Creating a Video Conference service with Kurento

And trying to avoid the nightmare of spike usage.
Agenda

- Me
  - me me me me me me.
- RTMP vs HLS vs WebRTC
  - A little of history.
- Kurento and not "others"
  - Jitsi, I’m looking at you
- PoleAccess.Online
  - Real life requirements
- Resiliency of the infrastructure
  - Distributing the load.
- Thank you
whoami<enter>

- User: criptos
- log start: 1996
- home: /america/mexico/mexico_city
- Process tree:
  --- Owner of Grupo Aullox.
  |--- Still developing core baking applications.
  --- Co-Owner PoleAccess.online
  |--- Main developer.
**RTMP vs HLS vs WebRTC**

RTMP: Too proprietary, remember flash? they do...

HLS: Apple response to RTMP, also closed.

In a nutshell, both protocols provides a lag between 13-45 seconds or more, require “additional” software (like obs) to stream and are proprietary.

WebRTC: Designed for the web, more flexible, using new codes, and still evolving (last add, redundant audio channels).

SFU! WebRTC Rulez!
MESH vs Selective Forward Unit vs Multipoint Conferencing Unit

**Mesh**
- Connections: 4 | 10
- Uplink: 4 mbps
- Downlink: 4 mbps
- Total: 20 mbps

**MCU**
- Connections: 1 | 5
- Uplink: 1 mbps
- Downlink: 1 mbps
- Total: 10 mbps

**SFU**
- Connections: 5 | 25
- Uplink: 1 mbps
- Downlink: 4 mbps
- Total: 25 mbps
Why Kurento?

- Kurento is a Media Server for WebRTC with advanced capabilities for media transformation.
- Is Open Source: [https://github.com/kurento](https://github.com/kurento)
- MCU by design.
- Pluggable.
- Node.js, Vanilla JS, Java, and whatever you like.
- Fairly good documentation.
- Fairly active community
Others...

- ANT, not so open source, community edition = lag, more oriented to be a contained service, like conference room too...

- Jitsi, too oriented to be a "conference room", "difficult to implement", I simply didn’t liked for my use case.

- Nginx RTMP, goodie, oldie, but not very flexible.

- Other implementation, please tell me...
Kurento in a nutshell.

- Please, use the docker image.
  - https://hub.docker.com/r/kurento/kurento-media-server

- Complex to build, too close to Ubuntu.

- Remember, STUN is required.

- GOOD tutorials, good examples at github.

- Is even better with a signalling server.
Kurento in a nutshell.
Kurento in a nutshell.
PoleAccess.online, the pandemic pushing innovation.
PoleAccess.online, Livestreaming with a twist.
Client specifications / CEO Wife orders.

- Oriented to show, not meetings.
  - Using MCU, artificially limit up to 4 hosts, multiple viewers.

- Able to inject video.
  - Kurento player can be added to the composite.
  - Users can be readdressed to player endpoint.

- Tips
  - Not about “video” but about architecture.

- Face 2 face video
  - Using MCU to implement SFU.
PoleAccess.online, Livestreaming with a twist.
Resiliency of the infrastructure

- Single server for 8 core, 32gb ram:
  - Kurento KMS Docker.
  - STUN Server.
  - VM for web page
- Longest stream: 4 continuous hour.
- 2 co-hosts.
- Inline Videos.
- 250 Viewers max, 135 average.
- 2 “proxy servers”
Resiliency of the infrastructure

NGINX RTMP to the rescue.
Virtual Server created and destroyed in demand (thanks vultr)
Good old HLS streaming, (yes with some lag, but no issues)
Questions?

Shoot!

email: criptos @ mobil.aullox.com
twitter: https://twitter.com/criptos
IG: https://www.instagram.com/photodot_poleaccess/
(yes I do photography too…)
Finish

Thank You