

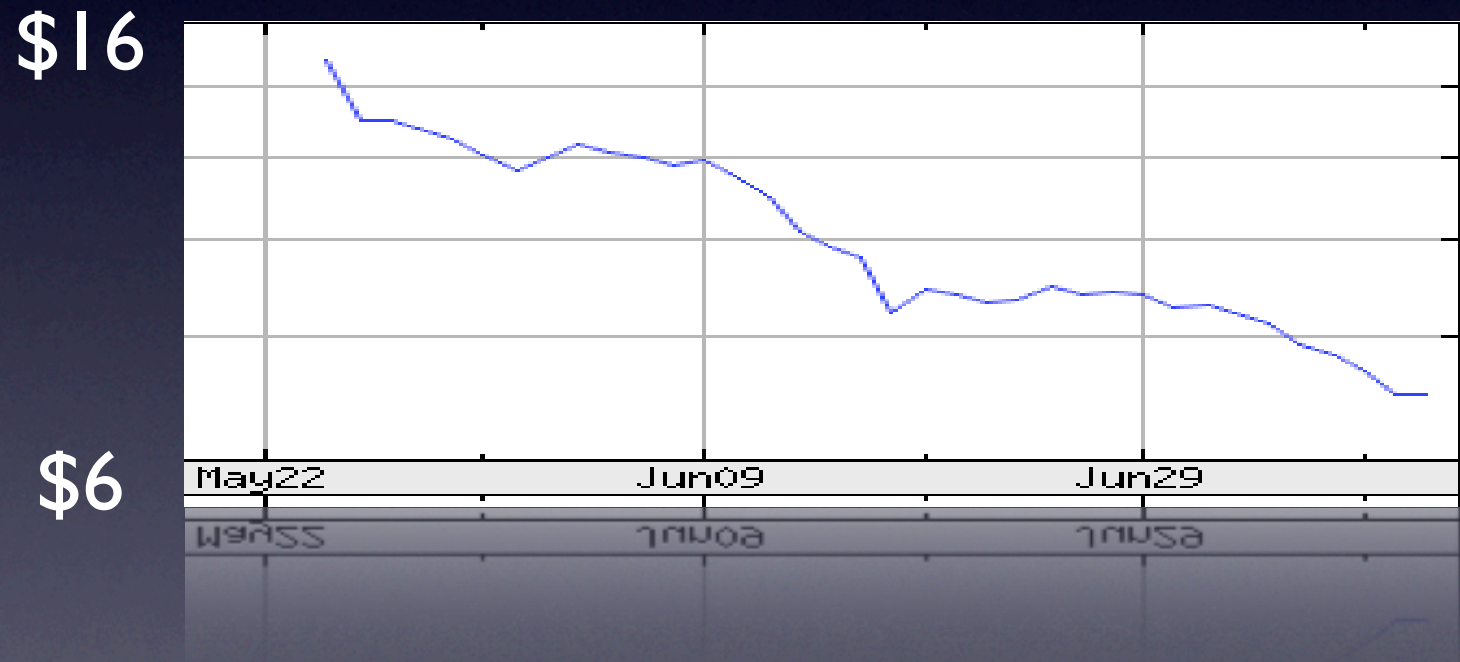
# respond\_to :voice

How rails and asterisk enable the voice interface revolution

Jonathan Palley  
[jpalley@idapted.com](mailto:jpalley@idapted.com)

# VoIP...The Next Big Thing?

The Promise: Cheap telephony make major telecoms irrelevant



Vonage: Losing money on every customer

Why is VoIP still so  
great?

# Voice Services and Interfaces

Web: **Rails**



Great Opportunities

VoIP: **Asterisk**

# The Plan

- Brief Overview of VoIP Technologies
- A look at existing tools
- Where I think we should go + demo

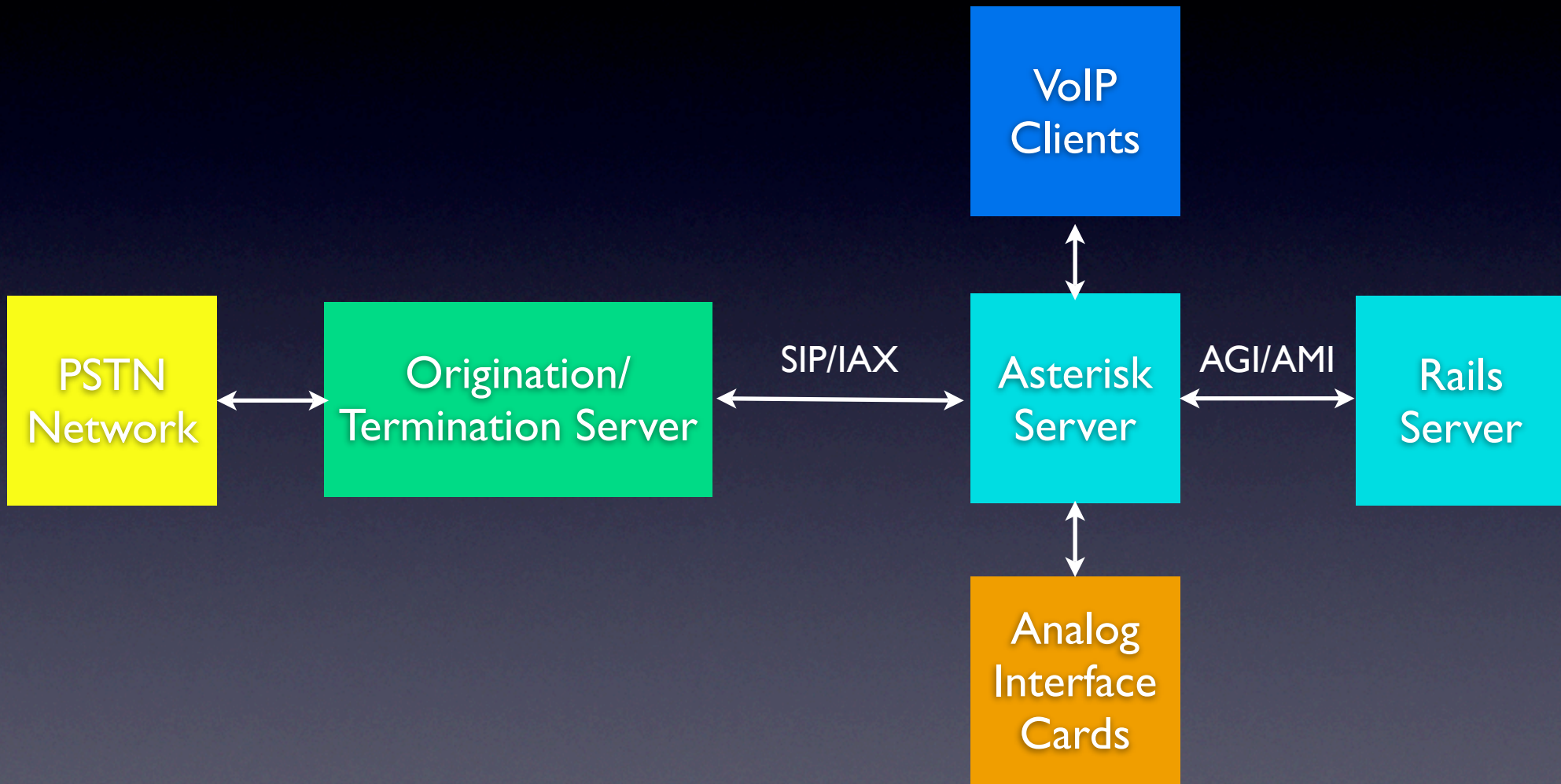
# VoIP/Web Concepts

Technology	Web	VoIP
Protocol	HTTP FTP RTP etc	SIP - Ind. Standard IAX - Best(?) H.323 - Obsolete Jingle - Upcoming? <del>Skype - Proprietary</del>
Codec	gzip jpg/gif/etc. wmv/rm/etc	g.711 - high bandwidth gsm - medium bandwidth g.729 - low bandwidth iLBC - low bandwidth
Server	apache/lighttpd	asterisk
Interactivity	CGI	AGI AMI

# Key Asterisk Concepts

- Channels - connections asterisk has with phones/other servers/etc.
- Dialplan - what happens when someone calls? Play a sound, connect to channel, etc.
- AGI - Simple stdin/out or TCP socket method in which scripts can interact with dialplan
- AMI - Method through which programs can interact with the Asterisk CLI

# Setup



# RAGI: Overview

- Uses ActionController like CallHandler
- Four classes: CallServer, CallHandler, CallConnection and CallInitiate
- Integrates with Rails Models
- Scripts written in linear command line fashion

# RAGI: Code

```
1  class SimonHandler < RAGI::CallHandler
2    APP_NAME = 'simon'
3    def dialup
4      play_sound("simon-welcome")
5
6      intArray = [rand(10)]
7      score = 0
8
9      # test the user's memory on the current int array
10     while memory_test(intArray)
11       score += 1
12       # grow the array by one digit
13       intArray << rand(10) # choose a number between 0 and 9 inclusive
14       wait(1)
15     end
16
17     score
18   end
19
20   def memory_test(intArray)
21     # say each number in the array
22     intArray.each do |number|
23       play_sound("simon-#{number}") # there are sounds like simon-6.gsm
24     end
25
26     resultStr = get_data("simon-beep", (intArray.length * 1500), intArray.length)
27
28     # Check if what they entered matches the int array
29     if (resultStr == intArray.join(""))
30       return true
31     else
32       play_sound("simon-gameover")
33       return false
34     end
35   end
36 end
```

# RAMI Overview

- Connects to AMI Interface
- Good for catching & initiating calls/acting on lines in non-linear fashion/receive events
- No rails integration

But what happens if you are developing a web/voice application?

- DHH RailsConf Keynote: Mime Types are Big
- Voice = Mime Type
- `respond_to` seems to be the way to go

# Introducing RAI - The Rails Asterisk Integrator

- Ok, it needs a better name.
- Make it easy to write voice interfaces/services for web apps.
- Built on top of RAGI (and eventually RAMI)

# Key Ideas

- Use `respond_to` and integrate right in `ActiveController`
- Just another view DSL (like RJS)
- Link calls to web sessions
- Route AMI events to controller actions using `routes.rb`
- Problems: `respond_to` gets messy/repetitive  
- looking for better way with same idea.

# Banking Demo

(will be available shortly along with rai, at <http://ww.idapted.com/developer>)

## Prototypical Code Snippet:

```
def index
  @balance=balance

  respond_to do |wants|
    wants.html {
      render
    }
    wants.voice{
      render_asterisk{|a|
        a.say_sequence("bank-welcome bank-account-balance #{say_amount(@balance)}")
        a.link_to_dtmf 'bank-index-menu', :links=>[{:digits=>1, :action=>"new"},
{:digits=>2, :action=>"list"}]
      }
    }
  end
end
```

# GET INVOLVED!

We need to get a community working on the key elements (RAGI/RAMI/RAI) of this stack...sharing problems, ideas, etc.

PLEASE contact me if you are interested:

[jpalley@idapted.com](mailto:jpalley@idapted.com)

Will be setting up a wiki/email list/etc. shortly!