



How to deploy WebRTC, Class5, Class4 and SBC Services within minutes

Andreas Granig

`<agranig@sipwise.com>`

Andrew Pogrebennyk

`<apogrebennyk@sipwise.com>`

Victor Seva

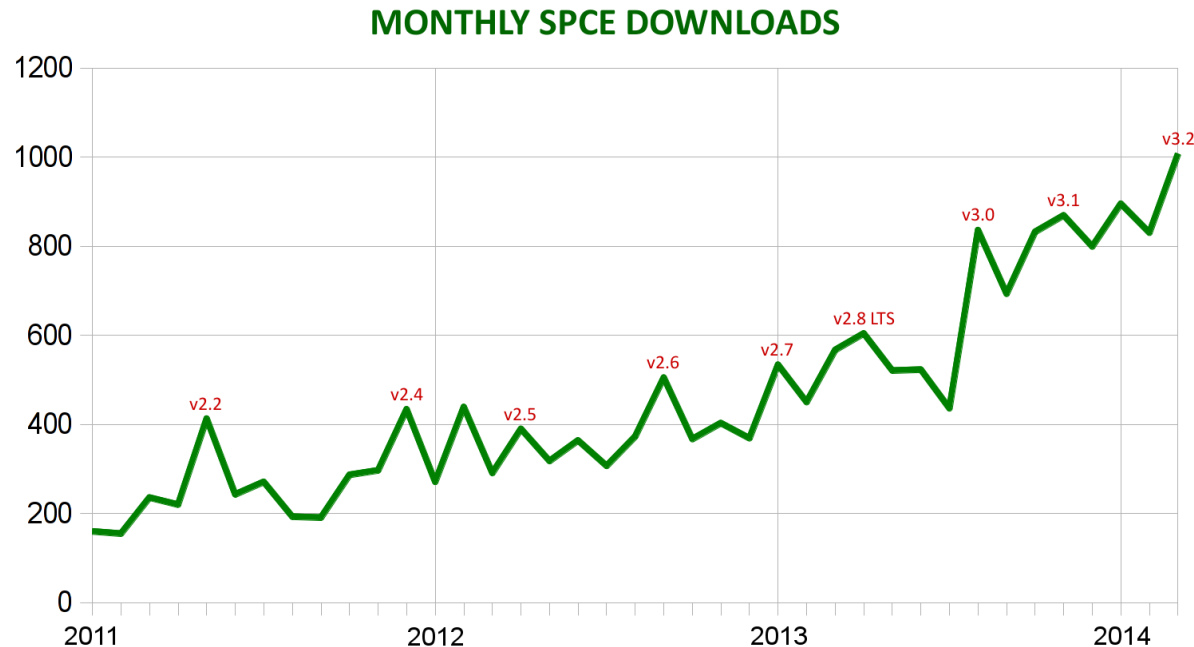
`<vseva@sipwise.com>`

What's the sip:provider CE?

- A turn-key appliance for real-time communication (voice, video, presence, IM)
- using SIP and XMPP
- for carrier environments with 50k+ subscribers and 2k+ parallel calls
- based on Kamailio, Prosody and Sipwise projects

Some Statistics

- First public release in December 2010
- 11 releases so far (latest is mr3.2.1)
- 18k+ downloads total, ~1k per month



Agenda

- Set up your VMs
- Hook up SIP, WebRTC and XMPP Clients
- Use Rewrite Rules, Peering and Billing
- Know the Architecture
- Manage Configuration Files
- Tweaks for different Use Cases

Set up your VMs



- <http://www.vagrantup.com>
- <http://www.virtualbox.org>
- https://s3-eu-west-1.amazonaws.com/kamailioworld2014/sip_provider_CE_mr3.2.1_vagrant.box

```
$ vagrant init spce sip_provider_CE_mr3.2.1_vagrant.box  
$ vagrant up
```

Accessing your SPCE

- `https://your-ip:1443`
`administrator/administrator`

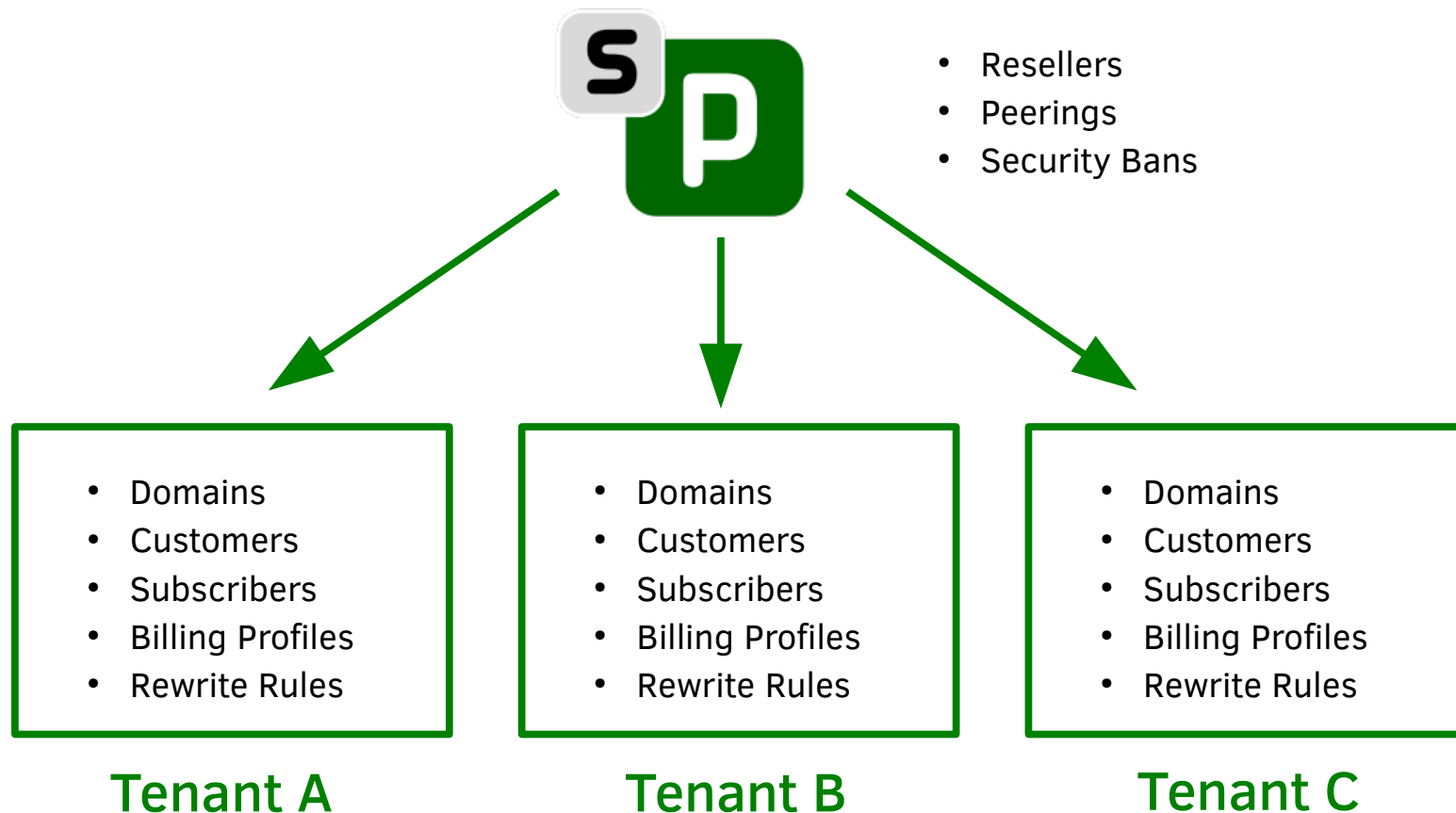


- `vagrant ssh`
`sudo -s`
`root/sipwise`



Creating a Reseller

- (Almost) Everything is Multi-Tenant!

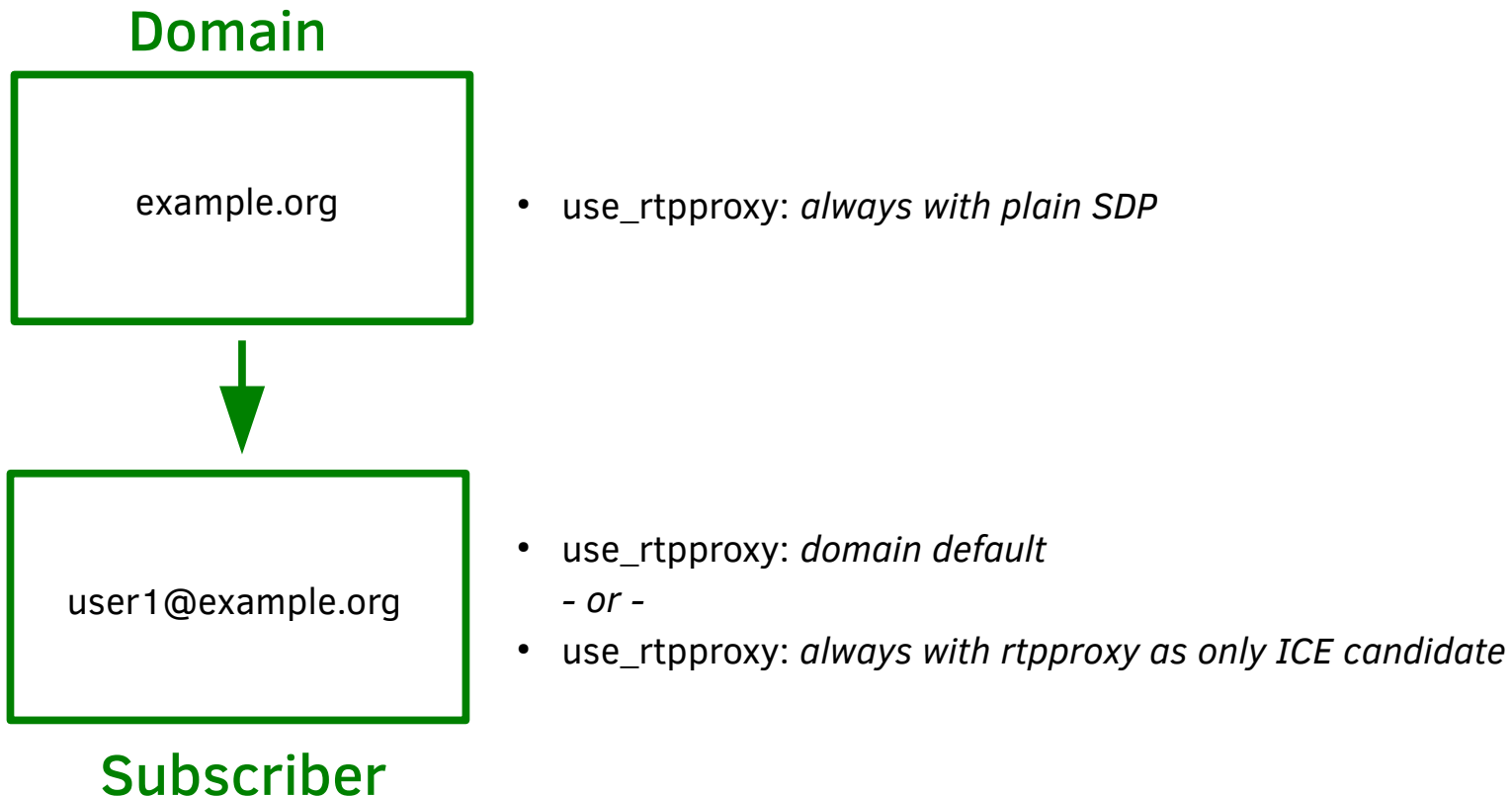


Creating a Reseller

- *default* Reseller, or your own:
- Settings → Resellers → Create with Default
- Adapt Default Values
 - Base Information
 - Contact Email
 - Admin Logins
- Create Billing Profile for Customers

Creating a Domain

- Domain-Preferences are defaults for Subscriber-Preferences

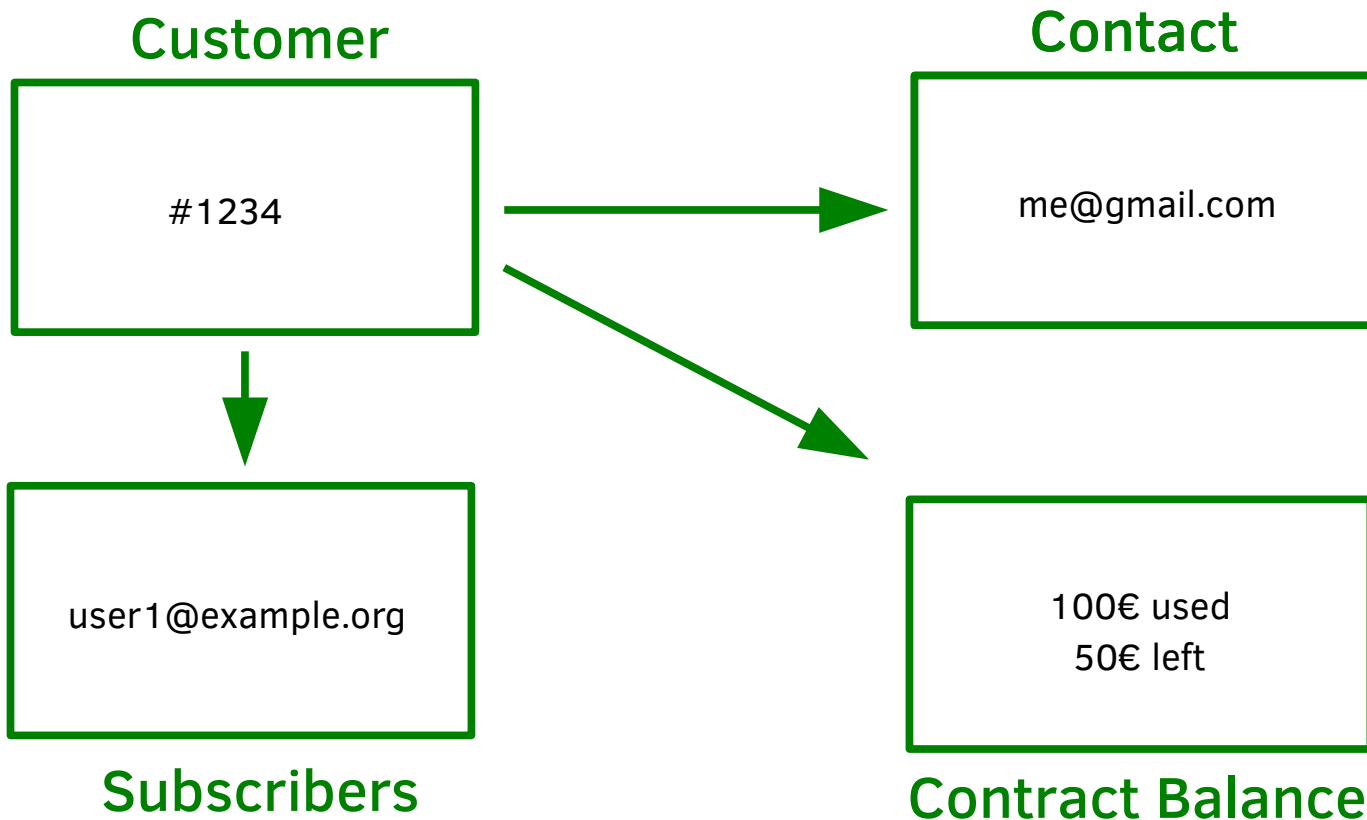


Creating a Domain

- *Your IP* as Domain, or your own:
- Settings → Domains → Create Domain
- Default Domain-Preferences are fine (for now)

Creating a Customer

- Customers are Billing Containers

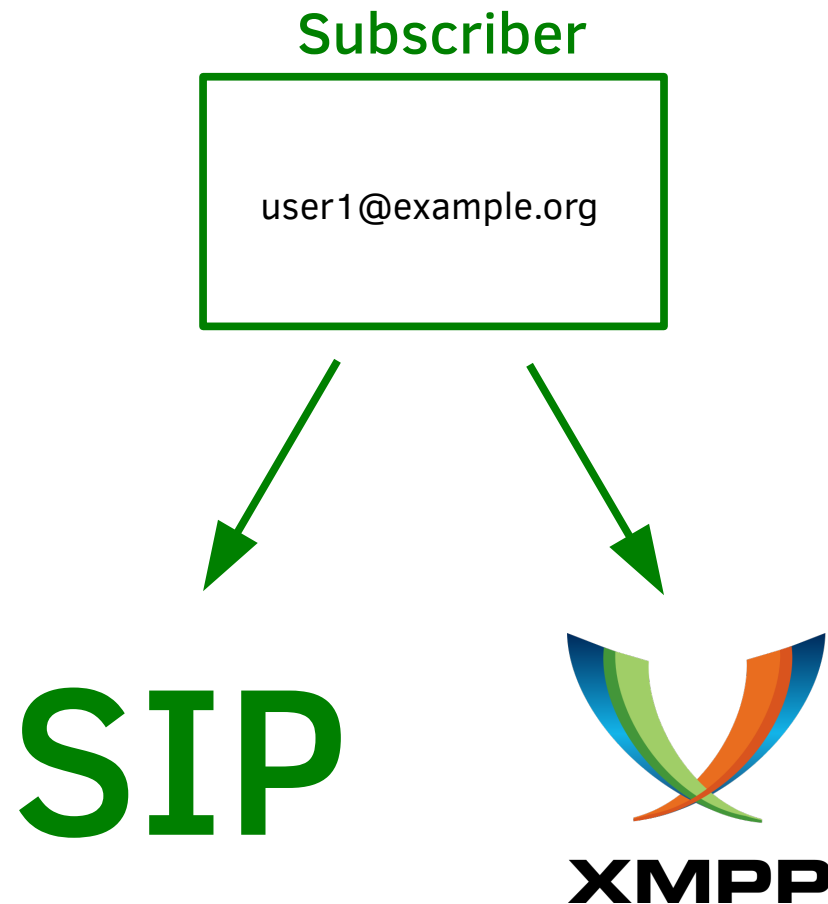


Creating a Customer

- Use an existing one, or your own:
- Settings → Customers → Create Customer
 - Select/Create a Contact
 - Select/Create a Billing Profile

Creating a Subscriber

- Now we're there!



Creating a Subscriber

- Use an existing one, or your own:
- Settings → Customers → *Your Customer*
→ Details → Create Subscriber
- or
- Settings → Subscribers → Create Subscriber

Connecting your Subscribers

- SIP and XMPP work out of the box
- SIP/TLS needs to be enabled
 - `vim /etc/ngcp-config/config.yml`
 - `kamailio → lb → tls → enable: 'yes'`
 - `ngcpcfg apply`



What about WebRTC?

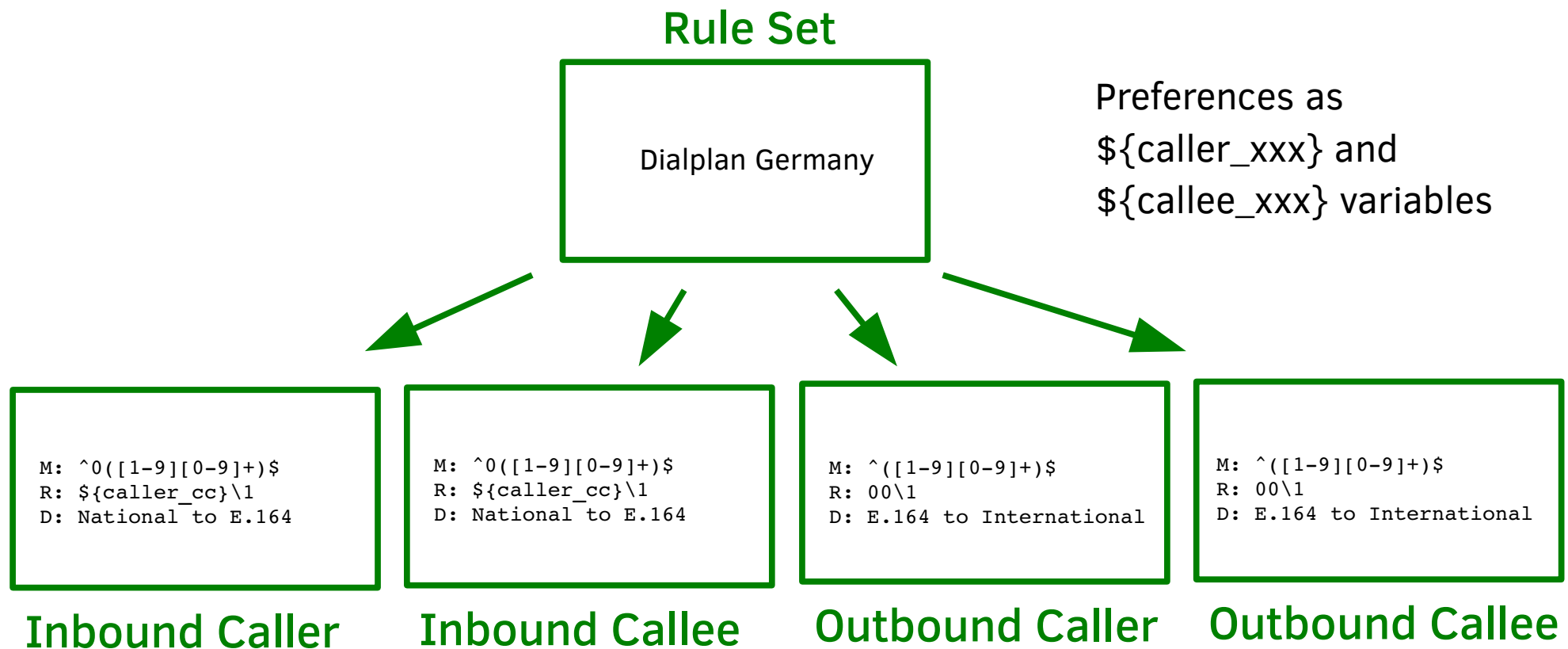
`ws://your-ip:5060/ws, wss://your-ip:5061/ws, wss://your-ip:1443/wss/sip/`

- Needs Preference Tweaking as of mr3.2.x
- Subscribers → Details → Preferences
 - NAT and Media Flow Control
 - `use_rtpproxy`: Never, Always with rtpproxy as additional/only ICE candidate
 - `srtplib_transcoding`: Transparent, Prefer SRTP
 - `rtcp_feedback`: Transparent, Prefer AVPF
- Depends on your Use Case (SIP ↔ WebRTC Bridging)



Rewrite Rules

- Your Dial-Plans in Perl Regex



Creating Rewrite Rules

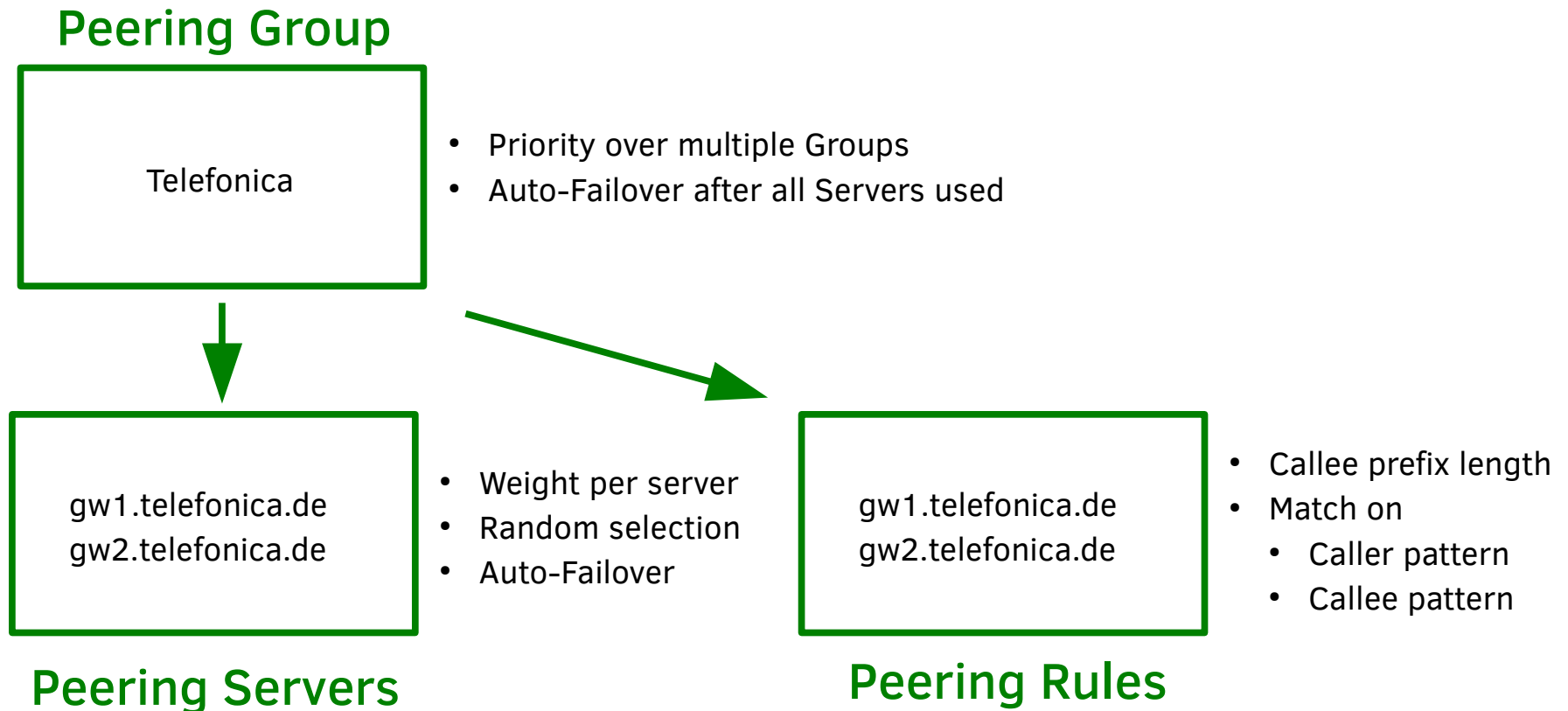
- Settings → Rewrite Rules
- Processing stops on first match (order matters)
- Assign to Subscribers, Domains and Peer
 - Preferences → `rewrite_rule_set`

Peerings

- Dynamic peering via ENUM
 - config.yml: kamailio → proxy → use_enum
- Dynamic peering via Foreign Domains
 - Dom-Preference: allow_out_foreign_domain
 - Dom-Preference: unauth_inbound_calls
- Static peering via Peering Groups
 - Everything not local goes to peer
 - Force inbound/outbound to peer via Preferences

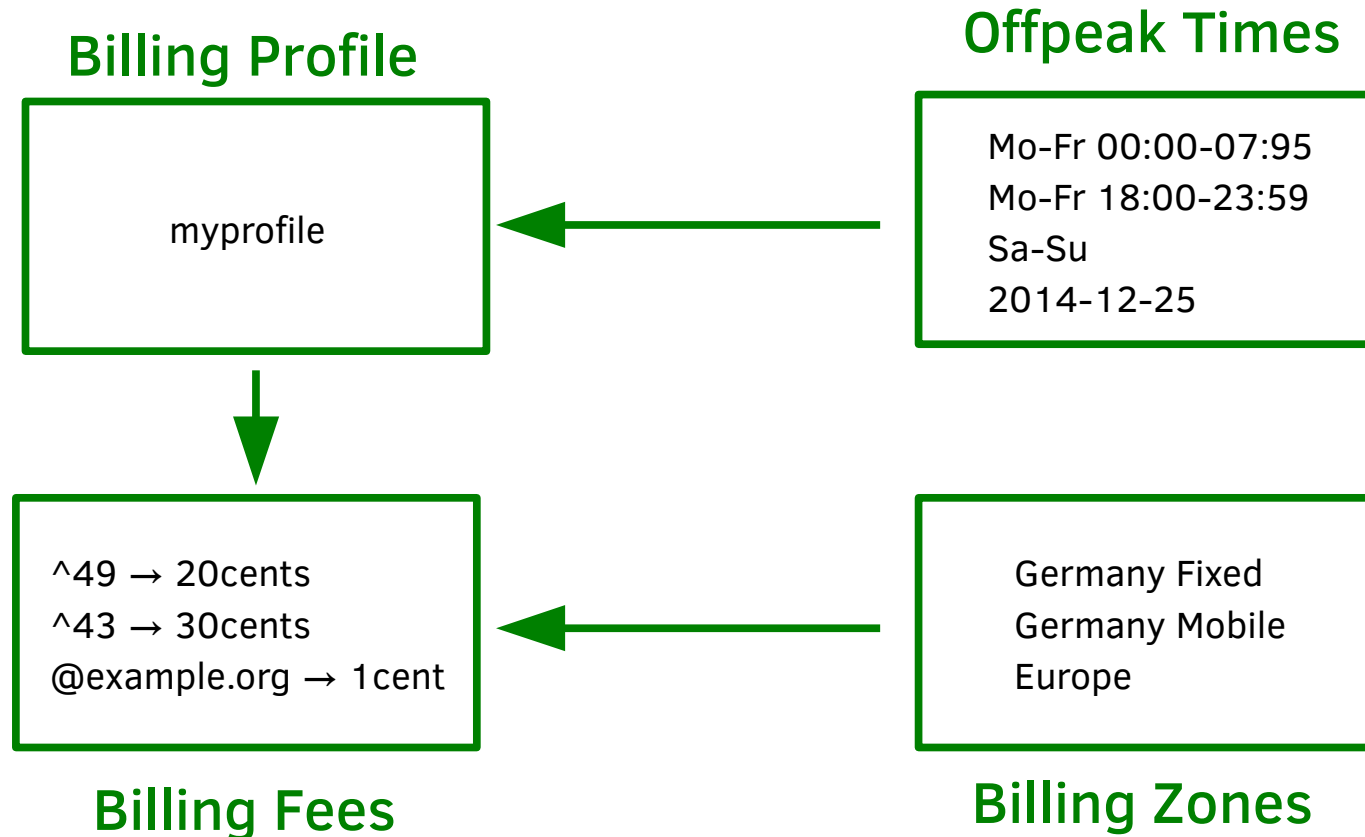
Static Peerings

- Settings → Peerings → Create Peering Group



Billing Profiles

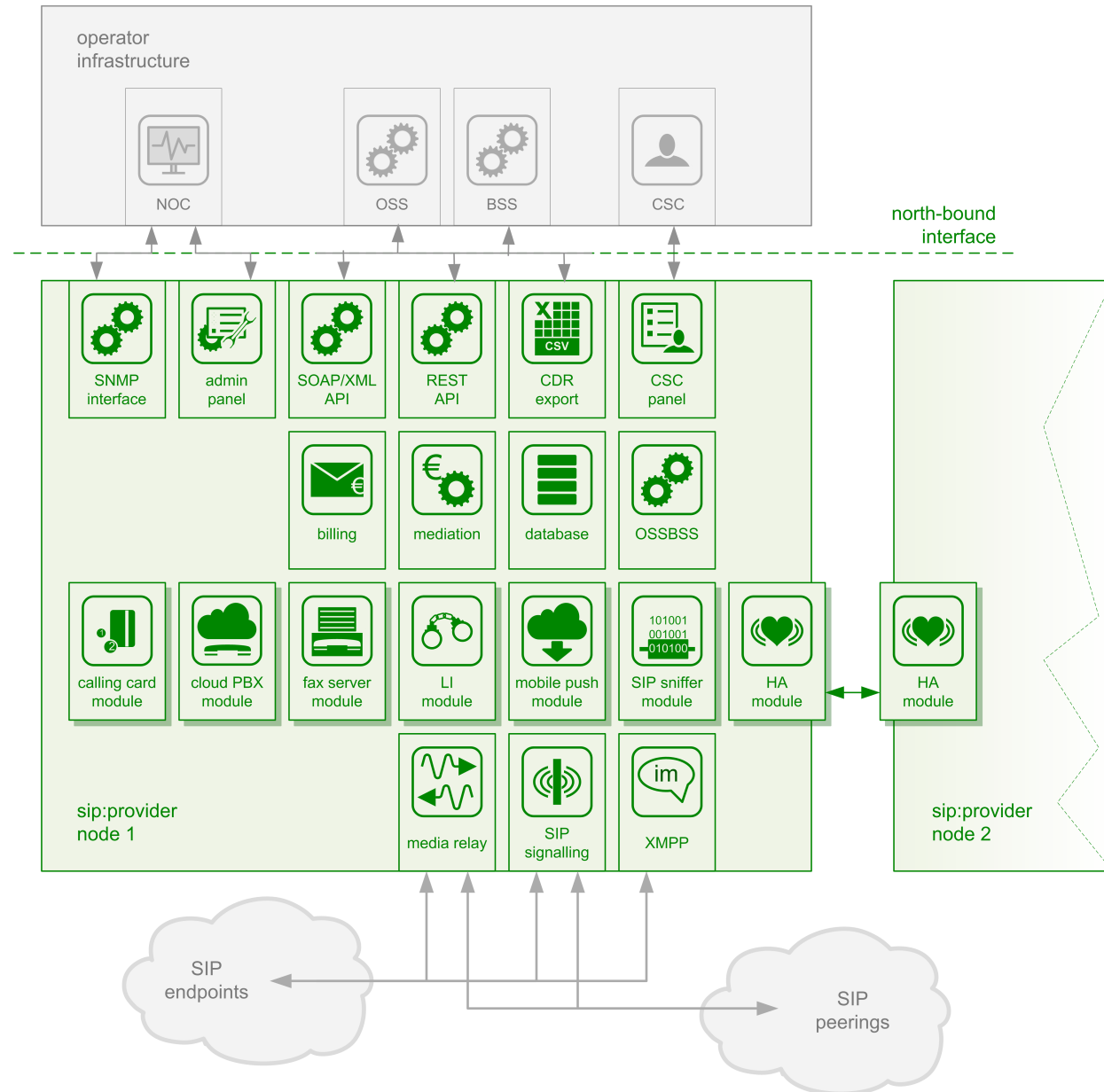
- Settings → Billing → Create Billing Profile



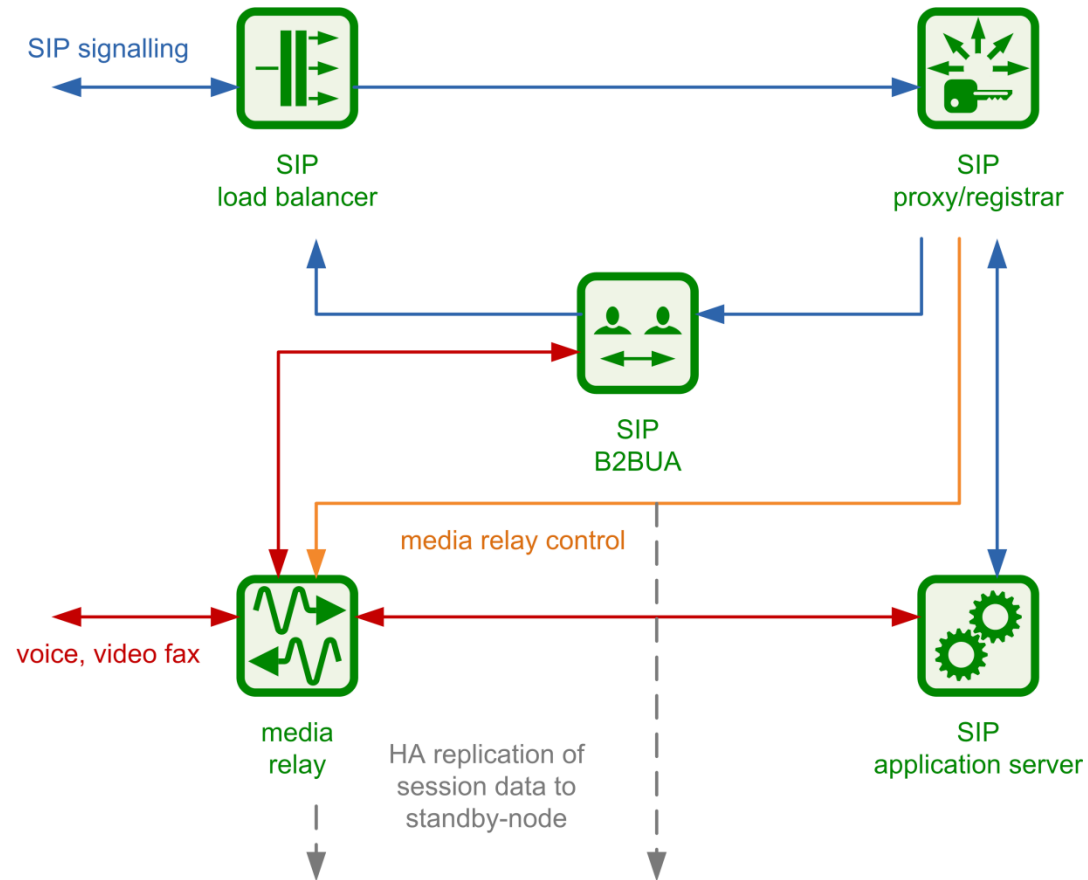
That's it for the operational part

Easy, eh?

sip:provider Architecture



SIP Components



Configuration Framework

- Templates!
- Controlled by:
 - `/etc/ngcp-config/config.yml`
 - `/etc/ngcp-config/constants.yml`
 - `/etc/ngcp-config/network.yml`
- Template sources:
 - `/etc/ngcp-config/templates/...`
- Backed by Git (but you don't use it directly)

Changing Configs

- Use customtt-Files!

```
$ cp \  
    /etc/ngcp-config/templates/etc/kamailio/lb/kamailio.cfg.tt2 \  
    /etc/ngcp-config/templates/etc/kamailio/lb/kamailio.cfg.customtt.tt2
```

- Apply changes

```
$ ngcpcfg apply
```

- Commits your changes, generates configs and restarts affected services (neat, eh?)
- Automatically tracks everything you drop into `/etc/ngcp-config/templates/`

Your Use Cases?

- If we wanted to do \$this...
- then you have to do \$that

Your turn!

More Questions?

@sipwise

<http://lists.sipwise.com/listinfo/spce-user>

agranig@sipwise.com