

Storing and annotating data in ^{my}Grid

Document Ref:	Storing and annotating data in ^{my} Grid
Author:	Chris Wroe
Institution:	University of Manchester
Date:	September 12, 2003
Pages:	5

Contents

1	Introduction	3
2	Biocore	3
3	Apache Slide	3
4	Questions	4

1 Introduction

The mIR has been instrumental in crystalizing ideas about what we need to store in myGrid how we need to present that repository and the role of annotation.

Several points/issues have become clear:

- The most user friendly way to present the contents is as a filesystem with most entries (if not all) made up of documents. At the moment responsibility for reconstructing the file system view rests with the client in this case the netbeans platform.
- Provision for annotating documents with extra metadata is a major part of the mIR schema. In fact you may want to annotate external resources held in some other repository. Saying that we/I haven't built any annotation clients.
- Need some mechanism for multiuser working with access control lists (which is a body of work) if we are to say it is a personalised yet collaborative environment.
- There are issues about how mutable documents can be. Are documents only ever created by import or workflow? Do we need to support collaborative editing and versioning?
- Is it best to build a single physical repository for an organisation or to provide a federation layer in which an organisation can aggregate multiple existing and future repositories?

Have any other projects come across these issues and have developed solutions?

2 Biocore

BioCore ¹ has built the BioFS :

”a database-managed filesystem that resides on the BioCoRE server. It is accessible via the web and is shared among everyone in a project. A user can work with files located in the BioFS without worrying about where the files are physically located. You can click on a PDB file and BioCoRE will automatically move the file to your machine and let you view the file using JMV or other molecular viewers.”

BioFS relies on the WebDAV protocol for transferring files. WebDAV stands for ”Web-based Distributed Authoring and Versioning”. It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers. WebDAV support is built into both Windows and Linux so the remote file system can be mounted on your desktop.

Although promising BioCore doesn't seem to deal much with annotation or with distributed heterogenous resources.

3 Apache Slide

The Apache Slide project ² does try to at least deal with heterogeneity delivering an abstracted filesystem to the client via WebDAV:

¹<http://www.ks.uiuc.edu/Research/biocore/>

²<http://jakarta.apache.org/slide/index.html>

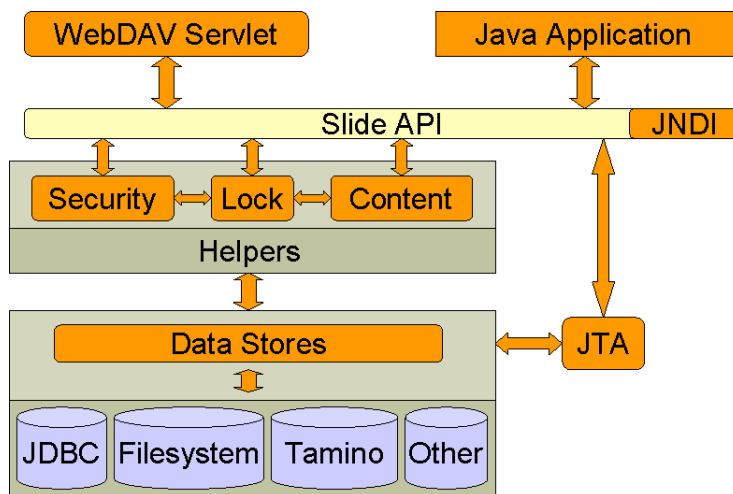


Figure 1: Slide internal architecture (<http://jakarta.apache.org/slide/architecture.html>)

”Conceptually, it provides a hierarchical organization of binary content which can be stored into arbitrary, heterogenous, distributed data stores. In addition, Slide integrates security, locking, versioning, as well as many other services. It can integrate and manage data stored within external repositories, requiring only small abstraction layers to be written for each repository. That way, Slide can integrate the data from various physical locations in a hierachical and unified way. Slide uses can range from managing intranet application content (such as the Exolab.org portal) to using it as a file server.”

The project itself seems a little under documented but its architecture looks interesting (if architectures can be interesting, see figure 1).

Although webDAV includes provision for property based metadata attached to each resource, it is not as powerful as RDF. Researchers within HP Labs (now defunct) middleware division developed and implemented an architecture which coupled Slide and RDF based metadata into a content management system (in their case for what they called Rich Media). [Maron and Kiner, 2003] Their architecture also included Lucene for content indexing (see figure 2).³

4 Questions

So I have more questions than answers around content management and metadata in myGrid:

- Should we cater for multiple legacy repositories and provide a federation layer to provide a single view to the user. e.g. over GIMS⁴ and a local mIR
- Would distributed query processing be useful in driving such a federation?

³<http://jakarta.apache.org/lucene/docs/index.html>

⁴<http://www.cs.man.ac.uk/img/gims/>

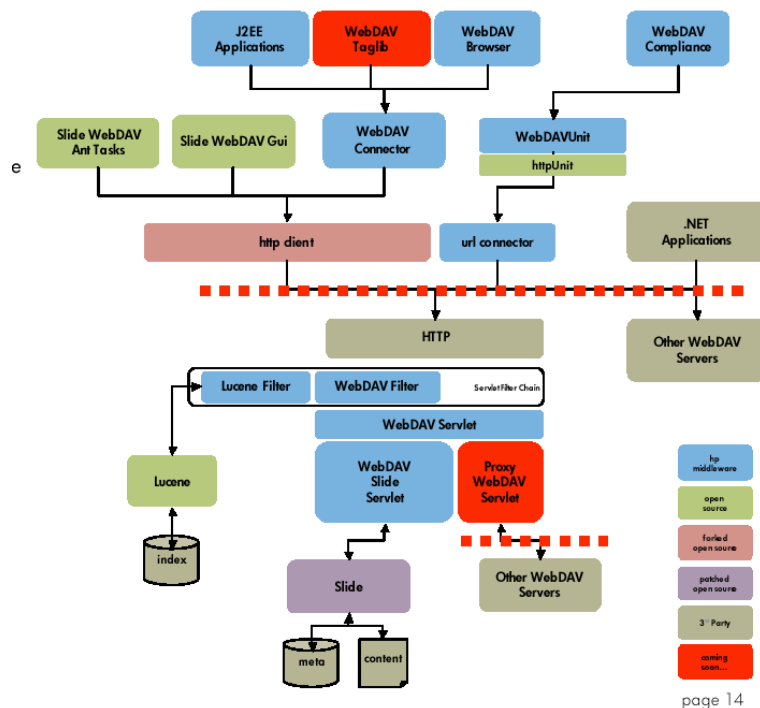


Figure 2: HP Rich Media Architecture

- Could we build upon Slide middleware in any way, rather than rely on the net-beans application to provide a file system view?
- Is it worth investigating the use of the webDAV protocol given its popularity and integration into desktop environments?
- Where exactly should the RDF annotation sit and are we going to build any annotation clients?
- Do we need to deal with content indexing of free text and XML documents?
- How much of a bespoke content management infrastructure do we need to implement? Are there any issues specific to ^{my}Grid either from the user or computer science viewpoint or could we just take someone else's?

References

[Maron and Kiner, 2003] Maron, J. and Kiner, J. (2003). Create rich media applications. http://www.fawcette.com/javapro/2003_01/magazine/features/jmaron/default_pf.asp.