Yazılım Mimarisi nedir?

-Yazılım öğelerini,

,302 onların arasındaki ilişkileri içeren yapılar kümesidir.

Bir sistemin yazılım mimarisi nelerden oluşur?

-Bir dizi yapılardan oluşur. Sistem hakkında fikir yürütebilmemiz için kullanılır.

Mimari Yapıların 3 Önemli Kategorisi?

-Modül

-Allocation

-C&C

Yazılım Mimarisi Tasarımı(hi-level design): Yazılımın üst düzey yapısını ve organizasyonunu geliştirir ve çeşitli bileşenleri tanımlar. Daha yüksek bir yapı.

Software Detailed Design: Yapıyı kolaylaştırmak için her bileşeni yeterli ayrıntıda belirtir

ABSTRACTİON(Soyutlama):

-Bir şeyin önemli taraflarını ortaya çıkarıp, daha az önemli olan kısımlarını görmezden gelen model.

-Matematik ve diğer bilimler için ---> İnformation İgnoring (Verilmek istenen çıkarım sağlamak)

-Bilgisayar için ---> İnformation Hiding-Örtme (Etkileşim kalıpları ortaya çıkarmak)

Abstraction Özellikleri:

1-)Seperation(Ayırma): Önemlileri al, önemsiz ve çelişkilileri at. Amaç ile mekanizma ayrımı. Tasarım ile implementation(hayata geçirme) ayrımı.

2-)Relevancy(İlgililik): İlgilileri al, ilgisizleri at

Abstraction Faydaları:

-Karmaşıklığı ortadan kaldırmak

-İspat desteği(örn kimyada NŞA)

-Maliyeti azaltır

Hangi Yapıları Mimari Kabul Ediyoruz?

-Sistem hakkında akıl yürütmeye hizmet etmeli

-Stakeholderları ilgilendiren taraf olmalı, stakeholderlar için önemli olmalı.

Aşağıda Verilen Açıklamalar Hangi Kalıplara Aittir? (Veya tam tersi)

Dağıtık Mimari ---> Belli başlı uygulama bileşenleri, daha küçük ayrı olarak yerleştirilen birimlere bölünür. Monolitik uygulamalara uygun bir model.

Mikrohizmet Mimarisi ---> Yapısında dağıtık mimari olan, tüm bileşenleri birbirinden ayrı ve uzaktan erişim protokolüne sahip model.(örn JMS,AMQP,REST,SOAP,RMI)

Mikrohizmet Mimarisi ---> İki ana kaynaktan evrilmiştir. Katmanlı Mimari Kalıp kullanılarak geliştirilen Monolitik Uygulamalar ve Hizmet Odaklı Mimari Kalıp aracılığı(SOA) ile geliştirilen Dağıtık Uygulamalar.

Mikrohizmet Mimarisi ---> Hizmet kavramını basitleştirerek, düzenleme ihtiyaçlarını ortadan kaldırarak karmaşıklığın üstüne gider.

Genel mimari modeline uygulanan birkaç ortak temel kavram vardır? Bu kavramlardan ilki ayrı yerleştirilmiş birimler kavramıdır.

---> Hız bileşenlerinin granülize edilmesi

SOA ---> Mimari yapıları karşılaştırırken, güçlü bir yapısı vardır,eşsiz düzeyde soyutlama, heterojen bağlantı, hizmet düzenlemesi, karmaşık, pahalı, uygulanımı zor, her yerde bulunur, genelde çoğu uygulama için fazla gelmektedir.

Bir mikrohizmet (microservices) mimarisinin en büyük zorluklarından biri?

- Hız bileşeninin doğru ayrıntı düzeyini tasarlamak

SOA(Hizmet Odaklı Mimari)'nın Zayıflıkları:
SOA tabanlı sistemleri inşa etmek genellikle çok karmaşık ve zordur.

Bağımsız hizmetlerin gelişimini kontrol etmiyorsunuz

Performans konusunda bir garantisi yok.

SOA'nın Temel Bileşenleri Nedir?

1-) Hizmet Sağlayıcıları (Service Providers/Clients)

2-) Hizmet Tüketicileri (Service Consumers)

3-) ESB

4-) Orchestration Server

5-) Registry of Services

Soa'nın ana faydası ve ana itici gücü nedir?

-Birlikte çalışabilirliktir (interoperability)

Aşağıdaki Problem İçin Hangi Mimari Yapı Önerilir?

Problem: Ultra büyük veri kümelerine sahip birçok uygulama için verileri sıralamak ve ardından gruplanmış verileri analiz etmek yeterlidir**. (**For many applications with ultra large data sets, sorting the data and then analyzing the grouped data is sufficent.)

Solution: The Map Reduce Pattern

Map-Reduce Kalıbının Elemanları?

Map Kısmı: Analiz çıkarma ve dönüştürme kısımlarını gerçekleştiren, birden çok işlemciye dağıtılmış, birden çok örneğe sahip bir işlevdir. (extract transformation)

Reduce Kısmı: Çıkarma-dönüştürme-yüklemenin, yük kısmını gerçekleştirmek için işlemciler arasında tek bir örnek olarak veya birden çok örnek olarak dağıtılabilen bir işlevdir.

The İnfrastructure: Altyapı, kurulumdan ve bulut sunucularından, aralarındaki verileri yönlendirmekten ve arızayı tespit edip kurtarmaktan sorumlu çerçevedir.

Dağıtık işlem yaparken, yükü mümkün olduğunca eşit dağıtmaya yarayan teknik (Hashing)?

-Map Reduce

Map Reduce Pattern Zayıflıkları?

-Performans

Aşağıda Verilen Resimler Hangi Kalıbın Modelidir?

-Layered Pattern (Presentation ---> Buisness ---> Persistence---> Database)

-Client-Server Pattern (ATM-Banka)

-Service Orianted Pattern (Web Browser)

-Map Reduce Pattern

-Microservices Pattern (User İnterface Layer/Client Components)

Program Karmaşıklığını/Girifitliğini Etkileyen Faktörler?

-Modülün büyüklüğü

-Programın modülleri arası ilişki

-Modülün iç unsurları arası ilişki

Modüller Arası İrtibatın İrdelenmesi İçin İki Temel Ölçü Vardır?

1-) **Cohesion --->** Modülün içindeki işlevsel sıkılığını inceler. O modül tek işlev yapsın der. Bir modül içindeki öğelerin ne kadar güçlü bağa sahip olduğunu ölçer.

2-) **Coupling --->** Modüller arası bağlılık.

Modüller arası bağımsızlık düzeyini **COUPLİNG** ile ölçüyoruz.

Bir modülün öğeleri arasındaki ilişkinin gücünü **COHESİON** ile ölçüyoruz.

Coupling Türleri ve Açıklaması?

1-Gevşek Bağ (Loosely Coupled): Modüller arasında küçük etkileşim

2-Kuvvetli Bağ (Tightly Coupled): Modüller arasında güçlü etkileşim. Yüksek sıklılık (highly cohesive).

3-Veri Bağı Data (Coupled): İki modül birbiriyle parametre kullanarak geçiş yapıyorsa ve bir dizi,değişken veya tablo aracılığı ile iletişim kuruyorsa.

4-Kontrol Bağı (Control Coupled): Bir modülden gelen veriler, diğerinin talimat icra sırasını yönlendirmek için kullanılıyorsa, bu iki modül kontrol bağlıdır.(örn bir modülde frag)

5-İçerik Bağı (Content Coupled): Bir modül başka modüle atıfta bulunuyorsa veya başka modülü değiştiriyorsa bu iki modül içerik bağlıdır.(örn bir modülden diğerine dallanma)

Cohesion Türleri ve Açıklaması?

1-İşlevsel Türdeşlik (Functional Cohesion): Bir modüldeki her öğe, bir ve yalnızca bir işlevin gerekli ve temel bir parçasıdır. Modülün her parçası diğerleriyle ilişkilidir ve modül, çalışması için gerekli olan her şeyi içerir.

2-Dayalı Sıralı Türdeşlik (Sequantial Cohesion): Bir modülün öğeleri, bir işlemin çıktısının diğerinin girdisi olduğu işlemler dizisinin farklı bölümlerinin ifasıyla ilişkilidir.

3-Sıralı Türdeşlik (Procedural Cohesion): Bir modülün öğelerinin tümü bir prosedürün parçasıdır. Belirli bir sırada atılması gereken belirli adımlar dizisi.

Tesadufi (Coincidental) **en zayıf ve en az istenen seviyedir.**

İşlevsel (Functional) **en güçlü ve en çok istenen seviyedir.**

Kalite nitelikleri talep ve ihtiyaçları bir uygulamanın işlevsel gereksinimlerinin nasıl karşılandığının birçok yüzünü yakalayan işlevsel olmayan talep ve ihtiyaçalrın parçasıdır.

Nonfunctional Requirments (Kağıdın boyutu,rengi)

Güvenlik Kalite Unsurları

1-Kimlik Doğrulama(Authentication): Kimliği doğrulanmış kullanıcılar erişim hakkına sahip

*Kimlik Doğrulama Tekniklerinin özellikleri:

Bildiğin bir şey - Taşıdığın Bir şey - Senin olan bir şey

2-Bütünlük:

3-Gizlilik:

4-İnkar Edilememe:

Güvenlik İçin Nelere Bakarız?

-Gönderdiğim yer ve kaynak doğru mu, genel veriye ortadaki adam erişebilir mi?

Birlikte Çalışılabilirlik(İnterobility) Nedir?

-İki veya daha fazla sistemin belirli bir bağlamda, arayüzler aracılığı ile anlamlıbilgi alışverişinde bulunma derecesi.

Birlikte Çalışılabilirliğin İki Önemli Tarafı Vardır?

1-Keşif: Bir hizmetin müşterisi yeri, kimliği ve hizmet arayüzünü keşfetmelidir.(koşu vaktinde veya öncesinde)

2-Tepkileri çekip çevirmek: 3 ayrı ihtimal var:

-hizmet, yanıtla birlikte istekte bulunana geri bildirimde bulunur

-Hizmet tepkisini bir başka sisteme gönderir.

- hizmet, yanıtını ilgili taraflara yayınlar.