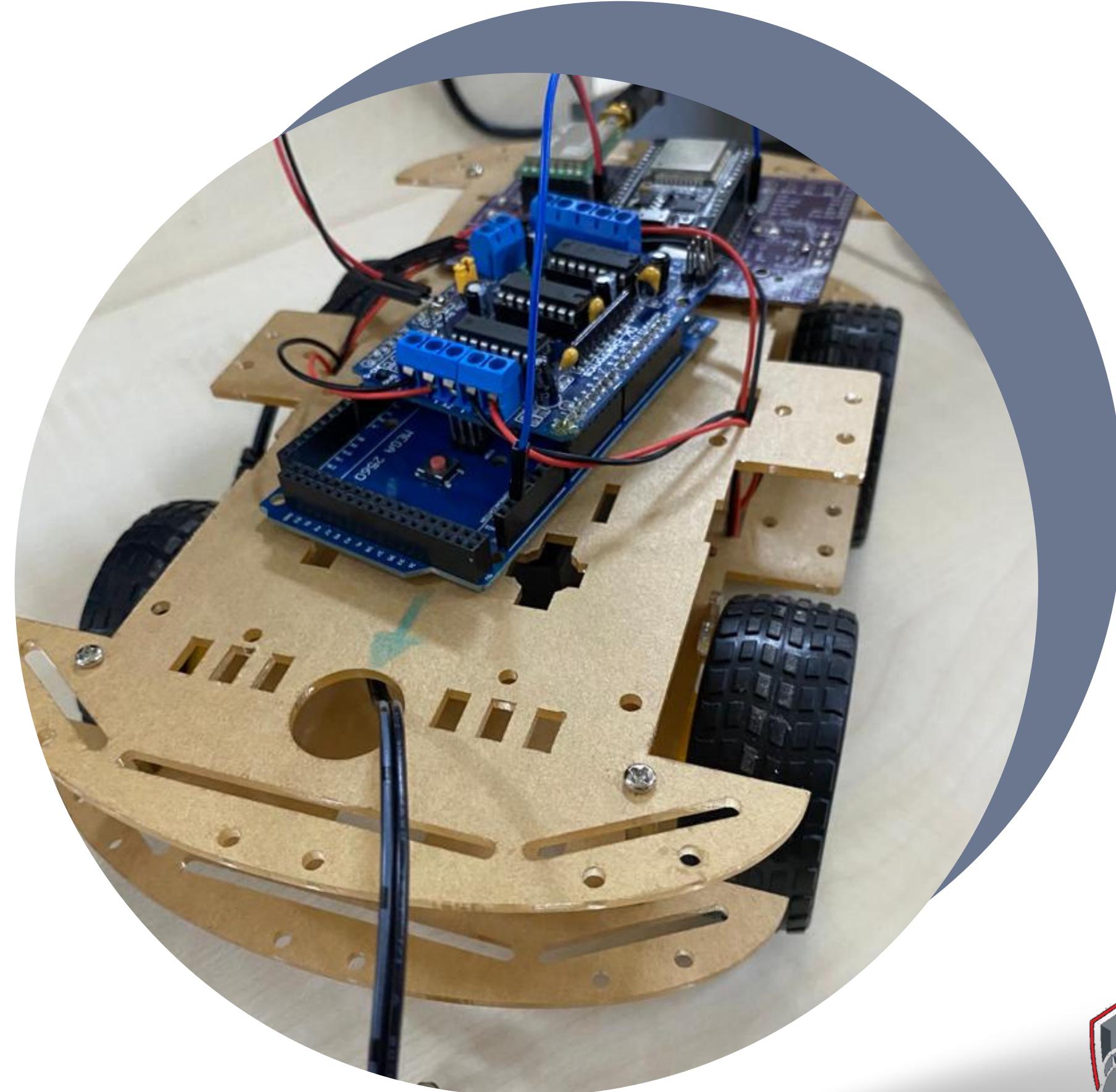


Teknofest - 2024

YİĞİDO TEAM

Sanayide Dijital Teknolojiler



Sunum İçeriği

Front-End Sunumu

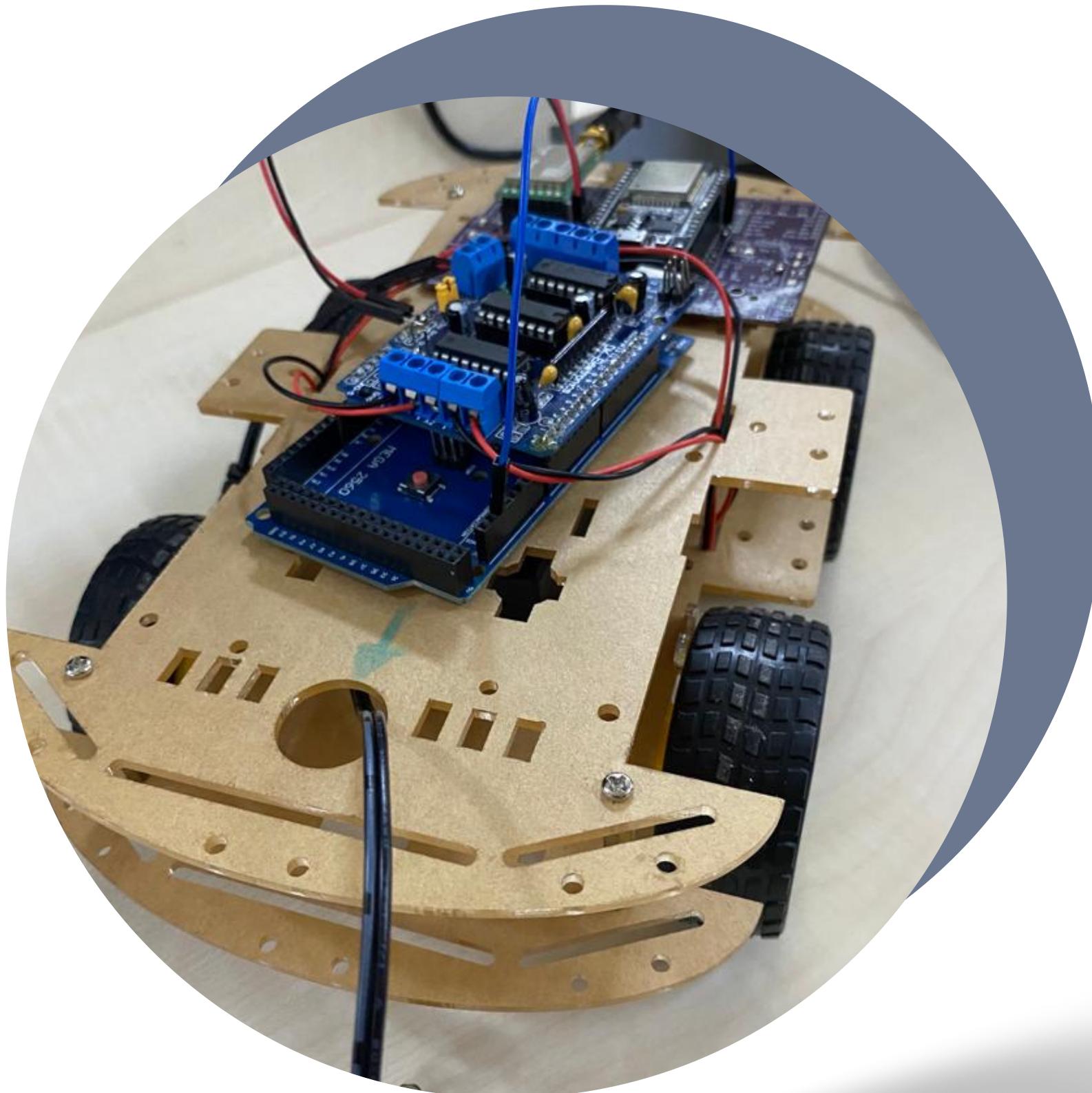
Mobil Sunumu

Back-End Sunumu

Server Sunumu

Elektronik Yapısının Sunumu

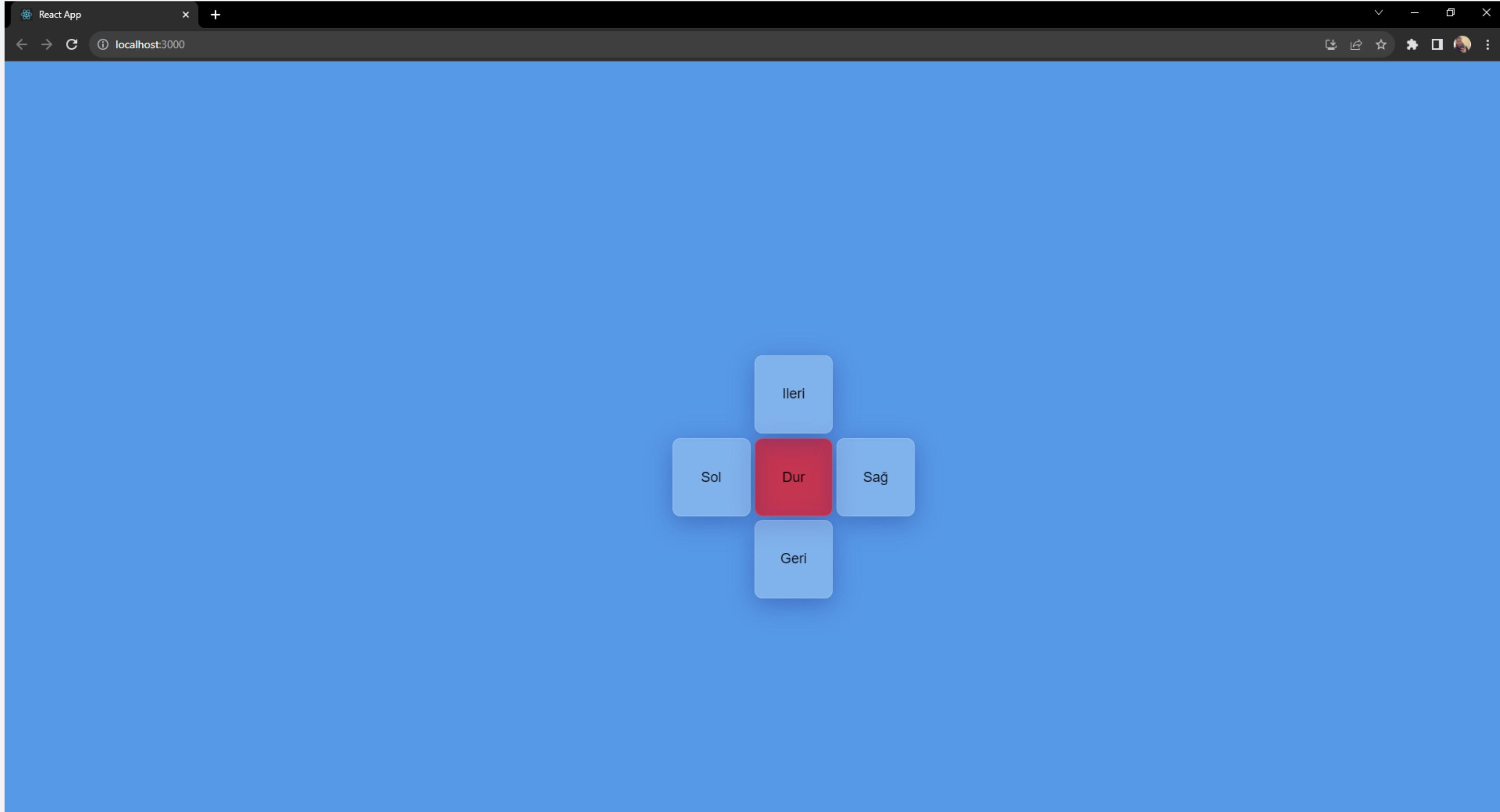




4WD uzaktan kontrolü
olan robotumuz hem
mobil hem web tabanlı
kontrole sahiptir.



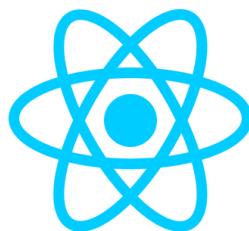
Web Kullanıcı Arayüzü



Web Kullanıcı Arayüzü

Frontend kısmında kullanılan teknolojiler:

→ axios



Kullanımı kolay, hızlı, hata durumlarında kolayca müdahale edilmesi, url ve verilerin esnek yönetim olanağıları için robot arayüzde axiosu tercih edildi.

React, bileşen tabanlı, performans odaklı ve geniş bir topluluk desteği sahip bir kullanıcı arayüzü kütüphanesi olduğu için tercih edildi.

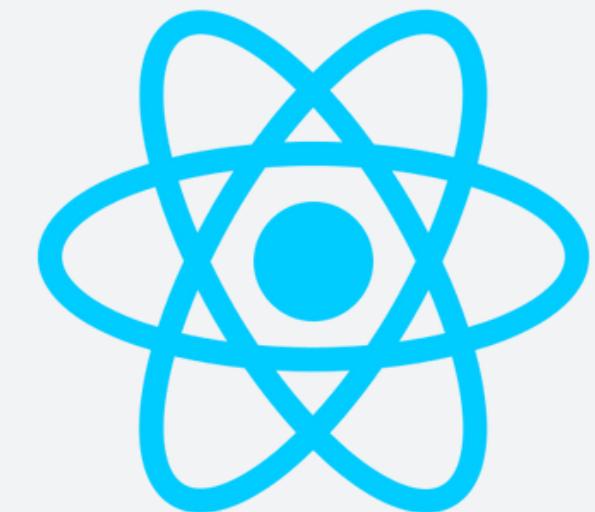


Web sayfalarını stil ve düzenleme için tercih edilir, çünkü görünümünü ayrı bir yapıdan ayırarak sayfa tasarımını esnek ve kolayca değiştirilebilir kılar.



REST API, basit ve standart HTTP protokolünü kullanarak veri alışverişi yapmayı kolaylaştırdığı için tercih etti.

→ axios



css



Mobil Kullanıcı Arayüzü

Mobil kısmında kullanılan teknolojiler:

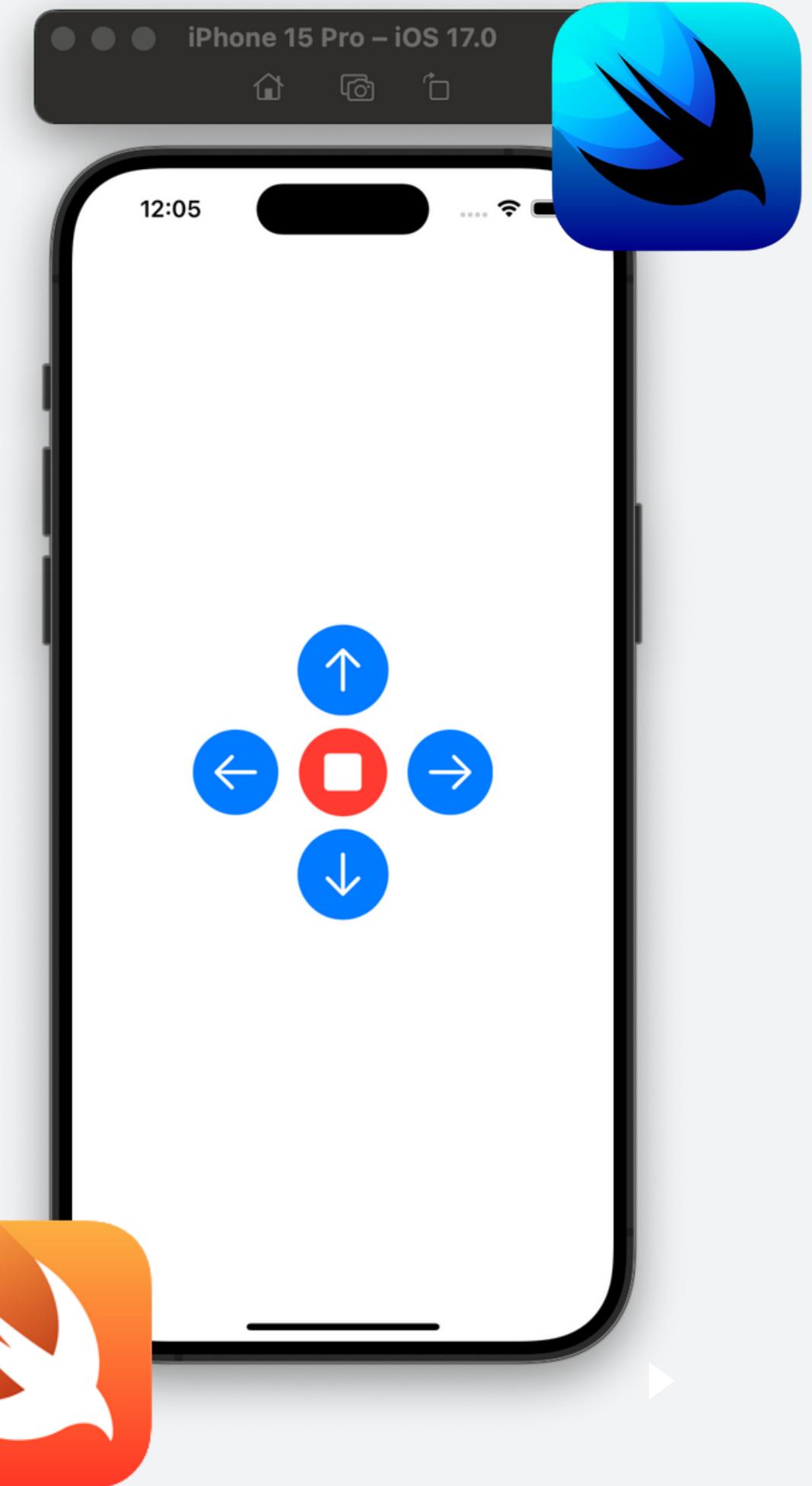
SwiftUI, iOS için kullanıcı arayüzü (UI) yapılmasını sağlayan bir programlama dilidir. SwiftUI çok az kod satırı ile arayüz tasarılanır.

SwiftUI robot tasarımlı uygulaması API'ye HTTP POST isteği yaparak kullanılan bir veri modeli ve işlevini içerir. Bu arayüz yön düğmelerine basıldığında isteği sunucuya gönderir.

Yön tuşlarını temsil eden beş düğme içeren bir arayüz tasarımlı vardır. Her düğmeye tıklanlığında ilgili yön bilgisini içeren JSON verisini oluşturur ve sendButtonData fonksiyonunu çağırarak bu veriyi bir API'ye gönderir.



REST (Representational State Transfer), web ve mobil için kullanılan bir mimari tarzıdır. REST API, sunucu ve istemciler arasında veri iletişimini sağlamak için kullanılır.



Mobil Kullanıcı Arayüzü

The image shows two side-by-side screenshots of the Xcode IDE. Both screenshots display the same project structure and code, illustrating the implementation of a mobile user interface using SwiftUI.

Project Structure:

- SwiftUI_REST
- SwiftUI_RESTTests
- SwiftUI_RESTUITests
- SwiftUI_REST...sLaunchTests
- Views

 - ContentView
 - ButtonData

- SwiftUI_RESTApp
- Assets
- Preview Content

ContentView.swift (Left Screenshot):

```
1 //  
2 //  ButtonData.swift  
3 //  SwiftUI_REST  
4 //  
5 //  Created by Şeymanur Köse on 30.10.2023.  
6 //  
7  
8 import Foundation  
9  
10 enum ButtonType: String, Codable {  
11     case up = "1"  
12     case down = "3"  
13     case left = "4"  
14     case right = "2"  
15     case stop = "0"  
16 }  
17  
18 struct ButtonData: Codable {  
19     var message: ButtonType  
20 }  
21  
22 func sendButtonData(data: ButtonData) {  
23     guard let jsonData = try? JSONEncoder().encode(data) else {  
24         print("JSON verisine dönüştürme hatası")  
25         return  
26     }  
27  
28     let url = URL(string: "http://192.168.1.98:8080/api/messages")! //teknofest ip  
29     var request = URLRequest(url: url)  
30     request.httpMethod = "POST"  
31     request.setValue("application/json", forHTTPHeaderField: "Content-Type")  
32  
33     request.httpBody = jsonData  
34  
35     let task = URLSession.shared.dataTask(with: request) { data, response, error in  
36         guard let data = data, error == nil else {  
37             print(error?.localizedDescription ?? "Veri yok")  
38             return  
39         }  
40  
41         let responseJSON = try? JSONSerialization.jsonObject(with: data, options: [])  
42         if let responseJSON = responseJSON as? [String: Any] {  
43             print(responseJSON)  
44         }  
45     }  
46     task.resume()  
47 }  
48  
49  
50  
51  
52  
53  
54  
55  
56
```

ContentView.swift (Right Screenshot):

```
1 //  
2 //  ContentView.swift  
3 //  SwiftUI_REST  
4 //  
5 //  Created by Şeymanur Köse on 30.10.2023.  
6 //  
7  
8 import SwiftUI  
9  
10 struct ContentView: View {  
11     var body: some View {  
12         ZStack {  
13             Color.white  
14                 .ignoresSafeArea()  
15  
16         VStack {  
17             Button(action: {  
18                 print("Yukarı yön tuşuna basıldı")  
19  
20                 let buttonData = ButtonData(message: .up)  
21                 if let jsonData = try? JSONEncoder().encode(buttonData) {  
22                     if let jsonString = String(data: jsonData, encoding: .utf8) {  
23                         print(jsonString)  
24                     }  
25                 }  
26  
27                 sendButtonData(data: buttonData)  
28             }) {  
29                 Image(systemName: "arrow.up")  
30                     .font(.largeTitle)  
31                     .foregroundColor(.white)  
32                     .padding()  
33                     .background(Color.blue)  
34                     .clipShape(Circle())  
35             }  
36  
37         HStack {  
38             Button(action: {  
39                 print("Sol yön tuşuna basıldı")  
40  
41                 let buttonData = ButtonData(message: .left)  
42                 if let jsonData = try? JSONEncoder().encode(buttonData) {  
43                     if let jsonString = String(data: jsonData, encoding: .utf8) {  
44                         print(jsonString)  
45                     }  
46                 }  
47  
48                 sendButtonData(data: buttonData)  
49             }) {  
50                 Image(systemName: "arrow.left")  
51                     .font(.largeTitle)  
52                     .foregroundColor(.white)  
53                     .padding()  
54                     .background(Color.blue)  
55                     .clipShape(Circle())  
56             }  
57         }  
58     }  
59 }
```

Backend Teknolojileri

Robotumuzda kullandığımız teknolojiler:



JAVA

17.0

En yeni kararlı sürümlerden biri ve docker image'si de aktif kullanılabilir olduğu için bu sürümü tercih ettim.



SPRING BOOT

3.1

Spring'in en son sürümü olması sebebiyle bu sürümü kullanmayı tercih ettim.



MAVEN

3.9

Derleyici ve otomasyon aracıdır. Bize bir ana yapılandırma dosyası sunar ve büyük bir bağımlılık deposu vardır.



DOCKER

24.0

Projemizi docker image'si haline getirerek sanal ortama taşıması sebebiyle bu teknolojiyi tercih ettim.



DOCKER COMPOSE

3.7

Bir çok Docker image'sini birlikte oluşturmayı sağlayan teknolojidir ve server dahil bir çok alanda kullanmayı tercih ettim.



POSTGRESQL

15.2

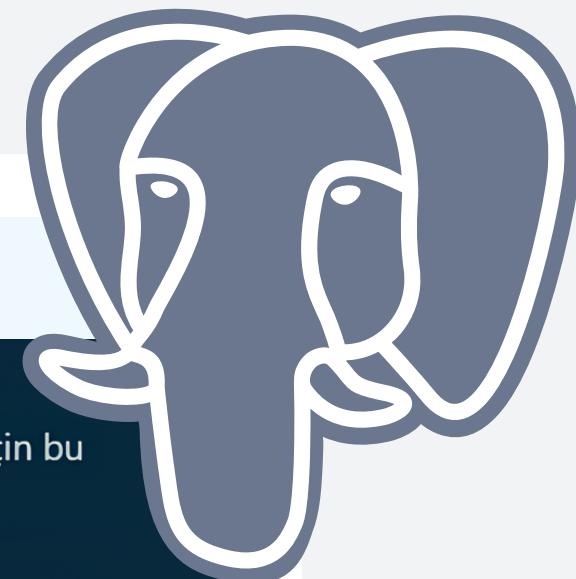
Verilerin kaydını tutmak için bir veritabanına ihtiyacımız vardı. Bu ihtiyacı gidermek için hali hazırda konfigürasyon ayarları elimizde bulunan bu veritabanını tercih ettim.



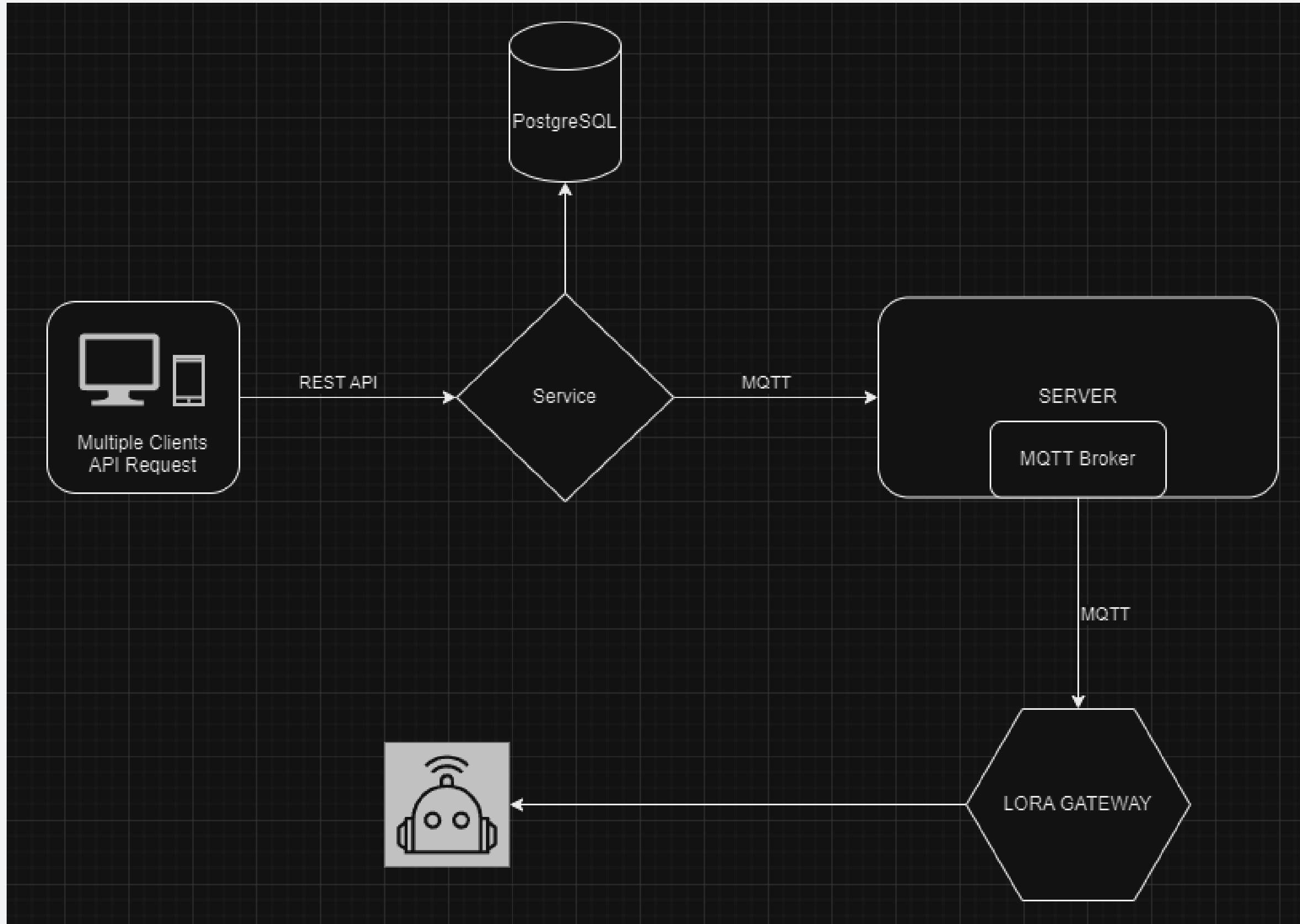
MOSQUITTO

2.0

MQTT Broker yapısını bize sağlayan açık kaynaklı bir teknoloji olduğu için bu teknolojiyi tercih ettim.



Akış Şeması:



Server

Server kısmında kullanılan teknolojiler:

LINUX :

Linux, birçok farklı platformda ve kullanım senaryosunda yaygın olarak kullanılan bir işletim sistemidir. Hem bireysel kullanıcılar hem de büyük kuruluşlar için güçlü ve esnek bir çözüm sunar.

Ubuntu Server ve Avantajları :

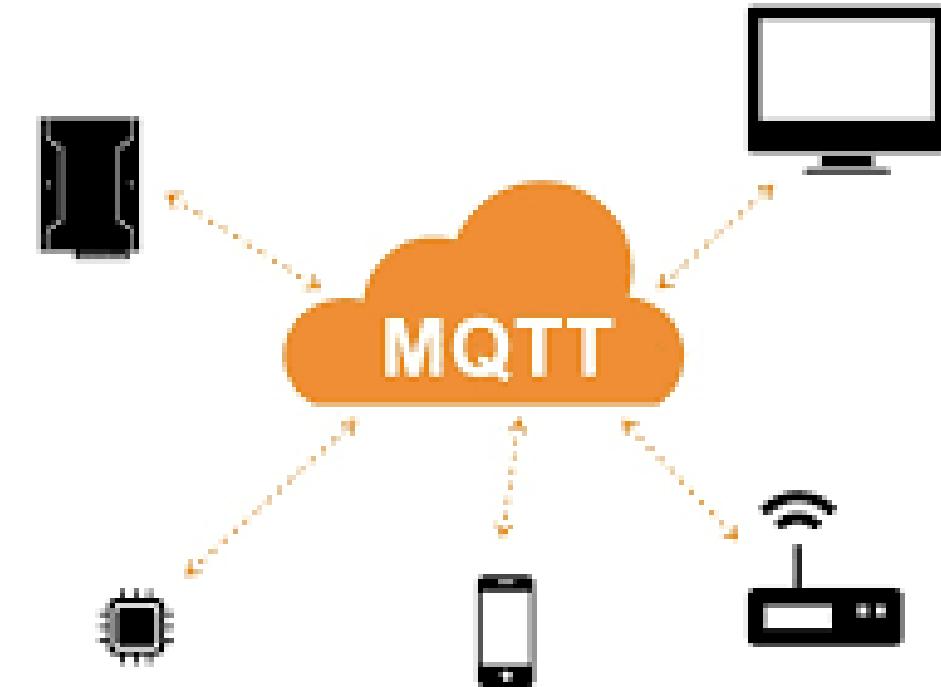
Ubuntu Server, Ubuntu'nun özgür ve açık kaynaklı bir işletim sistemi dağıtımıdır.

1-Stabilite

2-Açık Kaynaklı Yazılım

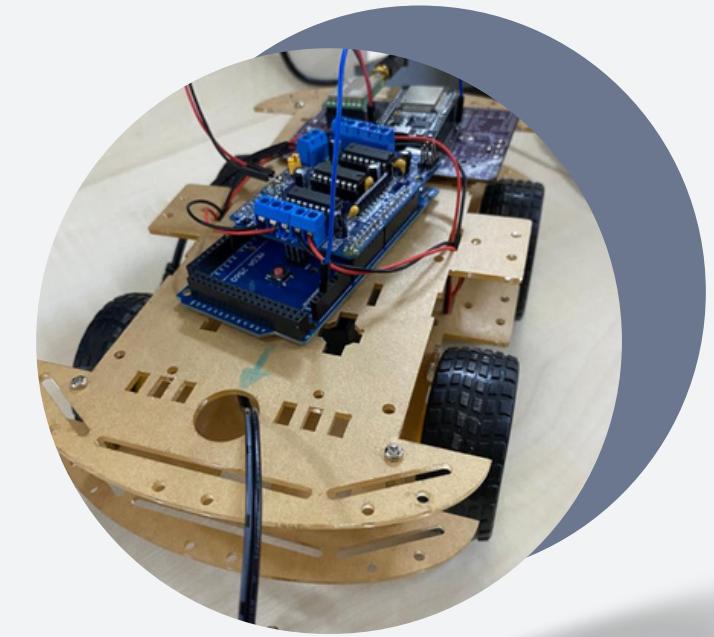
3-Güvenlik

Ubuntu sunucunuzda kurulu olan MQTT broker sayesinde, backend'in sabit bir broker IP adresi oluşturulur ve her istek bu sabit IP adresine yönlendirilir.





Robotun Elektronik Yapısı



ARDUNIO

- Arduino Mega kullanılarak robotumuza gerekli olan kodlar yüklenmiştir.

L293D MOTOR SHIELD

- Arduino'nun üzerine yerleştirdiğimiz motor sürücümüz sayesinde 4 motora güç verdik. Bu sayede robotta oluşabilecek denge problemlerini engelledik.

LORA GATEWAY

- Gateway kullanılarak iletişim güçlendirilmiş ve mesafe artırılmıştır.

ESP32

- Haberleşme için ESP32 kullanarak robotun uzaktan kontrolünü sağladık.

BATARYA

- Motorların hareketi için gerekli olan güç batarya sayesinde verilmiştir.
- 2S 18650 Li-Ion 2600mAh pil kullanılmıştır.

SON

DİNLEDİĞİNİZ İÇİN TEŞEKKÜRLER!