FOSDEM 20

Janus as a WebRTC "enabler"

Having fun with RTP and external applications

Lorenzo Miniero

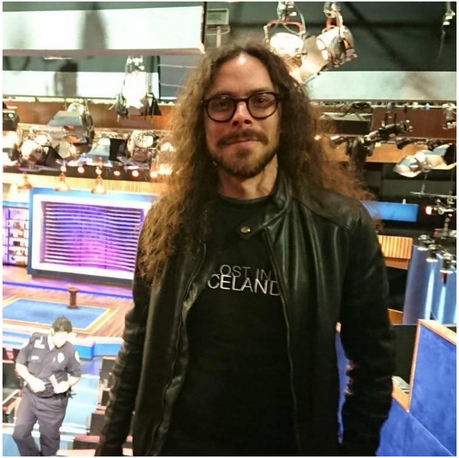
@elminiero

FOSDEM 2020 Real Time devroom

2nd February 2020, Brussels

Who am I?

FOSDEM 20



Lorenzo Miniero

- Ph.D @ UniNA
- Chairman @ Meetecho
- Main author of Janus

Contacts and info

- lorenzo@meetecho.com
- https://twitter.com/elminiero
- https://www.slideshare.net/LorenzoMiniero
- https://soundcloud.com/lminiero

Remember Janus?

Janus

General purpose, open source WebRTC server

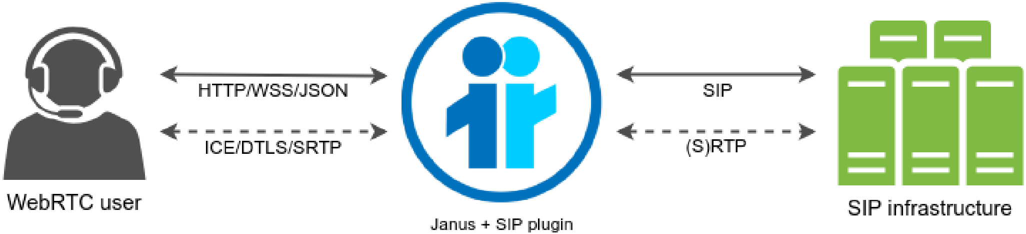
- https://github.com/meetecho/janus-gateway
- Demos and documentation: https://janus.conf.meetecho.com
- Community: https://groups.google.com/forum/#!forum/meetecho-janus

A quick recap: modular architecture

FOSDEM 20

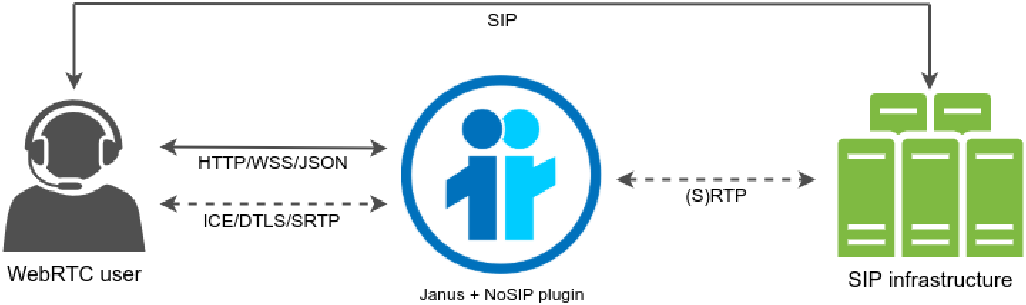
- The core only implements the WebRTC stack
  - JSEP/SDP, ICE, DTLS-SRTP, Data Channels, ...
- Plugins expose Janus API over different “transports”
  - Currently HTTP / WebSockets / RabbitMQ / Unix Sockets / MQTT / Nanomsg
- “Application” logic implemented in plugins too
  - Users attach to plugins via the Janus core
  - The core handles the WebRTC stuff
  - Plugins route/manipulate the media/data
- Plugins can be combined on client side as “bricks”
  - Video SFU, Audio MCU, SIP gatewaying, broadcasting, etc.

Exchanging RTP (1): SIP plugin



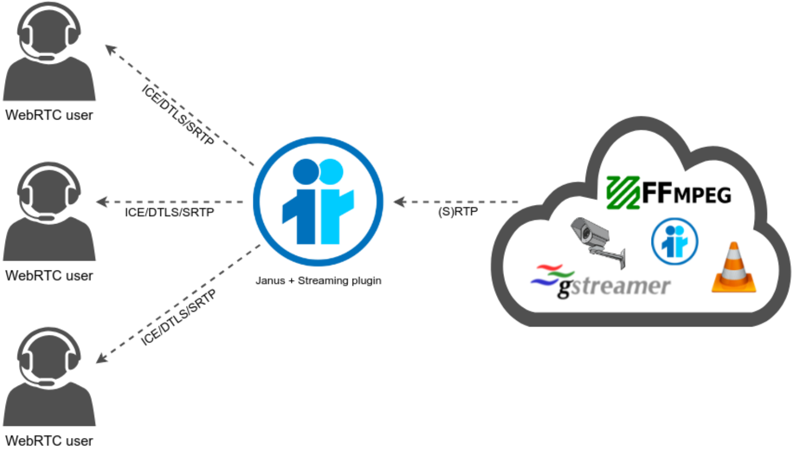
https://janus.conf.meetecho.com/docs/sip

Exchanging RTP (2): NoSIP plugin



https://janus.conf.meetecho.com/docs/nosip

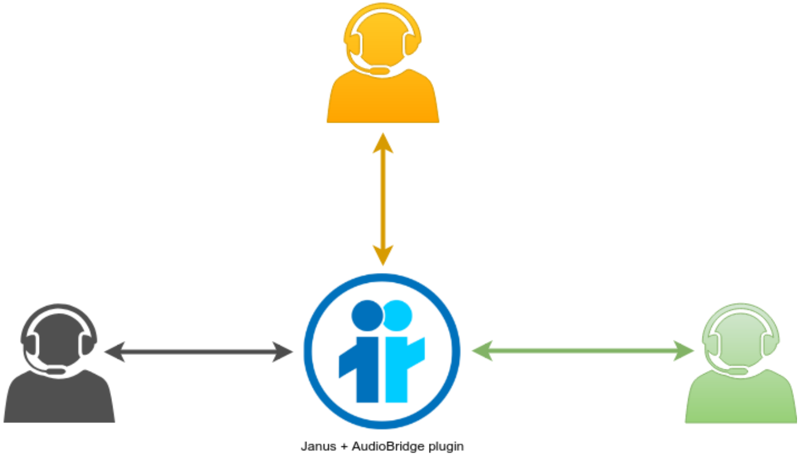
Re-broadcasting RTP: Streaming plugin



https://janus.conf.meetecho.com/docs/streaming

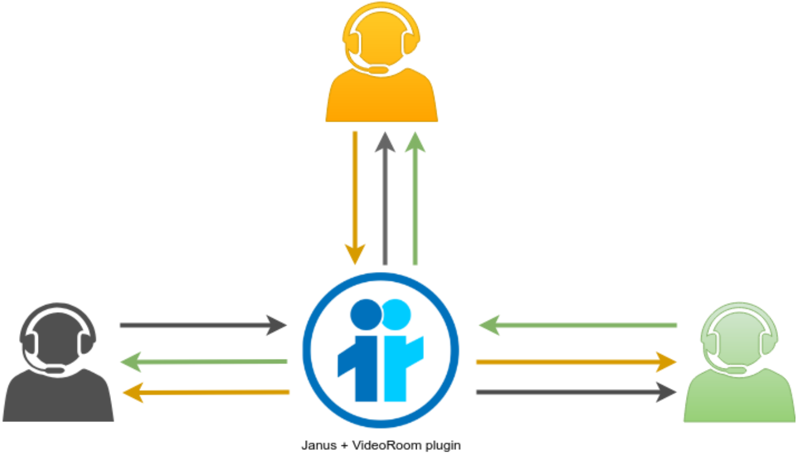
WebRTC-only MCU: AudioBridge plugin

FOSDEM 20



https://janus.conf.meetecho.com/docs/audiobridge

WebRTC-only SFU: VideoRoom plugin



https://janus.conf.meetecho.com/docs/videoroom

Enter “RTP forwarders”

- Plugins like AudioBridge and VideoRoom only deal with WebRTC users...
  - ... but all plugins have access to unencrypted media!
  - Why not just *forward* those unencrypted packets somewhere else?
- Quite simply, relay RTP packets to specified address
  - Plain RTP packets by default; SDES-SRTP supported, though
  - A lot of existing tools can handle RTP natively
  - You can take advantage of multicast networks easily
  - Simulcast streams can be forwarded individually
- Basic RTCP support can help in some scenarios
  - e.g., ask WebRTC peer to send a keyframe via a PLI

What about data channels?

Forwarded on UDP datagrams! (and so limited by MTU, at the moment...)

Enter “RTP forwarders”

- Plugins like AudioBridge and VideoRoom only deal with WebRTC users...
  - ... but all plugins have access to unencrypted media!
  - Why not just *forward* those unencrypted packets somewhere else?
- Quite simply, relay RTP packets to specified address
  - Plain RTP packets by default; SDES-SRTP supported, though
  - A lot of existing tools can handle RTP natively
  - You can take advantage of multicast networks easily
  - Simulcast streams can be forwarded individually
- Basic RTCP support can help in some scenarios
  - e.g., ask WebRTC peer to send a keyframe via a PLI

What about data channels?

Forwarded on UDP datagrams! (and so limited by MTU, at the moment...)

Enter “RTP forwarders”

- Plugins like AudioBridge and VideoRoom only deal with WebRTC users...
  - ... but all plugins have access to unencrypted media!
  - Why not just *forward* those unencrypted packets somewhere else?
- Quite simply, relay RTP packets to specified address
  - Plain RTP packets by default; SDES-SRTP supported, though
  - A lot of existing tools can handle RTP natively
  - You can take advantage of multicast networks easily
  - Simulcast streams can be forwarded individually
- Basic RTCP support can help in some scenarios
  - e.g., ask WebRTC peer to send a keyframe via a PLI

What about data channels?

Forwarded on UDP datagrams! (and so limited by MTU, at the moment...)

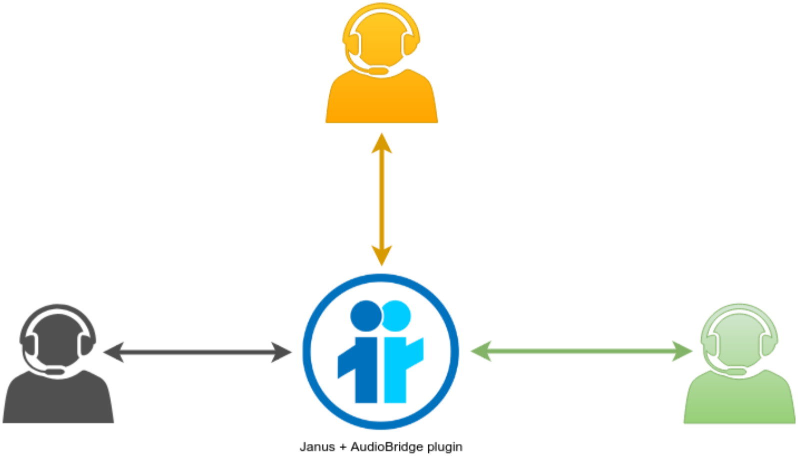
Enter “RTP forwarders”

- Plugins like AudioBridge and VideoRoom only deal with WebRTC users...
  - ... but all plugins have access to unencrypted media!
  - Why not just *forward* those unencrypted packets somewhere else?
- Quite simply, relay RTP packets to specified address
  - Plain RTP packets by default; SDES-SRTP supported, though
  - A lot of existing tools can handle RTP natively
  - You can take advantage of multicast networks easily
  - Simulcast streams can be forwarded individually
- Basic RTCP support can help in some scenarios
  - e.g., ask WebRTC peer to send a keyframe via a PLI

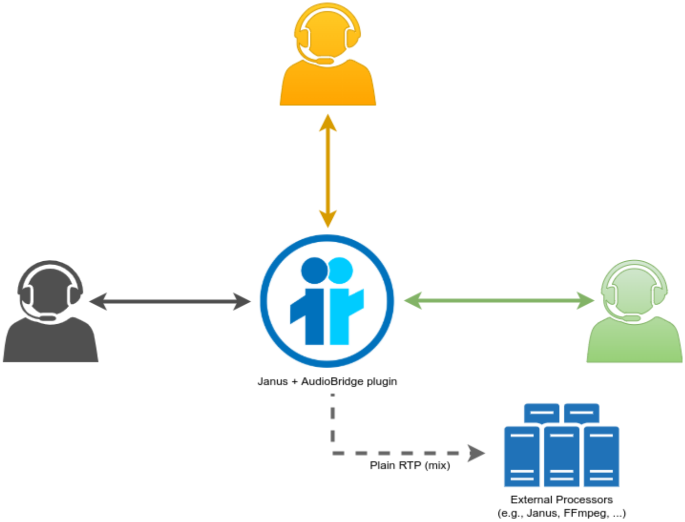
What about data channels?

Forwarded on UDP datagrams! (and so limited by MTU, at the moment...)

Forwarding an AudioBridge mix



Forwarding an AudioBridge mix



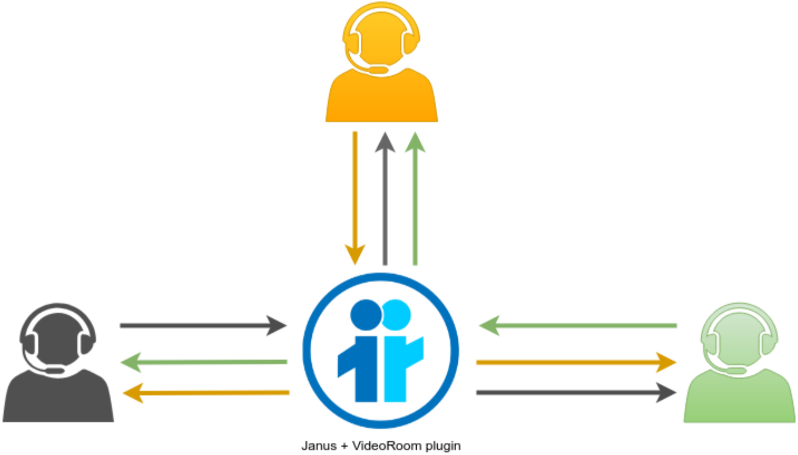
Forwarding an AudioBridge mix

FOSDEM 20

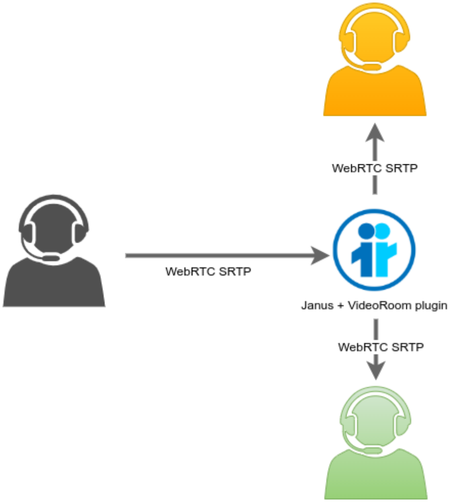
```
{
        "request" : "rtp_forward",
        "room" : <unique ID of the room to add the forwarder to>,
        "ssrc" : <SSRC to use to use when streaming; optional>,
        "ptype" : <payload type to use when streaming; optional>,
        "host" : <host address to forward the RTP packets to>,
        "host_family" : <ipv4|ipv6; optional>,
        "port" : <port to forward the RTP packets to>,
        "srtp_suite" : <length of auth tag (32 or 80); optional>,
        "srtp_crypto" : <crypto key (base64, like SDES); optional>,
        "always_on" : <forward silence when the room is empty; optional>
}
```

Forwarding individual VideoRoom streams

FOSDEM 20

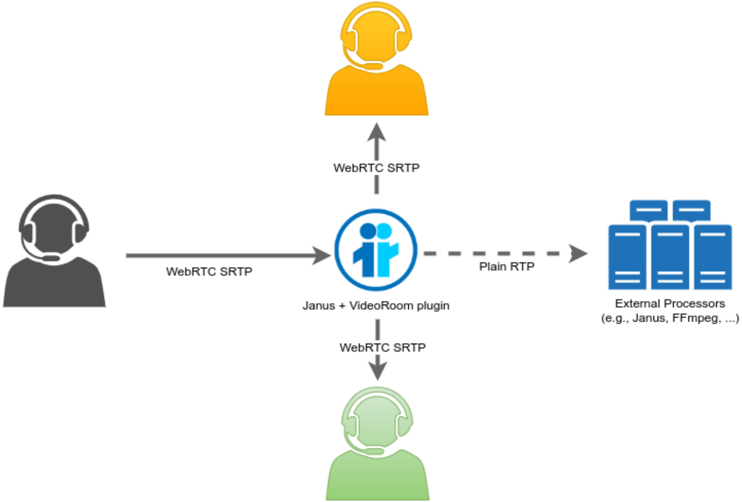


Forwarding individual VideoRoom streams



Forwarding individual VideoRoom streams

FOSDEM 20



Forwarding individual VideoRoom streams

FOSDEM 20

```
{
        "request" : "rtp_forward",
        "room" : <unique ID of the room the publisher is in>,
        "publisher_id" : <unique ID of the publisher to relay externally>,
        "host" : <host address to forward the RTP packets to>,
        "host_family" : <ipv4|ipv6; optional>,
        "audio_port" : <port to forward the audio RTP packets to>,
        "audio_ssrc" : <audio SSRC to use to use when streaming; optional>,
        "audio_pt" : <audio payload type to use when streaming; optional>,
        "audio_rtcp_port" : <latching port to get audio RTCP feedback; optional (unused)>,
        "video_port" : <port to forward the video RTP packets to>,
        "video_ssrc" : <video SSRC to use to use when streaming; optional>,
        "video_pt" : <video payload type to use when streaming; optional>,
           [.. video_XYZ_2 and video_XYZ_3 relay simulcast layers 1 and 2 ..]
        "video_rtcp_port" : <latching port to get video RTCP feedback; optional>,
        "data_port" : <port to forward the datachannel messages to>,
        "srtp_suite" : <length of auth tag (32 or 80); optional>,
        "srtp_crypto" : <crypto key (base64, like SDES); optional>
}
```

What can they be used for?

- Several use cases for such a functionality
  - Remote media processing
    - Recording outside of Janus
    - Translation to different technology (e.g., RTMP)
    - Videomixing or transcoding
    - Transcriptions
    - Identity verification
    - Machine learning / Artificial Intelligence
    - ...
  - Scalability
    - Large scale broadcasting
    - Spreading same conference room in different regions
    - ...
- More simply, whatever you might need need your favourite tool for
  - Gstreamer, FFmpeg, libav, VLC, OpenCV, etc.

What can they be used for?

- Several use cases for such a functionality
  - Remote media processing
    - Recording outside of Janus
    - Translation to different technology (e.g., RTMP)
    - Videomixing or transcoding
    - Transcriptions
    - Identity verification
    - Machine learning / Artificial Intelligence
    - ...
  - Scalability
    - Large scale broadcasting
    - Spreading same conference room in different regions
    - ...
- More simply, whatever you might need need your favourite tool for
  - Gstreamer, FFmpeg, libav, VLC, OpenCV, etc.

What can they be used for?

FOSDEM 20

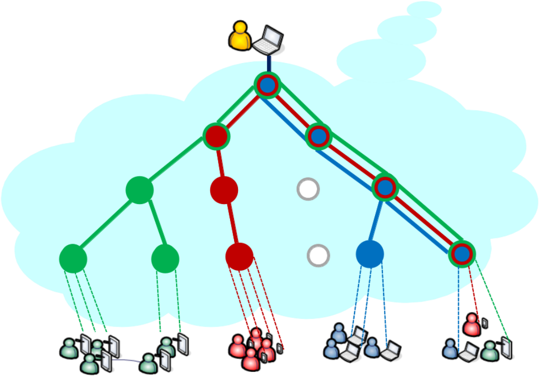
- Several use cases for such a functionality
  - Remote media processing
    - Recording outside of Janus
    - Translation to different technology (e.g., RTMP)
    - Videomixing or transcoding
    - Transcriptions
    - Identity verification
    - Machine learning / Artificial Intelligence
    - ...
  - Scalability
    - Large scale broadcasting
    - Spreading same conference room in different regions
    - ...
- More simply, whatever you might need need your favourite tool for
  - Gstreamer, FFmpeg, libav, VLC, OpenCV, etc.

What can they be used for?

- Several use cases for such a functionality
  - Remote media processing
    - Recording outside of Janus
    - Translation to different technology (e.g., RTMP)
    - Videomixing or transcoding
    - Transcriptions
    - Identity verification
    - Machine learning / Artificial Intelligence
    - ...
  - Scalability
    - Large scale broadcasting
    - Spreading same conference room in different regions
    - ...
- More simply, whatever you might need need your favourite tool for
  - Gstreamer, FFmpeg, libav, VLC, OpenCV, etc.

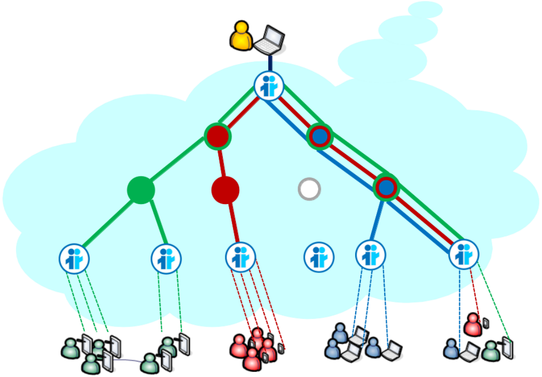
Large scale broadcasting

FOSDEM 20



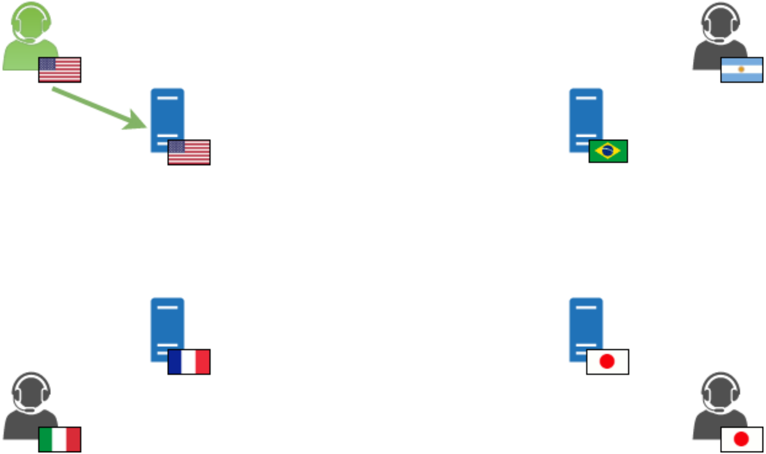
https://www.slideshare.net/LorenzoMiniero/scaling-webrtc-applications-with-janus

Large scale broadcasting



https://www.slideshare.net/LorenzoMiniero/scaling-webrtc-applications-with-janus

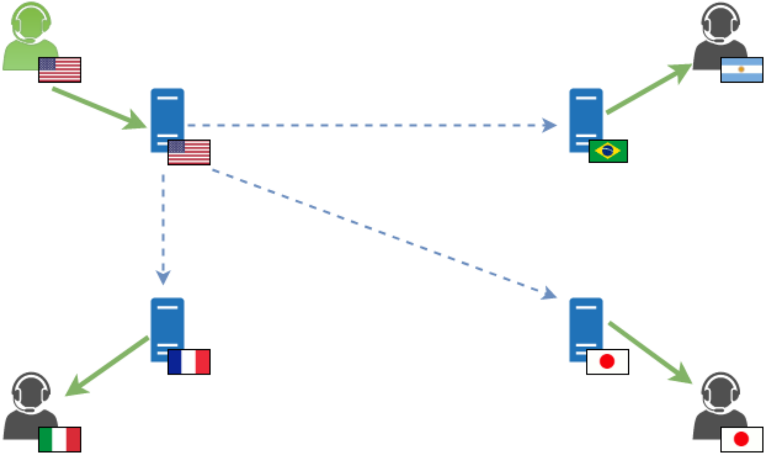
Spreading same room in different regions



https://www.slideshare.net/LorenzoMiniero/scaling-webrtc-applications-with-janus

Spreading same room in different regions

FOSDEM 20



https://www.slideshare.net/LorenzoMiniero/scaling-webrtc-applications-with-janus

What to do next?

FOSDEM 20

- The feature is pretty advanced already
  - Used in production in different contexts
  - Support for multicast and SDES-SRTP is a nice plus
- Some other plugins may benefit from the feature, though
  - SIP/NoSIP, for call monitoring without another server
  - Lua/DuktapeJS, for some easier-to-implement plugin logic
- Maybe implement RTP forwarding as a core feature instead?
  - Right now each plugin does its own (duplicated features)
  - Having it in the core would simplify extending to other plugins
  - Might even become an Admin API feature (e.g., Lawful Interception?)

What to do next?

FOSDEM 20

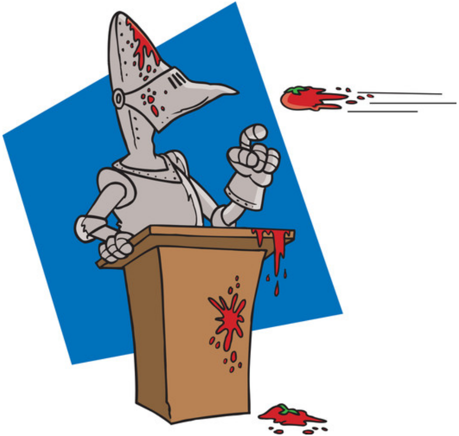
- The feature is pretty advanced already
  - Used in production in different contexts
  - Support for multicast and SDES-SRTP is a nice plus
- Some other plugins may benefit from the feature, though
  - SIP/NoSIP, for call monitoring without another server
  - Lua/DuktapeJS, for some easier-to-implement plugin logic
- Maybe implement RTP forwarding as a core feature instead?
  - Right now each plugin does its own (duplicated features)
  - Having it in the core would simplify extending to other plugins
  - Might even become an Admin API feature (e.g., Lawful Interception?)

What to do next?

FOSDEM 20

- The feature is pretty advanced already
  - Used in production in different contexts
  - Support for multicast and SDES-SRTP is a nice plus
- Some other plugins may benefit from the feature, though
  - SIP/NoSIP, for call monitoring without another server
  - Lua/DuktapeJS, for some easier-to-implement plugin logic
- Maybe implement RTP forwarding as a core feature instead?
  - Right now each plugin does its own (duplicated features)
  - Having it in the core would simplify extending to other plugins
  - Might even become an Admin API feature (e.g., Lawful Interception?)

Thanks! Questions? Comments?



Get in touch!

- https://twitter.com/elminiero
- https://twitter.com/meetecho
- http://www.meetecho.com