

i-ACS Bulut Tabanlı Geçiş Kontrol Sistemi – Sistem Tasarımı Dökümanı

1. Giriş

Bu döküman, Geçiş Kontrol Sistemi için önerilen sistem tasarımını detaylandırmaktadır. Tasarım, mikro servis mimarisi, uygulama geliřtirmesi, veritabanı seçimi, ölçeklenebilirlik ve hata ele alma stratejilerini içermektedir.

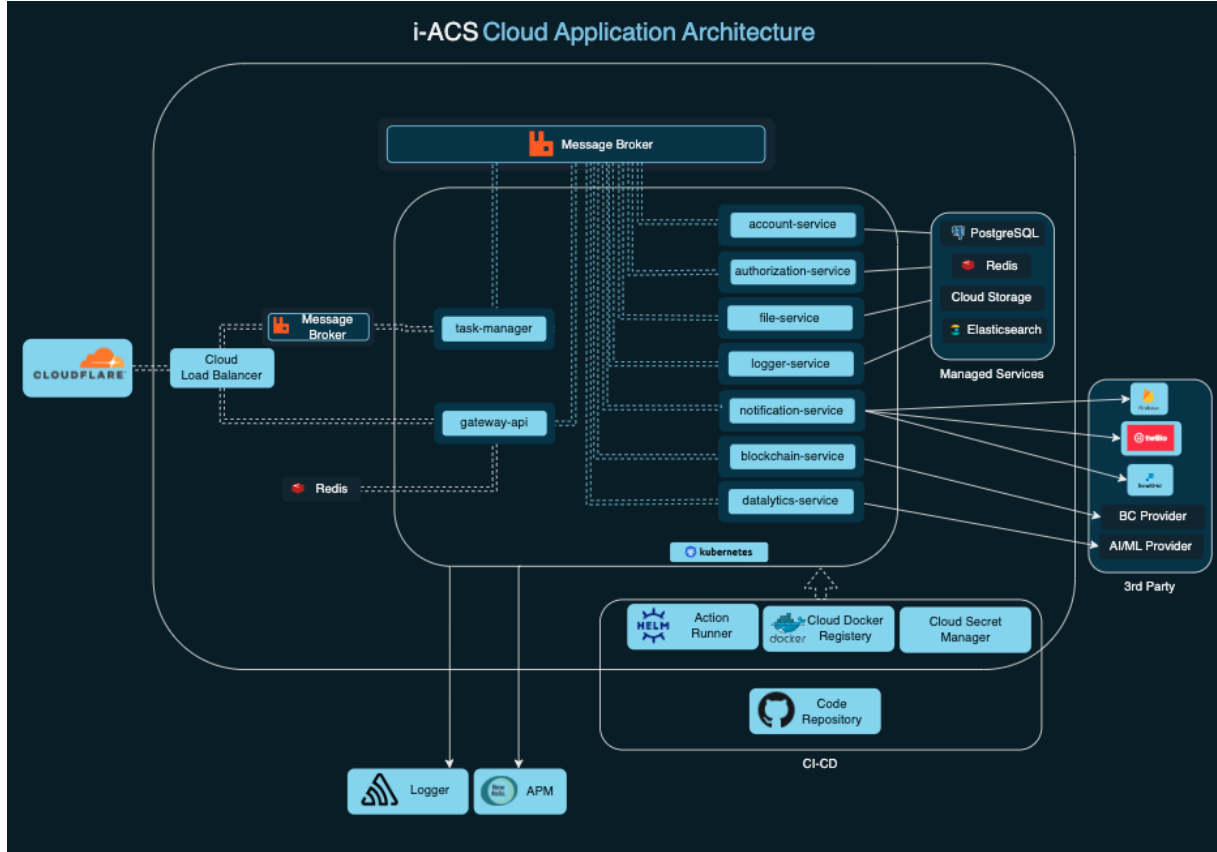
Proje kapsamında

- Dış dünya iletişimi olarak bulut **load-balancer** arkasından çalışmakta olan **bir http gateway**, **1 message broker** ile iletişim sağlanması planlanmakta olup, projenin 1 composite servis ve 7 mikro servis ile çalışması planlanmıştır.
- Dil bütünlüğü sağlanması adına AI/ML servisi haricinde tüm uygulamalar **NestJS** ile hazırlanması planlanmıştır. AI/ML yetenekleri konusunda çok geniş bir dökümantasyon ve destek kütüphanesi bulunan **Python** ile yapılması tercih edilmiştir.
- Proje hem cihazlarla senkronizasyon amacıyla hem de mikro servisler arasında iletişimi sağlamak amaçlı **RabbitMQ** kullanımı tercih edilmiş, tüm mikro servislerin **message-driven** mimariye uygun şekilde kodlanması planlanmıştır.
- Veri yapıları hakkında detay bilgiye sahip olunmadığı için temel seviyede veritabanı yapıları seçilmiş olup, bu kapsamda RDBMS olarak **PostgreSQL**, In memory veritabanı olarak **Redis**, Full Text Search kapsamında **Elasticsearch** kullanımı tercih edilmiştir.
- Proje kapsamında NFT kullanımının olabileceği göz önüne alınarak blockchain servisi hazırlanmıştır.
- Proje kapsamında ölçeklenme sorunu Kubernetes Node Scaling ile dikey büyüme, Helm Chart Pod Scaling ile de yatay ölçeklendirme sorunlarının önüne geçilmesi planlanmıştır.

2. Mimari Diyagram

Bu mimari tamamen Bulut Tabanlı bir sistem göz önüne alınarak hazırlanmış olup, on-side kurulum gerektiren durumlarda farklı bir topoloji değerlendirilmesi gerekmektedir.

Özellikle birden fazla bölgede faaliyet müşteriler için on-side talepleri üzerine EVM(Ethereum Sanal Makinası) forklarıyla L1, L2 blockchain çözümleri geliştirilebileceği öngörülmektedir.



3. Mikro servis Teknolojileri ve Veritabanı Seçimi

Tüm Uygulamaların Ortak Özellikleri

- **Interceptorlar** üzerinden **NewRelic** ile APM entegrasyonu yapılarak tüm isteklerin uçtan uca izlenmesi ve detaylı performans değerlendirmelerinin elde edilmesi planlanmaktadır.
- **Filterlar** üzerinden hata yakalama yapılarak **Slack** veya farklı bir 3. Parti uygulamaya iletilmesi planlanmaktadır.
- Tüm mikro servislerde **DTO** ile data doğrulamaları yapılarak parametre kaynaklı hatalara anlamlı cevaplar dönülmesi planlanmaktadır.
- Test Coverage oranı minimum **70%** olarak planlanmaktadır. Testlerin Git Flow kapsamında develop branche PR merge edilmesi durumunda Github Runner üzerinden çalıştırılması planlanmaktadır.
- Cloud bağımsızlığı sağlanması adına, CI süreçlerinin Github Actionslar ile yönetilmesi ve konfigürasyon yönetiminin k8s-secrets üzerinden kullanılması planlanmaktadır.
- Hazırlanan uygulamaların Dockerfile ile dockerize edilmesi ve helm chart ile K8s deployment yapılarak CD süreçlerinin hazırlanması planlanmaktadır.

3.1 Restful HTTP Gateway(gateway-api)

Teknoloji: Nest.JS, Typescript, Express

Veritabanı: İsteklerin yetkilendirmesini yapmak ve session yönetimini sağlamak adına en kısa sürede yanıt üretebilmek için in-memory veritabanı olarak **Redis** seçilmiştir.

Görevi: Sistem kullanıcılar ve cihazlardan(HTTP, WSS destekli) gelen istekleri **SAGA** orkestrasyonu mimarisi ile organize ederek ilgili servislerden verileri toplayıp dönüş yapmaktan sorumludur.

Özellikleri:

- Entegre Swagger Dökümantasyonu
- Redis üzerinden JWT tabanlı yetkilendirme kontrolü
- Endpoint bazlı rate-limit
- Redis üzerinden caching yeteneği

3.2 Task Manager(task-manager)

Teknoloji: Nest.JS composite servis, Typescript

Görevi: Message Broker destekli cihazlardan gelen isteklere consumer olarak davranıp, diğer mikro servislere producer olarak davranacak composite servis görevi görmektir. Bunun dışında raporlama veya notifikasyon gibi zamanlanmış görevleri çalıştırmaktan sorumludur.

3.3 Hesap Yönetim mikro servisi(accounting-service)

Teknoloji: Nest.JS mikro servis, Typescript, Typeorm

Veritabanı: Kullanıcı, Cihaz ve Kart bilgilerinin tutulması amaçlı **PostgreSQL** seçilmiştir.

Görevi: Kullanıcı hesapları, Cihaz ve Kart bilgilerinin tanımlanması ve bunlarla alakalı tüm CRUD operasyonlarının gerçekleştirilmesinden sorumludur.

Özellikleri:

- Data validasyonu yaparak anlamlı hatalar üretmek,
- TypeORM migration desteği

3.4 Geçiş Yetki Kontrol mikro servisi(auhorization-service)

Teknoloji: Nest.JS mikro servis, Typescript

Veritabanı: Geçiş işlemlerindeki en kısa sürede yanıt üretebilmek için in-memory veritabanı olarak **Redis** seçilmiştir. Veri yapısı olarak da cihaz bazında **Sets** kullanılması düşünülmüştür.

Görevi: Tüm geçiş kontrol yetkilerinin tutulması ve bunların yetkilendirme isteklerine cevap vermek ile sorumludur.

3.5 Dosya yönetim mikro servisi(file-service)

Teknoloji: Nest.JS mikro servis

Görevi: Proje kapsamında kullanıcı tarafından bulut depolama sistemine yapılacak tüm upload/download istekleri için geçici süreli ve güvenli tek kullanımlık bir stream oluşturmaktan sorumludur.

Özellikleri:

- Cloud Storage Entegrasyonu
- Cloud API Entegrasyonu

3.6 Loglama Servisi(logging-service)

Teknoloji: Nest.JS mikro servis

Veritabanı: Loglama ihtiyacı için geniş filtreleme yeteneği ve yüksek sayıda kayıta hızlı operasyon yapabilme yeteneği nedeniyle **Elasticsearch** kullanımı tercih edilmiştir.

Görevi: Tüm donanımlardan gelecek logların uygun bir formata ve koleksiyonlara dönüştürülerek ilgili veritabanına yazılmasını sağlamaktan sorumludur.

Özellikleri:

- Cloud Storage Entegrasyonu
- Cloud API Entegrasyonu
- Dosya tabanlı blockchain entegrasyonu

3.7 Notifikasyon Servisi(notification-service)

Teknoloji: Nest.JS mikro servis

Görevi: Kullanıcılara gönderilecek tüm SMS, Push Notification veya E-Mail notifikasyonların gönderimini sağlamaktan sorumludur.

Özellikleri:

- Twilio Entegrasyonu
- Sendgrid Entegrasyonu
- Firebase Entegrasyonu

3.8 Blockchain Servisi(blockchain-service)

Teknoloji: Nest.JS mikro servis, EVM base blockchain

Görevi: Seçilen blockchaine senkronizasyonu sağlayarak, Esnek geçiş kontrollerinde(Vendor Kullanıcıları, Passolig vb.) ihtiyaç duyulacak, hesap oluşturma, NFT üretme, sahiplik kontrolü gibi temel blockchain operasyonlarını yürütmekten sorumludur.

Özellikleri:

- Blockchain Entegrasyonu

3.9 Data Analiz Servisi(dataalytics-service)

Teknoloji: Python

Görevi: Loglanan tüm kullanıcı ve cihaz kayıtlarının periyodik olarak kontrollerini sağlayıp, geliştirilen AI/ML modelini güncellemek ve güncellenen modeli kullanarak altta belirtilen tahminleri raporlamaktan sorumludur.

- Kullanıcı bazlı anomali tespiti,
- Kullanıcı aktiviteleri bazlı anomali tespiti,
- Cihaz bazlı anomali tespiti,
- Cihaz aktiviteleri bazlı anomali tespiti,
- Cihaz hata tahminleme,
- Trend Analizi