

# Micro-Services @ RadioFrance

What if we could do it all over again ?

API Days Paris - janv. 2018

**radiofrance**

# Who am I ?

Rodolfo Ripado

@gaspaio

Tech Lead - Data Pipelines, APIs  
Digital IT Dept. @Radio France



franceinfo: MOUV'

What do we do at

**DIGITAL IT DEPT@RADIO FRANCE**

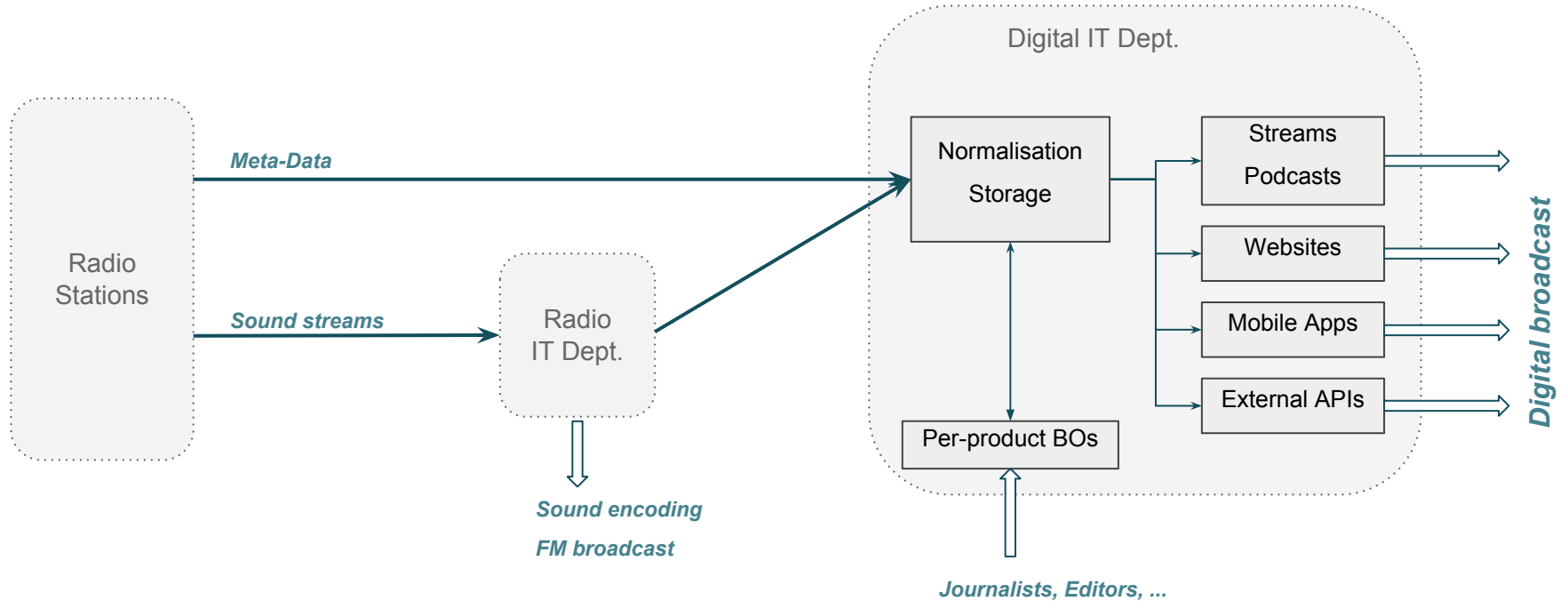
# Modeling and distribution of all digital Radio France data



# Modeling and distribution of all digital Radio France data

- Multiple websites
- Multiple APIs for internal and external “clients”
- Multiple mobile apps

# Radio France Digital



## November 2017 : some numbers

- **+40 million visits** (throughout all our products)
- **+50 million audio streams** (over all our players)
- **+55 million podcasts** downloaded
- **+35 million videos** watched

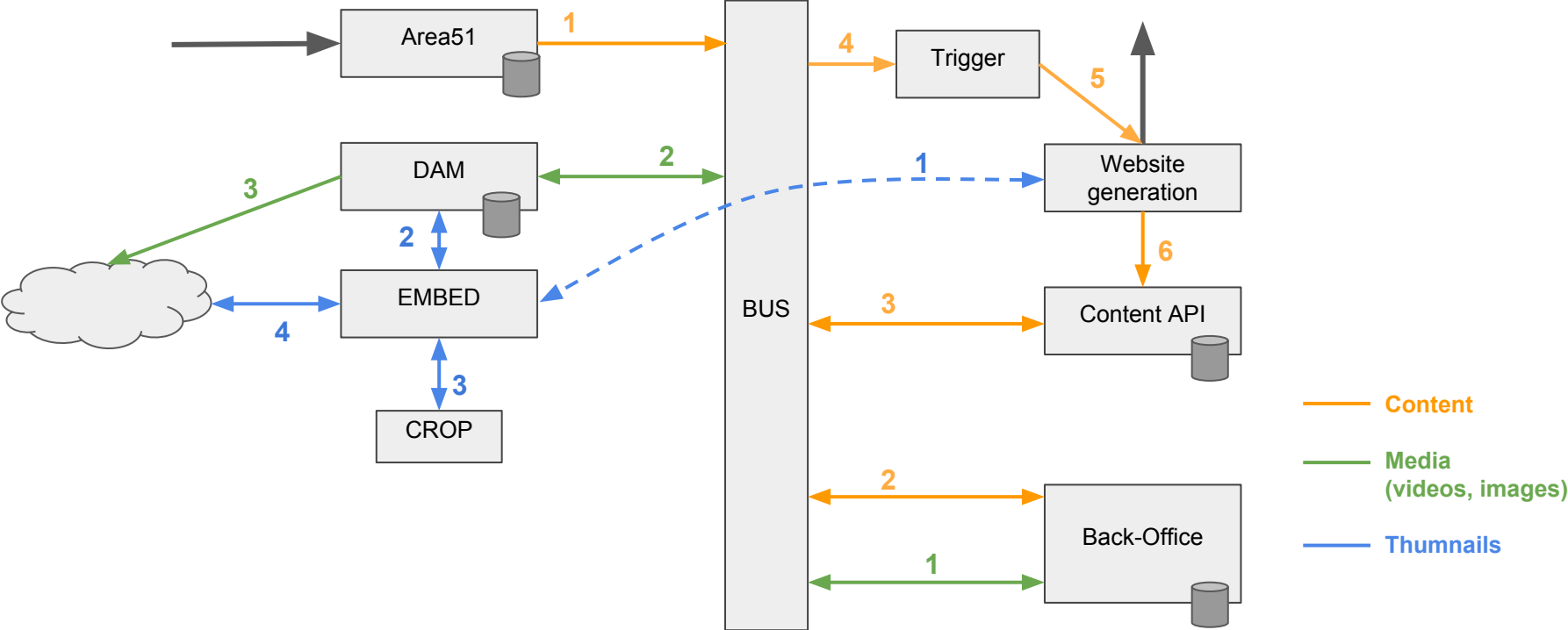


franceinfo:  MOUV'

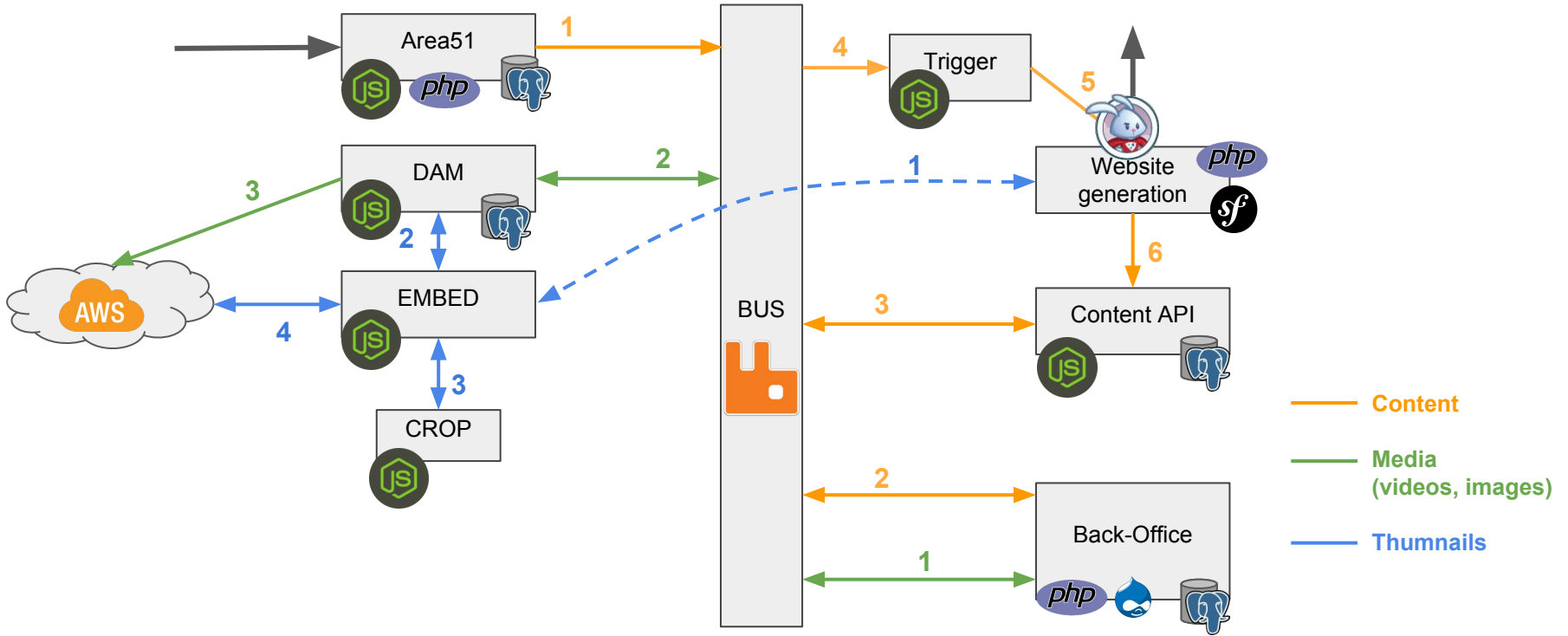
# TECHNICAL ARCHITECTURE



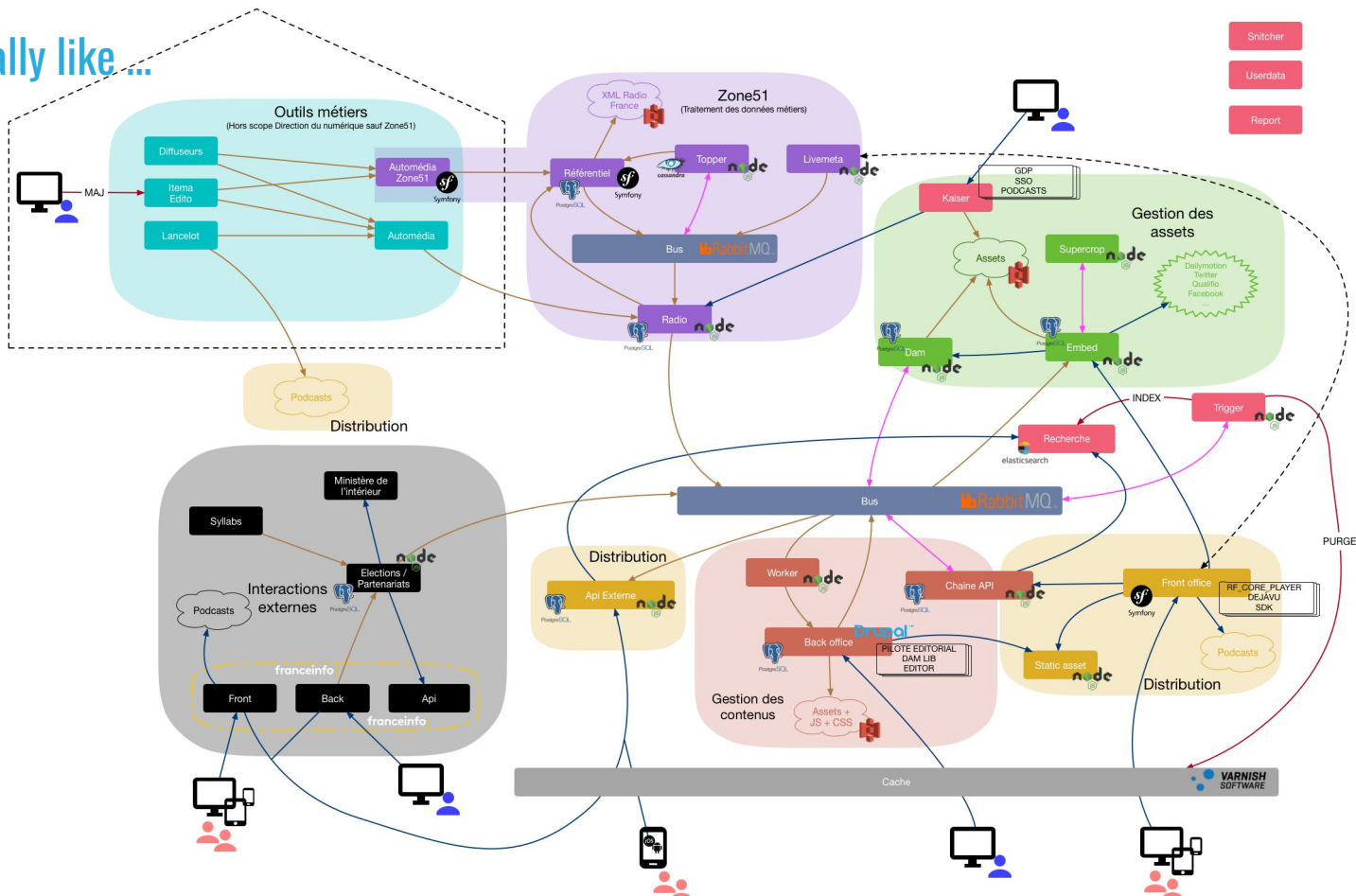
# A bird's eye view of our tech stack : generating a webpage



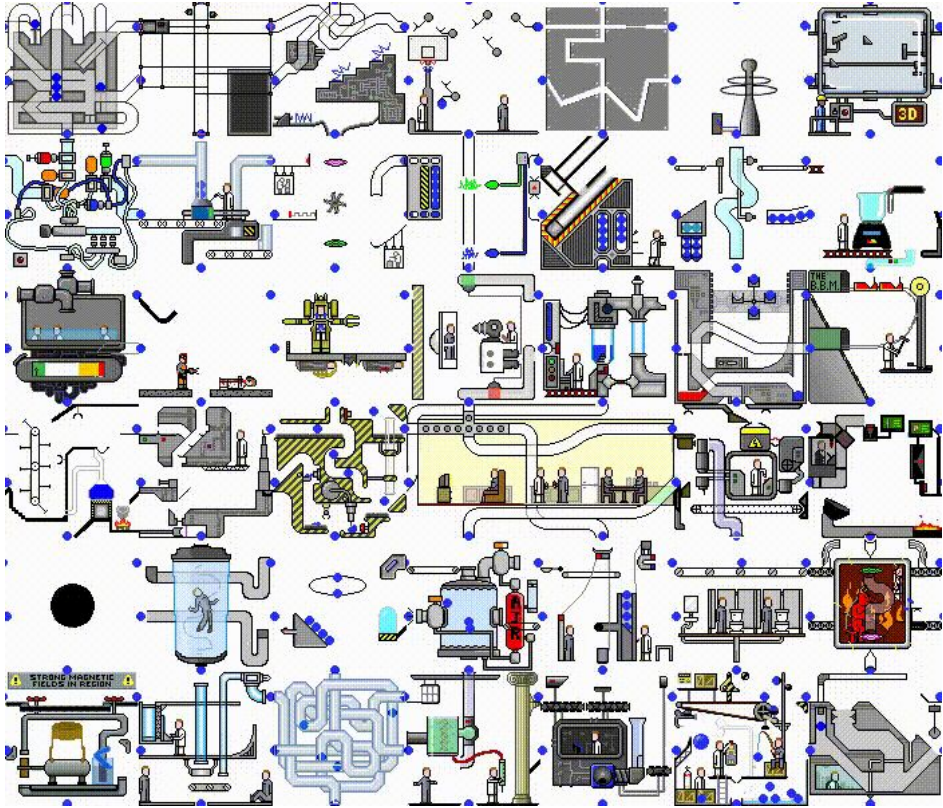
# A bird's eye view of our **tech stack** : generating a webpage



# What it is really like ...



Sometimes it just feels like ...



franceinfo: MOV

# LESSONS **LEARNED**

on

Micro-Service Architectures

Lessons learned

# Enforce a consistent and extensive logging policy



franceinfo: MOUV'

Lessons **learned**

# Enforce a consistent and extensive logging policy

ElasticSearch/Kibana (or SAAS alternative) are your greatest friends

Set a “token” for related events

Build meaningful dashboards



franceinfo: MOUV'

Lessons learned

# Limit the languages and tech-stacks across teams



franceinfo: MOUV'



Lessons **learned**

# Limit the languages and tech-stacks across teams

Avoid creating “language barriers” between teams

Helps dealing with (normal) developer turnover

Reduce complexity



franceinfo: 

The logo for franceinfo: MOUV', featuring the text "franceinfo:" in black and "MOUV'" in white text on a black square background.

Lessons **learned**

# Find the micro-services sweetspot



franceinfo: MOUV'

Lessons **learned**

# Find the micro-services sweetspot

In short : Each MS should provide a valuable service by itself

Its a work in progress : be prepared to refactor your services ...



franceinfo: The logo for MOUV', featuring the word "MOUV'" in white text on a black square background, with a white circle containing a black dot to the left of the text.

Lessons **learned**

# Let go of data-consistency across systems



franceinfo: 

Lessons **learned**

# Let go of data-consistency across systems

Each service that modifies the data keeps its own representation

Keep a Unique ID for each object (like application generated UUIDs)



franceinfo: The logo for MOUV', featuring the word "MOUV'" in white text on a black square background, with a white circle containing a black dot to the left of the text.

Lessons learned

# Event-sourcing



franceinfo: MOUV'

Lessons **learned**

# Event-sourcing

Adds complexity, but the ability to replay changes is (almost) priceless

We didn't fully implement it. We probably should have gone a lot deeper

Not easy to do with RabbitMQ



franceinfo: The MOUV' logo, consisting of the word "MOUV'" in a bold, sans-serif font, with a stylized circular icon to its right.

Lessons learned

# Read-write segregation

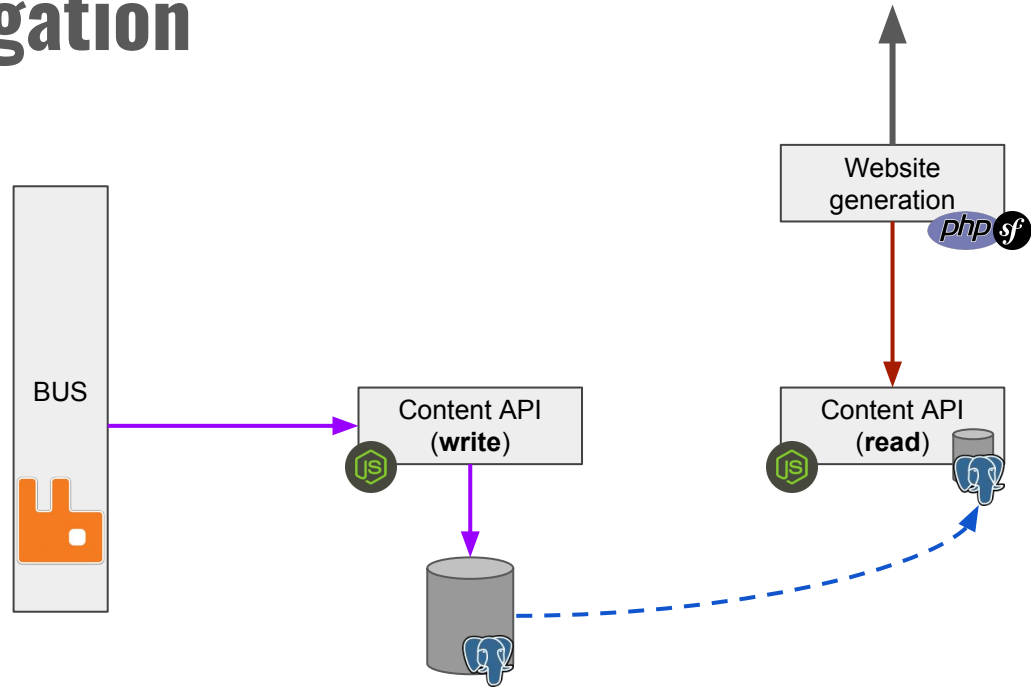


franceinfo: MOUV'



Lessons learned

# Read-write segregation



Lessons **learned**

# Follow an API specification



franceinfo: The logo for MOUV', featuring the word "MOUV'" in a stylized font with a circular symbol containing a dot, positioned to the right of the text.

Lessons **learned**

# Follow an API specification

Any spec is infinitely better than no spec. We use JSON-API

Keep names, fields, conventions consistent accross APIs and teams

Use schema validation



franceinfo: The logo for MOUV', featuring the letters "MOUV'" in white text on a black square background.

Lessons **learned**

# Dates : keep it simple



franceinfo: The logo for MOUV', featuring the word "MOUV'" in white text on a black square background, with a white circle containing a black dot to the left of the text.

Lessons **learned**

# Dates : keep it simple

Use UTC everywhere : from machines to apps to databases

Convert at the borders

Yes, sometimes real-life happens. But keep exceptions to a minimum



franceinfo: MOUV'

Lessons learned

# Caching : keep it simple



franceinfo: MOUV'

Lessons **learned**

# Caching : keep it simple

Avoid caching between “internal” APIs

Cache the end result (web pages, external API results, ...)

Manage load by other means : scaling, SQL design, worker throttling, ...



franceinfo: The logo for Mouv', featuring the word "Mouv'" in white text on a black square background.

Lessons learned

# Don't get Vendor-locked-in (if you can help it)



franceinfo: MOUV'



Lessons **learned**

# Don't get Vendor-locked-in (if you can help it)

Cloud vendor independence gets you a lot of leverage when negotiating prices

But : comes at a high cost - technical skills are expensive

Not for everybody : do your math carefully



franceinfo: The logo for MOUV', featuring the word "MOUV'" in white text on a black square background.



Thank you !

**I WANT TO  
BELIEVE**

**IN SOFTWARE CRAFTSMANSHIP !**

VENEZ COMMITER VOS DÉVELOPPEMENTS DURABLES



franceinfo: MOUV'



Can I haz questions plz ?

contact @rf\_atelier\_num, rodolfo.ripado@radiofrance.com

