JCR in Action

Content-based Applications with Apache Jackrabbit

Carstsen Ziegeler
cziegeler@apache.org

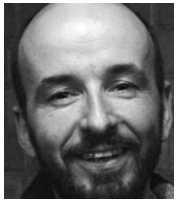
Day Software

Bertrand Delacreataz
bdelacretaz@apache.org

Day Software

Leading the Wave
of Open Source

About Bertrand Delacretaz



- Apache Software Foundation Member
  - Cocoon, FOP, Solr, Sling, Tika, Incubator
  - PMC: Cocoon, Tika, Sling
- 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

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

Session Planner

- XSLT and Xpath – Without the Pain
  Bertrand Delacretaz – Today 15:00
- Apache Sling (Fast Feather Track)
  Felix Meschberger – Thursday 10:00
- A Little REST and Relaxation
  Roy Fielding – Thursday 14:00
- Apache Sanselan (Fast Feather Track)
  Carsten Ziegeler – Thursday 15:00
- Getting Up to Speed with Apache iBatis
  Carsten Ziegeler – Thursday 16:30
- The Apache Portals Panel
  Carsten Ziegeler – Friday 10:00

Content Repository

- Generic application data store
- Structured and unstructured content
- Support small and large-scale data
- Locking, transactions, versioning, observation and searching

JSR 170: Content Repository for Java^{TM} technology API

- (Java) Standard
  - Supported by many vendors
  - Several open source solutions
- How do you connect to a CR?
- How do you interact with a CR?
- How do you query a CR?

Content Repository Features

- Hierarchical content
- Structured
  - Nodes and properties (with types)
- And/or unstructured
- Read only or read/write
- Fine and coarse-grained

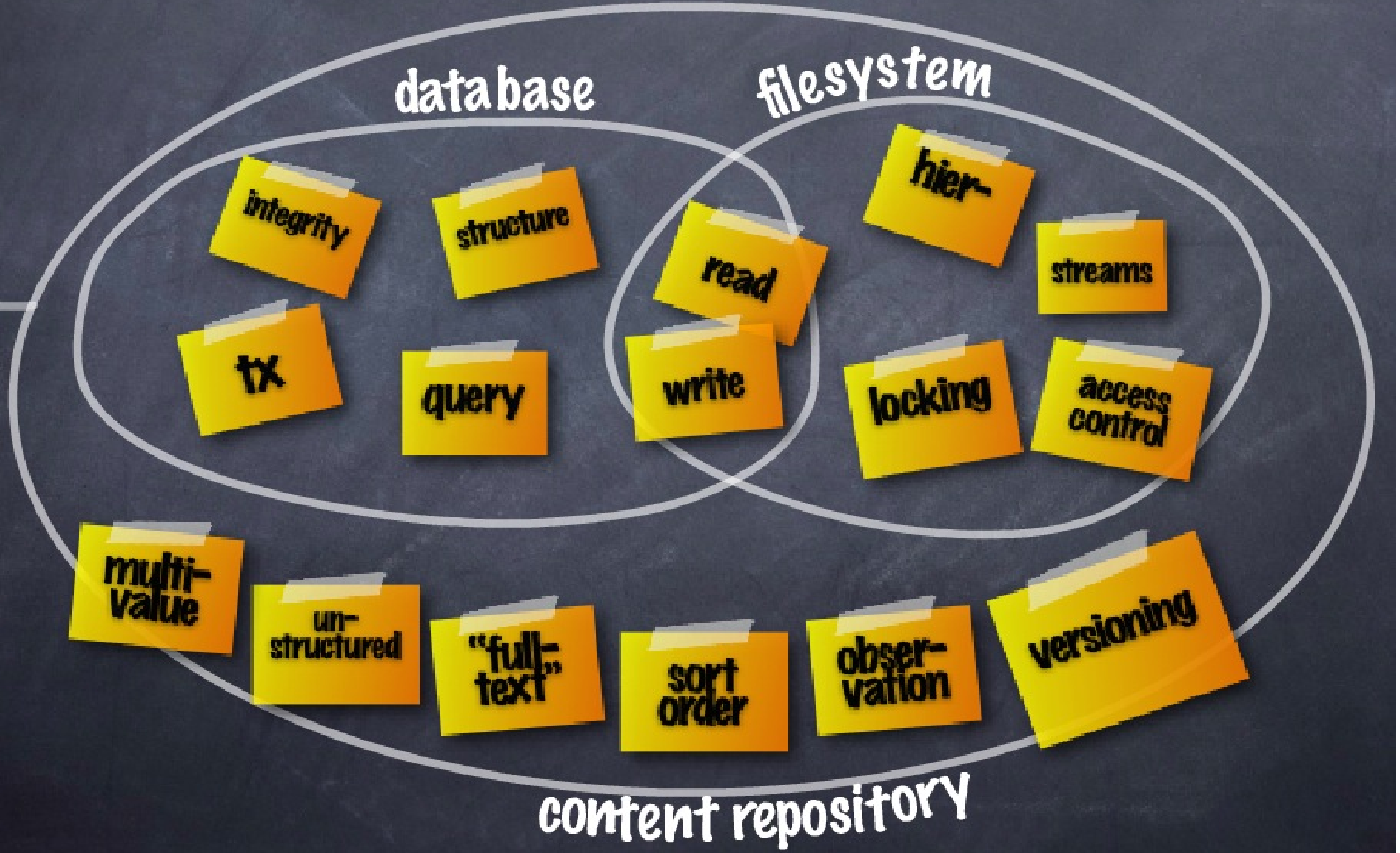
Content Repository Features

- Query (XPath)
- Export/Import (XML)
- Referential Integrity
- Access Control
- Versioning
- Observation
- Locking and Transactions (JTA)

Comparison

- File system
  - Hierarchival, unstructured, read/write
- Database
  - Structured, read/write
  - Referential integrity, transactions
- Content Repository
  - Advantages of a FS and a database
  - Plus observation, versioning etc.

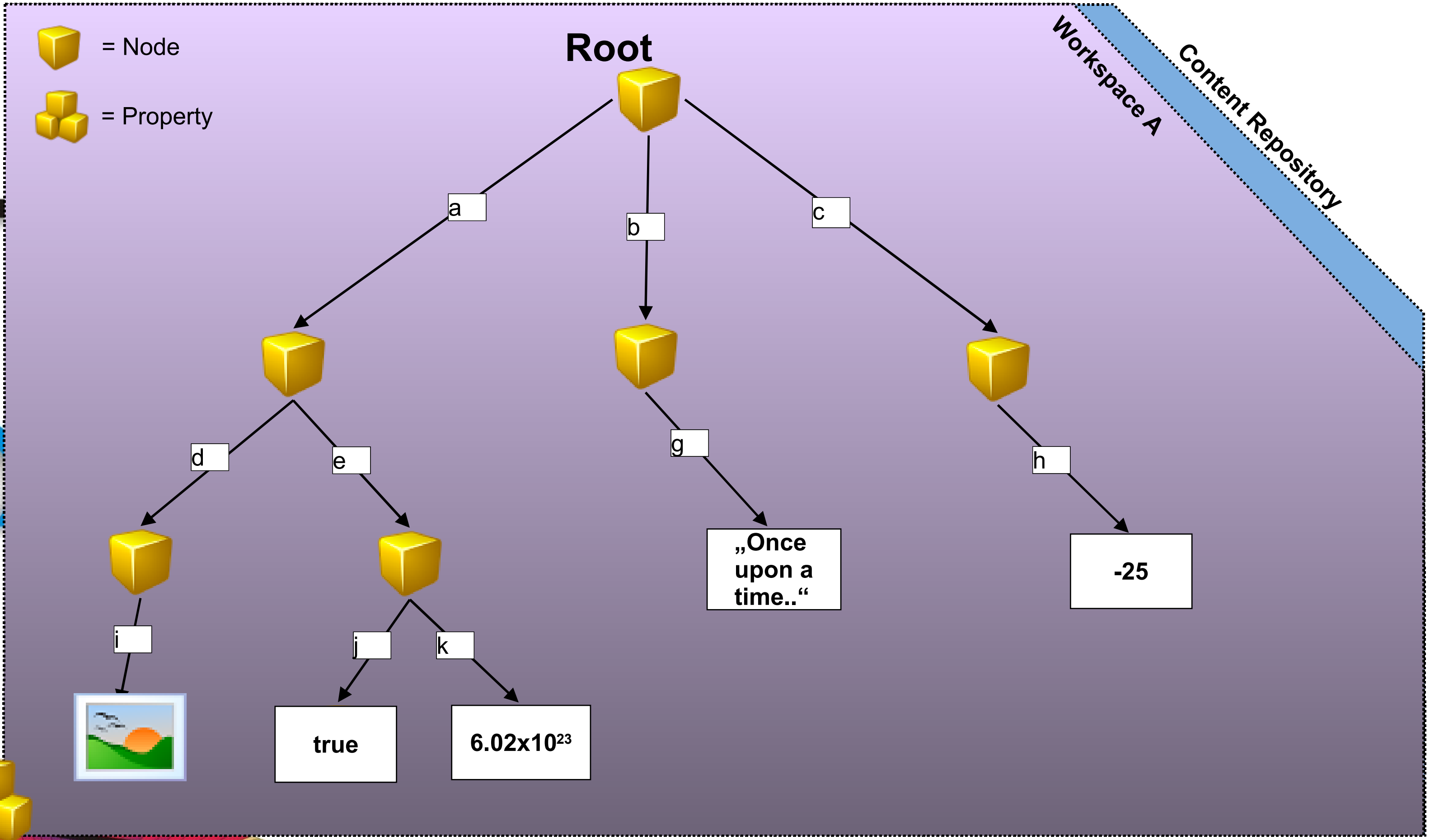
Best of both worlds.



The Repository Model

- Repository = one (or more) workspaces
- Workspace = a tree of items
- Item = Node or property
- Nodes provide the content structure
  - may have children
- Actual data is stored as values of properties
- Types and namespaces!

Nodes & Properties



Connecting to the Repository

Acquire the Repository object: exact mechanism is outside spec, but one possible way: JNDI

```
// Get the Repository object
InitialContext ctx = ...

Repository repository =
      (Repository)ctx.lookup("MyRepository");
```

Working with the Repository

Get your Credentials.

Then provide your Credentials and a Workspace name to get a Session.

```
// Get a Credentials object
Credentials credentials =
  new SimpleCredentials("MyName",
                        "MyPassword".toCharArray());
// Get a Session
Session mySession =
  repository.login(credentials, "Workspace A");
```

Traverse the Hierarchy

We have a Session object *mySession*, bound to the Workspace called “Workspace A”

We begin by getting the root node of the Workspace:

```
// Get root node
Node root = mySession.getRootNode()
```

and continue down the hierarchy:

```
// Go to the node you want
Node myNode = root.getNode("a/e");
```

Retrieve a Property

We have Node object *myNode*, the node located at /a/e

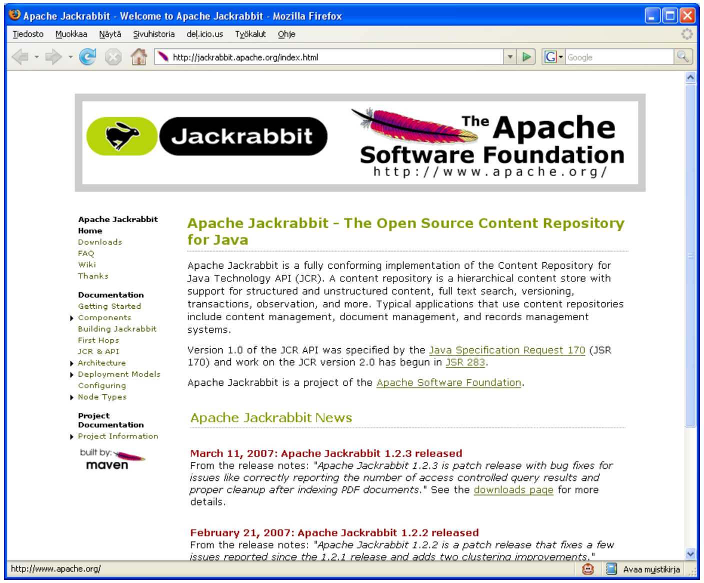
We get one of its properties, j :

```
// Retrieve a property of myNode
Property myProperty = myNode.getProperty("j");

// Get the value of the property
Value myValue = myProperty.getValue();

// Convert the value to the desired type
double myDouble = myValue.getDouble();
```

myDouble will contain the value 6.02 x 10^23

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, …


Apache Jackrabbit - Welcome to Apache Jackrabbit - Mozilla Firefox
Tiedosto
Muokkaa
Näytä
Sivuhistoria
del.icio.us
Työkalut
Ohje
http://jackrabbit.apache.org/index.html
Google
Jackrabbit
The Apache Software Foundation
http://www.apache.org/
http://www.apache.org/
Avaa muistikirja

Apache Jackrabbit
Home
Downloads
FAQ
Wiki
Thanks

Documentation
Getting Started
Components
Building Jackrabbit
First Hops
JCR & API
Architecture
Deployment Models
Configuring
Node Types

Project Documentation
Project Information

built by: maven

Apache Jackrabbit - The Open Source Content Repository for Java

Apache Jackrabbit is a fully conforming implementation of the Content Repository for Java Technology API (JCR). A content repository is a hierarchical content store with support for structured and unstructured content, full text search, versioning, transactions, observation, and more. Typical applications that use content repositories include content management, document management, and records management systems.

Version 1.0 of the JCR API was specified by the Java Specification Request 170 (JSR 170) and work on the JCR version 2.0 has begun in JSR 283.

Apache Jackrabbit is a project of the Apache Software Foundation.

Apache Jackrabbit News

March 11, 2007: Apache Jackrabbit 1.2.3 released
From the release notes: "*Apache Jackrabbit 1.2.3 is patch release with bug fixes for issues like correctly reporting the number of access controlled query results and proper cleanup after indexing PDF documents.*" See the downloads page for more details.

February 21, 2007: Apache Jackrabbit 1.2.2 released
From the release notes: "*Apache Jackrabbit 1.2.2 is a patch release that fixes a few issues reported since the 1.2.1 release and adds two clustering improvements.*"

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

- JCR and Apache Jackrabbit
- **Basic content modeling**
- References and search
- Advanced features
- Sample application
- 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

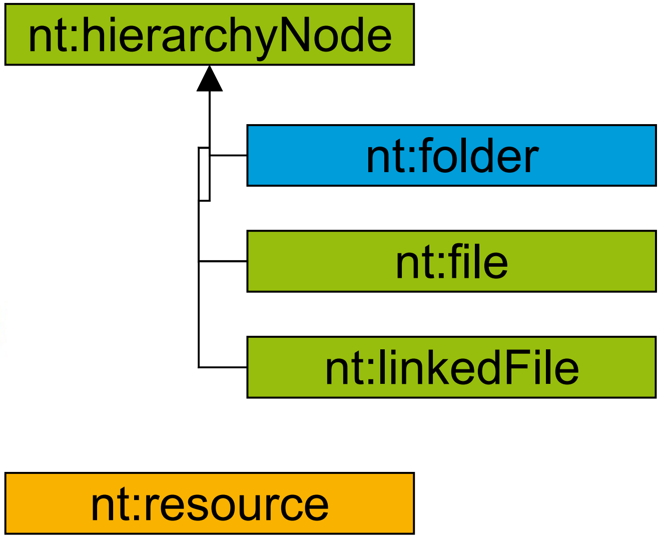
- Support alternative views like
  - predefined genres, or more ad-hoc searches
- Integrated handling of reviews, cover images, and other related content
- Staged publishing and timed releases of tunes or albums

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



nt:hierarchyNode

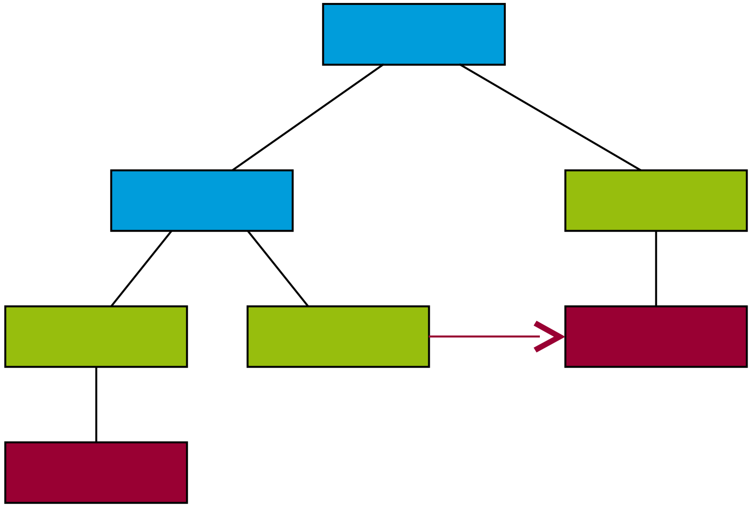
nt:folder

nt:file

nt:linkedFile

nt:resource

- Content hierarchy



Bottom-up modeling: Content types

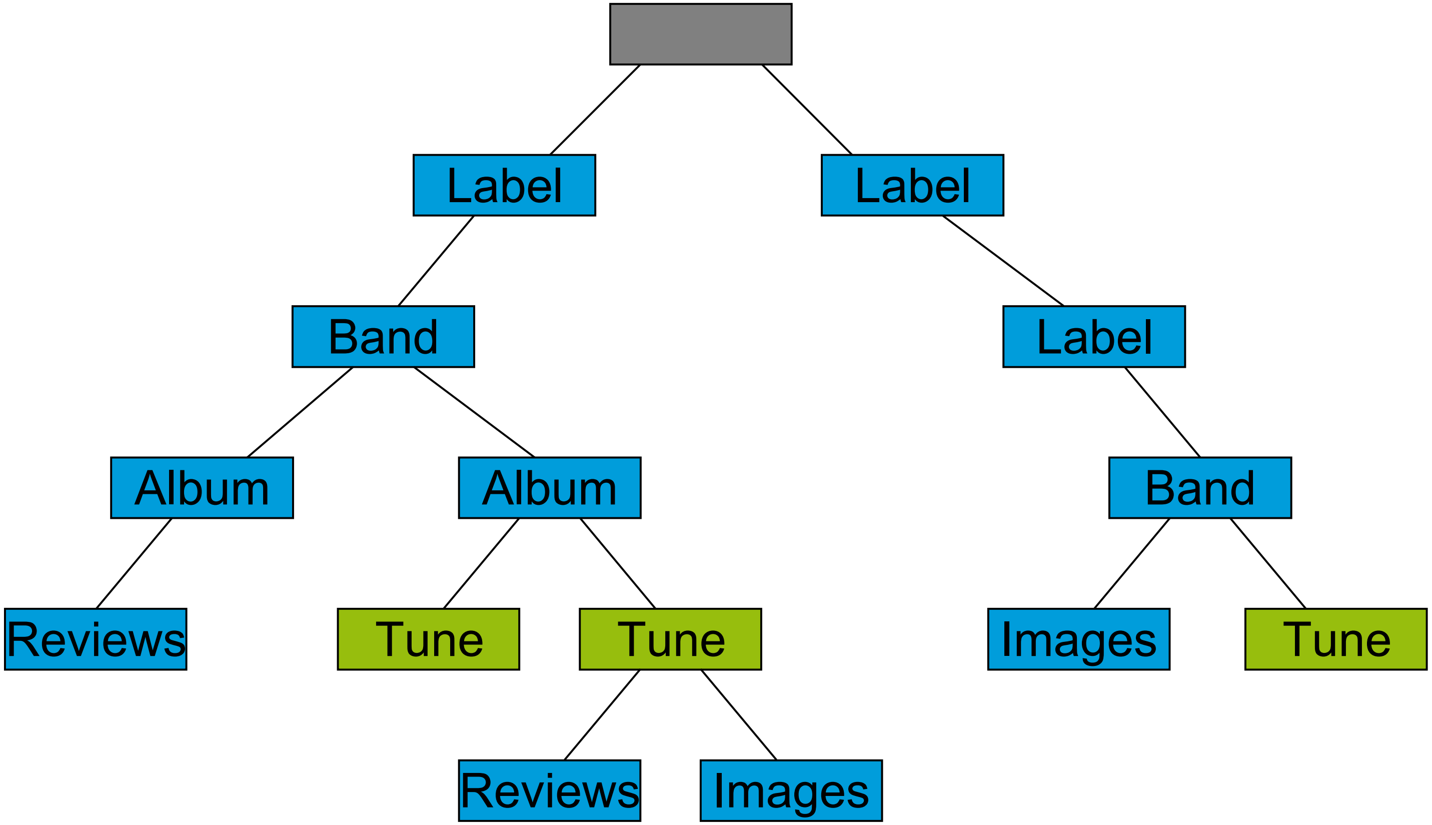
my:album > nt:folder
- artist (string)
- release date (date)

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

Content Modeling: Words of advice

- Checkout Jackrabbit wiki and mailing lists
  - "Davids Model"
- Look at existing node types
- Mixin node types possible

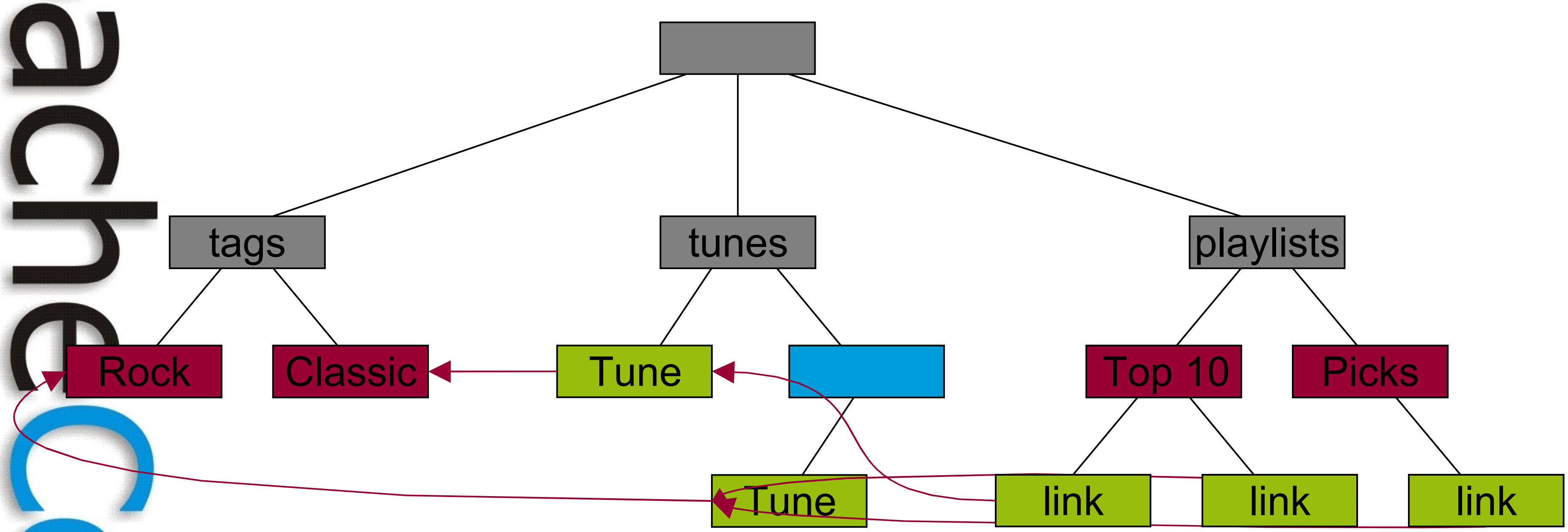
David's Model

- Rule #1: Data First, Structure Later. Maybe.
- Rule #2: Drive the content hierarchy, don't let it happen.
- Rule #6: Files are Files are Files.
- Look at http://wiki.apache.org/jackrabbit/DavidsModel

Agenda

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

Alternative Views: References



API:
Node.getReferences():PropertyIterator
Property.getNode():Node
Node.setProperty(String name, Node)

Alternative Views: Search

Looking for	XPath	SQL

Latest releases	/jcr:root/my:tunes//element(*,my:tune)
	[@released > xs:dateTime(‘…’)]
		SELECT * FROM my:tune
		WHERE
		jcr:path LIKE ‘/my:tunes/%’
		AND released > DATE ‘…’

Reviews with keywords	/jcr:root/my:tunes//element(*,my:review)
	/jcr:content[jcr:contains(.,’…’)]
		SELECT * FROM my:review
		WHERE
		jcr:path LIKE ‘/my:tunes/%’
		AND CONTAINS(*,‘…’)

API:
Session.getWorkspace().getQueryManager():QueryManager
QueryManager.createQuery(String stmt, String language):Query;
Query.execute():QueryResult

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
- References, path or name property

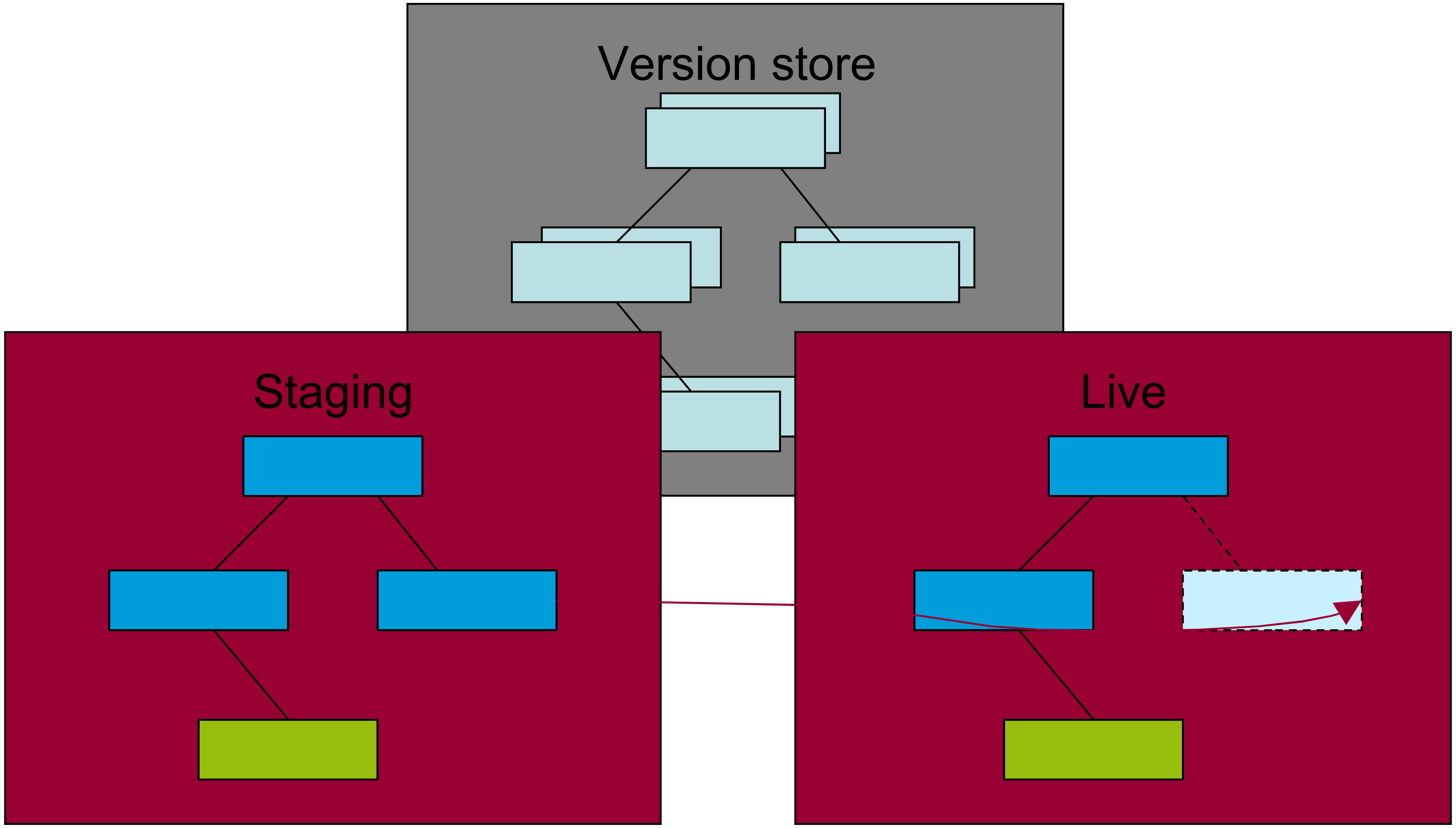
Alternative Views: Words of advice

- 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()

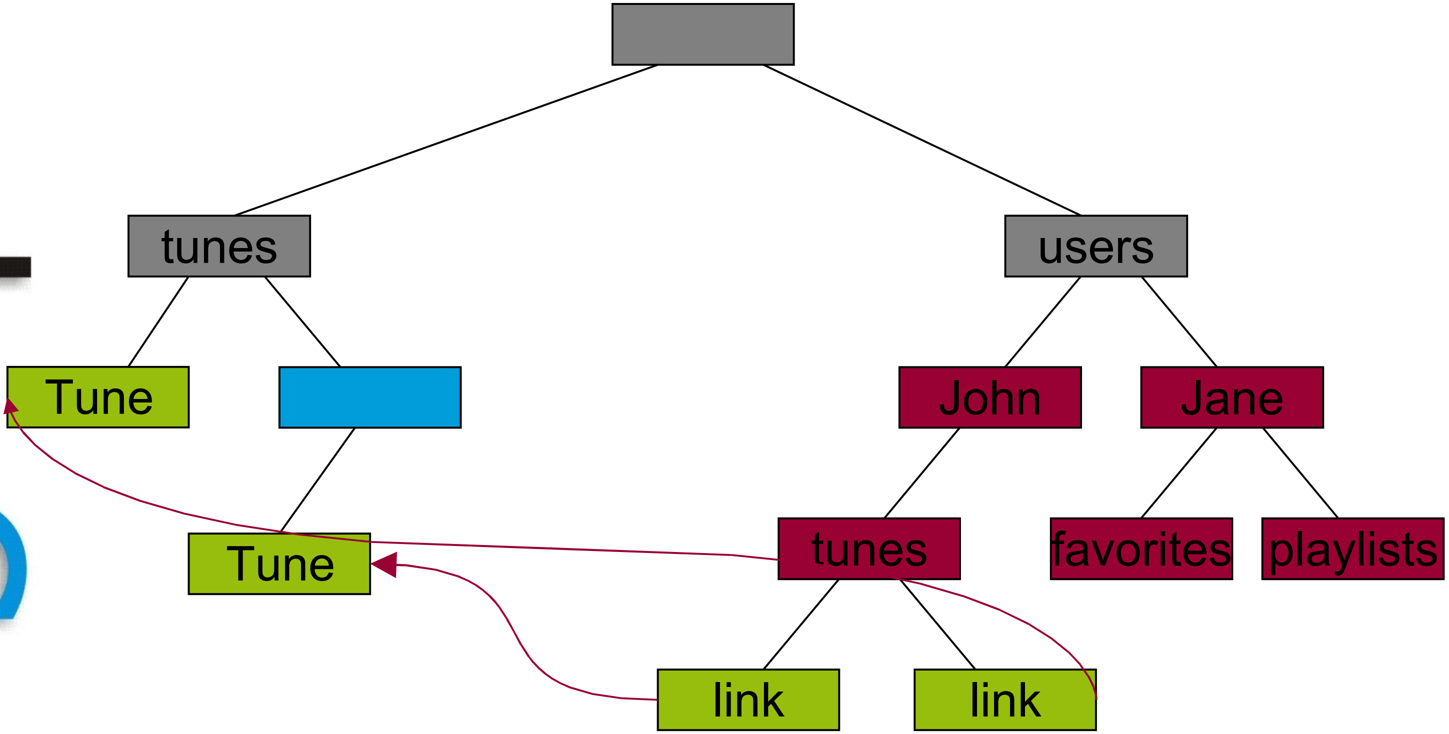
Agenda

- JCR and Apache Jackrabbit
- Basic content modeling
- References and search
- **Advanced features**
- Sample application
- 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 sing-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

Jackrabbit Configuration

- Workspace configuration
  - XML configuration
  - Persistence Managers
  - Query index configuration (Lucene)
  - File system configuration
- Check out the documentation

Observation

- Optional feature of the JCR specification
- Enables applications to register interest in events
- Monitoring events
- Responding to events

API:

```
ObservationManager:
addEventListener(EventListener listener, int eventTypes,
                 java.lang.String absPath, boolean isDeep,
                 java.lang.String[] uuid,
                 java.lang.String[] nodeTypeName, boolean noLocal)
```

Event Types

- Events can be of 5 different types
  - NODE_ADDED
  - NODE_REMOVED
  - PROPERTY_ADDED
  - PROPERTY_REMOVED
  - PROPERTY_CHANGED

Observation Events

- Describe changes to a workspace
- Dispatched on persistent change
- Provide the path of the item
- Provide the user ID
- Only provided to sessions with sufficient access privileges

Event Listeners

- Registered with a workspace
- Registration with optional filters
  - Like node types, paths
- Receive events for every change

API:

```
public void onEvent(EventIterator events);
```

Agenda

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

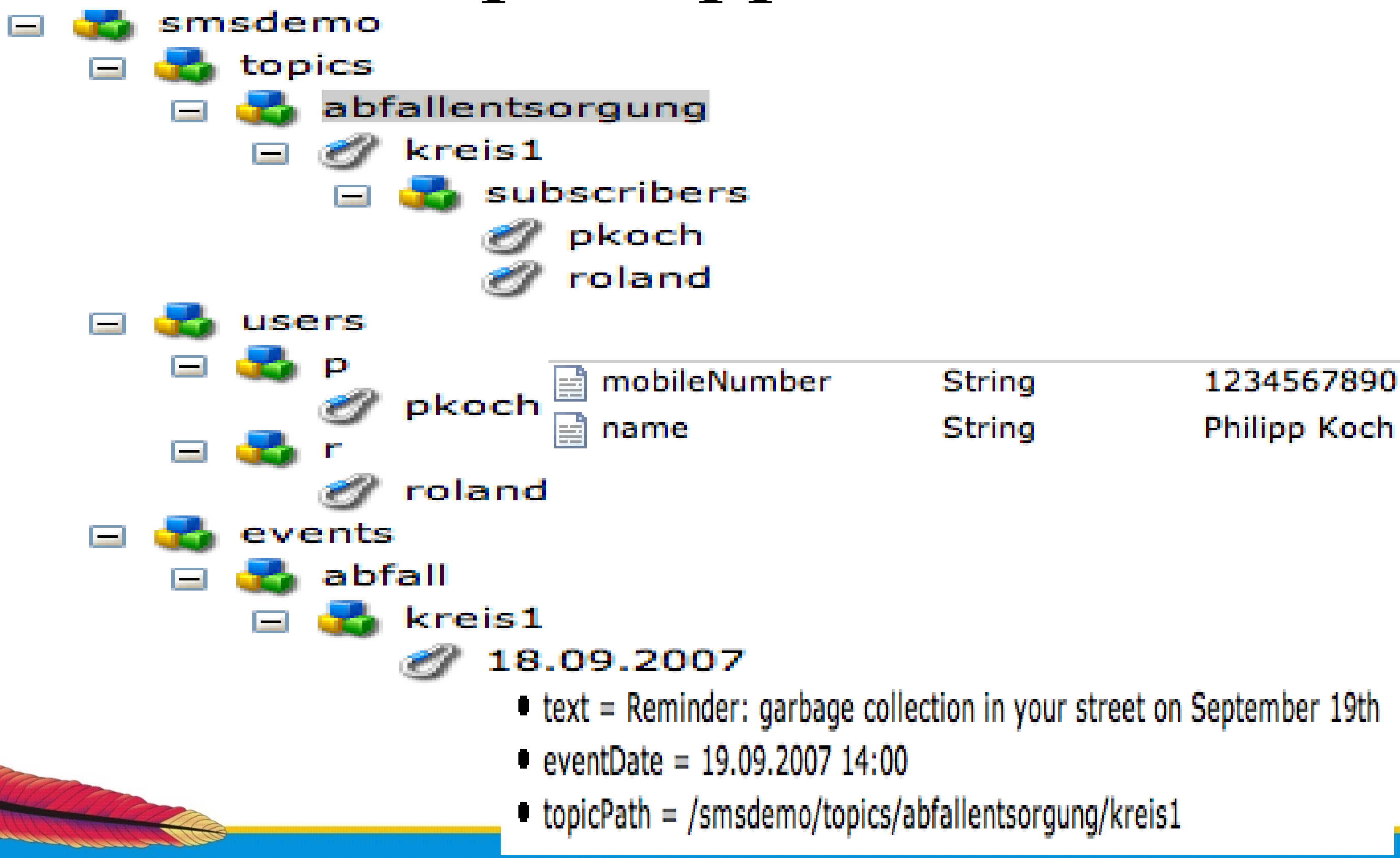
Getting Content

- Get the repository
- Login to a workspace
  - Returns a session
- Starting with the session
  - Navigate to the desired content
  - Get the query manager and search

Writing Content

- Get the repository
- Login to a workspace
  - Returns a session
- Navigate to the correct node
  - To change it
  - To add new nodes/properties
  - To remove nodes/properties
- Persist changes

SMS Sample Application



```
⊟ smsdemo
    ⊟ topics
        ⊟ abfallentsorgung
            ⊟ kreis1
                ⊟ subscribers
                      pkoch
                      roland
    ⊟ users
        ⊟ p
              pkoch
        ⊟ r
              roland
    ⊟ events
        ⊟ abfall
            ⊟ kreis1
                  18.09.2007
```

mobileNumber	String	1234567890
name	String	Philipp Koch

- text = Reminder: garbage collection in your street on September 19th
- eventDate = 19.09.2007 14:00
- topicPath = /smsdemo/topics/abfallentsorgung/kreis1

SMS Sample: Observation

```
observationSession
  .getWorkspace()
    .getObservationManager()
      .addEventListener(
        this,
        javax.jcr.observation.Event.PROPERTY_CHANGED |
        javax.jcr.observation.Event.NODE_ADDED,
        SMS_EVENT_ROOT, true, null,
        new String[] { SMS_EVENT }, true
);
```

SMS Sample: Event Handling

```
public void onEvent(EventIterator events) {
...
case javax.jcr.observation.Event.NODE_ADDED:

  Node newEvent = (Node) observationSession.getItem(path);

  String date = newEvent.getProperty(EVENT_DATE).getString();
  DateFormat formater = new SimpleDateFormat("dd.MM.yyyy
hh:mm:ss");
  Date pdate = formater.parse(date);

  addScheduledEvent(pdate, path);

  break;
```

SMS Sample: Event Processing

```
private void sendSms(Node event) throws RepositoryException {

  String topicPath = event.getProperty(TOPIC_PATH).getString();
  String text = event.getProperty(TEXT).getString();
  String smsSubscribers[] = getSmsSubscribers(topicPath);

  send(text, smsSubscribers);
}
```

SMS Sample: Event Processing

```
private String[] getSmsSubscribers(String topicPath) {

  Node topic = (Node) session.getItem(topicPath);
  Node subscriberNode = topic.getNode("subscribers“);

  NodeIterator subscribers = subscriberNode.getNodes();

  while (subscribers.hasNext()) {
    Node subscriber = subscribers.nextNode();

    String path= subscriber.getProperty(SMS_USER_PATH).getString();
    Node user = (Node) session.getItem(path);
    String mobileNumber =
    user.getProperty(SmsUserContent.MOBILE_NUMBER).getString();

  }
}
```

Advanced Development

- Apache Jackrabbit OCM
  - Map content to Java objects and vice versa
  - Similar to database ORMs
- Apache Sling
  - New project in incubation
  - REST based web framework
  - Leverages OSGi

Agenda

- JCR and Apache Jackrabbit
- Basic content modeling
- References and search
- Advanced features
- Sample application
- **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 Jackrabbit OCM and Apache Sling

Session Planner

- XSLT and Xpath – Without the Pain
  Bertrand Delacretaz – Today 15:00
- Apache Sling (Fast Feather Track)
  Felix Meschberger – Thursday 10:00
- A Little REST and Relaxation
  Roy Fielding – Thursday 14:00
- Apache Sanselan (Fast Feather Track)
  Carsten Ziegeler – Thursday 15:00
- Getting Up to Speed with Apache iBatis
  Carsten Ziegeler – Thursday 16:30
- The Apache Portals Panel
  Carsten Ziegeler – Friday 10:00

Visit our booth for infos, discussions **and jobs!**

Q&A