

# Jamstack & Edge Rendering Uygunluk & Karar Checklist Şablonu — Yazılım / Mimari (v1.0)

**Asset Amaç:** Bu asset, modern web projelerinde Jamstack veya Edge Rendering mimari geçiş kararını geçici bir teknoloji trendi ("hype") olarak görmek yerine tamamen ölçülebilir teknik kriterlere (trafik coğrafyası, içerik güncelleme SLA süreleri, dinamik fonksiyon yüzeyi ve geliştirici ekip olgunluğu) dayandırarak vermenizi sağlar. Proje kapsamındaki sayfa tiplerini statik, yarı-dinamik ve tam dinamik olarak sınıflandırarak hibrit mimari kurgusunu netleştirir; önbellek (cache) yönetimi ve teknik SEO QA gereksinimlerini canlıya alım öncesi operasyonel bir kontrol listesine çevirir. Doküman, otel zincirleri (dinamik fiyat/müsaitlik) ve B2B kurumsal sistemleri (yetkilendirmeli portallar) için özel karar notları barındırır.

**Kim Kullanır?:** Teknik Liderler (Tech Lead), Ajans Proje Yöneticileri (PM), SEO ve Performans Uzmanları, Ürün Sahipleri (Product Owner) ve DevOps Mühendisleri.

## Nasıl Kullanılır?

1. Web sitenizdeki tüm sayfa şablonlarını "statik, yarı-dinamik ve tam dinamik" olarak sınıflandırarak projenin Jamstack/Edge uygunluk skorunu belirleyin.
2. Sayfa tipi bazında önbellek geçersiz kılma (cache invalidation) ve edge işleme (network-first, stale-while-revalidate vb.) stratejilerini dökümandaki matrise işleyin.
3. Paylaşılan 14 günlük hızlandırılmış sprint planını takip ederek bir Kavram Kanıtlama (POC) veya hibrit mimari geçişi uygulayın; süreci Core Web Vitals (CWV) ve SEO QA teknik kapılarından (gate) geçirerek doğrulayın.

## TEMPLATE — Jamstack & Edge Rendering Uygunluk & Karar Checklist Şablonu

### A) Boş Şablon Alanları (Mimari & Kontrol Matrisi)

#### 1) Ölçüm & Önceliklendirme Checklist'i

*Projenin altyapısal dönüşüm kararı verilmeden önce teknik ve operasyonel durumun netleştirilmesi:*

- **Trafik Coğrafyası Dağılımı:** Sitenin kullanıcı trafiğinin coğrafi dağılımı net olarak haritalandırıldı mı?  Sadece TR /  Global Dağılım
  - **İçerik Güncelleme SLA Tanımı:** İçerik yönetim sisteminde (CMS) yapılan bir değişikliğin ön yüze yansımaya tolerans süresi nedir?  15 Dakika /  1 Saat /  1 Gün
  - **Dinamik Fonksiyon Yüzeyi:** Sitede istemci (client) veya sunucu tarafında gerçek zamanlı işlenmesi zorunlu alanlar nelerdir?  Otel Rezervasyon Motoru /  B2B Müşteri Portalı /  Ödeme Sistemi Entegrasyonu
  - **Cache Invalidation & Rebuild Yönetimi:** İçerik güncellendiğinde CDN önbelleğinin geçersiz kılınması (invalidation) ve sitenin yeniden derlenme (rebuild) süreçleri teknik olarak tanımlandı mı?
- 
- **SEO QA Gate Yapılandırması:** Arama motoru botlarının sayfaları kusursuz tarayabilmesi için teknik SEO kontrol mekanizması kuruldu mu?  robots.txt /

Dinamik XML Sitemap /  Canonical Etiketleri /  Schema İşaretlemeleri /  CWV Hedefleri

- Sınır ve API Güvenliği:** Edge network katmanında ve mikro servis API bağlantılarında güvenlik gereksinimleri planlandı mı?
- Geliştirici Ekip Teknolojik Olgunluğu:** Teknik operasyonu yürütecek ekipte gerekli altyapı araçları aktif mi?  CI/CD Pipeline /  İzole Staging Ortamı /  Gerçek Zamanlı Monitoring (İzleme)

## 2) Mimari Sorun giderme Havuzu (Problem $\rightarrow$ Kök Neden $\rightarrow$ Çözüm Matrisi)

Jamstack ve Edge Rendering geçiş süreçlerinde en sık karşılaşılan teknik engeller ve çözüm yolları:

### Sorun Çözüm Matrisi (image\_d678e6.png)

Problem	Kök Neden	Çözüm	Öncelik
Bayat İçerik	Yanlış cache stratejisi	Revalidate + invalidation planı	Yüksek
SEO Sinyali Karışıyor	Canonical / hreflang QA yok	SEO QA gate + otomatik test	Yüksek
Edge Maliyeti Artıyor	Gereksiz edge logic	Edge'i sınırlı kullan, hibrit yap	Orta
Operasyon Zorlaşıyor	Sahiplik belirsiz	RACI + runbook	Orta

## 3) 14 Günlük Hızlandırılmış Sprint Planı (POC & Hibrit Geçiş Dönemi)

Mimari dönüşümü risksiz bir şekilde hayata geçirmek için adım adım iki haftalık çalışma planı:

- Gün 1: Sınıflandırma:** Sayfa tiplerinin (statik, dinamik) listelenmesi ve genel uygunluk skorunun çıkarılması.
- Gün 2: Hedefleme:** Trafik coğrafyası analizi ve sayfa bazlı performans hedeflerinin (TTFB ve Core Web Vitals) belirlenmesi.
- Gün 3: Kapsam Belirleme:** Jamstack/Edge dönüşümü için MVP (Minimum Uygulanabilir Ürün) sınırlarının kilitlemesi.

- **Gün 4: Ön bellek Tasarımı:** Detaylı cache stratejisi taslağının oluşturulması ve CDN invalidation planının yazılması.
- **Gün 5: SEO Planlaması:** Teknik SEO gereksinimleri (canonical, dinamik sitemap, yapılandırılmış veri) kontrol listesinin hazırlanması.
- **Gün 6: Güvenlik Planlaması:** Edge network katmanı ve API uç noktaları (endpoints) için güvenlik checklist'inin tamamlanması.
- **Gün 7: Altyapı Kurulumu:** İzole POC (Proof of Concept) ortamının ve staging dağıtım (deployment) mimarisinin kurulması.
- **Gün 8: Statik Göçü:** Blog, destinasyon rehberleri veya kurumsal hakkımızda gibi statik sayfaların CDN-first mimarisine taşınması.
- **Gün 9: Dinamik İşleme:** Seçilen tek bir kritik sayfa tipi üzerinde Edge Rendering (SSR/ISR) POC denemelerinin yapılması.
- **Gün 10: Performans Ölçümü:** Yeni yapının TTFB ve CWV metriklerinin baseline (eski durum) verileriyle kıyaslanması ve loglanması.
- **Gün 11: SEO Simülasyonu:** Arama motoru botlarının tarama senaryoları için SEO QA gate yapısının ilk deneme çalışmasının (dry-run) yapılması.
- **Gün 12: İnce Ayar:** Testlerde tespit edilen hataların düzeltilmesi ve CDN cache tuning (ön bellek optimizasyonu) işlemlerinin yapılması.
- **Gün 13: Sürdürülebilirlik:** Canlıya alım sonrası operasyon için teknik dokümantasyonun tamamlanması ve sistem runbook kılavuzunun yazılması.
- **Gün 14: Kapanış:** POC sonuçlarını içeren teknik karar raporunun yönetime sunulması ve genel ürün yol haritasına (roadmap) işlenmesi.

#### 4) Öncesi / Sonrası Performans ve Kalite Matrisi (KPI Takibi)

Mimari dönüşümün sisteme getirdiği somut kazanımların ölçülmesi:

 Performans Değerlendirme Tablosu (image\_d6791e.png)

KPI	Önce	Sonra (30 gün)	Hedef
TTFB (global)	TBD	TBD	↓(Düşmeli)
CWV (LCP/INP/CLS)	TBD	TBD	↑(Yükselmeli)
Deploy Sonrası Hata	TBD	TBD	↓(Düşmeli)
SEO QA Bulgusu	TBD	TBD	↓(Düşmeli)

#### B) Nasıl Doldurulur? (5 Kritik Kural)

1. **Sayfaları Körlemesine Tam Statik Yapmayın:** Tüm web sitesini tek bir kalıba zorlamayın; otel oda fiyatı veya B2B stok durumu gibi anlık değişen alanları tam dinamik (Edge SSR/CSR) bırakırken, blog ve galeri gibi alanları tam statik (SSG) kurgulayarak hibrit model seçin.
2. **Kritik Sayfalarda Stale Veri Sürelerini (SLA) Netleştirin:** "Stale-while-revalidate" stratejisi kullanırken, kullanıcının ne kadar süre eski veri görebileceğini iş birimleriyle netleştirin; rezervasyon adımlarında fiyatların bayat (stale) kalmaması için invalidation mekanizmasını anlık tetikleyin.
3. **SEO Kontrol Kapısını (QA Gate) CI/CD'ye Bağlayın:** Edge rendering süreçlerinde sayfa kaynak kodunun (SSR) arama motorlarına boş veya hatalı gitme riskine karşı; her deploy öncesinde canonical, sitemap ve meta verilerini otomatik test eden yazılımsal bariyerler kurun.
4. **Edge Katmanında Aşırı Kod Yükünden Kaçının:** Edge Worker veya Cloudflare Functions katmanına çok ağır iş mantıkları (business logic) yazarak mimariyi şişirmeyin; edge katmanını sadece yönlendirme, basit yetkilendirme ve hızlı HTML manipülasyonu için hafif tutun.
5. **Ekibin Bakım ve Monitoring Kapasitesini Ölçün:** Altyapıyı dönüştürmeden önce ekibin CI/CD süreçlerine ve CDN log izleme araçlarına ne kadar hakim olduğunu puanlayın; izleme ve loglama mekanizması kurulmadan geçilen bir Jamstack mimarisi kör uçuşa benzer.

### C) Teknik Kontrol Listesi (Checklist)

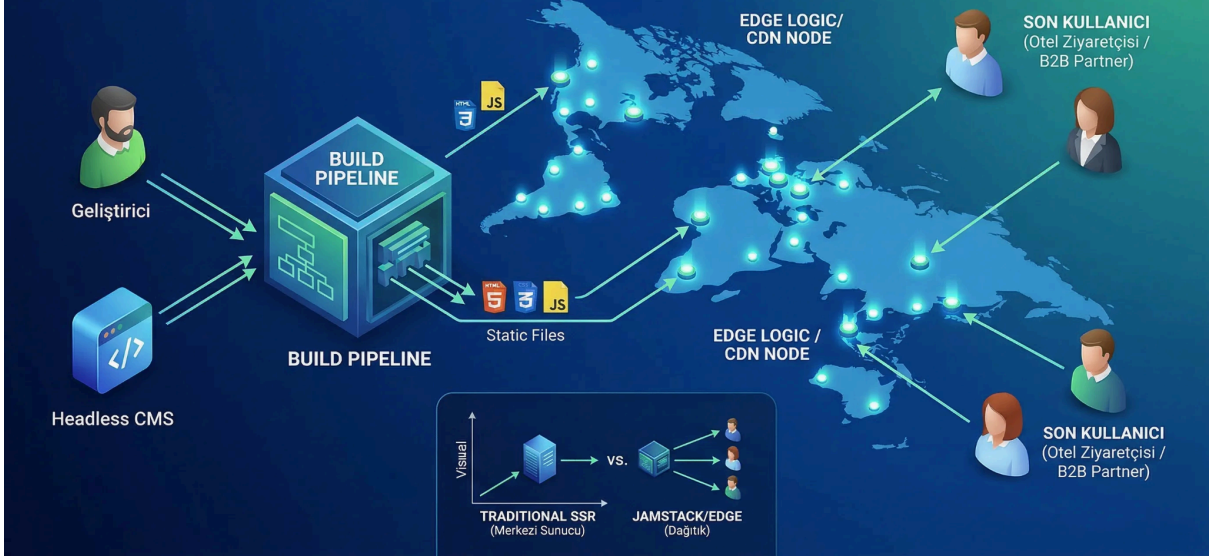
- [ ] **Sınıflandırma Yapıldı:** Web sitesindeki tüm sayfa şablonları statik ve dinamik iş yüklerine göre listelendi.
- [ ] **SLA Süreleri Belirlendi:** İçeriklerin güncellenme sıklığına göre CDN önbellek ömürleri ve bayat veri toleransları onaylandı.
- [ ] **SEO QA Gate Aktif:** Dağıtım (deploy) hattına entegre, arama motoru botlarının göreceği kaynak kodu denetleyen sistem kuruldu.
- [ ] **Edge Güvenliği Planlandı:** Edge katmanındaki API çağrıları için rate limiting ve temel web uygulama güvenlik önlemleri alındı.
- [ ] **14 Günlük Sprint Hazır:** POC kurulumundan canlı geçiş raporlamasına kadar olan 2 haftalık operasyon takvimi ekibe atandı.
- [ ] **KPI Göstergeleri Tanımlandı:** Dönüşümün başarısını kanıtlayacak TTFB, Core Web Vitals ve deploy hata takip metrikleri kilitlendi.

### Deliverables (Teslim Edilecek Çıktılar)

- Şirket Yönetimine Sunulacak 1 Adet Uygunluk Skoru ve Mimari Kapsam Karar Raporu
- Yazılım Ekibine İletilecek 1 Adet Sayfa Bazlı CDN Önbellek ve Edge Yönetim Stratejisi Dokümanı
- DevOps Takımı İçin 1 Adet Dağıtım Hattına Entegre Edilebilir Teknik SEO QA Gate Kontrol Listesi
- Sistem Güvenliğini Sağlayan 1 Adet Edge / API Sınır Güvenliği ve Yetkilendirme Notları Paketi
- Dönüşüm Başarısını Gösteren 1 Adet Öncesi / Sonrası POC Performans Değerlendirme Raporu

# Jamstack ve Edge Rendering Teknik Mimari Akışı

Corporatik Websiter



## [Architecture Hybrid Diagram]

Modern bir web projesinin hibrit altyapısını görselleştiren, 16:9 formatında tasarlanmış teknik bir mimari akış diyagramı. Gece siyahı zemin üzerinde, sol tarafta konumlanan "Kullanıcı İstekleri" dalgası, ortada parlayan bir kalkan olarak duran "Edge Network / CDN Global Nodes" katmanına ulaşıyor. Bu katmandan iki farklı yöne parlayan neon oklar dağılıyor: Birinci ok, statik sayfaları (SSG) doğrudan en yakın CDN önbelleğinden milisaniyeler içinde çekerek sol tarafa döndürürken; ikinci ok, yarı dinamik ve tam dinamik istekleri (Edge SSR) en yakın edge worker koduna veya mikro servis mimarisine yönlendirerek esnek veri akışını net bir şekilde modelliyor.



### [Performance Performance Dashboard Mockup]

*Teknik liderlerin altyapı geiři sonrasında elde ettikleri hız kazanımlarını takip edebilecekleri, 16:9 formatında minimalist bir performans takip paneli (dashboard) mockup görseli. Mat antrasit arka plan üzerinde, büyük ve parlayan neon yeřili metinlerle "Global TTFB: 35ms" ve "Core Web Vitals: PASS" ibareleri gururla sergileniyor. Ekranın alt yarısında yer alan iki adet çizgi grafik kartında, eski sunucu tabanlı mimarideki dalgalı ve yüksek gecikme süreleri (turuncu çizgi) ile Jamstack/Edge mimarisine geildikten sonraki tamamen düz, stabil ve sıfıra yakın gecikme çizgisi (neon mavi hat) yan yana kıyaslanarak görsel bir kanıt sunuyor.*