

# Blockchain Tabanlı Sertifika Sistemi - Teknik Rapor

## 1. Proje Mimari ve Tasarımı

### 1.1 Mimari

- **Frontend:** React.js ile geliştirilmiş, kullanıcı dostu arayüz.
- **Smart Contracts:** Ethereum blockchain üzerinde Hardhat/Truffle ile yazılmış akıllı kontratlar.
- **Blockchain:** Ethereum, sertifikaların doğrulanabilir ve değiştirilemez olmasını sağlar.
- **Bağlantı:** Web3.js veya Ethers.js ile frontend ve blockchain etkileşimi.

Kullanıcı <-> React Frontend <-> Ethers.js/Web3.js <-> Ethereum Blockchain (Smart Contracts)

**Docker Compose Yapısı:** Proje, Docker Compose ile çoklu servis olarak çalıştırılabilir:

- **ganache:** Ethereum test ağı için Ganache kullanılır. 10 test hesabı oluşturur, her blok 1 saniyede üretilir ve 0.0.0.0 üzerinden tüm IP'lerden erişilebilir.
- **hardhat:** Node.js üzerinde Hardhat çalıştırır, smart contractları deploy ve test eder.
- **client:** React frontend servisi, Ganache ve Hardhat ile etkileşir.
- **certnet:** Tüm servisler aynı özel ağ üzerinden iletişim kurar.

### 1.2 Tasarım Gerekçeleri

- **Güvenlik:** Ethereum blockchain değiştirilemezliği sağlar.
- **Doğrulanabilirlik:** Her sertifika hashlenip kontrata kaydedilir.
- **Esneklik:** Hardhat/Truffle ile test ve deploy kolaylığı.
- **Kullanıcı Deneyimi:** React ile modern, responsive UI.

## 2. Güvenlik ve KVKK Değerlendirmesi

### 2.1 Güvenlik

- Akıllı kontratlar test ağı üzerinde test edildi (Hardhat Local Network).
- Özel anahtarlar ve cüzdan bilgilerinin frontend dışında tutulması.
- Sertifikalar blockchain üzerinde hashlenmiş şekilde saklanır, kişisel bilgiler şifrelenebilir.

### 2.2 KVKK (Kişisel Verilerin Korunması)

- Sadece doğrulama için gerekli bilgiler blockchain'e kaydedilir.

- Kişisel veriler doğrudan blockchain'e kaydedilmez, hash veya anonimleştirilmiş veri kullanılır.
- Kullanıcıların veri üzerindeki hakları frontend aracılığıyla yönetilebilir.

## 3. Test Kanıtları

### 3.1 Hardhat Testleri

```
cd dapp  
npx hardhat test
```

- Tüm smart contract fonksiyonları doğru çalışıyor.
- Örnek testler: sertifika oluşturma, doğrulama, erişim kontrolü.

### 3.2 Frontend Testleri

- React uygulaması tarayıcıda çalıştırıldı ve Metamask bağlantısı test edildi.
- Sertifika oluşturma ve doğrulama iş akışları sorunsuz.

## 4. Sonuç

- Blockchain tabanlı sertifika sistemi güvenli, doğrulanabilir ve KVKK uyumlu olarak tasarlanmıştır.
  - React frontend ve Ethereum smart contract entegrasyonu başarılı bir şekilde gerçekleştirilmiştir.
  - Testler tüm kritik fonksiyonların doğru çalıştığını göstermektedir.
-