

AURORA e-İş Platformu

Web'i Tekrar Şekillendirmek



Web'i kurumlar için bütünüyle güçlendirir...

Problem

İnternet teknolojilerinin ortaya çıkışıyla kurumlar, Web'in sunduğu; global erişim, kesintisiz servis, ortak kullanım ve ciddi maliyet tasarrufu potansiyeli gibi, emsalsiz avantajları kısa sürede keşfettiler. Ama, tüm bu avantajlar belirli eksiklikleri de beraberlerinde getiriyordu. Bu eksikliklerin en önemli nedeni, Web'in orijin olarak uygulamaları yüklemek amacıyla değil, içerikleri yayınlamak amacıyla geliştirilmiş olmasıydı.

Bu orijinal spesifikasyonun neden olduğu etkileşim eksikliği, Web'in kurumsal uygulamalar için stratejik bir altyapı oluşturmasını da kısıtlamaktadır. Bu kısıtların aşılması için bazı çözüm yolları bulunmaktadır. Ama bu kısayol çözümler, kontrol edilmesi güç karışıklıklara, sıkıcı uygulama geliştirme döngülerine, ciddi bakım zorluklarına, hantal uygulama yükleme modellerine ve sonuç olarak da kontrol edilemeyen maliyetlere sebep olmaktadır.

Herhangi bir platform üzerinde bulunan Web tabanlı kurumsal uygulamaların, kullanıcı dostu ve gerçek zamanlı etkileşim vizyonu sunması bir zorunluluk haline almıştır. Böylesine gösterişli bir hedefe ulaşmak, mevcut Web teknolojilerinin "tıkla-ve-kullan" modelinin kısıtlamaları göz önüne alındığında, oldukça zor görünmektedir.

Günümüzde ise, kurumların gerçek anlamda aradıkları, türlerinin en iyi yaklaşımları olan kavramların bütünsel bir harmanlanmasıdır: **Web tabanlı çalışabilme, Masaüstü uygulamaların zengin fonksiyonelliğini içerme, sunucu/istemcilerin güçlü ve sürekli servis altyapısı, her yerden erişimin olanaklı kılınması, düşük sahip olma maliyeti.** Ama, Web istemcilerinin, Sıfır Geliştirme ve Yükleme Modeli (Zero Development and

Bu problemlerle karşılaşıyor musunuz ?

Kurumsal uygulamalarınızı Web tabanlı olarak sunmak istiyorsunuz. Ama;

- Web browser içinde benzer fonksiyonellikler sunamıyorsunuz
- Uygulamayı Web browser'ın dışında başlatamıyorsunuz
- Aynı uygulama olduğu halde, aynı görünüş ve kullanımı koruyamıyorsunuz
- GUI'nizin arkasında kod geliştirmek zorunda kalıyorsunuz
- GUI'nizi eskisi gibi prototipliyemiyorsunuz
- Uzman GUI programcıları istihdam etmek ve onlara çok ödemek zorunda kalıyorsunuz
- Excel, Word gibi dış uygulamalarla veri alışı (export/import) yapamıyorsunuz
- GUI'nizle çeşitli dış sistemlerle (akıllı kartlar gibi) entegre edemiyorsunuz
- HTTP(S) güçlükleriyle başedemiyorsunuz ya da daha esnek güvenlik konusunda daha zayıf iletişim yolları kullanıyorsunuz
- SOA'yla rekabet edebilecek bir sunucu tarafı iş uygulaması geliştiremiyorsunuz
- Dağıtık işlemleri yönetemiyorsunuz
- Web servislerin dışındaki B2B kaynaklarına erişemiyorsunuz
- Spring ve Hibernate gibi de facto uygulama çatılarını kullanamıyorsunuz
- Donanım konusunda büyük yatırımlar yapmak zorunda kalıyorsunuz

Deployment Model-ZDDM) ile sunabildikleri düşünüldüğünde, "masaüstü uygulama fonksiyonelliklerini içermek" oldukça iyimser bir kavram olarak görünmektedir. Eğer, sunucu tarafındaki modelin de, B2B kanalları üstünde mainframe benzeri bir güvenilir servis ve işlem desteği sunması beklenirse, kavramsal beklentiler daha da gösterişli bir durum almaktadır.

AURORA e-İş Platformu

Çözüm

Web üzerinde etkin kurumsal uygulamalar sunmak için, teknik kısıtlamalar sürekli olarak iş ihtiyaçları ile çatışmaktadır. Günümüzde, Web'in, teknolojik kısıtlarına rağmen, kurumsal uygulamalar için stratejik bir platform olarak adreslenmesini sağlayan teknoloji seçenekleri mevcuttur. Bu seçeneklerin çoğunluğu Zengin Internet Uygulaması (Rich Internet Application - RIA) ve/veya Kurumsal Internet Uygulaması (Enterprise Internet Application - EIA) modelleri içinde kategorize edilmektedir.

"İdeal" çözüm için ihtiyaçlara yönelik girişimler de vardır. Bu girişimler arasında DHTML/JavaScript, Macromedia Flash gibi RIA araçları, Swing/.NET teknolojileri kullanılarak geliştirilmiş kalın istemciler sayılabilir. Ama, bu girişimlerin hiçbiri, "ideal" çözümün talep ettiği ciddi ihtiyaçları bir kerede karşılayamamaktadır.



"İdeal" çözüm

- En üst seviyede masaüstü gücü kullanır
- Ultra-ince istemci yaklaşım
- İstemci tarafında sıfır kaynak kodu ile etkin kullanıcı yetenekleri
- İstemci tarafında kural çıkarsama motoru
- İstemci tarafında asenkronize rapor görselleştirme
- Etkin önbelleğe alma ve proxy mekanizmaları
- HTTPS dışında özelleştirilebilen güvenlik modelleri
- Diğer uygulamalarla kolay ve etkin entegrasyon
- İVYTS'ler dışında başka veri kaynaklarının da kullanımı
- Geliştirme yapmadan, hızlı bir şekilde Web servisleri sunumu
- Mevcut donanım ve yazılım yatırımlarını koruma
- Fazladan ölçekleme (upsizing) yerine doğru ölçekleme (rightsizing) yapılmasına imkan verir
- Platform bağımsızdır
- Mobil olarak genişleyebilir
- Arkasında iyi tanımlanmış bir metodoloji yer alır
- Hayat döngüsünün bütünü için araçlar ve teknikler barındırır
- Etkin prototipleme imkanı
- Web tabanlı bir geliştirme ortamına sahiptir
- İyi tanımlanmış bir veri geçişi stratejisi tanımlar
- Test edilmiş ve ispatlanmış bir başarı öyküleri
- Uygun bir lisanslama modeli

Aurora hakkında

Web'in yeniden şekillenmesindeki "ideal" çözümün ciddi ihtiyaçlarını karşılamak üzere, uluslararası kurumsal uygulamalar geliştirmedeki 15,000 adam-ay'lık tecrübesinden geçirilmiş olan know-how birikimiyle Cybersoft, Türkiye'nin ilk ve tek Yazılım Üretim Bandı (Software Product Line - SPL) olan **Aurora**'yı geliştirmiştir.

Aurora, Web tabanlı uygulamalar için RIA ve EIA modelleri, yazılım geliştirme süreci yönetim metodolojisi, gerçek görüntülü tasarım ve geliştirme ortamları, konfigürasyon yönetimi araç ve tekniklerinden oluşan temel bir altyapıyı barındıran platform bağımsız bir SPL'dir.

Aurora iş hayatının zorluklarına cevap veriyor...

Aurora kavramsal bir çerçeveden ibaret değildir. Tersine, altyapısal gücünü, bir temel bankacılık projesinin (3,000 adam-ay'lık bir proje) tasarım ve geliştirme aşaması için öngörülen toplam eforunu üçte bir oranında indirerek ve istenen çalıştırma performansı için düşünülen donanım konfigürasyonu beklentisini yarı yarıya azaltarak ispatlamıştır.

AURORA e-İş Platformu

Ayrıca, bu kurumun mevcut AS/400 programcılarının, 4 hafta gibi kısa bir sürede kurumsal Java ortamında da yetkin olabilmelerine olanak tanımıştır.

Bugün birçok kurum, Aurora'yı dünya çapında ismi olan uluslararası firmaların teknolojilerine tercih etmektedir. Aurora, birçok e-devlet uygulamasında aktif olarak kullanılmaktadır. Ayrıca, Cybersoft'un vizyon çözümlerinden biri olan I³ ERP (Kurumsal Kaynak Planlama) paketinin de temelidir.

Aurora kesinlikle fark yaratır...

Aurora'yı benzersiz kılan özelliği, yazılım geliştirme yaşam döngüsündeki gereksiz kodlama eforlarını ortadan kaldırarak, geliştirme maliyetlerini ciddi biçimde düşürmede gösterdiği rakipsiz yaklaşımıdır.

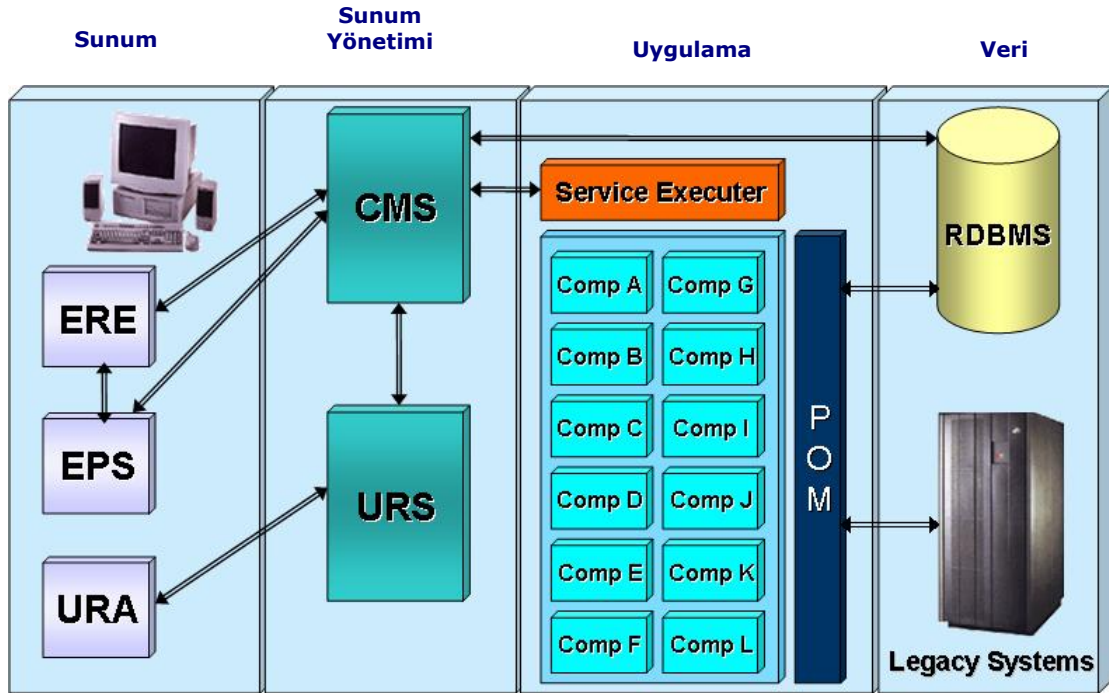
Aurora, bir Web uygulamasının her bir katmanının geliştirilmesinde kullanılacak teknikler ve araçlarla birlikte sunulmaktadır. Sunum Katmanı, kod gerektirmeden hazırlanmaktadır. İçerik Yönetimi Katmanı, kullanıcı oturumlarını ve özelleşmiş güvenlik çözümlerini yönetir. Uygulama Katmanı, doğru ölçeklendirme ve gerektiğinde genişleme modeli sayesinde benzersizdir. Veri Katmanı, her türlü veri kaynağını destekleyebilmektedir.

Aurora Sunum Katmanı

Çok iyi bilindiği gibi, Web uygulamaları geliştirmenin hataya en açık ve en zor aşamalarından biri olarak, sunum katmanının hazırlanması gösterilmektedir. Aurora, zengin istemcilerin, ultra-ince istemci yükleme yaklaşımı kullanılmadan yapılandırılabilmesi için Sıfır Geliştirme ve Yükleme Modeli (Zero Development and Deployment Model - ZDDM) sunmaktadır.

Aurora sadece ekran görselleştirmesi için çağdaş bir yol tanıtmakla kalmamakta; aynı zamanda raporlama ve çıktı alma ihtiyaçlarını da istemci tarafında dinamik olarak önbelleğe alma yolunu kullanarak güçlendirmektedir. J2EE ve .NET tarafından önerilen DHTML teknolojisi ve kendi-kendine-geliştir yaklaşımlarından ayrı olarak, Aurora XML teknolojisini ekran ve rapor görselleştirmeleri için de kullanmaktadır.

Aurora Sunum Katmanı, **User Interface Markup Language (UIML)** teknolojisinden türetilmiş bir yaklaşıma sahiptir. Aurora, UIML kavramını önceden öngörmüş ve platform bağımsız destek sağlamak için Swing GUI sınıflarını, Windows özgün istemci desteğini sağlamak için de ATL-bazlı özgün GUI sınıfları uyarlamıştır.



ERE: EBML Sunum Motoru
EPS: EBML Proxy Sunucusu
CMS: Sunum Yönetimi Sunucusu
URA: Evrensel Raporlama Aracı

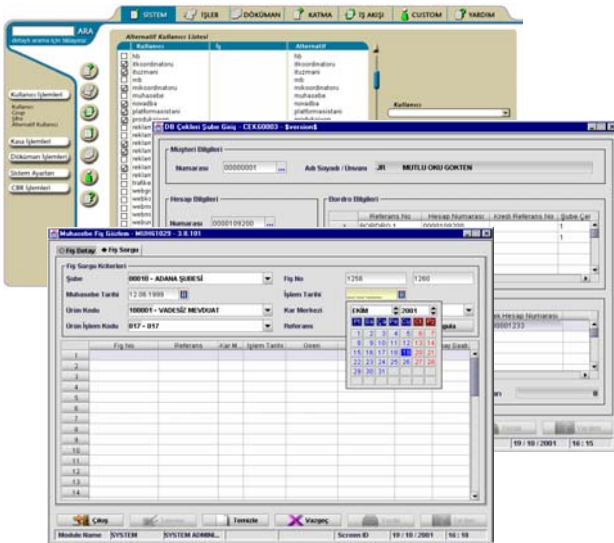
URS: Evrensel Raporlama Sunucusu
POM: Kalıcı Nesne Modeli
Comp: Bileşen
RDBMS: İlişkisel Veri Tabanı Yönetim Sistemi

AURORA e-İş Platformu

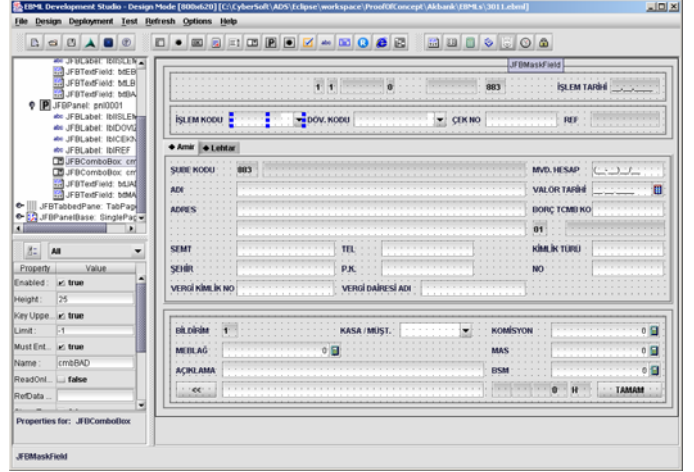
Görselleştirme için yeni bir XML

Aurora Platformu'nun tasarımcıları, UIML kavramını olduğu gibi kullanmayı tercih etmemişlerdir. Bunun nedeni, UIML teknolojisinin biçimleme ihtiyaçlarının yüksek olması ve orjinal spesifikasyonların, hızlı görselleştirme gerçekleştirilebilmesi için optimize edilmemiş olmasıdır. Ayrıca, tabedilmiş pencere bölümleri, hücre içi yazılabilir tablolar, ağaç tablolar (gezgin benzeri ekranlar) ve popup sorgulama kontrolleri gibi ileri arayüz kontrollerinin hazırlanması konusunda kısıtlı bir destek sağlayabiliyordu. Sonuç olarak, Aurora tasarımcıları, UIML'yi tekrar şekillendirip sadeleştirmişlerdir. Ayrıca JavaBean ve ActiveX teknolojileri kullanarak güçlendirip, ileri ekran kontrolleri ile zenginleştirmişlerdir. Bu çalışmaların sonucunda elde edilen yeni XML **Enhanced Bean Markup Language (EBML)** olarak adlandırılmıştır.

EBML, geçen birkaç yıl içinde geliştirilmiştir. Şu anda, tekrar kullanılabilir ekran bölümlerini sunabilmekte, mantıksal kontrol ve aritmetik işlem kurallarının tanımlanabilmesine olanak vermekte, lokal ve uzaktan metod çağırma işlemlerini yürütmekte, yapısal parçaları ayrı ayrı versiyonlayıp, önbelleğe alabilmekte, statik referans verisiyle çalışmakta ve aynı içeriği farklı yerel dillerde gösterebilmektedir (örneğin; aynı ekran, istemci tarafında herhangi bir kodlama ihtiyacı olmadan, uygun bir tasarımla İngilizce yada Fransızca olarak sunulabilmektedir).



Aurora kullanarak, zengin istemci uygulamaları, GUI geliştirirken sıfır kaynak kodu kullanarak oluşturulabilir.



EDS (EBML Design Studio) – zengin istemci uygulamaları için "WYSIWYG" bir geliştirme ortamı

Yeni XML görselleştirme motoru

EBML, Sunum ve İçerik Yönetimi katmanları arasında kavramsal bir köprü oluşturmaktadır. İstemci tarafında, bir Java applet ve ATL-tabanlı bir ActivateX'den bir uygulama parçası, görselleştirme motoru olarak çalışmaktadır. Bu motor **EBML Sunum Motoru (EBML Rendering Engine - ERE)** olarak adlandırılmaktadır. ERE, kullanıcı ilgili URL'e ilk bağlandığı zaman otomatik olarak yüklenebilen çok küçük boyutlu bir uygulamadır.

Web Browser'ların HTML içeriklerini sunmasına benzer olarak, ERE de, EBML ile ifade edilmiş olmaları kaydıyla, değişik ekranları sunabilmektedir. Güçlü tasarım ve geliştirme özellikleri dolayısıyla, Aurora üzerinde hazırlanmış olan Web uygulamalarının son kullanıcıları, kendilerini herhangi bir sunucu/istemci uygulaması kullanıyormuşçasına rahat hissedebilmektedirler. EBML sunumu, HTML sunumundan daha başarılı bir performans sunmaktadır. Bu performansın arkasındaki temel neden, MVC (Model-View-Controller) şablonunun kullanılmış olmasıdır. Bu şekilde, ekran tasarımının düzenlemeleri, gerçek verilerden ve metod çağırma işlemlerinden ayrıştırılarak önbelleğe alınmaktadır.

Önbelleğe alma yoluyla ağın etkin kullanılması

Kurumsal ölçekli Web uygulamaları, geniş hacimli verilerin sürekli olarak içerik sunucuları ve sunum katmanı arasında gidip

AURORA e-İş Platformu

gelmesini gerektirmektedir. Bu bağlamda, ekran yerleşimleri, referans veriler, gerçek veriler ve biçimlemeler sürekli olarak Web üzerinden aktarılmaktadır. Bu trafikte, Internet altyapısının verimini maksimize etmek amacıyla, transfer edilen veri büyüklüğünü azaltmayı amaçlayan önbellek kullanımı mekanizmaları öngörülmektedir. HTML terimleri kullanılacak olursa, statik içerik, proxy sunucuların kullanılmasıyla kolaylıkla önbelleğe alınabilmektedir. Ama, içerik dinamik öğeler barındırıyorsa, proxy sunucuların oldukça az yardımları olabilmektedir.

Aurora, bu probleme yönelik olarak bir çözüm oluşturmuştur. EBML, ekran tasarımları parçalarını (yeniden kullanılabilir bölgesel ekran parçalarının tasarımları dahil olmak üzere) referans ve gerçek verilerden ayırtedebilen tasarım özellikleri sayesinde HTML'den bir adım öndedir. Ayrıca, popup sorgu verilerini de, lokal veya uzaktan metod çağırma parametre/sonuçları gibi ayırtedebilmektedir. Internet'te akan verilerin bu tür bir kalibrasyonla ayırt edilebilmesi sayesinde, optimum önbelleğe alma algoritmaları hazırlanabilmektedir.

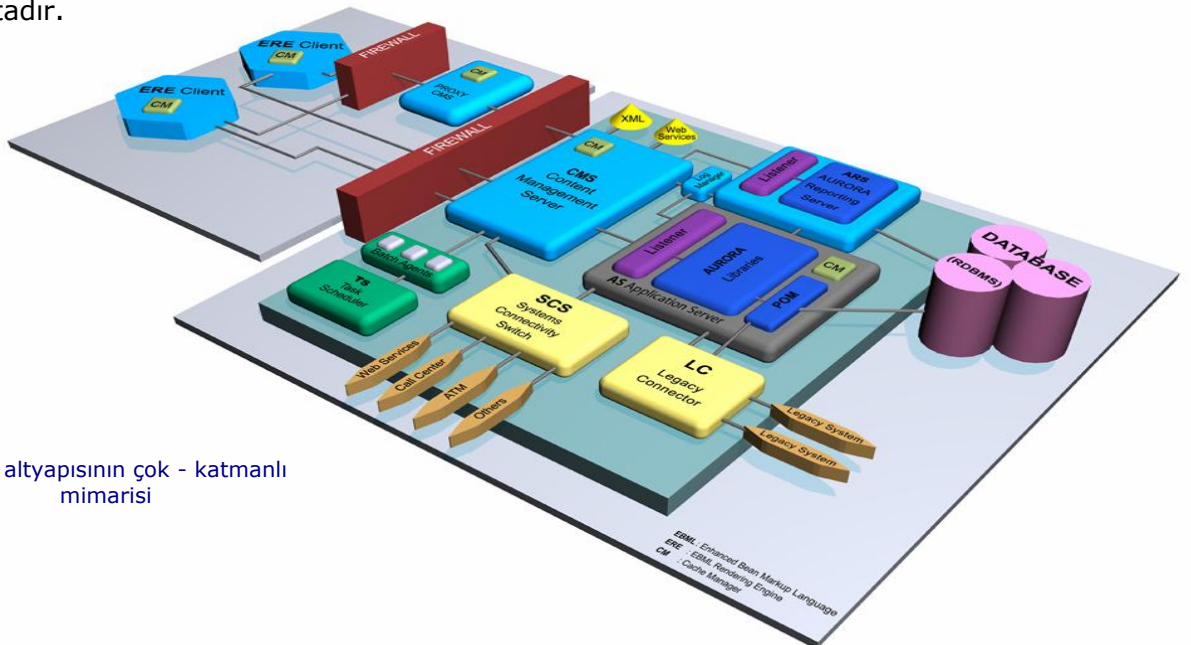
EBML Proxy Sunucusu (EBML Proxy Server - EPS), bu optimize edilmiş önbellek kullanımı mekanizmasının direkt bir çıktısıdır. EPS, hem istemciler, hem de sunucular üzerinde kullanılarak LAN'da kurulu birçok istemciyi Web'e bağlayan küçük bir araçtır. Uygun olarak konfigüre edildiğinde, genel bir EPS'in kontrol ettiği LAN'a bağlı her bir istemcinin, kendi önbelleklerini ayrıca yönetmeye ihtiyaçları kalmamaktadır.

Listeler ve raporlar

Web uygulamalarında liste ve raporların gerekmesi kaçınılmaz bir ihtiyaçtır. Ama, Web Browser'lar bu ihtiyaca cevap vermek aşamasında yeterince kullanışlı değildir. Varsayılan görselleştirme motorlarıyla browser'lar, özel metin yazıcı sürücülerini işletim sistemlerine tanıtılmadığı sürece, her şeyi grafiksel terimlerle yorumlarlar. Bu durum, geniş listelerin ve kapsamlı raporların dökümü işlerini ciddi olarak yavaşlatmaktadır.

Değişik birçok rapor formatının (PDF, Word, XML, HTML, vb.) görselleştirilmesi ve yazdırılması ise başlı başına ayrı bir sorun teşkil etmektedir. Bu noktada, Aurora'nın tamamen farklı, EBML yaklaşımına benzeyen bir yaklaşımı mevcuttur. Aurora tasarımcıları **URML (Universal Reporting Markup Language)** dilini geliştirmişlerdir. **URML**, Sunum Yönetimi Katmanı tarafından servis edilmekte olan liste ve raporların transfer edilmesi, görselleştirilmesi ve yazdırılması için kullanılmaktadır.

ERE'ye benzer olarak, **Evrensel Raporlama Aracı (Universal Reporting Agent - URA)**, rapor formatlarını ayırt edebilmekte, bunları istendiği şekilde görselleştirip ve yazım için en uygun teknolojiyi (metin veya grafik) seçebilmektedir. Ayrıca, standart-dışı dökümün söz konusu olduğu durumlarda, URA, istemciler arasındaki yazıcı paylaşımını da yönetebilmektedir.



Aurora altyapısının çok - katmanlı mimarisi

AURORA e-İş Platformu

Aurora Sunum Yönetimi Katmanı

Aurora'nın Sunum Yönetimi Katmanı, hızlı sorgulama ve raporlama ihtiyaçlarına cevap vererek, sunum ve uygulama katmanları arasında bir köprü olarak işlev görmektedir.

Sunum Yönetimi Katmanı iki ana bölümden oluşur. **Sunum Yönetimi Sunucusu (Content Management Server - CMS)**, bilinen Web sunucuları gibi, ama onlardan çok daha performanslı olarak işlev görmektedir. CMS, Aurora mimari tasarımının kalbi olduğundan, onun geliştirilmesinde özel bir itina gösterilmiştir. Aslında bu itina bir zorunluluk olarak adledilebilir; çünkü, tüm mimarinin genel performans ve kesintisiz çalışabilmesi direkt olarak CMS'yle bağlantılıdır.

Kullanıcıların yetkilendirilmesi CMS'nin temel fonksiyonları arasındadır. Bu yetkilendirme sonucunda, kullanıcı adı ve parolası, kullanıcı oturumunu temelini oluşturur. Ama, bu genel yetkilendirme fonksiyonları, kurumların spesifik ihtiyaçlarına göre kolaylıkla ayarlanabilir. CMS, kullanıcı bilgilerini önbelleğe almak üzere konfigüre edilmiştir. Yetkilendirme amacıyla veritabanı erişimi, sadece, önbellekteki kullanıcı bilgilerinin yeni girişi ve/veya güncellenmesi durumlarında söz konusudur. **Önbellek Yöneticisi** kullanıcı bilgilerinin güncellenmesi işlemlerini CMS adına koordine eder.

Geçerli bir kullanıcı adı - parola bilgisini aldığı anda, CMS, ilişkilendirilmiş menü istemciye EBML formatında gönderir; istemcide kurulu olan ERE, bu menüyü görselleştirir. Sunucu - istemci iletişimindeki bir diğer gelişme de versiyonlama bilgisinin kullanılmasıdır. CMS, nesnelerin versiyon bilgilerini tutar. Böylece, sadece yeni versyonlar istemciye iletilir. Dahası, referans veriler de CMS'de önbelleğe alınmaktadır. Dolayısıyla, bu tür statik verilerin alınması için ayrıca veritabanı bağlantısı kurulmasına gerek kalmamaktadır. Önbellek, CM tarafında oluşturulmakta ve güncellenmektedir.

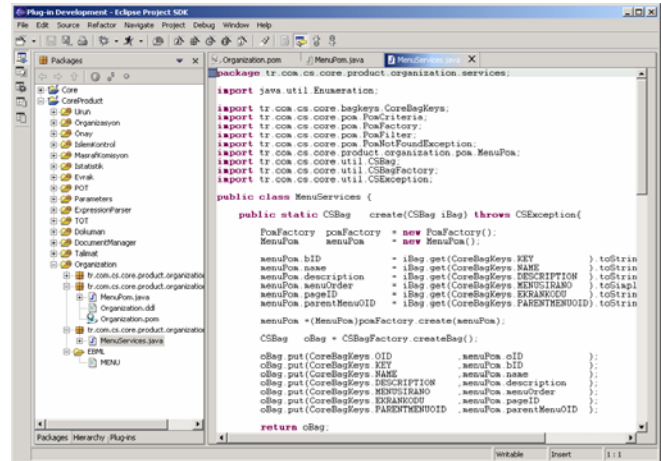
Yük dengeleme, CMS ve Uygulama Sunucuları arasındaki bağlantılar üzerinden yürütülmektedir. Ayrıca bu bağlantılar üzerinden yapılan servis çağırma işlevlerine

ilişkin istatistikler oluşturulmaktadır. Politika olarak, bir sonraki talep, o anda en az yüklü olan uygulama sunucusuna yönlendirilmektedir. Uygun olarak konfigüre edildiğinde, CMS, Web anahtarlarından da yardım alarak, global ağ yük dağılımının ağa bağlı birçok bilgisayarda CMS modülleri tarafından gerçekleştirilmesini sağlamaktadır.

Raporlama, CMS'nin ikinci kritik işlevidir. Raporlamanın, Web uygulamalar için bir dış bileşen olması tavsiye edilmektedir. Bu sayede, rapor geliştirme işlevleri, gerçek uygulama geliştirme işlevlerinden ayrı olarak ele alınabilmektedir. Aurora'da, raporlar ya direkt erişimle veritabanından, ya da CMS aracılığıyla uygulama sunucusunun servislerinden oluşturulabilmektedir.

Aurora Uygulama Katmanı

Uygulama Katmanı, sadece iş kodlarının yürütüldüğü platform olarak düşünülmelidir. Bu, sunum yönetiminin iş mantığı ve verilerin devamlılığı arasında kesin bir ayrımı ifade etmektedir.



```
import java.util.Enumeration;
import tr.com.cs.core.bagkeys.CoreBagKeys;
import tr.com.cs.core.poa.PoaFactory;
import tr.com.cs.core.poa.PoaFilter;
import tr.com.cs.core.poa.PoaNotFoundException;
import tr.com.cs.core.product.organization.PoaMenuPoa;
import tr.com.cs.core.util.CSBag;
import tr.com.cs.core.util.CSBagFactory;
import tr.com.cs.core.util.CSException;

public class MenuServices {

    public static CSBag create(CSBag iBag) throws CSException {
        PoaFactory poaFactory = new PoaFactory();
        MenuPoa menuPoa = new MenuPoa();

        menuPoa.bid = iBag.get(CoreBagKeys.KEY) } toStrin
        menuPoa.name = iBag.get(CoreBagKeys.NAME) } toStrin
        menuPoa.description = iBag.get(CoreBagKeys.DESCRPTION) } toStrin
        menuPoa.menuOrder = iBag.get(CoreBagKeys.MENUORDER) } toStrin
        menuPoa.pageID = iBag.get(CoreBagKeys.EKIBANCO) } toStrin
        menuPoa.parentMenuID = iBag.get(CoreBagKeys.PARENTMENUID) } toStrin

        menuPoa = (MenuPoa)poaFactory.create(menuPoa);
        CSBag oBag = CSBagFactory.createBag();

        oBag.put(CoreBagKeys.OID, menuPoa.oID);
        oBag.put(CoreBagKeys.KEY, menuPoa.bid);
        oBag.put(CoreBagKeys.NAME, menuPoa.name);
        oBag.put(CoreBagKeys.DESCRPTION, menuPoa.description);
        oBag.put(CoreBagKeys.MENUORDER, menuPoa.menuOrder);
        oBag.put(CoreBagKeys.EKIBANCO, menuPoa.pageID);
        oBag.put(CoreBagKeys.PARENTMENUID, menuPoa.parentMenuID);

        return oBag;
    }
}
```

İş Mantığı Java'da kodlanabilmektedir, Aurora'da bu amaçla hazırlanmış Eclipse plug-in'ler mevcuttur.

Aurora Platformu'nun tasarımcıları, iş mantığını oturtmak üzere Java'yı kullanmışlardır. Çünkü Java, genel yaklaşımlar sunması ve platform bağımsız olmasıyla, güvenilir bir geliştirme ve çalıştırma tabanı sunmaktadır.

AURORA e-İş Platformu

Kurumsal Web uygulamaların en önemli kritik başarı faktörlerinden biri de, iş mantığının, uygun bileşenlere (nesne kümelerine), hem bileşen içi ilişkileri maksimize edebilecek şekilde (tutarlılık), hem de bileşenler arası etkileşimleri minimize edebilecek şekilde (bağlaşım) bölümlendirilmesine olanak sağlamaktır. Aurora, iş ihtiyaçlarını etkin bir biçimde karşılamak üzere optimum büyüklüğü hedefleyen bir yaklaşıma sahiptir.

Aurora SOA ile uyumludur

Aurora'da her bir bileşen, *adlandırılmış servisler* ve özelleşmiş *girdi ve çıktı kümeleri* anlamında bilinen arayüzler sunmaktadır. Servis adı ile girdi ve çıktı parametreleri, tümüyle önceden tanımlanmakta ve merkezi bir havuz tarafından kontrol edilmektedir. Aurora uygulama katmanının parçası olan her bir JVM, bu tanımları, statik Java sınıflarının daha sonraki sorgulama işlevlerini hızlandırmak amacıyla, bu merkezi havuzdan önceden indirmektedir.

Nesnelerin sıralaması için, Aurora Platformu'nun tasarımcıları, hiyerarşik, kendi kendini ifade edebilen ve hızlı erişim sağlayabilen *Bag* sınıflarını geliştirmişlerdir. Bag sınıfları, tekdüze erişimi olanaklı kılmak adına kendilerini sıralayabilirler ve sıralamalarını bozabilmektedir.

Aurora herhangi bir veri kaynağını kullanabilir

Aurora, kurumsal Web uygulamaları için gelişmiş bir yaklaşıma sahiptir. Bu bağlamda, kapalı sistemlerin de içinde olduğu başka bilgi sistemlerine sorunsuz entegrasyonlar sunabilmektedir.

Bu noktada, Aurora Platformu'nun tasarımcıları tarafından, Sistem Bağlanırlık Anahtarı (**System Connectivity Switch - SCS**) olarak adlandırılan ayrı bir modül tanımlanmıştır. Bu modül, uygulama katmanı ve varolan bilgi kaynakları arasında entegrasyon sağlamaktadır.

Aurora platformunun bir parçası olarak, kapalı sistem entegrasyonlarının da dahil olduğu global işlem yönetimi, JCA'nın (Java Connector Architecture) talep ettiği XA-

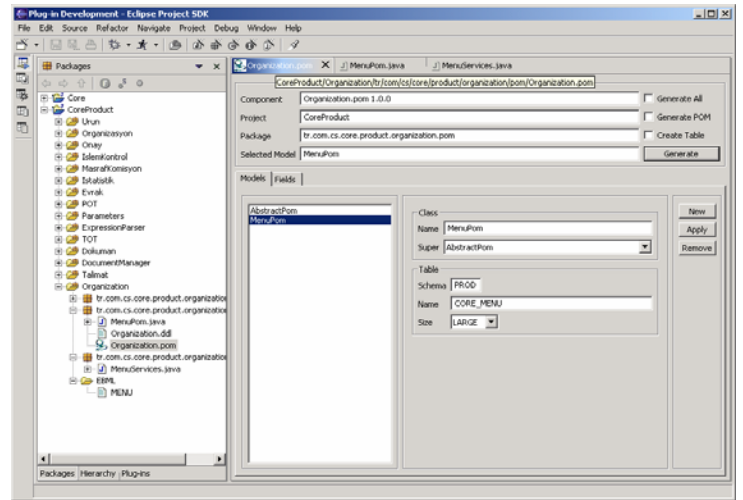
uyumlu araçların kullanılmasıyla, tamamen SCS tarafından yürütülmektedir.

Aurora Veri Katmanı

Aurora veri katmanı, JTS (Java Transaction Server) teknolojisi tarafından yönetilen XA-uyumlu bir İVYTS sunucudan oluşmaktadır.

Veri katmanında, Kalıcı Nesne Modeli (Persistent Object Model - POM) sınıfları, geliştiriciler tarafından, POM Studio (POM'ları yönetmek için kullanılan bir GUI ortamı) kullanarak tanımlanabilmektedir. POM Studio kullanarak, gerekli betikler, ilişkili DDL şemaları ve ilgili statik Java sınıfları, bu tanımlardan otomatik olarak üretilebilmektedir.

Aurora, RDBMS servislerini, anlamlı O2R adresleme katmanı (POM) aracılığıyla yönetmektedir. POM ayrıca, kapalı sistemlerle XA-uyumlu araçlar üzerinden veri bağlantılarının kurulmasına yardım etmektedir.



POM Studio, üzerinde tüm POM hiyerarşisini tanımlayabildiğiniz bir Eclipse plug-in'dir.

AURORA e-Business Platform

Cybersoft Enformasyon Teknolojileri

Merkez

Birlik Mahallesi
14. Sokak
No: 6/2 Çankaya
Ankara / Türkiye
Tel. +90 312 496 2057
Fax +90 312 496 2059

Araştırma & Geliştirme Ofisi

ODTÜ Teknokent
Silikon Blokları
Kat:1 No:18 ODTÜ
Ankara / Türkiye
Tel. +90 312 210 1940
Fax +90 312 210 1752

İstanbul Ofisi

Ataşehir Bulvarı
Ata 3/3 Plaza
Kat:3 Ataşehir
İstanbul / Türkiye
Tel. +90 216 455 5595
Fax +90 216 455 5079

www.cybersoft.com.tr

cybersoft@cs.com.tr