Asterisk Cluster with MySQL Replication

JR Richardson
Engineering for the Masses
Hubguru@gmail.com

Presentation Overview

- Reasons to cluster Asterisk
  - Load distribution
  - Scalability
- This presentation focuses on
  - Asterisk Realtime Architecture
  - MySQL Replication

Asterisk Realtime Architecture (ARA)

- ARA allows Asterisk to store configs and dial plan in a database
- Two modes of operation, Static and Realtime
- Static allows config file storage, sip.conf, etc…
- Realtime dynamically loads and updates objects
  - SIP Peers
  - Dial Plan
  - Voicemail boxes

ARA Benefits

- Share SIP peer authentication across cluster
  - SIP peer may register with registration server 1 or 2, the authentication information is pulled from the database and is known to both servers
- Share dial plan across cluster of Asterisk servers
  - dial plan is read from the database so regardless of which server a SIP peer registers with, the same call pattern is used

Realtime Database

- MySQL, database engine stores the tables and records
- Database will contain 3 basic tables
  - sip, SIP peer authentication info
  - extensions, dial plan
  - voicemail, voicemail boxes with options

MySQL Replication

- Master database is a central point for database manipulation, adds, moves, changes
- Changes are broadcast all slave databases
- All the Asterisk servers in the cluster have the same database information
- Replication occurs sub-second
- Replication works through a one-way, log shipping, asynchronous mechanism
- One server is designated as a Master and one or more servers are designated as Slaves

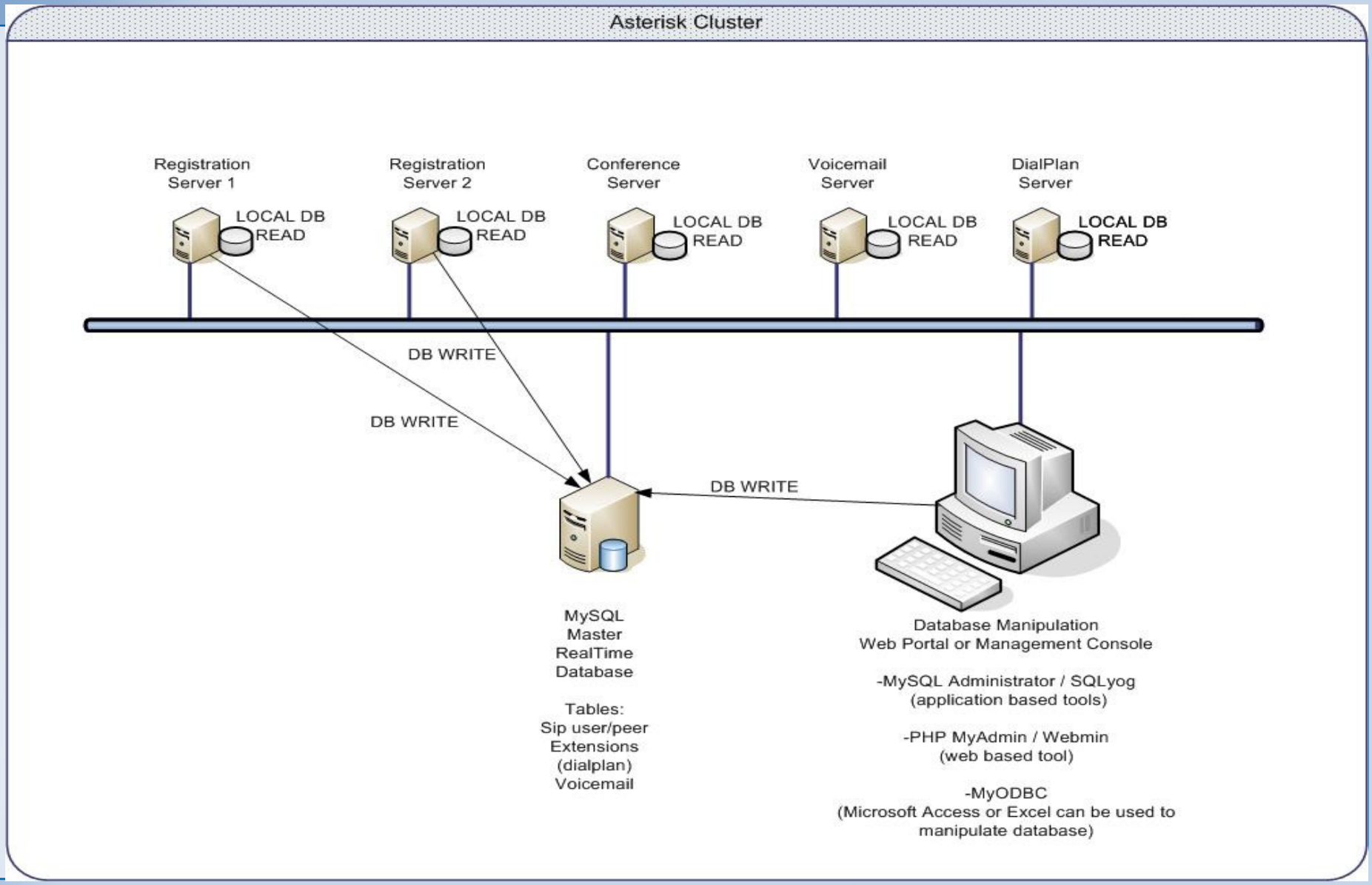
MySQL Replication (cont)

- As the Master receives updates, those changes are propagated to the slaves via a log file then executed
- Slaves have the same data as the Master
- No numerical limit to number of slaves
- MySQL recommends no more than 20 slaves
- Multiple masters used to feed more slaves

Why Use Replication?

- Fundamental reason is performance
- Asterisk must read every extension from the database
- Asterisk write activity is low compared to read activity
- Reads are performed for every step in call flow
- Asterisk reads locally on the host server
- Decreases network load and speeds data access time
- Added benefit, all service nodes have same data
- Master database failure, service nodes still operate
- Service node failure doesn't effect overall operation

Asterisk Cluster Pictorial



ARA and MySQL Replication Setup

- The sequence of events will be as follows:
- Install Asterisk
- Install MySQL
- Install Asterisk addons with patch 5881 applied
- Setup MySQL users
- Setup MySQL database and tables
- Configure MySQL replication master
- Configure MySQL replication slaves
- Configure res_mysql.conf
- Configure extconfig.conf

Install Asterisk

- ARA officially implemented in Asterisk 1.2
- Asterisk stable release can be downloaded from the Digium FTP site or testing release downloaded from SVN
- Installation guides can be found at:
  http://www.voip-info.org/wiki-Asterisk+installation+tips

Install MySQL

- Most Linux distributions have pre-built packages that can be installed from a package manager
- Source can be downloaded from MySQL and compiled locally

  http://dev.mysql.com/downloads/mysql/5.0.html
- **NOTE: MySQL replication requires MySQL 3.23 and above.

Install * Addons with Patch 5881

- Res_MySQL module enables ARA to communicate with MySQL
- Module is in the asterisk_addon package download from Digium ftp://ftp.digium.com
- Stable package only allows one read/write database definition
- You will need a patch downloaded from http://bugs.digium.com/view.php?id=5881
- Patch allows module to read and write to a separate servers
- **NOTE: The patch is committed to SVN, will be in stable * 1.6
- After applying patch, install addons, make, make install
- Copy the res_config.conf example file to /etc/asterisk
- Restart Asterisk so the module will be loaded

Setup MySQL Users

- Log into MySQL: # mysql -u root
- Switch to the mysql database: mysql> use mysql
- View existing users: mysql> select * from user\G;
- Set root password:
  - mysql> SET PASSWORD FOR root@localhost=PASSWORD('new_password');
- Create a regular user:
  - INSERT INTO user
  - values('%','asteriskdb',password('asterisk123'),
  - 'Y','Y','Y','Y','N','N','N','N','N','N','N','N','N');
- **Note: This is for MySQL 3.23, more fields are needed for newer versions, on-line MySQL documentation manual for newer version syntax: http://dev.mysql.com/doc/#refman
- Create a slave:
  - insert into user
  - values('%','asteriskslave',password('asteriskslave123'),
  - 'Y','N','N','N','N','N','Y','N','Y','Y','Y','N','N','N');
- Enable new users: mysql> flush privileges;

Setup Database and Tables

- Create database on master and slave servers
- Only create tables on the master server
- MySQL command will pull over tables from master
- Log into MySQL, create the database:
  - **mysql> CREATE database asteriskdb;**
- Display the databases:
  - **mysql> show databases;**
- Select the new database to add the tables:
  - **mysql> use asteriskdb**

Extensions Table

```
CREATE TABLE `extensions` (
 `id` int(11) NOT NULL auto_increment,
 `context` varchar(20) NOT NULL default '',
 `exten` varchar(20) NOT NULL default '',
 `priority` tinyint(4) NOT NULL default '0',
 `app` varchar(20) NOT NULL default '',
 `appdata` varchar(128) NOT NULL default '',
 `accountcode` varchar(20) default NULL,
 `notes` varchar(255) default NULL,
 PRIMARY KEY  (`context`,`exten`,`priority`),
 KEY `id` (`id`)
) TYPE=MyISAM;
```

Voicemail Table

```
CREATE TABLE `voicemail` (
 `uniqueid` int(11) NOT NULL auto_increment,
 `customer_id` varchar(11) NOT NULL default '0',
 `context` varchar(50) NOT NULL default '',
 `mailbox` varchar(11) NOT NULL default '0',
 `password` varchar(5) NOT NULL default '0',
 `fullname` varchar(150) NOT NULL default '',
 `email` varchar(50) NOT NULL default '',
 `pager` varchar(50) NOT NULL default '',
 `tz` varchar(10) NOT NULL default 'central',
 `attach` varchar(4) NOT NULL default 'yes',
 `saycid` varchar(4) NOT NULL default 'yes',
 `dialout` varchar(10) NOT NULL default '',
 `callback` varchar(10) NOT NULL default '',
 `review` varchar(4) NOT NULL default 'no',
 `operator` varchar(4) NOT NULL default 'no',
 `envelope` varchar(4) NOT NULL default 'no',
 `sayduration` varchar(4) NOT NULL default 'no',
 `saydurationm` tinyint(4) NOT NULL default '1',
 `sendvoicemail` varchar(4) NOT NULL default 'no',
 `delete` varchar(4) NOT NULL default 'no',
 `nextaftercmd` varchar(4) NOT NULL default 'yes',
 `forcename` varchar(4) NOT NULL default 'no',
 `forcegreetings` varchar(4) NOT NULL default 'no',
 `hidefromdir` varchar(4) NOT NULL default 'yes',
 PRIMARY KEY  (`uniqueid`),
 KEY `mailbox_context` (`mailbox`,`context`)
) TYPE=MyISAM;
**NOTE: `uniqueid` field name must remain for password updates to work properly
```

SIP Table

```
CREATE TABLE `sip` (
 `id` int(11) NOT NULL auto_increment,
 `name` varchar(80) NOT NULL default '',
 `accountcode` varchar(20) default NULL,
 `amaflags` varchar(13) default NULL,
 `callgroup` varchar(10) default NULL,
 `callerid` varchar(80) default NULL,
 `canreinvite` char(3) default 'yes',
 `context` varchar(80) default NULL,
 `defaultip` varchar(15) default NULL,
 `dtmfmode` varchar(7) default NULL,
 `fromuser` varchar(80) default NULL,
 `fromdomain` varchar(80) default NULL,
 `host` varchar(31) NOT NULL default '',
 `insecure` varchar(4) default NULL,
 `language` char(2) default NULL,
 `mailbox` varchar(50) default NULL,
 `md5secret` varchar(80) default NULL,
 `nat` varchar(5) NOT NULL default 'no',
 `deny` varchar(95) default NULL,
 `permit` varchar(95) default NULL,
 `mask` varchar(95) default NULL,
```

SIP Table (cont)

```
`pickupgroup` varchar(10) default NULL,
`port` varchar(5) NOT NULL default '',
`qualify` char(3) default NULL,
`restrictcid` char(1) default NULL,
`rtptimeout` char(3) default NULL,
`rtpholdtimeout` char(3) default NULL,
`secret` varchar(80) default NULL,
`type` varchar(6) NOT NULL default 'friend',
`username` varchar(80) NOT NULL default '',
`disallow` varchar(100) default 'all',
`allow` varchar(100) default 'gsm;ulaw;alaw',
`musiconhold` varchar(100) default NULL,
`regseconds` int(11) NOT NULL default '0',
`ipaddr` varchar(15) NOT NULL default '',
`regexten` varchar(80) NOT NULL default '',
`cancallforward` char(3) default 'yes',
`setvar` varchar(100) NOT NULL default '',
`fullcontact` varchar(80) default NULL,
PRIMARY KEY  (`id`),
UNIQUE KEY `name` (`name`),
KEY `name_2` (`name`)
) TYPE=MyISAM;
```

Setup Replication on Master

- Edit the /etc/mysql/my.cnf file with these parameters:
- Under the **[mysqld]** profile:
- Comment out
  - **#skip-networking**
- Enable replication, add these lines:
  - **server-id=1**
  - **log-bin=/var/log/mysql/mysql-bin.log**
  - **binlog-do-db=asteriskdb**
- The master server should always have a server-id=1
- log-bin sets location of replication bin files
- bin-log-db sets database to replicate
- Restart MySQL so the new changes take affect:
  - **# /etc/init.d/mysql restart**
- Access database via MySQL database administrator utility
  - MySQL Administrator, SQLyog, EMS SQL Manager for MySQL, Webmin, phpMyAdmin, many other programs

Setup Replication on Slaves

- Configure the /etc/mysql/my.cnf file:
- Under the **[mysqld]** profile:
  - **server-id=2**
  - **master-host=10.10.10.10**
  - **master-user=askeriskslave**
  - **master-password=askeriskslave123**
  - **master-connect-retry=60**
  - **replicate-do-db=asteriskdb**
- The server-id has to be a unique number for each server.
- Restart MySQL so the changes take affect: **# /etc/init.d/mysql restart**
- Log into MySQL on the server: **# mysql -u asteriskdb -p (password=asteriskdb123)**
- **mysql> use asteriskdb**
- **mysql> load table sip from master;**
- **mysql> load table extensions from master;**
- **mysql> load table voicemail from master;**
- **mysql> show tables;**
- **mysql> show master status\G;**
- **mysql> show slave status\G;**
- http://www.voip-info.org/wiki-Asterisk+RealTime

Setup res_mysql.conf

- Configure /etc/asterisk/res_mysql.conf
  - **[general]**
  - **dbname=asteriskdb**
  - **dbuser=asteriskdb**
  - **dbpass=asteriskdb123**
  - **dbport = 3306**
  - **dbsock = /tmp/mysql.sock**
  - **[read]**
  - **dbhost = 127.0.0.1**
  - **[write]**
  - **dbhost = 10.10.10.1**
- General section, info that spans both read and write databases
- Read section points to localhost database
  - **NOTE: dbhost=localhost does not work, must use 127.0.0.1
- Write section points to master database

Setup extconfig.conf

- Edit /etc/asterisk/extconfig.conf
  - **[settings]**
  - **extensions => mysql,asteriskdb,extensions**
  - **sipusers => mysql,asteriskdb,sip**
  - **sippeers => mysql,asteriskdb,sip**
  - **voicemail => mysql,astriskdb,voicemail**
- **extensions** is the mapping for the dialplan
- **sipusers** and **sippeers** is the mapping for SIP devices
- **voicemail** is the mapping for the voicemail system
- **mysql** is the driver
- **asteriskdb** is the database name
- Last parameter is the table name

ARA and MySQL Replication is Setup

- In extensions.conf, you must have a pseudo [context] with a switch statement to access the database:
  - **switch => Realtime/[context]@[family]**
  - **[default]**
  - **switch => Realtime/@**
- If context is left off, then it defaults to context name where the switch statement is, in this case [default]
- If family is left off, it defaults to [extensions]
- Above switch statement is the same as:
  - **switch => Realtime/default@extensions**

Operational Examples

- SIP device registers with Asterisk
- Asterisk gathers IP Address and Port number
- Asterisk writes to master database, ‘sip’ table
- The fields populated are ‘ipaddr’ and ‘port’
- This data propagates to all slaves
- Each server in cluster can contact SIP device
- Dial plan uses RealTime application to contact SIP device directly

Realtime Application

lab1*CLI> show application RealTime
  -= Info about application 'RealTime' =-

[Synopsis]
Realtime Data Lookup

[Description]
Use the RealTime config handler system to read data into channel variables.
RealTime(<family>|<colmatch>|<value>[|<prefix>])

All unique column names will be set as channel variables with optional prefix to the name.
e.g. prefix of 'var_' would make the column 'name' become the variable ${var_name}

Contact SIP Device Using Realtime

```
[lookupmysql]
exten => _X.,1,RealTime(sippeers|name|${EXTEN}|DN_)
exten => _X.,2,GotoIf($["${DN_ipaddr}" = ""]?${EXTEN},105:${EXTEN},3)
exten => _X.,3,Set(directdial=${DN_extenname}@${DN_ipaddr}:${DN_port})
exten => _X.,4,Dial(SIP/${directdial},15,rj)
exten => _X.,5,Macro(sendtovm,${EXTEN})
exten => _X.,6,Hangup
exten => _X.,105,Macro(sendtovm,${EXTEN})
exten => _X.,106,Hangup
```

RealTimeUpdate Application

lab1*CLI> show application RealTimeUpdate
  -= Info about application 'RealTimeUpdate' =-

[Synopsis]
Realtime Data Rewrite

[Description]
Use the RealTime config handler system to update a value
RealTimeUpdate(<family>|<colmatch>|<value>|<newcol>|<newval>)

The column <newcol> in 'family' matching column <colmatch>=<value>
    will be updated to <newval>

Update Database from Dial Plan

- Office closed “on”
- Office closed “off”

context	exten	priority	app	appdata
company-a	*20	1	RealTimeUpdate	extensions|app|company-a|appdata|on
company-a	*20	2	Hangup	
company-a	*21	1	RealTimeUpdate	extensions|app|company-a|appdata|off
company-a	*21	2	Hangup	

Use Updated Data Example

- Auto Attendent Check if Office is Closed “on” or “off”

Context	exten	priority	App	appdata
company-a	s	1	RealTime	extensions|app|company-a|OC_
company-a	s	2	GotoIf	$["${OC_appdata}" = "on"]?company-a_closed|s|1
company-a	s	3	Answer	
company-a	s	4	Wait	1
company-a	s	5	Set	TIMEOUT(digit)=2
company-a	s	6	Set	TIMEOUT(response)=2
company-a	s	7	BackGround	company-a
company-a	s	8	WaitExten	

RealTime Database Syntax

- ARA can not parse ‘,’ (commas)
- ‘,’ in database used for end-of-field
- ‘appdata’ fields must contain ‘|’ (pipes)
- In extensions.conf:

[default]
exten => 1001,1,Dial(SIP/1001,20,tr)

- In Database:

context	exten	priority	app	appdata
Default	1001	1	Dial	SIP/1001|20|tr

MySQL Replication versus Clustering

- Implementation expense is higher with clustering
  - Need Data nodes, SQL node and Management node
  - For data integrity, multiple data nodes are needed
  - MySQL Clustering prior to 5.1 requires all data reside in RAM
  - Setup and operation of cluster is more difficult than replication
- Cluster network usage much higher than replication
  - ARA constantly reading data from the database
  - SQL reads from Asterisk servers are across network
  - SQL node queries retrieved across network from Data nodes
  - It's possible to setup Asterisk servers as SQL nodes, but reads still traverse network from SQL node to Data nodes

Authenticate LD Calls via Database

- Asterisk Authenticate application is a great tool
- Prohibit long distance dialing unless PIN code entered
- Application reads PIN codes from file on local PBX
- Updates CDR(accountcode) field with PIN Code

- In a cluster arrangement, with dynamic SIP peers authentication becomes a challenge
- Maintaining up to date files with PIN codes on each registration server is not a good solution
- Read PIN codes from database using ARA
  - Create customer interface to pin database
  - PIN codes are known across the cluster
  - No editing flat files on each server
  - PIN put in CDR(userfield) not accountcode
  - It's just cool

Create PIN Table on Master DB

- Log into MySQL master database, select asteriskdb
  - **# mysql -u root -p** (enter root password

mysql> use asteriskdb

```
CREATE TABLE `pins` (
 `id` int(11) NOT NULL auto_increment,
 `company` varchar(20) NOT NULL default '',
 `pin` varchar(10) NOT NULL default '',
 `active` varchar(5) NOT NULL default 'no',
 `accountcode` varchar(20) NOT NULL default '',
 `notes` varchar(255) default NULL,
 PRIMARY KEY  (`company`,`pin`),
 KEY `id` (`id`)
) TYPE=MyISAM;
```

Pull PIN Table Over to Slaves

- Log into slave servers, pull over new table
  # mysql -u asteriskdb -p (password asteriskdb123)
  mysql> use asteriskdb
  mysql> load table pins from master;

- Add the following to /etc/asterisk/extconfig.conf
  pins => mysql,asteriskdb,pins
- Reload extconfig from the asterisk console so the new table mapping is activated:
- **asterisk*CLI> reload extconfig**
- Add PIN codes to the master database:

company	pin	active	accountcode
Company-a	1111-1234	yes	1111

Read PIN Codes in Dial Plan

context	exten	priority	app	appdata
company-a	_1NXXNXXXXXX	1	Answer	
company-a	_1NXXNXXXXXX	2	Wait	1
company-a	_1NXXNXXXXXX	3	NoOp	${CALLERID}
company-a	_1NXXNXXXXXX	4	Set	CDR(accountcode)=1111
company-a	_1NXXNXXXXXX	5	Read	pin|agent-pass|5|noanswer|3|
company-a	_1NXXNXXXXXX	6	NoOp	${pin}
company-a	_1NXXNXXXXXX	7	RealTime	pins|pin|1111-${pin}|ok_
company-a	_1NXXNXXXXXX	8	GotoIf	$["${ok_active}" = "yes"]?9:20
company-a	_1NXXNXXXXXX	9	Set	CDR(userfield)=${pin}
company-a	_1NXXNXXXXXX	10	Playback	auth-thankyou
company-a	_1NXXNXXXXXX	11	Dial	Zap/g1|60|
company-a	_1NXXNXXXXXX	12	Hangup	
company-a	_1NXXNXXXXXX	20	Playback	privacy-incorrect
company-a	_1NXXNXXXXXX	21	Set	NUMTRIES=$[1 + ${NUMTRIES}]
company-a	_1NXXNXXXXXX	22	GotoIf	$[${NUMTRIES} >= 2]?23:5
company-a	_1NXXNXXXXXX	23	Playback	vm-goodbye
company-a	_1NXXNXXXXXX	24	Hangup	

Summary

- Asterisk is great
- MySQL is great
- Put them together
- All of your dreams can come true!
- Thank You!
  - JR Richardson
  - Engineering for the Masses
  - hubguru@gmail.com