DL.org

Digital Library Interoperability,
Best Practices and Modelling Foundations

Architecture Interoperability

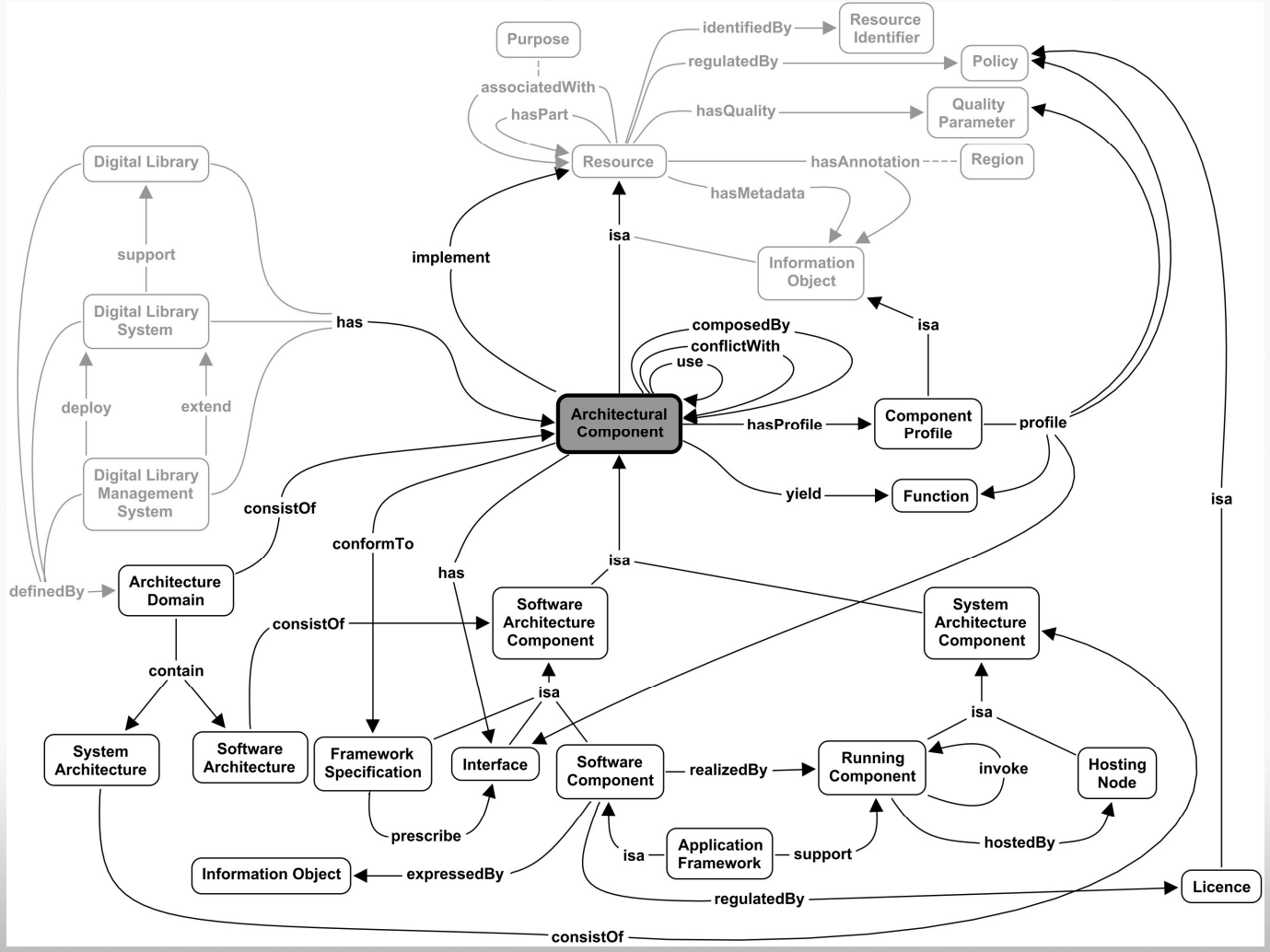
Pasquale Pagano ➡ Leonardo Candela

CNR-ISTI

The Architecture Domain

- 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



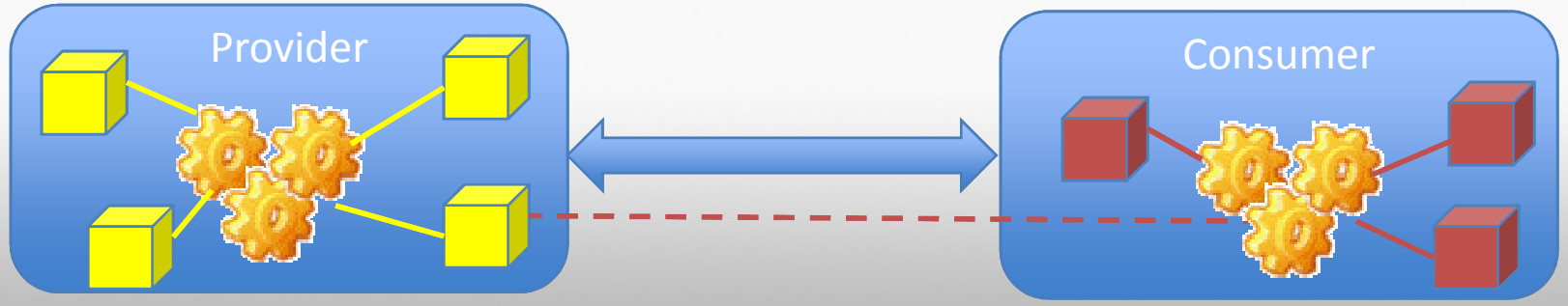
DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

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**

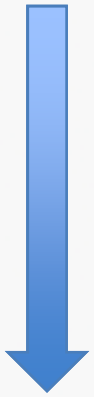
Architecture Interoperability: what is it

- **Two software systems** are willing to "share" an architectural component
  - **provider** is the owner of the component
  - **consumer** is interested in "using" that component
- Sharing requires a **common understanding** of some component **features**
- Sharing requires **communication** between provider and consumer



Architecture Interoperability: what it is (cont.)

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



Provider Costs
Usage Scenarios

Integration

Interoperability

Architecture Interoperability: the D4Science and DRIVER case

- D4Science and DRIVER have many commonalities
  - deliver a Service-oriented Infrastructure
    - catalog of Web Services, operation of SOA Applications, management of other resources for running services
  - adopt common design principles and patterns
  - adopt many common standards
- … but
  - rely on two software frameworks (D-Net & gCube)
  - result to be partially interoperable

… "details" make the difference …

Architecture Interoperability: the D4Science and DRIVER case (cont.)

- Resource Profile
  - DRIVER profile is equipped with a *blackboard*
  - D4Science profile is equipped with *software packages*
- Infrastructure Management
  - DRIVER provides for an orchestrator system that dynamically combines existing functionality to implement the desired behavior and QoS
  - D4Science provides for a management system that supports the definition, automatic deployment, and execution of applications and the maintenance of the required QoS

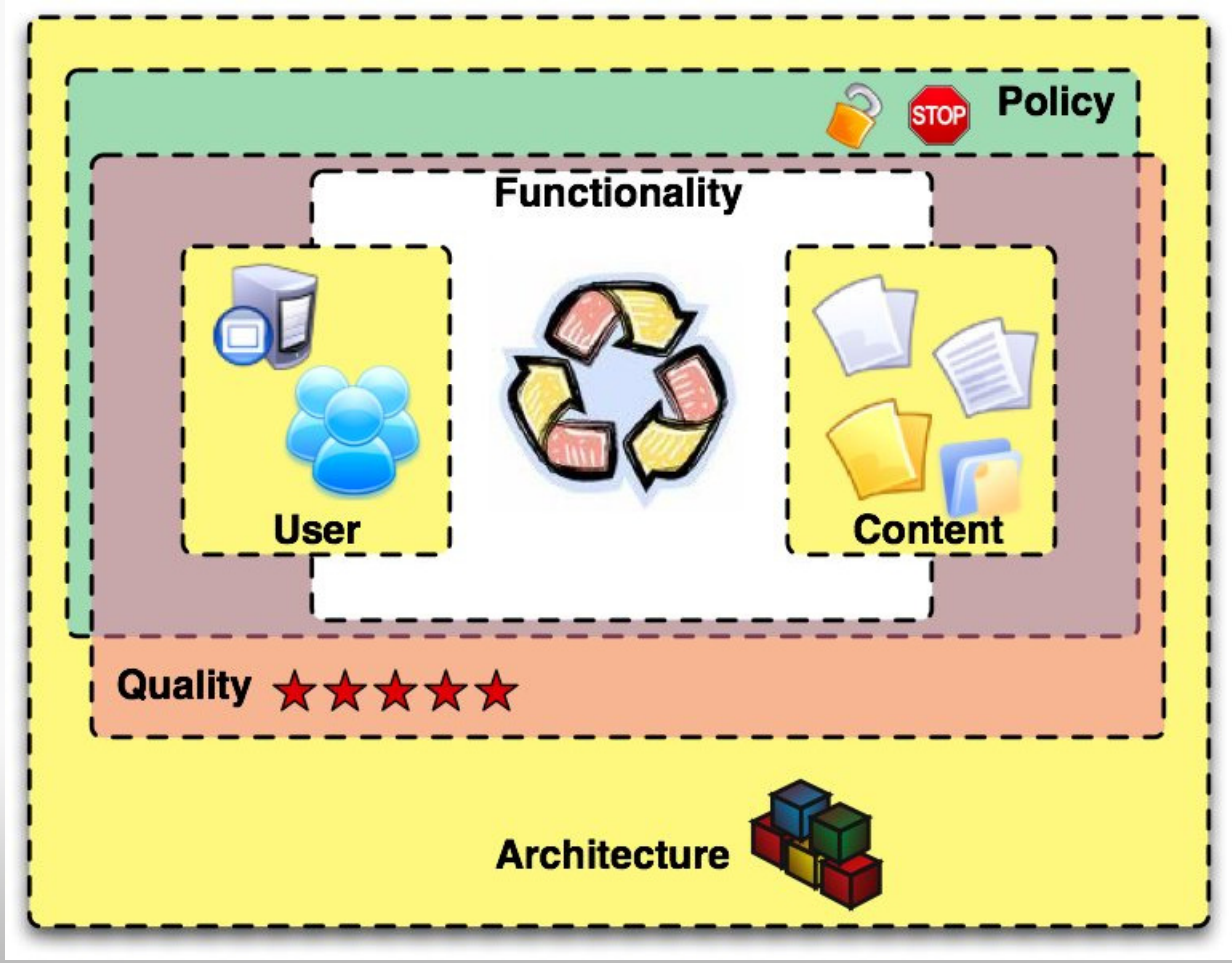
Architecture Component Feature: Component Profile

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

Application Framework and Architectural Interoperability Approaches

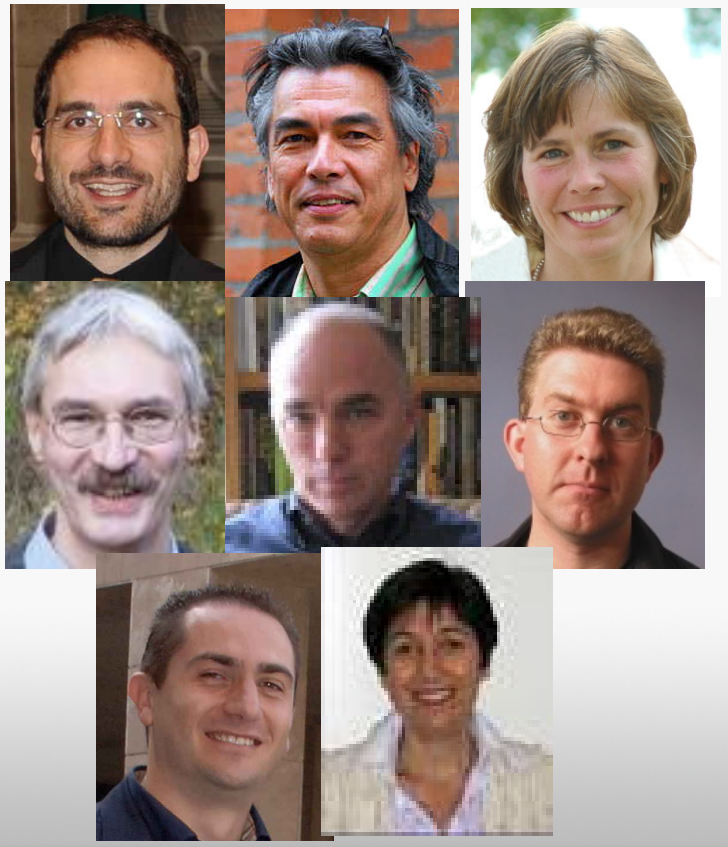
- (de facto) **Strong Standard** (the oldest one!)
  - e.g. Z39.50, SRU, OAI-PMH, OAI-ORE, SOAP+WSDL
  - very effective if agreed, autonomy Infringement
- **Families of standards**
  - multiple standards, negotiation
  - alleviates the autonomy infringement
- **Wrappers / Mediators / Proxies**
  - interoperability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

Architecture w.r.t. the other Domains



The DL.org Architecture Working Group

- 8 (+) members representing several projects



The DL.org Architecture Working Group Mission and Scope

Mission

- Identify and prioritize Architecture-oriented interoperability issues (✓)
- Analyze existing approaches (ongoing)
- Propose patterns and best practices

Scope

- ***Component Profile***, i.e. the “metadata”
- ***Application Framework***, i.e. the “glue”

… content storage and content access

Working Group Ongoing Activities and Expected Outcomes

- Design and testing of interoperability approaches in concrete scenario
  - e.g. D4Science -> Fedora
- Profile schemata promoting diverse interoperability enabling power
- Reference Architectures for Interoperability-oriented Application Framework

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Summing Up

- Architecture interoperability can be seen as the lowest / foundational level of interoperability
- Solutions can be borrowed from other [DL.org] domains (e.g. WSDL, WSRF)
- Standards are solutions to some extent only (e.g. OAI-PMH, OAI-ORE, WSDL)
  - guidelines and best practices are needed

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

workinggroups.wiki.dlrg.eu/index.php/Architecture_Working_Group

architecture@dlorg.eu

QUESTIONS?