



Architecture domain

Leonardo Candela

6th October 2010

DL.org Autumn School – Athens, 3-8 October 2010



Lecture outline

What is the Architecture

Architecture domain in the Reference Model

Architecture domain interoperability

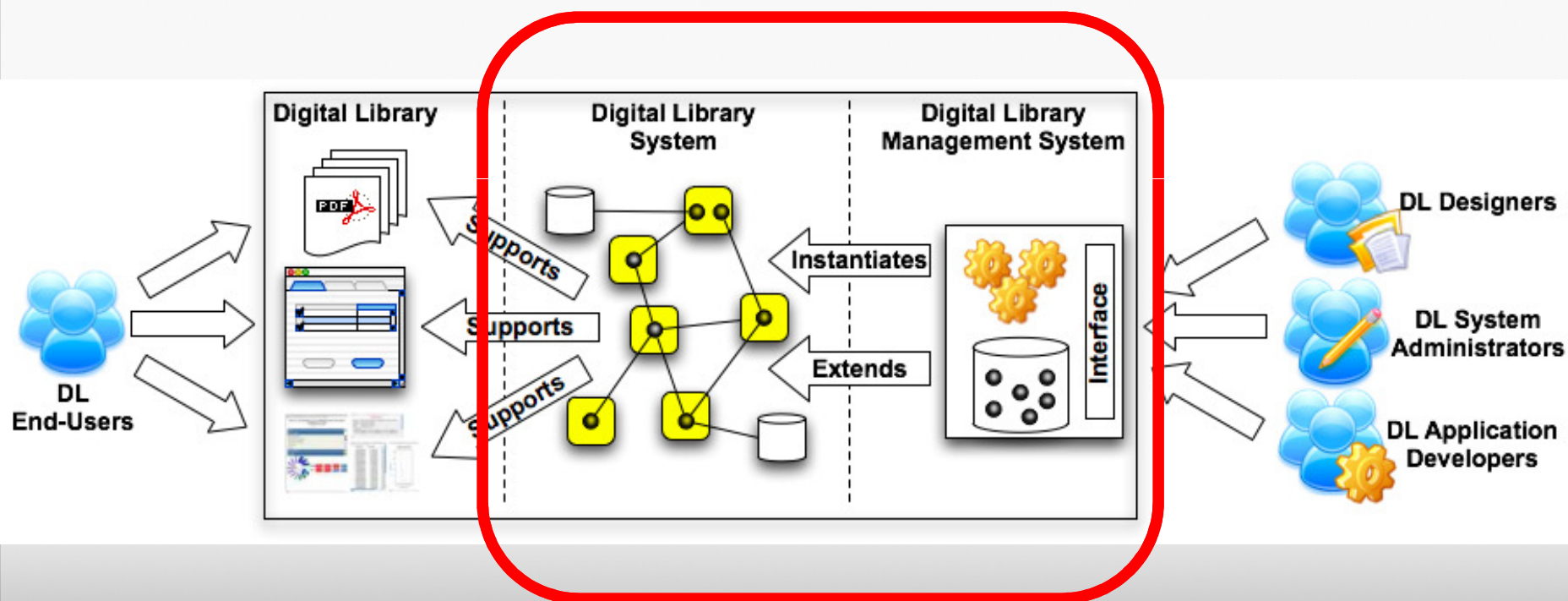
Hands-on Time

Architecture

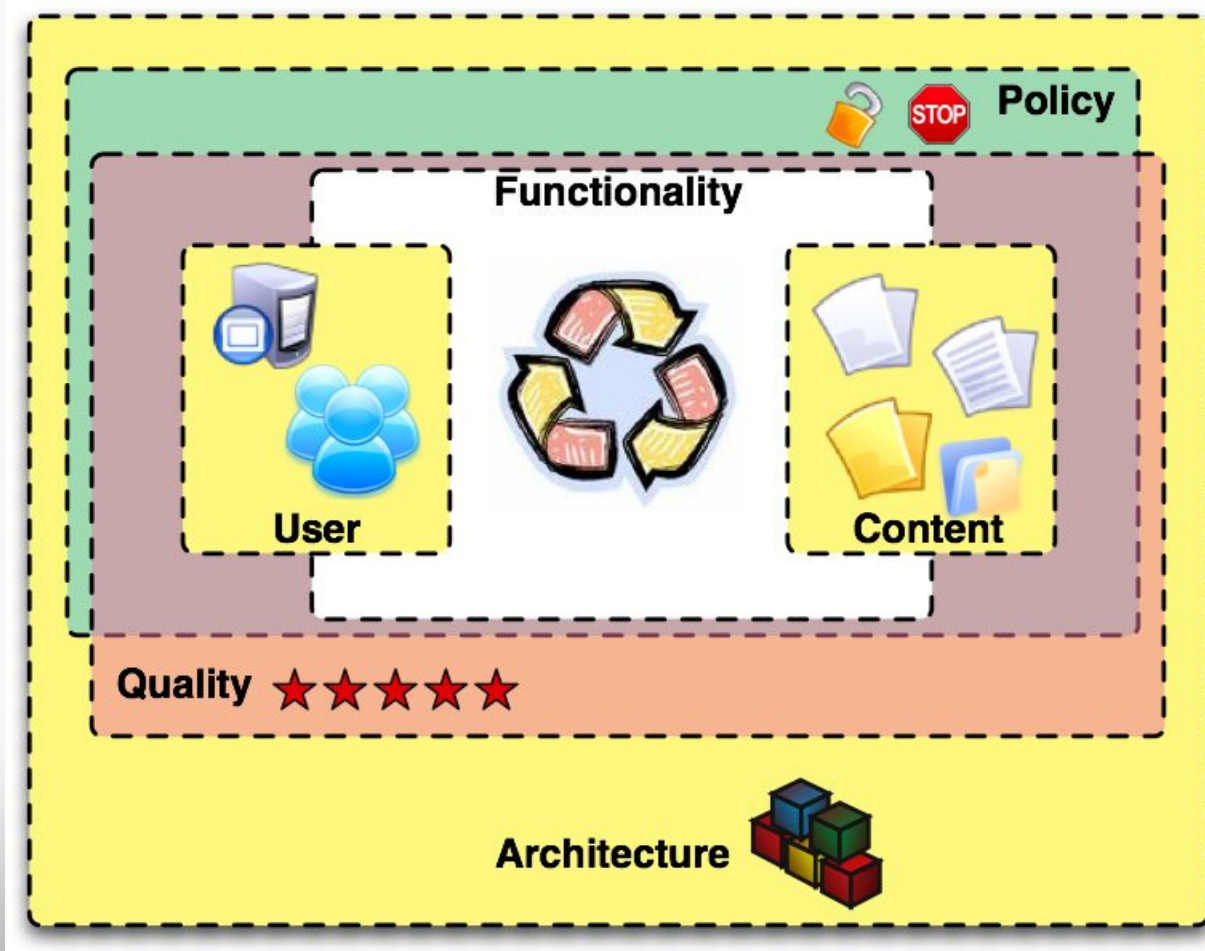
- *Oxford American Dictionary*
 - the art or practice of designing and constructing **buildings**.
 - the style in which a building is designed or constructed, esp. with regard to a specific period, place, or culture : *Victorian architecture*.
 - *the complex or carefully designed **structure of something** : the chemical architecture of the human brain.*
 - *the conceptual structure and logical organization of a **computer or computer-based system** : a client/server architecture.*

A domain for concrete software systems

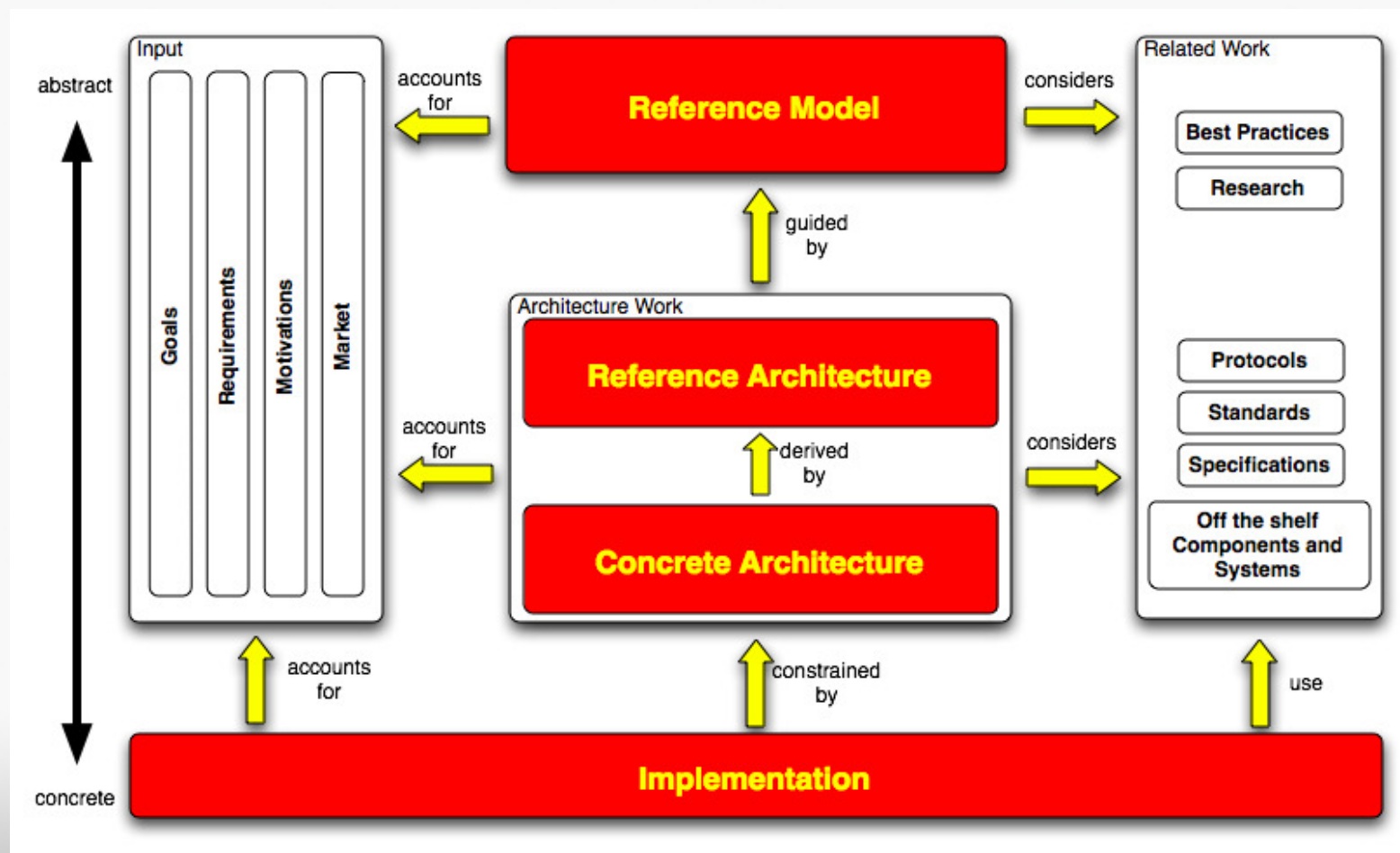
Architecture

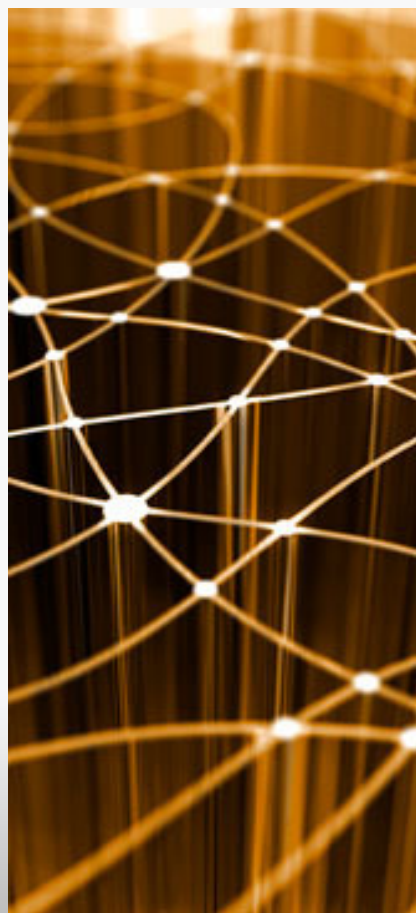


Architecture w.r.t. the other domains



Reference Frameworks





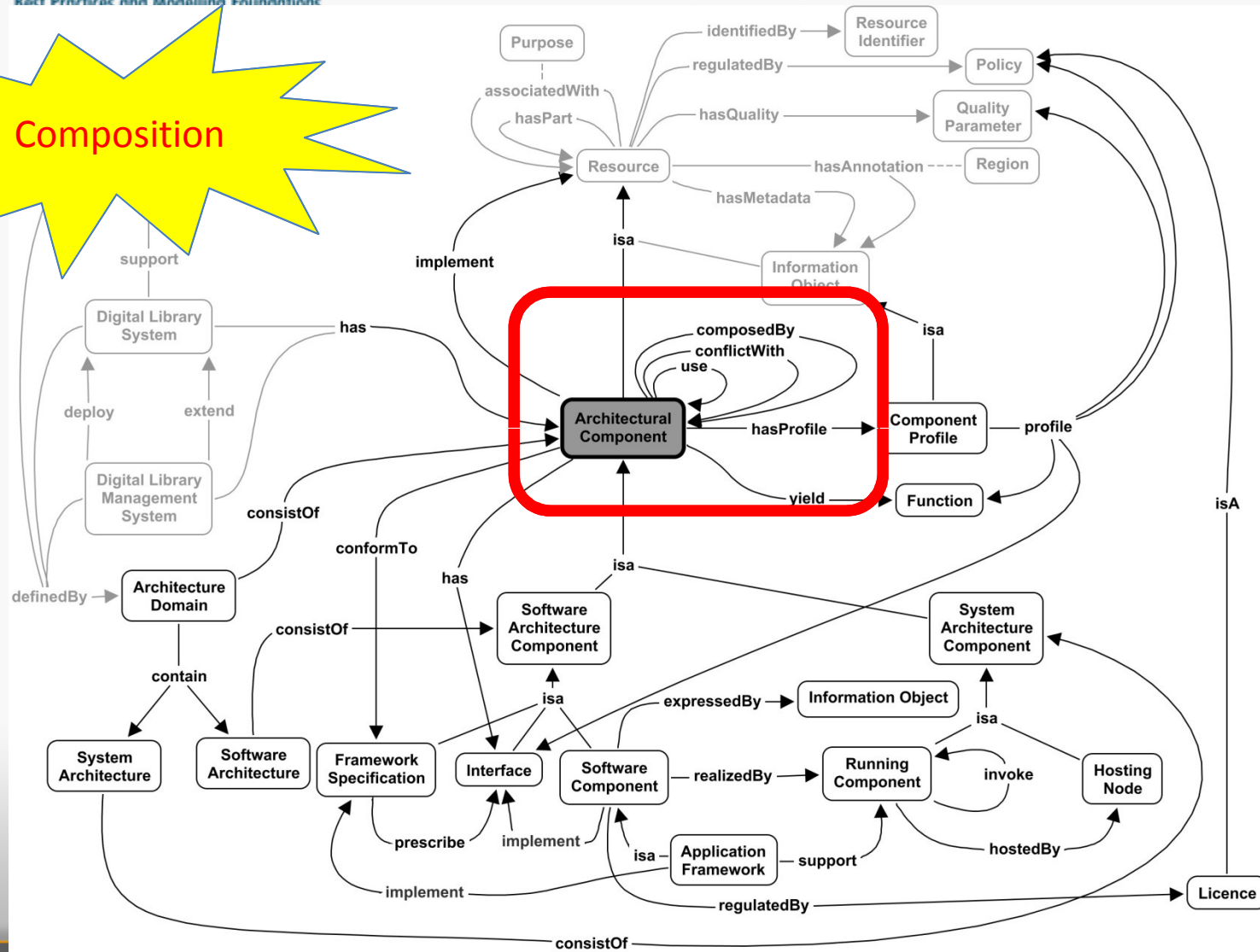
Architecture Domain: the Reference Model

The Architecture Domain in a Nutshell

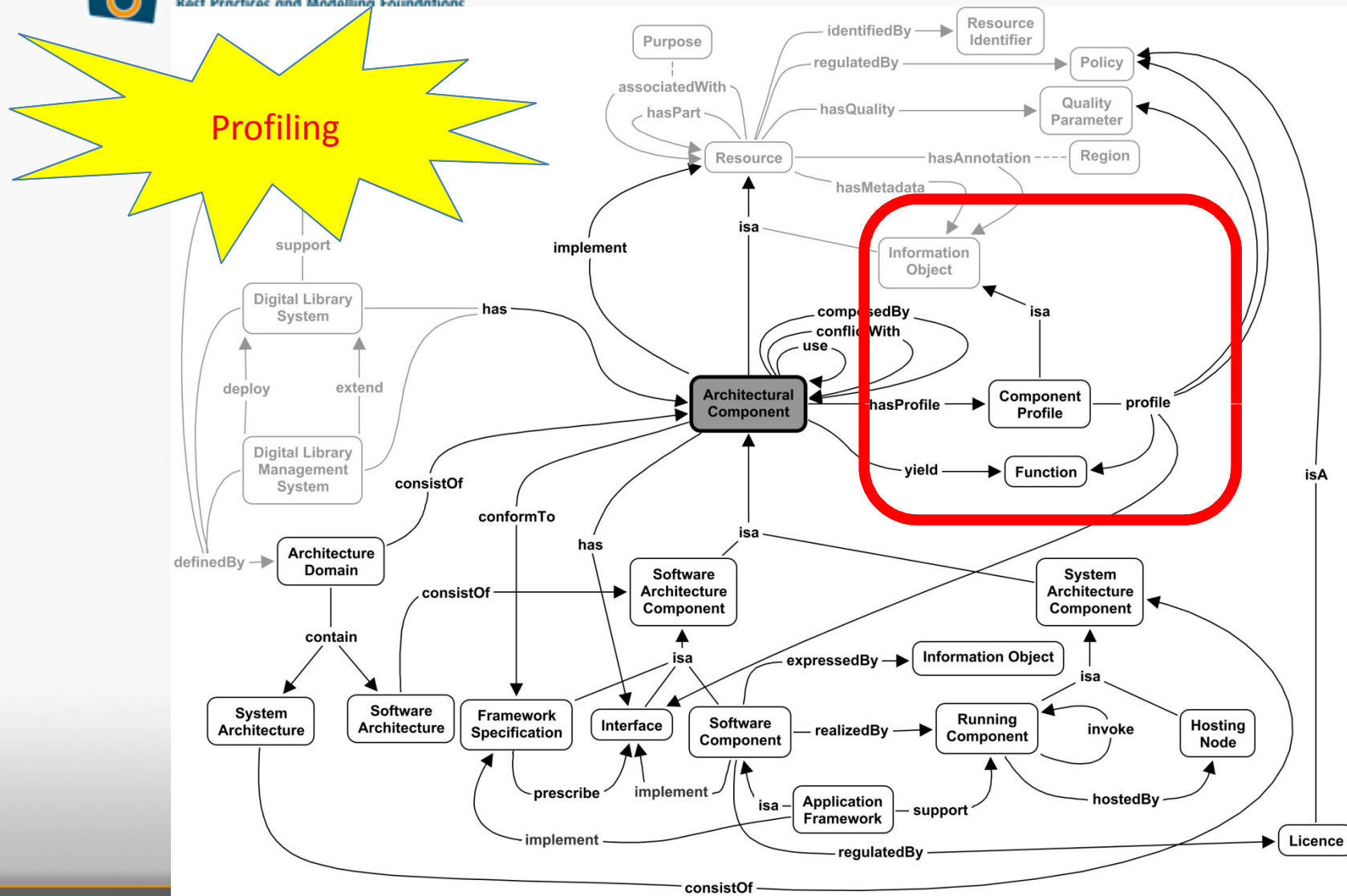
- Architecture of a system (DLS or DLMS) is the organization or structure of its architectural components
 - may be composed of **smaller components**
 - have a **component profile** (characterization)
 - interacting each other through their **interfaces**
 - conform to a **framework specification**
- **System Architecture**
 - System Architecture Component (Hosting Node and Running Component)
- **Software Architecture**
 - Software Architecture Component (Software Component, Interface, Framework Specification)

The Architecture Domain Map

Composition



The Architecture Domain Map

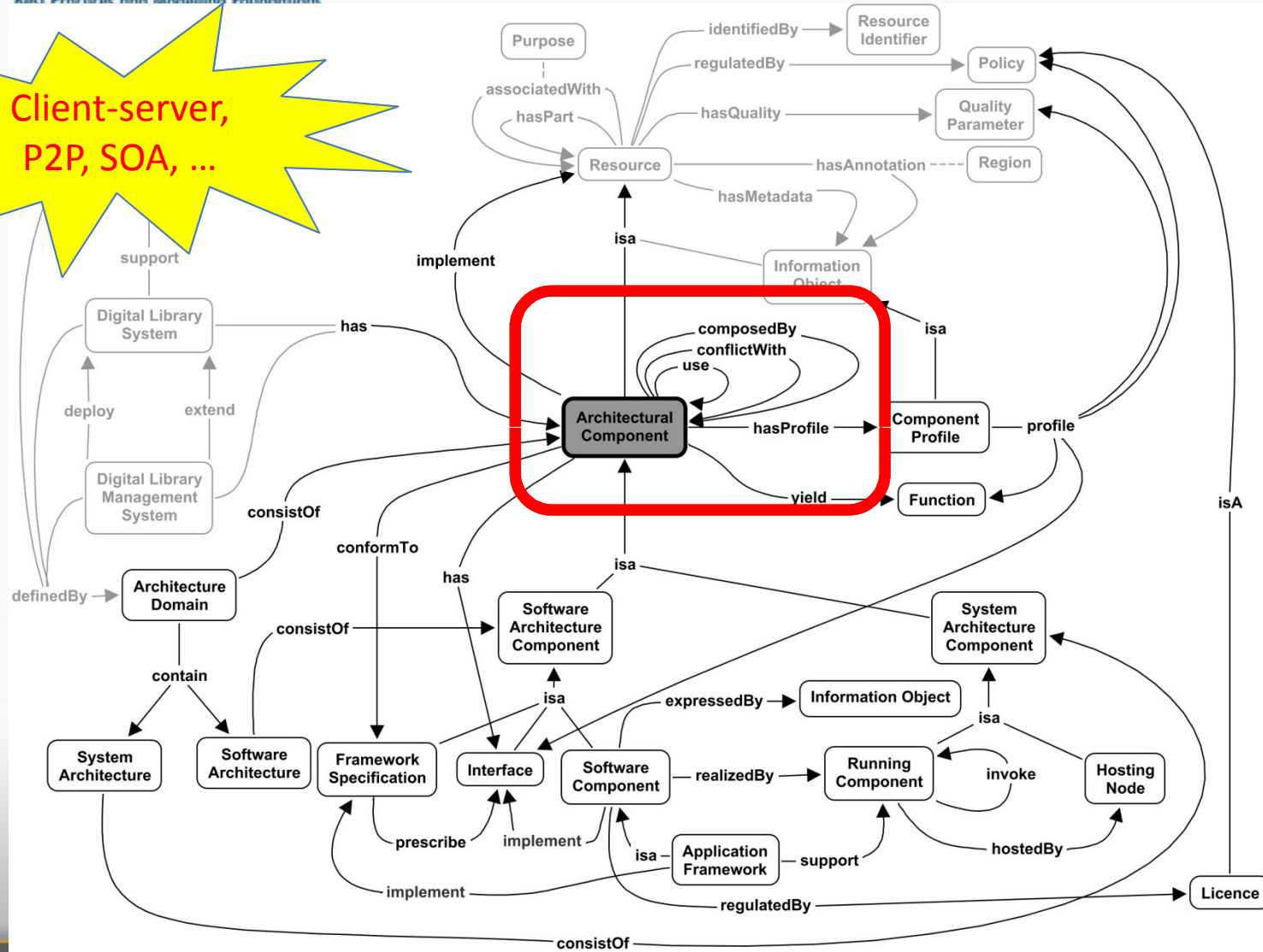


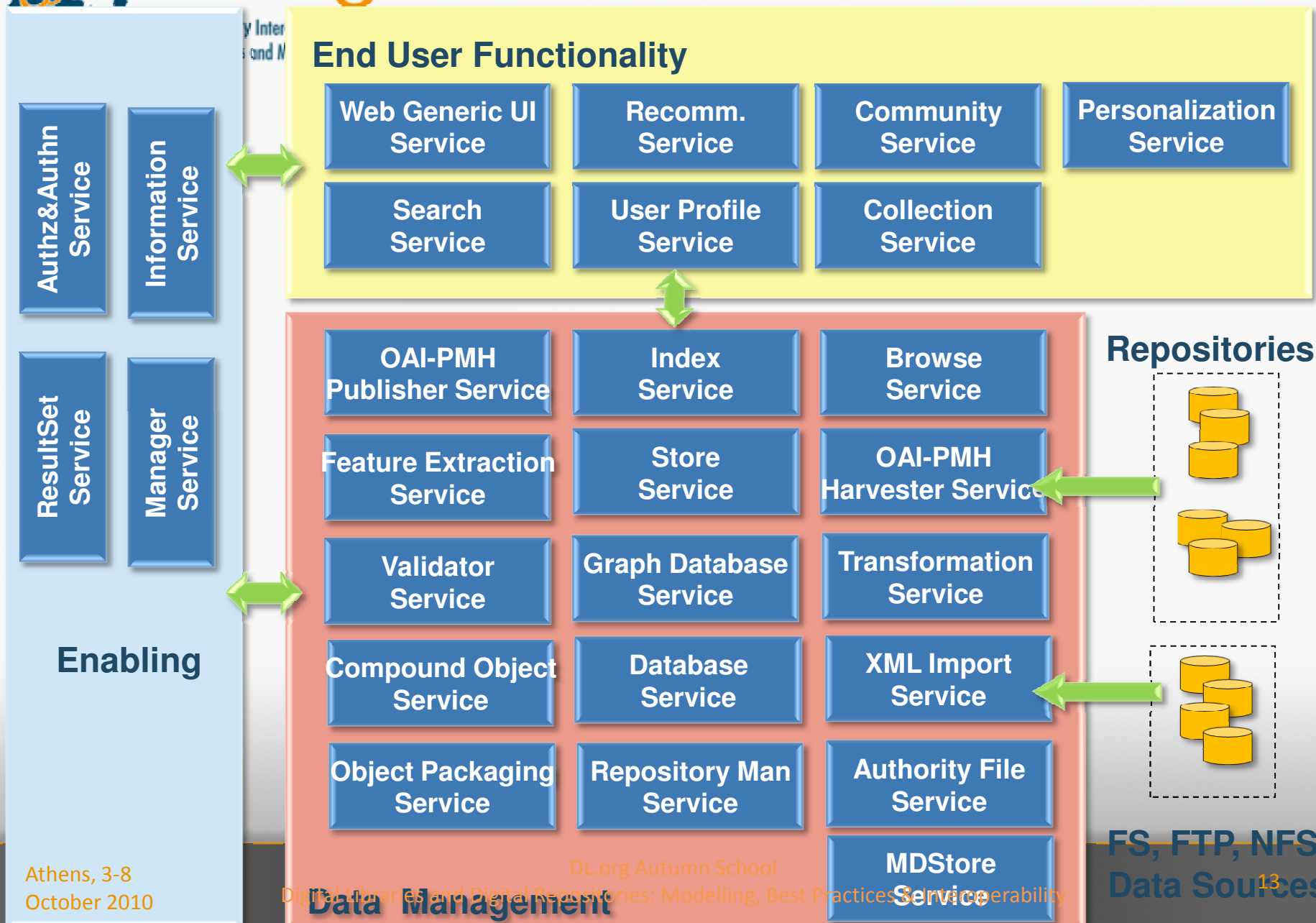
Component-based Approach Goodies

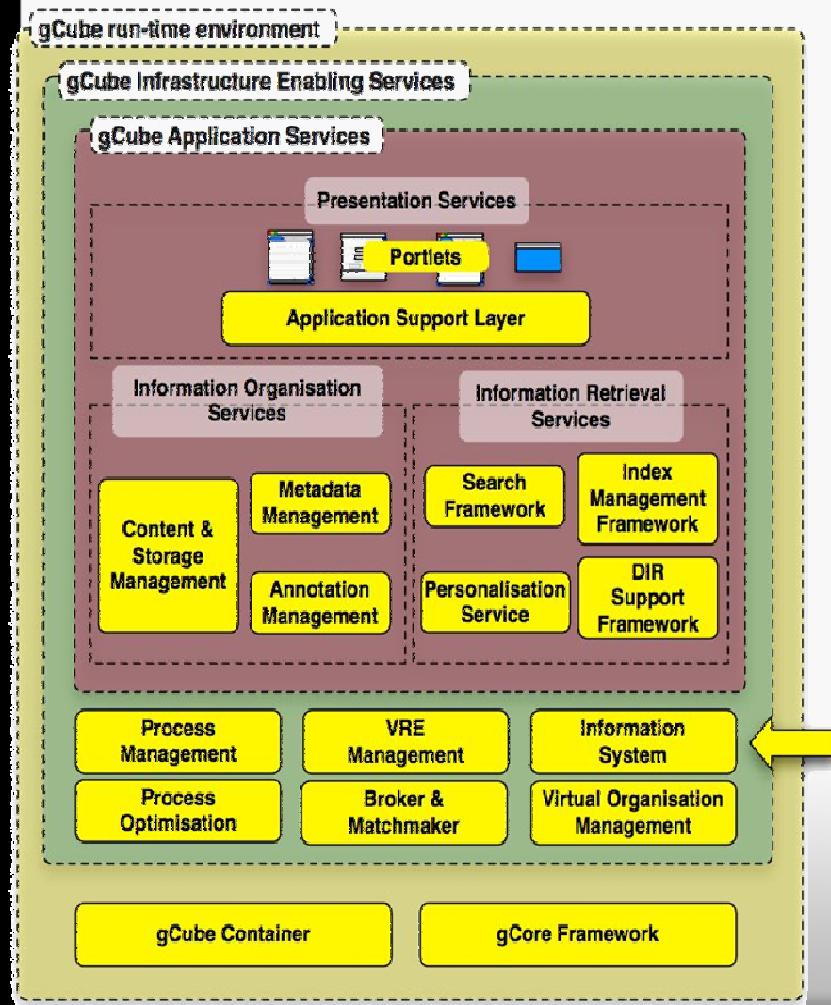
- The system is **assembled** from discrete executable components, which are **developed and deployed somewhat independently** of one another, and potentially by different players
- The system may be **upgraded with smaller increments**, i.e. by upgrading some of the constituent components only. In particular, this aspect is one of the **key points for achieving interoperability**, as upgrading the appropriate constituents of a system enables it to interact with other systems
- Components may be **shared** by systems; this creates opportunities for reuse, which contributes significantly to lowering the development and maintenance costs and the time to market
- Though not strictly related to their being component-based, component-based systems tend to be **distributed**

The Architecture Domain Map

Client-server,
P2P, SOA, ...







Presentation Services

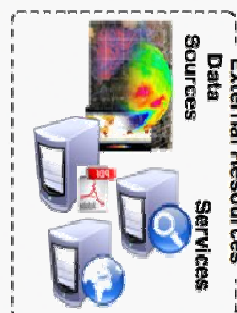
- ✓ Application Support Layer
- ✓ User Portlets
- ✓ Administrative Portlets
- ✓ Desktop clients

Information Retrieval Services

- ✓ Metadata Indexing
- ✓ Content Indexing
- ✓ Personalisation
- ✓ Content Source Description & Selection
- ✓ Data Fusion
- ✓ Search

Information Organisation Services

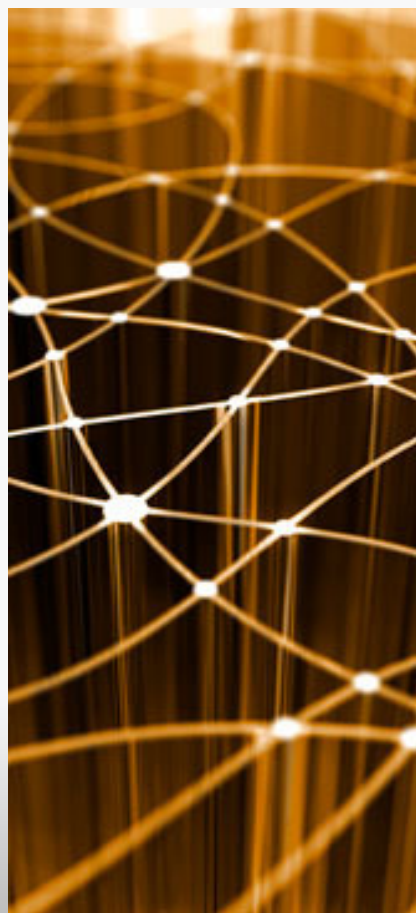
- ✓ Storage Management
- ✓ Collection Management
- ✓ Content Management
- ✓ Metadata Management
- ✓ Archive Import
- ✓ Metadata Brokerage
- ✓ Annotation Management
- ✓ Content Transformation
- ✓ Ontology Management



Enabling Elements

- ✓ Runtime Environment provision (gCore/gHN)
- ✓ Infrastructure Management, Monitoring and Self-reorganisation
- ✓ VRE Management
- ✓ VO and Security Support Services
- ✓ Process Execution

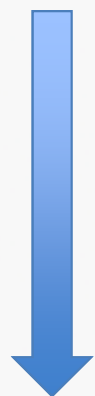




Architecture Domain Interoperability

Architecture

Interoperability: what it is



	Software Component	System Component
Standalone/proprietary	✗	✓
Standards Adoption	✗	✓
“Public” Specification	✗ ✓	✓

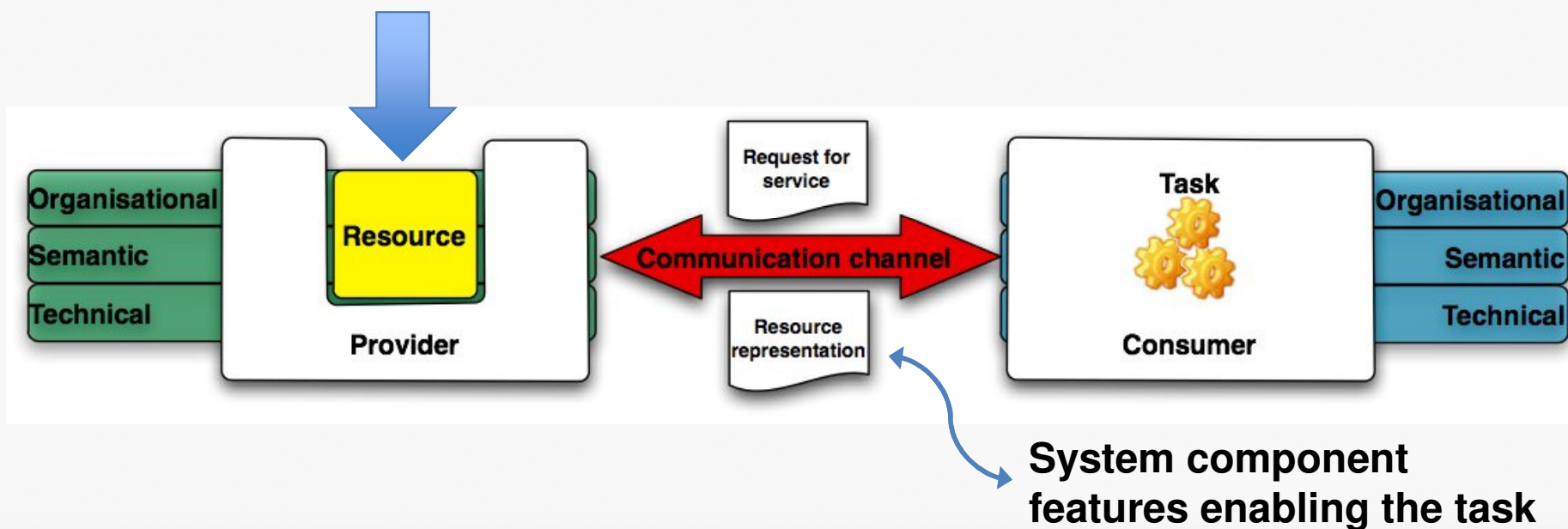
Provider Costs
Usage Scenarios

Integration

Interoperability

Architecture domain interoperability

System Component



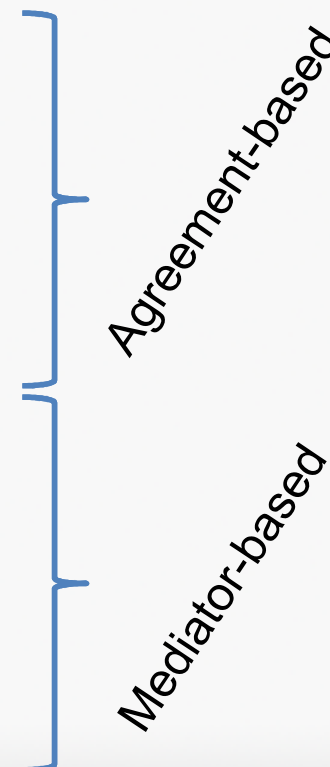
Architecture Component Feature: Component Profile

- A explicit characterization of the Architectural Component
- What is in a profile?
- Many commonalities with metadata
 - inherit from other domains
 - organisational, semantic and technical

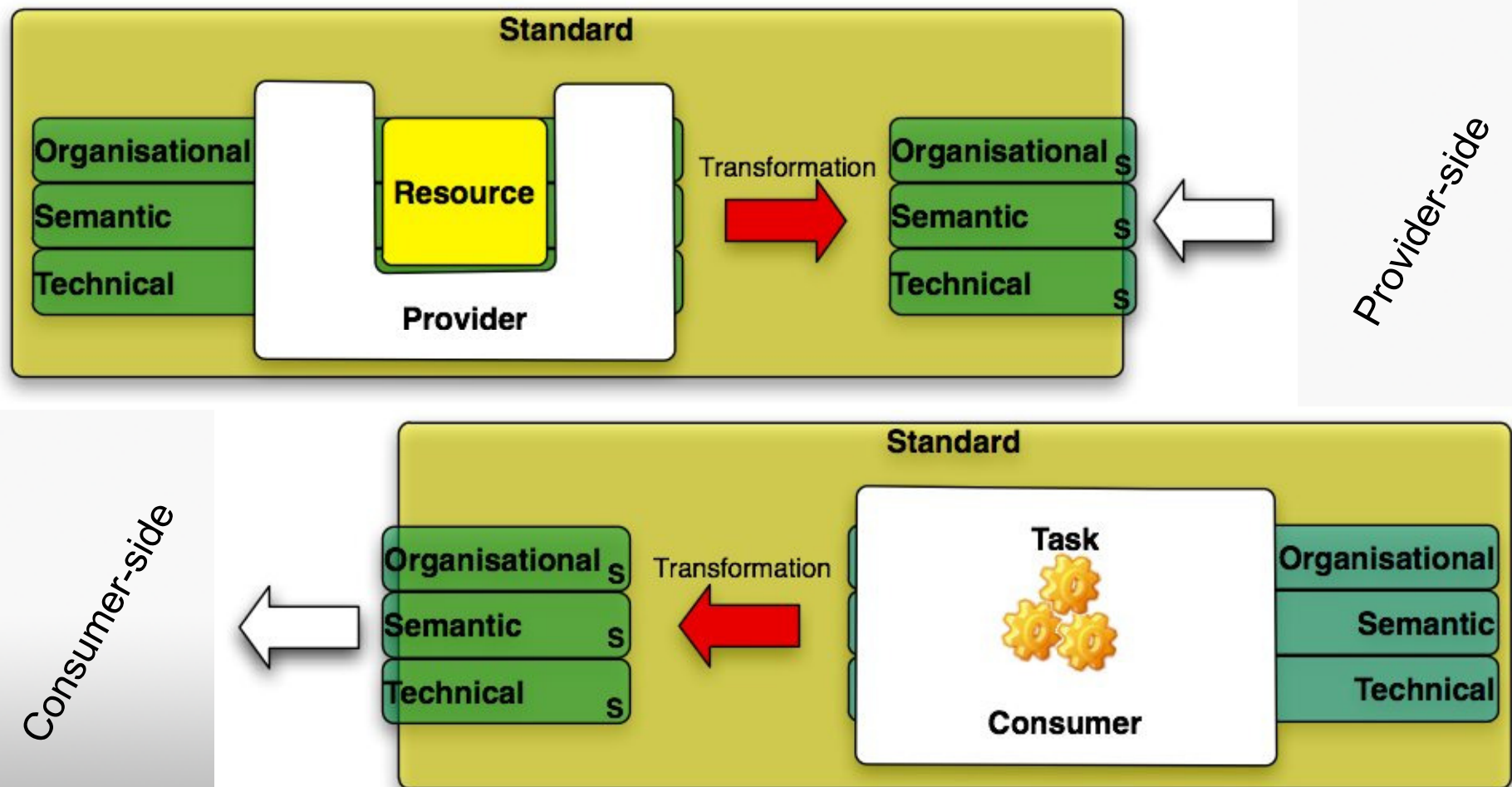
Content
User
Functionality
Policy
Quality
Architecture

Application Framework and Architectural Interoperability Approaches

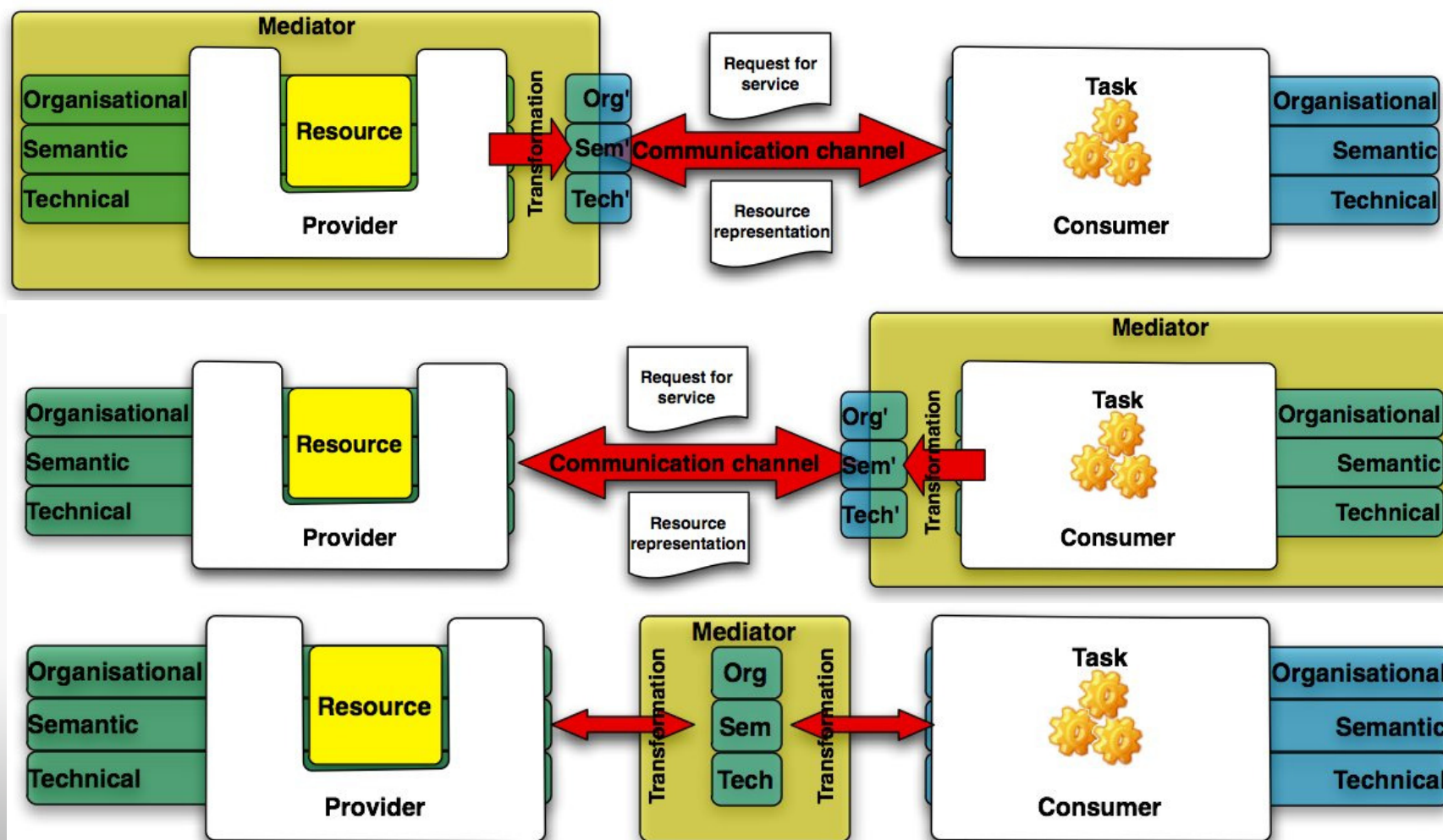
- (de facto) **Standard** (the oldest one!)
 - e.g. Z39.50, SRU, OAI-PMH, OAI-ORE, SOAP+WSDL
 - very effective if agreed, autonomy Infringement
- **Families of standards**
 - multiple standards, negotiation
 - alleviates the autonomy infringement
- **Wrappers / Mediators / Proxies**
 - interoperability machinery outside participants
 - strong in supporting autonomy
- **Specification-based / profile-based**
 - no prior arrangement, dynamic binding
 - support autonomy, requires standard / agreement
- **Blending Solutions**

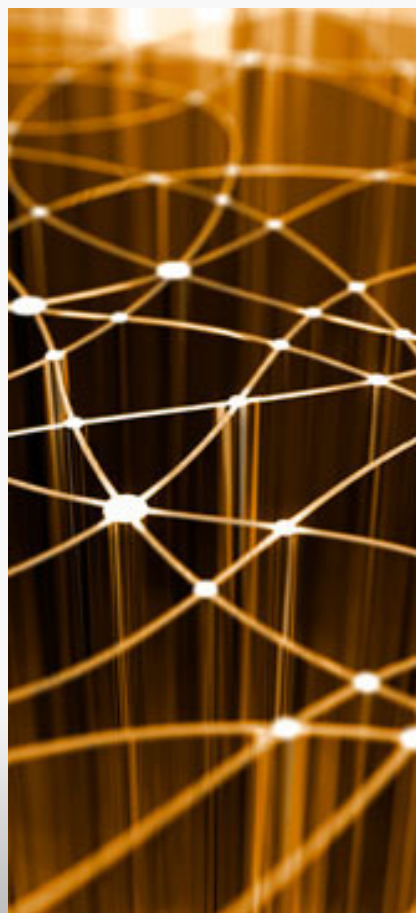


Agreement-based approaches architectures



Mediator-based approaches architectures

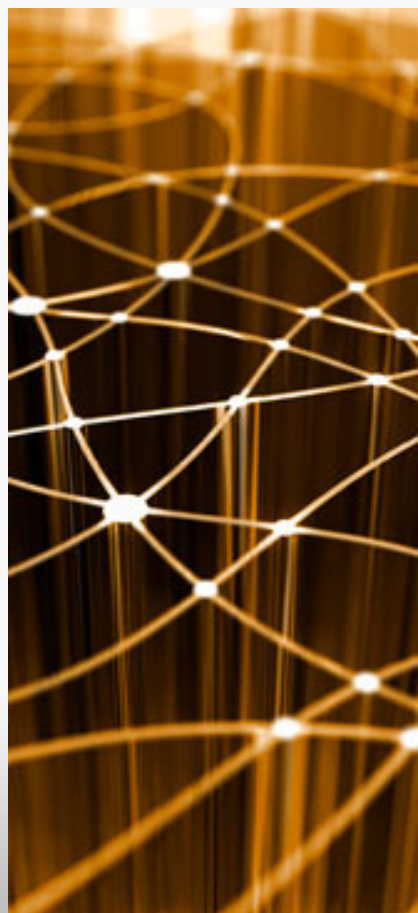




Architecture Domain: Hands-on Time

Exercises

- Identify and produce RM Architecture [& Content] domain enhancements
 - Each enhancements should be equipped with a motivation
 - Enhancements might be on the introduction of new concepts and/or relationships, on the revision of existing definitions as well as on exemplars
- Select one (or more) DL “system” and describe its Architecture [& Content] domain by relying on the Reference Model;
- Work on the Architecture [& Content] domain part of the interoperability scenario;



Thank you