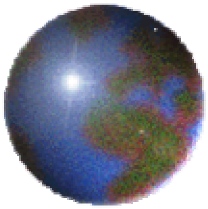


Autonomous Semantic Web Services

Katia Sycara
Carnegie Mellon University
e-mail: katia@cs.cmu.edu
www.cs.cmu.edu/~softagents

Outline

1. **The Vision**
2. What are Web Services
3. Industry Standards
4. Semantic Web Services
5. Semantic Mediation
6. Autonomous Semantic Web Services
7. Conclusion

Today’s Web

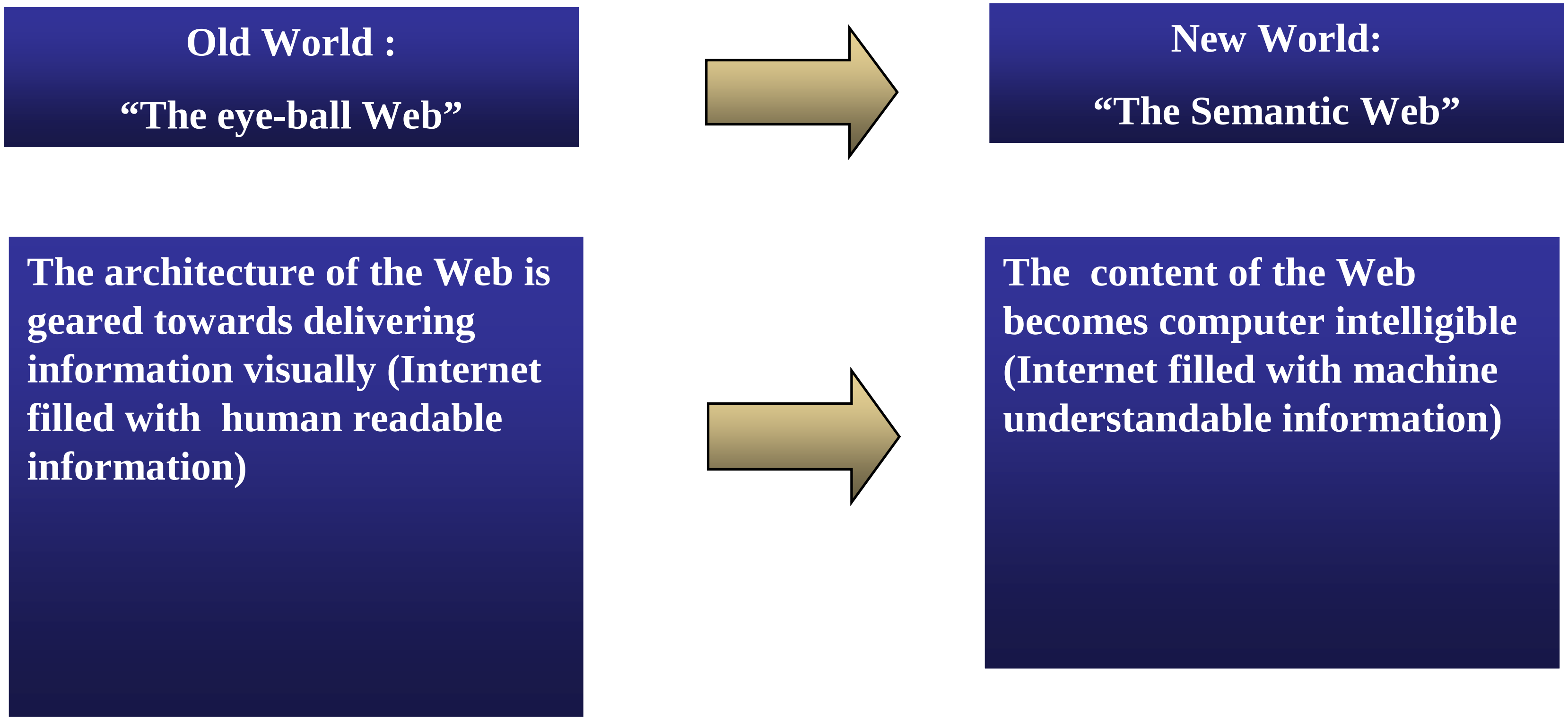
Facts and figures

- 3 billion websites
- 450 m Internet users
- Online B2B market volume 2000: $282 billion

Purpose

- Web designed for application to human interactions
  - Served its purpose well:
  - Information sharing: a distributed content library
  - B2C e-commerce
  - Non-automated B2B interactions

From the Internet to the Semantic Web



Old World :

“The eye-ball Web”

New World:

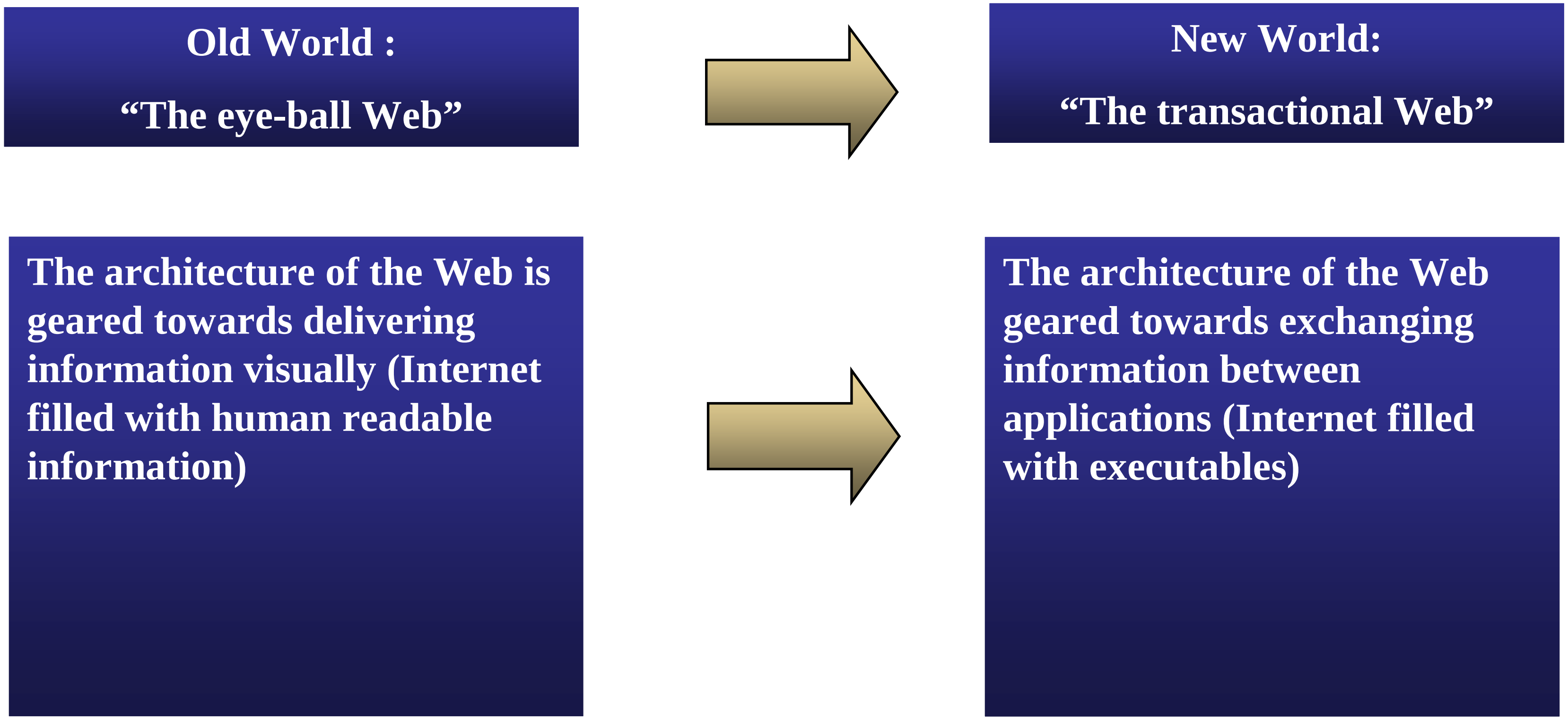
“The Semantic Web”

The architecture of the Web is geared towards delivering information visually (Internet filled with human readable information)

The content of the Web becomes computer intelligible (Internet filled with machine understandable information)

Source: IBM

From the Internet to Web Services



Old World :

“The eye-ball Web”

New World:

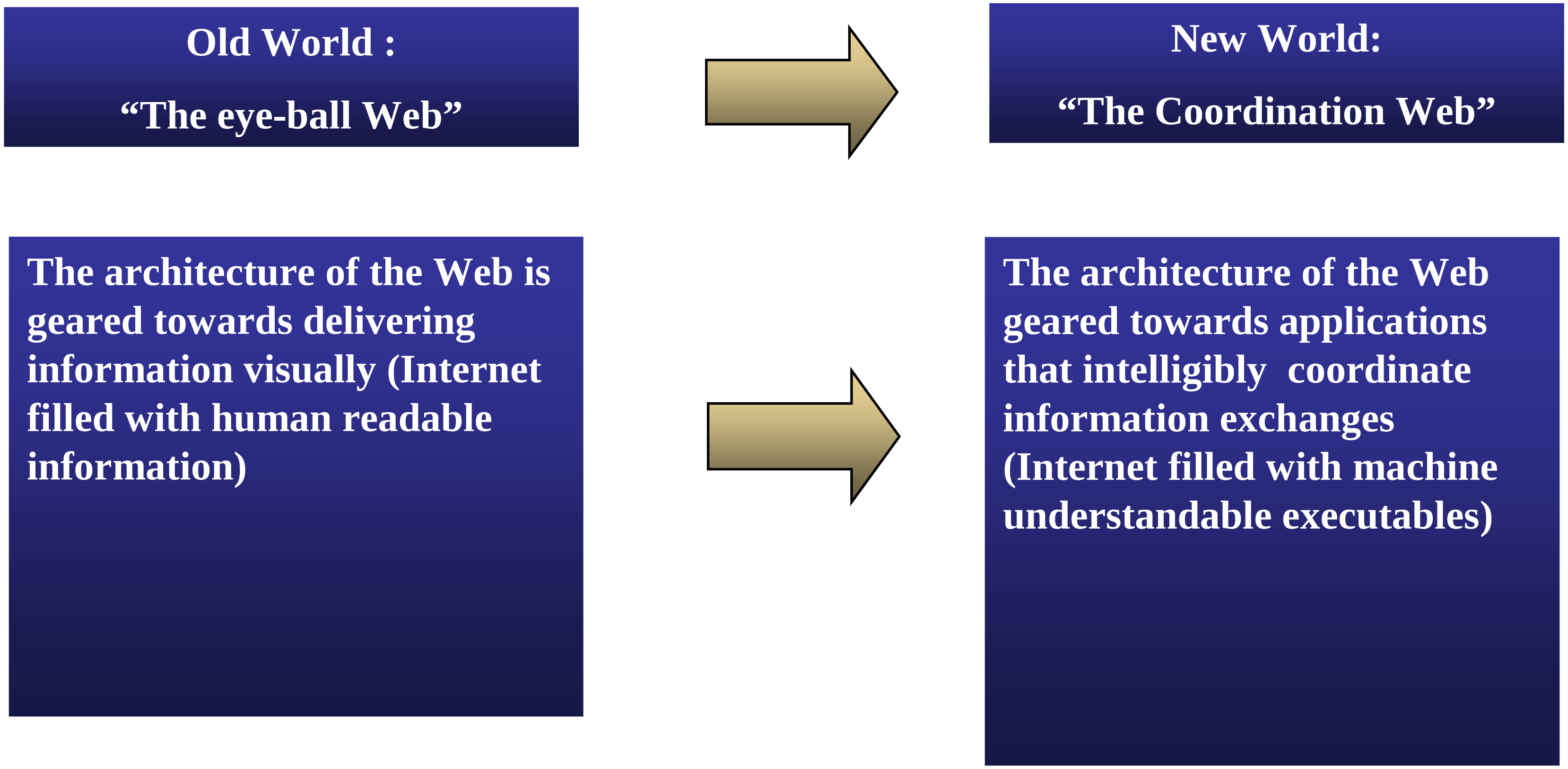
“The transactional Web”

The architecture of the Web is geared towards delivering information visually (Internet filled with human readable information)

The architecture of the Web geared towards exchanging information between applications (Internet filled with executables)

Source: IBM

From the Internet to Semantic Web Services



Old World :

“The eye-ball Web”

New World:

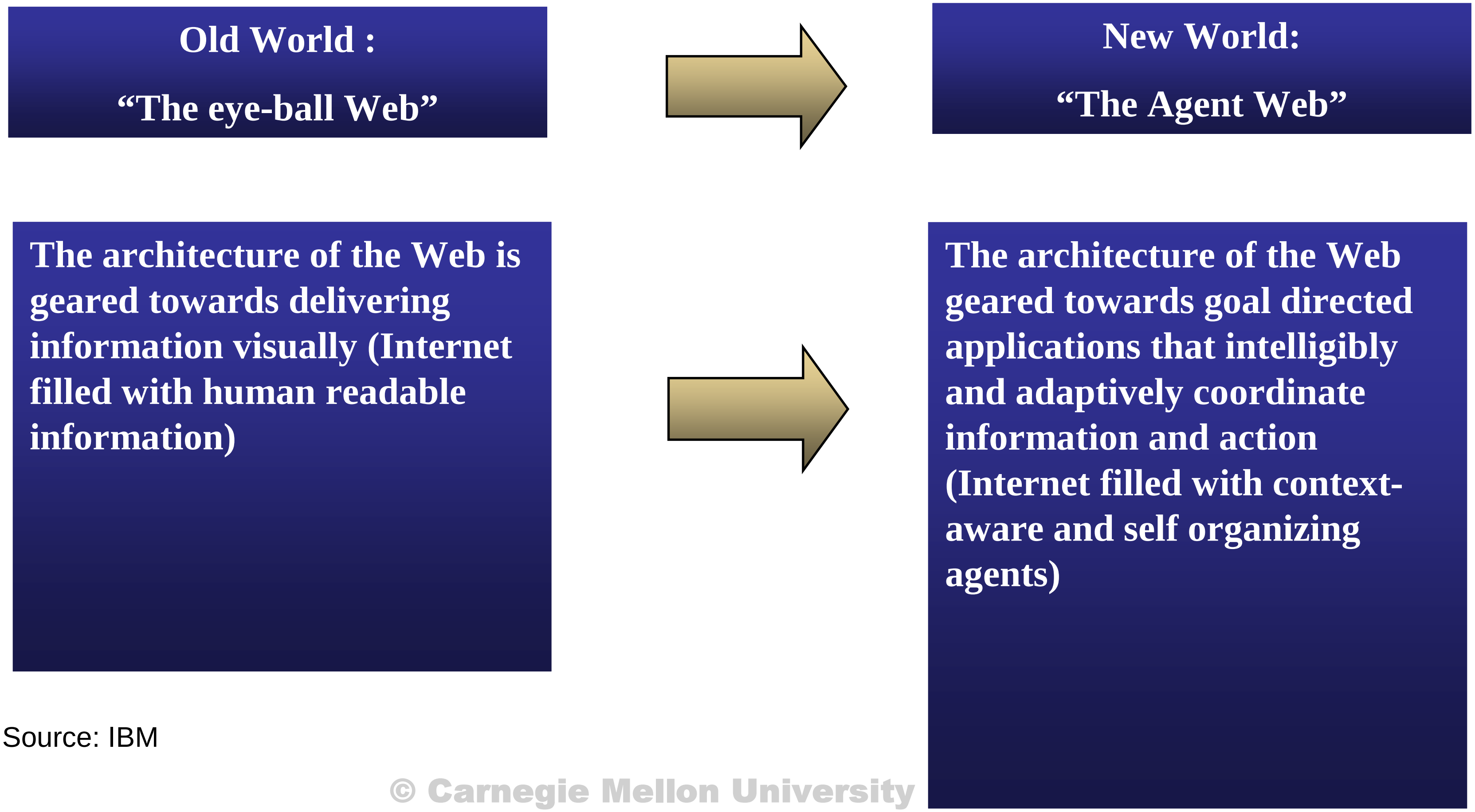
“The Coordination Web”

The architecture of the Web is geared towards delivering information visually (Internet filled with human readable information)

The architecture of the Web geared towards applications that intelligibly coordinate information exchanges (Internet filled with machine understandable executables)

Source: IBM

From the Internet to Autonomous Semantic Web Services



Old World :

“The eye-ball Web”

New World:

“The Agent Web”

The architecture of the Web is geared towards delivering information visually (Internet filled with human readable information)

The architecture of the Web geared towards goal directed applications that intelligibly and adaptively coordinate information and action (Internet filled with context-aware and self organizing agents)

Source: IBM

Outline

1. The Vision
2. **What are Web Services**
3. Industry Standards
4. Semantic Web Services
5. Semantic Mediation
6. Autonomous Semantic Web Services
7. Conclusion

What are Web Services

The Next Killer App

"Web services are expected to revolutionize our life in much the same way as the Internet has during the past decade or so." (Gartner)

"Just as the Web revolutionized how users talk to applications, XML transforms how applications talk to each other." (Bill Gates)

"By 2004, 40% of financial services transactions and 35% of online government services will be web service-based." (Gartner)

"Web Services will be bigger than Java or XML" (Rod Smith, VP of Emerging Technology, IBM)

What are Web Services?

Many Definitions Exist…

Web Services are self-describing components that can discover and engage other web services or applications to complete complex tasks over the Internet. (Sun Microsystems, Inc.)

It is software designed to be used by other software via Internet protocols and formats. (Forrester)

Web Services are loosely coupled software components delivered over the Internet via standards-based technologies like XML, and SOAP. (Gartner)

“Self-describing, self-contained, modular unit of application logic that provides some business functionality to other applications through an Internet connection…” (UDDI.org)

Web Services are Internet-based, modular applications that perform a specific business task and conform to a particular technical format. (IBM)

A web service is application logic that is programmatically available, exposed using the Internet. (Microsoft)

Web Services as a Software Architecture

"Web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. …

Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service."

IBM web service tutorial

Web Services as a programming technology

- The web is organized around URIs, HTML, and HTTP.
  - URIs provide defined IDs to refer to elements on the web
  - HTML provides a standardized way to describe document structures
    - allowing browsers to render information for the human reader
  - HTTP defines a protocol to retrieve information from the web.
- Web services require a similar infrastructure:
  - XML provides a meta language for defining documents
  - Standards required for communication, interface/signature description, protocol description and discovery.
    - e.g. UDDI, WSDL, and SOAP

The Impact of Web Services?

- Improvement of operations
- Agile business relationships
- Reduced cost and increased flexibility
- Shorter time-to-market for new products and services
- Leverage existing infrastructure

Web Services will remedy many expensive and painful problems of today’s business uses of IT:

- Connecting business systems inside a firm is a nightmare
- Inter-enterprise process orchestration is impossible
- Inflexible systems impede business adjustments
- Fragmented personal data frustrates users

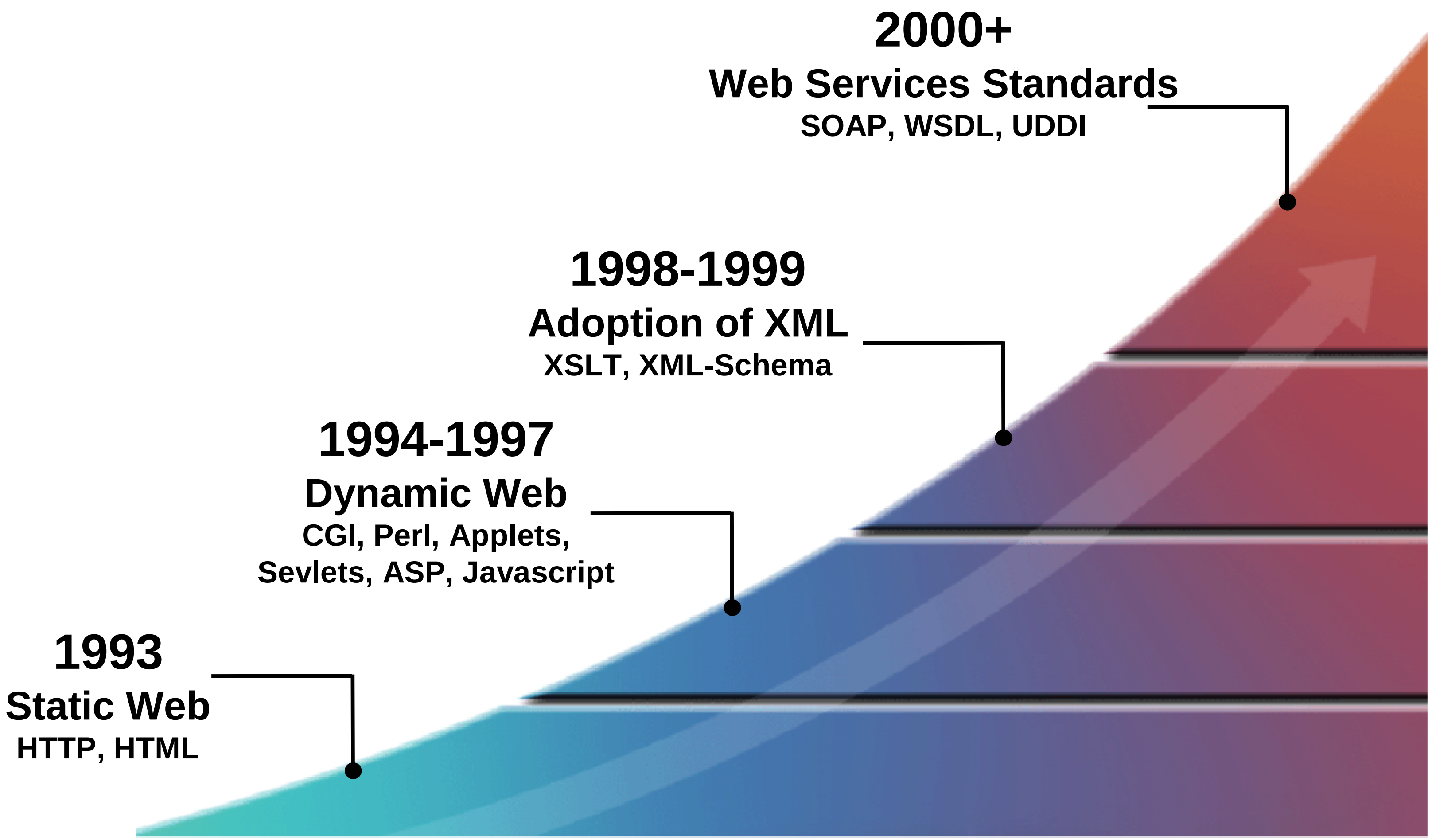
Key characteristics

- A Web Service is **accessible over the Web**.
- Web Services communicate using **platform-independent and language-neutral Web protocols**
- A Web Service provides a **specific functionality that can be used by other programs**
- A Web Service is **registered and can be located** through a Web Service Registry

So what is new about Web Services?

Component-Based Model	Web Services Model
Tightly coupled **software applications (high dependencies between systems)**	Loosely coupled **software applications (low dependencies between applications)**
Mainly designed for processes within the enterprise	**Mainly designed for** processes across enterprises
Uses different protocols and technologies **(e.g., Microsoft DCOM, CORBA)**	**Uses** common protocols and technologies **(e.g., XML, SOAP, WSDL, HTTP)**

Evolution of WWW Technologies & Tools



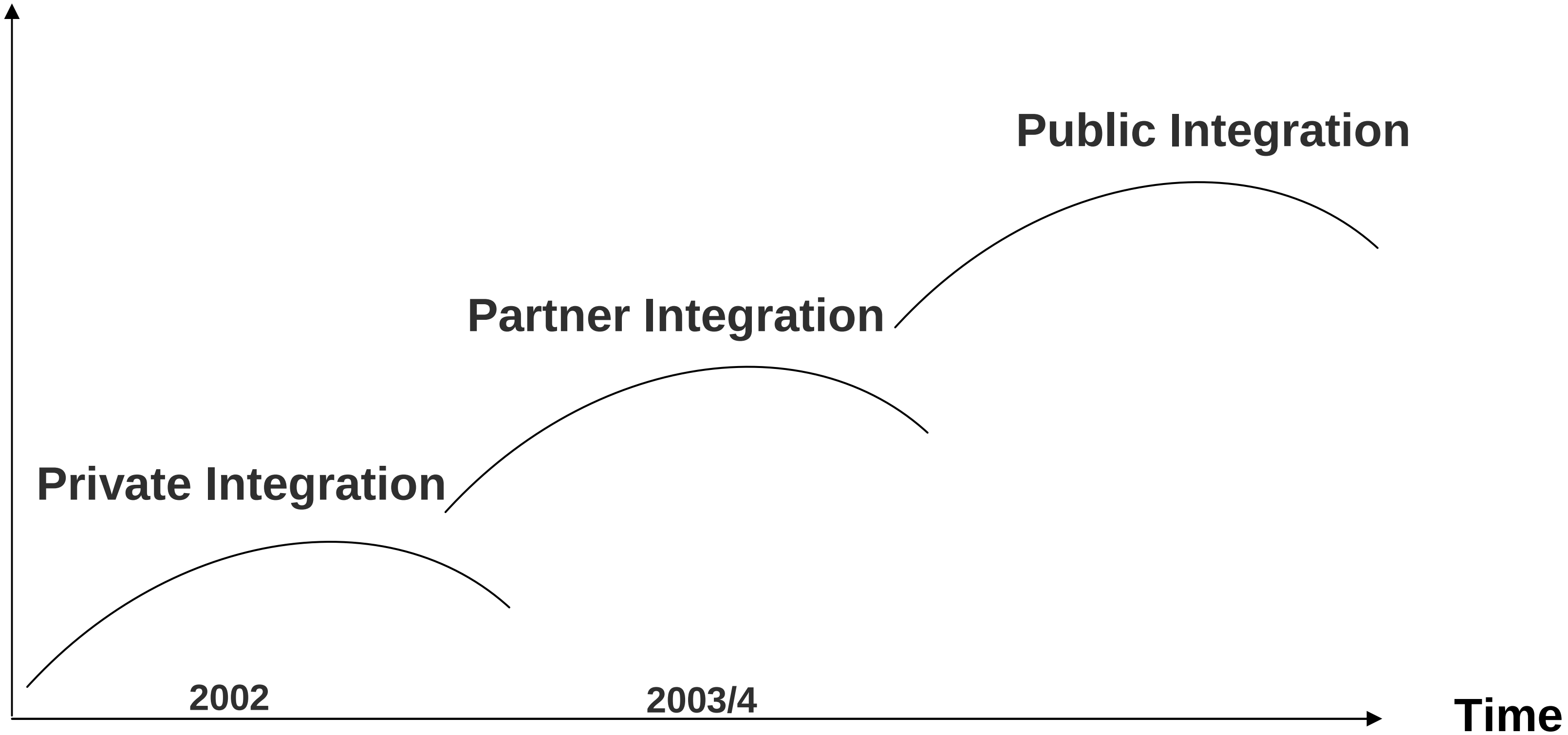
2000+
Web Services Standards
SOAP, WSDL, UDDI

1998-1999
Adoption of XML
XSLT, XML-Schema

1994-1997
Dynamic Web
CGI, Perl, Applets,
Sevlets, ASP, Javascript

1993
Static Web
HTTP, HTML

The Evolution of Web Services?



Private Integration

Partner Integration

Public Integration

2002

2003/4

Time

Defi-nition

Integration between internal applications

Integration between remote applications at separate firms or business units

Integration without a prior relationship

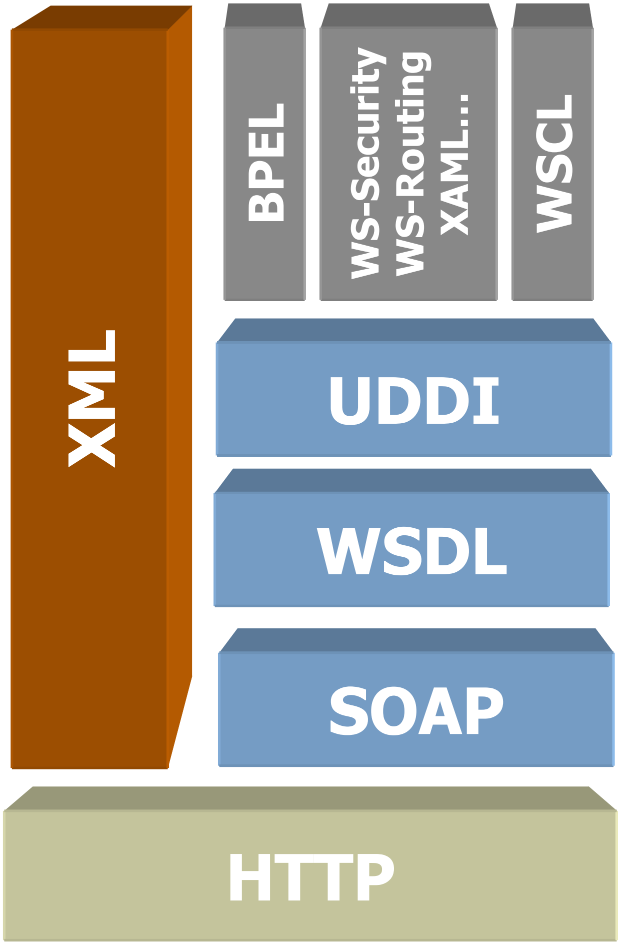
Future Evolution of Web Services

- Semantic Web Services: services whose description is in a machine understandable language with formal semantics
- Agent-based or Autonomous Semantic Web Services:
  - goal-directed
  - autonomous choice of partners
  - based on own current internal goals, internal attitudes and their projection of their future needs.
  - semantics support ASWS

Outline

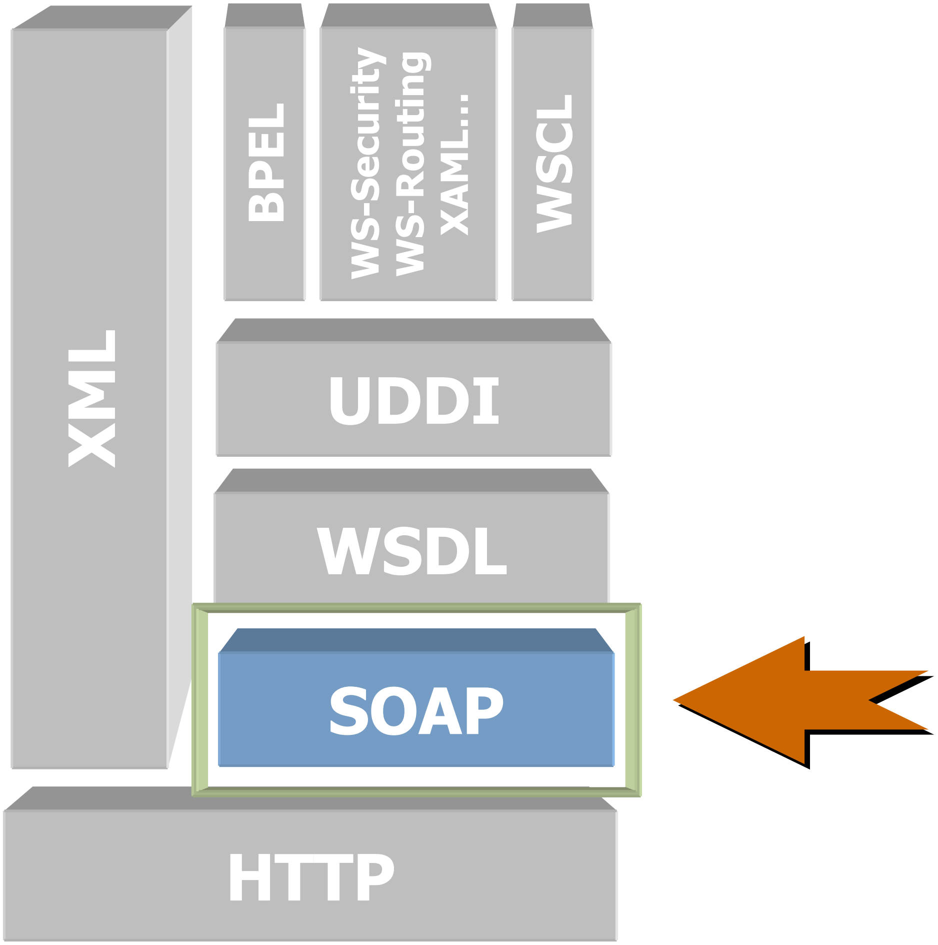
1. The Vision
2. What are Web Services
3. **Industry Standards**
4. Semantic Web Services
5. Semantic Mediation
6. Autonomous Semantic Web Services
7. Conclusion

Overview of Web Services Standards



- Data and Control Flow descriptions of Web Services; Security and Management
- A mechanism for registering and looking up web services
- Programmatic way of describing the Web Service Interface
- Web Services Communication protocol

Overview of Web Services Standards



Communicating between services using the **Simple Object Access Protocol - SOAP**

SOAP (Simple Object Access Protocol)

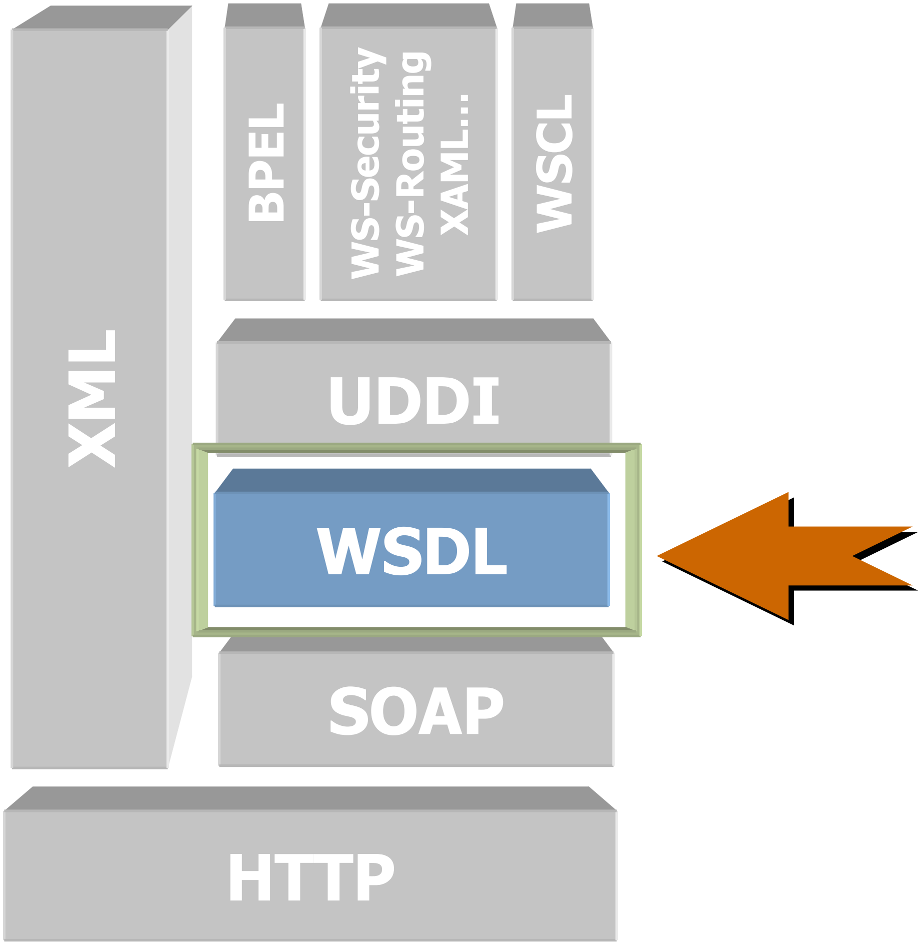
- XML based web services communication protocol
  - Provides message support for many Web Services standards such as WSDL, UDDI, and Microsoft’s .NET architecture
  - Uses GET/POST across http, thus providing a platform & language independent means of communicating

- SOAP documents contain:
  - Header
    - optional information about the transaction
  - Body
    - contains payload (e.g. a request or response)
    - may instead contain error/fault information if requests fail

Limitations of SOAP

- Unbounded message format
    - Requires a-priori agreement between Web Services on message format and protocol
    - Provided by higher level standards (e.g. WSDL)
- Has no communicative speech acts
    - No way to determine
        - The intention of the message sender
        - What the message is trying to achieve
- Agent communication languages such as FIPA KQML define speech acts, such as:
        - Basic query performatives (ask-one, ask-all,...)
        - Multi-response query performatives (stream-in,..)
        - Response (reply, sorry,…)
        - …

Overview of Web Services Standards

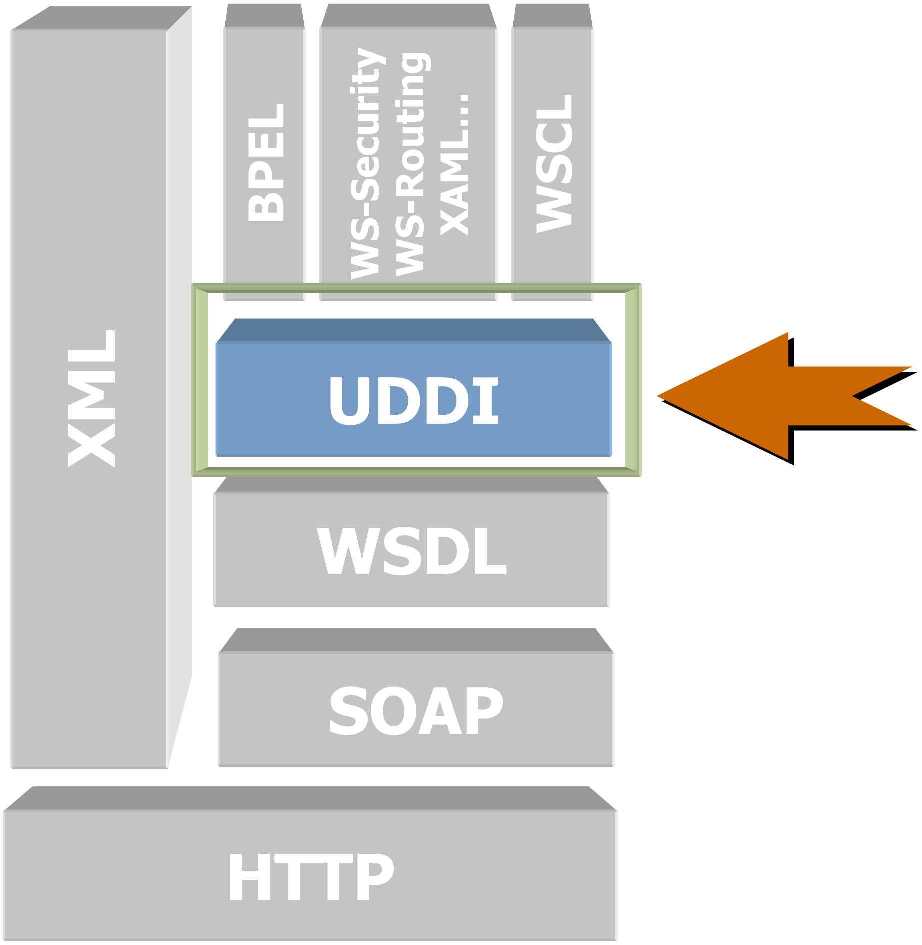


Defining transactions with the **Web Services Description Language - WSDL**

WSDL (Web Services Description Language)

- Structured mechanism to describe:
  - Abstract operations that a Web Service can perform
  - Format of messages it can process
  - Protocols it can support
  - Physical bindings to:
    - communication languages, e.g. SOAP or HTTP messages
    - Location of services, i.e. URI and port numbers
- XML based
- Supports simple transactions (operations)
  - E.g. request-response, solicit-response, etc.

Overview of Web Services Standards



Searching for Services using **Universal Discovery, Description & Integration - UDDI**

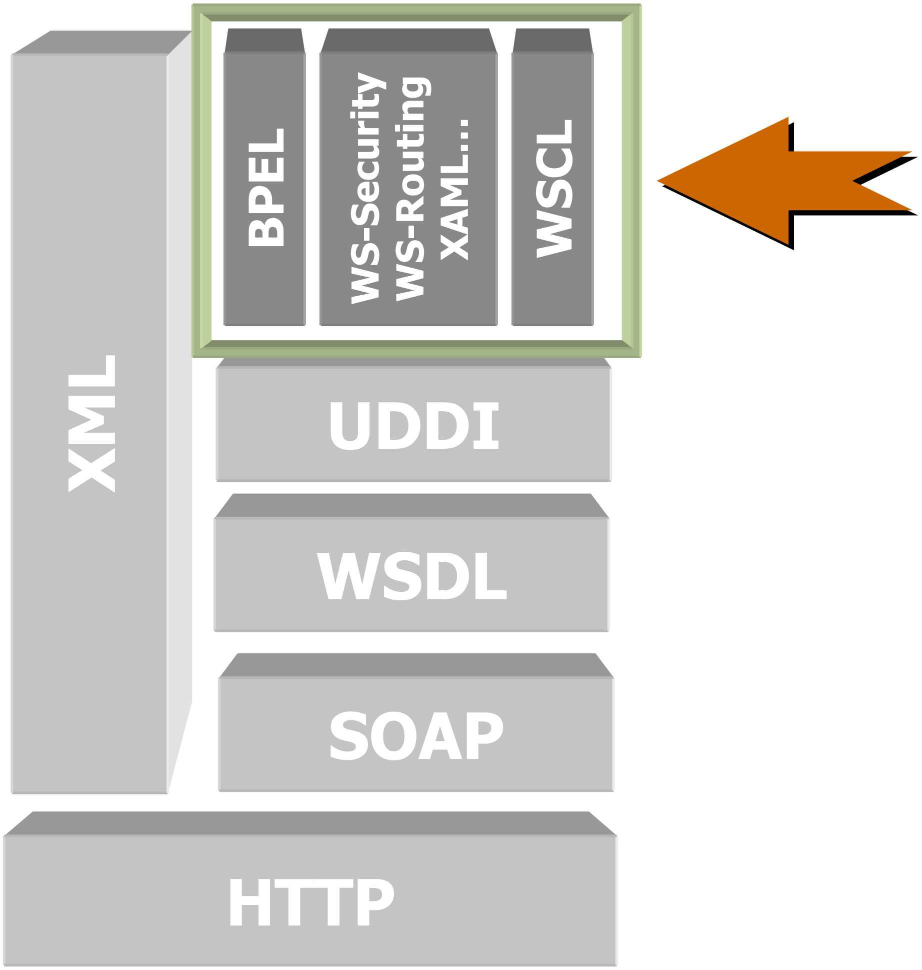
UDDI (Universal Discovery, Description & Integration)

- Yellow Pages Directory Service for Web Services
- Keyword searches based on standard taxonomies
  - NAICS (North American Industrial Classification System)
  - SIC (Standard Industrial Classification)
- White Pages lookup for
  - Service Providers, contact details, etc.
- Service types are registered as a unique tModel
- API to UDDI servers communicate using SOAP

UDDI Search Capabilities

- Four methods for searching available through API:
  - Find_business
    - Locate information about one or more businesses.
  - Find_binding
    - Locate specific bindings within a registered business.
  - Find_service
    - Locate specific service within a registered Buisness Entity.
  - Find_tModel
    - Locate one or more tModel Information structures.
- Arguments to searches are keyword based
  - Uses keywords to guide search:
    - *find business named IB**
  - Use tModels to find services with a feature:
    - *find all services with WSDL specification*

Overview of Web Services Standards



Defining the business process and protocols using **BPEL**

BPEL

- Description of how Web Services are composed
  - Flow Model describes the structure of the business process in terms of:
    - Activities
      - Describe the process steps
    - Data and Control Links
      - Represent sequencing rules and information flow
  - Global Model
    - Describes interaction between provider and requester
    - Mappings between internal operations and WSDL port types
    - BPEL Plug links map between control flow and WSDL operations

Limitations of Current Web Service Standards

- Lack computer understandable semantics
- Provide limited Interoperation
- Allow for keyword based service discovery
- Lack ability for run time service discovery, negotiation, execution monitoring and composition
- Lack mechanisms for describing business relations, or agreements, e.g. contracts

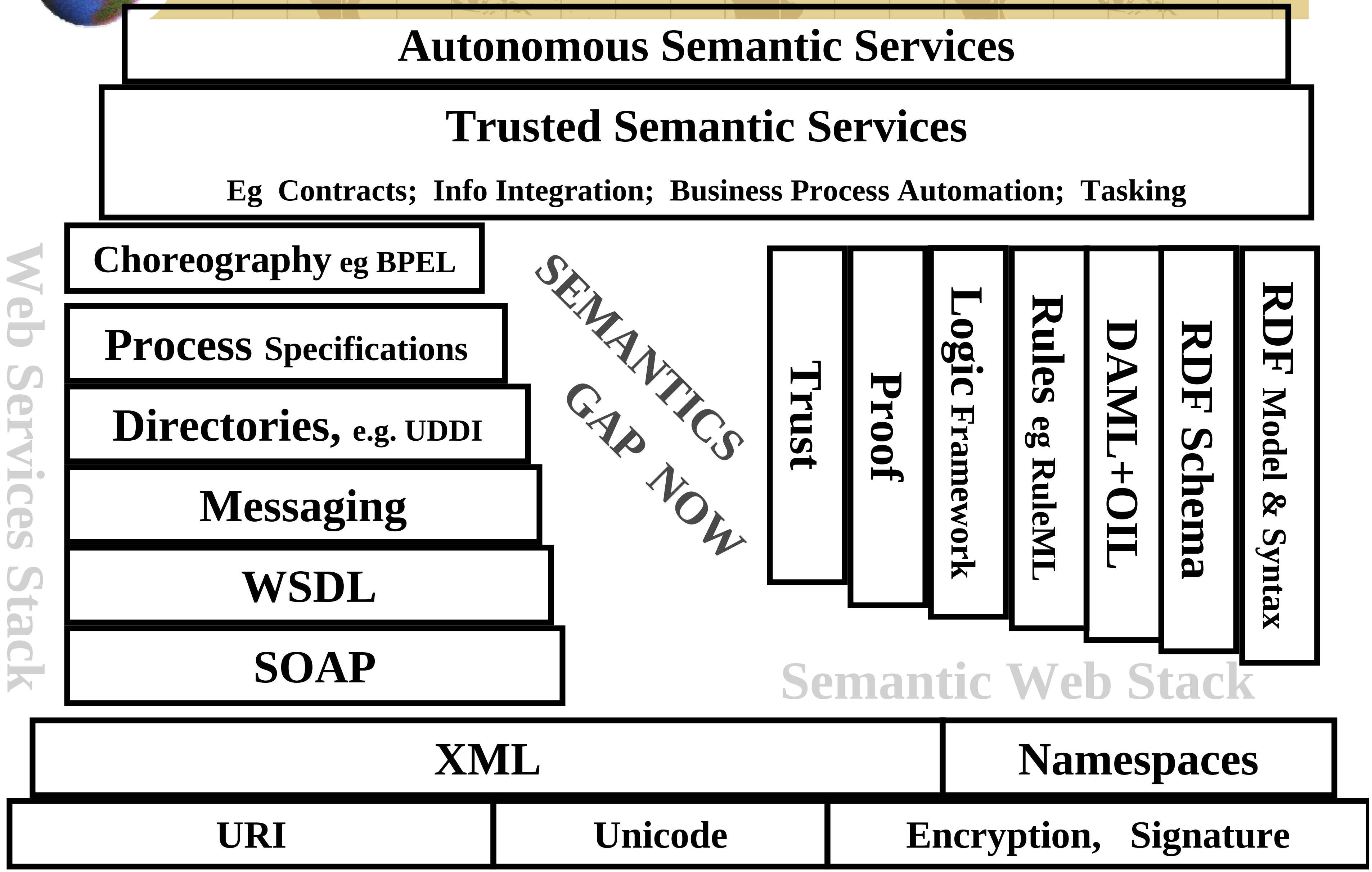
Outline

1. The Vision
2. What are Web Services
3. Industry Standards
4. **Semantic Web Services**
5. Semantic Mediation
6. Autonomous Semantic Web Services
7. Conclusion

The Need for Semantics

- Lack of Semantic Interoperability is a major hurdle for
  - Locating Services
    - Different terms used for advertisements and requests
  - Invoking Services
    - Constructing valid messages based on the published signature/interface of a service
  - Understanding
    - Interpreting the results of invoking a service
  - Composing Services
    - Combining the results of different services in a meaningful workflow
    - Constructing plans to achieve meta-goals based on available Services/Agents

Semantic Web Services Stack Diagram



Autonomous Semantic Services

Trusted Semantic Services

Eg Contracts; Info Integration; Business Process Automation; Tasking

Web Services Stack

Choreography eg BPEL

Process Specifications

Directories, e.g. UDDI

Messaging

WSDL

SOAP

SEMANTICS GAP NOW

Semantic Web Stack

Trust

Proof

Logic Framework

Rules eg RuleML

DAML+OIL

RDF Schema

RDF Model & Syntax

XML

Namespaces

URI

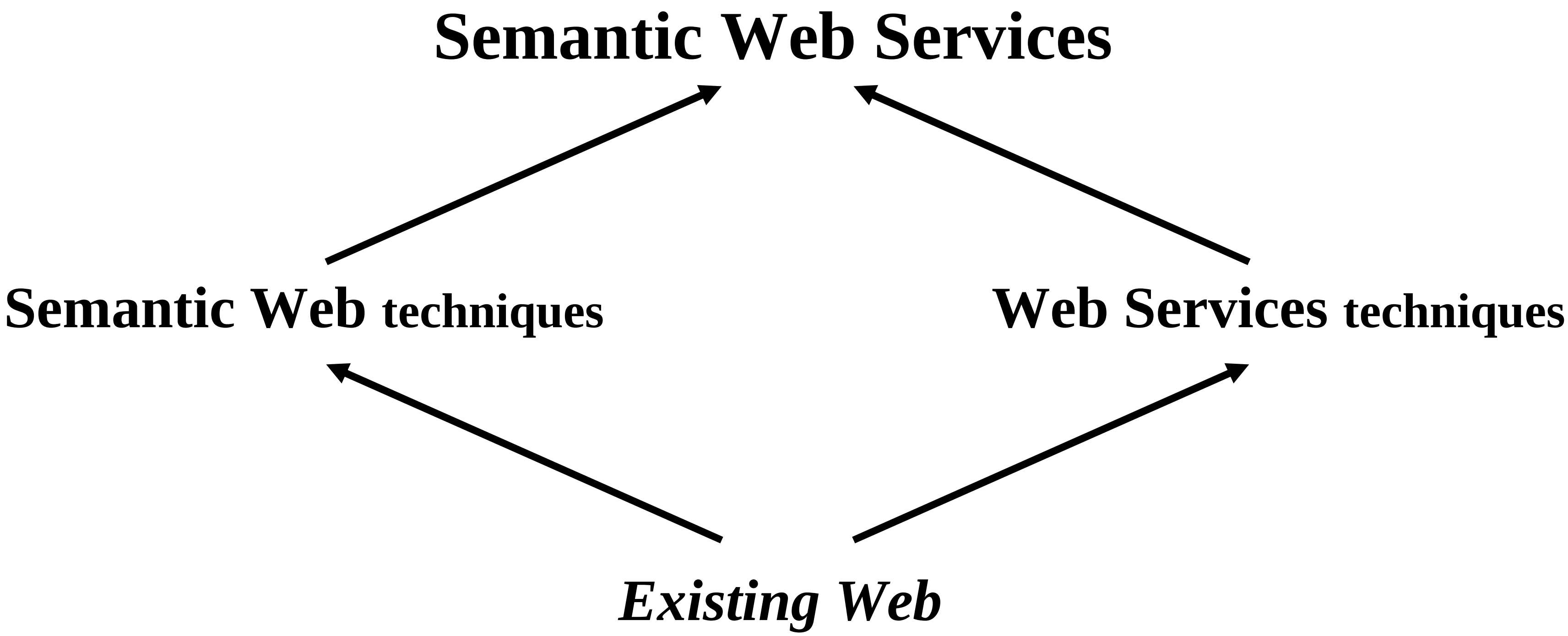
Unicode

Encryption, Signature

Semantic Web Services

- Semantic Service Description
  - Service discovery
  - Service invocation
  - Service negotiation
  - Service selection
  - Service composition
- Semantic Interoperation and Mediation Mechanisms

Next Generation Web



Semantic Web Services

Semantic Web techniques

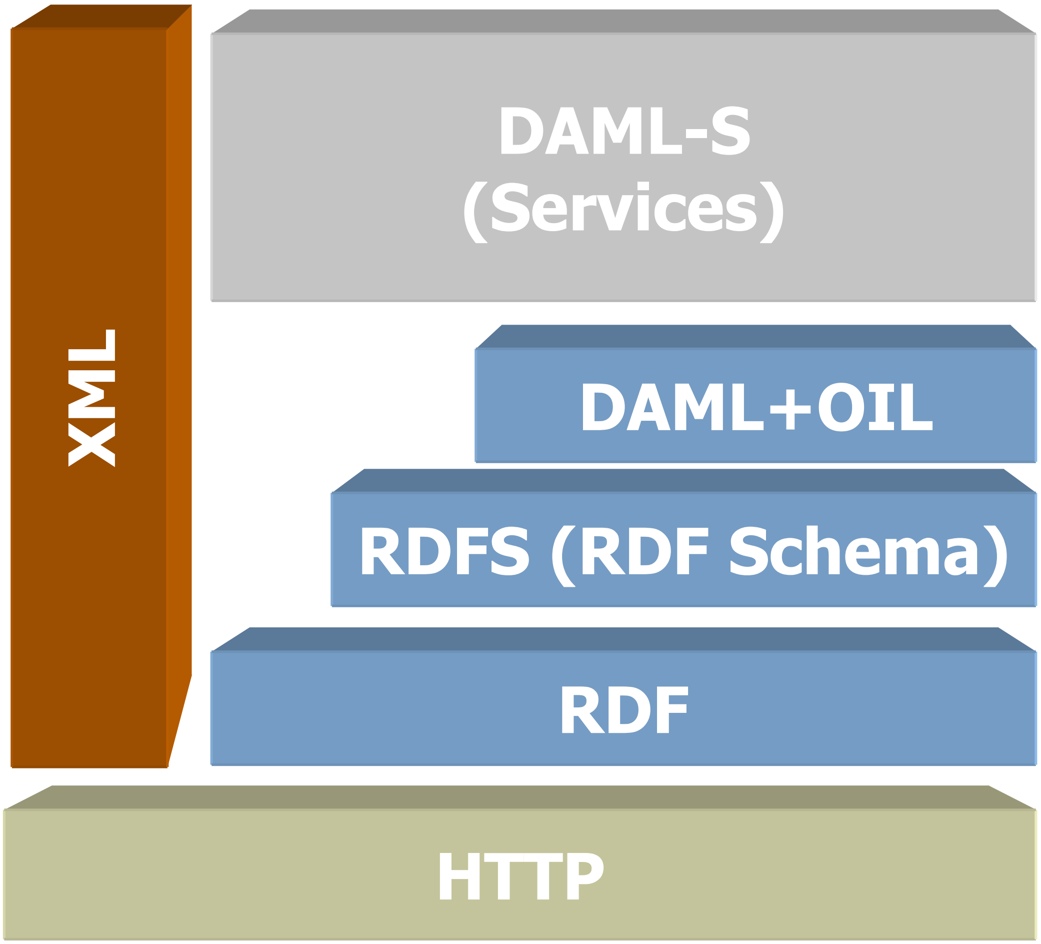
Web Services techniques

Existing Web

Semantic Web Services

- Complete don’t Compete
  - Augment the Web Services infrastructure with semantic information
  - Produce ontologies to describe Web Services
- Enabled by the Semantic Web
  - Capability (requirements) based discovery
  - Meaningful invocation
  - Automatic Web Services composition

Layered Approach to Language Development



DAML-S (Services)

XML

DAML+OIL

RDFS (RDF Schema)

RDF

HTTP

- The first major application of DAML+OIL
- Layer exists above DAML+OIL & RDF
- Future versions will build upon emerging layers (e.g. DAML-Rules etc)

DAML+OIL

- DAML+OIL extends RDF statements to provide a rich descriptive logic language
  - Provides restrictions and additional notations on properties
    - Cardinality restrictions
    - Notations include *inverseOf*, *Transitivity*, etc
  - Provides additional properties for class definitions
    - *Disjoint-with*, *complement-Of*, *intersectionOf*, etc
  - Provides universal & existential quantification through class restriction

DAML+OIL will be succeeded by the emerging Web Ontology language OWL

DAML-S

- **DAML-S**: A **D**ARPA **A**gent **M**arkup **L**anguage for **S**ervices
  - An upper ontology for describing properties & capabilities of Web services in an unambiguous, computer interpretable markup language.
  - DAML+OIL Ontology for (Web) services
- AI-inspired markup language:
  - tailored to the representational needs of Services
  - expressive power
  - well-defined semantics
  - ontologies support reuse, mapping, succinct markup, ...

http://www.daml.org/services

Acknowledgements to the DAML-S Web Services Coalition

CMU: Katia Sycara, Anupriya Ankolekar, Massimo Paolucci

BBN: Mark Burstein

ISI: Jerry Hobbs

U. Of Maryland: Bijan Parsia

Nokia: Ora Lassila

Southampton: Terry Payne

Stanford KSL: Sheila McIlraith, Honglei Zeng

SRI: David Martin

Yale: Drew McDermott

Several of these slides courtesy of Sheila McIlraith, Stanford KSL, Terry Payne, Southampton, & David Martin, SRI

DAML-S Objectives

- Provide:
  - an upper ontology for describing properties & capabilities of agents & (Web) services in an unambiguous, computer interpretable markup language.
- Desiderata:
  - an ontology of Web services
  - ease of expressiveness
  - enables automation of service use by agents
  - enables reasoning about service properties and capabilities

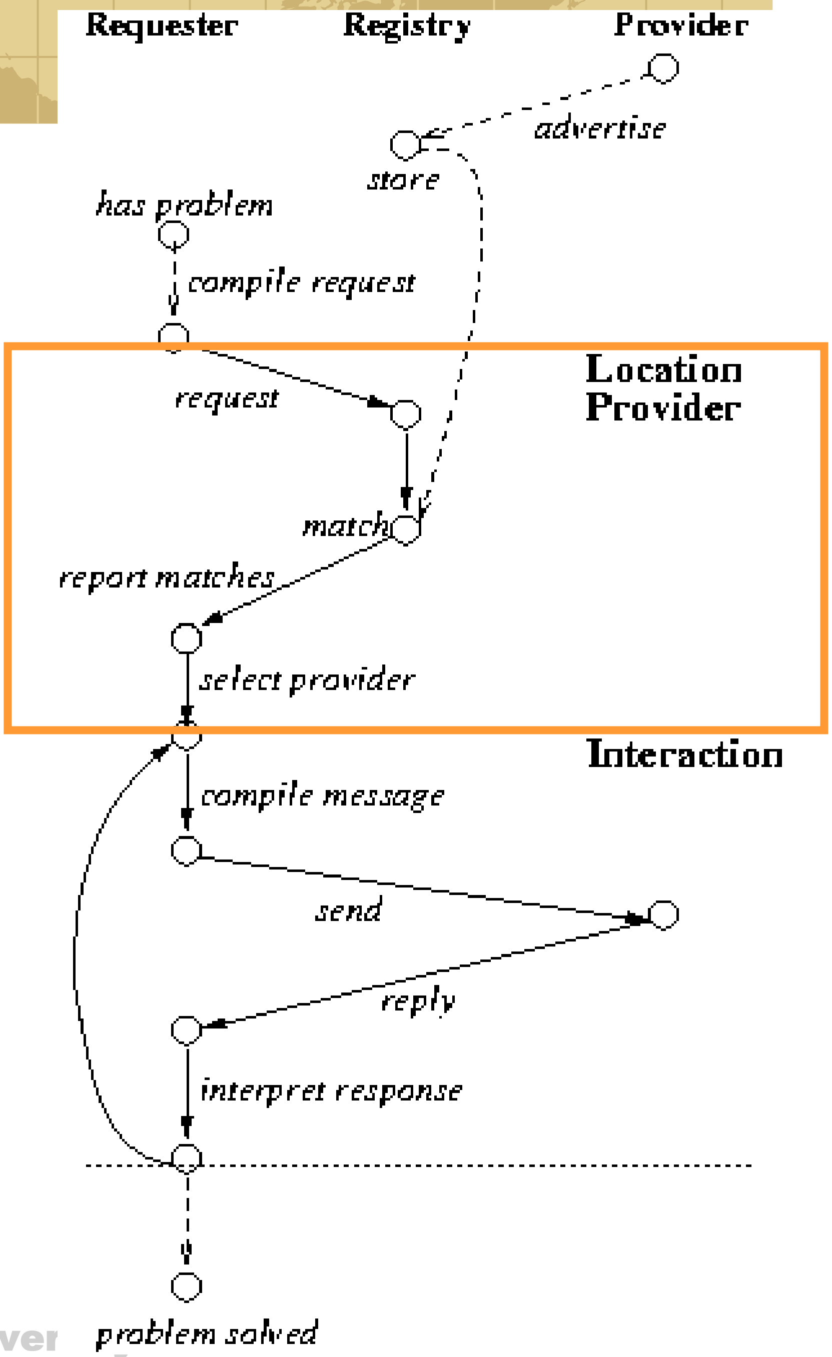
Automation enabled by DAML-S

- Web Service Discovery & Selection
  - Find me an airline that can fly me to Klagenfurt
- Web Service Invocation
  - Book flight tickets with Lufthansa to arrive on June 17th
- Web Service Composition & Interoperation
  - Arrange travel to CAiSE 2003.
- Web Service Execution Monitoring
  - Has the hotel room been reserved ?

DAML-S Elements

	DAML-S	Web Services Infrastructure
Discovery	Profile	UDDI API
Choreography	Process Model	WSCI BPEL4WS
Invocation	Grounding+ WSDL	WSDL

Discovery


Requester
Registry
Provider
advertise
store
has problem
compile request
Location Provider
request
match
report matches
select provider
Interaction
compile message
send
reply
interpret response
problem solved

DAML-S Service Profile (capability representation)

Functionality Description

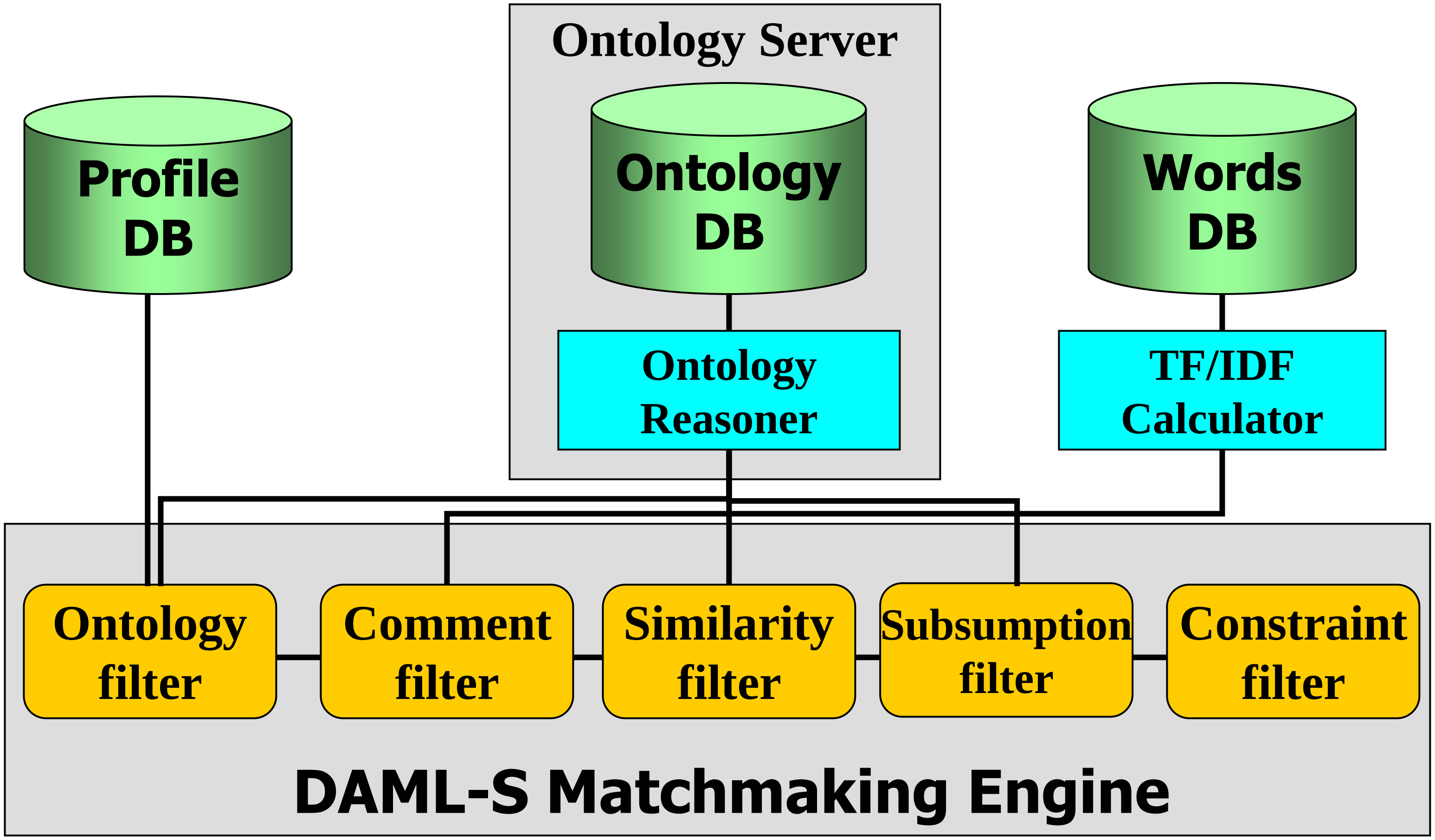
- ***Preconditions***
  - Set of conditions that should hold prior to the service being invoked
- ***Inputs***
  - Set of necessary inputs that the requester should provide to invoke the service
- ***Outputs***
  - Results that the requester should expect after interaction with the service provider is completed
- ***Effects***
  - Set of statements that should hold true if the service is invoked successfully.
  - Often refer to real-world effects
    - Package being delivered
    - Credit card being debited

DAML-S Service Profile

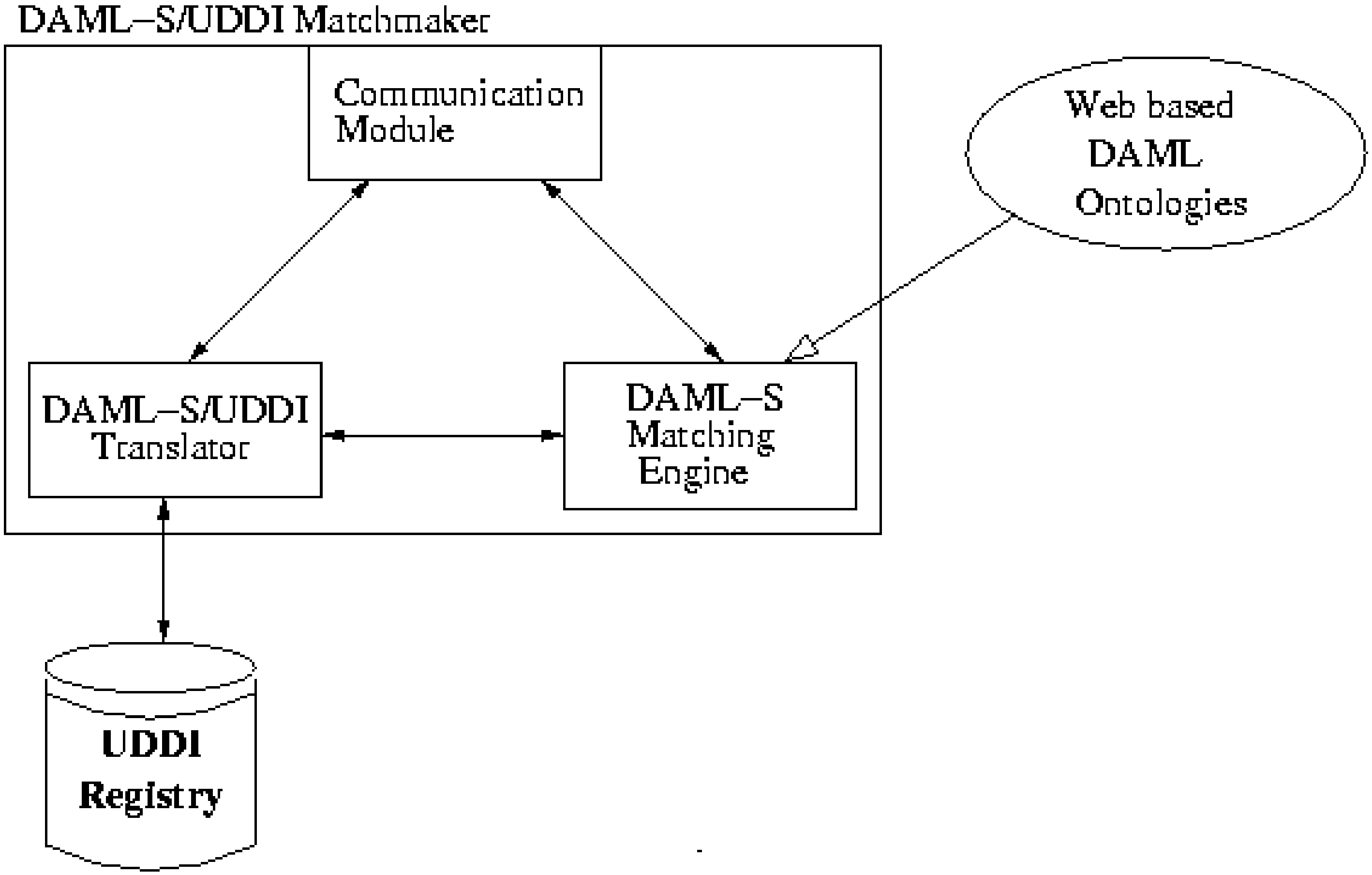
Non Functional Parameters

- ***Service Category***
- ***Company Information***
- ***Service Range***
- ***Quality Rating***
- ***Security requirements***
- ***Response time***
- ***Cost of invoking the service***

DAML-S Matchmaker Processing Module



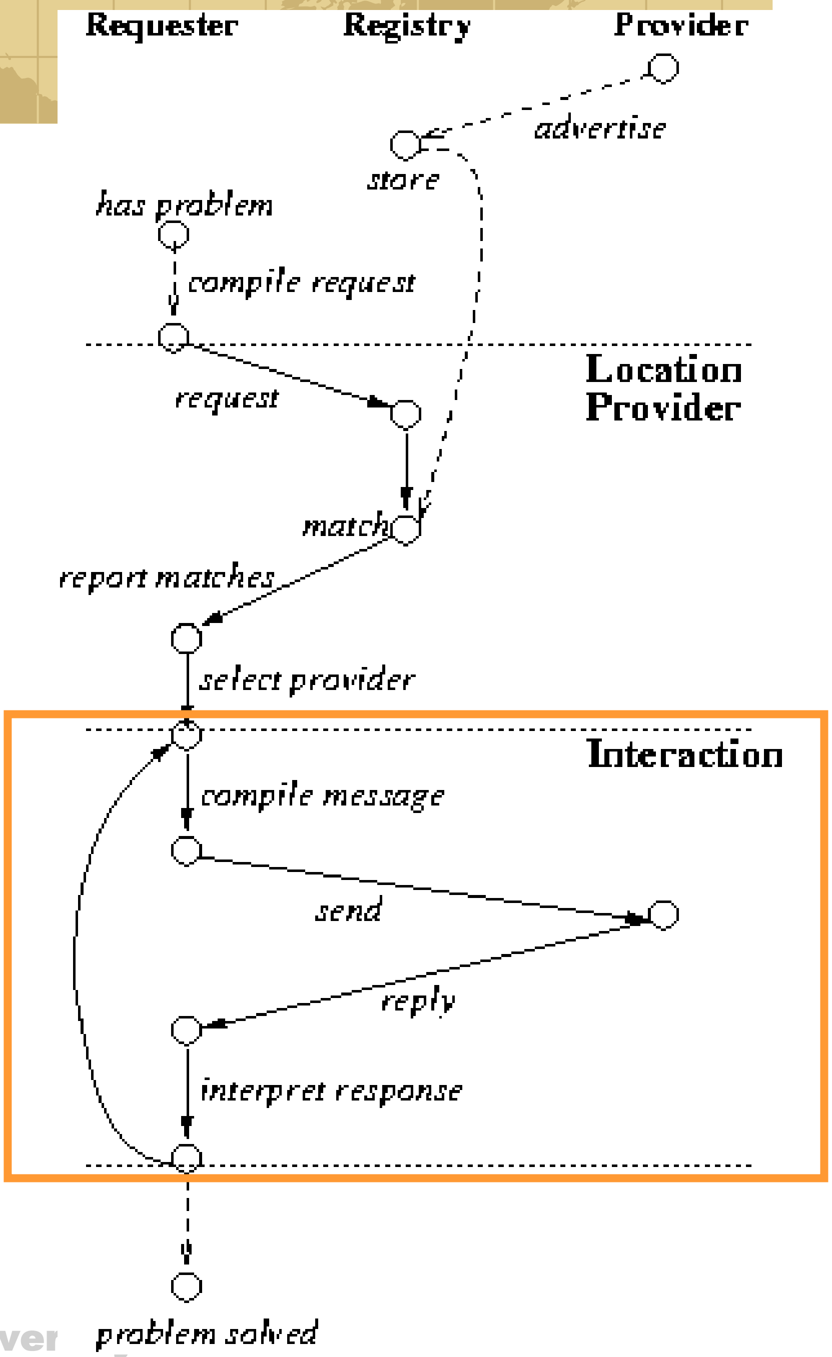
Architecture DAML-S/UDDI



DAML-S for P2P

- Use DAML-S to expand search mechanism on Gnutella P2P network
  - Search capabilities in Gnutella restricted to keyword search – No Semantic Information
- **Improve on Gnutella** by adding semantics in DAML and capability representation in DAML-S
  - Removes the need of centralized Registry
- Protocol:
  - Non DAML nodes allow requests to hop from node to node
  - DAML nodes reason about the requests that they receive and decide whether to accept the task

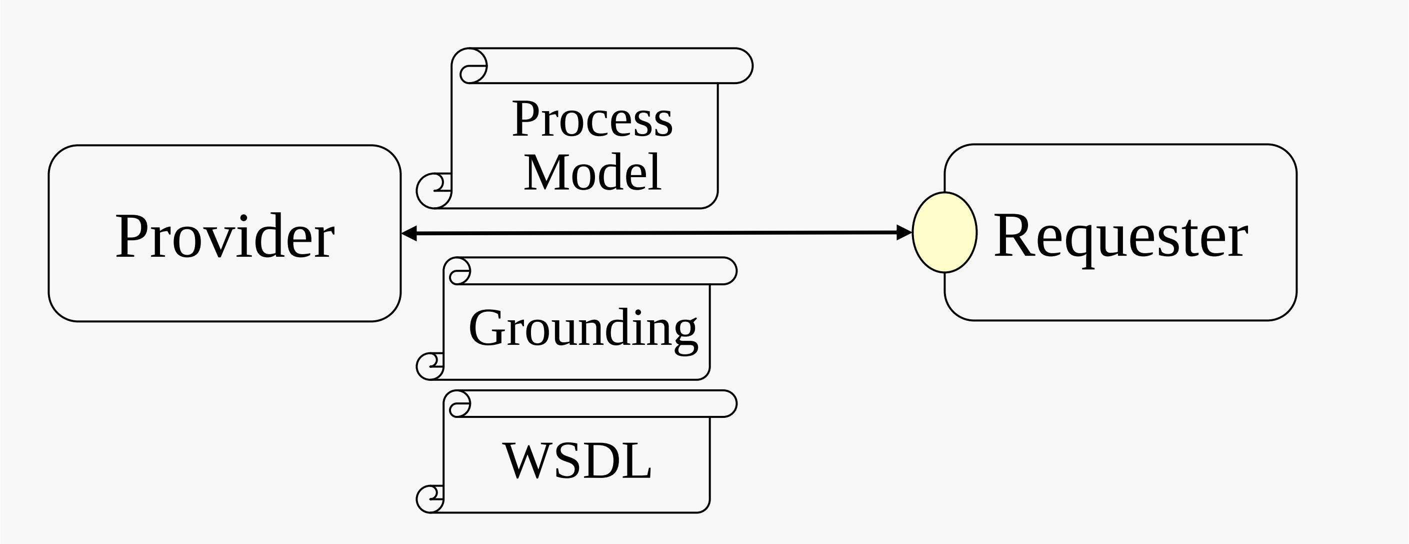
Invocation


Requester
Registry
Provider
advertise
store
has problem
compile request
Location Provider
request
match
report matches
select provider
Interaction
compile message
send
reply
interpret response
problem solved

Requirements of Interaction

- Shared Knowledge of interaction protocols:
  - What information the provider needs
  - When does it needs it
  - In what order (workflow)
  - In general interaction is peer to peer
- Shared understanding of content of messages
  - Ontologies to impart semantics
  - Logic framework for correct interpretation
- Agreement on ports and low level details

Autonomous Invocation



- Provider publishes
  - Process Model (black, glass or white box model)
  - Grounding, WSDL
- Requester uses them to initiate the interaction with the provider

Process Model

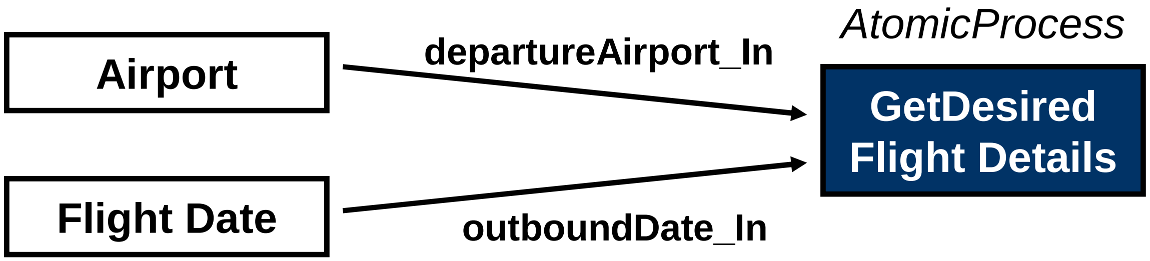
- Processes are conceived as:
  - Atomic
  - Simple
  - Composite
- Associated with each service is a set of:
  - Inputs
  - Outputs
  - Preconditions
  - Effects
- Invocable processes have an associated grounding:
  - Includes WSDL description to model:
    - Operation
    - Message formats
    - Ports & Bindings

Process Model

- Composite processes are compositions of simple or other composite processes in terms of constructs:
  - Sequence
  - if-then-else
  - Fork
  - Etc.
- Data flow and Control flow should be described for each composite service
- A black box and glass box view may be given of each composite service

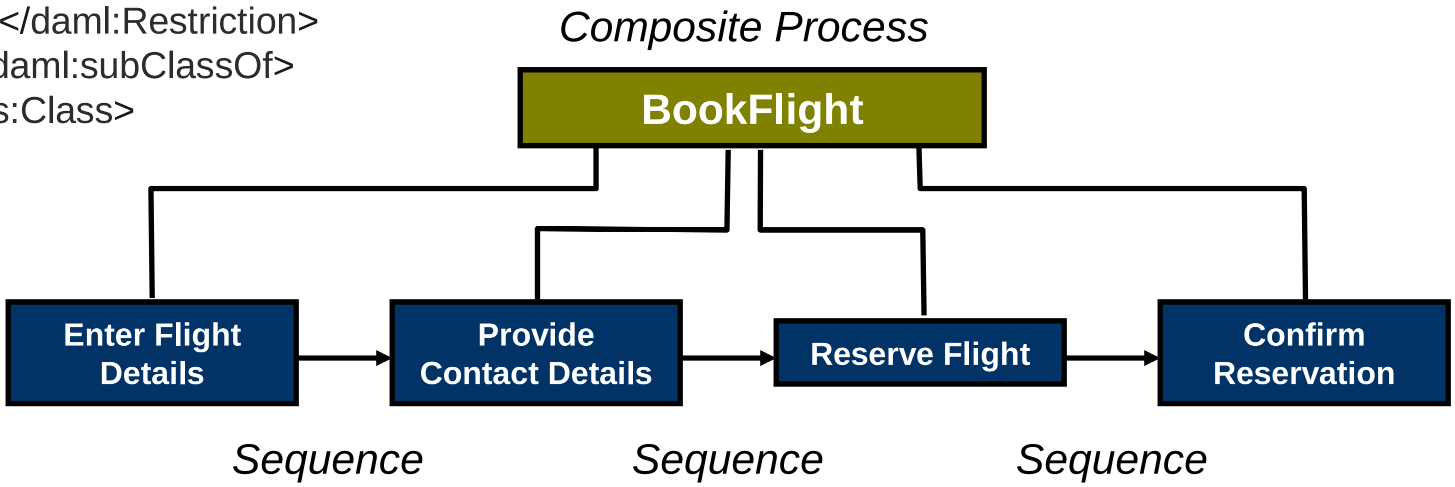
Atomic Process Example

```
<rdfs:Class rdf:ID       ="GetDesiredFlightDetails">
   <rdfs:subClassOf rdf:resource="http://www.daml.org/Process#AtomicProcess" />
</rdfs:Class>


<rdf:Property rdf:ID="departureAirport_In">
   <rdfs:subPropertyOf rdf:resource="http://www.daml.org/Process#input" />
   <rdfs:domain rdf:resource="#GetDesiredFlightDetails" />
   <rdfs:range rdf:resource="http://www.daml.ri.cmu.edu/ont/
              DAML-S/concepts.daml#Airport" />
</rdf:Property>

<rdf:Property rdf:ID="outbounDate_In">
<rdfs:subPropertyOf rdf:resource="http://www.daml.org/Process#input" />
   <rdfs:domain rdf:resource="#GetDesiredFlightDetails" />
   <rdfs:range rdf:resource="http://www.daml.ri.cmu.edu/ont/
            DAML-S/concepts.daml#FlightDate" />
</rdf:Property>
```



Composite Process Example

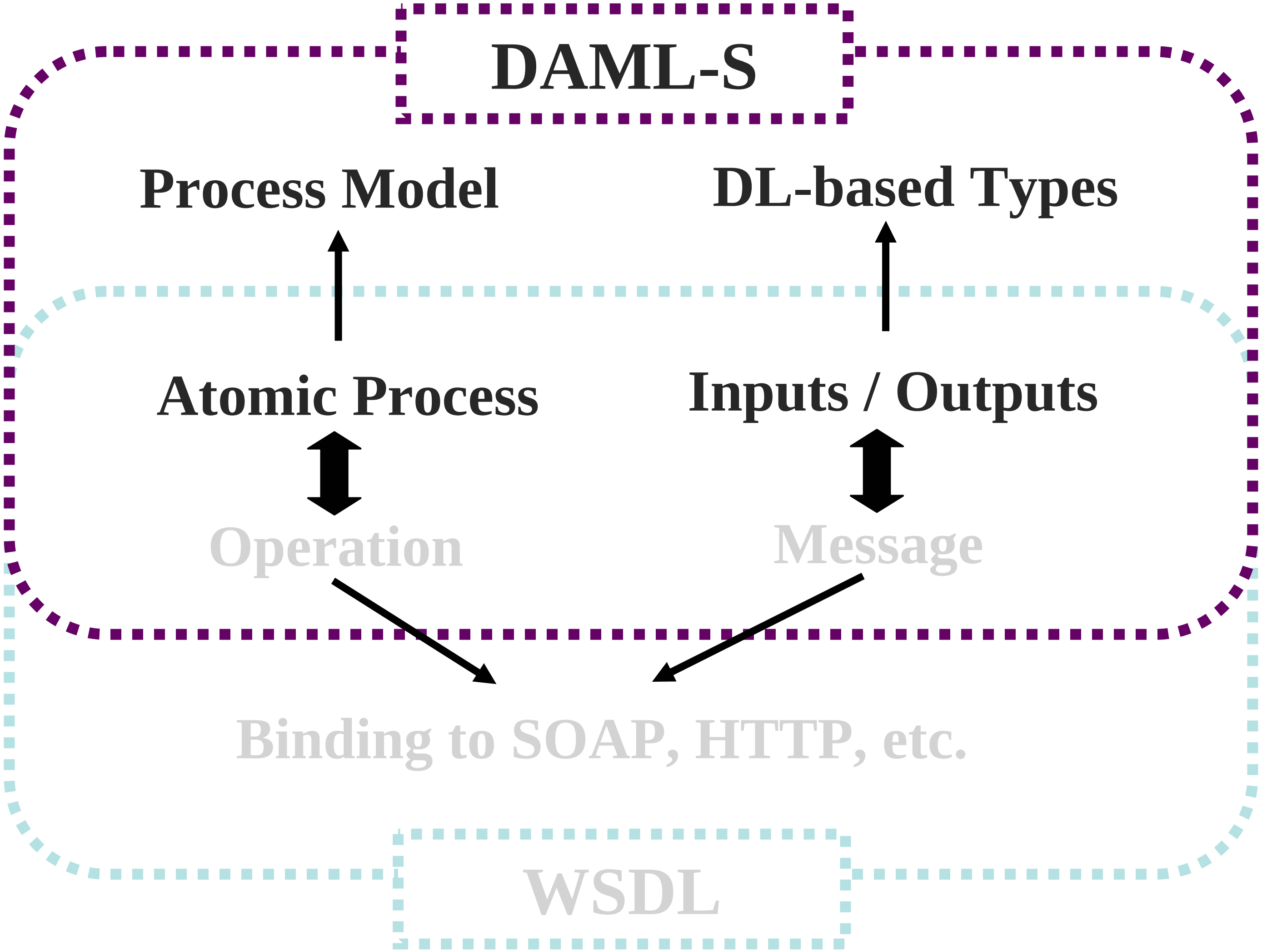
```
<rdfs:Class rdf:ID="BookFlight">
   <rdfs:subClassOf rdf:resource="#CompositeProcess" />
   <rdfs:subClassOf rdf:resource="http://www.daml.org/Process#Sequence" />
   <daml:subClassOf>
      <daml:Restriction>
         <daml:onProperty
            rdf:resource="http://www.daml.org/Process#components" />
         <daml:toClass>
            <daml:subClassOf>
               <daml:unionOf rdf:parseType="daml:collection">
                  <rdfs:Class rdfs:about="#ReserveFlight" />
                  <rdfs:Class rdfs:about="#ConfirmReservation" />
               </daml:unionOf>
            </daml:subClassOf>
         </daml:toClass>
      </daml:Restriction>
   </daml:subClassOf>
</rdfs:Class>
```



Grounding

- Specifies mapping to WSDL
  - Atomic Processes map to Operations
  - Inputs/Outputs described as messages
  - Specify XSLT transformations for mapping to and from DAML and XSD types

DAML-S / WSDL Binding



DAML-S

Process Model

DL-based Types

Atomic Process

Inputs / Outputs

Operation

Message

Binding to SOAP, HTTP, etc.

WSDL

DAML-S Virtual Machine

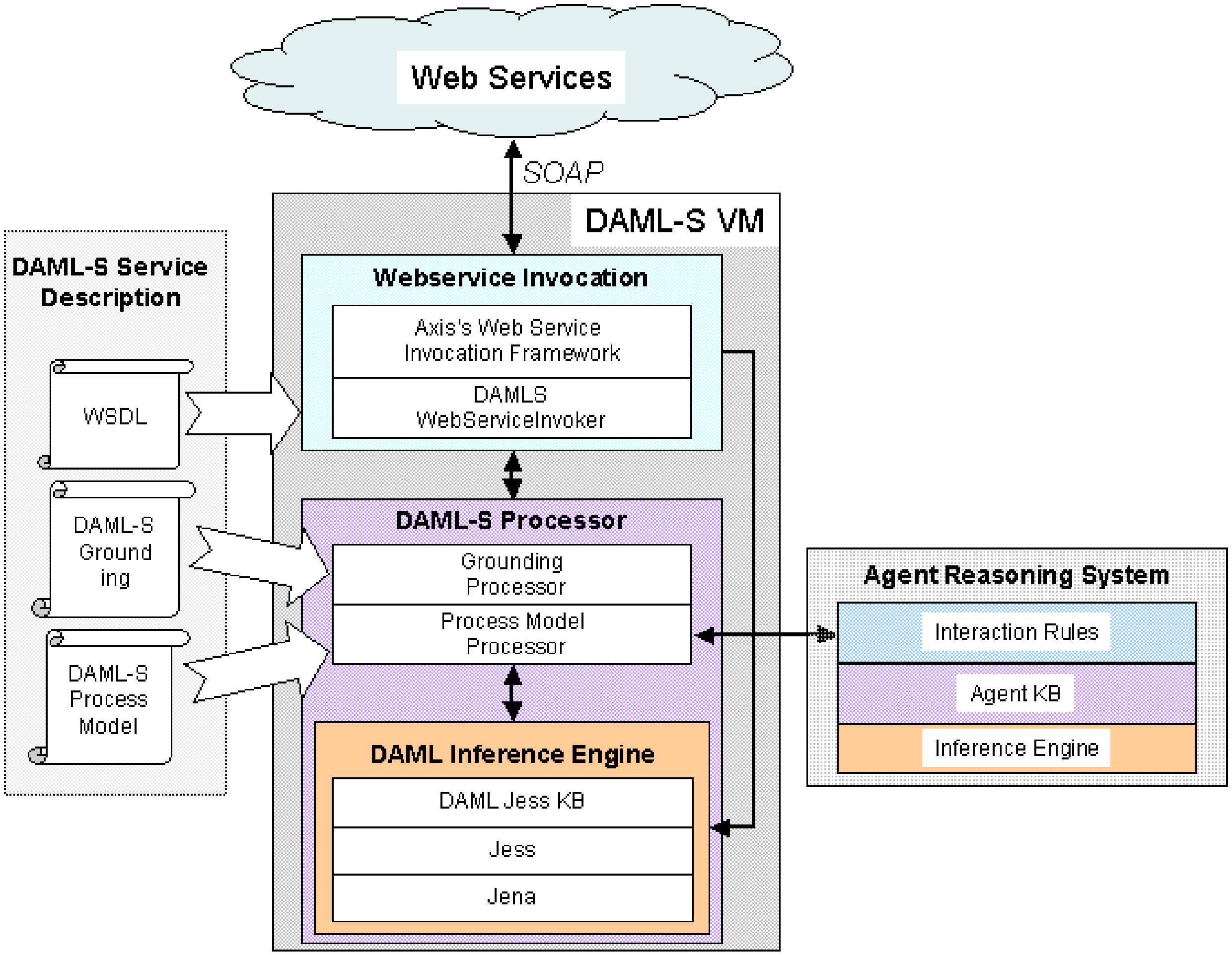
- **Webservice Invocation:** Uses Axis's WSIF to generate SOAP messages consistent with the WSDL specification of the web service grounding.

- **DAML-S Processor**
- **DAML Processor** provides the basic inferencing on DAML ontologies.
- **Process Model Processor** contains rules that implement process model execution semantics to execute the process model of web services.
- **Grounding Processor** uses web services Grounding specification to provide information to extract web services invocation information.

- **DAML Parser and Inference Engine:** DAML-S Descriptions (Process Model and Grounding Information) are parsed into JESS predicates

DAML-S VM

DAML-S processor that allows any Web service to interact with Web services using only DAML-S specifications



WSDL2DAML-S

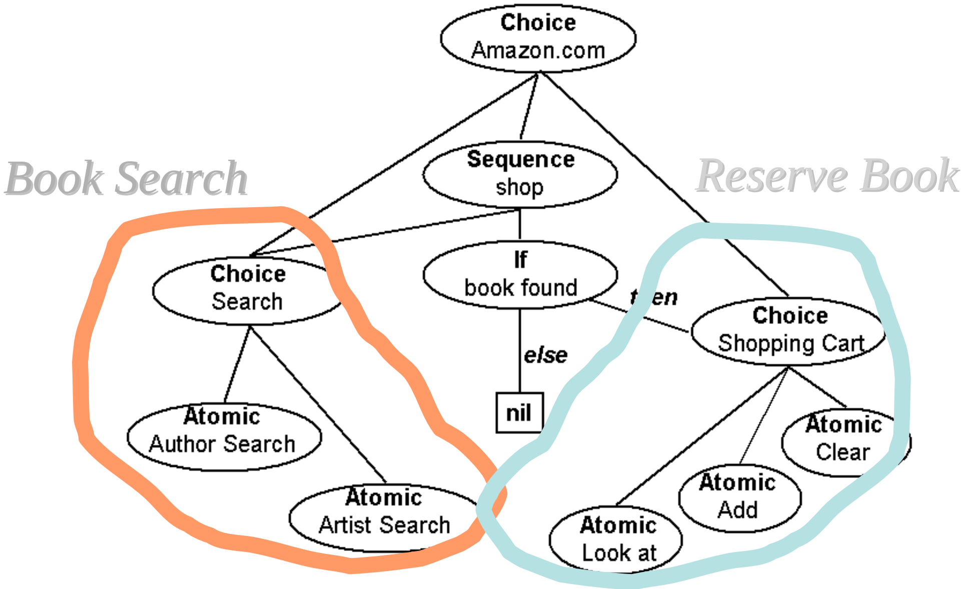
- WSDL widely used to describe Web services
  - Wide repositories of WSDL descriptions
    - www.salcentral.com   www.xmethods.com
- WSDL2DAMLS allows easy translation of WSDL documents in DAML-S
  - Automatic generation of Grounding
  - Partial generation of Process Model and Profile
  - Up to 80% of work required to generate a DAML-S description is done automatically
- Combined with Java2WSDL to provide
  Java2DAML-S

Composition of Services

- Services may themselves be composed by a number of other services.
  - Can be broken down into a hierarchy of subtasks
  - Subtasks may be part of a larger service offered by a service provider
    - e.g. process of logging into an account
  - May be offered by a different service provider
    - e.g. booking a hotel as part of a travel plan

DAMLzon: DAML-S for Amazon.com

- WSDL2DAML-S used to generate DAML-S for Amazon's Web Service
- DAML-S VM used to interact with Amazon Web service

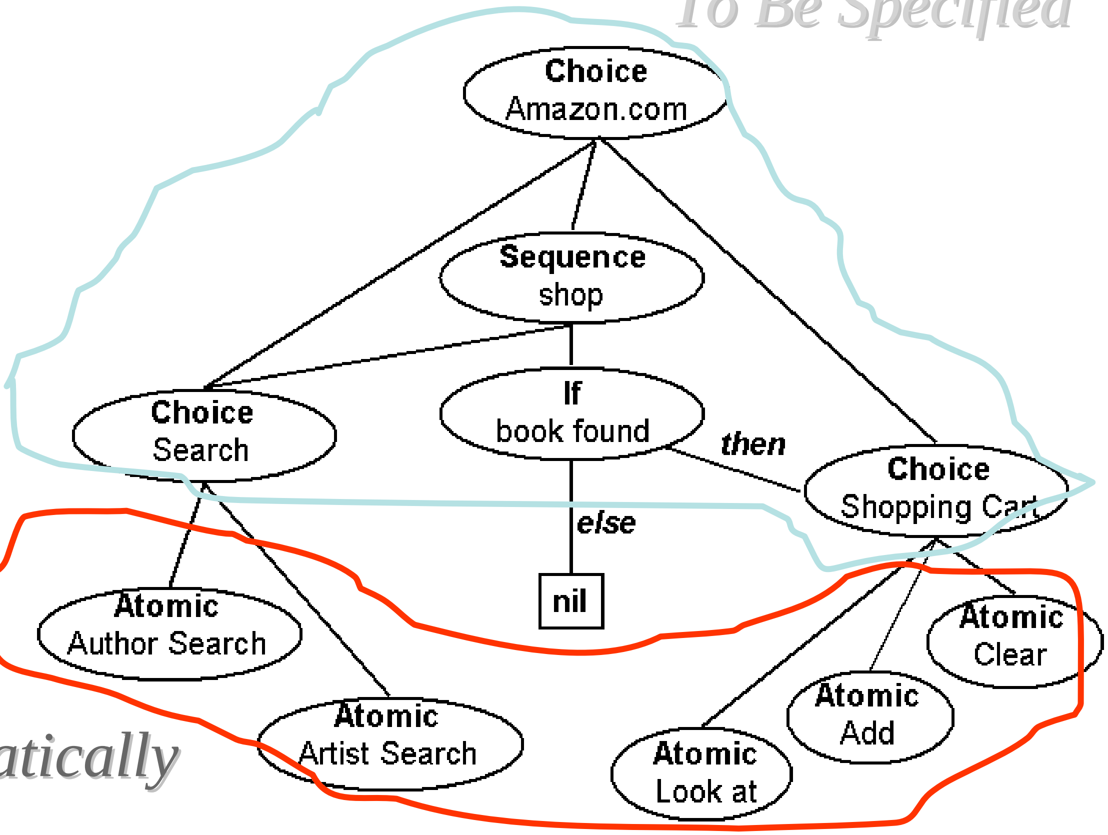


Process Model for Amazon.com

Example WSDL2DAML-S

- Atomic Processes are generated automatically
- Need to be specified
  - Composite Processes
  - mapping XSD to DAML

To Be Specified

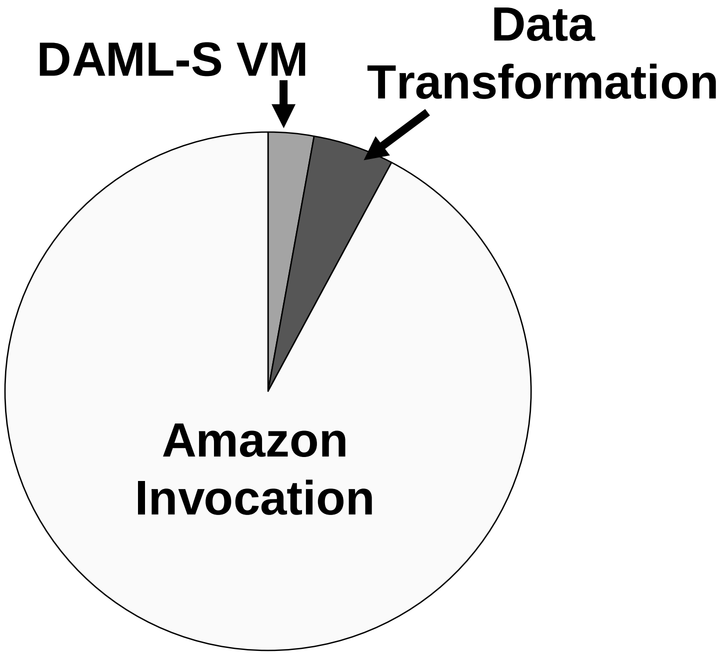


Generated Automatically

Process Model for Amazon.com

Performance

- DAML-S VM client on browsing+reserving task
- Analyzed data by computing:
  - Time required by DAML-S VM to execute Process Model
  - Time required for data transformation to fit Amazon requirements
  - Time required to invoke an operation on Amazon
- 98 runs total over 4 days in varying load conditions
- Results in milliseconds



	VM	Data Trsfm	Invocation
Average ***percentage***	83 *3%*	156 *5%*	2797 *92%*
Strd dev	107	146	1314

Outline

1. The Vision
2. What are Web Services
3. Industry Standards
4. Semantic Web Services
5. **Semantic Mediation**
6. Autonomous Semantic Web Services
7. Conclusion

Semantic Mediation of Services

- Service Discovery: Processing service provider capabilities and service requests
  - e.g. matchmaking, brokering, recommending, facilitating
- Ontological Mediation
- Semantic Interoperability
  - e.g. resolving semantic mismatches between service descriptions, or identifying articulations between ontologies
- Communication Mediation
- Trust Management

Mediating Communication

- Services may use different communication languages
  - e.g. SOAP vs. KQML
- Transactions may assume different protocols and policies
  - e.g. virtual marketplaces may assume different auction policies
- Semantic mismatch of knowledge may require intermediary to translate between ontologies

Mediating Reliability, Security & Trust

- Quality of Service
  - Providers should comply with data requirements, standards and policies regarding knowledge and data stored
- Trust Management
  - Provide guarantees that service providers provide the service they advertise
  - Escrow Services
  - Privacy and anonymization
  - Prevent abuse of shared, private information
    - such as selling contact information or preferences
- Security, Authorization and Verification services
  - Certification authorities and encryption keys

Outline

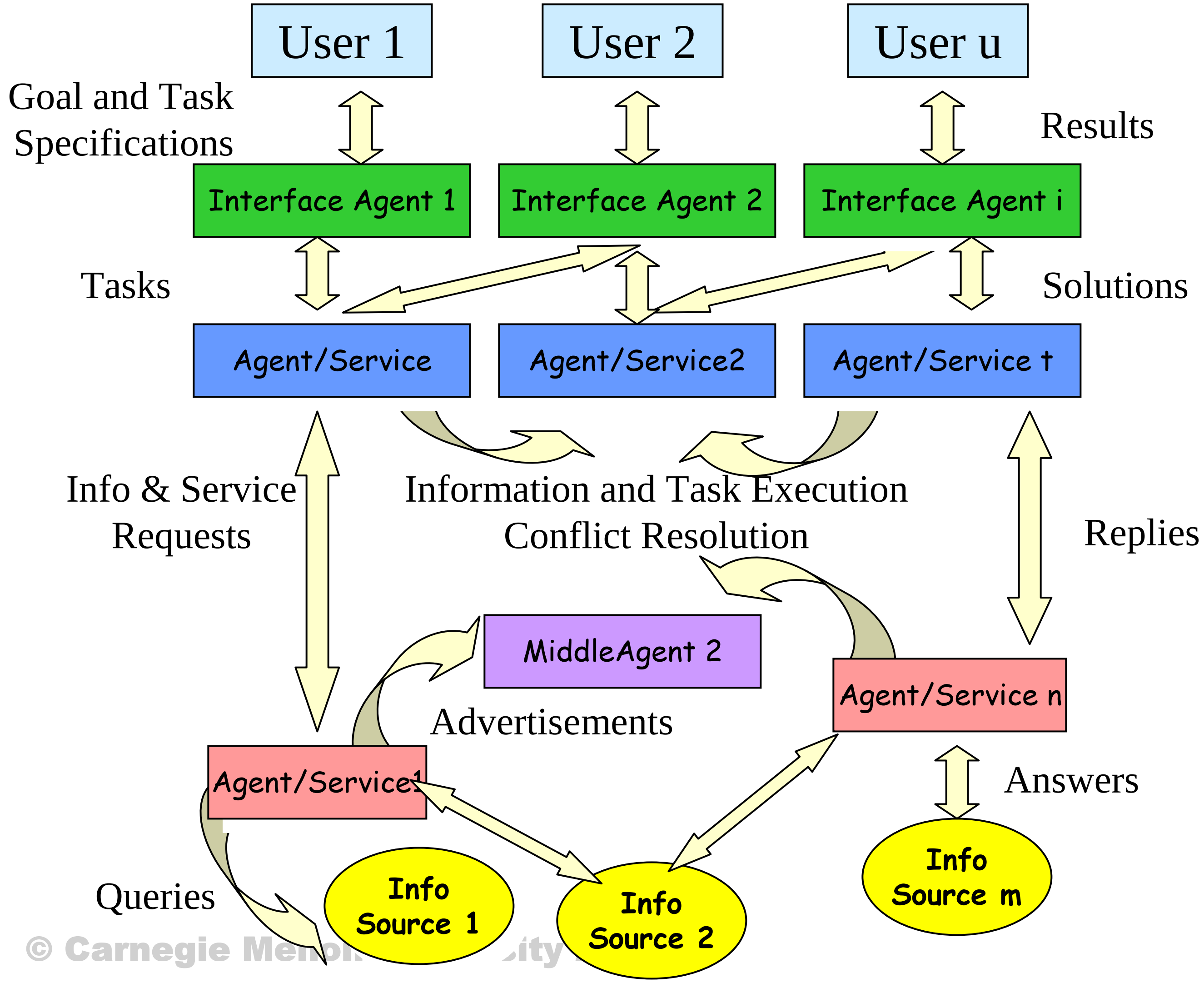
1. The Vision
2. What are Web Services
3. Industry Standards
4. Semantic Web Services
5. Semantic Mediation
6. **Autonomous Semantic Web Services**
7. Conclusion

Autonomous Semantic Web Services

- Open and Dynamic Environments
  - agents / services may be transient
  - agent locations change
    - system load balancing
  - agent identity changes
    - cannot predict its name
    - cannot predict the vocabulary used to describe it
- On the fly construction of plans that achieve user goals based on available services
  - Requires agents with planning abilities
- Service Redundancy
  - multiple/ competing service providers modelled at different levels of abstraction
  - differentiate on service parameters
    - speed, price, security, reliability, reputation, etc.

Multi-Agent Organization

Distributed adaptive collections of agents (autonomous services) that coordinate to do tasks on the user’s behalf

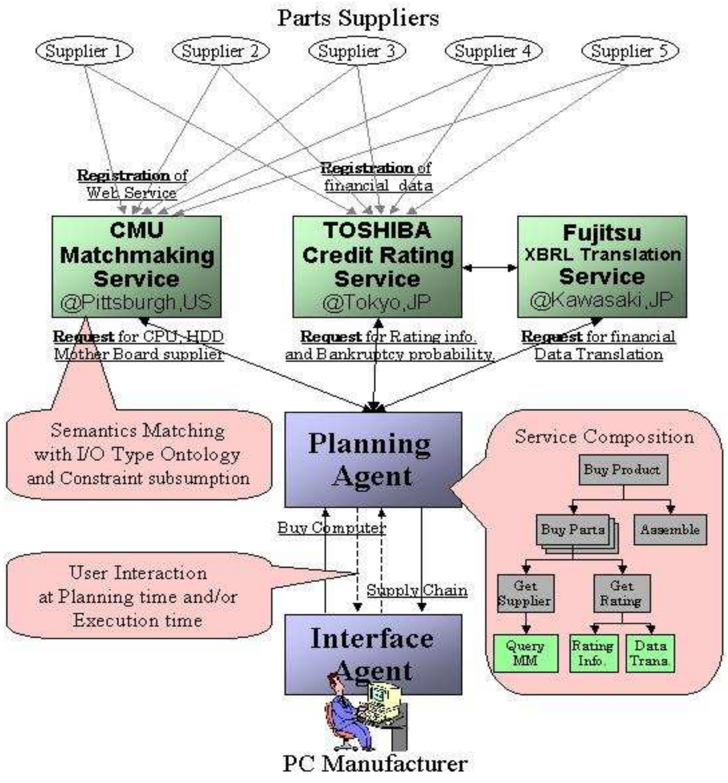


Agent / Agent Interactions

- User Intent Inferencing
- Task Decomposition and Delegation
- Assembly of Services
- Applicability to Physically Embodied Services/Robots

B2B Demo with Toshiba

- Task: organize supply chain for computer manifacturing


Parts Suppliers
Supplier 1
Supplier 2
Supplier 3
Supplier 4
Supplier 5
Registration of Web Service
Registration of financial data
CMU Matchmaking Service @Pittsburgh,US
TOSHIBA Credit Rating Service @Tokyo,JP
Fujitsu XBRL Translation Service @Kawasaki,JP
Request for CPU, HDD Mother Board supplier
Request for Rating info. and Bankruptcy probability.
Request for financial Data Translation
Semantics Matching with I/O Type Ontology and Constraint subsumption
Planning Agent
Service Composition
Buy Product
Buy Parts
Assemble
Get Supplier
Get Rating
Query MM
Rating Info.
Data Trans.
Buy Computer
Supply Chain
User Interaction at Planning time and/or Execution time
Interface Agent
PC Manufacturer

Achieve Ideals of Software Engineering

- Truly reusable and composable software components
- Self describing and self healing
- End User Programming
- Scalable, reliable, robust, and fault-tolerant computing
- Program by high-level service requirement descriptions

Conclusion

- Evolution of Web Services towards Semantic Web Services and finally towards Autonomous Agents
- DAML-S supports different aspects of the Web services interaction, e.g. discovery, autonomous invocation, composition
- The use of DAML-S does not result in a performance penalty
- Autonomous Agents required to take full advantage of semantics