



Spletne storitve





Porazdeljene storitve

- ▶ zbirka metod, ki jih je možno klicati na oddaljenih sistemih z uporabo običajnih spletnih protokolov
 - klic in izmenjava podatkov z uporabo XML, JSON, HTTP
- ▶ prejemniki storitev (odjemalci) so lahko spletne aplikacije, namizne aplikacije ali druge spletne storitve
- ▶ vrste storitev:
 - *spletne storitve* (Web Services, od 2002) in *WCF storitve* (Windows Communication Foundation, od 2006, niso omejene samo na spletne protokole)
 - arhitektura SOAP (Simple Object Access Protocol, zapis XML) in REST (Representational State Transfer, uporaba GET in POST)



WCF

- ▶ Windows Communication Foundation je izvajalno okolje in množica API-jev za izgradnjo storitveno-orientiranih aplikacij
- ▶ WCF se pogosto uporablja za implementacijo storitveno-orientiranih arhitektur
- ▶ podpira porazdeljeno računalništvo:
 - storitve imajo oddaljene odjemalce
 - odjemalec lahko uporablja več storitev
 - storitev lahko uporablja več odjemalcev
- ▶ povezovanje na storitev preko končnih točk:
 - naslov/povezava/pogodba
 - storitev posreduje pogodbo, ki ima naslov in lastnosti za povezavo
 - interakcija med končno točko in odjemalcem s SOAP sporočili
 - strežnik in odjemalec morata imeti kompatibilni končni točki



Arhitektura storitveno orientiranih implementacijskih ogrodij

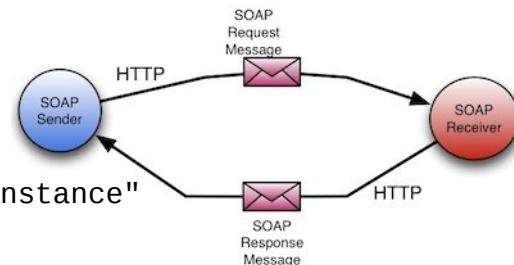
- ▶ storitveno orientirane arhitekture (SOA) temeljijo na programskih storitvah, ki vključujejo spletne storitve
- ▶ implementacije SOA zahtevajo orodja za načrtovanje in izvajalno infrastrukturo – SOA implementacijska ogrodja
- ▶ prednosti implementacijskih ogrodij SOA
 - učinkovitost
 - integracija aplikacij
 - razvoj in postavitve aplikacij
- ▶ zahteva implementacijskih ogrodij SOA
 - porazdeljena dogodkovno-omogočena arhitektura
 - podpora standardom
 - zanesljivost, skalabilnost, odporost na napake
 - varnost v porazdeljenih okoljih
 - spremembe procesov

SOAP

- ▶ SOAP (Simple Object Access Protocol): protokol za zapis zahtevkov/odgovorov storitev, ki temelji na obliki XML in komunikaciji preko HTTP (naslednik RPC-XML)
 - korenski element XML je ovojnica (Envelope), štiri možni elementi sporočila
 - vsebina je lahko zahtevek ali odgovor
- ▶ pošiljanje zahtevka SOAP storitvi, ki odgovori z XML formatiranim sporočilom s podatki
- ▶ zagotavlja zapis podatkov in podatkovnih tipov, ki jih uporabljajo metode – ovojnica SOAP definira strukturo sporočila

```
POST /paymentservice/PaymentService.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CalculatePayment xmlns="http://fri.org/">
      <loanAmt>10000</loanAmt>
      <intRate>5</intRate>
      <months>72</months>
    </CalculatePayment>
  </soap12:Body>
</soap12:Envelope>
```



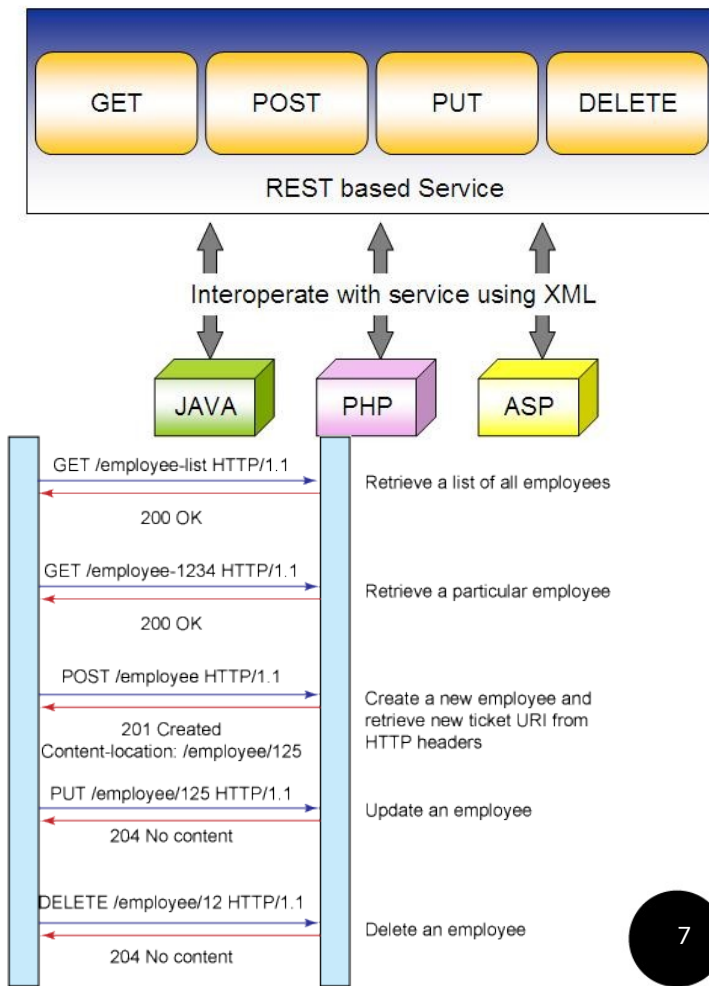


SOAP

- ▶ lastnosti:
 - razširljivost (npr. varnost, WS-routing, ...)
 - nevtralnost (npr. HTTP, SMTP, TCP, ...)
 - neodvisnost (npr. imperativni, deklarativni, funkcijski, ... model)
- ▶ specifikacija:
 - model procesiranja SOAP (procesiranje sporočil; vozlišča: pošiljatelj, prejemnik, pot, posrednik, končni prejemnik)
 - model razširitev SOAP (lastnosti in moduli)
 - okvir za SOAP protokole (definiranje povezav za protokole)
 - sporočilni konstrukt SOAP (definira strukturo sporočila)
- ▶ prednosti in slabosti:
 - uporaba različnih transportnih protokolov (različni OSI nivoji)
 - prehajanje preko požarnih zidov (uporaba HTTP)
 - počasnejše (XML je potraten)
 - fiksne vloge strank v interakciji (odjemalec - strežnik)

REST

- ▶ REST (Representational State Transfer) je način definiranja operacij za komunikacijo s storitvami, ki so identificirane z enoličnim naslovom URL
 - v primeru uporabe zahtevkov GET je možno predpomnjenje in pohitritev
- ▶ mapiranje metod HTTP za doseg različnih storitev: GET, POST, PUT, DELETE
- ▶ t.i. RESTful storitve niso omejene na vračanje odgovorov v obliki XML, temveč lahko tudi v oblikah JSON, HTML, multimedijske datoteke ipd.



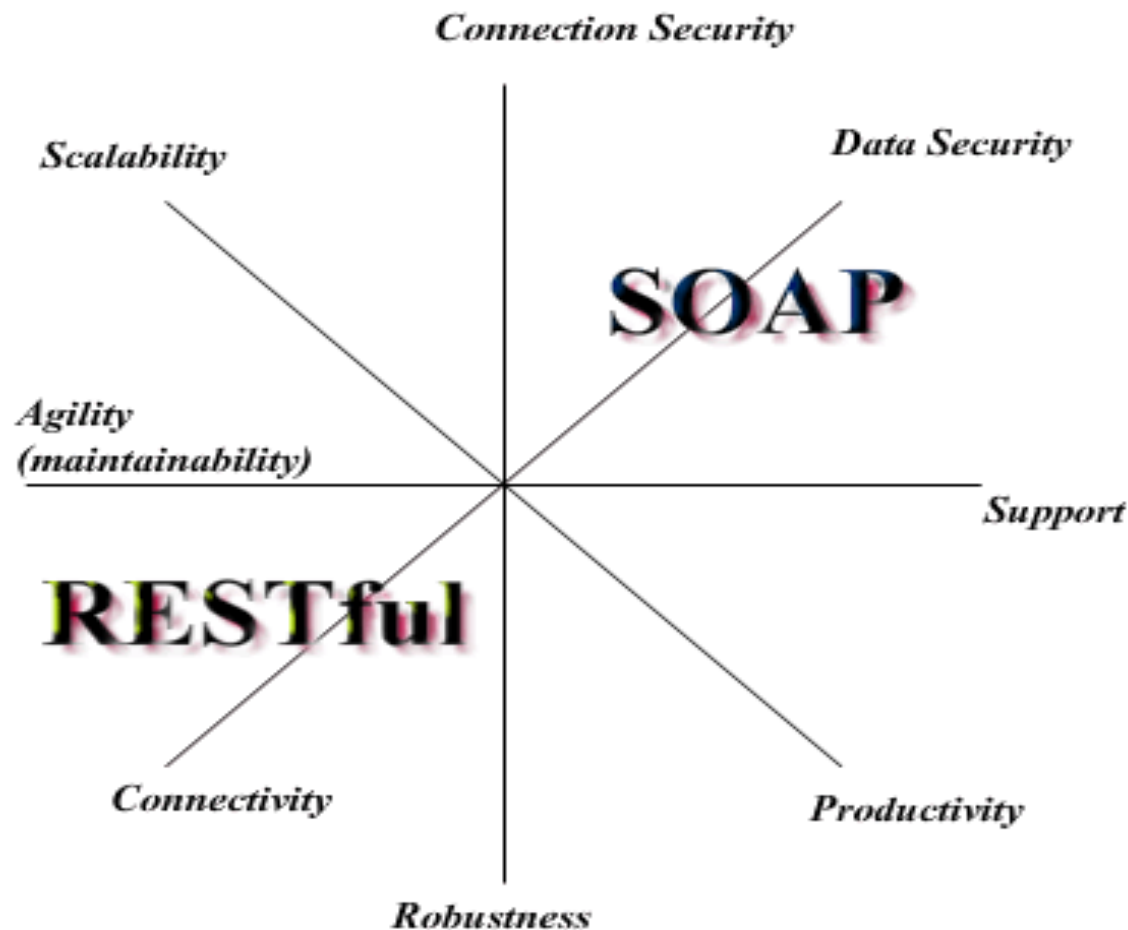


REST

- ▶ arhitektura:
 - komponente (programske instrukcije in notranje stanje)
 - konektorji (posredujejo pri komunikaciji koordinaciji ali kooperaciji)
 - podatki (informacija, ki se posreduje od/do komponente preko konektorjev: resursi, identifikatorji resursov, reprezentacije, ...)
- ▶ arhitekturne omejitve
 - strežnik-odjemalec (ločeni komponenti)
 - ni stanja (zagotavljanje vse potrebne informacije)
 - shranjevanje odgovorov (odgovori morajo definirati ali se shranjujejo)
 - sistem v slojih (se ne da ugotoviti ali je povezava na končni strežnik)
 - koda na zahtevo (prilagoditev funkcionalnosti na odjemalcu)
 - uniformen vmesnik (ločen razvoj posameznih komponent)
- ▶ RESTful spletne storitve ustrezajo REST arhitekturnim zahtevam
- ▶ RESTful API-je, ki temelji na HTTP, določajo:
 - osnovni URI npr. <http://example.com/resources>
 - internetni tip podatkov, npr. JSON
 - standardne HTTP metode
 - hipertekstne povezave za referenciranje stanja in povezanih resursov

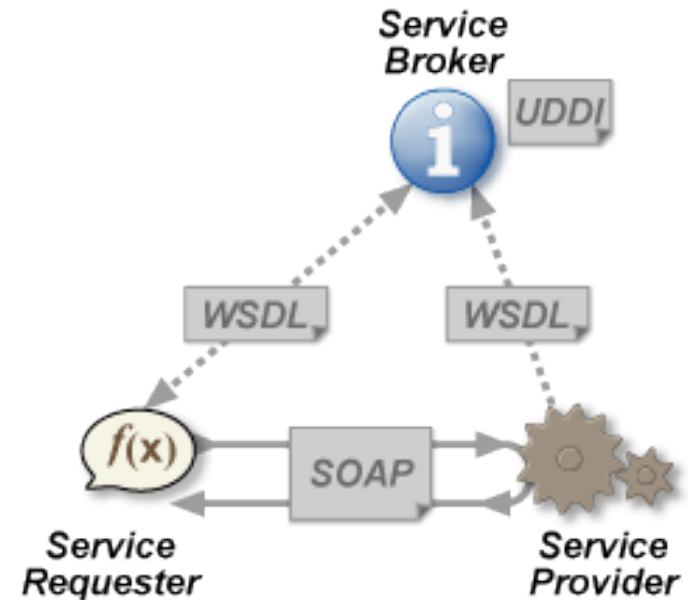


SOAP in REST



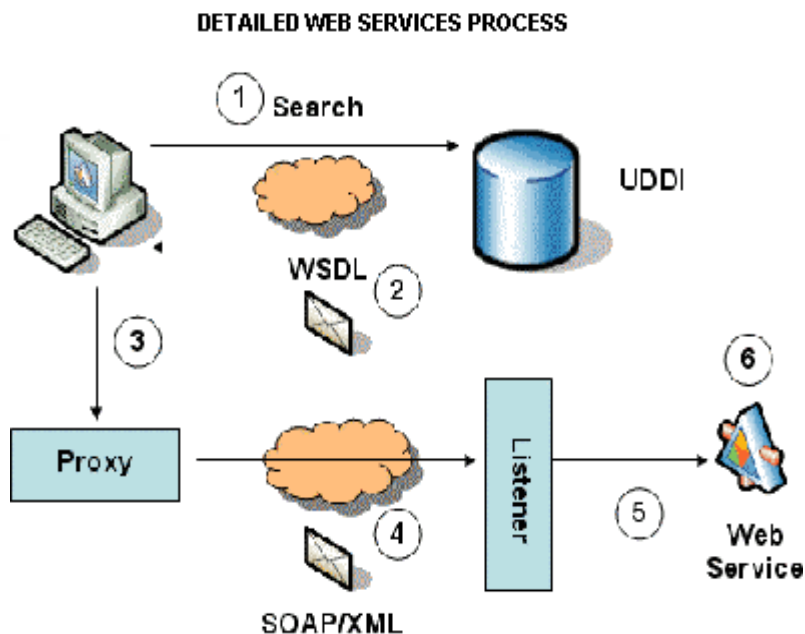
UDDI in WSDL

- ▶ kako najti ustrezno storitev in ugotoviti, kakšne podatke sprejema/vrača?
 - protokoli za opis in odkrivanje storitev!
- ▶ **UDDI (Universal Description, Discovery and Integration Service)** je protokol za izdelavo registra storitev. Nudi sredstva za povpraševanje baze, katere storitve so na razpolago. Vključuje bele, rumene in zelene strani.
- ▶ **WSDL (Web Services Description (Definition) Language)** ponuja standardiziran opis storitve (spremenljivke, protokoli) v obliki XML. WSDL določajo: “service”, “endpoint/port”, “binding”, “interface”, “operation”, “types”.



Izvedba spletne storitve

- ▶ klic spletne storitve se izvede preko lokalnega posrednika (proxy), ki je abstrakcija oddaljenih metod spletne storitve
- ▶ posrednik implementira protokol komunikacije z oddaljeno storitvijo





Implementacija spletnih storitev

- ▶ implementacija v razvojnih okoljih .NET/VS, Java(J2EEE)/Netbeans
- ▶ pogledali bomo dele primerov:
 - spletno SOAP storitev (.NET framework 3.5)
 - WCF SOAP storitev (.NET framework 4.0)
 - WCF REST storitev (.NET framework 3.5)

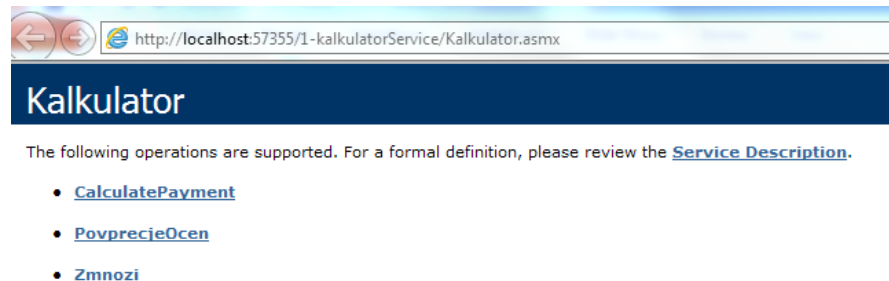




Spletna storitev (SOAP)

- direktiva strani, ki predstavlja spletno storitev
`<%@ WebService Language="C#" CodeBehind="~/App_Code/Service.cs" Class="Service" %>`
- programska koda v datoteki v ozadju .cs
 - atributi `WebService`, `WebMethod`
 - `WebServiceBinding` določa standard, ki ga implementira storitev (generira WSDL)

```
[WebService(Namespace = "http://mojastoritev.fri.org/")]  
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]  
public class Service : System.Web.Services.WebService  
{  
    [WebMethod]  
    public double PovprecjeOcen(string ocene) {  
  
        // vsebina metode  
  
    }  
}
```





Spletna storitev (SOAP)

Service

Click [here](#) for a complete list of operations.

PovprecjeOcen

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
ocene:	<input type="text"/>

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /povprecjeService/PovprecjeService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://mojastoritev.fri.org/PovprecjeOcen"

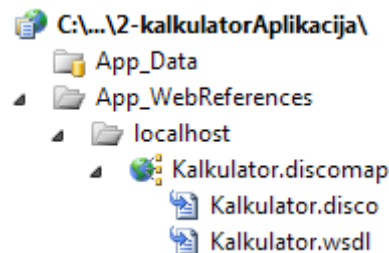
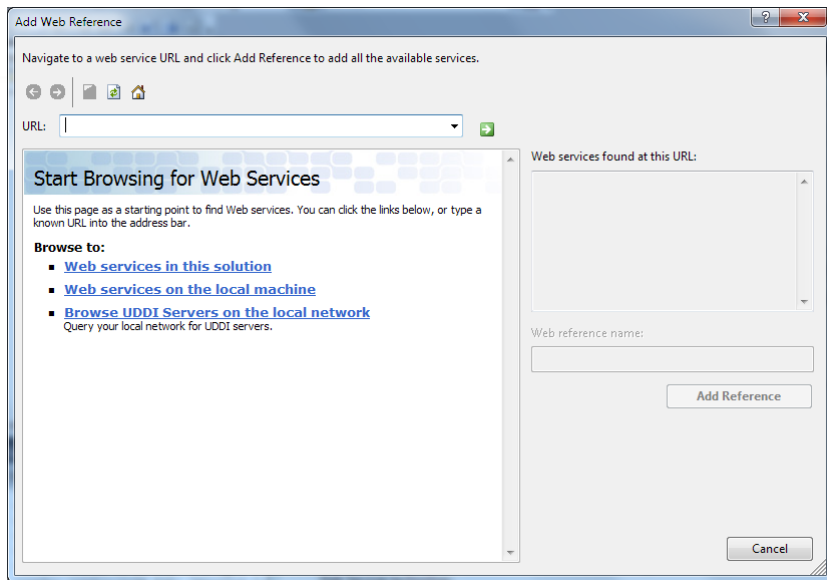
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www
  <soap:Body>
    <PovprecjeOcen xmlns="http://mojastoritev.fri.org/">
      <ocene>string</ocene>
    </PovprecjeOcen>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```



Spletna storitev (SOAP)

- ▶ izdelava odjemalca:
 - namestitev reference na spletno storitev
 - možna uporaba UDDI (UDDI Server umaknjen iz Windows Server 2008 R2)



Storitev WCF (SOAP)

- ▶ vsaka storitev WCF potrebuje tri komponente ABC:
 - naslov (**Address**): URL lokacija storitve
 - povezava (**Binding**): protokol, s katerim klient komunicira s storitvijo (SOAP, REST, opcije)
 - pogodba (**Contract**): vmesnik, ki predstavlja metode, ki jih storitev javno ponuja, storitev naj bi imela vsaj eno pogodbo

```
[ServiceContract]
public interface Idanvletu{
    [OperationContract]
    string Dan(int dan, int mesec, int leto);
}
```

```
public class Koledar : Idanvletu{
    public string Dan(int dan, int mesec, int leto)
    {
        // vsebina metode
    }
}
```


Storitev WCF (SOAP)

- ▶ izdelava odjemalca
 - uporaba posrednika

Koledar Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://localhost:51304/3-danvletuService/Service.svc?wsdl
```

You can also access the service description as a single file:

```
http://localhost:51304/3-danvletuService/Service.svc?singleWsdl
```

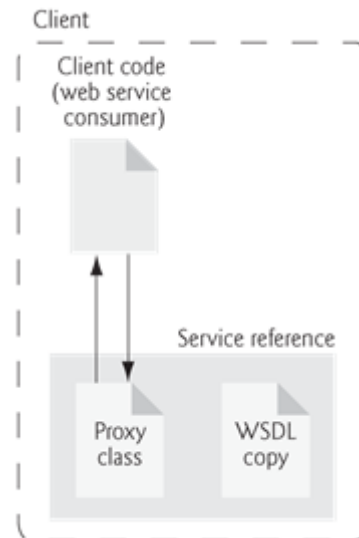
This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        IdanvletuClient client = new IdanvletuClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```



Storitev WCF (REST)

- ▶ dopolnitev WCF storitve z atributi, ki omogočajo naslavljanje metod storitev z uporabo naslovov URL
 - oznake v naslovu URL nastopajo kot parametri metod

```
[ServiceContract]
public interface Idanvletu{
    [OperationContract]
    [WebGet(UriTemplate = "/imedneva/{dan}/{mesec}/{leto}")]
    string Dan(string dan, string mesec, string leto);
}
```

- ▶ potrebno še spremeniti obnašanje storitve in povezave, parametra behaviorConfiguration in binding

