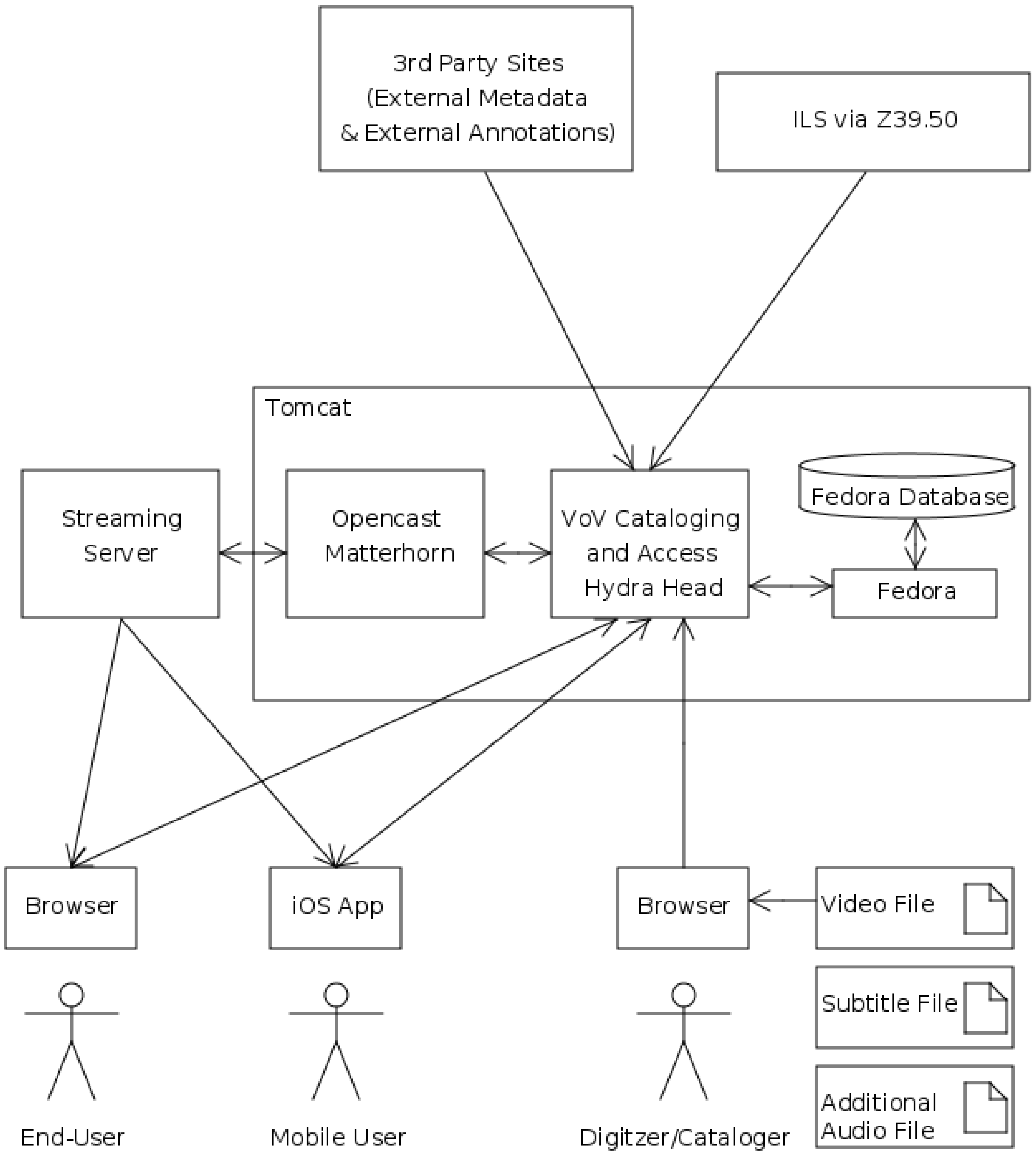
Architecture (Draft)

The picture below shows the draft architecture for Variations on Video.



Components

Opencast Matterhorn

Matterhorn would be used for transcoding and potentially additional analysis or transformation necessary to prepare a video for delivery. Matterhorn would probably need to be configured to run externally or on a different machine in a production setting.

See Technical Investigation - Opencast Matterhorn for some more discussion.

Streaming Server

The streaming server would serve the derivative files stored in the Video Data Store. Use of different streaming servers will be supported, but only two reference implementations, one commercial and one open source, will be provided. Red5 appears to be the best choice for open source streaming servers. It only supports RTMP streaming though. Among commercial streaming servers, Flash Media Server and Wowza stand out with Wowza supporting iOS devices while FMS only supports Flash. With respect to formats, AAC and H.264 in MP4 containers seems the best choice at the moment. Because it does not have the patent issues of H.264, VP8 might be a good choice if it matures before we begin implementation. See Streaming Servers and Video Formats for more information.

Fedora

An existing Fedora instance would be used to store the video metadata and potentially provide other available actions on the video content. See Technical Investigation - Fedora Commons for some more discussion.

VoV Cataloging and Access Hydra Head

This Hydra head would accept video input (optionally with additional subtitle, audio, and structural input) for ingest. The video and audio content would be passed along to the transcoder process and metadata would be stored in Fedora. Additionally, some level of cataloging could be done for the newly ingested item, potentially with a review workflow. The access tools would be a part of the Hydra / Blacklight search catalog that runs under Ruby on Rails. Ingest could be passed off to Fedora to handle. The user profile database would be embedded into this component or could be split out to an external database. Structural metadata could be serialized in METS or MPEG-7. At this time, METS appears to have greater traction in library applications, but MPEG-7 is used by Matterhorn.

ILS via Z39.50

Z39.50 could be used to pull in bibliographic data and associate an asset to a record in an institution's ILS. New work would need to be done to figure out the best way of extracting the necessary data from a MARC record for a video. Other protocols or other information sources should be considered to allow for flexibility when integrating into different library environments.

3rd Party Sites

3rd party sites could be used for additional metadata or for annotations either during the ingest/cataloging process or during an end-user's session with the web interface.

iOS App

An iOS application could be developed to reach iPhones and iPads. A browser-based option could potentially replace this option.

Video Player

Assuming RTMP will be the main protocol used for streaming video in VoV, the video player will need to support it. Also it would be good for the player to be easily adaptable for use on mobile devices. For these reasons, Open Video Player is a good choice as the starting point for our player. See Video Players for more information.

Open Questions

1. MPEG-7 or METS for structural metadata?
2. Flash Media Server or Wowza for the commercial streaming server?
3. How tightly integrated should Matterhorn be? Should we allow switching it out for another media processing pipeline like Kaltura or a local solution?
4. Do we need a layer of abstraction between the Hydra head and Matterhorn?
5. Is the Hydra head the best place to put additional web services needed for the player and iOS app?
    - It might be easiest to put these additional web services and the Matterhorn/media processing pipeline abstraction layer into a separate webapp.
6. Does it reduce system complexity or increase it by running Matterhorn and Hydra under Tomcat?

Use Cases

When a digitizer wants to add a container to their collection, they will first prepare the video (or audio) file(s) and any ancillary materials associated with the container. Using their browser, they will log into the VoV Hydra head and use its ingest forms to add the files and associated descriptive, structural, and technical metadata as necessary. The metadata will be stored in a Fedora item and the video (or audio) files will be sent off to Matterhorn for encoding and analysis via its workflow service. When Matterhorn finishes its workflow, it will distribute the derivative files (flavors) to the streaming server(s) and to Fedora along with any metadata derived from analysis. Optionally, descriptive metadata can be sent to the institution's discovery system. After ingest, adding metadata and editing existing metadata can be performed in the Hydra head. If a video (or audio) file is corrupt or low quality, the original file can be uploaded again or it can be sent through Matterhorn's encoding and analysis services again.

When a user wants to find a container, they would use the institution's discovery system if available and Hydra's Blacklight discovery system otherwise. Once found and passing authorization checks, the user can play the container and create bookmarks that are stored on the server. Additionally, video and audio from VoV can be made available to other tools via an HTML embeddable player (see unAPI and oEmbed). The embeddable player will have a way for the user to authenticate and will perform the necessary authorization checks.