

EPRINTS AND THE SUN STORAGETEK™ 5800 SYSTEM

A Persistent, Scalable and Interoperable Solution

White Paper

November 2007

Table of Contents

Introduction to repositories	3
An Overview of the Architecture	4
An In-Depth Look at the Implementation of the Architecture.....	7
Future planned extensions to EPrints	9
Additional information and references	9

Introduction to repositories

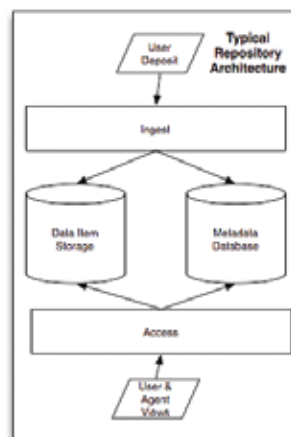
Higher education and research institutions are knowledge-centric, creating and disseminating information as part of their core business activities. Research (discovering new knowledge) and education (transferring knowledge) both require a sound information infrastructure, but in the past it is the administrative and business databases and operational websites that more frequently attract well-managed solutions. The curation of the intellectual output and activities of a university's faculty, the collected works of their departments and schools and the scholarly outputs of the institution as a whole have been dealt with in an ad-hoc way for short term benefits, if at all.

Repositories are a technology that occupy this niche, providing a solution to problems that universities and research institutions have hitherto ignored

- Short term accessibility of their research outputs, scholarly collections and teaching materials
- Long term preservation of their information assets
- Management, assessment and marketing of their activities

Repositories enable public, persistent digital collections of research, scholarship and teaching materials. Repositories encourage both institutional and personal responsibility for managing intellectual assets and facilitate a commitment to long-term knowledge curation.

At their most basic, digital repositories are a place to put things where they can be managed, used and reused for many purposes, by many stakeholders either internal to the institution, external collaborators, the academic and scientific community or the general public. This core functionality is captured by the OAIS reference model, which describes the high level operation of a repository focusing on ingest (the act of depositing an item in a repository), archival processes (the various acts of curation, maintenance and preservation) and access (the ability to use and reuse the material in various forms and formats).



The core infrastructure of a typical repository is concerned with archival processes applied to a database of content, the heart of the repository. This is mainly concerned with storing the data of the deposit (the article, video or database that the user ‘uploaded’ or ‘ingested’) and the metadata about the deposited data — the bibliographic and descriptive information that accompanied the deposit and contains the knowledge about how the item has already been published, certified, exhibited, reviewed, tagged or otherwise exposed to scrutiny.

In this capacity, the repository basically functions as a persistent object store – combining data streams and metadata descriptors. The repository has to simulate this functionality using a standard relational database, storage components and business logic middleware.

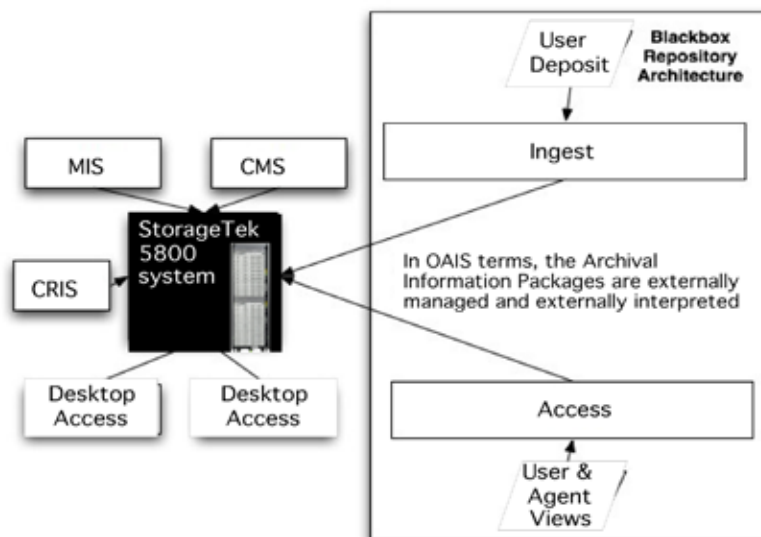
An Overview of the Architecture

The Sun StorageTek™ 5800 system, the first commercially available fixed content storage system, attempts to provide a significant part of this repository functionality as a black box. It is an autonomous, managed, persistent object and metadata store whose capacity scales to hundreds of terabytes.

At the most superficial level, simply having easy access to that quantity of storage can revolutionise the use to which repositories can be put – High Definition video, large collections of high resolution images, automated experimental data collection activities that span years and decades – diverse activities of an institution’s research community can all be accommodated within the institutional repository without recourse to external data storage services.

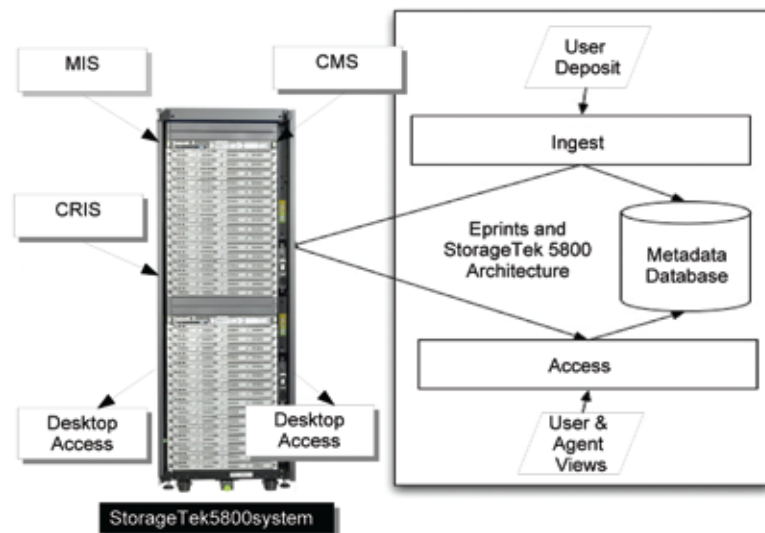
At a more profound level, the adoption of such an object store can help turn the repository into a kind of ‘thin client’, relieving it of the responsibility of simulating a persistent object store.

The following architecture diagram shows the repository middleware and storage functionality (providing a persistent data store and a searchable database of metadata fields) taken on by the storage product.



Taken to its logical extreme, the core of a repository could consist of a StorageTek 5800 system as the backend storage device with the repository application logic (API, policy implementation and user interaction) provided by any of the public, open source platforms such as EPrints. In fact, using the OAIS repository model and adopting a standard core Archival Information Package format, a repository could be built whose front end changed between EPrints and DSpace, or one that had multiple separate and simultaneous personalities. The advantages of such a system would be profound — the all-important data and metadata housed in the repository would not be dependent on the software that implemented or emulated the repository functionality. Access to the repository contents could be achieved directly by other applications through the StorageTek 5800 system interfaces — for example a scientist's data files could be available on a workstation desktop through WebDAV.

While this remains the ultimate goal of repository integration, the current implementation is less ambitious than this. The metadata is still managed within the repository system itself, but the data storage is handed over to the external storage device, the StorageTek 5800 system. This provides advantages for the repository in terms of scalable and autonomous storage management, making the repository an effective vehicle for all classes of scientific and scholarly material, while retaining a close connection with the metadata that affords the core suite of repository functionality.



Every data item that is stored in the StorageTek 5800 system has an Object Identifier (OID) and a set of metadata associated (based on a user-definable schema). The combination of data and metadata forms an object. The StorageTek 5800 system can retrieve objects by OID or by metadata query, but the repository does not store the OIDs that correspond to its data items, instead it locates them by a query for their persistent URL. Effectively the EPrints' web server rules are augmented to look for any missing data items by searching for the requested URL as a piece of metadata in the object store.

When the repository tries to deliver the bitstream from the filesystem and finds that it is missing, the EPrints web server subsystem (Apache) catches the error and simply tries to read the missing data from the StorageTek 5800 system device instead (by looking up the requested URL). This "look-aside" approach will work equally well on any web-based information system, not just an EPrints repository.

To enable external applications to make maximum use of the stored objects, a copy of the repository bibliographic metadata is stored in the StorageTek 5800 system as part of the externalized object.

This stage of integration represents a convincing increase in the usefulness of a repository and helps to move repositories from the "cottage industry" status (low quantities of home made goods all managed by hand) to an "industrial" footing (large quantities of machine-generated data managed autonomously). As desirable as this is, it is only the first step on the path increasing synergy between repository applications and fixed content storage system storage solutions.

An In-Depth Look at the Implementation of the Architecture

The instructions in this section describe an outline for installing an EPrints Repository so that it uses a Sun StorageTek 5800 system as an autonomous preservation storage solution. The intention is to help understand the process and the underlying architecture. Specific instructions and support for the installation can be found at <http://wiki.eprints.org/w/SunStorageTek> . Please note that these instructions are valid as at the product launch in November 2007 (using EPrints v3.03 and StorageTek firmware v1.1) but will be updated (and simplified) for subsequent versions of EPrints.

- 1 .First, create an EPrints repository as normal
 - a) Install EPrints on whatever host platform is being used, according to the normal instructions found on http://wiki.eprints.org/w/EPrints_Manual
 - b) Configure and populate your repository

Note that it is possible to add a StorageTek 5800 system unit to a live repository, although it is advisable to gain some experience with a test repository first.

2. Set up the StorageTek 5800 system
 - a) Install and set up the StorageTek 5800 system as normal. Refer to <http://docs.sun.com> and search for 5800. EPrints does not require exclusive use of the StorageTek 5800 system and no specific firewall or security options are imposed.
 - b) Add the EPrints schema, which currently consists of only two items of metadata
 - eprints.uri – The public URI of this data file as defined by the repository. This field is the target of a simple query that is performed by EPrints whenever it needs to locate a data file that has been transferred to the StorageTek 5800 system. No OIDs are ever stored by EPrints.
 - eprints.xml – The entire metadata of the eprint record that this data file is associated with. The metadata is stored in EP3 XML format. This metadata is currently unused by EPrints, but will enable files to be identified for independent auditing purposes.
3. In addition to these items, EPrints also makes use of the filesystem.mimetype field in the standard schema to augment the Apache rules for content type declaration based on file name extensions.

The Perl module includes the following methods and fields:

<code>new (host,port)</code>	Creates a new connection to the StorageTek 5800 system at the given host address and port number.
<code>query(q)</code>	Returns the OIDs of any items that satisfy the query on the StorageTek 5800 system. An example of a typical query is <code>eprints.uri = 'http://myrepository.org/1234/01/document.pdf'</code> Note that such a query might return 0, 1 or many OIDs, although only the case in which 1 OID is returned indicates a successful query.
<code>get_metadata(oid)</code>	Retrieves the complete set of metadata associated with an OID as hash.
<code>get_oid(oid,sub)</code>	Retrieves the data associated with an OID and streams it to the subroutine passed as a parameter. The Perl expression <code>sub { print \$_[0]; }</code> is an example of a subroutine that will print out the data from the StorageTek object one chunk at a time.
<code>store(fh,hash)</code>	Stores the data accessible through the file handle into a new StorageTek object with metadata given in the hash. The method returns an OID, but this isn't used by EPrints. e.g. <code>store(STDIN,{ "eprints.uri" => "http://a.org/1234/foo.txt", "filesystem.mimetype" => "text/plain" });</code>
<code>error</code>	A boolean indicating the existence of an error condition
<code>error_string</code>	A string explaining the error condition

4. Install the EPrints StorageTek 5800 system package, which consists of the following components
 - a. Apache Rewrite and StorageTek 5800 system look-aside handlers that attempt to serve missing documents from the StorageTek 5800 system instead of the local filesystem. These components are written in Perl and use the Perl-wrapped API (above).
 - b. EPrints administrator's command-line script ('reaper') that identifies the repository data files that are suitable for moving to the StorageTek 5800 system, copies them across and deletes them from the local file system. This script should enact local policies on data mobility and preservation, but currently moves all data files from eprint records that appear in the live repository. For technical reasons, all ingest procedures should have been completed before the data is moved as some processes (e.g. thumbnail creation or virus checking) assume the presence of the file on a local file system.
5. Add EPrints configuration file for StorageTek 5800 system Virtual IP (VIP) address and port number.

Future planned extensions to EPrints

To support the deposit of ultra-large items, the EPrints ingest process will be rewritten so that data can be directly uploaded to the StorageTek 5800 system object storage without being cached on an intermediate file system.

Regular backups of the repository internal state (eprints metadata, user data, repository history etc) will be made to the StorageTek to enable automatic repository backups.

Additional information and references

- i. Open Archival Information System (OAIS):
<http://nost.gsfc.nasa.gov/isoas/>
- ii. EPrints website:
<http://www.eprints.org/>
- iii. Sun StorageTek 5800 system
sun.com/storagetek/disk_systems/enterprise/5800/index.xml
- iv. OpenSolaris – StorageTek 5800 system
<http://www.opensolaris.org/os/project/honeycomb/>
- v. OpenSolaris
<http://www.opensolaris.org/os/>