Re-thinking Fedora's Storage Layer..

- Main wiki page: https://wiki.duraspace.org/x/syTS
  - Documents
    - OR '10 extended abstract
    - These presentation slides
    - Also contains original proposal and presentations from March 2010 London meeting
  - Issues for Discussion

Aaron Birkland, Cornell University USA (birkland@cs.cornell.edu)
Asger Askov Blekinge, State & University Library Aarhus, Denmark

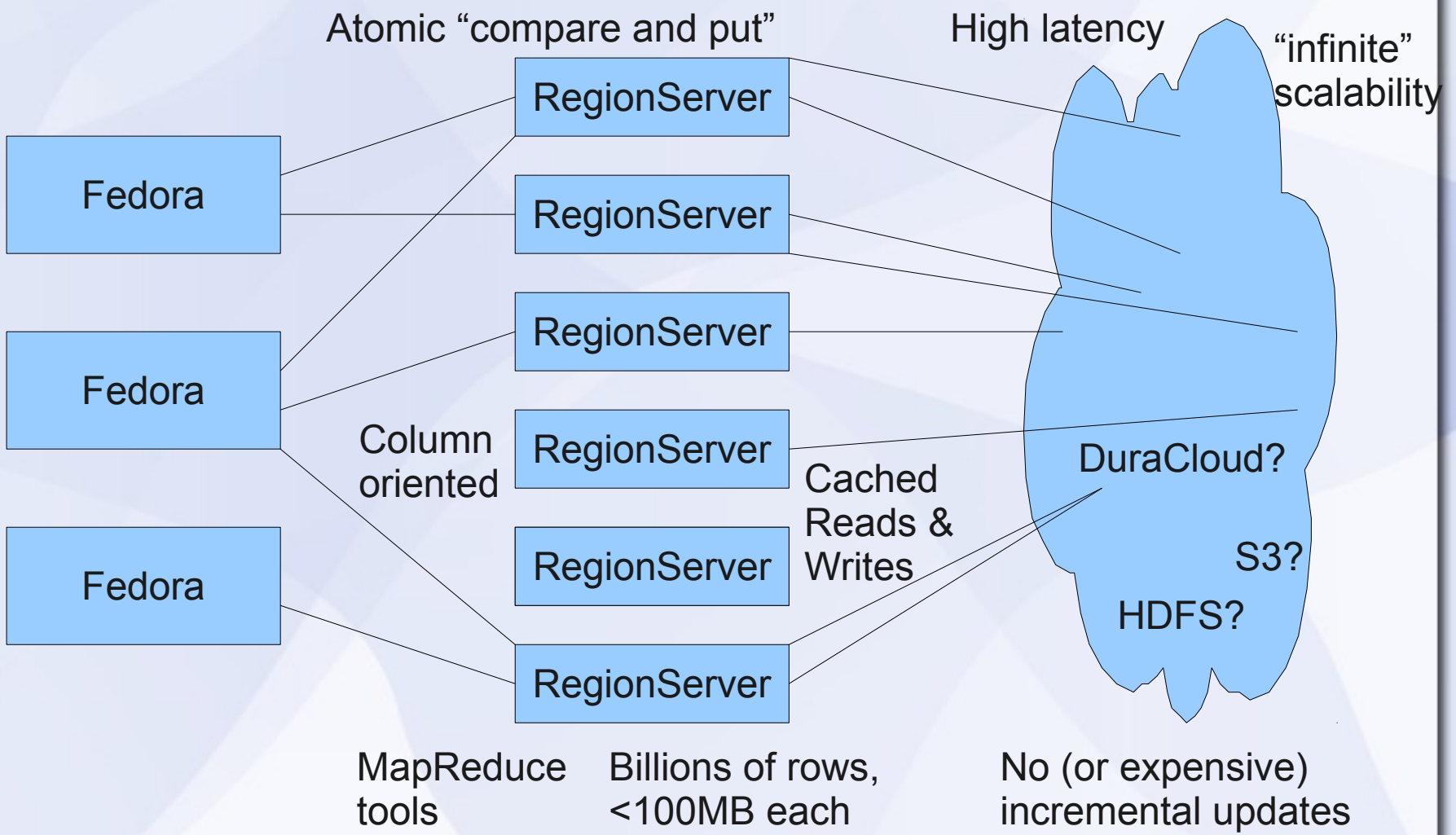
About this presentation..

- Overview high-level storage concept and motivation
- Identify potential applications, assumptions, and risks
- Request for feedback and participation
  - This is the most important outcome!

Motivation: Thought experiment

- What prevents Fedora from scaling horizontally? (multiple servers form a single 'repository')
- ... storing different kinds of data in different storage location/devices through its own API? (e.g. based on content model)
- ...preserving data in completely different structures?
  - On-disk zip archives containing foxml + datastreams?

Quick motivating sketch: Scalability using Apache HBase



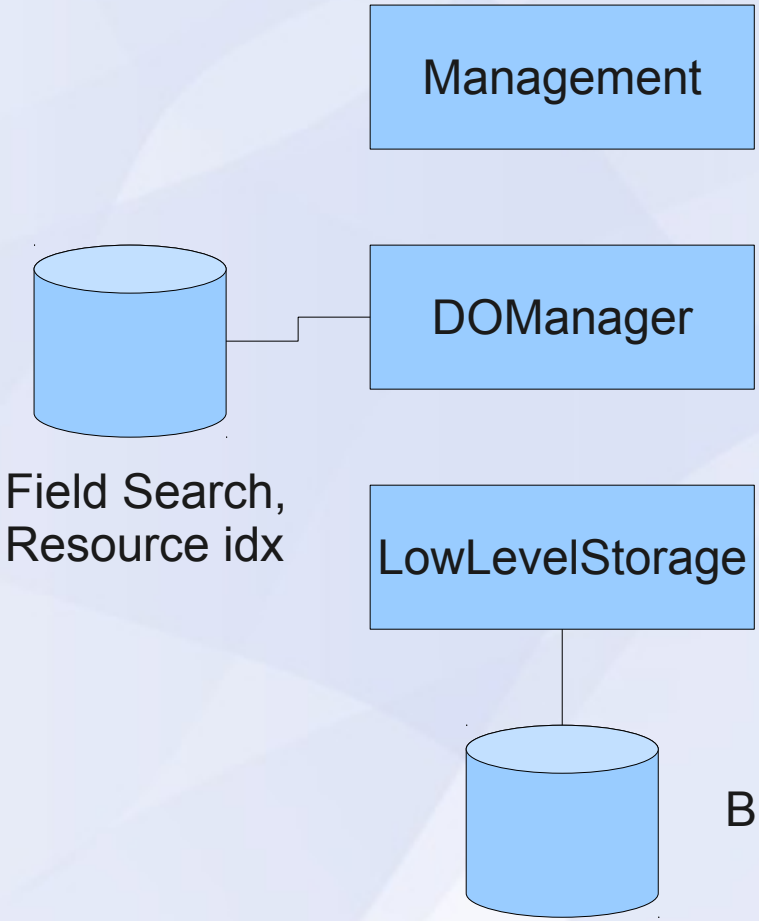
The “problem”

- Forced to store objects as object (foxml) blobs, and separate datastream blobs.
  - Locking, indexing, manipulation logic mostly intertwined.
- Pluggable storage impl would need to introspect on blob content in order to do something intelligent.
  - For datastreams, it does not have much contextual information to work with.

The “fix”

- Remove several hard assumptions within Fedora
  - One particular blob storage paradigm
  - Locking strategies
  - Indexing strategies
- Provide an explicit layer for plug-in, data-oriented services
  - Intelligent storage decisions
  - Data-oriented messaging, policy, caching

Fedora architecture excerpt



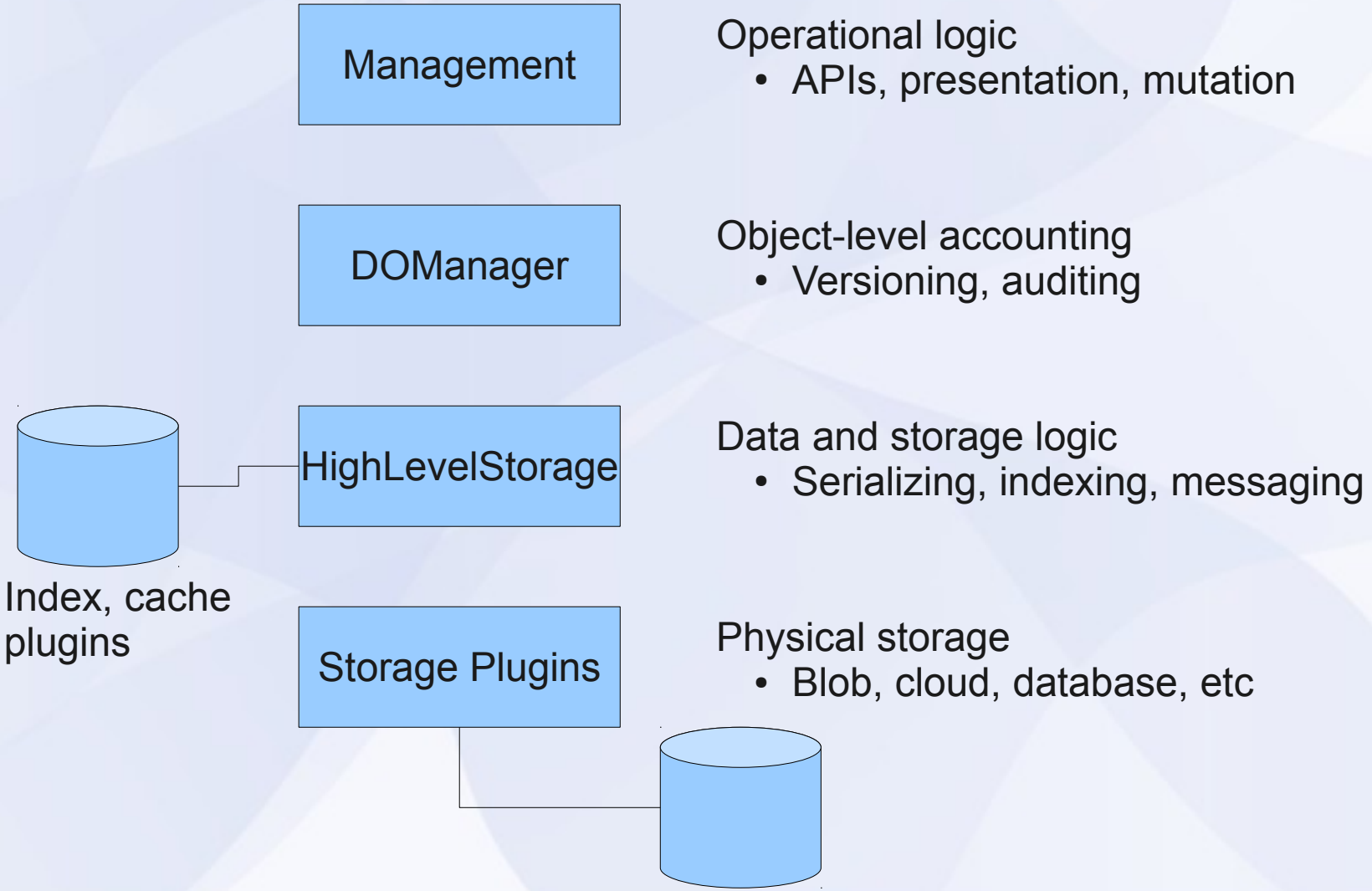
Operational logic
- APIs, presentation, mutation

Object-level accounting
- Versioning, auditing, indexing

Blob storage
- Datastreams, serialized objects

Blob storage plugins, e.g. Akubra

Modified architecture



Interface comparison (abridged)

```
void
addObject(String,
InputStream);

void
addDatastream(String,
InputStream);

void
replaceDatastream(String,
InputStream);

InputStream
retrieveObject(String);
```

```
Result
add(DigitalObject);

Result
update(DigitalObject,
DigitalObject)

Result
remove(DigitalObject)

DigitalObject read(PID);
```

Interface Explanation

Result add(DigitalObject);

Result update(DigitalObject, DigitalObject)

Result remove(DigitalObject)

DigitalObject read(PID)

- DigitalObject – Logical representation of a Fedora object (similar to the one that exists today)
- Result – could contain handle to asynchronous storage workflows

Interface Explanation

Writable

Result add(DigitalObject);

Result update(DigitalObject, DigitalObject)

Result remove(DigitalObject)

Readable

DigitalObject read(PID)

- Could further divide into 'readable' and 'writable' interfaces
- HighLevelStorage plugins would implement one.
- Index, JMS hook could be Writable, cache could be Readable

Implications and risks

- How flexible is too flexible? Foxml and Files can no longer be basic assumptions
  - ... though it should still remain the mainstream, default configuration
- Different technologies will have different preservation characteristics.
- It would seem to encourage reasoning about stored data outside of Fedora
- It will make Fedora even harder to describe

The way forward

- The decision to proceed in this direction needs to be vetted and verified by the community at large.
- Many design decisions still need to be made (see wiki)
- Start small! Use new interfaces to duplicate Fedora's current characteristics
  - high-level storage will merely **allow** new paradigms and methods. Creativity is left as an exercise for the community.

The plan so far

- Special topics meetings (watch the mailing list) to resolve key design decisions.
- Form a panel of interested individuals to assure that progress and decisions are made, and make the final recommendation on whether to proceed with a specific design.
- Have most key decisions made by the end of the year. Final decision at the next committers' summit?

If you are interested, or have something to say

- Make your thoughts/interest known!
  - Developers' mailing list
  - Talk to a committer
  - Comment on wiki page
  - Attend a committer meeting, or a special topic meeting
  - Watch lists for relevant announcements
  - Watch the wiki page (literally: sign in to wiki, go to tools->watch in upper right.)

High-level storage: Resources

- Main wiki page: https://wiki.duraspace.org/x/syTS
  - Documents
    - OR '10 extended abstract
    - These presentation slides
    - Also contains original proposal and presentations from March 2010 London meeting
  - Issues for Discussion