Introduction to Fedora 4 Features

Learning Outcomes

Understand the purpose of a Fedora repository

Understand the core features of the software

What is a Fedora Repository?

Secure software that stores, preserves, and provides access to digital materials

Supports complex semantic relationships between objects inside and outside the repository

Supports millions of objects, both large and small

Capable of interoperating with other applications and services

Fedora:
Specification vs. Implementation

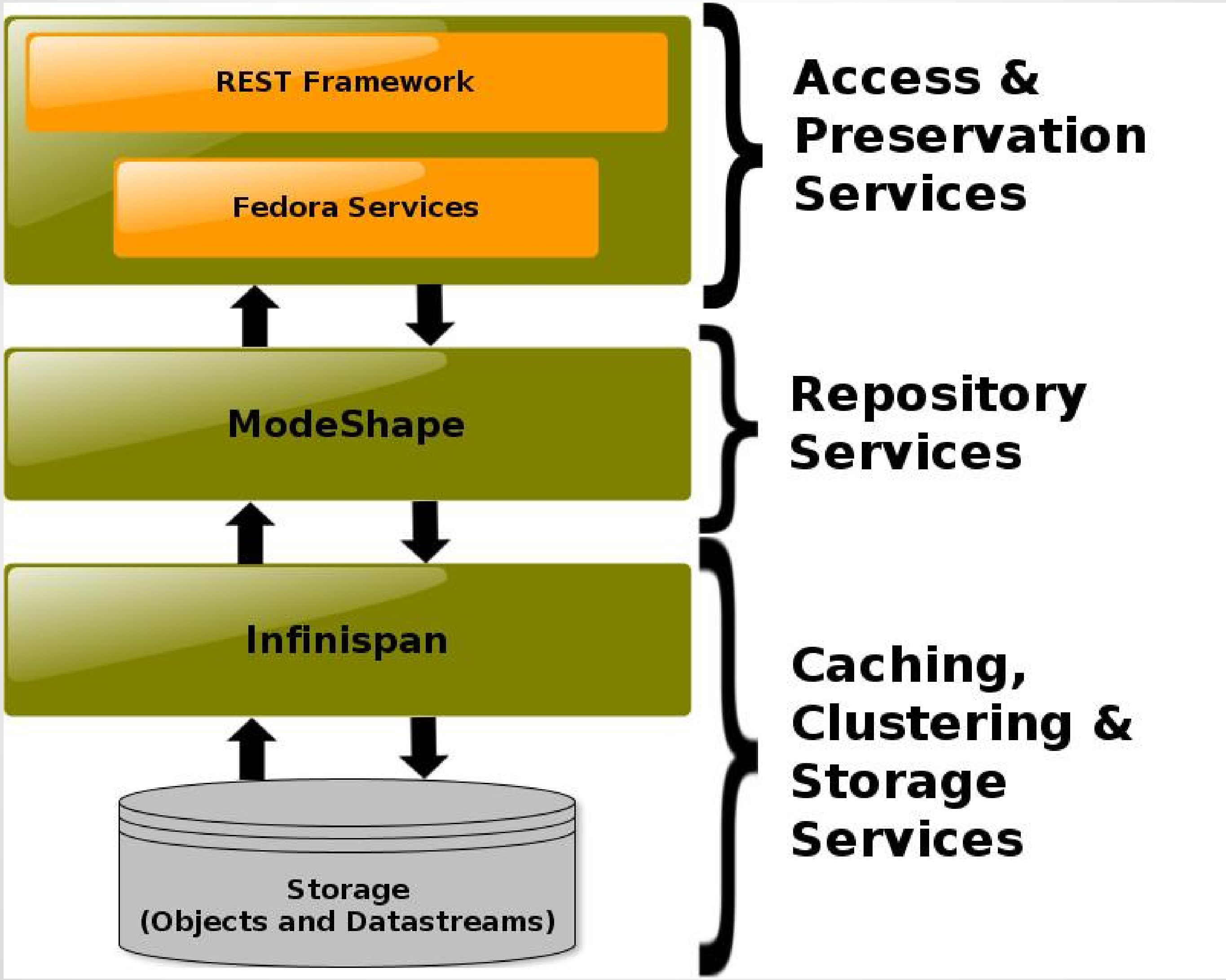
Specification

- Defining Fedora as a set of RESTful services
- Aligning with existing standards
- Enabling use of standard tooling / practices

Implementation

- Reference implementation built over JCR
- Alternate implementations are possible

Reference Implementation: Component Stack


REST Framework
Fedora Services
ModeShape
Infinispan
Storage
(Objects and Datastreams)
Access & Preservation Services
Repository Services
Caching, Clustering & Storage Services

Core Features

Core Features and Standards

CRUD - *Linked Data Platform (LDP)* ✔

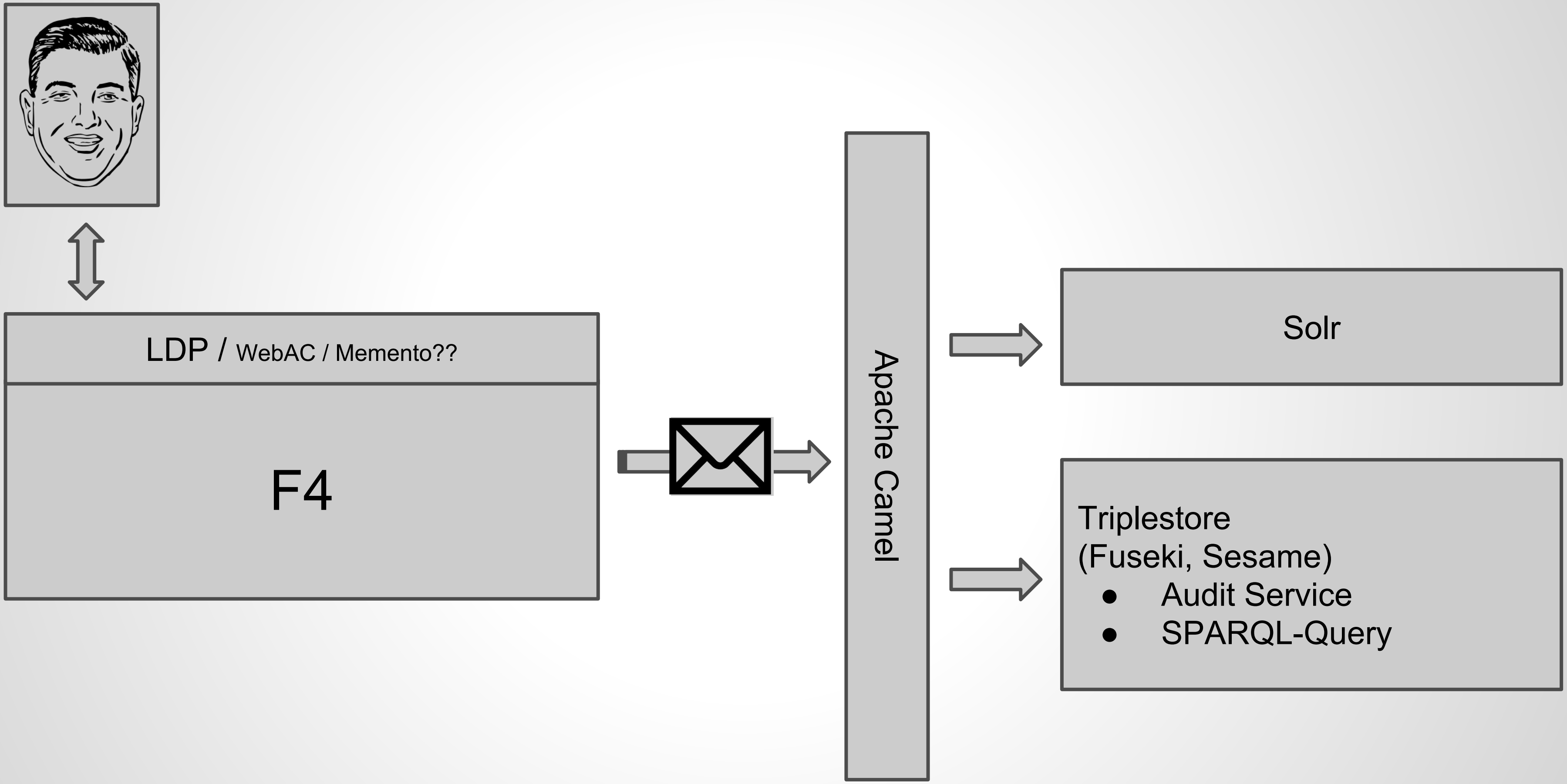
Versioning - *Memento*

Authorization - *WebAC* ✔

Batch Atomic Operations - (*a standard??)*

Fixity - *http://tools.ietf.org/html/rfc3230#section-4.3.2* ✔½

Fedora Vagrant Components



LDP / WebAC / Memento??

F4

Apache Camel

Solr

Triplestore
(Fuseki, Sesame)

- Audit Service
- SPARQL-Query

Hands-on: CRUD

http://localhost:8080/fcrepo/rest/
(fedoraAdmin:secret3)

Create a “cover” Container

PUT vs. POST

...Note: names in demo are only for readability

Make “cover” a pcdm:Object

```
PREFIX pcdm: <http://pcdm.org/models#>

INSERT {
  <http://localhost:8080/fcrepo/rest/cover>
  rdf:type
  pcdm:Object
}
WHERE { }
```

REDUX
Make “cover” a pcdm:Object

```
PREFIX pcdm: <http://pcdm.org/models#>

INSERT { <> a pcdm:Object }
WHERE { }
```

Batch Atomic Operations (BatchOps)

Multiple actions can be bundled together into a single repository event (BatchOps)

BatchOps can be rolled back or committed

Can be used to maintain consistency

Hands-on: BatchOps

Authorization

The authorization framework provides a plug-in point within the repository that calls out to an optional authorization enforcement module

Currently, four authorization implementations exist:

- No-op
- Role-based
- XACML and
- **WebAC**

Hands-on: AuthZ

Create following Containers

- “files”

...contained inside “cover”

- “my-acls”

...at top-level is fine

- “acl”

...contained inside “my-acls”

- “authorization”

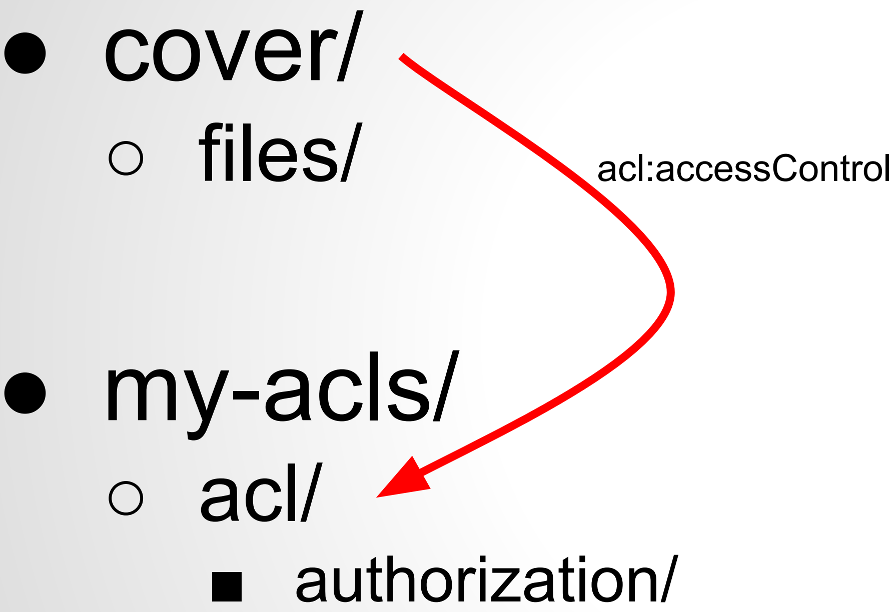
...contained inside “acl”

Final result (structure)

- cover/
  - files/

- my-acls/
  - acl/
    - authorization/

Final result (structure)



- cover/
  - files/
- my-acls/
  - acl/
    - authorization/

“cover” must point to its ACL

- *An ACL must have one or more authorizations*
- *“authorizations” define:*
  - *agent(s)*
  - *mode(s)*
  - *resource(s) or class*

Define the “authorization”

```
PREFIX acl: <http://www.w3.org/ns/auth/acl#>
PREFIX pcdm: <http://pcdm.org/models#>

INSERT {
    <> a acl:Authorization ;
    acl:accessToClass pcdm:Object ;
    acl:mode acl:Read, acl:Write;
    acl:agent "adminuser" .
} WHERE { }
```

Link “acl” to “cover”

-- Update “cover” resource --

```
PREFIX acl: <http://www.w3.org/ns/auth/acl#>

INSERT {
    <> acl:accessControl </fcrepo/rest/my-acls/acl>

} WHERE { }
```

Verify AuthZ

** Warning cURL sighting **

```
curl -i http://localhost:8080/fcrepo/rest/cover
> 401

curl -i -ufedoraAdmin:secret3 http://localhost:8080/fcrepo/rest/cover
> 200

curl -i -uadminuser:password2 http://localhost:8080/fcrepo/rest/cover
> 200

curl -i -utestuser:password1 http://localhost:8080/fcrepo/rest/cover
> 403
```

Versioning

Versions can be created on resources

Previous versions can be restored

Hands-on: Versioning

Create version “v0” of “cover”

Add dc:publisher to “cover”

```
INSERT {
  <> dc:publisher "The Press"
}
WHERE { }
```

Create version “v1” of “cover”

* Inspect and Revert

Hands-on: Fixity

Fixity

Over time, digital objects can become corrupt

Fixity checks help preserve digital objects by verifying their integrity

On ingest, Fedora can verify a user-provided checksum against the calculated value

A checksum can be recalculated and compared at any time via a REST-API request

Create some cover binaries

...contained inside “files”

cover.jpg
cover.tif

* Fixity
* Corrupt and test?

Success!