

# WSDL ve SOAP



# Web Servis Nedir?

- Web servisler başka bir sistemde çalışan ve **network üzerinden erişilebilen metotlardır**
- İstemci uygulamalar bu servisleri network üzerinden **tespit ederek**, uygun **input argümanları** ile çağırırlar, **dönen mesajları** da işlerler
- Web servislerinin tespit edilebilmesi ve arayüzlerine uygun çağrılar yapılabilmesi için **WSDL** kullanılmaktadır

# Web Services Description Language (WSDL)

- Sunucu tarafından sağlanan **servislerin (operasyon) neler** olduğunu tanımlar
- Herhangi bir **servise nasıl erişilebileceğini** gösterir
- Servise **erişim için hangi URL'in** (endpoint) kullanılacağını belirtir
- Request ve response **XML'lerinin yapısının** nasıl olması gerektiğini söyler

# WSDL Örneği

WS tarafından kullanılan veri tiplerini içeren bölümdür

```
<wsdl:definitions targetNamespace="http://www.java-egitimleri.com/greeting">
  <wsdl:types>
    <schema elementFormDefault="qualified"
      targetNamespace="http://www.java-egitimleri.com/greeting">
      <element name="helloWorldRequest">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="name"
type="string" />
          </sequence>
        </complexType>
      </element>
      <element name="helloWorldResponse">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="greeting"
type="string" />
          </sequence>
        </complexType>
      </element>
    </schema>
  </wsdl:types>
```

WS tarafından kullanılan veri tiplerini tanımlamak için XSD kullanılır

# WSDL Örneği

WS tarafından alınan ve dönülen mesajların tanımlandığı bölümdür



```
<wsdl:message name="helloWorldRequest">
    <wsdl:part element="tns:helloWorldRequest" name="helloWorldRequest" />
</wsdl:message>
<wsdl:message name="helloWorldResponse">
    <wsdl:part element="tns:helloWorldResponse" name="helloWorldResponse" />
</wsdl:message>
```

Her mesaj bir veya daha fazla bölümden (part) oluşur  
Her bölüm klasik programlamada metot parametrelerine benzetilebilir

# WSDL Örneği

WS tarafından sağlanan operasyonlar bu bölümde tanımlanır

```
<wsdl:portType name="HelloWorld">
  <wsdl:operation name="helloWorld">

    <wsdl:input message="tns:helloWorldRequest" name="helloWorldRequest"/>

    <wsdl:output message="tns:helloWorldResponse" name="helloWorldResponse"/>

  </wsdl:operation>
</wsdl:portType>
```

WS operasyonlarını ve bunların ilgili mesajlarını tanımlar  
Klasik programlama da interface'e, API'ye veya fonksiyon kütüphanesine benzetilebilir

# WSDL Örneği

WS tarafından kullanılan iletişim protokolünü içerir

```
<wsdl:binding name="HelloWorldSoap11" type="tns:HelloWorld">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="helloWorld">
        <soap:operation soapAction="" />
        <wsdl:input name="helloWorldRequest">
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output name="helloWorldResponse">
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
```

WS için mesaj formatını ve iletişim protokolü detaylarını belirler

# WSDL Örneği

WS servisinin erişim adresini tanımlar

```
<wsdl:service name="HelloWorldService">
```

```
  <wsdl:port binding="tns:HelloWorldSoap11"
    name="HelloWorldSoap11">
```

```
    <soap:address location="http://localhost:8080/petclinic/hello/" />
```

```
  </wsdl:port>
```

```
</wsdl:service>
```

```
</wsdl:definitions>
```

Bir önceki adımda **<binding>** bölümünde tanımlı servisi bir URL ile ilişkilendirir



# WSDL Operasyon Tipleri

- WSDL dört farklı tip operasyonu destekler
  - **One-way:** WS sunucu web servis çağrısını alır, fakat herhangi bir cevap dönmez
  - **Request-response:** WS sunucu web servis çağrısını alır ve bir cevap döner
  - **Solicit response:** WS sunucu istemciye bir mesaj gönderir ve cevabını bekler
  - **Notification:** WS sunucu istemciye bir mesaj gönderir, fakat yanıtını beklemez
- One-way ve request-response **en yaygın** kullanılanlardır

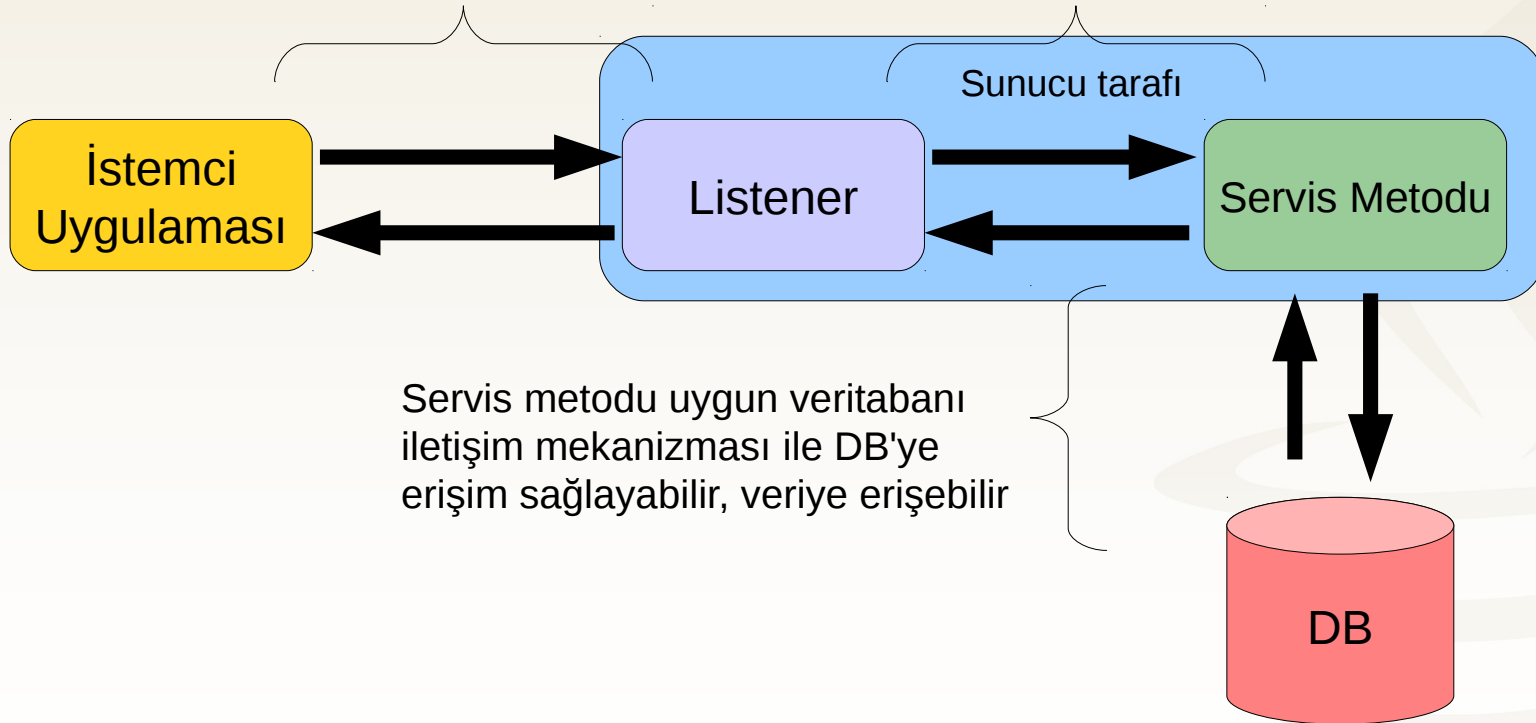
# Simple Object Access Protocol (SOAP)

- İstemci ile sunucu arasındaki iletişim **SOAP** ile sağlanır
- XML tabanlı **mesaj değişim (exchange) formatıdır**
- İstemci ve sunucu programları tamamen **farklı teknolojilerde** geliştirilmiş, **farklı işletim sistemlerinde** koşuyor olabilir
- Herhangi bir transport protokolüne **bağımlı değildir**, ancak **HTTP en popüler** transport protokolüdür

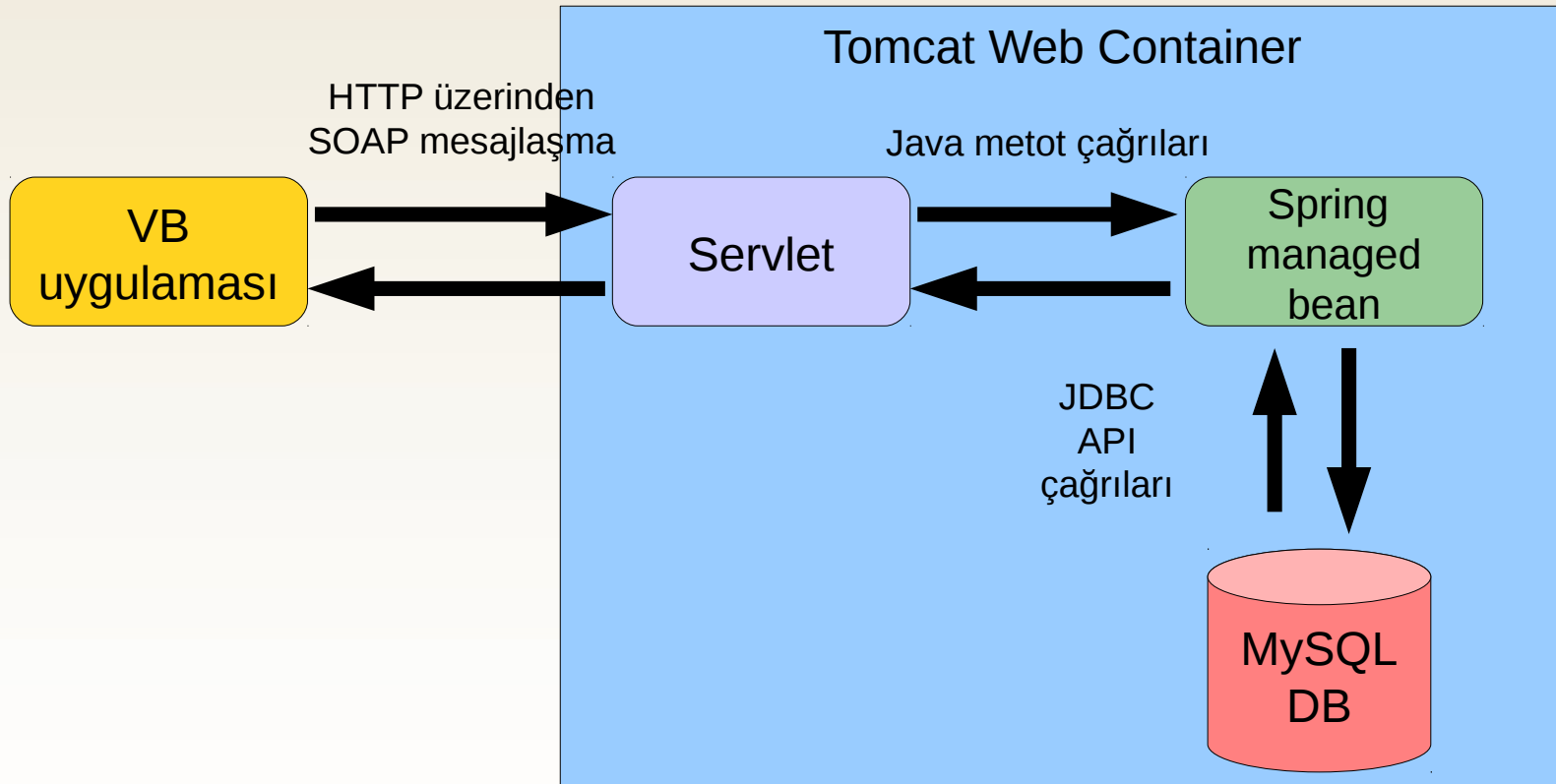
# SOAP ile Mesajlaşma Mimarisi

İstemci ve Listener arasında transport protokolü üzerinden SOAP request ve response'ları transfer edilir

Listener ve sunucu tarafındaki servis metodu genellikle aynı teknolojiye yazılmış programlardır Normal bir in-process metod request'i yapılır ve metod response'u dönülür



# SOAP ile Mesajlaşma Mimarisi ve HTTP



# SOAP Mesajının Yapısı

- Bir SOAP mesajı XML deklarasyonu ile başlar, envelope içerisinde **header** ve **body** bölümlerinden oluşur
- Hata bilgisi ise body içerisinde **fault** bölümünde taşınır

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/12/soap-envelope"
  SOAP-ENV:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <SOAP-ENV:Header>
    ...
  </SOAP-ENV:Header>

  <SOAP-ENV:Body>
    ...
    <SOAP-ENV:Fault>
      ...
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<SOAP-ENV:Fault>
  <SOAP-ENV:faultcode/>
  <SOAP-ENV:faultstring/>
  <SOAP-ENV:faultactor/>
  <SOAP-ENV:detail/>
</SOAP-ENV:Fault>
```

# SOAP Mesajının Yapısı

- **SOAP envelope**
  - XML mesajının bütününü temsil eder
- **SOAP header**
  - request'i tanımlamak için kullanılan herhangi bir opsiyonel bilgidir
  - Auth, authz, signature, enc gibi bilgiler de içerebilir
- **SOAP body**
  - Request ve response'un gerçek içeriğinin yer aldığı bölümdür
- **SOAP fault**
  - Hata durumunda sunucunun döndüğü hata bilgisidir

# SOAP Mesaj Binding

- SOAP spesifikasyonu **mesajların yapısı** ile ilgilenir
- Mesajların istemci ile sunucu arasında **nasıl transfer edileceğine** karışmaz
- Mesajın nasıl transfer edileceğini **SOAP binding** belirler
- SOAP kütüphaneleri **HTTP, SMTP, JMS** gibi farklı **transport protokolleri** için implemantasyonlar içerir

# SOAP Mesaj Binding

Bir web servisin hangi transport protokol üzerinden yürütüleceği ve Bu servisin adresi WSDL içerisindeki binding tanımları ile belirtilir

```
<wsdl:binding name="HelloWorldSoap11" type="tns:HelloWorld">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
</wsdl:binding>

<wsdl:service name="HelloWorldService">
    <wsdl:port binding="tns:HelloWorldSoap11" name="HelloWorldSoap11">
        <soap:address location="http://localhost:8080/petclinic/hello/" />
    </wsdl:port>
</wsdl:service>
```



# SOAP Binding Style

- **Document**

- SOAP mesaj içeriğinin bir **XML dokümandan** oluşacağını ve bir **XSD ile validate** edilebileceğini ifade eder

- **RPC**

- SOAP mesaj içeriğinin **bir metot çağrısının XML karşılığı** olacağını ifade eder
  - XML elemanları, metot ismine, metot parametrelerine karşılık gelir

# SOAP Body Encoding

- **Literal**
  - SOAP mesajının içerisindeki **verinin bir XML şema baz alınarak üretildiğini** belirtir
- **Encoded**
  - SOAP mesajı içerisinde yer alacak **verinin XML formatına nasıl dönüştürüleceğini** tanımlar

```
<wsdl:input name="helloWorldRequest">
  <soap:body use="literal" />
</wsdl:input>
```

# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

