JSR 170 (JCR)
Java Content Repository API
Introduction and examples

Bertrand Delacrétaz

bdelacretaz@apache.org (@work: www.day.com)
grep.codeconsult.ch

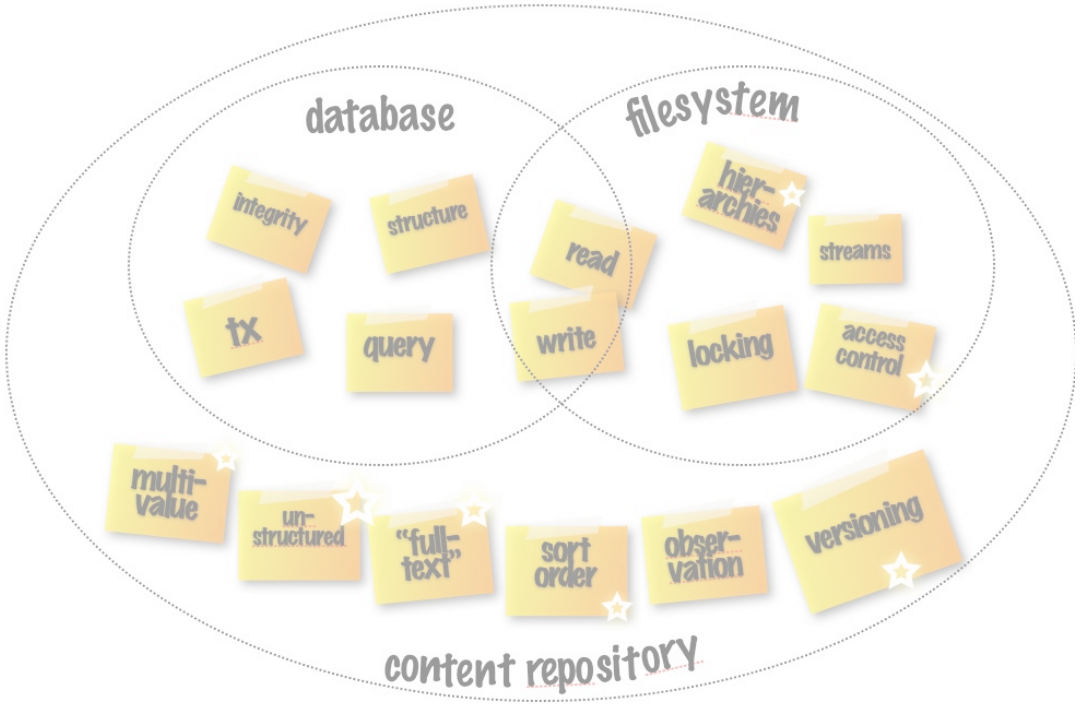
Geneva, November 21st, 2008

slides revision: 2008-11-21

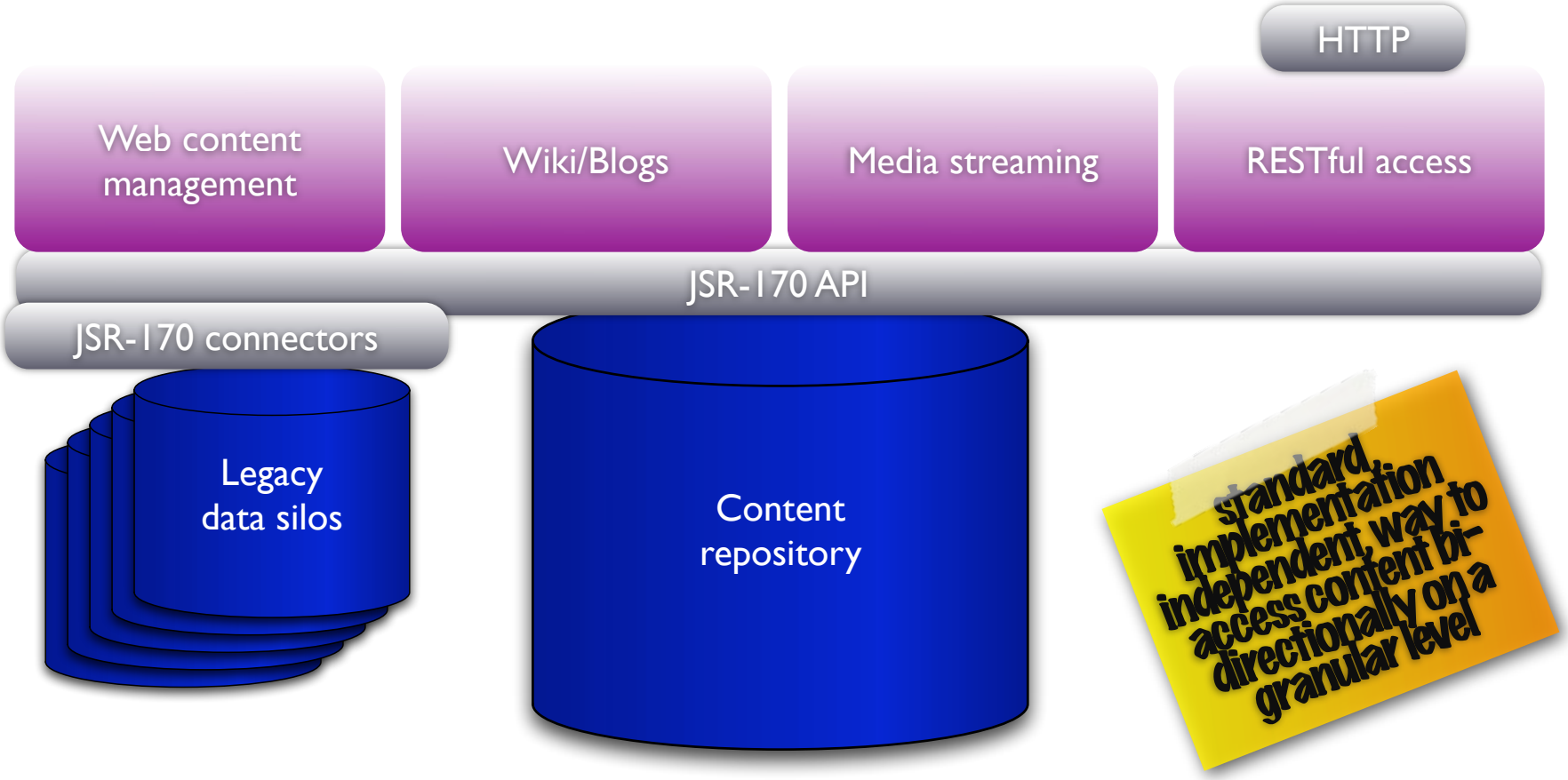
The plan

JCR intro
JCR API basics, Nodes, Properties, etc.
File annotation and search demo
Live demos using javashell
Content model examples

Intro


database
filesystem
integrity
structure
read
hier-archies
streams
tx
query
write
locking
access control
multi-value
un-structured
"full-text"
sort order
obser-vation
versioning
content repository

What?


HTTP
Web content management
Wiki/Blogs
Media streaming
RESTful access
JSR-170 API
JSR-170 connectors
Legacy data silos
Content repository
standard, implementation independent, way to access content bi-directionally on a granular level

JSR 170 - jcp.org spec

JSR-170

Content Repository for JavaTM technology API

Spec-Lead:
Day Software

Status:
Final Release 17-jun-2005

Expert Group:

JSR 283 (JCR 2.0)

JSR-170 | **JSR-283**

Content Repository for JavaTM technology API v2.0

Spec-Lead:
Day Software

Status:
Public Review Closed sep-2007

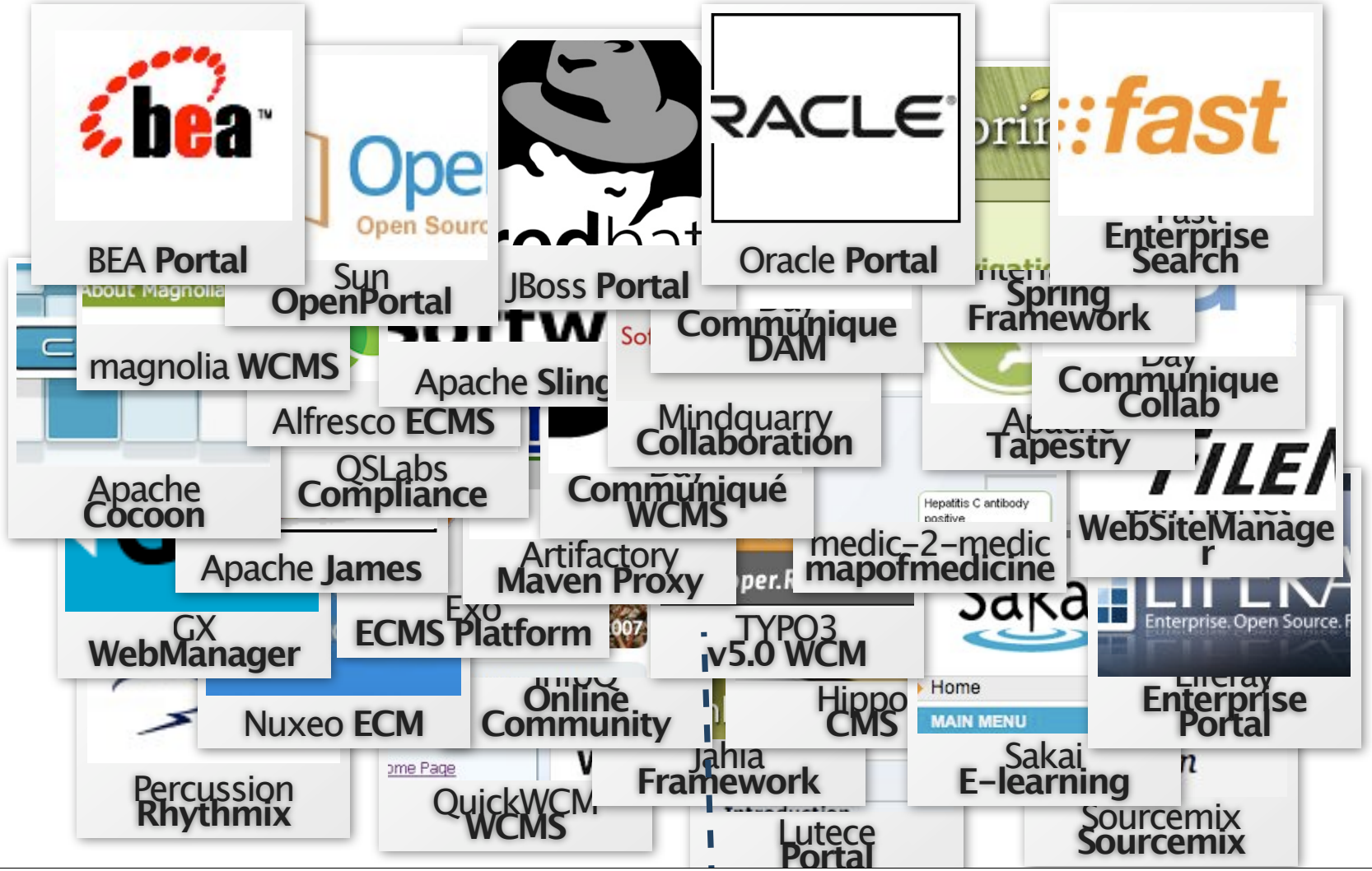
Java Community Process

Expert Group:

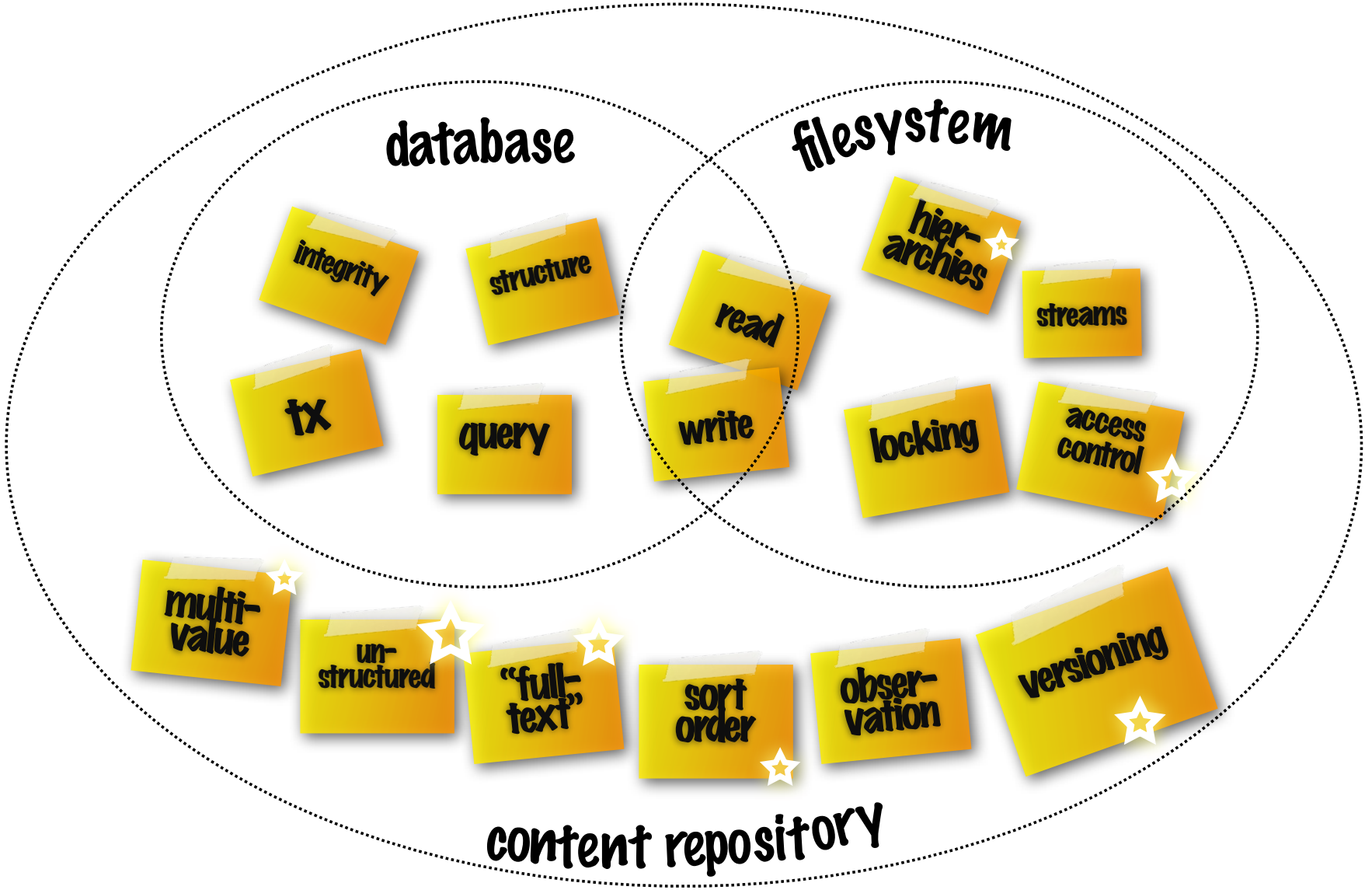
FileNet, Xythos, Oracle, OpenText, Nuxeo, IBM, bea, Apache, redhat, Interwoven, First, SAP, eXo, atg, MediaSurface, EMC documentum, Alfresco, GX, Sun, sas, Jackrabbit, Hummingbird, Mobius, Borland

Known compliant repositories

Some known applications



JSR 170 - best of both worlds



Clean and simple API

```
<%
childCount = node.getContentCount();
for(int i=0;i<childCount;i++) {
    IContent child = node.getCo
    ch
}
%>
```

```
<%
fndocs = new IFnObjSetDual
        fnfolder.getConten
  (idmFolde
int numDocs
for (int i
        IFn
        new
  (i)));
        IFn
        IFn
  ("Title")
        Str
    %><%=
    if (c
      cle
}
%>
```

```
<%
LAPI_DOCUMENTS documents = new LAPI_DOCUMENTS(session);
LLValue childTable = new LLValue();
documents.ListObjects(volumeID, folderID,
        null, null, LAPI_DOCUMENTS.PERM_SEE,
        childTable);
teValues();
ement();
me");
```

```
<%
// JSR 170:
NodeIterator children = node.getNodes();
while (children.hasNext()) {
    Node child = children.nextNode();
    Property title = child.getProperty
  ("Title");
    %><%= title %><br /><%
}
%>
```

```
<%
PageIterator children = page.getPages();
while (children.hasNext()) {
    Page child = children.nextPage();
    Container toplevel = child.getContent();
    Atom title = toplevel.getAtom("Title");
    %><%= title %><br /><%
}
%>
```

```
ents = new IDocumentsL
uments());
uments.cache();
documentCount = documents.getCount
(int i = 0; i<documentCount; i++)
    document = new IDocumentProxy
documents.getItemByIndex(i));
    String title = document.getTi
    %><%= title %><br/><%
```

JCR level 1

Read only
Simple & covers a large number of usecases

Fine & Coarse-grained
Content items small to large

Hierarchical
Parent child relationships, sort order

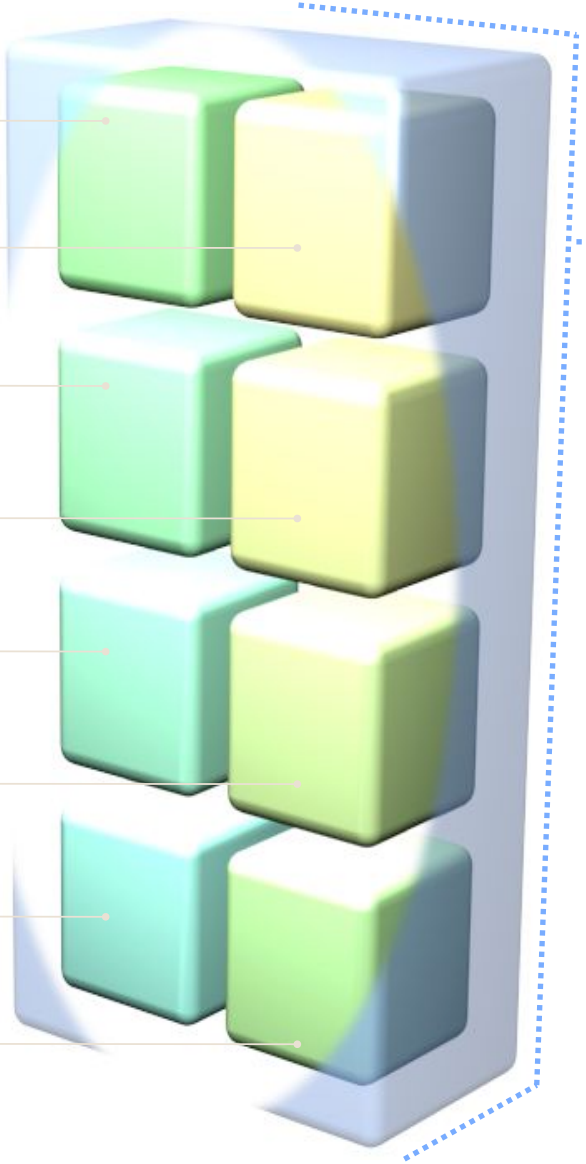
Namespaces
XML-like active namespace support

Property Types
String, Binary, Numbers, Calendar, ...

Node Types
Introspect complex content structures

Query (XPath)
Search and query the repository

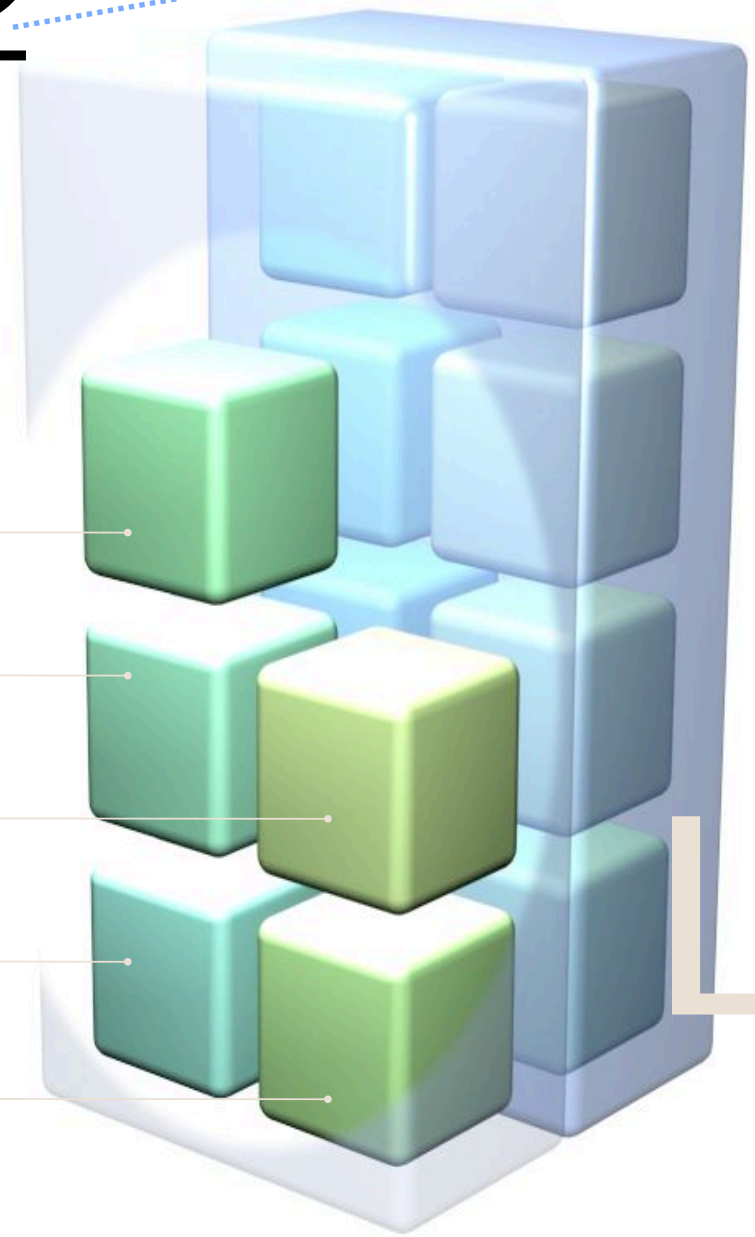
Export
Standardized XML content export



Level 1

L1

JCR level 2



Write
Save changes to a repository

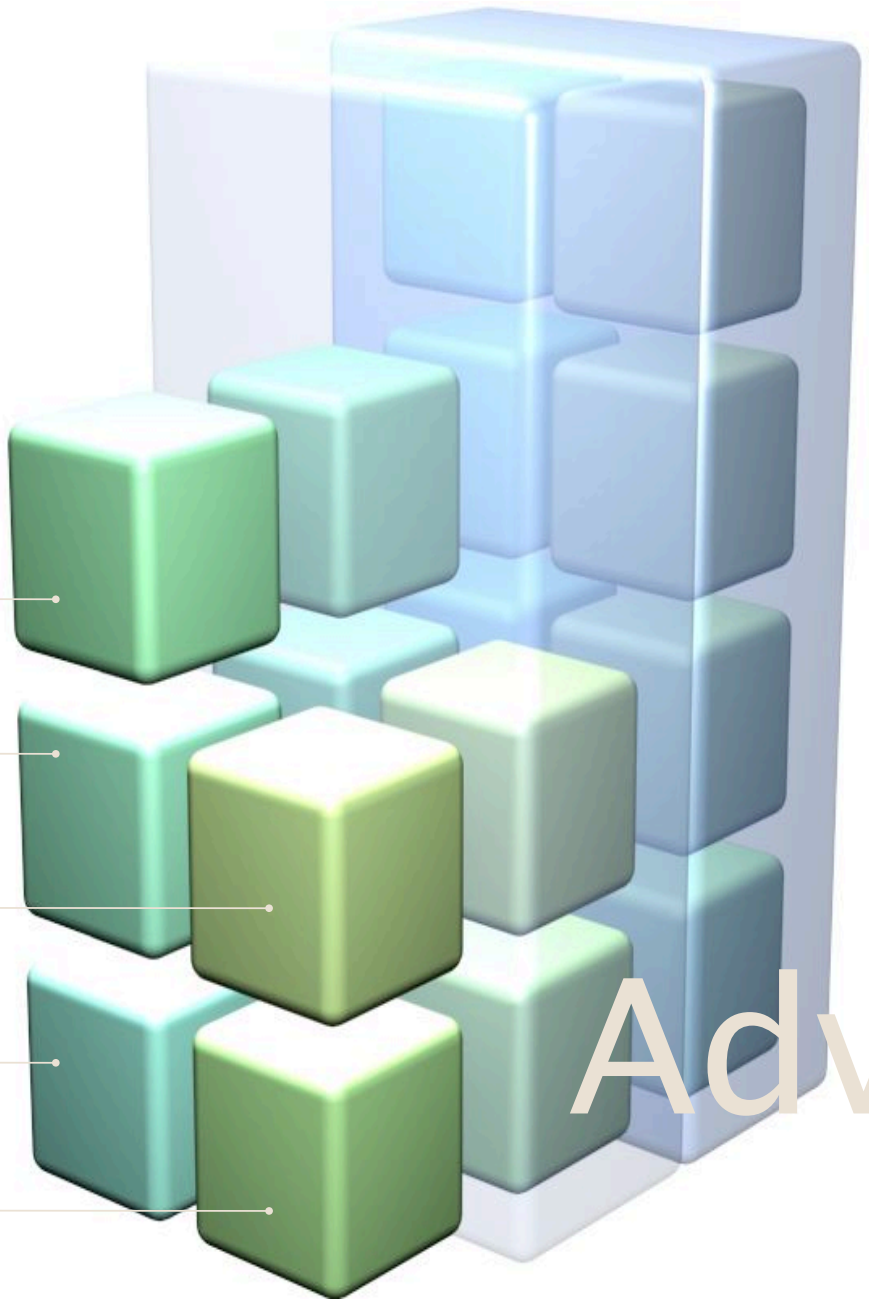
Unstructured
Support for weakly or unstructured Content

Import
Standardized Finegrained Content Import

Referential Integrity
Reference Tracking and Integrity

Access Control
Introspection

Optional



Versioning
Workspaces, Merge, Update, Label, ...

JTA Support
(XA) Transactions

Observation
Monitor changes in the Repository

Query (SQL)
Search the Content Repository using SQL

Locking
Session-based and persistent

Adv

JSR 170 basics

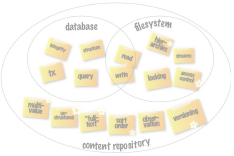
http://day.com/maven/jsr170/javadocs/jcr-1.0/

Repository access

// Repository is often provided by a separate webapp
Repository repository = ...get from environment, JNDI, etc.

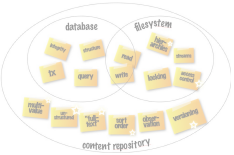
// Need a Session to access repository data
// A Repository can contain several Workspaces
Credentials c = ...get from HTTP request
String workspaceName = “default”;
Session session = repository.login(c, workspaceName);

// Ready to go
Node root = session.getRootNode();



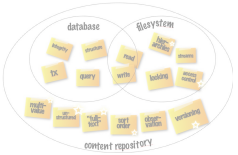
Node and Property basics

```
// get the / node
Node root = session.getRootNode();

// addNode(path, nodeType);
Node child = root.addNode("data", "nt:unstructured");

// setProperty(name, value);
// nt:unstructured node type == no constraints
Property p = child.setProperty("title", "hello JCR");
child.setProperty("date", Calendar.getInstance());
child.setProperty("published", false);
```



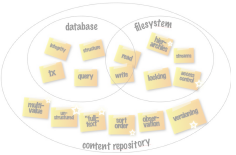
Multivalued Property

```
// write
String [] colors = { “red”, “green”, “blue” };
Property p = child.setProperty(“title”, colors);

// read
Value [] values = p.getValues();
for(Value v : values) {
  // ... do something with v
}
```



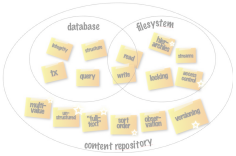
Binary streams

```
// File content is stored under the file node
String dataPath =
  "/data/music/fever.mp3/jcr:content/jcr:data";

// Item == parent class of Property and Node
Item it = session.getItem(dataPath);
if(it instanceof Property) {
  Property p = (Property)it;
  myStreamingService.stream(p.getStream());
}
```



Transient space

```
// same code as first example
Node root = session.getRootNode();
Node child = root.addNode(“data”, “nt:unstructured”);
child.setProperty(“title”, “hello JCR”);

// before this call, nothing is visible to other sessions
session.save();
```

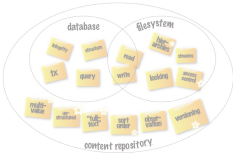


Locking

Node n = (Node)session.getItem(“/data”);

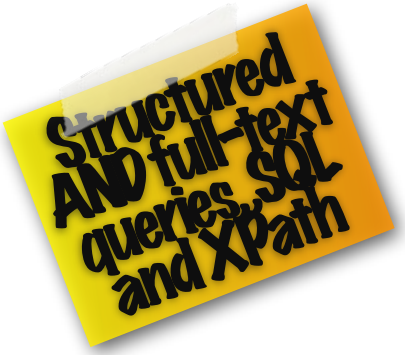
boolean **lockChildrenAsWell** = false;
boolean **unlockOnSessionClose** = false;
n.**lock**(lockChildrenAsWell, unlockOnSessionClose);

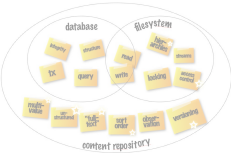
...and later, possibly in a different Session
Node n = (Node)session.getItem(“/data”);
n.**unlock**();



Searching

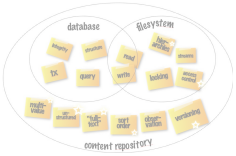
```
QueryManager qm =
session.getWorkspace().getQueryManager();

// SQL query (XPath is also supported)
String statement = "select * from nt:unstructured”;
Query q = qm.createQuery(statement, “sql”);
NodeIterator it = q.execute().getNodes();
while(it.hasNext()) {
  Node n = it.nextNode();
  doSomethingWith(n);
}
```





Observation

```
// Register a listener for Repository events
final int events =
  PROPERTY_CHANGED |PROPERTY_REMOVED;
ObservationManager om = session.getWorkspace
().getObservationManager();
om.addEventListener(this, events, “/data”, ...);

// Event callback
public void onEvent(EventIterator it) {
  while(it.hasNext()) {
    final Event e = it.nextEvent();
    doSomethingWith(e);
 }
```



JCR basics summary

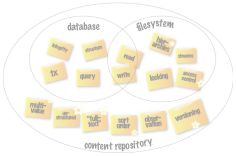
Get a **Session** on a **Workspace** in the **Repository**.
Create or retrieve **Node** and **Property** objects using the Session, with rich Property value types.
Save the Node, Property and/or Session **explicitely**.

Use the **QueryManager** to search - but only when needed, a good hierarchy goes a long way.

Use Node.**lock**() and Node.**unlock**() if needed.

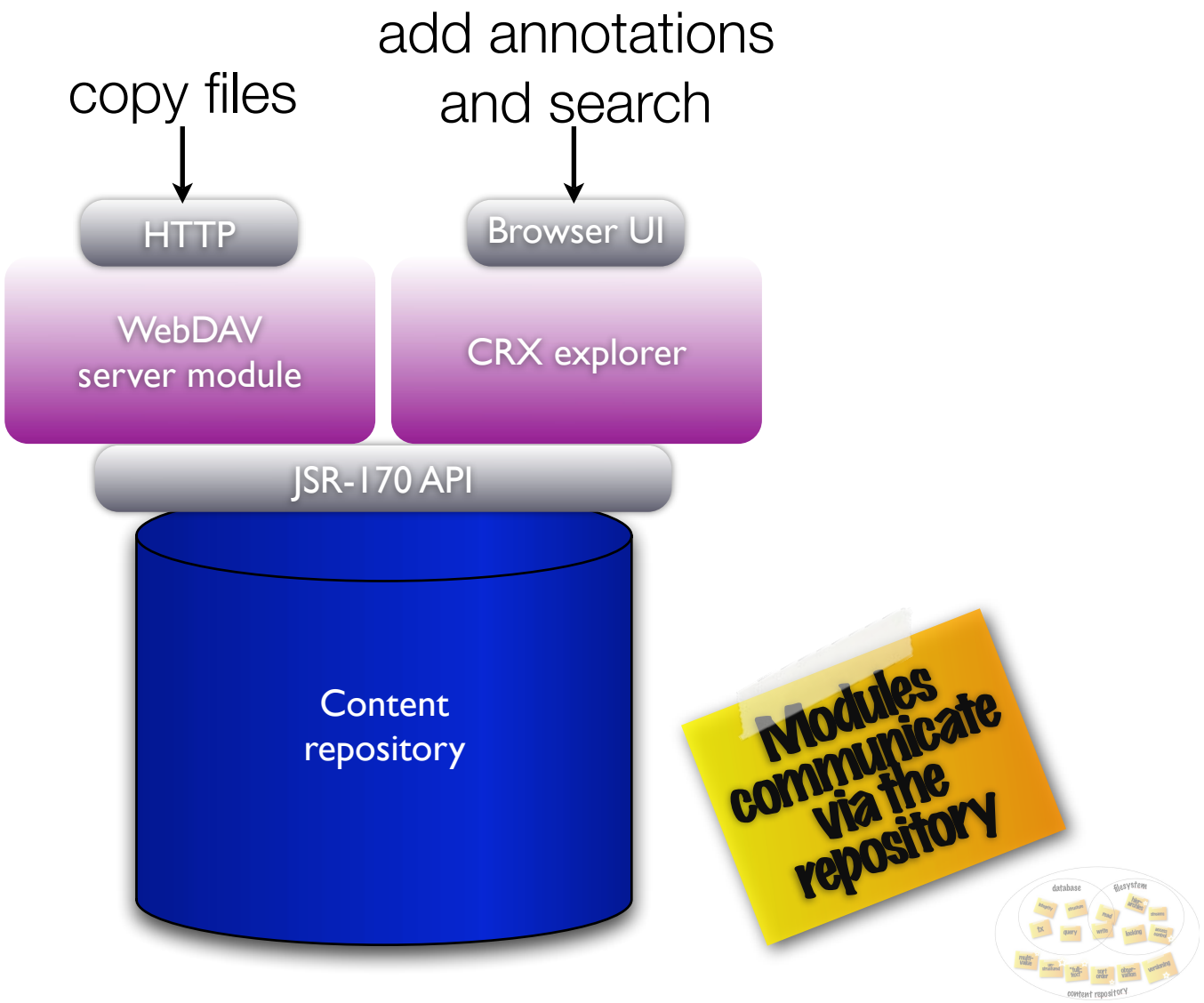
Register **EventListener** objects to be informed of specific changes to Repository items.

That's pretty much it.

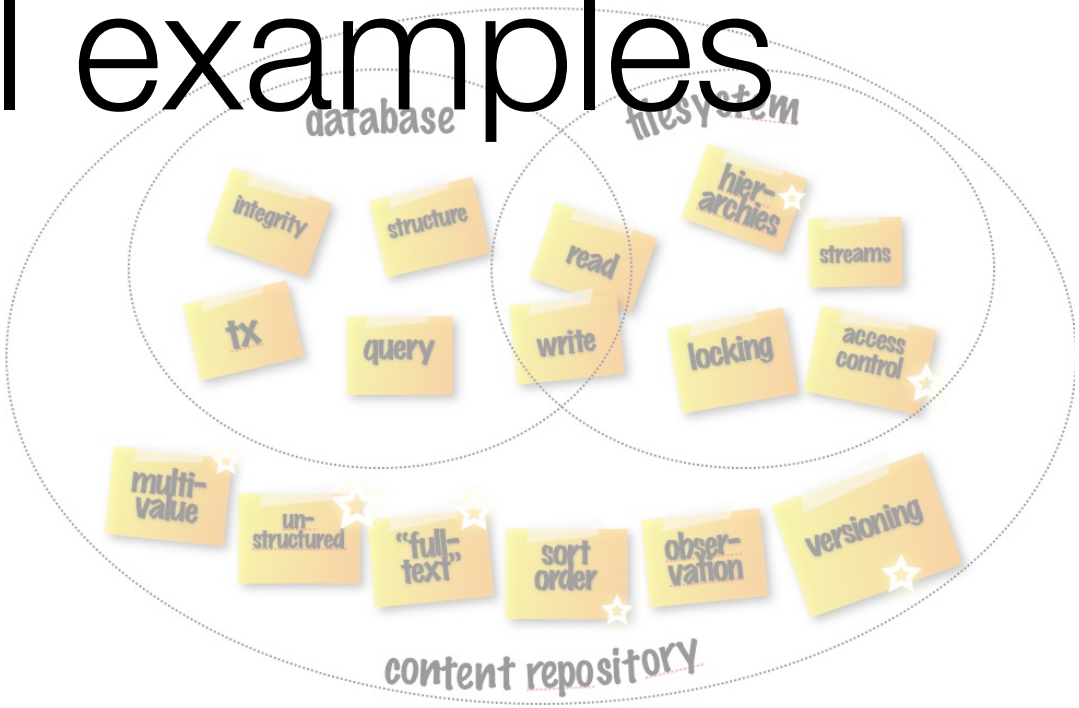


File annotation and (full-text) search demo

File annotation and search demo



Javashell examples



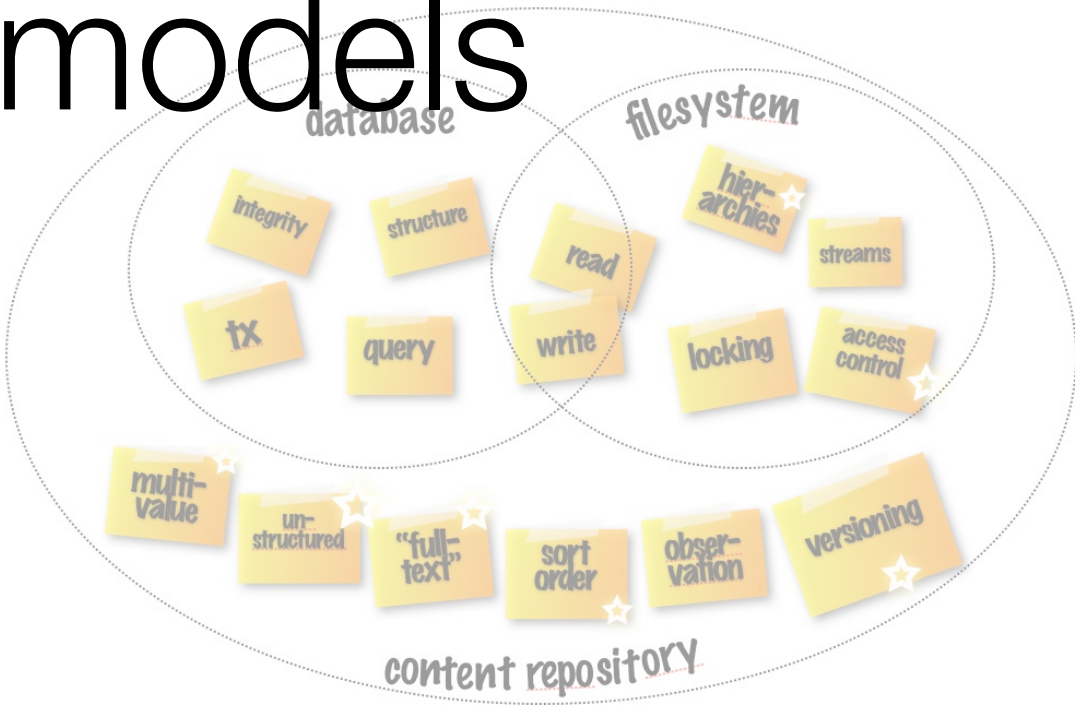
Javashell examples

Edit script
Create new script
First example
Add/retrieve/delete
Simple SQL or XPath search
Versioning
Tagging files
Observation

javashell: Simple SQL or

```
// javashell provides a JCR Session,
QueryManager qm = session.getWorkspa

// SQL query
String statement = "select * from nt
String language = "sql";

// Uncomment these two state
// statement = "//conte
// language = "xpath";
```

Content models

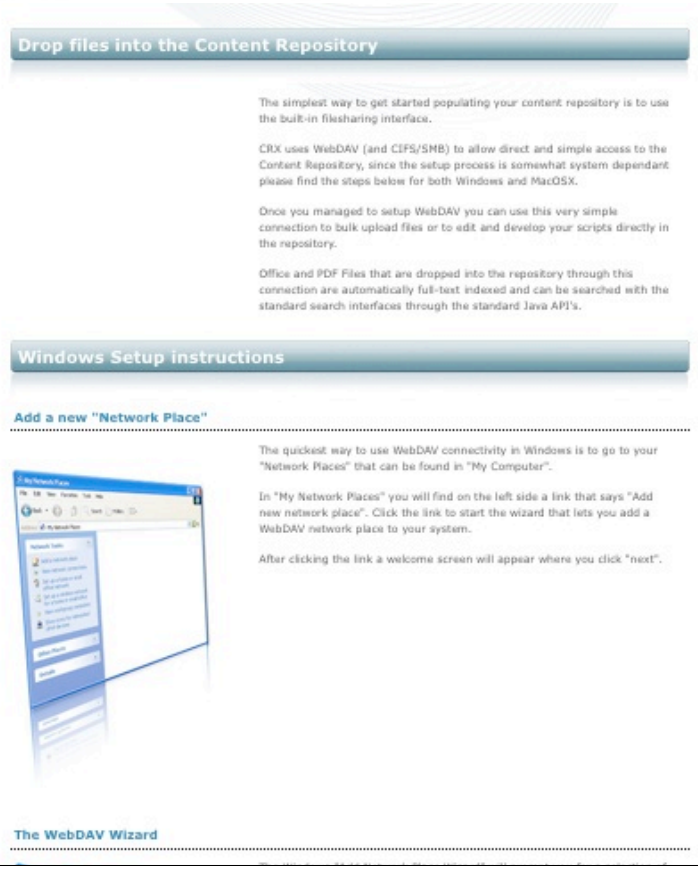


CRX docs content

Path /content/crx/docs/webdav/jcr:content

- /
  - etc
  - apps
  - var
  - libs
  - content
    - crx
      - docs
        - webdav
          - jcr:content
          - files
            - connecttoserver.jpg
            - networkplaces.jpg
            - webdavwizard.jpg
        - jcr:content
        - jcr:links

Name »	Type	Value
.	nt:unstructured	
body	String	<h1>Drop files into the Cont
jcr:description	String	A step by step guide that wa
jcr:title	String	WebDAV Setup



CRX docs - in detail

webdav nt:unstructured
- sling:resourceType: crx/docs

jcr:content nt:unstructured
- body: <h1>Drop files into the Content Repository</h1> <p...
- jcr:description: A step by step guide that walks you through the initial setup of ...
- jcr:title: WebDAV Setup

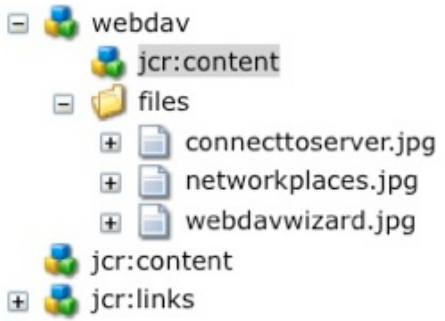
files sling:Folder
- jcr:created: 2008-11-19T15:50:48.436+01:00

connecttoserver.jpg nt:file
- jcr:created: 2008-11-19T15:50:48.437+01:00

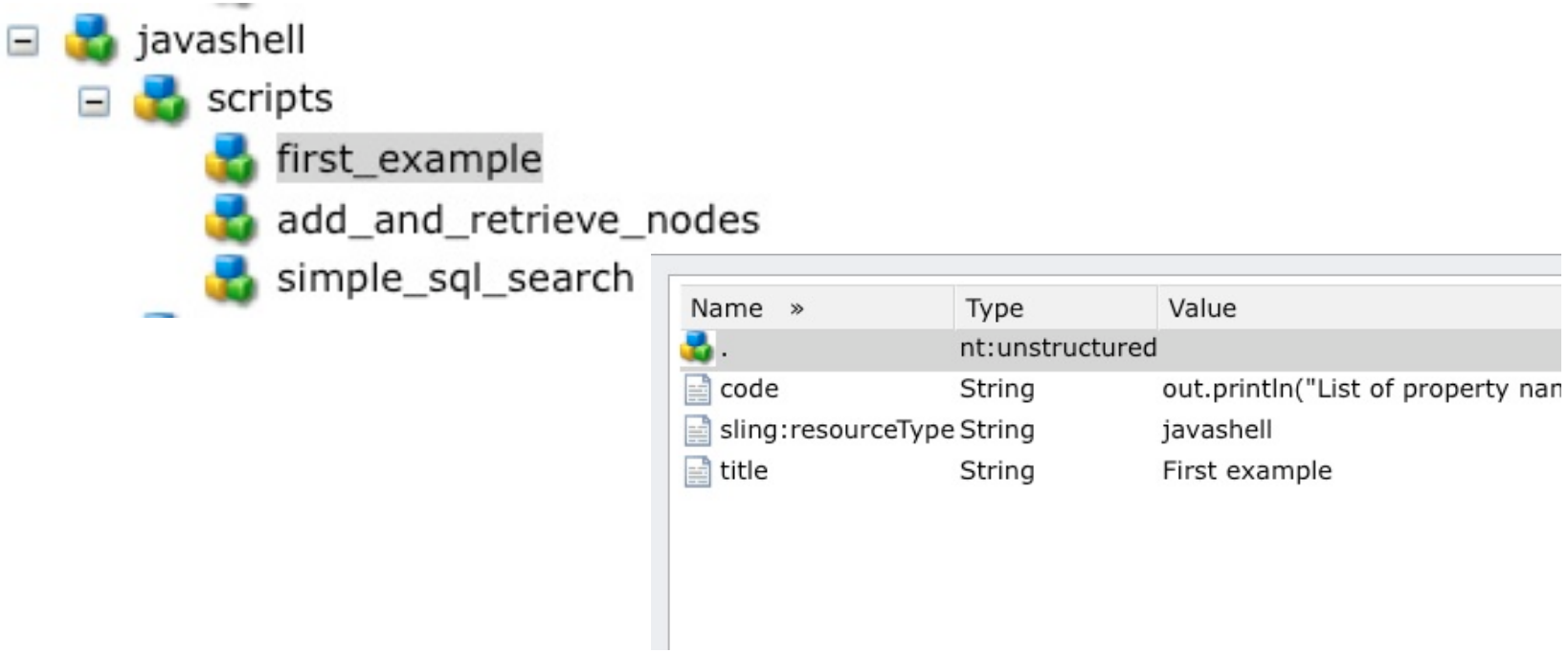
jcr:content nt:resource 20b99009-d728-455e-901f-f842f53665d1
- jcr:data: /9j/4AAQSkZJRgABAgAAZABkAAD/7AARRHVja3kAAQAEAAAAPAAA/+4ADkFkb2JlAGTAAAAAAf/bAIQA...
- jcr:lastModified: 2008-11-11T09:58:56.000+01:00
- jcr:mimeType: image/jpeg

networkplaces.jpg nt:file
- jcr:created: 2008-11-19T15:50:48.439+01:00

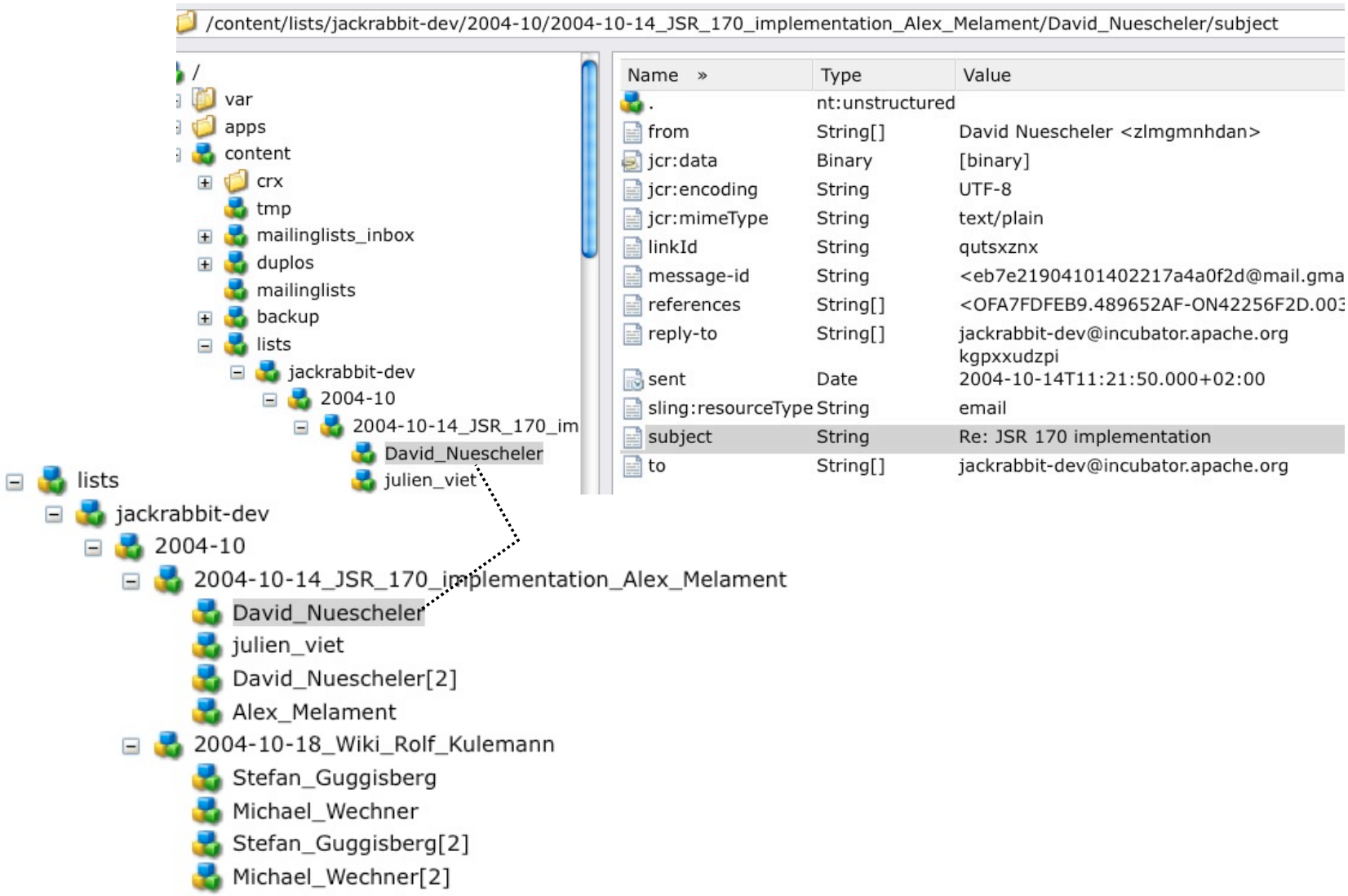
jcr:content nt:resource 65f2b426-0875-4f7e-9899-f8462f048b43



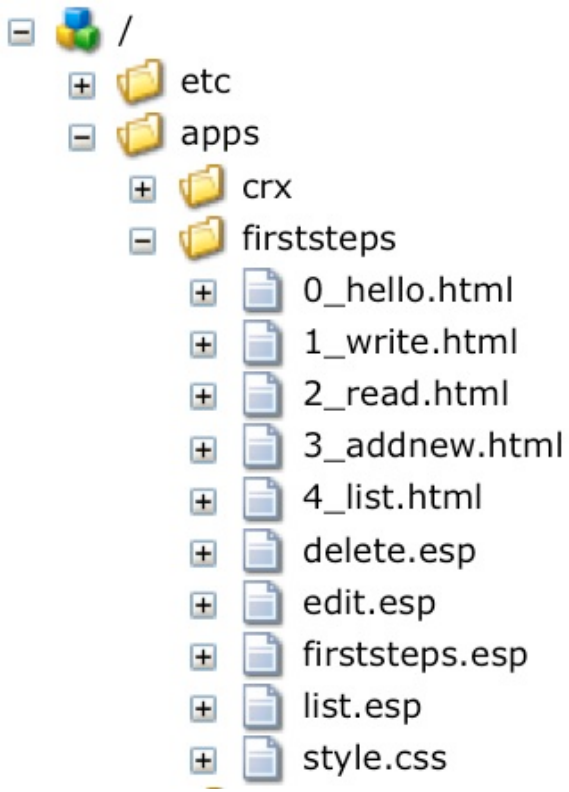
Javashell content



Mail content

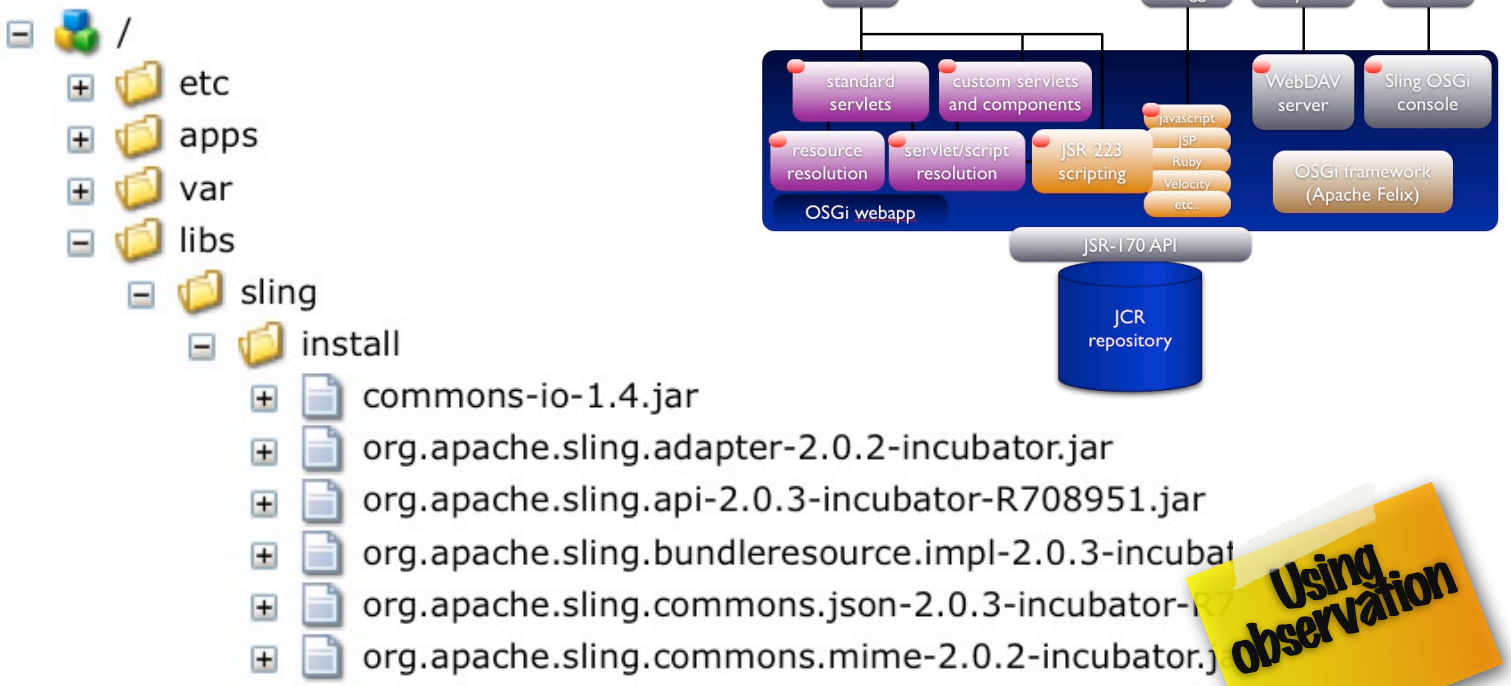


Templates / scripts



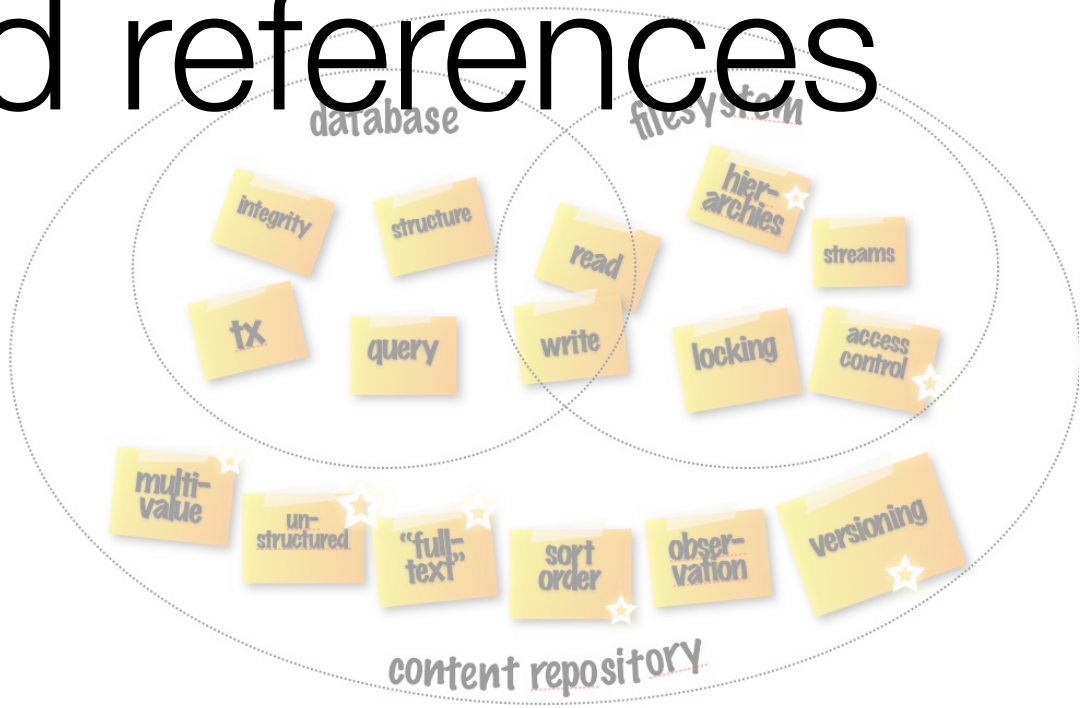
Template pages and processing scripts.
Path-based script selection in Sling.

Code and services!



Sling can load OSGi code modules, services and configurations dynamically from the repository.

Links and references



References

JCR spec, final release (fairly readable):
http://jcp.org/aboutJava/communityprocess/final/jsr170/index.html

List of articles, tutorials, etc.
http://wiki.apache.org/jackrabbit/JcrLinks

Sling: RESTful access to JCR + scripting framework and javashell:
http://incubator.apache.org/sling

My own JCR links:
http://delicious.com/bdelacretaz/jcr

More slides at slideshare.net (if that works...)
under the “jcr” and “sling” tags.