

**WEB UYGULAMALARINDA
WEB SERVİSLERİNDEN FAYDALANILARAK
ÇOKLU KULLANICI YÖNETİM SİSTEMİ**

Hale Nur BAŞ
Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Eylül-2007

JÜRİ VE ENSTİTÜ ONAYI

Hale Nur Baş'ın “Web Uygulamalarında Web Servislerinden Faydalanılarak Çoklu Kullanıcı Yönetim Sistemi” başlıklı Bilgisayar Mühendisliği Anabilim Dalındaki, Yüksek Lisans Tezi 24.07.2007 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

| | Adı-Soyadı | İmza |
|----------------------------|---------------------------------------|-------------|
| Üye (Tez Danışmanı) | : Prof. Dr. ALİ GÜNEŞ | |
| Üye | : Yard. Doç. Dr. HÜSEYİN POLAT | |
| Üye | : Yard. Doç. Dr. NİHAT ADAR | |

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
..... tarih ve sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

WEB UYGULAMALARINDA WEB SERVİSLERİNDEN FAYDALANILARAK ÇOKLU KULLANICI YÖNETİM SİSTEMİ

Hale Nur BAŞ

**Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman: Prof. Dr. Ali GÜNEŞ
2007, 54 sayfa**

Hızlı gelişen teknoloji, yaygınlaşan web uygulamaları ve kullanıcılarla interaktif iletişim kimlik yönetimi kavramını ortaya çıkartmıştır. Web uygulamalarında interaktif iletişim sonucunda kullanıcı tanımlama ve yetkilendirme için değişik yöntemler kullanılmaktadır.

Bir organizasyon içinde geliştirilen çeşitli web uygulamaları için kullanıcı tanımlama ve yetkilendirmede, her uygulama için ayrı ayrı kullanıcı tabloları, kullanıcılarla ilişkili roller tanımlanabilir ve uygulama içinde kod bazlı takip yapılabilmektedir. Fakat böyle bir mevcut yapı içinde kullanıcılar organizasyon yapısı içinde olduğundan her uygulamanın kendi kullanıcılarını yönetmesi, hem veri tekrarlanmasına, hem zaman kaybına, hem de organizasyon içinde bilgileri değişen kullanıcının takibini yaparak güncel verilerinin tutulmasında zorluklar çıkartmaktadır.

Bu tezde, web servislerinden faydalanarak merkezi bir kullanıcı yönetim uygulaması tasarlanarak geliştirilmiş ve organizasyon içindeki kullanıcı bilgileri mevcut LDAP(Lightweight Directory Access Protocol) sunucusundan alınarak veri tekrarı önlenmiş ve kullanıcı verisinin güncelliği korunmuştur.

Anahtar Kelimeler: Web Servisleri, Kullanıcı Tanımlama, Kullanıcı Yetkilendirme, Kimlik Yönetimi

ABSTRACT**Master of Science Thesis****MULTI USER MANAGEMENT SYSTEM
IN WEB APPLICATIONS
USING WEB SERVICES****Hale Nur BAŞ****Anadolu University
Graduate School of Sciences
Computer Engineering Program****Supervisor: Prof. Dr. Ali GÜNEŞ
2007, 54 pages**

Rapid developing technology, widespread web applications and the interactive communication with users introduced the concept of the identity management. As a result of the interactive communication of the web applications, different ways are used for user authentication and authorization.

For the many developed web applications in an organization, in the authentication and authorization, for each applications different user tables, and roles related with users can be defined and code based following in the application can be done. But in such an available structure cause the users are in the organization structure, managing of each applications users, causes both the repetition of data, loss of time, and difficulties of the handling current data of the changing user data in the organization.

In this thesis, a central user management application using web services is designed, developed and user information in the organization is taken from the LDAP(Lightweight Directory Access Protocol) server, so repetition of the data is avoided and current user data is saved.

Keywords: Web Services, Authorization, Authentication, Identity Management

İÇİNDEKİLER

| | <u>Sayfa</u> |
|--|--------------|
| ÖZET | i |
| ABSTRACT | ii |
| İÇİNDEKİLER..... | iii |
| ŞEKİLLER DİZİNİ..... | v |
| | |
| 1. GİRİŞ..... | 1 |
| 2. WEB SERVİSLERİ | 3 |
| 3. WEB SERVİSLERİ STANDARTLARI | 8 |
| 3.1 XML'in Tanımı..... | 10 |
| 3.1.1 XML'in önemi | 11 |
| 3.1.2 XML teknolojileri | 12 |
| 3.2 WSDL | 13 |
| 3.3 SOAP | 16 |
| 3.4 UDDI..... | 19 |
| | |
| 4. KULLANICI TANIMLAMA VE YETKİLENDİRME..... | 21 |
| 4.1 Strateji Tasarımı | 22 |
| 4.1.1 Kaynakların belirlenmesi | 22 |
| 4.1.2 Kullanıcı tanımlama stratejisi seçimi | 22 |
| 4.1.3 Kaynaklara erişim için kimlik seçimi | 23 |
| 4.1.4 Kimlik akışı belirleme..... | 24 |
| 4.1.5 Yetkilendirme yaklaşımı seçimi..... | 24 |
| 4.1.6 Kimlik akışının izleyeceği yol belirleme | 24 |
| 4.2 Kullanıcı Tanımlama Yaklaşımları | 25 |
| 4.2.1 Rol bazlı | 25 |
| 4.2.2 Kaynak bazlı | 25 |

| | |
|--|-----------|
| 5. ÇOKLU KULLANICI YÖNETİM SİSTEMİ | 27 |
| 5.1 Sistem Mimarisi | 30 |
| 5.2 Veritabanı Yapısı | 31 |
| 5.3 Veritabanında Tablolar Arası İlişkiler | 36 |
| 5.4 Veritabanında Kullanılan Saklı Yordamlar ve İşlevleri..... | 36 |
| 6. KULLANICI YÖNETİM SİSTEMİ ARAYÜZLERİ..... | 42 |
| 6.1 Kullanıcı Modülü | 43 |
| 6.1.1 Kullanıcı ekleme | 43 |
| 6.1.2 Kullanıcı bilgisi değiştirme | 44 |
| 6.1.3 LDAP kullanıcılarının uygulamalara atanması..... | 45 |
| 6.1.4 Web kullanıcılarının uygulamalara atanması..... | 46 |
| 6.1.5 Kullanıcıların listelenmesi | 46 |
| 6.2 Uygulama Modülü | 47 |
| 6.2.1 Uygulamaların eklenmesi..... | 47 |
| 6.2.2 Uygulama bilgilerinin değiştirilmesi | 48 |
| 6.3 Rol Modülü | 49 |
| 6.3.1 Rol ekleme | 49 |
| 6.3.2 Rol bilgilerinin değiştirilmesi | 50 |
| 6.4 Yardım | 51 |
| 7. SONUÇLAR VE ÖNERİLER..... | 52 |
| KAYNAKLAR | 53 |

ŞEKİLLER DİZİNİ

| | |
|--|----|
| Şekil 2.1. Web servisi modeli [4] | 5 |
| Şekil 2.2. Web servisi istemci ve sağlayıcısı arasındaki temel işlemler [6]..... | 7 |
| Şekil 3.1. Web servisi katmanları [7] | 9 |
| Şekil 3.2. WSDL doküman tipi [9]..... | 13 |
| Şekil 3.3. WSDL elemanları [11] | 14 |
| Şekil 3.4. WSDL'in üç temel elemanı ve yedi bölümü [1]..... | 15 |
| Şekil 3.5. WSDL dokümanının web servislerindeki kullanımı..... | 15 |
| Şekil 3.6. Farklı ortamların iletişimini sağlayan SOAP mesajları | 16 |
| Şekil 3.7. SOAP istemci ve SOAP sunucusu arasındaki iletişim [17]..... | 17 |
| Şekil 3.8. Bir SOAP mesajının yapısı [16]..... | 18 |
| Şekil 3.9. Bir SOAP istemci istek (request) mesajı..... | 18 |
| Şekil 3.10. Bir SOAP yanıt (response) mesajı | 19 |
| Şekil 3.11. UDDI veri tipleri [9] | 19 |
| Şekil 3.12. UDDI kurum kayıt sunucuları [4]..... | 20 |
| Şekil 4.1. .Net Web uygulamalarında güvenlik [19]..... | 21 |
| Şekil 4.2. Güvenilir alt sistem modeli [19] | 26 |
| Şekil 4.3. Impersonation/delegasyon modeli [19]..... | 26 |
| Şekil 5.1. Web uygulamaları ve kullanıcı tabloları | 27 |
| Şekil 5.2. Web uygulamaları, kullanıcı tabloları ve LDAP sunucusu..... | 28 |
| Şekil 5.3. Çoklu kullanıcı yönetim sistemi | 29 |
| Şekil 5.4. Çoklu kullanıcı yönetim sistemi teknik özellikler | 30 |
| Şekil 5.5. Veritabanı İlişkiler Diyagramı | 36 |
| Şekil 6.1. Kullanıcı Giriş Ekranı | 42 |
| Şekil 6.2. Kullanıcı Yönetimi Menüsü..... | 42 |
| Şekil 6.3. Kullanıcılar Menü İçeriği..... | 43 |
| Şekil 6.4. Kullanıcı Ekleme..... | 43 |
| Şekil 6.5. Kullanıcı Görüntüleme..... | 44 |
| Şekil 6.6. Kullanıcı Güncelleme..... | 44 |
| Şekil 6.7. Kullanıcı Bilgisi Değiştirme | 45 |
| Şekil 6.8. LDAP Kullanıcılarının Uygulamalara Atanması..... | 45 |

| | |
|---|----|
| Şekil 6.9. Web Kullanıcılarının Uygulamalara Atanması | 46 |
| Şekil 6.10. Web Kullanıcılarının Listelenmesi | 46 |
| Şekil 6.11. Uygulamalar Menü İçeriği | 47 |
| Şekil 6.12. Uygulama Ekleme | 47 |
| Şekil 6.13. Uygulama Bilgileri Görüntüleme..... | 47 |
| Şekil 6.14. Uygulama Bilgilerini Güncelleme | 48 |
| Şekil 6.15. Uygulama Bilgilerinin Değiştirilmesi | 48 |
| Şekil 6.16. Roller Menü İçeriği | 49 |
| Şekil 6.17. Rol Ekleme..... | 49 |
| Şekil 6.18. Rol Bilgisi Seçme..... | 50 |
| Şekil 6.19. Rol Bilgisi Güncelleme | 50 |
| Şekil 6.20. Yardım Menüsü..... | 51 |

1. GİRİŞ

Web uygulamalarının yaygınlaşması ve statik içerikten dinamik içeriğe geçiş sürecinde, kullanıcı kimliği ve yönetimi, uygulama bazında önem kazanmıştır. Web uygulamalarının çoğu, güvenlik nedeni ile kullanıcı kimliğini doğrulamaya ve kullanıcıların belirli güvenlik katmanlarını geçebilmeleri için kimlik denetimine ihtiyaç duymaktadır.

ASP.Net uygulamalarında güvenliği sağlamak için kullanıcıların kaynaklara nasıl erişeceğini belirlememiz gerekmektedir. Bunun için kullanıcı tanımlama ve sonraki adım olarak yetkilendirme süreçlerini ele almak gerekmektedir.

Kullanıcı Tanımlama; kullanıcının kimliğinin tespit edilme sürecidir. Kullanıcının kimliği LDAP sunucularından, veritabanından, bir XML(Genişletilebilir Etiketleme Dili) dosyasından hatta bir web servisinden bile takip edilebilir. Microsoft Passport, kullanıcı tanımlamasında kullanılan web servislerine güzel bir örnektir. .Net Framework ile birlikte kullanılmaya başlanan kullanıcı tanımlama mekanizması tasarlanmıştır. Asp.Net, kullanıcı tanımlamada kullanılan Windows, Cookie ve Passport güvenlik modlarını standart olarak desteklemektedir.

Kullanıcı Yetkilendirme; bir istekte bulunan kullanıcının talep ettiği kaynağa erişim hakkı olup olmadığının tespit edilmesi işlemidir. Bu işlem genelde, önceden tespit edilen kullanıcı rolleriyle sağlanır. Her uygulamanın tüm kaynaklarına erişebilen ve yönetim gibi üst seviye işlemleri gerçekleştiren bir 'Yönetici'si ve kısıtlı erişim hakkına sahip 'Kullanıcı'ları mevcuttur.

Bir organizasyon yapısı içinde iş süreçlerinin dijitalizasyonu sonucu, kullanıcılarla interaktif bir şekilde etkileşen web uygulamalarının, kullanıcıları yönetmek için çeşitli yollar izlemesi gerekmektedir.

- Veritabanında her uygulamanın kendi kullanıcı tabloları ve ilgili rolleri tutulabilir ve kod bazlı müdahale ile uygulama içinde kontrol edilebilir.

- Uygulamalar, kullanıcılarını veritabanında veya XML verisi içinde tutabilir, ilgili kimlik bilgilerini mevcut LDAP sunucusundan alabilir.

Organizasyon içindeki uygulamaların ve kullanıcıların sayısı arttıkça web uygulamalarının kullanıcı yönetim süreci gittikçe karmaşık bir yapı içine girerek standarttan uzaklaşacaktır. Ayrıca, kullanıcı bilgileri değiştikçe, güncel veriyi, uygulamaya ait veritabanında değiştirmek gerekecek, bu da hem zaman hem de veri tekrarlanması gibi olumsuz sonuçlara sebep olacaktır.

Ayrıca organizasyonun dışa açılan web uygulamaları (extranet veya internet) için istemci kimliklerini doğrulama ve yetkilendirme işlemlerini gerçekleştirmesi gerekmektedir.

Kullanıcı bilgilerinin, merkezi bir yerden kontrol edilmesi, sistemdeki web uygulamaları ile ilişkilendirilip gerekli yetkilendirmelerin yapılması yukarıda bahsetmiş olduğumuz veri tekrarları, zaman kaybı ve güncel veriyi tutmadaki sıkıntıların giderilmesine olanak tanımaktadır. Web servisleri, bilginin akışını kolaylaştırdığı ve erişiminin kolaylığı sebebiyle kullanıcı kimlik yönetim uygulamasında kullanılmakta ve LDAP (active directory) sunucusu ile bağlantı kurarak mevcut kullanıcılar için; SQL Server ile bağlantı kurarak web kullanıcıları için kimlik doğrulama, yetkilendirme ve kullanıcı bilgisinin yönetimini kolaylaştırmaktadır.

2. WEB SERVİSLERİ

Web sayesinde şirketler kendi içlerinde, diğer şirketlerle ve müşterileri ile aralarında bilgiye kolayca erişebilme imkanı bulmuştur. Fakat bu durum şirketlerin ihtiyaçlarını tam olarak karşılamaya yetmemektedir. Geliştirilen Web siteleri, uygulamalar, genelde, kullanıcılara sadece bilgiyi göstermekte, hatta bazen sadece broşür gibi kullanılmaktadır. Yani etkileşim, bilgisayar ile kullanıcı arasında bilgiyi göstermek şeklinde tek taraflı olmaktadır. Bilgisayar ile bilgisayar veya iki program arasında veri ve bilgi alışverişinin olmayışı şirketlerin Web'den daha fazla yararlanmalarına engel olmaktadır.

Bilgiye daha hızlı ve zahmetsizce ulaşma amacı, yazılan programların, geliştirilen protokollerin, hazırlanan web sitelerin kısaca bilgi çağıının ana temasıdır. Ayrıca kullanmakta olduğumuz gelişmiş uygulamaların ortak noktası farklı bilgisayarların, farklı ağların ve farklı programların aralarında bilgi alışverişi sırasında yaşadığı engelleri aşmaya çalışmasıdır. İşte XML tabanlı Web servisleri bu noktada devreye girmiştir. XML'in çıkışıyla birlikte bu farklı ortamların birbiriyle haberleşmesi olanaklı kılınmıştır. Artık uygulamaların üzerinde çalıştıkları işletim sistemi, kullandıkları veritabanı, yazılmış oldukları programlama dili farketmeksizin bir XML dosyasını okuma ve anlamlandırma becerisine sahip her ortam birbiriyle rahatlıkla iletişim kurabilir hale gelmiştir. 1998 senesinde Sun ve IBM'in ortak çalışmaları sonucu ortaya çıkardıkları Web Servisleri, farklı ortamların birbiriyle zahmetsizce haberleşmesi konusunda önemli bir adım olmuştur.

Web Servisleri, programları, nesnelere, veritabanlarını veya karmaşık iş fonksiyonlarını birbirine bağlayan Genişletilebilir Etiketleme Dili (XML: eXtensible Markup Language) uygulamalarıdır [1]. Bir program mesaj formunda yaratılmış bir XML dokümanı kullanarak, ağ üzerinden Web servisine bir istek gönderir, ve opsiyonel olarak XML dokümanı formunda bir cevap alır. Web servis standartları mesajın formatını, mesajın gönderileceği arayüzü, mesajı kullanan programların nasıl anlaşabileceğini ve Web servisleri arayüzlerinin nasıl keşfedileceği ve yayınlanacağını önceden tanımlar.

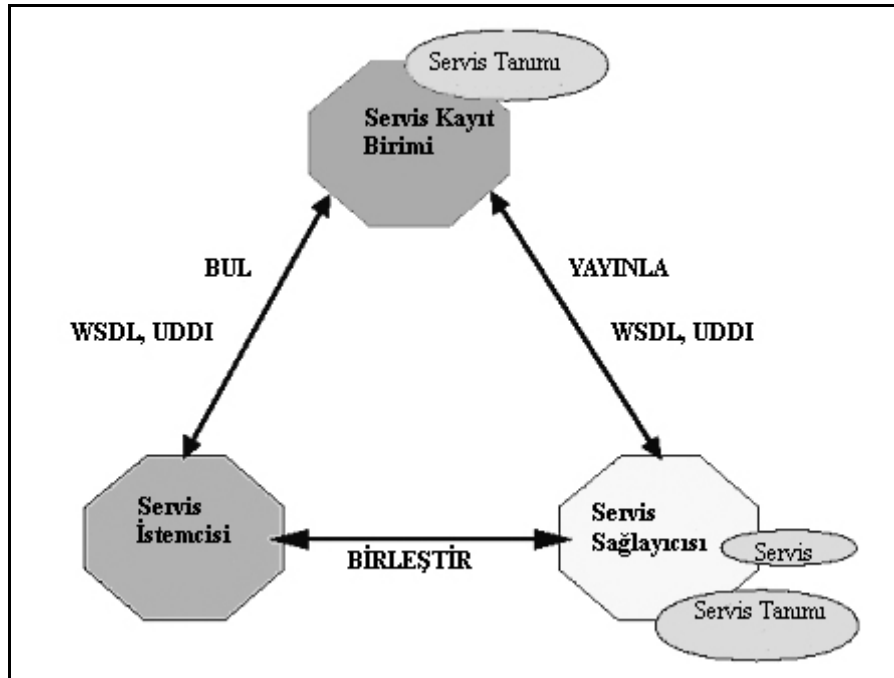
Web servisleri bir çok yazılım firması tarafından yoğun bir destek bulan bir modeldir. Web servisleri açık internet standartlarına dayanır. Bu standartlar henüz gelişme ve olgunlaşma aşamasında olan bu modelle ilgili olarak bu aşamada ortaya çıkan ve kullanılan çekirdek standartlar Basit Nesne Erişim Protokolü (SOAP: Simple Object Access Protocol), Web Servisi Tanımlama Dili (WSDL: Web Services Description Language) ve Evrensel Açıklama, Keşif ve Entegrasyon (UDDI: Universal Description, Discovery and Integration)'dur.

İnternet üzerinde uygulama bütünleşmesi için yöntem arayan bir çok yazılım geliştirici firma ve kurum web servisleri modeli konusuna yönelmişlerdir. Web servisleri modelini destekleyen Microsoft, IBM, Sun, HP, Oracle ve daha bir çok firma bu konuda yoğun bir şekilde çalışmakta ve web servisleri yazılım ve uygulama geliştirme araçlarını geliştiricilere sunmaktadırlar. Bu konudaki firmaların yoğun desteğinden dolayı uygulama bütünleşmesi konusunda ortaya çıkacak hakim ortamın web servisleri modeline dayanacağı yönündedir [2].

Aynı veya farklı amaca hizmet eden birden fazla firmanın ortak bir çatı altında toplanarak daha verimli çalışmalarının sağlanması gereksiniminden dolayı web servislerine ihtiyaç duyulmaktadır. Web servisleri geleceğin uygulama entegrasyonları ve dağınık mimarileri için büyük ümitler vaat eden ve hızla gelişen bir dizi standart ve uygulama teknolojileridir. XML Web Servisleri yazılan program kodunu çalıştıran bileşenlerdir. Sıradan bileşenlerden farkı bu işlevi farklı bilgisayarlara da sunmasıdır. XML Web Servislerine farklı bir bilgisayar ve farklı bir platformdan istemci olunabilir. Yani XML Web Servisleri farklı platformlar arasında ortak olarak kullanılabilen bileşenlerdir. XML Web Servisleri ile farklı platformlar arasında veri aktarımı da belli standartlar sayesinde oldukça kolaylaşmaktadır.

Web servisleri modeli üç ana birimin etkileşimine dayanır [3]. Şekil 2.1.'de gösterilen bu birimler şunlardır:

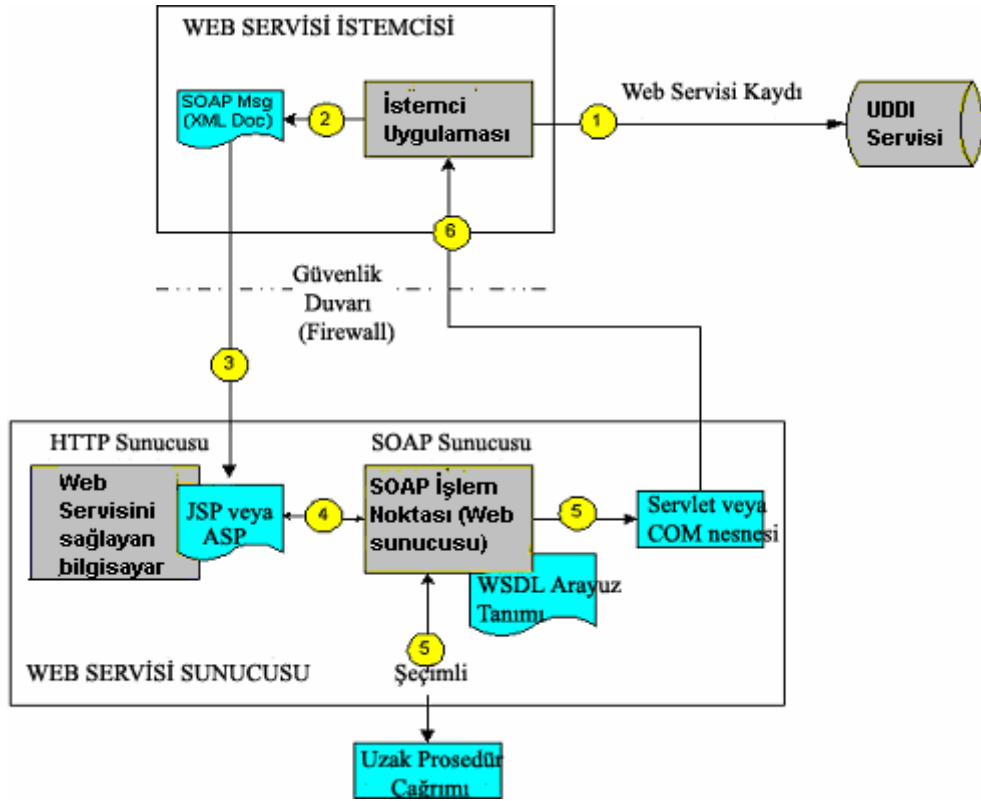
- **Servis Sağlayıcı (Service Provider):** Servis sağlayıcı istemcilerin sağlayıcıda bulunan servislere erişimini sağlar. Servis sağlayıcı kendi sitesinde bulunan web servisleri tanımını servis kayıt birimine (service registry) kaydederek bu servisinin nasıl çağrılacağını belirtir.
- **Servis İstemcisi (Service Requester):** Servis sağlayıcısında bulunan web servislerini çağırarak kullanan istemci uygulamalarıdır. Web servisinin nasıl çağrılacağını belirler ve ilgili parametreleri servis kayıt biriminden arayarak bulunup çağrılmasını sağlar.
- **Servis Kayıt Birimi (Service Registry):** 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 çağrılacağı konusunda tanım bilgileri içerir.



Şekil 2.1. Web servisi modeli [4]

Şekil 2.2.'deki şema bir web servisi istemcisinin bir servis sağlayıcıdan bir servisi çağırma aşamasındaki temel adımları göstermektedir [5]:

1. Web servisi istemcisi (SOAP Client) servis kayıt biriminden (UDDI) web servisini bulur.
2. İstemci bir SOAP mesajı hazırlar. SOAP mesajı bir XML belgesidir.
3. İstemci SOAP mesajını web sunucusu veya uygulama sunucusunda çalışan SOAP istek dinleyicisine gönderir. İstek dinleyici gelen isteklere cevap veren sunucu programlarıdır. Bu programlar bir Java Server Sayfaları (JSP: Java Server Pages), Aktif Server Sayfaları (ASP: Active Server Pages), Ortak Geçit Arayüzü (CGI: Common Gateway Interface) veya İnternet Sunucusu Uygulama Programı Arayüzü (ISAPI: İnternet Server Application Program Interface) programıdır.
4. SOAP sunucusu gelen SOAP mesajını çözümler (parse eder) ve gerekli parametreleri göndererek istenen nesnenin istenen yöntemini çağırır.
5. Çağrılan nesnedeki yöntem çalışır ve sonuçları SOAP sunucusuna gönderir. SOAP sunucusu gelen sonucu SOAP mesajı formatında biçimlendirerek istemciye gönderir.
6. İstemci gelen SOAP mesajının içindeki bilgileri alarak istekte bulunan programa gönderir.



Şekil 2.2. Web servisi istemci ve sağlayıcısı arasındaki temel işlemler [6]

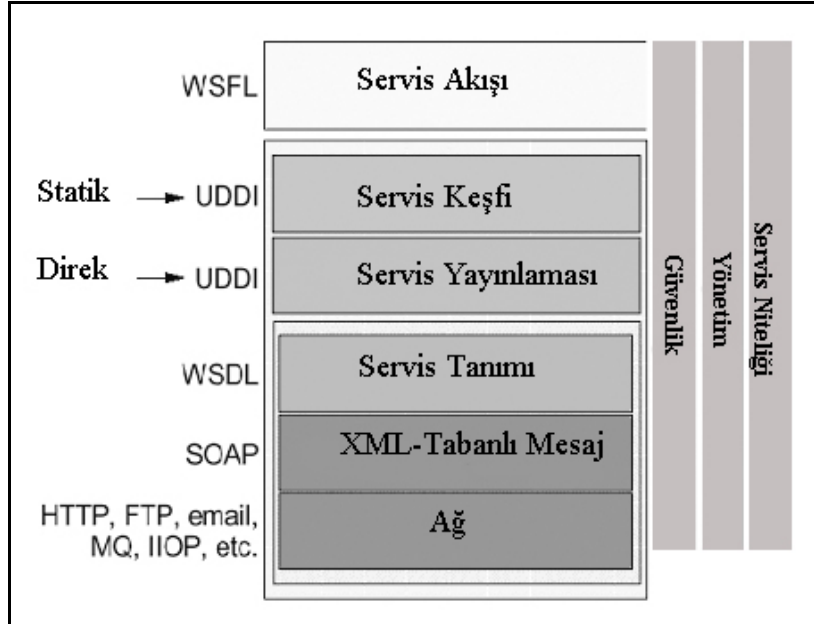
3. WEB SERVİSLERİ STANDARTLARI

Web Servisleri, XML tabanlı deęişik teknolojileri kullanır [1].

- **XML(Extensible Markup Language - Geniřletilebilir İřaretleme Dili)**, Web servislerinin temelini oluřturan, verinin tanımını ve nasıl iřleneceęini tanımlayan dili saęlayan temel yapıdır. Yapılandırılmıř belge ve verilerin evrensel formatıdır. XML sayesinde her birey ve kuruluř gerektięinde kendi veri ve belgelerini belirtmek iin kendi saklama ve aktarma biimlerini belirlemek zorunda kalmaz. Farklı taraflar ortak bir formatta anlařarak kolayca iletiřim kurabilirler. Worl Wide Web Consortium(W3C) tarafından standartlar belirlenir ve yayınlanır.
- **WSDL(Web Services Description Language- Web Servisi Tanımlama Dili)**, XML tabanlı bir teknolojidir. Web servis arayüzünü, veriyi ve mesaj tipini, etkileřim paternlerini ve protokol eřleřtirmelerini tanımlar.
- **SOAP(Simple Object Access Protocol - Basit Nesne Eriřim Protokolü)**, bir makineden bařka bir makinedeki nesnenin bir metodunu aęırmak ve sonucunu almak iin üretilmiř bir protokoldür.
- **UDDI(Universal Description Discovery Integration - Evrensel Açıklama, Keřif ve Entegrasyon)**, iř bilgisini saklama ve katologlama, Web servis arayüzlerine iřaretiler ile eriřerek bilgi alma iřlevi olan Web servis kayıtısı ve keřif mekanizmasıdır. Firmaların saęladıęı servislerin belirlenmesini ve eriřilebilmesini saęlayan sistemdir. Online olarak belli bir servisi veya iři arayan insanlar UDDI kullanarak aradıklarına eriřebilirler.

Mesaj alış verişi için XML standardını kullanır. XML'in bir standart olması ve herhangi bir platforma bağlı olmaması sayesinde XML Web Servisleri farklı platformların konuşmasını ve veri alışverişini sağlar. Bu sayede uygulama geliştirme aşamasında farklı sistemleri kullanan uygulamalar yazarken bu standartlar kullanılarak işlemler daha kolay bir şekilde yapılabilir.

Web servisleri açık internet standartlarına dayanır. Şekil 3.1. web servisi mimarisindeki temel katmanları göstermektedir. Bu katmanlarda belirtilen güvenlik, iş akışı, servis kalitesi ve yönetim gibi konulardaki web servisi standartları henüz araştırma aşamasındadır.



Şekil 3.1. Web servisi katmanları [7]

3.1 XML'in Tanımı

Genişletilebilir İşaretleme Dili (XML; eXtensible Markup Language), hem insanlar hem bilgi işlem sistemleri tarafından kolayca okunabilecek dökümanlar oluşturmaya yarayan, W3C tarafından tanımlanmış bir standarttır. Bu özelliği ile veri saklamanın yanında farklı sistemler arasında veri alışverişi yapmaya yarayan bir ara format görevi de görür. SGML'in (Standard Generalized Markup Language) basitleştirilmiş bir alt kümesidir.

XML'in mucidi ve geliştiricisi, HTML'i (Hyper Text Markup Language) de icad etmiş olan Tim Berners Lee'dir. HTML dilinde verilerin yeterince tanınmaması, daha çok görsellik ile ilgili olmasından dolayı doğmuştur. Dilin düzenlenmesi de W3C'nin sorumluluğundadır.

Günümüzde birçok yazılım, diğer yazılımlarla veri alışverişinin XML formatı üzerinden yapmaktadır. Ayrıca XML'i esas format olarak kullanan uygulamalara, veri tabanlarına rastlamak mümkündür.

Microsoft'un geliştirdiği .NET teknolojisinde kullanılan DataSet nesneleri XML formatındadır. Ayrıca XML, ofis uygulamalarının alt yapısı haline getirilmiştir.

İçeriğin, doküman yapısının ve şeklin birbirinden ayrı ele alınması XML'i içerik yönetim sistemlerinin ideal formatı haline getirmiştir.

XML dokümanları ağaç veri yapısında olurlar. Bağımsız imler yapıyı oluştururken, içerik ya imin özelliği olarak ya da iki im arasında gösterilir. Yapıyla ilgili ayrıntılar DTD (Document Type Definition) ya da XML Schema adı verilen harici dokümanlar ile tanımlanır. Aşağıdaki örnek bir XML dökümanında verinin nasıl belirtildiğini göstermektedir[8].

```
<kullanici>
  <kullanici id="1342">
    <ad>Hale</ad>
    <soyad>Baş</soyad>
  </kullanici>
<kullanici id="1343">
```

```

    <ad>Cengiz</ad>
    <soyad>Demre</soyad>
  </kullanici>
  <kullanici id="1344">
    <ad>Emine</ad>
    <soyad>Fethi</soyad>
  </kullanici>
  <kullanici id="1345">
    <ad>Günsel</ad>
    <soyad>Kara</soyad>
  </kullanici>
</kullanıcılar>

```

3.1.1 XML'in önemi

XML geniş bir alanda önemli bir etkiye sahiptir. Çeşitli organizasyonların ve kişilerin XML adaptasyonundan bu derece etkilenmelerinin sebepleri olarak aşağıdaki maddeleri listeleyebiliriz [17].

- XML dosyaları çok rahat okunabilmektedir. Metin halinde tasarlandığından en kötü durumlarda bile içerik okunarak anlaşılabilir. İkili (binary) formatında olmadığı için standart metin editörlerinden görsel (visual) geliştirme ortamına kadar herhangi bir araçla oluşturulup, geliştirilebilir.
- XML oldukça geniş bir alanda destek gördüğünden, Web browserlarında, veritabanlarında, işletim sistemleriyle çok fazla araç ve geliştirme ortamlarınca desteklendiğinden küçük ve orta ölçekli şirketler tarafından kolay ve ucuz bir şekilde XML formatı şeklinde veri alışverişi yapılmaktadır.
- SQL Server 2005 gibi önemli ilişkisel veritabanları XML verisi saklama, okuma ve oluşturma gibi özelliklere sahiptir.
- Web sayfalarının görüntülenmesi ve rapor oluşturmada çeşitli XML yorumlayıcıları ve dönüştürücülerine XML desteği sağlayan çok geniş teknoloji kitlesi vardır.
- Gösterim önemli olduğu için stylesheet standardı Genişletilebilir Stil Dili (XSL: eXtensible Stylesheet Language) verinin nasıl sergileneceğini gösterir. Bunu HTML ile de yapabiliriz. Ancak

veriyi arama programları ile işleyemeyiz. Daha önemlisi XML özgün tarzda olduğu için tamamen farklı stylesheet kullanıp çıktıyı postscript, pdf, text veya henüz keşfedilmemiş bir çok formatta üretebiliriz.

- XML platformdan bağımsızdır. Bu sayede uygulama, dil, işletim sistemi gibi kısıtlamalara bağlı değildir. Şu anda hemen her işletim sistemi ve uygulama XML belgelerini okumak ve yazmak için yerleşik özelliklere sahiptir. Bütün XML belgeleri unicode tabanlıdır. İşletim sistemleri farklı kod sayfalarını (code page) kullandıkları için farklı dillerdeki belgeleri görüntüleyemezler. Ancak unicode kullanımı farklı dillerdeki belgelerin görüntülenmesine olanak tanır.

3.1.2 XML teknolojileri

XML; veri işaretleme dili, değişik içerik modelleri, bağlama modeli, isimuzayı modeli ve farklı iletişim mekanizmalarını barındıran bir yapıdır. Aşağıdakiler web servislerinin temelini oluşturan XML ailesinin önemli elemanlarıdır [1]:

- **XML v1.0:** Soyut veri modeli ve serileştirme formatı sunan temel elemanla ilişkilendirilmiş elemanların, niteliklerin ve tagların tanımlanmasıyla ilgili kurallardan oluşur.
- **XML schema:** Veri tipleri, içerik, yapı ve XML dokümanlarıyla ilişkili izin verilen elemanları tanımlayan XML dokümanlarıdır. Ayrıca doküman elemanlarıyla ilişkili anlamsal işleme komutlarının tanımlanmasında kullanılır.
- **XML namespaces:** XML doküman elemanları ve uygulamalara tekil isimler kazandırır.
- **XML Information Set:** XML doküman parçasının tutarlı ve soyut gösterimidir.

- **Xpointer:** Dokümanın belli bir kısmını gösteren işaretçidir. XPath, XML dokümanını aramada kullanılan ifadelerdir. XLink, bir çok XML dokümanını aramada kullanılır.
- **Extensible Stylesheet Language Transformations (XSLT):** XML dokümanını başka bir XML dokümanına çevirmek için ya da XML olmayan bir formata aktarmak için kullanılır.
- **DOM (Document Object Model) and SAX (Simple API for XML):** Programlama kütüphaneleri ve XML dokümanı ayrıştırıcısıdır. Bir ağaç yapısı yaratır ve XML elemanların hepsini bir bir okuyarak cevap verir.

3.2 WSDL

WSDL, Web Servislerini mesajları işleyen ağ uç noktalar kümesi olarak tanımlayan bir XML dilidir [9]. Bir WSDL servis tanımı, bir dizi işlemler ve mesajlar için soyut bir tanım, bu işlemler ve mesajların bağlanması için tam bir tanım, ve bu bağlanma için ağ uç noktaları için belirtilmiş tanımlardan oluşur.



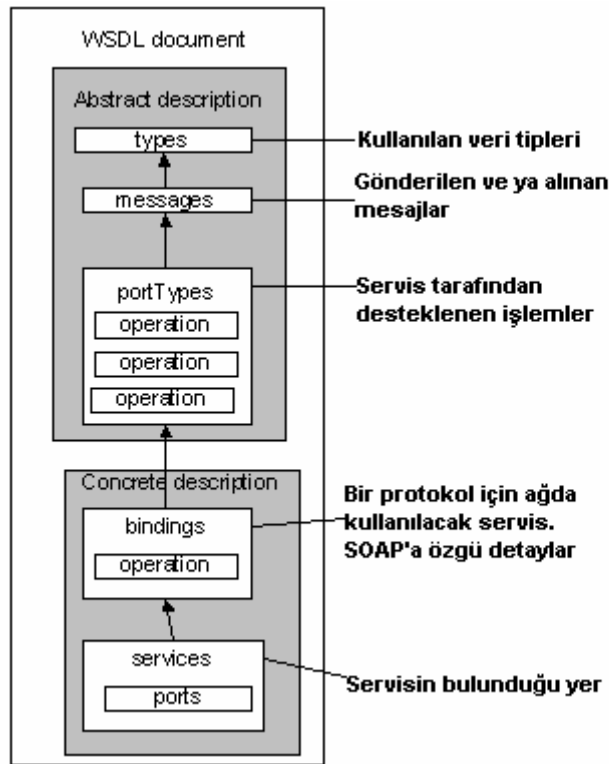
Şekil 3.2. WSDL doküman tipi [9]

WSDL, bir uygulamanın bir web servisini kullanması için web servisinin nasıl çağrılacağını, arayüzünü, protokollerini ve kodlama standartlarını belirtir. WSDL bu standartları XML dili tanımlayarak gerçekleştirir. Bu tanımlama mesaj alışveriş yetisine sahip iletişim uç noktaları kümesi olarak tanımlanan ağ

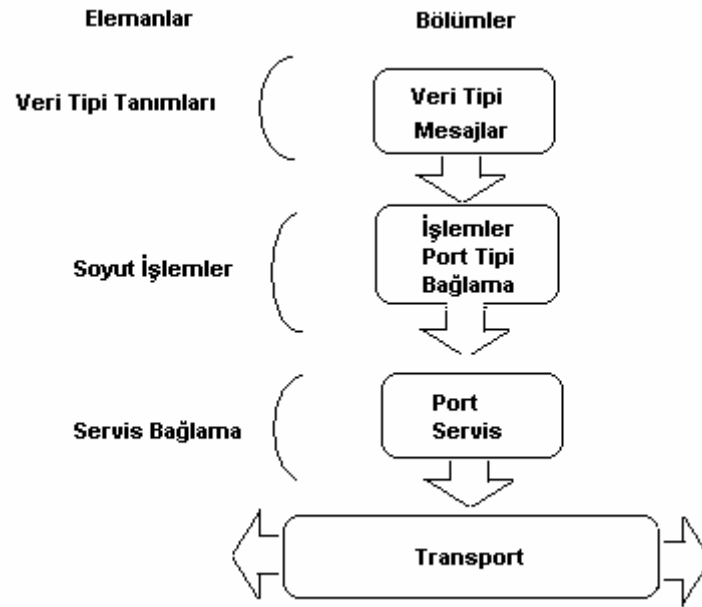
servislerini içerir. Web servisi tanımı işlemler, giren ve çıkan mesaj formatları, ağ ve port adresleri gibi bilgileri tanımlar.

Bir web servisi tanım belgesi ağ servisleri tanımında aşağıdaki elemanları içerir [10]:

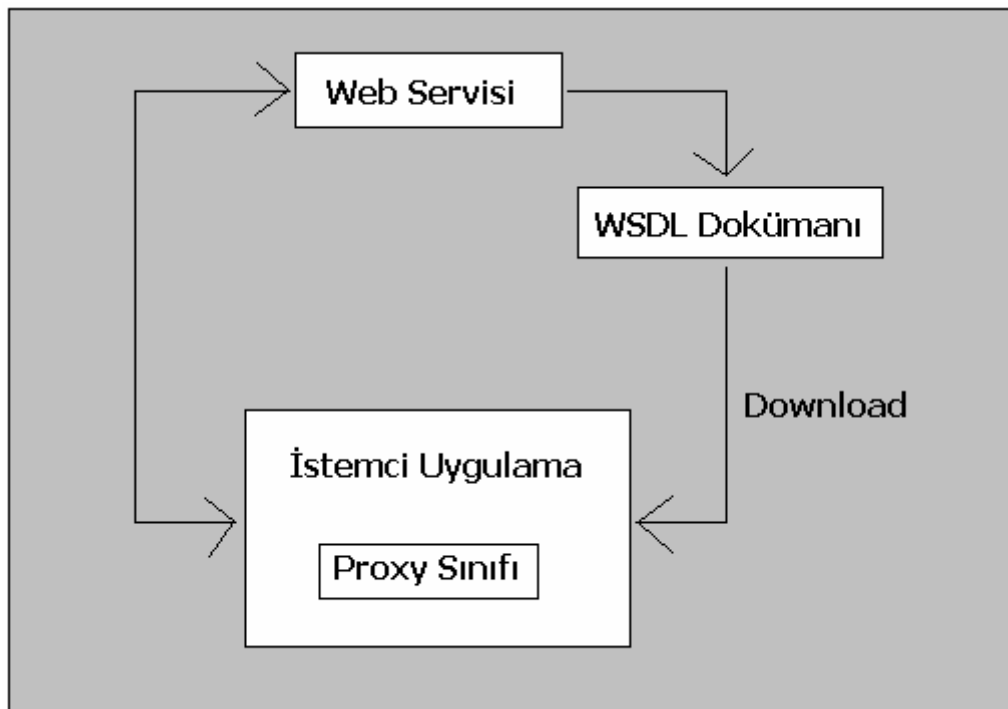
- *Types*: XSD gibi kodlama sistemi kullanarak mesajlarda kullanılacak veri tip tanımlarını içerir.
- *Message*: İletişimde kullanılacak soyut mesajları tanımlar.
- *Operation*: Servis tarafından desteklenen soyut işlem tanımlarıdır.
- *PortType*: Bir veya daha çok uç noktaların desteklediği soyut işlemler kümesidir.
- *Binding*: Belirli bir port tipi için belirlenen protokol ve veri formatı tanımlamalarıdır.
- *Port*: Bağlama ve web adresinden oluşan servis noktasını tanımlar.
- *Service*: İlişkili uç noktalar kümesidir.



Şekil 3.3. WSDL elemanları [11]



Şekil 3.4. WSDL'in üç temel elemanı ve yedi bölümü [1]

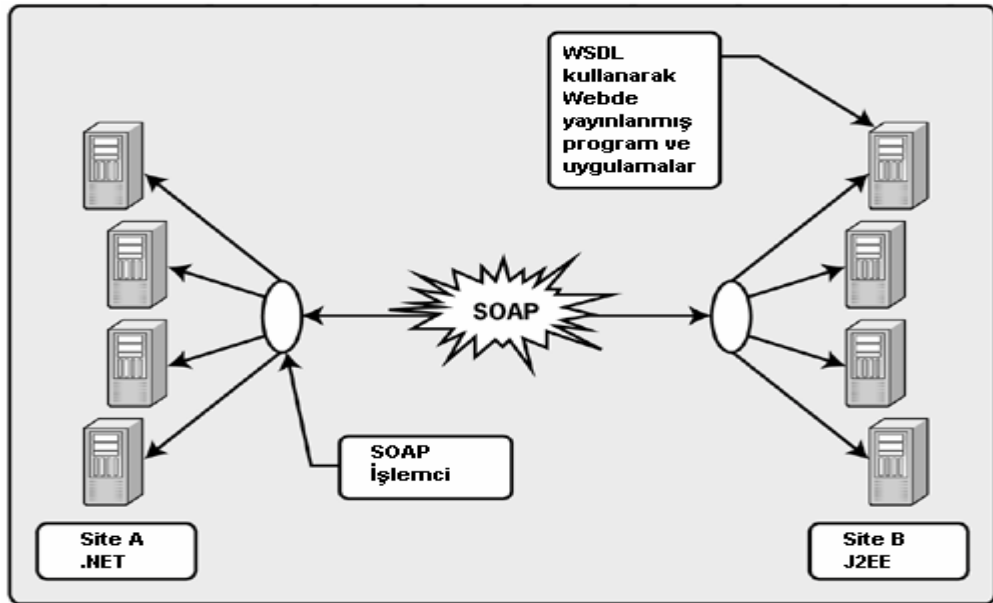


Şekil 3.5. WSDL dokümanının web servislerindeki kullanımı

3.3 SOAP

SOAP, nesnelerin birbirleriyle platform bağımsız iletişimi sağlar [15]. XML verisinin tanımlanması yapılmış ve WSDL mesajının iletişimi ve işlenmesini destekleyen gerekli servis hazır ise, bu mesajın bir bilgisayardan diğerine nasıl bir yolla gönderileceğinin ve karşı tarafta nasıl işleneceğinin tanımlanması gerekmektedir.

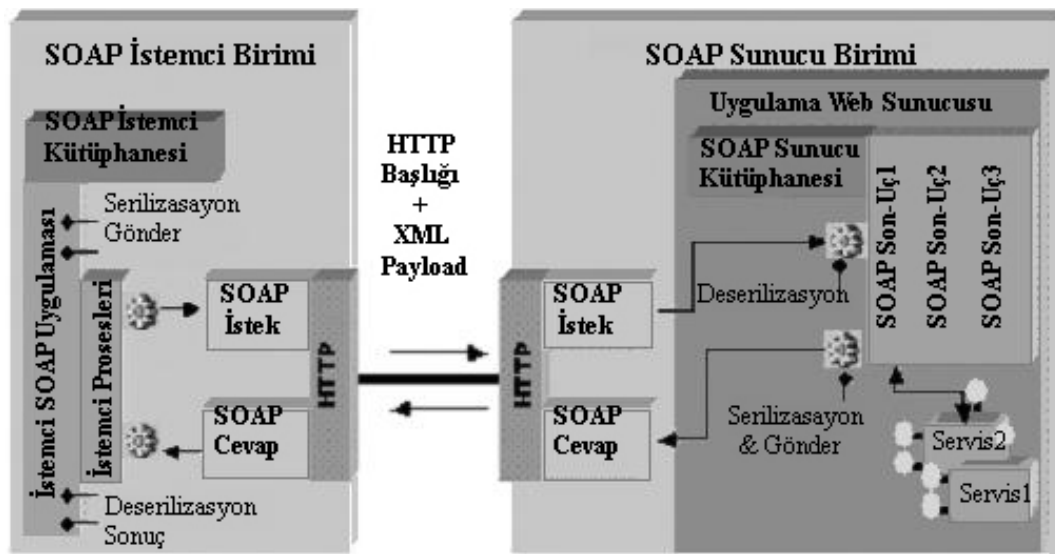
SOAP tanımları, internet üzerinden gönderilecek XML verisinin alışverişi için bir mesajlaşma iskeletinden oluşur. Mesajlaşma iskeleti işletim sistemi, programlama dili ve dağıtık platformlara bakarsak oldukça doğal, basit ve geliştirilmesi kolaydır. SOAP web servislerini birbirine bağlamak için bir iletişim mekanizması sağlar [1].



Şekil 3.6. Farklı ortamların iletişimini sağlayan SOAP mesajları

Uygulama bütünleştirme için ara yazılım (middleware) olarak adlandırılan RPC(Remote Procedure Call), IIOP (Internet Inter - ORB Protocol) ve Java Uzak Metot Çalıştırma (RMI: Remote Method Invocation) çözümleri internet ortamında iletişim sağlama konusunda yetersiz kaldığından XML tabanlı SOAP önem kazanmıştır.

Şekil 3.7. bir SOAP istemci ve sunucusu arasındaki iletişimi göstermektedir. Bir SOAP uygulaması geliştirmek için istemci ve sunucuya SOAP geliştirme araçları ile birlikte gelen kütüphanelerin yüklenmesi gerekir. Bu kütüphaneler bir XML çözümleyicisi (parser) ve SOAP işlemcisi içerir. İstemci SOAP uygulaması bir istek mesajı oluşturarak bu isteği SOAP sunucusunda tanımlanmış servis uç noktalarının (end-point) çalıştırması için gönderir. SOAP sunucu ilgili servisi çalıştırdıktan sonra SOAP yanıt mesajı hazırlar. Hazırlanan SOAP yanıt mesajı istemciye iletilir.

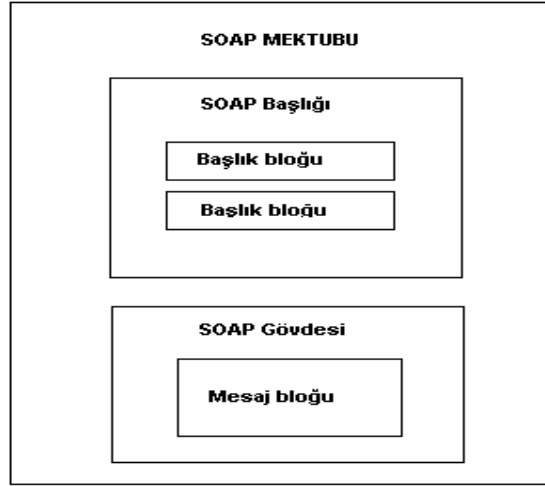


Şekil 3.7. SOAP istemci ve SOAP sunucusu arasındaki iletişim [17]

SOAP aşağıdaki temel kısımlardan oluşur:

- **Envelope:** Mesajın başlangıcını ve bitişini tanımlar.
- **Header:** Mesajın ara veya son işlem sırasında kullanılabilir herhangi bir opsiyonel niteliklerini içerir.
- **Body:** Gönderilmiş mesajdan oluşan XML verisi içerir.
- **Attachment:** Ana mesaja eklenmiş bir veya daha fazla dokümandan oluşur (sadece eklerden oluşan SOAP).
- **RPC interaction:** SOAP'la RPC iletişiminin nasıl modelleneceğini tanımlar.
- **Encoding:** Mesajla iletilen basit veya karmaşık verinin nasıl gösterileceğini tanımlar.

Şekil 3.8. HTTP protokolü ile gönderilen bir SOAP mesajını göstermektedir. SOAP mesajı HTTP POST metodu veri paketinin içinde gönderilir. Bir SOAP mesajı bir SOAP zarfından (SOAP envelope) oluşur. SOAP zarfı seçimlik bir SOAP başlığı (SOAP header) ve SOAP gövdesinden (SOAP body) oluşur. SOAP gövdesi çağırılacak metot ve metodun içerdiği parametreleri içerir.



Şekil 3.8. Bir SOAP mesajının yapısı [16]

Şekil 3.9. ve 3.10. bir SOAP istek ve yanıt mesajının içeriğini göstermektedir.

```

POST /WSIdManage/Service.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/isUserAuthenticate"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <isUserAuthenticate xmlns="http://tempuri.org/">
      <username>string</username>
      <password>string</password>
    </isUserAuthenticate>
  </soap:Body>
</soap:Envelope>
  
```

Şekil 3.9. Bir SOAP istemci istek (request) mesajı

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

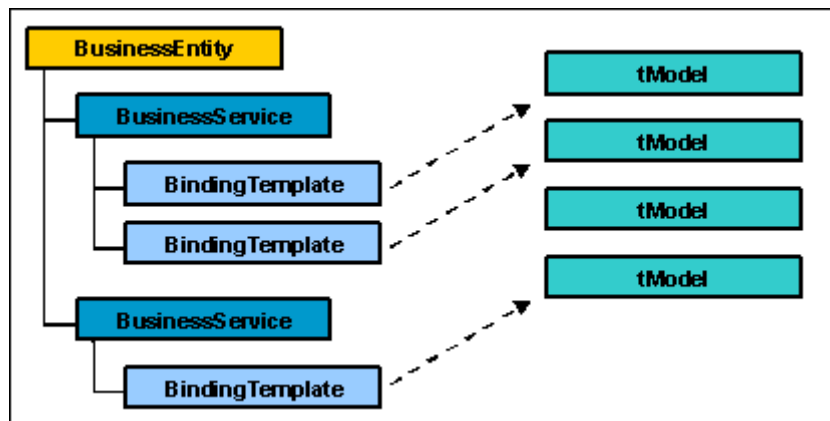
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <isUserAuthenticateResponse xmlns="http://tempuri.org/">
      <isUserAuthenticateResult>boolean</isUserAuthenticateResult>
    </isUserAuthenticateResponse>
  </soap:Body>
</soap:Envelope>

```

Şekil 3.10. Bir SOAP yanıt (response) mesajı

3.4 UDDI

UDDI, servis tanımlarının yayınlanması ve bulunması için bir metod sağlar. UDDI veri kayıtları, hem iş hem de servis bilgisinin tanımlanmasına destek verir. WSDL'de tanımlanmış servis tanım bilgisi UDDI kayıtlarında bulunan bilginin tamamlayıcısıdır. UDDI birçok farklı servis tanımlamalarına destek verir.

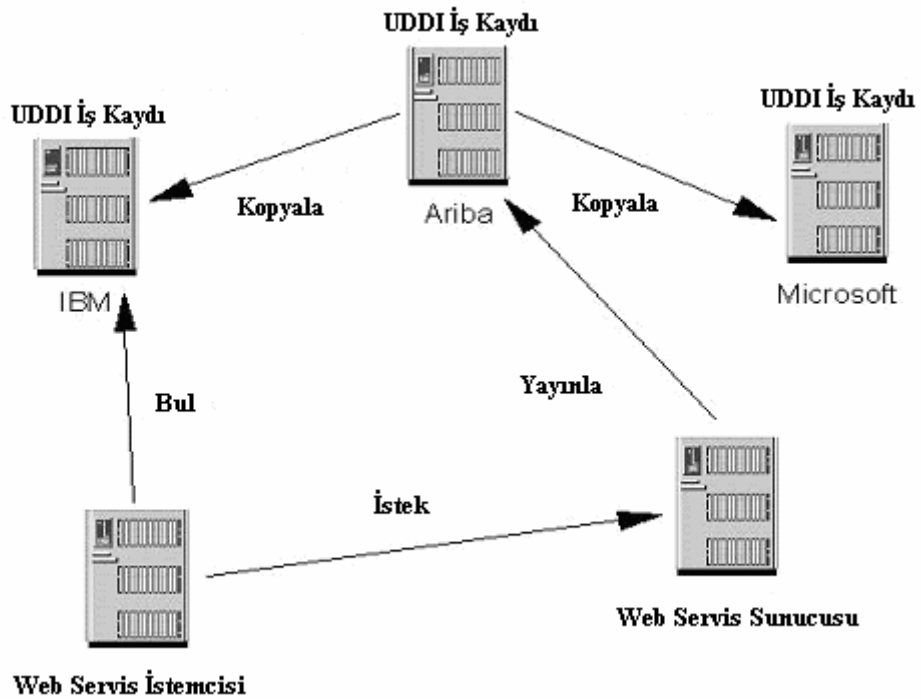


Şekil 3.11. UDDI veri tipleri [9]

UDDI, bir web servisini kullanmak için kullanıcının web servisi sağlayan kurumları ve bu kurumların verdikleri web servislerinin neler olduğunu bilmesini sağlar. UDDI kısaltmasında geçen evrensel, tanım, buluş ve bütünleştirme kelimelerinin ifade ettiği gibi UDDI kurumların kendilerini, sağladıkları servisleri

yayınlayarak tanımlamalarını, ve bu bilgilerin daha sonra diğer kurumlarca taranıp bulunmasını sağlayan bir standarttır[12-15].

UDDI Kurum Kayıt Servisi (UDDI Business Registry) kurum ve web servisleri bilgilerini saklayan sunuculardır. Bu sunucular servis sağlayıcılarından gelen bilgileri kendi veritabanlarına kayıt ederek diğer kurumların erişimine açar. Şu anda aktif olarak çalışan kurum kayıt sunucuları uddi.microsoft.com ve uddi.ibm.com'dur. Şekil 3.12.'de görüldüğü gibi bu sunucular kendilerine kayıt edilen bilgileri diğer sunuculara da kopyalayarak kolay ve hızlı bir şekilde erişilmesini sağlarlar[12]. UDDI sunucuları kurum ve servis kayıt, güncelleme ve tarama işlemlerini web servisleri (SOAP mesajları) ile gerçekleştirir.



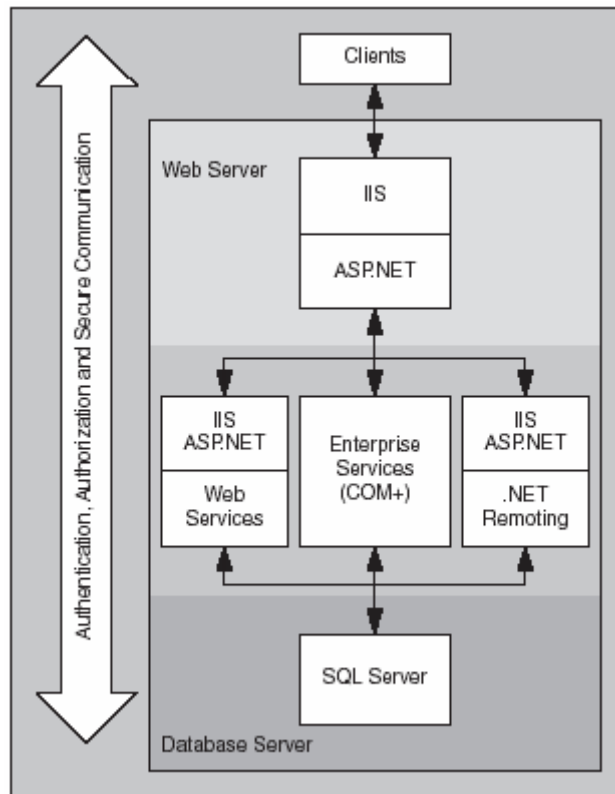
Şekil 3.12. UDDI kurum kayıt sunucuları [4]

4. KULLANICI TANIMLAMA VE YETKİLENDİRME

Dağıtık web uygulamalarında kullanıcı tanımlama ve yetkilendirme stratejisi geliştirmek oldukça zorlu bir iştir. Uygulama geliştirmenin erken saflarında yapılması belli başlı birçok güvenlik riskinin engellemesine olanak tanımaktadır. Kullanıcı tanımlama ve yetkilendirme birlikte düşünülmesi gerekmektedir, çünkü;

- Anlamlı bir kullanıcı tanımlama politikası yetkilendirilmiş kullanıcılara ihtiyaç duyar.
- Kullanıcıların tanımlandığı yol, yönetimdeki mevcut kısıtlar için kullanılacak yöntemleri belirtmeyi sağlamaktadır.

Bu yöntemlerden ASP.NET dosya yetkilendirme, Enterprise servis rolleri (COM+), Windowsun ACL'leri tanımlanmış Windows kimliğine ihtiyaç duyarlar. ASP.Net URL yetkilendirmesinde ve .Net rollerinde ise böyle bir gereklilik yoktur.



Şekil 4.1. .Net Web uygulamalarında güvenlik [19]

4.1 Strateji Tasarımı

Aşağıdaki adımlar kullanıcı tanımlama ve yetkilendirme stratejisi geliştirmek için kullanılabilir.

1. Kaynaklar belirlenir.
2. Kullanıcı tanımlama stratejisi seçilir.
3. Kaynaklara erişim için Kimlikler seçilir.
4. Kimlik akışı belirlenir.
5. Yetkilendirme yaklaşımı belirlenir.
6. Kimlik akışının nasıl olacağı belirlenir.

4.1.1 Kaynakların belirlenmesi

Kullanıcı tarafından erişilebilecek kaynakların neler olduğu tanımlanmaktadır. Bunlar:

Web sunucusu kaynakları; Web sayfaları, Web servisleri, HTML gibi statik içerik olabilir.

Veritabanı kaynakları; kullanıcıya veya uygulamaya özel veriler olabilir.

Network kaynakları; uzak dosya sistem ve Active Directory gibi klasör depoları verisi olabilir.

4.1.2 Kullanıcı tanımlama stratejisi seçimi

İki temel yaklaşım mevcuttur:

- **Rol bazlı**

Operasyonlara erişim roldeki tanımlanmış güvenlik kısıtlarına göre sağlanmaktadır. Roller uygulama içerisinde aynı güvenlik seviyesine sahip kullanıcıları gruplamak için kullanılmaktadır, Yöneticiler, Çalışanlar gibi. Kullanıcılar rollere atanarak o rollerin yapabileceği operasyonları gerçekleştirebilmektedir.

- **Kaynak bazlı**

Farklı kaynaklar için Windows ACL(Access Control List) listesi kullanılarak güvenlik sağlanmaktadır. İşlem yapmak isteyen kişi için işletim sistemi bazında standart erişim kontrolleri yapılmaktadır. Tüm kaynak erişimleri kullanıcının güvenlik içeriğine göre sağlanmaktadır. Bu yaklaşım uygulamanın ölçeklenebilirliğini oldukça etkilemektedir, çünkü bağlantı havuzu uygulamanın orta katmanında etkili bir biçimde kullanılamamaktadır.

.Net Web uygulamalarında ölçeklenebilirlik önemli bir kriter olduğundan rol bazlı yaklaşım en iyi seçim olmaktadır. Bazı küçük ölçekli kullanıcıya dosya sunan intranet uygulamalarında Windows ACL'lerinin kullanılması mantıklı olabilmektedir.

Rol bazlı kullanıcı tanımlamada aşağıdaki yol izlenebilir:

- Web uygulamaları içinden kullanıcıları tanımlama.
- Kullanıcıları rollerle ilişkilendirme.
- Rollere göre operasyon erişimlerini tanımlama.
- Veritabanı gibi arka tarafta erişim olacak sunuculara kullanıcıların erişimi için gerekli yetkilerin verilmesi.

4.1.3 Kaynaklara erişim için kimlik seçimi

Uygulamanın katmanları arasında kimlerin kaynaklara erişebileceğinin tanımlanmasıdır. Web uygulamalarından erişilen Web Servisleri, Enterprise servisleri, .Net Uzak Bileşenleri bu kaynaklardandır. Tüm durumlarda, kaynak erişim için kullanılan kimlik aşağıdakilerden biri olabilir:

- Çağıranın orjinal kimliği,
- Süreç kimliği,
- Servis hesabı,
- Kişiselleştirilmiş kimlikler.

4.1.4 Kimlik akışı belirleme

Uygulamada kaynağa erişmek için kullanıcının izleyeceği yol üzerinde geçeceği aşamalar belirlenmektedir. Eğer kullanıcı bazlı erişim söz konusu ise o kullanıcının erişeceği veritabanı gibi kaynakların söz konusu kullanıcının erişimine de izin veriyor olması gerekmektedir.

4.1.5 Yetkilendirme yaklaşımı seçimi

Bu aşamada iki önemli kriter göz önünde bulundurulur:

- Kullanıcıların kullandığı ortam (Windows accountları var mı hangi browserı kullanıyorlar).
- Uygulamanın ihtiyaç duyduğu kullanıcı tipleri belirleme (impersonation), delegasyon ve hesap denetimi.

4.1.6 Kimlik akışının izleyeceği yol belirleme

Uygulama seviyesinde ya da işletim sistemi seviyesinde güvenli içerik sağlamak için kimlik akışı belirlenmektedir.

Uygulama seviyesi için, metod ve stored prosedür parametreleri kullanılmaktadır. Kimlik akışı için aşağıdakiler desteklenmektedir:

- Kullanıcı başına tanımlanan güvenli sorgu parametreleri kullanılır, `SELECT x,y FROM SomeTable WHERE username="hale"`
- Herhangi bir uygulama seviyesinde denetim için özelleştirme yapılır.

İşletim sistemi seviyesi kimlik akışı için aşağıdakiler desteklenmektedir:

- Platform seviyesinde denetleme (Windows ve ya SQL Server gibi).
- Windows kullanıcı kimlikleri kullanılarak kişi bazlı yetkilendirme.

4.2 Kullanıcı Tanımlama Yaklaşımları

İki temel yaklaşım vardır. Bunlar:

4.2.1 Rol bazlı

Kullanıcılar uygulama tanımlı mantıksal rollere ayrılmıştır. Belirli bir rolün üyeleri uygulama içinde aynı haklara sahiptir. İşlemlere erişim bu rollere atanan kullanıcıların yetkilendirmesiyle gerçekleşmektedir.

Kullanıcılar tanımlanırken rollerle ilişkilendirilir ve o rolün gerçekleştirebileceği tüm haklara sahip olmaktadır.

Rol atamanın gerçekleştirebileceği senaryolar şunlardır:

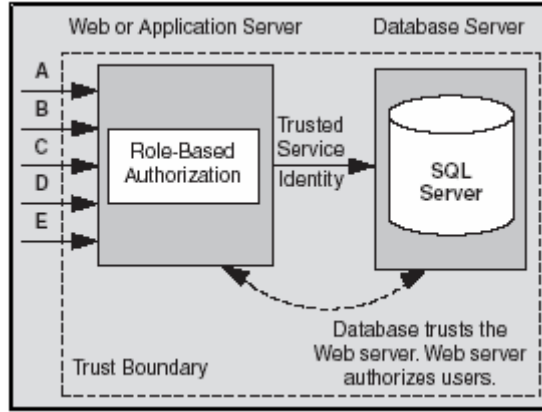
- Arka uç bir kaynakta roller atanabilir. Bu bir veritabanı olabilir.
- Ön uç olan web uygulamaları içinde roller tanımlanarak kişilerle ilişkilendirilebilir.
- Üçüncü opsiyonda önuç ve arkauç arasındaki Enterprise servisleri gibi ara katmanda rol atamasını gerçekleştirmektedir.

4.2.2 Kaynak bazlı

Kaynaklar Windows ACL'leri ile güvenli hale getirilmektedir. ACL'ler hangi kullanıcının hangi kaynaklara erişebileceğini ve hangi işlemleri gerçekleştirebileceğini (okuma, yazma, silme vs.) bilgilerini tutmaktadır. Bu yaklaşım genellikle dosyalar gibi kaynakların güvenliğini sağlamak için en iyi yaklaşım olmaktadır. .Net Web uygulamalarında kullanılan iki temel erişim modeli vardır. Bunlar:

- **Güvenilir Alt Sistem Modeli**

Bu modelde, orta katman servisleri alt katmana doğru inen servis ve kaynaklara erişim için sabit bir kimlik kullanır. Aşağıdaki resimde veritabanı çağırıcı kimliğini tanımlamak için orta katmana güvenilir ve sadece kimliğini tanımladığı çağırıcı için veritabanına erişim hakkı verir.



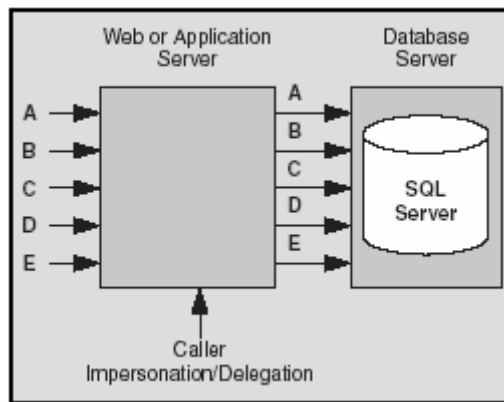
Şekil 4.2. Güvenilir alt sistem modeli [19]

- **Kullanıcı Tipleri Belirleme(impersonation)/ Delegasyon Modeli**

Bu modelde, bir servis ya da bileşen, kullanıcının bir sonraki ulaşacağı kaynağa geçmeden önce kullanıcının kimliğini taklit eder. Eğer bir sonraki servis aynı bilgisayarda ise taklit başarılı olur. Servis başka bir bilgisayardaysa delegasyon gereklidir.

Delegasyonun sonucu olarak, kullanıcının erişeceği kaynak için güvenlik içeriği kullanılır. Bu model birkaç sebepten dolayı kullanılabilir. Bunlar:

- Kullanıcının orjinal kimliğinin kullanılarak alt seviye erişim yetkilendirmesini kullanıcı başına yapılmasına izin verir.
- Alt seviye servisin işletim sistem seviyesinde denetleme özelliklerini kullanmasına izin verir.

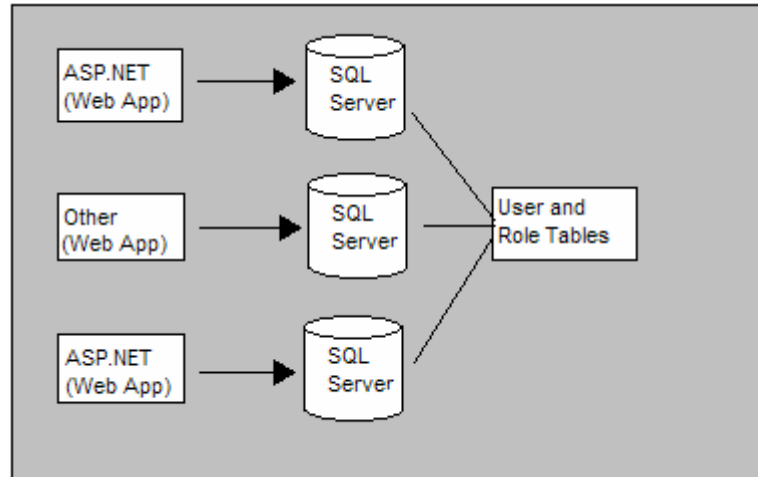


Şekil 4.3. Impersonation/delegasyon modeli [19]

5. ÇOKLU KULLANICI YÖNETİM SİSTEMİ

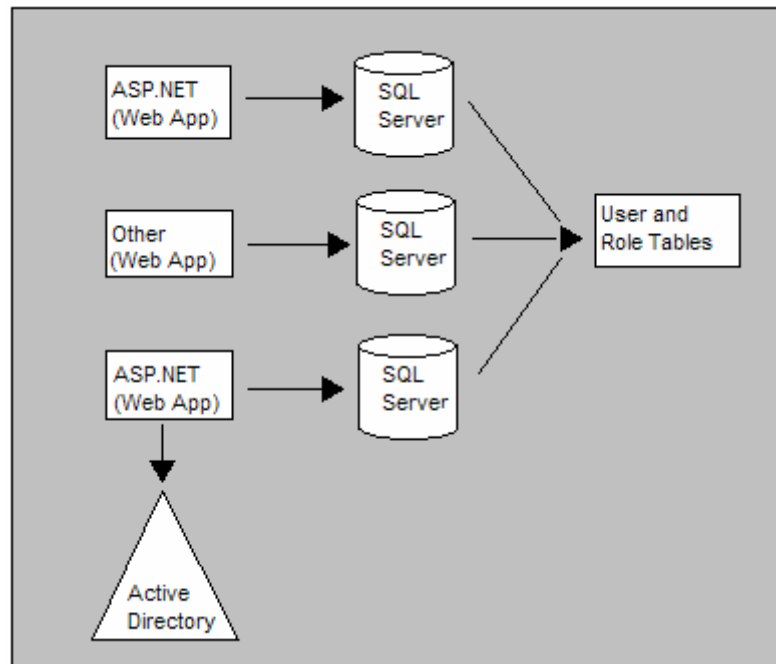
İnternetin yaygınlaşmasıyla birlikte birçok işletme iş süreçleri dijitize ederek lokasyon bağımsız web uygulamaları gerçekleştirmiştir. Bu web uygulamalarının kullanıcıları kendi çalışanları, iş ortakları ve müşterileri olabilmektedir. Kaynaklara erişimdeki güvenlik ve kişilerin yetkilerinin farklılaşması sebebiyle kullanıcı kimliği tanımlama ve aynı yetkilere sahip kullanıcıların roller altında gruplayarak erişimin denetlenmesi ihtiyacı ortaya çıkmaktadır.

Organizasyon içinde, intranet için geliştirilen web uygulamalarının bir kısmı, kendi kullanıcılarını ilgili veritabanında kullanıcı ve ilişkili rol tabloları tanımlayarak yönetmektedir. Şekil 5.1’de web uygulamaları ve ilişkili kullanıcı tabloları yer almaktadır. Bu yapı kullanıcı verilerinin tekrarlanmasına yol açtığından çok etkin olmamakla birlikte verinin tutarlılığı sağlanamamaktadır. Ayrıca her uygulama kendi kullanıcıları için kullanıcı bilgilerini sisteme ekleme, güncelleme ve silme fonksiyonlarını kod içinde gerçekleştirmek zorundadır. Bu şekilde hem uygulama hantallaşacak hem de zaman kaybı oluşacaktır.



Şekil 5.1. Web uygulamaları ve kullanıcı tabloları

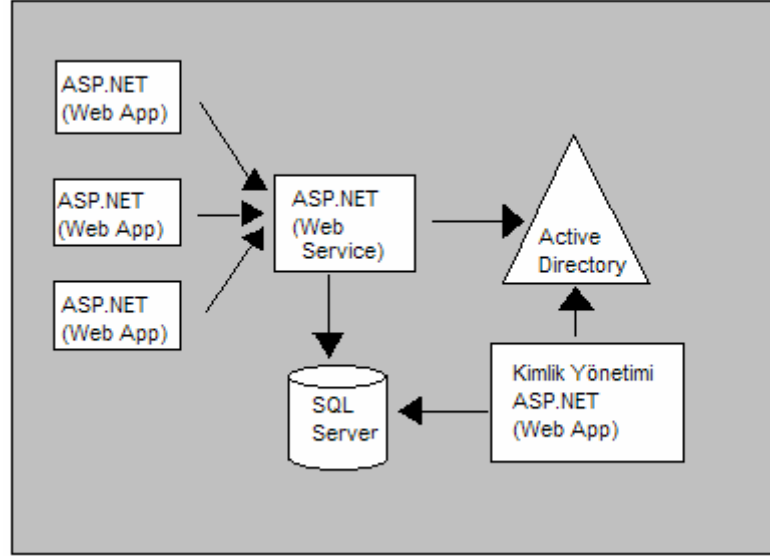
Kullanıcı yönetimi için gerçekleştirilen bir diğer yöntem ise, Şekil 5.2’de görüldüğü gibi kullanıcının anahtar olabilecek kimlik bilgilerini tutarak ilişkili rollerle tanımlamak, kişi bilgilerini almak için LDAP sunucusunu kullanarak kullanıcının organizasyonda tanımlanmış verilerine ulaşmaktır. Bu şekilde kullanıcı verisinin güncelliği ve kullanıcının organizasyon bilgileriyle tutarlılığı sağlanmaktadır. Buradaki problem ise her uygulamanın LDAP sunucusuna erişmek için uygulama tarafında kod yazarak ilgili fonksiyonları oluşturmasından kaynaklanan kodlama tekrarı ve doğal sonucu olan zaman kaybıdır.



Şekil 5.2. Web uygulamaları, kullanıcı tabloları ve LDAP sunucusu

Çoklu kullanıcı yönetim sisteminde, yukarıda bahsettiğimiz problemlere çözüm getirebilecek bir yapı tasarlanmıştır. Bu yapıda merkezi bir Kimlik Yönetim web uygulaması gerçekleştirilmektedir. Çoklu kullanıcı yönetim uygulamasında, internet ve sistemde bulunan intranet kullanıcıları için merkezi bir veritabanı kullanılmış, internet kullanıcılarının kimlik bilgileri veritabanında tutulurken, mevcut kullanıcılar için LDAP sunucusu ile bağlantı kurularak kullanıcının organizasyon içindeki bilgileri alınmaktadır.

Şekil 5.3.'te çoklu kullanıcı yönetim sistemi yapısı gösterilmektedir. Uygulamaların kullanıcılara ve ilişkili rollere erişimi için web servisleri mimarisini kullanılmakta, kullanıcı kimlik bilgileri web servisleri ile web uygulamalarına sunulmaktadır.



Şekil 5.3. Çoklu kullanıcı yönetim sistemi

Çoklu kullanıcı sisteminin avantajları aşağıda listelenmiştir:

- Merkezi yapı ile ölçeklenebilirliği artırılması,
- Kullanıcı bilgi tekrarının önlenmesi,
- Kullanıcı kimliğinin güncelliğinin korunması,
- Verinin tutarlılığının sağlanması,
- Her uygulama için ayrı ayrı kodlamanın ortadan kalkması,
- Zaman kayıplarının engellenmesi.

5.1 Sistem Mimarisi

Çoklu kullanıcı yönetim sistemi üç-katmanlı mimari yaklaşımı ile Visual Studio .Net 2005 platformunda C# dili kullanılarak ASP.Net uygulaması olarak kodlanmış ve web servisleri kullanılarak gerçekleştirilmiştir. ASP.Net platformu birçok programlama diline destek vermektedir. Bu yüzden web servisleri farklı dillerden meydana gelmiş olabilir.

Çoklu kullanıcı yönetim sistemi veritabanı sunucusu olarak Microsoft SQL Server 2000 kullanılmaktadır. Tablo 5.4 sistemde kullandığımız bileşenleri göstermektedir.

| | |
|----------------------------|------------------------------|
| Veritabanı Sunucusu | Microsoft SQL Server 2000 |
| Veritabanı Adı | SSODB |
| Platform | .Net Framework Server |
| Programlama Dili | C# (Asp.Net) |
| Geliştirilen Ortam | Microsoft Visual Studio 2005 |
| SOAP Servisi | Microsoft SOAP 3.0 |
| SOAP Versiyonu | SOAP 1.2 |
| XML Servisi | Microsoft XML 4.0 |

Şekil 5.4. Çoklu kullanıcı yönetim sistemi teknik özellikler

5.2 Veritabanı Yapısı

Sistemde kullanılmak üzere sistem ve web kullanıcıları için tablolar oluşturulmuştur. Web kullanıcıları için web servisleri veritabanı ile veri alışverişini gerçekleştirmektedir. Aşağıda, SSODB veritabanını meydana getiren SQL sorgusu yer almaktadır.

Kimlik Yönetimi Veritabanı

Veritabanı adı: SSODB

```
CREATE DATABASE [SSODB]
ON
    (NAME = N'SSODB_Data',
    FILENAME = N'C:\Program Files\Microsoft SQL
    Server\MSSQL\data\SSODB_Data.MDF' ,
    SIZE = 2,
    FILEGROWTH = 10%)
LOG ON
    (NAME = N'SSODB_Log',
    FILENAME = N'C:\Program Files\Microsoft SQL
    Server\MSSQL\data\SSODB_Log.LDF' ,
    SIZE = 2,
    FILEGROWTH = 10%)
COLLATE Latin1_General_CI_AS
```

SSODB veritabanı içerisinde bulunan tablolar şunlardır:

- **Acting Tablosu**

Kullanıcıların yetkilerini başka bir kullanıcıya devretmesi için gelecekte kullanılması amacıyla tasarlanmıştır.

```
CREATE TABLE [dbo].[Acting] (
    [acting_id] [int] NOT NULL ,
    [app_id] [int] NULL ,
    [actor_id] [nvarchar] (50) COLLATE Latin1_General_CI_AS NULL ,
    [acted_id] [nvarchar] (50) COLLATE Latin1_General_CI_AS NULL ,
    [date_from] [datetime] NULL ,
    [date_to] [datetime] NULL
```

```
) ON [PRIMARY]
GO
```

- **Application_Conn Tablosu**

Uygulamaların kullandığı veritabanı kaynakları gibi bağlantılarla ilişkilendirildiği tablodur.

```
CREATE TABLE [dbo].[Application_Conn] (
    [app_id] [int] NOT NULL ,
    [conn_id] [int] NOT NULL
) ON [PRIMARY]
GO
```

- **Application_Pool Tablosu**

Kullanıcı yönetim sistemini kullanacak web uygulamalarının kendilerini sisteme tanıtmaları için gereken tablodur.

App_id anahtar saha olup otomatik sıralı sayılarla verilmektedir.

Description sahası uygulamanın tanımı için kullanılmaktadır.

Url sahası uygulamanın adresini vermektedir.

isWeb sahası uygulamanın intranet ya da extranet uygulaması olup olmadığını tutar. 0 intranet, 1 extranet, 2 ise web uygulaması olduğunu göstermektedir.

```
CREATE TABLE [dbo].[Application_Pool] (
    [app_id] [int] IDENTITY (1, 1) NOT NULL ,
    [description] [nvarchar] (50) COLLATE Latin1_General_CI_AS NULL ,
    [url] [nvarchar] (100) COLLATE Latin1_General_CI_AS NULL ,
    [isWeb] [char] (1) COLLATE Latin1_General_CI_AS NULL
) ON [PRIMARY]
GO
```


- **Connection_Pool Tablosu**

Kaynakların tutulduğu tablodur. Uygulamaların erişmek istediği kaynaklar burada tanımlanır.

Conn_id anahtar saha olup otomatik sıralı sayılarla verilmektedir.

Description sahası bağlantının tanımını için kullanılmaktadır.

Type sahası kaynağın tipini belirtmektedir(LDAP, XML, SQL, Text File, Oracle).

Conn_user bağlantı için gerekli kullanıcı adıdır.

Conn_password bağlantı için gerekli şifredir.

Host sahası kaynağın bulunduğu IP(Internet Protocol) adresini tutar.

Connection_string ise bağlantı metnidir. Örnek, "Data Source=HALE;Initial Catalog=SSODB;User ID=ssodb_user;Password=1234"

CREATE TABLE [dbo].[Connection_Pool] (

[conn_id] [int] IDENTITY (1, 1) NOT NULL ,

[description] [nvarchar] (50) COLLATE Latin1_General_CI_AS NULL ,

[type] [nvarchar] (10) COLLATE Latin1_General_CI_AS NULL ,

[conn_user] [nvarchar] (50) COLLATE Latin1_General_CI_AS NULL ,

[conn_password][nvarchar](50) COLLATE Latin1_General_CI_AS NULL ,

[host] [nvarchar] (15) COLLATE Latin1_General_CI_AS NULL ,

[connection_string] [nvarchar] (50) COLLATE Latin1_General_CI_AS

NULL

) ON [PRIMARY] GO

- **LDAPUsersInRoles Tablosu**

İntranet uygulamaları için kullanıcıların LDAP kullanıcı adını kullanarak, uygulamalarla ilişkili rollerle, bağlantısının gerçekleştirildiği tablodur.

User_id LDAP'daki "samaccountname"e karşılık gelen kullanıcı adıdır.

Role_id Roles tablosunda herhangi bir uygulama için tanımlanmış rolün anahtar sahasıdır.

Bir kullanıcı birden çok rolle ilişkilendirilebilmektedir.

```

CREATE TABLE [dbo].[LDAPUsersInRoles] (
    [user_id] [nvarchar](50) COLLATE Latin1_General_CI_AS NOT NULL ,
    [role_id] [int] NOT NULL
) ON [PRIMARY]
GO

```

- **Roles Tablosu**

Web uygulamaları için kullanıcıların yetkileri bazında gruplandırılarak oluşan rollerin tanımlarının tutulduğu tablodur. Her uygulama yaratılırken default olarak “USER” rolü oluşturulur.

Role_id anahtar saha olup otomatik sıralı verilir(1, 2, 3, ..vb.).

Description rolün adını tutan sahadır(Admin,PE ..vb.).

App_id sahası Application_Pool tablosu ile ilişkiyi sağlayan yabancı anahtardır.

```

CREATE TABLE [dbo].[Roles] (
    [role_id] [int] IDENTITY (1, 1) NOT NULL ,
    [description] [nvarchar] (50) COLLATE Latin1_General_CI_AS NULL ,
    [app_id] [int] NULL
) ON [PRIMARY]
GO

```

- **WebUsers Tablosu**

Extranet ve web uygulamaları için sistemde olmayan kullanıcıların kimlik bilgilerinin tutulduğu tablodur.

User_id sisteme login olurken kullanılan kullanıcı adı alanı ve anahtar sahadır.

Password, kullanıcı şifresini tutan sahadır.

Email, kullanıcının e-mail adresini tutar.

Employee_id extranet kullanıcıları için gerekli çalışan numarasıdır.

Ad_Soyad kullanıcı adı ve soyadını tutar.

```
CREATE TABLE [dbo].[WebUsers] (  
    [user_id] [nvarchar](50) COLLATE Latin1_General_CI_AS NOT NULL ,  
    [password][nvarchar](10) COLLATE Latin1_General_CI_AS NOT NULL ,  
    [email] [nvarchar] (30) COLLATE Latin1_General_CI_AS NOT NULL ,  
    [employee_id] [int] ,  
    [ad_soyad] [nvarchar] (50) COLLATE Latin1_General_CI_AS NOT  
NULL  
    ) ON [PRIMARY]  
GO
```

- **WebUsersInRoles Tablosu**

Extranet ve web uygulamaları için kullanıcıların WebUsers tablosundaki kullanıcı adını kullanarak, uygulamalarla ilişkili rollerle, bağlantısının gerçekleştirildiği tablodur.

User_id WebUsers tablosundaki kullanıcı adına karşılık gelen yabancı anahtar sahasıdır.

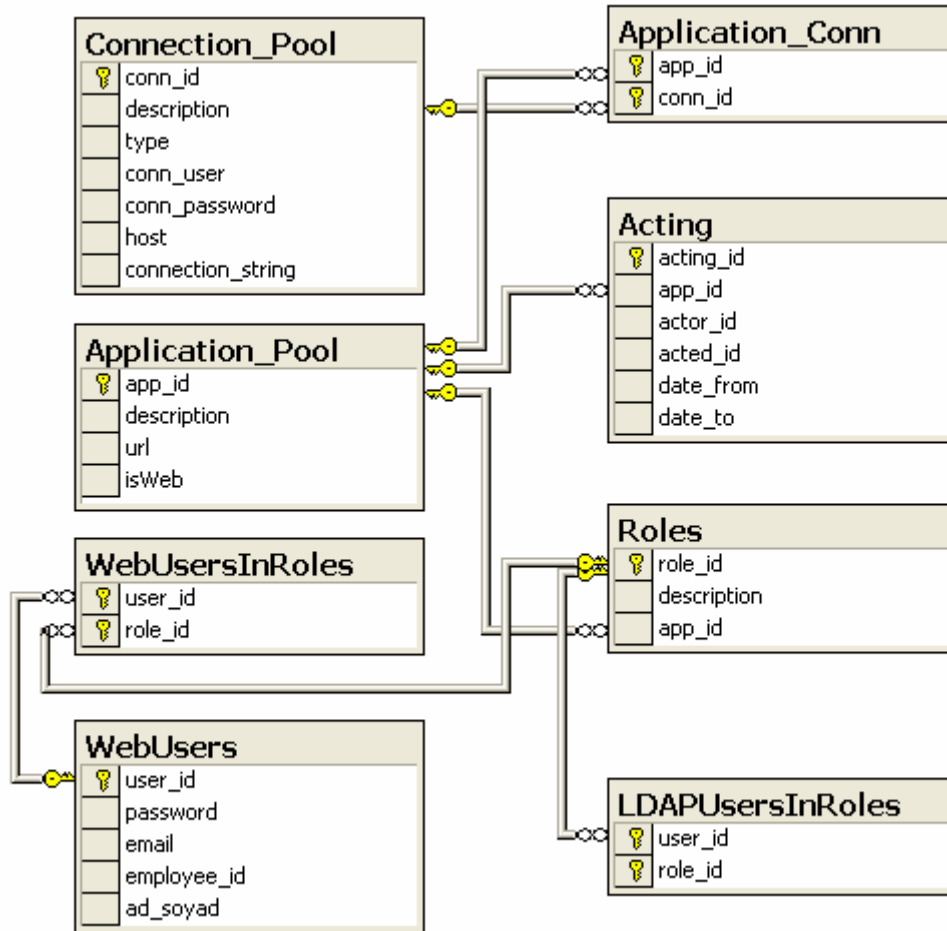
Role_id Roles tablosunda herhangi bir uygulama için tanımlanmış rolün anahtar sahasıdır.

Bir kullanıcı birden çok rolle ilişkilendirilebilmektedir.

```
CREATE TABLE [dbo].[WebUsersInRoles] (  
    [user_id] [nvarchar](50) COLLATE Latin1_General_CI_AS NOT NULL ,  
    [role_id] [int] NOT NULL  
    ) ON [PRIMARY]  
GO
```

5.3 Veritabanında Tablolar Arası İlişkiler

Şekil 5.5'te SSODB veritabanında yer alan tablolar arası ilişkiler gösterilmektedir. Kullanıcıların uygulamalarla roller sayesinde ilişkilendirilme tabloları “WebUsersInRoles” ve “LDAPUsersInRoles” dur. Acting tablosu geleceğe yönelik vekalet verme işlemleri için kullanıcı yetkilerinin bir başka kullanıcıya aktarılabilmesi için tasarlanmıştır.



Şekil 5.5. Veritabanı İlişkiler Diyagramı

5.4 Veritabanında Kullanılan Saklı Yordamlar ve İşlevleri

Veri tabanlarında temel SQL komutlarının yanında temel programlama işlevlerini de yerine getiren, her bir veri tabanı yönetimi sistemine özgü ayrı bir gösterim kullanılarak kodlanan yordamlardır. Bu yordamlara bir parametre gönderilerek çalıştırılırlar. Bunun yanında parametresiz çalışan saklı yordamlar da

oluşturulabilir. Saklı yordam kendisine verilen parametreler ve kodlanışı doğrultusunda belli işleri gerçekleştirir ve eğer varsa çıktı parametrelerini döndürür.

Saklı yordamlar modern ilişkisel veri tabanı yönetim sistemlerinde çok önemlidir. Derlendikleri için çok hızlı bir şekilde çalışırlar. Bunun yanında veri tablolarında belli işleri yerine getirirken üçüncü parti bir program kullanıldığında çok fazla veri alışverişi olacağından bunun yerine saklı yordamlar kullanmak işlerin çok daha hızlı ve sorunsuz bir şekilde çalışmasını sağlayacaktır.

Saklı yordamlar veri tabanında oluşturulduktan sonra bir tetikleyici (trigger), bir SQL sorgusu, üçüncü parti bir yazılımdan veya diğer bir saklı yordamın içerisinden çağrılabilirler.

Saklı yordamlar derlenmiş SQL cümlecikleridir. Bunlar birer veritabanı nesnesi oldukları için, veritabanı içinde yer alırlar. Bu nedenle veritabanı bir yere taşınmak istendiğinde otomatik olarak, saklı yordamlar taşınmaktadır. Saklı Yordam'lar derlenmiş olsalar da makine diline dönüştürülmemektedir. Çalıştırmak istediğimiz SQL cümleciklerini bir Saklı Yordam içine yerleştirerek, bunun bir veritabanı nesnesi haline gelmesini ve çalıştırıldığında doğrudan, veritabanı yöneticisini üzerinde barındıran sunucu makinede işlemlerini sağlarız. Bu doğal olarak, istemci makinelerdeki iş yükünü azaltır ve performansı artırır. Bir program içinde çalıştırılan SQL cümleleri, Saklı Yordam'lardan çok daha yavaş sonuç döndürmektedir. Dolayısıyla Saklı Yordamlar özellikle çok katlı mimariyi kullanan uygulamalarda faydalıdır.

Saklı Yordam kullanmanın avantajlarını genel hatları ile özetlersek;

- Taşınabilirlik
- Hız
- Güvenlik
- Performans
- İstemci yükünün azalması
- Network trafiğinin azalması

SSODB veritabanı içerisinde oluşturulan saklı yordamlardan bazılarını şunlardır:

Application_PoolInsert Saklı Yordamı: Kullanıcı yönetimi içinde Application_Pool sınıfından nesne oluşturulup veritabanına eklenmek istendiği zaman kullanılan saklı yordamdır. Bilgiler Application_Pool tablosuna ekledikten sonra, otomatik verdiği anahtar saha olan app_id değerini döndürür.

```
CREATE PROCEDURE [dbo].[Application_PoolInsert] (
    @app_id int OUTPUT,
    @description nvarchar(50),
    @url nvarchar(100))
AS
SET NOCOUNT ON
INSERT INTO [Application_Pool] (
    [description],
    [url]
) VALUES (
    @description,
    @url)
SET @app_id = SCOPE_IDENTITY()
GO
```

Application_PoolDelete Saklı Yordamı: Application_Pool tablosundan bilgileri silinmek istenen uygulamanın app_id'sini parametre olarak alır ve kaydı siler. Bu silme sırasında o uygulamayla ilişkili roller ve rollere atanmış kullanıcı bilgileri de tablolar arası ilişkiler sebebiyle silinmektedir.

```
CREATE PROCEDURE [dbo].[Application_PoolDelete] (
    @app_id int)
AS
SET NOCOUNT ON
DELETE FROM
    [Application_Pool]
WHERE
```

```
[app_id] = @app_id  
GO
```

Application_PoolSelect Saklı Yordamı: Application_Pool tablosundan bilgilerine ulaşılmak istenilen uygulamanın app_id'si parametre olarak verilerek uygulama bilgileri alınmaktadır.

```
CREATE PROCEDURE [dbo].[Application_PoolSelect](  
    @app_id int)  
AS  
SET NOCOUNT ON  
SELECT  
    [app_id],  
    [description],  
    [url]  
FROM  
    [Application_Pool]  
WHERE  
    [app_id] = @app_id  
GO
```

Application_PoolSelectAll Saklı Yordamı: Application_Pool tablosunda yer alan tüm uygulamalar ve bilgilerine ulaşılmak için kullanılmaktadır.

```
CREATE PROCEDURE [dbo].[Application_PoolSelectAll]  
AS  
SET NOCOUNT ON  
SELECT  
    [app_id],  
    [description],  
    [url]  
FROM  
    [Application_Pool]  
GO
```

WebUsersInsert Saklı Yordamı: Kullanıcı yönetimi içinde WebUser sınıfından nesne oluşturulup veritabanına eklenmek istendiği zaman kullanılan saklı yordamdır. Bilgiler WebUsers tablosuna kullanıcı bilgileri olarak eklenir.

```
CREATE PROCEDURE [dbo].[WebUsersInsert] (  
    @user_id nvarchar(50),  
    @password nvarchar(10),  
    @email nvarchar(30),  
    @employee_id int,  
    @ad_soyad nvarchar(50))  
AS  
SET NOCOUNT ON  
INSERT INTO [WebUsers] (  
    [user_id],  
    [password],  
    [email],  
    [employee_id],  
    [ad_soyad]  
) VALUES (  
    @user_id,  
    @password,  
    @email,  
    @employee_id,  
    @ad_soyad  
)  
GO
```

WebUsersDelete Saklı Yordamı: WebUsers tablosundan bilgileri silinmek istenen kullanıcının user_id'sini parametre olarak alır ve kullanıcı kaydını siler. Bu silme sırasında o kullanıcının WebUsersInRole tablosunda atanmış olduğu uygulama rolleriyle olan bilgileri de silinmektedir.


```
CREATE PROCEDURE [dbo].[WebUsersDelete] (  
  
    @user_id nvarchar(50))  
  
AS  
  
SET NOCOUNT ON  
  
DELETE FROM  
    [WebUsers]  
WHERE  
    [user_id] = @user_id  
  
GO
```

WebUsersSelect Saklı Yordamı: WebUsers tablosundan bilgilerine ulaşılmak istenilen web kullanıcısının user_id'si parametre olarak verilerek kullanıcı bilgileri alınmaktadır.

```
CREATE PROCEDURE [dbo].[WebUsersSelect] (  
  
    @user_id nvarchar(50))  
  
AS  
  
SET NOCOUNT ON  
  
SELECT  
    [user_id],  
    [password],  
    [email],  
    [employee_id],  
    [ad_soyad]  
FROM  
    [WebUsers]  
WHERE  
    [user_id] = @user_id  
  
GO
```

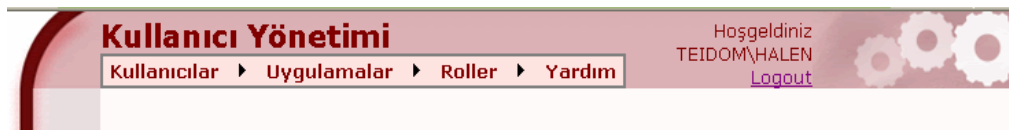
6. KULLANICI YÖNETİM SİSTEMİ ARAYÜZLERİ

Kullanıcı Yönetim Sistemi'nin kullanıcıları uygulama geliştirmeden sorumlu yazılım uzmanlarıdır. Her uygulama geliştiren kişinin uygulamasını sisteme kayıt etmesi için sisteme giriş yapması gerekmektedir. Şekil 6.1'de "Kullanıcı Yönetimine Giriş" ekranı gösterilmektedir.

Şekil 6.1. Kullanıcı Giriş Ekranı

Kullanıcı Yönetim Sistemi'ne giriş yaptıktan sonra karşımıza Şekil 6.2 deki gibi Kullanıcı Yönetimi menüsü gelmektedir. Bu menüde:

- Kullanıcılar,
- Uygulamalar,
- Roller,
- Yardım bulunmaktadır.



Şekil 6.2. Kullanıcı Yönetimi Menüsü

Sağ köşede kullanıcının kimlik bilgisi ve sistemden çıkış yapabilmek için gerekli "Logout" düğmesi bulunmaktadır.

6.1 Kullanıcı Modülü

Kullanıcı modülünde kullanıcılar sisteme eklenebilmekte(web kullanıcıları), bilgileri değiştirilebilmekte, LDAP kullanıcıları veya Web kullanıcıları uygulamalara atanabilmekte ve sistemde bulunan kullanıcılar görüntülenebilmektedir. Şekil 6.3'te "Kullanıcılar" menüsünün alt menüsü gösterilmektedir.



Şekil 6.3. Kullanıcılar Menü İçeriği

6.1.1 Kullanıcı ekleme

Kullanıcı ekleme alt menüsü ile web kullanıcı bilgileri Şekil 6.4'teki gibi sisteme eklenebilmektedir.

Şekil 6.4. Kullanıcı Ekleme

Kullanıcı bilgileri eklendikten sonra Şekil 6.5’de gösterilen “Kullanıcı Görüntüleme” ekranı çıkmaktadır. Bu ekranda eklediğimiz kullanıcının bilgileri görüntülediğinden doğruluğu test edilerek düzeltme yapılabilmektedir.

| Kullanıcı Görüntüleme | |
|--|-----------------------|
| Kullanıcı Bilgisi | |
| Kullanıcı Adı: | halen |
| Adı Soyadı : | Hale Nur Baş |
| Email : | halenur.bas@yahoo.com |
| Çalışan Numarası : | 1342 |
| Güncelle Sil | |

Şekil 6.5. Kullanıcı Görüntüleme

Kullanıcı bilgilerinde değişiklik yapmak için “Güncelle” düğmesi tıklanarak değişiklik yapılmakta ve Şekil 6.5’de görüldüğü gibi “Güncelle” düğmesine basılarak değişiklik kaydedilmektedir.

| Kullanıcı Görüntüleme | |
|--|--|
| Kullanıcı Güncelle | |
| Kullanıcı Adı : | halen |
| Email: | <input type="text" value="halenur.bas@yahoo.com"/> |
| Çalışan Numarası: | <input type="text" value="1378"/> |
| Ad Soyad: | <input type="text" value="Hale Nur Baş"/> |
| Güncelle İptal | |

Şekil 6.6. Kullanıcı Güncelleme

6.1.2 Kullanıcı bilgisi değiştirme

Kullanıcı bilgilerinde daha sonradan yapılmak istenilen değişiklikler için Şekil 6.7’de görüldüğü gibi kullanıcı listesinden değişiklik yapılmak istenen kişi seçilir. Sonra “Değiştir” düğmesine basılarak kullanıcı bilgilerine gidilir.

Şekil 6.7. Kullanıcı Bilgisi Değiştirme

Şekil 6.5’de görülen “Kullanıcı Görüntüleme” ekranı gelmekte ve kullanıcı bilgilerinde değişiklik yapmak için “Güncelle” düğmesi tıklanarak değişiklik yapılmakta ve Şekil 6.6’da görüldüğü gibi “Güncelle” düğmesine basılarak değişiklik kaydedilmektedir.

6.1.3 LDAP kullanıcılarının uygulamalara atanması

Kullanıcıların uygulamalarla ilişkilendirilmesi için “Kullanıcı Ata” alt menüsünden “LDAP Kullanıcısı” seçilir. Şekil 6.8’de görülen “LDAP Kullanıcıların Uygulamalara Atanması” ekranı gelir. İlgili uygulama seçilir, rol tanımına o uygulamanın rolleri gelir, rol bilgisi de seçildikten sonra LDAP’tan kullanıcı kutucuğuna gelen kullanıcı seçilerek “Kaydet” düğmesi tıklanır. Tıklandıktan sonra Şekil 6.8’in alt kısmında listelenen “Seçili Uygulamayla İlişkili LDAP Kullanıcılar” listesinde görüntülenir.

| Kullanıcı Adı | Rol | |
|---------------|-------|---------------------|
| HALEN | Admin | Sil |

Şekil 6.8. LDAP Kullanıcılarının Uygulamalara Atanması

6.1.4 Web kullanıcılarının uygulamalara atanması

Web kullanıcıların uygulamalarla ilişkilendirilmesi için “Kullanıcı Ata” alt menüsünden “Web Kullanıcısı” seçilir. Şekil 6.9’da görülen “Kullanıcıların Uygulamalara Atanması” ekranı gelir. İlgili uygulama seçilir, rol tanımına o uygulamanın rolleri gelir, rol bilgisi de seçildikten sonra kullanıcı seçilerek “Kaydet” düğmesi tıklanır. Tıklandıktan sonra Şekil 6.9’un alt kısmında listelenen “Seçili Uygulamayla İlişkili Kullanıcılar” listesinde görüntülenir.

Kullanıcıların Uygulamalara Atanması

Yeni Kullanıcı Ata

İlgili Uygulama: ▼

Rol Tanımı : ▼

Kullanıcı : ▼

Seçili Uygulamayla İlişkili Kullanıcılar

| Ad Soyad | Rol | |
|-----------|------|---------------------|
| Cenk Uzel | User | Sil |

Şekil 6.9. Web Kullanıcılarının Uygulamalara Atanması

6.1.5 Kullanıcıların listelenmesi

Kullanıcı bilgilerinin listelenmesi için “Kullanıcılar” alt menüsünün altında “Web Kullanıcıları” seçilir ve Şekil 6.10’daki gibi “Web Kullanıcı Listesi” görüntülenir.

Kullanıcı Yönetimi

Kullanıcılar ▶ Uygulamalar ▶ Roller ▶ Yardım

Web Kullanıcı Listesi

| Kullanıcı Adı | Email | Çalışan Numarası | Git |
|---------------|-----------------------|------------------|---------------------|
| cenku | cenku@tei.com.tr | 1271 | Git |
| halen | halenur.bas@yahoo.com | 1342 | Git |

Şekil 6.10. Web Kullanıcılarının Listelenmesi

6.2 Uygulama Modülü

Uygulama modülünde kullanıcılar sisteme geliştirmiş oldukları uygulama bilgilerini eklemekte, bu bilgileri değiştirebilmektedirler. Şekil 6.11’de “Uygulamalar” menüsünün alt menüsü gösterilmektedir.



Şekil 6.11. Uygulamalar Menü İçeriği

6.2.1 Uygulamaların eklenmesi

Uygulama ekleme alt menüsü ile uygulama bilgileri Şekil 6.12’deki gibi sisteme eklenebilmektedir.

The image shows a form titled 'Uygulama Ekleme'. It has a sub-section titled 'Yeni Uygulama'. There are two input fields: 'Uygulama Tanımı : Anket Uygulaması' and 'URL : http://webserver/anket'. Below the fields is a 'Kaydet' button.

Şekil 6.12. Uygulama Ekleme

Uygulama bilgileri eklendikten sonra Şekil 6.13’de gösterilen “Uygulama Görüntüleme” ekranı çıkmaktadır. Bu ekranda eklediğimiz uygulama bilgileri görüntülediğinden doğruluğu test edilerek düzeltme yapılabilir.

The image shows a screen titled 'Uygulama Görüntüleme'. It has a sub-section titled 'Uygulama Bilgisi'. There are two fields: 'Uygulama Tanımı: Anket Uygulaması' and 'Url : http://webserver/anket'. Below the fields is a 'Güncelle' button.

Şekil 6.13. Uygulama Bilgileri Görüntüleme

Uygulama bilgilerinde deęişiklik yapmak için ‘‘Güncelle’’ düğmesi tıklanarak deęişiklik yapılmakta ve Şekil 6.14’te görüldüğü gibi ‘‘Güncelle’’ düğmesine basılarak deęişiklik kaydedilmektedir.

Şekil 6.14. Uygulama Bilgilerini Güncelleme

6.2.2 Uygulama bilgilerinin deęiştirilmesi

Uygulama bilgilerinde daha sonradan yapılmak istenilen deęişiklikler için Şekil 6.15’te görüldüğü gibi uygulama listesinden deęişiklik yapılmak istenen uygulama tanımı seçilir. Sonra ‘‘Deęiştir’’ düğmesine basılarak uygulama bilgilerine gidilir.

Şekil 6.15. Uygulama Bilgilerinin Deęiştirilmesi

Şekil 6.13’te görülen ‘‘Uygulama Görüntüleme’’ ekranı gelmekte ve uygulama bilgilerinde deęişiklik yapmak için ‘‘Güncelle’’ düğmesi tıklanarak deęişiklik yapılmakta ve Şekil 6.14’te görüldüğü gibi ‘‘Güncelle’’ düğmesine basılarak deęişiklik kaydedilmektedir.

6.3 Rol Modülü

Rol modülünde kullanıcılar sisteme geliştirmiş oldukları uygulamalarla ilişkilendirmek istedikleri rol bilgilerini eklemekte, bu bilgileri değiştirebilmektedirler. Şekil 6.16’da “Roller” menüsünün alt menüsü gösterilmektedir.



Şekil 6.16. Roller Menü İçeriği

6.3.1 Rol ekleme

Rol ekleme alt menüsü ile uygulamalara rol bilgileri Şekil 6.17’deki gibi eklenebilmektedir. Eklenen roller “Seçili Uygulamayla İlişkili Roller” başlığı altında aynı ekranda kullanıcıya listelenmektedir. Kullanıcı eklediği bir rolü listeden seçerek silebilmektedir.

Rol Ekleme

Yeni Rol

İlgili Uygulama: ▼

Rol Tanımı :

Seçili Uygulamayla İlişkili Roller

| Rol Tanımı | |
|------------|---------------------|
| User | Sil |

Şekil 6.17. Rol Ekleme

6.3.2 Rol bilgilerinin değiştirilmesi

Rol bilgilerinde daha sonradan yapılmak istenilen değişiklikler için Şekil 6.18’de görüldüğü gibi önce ilgili uygulama seçilir, daha sonra rol listesinden değişiklik yapılmak istenen rol seçilir. Sonra “Değiştir” düğmesine basılır ve aşağısında yer alan “Seçili Uygulamayla İlgili Değiştirilecek Rol” bölümüne seçmiş olduğumuz rol bilgisi gelmektedir.

Rol Değiştirme

Değiştirilecek Rol

İlgili Uygulama: ▼

Rol Tanımı : ▼

Seçili Uygulamayla İlişkili Değiştirilecek Rol

Rol Görüntüle

| | |
|-------------|------|
| Rol Tanımı: | User |
|-------------|------|

[Güncelle](#)

Şekil 6.18. Rol Bilgisi Seçme

“Seçili Uygulamayla İlgili Değiştirilecek Rol” bölümüne gelmiş olan rol bilgisini değiştirmek için “Güncelle” düğmesine basılır ve Şekil 6.19’da görülen rol güncelleme ekranı açılır. Bu ekranda yeni rol tanımı girilmekte ve “Güncelle” düğmesine basılarak bilgiler saklanmaktadır.

Rol Değiştirme

Değiştirilecek Rol

İlgili Uygulama: ▼

Rol Tanımı : ▼

Seçili Uygulamayla İlişkili Değiştirilecek Rol

Rol Güncelle

| | |
|--------------|---|
| Rol Tanımı : | <input style="width: 150px;" type="text" value="User"/> |
|--------------|---|

[Güncelle İptal](#)

Şekil 6.19. Rol Bilgisi Güncelleme

6.4 Yardım

Yardım modülünde ise kullanıcılara sistemle ilgili bilgi verilmek amacıyla modüller ve o modüller altında gerçekleştirilebilecekleri işlemler gösterilmektedir. Şekil 6.20’de “Yardım” menüsü ile karşımıza çıkan ekran görüntüsü yer almaktadır.

| Kullanıcı Yönetimi | |
|---|-------------------------------|
| Hoşgeldiniz TEİDOMHALEN Logout | |
| Kullanıcılar ▶ | Uygulamalar ▶ |
| Roller ▶ | Yardım |
| Kullanıcı Yönetim Uygulaması Yardım MODÜLLER VE İŞLEMLER | |
| Kullanıcı Modülü | |
| Sistem kullanıcısı ekleme(LDAPtan) Web kullanıcısı ekleme Sistem kullanıcısı bilgisi görüntüleme Web kullanıcısı bilgisi görüntüleme Sistem kullanıcısı bilgisi değiştirme Web kullanıcısı bilgisi değiştirme Sistem kullanıcısını uygulamalar ve ilgili rollerle ilişkilendirme Web kullanıcısını uygulamalar ve ilgili rollerle ilişkilendirme | |
| Uygulama Modülü | |
| Uygulama bilgisi ekleme Uygulama bilgisi görüntüleme Uygulama bilgisi değiştirme | |
| Rol Modülü | |
| İlgili uygulamalara ilişkili rollerin eklenmesi Rollerin değiştirilmesi | |

Şekil 6.20. Yardım Menüsü

7. SONUÇLAR VE ÖNERİLER

Çoklu kullanıcı yönetim sistemi, organizasyon içindeki uygulamaların ve kullanıcıların merkezi bir yerden yönetilmesini, uygulamalarla kullanıcıların roller aracılığıyla ilişkilendirmesini ve bir standart oluşturulmasını olanaklı kılmaktadır. Her uygulamanın yöneticisi sistemde kayıtlı kullanıcılar için ya da web kullanıcıları için ilgili rolleri tanımlamakla ve uygulama için kullanıcı sistemine giriş yapmakla yetkilidir. Bu girişleri yaptıktan sonra kullanıcıların ilgili rollerle ilişkilendirilmesi için kullanıcı yönetim sistemini veya ilgili web servisini kullanabilmektedir.

LDAP'ta kullanıcı bilgisi değişikliklerinde uygulama yöneticilerinin herhangi bir takip yapması gerekmediğinden zaman kayıpları önlenmekte ve verinin güncelliği sağlanmaktadır.

Organizasyonun dışı açılan web uygulamaları (extranet veya internet) için istemci kimliklerini doğrulama ve yetkilendirme işlemleri Web servisleri ile gerçekleşmektedir. Web servisleri SQL Server ile bağlantı kurarak kullanıcı yönetim uygulamasının bir bölümü olan web kullanıcıları için kimlik doğrulama, yetkilendirme ve kullanıcı bilgisinin yönetimini kolaylaştırmaktadır.

Kullanıcı yönetiminde kullanılan Directory servisleri ile Active Directory'de özelleştirilmiş kullanıcı verilerinin alınması (kullanıcının çalışan numarası, çalıştığı birim bilgisi, ..vb.) durumunda, uygulamanın birlikte çalışacağı LDAP sunucusuna göre değişiklikler yapılarak uygulamanın çalışacağı ortamda rahatlıkla adaptasyonu sağlanabilmektedir. Bu uygulamada çalışan numarası olarak LDAP sunucusunda tanımlanan "Extensionattribute1" kullanılmıştır.

Ayrıca SQL veritabanı kullanıcıların birbirlerine vekalet bırakabilmeleri için gerekli yapıyı içerecek şekilde tasarlanmış olduğundan kullanıcı yönetim sisteminde belli tarihler aralığında kullanıcıların yetkilerini birbirine devretmelerine olanak sağlayacak modüller eklenerek geliştirme yapılabilir.

KAYNAKLAR

- [1] Newcomer, E., *Understanding Web Services: XML, WSDL, SOAP and UDDI*, Addison Wesley Professional, A.B.D., 2002
- [2] Akyokus, S., “Web Servisleri: İnternet Devriminde İkinci Aşama?”, Doğuş Üniversitesi, Bilgisayar Mühendisliğı, 2001
<http://inet-tr.org.tr/inetconf7/program/64.html>
- [3] Bosworth, A., “Developing Web services”, IEEE CNF, Data Engineering, Proceedings. 17th International Conference, Nisan 2001
- [4] www.ibm.com
- [5] King, C., “Getting Started on Developing Web Services”,
http://www.sun.com/bigadmin/content/developer/howtos/developing_websev.html
- [6] www.sun.com
- [7] Myerson, J., “Advancing the Web services stack”, Systems Architect/Engineer, Ocak 2002
<http://www.ibm.com/developerworks/webservices/library/ws-wsa/>
- [8] http://tr.wikipedia.org/wiki/Geni%C5%9Fletilebilir_i%C5%9Faretleme_dili
- [9] Brittenham, P., Ehnebuske, D., Graham, S., “Understanding WSDL in a UDDI registry”, Kasım 2001
<http://www.ibm.com/developerworks/webservices/library/ws-wsdl/>
- [10] Christensen, E., Curbera, F., Meredith, G., Weerawarana, S., “Web Services Description Language (WSDL) 1.1, W3C Note”, Mayıs 2001,
<http://www.w3.org/TR/wsdl>
- [11] Tyagi, S., “Patterns and Strategies for Building Document-Based Web Services”, Eylül 2004
<http://java.sun.com/developer/technicalArticles/xml/jaxrpcpatterns/index2.html>
- [12] Bellwood, T., “Understanding UDDI”, Temmuz 2002
<http://www.ibm.com/developerworks/library/ws-featuddi/>
- [13] “UDDI Technical White Paper”,
http://uddi.org/pubs/Iru_UDDI_Technical_White_Paper.PDF

- [14] uddi.microsoft.com
- [15] Clements, T., “Overview of SOAP”, Ağustos 2001,
[http://www.sun.com/bigadmin/content/developer/howtos/
overview_soap.html](http://www.sun.com/bigadmin/content/developer/howtos/overview_soap.html)
- [16] Godel, H., “SOAP, .NET, and COM - An Introduction: Part I”, Şubat 2004
[http://www.c-sharpcorner.com/UploadFile/jgodel/SOAPNETCOM
IntroductionpartI11162005042800AM/
IntroductionpartI.aspx](http://www.c-sharpcorner.com/UploadFile/jgodel/SOAPNETCOMIntroductionpartI11162005042800AM/SOAPNETCOMIntroductionpartI.aspx)
- [17] Thangarathinam, T., *Professional ASP.NET 2.0 XML*, 2006
- [18] “ASP.NET’te kimlik doğrulama”
[http://technet2.microsoft.com/WindowsServer/tr/Library/673ad9fe-9e97-
4669-a1f5-4383f45a3f7b1055.mspx?mfr=true](http://technet2.microsoft.com/WindowsServer/tr/Library/673ad9fe-9e97-4669-a1f5-4383f45a3f7b1055.mspx?mfr=true)
- [19] Meier, J.D., Mackman, A., Dunner, M., Vasireddy, S., “Building Secure ASP.NET Applications”, Microsoft Corporation, Kasım 2002, Son Revizyon Ocak 2006 [http://msdn2.microsoft.com/en-
us/library/aa302415.aspx](http://msdn2.microsoft.com/en-us/library/aa302415.aspx)