



YMT 312-Yazılım Tasarım Ve Mimarisi

Yazılım Mimarileri

Fırat Üniversitesi Yazılım Mühendisliği Bölümü

Bölüm-13

Bu Haftaki Konular

İstemci Sunucu Mimarileri.....5

Sunucu Türleri.....11

WWW Standartları.....20

Katmanlı Mimariler.....21

WAP Mimarisi.....36

Amaçlar

- İstemci ve Sunucu Arasındaki Farkları Kavramak
- Sunucu Türlerini Öğrenmek
- Web Sunucusu Yapısını Kavramak
- WWW Standartlarını Öğrenmek
- Yazılım Geliştirmede Katmanlı Mimari Yapılarının Avantajlarını Görmek
- Katmanlı Mimarilerin Neler Olduğunu Öğrenmek
- İki Katmanlı ve Üç Katmanlı Mimari Arasındaki Farkları Tartışmak
- WAP Mimarisi Hakkında Bilgilenmek



GİRİŞ

- 1980'li yılların ortalarına kadar ana bilgisayarların yoğun olarak kullanıldığı dönemde, yazılımların hemen hemen tümüyle ana bilgisayarda olduğu, kullanıcıya ilişkin tüm denetim ve paylaşım işlemlerinin yine ana bilgisayarlar üzerinde yapıldığı gözlenmektedir.
- Grafiksel kullanıcı arabirimi desteklenmemekte ve "akılsız" uçlar kanalıyla bilgisayara bağlantı yapılabilmektedir.
- Programlamanın günümüze göre daha kolay olduğu bu yapılarda, coğrafik olarak birbirinden ayrı bulunan veri tabanlarına erişim ve işlem oldukça sınırlı olmuştur.
- "Ana çatı" ya da "ana bilgisayar" mimarileri olarak adlandırılan bu mimarileri, 80'li yılların ortalarında "kütük paylaşım mimarileri" izlemiştir.
- Bu mimarilerde ise bilgiler bir sunucu üzerinde bulunmakta ve kullanıcılar kendi mikrobilgisayarları üzerinde iş yaparak bu bilgileri paylaşmaktadır.
- Gerek "ana bilgisayar" gerekse "kütük paylaşımı" mimarileri günümüzde artık kullanılmamakta ve "geleneksel" mimariler olarak adlandırılmaktadır. Bu nedenle bu bölümde bu mimarilere yer verilmemiştir.
- Günümüzde yaygın olarak kullanılan istemci/sunucu mimarileri ve yeni teknolojiler olan wap mimarisi ve bileşen tabanlı mimariler bu bölümde daha geniş olarak incelenmektedir.

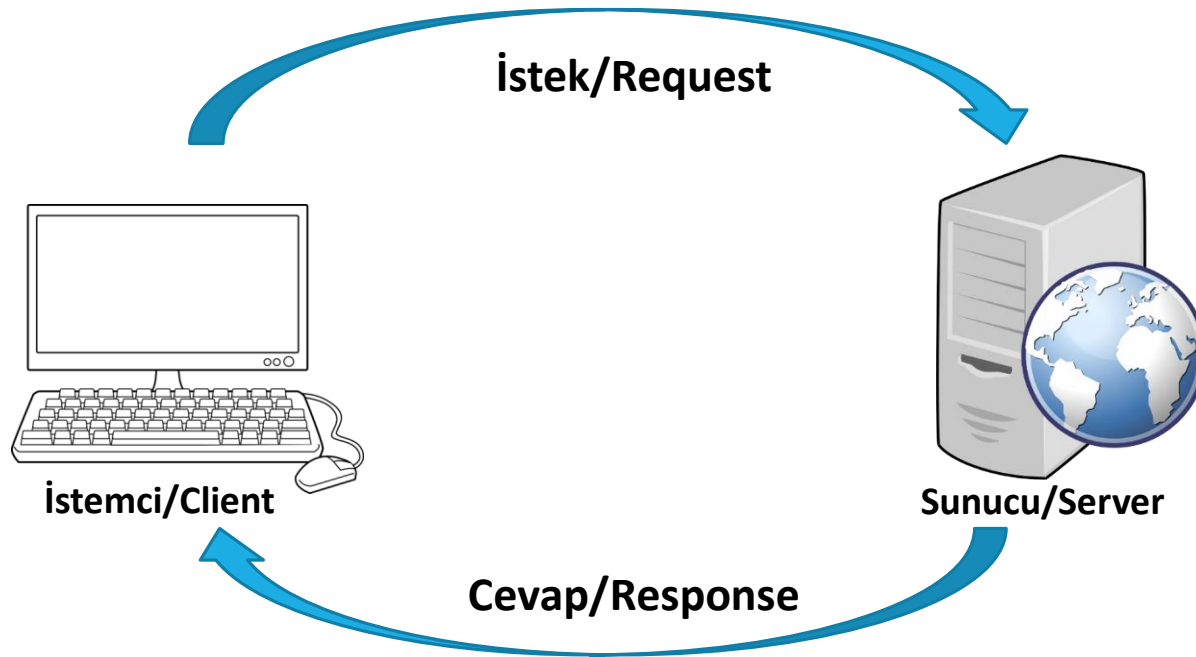
İstemci Sunucu Mimarileri

- Dağıtık bilgi işleme ortamlarının gelişmesinin etkisiyle, istemci sunucu (İ/S) mimarileri yaygınlaşmıştır.
- İstemci sunucu mimarisinde, yerel alan ağları, geniş alan ağı dağıtık veri işleme ve yönetim ve açık sistem ortamları bulunmaktadır.
- Yalın tanımıyla İ/S mimarisinde, sunucu yazılımı, istemci yazılımdan gelen istekleri yerine getirerek sonucu istemciye iletir.
- İstemci sunucudan gelen sonuçları değiştirerek ya da olduğu gibi kullanıcıya sunar.
- İ/S mimarilerinde temel etmen, donanım değil yazılımdır.
- Geleneksel mimarilerin tersine, uygulama yazılımı, istemci ve sunucu bilgisayarlarda çalıştırılmak üzere her iki platform için geliştirilir.
- İstemci ve sunucu arasındaki veri iletişimi ve denetim ilişkisi yazılım aracılığı ile yönetilir.
- İ/S mimarileri değişik yapılarda ve ölçeklerde oluşturulabilir.

İstemci Sunucu Mimarileri

- Küçük bir yerel alan ağı üzerinde tek sunucu ile çalışabilecek İ/S mimarileri oluşturulabileceği gibi, binlerce kullanıcıyı yüzlerce sunucuya bağlayan ve geniş alan ağları üzerinde çalışan e-posta servislerini içeren mimariler oluşturulabilir.
- Ölçeği ne olursa olsun tüm istemci sunucu mimarisindeki yazılımların ortak özellikleri aşağıda belirtilmektedir.
 - Kullanıcılar genel olarak bir yerel ağ üzerinde bulunan sunucularla çalışırlar. Dolayısıyla tipik bir İ/S mimarisi varsa geniş alan ağına iletişim sunucusu ile bağlı bir yerel ağ üzerinde kurulur.
 - Standart bir İ/S mimarisi yoktur, ancak standart İ/S bileşenleri vardır.
 - Sunucular genellikle belli işlere atanmış güçlü iş istasyonlarından (örneğin kütük sunucuları, yazıcı sunucuları vb) oluşturulur.
 - İstemciler ise kişisel bilgisayar sınıfındaki bilgisayarlardır. El bilgisayarları, akıllı telefon gibi istemciler giderek yaygınlaşmaktadır.
 - Tipik yerel ağ bileşenleri ise, köprüler, yönlendiriciler, alıcılar, öbekler ve geçitlerden oluşur.
 - Genellikle her sunucu için gerekli kullanıcı sayısı 10 ile 50 arasında değişir.
 - Daha geniş ağları oluşturmak amacıyla alt ağlar oluşturma biçiminde tabakalama yapılır.

İstemci Sunucu Örnek



İstemci

- İstemci, sunucuya bir iş yapmak ya da servis sağlamak üzere ileti gönderen bir süreç ya da yazılım olarak tanımlanabilir.
- İstemci programları genellikle uygulamanın kullanıcı arayüzünü yönetirler.
- Kullanıcı tarafından girilen verinin doğrulanması ve kullanıcı isteğinin sunuculara ayrıştırılması gibi işlemler İstemci yazılımları aracılığı ile yapılır.
- Özetle, bir uygulama yazılımının kullanıcı arayüzü ile ilgili yazılım parçaları istemciler üzerinde çalışır.
- İ/S mimarisinin gelişmesi, kullanıcıya, kişisel bilgisayar tabanlı istemciler kullanmasından dolayı grafiksel arayüz olanakları sağlamıştır.

Sunucu

- İstemci tarafından gerekenen isteği yerine getiren süreç ya da yazılımlar sunucu yazılımları olarak adlandırılır.
- Sunucu, istemci isteğini alır, gerekli işlemleri yapmak amacıyla veri tabanlarında erişim ve güncleme yapar.
- Sonucu istemciye iletir. Sunucu yazılımları ağ üzerinde bir başka bilgisayarda çalışır.
- Sunucu, bir ana bilgisayar işletim sistemi, kütük sunucusu ya da uygulama sunucusu olabilir.
- Sunucular, uygulamanın arka yüzdeki işlemlerini yapar.

Sunucu

- Tipik bir sunucu yazılımının özellikleri aşağıda belirtilmektedir.
 - Sunucu yazılımı zamanının büyük bir kısmını istemciden gelecek istekleri beklemekle geçirir. Söz konusu istekler, iletişim oturumu aracılığı ile ve ileti biçiminde ulaştırılır. Bazı sunucular her istemci için ayrı oturum atarken, diğerleri de tekrar kullanılabilir oturumlar için dinamik atama yaparlar. Sunucu için başarılı olmanın ön koşulu, istemcilere her zaman yanıt verebilmesidir.
 - Sunucu, istemcinin isteğini yerine getirmek durumundadır. Ancak istemci yalnızca bir sunucuya bağlı kalmayabilir. Öte yandan bir sunucu da eş zamanlı olarak birden fazla istemciye servis verebilir.
 - Sunucular, istemcilere değişik düzeylerde servis öncelikleri verebilir.
 - Sunucu, uygulama yazılımının içeriğinden farklı olarak, arka planda başka işlemler yapar.
 - Sunucu yazılımı tipik bir görev-kritik yazılımdır. Sunucu durduğunda, bağlı tüm istemciler etkilenir. Bu nedenle sunucu yazılımının çalışacağı ortam çok güvenli olmalıdır.
 - Sunucu ortamı, ölçeklenebilir, yükseltilebilir ve modüler olmalıdır.

Sunucu Türleri

➤ Sunucular için farklı sınıflandırmalar yapılabilir.

➤ Bu bölümde:

1. Kütük Sunucuları
2. Veri Tabanı Sunucuları
3. Ara İşlem Sunucuları
4. Nesne Sunucuları
5. Web sunucuları

biçiminde bir ayırım verilmektedir.



Kütük Sunucuları

- Bilgisayar ağı üzerindeki tüm kütüklerin paylaşımı, kütük sunucuları aracılığı ile yapılır.
- Veri servisinin en temel biçimi, istemcinin isteğini kütük sunucu aracılığı ile ağa iletmesi ve sunucunun isteği inceleyerek ağ üzerinde ileti değişimi yoluyla istenen veriye ulaşmasının sağlanmasıdır.
- Yerel ağ üzerindeki her türlü belge, resim, veri kütüğü vb. nesnelerin paylaşımı kütük sunucuları aracılığı ile yapılır.

Veri Tabanı Sunucuları



- İstemci bir SQL komutu ya da komut kümesini veri tabanı sunucusuna iletir.
- Her SQL komutunun sonucu, ağ üzerinden istemciye iletilir. SQL isteğini işleyen kod aynı bilgisayarda bulunur.
- Sunucu, ileti aktarma işlemleri ile uğraşma yerine, işlem gücünü veri tabanından istenen verilerin bulunması ve işlenmesi işlemleri ile uğraşır.
- SQL sunucuları İ/S uygulamaları oluşturmak için iyi bir örnektir.
- Oracle, Sybase gibi bir çok veri tabanı yönetim sistemi geliştiricileri için SQL standart veri tabanı dili haline gelmiştir.

Ara İşlem Sunucuları

- İstemcinin uzakta bulunan yordamları çağırması durumunda bu yordamların çalıştırıldığı sunucular ara işlem sunucu olarak adlandırılır.
- Uzakta bulunan yordamlar bir grup SQL deyimi çalıştırırlar. Bu SQL deyimleri, ara işlem olarak tanımlanır.
- Ara işlem sunucu ile, hem istemci hem de sunucu tarafında yazılımlar hazırlanarak uygulama geliştirilebilir.
- İstemci tarafındaki yazılım genellikle grafiksel kullanıcı arabirimi içerir.
- Sunucu yazılımı ise bir veritabanı üzerinde çalışacak SQL deyimlerinden oluşur.
- Bu uygulamalar **çevrim içi ara işlem işleme (ÇİAİ-OLTP)** olarak adlandırılır.
- ÇİAİ uygulamaları, veri tabanı üzerinde sıkı denetim, güvenlik ve tamlık gerektirir. İki tür ÇİAİ uygulaması vardır:

Ara İşlem Sunucuları

- Kritik ÇİAİ uygulamalarını etkin ve iyi yönetebilmek için ara işlem yöneticileri kullanılır.
 - Ara işlem yöneticisi, ara işlemlerin yönetimi ve işlenmesi için bir "işletim sistemi" olarak tanımlanabilir.
 - Temel olarak;
 - Süreç yönetimi (sunucu süreçlerinin başlatılması, yönetimi ve iş yükünün dengelenmesi) ve
 - Ara işlem yönetimi
- işlevlerini yerine getirirler.

Nesne Sunucuları

- İ/S uygulaması, birbiri ile iletişimde bulunan bir nesne kümesi olarak geliştirilir.
- İstemci nesneleri, sunucu nesneleri ile bir Nesne İstek Yöneticisi (NİY - ORB) aracılığı ile haberleşir.
- İstemci, uzakta bulunan bir nesneye ilişkin bir metodu çalıştırmak istediğinde NİY, ilgili nesne sınıfına ilişkin varlık oluşturur, istenen metodu çağırır ve sonucu istemci nesnesine iletir.
- Sunucu nesneleri, paylaşım ve eşzamanlılığı desteklemelidirler.

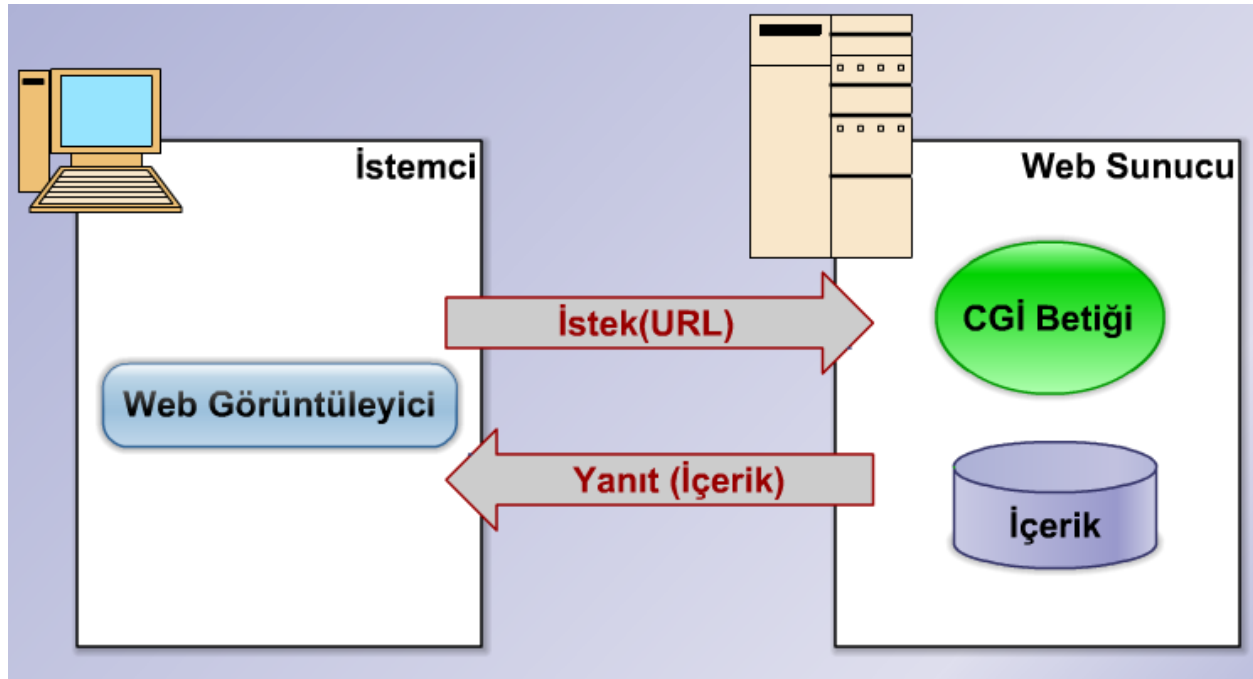
Ticari NİY'lere örnek olarak Digital ObjectBroker, IBM SOM 3.0, Sun's NEO, HP ORB Plus, Expersoft's PowerBroker, ve Iona's Orbix verilebilir.

Web Sunucuları



- Dünya çapındaki Web, ilk galaksiler arası İ/S uygulaması olarak tanımlanabilir.
- Bu yeni model zayıf, taşınabilir ve "**evrensel**" istemcilerin çok şişman sunucularla iletişimini sağlar.
- En yalın biçimiyle bir web sunucu, istemcinin ismiyle aradığı bir belgeyi kendisine sunar.
- Web, küresel bir hipertekst sistemidir.
- Hipertekst, aynı bilgisayar yada bir bilgisayar ağı üzerinde bulunan ilgili belgeleri birbirine bağlayan bir yazılım sistemidir.
- İstemci ve sunucular, adına Hypertext Transfer Protokolü (HTTP) denilen Uzak Yordam Çağırma (UYÇ -RPC Remote Procedure Call) benzeri bir protokol aracılığı ile iletişim kurar.

Web Sunucuları



Web Sunucuları

HTTP:

1. İ/S bağlantısı kurar,
 2. Parametreleri gönderir ve alır,
 3. İ/S bağlantısını kapatır.
- HTTP istemci ve sunucuları ileti içeriklerini göstermek amacıyla internetin MIME veri gösterim yöntemini kullanır.



WWW Standartları



- Genel amaçlı uygulamaların geliştirilmesi amacıyla oluşturulan www standartları geliştirilmiştir.
- Bu standartlar şu şekildedir;
 - **Standart isimlendirme Modeli:** Bütün sunucuların isimlendirilmesi ve içeriği için internet Tek Kaynak Yerleştirici (URL-Uniform Resource Locator) standardı (RFC1738,RFC1808)
 - **İçerik Türü:** WWW üzerindeki her tür içerik, web görüntüleyicilerin işleyebileceği özel bir türe sahiptir (RFC2045, RFC2048)
 - **Standart İçerik Biçimi:** Bütün web sunucuları bir dizi standart içerik biçimini destekler. Bunlar arasında Hiper Metin Biçimlendirme Dili (HTML-Hypertext Markup Language), JavaScript betik dili(ECMA Script, Javascript) en yaygın olanlarıdır.
 - **Standart Protokol:** Web görüntüleyicilerin herhangi bir web sunucu ile iletişimini sağlayan ağ protokol standartları arasında en yaygın olan, HiperText Transfer Protokolüdür (HTTP).

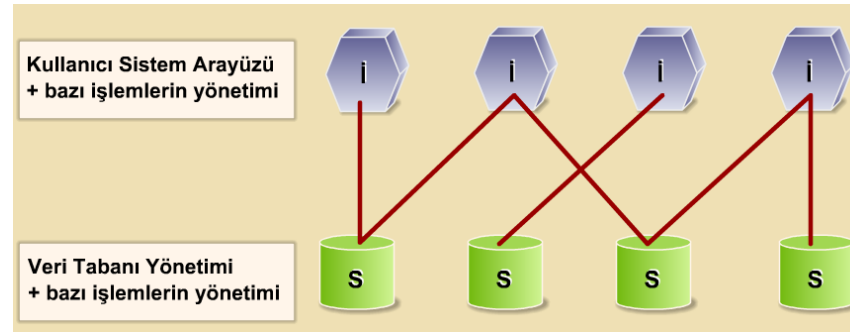
İki Katmanlı Mimariler

➤ İki katmanlı mimarilerde, uygulamanın üç temel bileşeni iki katmanda bulunur.

➤ Bu bileşenler:

- Kullanıcı Arayüzü (Grafiksel Kullanıcı Arabirimi, görüntü yönetimi),
- İşlem yönetimi (ya da uygulama mantığı) ve
- Veritabanı yönetimi (veri ve kütük servisleri)

biçimindedir.



İ: İstemci

S: Sunucu

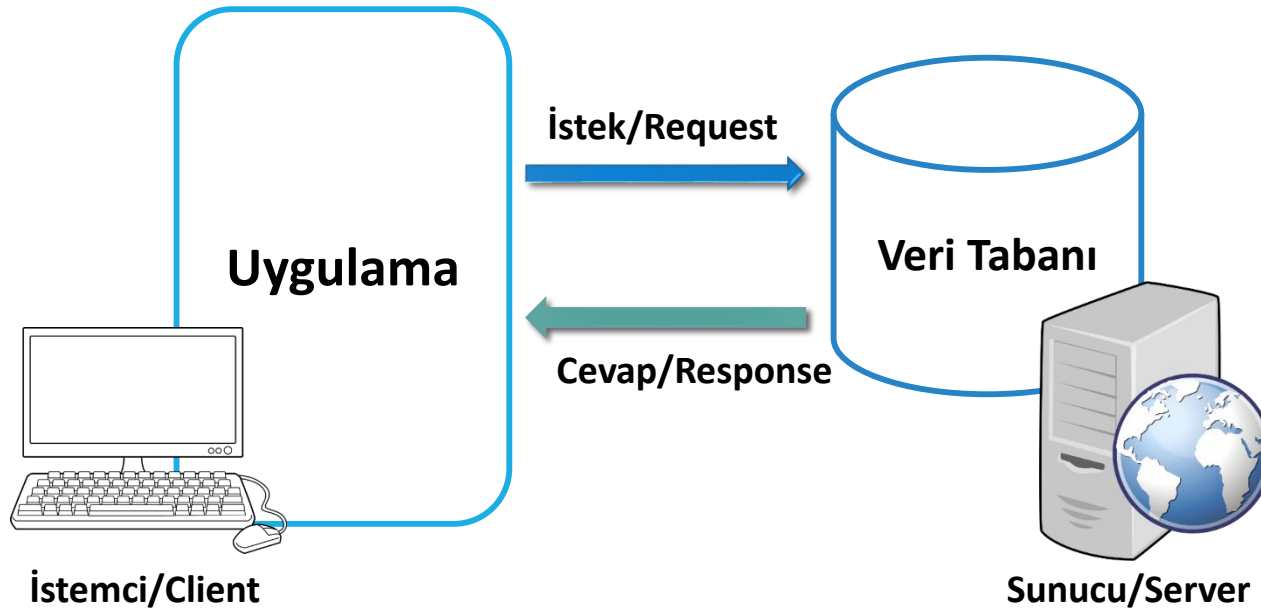
İki Katmanlı Mimariler

- Kullanıcı arayüzü istemci üzerinde bulunur. Veri tabanı yönetimi ise sunucu üzerine yerleştirilir. İşlem yönetimi ya da uygulama mantığı ise istemci ve sunucu arasında paylaşılır.
- Birçok iki katmanlı tasarımda, uygulamanın büyük bir kesimi sunucu ortamında bulunur. Bu yolla esneklik ve ölçeklenebilirlik arttırılır.
- Bu mimari, homojen ortamlarda başarılı sonuçlar verir.
- İki katmanlı mimariler, 4. kuşak programlama platformlarının da desteği ile, günümüzde, endüstride oldukça yaygınlaşmış ve başarılı bir biçimde uygulanmaktadır.

İki Katmanlı Mimariler

- İki katmanlı mimari, bilgisayar ağı ve sunucunun yeteneklerine göre belirli bir sınıra kadar ölçeklenebilir.
- 100 kullanıcıya kadar gözlemlenmiş uygulamalar bulunmaktadır. Bunun nedeni, bilgisayar ağının belirli bir yoğunluğa ulaşması ve sunucu üzerindeki işlem yoğunluğudur.
- Hiçbir veri değişimi olmazsa bile yalnızca ağ üzerinde dolaşan "istemcileri canlı tutma" iletileri belirli bir yoğunluğa neden olur.
- Sunucu üzerinde işlem yapan her istemci bir darboğaz oluşturmaya neden olabilir.
- Veri tabanı sunucusu üzerinde geliştirilen uygulama mantığı, birlikte çalışılabilirliği azaltır. Çünkü kod, ilgili VTYS dili ile yazılmıştır ve ticari veri tabanları için tanımlanmış standart bir VTYS dili yoktur. Bu nedenle kod bakımı zorlaşır.
- Öte yandan bu diller, standart programlama dilleri kadar yetenekli değildir. İstemci tarafında, sunucuya erişim amacıyla, SQL ya da UYÇ kullanılır.
- Yapılacak her değişikliğin tüm istemcilere yansıtılması gerekliliğinden dolayı, istemci tarafının yönetimi ve bakımı kolay değildir. Ayrıca, bu mimari toplu iş düzeni için uygun değildir.

İki Katmanlı Mimari Örnek

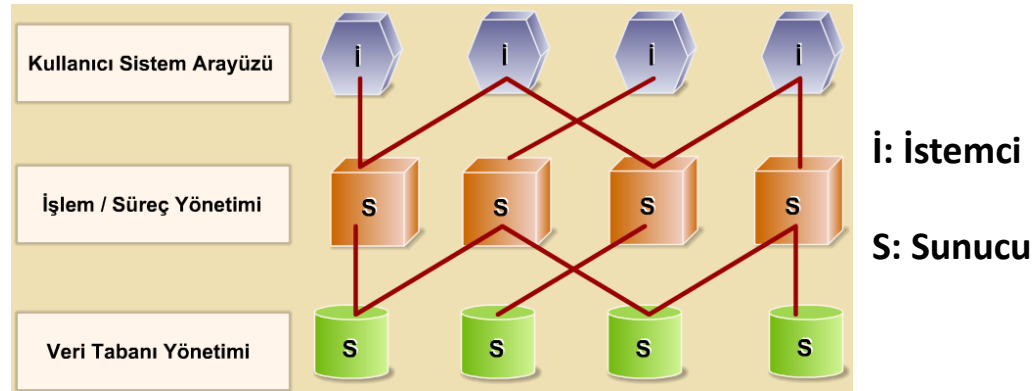


Üç Katmanlı Mimariler -1

Üç katmanlı mimarilerde, uygulamanın üç temel bileşeni, üç bağımsız katmanda bulunur.

Bu bileşenler:

- Kullanıcı (sistem) Arayüzü,
- İşlem Yönetimi (Uygulama Mantığı) ve
- Veri tabanı servisleri



Üç Katmanlı Mimariler -2

- Üç katmanlı mimarilerde veri sunucusu yalnızca veritabanı bilgilerini içerir ve uygulama mantığı ile ilişkisi yoktur.
- Bu nedenle uygulama mantığına ilişkin herhangi bir VTYS dili kodlaması bu sunucu için gerekmez.
- Bu mimarilerde ara katman sunucusu olan uygulama sunucusu, uygulama mantığını tümüyle içerdiğinden, bakım kolaylığı, esneklik ve ölçeklenebilirlik sağlanır.
- Değişiklik yönetimi de kolaylaşır. Uygulama sunucusu üzerinde yapılan değişiklikler istemcileri etkilemez.
- Buna ek olarak, üç katmanlı mimariler güvenilir işlem yapma olanağı sağlar.
- Üç katmanlı mimarilerde, iki katmanlı mimarilerin tersine yüzlerce kullanıcının desteklenmesi olasıdır. Heterojen yapılar için uygundur. Bu mimari 90'lı yılların başından bu yana kullanılmaktadır.

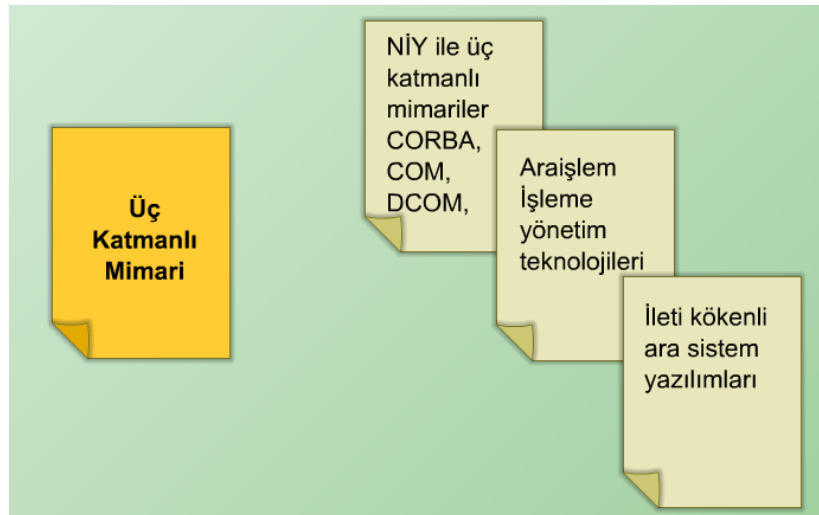
Üç Katmanlı Mimariler -2

- En büyük zorluklardan biri, yeterince destekleyen yazılım aracının bulunmayışı ve karmaşıklığıdır.
- Günümüzde geliştirilmiş olan yazılım geliştirme yöntem ve metodoloji çoğu iki katmanlı İ/S sistemlerine yöneliktir.
- Diğer bir zorluk ise, bu üç katmanın ayrıştırılmasında oluşmaktadır. Bu konuda kullanılabilecek ayrıştırma kriterlerinin bazıları:
 - Geliştirme ve sınama kolaylığı,
 - Yönetim kolaylığı,
 - Sunucuların ölçeklenebilirliği ve
 - Performans

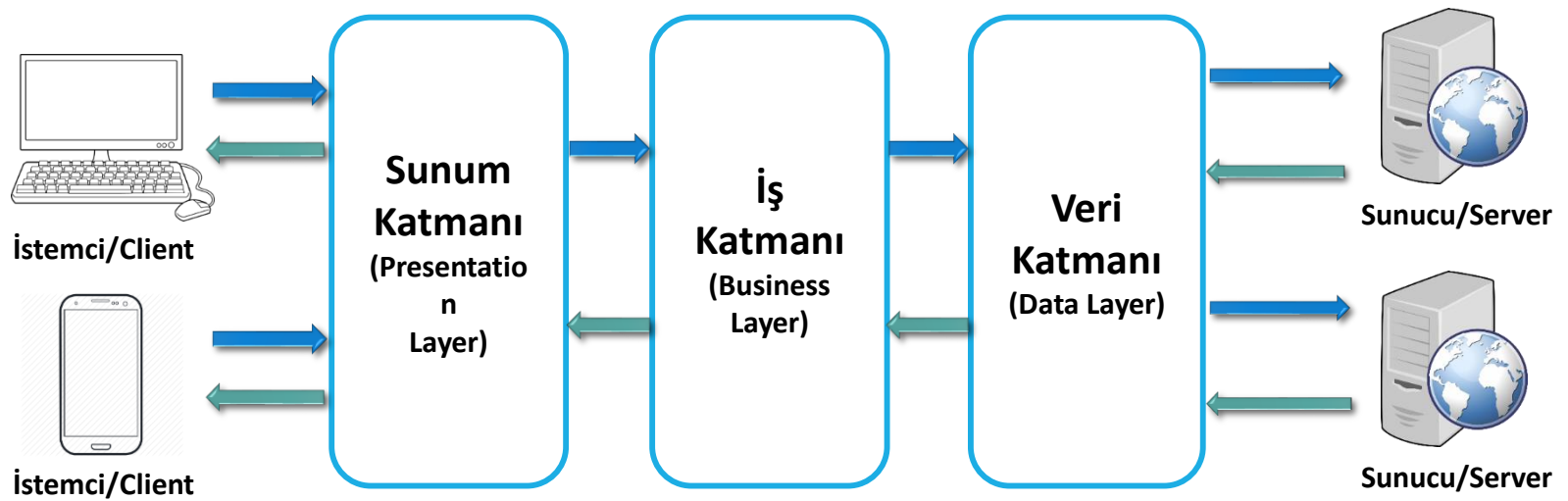
olarak verilebilir.

Üç Katmanlı Mimariler -2

- Üç katmanlı mimari üzerinde geliştirilmiş alt yapı yazılımları uygulamalarına örnek olarak aşağıdakiler gösterebilir.



Üç Katmanlı Mimari Örnek



İki ve Üç Katmanlı Mimarilerin Karşılaştırması

- İki katmanlı mimarilerde, uygulama mantığı ve veri tabanının aynı bilgisayarda yer alması, sunucu tarafında geliştirilen uygulamanın VTYS'nin kendi dilinde olması nedenleriyle birlikte çalışılabilirlik azalmaktadır.
- Üç katmanlı mimarilerde ise, ara bir katmanın oluşu bu sınırlamayı kaldırmaktadır.
- Uygulama mantığının bu ara katmanda oluşu, istemciyi veri tabanı sunucusundan bağımsız duruma getirmektedir.

İki ve Üç Katmanlı Mimarilerin Karşılaştırması

Kriter	İki Katmanlı Mimari	Üç Katmanlı Mimari
Birlikte Çalışabilirlik	Sınırlı	Yüksek
Taşınabilirlik	Az taşınabilir	Daha fazla taşınabilir
Yeniden Kullanılabilirlik	Sınırlı	Yeniden Kullanılabilir
Karmaşıklık	Yalın	Karmaşık
Bakım	Zor	Kolay
Geliştirme Süresi	Daha az başlangıç geliştirme süresi	Daha az ardışık geliştirme süresi
Ölçeklenebilirlik	100 kullanıcıya kadar	1000 kullanıcıya kadar

İki ve Üç Katmanlı Mimarilerin Karşılaştırma Kriterleri

- İki katmanlı mimarilerde, **taşınabilirlik** bir bakıma sınırlıdır.
- Mevcut iki katmanlı mimari uygulamaları, yordamlara ilişkin kodları tekrar üretmeden bir başka ortama taşınamamaktadır.
- Üç katmanlı mimarilerde ise, ara katman aracılığı ile bu sorun çözülmektedir.
- Ara katman yazılımının modüler biçimde geliştirilmesi, herhangi bir sorunla karşılaşmaksızın sunucu kesiminde ya da istemci kesiminde herhangi bir zorlukla karşılaşılmaz.
- İki katmanlı uygulamalarda **yeniden kullanılabilirlik** ana hedef değildir ve oldukça sınırlıdır.
- Üç katmanlı mimariler ise değişik açılardan yeniden kullanılabilirliği sağlarlar.
- İlk olarak, ara katmanda geliştirilmiş olan uygulama mantığı yazılımı fazla değiştirilmeden başka uygulamalarda yeniden kullanılabilir. Diğer taraftan NİY modeli, nesne kökenli yaklaşımla geliştirilmiş uzak nesnelerin yeniden kullanılmasını olanaklı kılar.

İki ve Üç Katmanlı Mimarilerin Karşılaştırma Kriterleri

- **Karmaşıklık** açısından incelendiğinde, iki katmanlı mimarilerin, üç katmanlı mimarilere oranla daha yalın olduğu söylenilebilir.
- En azından bu yalınlığın nedeni üç katman yerine iki katman bulunmasıdır.
- Buna ek olarak, uygulama geliştirme iki katmanlı mimarilerde daha kolaydır.
- Üç katmanlı mimari yazılımlarının geliştirilmesine yönelik araç, yöntem ve metodolojilerin azlığı da bu mimariler için bir başka karmaşıklık nedenidir.
- **Bakım** açısından bakıldığında ise üç katmanlı mimarilerin iki katmanlı mimarilere oranla daha kolay bakılabilir olduğu açıktır.
- Bu kolaylık ara katman sayesinde sağlanır.

i/S Mimarisinin Yararları ve Aksak Yönleri

İstemci sunucu mimarisi yazılımlarının yararları aşağıda belirtilmektedir.

- Gelişen masaüstü teknolojisinin etkin olarak kullanılmasını olanaklı kılar. Bu yolla maliyetleri azaltır.
- Varolan altyapı ve aygıtları kullanarak kurumların yapılarını korur.
- İşlem ve işlenecek verinin birbirine yakın olmasını sağlayarak ağ trafiğini azaltır ve yanıt süresini kısaltır. Ağ için gerekli bant genişliği düşer.
- Anabilgisayar mimarileri yazılımlarına oranla yazılım geliştirme süresi kısalır. Önyüz ve arka yüz işlemleri ayrı olduğundan bakım kolaylaşır.
- Grafikselle Kullanıcı Arayüzünü destekler. Böylelikle eğitim için gerekli yatırım maliyetlerini azaltır.
- Açık sistem mimarisini destekler. Bu yolla kullanıcılara marka esnekliği sağlar.
- Yeni yazılım bileşenlerinin eklenmesi oldukça kolaydır.

i/S Mimarisinin Yararları ve Aksak Yönleri

i/S mimarisinin aksak yönleri ise:

- Uygulama mantığının büyük ölçüde sunucu üzerinde gerçekleştirilmesi durumunda darboğaz oluşabilir.
- Özellikle, birlikte çalışabilecek biçimde tasarlanmış dağıtık uygulamalar, dağıtık olmayan uygulamalara oranla çok daha fazla karmaşıktır.
- Varolan geliştirme metodolojileri ve yöntem ve araçlar sınırlıdır.

biçiminde özetlenebilir.

Sonuç olarak i/S mimarileri yazılım bakım maliyetlerini düşürür, yazılım taşınabilirliğini arttırır ve ağ performansını etkinleştirir.

WAP MİMARİSİ



- Kablosuz Uygulama Protokolü WAP (Wireless Application Protocol), kablosuz iletişim ağları üzerinde uygulama geliştirmek ve işletmek amacıyla geliştirilmiştir.
- WAP kablosuz aygıtlar (telsiz telefonlar, cep telefonları, kişisel el bilgisayarları vb) için bir uygulama altyapısı ve ağ protokolleri tanımlar.
- WAP, oldukça hızlı gelişen iki temel teknoloji üzerinde konumlanmıştır: **Kablosuz Veri ve Internet.**
- İnternet üzerinde geliştirilen teknolojilerin çoğu masaüstü ya da daha büyük boyutlu bilgisayarlar için tasarlanmıştır.
- Cep telefonu ve benzeri kablosuz aygıtların hesaplama ve bilgi saklama yetenekleri daha sınırlıdır.

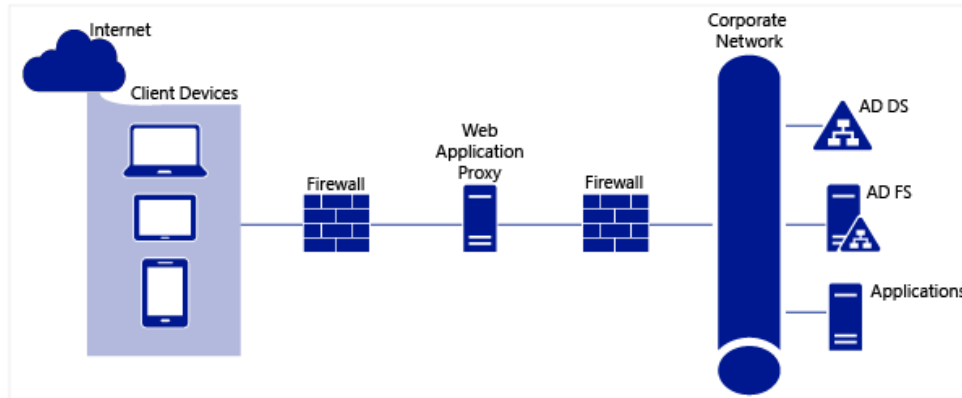
WAP teknolojisi Amaçları

- Daha az güçlü merkezi işlem birimi,
- Daha az ana bellek gücü,
- Sınırlı güç tüketimi
- Farklı giriş-çıkış aygıtları



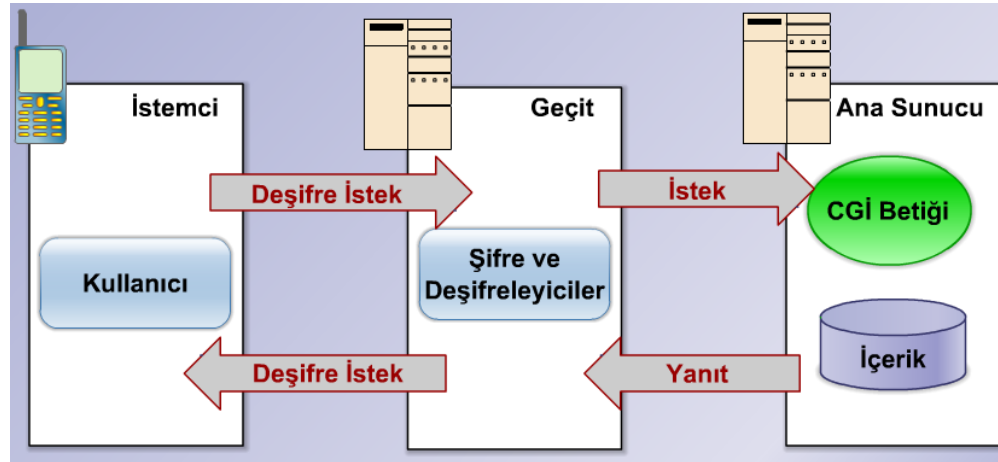
WAP Programlama Modeli

- WAP programlama modeli, www programlama modeli ile oldukça benzerlikler göstermektedir.
- www programlama altyapısında kullanılan araç ve yöntemler, daha küçük ölçekte olmak üzere WAP programlama platformunda kullanılabilir.
- Kablosuz ortamı www ile birleştirmek amacıyla, WAP, proksi teknolojisini kullanmaktadır.



WAP Programlama Modeli

WAP Proxy tipik olarak aşağıda belirtilen işlemleri yerine getirmektedir:



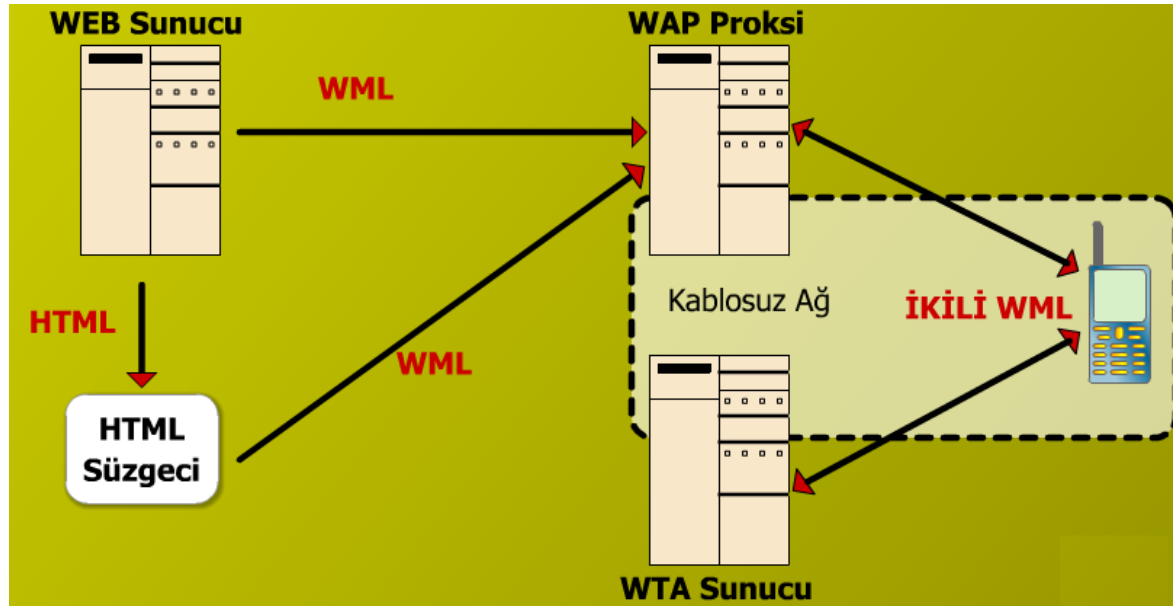
a. Protokol Ağ Geçidi: Bu geçit WAP, protokolünü (WSP,ETP,WTLS, ve WDTP) www protokolüne (http ve TCP/IP) dönüştürür.

b. İçerik şifreleyici ve deşifreleyiciler: Ağ üzerinde dolaşan veri hacmini sınırlamak amacıyla şifreleme ve deşifreleme işlemleri bu araçlar aracılığı ile yapılır.

WAP Ağı Nasıl Çalışır?

- WAP istemci, kablosuz ağ üzerinde iki sunucu ile iletişim kurmaktadır.
- WAP proxy, WAP isteklerini WWW isteği biçimine dönüştürmekte ve web sunucudan gelen yanıtları ikili biçim şeklinde şifrelemektedir.
- Web sunucunun, WAP içeriği sağlaması (örneğin WML) durumunda WAP proksi bu içeriği doğrudan web sunucudan almaktadır.
- Web sunucunun www içeriği (örneğin HTML) sağlaması durumunda ise www içeriğini WAP içeriğine dönüştürecek bir HTML süzgeci kullanılmaktadır.
- Örneğin, HTML süzgeci HTML kodunu WML koduna dönüştürmektedir.
- Kablosuz Telefon Uygulama (WTA-Wireless Telephony Application) sunucusu, WAP istemcisine doğrudan yanıt veren ağ geçit sunucusu için iyi bir örnektir.
- WTA, kablosuz ağ sağlayıcıların Telekom alt yapısını kullanır.

WAP Ağı Çalışma Örnek



Bileşen-Tabanlı Mimariler

- Günümüzde, internet kullanımının giderek yaygınlaşması ve elektronik ticaret, e-iş kavramlarının uygulamaya geçirilmesi, uygulama ortamlarının dağıtık ve heterojen yapıya dönüşmesi sonucunda "**yeniden kullanılabilirlik**" olgusu giderek önem kazanmış ve bileşen tabanlı mimari kavramı gündeme gelmiştir.
- Donanım üreticilerinin yıllardır standart bileşenler üretilip bu bileşenleri birleştirilerek ürün oluşturdukları bilinmektedir.
 - Örneğin, bir otomobil, aynı ya da farklı üreticilerden edinilen parçaların bir araya getirilmesi ile oluşturulabilir.
- Günümüzde otomobil üretimi büyük ölçüde bu biçimde yapılmaktadır. Donanım üreticilerinin bu üretim tarzını yazılım üretimine uygulamaya çalıştığımızda "**Bileşen Tabanlı**" yazılım mimarileri ortaya çıkar.
- Yazılım bileşeni, "**çalışan yazılım yapıtaşları**" ya da işletim zamanında birlikte çalışmak üzere bütünleştirilebilen yazılım parçası olarak tanımlanmaktadır.

Bileşen-Tabanlı Mimariler

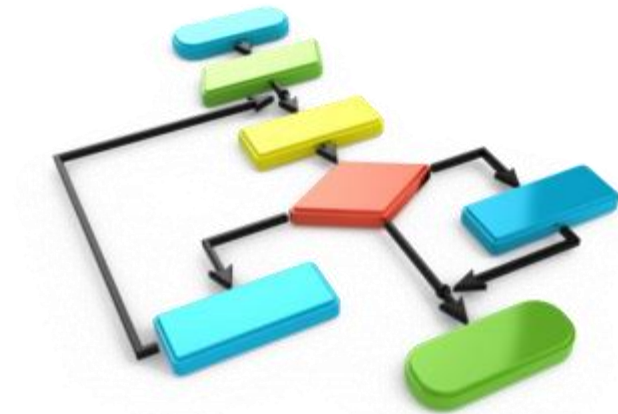
- Küçük boyutlu yazılım teknolojilerine örnek olarak: java Beans, OLE/COM ve DCOM verilebilir.
- Bir yazılım bileşeni, üç temel parçadan oluşur: **Kullanıcı Arayüzü, Yöntemler ve Amaç Kod.**
- Arayüz, bileşenin nasıl kullanacağını; yöntemler, bileşenin işlevselliğini; amaç kodu ise bileşenin doğrudan çalışabilir kodunu içerir.
- Bir bileşen, **açıklayıcı, içerilebilir, değiştirilebilir ve genişletilebilir** özellikleri içermelidir. Bu dört özellik bir bileşen oluşturmak için sağlanması zorunlu koşullardır.
- Öte yandan, bir bileşen uygulama, dil ya da platform ile sınırlı değildir.
- Bileşen tabanlı mimarilerde, ayrıntı tasarım daha az önem kazanmakta, çalışma daha çok genel tasarım ve arayüz tasarımı üzerinde yoğunlaşmaktadır.

Bileşen Tabanlı Yazılım Geliştirmede Kullanılan Metodolojiler

Bileşen tabanlı yazılım geliştirmede kullanılan metodolojiler:

- Bileşenin destekleyeceği iş fonksiyonlarına dayalı sistem gereklerinin açık tanımlamaları,
- Sistemin her süreç için davranış çözümlemesi,
- Arayüz tanımları ve aralarındaki ilişkiler,
- Birleştirme, yeniden kullanım ve sistem iş akış kodlarının ayrıntılı tanımları

üzerinde yoğunlaşırlar.



Özet

İstemci: Sunucuya bir iş yapmak ya da servis sağlamak üzere ileti gönderen bir süreç ya da yazılım olarak tanımlanabilir.

Sunucu: İstemci tarafından gereken istegi yerine getiren süreç ya da yazılımlar sunucu yazılımları olarak adlandırılır.

Sunucu Türleri:

1. Kütük Sunucuları
2. Veri Tabanı Sunucuları
3. Ara İşlem Sunucuları
4. Nesne Sunucuları
5. Web sunucuları

şeklindedir.

HTTP:

1. İ/S bağlantısı kurar,
2. Parametreleri gönderir ve alır,
3. İ/S bağlantısını kapatır.



Özet

WWW standartları şu şekildedir.

- **Standart isimlendirme Modeli:** Bütün sunucuların isimlendirilmesi ve içeriği için internet Tek Kaynak Yerleştirici (URL-Uniform Resource Locator) standardı (RFC1738,RFC1808)
- **İçerik Türü:** WWW üzerindeki her tür içerik, web görüntüleyicilerin işleyebileceği özel bir türe sahiptir (RFC2045, RFC2048)
- **Standart İçerik Biçimi:** Bütün web sunucuları bir dizi standart içerik biçimini destekler. Bunlar arasında Hiper Metin Biçimlendirme Dili (HTML-Hypertext Markup Language), JavaScript betik dili(ECMA Script, Javascript) en yaygın olanlarıdır.
- **Standart Protokol:** Web görüntüleyicilerin herhangi bir web sunucu ile iletişimini sağlayan ağ protokol standartları arasında en yaygın olan, HiperText Transfer Protokolüdür (HTTP).

İki katmanlı mimarilerde, uygulamanın üç temel bileşeni iki katmanda bulunur. Bu bileşenler: Kullanıcı Arayüzü (Grafiksel Kullanıcı Arabirimi, görüntü yönetimi), İşlem yönetimi (ya da uygulama mantığı) ve Veritabanı yönetimi (veri ve kütük servisleri)'dir. Üç katmanlı mimarilerde, uygulamanın üç temel bileşeni, üç bağımsız katmanda bulunur. Bu bileşenler: Kullanıcı (sistem) Arayüzü,İşlem Yönetimi (Uygulama Mantığı) ve Veri tabanı servisleridir.

Sorular

1. Yazılım Mimarisi terimini açıklayınız.
2. Yazılım mimarilerini sınıflandırınız.
3. Geleneksel yazılım mimarileri ile istemci sunucu mimarileri arasındaki farklılıkları belirtiniz.
4. Çevrim içi araışlem işlemeyi tanımlayınız.
5. 2-katmanlı mimari ve 2-katmanlı mimarileri karşılaştırınız.
6. 2 katmanlı ve 3 katmanlı mimarilere uygun olmayan uygulama örnekleri veriniz.
7. Katmanlı mimarilerin uygulama alanlarını belirtiniz.
8. Ağ trafiği üzerindeki etkileri açısından iki ve üç katmanlı İ/S mimarilerini karşılaştırınız.
9. Web mimarisinde kullanılan standartları belirtiniz. Standart kullanımının önemini açıklayınız.
10. WAP mimarisinin ayırıcı özellikleri nelerdir?
11. WAP uygulamaları geliştirme ve kullanılan araç ve yöntemler açısından diğer uygulamalarla kıyaslayınız.
12. Bileşen tabanlı uygulama mimarisini açıklayınız.
13. Bileşen tabanlı uygulama mimarisi ile istemci sunucu mimarisini kıyaslayınız.
14. Bileşen ile nesne arasındaki benzerlik ve farklılıkları belirtiniz.

Kaynaklar

- “Software Engineering A Practitioner’s Approach” (7th. Ed.), Roger S. Pressman, 2013.
- “Software Engineering” (8th. Ed.), Ian Sommerville, 2007.
- “Guide to the Software Engineering Body of Knowledge”, 2004.
- ” Yazılım Mühendisliğine Giriş”, TBİL-211, Dr. Ali Arifoğlu.
- ”Yazılım Mühendisliği” (2. Basım), Dr. M. Erhan Sarıdoğan, 2008, İstanbul: Papatya Yayıncılık.
- Kalıpsız, O., Buharalı, A., Biricik, G. (2005). Bilgisayar Bilimlerinde Sistem Analizi ve Tasarımı Nesneye Yönelik Modelleme. İstanbul: Papatya Yayıncılık.
- Buzluca, F. (2010) Yazılım Modelleme ve Tasarımı ders notları (<http://www.buzluca.info/dersler.html>)
- Hacettepe Üniversitesi BBS-651, A. Tarhan, 2010.
- Yazılım Proje Yönetimi, Yrd. Doç. Dr. Hacer KARACAN