

Making and taking calls with octothorpe

Matt Behrens · <https://www.zigg.com/> · @zigg

PyOhio · July 26, 2014

Introduction

- Matt Behrens, from Grand Rapids, Michigan
- Developer for over three decades, 1½ on Python
- Unix/Linux fan since the mid '90s
- Spent a few years developing and maintaining telephony and communications software for Linux

The Asterisk platform

Asterisk

- Runs on Unix platforms but mostly Linux
- Provides an abstraction to various telephony technologies
 - SIP/RTP
 - DAHDI (analog and digital)
 - Other, more esoteric channel drivers... in various states of maintenance

How to get Asterisk

- From <http://www.asterisk.org/downloads>:
 - Source code, build it yourself
 - AsteriskNOW distribution, install it on your own system
- Add packages to RHEL/CentOS 6; see <https://wiki.asterisk.org/wiki/display/AST/Asterisk+Packages>

Asterisk basics

- Almost all Asterisk configuration lives in `/etc/asterisk`
- `extensions.conf` holds the dialplan, Asterisk's own internal scripting language
- Dialplan is required, but limited in capability

Dialplan example

```
[default]
```

```
; This is a simple test to answer the line and play a "Hello, world!"  
; sound.
```

```
exten => 401,1,Answer  
same => n,Playback(silence/1&hello-world&silence/1)  
same => n,Hangup
```

```
; This is another test that will answer the line and dial through to  
; extension 202.
```

```
exten => 402,1,Answer  
same => n,Playback(silence/1)  
same => n,Dial(SIP/202)  
same => n,Playback(silence/1)
```

Channels

- A Channel is a dynamic connection between a telephony device and Asterisk
- A phone connected to Asterisk is one channel
- Calling another phone spawns a new channel; the channels are linked or bridged together to create the familiar phone call

Asterisk development options

Modules

- Loadable extension modules written in C
- Subject to GPL
- Most flexible option
- Most of Asterisk's functionality is implemented in extension modules shipped with Asterisk
- Primarily for adding new functionality that does not exist

Asterisk Gateway Interface

- Dialplan AGI call spawns an external process or connects to a listening socket
- Can be written in any language that can handle stdin/stdout or sockets
- One AGI session per call; no cross-channel state or server awareness
- Command set targeted toward interaction

Asterisk Manager Interface

- Client connects to Asterisk on the AMI port
- Global events for the Asterisk system are sent to the AMI client
- AMI client issues actions to Asterisk at any time; Asterisk issues responses
- Command set targeted toward server/channel management

AMI traffic example

Action: Login
Username: manager
Secret: secret

Response: Success
Message: Authentication accepted

Event: FullyBooted
Privilege: system,all
Status: Fully Booted

Event: PeerStatus
Privilege: system,all
ChannelType: SIP
Peer: SIP/201
PeerStatus: Unregistered
Cause: Expired

Event: PeerStatus
Privilege: system,all
ChannelType: SIP
Peer: SIP/201
PeerStatus: Registered
Address: 172.20.64.1:64142

Event: Newchannel
Privilege: call,all
Channel: SIP/201-0000000a
ChannelState: 0
ChannelStateDesc: Down
CallerIDNum: 201
CallerIDName: 201
AccountCode:
Exten: 401
Context: default
Uniqueid: 1395325658.10

Which one will you
choose?

AsyncAGI

- Dialplan AGI call to `agi:async` emits an AMI event and waits for the AMI client
- AGI commands are sent as AMI actions
- Everything continues to run over the AMI socket; no extra processes or listening sockets needed
- But you need a good way to demultiplex all the events, actions, and responses...

Enter Twisted

Twisted

- Python's venerable asynchronous communications library—since 2002
- Includes many common protocols and building blocks for more protocols
- Easy to move between callback-driven and “inline callback” styles as needed

Twisted terminology

- Reactor: supplied by Twisted, calls functions and methods when data is available
- Deferreds: returned by a function when a result is not immediately available; can have callbacks and err backs attached to it
- Deferred chaining: when a callback itself returns a Deferred—highly optimized in Twisted
- Example: [AMIProtocol.loginMD5](#) from `ami.py`

octothorpe

- An AMI protocol implementation for Twisted
 - AMIProtocol: handles events, actions, and responses, spawns Channels and routes events appropriately
 - AsyncAGIProtocol: all of the above plus support for driving calls via AsyncAGI
- Uses Deferreds to turn actions and responses into function calls

That's enough talk, on
to the demos

Follow along with me

- <https://github.com/zigg/octothorpe>
- All examples demoed here are in `doc/examples`
- If you have recently used the Vagrant box `nrel/CentOS-6.5-x86_64`, you can even run the demos yourself... just `vagrant up`

The setup

- Asterisk from Digium's RPM repositories running inside a CentOS 6 virtual machine
- Minimal configurations (you can see these in etc/asterisk in the source)
- Telephone.app (free in the Mac App Store) connected via host-only network as extension 201
- Digium SIP desk phone connected via bridged network as extension 202

Dialplan

- `etc/asterisk/extensions.conf`
 - 401: say “hello, world!” and hang up
 - 402: dial extension 202 (desk phone), hang up when done
 - 403: go into AsyncAGI mode

amiwatch

- Watch events fly by on the Asterisk Manager Interface
- Handles connection and login
- Doesn't *do* anything with calls—we'll use existing Asterisk dialplan for that

chanwatch

- amwatch plus Channel spawning
- Shows you how events get routed to Channels
- Shows you some of the built-in functionality of the Channel object, like accumulating channel variables

agihello

- First interactive call demo
- When AsyncAGI starts we:
 - Answer the call
 - Say “Hello, world!”
 - Hang up the call

orighello

- Same as agihello, except in the opposite direction
- Originate is used to spawn a call and then execute the AGI call to go into AsyncAGI mode on that channel when answered
- No dialplan required

dtmf

- Listens for and speaks back DTMF using the `captureDTMF` function
- Introducing: inline callbacks!

ivr

- The biggest example of them all
- Passcode required
- Option 1: call another extension
- Option 2: hang up

Other applications

- DTMF injection
 - External process makes an RPC call with an extension number and a DTMF digit; if that extension number exists the DTMF is injected on the line to control a device
- Emergency Monkey System
 - 100 virtual hotlines calling 100 virtual agents and screeching monkeys at each other

Wrapping up

Further reading

- Asterisk: The Definitive Guide: <http://asteriskdocs.org/>
- Twisted: <http://twisted.readthedocs.org/>