Crowdfunding

Michael Stypa (plapadoo)
Lars Kiesow (ELAN e.V.)

December 13, 2018

Type	Talk (30min)
Target Audience	Everyone

Last year, ELAN e.V. and plapadoo initiated a successful crowdfunding initiative for updating and improving core elements of Opencast.

We saw this as an important task for the future of Opencast as it helps us to make some key components work with Java 9+ (something we could not avoid for long anyway) as well as improve the performance and ensure scalability of Opencast's scheduler.

This talk strives to give an overview about how this crowdfunding initiative came about and what the current status of the separate goals is. What has been updated already? What has changed or will change at all? How did we approach the tasks?

This talk will not go into any technical details. If you are interested in that, look out for related sessions in the technical track.

Update Apache Karaf

Apache Karaf is the OSGI library all of Opencast is based on. It provides us with a set of common libraries we can use and which are ensured to be secure, up-to-date and work together. To get these updated libraries and, arguably more important, to support Java 9, we need an upgrade to Karaf 4.2.x.

Scheduler Conflict Check Performance

Opencast allows to schedule recordings using the Administrative User Interface (Admin UI) or the External API. When new events are added, the system has to check for conflicts since two events cannot be recorded at the same time by the same capture agent.

Due to an architectural flaw, the checks performed by the system are currently very slow. The more events and capture agents you have, the more complex and slower this check gets. This leads to a slow, unresponsive UI on the one hand and can bring the whole system down on the other hand, in case someone creates many scheduled events within a short period. This can happen by accident or normal usage and represents a hidden risk for the availability of your Opencast installation.

Changing the underlying data structure requires reworking a fair bit of code. With scalability in mind, a new implementation will speed up the conflict checking significantly. This will improve user experience and availability on all sizes of Opencast installations.

Update Elasticsearch

Opencast is on Elasticsearch version 1.7.6. Again, this does not work with Java 9 and is furthermore not supported anymore. This means it will not get any future security updates or any other maintenance.

The Opencast community has already discussed that upgrading this is highly desirable and something that needs to be done in the long run. This not only ensures Java 9 compatibility but also serves as a backend for an updated player/portal and to replace the old Solr services. It was also one of the most discussed points in the recent meetings of Opencast's new high-availability working group.

The search indexes are what currently power (and limit!) all of our user interfaces. In case of Elasticsearch, its inability to run distributed currently makes it a single point of failure. That is a problem when you think about highly available systems. To put it bluntly: Upgrading Elasticsearch is one of the first steps we need to do when thinking about removing the admin node as a single point of failure.