

Web servisai

WS-Addressing

Osvaldas Grigas

WS-Addressing

- Dalis informacijos, susijusios su SOAP pranešimo transportavimu, paprastai yra nurodoma transporto protokolo lygyje, pvz. HTTP:
 - Gavėjo serveris - HTTP header'yje "Host"
 - Web serviso adresas - iškart po užklauso metodo (GET, POST ir t.t)
 - Web serviso operacija - header'yje "SOAPAction" (SOAP 1.1)
- Kadangi SOAP pranešimai yra neutralūs transporto atžvilgiu, kyla problema, kaip visą "papildoma" informaciją perduoti kitais protokolais, pvz. SMTP, TCP, XMPP ir kt.

WS-Addressing

- WS-Addressing leidžia visą informaciją, susijusią su SOAP pranešimo transportu, įkelti į patį SOAP pranešimą
 - "True transport-independence"
 - Padeda įgyvendinti SOA architektūros principus
- Specifikaciją sukūrė Microsoft, IBM, BEA, Sun ir SAP
- 2006 m. WS-Addressing tapo W3C rekomendacija

WS-Addressing pavyzdys

```
<Envelope xmlns="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <Header>
    <wsa:To>http://example.com/Servisas</wsa:To>
    <wsa:Action>http://example.com/Servisas/Operacija</wsa:Action>
  </Header>
  <Body>
    ...
  </Body>
</Envelope>
```

- Blokas "To" nurodo galutinio gavėjo adresą
- Blokas "Action" nurodo serviso operaciją (jei reikia)

Asinchroninė komunikacija

Asinchroninė komunikacija

- WS-Addressing suteikia galimybę dviems Web servisams bendrauti asinchroniškai. Pavyzdžiui:
 - Servisas A siunčia pranešimą servisui B
 - Pranešimas identifikuojamas bloke "MessageID"
 - Servisas A, nelaukdamas atsakymo, toliau daro kitus darbus
 - Servisas B gauna pranešimą ir atlieka kokį nors ilgai trunkantį veiksmą
 - Baigęs darbą, servisas B siunčia pranešimą servisui A
 - Pranešime pažymima, kad tai yra atsakymas į ankstesnį serviso A pranešimą
 - Tam naudojamas blokas "RelatesTo" su ankstesne "MessageID" reikšme
 - Servisas A gauna atsakymą ir apdoroja gautą informaciją
- Bloke "ReplyTo" galima nurodyti, kokių adresu siųsti atsakymą
 - Tai nebūtinai turi būti pradinis siuntėjas
- Bloke "FaultTo" galima nurodyti, kam siųsti klaidos pranešimą
 - Vėlgi, tai gali būti arba siuntėjas, arba kokia nors trečioji šalis

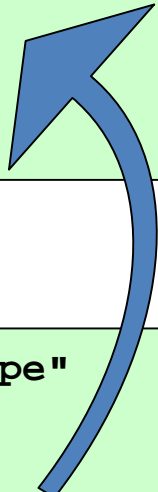
Asinchroninė komunikacija

Asinchroninė užklausa:

```
<Envelope xmlns="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <Header>
    <wsa:ReplyTo>
      <wsa:Address>http://example.com/Pradinis_Siuntejas</wsa:Address>
    </wsa:ReplyTo>
    <wsa:MessageID>6B29FC40-CA47-1067-B31D-00DD010662DA</wsa:MessageID>
  </Header>
  ...
</Envelope>
```

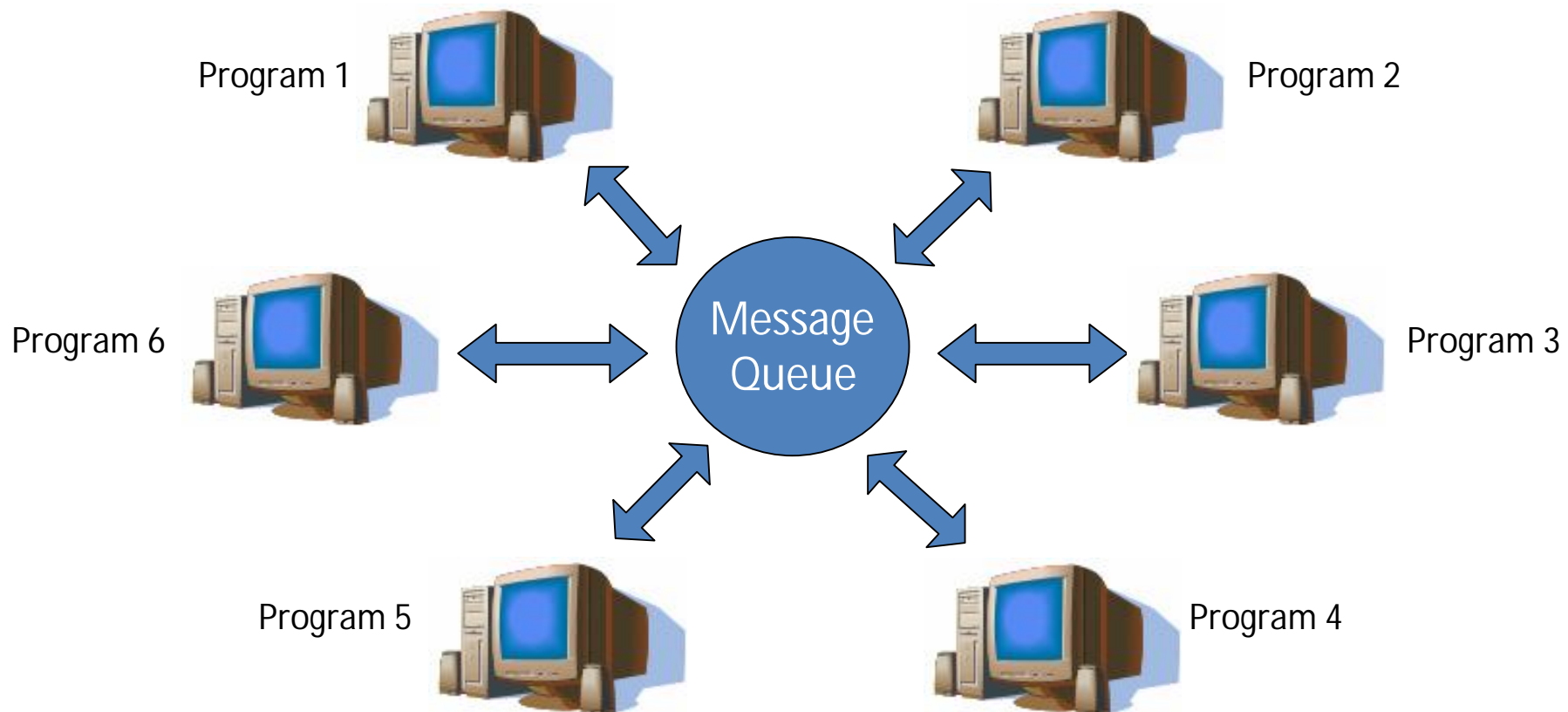
Asinshroninis atsakymas:

```
<Envelope xmlns="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <Header>
    <wsa:RelatesTo>6B29FC40-CA47-1067-B31D-00DD010662DA</wsa:RelatesTo>
  </Header>
  ...
</Envelope>
```



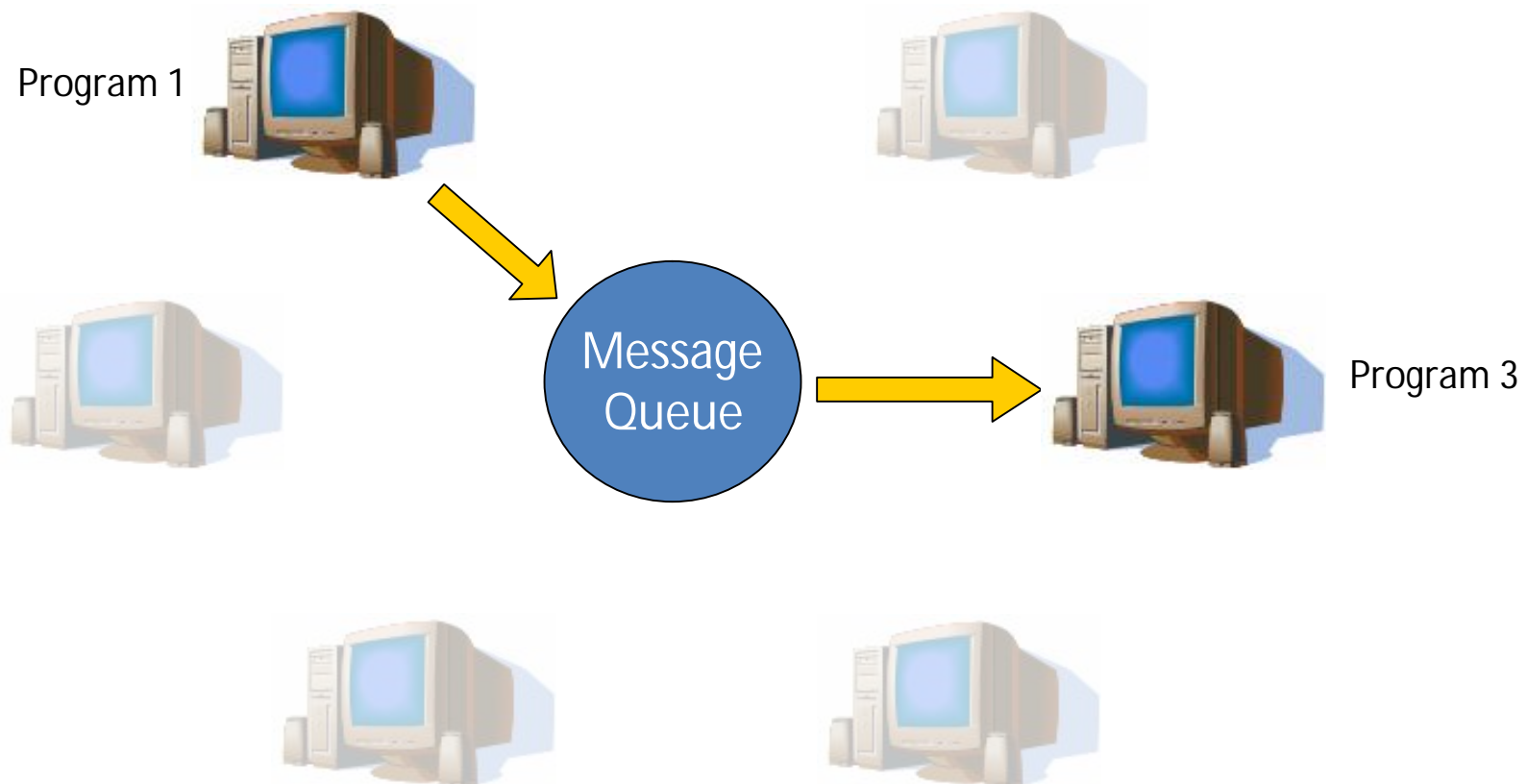
Message-Oriented Middleware

- Message-oriented middleware (MOM) - specifinė kliento/serverio infrastruktūra, pagrįsta asinchroniais pranešimais
- Nuo seno naudojama sistemų integravimui
- Supaprastina komunikaciją tarp skirtingų programų, OS ir pan.



MOM ir WS-Addressing

- Kai Web servisai naudojami MOM infrastruktūroje, asinchroninė komunikacija realizuojama su WS-Addressing
 - Šiuo atveju tiesioginis gavėjas (Message Queue) nesutampa su galutiniu gavėju (Program 3), todėl jį būtina nurodyti WS-Addressing bloke "To"
 - Jei laukiama atsakymo, būtina nurodyti pradinį siuntėją bloke "ReplyTo"





Endpoint Reference

Endpoint Reference

- Kartais norint identifikuoti galutinį gavėją, vien adreso neužtenka
- Ypač intranetinėse/MOM aplikacijose, kai yra poreikis kontroliuoti servisų gyvavimo ciklą (lifecycle)
 - Abstraktus servisas traktuojamas kaip "klasė", turinti vieną adresą
 - Konkretus servisas traktuojamas kaip tos klasės "objektas"
 - pasiekiamas per tą patį adresą
 - bet operuoja atskirai nuo visų kitų "objektų"
 - laiko savo privačius "sesijos" duomenis
- Pavyzdžiui, pirkinių krepšelio servisas "ShoppingCart" gali sukurti po atskirą savo paties "objektą" kiekvienam klientui
 - Tokiu atveju konkrečiam "ShoppingCart" servisui identifikuoti reikėtų:
 - Jo bendrojo adreso
 - Ir kliento "sesijos" identifikatoriaus

Endpoint Reference

- WS-Addressing tai vadina *Endpoint Reference (EPR)*
- Serviso EPR sudaro jo adresas + papildomi parametrai
 - Tie papildomi parametrai vadinami "Reference Parameters"
 - WS-Addressing neapibrėžia nei vieno tokio parametro, bet leidžia tą padaryti pačioms aplikacijoms

```
<wsa:EndpointReference>  
  <wsa:Address>http://example.com/ServiceAddress</wsa:Address>  
  <wsa:ReferenceParameters>  
    <my:SessionID xmlns:my="my-service">43513798516</my:SessionID>  
  </wsa:ReferenceParameters>  
</wsa:EndpointReference>
```