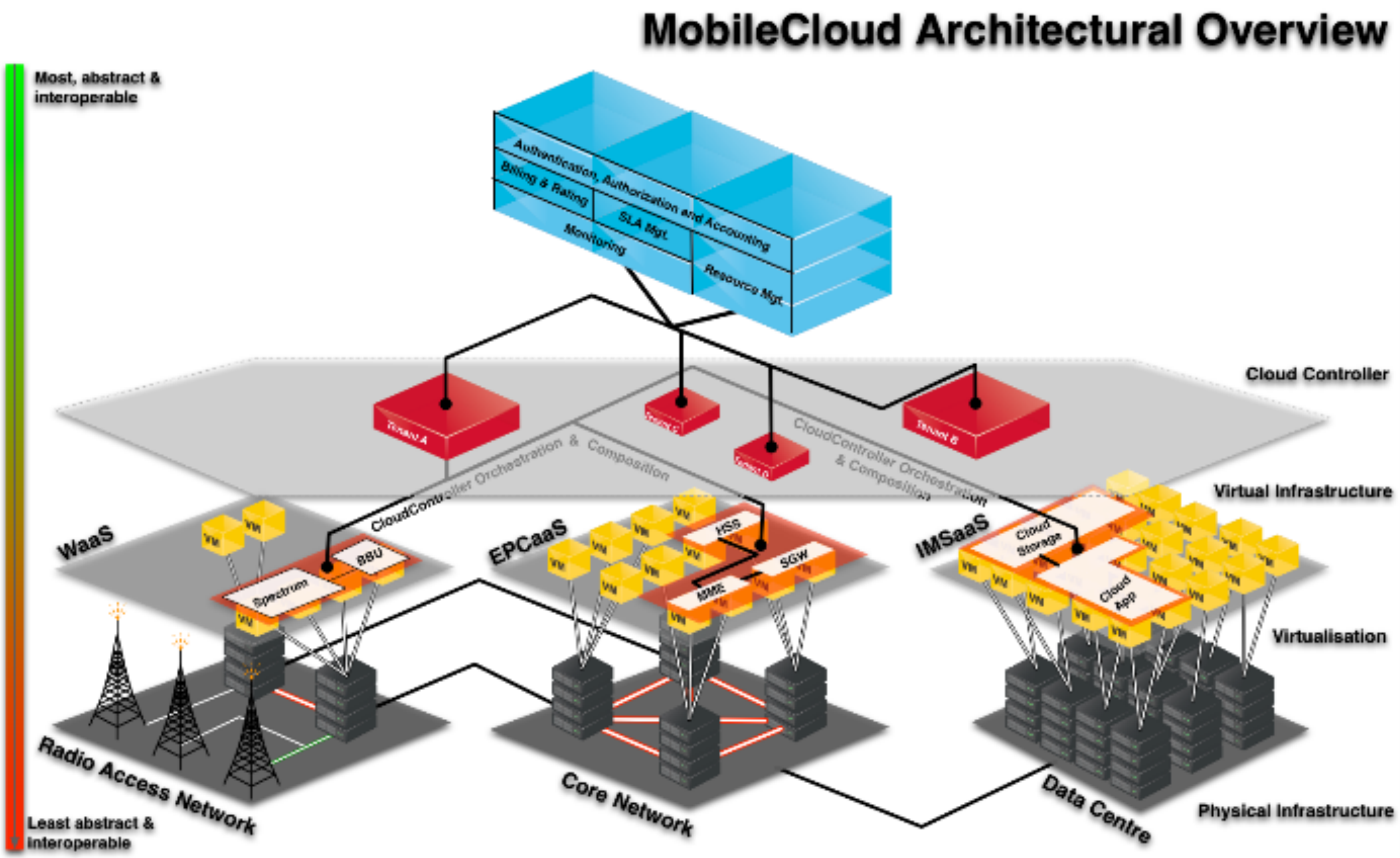
Macro-Data Centre Deployments

- Centralised deployment with access to cheap resources (power) e.g. metropolitan areas.
- Workloads can be migrated near to user on macro-data centre.


MobileCloud Virtual Infrastructure: 1
MobileCloud Virtual Infrastructure 2
MC Virtual Infra. 3
VM
RAN
MC Physical Infrastructure 1
Core Network
Data Centre
MC Physical Infrastructure 3
RAN
MC Physical Infrastructure 2

Mobile Cloud Networking

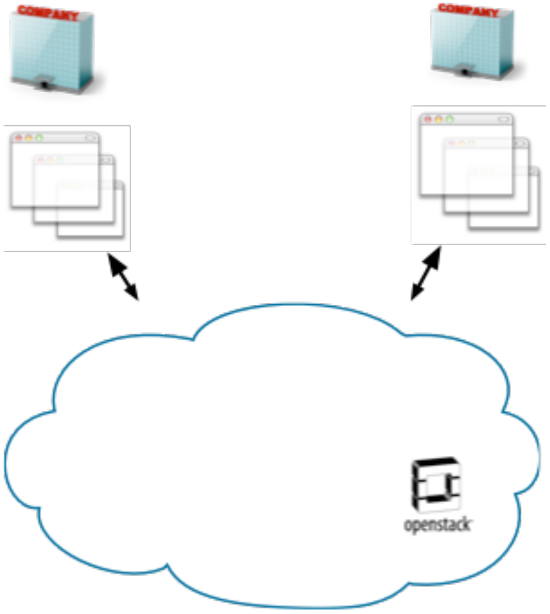
High Level Architecture



A few use-case scenarios

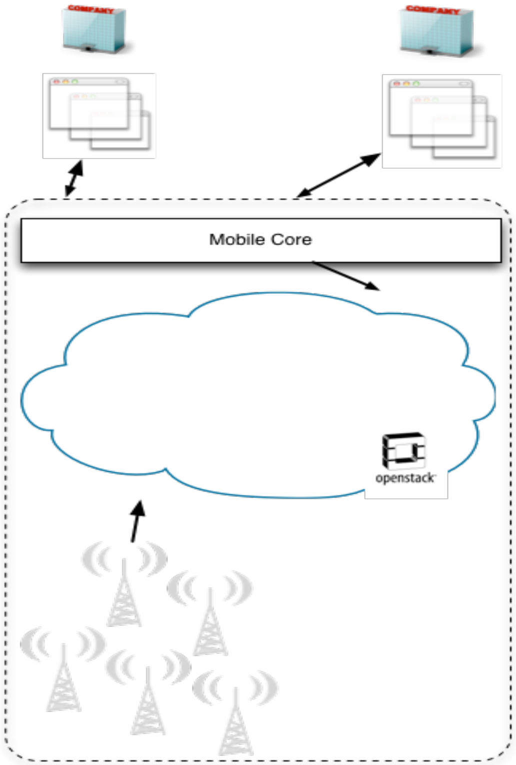
Motivating the need for a generic RCBaaS …

Pure IaaS operator scenario



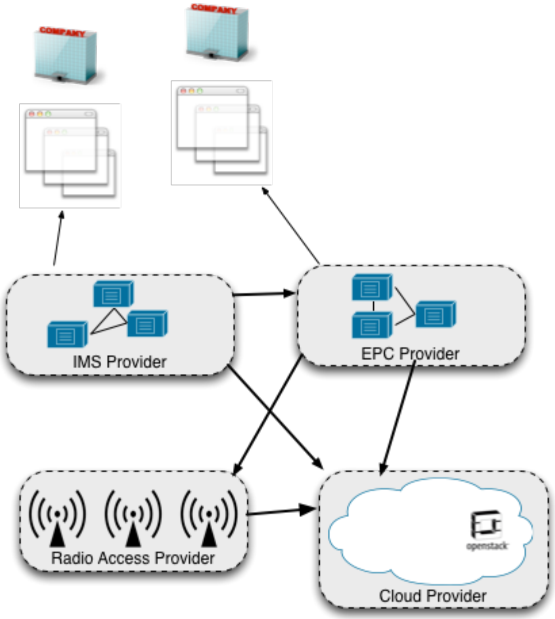
Rating-charging-billing solution needed from the cloud provider's perspective …

Cloud + Telco integrated scenario



Rating-charging-billing solution needed in a composite service offering

MCN ecosystem scenario



Rating-charging-billing solution needed in a multi-stake holder scenario …

How things are done typically …

- You have a service …
- You design a billing mechanism for this service
- The platform is highly specialized for your need
- In future, you expand your service offerings
- The old billing platform is now no longer relevant
- You redesign your billing strategy
- You make new tools and integrate with the service portfolios
- And the process continues …

But the unifying principle is …

You bill customers based on resources and services they have consumed.

❑ You measure what was consumed

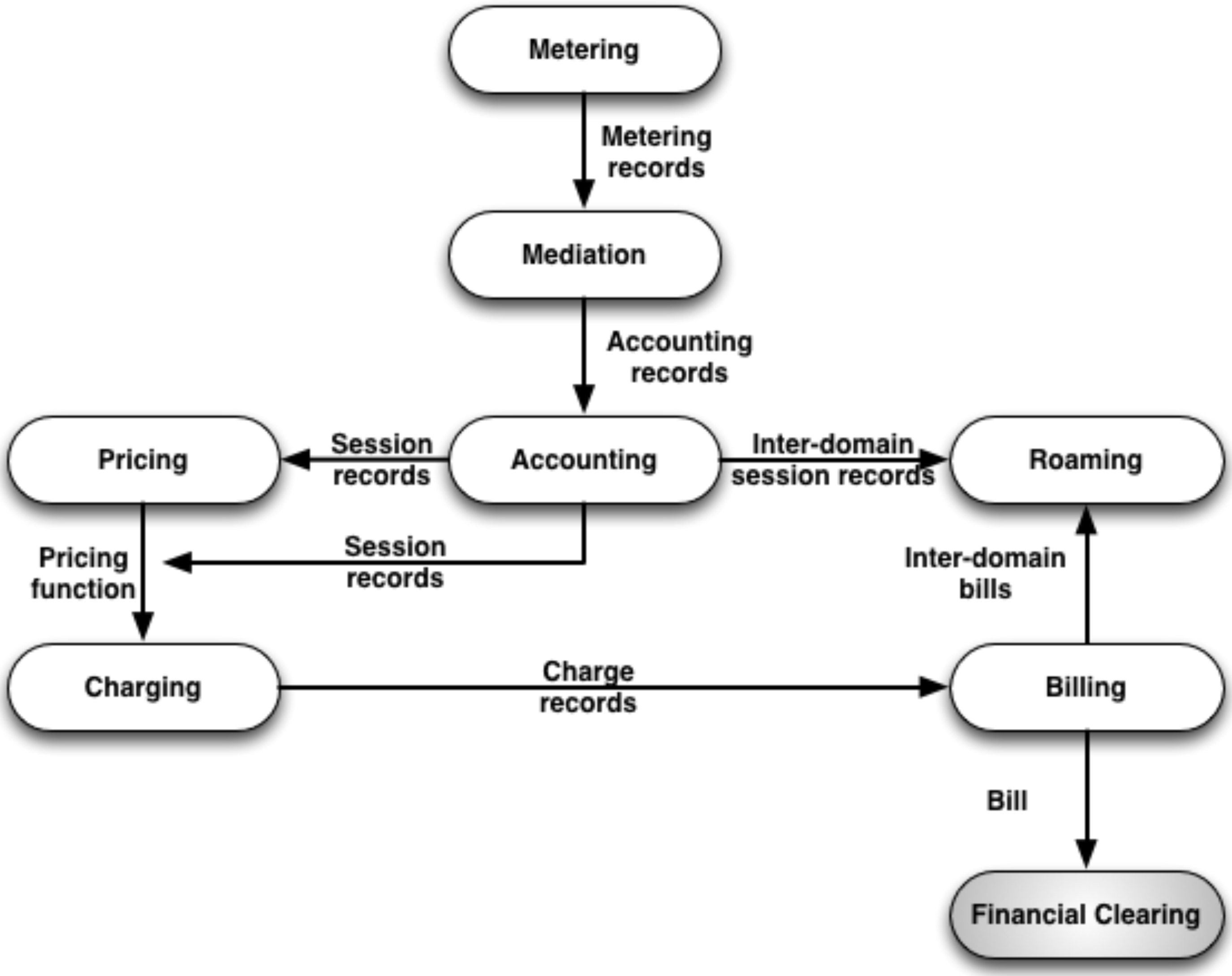
❑ You fix the price strategy

❑ You bill your customers based on the usage data and the pricing strategies!

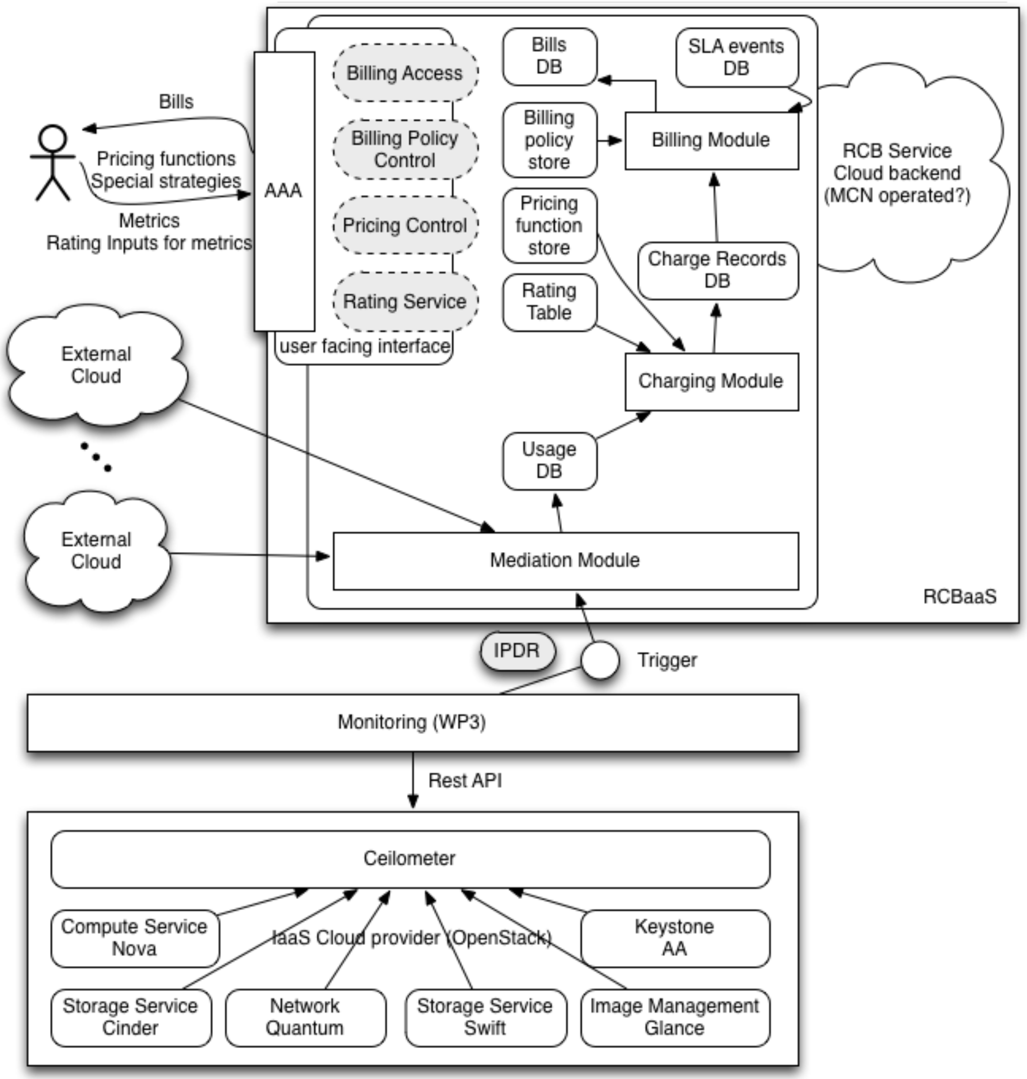
So why not do the generic rating-charging-billing platform (correctly) where everything is parameterized and configurable?

Accounting process

Our basic guiding force …



High level architecture

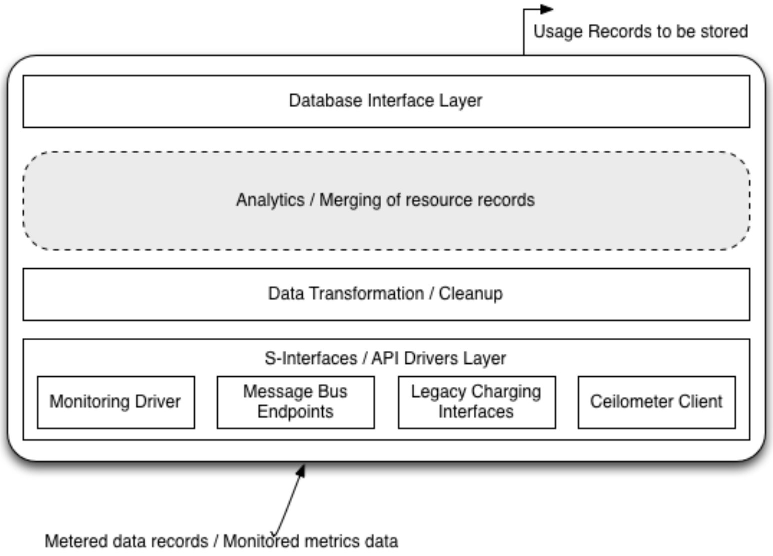


- Closely aligned with accounting process
- Highly modular.
- Distributed design very conducive for HA solutions

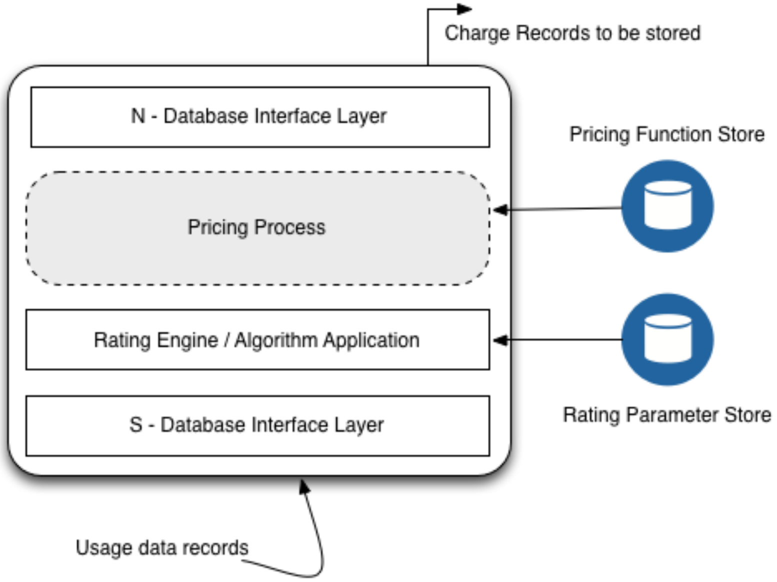
Individual Components …

MCN Mobile Cloud Networking

Mediation Module Architecture

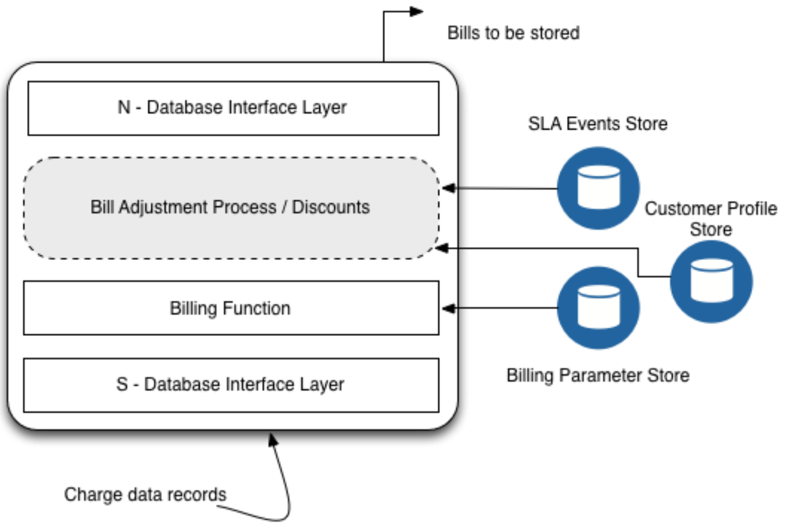

Usage Records to be stored
Database Interface Layer
Analytics / Merging of resource records
Data Transformation / Cleanup
S-Interfaces / API Drivers Layer
Monitoring Driver
Message Bus Endpoints
Legacy Charging Interfaces
Ceilometer Client
Metered data records / Monitored metrics data

Charging Module Architecture

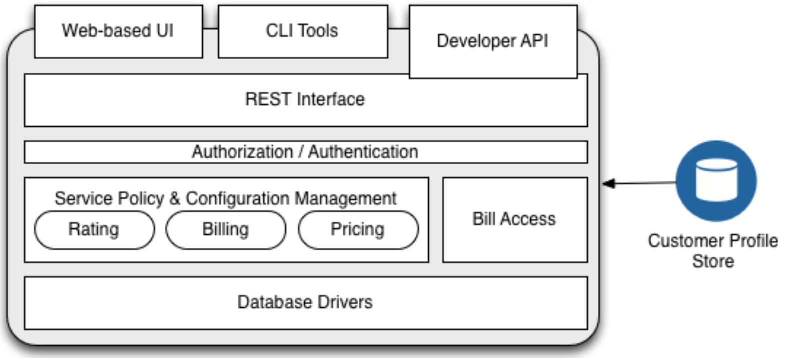

Charge Records to be stored
N - Database Interface Layer
Pricing Function Store
Pricing Process
Rating Engine / Algorithm Application
Rating Parameter Store
S - Database Interface Layer
Usage data records

Individual Components …

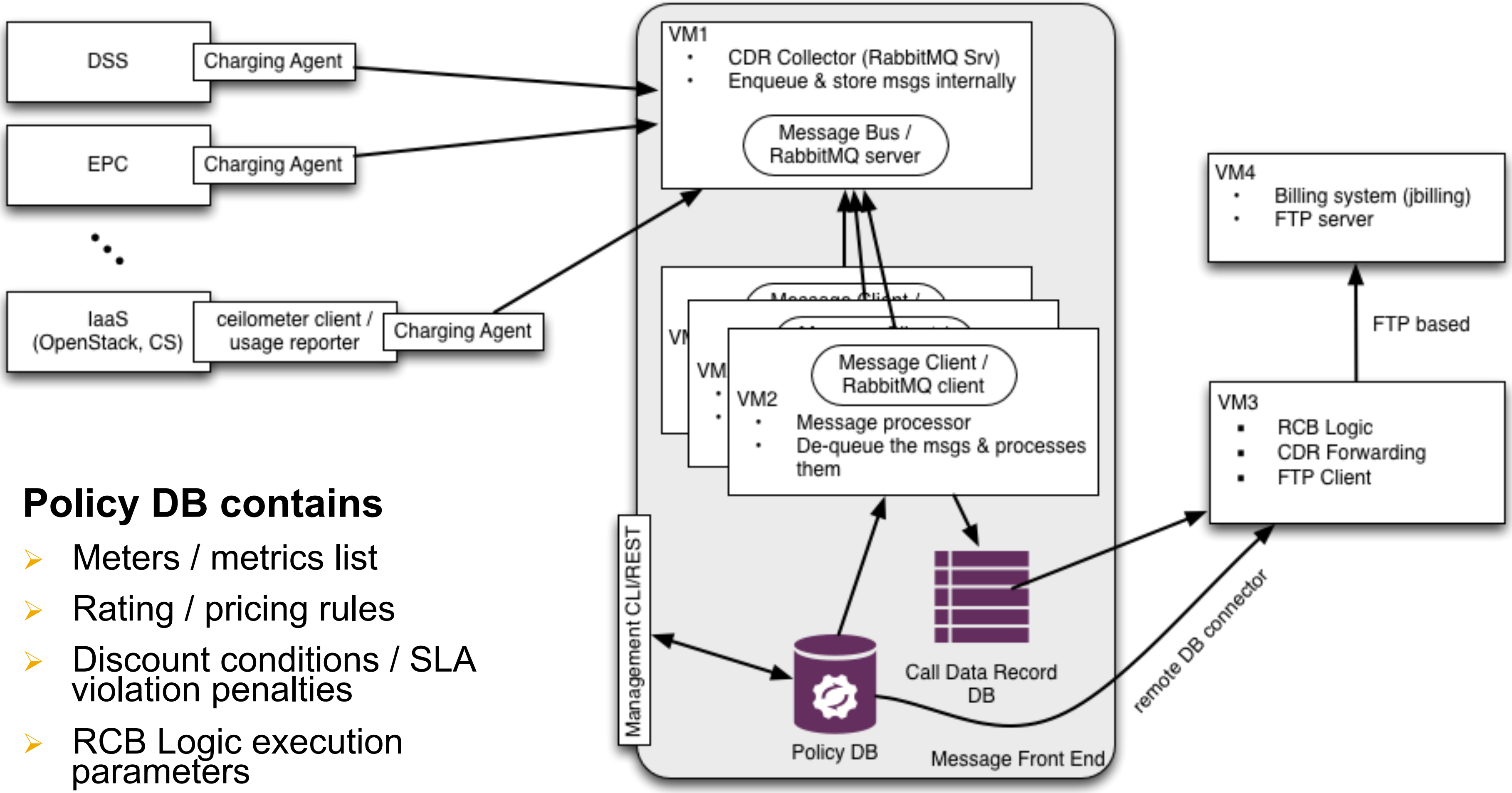
Billing Module Architecture



UI Architecture



First MCN prototype

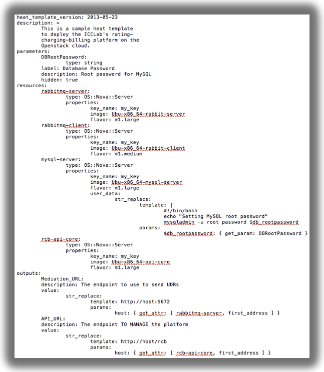


Policy DB contains

- Meters / metrics list
- Rating / pricing rules
- Discount conditions / SLA violation penalties
- RCB Logic execution parameters

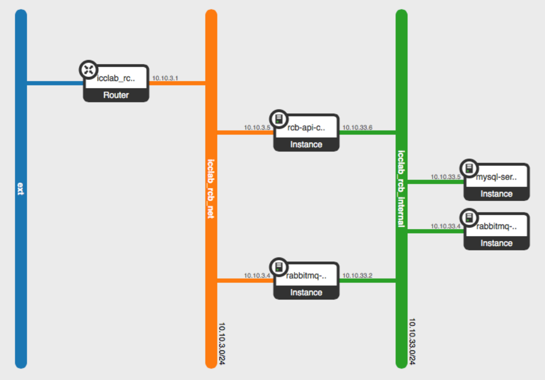
Provisioning on demand

Rating charging billing as a service



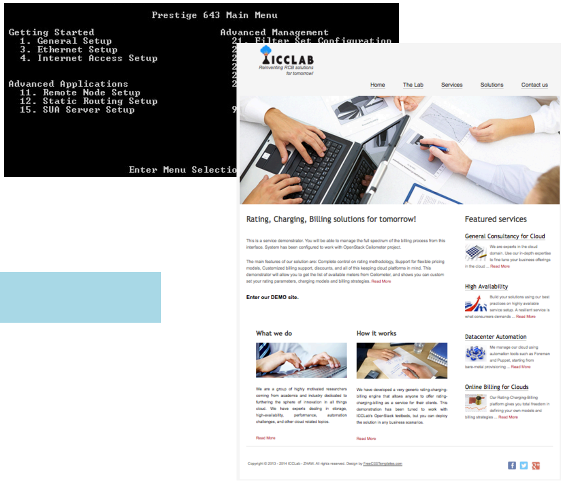
```
heat_template_version: 2013-05-23
description: >
        This is a sample heat template
        to deploy the ICCLab's rating-
        charging-billing platform on the
        Openstack cloud.
parameters:
        DBRootPassword:
                type: string
        label: Database Password
        description: Root password for MySQL
        hidden: true
resources:
        rabbitmq-server:
                type: OS::Nova::Server
                properties:
                        key_name: my_key
                        image: Ubu-x86_64-rabbit-server
                        flavor: m1.large
        rabbitmq-client:
                type: OS::Nova::Server
                properties:
                        key_name: my_key
                        image: Ubu-x86_64-rabbit-client
                        flavor: m1.medium
        mysql-server:
                type: OS::Nova::Server
                properties:
                        key_name: my_key
                        image: Ubu-x86_64-mysql-server
                        flavor: m1.large
                        user_data:
                                str_replace:
                                        template: |
                                                #!/bin/bash
                                                echo "Setting MySQL root password"
                                                mysqladmin -u root password $db_rootpassword
                                        params:
                                                $db_rootpassword: { get_param: DBRootPassword }
        rcb-api-core:
                type: OS::Nova::Server
                properties:
                        key_name: my_key
                        image: Ubu-x86_64-api-core
                        flavor: m1.large
outputs:
        Mediation_URL:
        description: The endpoint to use to send UDRs
        value:
                str_replace:
                        template: http://host:5672
                        params:
                                host: { get_attr: [ rabbitmq-server, first_address ] }
        API_URL:
        description: The endpoint TO MANAGE the platform
        value:
                str_replace:
                        template: http://host/rcb
                        params:
                                host: { get_attr: [ rcb-api-core, first_address ] }
```

heat



Service management interface

Service user interface



```
                        Prestige 643 Main Menu

Getting Started                        Advanced Management
  1. General Setup                        21. Filter Set Configuration
  3. Ethernet Setup                       2
  4. Internet Access Setup                2
                                          2
                                          2
Advanced Applications                     2
  11. Remote Node Setup
  12. Static Routing Setup
  15. SUA Server Setup                    9



                 Enter Menu Selectio
```

Rating, Charging, Billing solutions for tomorrow!

This is a service demonstrator. You will be able to manage the full spectrum of the billing process from this interface. System has been configured to work with OpenStack Ceilometer project.

The main features of our solution are: Complete control on rating methodology, Support for flexible pricing models, Customized billing support, discounts, and all of this keeping cloud platforms in mind. This demonstrator will allow you to get the list of available meters from Ceilometer, and shows you can custom set your rating parameters, charging models and billing strategies. Read More

Enter our DEMO site.

What we do

We are a group of highly motivated researchers coming from academia and industry dedicated to furthering the sphere of innovation in all things cloud. We have experts dealing in storage, high-availability, performance, automation challenges, and other cloud related topics.

Read More

How it works

We have developed a very generic rating-charging-billing engine that allows anyone to offer rating-charging-billing as a service for their clients. This demonstration has been tuned to work with ICCLab's OpenStack testbeds, but you can deploy the solution in any business scenarios.

Read More

Featured services

General Consultancy for Cloud

We are experts in the cloud domain. Use our in-depth expertise to fine tune your business offerings in the cloud ... Read More

High Availability

Build your solutions using our best practices on highly available service setup. A resilient service is what consumers demands ... Read More

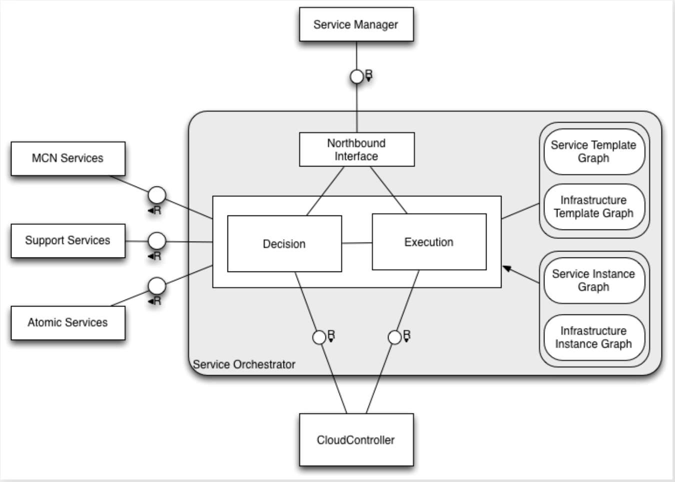
Datacenter Automation

Me manage our cloud using automation tools such as Foreman and Puppet, starting from bare-metal provisioning ... Read More

Online Billing for Clouds

Our Rating-Charging-Billing platform gives you total freedom in defining your own models and billing strategies ... Read More

Copyright © 2013 - 2014 ICCLab - ZHAW. All rights reserved. Design by FreeCSSTemplates.com



Aligned to the MCN orchestration design recommendations.

State of the development

Roadmap, next steps (MCN perspective)

What is in the code today –

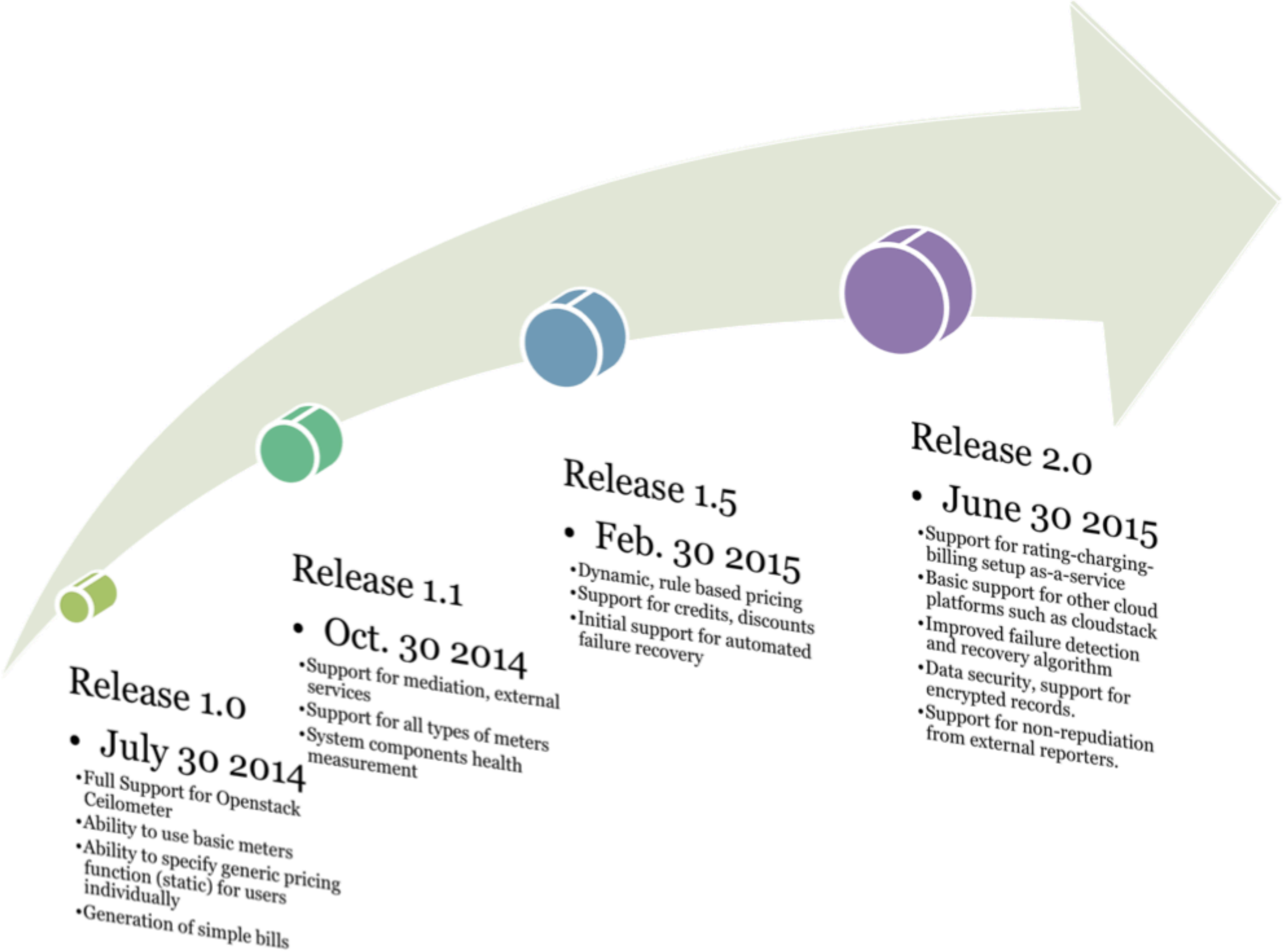
- ✓ OpenStack usage data extraction through telemetry-client
- ✓ Ability to receive resource usage data from other services (legacy services)
- ✓ Storage of CDRs into a CDR store
- ✓ Forwarding of CDRs to the billing software (jBilling)

What is planned in the near future –

- ✓ Common interface with the MCN monitoring service
  - ✓ All metrics and usage data to be extracted from this monitoring service
- ✓ Service customization and management endpoint (REST interface)
- ✓ RCB policy engine implementation (decision trees, rule based inference …)

State of the development …

Roadmap from CYCLOPS development perspective



Release 1.0

- July 30 2014
  - Full Support for Openstack Ceilometer
  - Ability to use basic meters
  - Ability to specify generic pricing function (static) for users individually
  - Generation of simple bills

Release 1.1

- Oct. 30 2014
  - Support for mediation, external services
  - Support for all types of meters
  - System components health measurement

Release 1.5

- Feb. 30 2015
  - Dynamic, rule based pricing
  - Support for credits, discounts
  - Initial support for automated failure recovery

Release 2.0

- June 30 2015
  - Support for rating-charging-billing setup as-a-service
  - Basic support for other cloud platforms such as cloudstack
  - Improved failure detection and recovery algorithm
  - Data security, support for encrypted records.
  - Support for non-repudiation from external reporters.

Further information …

Code repositories, project deliverable reports…

- Code Repositories
  - https://git.mobile-cloud-networking.eu/wp5/rcbaas (restricted at the moment)
  - https://github.com/icclab/cyclops-web (ICCLab's OpenStack RCB engine front-end, open source)
  - https://github.com/icclab/cyclops-web (ICCLab's OpenStack RCB engine, open source)
- Documents, Deliverables
  - http://mobile-cloud-networking.eu/site/ (Project site)
- Demo Sites / Videos
  - Cyclops demo: https://www.youtube.com/watch?v=ZIwwVxqCio0
  - Online demo site: http://cyclops.cloudcomplab.ch/

More about InIT Cloud Computing Lab @ www.cloudcomp.ch

Twitter: @ICC_Lab

THANK YOU!

MCN
Mobile Cloud Networking

© 2012-2015 MCN. All Rights Reserved

MCN Mobile Cloud Networking

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright 2012 - 2015 by MCN Consortium.