AstriDevCon 2016

Scaling Asterisk Horizontally

Presented by
Nicolas Bouliane

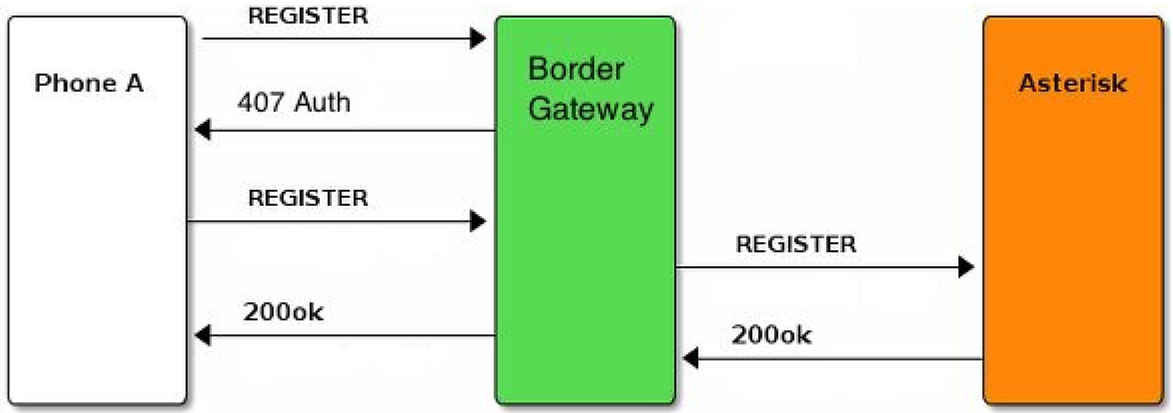
JIVE

Jive ?

- Founded in 2006.
- provides enterprise-grade Hosted VoIP and Unified Communications to businesses and institutions.
- ~500 employees, >60 developers.

Old Asterisk Setup

- Virtual Machine based Asterisk 1.8.
- Cassandra as SIP Registration backend.
- 1:1 Relationship between phone and Asterisk instance.

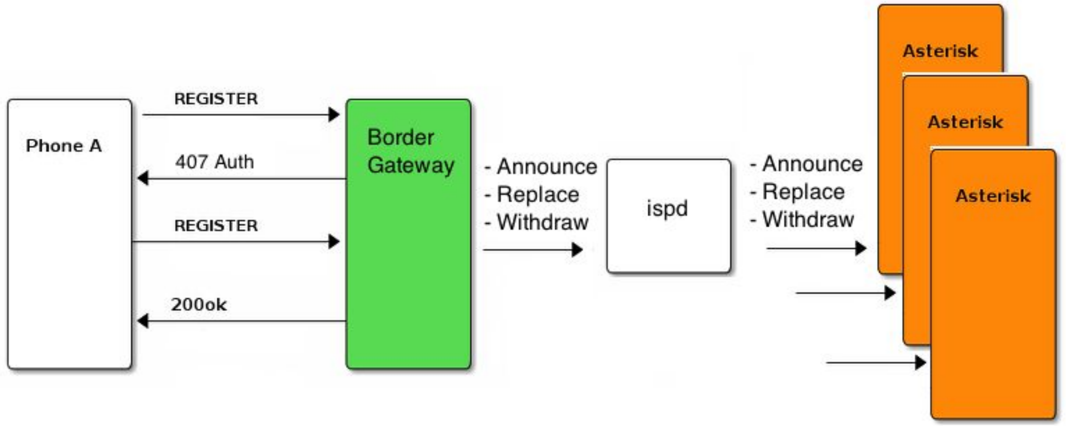


Problem with old setup

- Hard to move accounts from one Asterisk to another.
- Phones has to re-register, hence adding delays.
- Slowly migrating from one Asterisk to another cause problems when SIP REGISTER timeout.
- Not as dynamic as we would like.

Scaling Asterisk

- Moving to Dockerized Asterisk 11.
- Event based SIP REGISTER.
- 1:1 relationship with BGW only.

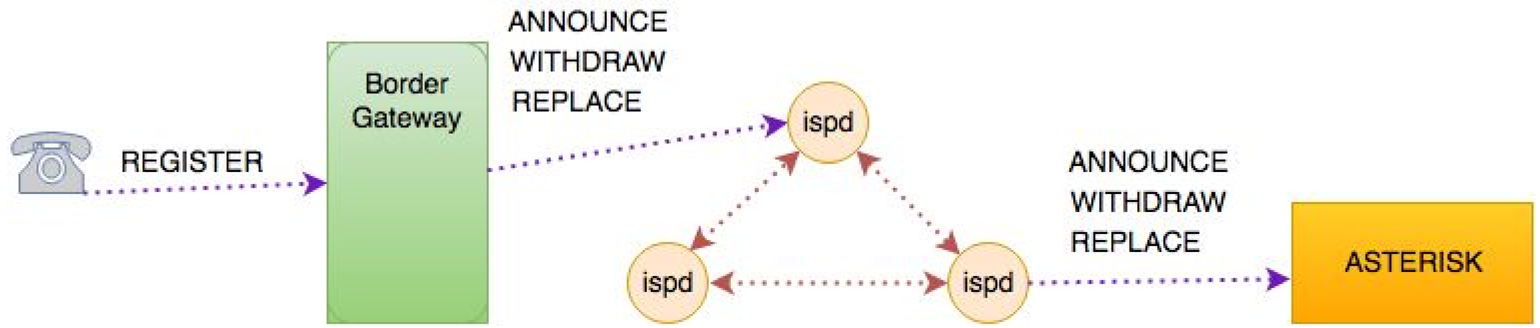


Improvement

- Easier to spawn new Instance of Asterisk.
- Each Asterisk have a realtime view of all the SIP REGISTER.
- Easier to move clients from one Asterisk to another one.
- ...Another step towards a dynamic pool of Asterisk ressource.

How does ispd works ?

- Ispd is just a data reflector, as in bgp where you can have route reflectors.
- Isdp can create as many table (or channel) as needed. Here we could use a table named SIP_REGISTRATION.



How does ispd works ?

- Also useful to dynamically discover all the available Asterisk instances.
- Use a TCP connection, if the connection dies, ispd will remove the record from the table, and everyone listening to the table, will receive the ‘withdraw’ (it’s a reflector).

Event based registration

- Custom module (realtime sipregs lookup)
  - Receive ANNOUNCE, WITHDRAW, REPLACE from ispd, keep info in memory.
  - call chan_sip custom function
    - Update the device state
    - Update peer object
  - (Migration) Expire any timer, destroy old backend association.

Hacking our way directly into chan_sip

- Our custom real time sipregs module call our custom chan_sip function to update the peer

```
static void update_cachedpeer(const char *aor, int state)
{
        const struct ast_channel_tech *chan_tech;
        if ((chan_tech = ast_get_channel_tech("SIP")) == NULL)
                ast_log(LOG_WARNING, "he SIP channel tech is unavailable\n");
        else
                chan_tech->update_cachedpeer(aor, sipreg_lookup_aor(aor), state);
}
```

- ## Search peer object and clear again old realtime DB

```
static void sip_update_cachedpeer(const char *peername, struct ast_variable *varregs, int state)
{
	struct sip_peer *peer, tmp;
	char *tablename;
	int realtimeregs;

	ast_copy_string(tmp.name, peername, sizeof(tmp.name));

	if ((peer = ao2_t_find(peers, &tmp, OBJ_POINTER, "ao2_find, update cached peer")) == NULL) {
		realtimeregs = ast_check_realtime("sipregs");
		tablename = realtimeregs ? "sipregs" : "sippeers";
		ast_update_realtime(tablename, "name", peername, "fullcontact", "", "ipaddr", "", "port",
			"", "regseconds", "0", "regserver", "", "useragent", "", "lastms", "0", SENTINEL);
		return;
	}
```

- Expire every timer if needed, needed during transition
- Asked for a reset ? just clear the ip address

```
if (peer->expire > 0) {
        AST_SCHED_DEL_UNREF(sched, peer->expire,
                unref_peer(peer, "removing register expire ref"));
        destroy_association(peer);     /* remove registration data from storage */
}

if (state == 0) {

        ao2_lock(peer);
        ast_sockaddr_setnull(&peer->addr);
        ao2_unlock(peer);
```

- ## Update peer object with data from varregs

```
} else if (varregs != NULL) {
  …
        for (v = varregs; v; v = v->next) {
                if (!strcasecmp(v->name, "fullcontact")) {
                ast_string_field_set(peer, fullcontact, v->value);
                } else if (!strcasecmp(v->name, "username")) {
                ast_string_field_set(peer, username, v->value);
                } else if (!strcasecmp(v->name, "useragent")) {
                ast_string_field_set(peer, useragent, v->value);
                } else if (!strcasecmp(v->name, "ipaddr")) {
                ast_sockaddr_parse(&peer->addr, v->value, PARSE_PORT_FORBID);
                ast_sockaddr_set_port(&peer->addr, 5060);
        …
```

What next ?

- Looking into pjsip integration into Asterisk if we get better control.
- Going further in treating Asterisk as a pool of dynamic resources.

Thank you for your work on Asterisk !

Questions ?