



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Rozproszone systemy Internetowe

Transport komunikatów WS: protokół SOAP

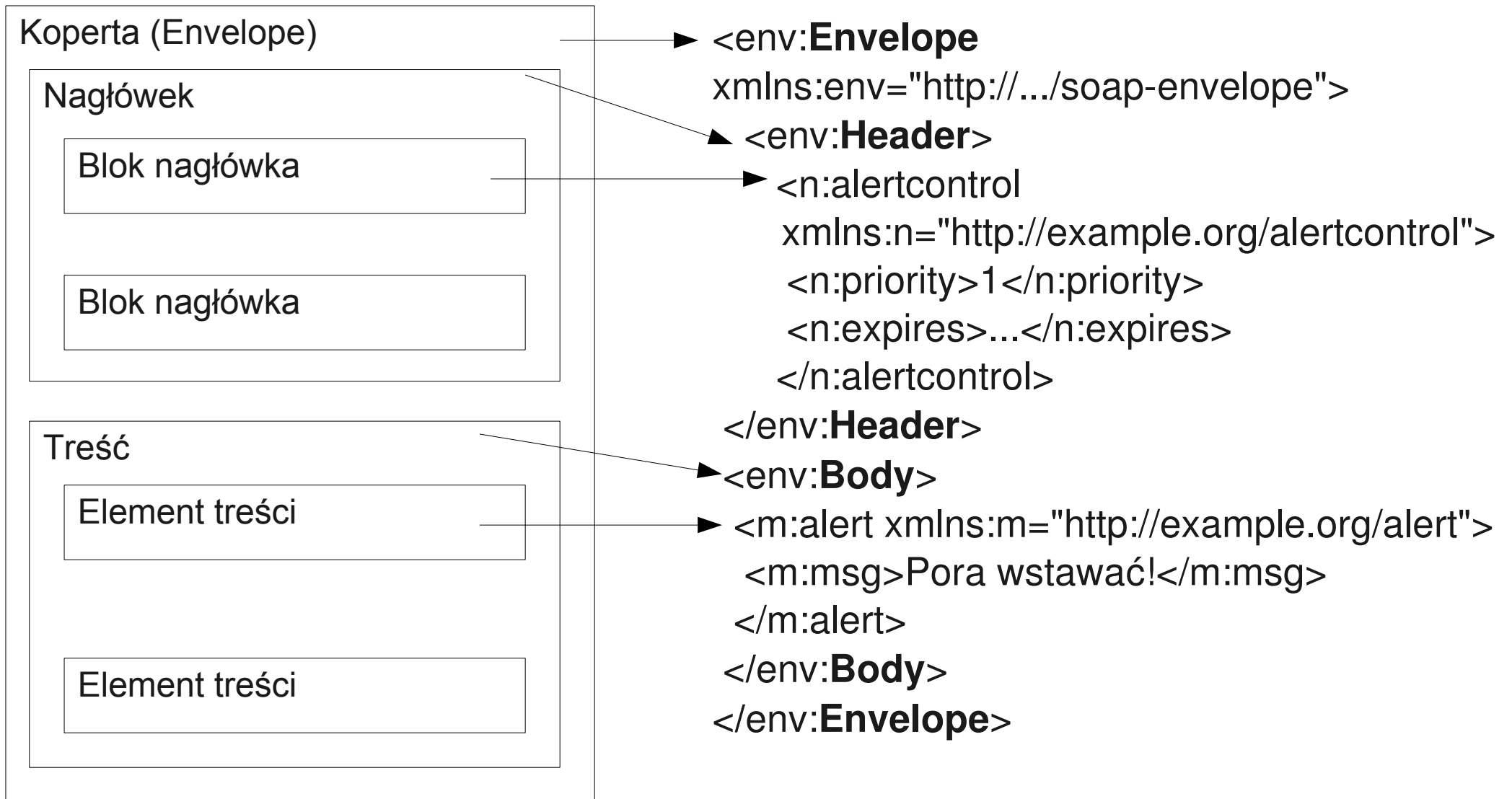
„Podniesienie potencjału uczelni wyższych jako czynnik rozwoju gospodarki opartej na wiedzy”

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

Simple Object Access Protocol

- Bezstanowy protokół komunikacyjny, oparty na standardzie XML
- Prosty i elastyczny, działa w zdecentralizowanym środowisku
- Standard definiuje:
 - Model przetwarzania wiadomości (SOAP Processing Model)
 - Rozszerzanie własności protokołu (SOAP Extensibility Model)
 - Wykorzystywanie istniejących protokołów transportowych (SOAP Protocol Binding)
 - Ogólną strukturę wiadomości (SOAP Message)
- Pojęcia podstawowe:
 - Węzeł SOAP oraz określenia jego funkcji w procesie przetwarzania (*[initial] sender, [ultimate] receiver, intermediary*)
 - Rola (role) przyjmowane przez węzeł
 - *http://www.w3.org/2003/05/soap-envelope/role/next / ultimateReceiver / none*

Struktura wiadomości

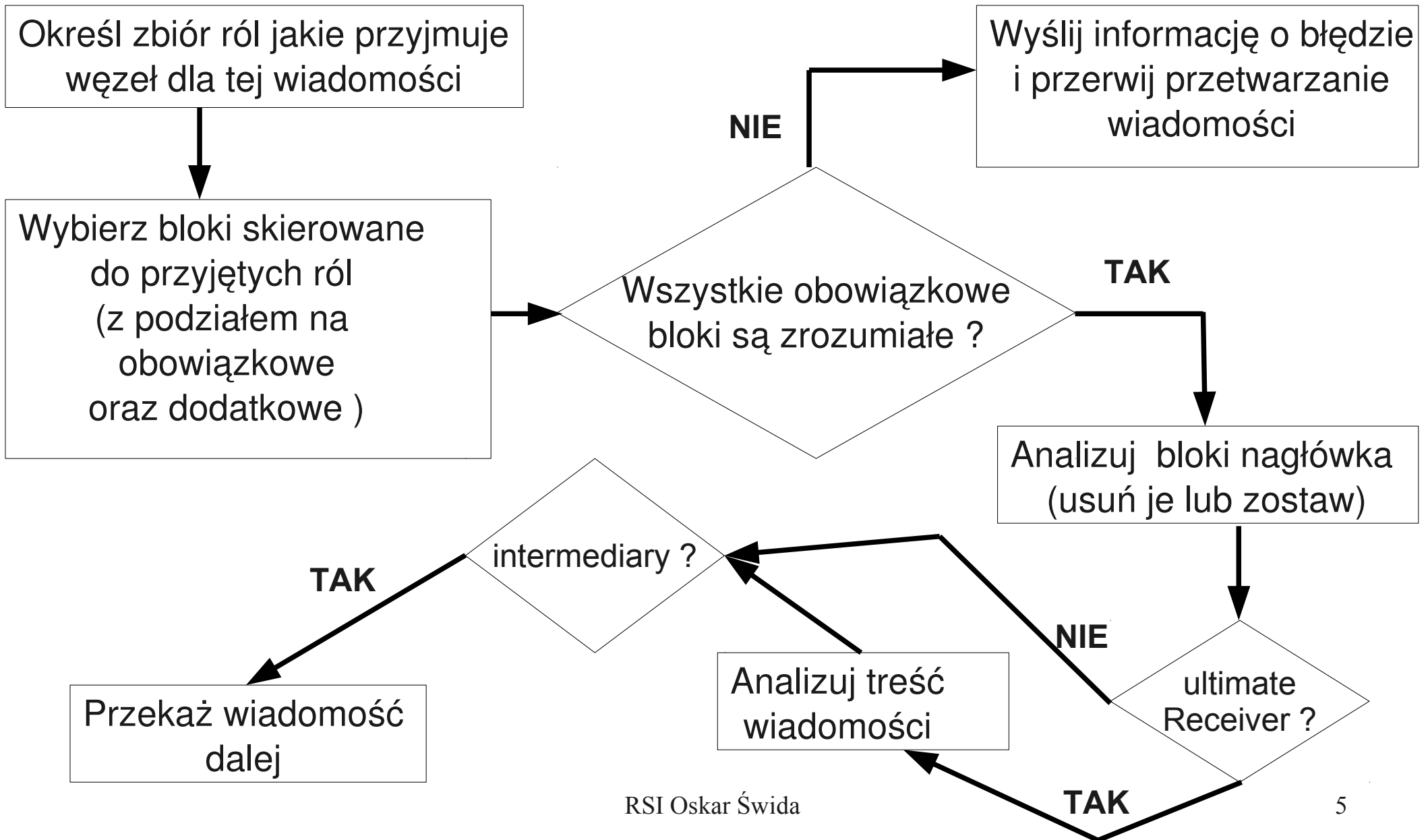


Przetwarzanie

*Wiadomość SOAP (**SOAP Message**) jest przesyłana od nadawcy (**SOAP Sender**) poprzez węzły (**SOAP Node [Intermediary]**) do odbiorcy (**SOAP Receiver**), przy użyciu dowiązanego protokołu transportowego (**SOAP Binding**), z ewentualnym wykorzystaniem dodatkowych rozszerzeń (**SOAP Feature**) opisanych za pomocą bloków wewnątrz wiadomości (**SOAP Module**).*

- Podstawowy model SOAP to jednokierunkowe przesyłanie wiadomości
- Każdy węzeł przyjmuje zbiór ról podczas przetwarzania pojedynczej wiadomości (next - każdy, none - żaden, ultimateReceiver - właściwy odbiorca)
- Bloki nagłówka
 - Skierowane do konkretnej roli (atrybut **role**)
 - Pojęcie "rozumienia" bloku oraz atrybut **mustUnderstand**
 - Treść wiadomości przetwarzana jedynie przez właściwego odbiorcę

Przetwarzanie c.d.



Przetwarzanie c.d.

- Przekazywanie wiadomości (forwarding)
 - Semantyczny mechanizm wstawiania (reinsertion)
 - atrybut **relay** (bloki, których nie przetwarzamy ale są do nas skierowane)

Rozszerzenia (Feature)

- Specyfikacja = Nazwa (URI) + Informacja potrzebna węzłom(Automat) + Model przetwarzania + Informacja przesyłana pomiędzy węzłami
- Model komunikacyjny (Message Exchange Pattern) - dodatkowo:
 - cykl życia wiadomości
 - rozmówcy biorący udział w komunikacji
 - zakończenie komunikacji (poprawne i niepoprawne)
- Moduł (Module) - specyfikacja składni i semantyki rozszerzenia

Dowiązania do protokołu transportowego

- HTTP - metoda GET (model komunikacyjny *SOAP Response*)

```
GET /www.przyklad.pl/akcja?kod=FT35ZBQ HTTP/1.1
Host: www.przyklad.pl
Accept: text/html;q=0.5, application/soap+xml
```

- HTTP - metoda POST (model *SOAP Request-Response*)

```
POST /akcja HTTP/1.1
Host: www.przyklad.pl
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: ...
```

```
<?xml version='1.0' ?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope" >
  <env:Header> ...
```

Message Transfer Optimization Mechanism

```
<?xml version='1.0' ?>  
<Envelope>  
  <Header>  
    <x>  
      ...  
    </x>  
  </Header>
```



Dane binarne ? base64 ???

Wieloczęściowy komunikat HTTP
Content-type: multipart/related
<xop:Include href="cid: ..."/>

Przykład

```
<?xml version='1.0' ?>
<env:Envelope
  xmlns:env=
    "http://.../soap-envelope">
  <env:Header>
    <m:reservation
      xmlns:m="http://reservation"
      env:role="http://.../next"
      env:mustUnderstand="true">
      <m:reference>
        uuid:093a2da1
      </m:reference>
      <m:dateAndTime>
        2001-11-29T13:20:00.000-05:00
      </m:dateAndTime>
    </m:reservation>
    <n:passenger
      xmlns:n="http://employees"
      env:role="http://.../next"
      env:mustUnderstand="true">
      <n:name>Ala Kowalska</n:name>
    </n:passenger>
  </env:Header>
```

```
<env:Body>
  <p:itinerary
    xmlns:p="http://travel">
    <p:departure>
      <p:departing>...</p:departing>
      <p:arriving>...</p:arriving>
      <p:departureDate />
      <p:departureTime />
      <p:seatPreference />
    </p:departure>
    <p:return>
      <p:departing>...</p:departing>
      <p:arriving>...</p:arriving>
      <p:departureDate />
      <p:departureTime />
      <p:seatPreference/>
    </p:return>
  </p:itinerary>
</env:Body>
</env:Envelope>
```

Błędy: przykład

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
              xmlns:rpc='http://www.w3.org/2003/05/soap-rpc'>
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env:Sender</env:Value>
        <env:Subcode>
          <env:Value>rpc:BadArguments</env:Value>
        </env:Subcode>
      </env:Code>
      <env:Reason>
        <env:Text xml:lang="en-US">Processing error</env:Text>
        <env:Text xml:lang="pl">Błąd przetwarzania</env:Text>
      </env:Reason>
      <env:Detail>
        <e:myFaultDetails xmlns:e="http://example.org/faults">
          <e:message>Name does not match card number</e:message>
          <e:errorCode>999</e:errorCode>
        </e:myFaultDetails>
      </env:Detail>
    </env:Fault>
  </env:Body></env:Envelope>
```

Zagadnienia do samodzielnego przemyślenia/zbadania

- Definicja własnych modeli komunikacyjnych (Message Exchange Pattern)
- Definicja własnych dowiązań transportowych
- Oddzielna specyfikacja SOAP do wywołań RPC
- Rozszerzenie określane jako ***Web Method***

Dyskusja swobodna

- Standard komunikacji dla strukturalnych wiadomości
- XML jako uniwersalna platforma wymiany informacji
- Wady i zalety SOAP
- Technologie wykorzystujące protokół
 - Web Services (WSDL, UDDI)
 - AJAX (Asynchronous Javascript And XML)
- SOAP over E-mail