19 Wonderful Years of Asterisk

Matthew Fredrickson

@creslin287

Personal Background

Who are you and what have you done with Matt Jordan!!?

- **Worked at Digium since 2001 in various developmental capacities**
- **Worked on Asterisk at different times**
- **Maintained libpri and DAHDI for many years**
- **Wrote an SS7 stack for Asterisk (libss7)**
- **Worked on WebRTC related initiatives for the last few years**
- **Now Project Lead of the Asterisk project**

A little bit of history

Asterisk 1.0 - First major release, ISDN support, H.323, MGCP, AGI, SIP

[many changes later]

Asterisk 1.6 - Wideband audio, SS7 support

[…some time passes…]

A little bit of history

Asterisk 11 - Beginnings of WebRTC support in chan_sip

Asterisk 12 - chan_pjsip

Asterisk 13 - ARI, more PJSIP

Asterisk 14 - More ARI, more PJSIP, and Async DNS

A little bit of history (cont'd)

What about Asterisk 15?

Today's Topics

- **BUNDLE**
- **Unified Plan**
- **RTP Layer Improvements**
- **New Video Features in Asterisk**

WebRTC Technologies - BUNDLE

What on earth is BUNDLE and why do I care?

- **A long time ago in a galaxy far, far away….**
- **Audio and video media for a call is exchanged using a protocol called RTP (Real Time Protocol).**
- **Traditionally, audio RTP and video RTP for a phone call has been carried on separate UDP ports.**
- **Inherent information about those streams was logically separated and not explicitly connected at a transport level.**
- **Example: UDP port 10000 for audio and port 10002 for video**

WebRTC Technologies - BUNDLE

What on earth is BUNDLE and why do I care?

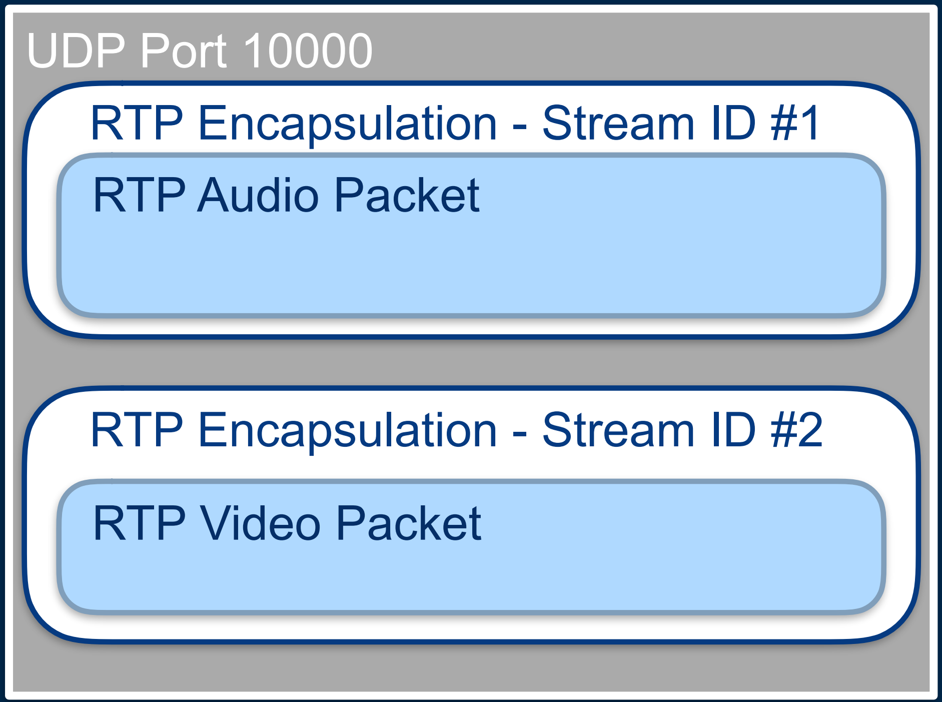
- WebRTC developers realized that for connecting browsers behind NAT, it was better to reduce the number of UDP ports required to be connected for a call to be established.

They asked: How could we accomplish this?

WebRTC Technologies - BUNDLE

What on earth is BUNDLE and why do I care?

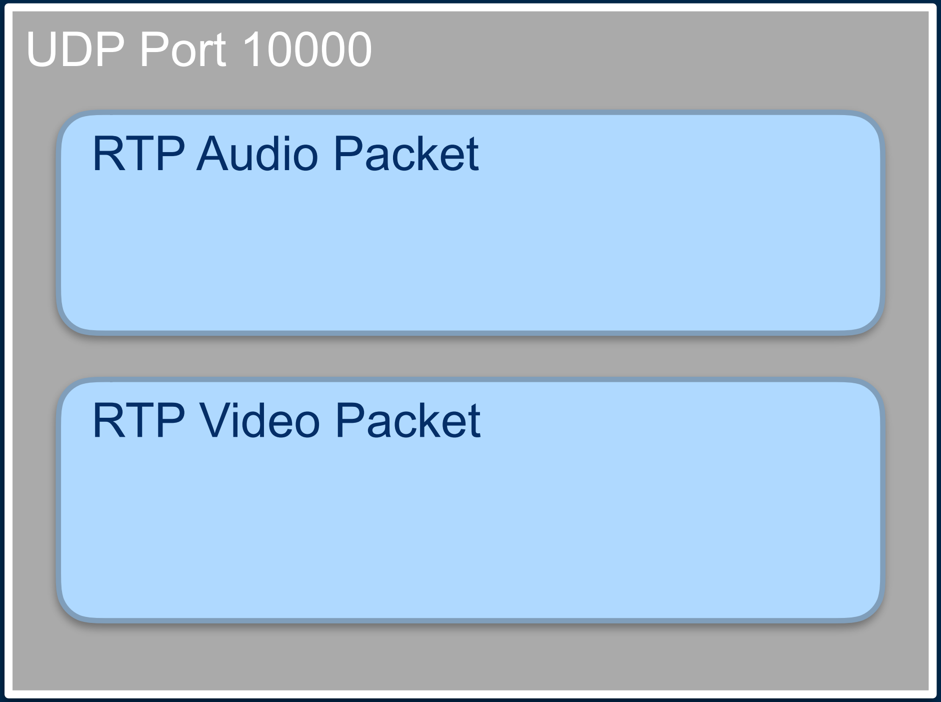
Option 1: Add an additional encapsulation protocol around the RTP messages to explicitly identify and multiplex multiple RTP streams on the same UDP port.



WebRTC Technologies - BUNDLE

What on earth is BUNDLE and why do I care?

Option 2. Just dump both RTP streams on the same port and use RTP specific heuristics to identify which RTP stream is audio and which one is video.



WebRTC Technologies - BUNDLE

What on earth is BUNDLE and why do I care?

- **In an effort to better preserve transport layer compatibility with existing RTP endpoints and devices, the second option was chosen.**
- **No additional encapsulation and labeling was necessary since sufficient implicit information was already available in the RTP stream itself.**
- **Asterisk 15 now supports BUNDLE to better interoperate with browser based endpoints.**

WebRTC Technologies - Unified Plan

What on earth is Unified Plan and why do I care?

- **WebRTC 1.0 uses SDP (Session Description Protocol) to represent media session attributes.**
- **SDP is a text based description of the types and attributes of media streams within a call.**
- **SDP is used by SIP, MGCP and possibly other protocols to transmit that information.**
- **SDP had limitations when describing multiple streams (particularly video streams) and some newer attributes**

WebRTC Technologies - Unified Plan

A few different proposals were given as options to solve SDP's limitations:

- Plan B - Currently implemented in Chrome and other endpoints, first cut at solving the problem.
- Unified Plan - Final ratified solution by WebRTC working group. Implemented in Firefox, and soon to be implemented in Chrome.

WebRTC Technologies - Unified Plan

Important things to realize:

- **Asterisk 15 implemented the Unified Plan, since it's the way forward for the WebRTC standards**
- **For existing browsers (Chrome) that don't speak Unified Plan yet, there are SDP translators available that convert between Plan B and Unified Plan**
- **As a developer, you'll need to include an SDP translator in your browser stack in case you're running on a browser that only supports Plan B.**
- **Included with Digium's sample browser client, and integrated with JSSIP stack.**

WebRTC Technologies - RTP layer improvements

RTP sequence number gap preservation:

- **In RTP, every packet in the media stream is given a sequence number.**
- **The sequence number is monotonically incrementing (should increase by one on every packet received)**

WebRTC Technologies - RTP layer improvements

If there are sequence number gaps in the incoming RTP stream, it usually means one of two things:

1. Packet loss (Packet 1, 2, 3, 5, 6)

2. Packet re-ordering (Packet 1, 2, 4, 3, 5)

WebRTC Technologies - RTP layer improvements

RTP Sequence Number Gaps:

- **For audio streams (where asterisk has historically spent most of its time) sequence number gaps have been less noticeable. Audio streams seem less impacted by that loss of information.**
- **For video codecs, we discovered that this was a big problem. A single lost or reordered packet can cause huge quality deterioration.**
- **Asterisk now preserves end to end sequence number gaps and relative packet ordering so that codecs on endpoints can act intelligently with that information.**

The historical problem with video

- Limited clients: commodity SIP video phones and SIP desktop clients have traditionally have been single stream limited.

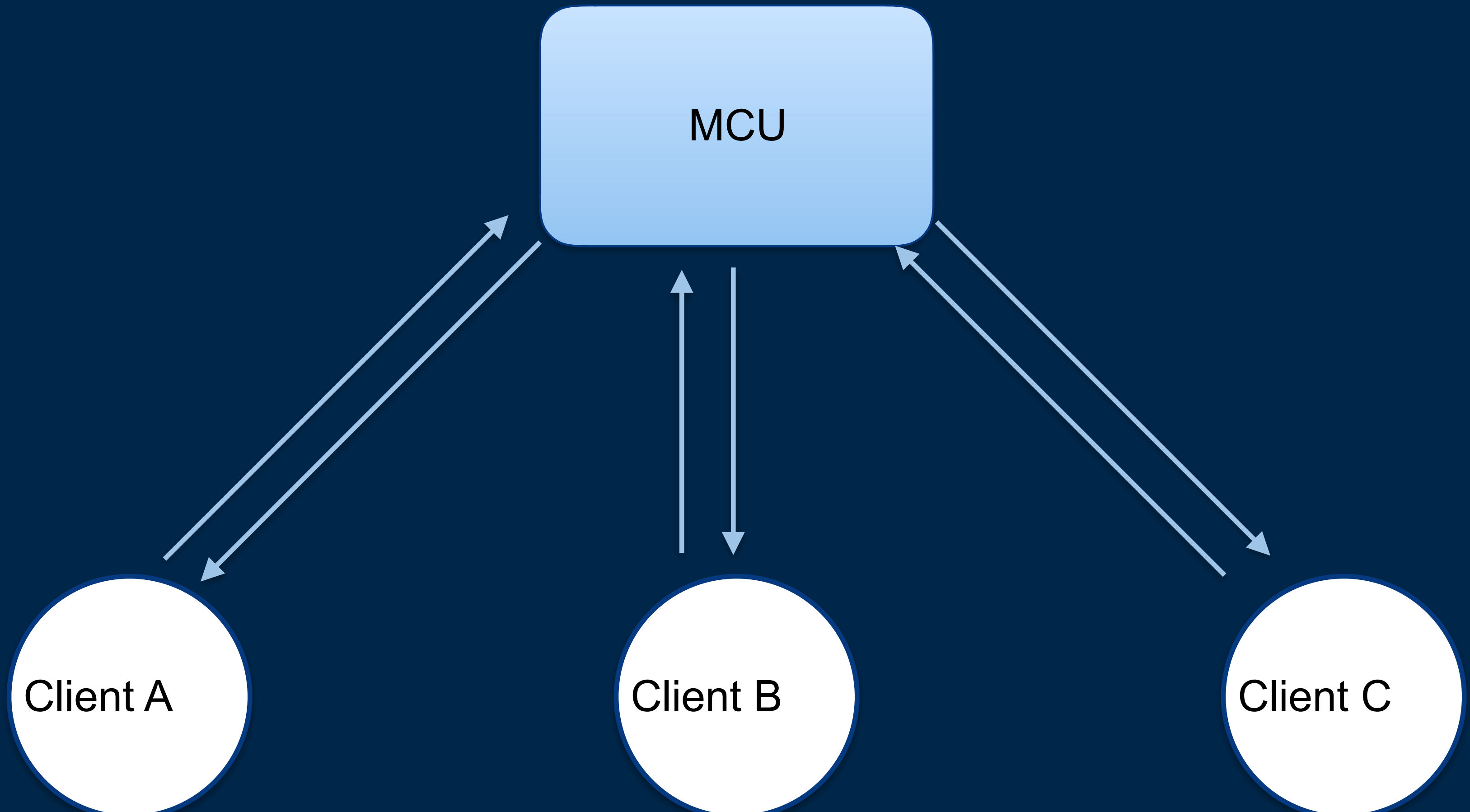
- Premium clients with rich end user experience were historically large and very expensive (think Cisco telepresence, Tandberg solutions, etc)

- Video MCU (multipoint control unit) was considered the defacto standard for providing a multi-user experience (so mix everybody's video stream into a brady bunch stream)

- Requires lots of CPU power on the MCU mixing box.
"Video is expensive, yo?" - file

What is an MCU (Multipoint Control Unit)?

N participants, each sending one video stream and receiving 1 pre-mixed video stream back from the MCU.



Video gets better - WebRTC technology

- Rich video clients started to get less proprietary and more powerful (browsers)

- Browsers can receive and decode multiple separate video streams at once.

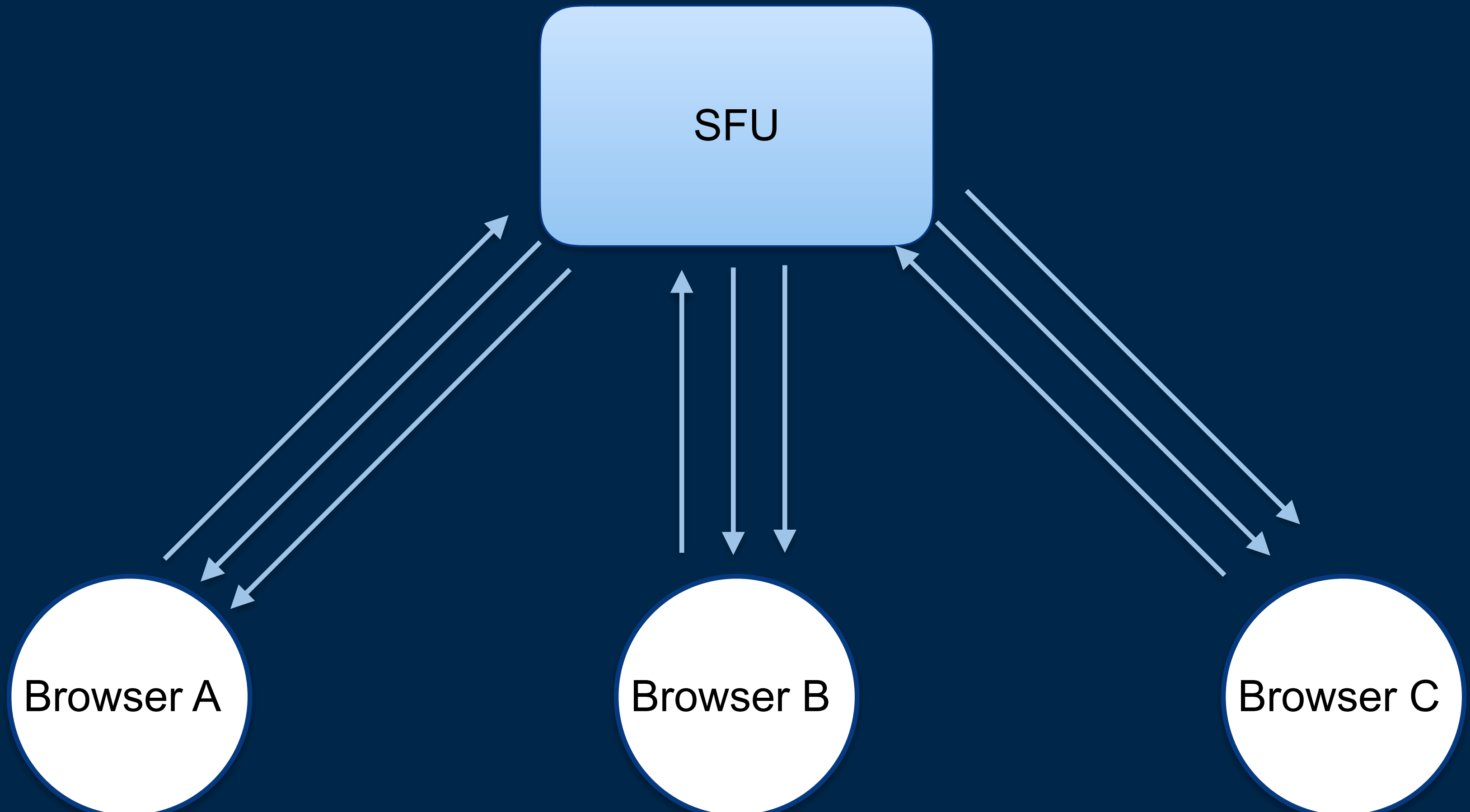
- Rich front end rendering language (HTML+CSS+javascript)

- Video for the common man

- SFU conferencing approach becomes the new norm

What is a SFU (Selective Forwarding Unit)?

N participants, each sending one video stream and receiving N-1 video streams from other participants.



Why SFU?

- If one participant wants to have large picture focus on the presenter for a particular period of time, they can.
- If another participant is still taking notes from a screen share, they control their focus (so they can keep the screen share large for an additional period of time if desired)
- Much less CPU usage on middle box

Asterisk 15 - Video streams for the common man

Asterisk 15 does video better than any prior version of Asterisk:

- **Multi stream enhancements to the core of Asterisk - the old single-video/single-audio stream per call limitation is broken.**
- **Asterisk core allows renegotiation of number of video streams and audio streams as well as their attributes on demand.**
- **app_confbridge now has support to be a generic SFU (selective forwarding unit) - All video streams go to all participants**

How can I try it?

1. Get a copy of Asterisk 15 built and installed

2. Get a copy of CyberMegaPhone2000:

https://github.com/asterisk/cyber_mega_phone_2k

3. Setup asterisk's http/websocket server in http.conf:

```
[general]
enabled=yes
bindaddr=0.0.0.0
bindport=8088
tlsenable=yes
tlsbindaddr=0.0.0.0:8089
tlscertfile=/etc/asterisk/keys/asterisk.pem
```

How can I try it?

4. Setup pjsip.conf:

[transport_wss]

type=transport

bind=0.0.0.0

protocol=wss

How can I try it?

4. Setup pjsip.conf (cont'd):

[guest]

type=aor

max_contacts=5

How can I try it?

4. Setup pjsip.conf (cont'd):

```
[guest]
type=endpoint
context=default
direct_media=no
allow=!all,ulaw,vp8,h264
aors=guest
max_audio_streams=10
max_video_streams=10
webrtc=yes
dtls_cert_file=/etc/asterisk/keys/asterisk.pem
dtls_ca_file=/etc/asterisk/keys/ca.crt
```

How can I try it?

5. Setup confbridge.conf:

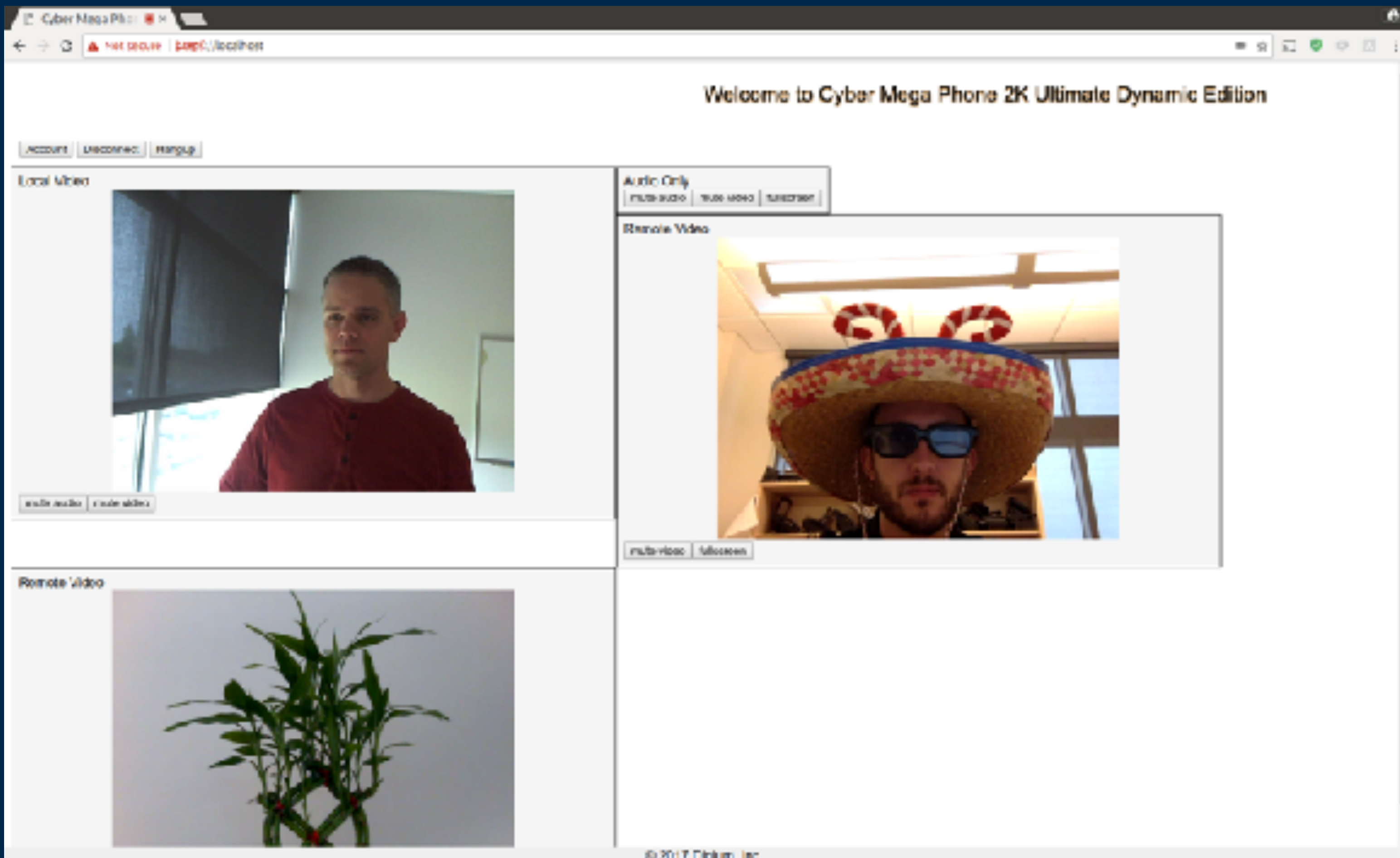
```
[general]

[default_bridge]
type=bridge
video_mode=sfu

[default_user]
type=user
music_on_hold_when_empty=yes
music_on_hold_class=default
```

How can I try it?

6. Setup extensions.conf:

```
[default]
exten => video-conference,1,Answer()
exten => video-conference,2,ConfBridge(guest)
exten => video-conference,3,Hangup()
```

How can I try it?



How can I try it?

Potential gotchas:

- **Generating the certificate files for Asterisk:**

https://wiki.asterisk.org/wiki/display/AST/WebRTC+tutorial+using+SIPML5

See the section labeled “Create Certificates”

- **For any other difficulties and issues, see the blog post at:**

http://blogs.asterisk.org/2017/09/20/asterisk-15-multi-stream-media-sfu/

How can an ARI developer utilize SFU mode?

When creating a bridge, add the option at creation time (in your language of choice):

type='video_sfu'

So using the python client:

```
client = ari.connect('http://localhost:8088', 'asterisk', 'asterisk')

bridge = client.bridges.create(type='video_sfu')
```

What will YOU build today?

What else is new in Asterisk 15?

- Platform Improvements
- Miscellaneous Other Improvements
- Video, WebRTC, and more, Oh My!

Platform Improvements

- GCC 7 fixes
- Build fixes for FreeBSD when missing crypt.h
- Build fixes for the Gnu HURD
- Alembic support for MS-SQL
- PJPROJECT bundled support is enabled by default

Miscellaneous Other Improvements

- New Asterisk sounds release (1.6)

- Google OAuth 2.0 protocol support for XMPP/Motif

- Binaural audio support patches for confbridge were merged

- debug_utilties: ast_coredumper

- debug_utlities: ast_loggrabber

Video, WebRTC, and more, Oh My!

- Support for RTCP-MUX
- ‘webrtc’ endpoint option in res_pjsip.conf
- VP9 passthrough support
- ICE interface blacklist option added to rtp.conf

Asterisk 16 - What's next?

- Fleshing out Asterisk's SFU APIs
- Improving Asterisk's video resilience in poor network environments
- PJSIP performance improvement

Contribution Statistics for 15

Asterisk 15 contribution statistics:

- 924 Commits

- 82 Individual contributors (according to commit authorship)

General project statistics:

- Nearly 2400 merged code reviews on gerrit (for all branches) since DevCon last year.

Contribution Statistics for 15

Top contributors (by # of commits) outside of Digium:

```
104 Sean Bright
42 Corey Farrell
39 Alexander Traud
20 Alexei Gradinari
19 Tzafrir Cohen
15 Torrey Searle
11 Walter Doekes
```

Reminder

- 11 went EOL in October. No more security fixes or bug fix fixes. Get off that branch! (particularly if you run WebRTC)

- Asterisk 15 was not an LTS - but 16 should get us back on track and be the next LTS.

Thanks!

THANK YOU!

Follow me @creslin287 on twitter.

digium

The Asterisk Company

Empowering Communication

www.digium.com • www.asterisk.org

Project Background

Asterisk 11 (LTS) was released in October of 2012

Asterisk 12 was released in December of 2013

Asterisk 13 (LTS) was released in October of 2014

Asterisk 14 was released in September of 2016

Asterisk 15 was released in October of 2017

LTS versus Standard release

- LTS - Long term support

- LTS releases (11, 13) - bug fixes for 4 years, followed by 1 year of only security fixes.

- Standard (12, 14, 15) - bug fixes for 1 year, followed by 1 year of only security fixes.