CES (Central Encoding System) and the CERN Video Player

René Fernández Sánchez – IT-CDA-IC

6/21/21
CES
(Central Encoder System)
1/2
What is CES and why do we need it?

- At CERN, we have more than 40 encoders in webcast adapted rooms.
- All these encoders need to be managed, either individually or from a Central place.
- CES (Central encoder System) allows us and the operators to handle all these devices from a single place.
- We use Indico as a single source of truth to keep track of the event’s information.
- This is an application for operators
What was CES doing until now?

1. Fetch events from Indico
2. Assign these events to rooms where encoders are configured
3. Start/stop the webcasts and recordings on these rooms
4. Show preview of the running webcasts/recordings
5. Store the recordings in DFS
What can CES do now?

• All the tasks mentioned on the previous slide +
  1. Keep track of the processing status of an event and its contributions (Prev. Micala)
  2. Ingest into Opencast the recordings and their metadata for the postprocessing tasks (New)
  3. Publish to Indico and CDS the videos generated by Opencast (Prev. Micala)
  4. Import videos from a URL and from the Zoom Cloud Recordings (handy when there is a pandemic and all the recordings come from outside CERN) (New)
Why the change?

• Taking some features from Micala (legacy postprocessing system) and moving the transcoding tasks to Opencast
## Technologies comparison

- Standardizing technologies across section applications

<table>
<thead>
<tr>
<th></th>
<th>New CES</th>
<th>CES</th>
<th>Micala</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OS</strong></td>
<td>Centos 7/8</td>
<td>Windows 2012</td>
<td>Windows 2016</td>
</tr>
<tr>
<td>Python</td>
<td>3.X</td>
<td>2.7.4</td>
<td>2.7.10</td>
</tr>
<tr>
<td>Deployment</td>
<td>Openshift</td>
<td>VM</td>
<td>VM</td>
</tr>
<tr>
<td>Framework</td>
<td>Flask 1.2</td>
<td>Django 1.8.6</td>
<td>Pure python + Angular 1</td>
</tr>
<tr>
<td>Library for the tasks</td>
<td>Celery</td>
<td>Celery</td>
<td>-</td>
</tr>
<tr>
<td>Transcoding software</td>
<td>Opencast</td>
<td>-</td>
<td>FFmpeg, Sorenson, Handbrake</td>
</tr>
<tr>
<td>Authentication</td>
<td>OpenID</td>
<td>Shibboleth</td>
<td>Shibboleth</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Central Monitoring</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Features (1/3): Start/Stop events and monitor

1. Select the encoders to use
2. Clicking start will start all the encoders selected in one go
3. Monitor the events and stop the encoders
Features (2/3): Track processing and publishing status

- Several steps for all the postprocessing and publishing tasks
  - Example:
    - Ingest media package into Opencast
    - Update contribution’s ACL’s
    - Handle agreements
    - Publish to Indico and CDS
Features (3/3): Zoom Cloud recording integration

- Input the Zoom ID and download the zoom recording making it ready to ingest into Opencast. Better user experience for end users.
About the CERN Video Player

- Already used on the Webcast Website
- Based on a Opensource software: Paella Player (Already contributed with PRs)
- Backwards compatible with all the lectures on the media archive
- Support for dual streams out of the box (and more than 2 streams)
  - Synchronizes automatically both videos
  - Multiple view modes for camera and slides
- Fully integrated with Opencast
- Support for: Thumbnails, Subtitles/Transcriptions/ HLS, WEBRTC (soon) and auto bitrate
- Integrated with Matomo (Webanalytics) and OpenID
CERN Video Player: How it looks
CERN Video Player: OpenID + Apache integration
Questions?