

# Web servisai

WSDL

Osvaldas Grigas

# Web servisų aprašymas

- Kiekvienas web servisas yra unikalus
  - Jis turi *adresą* (arba kelis adresus), kuriuo į jį galima kreiptis.
  - Jis supranta tik tam tikros struktūros įeinančius pranešimus ir siunčia atgal tik apibrėžtos struktūros išeinančius pranešimus. Toks apsikeitimas pranešimais vadinamas *operacija*. Kelios susijusios operacijos sudaro *interfeisą*. Web servisas gali turėti vieną arba daugiau interfeisų.
  - Kiekvienas web serviso interfeisas gali būti pasiekiamas naudojant tik tam tikrą komunikacijos/serializacijos/transporto *protokolų* kombinaciją (pvz. SOAP/XML/HTTP). Skirtingiems interfeisams gali reikėti to paties protokolo, o gali reikėti ir skirtingų protokolų. Kartais vienas interfeisas gali būti pasiekiamas keliais protokolais.
- Kad galėtų sėkmingai bendrauti su web servisu, klientas turi žinoti visas šias technines detales, taigi jos turi būti aprašytos mašiniškai suprantamu pavidalu ir pasiekiamos *online*

# WSDL

- *WSDL* = Web Services Description Language
- Tai populiariausia web servisų aprašymo kalba
- Naudoja XML formatą
- Dvi versijos: WSDL 1.1 ir WSDL 2.0
- WSDL 1.1
  - Sukurta Microsoft, IBM ir Ariba. 2001 m. nusiųsta į W3C standartizavimui
  - Ši versija taip niekada ir nebuvo standartizuota
  - Problemos: nelanksti, gremėzdiška ir kai kuriose vietose neaiški
  - Nelabai tinka REST stiliaus servisams aprašyti
  - Bet naudojama jau ilgą laiką ir iki šiol pati populiariausia
- WSDL 2.0
  - Visiškai perrašyta specifikacija su žymiais pakeitimais
  - Leidžia lanksčiai, lakoniškai ir vienareikšmiškai aprašyti bet kokį web servisą
  - 2007 m. išleista kaip W3C rekomendacija ("standartas")
  - Bet šią versiją vis dar mažai kas palaiko, todėl ji kol kas naudojama retai

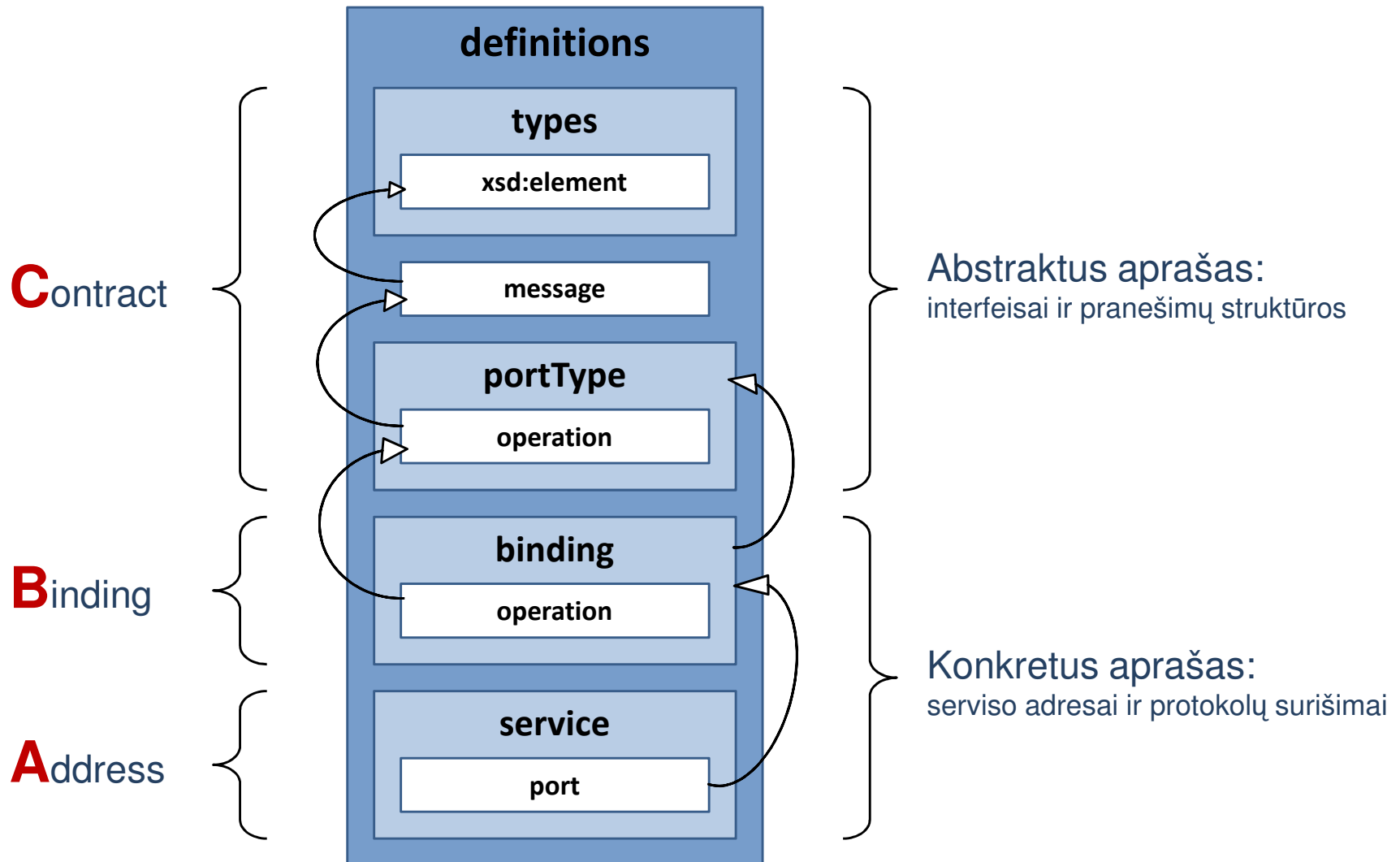
# WSDL "abėcėlė"

**A**ddress  
**B**inding  
**C**ontract



Tokias web serviso savybes  
aprašo WSDL dokumentas

# WSDL 1.1 struktūra



# WSDL 1.1 struktūra

```
<definitions>
```

```
<!-- <import/> -->
```

```
<types>
```

```
<schema>
```

```
<element name="element1"/>
```

```
<element name="element2"/>
```

```
</schema>
```

```
</types>
```

```
<message name="message1"/>
```

```
<message name="message2"/>
```

```
<portType name="interface1">
```

```
<operation name="operation1"/>
```

```
<operation name="operation2"/>
```

```
</portType>
```

```
<binding name="binding1">
```

```
<operation name="operation1"/>
```

```
<operation name="operation2"/>
```

```
</binding>
```

```
<binding name="binding2">
```

```
<operation name="operation1"/>
```

```
<operation name="operation2"/>
```

```
</binding>
```

```
<service name="service1">
```

```
<port name="port1"/>
```

```
<port name="port2"/>
```

```
</service>
```

```
</definitions>
```

- WSDL apraše galima išskirti dvi dalis: abstrakčią ir konkrečią
- *Abstrakti* dalis apibrėžia serviso kontraktą, t.y. interfeisus, pranešimus (įeinančius, išeinančius ir klaidos) bei duomenų struktūras.
  - Abstrakti dalis atsako į klausimą, *ką* daro web servisas, t.y. kokios jo operacijos
- *Konkrečioje* dalyje nurodoma, koku adresu reikia kreiptis į servisą bei kokius transporto, serializacijos ir komunikacijos protokolus reikia naudoti.
  - Konkreti dalis atsako į klausimus, *kur* yra servisas ir *kaip* kviesti jo operacijas
- Šios dalys kartais būna laikomos atskiruose WSDL failuose, nes kai kuriais atvejais abstrakčią dalį gali pakartotinai panaudoti keli konkretūs servisų aprašai: tuomet konkretus aprašas importuoja abstraktų aprašą, naudodamas <import> elementą.

# Šakninis elementas

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"  
  xmlns:tns="urn:my-service"  
  targetNamespace="urn:my-service">  
  
</definitions>
```

- Šakninis elementas `<definitions>` attribute *targetNamespace* nurodo web serviso vardų erdvę.
  - Vardų erdvė dažnai sutampa su web serviso adresu, tačiau ji gali būti bet koks URI.
- Po šia vardų erdve atsiduria WSDL aprašo komponentai: `<message>`, `<portType>`, `<binding>`, ir `<service>`. Visi šie komponentai turi vardus (atributas *name*), yra tarpusavyje susiję ir laiko nuorodas vieni į kitus. Tokios nuorodos turi būti kvalifikuotos su web serviso vardų erdve.
  - Pavyzdžiui, nuoroda į komponentą vardu "interface1" turi būti "tns:interface1", kur "tns" yra web serviso vardų erdvės prefiksas, paprastai nurodomas šaknyje.

# Duomenų struktūros

```
<types>
  <schema targetNamespace="urn:my-service" xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="PurchaseOrder">
      <complexType>
        <sequence>
          <element name="itemName" type="string"/>
          <element name="quantity" type="int"/>
        </sequence>
      </complexType>
    </element>
    <element name="Invoice">
      <complexType>
        <sequence>
          <element name="number" type="int"/>
          <element name="orderDate" type="dateTime"/>
        </sequence>
      </complexType>
    </element>
  </schema>
</types>
```

- Pranešimuose naudojamas duomenų struktūras (duomenų tipus) apibrėžia XML Schema, esanti elemente <types>
- Schemas *targetNamespace* paprastai sutampa su web serviso *targetNamespace*
- Vietoj XML Schemas leidžiama naudoti ir kitą kalbą (pvz. DTD), tačiau XML Schemas palaikymas yra plačiausias ir dėl to dažniausiai naudojamas



# Pranešimai

```
<message name="PurchaseInput">  
  <part name="body" element="tns:PurchaseOrder"/>  
</message>
```

```
<message name="PurchaseOutput">  
  <part name="body" element="tns:Invoice"/>  
</message>
```

- Kiekvienas įvesties, išvesties ir klaidos pranešimas aprašomas elementu `<message>`
- Pranešimas gali būti sudarytas iš kelių dalių
- Kiekviena pranešimo dalis nurodoma `<part>` atribute *element*, kuris rodo į XML Schemoje aprašytą duomenų struktūrą
- *Document-style* web servisuose rekomenduojama nenaudoti pranešimų, kuriuos sudaro daugiau nei viena dalis, o WSDL 2.0 apskritai panaikino tokią galimybę
- *Document-style* web servisuose `<part>` atributas *name* neturi reikšmės
  - tačiau RPC stiliaus web servisuose jis dažnai nurodo procedūros parametro vardą

# Interfeisai

```
<portType name="ShoppingInterface">
  <operation name="Purchase">
    <input message="tns:PurchaseInput"/>
    <output message="tns:PurchaseOutput"/>
    <fault name="fault1" message="tns:PurchaseFault"/>
  </operation>
  .....
</portType>
```

- Servisas gali turėti vieną arba daugiau interfeisų. Dažniausiai tik vieną
- Kiekvienas interfeisas aprašomas elemente <portType>
- Interfeiso operacijos aprašomos elementuose <operation>
- Operacija suprantama kaip apsikeitimas pranešimais, todėl nurodomi įvesties (<input>), išvesties (<output>) ir galimi klaidos (<fault>) pranešimai
  - Atributai "message" rodo į anksčiau aprašytus <message> komponentus
- *Fault* pranešimas siunčiamas vietoj *Output* pranešimo tuo atveju, jei, tarkim, atitinkamas web serviso klasės metodas išmeta Exception

# Surišimas su protokolais

```
<binding name="SoapBinding" type="tns:ShoppingInterface">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="Purchase">
    <soap:operation soapAction="PurchaseAction"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  .....
</binding>
```

- Surišimas su protokolais (binding'as) - sudėtingiausia WSDL aprašo dalis.
- Elementas <binding> aprašo, kaip turi būti kviečiama konkretaus interfeiso (nurodyto atribute *type*) kiekviena operacija, t.y. kokie protokoliai naudojami komunikacijai, pranešimų serializavimui ir perdavimui tinklu.
- Kadangi protokolų yra daug ir įvairių, o jų aprašymas labai skiriasi, WSDL leidžia naudoti taip vadinamus *binding extensions* - elementus iš kitos vardų erdvės.
- Pavyzdyje aprašytas toks surišimas: komunikacijos protokolas SOAP, transporto protokolas HTTP, serializuojama tekstiniu XML (nes nenurodyta kitaip), naudojant *document-style* pranešimus (*style="document"*), kurių turinys apibrėžtas XML Schema (*use="literal"*).
- Vienas interfeisas gali turėti kelis binding'us, pvz. SOAP ir REST.

# Serviso galiniai taškai

```
<service name="ShoppingService">
  <port name="SoapPort" binding="tns:SoapBinding">
    <soap:address location="http://www.example.com/ShoppingService/SOAP"/>
  </port>
  <port name="RestPort" binding="tns:RestBinding">
    <soap:address location="http://www.example.com/ShoppingService/REST"/>
  </port>
</service>
```

- Elementas <service> aprašo web servisą kaip galinių taškų (endpoints) rinkinį.
- Kiekvieną galinį tašką atstovauja elementas <port>.
- Galinis taškas iš esmės yra adresas (URL), per kurį pasiekiamas konkretus web serviso binding'as. Taigi kiekvienam binding'ui tenka po unikalų adresą.
  - Galinio taško binding'as nurodomas atribute *binding*
  - Galinio taško adresas nurodomas skirtingai, priklausomai nuo binding'o. Pavyzdžiui, naudojant SOAP binding'ą, adresas nurodomas elemente <soap:address>
- Jeigu WSDL dokumente yra keli interfeisai, paprastai kiekvienam iš jų bus sukurtas atskiras <service> elementas.

# WSDL 2.0

# WSDL 2.0 struktūra

