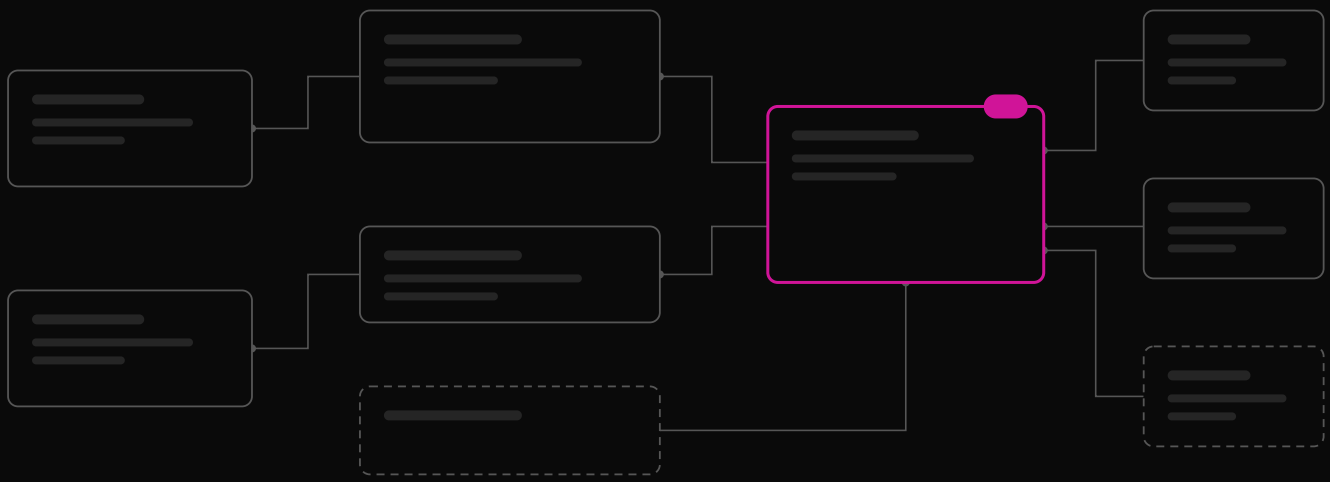
ÖRNEK RAPOR

# Mimari Değerlendirme

## Müşteri A (kurgusal)

B2B sipariş ve sevkiyat yönetim platformu
Eylül 2026



**ÖRNEK** **Örnek. Tüm adlar, rakamlar ve bulgular kurgusaldır.**

Bu belge, bir Future Formative Mimari Değerlendirmesi'nin biçimini ve derinliğini gösterir. Müşteri A gerçek bir şirket değildir.

Future Formative Technology LTD · London, UK

Future Formative Teknoloji A.Ş. · İzmir, Türkiye

BU ÖRNEK HAKKINDA

# Kurgusal verilerle kısaltılmış bir örnek

Bu belge, bir Mimari Değerlendirme'nin sonunda teslim edilen raporu tanıtır. Kapak dışında, genellikle 40 ila 60 sayfalık bir bulgular raporundan kısaltılmış 13 sayfa içerir.

> **Müşteri A kurgusaldır**
>
> Bu örnekteki şirket, sistemi, tüm rakamlar ve tüm bulgular uydurulmuştur. Gerçek hiçbir şirket, ürün ya da kişi anlatılmaz. Senaryo, bu yaşta ve büyüklükteki platformlarda sık görülen sorunları bir araya getirir; veriler gerçek olmasa da akıl yürütme gerçekçidir.

## Mimari Değerlendirme

| | |
|---|---|
| SÜRE | 2 ila 3 hafta, sabit kapsam |
| ERİŞİM | Salt okunur. Sistemlerinizde hiçbir şey değiştirilmez. |
| ÇIKTILAR | Bulgular raporu · risk matrisi · mevcut ve hedef durumun C4 diyagramları · efor aralıkları ve her aşama için bir metrik içeren 12 aylık yol haritası · başlangıç ölçümleri · 90 dakikalık sunum oturumu |
| MÜLKİYET | Tüm çıktılar müşteriye aittir. |

## Gerçek bir raporda ayrıca

- **Tam kanıt eki:** her bulgunun arkasındaki sorgular, log kesitleri, kod referansları ve dashboard çıktıları.
- **Diyagram kaynak dosyaları:** ekibinizin güncel tutabilmesi için tüm C4 diyagramlarının düzenlenebilir kaynakları.
- **Sunum oturumu:** bulguları mühendislik yönetiminizle birlikte ele alıp öncelikleri belirlediğimiz 90 dakikalık oturum.
- **Bulguların tam hali:** bu örnekte her bulgu tek bir bloğa indirilmiştir.

## İçindekiler


Bu örnek hakkında 2
Yönetici özeti 3
Kapsam, sorular ve yöntem 4
Sistem bağlamı 5
Mevcut mimari 6
Başlangıç ölçümleri 7
Bulgular: F1 ila F7 8
Risk matrisi 11
Hedef mimari 12
12 aylık yol haritası 13
Yapılmaması gerekenler, sonraki adımlar, sözlük 14


## Okuma notları

**Efor** mühendis-haftası cinsinden bir aralık olarak verilir; örneğin 3 ila 5 mühendis-haftası.

**Önem derecesi** üç düzeydir: Yüksek, Orta, Düşük ila orta. Sayfa 11 her bulguyu olasılık ve etkiye göre konumlandırır.

**Maliyet** yalnızca göreli değişim olarak yer alır. Hiçbir yerde tutar ya da fiyat bulunmaz.

YÖNETİCİ ÖZETİ

# Sorun monolitin kendisi değil; nasıl yüklendiği ve nasıl yayına alındığı.

Yavaşlamaların, mükerrer sevkiyatların ve artan bulut maliyetinin somut, çözülebilir nedenleri var. Hiçbiri mikroservis gerektirmiyor; 12 ayda ölçülebilir dört aşama hepsini ele alıyor.

**7,9 sn**
ay sonu sipariş p95; 30 günde 2,8 sn

**%22**
değişiklik hata oranı; haftalık manuel sürüm

**+%41**
12 ayda bulut harcaması; sipariş +%12

**%0,3**
siparişlerde mükerrer sevkiyat olayı oranı

## CTO'nun üç sorusuna yanıtlar

SORU 1

### Ay sonu neden tüm platformu yavaşlatıyor?

Ay sonu raporları ve CSV dışa aktarımları birincil veritabanında çalışıyor ve orders tablosunu kilitliyor; bu yüzden sipariş gönderiminin p95 değeri neredeyse üç katına çıkıyor (2,8 sn'den 7,9 sn'ye). Bekleyen istekler thread'leri tuttuğu için yalnızca raporlar değil, tüm sayfalar yavaşlıyor. Bir okuma replikası zaten var ama kullanılmıyor: raporlamayı oraya taşımak en hızlı kazanım (F1).

SORU 2

### Mikroservislere geçmeli miyiz?

Şimdi değil. Nedenler veri erişimi, timeout'suz çağrılar, mesaj işleme ve teslimat süreci; kodu bölmek hiçbirini çözmez. Servisler, timeout'u olmayan sipariş akışına yeni ağ çağrıları, build'i 46 dakika süren ve değişiklik hata oranı %22 olan sürece de daha fazla sürüm ekler. Önce monoliti modüler hale getirin (Orders, Shipments, Billing, Tenancy); servisi ancak bir modül bağımsız ölçeklenme ya da sürüm gerektirdiğinde ayırın.

SORU 3

### Bulut maliyeti neden sipariş hacminden daha hızlı arttı?

Harcama 12 ayda %41 artarken sipariş hacmi %12 arttı. Veritabanı ay boyunca ay sonu zirvesine göre boyutlandırılmış ve üretim dışı dört ortam 7/24 çalışıyor. Raporlama birincil veritabanından ayrıldığında (F1) veritabanı doğru boyutlandırılabilir ve üretim dışı ortamlar zamanlamaya bağlanabilir (F7); böylece harcama sipariş hacminden hızlı artmaz.

## Öncelikli riskler

**F1** Raporlama ve dışa aktarımlar birincil veritabanında
Muhtemel · yüksek etki

**F2** Timeout'u olmayan altı senkron çağrı
Muhtemel · yüksek etki

**F3** Idempotent olmayan mesaj işleyiciler
Kesin · orta etki · ayda yaklaşık 20 saat temizlik

## İlk 90 gün

- Sipariş gönderiminde timeout ve circuit breaker F2
- Idempotency anahtarları, inbox ve outbox tabloları F3
- Sipariş akışında OpenTelemetry ile izleme F6
- Raporlar ve dışa aktarımlar okuma replikasına F1

**3. ayın sonunda kontrol:** ay sonu sipariş p95 değeri 3 sn'nin altında ve sıfır mükerrer sevkiyat.

KAPSAM VE YÖNTEM

# Kapsam, sorular ve yöntem

Üç hafta, salt okunur erişim ve CTO'nun üç sorusu. Bu rapordaki her bulgu bu sorulardan en az birine yanıt verir.

## Sorular

| # | CTO'NUN SORUSU | YANITLAYAN BULGULAR |
|---|---|---|
| 1 | Ay sonu neden tüm platformu yavaşlatıyor? | F1 F2 F5 |
| 2 | Mikroservislere geçmeli miyiz? | F2 F3 F4 F6 |
| 3 | Bulut maliyeti neden sipariş hacminden daha hızlı arttı? | F7 F1 |

## Kaynaklar

| KAYNAK | KULLANDIKLARIMIZ | DÖNEM |
|---|---|---|
| **Kod** | Monolitin, iki .NET 8 servisinin ve React ön yüzün repoları | Mevcut durum |
| **Telemetri** | İstek, uygulama ve veritabanı metrikleri ve logları | 30 gün |
| **Olaylar** | 31 olay (incident) kaydı, 4'ü P1 | 12 ay |
| **Bulut faturası** | Servis ve ortam bazında harcama | 12 ay |
| **Teslimat** | Sürüm kontrolü, CI çalıştırmaları ve sürüm kontrol listesi | 30 gün |
| **Görüşmeler** | Mühendislik, operasyon, ürün ve destek alanlarından 9 rol. İsim kaydedilmedi. | 1. ve 2. hafta |

## Üç hafta

1. HAFTA
### Anlama
Açılış toplantısı ve salt okunur erişim. Mimari tanıtım, kod okuma, ilk görüşmeler. Başlangıç ölçümlerinin tanımlanması.

2. HAFTA
### Ölçme
Telemetri, olay ve maliyet analizi. Kalan görüşmeler. Her hipotezin veriyle sınanması.

3. HAFTA
### Öneri
Bulgular, risk matrisi, hedef mimari ve yol haritası. Raporun teslimi, ardından 90 dakikalık sunum oturumu.

## Kapsam ve sınırlar

### Kapsamda
Sipariş ve sevkiyat platformu: ASP.NET monolit, kargo entegrasyonu ve bildirim servisleri, React ön yüz, PostgreSQL, RabbitMQ, Redis, AWS altyapısı (EC2 ve RDS, tek bölge) ile build ve sürüm süreci.

### Kapsam dışında
Güvenlik testleri ve kullanıcı arayüzü tasarımı.

### Sınırlar
- Salt okunur: yük testi yapılmadı, hiçbir şey değiştirilmedi. Ay sonu davranışı yeniden üretilmedi, telemetriden okundu.
- 30 günlük telemetri penceresi bir ay sonunu kapsıyor.
- Maliyet, servis ve ortam bazında göreli değişim olarak analiz edildi. Bu örnekte tutar gösterilmez.
- Efor aralıkları kod tabanını bilen mühendisleri varsayar; teklif değil, planlama aralığıdır.
- Görüşmelerdeki ifadeler ipucu olarak ele alındı ve veriyle doğrulandı.

MEVCUT DURUM · C4 SEVİYE 1

# Sistem bağlamı

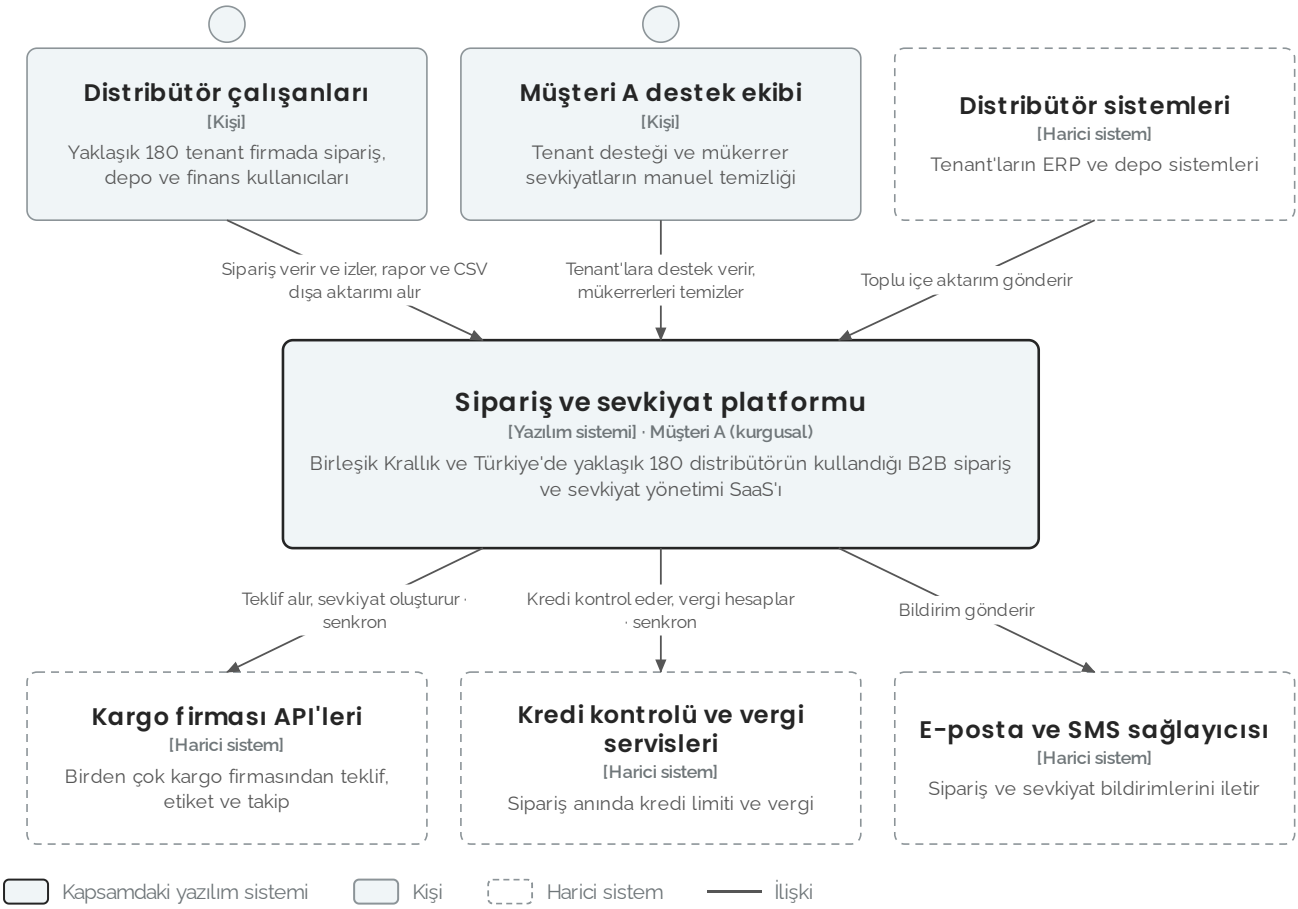
Platformu kimlerin kullandığı ve platformun hangi sistemlere bağımlı olduğu. Bu seviyede platform tek bir sistemdir; içini bir sonraki sayfa açar.

**Yaklaşık 180**
tenant firma, Birleşik Krallık ve Türkiye'de

**38 mühendis**
5 ekipte

**7 yıllık**
ASP.NET monolit ve iki .NET 8 servisi

**1 bölge**
AWS üzerinde: EC2 ve RDS



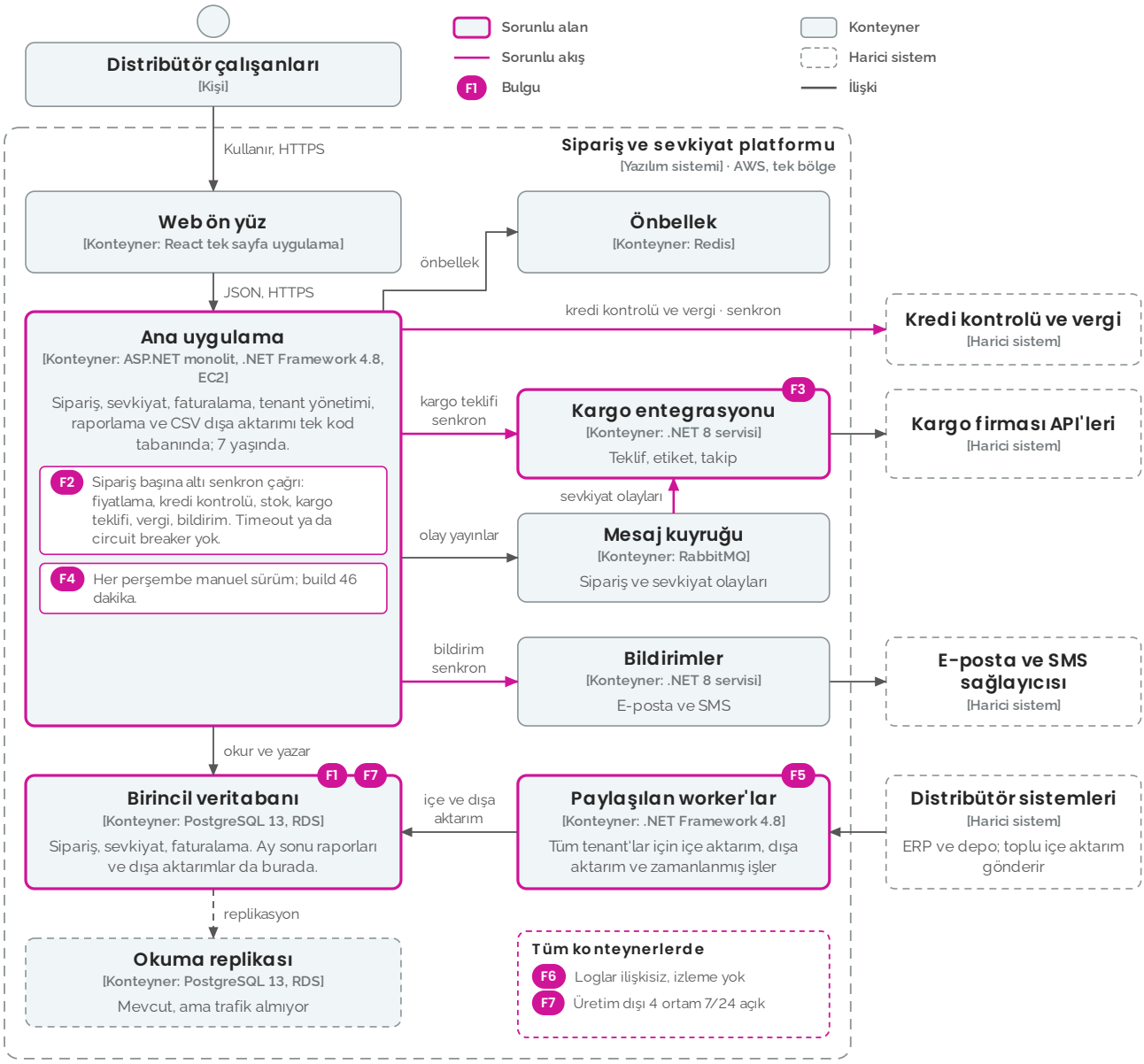

## Bağlam neyi gösteriyor

- Her sipariş gönderimi harici sistemleri bekler: kargo firması API'leri, kredi kontrolü ve vergi servisleri, bildirim servisi üzerinden de e-posta ve SMS sağlayıcısı. Timeout olmadığından onların gecikmesi Müşteri A'nın gecikmesi olur F2
- Ay sonunda distribütör çalışanları rapor ve CSV dışa aktarımı çalıştırırken siparişler gelmeye devam eder. İkisi de aynı veritabanına yüklenir F1
- Distribütör sistemleri toplu içe aktarım gönderir; bunlar tüm tenant'ların paylaştığı worker'larda işlenir F5
- Müşteri A destek ekibi mükerrer sevkiyatları temizlemek için ayda yaklaşık 20 saat harcar F3
- Tüm tenant'lar tek bir kurulumu paylaşır; sürüm haftada bir, perşembe günleri manuel kontrol listesiyle yapılır F4

MEVCUT DURUM · C4 SEVİYE 2

# Mevcut mimari: konteynerler

Platformun ayrı dağıtılan parçaları ve birbirleriyle bağlantıları. Sorunlu alanlar renkli çerçeveyle gösterilir ve ilgili bulguyla etiketlenir.




**F1** Ay sonu raporları ve CSV dışa aktarımları birincil veritabanında; replika boşta.

**F2** Sipariş başına altı senkron çağrı; timeout ya da circuit breaker yok.

**F3** Handler'lar idempotent değil; yeniden teslim mükerrer sevkiyat üretiyor.

**F4** Perşembe günleri haftada bir manuel sürüm; build 46 dakika.

**F5** Büyük tenant'ların toplu içe aktarımları paylaşılan worker'ları doyuruyor.

**F6** Loglar ilişkilendirilmiyor; dağıtık izleme yok.

**F7** Ay boyunca ay sonuna göre boyutlandırılmış; üretim dışı dört ortam 7/24 açık.

BAŞLANGIÇ ÖLÇÜMLERİ

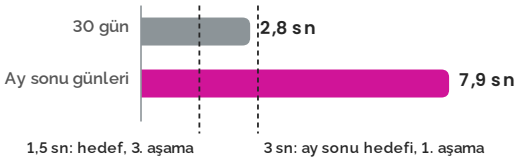
# Platform bugün nerede duruyor

Yol haritasının başlangıç çizgisi: her aşama bu değerlere karşı yapılan bir kontrolle biter.

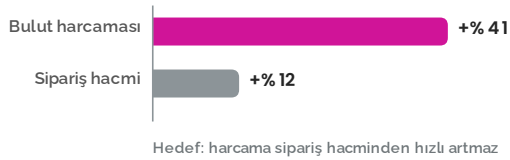
| ALAN | METRİK | BAŞLANGIÇ | HEDEF VE AŞAMA |
|---|---|---|---|
| GECİKME | Sipariş gönderimi gecikmesi, p95 | **2,8 sn** | 1,5 sn'nin altı · 7. ila 9. ay |
| | Sipariş gönderimi gecikmesi, p99 | **9,4 sn** | İzleniyor |
| | Ay sonu günlerinde sipariş gönderimi, p95 | **7,9 sn** | 3 sn'nin altı · 1. ila 3. ay |
| | Arama gecikmesi, p95 | **1,2 sn** | İzleniyor |
| TESLİMAT | Dağıtım sıklığı | **Haftada 1** | İzleniyor |
| | Değişiklik teslim süresi, medyan | **9 gün** | 3 günün altı · 4. ila 6. ay |
| | Değişiklik hata oranı | **%22** | %10'un altı · 4. ila 6. ay |
| | Hizmeti geri getirme süresi, medyan | **3 sa 50 dk** | İzleniyor |
| | CI build süresi | **46 dk** | İzleniyor |
| | Kararsız testler yüzünden başarısız test koşuları | **%7** | İzleniyor |
| GÜVENİLİRLİK | 12 aydaki olaylar | **31, bunların 4'ü P1** | İzleniyor |
| | Mükerrer sevkiyat olayı üreten siparişler | **%0,3** | Sıfır · 1. ila 3. ay |
| | Mükerrerlerin manuel temizliği | **Ayda yaklaşık 20 sa** | Gerek kalmaz · 1. ila 3. ay |
| MALİYET | Bulut harcaması ve sipariş hacmi, 12 ay | **+%41 ve +%12** | Harcama sipariş hacminden hızlı artmaz · 10. ila 12. ay |

Dönem: aksi belirtilmedikçe 30 gün. İzleniyor: kendine ait bir hedef olmadan her aşama kontrolünde raporlanır.

**Sipariş gönderimi p95, saniye**




**12 aydaki artış**




## Nasıl ölçüldü

### Gecikme

Load balancer ve uygulama istek logları, tüm tenant'lar, 30 gün. Ay sonu günleri: ayın son iki iş günü ve ertesi ayın ilk iş günü.

### Teslimat

Sürüm kontrolü ve sürüm kayıtları; teslim süresi commit'ten üretime kadar ölçülür. Değişiklik hata oranı: geri alma ya da hotfix gerektiren veya olaya yol açan sürümlerin payı. Geri getirme süresi: olayın başlangıcından hizmetin geri gelmesine kadar. Build süresi ve kararsız koşular CI loglarından.

### Güvenilirlik

12 aylık olay kayıtları. Mükerrerler: mesaj loglarında aynı sipariş için tekrarlanan sevkiyat olayları. Temizlik saatleri destek kayıtlarından.

### Maliyet

Servis ve ortam bazında gruplanmış bulut faturası, aynı 12 aydaki sipariş sayılarıyla karşılaştırıldı. Yalnızca göreli değişim; tutar yok.

BULGULAR

# Bulgular

Risk sırasına göre yedi bulgu. Her blok aynı altı bölümden oluşur.

Önem derecesi: **Yüksek** **Orta** **Düşük ila orta**

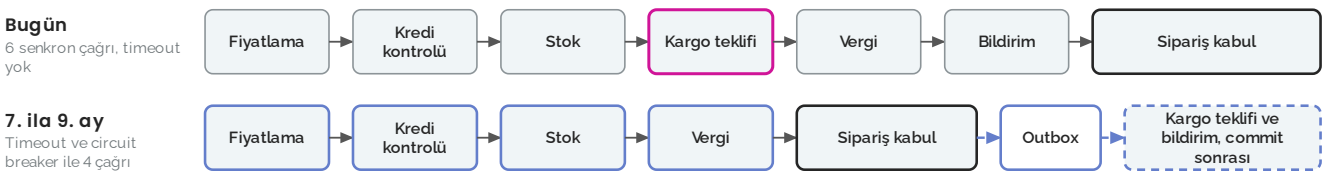
## F1 Raporlama ve dışa aktarımlar birincil veritabanında çalışıyor — Yüksek

| | |
|---|---|
| KANIT | Ay sonu raporları ve CSV dışa aktarımları birincil PostgreSQL 13 örneğinde çalışıyor ve okuma sırasında orders tablosunu kilitliyor. Ay sonu günlerinde sipariş gönderiminin p95 değeri 2,8 sn'den 7,9 sn'ye çıkıyor. Okuma replikası hiç trafik almıyor. |
| ETKİ | Sipariş gönderimi gecikmesi, distribütörlerin platforma en çok dayandığı günlerde tüm tenant'lar için neredeyse üç katına çıkıyor. Bekleyen istekler thread'leri tuttuğundan yavaşlama tüm sayfalara yayılıyor. |
| KÖK NEDEN | Raporlama monolitin içinde, işlem tablolarının üzerine kurulmuş. Dışa aktarma işi tutarlı bir anlık görüntü için tabloyu kilitliyor; oysa repeatable read bir transaction aynı tutarlılığı yazmaları engellemeden sağlar. Analitik yükü sipariş trafiğinden ayıran bir yapı yok. |
| ÖNERİ | **1. ila 3. ay:** ay sonu raporlarını ve CSV dışa aktarımlarını mevcut okuma replikasına yönlendirin, tablo kilidini kaldırın. **7. ila 9. ay:** change data capture (CDC) ile beslenen bir raporlama deposu; birincil veritabanına raporlama yükü gelmez. |
| EFOR ARALIĞI | **3 ila 5 mühendis-haftası** |
| BAĞIMLILIKLAR | Başlamak için yok. F7'nin önünü açar: ay sonu zirvesi ortadan kalkınca veritabanı doğru boyutlandırılabilir. |

## F2 Sipariş gönderiminde altı senkron çağrı — Yüksek

| | |
|---|---|
| KANIT | Sipariş gönderimi fiyatlama, kredi kontrolü, stok, kargo teklifi, vergi ve bildirim çağrılarını sırayla yapıyor; hiçbirinde timeout ya da circuit breaker yok. p99 değeri 9,4 sn; örneklediğimiz yavaş isteklerde sürenin çoğu kargo teklifini beklerken geçiyor. |
| ETKİ | Yavaş tek bir kargo API'si, havuz tükenene kadar istek thread'lerini bloke ediyor ve tüm platform yavaşlıyor. Her bağımlılığın gecikmesi müşterinin beklemesine ekleniyor. |
| KÖK NEDEN | Sipariş akışı yedi yıl boyunca tek bir isteğe yeni çağrılar eklenerek büyümüş. Kargo teklifi ve bildirimin sipariş kabul edilmeden önce tamamlanması gerekmiyor, ama aynı senkron zincirde duruyorlar. |
| ÖNERİ | **1. ila 3. ay:** altı çağrının hepsine timeout ve circuit breaker, her biri için tanımlı bir fallback. **7. ila 9. ay:** kargo teklifi ve bildirim, sipariş commit edildikten sonra transactional outbox üzerinden asenkron çalışır. |
| EFOR ARALIĞI | **4 ila 6 mühendis-haftası** |
| BAĞIMLILIKLAR | İzleme (F6) önce hangi çağrının düzeltileceğini gösterir. 2. kısım, F3 ile gelen outbox tablolarını kullanır. |

ŞEKİL · SİPARİŞ GÖNDERİM AKIŞI: BUGÜN VE 7. İLA 9. AYDAN SONRA



BULGULAR, DEVAM

## F3 Mesaj işleyiciler idempotent değil

Yüksek (tekrarlayacağı kesin)

| | |
|---|---|
| KANIT | Siparişlerin %0,3'ü mükerrer bir sevkiyat olayı üretiyor. Örneklediğimiz mükerrerlerin hepsi, consumer yeniden başladıktan ya da bir onay (ack) kaybolduktan sonraki RabbitMQ yeniden teslimini izliyor. Temizlik ayda yaklaşık 20 saat manuel iş. |
| ETKİ | Mükerrer sevkiyat olayları kargo firmalarına ve müşterilere ulaşıyor ve elle geri alınması gerekiyor. Sorunun tekrarlayacağı kesin: RabbitMQ en az bir kez teslim eder; yeniden teslim bir arıza değil, normal davranıştır. |
| KÖK NEDEN | Handler'lar her mesajın tam olarak bir kez geleceğini varsayıyor. Hangi mesajların işlendiğini kaydeden bir mekanizma olmadığından yeniden teslim edilen mesaj bir kez daha işleniyor. |
| ÖNERİ | Her mesajda bir idempotency anahtarı, her consumer'da işlenen anahtarları kaydeden bir inbox tablosu ve durum değişikliğiyle mesajını tek transaction'da yazan bir outbox tablosu. |
| EFOR ARALIĞI | **2 ila 3 mühendis-haftası** |
| BAĞIMLILIKLAR | Yok. İlk bu yapılmalı: F2'nin 2. kısmı aynı inbox ve outbox tablolarının üzerine kurulur. |

## F4 Sürüm süreci

Orta

| | |
|---|---|
| KANIT | Perşembe günleri kontrol listesiyle haftada bir manuel sürüm. CI build'i 46 dakika sürüyor; test koşularının %7'si tekrar denemede geçen kararsız (flaky) testler yüzünden başarısız oluyor. Teslim süresi medyanı 9 gün, değişiklik hata oranı %22, geri getirme süresi medyanı 3 sa 50 dk. |
| ETKİ | Değişiklikler bir haftaya kadar bekliyor ve büyük paketlerle çıkıyor; her sürüm daha riskli ve bir hatayı tek bir değişikliğe bağlamak zor. Kararsız testler ekiplere kırmızı build'leri görmezden gelmeyi öğretiyor. |
| KÖK NEDEN | Tüm monolit için tek bir test paketi sırayla çalışıyor, uzun ömürlü branch'ler sürümden hemen önce birleştiriliyor ve geri alma, kontrol listesinde manuel bir adım. |
| ÖNERİ | Test paketini bölün ve paralel çalıştırın, kararsız testleri karantinaya alıp takip edilen bir listeden düzeltin, feature flag'lerle trunk-based development'a geçin ve geri almayı otomatikleştirin. |
| EFOR ARALIĞI | **Ekipler genelinde 4 ila 8 mühendis-haftası** |
| BAĞIMLILIKLAR | Her türlü servis ayrıştırmasından önce gerekli: daha fazla dağıtım birimi, daha hızlı ve daha güvenli bir sürüm süreci ister. |

## F5 Toplu içe aktarımlarda tenant izolasyonu yok

Orta

| | |
|---|---|
| KANIT | Büyük tenant'ların toplu içe aktarımları, diğer tüm tenant'ların işleriyle aynı worker havuzunda çalışıyor. Büyük içe aktarımlarda worker'lar doyuyor ve diğer işler kuyrukta bekliyor. |
| ETKİ | Bir tenant'ın içe aktarımı diğerlerinin içe aktarım, dışa aktarım ve zamanlanmış işlerini geciktiriyor: gürültülü komşu (noisy neighbour) etkisi. Risk, her yeni büyük tenant ile büyüyor. |
| KÖK NEDEN | Tenant başına sınırı olmayan, ilk gelen ilk işlenir mantığıyla çalışan tek bir paylaşılan kuyruk ve worker havuzu. |
| ÖNERİ | Tenant başına kuyruklar ve hız sınırları (rate limit), ayrıca içe aktarımların diğer işleri aç bırakamaması için ayrı içe aktarım worker'ları. |
| EFOR ARALIĞI | **3 ila 4 mühendis-haftası** |
| BAĞIMLILIKLAR | Yok. İzleme (F6) etkiyi tenant bazında görünür kılar. |

BULGULAR, DEVAM

## F6 Gözlemlenebilirlik eksikleri — Orta

| | |
|---|---|
| KANIT | Loglar konteyner bazında tutuluyor ve korelasyon kimliği taşımıyor; dağıtık izleme (distributed tracing) yok. Olay kayıtlarında hizmeti geri getirme süresinin çoğu nedeni bulmaya harcanmış. |
| ETKİ | Olayların teşhisi saatler sürüyor (hizmeti geri getirme süresi medyanı 3 sa 50 dk) ve bir müşterinin gerçekte ne yaşadığını kimse göremiyor. |
| KÖK NEDEN | Loglama bileşen bileşen büyümüş. Monolit ve iki .NET 8 servisi hiçbir zaman tek bir sistem olarak enstrümante edilmemiş ve hizmet seviyesi hedefleri (SLO) tanımlı değil. |
| ÖNERİ | Önce sipariş akışında ön yüz, monolit, servisler, veritabanı ve mesaj kuyruğu boyunca OpenTelemetry ile izleme. Ardından üç müşteri yolculuğu için SLO: sipariş gönderme, arama ve sevkiyat takibi. |
| EFOR ARALIĞI | **2 ila 4 mühendis-haftası** |
| BAĞIMLILIKLAR | Yok. Erken başlayın: izleme, diğer her değişikliğin etkisini ölçer. |

## F7 Gereğinden büyük ve atıl altyapı — Düşük ila orta

| | |
|---|---|
| KANIT | Birincil veritabanı ay boyunca ay sonu zirvesine göre boyutlandırılmış. Üretim dışı dört ortam haftanın 7 günü, günün 24 saati çalışıyor. Bulut harcaması 12 ayda %41, sipariş hacmi %12 arttı. |
| ETKİ | Harcama işten daha hızlı büyüyor. Etki bir kesinti değil, sürekli bir eğilim; önem derecesinin daha düşük olmasının nedeni bu. |
| KÖK NEDEN | Kapasite ay sonunu atlatmak için artırılmış (bkz. F1) ve bir daha düşürülmemiş. Üretim dışı ortamların çalışma takvimi yok. |
| ÖNERİ | F1'den sonra veritabanını normal yüke göre doğru boyutlandırın ve üretim dışı ortamları mesai saatlerine göre zamanlayın. Beklenen etki: harcama artışı sipariş hacmiyle yeniden uyumlu hale gelir. |
| EFOR ARALIĞI | **1 ila 2 mühendis-haftası** |
| BAĞIMLILIKLAR | F1 gerekli. Raporlama birincil veritabanından ayrılmadan yapılacak bir küçültme ay sonu yavaşlamasını geri getirir. |

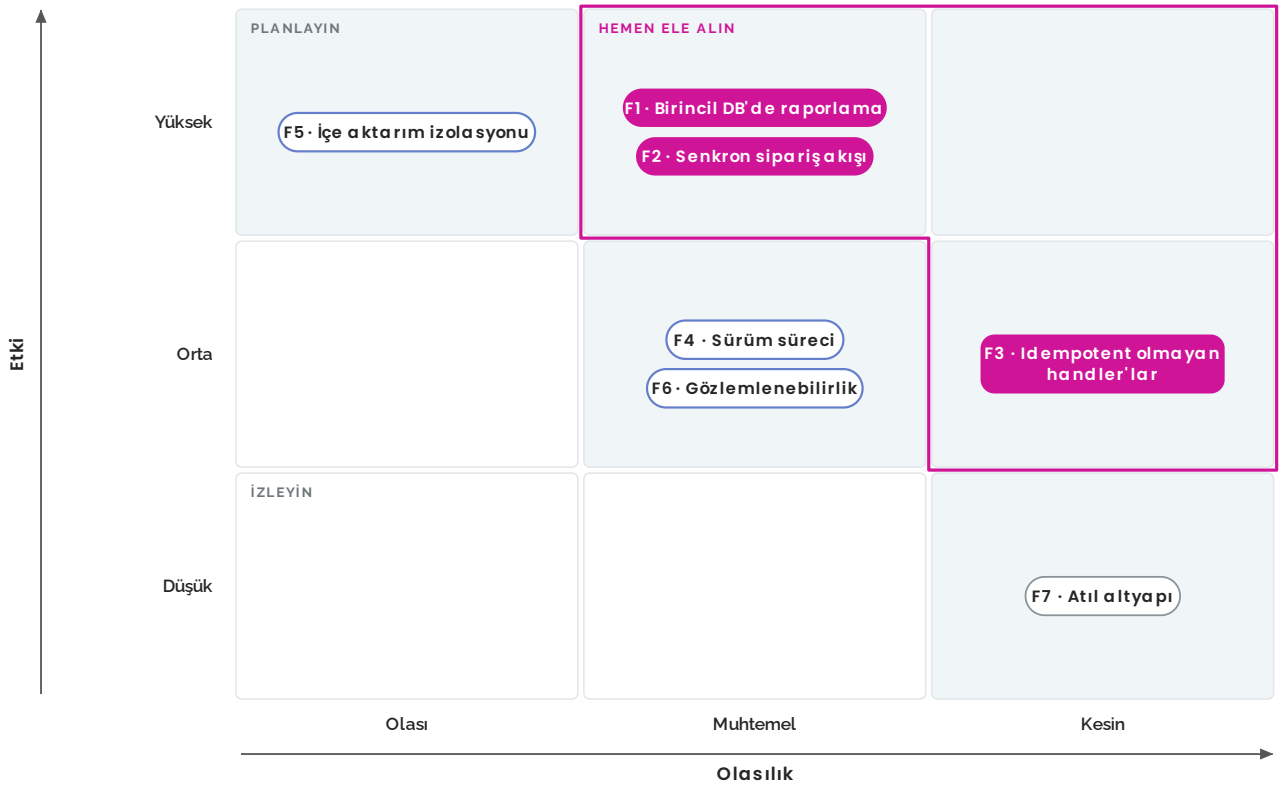
## Bulgulara genel bakış

| BULGU | ÖNEM DERECESİ | EFOR ARALIĞI, MÜHENDİS-HAFTASI | YOL HARİTASI AŞAMASI |
|---|---|---|---|
| F1 Raporlama ve dışa aktarımlar birincil veritabanında | Yüksek | 3 ila 5 | 1 ve 3 |
| F2 Sipariş gönderiminde altı senkron çağrı | Yüksek | 4 ila 6 | 1 ve 3 |
| F3 Mesaj işleyiciler idempotent değil | Yüksek | 2 ila 3 | 1 |
| F4 Sürüm süreci | Orta | 4 ila 8, ekipler genelinde | 2 |
| F5 Toplu içe aktarımlarda tenant izolasyonu yok | Orta | 3 ila 4 | 2 |
| F6 Gözlemlenebilirlik eksikleri | Orta | 2 ila 4 | 1 ve 4 |
| F7 Gereğinden büyük ve atıl altyapı | Düşük ila orta | 1 ila 2 | 4 |

RİSK MATRİSİ

# Riskler nerede duruyor

Her bulgu, önümüzdeki 12 ayda zarar verme olasılığına ve vereceği zararın büyüklüğüne göre yerleştirildi. Çerçeveli hücreler önce ele alınır.




## Nasıl okunur

| | |
|---|---|
| OLASILIK | **Olası:** belirli koşulların bir araya gelmesini gerektirir. **Muhtemel:** normal yük altında 12 ay içinde beklenir. **Kesin:** bugün yaşanıyor ve tekrarlayacak. |
| ETKİ | **Düşük:** maliyet ya da verimlilik. **Orta:** bozulan hizmet ya da manuel iş. **Yüksek:** platform genelinde yavaşlama ya da başarısız siparişler. |
| ÖNEM | Etiket stiliyle gösterilir: Yüksek için dolu, Orta için mavi çerçeve, Düşük ila orta için gri çerçeve. |

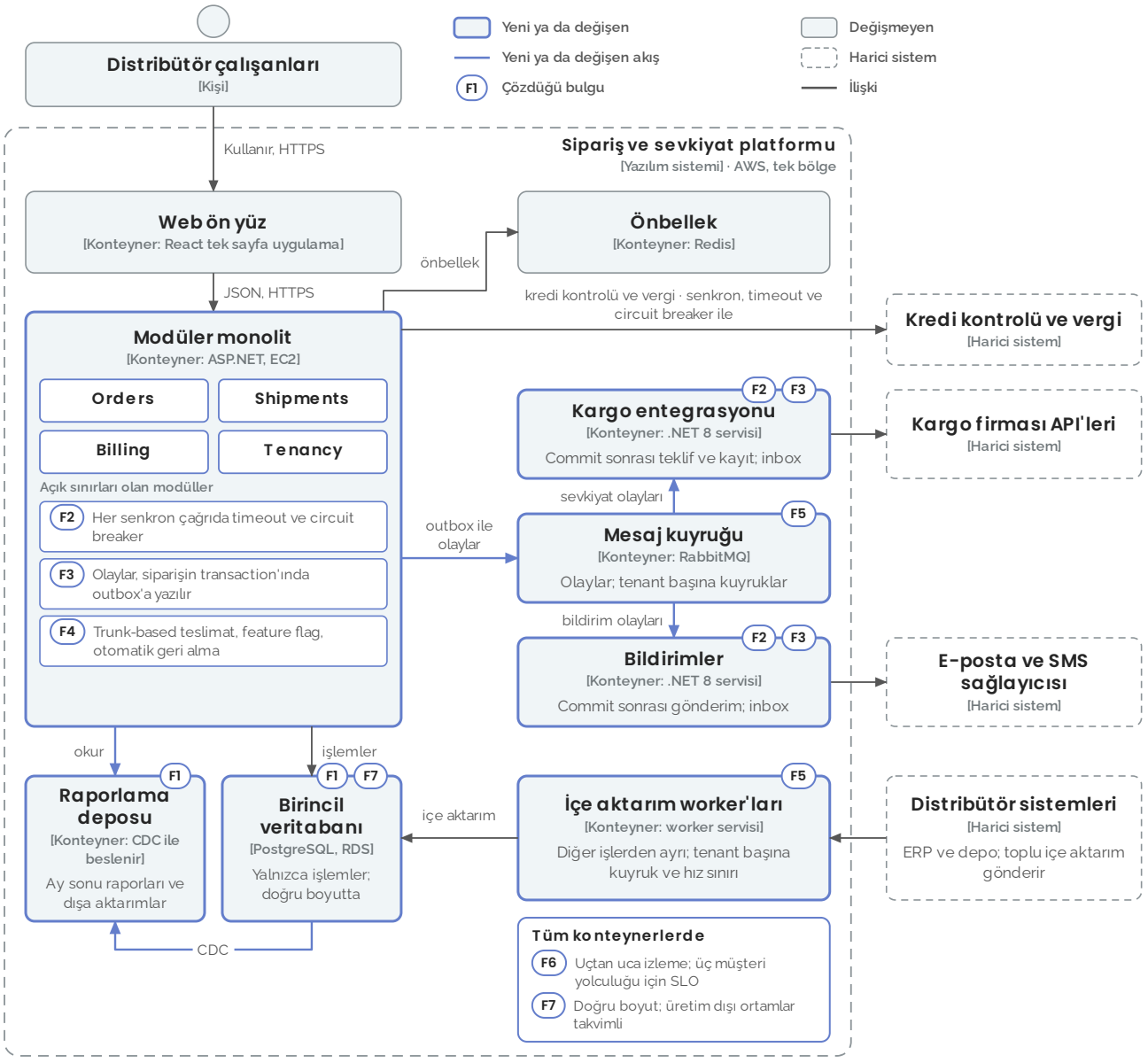
## Bize ne söylüyor

- **F1, F2 ve F3** hemen ele alınacak bölgede. Üçü de 1. aşamada, 1. ila 3. ayda başlar.
- **F6** orta düzeyde, ama izleme 1. aşamada başlar; çünkü diğer her değişikliği o ölçer.
- **F4** muhtemel ve orta: ileride yapılacak her servis ayrıştırmasının ön koşulu.
- **F5** bugün yalnızca olası, ama etkisi yüksek ve her büyük tenant ile büyüyor.
- **F7** kesin ama etkisi düşük: müşteri değil para kaybettirir ve F1'i bekler.

HEDEF DURUM · C4 SEVİYE 2

# 12. aydaki hedef mimari

Monolit modüler hale gelir, raporlama birincil veritabanından ayrılır ve sipariş akışı asenkron olur. Bir modül bağımsız ölçeklenme ya da bağımsız sürüm gerektirmedikçe hiçbir servis ayrılmaz.




## Hedefin arkasındaki ilkeler

1. **Önce modüller, sonra servisler.** Modül ancak bağımsız ölçeklenme ya da sürüm için servis olur.
2. **Birincil veritabanı işlemler içindir.** Raporlar önce replikadan, sonra CDC ile beslenen depodan okur.
3. **Önce sipariş kabul edilir.** Kargo teklifi ve bildirim commit'ten sonra outbox ile gelir.
4. **Her mesaj iki kez gelebilir.** Handler'lar idempotency anahtarı ve inbox tablosu kullanır.
5. **Hiçbir tenant diğerlerini aç bırakmaz.** İçe aktarımlar kendi worker'larında, tenant sınırlarıyla çalışır.
6. **Müşteri deneyimi ölçülür.** Her sipariş uçtan uca izlenir; üç yolculuğun SLO'su vardır.
7. **Kapasite talebi izler.** Normal yüke göre boyut; üretim dışı ortamlar takvimle çalışır.

12 AYLIK YOL HARİTASI

# Her biri bir ölçümle biten dört aşama

Her aşama birkaç iş kaleminden ve sayfa 7'ye göre ölçülen tek bir metrikten oluşur.

| 1. ay | 2. ay | 3. ay | 4. ay | 5. ay | 6. ay | 7. ay | 8. ay | 9. ay | 10. ay | 11. ay | 12. ay |
|---|---|---|---|---|---|---|---|---|---|---|---|

1. AŞAMA · 1. İLA 3. AY

## Stabilize et

- Sipariş akışında timeout ve circuit breaker F2
- Idempotency anahtarları, inbox ve outbox tabloları F3
- Sipariş akışında OpenTelemetry ile izleme F6
- Raporlar ve dışa aktarımlar okuma replikasında F1

AŞAMA METRİĞİ

**Ay sonu sipariş p95 değeri 3 sn'nin altında; sıfır mükerrer sevkiyat.**

2. AŞAMA · 4. İLA 6. AY

## Güvenli teslimat

- Test paketini böl ve paralelleştir; kararsız testleri karantinaya al F4
- Feature flag'lerle trunk-based development; otomatik geri alma F4
- Tenant başına kuyruk ve hız sınırı; ayrı içe aktarım worker'ları F5

AŞAMA METRİĞİ

**Değişiklik hata oranı %10'un altında; teslim süresi 3 günün altında.**

3. AŞAMA · 7. İLA 9. AY

## Sınırları çiz

- Modül sınırları: Orders, Shipments, Billing, Tenancy
- Outbox akışı: kargo teklifi ve bildirim asenkron olur F2
- Change data capture ile beslenen raporlama deposu F1

AŞAMA METRİĞİ

**Sipariş p95 değeri 1,5 sn'nin altında; birincil veritabanında raporlama yükü yok.**

4. AŞAMA · 10. İLA 12. AY

## Maliyet ve ölçek

- Doğru boyutlandırma ve üretim dışı ortam takvimleri F7
- SLO gözden geçirmesi F6
- Kapasite modeli
- İlk servis ayrıştırmasına ilişkin karar

AŞAMA METRİĞİ

**Bulut harcaması sipariş hacminden hızlı artmıyor.**

## Aşamalar boyunca bulgular

| BULGU | 1. AŞAMA | 2. AŞAMA | 3. AŞAMA | 4. AŞAMA |
|---|---|---|---|---|
| F1 | Okuma replikası | | CDC raporlama deposu | |
| F2 | Timeout, breaker | | Outbox akışı | |
| F3 | Anahtar, inbox, outbox | | | |
| F4 | | Test, trunk, geri alma | | |
| F5 | | Tenant kuyruğu, worker | | |
| F6 | İzleme | | | SLO gözden geçirme |
| F7 | | | | Boyutlandırma, takvim |
| Mimari | | | Modül sınırları | Kapasite, ayrıştırma |

## Sırayı belirleyen bağımlılıklar

- **Önce F6:** izleme sonraki her değişikliği ölçer.
- **F7'den önce F1:** küçültme ancak raporlama birincil veritabanından ayrıldıktan sonra güvenlidir.
- **F2'nin 2. kısmından önce F3:** outbox akışı inbox ve outbox tablolarını yeniden kullanır.
- **Her ayrıştırmadan önce F4:** daha fazla dağıtım birimi daha hızlı ve güvenli bir sürüm süreci ister.

### 12. aydaki karar

İlk servisi yalnızca bir modül bağımsız ölçeklenme ya da bağımsız sürüm gerektiriyorsa ayırın. Kanıtı kapasite modeli ve SLO gözden geçirmesi sağlar; ikisi de bu ihtiyacı göstermiyorsa modüler monolit kalır.

ÖNERİLER

# Yapmamanızı önerdiklerimiz

### Baştan yazmak

Özellik geliştirmeyi uzun süre dondurur ve mevcut kodun zaten çözmüş olduğu sorunları geri getirir. Bu rapordaki her bulgu yerinde düzeltilebilir.

### Şimdi mikroservislere geçmek

Timeout, idempotency ve teslimat süreci düzelmeden yeni ağ çağrıları ve sürümler ekler. 10. ila 12. ayda veriyle yeniden değerlendirin.

### RabbitMQ'yu değiştirmek

Mükerrerlerin nedeni mesaj kuyruğu değil, idempotent olmayan handler'lar. Her mesaj kuyruğu bir mesajı yeniden teslim edebilir.

### Teslimat düzelmeden Kubernetes

Yazılımın nasıl çalıştığını değiştirir, ne kadar güvenli yayına alındığını değil. 46 dakikalık build ve %22 değişiklik hata oranıyla riski azaltmaz, artırır.

## Sonraki adımlar

1. **Sunum oturumu (90 dakika):** bulguları mühendislik yönetimiyle birlikte ele alın ve öncelikleri netleştirin.
2. **Sorumlular:** 1. aşamadaki her iş kalemi için bir sorumlu belirleyin ve başlangıç ölçümleri dashboard'unu kurun.
3. **1. aşama:** timeout ve circuit breaker, idempotency, izleme, raporların okuma replikasına taşınması.
4. **Yeniden ölçün:** 3. ayın sonunda, sayfa 7'deki başlangıç ölçümlerine göre.

Yol haritası, Müşteri A ekiplerinin kendi başlarına yürütebileceği şekilde yazılmıştır. Diyagram kaynakları dahil tüm çıktılar müşteriye aittir.

## Sözlük

**p95 ve p99**
İsteklerin %95'inin (ya da %99'unun) altında kaldığı değer. 2,8 sn'lik bir p95, her 20 istekten birinin 2,8 sn'den uzun sürdüğü anlamına gelir.

**Değişiklik hata oranı**
Üretimde hataya yol açan ve geri alma, hotfix ya da olay müdahalesi gerektiren sürümlerin payı.

**Transactional outbox**
İş değişikliğiyle aynı veritabanı transaction'ında yazılan bir tablo. Bir relay bu satırları mesaj olarak yayınlar; böylece hiçbir mesaj kaybolmaz ve gerçekleşmemiş bir değişiklik için mesaj gönderilmez.

**CDC (change data capture)**
Satır değişikliklerinin veritabanı logundan başka bir depoya, burada raporlama deposuna, birincil veritabanını sorgulamadan aktarılması.

**SLO (hizmet seviyesi hedefi)**
Bir müşteri yolculuğu için ölçülebilir hedef; örneğin belirlenen süre içinde gönderilen siparişlerin payı.

**Circuit breaker**
Hata veren bir bağımlılığı bir süre çağırmayı durduran koruma; böylece yavaş bir sistem diğerlerini bekletemez.

**Idempotent**
İki kez işlenmesi güvenli: tekrar gelen bir mesajın ek etkisi olmaz.

**C4 modeli**
Yazılım mimarisini dört yakınlaştırma seviyesinde çizme yöntemi: bağlam, konteynerler, bileşenler ve kod.