

# Accessing and Manipulating Ontologies using Web Services

Olivier Dameron, Natalya F. Noy,  
Holger Knublauch, Mark A. Musen

SMI - Stanford University

# Context

- Semantic Web
  - Ontology scope
  - Web Services scope
- Ontology and WS address complementary needs
- acceptance in industry

# Current limitations: WS need semantic descriptions

- WS operate on data that conform to an (implicit) understanding shared between the provider and the client
  - WSDL, UDDI
  - Still require human intervention
    - for composition
    - for interfacing
- Need explicit description of
  - What WS do
  - What they operate on

# Current limitations: WS need semantic descriptions

- WS operate on data that conform to an (implicit) understanding shared between the provider and the client
  - WSDL, UDDI
  - Still require human intervention
    - for composition
    - for interfacing
- Explicit description of
  - What WS do
  - What they operate on

**OWL-S**  
**ONTOLOGIES**

# Current limitations: WS need ontology-manipulation tools

- access to semantic descriptions
- process of semantic descriptions
- WS also need to access to these tools

## **OWS** Ontology Web Services

- Ontology manipulation tools
- Implemented as Web Services

# Ontology Web Services

∃ some generic ontology manipulation tools

- standard bricks for leveraging SW development (so that previous investments start to pay off)
- cumulative effect
- allow to build on technologies accepted by the business world (amazon, google...)
- for human AND for programs (automation)

These tools can be implemented as Web Services

# Approach

1. Identify major classes of Ontology manipulation tools

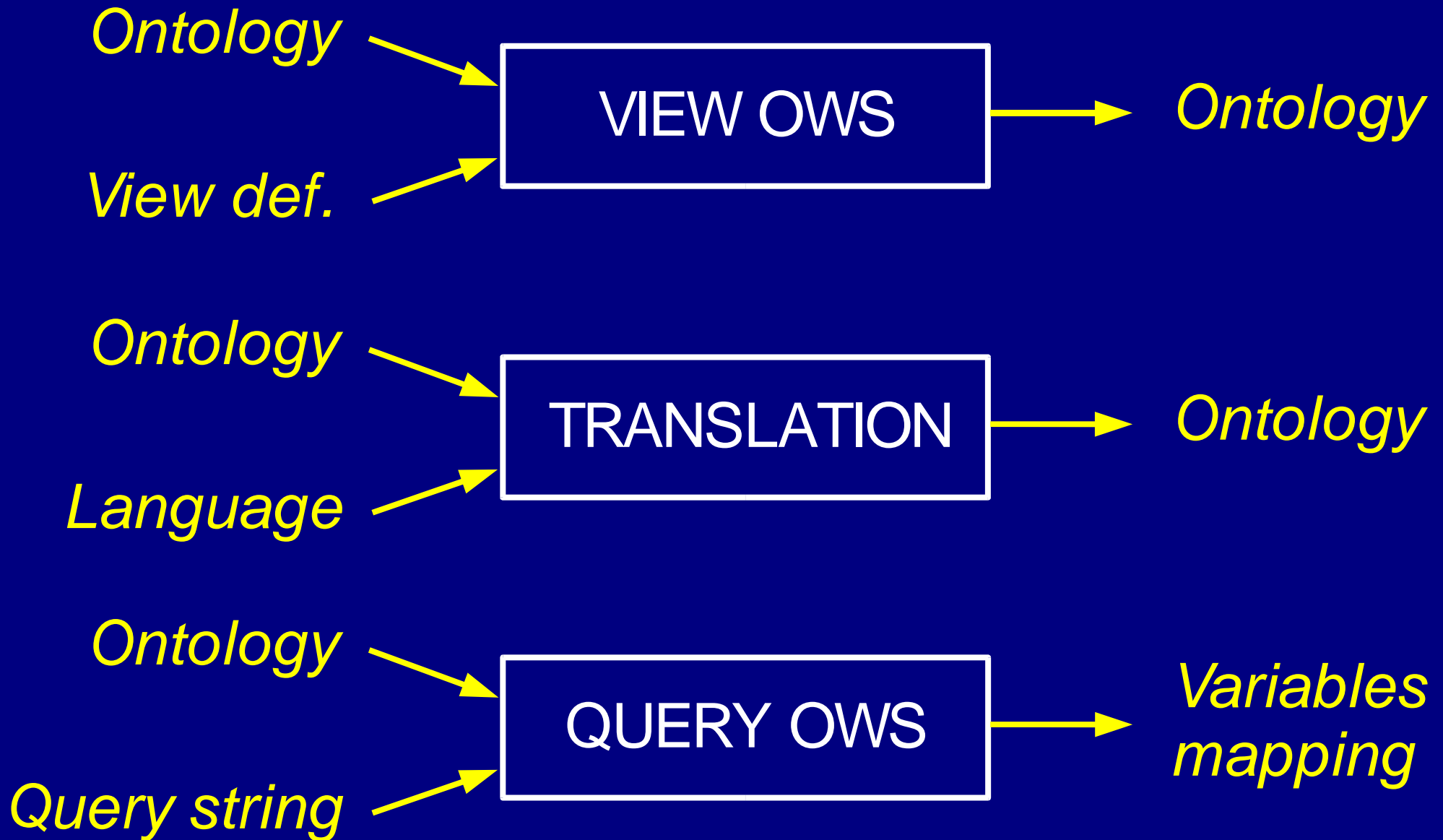
2. Architecture

- Web Services
  - assessing relevance of regular WS
  - achieving interoperability
- OWL-S description of OWS

# OWS Capabilities

- Queries
- Views
- Translations
- Mapping
- Versioning
- Merging
- Reasoning

# OWS Capabilities





OWLClasses 
  Properties 
  Forms 
  Individuals 
  Metadata

**SUBCLASS RELATIONSHIP**

FOR PROJECT: ● OntologyWebServices\_ProfileHierarchy

**ASSERTED HIERARCHY:**

- process:ConditionalResult
- process:ConditionalOutput
- ▶ ○ process:ControlConstruct ≡ process:ProcessC
- process:Participant
- ▶ ○ process:ProcessComponent ≡ process:Contro
- process:Result
- process:ValueOf
- profile:ServiceCategory
- ▶ ○ profile:ServiceParameter
- service:Service
- service:ServiceGrounding
- ▶ ○ service:ServiceModel
- ▼ ○ service:ServiceProfile
  - ▼ ○ profile:Profile
    - ▼ ○ OntologyServiceProfile
      - OntologyMappingProfile
      - OntologyMergingProfile
      - ▼ ○ OntologyQueryProfile
        - OntologyRDQLQueryProfile
        - OntologyRQLQueryProfile
      - ▼ ○ OntologyReasoningProfile
        - OntologyClassificationProfile
        - OntologyTranslationProfile
        - OntologyVersioningProfile
        - OntologyViewProfile
- ▶ ○ shadow-rdf:List
- swrl:AtomList
- ▶ ○ swrl:Variable

**CLASS EDITOR**

FOR CLASS: ○ OntologyRQLQueryProfile (instance of owl:Class)

Name

OntologyRQLQueryProfile

RDFS:COMMENT:

**ANNOTATIONS:**

Property	Value	Lang
rdfs:comment	OntologyQueryProfile p...en	

Asserted | Inferred

**ASSERTED CONDITIONS:**

- NECESSARY & SUFFICIENT
- NECESSARY
- OntologyQueryProfile
- ∃ profile:hasInput (process:parameterValue ∃ "ht"
- INHERITED
- ∃ profile:hasInput (process:parameterType ∃ "htt"
- ∃ profile:hasInput (process:parameterType ∃ "htt"
- ∃ profile:hasInput (process:parameterType ∃ "htt"
- profile:serviceName = 1 [from profile:Profile]
- profile:textDescription = 1 [from profile:Prof...]

**PROPERTIES:**

gyWebServices\_OntologyQuery.owl#RQL"  
 gyWebServices\_OntologyQuery.owl#OntologyQuer  
 gyWebServices\_Ontology.owl#Ontology"  
 \_OntologyQuery.owl#OntologyQuery"  
 CompositeProcess)

**DISJOINTS:**

# Architecture

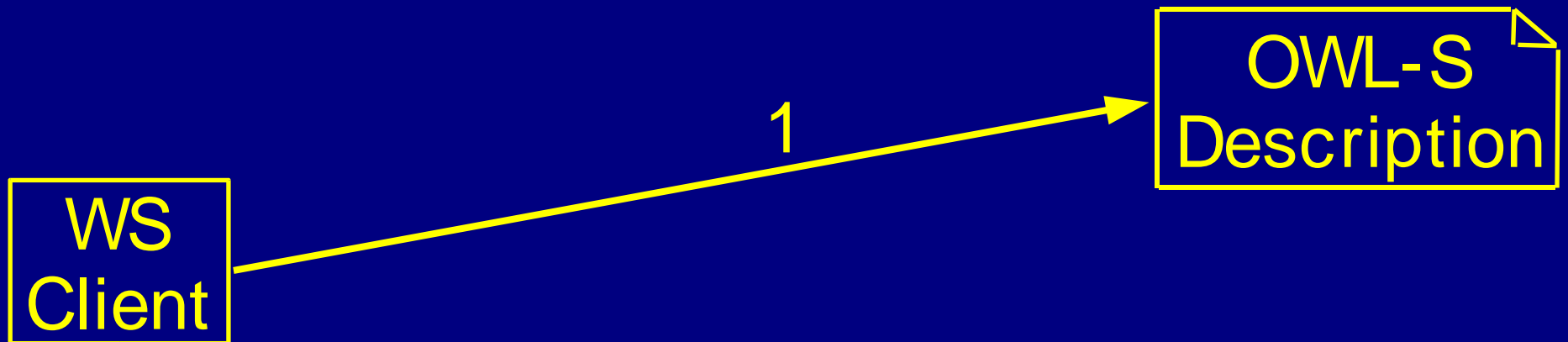
## Semantic description of (regular) Web Services

- (semi) automatic discovery
- automatic invocation
- composition and interoperability

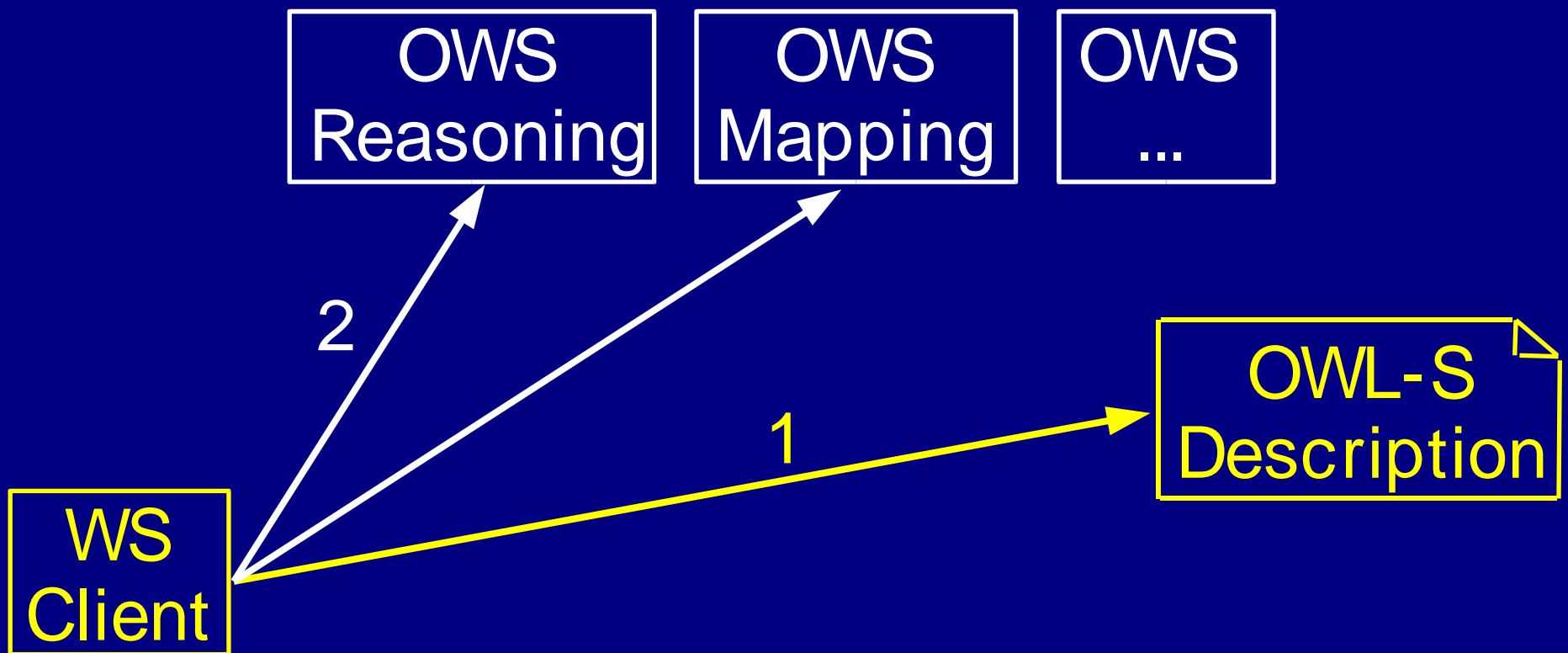
## Interconnection OWL-S / OWS

- hinges on OWL-S
- enable use of OWL-S

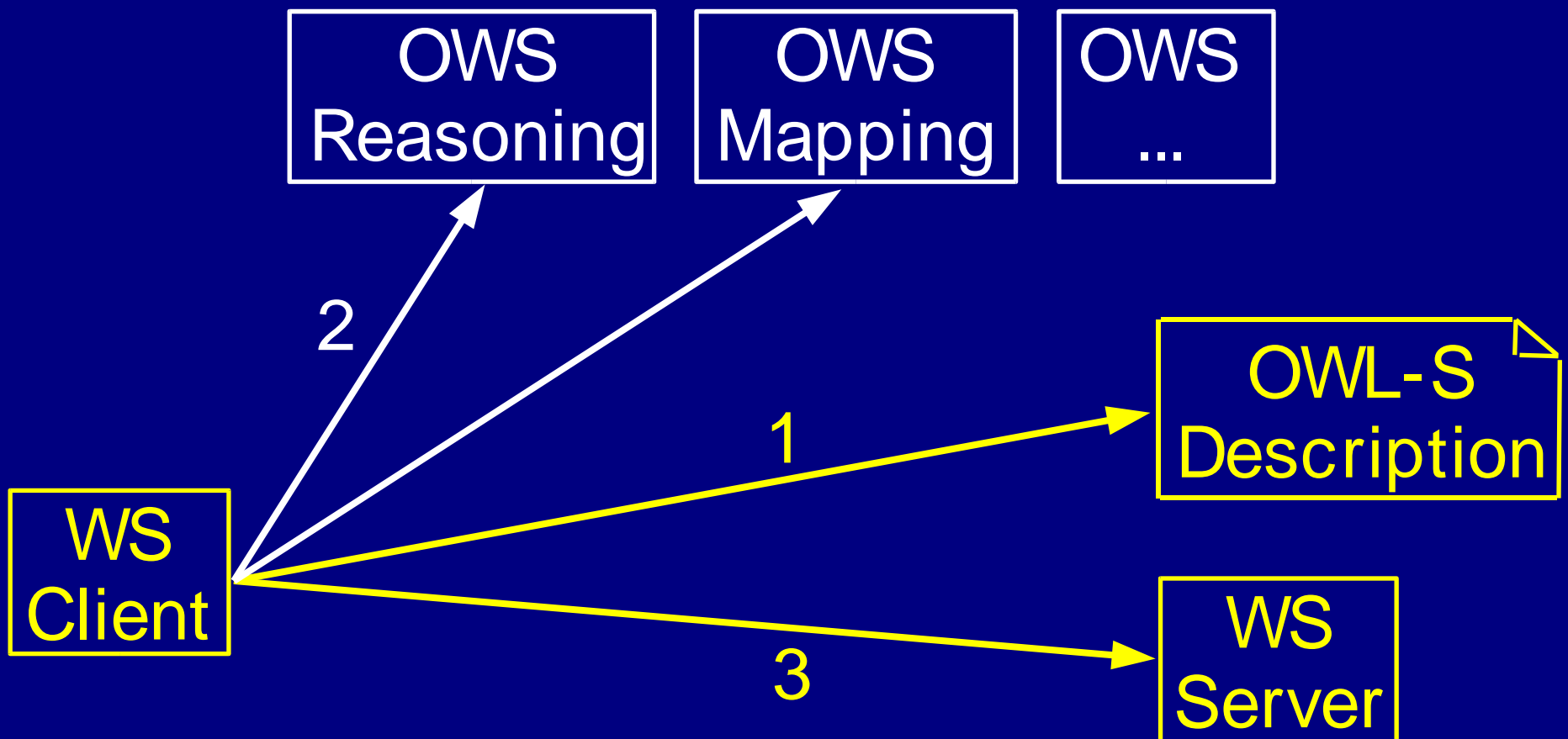
# OWS for assessing WS relevance



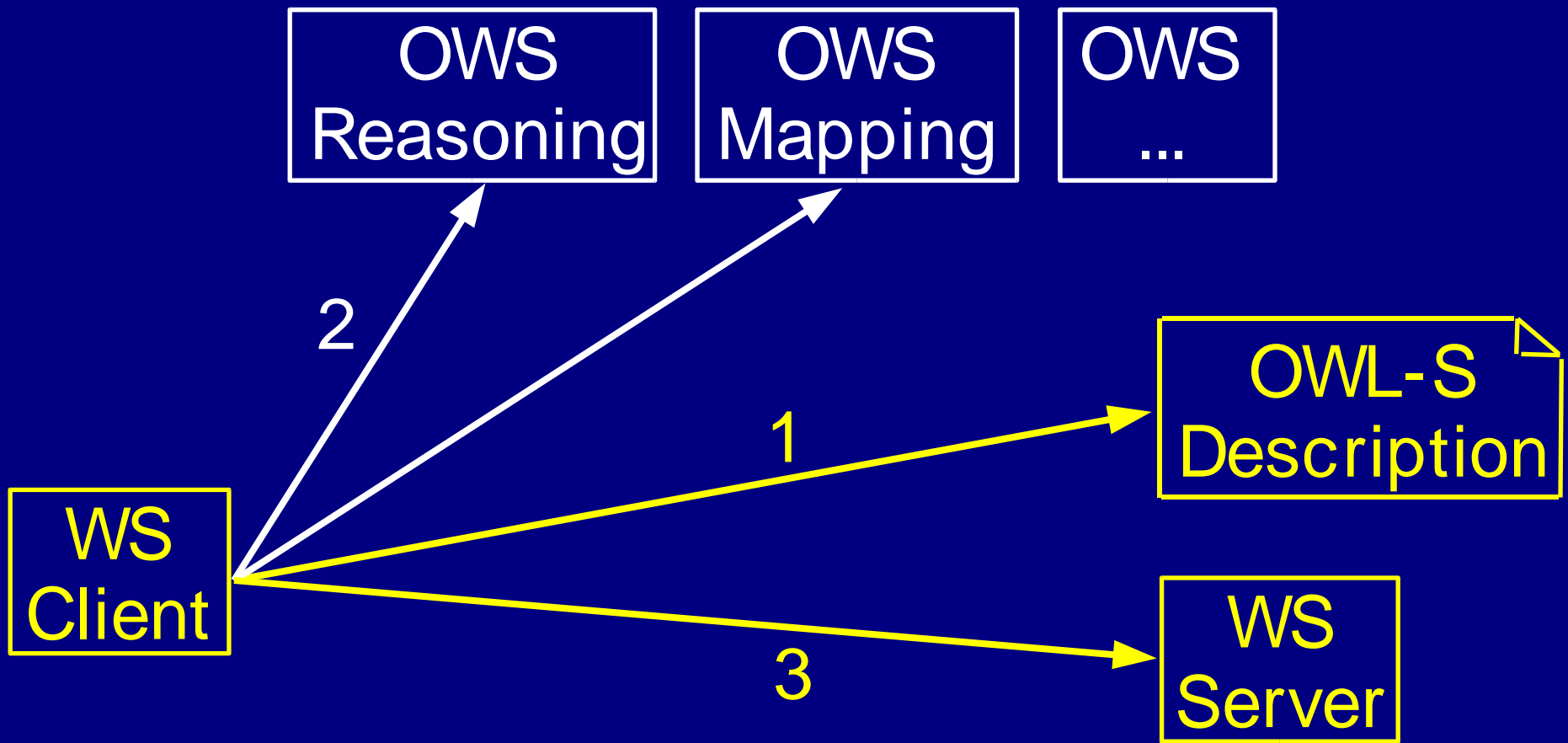
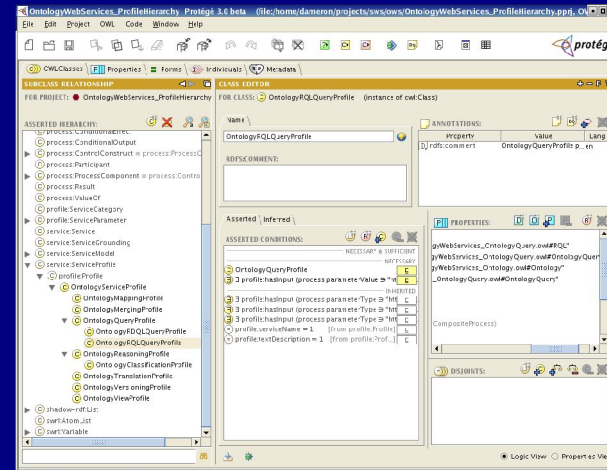
# OWS for assessing WS relevance



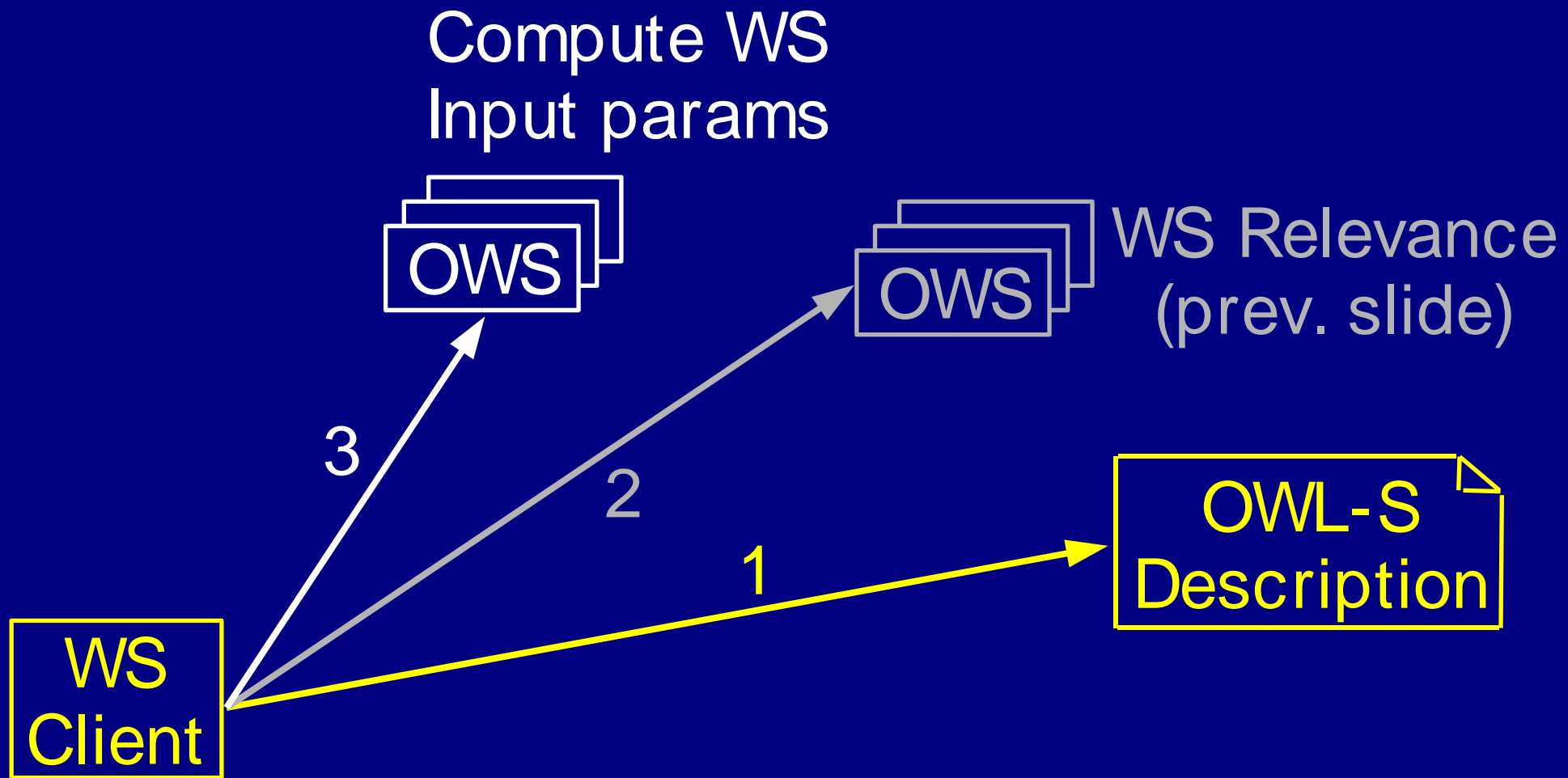
# OWS for assessing WS relevance



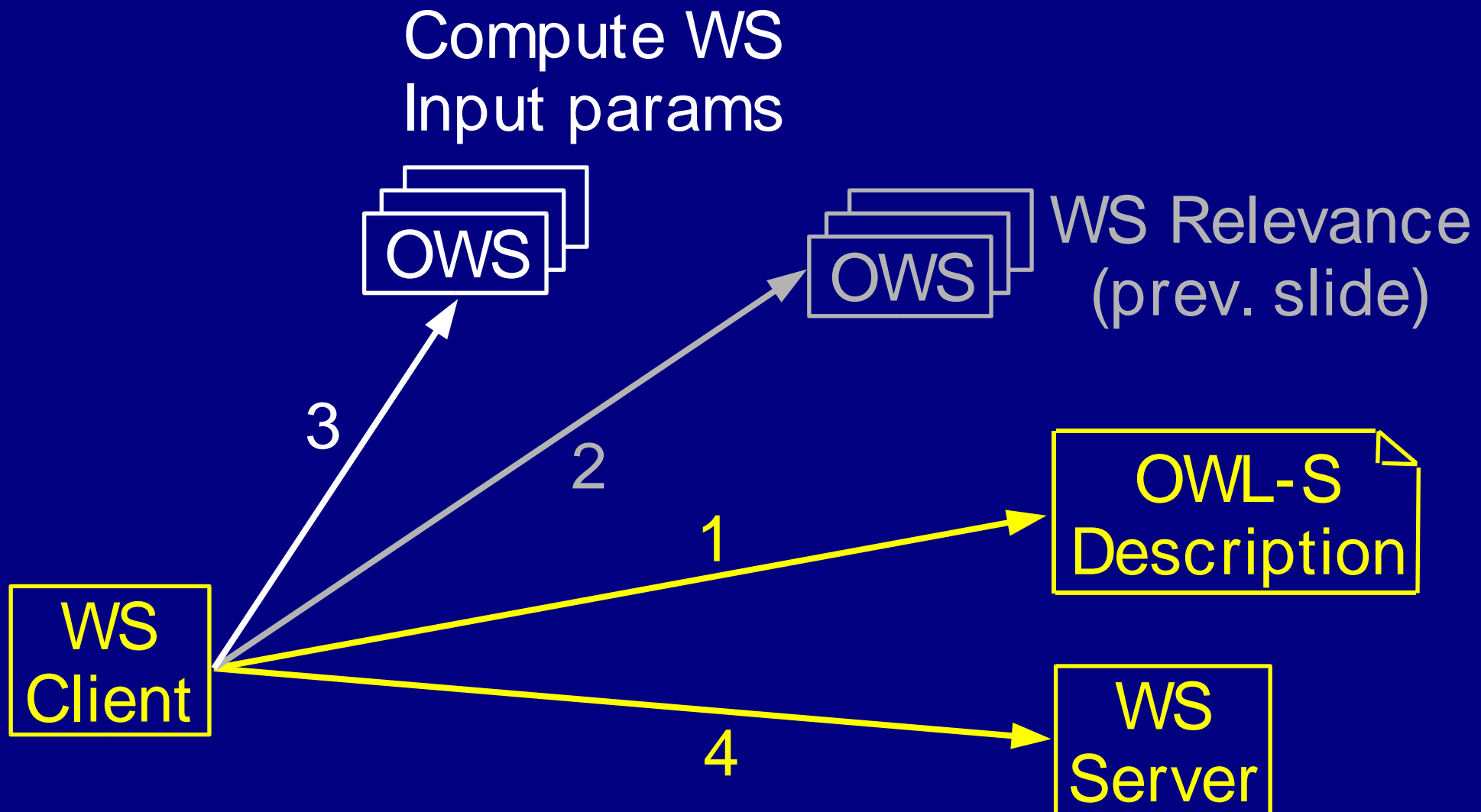
# OWL-S description of OWS may be necessary



# OWS for semantic interoperability



# OWS for semantic interoperability



# OWS for semantic interoperability

Compute WS  
Output params

Compute WS  
Input params



WS Relevance  
(prev. slide)



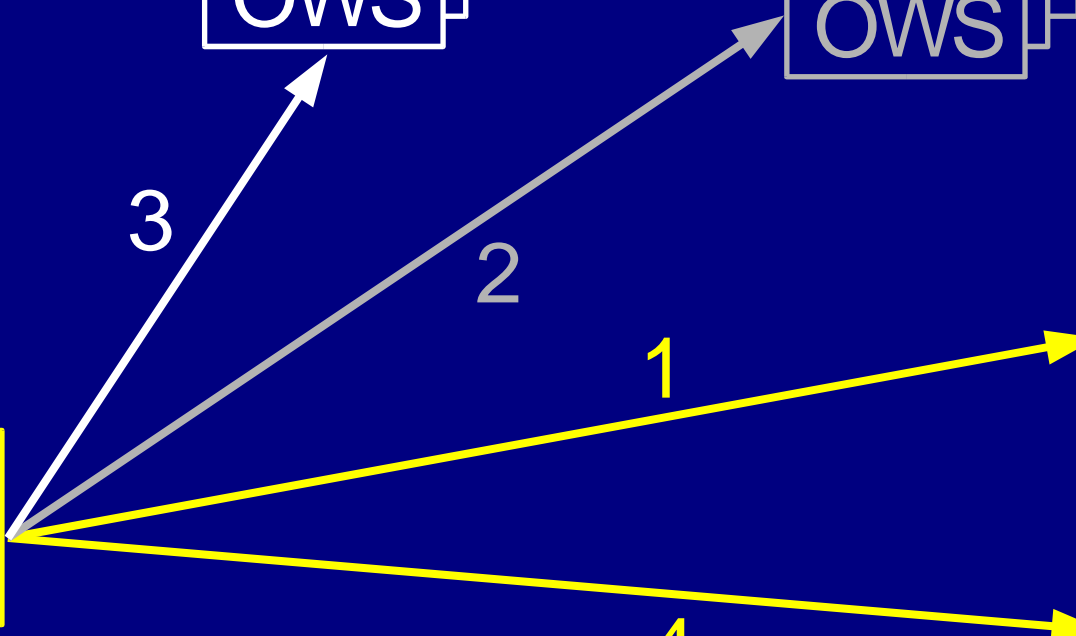
5

3

2

1

4



# Discussion

Relies on existing widespread technologies

- from design to implementation era

Not suited for every business domain

- ontology of the domain (amazon vs. medical app)
- semantically rich and structured domains

Chicken and egg problem

- OWS broker ?

# Conclusion

- Identify some generic ontology-manipulation functions
- Use of OWS for linking isolated resources
  - OWS = semantic duct tape
- OWS also play a role in the SW development
  - Automatic (as much as possible) processing of OWL-S descriptions