Release 1 developer concerns

Based on work to date these are the issues that need time for proper investigation prior to a Release 1 pilot. Technical components relating to each issue are documented with links to the relevant modules. Previously this just encompassed concerns about scalability.

Management of large bitstreams

Right now video and audio clips are limited to 250MB. This is partially for testing purposes (so that encodes finish efficiently) and partially for disk concerns on the development server. Providing an artificial limit through the web interface is fine but there needs to be some mechanism for arbitrarily large bitstreams. An example might be an opera from the Jacobs School of Music. Uploading a 70GB file through the ingest workflow is cumbersome as well as likely to time out.

Currently Matterhorn preserves a master copy along with the derivatives which increases the need for extra storage.In addition we are also keeping a local copy within the hydrant head that requires large storage as well.

Needs

Identify tradeoffs between storage limitations and functionality of system

Identify consequences for long running Matterhorn processes

Identify performance load on the server when encoding complex video

Understand how to build a Dropbox like service for large files (SFTP, web, or something else?)

References

Master File source

Supported file formats

Audio and video formats are currently identified based on a combination of MIME type and file extension. While this works it is not very scalable especially for complicated cases like MOV. Other scenarios that have not been tested include different codecs within a container. For example an MP4 file could be encoded with an almost infinite number of x264 options. Some may work on a computer but not mobile while others may result in larger file sizes than needed.

This also ties into the need to extract technical metadata which are discussions to have to with the metadata team members.

Needs

Learn how to use a tool like MediaInfo to automatically determine content type

Determine a list of supported formats that are well documented for pilot sites

Identify minimally supported profiles for x264 and other codecs (ignoring edge cases)

Research which file formats are supported by the Matterhorn encoding process

References

FITS (identification tool)

MediaInfo (metadata extract tool)

Media Object controller

x264 encoding options

Mobile support

There has been very little testing of the current feature set on mobile devices. Even on different desktop browsers certain bugs are present. Since one of the Release 1 goals is a system that can be used in both environments some time is needed to identify problems, design solutions, and implement them.

Needs

Test discovery using the search and browse interface

Identify problem areas with playback (such as scrubbing, segment selection, etc)

Document known issues that won't be resolved before Release 1

References

Twitter Bootstrap framework

Ingest versus playback

Right now the system consists of a single interface. Most of the assumptions revolve around the needs of people loading content into the system as opposed to students, researchers, or others who are more interested in consumption of content. As developers we have discussed the idea of a separate hydra head that focuses more on this aspect.

Needs

Identify overlapping concerns between the needs of catalogers / digitizers and students / researchers

Estimate extra development cycles required to make an interface that works well on the desktop and mobile

Refactor reusable components so that work already completed can be reused (Engage player for example)

Figure out a useful approach for pagination of facets, search results, and other places where there could be many values to list

References

Access control wireframes

Policy manager

Search and discovery

Right now the interface lacks any pagination controls. On the home page for instance there is no way to browse more than the list of results presented in the 'My Items' and 'Recent Items' areas. Also faceted filtering is currently using the defaults instead of fitting into the needs of the Avalon application.

Search can only be done on a keyword level at the moment. This means that you can't limit to Title, Field, or items ingested this semester. The requirements for catalogers and digitizers are also different than those for instructors who just want to use material in a classroom. How should we handle these needs? Is a one size fits all approach going to work or do we need multiple ways of discovering content?

Needs

Knowing which facets are useful for different users (ingesters versus instructors)

Understanding how to efficiently navigate a large result set

References

Pagination examples (Yahoo UI pattern library)

Stanford shelf browse (in development)

Streaming availability of resources

At this point it is unknown how to handle serving up multiple streams at once and what the performance impact will be under heavy loads. It is assumed that any system should be able to stand up to peak loads around midterms and finals without grinding to a slow crawl.

There are also performance questions about securing a stream. What will be the impact of trying to verify access? Testing is needed to understand the implications of different approaches to security. Also unknown is how to handle embedding or remote linking from sources outside of Avalon.

Needs

Identify bottlenecks when streaming multiple content at the same time

Understand how to tune and document the media servers (Red5, Flash Media Server, and WOWza)

Bandwidth requirements and server load with multiple streams

Securing streams without increasing performance issues

Integration with services like Kaltura

References

Red5 (media server)

Flash Media Server (media server)

WOWza (media server)