



JCR in Action

Content-based Applications with Apache Jackrabbit

Carsten Ziegeler

chiegeler@apache.org

Day Software

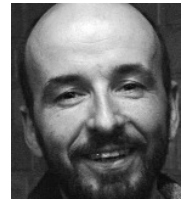
Bertrand Delacretaz

bdelacretaz@apache.org

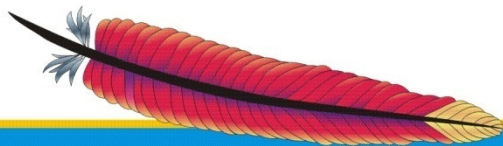


Day Software

About Bertrand Delacretaz



- Apache Software Foundation Member
 - Formerly active on Cocoon, FOP, SOLR
 - Currently active on Sling, Tika, and mentoring Pig and Tika
- Senior Developer at Day Software
- Living in Lausanne, Switzerland



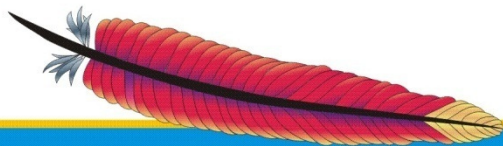
About Carsten Ziegeler



- Apache Software Foundation Member
 - Cocoon, Excalibur, Pluto, Felix, Incubator, Sling, Sanselan
 - PMC: Cocoon, Incubator, Portals, Felix, Excalibur (Chair)
- Senior Developer at Day Software
- Article/Book Author, Technical Reviewer
- JSR 286 spec group (Portlet API 2.0)

Agenda

- **Apache Jackrabbit**
- Basic content modeling
- References and search
- Advanced features
- Summary and questions



Apache Jackrabbit

JSR 170 reference implementation

Apache TLP since 2006

Looking back

- 1.0: April 2006
- 1.1: October 2006
- 1.2: January 2007
- 1.3: April 2007

Looking forward (tentative)

- 1.4: 2007
- 2.0: 2008 (JCR 2.0 RI)
- announce-subscribe@jackrabbit.apache.org

Components

- Core, API, RMI, WebDAV, webapp, JCA, ...

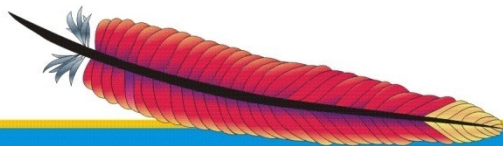


<http://jackrabbit.apache.org/>



Words of advice

- Read (or browse) the JCR specification
 - jcr-1.0.jar included
- Getting started with Jackrabbit
 - jackrabbit-webapp: Drop-in deployment
 - First Hops: Embedded repository
 - Take your time



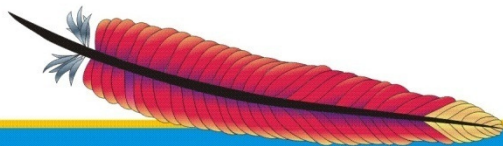
Words of advice

- Resources
 - Mailing lists: Excellent support, but may delay you
 - Website: Some good parts, but not complete or very well structured
 - Wiki: End user experience
 - FAQ: outdated
 - Issue tracker: Good response time



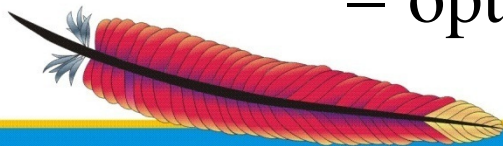
Agenda

- Apache Jackrabbit
- **Basic content modeling**
- References and search
- Advanced features
- Summary and questions



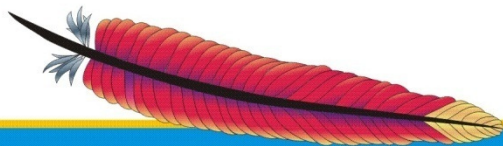
Introducing JCR Tunes

- JCR example application
- “How to implement a music store or library with JCR?”
- Designed to showcase JCR features and best practices
- Store and manage individual “tunes”
 - optionally organized in albums, etc.



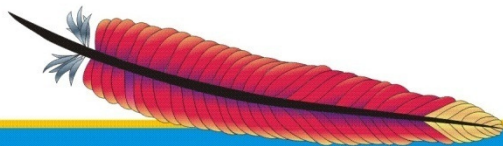
Introducing JCR Tunes II

- Support alternative views like
 - predefined genres, or more ad-hoc searches
- Integrated handling of reviews, cover images, and other related content
- Staged publishing and timed releases of tunes or albums



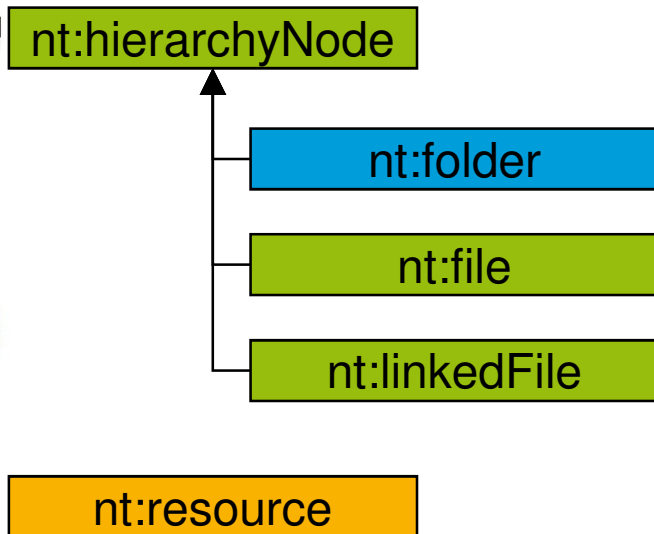
Introducing JCR Tunes III

- Personalization for things like settings, favorites, personal play-lists, etc.
- Extensibility and flexibility 😊

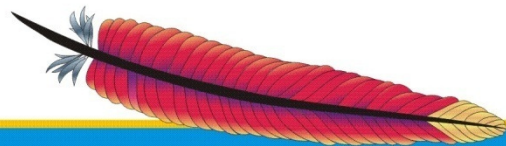
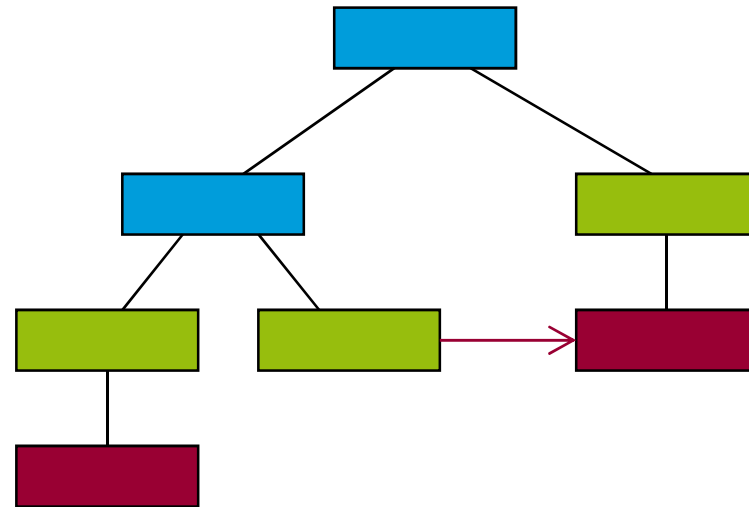


Starting point: Leverage the standard node types

Type hierarchy



- Content hierarchy



Bottom-up modeling: Content types

my:album > nt:folder
- artist (string)
- release date (date)

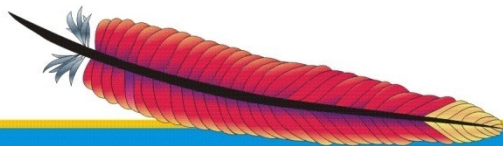
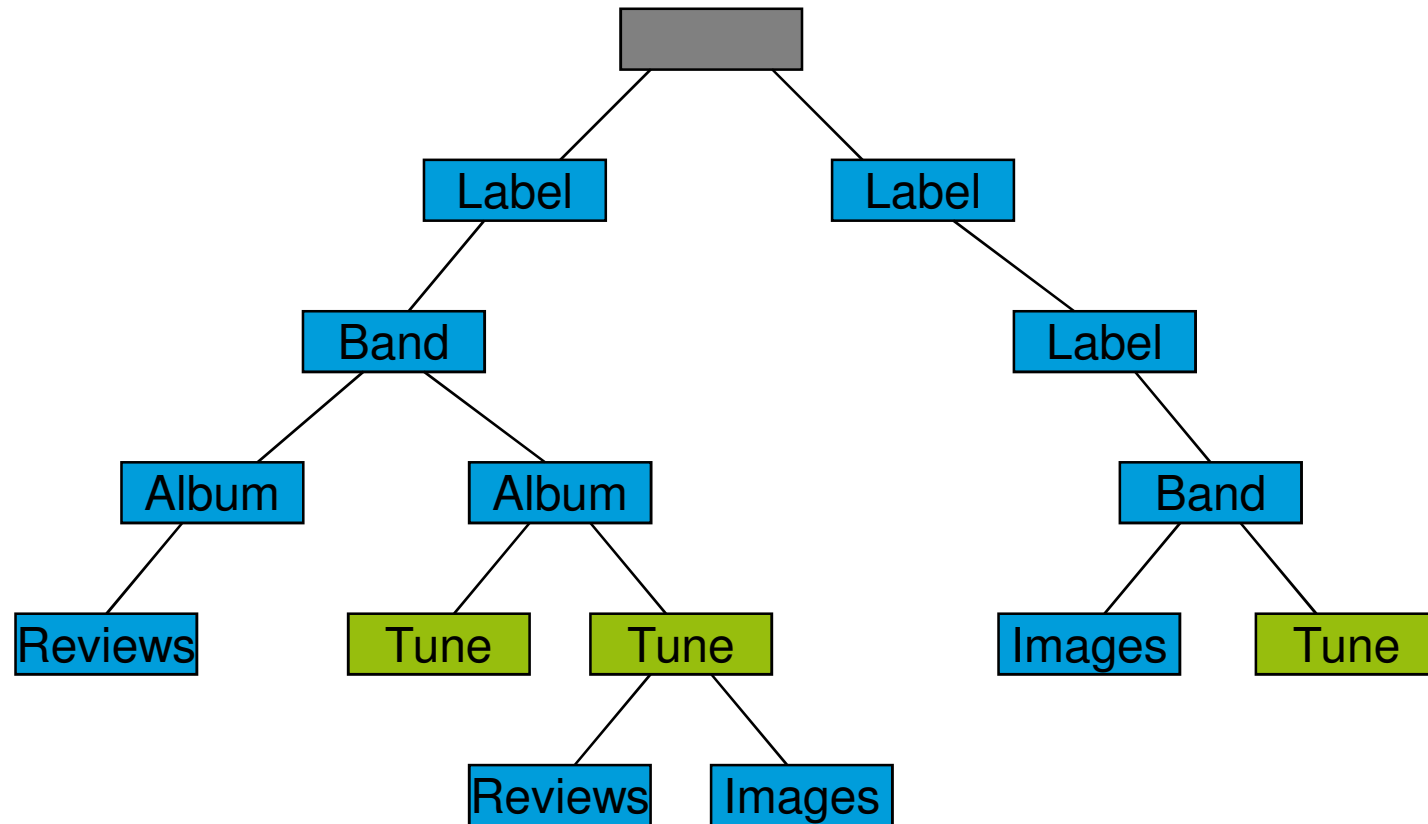
my:review > nt:file
- author (string)
- star rating (long)

my:tune > nt:file
- artist (string)
- release date (date)

my:resource > nt:resource
- codec (string)
- bitrate (long)

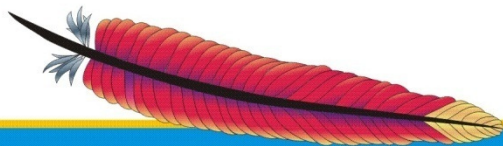


Top-down modeling: Content hierarchies



Content Modeling: Words of advice

- Namespaces
 - Use a single namespace per company or application
 - Use a reasonably unique namespace prefix
 - Prefixed names for structured content
 - Default namespace for unstructured content

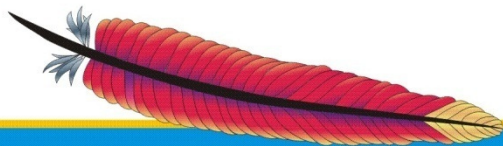


Content Modeling: Words of advice

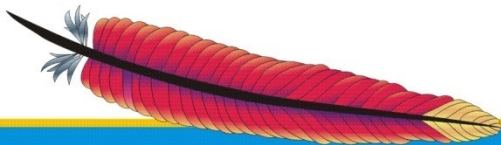
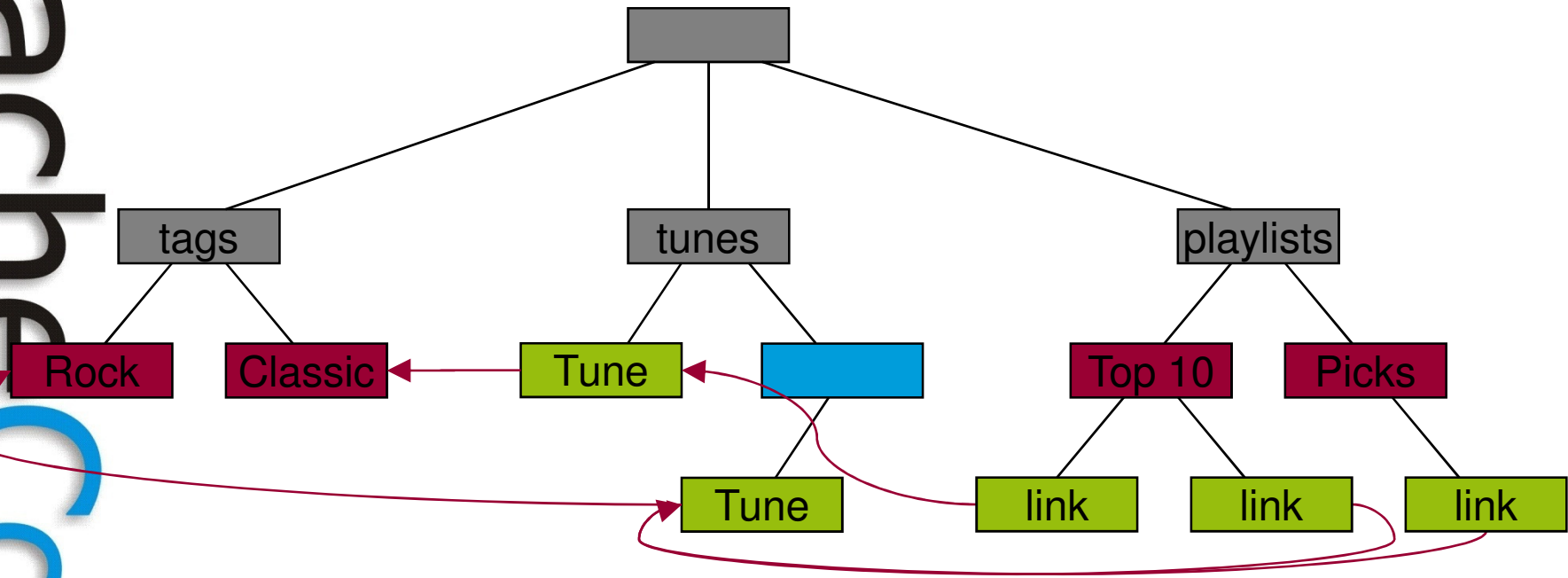
- Use an application root node
 - /my:content
 - Good for searching, backup, and migration
- Avoid flat hierarchies
 - User interface complexity
 - Jackrabbit performance
- Content-driven design
 - Design your content before your application

Agenda

- Apache Jackrabbit
- Basic content modeling
- **References and search**
- Advanced features
- Summary and questions

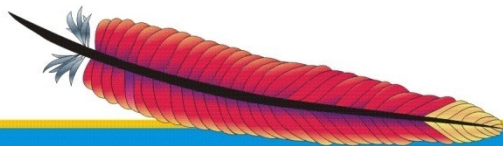


Alternative Views: References



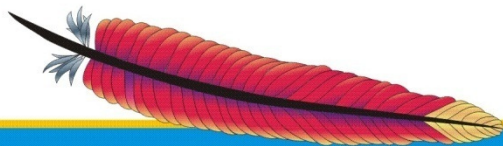
Alternative Views: Search

Looking for	XPath	SQL
Latest releases	<pre>/jcr:root/my:tunes//element(*,my:tune) [@released > xs:dateTime('...')]</pre>	<pre>SELECT * FROM my:tune WHERE jcr:path LIKE '/my:tunes/%' AND released > DATE '...'</pre>
Reviews with keywords	<pre>/jcr:root/my:tunes//element(*,my:review) /jcr:content[jcr:contains(.,'...')]</pre>	<pre>SELECT * FROM my:review WHERE jcr:path LIKE '/my:tunes/%' AND CONTAINS(*,'...')</pre>



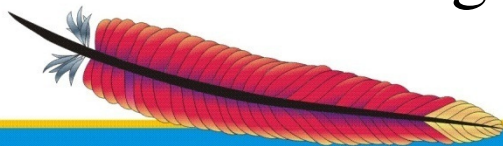
Alternative Views: Words of advice

- Moderate use of references
 - Circular references only within a subtree
 - Plan for backup and content migration
 - Jackrabbit performance: max 10k references to a single node
- Best search performance when selecting a small subset of content



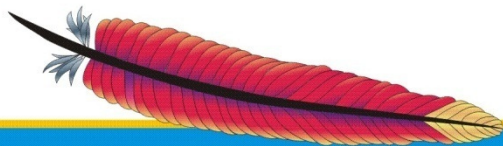
Alternative Views: Words of advice

- No joins or aggregate searches
- Full text indexing of binary properties only for jcr:data in nt:resource nodes
- Formatting date queries
 - ISO 8601 as the string format
 - `session.getValueFactory().createValue(Calendar.getInstance()).getString()`

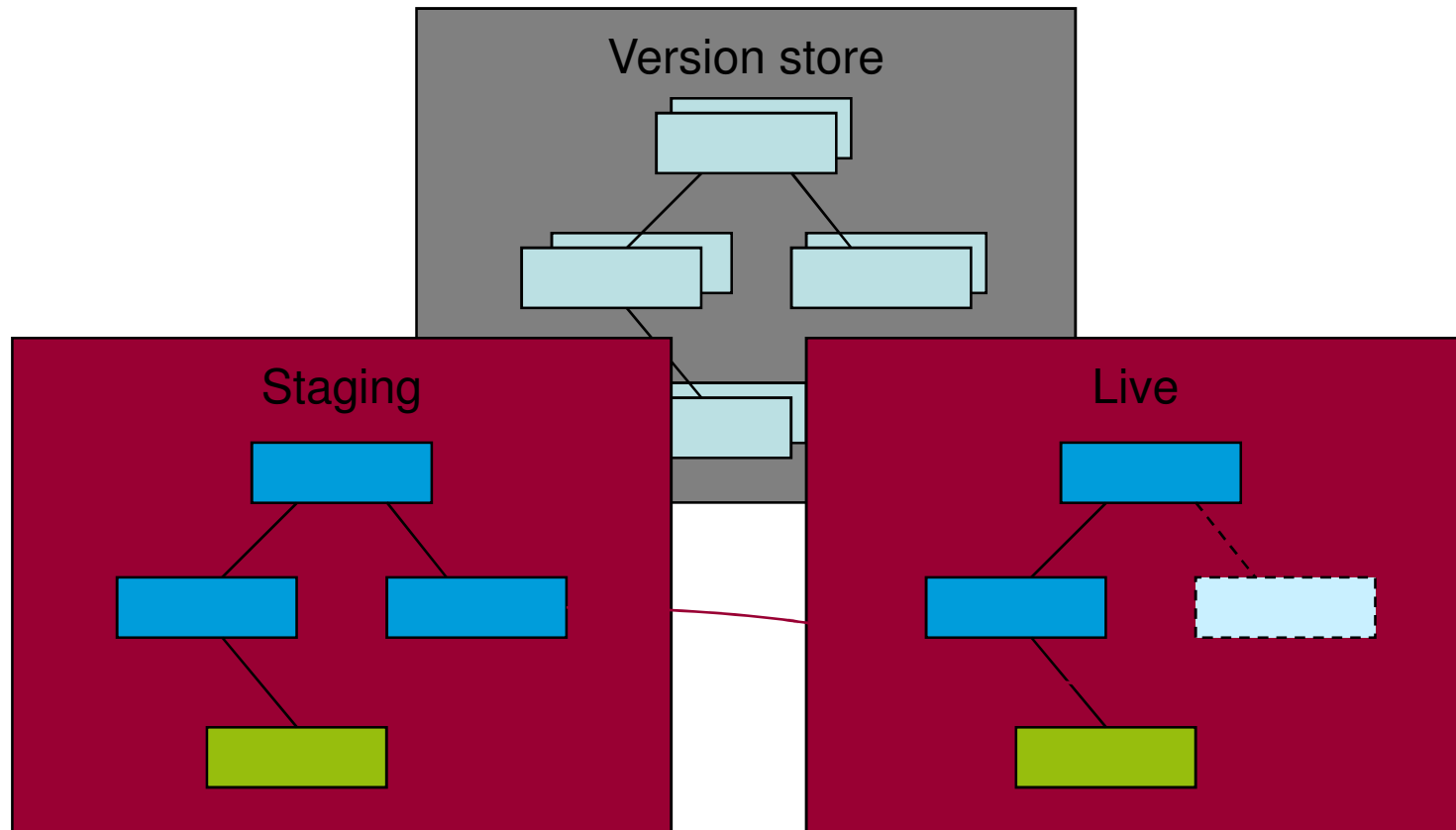


Agenda

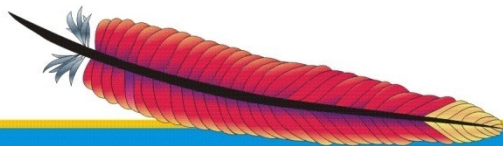
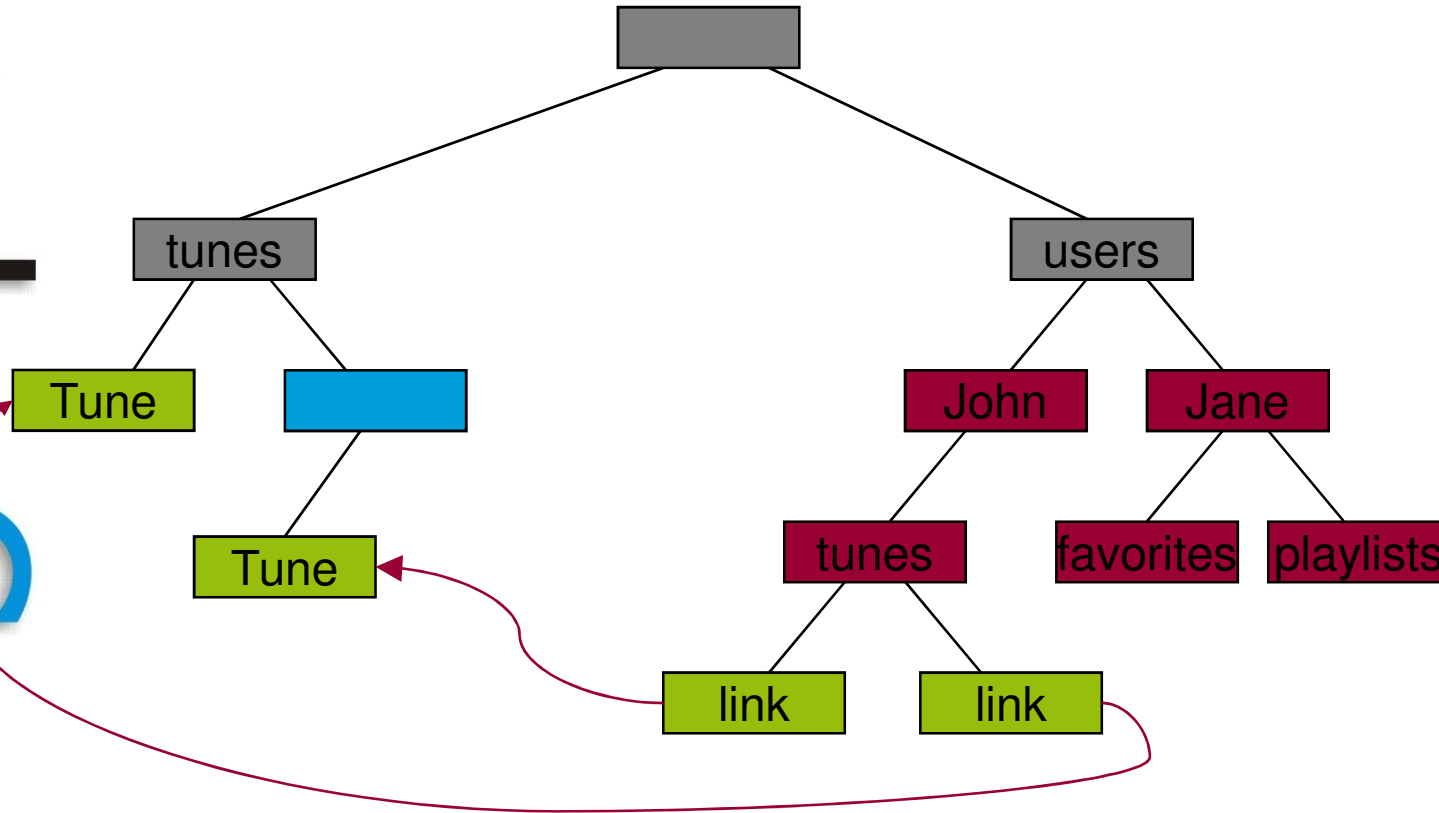
- Apache Jackrabbit
- Basic content modeling
- References and search
- **Advanced features**
- Summary and questions



Staged Publishing: Versioning

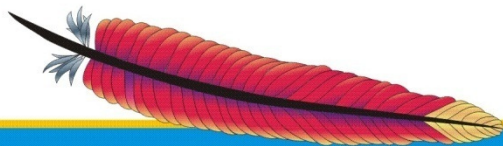


Personalization



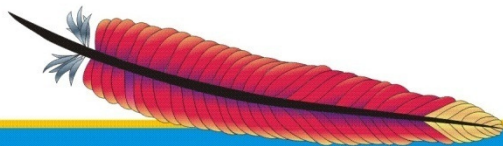
Authentication and Authorization

- Java Authentication and Authorization Services (JAAS)
 - Mostly the authentication part is currently used by Jackrabbit
 - Pluggable authentication components
 - Support for single sign-on



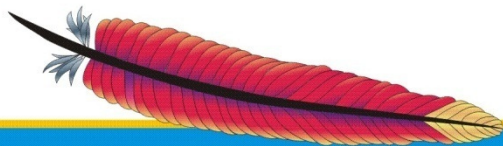
Authentication and Authorization

- Custom AccessManager interface in Jackrabbit
 - Pluggable authorization components
 - The default implementation supports only global read, write, and admin access
 - More advanced implementations are proprietary



Authentication and Authorization

- Repository-level authentication and authorization applies to all clients
 - Better than application-level authorization



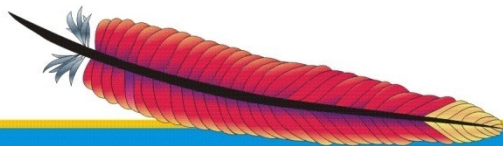
Agenda

- Apache Jackrabbit
- Basic content modeling
- References and search
- Advanced features
- **Summary and questions**



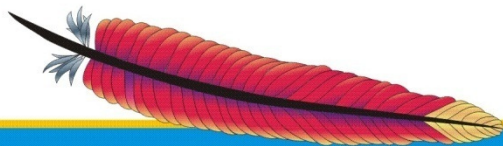
Conclusion

- (Nearly) Everything is content
 - Application content
 - HTML pages, CSS and JavaScript files, static images
 - Documentation, resource bundles, etc.
 - With versioning, export/import, full text search, etc.

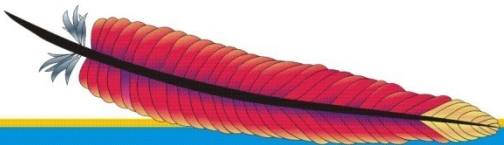


Conclusion

- Web-friendly
 - Trivial URI mapping
 - WebDAV access for free
 - Dispatch on node type, not on URI path
- In your application?
- Look at Apache Sling



ApacheCon



Q&A