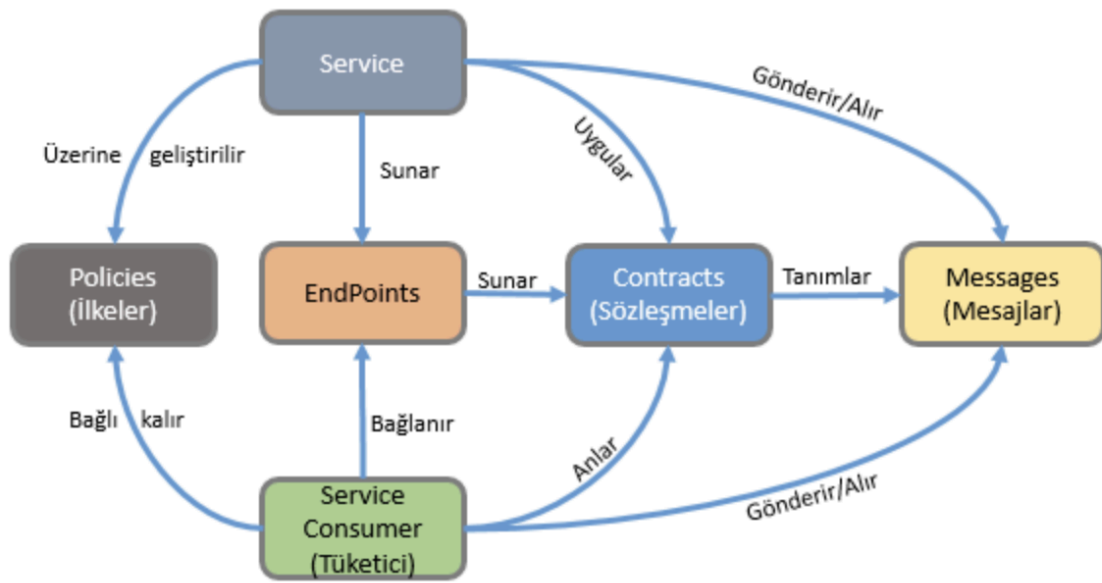


## SOA

SOA (Service-Oriented Architecture), yazılım sistemlerini modüler ve esnek bir şekilde tasarlamak için kullanılan bir mimari yaklaşımdır. SOA, bir organizasyonun farklı yazılım bileşenlerini, iş süreçlerini ve hizmetleri birbirine bağlamak ve entegre etmek amacıyla geliştirilmiştir. Bu mimari, işlevselliği hizmetlere ayırmak ve bu hizmetleri birbirine bağlayarak geniş ölçüde yeniden kullanılabilir, esnek bir sistem oluşturmayı hedefler.

Örnek olarak bu diagrama bakabiliriz.

<https://devnot.com/2015/soa-service-oriented-architecture-nedir/> 'den alınmıştır.



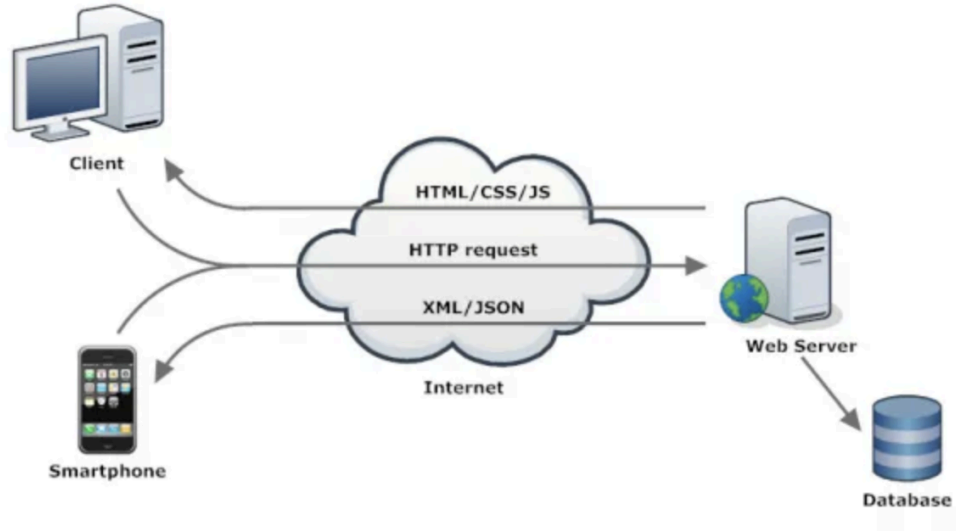
Her bir servis(Service) sözleşmeler(Contracts) yoluyla iş süreçleri ve bunlara bağlı davranışlar sunar ki bu sözleşmeler keşfedilebilir EndPoint' ler üzerinden taşınabilecek mesajları(Messages) tanımlarlar. Bir servisin davranışı(Behavior), endüstriyel anlamda standartlaşmış çeşitli ilkeler(Policies) ile uyumlu olmak zorundadır. Servisleri tüketen taraflar(Service Consumers) ise, sözleşmeler ve mesajlar yoluyla tanımlanmış iş fonksiyonelliklerine(Business Functions) ulaşmaktadır.

## Web Service

Web servis HTTP protokolü üzerinden diğer sistemlere/cihazlara hizmet veren yapılardır. Farklı sistemlerin birbirleriyle iletişim kurmasını sağlayan ve genellikle XML veya JSON formatında veri alışverişi yaparlar. REST ve SOAP olmak üzere 2 çeşit web service olduğunu söyleyebiliriz. Soap (Simple Object Access Protocol) bilgisayar ağlarında web hizmetlerinin uygulanmasında yapılandırılmış bilgi alışverişi için kullanılan bir protokoldür. Cevap olarak XML sonuç döndürmektedir.

Resim

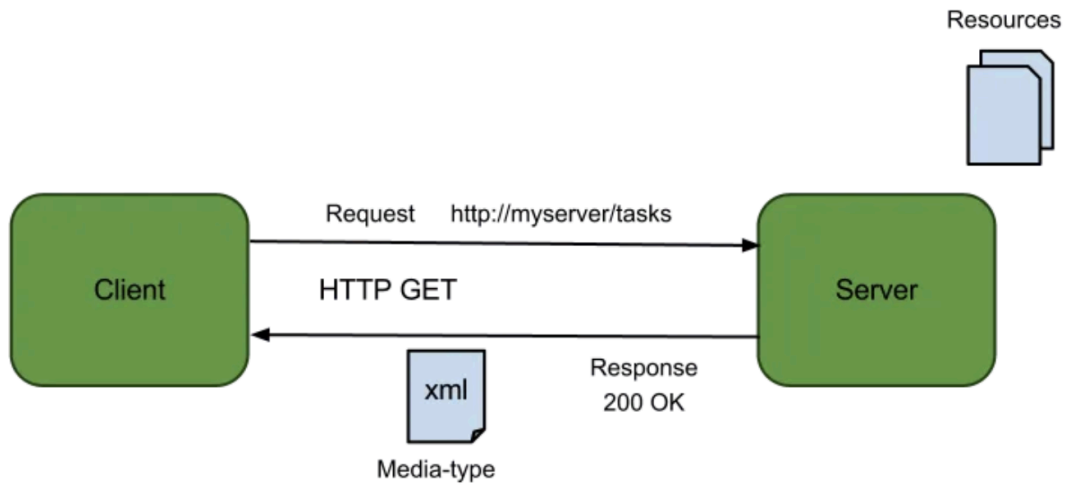
<https://medium.com/@ilkebasalak/web-servis-soap-ve-rest-nedir-8e7e03f28a9b> 'den alınmıştır.



## Restful Service

İstemci-sunucu arasında hızlı ve kolay şekilde iletişim kurulmasını sağlayan bir servis yapısıdır. REST API, istemci ve sunucu arasında XML, JSON, HTML, TEXT başta olmak üzere pek çok formatta veri taşıyarak uygulamaların haberleşmesini sağlar. REST standartlarına uygun yazılan web servislerine RESTful servisler denir.

Resim <https://medium.com/@kdrcandogan/web-servis-soap-verest-93930908a465> 'den alınmıştır.



## HTTP Methods

HTTP metotları, bir web servisi veya RESTful servisi üzerinden belirli bir kaynak üzerinde gerçekleştirilecek işlemleri belirtir.

GET: Belirli bir kaynağın okunması için kullanılır. Örneğin, bir RESTful servis üzerinden bir makalenin detaylarını almak için GET metodu kullanılabilir.

POST: Yeni bir kaynak oluşturmak veya mevcut bir kaynağı güncellemek için kullanılır. Örneğin, bir web servisi üzerinden yeni bir müşteri oluşturmak için POST metodu kullanılabilir.

PUT: Belirli bir kaynağı güncellemek için kullanılır. Örneğin, bir RESTful servis üzerinden bir makalenin içeriğini güncellemek için PUT metodu kullanılabilir.

DELETE: Belirli bir kaynağı silmek için kullanılır. Örneğin, bir RESTful servis üzerinden bir kullanıcıyı silmek için DELETE metodu kullanılabilir.

HEAD: GET metoduyla benzer işleve sahiptir ancak geri dönen yanıtta mesaj gövdesi bulunmaz (yani başlıklar ve içerikleri GET metoduyla aynıdır). Bu nedenle GET mesajı gönderilmeden önce bir kaynağın var olup olmadığını kontrol etmek için kullanılabilir.

CONNECT: Bir proxy sunucu üzerinden başka bir sunucuya bağlanmak ve proxy sunucuyu bir tünel gibi kullanmak için kullanılır.

OPTIONS: Bu metod belli bir kaynak için kullanılacak HTTP metodlarını sunucudan sorgulamak için kullanılır.

TRACE: Teşhis amaçlı kullanılan bir metoddur. Sunucu bu metodla gelen istek mesajının içeriğini aynen yanıt gövdesinde geri göndermelidir. Bu yöntemle sunucu ile istemci arasında bir vekil sunucu varsa bu sunucunun ve yaptığı değişikliklerin tespiti mümkün olabilir.

PATCH: Bu metod bir kaynağa istediğiniz küçük çaplı değişimi yapmanızı sağlar.

SEARCH: Bir dizinin altındaki kaynakları sorgulamak için kullanılır

Örnek olarak ki bir müşteri yönetimi uygulaması var. Müşteri bilgilerini RESTful bir servis üzerinden yönetiyoruz.

Müşteri listesini almak için: GET /customers

Yeni bir müşteri eklemek için: POST /customers

Belirli bir müşteriyi güncellemek için: PUT /customers/{customerID}

Belirli bir müşteriyi silmek için: DELETE /customers/{customerID}