Çok Katmanlı WEB Tabanlı Uygulamalarda Yetkilendirme Problemi

Yenal Göğebakan
cyberSoft, Ankara, TURKEY
yenal.gogebakan@cs.com.tr

Abstract

Yetkilendirme amaçlı geliştirilmiş bulunan çeşitli çözümler kaynağa erişim hakkı kararını tek bir boyuta göre vermektedir; kullanıcının kimliği veya organizasyondaki rolu gibi. Oysa özellikle büyük ölçekli WEB tabanlı merkezi uygulamalarda erişim yetkisi kararının birden çok ve karmaşık boyutlar dikkate alınarak verilmesi gerekmektedir (veri, kullanıcının hangi organizasyonda olduğu, islem miktari vb.)

Bu bildiride yetkilendirme problemi için birden çok boyutu dikkate alan, gerektiğinde erişim hakkı kararı için iş mantığı kurallarının kullanılabildiği, fakat bunun iş mantığı programlarının dışında ayrıca sistematik bir şekilde yapılabildiği, tüm yetkilendirme işlemlerinin kayit altina alınabildiği ve güvenlik gereksinimlerinin belirlediği tanıma yöntemi ile entegre çalışan ve cyberSoft tarafından geliştirilmiş bir çözüm olan "cyberSoft Authentication & Authorization Service" (CSAAS) anlatılmaktadır.

1. Giriş

Günümüz uygulamalarında kullanılan basit "kaynak - erişim hakkı" yetkilendirme çözümü iş gereksinimlerinin her geçen gün karmaşıklaşması sonucu yetersiz hale gelmiş, karmaşık yetkilendirme gereksinimleri programların içerisine gömülmüş ek kodlama ile sağlanmaya çalışılmış, bu ise anlaşılması ve bakımı zor programlara sebep olmuştur. Özellikle kullanıcı sayısındaki fazlalık, kullanıcı tiplerindeki çeşitlilik ve buna bağlı olarak yetkilendirme ihtiyaçlarındaki karmaşıklık WEB tabanlı Internet uygulamalarında yeni çözüm arayışlarına yol açmıştır.

Yetkilendirme çözümü kapsamında kullanılan Rol Tabanlı Erişim Kontrolü (RBAC - Role Based Access Control) [1], Erişim Kontrol Listeleri (ACL - Access Control List) veya Kaynak Erişim Kontrolü (RAD - Resource Access Decision) [2] gibi çözümler tek başlarına büyük ölçekli uygulamalarda yetersiz kalmaktadır.

Genel olarak sistem tarafından doğrulanmış (Authenticated) kullanıcıların yetkilendirme modeli basit olarak aşağıdaki şekilde tanımlanabilir :

Kullanıcı → (**Kaynak**, **İşlem**)

Bu modelde "Yetkilendirme" işlemi kullanıcı ile kaynak ve o kaynak üzerinde tanımlanmış işlem çiftleri arasında bir ilişki olarak tanımlanır. Yetki sadece kaynak erişimi için tanımlanırsa **işlem** boş olabilir.

Yetkilendirme modeli üç temel varsayım üzerine dayanır :

1. *Güvenli sistem* : (**Kaynak**, **İşlem**) çiftine yetki sadece ve sadece kullanıcı ile arasında tanımlanan ilişki varsa geçerlidir, ilişki yoksa yetkilendirme yoktur.
2. *Gereken en az yetki* : Kullanıcıya sadece gerekli olan en alt seviye yetki verilir.
3. *Yetkilendirme sistemi bilinci* : Uygulamalar yetkilendirme ihtiyaçları için bilinçli olarak bu modeli kullanır. Uygulamalar modeli kullanmadan yetkilendirme için karar vermezler.

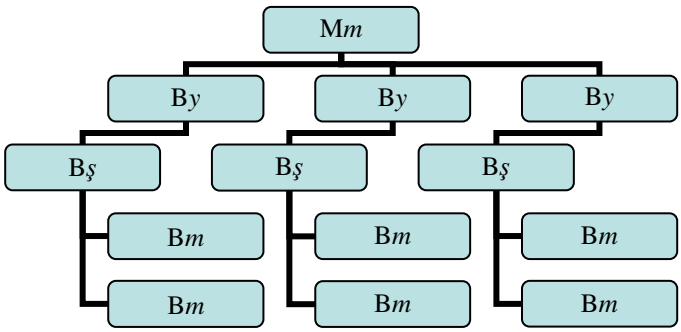
Yetkilendirme konusunda bu bildiride önerilen çözümü detaylı olarak ortaya koymak için tipik bir uygulama ve yetkilendirme ihtiyacı bir sonraki bölümde anlatılmıştır. Daha sonra yetkilendirme modelinde belirtilen ilişkiyi tanımlayan iki yaygın çözüm anlatılmış ve özellikleri verilmiştir. Bir sonraki bölümde önerilen CSAAS sistemi anlatılarak yetkilendirme problemi için geliştirilen çözüm anlatılmış ve sonuç bölümünde sistemin gelişmesi için öneriler sıralanmıştır.

2. Örnek Problem

WEB tabanlı uygulamalarda yetkilendirme problemi; uygulamaların ve verinin merkezi olmasından kaynaklanan tüm yetkilendirme kararlarının merkezi olarak alınması ihtiyacı nedeniyle daha fazla önem arzeder. Yetkilendirme kararlarının dayanağı olan kullanıcı kimliği, organizasyon, lokasyon gibi bilgiler tüm sistemi kapsayacak şekilde yetkilendirme sistemi içerisinde kullanılmalıdır.

Tanımlanan örnek problem bir bankanın kredi kullandırım sistemi ile ilgilidir. ***A*** Bankası belirli bir miktarı aşan kredileri (**K*büyük***) genel merkez, bu miktarın altındaki kredileri ise şubeleri aracılığı ile müşterilerine kullandırmaktadır. Kredi başvuruları her iki durumda da şubelere yapılmakta ve orada kredi miktarına göre değerlendirilmekte veya genel merkeze yönlendirilmektedir. Şubelerde değerlendirilen krediler memur (**B*m***), şef (**B*ş***) ve yönetici (**B*y***) onayı ile sonuçlandırılmaktadır. Genel merkeze yönlendirilen değerlendirmeler ise şube yöneticisi onayı ile Merkez Krediler müdürü (**M*m***) tarafından sonuçlandırılmaktadır.

A Bankasının organizasyon yapısı aşağıdaki şekilde tanımlanabilir :



Şekil 1

Kredilerin hangi şubeye gideceği müşteri adresi ile ilgilidir. Müşteriler sadece bağlı oldukları şubeye kredi başvurusunda bulunabilirler.

Şubeler sadece kendi müşterileri ile ilgili kredi bilgilerini görebilirler, diğer şubelerin kredi işlemleri ve bu işlemler ile ilgili bilgilere erişemezler.

Şubelerde personel durumuna göre şef veya yöneticilik başka servislerin şef veya yöneticileri tarafından yapılabilmektedir.

Doğal olarak yukarı doğru bir yetki hiyerarşisi mevcuttur. Memur ve şeflerin yapabildiği tüm işlemleri yöneticiler de yapabilmektedir.

Bir sonraki bölümde bahsedilen yetkilendirme ihtiyaçları için kullanılabilecek iki sistemin (RBAC ve RAD) nasıl bir çözüm sundukları ve eksiklikleri belirtilmiştir.

3. Rol Tabanlı Erişim Kontrolü (RBAC)

3.1. Rol Tabanlı Erişim Kontrolü (RBAC)

Rol Tabanlı Erişim Kontrolü (RBAC) yetkilendirme çözümü, giriş bölümünde bahsedilen modeldeki

Kullanıcı → (Kaynak, İşlem)

ilişkisini kişiler yerine organizasyona dayalı, rol olarak adlandirabileceğimiz genel yapılar üzerine kurmuştur. Bu çözümün dayandığı temel ilke organizasyonel yapıya dayalı rollerin, kişilerin aksine, kalıcı olduğu ve zaman içerisinde çok az değiştiğidir.

Roller; organizasyonel birimleri, sorumlulukları, yapılan belirli işleri tanımlayabilir ve genelde ağaç yapısında oluşturulurlar. Kullanıcılar birden fazla role ait olabilirler ve yetkiler de bu rollere verilir. Her bir rol birden fazla kaynak ve işlem yetkisini tanımlayabilir.

RBAC çözümünde yukarıdaki ilişki aşağıdaki şekilde yeniden yazılabilir :

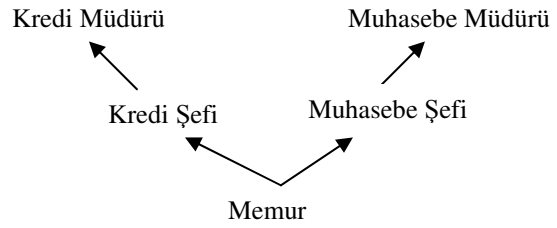
Kullanıcı → Roller → (Kaynak, İşlem)

Bu yapida kişiler bir veya birden fazla role atanirlar ve yetkiler de bu rollere verilir. Böylece, rollerin değişmeyeceği veya daha az değişeceği öngörülerek kullanıcılar değişse bile yetkilendirme yapısını değiştirmeye gerek olmayacaktır.

RBAC çözümünde roller hiyerarşik bir yapıda oluşturulurlar ve bir üst kademedeki rolun yetkileri altindaki rol tarafindan otomatikman sahip olunur. [4]

Şekil 2 de tipik bir rol ağacı gösterilmiştir. Bu ağaçta tanımlanan roller örnek problemde bahsedilen pozisyonlarla uyumludur. Rol ağacının kök bölümünde memur rolü tanımlanmıştır. Bu role tipik bir kullanıcının sahip olduğu yetkiler atanmıştır. 1. Giriş bölümünde bahsedilen modeldeki 2. kural gereği bu role memur rolünün sahip olması gereken en az yetki verilir.

Daha sonra iki ayrı birimin şef rolleri gösterilmiştir. Örnek olarak muhasebe şefi rolüne, muhasebe şefinin yapacağı işlemler ile ilgili yetkiler verilirken, kredi şefi rolüne kredi işlemleri ile ilgili yetkiler verilir. Bu roller ayni zamanda memur rolüne verilen yetkilere de sahip olurlar. Bu RBAC çözümünün tanımından gelen bir özelliktir [4].



Şekil 2. Ters (Inverted) Role Ağacı

Kullanıcılar roller için belirlenmiş yetkilere, bu rollere atanarak sahip olurlar. Muhasebe müdürü rolüne atanmış bir kullanıcı sırasıyla memur, muhasebe şefi ve muhasebe müdürü rollerine atanmış tüm yetkilere sahip olur Böylece yetkiler bir hiyerarşi içerisinde yönetilmiş olur.

Müdür kadrosundaki kullanıcı ise, bu örnekte, hem kredi müdürü hem de muhasebe müdürü rollerine atanarak her iki rolün yetkilerine de sahip olur.

Roller kullanıcılara belirli bir süre için de verilebilir.

RBAC çözümünün bir ileriki aşamasında kullanıcılar iki farklı gruptaki rollere atanmaktadır. Bunlardan ilki kullanıcının sahip olduğu rolleri, diğer ise sahip olduğu roller içerisinde eksiltilecek rolleri gösterir. Böylece muhasebe müdürü rolüne atanan bir kullanıcıdan muhasebe şefi yetkileri alınabilir.

3.2. RBAC Değerlendirme

RBAC çözümü örnek problemde bahsedilen bir çok sorunu çözmektedir. Yetkilerin bir hiyerarşi içerisinde yönetilebilir ve devredilebilir olması çok önemli bir avantajdır. İzine çıkan personelin sahip olduğu rollere yerine vekaleten bakan personel atanarak kolayca işlemler kesintisiz devam edebilir.

Yönetici, şef ve Memur kadrolari için yetkiler tanımlanabilir ve bu tüm organizasyon içerisinde standard olarak kullanılabilir. Terfi eden kullanıcılar yeni rollere kolaylıkla atanarak yeni pozisyonlarının gerektirdiği yetkilere sahip olurlar.

Sistem içerisinde geliştirilen yeni uygulamalar (kaynak ve işlem çiftleri) kolaylıkla roller ile eşleştirilerek uygun yetkilerle kullanıma alınabilirler. Tüm kullanıcılara tek tek yetki vermeden sadece rollere yetki tanımlayarak bu işlem yapılabilir.

WEB tabanlı uygulamaların hepsinde görülen merkezi program anlayışı gereği tüm organizasyonun merkezi olarak tek bir yapıda tutulması, RBAC çözümünün çok önemli bazı uygulama ihtiyaçlarını karşılamamaktadır. Örnek problemimizde tüm şubelerin şef kadrosundaki kullanıcıları şef rolüne atanmış olup bu rolün yetkilerine sahiptir. Fakat, şubenin sadece kendi alanı içerisindeki müşteriye hizmet etme koşulu RBAC tarafından sağlanamaz. Sistem yetkileri sadece rol tabanlı olarak tanımlamıştır. Oysa iş ihtiyacı bazı işlemler için rol dışında başka bilgilere göre yetki kararının verilmesini gerektirmektedir ve RBAC buna kendi tanımı içerisinde bir çözüm sunamamaktadır.

Şefe verilen kredi onay yetkisi de sınırlıdır. Kredi miktarının büyüklüğüne göre şef rolünün sahip olduğu onay yetkisi geçersiz olabilmektedir.

RBAC yetkilendirme konusunda çok önemli bir gelişme olmasına karşın yukarıda belirtildiği gibi iş mantığı bilgisi gereken konularda yetersiz kalmaktadır. Bu eksiklik genelde iş mantığı içerisine konulan program parçaları ile giderilmekte olup bakımı zor ve anlaşılmaz, düşük performanslı programlara sebep olmaktadır

4. Kaynak Erişim Kontrolü (RAD)

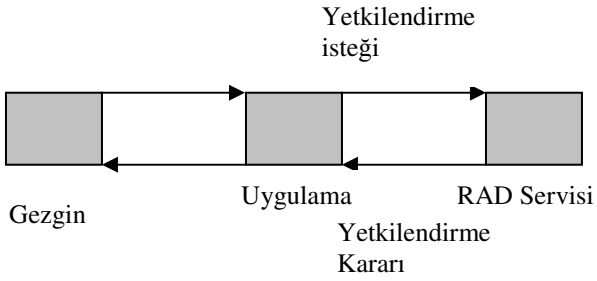
4.1. Kaynak Erişim Kontrolü (RAD)

Kaynak Erişim Kontrol (RAD) [2] çözümü yetkilendirme ihtiyacı için gereken programlamanın iş mantığı programlarının içerisine konulmadan gerçekleştirilmesini sağlamak için geliştirilmiştir. Uygulamaların istediği hassasiyet ve incelikte yetkilendirme kararlarının alınmasına imkan vermektedir. RBAC çözümünün en büyük eksikliği olan uygulama alanına ait veri ve iş mantığını yetkilendirme kararlarını alırken kullanabilir. RAD çözümünün bir diğer avantajı da yetkilendirme kararı için uygulama sistemi dışındaki kaynaklardan (geçiş kartları, legacy sistemler vb.) yararlanabilmesidir. Bu özelliği sayesinde PKI, bioSecurity gibi çözümler kolaylıkla entegre edilebilir.

RAD, 1. Giriş bölümünde tanımlanan modelin 3. özelliğine dayanarak uygulamaların her istek (bir kaynak üzerinde tanımlanmış bir işlemi gerçekleştirme

isteği) öncesi yetkilendirme konusunda bir karar almasını öngörür. Bu istek iş mantığı programlarının içine gömülü olmayıp, genel olarak sistem tarafından otomatik olarak yapılan bir işlem olarak düşünülmelidir.

RAD çözümü aşağıdaki gibi tanımlanabilir :



Şekil 3.

Şekil 3. de WEB tabanlı bir uygulama için RAD kullanımı gösterilmiştir. Gezgin kullanıcının uygulamalara eriştiği arayüzdür. Sadece sunum mantiği gezgin üzerinde olup iş mantığı uygulama olarak gösterilen ara katman üzerinde geliştirilmiştir. Gezgin tarafından yapılan tüm uygulama servis çağrıları uygulama katmanı tarafından önce otomatik olarak RAD servisine gönderilir. RAD servisi kendi içerisinde yetkilendirme kararını vererek uygulamaya döner. Merkezi uygulama, RAD dan gelen cevap üzerine talep edilen servisi çalıştırır veya çalıştırmaz.

Yukarıdaki sistemde belirtilmesi gereken en önemli konu yetkilendirme kararlarının uygulama tarafından otomatik olarak RAD servisine gönderildiğidir. İş mantığı programlarının içerisine ek program koymaya gerek yoktur.

RAD sisteminin yetkilendirme kararlarını verebilmesi için uygulama verilerine ve/veya dış sistem verilerine ihtiyacı olabilir. Uygulama verilerinin bir şekilde RAD sistemine geçirilmesi gerekmektedir. Dış sistem verileri ise RAD tarafından toplanır.

RAD sistemleri, bir önkoşul olmamasına rağmen genel olarak CORBA veya RMI servisi olarak geliştirilirler. Dinamik olarak yüklenen program kütühanesi (DLL) olarak da geliştirilmiş sürümleri mevcuttur.

Yetkilendirme kararları RAD içerisine konulan Yetki Politikasi (Security Policy) programları ile sağlanır. Bu politikalar genel olarak ikiye ayrılır; uygulamalardan bağımsız sistemin genelinde kullanılan ortak politikalar ve uygulamalara bağımlı, özel iş ihtiyaçları tarafından belirlenen politikalar. Her iki halde de yetki politikası programları sisteme kolayca eklenebilir ve çıkarılabilir.

Yetki politikaları, yetkilendirme kararları verirken ihtiyaç duyacakları verileri iki şekilde elde ederler. Uygulama bazlı veriler RAD tarafından otomatik olarak politika programına geçirilir. Sistem bazında alınacak veriler ile dış sistemlerden sağlanacak veriler ise yetki politikası programları tarafından sağlanır (JDBC, Http etc.)

4.2. RAD Değerlendirme

RAD çözümü özellikle uygulama alanı verisi gereken yetkilendirme kararları için ideal bir çözümdür. Istenildiği kadar hassasiyetle yetkilendirme kararları alınabilir.

RBAC çözümünün yetersiz kaldığı, şubenin sadece kendi bölgesindeki müşterilere hizmet edebilmesi yetkisi, RAD kullanılarak kolaylıkla çözümlenebilir. Kredi miktarlarına göre memur veya şefin onaylayabileceği krediler de bu çözüm ile basit bir yetki isteği ile belirlenir.

Günün saati, şubenin coğrafi konumu, ait olduğu bölge, müşterinin geçmiş işlemleri, yapılan işlemin büyüklüğü gibi verilerin zorunlu olarak kullanıldığı yetkilendirme kararları RAD bünyesinde standard bir yöntemle çözümlenebilir. Şubelerin sadece kendi yaptıkları kredi anlaşmalarını görmeleri, şeflerin sadece belirli miktarın altındaki kredi anlaşmalarını görebilmeleri ve genel merkezin tüm anlaşmaları görebilmesi RAD çözümü ile entegre çalışacak şekilde geliştirilecek bir filtreleme mekanizmasi ile mümkün olabilir.

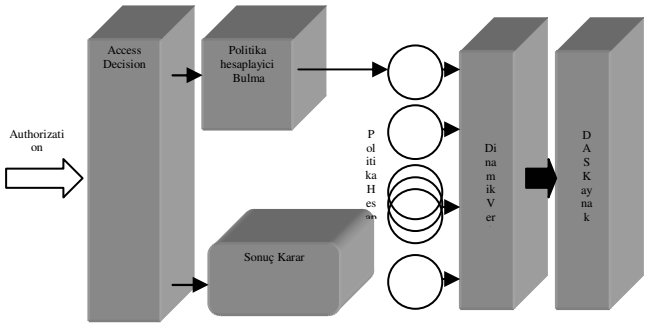
RAD ın esnek yapısı geliştirme ve bakım zorluğunu da beraberinde getirmektedir. RBAC ın sunduğu genel bir yetkilendirme yapısı sunmadığı için iş mantığı ve yetki sisteminde yapılacak değişikliklerin RAD sistemine yansıtılması başlı başına bir iştir. Özellikle uygulama veri yapılarının değişimine karşı RAD çok hassastır.

5. CSAAS Çözümü

cyberSoft firması özellikle enterprise ölçekli WEB tabanlı uygulamalar geliştirmekte olup bir süredir bahsedilen yetkilendirme problemini giderecek, bu yetkilendirme konusunda tüm projelerde kullanılacak, bakımı ve kullanımı kolay bir araç arayışında idi. Bu amaçla firma bünyesinde çeşitli yazılım mimarileri ve çözümler hakkında araştırmalar yapılmış, teorik ve

pratik bir çok çözüm incelenmiştir. Önceki bölümlerde bahsedilen RBAC ve RAD gibi genel kabul görmüş çözümler çeşitli projelerde kullanılmış fakat istenilen kalitede bir sonuç elde edilememiştir. Bunun üzerine RBAC ve RAD çözümlerini birleştiren, enterprise ölçekli uygulamaların ihtiyaçlarına cevap veren, yüksek performanslı bir yetkilendirme çözümü olan CSAAS tasarlanarak geliştirilmiştir. Şu anda 3 büyük ölçekli projede (ki bunlardan biri Internet uygulaması diğeri de 10,000 kullanıcılı bir uygulamadır) kullanılmakta olup bir program dahilinde yeni özellikler eklenmektedir. [6]

CSAAS mimarisi aşağıda verilmiştir :



Şekil 4.

CSAAS mimari olarak 3 temel parçadan oluşur.

1. *Dinamik Veri Servisi* : Yetkilendirme kararlarını verecek politikaların ihtiyaç duyacakları ve uygulamadan gelmeyen harici verilere CSAAS bünyesinden kolaylıkla erişilmesini sağlayan servis. Mevcut sürümde JDBC uyumlu veri tabanları, LDAP ve text kütüklere erişim sağlanmaktadır.
2. *Politika Hesaplayıcı Belirleme Servisi* : Yetkilendirme istenilen servis için, yani (kaynak, işlem) ikilisi için hangi yetki politikası program veya programlarının çalıştırılacağını belirleyen servis. CSAAS de birden fazla politika yetkilendirme için kullanılır. Böylece küçük, tekrar kullanılabilir politikalar yazmak mümkündür.
3. *Karar Servisi* : Birden fazla politika hesaplama programından oluşmuş yetkilendirme servisleri için politika kararlarını toplayarak onlardan sonuç kararı üreten servistir. AND, OR ve MAJOR karar vericileri CSAAS içerisinde hazır olarak sunulmakadır.

CSAAS bünyesinde yetkilendirme politikası programları Java Script programları, Java sınıfları ve CSAAS'a özgü bir şekilde geliştirilmiş declarative kurallar olmak üzere 3 şekilde geliştirilebilir. Örnek bir yetki politikası programı olarak RBAC politika programı gösterilebilir. CSAAS içerisinde RBAC bir politika olarak tasarlanmış ve geliştirilmiştir. Uygulamalardan kullanıcı ID sini veri olarak alır, dinamik veri servisinden o kullanıcıya ait rolleri ve roller için tanımlanmış yetkileri alır ve istenilen işlem için yetki kararını verir.

Uygulamalar yetkilendirme isteklerini, iş mantığı programlarına birşey eklemeden veya tek bir yerde ekleyerek CSAAS a yönlendirirler. Şekil 4'ün en solundaki blok olarak gösterilen kısım bu istekleri alır ve uygulamadan gelen verilerle birlikte politika hesaplayıcı belirleme servisine geçirir. CSAAS'ın bu bölümü RMI servis olarak geliştirilmiştir.

Politika hesaplayıcı belirleme servisi tarafından belirlenen politik programları çalıştırılır ve sonuçları karar servisi tarafından birleştirilerek yetki kararı oluşturulur. Yetki politikaları programları çalışırken gerek duyduklari verileri dinamik veri servisinden alırlar.

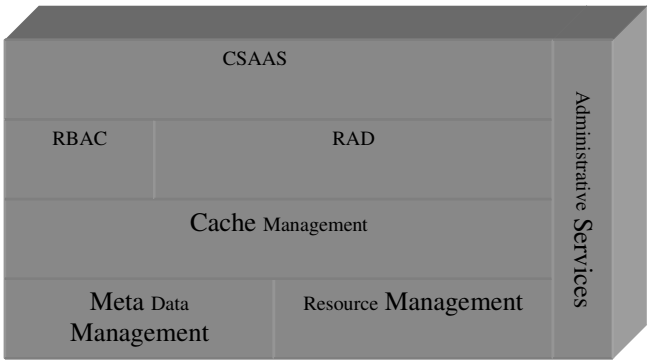
Örnek problemde belirtilen her şubenin sadece kendi bölgesine ait kredi bilgisine erişmesi yetki ihtiyacı CSAAS kullanılarak kolaylıkla giderilebilir. Bu amaçla gereken yetki politikası programı declarative bir kural olarak tanımlanır ve sisteme tanıtılır. Gerekli ek tanımlamalar da yapıldıktan sonra her bir kredi tanımlama isteği CSAAS'a otomatik olarak yönlendirilir. Önce RBAC politikasından geçen istek, kullanıcının şef veya yönetici rollerine sahip olma durumuna göre yetkilendirilir veya reddedilir. Bahsedilen senaryo RBAC ve RAD yetkilendirme servislerinin doğrudan adresleyemediği bir durumun iş mantığı programlarına ekleme yapılmadan çözülmesini göstermek açısından bir örnektir.

CSAAS servisinin mevcut sürümü Java RMI teknolojisi kullanılarak geliştirilmiştir. Böylece hemen tüm sistemlerde (UNIX, Linux, MS Windows) çalışması mümkündür. Ayrıca ugulama programlarının geliştirilme ortamına bağımlı değildir, Java, CORBA, J2EE, Servlet gibi ortamlarda geliştirilmiş uygulamalar tarafından kolaylıkla kullanılabilir.

Kullanıcı doğrulama servisi CSAAS in içerisinde değiştirilebilir bir modul olarak tasarlanıp geliştirilmiştir. Günümüz ihtiyaçlarına göre yeni kullanıcı tanımlama modülleri (smartCard, biolojik kullanıcı doğrulama teknikleri vb.) kolaylıkla sisteme eklenebilir. Şu anda kullanılan doğrulama modülü JDBC uyumlu bir veri tabanı veya LDAP üzerinde tutulan

kullanıcı ID ve şifre (password) ikilisinin kullanıcıdan alınarak kontrol edilmesi yöntemini kullanmaktadır.

CSASS servisi, mimari olarak çeşitli katmanlardan oluşmuştur.



Bu katmanlardan "Cache Yönetim" katmanı henüz geliştirilmemiştir.

6. Sonuç

Bu bildiride özellikle büyük ölçekli WEB tabanlı uygulamalar için kullanılabilecek olan bir yetkilendirme çözümü olan CSAAS anlatılmıştır.

RBAC ve RAD çözümlerinin problemleri belirtilmiş ve bu çözümlerin eksiklikleri örnek bir problem aracılığı ile ortaya konulmuştur. Günümüz uygulamalarının daha çok WEB tabanlı olması ve programların merkezi olarak çalışması bahsedilen problemlerin daha da büyümesine sebep vermiştir. Ayrıca her iki çözüm de sistem yönetimi ve bakım ihitiyacı olarak çok büyük yükler getirmektedir.

Bahsedilen problemleri giderecek, bakımı kolay, enterprise ölçekli uygulamaların ihtiyaç duyduğu peformans değerini yakalayabilecek ve uygulamalar tarafından kolayca kullanılabilecek bir yetkilendirme sistemi olan CSAAS, cyberSoft tarafından tasarlanıp geliştirilmiştir.

Tüm yetkilendirme ihtiyaçlarına cevap verecek bir servis olarak geliştirilen CSAAS, kullanılması halinde uygulamalar için kritik bir parça olmaktadır. Bu yüzden mutlaka "load balance" ve "failover" özellikleri eklenmelidir. Ayrica performansı artırmak için cache yönetim katmanı geliştirilmelidir.

Referans

[1] Ravi S. Sandhu "Role Based Access Control"

[2] OMG Specification "RAD v1.0"

[3] Role-Based Access Control Models (RBAC) : Fetures and Motivations, David Ferrailo et Ali.,Computer Security Applications Conference , December 1995

[4] "Inheritance Properties of Role Hierarchies", W.A.Jansen, National Instittute of Satandards and Technology

[5] Internet Resource Access Control, The Butterfly Model, NorCom WhitePaper

[6]CSAAS, cyberSoft Authentication & Authorization Service, 2000, WhitePaper