

## SOA - Web Service - Restful Service - HTTP methods kavramlarını örneklerle açıklayınız.

**SOA:** SOA(Service-Oriented Architecture) uygulama geliştirme ve veya bunların bir araya getirilmesi sürecinde bir aşamadır. Interfaceleri kullanarak yazılım bileşenlerini yeniden kullanılabilir hale getirmenin bir yolunu söyler. SOA'nın 6 temel bileşeni vardır:

**Service:** Her serviste, genelde, ayrı bir işlevsellik vardır.

**Policies:** Bir servisin kullanıcılar tarafından kullanılabilmesi için kurallar koyar.

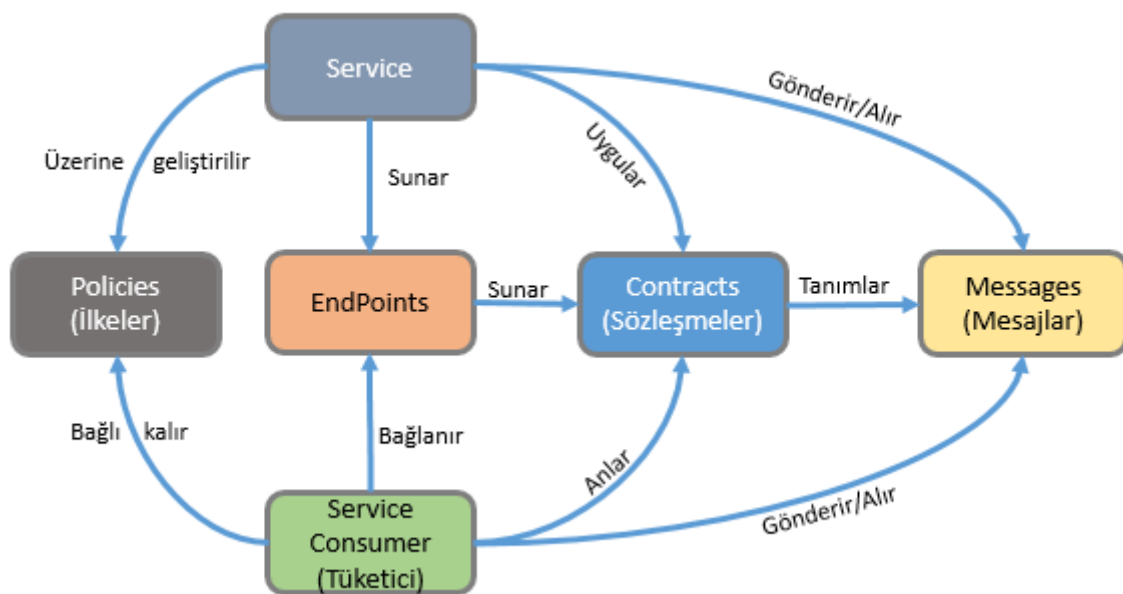
**EndPoints:** Servisin bulunduğu adrestir. Contractlar EndPointlerden çıkar.

**Contracts:** Servisten çıkan mesajların tümü bir sözleşme içinde aktarılır.

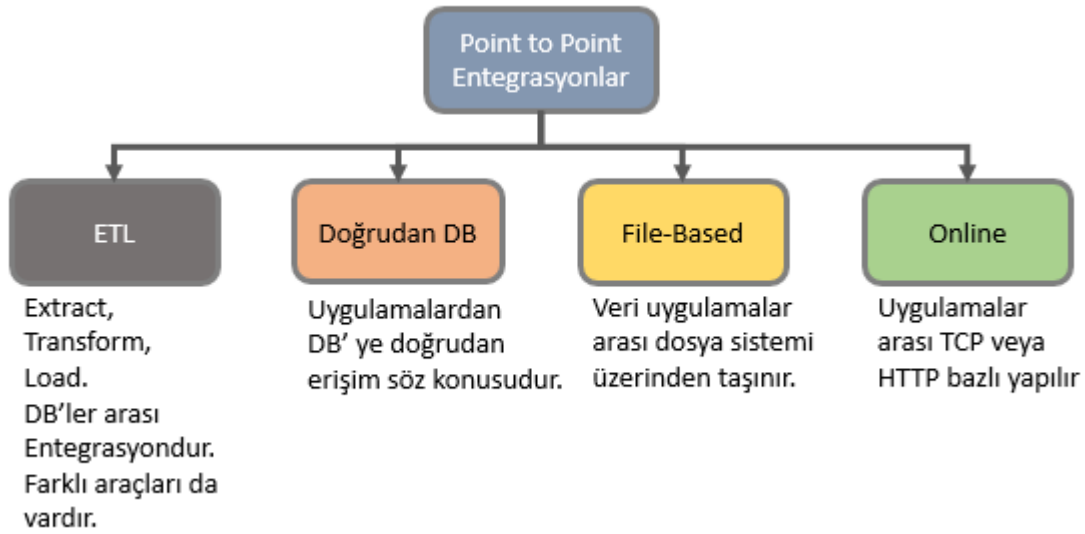
**Messages:** SOA'nın iletişim birimidir.

**Service Consumer:** Mesajlar yoluyla servisle iletişim kuran, servisin hizmetinden faydalanan yapılarıdır. Bu başka bir servis olabilir ya da bir uygulama da olabilir.

Bu bileşenler arasında yakın bir ilişki vardır. Kısaca özetlemek gerekirse; servis, sözleşmeler aracılığıyla işlevler ve bunlara bağlı hareketler sunar. Bu sözleşmeler mesajları, EndPointler üzerinden taşır. Ayrıca bir servis, standart hale gelmiş policylere uymalıdır. Consumer ise sözleşmeleri ve dolayısıyla da mesajları alarak servisin içeriğine ve verdiği hizmete ulaşır.



SOA, özellikle point-to-point entegrasyonu kullanan, yani iki uygulamayı doğrudan birbirine bağlamak için özelleştirilmiş kod kullanan sistemlerde bağımlılıklara çözüm getirir. Birçok sistem, gelen talep doğrultusunda zamanla genişler ve sistemler arası bir paylaşım gerektirir. Bu paylaşımlar da entegrasyonlarla yapılır ve point-to-point gibi entegrasyonlar kullanılırsa karmaşıklık sürekli artar.



Dolayısıyla, SOA'nın yukarıdaki diyagramından, çok daha etkili bir paylaşım sistemi sunduğunu anlayabiliriz.

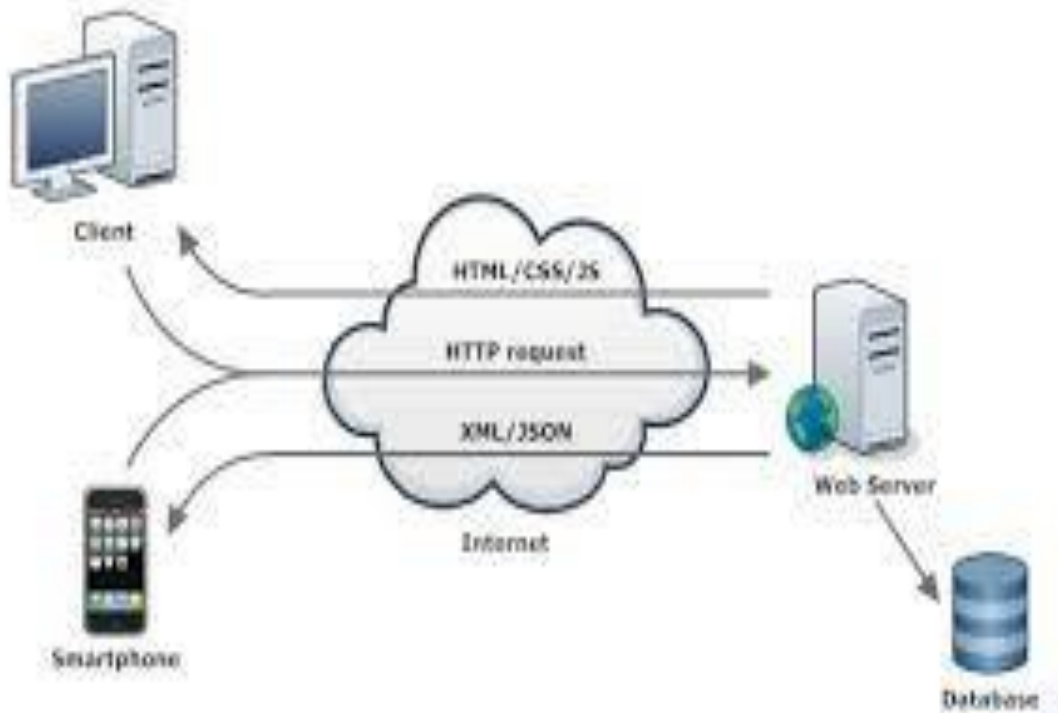
SOA'nın birçok kullanım alanı vardı. Birkaç örnek verecek olursak,

1. Birçok ordu ve hava kuvvetleri, durumsal farkındalık sistemlerini(çevresel unsurların algılanması, pilotun kendi aracı ve diğer hava araçlarının durumları hakkında bilgi sahibi olması vs) konuşlandırmak için SOA kullanır.
2. Sağlık hizmetlerinin sunumunun geliştirilmesi
3. Bir uygulama, cihaz içindeki GPS'i kullanmak için GPS'e ihtiyaç duyar. SOA'nın mobil çözümlerinden biri budur.

**Web Service:** Diğer sistemlere HTTP protokolü ile hizmet veren yapıdır. XML(verileri tutmak ve farklı sistemler arasında paylaşmak için XML denen metin işleme dili ile hazırlanan dökümanlar, .xml uzantılı dosyalarda depolanır), JSON(veri yapılarını bir dilden diğer dillere ya da platformların anlayabildiği formatlara çevirmeye yarayan veri değişim biçimi), CSV(veri girme, derleme, işleme amaçlarıyla oluşturulan tablolara izin veren dosyalar, farklı programlar

arasında veri alışverişine imkan tanır) gibi ortak biçimlerle veri alışverişi sağlar. Bu durumdan dolayı da Web Service, çoğunlukla, farklı sistemler arasında, önceden belirlenmiş ortak bir biçim ile veri transferi yapar.

Örneğin, A diliyle yazılmış bir uygulama, kullanıcı talebi üzerine, B dilini baz alan bir sistemde kullanılmak istenir. Sistemlerin kullanacağı ve kullanılması zorunlu olan ortak bir biçim ile bu işlem sağlanabilir.



**RESTful Service:** RESTful Service, REST mimarisiyle hazırlanmış web servis yapılarıdır. Peki, REST nedir? REST, programlama dilinden bağımsız olarak web servis oluşturmaya yarar. Web Service gibi, HTTP protokolünü ve bunun barındırdığı http yöntemleri(bir sonraki madde) ve durum kodlarını(işlem sonucu ile ilgili bilgi almak için kullanılır, örneğin herkesin bildiği 404 NOT FOUND kodu istenen işlem yoksa kullanılır) kullanarak taleplere yine XML, JSON ya da CSV olarak karşılık verir.

REST'in en önemli avantajı, mesela SOAP(web servis yöntemlerinden biri) için zorunlu olan WSDL(Web Service Description Language – web servisi kullanabilmek için gerekli bilgileri(web servis içindeki metod, parametre vs) veren standart) gerekliliğinin olmamasıdır.

**HTTP Methods:** Sunucu ile kullanıcı arasındaki iletişim yöntemini belirleyen metotlara HTTP metotları denir.

**GET:** Verileri listelemek ya da görüntülemek için kullanılır. Örneğin;

GET: /orders/{id}

Get ile sipariş detaylarını çağırabiliriz.

**POST:** Veri eklemek ya da mevcut veriyi güncellemek için kullanılır. Örneğin;

POST: /orders/

POST ile yeni sipariş oluşturabiliriz.



**PUT:** Put da veriyi güncellemek için kullanılır. POST'tan farklı olarak idempotenttir. Yani bir defa çağırıldığında alınan sonuç ile birden fazla çağırıldığında alınan sonuç aynıdır. Örneğin;

PUT: /orders/{id}

PUT ile sipariş güncelleyebiliriz.

**PATCH:** Verinin bir kısmını günceller. Örneğin;

PATCH /adresses/1

Id'si 1 olan adresi döner.

**DELETE:** Veriyi siler. Örneğin;

DELETE/adresses/id

Id'si verilen veriyi siler.