

Akubra Intro and Update

Below are the notes I used for the Akubra update in today's architecture meeting.

What's Akubra About?

- A standard Java interface for reading/writing files, but at a different level of abstraction than a filesystem
- Transactional by design (but implementations may ignore transaction semantics)
- Exploring web-based exposure
- From the Akubra wiki: [Requirements and Goals](#)

Note: The Akubra wiki is hosted at <http://topazproject.org/akubra>

Anyone is welcome to sign up to the dev and general mailing lists.

Filesystem vs. BlobStore

Common filesystems:

- Have directories
- Can provide system metadata about files (e.g. size, modified date)
- Allow partial reads and writes of files

An Akubra BlobStore:

- Has a collection of URI-addressable bitstreams (no "directories")
- Only provides the size of each file -- is not concerned with other system metadata (yet?).
- May allow partial reads (InputStreams can skip()...)
- Does not allow partial writes

Java API

This is in flux. We are currently testing the design with a simple filesystem implementation.

[Blob](#) (A finite, readable bitstream)

[BlobStore](#) (For getting connections)

[BlobStoreConnection](#) (For CRUD operations)

Transactions

Level of support varies per-implementation (some can "fake it")

Why: To execute a mixed set of CRUD operations of several files as one atomic unit of work.

Observation: We can build a transactional blob store on top of a non-transactional one...with the help of a DB.

Example non-transactional BlobStore: FSBlobStore (see [FSBlobStoreConnection](#))

Higher-level BlobStore TBD:

- Uses FSBlobStore to persist data
- Uses database to support transactions (via id mapping)

Other possible storage Plug-Ins:

- S3 (anything based on current LLStore should be easy to port over)
- ZFS (already transactional, does not need layering)
- Centera (content-addressable...ids not available till content is written)
- Sam/QFS (hierarchical storage implies graceful handling of delays...not a use case we've factored in yet)

Web-based exposure?

- Opens up use of akubra-java impls to other (remotely-running) programs
- Allows an akubra impl that's a client to remote akubra instance
- Lots of interesting possibilities!