

Streaming Science Globally: CERN's Live Streaming Service

René Fernández Sánchez - CERN

April 2024

Presentation

Hi there! I'm René!

- Software engineer
- Working at CERN since 2015
- Working on Videoconference, Transcription and Translation and Web lectures services
- Currently Live Streaming Service Manager

Luarca, Spain



I'm from here!

Table of contents



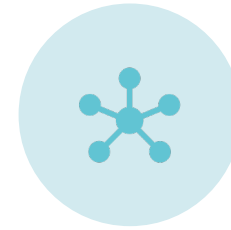
1. A BRIEF HISTORY
OF THE WEBCAST
SERVICE



2. ARCHITECTURE
AND
TECHNOLOGIES



3. OTHER
IMPORTANT PIECES
IN OUR SYSTEM



4. INTEGRATION
WITH OTHER
SERVICES



5. CAPACITY AND
STATS



6. LESSONS
LEARNT



7. FUTURE
DIRECTIONS



8. CONCLUSIONS

1. A bit of history: From 1999 to 2010

First CERN webcast website: 1999

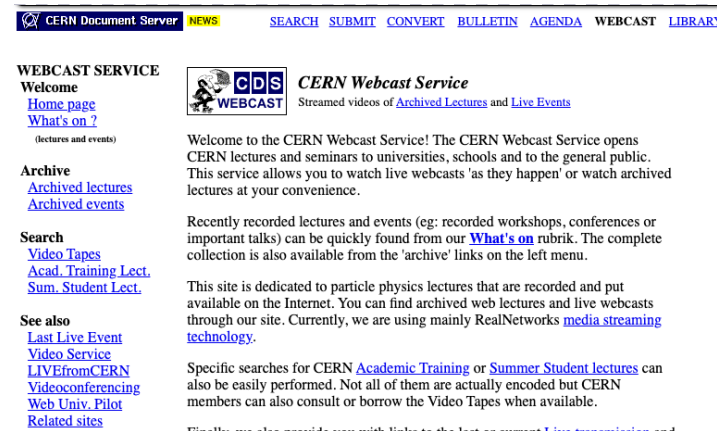
- Real Media, QuickTime Server, PHP...
- Apache Server
- Clicking links to open the player



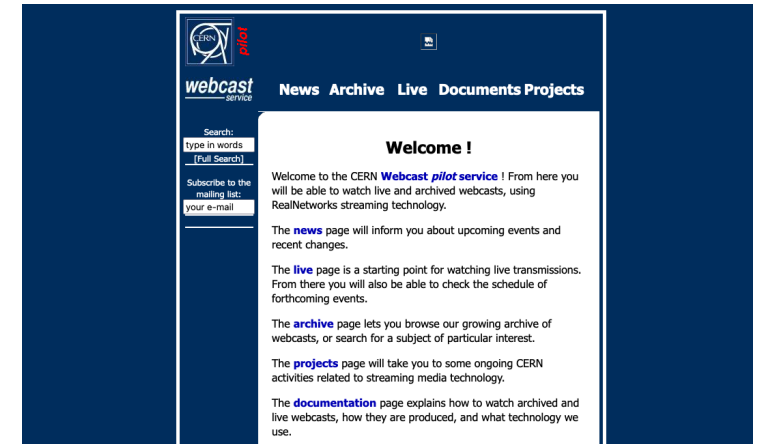
1999

In 2010:

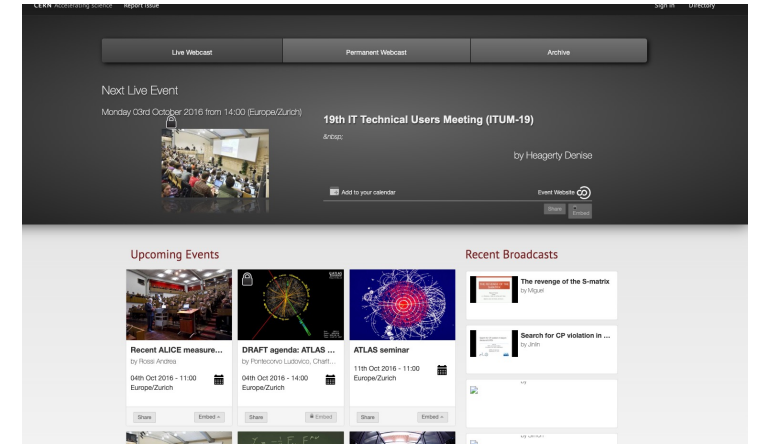
- Wowza Streaming Engine
- Still PHP
- Flash
- Microsoft IIS (Internet Information Services)



2003



2000



2010

1. Nowadays

✓ Wowza Streaming Engine

~~FLASH~~ - HLS (HTTP Live Streaming)

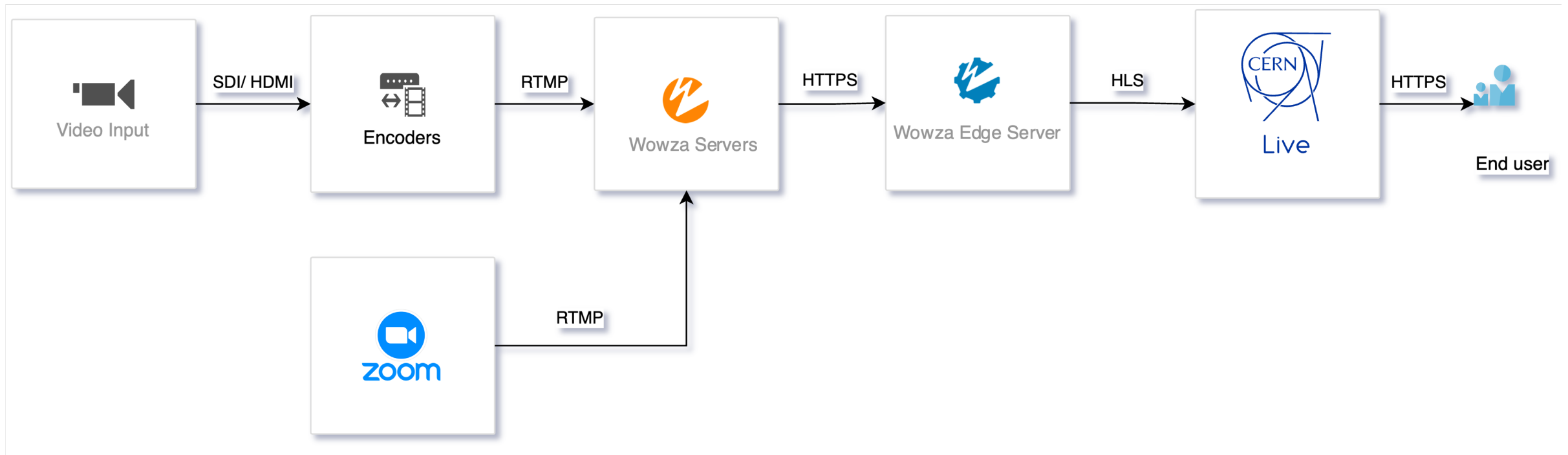
~~JS~~ - Openshift

~~PHP~~ – Python and React

The screenshot shows the CERN Webcast Website interface. At the top, there is a navigation bar with the CERN logo, 'Accelerating Science', and user information: 'Logged in as: Rene Fernandez Sanchez (fernanre)'. There are links for 'Logout' and 'Directory'. Below the navigation bar, the main content area features a large banner for a 'NOW LIVE' event. The event card for the '2024 CERN openlab Technical Workshop' includes the presenter 'Maria Girone', the date and time 'Tuesday 26 Mar 2024, 09:00 → Wednesday 27 Mar 2024, 17:00 (Europe/Zurich)', and the location '81/R-003C'. A 'Watch webcast >' button is present. Below the banner, there is a section titled 'UPCOMING LIVE EVENTS'. The first event is 'Measurement of $D^0D^0-\bar{D}^0D^0$ mixing and search for CP violation with $D^0 \rightarrow K^+\pi^-D^0 \rightarrow K+\pi$ -decays' by Roberto Ribatti, scheduled for Tuesday 26 Mar 2024, 11:00 → 12:15 (Europe/Zurich) at location 500/1-001. A 'More Information >' button is provided. The second event is 'ATLAS Weekly' by Manuella Vinciter, scheduled for Tuesday 26 Mar 2024, 14:00 → 15:40 (Europe/Zurich) at location 40/S2-C01. The ATLAS logo is visible next to the event title.

2. Architecture

2. Architecture in a nutshell

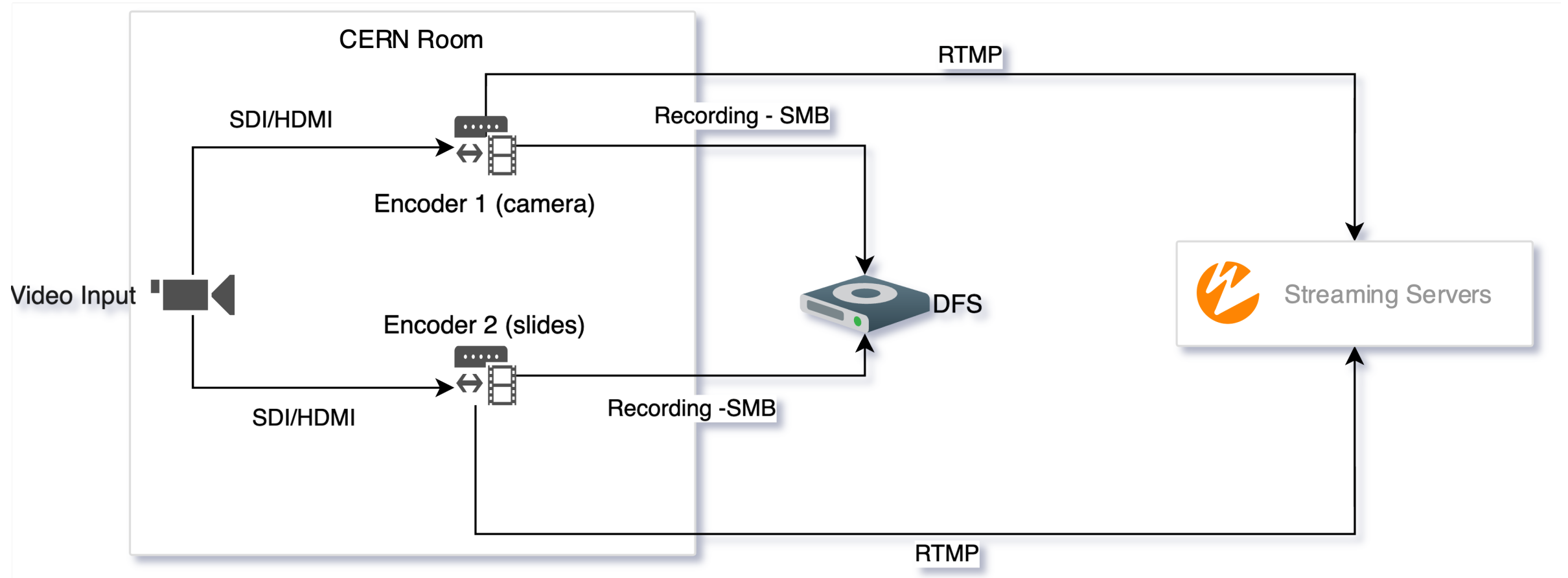


2. Architecture Zoom In: Input and Encoders (1/2)

- Currently support for 21 CERN rooms integrated with Indico
- Input: SDI/HDMI
- Output: RTMP
- Encoders:
 - Epiphan Pearl Nano
 - Matrox Monarch HDX (phasing out)
 - Matrox Monarch HD (phasing out)
- Devices allow not only streaming but also recording

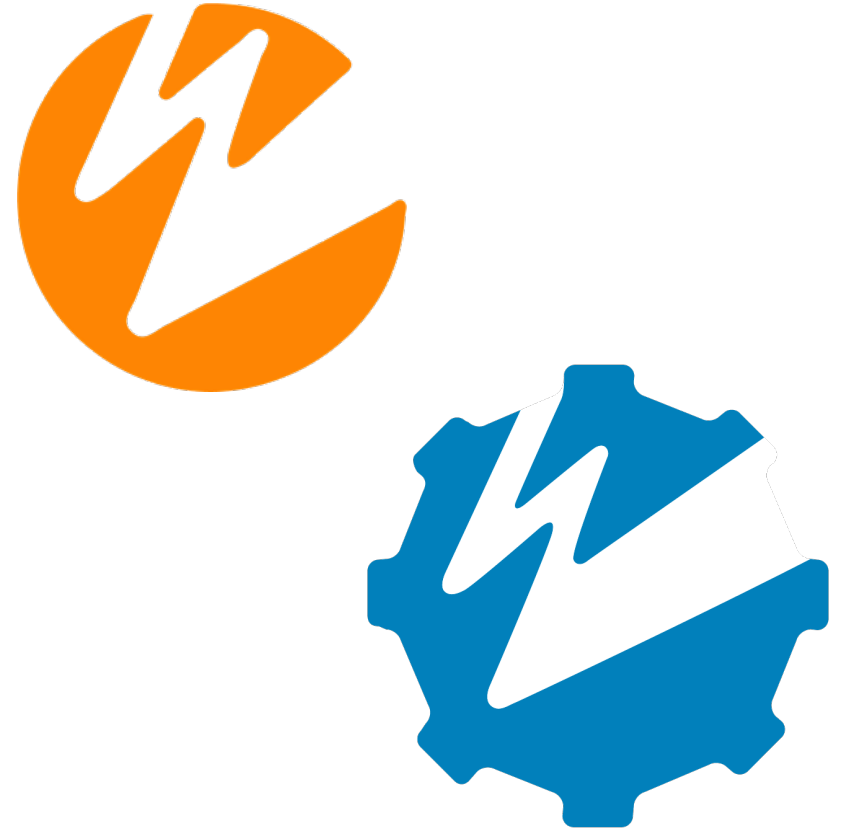


2. Architecture Zoom In: Input and encoders (2/2)

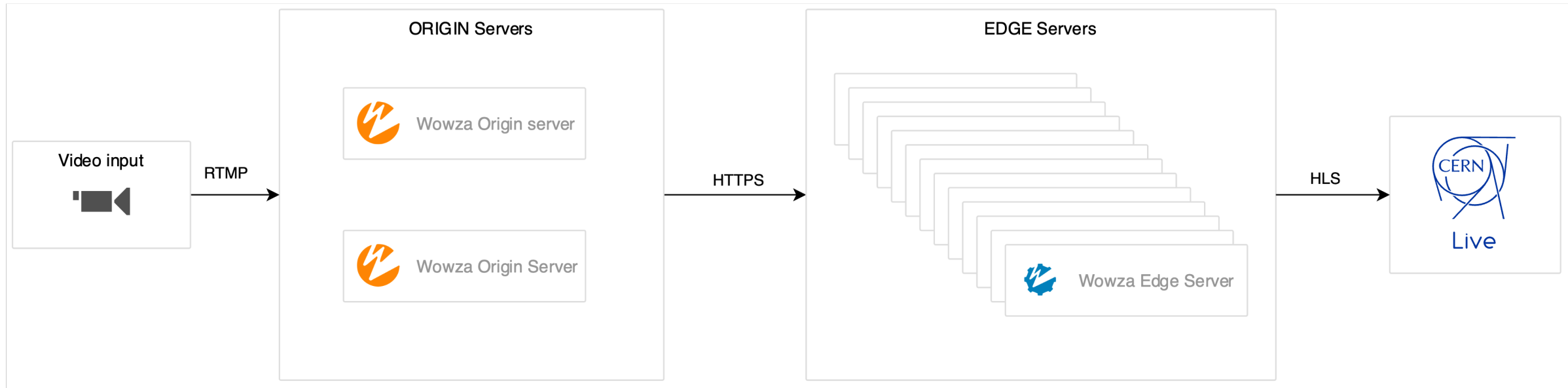


2. Architecture Zoom In: Streaming servers (1/2)

- **WOWZA Streaming Engine**
 - Streaming infrastructure run on-premise
- **Puppet Managed servers**
 - Centos 7 -> RHEL 9
- **Origin Servers:**
 - Bare metal machines
 - In charge of the transcoding of the input signal
 - 2 servers: 1 primary / 1 backup
 - Input: RTMP
- **Edge Servers**
 - Middlemen between the origin server and the client
 - Pull the stream chunks from the origin servers if they don't have them yet.
 - Clients connect to these servers
 - 15 servers (Openstack Virtual Machines)
 - Output: HLS
- **Support for other clients: LST telescopes or the NP detectors.**



2. Architecture Zoom In: Streaming servers (2/2)



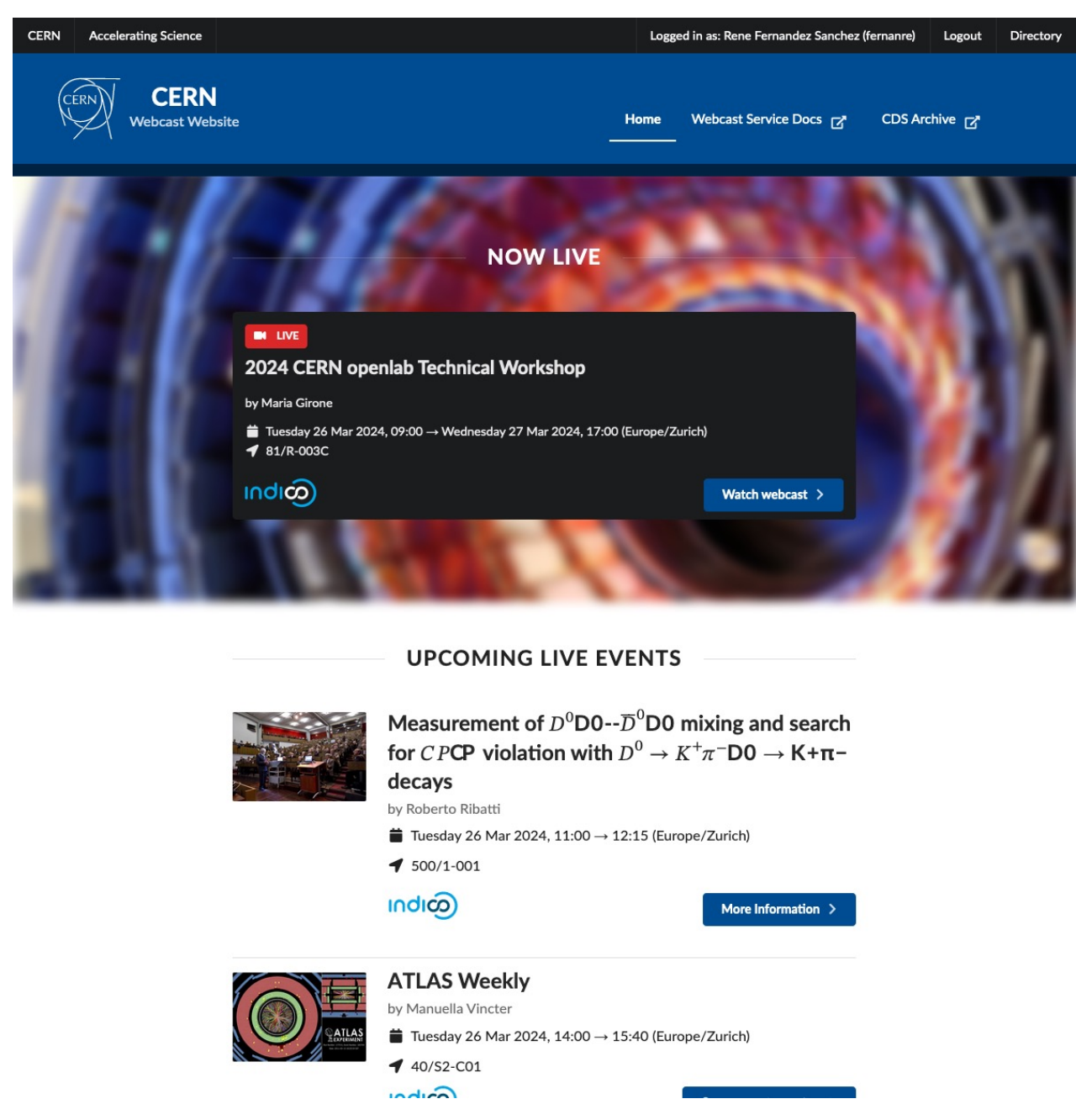
3. Architecture Zoom In: Live Events Website (1/2)

Website:

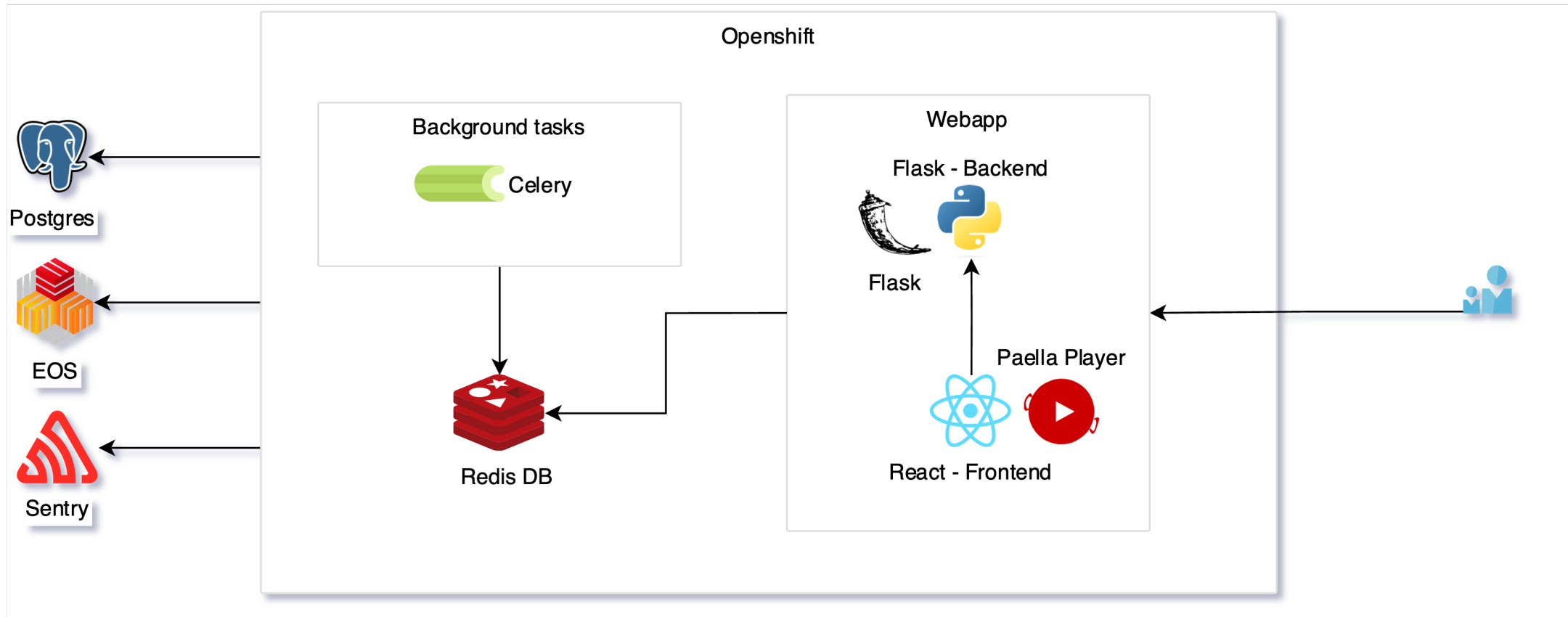
- **Backend:**
 - Python (Flask)
 - Postgres
- **Frontend:**
 - React

Technologies:

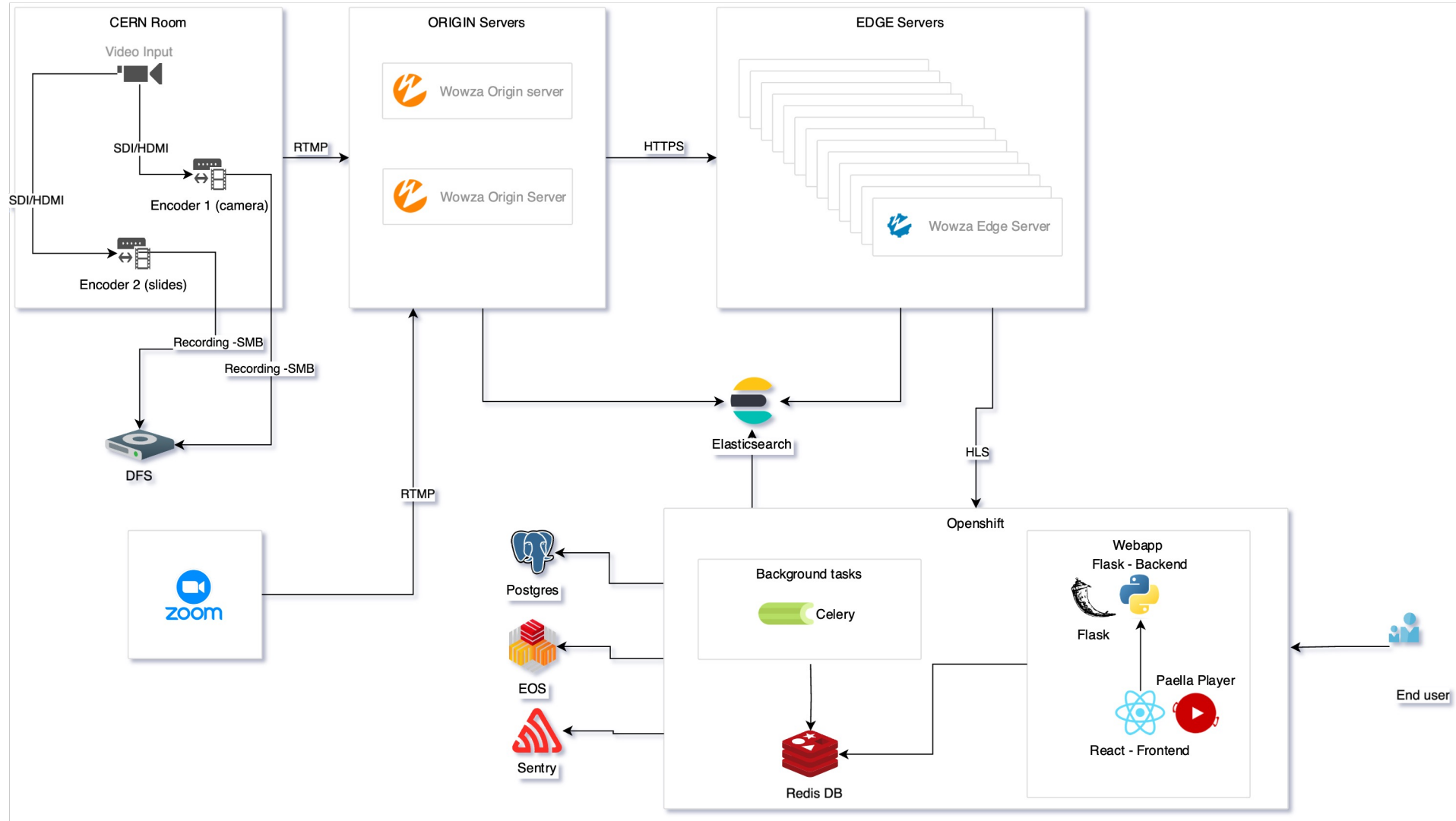
- Server: Openshift (Containers)
- Authentication: OpenID (Keycloak)
- Registry: Harbor
- CI/CD: Gitlab CI



2. Architecture Zoom In: Live Events Website (2/2)



2. Architecture Zoom Out



3. Other Important pieces of our system

3. Live Stream Player: Paella Player v7

- Paella Player v7
- Opensource player
 - Developed by UPV (Universitat Politècnica de València, Spain)
- Multiple layouts
- Multiple qualities
- Dual streams
- Used as well for VOD and as Opencast default player

3. Live events backup website

Static website created with Astro

When an event is updated in the main website, we create its homonym in the fallback website.

- We use the Gitlab API to create a commit in the repository
- The commit triggers the CI that builds the site

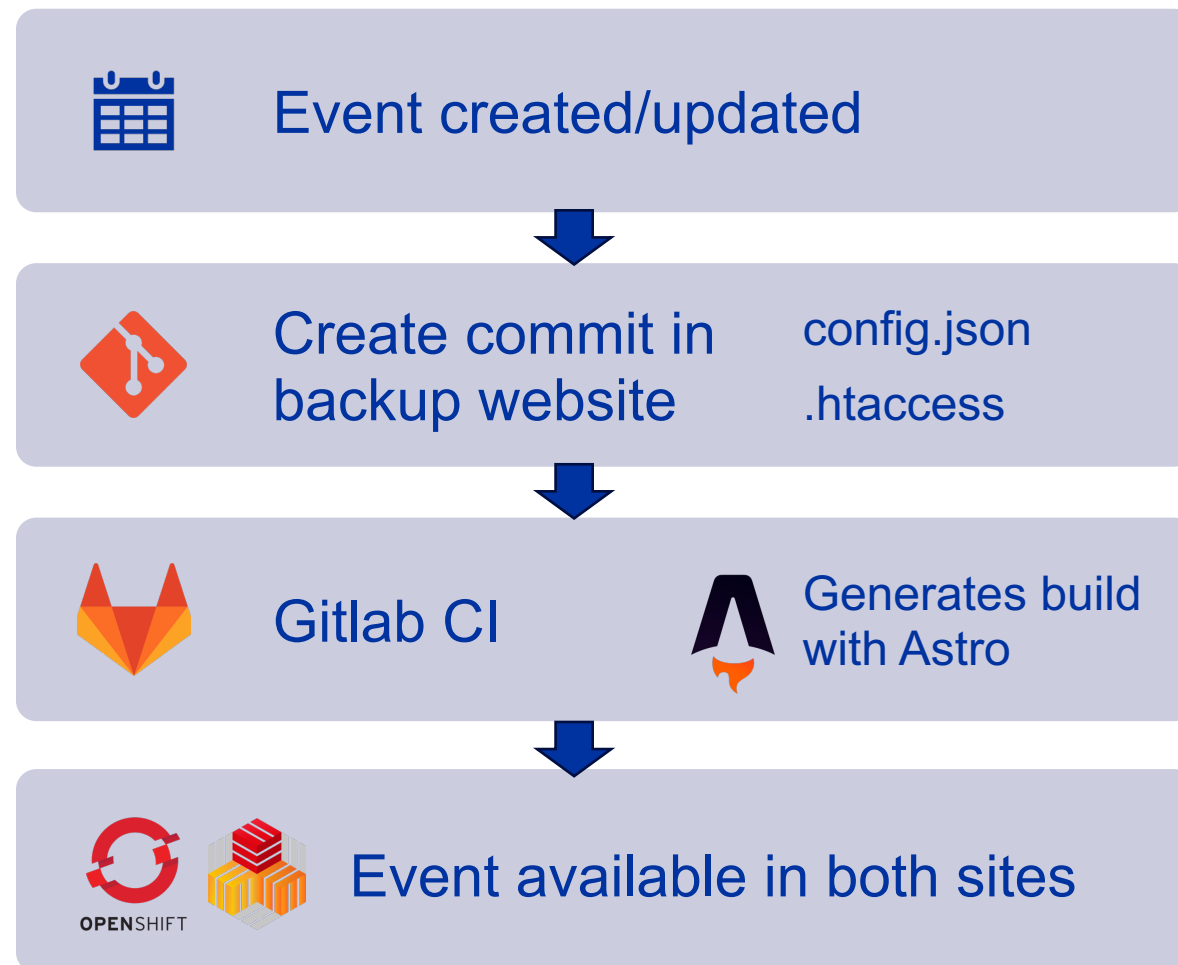
Completely static (HTML + JS)

Hosted in EOS

We can disable authentication manually if needed

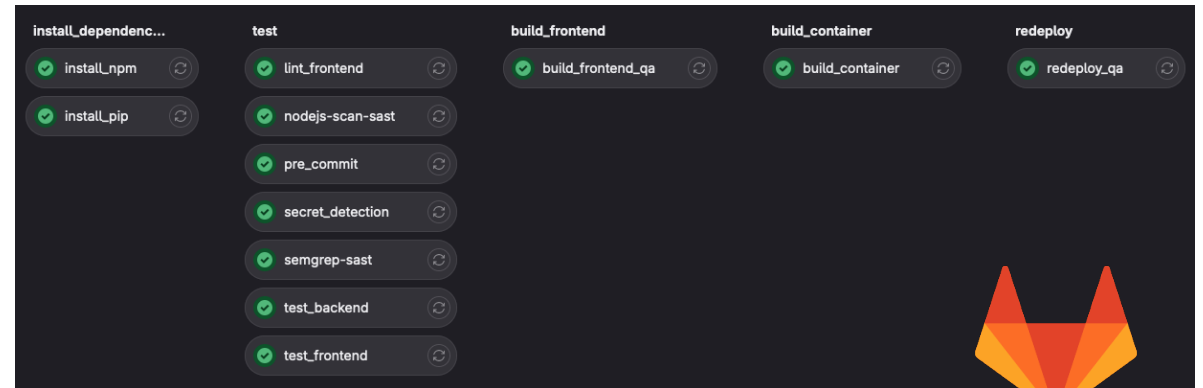
We keep the same URL structure to redirect our users transparently from OpenShift.

- <https://live.cern.ch/event/i12345>
- <https://webcast-lite.web.cern.ch/event/i12345>
- We use Nginx



3. Gitlab CI/CD overview

- **CI improved our productivity**
- **Workflow:**
 - **Dependencies:**
 - Using Poetry and Npm to manage and cache Python and Node dependencies.
 - **Test and code quality:**
 - Using Pre-commit, ESLint, Flake8 and Mypy.
 - Searching for secrets and vulnerabilities.
 - Test
 - **Build and Deployment:**
 - Building Docker images
 - Storing images in Harbor
 - Deploying them in Openshift



CI/CD

4. Integration with other services

- **Recordings / Weblecture Service**
 - CES: Tool used to manage the encoders and the recordings for VoD
 - Fully integrated with Indico
 - Otherservices: Indico, CDS and CDS Videos
- **Zoom / Videoconference Service**
 - Zoom allows to set up a custom live streaming service that is integrated with our streaming servers.
 - It was widely used during the pandemic.
- **Transcription and Translation / Transcription and Translation Service**
 - Offline captions: Generated by pilot service: TTaaS. Custom model adapted for HEP, trained with ~500k publications from the CERN Document Server (CDS)
 - Live captions: Not (yet) available. Investigating how to integrate them with Wowza.
 - Very good feedback from our user community
 - 2 official languages: English and French

5. Numbers!

5. Monitoring

Using ELK stack

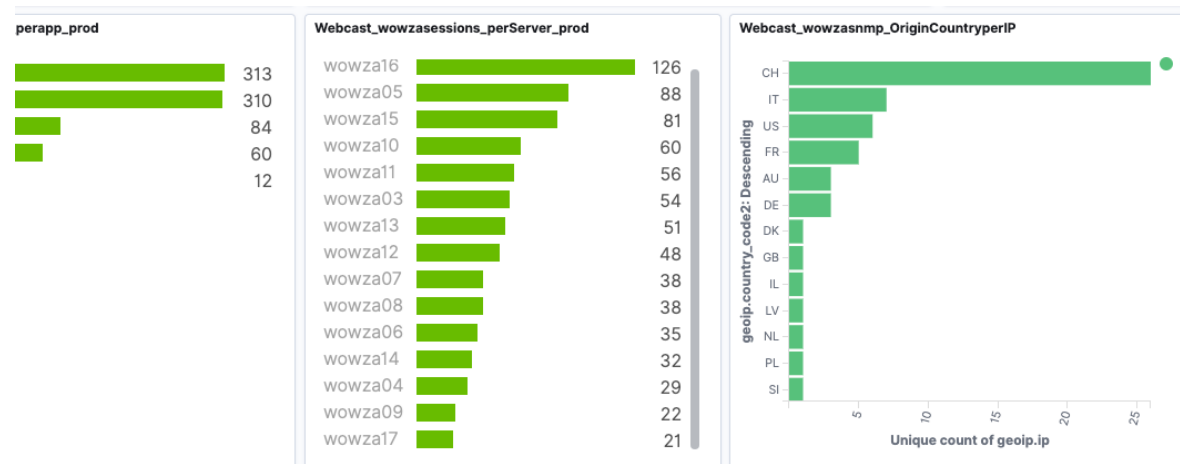
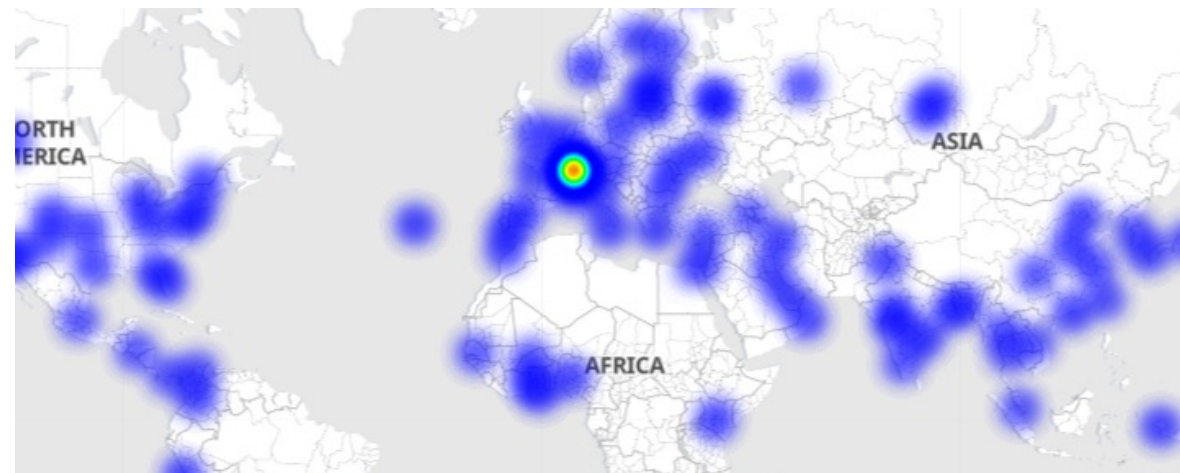
- Elastic
- Logstash
- Kibana

Internal usage

- Memory, Disk, CPU, Server Usage
- Countries

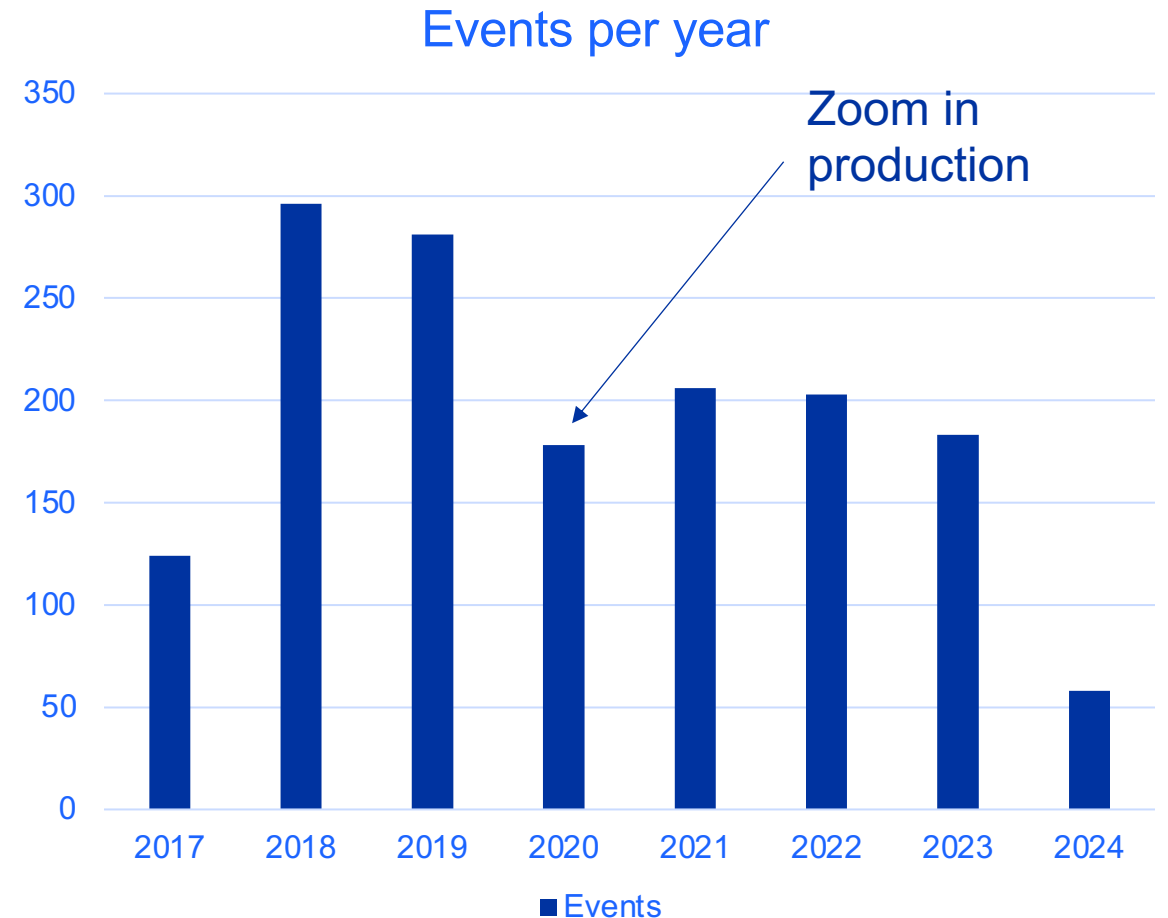
Public usage (Grafana)

- Unique viewers per event



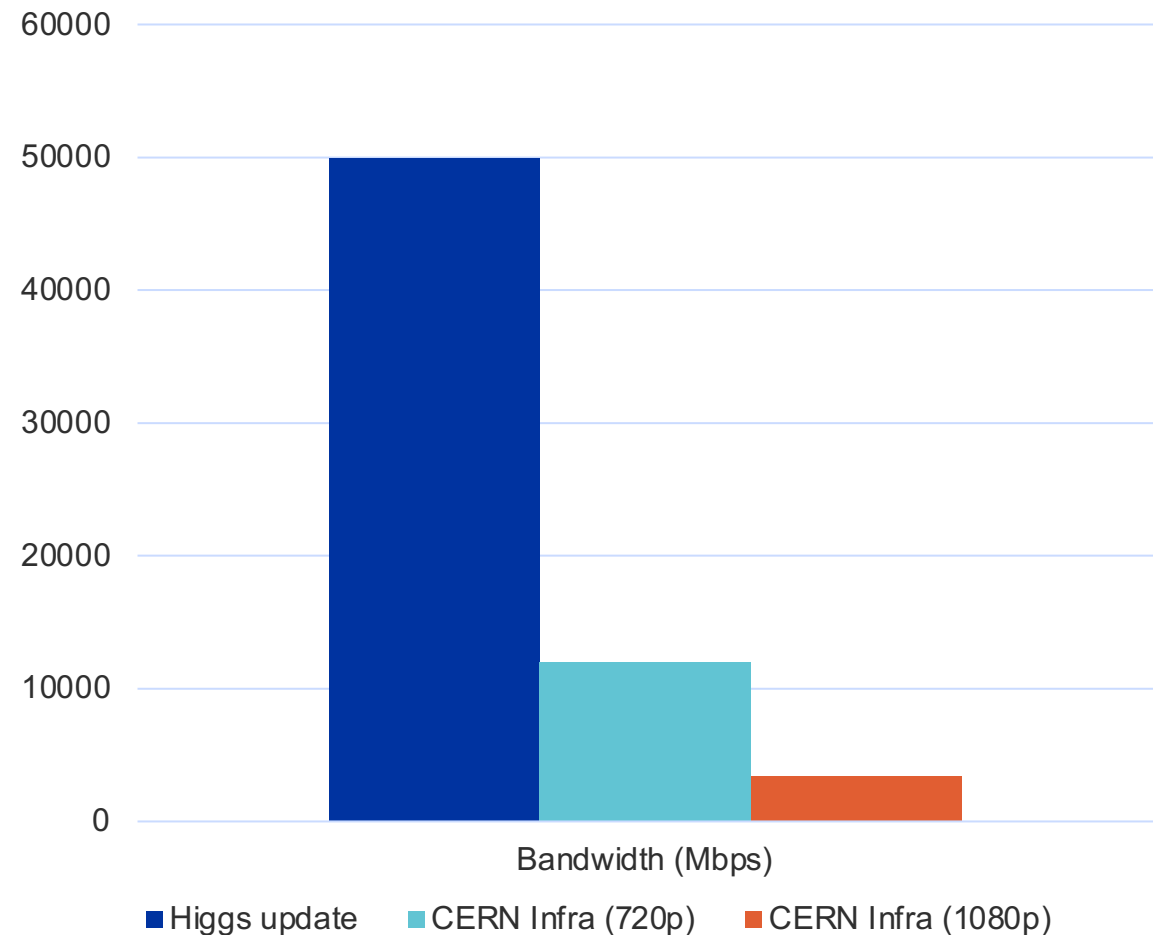
5. Live Events per year

- **Live events**
 - ~200 per year.
 - Peak in 2018 with almost 300.
- **Unique viewers**
 - Numbers can vary a lot depending on the event.
 - From 20 people connected to ~5000.
- **Events most watched of the last 2 years**
 - 10th Anniversary of Higgs Discovery.
 - ~5000
 - DG's presentations to personnel.
 - ~2000
- **Relying on live streaming for larger and more visible events**



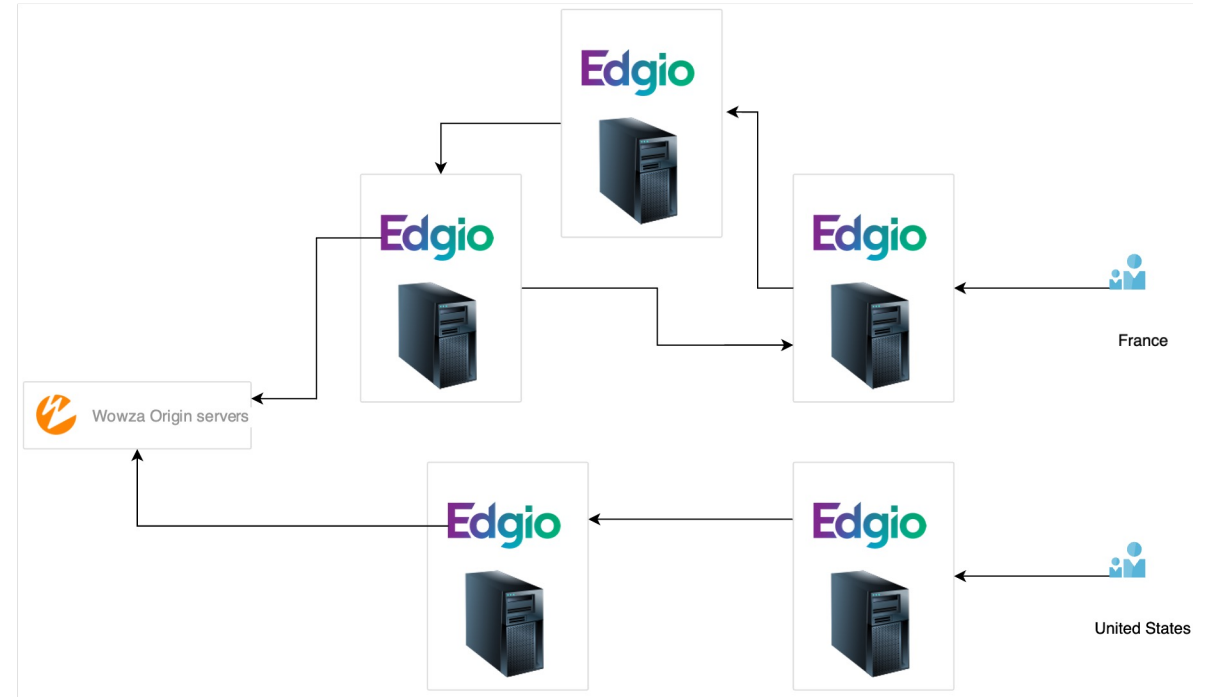
5. Capacity

- **~ 3400 - 6000 users with our infra**
 - ~ 10Gbps bandwidth
- **We measure our capacity using Apache Jmeter**
 - Reference: 1080p ~ 3Mbps
- **Peak:**
 - During the Higgs Boson 2012 announcement
 - 50k simultaneous connections
 - 500k unique viewers
 - 40 Gbps bandwidth peak
- **With our CDN, we don't have to worry**



5. CDN (Content Delivery Network)

- **Using Edgio**
 - Instead of connecting to the edge servers, clients connect to Edgio servers
 - Edgio has more than 300 pop (Points of presence) around the world.
- **Events expecting high traffic or from different locations in the world.**
- **Reduces latency if clients are far from origin.**



6. Lessons learnt



Scalability must be fast

Openshift auto scale doesn't fit our use case.



A chain is only as strong as its weakest link

If a dependency fails, the whole system can follow.



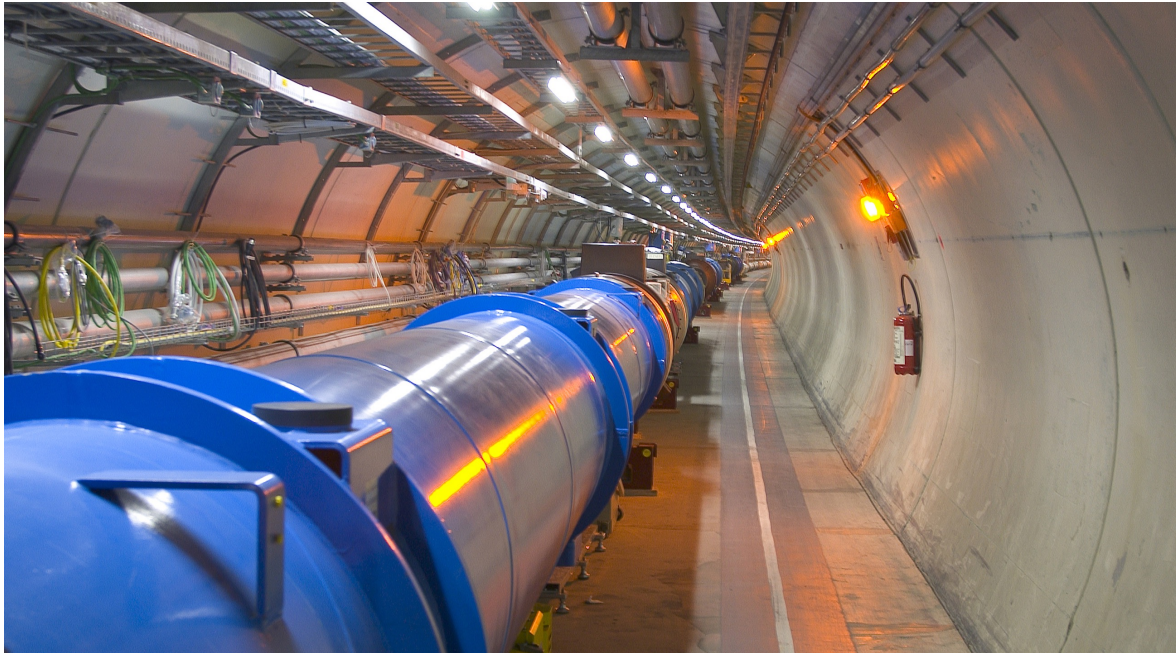
Live streaming events can be very challenging

Not much time to fix errors
Everything must be well tested beforehand.
Response must be fast!

7. Future directions

- **Goodbye Webcast, hello Live Streaming**
 - Rebranding after 20 years to align with industry.
- **Moving to 1 stream**
 - Using a composite view. Less bandwidth consumption. More scalable. Align with industry.
- **Live transcriptions**
 - Science must be accessible. Studying the possibility of automatic close captioning.
- **Other improvements:**
 - Secure tokens: Used to secure the stream distribution.
 - Moving to Load Balancing as a Service: Aim to improve the stability with the streams and ensure affinity.
 - Low Latency HLS (LLHLS): ~5 seconds instead of 30. Useful for monitoring or Q&A

8. Conclusions



Very stable: Service with more than 20 years.

Goal: Aimed at high visibility broadcasts

On-premise: Full control and privacy

Future: Greater accessibility and reliability

Thank you!



home.cern