T.C.

HALİÇ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENLİSLİĞİ ANABİLİM DALI

YÖNETİM BİLİŞİM SİSTEMLERİ PROGRAMI

WEB SERVİSLERİ İLE ELEKTRONİK TİCARET UYGULAMASI

YÜKSEK LİSANS TEZİ

Hazırlayan

HÜLYA ÇELİK

Tez Danışmanı

Prof.Dr.ALİ OKATAN

Ocak, 2008
İSTANBUL

T.C.

HALİÇ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENLİSLİĞİ ANABİLİM DALI

YÖNETİM BİLİŞİM SİSTEMLERİ PROGRAMI

WEB SERVİSLERİ İLE ELEKTRONİK TİCARET UYGULAMASI

YÜKSEK LİSANS TEZİ

Hazırlayan

HÜLYA ÇELİK

Tez Danışmanı

Prof.Dr.ALİ OKATAN

Ocak, 2008
İSTANBUL

İÇİNDEKİLER

ÖNSÖZ

Yüksek lisans öğrenimim sırasında ve tez çalışmam boyunca, desteğini ve yardımlarını esirgemeyen değerli danışman hocam Sayın Prof. Dr. Ali Okatan'a ve hep yanımda olan en büyük destekçim eşim Uygar'a en içten dileklerimle teşekkür ederim.

Ocak 2008
Hülya ÇELİK

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

KISALTMALAR

XML	eXtensible Markup Language
W3C	World Wide Web Consortium
WS-I	Web Services Interoperability Organization
OASIS	Organization for the Advancement of Structured Information Standards
LA	The Liberty Alliance
HTTP	Hyper Text Transfer Protocol
RPC	Remote Procedure Call
COM	Component Object Model
DCOM	Distrubuted Component Object Model
EDI	Electronic Data Interchange
SOAP	Simple Object Access Protocol
WSDL	Web Service Definition Language
UDDI	Universal Description, Discovery and Integration
GXA	Global eXtensible Markup Language Architecture
ebXML	e-business eXtensible Markup Language
BPEL4WS	Business Process Execution Language for Web Services
WSFL	Web Services Flow Language
CORBA	Common Object Request Broker Architecture
RMI	Remote Method Invocation
SAML	Security Assertion Markup Language
SMTP	Simple Mail Transfer Protocol
FTP	File Transfer Protocol
MQ	Message Queue
JSP	Java Server Pages
ASP	Active Server Pages
CGI	Common Gateway Interface
ISAPI	Internet Server Application Programming Interface
SOA	Service Oriented Architecture
B2B	Business To Business
B2C	Business To Customer
C2B	Customer To Business
C2C	Customer To Customer
B2E	Business To Employee

T.C.
HALİÇ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YÖNETİM BİLİŞİM SİSTEMLERİ PROGRAMI
YÜKSEK LİSANS TEZİ

WEB SERVİSLERİ İLE ELEKTRONİK TİCARET UYGULAMASI

Hülya ÇELİK

Tez Danışmanı : Prof.Dr. Ali OKATAN
Ocak 2008

ÖZET

Günümüzde iş yapılanmaları firmaların kendi işlerine odaklanmaları, işin gerektirdiği tamamlayıcı konular için ise diğer firma ve kurumlardan destek almaları şeklinde gerçekleşmektedir. Böylelikle firmalar tarafından sunulan hizmetin kalitesi artmaktadır.

İnternet üzerinden satış yapan bir firmanın, ürün tedariği için tedarikçilerle, lojistik hizmetleri için kargo firmalarıyla, döviz kuru bilgileri için finansal veri sağlayan kurumlarla, ödemeler için ise bankalar ile iş süreçlerini birleştirmesi gerekir. Bu hizmetleri ilgili firma veya kurumların internet siteleri üzerinden tek tek yapmak mümkün olduğu halde, elektronik ticaret uygulamasındaki işbirliği için bu bir çözüm olmamaktadır. Web servisleri bu noktada devreye girmekte ve bilginin paylaşımı için görsel ve uygulamalı web siteleri yerine XML (eXtensible Markup Language) tabanlı platformdan bağımsız yeni bir model sunmaktadır.

Bu çalışmada öncelikli olarak web servisleri mimarisi ve standartları ele alınmış, protokolleri hakkında karşılaştırılmalı bilgi verilmiştir. Ayrıca web servislerinin sunmuş olduğu avantajlardan bahsedilmiş ve web servislerine dayalı çözümlerden örnekler verilmiştir.

Çalışmanın devamında, elektronik iş ve elektronik ticaret kavramları üzerinde durulmuş ve örnek bir B2B (İşten İşe) web servisi uygulaması gerçeklenmiştir. Uygulamada konu olarak, interaktif bir sigorta broker şirketi ele alınmış, müşterilerine çeşitli sigorta şirketlerinin ürünlerini sunduğu ve satın almalarını sağladığı düşünülmüştür. Broker şirketinin, çeşitli tercih ve özelliklere göre sigorta şirketlerinin ürünlerini araştırması, incelemesi ve müşterilere sunması işi oldukça zaman kaybettirici ve maliyetli bir iştir. Bunun çözümü olarak sigorta şirketlerinin ve brokerin birlikte çalışabilirliliğini sağlayan ve müşterilere uygun ürünleri sunabilen, web servislerini barındıran örnek bir uygulama geliştirilmiştir.

En son bölümde yapılan çalışma ile ilgili bir değerlendirme bulunmaktadır.

Anahtar Kelimeler : Web servisleri, XML, elektronik iş, elektronik ticaret, B2B

T.C.
HALİÇ UNIVERSITY
INSTITUTE OF SCIENCE
COMPUTER ENGINEERING
MANAGEMENT OF INFORMATION SYSTEMS
MASTER THESIS

APPLICATION OF ELECTRONIC COMMERCE WITH WEB SERVICES

Hülya ÇELİK

Supervisor: Prof.Dr. Ali OKATAN
January 2008

SUMMARY

Today business establishment is builded in focusing on their own occupation by firms, while these establishments are also builded by getting support from other firms and organizations for auxiliry matters that required for business. Hence, the quality of service offered by firms increases.

A firm which makes sales via internet, should integrate business processes with suppliers for producting supplying, with cargo firms for logistic services, with corporations that offer financial data for exchange rate, and with banks for deposits. Although it is possible to accomplish these services one by one via web sites of each related firm/corporation, this is not a plausible solution for collaboration on e-commerce execution. At this point, web services come to scene and offer a new model which is XML (eXtensible Markup Language) based and platform-independent instead of visual and applied web sites.

First of all, in this study, Web Services architecure and standarts were examined, and reciprocal information were given about their protocols. Besides, offered advantages of web services were mentioned and some examples of solutions which related to web services were given.

At the following part, generally electronic business and electronic commerce definitions were emphasized and a B2B (Business to Business) web service application was implemented. In this application, an interactive insurance broker company was discussed, and it was assumed that this company provided products of various insurance companies and also ensured selling to its costumers as a subject. It is too costly and time consuming for broker companies to search products of insurance companies, to examine and to offer these products to their costumers according to various preferences and features. As a solution to this issue, a sample application has been developed including web services that provides interoperability between insurance companies and this broker, and offers appropriate products.

Lastly, the evaluation of the study is given.

Keywords : Web services, XML, e-business, e-commerce, B2B

1. GİRİŞ

Bilgi her çağda farklı kanallar üzerinden paylaşılmış ve dağıtılmıştır. Basitliği ve yaygın kullanımı sayesinde internet günümüzün en yaygın paylaşım kanalı olmuştur.

İlk başlarda yerel ağlardaki belge paylaşımı tabanlı bilgi alışverişi yerini zamanla web sitelerine ve web uygulamalarına bırakmıştır. Artan web siteleri ve uygulamaları ise iş yapış biçimlerimizde köklü değişikliklere yol açmış, bazı işletmeler için iş süreçlerini web üzerinden yaptırma gereksinimini doğurmuştur. Bu şekilde internet üzerinden artan hizmet çeşitleri işletmelerin diğer işletmelerle olan iş süreçlerini bütünleştirme gereksinimini ortaya çıkarmış, bunun sonucunda da web servisi kavramı hayatımıza girmiştir.

Web servisleri, bilginin paylaşımı için görsel ve uygulamalı web siteleri yerine XML tabanlı, platformdan bağımsız yeni bir model sunmaktadır. Örneğin, sürekli değişen döviz kurları nedeniyle güncel kura ihtiyacınız olduğunda bunu almak için web servisleri öncesinde iki yol vardı. Birincisi, merkez bankası web sayfasından kurları alıp, kendi sisteminize girmektir. Böylece istenildiği zaman kur bilgilerine ulaşılabilinir, ama sorun, bu web sayfasının değişimlerinden sürekli haberdar olmaktır. Sayfada oluşacak ufak bir format kayması, çok büyük kur değerleri ile iş yapmanıza yolaçacaktır. İkinci yol ise merkez bankası ile bir iletişim formatı geliştirmek ve kur bilgilerini belirlenen standartlarla elde etmektir. Bu yöntemin sakıncaları ise merkez bankası açısından her yeni hizmet almak isteyen firma için yeni standartlar belirlemek zorunda kalmak ve servis alan kuruluş açısından da bu standartlarda yapılan değişikliklerden merkez bankasını bilgilendirmek olacaktır. Web servisleri bu sorunları ortadan kaldırıp bilgi paylaşımına olanak sağlamaktadır.

Bu çalışmada önce web servisleri hakkında genel bilgiler verilerek mevcut yazılım mimarileri gözden geçirilmiş, daha sonra ayrıntılı olarak web servis modeli, web servis yığını ve standartları karşılaştırılmalı olarak anlatılmıştır.

İkinci bölümde web servislerinin en yoğun uygulama alanları olan, elektronik iş, elektronik ticaret ve elektronik devlet kavramları üzerinde durulmuş ve güncel örnekler verilmiştir.

Üçüncü bölümde, XML ve web servislerinin bir çok alan ve sektörde değişikliklere yol açmasıyla beraber, mevcut kurumların da yapılarının servis odaklı hale getirmelerini amaçlayan servis yönelimli mimariden (SOA) bahsedilmektedir.

Çalışmanın dördüncü bölümünde ise, araç sigortalama konusu ele alınarak, bir sigorta broker firmasının sigorta şirketleriyle olan elektronik ticaret ilişkisi üzerinde durulmuş ve web servislerini barındıran bir örnek uygulama geliştirilmiştir.

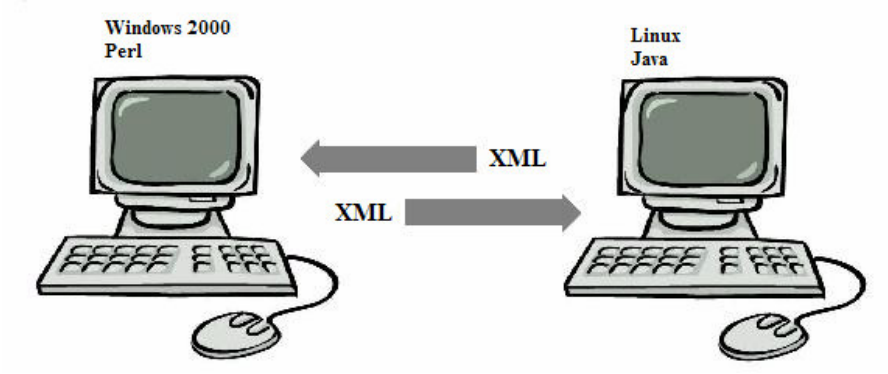
2. WEB SERVİSLERİ HAKKINDA

2.1. Web Servisi Nedir?

Web servisleri, birlikte-çalışabilir, dağıtık uygulamalar sunan, herhangi bir dilde ve platformdan bağımsız web erişimli uygulamaları kapsayan bir sistem entegrasyon yöntemidir. İnternet Komitesi W3C (World Wide Web Consortium) tarafından yapılan resmi tanımıyla web servisi, bilgisayarlar arasında ağ üzerinden etkileşimi ve uyumluluğu sağlayacak yazılım sistemidir. Günümüzde birbiriyle haberleşecek sistemleri gerçeklemek için en çok tercih edilen yöntem web servisidir.

Web servisleri, Genişletilebilir Etiketleme Dili (XML: eXtensible Markup Language) tabanlı mesajlaşmayı esas alır. Bu nedenle, eski entegrasyon sistemlerinin aksine, haberleşecek sistemlerin birbirlerinin gerçeklenmelerinden haberdar olması veya platformlarının uyumlu olması gerekmez. Şekil 2.1 de gösterildiği gibi Java dili ile geliştirilmiş ve Linux sistem üzerinde çalışan bir uygulama ile Perl dili ile geliştirilmiş ve Windows işletim sistemi üzerinde çalışan bir uygulama, birbirlerinin çalışma ortamlarından bağımsız olarak, XML iletişim standartları aracılığıyla iletişim kurabilir.

Şekil 2.1 Basit bir web servisi örneği



Web servisi adından da anlaşılacağı gibi web üzerinden servis veren program parçacıklarıdır. Web üzerinden verilen bu servisler standart Transfer Protokolü (HTTP : Hyper Text Transfer Protocol) HTTP protokolü ile olmaktadır. Bu da herkesin bu protokol vasıtası ile bir web servisine ulaşabileceğini göstermektedir.

Bir kullanıcının HTTP üzerinden bir web servisini kullanmasına Uzak İşlev Çağırma (RPC : Remote Procedure Call) denmektedir. Aslında RPC, aynı ağ içindeki farklı bir makinadaki bir fonksiyona ulaşabilme fikriyle ortaya çıkmış bir teknolojidir. HTTP üzerinden yapılan bu çağrımlara karşı Nesne Erişim Protokolü (SOAP : Simple Object Access Protocol), dediğimiz protokol XML çıktıları üretir. Bu sayede standart bir veri paylaşım aracı olan XML ile istenilen veriler alıp kullanılabilinir. Aslında kabaca düşündüğümüzde web servisleri dağıtık yapıdaki sistemlerde çokça kullanılan DCOM teknolojisinin web üzerinden binary olmayan düzeydeki uygulamalarıdır.

2.2. Yazılım Uygulamalarının Mevcut Mimarileri

Web servislerini daha iyi anlayabilmek için internet uygulamalarının mevcut mimarisini gözden geçirmek gerekmektedir:

2.2.1. Monolitik Mimari (Tek Parça Uygulamalar)

Monolotik uygulamalar,.sadece tek bir markaya özgü kullanıcı arabirimi altında çeşitli değişik servisi bir araya getirirler. Tek bir parçadan ibaret ve çok büyük bir program olarak tasarlanırlar ve programın bir bölümünü geliştirebilmek için bütün uygulamayı tekrar düzenlemek, hatta tekrar derlemek gerekir. Çoğu zaman belli bir platforma veya teknolojilere bağlıdırlar ve kolayca genişletilemezler. Uygulamaların çalışabilmesi için server teknolojisi olan mainframe sunucular kullanılır. Birden fazla uygulamanın bir arada çalışmasını sağlamak ise başlı başına bir proje demektir. En önemlisi uygulama mantığının parçaları bir uygulamadan bir diğerine aktarılıp tekrar kullanılamaz. Örneğin bir web sitesi ile bir masaüstü uygulaması arasında bir bütünleştirme için monolitik mimaride basit bir çözüm bulunmamaktadır.

2.2.2. İstemci/Sunucu Mimarisi

Monolitik mimarideki problemlere karşı çeşitli çözümler geliştirilmiştir. Bu çözümlerden ilki uygulamaları server (sunucu) ve client (istemci) tarafına bölmektir. Client/Server Model adı verilen bu çözümde arayüzün ve bazı işlemlerin istemcide, veri erişimi ve diğer işlemlerin sunucuda bulunacak şekilde paylaştırılması sonucu monolitik uygulamaların karmaşıklığına kısmen çözüm bulunabilmesine karşın bileşenlerin tekrar kullanılamaması sorununa bu model

bir çözüm olamamıştır. Bunun da çözümü olarak Bileşen Modeli (COM : Component Object Model) geliştirilmiştir.

2.2.3. Bileşen - COM Teknolojisi ve Dağıtık Uygulamalar

COM teknolojisi sayesinde geliştiriciler istedikleri dilde bir bileşen geliştirebilir ve çeşitli platformlarda kullanılabilirler. Markaya özgü dosya formatları kullanılarak bilgi alışverişine daha az ihtiyaç duyulmaktadır. COM bileşenleri, geliştiricilerin program işlevlerini iyi tanımlanmış arabirimler sayesinde öz ve tekrar kullanılabilir parçalar halinde paketlenmesine olanak verir. Nesne tabanlı bir programlama modeli olan COM uygulamaların birlikte kullanılabilirliğini sağlamıştır. COM uygulamaları dağıtık uygulamalarda kullanılması için evrim geçirerek DCOM (Distrubuted COM : Dağıtık COM) adını almış ve dağıtık uygulamalar yazmaya elverişli hale gelmiştir.

2.2.4. Servis Tabanlı Mimari

İlk 1975 senesinde ağ üzerinde veri değişimini sağlayan ve zamanla bir standart haline gelen Elektronik Veri Değişimi (EDI : Electronic Data Interchange), farklı kuruluşlar arasındaki yapısal veri değişimine öncülük ederek servis tabanlı teknolojinin ilk adımı oldu. Daha sonra 1998 senesinde basit bir işaretleme dili oluşturmak amacıyla W3C tarafından tasarlanan XML dilinin bir standart olarak kabul edilmesiyle beraber Sun ve IBM, farklı ortamların birbiriyle zahmetsizce haberleşmesi konusunda ortak çalışmalar yürüttü ve bu çalışmalar web servislerinin doğuşuna öncülük etti.

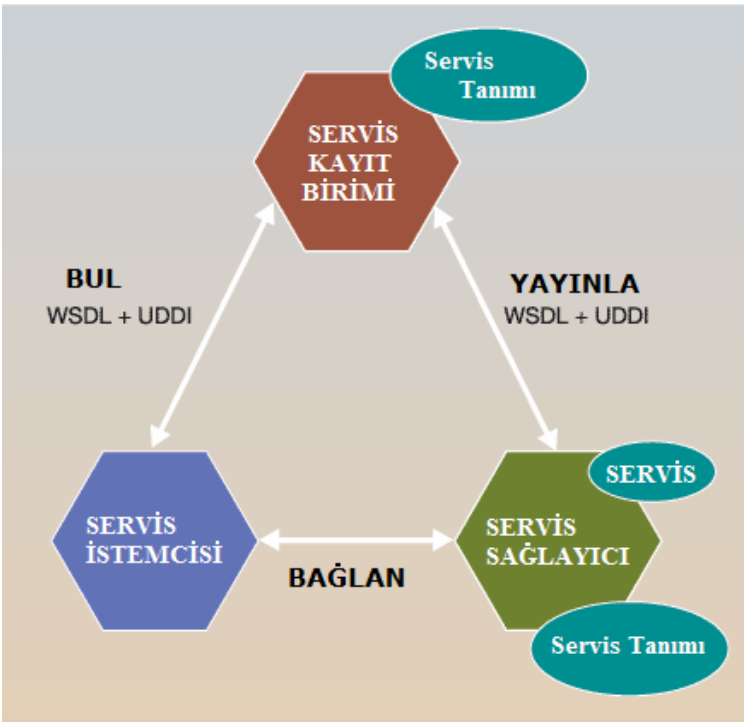
Web servisleri için COM teknolojisinin masaüstü uygulamalarda yaptığı evrimin web de gerçekleştirdiğini söyleyebiliriz. COM'dan farklı olarak web servisleri açık standartlar temeli üzerine inşa edilmiştir.

Farklı bilgisayarların, farklı ağların ve uygulamaların olduğunu düşünürsek aralarındaki bilgi alışverişini sağlamada web servislerinin önemi bir kez daha artmaktadır. Web ortamında geliştirilebilen, çağrılarak erişilebilen modüler uygulanabilen web servisleri özellikle kurumsal iş süreçlerinin geliştirip yaygınlaştırılmasına öncülük etmiştir. Dağıtık nesne teknolojileri ve internetin gelişimi bilgi-temelli internetten servis-temelli internet uygulamalarına geçişi sağlamıştır.

2.3. Web Servis Modeli

Web servisleri mimarisi, servis sağlayıcı, servis kayıt birimi ve servis istemcisi rolleri arasındaki etkileşimler üzerine kuruludur. Etkileşimler yayınla, bul ve bağlan işlemlerini içerir. Bu roller ve işlemler birlikte Şekil 2.2. de gösterildiği gibi web servisleri bileşenleri (web servisinin yazılım modülü ve tanımı) üzerinde çalışırlar.

Şekil 2.2. Web servis modeli, rolleri ve işlemleri (Kaynak IBM)



Tipik bir senaryoda, bir servis sağlayıcı ağ üzerinden erişilebilir yazılım modülüne yani bir web servisinin gerçekleştirimine ev sahipliği yapar. Servis sağlayıcı, web servisi için bir servis tanımı yapar (WSDL ile) ve bunu bir servis istemcide yada bir servis kayıt biriminde (UDDI de) yayınlar. Servis istemcisi, servis tanımını yerel olarak yada bir servis kayıt biriminden almak için bul işlemini kullanır ve servis sağlayıcıya bağlanmak ve web servisi gerçekleştirimini kullanmak (SOAP ile) için servis tanımını kullanır. Veri ve mesajlar HTTP üzerinde XML olarak aktarılır. Web servisi, yeni gereksinimlere karşı değiştirilebilir ya da elde edilebilirliği ortadan kalktığında yayınlanmasına son verilebilir.

Web servis modelinde rollerin tanımı şöyledir:

- Servis sağlayıcı (Service provider) : İstemcilerin sağlayıcıda bulunan servislere erişimini sağlar. Servis sağlayıcı kendi sitesinde bulunan web servisleri tanımını servis kayıt birimine (service registery) kaydederek bu servisinin nasıl çağrılacağı belirtir.
- Servis istemcisi (Service requestor) : Servis sağlayıcısında bulunan web servislerini çağırarak kullanan istemci uygulamalardır. Web servisinin nasıl çağrılacağı ve ilgili parametreleri servis kayıt biriminden arayarak bulur ve çağırır.
- Servis kayıt birimi (Service registery) : 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.

Bir uygulamanın web servislerinden yararlanabilmesi için şu üç işlemin olması gerekir:

- servis tanımlarının yayınlanması,
- servis tanımlarının aranması ve
- servis tanımına göre servislerin çağrılması (bağlanılması).

Bu işlemler tek başlarına yada birbirlerini takip eden sırada olabilirler. Detayları ise şöyledir:

- Yayınlama: Bir servise erişilebilmesi için servis tanımının yayınlanması gerekir. Böylece, servis istemci o servisi bulabilir. Uygulamanın gereksinimlerine göre yayınlanma yeri değişebilir (servis kayıt birimi, servis istemci v.b.).
- Bulma: Bu işlemde, servis istemci direk olarak yada ihtiyaç duyulan servisin tipine göre servis kayıt birimini sorgulayarak bir servis tanımı alır. Bulma işlemi servis istemcinin iki farklı yaşam döngüsünde gerçeklesebilir: Program geliştirme için servisin arayüz tanımının tasarım aşamasında alınması, servisin çağrılması için servisin bağlanma ve yer tanımının çalıştırma aşamasında alınması.
- Bağlanma: Sonuçta, bir servisin kullanılması için çağrılması gerekir. Bağlanma işleminde, servis istemci, servisin yerini bulmak, bağlantı kurmak ve çağırmak için servis tanımındaki bağlanma bilgilerini kullanarak çalıştırma aşamasında servis ile bir etkileşim başlatır.

2.4. Web Servisi Yığını

Yayınlama, bulma ve çalıştırma işlemlerini birlikte çalışabilir bir tarzda gerçekleştirebilmek için, standartları her bir katmanda birleştiren bir web servisleri yığını olmalıdır. Tablo 2.1. de kavramsal web servisleri yığını gösterilmektedir. Üstteki her bir katman, alttaki katmanlar tarafından sağlanan kabiliyetler üzerine kurulmuştur. Dikey katmanlar ise yığının her katmanında sağlanması gereken gereksinimleri gösterir.

Tablo 2.1. Web servisi yığını (Kaynak: IBM)

	Yatay Katmanlar	**Dikey Katmanlar**		
WSFL	Servis Akışı	Güvenlik	Yönetim	Servis Kalitesi
Statik.....UDDI	Servis Keşfi			
Dinamik.....UDDI	Servis Yayınlaması			
WSDL	Servis Tanımı -servis gerçekleştirimi -servis arayüzü Güvenli mesajlaşma			
SOAP	XML Tabanlı Mesajlaşma			
HTTP, SMTP, FTP, MQ	Ağ Katmanı			

Bir Web servisi için kaçınılmaz olan özellikler; ağ ile erişebilirlik (HTTP protokolü) ve bu ağ üzerinden kullanılabilirlik (SOAP) ve son olarak servisin tanımlanabilirliği (WSDL) olduğundan, bu yığındaki ilk üç katman herhangi bir web servisini kullanmak için koşulsuz olarak gereklidir. En basit yığın; ağ katmanı için HTTP, XML mesajlaşma katmanı için SOAP ve servis tanımla katmanı için WSDL`den oluşur. Bu üç katmanlı yığın kurumlar arası veya kamusal tüm web servislerinin desteklemesi gereken birlikte işleyebilir temel yığındır. Özellikle kurum içi ya da özel web servisleri diğer ağ protokollerini ve dağıtık programlama teknolojilerini destekleyebilir.

Yığının alttaki üç katmanı, uyum ve birlikte işleyebilirlik teknolojilerini tanımlarken, sonraki iki katman (servis yayınlama ve servis keşfi) çeşitli çözümler ile gerçekleştirilebilir.

Web servisi yığınının katmanları şöyledir:

- Ağ Katmanı : Web servisi yığınının esas dayanak noktası bilgisayar ağıdır. Web servislerinin bir istemci tarafından çağrılabilmeleri için ağ üzerinden erişilebiliyor olmaları gerekir. İnternet üzerine kamuya açılmış web servisleri ortak olarak kullanılan ağ protokollerini kullanırlar. HTTP protokolü, tüm ortamlardaki kullanılabilirliğinden dolayı, internet üzerinde kullanılabilen web servisleri için fiili bir standarttır. SMTP ve FTP dahil diğer internet protokolleri de desteklenebilir.
- XML tabanlı mesajlaşma katmanı mesajlaşma protokolünün temel şartı olarak XML`in kullanımını gösterir. Bir çok sebepten dolayı XML mesajlaşma protokolü olarak SOAP seçilmiştir. Bu sebeplerin bazılarını şu şekilde sıralayabiliriz:

  - XML kullanarak uzaktan metot çağırma işlemi ve doküman merkezli mesajlaşma için standartlaştırılmış bir kaplama mekanizmasıdır.

  - Basittir; yük olarak sadece bir XML zarfı ile gerçekleştirilen bir HTTP POST metodundan ibarettir.

  - HTTP POST üzerinde tercih edilir çünkü iletilen mesaja, SOAP başlıklarını ve işlemin ve fonksiyonun standart kodlamasını ekleyebilir.

  - SOAP mesajları, web servisleri mimarisindeki, yayınlama, bulma ve bağlanma işlemlerini destekler.
- Servis tanımımlaması katmanı, tanım dokümanları katmanıdır. İlk olarak, WSDL XML tabanlı servis tanımı için fiili bir standarttır. Yazılım bütünleştirmede kullanılan Web servislerini desteklemek için gerekli olan minimum servis tanımıdır. WSDL, ara yüzü ve servis etkileşimindeki mekaniği tanımlar. İş içeriğini, servis kalitesini ve servisten servise ilişkileri belirtmek için ilave tanımlar gereklidir. WSDL dokümanı, web servislerine ait bu daha ileri seviyedeki varlıkların tanımlanması için diğer tanımlayıcı dokümanlarla tamamlanabilir. Örnek olarak iş içerigi, WSDL dokümanına ek olarak UDDI veri yapılarının kullanımıyla tanımlanır. Servise ait akış ve tümleşimi bir Web Servisi Akış Dili (Web Services Flow Language : WSFL) dokümanı ile tanımlanır.
- Servis yayınlaması ve Servis keşfi: Bir WSDL dokümanının bir servis istemcisi tarafından bulunabilmesini sağlayan, servis istemcisinin yaşam döngüsünün herhangi bir seviyesindeki herhangi bir faaliyet servis yayınlamasını niteler. Bu katmandaki en basit ve en statik örnek, servis sağlayıcısının, WSDL dokümanını direkt olarak servis istemcisine göndermesidir. Buna direkt yayınlama denir. E-posta direkt yayınlamada kullanılabilecek

araçlardan biridir. Direkt yayınlama statik olarak sınırlandırılmış uygulamalar için kullanışlı bir yöntemdir. Alternatif olarak, servis sağlayıcısı servisi tanımlayan WSDL dokümanını yerel bir WSDL kayıtçısında, özel bir UDDI kayıtçısında veya bir UDDI operatör düğümünde yayınlayabilir.

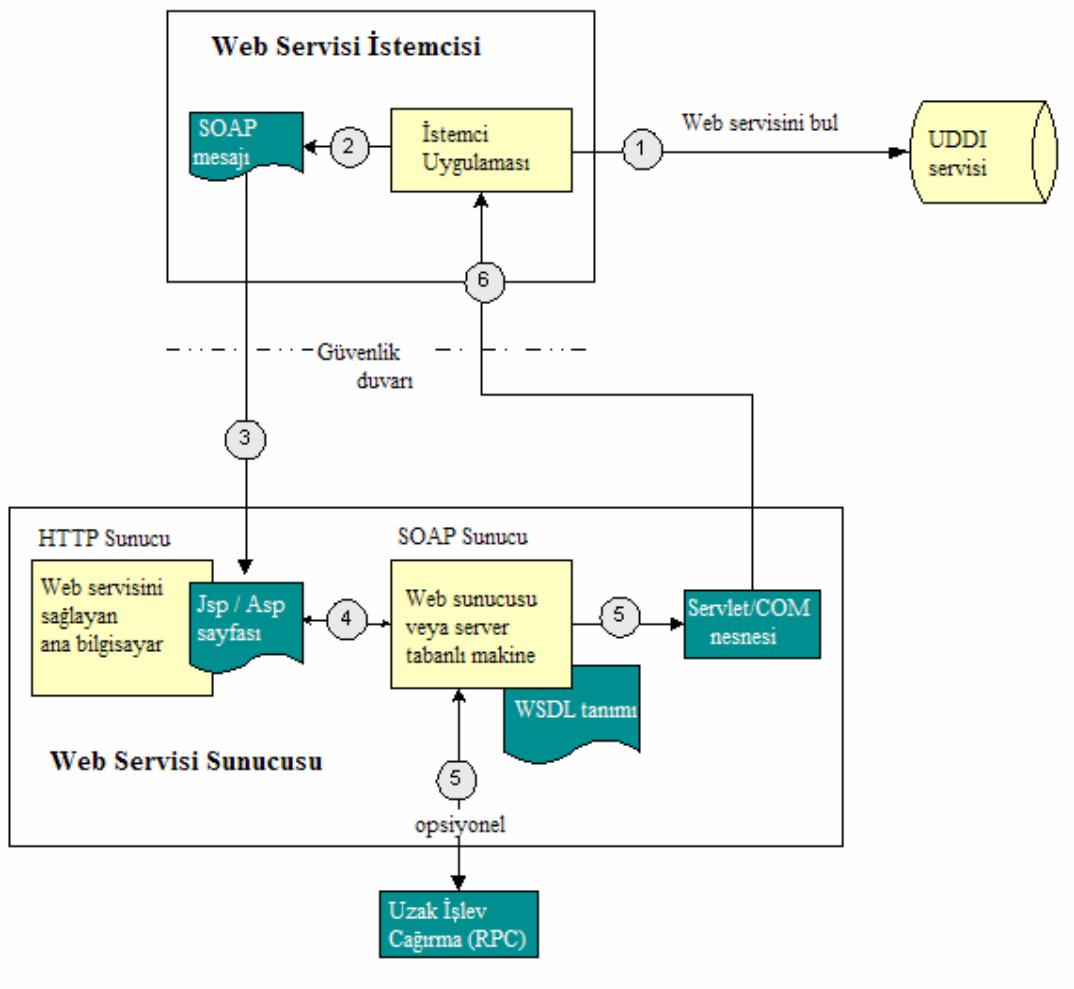
- Servis akışı : Bir web servisi gerçekleştirimi bir yazılım modülü olduğu için, bir web servisini web servislerini birleştirerek üretmek normal ve mümkündür. Bir web servisi bileşimi çeşitli amaçlar için çeşitli roller oynayabilir. Kurum içi web servisleri, kamuya açık tek bir web servisi ara yüzünü oluşturmak için işbirliği yapabilirler, ya da farklı kurumlara ait web servisleri, makineden makineye, şirketten şirkete hareketleri (transaction) gerçekleştirmek için iş birliği yapabilirler. En üstteki servis akışı katmanı, servisten servise iletişimlerin, işbirliklerinin ve akışların nasıl gerçekleştirildiklerini tanımlar. WSFL bu etkileşimleri tanımlamak için kullanılır. Kredi kartı süreçlerinden üretim şemasına kadar parça parça işleri kapsayan uygulama servisleri bu katmanda yer alır.

2.5. Web Servisi İle İletişim

Web servisi istemcisinin bir servis sağlayıcıdan bir servisi çağırması aşamasındaki temel adımları Şekil 2.3. de gösterilmiş olup şöyledir :

1. Web Servisi İstemcisi (SOAP Client) servis kayıt biriminden (UDDI) Web servisini bulur.
2. İstemci bir SOAP mesajı hazırlar. Bu SOAP mesajı bir XML belgesidir.
3. İstemci SOAP mesajını Web sunucu veya uygulama sunucusunda çalışan SOAP istek dinleyicisine gönderir. İstek dinleyiciler gelen isteklere cevap veren sunucu programlardır. Bir JSP, ASP, CGI veya ISAPI programı olabilir.
4. SOAP sunucu, gelen SOAP mesajını ayrıştırır ve gerekli parametreleri göndererek istenen nesnenin istenen yöntemini çağırır.
5. Çağırılan 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 istekde bulunan programa gönderir.

Şekil 2.3. Web servisi istemcisinin ve sağlayıcı arasındaki temel adımlar



2.6. Web Servisi Standartları

2.6.1. XML (eXtensible Markup Language)

XML veri standardı, platformdan bağımsız oluşu ve entegrasyon ve standardizasyon kolaylığı ile web servisleri teknolojisinin temel yapı taşlarından biridir. XML, kişilerin kendi sistemlerini oluşturabilecekleri, kendi etiketlerini tanımlayarak çok daha rahat ve etkin programlama yapabilecekleri ve bu belirlenen etiketleri kendi yapıları içerisinde standardize edebilecekleri esnek, genişleyebilir ve kolay uygulanabilir bir meta dildir.

XML verilerin transferi, depolanması, sorgulanması ve yönetiminde; veriye içerik değeri katması, ihtiyaç duyduğumuz sistemi yaratabilme esnekliği sunması, dağınık verilerin

kümelenmesi, karşılaştırma yapma kolaylığı, farklı veri formatlarını ve dilleri destekleyebiliyor olması ve tüm sistemlerle çalışabilme özelliğiyle bugün ve gelecekte ihtiyaç duyduğumuz veri standardıdır. Veri öbekleri XML'le bilgiye dönüşür.

XML kullanımının yaygınlaşmasının nedenlerini şöyle sıralayabiliriz:

1. Bilgiye içerik değeri katar. Etiketler ve diğer XML bileşenleri; veriyi yorumlamada, sorgulamalarda, akıllı veri işlemede (data mining) ve buna benzer diğer operasyonlarda kullanıcıya içerik bilgisi sağlar.

2. Dağıtılmış veriler için tek bir sunucudan görünüş sağlayan XML'in, erişim sağladığı bir çok verinin öğeleri, değişik veri tabanları içerisinde bulunabilir. XML ile bu verilere tek bir sunucu üzerinden bakılıyormuş gibi erişilebilir.

3. Verilerin sınıflandırması, her uygulamanın özelliklerine göre bire bir belirlenebildiğinden çeşitli uygulamalara yerinde ve etkin çözümler sağlayabilir. Özellikle verilerin iç içe geçirilebilir olmasıyla, klasik ilişkisel veri tabanlarındaki gibi tablolar arası ilişkilendirme işleminden tasarruf edildiğinden sorgulama ve operasyonlarda yüksek performans artışı ve kolaylık sağlar.

4. XML, sabit bir etiketler kümesi içermediğinden ve istenildiği kadar uygulamaya özel yeni etiket yaratılabildiğinden, genişleyebilen ve esnek bir veri standardıdır. İlişkisel veri tabanlarının dizayn güncellemeleri, XML'e göre daha çok zaman alır ve genelde sistemin performansını önemli ölçüde düşürür.

5. XML etiketleri doğal dille yazıldığından anlaşılması kolaydır. Böylece her düzeydeki çalışan veri etiketlerini kolayca okuyarak verinin içeriği hakkında bilgi sahibi olabilir.

6. İçeriği gösterimden ayırır. XSL style sheetleri tarafından oluşturulan görünüş ve veri yapısı bilgileri, XML ile hazırlanmış bir belgenin görünüşünün içeriğe dokunmadan değiştirilebilmesini sağlar.

7. Sektör içi ortak standartların geliştirilebilmesine ortam sağladığından, aynı sektördeki firmaların veri paylaşımını kolaylaştırır.

8. Sıradan veri tabanlarında; veri kayıtları, belirli şemalara ihtiyaç duyar; oysa XML belgeleri bu tür tanımlamalara ihtiyaç duymadan saklanabilir çünkü XML etiket ve özelliklerden oluşan meta veriler içerir.

9. Çeşitli veri türleriyle kullanılabilir. XML belgeleri, çoklu ortam verilerinden (resim, ses, video) aktif bileşenlere (Java Appletleri, ActiveX) kadar birçok olası veri türünü içerebilir.

10. Çok dilli belgeleri ve unicode'u destekleyen XML, uygulamaların uluslararası hale getirilmesinde önemli avantaj sağlar.

2.6.2. SOAP, Basit Nesne Erişim Protokolü (Simple Object Access Protocol)

Dağıtık uygulamalarda ve web servislerinin haberleşmesinde kullanılmak üzere tasarlanan, Uzak İşlev Çağırma (RPC : Remote Procedure Call) modelini kullanan, istemci/sunucu mantığına dayalı bir protokoldür. Dağıtık uygulamalarda ve uzaktan mesajlama (Remote messaging) için yeni bir teknoloji olmasına karşılık, alt yapısı sayesinde web üzerinde kullanmak için en uygun protokol olması nedeniyle web servislerinin adı SOAP la birlikte anılır olmuştur.

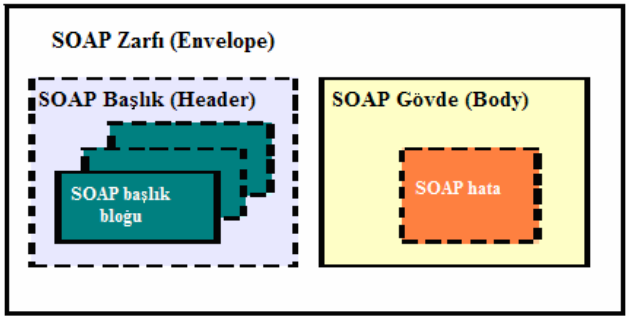
SOAP, servisler arasında yapılandırılmış bilgi alışverişini sağlayan XML-tabanlı bir iskelet sağlar. Bu bilgi SOAP Envelope (Zarf) içinde, Header (Başlık) ve Body (Gövde) şeklinde biçimlenir ve teorik olarak birçok iletişim protokolü üzerinden taşınabilir. Fakat sadece HTTP-bağı resmi olarak tanımlanmıştır ve bugün kullanımdadır. SOAP, RPC stilinde etkileşim sağlar, ve uzak fonksiyon çağrıları ve doküman-stili iletişim gibi, ve mesaj içeriği sadece WSDL'deki XML şema tanımına göre hazırlanır. Çağrı sonuçları isteğe bağlı olarak cevap mesajı ile birlikte döndürülür veya hata mesajı döndürülebilir. Aslında geleneksel programlama dillerindekine benzerdir.

Protokol dört parçadan oluşmaktadır :

- Mesaj içeriğini ve nasıl işleneceğini açıklayan bir zarf
- Uygulamalar tarafından tanımlanabilen veri türleri için kodlama kuralları
- Uzaktaki prosedürlerin tepkilerini temsil edebilmek için bir konvansiyon
- Mesaj değiş-tokuşu için bir bağlanma konvansiyonu

Şekil 2.4. de bir SOAP mesajının temel yapısı resmedilmiş olup Şekil 2.5 de SOAP mesajına bır ornek verilmiştir.

Şekil 2.4. SOAP mesajının temel yapısı (Kaynak : IBM)



Şeki 2.5. Örnek bir SOAP Mesajı

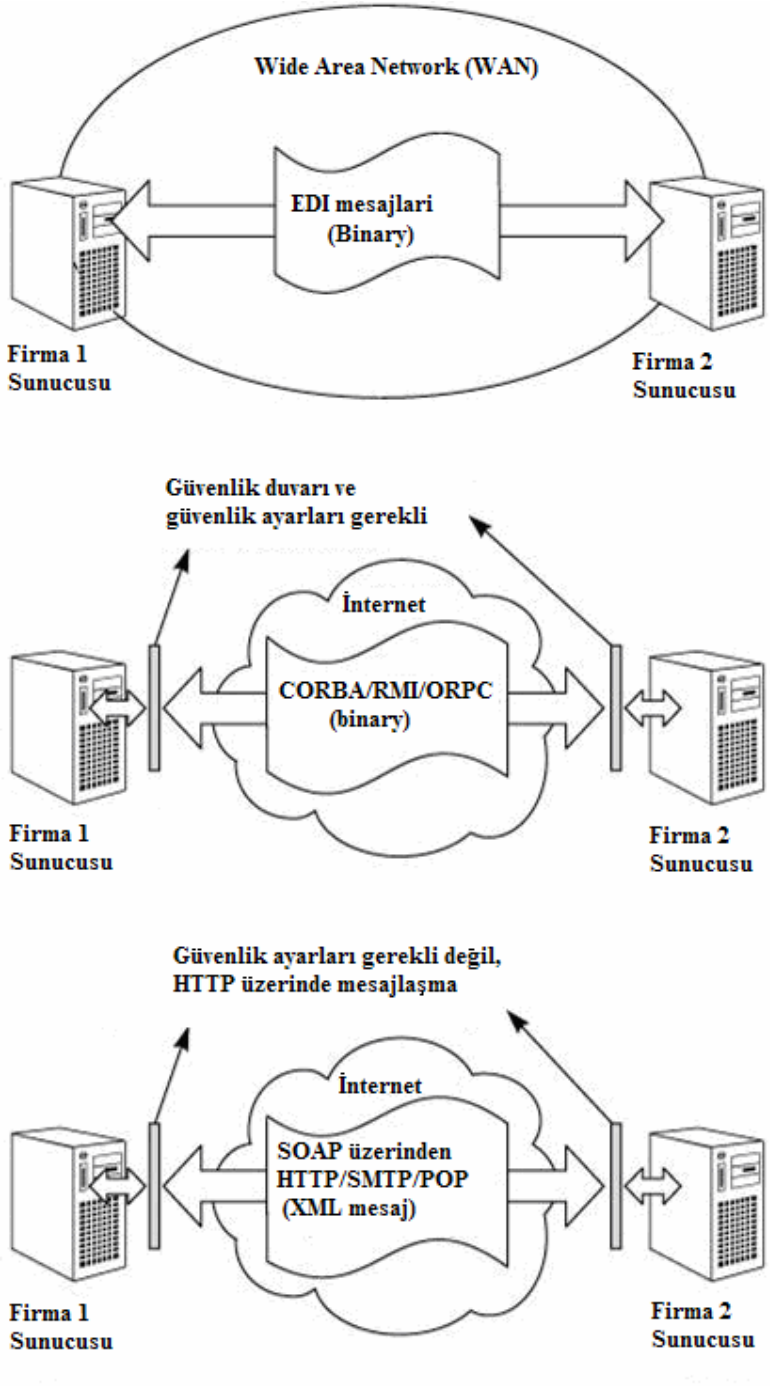
```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=”http://schemas.xmlsoap.org/soap/envelope/”>
   <SOAP-ENV:Header>
   ….
   ….
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
   ….
   ….
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.6.2.1. SOAP ve Diğer İletişim Teknolojileri

Daha önce CORBA (Common Object Request Broker Architecture), Java RMI (Remote Method Invocation) ve DCOM (Distributed COM) gibi çeşitli yapıların çözmeye çalıştığı, dağıtık ve farklı platformlarda bulunabilen yazılım objelerinin bir arada çalıştırılması problemi için tasarlanan SOAP'un en büyük avantajı herhangi bir platform ve sistemden bağımsız, tamamen yansız çalışmasıdır.

Önceki çözümler bağlantının iki ucunda uyumlu sistemler bulunmasını şart koşmaktadır. Örneğin RMI iki uçta Java, DCOM iki uçta Windows ve CORBA iki uçta aynı ORB uygulamasının bulunmasını gerektirmektedir. Aynı şekilde EDI ile iletişimde, kurumlar arasında özel bir ağ kurulu olmalı ve mesajlaşma formatının mutlaka belirlenmiş olması gereklidir. SOAP ise bu problemi İstem/Yanıt mesajlaşmalarını birer birer zarflayarak çözmüş bulunmaktadır. Şekil 2.6. de SOAP ve en bilinen ve yaygın olarak kullanılan diğer iletişim teknolojilerinin karşılaştırılması gösterilmektedir.

Şekil 2.6. SOAP ve diğer iletişim teknolojilerinin karşılaştırılması



SOAP'u benzeri protokollerden ayıran en belirgin ve üstün özelliği yapısının (daha doğru bir ifade ile mesaj formatının) XML üzerine kurulu olmasıdır. Dolayısıyla SOAP, XML in sağladığı esneklik, kolaylık ve platform bağımsızlığı özelliklerini içerir. XML''in bütün

bilgisayar dünyası tarafından kabul görmüş sağlam bir standart olması ve uygulamalarda yaygın olarak kullanılmasından dolayı, text dosyası okuyup, XML işleyebilen bütün platform ve uygulamalar, kolaylıkla SOAP bilgisine de erişip işleyebilirler. SOAP mesaj formatı olarak XML, transfer protokolü olarakta HTTP kullanır. SOAP'e alternatif teknolojiler, kullanıldıkları mimariler aynı olsa bile, güvenlik duvarı (firewall) tarafından aralarında haberleşecekleri portlardaki veri akışına izin verilmedikçe, güvenlik duvarları arkasında varlık gösteremezler. Diğerlerinin aksine SOAP, HTTP protokolünü kullandığı ve taşıdığı mesajlar düz yazı dosyası halinde olduğu için güvenlik duvarı dostudur.

2.6.3. WSDL, Web Servisleri Tanımlama Dili (Web Service Definition Language)

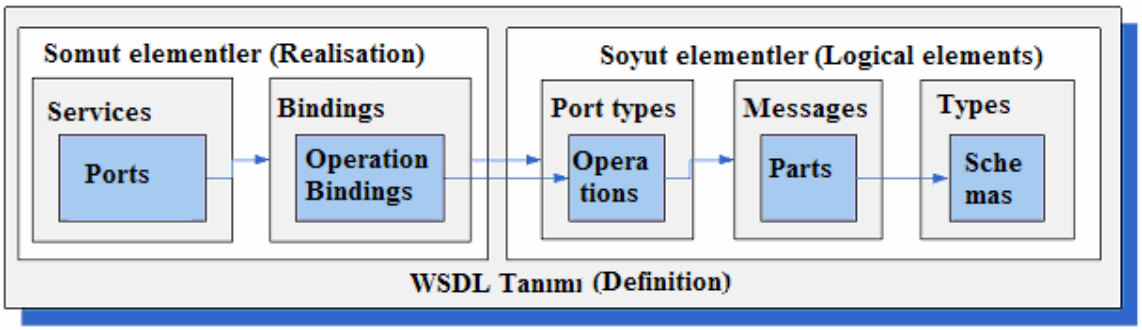
XML tabanlı bir tanımlar dağarcığı olan WSDL'in fonksiyonu, servisler için arayüz tanımlamaktır. WSDL, sağlayıcı ve arayıcıların kullanacağı istem/yanıt mesajlarının formatlarını açıklayabilecekleri soyut bir dildir.

WSDL, web servislerindeki parametreler, dönüş değerleri ve internet üzerinde hangi iletim protokolünü kullanacağı gibi ayrıntılar ile beraber, bir istemcinin web servisi ile etkileşimini belirleyen XML tabanlı bir standart sunar. Böylece bir servisin ne yaptığı, nerede bulunduğu ve nasıl harekete geçirilebileceği anlatılabilir.

Web servisleri, servis sözleşmesini belirtmek için WSDL'i kullanırlar. Servis sözleşmesi (contract), iletiler ile işletilecek uç noktaları (endpoint) kümesinin tarifi ve bu uç noktalara gönderilecek XML iletilerinin uyması gereken biçemi belirtir. Uç nokta (endpoint), WSDL'e göre biçemlendirilmiş iletileri kabul eden ağ adresidir ve port ise bir ileti için kullanılan servis uç noktasına verilen isimdir.

Şekil 2.7. da WSDL parametreleri ve elemanları gösterilmiştir. Buna göre parametrelerin tanımları şöyledir:

Şekil 2.7. WSDL dokümanının yapısı



<definitions> WSDL dokumanı tamamen XML yapısındadır ve tüm içerik definitons kök öğesi içinde bulunur.

- Somut elementler (Realisation):

  <binding> Bu bölümde web servisinin mesaj yapısı (SOAP, HTTP, MIME) ve mesaj iletim protokolu (HTTP, HTTPS, FTP, SMTP, JMS), dokümanın stili (document, rpc), her bir operasyonun tipi (literal, encoded) belirlenir.

  <service> Web servisine giriş noktalarını temsil eder. Bu giriş noktaları sayesinde istemci web servisine erişir.

- Soyut elemanlar (Logical elements):

  <operations> İşlem ve bu işleme dair girdi/çıktı parametreleri ile aykırı durumları ifade eder.

  <portType> Bu kısımda web servisi üzerinden gerçekleşecek operasyonlar tanımlanır. Her bir operasyon için istemci trafından çağrılması ve ilgili operasyonun üreteceği değerlerin istemciye hangi mesaj ile taşınacağı tanımlanır.

  <types> Bu kısımda SOAP mesajlarında kullanılacak veri tipleri tanımlanır. Burada tanımlanmış bir veri tipi başka bir veri tipinin bir elemanı olabilir.

  <message> Mesajlar bir web servisi metodu ile istemci arasındaki veri değişimini belirtir. Örneğin istemci döviz kur bilgisini istediğini belirten bir mesaj gönderdiğinde web servis metoduda içeriği farklı bir mesaj olarak geri gönderir.

  Şekil 2.8 de bir WSDL doküman örneği verilmiştir.

Şekil 2.8. WSDL doküman orneği

```
<message name="getTermRequest">
<part name="term" type="xs:string"/>
</message>
<message name="getTermResponse">
<part name="value" type="xs:string"/>
</message>
<interface name="glossaryTerms">
<operation name="getTerm">
<input message="getTermRequest"/>
<output message="getTermResponse"/>
```

```
</operation>
</interface>
<binding type="glossaryTerms" name="b1">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation>
<soap:operation soapAction="http://example.com/getTerm"/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
```

2.6.4. UDDI, Evrensel Açıklama, Bulma ve Bütünleştirme (Universal Description, Discovery and Integration)

Evrensel Açıklama, Bulma ve Bütünleştirme (UDDI), OASIS (Organization for the Advancement of Structured Information Standards) tarafından geliştirilmiş, web hizmetleri hakkında bilgi yayınlamak ve bulmak için kullanılan bir endüstri belirtimidir. UDDI kullanarak, web servis sağlayıcıları sundukları servisleri yayınlayabilir ve web servis kullanıcıları uygun servisler için arama yapabilir. Bu yüzden UDDI, sırasıyla bir veritabanı veya depo olarak sunulur. Aslında kendisi de bir web servisidir. Örneğin uddi.microsoft.com ve uddi.ibm.com

UDDI sunucuları kurum ve servis kayıt, güncelleme ve tarama işlemlerini web servisleri (SOAP mesajları) ile gerçekleştirir. Etkileşim adımları ise şöyledir:

- Standart kurumları ve yazılım şirketleri, tModel (type model ya da technical model) adlı temel belirtimleri tanımlar ve UDDI'ye kaydeder.
- Organizasyonlar iş tanımlarını ve sundukları servisleri UUID (Unique Universal Identifier) anahtarları ile kaydeder.
- Uygulamalar, arama motorları ve aracı yazılımlar, aradıkları ölçütlere uygun servisleri bulmak için UDDI kayıtlarını kullanır.
- Uygun bir servis bulunduktan sonra, uygun servis arayüzü kullanılarak servis çalıştırılır.

UDDI ın en önemli bileşenlerinden biri UDDI Kurum Kayıt Servisi (UDDI Busines Registration) dir. Bu bileşen, firmanın isminin ve bu firmaya ait web servisinin bulunduğu bir XML dosyasıdır. UDDI Kurum Kayıt Servisinin içeriği temelde üç bölümden oluşur;

- Adres, kontakt ve genel bilinmesi gereken bilgilerin bulunduğu "beyaz sayfalar",
- XML Web Servisleri ni kategoriler halinde listeleyen "sarı sayfalar" ve
- Web servisi sahibinin sunduğu, servisle ilgili teknik bilgilerin bulunduğu "yeşil sayfalar"

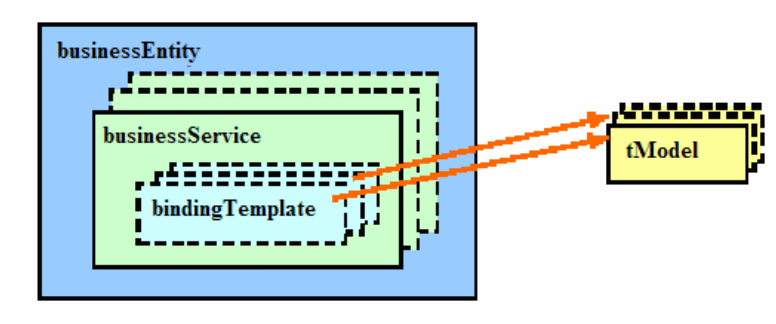
UDDI temelde XML Şeması (XML Schema) kullanır. Şekil 2.9. da gösterildiği gibi UDDI'ın XML Şemasında servis kullanıcıların servisi kullanabilmesi için bilmesi gereken üç tip bilgi vardır. Bunlar, şirket-iş bilgileri (Business Information), servis bilgileri ve bağlantı bilgileri (Service Information and Binding Information) ve servisle ilgili teknik bilgiler (Information about specificaton for services) dir.

Bu üç tip bilgi UDDI Kurum Kayıt Servisinin içeriğiyle yani beyaz, sarı ve yeşil sayfalar ile tutarlıdır. Şirket-iş Bilgileri [Business information] : Servisi veren şirketle ilgili bilgilerin bulunduğu elementtir. Bu elementin içinde isim, kontakt, açıklama, tanımlamalar, kategoriler gibi direkt şirketinizi ilgilendiren bilgiler bulunur. Xml Şemasında bu element <businessEntity> diye geçer. Servis Bilgileri [Service Information]: Servis ile ilgili teknik bilgilerin bulunduğu "yeşil sayfa" bilgileridir. Şemada, <businessEntity> elementinin alt elementleri olan <businessService> ve <bindingTemplete> elementlerden oluşur. Bunlardan ilki, yani <businessService> elementi birbirine benzer servislerin bir arada gruplandığı elementtir. Örneğin satışla ilgili servisler, siparişle ilgili servisler gibi gruplara ayrılır.

Her bir businessService elementin içinde bahsettiği servis gruplarına ait adresler bulunur. Eğer bu servis başka bir servisi de çağırması gerekiyorsa onunla ilgili bilgiler de ayrıca bulunur.

Web servisi çağırmak için gereken asıl bilgiler servis bilgilerindeki ikinci element olan <bindingTemplete> elementindedir. Bu elementin içinde <tModel> denen özel bir element vardır. Bu elementin içinde de isim, açıklama ve URL bilgileri bulunur.

Şekil 2.9. UDDI XML Şeması



2.6.5. ebXML (e-business XML)

ebXML, XML'in teknik özelliklerini kullanarak ve elektronik verinin değişimi için yöntemleri tanımlayarak bir mesajlaşma çerçevesi sunar. ebXML'in amacı değişik endüstri sektörlerinde ortaya çıkan XML tanımlarını ortadan kaldırmak değil, farklı sektörler ve bu sektörlere bağlı kuruluşlar için ortak bir birlikte çalışabilir çerçeve sunmaktır.

ebXML, OASIS ve Birleşmiş Milletlerin Ticaret ve Elektronik İş biriminin müşterek projesidir. ebXML'in geliştirilme nedeni XML kullanarak iş verilerin alış verişini sağlamaktır. ebXML açık kaynaklı bir alt yapıya sahip olup, mesajların şirketler/kurumlar arasındaki alış verişini sağlamakta ve ticari ilişkileri yönetmekte, bilinen verilerle iletişimi gerçekleştirmekte, iş süreçlerinin tanımlanması ve kaydının tutulmasını sağlamaktadır.

Bu standart, şirketlerin kritik servis bilgilerini yayınlayabileceği UDDI benzeri bir kayıt mekanizması öngörmektedir. Ancak UDDI'dan öte olarak ortaklık ve işlem anlaşmalarının da kurulmasına fırsat tanımaktadır. Bunu sağlamak için güvenlik, servis kalitesi, iş akışı gibi bir çok provizyonu gerçekleştirecek şekilde tasarlanmıştır. Bu haliyle endüstriyel uygulamalara daha çok hitab eden ebXML, şirketler arasında işlevsel sorumlulukları belirleyen İşbirliği Protokol Profilleri'nin (CPP – Colloboration Protocol Profile) oluşturulmasını ve bunların İşbirliği Protokol Anlaşmaları (CPA – Colloboration Protocol Agreement) ile karara bağlanmasını sağlamaktadır.

2.6.6. BPEL4WS (Business Process Execution Language for Web Services)

BPEL ilk olarak Microsoft'un öncülüğünde, IBM ve BEA tarafından BPEL4WS (BPEL for Web Services) olarak geliştirilmiştir.

BPEL4WS, XML tabanlı bir programlama dilidir. Bu standart ile Web servis etkileşimi tanımlanabilir ve birlikte çalışma protokolleri kullanılabilir. Ayrıca BPEL4WS dili, iş süreçleri oluşturmak amacıyla farklı aktivitelerin birleştirilmesinde de kullanılabilir.

BPEL4WS web servisleri için geliştirilmiş olan iş akışı tabanlı birleşim dilidir. Farklı tiplerde basit aktiviteler içermektedir. Bunlar sayesinde uygulamaların birbirlerini tetiklemelerine ve mesajlaşabilmelerine olanak sunmaktadır. Etkileşim sırasında bekleme sağlamakta yada veriyi bir noktadan diğer bir noktaya taşımayı desteklemektedir. Hata durumlarını gösterebilmekte ve tüm bileşimi sonlandırabilmektedir .

2.6.7. WS-Security(WS-Güvenlik) Standardı

WS-Security, web servisleri için standardize edilmiş bir güvenlik modelini öngörür. Bu anlamda mesaj bütünlüğü ve mesaj gizliliği gibi özellikleri ile karakterizedir. Mesaj bütünlüğü, mesajların imza ve lisanslar sayesinde bozulmaksızın iletilebileceği anlamına gelirken, mesaj gizliliği ise mesajların şifrelenerek gönderilebilmesini ifade eder.

WS-Security spesifikasyonları Microsoft, IBM ve Verisign tarafından geliştirilmiştir, ve daha sonra Web Servisleri Mimarisi (Web Services Architecture, WSA) olarak adlandırılmıştır. WS-Security spesifikasyonları bu alandaki bütün diğer spesifikasyonlar için temel oluşturmuş, mesaj-tabanlı güvenlik alışverişi için altyapıyı hazırlamıştır. Birlikte çalışabilir web servisleri oluşturmadaki öneminden dolayı, OASIS e verilmiş ve gerekli komite işlemlerinden sonra, resmi olarak kabul edilmiş bir standart olmuştur.

2.6.8. Diğer WS-* Standartları

Web servislerinin temel standartlarını, HTTP, XML, SOAP, WSDL ve UDDI standartları oluşturmaktadır.

WS-* olarak adlandırılan ikinci jenerasyon web servisleri standartları ise Küresel Web Servisleri Mimarisi (GXA) ni oluştumakta olup, hala geliştirilmeye açıktırlar. Tablo 2.2 de GXA tanımlamaları ve açıklamaları verilmiştir.

Tablo 2.2. GXA'nın bileşenleri

Tanım	Açıklama
WS-Security	Servislerde kimlik tanımlanması, yetkilendirme, ve diğer güvenlik işlemleri.
WS-Policy	Çalışma zamanında servis gereksinimlerini istemcilere bildirmek
WS-SecurityPolicy	Güvenlik politikası - WS-Security ile güvenlik işlemlerinde
WS-PolicyAssertions	Mesajla ilgili güvenlik politikalarının açıklamaları
WS-PolicyAttachment	Politikaların uygulanmış hali
WS-Trust	Güvenlik parçacık değişimi ve güvenli ilişkilerin yönetilmesi
WS-Privacy	Organizasyonların web servislerinin kullanılması veya yönetilmesinde kendi özel politikalarını uygulamaları
WS-Routing	SOAP mesajlarını stateless ve asenkron biçimde göndermeyi mümkün kılma
WS-Referral	SOAP router'lar ile mesaj yollarını belirleme
WS-Coordination	Standartlaşmış bir context yönetimi ve çoklu servis yapıları elde etmek
WS-Transaction	Dağıtık transaction özellikleri
WS-SecureConversation	Güvenli iletişim için altyapı oluşturulması
WS-Federation	Kerberos ve public key ınfrastructure(pkı) mekanizmalarını destekler
WS-Authorization	Güvenlik politikalarının yetkileri ve yönetimi
WS-Inspection	Web servisleri denetleme dili
DIME	Direct Internet Message Encapsulation- Binary mesaj gönderme
WS-Attachments	Mesaj içeren ek gönderme

2.7. Web Servislerinin Sundukları

Web Servislerinin sunduğu avantajlar şunlardır:

a. İşbirliktelik: Farklı platform ve işletim sistemlerindeki farklı uygulamaların beraber çalışabilmelerini sağlaması web servislerinin en önemli avantajıdır.

b. Entegrasyon: Web servisleri, farklı yer ve farklı yazılımlara sahip firmaların bütünlesmiş bir servis altında kolayca toplanabilmelerini sağlar.

c. Tekrar kullanılabilirlik: Web servisleri, hizmetlerin farklı uygulamalarda kullanılabilmesini sağlar.

d. Servis şeklinde yazılım: Yazılımların servisler seklinde sunulmasına olanak sağlayarak farklı platformlarda rahat kullanım sağlar. Bileşen olarak kullanılabilmesi esnek bir yapı sunar.

e. Erişilebilirlik: Servisler platform bağımsız olmaları sayesinde kişisel bilgisayar, cep bilgisayarı, cep telefonları gibi birçok istemciye hizmet verebilir.

f. İzolasyon: Web servisleri sağlayıcılarına internet ortamında hizmet sunma konusunda yeni bir katman oluşturarak hizmeti diğer faktörlerden izole eder ve risk faktörünü azaltır.

g. Kolay kurulabilirlik: Web servisleri standart internet teknolojileri üzerinden dağıtılır, bu nedenle dünyanın bir ucundaki ve hatta firewall (güvenlik duvarı) kullanan sunuculara bile rahatlıkla ulaşılabilir.

2.8. Standartlar Komitesi

Web servisleri standartlarını geliştiren ve destekleyen organizasyonlara kısaca göz atacak olursak bu alanda çalışan bir çok grup ve konsorsiyum bulunmaktadır. Bunların en önemlileri şunlardır:

W3C (The World Wide Web Consortium), en çok bilinen, web ile ilgili birçok standardı elinde bulunduran ve çalışma grupları biçiminde geliştiren bir gruptur. Web servisleri ile ilgilendigi konularsa, XML Şemaları, SOAP, XML-dsig, XML-enc ve WSDL standartlarıdır.

OASIS (Organization for the Advancement of Structured Information Standards.) daha çok, güvenlikle ilgili web Servislerine özel standartlarla ilgilenir. Komite bazlı çalışır ve geliştirilecek standartlar için teknik komiteyi oluşturur. OASIS, WS Security (Web Servisleri Güvenliği) ve SAML (Security Assertion Markup Language) standartlarını elinde bulundurur. SAML, XML tabanlı bir çerçeve olup kullanıcı kimlik doğrulama, yetkilendirme ve nitelik bilgilerini iç veya ortak kuruluşlara aktarmak için kullanılır.

Web Services Interoperability grubu (WS-I) birlikte çalışabilir web servisleri için genel bir yapı oluşturmak için kurulmuştur. WS-I daha çok, başka yaygın şekilde kabul edilen standardı alıp, web servis uygulamalarına uyumları için gereken profilleri ve gereksinimler geliştirmektedir. Özel olarak, BSP (Basic Security Profile-Temel Güvenlik Profili)'leri OASIS'in WS-Security (WS-Güvenlik) standartlarına dayanır ve birlikte çalışabilirlik iddia eden web servislerindeki gereken ve isteğe bağlı olan güvenlik özelliklerini belirler.

Tablo 2.3 da web servisi standartları ve standartları geliştiren kuruluşlar özetlenmiştir.

Tablo 2.3. Web Servisi Standartlarını Belirleyen Organizasyanlar

Organizasyon	Web sayfası	Tanım	Standart
World Wide Web Consortium (W3C)	www.w3c.org	500'den fazla üyesi olan ve internet standartlarını belirleyen kurum	XML SOAP HTTP WSDL
Web Services Interoperability Organization (WS-I)	www.ws-i.org	Çoğu yazılım sağlayıcısı olan, web servisleri ile ilgili standartları belirleyen kurumlar	BPEL4WS WS-Security WS-Transaction WS-Coordination WS-Attachments WS-Inspection WS-Referral WS-Routing
Organization for the Advancement of Structured Information Standards (OASIS)	www.oasis-open.org www.uddi.org	E-iş standartlarını belirleyen bir konsorsiyum	UDDI SAML 1.0
UN/CEFACT (United Nations Centre for Trade Facilitation)	www.unece.org/cefact	BM tarafından finanse edilir ve ticari standartları belirler. ebXML standartı için OASIS ile birlikte çalışmaktadır.	ebXML
The Liberty Alliance (LA)	www.projectliberty.org	Birlikte çalışabilir bir identity federation iskeleti geliştirmek için kurulmuştur. Bu iskelet SAML V.2.0 standartı ile ilişkilidir.	Liberty Identity Federation Framework (ID-FF).

Yukarıda listelenen organizasyonların yanında, web servisleri güvenlik aktivitelerini ilerleten, kalıcı kurulan ama az yaşayan, başka endüstri kuruluşları da vardır. Bunlar genelde yazılım endüstrisinin önde gelen, örnegin Microsoft, IBM, Verisign, BEA, Sun, ve digerleri gibi, ve belirli konularda çalışmak üzere katılan kuruluşlardır. Bu çoklu aktivitelerin sonuçları, belirli bir olgunluğa ulaştığında, yeni endüstri standardı olarak standartlar komitesine sunulur.

3. WEB SERVİSLERİ VE UYGULAMA ALANLARI

3.1 Elektronik İş (E-İş)

E-iş, bir işletmenin tüm iş, iletişim ve işbirliği faaliyetlerini bilgi teknolojilerini kullanarak yürütmesi ve yönetmesidir.

Günümüz e-iş dünyasında değer yaratabilmenin yolu, internet üzerinde işbirliğine, entegrasyona, bilgilere, servislere ve uygulamalara zaman ve ortamdan bağımsız erişime yönelik bir çaba ortaya koymaktan geçmektedir. E-iş'in geçirdiği gelişim sürecinin bugün geldiği noktada işletmelerin iş yapış şekillerinde köklü değişikliklere gitmeleri zorunlu olmuştur.

Şirketler internet teknolojisi yardımıyla yeni iş fırsatları yaratmak ve operasyonel verimliliğe ulaşmak için e-iş yönetim modelini uygulamaktadırlar. E-iş internet altyapısı üzerinde yapılan e-ticaret, CRM (Müşteri İlişkileri Yönetimi) , SCM (Tedarik Zincir Yönetimi) , ERP (Kurumsal Kaynak Planlama) , HR (İnsan Kaynakları Yönetimi) ve İntranet kavramlarını kapsayan bir iş yapış biçimidir.

E-İş'in ulaşmaya çalıştığı son noktayı, Bill Gates'in dijital sinir sistemi kavramıyla açıklamaya çalıştığı; mükemmel şekilde işleyen insan sinir sisteminin, işletme bünyesine uygulanması olarak ifade edebiliriz (Gates,1999). Bu sistem, işletmenin çevresini algılayabilmesini, gerekli tepkileri gösterebilmesini, rakiplerin yol açtığı tehlikeleri ve müşterilerin ihtiyaçlarının belirlenip zamanında giderici tedbirlerin alınmasını, gerekli tüm kararların gerektiği yer ve zamanda alınabilmesini sağlayan dijital işlemler bütününü ifade etmekte ve internet, intranet, extranet ağları ve özellikle web teknolojilerinin kullanılmasını içermektedir.

3.1.1. E-İş Modelleri

E-iş modellerini 4 ana grupta toplamak mümkündür:

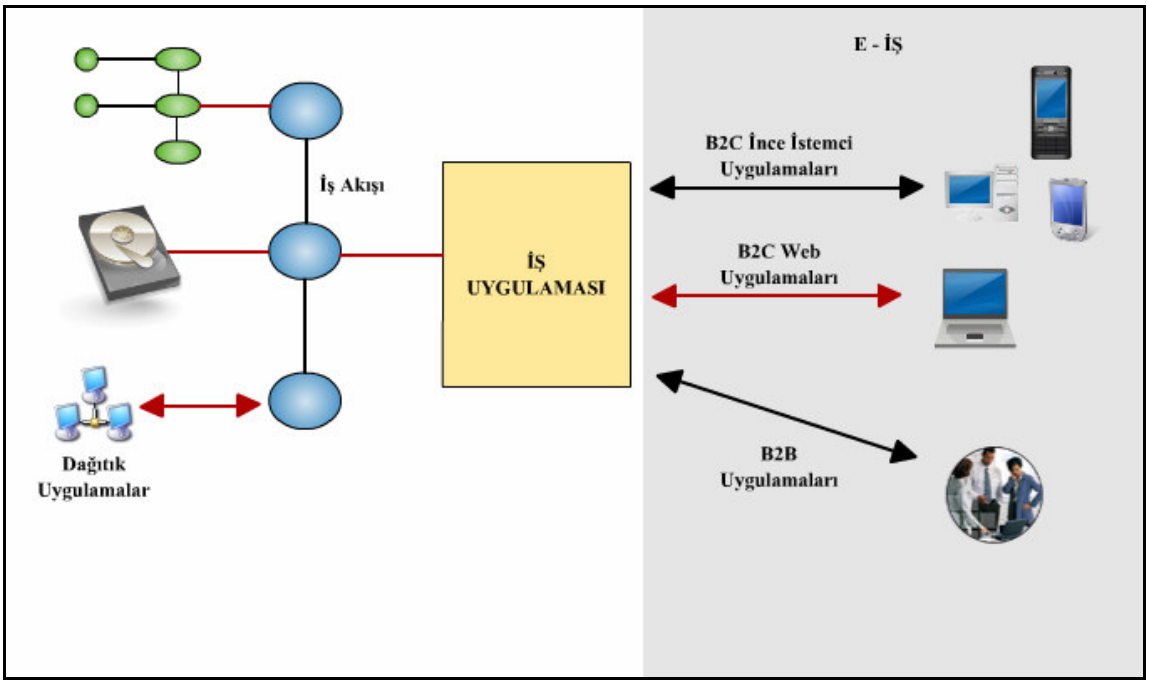
- İşten İşe (B2B - Business-To-Business) : İş dünyasında yüzlerce ortak iş ve ticaret yapan şirket, farklı endüstriler, uyumsuz bilgi sistemleri bulunmaktadır. Bugün işletmeler için en kritik problem bu farklı ortamdaki firma ve karmaşık sistemin arasındaki entegrasyonun sağlanmasıdır. Bu entegrasyon ile B2B olarak adlandırılan işletmeler arası e-ticaret gerçekleşmektedir.

Ülkemizdeki B2B altyapısını genellikle büyük firmaların oluşturduğu bayilik otomasyonunda görmekteyiz. Bu şekilde kapalı bir ağ yapısında, firmaya bağlı bayiler, tedarik, stoklama, dağıtım, pazarlama vs gibi iş süreçlerini etkin şekilde yürütme olanağına kavuşmaktadırlar. Özellikle otomotiv, beyaz eşya ve lojistik gibi sektörlerde B2B uygulamaları vazgeçilmez hale gelmiştir.

- Firmadan Müşteriye (B2C - Business-To-Customer) : Kurumlar ve kişiler arasındaki ticari ilişkileri kapsar. Genel olarak, ticari kurumların kendi web sitelerinden pazarlama ve satış yapabilmelerini sağlayan tüm işlemler B2C kapsamındadır. B2C, KOBİ (Küçük ve Orta Ölçekli İşletmeler)´ ler için tedarik pazarı olarak da nitelenebilir. KOBİ´ ler bu şekilde internet üzerinden, mal ve hizmet üretim aşamasında ihtiyaç duydukları ürün tedariklerini ya da tam tersi, başka KOBİ´ lerin ihtiyaç duyduğu ara malların toptan satışlarını kolaylıkla yapabilme olanağına kavuşabilmektedirler.
- Müşteriden Firmaya (C2B - Customer to Business) : C2B modelinde alıcılar (müşteriler) satıcılara (şirket) almak istedikleri malın veya hizmetin fiyatını belirleyerek bildirirler. Satıcı fiyatı kabul ettiği takdirde işlem gerçekleşmiş olur.
- Müşteriden Müşteriye (C2C - Customer to Customer) : Malını satmak isteyenler ve mal almak isteyen kişiler arasındaki ticari ilişkileri kapsayan e-ticaret şeklidir. Satmak istenilen mal (kitap,elektronik eşya vs..) fiyatı ile birlikte yayınlanır. Alıcılar kendi fiyat tekliflerini verir ve son kararı satıcı vererek e-ticaret gerçeklenir.
- Firmadan Çalışana (B2E - Business to Employee) : Şirketlerin intranet aracılığıyla çalışanları ile kurduğu ilişkilerin temsil edildiği e-iş modelidir.

Şekil 3.1. de e-iş ve iş modellerinin dinamik bir görüntüsü yer almaktadır. Web servisleri hem B2B hem de B2C entegrasyonuna imkan veren bir teknoloji olup, şekil üzerinde kırmızı çizgi ve oklarla ifade edilmiştir.

Şekil 3.1. Dinamik e-işten bir görünüş (Kaynak: IBM)



Web servislerinin kullanılmasının esas avantajı ucuz birlikte çalışabilirliktir. İş fonksiyonlarının web servisleri olarak sunulması, yetkili kesimlerin kullandıkları platform ve programlama diline bakılmaksızın bu servisleri çağırabilmelerine olanak verir. Bu, şirket sınırlarını aşan iş süreçlerinin entegrasyonu için gereken zamanı ve çabayı azaltır ve para tasarrufu sağlar.

3.2. Elektronik Ticaret (E-Ticaret)

OECD (Ekonomik İşbirliği ve Kalkınma Teşkilatı) e-ticareti "İnternet gibi açık, mülkiyet dışı tutanakları kullanan ve standartları belli ağlar üzerinden yapılan ekonomik işlemlerin tümü" olarak tanımlıyor.

Bugün internetin sahip olduğu potansiyel e-ticaretle birlikte daha iyi kavranmaya başlanmış ve e-ticaret yapan firmalar web servisleri mimarisini hızla oluşturmaya başlamışlardır. Önceleri yapılan uygulamalar sadece makinelerde yaşıyor, örneğin bir firma bir uygulama geliştirdiğinde bu uygulama sadece kendi içinde kalıyordu. Geliştirilen bir uygulamanın firmanın iş yaptığı bayiler ya da tedarikçilere de ulaşmak durumunda kalmasıyla beraber uygulamaların internet üzerine taşınması gerekliliği ortaya çıktı.

İnterneti daha fazla kullanmayı hedefleyen bu yaklaşıma Microsoft "Net" derken, Oracle "network services", IBM "web services", Sun ise "open network environment" adını verdiler.

E-ticaret işlemlerine birçok örnek verilebilir : Mal ve hizmetlerin elektronik alışverişi, üretim planlaması yapma ve üretim zinciri oluşturma, tanıtım, reklam ve bilgilendirme, sipariş verme, anlaşma yapma, elektronik banka işlemleri ve fon transferi, gümrükleme, elektronik ortamda üretim izleme, elektronik ortamda sevkiyat izleme, elektronik ortamda kamu alımları, elektronik para ile ilgili işlemler, elektronik hisse alışverişi ve borsa, ticari kayıtların tutulması ve izlenmesi, doğrudan tüketiciye pazarlama, sayısal imza, elektronik noter, anında bilgi oluşturma ve aktarma, elektronik ortamda vergilendirme ve fikri mülkiyet haklarının transferi gibi.

3.3. Elektronik Devlet (E-Devlet)

En yalın tanımıyla e-devlet; devletin vatandaşlara karşı yerine getirmekle yükümlü olduğu görev ve hizmetler ile vatandaşların devlete karşı olan görev ve hizmetlerinin karşılıklı olarak elektronik iletişim ve işlem ortamlarında kesintisiz ve güvenli olarak yürütülmesidir.

Web servislerinin ülkemizde en fazla uygulanış alanının e-devlet projeleri olduğunu görmekteyiz. E-devlet platformunda web servislerinin uygulandığı belli başlı örnekleri şunlardır:

- TCMB Döviz kurlarını, XML teknolojisi sayesinde bir çok yazılım eş zamanlı olarak elde edebilmektedir. (http://www.tcmb.gov.tr/kurlar/today.xml)
- Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü'nün yürüttüğü MERNİS (Merkezi Nüfus İdaresi Sistemi) projesi kapsamında yer alan Kimlik Bilgileri Paylaşım Sistemi (KPS), XML web servisleri ile yazılım geliştiricilerinin kullanımına sunulmuştur.
  (http://tckimlik.nvi.gov.tr/Web/WebServices.aspx)
- Sermaye Piyasası Kurulu da e-devlet kapsamında web servisi sunan kurumlar arasındadır. Yatırım Fonları Portföy Değerleri ve İşlem Yasaklılar listesini bu alanda yazılım geliştirenlerin hizmetine sunmaktadır.
  (http://www.spk.gov.tr/webservices/ ve
  http://www.spk.gov.tr/webservices/index_main.html)
- E-Bildirge ile SSK artık bildirgelerini XML formatında kabul edebilmektedir. Bu XML belgeleri oluşturabilmek için bir Excel belgesi sunmaktadır.

(http://www.ssk.gov.tr/sskdownloads/anasayfa/xls2xml.xls)

- Sosyal Güvenlik kurumu kapsamında oluşturulan Medula, Genel Sağlık Sigortası (GSS) ile sağlık tesisleri arasında, sağlık tesislerinin iç süreçlerine müdahale etmeksizin fatura bilgisini elektronik olarak toplamak ve hizmetlerin ödenmesini gerçekleştirmek için oluşturulmuş bir sistemdir. Kendi hastane bilgi sistemlerine sahip olan sağlık tesisleri GSS ile web servisleri üzerinden haberleşmektedir.

  (http://www.ssk.gov.tr/sgk/medula.html)
- Web Servisleri Elektronik Fatura Sistemi (w-ELF), SSK'nın özel sağlık tesisleri için geliştirmiş olduğu yeni provizyon ve geri ödeme sistemi olup, sistem entegrasyonu web servisleri ile sağlanmıştır.

  (http://www.ssk.gov.tr/wps/portal/!ut/p/_s.7_0_A/7_0_5RS?cpid=577)

Ayrıca bu örneklerin dışında, TAKBİS (Tapu ve Kadastro Bilgi Sistemi), Say2000 (Saymanlık Otomasyon Sistemi), VEDOP (Vergi Daireleri Otomasyon Projesi), POLNET (Polis Bilgisayar Ağı), DMO elektronik satış uygulaması,. GİMOP (Gümrük İdarelerinin Modernizasyonu Projesi) uygulamalarında da web servislerinden yararlanılmıştır.

4. GELECEĞİN MİMARİSİ : SOA (SERVİS YÖNELİMLİ MİMARİ)

Web servisleri, XML ve ilgili standartların oluşturulmasıyla başlayan birlikte çalışılabilirlik, çoklu platform kullanımı, farklı uygulamalar arasındaki entegrasyon, bugün yeni bir sistem mimarisine öncülük etmiştir. Servis Yönelimli Mimari – Service Oriented Architecture – SOA

Servis Yönelimli Mimari, dağıtık uygulamalardaki kabiliyetlerin yapıtaşları (servis) şeklinde yeniden tasarlanmasına ve bu yapıtaşların farklı şekillerde birleştirilmesiyle yeni iş servislerinin oluşturulmasını sağlayan bir mimaridir.(Oracle)

Servis yönelimli mimarinin prensipleri şunlardır:

1. Servisler tekrar kullanılabilirler
2. Servisler ortak bir anlaşmayı(kontratı) paylaşırlar.
3. Servisler gevşek bağlaşımlıdır(loosely coupled).
4. Servisler alt yapısındaki mantıktan kullanıcıları soyutlarlar.
5. Servis yönelimli mimaride mevcut servisler geliştirilebilir ve genişletilebilir olmalıdır.
6. Servisler otonomdur.

7. Servisler durum bilgisi içermezler.
8. Servisler keşfedilebilir olmalıdır.

SOA'nın yapısını oluşturan web servisleri ve XML, kurumlar arası birlikte işlerlik sağlayacak standartların oluşmasındaki anahtar teknolojilerdir. B2C hareketlerinin interaktif kullanıcıları normalde web uygulamalarına erişirken, web servisleri, SOA modeli kullanarak B2B zincirleri oluştururlar.

SOA modeli de web servislerinde olduğu gibi şu üç ana unsurdan oluşur:

- Servis Sağlayıcı, servisi kurar ve bunu bir Servis Brokerı'na kaydolarak "yayınlar"
- Servis Arayıcı, Servis Brokerı'nın kayıtlarını arayarak gerek duyduğu servisi "bulur"
- Servis Arayıcı, sunulan servisi kullanmak üzere, Servis Sağlayıcıya "bağlanır"

Servis Yönelimli Mimari, dağıtık uygulamaların implementasyonu için ideal mimaridir. Bugün pek çok iş uygulaması aşırı 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 sebebtir. Ayrıca bu durum, değişen iş gereksinimlerinin karşılanmasını ve ileride yapılması muhtemel modifikasyonları da zorlaştırır.

Servis yönelim bir mimari için XML entegrasyonun şu aşamalarda yapılması gerekir:

- Kurumsal mevcut sistemlerin entegrasyonu : Kullanılmakta olan sistem, uygulama ve veri tabanlarından mevcut yapıya ilişkin bilgi alınır, daha sonra bu yapı web servisleri matığına göre yeniden şekillendirilir. Böylece mevcut sistemlerdeki verilerin geri dönüşümü (yeni sisteme göre kullanılmaları) sağlanır. Entegrasyon projesinin karmaşıklığı ve maliyeti azaltılmış olur.

- Kurumsal servis entegrasyonu : Standart tabanlı XML kullanarak, mevcut IT kaynaklarının ve servislerin kurum tarafından kontrolü sağlanır. Böylece servislerin, esnek mimari sayesinde, kurumsal gereksinimlerle uyumluluğu arttırılır. Çoklu uygulamaların (multiple applications) bilgi, akış ve yapılarını kümeleyen (bir araya getiren) esnek, kompozit uygulamalar yaratır. Satıcıların standart tabanlı mimarilere bağlı kalmasını önler.

- Kurumsal bilgi entegrasyonu : Kullanıcıların iş terimleri kullanarak aradıkları bilginin, formatından ya da bulunduğu sistemden bağımsız olarak veri kaynaklarında bulunması,

erişilebilir olması ve taşınmasını sağlayan bir gateway oluşturur. Böylece kullanıcının ya da iş akışının ihtiyacına göre veri sunumları yapılabilir ya da diğer türdeki sunumlarla birleştirilir. Her türlü bilgisayara ve uygulamalara tam zamanlı bilgi taşınır.

Günümüzün değişen rekabet koşullarında şirketlere önemli avantajlar sağlayan Servis Odaklı Mimari web servislerinin ortaya çıkmasından sonra bilişim dünyasında daha fazla bilinen bir kavram haline gelmiştir. Büyüme ivmesi yakalayan şirketler, zamanla karmaşık yazılım mimarilerini yönetmekte zorlanmaya başlamışlardır. Geleneksel mimarilerin yeterli olmadığını kavrayan kuruluşlar, günümüzün rekabet koşullarına daha hızlı uyum sağlayabilmek için yeni yollar aramaya yönelmişlerdir. SOA, farklı sistemler, uygulamalar ve organizasyonlar arasında sağlam, esnek, güvenilir ve genişletilebilir bir bütünleşme altyapısının kurulmasını sağlamaktadır. SOA sayesinde, kurum süreçlerinin büyük çoğunluğu standart bir otomasyon temeline oturtulmakta ve iş süreçleri bu şekilde hızlanmaktadır.

Batı Avrupa ve ABD'de yapılan araştırmalara göre büyük işletmelerin yaklaşık yüzde 70'i SOA uygulamalarını hayata geçirmiş durumdadır. Bunların yüzde 29'u SOA'ya işletme seviyesinde tam destek verirken, yüzde 19'u ise SOA'yı iş dönüşüm stratejilerinin önemli bir parçası olarak görmektedir. Günümüzde artık SOA'nın yeni yazılım geliştirme modeli olacağına ve 40 yıllık yazılım mimarilerini ortadan kaldıracağına kesin gözüyle bakılmaktadır. SOA pazarının ise dünyada önümüzdeki 3-4 yıl içinde 200 milyar doları aşacağı tahmin edilmektedir.

Günümüzde birçok sektörden şirket SOA platformuna geçmek isterken, SOA'ya ilişkin olarak en iyi örnekler telekom ve finans sektöründe gerçekleşmektedir. Önümüzdeki yıllarda orta katman, iş orkestrasyonu ve servis kayıt yapıları gibi SOA bileşenlerinin daha standart hale getirilmesi gerektiğine dikkat çekilmektedir.

Artık SOA ve SOA yönetimi birçok büyük firmanın kapsamına girmiş durumdadır. Bundan amaç son 10 yılda internet ve altyapıya yapılan büyük yatırımın, gerçekten yüksek verim ve düşük maliyet ile geri kazanılması düşüncesidir. Analistler önümüzdeki birkaç yıl içinde şirketlerin SOA'ya 20 milyar dolardan daha fazla yatırım yapmasını öngörmektedir.

5. WEB SERVİSLERİ İLE ÖRNEK BİR ELEKTRONİK TİCARET UYGULAMASI

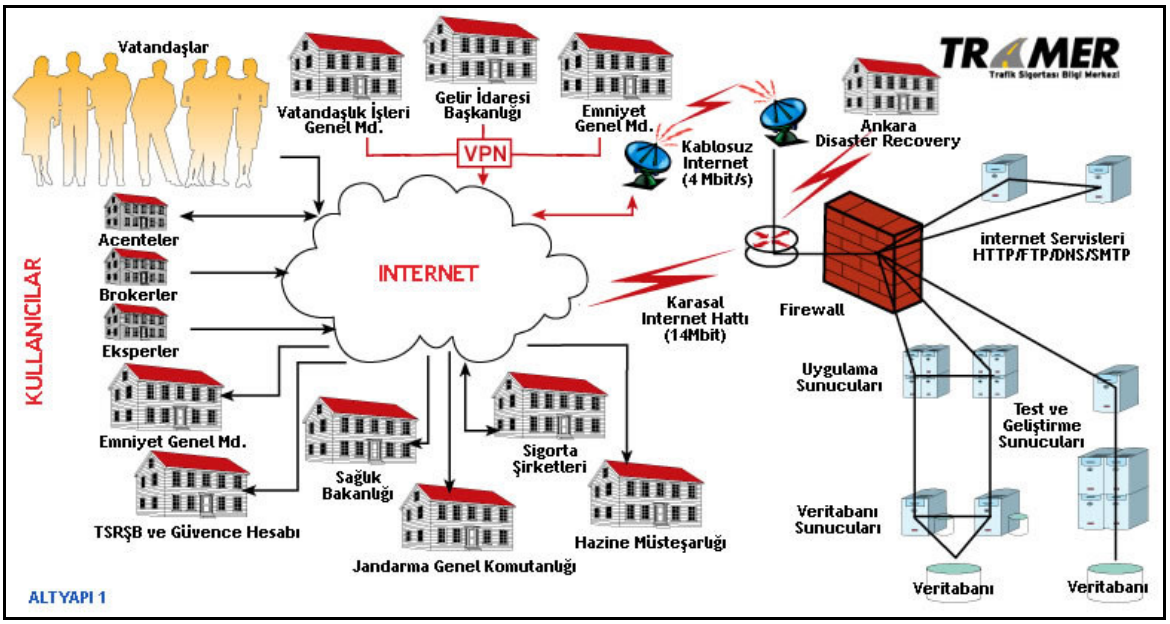
5.1. Araç Sigortalama ve TRAMER Hakkında

Sigortacılık ürün ve hizmet çeşitliliği açısından günümüzün karmaşık iş yapılarından birisidir. Sigortalanacak konular tek tek ele alındığında, her bir konu riskler, fiyatlandırma ve satış bakımından birbirlerinden farklılık göstermektedir.

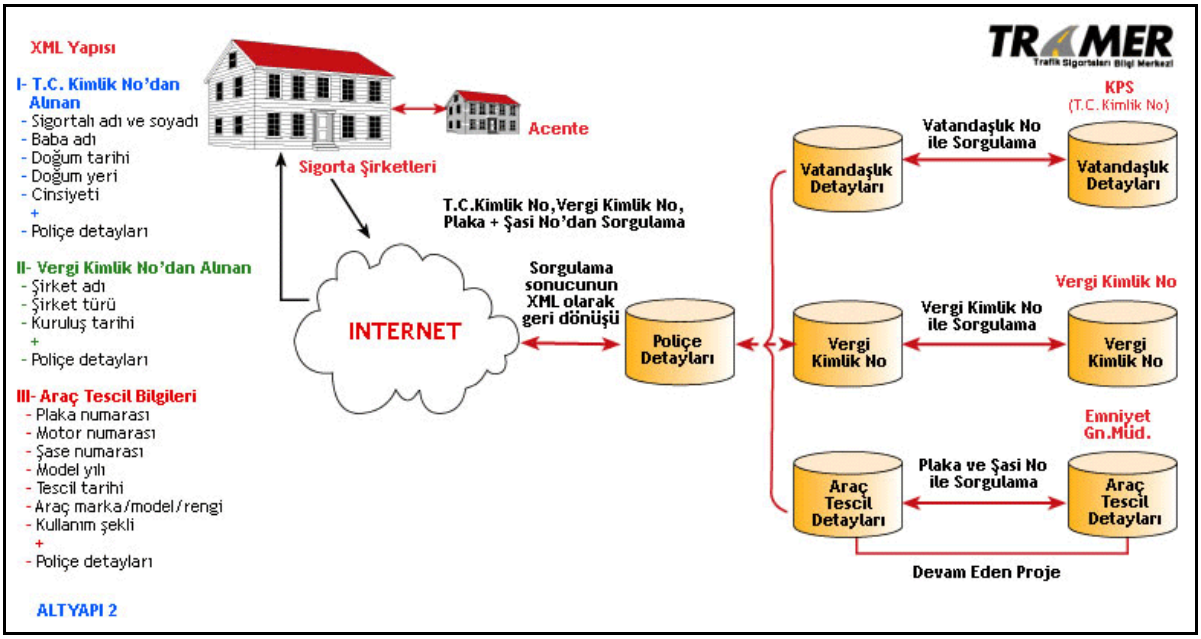
En bilinen sigorta konularından biri olan araç sigortalama, sigorta şirketlerinin gerek birbirleri gerekse ilgili pek çok kurum ile bilgi ve hizmet paylaşımını gerektirir. Aracın ve sahibinin geçmiş sicili, yeni yapılacak sigorta poliçelerini doğrudan etkiler. Bu konuda doğru bilgiye ulaşmada, gerektiğinde geçmiş poliçeler için diğer sigorta şirketlerinin kayıtları, kaza, hasar ve trafik cezaları için emniyet kayıtları, vergi borçları, satış ve devir gibi konular için maliye kayıtları, şahıs bilgilerinin doğruluğu için nüfus idaresi kayıtları gibi geniş bir alanın taranması gerekir. Tüm bu işlemlerin yazışmalar veya doğrudan şahıslar ile yapılması büyük zaman ve iş gücü kaybına yol açacağı gibi doğru bilgiye ulaşmayı da güçleştirmektedir. Bu amaçla tüm poliçe bilgileri ve bunların hasar ve ödeme kayıtlarının tutulduğu ilişkisel bir veritabanı oluşturulmuştur. Trafik Sigortaları Bilgi Merkezi (TRAMER) adı verilen bu sistemin içinde bulunan kullanıcılar yetkileri çerçevesinde veritabanlarındaki kayıtları sorgulayabilmektedirler.

TRAMER'in amacı, tüm sigorta şirketlerinin trafik sigortası poliçe kayıtlarının tutulduğu ilişkisel bir veritabanı oluşturmak, trafik sigortasına ilişkin hasar verilerini almak ve bu kayıtları sigorta kayıtları ile ilişkilendirmektir. Şekil 5.1 de TRAMER'in alt yapısı ve hizmet verdiği kullanıcılar gösterilmiştir. Şekil 5.2. de ise sigorta şirketlerinin, TC Kimlik no, vergi no, motor no ve saşi no parametreleri ile TRAMER den XML tabanlı olarak bilgi sorgulaması resmedilmiştir.

Şekil 5.1. TRAMER Altyapısı ve Kullanıcıları (Kaynak:Tramer)



Şekil 5.2. TRAMER den Bilgi Sorgulama (Kaynak:Tramer)



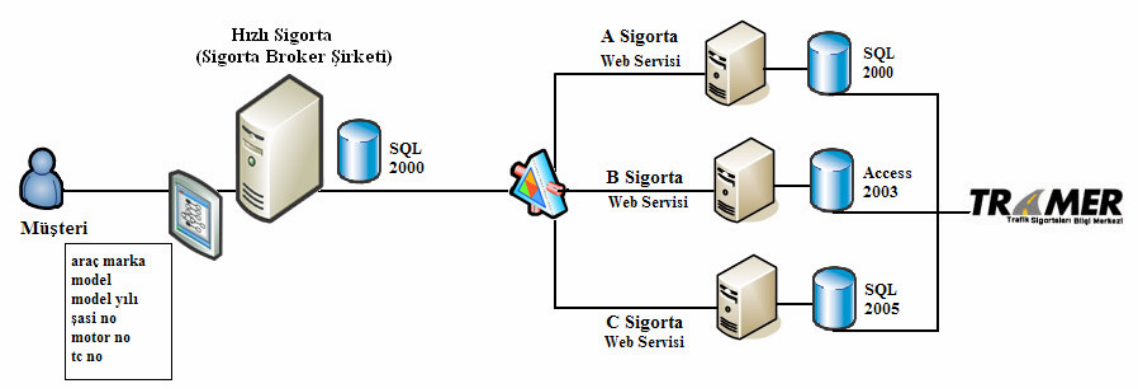
5.2. Uygulama Hakkında

Bu bölümde araç sigortalama konusunda çok sayıda sigorta şirketinden web servisleri kullanarak teklifler alıp, bu teklifleri sigorta poliçesi yaptırmak isteyen müşterilerine, karşılaştırmalı olarak sunan bir sigorta broker firmasının elektronik ticaret çözümü incelenmiştir.

Sigorta poliçesi almak isteyen bir müşteri noktasından baktığımızda, doğru sigorta poliçesinin satın alınması, hangi sigorta şirketinin seçilmesi gerektiği ile başlayan zorlu bir süreçtir.

Sigorta broker firması noktasından baktığımızda ise çeşitli tercih ve özelliklere göre sigorta şirketlerinin ürünlerinin araştırılması, incelenmesi ve müşterilere sunulması işi oldukça zaman kaybettirici ve maliyetli bir iştir. Bunun çözümü olarak sigorta şirketlerinin ve brokerin birlikte çalışabilirliliğini sağlayan ve müşterilere uygun ürünleri sunabilen, web servislerini barındıran örnek bir uygulama Şekil 5.3. deki gibi temsil edilmiştir.

Şekil 5.3. Müşteri, Sigorta Brokeri ve Sigorta Şirketleri



Uygulamada 3 ayrı bağımsız sigorta şirketi ve bu şirketlerin ürünlerini karşılaştırmalı olarak sunan bir broker şirketi mevcuttur. Taraflar şunlardır:

A Sigorta Şirketi

B Sigorta Şirketi

C Sigorta Şirketi ve

Hızlı Sigorta Broker Şirketi

Burada A, B ve C Sigorta şirketleri, bir broker aracılığı ile müşterilere, istedikleri özelliklere uygun teklifleri kendi web servisleriyle sunarlar. Aslında birer web servisi sunucusu görevini yaparlar. Diğer yandan A, B ve C Sigorta şirketleri brokera teklifleri göndermeden önce TRAMER'den sorgulama yapmaları gereklidir. Bu durumda A, B ve C Sigorta şirketleri TRAMER'den web servisini kullanan birer istemci rolündedirler.

Uygulamada müşteriye ilgilendiği teklifleri sunan, Hızlı Sigorta broker şirketi ile A, B ve C Sigorta şirketleri arasındaki süreç konu alınmıştır. TRAMER den bilgi sorgulama kısmı uygulamaya dahil edilmemiştir.

5.3. Uygulamanın Akış Süreci

Aracına kasko yaptırmak isteyen müşteri ilk olarak Şekil 5.3. deki arayüz ile aracına ait özellikleri girmektedir. * ile gösterilen alanlar girilmesi zorunlu alanlardır. Aslında girilen bu bilgiler, ilgili sigorta şirketlerinin tekliflerini gönderebilmesi için istediği parametrelerdir. Bu parametreler şunlardır: araç markası, araç modeli, aracın model yılı, aracın plakası, aracın şasi numarası ve motor numarası ile ruhsat sahibinin TC kimlik numarası ve adı soyadıdır.

Şekil 5.4. Hızlı Sigorta uygulamasının önyüzü

Müşteri daha sonra Teklifleri Göster tuşuna tıklar ve Şekil 5.5 deki ve Şekil 5.6 daki gibi örnek kasko poliçe teklifleri ve ilgili poliçenin teminat kapsamları ekrana gelir.

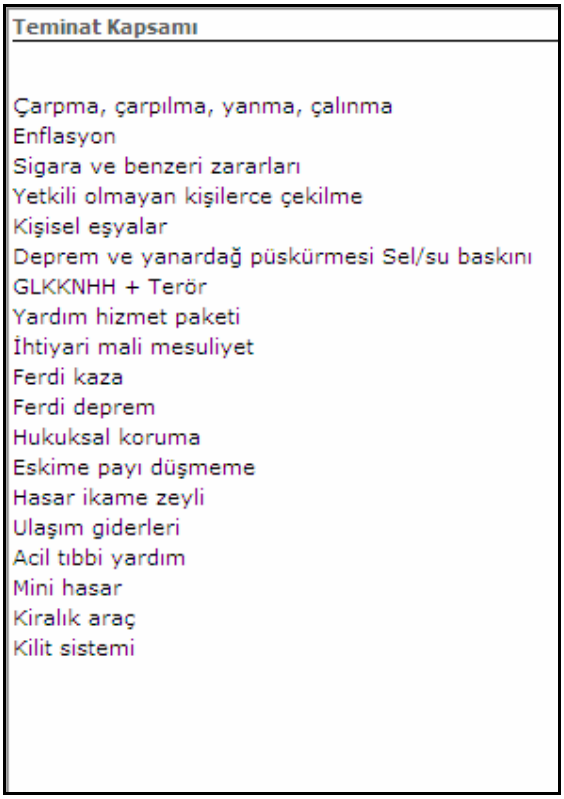
Şekil 5.5. Sigorta şirketlerinin sunduğu kasko teklifleri

KASKO TEKLİFLERİ

	Sigorta Şirketi	Teminat Kapsamı	İndirim Oranı Açıklama	Peşin Fiyatı	Taksitli Fiyatı	Ödeme Seçenekleri
◉	A Sigorta	Teminat Detayı...	%15 %10 hasarsızlık + %5 peşin ödeme	1.388,45 YTL	1.446,33 YTL	1 peşin +8 taksit
○	B Sigorta	Teminat Detayı...	%5 peşin ödeme	840 YTL	910 YTL	6 taksit
○	C Sigorta	Teminat Detayı...	%10 %5 hasarsızlık + %5 peşin ödeme	809,05 YTL	894,27 YTL	Bonus karta 6 taksit

SATIN AL

Şekil 5.6. Kasko poliçesindeki teminat detayları ekranı



Teminat Kapsamı

Çarpma, çarpılma, yanma, çalınma
Enflasyon
Sigara ve benzeri zararları
Yetkili olmayan kişilerce çekilme
Kişisel eşyalar
Deprem ve yanardağ püskürmesi Sel/su baskını
GLKKNHH + Terör
Yardım hizmet paketi
İhtiyari mali mesuliyet
Ferdi kaza
Ferdi deprem
Hukuksal koruma
Eskime payı düşmeme
Hasar ikame zeyli
Ulaşım giderleri
Acil tıbbi yardım
Mini hasar
Kiralık araç
Kilit sistemi

5.4. Uygulamanın Mimarisi ve Bileşenleri

Her biri ayrı sistem ve veritabanlarına sahip sigorta şirketleri, broker firmasına istenilen teklifleri web servisleri üzerinden XML tabanlı olarak göndermektedirler. Firmalardan,

A Sigorta firması SQL 2000 veritabanını,

B Sigorta firması Access 2003 veritabanını,

C Sigorta firması SQL 2005 veritabanını,

Sigorta broker firması ise SQL 2000 veritabanını kullanmaktadırlar.

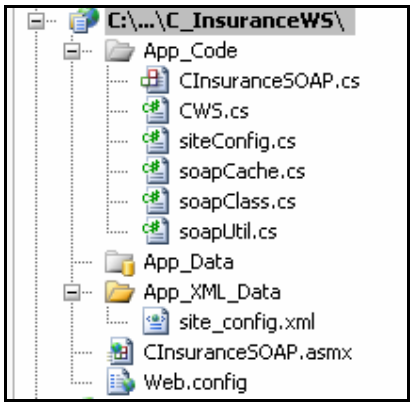
Farklı veritabanları düşünülerek tasarlanmış olan uygulama ASP.NET platformunda Visual Studio 2005'de C# dili ile yazılmıştır. Veritabanları uygulama için örnek tablolar barındırmakta olup, SQL veritabanı tarafında yordamlar (stored procedures) kullanılmıştır.

Tablo 5.1. Uygulamadaki web servisi metodları ve görevleri

Metod Adı	**Görevi**
TestToServiceIsUP()	Web servisinin aktif olup olmadığının kontrolünü sağlar
GetAQuotes()	A Sigortasının tekliflerini gönderir
GetAQuoteDetail()	A Sigortasının sunduğu poliçenin teminat kapsamının verir
GetBQuotes ()	B Sigortasının tekliflerini gönderir
GetBQuoteDetail()	A Sigortasının sunduğu poliçenin teminat kapsamının verir
GetCQuotes	C Sigortasının tekliflerini gönderir
GetCQuoteDetail()	A Sigortasının sunduğu poliçenin teminat kapsamının verir

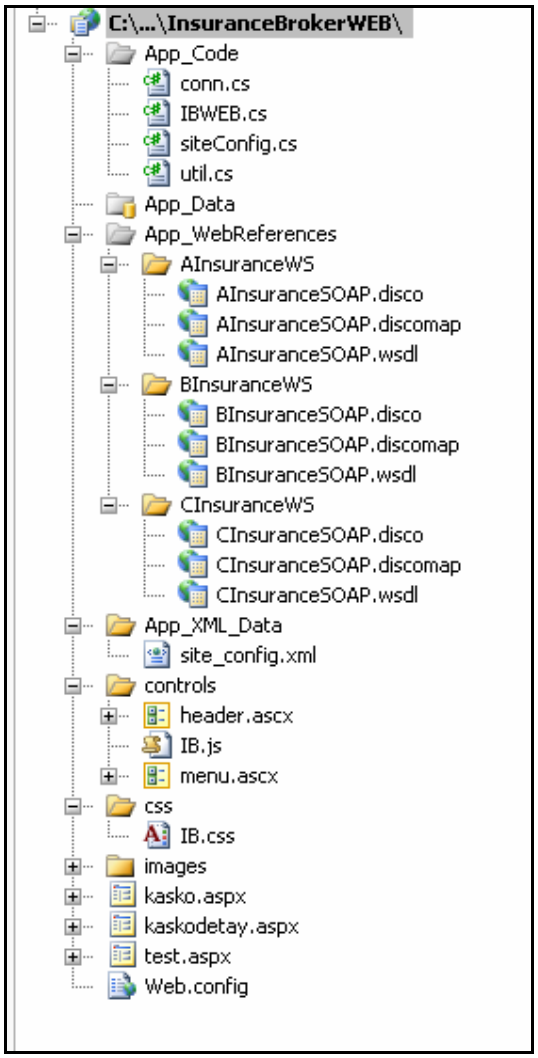
Şekil 5.7 da C sigortasının web servis proje ekranı gösterilmiştir.

Şekil 5.7. C Sigota Şirketi web servisi proje ekranı



Şekil 5.8 de uygulamadaki broker şirketinin proje ekranı gösterilmiştir.

Şekil 5.8. Hızlı Sigorta şirketinin web projesi ekranı



Uygulamadaki web servisleri metodlarının ve kullanılan bazı sınıfların kodları eklerde ayrıca sunulmuştur.

6. SONUÇ

Özet olarak, elektronik iş dünyasında var olup değer yaratabilmenin yolu, internet üzerinde işbirliğine, entegrasyona, bilgilere, servislere ve uygulamalara zaman ve ortamdan bağımsız erişime yönelik bir çaba ortaya koymaktan geçmektedir.

XML ve web servislerinin, iş modelleri, iş süreçleri ve yazılım süreçleri gibi birçok alan ve sektörde değişikliklere yol açtığını görmekteyiz. Tüm bu model ve süreçteki değişiklikler sayesinde, önümüzdeki yıllarda artık tüm kurumların mevcut mimarilerini servis odaklı hale getirmeye başlamaları sözkonusudur.

Bugün elektronik ticareti yapabilmek için şirketin boyutu önemli olmadığı gibi interneti bilip bilmemekte hatta bilgisayar kullanıp kullanmamakta elektronik iş yapabilmek için bir kıstas değildir. Oysa o veya bu şekilde firmalar kendi işlerini internete, servis yönelimli modellere ve birlikte çalışabilirliliğe adapte etmezlerse, gelecekte bu firmaların iş yapma şansları son derece azalacaktır.

"İleride web sayfalarından çok web servisleri olacaktır. Şirketler internet üzerinde yeni iş modelleri oluşturacaklar ve bu modellerde web servisleri geçerli olacaktır."

Frank Moss

Massachusetts Teknoloji Enstitüsü Medya Laboratuvarı Başkanı

KAYNAKLAR

ALGAN, Sefer "C# Web Servislerine Giriş", 2003
Erişim : http://www.csharpnedir.com/makalegoster.asp?Mid=49

AKYOKUŞ, Doç. Dr. Selim, "Web Servisleri:Internet Devriminde İkinci Aşama", 2001

CHİUSANO, Joseph M.- HAMİLTON, Booz Allen, "Web Services Security and More"
Erişim : http://www.developer.com/design/article.php/2171031

CLEMENTS, Tom, "Overview of SOAP", Sun Developer Network, 2002
Erişim : http://java.sun.com/developer/technicalArticles/xml/webservices/

GISOLFI, Dan, "Web services architect: Part 1", 2001
Erişim : http://www.ibm.com/developerworks/webservices/library/ws-arc1/

GLASS, Graham,"The Web services (r)evolution: Part 1", 2001
Erişim : http://www.ibm.com/developerworks/webservices/library/ws-peer1.html

GLASS, Graham,"The Web services (r)evolution: Part 4", 2001
Erişim : http://www.ibm.com/developerworks/webservices/library/ws-peer4/?n-ws-141

GOTTSCHALK-GRAHAM et al, "Introduction to Web services architecture", 2001
Erişim : http://www.research.ibm.com/journal/sj/412/gottschalk.html

KALMAN, Selin, BTHaber, "İş süreçlerinin maestrosu: SOA"
Erişim : http://www.bthaber.com/haber.phtml?yazi_id=590000151

LEVITT, Jason, "From EDI To XML And UDDI: A Brief History Of Web Services", InformationWeek, 2001
Erişim : http://www.informationweek.com/story/IWK20010928S0006

MCGOVERN, James-AMBLER, Scott W. et al, "A Practical Guide to Enterprise Architecture", Prentice Hall

MYERSON, Judith, "Advancing the Web services stack", 2002
Erişim : http://www.ibm.com/developerworks/webservices/library/ws-wsa/

SÜZER, Hande D., Capital Dergi, "Yarının IT Stratejisi", 2002
Erişim : http://www.capital.com.tr/haber.aspx?HBR_KOD=1524

SWITHINBANK, Peter- GIGANTE, Francesca et al, "WebSphere and .NET Interoperability Using Web Services", IBM RedBooks, 2005

TAŞDELEN, Aykut, Erişim : http://www.aspnedir.com/Article/Articles.aspx?CatID=10

WALSH, Norman, "A Technical Introduction to XML"
Erişim : http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN84

WALSH, Norman, "What is XML?", O'reilly xml.com
Erişim : http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN84

ŞENYURT, Burak Selim, "XML Web Servisleri"
Erişim : http://www.csharpnedir.com/makalegoster.asp?MId=370

W3C Working Group Note,"Web Services Architecture",2004
Erişim : http://www.w3.org/TR/ws-arch/#whatis

Diğer Kaynaklar

Innova Bilişim Çözümleri, Erişim : http://inet-tr.org.tr/inetconf11/bildiri/102.doc

OWASP Guide 2.0.1-2005

Software AG, Erişim : http://www.softwareag.com

Technology Reports, http://xml.coverpages.org/wsdl.html
Ulusal Yazılım Mimarisi Konferansı, 20-21 Kasım 2006
http://www.java.name.tr
http://www.msakademik.net
http://www.soaprinciples.com

EKLER

Ek 1. AInsuranceSOAP.cs

```
    /// <summary>
    /// Summary description for AInsuranceSOAP
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    public class AInsuranceSOAP : System.Web.Services.WebService
    {
        AWS IWS = new AWS();
        [WebMethod()]
        public string TestToServiceIsUP()
        {
            return "A Web Service is UP " + System.DateTime.Now;
        }

        [WebMethod()]
        public soapClass.xmlResult getAQuotes(string marka, string model, string modelyil,
string sasino, string motorno, string tcno)
        {
            soapClass.xmlResult rtn = new soapClass.xmlResult();
            DataTable tbl = new DataTable();
            SqlConnection conn;
            conn = IWS.getConn();
            try
            {
                if (conn.State != ConnectionState.Open)
                    conn.Open();
                SqlCommand cmd = new SqlCommand("execute pATeklifi @aracmarka = '" +
marka + "'", conn);
                SqlDataAdapter dta = new SqlDataAdapter();
                dta.SelectCommand = cmd;
```

```
            dta.Fill(tbl);
            tbl.TableName = "t_AQuotes";
            rtn.XML = soapUtil.getConvertToXml(tbl);
            rtn.success = true;
            tbl.Dispose();
        }
        catch (Exception ex)
        {
            rtn.exception = ex.Message;
        }
        finally
        {
            conn.Close();
        }
        return rtn;
    }

    [WebMethod()]
    public soapClass.xmlResult getAQuoteDetail(int teklifno)
    {
        soapClass.xmlResult rtn = new soapClass.xmlResult();
        DataTable tbl = new DataTable();
        SqlConnection conn;
        conn = IWS.getConn();
        try
        {
            if (conn.State != ConnectionState.Open)
                conn.Open();
            SqlCommand cmd = new SqlCommand("execute pATeklifDetay @teklifno = " +
teklifno, conn);
            SqlDataAdapter dta = new SqlDataAdapter();
            dta.SelectCommand = cmd;
```

```
            dta.Fill(tbl);
            tbl.TableName = "t_AQuoteDetail";
            rtn.XML = soapUtil.getConvertToXml(tbl);
            rtn.success = true;
            tbl.Dispose();
        }
        catch (Exception ex)
        {
            rtn.exception = ex.Message;
        }
        finally
        {
            conn.Close();
        }
        return rtn;
    }
}
```

Ek 2. SOAPcache.cs

```
    /// <summary>
    /// Summary description for soapCache
    /// </summary>
    public class soapCache
    {
        CWS IWS = new CWS();

        public DataTable getCQuotes_DataTable(string marka, string model, string modelyil,
string sasino, string motorno, string tcno)
        {
            DataTable dt;
            dt = (DataTable)System.Web.HttpContext.Current.Cache["t_CQuotes"];
            if (dt == null)
            {
                int sure = siteConfig.getGeneralCacheExpirationTime();
                dt = new DataTable();
                SqlConnection conn;
                SqlCommand cmdSelect;
                SqlDataAdapter dta = new SqlDataAdapter();
                conn = IWS.getConn();
                if (conn.State != ConnectionState.Open)
                    conn.Open();
                cmdSelect = new SqlCommand("SELECT * FROM tTeklif WHERE aracmarka='"
+ marka + "'", conn);
                dta.SelectCommand = cmdSelect;
                dta.Fill(dt);
                dt.TableName = "t_CQuotes";
                System.Web.HttpContext.Current.Cache.Add("t_CQuotes", dt, null,
DateTime.Now.AddMinutes(sure), System.Web.Caching.Cache.NoSlidingExpiration,
System.Web.Caching.CacheItemPriority.High, null);
                conn.Close();
```

```
        }
            //Response.Write("getting from Database");
        else
        {
            //Response.Write("getting from Cache");
        }
        return dt;
    }

    public DataTable getCQuoteDetail_DataTable(int teklifno)
    {
        DataTable dt;
        dt = (DataTable)System.Web.HttpContext.Current.Cache["t_CQuoteDetail"];
        if (dt == null)
        {
            int sure = siteConfig.getGeneralCacheExpirationTime();
            dt = new DataTable();
            SqlConnection conn;
            SqlCommand cmdSelect;
            SqlDataAdapter dta = new SqlDataAdapter();
            conn = IWS.getConn();
            if (conn.State != ConnectionState.Open)
                conn.Open();
            cmdSelect = new SqlCommand("SELECT * FROM tTeklifDetay WHERE
teklifno=" + teklifno, conn);
            dta.SelectCommand = cmdSelect;
            dta.Fill(dt);
            dt.TableName = "t_CQuoteDetail";
            System.Web.HttpContext.Current.Cache.Add("t_CQuoteDetail", dt, null,
DateTime.Now.AddMinutes(sure), System.Web.Caching.Cache.NoSlidingExpiration,
System.Web.Caching.CacheItemPriority.High, null);
            conn.Close();
```

```
        }
        //Response.Write("getting from Database");
        else
        {
            //Response.Write("getting from Cache");
        }
        return dt;
    }
}
```

Ek 3. SOAPclass.cs

```
/// <summary>
/// Summary description for soapClass
/// </summary>
public class soapClass
{
    public class soapResult
    {
        public bool success = false;
        public string exception = "";
        public string msg = "";
        public int number = 0;
        public byte status = 1;
    }

    public class xmlResult
    {
        public bool success = false;
        public string exception = "";
        public string XML = "";
        public byte status = 1;
    }

    public class AuthHeader : SoapHeader
    {
        private string _memInfo_Xml = "";
        private bool _IsAuthenticated = false;



        public string memInfo_Xml
        {
            get { return _memInfo_Xml; }
```

```
            //_result_Xml = value
            set { }
        }

        public bool IsUserAuthenticated(ref string rtn)
        {
            try
            {
                _IsAuthenticated = true;
            }

            // _IsAuthenticated = False

            catch (Exception ex)
            {
                _IsAuthenticated = false;
                rtn = "Error : " + ex.Message;
            }
            return _IsAuthenticated;
        }
    }

}
```

Ek 4. SOAPutil.cs

```
/// <summary>
/// Summary description for soapUtil
/// </summary>
public class soapUtil
{

    public static string getConvertToXml(DataSet ds)
    {
        string rtn;
        System.IO.StringWriter sw = new System.IO.StringWriter();
        XmlTextWriter xmltw = new XmlTextWriter(sw);
        ds.WriteXml(xmltw, XmlWriteMode.WriteSchema);
        rtn = sw.ToString();
        sw.Dispose();
        return rtn;
    }

    public static string getConvertToXml(DataTable tbl)
    {
        string rtn;
        System.IO.StringWriter sw = new System.IO.StringWriter();
        XmlTextWriter xmltw = new XmlTextWriter(sw);
        tbl.WriteXml(xmltw, XmlWriteMode.WriteSchema);
        rtn = sw.ToString();
        sw.Dispose();
        return rtn;
    }

    public static XmlTextReader strToXmlTextReader(string strXml)
    {
        System.IO.StringReader sw = new System.IO.StringReader(strXml);
```

```
        XmlTextReader xmltR = new XmlTextReader(sw);
        return xmltR;
    }
}
```

Ek 5: Uygulamadan bir WSDL Belgesi (CInsuranceSOAP.asmx?wsdl)


```
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
- <wsdl:types>
- <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
- <s:element name="TestToServiceIsUP">
  <s:complexType />
  </s:element>
- <s:element name="TestToServiceIsUPResponse">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="TestToServiceIsUPResult"
type="s:string" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="getCQuotes">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="marka" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="model" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="modelyil" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="sasino" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="motorno" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="tcno" type="s:string" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="getCQuotesResponse">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="getCQuotesResult"
type="tns:xmlResult" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:complexType name="xmlResult">
- <s:sequence>
```

<s:element minOccurs="1" maxOccurs="1" name="success" type="s:boolean" />
<s:element minOccurs="0" maxOccurs="1" name="exception" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="XML" type="s:string" />
<s:element minOccurs="1" maxOccurs="1" name="status" type="s:unsignedByte" />
</s:sequence>
</s:complexType>
<s:element name="AuthHeader" type="tns:AuthHeader" />
- <s:complexType name="AuthHeader">
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="memInfo_Xml" type="s:string" />
</s:sequence>
<s:anyAttribute />
</s:complexType>
- <s:element name="getCQuoteDetail">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="teklifno" type="s:int" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="getCQuoteDetailResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="getCQuoteDetailResult"
type="tns:xmlResult" />
</s:sequence>
</s:complexType>
</s:element>
</s:schema>
</wsdl:types>
- <wsdl:message name="TestToServiceIsUPSoapIn">
<wsdl:part name="parameters" element="tns:TestToServiceIsUP" />
</wsdl:message>
- <wsdl:message name="TestToServiceIsUPSoapOut">
<wsdl:part name="parameters" element="tns:TestToServiceIsUPResponse" />
</wsdl:message>
- <wsdl:message name="getCQuotesSoapIn">
<wsdl:part name="parameters" element="tns:getCQuotes" />
</wsdl:message>
- <wsdl:message name="getCQuotesSoapOut">
<wsdl:part name="parameters" element="tns:getCQuotesResponse" />
</wsdl:message>
- <wsdl:message name="getCQuotesAuthHeader">
<wsdl:part name="AuthHeader" element="tns:AuthHeader" />
</wsdl:message>
- <wsdl:message name="getCQuoteDetailSoapIn">
<wsdl:part name="parameters" element="tns:getCQuoteDetail" />

</wsdl:message>
- <wsdl:message name="getCQuoteDetailSoapOut">
<wsdl:part name="parameters" element="tns:getCQuoteDetailResponse" />
</wsdl:message>
- <wsdl:message name="getCQuoteDetailAuthHeader">
<wsdl:part name="AuthHeader" element="tns:AuthHeader" />
</wsdl:message>
- <wsdl:portType name="CInsuranceSOAPSoap">
- <wsdl:operation name="TestToServiceIsUP">
<wsdl:input message="tns:TestToServiceIsUPSoapIn" />
<wsdl:output message="tns:TestToServiceIsUPSoapOut" />
</wsdl:operation>
- <wsdl:operation name="getCQuotes">
<wsdl:input message="tns:getCQuotesSoapIn" />
<wsdl:output message="tns:getCQuotesSoapOut" />
</wsdl:operation>
- <wsdl:operation name="getCQuoteDetail">
<wsdl:input message="tns:getCQuoteDetailSoapIn" />
<wsdl:output message="tns:getCQuoteDetailSoapOut" />
</wsdl:operation>
</wsdl:portType>
- <wsdl:binding name="CInsuranceSOAPSoap" type="tns:CInsuranceSOAPSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="TestToServiceIsUP">
<soap:operation soapAction="http://tempuri.org/TestToServiceIsUP" style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="getCQuotes">
<soap:operation soapAction="http://tempuri.org/getCQuotes" style="document" />
- <wsdl:input>
<soap:body use="literal" />
<soap:header message="tns:getCQuotesAuthHeader" part="AuthHeader" use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="getCQuoteDetail">
<soap:operation soapAction="http://tempuri.org/getCQuoteDetail" style="document" />
- <wsdl:input>
<soap:body use="literal" />

<soap:header message="tns:getCQuoteDetailAuthHeader" part="AuthHeader" use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:binding name="CInsuranceSOAPSoap12" type="tns:CInsuranceSOAPSoap">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="TestToServiceIsUP">
<soap12:operation soapAction="http://tempuri.org/TestToServiceIsUP" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="getCQuotes">
<soap12:operation soapAction="http://tempuri.org/getCQuotes" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
<soap12:header message="tns:getCQuotesAuthHeader" part="AuthHeader" use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="getCQuoteDetail">
<soap12:operation soapAction="http://tempuri.org/getCQuoteDetail" style="document" />
- <wsdl:input>
<soap12:body use="literal" />
<soap12:header message="tns:getCQuoteDetailAuthHeader" part="AuthHeader" use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="CInsuranceSOAP">
- <wsdl:port name="CInsuranceSOAPSoap" binding="tns:CInsuranceSOAPSoap">
<soap:address location="http://localhost:1872/C_InsuranceWS/CInsuranceSOAP.asmx" />
</wsdl:port>
- <wsdl:port name="CInsuranceSOAPSoap12" binding="tns:CInsuranceSOAPSoap12">

<soap12:address location="http://localhost:1872/C_InsuranceWS/CInsuranceSOAP.asmx"
/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

ÖZGEÇMİŞ

Hülya ÇELİK, 1975 yılında İstanbul'da dünyaya geldi. Marmara Üniversitesi Teknik Eğitim Fakültesi, Bilgisayar-Kontrol Öğretmenliği programından mezun oldu. İş hayatına 1998 yılında Bilginç Bilgisayar ve Koç-İdea'da eğitmen olarak başladı. Daha sonra sırasıyla Yapı Merkezi İnşaat Sanayii ve Uzel Holding'de Bilgi Teknolojileri – Yazılım ve Sistem Geliştirme bölümlerinde, uzman ve proje yöneticisi olarak görevler aldı.

Halen Haliç Üniversitesi bünyesinde, web ve yazılım konuları üzerinde çalışmalarını sürdürmektedir.