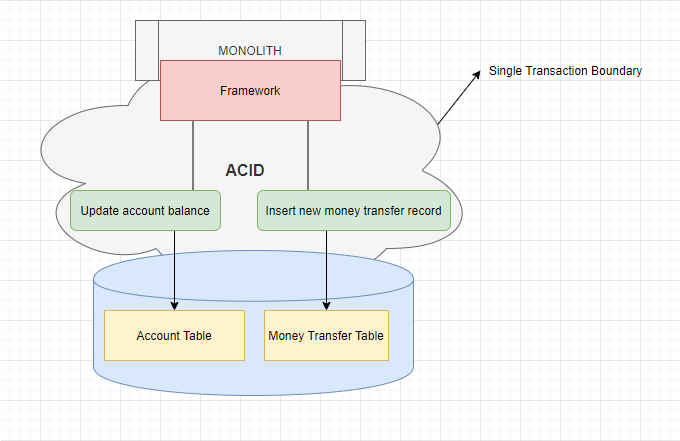
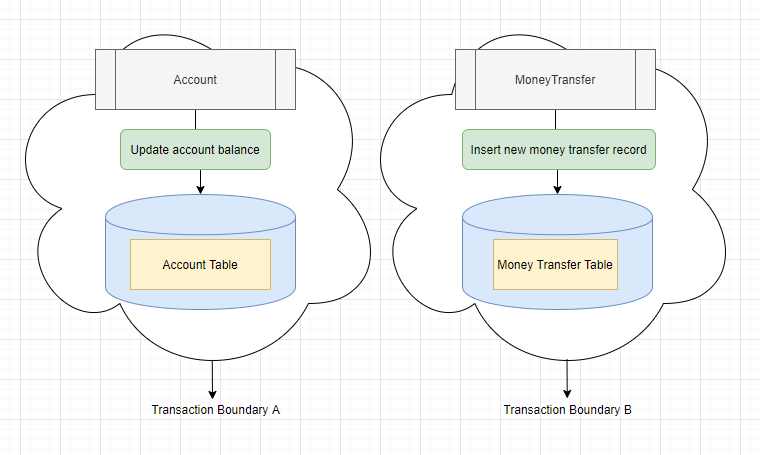
# SAGA PATTERN

**TRANSACTION TÜRLERİ:**

**MONOLOTHİC**

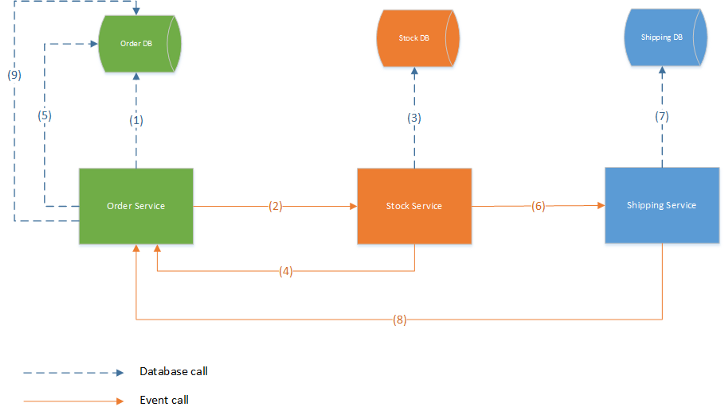


**DISTRIBUTED TRANSACTION**



## KOREOGRAFI

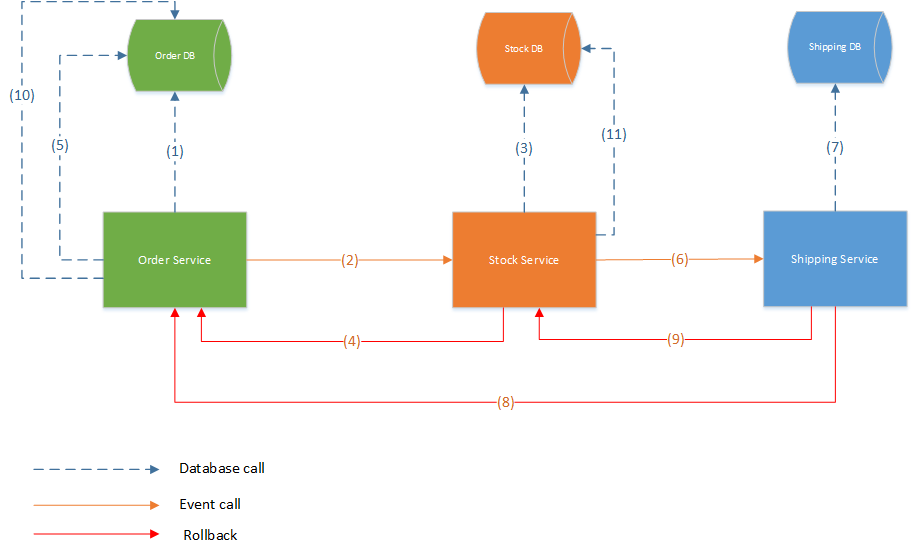
1. Koreografi:Danstaki adımların veya hareketlerin bir dizisidir.
2. Her hizmet diğer hizmetlerin olaylarını üretir ve dinler, diğer hizmetlerden gelen olaya göre bir sonraki eyleme karar verir.



**HATASIZ SENARYO:**

1. SİPARİŞ servisi veritabanında durumu DOĞRULANIYOR olan bir kayıt oluşturur.
2. SİPARİŞ servisi SİPARİŞ BAŞARIYLA OLUŞTURULDU EVENT’ini yayınlar.STOK servisi bu EVENT’i dinler.
3. STOK servisi, STOK veritabanındaki ÜRÜN SAYISINI günceller.
   1. STOK servisi EVENT fırlatır.SİPARİŞ ve SEVKIYAT servisleri bu eventi dinler.
4. SİPARİŞ servisi , STOKC eventini yakalar.
5. SİPARİŞ servisi siparişin durumunu SEVKIYAT olarak günceller.
6. GÖNDERİ servisi,STOK eventini yakalar.
7. GÖNDERİ servisi veritabanında kayıt oluşuturur.
8. GÖNDERİ servisi ,GÖNDERİ BAŞARIYLA KAYDEDİLDİ EVENT’i yayınlar.SİPARİŞ servisi bu olayı dinler .
9. SİPARİŞ servisi ,SİPARİŞ durumunu BİTİR olarak değiştirir.

**HATALI SENARYO:**



**SENARYO 1: Stok servisinde hata oluştu**

1. SİPARİŞ servisi veritabanında durumu DOĞRULANIYOR olan bir kayıt oluşturur.
2. SİPARİŞ servisi SİPARİŞ BAŞARIYLA OLUŞTURULDU EVENT’ini yayınlar.STOK servisi bu EVENT’i dinler.
3. STOK servisi, STOK veritabanındaki ÜRÜN SAYISINI güncellerken hata oluştu.
4. STOCK servisi ROLLBACK EVENT’i yayınlar.
5. SİPARİŞ servisi veritabanındaki kaydı BAŞARISIZ olarak işaretler.

**SENARYO 2:SİPARİŞ ve STOK servisleri başarıyla işlem gördü.SEVKIYAT servisinde hata oluştu.**

1. SEVKIYAT servisi kayıt oluştururken hata oluştu.
2. SEVKIYAT servisinin yayınlamış olduğu ROLLBACK eventini dinleyen SİPARİŞ servisi eventi yakalar.
3. SEVKIYAT servisinin yayınlamış olduğu ROLLBACK eventini dinleyen STOK servisi eventi yakalar.
4. SİPARİŞ servisi veritabanındaki kaydı BAŞARISIZ durumuna çeker.
5. STOK servisi veritabanındaki ürün sayısını önceki değerine çeker.

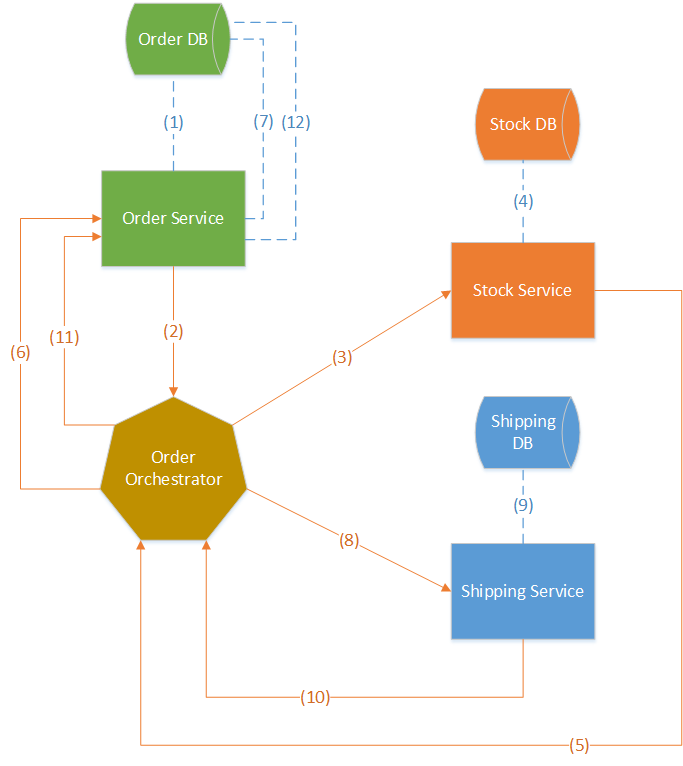
**HER TRANSACTION için ORTAK BİR ID paylaşılır.Hata fırlatıldığında her servis hangi TRANSACTION ile ilgili olduğunu bilir.**

**2-4 arası TRANSACTION içeren yapılar için öneriliyor.Sayı arttıkça event kompleksitisi artıyor.**

## ORKESTRASYON

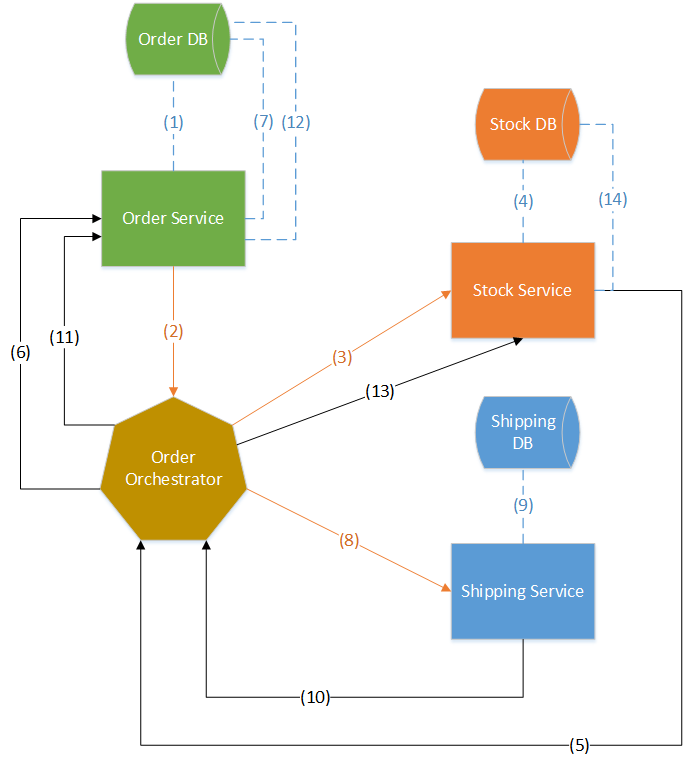
1. SAGA modelin karmaşıklığını ele almak için,ORKESTRATÖR olarak bir süreç yöneticisi eklemek normaldir.
2. SÜREÇ YÖNETİCİSİ
   1. OLAYLARI DİNLEMEKTEN ve UÇ NOKTALARI tetiklemekten sorumludur.
3. SİPARİŞ ORKESTRATÖRÜ -- İŞLEM YÖNETİCİSİ – SÜREÇ YÖNETİCİSİ
   1. Hangi işlemin ne zaman gerçekleştirilmesi gerektiğini söyleme için her SERVİS ile KOMUT/YANIT yöntemiyle iletişim kurar.

**BAŞARILI SENARYO:**



1. SİPARİŞ servisi veritabanında DOĞRULANIYOR durumuyla bir sipariş kaydı oluşturur.
2. SİPARİŞ servisi ,SİPARİŞ ORKESTRATÖRÜ’ne “ SİPARİŞ BAŞLATMAK “ eventini yayınlar.
3. SİPARİŞ ORKESTRATÖRÜ , STOK servisine “STOKU GÜNCELLE KOMUTU” gönderir.
4. STOK servisi ,veri tabanındaki ilgili kaydı günceller.
5. STOK servisi,SİPARİŞ ORKESTRATÖRÜ’ne “STOK GÜNCELLENDİ” eventini yayınlar.
6. SİPARİŞ ORKESTRATÖRÜ , STOK’un başarıyla güncelenen bilgileriyle SİPARİŞ servisine “SİPARİŞ GÜNCELLE” eventi gönderir.
7. SİPARİŞ servisi veritabanında kaydı “NAKLİYE” olarak günceller.
8. SİPARİŞ ORKESTRATÖRÜ ,GÖNDERİ servsine “GÖNEERİ KAYDETME” eventi gönderir.
9. GÖNDERİ servisi, veritabanına kayıt ekler.
10. GÖNDERİ servisi, SİPARİŞ ORKESTRATÖRÜ’ne “GÖNDERİ KAYDEDİLDİ” eventi gönderir.
11. SİPARİŞ ORKESTRATÖRÜ ,SİPARİŞ servisine “SİPARİŞ GÜNCELLE” eventi gönderir.
12. SİPARİŞ servisi kaydı “ BİTİR” olarak günceller.

**BAŞARISIZ SENARYO:**



**SENARYO-1 :STOK servisinde hata var**

1. STOK servisi ,SİPARİŞ ORKESTRATÖRÜ’ne “STOKTA KALMADI” eventini yayınlar
2. SİPARİŞ ORKESTRATÖRÜ ,Stokun güncellenemediği bilgileriyle SİPARİŞ servisine “SİPARİŞİ GERİ ALMA ” eventini yayınlar.
3. SİPARİŞ servisi sipariş kaydını BAŞARISIZ olarak günceller.

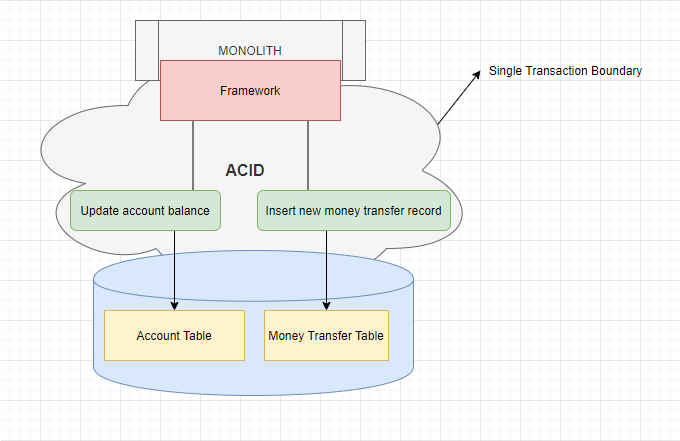
**Senaryo 2:GÖNDERİM servisi hata alır.**

1. GÖNDERİM servisi ,SİPARİŞ ORKESTRATÖRÜ’ne “GÖNDERİ BAŞASIZ” eventi gönderir.
2. SİPARİŞ ORKESTRATÖRÜ , SİPARİŞ servisine “GERİ ALMA” eventi gönderir.
3. SİPARİŞ servisi ,kaydı “BAŞARISIZ” olarak günceller.
4. SİPARİŞ ORKESTRATÖRÜ , STOK servisine “GERİ ALMA” eventi gönderir.
5. STOK servisi ilgili kaydı önceki görüntüsüne alır.

**ORKESTRATÖRÜ modellemenin standart yolu ,her dönüşümün bir komuta veya mesaja karşılık geldiği DURUM MAKİNE sidir. PROCESS FLOW çizmek..**

**TRANSACTION TÜRLERİ:**

**MONOLOTHİC**



**DISTRIBUTED TRANSACTION**

