

IT architecture for supporting business interoperability through the use of semantic annotation

Stelios Pantelopoulos
Head of European Projects Department

May 2009

Presentation Layout

- FUSION
 - Short Info
 - Motivation and Research Challenges
 - Project Results
- EMPOWER
 - Short Info
 - Architectural overview

FUSION Short Info

- Title: Business process fusion based on Semantically-enabled Service-oriented Business Applications
- Concept of SE-SOBA
 - BA = Business Applications
 - SO = Service Oriented
 - SE = Semantically Enriched
- Scope: Promote efficient business collaboration and interconnection between enterprises (including SMEs) by developing a framework and innovative technologies for the semantic fusion of heterogeneous service-oriented business applications
- Duration: 2006 – 2008
- 14 partners
 - BA Providers (SAP – ERP, SingularLogic – ERP, CAS – CRM)
 - IT Integrators (InfomatiX, BEELC)
 - Research Institutes (ICCS, SEERC, BUTE, TUK)
 - Pilot users (Germanos, Interjob, Pharos)
 - Dissemination issues (BCCI)

High Level motivations

- **Exploitation of previous knowledge and expertise** (Ontologies and Semantic Web technologies)
 - in the domain Enterprise Application Integration (EAI)
- Interested in **exploring the several aspects of Semantic Web Services** underlying technologies
 - life-cycle of SWS (discovery, annotation, publication, etc)
 - technologies (i.e. SA-WSDL, WSDL-S, WSMO, OWL-S)
- Interested in **examining the applicability** of Semantic Web technologies to **real-life, already deployed systems**
 - e.g. ERP, CRM, HRM

EAI Challenges

- The goal of EAI is to **integrate and streamline heterogeneous business processes across different applications** and business units
- Current EAI trends and technologies (SOA, ESB, Web Services) are up to now **quite mature**
- However, if we try **to increase the level of automation** we confront several problems and **challenges**, such as
 - data and message level heterogeneities between interoperating services,
 - insufficient search and discovery of published Web Services in a common registry, and
 - inadequate process composition with regard to the desired functionality and the operational requirements.

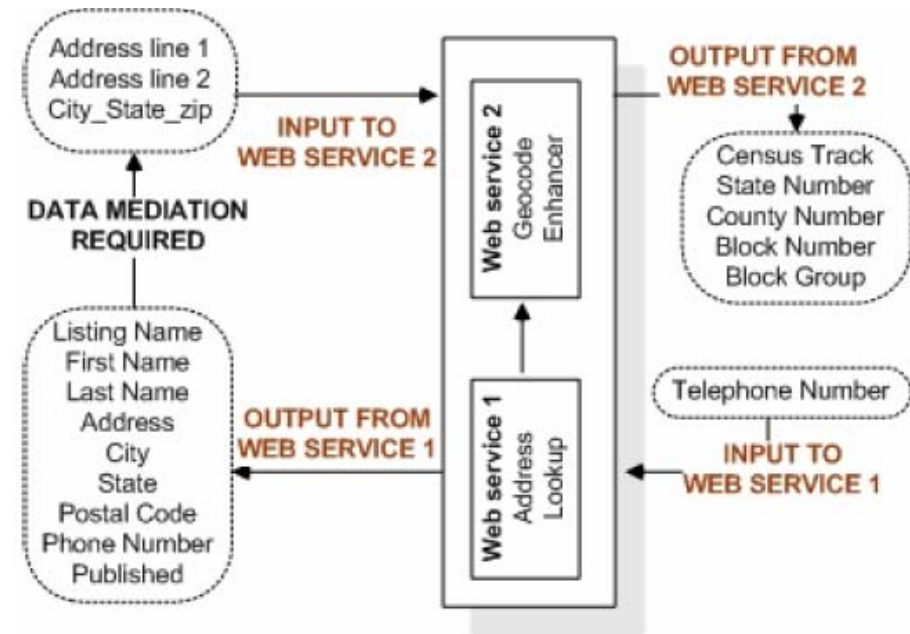
Weaknesses of Syntactic Technologies

- Based on exclusively **syntax-oriented** technologies
- Not defining formally the **semantics** of
 - web services interfaces
 - the data structures of the messages web services exchanges
- Hard to manually interconnect heterogeneous applications
- Impeding the automation regarding application integration and data exchange
- Traditional web services technologies are weak to address:

*the **formalization** and the **documentation** of the **semantics** related to the interfaces and the data structures of the deployed web services*

Examples of Message Heterogeneities

- **Domain Incompatibility**
 - semantically alike attributes have different descriptive names
 - Student (ID, Name) vs. Student (SSN, Name)
- **Entity Definition**
 - semantically alike entities have different descriptive names
 - EMPLOYEE (ID, Name) vs. WORKER (ID, Name)
- **Abstraction Level Incompatibility**
 - combination of domain and entity incompatibilities
 - semantically similar entities are represented at different levels of generalization / abstraction



Research Objectives

- **Semantically-enriched approach for dynamic data mediation** in Enterprise Application Integration scenarios
 - based on Ontologies, Semantic Web and Semantic Web Services Technologies
- Our approach focuses on the **semantic resolution of message level heterogeneities between collaborative enterprise services** exposed from the participating business systems
 - facilitating automatic, dynamic data mediation during execution time by providing formal transformations of the input and output messages (of the participating Web Services) to a common reference model, i.e. an enterprise data ontology

The needs of semantics in EAI

- A **semantically-enriched approach in EAI** will
 - hide systems, syntax, and structural heterogeneity
 - eliminate the need for knowing
 - the contents and structure of information resources
 - the structure and architecture of heterogeneous enterprise applications
 - provide a shared and common understanding of
 - data, services and processes
 - how to facilitate communication between people and IT systems
- We developed
 - An Enterprise Interoperability Ontology
 - that captures and represents formally all entities involved in the EAI scenarios i.e. data, services and processes
 - the Semantic Annotation and Profiling platform
 - which facilitates annotation of input and output message parts of native Web Service interfaces with business data entities

Starting with an in-depth state-of-the-art review and analysis

- the **industrial** perspectives
 - SAP Enterprise Services Architecture
 - IBM Service-Oriented Modeling Architecture
 - ORACLE Fusion Architecture
- the **research** perspectives
 - mathematical foundations
 - DL, FOL, Frame-logic
 - ontology editors
 - Ontolingua, Protégé, OntoEdit)
 - ontology reasoners
 - FaCT++, RacerPro, KAON2, Pellet
- the **Semantic Web Services** technology
 - SA-WSDL, WSDL-S, OWL-S, WSMO

Criteria	Ontolingua	Protégé	OntoEdit
Clarity of interface	3	5	5
Interface Consistency	4	5	5
Meaning of Commands	2	4	4
Local installation	0	5	5
Updating speed	2	4	4
Help system	4	5	2
Ontology overview	2	5	5
Stability of the tool			
User support			
Features of free version			
Visualization			
Total			

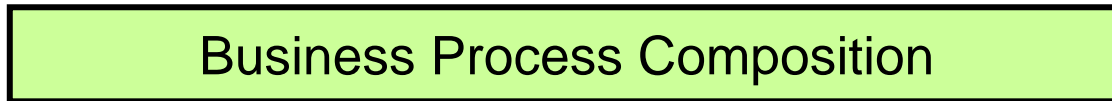
Criteria	Ontolingua	Protégé	OntoEdit
Multiple inheritance	4	5	0
Exhaustive decomposition	4	0	0
Disjoint decomposition	5	5	5
Example ontologies	3	5	0
Ontology Library	5	3	0
Java based	0	5	5
Database Backend	0	5	0
Total	21	28	10

System	Language	Source(s)	Interface(s)	Algorithm	Distribution
Cerebra	C++	OWL	OWL-API & WSDL	tableau	commercial
FaCT++	C++	OWL	DIG	tableau	open source
KAON2	Java	OWL & SWRL	DIG	resolution	free for non-commercial use
Pellet	Java	OWL	OWL-API, DIG, Jena	tableau	open source
RacerPro	LISP	OWL & SWRL	DIG	tableau	commercial

Implementation Level
 0 = Nil
 1 = Poor
 2 = OK
 3 = Good
 4 = Very Good
 5 = Excellent

We have come up with an “open” EAI Reference Model...

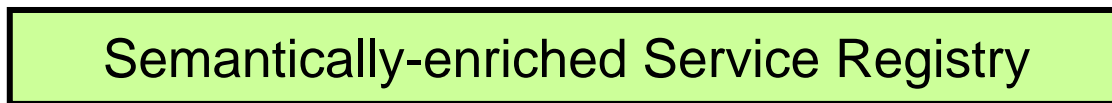
Process Composition



Generation of **abstract** and **executable** process models



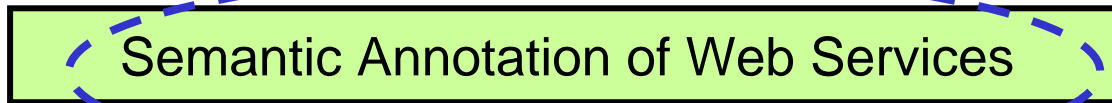
Semantic Registry



Publication of the semantic profiles on a semantically registry



Semantic Web Services



Creation of the semantic profiles (**SA-WSDL**) of the exposed services

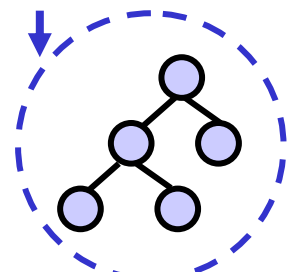
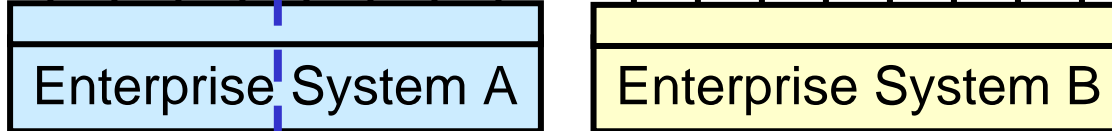


Enterprise Services



Precondition:
Exposed functionality utilizing WS technology

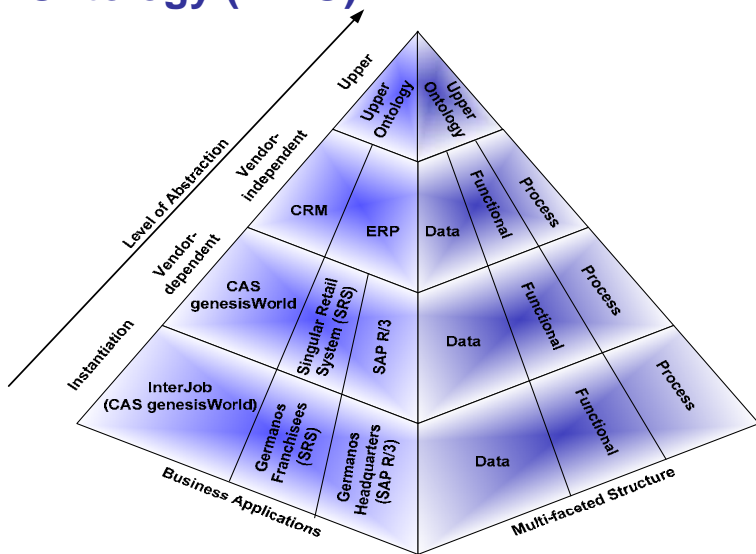
Enterprise Systems



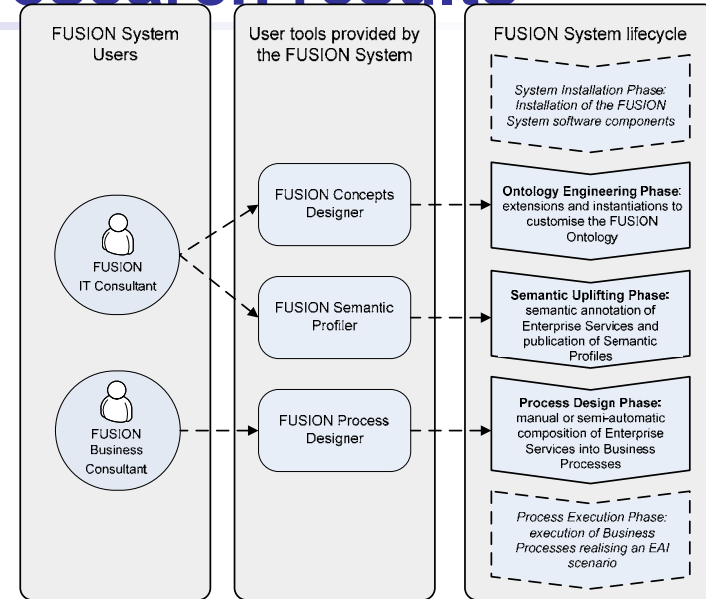
Ontological Reference Model:
Enterprise Interoperability Ontology (ENIO)

...and a set of FUSION innovative research results

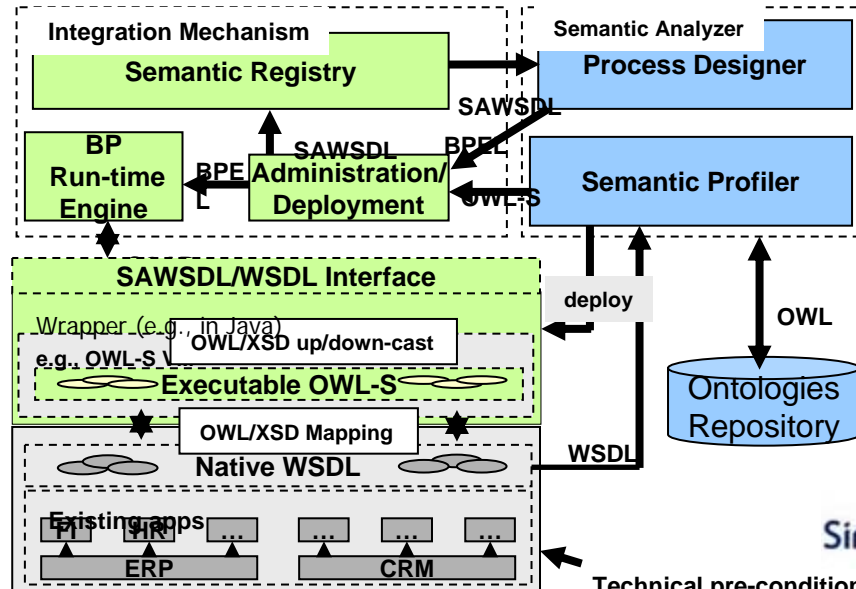
The FUSION Ontology
The Enterprise Interoperability
Ontology (ENIO)



The FUSION
Methodology

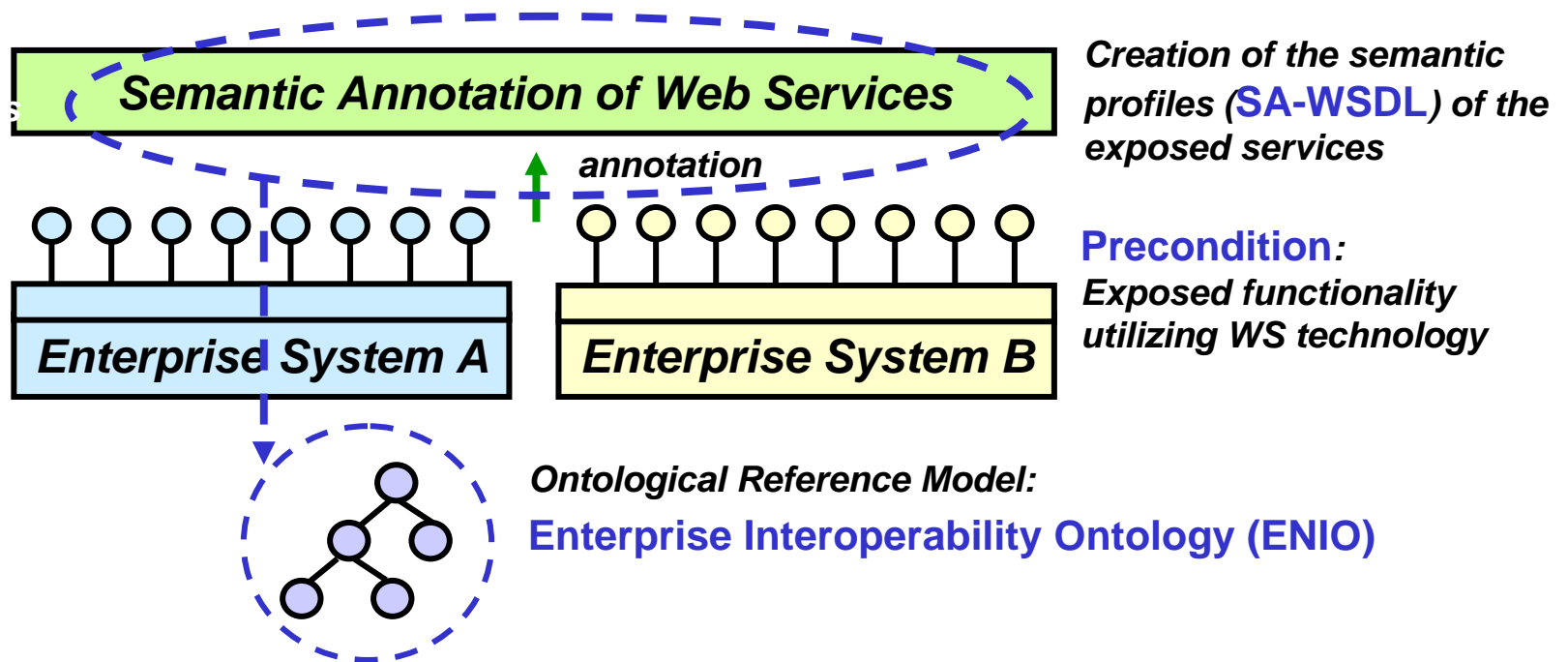


The FUSION
Toolset



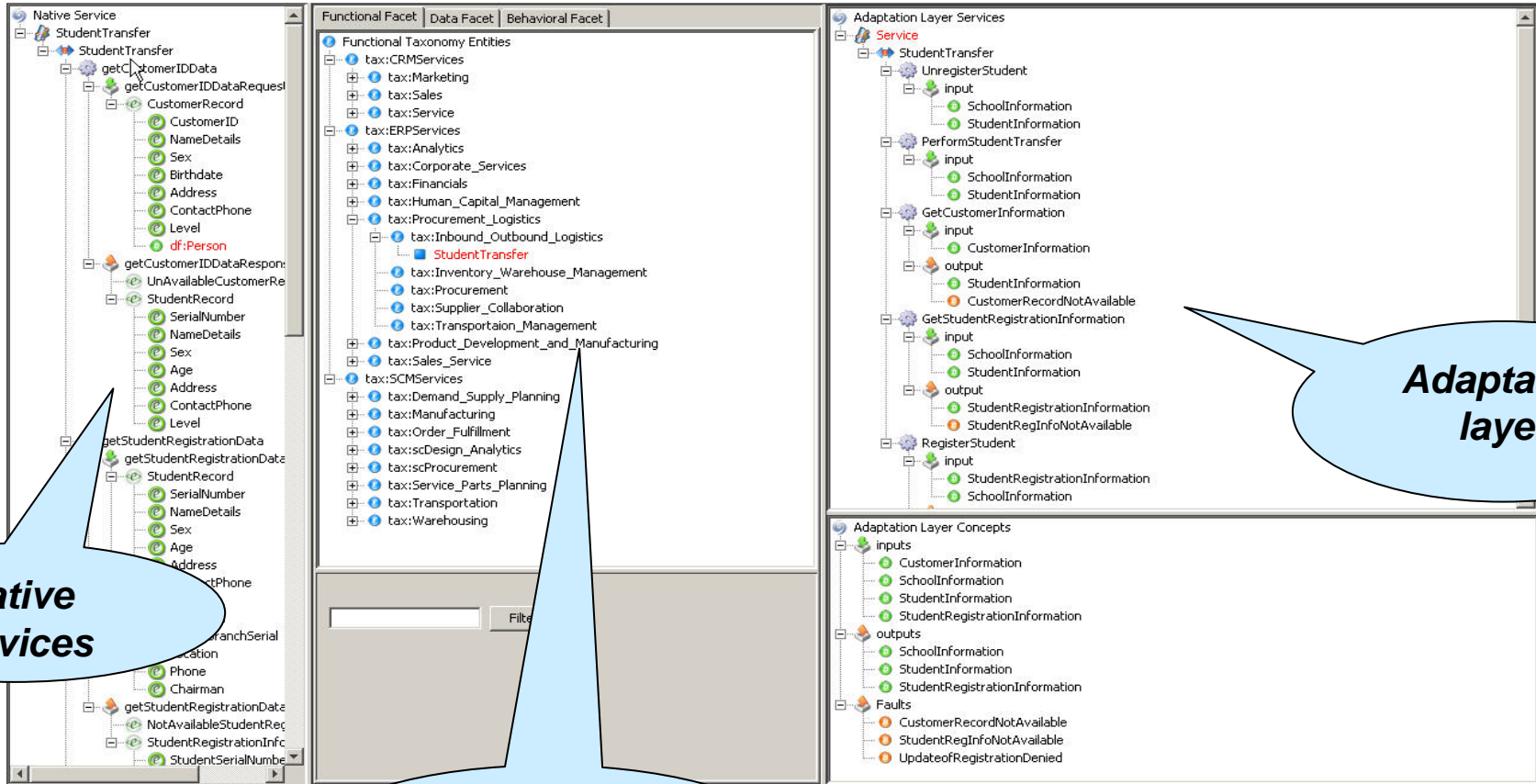
The SEAP Platform (1 of 3)

- SEAP (Semantic Annotation and Profiling) allows
 - selection and visualization of the native Web Services interfaces and the data-intensive enterprise interoperability ontology and
 - facilitates the user to annotate (through a drag and drop utility) the input and output message parts of the selected Web Service interface with business data entities of the ENIO ontology.



The SEAP Platform (2 of 3)

The SEAP platform is available at: <http://sourceforge.net/projects/fusionprofiler>

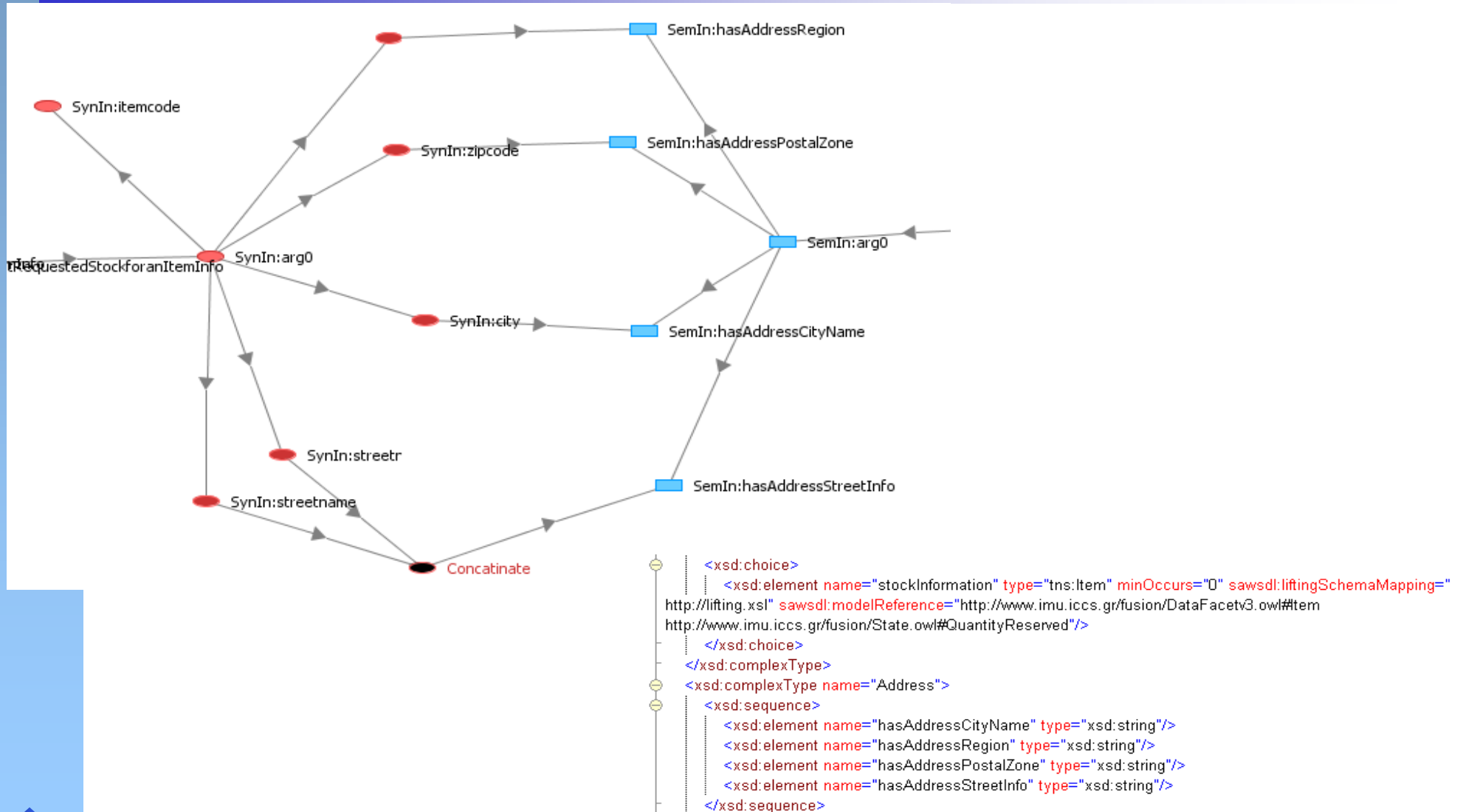


Native services

Ontology browser and filtering

Adaptation layer

The SEAP Platform (3 of 3)



```

<xsd:choice>
  <xsd:element name="stockInformation" type="tns:Item" minOccurs="0" sawsdl:liftingSchemaMapping="
http://lifting.xml" sawsdl:modelReference="http://www.imu.iccs.gr/fusion/DataFacetv3.owl#Item
http://www.imu.iccs.gr/fusion/State.owl#QuantityReserved"/>
</xsd:choice>
</xsd:complexType>
<xsd:complexType name="Address">
  <xsd:sequence>
    <xsd:element name="hasAddressCityName" type="xsd:string"/>
    <xsd:element name="hasAddressRegion" type="xsd:string"/>
    <xsd:element name="hasAddressPostalZone" type="xsd:string"/>
    <xsd:element name="hasAddressStreetInfo" type="xsd:string"/>
  </xsd:sequence>

```

Demo

EMPOWER Project

- And now the “FUSION inspired” EMPOWER: A semantic Service Oriented Private Adaptation Layer enabling the next generation, interoperable, and easy-to-integrate software products of European Software SMEs

An innovative framework and the enabling technologies

Result

that will allow the IT SMEs

Beneficiaries

to

create their next generation loosely coupled,
interoperable and easy to integrate **commercial-off-**
the self software products

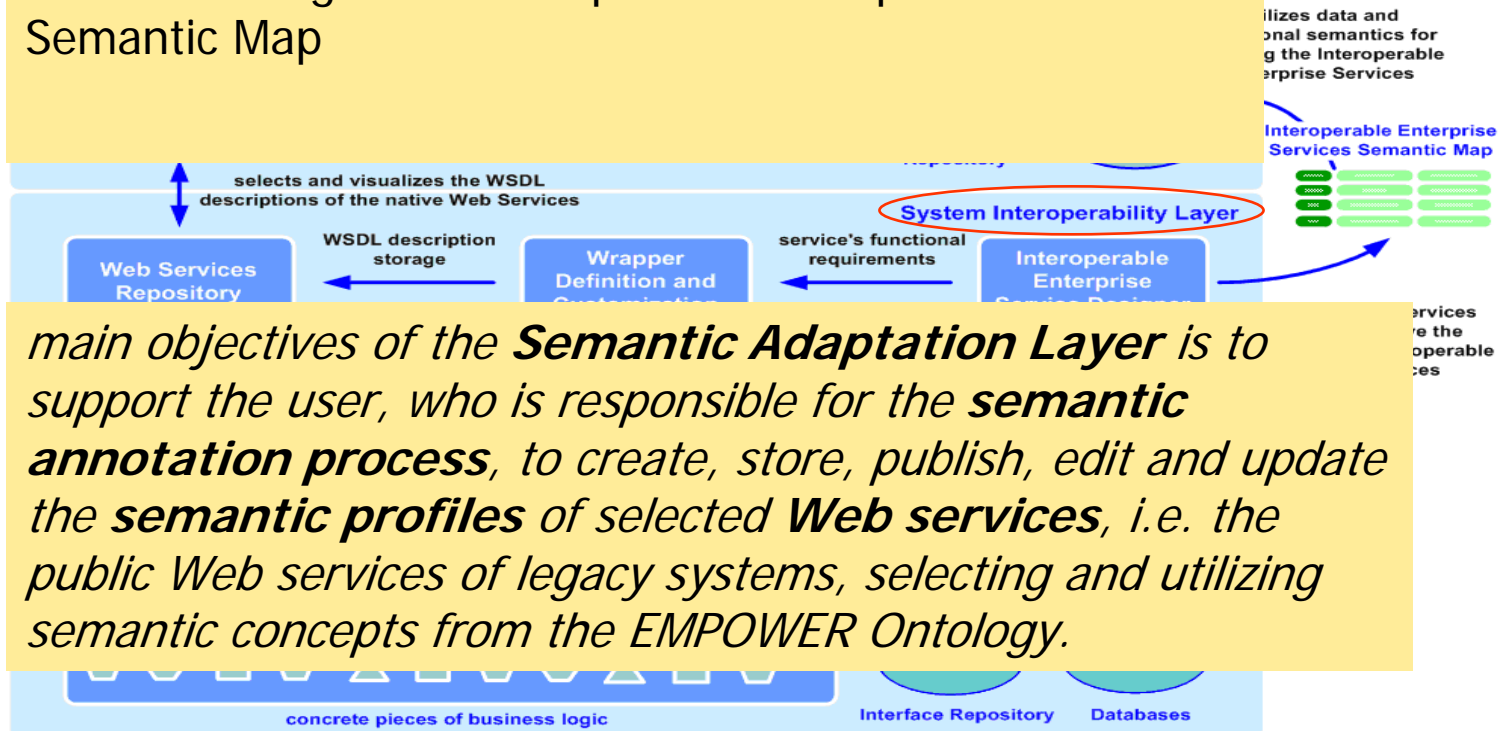
To do what

leveraging the quality of the application software and the integration services delivered to customers

Will be applied in 5 different systems of 5 SMEs

Main Architecture

The **System Interoperability Layer** addresses the underlying **technology requirements** of the proposed EMPOWER Framework, i.e. the ad-hoc definition and extraction (as Web Services) of specific pieces of system's functionality, making use of the granularity defined through the Interoperable Enterprise Services Semantic Map



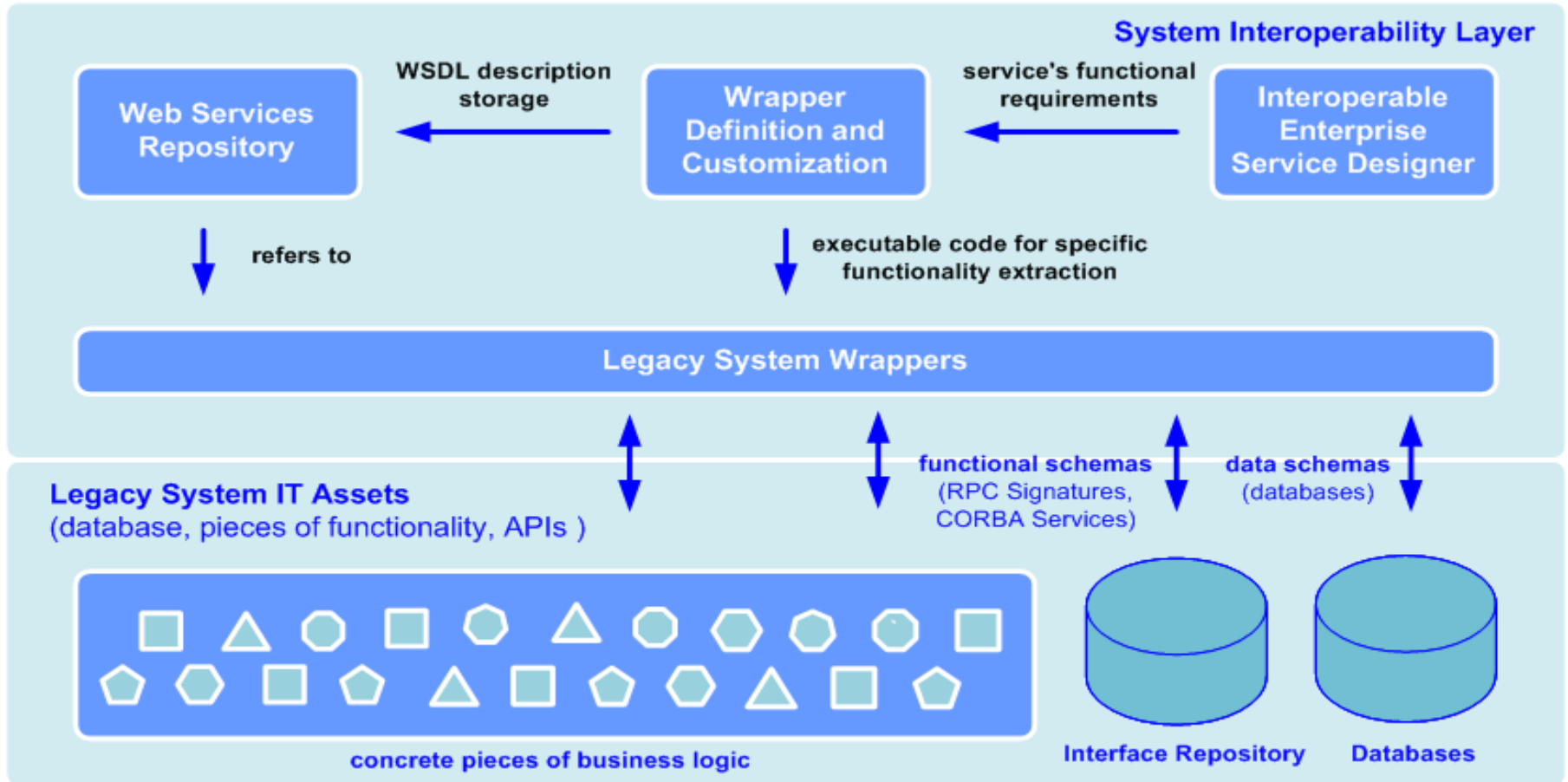
*main objectives of the **Semantic Adaptation Layer** is to support the user, who is responsible for the **semantic annotation process**, to create, store, publish, edit and update the **semantic profiles** of selected **Web services**, i.e. the public Web services of legacy systems, selecting and utilizing semantic concepts from the EMPOWER Ontology.*

System Interoperability Layer at a Glance

- facilitates the **service-oriented adaptation** of proprietary software products,
- realizes the **SOA and Web Services –enablement** of the application software products
- supports the **extraction** of custom **pieces** of application software product **functionality**
 - addressing the level of the “**Interoperable Enterprise Services**”
- utilizes **Web Services technologies** on top of **SQL-based database access, XML-RPC calls** and/or **file access** utilities

*With the term “**Interoperable Enterprise Services**”, we define “**services** exposed from enterprise applications, **realizing a specific piece of functionality**, with **typical Web Service interfaces** containing visible or not, **interior, complex behavioural process model** encapsulating a set of distributed business logic in a **single point of interaction**”*

Semantic Interoperability Layer Architecture



The System Interoperability Layer Components (1/2)

■ Interoperable Enterprise Service Designer

- allows the user-friendly, graphical **and ontology-assisted design of pre-selected pieces of functionality** of a given application software product
- defines the desired **service's functionality**, as well as the **input and output messages**,
 - utilizing entities and relations documented in the conceptual models (i.e. EMPOWER Ontology and Enterprise Service Semantic Map
 - so as to follow the level of granularity identified by the Interoperable Enterprise Services
- **generates automatically** the respective **WSDL description (interface)** of the designed service

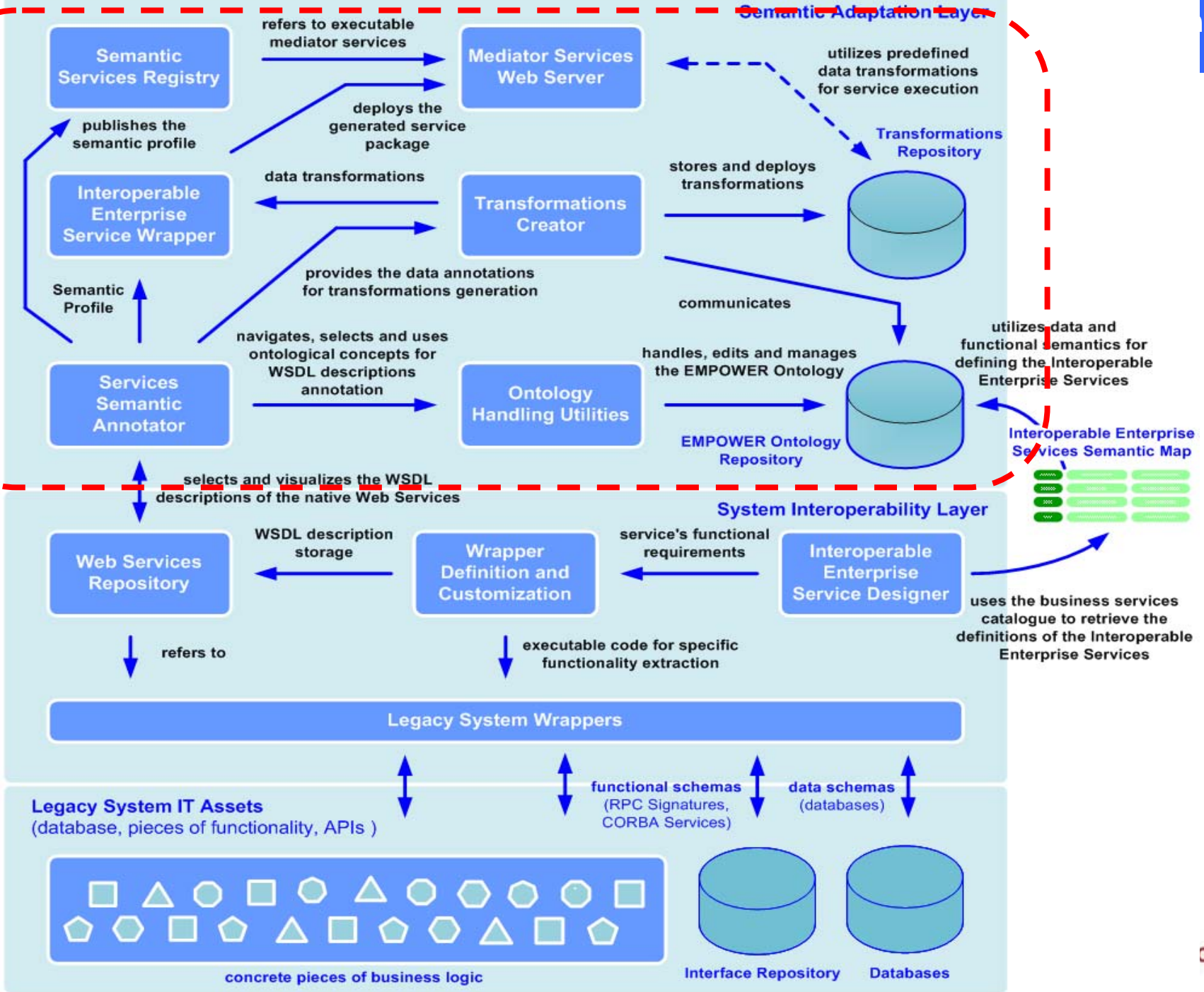
The System Interoperability Layer Components (2/2)

■ Wrapper Definition and Customization

- enables the **low-level implementation** of the designed Web Service
- **addresses the design requirements** specified in the Interoperable Enterprise Service Designer
- supports the **development and testing** of the respective **service wrapper**
 - encapsulating specific **functional** (e.g. RPC signatures and COBOL services) and **data** (e.g. databases) **schemes** of the given legacy system

■ Web Services Repository

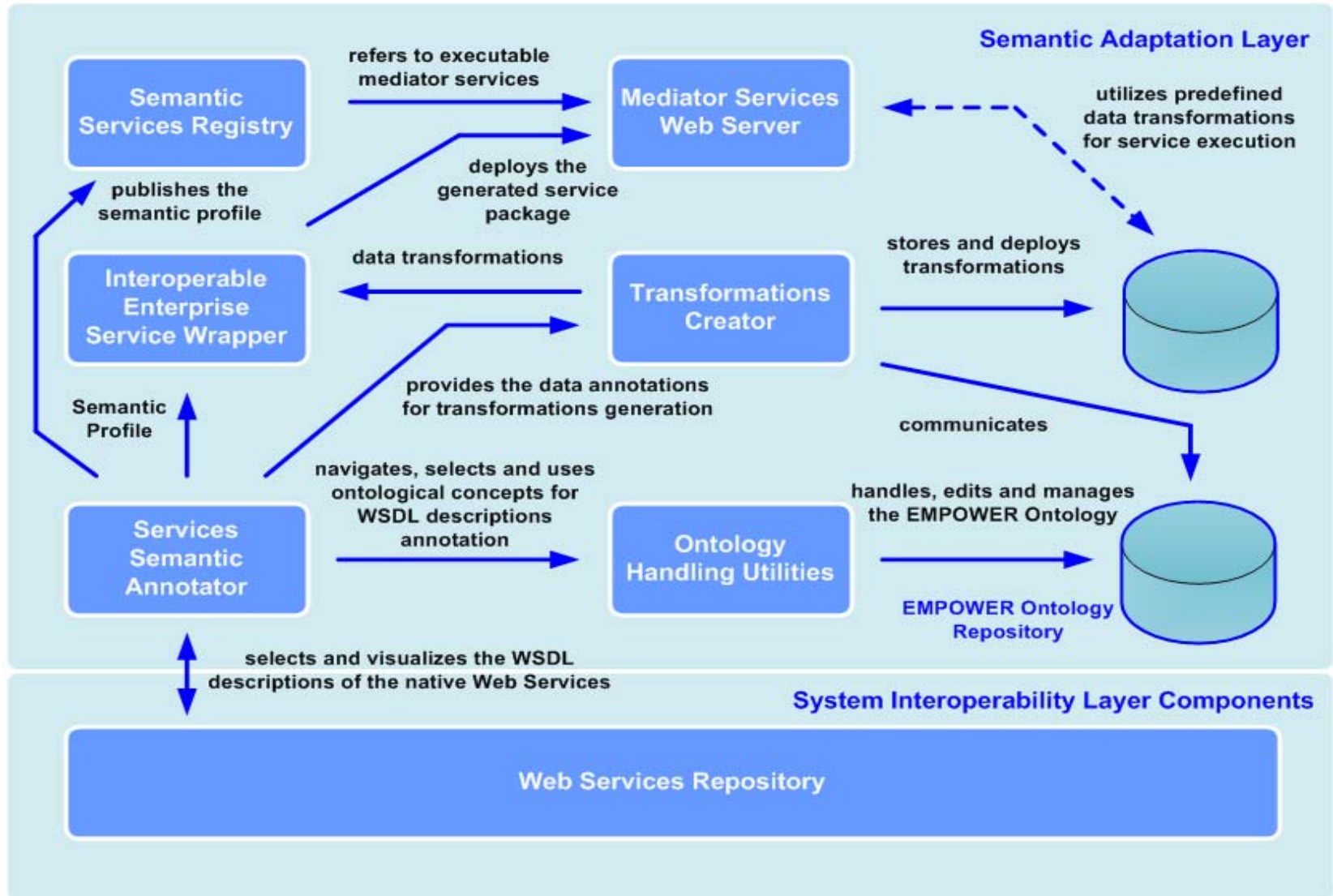
- provides the **storage area** for the created, native **Web Services' interfaces**, referring to the **software instances** of the **deployed executable services' wrappers**



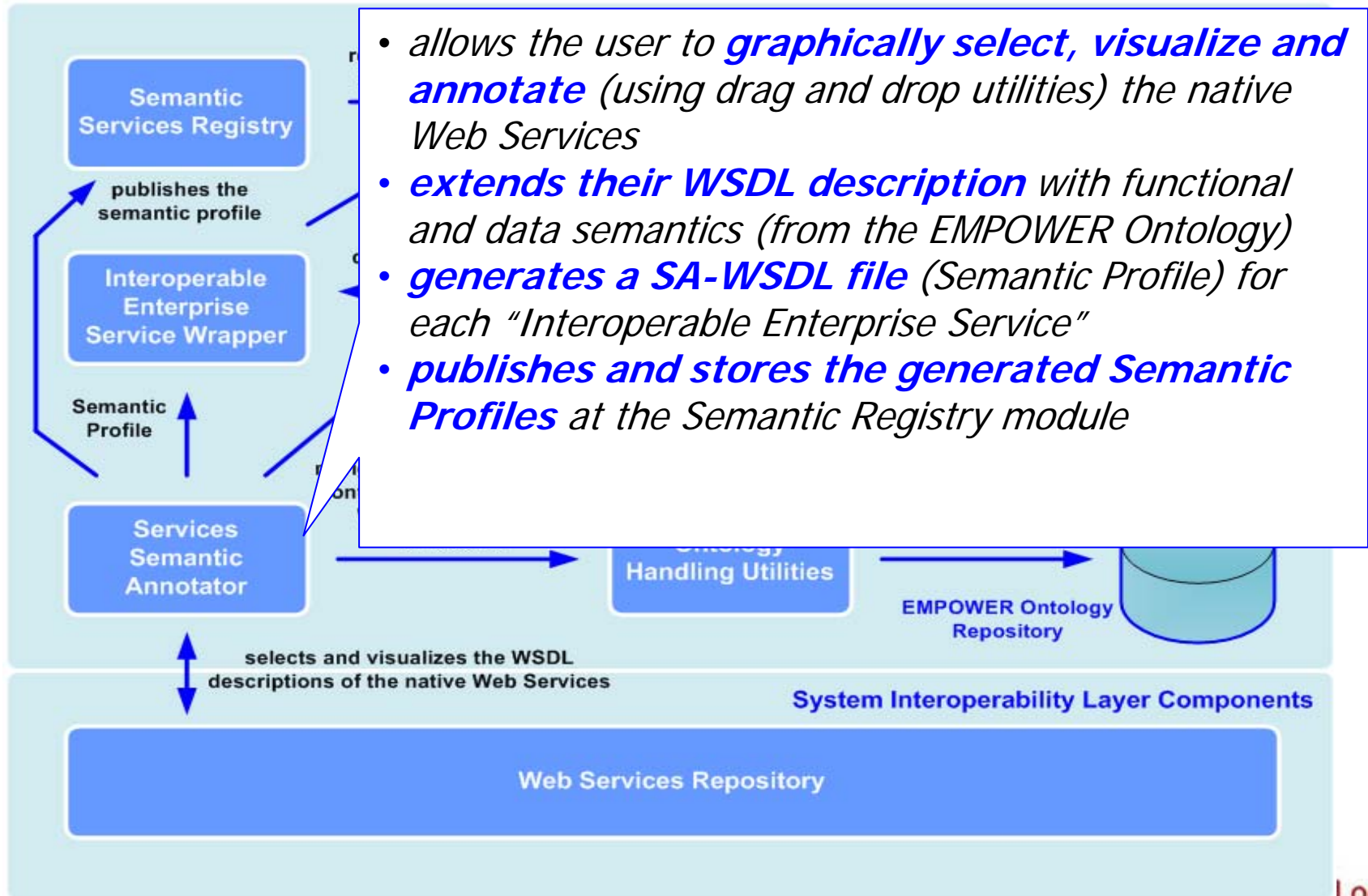
In the context of the EMPOWER Approach, SAL supports

- the **creation of the Semantic Profiles** of the exposed Interoperable Enterprise Services (semantic uplifting), involving
 - the **semantic annotation of their operation functionality**
 - by selecting and importing functional semantics of the EMPOWER Ontology
 - the **semantic description of their input and output elements**
 - by utilizing data / information semantics of the EMPOWER Ontology
 - the **definition of valid transformations** among the input and output messages of the Enterprise Services and the EMPOWER Ontology
 - facilitating the exchange of ontological individuals between the Interoperable Enterprise Services
 - enabling dynamic data mediation among the collaborative services
- the publication and storage of the developed Semantic Profiles in a **semantically-enriched business services registry**
 - supporting **ontology-assisted categorization, search and discovery** of the registered (grounded) Semantic Web Services based on functional semantics

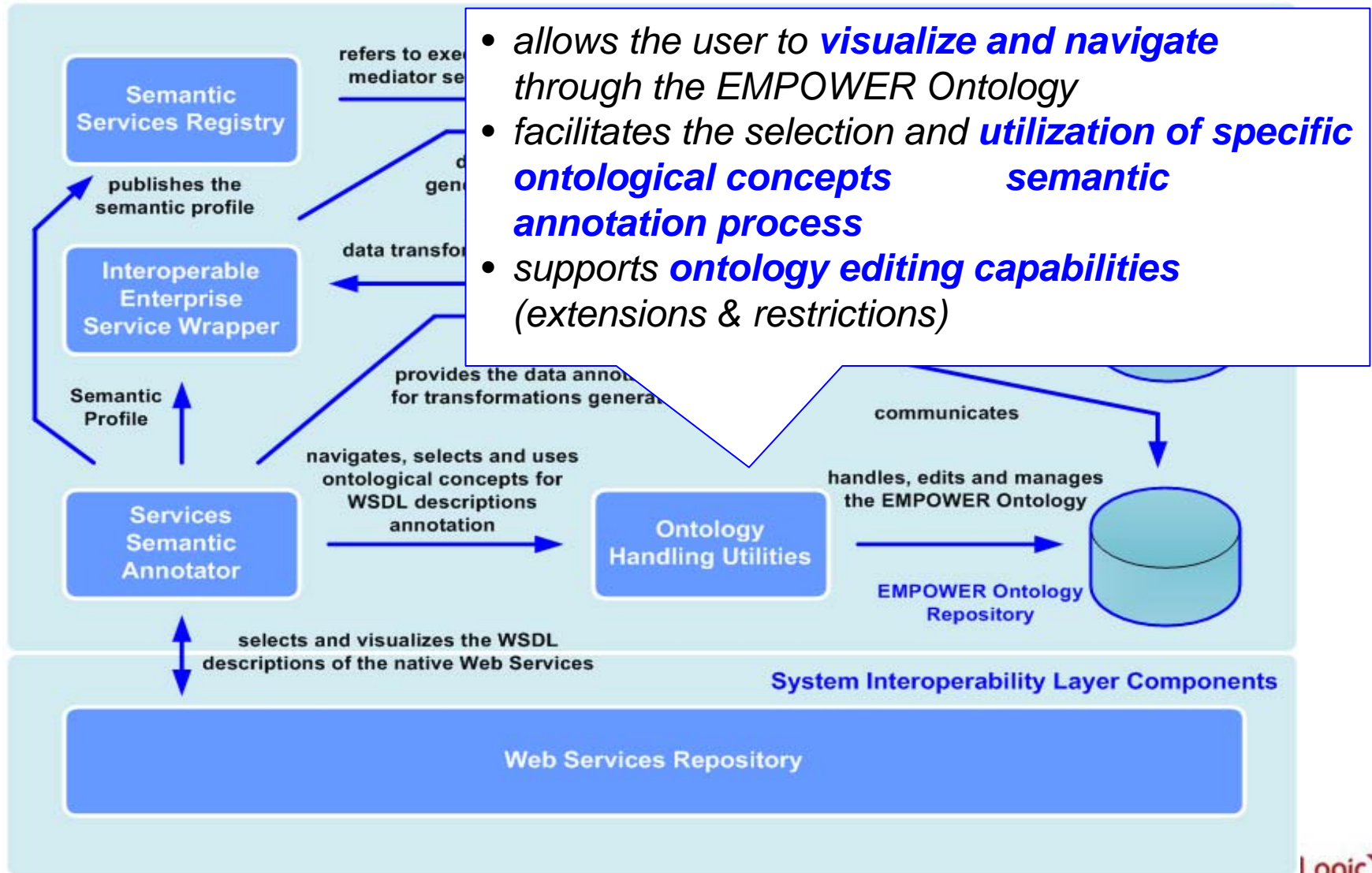
Semantic Adaptation Layer Conceptual Architecture



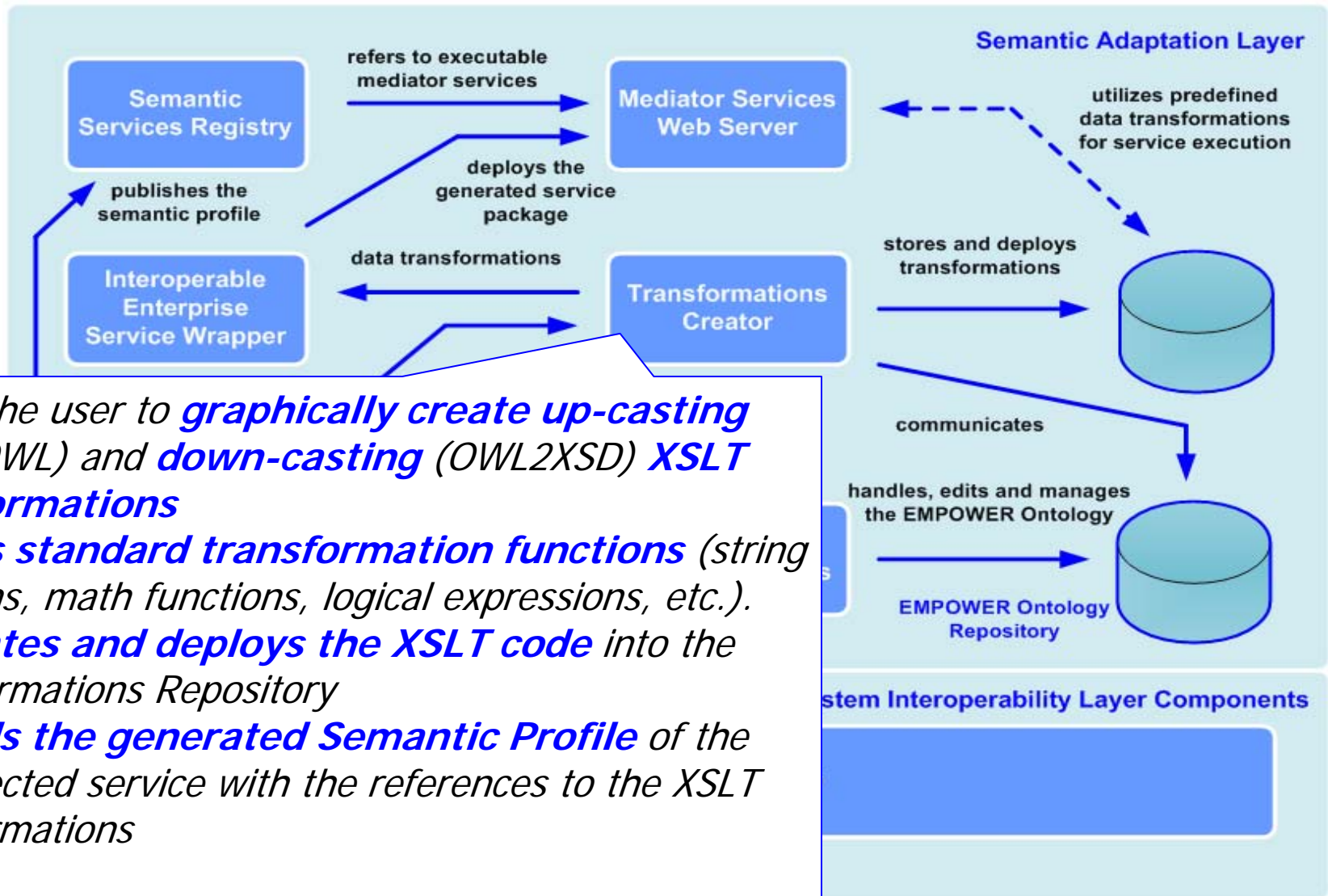
SAL: Services Semantic Annotator



SAL: Ontology Handling Utilities

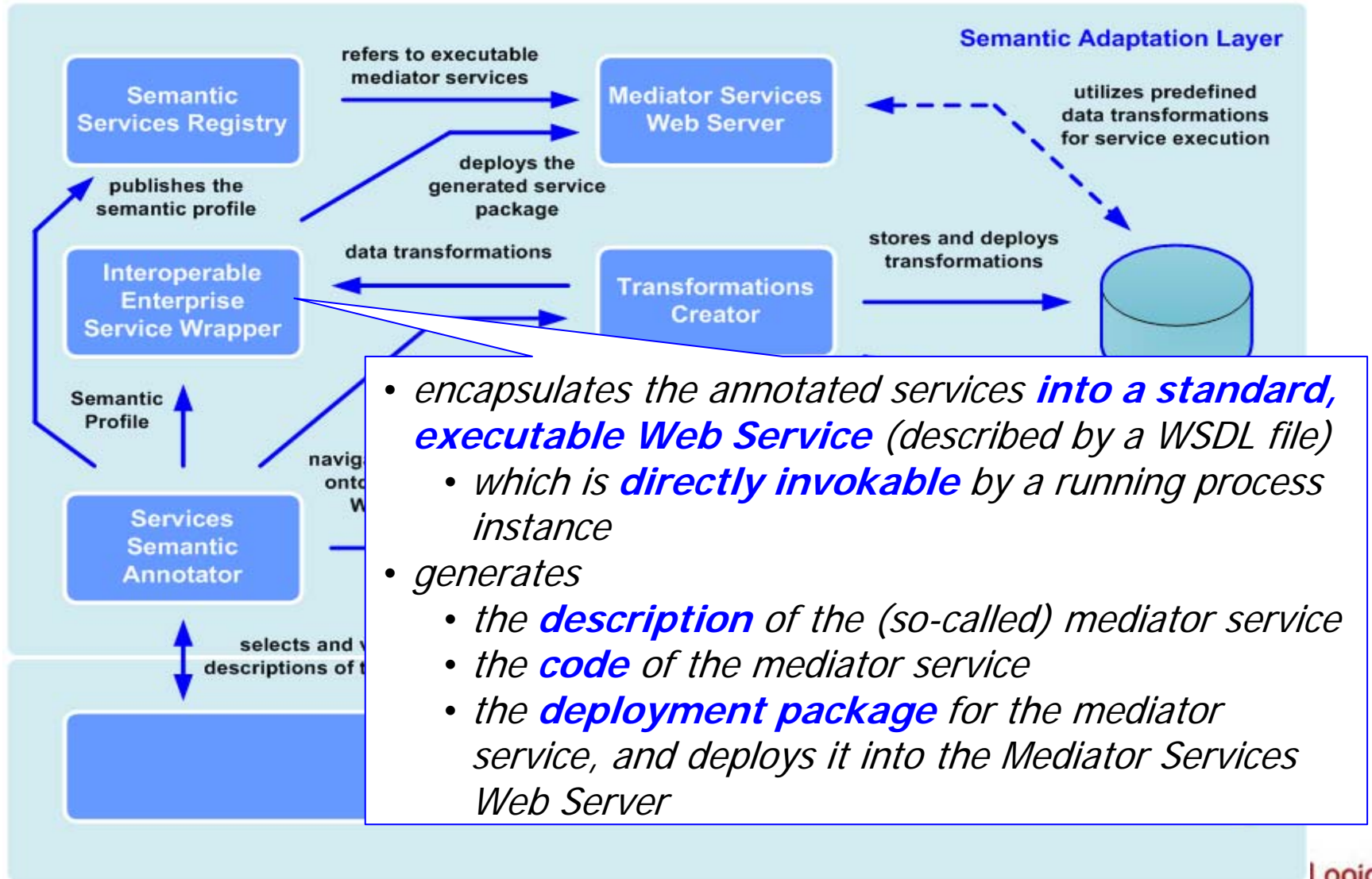


SAL: Transformations Creator

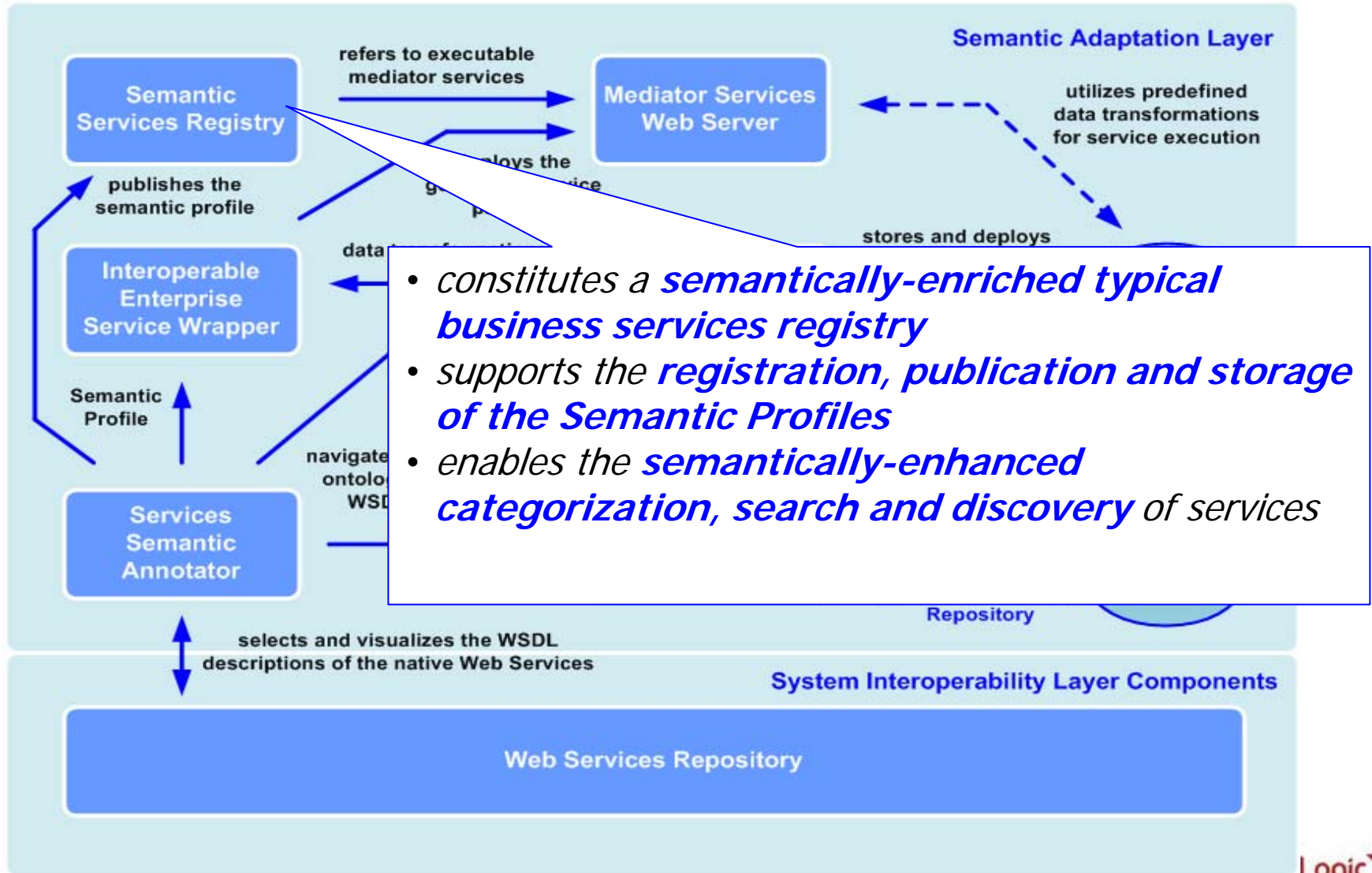


- allows the user to **graphically create up-casting (XSD2OWL) and down-casting (OWL2XSD) XSLT transformations**
- **utilises standard transformation functions** (string functions, math functions, logical expressions, etc.).
- **generates and deploys the XSLT code** into the Transformations Repository
- **extends the generated Semantic Profile** of the pre-selected service with the references to the XSLT transformations

SAL: Interoperable Enterprise Service Wrapper



SAL: Semantic Services Registry



Thank you