Alex Goulis

Using OpenSIPS as your registrar to deliver load balanced class 5 features with Freeswitch

OpenSips Workshop @Cluecon
Chicago 2015

My experience

- Designing multi-tenant business VoIP platforms since 2009
- Lead developer for Ratetel's Virtual PBX and trunking platform
- First certified OpenSIPS professional

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

Opensips configuration (multi-tenant)

- Use dispatcher module to catalog your freeswitch servers
- Will automatically remove servers when down, and add them when available again.
- Dispatcher groups are configured with a primary and secondary server
- Physically shard your tenants across groups to spread resources available
- Problems will occur if traditionally load balanced

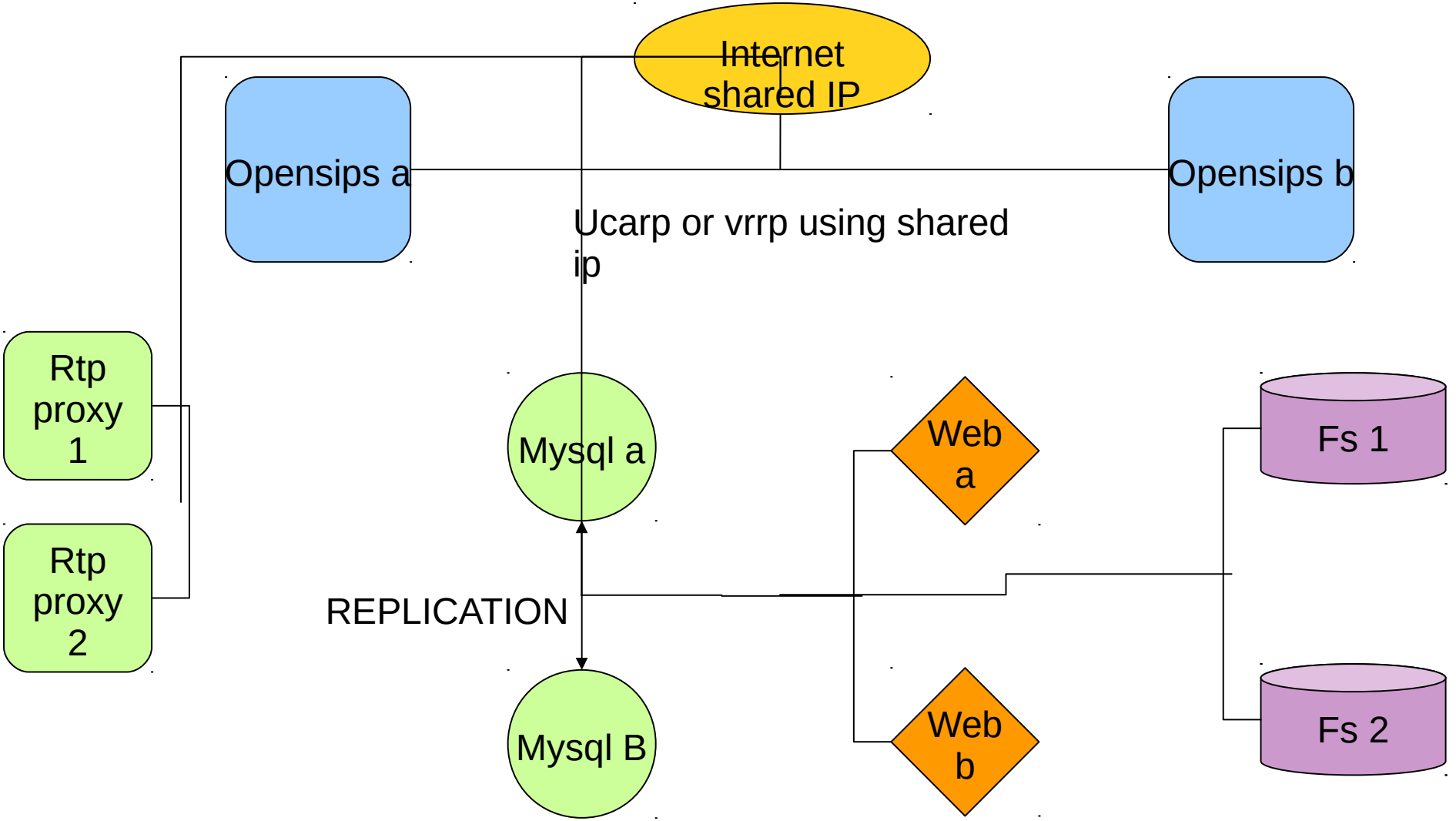
High Availability

- Same dc failover for OpenSIPS using servers ucarp or vrrp
- Automatic removal of fs servers when down
- Automatically includes fs servers when they become available
- HA Proxy for web servers
- Mysql backend for fs configuration

Registering at Opensips

- Takes the load off of Freeswitch
- Large numbers of registrations will start disrupting normal operations
- Opensips was built to handle registrations
- No presence info at Freeswitch, but handled by Opensips
- Freeswitch servers now act as just feature servers

Network diagram



mod_xml_curl

- Freeswitch module to feed realtime xml data to freeswitch including configuration data
- Simple to implement using many languages (php, python, java, etc)
- fs_curl available in fs contrib, great primer
- Using fs_curl can be configured using included db tables or via custom xml hooks
- We will use PHP with xml writer for our custom extensions

Mysql db

- Powerful and free
- Provides backend storage for configurations in both OpenSIPS and Freeswitch
- HA is implemented using Master/Master
- For geo located databases use Percona cluster

Apache web server

- Most widely used http server
- Built in support for PHP
- Resilient and easy to scale
- Easy to configure

Php xml writer

- Shipped with PHP
- Easy to understand and simple to use
- Builds structured XML for Freeswitch to consume

Freeswitch call groups

```
$obj -> xmlw -> startElement('section');

$obj -> xmlw -> writeAttribute('name', 'dialplan');

$obj -> xmlw -> writeAttribute('description', 'FreeSWITCH Dialplan');

$obj -> xmlw -> startElement('context');

$obj -> xmlw -> writeAttribute('name', 'ring_group');

$obj -> xmlw -> startElement('extension');

$obj -> xmlw -> startElement('condition');

$obj -> xmlw -> startElement('action');

$obj -> xmlw -> writeAttribute('application', 'bridge');

$obj -> xmlw -> writeAttribute('data', 'sofia/1021%${domain_name},sofia/multi.domain-a.com/1022%${domain_name}');

$obj -> xmlw -> endElement();

$obj -> xmlw -> endElement(); //</extension>

$obj -> xmlw -> endElement(); //</context>

$obj -> xmlw -> endElement(); //</section>
```

Freeswitch voicemail

```
$obj -> xmlw -> startElement('section');

$obj -> xmlw -> writeAttribute('name', 'dialplan');

$obj -> xmlw -> writeAttribute('description', 'FreeSWITCH Dialplan');

$obj -> xmlw -> startElement('context');

$obj -> xmlw -> writeAttribute('name', 'ring_group');

$obj -> xmlw -> startElement('extension');

$obj -> xmlw -> startElement('condition');

$obj -> xmlw -> startElement('action');

$obj -> xmlw -> writeAttribute('application', 'voicemail');

$obj -> xmlw -> writeAttribute('data', 'default $${domain} ${dialed_extension}');

$obj -> xmlw -> endElement();

$obj -> xmlw -> endElement(); //</extension>

$obj -> xmlw -> endElement(); //</context>

$obj -> xmlw -> endElement(); //</section>
```

Freeswitch conference

```
$obj -> xmlw -> startElement('section');

$obj -> xmlw -> writeAttribute('name', 'dialplan');

$obj -> xmlw -> writeAttribute('description', 'FreeSWITCH Dialplan');

$obj -> xmlw -> startElement('context');

$obj -> xmlw -> writeAttribute('name', 'conference');

$obj -> xmlw -> startElement('extension');

$obj -> xmlw -> startElement('condition');

$obj -> xmlw -> startElement('action');

$obj -> xmlw -> writeAttribute('application', 'conference');

$obj -> xmlw -> writeAttribute('data', conference_name');

$obj -> xmlw -> endElement();

$obj -> xmlw -> endElement(); //</extension>

$obj -> xmlw -> endElement(); //</context>

$obj -> xmlw -> endElement(); //</section>
```

Making it work together

OpenSips Workshop @Cluecon
Chicago 2015

Headers for multi-tenant

OpenSips Workshop @Cluecon
Chicago 2015