

# STREAMING @RADIO FRANCE

National radios broadcasting at scale



Liquidshop 1.0 2021-01-17

Youenn Piolet, Cloud Engineer & DevOps

 [youenn.piolet@radiofrance.com](mailto:youenn.piolet@radiofrance.com)

# HEY!

I'm a Sys~Net~Cloud~DevOps Engineer at Radio France.

Been working here since 2018, in the infrastructure & streaming teams.  
Also music producer and sound enthusiast. Feel free to reach!

[youenn.piolet@radiofrance.com](mailto:youenn.piolet@radiofrance.com)

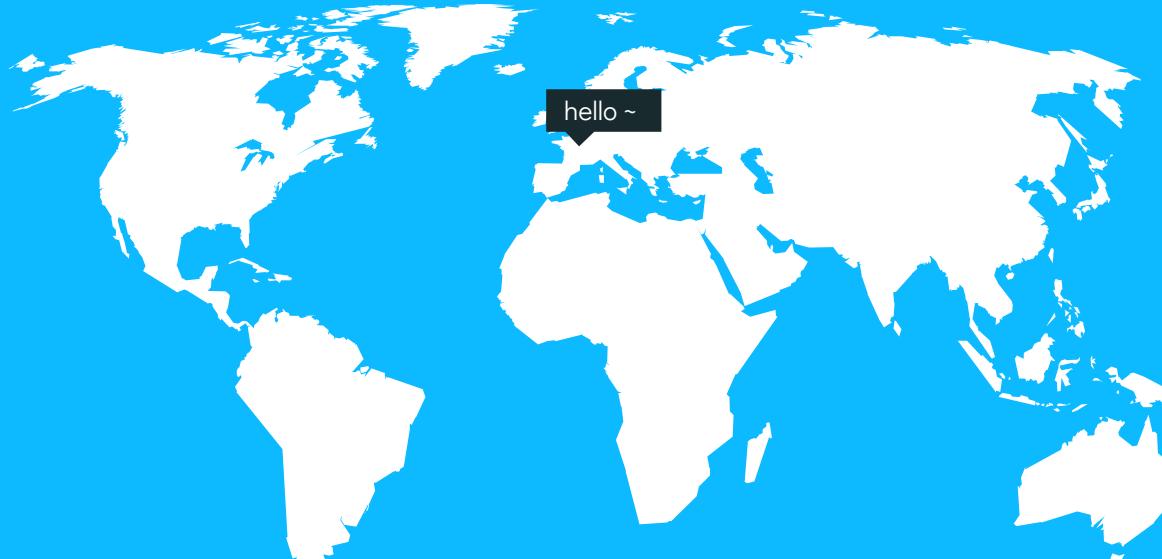
<https://keybase.io/ypiolet>

<https://merveilles.town/@schematicwizard>

<https://github.com/uZer>



# OUR WORKPLACE





radiofrance

# WE ARE radiofrance



## OUR MISSIONS

Information | Education | Entertainment | Culture  
Public service with **903 journalists**, 9 special reporters  
**1058 live events**, 243 897 visitors in **2019**  
National Symphony Orchestra

## “US”, THE NUMERIC BRANCH OF RADIO FRANCE

~200 co-workers handling the presence of Radio France on the Internet (Developers, Infrastructure Engineers, Designers, Marketing Teams, Innovation experts and Data Engineers)

## WE BROADCAST AUDIO

7 national channels  
44 local channels  
23 webradios

**71 Million** listeners for on-demand content per month

**69 Million** monthly web visitors  
(doesn't include France Info)

# ABOUT THIS PRESENTATION

1. Getting rid of the black boxes
2. Build: an infrastructure for audio streaming in the cloud
3. Operate: The tools we built around Liquidsoap

# GETTING RID OF THE BLACKBOXE

## CHAPTER 1

The story of  
how we took control

# BACK IN 2019

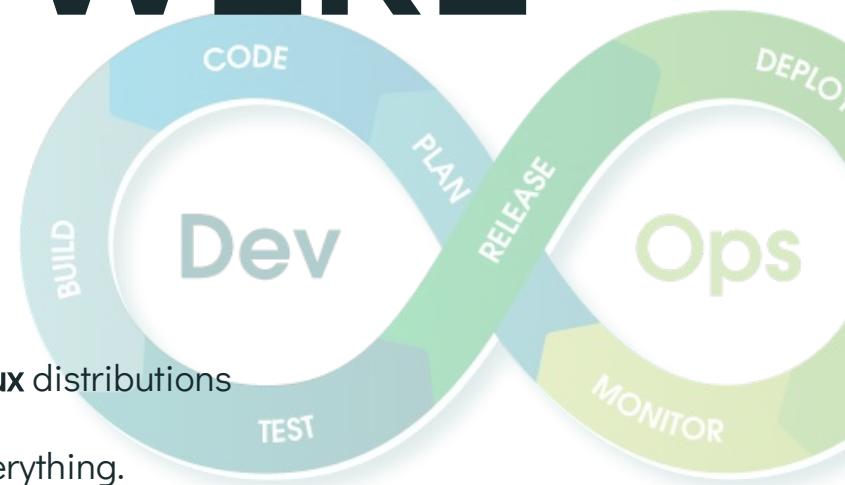
- | **Incidents** almost every day at some point
- | **Inability to serve our content** during the morning peaks
- | A lot of listeners were frequently disconnected during the day
- | Outdated streaming service that gave us no customer satisfaction
- | **Lack of control interface** (Phone calls != API)
- | **Lack of evolutivity**

**In a nutshell:**

Many calls, many failures, **unhappy listeners, unhappy editorialists**

# WHO WE WERE

- | Just a bunch of people hosting websites, lol
- | ...working on a **cloud based infrastructure**
- | ...with **open source technologies** only and **Linux** distributions
- | ...with **Kubernetes** at the core of our platform
- | We code and operate **APIs**, **industrializing** everything.



# WHO WE WERE NOT

- | Audio experts
- | Transport experts
- | Support teams
- | ~~Infrastructure Engineers~~ Server buyers



# WHAT WE ALREADY HAD

- | An industrialized and resilient infrastructure in the cloud
- | Strong knowledge about web services hosting
- | Encoded audio sources available in a local network at “La Maison de la Radio”

# 2,000,000+

Per day listeners on our streams

*What we had  
to handle*

# 300,000+

Simultaneous listeners during our morning peaks

# 200+ audio streams

74 radio channels with multiple audio qualities

**Multiple sources** for each of our channels

# FIRST IDEAS

- | Adaptive bitrate streaming and buffering for audio streaming are great
- | HLS can work for us, our former job is to operate HTTP services
- | Icecast is NOT great, but will be inevitable for old Internet connected radios
- | We want control over the qualities of the streams we're broadcasting (bitrate, profile, encoding, ...)



# PROTOCOLS

*...just a bunch  
of them*

## ICECAST

- | Streaming server
- | **Master + Relay** Architecture
- | Single bitrate
- | No caching (well... no.)

PERSISTENT TCP CONNECTIONS

## HLS

- | **Segmented** audio/video **files**
- | Sliding Playlist (m3u8)
- | Adaptive bitrate
- | Better mobile experience
- | Cacheable content

JUST “STATIC” FILES

# NEXT IDEAS

- | If we do HLS, we need a way to **generate the audio segments** and the playlist
- | For each station, we need a way to **select/switch** source audio streams without disconnecting everyone.
- | We need a way to **remux/re-encode** source audio streams with different audio codecs, and different quality settings -> producing multiple audio streams per radio channel

# STUDYING THE STATE OF THE ART

- | How to produce and operate **resilient** **livestreams**
- | How to maximize the **sound quality** and **stability** for listeners
- | How to **scale** (and do what our professional service provider couldn't do)
- | How to **monitor** things, the way we usually do
  
- | Wait, how do we **ship audio to the cloud?**

# REFUSING FAILURE

- | In case of source failure, we want a **failover mechanism to switch source instantaneously**
- | We want to detect blank audio
- | We want to build something resilient (multiple availability zones)
- | We want to build something we can scale if needed

# ON THE MARKET?

I A lot of good products, but....

- Cost?
- Flexibility?
- Ownership?
- Open Source?

We are a public service, and need to rely on a long term solution

# LIQUIDSOAP!

- | Audio and video streaming language
- | Radio As Code ❤️
- | A pipeline for our audio transformations
- | API compliant!
- | Open Source | Professional use cases | Maintained & Active | 🇫🇷
- | Possible collaboration with Radio France

This is the perfect open source project for our needs

# DIFFUSION?

- | We already have the infrastructure, we just need to develop a few new services
- | We can re-use our current monitoring tools
- | We can stream in HLS, just like we host web content.

A blurred background image showing a person's hands holding a ruler and a pencil, working on a drawing or sketch on a light-colored surface.

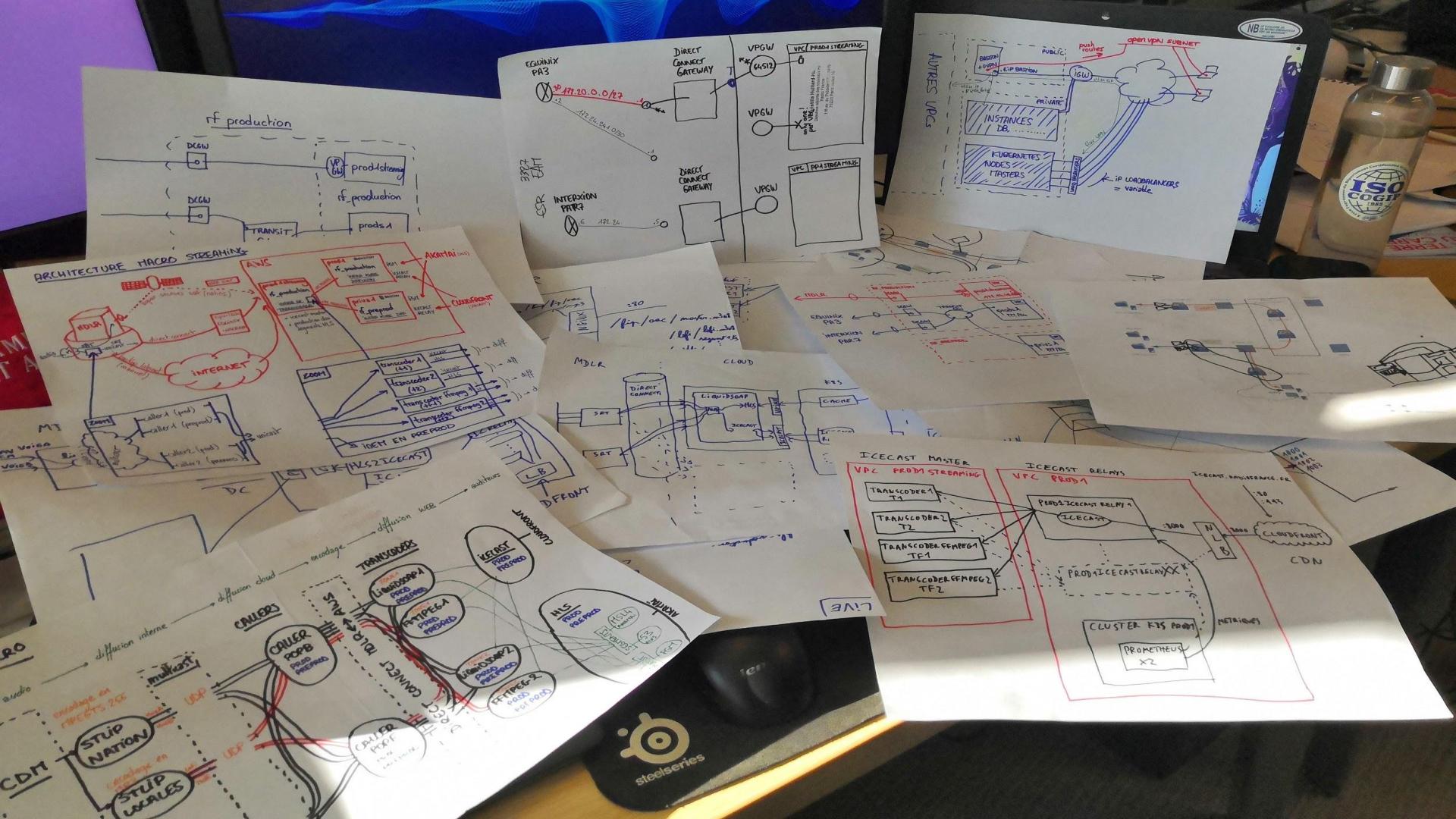
so, ACTUALLY...

**WE CAN  
DO IT.**

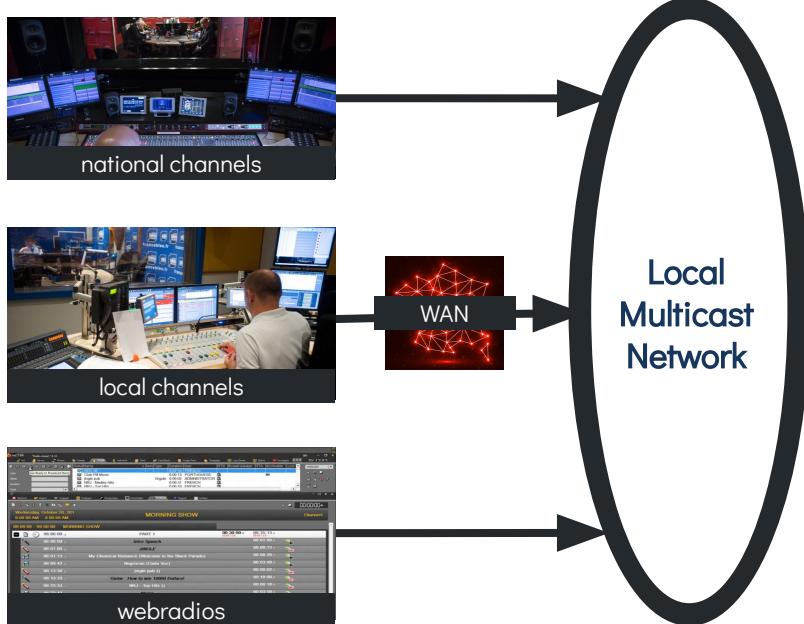
# BUILDING THE INFRA- STRUCTURE

## CHAPTER 2

Streaming from the Cloud



# TRANSPORT: THE MULTICAST PROBLEM

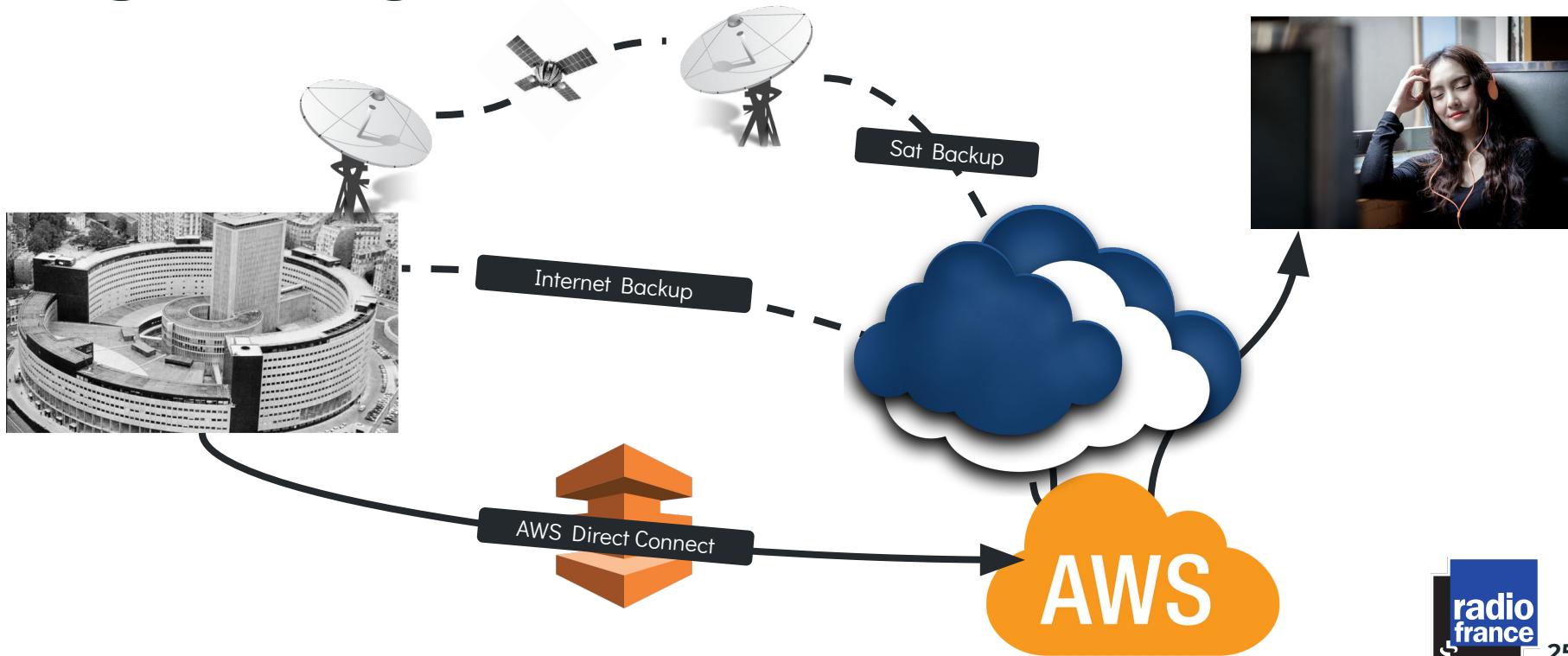


?

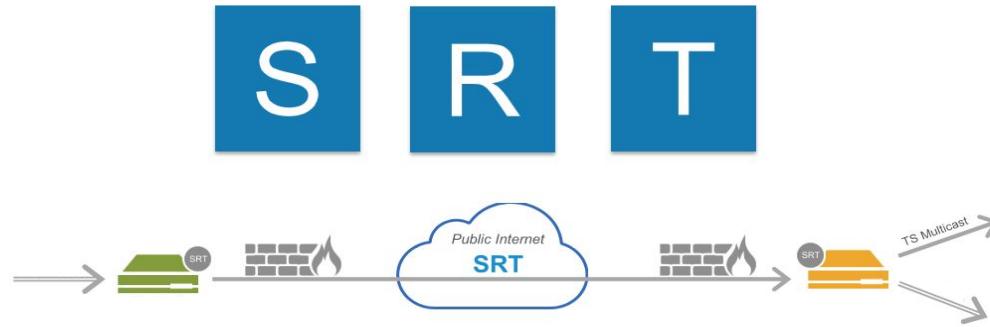


No multicast in AWS!

# TRANSPORT: RESILIENCE

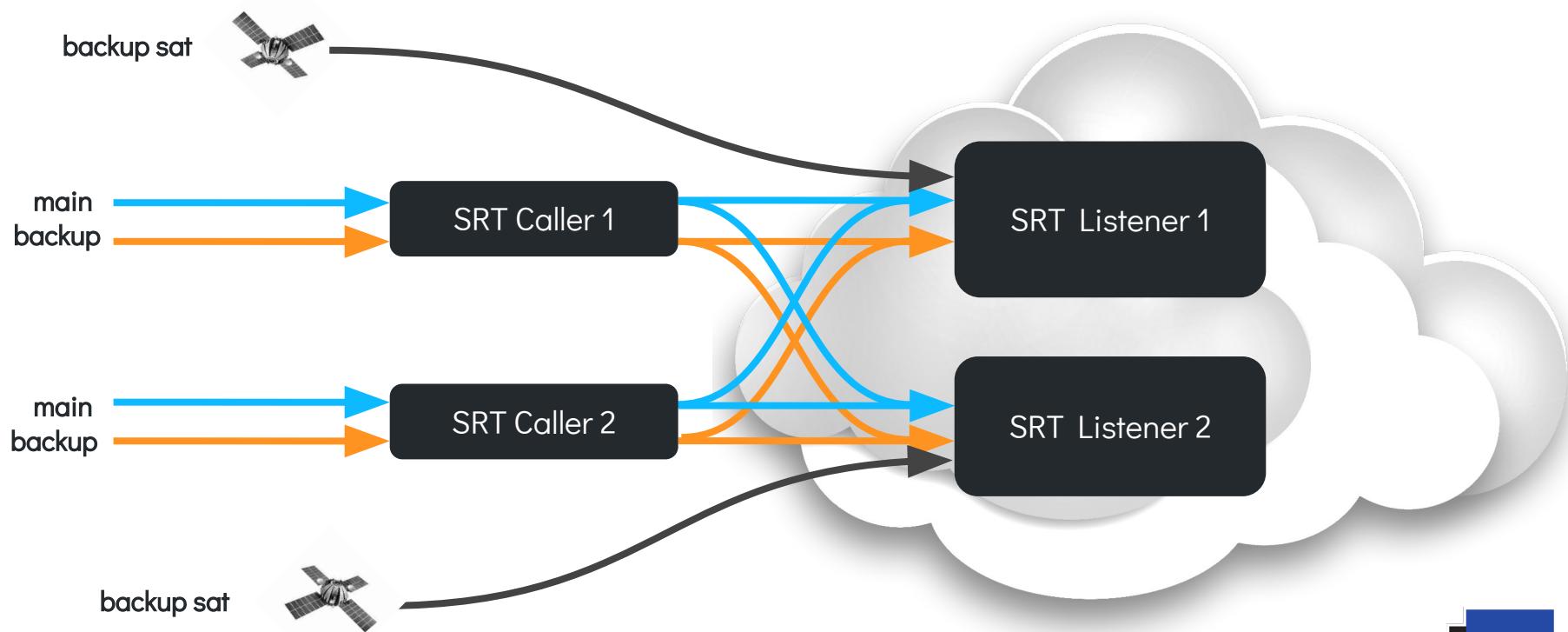


solution:  
**SRT**

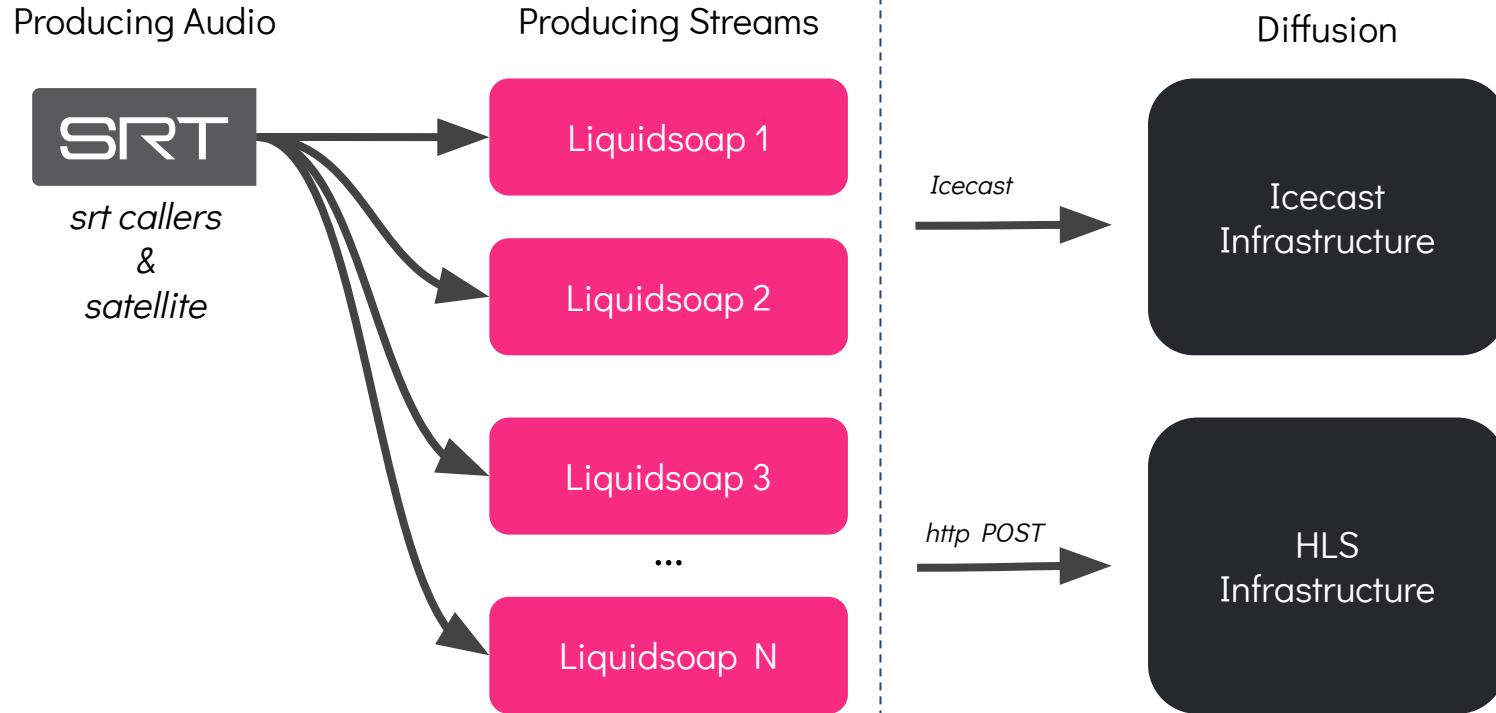


**Secure | Reliable | Transport**

# SRT COMMUNICATIONS



# MACRO ARCHITECTURE



# LIQUIDSOAP

## CURRENT USAGES

### Receive

SRT listener

### Encode / Mux

Encode in AAC

Encode in MP3

Produce HLS

Produce Icecast

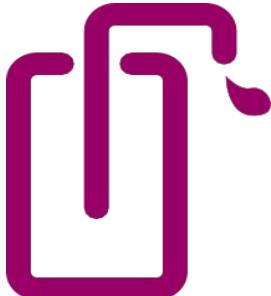
### Control

Switch between sources

Fallback logic + Safe blank

Expose metrics

Expose API



# RADIO AS CODE: INPUTS

```
main_caller1 = buffer(fallible=true, input.srt(port=10000))
main_caller2 = buffer(fallible=true, input.srt(port=10001))
backup_caller1 = buffer(fallible=true, input.srt(port=10002))
backup_caller2 = buffer(fallible=true, input.srt(port=10003))
override_caller1 = buffer(fallible=true, input.srt(port=10004))
override_caller2 = buffer(fallible=true, input.srt(port=10005))
sat_sat1 = buffer(fallible=true, input.srt(port=10006))
safe_blank = blank()
```

# RADIO AS CODE: FALLBACK LOGIC

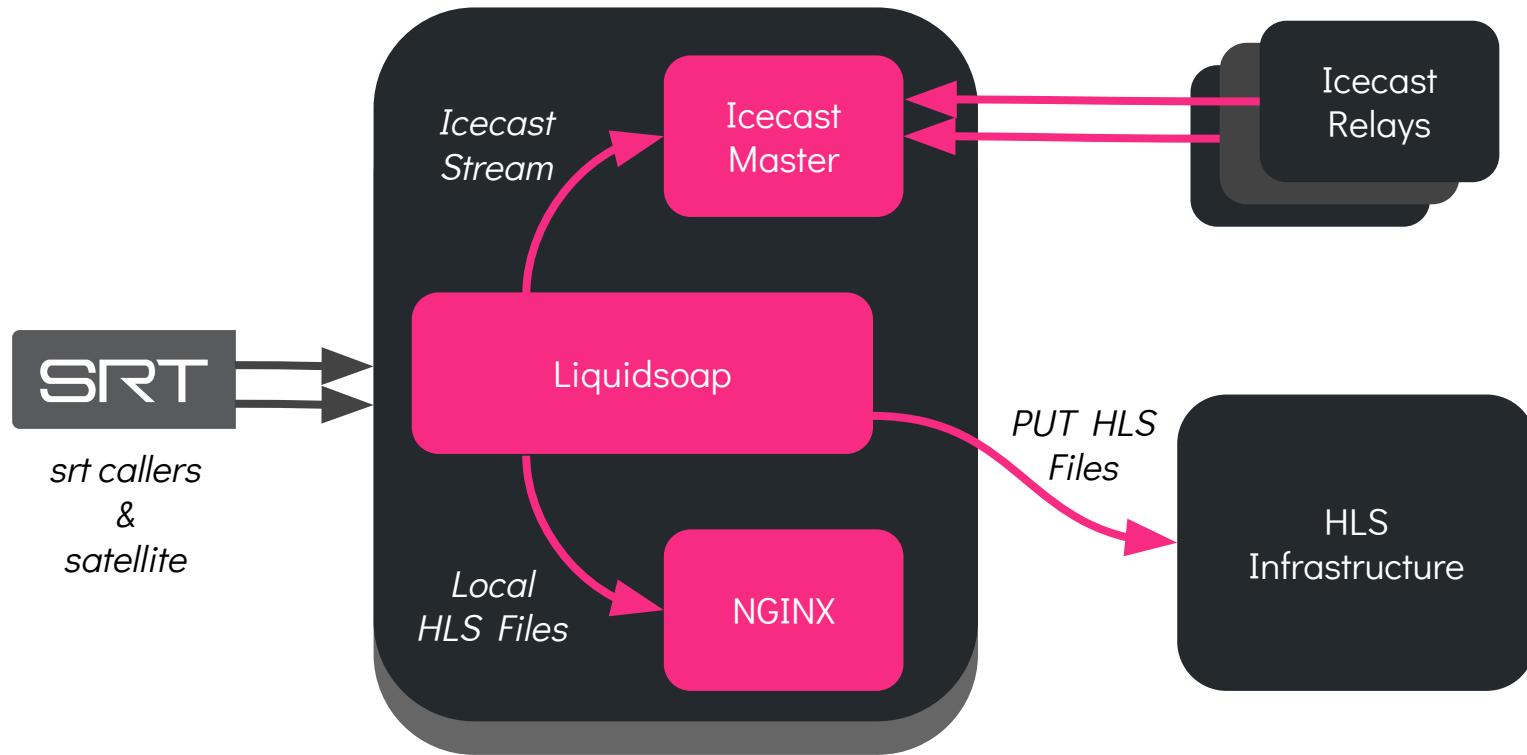
```
live = switch(  
  [· (is_playing("main_caller1"),main_caller1),  
  · (is_playing("main_caller2"),main_caller2),  
  · (is_playing("backup_caller1"),backup_caller1),  
  · (is_playing("backup_caller2"),backup_caller2),  
  · (is_playing("override_caller1"),override_caller1),  
  · (is_playing("override_caller2"),override_caller2),  
  · (is_playing("sat_sat1"),sat_sat1) · ])
```

```
radio_prod = fallback(  
  [· live,  
  · · main_caller1,  
  · · main_caller2,  
  · · backup_caller1,  
  · · backup_caller2,  
  · · sat_sat1,  
  · · safe_blank · ])
```

# AUDIO FORMATS

HLS <i>HTTP Protocol</i> segments	AAC	<b>hifi (192kbps) aac_lc</b> <b>midfi (96kbps) aac_lc</b> <b>lofi (32kbps) he_aac_v2</b> <i>Spectral Band Replication (SBR) + Parametric Stereo (PS)</i>	<div><div><span>AAC LC</span><span>SBR</span><span>PS</span></div><p>AAC Profile</p><p>High Efficiency AAC Profile</p></div> <div><p>High Efficiency AAC v2 Profile</p></div>
ICECAST <i>Connected Protocol</i> TCP	AAC	<b>hifi (192kbps) aac_lc</b> <b>midfi (96kbps) aac_lc</b> <b>lofi (32kbps) he_aac_v2</b> <i>Spectral Band Replication (SBR) + Parametric Stereo (PS)</i>	
	MP3	<b>midfi (128kbps)</b> <b>lofi (32kbps)</b>	

# TRANSCODERS

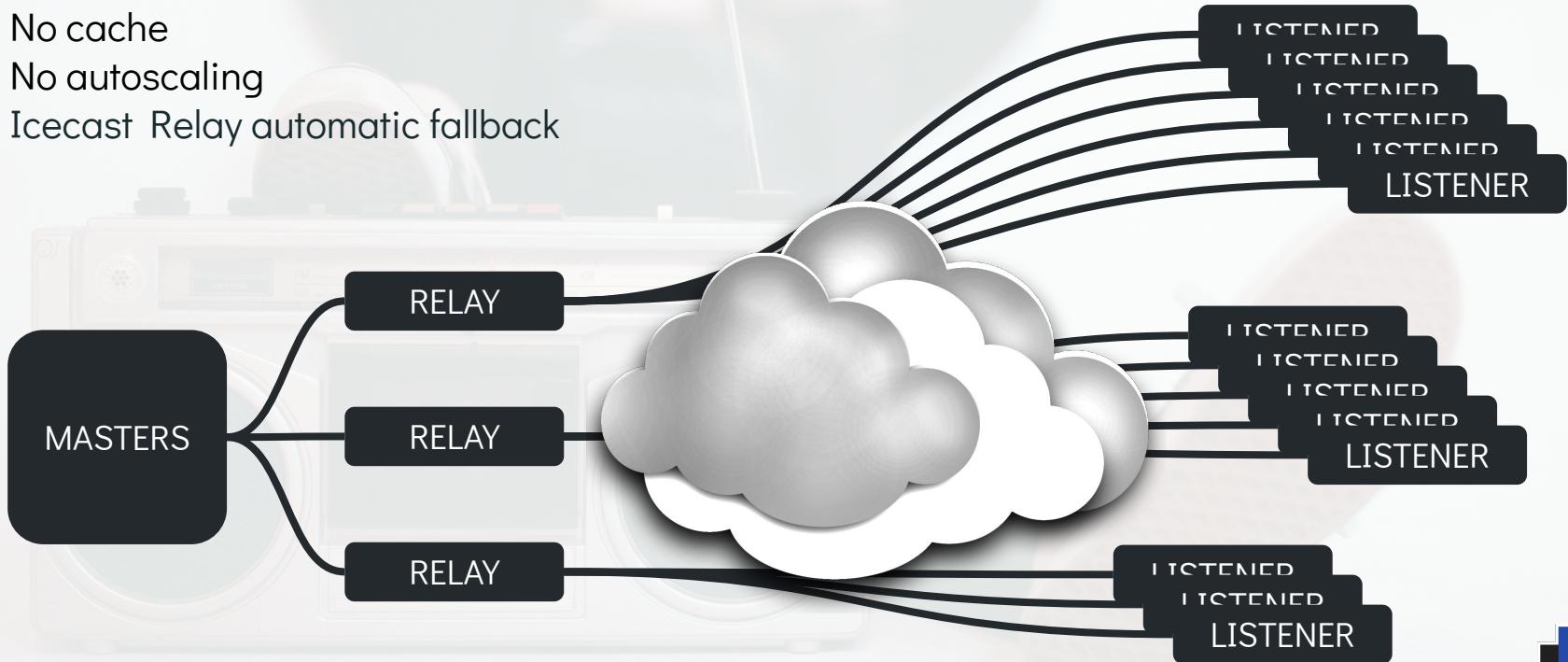


# ICECAST ARCHITECTURE

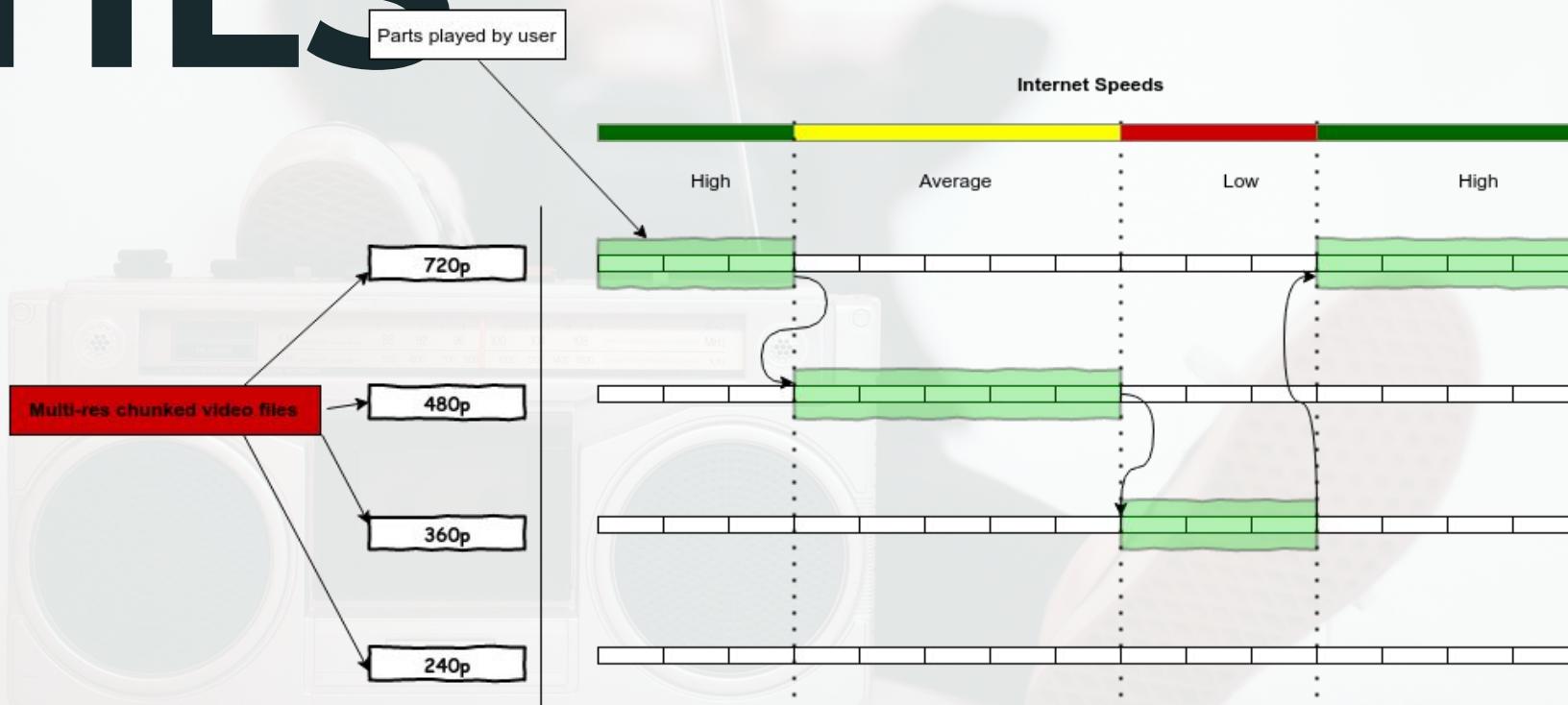
No cache

No autoscaling

Icecast Relay automatic fallback



# HLS



# HLS

```
curl https://stream.radiofrance.fr/fip/fip.m3u8
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-STREAM-INF:PROGRAM-ID=0,BANDWIDTH=78000,CODECS="mp4a.40.2"
fip_lofi.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=0,BANDWIDTH=160000,CODECS="mp4a.40.2"
fip_midfi.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=0,BANDWIDTH=252000,CODECS="mp4a.40.2"
fip_hifi.m3u8
```

```
curl https://stream.radiofrance.fr/fip/fip_hifi.m3u8
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:1576284
#EXT-X-TARGETDURATION:4
#EXT-X-START:TIME-OFFSET=0
#EXTINF:4.000,
/msl4/fip/prod1transcoder2/fip_aac_hifi_4_1604416229_1576284.ts
#EXTINF:4.000,
/msl4/fip/prod1transcoder2/fip_aac_hifi_4_1604416229_1576285.ts
#EXTINF:4.000,
/msl4/fip/prod1transcoder2/fip_aac_hifi_4_1604416229_1576286.ts
#EXTINF:4.000,
/msl4/fip/prod1transcoder2/fip_aac_hifi_4_1604416229_1576287.ts
#EXTINF:4.000,
/msl4/fip/prod1transcoder2/fip_aac_hifi_4_1604416229_1576288.ts
#EXTINF:4.000,
/msl4/fip/prod1transcoder2/fip_aac_hifi_4_1604416229_1576289.ts
#EXTINF:4.000,
/msl4/fip/prod1transcoder2/fip_aac_hifi_4_1604416229_1576290.ts
```

# HLS INFRASTRUCTURE

## Segment manipulation

Upload to CDN

Store timestamps in a database

## Playlist generation

Timeshift

Adaptive quality

Choose segment source

## Scaling & high availability

Scalable ingresses

Scalable APIs + Cache

CDN for the segments

# OPERATIONS

## AND TOOLING

### CHAPTER 3

Developing the tools to  
operate our streaming  
platform

# OBSERVABILITY



## Exporters

Node Exporter  
Process Exporter  
**Liquidsoap (native)**  
Icecast Exporter  
NGINX Exporter  
Custom API Exporters

+ Logs



Prometheus



logstash

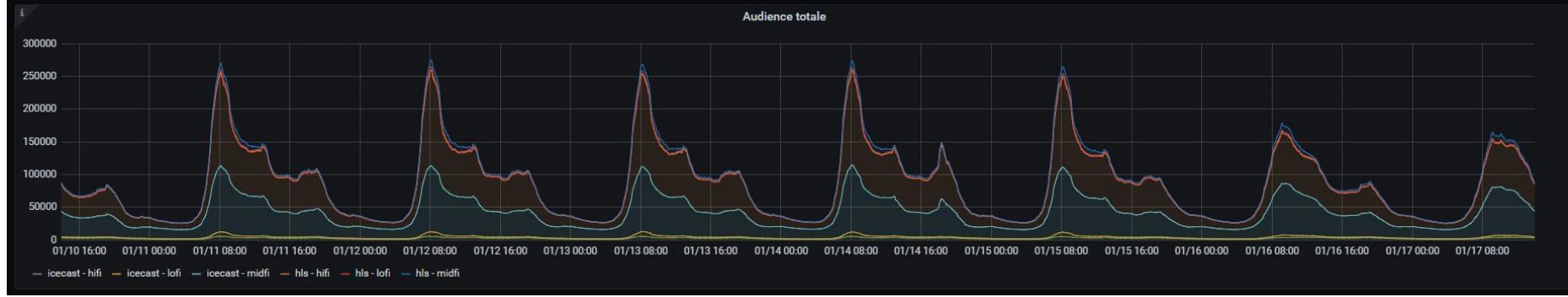


Grafana



kibana

# OBSERVABILITY



prod1transcoder1					
Radio ↑	Real Live Source	Preferred Live Source	Is Blank	Voice A	Voice B
fb1071	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fip	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipelectro	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipgroove	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipjazz	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipnouveautés	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fippop	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipreggae	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fiprock	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipworld	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fluxaeterna	voieA_caller1	voieA_caller1	No blank	All ready	No Voice ...
franceculture	voieA_caller1	voieA_caller1	No blank	All ready	Sat ready

prod1transcoder2					
Radio ↑	Real Live Source	Preferred Live Source	Is Blank	Voice A	Voice B
fb1071	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fip	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipelectro	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipgroove	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipjazz	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipnouveautés	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fippop	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipreggae	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fiprock	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fipworld	voieA_caller1	voieA_caller1	No blank	All ready	All ready
fluxaeterna	voieA_caller1	voieA_caller1	No blank	All ready	No Voice ...
franceculture	voieA_caller1	voieA_caller1	No blank	All ready	Sat ready

# CUSTOM API & SERVICES

## STREAM-API & ANSIBLE

### streamAPI

#### routes

All routes are prefixed with v1

#### get stations

GET /v1/stations

Returns all stations

query params :

value	description	default value
name	filter output based on the station name	
url_id	append id= url_id to the stream urls	radiofrance

- | As code referential (versioning / revert / review...)
  - liquidsoap
  - srt callers
  - icecast
  - nginx...
- | Define inputs, outputs, fallback mechanisms
- | API to make settings available in JSON

# CUSTOM API & SERVICES

## INTERFACE

### Filtres

Stations: Preferred Transcoders: Preferred Livesources: Tags:

fip  preferred transcoders ... preferred livesources ... tags ...

### Basculer

Basculer 0 station vers le **preferred transcoder**  ou la **preferred livesource**

override\_caller1  
override\_caller2  
sat\_sat1  
voieA\_caller1  
voieA\_caller2  
voieB\_caller1  
voieB\_caller2

### Stations

<input type="checkbox"/>	Station	Preferred transcoder	Transcoder	Preferred Livesource	resource
<input type="checkbox"/>	fip	transcoder2	transcoder1	voieA_caller1	voieA_caller1
			transcoder2	voieA_caller1	voieA_caller1

# CUSTOM API & SERVICES

## SWITCH CONTROLLER

- | Operate liquidsoap cluster
- | Make complex decisions based on liquidsoap and other datasources
- | Notify admins



# EVOLUTIONS WITH LIQUIDSOAP 2.0



- | **ENCODING:** Usage of advanced ffmpeg filters
- | **ENCODING:** Encode once, mux in HLS/Icecast
- | **OUTPUTS:** MPEG-DASH can be interesting
- | **SRT:** Replacing our srtcallers with liquidsoap
- | **SRT:** Export metrics
- | **SRT:** Do actual SRT

# THANKS!

## WE HIRE

<https://www.welcometothejungle.com/fr/companies/radio-france>

**More info:**

[https://archive.fosdem.org/2020/schedule/event/om\\_audio\\_streaming/](https://archive.fosdem.org/2020/schedule/event/om_audio_streaming/)