Servisi

UDIS 2011/2012

Servis

- Servisom, u najširem smislu, možemo nazvati aplikaciju koja je dizajnirana i realizirana na način da se ne koristi kao zasebna aplikacija već se funkcionalnosti te aplikacije pozivaju od strane drugih aplikacija. Sučelje servisa na neki način opisuje funkcionalnosti koje servis nudi, a aplikacije koje pozivaju funkcionalnosti servis komuniciraju preko nekakvog komunikacijskog mehanizma.
- The term service refers to a set of related software functionalities that can be reused for different purposes, together with the policies that should control its usage. (http://en.wikipedia.org/wiki/Service_(systems_architecture))

Servisi

- U najširem smislu, servisom možemo smatrati serverski dio svake server-klijent arhitekture softverskog sustava.
- Npr. kada ste radili soket klijenta za kvadriranje korisnik uopće ne treba biti svjestan da se dio funkcionalnosti izvršava na serveru. On je u interakciji sa klijentskom aplikacijom kojoj uslugu (servis) kvadriranja pruža serverska aplikacija.

Servisi

- U najširem smislu riječi servis svaka web aplikacija je servis.
- Na klijentskoj strani imamo web browser koji traži uslugu od web servera koji mu pruža tu uslugu u obliku isporuke različitog sadržaja.
- Najčešće se servis realizira kao mrežna aplikacija, tj. funkcionalnostima servisa se pristupa preko računalne mreže koristeći neki mrežni komunikacijski protokol za komunikaciju između servisa i aplikacije koja koristi funkcionalnosti servisa.

Servisi

- No servisi mogu biti realizirani i tako da se pozivaju lokalno na istom računalu, npr. Windows Servisi ili Unix deamon aplikacije – to su aplikacije koje se vrte u *backgroundu* i koji nisu namijenjene direktnoj interakciji sa korisnikom, već drugim aplikacijama pružaju različite funkcionalnosti.
- Ovdje se javlja još jedno svojstvo servisa, a to je da nisu namijenjeni direktnoj interakciji sa korisnikom već isključivo interakciji sa drugim aplikacijama.

Servisi

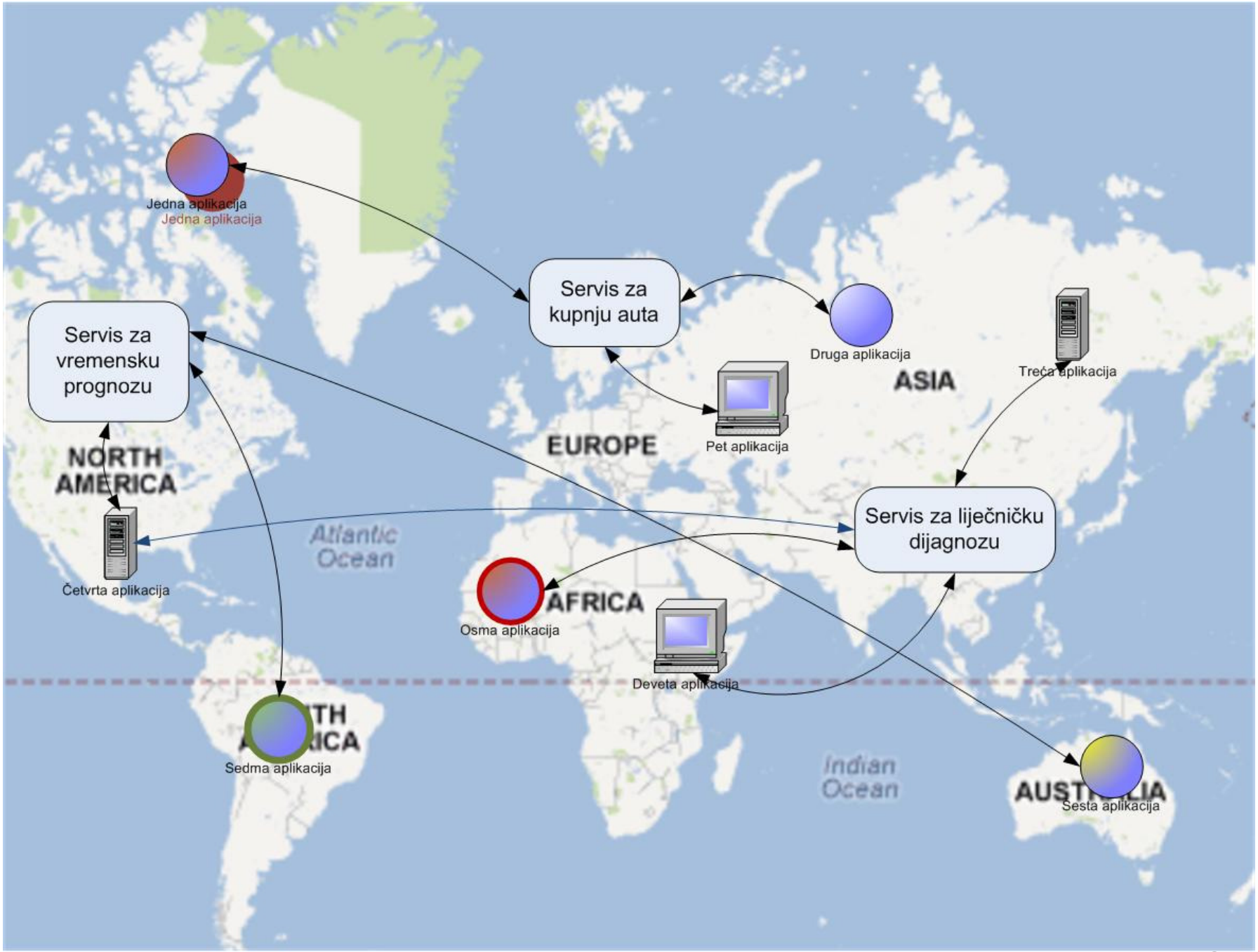
- Obično se pod pojmom servisa, u užem značenju pojma, podrazumijeva mrežno dostupna aplikacija koji koristi HTTP protokol kao transportni protokol tj. web servis, a u još užem značenju se podrazumijeva aplikacija koja koristi HTTP protokol kao transportni protokol, WSDL (*Web Service Description Language*) protokol za definiranje sučelja servisa, te SOAP (*Simple Object Access Protocol* ili *Service Oriented Architecture Protocol*) kao komunikacijski protokol.

Servisi

- Pristup realizaciji aplikacije kroz servisnu paradigmu postoji odavno, ali se 2000-tih formalizira i u biti se pokušava da se servisna paradigma u potpunosti otvori tako da omogući kreiranje mreže servisa koji su heterogeni (realizirani na različitim platformama, u različitim programskim jezicima, tehnologijama i sl.), čija su sučelja unificirana (sučelje se prema klijentima pruža na ujednačeni način), koji imaju unificirani način komunikacija (npr. pomoću definiranja otvorenih standarda poput SOAP-a)

Servisi

- Cilj (ili ideja) ovakvih napora u postavljanju standarda i razvoju tehnologija je korištenje (u tom trenutku) već razvijene mrežne infrastrukture tako da pojedine kompanije razvijaju različite servise koje onda druge kompanije ne trebaju razvijati već kupuju ili unajmljuju od vlasnika servisa.


Jedna aplikacija
Jedna aplikacija
Servis za kupnju auta
Druga aplikacija
Treća aplikacija
ASIA
Servis za vremensku prognozu
Pet aplikacija
EUROPE
NORTH AMERICA
Servis za liječničku dijagnozu
Atlantic Ocean
Četvrta aplikacija
AFRICA
Osma aplikacija
Deveta aplikacija
Sedma aplikacija
Indian Ocean
Sesta aplikacija

Servisi

- Tehnologije koje se koriste za realizaciju servisa trebaju uključivati:

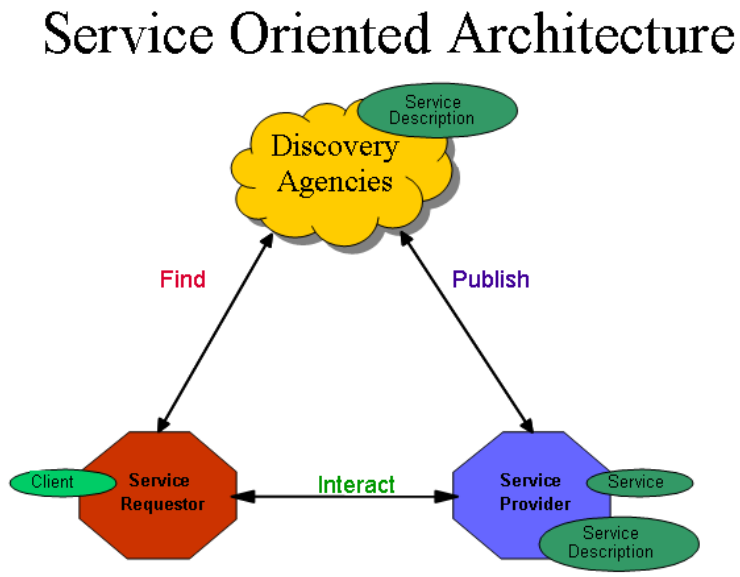
1. Mehanizam opisa sučelja servisa (funkcionalnosti)
2. Transportni mehanizam za prijenos podataka između klijenta i servisa
3. Komunikacijski mehanizam (tko šalje kakve podatke, kojim redosljedom,...)
4. Adresni mehanizam

Servisno orijentirana arhitektura

- 2000-tih se pokušava postaviti standard Servisno Orijentiranje Arhitekture (SOA arhitekture) softverskog sustava od strane OMG organizacije (http://soa.omg.org/).
- Postavljena je definicija kao i svojstva takvih softverskih sustava:
- Servisno Orijentirana Arhitektura je softverska arhitektura u kojoj se funkcionalnost aplikacije izlaže u obliku servisa.

Servisno orijentirana arhitektura

- Servis je ponašanje (ili funkcija) koje jedna komponenta pruža drugim komponentama na osnovu dogovorenog sučelja.



Servisno orijentirana arhitektura

- Servis ima mrežno adresirano sučelje. Klijent pristupa servisu preko tog njegovog sučelja.
- SOA arhitektura se zasniva na dizajniranju i izvedbi sustava korištenjem heterogenih (različitih) mrežno adresabilnih softverskih komponenti. Mrežno adresiranje softverske komponente može se realizirati na različite načine (npr. URI http://www.fesb.hr/nekiservis, tcp:161.53.166.3:5555, …)
- Servis se može pronaći dinamički, tj. klijent ne mora znati gdje je servis.

Servisno orijentirana arhitektura

- Danas se pod SOA arhitekturom definira skup principa i metodologija za dizaniranje i razvoj softvera u obliku interoperabilnih servisa.
- Service-Oriented Architecture (SOA) is a set of principles and methodologies for designing and developing software in the form of interoperable services. SOA generally provides a way for consumers of services to be aware of available SOA-based services.

Izvor: www.wikipedia.org

Softver kao servis

- U zadnje vrijeme se koristi novi “buzz” termin – Software as a Service (SaaS) kojim se ideja servisa povezuje sa cloud softverskom arhitekturom (cloud computing – računarstvo u oblacima ☺).
- Software as a service (SaaS), sometimes referred to as "on-demand software", is a software delivery model in which software and associated data are centrally hosted on the cloud. SaaS is typically accessed by users using a thin client via a web browser.

http://en.wikipedia.org/wiki/Software_as_a_service

Softver kao servis

- An online service, also known under the title of Software as a Service (SaaS), is a service provided by a software application running online and making its facilities available to users over the Internet via an interface (be that HTML presented by a web-browser such as Firefox, via a web-API or by any other means).
- With an online-service, in contrast to a traditional software application, users no longer need to ‘possess’ (own or license) the software to use it. Instead they can simply interact via a standard client (such as web-browser) and pay, where they do pay, for use of the ‘service’ rather than for ‘owning’ (or licensing) the application itself.

http://opendefinition.org/software-service/

Cloud computing

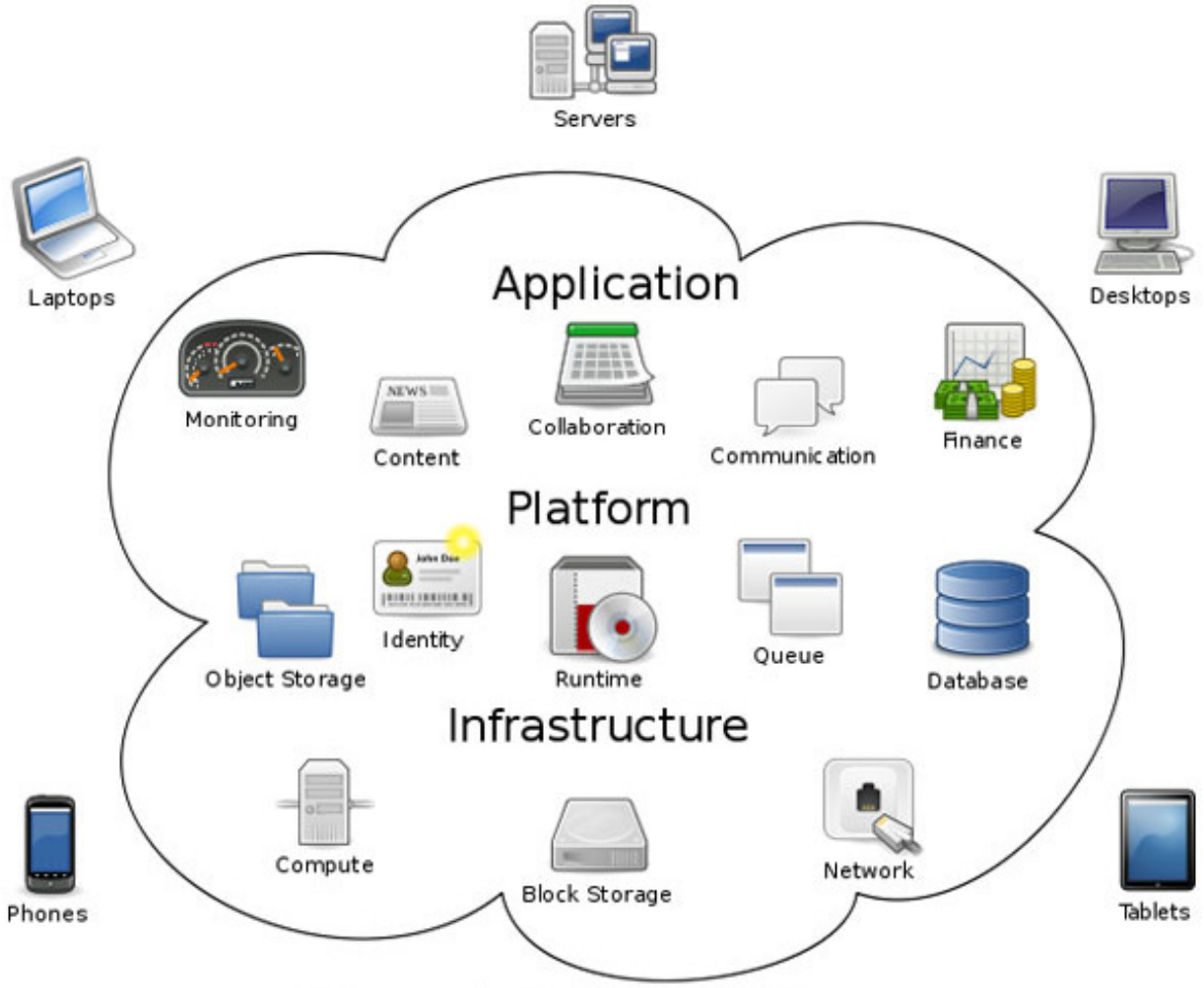
- Cloud computing pruža “neograničene” računalne i skladišne resurse kao servis krajnjim korisnicima. Naravno da krajnji korisnici koriste raspoloživi računalni i skladišni kapacitet koristeći cloud aplikacije.
- Cloud computing refers to the delivery of computing and storage capacity as a service to a heterogeneous community of end-recipients.

http://en.wikipedia.org/wiki/Cloud_computing

Cloud computing

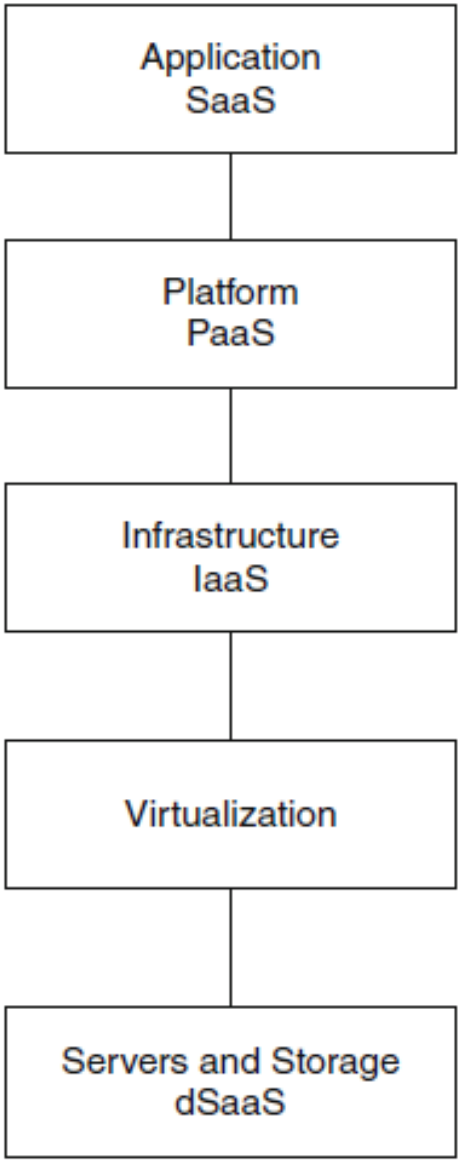
- Cloud computing can be defined as a new style of computing in which dynamically scalable and often virtualized resources are provided as a services over the Internet. (Handbook of Cloud Computing, Borko Furht, Armando Escalante, Springer, 2010.)
- Cloud ne mora uvijek biti dostupan preko Interneta, tj. obično se cloud dijeli na *public* (ili *external cloud*) (dostupan preko Interneta svima besplatno ili uz plaćanje), te na *private* (ili *internal cloud*) koji je zatvoren na intranetu.

Cloud computing



Cloud computing slojevi

- Definicija CC i SaaS izgledaju iste, ali SaaS je u biti podskup od CC koji pokriva aplikacijsku razinu CC.
- Često se tri sloja (Infrastructure-as-a-service (IaaS), Virtualization, data-Storage-as-a-Service (dSaaS)) promatraju kao jedan koji osigurava računalnu infrastrukturu (procesorsku snagu, kapacitet trajne pohrane podataka, mrežnu pojasnu širinu ....).
- Platform-as-a-Service (PaaS) uključuje i potreban sistemski softver za aplikacijsku razinu (operacijski sustav, Web server, ...).



Web servis

- Tipovi web servisa:

1. SOAP (klasični web servis)
2. REST (*Representational State Transfer*)
3. XML-RPC (razvojem XML-RPC je na kraju dobiven SOAP)

- Web API (Application Programming Interface) je programersko sučelje za korištenje web servisa.

REST Web servis

- REST (*Representational State Transfer*) je protokol u kojem su status servisa i funkcionalnost podijeljeni u resurse.
- Svaki resurs je jedinstveno adresiran korištenjem globalnog identifikatora (URI - *Uniform Resource Identifier*), tj. jedan URI predstavlja jedan objekt servisa na kojem se mogu pozivati različite funkcionalnosti.

REST Web servis

- Jednostavnije – REST Web servis je servis koji se poziva HTTP metodom (GET, POST, DELETE, ...) tako da URI koji se dohvaća identificira servis, a funkcionalnost servisa se poziva ili kroz dio upitnog URI-ja (dio URI-ja iza znaka ?) ako se koristi GET metoda ili u tijelu HTTP zahtjeva.
- Npr. http://maps.googleapis.com/maps/api/staticmap?center=%C5%A1ibenik&visible=%C5%A0ibenik,+%C5%A0ibensko-kninska+%C5%BEupanija&size=512x512&sensor=true

REST Web servis

- Formatiranje podataka koje klijent šalje servisu kao i formatiranje podataka koje servis šalje nazad klijentu nije strogo definirano.
- U ovom primjeru su podaci koje klijent šalje servisu formatirani u obliku parametar=vrijednost (center=%C5%A1ibenik&visible=%C5%A0ibenik,+%C5%A0ibenik-kninska+%C5%BEupanija&size=512x512&sensor=true), a povratna vrijednost je png slika.

REST Web servis

- Web API koji koristite u ovom slučaju nije ništa drugo nego HTTP zahtjev sa određenim formatom URI-ja koji je definiran u popratnoj dokumentaciji. (https://developers.google.com/maps/documentation/staticmaps/)
- Znači kada želite koristiti postojeći REST Web servis trebate imati opis tog servisa, kako pozivati funkcionalnosti servisa, kako formatirati podatke,... tj. koristite web API za taj servis.

REST Web servis

- Često se podaci formatiraju u XML (*eXtensible Markup Language*) ili u JSON (JavaScript Object Notation) formatu.
- Primjer: http://www.flickr.com/services/api/
- Često se servis koji koristi JSON format naziva JSON Web servis, ali to je u biti REST servis koji se oslanja na JSON format.
- Primjer: http://www.geonames.org/export/JSON-webservices.html

REST Web servis

- JSON format (kao i XML) otvoreni standard za pohranu podataka u formatu koji je i *human-readable* s time da je JSON povezan sa JavaScript jezikom i ustvari se smatra "dijelom" JavaScripta te je moguće "lako" pretvaranje JSON zapisa u JavaScript objekt.

```
<?xml version="1.0" encoding="UTF-8"?>
<rsp stat="ok">
<method>flickr.test.echo</method>
<format>rest</format>
<foo>bar</foo>
<api_key>17c844fcc5ab5a626150be9c6dd0
    6fbf</api_key>
</rsp>
```

```
jsonFlickrApi(
{"method":{"_content":"flickr.test.echo"},
    "format":{"_content":"json"},
    "foo":{"_content":"bar"},
    "api_key":{"_content":"17c844fcc5ab5a
    626150be9c6dd06fbf"}, "stat":"ok"})
```

XML-RPC Web servis

- XML-RPC jest protokol za udaljeno pozivanje procedure koji koristi XML (*Extensible Markup Language*) poruke i HTTP protokol kao prijenosni protokol. (www.xmlrpc.com/)
- To je vrlo jednostavan protokol koji sadrži samo par tipova podataka i naredbi. XML tagovi su predefinirani (njihovo ime i što sadržavaju).
- Dodavanjem novih funkcionalnosti standard je evoluirao u SOAP protokol.

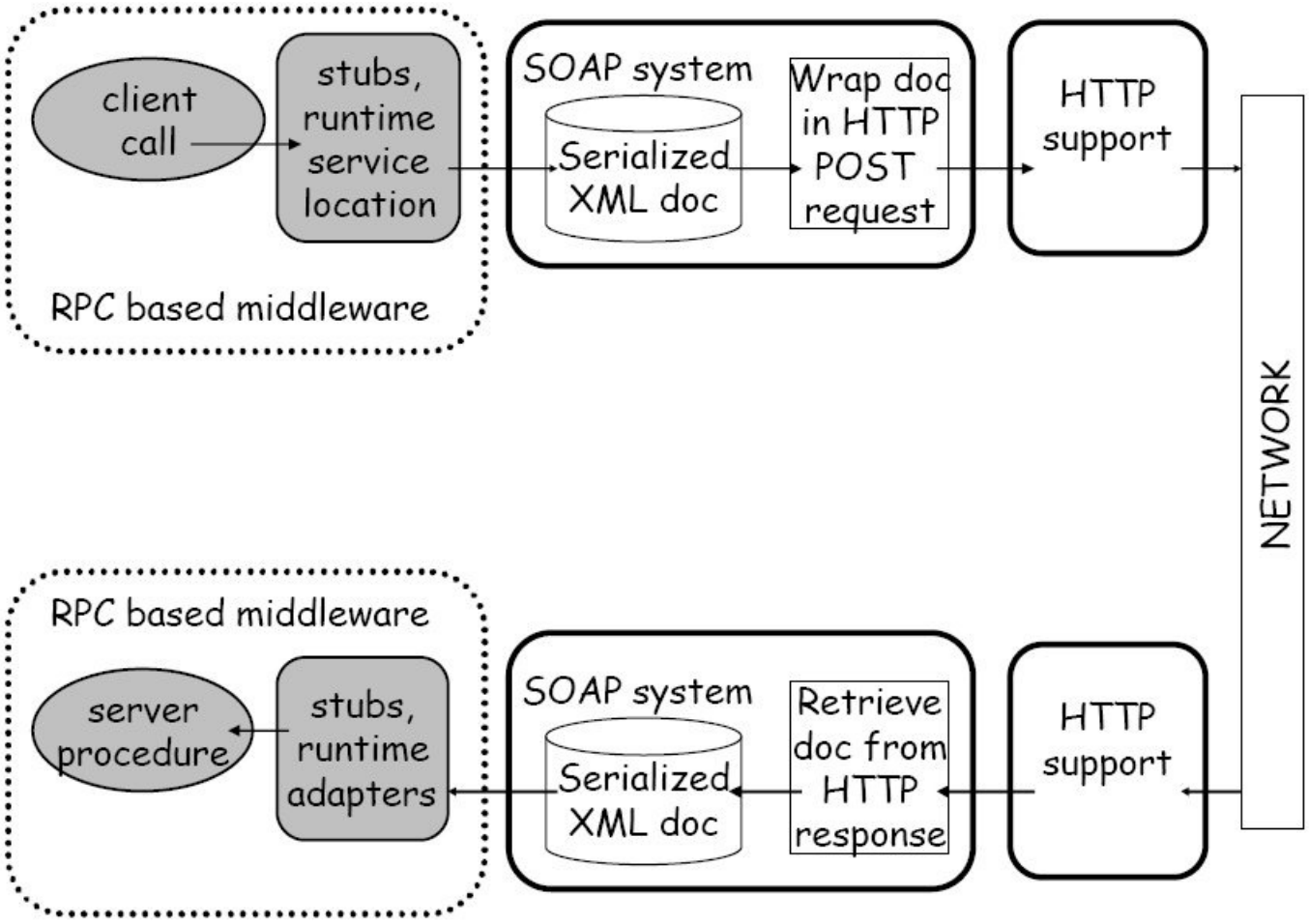
SOAP Web servis

- Ovaj tip servisa se oslanja na SOAP (*Simple Object Access Protocol* ili *Service Oriented Architecture Protocol*) protokol za komunikaciju između servisa i klijenta.
- SOAP protokol definira točno format i sadržaj SOAP poruka koje razmijenjuju servis i klijent.
- SOAP olakšava interoperabilnost različitih platformi i programa.

SOAP protokol

- Najvažnija primjena SOAP protokola je kao RPC protokola.
- SOAP se oslanja na XML tehnologiju tj. poruke su u XML formatu.
- SOAP koristi HTTP protokol kao transportni protokol.

SOAP protokol



SOAP poruka

- SOAP poruka se sastoji od SOAP omotača (*Envelope*), SOAP zaglavlja (*Header*) i SOAP tijela (*Body*).
- Omotač je izvorni (*root*) element svakog SOAP dokumenta i sastoji se od opcionalnog zaglavlja i obaveznog tijela.

SOAP poruka

- Zaglavlje i tijelo mogu sadržavati bilo kakav XML kod uz uvjetom da je važeći, dobro oblikovan, kvalificiran imenskim prostorom i da ne sadrži DTD (*Document Type Definition*) reference.
- Zaglavlje služi za transport informacija koje nisu direktno povezane uz sadržaj same poruke.
- Zaglavlje se sastoji od blokova zaglavlja (*headerblocks*) koji sadrže informacije važne za procesiranje SOAP poruke.
- Tijelo je obavezno u svakoj SOAP poruci i prenosi stvarnu informaciju.

SOAP poruka

- Unutar primjera poruke u zaglavlju se nalaze dva bloka zaglavlja sa imenskim prostorima ns1 i ns2.
- Stvarna informacija koja se želi prenijeti SOAP porukom sadržana je u *message* elementu koji se nalazi unutar tijela.
- *Message* element je određen imenskim prostorom xyz.
- *Message* nije ključna riječ već je to neka od poruka tj. operacija definiranih za taj Web servis.

```
<SOAP:Envelope
xmlns:SOAP="http://www.w3.org/2003/05/soap-
    envelope">
<SOAP:Header>
<ns1:headerBlock1
    xmlns:ns1="http://www.example.org/ns1">
... </ns1:headerBlock1>
<ns2:headerBlock2
    xmlns:ns2="http://www.example.org/ns2">
... </ns2:headerBlock2>
</SOAP:Header>

<SOAP:Body>
<xyz:message xmlns:xyz="http://www.xyz.net">
... </xyz:message>
</SOAP:Body>
</SOAP:Envelope>
```

SOAP poruka

- SOAP poruka može imati više različitih XML elemenata u zaglavlju i tijelu.
- Da bi se izbjegli konflikti između imena moraju se koristiti različiti imenski prostori.
- Imenski prostor http://schemas.xmlsoap.org/soap/envelope definira imenski prostor standardnih SOAP elemenata omotača, zaglavlja i tijela. To je ujedno i link na XML shemu. Ovaj imenski prostor određuje verziju SOAP-a i obavezan je.
- Svaki blok zaglavlja u zaglavlju trebao bi imati svoj imenski prostor.

SOAP poruka

- Aplikacija mora biti u mogućnosti pravilno interpretirati informacije koje se prenose SOAP porukom. Stoga SOAP specifikacija definira stilove kodiranja (*Encoding Styles*).
- Stil kodiranja je skup pravila koji precizno definira način kodiranja različitih tipova podataka koji se koriste u SOAP poruci.
- SOAP definira standardni stil kodiranja koji je određen http://www.w3.org/2003/05/soap-encoding.

SOAP poruka

- SOAP tipovi podataka su podijeljeni u dvije kategorije:
  - Skalarni tipovi (*scalar types*)
  - Složeni tipovi (*compound types*)
- Skalarni tipovi zadrže samo jednu vrijednost. SOAP je preuzeo sve ugrađene jednostavne tipove podataka iz specifikacije XML sheme *DataTypes*.
- Popis svih jednostavnih tipova podataka možete vidjet na http://www.w3.org/TR/2000/WD-xmlschema-0-20000407

SOAP poruka

- Složeni tipovi sadrže više vrijednosti.
- Složeni tipovi su podijeljeni na nizove (*arrays*) i strukture (*structs*).
- Nizovi mogu sadržavati više vrijednosti gdje svaka vrijednost ima svoj redni broj.
- Strukture isto tako mogu sadržavati više vrijednosti, a svaka vrijednost je određena *accessor* imenom.

SOAP poruka

- Kod definiranja SOAP nizova moramo navesti duljinu niza i tip elemenata.
- Prvo moramo postaviti xsi:type atribut na *Array* gdje je *Array* definiran *soap-encodingom*.
- Zatim moramo postaviti *arrayType* atribut koji određuje tip elemenata unutar niza.
- Svaki element unutar niza određen je elementom *item*.

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
xmlns:SOAP-
    ENV="http://www.w3.org/2001/09/soap-
    envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<ns1:getPriceListResponse
xmlns:ns1="urn:examples:pricelistservice"
SOAP-
    ENV:encodingStyle="http://www.w3.org/2001/0
    9/soap-encoding">
<return
xmlns:ns2="http://www.w3.org/2001/09/soap-
    encoding"
xsi:type="ns2:Array"
    ns2:arrayType="xsd:double[2]">
<item xsi:type="xsd:double">57.99</item>
<item xsi:type="xsd:double">12.99</item>
</return>
</ns1:getPriceListResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP poruka

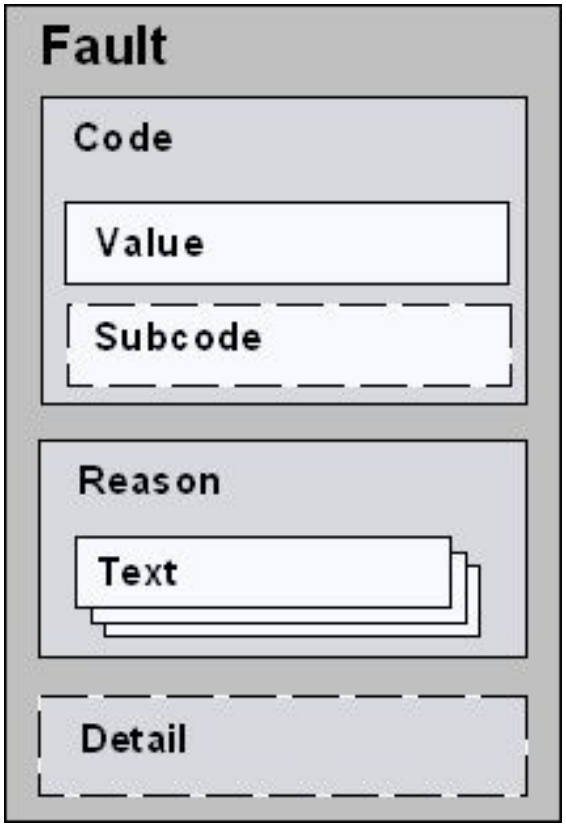
- Strukture sadrže više vrijednosti gdje je svaka vrijednost određena *accessorom*.
- U ovom primjeru *accessori* su: *name*, *price* i *description*.
- Za razliku od nizova svaki element može biti različitog tipa.

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
xmlns:SOAP-
    ENV="http://www.w3.org/2001/09/soap-
    envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchem
    a-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchem
    a">
<SOAP-ENV:Body>
<ns1:getProductResponse
xmlns:ns1="urn:examples:productservice"
SOAP-
    ENV:encodingStyle="http://www.w3.org/200
    1/09/soap-encoding">
<return xmlns:ns2="urn:examples"
    xsi:type="ns2:product">
<name xsi:type="xsd:string">Web
    Camera</name>
<price xsi:type="xsd:double">31.99</price>
<description xsi:type="xsd:string">VGA, 30
    fps</description>
</return>
</ns1:getProductResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP upravljanje greškama

- SOAP nudi mehanizam reagiranja na greške koje mogu nastati prilikom procesiranja SOAP poruka tako što obavijesti pošiljaoca poruke o nastanku greške.
- Opis greške prenosi se unutar *Fault* elementa koji se nalazi unutar tijela poruke s greškom.

Struktura *SOAP Fault* elementa



SOAP upravljanje greškama

- *Code* element sadrži *Value* element, koji određuje vrstu greške.
- SOAP specifikacija definira standardne kodove greški (*Fault Codes*) koji se moraju koristiti unutar *Value* elementa.
- *Reason* se koristi da bi mogli ispisati poruku o greški razumljivu za korisnika. *Reason* element sadrži jedan ili više *Text* elemenata koji se razlikuju u vrijednostima *lang* atributa. *Lang* atribut omogućava prikazivanje poruka o greški na različitim jezicima.
- *Detail* element sadrži dodatne informacije o nastaloj greški specifične za određenu aplikaciju.

```
<SOAP:Envelope
xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope">
<SOAP:Body>
<SOAP:Fault>
<SOAP:Code>...</SOAP:Code>
<SOAP:Reason>
<SOAP:Text xml:lang="en-US">Missing parameter</SOAP:Text>
<SOAP:Text xml:lang="hr">Nedostaje Parameter</SOAP:Text>
</SOAP:Reason>
<SOAP:Detail>...</SOAP:Detail>
</SOAP:Fault>
</SOAP:Body>
</SOAP:Envelope>
```

Web servis

- Dobar pregled postojećih web API-ja možete naći na: http://www.programmableweb.com/apis/directory

- W3 konzorcijum informacije na http://www.w3.org/standards/webofservices/

Web servisi

- Sučelje web servisa je u primjerima do sada bilo opisano u dokumentaciji web API-ja servisa.
- WSDL jezik (Web Service Description Languga) definiran je za opisivanje sučelja servisa preko kojeg klijenti mogu “automatizirati” kreiranje zahtjeva prema funkcionalnostima servisa. Inicijalno su web servisi i bili definirani vrlo formalno sa WSDL opisom, koristili su strogo formatirane poruke (SOAP) protokol, ali se pokazalo da se web servisi mogu definirati i koristiti puno neformalnije.

WSDL

- WSDL su razvili Microsoft i IBM.
- Danas razvoj WSDL radi W3C konzorcijum.
- Trenutne verzije WSDL i SOAP standarda možete pronaći na http://www.w3.org/2002/ws/
- WSDL pruža definirani, zajednički način opisivanja servisa koji klijent koristi kako bi mogao komunicirati sa servisom.