Janus, SignalWire and FreeSWITCH

JanusCon 2019 - Luca Pradovera

A little introduction

Luca Pradovera

Sales Engineering Manager @ SignalWire

+ 20 years of RTC experience (from Flash to SIP to WebRTC, via XMPP!)
+ Pioneered the voice application approach at Mojo Lingo
+ Core contributor to the Adhearsion Ruby framework
+ Now bringing my experience to help disrupt the programmable telecoms industry

What is the presentation about?

David Duffett said this slide is important, so...

- What is FreeSWITCH?
- How can I use it with Janus?
- What is SignalWire?
- Tying it all together

FreeSWITCH 1.10, SignalWire, and much more

What is FreeSWITCH?

Helping people communicate since 2008

+ A free and open-source application server for real-time communication, WebRTC, telecommunications, video and VoIP

+ Built around a solid core, has a module-based approach to provide features

+ WebRTC(mod_verto), conferences (mod_conference), XML dialplans via HTTP (mod_xml_curl) and many others

+ Used as a PBX, as a switch, as an SBC, as a media server, as a WebRTC server...

State of the project

+ Version 1.10 was just released (August 2019), including:
  - Raspberry PI support
  - Debian Buster packages
  - New unit test framework
  - MariaDB module
  - Ability to run Javascript code in the background
  - Many, many more!

https://freeswitch.com/index.php/2019/08/19/freeswitch-1-10-0-release/

+ **SignalWire is the official sponsor of the FreeSWITCH project, that remains free and open source**

FS is moving to GitHub!

+ Stash wasn’t great, we know that!
+ Re-engage and re-energize the community
+ Required a lot of work on tooling, integrating double commits, internal process changes
+ Made possible by SignalWire
+ Contributions still drive the project, so come and help us!

FreeSWITCH and WebRTC

MCU vs. SFU

+ FreeSWITCH uses an MCU (**Multipoint Control Unit**) approach to conferencing

+ Video is mixed into a single stream, with layouts that can be defined according to many rules

+ Uses less resources on clients, more on server, suited for VoIP and mobile applications

+ Signaling is driven by Verto, a JSON-based protocol and related module

MCU video mode

Many streams in, one stream out

Example layout: 1_up_top_left+9

FreeSWITCH + JANVS WEBRTC GATEWAY

equals SUCCESS!

Complementary platforms

MCU vs. SFU - why not both?

+ Janus provides an **SFU**

+ FreeSWITCH provides video mixing, but also SIP interop and general RTC features

+ Both platforms speak **WebRTC** and **SIP**

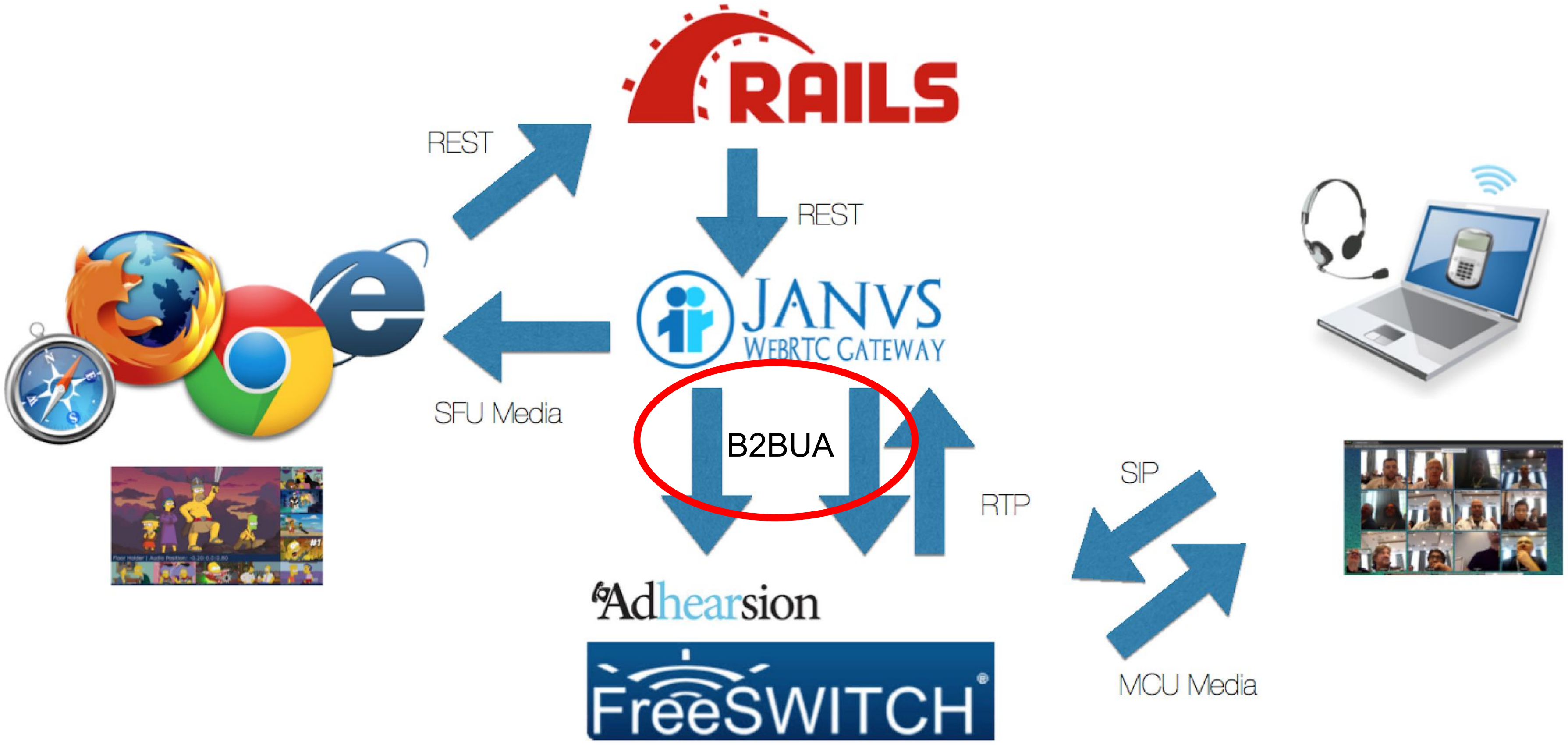
+ Signaling is different but it is always “just” JSON

Our use case

Hybrid MCU - SFU

+ Customer required SFU mode on desktop (for video quality) and MCU mode on mobile (for performance)
+ SIP bridging needed because of legacy conference room equipment
+ Recording the MCU with audio gives a ready-to-use rebroadcast format

+ **Solution: A B2BUA-like agent operating on WebRTC**


REST
RAILS
REST
JANVS
WEBRTC GATEWAY
SFU Media
B2BUA
RTP
SIP
Adhearsion
FreeSWITCH
MCU Media

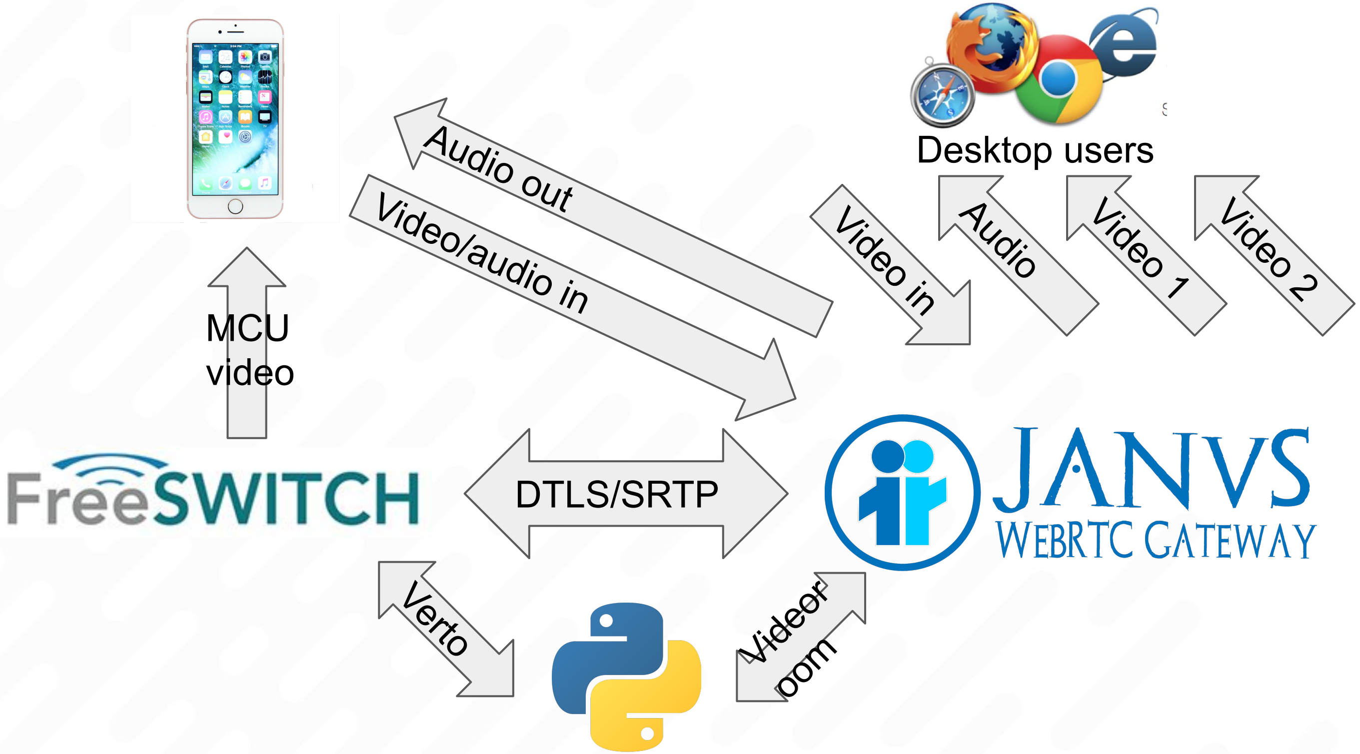
The implementation

Like Giacomo's, but in Python. There, done.

+ Python client connects to:
    - FreeSWITCH using Verto
    - Janus using the VideoRoom plugin as a subscriber only
+ Everyone joins the Janus room
+ B2BUA connects each stream to the FS conference
+ FS conference is available with MCU mode video and mixed audio

A better diagram

Call flows



Challenges

+ Managing participants is only easy in the current model
+ Still requires a lot of external state and orchestration
+ Solving complex NAT situations can be complicated
+ On the plus side, Python proved a great platform for this

What is next?

+ The architecture is very simple but there is no SIP access yet (subscriber only B2BUA)
+ As a result, that is much simpler and actually **works**
+ Next step is to add SIP via making the B2BUA a publisher
+ We are going to build that on...

SignalWire

What is SignalWire?

How we can help you succeed

What do we do?

Programmable communications

+ Cloud-based voice, messaging and video

+ SignalWire is a SaaS offering that brings the best of FreeSWITCH to the cloud, adding new features and at an unbeatable price

+ The company was founded because we believe in giving people innovative, disruptive features and not nick-and-diming them with pricing

+ One of the founders is Anthony Minessale, who also founded the FreeSWITCH project

Control APIs

Something new, something old

+ Relay
  - JSON-based Websocket control channel
  - Real time control with minimal latency
  - Event streams and handlers

+LāML - Legacy Antiquated Markup Language
  - XML-based webhook interface
  - Paired with a REST API for outbound calls and other interactions

Relay vs. LaML

Relay

```
{
                "node_id": "<UUID>",
                "call_id": "<UUID>",
                "control_id": "<ID>",
                "record": {
                    "audio": {
                        "beep": true|false,
                        "format": "mp3|wav...",
                        "stereo": true|false,
                        "direction":
"listen|speak|both",
                        "initial_timeout": 5.0,
                        "end_silence_timeout": 1.0,
                        "terminators": "#*"
                    }
                }
            }
```

LaML

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
    <Say>


    </Say>
    <Record
        action="http://your-application.com/record.php
        method="GET"
        maxLength="15"
        finishOnKey="#"
        />
  </Response>
```

SignalWire capabilities that Janus can leverage

+ SIP connectivity, PSTN numbers

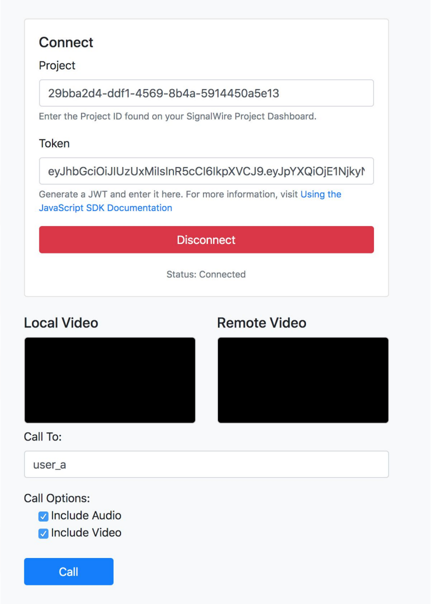
+ WebRTC conferences and 1-1 calls

+ Ability to `tap` into an arbitrary RTP stream and send it to an endpoint

+ Recording, playback, events

So, what’s working now?

Testing integration pathways



Coming soon

What's next for the Janus/SignalWire integration?

+ Making the Python connector release-able

+ Exploring the publisher connection possibilities

+ Making it as easy as possible for Janus to use SignalWire features

+ ...RIPP? (in the FAR future!)

SignalWire

Thank you!

Luca Pradovera
luca@signalwire.com