

WEB SERVİS MİMARİLERİ

Web Servis Tanımı

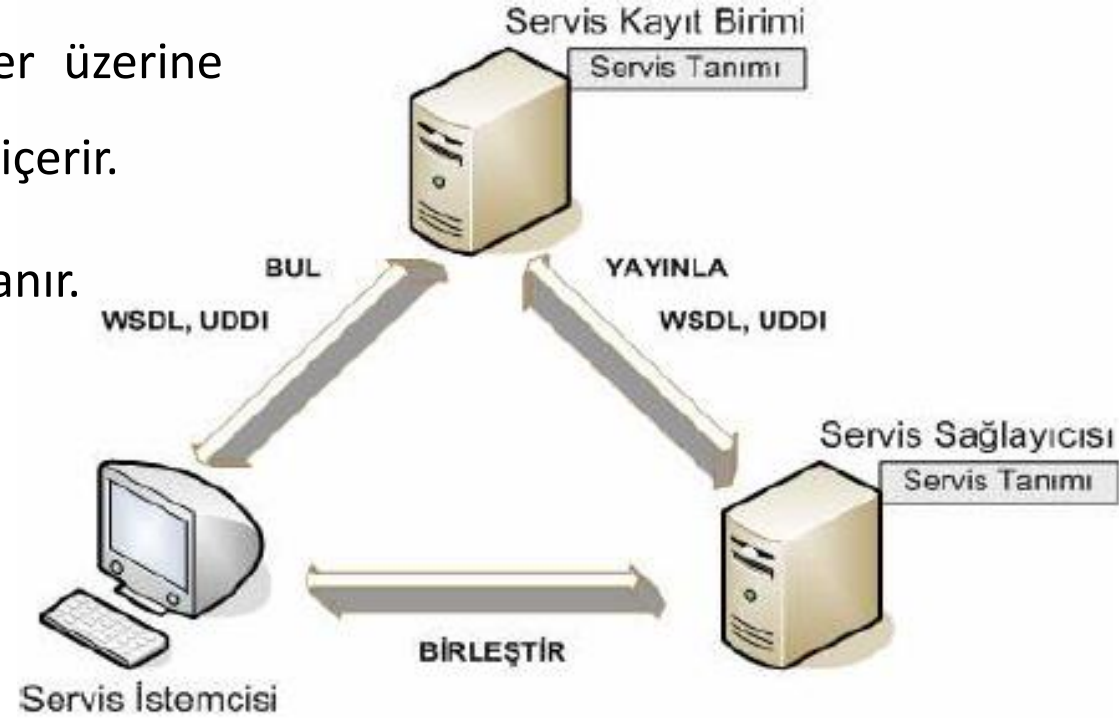
- W3C (World Wide Web Consortium) tarafından yapılan resmi tanımıyla web servisleri, bilgisayarlar arasında ağ üzerinden etkileşimi ve uyumluluğu sağlayacak yazılım sistemleridir.
- Günümüzde birbiriyle haberleşecek sistemleri gerçeklemek için en çok tercih edilen yöntem web servisleridir.
- Bir web servisi, standartlaşmış XML mesajlaşma sistemini kullanan ve internet üzerinde ulaşılabilir olan yazılım parçalarıdır.
- Bir web servisine her türlü iletişim için, XML kodlama kullanılır. Örneğin, bir istemci web servisine XML bir mesaj göndererek istekte bulunur ve ardından ilgili XML cevap için bekler. Çünkü bütün iletişim XML üzerinedir, web servisleri hiçbir işletim sistemine ya da programlama diline bağlı değildir. Yani Java ile Perl ya da Windows uygulamasıyla Unix uygulaması konuşabilmektedir.

Web Servis Tanımı-devam

- Web servisleri, internet üzerinde tanımlanan, yayınlanan, bulunan ya da çağırılan, modüler, dağıtık, dinamik uygulamalardır. Bu uygulamalar lokal, dağıtık ya da web tabanlı olabilirler. Web servisleri TCP/IP, HTTP, Java, HTML ve XML gibi açık standartlar üstüne kuruludur.
- Bir web servisi, uygulamalar ve sistemler arasında veri alışverişi için kullanılan açık standartlar ve protokoller topluluğudur denilebilir.
- Çeşitli programlama dillerinde yazılan ve çeşitli platformlar üzerinde çalıştırılan yazılım uygulamaları, internet gibi büyük bilgisayar ağları üzerinde veri alışverişi yapmak için web servislerini kullanabilirler.
- Asıl fikir, servisleri internet üzerinde dağıtmak ve onları istemciler için ulaşılabilir hale getirmektir. Yani haberleşecek sistemlerin birbirlerinden haberdar olması veya platformlarının uyumlu olması gerekmez.
- XML Web Servislerine farklı bir bilgisayar ve farklı bir platformdan istemci olunabilir. Bu birlikte çalışabilirlik özelliği (Java ve Python ya da Windows ve Linux uygulamaları gibi) açık standartların kullanımı sayesinde.

Web Servis Modeli

- Web servisleri modeli üç temel birimin, servis sağlayıcı, servis kayıt birimi ve servis istemcisi, arasındaki etkileşimler üzerine kuruludur. Etkileşimler yayınla, bul ve bağlan işlemlerini içerir.
- Web servisleri modeli uc temel birimin etkilesimine dayanır.
 - Servis Sağlayıcısı (Service Provider)
 - Servis İstemcisi (Service Requester)
 - Servis Kayıt Birimi



Web Servis Modeli-devam

- ***Servis Sağlayıcısı (Service Provider)***

Servis sağlayıcı istemcilerin sağlayıcıda bulunan servislere erişimini sağlar. Servis sağlayıcı kendi sitesinde bulunan web servisleri tanımını servis kayıt birimine (service registry) kaydederek bu servisinin nasıl çağrılacağını belirtir.

- ***Servis İstemcisi (Service Requester)***

Servis sağlayıcısında bulunan web servislerini çağırarak kullanan istemci uygulamalarıdır. Web servisinin nasıl çağrılacağını belirler ve ilgili parametreleri servis kayıt biriminden arayarak bulur ve çağırır.

- ***Servis Kayıt Birimi***

Servis sağlayıcılarının yayınladıkları web servisi tanımlarını saklar ve aranıp bulunmasını sağlar. Servis sağlayıcıları servis kayıt birimini tarayarak istediği servisler hakkında bilgi alabilir. Servis kayıt birimi her servisin nasıl çağırılacağı konusunda tanım bilgileri içerir.

Web Servis Modeli-devam

Bir web servisi istemcisinin, servis sağlayıcıdan herhangi bir servisi çağırma aşamasındaki temel adımları:

1. Web servisi istemcisi (SOAP Client) servis kayıt biriminden (UDDI) web servisini bulur.
2. İstemci bir SOAP mesajı hazırlar. SOAP mesajı bir XML belgesidir.
3. İstemci SOAP mesajını web sunucusu veya uygulama sunucusunda çalışan SOAP istek dinleyicisine gönderir. İstek dinleyici gelen isteklere cevap veren sunucu programlarıdır.

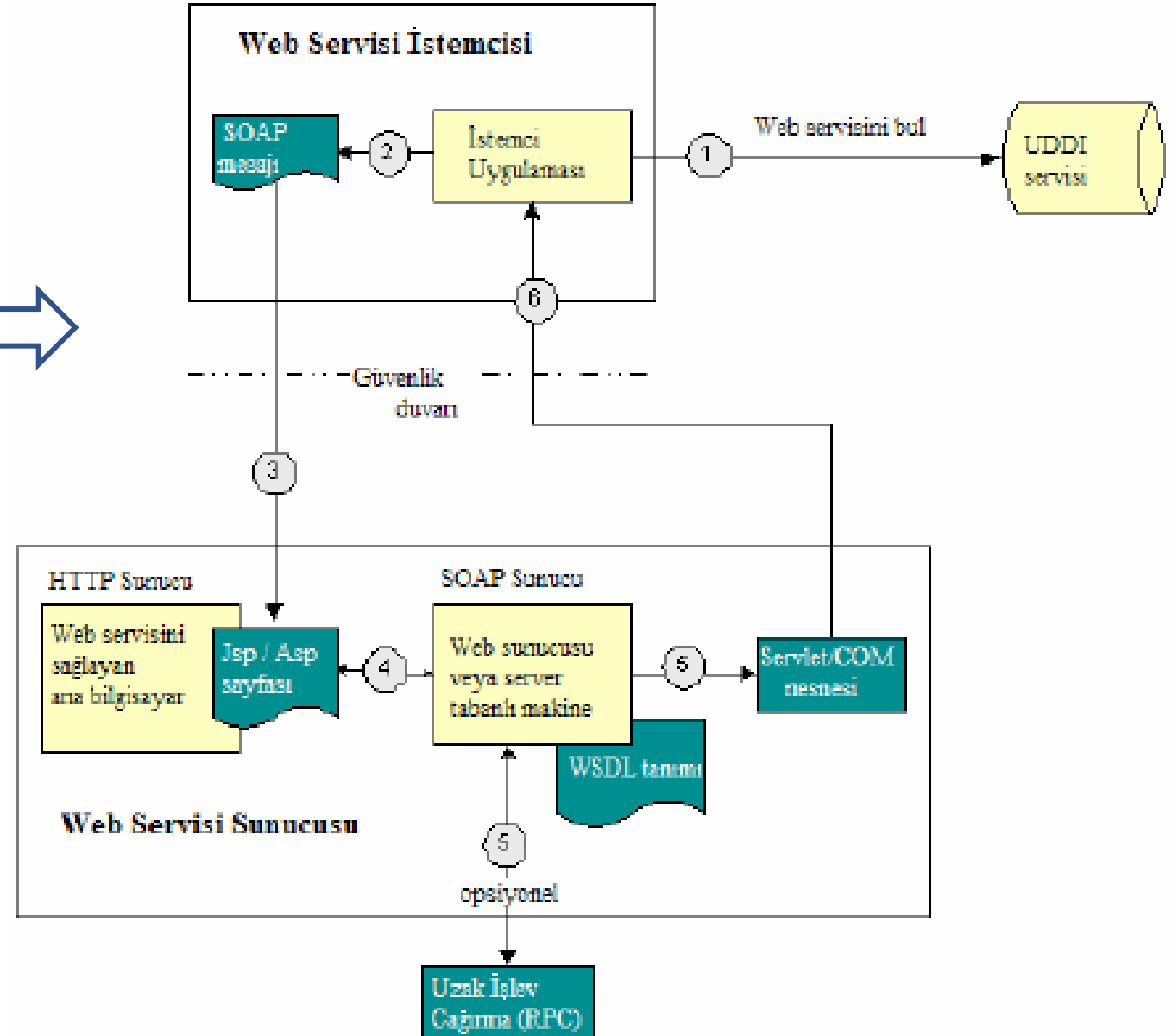
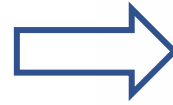
Bu programlar bir Java Sunucu Sayfaları (JSP: Java Server Pages), Aktif Sunucu Sayfaları (ASP: Active Server Pages), Ortak Gecit Ara yuzu (CGI: Common Gateway Interface) veya Internet Sunucusu Uygulama Programı Ara yuzu (ISAPI: Internet Server Application Program Interface) programıdır.

Web Servis Modeli-devam

4. SOAP sunucusu gelen SOAP mesajını çözümler ve gerekli parametreleri göndererek istenen nesnenin istenen yöntemini çağırır.
5. Çağrılan nesnedeki yöntem çalışır ve sonuçları SOAP sunucusuna gönderir. SOAP sunucusu gelen sonucu SOAP mesajı formatında biçimlendirerek istemciye gönderir.
6. İstemci gelen SOAP mesajının içindeki bilgileri alarak istekte bulunan programa gönderir.

Web Servis Modeli-devam

Web servisi istemci ve sağlayıcısı arasındaki temel işlemler

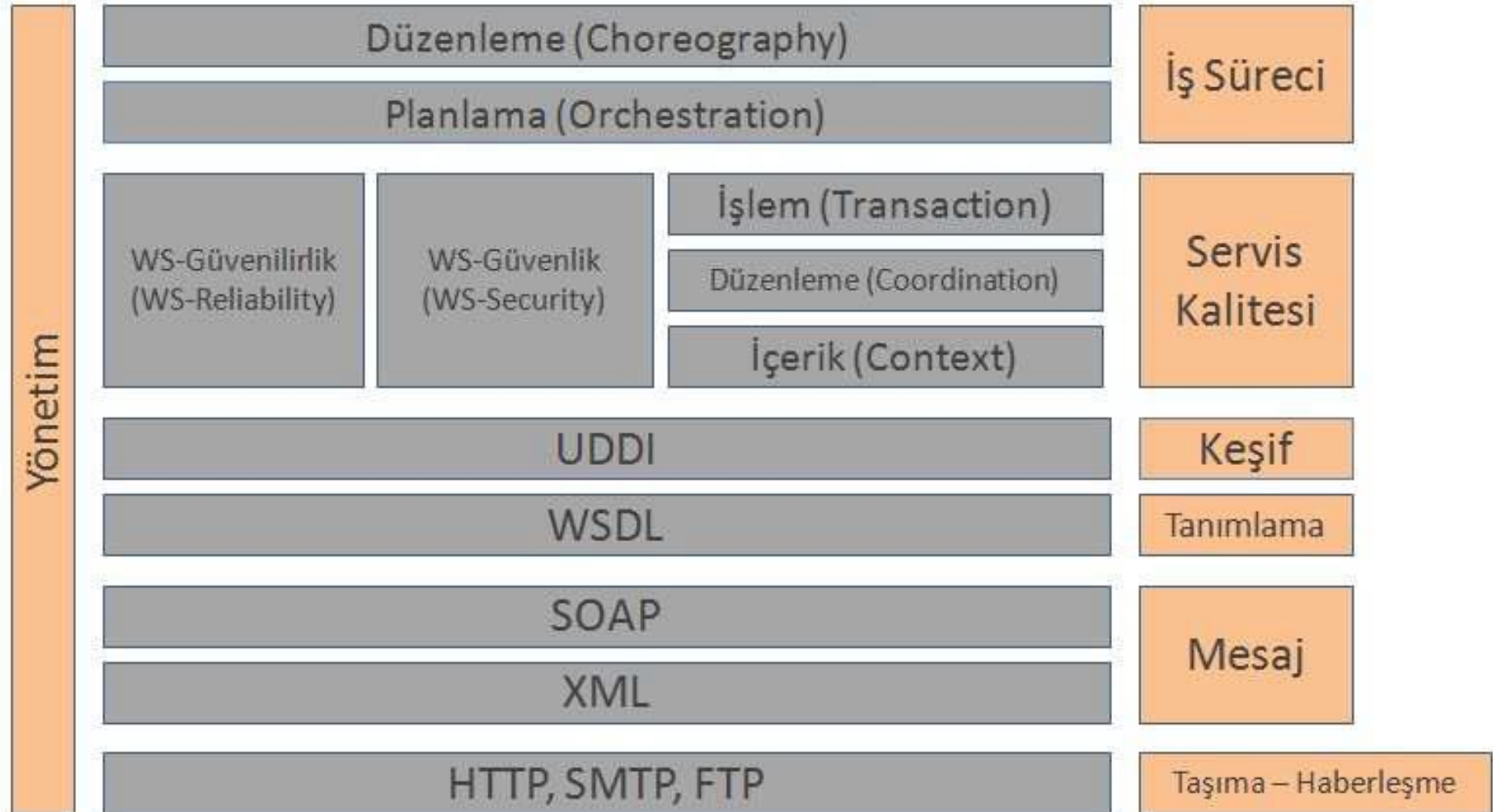


Web Servis Modeli-devam

- İlk olarak servis sağlayıcı web servislerinin tanımını yapar (WSDL) ve servis kayıt biriminde ya da servis istemcisinde yayınlar.
- Ardından web servis istemcisi talep ettiği servisi almak için kendi üzerinde yani yerel olarak ya da kayıt birimi üzerinde bu servisi arar, bulur.
- Son olarak web servis istemcisi elde ettiği bilgileri kullanarak bu servise istekte bulunur.
- İstemci servis sağlayıcıya istekte bulunurken SOAP istek mesajını kullanır ve geriye donen mesaj ise SOAP cevap mesajıdır.
- Veriler ve mesajlar HTTP üzerinden XML olarak aktarılmaktadır.

Web Servis Protokol Yığını ve Web Servis Standartları

Web servislerinin temelini oluşturan belirli ve tamamlayıcı standartlar yığını



Web Servis Protokol Yığını ve Web Servis Standartları

Web servisi mimarisindeki temel katmanlar



Web Servis Protokol Yığını ve Web Servis Standartları

- Web servislerine bakıldığında çok karmaşık gibi görünse de, web servislerinin çekirdeği olarak tanımlanan 3 standart vardır.
- Bu standartlar
 - web servis protokolü SOAP,
 - web servis tanımlama dili olan WSDL,
 - web servisleri yayınlamak ve keşfetmek için kullanılan UDDI'dır.
- Bu çekirdek standartlar XML ailesi özellikleri (XML ve XML Schema) ve IETF HTTP(S) standartları üzerine inşa edilmiştir.

Web Servis Protokol Yığını ve Web Servis Standartları

XML ve onunla ilgili standartları anlamak web servis çekirdek standartlarını daha iyi anlamak için temel oluşturur.

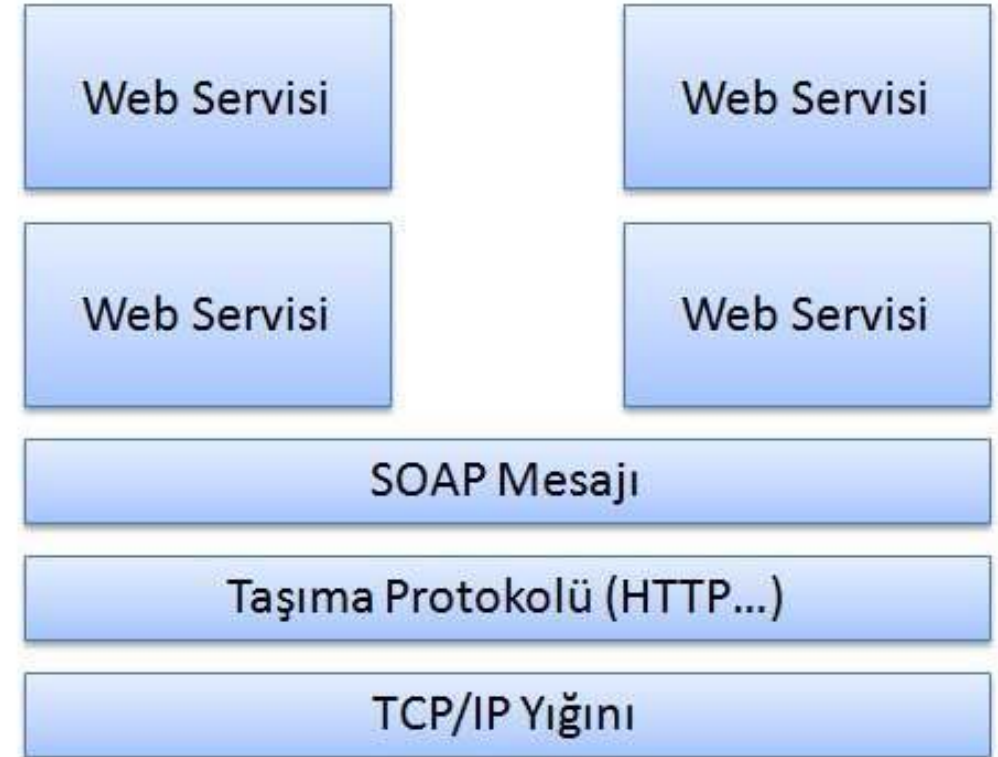
- **SOAP (Simple object access protocol)** : İnternet üzerinde web servislerini çalıştırmak için kullanılan protokol
- **WSDL (Web services description language)** : Web servislerini tanımlama dili
- **UDDI (Universal description, discovery and integration)** : Web servislerinin indekslenip bulunduğu kayıt servisi

SOAP (Simple Object Access Protocol)

- SOAP, web servisler tarafından kullanılan mesajlaşma protokolüdür. Dağıtılmış platformların birlikte çalışabilmesini sağlamak için tasarlanmıştır.
- Bu amaç diğer başarılı web protokolleri ile basitlik, esneklik, güvenlik duvarı kullanım kolaylığı,XML tabanlı mesajlaşma ile sağlanan platform bağımsızlığı gibi aynı prensipleri izleyerek gerçekleştirilir. Yeni bir teknolojik yenilik sunmak yerine, SOAP, web üzerindeki dağıtık iletişimleri standartlaştırmak amacıyla, yalnızca var olan internet teknolojilerini kullanımının bir şekilde derlenmesini göstermektedir.
- SOAP mevcut internet altyapısında olan router, firewall ve proxy sunucularda herhangi bir değişiklik yapmadan kolayca çalışmaktadır. SOAP genellikle web kaynaklarına ulaşmak için web tarayıcılar tarafından kullanılan HTTP üzerinden değiştirilir. HTTP SOAP mesajlarının gönderilip alınması için etkili bir yol oluşturmaktadır.

SOAP (Simple Object Access Protocol)-devam

- SOAP mesajlarının taşınması için HTTP den farklı protokollerin de kullanılabileceğini göstermektedir.
- SOAP'nin rolü mesajın nasıl biçimlendiğiyle ilgilidir, nasıl gönderildiğiyle ilgili değildir.
- HTTP iletişim protokolleri arasında en çok kullanılan protokoldur. Ancak SMTP ya da FTP protokolleri gibi diğer protokoller de kullanılabilir.



SOAP (Simple Object Access Protocol)-devam

- SOAP mesaj yapısı oldukça basittir. Mesajda Header (başlık) ve Body (gövde) olarak adlandırılan iki bölüm bulunmaktadır. Bu iki kısım Envelope (zarf) adı verilen eleman içerisinde bulunmaktadır. Aşağıda basit bir SOAP mesaj yapısı yer almaktadır.

```
<SOAP-ENV:Envelope xmlns:SOAP-  
ENV="http://schemas.xmlsoap.org/soap/envelope/">  
  <SOAP-ENV:Header>  
    ....  
  </SOAP-ENV:Header>  
  <SOAP-ENV:Body>  
    ....  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```


SOAP (Simple Object Access Protocol)-devam

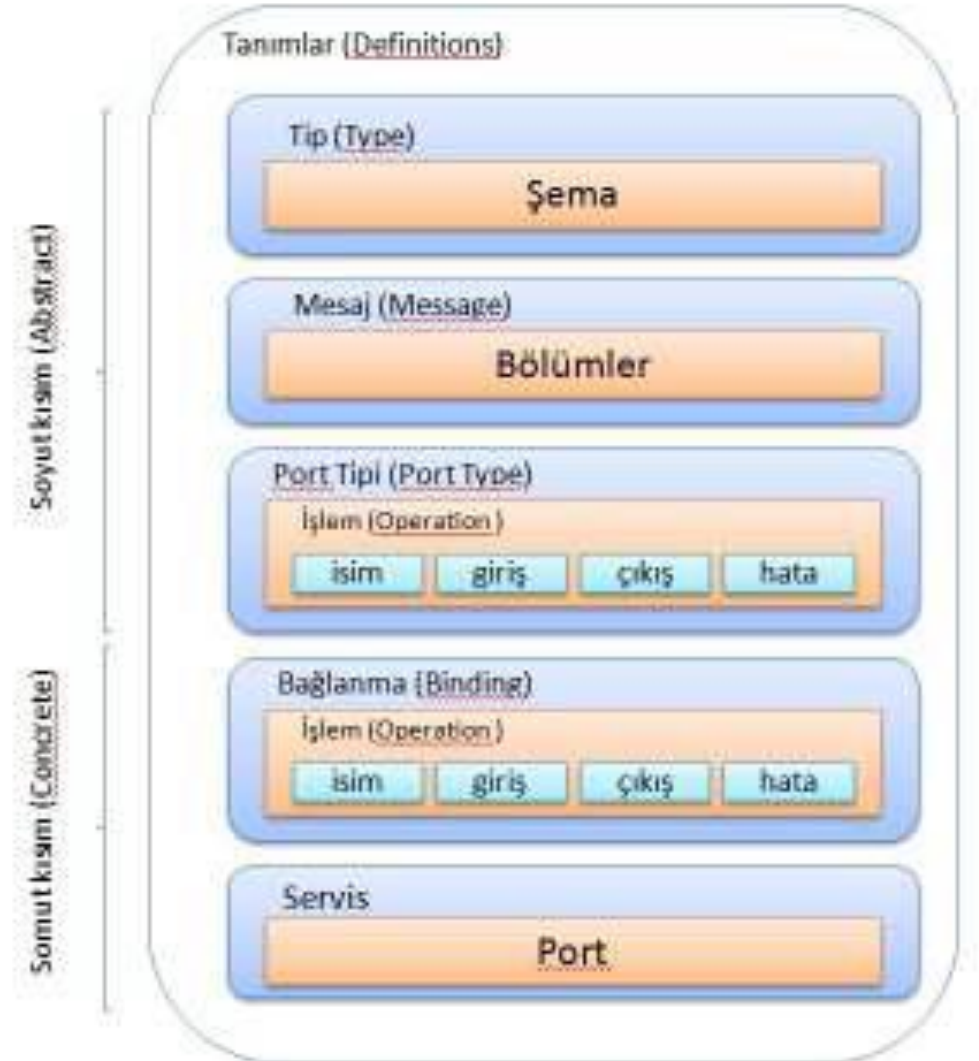
- İstemci SOAP uygulaması bir SOAP istek mesajı oluşturarak bu isteği SOAP sunucusunda tanımlanmış servis uc noktalarından (end point) birisi tarafından çalıştırılması için gönderir.
- SOAP sunucu ilgili servisi çalıştırdıktan sonra SOAP yanıt mesajı hazırlar.
- Hazırlanan SOAP yanıt mesajı istemciye iletilir.
- SOAP mesajı HTTP POST metodu veri paketinin içinde gönderilir.
- SOAP govdesi çağırılacak metot ve metodun icerdiği parametreleri icerir.
- SOAP govdesi icinde kodlanarak gonderilen bu mesaj, web servisi tarafından cozulur, gerekli parametreler ve metot çağırım bilgileri esliğinde islemlerini gerceklestirdikten sonra, istemciye döndüreceği cevap bilgileri için, yine SOAP protokolune uygun XML mesajlarını olusturur.
- Bu mesajlar HTTP uzerinden istemci uygulamaya ulasır, burada cozulur ve değerlendirilir. HTTP protokolunu desteklediğinden, SOAP sayesinde web servisine, platformdan bağımsız çağırımlar yapılabilir.

WSDL (Web Services Description Language)

- Bir uygulamanın bir web servisini kullanması için web servisinin nasıl çağırılacağı, ara yuzunun, hangi protokollerin ve kodlama standartlarının kullanıldığı belirtilmesi gerekir.
- WSDL web servisini tanımlayan bir XML belgesidir. Web servisi tanımı işlemler, giren ve çıkan mesaj formatları, ağ ve bağlantı noktası adresleri gibi bilgileri tanımlar.
- WSDL dokümanı bir servis sağlayıcı için uc nokta (endpoint) tanımlar.
- WSDL servis ara yuzunun genel tanımlarını içermektedir, böylece servis sağlayıcıyı çağırmak isteyen istemciler mesajların nasıl oluşturulduğunu bilirler.
- İstemci bu dosyayı kendi bilgisayarına indirdikten sonra, o web servisinin metotlarını kullanabilir.

WSDL Doküman Yapısı

- **Veri Tipi (Types)**: Mesajlarda kullanılacak veri tiplerini belirtir.
- **Mesaj (Message)**: İletişimde kullanılacak mesajları tanımlar.
- **Port Tipi (PortType)**: Web servisinin içerdiği metotları ve ilgili mesajları tanımlar.
- **Bağlanma (Binding)** : İşlem ve mesajlarda kullanılacak veri formatlarını tanımlar.
- **Port**: Binding ve web adresinden oluşan servis noktasını tanımlar. Web adresi servisin çalıştırılacağı URL'dir.
- **Servis**: Kullanılan bağlantı noktalarının kumesidir.



UDDI (Universal description discovery and integration)

- Bir web servisini kullanmak için kullanıcının web servisi sağlayan kurumları ve bu kurumların verdikleri web servislerinin neler olduğunu bilmesi gerekir.
- UDDI kısaltmasında geçen Evrensel, Tanım, Bulus ve Butunleştirme kelimelerinin ifade ettiği gibi UDDI, platformdan bağımsız, XML tabanlı kayıtları tutar ve internet üzerinde yayınlar.
- Ayrıca OASIS (Yapılandırılmış Bilgi Standartları Orgutu) tarafından desteklenen bir girişim olan UDDI, kurumların sağladıkları servisleri yayınlamasını, bu bilgilerin daha sonra diğer kurumlarca taranıp bulunmasını, servislerin ve yazılım uygulamalarının internet üzerinde nasıl iletişimde olduğunu tanımlamasını sağlayan bir standarttır.

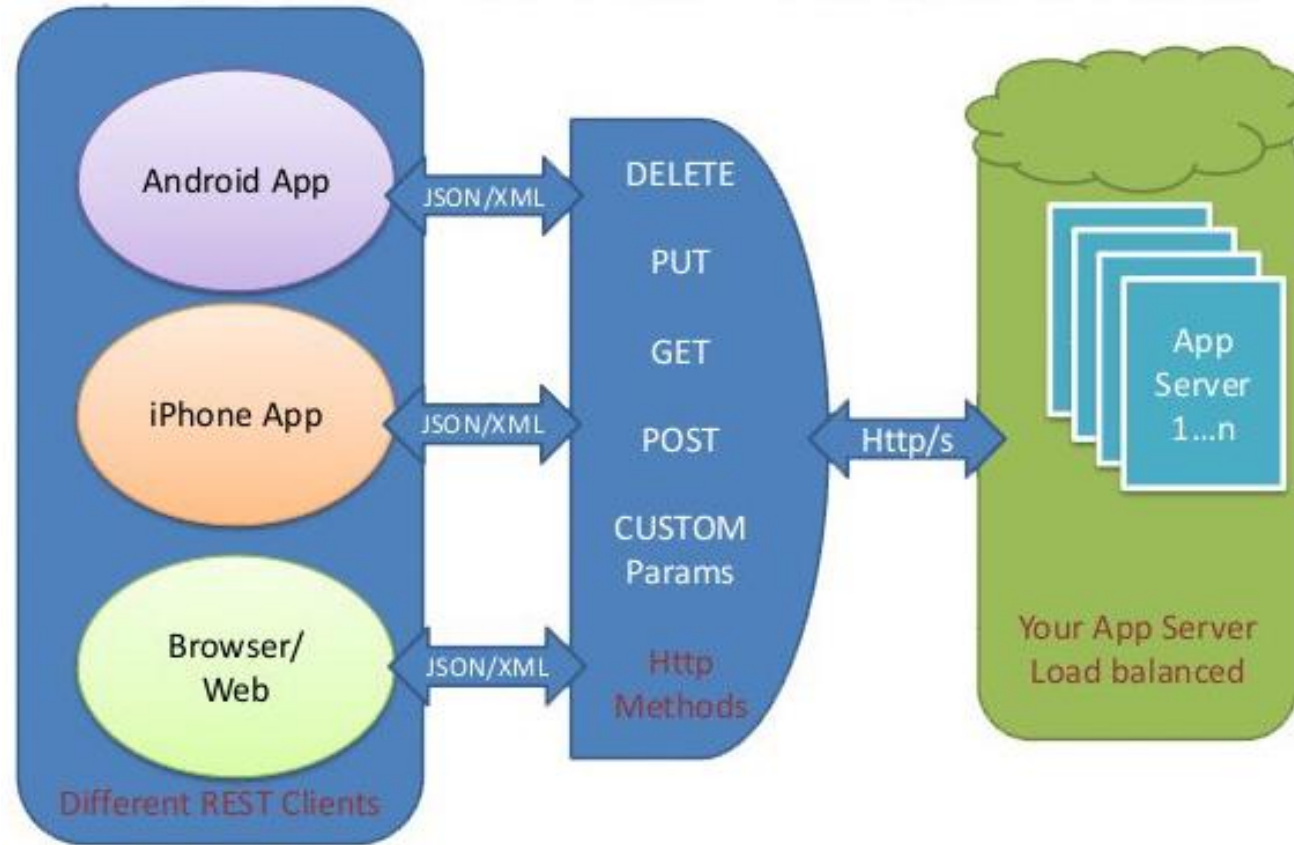
UDDI (Universal description discovery and integration)-devam

- UDDI Kurum Kayıt Servisi (UDDI Business Registry) kurum ve web servisleri bilgilerini saklayan sunuculardır.
- Bu sunucular servis sağlayıcılarından gelen bilgilerini kendi veritabanlarına kayıt ederek diğer kurumların erişimine acar. Su anda aktif olarak çalışan kurum kayıt sunucuları uddi.microsoft.com ve uddi.ibm.com 'dur.
- Bu sunucular kendilerine kayıt edilen bilgileri diğer sunuculara da kopyalayarak kolayca hızlı bir şekilde erişilmesini sağlarlar.
- UDDI sunucuları kurum ve servis kayıt, guncelleme ve tarama işlemlerini web servisleri (SOAP mesajları) ile gerçekleştirir

REST API (Representational state transfer)

- Web protokolleri ve teknolojilerini kullanan bir dağıtık sistemdir.
- REST prensiplerini sağlayan sistemler, RESTful olarak sınıflandırılır.
- SOAP ve WSDL tabanlı Web servislerine spesifikasyon olarak daha basit bir alternatif olarak geliştirilmiştir.
- RESTful sistemler çoğunlukla HTTP protokolü üzerinden Web tarayıcıları tarafından sayfaların transferinde de kullanılan HTTP methodları (GET,POST,PUT,DELETE v.s.) ile haberleşirler.
- Aktarılan verinin formatı HTML, JSON, XML ya da farklı bir tipte olabilir.
- Aktarılan verinin tipi ve özellikleri istemci ve sunucu tarafından HTTP protokolünde yer alan İngilizce: content-type (içerik tipi) ve benzeri metaveri ile tanımlanır.

REST API Mimari



REST API Mimari Özellikleri

- REST mimarî tarzıyla etkilenen mimarî özellikler şunlardır:
 - Performans - bileşen etkileşimleri, kullanıcının tespit ettiği performans ve ağ verimliliğinde dominant faktör olabilir.
 - Çok sayıda bileşeni ve bileşenler arası etkileşimi ölçeklenebilirlik yapmak.
 - Arayüzlerin sadeliği
 - Değişen ihtiyaçları karşılamak için (uygulama yürütülürken dahi) bileşenlerin değiştirilebilirliği
 - Hizmet vekillerince (service agents) tarafından bileşenlerarası haberleşmenin görünürlüğü
 - Program kodunu veriyle beraber hareket ettirerek bileşenlerin taşınabilirliği
 - Sistem seviyesinde bileşenlerde, bağlantılarda veya verideki hatalara rağmen hataya karşı direnç güvenilirliği

REST API Mimari Sınırlamaları

- REST'in mimarî özellikleri, bağlantılara, bileşenlere ve veri elementlerine belli etkileşim sınırlamaları uygulayarak erişilir. Aşağıdaki özelliklere sahip uygulamalar, "RESTful" olarak tasvir edilir.
- Bu sınırlamalarla uyumlu olma ve böylece REST mimarî stiliyle uyuşma, her çeşit hipermedya sisteminin performans, ölçeklenebilirlik, sadelik, değiştirilebilirlik, görünürlük, taşınabilirlik ve güvenilirlik gibi cazip fonksiyonel olmayan özelliklere sahip olmasını sağlar.
 - İstemci-sunucu
 - Durumsuzluk
 - Ön belleklenebilme
 - Tabakalı sistem
 - Talep üzerine kod
 - Değişmeyen arayüz

XML (Extensible markup language)



- XML web üzerinde veri değişimi için standart haline gelen metin temelli bir isaretleme dilidir.
- HTML'den farklı olarak XML etiketleri verinin nasıl gösterileceğini değil, verinin nasıl tanımlanacağını gösterir. İnternet üzerinde platformdan bağımsız veri alıs-verisinde XML kullanılmaktadır.
- XML'de veriler etiket ve sonlandırıcı etiket arasında paketlenerek gönderilir.
- XML'de veriler iç içe etiketlerden oluştuğu için XML'in hiyerarsik bir yapısı vardır.
- XML, İnternet Komitesi (W3C) tarafından ortaya çıkarılmıştır.
- XML farklı uygulamalarda, verilerin bütünleştirilmesinde arac olarak kullanılır. Yazılım firmaları XML'e büyük destek vermektedir.
- XML, Firmadan Firmaya (B2B: Business-to-Business) olan uygulamalarda standart bir dil ve belge türüdür.

XML (Extensible markup language)-devam

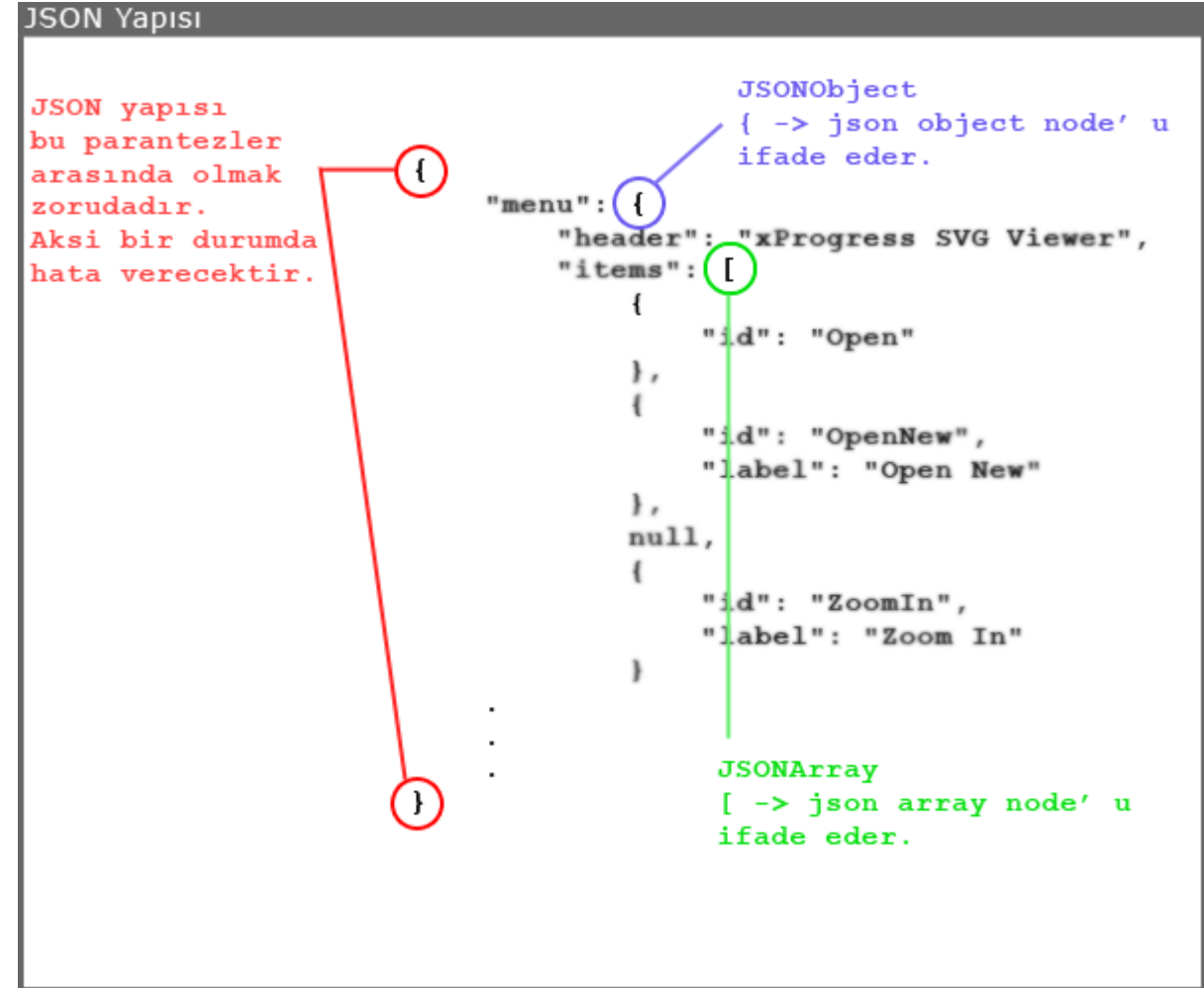
- XML metin halinde olup ikili (binary) formatında olmadığı için standart metin editorlerinden gorsel (visual) gelistirme ortamına kadar herhangi bir araçla olusturulup, gelistirilebilir. Diğer taraftan bir veritabanı büyük miktarda XML verisi depolayabilir.
- XML, veri tanımlaması sayesinde, ne cesit verinin (metin, sayı, vb.) olduğunu gosterir. Dsaretleme etiketleri bilgiyi belirttiği ve veriyi kısımlara ayırdığı için bir e-posta programı onu isleyebilir, bir arama programı belirli kişilere yollanacak mesajları arayabilir, bir tarayıcı ile de XML cozumlenebilir. Kısaca, bilginin farklı bolumleri tanımlandığından bunlar farklı uygulamalarda farklı sekillerde kullanılabilir.
- XML uygulama, dil, isletim sistemi gibi kısıtlamalara bağılı değildir. Su anda hemen her isletim sistemi ve uygulama XML belgelerini okumak ve yazmak için yerlesik ozelliklere sahiptir. Butun XML belgeleri unicode tabanlıdır. Dsletim sistemleri farklı kod sayfalarını (code page) kullandıkları için farklı dillerdeki belgeleri görüntuleyemezler. Ancak, unicode kullanımı farklı dillerdeki belgelerin görüntulenmesine olanak tanır.

JSON (JavaScript Object Notation) { j s o n }

- **JSON** (JavaScript Object Notation) hafif bir veri değişim formatıdır. İnsanların okuyup yazabilmesi kolaydır. Makinaların tarayıp, yaratabilmesi kolaydır.
- JSON, tamamen programlama dillerinden bağımsız, ancak C türevi dillere (C, C++, C#, Java, JavaScript, Perl, Python ve daha pek çoğu), yazılış bakımından çok benzeyen bir veri tanımlama formatıdır. Bu özellikler, JSON'u veri değiş tokuşu için ideal hale getirmektedir.
- JSON iki yapı üzerine kurulmuştur:
 - **İsim/değer çifti koleksiyonu:** Çeşitli programlama dillerinde bu, "*object*, record, struct, dictionary, hash table, keyed list veya associative array" olarak da tanımlanmıştır.
 - **Sıralı değer listesi:** Çoğu programlama dilinde bu, "*array*, vector, list veya sequence" olarak tanımlanır.

JSON-devam

- JSON yapısal olarak 5 veri tipine sahiptir:
 - Number
 - String
 - Array: "name":[...] şeklinde kullanılır.
 - Boolean: true|false
 - Object: "name":{..} şeklinde kullanılır.
 - Null

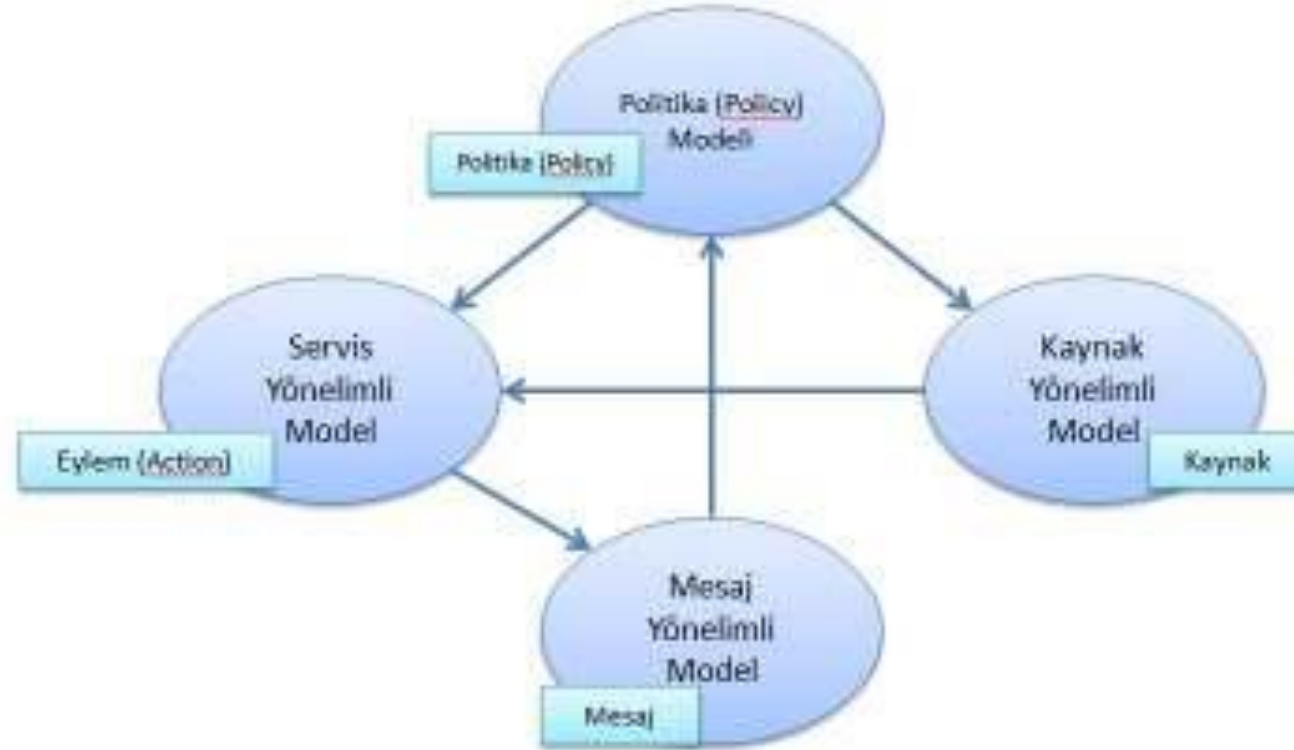


Web Servislerin Özellikleri

- Web servisleri bağımsız uygulamalardır.
- Web servisleri dilden bağımsız ve birlikte çalışabilir özelliktedir.
- Web servisleri doğası gereği açık ve standartlara dayalıdır, kolay ve hızlı dağıtılır.
- Web servisleri düşük bağımlıdır (loose coupling).
- Web servisleri kendi kendini tanımlar (self-describing).
- Çalışma zamanında entegrasyon sağlarlar.
- Etkin uygulama geliştirme sağlarlar.
- Kapsulleme (Encapsulation) özellikleri vardır.

Web Servis Mimarileri

- Web servis mimarilerinin şekilde gösterildiği üzere 4 modeli bulunmaktadır.



SOA (Servis yönelimli mimari)

- Servis yönelimli mimari (SOA), birçok uygulamanın kendi modüllerini veya fonksiyonlarını (bunlara servis deniyor) başka uygulamalar tarafından kullanılacak şekilde tasarladığı bir mimari kavramdır.
- SOA, bağımsız servisler olarak iş mantığının sağlanmasına olanak tanır. İş modellerinin çok hızlı bir şekilde değişmesinden dolayı birçok kuruluş servis yönelimli mimari ile gelişmektedir.
- SOA'da servislerin keşfedilebilir olması, kendi tanımlanabilmesi, birlikte çalışabilirliği, düşük bağımlılığı ve birleştirilebilir özelliklerde olması, işyerlerinde bir tasarım yapılırken bu teknolojinin daha kolay kullanılabilmesine olanak tanır ve radikal değişimler için bir potansiyel oluşturur.
- Service yönelimli mimari, dağıtık uygulamalar için ideal bir mimaridir. Bugün pek çok web servis uygulaması asırı bağımlı alt sistemlerden oluşmaktadır. Bu durum, alt sistemlerden birinde yapılacak bir değişikliğin uygulamanın tümünü olumsuz etkilemesine neden olmaktadır. Bu kırılgan yapı, bakım maliyetlerinin yüksek olmasındaki birincil sebeptir. Ayrıca bu durum, değişen iş gereksinimlerinin karşılanmasını ve ileride yapılması muhtemel modifikasyonları zorlaştırmaktadır.

Mesaj yönelimli model (Message oriented model)

- Mesaj yönelimli model, mesajların içeriğiyle ya da nedenleriyle ilgilenmeden yalnızca mesajlarla, mesaj yapısıyla ve mesajların iletimiyle ilgilenir.
- Bu modelde, bir mesajın içeriğinin anlamsal önemiyle ya da diğer mesajlarla ilişkisiyle ilgilenilmez.
- Fakat, mesaj yönelimli model mesajın yapısıyla, mesajın göndereni ve alıcısı arasındaki ilişkiye ve mesajın nasıl iletildiği üzerine yoğunlaşılır

Diğer Web Servis Modelleri

- **Kaynak yönelimli model (Resource oriented model)**
 - Bu model mevcut kaynaklarla ve bu kaynakların sahibiyle ilgilenir.
 - Kaynaklar web servislerinin ve web in altında yatan temel kavramlardır. Örneğin, bir web servisi bu model için önemli bir kaynağın belirli bir türüdür.
- **Politika modeli (The policy model)**
 - Politika modeli servislerin ve ajanların davranış kısıtlamaları üzerine yoğunlaşır.
 - Politikalar aktif kaynakların yanı sıra dokümanlara (servis tanımlamaları gibi) da uygulanabilir.

WebSocket Kavramı

- **WebSocket**, tek bir [TCP](#) bağlantısı üzerinden tam çift yönlü iletişim kanalı sağlayan bir bilgisayar [iletişim protokolüdür](#).
- WebSocket protokolü [IETF](#) tarafından 2011 yılında [RFC](#) 6355 ile standart hale getirilmiş ve WebIDL içerisindeki WebSocket API [W3C](#) tarafından standart hale getirilmektedir.
- WebSocket, web tarayıcılarında ve web sunucularında uygulanmak üzere tasarlanmıştır, fakat herhangi bir istemci veya sunucu uygulaması tarafından uygulanabilmektedir.
- WebSocket protokolü, TCP tabanlı bağımsız bir protokoldür. HTTP ile tek ilişkisi, HTTP sunucuları tarafından bir Upgrade isteği olarak yorumlanmasıdır.

WebSocket-devam

- WebSocket protokolü, sunucuya ve sunucudan gerçek zamanlı veri aktarımını sağlayarak, tarayıcı ile web sunucusu arasında etkileşimi sağlamaktadır.
- Bu, sunucunun istemci istemeden tarayıcıya içerik gönderebileceği ve bağlantıyı açık tutarak istediği zaman mesaj alabilmesini veya gönderebilmesini sağlayan standart bir yöntem ile sağlanmaktadır. Bu şekilde, tarayıcı ile sunucu arasında iki yönlü devam eden bir iletişim gerçekleşebilmektedir.
- İletişim TCP 80 portu (veya TLS ile şifrelenmiş bağlantılarda 443 portu) üzerinden gerçekleşmektedir ve bu, güvenlik duvarı kullanarak Internet web trafiğini engelleyen ortamlar için bir avantaj olmaktadır.
- HTTP'den farklı olarak, WebSocket protokolü tam çift yönlü iletişim sağlamaktadır. Ayrıca WebSocket TCP protokolünün üzerinde mesaj akışına imkan tanımaktadır. TCP tek başına, mesaj kavramı kullanmadan byte akışı ile ilgilenmektedir.

WebSocket-El sıkışma

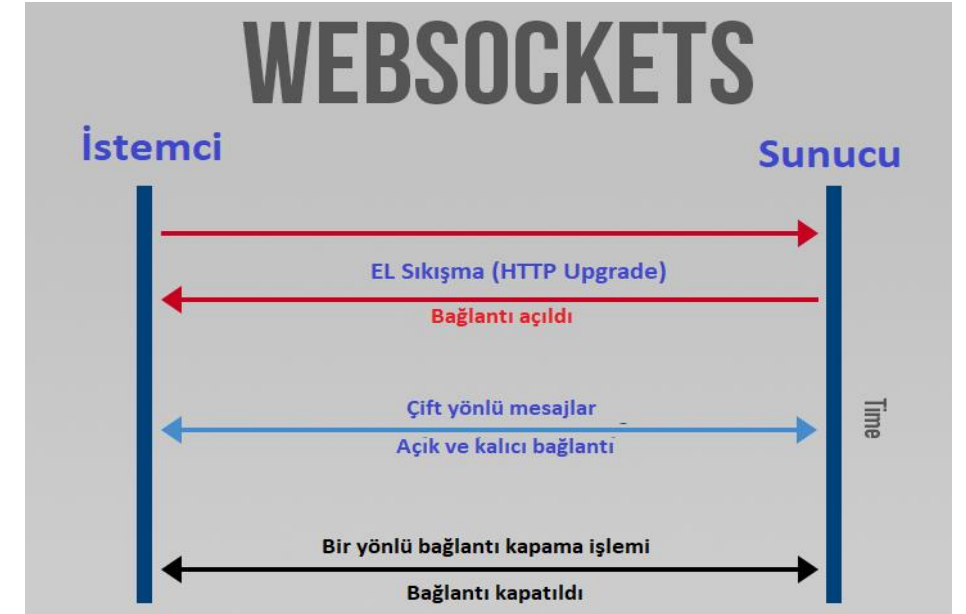
- Bir WebSocket bağlantısı kurmak için istemci, yandaki örnekte gösterildiği gibi sunucunun WebSocket el sıkışma cevabı gönderdiği bir WebSocket el sıkışma isteği göndermektedir.

- İstemci isteği:**

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
Origin: http://example.com
```

Sunucunun cevabı:

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGwk=
Sec-WebSocket-Protocol: chat
```



WebSocket-El sıkışma-devam

- El sıkışma, sunucuların aynı port üzerinden hem HTTP bağlantılarını hem de WebSocket bağlantılarını işleyebilmesi açısından HTTP'ye benzemektedir. Bağlantı kurulduğunda, iletişim, HTTP protokolüne uymayan iki taraflı bir ikili protokole dönüşmektedir.
- Upgrade başlığına ek olarak, istemci [base64](#) ile kodlanmış rastgele bytelardan oluşan bir Sec-WebSocket-Key başlığını da göndermektedir. Sunucu ise Sec-WebSocket-Accept başlığı ile anahtar değerinin özeti ile cevap vermektedir.
- Bu yöntem, önceki bir WebSocket trafiğinin tekrar oynatılmasını engellemeye çalışmaktadır ve herhangi bir kimlik doğrulama, gizlilik veya bütünlük kontrolü sağlamamaktadır.
- Özet fonksiyonu 258EAF5-E914-47DA-95CA-C5AB0DC85B11 sabit değerini Sec-WebSocket-Key başlığındaki (base64 kodlaması çözülmemiş) değere eklemekte, [SHA-1](#) özet fonksiyonunu uygulamakta ve sonucu base64 kullanarak kodlamaktadır.

WebSocket-El sıkışma-devam

- Bağlantı kurulduktan sonra, istemci ve sunucu WebSocket verisini tam çift yönlü modda gönderebilmektedir.
- Veri, [payload](#) tarafından takip edilen küçük bir başlık ile minimal bir şekilde çerçevelenmektedir.
- WebSocket iletişimi, tek bir mesajın birkaç veri çerçevesi içerisine dağılabileceği mesajlarla tanımlanmaktadır.
- Bu, verinin ilk kısmının bilindiği fakat mesajın toplam büyüklüğünün bilinmediği durumlarda, mesajların gönderilebilmesine izin vermektedir (FIN biti ile işaretlenip sona erişilene kadar veri çerçevelerini arka arkaya göndermektedir).
- Protokol üzerinde yapılan eklentilerle, farklı veri akışlarının eş zamanlı olarak çoklanması için de kullanılabilir.

WebSocket-Proxy Geçişi

- WebSocket protokolü istemci uygulamaları, web tarayıcısının hedef makine ve porta bağlanırken bir vekil sunucu kullanıp kullanmadığını tespit etmeye çalışmaktadır ve eğer kullanıyorsa, kalıcı bir tünel oluşturmak için HTTP CONNECT metodunu kullanmaktadır.
- WebSocket protokolünün kendisi vekil sunucuların ve güvenlik duvarlarının farkında olmasa da, HTTP'ye uygun bir el sıkışmayı desteklemektedir ve böylece HTTP sunucularının varsayılan HTTP ve HTTPS portlarını (80 ve 443) bir WebSocket ağ geçidiyle veya sunucusuyla paylaşmasına izin vermektedir.
- WebSocket protokolü, WebSocket ve Güvenli WebSocket bağlantılarını belirtmek için ws:// ve wss:// örneklerini tanımlamaktadır.
- Eğer şifreli WebSocket bağlantısı kullanılıyorsa, Güvenli WebSocket bağlantısında Transport Layer Security (TLS) kullanılması, tarayıcı bir vekil sunucu kullanacak şekilde yapılandırıldığında, HTTP CONNECT komutunun çalıştırılmasını sağlamaktadır.

Javada WebSocket Kullanımı

- **onOpen** : WebSocket bağlantısı istemci ile sunucu arasında gerçekleştiği sırada bu fonksiyon çalışır. Burada kullanıcının sisteme bağlantı esnasında yapılması gerekenler yapılabilir. Bir class'ın yapılandırıcısı gibi düşünebilirsiniz.
- **onClose**: WebSocket bağlantısı kesildiği zaman yani kullanıcı server ile bağlantısını kestiği zaman bu fonksiyon çalışır.
- **onMessage**: Servera websocket aracılığı ile mesajın gelmesi durumunda çalışacak olan fonksiyondur. Gelen mesaj hedef usera burada gönderilebilir.
- **onError**: Serverde karşılaşılan hataların return olacağı fonksiyondur