Building CLASS 5 CDRs with OpenSIPS and RabbitMQ

Alex Goulis
My experience

• Designing multi-tenant business VoIP platforms since 2009
• Lead developer for Ratetel's Virtual PBX and trunking platform
• First certified OpenSIPS professional
• Ratetel is the US sales partner for OpenSIPS Solutions
Advantages of using OpenSIPS

- Highly scalable
- Stable code base
- Can handle tens of thousands of registrations
- Central point for presence and billing
- Dynamic routing
- Packet mangling to alter packets for custom purposes
- Highly available
Advantages of using Freeswitch

- Supports more concurrent calls than most other open source PBXs (asterisk)
- Rich media handling capabilities
- Many different config methods (flat xml, lua, dynamic xml, many others)
- Stable code base and long time affinity with Opensips
- So many class 5 features, even ones you didn't think you needed
What is RabbitMQ?

- RabbitMQ is an open source message broker software (sometimes called message-oriented middleware) that implements the Advanced Message Queuing Protocol (AMQP).

- The RabbitMQ server is written in the Erlang programming language and is built on the Open Telecom Platform framework for clustering and failover.
Advantages of using RabbitMQ

- Robust messaging for applications
- Easy to use
- Runs on all major operating systems
- Supports a huge number of developer platforms
- Open source and commercially supported
- Reliable queuing
- Topic-based publish-and-subscribe messaging
- Flexible routing, transactions, and security.
Traditional CDR problems

- Class 4 and Class 5 are usually separate systems from a CDR perspective.
- There are 2 sets of CDRs generated with different data, but most importantly different call-ids.
- Class 5 events during a call are not naturally logged to CDR, especially call transfers and dtmf input.
- Complex routing plans make it more problematic
Network diagram

Supplies call start/answer/end times for billing

Opensips

Rabbit Consumer

Mysql

Fs

Supplies events such as call state, routing and dtmf

App Server

Web Server

Store data in mysql

Make live call info available to web or app servers
Event Methodology

- Events can be consumed from the RabbitMQ server by any choice of clients available.
- OpenSIPS is responsible for information related to call's start and end time, as well as marking billable time at the start of media.
- Freeswitch will append call information based on events occurring in the CLASS 5 layer such as where the call is routed, when it's parked, put on hold, transferred, etc.
Who's handling what...

OpenSIPS
- Registrations
- IP Authentication
- Carrier facing
- nat

Freeswitch
- Call routing
- Voicemail
- Ivr
- Ring groups
- Queues
- Conferences
Opensips configuration

- loadmodule "event_rabbitmq.so"
  - librabbitmq-dev required
- modparam("event_rabbitmq", "heartbeat", 3)
  - Enables heartbeat support for the AMQP communication. If no heartbeat from server is received within the specified interval, the socket is automatically closed.
  - Prevents OpenSIPS from blocking while waiting for a response from a dead rabbitmq-server. The value represents the heartbeat interval in seconds
Opensips Configuration

- modparam("event_rabbitmq", "sync_mode", 0)
  - 0 = default (async non-blocking)
  - 1 = synchronous (opensips waits for response)
- subscribe_event("E_RABBITMQ_EVENT", "rabbitmq:127.0.0.1/queue");
- raise_event("E_RABBITMQ_EVENT");
- The maximum length of a datagram event is 16384 bytes
Raising events in OpenSIPS

- Inject variables like CALLID, SRC, DST, starttime into avp variables
- `raise_event("E_SIP_MESSAGE", $avp(attrs), $avp(vals))`
- Calling this function on INVITE will send the first event to open a CDR record
- Calling this function on reply route will signal the start of media (billable time)
- Calling this function on BYE or CANCEL will signal the close of the CDR record
Opensips Configuration

- Because new call-ids will be generated when calls are sent to CLASS 5, we must find a way to bind them to CLASS 4.
  
- `append_hf("X-ORIGINAL-CALLID: $ci\r\n");`
  
- All calls delivered to CLASS 5 will have this callid to reference as it's made available as a variable in all events sent from Freeswitch.
Freeswitch Configuration

- autoload_configs/modules.conf.xml
  - Add `<load module="mod_amqp"/>
- autoload_configs/amqp.conf.xml

```xml
<profile name="default">
  <connections>
    <connection name="primary">
      <param name="hostname" value="localhost"/>
      <param name="virtualhost" value="/"/>
      <param name="username" value="guest"/>
      <param name="password" value="guest"/>
      <param name="port" value="5672"/>
      <param name="heartbeat" value="3"/>
    </connection>
  </connections>
</profile>
```
Freeswitch Events

• Customize the Event Filter by editing the following lines. The default captures channel create and destroy, fs heartbeat, and dtmf.

<param name="eventFilter" value="SWITCH_EVENT_ALL"/>

<param name="event_filter" value="SWITCH_EVENT_CHANNEL_CREATE,SWITCH_EVENT_CHANNEL_DESTROY,SWITCH_EVENT_HEARTBEAT,SWITCH_EVENT_DTMF,SWITCH_EVENT_CHANNEL_HOLD,SWITCH_EVENT_CHANNEL_UNHOLD,SWITCH_EVENT_CHANNEL_PARK,SWITCH_EVENT_CHANNEL_UNPARK,SWITCH_EVENT_CHANNEL_STATE,SWITCH_EVENT_CHANNEL_ANSWER,SWITCH_EVENT_CHANNEL_CALL_STATE"/>
Freeswitch Events

- Bind the original call-id to new channels
- Use events to follow call activity in realtime
  - `<action application="set" data="sip_h_X-ORIGINAL-CALLID=${sip_h_X-ORIGINAL-CALLID}"/>
- Track answers, hangups, transfers for basic CDR creation
- Enhance by injecting call data like DTMF, call parking/unparking, call hold/unhold, recording start/stop, CHANSPY events
Freeswitch Events

- **CHANNEL_ANSWER**
  - Will provide all channel variables including custom sip headers in the event
  - First bind on original callid

- **CHANNEL_BRIDGE**
  - Used to detect transfers as it provides all channel variables for both legs to be bridged

- **CHANNEL_HANGUP_COMPLETE**
  - Used to detect call hangup, all variables and sip headers available
Other Events in Opensips

- Can be used to track a multitude of other events in OpenSIPS as needed.
- Examples:
  - Alerts when counters are breached
  - Alerts when gateways become available/unavailable
  - Alerts when users register/unregister
  - Alerts when calls fail
  - Alerts on attacks such as floods, etc
Other Events in Freeswitch

- Like in OpenSIPS, can be used to track many different kinds of events.
- Examples:
  - Conference rooms and user actions within
  - Voicemail box info after exiting mod_voicemail
  - Pin failures for call authorizations
  - Sending system status
  - Sending status of apps executed by dialplan
Thank You!

Alex Goulis
a.g@ratetel.com

Questions?