

# Asterisk Primer

Presented at Apricot, Bali, Feb 26<sup>th</sup> 2007

Marc Blanchet

Viagénie

[Marc.Blanchet@viagenie.ca](mailto:Marc.Blanchet@viagenie.ca)

<http://www.viagenie.ca>

# Credentials



- 20+ years in IP networking and Unix, with 10 years on IPv6...
- IP engineering standards(IETF):
  - Wrote IETF drafts and RFCs.
  - Co-chaired internationalized domain names (idn) IETF wg
- Authoring:
  - Book: Migrating to IPv6, Wiley, 2006.
  - Cisco IPv6 course (co-author)
  - Tutorials on IP, security, Ipv6, etc... at many conferences, organisations
- IPv6forum: co-founder, board member. North American Ipv6 Task Force: steering group member.
- Asterisk developer, co-ported Asterisk to IPv6.
- President of Viagénie, consulting in advanced IP networking. Helping providers, enterprises, manufacturers and governments. IPv6, VoIP, Asterisk, Security, Internationalization, etc.

# Plan



- Asterisk
- First call: Hello World!
- Call between two phones
- Calling to/from the PSTN
- Small office Setup
  - voicemail, roadwarriors, directory, interactive voice response, voip provider, etc...

# Asterisk



- <http://www.asterisk.org>
- “Asterisk® is a complete IP PBX in software. It runs on a wide variety of operating systems including Linux, Mac OS X, OpenBSD, FreeBSD and Sun Solaris and provides all of the features you would expect from a PBX including many advanced features that are often associated with high end (and high cost) proprietary PBXs. Asterisk's architecture is designed for maximum flexibility and supports Voice over IP in many protocols, and can interoperate with almost all standards-based telephony equipment using relatively inexpensive hardware.
- Asterisk® is released as open source under the GNU General Public License (GPL), meaning that it is available for download free of charge. Asterisk® is the most popular open source software available, with the Asterisk Community being the top influencer in VoIP.

# Bridging everything together

- Asterisk:
  - bridges technologies together:
    - PSTN: analog, ISDN
    - Voice codings
    - VoIP: SIP/SDP/RTP, Skinny, H323, IAX, MGCP,
    - IP, linux, HTTP, DNS, ENUM
    - Messaging: Jabber, SMS, ...
    - Text to Speech
  - has a whole set of PBX features
  - all together creates a great framework and playground for innovative applications.
- Is an open-source project, supported by Digium, founded by Mark Spencer, author of Asterisk.

# Some Asterisk Features



- Bridging between any channel (PSTN, VoIP, ...) using any technology.
  - Transcoding between any channel
- Automated Attendant, Interactive Voice Response, Directory, Music on Hold, Call Detail Records, Text-to-speech
- Call Forward, Call Monitoring, Call Parking, Call Queuing, Call Recording, Call Routing, Call Transfer, Call Waiting, Blind Transfer, Remote Call Pickup, Caller ID, Voicemail
- Conferencing, Follow-me, Trunking
- Call centers, Call queues, Call agents, Predictive Dialing
- Database Integration
- E911, ENUM
- Fax Transmit and Receive (3rd Party OSS Package), SMS
- All free! And relatively easy to configure and use.

# Asterisk Distributions



- Main: <http://www.asterisk.org>
- Packaged with user interface for configuration:
  - asterisknow.org (from digium)
  - trixbox.org (with additional tools such as configuring phones, CRM, etc...)
- astlinux.org:
  - Embedded
  - Live CD
- openpbx.org
  - fork of asterisk 1.2. open to any open source code contribution.

# Tutorial Scope

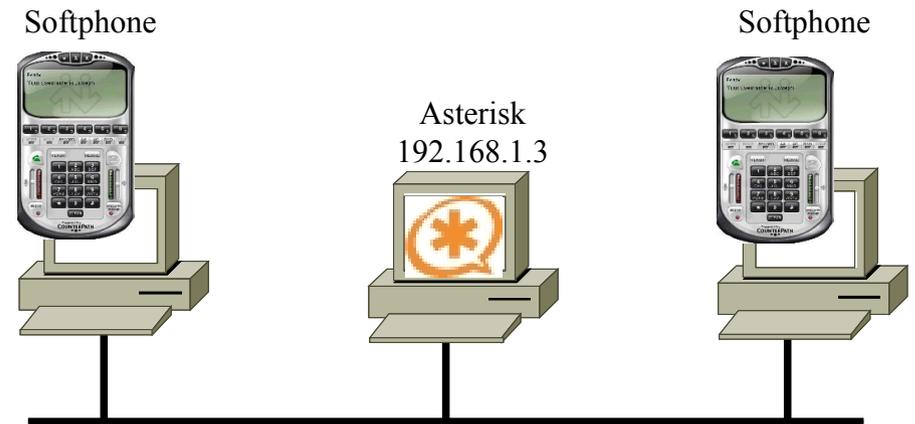


- Asterisk is pretty complex
  - In its sample config directory, there are 7500 lines of config data over 60 config files.
    - What is important? Which one should I care?
  - Bridges telephony, IP, SIP/SDP/RTP/TLS/..., security, NAT, Linux device drivers, jabber, sms, and others all together, each has his own complexity, dictionary, semantics, etc...
- This tutorial:
  - Step by step. Bottom-up approach: starts with minimal config and grow. Every step explained.
  - Not comprehensive, not everything. On purpose.
  - Practical with typical use scenarios
  - All config files, no nice web interface...

# First Call: Hello world!



- Setup:
  - Asterisk on linux
  - SIP softphones
  - On same LAN
- Call:
  - Dial 2 and hear “hello world”



# Installing Asterisk



- Main (i.e. asterisk.org) distribution components:
  - Asterisk itself, which includes everything except:
  - Zaptel: drivers for PSTN hardware cards
  - LibPRI: drivers for ISDN PRI interfaces
- Must compile/install drivers before asterisk:
  - Zaptel and libpri before asterisk.
- This tutorial discusses asterisk and zaptel.

# Installing Asterisk



- <http://ftp.digium.com/pub/zaptel/zaptel-1.4-current.tar.g>
- cd zaptel-1.4
- ./configure
- make
- make install
- <http://ftp.digium.com/pub/asterisk/asterisk-1.4-current.tar.gz>
- cd asterisk-1.4
- ./configure
- make
- make install

# Installing Asterisk (cont.)



- Some optional steps:
  - make menuselect
    - Choose compilation options, modules, sound files, etc...
    - Before doing make
  - Sound files:
    - When doing 'make install', it fetches sound files from digium web site.
  - make samples
    - Populates /etc/asterisk with sample configuration files
- This tutorial starts with an empty /etc/asterisk configuration directory and builds every config file.

# Basic Configuration Syntax



- To assign a value to a variable, use: `variable_name = value`.
- Comments start with a “;”. However, a few configuration files such as `/etc/zaptel.conf` do not obey that rule, so be careful.
- A context is started with `[context_name]`. A context groups statements together. All lines following a context start are part of the context until a new context is defined.
- A statement within a configuration file has the syntax: `statement_name => argument1, argument2, argument3, ...`
- All config files are in `/etc/asterisk`

# Configuring SIP



- SIP implemented as a module: `chan_sip.so` which reads its configuration file: `sip.conf`
- Each SIP phone has an entry in `sip.conf`

```
# cat sip.conf
```

```
[test1]           = Name of the phone (also username for SIP authentication)
type=friend       = can send and receive call
host=dynamic      = has a dynamic IP
secret=mysecret1  = password to authenticate to asterisk
disallow=all      = use only gsm codec; usually, don't need to specify codecs.
allow=gsm         here is for limiting modules. More on this later.
```

# Configuring the SIP Phone



- Using Counterpath X-Lite free softphone as example; available on windows, mac and linux.
- System Settings->SIP proxy->Default tab
  - Enabled: Yes = enable this config
  - Display name: Joe Smith = caller id
  - Username: test1 = authentication data to the SIP proxy (i.e. Asterisk)
  - Password: mysecret1
  - Domain/Realm: example.com
  - SIP proxy: 192.168.1.3 = IP address of Asterisk

# Relationships



```
# cat sip.conf
```

```
[test1]
```

```
type=friend
```

```
host=dynamic
```

```
secret=mysecret1
```

```
disallow=all
```

```
allow=gsm
```

- SIP Phone:
- System Settings->SIP proxy->Default tab
  - Enabled: Yes
  - Display name: Joe Smith
  - Username: test1
  - Password: mysecret1
  - Domain/Realm: example.com
  - SIP proxy: 192.168.1.3

# Dial Plan



- When a call is established with Asterisk, the dial plan, defined in `extensions.conf`, is executed.
- Example: when dialing 2, Asterisk will say “hello-world”.

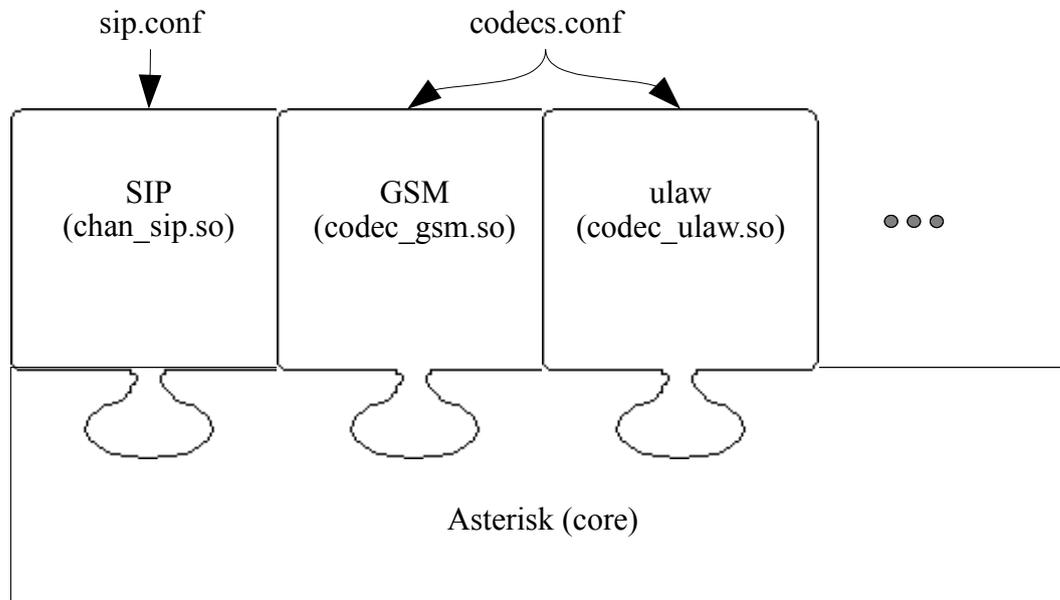
```
# cat extensions.conf
```

```
[default]                = Context (more on this later)
exten => 2,1,BackGround(hello-world) = when 2 is dialed, 1) play the 'hello-world' sound file
exten => 2,2,Hangup()      =                               and 2) terminate the call
```

- **Syntax:** `exten => extension_number, priority, action`

Note: A default asterisk installation contains the “hello-world” sound file.

# Asterisk Architecture



- (loadable) modules such as `chan_sip.so`, `codec_gsm.so`, `codec_ulaw.so`, ...
- Each has his own configuration file read at loading time.
- List of modules is defined in `modules.conf`
- Modules are located in `/usr/lib/asterisk/modules`

# Modules



- To narrow the scope of the configuration, this tutorial specifies the necessary Asterisk modules in `modules.conf` file.

```
# cat modules.conf
```

```
[modules]
```

<code>autoload=no</code>	= do not load all modules by default
<code>load =&gt;pbx_config.so</code>	= load the dialplan module (reads <code>extensions.conf</code> )
<code>load =&gt;codec_gsm.so</code>	= load gsm codec module
<code>load =&gt;format_gsm.so</code>	= load gsm reader module
<code>load =&gt;res_features.so</code>	= load features module
<code>load =&gt;app_dial.so</code>	= load the dial application module
<code>load =&gt;chan_sip.so</code>	= load the SIP channel module

Note: usually, don't need to touch this file. By default, Asterisk loads about every module.

# asterisk.conf



- Specifies location of directories for config files, modules, sounds, ...

```
# cat asterisk.conf
```

```
[directories]
```

```
astetcdir => /etc/asterisk
```

```
astmoddir => /usr/lib/asterisk/modules
```

```
astvarlibdir => /var/lib/asterisk
```

```
astagidir => /var/lib/asterisk/agi-bin
```

```
astspooldir => /var/spool/asterisk
```

```
astrundir => /var/run
```

```
astlogdir => /var/log/asterisk
```

- N.B. The sample file is ok.

Hint: if you want to test multiple sets of configuration files:

- create a directory for each set
- Change the astetcdir variable in asterisk.conf to the target directory
- Start asterisk by specifying the location of the desired asterisk.conf file:
  - Asterisk -C /a/b/asterisk.conf

# Hello World Whole Configuration



FILE	CONTENT
asterisk.conf	<pre>[directories] astetcdir =&gt; /etc/asterisk astmoddir =&gt; /usr/lib/asterisk/modules astvarlibdir =&gt; /var/lib/asterisk astagidir =&gt; /var/lib/asterisk/agi-bin astspooldir =&gt; /var/spool/asterisk astrundir =&gt; /var/run astlogdir =&gt; /var/log/asterisk</pre>
extensions.conf	<pre>[default] <b>exten =&gt; 2,1,BackGround(hello-world)</b> exten =&gt; 2,2,Hangup()</pre>
modules.conf	<pre>[modules] autoload=no load =&gt;pbx_config.so load =&gt;codēc_gsm.so load =&gt;format_gsm.so load =&gt;res_features.so load =&gt;app_dial.so load =&gt;chan_sip.so</pre>
sip.conf	<pre>[test1] <b>type=friend</b> <b>host=dynamic</b> <b>secret=mysecret1</b> disallow=all allow=gsm</pre>

# Hello world!



- Start Asterisk: `/usr/sbin/asterisk -cvvvv`
  - `-c` to get the Asterisk CLI on the terminal
  - `-vvvvv` to get some verbosity of what is going on.
  - (When in production, just start: `/usr/sbin/asterisk` )
- Start the softphone
- Dial 2 and press Send
- Hear “Hello world” and call terminates.

# Console Messages



- Registered SIP 'test1' at 192.168.1.103 port 5060 expires 1800
  - SIP phone test1 authenticated and registered by Asterisk
- Saved useragent "X-Lite release 1105x" for peer test1
  - Displays the SIP phone information string
- Executing BackGround("SIP/test1-a866", "hello-world") in new stack
  - SIP phone dialed 2 and Dialplan first instruction (background) is executed. Asterisk uses a unique id for each call.
- Playing 'hello-world' (language 'en')
- Executing Hangup("SIP/test1-a866", "") in new stack

# Second SIP phone



- Create similar entry in sip.conf for second SIP phone:

```
# cat sip.conf
[test1]
type=friend
host=dynamic
secret=mysecret1
disallow=all
allow=gsm

[test2]
type=friend
host=dynamic
secret=mysecret2
disallow=all
allow=gsm
```

# Second SIP phone



- Configure second SIP phone:
  - System Settings->SIP proxy->Default tab
    - Enabled: Yes
    - Display name: Fred Smith
    - Username: test2
    - Password: mysecret2
    - Domain/Realm: example.com
    - SIP proxy: 192.168.1.3

# Extensions for SIP phones



- Create extensions for the two phones so they can be called.

```
# cat /etc/asterisk/extensions.conf
```

```
[default]
```

```
exten => 2,1,BackGround(hello-world)
```

```
exten => 2,2,Hangup()
```

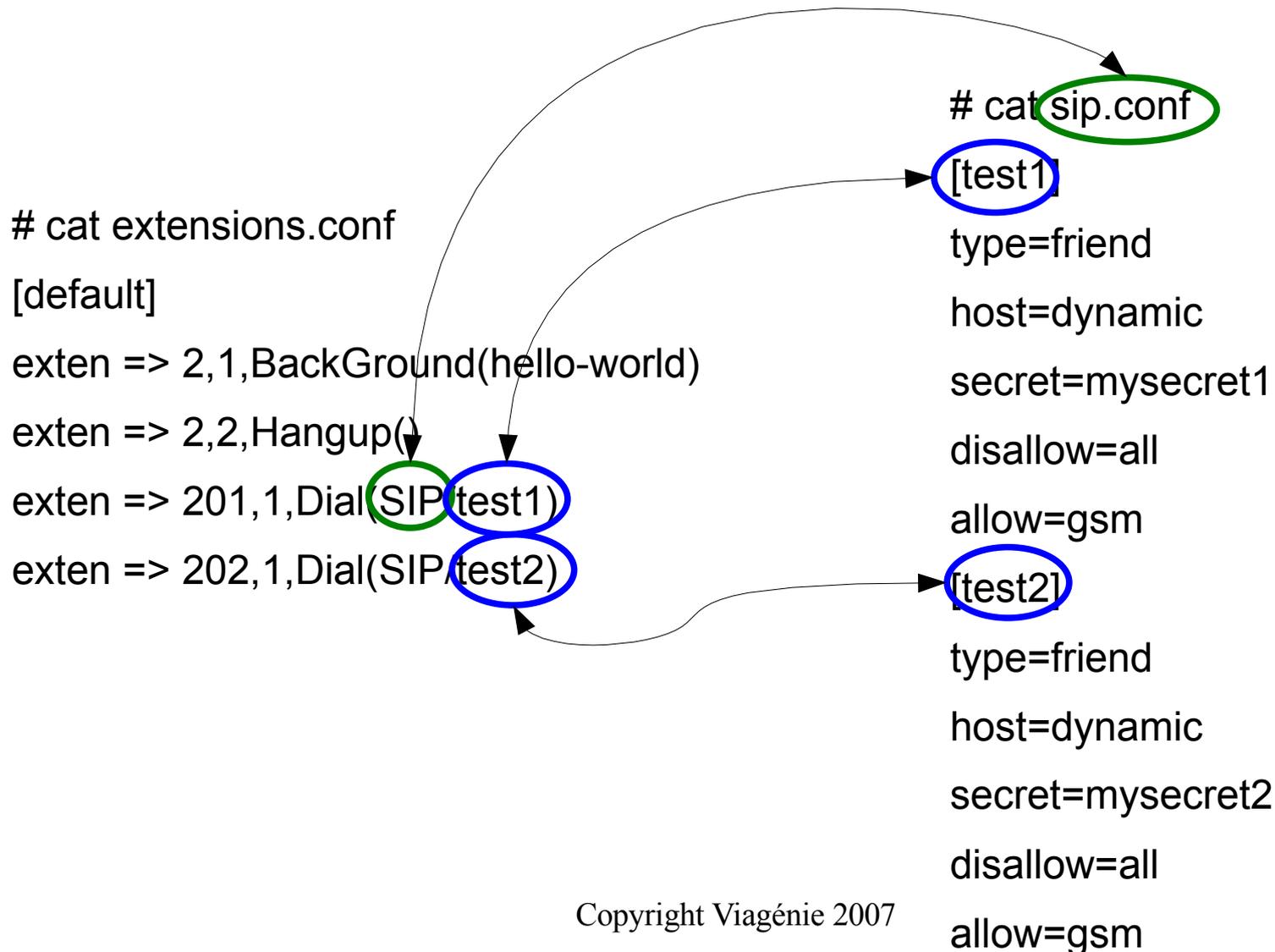
```
exten => 201,1,Dial(SIP/test1)
```

= When 201 is dialed, establish a SIP call with the 'test1' phone, defined in sip.conf.

```
exten => 202,1,Dial(SIP/test2)
```

= similar for 202.

# Relationships



# Enabling the Changes



- Reload the modules whose config files have changed:
  - CLI> module reload pbx\_config.so (for extensions.conf)
    - Or CLI> dialplan reload
  - CLI> module reload chan\_sip.so (for sip.conf)
- Or reload asterisk:
  - CLI> reload
- Or restart Asterisk:
  - CLI> stop now
  - # /usr/sbin/asterisk -cvvvv

# Call between the Two Phones



- Start the two softphones
- On test1, Dial 202 to reach the second phone and press Send
- Answer the call on test2.

# Troubleshooting



Task	How to
Verify that there is IP connectivity between the three computers (2 softphone and Asterisk)	Use the ping command between the computers.
Verify that Asterisk is running a SIP listener.	A SIP listener should be on UDP port 5060. On the Asterisk server, use “netstat -aun” to see the following line: <code>udp 0 0 0.0.0.0:5060 0.0.0.0:*</code>
Verify that the softphone can communicate with Asterisk.	On the softphone computer, use nmap, if available, to test if the softphone can reach the Asterisk SIP listener (192.168.1.3): <code>nmap -p 5060 -sU 192.168.1.3</code>
Verify any authentication issue with the softphone.	Look on the Asterisk console or in the logging directory (/var/log/asterisk) for errors.
Verify SIP interaction	On Asterisk console, type “sip debug” and restart the call. One should see details of each SIP packet sent and received by Asterisk.

# Troubleshooting



ISSUE	POSSIBLE RESOLUTIONS
SIP listener is not up on Asterisk.	The Asterisk server hostname and IP address is not registered in /etc/hosts or DNS. Add the hostname and IP address in /etc/hosts.
Packets are flowing only in one direction.	Make sure your host firewall (for example iptables) is not configured too restrictive. To temporary test this issue, disable it(/etc/init.d/iptables stop) and then see if packets are flowing in both directions. Use “iptables -L” to list the current filtering rules. Use “tcpdump -vn udp and port 5060” to view the SIP signaling traffic on the LAN. Make sure there is no NAT between Asterisk and the softphones.
No sound, but packets are flowing and SIP is running	Make sure the volume is not muted on the softphones. Make sure the softphone can acquire the audio device.
Hearing echo.	Don't care about this now, but if it is too annoying, then see section 16.3 which discusses echo canceling.
Changes made to sip.conf do not seem to be taken into account.	Make sure to reload the SIP module by issuing “module reload chan_sip.so” on the console or restart Asterisk by issuing: “stop now” and “/usr/sbin/asterisk -cvvvvv”.

# Troubleshooting

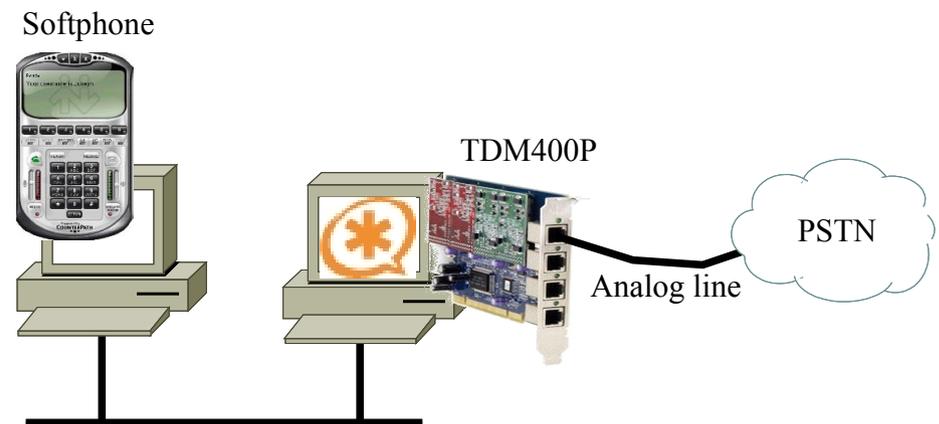


<b>COMMAND</b>	<b>DESCRIPTION</b>
sip debug	Print detailed debugging information to the console about SIP sessions.
sip show peers	Lists the SIP phones attached to Asterisk.
sip show peer test1	Show the information about the 'test1' SIP phone.
stop now	Stops Asterisk.
module reload chan_sip.so	Reload the SIP module which includes rereading the sip.conf configuration file.
module reload pbx_config.so	Reread the extensions.conf configuration file and reload the module.
dialplan reload	(Only) Reread the extensions.conf configuration file.

# Connecting to the PSTN



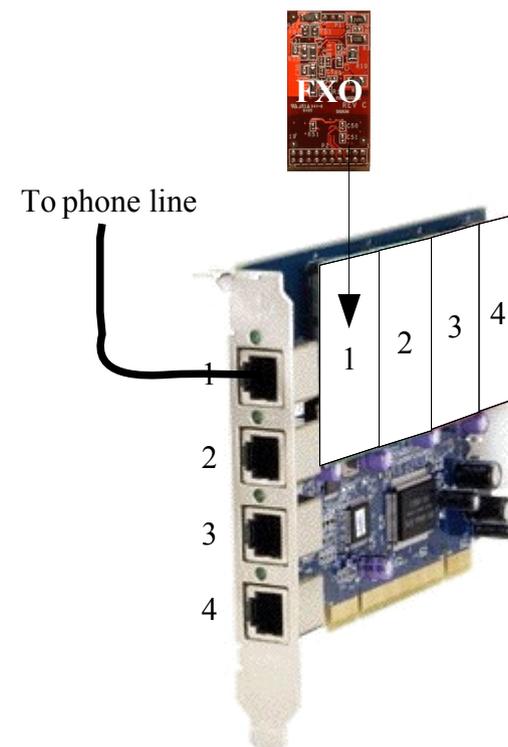
- Setup:
  - Asterisk on linux
    - with a Digium TDM400P card
    - Connected to the PSTN
  - Another computer with softphone



# Digium TDM400P



- Cheapest Digium card.
- Supported by Asterisk (others are also supported)
- Has 4 slots for daughter cards:
  - FXO (Foreign eXchange Office): to connect to an analog line
  - FXS (Foreign eXchange Station): to connect to an analog phone
- This tutorial: one FXO daughter card connected to the PSTN
- Note: the power connector on the main card is only used if phones are attached to FXS daughter cards



# Terminology



- An FXO card/port:
  - Connects to an analog line on the PSTN
  - Uses FXS signaling
- An FXS card/port:
  - Connects to an analog phone
  - Uses FXO signaling
  -

# Installing Zaptel



- Zaptel (Zapata Telephony) are the drivers used for PSTN cards. Zaptel software is managed by Digium but is separate from and independent of Asterisk.
- The zaptel install created loadable kernel modules to support the hardware card.

```
# cat /etc/zaptel.conf      = zaptel config file is in /etc
fxsks=1                    = configure port 1 for FXS
                            signaling(fxs) with kewlstart (ks)
```

# Loading Zaptel



- Kernel module

```
# /sbin/modprobe -v wctdm
```

```
# /sbin/lsmmod
```

Module	Size	Used by
--------	------	---------

wctdm	43968	0
-------	-------	---

zaptel	202244	1 wctdm
--------	--------	---------

crc_ccitt	2241	1 zaptel
-----------	------	----------

- Configuring zaptel

```
# /sbin/ztcfg -vvvv
```

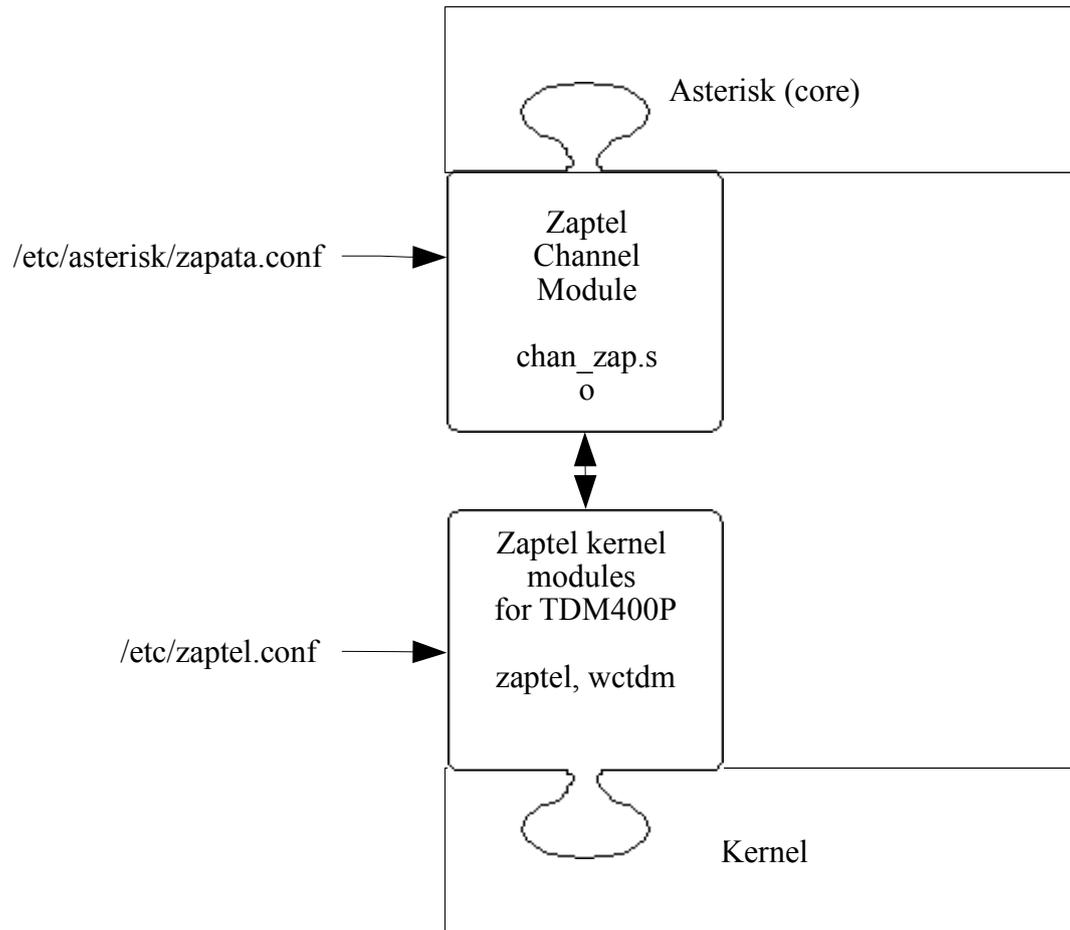
```
Zaptel Configuration
```

```
=====
```

```
Channel map:
```

```
Channel 01: FXS Kewlstart (Default) (Slaves: 01)
```

# Using zaptel in Asterisk



# Configuring Zaptel for Asterisk



- `/etc/asterisk/zapata.conf`
- Order of statements is important. Any above statement is still in force.

```
# cat /etc/asterisk/zapata.conf
```

```
[channels]
```

= channels definitions

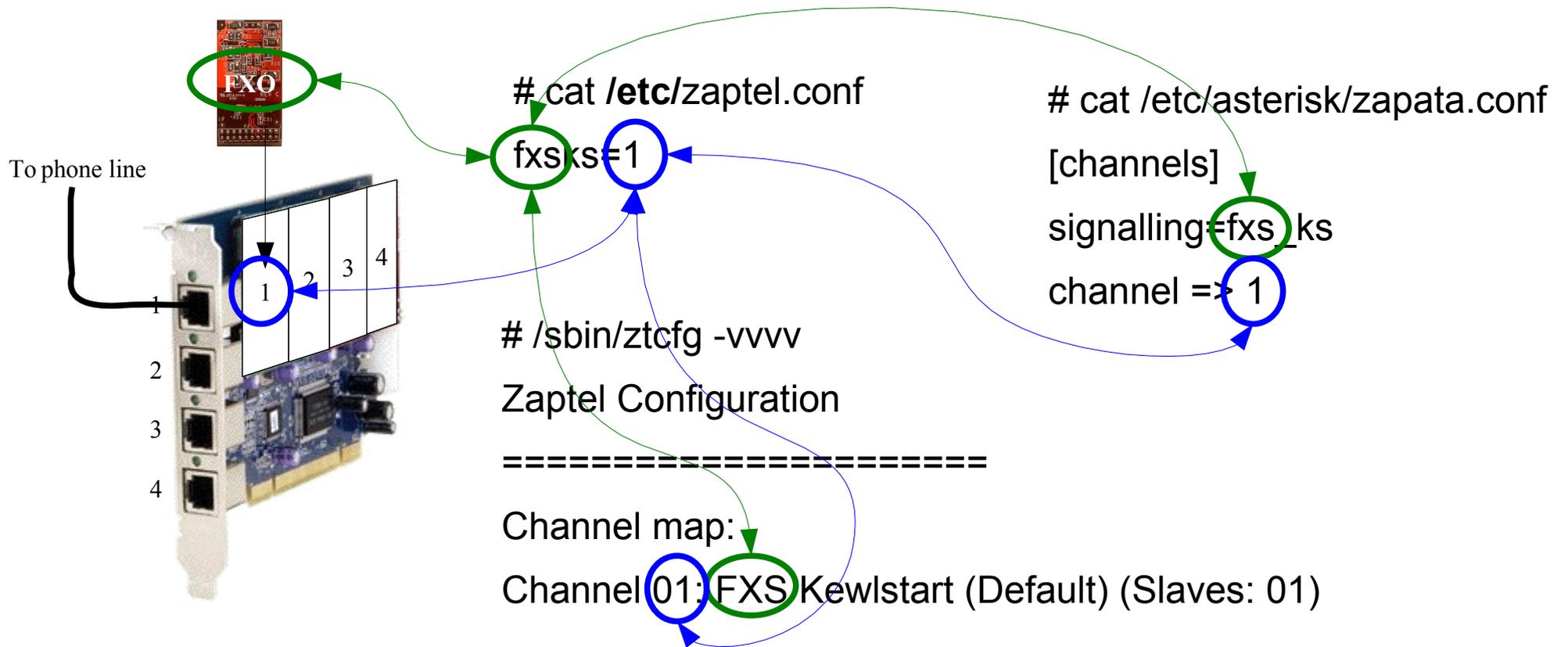
```
signalling=fxs_ks
```

= signalling is fxs\_ks

```
channel => 1
```

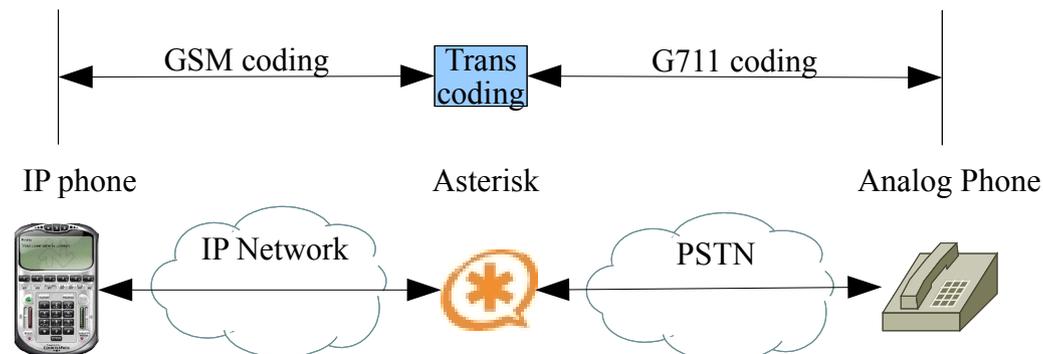
= apply all current settings (i.e. signalling=fxs\_ks)  
to channel 1

# Relationships



# Codecs

- (Analog) Voice is encoded into frames. Different coders/decoders (codec) exist for various conditions: low-bandwidth, limited high fidelity (voice versus music), etc...
- VoIP phones use some codecs (ex: GSM), PSTN use some others (ex: G711 also known as ulaw/alaw).
- When traversing channels, Asterisk transcode the voice between different codecs.



# Adding PSTN-related Modules



```
# cat /etc/asterisk/modules.conf
```

```
[modules]
```

```
autoload=no
```

```
load =>pbx_config.so
```

```
load =>codec_gsm.so
```

```
load =>codec_alaw.so           = load the alaw/ulaw codec modules
```

```
load =>codec_ulaw.so
```

```
load =>format_gsm.so
```

```
load =>res_features.so
```

```
load =>app_dial.so
```

```
load =>chan_sip.so
```

```
load =>chan_zap.so           = load the zaptel module which reads zapata.conf
```

# Calling the PSTN



- When calling outside, use 9 then the number. (ex.: user dials: 9-1-418-555-1212)
- In dial plan, use a pattern match: “\_9.”
  - “\_” means start of a pattern
  - “9” is the first dialed digit
  - “.” means one to any number of digits.
- Dial into zaptel channel 1 the dialed number:
  - “Dial(ZAP/1/\${EXTEN})”
- but do not send the first digit (“9”) to the PSTN:  
\${EXTEN:1}
  - exten => \_9.,1,Dial(ZAP/1/\${EXTEN:1})

# Dial Plan



- Dialing:
  - 2 to hear “hello-world”
  - 201 to reach test1 SIP phone
  - 202 to reach test2 SIP phone
  - 9 and a number to call the PSTN

```
# cat extensions.conf
[default]
exten => 2,1,BackGround(hello-world)
exten => 2,2,Hangup()
exten => 201,1,Dial(SIP/test1)
exten => 202,1,Dial(SIP/test2)
exten => _9.,1,Dial(ZAP/1/${EXTEN:1})
```

```

exten => 2,1,BackGround(hello-world)
exten => 2,2,Hangup()
exten => 201,1,Dial(SIP/test1)
exten => 202,1,Dial(SIP/test2)
exten => _9.,1,Dial(ZAP/1/${EXTEN:1})

```

# Dial Plan



Dialed number	Actions executed	Description
914165551212	Dial(ZAP/1/14165551212)	Executes a phone call to 14165551212 onto the PSTN.
9011331231234	Dial(ZAP/1/011331231234)	Executes a phone call to 011331231234 onto the PSTN.
201	Dial(SIP/test1)	Executes a SIP call to user phone test1.
911	Dial(ZAP/1/11)	Executes a phone call to 11 onto the PSTN. '911' is an emergency call, and on the PSTN, dialing 11 is not the emergency call. So this pattern is not appropriate for 911. A specific pattern for 911 should be defined or the 911 should be dialed differently. Give it a try!
9	Error	The '_9.' pattern requires at least one digit after the 9. Asterisk will send an error code to the calling phone stating that the number is incomplete (error code is "484 address incomplete")

# Calling into the PSTN



- On softphone, dial “9-555-1212”:
  - Asterisk match the pattern with “\_9.”
  - Dial(ZAP/1/\${EXTEN:1} => Dial(ZAP/1/5551212)
  - Bridge the two legs: SIP(phone to asterisk) and ZAP (asterisk to PSTN)
  - Transcode all voice frames if necessary

#	Console Messages
1	-- Executing Dial("SIP/test1-76a4", "Zap/1/5551212") in new stack
2	-- Called 1/5551212
3	-- Zap/1-1 answered SIP/test1-76a4
4	-- Hungup 'Zap/1-1'

# Receiving a call from the PSTN



- With the current configuration, Asterisk does not know what to do when a call enters from the PSTN into zaptel.

- We want Asterisk to:

- Answer the inbound call
- Say welcome
- Wait for the caller to dial an extension
- Bridge the pstn call with the dialed extension

```
# cat extensions.conf
```

```
[default]
```

```
exten => s,1,Answer()
```

```
exten => s,2,BackGround(hello-world)
```

```
exten => s,3,WaitExten()
```

```
exten => 2,1,BackGround(hello-world)
```

```
exten => 2,2,Hangup()
```

```
exten => 201,1,Dial(SIP/test1)
```

```
exten => 202,1,Dial(SIP/test2)
```

```
exten => _9.,1,Dial(ZAP/1/${EXTEN:1})
```

- The 's' (start) extension is used for inbound calls into dial plan.

# Troubleshooting



Issue	Possible resolutions
The ztcfg program reports errors.	Verify that signaling for the FXO port is fxsks in zaptel.conf.
Zaptel modules are not loaded	The computer might have been rebooted and the kernel did not automatically load the modules. Issue a modprobe -v wctdm command.
Calls are not established to the PSTN	Verify that the signaling for the channel 1 in /etc/asterisk/zapata.conf matches the signaling for port 1 in /etc/zaptel.conf
Asterisk does not recognize any zaptel channel.	If the computer has rebooted, make sure the kernel modules were reloaded by the modprobe -v wctdm command.
Changes made to the zaptel.conf are not recognized correctly.	Use the ztcfg -vvvv to reread the zaptel.conf file. If still unsuccessful, remove the kernel drivers using the modprobe -r -v wctdm command and then reload them.

# Useful Commands



Command	Description
zap show status	Shows the status of the zaptel hardware.
zap show channels	Lists the zaptel channels available.
zap show channel 1	Shows the details of the zaptel channel 1.
module reload chan_zap.so	Reload the zaptel module which includes rereading the zapata.conf configuration file.
module reload pbx_config.so	Reread the extensions.conf configuration file.
dialplan reload	Reread the extensions.conf configuration file. (similar to reload pbx_config.so)
stop now	Stops Asterisk

# Small Office Setup



- one office
- 3 PSTN lines
- 1 VoIP provider line (used for long-distance calls)
- SIP phones on employees desks
- Employees sometimes working from home or are on the road, but want to be reachable as if they were at the office
- Interactive voice response(IVR) using a personalized greeting message
- Voicemail, access to voicemail, voicemail notifications by email
- Directory

# Information Gathering



<b>Office Extension</b>	<b>Name</b>	<b>Voicemail access password</b>	<b>Email (for voicemail notification)</b>	<b>Mobile number</b>
101	John Smith	1234	<a href="mailto:js@example.org">js@example.org</a>	555-1234
102	Mary Smith	4567	<a href="mailto:ms@example.org">ms@example.org</a>	555-2345
103	Peter Smith	6789	<a href="mailto:ps@example.org">ps@example.org</a>	None

<b>User</b>	<b>Phone</b>	<b>Extension</b>	<b>Phone password</b>
John	Office	101	pw101
	Home or on the road	151	pw151
Mary	Office	102	pw102
	Home or on the road	152	pw152
Peter	Office	103	pw103

# Information Gathering



User	Situation	Action
John	New call	Ring desk phone, home office phone and cell phone, all at the same time.
	Not answering	Send to voicemail. Notify by email with voice message attached.
Mary	New call	Ring desk phone, home office phone.
	Not answering	Ring cell phone. If cell phone is not answered, forward to Peter.
Peter	New call	Ring desk phone.
	Not answering	Send to voicemail.

Extension	Action
Greeting	When someone calls in, say 'Welcome to ACME. Please dial the extension now or access the directory using the pound key' Then wait for an extension to be pressed.
500	Users access to their voicemail. They should give their extension and their password.
0	Operator extension. Since no one in the office is a dedicated operator, and since people may not be there, ring all employees and the first one available will answer. Copyright Viagénie 2007
8	Access to the company directory.

# Information Gathering



<b>Situation</b>	<b>Action</b>
Incoming call on the main number (either primary or secondary line)	Call coming from the PSTN. Answer and send to the main IVR menu ('Welcome to ACME inc., dial the extension number of the person you are calling').
Outgoing local call	Prefer the PSTN provider because local calls are free. Use VoIP provider as last resort (when all lines are busy) because the local VoIP calls are billed by the minute.
Outgoing long-distance call	Prefer the VoIP provider because long-distance calls are cheaper. Use PSTN provider as last resort (when all VoIP lines are busy) since their long-distance rates are higher.

# Starting Configuration



File	Content
asterisk.conf	[directories] astetcdir => /etc/asterisk astmoddir => /usr/lib/asterisk/modules astvarlibdir => /var/lib/asterisk astagidir => /var/lib/asterisk/agi-bin astspooldir => /var/spool/asterisk astrundir => /var/run astlogdir => /var/log/asterisk
/modules.conf	[modules] autoload=no load =>pbx_config.so load =>codec_gsm.so load =>codec_alaw.so load =>codec_ulaw.so load =>format_gsm.so load =>res_features.so load =>app_dial.so load =>chan_sip.so load =>chan_zap.so

# Configuring Office Phones



- Office phone template:

sip.conf:

[103]

type=friend

host=dynamic

secret=pw103

extension.conf:

exten => 103, 1, Dial(SIP/103)

# Configuring On the Road Phones



- On the road phones:
  - Most likely behind NAT
  - Not always on

- **template:**

```
sip.conf:  
[152]  
type=friend  
host=dynamic  
secret=pw152  
nat=yes  
qualify=yes
```

- nat=yes tells asterisk to take into account the external IP and port numbers of the phone are different then the one within the payload of SIP and SDP.
- qualify=yes tells asterisk to verify regularly if the phone is still there. If not there, then won't try to send a call to it.

# Voice Mail



- Add some modules

```
# cat >> /etc/asterisk/modules.conf  
load=>format_wav.so  
load=>res_adsi.so  
load=>app_voicemail.so
```

Voicemail stores messages in wav format  
Voicemail uses ADSI for signaling a new message to the user phone (blinking led)  
Voicemail application

- Create user entries

```
# cat /etc/asterisk/voicemail.conf  
[default]  
103 => 6789,Peter Smith, ps@example.org
```

103: User extension  
6789: voicemail user access password (to read his email)  
Peter Smith: Directory entry  
[ps@example.org](mailto:ps@example.org): email to send a copy of new voice messages

# Voice Mail



- Leaving a voicemail is used when
  - the phone rings and the call is not answered after a timeout
  - Or when the phone is busy.
- In the dialplan:
  - Timeout: asterisk moves to the next priority
  - Busy: asterisk moves to the priority number +101
    - Exemple:
      - exten => 111, **1**, Dial(SIP/test1).
      - Next priority executed is  $1 + 101 = 102$
      - exten => 111, **102**, .....

# Voice Mail



- **New user template with voicemail:**

exten => 103, 1, Dial(SIP/103)

**exten => 103, 2, Voicemail(103)**      Timeout: no answer. Go to voicemail

**exten => 103, 3, Hangup()**              after leaving a message in voicemail, hung up

**exten => 103, 102, Voicemail(103)**      phone busy. Go to voicemail

- **Users accessing their voicemail:**

exten => 500, 1, VoiceMailMain()

- Voicemailmain() is a complete application for accessing voicemail: with menus, authentication of users, greeting message, reading/deleting/archiving messages, etc...

# Voice Mail



- New message indication:

```
# cat sip.conf
```

```
[103]
```

```
type=friend
```

```
host=dynamic
```

```
secret=pw103
```

```
mailbox=103
```

- MWI (Message Waiting Indication).
- SIP phone send a SUBSCRIBE message to the voicemail, so that when a new voice mail message is left, the SIP phone will receive an advertisement and will display “New message waiting” or led blinking or ...

# Voice Mail Configuration Summary



- For each user:

```
# cat voicemail.conf
```

```
103 => 6789,Peter Smith, ps@example.org
```

```
# cat extensions.conf
```

```
exten => 103, 1, Dial(SIP/103)
```

```
exten => 103, 2, Voicemail(103)
```

```
exten => 103, 3, Hangup()
```

```
exten => 103, 102, Voicemail(103)
```

```
# cat sip.conf
```

```
[103]
```

```
type=friend
```

```
host=dynamic
```

```
secret=pw103
```

```
mailbox=103
```

```
# cat extensions.conf
```

```
exten => 500, 1, VoiceMailMain()
```

# Relationships



```
# cat voicemail.conf
```

```
103 => 6789, Peter Smith, ps@example.org
```

```
# cat extensions.conf
```

```
exten => 103, 1, Dial(SIP/103)
```

```
exten => 103, 2, Voicemail(103)
```

```
exten => 103, 3, Hangup()
```

```
exten => 103, 102, Voicemail(103)
```

```
# cat sip.conf
```

```
[103]
```

```
type=friend
```

```
host=dynamic
```

```
secret=pw103
```

```
mailbox=103
```

# Working Remotely



- User requirements:
  - When a call arrives, ring simultaneously office and home/on the road phones.
    - Use '&' to add more devices in the dial string.
  - If no answer, then ring my cell phone
  - If no answer, then ring receptionist.

```
# cat extensions.conf
```

```
exten => 102, 1, Dial(SIP/102&SIP/152)
```

```
exten => 102, 2, Dial(Zap/1/5552345)
```

```
exten => 102, 3, Dial(SIP/103)
```

# Directory



- Add the directory module:  

```
# cat >> /etc/asterisk/modules.conf
```

```
load=>app_directory.so
```
- Access to directory by dialing 8:  

```
# cat extensions.conf
```

```
exten => 8,1,Directory(default)
```
- Directory application:
  - Have a greeting message (please press the last name of the party you are calling)
  - Suggest extensions and read names
  - Control the flow
- Data (names, context) is taken from voicemail.conf.

# Module Management



- This tutorial shows how to insert new modules into Asterisk, one at a time, in `modules.conf`. The purpose is to show how it works.
- However, this method is more difficult to manage in a typical asterisk deployment which uses many many features, because:
  - the large number of modules to specify
  - the loading order is (somewhat) important because of dependencies in modules
- Therefore, one might want to use in `modules.conf`:
  - `autoload=yes`
  - Which loads all modules automatically.

# Interactive Voice Response



- Requirement: when someone dial '0', ring all phones.  
(small company: everyone is the receptionist)  
exten => 0,1,Dial(SIP/101&SIP/102&SIP/103)
- Greeting message:
  - Need to create a sound file coded in one of the supported codecs
  - Can use asterisk to create that file. The Record() application is used to record the voice on the call to a file. Use your phone and dial an extension that records the voice

# Interactive Voice Response



- Greeting message:
  - Temporary extension to create the sound file:
    - exten => 501,1,Wait(2)
    - exten => 501,2,Record(acme-greeting.gsm)
    - exten => 501,3,Wait(2)
    - exten => 501,4,Playback(acme-greeting.gsm)
    - exten => 501,5,Hangup()
  - When recorded, update the “start” extension (used when a call land into asterisk):
    - exten => s,1,Answer()
    - exten => s,2,Background(acme-greeting)
    - exten => s,3,WaitExten()

# Configuring PSTN access



- Requirement: 3 analog lines. One fax. Two others grouped together by telco for inbound calls. All three can be used for outbound calls.

Port Number	Function
1	Main number
2	Fax
3	Secondary line

```
# cat /etc/zaptel.conf
```

```
fxsks=1-3
```

```
# /sbin/ztcfg -vvvv
```

= 3 FXO daughter cards

connected to the 3 analog lines

# Configuring PSTN access



```
# cat /etc/asterisk/zapata.conf
```

```
[channels]
```

```
signalling=fxs_ks
```

```
group=1
```

= Channels 1-3 are grouped

```
channel => 1-3
```

together as g1 (group 1)

- Use PSTN to dial out local area. Local area is defined as area codes 791, 792 that you need to dial

```
exten => _979[12].,1,Dial(ZAP/g1){EXTEN:1}
```

- ZAP/g1 refers to the group of 3 lines. Asterisk will use the first available line of the group.

# Using the VoIP Provider



- Used primarily for terminating long-distance calls (cheaper) to the SIP-based VoIP provider
- Asterisk is initiating calls, as a SIP client to the VoIP provider SIP server.

```
# cat >> sip.conf
```

```
[voip-outgoing]
```

```
type=peer
```

```
username=acme
```

```
secret=acmesecret
```

```
host=sip.voipprovider.com
```

```
fromuser=acme
```

```
fromdomain=voipprovider.com
```

= New SIP definition

= an asterisk peer can only receive calls

= authentication of asterisk

to the VoIP provider

= VoIP provider SIP server

= sets the SIP from field so that VoIP provider

knows who you are. By default, Asterisk would put

the originating SIP device identification (i.e.

Desktop phone)

# SIP peers



- 3 types of SIP peers in sip.conf:

<b>SIP peer type</b>	<b>Description</b>
type=peer	SIP peer used only for outgoing calls
type=user	SIP peer used for incoming calls
type=friend	SIP peer used for incoming and outgoing calls. Friend is the combination of peer and user.

# Using the VoIP Provider



- Using the VoIP provider for long-distance calls.
- 9-1-.... for north-america
- 9-011-... for elsewhere

```
# cat extensions.conf
```

```
exten => _91.,1,Dial(SIP/voip-outgoing/${EXTEN:1})
```

```
exten => _9011.,1,Dial(SIP/voip-outgoing/{EXTEN:1})
```

# Current Dial Plan



```
# cat extensions.conf
default]
;--- incoming calls ---
exten => s,1,Answer()
exten => s,2,Background(acme-greeting)
exten => s,3,WaitExten()
;--- Phones ---
; John office phone
exten => 101,1,Dial(SIP/101)
exten => 101,2,Voicemail(101)
exten => 101,3,Hangup()
exten => 101,102,Voicemail(101)
; John Home office phone (not shown)
; Mary office phone (not shown)
; Mary home office phone (not shown)
; Peter office phone (not shown)

; --- IVR ---
exten => 0, 1, Dial(SIP/101&SIP/102&SIP/103)
exten => 8,1,Directory(default)
exten => 500,1,VoiceMailMain()
; --- outgoing calls ---
exten=>_979[12].,1,Dial(ZAP/g1/${EXTEN:1})
exten => _91.,1,Dial(SIP/voip-outgoing/${EXTEN:1})
exten => _9011.,1,Dial(SIP/voip-outgoing/{EXTEN:1})
; --- temporary to record greeting---
exten => 501,1,Wait(2)
exten => 501,2,Record(acme-greeting.gsm)
exten => 501,3,Wait(2)
exten => 501,4,Playback(acme-greeting.gsm)
```

# Current Dial Plan



- Considerations:
  - Templates for user phones
    - Use macros
  - One single “context”:
    - Everybody entering can do everything.
    - a caller from outside can do long-distance calls on your bill... :(
    - A caller can re-record your greeting message...
    - Solution: create multiple contexts. For example, incoming calls may only dial phone extensions.
  - Local calls pattern is not specific enough.

# Contexts



- Defined as [*context\_name*]
- Groups the following statements until next context
- Contexts are mutually exclusive
  - `exten 123` may exist in `context1` and `context2`.
- Contexts may include other contexts (recursively) using the `include => including_context_name` statement.

# Contexts



- Channel configurations may specify an entering context.
  - ex. sip.conf: all calls coming from test1 will enter the 'internal' context in the dial plan.  
[test1]  
host=dynamic  
...  
context=internal
- Good practice is to
  - group together, into separate contexts, extensions at the same level of privilege
  - set channel config to enter into the most limited appropriate context

# New Contexts



- [default] : for entering the dial plan
- [phones] : company phone extensions
- [ivr]: menus of the IVR
- [pstn\_outgoing]: dialing out to the PSTN
- [voip\_outgoing]: dialing out to the VoIP provider
- [temporary]: to record the greeting message
- [external] includes: default, phones, ivr
  - Used by external callers
- [internal] includes: external, pstn\_outgoing, voip\_outgoing, temporary
  - Used by employees

# New Dial Plan



[default]

;--- incoming calls ---

exten => s,1,Answer()

exten => s,2,Background(acme-greeting)

exten => s,3,WaitExten()

**[phones]**

;--- Phones --- (not shown)

; John, Mary, Peter office phones

; (not shown)

**[ivr]**

; --- IVR --

exten => 0, 1,

Dial(SIP/101&SIP/102&SIP/103)

exten => 8,1,Directory(default)

exten => 500,1,VoiceMailMain()

; --- outgoing calls ---

**[pstn\_outgoing]**

exten=>\_979[12].,1,Dial(ZAP/g1/\${EXTEN:1})

**[voip\_outgoing]**

exten => \_91.,1,Dial(SIP/voip-outgoing/\${EXTEN:1})

exten =>\_9011.,1,Dial(SIP/voip-outgoing/{EXTEN:1})

**[temporary]**

; --- temporary to record ---

exten => 501,1,Wait(2)

exten => 501,2,Record(acme-greeting.gsm)

exten => 501,3,Wait(2)

exten => 501,4,Playback(acme-greeting.gsm)

exten => 501,5,Hangup()

# New Dial Plan (cont.)



## **[external]**

include => default

include => phones

include => ivr

; internal is for employees. Includes all!

## **[internal]**

include => external

include => pstn\_outgoing

include => voip\_outgoing

include => temporary

# Adjusting inbound channels



- Adjusting inbound channels to use contexts
- sip.conf (for employees SIP devices)

```
[103]
```

```
...
```

```
context=internal
```

- zapata.conf (for calls coming from the PSTN)

```
[channels]
```

```
...
```

```
context=external
```

```
channel => 1-3
```

# Polishing Extensions



- When adding and removing steps within an extension, one has to renumber the priorities.
- If one priority is missing (ie. 1,2,4,5), then Asterisk stops (i.e. 1,2 then stops).
- To avoid, use 'n' (next) priority for all priorities except 1.

exten => 501,1,Wait(2)

exten => 501,2,Record(acme-greeting.gsm)

exten => 501,3,Wait(2)

exten => 501,4,Playback(acme-greeting.gsm)

exten => 501,5,Hangup()

exten => 501,1,Wait(2)

exten => 501,**n**,Record(acme-greeting.gsm)

exten => 501,**n**,Wait(2)

exten => 501,**n**,Playback(acme-greeting.gsm)

exten => 501,**n**,Hangup()

# Polishing Extensions



- If dialed digits do not match in current context, Asterisk goes to the 'i' extension (in the current context)
- If no dialed digits after timeout, Asterisk goes to the 't' extension.
- If these extensions do not exist, then hung up.

exten => i,1,Background(invalid&extension)

exten => i,n,Goto(s,1)

exten => t,1,Background(please-try-again)

exten => t,n,Goto(s,1)

# Polishing Extensions



- Dial() to a SIP channel (such as SIP phone) has a default timeout of 60 seconds. Too long.
- Change globally in sip.conf:
  - rtptimeout=10
- Or set it as option to Dial()

```
exten => 102,1,Dial(SIP/102&SIP/152,10)
```

# Extension Pattern Matching



- More precise pattern matching means:
  - Faster feedback to the user for wrong number, since the dialed number is not valid with any pattern match in the dial plan
  - More security
  - More control of routing, etc..

<b>Pattern character</b>	<b>Description</b>	<b>Regex equivalent</b>
–	Starts the pattern matching expression	
X	0-9	[0-9]
Z	1-9	[1-9]
N	2-9	[2-9]
[]	One digit within the specified set. The '-' character within a set means a range. For example, [14-9] is the pattern for a digit in this set: 1, 4, 5, 6, 7, 8, 9.	[]
.	One or more character	.+

# Use of Pattern Matching



- Example using North American Numbering Plan (NANP) :

<b>Pattern</b>	<b>Description</b>
_NXXXXXX	Local number in North-America.
_NXXNXXXXXX	Local number in North-America with 10 digits dialing.
_1NXXNXXXXXX	A long-distance number in North-America.
_011.	A long-distance number outside of North-America.
_1900NXXXXXX	Number in North-America where calls are billed to the calling party.

# NANP and Dial Plan



Current:

```
[pstn_outgoing]
```

```
exten=> _979[12].,1,Dial(ZAP/g1/${EXTEN:1})
```

```
[voip_outgoing]
```

```
exten => _91.,1,Dial(SIP/voip-outgoing/${EXTEN:1})
```

```
exten => _9011.,1,Dial(SIP/voip-outgoing/{EXTEN:1})
```

New:

```
[pstn_outgoing]
```

```
exten=> _979[12]NXXXXXX,1,Dial(ZAP/g1/${EXTEN:1})
```

```
[voip_outgoing]
```

```
exten => _91NXXNXXXXXX,1,Dial(SIP/voip-outgoing/${EXTEN:1})
```

```
exten => _91900NXXXXXX,1,Congestion()
```

```
exten => _9011.,1,Dial(SIP/voip-outgoing/{EXTEN:1})
```

# Conferencing



- Implemented by the MeetMe() application
- Requirements:
  - One conference room accessible with the '1234' password
  - One conference room accessible without a password

```
# cat meetme.conf
```

```
[rooms]
```

```
conf => opened
```

```
conf => private,1234
```

```
# cat extensions.conf
```

```
exten => 600,1,MeetMe(opened)
```

```
exten => 601,1,Meetme(private)
```

# Conclusion



- This tutorial the essentials to start using Asterisk:
  - Making the “hello-world” SIP call
  - Calling between two SIP phones
  - Calling to the PSTN
  - Small office setup
- It showed 1% of power of Asterisk!

# Questions?



This presentation is based on material that will appear in an upcoming book “Asterisk Essentials” to be published by Wiley, Fall 2007.

Any feedback will be greatly appreciated.

Marc.Blanchet@viagenie.ca

This presentation: <http://www.viagenie.ca/publications/>

## References

- Marc Blanchet, Migrating to IPv6, Wiley, 2006, ISBN 0-471-49892-0, <http://www.ipv6book.ca>