Usługi sieciowe (Web Services)

Karol Kański

Seminarium “Systemy Rozproszone”

14 października 2010

Agenda

1. Idea i historia usług sieciowych
2. Różne podejścia do tworzenia usług sieciowych
3. Języki opisu usług sieciowych
4. Rejestry usług sieciowych
5. Porównanie najpopularniejszych podejść
6. Szyny korporacyjne
7. Języki modelowania procesów biznesowych
8. Narzędzia i biblioteki
9. The Mobility Project

Motywacja i idea

Główny cel:

- Integracja aplikacji i elektroniczna wymiana danych (Electronic Data Interchange - EDI)

Droga prowadząca do osiągnięcia go:

- Poprzez użycie najbardziej rozpowszechnionych i akceptowanych technologii i standardów (sieć WWW, protokół HTTP, język XML)

Definicje usługi sieciowej

Definicja W3C (ze specyfikacji Web Services Architecture)
A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Definicja z polskiej Wikipedii
Komponent programowy niezależny od platformy i implementacji, dostarczający określonej funkcjonalności.

Definicja z angielskiej Wikipedii
Typically application programming interfaces (API) or Web APIs that are accessed via Hypertext Transfer Protocol (HTTP) and executed on a remote system hosting the requested services.

Historia

- SOAP (Simple Object Access Protocol) – 1998
- REST (Representational State Transfer) – 2000
- UDDI (Universal Description Discovery and Integration) – 2000
- WSDL (Web Services Description Language) – 2001
- Cała reszta specyfikacji spod znaku WS-* – w zasadzie od 1998 roku ciągle pojawia się coś nowego

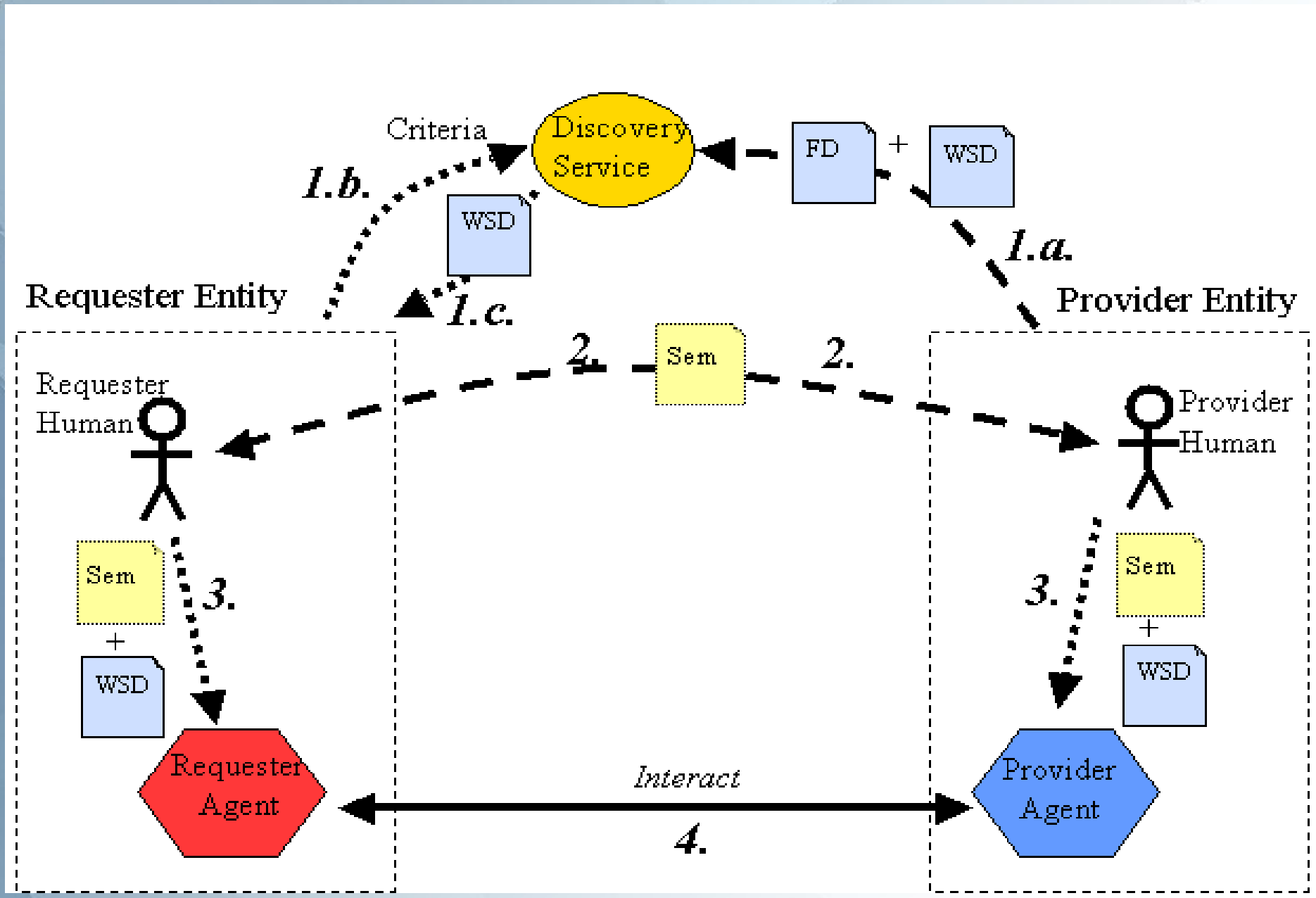
Ogólny schemat postępowania

Dostawca usługi:

- Tworzy usługę
- Definiuje ją za pomocą języka opisu usług
- Publikuje ją w rejestrze usług

Klient:

- Wyszukuje usługę w rejestrze
- Pobiera definicję usługi
- Używa usługi



Źródło: www.w3.org

Tworzenie WS - podejścia

Główne trendy w rozwoju usług sieciowych to dziś:

- Big Web Services (zorientowane na usługi)
- RESTful Web Services (zorientowane na zasoby)
- Hybrydy powyższych

Tworzenie WS - podejścia

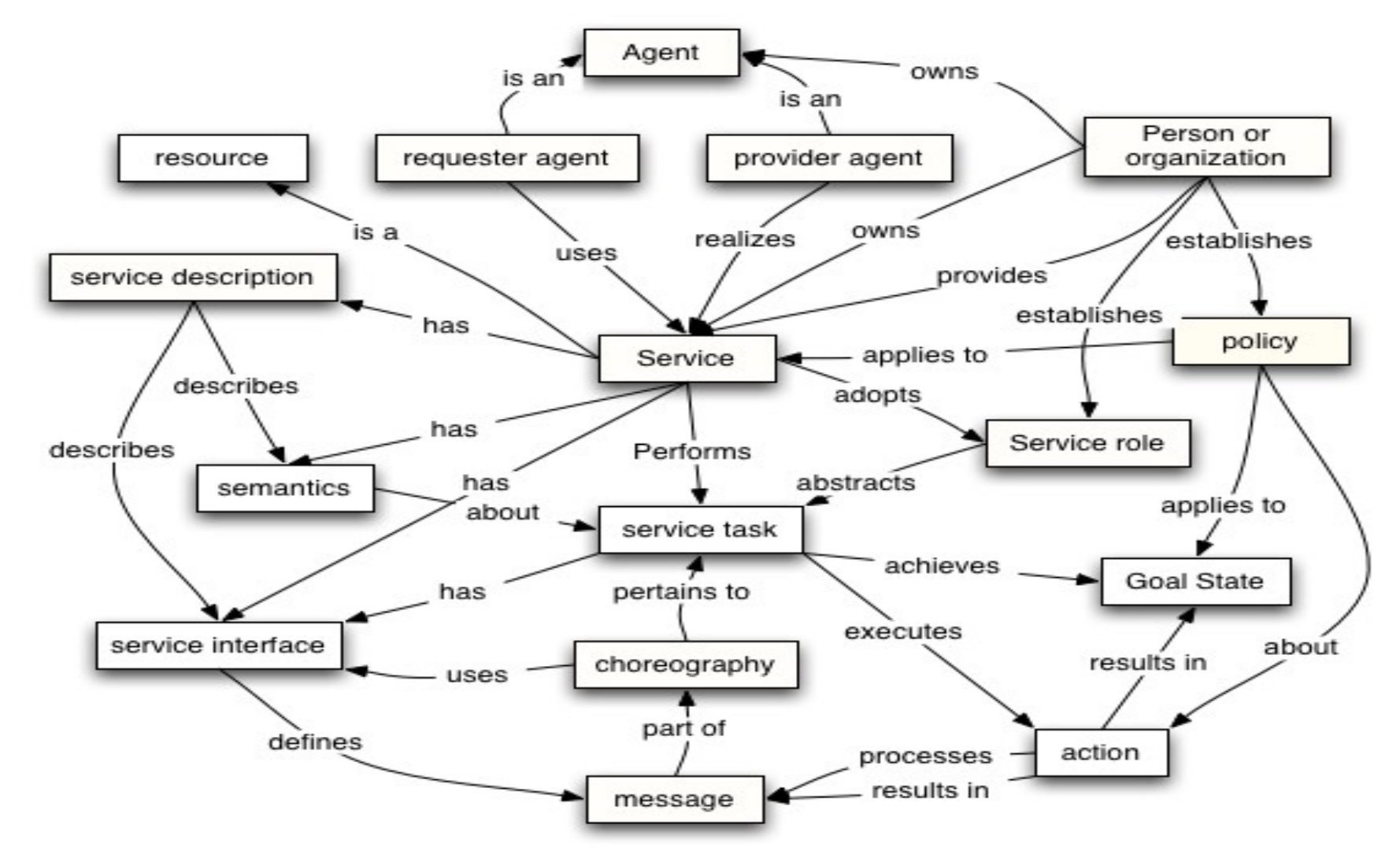
REST:

- Amazon
- eBay
- Yahoo
- Youtube
- Facebook

Big Web Services:

- Amazon
- eBay
- Google

SOA

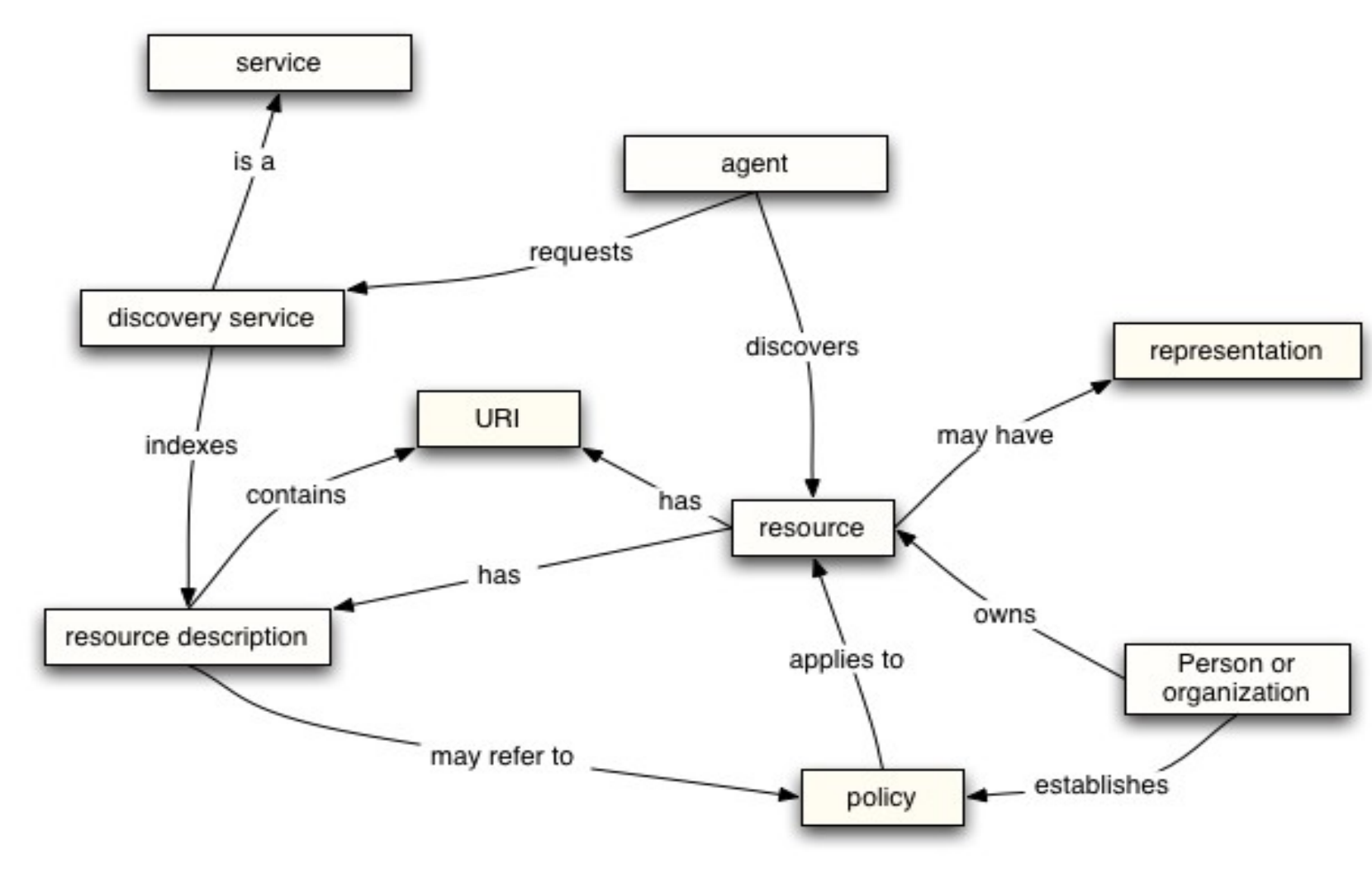


Źródło: www.w3.org

SOA

- Podejście zorientowane na usługi
- Związane z Big Web Services
- Język opisu usług – WSDL
- Rejestr usług – UDDI
- Wywołanie usługi – HTTP GET + protokół SOAP
- Zwrócenie wyników – odpowiedź na HTTP GET + protokół SOAP

ROA



Źródło: www.w3.org

ROA

- Podejście zorientowane na zasoby
- Związane z stylem REST
- Pobieranie zasobów
- Język opisu usług – WADL lub WSDL 2.0
- Rejestr usług – w stylu REST (deskryptory jako zasoby) ale da się też UDDI
- Wywołanie usługi – metody HTTP
- Zwrócenie wyników – odpowiedź na HTTP

Przesyłanie danych - BWS

- Wywołanie usług odbywa się za pomocą komunikatu HTTP z metodą GET
- Do przesyłania danych używa się protokołu SOAP

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
...
</soap:Header>

<soap:Body>
...
  <soap:Fault>
  ...
  </soap:Fault>
</soap:Body>

</soap:Envelope>
```

SOAP

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.mimuw.edu.pl/students">
  <m:GetStudent>
    <m:StudentID>10</m:StudentID>
  </m:GetStudent>
</soap:Body>

</soap:Envelope>
```

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.mimuw.edu.pl/students">
  <m:GetStudentResponse>
    <m:Student>
      <m:FirstName>Karol</m:FirstName>
      <m:LastName>Kański</m:LastName>
    </m:Student>
  </m:GetStudentResponse>
</soap:Body>

</soap:Envelope>
```

Przesyłanie danych - REST

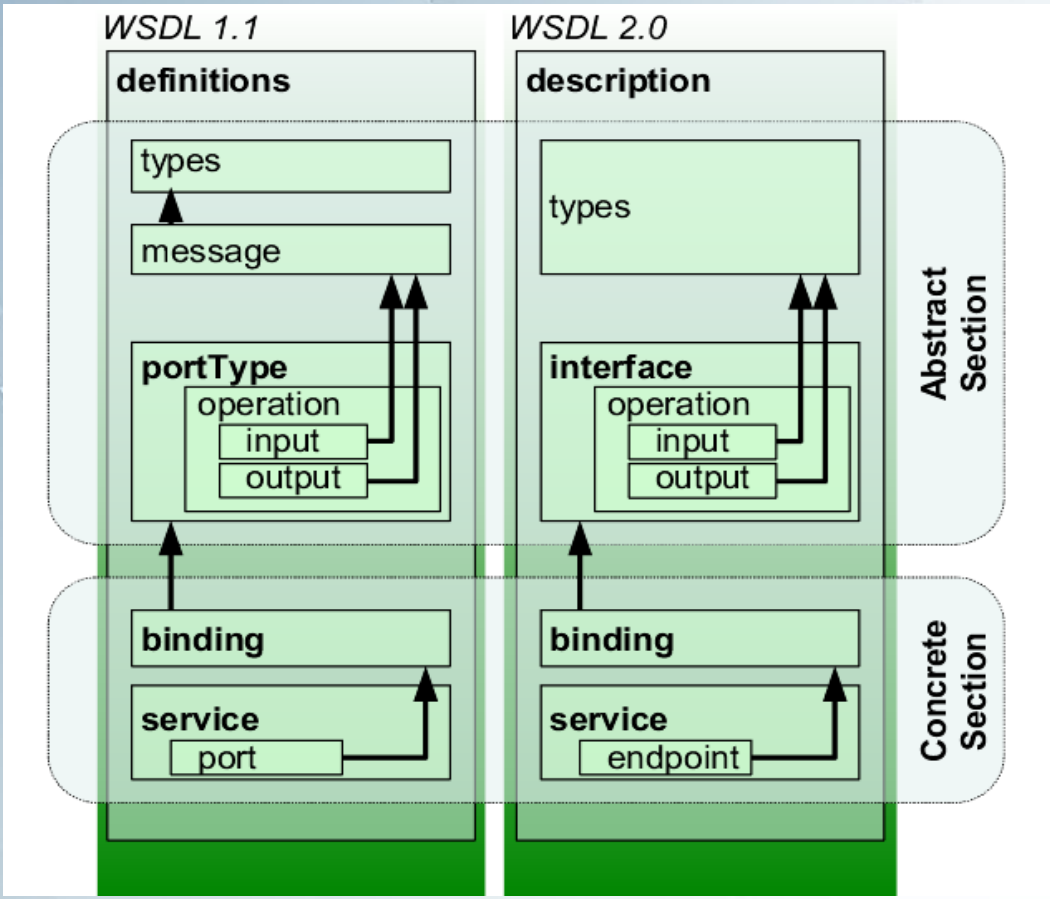
- Wywołanie usługi (czyli w tym wypadku pobranie zasobu) odbywa się za pomocą komunikatów HTTP, przy czym różne metody mają różne znaczenie.
- Parametry w URL
- Wynik to najczęściej HTML, XML lub JSON, ale tak naprawdę może nim być cokolwiek

REST

Zasób	URI kolekcji, np. http://example.com/resources/	URI elementu, np. http://example.com/resources/142
GET	Lista URI członków kolekcji. Ewentualnie inne informacje o członkach kolekcji.	Reprezentacja danego elementu.
PUT	Zamienia całą kolekcję na inną.	Aktualizuje dany element kolekcji. Tworzy go jeśli dany element nie istniał.
POST	Tworzy nowy wpis w kolekcji.	Traktuje element jako kolekcje i tworzy w niej nowy wpis.
DELETE	Usuwa całą kolekcję.	Usuwa dany element kolekcji.

Opis usług - BWS

- Usługi opisywane w języku WSDL (dwie wersje: 1.1 – popularniejsza, 2.0)



Źródło: en.wikipedia.org

WSDL

```
<definitions>
  <types/>
  <message name="errorMessage">
    <part name="parameters" element="tns:error"/>
  </message>
  <message name="getStudentRequest">
    <part name="parameters" element="xs:integer"/>
  </message>
  <message name="getStudentResponse">
    <part name="parameters" element="tns:Student"/>
  </message>
  <portType name="StudentData">
    <operation name="getStudent">
      <input name="getStudentRequest" message="getStudentRequest"/>
      <output name="getStudentResponse" message="getStudentResponse"/>
      <fault name="fault" message="errorMessage"/>
    </operation>
  </portType>
  <binding name="StudentDataBinding" type="StudentData">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getStudent">
      <input><soap:body use="literal"/></input>
      <output><soap:body use="literal"/></output>
      <fault><soap:fault name="fault" use="literal"/></fault>
    </operation>
  </binding>
  <service name="StudentDataService">
    <port name="StudentData" binding="StudentData">
      <soap:address location="http://www.mimuw.edu.pl:8080/StudentDataService/StudentDataPort"/>
    </port>
  </service>
</definitions>
```

Opis usług – REST

- Językiem opisu usług jest WADL, ale może też być nim WSDL 2.0

```
<application xmlns="http://research.sun.com/wadl/2006/07">
 <grammars>
  <include href="StudentData.xsd"/>
  <include href="SError.xsd"/>
 </grammars>
 <resources base="http://www.mimuw.edu.pl:8080/StudentDataService/">
  <resource path=”students”>
   <method href="#getStudent"/>
  </resource>
 </resources>
 <method id="getStudent" name="GET">
  <request>
   <query_variable name=”id” type=”xsd:int” required=”true”/>
  </request>
  <response>
   <representation mediaType="application/xml" element="StudentData"/>
   <fault id="GetStudentError" status="400" mediaType="application/xml" element="Error"/>
  </response>
 </method>
</application>
```

Znajdowanie usług - BWS

Dla BWS rejestrem usług jest rejestr UDDI, który składa się z trzech komponentów:

- White Pages – dane kontaktowe dostawcy usługi (nazwa firmy, adres itp.)
- Green Pages – semantyka i techniczne informacje o usłudze (WSDL)
- Yellow Pages – przemysłowa kategoryzacja usługi

Znajdowanie usług - REST

- Można wykorzystać rejestr UDDI
- Częściej spotykanym pomysłem jest stworzenie rejestru w stylu REST
- Zasobami są wtedy deskryptory usług

REST vs. Big Web Services

Big Web Services	REST
• Sprawdzanie typów • Duże wsparcie ze strony bibliotek i narzędzi • Standard • Wbudowana obsługa błędów	• Lekki • Wyniki czytelne dla człowieka • Łatwość tworzenia • Brak standardu • Specyficzne wykorzystanie metod HTTP

Szyny korporacyjne (ESB)

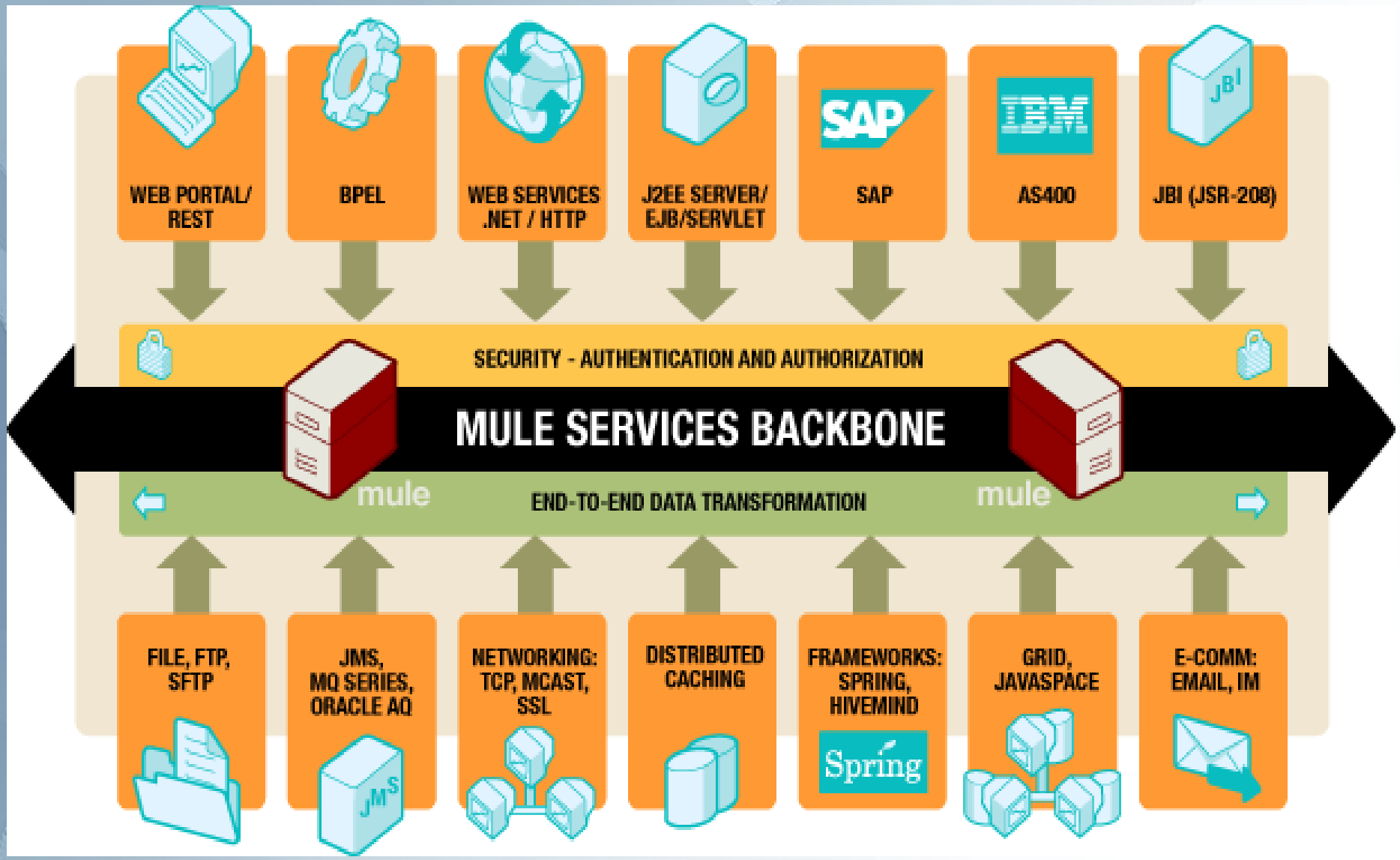
Problem:

- zintegrować systemy używające różnych protokołów, formatów danych itp.

Rozwiązania:

- Enterprise Application Integration (EAI)
- Enterprise Service Bus (ESB)

ESB



Źródło: http://www.mulesoft.org

Obszary działania ESB

- Invocation
- Routing
- Mediation
- Messaging
- Process choreography
- Service orchestration
- Complex event processing
- Other quality of service
- Management

Języki modelowanie procesów biznesowych

Służą do opisu procesów biznesowych w terminach wywołań usług sieciowych. Do najpopularniejszych należą:

- Business Process Execution Language (BPEL)
- Business Process Modeling Language (BPML) – nadzbiór BPEL

BPEL

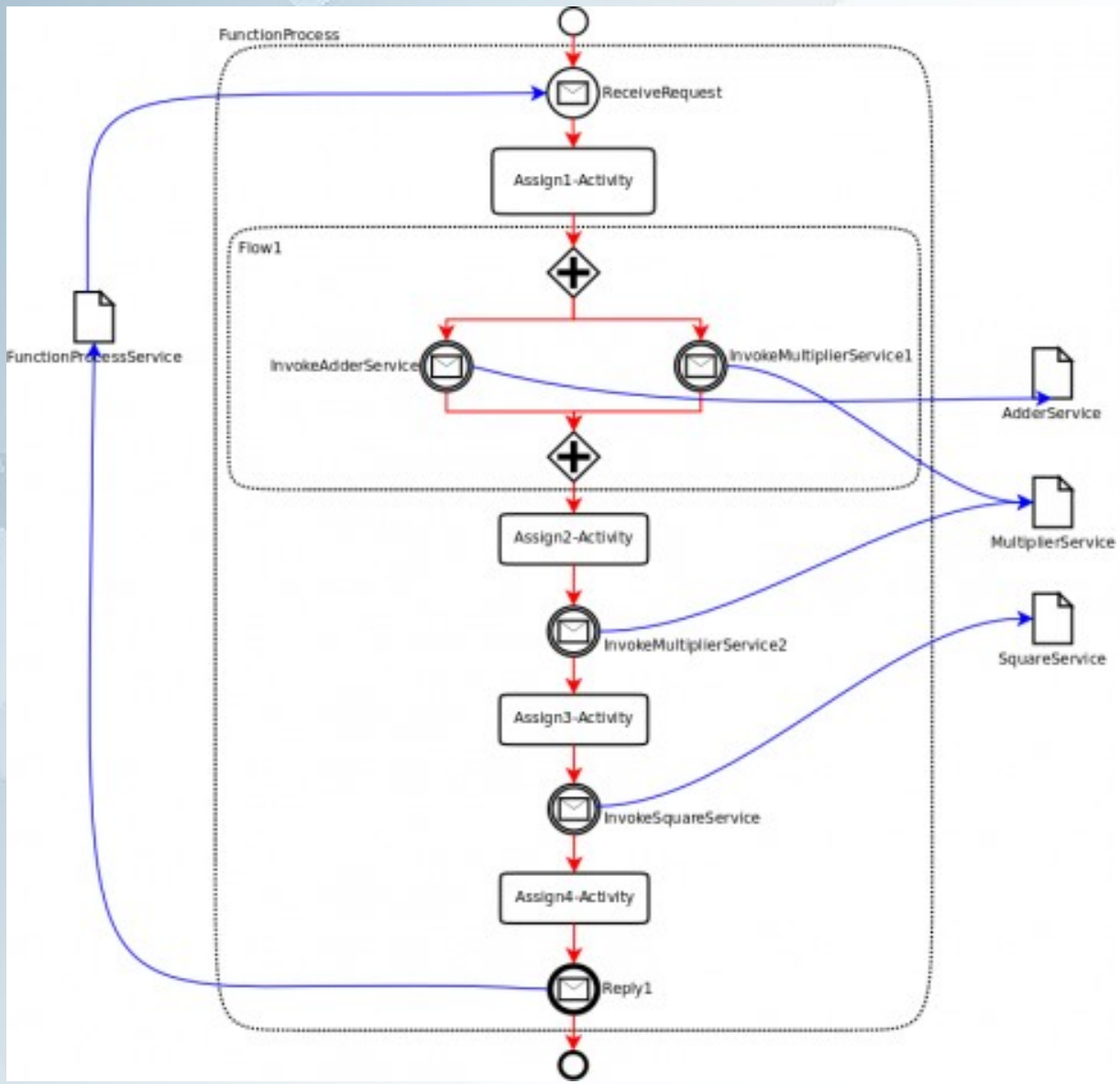
- Powstał z języków: Web Services Flow Language (IBM) i XLANG (Microsoft)
- Odpowiedzialny za warstwę abstrakcji i wykonania
- Brak możliwości edycji graficznej, ale do tego można wykorzystać Business Process Modeling Notation (BPMN)

BPEL - znaczniki

- <invoke>
- <receive>
- <reply>
- <assign>
- <throw>
- <wait>
- <terminate>
- <sequence>
- <flow>
- <switch>
- <while>
- <pick>
- <variable>
- <partnerLink>

Workflow

$$f(x,y) = [(x+y)^{(2)} * (x*y)^{(1)}]^{(3)}$$



Źródło: http://wso2.org/

Biblioteki i narzędzia

BWS:

- Java
- .NET

REST:

- Java
- Python
- Ruby
- .NET

Biblioteki i narzędzia - Java

BWS:

- Apache CXF
- Java Web Services Development Pack

REST:

- restlet
- JSR 311

The Mobility Project

- Oprogramowanie do obsługi wymian studenckich
- Wymiana danych pomiędzy instytucjami
- W przyszłości obsługa dużej liczby instytucji
- Sensowne rozwiązanie: wykorzystanie usług sieciowych
- Stan obecny: implementacja za pomocą BWS
- Mój cel: implementacja REST

Bibliografia

- http://www.w3.org
- http://www.w3schools.com/
- http://pl.wikipedia.org
- http://en.wikipedia.org
- http://wso2.org
- http://wazniak.mimuw.edu.pl ;)
- http://www.ics.uci.edu/~fielding/pubs/dissertation