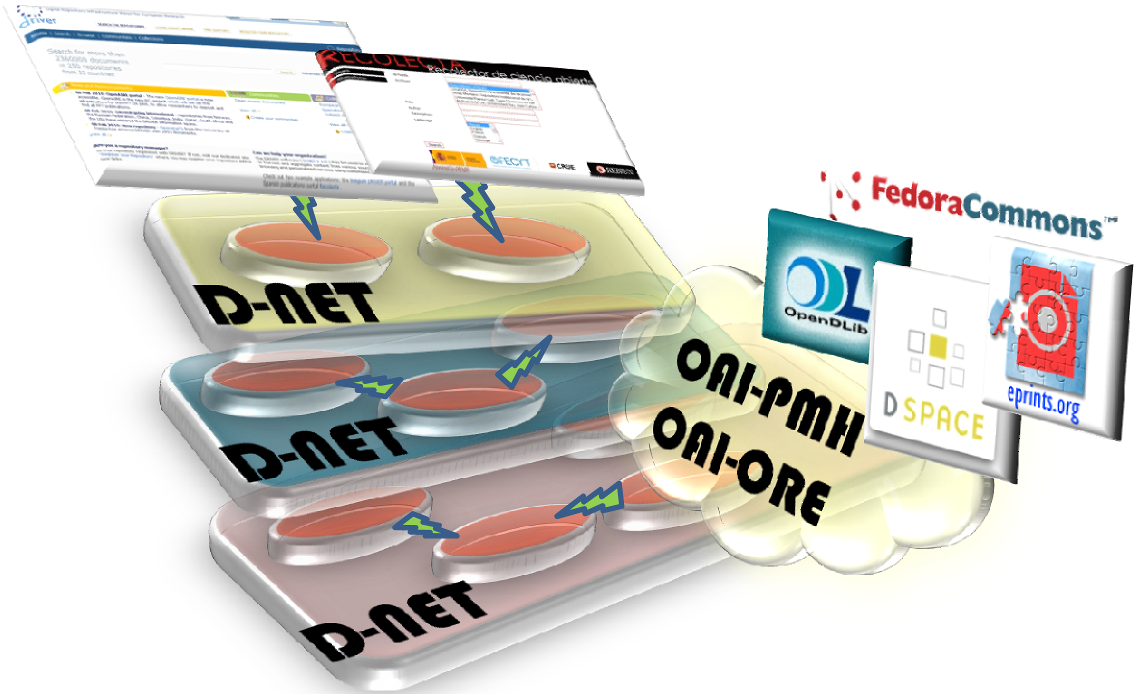
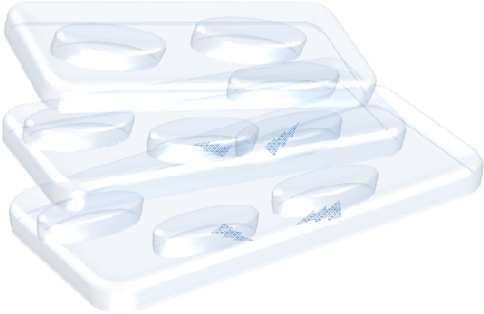
Interoperability Patterns in Digital Library Systems Federations

Paolo Manghi, Leonardo Candela, Pasquale Pagano

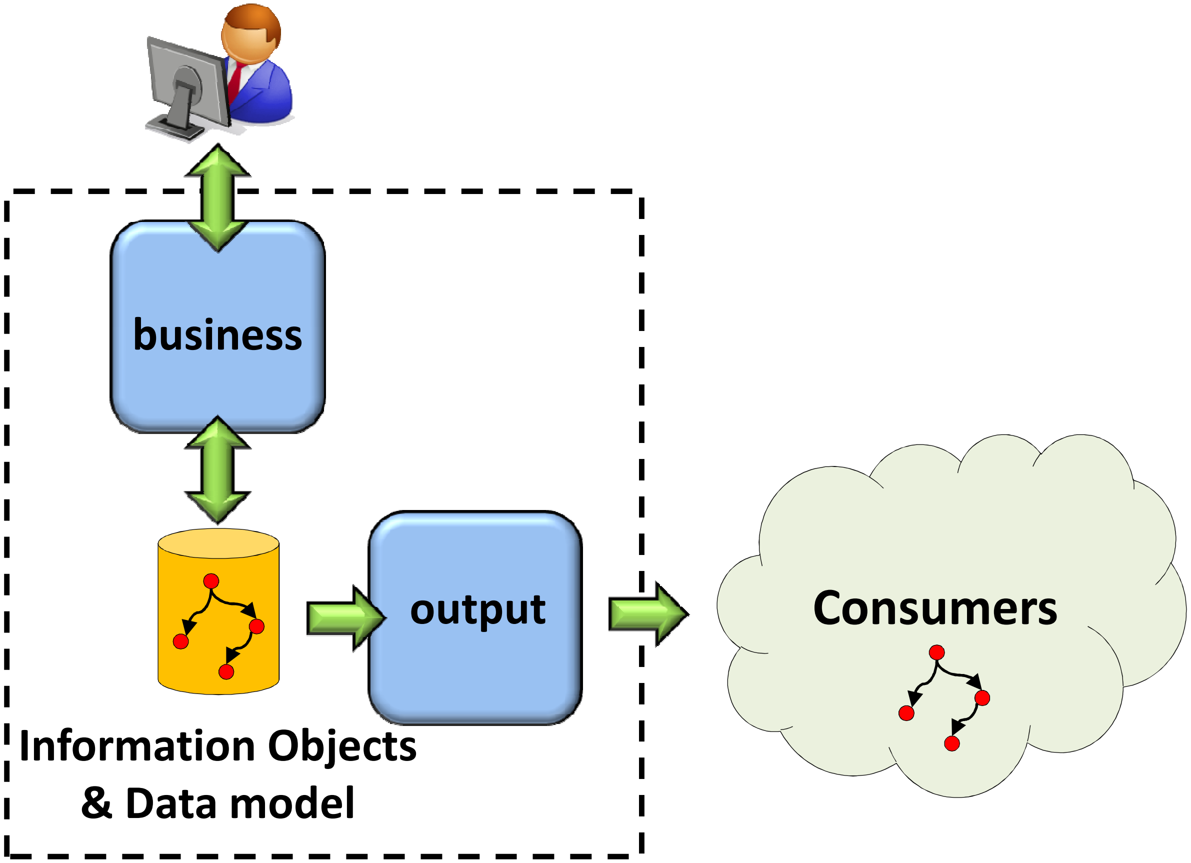




Outline

- Digital Library System Federations
- Interoperability issues
- Data impedance mismatch
    – Structural, semantic and granularity mismatch
- Solution: D-NET Software Toolkit

Digital Library Systems



Digital Library Systems Federations (DLSFs)

- Motivations
  - On-line availability of “fragmented” research outcomes
  - Multidisciplinary character of modern research
  - Increased speed of research life-cycle, i.e., immediate availability and access to research outcome
  - Others…

DLSFs

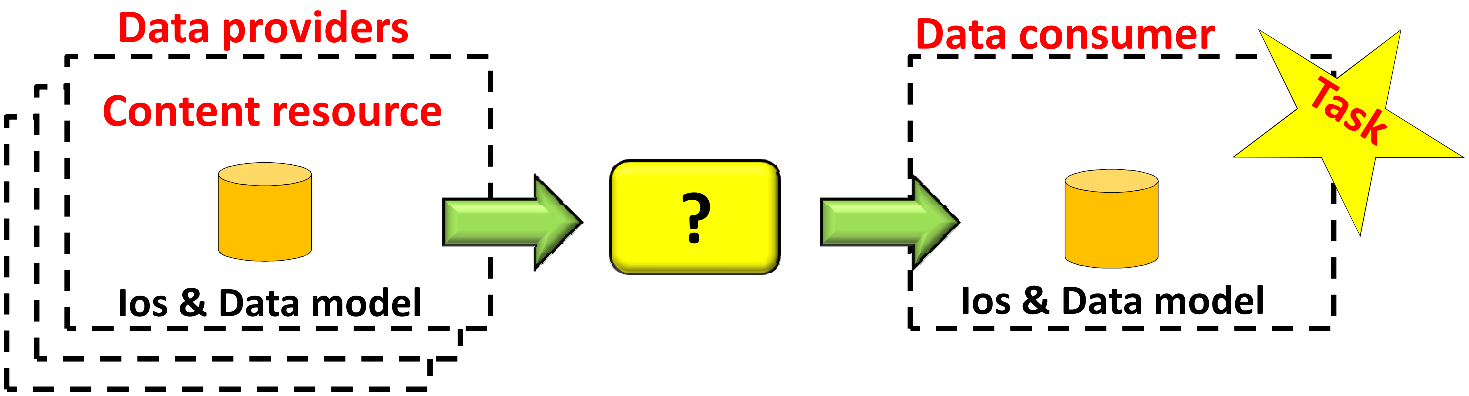
- OAI-PMH archive/libraries/repository federations
  - e.g., Europeana, OCLC-OAIster, BASE, NARCIS
- Community-oriented data infrastructures
  - e.g., DRIVER, SAPIR, CLARIN, EFG, HOPE, D4Science

DLSFs and the DL.org interoperability framework

- **Providers** = *Digital Library Systems or Data Providers*
- **Consumer** = *Service Provider, s*oftware system specially devised for
  - Collecting input **content resources** (information objects, e.g., metadata, payloads, compound objects) from a set of data providers
  - From input information objects, producing a uniform “information space” of output information objects, required by the consumer to perform a given **task**

DLSFs and the DL.org interoperability framework

- **Providers** = *Digital Library Systems or Data Providers*
- **Consumer** = *Service Provider*



Data providers

Content resource

Ios & Data model

?

Data consumer

Task

Ios & Data model

DLSFs: content interoperability

- “Obstacles” encountered by a **data provider** (**DLS)** willing to offer useful **information objects** to a service provider to accomplish its **task**
- “Obstacles” encountered by a **service provider** willing to accomplish its **task** by accessing the **information objects** of a **data provider** which it considers useful

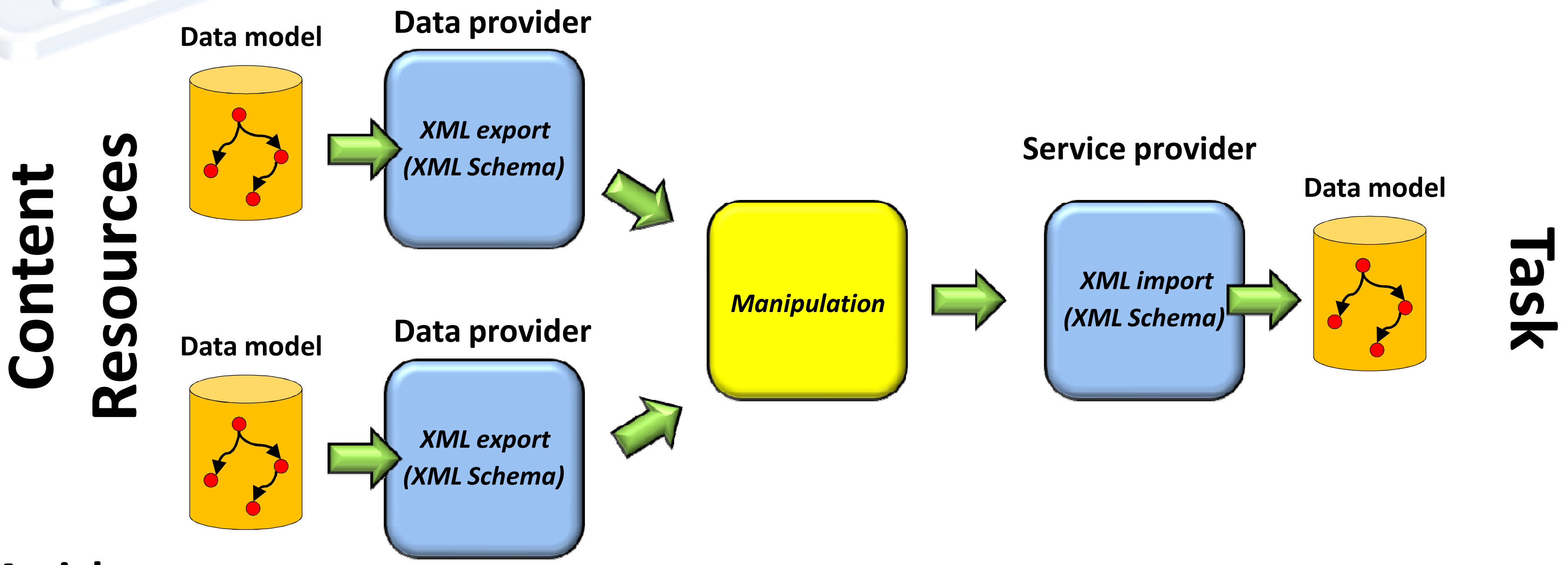
DLSFs: content interoperability issues

- Low-level issues: “How to exchange objects”
  - Identifying common **on-the-wire data-exchange practices**
- High-level issues: “How to harmonize information objects data models”
  - Resolve **data impedance mismatch** problems arising from distinct data models of data and service providers

Low-level issues: “How to exchange information objects”

- Adoption of XML as lingua-franca and standard data-exchange protocols, e.g., OAI-PMH, OAI-ORE, ODBC
  - XML schema for data model
  - Data providers implement *exporting components*: information objects → XML files
  - Service provider implement *importing component*: XML files → information objects
- Worth noticing:
  - Equal data models does not mean equal XML schemas
  - Data and service providers may manage information objects as XML files (e.g., native XML DBs)

High-level issues: data impedance mismatch



Content Resources

Data model

Data provider

XML export (XML Schema)

Data model

Data provider

XML export (XML Schema)

Manipulation

Service provider

XML import (XML Schema)

Data model

Task

```
<Article>
        <Title> “Interoperabilty patterns…”
        </Title>
        <Authors> “Paolo Manghi, Leonardo
        Candela…”
        </Authors>
        <Date > “September 2010”
        </Date>
</Article>
```

Definitions:

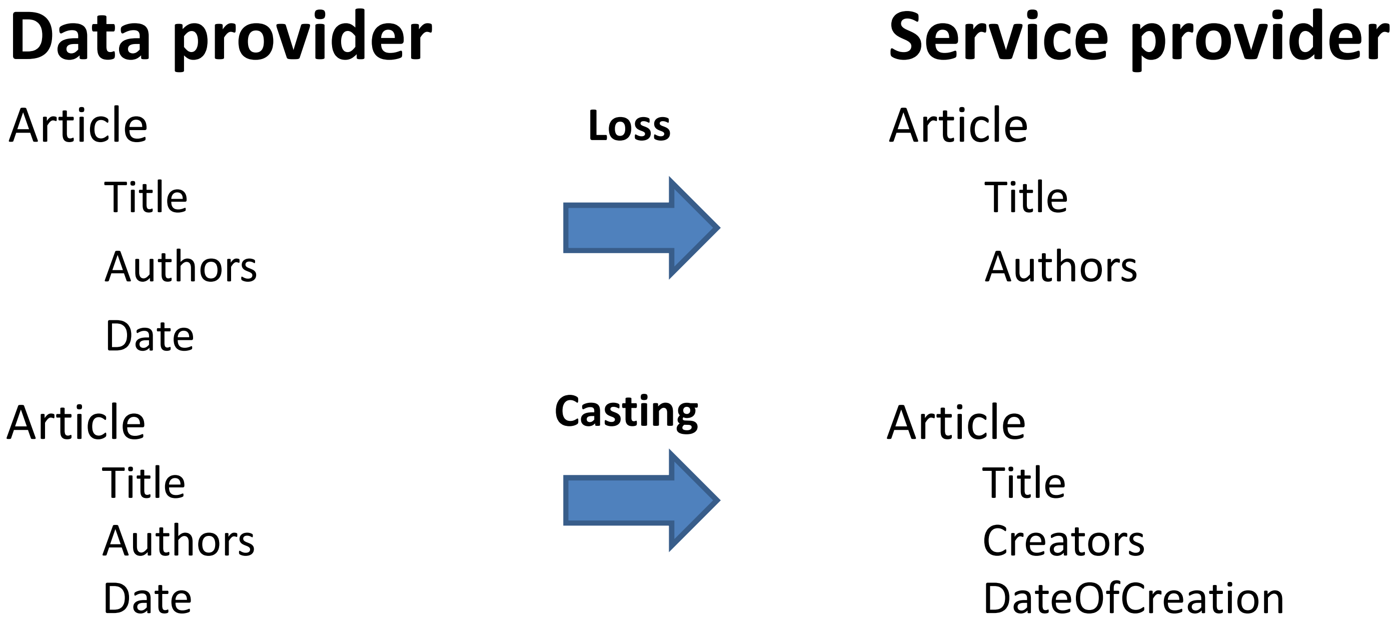
- Schema *path*: **Article.Title**
- Schema *leaf*: **“September 2010”**

High-level issues: data impedance mismatch

- *Data model impedance mismatch*
  - Data and service providers XML schemas do not match, either **structurally** (schema paths) or **semantically** (schema leaves)
- *Granularity impedance mismatch*
  - XML encodings of information objects at the service provider and data providers adopt different levels of granularity.

Structural heterogeneity

(Data model impedance mismatch)



Data provider		Service provider
Article Title Authors Date	**Loss** →	Article Title Authors
Article Title Authors Date	**Casting** →	Article Title Creators DateOfCreation

Semantic heterogeneity

(Data model impedance mismatch)

Data provider

Article

Title “Interoperability…”
Authors “Paolo Manghi, …”
Date “September 2010”

Formats →

Service provider

Article

Title “Interoperability…”
Authors “Manghi, P., …”
Date “01-09-2010”

Data provider

Article

Title “Interoperability…”
Authors “Paolo Manghi, …”
Date “September 2010”

Dervation/ Inference →

Service provider

Article

Title “Interoperability…”
Authors “Paolo Manghi, …”
Date “September 2010”
TitleLanguage “EN”

Semantic&Structural heterogeneity

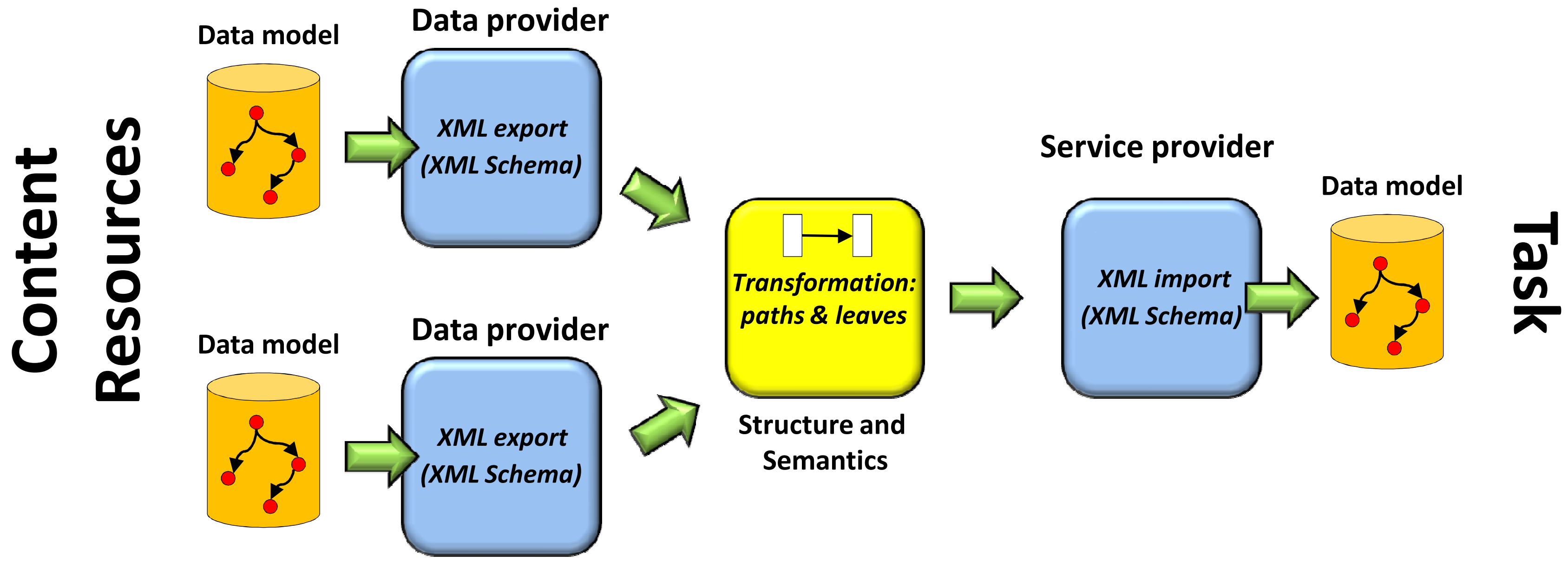
(Data model impedance mismatch)

- Article
  - Title “Interoperability…”
  - Authors “Paolo Manghi, …”
  - Date “September 2010”

→

- Article
  - Title “Interoperability…”
  - Creator
    - Name “Paolo”
    - Surname “Manghi”
  - Creator
    - Name “Leonardo”
    - Surname “Candela”
  - Date “September 2010”

Tackling the data model impedance mismatch: transformation components



Use-cases:

- All data providers have the same XML schema
- Data providers have different XML schemas

Data providers with equal XML schema

(Data model impedance mismatch solutions)

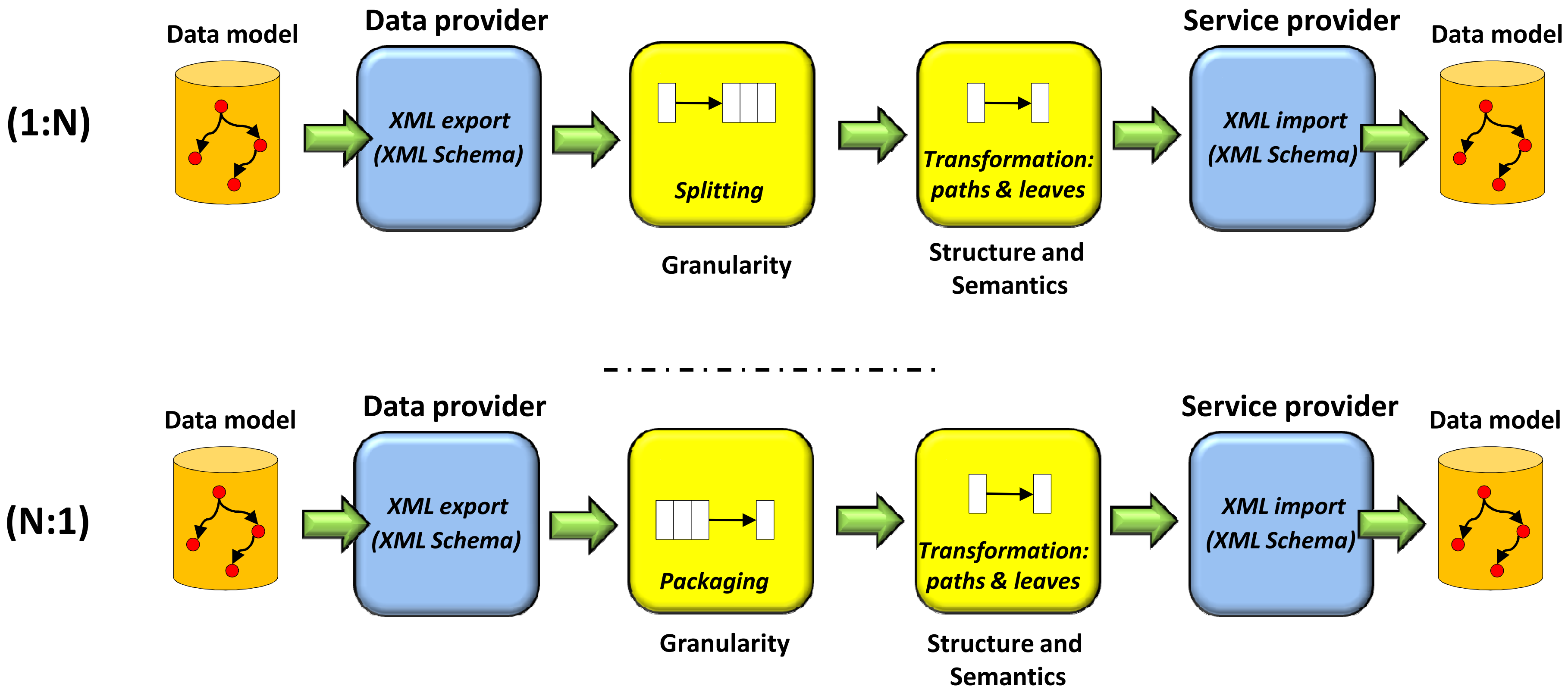
- The transformation component considers **one mapping** from such common XML schema onto the service provider schema
  - Output schema leaves (identified by output schema paths) are generated by processing input leaves (identified by schema paths) through transformation functions *F*
- The complexity of the *F*’s can be arbitrary:
  - *feature extraction* functions: taking a URL, downloading the file (e.g., HTML, PDF, JPG) and returning content extracted from it
  - *conversion* functions: translation from vocabulary to vocabulary
  - *transcoding* functions: leaf format to leaf format (e.g., date formats);
  - *regular expression:* generating one leaf from a set of leaves (e.g., generating a person name leaf by concatenating name and surname originally kept in two distinct leaves).

Data providers with different XML schemas

(Data model impedance mismatch solutions)

- The transformation component must consider **multiple mappings** from the diverse input XML schemas onto the service provider XML schema of the service provider
- Simple scenario: pre-determined set of data providers
  - Providing one transformation component as the one described for the previous scenario for each set of data providers with the same schema
- Complex scenario: undetermined number of data providers is expected, possibly bearing different XML schema
  - Providing general-purpose components, capable of managing (create, remove, update) a set of “mappings”
  - Mappings are named lists of pairs (*input paths, F, output path*)
  - The component may allow for the addition of new *F*’s

Granularity impedance mismatch



(1:N)

Data model

Data provider

XML export (XML Schema)

Splitting

Granularity

Transformation: paths & leaves

Structure and Semantics

Service provider

XML import (XML Schema)

Data model

(N:1)

Data model

Data provider

XML export (XML Schema)

Packaging

Granularity

Transformation: paths & leaves

Structure and Semantics

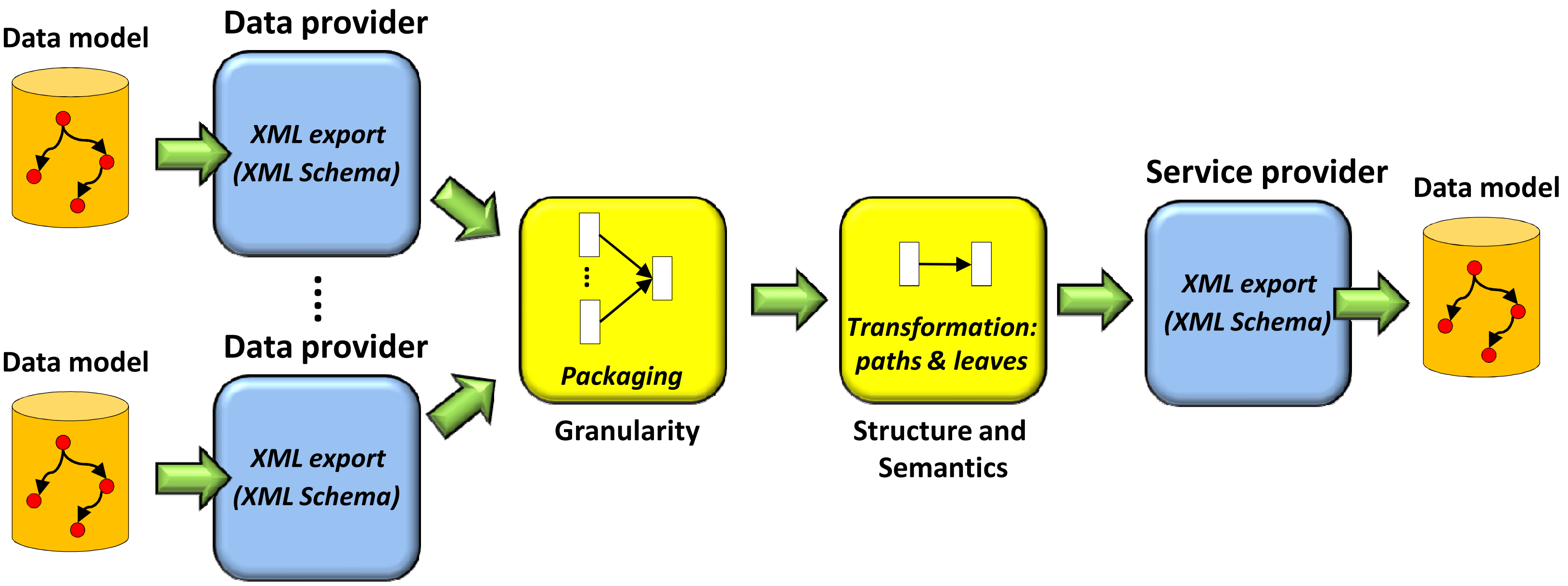
Service provider

XML import (XML Schema)

Data model

Granularity impedance mismatch

(1xM:1)



Data model

Data provider

XML export (XML Schema)

⋮

Data model

Data provider

XML export (XML Schema)

Packaging

Granularity

Transformation: paths & leaves

Structure and Semantics

Service provider

XML export (XML Schema)

Data model

Architecture of interoperability solutions

- "Bottom-up" federations, e.g., DAREnet-NARCIS
  - Realized by organizations who have control over the set of participating data providers,
  - Agree on common data model and XML schema so that no interoperability issues occur
- "Open" federations, e.g., the DRIVER repository infrastructure
  - Federations "attractive" to data providers, which are willing to adhere to given "data model" specifications ("guidelines") in order to join the aggregation
  - Transformation: data providers are responsible of structural interoperability (typically light-weight transformation issues); semantics interoperability is typically responsibility of service provider
  - Packaging/splitting not required

Architecture of interoperability solutions

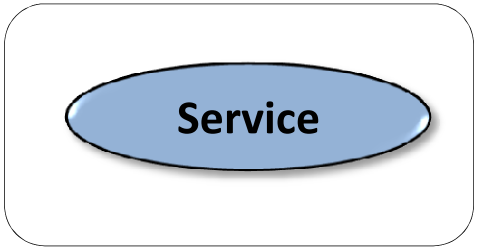
- “Community-oriented“ federations, e.g., the European Film Gateway project
  - Data providers handling the same typology of content invest on the realization of a service provider to enable cross-provider functionality
  - Define a common data model on the service provider
  - Packaging/splitting: if needed, typically occurs at the service provider side
  - Transformation: may occur at the data provider side (before XML export takes place) or data providers are directly involved in the definition of mappings on the service provider
- “Top-down” federations, e.g., OAIster-OCLC project, BASE search engine
  - Realized by organizations willing to deliver a service provider to offer functionality over data providers whose content is openly reachable.
  - Service provider deals with any interoperability issues

D-NET Software Toolkit: general-purpose DLCLs

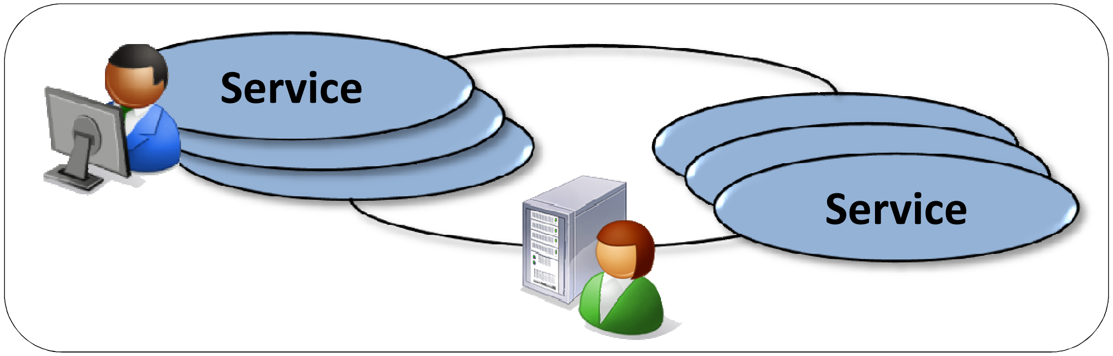
- General-purpose framework for the realization and maintenance of context-specific DLCLs
  - Management of information objects of arbitrary data models
  - Management of DLSs of several typologies (e.g., OAI, ODBC, FTP)
  - Construction of personalized and automated data workflows
  - Management of robustness and scalability parameters
  - DLSs life-cycle administration tools
  - Extensibility with new functionality

D-NET Software Toolkit
The solution…

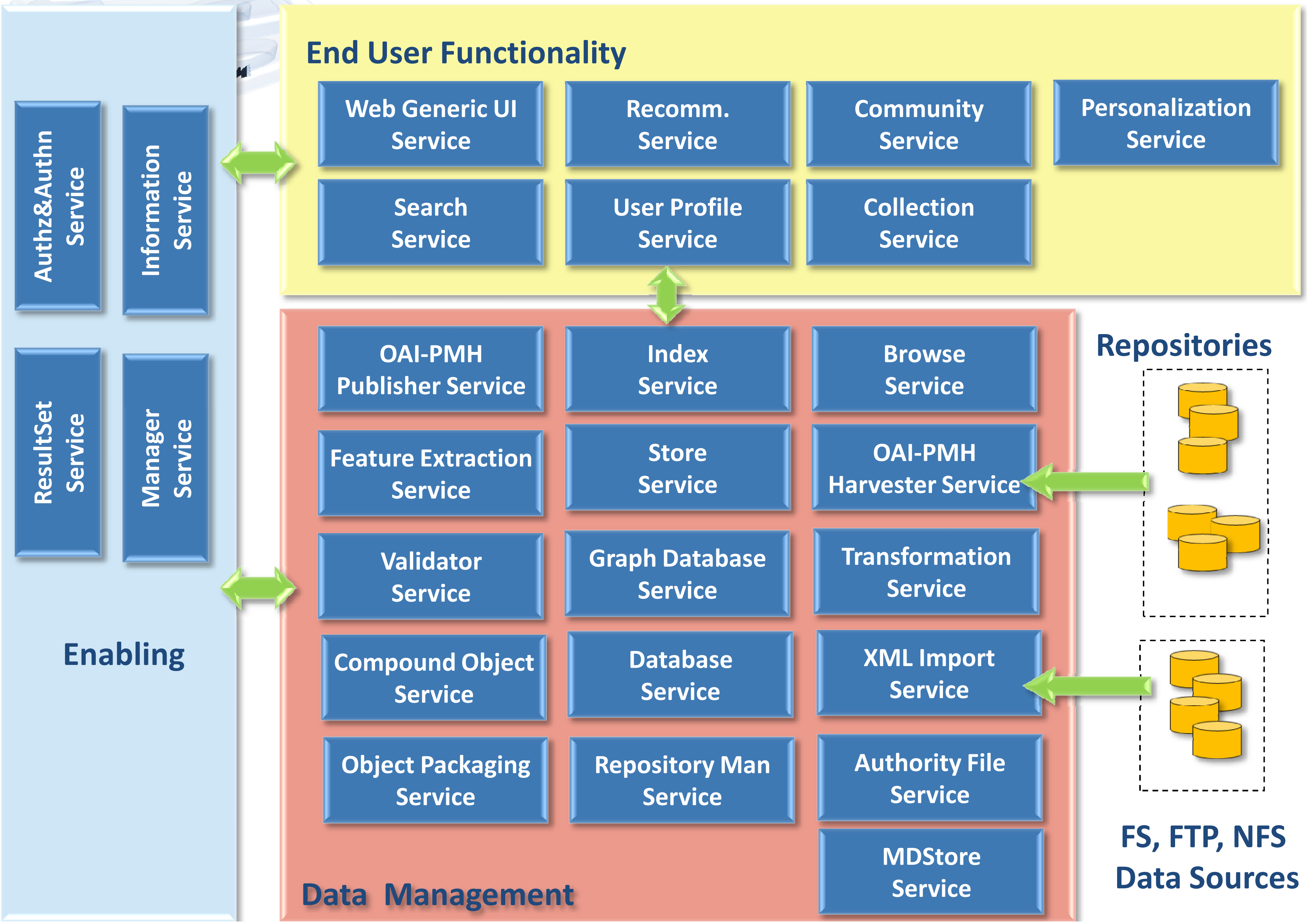
- *Service Kits* supporting realization of “personalized” DLSFs by exploiting customizability, extensibility and modularity features



- *Service-oriented infrastructure features* (autonomicity, distribution and sharing) to support scalable and robust production systems



D-NET: Service Kits



End User Functionality

Web Generic UI Service

Recomm. Service

Community Service

Personalization Service

Search Service

User Profile Service

Collection Service

OAI-PMH Publisher Service

Index Service

Browse Service

Feature Extraction Service

Store Service

OAI-PMH Harvester Service

Validator Service

Graph Database Service

Transformation Service

Compound Object Service

Database Service

XML Import Service

Object Packaging Service

Repository Man Service

Authority File Service

MDStore Service

Data Management

Authz&Authn Service

Information Service

ResultSet Service

Manager Service

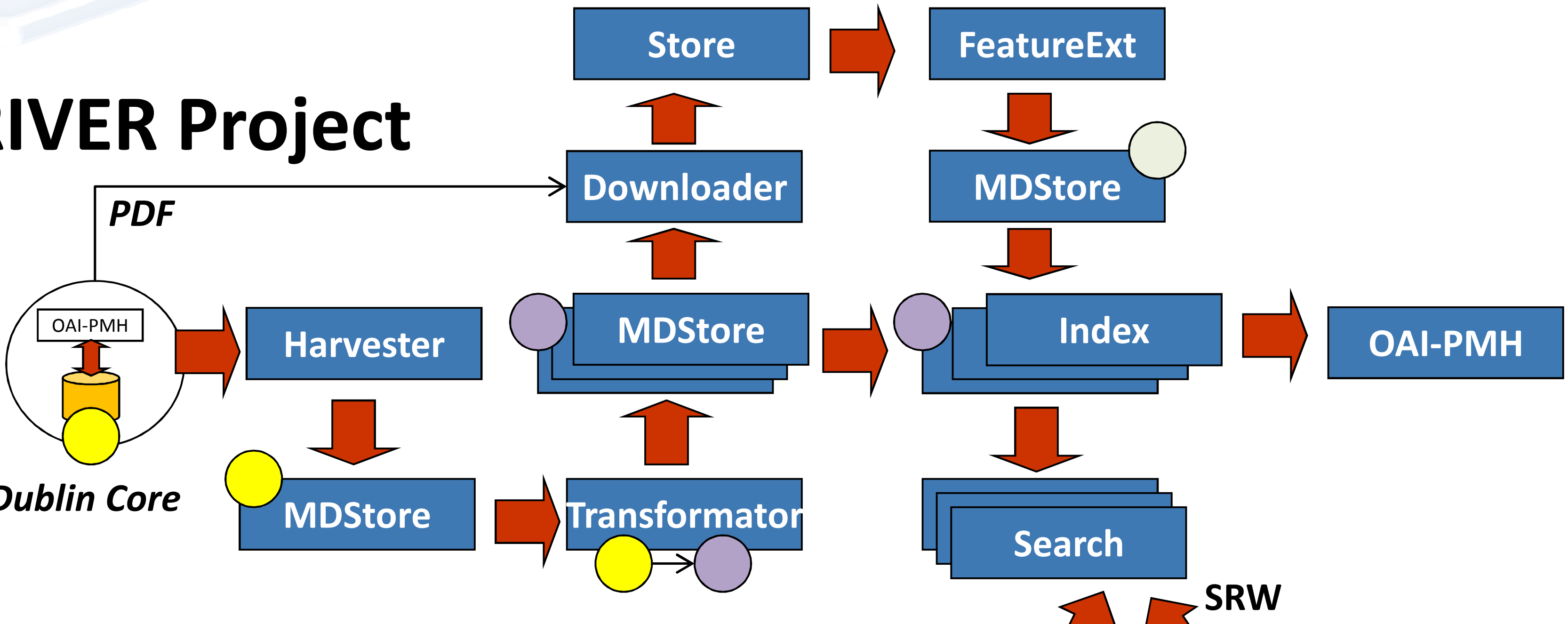
Enabling

Repositories

FS, FTP, NFS Data Sources

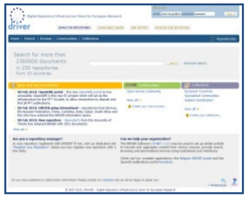
Modularity, customizability, sharing (and orchestration)

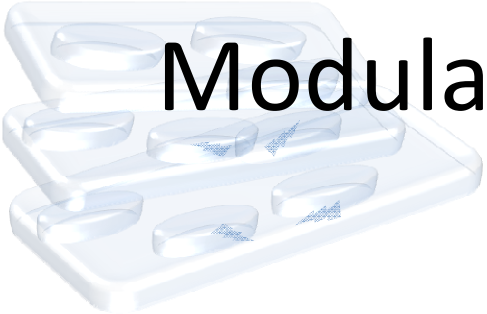
DRIVER Project



Metadata Formats

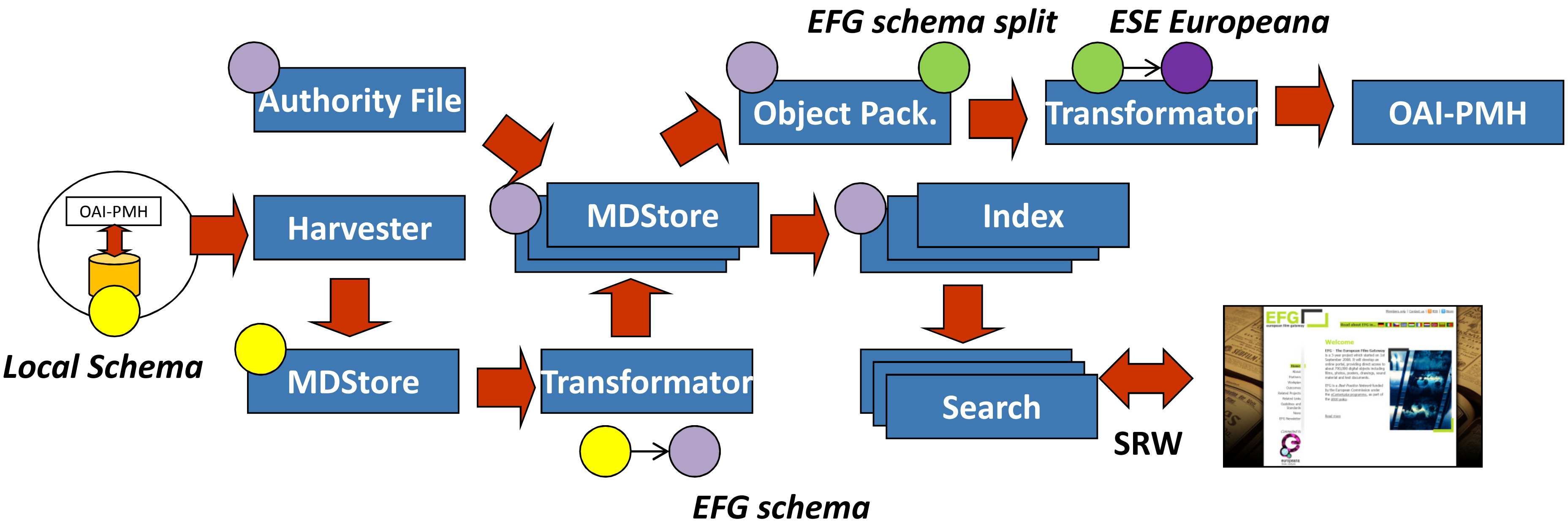
- ● *Dublin Core*
- ● *DRIVER Metadata Format*
- ○ *Intermediate extraction format*





Modularity, customizability, sharing (and orchestration)

EFG Project



● ● ● ●

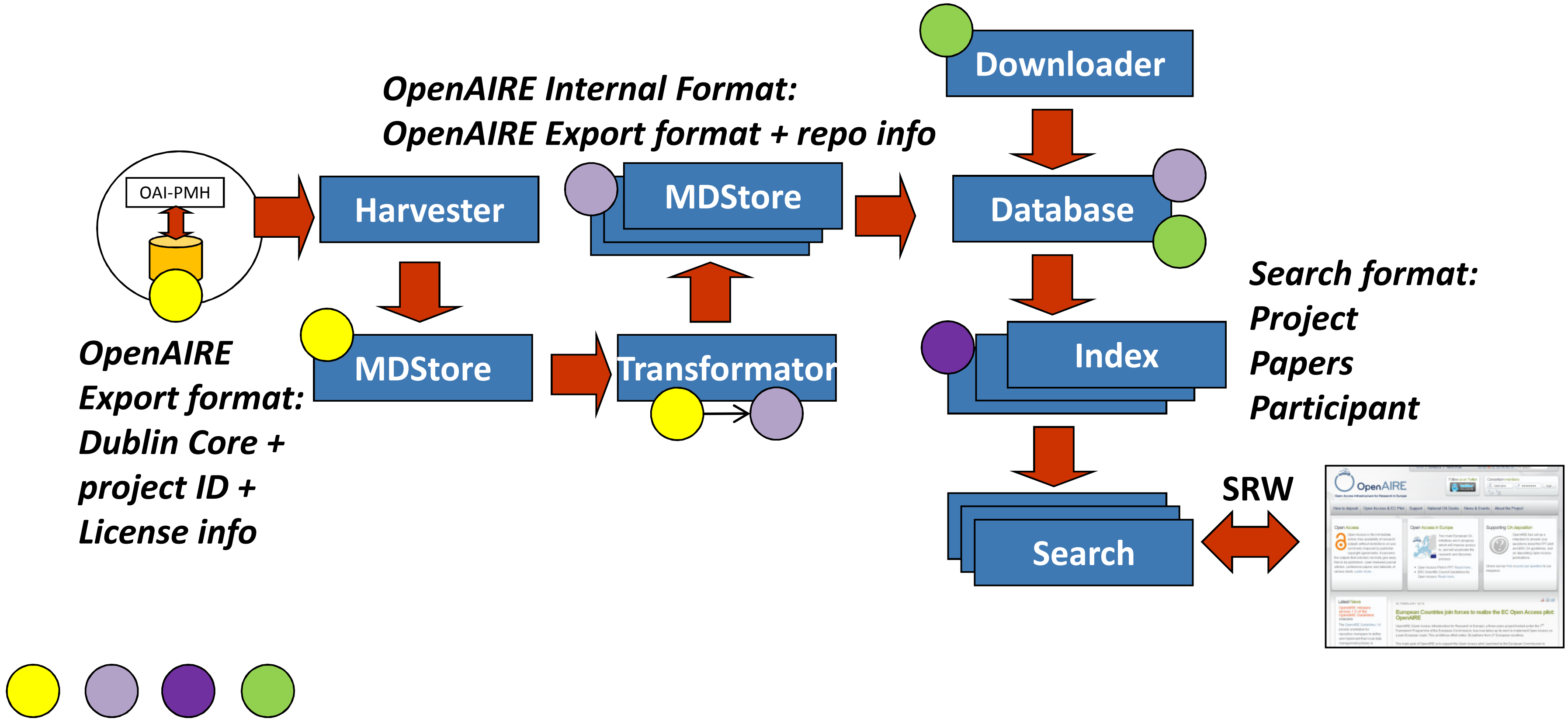
Metadata Formats

Modularity, customizability, sharing (and orchestration)

OpenAIRE Project

Project and Participants Format (from EC)

OpenAIRE Internal Format: OpenAIRE Export format + repo info



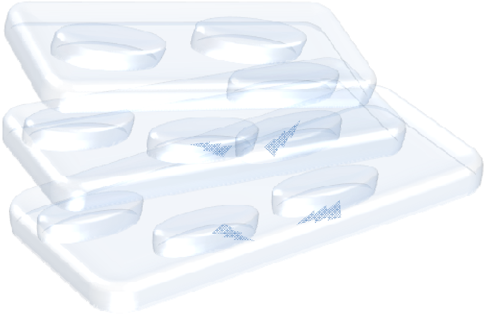
Search format: Project Papers Participant

OpenAIRE Export format: Dublin Core + project ID + License info

Metadata Formats

D-NET’s uptake

- DRIVER project
  - 250 repositories (34 countries), 2,300,000+ items
  - search.driver.research-infrastructures.eu
- European Film Gateway EC project
  - 14 archives, 300,000 items, compound object data model
  - www.europeanfilmgateway.eu
- OpenAIRE EC pilot
  - Harvesting, depositing and statistics of publications and EC project data
  - www.openaire.eu
- HOPE project
  - +20 archives, millions of items, compound object data model
  - www.iisg.nl/news/hope.php
- ScholarLynk
  - R2D2 Project: Microsoft Research Cambridge and D-NET



Experimentation

- Experimentation of deployment of new D-NET repository infrastructures
  - China, India, Portugal, Belgium, Spain, Slovenia
  - Upcoming: Greece and Bulgaria

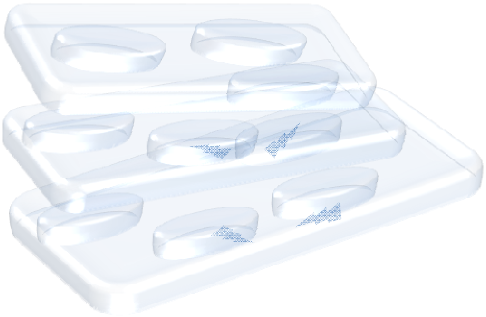
D-Net Software Toolkit

- Software packages
  - Open Source Apache License
  - Release v1.0 (production) and v1.2 (beta)
  - Release v2.0 (beta): Enhanced Publication
- Under continuous refinement

www.d-net.research-infrastructures.eu

Technical Team

- CNR-ISTI: Istituto di Scienze e Tecnologie Informatiche, Centro Nazionale delle Ricerche, Pisa, Italy
- NKUA: Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece
- UNIBI: Universität Bielefeld, Germany
- ICM: Interdisciplinary Centre for Mathematical and Computational Modeling, Uniwesytet Warszawski, Poland



Questions?