The SCAPE Repository
Reference Implementation

Matthias Hahn
FIZ Karlsruhe

SCAPE Training Event
British Library, September 16-17th 2013
The SCAPE Repository Reference Implementation

- Introduction
- Integration
  - SCAPE Digital Object Model and SCAPE APIs
  - Demo 1: create sample objects
  - Demo 2: test the Connector API
- Repositories
  - Fedora 3 & Akubra HDFS
  - Fedora 4
  - SCAPE Loader Application
- Demo 3: SCAPE Loader App with Fedora 4 Cluster

This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Introduction – What’s the problem

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

Credit: Todd Lindeman and Brian Vastag/The Washington Post
Introduction - It’s getting worse...

Global Storage Capacity
In Exabytes*

*) 1 Million Terabytes **
**) 1 Million Megabytes

Introduction - Limits

- Large and complex objects
- Large and heterogenous collections
- Lack of automation, QA and tool support
- Systems don’t scale
Introduction - Scalability

- Load Scalability
- Functional Scalability
- Maintenance Scalability
- Geographic Scalability
- Economic Scalability
- Administrative Scalability
Scalable Preservation Environment

- Interfaces to Workflow execution
- Interfaces to Planning and Watch
- Support of Provenance information
- Interfaces to computation cluster
- Load Scalability tests
- Digital Object Model
This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
SCAPE Digital Object Model

- Premis & METS
  - Premis: international standard for metadata to support the preservation of digital objects and ensure their long-term usability.
    - [http://www.loc.gov/standards/premis/](http://www.loc.gov/standards/premis/)
    - We are using Version 2.2
  - METS Metadata Extension & Transmission Standard
    - container format to be able to deal with complex objects
    - [http://www.loc.gov/standards/mets/](http://www.loc.gov/standards/mets/)

This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Building a METS Document

1. Expressing the Structure of complex objects
2. Linking Structure with Content
3. Linking Structure with Descriptive Metadata
4. Linking Structure and Content Files with Administrative metadata

```xml
<METS:mets>
  <METS:metsHdr />
  <METS:dmdSec />  Header
  <METS:amdSec />  Descriptive MD
  <METS:fileSec />  Administrative MD
  <METS:structMap />  File list and groups
  <METS:structMap />  hierarchical structural Map of the object
                       (e.g. pages of book)
  <METS:structLink />  cross links inside the document
  <METS:behaviourSec />  e.g. applications to view content
</METS:mets>
```
SCAPE Digital Objects – SCAPE METS

- **Descriptive Metadata**
  - DC

- **Technical Metadata**
  - AudioMD
  - VideoMD
  - TextMD
  - FITS
  - NISO/MIX

- **Provenence**
  - PREMIS

- **Source**
  - DC

- **Rights**
  - PREMIS RIGHTS

```xml
<mets:mets PROFILE="SCAPE" OBJID="fiz.karlsruhe.09601"
xsi:schemaLocation="http://www.loc.gov/METS/
http://www.loc.gov/standards/mets/mets.xsd">
  <mets:dmdSec ID="DC"></mets:dmdSec>
  <mets:amdSec>
    <mets:techMD ID="object1"></mets:techMD>
    <mets:rightsMD ID="rights1"></mets:rightsMD>
    <mets:sourceMD ID="source1"></mets:sourceMD>
    <mets:digiprovMD ID="event1"></mets:digiprovMD>
    ...
  </mets:amdSec>
  <mets:fileSec></mets:fileSec>
  <mets:structMap></mets:structMap>
</mets:mets>
```
SCAPE Digital Object Model

Document available on Github
https://github.com/openplanets/scape-apis
Serialize / Deserialize SCAPE Digital Objects

- Premis: Intellectual Entity - Representation - File - Bitstream as Java Objects.
- Intellectual Entities get serialized/deserialized from/to the METS format
- Representations, Files, Bitstreams get serialized/deserialized by JAX-B directly.

```
HttpGet get = new HttpGet(SCAPE_URL + "/entity/entity-2");
HttpResponse resp = this.client.execute(get);

IntellectualEntity ie = this.marshaller.deserialize(IntellectualEntity.class,
resp.getEntity().getContent());
```

Code available on Github:
https://github.com/openplanets/scape-platform-datamodel
Platform Data Model – Code Level

- **ScapeMarshaller.java**
  - takes care of serialization/deserialization of Intellectual entities
  - Takes conversion classes to convert your data into SCAPE Digital Object as IntellectualEntity
  - Whole bunch of Junit Tests can be found at [https://github.com/astromatthias/scape-fedora4-test](https://github.com/astromatthias/scape-fedora4-test)
Platform Data Model – Converter

- Conversion into SCAPE Digital Object Model

Add your own converter if you want to convert your METS (theoretically any other format) into SCAPE METS Java Objects.

e.g. covers

- google book
- scan data

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
// create entities and post the entity

IntellectualEntity ie =
    TestUtil.createTestEntityWithMultipleRepresentations("entity-2");
this.postEntity(ie);

The TestUtil class helps you to create an entity with no or multiple representations and can be used to test your SCAPE Connector API implementation.

Refer to
https://github.com/astromatthias/scape-fedora4-test
https://github.com/openplanets/scape-platform-datamodel
• Create a sample set of objects
SCAPE APIs

This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
• Connector API
  • The Connector API's purpose in the SCAPE platform is to integrate different repositories with the various SCAPE components
  • The content repository exposes a RESTful API
• Use Cases
  • Loader application for batch ingest
  • Request Intellectual Entities by the computation cluster
  • Update Intellectual Entities with provenance information
  • Partially fetching large scale files (e.g. named bitstreams in ARC container)

Document available on Github
https://github.com/openplanets/scape-apis
Ingest an Intellectual Entity
Ingest an Intellectual Entity asynchronously
Retrieve an Intellectual Entity
Retrieve a version list for an Intellectual Entity
Retrieve a File
Retrieve named bit streams
Retrieve a metadata record
Retrieve a set of Intellectual Entities
Retrieve the lifecycle status of an entity
Retrieve a Representation
Update an Intellectual Entity
Update a Representation of an Intellectual Entity
Update the metadata of an Intellectual Entity
Search Intellectual Entities in a collection
Search Representations in a collection
Search Files in a collection
Retrieve an Intellectual Entity
Retrieval of entities is done via a GET request. The parameter `useReferences` controls whether the response is created using references to the metadata via `<mdRef>` elements or if the metadata should be wrapped inside `<mdWrap>` elements in the METS document.

**Path:**
/entity/<entity-id>/<version-id>?useReferences=[yes|no]

**Method:**
HTTP/1.1 GET

**Parameters:**
- entity-id: *the id of the requested Intellectual Entity*
- version-id: *the version of the requested entity (optional)*
- useReferences: *Whether to wrap metadata inside `<mdWrap>` elements or to reference the metadata using `<mdref>` elements. Defaults to yes.*

**Produces:**
A XML representation of the requested Intellectual Entity version

**Content-Type:**
text/xml
SCAPE APIs

• Plan Management API
  • The Plan Management API is a set of HTTP endpoints for serving content in a SCAPE environment. It’s purpose is to integrate the various components in the SCAPE platform that handle preservation plans.
  • Use Cases
    • Deploying new plans in the Repository
    • Get state information about plans
    • Execute a preservation plan on the Workflow Execution Environment
    • Change existing plans
    • Get a specific plan based on some criteria
    • Reservation of Identifiers
Retrieve a plan
An endpoint for plan retrieval is exposed by the repository. Plans are returned according to the Plato Version 4 XML schema, that is currently in development.

**Path**
/plan/<id>

**Method**
HTTP/1.1 GET

**Parameter**
id *the id of the plan to be fetched from the repository*

**Produce**
A XML representation of the plan

Available on Github:
https://github.com/openplanets/scape-apis
Tests of the API - Demo

• Test of the Connector API
Recap!

SCAPE Integration
- SCAPE Digital Object Model
- SCAPE Platform Model (Code)
- Connector API
- Plan Management API

Other cool stuff
- Akubra HDFS Storage Layer [Fedora 3]
- SCAPE Loader Application [Demo]

Scalability
- Fedora 4
Repositories Storage Layer

Loose Coupling - Export Use Cases

Repository and Hadoop cluster are separated instances. Storage and computation are distinct. Data needs to be exported to the computation cluster via Connector API.
Repository and Hadoop are wired together via akubra plugins. Repository, computation and storage is just one big infrastructure.

Other Implementations of akubra with iRODS and DELL DX Object Storage exists
Hadoop as a Storage Layer – Use Cases

• No moving of data to the cluster needed

• Data can be moved very efficiently using Hadoop tools like distcp to another cluster or location on the same cluster.

• Repository can use MapReduce Jobs to monitor / computation tasks direct on the data.

• But HDFS is an immutable file system (write once)
  • Update of files = delete / create cycle
  • Not optimized for random file access
  • Small file problem
Hadoop Issue – Small File Problem

• **Sequence File**
  • Key (filename)-Value (file content) file
  • Can be slow to create a sequence file of existing data
  • Write data directly into a sequence file better option than packing afterwards.
  • No random access.
  • This file is append-only.

• **HBASE**
  • Random Access
  • realtime read/write access

More:
**Akuba HDFS is a storage layer for Fedora 3**
http://wiki.opf-labs.org/display/SP/akuba-hdfs
Fedora 3 to 4

• What’s the problem?
  • A large, aging codebase
  • Declining year-on-year number of developers
  • Declining year-on-year number of commits
  • => slow to develop new features
  • => hard to attract new developers

Change in culture: Customer driven Data driven

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Fedora 4 Use Cases

- Repository scales horizontally
- Repository supports storage services (e.g. Amazon Glacier)
- Ingesting large files into the repository
- Policy-driven Storage
  - Separating objects and datastreams into independent stores.
  - Separating large resources and small resources, for some simple definition of "large" and "small", into independent stores.
  - Separating objects and their datastreams into independent stores based on the owner of the object.
- Many more....

https://wiki.duraspace.org/display/FF/Use+Cases
Fedora 4 – Some Features

Durability
- Self-Healing
- Transactions
- Clustering for high availability
- Metric and reporting

Performance
- Batch Operations
- Clustering for Scalability
- File Storage Connector / Projection, aka „instant ingest“

Flexibility
- Eventing, messaging & web hooks
- Policy driven storage
- Sequencer (extract metadata)
- Storage options
- Easy install & deployment

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Fedora 4 - Architecture

This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Fedora 4

- Fedora Futures Wiki
  - [https://wiki.duraspace.org/display/FF/Fedora+Futures+Home](https://wiki.duraspace.org/display/FF/Fedora+Futures+Home)
- Fedora Google Tech Group
  - [https://groups.google.com/forum/#!forum/ff-tech](https://groups.google.com/forum/#!forum/ff-tech)
- Fedora 4 Alpha 1 Release
  - [https://github.com/futures/fcrepo4/releases/fcrepo-4.0.0-alpha-1](https://github.com/futures/fcrepo4/releases/fcrepo-4.0.0-alpha-1)
- Fedora 4 SCAPE Connector API impl
  - [https://github.com/openplanets/scape-fcrepo4-connector](https://github.com/openplanets/scape-fcrepo4-connector)
- Fedora 4 SCAPE Plan Management API
  - [https://github.com/openplanets/scape-fcrepo4-planmanagement](https://github.com/openplanets/scape-fcrepo4-planmanagement)
- Fedora 4 SCAPE war file

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
METS Loading Demonstration with Fedora 4

Test Cluster at Steinbuch Center for Computing SCC

This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
The METS Loader Application

- Loads SIPs [Submission Information Package] into your repository
- Reads SIPs from local filesystem, HDFS (from sequence files) or a ZIP file
- Reports the ingest lifecycle states periodically
- Logs the ingest progress
- Easy to configure via command line interface
- Can be used by any repository that exposes the Data Connector API (CRU[D] RESTful Service).
- source code on Github.
  https://github.com/shaibe/loader-app

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
The Demo

• The Loader Application registers the METS files for ingest.

• The Loader Application posts the METS files to the repository using the REST endpoint for asynchronous ingest.

• The SIPs will be queued by the repository for ingest.

• The Loader Application receives from the repository the ID of the Digital Object.

• The repository stores and indexes the content – creating AIP [Archival Information Package].

• The Loader Application retrieves periodically the lifecycle state of the objects.

This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).