



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI



**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



**Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego**

# Rozproszone systemy internetowe

## Wprowadzenie do usług WWW (Web Services)

**„Podniesienie potencjału uczelni wyższych jako czynnik rozwoju gospodarki opartej na wiedzy”**

Nr umowy: UDA-POKL.04.01.01-00-143/09-00

# Definicje

Dokument Web Services Architecture :

<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>

Usługa WWW (Web Service) to system:

- realizujący pewną funkcjonalność, z której można skorzystać poprzez automatyczną interakcję w sieci (maszyna-maszyna)
- posiadający interfejs opisany w języku, który automaty mogą analizować (WSDL)
- inne systemy współdziałają z WS w sposób przewidziany w opisie wykorzystując różne standardy WWW do wymiany komunikatów (HTTP,XML,SOAP)



KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



# Definicje c.d.

- Usługa (service) - abstrakcja opisywana przez oferowaną funkcjonalność
- Agent - konkretny moduł programowy lub sprzętowy realizujący funkcjonalność usługi
- Usługodawca (provider) i usługobiorca (requester)
  - entity - jednostka/organizacja
  - agent - oprogramowanie



# Definicje c.d.

- Opis usługi (service description) - specyfikacja interfejsu określająca formaty komunikatów, typy danych, protokoły transportowe i serializację
- Semantyka usługi (service semantics) - cel i konsekwencje interakcji

Istotą WS nie jest konkretna realizacja a współpraca pomiędzy usługodawcą a usługobiorcą.

Aby zapewnić zgodną współpracę,

**Web Services Interoperability Organization (WS-I)**

publikuje tzw. *profile* - zestawy specyfikacji

w konkretnych wersjach z szeregiem wymagań i ograniczeń.



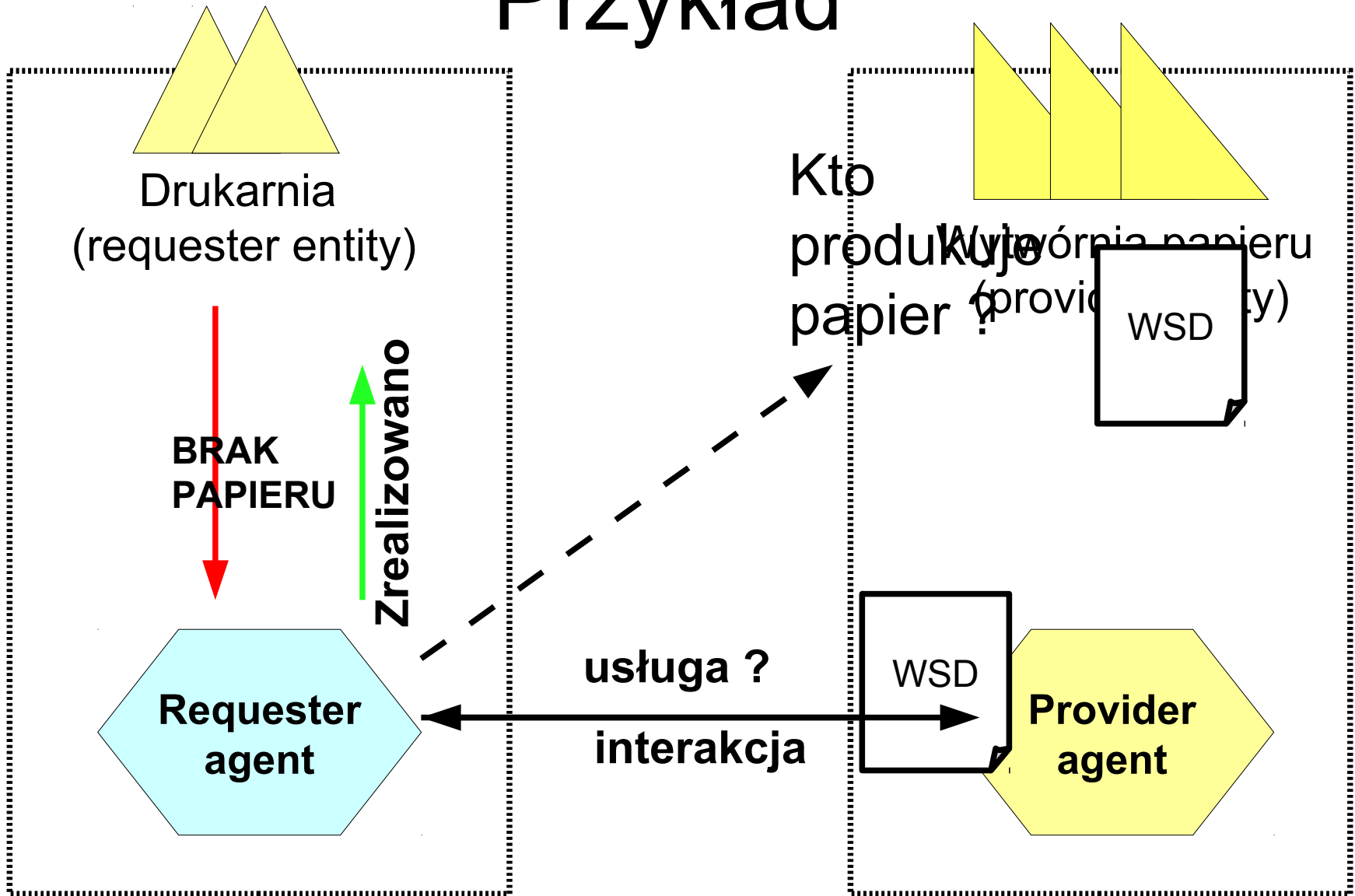
KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



# Przykład



# Cztery modele w architekturze WS

- Model komunikatów (Message Oriented)
- Model usług (Service Oriented)
- Model zasobów (Resource Oriented)
- Model polityk (Policy Oriented)



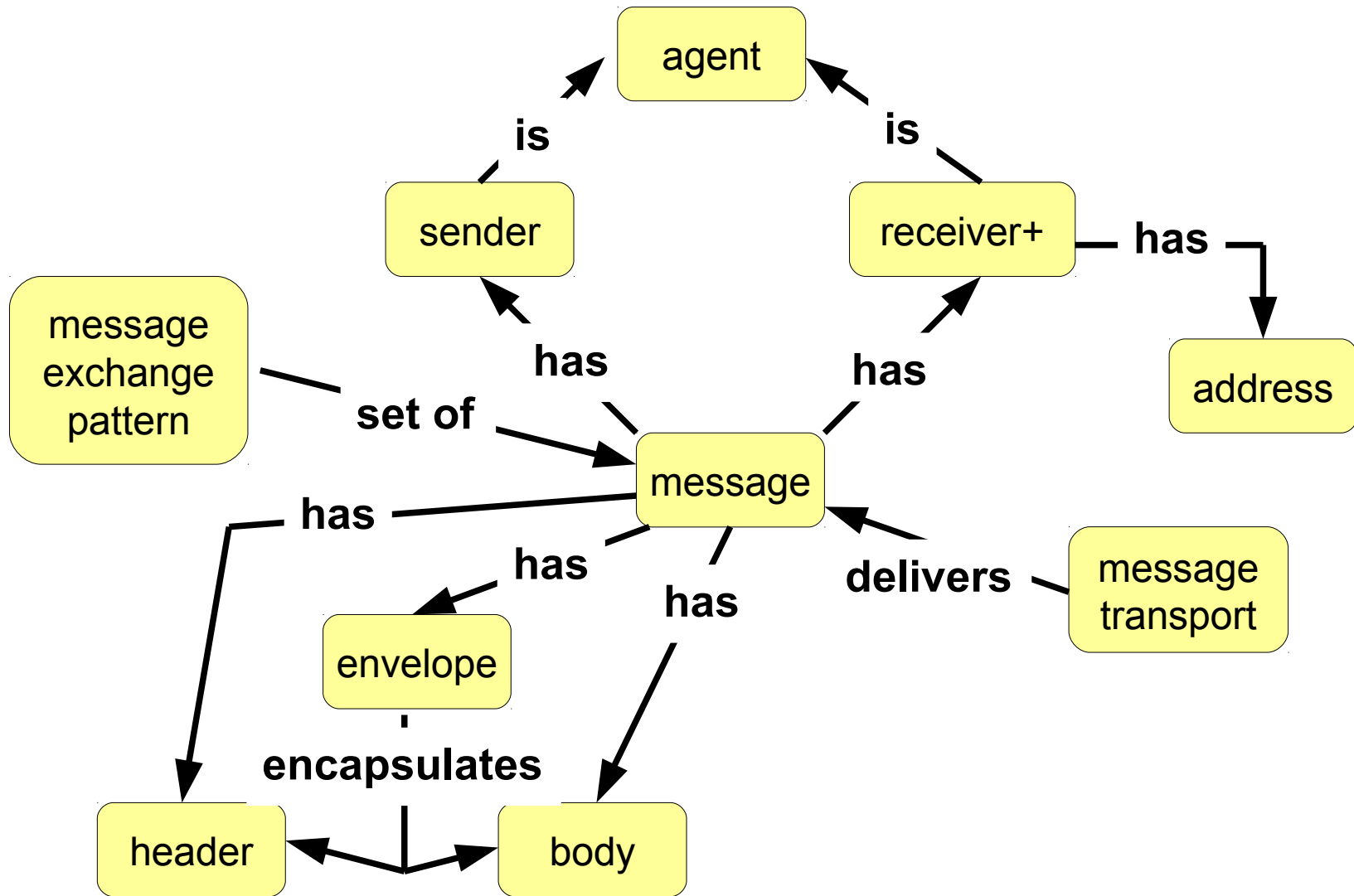
KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI



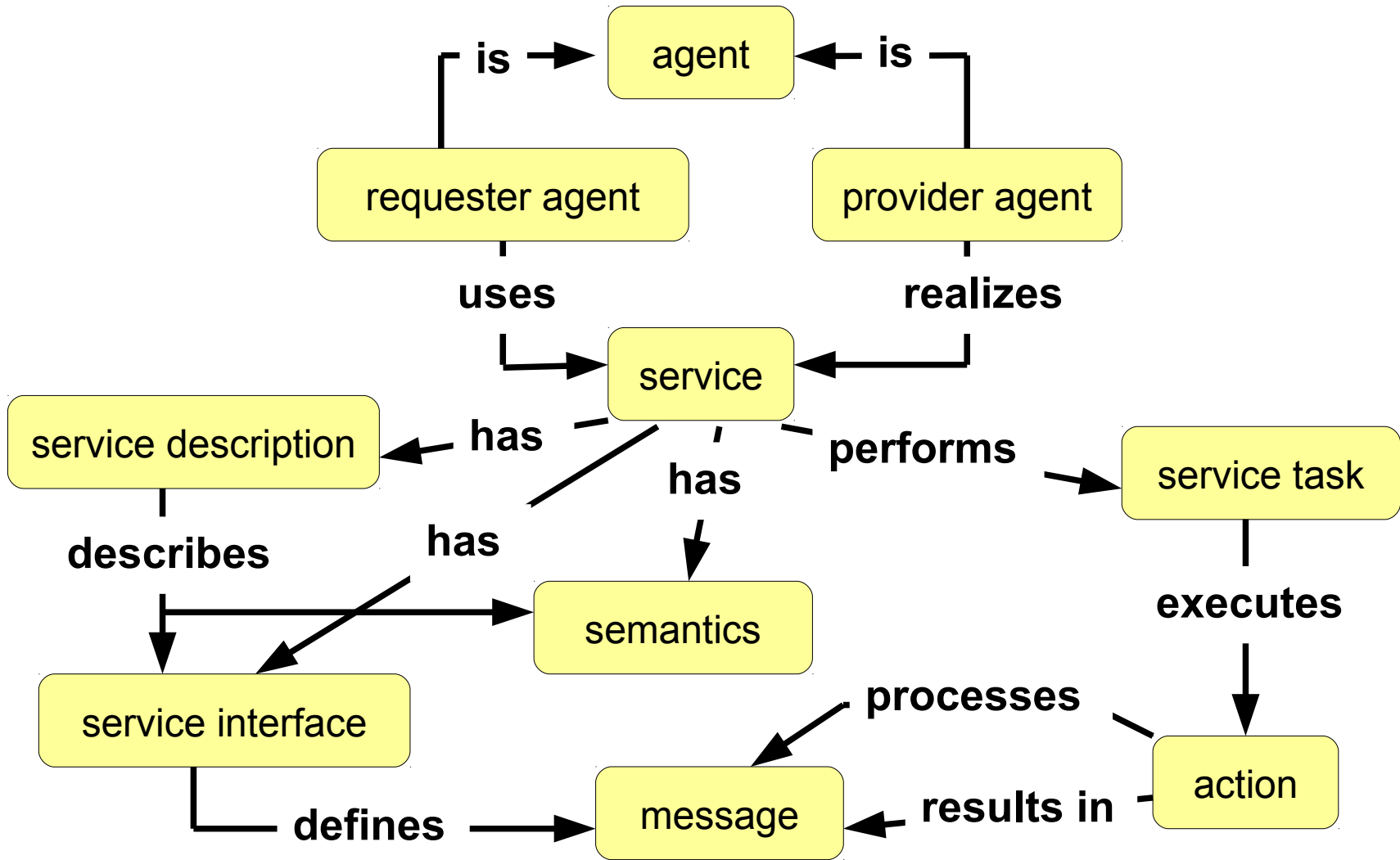
UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



# Model komunikatów (fragment)

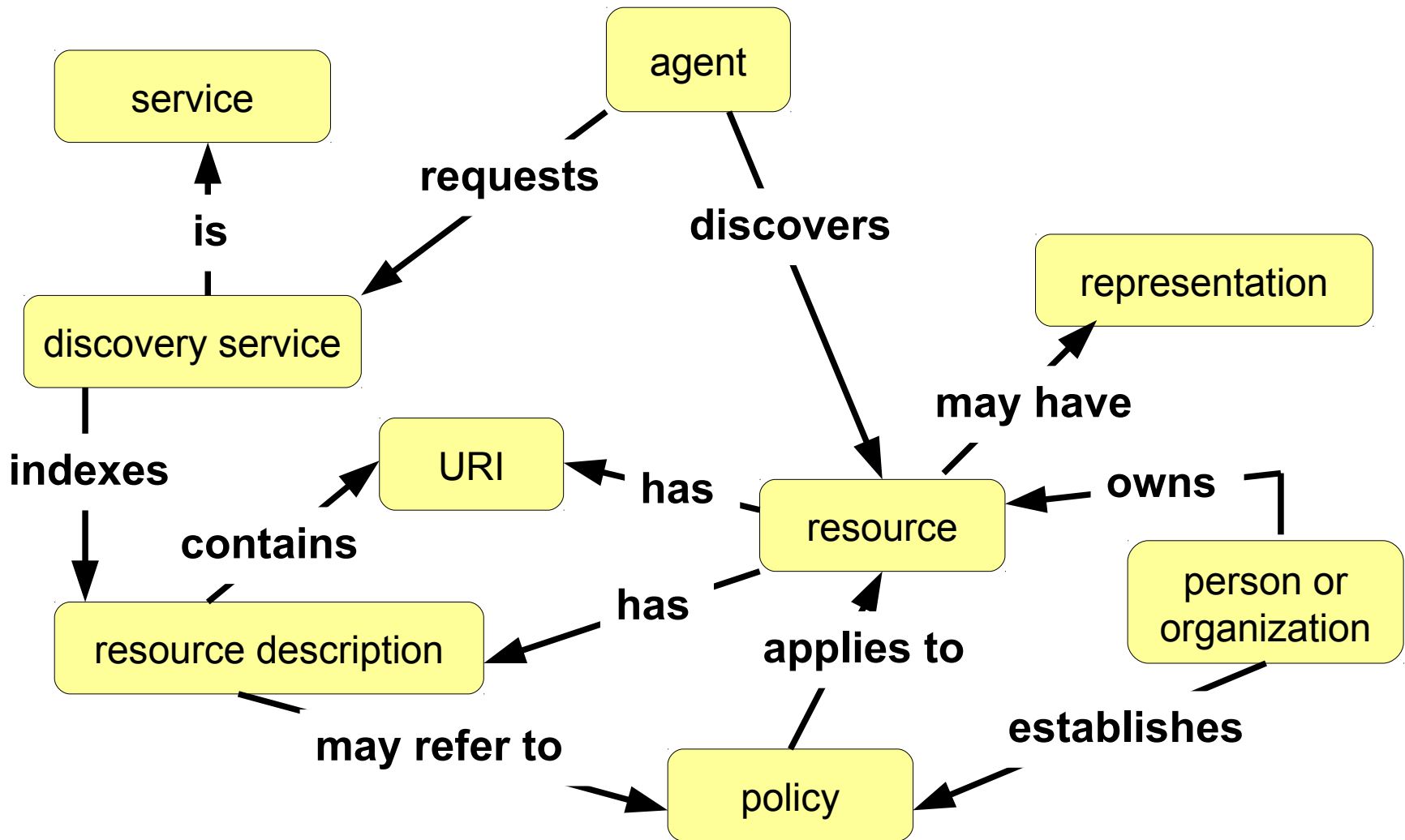


# Model usług (fragment)

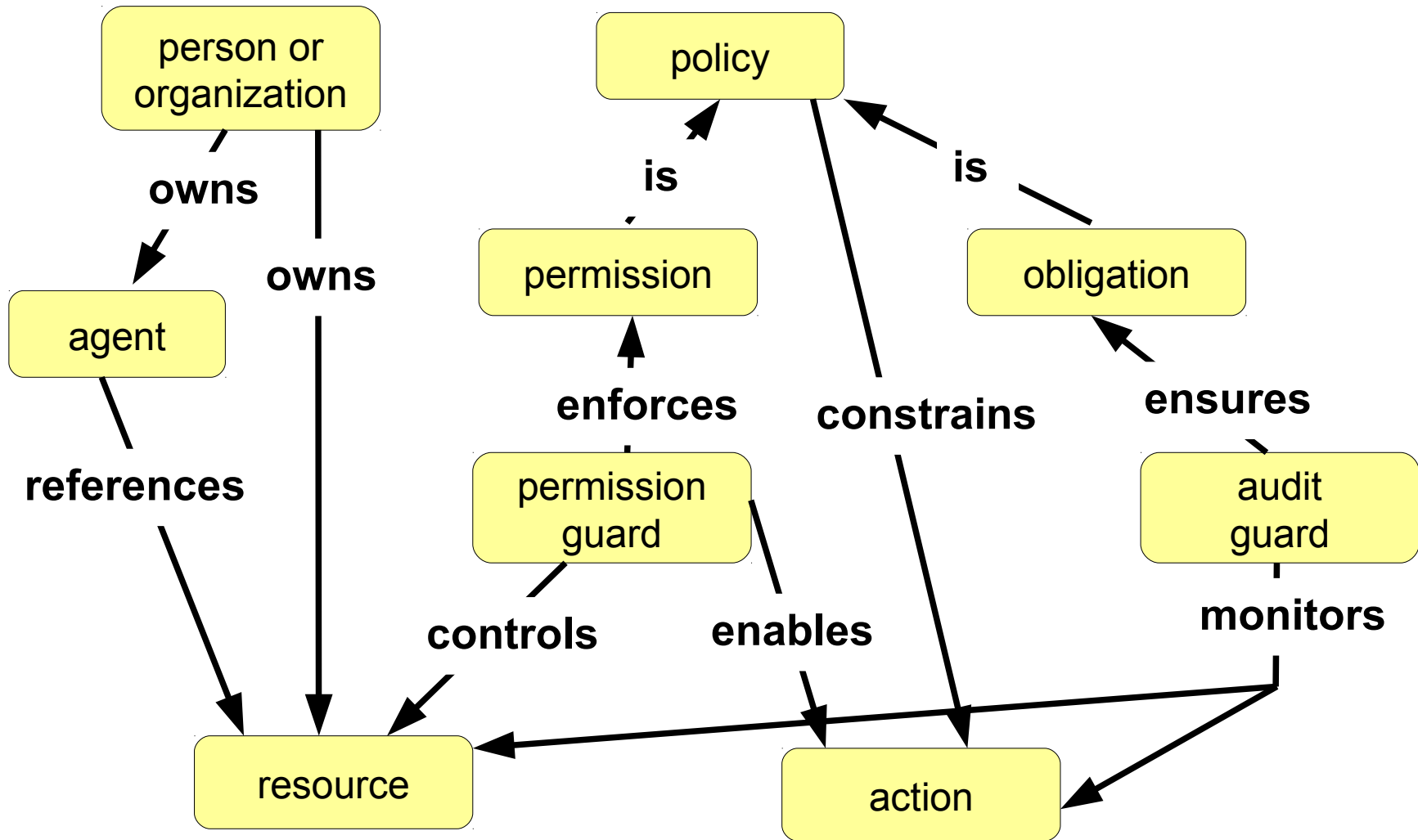




# Model zasobów



# Model polityk (fragment)



# Style

- Projektowania i realizacji
  - top-down: WSD => kod
  - bottom-up: kod => WSD
- Wykorzystania
  - Remote Procedure Call
  - Service Oriented Architecture
  - Representational State Transfer



# Service Oriented Architecture

- Usługa to logiczna reprezentacja programów, baz danych, procesów biznesowych definiowana przez oferowaną funkcjonalność
- Usługa jest formalnie opisywana przez zestaw komunikatów przesyłanych pomiędzy agentami a nie własności tych agentów. Wewnętrzna struktura agenta jest nieistotna.
- Usługa jest opisywana przez metadane, które można analizować automatycznie (maszynowo). Opis zawiera tylko to, co jest potrzebne do wykorzystania usługi. Semantyka usługi powinna być udokumentowana poprzez jej opis (jawnie bądź nie).



# Service Oriented Architecture

- Usługi wykonują zazwyczaj niewiele operacji przy stosunkowo dużych i skomplikowanych komunikatach
- Usługi powinny być zorientowane na wykorzystanie sieci jako platformy komunikacyjnej
- Komunikaty wysyłane są w standardowej postaci, niezależnej od platformy na jakiej pracują agenci (XML jest naturalnym wyborem)

**SOA nie jest równoważne Web Services**



# Standardy WS ... fragment

## Komunikaty

SOAP,  
SOAP Extensions

## Bezpieczeństwo

WS-Security,  
XML Signature,  
XML Encryption ...

## Opis

WSDL

## Podstawa

XML, XML Schema,  
XML Namespaces, XPath, ...

## Wymiana metadanych

WS-Discovery,  
WS-Policy, UDDI ...

## Transport

HTTP, FTP, SMTP  
JMS, IIOP, ...

## Dodatkowe własności

Web Services  
Resource Framework ...

## Niezawodność

WS-ReliableMessaging,  
WS-Reliability



# Przykład usługi: kalkulator

- Funkcjonalność: dodawanie i mnożenie dwu liczb całkowitych
- Komunikaty
  - $\text{Suma}(x, y) \longrightarrow \text{Wynik}(x+y)$
  - $\text{Iloczyn}(x, y) \longrightarrow \text{Wynik}(x*y)$
- Model komunikacyjny "request-response" a zatem styl RPC
- Narzędzie: Apache Axis2  
*<http://ws.apache.org/axis2/>*

# Kalkulator: opis usługi w Axis2

```
<service name="Kalkulator" scope="application">  
  <description> Prosty kalkulator </description>  
  <messageReceivers>  
    <messageReceiver  
      mep="http://www.w3.org/2004/08/wsdl/in-out"  
      class="org.apache.axis2.rpc.receivers.RPCMessageReceiver"  
    />  
  </messageReceivers>  
  <parameter name="ServiceClass">  
    wyklad.Kalkulator  
  </parameter>  
</service>
```





# Kalkulator: agent wywołujący (biblioteka "suds" dla języka Python)

```
from suds import *  
from suds.client import Client  
  
agent = Client("http:// ... ")  
  
results = agent.service.su... (4 10)  
print results
```

Operacja

Opis  
usługi



# Zagadnienia do samodzielnego przemyślenia/zbadania

- SOA 2.0 (event-driven SOA) - aplikacje sterowane zdarzeniami ?
- Wiarygodność implementacji usługi - jak opisać semantykę ?
- Model SOA a koncepcja "Software as Service"
- *Orchestration a choreography* - dwa podejścia do konstrukcji aplikacji w modelu SOA

