IT522 – Yazılım Mühendisliği 2021

PhD Furkan Gözükara, Toros University

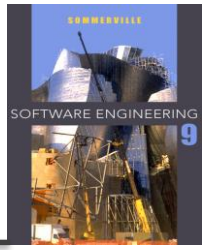
https://github.com/FurkanGozukara/Yazilim-Muhendisligi-IT522-2021

Ders 6

Mimari Tasarım

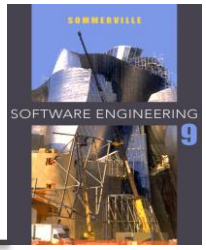
Kaynak : https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Presentations/index.html

Ders 6'da İşlenen Konular



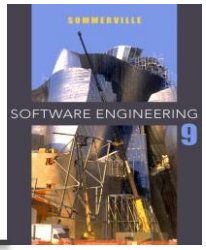
- Mimari tasarım kararları
- Mimari görünümler
- Mimari desenler
- Uygulama mimarileri

Yazılım Mimarisi



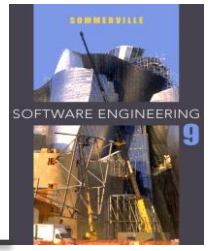
- Bir sistemi oluşturan alt sistemleri ve alt sistem kontrolü ve iletişimi çerçevesini belirlemeye yönelik tasarım süreci mimari tasarımdır.
- Bu tasarım sürecinin çıktısı, yazılım mimarisinin bir açıklamasıdır.

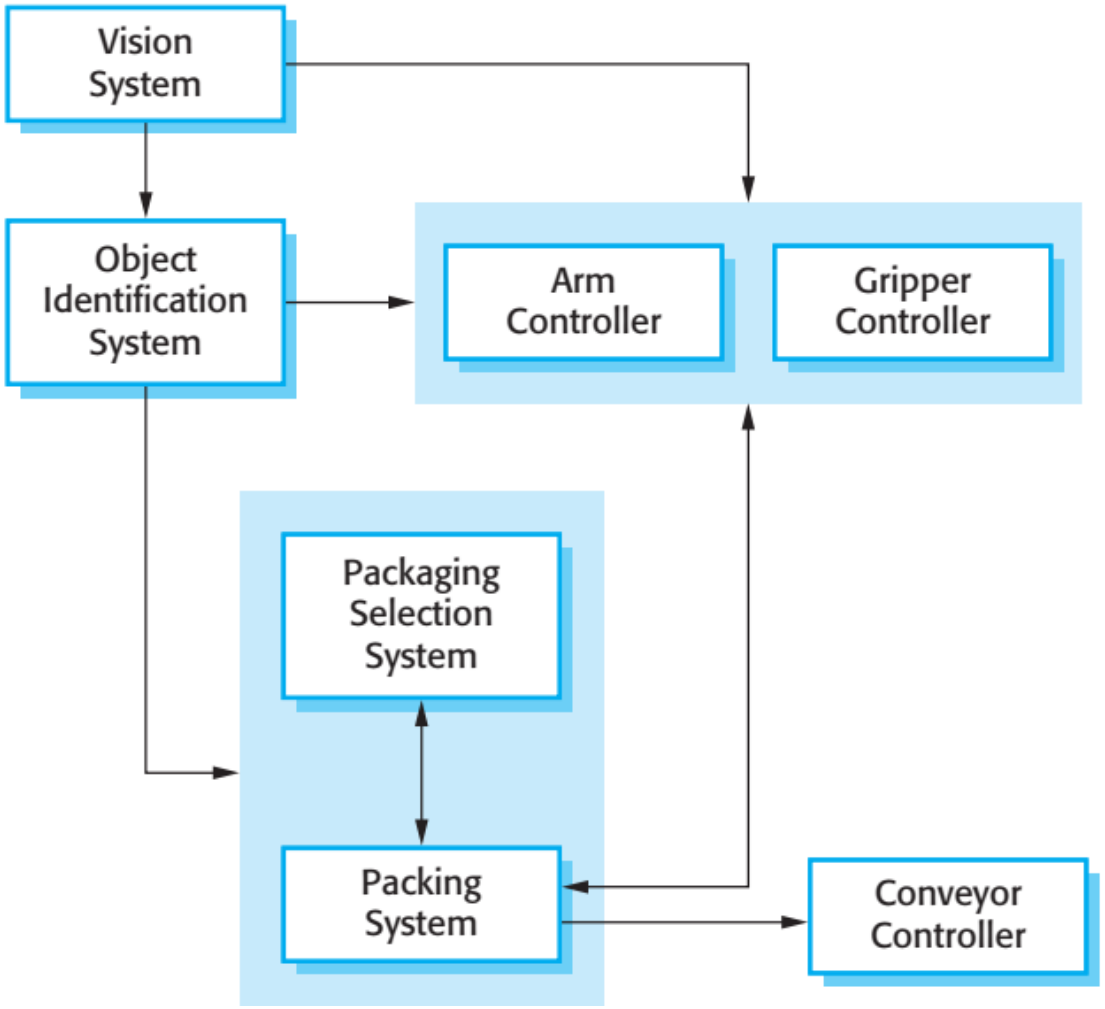
Mimari Tasarım



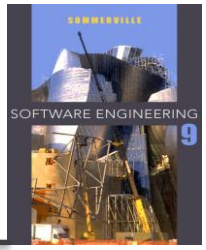
- Sistem tasarım sürecinin erken bir aşaması.
- Spesifikasyon ve tasarım süreçleri arasındaki bağlantıyı temsil eder.
- Genellikle bazı spesifikasyon faaliyetlerine paralel olarak gerçekleştirilir.
- Ana sistem bileşenlerinin ve bunların iletişimlerinin tanımlanmasını içerir.

Paketleme Robotu Kontrol Sisteminin Mimarisi



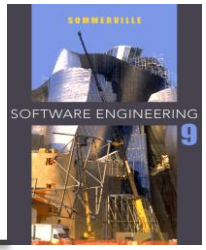


Mimari Soyutlama



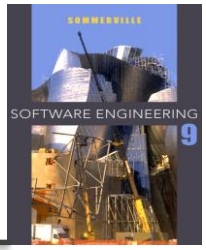
- Küçükte mimari, bireysel programların mimarisiyle ilgilenir. Bu düzeyde, tek bir programın bileşenlere ayrışması ile ilgileniyoruz.
- Genel olarak mimari, diğer sistemleri, programları ve program bileşenlerini içeren karmaşık kurumsal sistemlerin mimarisiyle ilgilidir. Bu kurumsal sistemler, farklı şirketler tarafından sahip olunan ve yönetilen farklı bilgisayarlara dağıtılır.

Açık Mimarinin Avantajları



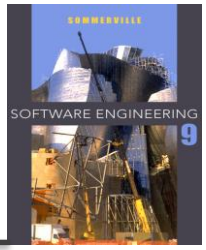
- Paydaş iletişimi
  - Mimari, sistem paydaşları tarafından bir tartışma odağı olarak kullanılabilir.
- Sistem Analizi
  - Sistemin işlevsel olmayan gereksinimlerini karşılayıp karşılamadığının analizinin mümkün olduğu anlamına gelir.
- Büyük ölçekli yeniden kullanım
  - Mimari, bir dizi sistemde yeniden kullanılabilir olabilir
  - Ürün hattı mimarileri geliştirilebilir.

Mimari Temsiller



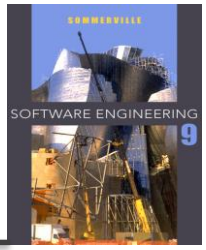
- Varlıkları ve ilişkileri gösteren basit, gayri resmi blok diyagramlar, yazılım mimarilerini belgelemek için en sık kullanılan yöntemdir.
- Ancak bunlar, anlambilimden yoksun oldukları, varlıklar arasındaki ilişki türlerini veya mimarideki varlıkların görünür özelliklerini göstermedikleri için eleştirildi.
- Mimari modellerin kullanımına bağlıdır. Model semantiği gereksinimleri, modellerin nasıl kullanıldığına bağlıdır.

Kutu ve Hat Diyagramları



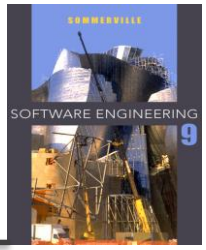
- Çok soyut - ne bileşen ilişkilerinin doğasını ne de alt sistemlerin dışarıdan görülebilen özelliklerini göstermezler.
- Ancak paydaşlarla iletişim ve proje planlaması için kullanışlıdır.

Mimari Modellerin Kullanımı



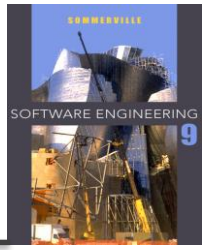
- Sistem tasarımı hakkında tartışmayı kolaylaştırmanın bir yolu olarak
  - Bir sistemin üst düzey mimari görünümü, detaylarla dağınık olmadığı için sistem paydaşlarıyla iletişim ve proje planlaması için kullanışlıdır. Paydaşlar bununla ilişki kurabilir ve sistemin soyut bir görünümünü anlayabilir. Daha sonra ayrıntılı olarak karıştırılmadan sistemi bir bütün olarak tartışabilirler.
- Tasarlanmış bir mimariyi belgelemenin bir yolu olarak
  - Buradaki amaç, bir sistemdeki farklı bileşenleri, arayüzlerini ve bağlantılarını gösteren eksiksiz bir sistem modeli üretmektir.

Mimari Tasarım Kararları



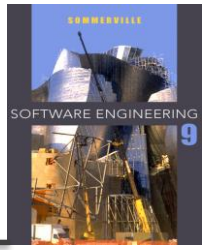
- Mimari tasarım üretici bir süreçtir, bu nedenle süreç, geliştirilmekte olan sistemin türüne bağlı olarak farklılık gösterir.
- Bununla birlikte, bir dizi ortak karar tüm tasarım süreçlerini kapsar ve bu kararlar sistemin işlevsel olmayan özelliklerini etkiler.

Mimari Tasarım Kararları



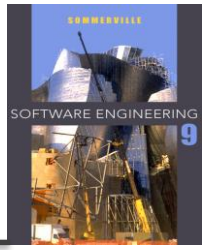
- Kullanılabilecek genel bir uygulama mimarisi var mı?
- Sistem nasıl dağıtılacak?
- Hangi mimari tarzlar uygundur?
- Sistemi yapılandırmak için hangi yaklaşım kullanılacak?
- Sistem modüllere nasıl ayrıştırılacak?
- Hangi kontrol stratejisi kullanılmalıdır?
- Mimari tasarım nasıl değerlendirilecek?
- Mimari nasıl belgelenmelidir?

Mimari Yeniden Kullanım



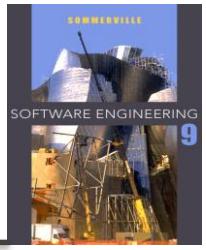
- Aynı etki alanındaki sistemler genellikle etki alanı kavramlarını yansıtan benzer mimarilere sahiptir.
- Uygulama ürün serileri, belirli müşteri gereksinimlerini karşılayan varyantlara sahip bir çekirdek mimari etrafında oluşturulmuştur.
- Bir sistemin mimarisi, birden fazla mimari desen veya "stil" etrafında tasarlanabilir.
  - Bunlar bir mimarinin özünü yakalar ve farklı şekillerde örneklenebilir.
  - Bu Bölümde daha sonra tartışılacaktır.

Mimari ve Sistem Özellikleri



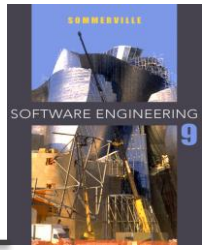
- Verim
  - Kritik işlemleri yerelleştirin ve iletişimi en aza indirin. Az iş yapan bileşenler yerine büyük parçalar kullanın.
- Güvenlik
  - İç katmanlarda kritik varlıklara sahip katmanlı bir mimari kullanın.
- Emniyet
  - Az sayıda alt sistemde güvenlik açısından kritik özellikleri yerelleştirin.
- Kullanılabilirlik
  - Hata toleransı için fazladan bileşenleri ve mekanizmaları dahil edin.
- Sürdürülebilirlik
  - Az iş yapan, değiştirilebilir bileşenler kullanın.

Mimari Görünümler



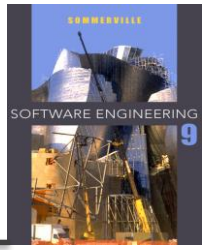
- Bir sistemin mimarisini tasarlarken ve belgelerken hangi görünümler veya perspektifler kullanışlıdır?
- Mimari modelleri tanımlamak için hangi gösterimler kullanılmalıdır?
- Her mimari model, sistemin yalnızca bir görünümünü veya perspektifini gösterir.
  - Bir sistemin modüllere nasıl ayrıştırıldığını, çalışma zamanı süreçlerinin nasıl etkileşimde bulunduğunu veya sistem bileşenlerinin bir ağda dağıtıldığı farklı yolları gösterebilir. Hem tasarım hem de dokümantasyon için, genellikle yazılım mimarisinin birden çok görünümünü sunmanız gerekir.

Yazılım Mimarisinin 4 + 1 Görünüm Modeli



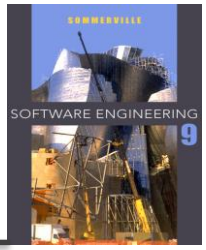
- Sistemdeki anahtar soyutlamaları nesneler veya nesne sınıfları olarak gösteren mantıksal bir görünüm.
- Çalışma zamanında sistemin etkileşimli işlemlerden nasıl oluştuğunu gösteren bir süreç görünümü.
- Yazılımın geliştirme için nasıl ayrıştırıldığını gösteren bir geliştirme görünümü.
- Sistem donanımını ve yazılım bileşenlerinin sistemdeki işlemciler arasında nasıl dağıtıldığını gösteren fiziksel bir görünüm.
- Kullanım senaryoları veya senaryoları ile ilgili (+1)

Mimari Desenler



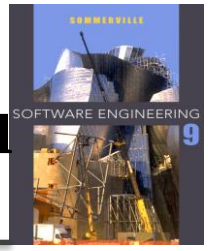
- Desenler/kalıplar, bilgiyi temsil etmenin, paylaşmanın ve yeniden kullanmanın bir yoludur.
- Mimari desen, farklı ortamlarda denenmiş ve test edilmiş, iyi tasarım uygulamasının stilize bir açıklamasıdır.
- Desenler, ne zaman yararlı oldukları ve ne zaman yararlı olmadıkları hakkında bilgi içermelidir.
- Desenler, tablo ve grafiksel açıklamalar kullanılarak gösterilebilir.

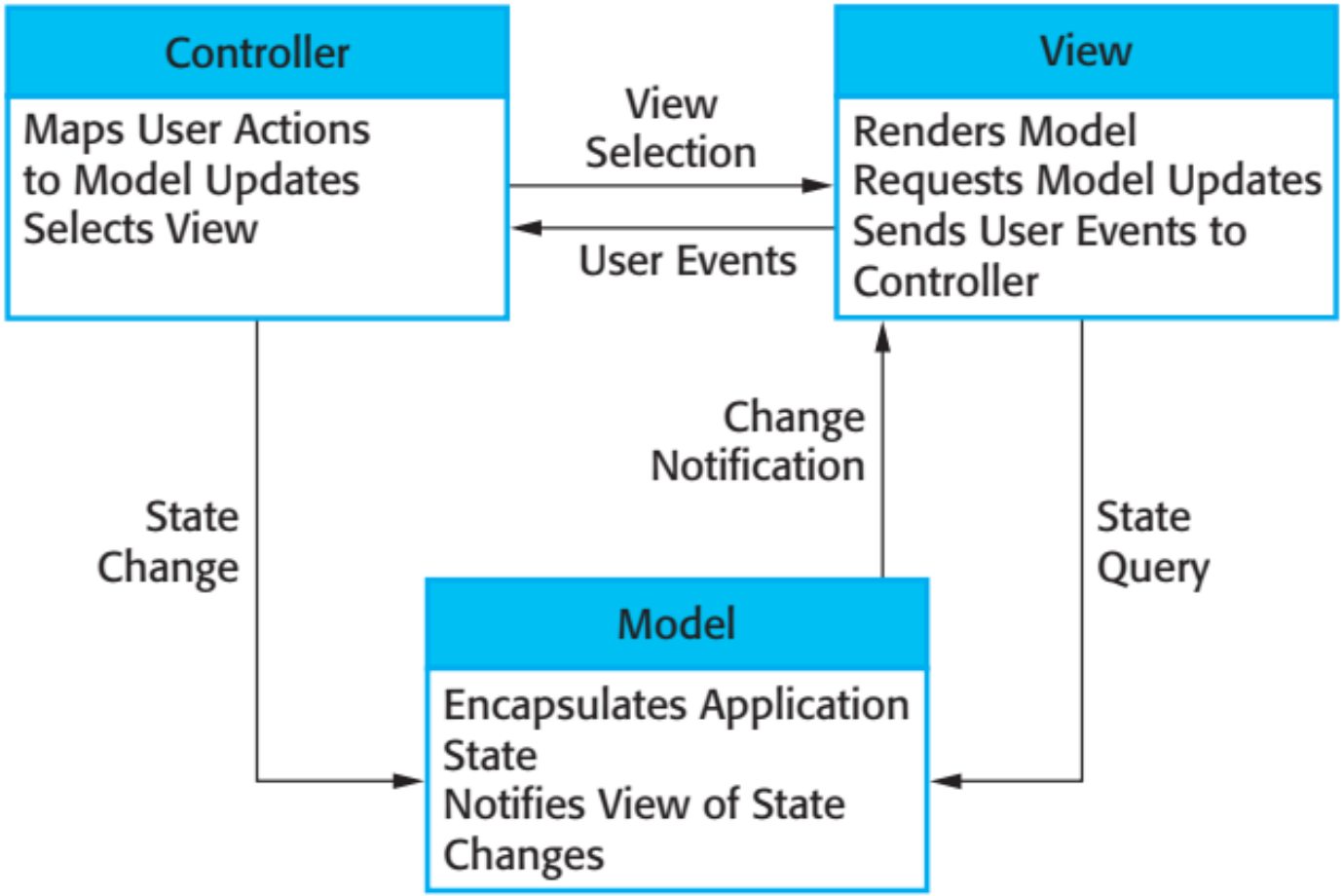
Model-Görünüm-Denetleyici (MVC) Deseni



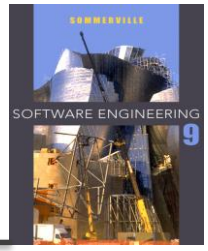
İsim	MVC (Model-Görünüm-Denetleyici)
Açıklama	Sunumu ve etkileşimi sistem verilerinden ayırır. Sistem, birbiriyle etkileşime giren üç mantıksal bileşene göre yapılandırılmıştır. Model bileşeni, sistem verilerini ve bu verilerle ilgili işlemleri yönetir. View bileşeni, verilerin kullanıcıya nasıl sunulacağını tanımlar ve yönetir. Denetleyici bileşeni, kullanıcı etkileşimini (ör. Tuşlara basma, fare tıklamaları vb.) yönetir ve bu etkileşimleri Görünüm ve Model'e aktarır. Şekil 6.3'e bakın.
Misal	Şekil 6.4, MVC modeli kullanılarak düzenlenen web tabanlı bir uygulama sisteminin mimarisini göstermektedir.
Kullanıldığında	Verileri görüntülemenin ve bunlarla etkileşim kurmanın birden fazla yolu olduğunda kullanılır. Ayrıca, verilerin etkileşimi ve sunumu için gelecekteki gereksinimler bilinmediğinde de kullanılır.
Avantajlar	Verinin temsilinden bağımsız olarak değişmesine izin verir ve bunun tersi de geçerlidir. Hepsinde gösterilen tek bir sunumda yapılan değişikliklerle aynı verilerin farklı şekillerde sunumunu destekler.
Dezavantajları	Veri modeli ve etkileşimler basit olduğunda ek kod ve kod karmaşıklığı içerebilir.

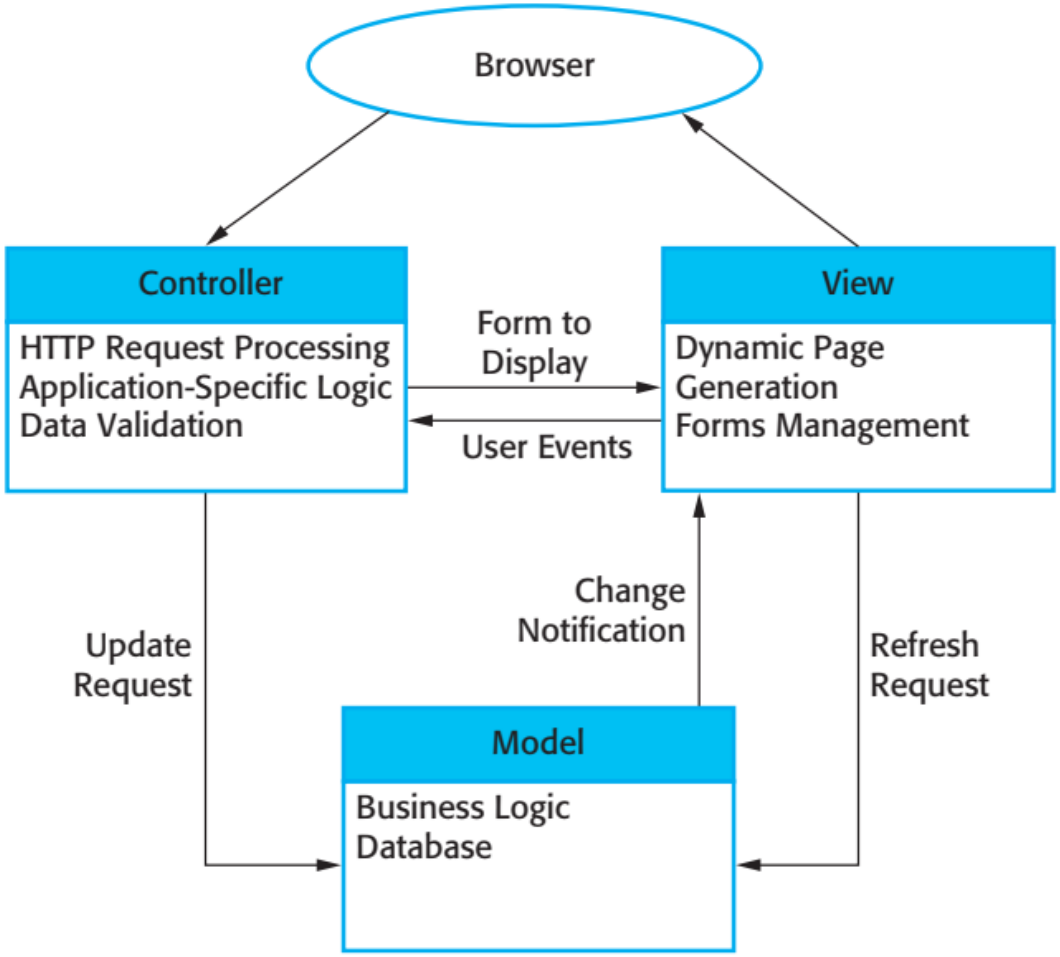
Model-Görünüm-Denetleyici Organizasyonu



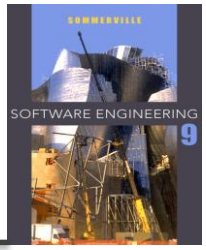


MGD(MVC) Modelini Kullanan Web Uygulaması Mimarisi





Katmanlı Mimari

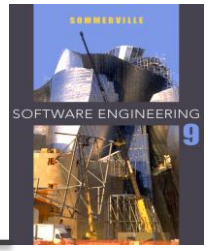


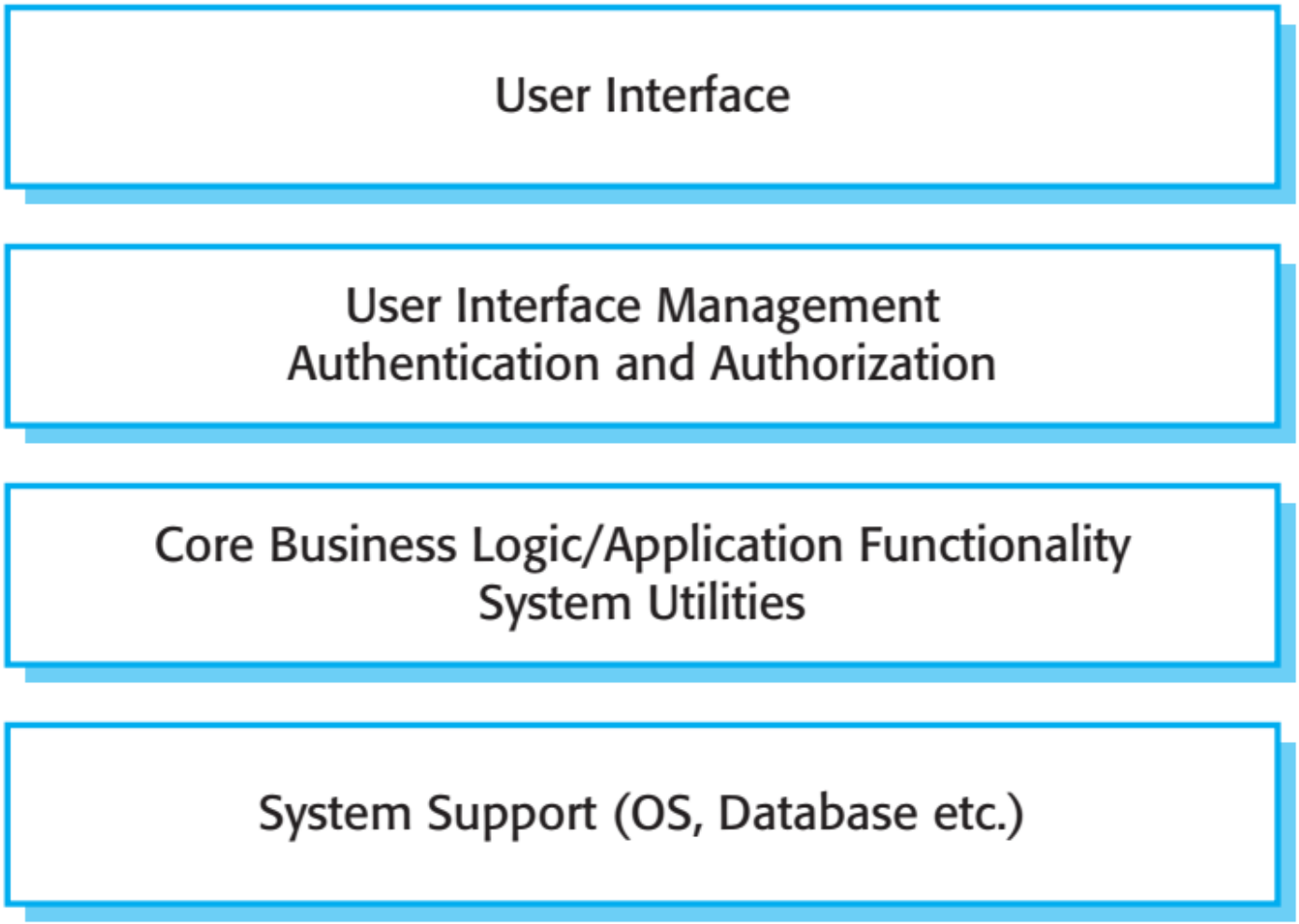
- Alt sistemlerin arayüzünü modellemek için kullanılır.
- Sistemi, her biri bir dizi hizmet sağlayan bir dizi katman (veya soyut makineler) halinde düzenler.
- Alt sistemlerin farklı katmanlardaki artımlı gelişimini destekler. Bir katman arayüzü değiştiğinde, yalnızca bitişik katman etkilenir.
- Ancak, sistemleri bu şekilde yapılandırmak için genellikle yapaydır.

Katmanlı Mimari Desen

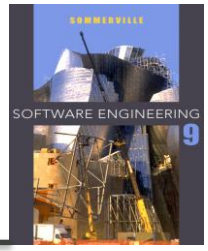
İsim Soyisim	Katmanlı mimari
Açıklama	Sistemi, her katmanla ilişkili ilgili işlevselliğe sahip katmanlar halinde düzenler. Bir katman, üstündeki katmana hizmetler sağlar, böylece en alt düzey katmanlar, sistem genelinde kullanılması muhtemel temel hizmetleri temsil eder. Şekil 6.6'ya bakın.
Misal	Şekil 6.7'de gösterildiği gibi, farklı kütüphanelerde tutulan telif hakkı belgelerini paylaşmak için bir sistemin katmanlı modeli.
Kullanıldığında	Mevcut sistemlerin üzerine yeni tesisler inşa ederken kullanılır; geliştirme, her takımın bir işlevsellik katmanından sorumlu olduğu birkaç ekibe yayıldığında; çok seviyeli güvenlik için bir gereksinim olduğunda.
Avantajlar	Arayüz korunduğu sürece tüm katmanların değiştirilmesine izin verir. Sistemin güvenilirliğini arttırmak için her bir katmanda yedek tesisler (örn. Kimlik doğrulama) sağlanabilir.
Dezavantajları	Uygulamada, katmanlar arasında temiz bir ayrım sağlamak genellikle zordur ve yüksek düzeyli bir katmanın, hemen altındaki katman yerine doğrudan alt düzey katmanlarla etkileşime girmesi gerekebilir. Performans, her katmanda işlenirken bir hizmet talebinin birden çok düzeyde yorumlanması nedeniyle bir sorun olabilir.

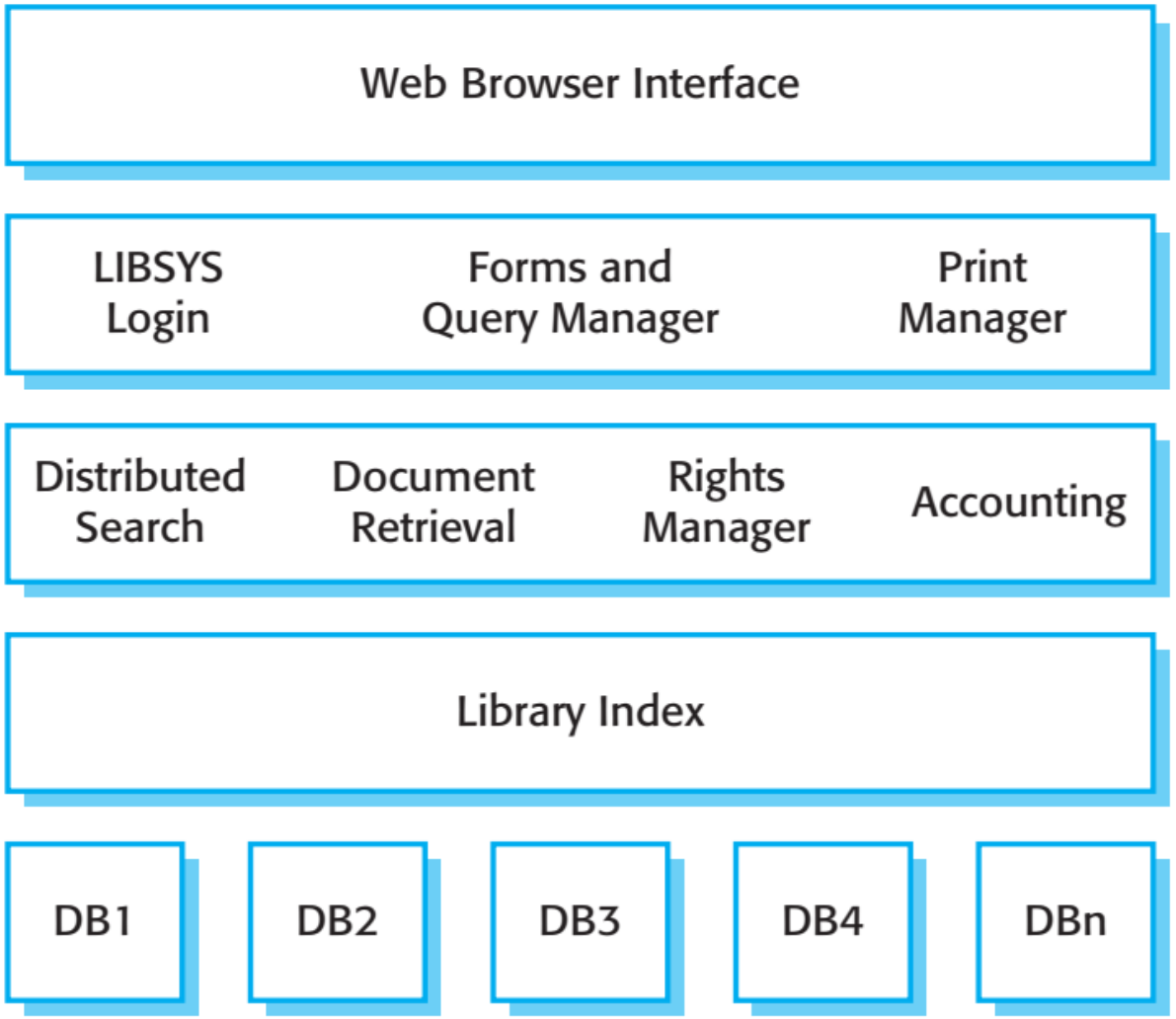
Genel Bir Katmanlı Mimari



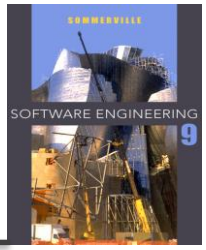


Kütüphane Yönetim Sisteminin Mimarisi

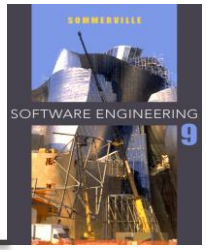




Bölüm 1'in Anahtar Noktaları



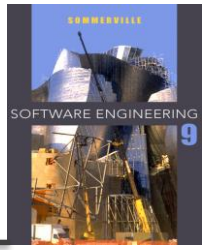
- Bir yazılım mimarisi, bir yazılım sisteminin nasıl organize edildiğinin bir açıklamasıdır.
- Mimari tasarım kararları, uygulama türü, sistemin dağılımı, kullanılacak mimari tarzlar hakkındaki kararları içerir.
- Mimariler, kavramsal bir görünüm, mantıksal bir görünüm, bir süreç görünümü ve bir geliştirme görünümü gibi birkaç farklı perspektif veya bakış açısıyla belgelenebilir.
- Mimari modeller, genel sistem mimarileri hakkındaki bilgileri yeniden kullanmanın bir yoludur. Mimariyi tanımlar, ne zaman kullanılabileceğini açıklar ve avantajlarını ve dezavantajlarını açıklarlar.



Ders 6 - Mimari Tasarım

Bölüm 2

Depo Mimarisi

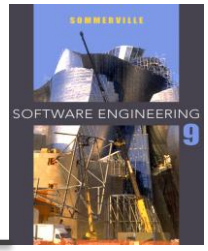


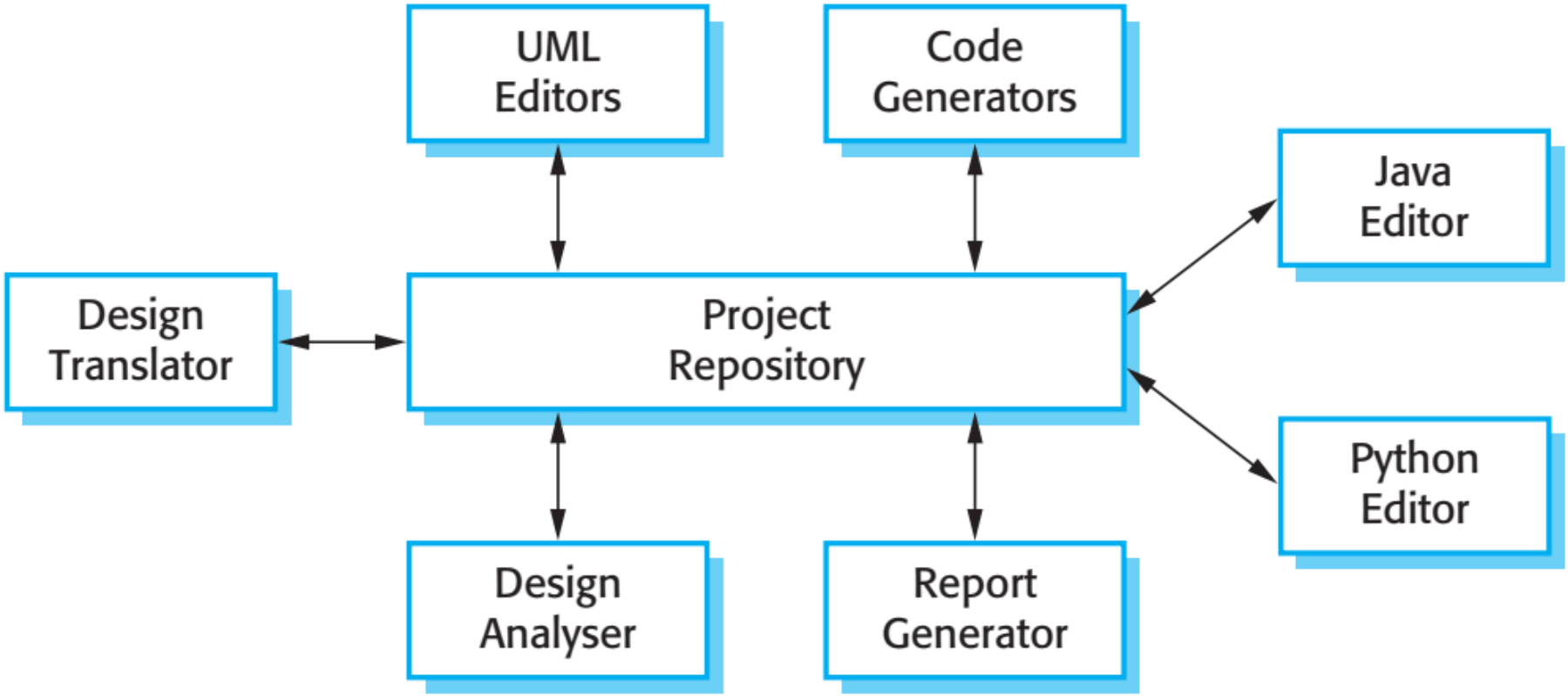
- Alt sistemler veri alışverişinde bulunmalıdır. Bu iki şekilde yapılabilir:
  - Paylaşılan veriler, merkezi bir veritabanında veya depoda tutulur ve tüm alt sistemler tarafından erişilebilir;
  - Her bir alt sistem kendi veritabanını korur ve verileri diğer alt sistemlere açıkça aktarır.
- Büyük miktarda veri paylaşılacağı zaman, depo paylaşımı modeli en yaygın şekilde kullanılır ve bu verimli bir veri paylaşım mekanizmasıdır.

Depo Deseni

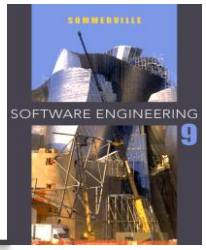
İsim Soyisim	Depo
Açıklama	Bir sistemdeki tüm veriler, tüm sistem bileşenlerinin erişebildiği merkezi bir depoda yönetilir. Bileşenler doğrudan bilgi havuzu aracılığıyla etkileşime girmez.
Misal	Şekil 6.9, bileşenlerin bir sistem tasarım bilgisi deposunu kullandığı bir IDE örneğidir. Her bir yazılım aracı, daha sonra diğer araçlar tarafından kullanılmak üzere mevcut olan bilgileri üretir.
Kullanıldığında	Bu modeli, uzun süre saklanması gereken büyük hacimli bilgilerin üretildiği bir sisteminiz olduğunda kullanmalısınız. Ayrıca, veri havuzuna veri dahil edilmesinin bir eylemi veya aracı tetiklediği veri odaklı sistemlerde de kullanabilirsiniz.
Avantajlar	Bileşenler bağımsız olabilir — diğer bileşenlerin varlığını bilmelerine gerek yoktur. Bir bileşen tarafından yapılan değişiklikler tüm bileşenlere yayılabilir. Hepsi tek bir yerde olduğu için tüm veriler tutarlı bir şekilde yönetilebilir (örneğin, aynı anda yapılan yedeklemeler).
Dezavantajları	Depo tek bir hata noktasıdır, bu nedenle depodaki sorunlar tüm sistemi etkiler. Arşiv aracılığıyla tüm iletişimin organize edilmesinde verimsizlikler olabilir. Depoyu birkaç bilgisayara dağıtmak zor olabilir.

Bir IDE İçin Bir Depo Mimarisi





İstemci-Sunucu Mimarisi

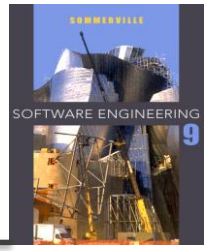


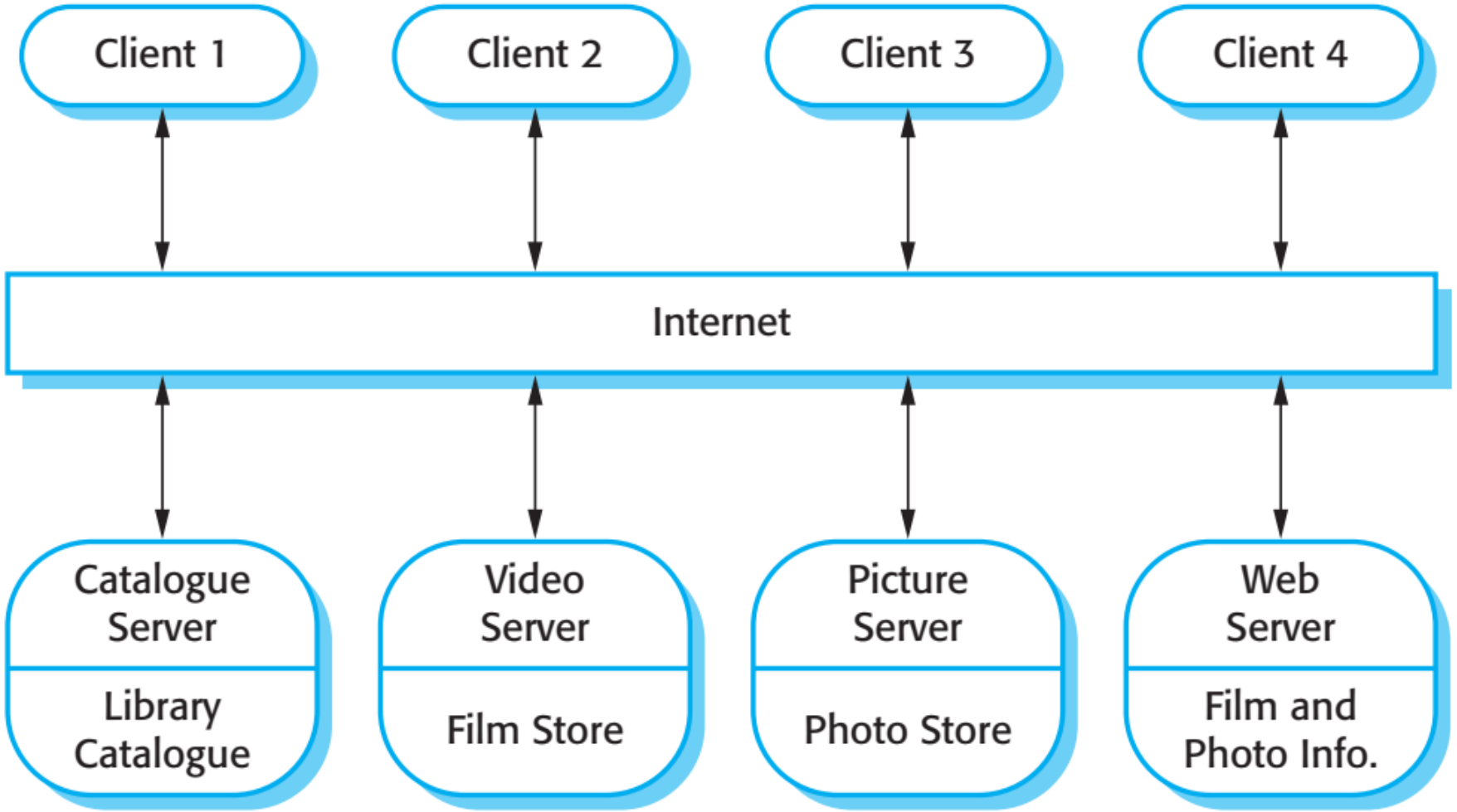
- Verilerin ve işlemenin bir dizi bileşene nasıl dağıtıldığını gösteren dağıtılmış sistem modeli.
  - Tek bir bilgisayarda uygulanabilir.
- Yazdırma, veri yönetimi vb. gibi belirli hizmetler sağlayan bağımsız sunucular seti.
- Bu hizmetleri arayan müşteri grubu.
- İstemcilerin sunuculara erişmesine izin veren ağ.

İstemci-Sunucu Deseni

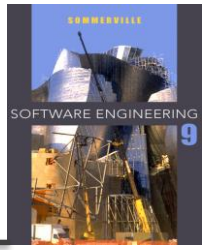
İsim Soyisim	Müşteri sunucusu
Açıklama	Bir istemci-sunucu mimarisinde, sistemin işlevselliği, her hizmetin ayrı bir sunucudan teslim edildiği hizmetler şeklinde düzenlenir. İstemciler bu hizmetlerin kullanıcılarıdır ve bunlardan yararlanmak için sunuculara erişirler.
Misal	Şekil 6.11, bir istemci-sunucu sistemi olarak düzenlenen bir film ve video / DVD kitaplığı örneğidir.
Kullanıldığında	Paylaşılan bir veritabanındaki verilere çeşitli konumlardan erişilmesi gerektiğinde kullanılır. Sunucular çoğaltılabildiğinden, bir sistem üzerindeki yük değişken olduğunda da kullanılabilir.
Avantajlar	Bu modelin temel avantajı, sunucuların bir ağ üzerinden dağıtılabilmesidir. Genel işlevsellik (örneğin, bir baskı hizmeti) tüm istemciler tarafından kullanılabilir ve tüm hizmetler tarafından uygulanması gerekmez.
Dezavantajları	Her hizmet tek bir hata noktasıdır, bu nedenle hizmet reddi saldırılarına veya sunucu arızasına açıktır. Sistemin yanı sıra ağa da bağlı olduğu için performans tahmin edilemez olabilir. Sunucular farklı kuruluşlara aitse yönetim sorunları olabilir.

Bir Film Kitaplığı İçin Bir İstemci-Sunucu Mimarisi



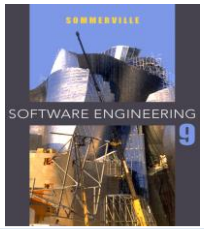


Boru ve Filtre Mimarisi



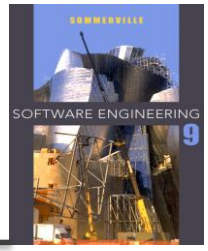
- İşlevsel dönüşümler, çıktı üretmek için girdilerini işler.
- Boru ve filtre modeli olarak adlandırılabilir (UNIX kabuğundaki gibi).
- Bu yaklaşımın çeşitleri çok yaygındır. Dönüşümler sıralı olduğunda, bu, veri işleme sistemlerinde yaygın olarak kullanılan bir seri sıralı modeldir.
- Etkileşimli sistemler için pek uygun değil.

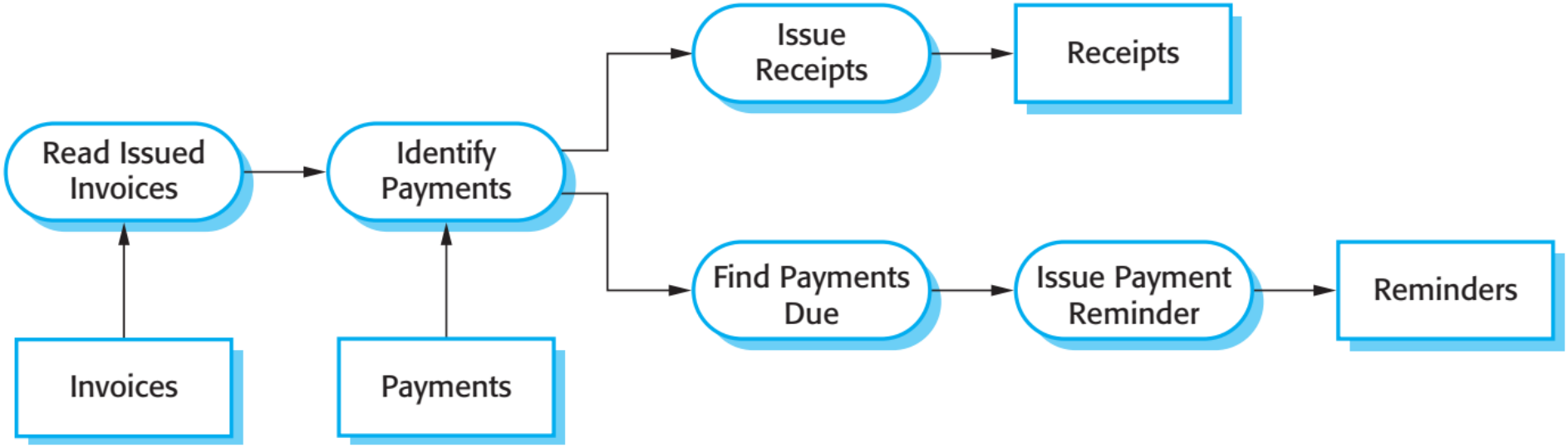
Boru ve Filtre Modeli



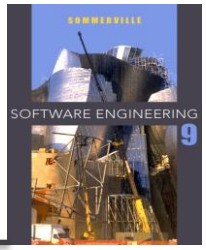
İsim Soyisim	Boru ve filtre
Açıklama	Bir sistemdeki verilerin işlenmesi, her bir işleme bileşeni (filtre) ayrı olacak ve bir tür veri dönüşümü gerçekleştirecek şekilde düzenlenir. Veri, işlenmek üzere bir bileşenden diğerine akar (bir boruda olduğu gibi).
Misal	Şekil 6.13, faturaları işlemek için kullanılan bir boru ve filtre sistemi örneğidir.
Kullanıldığında	Girişlerin ilgili çıktıları oluşturmak için ayrı aşamalarda işlendiği veri işleme uygulamalarında (hem toplu hem de işlem tabanlı) yaygın olarak kullanılır.
Avantajlar	Anlaşılması kolaydır ve dönüşümün yeniden kullanımını destekler. İş akışı stili, birçok iş sürecinin yapısıyla eşleşir. Dönüşümler ekleyerek evrim yapmak basittir. Sıralı veya eşzamanlı bir sistem olarak uygulanabilir.
Dezavantajları	Veri aktarımı için format, iletişim halindeki dönüşümler arasında kararlaştırılmalıdır. Her dönüşüm, girdisini ayrıştırmalı ve çıktılarını kararlaştırılan biçime göre ayrıştırmalıdır. Bu, sistem ek yükünü artırır ve uyumsuz veri yapılarını kullanan işlevsel dönüşümleri yeniden kullanmanın imkansız olduğu anlamına gelebilir.

Boru ve Filtre Mimarisine Bir Örnek



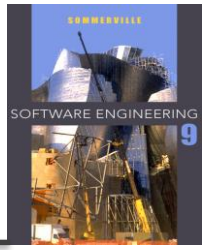


Uygulama Mimarileri



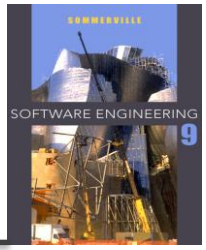
- Uygulama sistemleri, organizasyonel bir ihtiyacı karşılayacak şekilde tasarlanmıştır.
- İşletmelerin pek çok ortak noktası olduğu için, uygulama sistemleri de uygulama gereksinimlerini yansıtan ortak bir mimariye sahip olma eğilimindedir.
- Genel bir uygulama mimarisi, belirli gereksinimleri karşılayan bir sistem oluşturmak için yapılandırılabilen ve uyarlanabilen bir tür yazılım sistemi için bir mimaridir.

Uygulama Mimarilerinin Kullanımı



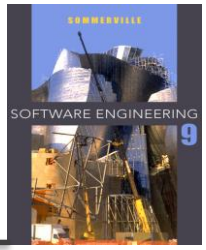
- Mimari tasarım için bir başlangıç noktası olarak.
- Tasarım kontrol listesi olarak.
- Geliştirme ekibinin çalışmalarını organize etmenin bir yolu olarak.
- Yeniden kullanım için bileşenleri değerlendirmenin bir yolu olarak.
- Uygulama türleri hakkında konuşmak için bir kelime dağarcığı olarak.

Uygulama Türlerine Örnekler



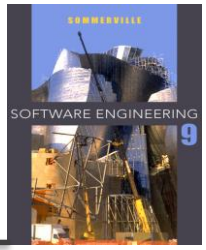
- Veri işleme uygulamaları
  - İşleme sırasında açık kullanıcı müdahalesi olmadan verileri toplu olarak işleyen veriye dayalı uygulamalar.
- İşlem işleme uygulamaları
  - Kullanıcı isteklerini işleyen ve bir sistem veritabanındaki bilgileri güncelleyen veri merkezli uygulamalar.
- Olay işleyici sistemler
  - Sistem eylemlerinin, sistem ortamındaki olayları yorumlamaya bağlı olduğu uygulamalar.
- Programlama dili işleme sistemleri
  - Kullanıcıların niyetlerinin sistem tarafından işlenen ve yorumlanan resmi bir dilde belirtildiği uygulamalar.

Uygulama Türü Örnekleri



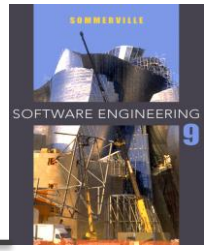
- Burada odak noktası, işlem işleme ve dil işleme sistemleridir.
- İşlem işleme sistemleri
  - E-ticaret sistemleri;
  - Rezervasyon sistemleri.
- Programlama dili işleme sistemleri
  - Derleyiciler;
  - Komut tercümanları.

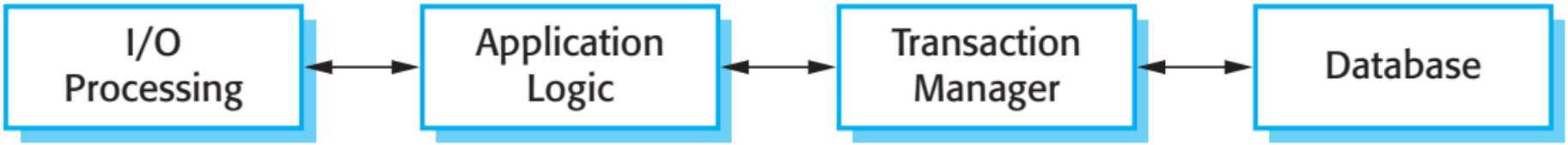
İşlem İşleme Sistemleri



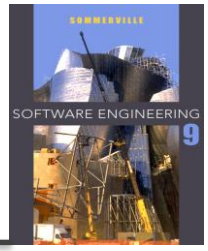
- Kullanıcıların bir veritabanından bilgi taleplerini veya veritabanını güncelleme taleplerini işleyin.
- Kullanıcı perspektifinden bir işlem şu şekildedir:
  - Bir hedefi karşılayan tutarlı işlemler dizisi;
  - Örneğin - Londra'dan Paris'e uçuşların zamanlarını bulun.
- Kullanıcılar, daha sonra bir işlem yöneticisi tarafından işlenen hizmet için eşzamansız isteklerde bulunur.

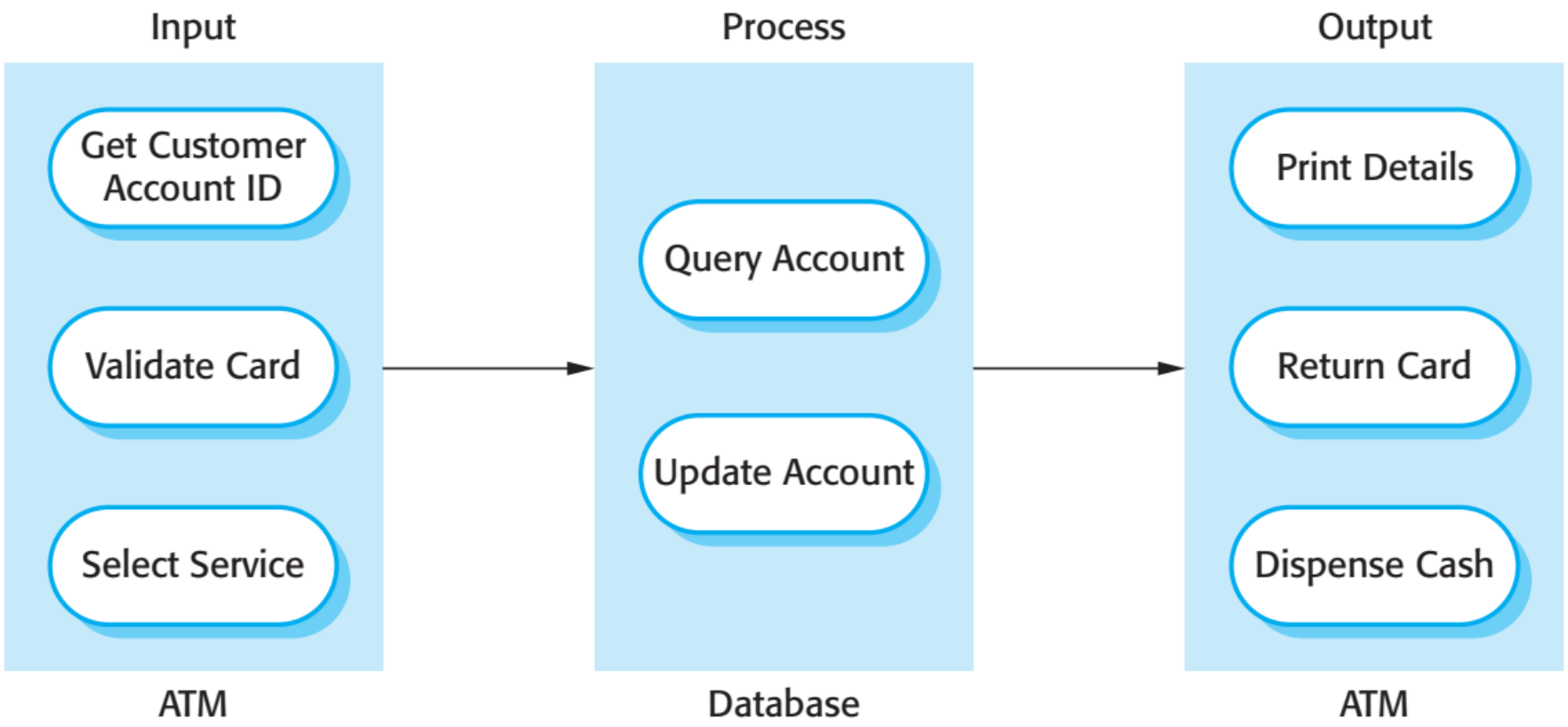
İşlem İşleme Uygulamalarının Yapısı



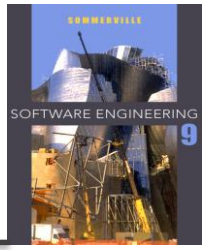


Bir ATM Sisteminin Yazılım Mimarisi



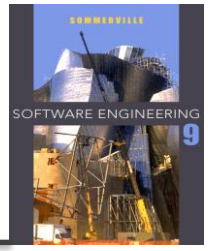


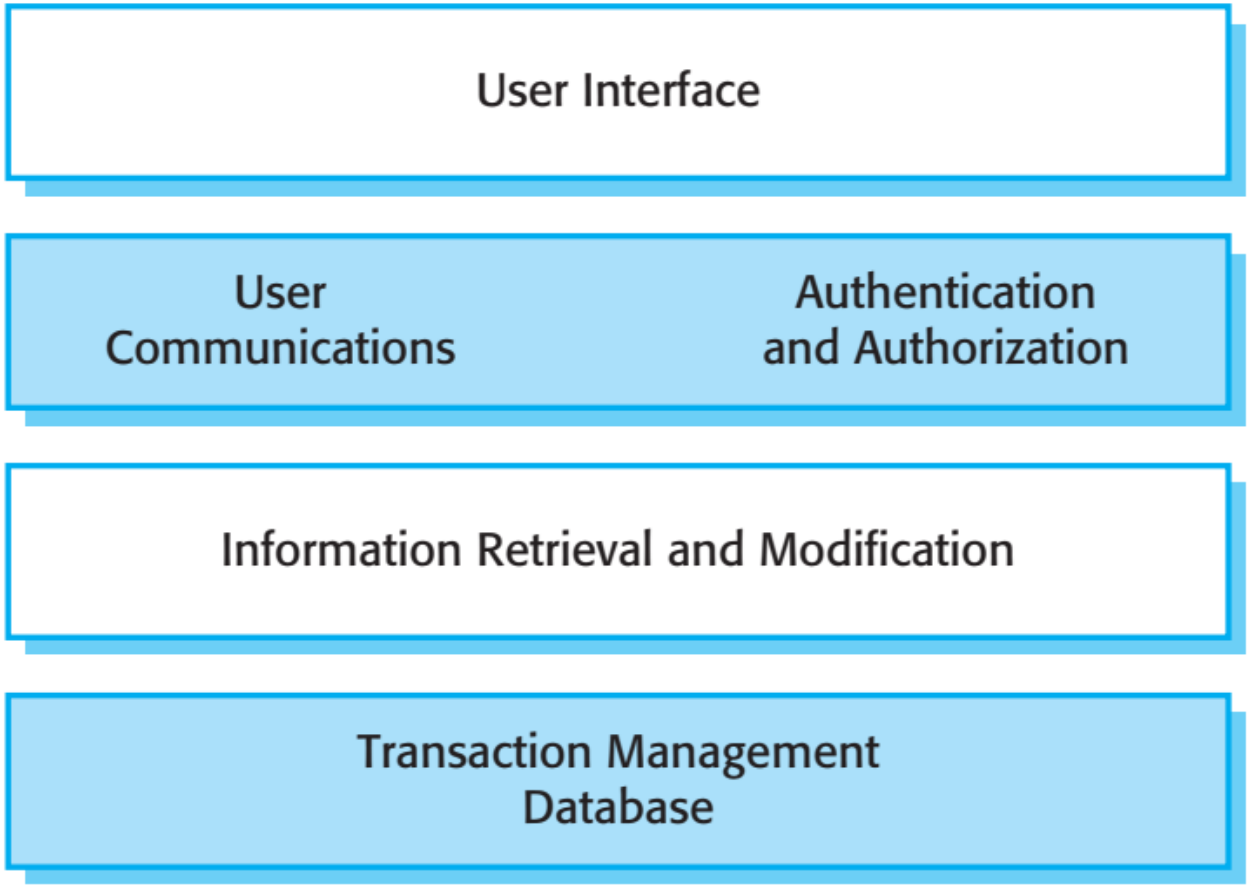
Bilgi Sistemleri Mimarisi



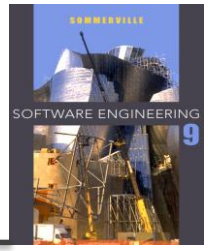
- Bilgi sistemleri, katmanlı bir mimari olarak organize edilebilen genel bir mimariye sahiptir.
- Bunlar işlem tabanlı sistemlerdir çünkü bu sistemlerle etkileşim genellikle veritabanı işlemlerini içerir.
- Katmanlar şunları içerir:
  - Kullanıcı arayüzü
  - Kullanıcı iletişimi
  - Bilgi alma
  - Sistem veritabanı

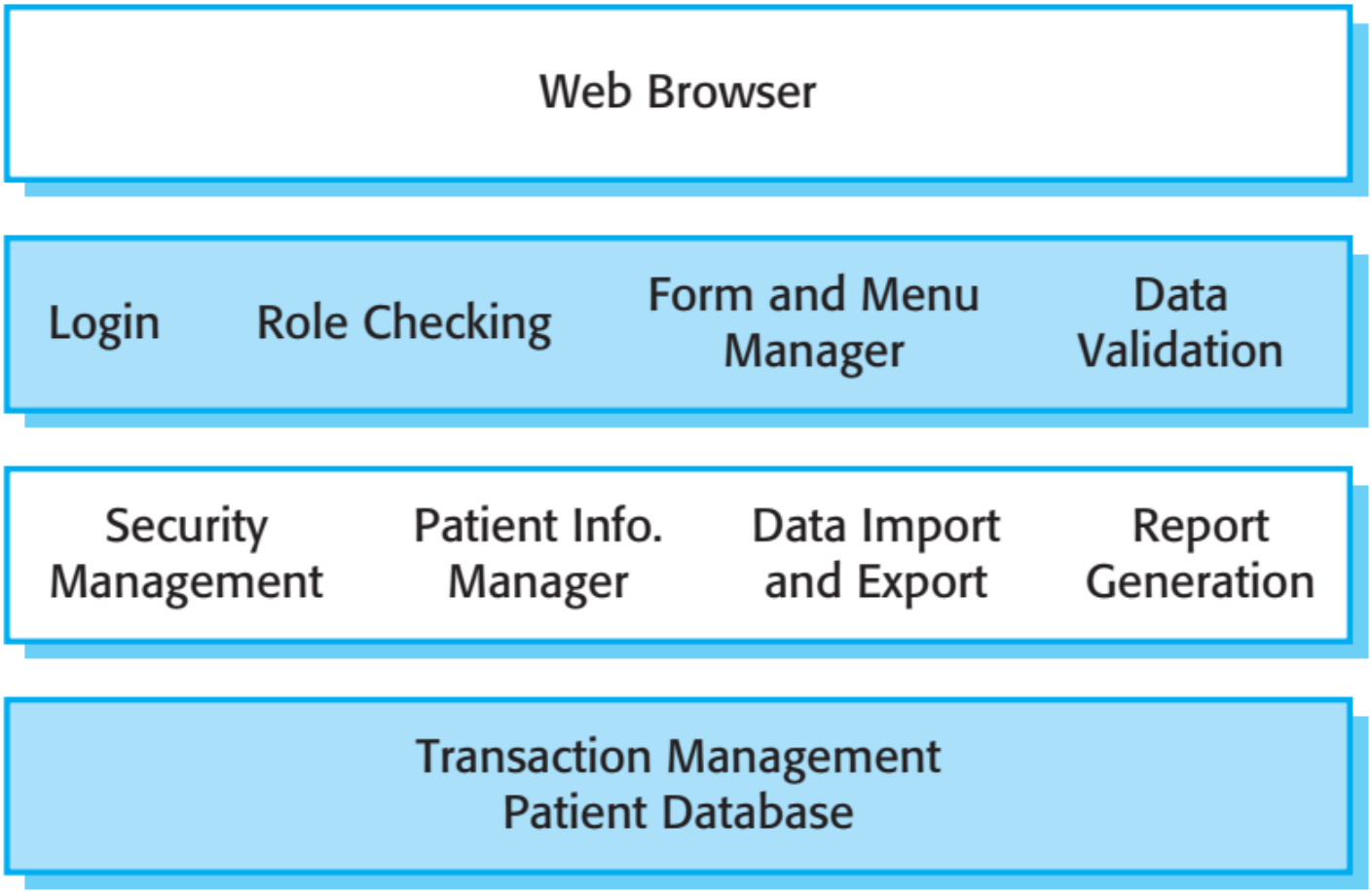
Katmanlı Bilgi Sistemi Mimarisi



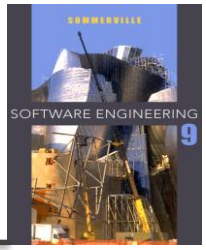


AH-HYS'nin Mimarisi



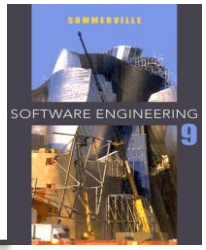


Web Tabanlı Bilgi Sistemleri



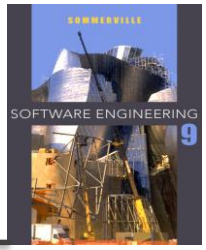
- Bilgi ve kaynak yönetimi sistemleri artık genellikle kullanıcı arayüzlerinin bir web tarayıcısı kullanılarak uygulandığı web tabanlı sistemlerdir.
- Örneğin, e-ticaret sistemleri, mallar veya hizmetler için elektronik siparişleri kabul eden ve daha sonra bu mal veya hizmetlerin müşteriye teslimatını düzenleyen İnternet tabanlı kaynak yönetim sistemleridir .
- Bir e-ticaret sisteminde, uygulamaya özgü katman, kullanıcıların bir dizi ürünü ayrı işlemlere yerleştirebileceği ve ardından tek bir işlemde hepsi için ödeme yapabileceği bir 'alışveriş sepetini' destekleyen ek işlevler içerir.

Sunucu Entegrasyonu



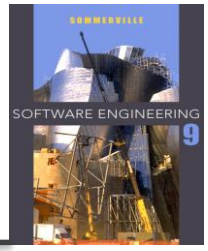
- Bu sistemler genellikle çok katmanlı istemci sunucu / mimariler olarak uygulanır (Ders 18'de tartışılmıştır)
  - Web sunucusu, bir web tarayıcısı kullanılarak uygulanan kullanıcı arayüzü ile tüm kullanıcı iletişimlerinden sorumludur;
  - Uygulama sunucusu, uygulamaya özel mantığın yanı sıra bilgi depolama ve erişim taleplerinin uygulanmasından sorumludur;
  - Veritabanı sunucusu, bilgileri veritabanına ve veritabanından taşır ve işlem yönetimini gerçekleştirir.

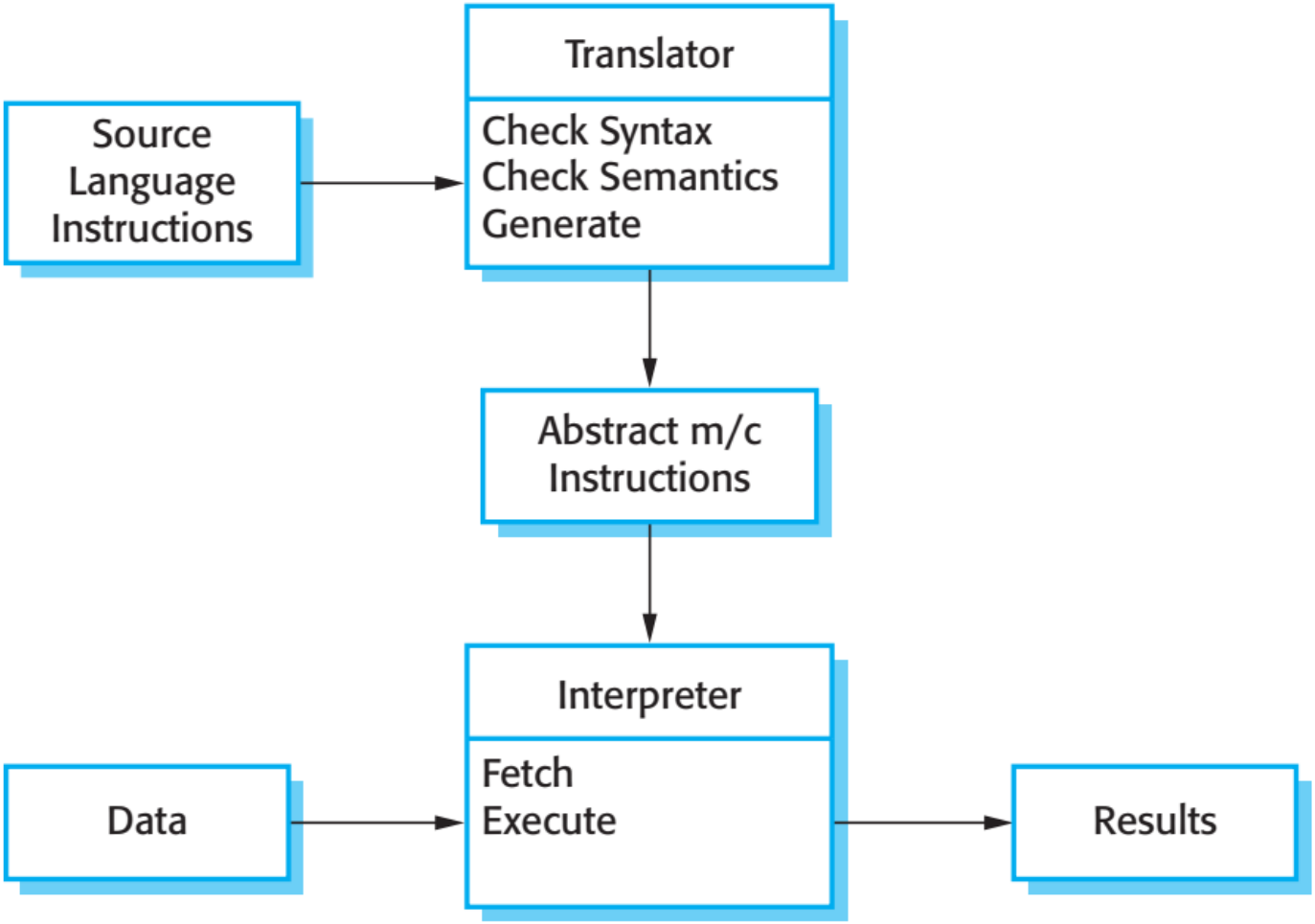
Dil İşleme Sistemleri



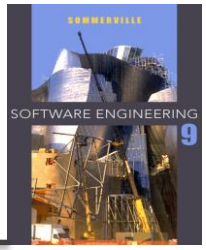
- Doğal veya yapay bir dili girdi olarak kabul edin ve o dilin başka bir temsilini oluşturun.
- İşlenmekte olan dildeki talimatlara göre hareket edecek bir tercüman içerebilir.
- Bir problemi çözmenin en kolay yolunun bir algoritmayı veya sistem verilerini tanımlamak olduğu durumlarda kullanılır
  - Meta durum araçları, araç açıklamalarını, yöntem kurallarını vb. işler ve araçlar üretir.

Bir Dil İşleme Sisteminin Mimarisi



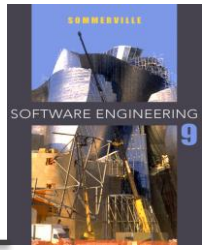


Derleyici Bileşenleri



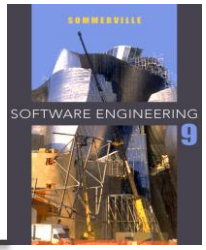
- Giriş dili belirteçlerini alan ve bunları dahili bir forma dönüştüren bir sözcük analizcisi.
- Çevrilen metinde kullanılan varlıkların adları (değişkenler, sınıf adları, nesne adları vb.) hakkında bilgi tutan bir sembol tablosu.
- Çevrilen dilin sözdizimini kontrol eden bir sözdizimi çözümleyicisi.
- Derlenmekte olan programı temsil eden dahili bir yapı olan bir sözdizimi ağacı.

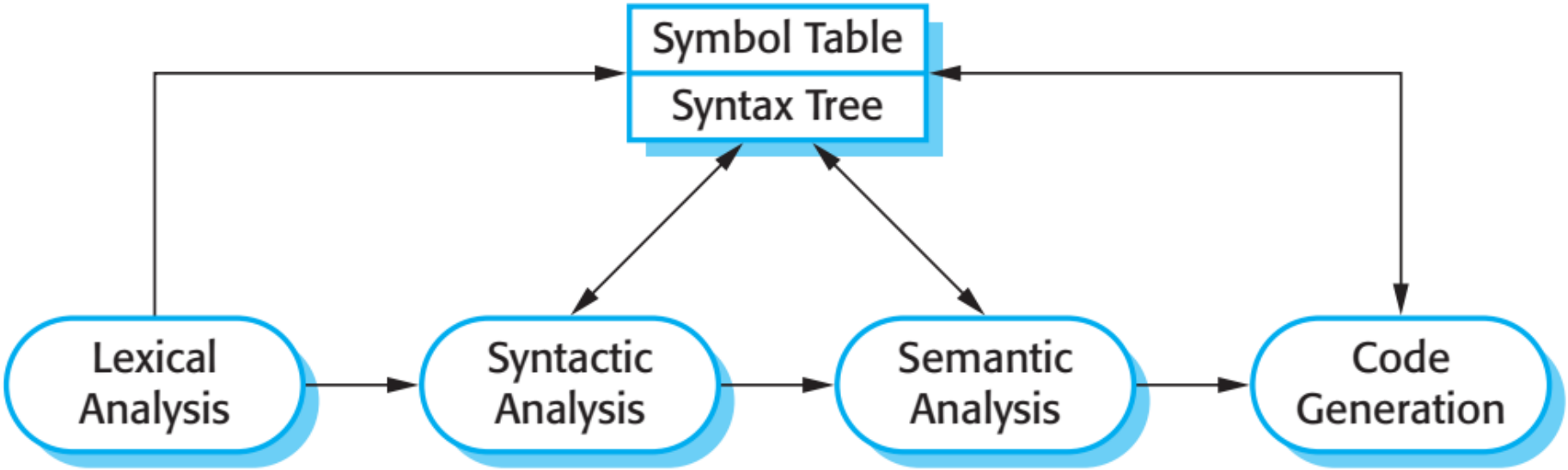
Derleyici Bileşenleri



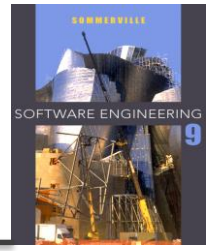
- Giriş dili metninin anlamsal doğruluğunu kontrol etmek için sözdizimi ağacından ve sembol tablosundan gelen bilgileri kullanan bir anlamsal analizci.
- Sözdizimi ağacında 'yürüyen' ve soyut makine kodu üreten bir kod üreteci.

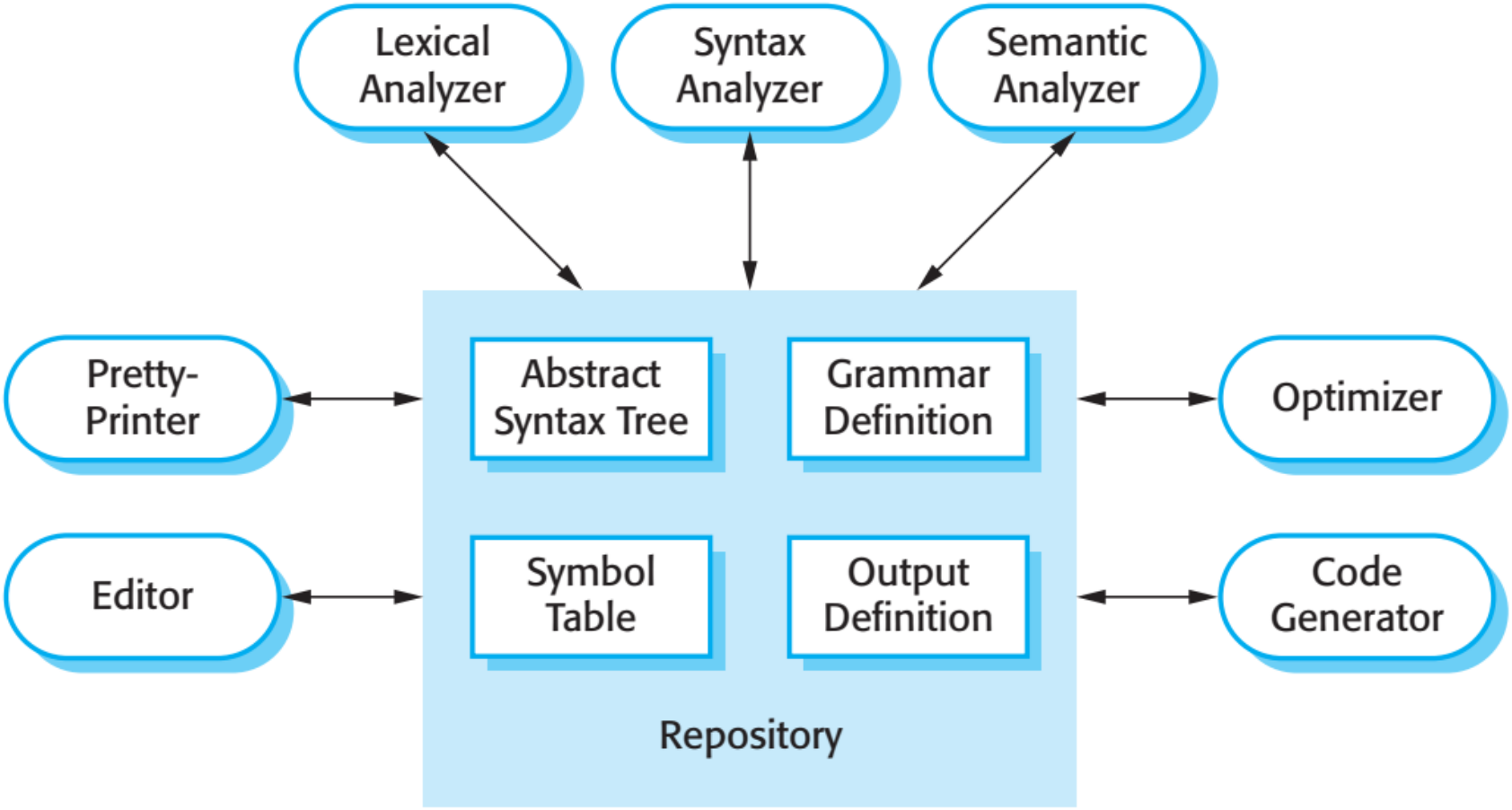
Bir Boru ve Filtre Derleyici Mimarisi



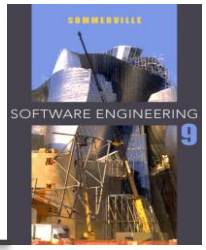


Bir Dil İşleme Sistemi İçin Bir Havuz Mimarisi





Bölüm 2'nin Anahtar Noktaları



- Uygulama sistemi mimarilerinin modelleri, uygulamaları anlamamıza ve karşılaştırmamıza, uygulama sistemi tasarımlarını doğrulamamıza ve yeniden kullanım için büyük ölçekli bileşenleri değerlendirmemize yardımcı olur.
- İşlem işleme sistemleri, bir veri tabanındaki bilgilere uzaktan erişilmesine ve birkaç kullanıcı tarafından değiştirilmesine izin veren etkileşimli sistemlerdir.
- Dil işleme sistemleri, metinleri bir dilden diğerine çevirmek ve giriş dilinde belirtilen talimatları gerçekleştirmek için kullanılır. Bir çevirmen ve üretilen dili çalıştıran soyut bir makine içerirler.