digium
The Asterisk Company

Asterisk and Node.js

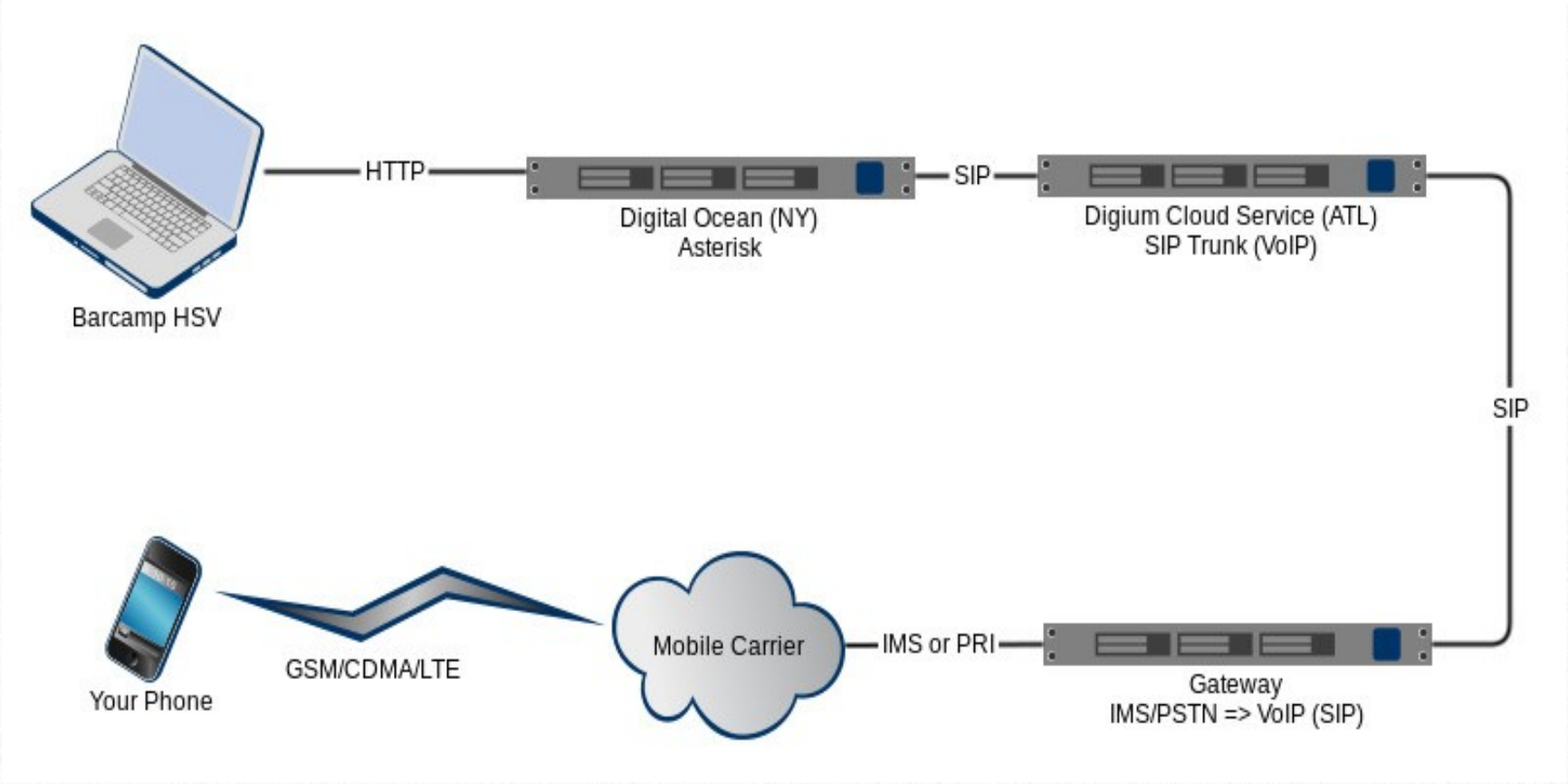
Matt Jordan
@mattcjordan
Director of Technology, Digium

Prisoner's Dilemma Conference

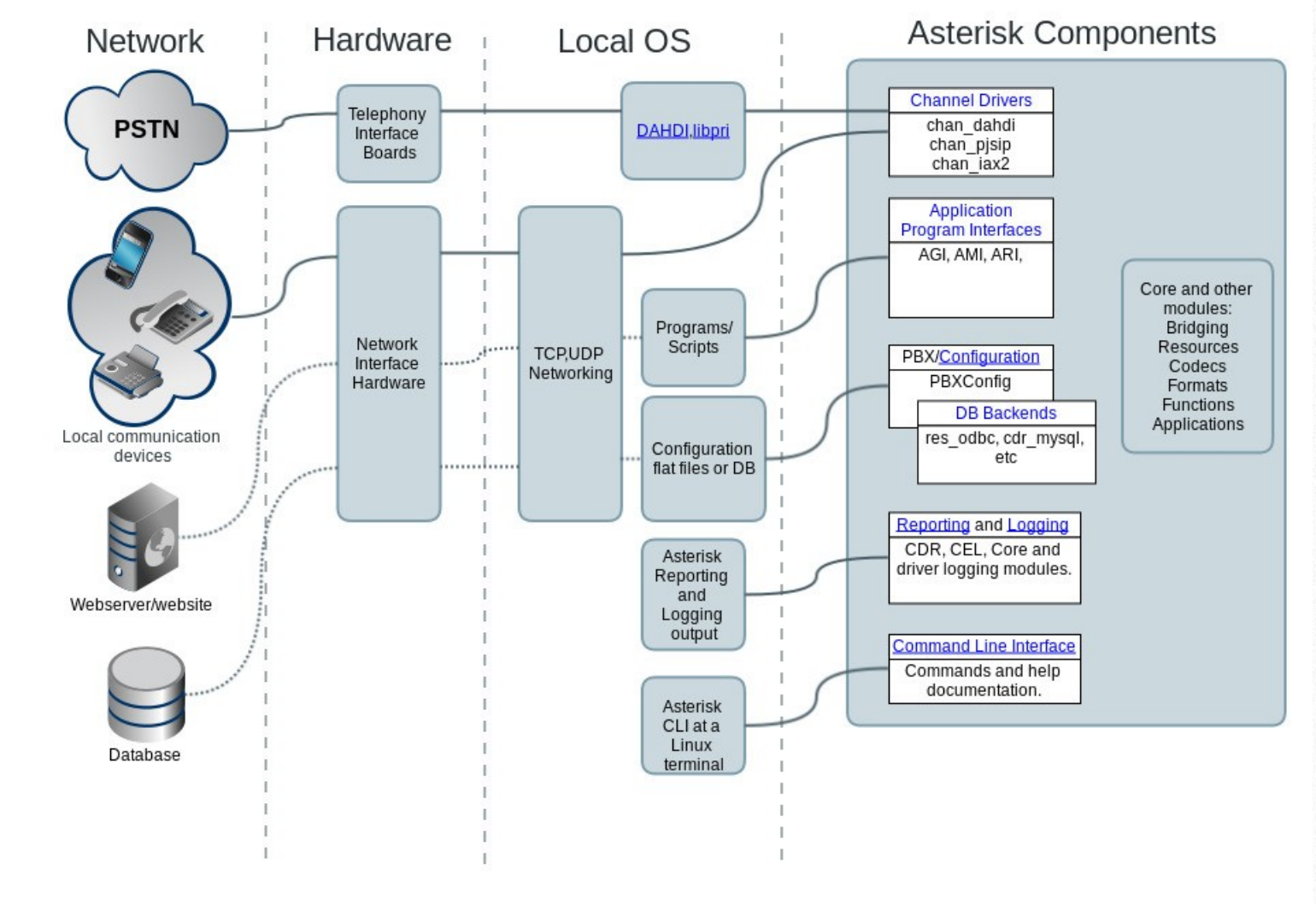
digium
The Asterisk Company

256 970-2858

Setup


HTTP
Digital Ocean (NY)
Asterisk
SIP
Digium Cloud Service (ATL)
SIP Trunk (VoIP)
Barcamp HSV
SIP
Your Phone
GSM/CDMA/LTE
Mobile Carrier
IMS or PRI
Gateway
IMS/PSTN => VoIP (SIP)

What is Asterisk?



Traditional Asterisk

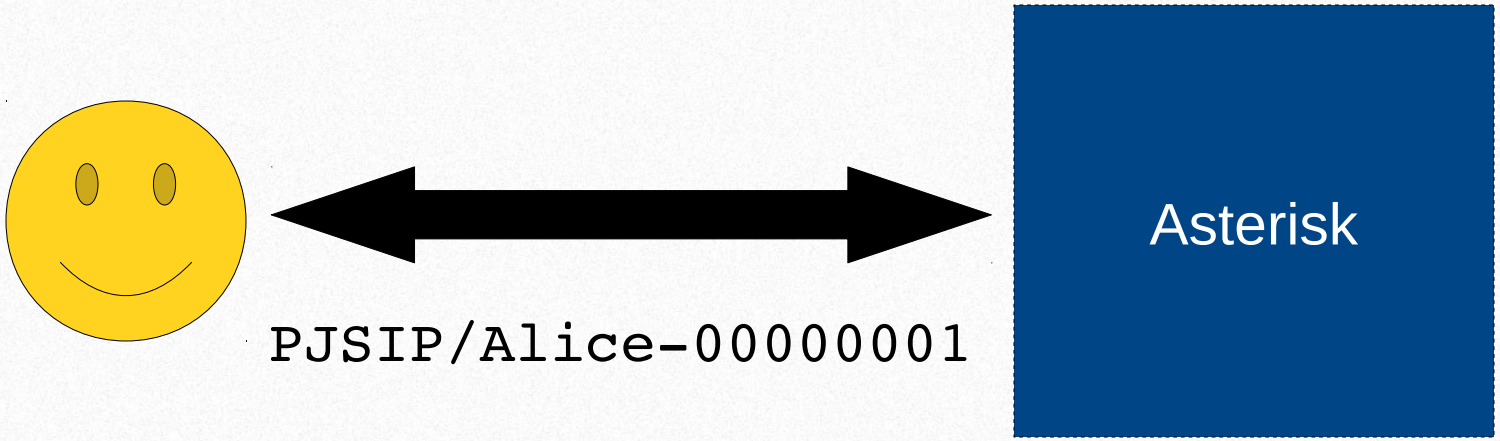
```
<html>
  <head>
    <title>Hello World Demo</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

```
exten => 100,1,Answer()
exten => 100,n,Wait(1)
exten => 100,n,Playback(hello-world)
exten => 100,n,Hangup()
```

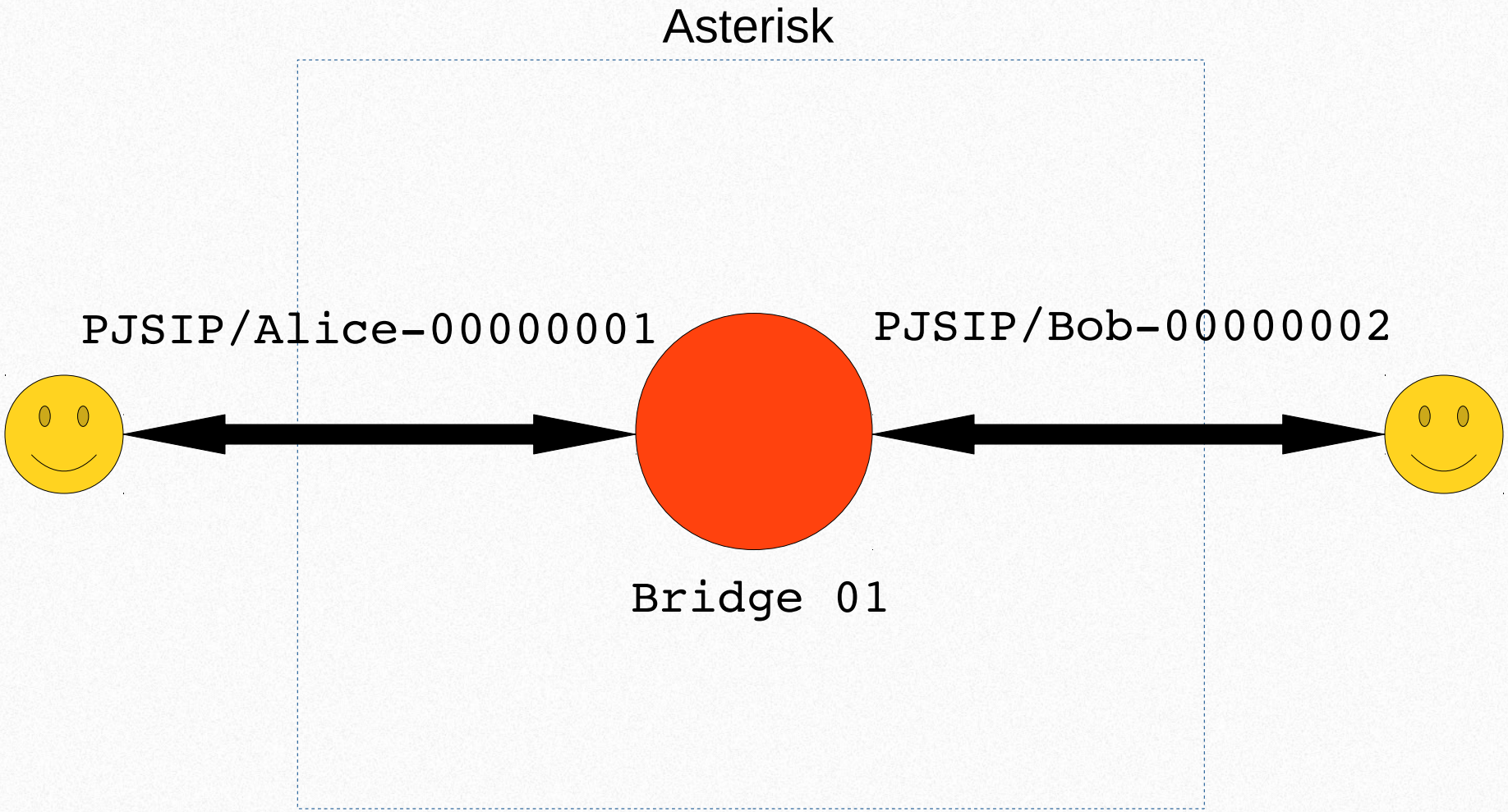
ARI: Asterisk REST API

- RESTful API
  - Communication is asynchronous
  - HATEOAS doesn't work well
- Three pieces
  - REST API: Asterisk primitives
  - Websocket: Events
  - Dialplan application to hand off the channel
- Build your own communication app
  - Any language (HTTP)
  - Asterisk as an Engine

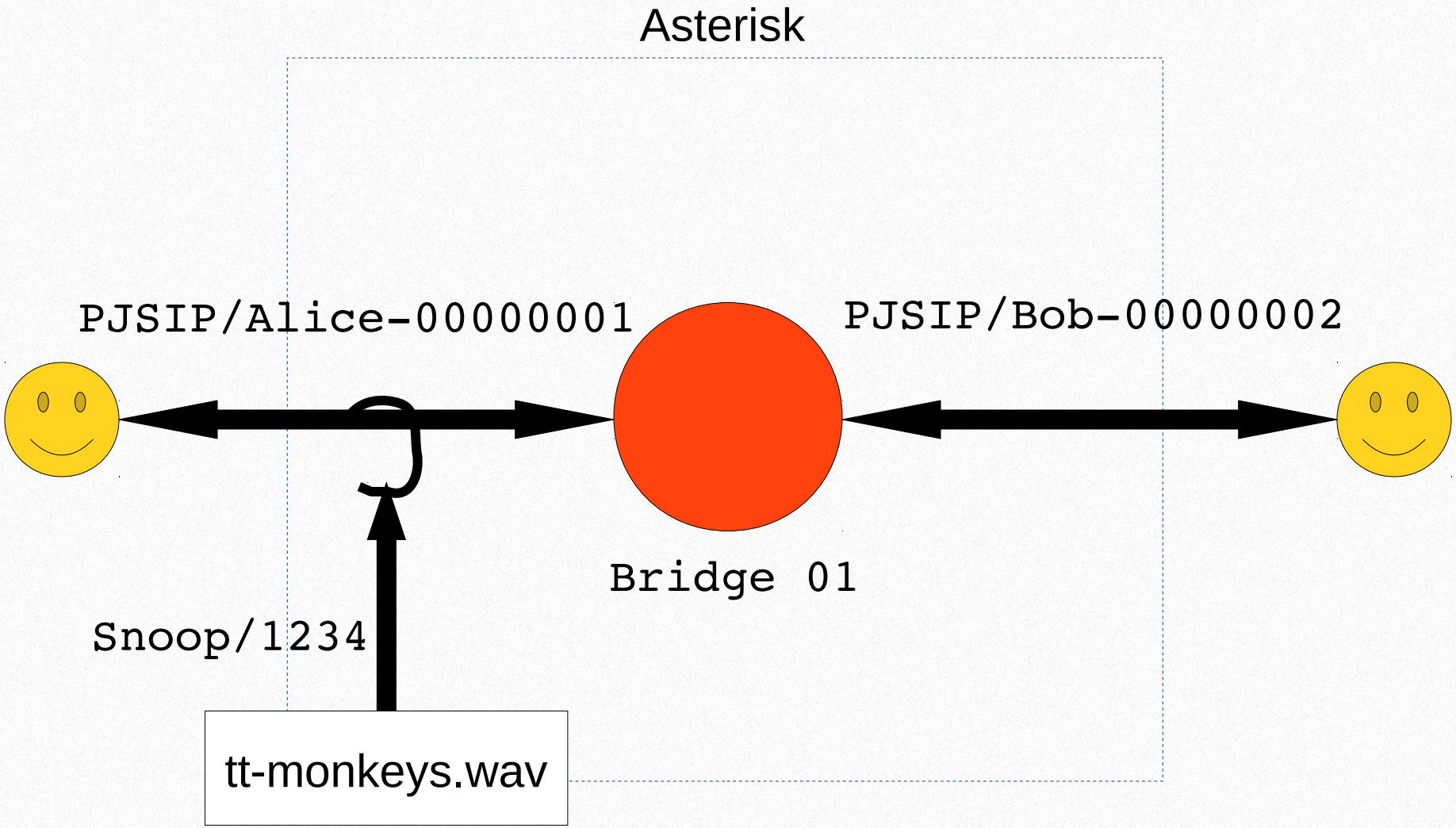
Asterisk Primitives: Channels



Asterisk Primitives: Bridges



Asterisk Primitives: Media



ARI Examples

- `POST http://localhost:8088/ari/channels?endpoint=PJSIP/Alice&app=demo-conf`
- `POST http://localhost:8088/ari/bridges/1234/addChannel?channel=12983.1`
- `GET http://localhost:8088/ari/bridges`
- `POST http://localhost:8088/ari/channels/12983.1/play?media=sound:tt-monkeys`
- `POST http://localhost:8088/ari/playbacks/12345/stop`

node-ari-client

- JavaScript wrapper around REST API
  - `channel.answer()`
  - `bridge.addChannel({channel: channelId});`
- Dispatching of WebSocket events
  - `channel.on('StasisStart', function (event, channel));`

https://github.com/asterisk/node-ari-client

Why Node.js?

- Designed for asynchronous applications
  - Communication is asynchronous
  - Events typically drive communication apps
- REST APIs + Node.js
  - Easily parseable
  - Asterisk uses Swagger
- Scalability
  - Logic bogs Asterisk down
  - Logic will often scale differently than media operations

Conference App

```
ari.connect('http://{address}:8088', 'asterisk',
    '{password}')
.then(function (client) {
  var conference;

  client.on('StasisStart', onStasisStart);
  // Used only to show when a channel hangs up
  client.on('StasisEnd', onStasisEnd);

  console.log('Starting...');
  client.start('conf-demo');
});
```

Conference App

```
function onStasisStart(event, channel) {

  return getOrCreateConference()
    .then(function () {
      channel.on('ChannelDtmfReceived',
        onDtmfReceived);

      channel.answer();
    })

    ...

}
```

Conference App

```
function onStasisStart(event, channel) {

    ...
    .then(function () {
      return conference.addChannel(
        { channel: channel.id });
    })
    .then(function () {
      return conference.play(
        { media: 'sound:beep' });
    });

}
```

Conference App

```
function onDtmfReceived(event, channel) {
  if (event.digit !== '#') {
    return;
  }

  // Pick a random channel to play monkeys to
  .then(function (prisoner) {
    return prisoner.snoopChannel({
      whisper: 'out',
      app: 'conf-demo'});
  });
}
```

Conference App

```
function onStasisStart(event, channel) {

    if (channel.name.substring(0, 5) ===
        'Snoop') {
      return channel.play({
          media: 'sound:tt-monkeys'})
      .then(function (playback) {
        playback.on('PlaybackFinished',
          function (event, playback) {
            channel.hangup();
        });
      });
  ...
```

Questions?

https://github.com/matt-jordan/conf-demo