Repositories, tools, systems

May 19, 2014

Kresimir Duretec (duretec@ifs.tuwien.ac.at)

Agenda

- Digital Object Repository
  - What is a Digital Object Repository
  - FEDORA
  - Implementations
- Systems
  - Repositories
  - Preservation Monitoring
  - Preservation Policies
  - Preservation Planning
- Real world example from the Austrian State Archive

Digital Object Repository (DOR)

- Digital Object Repository is a software system that provides a data management solution for storing content and metadata about digital objects

Fedora

- Fedora – Flexible Extensible Digital Object Repository Architecture (http://fedorarepository.org/)

- originally developed at Cornell University
- open source software
- it is under the stewardship of the DuraSpace (http://duraspace.org)
- many different implementations
- Not related to the Linux distribution Fedora

FEDORA – what it is

- conceptual framework
  - set of abstractions about digital information
- provides basis for software systems that can manage digital information
  - ensuring long-term durability of the information
  - making information directly available to be used in a variety of ways
- foundation upon which to build different information management systems for different use cases
- **not a full solution !**

FEDORA basics

- all content is managed as data objects which are composed of components (datastreams)
- components (datastreams)
  - content
  - metadata
- datastream
  - managed directly by a repository
  - managed externally and delivered to the repository as needed

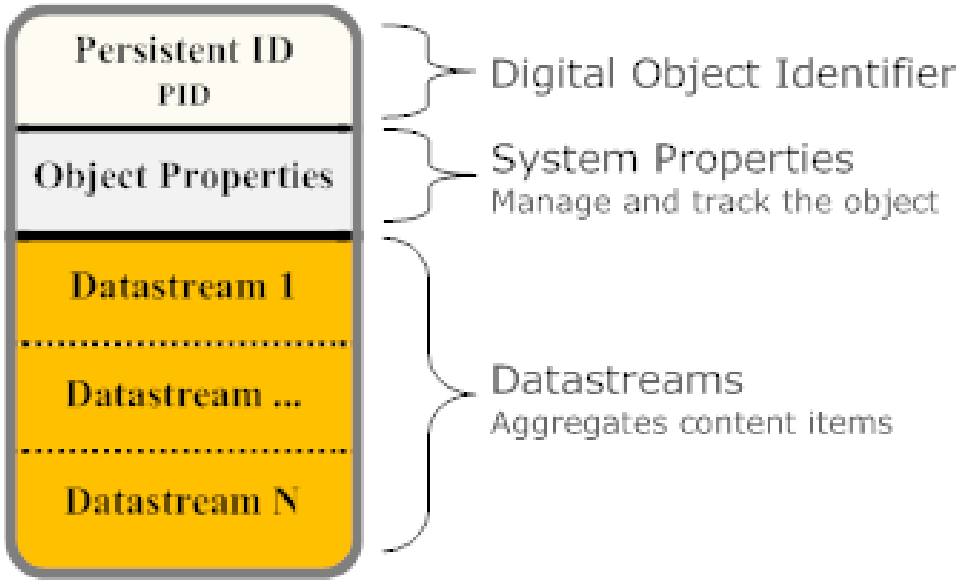
FEDORA basics

- digital object:
  - can have relationships to any number of other objects
    - representing complex information
  - represented by an XML file which is managed in the file system
    - the whole repository can be reconstructed from the XML files
- definition for different views of the digital objects
  - virtual datastreams
  - object behaviours

Fedora Digital Object Model

- fundamental building block
- generic digital object model
  - images
  - documents
  - multimedia
  - any other complex object
- uses “compound digital object” design
  - aggregates one or more content items in the same digital object
- content
  - stored locally
  - stored externally and referenced by the digital object
- defined in XML schema language (http://www.fedora.info/definitions/1/0/foxml1-0.xsd)

Fedora Digital Object Model

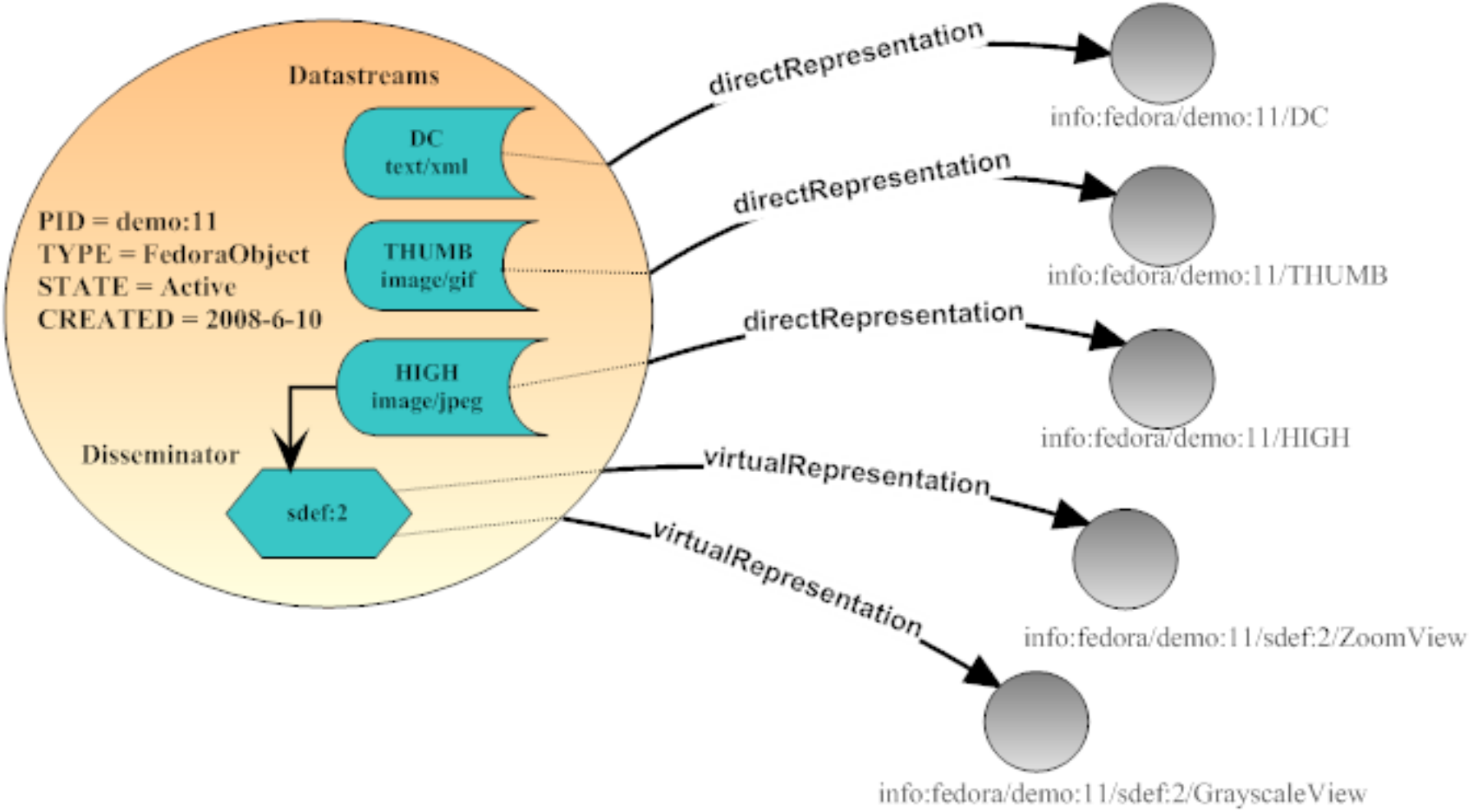


- **PID**: a persistent unique identifier for the object
- **Object Properties**: descriptive properties that are necessary to track the object in a repository
- **Datastream(s)**: element which represents a content item

Datastreams

- represent a content item
- records useful attributes
  - MIME-type
- treated as a raw bitstream
  - user decides how to interpret it
- datastream identifier
  - unique within the digital object scope
  - 4 reserved identifiers
    - DC (DoublinCore)
    - AUDIT (records all changes made to an object)
    - RELS-EXT (digital objects relationships)
    - RELS-INT (datastreams relationships)

Digital Object Model – Access Perspective



Digital Object Model – Access Perspective

- access points for disseminating a representation of the digital object

- two types :
  - direct representation
    - deliver the bitstream
  - virtual representation

Four types of Fedora Digital Objects

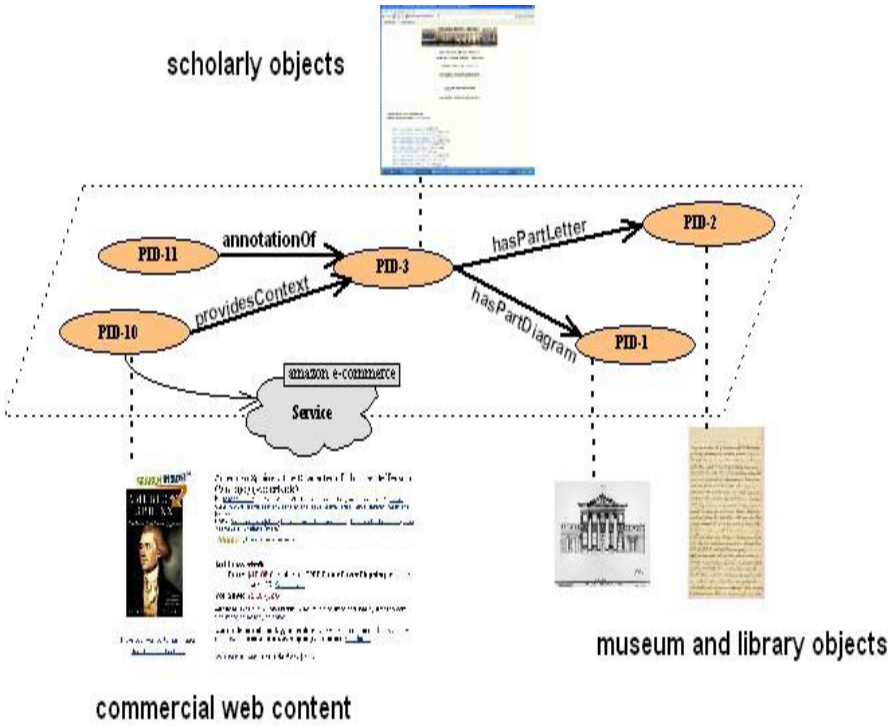
- **Data Object** :
  - represent a digital content entity
  - simplest and most common
- **Service Definition Object**
  - control object
  - store a model of a Service
    - set of Operations that a Data Object supports
  - does not define how Operations are performed

Four types of Fedora Digital Objects

- **Service Deployment Object**
  - control object
  - describes how a specific repository will deliver the Service Operations described in a Service Definition Object
- **Content Model**
  - control object introduced as a part of Content Model Architecture
  - model that characterizes a class of digital objects
    - which operations are permitted, excluded, required ...

Digital Object Relationships

- object can be related to other objects
  - collections
  - derived objects
  - descriptions
  - ...
- Digital Object Relationships is a way how to express those relationships



Digital Object Relationships

- organize objects in collections
- define bibliographic relationships among objects
- define semantic relationships among resources
- model a network overlay
- encode natural hierarchies of objects
- make cross-collection linkage

Digital Object Relationships

- Storing relationships
  - datastreams
    - Relationship-External (RELS-EXT)
      - digital object relationships
    - Relationship-Internal (RELS-INT)
      - relationship from datastreams

- Fedora automatically indexes the RELS-EXT and RELS-INT for all objects
  - unified graph
  - can be queried using SPARQL

Example of relations

```
<rdf:Description rdf:about="info:fedora/demo:99">
  <fedora:isMemberOfCollection rdf:resource="info:fedora/demo:c1"/>
  <myns:isPartOf rdf:resource="info:fedora/mystuff:100"/>
  <myns:owner>Jane Doe</myns:owner>
</rdf:Description>
```

RELS-EXT

```
<rdf:Description rdf:about="info:fedora/demo:99/Thumbnail">
  <myns:isThumbnailOf rdf:resource="info:fedora/demo:99/FullSizeImage"/>
</rdf:Description>
<rdf:Description rdf:about="info:fedora/demo:99/FullSizeImage">
  <myns:hasImageSize>1600 x 900</myns:hasImageSize>
</rdf:Description>
```

RELS-INT

Content Model Architecture

- working with digital content results in developing “patterns of expression”
  - books
  - journals
  - articles
  - collections
  - ...
- reduce the effort to
  - create, capture, ingest, store, manage, access

Content Model Architecture

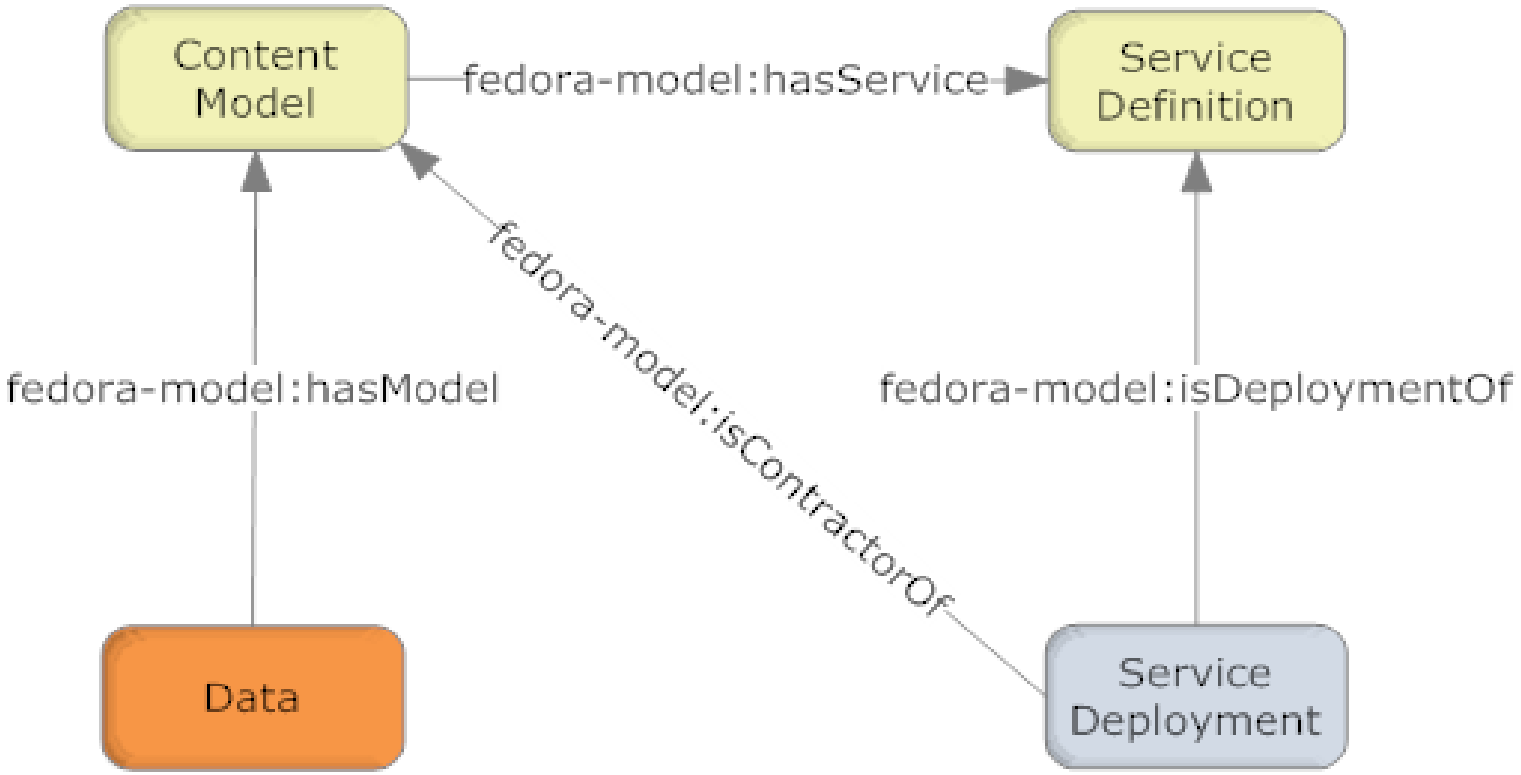
- Content Model
  - content structure as used by publishers and other traditional content – related professions
  - a computer model describing an information representation and processing architecture

Content Model Architecture

- Content Model Architecture
  - integrated structure for persisting and delivering characteristics of digital objects

- Two approaches to content models
  - complex single-object models (compound)
  - multi-object models (atomistic or linked)

Content Model Architecture

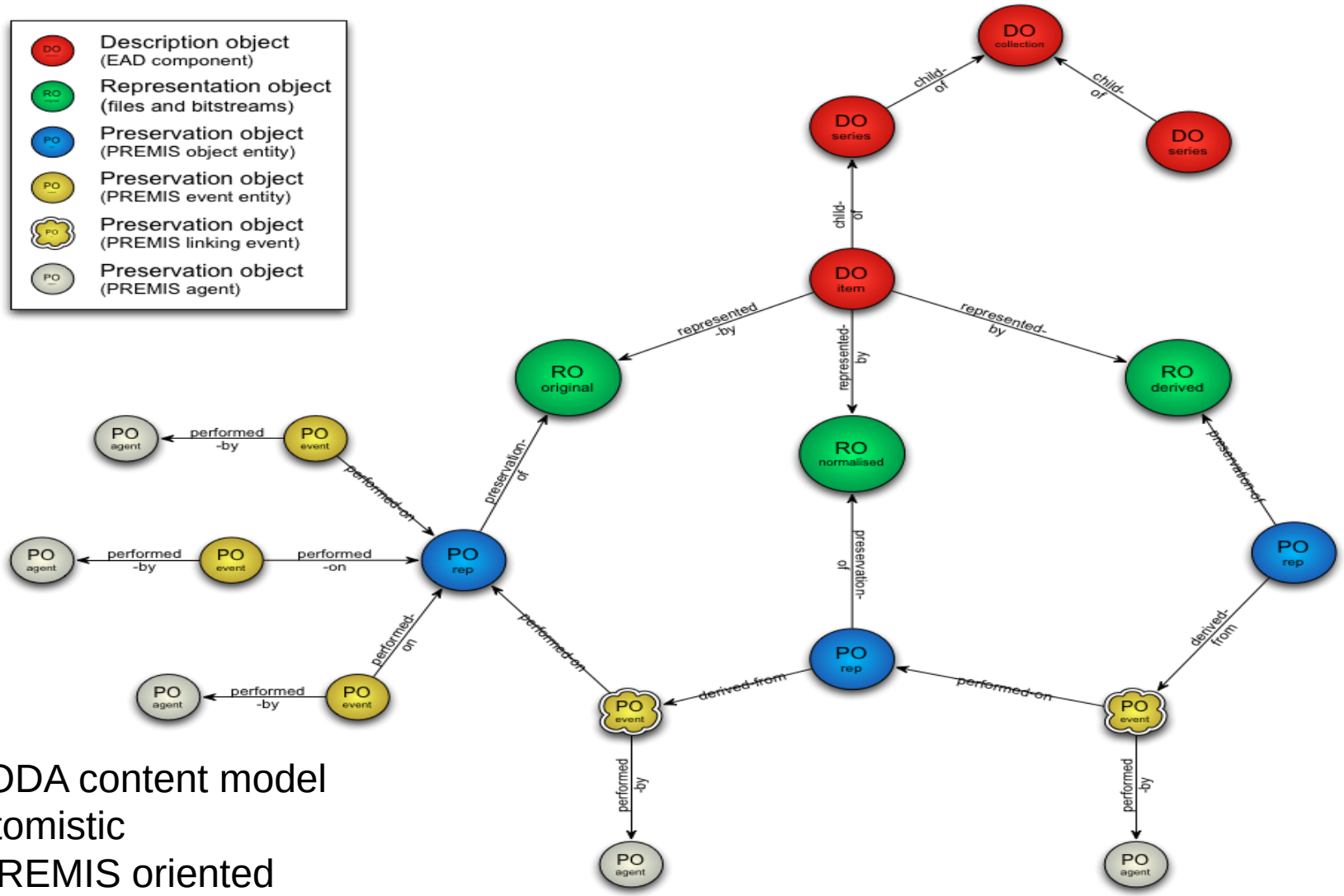


- reference model
- can be changed and adapted

Fedora based implementations

- Fedora is not a final product !
- It needs to be adapted to special use cases
- Many different Fedora based implementation already out there
  - ESciDoc (https://www.escidoc.org/)
    - eResearch environment with focus on scientific communities
  - Hydra (http://projecthydra.org/)
    - Ruby on Rails framework for building digital asset management applications
  - Islandora (http://islandora.ca/)
  - RODA (http://www.roda-community.org/)
    - OAIS compliant service oriented digital repository system for preserving government data

RODA



RODA content model
- atomistic
- PREMIS oriented

Other solutions

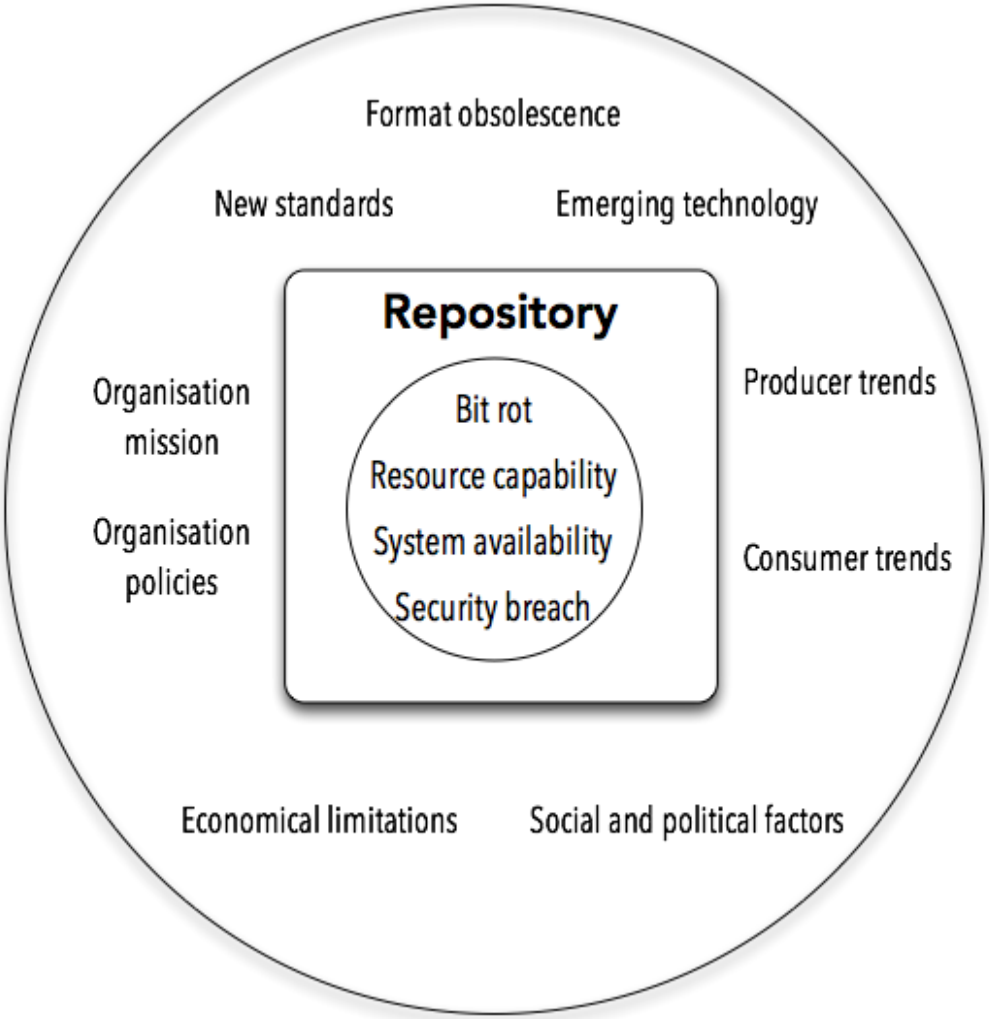
- Fedora and Fedora based repositories are only one possible solution

- There are many other implementations
  - Based on micro-services
    - Archivematica (https://www.archivematica.org)
    - IRods (http://irods.org/)
  - Commercial solutions
    - Tessella (http://tessella.com/)
    - exLibris (http://www.exlibris.co.il/)

Systems and tools

- Preservation monitoring
- Preservation policies
- Preservation planning
- Bringing everything together

Why do we need monitoring ?



RISKS

OPPORTUNITIES

What do we have ?

- Digital Format Registries
  - examples
    - PRONOM (http://www.nationalarchives.gov.uk/PRONOM/Default.aspx)
    - UDFR (http://www.udfr.org/)
  - Lack of coverage
  - Statically-defined generic risks
  - Lack of structure in risks
  - Focus on format obsolescence
- Tools for monitoring
  - Total dependency on format registries
- Technology watch reports
  - Machine unreadable

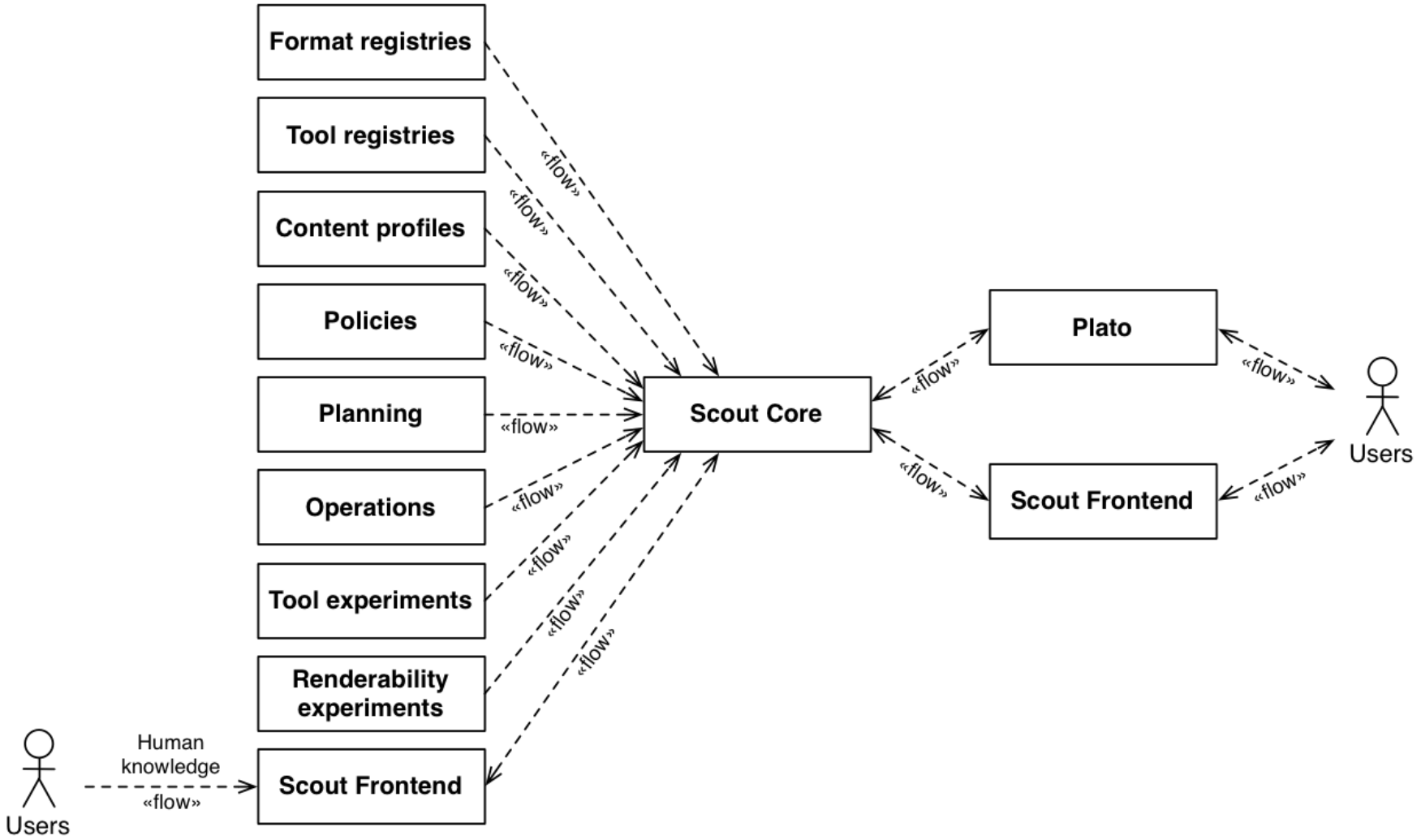
Scout – Preservation Monitoring System

- open source preservation monitoring system
- https://github.com/openplanets/scout

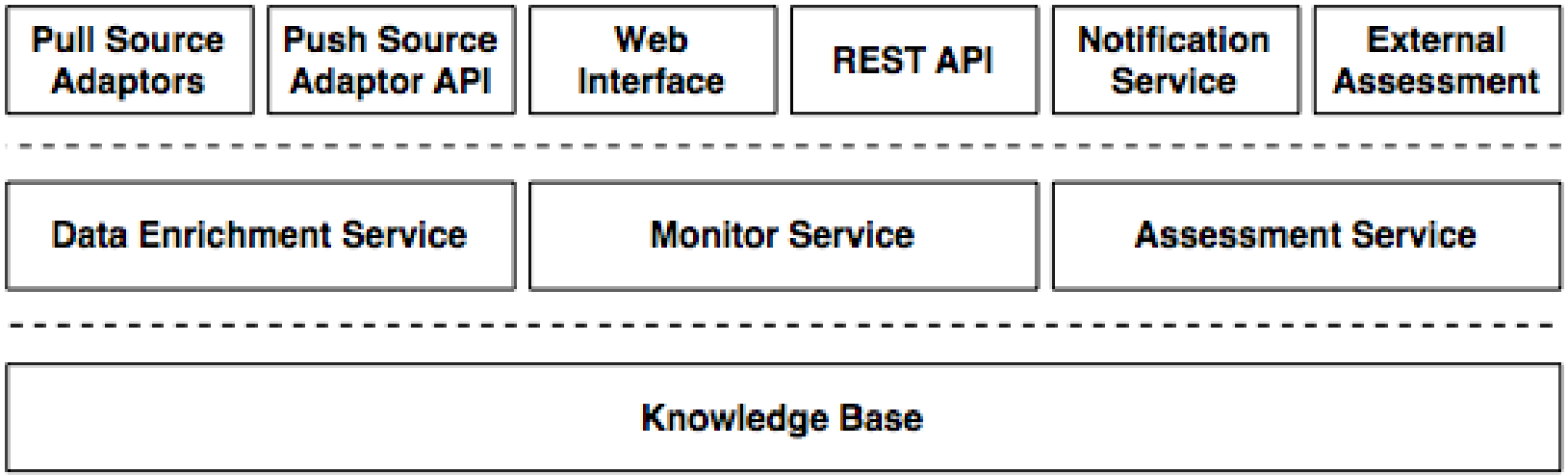
- developed under the SCAPE project (http://www.scape-project.eu/)

- developed by Technical University of Vienna and KEEP Solutions (Portugal)

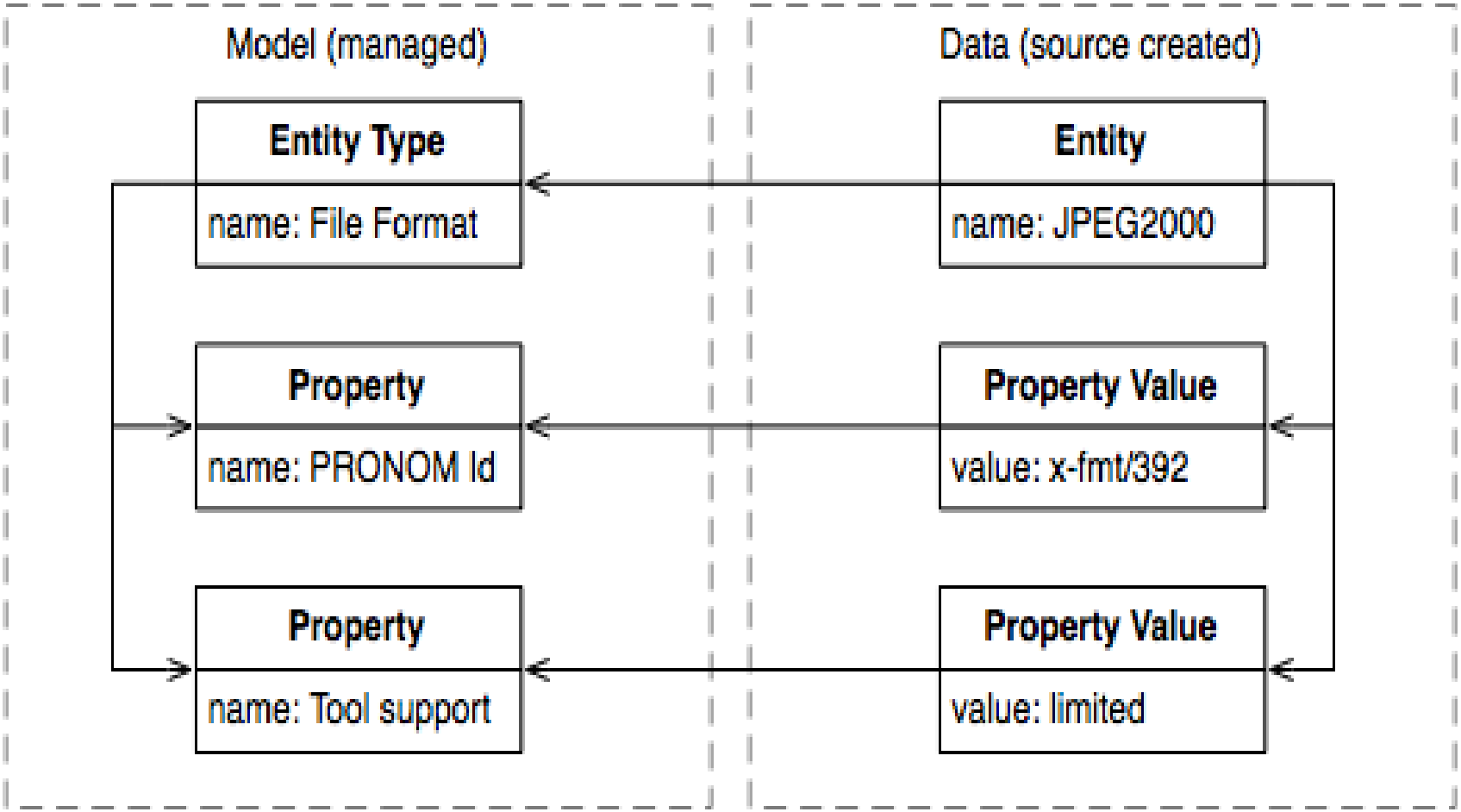
Scout – information flow



Scout – architecture



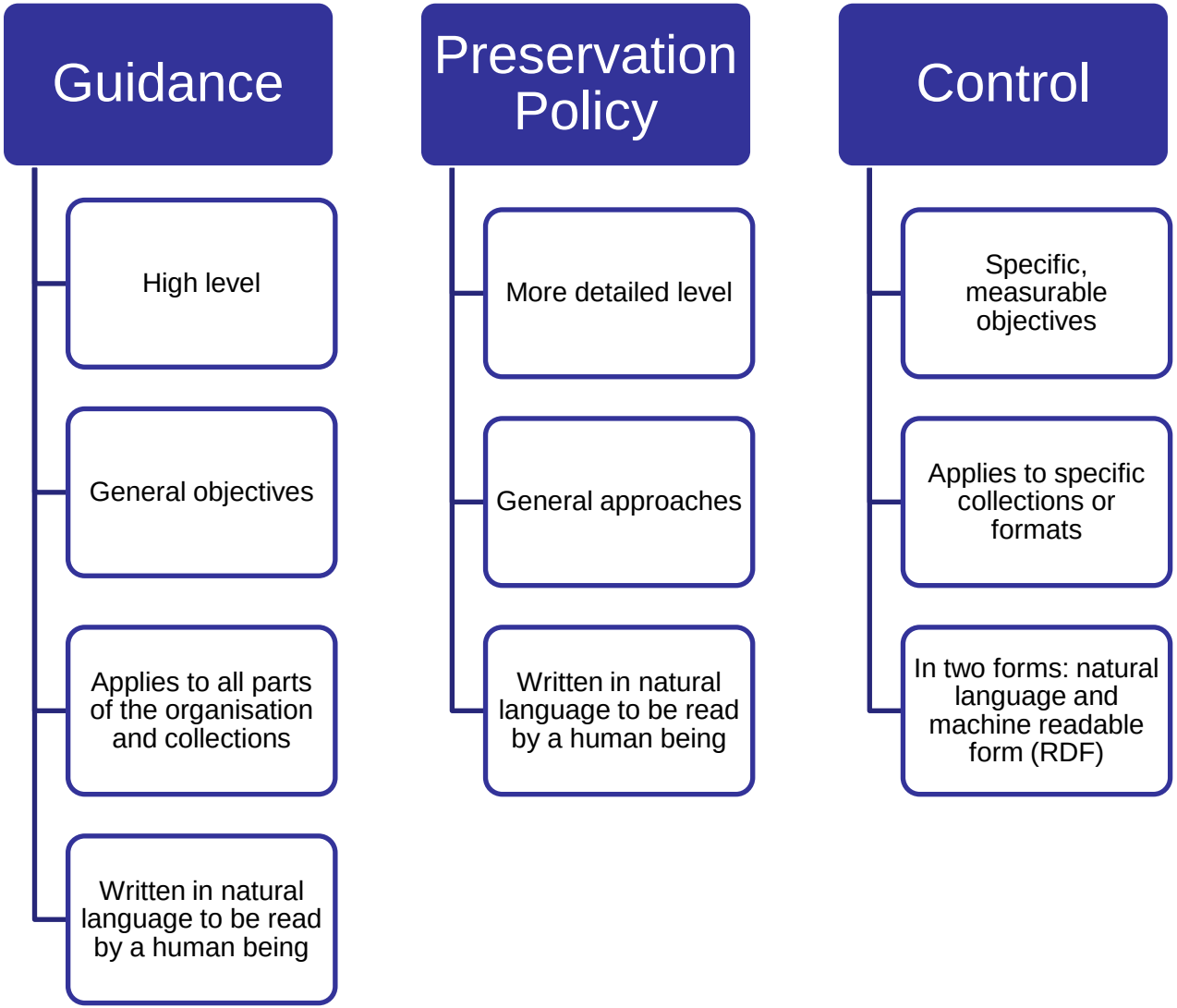
Scout data model – knowledge base



Business policies

- Business Policies exist to govern; that is, control, guide, and shape the Strategies and Tactics. They define what can be done and what must not be done, and may indicate how, or set limits on how, it should be done (Object Management Group (2010). *The Business Motivation Model v1.1)*

SCAPE preservation policy levels



Guidance policy

The State and University Library aims to preserve the digital collections in data formats suitable for digital preservation.

The library maintains a list of suitable data formats and cooperates with other institutions about announcing suitable data formats and maintaining information about these suitable formats.

Taken from : http://en.statsbiblioteket.dk/about-the-library/ddpolicy

- High level defined on the management level of an organization
- Written in a natural language
- They cover
  - Preservation goals and strategies of an organization
  - Designated communities
  - Digital objects
  - Metadata
  - Authenticity
  - Rights
  - Standards
  - Organisation
  - Storage

Procedure policy

3.2.1 All raw data will be curated in well-defined formats, for which the means of reading the data will be made available by the Facility

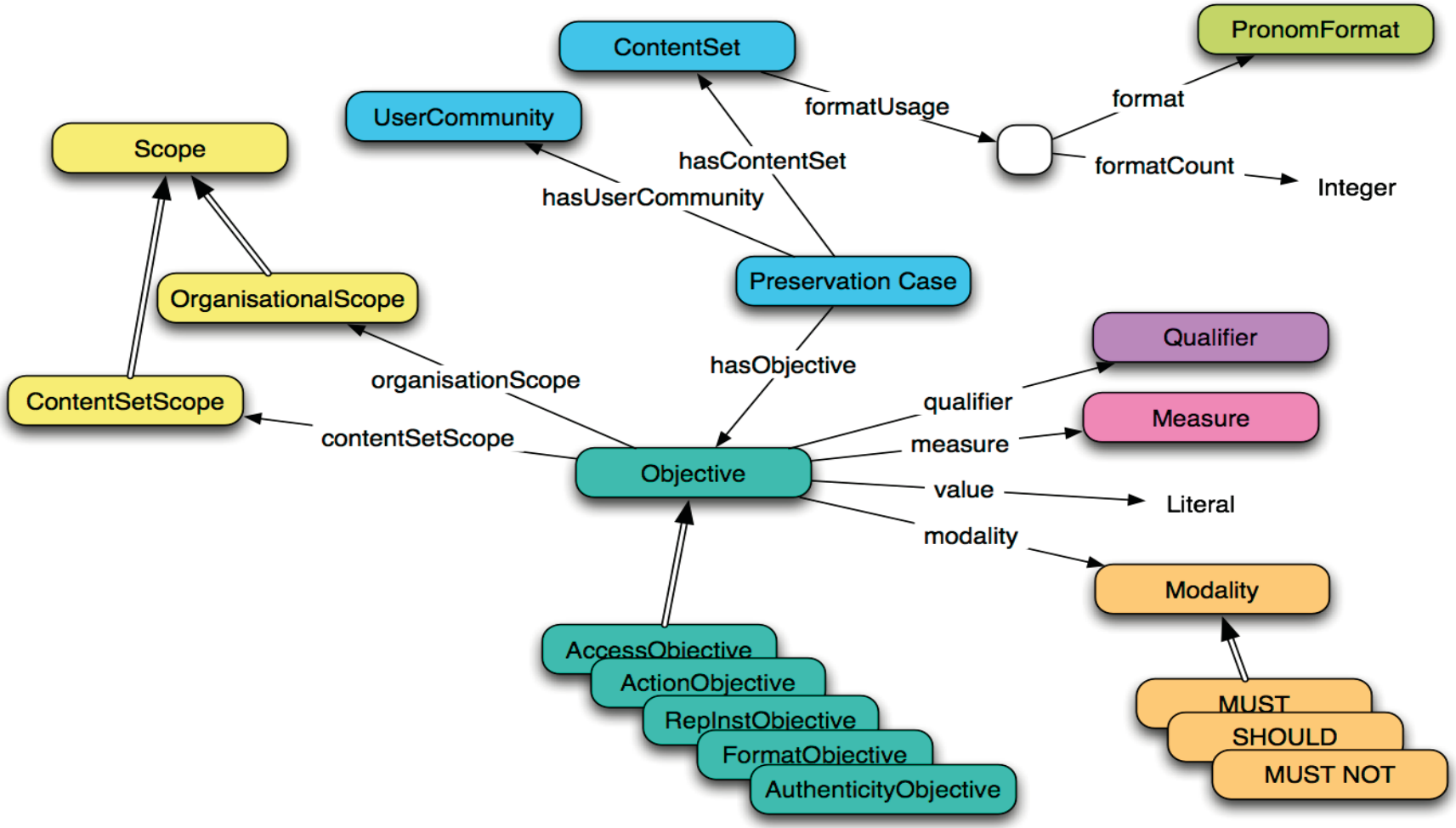
http://www.isis.stfc.ac.uk/user-office/data-policy11204.html

- Natural language human readable elements
- They can focus on the whole organization or might cover only a particular collection or material type

Control policy

- Control policies : practicable elements of governance that relate to clearly identified entities in a specified domain model (Kulovits et al. Open Preservation Data: Controlled vocabularies and ontologies for preservation ecosystems)

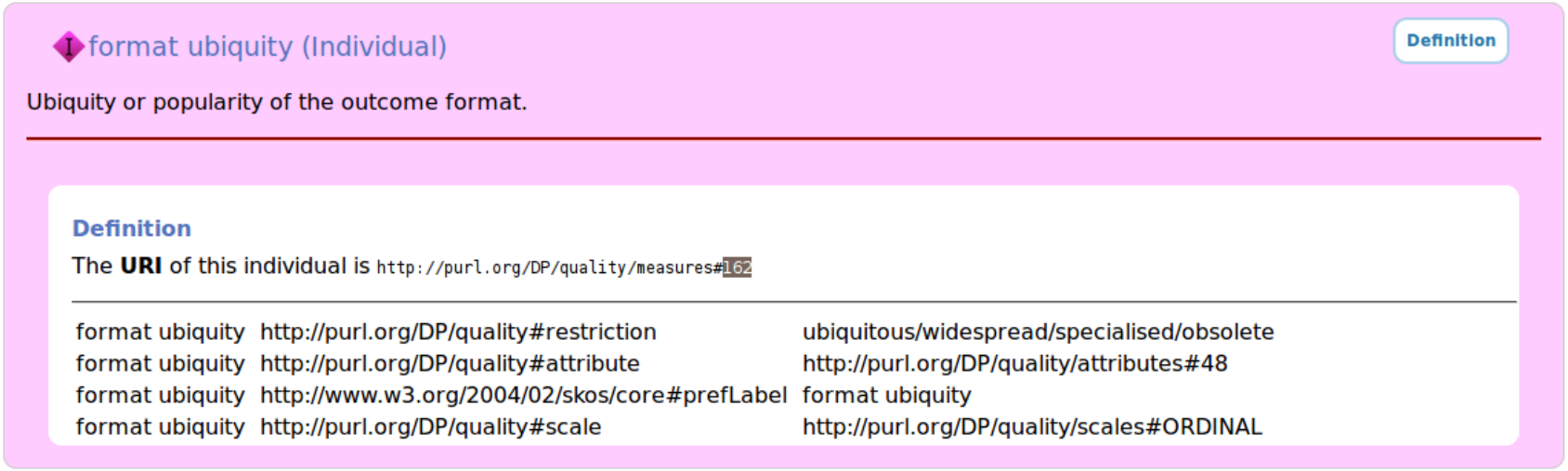
Control policies



(Kulovits et al. Open Preservation Data: Controlled vocabularies and ontologies for preservation ecosystems)

Vocabulary

- properly defined measures (http://purl.org/DP/quality/measures)
- each measure has URI
  - it can be linked
- around 400 measures defined

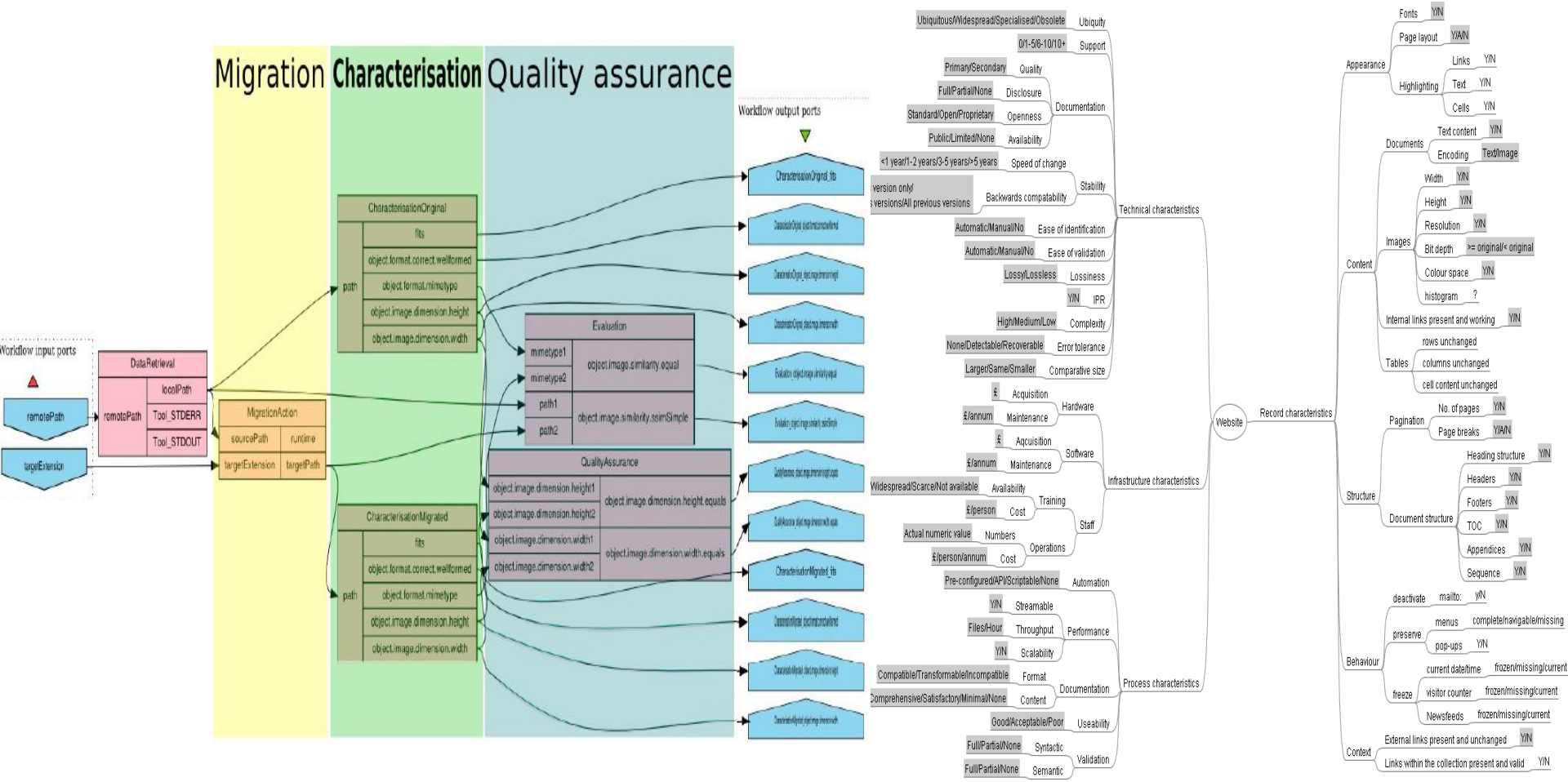


Control Policy

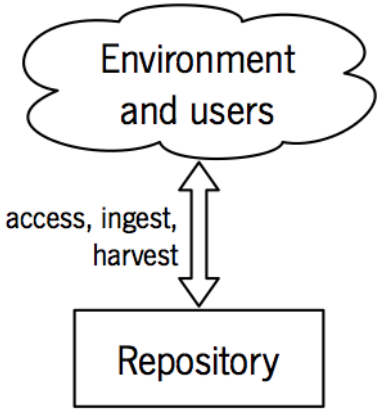
```
<http://www.oesta.gv.at/policies/UseOfFormatShouldBeWidespread>
    control-policy:measure <http://purl.org/DP/quality/measures#162> ;
    control-policy:modality modalities:SHOULD ;
    control-policy:value "widespread"^^xsd:string ;
    a control-policy:FormatObjective .
```

- Control policy UseOfFormatShouldBeWidespread
- Measure : format ubiquity
- Modality : SHOULD
- Value: widespread
- Is type of FormatObjective

Preservation planning (Plato)



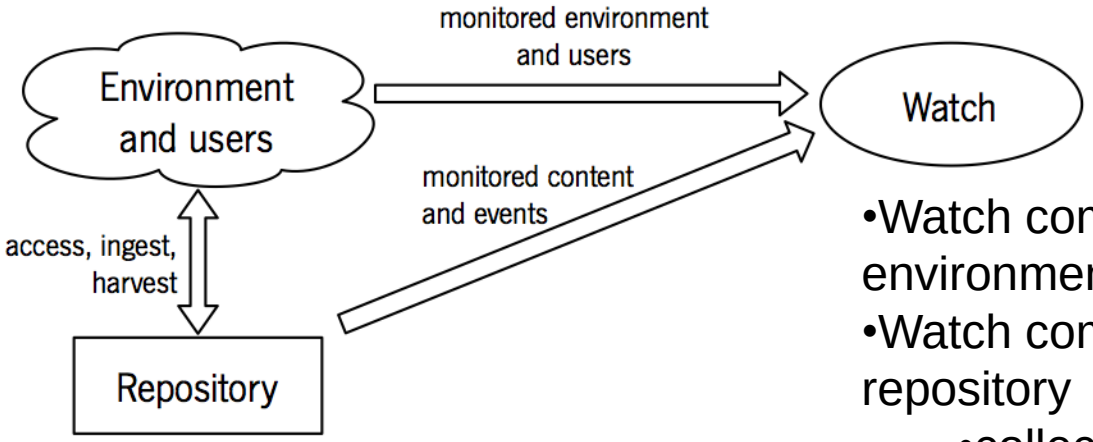
Bringing everything together



- repository is in an environment
- users create and store new content and access already stored

Author of the diagram Luis Faria

Bringing everything together

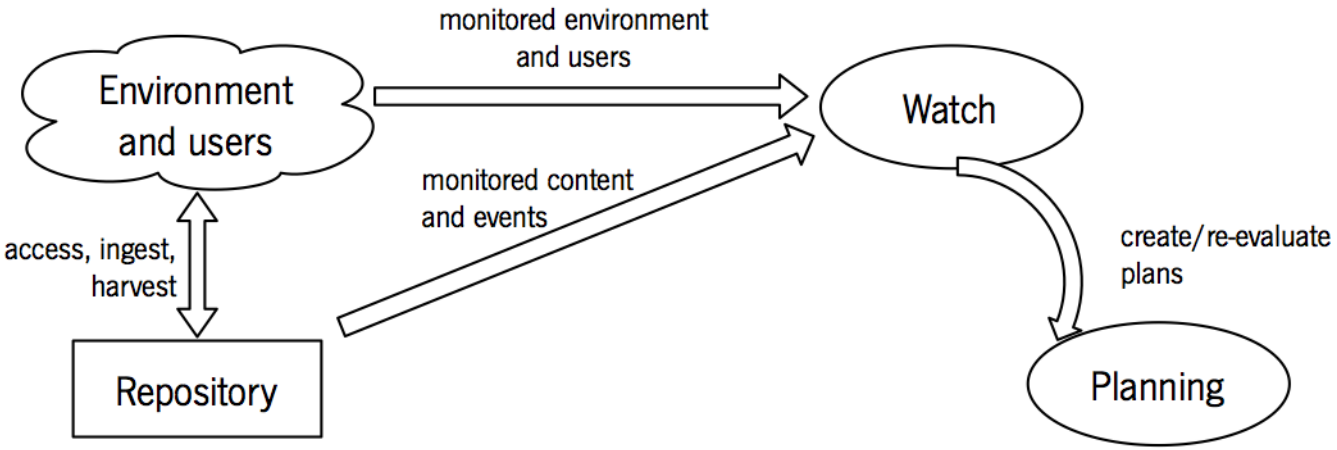


- Watch component monitors environment
- Watch component monitors repository
  - collection profiles
  - repository events

Author of the diagram Luis Faria

Bringing everything together

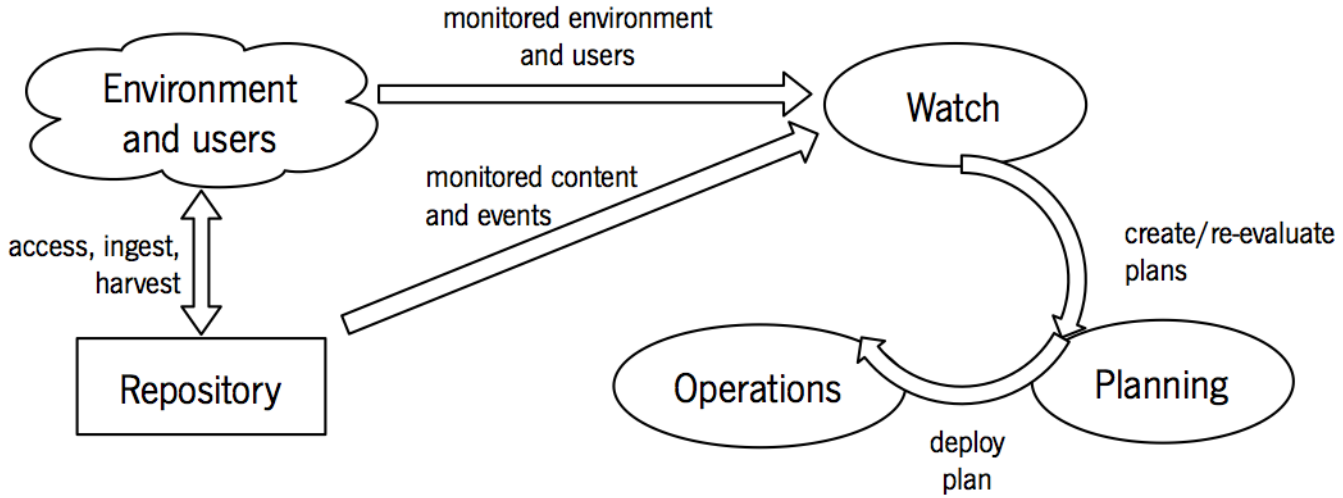
- Watch component notifies Planning about detected risk or opportunity
- Planner starts a preservation planning process and creates a preservation plan



Author of the diagram Luis Faria

Bringing everything together

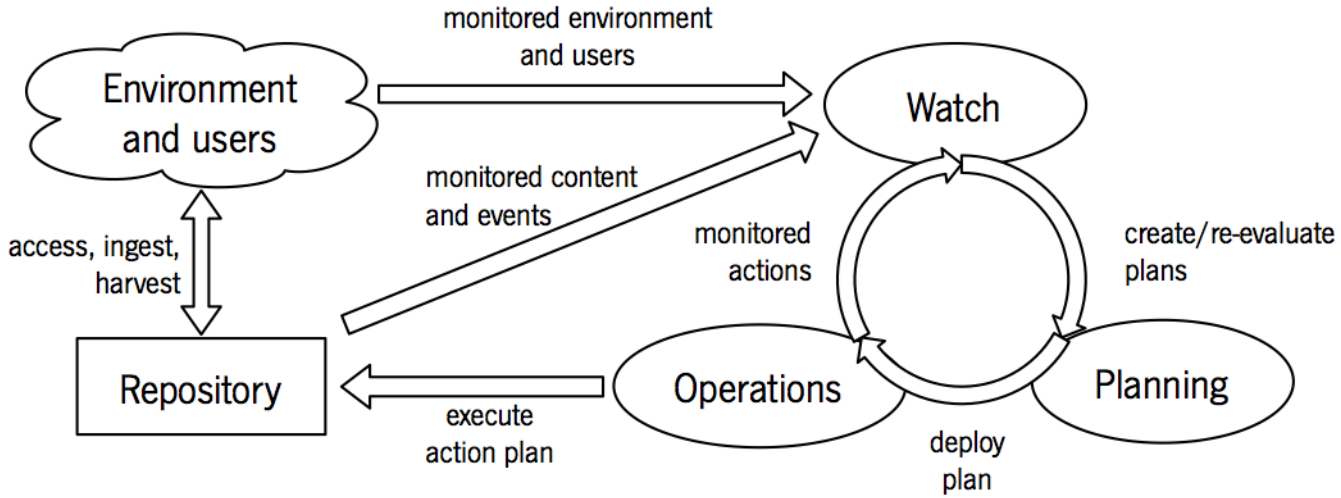
•created plan is deployed to operations



Author of the diagram Luis Faria

Bringing everything together

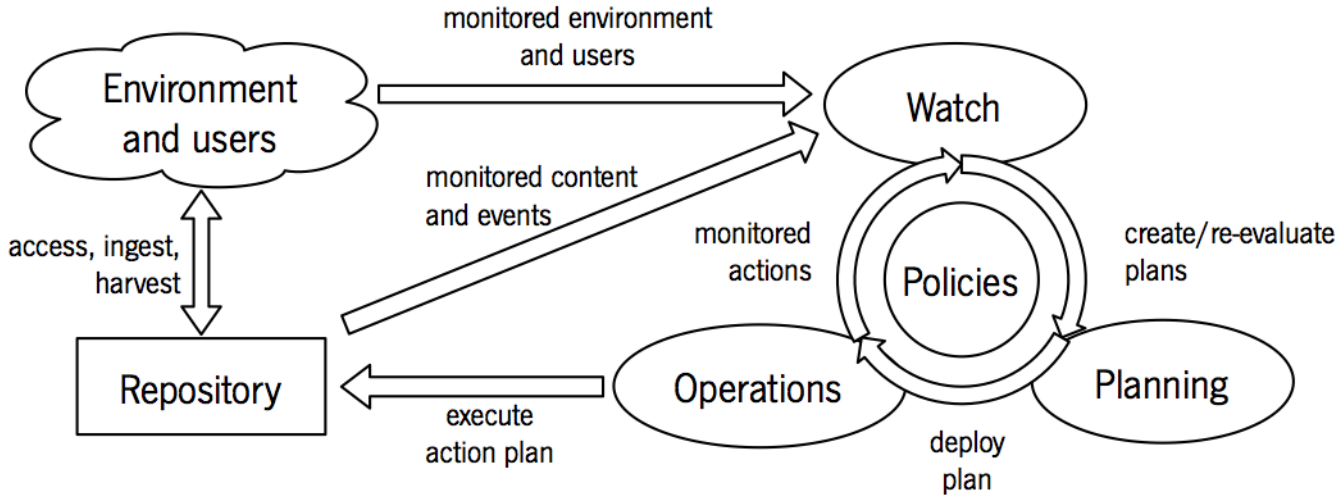
- operations execute the plan
- execution generates events
- events (repository events)
are monitored by the wacth component



Author of the diagram Luis Faria

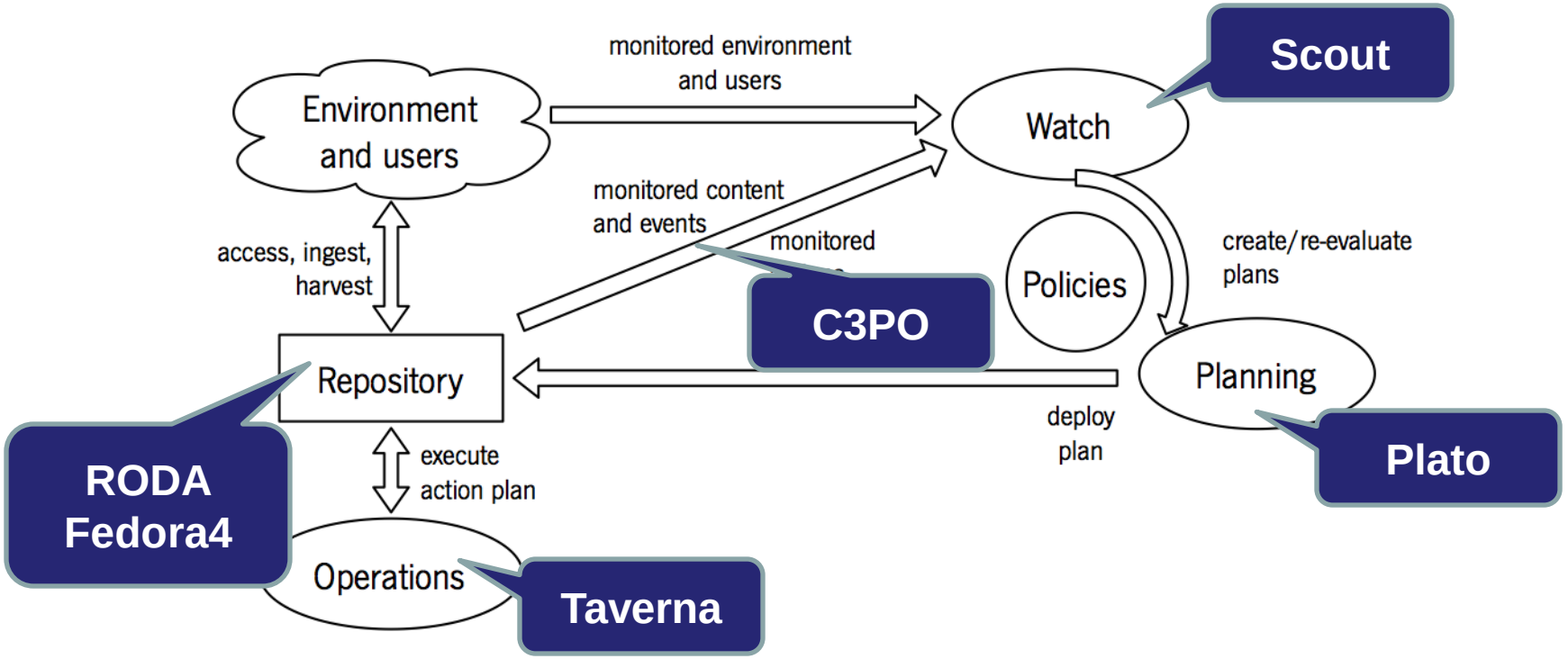
Bringing everything together

- everything is regulated by policies
- control policies



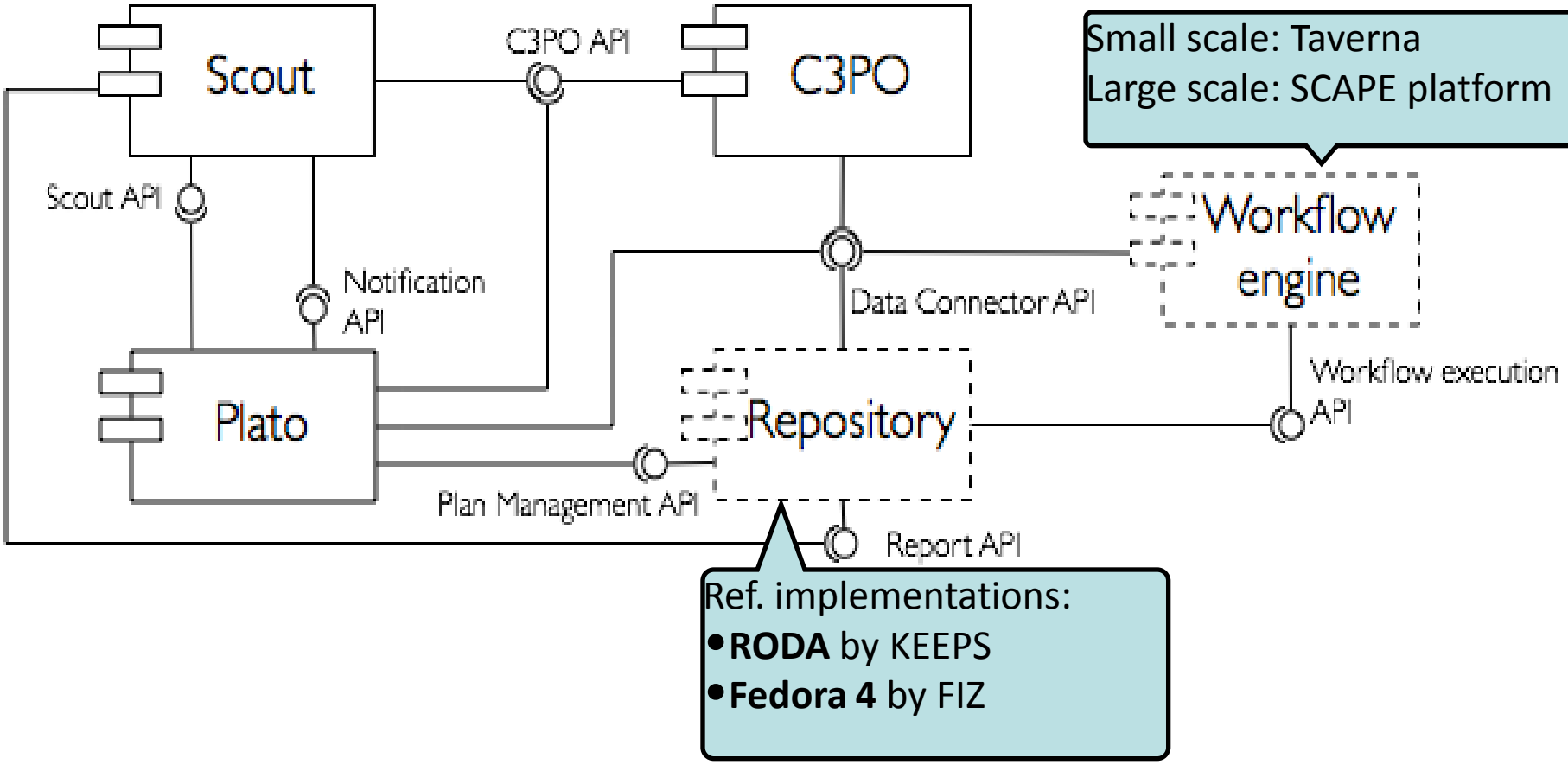
Author of the diagram Luis Faria

Bringing everything together



Author of the diagram Luis Faria

Technical implementation



Author of the diagram Luis Faria

Next lecture

- 2. June 2014 16:00 – 18.00 CET
- two parts
  - real world example from the Austrian State Archive
  - experiences in preserving audio visual content
    - www.mediathek.at
- these lectures will be included in the exam