

Kje smo?

- 1. Uvod
- 2. Informacijski sistemi
- 3. Poslovni sistem, poslovni proces in organizacija
- 4. E-poslovanje
- 5. Informacijske tehnologije
- ⇒ 6. Arhitekture porazdeljenih sistemov

6. Arhitekture porazdeljenih sistemov

- Vsebina poglavja:
 - ⇒ 6.1 Uvod
 - 6.2 Arhitekture odjemalec strežnik
 - 6.3 Arhitekture porazdeljenih objektov
 - 6.4 Arhitekture P2P
 - 6.5 SOA - Storitveno usmerjena arhitektura

6.1 Uvod

- Porazdeljen sistem je sistem, kjer se procesiranje izvaja na več računalnikih.
- Računalniki med seboj komunicirajo preko omrežja.
- Program je razdeljen na več delov, ki se vzporedno izvajajo na več računalnikih.
- Porazdeljen sistem ima pogosto opravka z:
 - heterogenimi okolji,
 - nepredvidljivimi latenčnimi časi in kapacitetami omrežja,
 - Nepredvidljivimi napakami na omrežju ali računalnikih povezanih v omrežje
- Večina sodobnih IS je zasnovanih v obliki porazdeljenih sistemov.

478

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.1 Uvod

Značilnosti

- Deljenje virov
- Odprtost sistema
- Sočasnost
- Razširljivost
- Odpornost na napake

Slabosti

- Kompleksnost
- Varnost
- Upravljanje
- Nepredvidljivost

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.1 Uvod

- **Vmesni sloj** (middleware) je programska oprema, ki upravlja in podpira delovanje različnih komponent porazdeljenega sistema.
- Pri dobri arhitekturi razvijalec vmesni sloj uporablja, brez da bi vedel podrobnosti njegovega delovanja.
- Vmesni sloj je tipično kupljena programska oprema, ki jo programerji le uporabijo.

6.1 Uvod

- Vmesni sloji so ključen element pri **sistemski integraciji** (povezavi različnih sistemov v enotno, celovito rešitev).

6.1 Uvod

- Arhitekture porazdeljenih sistemov:
 - Arhitekture odjemalec-strežnik
(strežniki izvajajo porazdeljene storitve, ki jih uporabljajo odjemalci)
 - Arhitekture porazdeljenih objektov
(ni razlikovanja med odjemalci in strežniki, katerikoli objekt lahko nudi storitve ali uporablja storitve drugega objekta)
 - Storitveno usmerjene arhitekture
(temeljijo na splošno dostopnih storitvah)

6. Arhitekture porazdeljenih sistemov

- Vsebina poglavja:
 - 6.1 Uvod
 - ⇒ 6.2 Arhitekture odjemalec strežnik
 - 6.3 Arhitekture porazdeljenih objektov
 - 6.4 Arhitekture P2P
 - 6.5 SOA - Storitveno usmerjena arhitektura

6.2 Arhitekture odjemalec strežnik

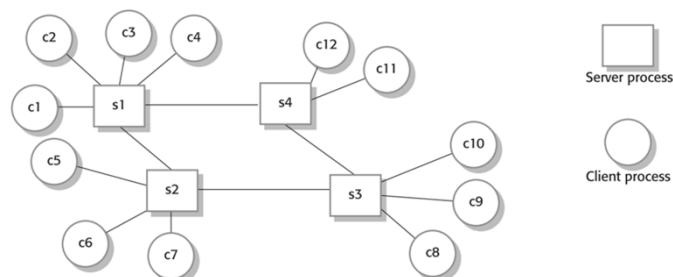
- Osnovni komponenti arhitekture odjemalec-strežnik sta odjemalec (zahteva storitev) in strežnik (izvaja storitev).
- Procesiranje strežnik-odjemalec naj bi potekalo relativno uravnoteženo, pri čemer specializirane storitve izvaja strežnik.
- Sistem si lahko predstavljamo kot množico storitev, ki jih zagotavljajo strežniki in množico odjemalcev, ki uporabljajo te storitve
- Odjemalci poznajo strežnike, strežnikom pa ni potrebno poznati odjemalcev.

484

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.2 Arhitekture odjemalec strežnik

- Odjemalce in strežnike lahko predstavimo kot logične procese
- Preslikava med proces in procesorji pogosto ni 1:1.



485

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.2 Arhitekture odjemalec strežnik

- V porazdeljenem sistemu lahko sodelujejo različni strežniki:
 - Datotečni strežnik (File server)
 - Podatkovni strežnik (Database server)
 - Strežnik za skupinsko delo (Groupware server)
 - Spletni strežnik (Web server)
 - Poštni strežnik (Mail server)
 - Strežnik objektov (Object server)
 - Strežnik za tiskanje (Print server)
 - Aplikacijski strežnik (Application server)
 - ...

6.2 Arhitekture odjemalec strežnik

- Porazdeljeni sistemi delujejo na arhitekturah z več nivoji (ravni).
- Najbolj znani sta dvo-nivojska (model težkega odjemalca) in tro-nivojska arhitektura (model lahkega odjemalca).

6.2 Arhitekture odjemalec strežnik

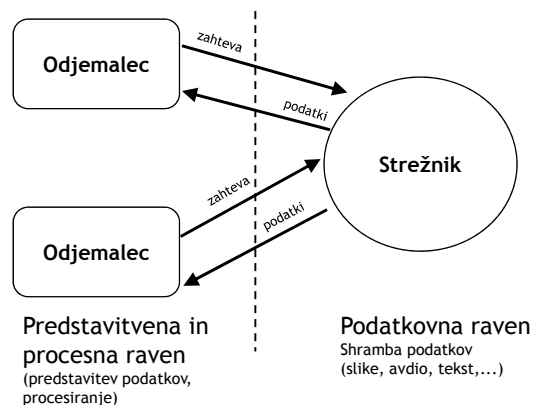
- Dvo-nivojsko arhitekturo (model težkega odjemalca) sestavlja:
 - Predstavitevna in logična raven (predstavitev in procesiranje podatkov, ki jih vrne podatkovni strežnik); Fizično realizirana na odjemalcu.
 - Podatkovna raven (shramba podatkov); Fizično realizirana na strežniku.
- Dvo-nivojska arhitektura je smiselna, kadar je malo procesiranja podatkov.
- Primeri: datotečni strežnik, podatkovni strežnik, itd. do katerih neposredno dostopajo odjemalci

488

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.2 Arhitekture odjemalec strežnik

Primer dvo-nivojske arhitekture



489

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.2 Arhitekture odjemalec strežnik

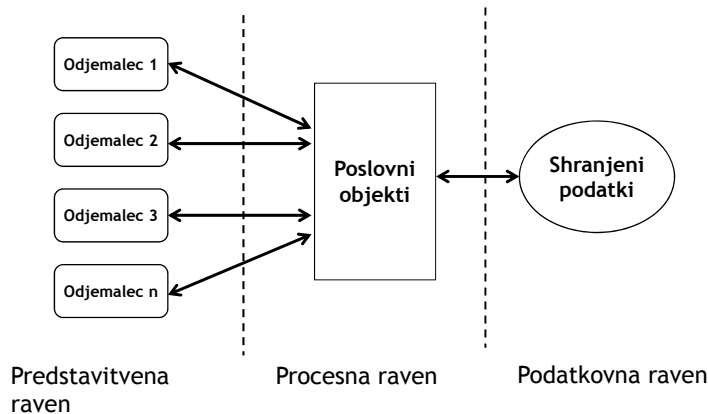
- Slabosti dvo-nivojske arhitekture se pokažejo, če:
 - imamo veliko odjemalcev;
 - je potrebno veliko procesiranja;
- Posledice:
 - Visoki stroški skrbništva, delovanja in vzdrževanja sistema;
 - Visoki stroški vpeljave tehnoloških trendov;
 - Slaba izkoriščenost programske opreme. Tipičen uporabnik v povprečju uporablja cca 10% zmogljivosti programske opreme, ki mu je na voljo.

6.2 Arhitekture odjemalec strežnik

- Tri-nivojska arhitektura (model lahkega odjemalca)
 - pojavi se kot rešitev težav dvo-nivojske arhitekture.
- V tri-nivojski arhitekturi se predstavitveni in procesni nivo ločita. Nivoji so:
 - Predstavitvena raven (vmesnik med uporabnikom in sistemom);
 - Procesna raven (procesiranje, ki ga zahteva sistem); Fizično realizirana na aplikacijskem strežniku.
 - Podatkovna raven (shramba za trajne podatke); Fizično realizirana na podatkovnem strežniku.

6.2 Arhitekture odjemalca strežnika

Tipična tri-nivojska arhitektura



492

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.2 Arhitekture odjemalca strežnika

• Glavne prednosti tri-nivojske arhitekture:

- Neodvisnost aplikacije od tehnologije za shrambo podatkov;
- Procesiranje se seli iz odjemalca na strežnik (cenejša vpeljava, skrbništvo, vzdrževanje in delovanje);
- Procesiranje se nanaša na objekte - skladno z objektno paradigmo;
- Odjemalci lahko uporabljajo podatke različnih podatkovnih virov (na različnih podatkovnih strežnikih),
- Zmanjšanje števila povezav (pri dvo-nivojski arhitekturi $M \times N$ povezav, pri tri-nivojski pa samo $M+N$).
- Večja zanesljivost delovanja sistema;
- Večja prilagodljivost in odprtost sistemov za nove tehnologije;

493

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.2 Arhitekture odjemalec strežnik

- Poleg opisanih arhitektur odjemalec strežnik poznamo tudi več-nivojske arhitekture (podobno kot 3-nivojske, le da imamo dodatne nivoje; tipične za spletne aplikacije, ki se povezujejo z ostalimi poslovnimi IS)

6. Arhitekture porazdeljenih sistemov

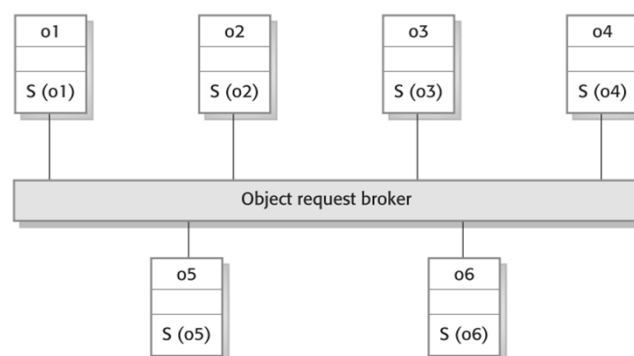
- Vsebina poglavja:
 - 6.1 Uvod
 - 6.2 Arhitekture odjemalec strežnik
 - ⇒ 6.3 Arhitekture porazdeljenih objektov
 - 6.4 Arhitekture P2P
 - 6.5 SOA - Storitveno usmerjena arhitektura

6.3 Arhitekture porazdeljenih objektov

- Arhitekture porazdeljenih objektov ne razlikujejo med odjemalci in strežniki.
- Porazdeljive komponente so objekti, ki zagotavljajo storitve drugim ali uporabljajo storitve drugih objektov.
- Komunikacija med objekti poteka preko vmesnega sloja - ORB (object request broker).

6.3 Arhitekture porazdeljenih objektov

Konceptualna zasnova arhitekture porazdeljenih objektov



6.3 Arhitekture porazdeljenih objektov

- Prednosti:

- Načrtovalcem se ni potrebno že na začetku odločiti kje in kako bodo postavljene posamezne storitve
- Odprta arhitektura, ki dopušča dodajanje novih virov, ko je to potrebno
- Sistem je fleksibilen in prilagodljiv
- Sistem je mogoče dinamično prilagajati z migracijo objektov po omrežju

- Slabosti:

- Arhitekture porazdeljenih objektov so bolj kompleksne za načrtovanje kot arhitekture odjemalec strežnik.

6.3 Arhitekture porazdeljenih objektov

- V preteklosti je bilo razvitih več standardov in tehnologij za podporo porazdeljenih objektov (vmesni sloji), najbolj znana:
 - CORBA (Common Object Request Broker Architecture) - OMG,
 - DCOM (Distributed Component Object Model) - Microsoft,
- Predvidevalo se je, da bodo postali izhodiščni model za ponovno uporabo kode in storitev preko Interneta, vendar:
 - Vrsta težav pri delovanju preko požarnih zidov in pri delovanju na anonimnih računalnikih, kjer varnost ni zagotovljena
 - Posledično se je namesto CORBA in DCOM uveljavila kombinacija HTTP zahtev in spletnih brskalnikov (spletne storitve, naknadno še SOA)
- Omenjene standarde danes nadomeščajo standardi in tehnologije, ki temeljijo na spletnih storitvah in SOA.

6. Arhitekture porazdeljenih sistemov

- Vsebina poglavja:
 - 6.1 Uvod
 - 6.2 Arhitekture odjemalec strežnik
 - 6.3 Arhitekture porazdeljenih objektov
 - ⇒ 6.4 Arhitekture P2P
 - 6.5 SOA - Storitveno usmerjena arhitektura

6.4 Arhitekture P2P

- Sistemi P2P so decentralizirani sistemi v katerih lahko procese izvaja katerokoli vozlišče v omrežju
- Sistemi so zasnovani tako, da izkoriščajo prednosti računske moči velikega števila računalnikov povezanih v omrežje
- Čeprav se velik del P2P sistemov uporablja za zasebno rabo, pa je model uporablja tudi za poslovne sisteme

6. Arhitekture porazdeljenih sistemov

- Vsebina poglavja:

- 6.1 Uvod
- 6.2 Arhitekture odjemalec strežnik
- 6.3 Arhitekture porazdeljenih objektov
- 6.4 Arhitektura P2P

⇒ 6.5 SOA - Storitveno usmerjena arhitektura

6.5 Storitveno usmerjena arhitektura (SOA)

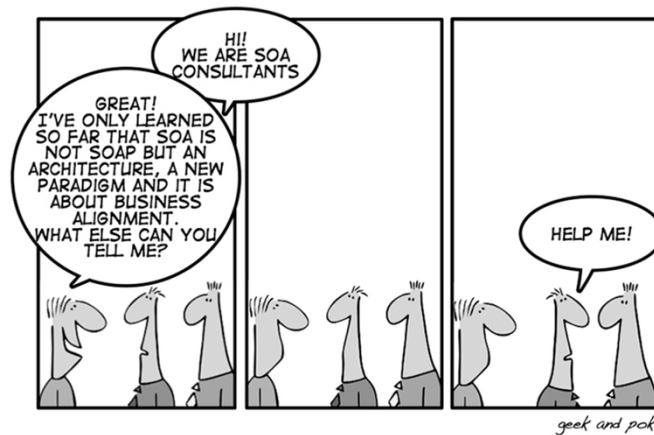
- Vsebina:

⇒ Kaj je SOA in glavna načela storitvene usmerjenosti

- Temeljni pojmi SOA
- Tehnološko ozadje
- SOA kot storitveni nivo
- Življenjski cikel SOA
- Primernost

6.5 Storitveno usmerjena arhitektura (SOA)

- Kaj je SOA?



504

<http://geekandpoke.typepad.com>



Čunalništvo in informatika
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Storitveno usmerjena arhitektura (SOA) temelji na ohlapno povezanih sistemih, združenih v celoto, pri čemer so posamezni deli med seboj neodvisni in tečejo na poljubnih platformah.
- Gre za koncept, ki je zaradi težav s kompleksnimi sistemi, za katere značilen problem so visoki stroški povezovanja in vzdrževanja, v zadnjih letih postal zelo popularen.
- Osnovni gradniki storitveno usmerjene arhitekture so seveda storitve.

505

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Glavna načela storitvene usmerjenosti so (1/2):
 - Šibka sklopljenost - storitve ohranjajo odnose, ki minimizirajo odvisnosti med njimi in ohranjajo le zavedanje ena druge.
 - Storitvena pogodba - storitve se držijo komunikacijskega dogovora, ki je določen z enim ali več opisom storitev in podobnih dokumentov.
 - Neodvisnost - storitev je neodvisna od drugih storitev, in sicer v smislu nadzora nad svojo logiko.
 - Abstrakcija - razen opisa storitve je logika storitve nedostopna zunanjemu svetu.

6.5 Storitveno usmerjena arhitektura (SOA)

- Glavna načela storitvene usmerjenosti so (2/2):
 - Ponovna uporaba - logika je razdeljena, ločena oziroma razbita v različne storitve z namenom možnosti ponovne uporabe.
 - Storitve minimizirajo količino informacij, ki pripada določeni aktivnosti - so brez stanja.
 - Odkrivanje - storitve so načrtovane tako, da jih lahko opišemo in najdemo; tako lahko do njih dostopamo preko temu namenjenih mehanizmov.

6.5 Storitveno usmerjena arhitektura (SOA)

- Vsebina:

- Kaj je SOA in glavna načela storitvene usmerjenosti

- ⇒ Temeljni pojmi SOA

- Tehnološko ozadje
- SOA kot storitveni nivo
- Življenjski cikel SOA
- Primernost

6.5 Storitveno usmerjena arhitektura (SOA)

- Temeljni pojmi SOA:

- Storitve
- Opis storitve
- Sporočilo

6.5 Storitveno usmerjena arhitektura (SOA)

• Storitev (1/2):

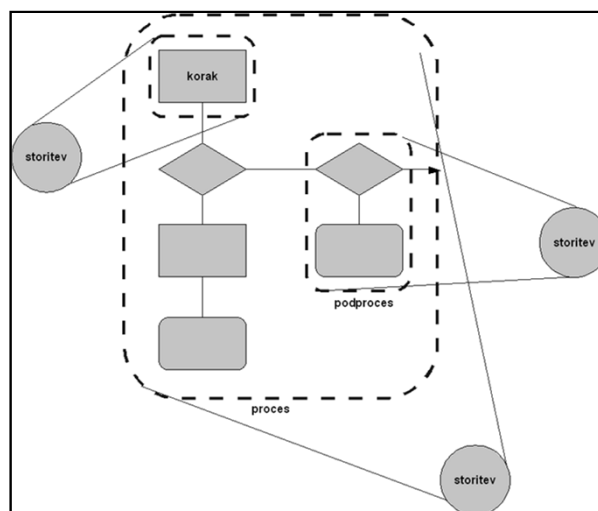
- Storitev je ponovljivo poslovno opravilo, npr. preveri stanje na računu, odpri nov račun.
- Storitvi pravimo tudi storitveno usmerjena logična enota.
- Da ohranjajo neodvisnost, storitve zajemajo logiko znotraj točno določenega konteksta. Ta kontekst je lahko specifičen za poslovno opravilo, poslovno entiteto ali kakšen drug logičen skupek.
- Logika, ki jo storitev vsebuje, je namenjena za reševanje določenega problema, katerega kompleksnost je lahko zelo različna.

510

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

• Storitev (2/2):



511

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Opis storitve (1/2):

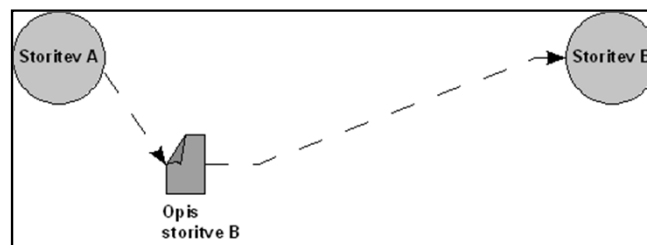
- Znotraj SOA storitve lahko uporabljajo druge storitve ali drugi programi. Da bi storitev lahko uporabili morajo poznati njeno delovanje. Zato je potreben opis storitve.
- V osnovi mora opis storitve zajemati ime storitve, podatke, ki jih pričakuje na vходу ter podatke, ki jih vrne na izhodu.
- Z uporabo koncepta opisa storitev dosežemo šibko sklopljenost sistema.

512

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Opis storitve (2/2):



513

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Sporočilo:
 - Sporočilo je neodvisna enote komunikacije.
 - Ko storitev pošlje sporočilo, nad njim zgubi nadzor (samoobvladovanje delov procesne logike).
 - Sporočilo lahko obravnavamo kot način komunikacije, ki ohranja šibko sklopljenost.

6.5 Storitveno usmerjena arhitektura (SOA)

- Vsebina:
 - Kaj je SOA in glavna načela storitvene usmerjenosti
 - Temeljni pojmi SOA
 - ⇒ Tehnološko ozadje
 - SOA kot storitveni nivo
 - Življenjski cikel SOA
 - Primernost

6.5 Storitveno usmerjena arhitektura (SOA)

- Čeprav SOA ne predpisuje uporabe konkretnih tehnologij, pa večina spletnih storitev danes temelji na:
 - WSDL (Web Service Description Language) - podaja opis storitve
 - SOAP - format sporočanja med storitvijo in uporabnikom
 - UDDI - standardiziran format za registracijo storitev

6.5 Storitveno usmerjena arhitektura (SOA)

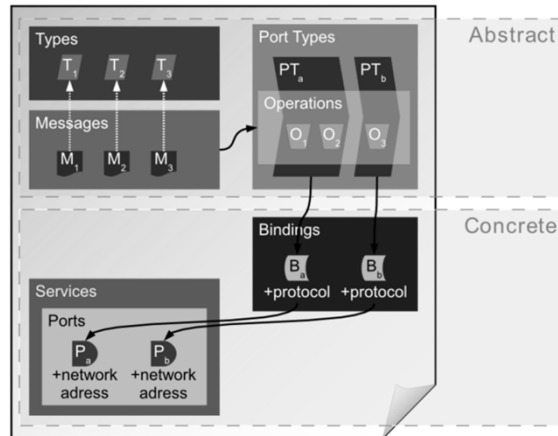
WSDL

- Kaj je WSDL (1/2):
 - opredeljuje spletne storitve kot zbirke vrat (port, endpoint).
 - storitve opisuje v formatu XML
 - povezuje konkretno izvedbo storitve z njeno abstraktno definicijo
 - definicija opredeljuje:
 - abstraktne tipe
 - abstraktne vmesnike (klici operacij oz. sporočila)
 - tipe vrat v katerih so združeni abstraktni vmesniki
 - povezave med tipi vrat in konkretnimi storitvami

6.5 Storitveno usmerjena arhitektura (SOA)

WSDL

- Kaj je WSDL (2/2):



518

računalništvo in informatika
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

WSDL

- Primer WSDL (1/3): abstraktni tipi

```
<!-- Abstract types -->
<types>
  <xs:schema xmlns="http://www.example.com/wsd120sample"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.example.com/wsd120sample">

    <xs:element name="request">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="header" maxOccurs="unbounded">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="name" type="xs:string" use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="body" type="xs:anyType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="method" type="xs:string" use="required"/>
        <xs:attribute name="uri" type="xs:anyURI" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</types>
```

519

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

WSDL

- Primer WSDL (2/3): abstraktni vmesniki

```
<!-- Abstract interfaces -->
<interface name="RESTfulInterface">
  <fault name="ClientError" element="tns:response"/>
  <fault name="ServerError" element="tns:response"/>
  <fault name="Redirection" element="tns:response"/>
  <operation name="Get" pattern="http://www.w3.org/ns/wsdli/in-out">
    <input messageLabel="GetMsg" element="tns:request"/>
    <output messageLabel="SuccessfulMsg" element="tns:response"/>
  </operation>
  <operation name="Post" pattern="http://www.w3.org/ns/wsdli/in-out">
    <input messageLabel="PostMsg" element="tns:request"/>
    <output messageLabel="SuccessfulMsg" element="tns:response"/>
  </operation>
  <operation name="Put" pattern="http://www.w3.org/ns/wsdli/in-out">
    <input messageLabel="PutMsg" element="tns:request"/>
    <output messageLabel="SuccessfulMsg" element="tns:response"/>
  </operation>
  <operation name="Delete" pattern="http://www.w3.org/ns/wsdli/in-out">
    <input messageLabel="DeleteMsg" element="tns:request"/>
    <output messageLabel="SuccessfulMsg" element="tns:response"/>
  </operation>
</interface>
```

520

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

WSDL

- Primer WSDL (3/3): povezava s konkretno storitvijo in vrata

```
<!-- Concrete Binding with SOAP-->
<binding name="RESTfulInterfaceSoapBinding" interface="tns:RESTfulInterface"
  type="http://www.w3.org/ns/wsdli/soap"
  wssoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/"
  wssoap:mepDefault="http://www.w3.org/2003/05/soap/mep/request-response">
  <operation ref="tns:Get" />
  <operation ref="tns:Post" />
  <operation ref="tns:Put" />
  <operation ref="tns>Delete" />
</binding>

<!-- Web Service offering endpoints for both the bindings-->
<service name="RESTfulService" interface="tns:RESTfulInterface">
  <endpoint name="RESTfulServiceHttpEndpoint"
    binding="tns:RESTfulInterfaceHttpBinding"
    address="http://www.example.com/rest/" />
  <endpoint name="RESTfulServiceSoapEndpoint"
    binding="tns:RESTfulInterfaceSoapBinding"
    address="http://www.example.com/soap/" />
</service>
```

521

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Kaj je SOAP (1/2)?

- včasih: "Simple Object Access Protocol"; od različice 1.2 le še SOAP
- SOAP je specifikacija protokola za izmenjavo strukturiranih podatkov pri implementaciji spletnih storitev v računalniških omrežjih oz. format sporočanja med storitvijo in uporabnikom storitve.
- Sporočila so zapisana v obliki XML.
- SOAP lahko uporablja različne transportne protokole. Tipično se uporablja HTTP, vendar je mogoče uporabiti tudi druge (npr. SMTP).

6.5 Storitveno usmerjena arhitektura (SOA)

- Kaj je SOAP (2/2)?

- primer uporabe SOAP: "Sporočilo v obliki SOAP pošljemo določeni spletni storitvi (npr. vrednost delnic na borzi) skupaj s parametri za iskanje. Spletna storitev vrne odgovor v obliki XML (npr. vrednosti delnic).

6.5 Storitveno usmerjena arhitektura (SOA)

• Primer SOAP (1/2): zahteva

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?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.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>

</soap:Envelope>
```

6.5 Storitveno usmerjena arhitektura (SOA)

• Primer SOAP (2/2): odgovor

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?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.example.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>

</soap:Envelope>
```

6.5 Storitveno usmerjena arhitektura (SOA)

UDDI

- Kaj je UDDI?

- “Universal Description Discovery and Integration”
- Omogoča objavlanje seznamov storitev ter iskanje ustreznih storitev po teh seznamih.
- UDDI je posebna vrsta spletne storitve, ki upravlja s podatki o ponudnikih storitev

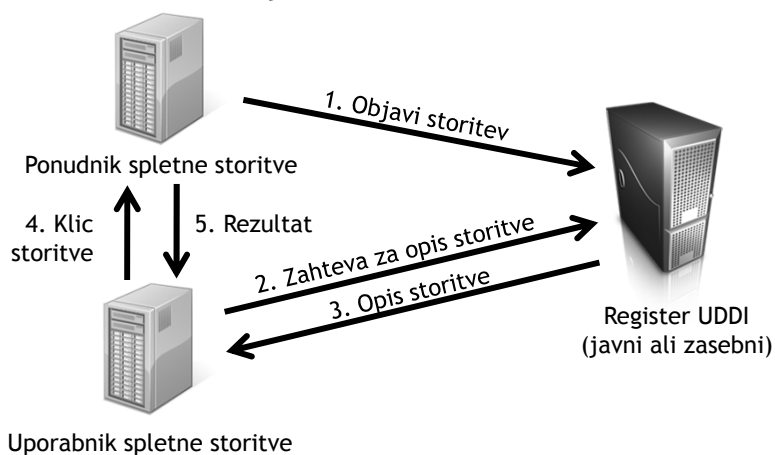
526

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

UDDI

- Kako UDDI deluje?



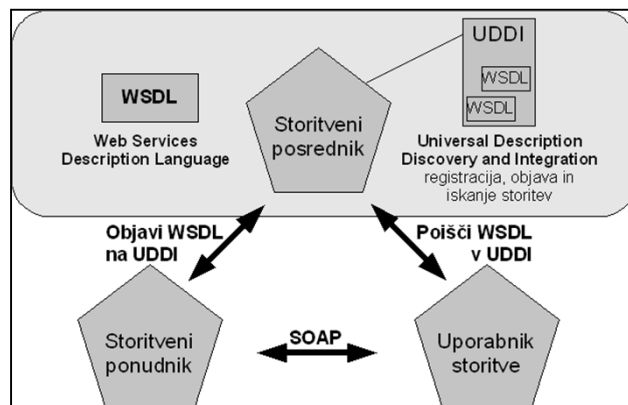
527

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

UDDI

- WSDL, SOAP, UDDI in spletne storitve



528

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Vsebina:
 - Kaj je SOA in glavna načela storitvene usmerjenosti
 - Temeljni pojmi SOA
 - Tehnološko ozadje
- ⇒ SOA kot storitveni nivo
 - Življenjski cikel SOA
 - Primernost

529

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Osnovne karakteristike SOA (1/2):

- Enkapsulacija
 - Implementacija strežnika je skrita odjemalcu, saj strežnik objavi zgolj vmesnik, tako da je uporabniku podrobnost implementacije skrita.
- Strežnik ni vezan na fizično lokacijo
 - Strežnik se prijavi v imenik storitev, katero uporabnik povpraša o mrežnem naslovu storitve.
 - Če strežnik zamenja fizično lokacijo, se, ob ponovni prijavi v imenik, ažurira mrežni naslov in uporabniku ni potrebno pomniti naslova storitve

530

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Osnovne karakteristike SOA (2/2):

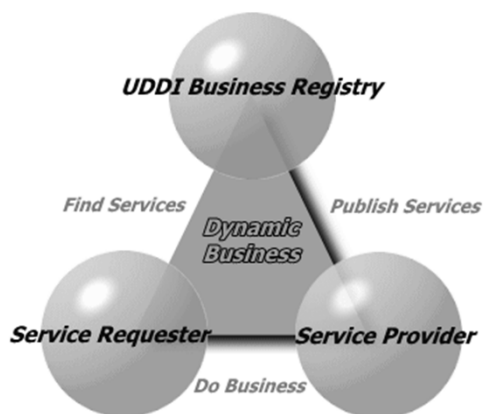
- Skupno delovanje aplikacij
 - Aplikacije, ki se izvajajo na različnih platformah, skupno delujejo tako, da vsaka aplikacija izpostavi vmesnik z uporabo standardnega protokola.

531

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Enostavna ideja SOA:

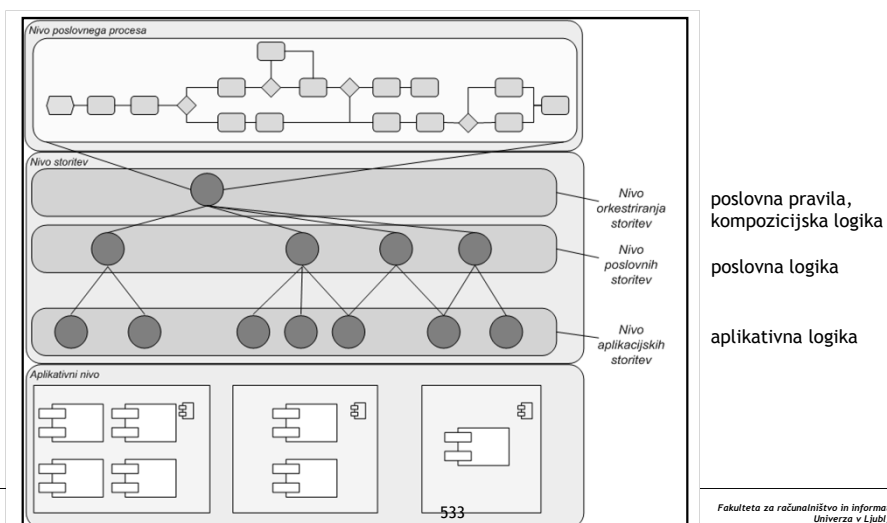


532

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

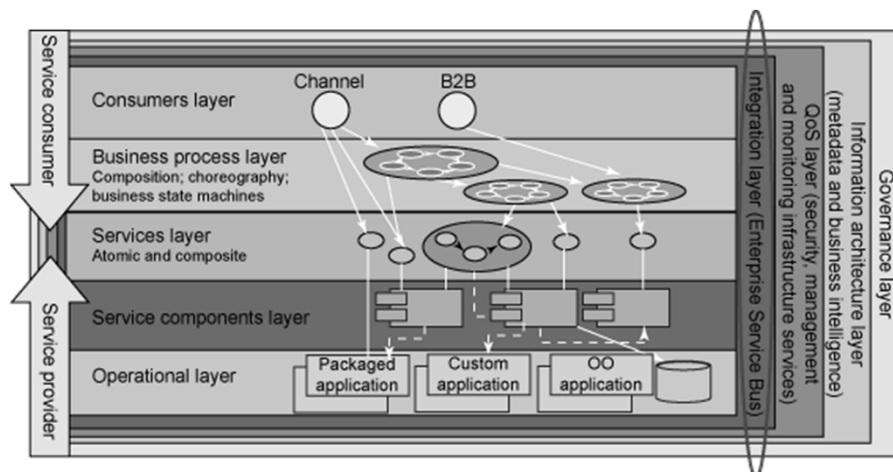
- SOA kot storitveni nivo med poslovnim in aplikativnim nivojem



Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

Primer - Enterprise Service Bus (IBM)



534

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

• Vsebina:

- Kaj je SOA in glavna načela storitvene usmerjenosti
- Temeljni pojmi SOA
- Tehnološko ozadje
- SOA kot storitveni nivo

⇒ Življenjski cikel SOA

- Primernost

535

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- SOA je IT arhitekturni stil, ki podpira storitveno usmerjenost. (IBM)
- Življenski cikel SOA lahko razdelimo v naslednje faze:
 - modeliranje
 - sestavljanje
 - razvoj
 - upravljanje



536

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Modeliranje:
 - Faza modeliranja se začne z analizo poslovnih zahtev ter simulacijo in optimizacijo PP.
 - Nastali PP so uporabljeni za načrtovanje storitev.
 - Z razvitim modelom dosežemo skupno razumevanje PP, ciljev in rezultatov med poslovnimi in IT uporabniki.
 - Model predstavlja osnovo za merjenje poslovne učinkovitosti.



537

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

• Sestavljanje:

- Pri fazi sestavljanja zgradimo storitve iz obstoječih sistemov (ERP, finančni sistemi, itd.).
- S pomočjo knjižnice obstoječih storitev lahko poiščemo in opredelimo storitve v organizaciji, v primeru da le ta ne obstaja, pa jo lahko zgradimo na novo.
- Ko pripravi vse ustrezne storitve jih s pomočjo koreografije združimo v poslovni proces.



538

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

• Razvoj:

- V tej fazi se ukvarjamo z izvajalnim okoljem, kjer se PP izvajajo.
- Ko je PP oblikovan, ga lahko objavimo.
- Izvajalno okolje je optimizirano za izvajanje ključnih PP ter je hkrati prilagodljivo in deluje dinamično.
- S storitveno usmerjenim pristopom prav tako zmanjšamo ceno in kompleksnost, povezano z vzdrževanjem številnih samostojnih integracij.



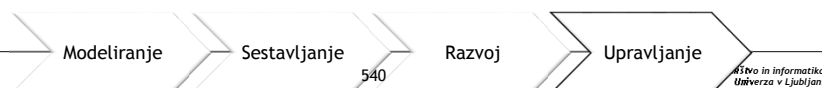
539

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

• Upravljanje:

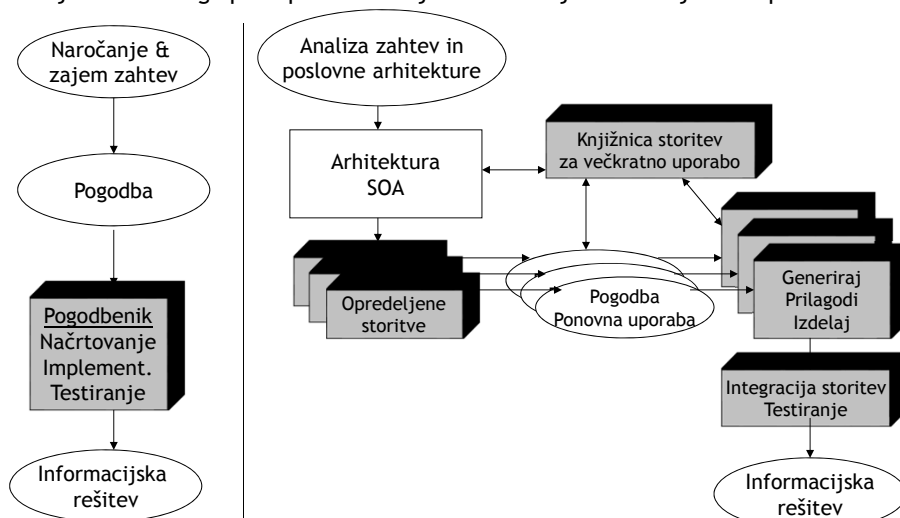
- Faza vključuje obvladovanje razpoložljivosti storitve in njen odzivni čas, kot tudi obvladovanje virov, ki jih uporablja.
- S kontroliranjem kritičnih kazalcev (KPI) v realnem času lahko pravočasno preprečimo, izoliramo ali odpravimo probleme.
- V tej fazi se ukvarjamo tudi z vzdrževanjem verzij procesov.
- Ključna prednost faze je v izboljšani podpori sprejemanja poslovnih odločitev, ki jih lahko sprejmemo prej kot smo to lahko storili predhodno.



Školo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

Primerjava klasičnega postopka naročanja IR in razvoja/naročanja IR z uporabo SOA



541

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Vsebina:

- Kaj je SOA in glavna načela storitvene usmerjenosti
- Temeljni pojmi SOA
- Tehnološko ozadje
- SOA kot storitveni nivo
- Življenjski cikel SOA

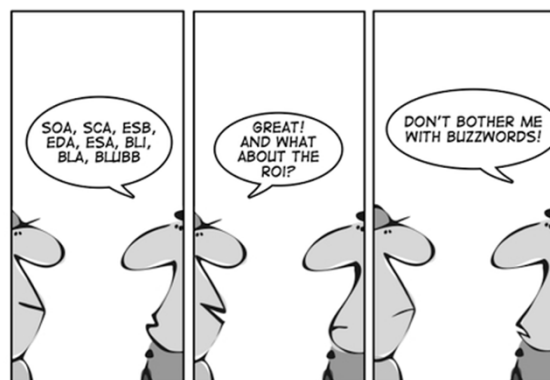
⇒ Primernost

542

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

- Primernost uporabe SOA?



GEEK AND POKE

<http://geekandpoke.typepad.com>



543

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

6.5 Storitveno usmerjena arhitektura (SOA)

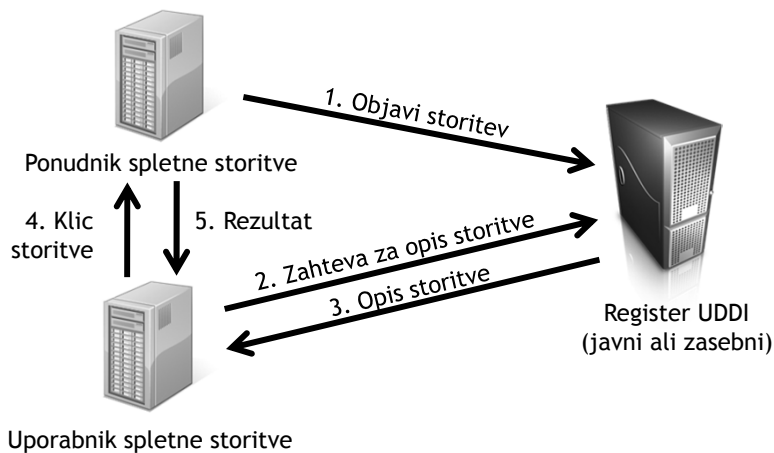
- SOA ni generična rešitev za vse problemske domene, npr:
 - V stabilnem (nespremenljivem) okolju ponavadi cena vpeljave presega učinkovitost naložbe.
 - Ni potrebe po ponujanju aplikacij v obliki storitev zunanjim poslovnim partnerjem.
 - Realnočasovni sistemi niso primerni za SOA, saj le ta temelji na ohlapno povezani asinhronski komunikaciji, ki ne zagotavlja najhitrejšega odziva.

Ponovitev

- Kakšna je vloga WSDL?
 - Znotraj SOA storitve lahko uporabljajo druge storitve ali drugi programi. Da bi storitev lahko uporabili morajo poznati njeno delovanje. Zato je potreben opis storitve. WSDL je konkretna tehnološka rešitev za opis storitve.

Ponovitev

- Kako deluje UDDI?



546

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

Ponovitev

- V kateri fazi življenjskega cikla SOA začnemo z izgradnjo storitev iz obstoječih sistemov (ERP, itd.)?
 - V fazi sestavljanja
- Zakaj je uporaba SOA v stabilnem okolju tipično manj primerna?
 - V stabilnem (nespremenljivem) okolju navadno cena vpeljave presega učinkovitost naložbe.

547

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani