Advanced Least Cost Routing with Kamailio using CGRateS

Dan Christian Bogos
dan.bogos@itsyscom.com

Kamailio World, May 2015

Our Background

ITSYSCOM
Engineered Solutions

Located in Bavaria/Germany, over 8 years of experience with architecting server side solutions in VoIP environment

Platform implementations covering both wholesale and retail business categories

Responsibly understanding real-time processing constrains and the seriousness of live system outages

CGRateS
carrier grade realtime charging

About CGRateS

Charging/Billing engine

Plug-able into existing billing infrastructure
Accommodate new components into ISP/ITSP network (eg: add new VoIP switch, SMS Service, Data stream)
Non-intrussive into existing setups

Modular architecture

Easy to enhance by rewriting specific components - JSON/HTTP/GOB RPC API

Performance Oriented

Built-in transactional cache system (data ageing, live counters)
Asynchronous processing with micro-threads

Feature-rich

Multi-tenancy, derived charging, account bundles, LCR, CDRStats, rates history, etc
Agile in developing new features

Test driven development

Aprox. 900 tests as part of the build system

CGRateS
carrier grade realtime charging

About CGRateS (2)

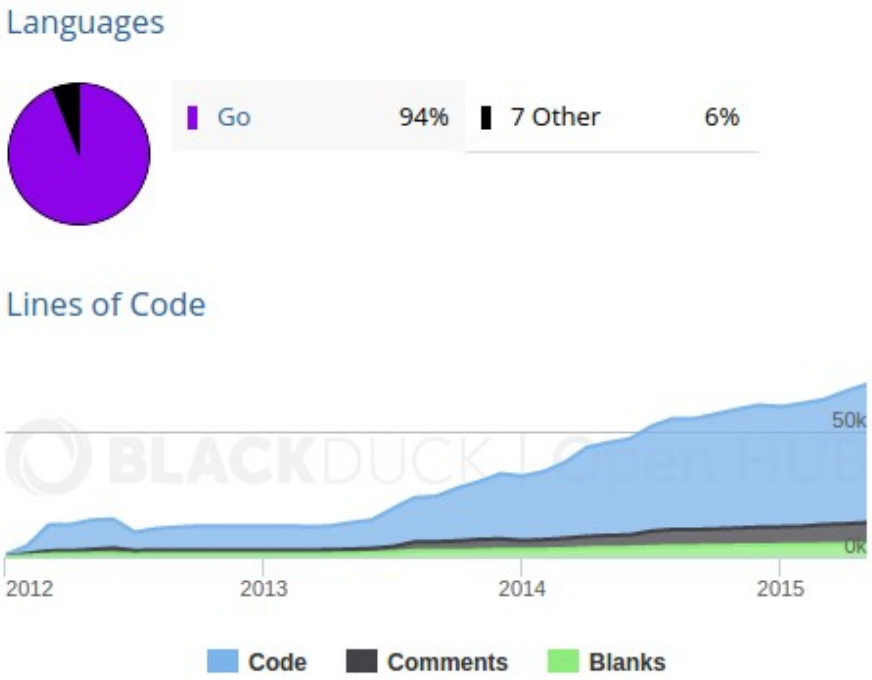
In a Nutshell, cgrates...

··· has had 2,722 commits made by 6 contributors
representing 55,066 lines of code

··· is mostly written in Go
with an average number of source code comments

··· has a codebase with a long source history
maintained by a average size development team
with stable Y-O-Y commits

··· took an estimated 14 years of effort (COCOMO model)
starting with its first commit in January, 2012
ending with its most recent commit about 7 hours ago



Actively maintained

*stats provided by openhub.net

CGRATES
carrier grade realtime charging

RATING

- Functionality: calculate costs for events
- Isolated in calculations from other subsystems
- Fully cache driven, async processing
- Referenced from other subsystems (eg: Accounting, LCR)
- Standalone component, RPC/in-process accessible

ACCOUNTING

- Functionality: maintain accounts with balances
- Partial cache driven (accounts are kept in dataDb/Redis).
- Async processing with account locking
- Real-time fraud detection/mitigation at account level during balance operations (locked stage).
- Queued/scheduled operations on accounts

CDR SERVER

- Functionality:
  - store CDRs from various sources
  - rate CDRs using Rating subsystem
  - replicate CDRs (rated or raw ones) via RPC/HTTP to other servers
  - provide rated/raw CDRs to CDR Stats subsystem
- Asynchronous processing
- Standalone component, RPC/in-process accessible

LCR

- Functionality: compute real-time LCR
- Fully cache driven, async processing
- Depending on strategy used, references real-time data from other subsystems (EG: Rating, Accounting, CDR Stats)

HISTORY SERVER

- Functionality: archive rate changes using GIT in human readable JSON format
- Async, fully cached with scheduled disk dumps
- Standalone component, RPC/in-process accessible

CDR STATS

- Functionality: calculate CDR stats in real-time based on data received from various sources
- Real-time fraud detection/mitigation with actions triggered.
- Async, fully cached
- Standalone component, RPC/in-process accessible

CGRateS subsystems

CGRATES
carrier grade realtime charging

CGR-LCR Overview

Core component logic
Internally or remotely accessible through APIer or RATER components
Non-intrusive, injects supplier information into Kamailio pseudovariables

Tightly coupled with ACCOUNTING subsystem
Provides LCR over bundles

Integrate traffic patterns
Computes LCR for specific call duration

Advanced profile selection mechanism
Filter on Direction,Tenant,Category,Account,Subject,Destination
Weight based prioritization
Activation time

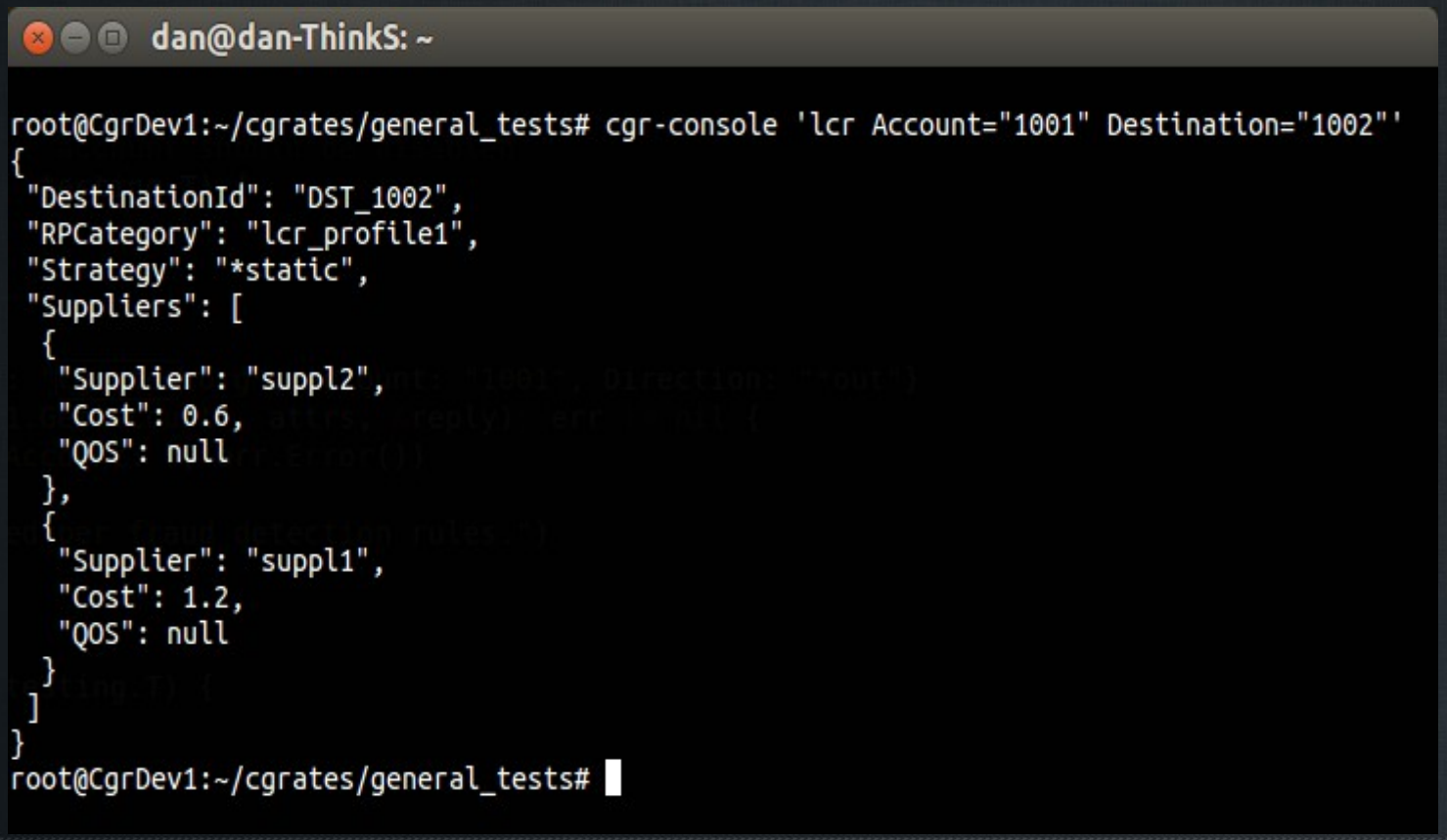
Extended functionality through multiple strategies
*static, *least_cost, *highest_cost, *qos_thresholds, *qos
Flexible strategy parameters

CGRateS
carrier grade realtime charging

*static

Classic way of LCR, suppliers ordered based on configured rule parameters

“*out,cgrates.org,call,1001,*any,DST_1002,lcr_profile1,*static,suppl2;suppl1,2014-01-14T00:00:00Z,10”



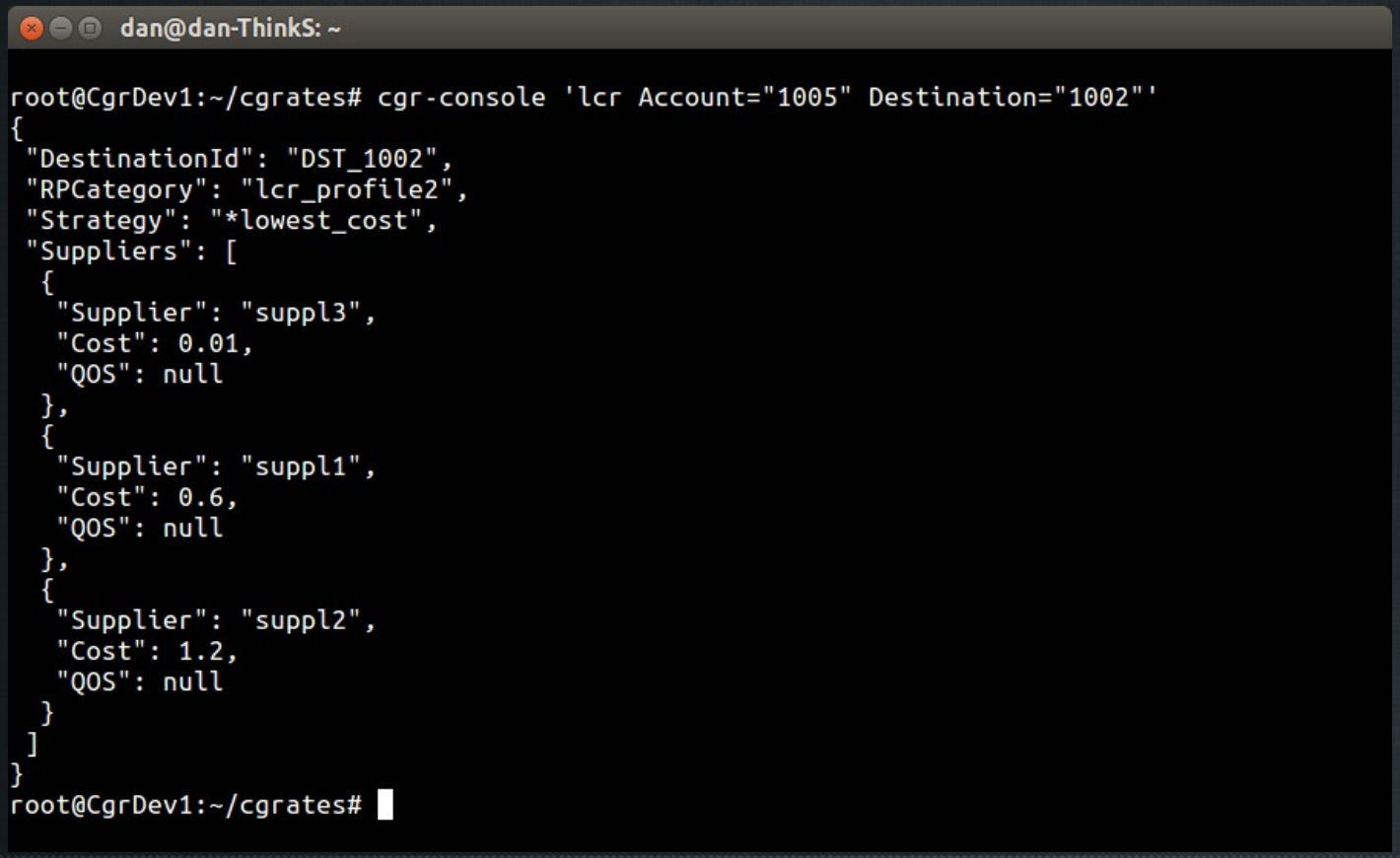
LCR Strategies (1)

CGRATES
carrier grade realtime charging

*lowest_cost

Use supplier with least cost

“*out,cgrates.org,call,*any,*any,*any,lcr_profile1,*lowest_cost,,2014-01-14T00:00:00Z,10”



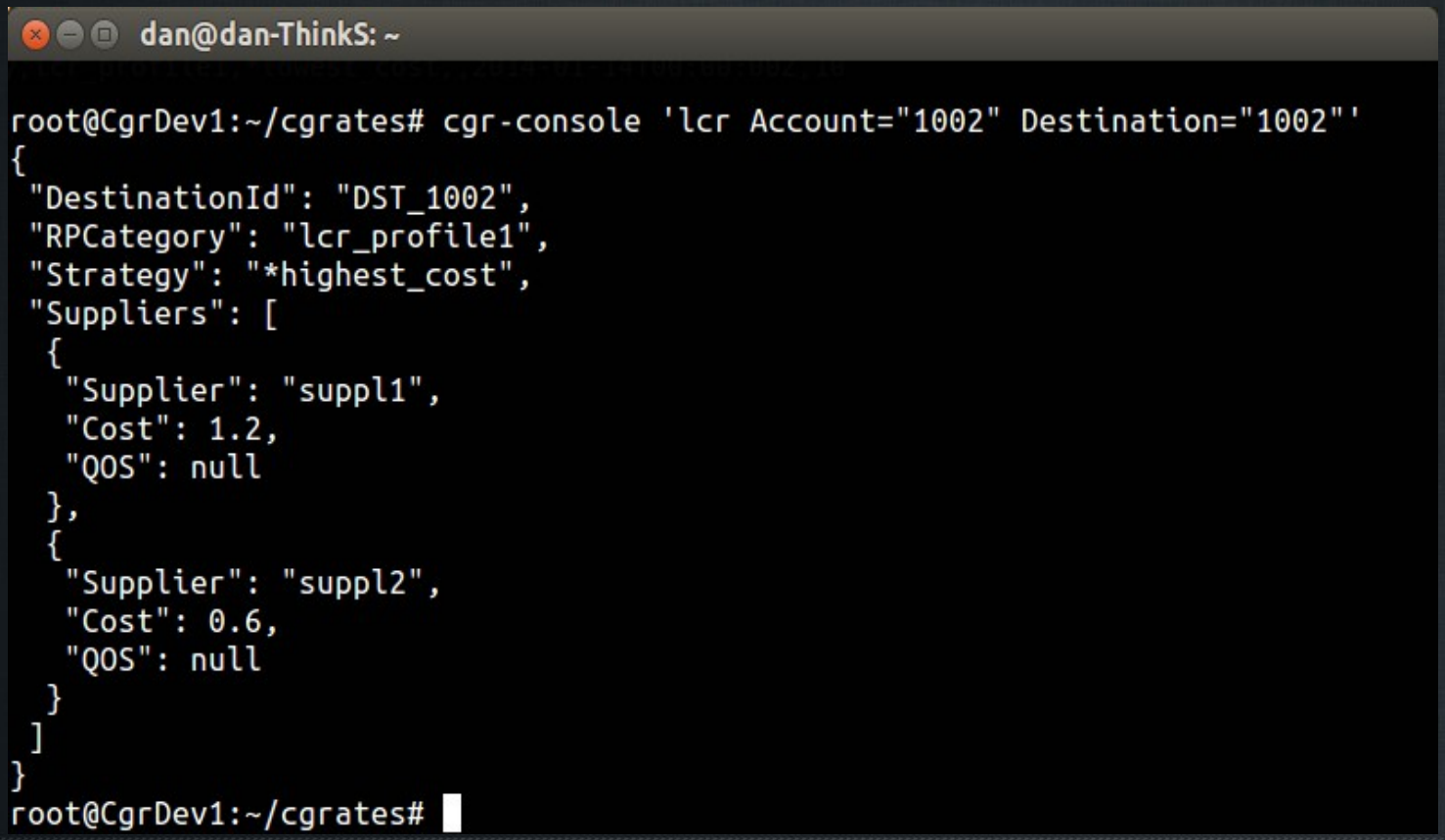
LCR Strategies (2)

CGRATES
carrier grade realtime charging

*highest_cost

Use supplier with highest cost

“*out,cgrates.org,call,1002,*any,DST_1002,lcr_profile1,*highest_cost,,2014-01-14T00:00:00Z,10”



LCR Strategies (3)

CGRATES
carrier grade realtime charging

*qos_threshold

Supplier with lowest cost, matching QoS thresholds min/max for ASR, ACD, TCD, ACC, TCC

“*out,cgrates.org,call,1003,*any,DST_1002,lcr_profile1,*qos_threshold,20;;2m;;;;;;;,2014-01-14T00:00:00Z,10”

dan@dan-ThinkS: ~

```
root@CgrDev1:~/cgrates/general_tests# cgr-console 'lcr Account="1003" Destination="1002"'
{
 "DestinationId": "DST_1002",
 "RPCategory": "lcr_profile1",
 "Strategy": "*qos_threshold",
 "Suppliers": [
  {
   "Supplier": "suppl1",
   "Cost": 1.2,
   "QOS": {
    "ACC": 0.35,
    "ACD": 120,
    "ASR": 100,
    "TCC": 0.7,
    "TCD": 240
   }
  }
 ]
}
root@CgrDev1:~/cgrates/general_tests#
```

LCR Strategies (4)

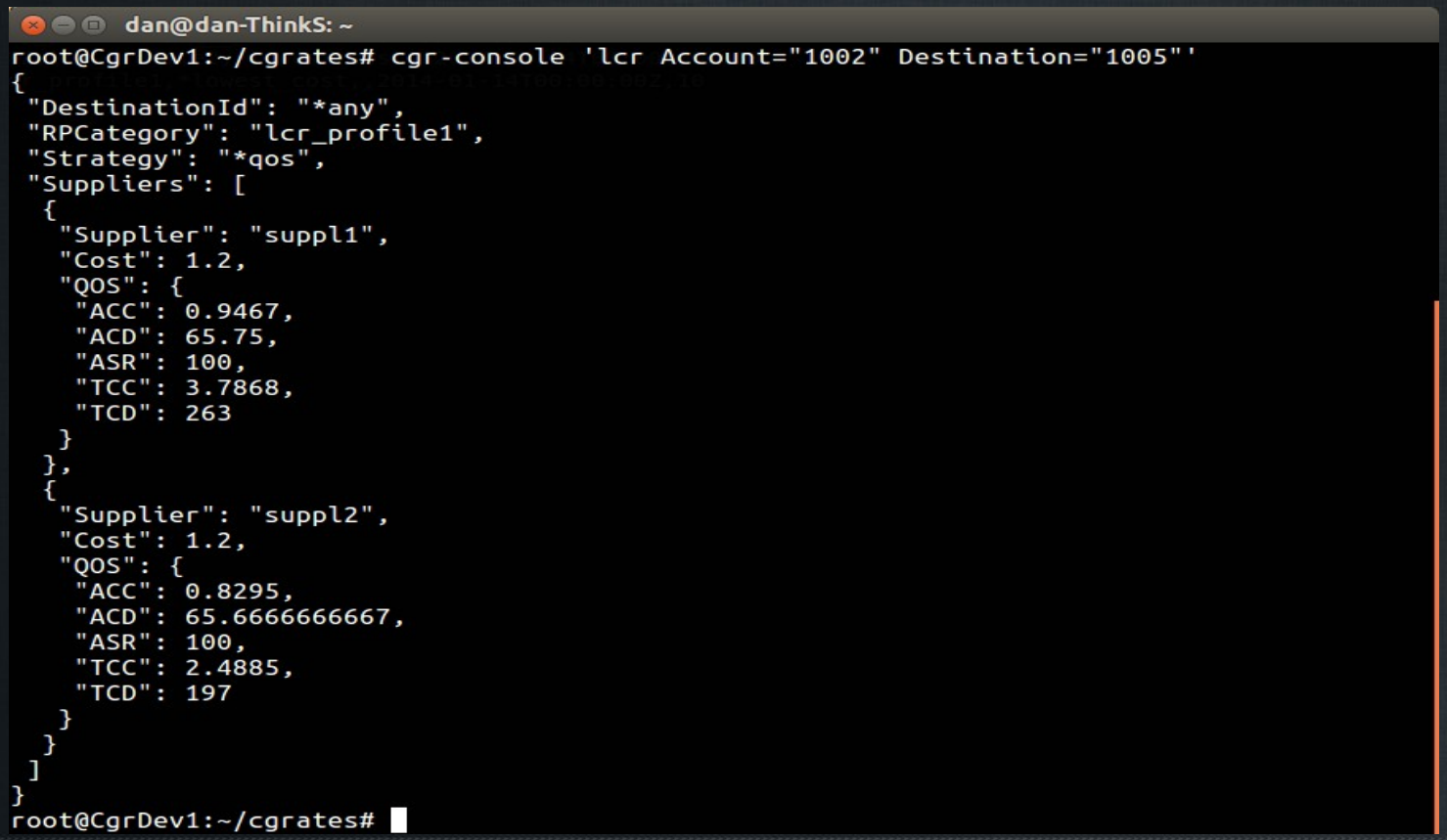
CGRATES

carrier grade realtime charging

*qos

Supplier with best quality, independent on cost

“*out,cgrates.org,call,1002,*any,*any,lcr_profile1,*qos,,2014-01-14T00:00:00Z,10”



LCR Strategies (5)

CGRATES
carrier grade realtime charging

Kamailio Integration (1)

Inside CGR_AUTH_REQUEST route

```
T 2015/05/28 12:35:38.932624 127.0.0.1:8448 -> 127.0.0.1:54118 [AP]
250:{"event":"CGR_AUTH_REQUEST",
.."tr_index":"9685",
.."tr_label":"753972297",
.."cgr_reqtype":"*pseudoprepaid",
.."cgr_tenant":"cgrates.org",
.."cgr_account":"1003",
.."cgr_destination":"1001",
.."cgr_setuptime":"1432809338",
.."cgr_computelcr":"true"},
##
T 2015/05/28 12:35:38.938287 127.0.0.1:54118 -> 127.0.0.1:8448 [AP]
140:
{"Event":"CGR_AUTH_REPLY","TransactionIndex":9685,"TransactionLabel":753972
297,"MaxSessionTime":2940,"Suppliers":"suppl1,suppl2","Error":""},
```

Kamailio Integration (2)

Inside CGR_LCR_REQUEST route

```
T 2015/05/28 12:31:32.440824 127.0.0.1:8448 -> 127.0.0.1:54014 [AP]
189:{"event":"CGR_LCR_REQUEST",
.."tr_index":"45485",
.."tr_label":"159484172",
.."cgr_tenant":"cgrates.org",
.."cgr_account":"1002",
.."cgr_destination":"1001",
.."cgr_setuptime":"1432809092"},
##
T 2015/05/28 12:31:32.441844 127.0.0.1:54014 -> 127.0.0.1:8448 [AP]
138:
{"Event":"CGR_LCR_REPLY","TransactionIndex":45485,"TransactionLabel":159484
172,"MaxSessionTime":-1,"Suppliers":"suppl2,suppl1","Error":""},
```

Where to go from here

Website
http://www.cgrates.org

Documentation
http://cgrates.readthedocs.org

Code + issues tracker
https://github.com/cgrates/cgrates

Support
Google group: CGRateS
IRC Freenode: #cgrates

CGRateS
carrier grade realtime charging

Thank you!

Questions?

CGRateS
carrier grade realtime charging