Mimari Fizibilite Raporu (Redesign Dönüşümü)

# 1. Amaç

Bu rapor, redesign dönüşümü kapsamında mevcut ve hedef teknolojilerin karşılaştırmalı değerlendirmesini yaparak uygulanabilirliklerini ve muhtemel risklerini analiz etmeyi amaçlamaktadır.

# 2. Teknoloji Değerlendirmesi

| Teknoloji | Mevcut Durum | Önerilen Durum | Açıklama |
| --- | --- | --- | --- |
| React | 18 | 19 | Server Components ve Concurrent özelliklerle daha yüksek performans. |
| Next.js | 14 | 15 (Son sürüm) | App Router, React 19 uyumu, Edge optimizasyonları. |
| State Yönetimi | Context API | Zustand + Context | Minimal ve performanslı Zustand kullanılacak. |
| Module Federation | Var | Devam edecek | Microfrontend mimarisi korunacak. |
| React Query | Kısmen var | Tamamlayıcı kullanım | Data fetching & caching standartlaştırılacak, API entegrasyonları için ortak katman. |
| TypeScript | 5.5 | 5.9(Son sürüm) | Yeni TS özellikleri ile tip güvenliği ve DX iyileşecek. |
| ESLint | 8.57 | Son sürüm | Kod kalitesi ve standartları için güncellenmiş lint kuralları. |
| SCSS | Var | Son sürüm | Style guide ile standartlaştırılmış SCSS kullanımı devam edecek. |
| i18Next | Var | Full type support | Çeviri süreçlerinde tip güvenliği artırılacak. |
| UI Kit (TK-UI) | Legacy bileşenler | Yeni TK-UI (Accessible + Design Tokens) | Erişilebilirliği yüksek, WCAG uyumlu bileşenler. Design tokens ile tutarlı tasarım. |
| Changelog | Yok | Yeni | Değişiklikler düzenli takip edilecek. |
| Versioning | Kısmen var | SemVer standardı | Sürüm yönetimi netleşecek. |
| Deploy Preview | Yok | Yeni | PR bazlı otomatik preview ortamları. |
| AI & Kod Standartları | Yok | GitHub Copilot + custom instructions | Kodlama standartları AI ile desteklenecek, prompt tabanlı rehberlik. |

# 3. Mimari Etki Analizi

- Performans:   
 \* React 19 + Next.js 15: SSR/ISR optimizasyonu.   
 \* Zustand + React Query: daha az re-render, gelişmiş state yönetimi.  
- Geliştirici Deneyimi (DX):   
 \* TK-UI + Design Tokens: tutarlı ve hızlı geliştirme.   
 \* Deploy Preview: QA/test kolaylığı.   
 \* SemVer + Changelog: Şeffaf release yönetimi.   
 \* GitHub Copilot AI destekli kod yazımı: custom promptlarla standartların korunması.  
 \* TypeScript 5.9 + ESLint: daha güçlü tip kontrol ve kod standardizasyonu.  
- Ölçeklenebilirlik:   
 \* Module Federation: domain bazlı ayrışma, bağımsız deploy.  
- Bakım Kolaylığı:  
 \* i18Next type safety: çeviri hatalarının önlenmesi.   
 \* TK-UI: erişilebilirlik ve sürdürülebilir UI standardı.   
 \* Kod standartlarının AI ile otomatize edilmesi: daha az teknik borç.  
 \* SCSS: standart stil yönetimi ile UI tutarlılığı.

# 4. Sonuç ve Öneri

Bu dönüşüm ile:  
- Modern React/Next stack üzerine inşa edilmiş, erişilebilirliği yüksek ve sürdürülebilir bir mimari elde edilecek.  
- Design Tokens ve TK-UI ile tasarım tutarlılığı sağlanacak.  
- AI destekli kod standartları ile üretkenlik artacak, best practice’ler korunacak.  
- Versioning, Changelog, Deploy Preview süreçleriyle release yönetimi kurumsallaşacak.  
- TypeScript + ESLint güncellemeleri ve SCSS standartları ile kod kalitesi artacak.