matterhorn

Building services to support lecture capture, media processing, and distribution

March 8, 2010

The Opencast planning grant was funded by the William and Flora Hewlett Foundation and the Andrew W. Mellon Foundation

Opencast Matterhorn project is funded in part by the Andrew W. Mellon Foundation with William and Flora Hewlett Foundation board with substantial contributions from the Matterhorn Partner Institutions

Agenda

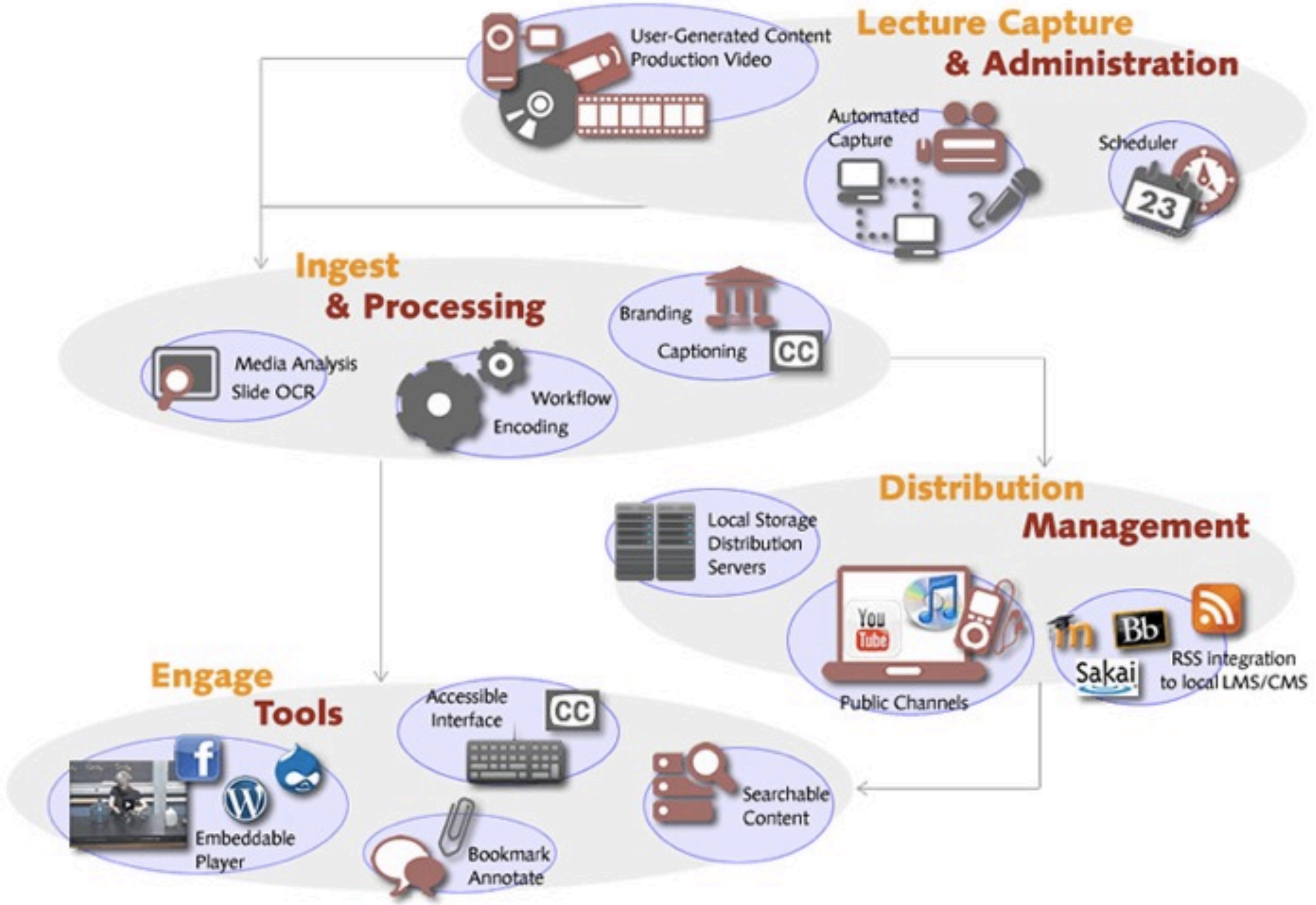
Scope

Services

Implementation

Roadmap

Scope


Lecture Capture & Administration
User-Generated Content Production Video
Automated Capture
Scheduler
Ingest & Processing
Branding
Captioning
Media Analysis Slide OCR
Workflow Encoding
Distribution Management
Local Storage Distribution Servers
Public Channels
RSS integration to local LMS/CMS
Sakai
Engage Tools
Accessible Interface
Embeddable Player
Bookmark Annotate
Searchable Content

Services

Capture Agent	Capture agent control and confidence monitoring
Scheduling	Produces iCalendar feeds for agents
Ingest	Temporary storage for media and metadata
Media Inspection	Extracts technical metadata (e.g. codecs, formats, bit rates) from media
Caption Handling	Caption file handling and conversion
Media Composer	Media encoding, transcoding, muxing, captioning
Media Analysis	Chapter detection, video OCR, speech-to-text
Distribution	Sends media and metadata to various channels (e.g. iTunes, youTube)
Search	Find media based on static and time-based metadata
Workflow	Orchestrates media processing and distribution services

Services

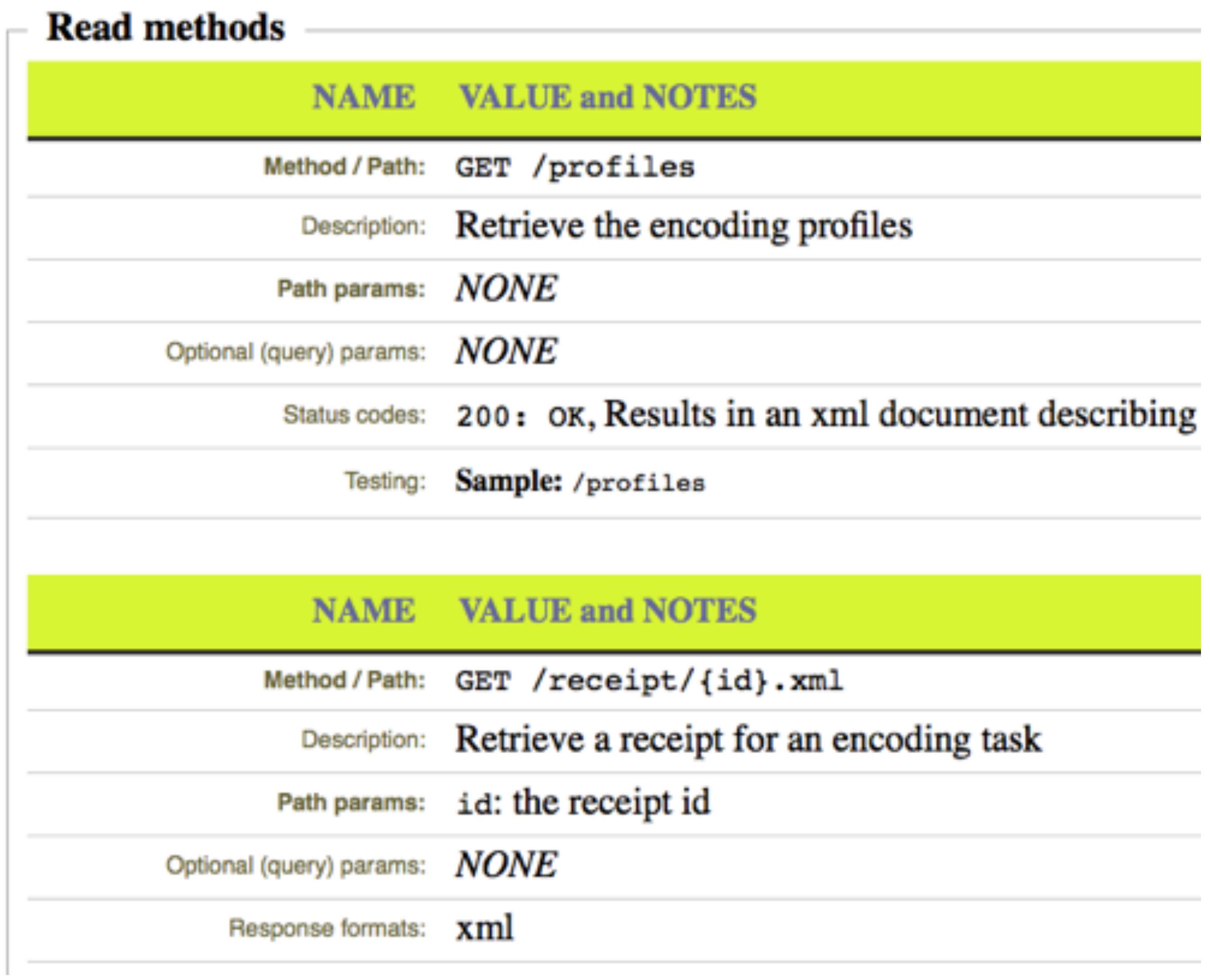
- Available as Java APIs and REST endpoints
- Multiple implementations
- Flexible deployment options

COMMUNITY | PROJECT
www.opencastproject.org

Service Contracts

Each REST contract is described in two ways:

[illegible]

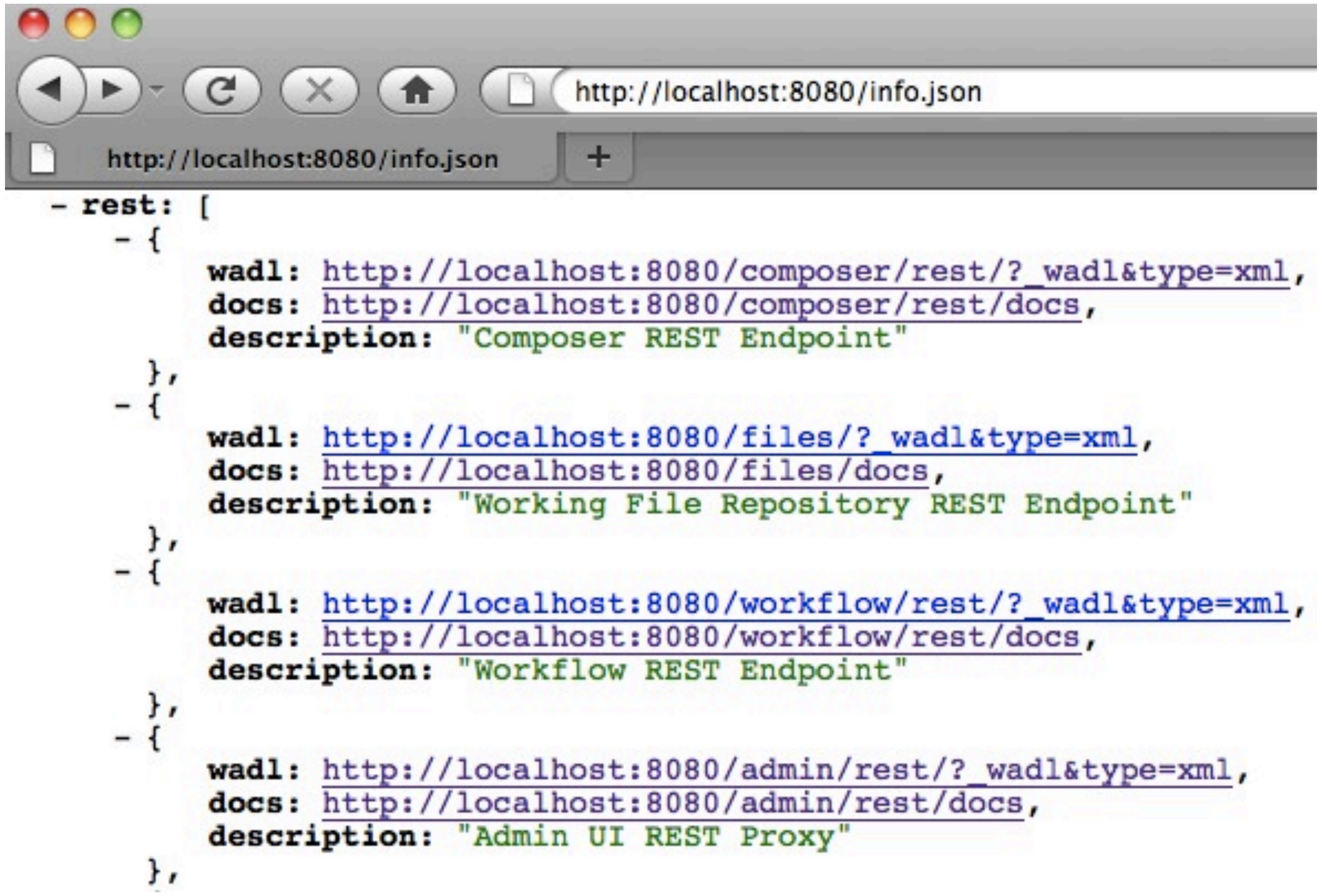

Read methods

NAME	VALUE and NOTES
Method / Path:	`GET /profiles`
Description:	Retrieve the encoding profiles
Path params:	*NONE*
Optional (query) params:	*NONE*
Status codes:	`200: OK`, Results in an xml document describing
Testing:	**Sample:** `/profiles`

NAME	VALUE and NOTES
Method / Path:	`GET /receipt/{id}.xml`
Description:	Retrieve a receipt for an encoding task
Path params:	`id`: the receipt id
Optional (query) params:	*NONE*
Response formats:	xml

Service Contracts

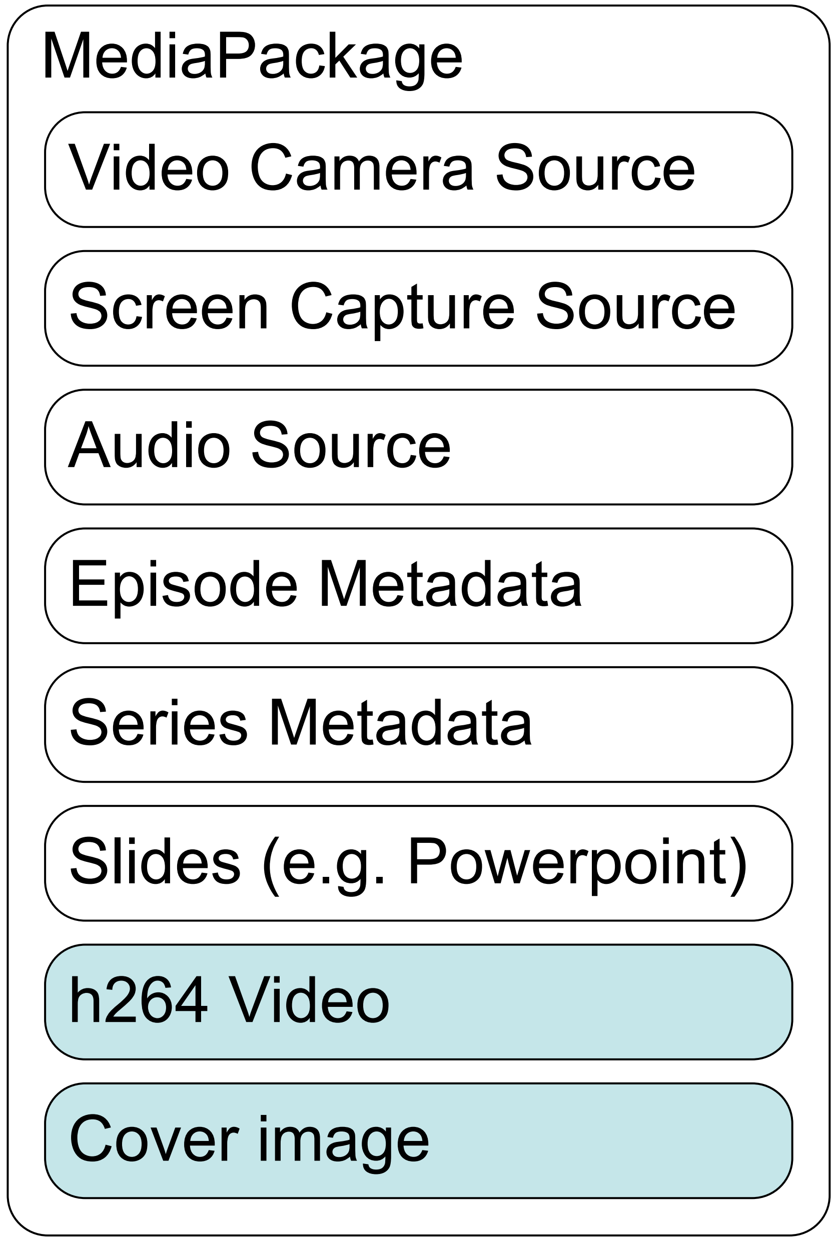
Each endpoint is discoverable and self describing


http://localhost:8080/info.json
http://localhost:8080/info.json

```
- rest: [
   - {
        wadl: http://localhost:8080/composer/rest/?_wadl&type=xml,
        docs: http://localhost:8080/composer/rest/docs,
        description: "Composer REST Endpoint"
     },
   - {
        wadl: http://localhost:8080/files/?_wadl&type=xml,
        docs: http://localhost:8080/files/docs,
        description: "Working File Repository REST Endpoint"
     },
   - {
        wadl: http://localhost:8080/workflow/rest/?_wadl&type=xml,
        docs: http://localhost:8080/workflow/rest/docs,
        description: "Workflow REST Endpoint"
     },
   - {
        wadl: http://localhost:8080/admin/rest/?_wadl&type=xml,
        docs: http://localhost:8080/admin/rest/docs,
        description: "Admin UI REST Proxy"
     },
```

Service Contracts

- Produce and/or Consume MediaPackages
  - Media tracks (e.g. audio, video)
  - Metadata catalogs (e.g. Dublin core, IEEE LOM, MPEG-7)
  - Attachments (everything else)
- Services add MediaPackage elements
  - Transcoded tracks
  - Time-based metadata
  - Images (e.g. cover art, scene detection)



opencast COMMUNITY | PROJECT www.opencastproject.org

MediaPackage (Simplified)

```
<mediapackage>
  <media>
    <track id=”track-1” type="presenter/source"><url/></track>
  </media>
  <metadata>
    <catalog type=”mpeg7/audio” ref="track:track-1"><url/></catalog>
  </metadata>
  <attachments>
    <attachment type="cover" ref="track:track-1"><url/></attachment>
  </attachments>
</mediapackage>
```

Implementation

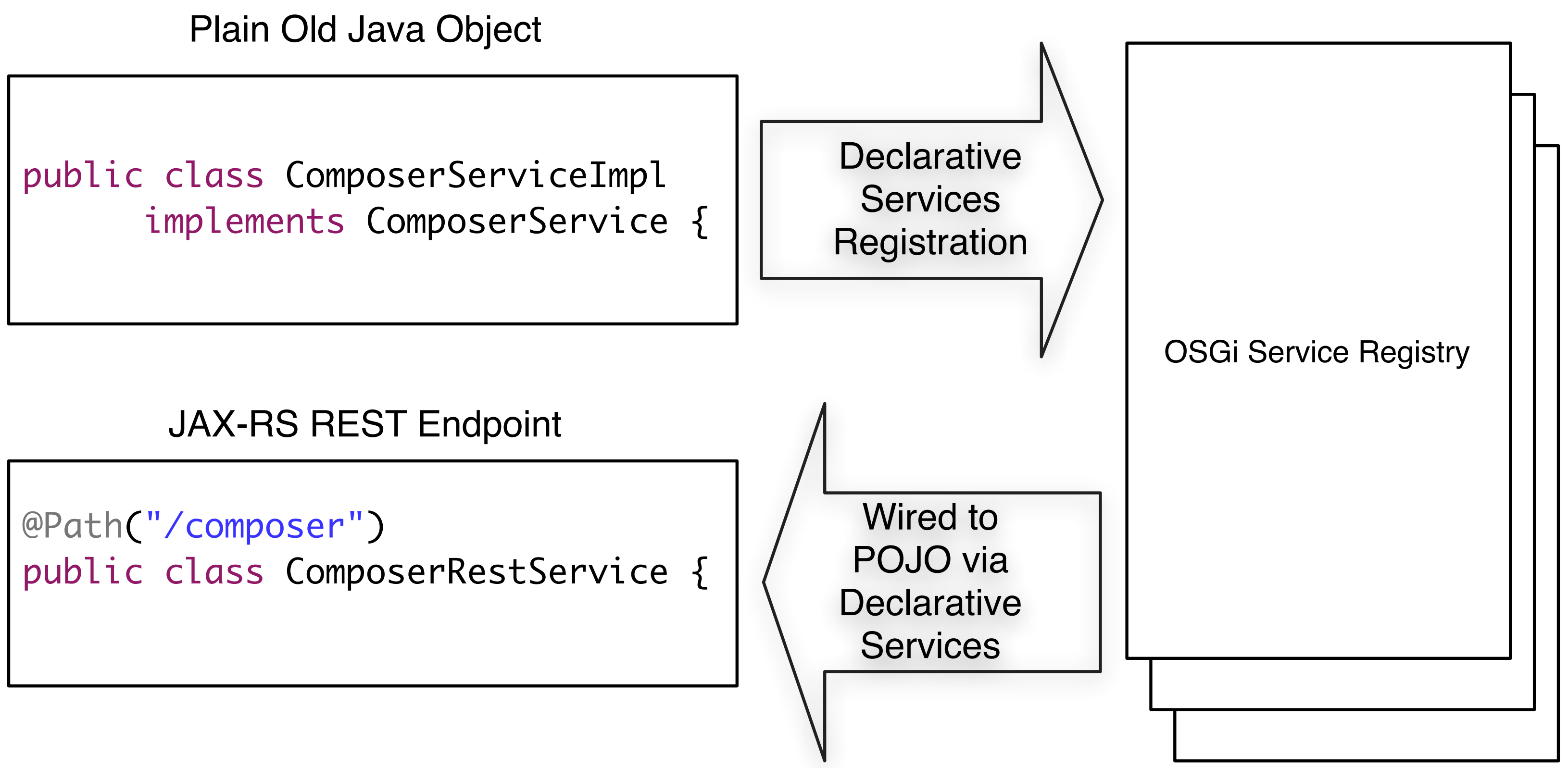
Server-side

- OSGi -- *dynamic module system for Java*
- JAX-RS -- *RESTful services*
- JAXB -- *Data binding for XML, JSON, fastinfoset, etc.*
- JPA -- *persistence to relational databases*

Client-side

- XSLT
- jQuery
- Flash
- JS-Flash bridge

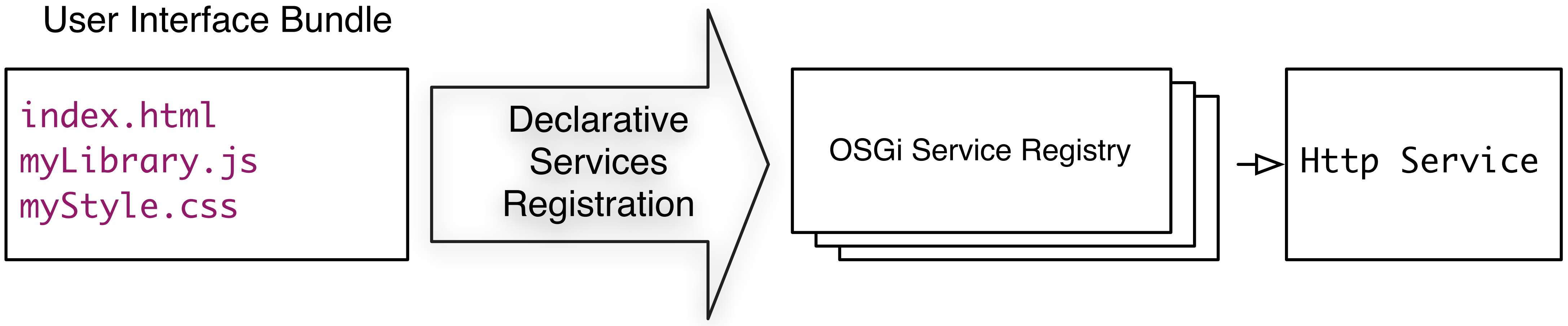
Implementation (Service)


Plain Old Java Object
Declarative Services Registration
OSGi Service Registry
JAX-RS REST Endpoint
Wired to POJO via Declarative Services

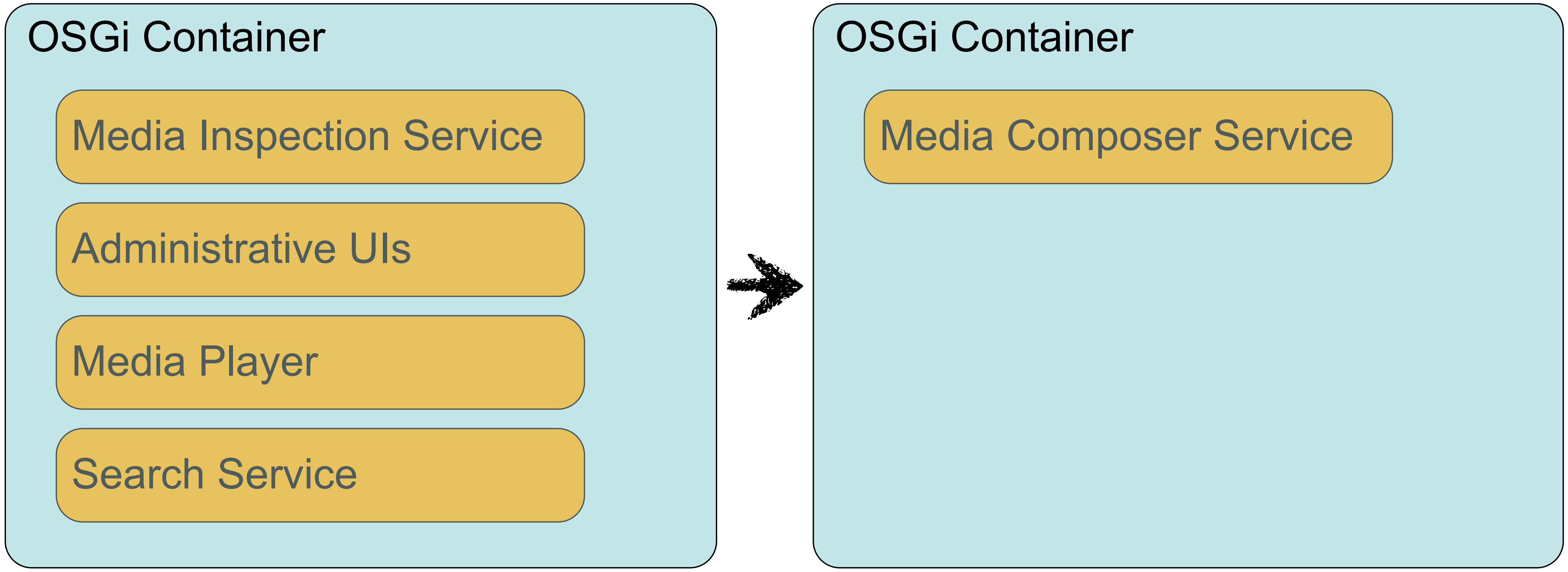
```
public class ComposerServiceImpl
      implements ComposerService {
```

```
@Path("/composer")
public class ComposerRestService {
```

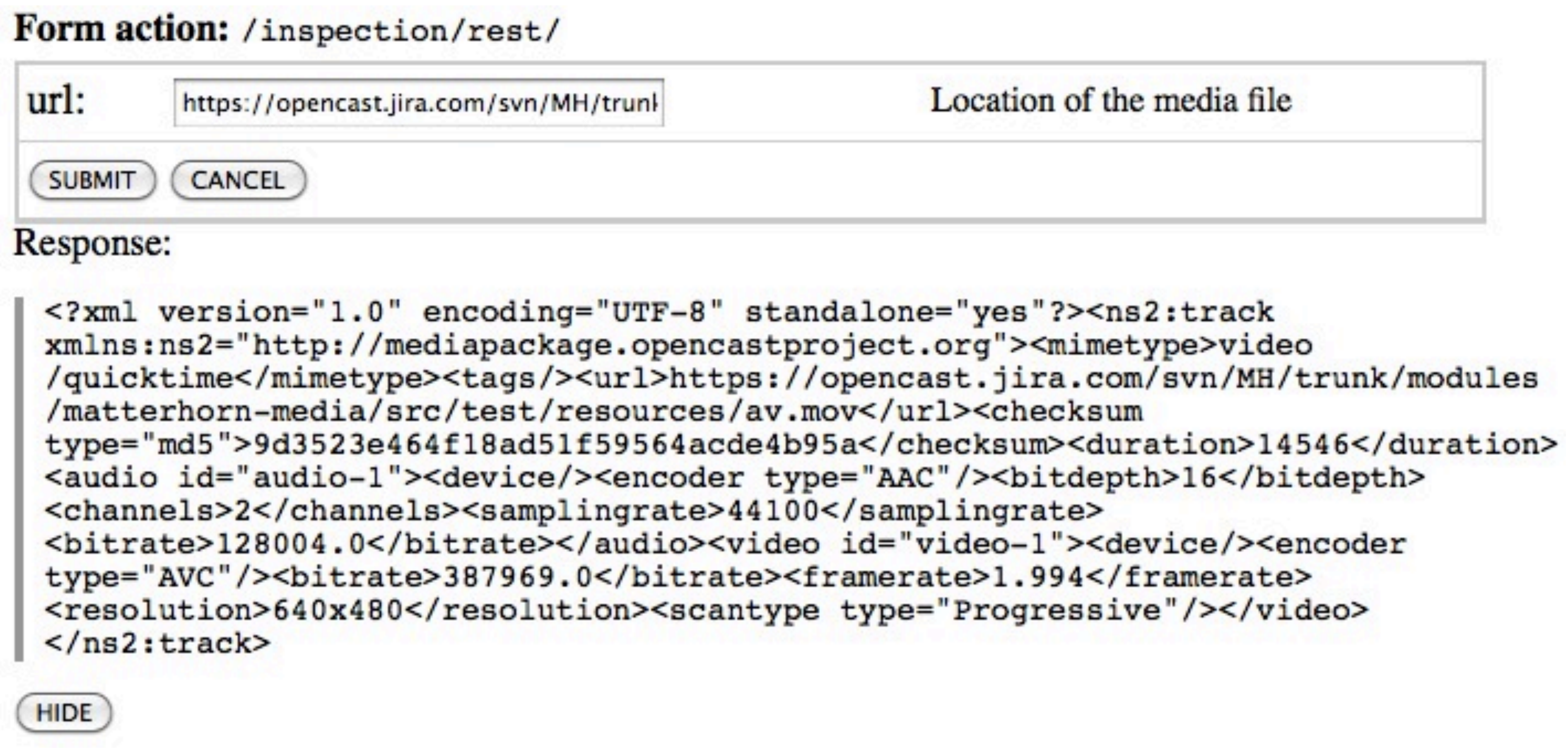
Implementation (User interface)



Implementation (deployment)



Examples: Media Inspection



Form action: /inspection/rest/

url:	https://opencast.jira.com/svn/MH/trunk	Location of the media file

SUBMIT CANCEL

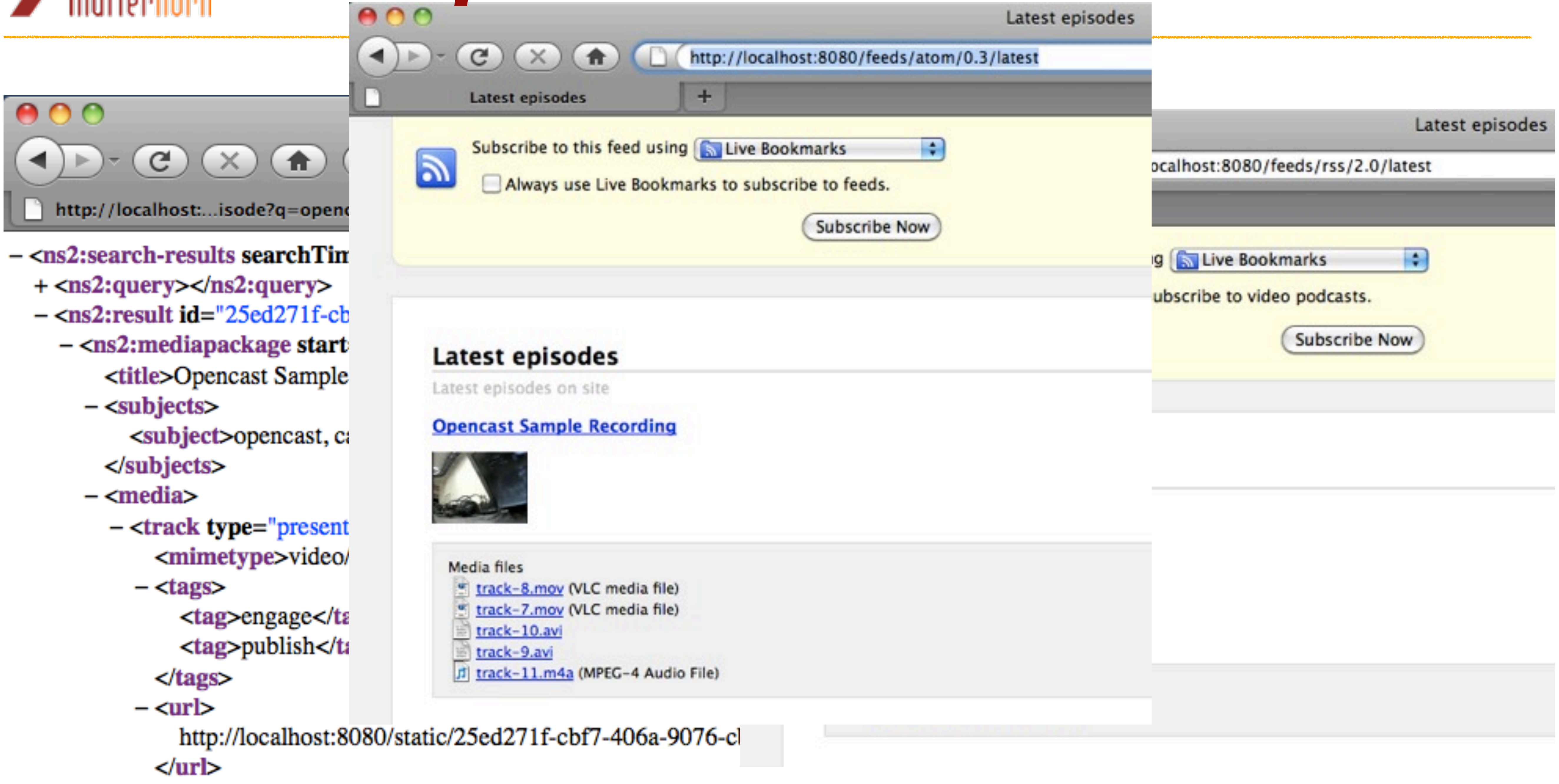
Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns2:track
xmlns:ns2="http://mediapackage.opencastproject.org"><mimetype>video
/quicktime</mimetype><tags/><url>https://opencast.jira.com/svn/MH/trunk/modules
/matterhorn-media/src/test/resources/av.mov</url><checksum
type="md5">9d3523e464f18ad51f59564acde4b95a</checksum><duration>14546</duration>
<audio id="audio-1"><device/><encoder type="AAC"/><bitdepth>16</bitdepth>
<channels>2</channels><samplingrate>44100</samplingrate>
<bitrate>128004.0</bitrate></audio><video id="video-1"><device/><encoder
type="AVC"/><bitrate>387969.0</bitrate><framerate>1.994</framerate>
<resolution>640x480</resolution><scantype type="Progressive"/></video>
</ns2:track>
```

HIDE

COMMUNITY | PROJECT
www.opencastproject.org

Examples: Search



```
http://localhost:...isode?q=openc
– <ns2:search-results searchTim
  + <ns2:query></ns2:query>
  – <ns2:result id="25ed271f-cb
    – <ns2:mediapackage start
        <title>Opencast Sample
      – <subjects>
          <subject>opencast, ca
        </subjects>
      – <media>
        – <track type="present
            <mimetype>video/
          – <tags>
              <tag>engage</ta
              <tag>publish</ta
            </tags>
          – <url>
              http://localhost:8080/static/25ed271f-cbf7-406a-9076-c
            </url>
```

Latest episodes

http://localhost:8080/feeds/atom/0.3/latest

Subscribe to this feed using Live Bookmarks

Always use Live Bookmarks to subscribe to feeds.

Subscribe Now

Latest episodes

Latest episodes on site

Opencast Sample Recording

Media files
- track-8.mov (VLC media file)
- track-7.mov (VLC media file)
- track-10.avi
- track-9.avi
- track-11.m4a (MPEG-4 Audio File)

ocalhost:8080/feeds/rss/2.0/latest

g Live Bookmarks

ubscribe to video podcasts.

Subscribe Now

Examples: Workflow

```
    <definition>
  <id>default</id>
  <operations>
    <operation id="inspect"></operation>
    <operation id="compose">
      <configurations>
        <configuration key="source-audio-flavor">presenter/source</configuration>
        <configuration key="source-video-flavor">presenter/source</configuration>
        <configuration key="target-tags">engage</configuration>
        <configuration key="encoding-profile">flash.http</configuration>
      </configurations>
    </operation>
    ...
  </operations>
</definition>
```

COMMUNITY | PROJECT
www.opencastproject.org

Version 1.0 Deliverables

- AuthN (openID?)
- AuthZ (LDAP role-based?)
- Bookmarking, annotations, and usage stats
- Integration with campus information systems
- Scheduling
- Classroom capture capabilities
- Instructor preferences

COMMUNITY | PROJECT
www.opencastproject.org

Version 2.0 Deliverables ?

- Breaking out of the "Media Package"
  - Sharing time-based, user-generated metadata through social graphs
- Enhanced media composition
  - e.g. picture-in-picture, scene-based stream selection
- Usage statistics across distribution channels

COMMUNITY | PROJECT
www.opencastproject.org

Contacts and Questions

- Project Website
  - http://www.opencastproject.org/
- Mailing List
  - http://lists.opencastproject.org/
- IRC
  - #opencast on irc.freenode.net
- Josh Holtzman: jholtzman@berkeley.edu