

UVOD

PROIZVOD je materijalno dobro koje je izgradio čovjek, stroj ili je nastala prirodnim procesom.

USLUGA je promjena stanja osobe ili dobra u vlasništvu nekog ekonomskog entiteta koja je nastala djelovanjem nekog drugog ekonomskog entiteta uz pristanak primatelja usluge.

Razredi usluga:

- osobne usluge – osobni kontakt, ograničena vremenska i prostorna dostupnost, minimalna uključenost korisnika
- samoposlužne usluge – izravni kontakt, djelomično ograničena vremenska i prostorna dostupnost, srednja uključenost korisnika
- elektroničke usluge – el.isporuka, neograničena dostupnost, visoka uključenost korisnika

RZNU – skup načela koja omogućuju razvoj sustava koji koriste ili ostvaruju računalne usluge:

- upravljane usluge – načela stvaranja samostalnih usluga (max. složenost, min. razvijenost)
- složene usluge – načela stvaranja složenih usluga
- osnovne usluge – načela osnovnog stvaranja i korištenja usluga (min. složenost, max. razvijenost)

Dva načina ostvarivanja usluga:

- REST(RESTFull)
 - o usluga = web stranica
 - o primjenjuju se za ostvarivanje izravnog kontakta s krajnjim korisnicima
 - o obično uključuju samo jednu tvrtku
- web services (WS-*)
 - o standardizirani način ostvarivanja usluga
 - o širi skup mogućih primjena
 - o za objedinjavanje poslovnih sustava

Načela ostvarivanja usluga:

- osobne usluge
 - o standardan jezik poruka (XML)
 - o način razmjene poruka (WS-*: SOAP, HTTP; REST: HTTP)
 - o opis sučelja usluge (WS-*: WSDL; REST: WADL)
 - o registar usluga (WS-*: UDDI; REST: propriatery)

- složene usluge
 - sigurnost (WS-*: security; REST: SSL, TLS)
 - asinkroni pozivi (WS-events, WS-notification; REST: metoda višestrukog dohvata)
 - orkestracija usluga (WS-*: WS-BPEL, SSCL, CL; REST: vlasnička rješenja, mashups)
 - pogled na niz poziva usluga iz perspektive jednog sudionika
 - modelira proces
 - koreografija (WS-*: WS-BPEL, SSCL, CL; REST: vlasnička rješenja, mashups)
 - sveukupni pogled na sljed poziva usluga potreban da se ostvari određena funkcionalnost
 - npr. narudžba -> predračun -> uplata -> račun
 - usluge sa stanjem (WS: WS-RF; REST: vlasnička rješenja)
 - transakcije (WS: WS-Transactions, WS-Coordination; REST: vlasnička rješenja)
- upravljane usluge
 - ugovor o kakvoći usluga (WS: WSLA; REST: vlasnička rješenja)
 - automatska izgradnja (istraživački projekti)
 - obilježja usluga (istraživački projekti)

WADL (Web Application Description Language) je XML jezik koji opisuje način pristupa REST Web uslugama.

RAČUNARSTVO ZASNOVANO NA USLUGAMA

Motivacija:

- programski sustavi namijenjeni čovjeku
 - korisničko sučelje, interakcija čovjek-računalo, unos podataka, prikaz rezultata, upravljanje sustavom, korištenje sustava, web stranice, shell, računalne igre, aplikacije za mobilne uređaje
- programski sustavi namijenjeni računalu
 - programsko sučenje, interakcija računalo-računalo, automatizacija računalnih procesa, višestruko iskorištavanje ostvarenih sustava, razvoj složenih sustava

Modeli razvoja programskih sustava:

- model zasnovan na proceduralnoj paradigmi
- model zasnovan na objektima
- model zasnovan na komponentama
 - višeprocesno ostvarenje; raspodijeljeno izvođenje
- model zasnovan na uslugama
 - višeprocesno ostvarenje, različita tehnološka okruženja izvođenja usluga

USLUGE su samostalne programske jedinice koje putem standardiziranih sučelja omogućuju pristup raznovrsnim funkcionalnostima (IPAK, nema općeprihvaćene definicije) :

- programske jedinice
- samostalne i cjelovite (neovisne o drugim uslugama)
- platformski nezavisne
- formalno definirana programska sučelja (WSDL, WADL, URL)
- otvorena sučelja i komunikacijski protokoli
- dostupne putem računalne mreže

Načela izgradnja usluge:

- slaba povezanost
- kompozicija usluga
 - složeni sustavi ostvaruju se povezivanjem usluga
- opisi usluga
 - značajke usluge bez detalja njezina programskog ostvarenja
 - ostvarenje primjenom formalnih jezika (XML)
 - mogućnosti usluge, pristupna sučelja usluge, ponašajna svojstva usluge, kvaliteta usluge, neovisnost o ostvarenju usluge
- prilagodljivost
 - mogućnost izmjene vlastitih radnih postavki tijekom izvođenja
- krupna zrnatost
 - mali broj usluga složenije funkcionalnosti

Računarstvo zasnovano na uslugama

- smanjenje broja komunikacijskih poruka
 - oblikovanje po uzoru na usluge stvarnog svijeta
 - povećanje višestruke iskoristivosti usluga
- otvoreni standardi
 - neovisnost usluga o računalnim platformama
 - opis usluge, strukture podataka, komunikacijski protokoli
- upravljanje stanjem
 - usluge bez i sa čuvanjem stanja
 - smanjiti ovisnosti usluge o računalnim sredstvima zbog pohrane stanja
 - omogućiti izdvajanje upravljanja stanjem u zasebni sustav

Arhitektura zasnovana na uslugama:

- poslužitelj usluge
 - ostvaruje i postavlja uslugu
 - gradi dokument opisa usluge
 - objavljuje opis usluge u imeniku usluga
- korisnik usluge
 - usmjerava upit imeniku usluga za uslugama
 - prihvća skup opisa usluga od imenika
 - odabire najpovoljniju uslugu
 - koristi odabranu uslugu

Primjena usluga

Klasični programski sustavi:

- sustav se izvodi na računalu korisnika
- sustav kao proizvod
- nedostaci : postavljanje, održavanje, dostupnost, ekonomski modeli, zaštita intelektualnog vlasništva

Programski sustavi kao usluge:

- sustav se izvodi na udaljenom računalu kao usluga
- sustav kao usluga
- prednosti: nema postavljanja, održavanje od strane proizvođača, dostupni su sa bilo kojeg računala, zaštita intelektualnog vlasništva
- ekonomski modeli: besplatne usluge, naplata po korištenju usluge, naplata po vremenskom razdoblju, naplata po kvaliteti usluge

Cloud computing:

- stil računarstva u kojem se računalna sredstva, platforme i aplikacije vizualiziraju i izlažu kao usluge
- načela i svojstva:
 - virtualizacija programskih i sklopovskih sredstava (*virtualization*)
 - automatizirana elastičnost razmjernog rasta (*elastic scalability*)
 - naplata po korištenju usluge (*pay-per-use*)

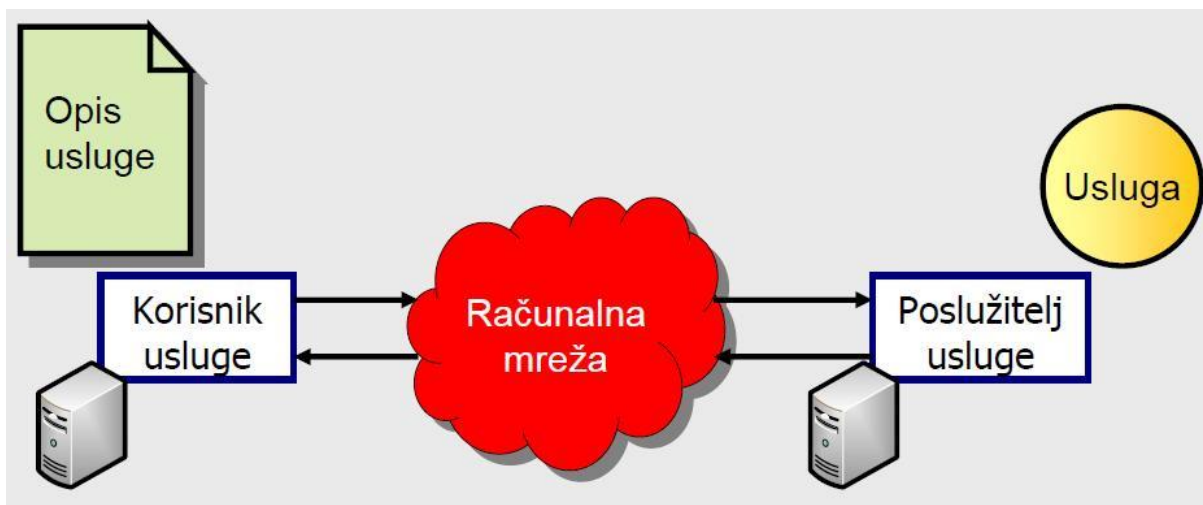
Računalna sredstva kao usluge:

- virtualizacija osnovnih računalnih i mrežnih sredstava:
 - izvođenje programskih sustava (*computing*)
 - pohrana podataka (*storage*)
 - slanje i primanje podataka mrežom (*networking*)
 - posredničke usluge (*middleware services*)

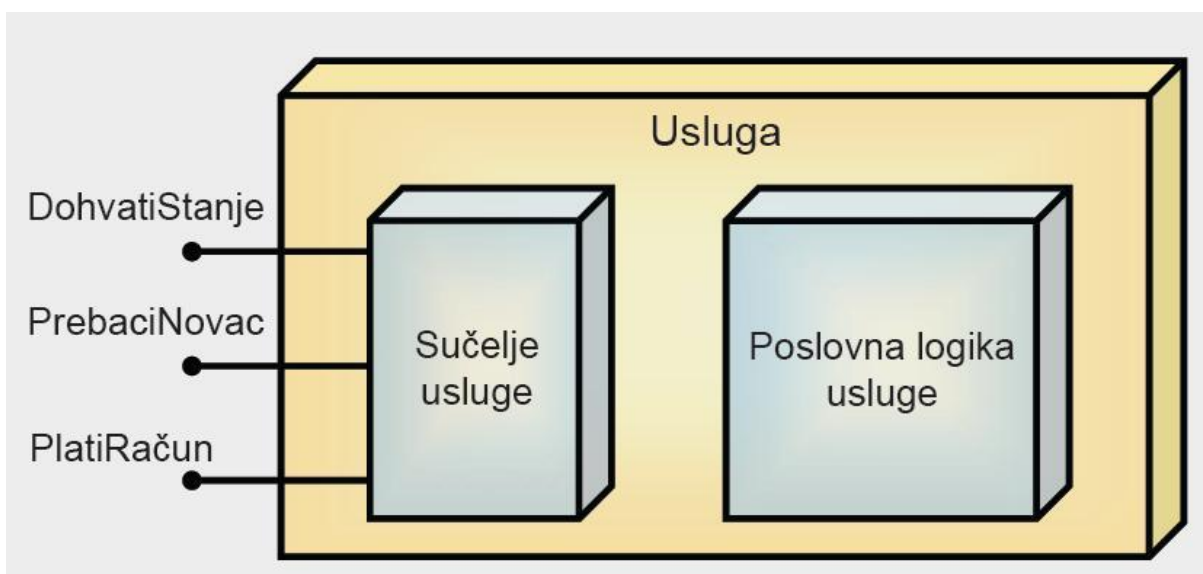
Računalne okoline kao usluge:

- programska okolina za izgradnju i izvođenje primjenskih sustava (programski model; biblioteka usluga za transparentno korištenje računalnih sredstava)

NAČELA IZGRADNJE I KORIŠTENJA USLUGA



Slika 1 Prikaz korištenja usluge



Slika 2 Sadržaj usluge

- sučelje usluge (*server stub*) :
 - deserijalizacija poruke zahtjeva
 - poziv operacije
 - serijalizacija poruke odgovora

Računarstvo zasnovano na uslugama

Jezik za izgradnju opisa sučelja:

- konzistentan opis sučelja svih usluga (formalni jezici; opis strukture poruka)
- raznorodni sustavi (tekstualni jezici; XML)

Jezik za izgradnju poruka:

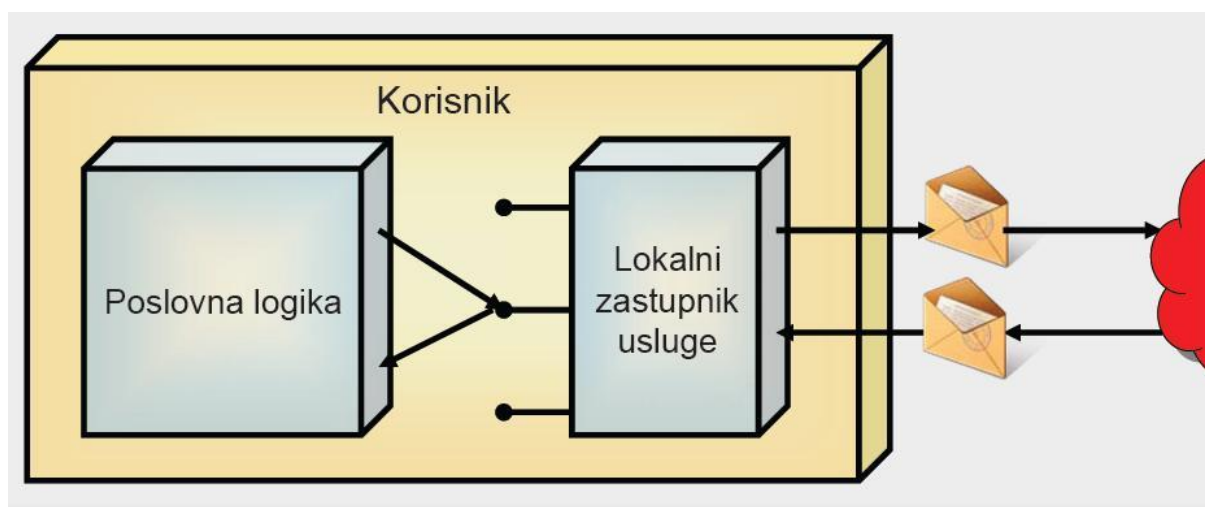
- zapis poruka u skladu s opisom sučelja (formalni jezici; serijalizacija i deserijalizacija)
- raznorodni sustavi (tekstualni jezici; XML)

Poslužitelj:

- pristup usluzi putem mreže
- poziv sučelja usluge
- protokol za razmjenu poruka i prijenos podataka mrežom

Okolina za razvoj i izvođenje usluga:

- razvoj poslovne logike usluge
- automatizirana izgradnja sučelja usluge
- automatizirano povezivanje s poslužiteljem
- potpora izvođenju usluge



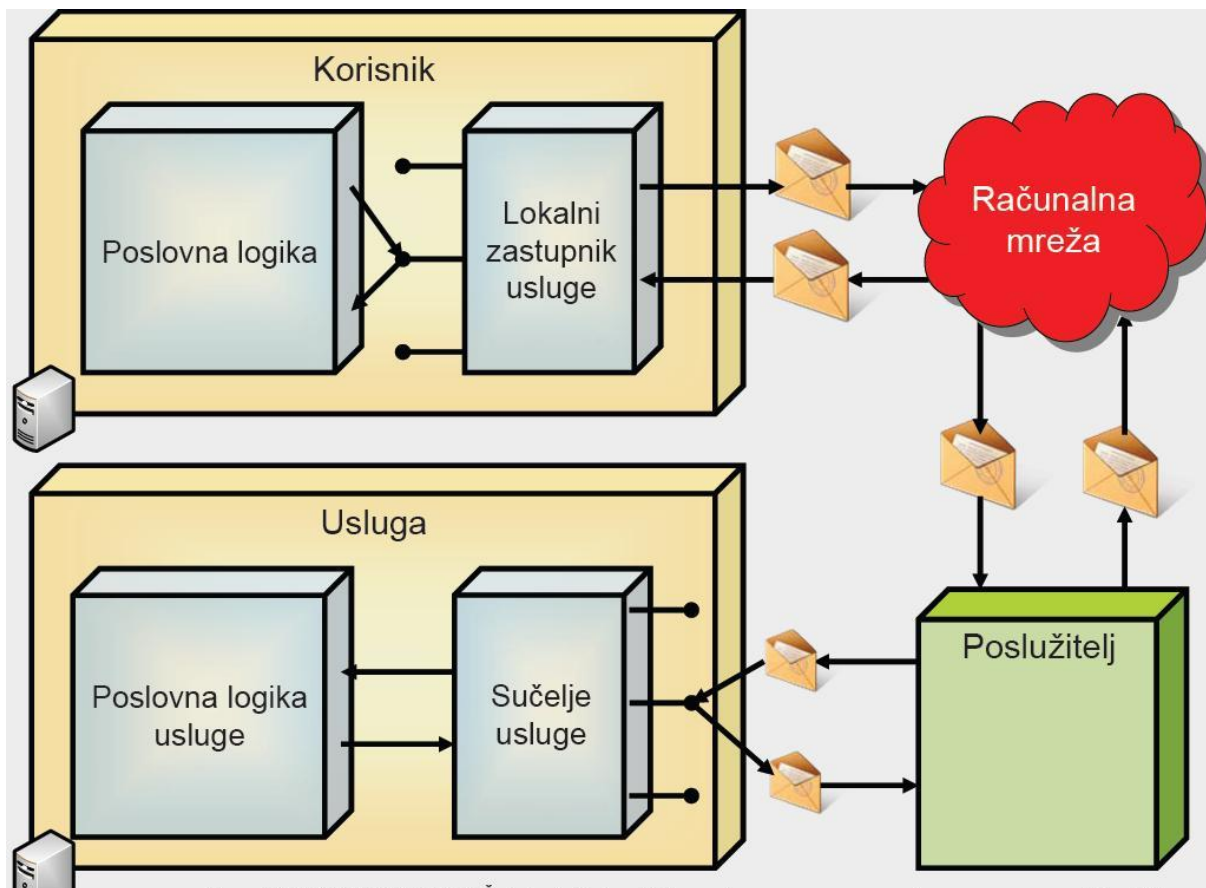
Slika 3 Izgled korisnika

Lokalni zastupnik usluge (*client stub*):

- stvaranje poruke zahtjeva (serijalizacija)
- slanje zahtjeva usluzi putem mreže
- prihvaćanje odgovora (deserijalizacija)

Okolina za razvoj korisnika usluga:

- razvoj poslovne logike programa korisnika
- automatizirana izgradnja lokalnog zastupnika usluge
- potpora izvođenju



Slika 4 Komunikacija usluga-korisnik

Mrežne usluge

Osnovni standardi izgradnje mrežnih usluga:

- standard URL (adresiranje)
- protokol HTTP (prijenos podataka)
- jezik XML (zapis podataka)

HTTP zahtjev	Linija zahtjeva	Operacija	Sredstvo	Verzija
	Zaglavlja	Parovi "ključ: vrijednost"		
	Prazna linija	Prazna linija		
	Tijelo poruke	Tijelo poruke		

```
POST /index.html HTTP 1.1
```

```
Host: www.example.com
```

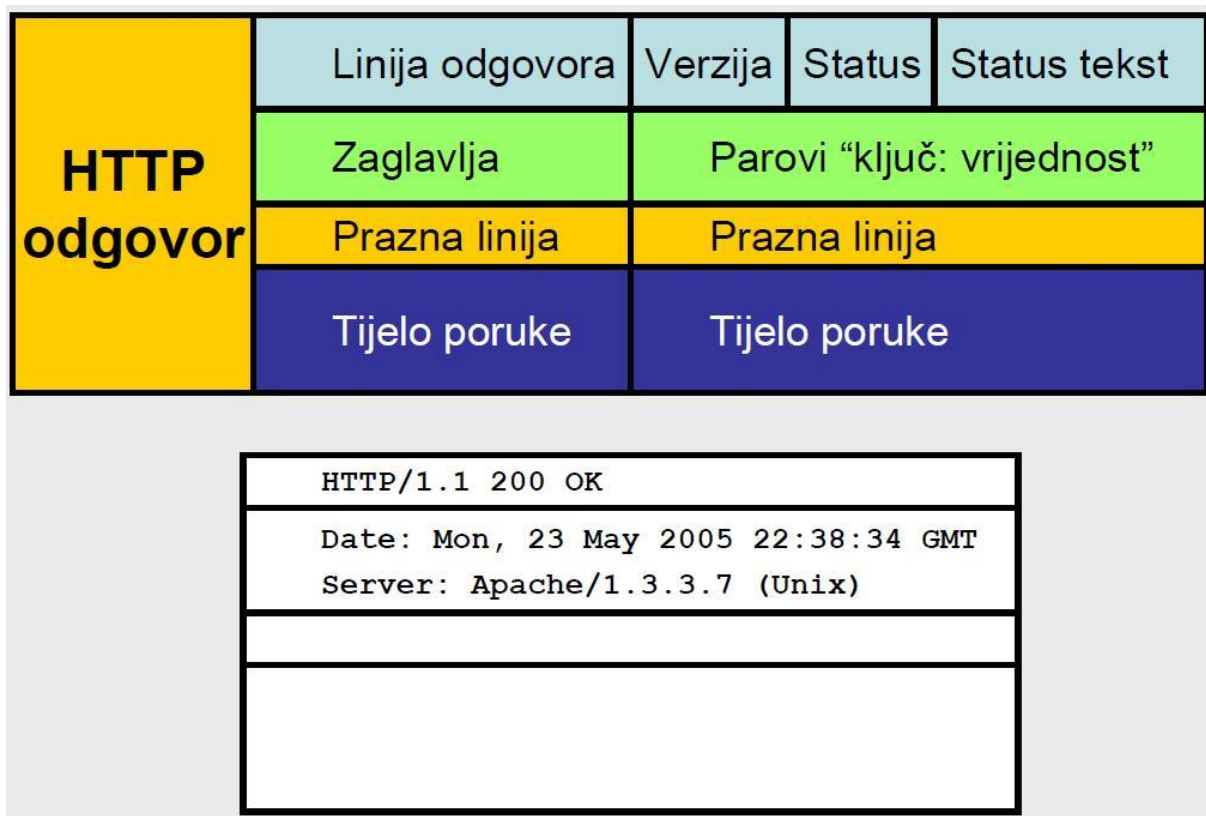
```
Content-Type: text/html;
```

```
<html> <head></head>
```

```
<body> Hello world! </body>
```

```
</html>
```

Slika 5 HTTP zahtjev



Slika 6 HTTP odgovor

XML + XSD

JSON zapis podataka:

- tekstualni formalni zapis opće namjene (strojno obradiv, razumljiv ljudima)
- platformski nezavisan (primjena u raznorodnim sustavima)
- struktura u obliku objekata i polja
- ne toliko čitljiv čovjeku kao XML, puno lakši i jednostavniji od XML-a
- standardne strukture podataka

JSON vs XML:

- JSON bolji za razmjenu podataka (data oriented)
- XML bolji za razmjenu dokumenata (document oriented)

REST (Representational State Transfer)

Teorijski model REST

Teorijski model programske arhitekture; raspodijeljeni sustavi; nastao iz WWW.

Osnovni elementi arhitekture REST:

- data elements (podaci)
 - o tri komponente podataka
 - sredstvo – usluga dostupna na webu
 - adresa – URI na kojoj je sredstvo dostupno
 - prikaz – kao HTML u browseru
- connectors (sučelja)
 - o služe za razmjenu podataka
 - o osnovni tipovi: klijent i poslužitelj
 - o sučelje komponenti daje uslugu, komponenta može imati oba sučelja!
 - o ostale vrste sučelja: priručna memorija, razrješitelj (resolver)
- components (komponente)
 - o korisnički agent (user agent), izvorni poslužitelj (origin server), posrednik (proxy)

Ograničenja na web:

- klijent/poslužitelj – podjela zadaća (nezavisan razvoj klijenta i poslužitelja)
- poslužitelj bez pohrane stanja
- priručna memorija na klijentu kao opcija
- standardizirana sučelja
- slojevitost komponenti
- izvršenje koda na zahtjev

Zaključak: REST koristi određen skup metoda; kao apstraktni model ne predviđa koje su to metode; može biti ostvaren mnoštvom različitih tehnologija i protokola.

RESTFull HTTP

Svako sredstvo mora imati jedinstveni identifikator – URI na webu.

SCHEMA	HOST	PATH	QUERY	FRAGMENT
http://	www.domain.com	/name/of/source	?limit=10&off=0	#bookmark

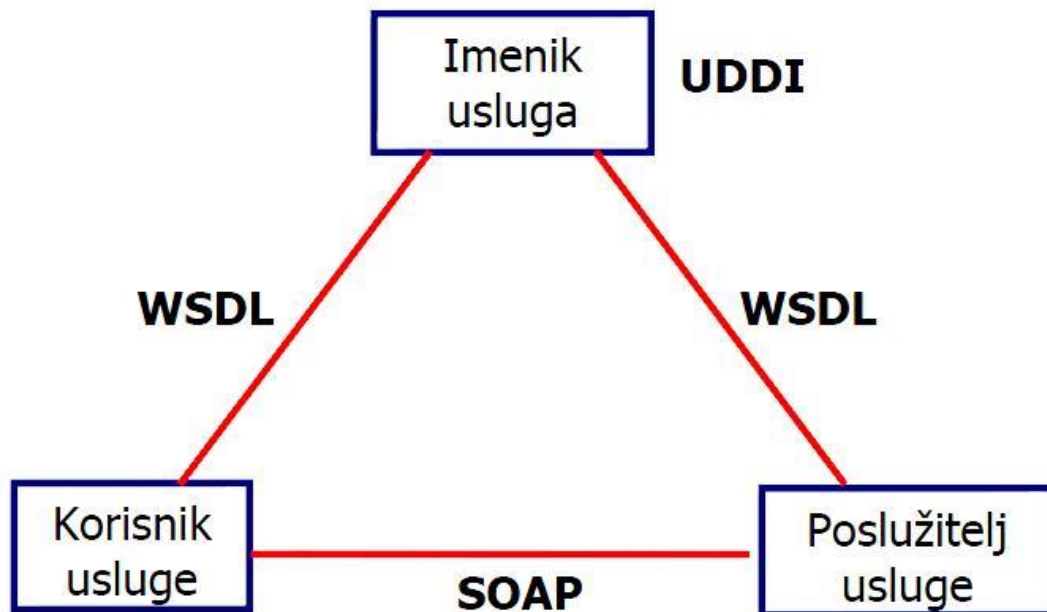
HTTP metode:

- GET – za dohvat podataka, „safe method“, idempotentna metoda
- PUT – za osvježavanje podataka, nije „safe method“, idempotentna i atomarna
- DELETE – za brisanje podataka, nije „safe method“, idempotentna i atomarna metoda
- POST – za stvaranje podataka, „Swiss army knife“, „Side effect method“, nije idempotentna
- analogija sa bazama (CRUD – Create, Read, Update, Destroy)
- HEAD – odgovor bez tijela odgovora; korisno za dohvat metapodataka

- TRACE – poslužitelj vrati zahtjev koji je primio; za praćenje proxya
- OPTIONS – poslužitelj vrati skup podržanih metoda za zadani URL

WS-*

Osnovna arhitektura (Web Services, WS-*)



WS-* je skup specifikacija koje se odnose na web usluge.

Radni okvir za razmjenu poruka

U najnižim dijelovima Web Services stoga.

Osnovni standardi i tehnologije za razmjenu poruka:

- XML – format za razmjenu podataka
- SOAP – standardan način *zapisa* poruka (SOAP poruka je XML dokument)
- WS-Addressing – ostvaruje identifikaciju pošaljitelja i primatelja poruka

XSD

Primjer XML-a:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<shiporder orderid="889923" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="shiporder.xsd">
  <orderperson>John Smith</orderperson>
  <shipto>
    <name>Ola Nordmann</name>
    <address>Langgt 23</address>
    <city>4000 Stavanger</city>
    <country>Norway</country>
  </shipto>
  <item>
    <title>Empire Burlesque</title>
    <note>Special Edition</note>
    <quantity>1</quantity>
    <price>10.90</price>
  </item>
  <item>
    <title>Hide your heart</title>
    <quantity>1</quantity>
    <price>9.90</price>
  </item>
</shiporder>
```

Pripadajući XSD:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="shiporder">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="orderperson" type="xs:string"/>
        <xs:element name="shipto">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="address" type="xs:string"/>
              <xs:element name="city" type="xs:string"/>
              <xs:element name="country" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="item">
          <xs:complexType>
            <xs:sequence>
```

```

    <xs:element name="title" type="xs:string"/>
    <xs:element name="note" type="xs:string" minOccurs="0"/>
    <xs:element name="quantity" type="xs:positiveInteger"/>
    <xs:element name="price" type="xs:decimal"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="orderid" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:schema>

```

SOAP

Komunikacijski protokol; format za razmjenu datoteka zasnovan na XML-u (neovisan o jeziku i platformi).

SOAP poruka:

- osnovna komunikacijska jedinica među SOAP čvorovima
- sastoji se od *omotnice* koja može imati više *zaglavlja* – omotnica mora imati *tijelo* koje sadrži skup korisničkih informacija
- SOAP čvorovi primaju i šalju SOAP poruke (poruke prolaze kroz više čvorova)
- blok zaglavlja predstavlja specifikaciju obrade
- usmjeravanje: dodati standardni blok za usmjeravanje unutar zaglavlja

SOAP programski modeli:

- RPC (Remote Procedure Call)
 - o koriste se procedure napisane u nekom jeziku – zahtjev sadržava ime procedure i skup parametara, a odgovor sadržava povratnu vrijednost procedure
- EDI (Electronic Document Interchange)
 - o zasnovan na dokumentima
 - o osnova za automatiziranje poslovnih transakcija

SOAP Encoding – serijalizacija podataka; preporuka, ali nije obavezno korištenje (slično XSD).

SOAP Binding:

- serijalizacija poruke u prijenosni sloj; moguće korištenje više mrežnih protokola
- najpopularniji HTTP (firewall, robusna infrastruktura, jednostavan, otvoren...)

WSDL (Web Services Description Language)

Opis programskog sučelja Web usluge.

Osnovni elementi:

- operation – metode, funkcije
- message – tipizirani podaci koji su ulazi ili izlazi operacija
- types – XSD tipovi podataka

- portType – skup operacija (sučelje)
- binding – povezuje portType s protokolom
- port – krajnja točka
- service – skup portova

WS-Addressing

Opisuje mehanizme koji osiguravaju ispravnu isporuku poruka prema odgovarajućim krajnjim točkama usluga.

Osnovni koncepti – krajnje točke usluga; WS-Addressing zaglavlja.

Krajnje točke usluga – REFERENCA na krajnju točku (različito od WSDL port strukture!).

Adresa, skup značajki krajnje točke i parametri krajnje točke.

DVIJE KRAJNJE TOČKE SU JEDNAKE AKO IMAJU ISTU ADRESU I SKUP ZNAČAJKI.

Dva obavezna zaglavlja: „To“ (krajnja točka destinacije) i „Action“ (URI koji predstavlja semantiku poruke).