Bölüm 6

Mimari Tasarım
(Architectural Design)

İçerik

- Mimari tasarım kararları
- Mimari bakış açıları
- Mimari desenler
- Uygulama mimarileri

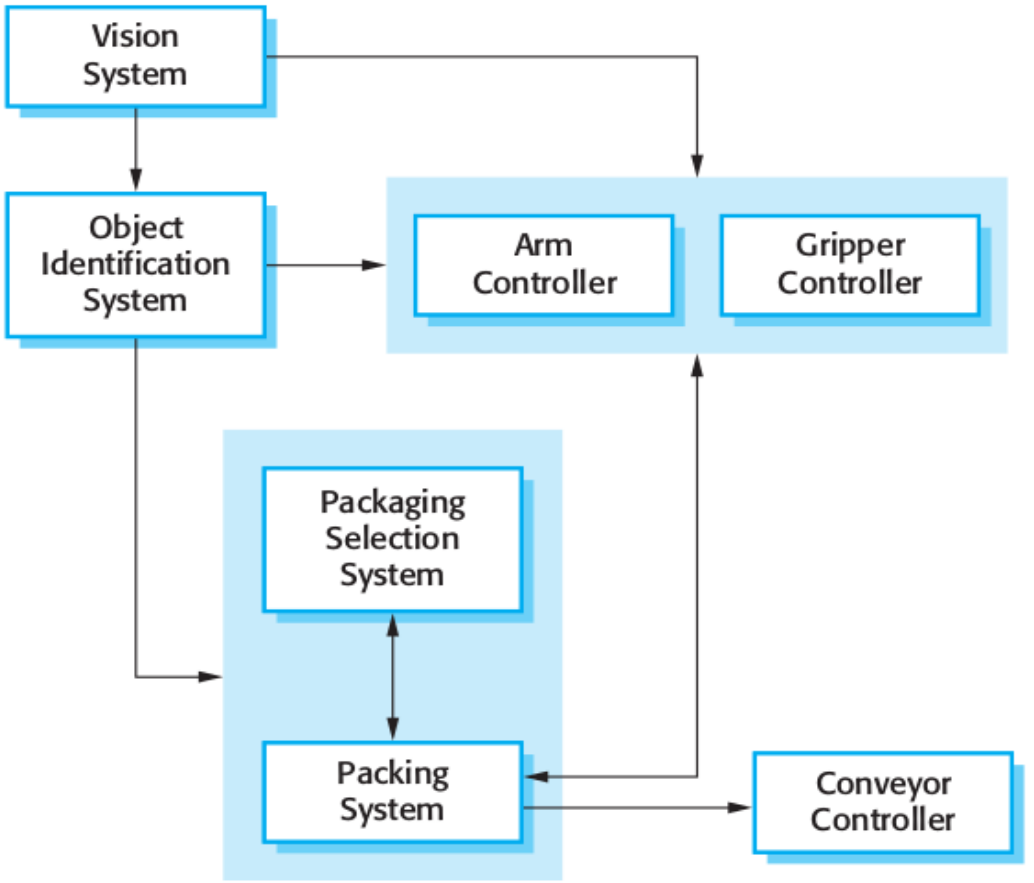
Yazılım Mimarisi

- Bir sistemi oluşturan alt sistemleri ve bu alt sistemlerin kontrolü ve iletişimi için kullanılan çatıyı tasarlama sürecine mimari tasarım diyoruz
- Bu tasarım sürecinin çıktısı, yazılım mimarisinin tanımlamasıdır

Mimari Tasarım

- Sistem tasarımı sürecinin ilk adımlarındandır
- Yazılımın tanımı ile tasarım süreçleri arasındaki bağı temsil eder
- Genellikle bazı tanımlama işleri ile paralel olarak yürütülür
- Temel sistem bileşenlerini ve onlar arasındaki iletişimi tanımlamayı kapsar

Paketleme Robotu Kontrol Sistemi



Mimari Soyutlama

- Küçük sistemlerde mimari,tekil programların mimarisi ile ilgilidir. Bu seviyede, tek programın bileşenlerine hangi yolla ayrıldığı ile ilgileniriz
- Büyük sistemlerde mimari, farklı sistemleri, programları ve program bileşenlerini içeren karmaşık iş sistemleriyle ilgilidir. Bu iş sistemleri farklı şirketlerin bilgisayarları üzerine dağıtılmış olabilir

Açık Mimarinin Yararları

- Paydaşlarla iletişim
  - Mimari paydaşların sistem üzerinde fikir yürütmeleri için kullanılabilir
- Sistem analizi
  - Sistemin fonksiyonel olmayan gereksinimlerini karşılamasının mümkün olup olmadığının analizidir
- Büyük ölçekli yeniden kullanım
  - Birçok sistemde yeniden kullanılabilecek bir mimari olabilir
  - Ürün-hattı mimarileri geliştirlebilir

Mimari Gösterim Şekilleri

- Yazılım mimarisinin dokümantasyonu için en çok kullanılan yöntem, bileşenlerin ve ilişkilerinin gösterildiği, basit ve gayrıresmi blok diyagramlarıdır
- Fakat bu yöntem mimarideki bileşenler arasındaki ilişkilerin türlerini ve bileşenlerin görünür özelliklerini göstermediğinden, mantıksallığın eksikliğinden dolayı eleştirilir.
- Modelin anlamsallığına duyulan gereksinim modellerin kullanımına bağlıdır

Kutu-çizgi Diyagramları

- Çok soyut – Bileşenler arasındaki ilişkilerin doğasını veya alt sistemlerin dışarıdan görünebilen özelliklerini göstermezler
- Fakat, paydaşlarla iletişimde ve proje planlaması için kullanışlıdırlar

Mimari Modellerin Kullanımı

- Sistem tasarımı hakkında tartışabilmek için bir yoldur
  - Detaylardan uzak olduğu için, sistem paydaşları ile iletişim kurmak ve proje planlaması yapmak için kullanışlıdır
- Tasarlanan mimarinin dokümantasyonunu yapmak için bir yoldur
  - Buradaki amaç sistemdeki farklı bileşenleri, arayüzlerini ve bağlarını göstermek için tam bir sistem modeli üretmektir

Mimari Tasarım Kararları

- Mimari tasarım, yaratıcı bir süreçtir ve bu süreç, geliştirilen sistemin tipine göre farklılıklar gösterir
- Fakat, tüm tasarım süreçlerinde yer alan bazı genel kararlar vardır ve bu kararlar sistemin fonksiyonel olmayan özelliklerini etkiler

Mimari Tasarım Kararları

- Kullanılabilecek genel bir uygulama mimarisi var mı?
- Sistem nasıl dağıtılacak?
- Hangi mimari tarzlar uygundur?
- Sistemi yapılandırmak için hangi yaklaşım kullanılacak?
- Sistem modüllere nasıl ayrıştırılacak?
- Hangi kontrol stratejisi kullanılmalı?
- Mimari tasarımın değerlendirmesi nasıl yapılacak?
- Mimari nasıl dokümana dönüştürülecek?

Mimarinin Yeniden Kullanımı

- Aynı çalışma alanındaki sistemler genellikle o alanın özelliklerini yansıtan benzer mimarilere sahiptir
- Uygulama ürün hatları çekirdek bir mimari üzerine belirli müşterilerin gereksinimleri doğrultusunda çeşitlendirilerek yapılandırılır
- Bir sistem, bir veya daha fazla mimari desen ('stil') üzerinden geliştirilebilir

Mimari ve Sistem Özellikleri

- Performans
  - Kritik operasyonları yerelleştir ve iletişimi en aza indir. Büyük bileşenler kullan
- Güvenlik
  - Katmanlı mimari kullan
- Emniyet
  - Emniyet için kritik olan özellikleri az sayıda alt sistemin içine yerleştir
- Uygunluk
  - Hataya karşı tolerans için bileşenlerin birden fazla örneğinin bulunduğu mekanizmalar kullan
- Bakım kolaylığı
  - Küçük ve kolay değiştirilebilir bileşenler kullanUse fine-grain, replaceable components

Mimari Perspektifler

- Sistem mimarisini tasarlarken ve belgelendirirken hangi bakış açıları kullanışlıdır?
- Mimari modelleri tanımlamak için hangi notasyonlar kullanılmalı?
- Her mimari model sistemin sadece bir perspektifini gösterir
  - Sistemin bileşenlerine nasıl ayrıldığını, çalışma-zamanı süreçlerinin etkileşimlerini veya sistem bileşenlerinin bir ağ üzerine nasıl dağıtıldığını da göstermelidir. Tasarım ve belgeleme işleri için genellikle yazılım mimarisi farklı açılardan gösterilir

Yazılım Mimarisinin 4+1 Bakış Modeli

- Mantıksal bakış – Sistemdeki ana soyutlamaları nesneler veya nesne sınıfları şeklinde gösterir
- Süreç bakışı – Çalışma zamanında sistemin etkileşimli süreçlerden nasıl oluşturulduğunu gösterir
- Geliştirim bakışı – Yazılımın geliştirme için nasıl ayrıştırıldığını gösterir
- Fiziksel bakış – Sistemin donanımını ve yazılım bileşenlerinin sistemdeki işlemcilere nasıl dağıtıldığını gösterir
- Kullanım durumu veya senaryolarla ilişkili (+1)

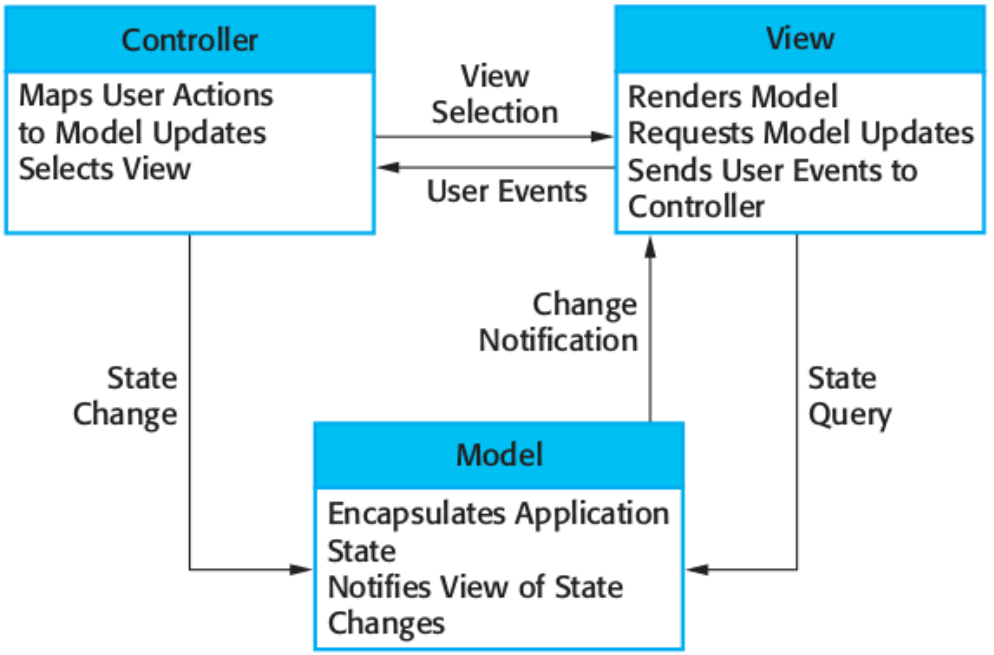
Mimari Desenler

- Desenler bilginin gösterimi, paylaşımı ve yeniden kullanımı için araçlardır
- Bir mimari desen, farklı ortamlarda denenmiş ve test edilmiş, iyi bir tasarımın tanımıdır
- Desenler, ne zaman kullanışlı ne zaman kullanışsız olduklarını belirten bilgiyi içermelidir
- Desenler, tablo halinde ve çizgesel olarak gösterilebilirler

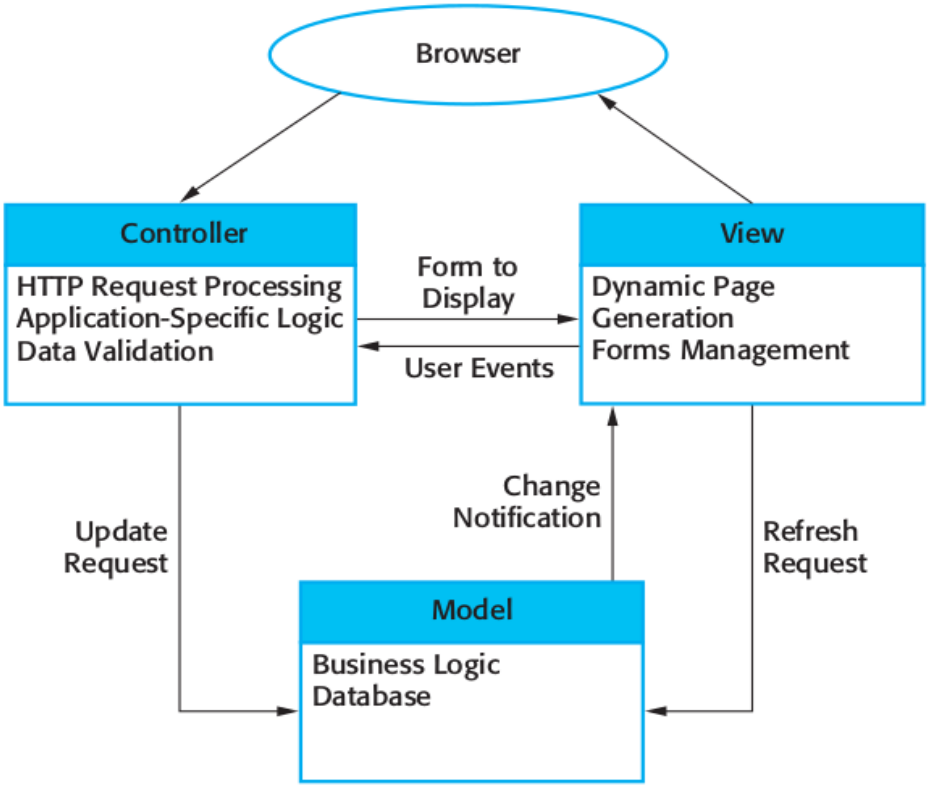
MVC (Model-View-Controller) Mimari Tasarım Deseni

Ad	MVC (Model-View-Controller)
Tanım	Sistem verisinden sunum ve etkileşimi ayırır. Sistem, birbiriyle etkileşen üç mantıksal bileşene ayrılır. Model bileşeni veriyi ve veri üzerindeki işlemleri yönetir. View bileşeni verinin kullanıcıya nasıl sunulduğunu tanımlar ve yönetir. Controller bileşeni kullanıcıdan gelen hareketleri (fare tıklaması, klavyede tuşa basma gibi) yönetir ve bunları View ve Model'e iletir.
Örnek	Bir sonraki slaytta web-tabanlı uygulama için MVC deseni kullanılarak sistemin nasıl yapılandırıldığı görülmektedir.
Ne zaman kullanılır?	Veriyi görüntülemenin ve veri ile etkileşimin birkaç yöntemi olduğunda kullanılır. Aynı zamandaverinin sunumu ve etkileşimi konusunda gelecek gereksinimler belirsiz ise kullanılır.
Avantajları	Veri değişiminin verinin gösteriminden bağımsız olarak yapılmasına izin verir (tersi de doğrudur). Aynı verinin birden fazla gösterimini destekler.
Dezavantajları	Veri modeli ve etkileşimler basit olduğunda çok fazla kod ve kod karmaşıklığı içerebilir.

MVC (Model-View-Controller) Mimari Tasarım Deseni



MVC ile Web Uygulama Mimarisi



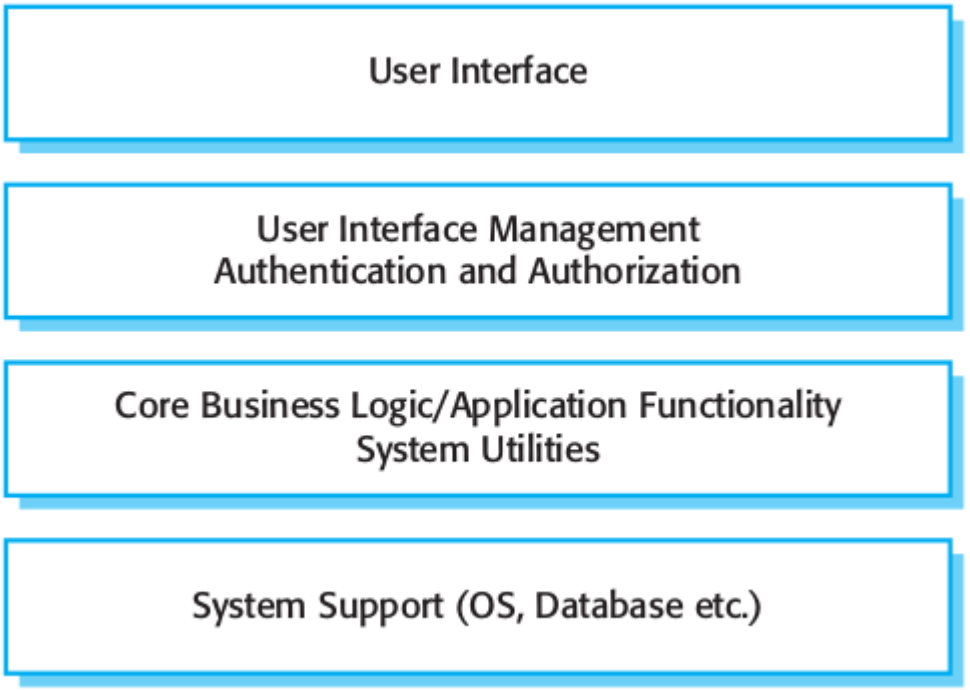
Katmanlı Mimari

- Alt sistemlerin arayüzlerini modellemede kullanılır
- Sistemi her biri bir dizi servis sağlayan birkaç katmana ayırır (veya soyut makineye)
- Alt sistemlerin farklı katmanlarda artımlı geliştirimini destekler
- Bir katmanın arayüzü değişirse sadece komşu katman etkilenir
- Fakat sistemleri bu şekilde yapılandırmak yapaydır

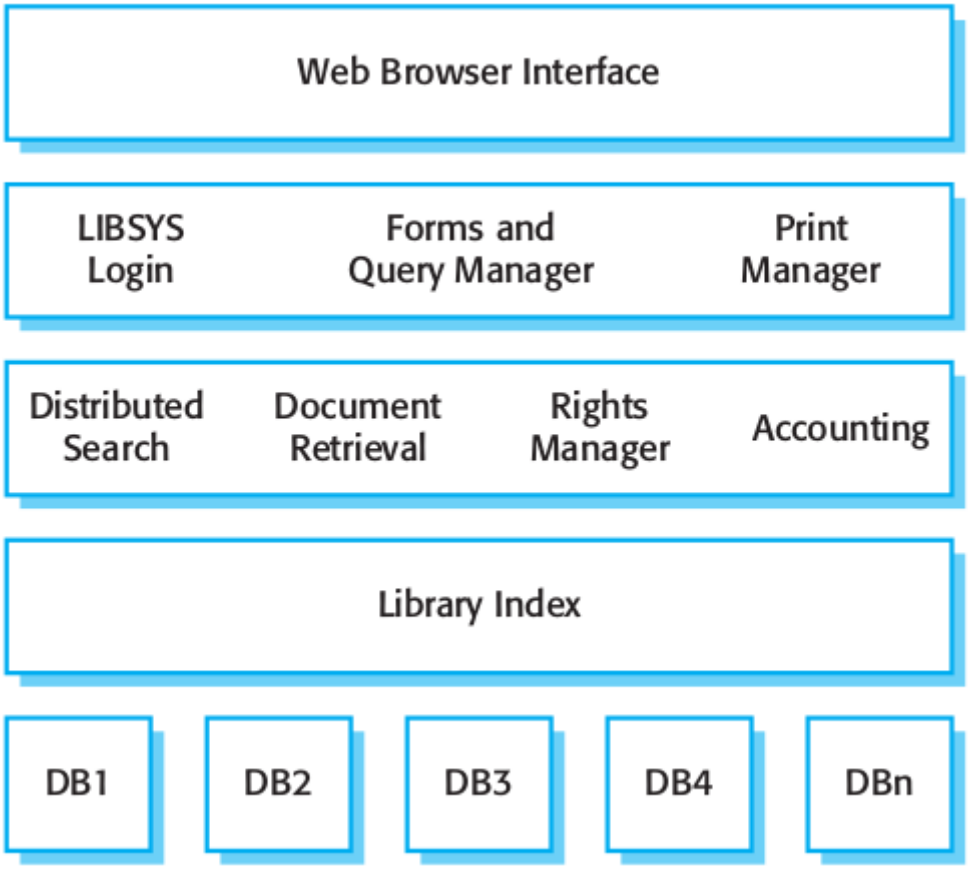
Katmanlı Mimari Deseni

Ad	Layered Architecture
Tanım	Sistemi ilgili fonksiyonlarına göre katmanlara ayırır. Her katman bir üst katmana servisler sunar; en alt seviyedeki katmanlar tüm sistemde kullanılacak çekirdek servisleri temsil eder.
Örnek	Farklı kütüphanelerde bulunan telif dokümanlarını paylaşmak için katmanlı mimari kullanan bir sistem bir sonraki slaytta görülmektedir.
Ne zaman kullanılır?	Varolan sistemlerin üzerine yeni imkanlar ekleneceği zaman; her takımın görevinin bir katmanın fonksiyonlarını geliştirmek olduğu dağıtık takımlardan oluşan bir yazılım geliştirme ekibi olduğu zaman; Çok seviyeli güvenliğe gereksinim olduğu zaman
Avantajları	Katmanların bütünüyle değiştirebilmesine izin verir. Bazı servislerin her katmanda tekrarlanması sistem bağlılığını arttırır (örn. Kimlik denetleme).
Dezavantajları	Pratikte katmanlar arası keskin sınırlar belirlemek genellikle zordur ve bazı üst düzey katmanların sadece kendi komşuluğundaki katman aracılığıyla değil doğrudan da alt katmanlarla iletişime geçmesi gerekebilir. Bir talebin yorumlanması her katmanda ayrı ayrı yapılacağından performans sorunları oluşabilir.

Genel Bir Katmanlı Mimari



Kütüphane Sistemi İçin Mimari



Önemli Noktalar

- Bir yazılım mimarisi yazılım sisteminin nasıl düzenlendiğini tanımlar
- Mimari tasarım kararları uygulamanın türüne, sistemin nasıl dağıtıldığına ve kullanılacak mimari stillere karar vermeyi içerir
- Mimariler, mantıksal bakış, süreç bakışı, geliştirim bakışı ve kavramsal bakış gibi farklı perspektiflerden belgelenebilir
- Mimari desenler genel sistem mimarileri hakkındaki bilginin yeniden kullanımına yönelik araçlardır. Mimariyi tanımlar, hangi durumlarda kullanımının uygun olacağını açıklar ve avantaj ve dezavantajlarını belirtirler

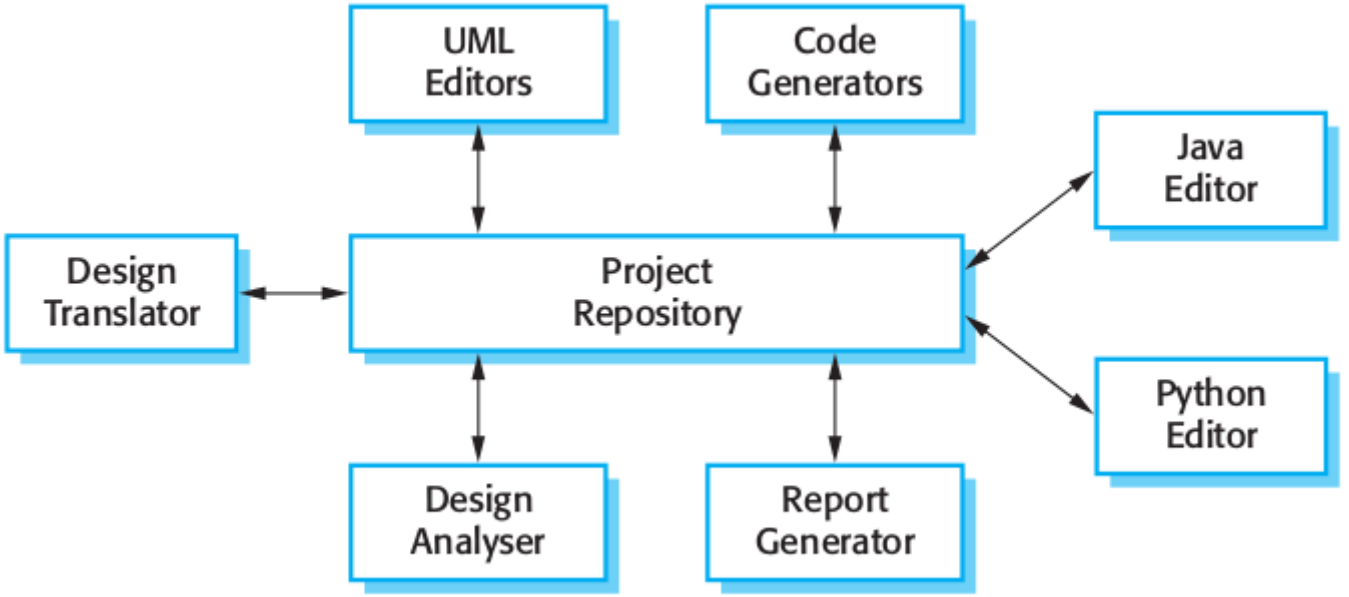
Veri Havuzu Mimarisi

- Alt sistemler arasında veri değişimi söz konusudur. Bu iki yolla yapılabilir:
  - Paylaşılan veri merkezi bir veritabanı veya veri havuzunda tutulur ve tüm alt sistemler tarafından erişilebilir
  - Her alt sistem kendi veritabanına sahiptir ve diğer sistemlere veriyi açık bir şekilde iletir
- Büyük miktarda verinin paylaşılacağı durumlarda veri havuzu modeli ile paylaşım en yaygın ve etkin veri paylaşım mekanizmasıdır

Veri Havuzu Deseni

Ad	Repository
Tanım	Sistemdeki tüm veri, tüm sistem bileşenlerinin erişebileceği bir veri havuzunda yönetilir. Bileşenler doğrudan değil veri havuzu aracılığıyla etkileşime girerler.
Örnek	Bir sonraki slaytta sistem tasarım bilgisinin veri havuzunu paylaşan bir IDE örneği bulunmaktadır.
Ne zaman kullanılır?	Çok miktarda verinin üretilip uzun süre saklanmasını gerektiren sistemlerde kullanılmalıdır. Veri-güdümlü sistemlerde de kullanılabilir.
Avantajları	Bileşenler bağımsız olabilir – diğer bileşenlerin varlığından haberdar olmak zorunda değildirler. Bir bileşende yapılan değişiklik diğer bileşenlere de yayılabilir. Veri tutarlılığı kolaylıkla sağlanabilir
Dezavantajları	Veri havuzunda oluşan problemler tüm sistemi etkiler. Sadece veri havuzu üzerinden iletişim kurulmasının düzenlenmesi etkinliği düşürebilir. Veri havuzunu birçok bilgisayar üzerine dağıtmak kolay olmayabilir.

Veri Havuzu Deseni (IDE örneği)



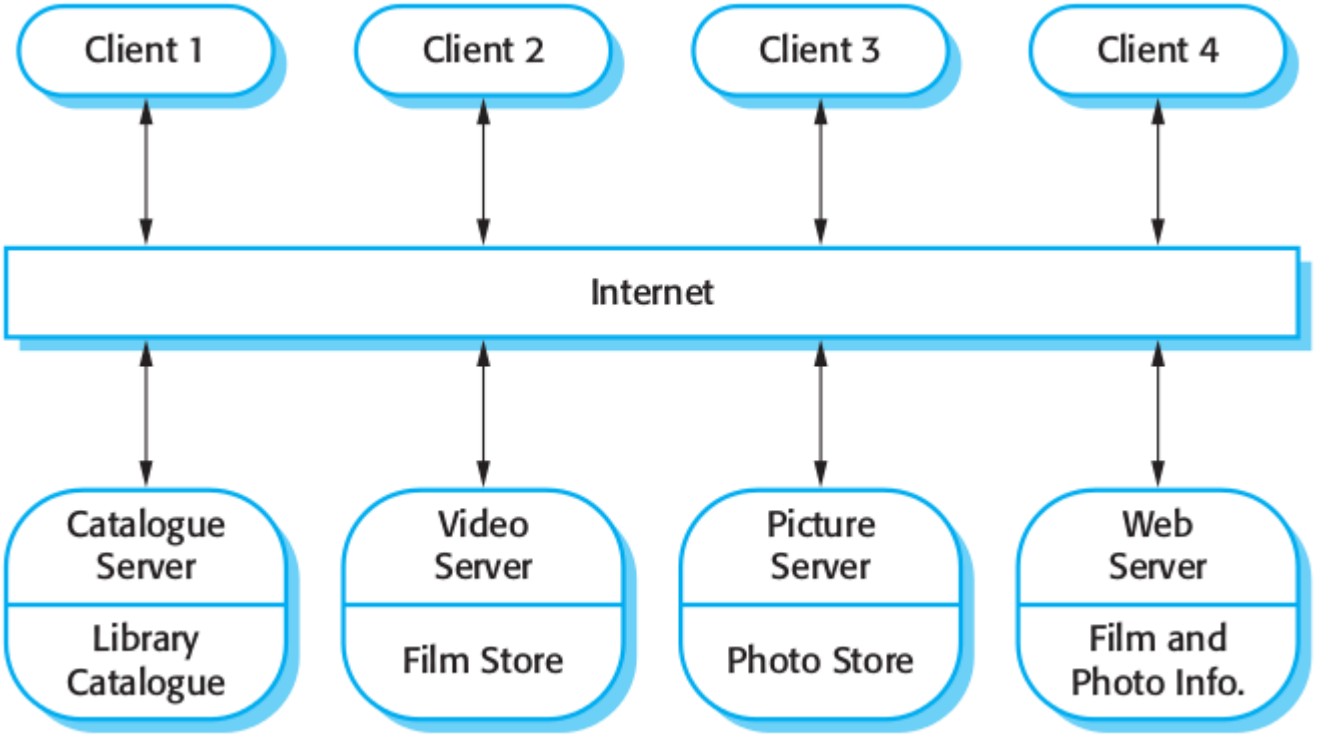
İstemci-Sunucu Mimarisi

- Verinin ve süreçlerin birçok bileşen üzerine nasıl dağıtılacağını gösteren dağıtık sistem modelidir
  - Tek bir bilgisayar üzerinde gerçekleştirilebilir
- Yazdırma, veri yönetimi gibi özel servisleri sağlayan bağımsız sunucular
- Bu servisleri çağıran istemciler
- İstemcilierin sunuculara erişimini sağlayan ağ

İstemci-Sunucu Mimarisi

Ad	Client-Server
Tanım	İstemci-sunucu mimarisinde sistemin fonksiyonelliği herbiri ayrı sunucudan gelen servisler şeklinde düzenlenmiştir. İstemciler de bu servislerin kullanıcılarıdır ve sunucuları kullanmak için onlara erişirler.
Örnek	Bir sonraki slaytta istemci-sunucu sistemi şeklinde düzenlenmiş video/DVD kütüphanesi örneği verilmiştir.
Ne zaman kullanılır?	Paylaşılan bir veritabanındaki veriye erişim birçok lokasyondan yapılmak zorundaysa kullanılır. Sunucular çoğullanabildiğinden sistem yükünün değişken olduğu zamanlarda da kullanılabilir.
Avantajları	Bu modelin temel avantajı sunucuların bir ağ üzerine dağıtılabilmesidir. Yazıcı servisi gibi genel fonksiyonlar tüm istemcilere açık olabilir ve onların tüm servisler tarafından gerçekleştirilmelerine gerek yoktur.
Dezavantajları	Herbir servis saldırılara ve sunucu hatalarına karşı hassastır. Sistemin kendisi kadar ağa da bağlı olduğundan, performans tahmin edilebilir değildir. Eğer sunucular farklı kuruluşlara aitse yönetimsel sorunlar çıkabilir.

İstemci-Sunucu Mimarisi



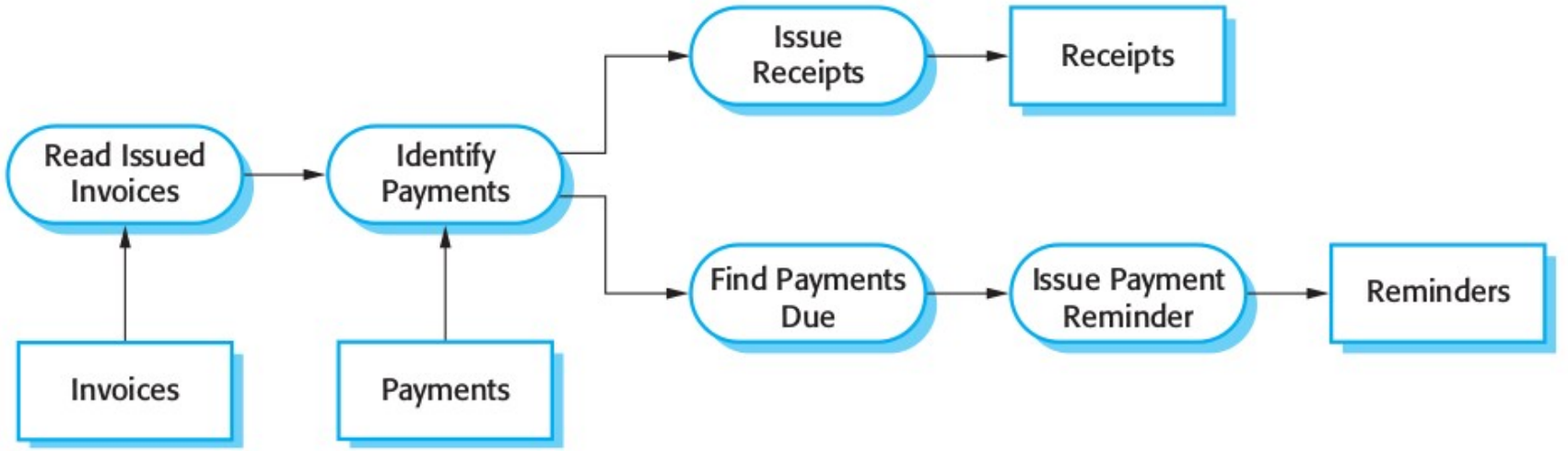
Boru ve Filtre Mimarisi

- Çıktıları üretmek için girdilerin işlenmesi, dönüştürülmesi
- Bu yaklaşımın çeşitleri vardır.
- Dönüşümler sıralı ise bu veri işleme sistemlerinde çok kullanılan sıralı yığın modelidir
- Etkileşimli sistemler için uygun değildir

Boru ve Filtre Mimarisi

Ad	Pipe and Filter
Tanım	Veri işlenmek üzere bir bileşenden diğer bileşene akar. Sistemdeki veri işleme düzenlenmiştir böylece herbir işleme bileşeni bir tipte veriyi dönüştürür.
Örnek	Bir sonraki slaytta faturalama örneği bulunmaktadır.
Ne zaman kullanılır?	Genellikle girdilerin ayrı ayrı fazlarda işlendiği veri işleme uygulamalarında kullanılır.
Avantajları	Anlaşılması kolaydır ve dönüşümün yeninden kullanılmasını destekler. İş akışı stili çoğu iş uygulaması ile uyum gösterir. Dönüşüm ekleyerek evrimin gerçekleştirilmesi kolaydır. Sıralı veya eş zamanlı olarak gerçekleştirilebilir.
Dezavantajları	Veri transferi için kullanılacak biçime daha önceden karar verilmelidir. Her dönüşüm girdisini ve çıktısını bu biçime göre çözümlemelidir. Bu sistem yükünü arttırır ve birbirine uyumsuz veri yapıları kullanan dönüşümleri yeniden kullanmak imkansızdır.

Boru ve Filtre Mimarisi (fatura örneği)



Uygulama Mimarileri

- Uygulama sistemleri bir kurumun belirli bir ihtiyacını karşılamaya yönelik olarak tasarlanır
- İş dünyasında ortak olan çok fazla yön vardır, dolayısıyla uygulama sistemleri de uygulama gereksinimlerini yansıtan ortak mimariye sahip olma eğilimindedir
- Genel bir uygulama mimarisi, belirli gereksinimleri karşılayan bir sistem yaratmak için uyarlanmış olan bir yazılım için geliştirilmiş bir mimaridir

Uygulama Mimarilerinin Kullanımı

- Mimari tasarım için bir başlangıç noktası olarak
- Tasarım kontrol listesi olarak
- Geliştirim takımının işlerini organize etmek için
- Yeniden kullanım için bileşenleri değerlendirme aracı olarak
- Uygulama türleri ile ilgili konuşabilmek için

Uygulama Türlerine Örnekler

- Veri işleme uygulamaları
  - Veriyi kullanıcı dahli olmadan yığınlar halinde işleyen uygulamalar
- İşlem uygulamaları
  - Kullanıcı taleplerini işleyen ve veritabanında güncelleme yapan veri-merkezli uygulamalar
- Olay işleme uygulamaları
  - Sistem hareketlerinin sistemin çevresindeki olayların yorumlanmasına bağlı olduğu uygulamalar
- Dil işleme sistemleri
  - Kullanıcı isteklerinin sistem tarafından işlenip yorumlanan resmi bir dilde belirtildiği uygulamalardır

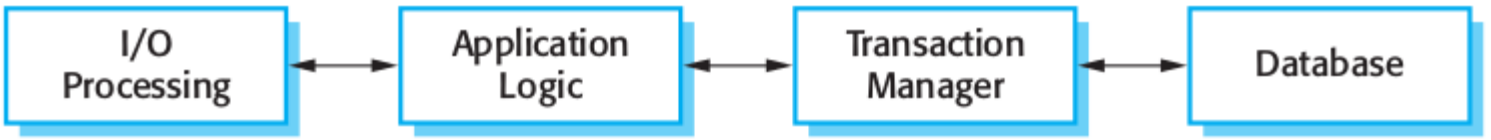
Uygulama Türü Örnekleri

- Burada ticari işlem ve dil işleme sistemlerine odaklanılmıştır
- İşlem sistemleri
  - E-ticaret sistemleri
  - Rezervasyon sistemleri
- Dil işleme sistemleri
  - Derleyiciler
  - Komut yorumlayıcıları

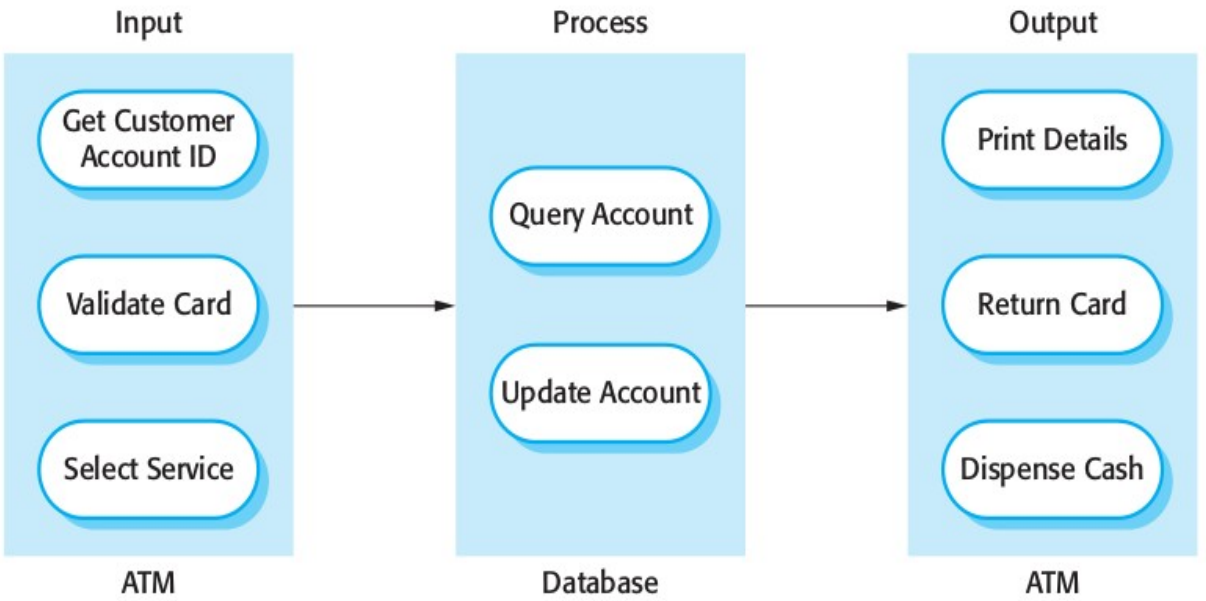
İşlem Uygulamaları

- Kullanıcılar veritabanından bilgi çekmek veya veritabanını güncellemek isterler
- Kullanıcı perspektifinden bir işlem:
  - Bir hedefi gerçekleştirmek için gerçekleştirilen uyumlu bir dizi işlem
  - Örneğin: İstanbul'dan İzmir'e uçuş saatlerini bulmak
- Kullanıcılar sistemden, daha sonra işlem yöneticisi tarafından işlenecek olan asenkron servis taleplerinde bulunurlar

İşlem Uygulamalarının Yapısı



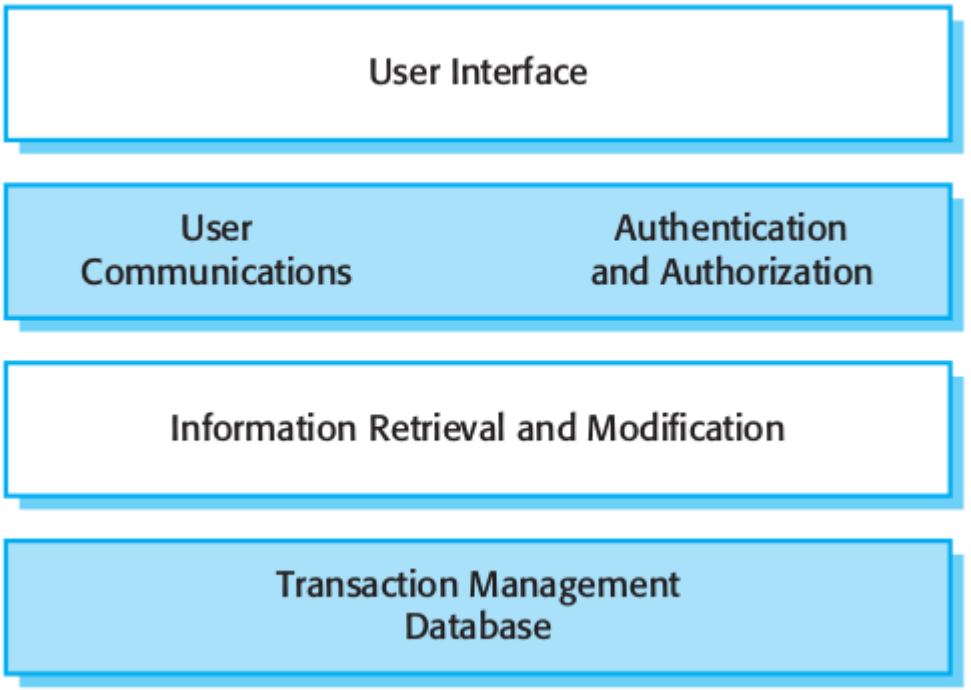
ATM Sistemi İçin Yazılım Mimarisi



Bilgi Sistemleri Mimarisi

- Bilgi sistemleri katmanlı mimari seklinde düzenlenebilen genel bir mimarye sahiptir
- Bunlar genelde veritabanı işlemlerini içeren sistemlerdir
- Katmanlar:
  - Kullanıcı arayüzü
  - Kullanıcı haberleşmesi
  - Bilgi çekme
  - Sistem veritabanı

Katmanlı Bilgi Sistemleri Mimarisi



Web-tabanlı Bilgi Sistemleri

- Bilgi ve kaynak yönetimi sistemleri şimdilerde genellikle kullanıcı arayüzünün web tarayıcısı olduğu web tabanlı sistemlerdir
- Örneğin, e-ticaret sistemleri, ürün veya servis siparişlerinin elektronik olarak alındığı ve bu ürün ve servislerin kullanıcıya tesliminin yapıldığı internet tabanlı kaynak yönetim sistemleridir.

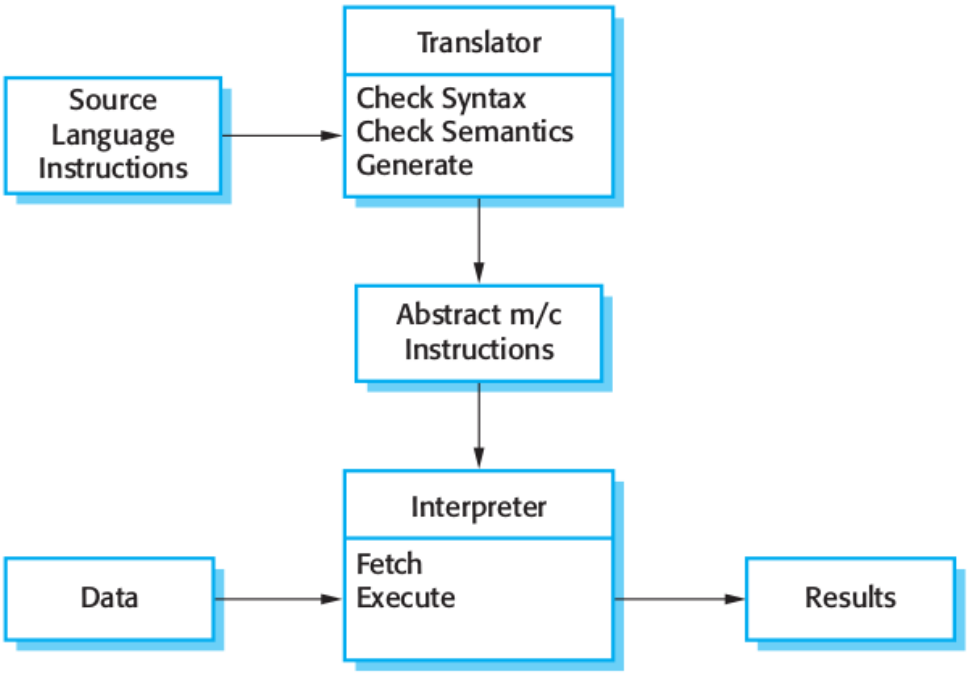
Sunucu Gerçekleştirimi

- Bu sistemler genellikle çok katmanlı istemci-sunucu mimarileri ile gerçekleştirilirler
- Web sunucusu web tarayıcı ile gerçekleştirilen tüm kullanıcı iletişiminden sorumludur
- Uygulama sunucusu uygulamaya özel mantığın geliştirilmesinden olduğu gibi bilgi depolama ve bilgi çekme isteklerinden de sorumludur
- Veritabanı sunucusu veritabanından bilgi almak ve veritabanına bilgi göndermek gibi işlemlerin yönetimini yapar

Dil İşleme Sistemleri

- Bir doğal veya yapay dili girdi olarak alır ve o dilin başka bir gösterimini üretir
- Bir yorumlayıcı içerebilir
- Bir problemi çözmek için kullanılabilecek en kolay yolun bir algoritma veya sistem verisi tanımlama olduğu durumlarda kullanılır

Dil İşleme Sistem Mimarisi



Derleyici Bileşenleri

- Sözcük analizcisi, tokenları girdi olarak alır ve bunları bir tür iç forma dönüştürür
- Sembol tablosu, çevrilen metindeki varlıkların isimlerini tutar
- Sözdizimi analizcisi, çevrilen dilin sözdizimi kurallarını kontrol eder
- Sözdizimi ağacı, derlenen programı temsil eden bir veri yapısıdır
- Mantıksal analizci, sözdizimi ağacındaki veriyi ve sembol tablosunu kullanarak girdi metninin mantıksal olarak doğruluğunu kontrol eder
- Kod üreticisi, sözdizimi ağacı üzerinde gezer ve soyut makine kodunu oluşturur

Önemli Noktalar

- Uygulama sistemleri mimari modelleri bize uygulamaları kıyaslamamız ve anlamamız, uygulama sistem tasarımlrını güncellememiz ve geniş ölçekli bileşenleri yeniden kullanım için değerlendirmemiz için yardım eder
- İşlem sistemleri bir veritabanında bulunan bilgilerin uzaktan erişimine ve çoğul kullanıcılar tarafından değiştirilmesine izin veren sistemlerdir
- Dil işleme sistemleri bir dilde yazılmış metinleri başka bir dile çevirerek metindeki komutları gerçekleştirmek için kullanılır. Üretilen dili çalıştırmak için bir çevirmen ve soyut makine içerirler

Görüşmek üzere