What to do with 10^6 Books? Install aDORe!

Herbert Van de Sompel

Digital Library Research & Prototyping Team
Research Library
Los Alamos National Laboratory

herbertv@lanl.gov
http://public.lanl.gov/herbertv/

Los Alamos NATIONAL LABORATORY

aDORe Presentation Based on the Paper

Herbert Van de Sompel, Ryan Chute, Patrick Hochstenbach. The aDORe Federation: Digital Repositories at Scale. 40 pages. International Journal on Digital Libraries. Special Issue on Very Large Digital Libraries. In Press, 2008. Preprint available at <http://arxiv.org/abs/0803.4511>

The aDORe Project: a Lot of People and a Lot of Work

- The aDORe design and code-base is the results of the research efforts & technical expertise of the following people:
  - Jeroen Bekaert, PhD
  - Luydimilla Balakireva, PhD
  - Ryan Chute
  - Patrick Hochstenbach
  - Henry Jerez, PhD
  - Xiaoming Liu, PhD
  - Herbert Van de Sompel, PhD

- Henry Jerez and Jeroen Bekaert received their PhDs based on research that was directly related to aDORe.

- The current aDORe version was developed by Ryan Chute and Luydimilla Balakireva.

The aDORe Project: Major Drivers

- Concrete need to design and implement a solution to ingest, store, access the vast and growing collection of the LANL Research Library.
  - Scale!
    - The order of magnitude for the amount of objects is ~ 10^7
  - Existing open source solutions (at that time) did not meet our scale requirements.
- Interest in repository interoperability, cf. involvement in OAI-PMH, NISO OpenURL, OAI-ORE
- Interest in digital preservation, cf. NDIIP funding

Core characteristics of aDORe Archive and Federation software

- Standards-based:
  - MPEG-21 Digital Item Declaration, the MPEG-21 Digital Item Identification, URI, info URI, OAI-PMH, NISO OpenURL, SRU, Information Environment Service Registry, Internet Archive ARC file format, OAIS concepts, XML, XML Schema, XQuery.
- Component-based, highly modular:
  - Multiple content repositories, Identifier Locator, Service Registry, Format Registry, Semantic Registry, Harvesting front-end, Dissemination front-end
  - 3-Tier federation architecture
  - Modularity not exposed to downstream applications
- Protocol-based:
  - All components expose HTTP-based service interfaces
  - All “read” services based on 4 standards: OAI-PMH, NISO OpenURL, SRU, XQuery.
  - Interaction between modules is protocol-driven.

The aDORe environment @ LANL, September 9 2008

- 90,155,141 Digital Objects
- 216,653,688 Datastreams
- ~ 9,700 autonomous repositories:
  - ~ 4,200 XMLtapes: XML-based Surrogates for Digital Objects
  - ~ 5,500 ARCfiles: Datastreams of Digital Objects
- > 550,000,000 identifiers

I was not joking when I just said Scale, Modularity, Federation

The aDORe Federation software

- Available at:
  <http://african.lanl.gov/aDORe/projects/aDOReFederation>
  All credits to Ryan Chute & Luydimilla Balakireva

- This is a major update to the aDORe Archive:
  - Updates the Tier-1 aDORe Archive
  - Implements the 3 Tiers of the architecture instead of only Tier-1

- In production at LANL Research Library for over 1 year
  - Lucene/SOLR search engine built on top

Before you throw out your current repository solution …

- aDORe is a bare-bones type repository:
  - Provides a variety of machine-interfaces for ingest and retrieval
  - Has no human interfaces, except for adminstrator-level interfaces
  - Applications are laid on top of aDORe
  - Human interfaces are provided by these overlaid applications
- aDORe is attractive for large collections of relatively stable objects [write once - read many - edit none - delete none]

aDORe could be used as a plug-in storage component for other repository solutions: BOF later today!

Insights in aDORe: A combination of …

- Conceptual:
  - aDORe Federation architecture: A high-level, 3-Tier architecture for the federation of distributed repositories.
- Concrete:
  - The aDORe Archive storage solution (XMLtapes/ARCfiles) - Tier-1 of the aDORe Federation architecture.
  - The aDORe Federation software - an implementation of the 3-Tier architecture, with the aDORe Archive in Tier-1.
- Some compromises when explaining the architecture because lack of time.
- Read the Federation paper for full info. Take a drug of your choice before you do so …

The aDORe Federation Architecture: Goal

- Facilitate a uniform manner for client applications to discover and access content objects available in a group of distributed repositories.
- Single repository behavior for a group of distributed repositories.

- Note that these distributed repositories can very well be “hidden” and that only the federated result is made “public”.

- Not about uniform approaches to add, update, delete objects in repositories.
  - Considered the responsibility of individual repositories.
  - However, changes are made apparent to the federation.

The aDORe Federation Software: Goal

- Ingest, Store and Access a lot of stuff

Overview of aDORe Archive and Federation

- Content Objects
  - Digital Objects, Surrogates, Datastreams
  - MPEG-21 DIDL for XML Surrogates
  - Identification of Digital Objects, Surrogates, Datastreams
- Implementation of the 3-Tier aDORe Federation Architecture
  - Tier 1: Storing Digital Objects
    - A multitude of autonomous distributed content repositories (ARCfiles and XMLtapes) with a set of service interfaces
  - Tier-2: Locating Service Interfaces, Digital Objects, Surrogates, and datastreams
    - Service Registry
    - Identifier Locator
  - Tier-3: Providing federated access to the autonomous distributed repositories:
    - Federator: Harvesting Surrogates
    - Resolver: Requesting (services pertaining to) Digital Objects, Surrogates, Datastreams

Content Objects

- 3 types of Content Objects:
  - Digital Object (conceptual: aggregation of Datastreams and Properties)
  - Surrogate (concrete: XML-based representation of Digital Object)
    - LANL uses the ISO-standardized MPEG-21 DIDL format
    - aDORe Archive can deal with any XML-based CO format
  - Datastream (concrete: the real stuff)

Content Objects

- Core enabling properties in the aDORe:
  - Identification (non-protocol URI)
  - Location (protocol URI)
  - time-stamp (ISO8601)

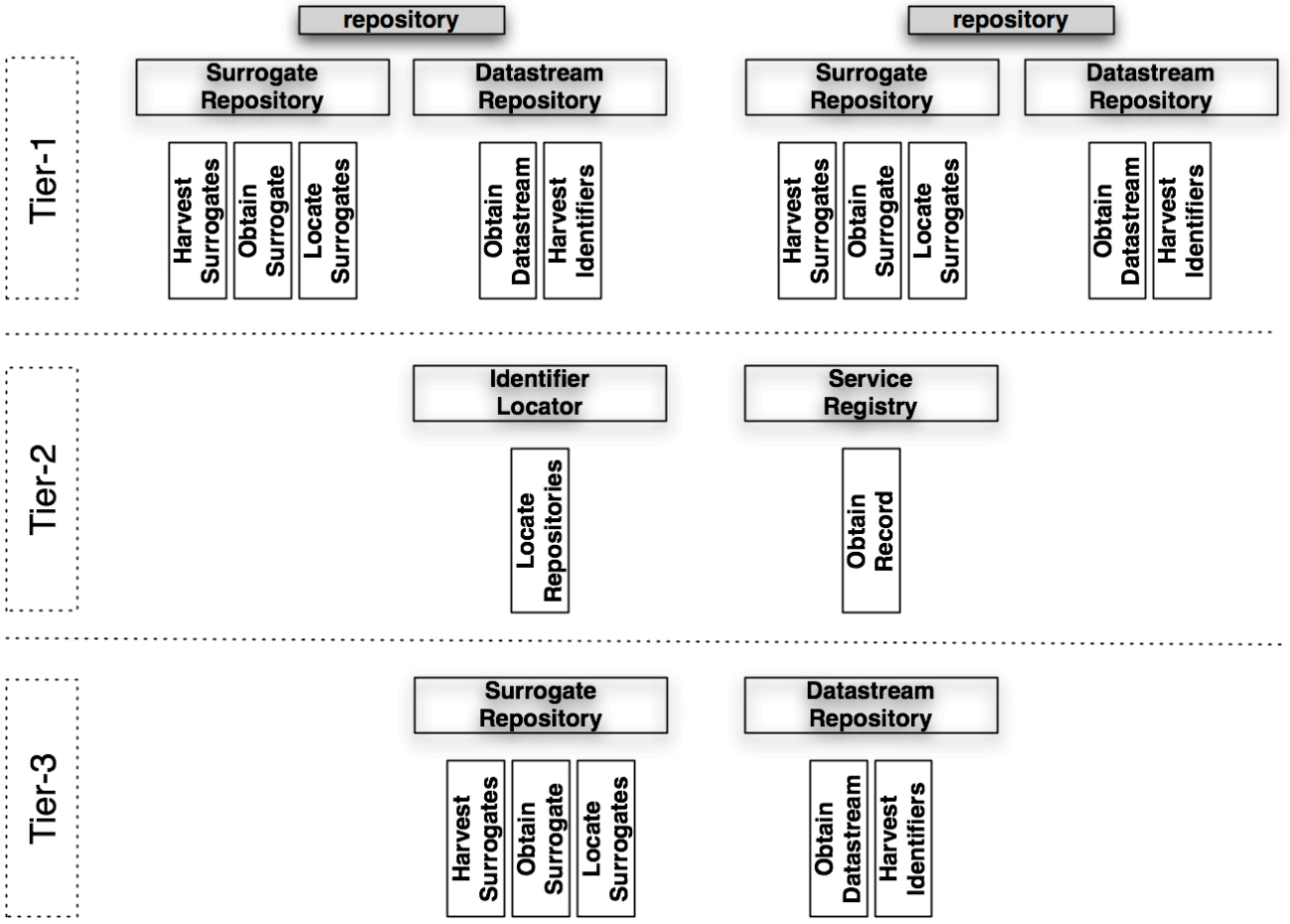
  of these Content Objects

- In the aDORe Archive:
  - identification via non-protocol-based URIs
    - Digital object: `info:doi/10.145/september2008-rchute`
    - Surrogate: `info:lanl-repo/i/a6453373ed5ce`
    - Datastream: `info:lanl-repo/ds/99eeab724ef2`
  - no location
  - time-stamp (ISO8601)

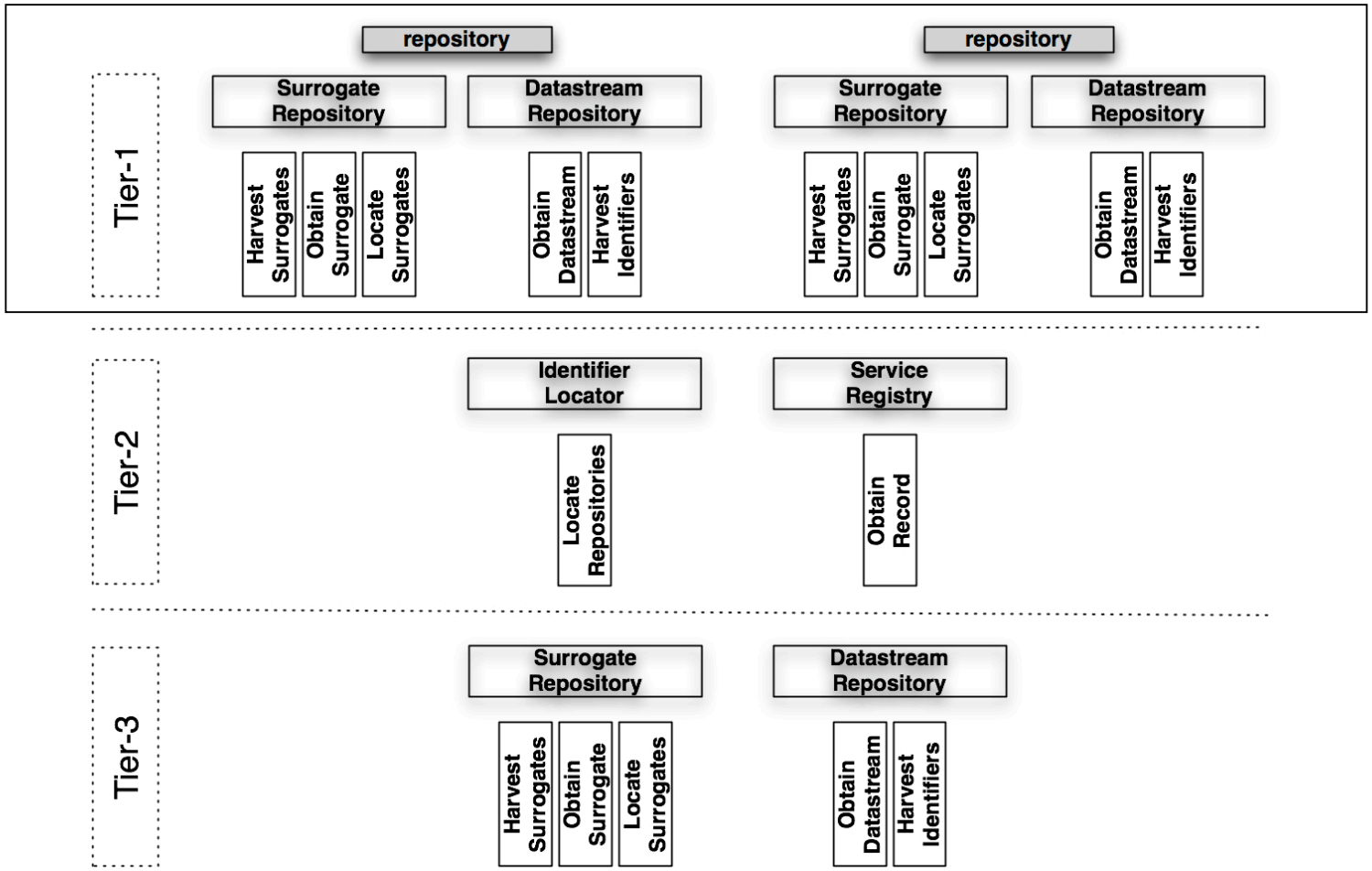
Content Objects

- One or more Surrogates can correspond with a Digital Object in a federation:
  - Digital Object with same URI may exist in multliple repositories
  - Single repository may have multiple Surrogates for a Digital Object
- A Datastream can be part of multiple Digital Objects

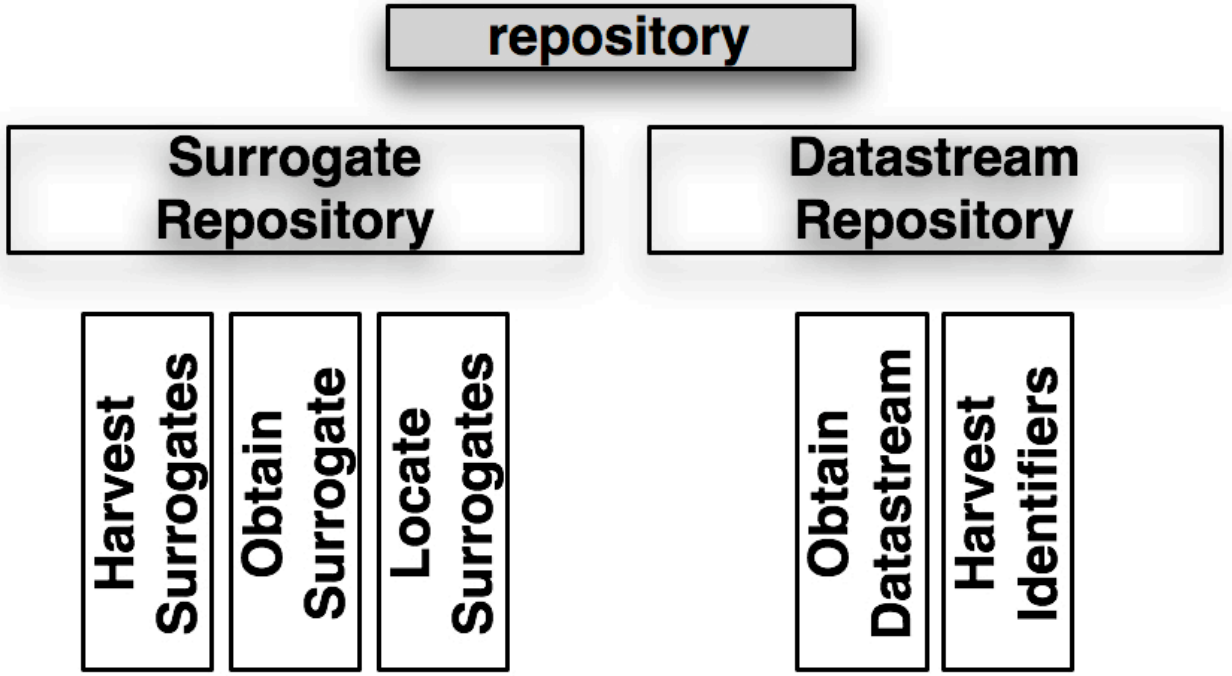
aDORe Federation Architecture



aDORe Federation Architecture: Tier-1

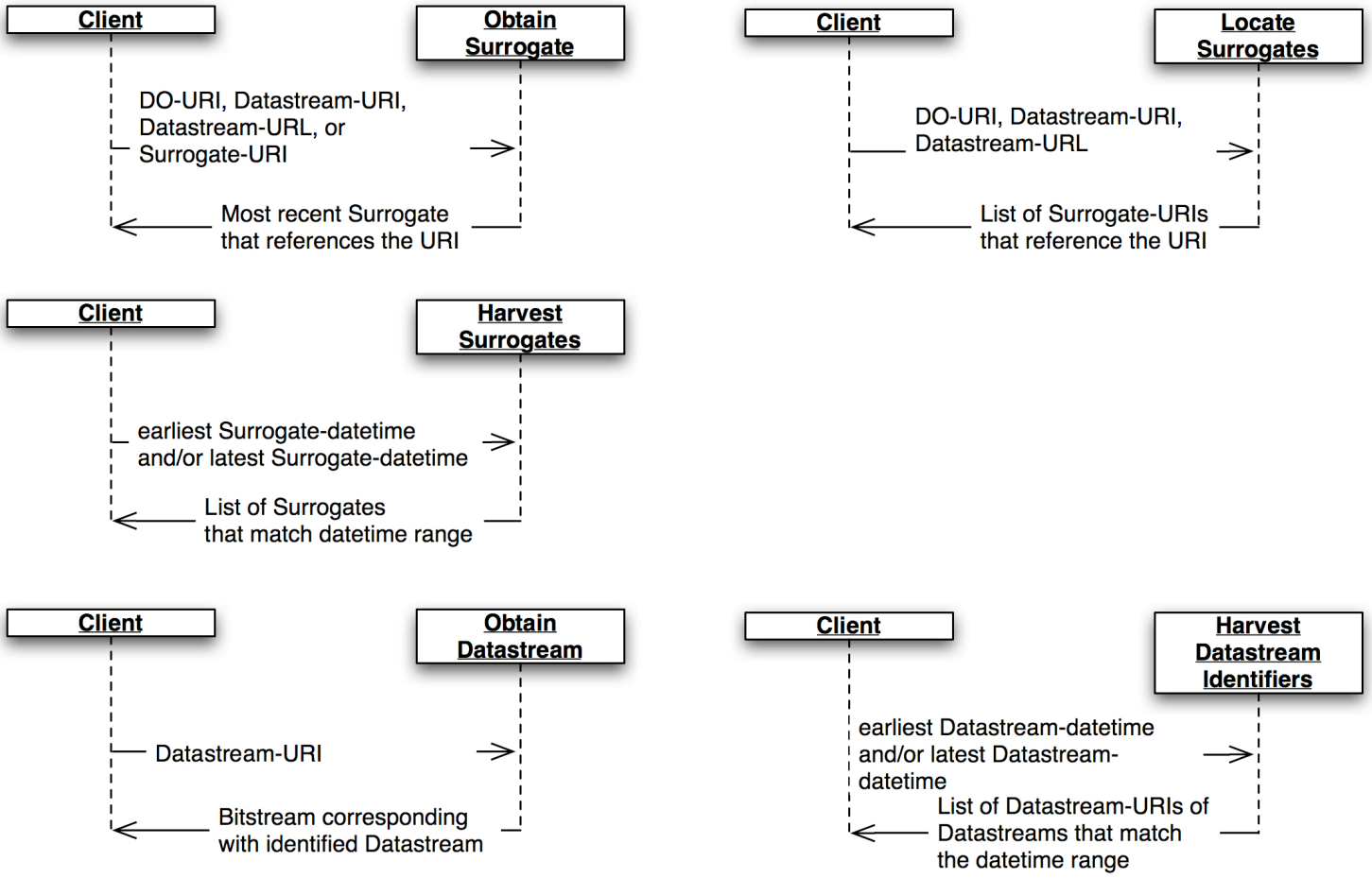


Tier-1: Surrogate and (sometimes) Datastream Repositories

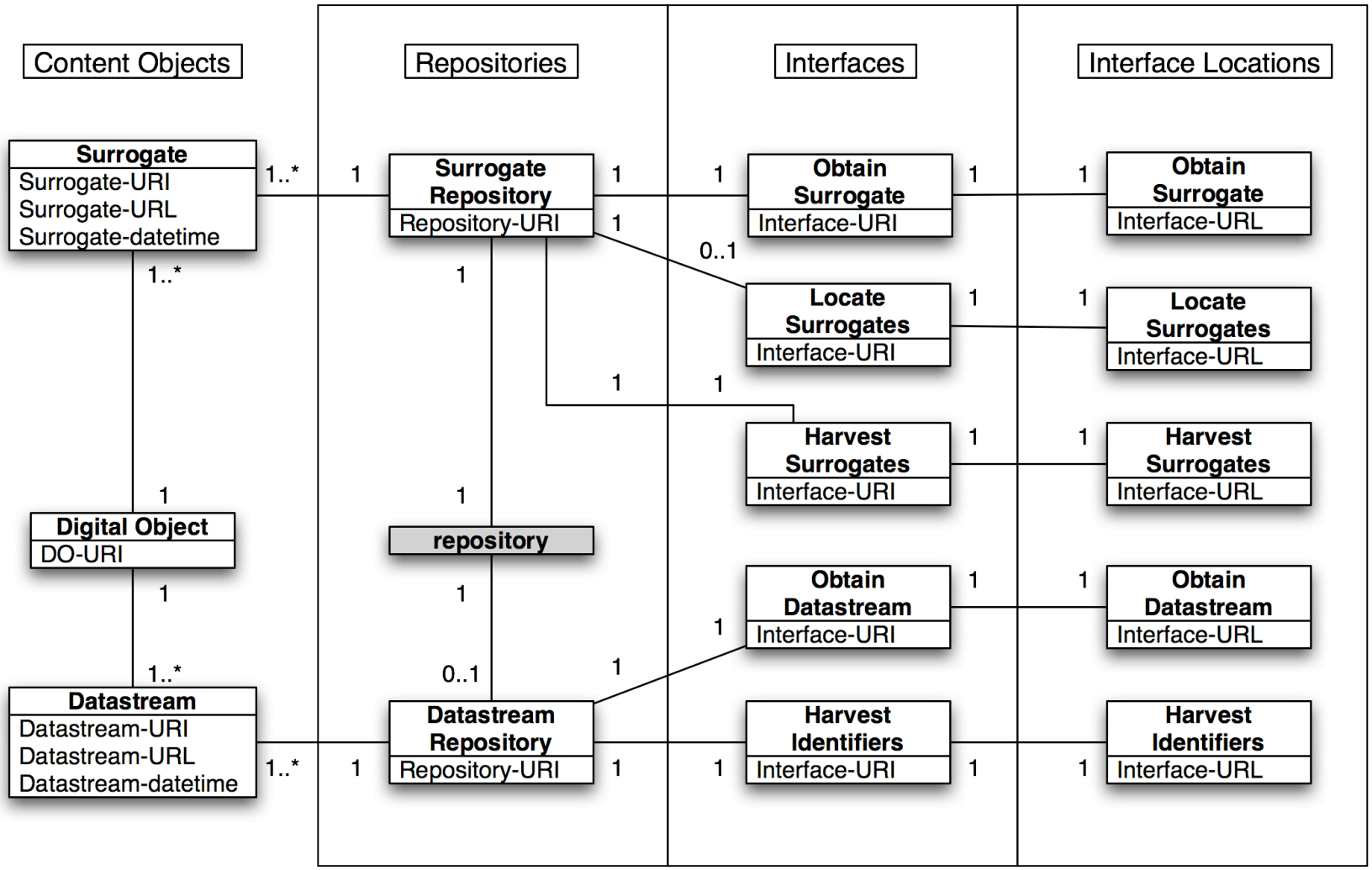


- Surrogate Repositories, Datastream Repositories as well as their Interfaces identified by URI
- Interfaces leverage identification, time-stamping of Content Objects
- Datastream Repository only when using (non-protocol-based) Datastream-URIs

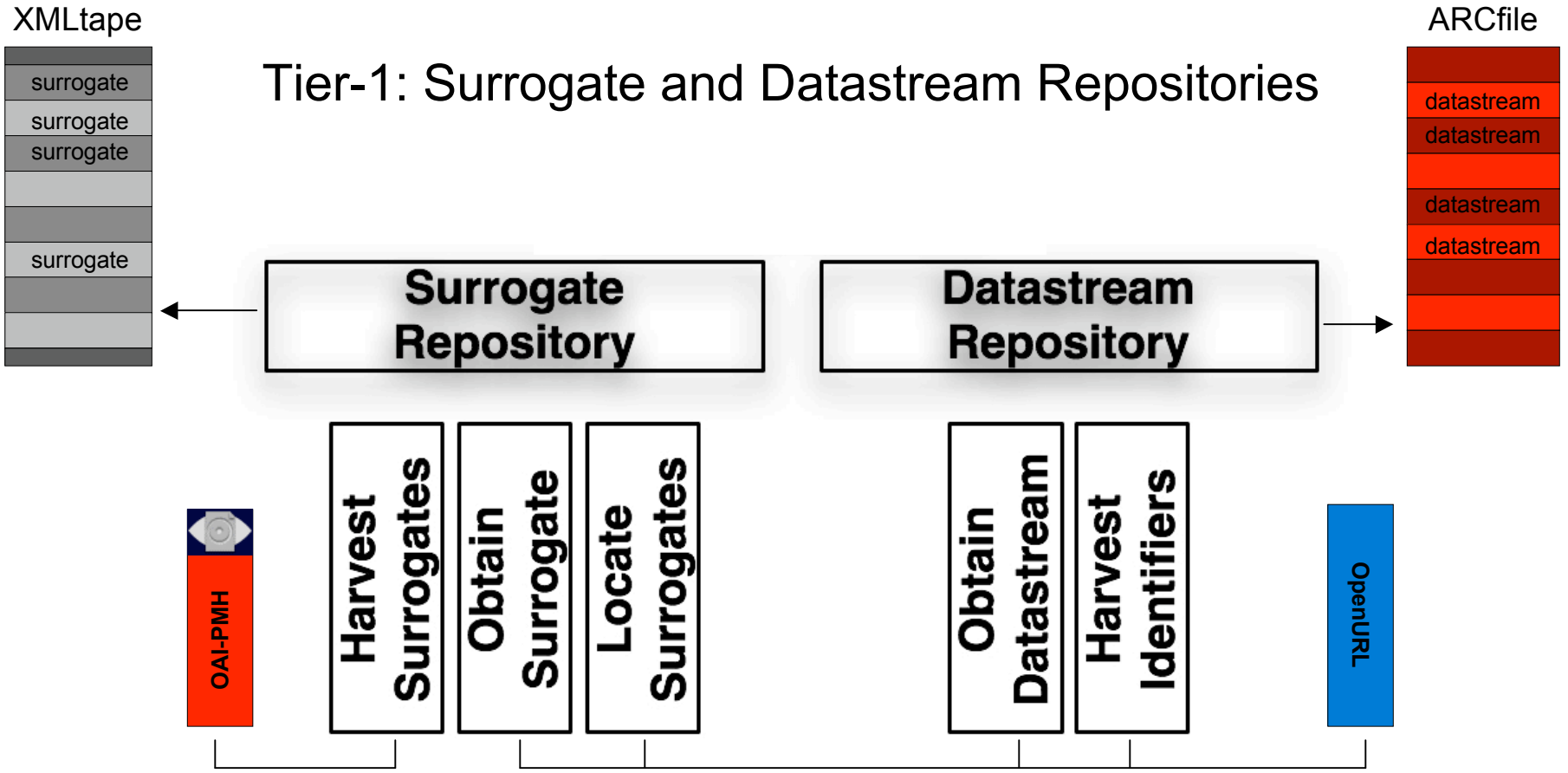
aDORe Federation Architecture: Tier-1



aDORe Federation Architecture: Tier-1

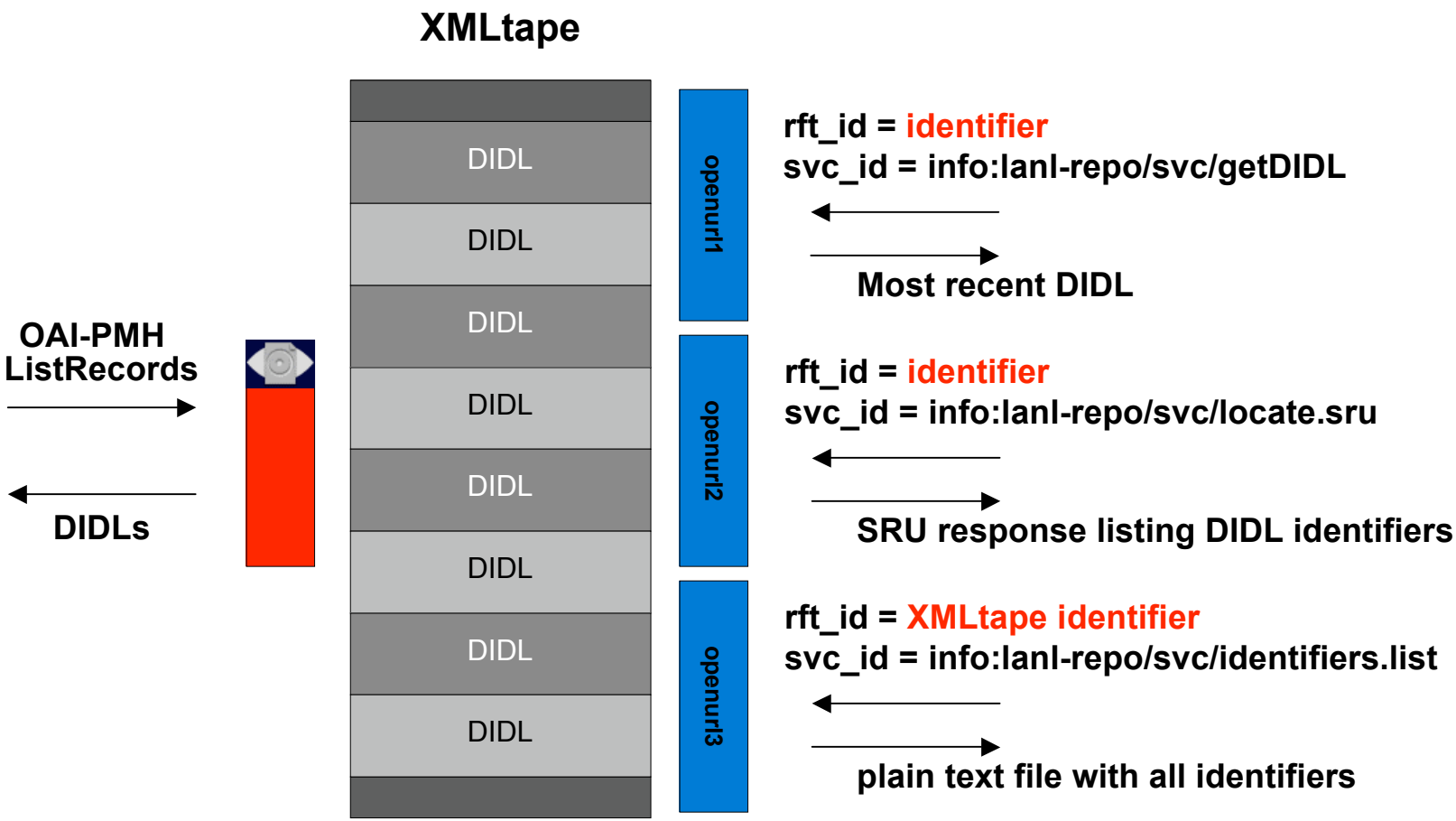


Tier-1: Surrogate and Datastream Repositories

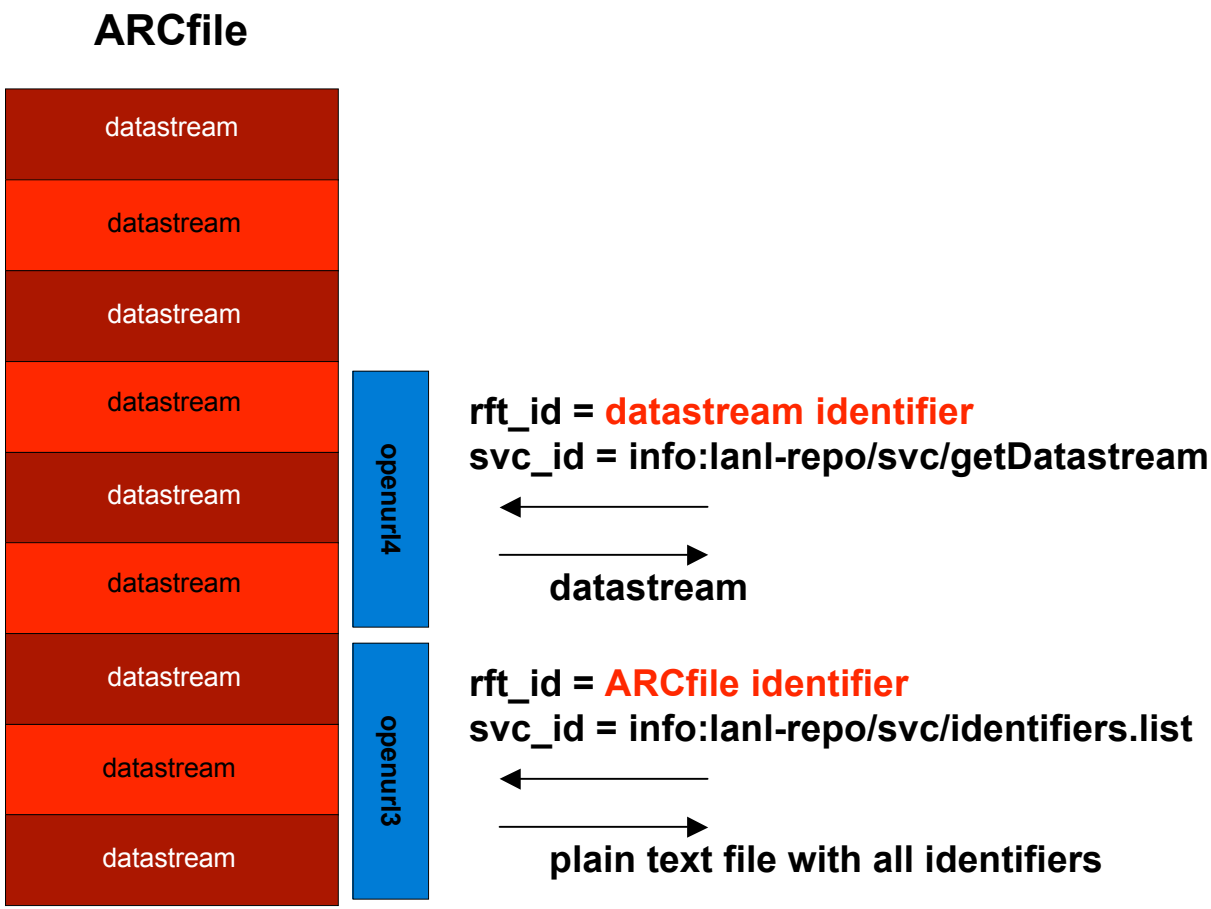


aDORe Archive

aDORe Archive : XMLtapes (Surrogate repository)



aDORe Archive : ARCfiles (Datastream repository)



Tier-1 Implementation - aDORe Archive

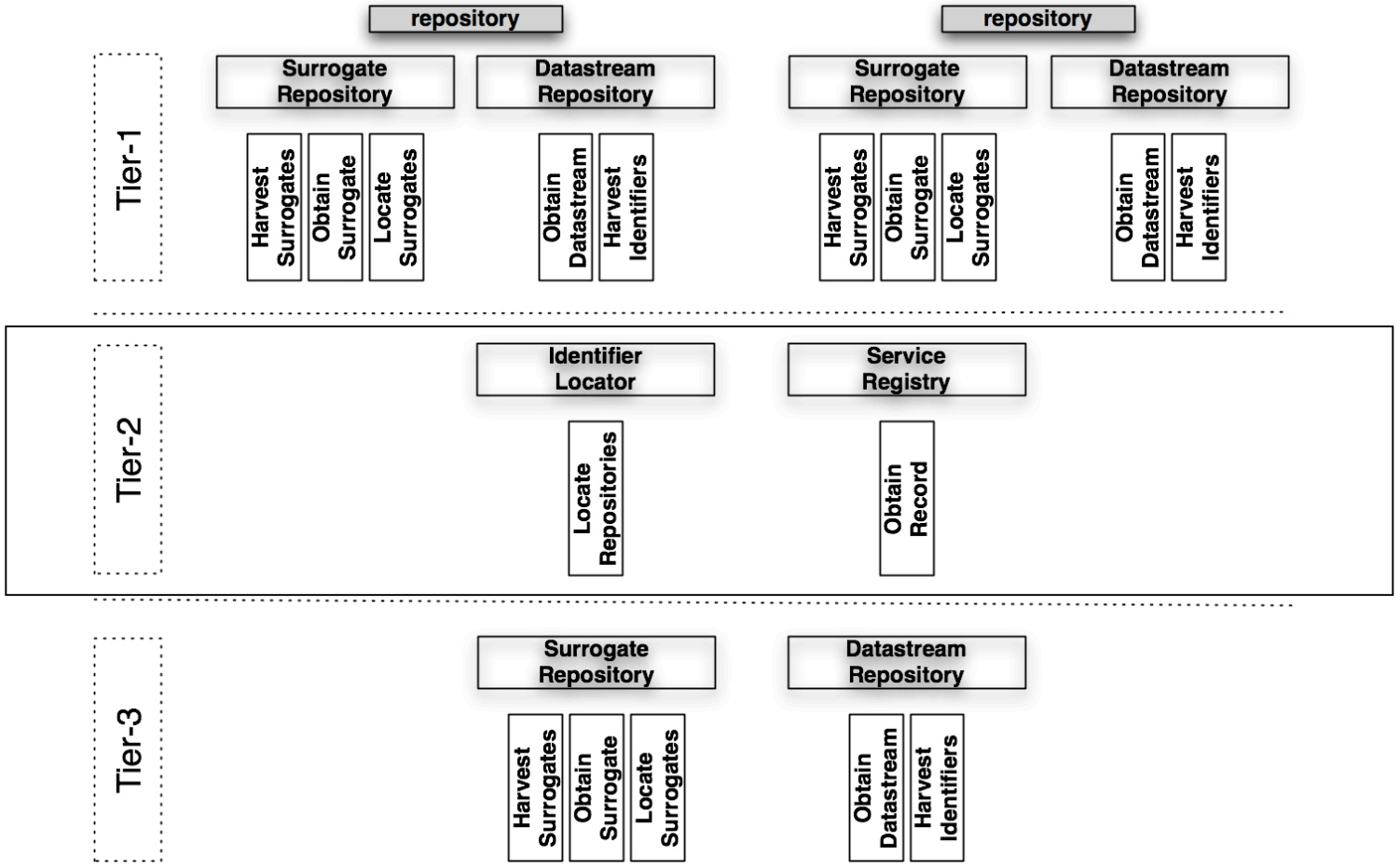
- Surrogate Repository (adore-xmltape)
  - 5 aDORe Modules (111 Classes with ~10,000 lines including comments and copyright)
  - Read/Write/Index API: adore-xmltape
  - Large Repository Index Implementation: adore-xmltape-indexbdb
    - Uses Oracle Berkeley DB Java Edition (Separate due to GPL)
  - OAI-PMH: adore-archive-accessor
    - Based on OCLC's OAICat software
  - OpenURL: adore-xmltape-resolver
    - Provides Obtain Surrogate, Locate Surrogate, and Harvest Identifiers services
    - Based on OCLC's OpenURL Resolver software
  - OpenURL: adore-xmltape-xquery
    - Generic XQuery Engine for XMLtapes
    - Extensible Plug-in Framework for new query and return format types
    - Based on OCLC's OpenURL Resolver software

Tier-1 Implementation - aDORe Archive

- Datastream Repository (adore-arcfile)
  - 2 aDORe Modules (35 Classes wth ~4000 lines including comments and copyright)
  - Read/Write/Index API: adore-arcfile
    - Based on ARC API in Internet Archive's Heritrix software
  - OpenURL: adore-arcfile-resolver
    - Provides Obtain Datastream and Harvest Identifiers services
    - Based on OCLC's OpenURL Resolver software

- Repository Registries (adore-xmltape-registry & adore-arcfile-registry)
  - Repository metadata stored in MySQL Databases
  - OAI-PMH Interface for each registry, based on OCLC's OAICat software

aDORe Federation Architecture: Tier-2

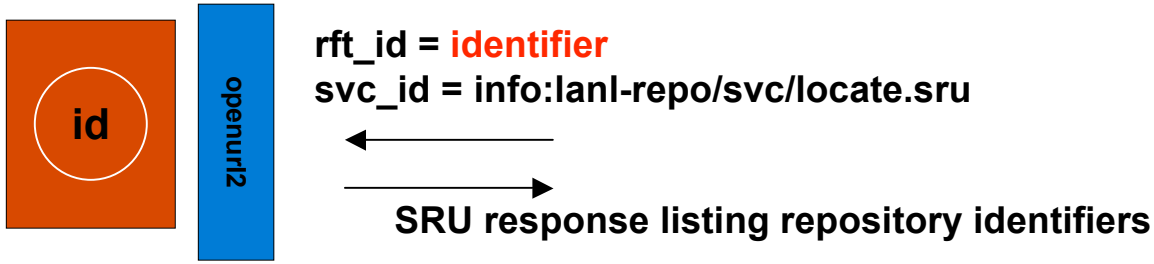


Tier-2: Identifier Locator

- Look-up table:
  - Identifiers of Content Objects
  - Identifiers of Datastream or Surrogate Repositories that make these Content Objects accessible
- Populated by recurrently interacting with Harvest Surrogates and Harvest Datastream Identifier interfaces of all Tier-1 repositories.
- Identifier Locator knows about these interfaces via the Service Registry.

aDORe: Identifier Locator

identifier locator	
identifier	**Repository identifier**
info:lanl-repo/i/1	info:lanl-repo/xmltape/5
info:doi/10.145/22756	info:lanl-repo/xmltape/5
info:pmid/35534372	info:lanl-repo/xmltape/7
info:lanl-repo/ds/22	info:lanl-repo/xmltape/5
info:lanl-repo/ds/22	info:lanl-repo/arc/33
info:doi/10.145/22756	info:lanl-repo/xmltape/10



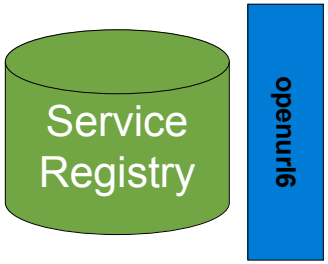
Tier-2: Service Registry

- Keeps track of all components in the federation. In essence 2 look-up tables.
- Look-up Table 1:
  - URI of component (e.g. Repository-URI)
  - Matching Interface-URIs (and Interface type)
- Look-up Table 2
  - Interface-URI
  - Interface-URL
- Implementation based on JISC Information Environment Service Registry lay-out
- OAI-PMH, OpenURL and SRU service interfaces

aDORe: Service Registry

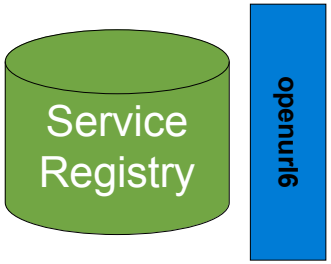
Service registry			
component identifier	**service identifier**	**supported standard**	**service location**
info:lanl-repo/xmltape/aa34	info:lanl-repo/int/aa34/oaipmh2	oaipmh2	http://…
info:lanl-repo/xmltape/aa34	info:lanl-repo/aa34/openurl-aDORe1	openURL-aDORe1	http://…
info:lanl-repo/xmltape/aa34	info:lanl-repo/int/aa34/openurl-aDORe2	openURL-aDORe2	http://..
info:lanl-repo/xmltape/aa34	info:lanl-repo/int/aa34/openurl-aDORe3	openURL-aDORe3	http://..
info:lanl-repo/arc/bee4	info:lanl-repo/int/bee4/openurl-aDORe3	openURL-aDORe3	http://..
info:lanl-repo/arc/bee4	info:lanl-repo/int/bee4/openurl-aDORe2	openURL-aDORe4	http://..
info:/lanl-repo/idlocator	info:lanl-repo/int/idlocator/openurl-aDORe2	openURL-aDORe2	http://..
info:/lanl-repo/svcreg	info:lanl-repo/int/svcreg/pmp	pmp	http://..
info:/lanl-repo/svcreg	info:lanl-repo/int/svcreg/oaipmh2	oaipmh2	http://..
…			

aDORe: Service Registry



rft_id = repository identifier

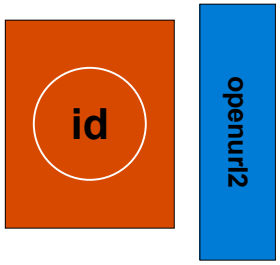
IESR description listing repository information, incl. service identifiers and supported standards



rft_id = service identifier

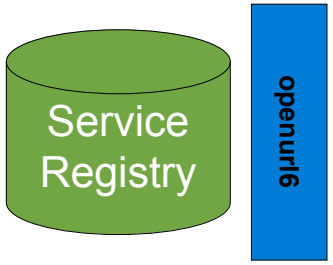
IESR description listing service location

Los Alamos NATIONAL LABORATORY



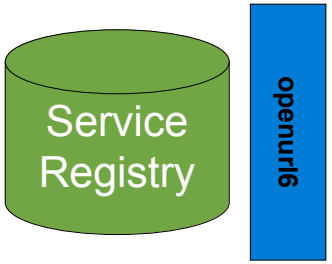
rft_id = identifier
svc_id = info:lanl-repo/svc/locate.sru

SRU response listing repository identifiers



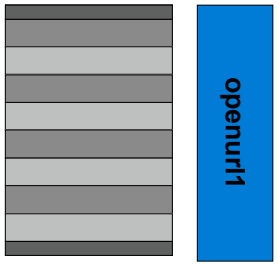
rft_id = repository identifier

IESR description listing repository information, incl. service ids



rft_id = service identifier

IESR description listing service location



rft_id = identifier
svc_id = info:lanl-repo/svc/getDIDL

Most recent DIDL

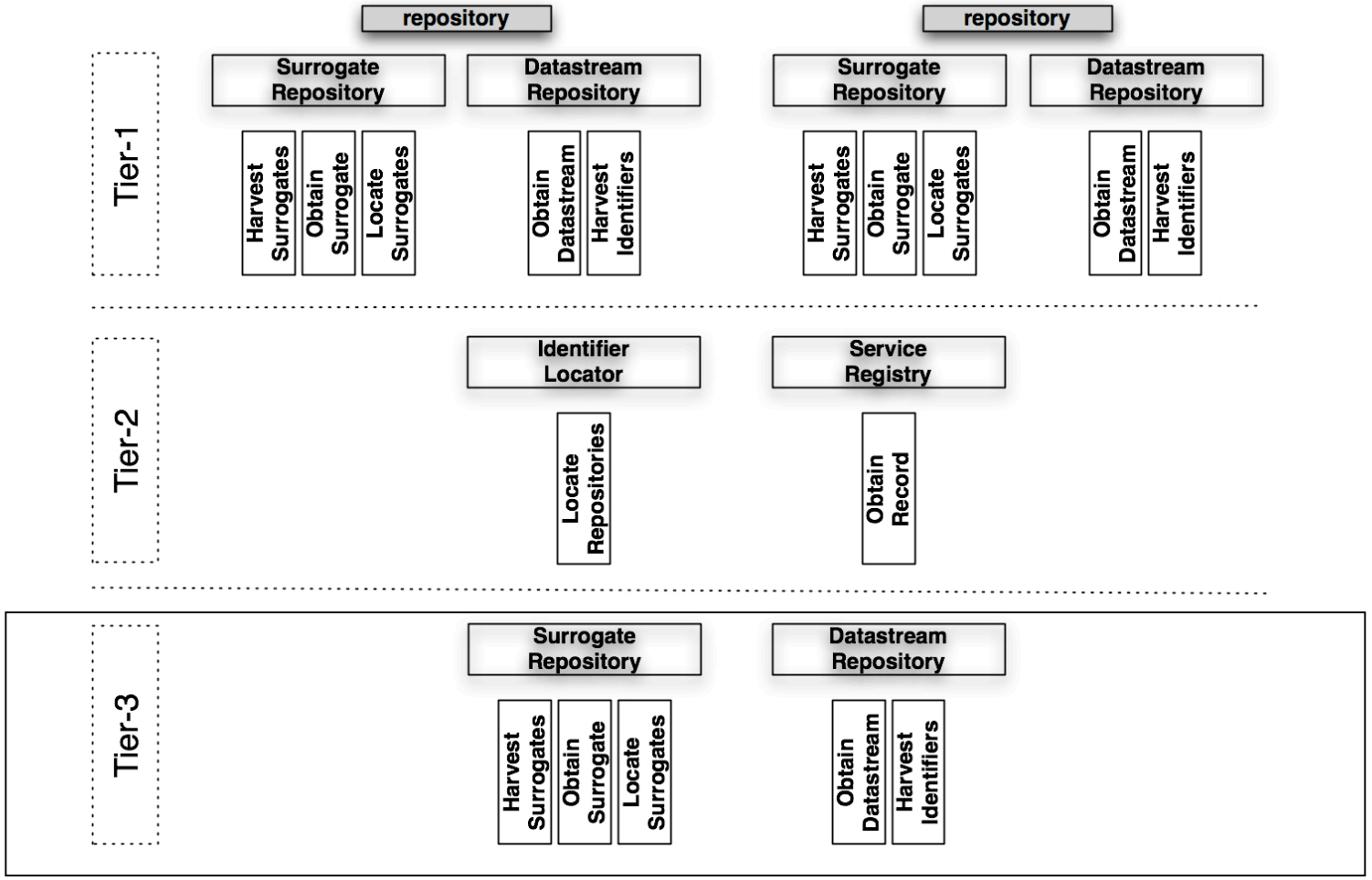
Tier 2 Implementation - aDORe Federation

- Identifier Locator (adore-id-locator)
  - 1 aDORe Module (12 Classes with ~1000 lines including comments and copyright)
  - Stores [identifier, repository identifier, ingestion date] in an in-memory MySQL instance.
  - Goal: Linear Scalability at 1 Billion identifiers with sub-10ms
  - Loaded by retrieving identifiers from aDORe repositories using their openurl-aDORe3 service interface
  - Ingest API: Load API provides direct db load via JDBC
  - Search API: Direct API search or OpenURL service interface
  - OpenURL Interface:
    - Provides Locate Surrogate/Datastream service
    - Based on OCLC's OpenURL Resolver software

Tier 2 Implementation - aDORe Federation

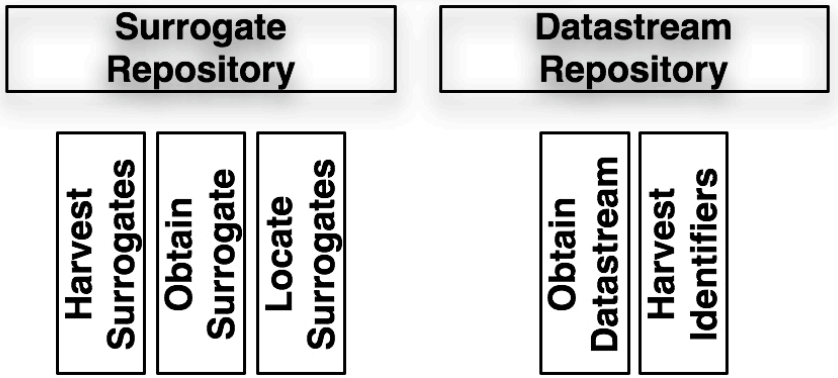
- Service Registry (adore-service-registry)
  - 1 aDORe Module (27 Classes wth ~3000 lines including comments and copyright)
  - Service metadata stored in MySQL Database
  - Schema based on the Ockham Service Registry IESR-based database
  - Read/Write/Delete API
  - OpenURL Interface:
    - Provides access to Collection-level and Service-level records.
    - Based on OCLC's OpenURL Resolver software
  - OAI-PMH Interface:
    - Provides access to Collection-level and Service-level records.
    - Based on OCLC's OAICat software

aDORe Federation Architecture: Tier-3

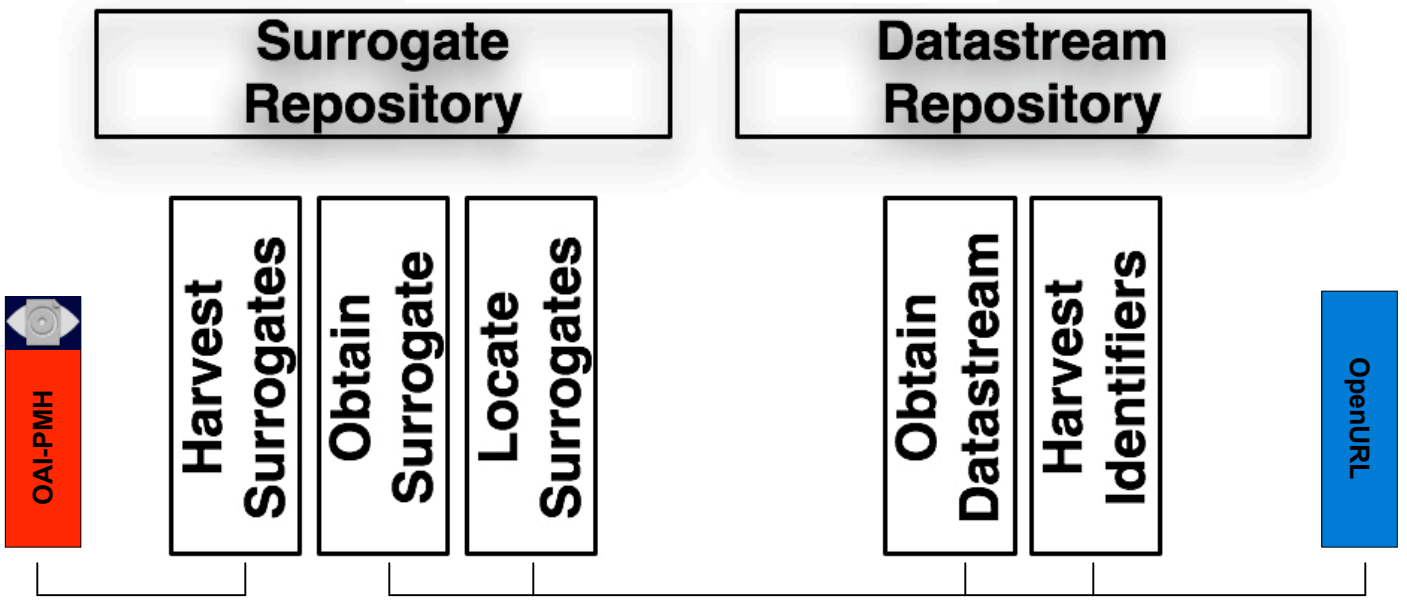


Tier-3: aDORe front-ends

- Expose all the Repositories of Tier-1 as one Surrogate Repository and one Datastream Repository in Tier-3.
- The interfaces of these Tier-3 Repositories interact with the interfaces of Tier-2 and Tier-1 components to respond to requests.



aDORe: OAI-PMH Federator and OpenURL Resolver



aDORe Federation software

Tier 3 Implementation - aDORe Federation

- OAI-PMH Federator (adore-federator)
  - 1 aDORe Module (22 Classes with ~2000 lines including comments and copyright)
  - Single point of access to harvest Surrogate Records from multiple aDORe Tier-1 Repositories.
  - Interacts with Service Registry, Identifier Locator (for OAI-PMH GetRecord) and aDORe repositories
  - Supports DIDL, and can support other compound object formats (e.g. METS, etc.)
  - OAI-PMH Interface:
    - Interface to batch harvesting of all Surrogates across all Tier-1 Repositories
    - Based on OCLC's OAICat software

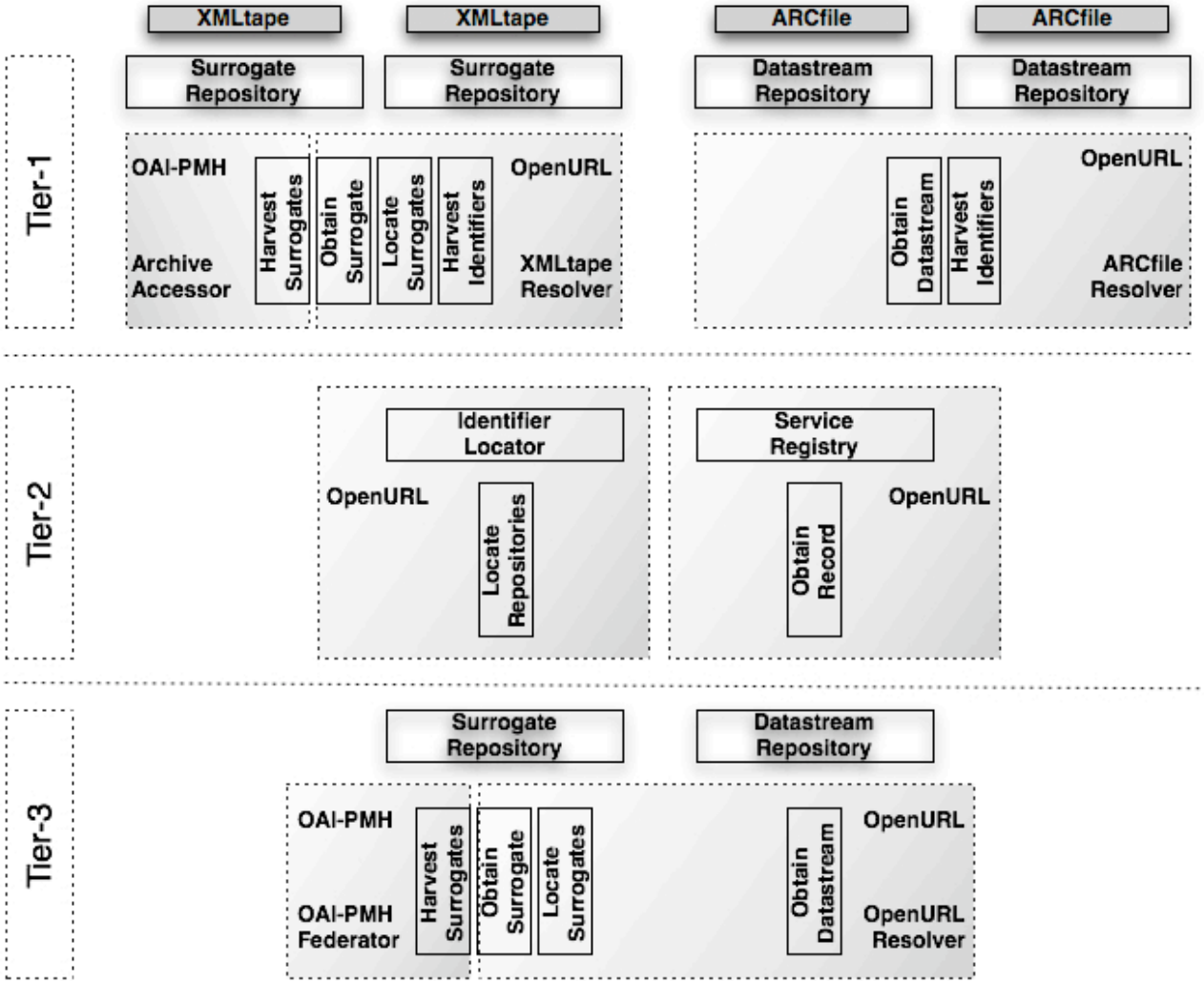
Tier 3 Implementation - aDORe Federation

- OpenURL Resolver (adore-disseminator)
  - 1 aDORe Module (55 Classes wth ~5000 lines including comments and copyright)
  - Supports the core aDORe Tier-1 Repository OpenURL services.
  - Powered by a rule engine that dynamically decides which services are available.
  - Interacts with Service Registry, Identifier Locator, aDORe repositories, rule engine, and transformation services to generate responses.
  - OpenURL Interface:
    - Provides access to core repository services and custom transformation services
    - Based on OCLC's OpenURL Resolver software

aDORe: there is a bit more …

- Tier-3: A **rule engine** dynamically decides which disseminations are available for a specified content object based on properties of the object (format, semantics, collection, creation date, …).
  - All dissemination requests expressed as OpenURLs
- Tier-3: A **PermaLink application** with a Cool URI syntax allows for long-term, Web 2.0 style addressability of content objects and disseminations.
  - Facilitates reuse in other applications, including end-user apps
- Tier-3: Descriptions of all aDORe objects (Datastreams, Surrogate, dynamic disseminations) will be made available in a manner **compliant with OAI-ORE** (RDF/XML, Atom).
  - Facilitates reuse in other applications, including end-user apps
- Tier-1: **Generic XQuery interface** to XMLtapes supports arbitrary queries.
  - Slow but cheap: non-indexed based search of large XML repositories

aDORe Federation software



Los Alamos NATIONAL LABORATORY

But talking about 10^5 books:

Introducing aDORe djatoka to view them …

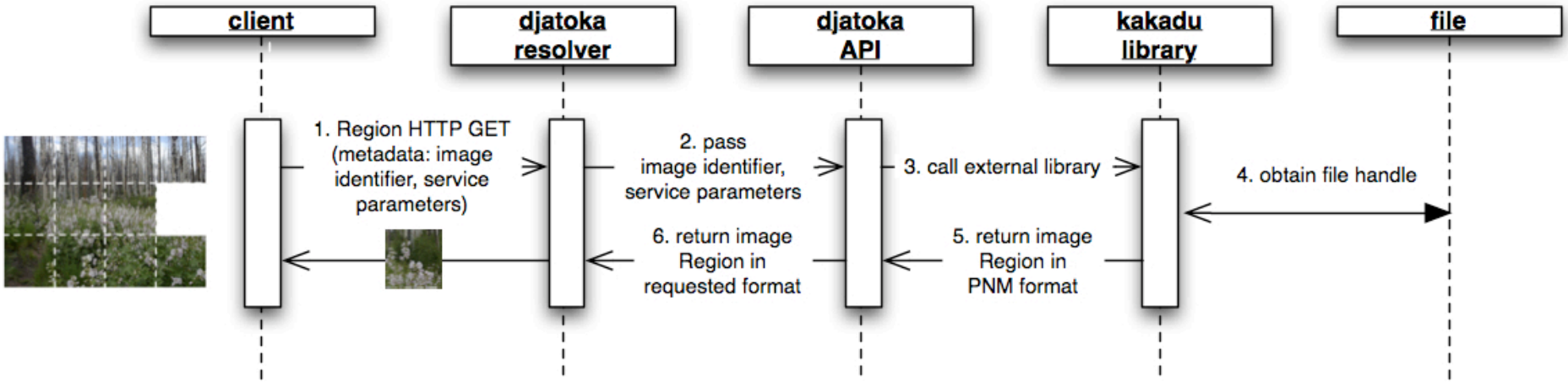
aDORe djatoka Paper

Ryan Chute, Herbert Van de Sompel. Introducing djatoka: A Reuse Friendly, Open Source JPEG 2000 Image Server. D-Lib Magazine, Volume 14 Number 9/10. Available at <http://dx.doi.org/10.1045/september2008-chute>

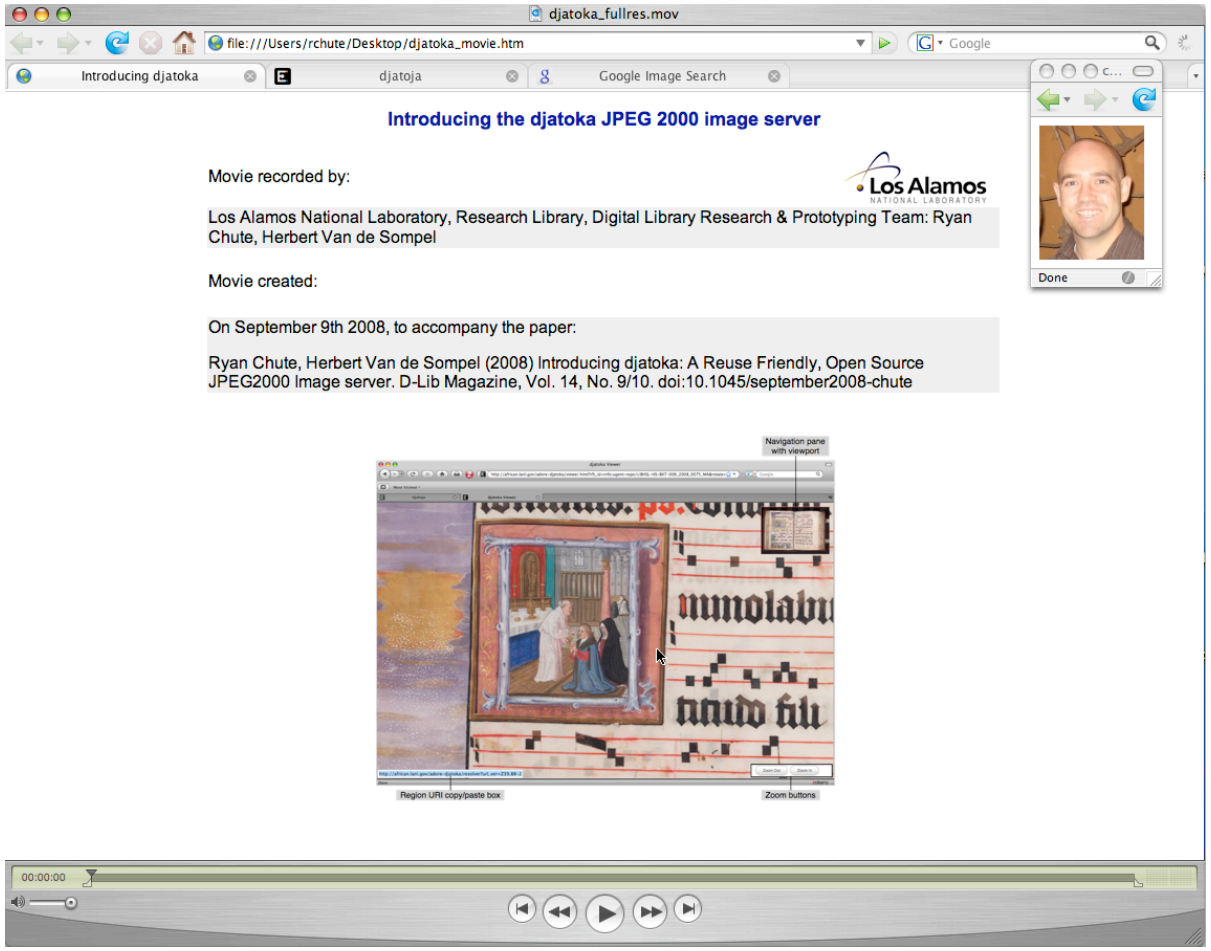
aDORe djatoka: a reuse friendly, open source JPEG 2000 image server

- Use of the ISO-standardized JPEG 2000 format as the service format
- Java-based open source solution built around the Kakudu JPEG 2000 library
- Geared towards reuse through URI-addressability of all image disseminations including regions, rotations, and format transformations
- Provision of a consistent, guessable URI pattern for image disseminations based on the ANSI/NISO OpenURL standard
- Provision of an extensible service framework for image disseminations enabled by OCLC's Java OpenURL package
- Availability of image disseminations in a range of image formats
- Availability of image disseminations for locally stored JPEG 2000 files, as well as for Web-accessible images in a variety of formats
- Configurable server-side, file-based caching
- Ajax-based client reference implementation, which allows panning, zooming, and selecting the URI of the current view

aDORe djatoka: the Flow

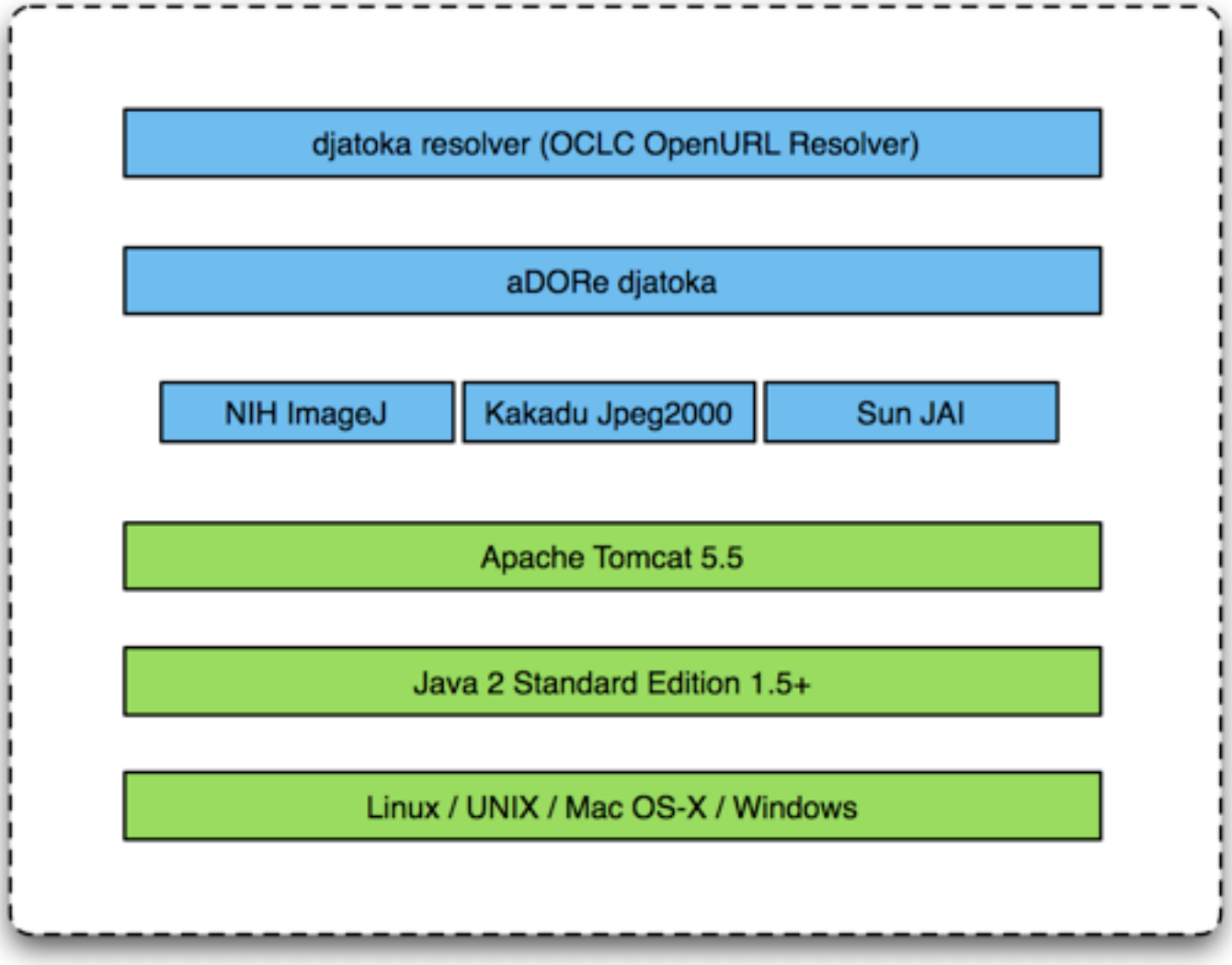


aDORe djatoka: the Show



<http://african.lanl.gov/aDORe/projects/djatoka/djatoka_release.mov>

aDORe djatoka: The Stack



aDORe djatoka: the Download

- Available at:
  <http://african.lanl.gov/aDORe/projects/djatoka>
  All credits to Ryan Chute

- SourceForge effort at:
  <http://sourceforge.net/projects/djatoka>

- Demonstrations at:
  <http://african.lanl.gov/aDORe-djatoka>
  <http://www.antifonarium-tsgrooten.be/highlights.htm>

aDORe djatoka: the Applause …

We're listening …