Content Management

JSR-170 and Content Hosting & Repository Integration

Dr Ian Boston
CTO,
CARET, University of Cambridge

Good afternoon, my name is Ian Boston, I am CTO for CARET at the University of Cambridge and a core committer to the sakai code base.

Technical Presentation

- The importance of Content Management in VLE/VRE environments.
  - Content Management
  - Sakai Content Hosting Service
  - Java Content Repository Standards. JSR-170, JSR-283
  - Institutional Repositories and Preservation

I am here to talk you about Sakai and Content management. Progress on work with JSR-170 and the future plans, and where Sakai fits within the institution relative to institutional repositories.

Content Management

"Content management, or CM, is a set of processes and technologies that support the evolutionary life cycle of digital information. This digital information is often referred to as content or, to be precise, digital content. Digital content may take the form of text, such as documents, multimedia files, such as audio or video files, or any other file type which follows a content lifecycle which requires management." Wikipedia

What is content management. The management of the lifecycle surrounding content, from it creation through use, modification and reuse through to archiving an preservation.

Content Lifecycle

- Supports the Lifecycle of Digital Information
- Process Driven
- With roles, but open to all.

"Content management, or CM, is a set of processes and technologies that support the evolutionary life cycle of digital information. This digital information is often referred to as content or, to be precise, digital content. Digital content may take the form of text, such as documents, multimedia files, such as audio or video files, or any other file type which follows a content lifecycle which requires management." Wikipedia

The lifecycle is managed by processes, informal and formal depending on context.

The roles within those processes vary depending on the level of value attached to the activity and the participants.

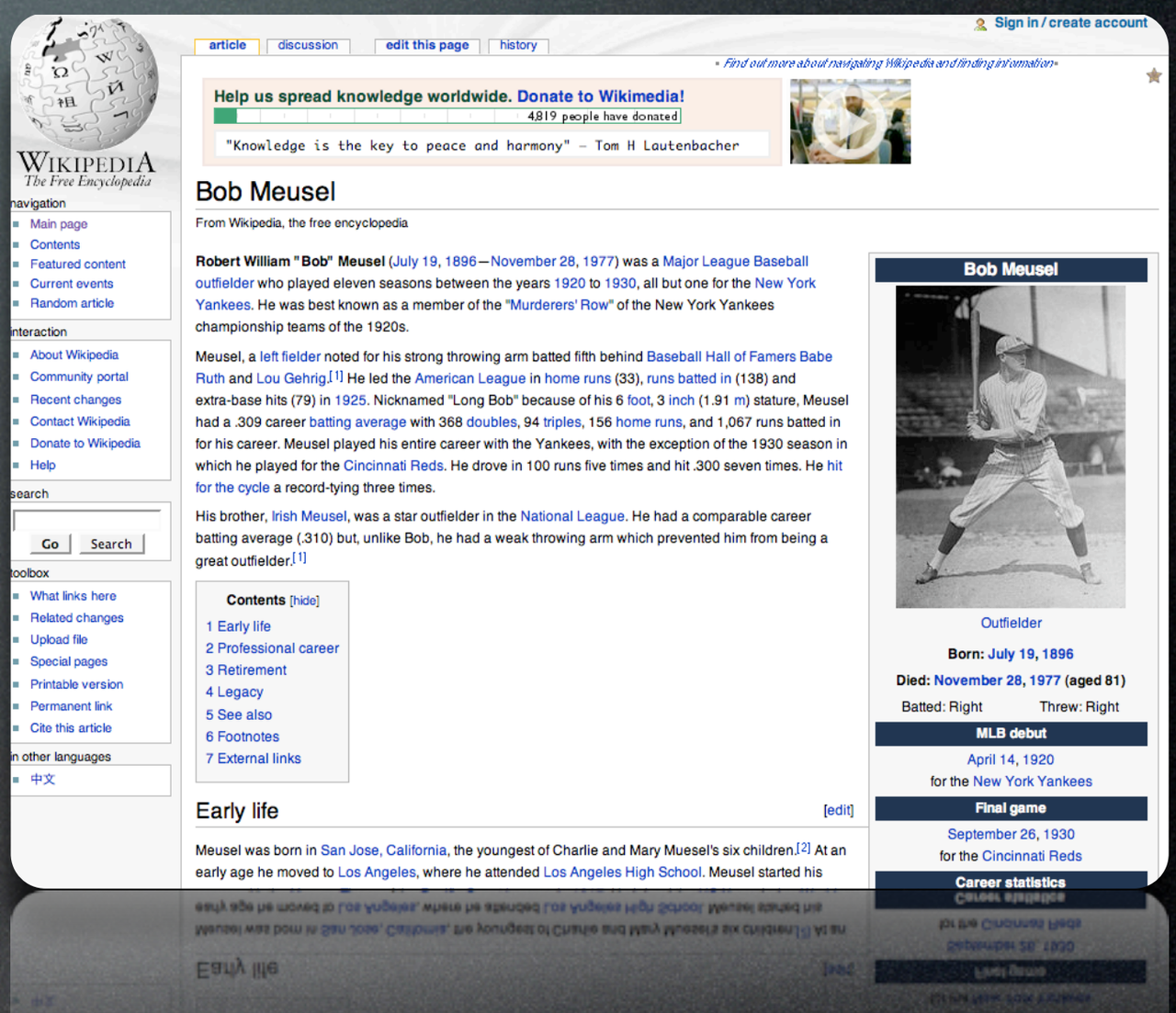
Content Lifecycle

All content follows a lifecycle, most of the time we dont even know it.

Its created for use. Early drafts are improved and reused.
As more value becomes attached, review leads to acclaim from peers, the media and community. Preserving the value becomes important, and that preservation leads to reuse.
We throw away huge volumes of content, as we should do.
Only preserving that which is of value.

CM Creation Environments

- Structured
  - publication driven
- Chaotic
  - social user created
- Collaborative
  - research and education



The creation environment drives the nature of the content management systems.

Prior to the rise of Web 2.0 and social networks, formal content management was found in corporate websites, project offices, legal departments where the risk associated with failing to manage the content were great enough to formalize the processes.

With the rise of Web 2.0 we have all been engaged in creating and managing content as users. The history of any page on wikipedia shows this in progress.

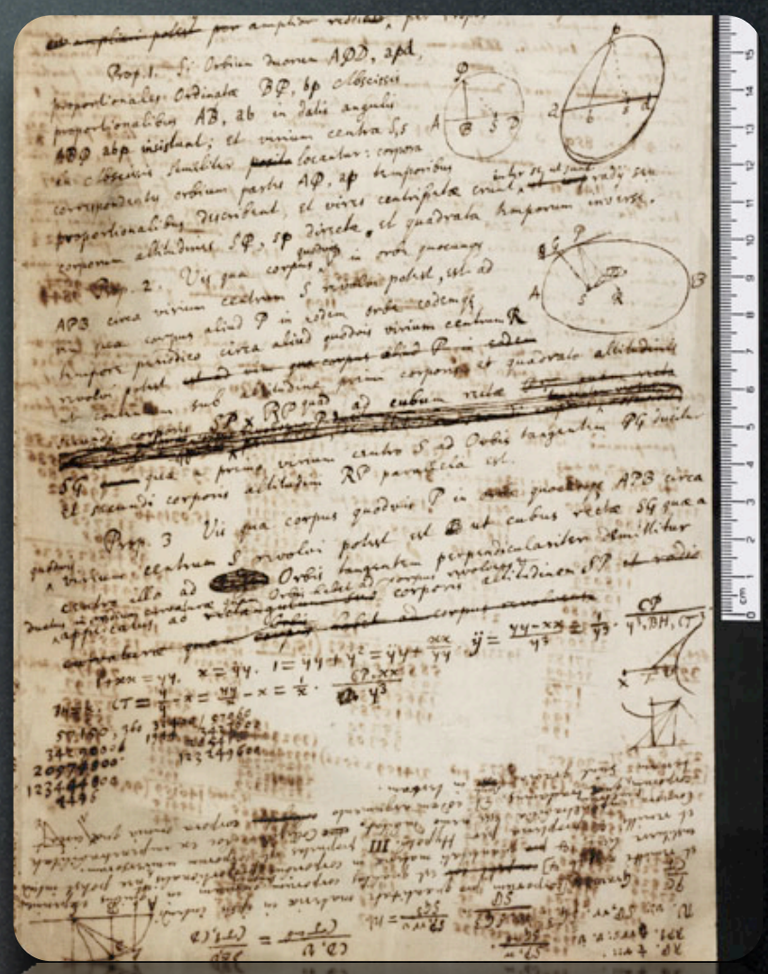
Commercial interests, beliefs and misinformation are corrected and moderated for the good of the many by an army of editors supported by a open community.

More recently, in protected environments web 2.0 content management is supporting rapid collaboration. Google Docs, Spereadsheets.

Sakai is an example of such a collaborative environment, and its flavor of content management is targeted at supporting this collaboration.

CM Preservation Environments

- Unstructured
  - backup, tape, cardboard boxes
- Managed
  - Meta Data, Format Conversion, Secure and cared for.



Having created the content, invested time and money in it, we preserve the content.

It may be as unreliable and chaotic as a backup.

In the 70s reams of Social Science research was put into cardboard boxes and stored.

Over generations Libraries have been tasked with preserving manuscripts. More recently, these manuscripts have been digitized.

At Cambridge there are numerous digitization projects of valuable manuscripts. This is a page from one of Newtons notebooks.

The Parker Library at Cambridge is digitising the original works of William Chaucer in collaboration with Stanford University. But, preservation of digital assets is not always about the preservation of ancient artifacts.

New Research Datasets contain immense value for future researchers. If left on tape or backup, these are lost forever, there are certainly holes in the last 10 years.

But, we should not keep everything. Managed preservation identifies those items that are worth preserving from the sea of digital data.

Sakai Content Hosting

Sakai content hosting is one small part of this larger content management environment.

It supports the creation and sharing of content within he educational and research environment.

Collaborative Content

- Stores Files
  - Uploaded by authorized users
    - Accessible to others
  - Captures metadata
  - Life blood of the VLE/ VRE
- A Sakai Service
  - Resources
  - Assignments
  - Portfolio

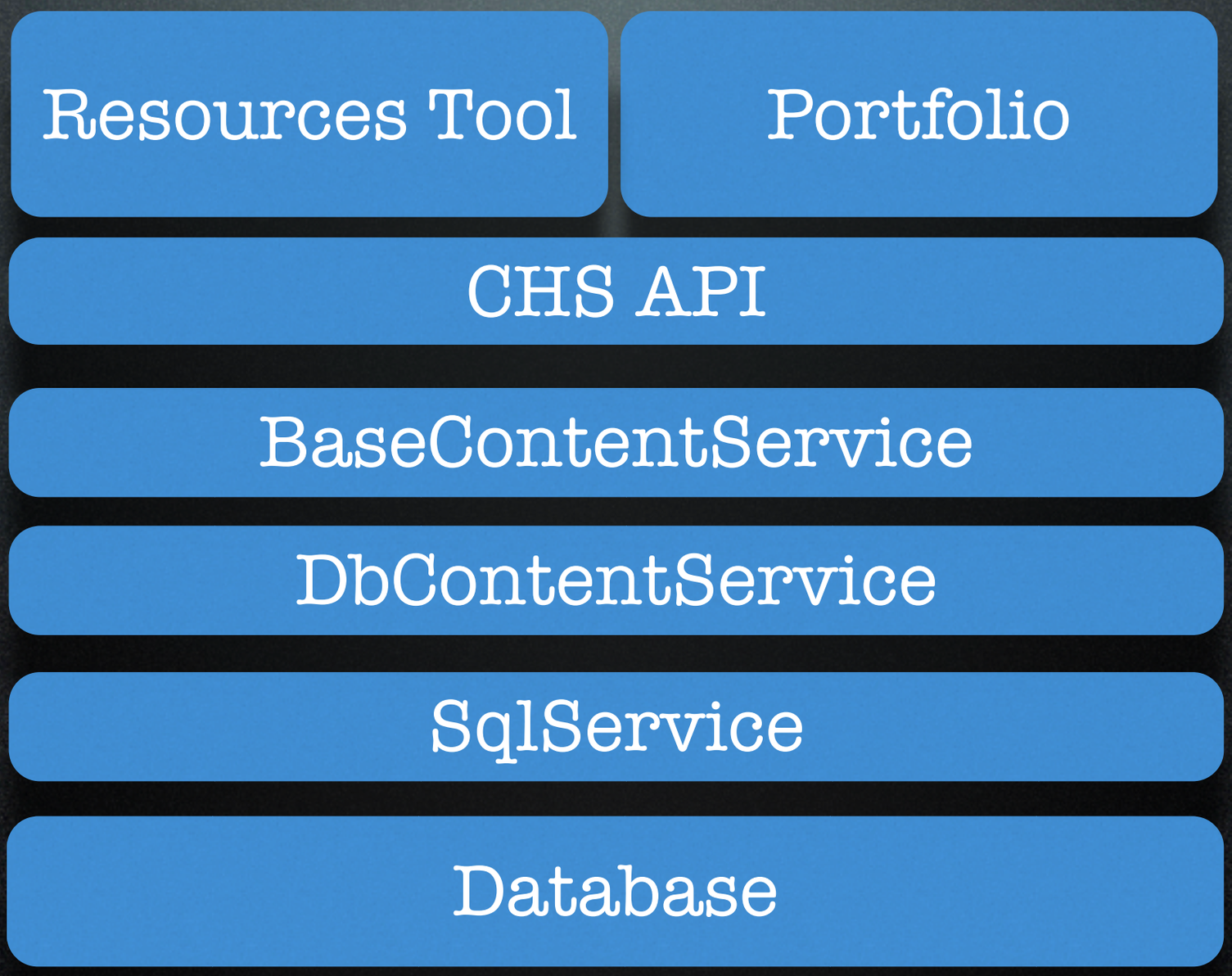
Put simply it stores files and controls access to those files. It collects them together in a work-site for organization and make the content contained within them accessible to members of that work-site.

It is the central service supplied by Sakai and without it much of the value of Sakai would be lost. We have many course sites at Cambrdige where 90% of the activity is in the Resources tool.

It becomes more than a simple content store when it engages with the content management process.
Timed release enables course administrators to configure the whole year so that content is released according to a schedule. The capture of basic metadata assists reuse and occasionally encourages more complete meta data capture.

It works as a Sakai service, and is used by the Resources Tool, Portfolio Tools, Wiki for attachments, Announcements and others.

Technical Structure

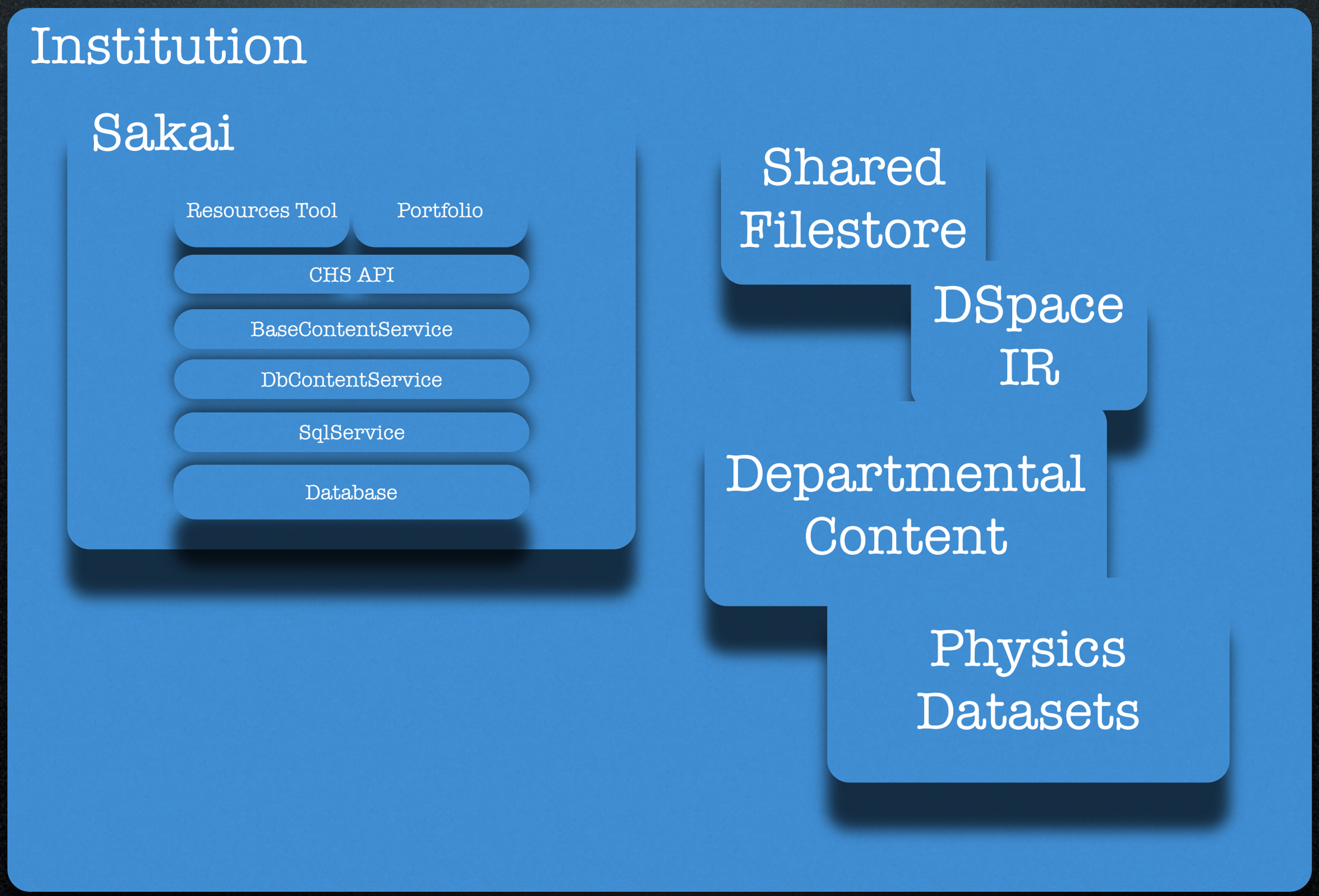


The Structure of the Content Hosting Service is a

BaseContentService, that implements the API,

and then an extension class that specializes the BaseContentService to use database storage.

Technical Structure



The technical structure of sakai sits within the institution as an island of content.

There will be other islands more so or less so integrated with one another.

DSpace@Cambridge is out Institutional Repository for preservation. Many departments have their own file stores and central computing services provide shared file store across the campus.

The DbContentHosting Service does not integrate with the campus, and is outside any content management infrastructure. Consequently it relies on users uploading content to course and project sites.

Java Content Repository

As Information Technology advances, we no longer need to search for local solutions to common problems. The standards movement is constantly collaborating to provide cross vendor standards that open up competition and options for us all, whilst attempting to reduce incompatibility. Java Content Repository is one of these standards.

JCRv1 JSR-170

- Java Community Process
  - Java Standards Request submitted October 2003 by David Nuescheler of Day Software
    - Wide industry support
- Final Release June 2005

JCR version 1, or JSR-170 was/is a Java Standards Request submitted to the Java Community Process, run by Sun Microsystems, by Day Software in 2003. Day Software, then, probably a Small content management software vendor, saw the need to arrive at a standard specification, reference implementation and validation toolkit to enable interoperability between large scale content management systems.

The JSR-170 standard received wide interest and participation from big vendors with a stake in the space.

The Apache Foundation formed the Jackrabbit project and pulled in some of their best contributors to work on the Reference Implementation.

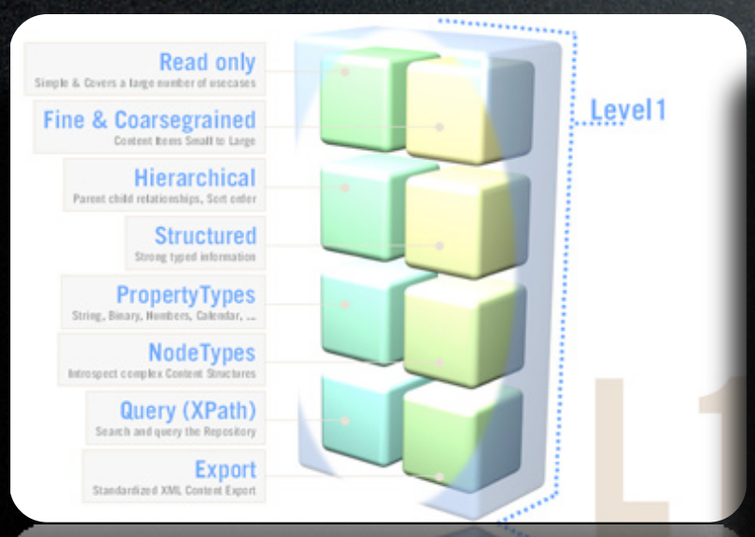
As the standard was released, so was Jackrabbit as a RI, and the Validation Tool Kit. Day software contributed heavily and there is a considerable amount of shared code base between Jackrabbit and Day's commercial products.

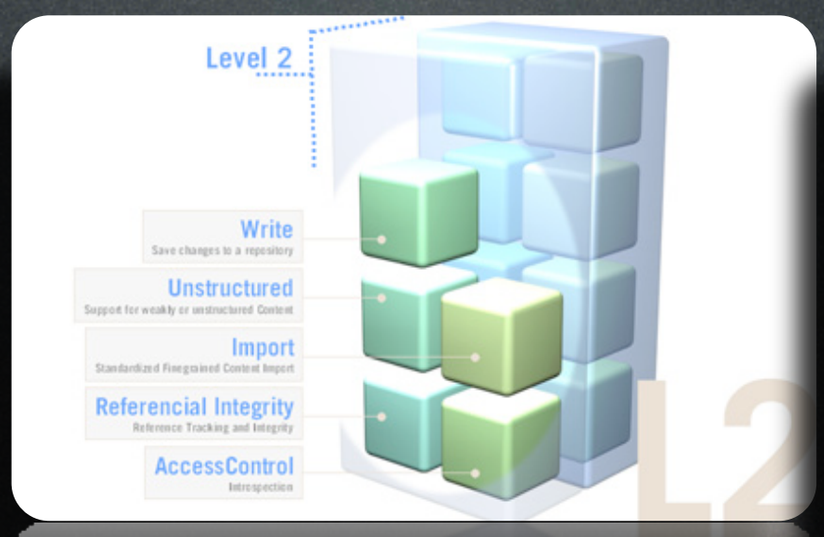
Jackrabbit is not just a play RI, is a fully functional, production quality JSR-170 implementation.

Unfortunately in spite of wide industry interest, there has been limited adoption of the JSR-170 standard in commercial offerings. This is as a result some of the parts that were left out of the standard, and the commercial vendors needing a validated business model for participating.

JCR Levels

- Level 1 : Ease of Adoption, Covering many use cases
- Level 2 : Write-able Repository
- Advanced Options





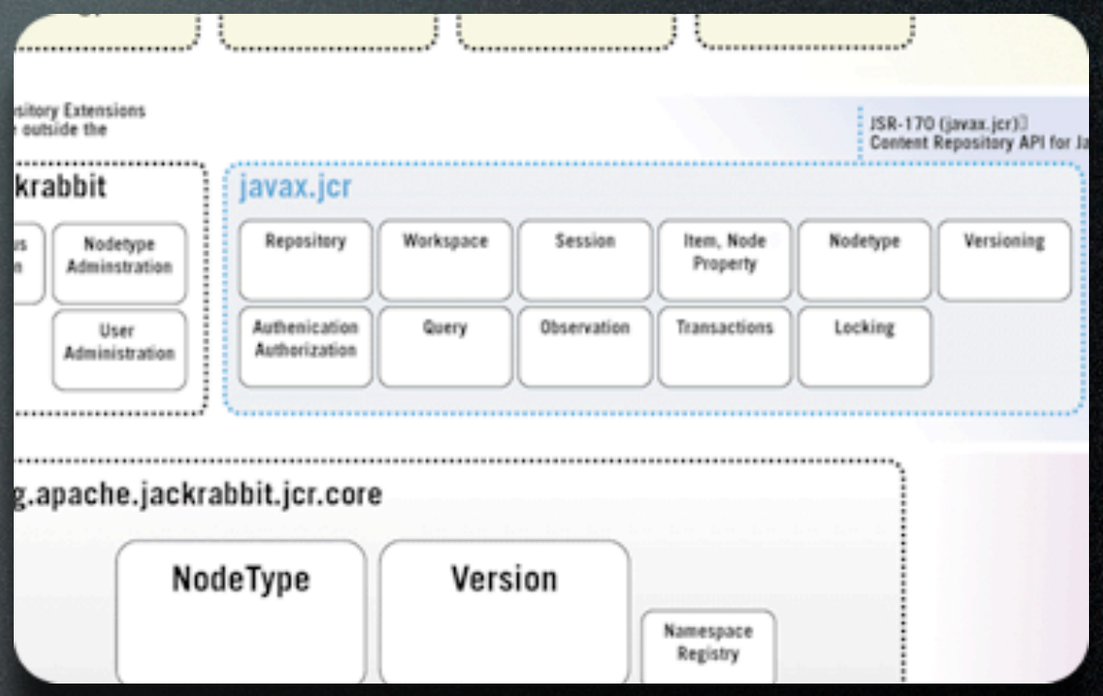
Diagrams from http://jackrabbit.apache.org © 2004-2007 The Apache Software Foundation

The standard expresses 3 levels of compliance.
Level 1 allowing read only re-use of content
Level 2 enabling read-write with some additional supporting functionality,
and Advanced offering the full set of functionality.

Most Content Repositories will have all the features of JSR-170, however some may decide not to expose all those features. Jackrabbit, naturally has all.

JCR Concepts

- Nodes, Items, Properties
- Workspace
- Repository
- Versioning, Node Types
- Observation,Transactions, Locking



It is the simplicity of the concepts within the a JCR that make is usable.
The concepts were, no doubt developed by the combined knowledge a large expert group over the period of 2 years.

A JCR is based on a Repository, within that repository there are work spaces.
and within each workspace a hierarchical tree of Items. Those items being nodes and properties.

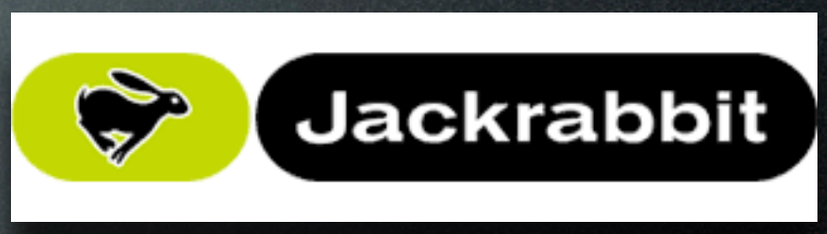
Nodes being containers for other Nodes and properties.
The repository itself manages versioning of any item and the definition of the node types, however in JSR-170 the definition of a node type itself was outside the specification.

The advanced implementations of the standard will expose observation of nodes, transactions and explicit locking.

Querying and searching by various means is also part of the standard.

JCR Implementations

- Open Source
  - Apache Jackrabbit RI
  - Alfresco
  - Exo
- Commercial
  - Xythos (active with Sakai)
  - many others



The JCR Implementations, that Sakai has had contact with are Jackrabbit which integrates well within Sakai, Alfresco which has some dependency issues still to be addressed, and Exo which is under investigation. Commercially, Xythos have recently released a JSR-170 implementation based on the Jackrabbit Service Provider Interface (SPI).

This is an interface that sits in the depths of Jackrabbit and enables other systems to gain JSR-170 compliance at lower effort.

Julian Reschke did this implementation and I worked closely with him to iron out real use problems by embedding it inside Sakai, before it was announced. There are other JSR-170 implementations out there, I believe Oracle have one.

JCRv2 JSR-283

- Public 13 July 2007
- Jackrabbit Trunk
- Much greater adoption.

Alfresco, Inc. Alkan, Kilinc. Apache Software Foundation. Art Technology Group Inc.(ATG). Asplund, Marko. BEA Systems. Borland Software Corporation. CoreMedia AG. Dambekalns, Karsten. Day Software, Inc. EMC Corporation. eXo Platform SARL. Filenet Corporation. Flickman, Michael. Flowers, Andrew. Fraunhofer-Gesellschaft Institute FIRST. Gershon, Gary M. GX Creative Online Development BV. HIPPO. IBM. Interwoven. Mediasurface Ltd. Mobius Management Systems, Inc. Myers, Jimmy D. NCsoft Corporation. Niemeyer, Patrick D. NUXEO. Open Source Applications Foundation. Open Text Corporation. Oracle. Pimenta, Marshall. Raboch, Walter. Red Hat Middleware LLC. Reschke, Julian. SAP AG. SAS Institute Inc. Sun Microsystems, Inc.Tiwari, Shashank. Vignette. Wadia, Zubin. Weisinger, Richard. Wilkerson, Peter L.Winters, James. Wipro Technologies. Xythos Software, Inc. Zitting, Jukka

Some commercial vendors felt too early to commit heavily due to the uncertainties around access control and node type management. Again Day Software proposed JCRv2 in the form of JSR-283 which went public in July 2007. Over the past 2 years has attracted a far greater level of interest and commitment.

Jackrabbit trunk has already implemented JSR-283 and there are bound to be other commercial implementations. Those who have bound to the Jackrabbit Service Provider Interface may only have to do minimal rework to gain JSR-283 compliance.

JCR & Sakai

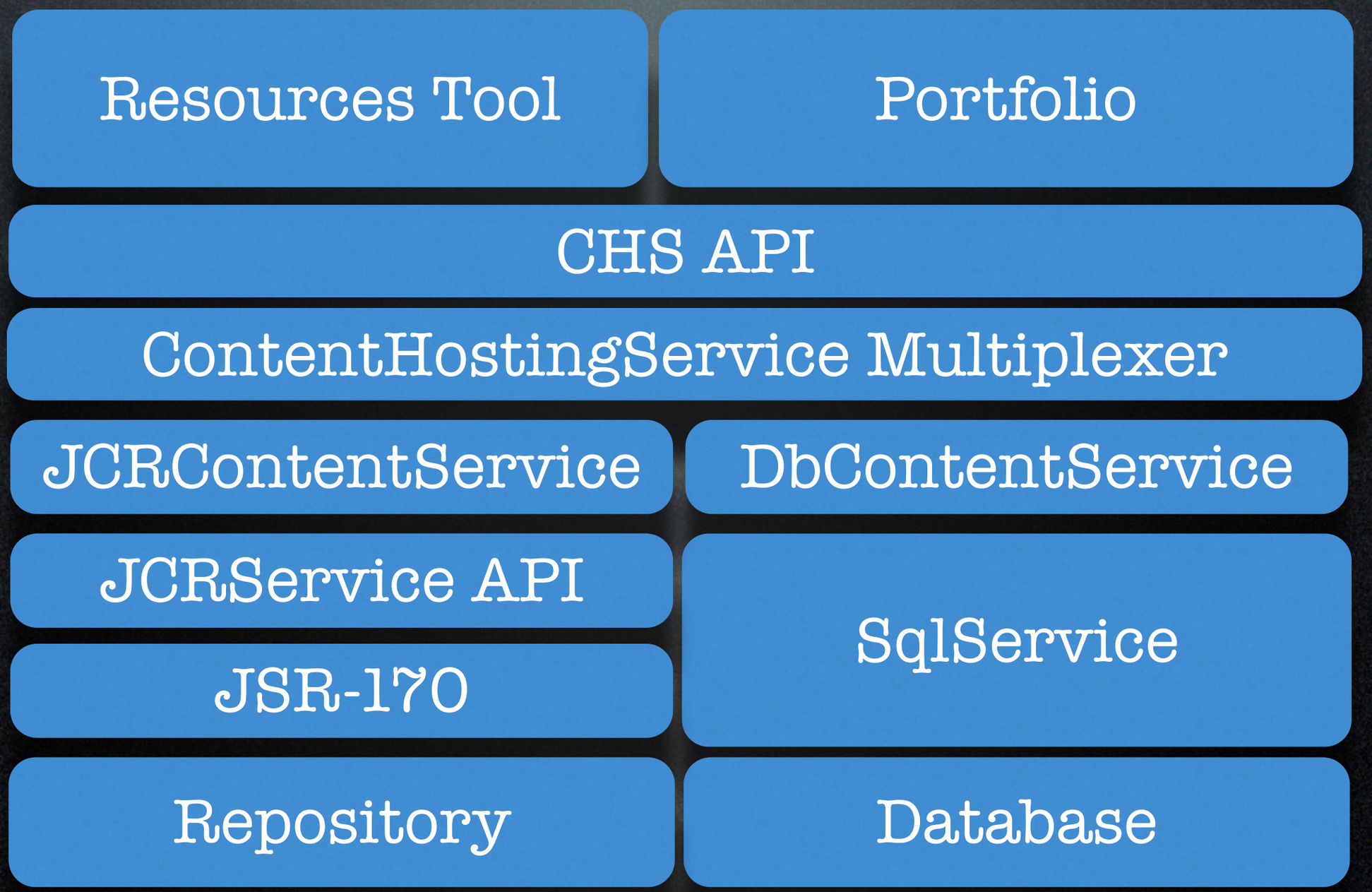
So 170 and 283 was all very interesting as a standards track, but where does it leave Sakai.

We have a content island sitting on a Database that has served those running it well over the past 3-4 years.

When compared to JSR-170, Sakai CHS has only a handful of the features that the expert group considered relevant to a Java Content Repository. We have a choice, adopt the standard or re-invent it.

We adopt.

JCR-CHS-Sakai 2.4.x/2.5



But we need to migrate the community, so in 2.4.x, 2.5 and trunk there is a JCRService API, which is used by a JCRContentService. This service, which implements the CHS API is used by a CHS multiplexer that enables both the DB service and the JCR service to co-exist inside the same sakai installation.

Only one repository is active above the API.

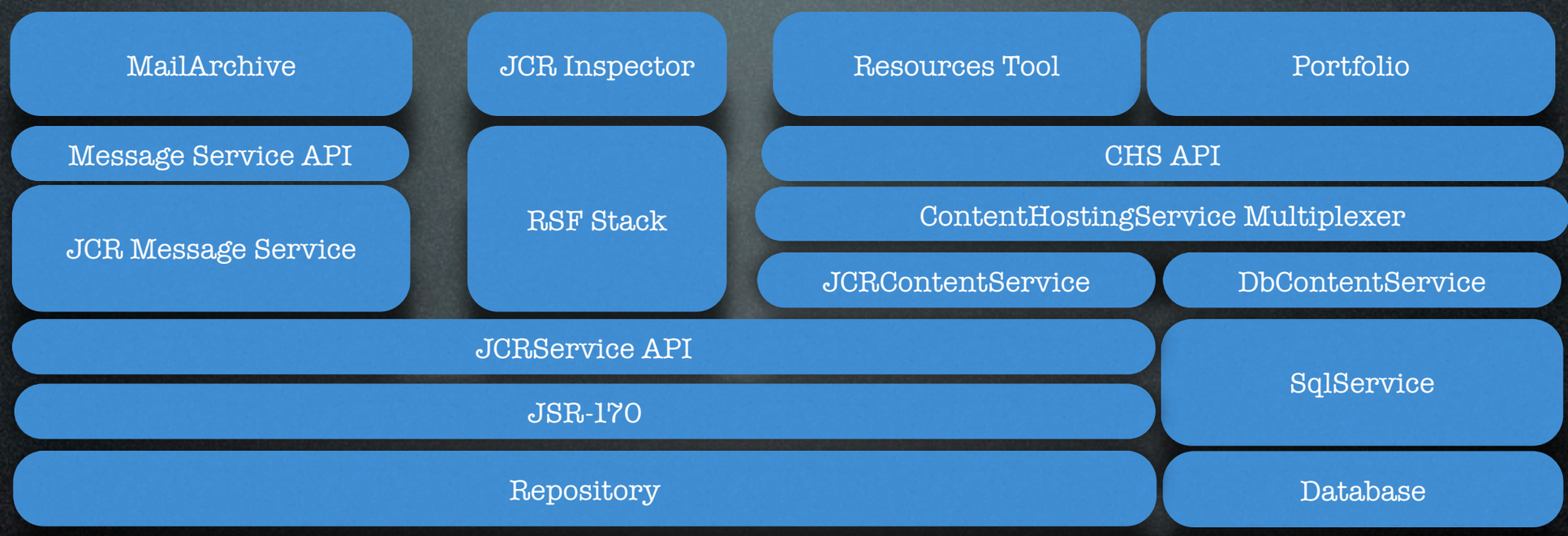
Why?

Michigan, have 3M files and 1.5TB of content.
Indian have about 2M files and 2TB of content.
Cambridge have 200K files and 100GB of content.

We might be able to schedule the downtime, but the larger schools will not be able to schedule downtime to perform a migration, so if both services are live, we can migrate content with the system up and running.

It also enables schools to make a choice, JCR or DB. Sakai will continue to support the DB Content Hosting service for some time, but eventually it will become deprecated.

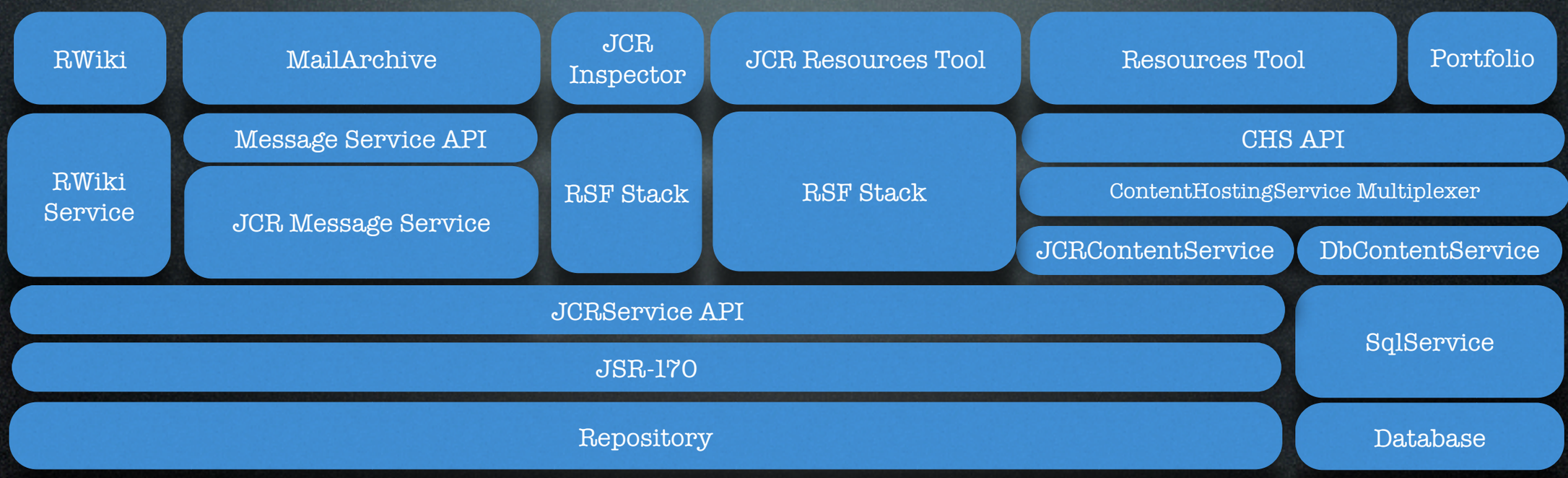
JCR-Sakai 2.4/2.5



In 2.4.x and 2.5 the benefits of JCR start to become apparent.
We have a JCRMessageService that supports the current mail archive and allows content to be stored, as content in the JCR repository.
We have a RSF stack that makes it really easy to create a tool that uses JSR-170 via the JCRService api, really easily, and the Content Hosting Service can be backed by JCR.

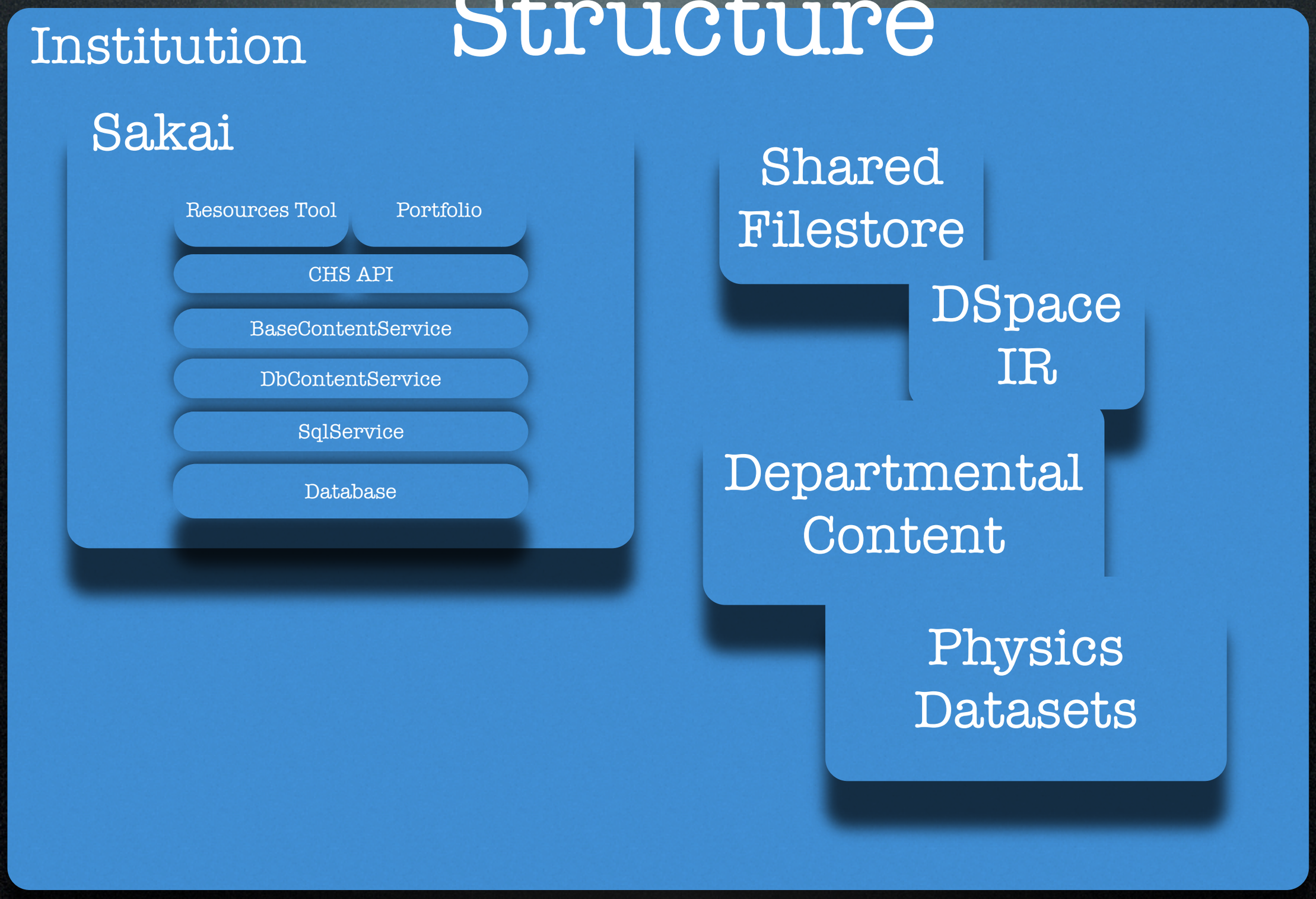
For operations teams, JCR makes lower demands on the Database than DbContentService, and is faster under load. Before we did fixes in 2.4.x and 2.5, webdav was about 10 times slower in Db compared to JCR. Now its about 2 times slower under load, but Jackrabbit only opens a limited number of DB connections regardless of the active number of sessions. It has its own transaction monitor.

JCR-Sakai after 2.5



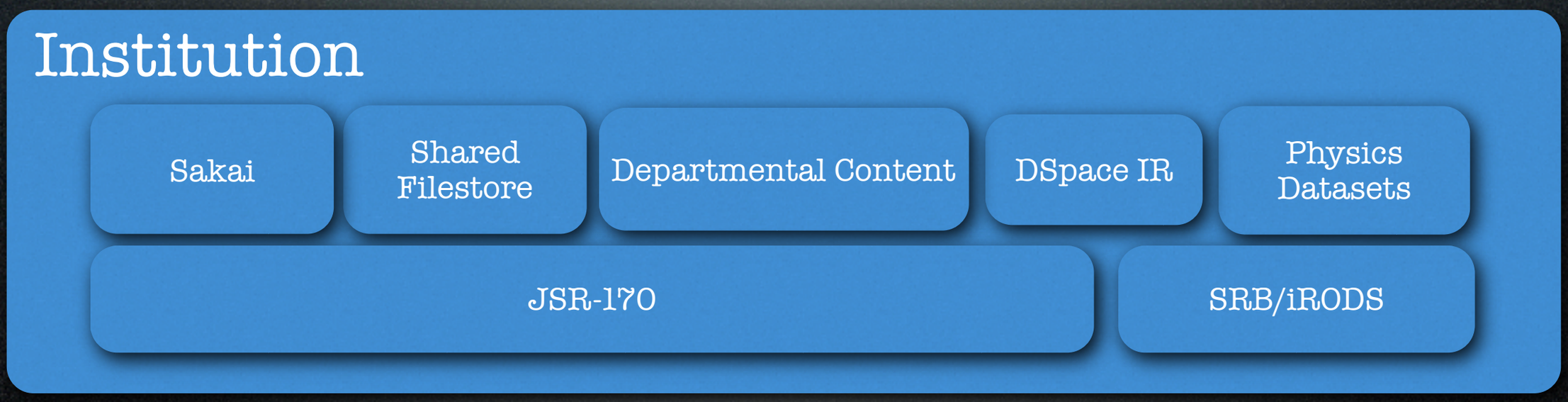
And after 2.5 we can start to move the content centered tools onto the JCR repository. Wiki didn't originally use CHS because it had no versioning, this may be migrated. We can build many views into content with the RSF Stack directly onto JSR-170, but the 170 API is so easy to use it would be quick to work with any view technology.

Current Technical Structure



Previously we existed as an content island within the institution. Unable to particiapate in existing Content Management processes.

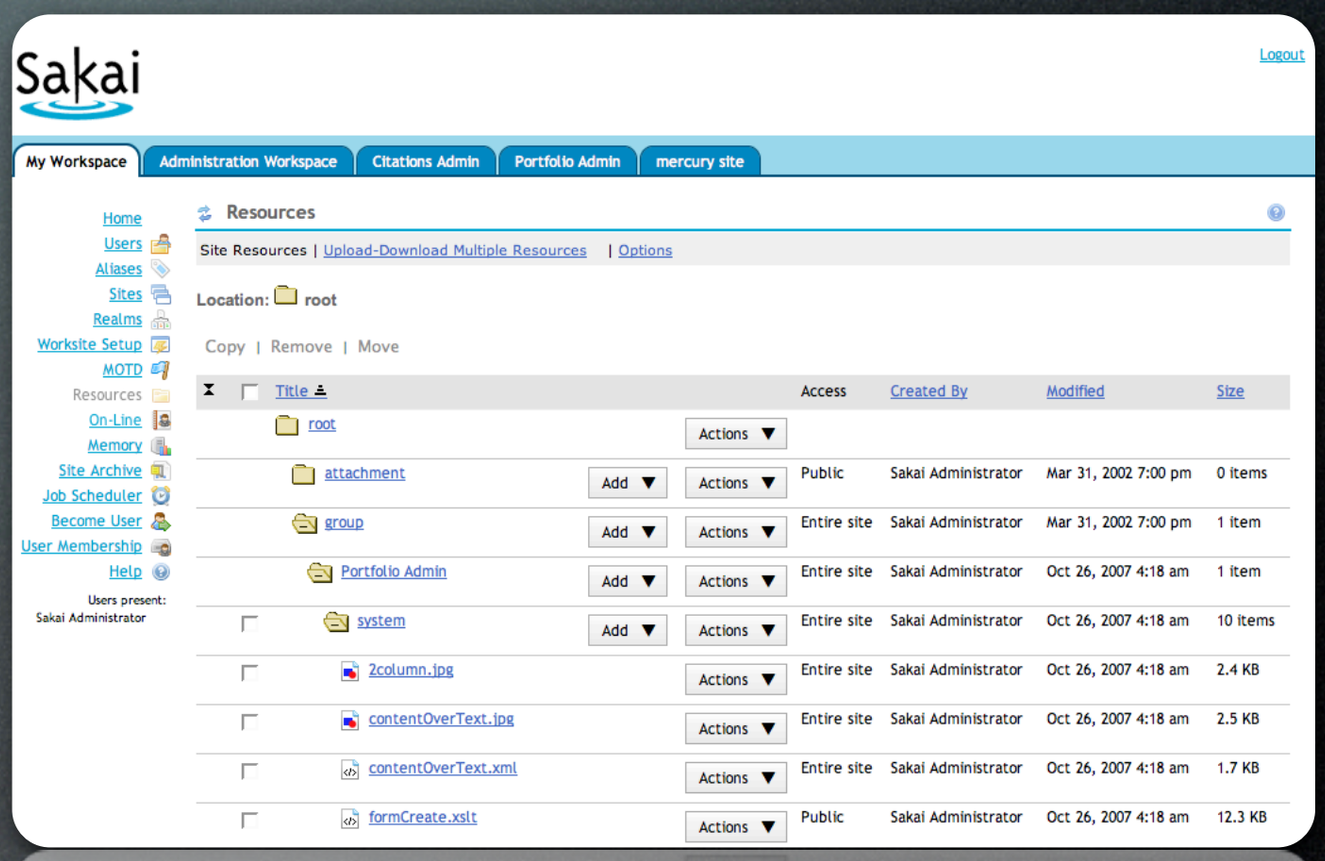
Campus JCR



But where there is shared content infrastructure strategy, Sakai can integrate. MIT is evaluating Alfresco. New York University is a Xythos customer. Cambridge is, as always in chaos, but we will be deploying Jackrabbit under Sakai, and perhaps the wider University will adopt a uniform strategy in this space.

Today - Jackrabbit

- JCRService as Jackrabbit
- Cluster aware
- Interst ?
  - Cambridge (Jan production)
  - Indiana (evaluating)
  - Michigan (testing)

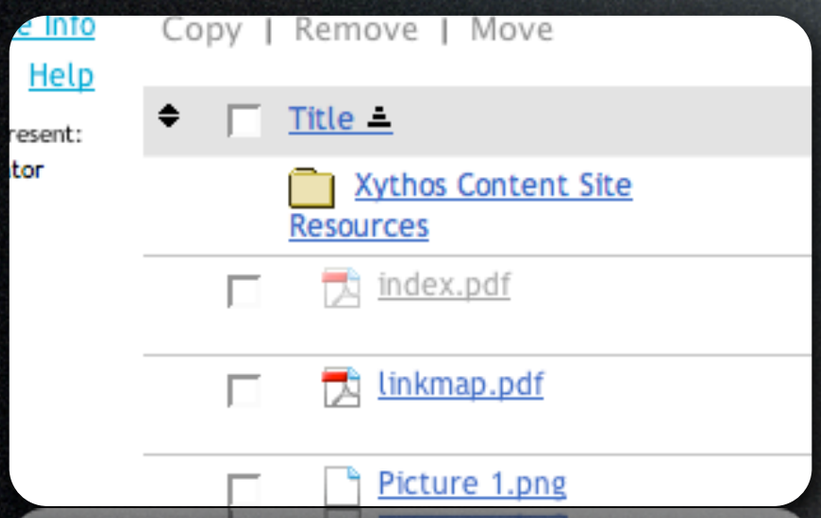


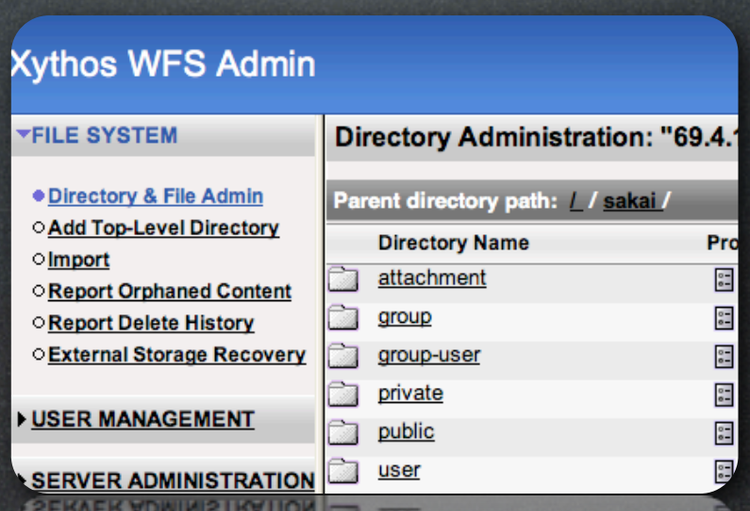
And today, Resources with Jackrabbit..... spot the difference ? None

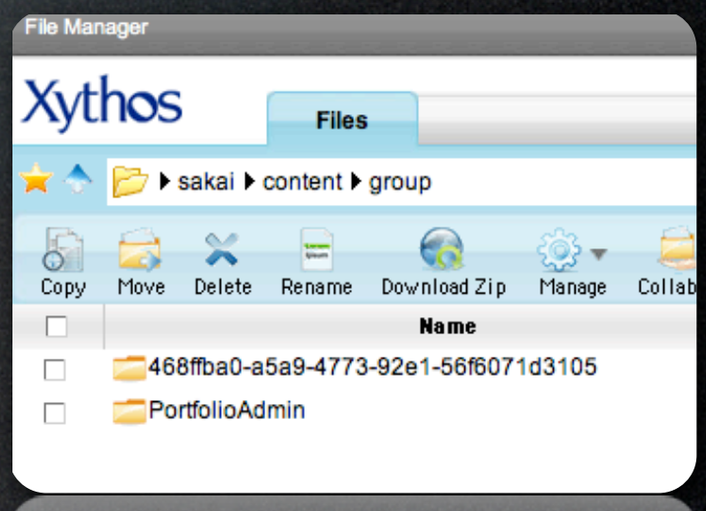
The JCRService for jarckrabbit is about 9 months old. I helped fix a number of Cluster Bugs in Jackrabbit 1.3 in april, so its now cluster aware. I think we will be in production in january. Indiana are interested in evaluating. Michigan have offered help load testing with their data.

Today - Sakai on Xythos

- JCRService using Xythos JSR-170 Beta
- Interst ?
- Xythos :), NYU (Testing), Educause 2007







Screenshots from Educause 2007 Sakai & Xythos Demo Server

And Xythos.

Sakai on Xythos was demoed at Educause 2007. I havent had a chance to talk to Kevin Wiggen their CTO since, but he was excited before.
NYU have offered to test.

Repository Integration

So we have looked at the content creation and use within Sakai. We have looked at how sakai will make use of JCR. But the IR's are still disconnected.

Other Content Repositories

- Preservation
- Networked Filesystems
- Other content stores

There are content stores outside of JCR.
Ones focused on long term preservation. Content for the next 100 years.
There are networked file systems supporting distributed daily computing.
None of this should come under the remit of a Sakai instance, but a Sakai instance should be able to provide collaborative access to these repositories.

Preservation

- Teaching and Learning content
  - Key lecture series
- Research
  - Reuse of Datasets
  - Papers, Notes, Publications

On the preservation front, we may want to capture valuable key lectures.
The next Nobel Prize winner.
Research Datasets are often time consuming to collect and have reuse value, especially in social science.
Then there are the outputs, although there is tension between the academic publishers and the libraries, partially due to inflation we have seen in recent years.

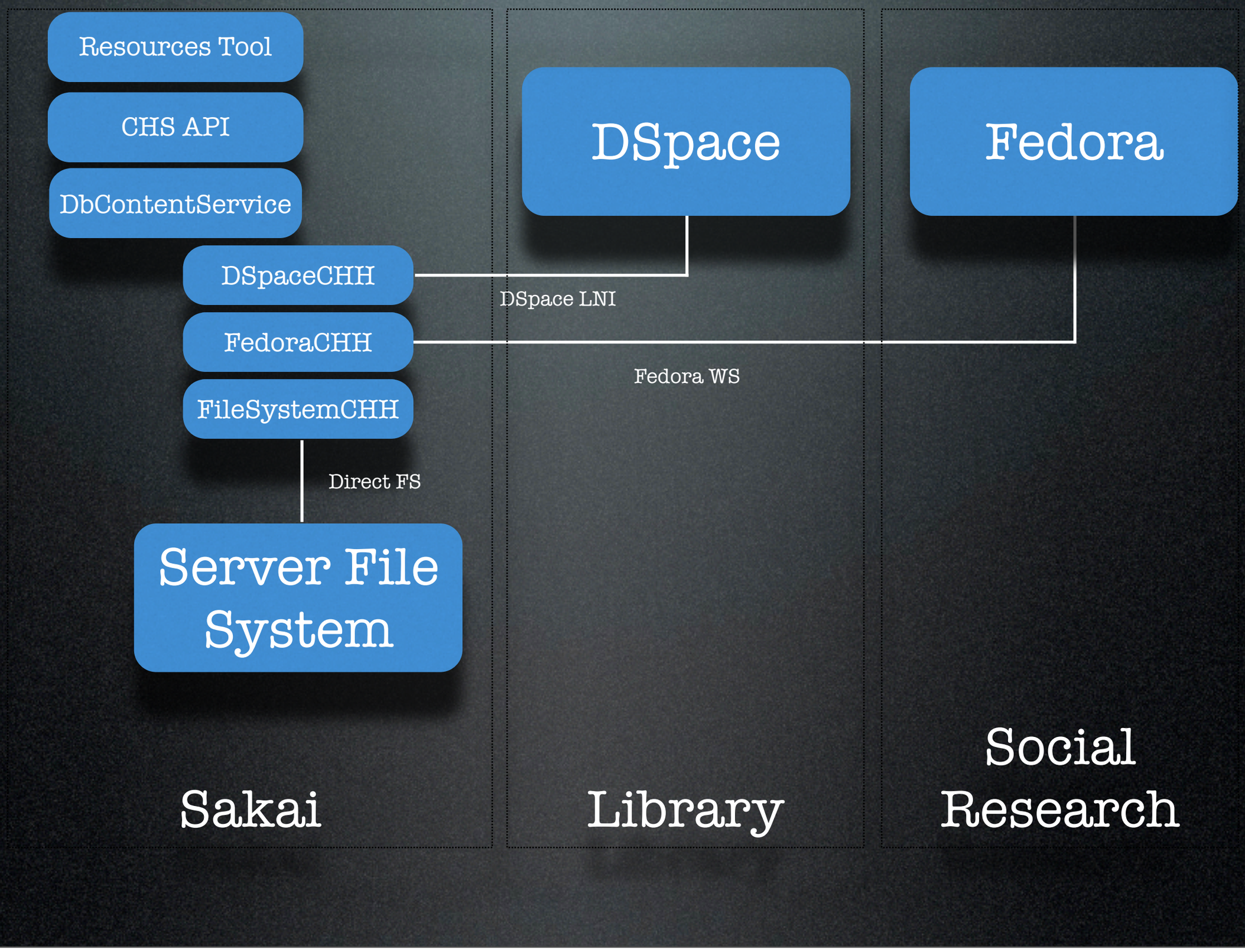
Working Content

- Live Grid DataSets
- Networked File Store

In the non preservation space, networked file systems and grid storage predominate.

Sakai would not want to absorb this content, but it should make it possible to interact with it in more than a read only way.

Integration With Sakai



To integrate these sources with sakai, we have created a Content Hosting Handler.

Simple API that enables content from other repositories to be mounted within Sakai.
We have read write connectors to DSpace, Fedora and the straight forward Filesystems.

Sakai talks to DSpace and Fedora over the network using WebDAV like protocols, and to any filesystem the server can mount, with direct file access.

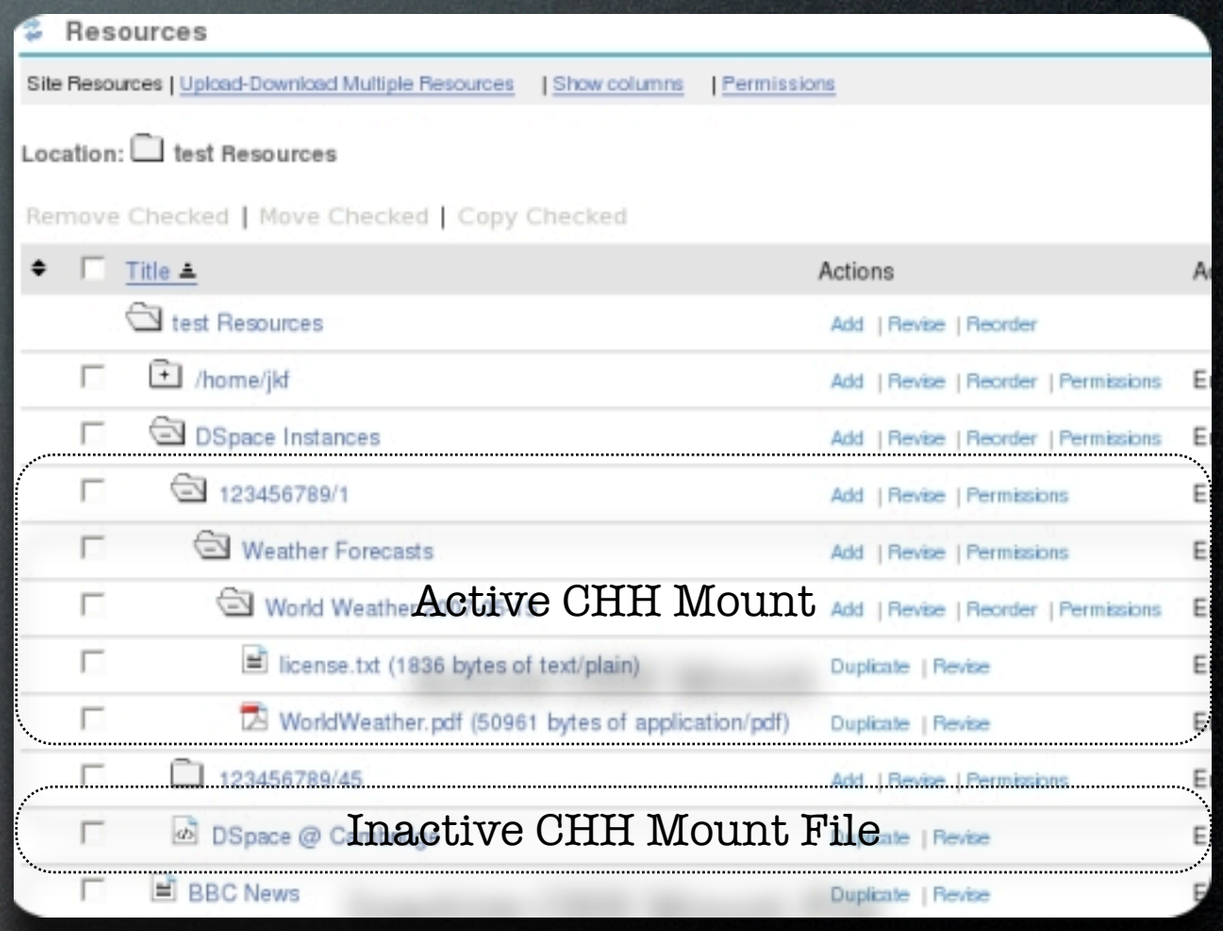
Mounting DSpace in Sakai

- Upload a Mount file

- <mountpoint endpoint="http://johnf%40caret.cam.ac.uk:password@dspace.lib.cam.ac.uk/dspace-lni/lni/DSpaceLNI" baseHandle="123456789/45" searchable="false"/>

- Associate it with a Content Hosting Handler

- The Mount file becomes a collection



This will make more sense with and example.

To make a DSpace collection available inside Resources, we upload an XML mount file into Resources and associate it with a Content Hosting Handler.

Once associated, the Content Hosting Handler turns the real resource into a virtual collection, that acts just like the real sakai content tree.

So you can copy or upload files directly from Sakai into DSpace.
We can do the same for Fedora and any File system that the Server that Sakai is running on can mount on its filesystem.

Ingest Volume and Quality

- Institutional Repositories validate meta-data
- CHH increases volume
- Content Management, Process and Policy

This generates problems. IR's have ingest workflows that ensure the quality of the ingested items. The introduction of a CHH connector, rapidly fills that queue. The CHH connector needs to be aware of the IR policies and ideally enforce or at least inform the users of those policies.

Cambridge Tetra Repositories Enhancement Project (CTREP)

"The CTREP Project Aims to deliver a stepwise increase in the use of DSpace@Cambridge within Cambridge Departments and Faculty by integrating with CamTools. The increased deposit and reuse activity enabled by integration with CamTools will be accompanied by a programme evaluation of use, policy definition and automation within the scope of that link to reduce the administrative load associated with this increase in use. In parallel, as part of the project, the University of Highlands and Islands Millennium Institute, UHI, will be running a pilot integration between Sakai and Fedora using the same framework developed for the Sakai/CamTools DSpace integration. The project is collaborating and being advised by the Digital Libraries research group at MIT in the field of policy development and expression."

The CTREP project at cambridge is investigating this space by deploying the DSpace CHH connector into production, connecting it to DSpace@Cambridge and evaluating the increase in load on the Libraries.

We are working with the PLEDGE project at MIT libraries to take the policy expression language developed in conjunction with CSAIL and the SRB/iRODS work at San Diego.

Content Management and Sakai

- Content is Vital to Sakai
- Sakai migrate to use JSR-170/283
- We are investigating the impact of integration with Institutional Repositories

To recap,

Content is the life blood of Sakai, and most tools deal in content.
Since JCR is a viable standard that has many of the features we need, Sakai will use it.

And Sakai is not an island within the institution, it needs to integrate, however we are watching the impact of that integration

Thank you and Questions

SO, that is Content Management in the Sakai context. Any questions.