Web Services Modelling Ontology

John Domingue

Semantic Web Service Broker

Client

Services

Web Service Modelling Ontology (WSMO)

WSMO Design Principles

- Web Compliance
- Ontology-Based
- Strict Decoupling
- Centrality of Mediation
- Ontological Role Separation
- Description versus Implementation Execution Semantics
- Service versus Web service

WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Provide the formally specified terminology of the information used by all other components

Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Goals

Ontologies
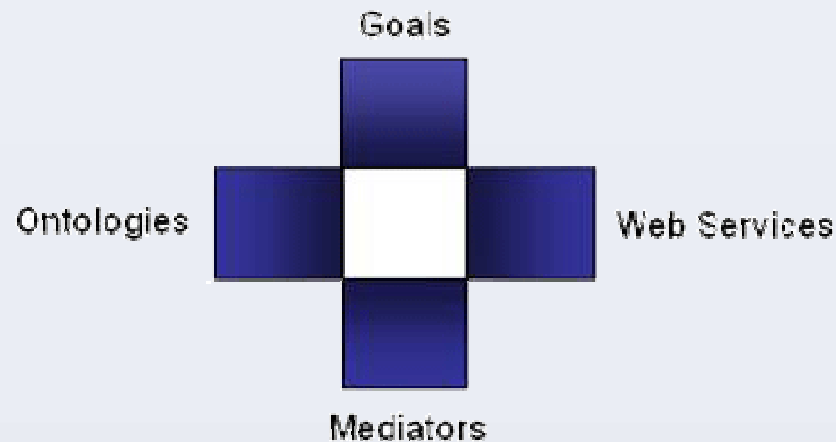
Web Services

Mediators

Connectors between components with mediation facilities for handling heterogeneities

Non-Functional Properties

- Every WSMO element can be described by properties that contain relevant, non-functional aspects.

- Sample information sets are:
  - Dublin Core Metadata Set:
    - For resource management
  - Versioning Information
    - For evolution support
  - Quality of Service Information
    - For availability, stability
  - Other
    - WSMO non functional properties are extensible

Non-Functional Properties List

Dublin Core Metadata

- Contributor
- Coverage
- Creator
- Description
- Format
- Identifier
- Language
- Publisher
- Relation
- Rights
- Source
- Subject
- Title
- Type

Quality of Service

- Accuracy
- NetworkRelatedQoS
- Performance
- Reliability
- Robustness
- Scalability
- Security
- Transactional
- Trust

WSMO Top Level Notions

Objectives that a client wants to
achieve by using Web Services

Goals

Provide the
formally specified
terminology
of the information
used by all other
components

Ontologies

Web Services
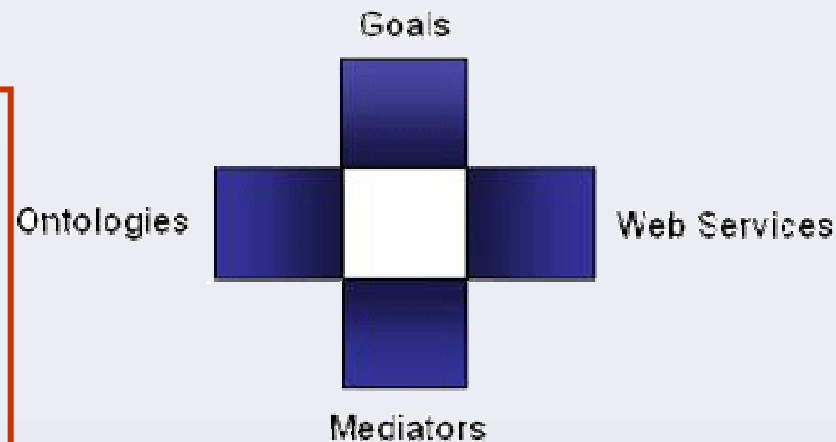
Semantic description of
Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Mediators

Connectors between components
with mediation facilities for handling
heterogeneities

Ontology Description and Usage

- Ontologies are used as the 'data model' throughout WSMO
  - WSMO is defined in terms of itself
  - All data-types used in Web Service interfaces are ontology concepts
  - Discovery, mediation and composition are based on ontology reasoning
- WSMO Ontology Language WSML
  - Conceptual syntax for describing WSMO elements
  - Logical language for axiomatic expressions (WSML Layering)

WSMO Ontology Design

- Modularization
  - import / re-using ontologies
- De-Coupling
  - heterogeneity handled by OO Mediators

Ontology Specification

- **Non functional properties** (see before)
- **Imported Ontologies**
  - importing existing ontologies where no heterogeneities arise
- **Used mediators**
  - OO Mediators (ontology import with terminology mismatch handling)
- **Ontology Elements:**

Concepts	set of concepts that belong to the ontology, incl.
Attributes	set of attributes that belong to a concept
Relations	define interrelations between several concepts
Functions	special type of relation (unary range = return value)
Instances	set of instances that belong to the represented ontology
Axioms	axiomatic expressions in ontology (logical statement)

WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Goals

Provide the formally specified terminology

of the information used by all other components

Ontologies

Web Services

Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Mediators

Connectors between components with mediation facilities for handling heterogeneities

Goals

- Ontological De-coupling of Requester and Provider
- Derived from task / problem solving methods/domain model
- Structure and reuse of requests
  - Search
  - Diagnose
  - Classify
  - Personalise
  - Book a holiday
- Requests may in principle not be satisfiable
- Ontological relationships & mediators used to link goals to Web services

Goal Specification (1/2)

- **Non functional properties**
- **Imported Ontologies**
- **Used mediators**
  - *OO Mediators:* importing ontologies with heterogeneity resolution
  - *GG Mediator:*
    - Goal definition by reusing an already existing goal
    - allows definition of **Goal Ontologies**

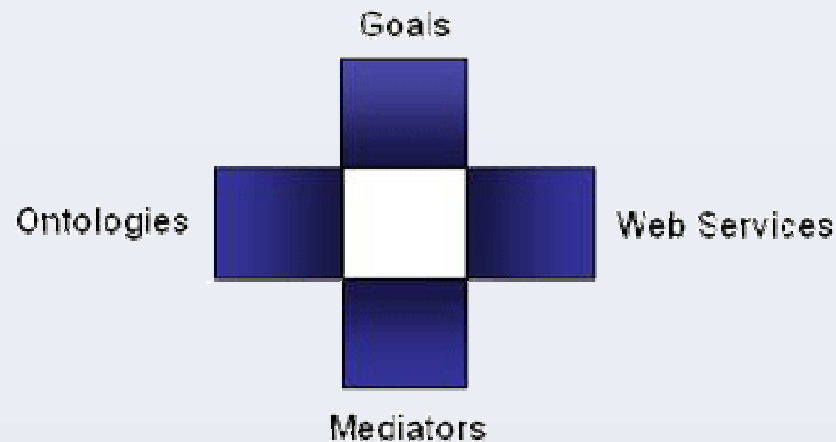# Goal Specification (2/2)

- **Requested Capability**
  - describes service functionality expected to resolve the objective
  - defined as capability description from the requester perspective
- **Requested Interface**
  - describes communication behaviour supported by the requester for consuming a Web Service (Choreography)
  - Restrictions / preferences on orchestrations of acceptable Web Services

WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Provide the formally specified terminology of the information used by all other components

Goals

Ontologies

Web Services

Mediators

Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Connectors between components with mediation facilities for handling heterogeneities

WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

- Advertising of Web Service
- Support for WS Discovery

Non-functional Properties

Capability

DC + QoS + Version + financial

functional description

client-service interaction interface for consuming WS
- External Visible Behavior
- Communication Structure
- 'Grounding'

Web Service Implementation

(not of interest in Web Service Description)

ws

ws

ws

realization of functionality by aggregating other Web Services
- functional decomposition
- WS composition

Choreography --- Service Interfaces --- **Orchestration**

WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

DC + QoS + Version + financial

- Advertising of Web Service
- Support for WS Discovery
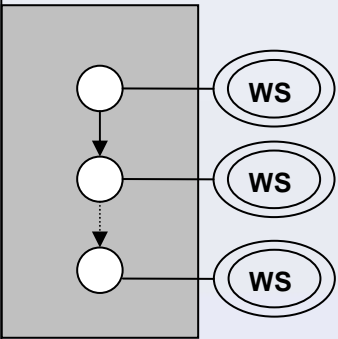
Capability

functional description

client-service interaction interface for consuming WS
- External Visible Behavior
- Communication Structure
- 'Grounding'

Web Service Implementation

(not of interest in Web Service Description)

WS

WS

WS

realization of functionality by aggregating other Web Services
- functional decomposition
- WS composition

Choreography --- Service Interfaces --- **Orchestration**

WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

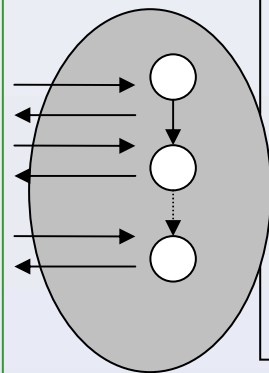DC + QoS + Version + financial

- Advertising of Web Service
- Support for WS Discovery

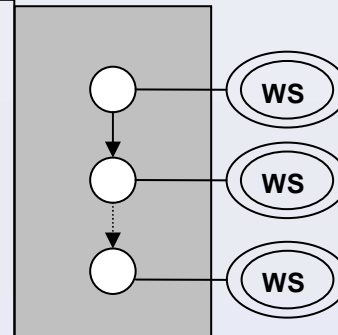**Capability**

functional description

client-service
interaction interface
for consuming WS
- External Visible
  Behavior
- Communication
  Structure
- 'Grounding'

**Web Service
Implementation**

**(not of interest in Web
Service Description)**

ws

ws

ws

realization of
functionality by
aggregating
other Web Services
- functional
  decomposition
- WS composition

Choreography --- Service Interfaces --- **Orchestration**

Capability Specification (1/2)

- **Non functional properties**
- **Imported Ontologies**
- **Used mediators**
  - OO Mediator: importing ontologies with mismatch resolution
  - WG Mediator: link to a Goal wherefore service is not usable a priori

Capability Specification (2/2)

- **Pre-conditions**
  - What a web service expects in order to be able to provide its service
  - Define conditions over the input.
- **Assumptions**
  - Conditions on the state of the world that has to hold before the Web service can be executed
- **Post-conditions**
  - Describes the result of the WS in relation to the input, and conditions on it
- **Effects**
  - Conditions on the state of the world that hold after execution of the Web service (i.e. changes in the state of the world)

WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

- Advertising of Web Service
- Support for WS Discovery

Non-functional Properties

Capability

DC + QoS + Version + financial
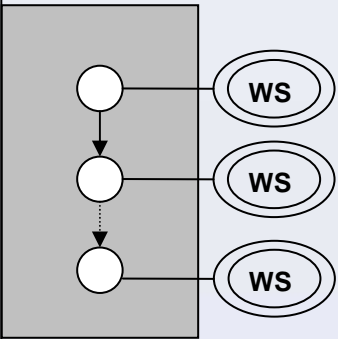
functional description

client-service
interaction interface
for consuming WS
- External Visible
  Behavior
- Communication
  Structure
- 'Grounding'

**Web Service
Implementation**

**(not of interest in Web
Service Description)**

ws

ws

ws

realization of
functionality by
aggregating
other Web Services
- functional
  decomposition
- WS composition

Choreography --- Service Interfaces --- **Orchestration**
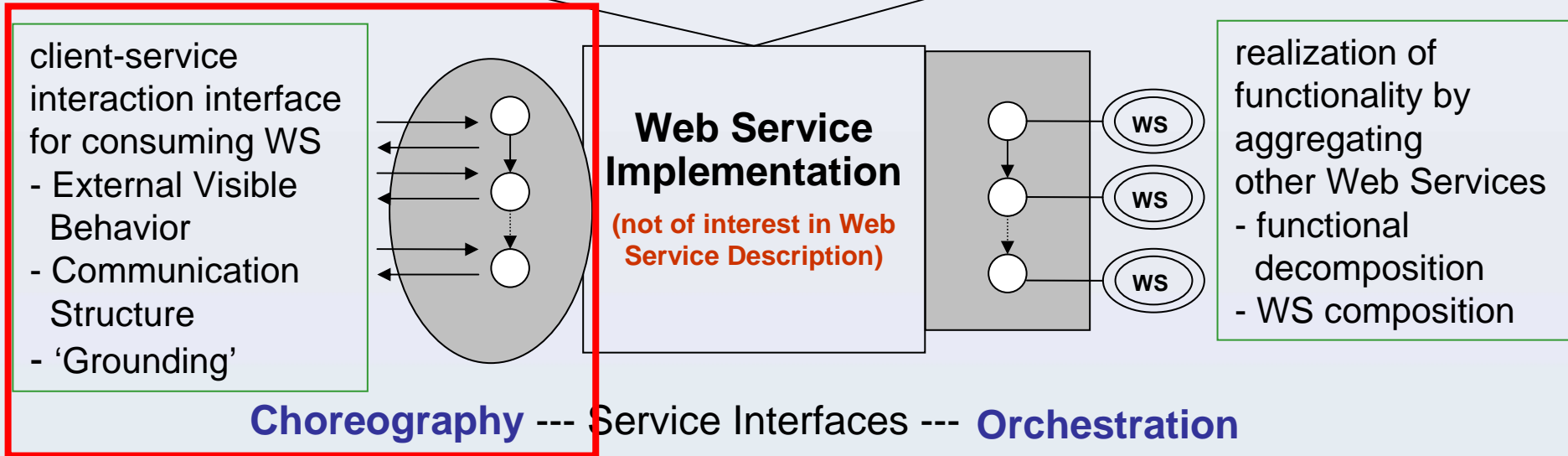
Choreography & Orchestration

VTA example:

When the service is *requested*

When the service *requests*

Date

Time

Flight, Hotel

Error

Confirmation

VTA Service

Date, Time

Hotel

Error

Confirmation

Hotel Service

Date, Time

Flight

Error

Confirmation

Flight Service
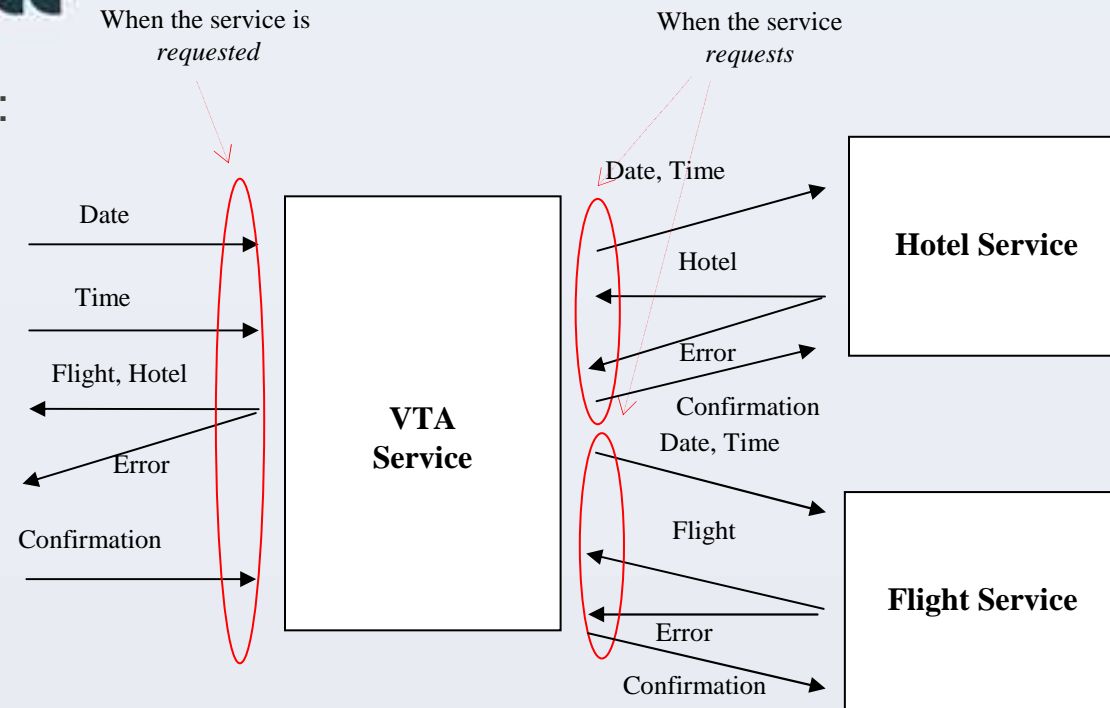
Choreography Aspects (1/2)

- *Interface for consuming Web Service*
  - External Visible Behavior
    - those aspects of the workflow of a Web Service where Interaction is required
    - described by workflow constructs: sequence, split, loop, parallel
  - Communication Structure
    - messages sent and received
    - their order (communicative behavior for service consumption)
    - choreography related errors (e.g. input wrong, message timeout, etc.)

Choreography Aspects (2/2)

- *Interface for consuming Web Service*
  - Grounding
    - concrete communication technology for interaction
  - Formal Model
    - reasoning on Web Service interfaces (service interoperability)
    - allow mediation support on Web Service interfaces

WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

DC + QoS + Version + financial

- Advertising of Web Service
- Support for WS Discovery
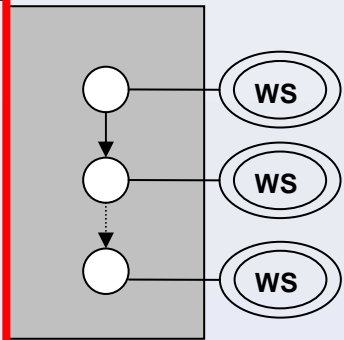
Capability

functional description

client-service interaction interface for consuming WS
- External Visible Behavior
- Communication Structure
- 'Grounding'
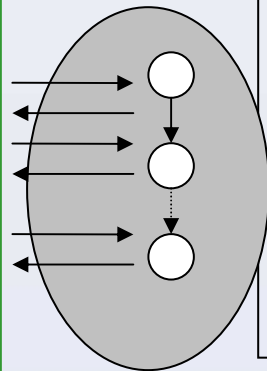
Web Service Implementation
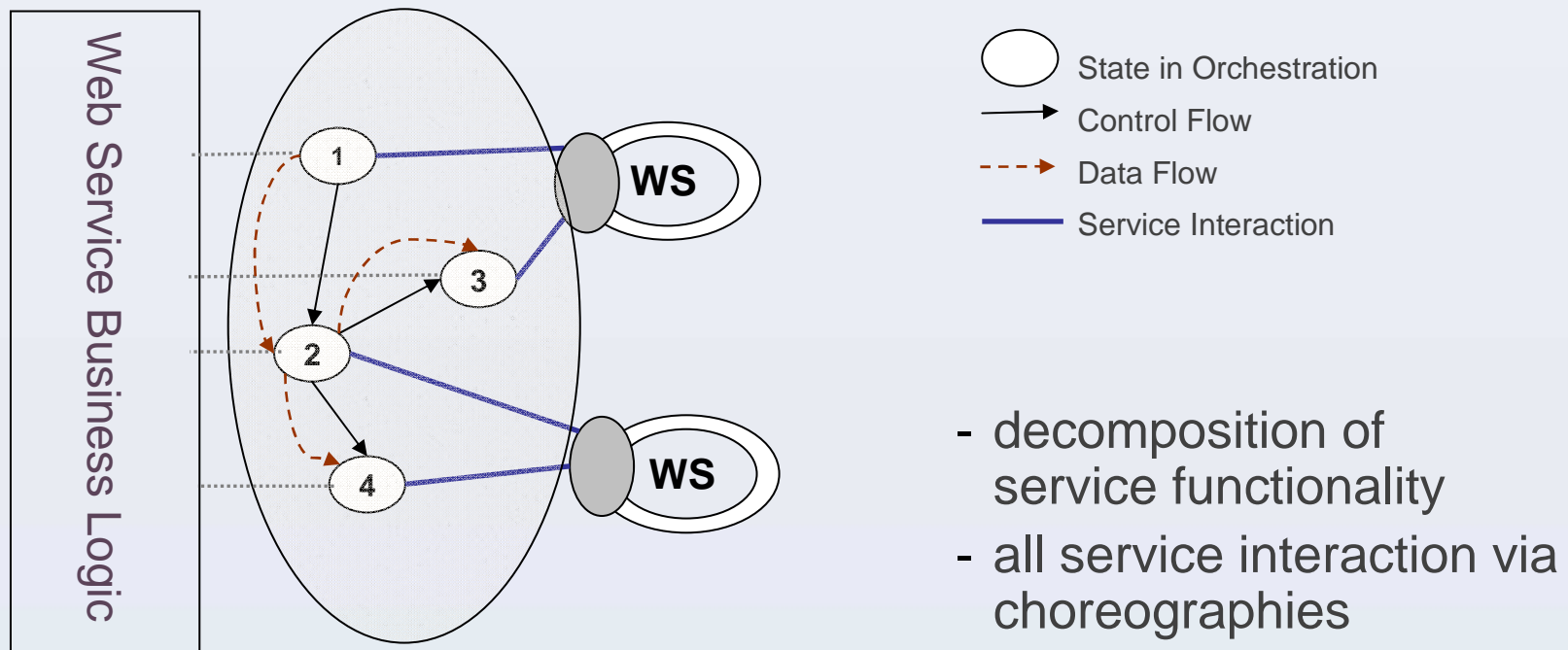
(not of interest in Web Service Description)

realization of functionality by aggregating other Web Services
- functional decomposition
- WS composition

ws

ws

ws

Choreography --- Service Interfaces --- **Orchestration**

Orchestration Aspects

Control Structure for aggregation of other Web Services

Web Service Business Logic

Legend:
- ⬭ State in Orchestration
- → Control Flow
- --→ Data Flow
- — Service Interaction

- decomposition of service functionality
- all service interaction via choreographies
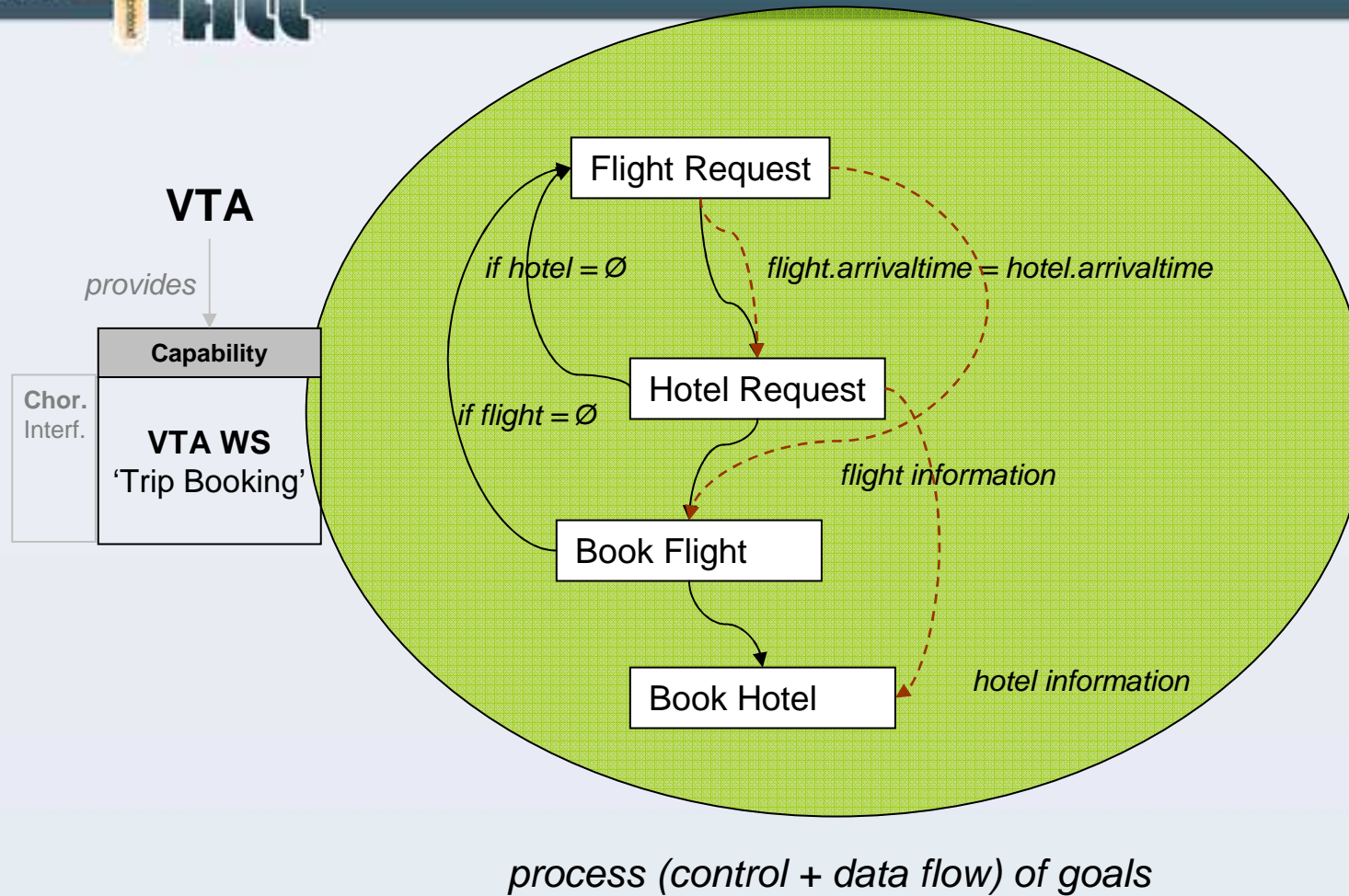
Orchestration Aspects

- Service interfaces are concerned with service consumption and interaction
- Choreography and Orchestration as sub-concepts of Service Interface

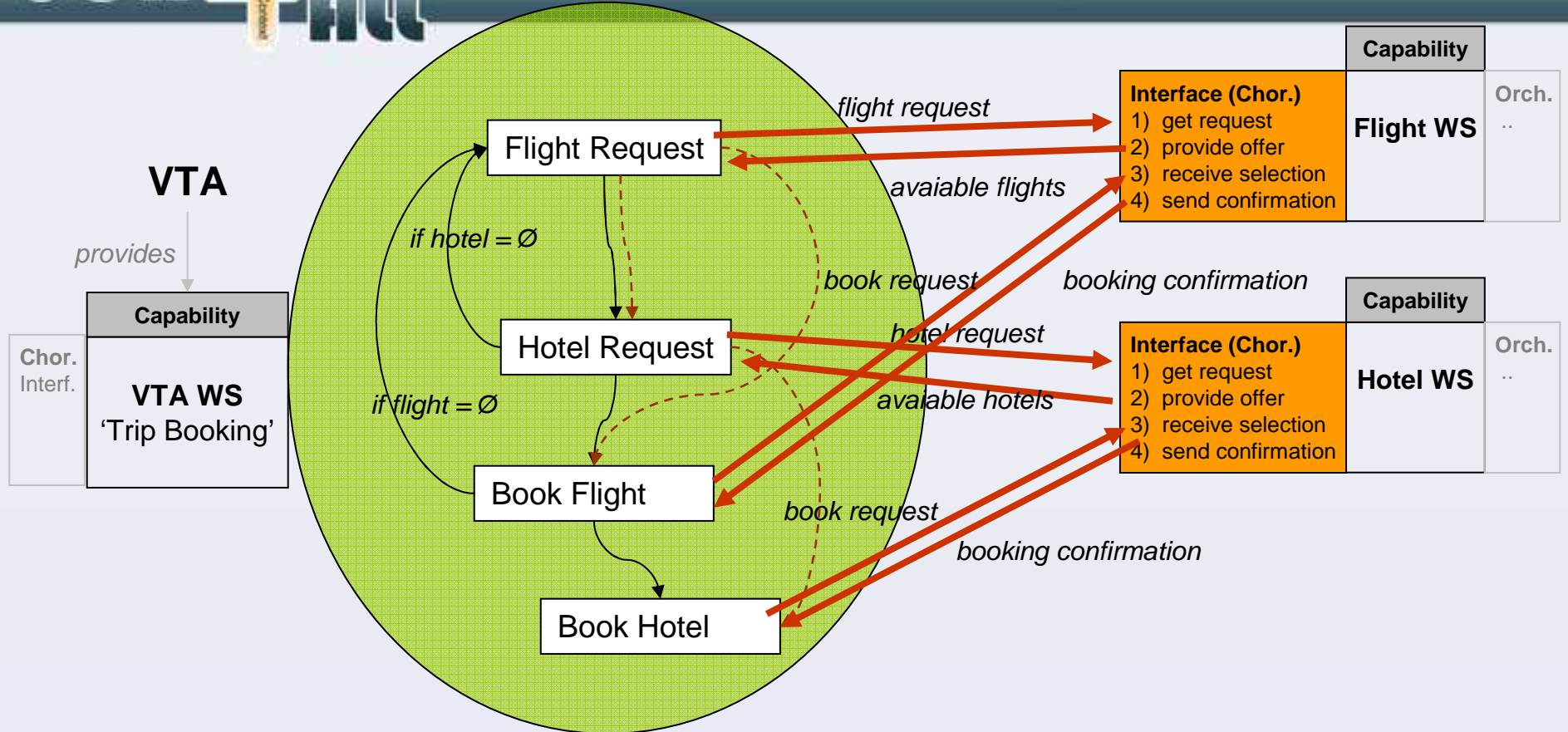# Common requirements for service interface description

- Represent the dynamics of information interchange during service consumption and interaction
- Support ontologies as the underlying data model
- Appropriate communication technology for information interchange
- Sound formal model / semantics of service interface specifications in order to allow operations on them.

Orchestration Definition

VTA

provides

Capability

Chor.
Interf.

VTA WS
'Trip Booking'

Flight Request

if hotel = Ø

flight.arrivaltime = hotel.arrivaltime

Hotel Request

if flight = Ø

flight information

Book Flight

Book Hotel

hotel information

process (control + data flow) of goals

Runtime Orchestration

VTA

provides

Capability

Chor. Interf.

VTA WS
'Trip Booking'

Flight Request

if hotel = Ø

Hotel Request

if flight = Ø

Book Flight

Book Hotel

flight request

avaiable flights

book request

hotel request

avaiable hotels

booking confirmation

book request

booking confirmation

Capability

Interface (Chor.)
1) get request
2) provide offer
3) receive selection
4) send confirmation

Flight WS

Orch.
..

Capability

Interface (Chor.)
1) get request
2) provide offer
3) receive selection
4) send confirmation
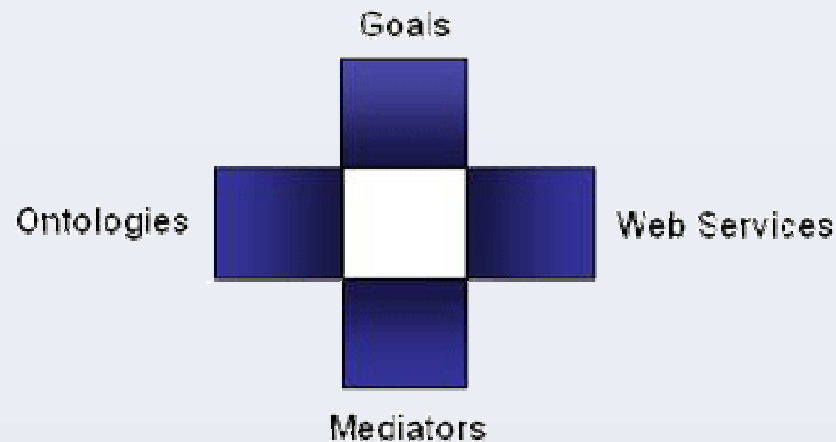
Hotel WS

Orch.
..

process (control + data flow) between "states"
+ communication behavior of orchestrating Web Service

WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Goals

Provide the formally specified terminology

of the information used by all other components

Ontologies

Web Services

Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Mediators

Connectors between components with mediation facilities for handling heterogeneities

Mediation (Wiederhold, 94)

- Mediators as components that resolve mismatches
- Declarative Approach
- Semantic description of resources
- 'Intelligent' mechanisms that resolve mismatches independent of content
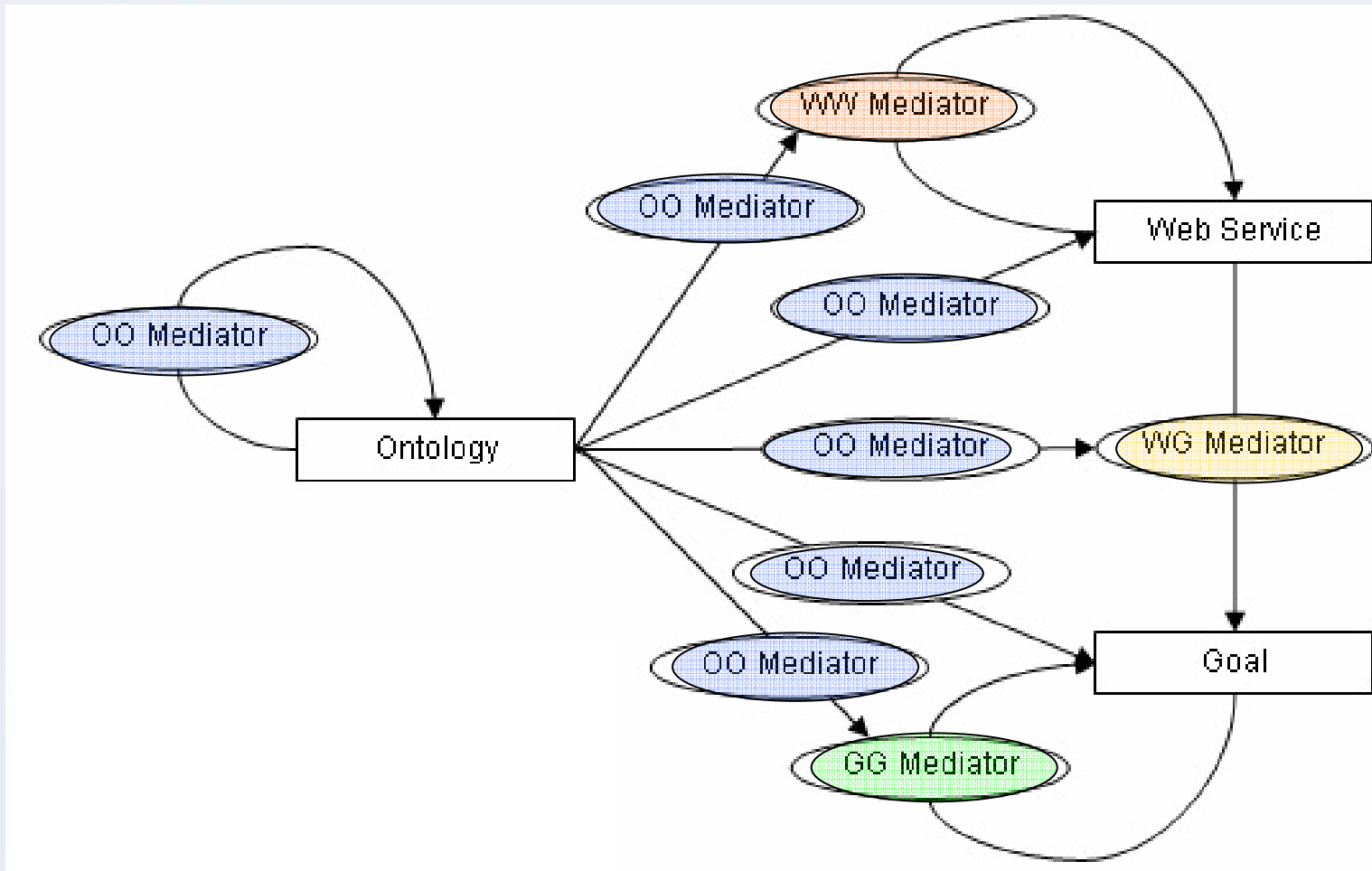- Mediation cannot be fully automated (integration decision)

Mediation

- For 1$ on programming, $5 - $9 on integration
- Mismatches on structural / semantic / conceptual / level
- Assume (nearly) always necessary
- Description of role

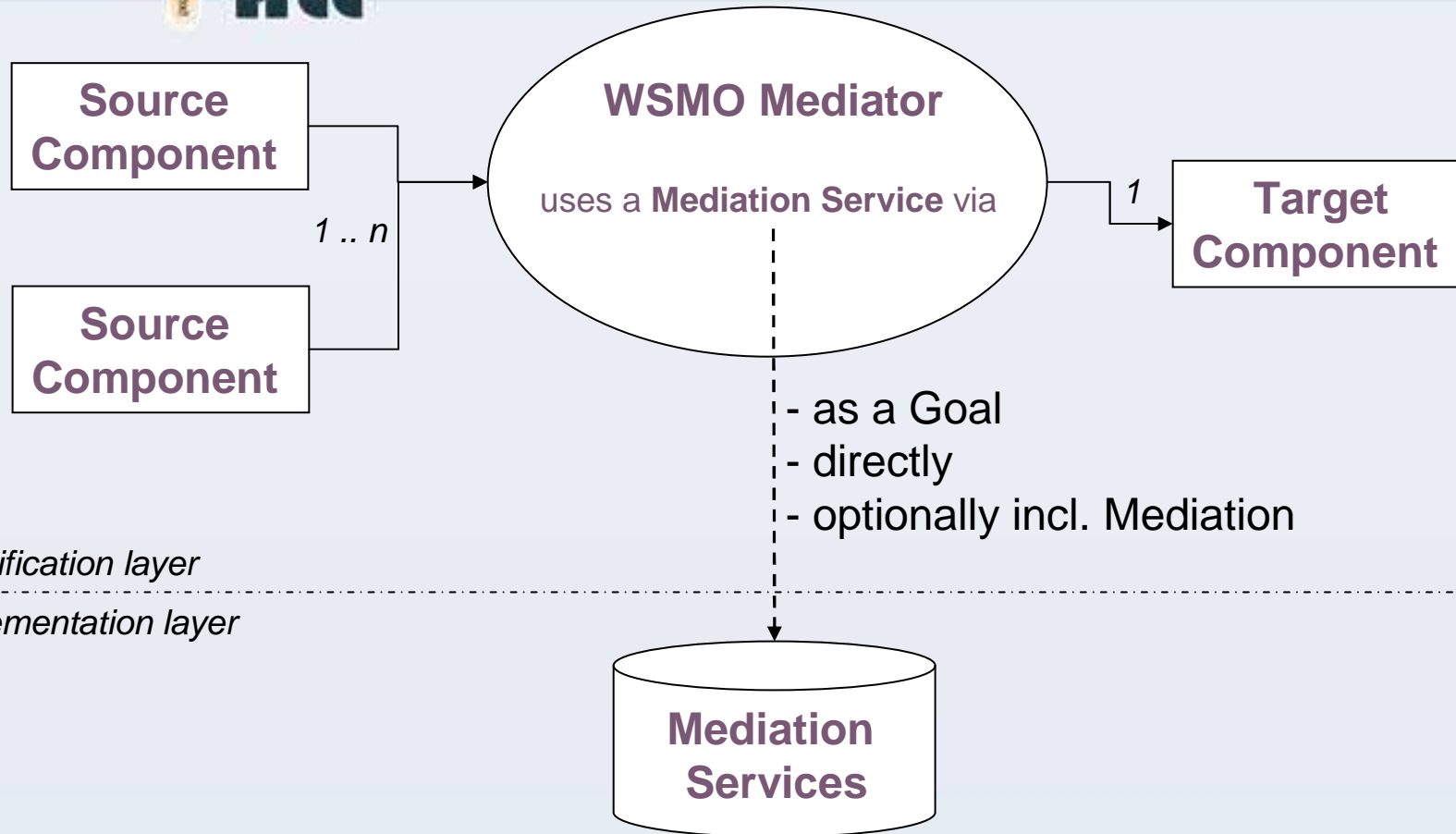# Levels of Mediation within Semantic Web Services

- **Data Level**
  - mediate heterogeneous <u>Data Sources</u>
- **Functional Level**
  - mediate mismatches between <u>Web Service/Goal</u> and <u>Web Service/Goals</u> functionalities
- **Process/Protocol Level**
  - mediate heterogeneous <u>Business Processes</u>/<u>Communication Patterns</u>

- **Layers of Mediators**
  - **Specification Layer** – WSMO Mediators
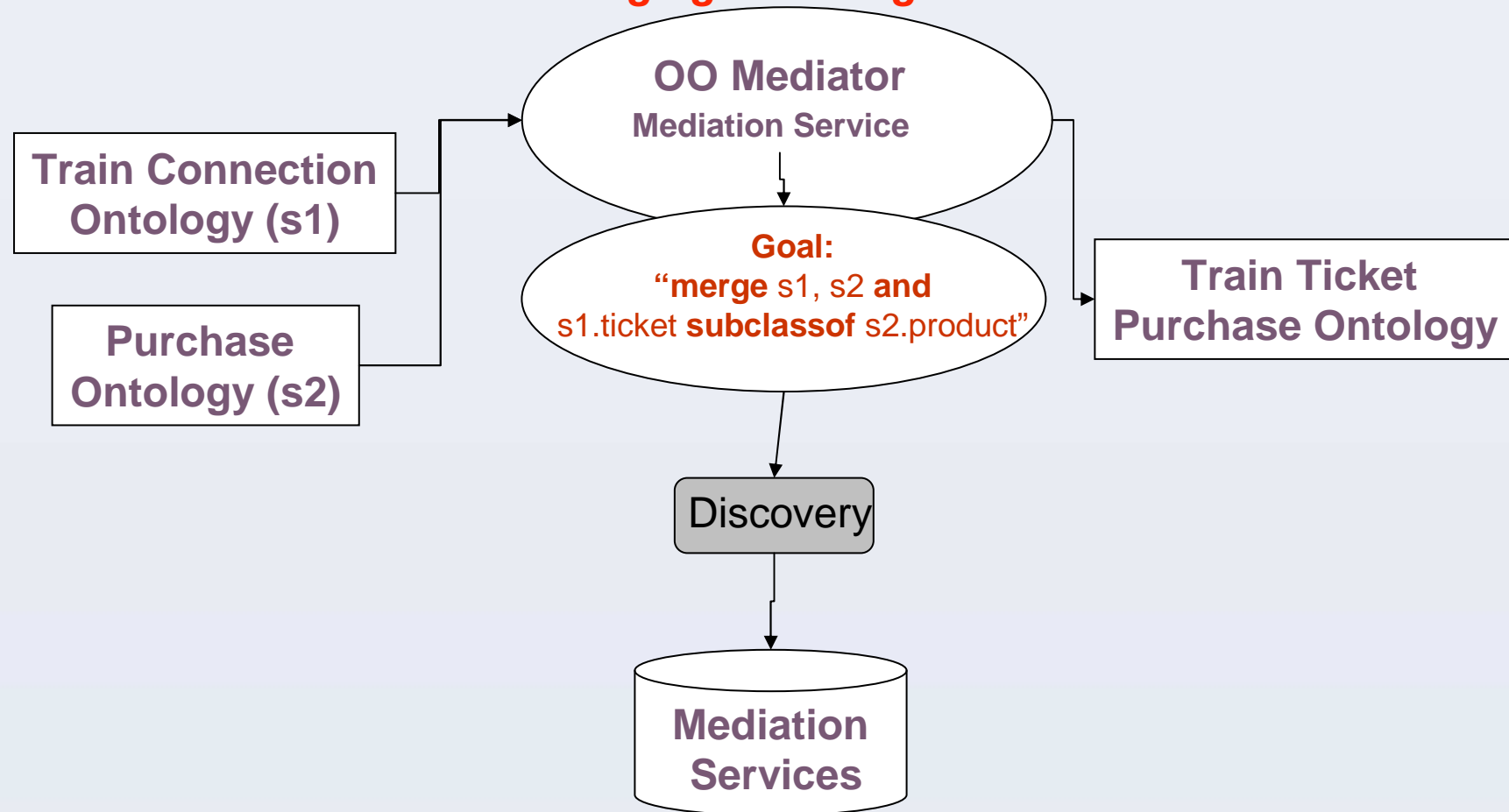 - **Implementation Layer** – Levels of Mediation

WSMO Mediators Overview

Mediator Structure

# OO Mediator - Example

Merging 2 ontologies

OO Mediator
Mediation Service

Train Connection Ontology (s1)

Purchase Ontology (s2)

Goal:
"**merge** s1, s2 **and**
s1.ticket **subclassof** s2.product"

Train Ticket Purchase Ontology

Discovery

Mediation Services

GG Mediators

- Support specification of Goals by re-using existing Goals
- Allow definition of **Goal Ontologies** (collection of pre-defined Goals)
- Terminology mismatches handled by OO Mediators

# GG Mediator Example
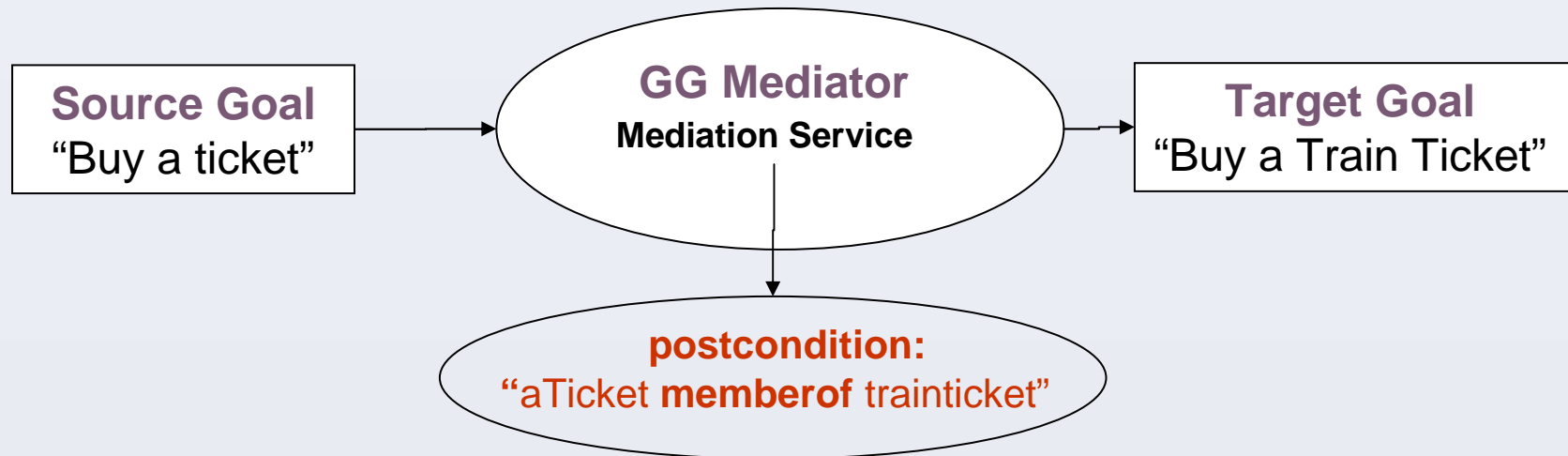
WG Mediators

- Link a Web Service to a Goal and resolve occurring mismatches
- Match Web Service and Goals that do not match a priori
- Handle terminology mismatches between Web Services and Goals
  - broader range of Goals solvable by a Web Service

WW Mediators

- Enable interoperability of heterogeneous Web Services
  - support automated collaboration between Web Services
- **OO Mediators** for terminology import with data level mediation
- Protocol Mediation for establishing valid multi-party collaborations
- Process Mediation for making Business Processes interoperable

WW Mediator Example
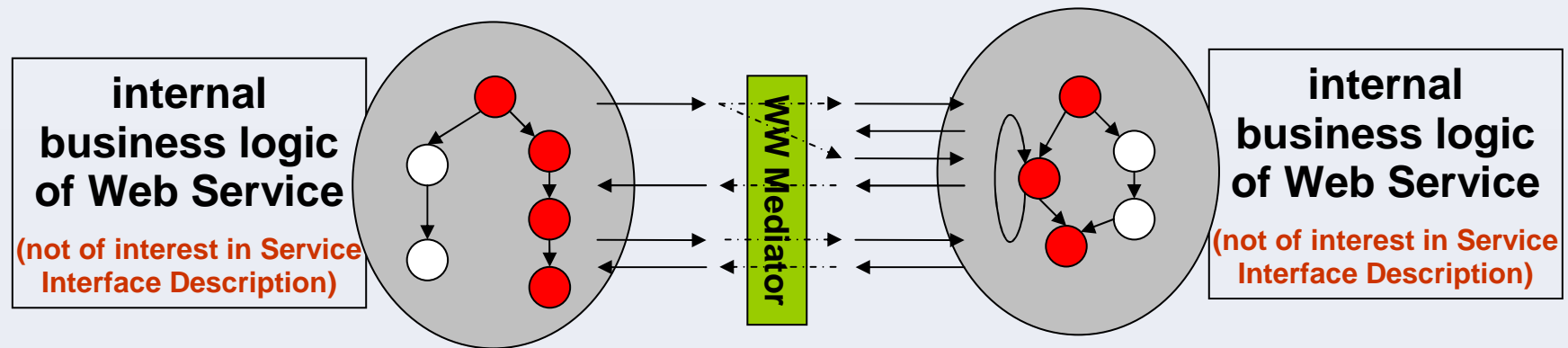
Data Level Mediation (1/2)

- Scope
  - Solving terminological mismatches

- Related Aspects / Techniques:
  - Ontology Integration (Mapping, Merging, Alignment)
  - Data Lifting & Lowering
  - Transformation between Languages / Formalisms

Data Level Mediation (2/2)

- **Terminology Mismatches Classification**
  - Conceptualization Mismatches
    - same domain concepts, but different conceptualization
    - different levels of abstraction
    - different ontological structure
    - => resolution only includes human intervention
  - Explication Mismatches
    - mismatches between:
      - T (Term used), D (definition of concepts), C (real world concept)
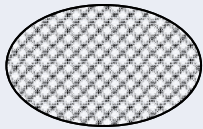   - => automated resolution partially possible

Functional Level Mediation (1/2)

- Scope
  - Solving functional mismatches between goals and/or ws
- Related Aspects/Techniques
  - Discovery
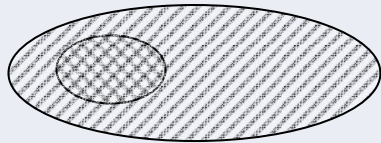 - Semantic Matchmaking
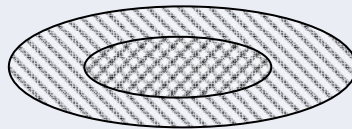- Matchmaking Mismatches

Functional Level Mediation (2/2)

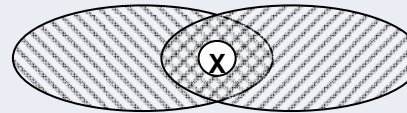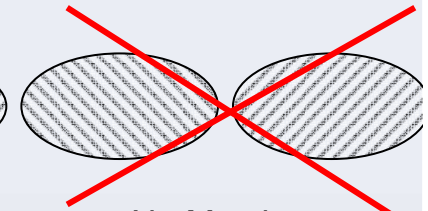◯ = G/WS   ◯ = G/WS

| Exact Match | PlugIn Match | Subsumption Match | Intersection Match | No Match |

Process Level Mediation (1/2)

- Scope
  - Resolves communication mismatches and establish behavior compatibility
- Related Aspects/Techniques
  - Data and control flow composition
- Process Mismatches
  - Signature terminology mismatches (need for data level mediation)
  - Communication/behavior mismatches
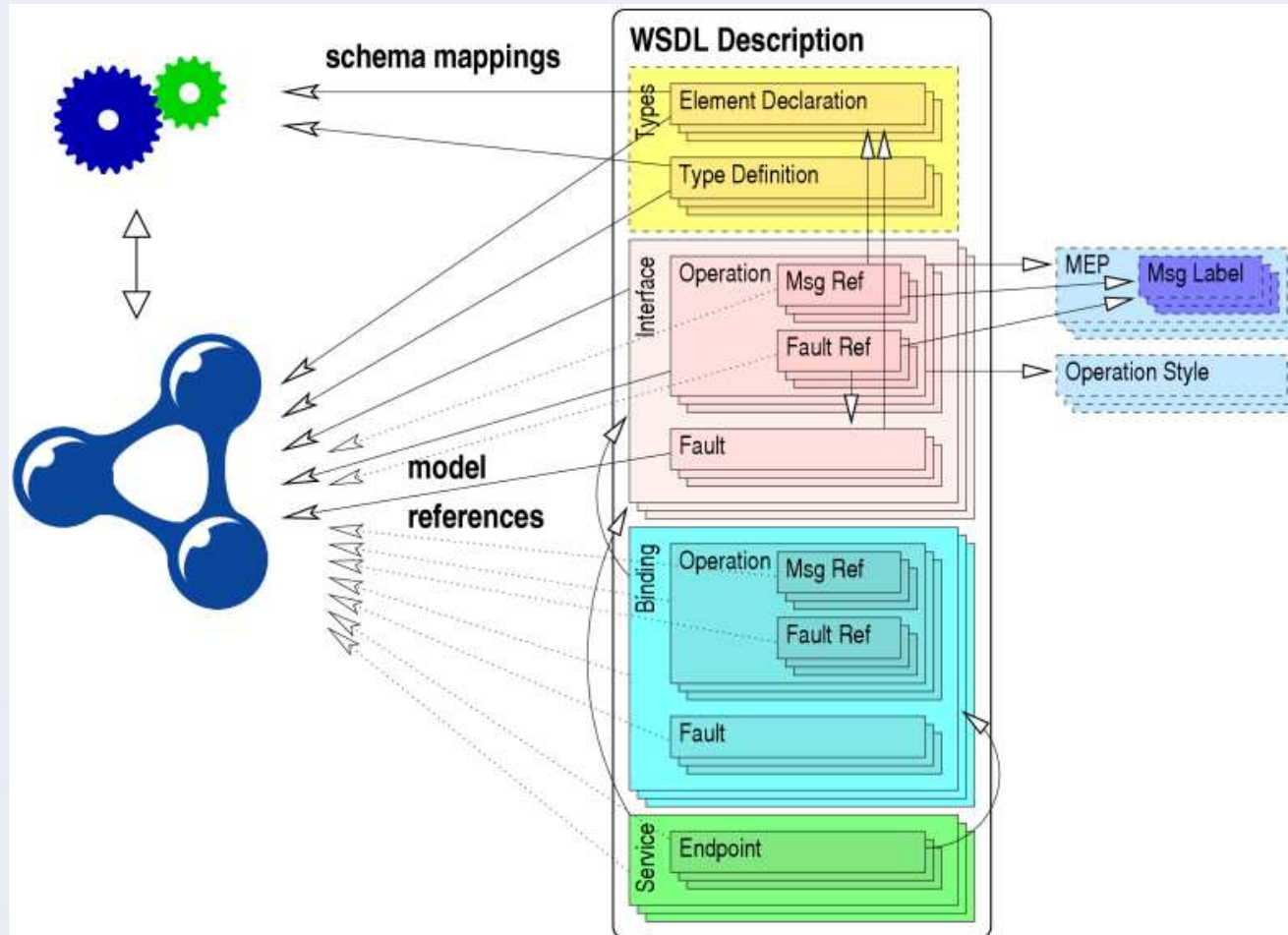
Process Level Mediation (2/2)
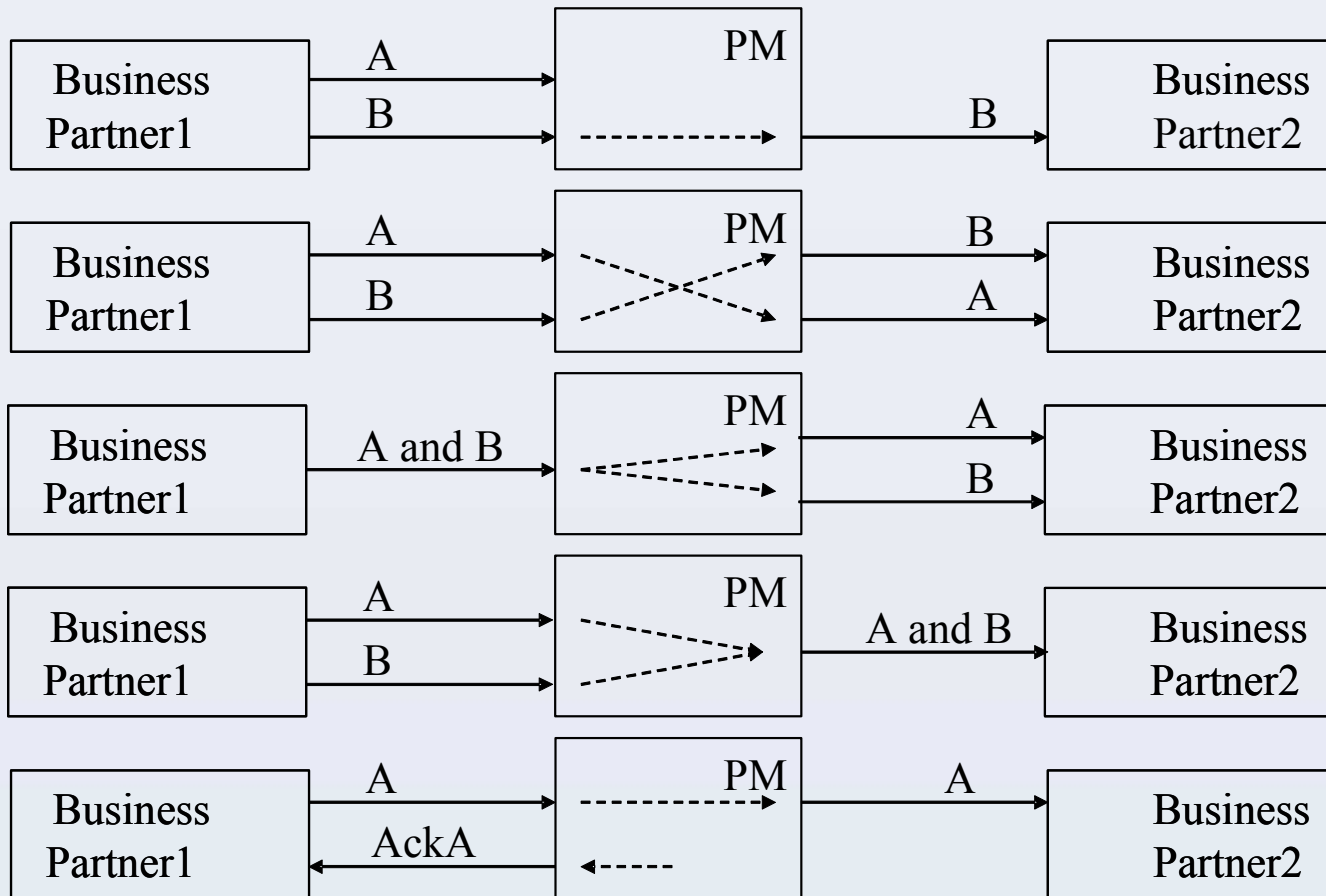
WSMO Work in SOA4All

- ## WSMO-Lite
  - a lightweight ontology which uses RDFS as the description language and defines mechanisms to annotate WSDL descriptions using SAWSDL.

- ## MicroWSMO
  - an annotation mechanism for RESTful services.

SAWSDL in a picture

Summary

- **Semantic Web Services**
  - Potential to cope with Web scale
  - Applies SW to automate application development through reuse of Web services
- **WSMO**
  - Ontology describing Web services
  - Goals, Mediators, Web Services
  - Choreography and Orchestration

Relevant URLs

- **WSMO**
  - http://www.wsmo.org/
- **Conceptual Models of Services**
  - http://cms-wg.sti2.org/

Thanks