

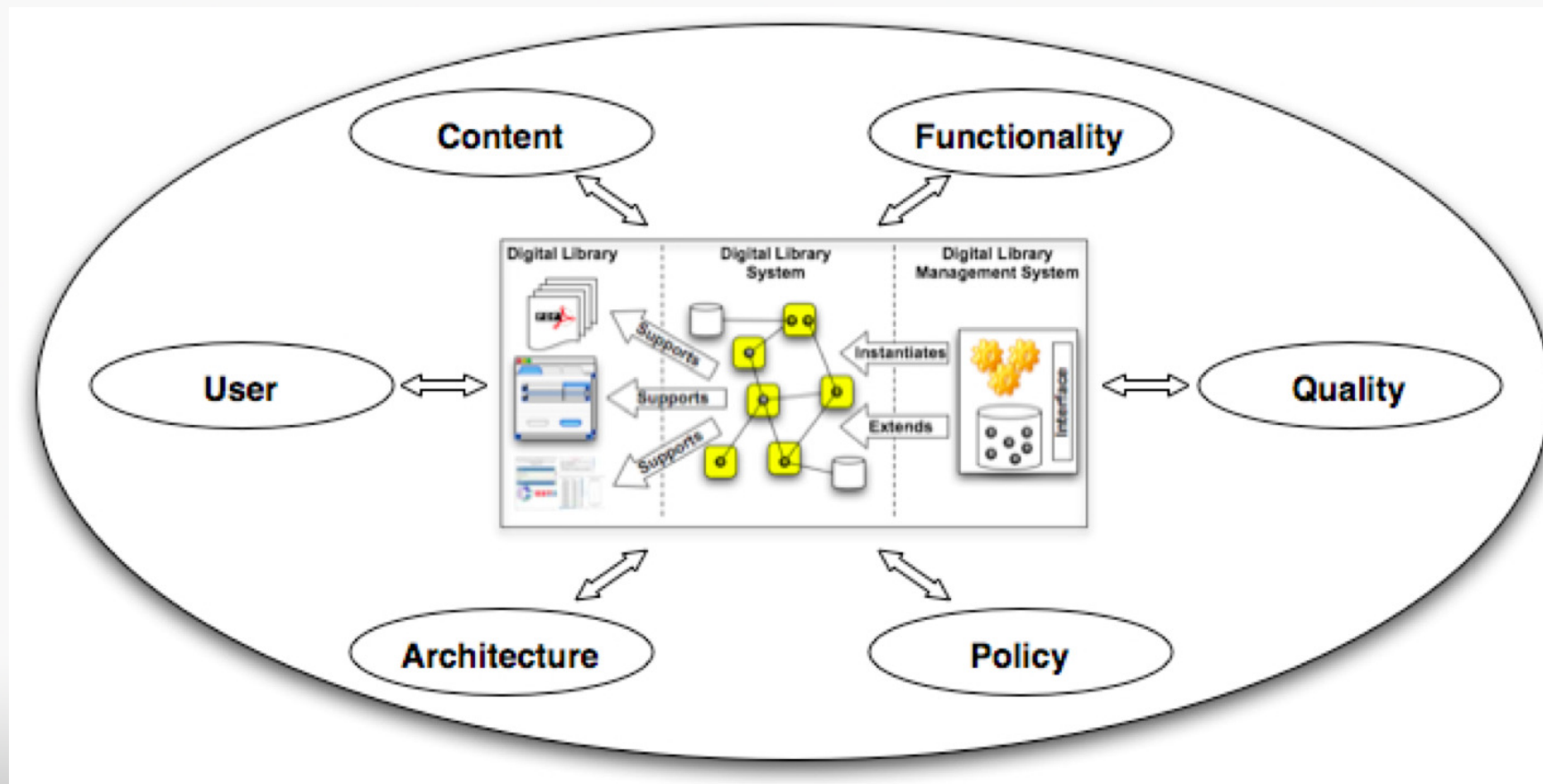


The DL.org Functionality & Content Concepts.

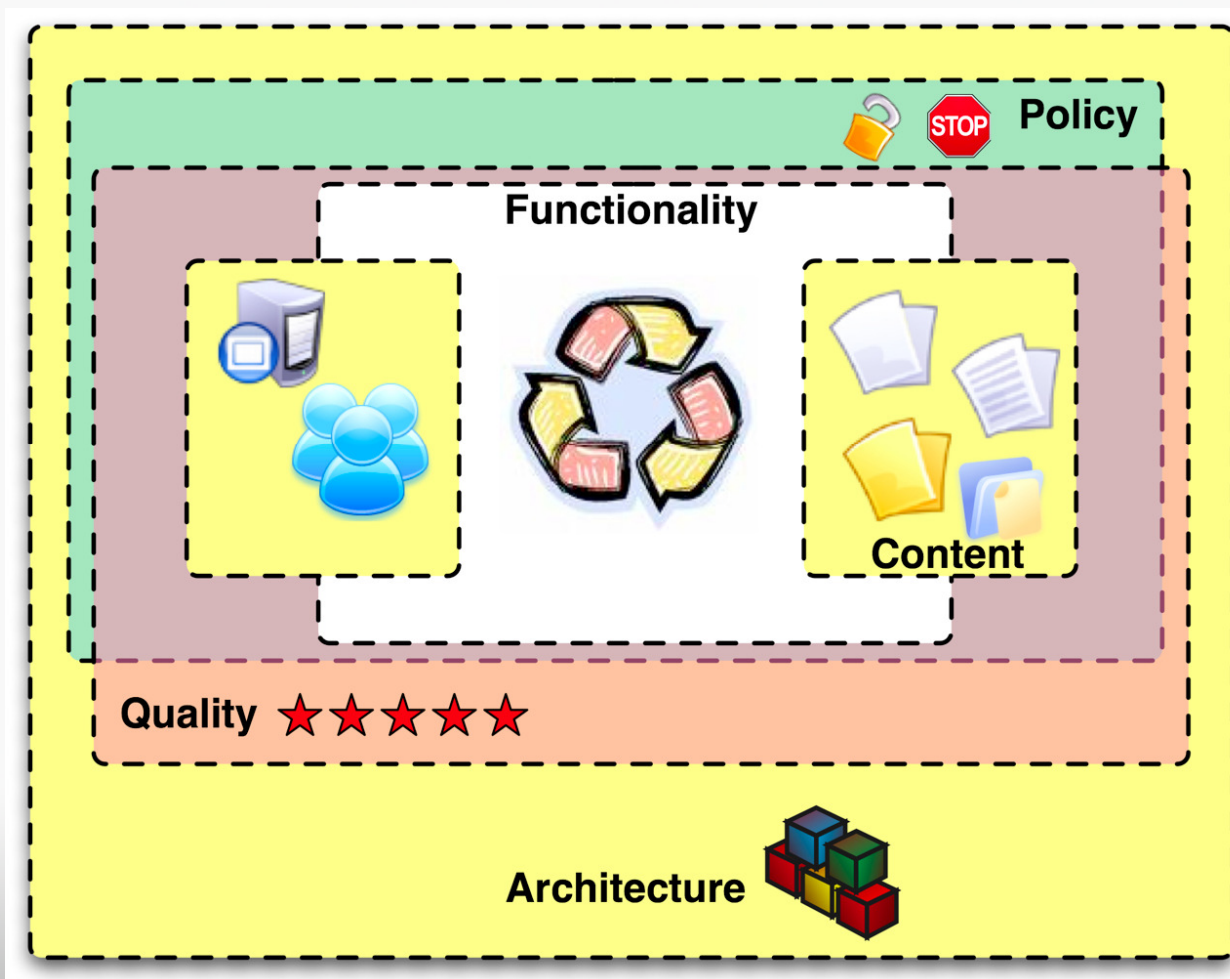
Carlo Meghini
CNR-ISTI
carlo.meghini@isti.cnr.it

DILL Parma, 9 November 2010

Reference Model

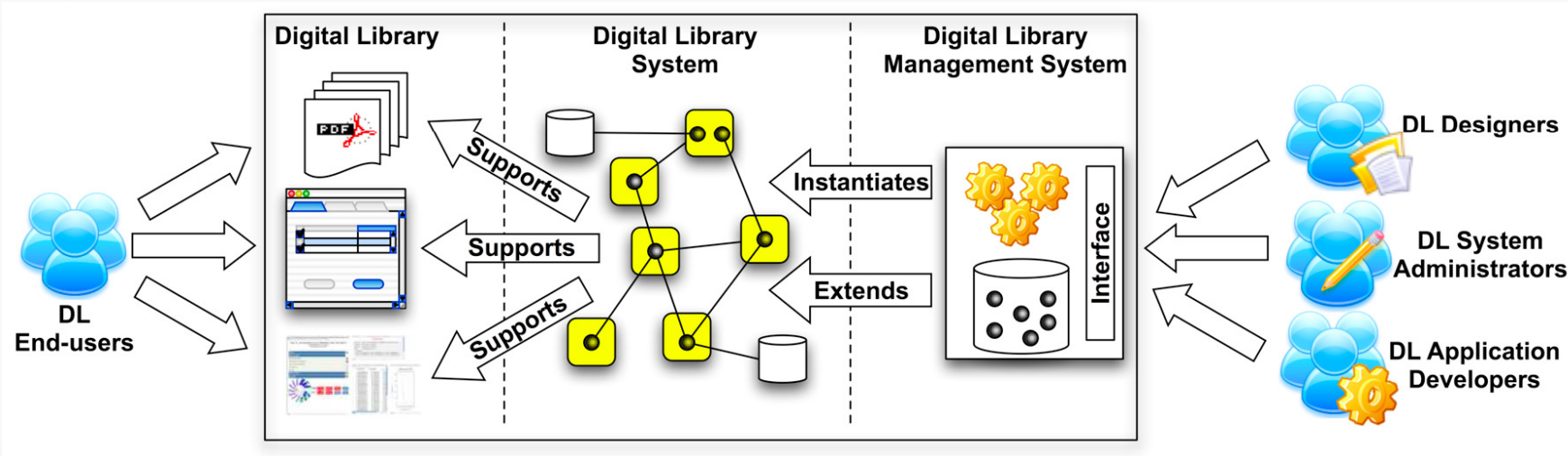


Functionality is the core

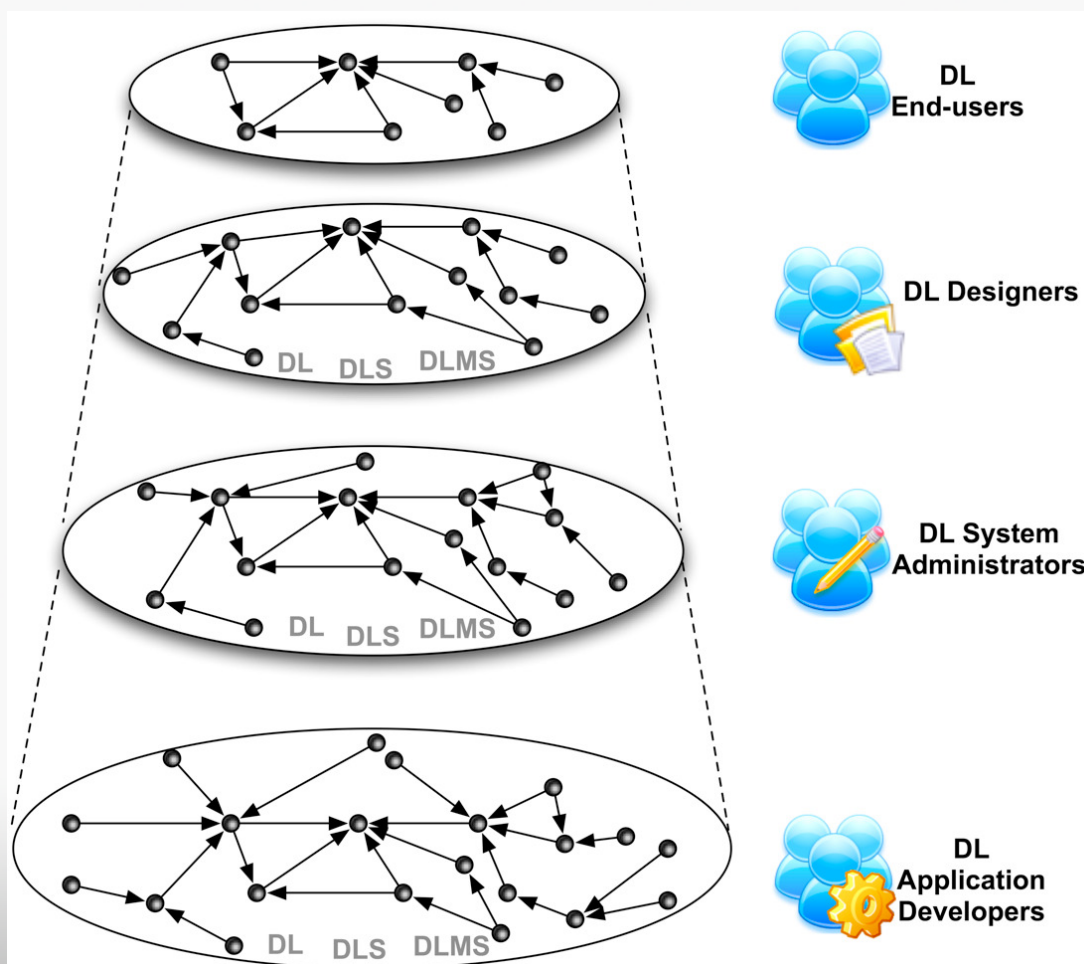


Functionality for ?

- Actors who use the DL functionality:



Relationships among users



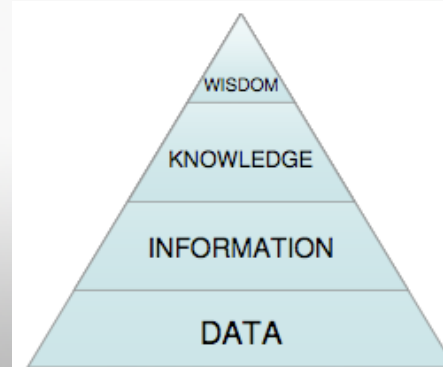
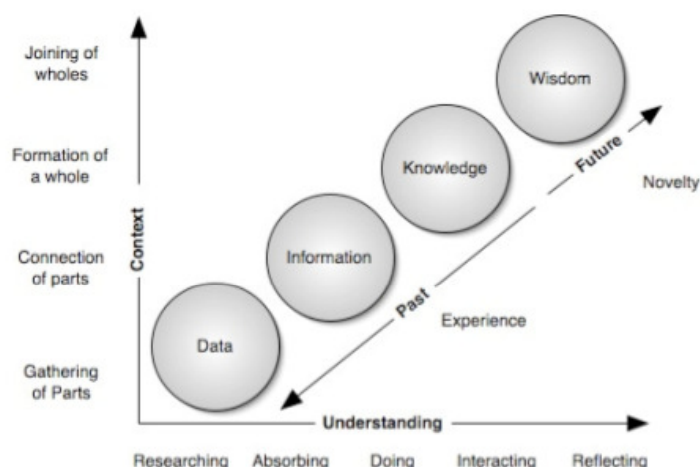
What is [the “Digital Library”] Content?

- Digital Libraries support intellectual activity across logical, conceptual, physical, temporal or personal borders
 - From a content-centric system to a **person-centric** system
 - From static storage and retrieval of information to **facilitation** of **communication**, **collaboration** and other forms of **interaction** among scholars, scientists, researchers or the general public
 - From handling mostly centrally located text to synthesising **distributed multimedia document** collections, **sensor** data, mobile information and pervasive computing services

From data to wisdom a.k.a. the DIKW hierarchy

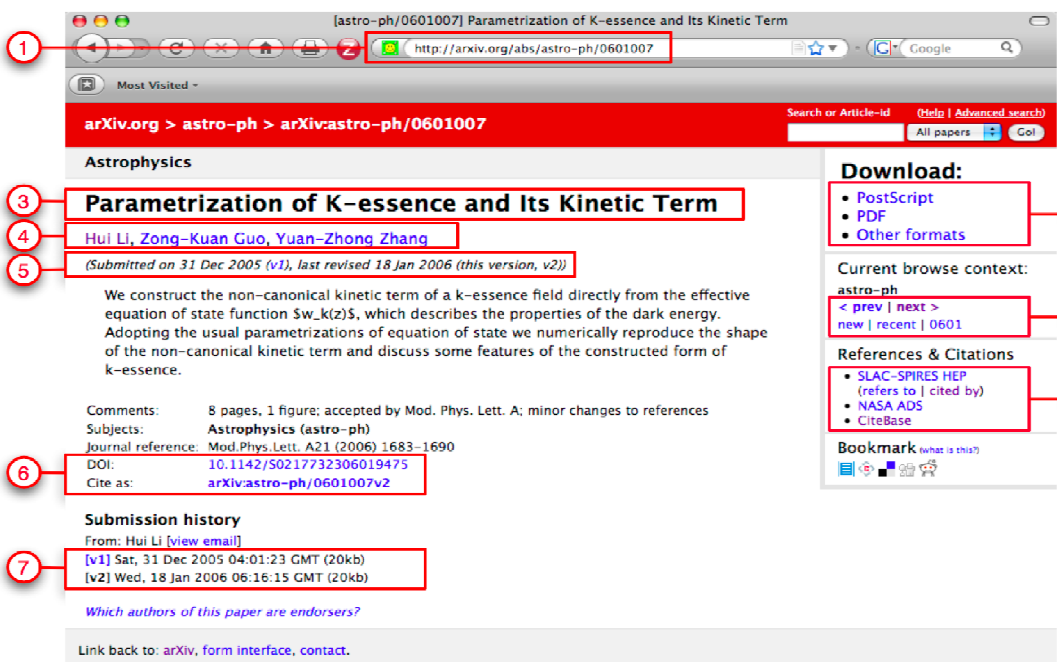
- **data** = raw facts
- **information** = processed data, connected data
- **knowledge** = application of information, appropriate collection of information
- **wisdom** = processed knowledge

www.systems-thinking.org



“Papers” today

Aggregations



1. The URI of the human start page for the arXiv document.

2. The formats in which the document is available: constituents of the aggregation.

3. The title of the document.

4. The authors of the document.

5. The creation and last modification date of the document.

6. Identifiers of entities that are in some manner equivalent to this document. For example, the DOI of a peer-reviewed article.

7. The versions of this document.

8. Links to other arXiv documents in the same collection.

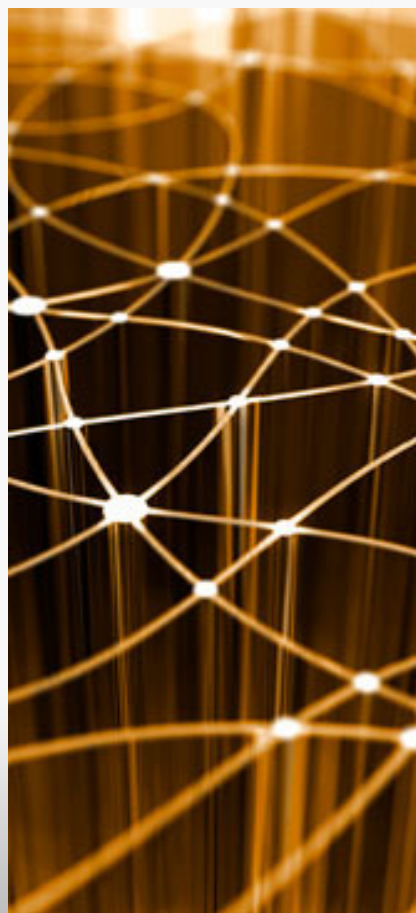
9. Citations made by this document, and citations it received from other documents.



OAI Object Reuse & Exchange
Herbert Van de Sompel & Michael L. Nelson
Woodruff Library, Emory University, October 1 2009







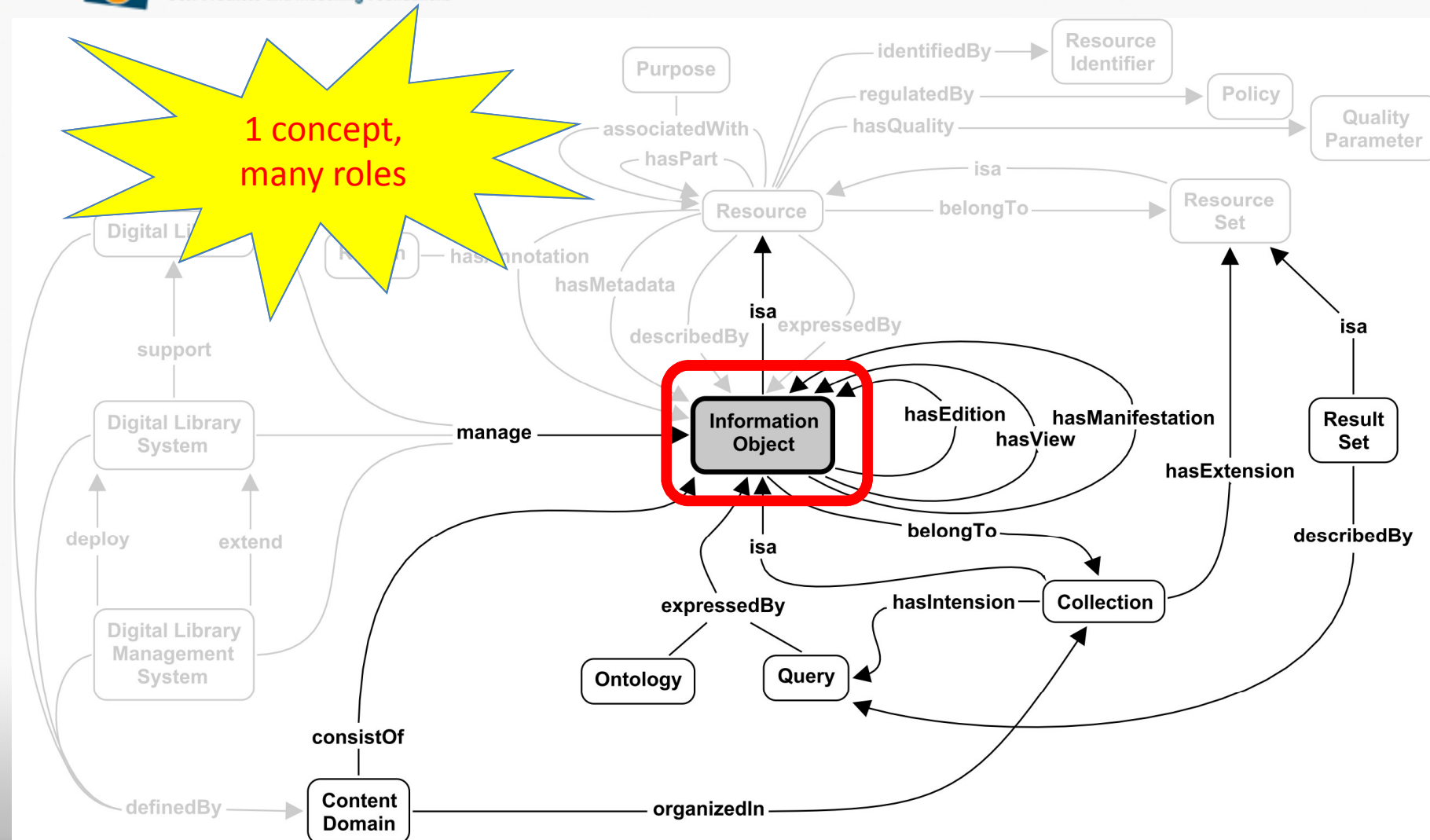
Content Domain: the Reference Model

Content Domain

One of the six main concepts characterising the Digital Library universe. It represents the various aspects related to the modelling of information managed in the Digital Library universe to serve the information needs of the *Actors*.

- Encompasses the **data** and **information** that the Digital Library handles and makes available to its users
- Encompasses the diverse **range of information objects**, including such resources as objects, annotations and metadata
- It is composed of a set of information objects organised in **collections**

Content Domain: the map



Information Object

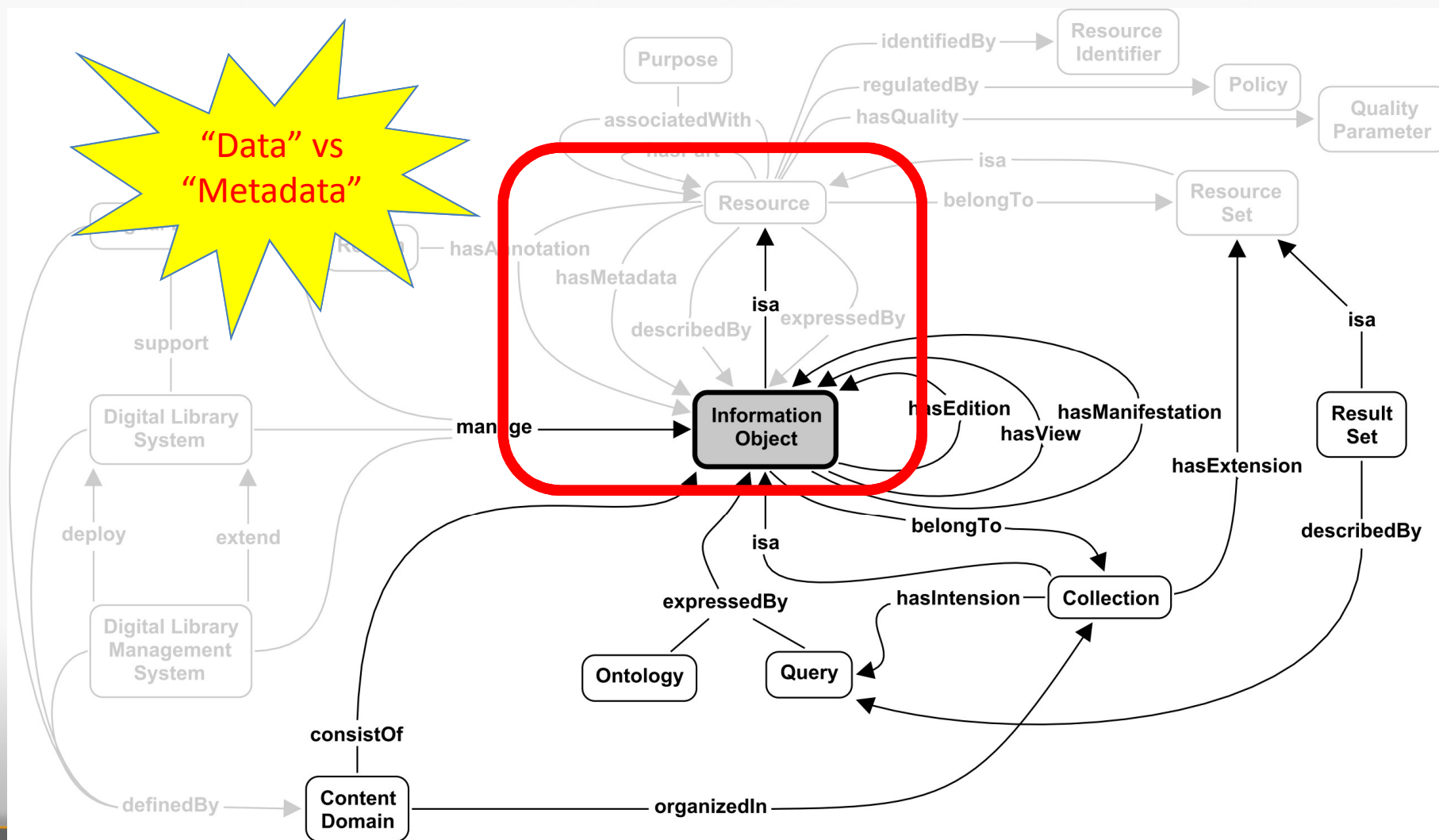
The main *Resource* of the *Content Domain*. An *Information Object* is a *Resource* identified by a *Resource Identifier*. It may have *Metadata*, *Annotations* and multiple *Editions*, *Views*, *Manifestations*, which are also represented as *Information Objects*. In addition, it may have *Quality Parameters* and *Policies*.

Information Object (cont.)

As an *Information Object* is a *Resource*, it inherits all its features

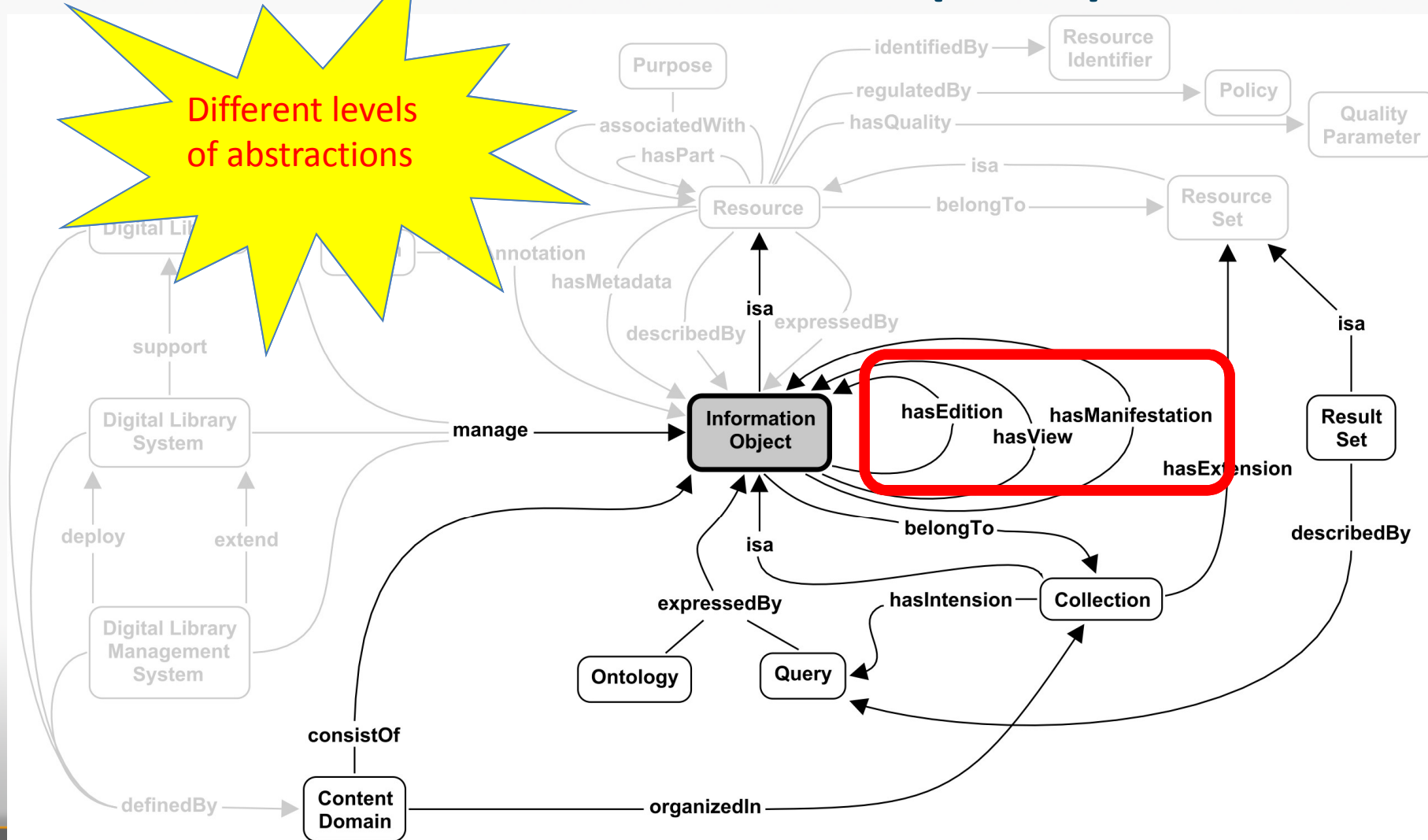
- has a unique identifier (*Resource Identifier*) also known as the information object identifier;
- is arranged according to a format (*Resource Format*) also known as the document model;
- can arbitrarily be **composed** (<*hasPart*> and <*associatedWith*>) to capture compound artefacts;
- is characterised by various *Quality Parameters* each capturing different object quality facets (<*hasQuality*>);
- is regulated by *Policies* (<*regulatedBy*>) governing every aspect of its lifetime; and
- can be described or augmented by *Metadata* (<*hasMetadata*>) and *Annotations* (<*hasAnnotation*>)

Content Domain: where is the rest?

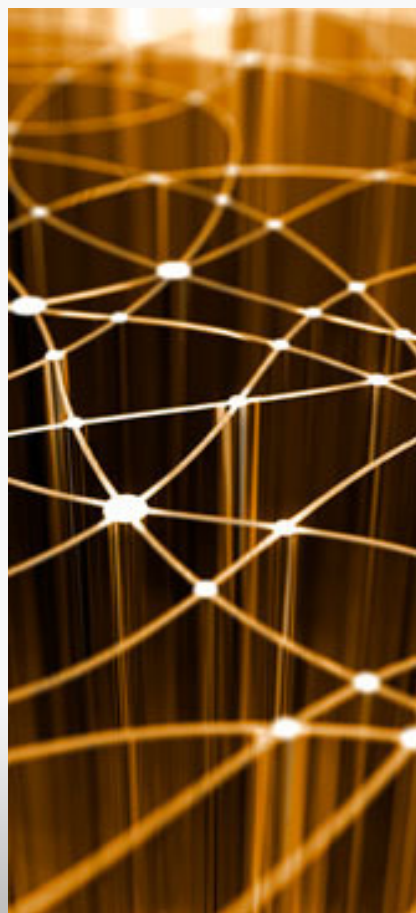


Content Domain: where is the rest? (cont.)

Different levels of abstractions







Functionality Domain: the Reference Model

The Functionality Domain

Captures all services that are offered on
Resources and activities that can be observed
by *Actors* in a Digital Library

Function

Wikipedia (Oct. 4, 2010, 11:12:56 CET)

- *In the abstract set-theoretic approach, a function is a relation between the domain and the codomain that associates each element in the domain with exactly one element in the codomain.*
- *An example of a function with domain $\{1,2,3\}$ and codomain $\{2,3,4\}$ associates 1 with 2, 2 with 3, and 3 with 4.*

A special kind of function

Réponse n° 1



Domaine	peinture
Type d'objet	tableau
Titre	PORTRAIT DE MONA LISA (1479-1528) ; DITE LA JOCONDE
Auteur/exécutant	LEONARDO DI SER PIERO DA VINCI ; VINCI Léonard de (dit)
Précision auteur/exécutant	Vinci, 1452 ; Amboise, 1519
Ecole	Italie
Période création/exécution	1er quart 16e siècle
Millésime création/exécution	1503 entre ; 1506 et
Genèse	oeuvre en rapport ; reproduit en gravure
Historique	commandé par le florentin Francesco del Giocondo, époux de Mona Lisa entre 1503 et 1506 ; nombreuses copies dont une conservée au Louvre ; gravé par Fauchery, par Filhol, par Landon
Matériaux/techniques	peinture à l'huile ; bois
Mesures	77 H ; 53 L
Sujet représenté	portrait (Mona Lisa, femme, à mi-corps, de trois-quarts, assis, accoudé, loggia, Italien) ; fond de paysage (montagne, rocher, cours d'eau, pont, plaine, route)
Date sujet représenté	1479-1528
Lieu de conservation	Paris ; musée du Louvre département des Peintures
	 Musée de France au sens de la loi n° 2002-5 du 4 janvier 2002
Statut juridique	propriété de l'Etat ; musée du Louvre département des Peintures
Anciennes appartenances	François Ier ; Couronne de France
Numéro d'inventaire	INV 779
Commentaires	légère diminution du tableau sur les côtés (environ 7 mm) ; acheté vraisemblablement vers 1519, après la mort de l'artiste
Bibliographie	HEYDENRICH 6 ; OTTINO DELLA CHIESA 31 ; VILLOT I 484 ; HAUTECOEUR 1601 ; C.S.I. 1981, P 192
Copyright notice	© Musée du Louvre, © Direction des Musées de France, 1999
Crédits photographiques	© Réunion des musées nationaux ; © Hervé Lewandowski ; © Thierry Le Mage
	 commande reproduction et/ou conditions d'utilisation
	renseignements sur le musée
	000PE025604

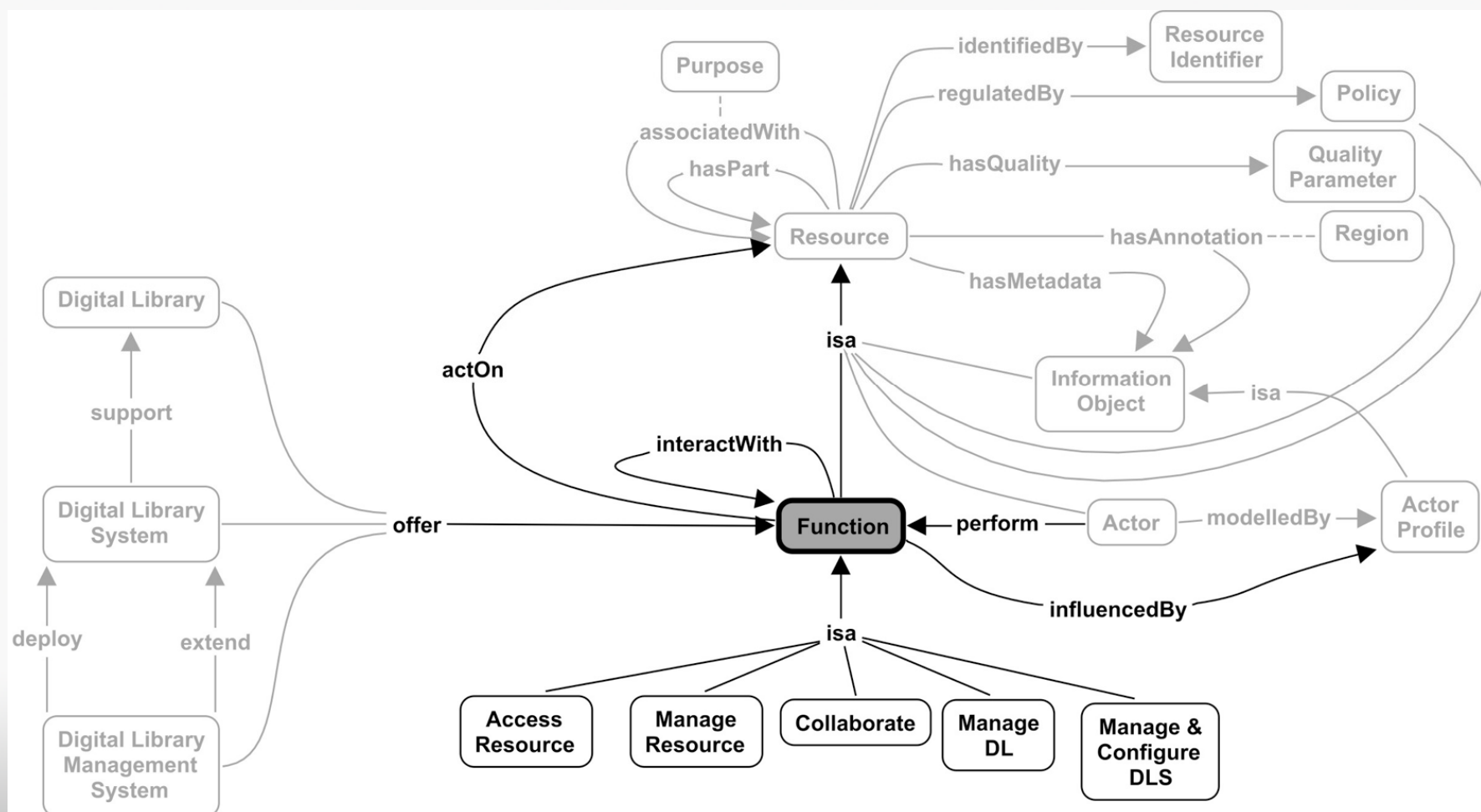
In Computer Science

- A function is a (logical) machine that performs a specific task.
 - In a DL, there are a number of such machines, ready to be used by the (authorized) user
- In order to use a function, the function must be applied
 - i.e. the machine must be started
 - push a button on a GUI
 - enter a URL in a browser
 - type some text in a terminal window

In Computer Science

- When applied, a function becomes a process:
 - has input parameters
 - has output parameters
 - may change the DL
- Querying
- Inserting an object
- but there is a lot more ...

The Map



A Function is-a Resource

- it has a unique identifier (Resource Identifier)
 - is it an information resource or a non-information resource?
- it has structure:
 - can be atomic
 - composed of no other functions
 - the composition of simpler functions, which results in an arbitrarily structured workflow:
 - <hasPart> A function has functions as its parts
 - <associatedWith> A function is associated with a function for a Purpose

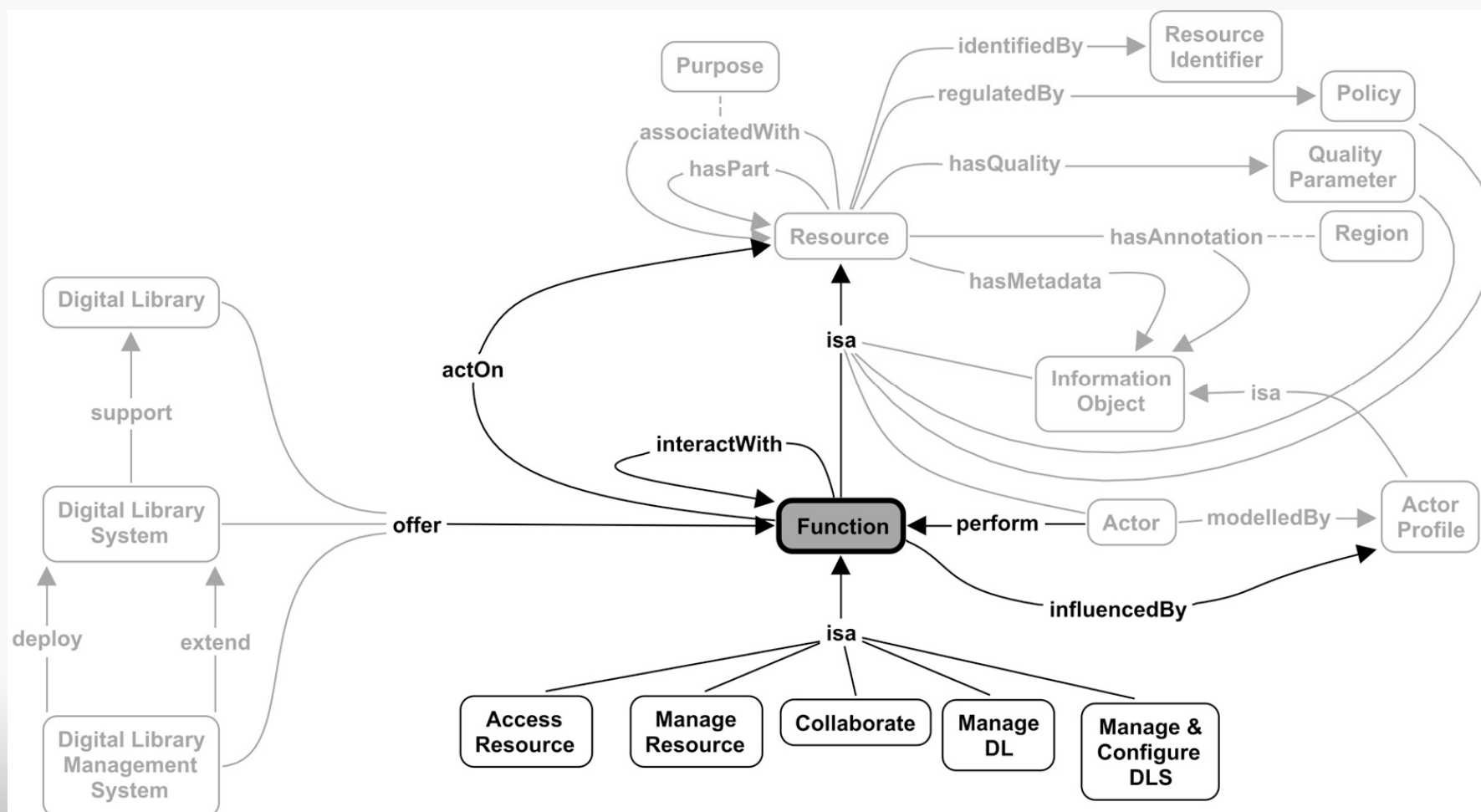
A Function is-a Resource

- it is characterised by various Quality Parameters covering various quality aspects (<hasQuality>)
 - synchronous vs. asynchronous
 - efficient
 - robust
 - state-full vs state-less
 - CPU-bound vs I/O-bound
- its lifetime and behaviour are regulated by Policies (<regulatedBy>)
 - which Actors are allowed to perform the Function in a certain context
 - which billing schema applies to the Function
- it can be enriched with Metadata (<hasMetadata>)
 - there are many languages for describing functions, from technically-oriented ones (such as WSDL) to more semantically-oriented ones (DAML-S)
 - Functions are searchable, like any other resource

A Function is-a Resource

- it can be enriched with Annotation (<hasAnnotation>)
 - pre-formalization
 - helping the interpretation
 - social activity

The Map



A Function is a Function

- A Function acts on Resources (<actOn>)
 - Resources means not only Information Objects but also Functions, Actor profiles, Policies, etc.
- A Function interacts with other Functions (<interactWith>)
 - Orders functions within a workflow

How is a Function born?

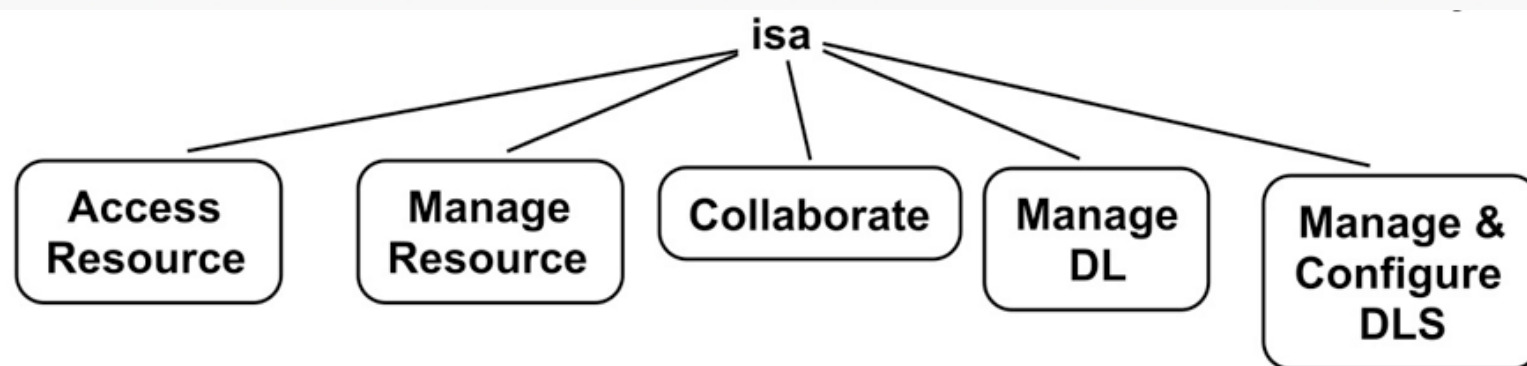
The result of a complex process/chain.

A simplified waterfall picture (largely inspired by the process in Europeana):

- Business sees a new opportunity and defines a need
- Potential users elaborate the need in form of a (system of) requirement(s)
- Conceptual modellers turn requirements into a functional specification
- Developers turn the functional specification into a technical specification
- Business performs cost/benefit analysis on the technical specification and (sometimes) signs it off for implementation
- Developers turn the technical specification into software
- Quality controllers test the software to check whether it meets quality parameters
- Users tests the software to check whether it meets the initial requirements
- System administrators deploy the software into the architecture, and subsequently make sure the software operates correctly as the context around the DL evolves.

What Functions in a DL?

- Each Digital Library may have its own set of *Functions* depending on its underlying business models.
- *Function* is specialised into five other concepts that still represent quite general classes of activities.



Access

Access Functions help in identifying and obtaining
Resources.

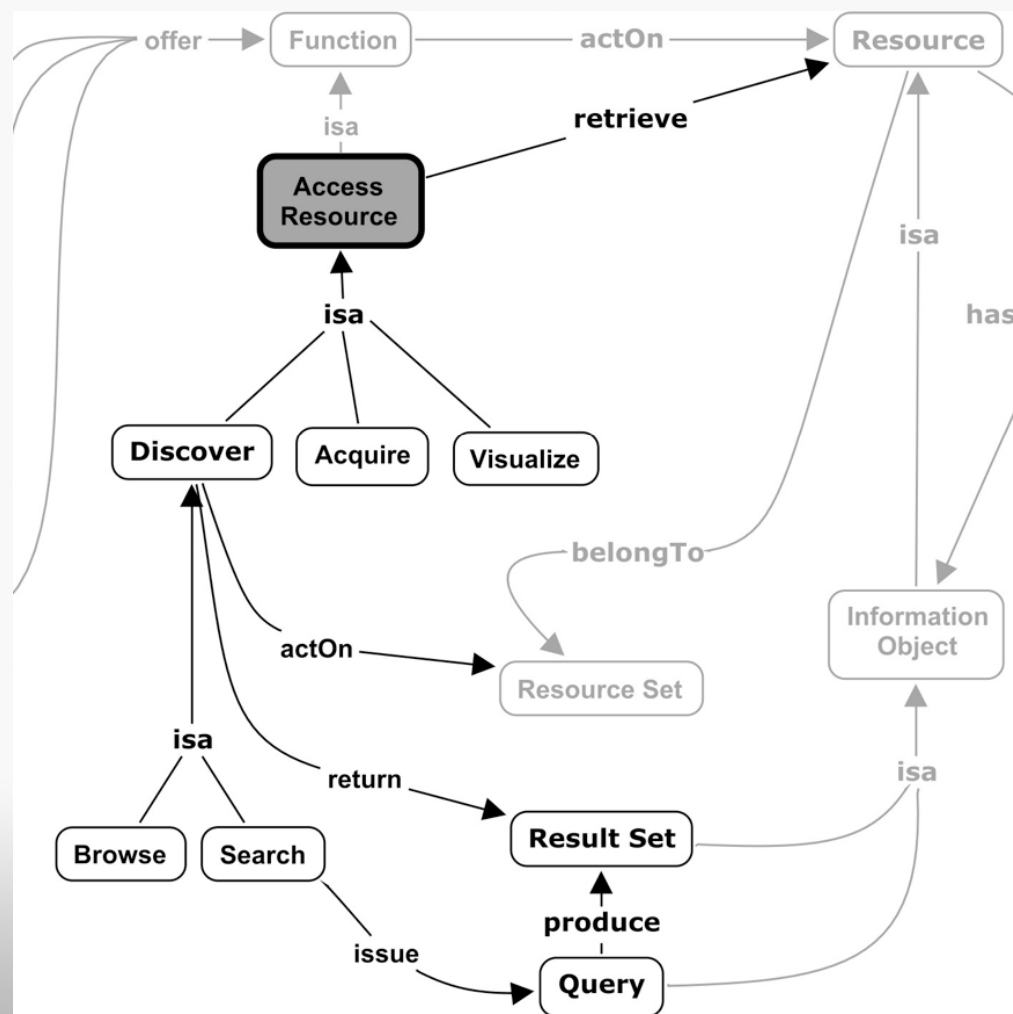
Access Resource encompasses all machines related
to

- requesting
- locating
- retrieving
- transforming
- representing in a ‘material form’
a *Resource*

Access Functions do not modify the DL

Access functions

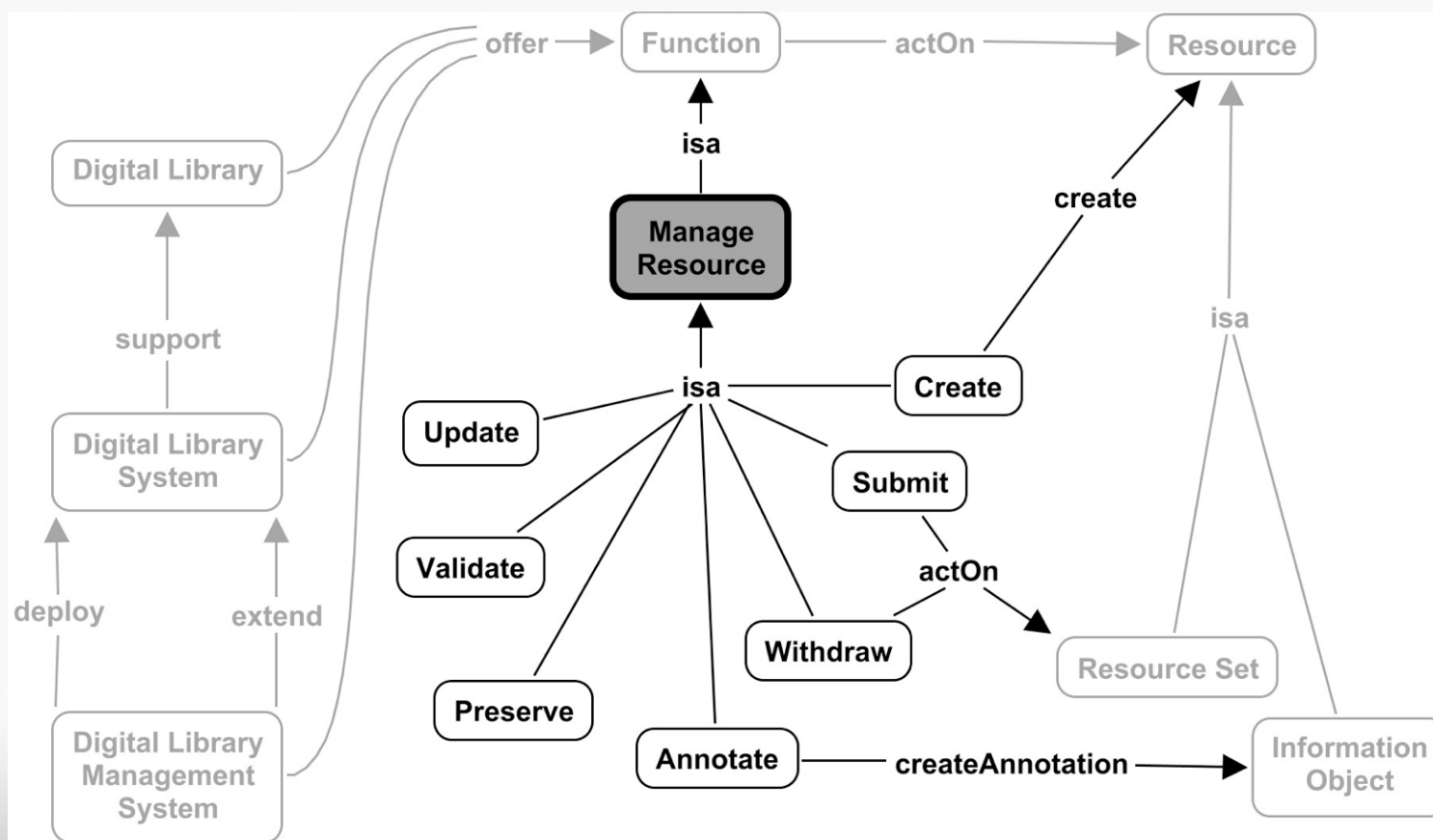
C32 Access Resource
C33 Discover
C34 Browse
C35 Search
C36 Acquire
C37 Visualise



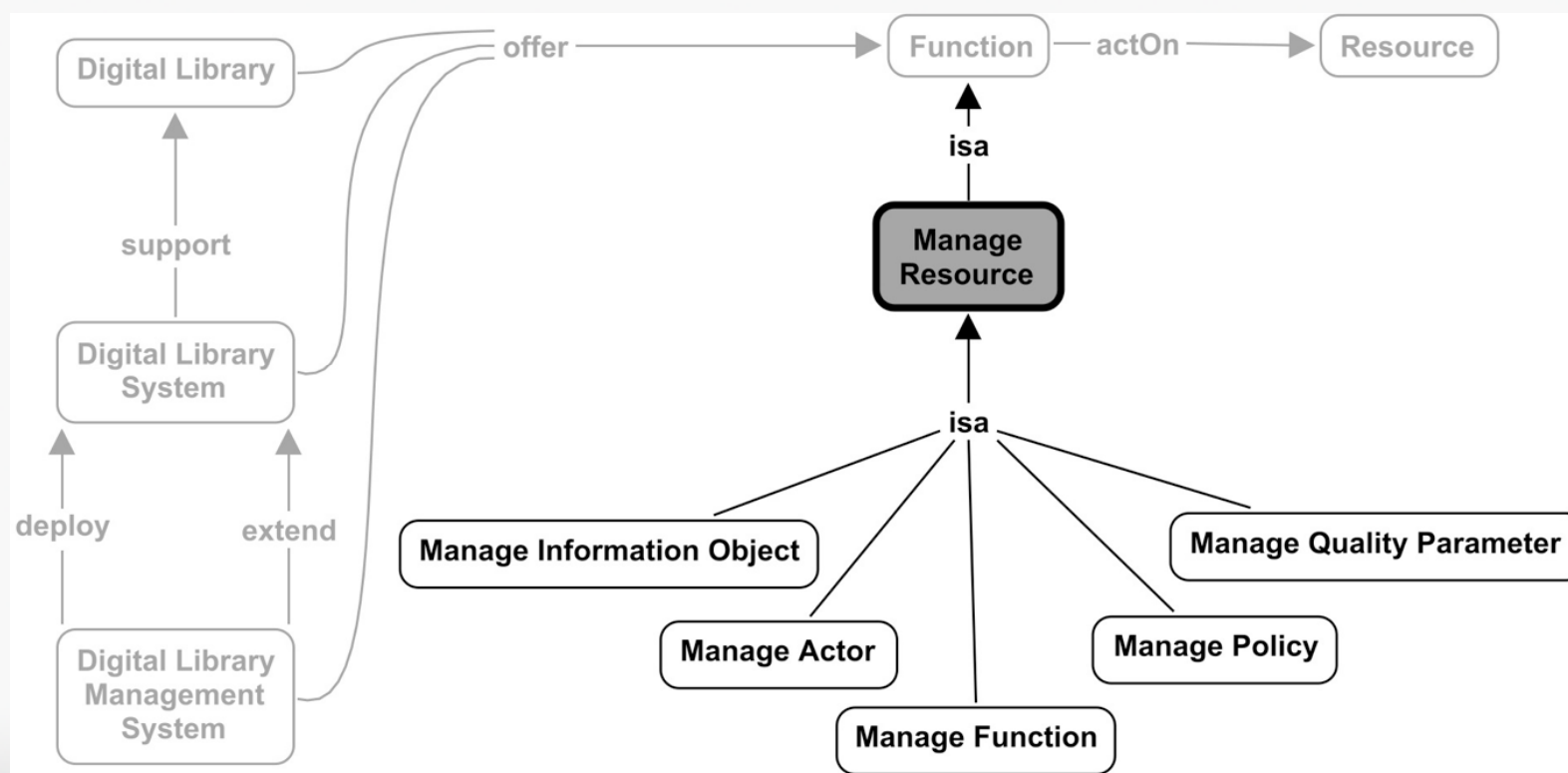
Manage Resource

- includes all machines for
 - creating new Resources
 - inserting them into the DL
 - deleting old Resources
 - updating existing Resources,
 - converting or transforming existing Resources.

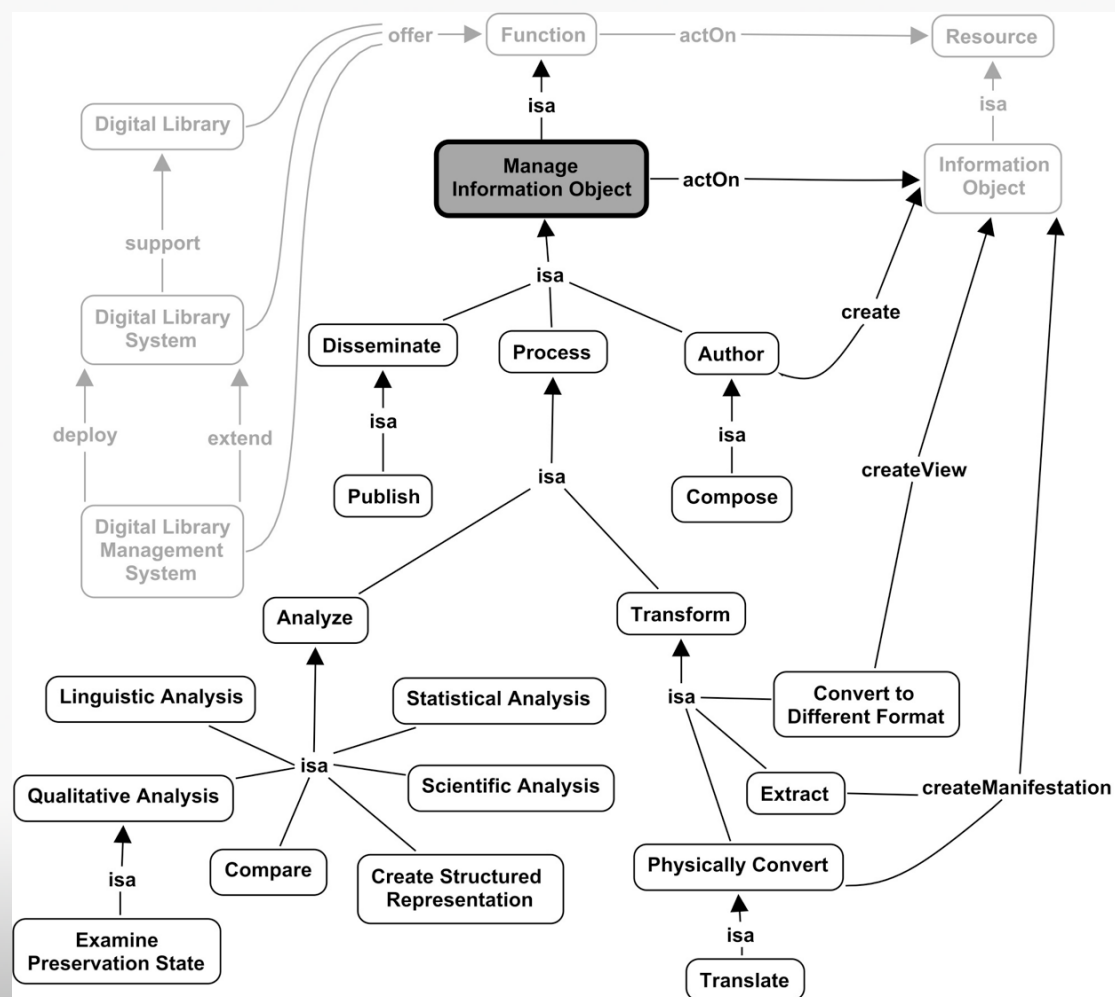
Manage Resource for all Resources



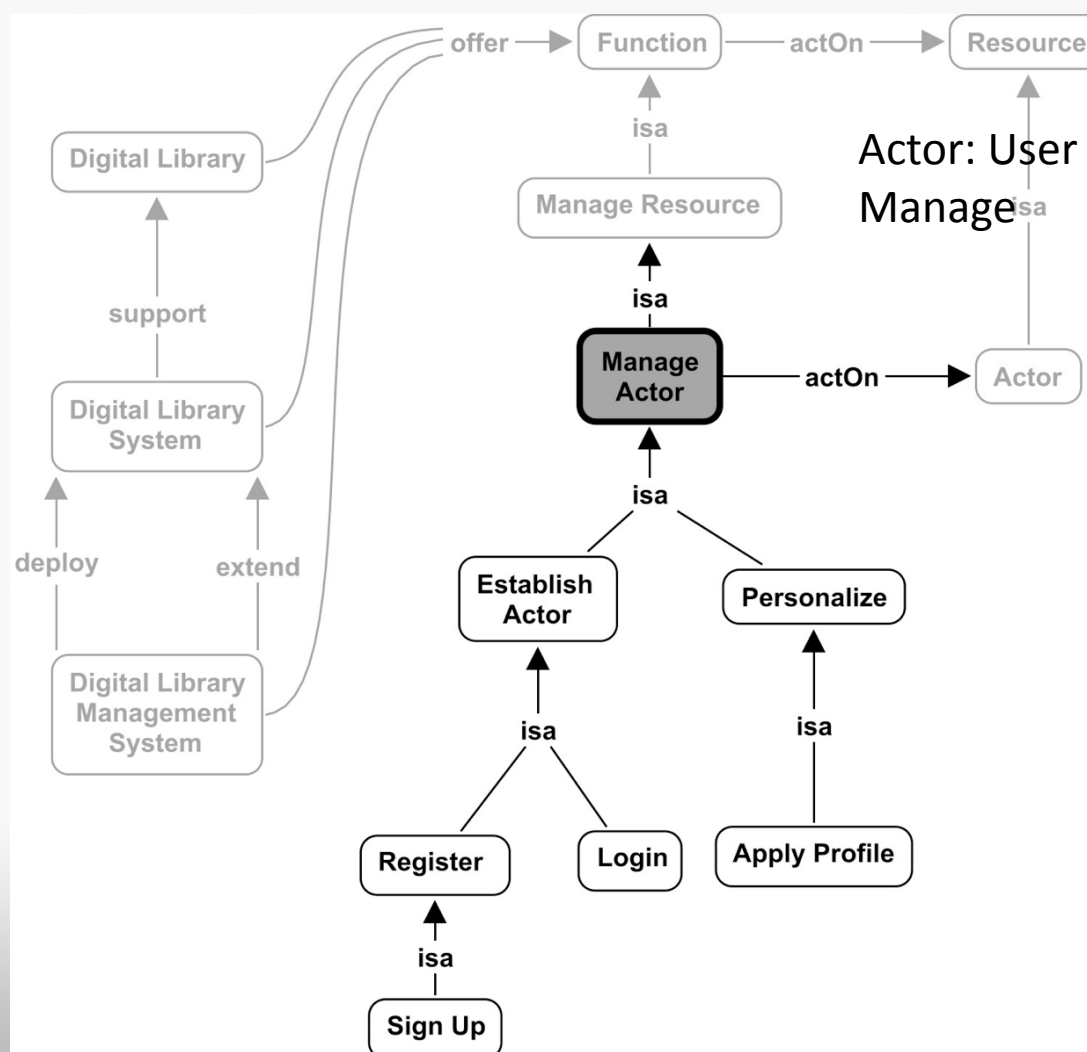
Manage Resource per Resource Type



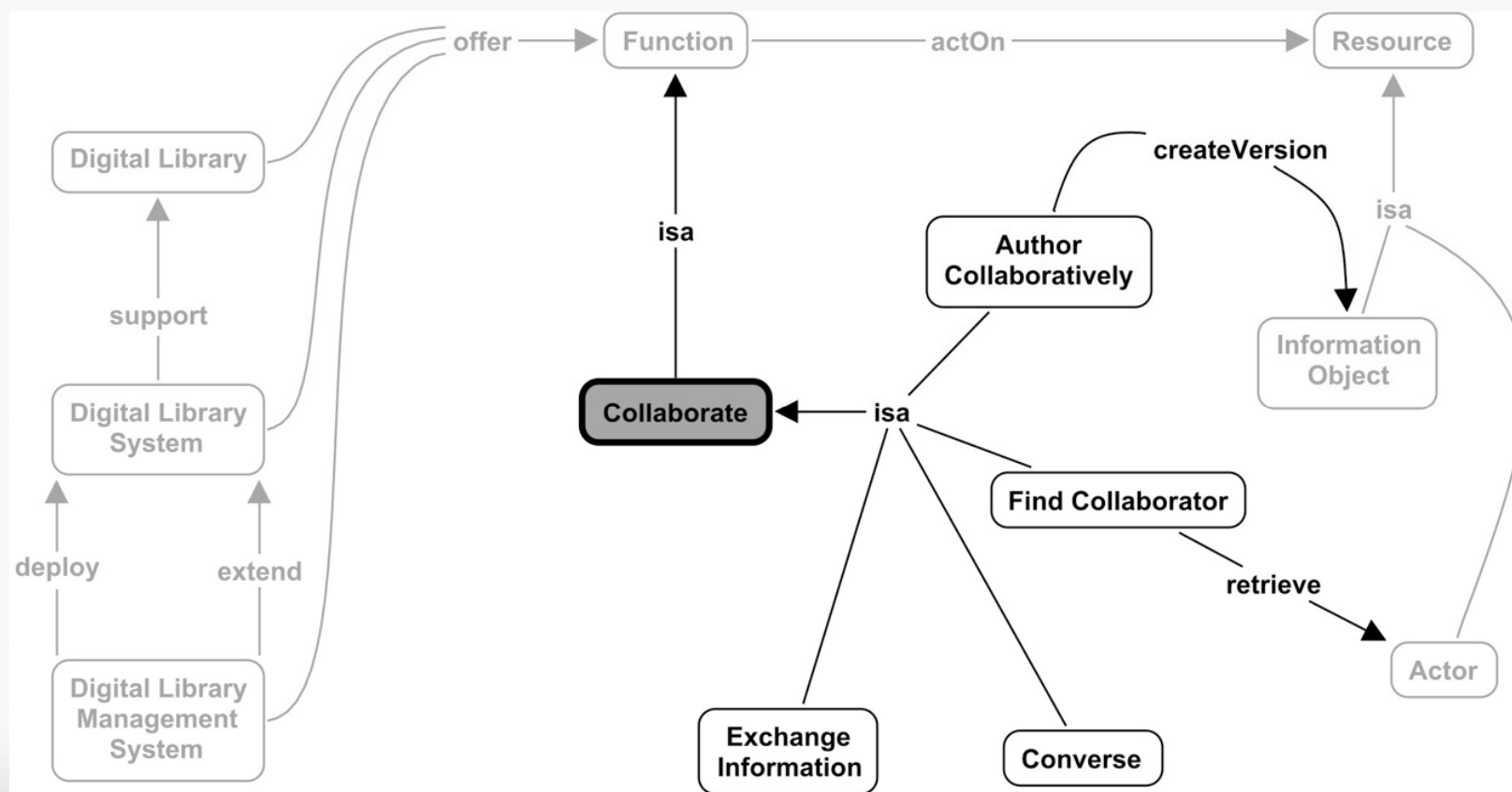
Manage Information Object



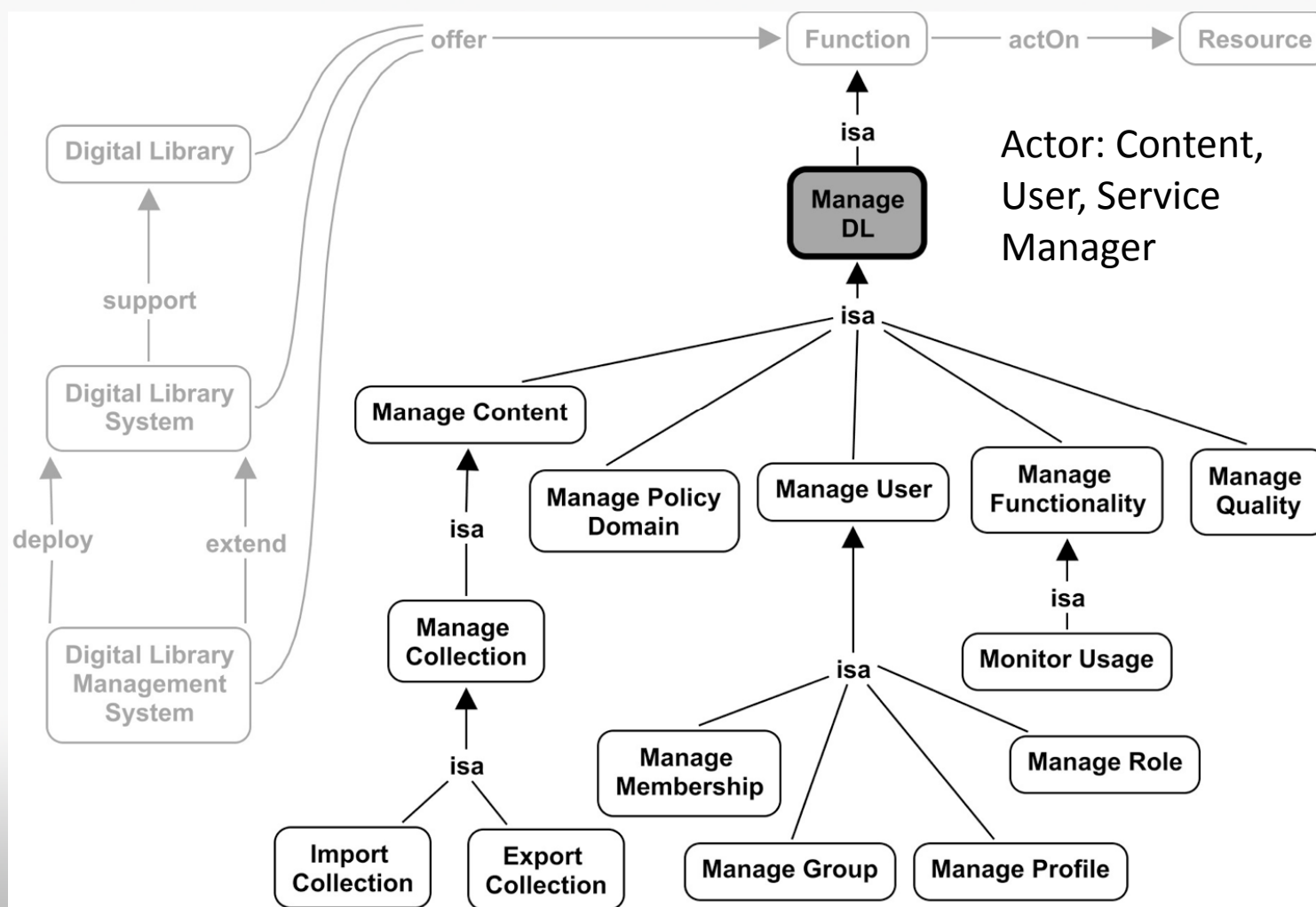
Manage Actor



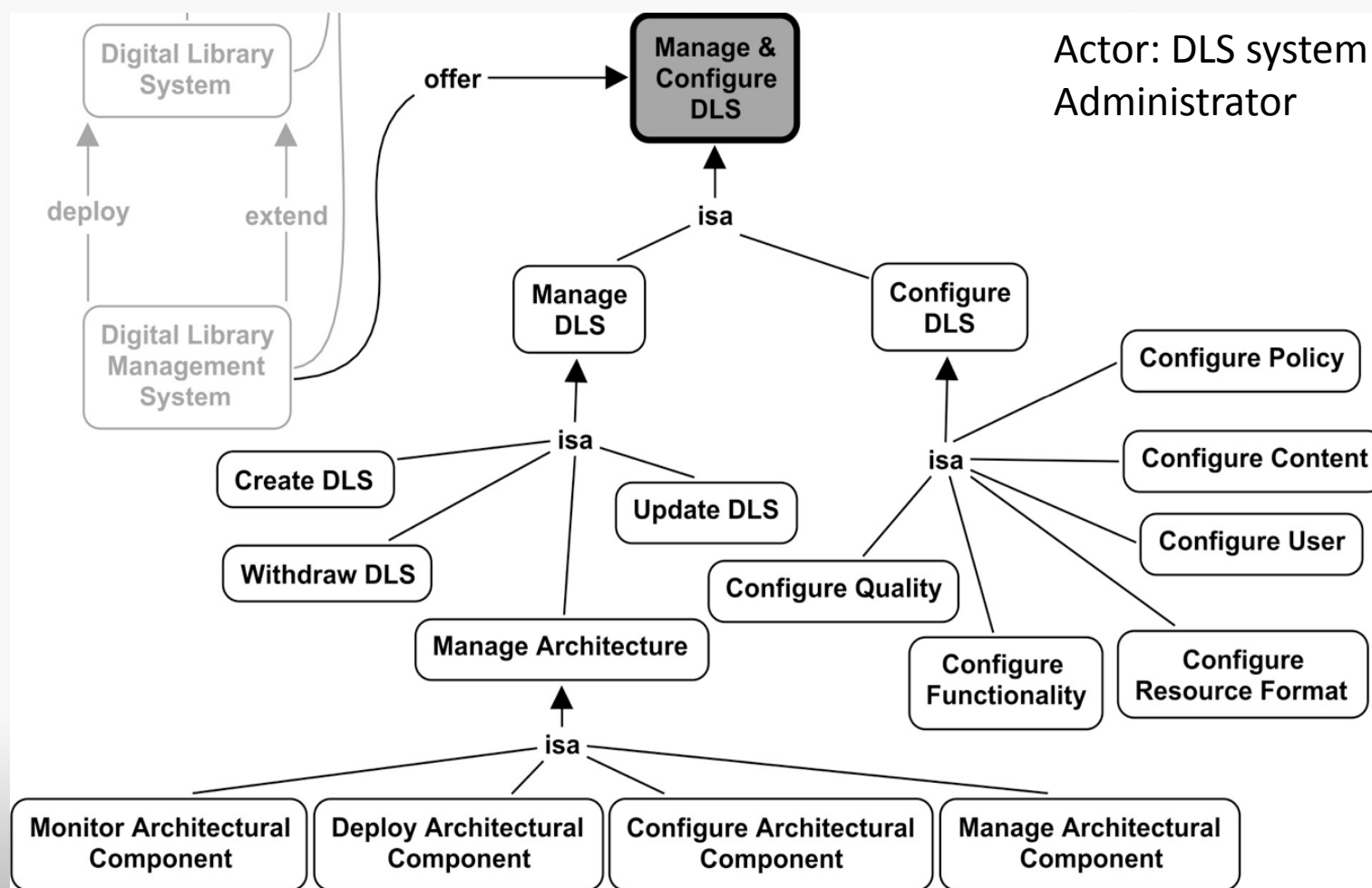
Collaborate



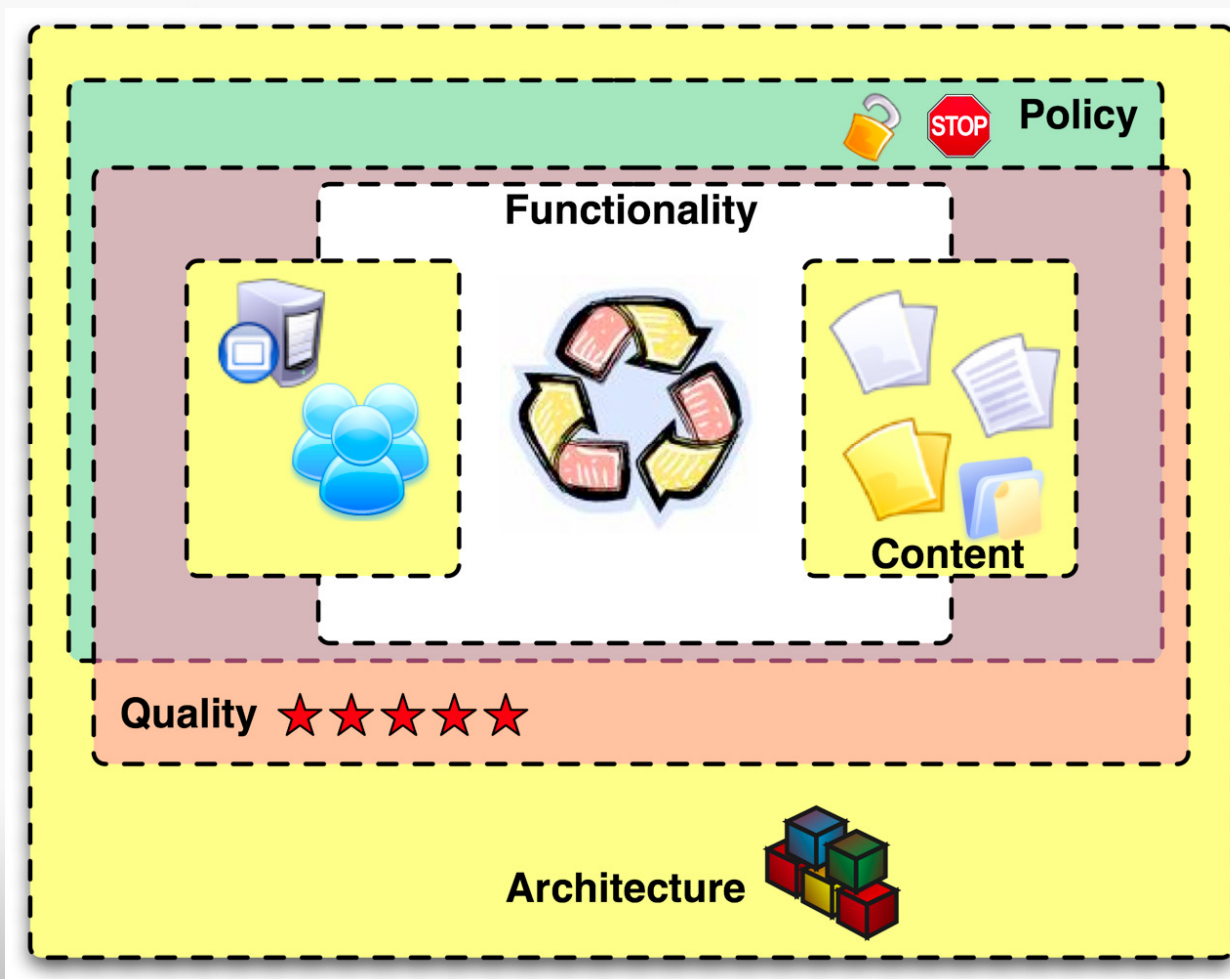
Manage DL

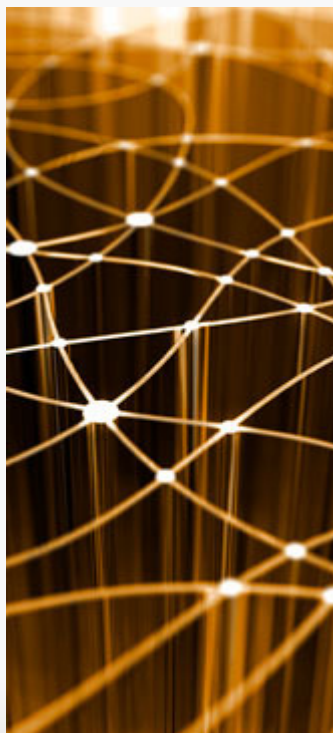


Manage & Configure DLS



Functionality is the core





Thank you
wiki.dlorg.eu/index.php/Reference_Model

