MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİDE UZAKTAN İZLEME KONTROL ve BAKIM YÖNTEMLERİNE ÇÖZÜM ÖNERİSİ OLACAK BİR CİHAZ TASARIMI

MEHMET BEZENK

YÜKSEK LİSANS TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
(TÜRKÇE) ANABİLİM DALI
Elektrik-Elektronik Mühendisliği Programı

DANIŞMAN
Prof. Dr. Hayriye KORKMAZ

İSTANBUL,2024

MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Öğrencisi Mehmet BEZENK'in "Endüstride Uzaktan İzleme, Kontrol ve Bakım Yöntemlerine Çözüm Olabilecek Bir Cihaz Tasarımı" başlıklı tez çalışması, 23 Ekim 2024 tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

Jüri Üyeleri

Prof. Dr. Hayriye KORKMAZ (Danışman)
Marmara Üniversitesi.....(İMZA).....

Dr. Öğretim Üyesi Ulvi BAŞPINAR (Üye)
Marmara Üniversitesi.....(İMZA).....

Dr. Öğretim Üyesi Musa AYDIN (Üye)
Fatih Sultan Mehmet Vakıf Üniversitesi.....(İMZA).....

ONAY

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve …….sayılı kararı ile Mehmet BEZENK'in Elektrik-Elektronik Mühendisliği(Türkçe) Anabilim Dalı Elektrik-Elektronik Mühendisliği Programında Yüksek Lisans derecesi alması onanmıştır.

Fen Bilimleri Enstitüsü Müdürü
Prof. Dr. Bülent EKİCİ

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

Mehmet Bezenk, a Master of Science student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended his thesis entitled "Device Designing Monitoring, Control and Maintenance Management", on October 23, 2024 and has been found to be satisfactory by the jury members.

Jury Members

Prof. Dr. Hayriye KORKMAZ (Advisor)
Marmara University.....(SIGN).....

Dr. Öğretim Üyesi Ulvi BAŞPINAR (Jury Member)
Marmara University.....(SIGN).....

Dr. Öğretim Üyesi Musa AYDIN (Jury Member)
Fatih Sultan Mehmet Vakıf University.....(SIGN).....

APPROVAL

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Mehmet BEZENK be granted the degree of Master of Science in Department of Electrical-Electronics Engineering, Electrical-Electronics Engineering Program on…….. (Resolution no .....).

Director of the Institute
Prof. Dr. Bülent EKİCİ

TEŞEKKÜR

Yüksek lisans eğitimim için destek veren ve çalışanı olduğum Eliar Elektronik Şirketi yöneticilerine tez çalışmamda değerli katkılarını ve desteklerini esirgemeyen tez danışmanım Prof. Dr. Hayriye KORKMAZ'a, yüksek lisans boyunca aldığım derslerle ufkumu genişleten ve farklı bakış açıları kazanmamı sağlayan değerli öğretmenlerime teşekkürü borç bilirim.

Bu süreçte sağladıkları destek ve motivasyon için değerli aileme teşekkürü bir borç bilirim.

Ekim 2024 **Mehmet BEZENK**

İÇİNDEKİLER

ÖZET i

ABSTRACT ii

SEMBOLLER ve BİRİMLER iii

KISALTMALAR iv

ŞEKİL LİSTESİ v

TABLO, ALGORİTMA VE KONFİGÜRASYON LİSTELERİ vi

1. GİRİŞ 1

1.1. Endüstride Uzaktan Erişim Modelleri 1

2. BENZER ÇALIŞMALAR 5

2.1. Literatürdeki Kaynak Araştırması 5

3. AĞ GEÇİDİ (GATEWAY) CİHAZI GELİŞTİRME AŞAMALARI 9

3.1. Donanım Seçimi ve Kurulumu 9

3.2. Node-RED Uygulamasının Kurulumu ve Yapılandırması 13

3.3. Iptables Yapılandırması 18

3.4. Hotspot Kurulumu 19

3.5. Nginx Proxy Kurulumu ve Stream Yapılandırması 20

3.6. Wireguard Sunucu ve Arayüz Ekranı 22

3.7. Android Studio ile Mobil Uygulama Geliştirme 25

3.8. Donanım Tasarımı ve İşlevleri 33

4. UZAKTAN İZLEME ve KONTROL 37

4.1. Senaryo 1: Mobil bir cihaz üzerinden uzak makineye bağlantı 37

4.2. Senaryo 2: Bir PC üzerinden uzak makineye bağlantı 39

4.3. Senaryo 3: Bir PC üzerinden uzak makinede yazılım güncellemesi 40

5. SONUÇLAR 42

6. KAYNAKLAR 44

ÖZET

ENDÜSTRİDE UZAKTAN İZLEME, KONTROL ve BAKIM YÖNTEMLERİNE ÇÖZÜM ÖNERİSİ OLABİLECEK BİR CİHAZ TASARIMI

Son yıllarda, Nesnelerin İnterneti (IoT) teknolojilerinin hızla gelişmesi ve endüstride evden çalışma modelinin benimsenmesi veya denizaşırı ülkelerden yapılan uzaktan işlerin artması nedeniyle, uzaktan cihaz erişimi ve yönetimi ile uzaktaki cihazlar ile veri alışverişine olan ihtiyaç önemli ölçüde artmıştır. Bu durum, endüstriyel ortamlarda cihazlara erişim ve kontrol yöntemlerini yeniden şekillendirmekte; güvenli ve verimli bir yönetim altyapısının gerekliliğini vurgulamaktadır. Bu tez çalışmasında, Ethernet portu üzerinden bağlanan IoT cihazlarına güvenli ve etkili uzaktan erişim sağlamayı amaçlayan yeni bir sistem geliştirmeye odaklanılmıştır. Önerilen sistemin temel işlevleri, PLC, HMI, DCS vb. cihazlara yazılım yükleme, mevcut yazılımları güncelleme ve veri toplama ve izleme kontrol paneli olarak hizmet verme olarak sayılabilir. Bu işlevler sayesinde, sistem yöneticileri, cihazların yazılım sürümlerini kolayca güncelleyebilir ve bakım yapmalarını sağlayarak, cihazların sürekli güncel ve güvenli kalmasını temin edebilirler.

Sistemin merkezinde, yüksek güvenlik standartlarına sahip modern bir VPN çözümü olan WireGuard ve düşük maliyetli, yüksek performanslı Orange Pi Zero geliştirme kartı bulunmaktadır. WireGuard VPN sunucusunun Orange Pi Zero üzerinde kurulması, uzaktan erişim sırasında yüksek düzeyde veri güvenliği sağlar. Geliştirilen cihazın kablosuz ağının yapılandırılması ve yönetimi için Node-RED üzerinde geliştirilen bir kontrol paneli kullanılmaktadır. Bu web tabanlı kontrol paneli, kullanıcılara Orange Pi Zero'nun bulunduğu yerde gerçek zamanlı WiFi ağlarını tarama, seçilen ağa bağlanma ve gerekli ağ ayarlarını yapılandırma olanağı sunar. Mobil uygulama, Flutter ortamında geliştirilmiştir; ağ yapılandırması ise, mobil cihazda Node-RED Kullanıcı Arayüzü üzerinden gerçekleştirilmiştir.

Bu çalışma, güvenlik ve yönetilebilirlik açısından IoT sistemlerine yenilikçi bir çözüm sunmayı, açık kaynak ve düşük maliyetli teknolojileri kullanarak IoT altyapılarını optimize etme konusunda değerli bilgiler sunmayı amaçlamaktadır. Elde edilen sonuçlar, benzer sistemlerin geliştirilmesi ve uygulanması için önemli bir referans oluşturmakta ve bu alandaki araştırmalara katkıda bulunmayı hedeflemektedir.

Anahtar Kelimeler: Wireguard VPN; Uzak Cihazlara Erişim, VPN Uygulamaları, Endüstriyel IoT, Uzaktan İzleme ve Kontrol, Uç Bilişim

ABSTRACT

A DEVICE DESIGN OFFERING SOLUTIONS for REMOTE MONITORING, CONTROL and MAINTENANCE METHODS IN INDUSTRY

In recent years, the need for remote device access and management, as well as data exchange from remote devices has increased significantly due to the rapid development of Internet of Things (IoT) technologies and the adoption of home-office model in the industry or remote jobs from overseas. This situation is reshaping the methods of access and control of the devices in industrial environments and highlighting the necessitating a secure and efficient management infrastructure. This study is focused on developing an innovative system aiming to provide a secure and effective remote access to IoT devices connected via an Ethernet port. The primary functions of the proposed system are to offer a platform for uploading software to these devices (PLC, HMI, DCS, etc.), updating existing software, and serving as a data acquisition and monitoring dashboard. This functionality enables system administrators to easily update and maintain the software versions of devices, ensuring they remain continuously up-to-date and secure.

At the core of the system is WireGuard, a modern VPN solution with high-security standards and a the low-cost, high-performance Orange Pi Zero development board. The installation of the WireGuard VPN server on the Orange Pi Zero ensures a high level of data security during remote access. For the configuration and management of the wireless network of the developed device, a dashboard developed on Node-RED is employed. This web-based dashboard provides users the ability to scan real-time WiFi networks at the location of the Orange Pi Zero, connect to the selected network, and configure the necessary network settings. The mobile application is developed using the Flutter development environment, and network configuration is performed via the Node-RED User Interface on the mobile device.

This study aims to provide an innovative solution to IoT systems in terms of security and manageability, offering valuable insights into optimizing IoT infrastructures using open-source and low-cost technologies. The results serve as an important reference for the development and implementation of further similar systems and aim to contribute to research in this field.

KeyWords: Wireguard VPN; Remote Device Access, VPN Applications, Industrial IoT, Remote Monitoring and Control, Edge Computing

SEMBOLLER ve BİRİMLER

V: Volt (Gerilim)

C: Santigrat Derece (Sıcaklık Ölçümü)

m/dak: Metre/Dakika (Hız Ölçümü)

Mbps: Megabit per second (Saniyede Megabit)

Gbit/s: Gigabit per second (Saniyede Gigabit)

GB: Gigabyte (Gigabayt)

M: Milyon

KISALTMALAR

IOT: Internet of Things (Nesnelerin İnterneti)

IIOT: Industrial Internet of Things (Endüstriyel Nesnelerin İnterneti)

HMI: Human-Machine Interface (İnsan-Makine Arayüzü)

PLC: Programmable Logic Controller (Programlanabilir Mantık Denetleyici)

JSON: JavaScript Object Notation

VPN: Virtual Private Network (Sanal Özel Ağ)

UI: User Interface (Kullanıcı Arayüzü)

SDK: Software Development Kit (Yazılım Geliştirme Kiti)

CPU: Central Processing Unit (Merkezi İşlem Birimi)

GPIO: General Purpose Input/Output (Genel Amaçlı Giriş/Çıkış)

DC: Direct Current (Doğru Akım)

LED: Light Emitting Diode (Işık Yayan Diyot)

API: Application Programming Interface (Uygulama Programlama Arayüzü)

WiFi: Wireless Fidelity (Kablosuz Bağlantı)

IPsec: Internet Protocol Security (İnternet Protokolü Güvenliği)

L2TP/IPsec: Layer 2 Tunneling Protocol with Internet Protocol Security (İnternet Protokolü Güvenliği ile Katman 2 Tünelleme Protokolü)

PPTP: Point-to-Point Tunneling Protocol (Noktadan Noktaya Tünelleme Protokolü)

SSL: Secure Sockets Layer (Güvenli Yuva Katmanı)

AES-256: Advanced Encryption Standard 256-bit (Gelişmiş Şifreleme Standardı 256-bit)

DPDK: Data Plane Development Kit (Veri Düzlemi Geliştirme Kiti)

IKEv2: Internet Key Exchange version 2 (İnternet Anahtar Değişimi sürüm 2)

ISP: Internet Service Provider (İnternet Servis Sağlayıcı)

MPPE: Microsoft Point-to-Point Encryption (Microsoft Noktadan Noktaya Şifreleme)

PSK: Pre-Shared Key (Ön Paylaşımlı Anahtar)

TCP: Transmission Control Protocol (İletim Kontrol Protokolü)

UDP: User Datagram Protocol (Kullanıcı Datagram Protokolü)

VNF: Virtual Network Function (Sanal Ağ Fonksiyonu)

VPNaaS: VPN as a Service (Hizmet Olarak VPN)

RAM: Random Access Memory (Rastgele Erişimli Bellek)

LAN: Local Area Network (Yerel Alan Ağı)

OEM: Original Equipment Manufacturer (Orijinal Ekipman Üreticisi)

ŞEKİL LİSTESİ

Şekil 1.1 VPN Türleri Arasında Performans Karşılaştırması[26] 4
Şekil 3.1 Sistem Mimarisi 9
Şekil 3.2 Orange Pi Zero 10
Şekil 3.3 Node-RED Uygulaması 15
Şekil 3.4 Node-RED Ağ Ayar Sayfası 17
Şekil 3.5 Wireguard Dashboard Kullanıcı Girişi Sayfası 25
Şekil 3.6 AndroidManifest Konfigürasyonu 26
Şekil 3.7 Pubspec.yaml Konfigürasyonu 28
Şekil 3.8 Dart Programlamada Ana Sayfa 29
Şekil 3.9 Mobil Cihaz Üzerinde Gateway Cihazına Erişim Sayfası 30
Şekil 3.10 Mobil Cihaz ile Node-RED Dashbord Ekranına Erişim 31
Şekil 3.11 Mobil Cihaz Üzerinde Aktif Kullanıcı Sayfasının Gösterimi 32
Şekil 3.12 Ağ Geçidi Cihazın Açık Devre Şeması 33
Şekil 3.13 Orange Pi Zero Kartının Montajı 34
Şekil 3.14 Tasarlanan Cihazın farklı yönlerden görünüşü a)Ön, b)Arka, c)Üst ve d)Alt 35
Şekil 4.1 Gateway Cihazının Boyama Makinesine Bağlantı Şeması 37
Şekil 4.2 Mobil Cihaz ile Kontrolör Paneline Erişim 38
Şekil 4.3 InfluxDB Veritabanında Su Tüketimine Ait Grafik 39
Şekil 4.4 Bilgisayar ile Ram Makinesine Uzaktan Erişim Şeması 39
Şekil 4.5 Grafana Veri Görselleştirilmesi 40
Şekil 4.6 Bilgisayar ile PPD Makinesine Uzaktan Erişim Şeması 40
Şekil 4.7 PPD Makine Plc Yazılımı 41

TABLO, ALGORİTMA VE KONFİGÜRASYON LİSTELERİ

Tablo 1.1 Popüler VPN Türlerinin Karşılaştırılması.....3
Tablo 3.1 Orange Pi Zero ve Diğer Geliştirme Kartlarının Karşılaştırılması.....10
Tablo 5.1 Endüstride Kullanılan Cihazlar ile Bu Cihazın Karşılaştırılması.....43

Konfigürasyon 3.1 Wireguard Sunucu Yapılandırması 11
Konfigürasyon 3.2 Node-RED Sistem Servis Yapılandırması 14
Konfigürasyon 3.3 Nginx Port Proxy Yapılandırması 21
Konfigürasyon 3.4 Wireguard Server Dosya Yapılandırması 22
Konfigürasyon 3.5 Node-RED Sistem Servis Yapılandırması 24

Algoritma 3.1 Wlan Bağlantısı Durum Kontrolü.....15
Algoritma 3.2 Ethernet Bağlantısı Durum Kontrolü 16
Algoritma 3.3 Hotspot Modu Yönetimi 16
Algoritma 3.4 Hotspot Modu Kontrolü 17

1. GİRİŞ

Son yıllarda hızla gelişen nesnelerin interneti (IoT) teknolojileri, uzaktan cihazlara erişim ve yönetimin önemini büyük ölçüde artırmıştır. IoT, günlük yaşamdan endüstriyel uygulamalara kadar geniş bir yelpazede kullanılmakta ve cihazlar arasındaki veri alışverişi ve entegrasyonunu kolaylaştırmaktadır. Bu teknolojik ilerleme, özellikle endüstriyel ortamlarda bulunan cihazlara erişimi ve kontrolü yeniden şekillendirerek güvenli ve etkin bir yönetim altyapısının oluşturulmasını zorunlu kılmaktadır. Bu tez çalışmanın amacı, Ethernet arayüzü üzerinden bağlı IoT cihazlarına uzaktan, güvenli ve efektif bir şekilde erişim sağlayarak, algılayıcılardan toplanan verilerinin izlenmesi, sistem kontrolü ve cihazlara yazılım yüklemek veya mevcut yazılımları güncellemeye izin veren bir sistem geliştirmektir. Bu işlevsellik, sistem yöneticilerine cihazların yazılım sürümlerini kolayca güncelleyebilme ve bakım yapabilme imkânı sunar. Böylece cihazların sürekli güncel ve güvenli kalmasını sağlar. Bu çalışma, Güvenlik, Kullanıcı Dostu, Esneklik ve Maliyet gibi kriterler göz önüne alınarak yapılmıştır.

Bu çalışmanın amacı, endüstriyel IoT uygulamalarında güvenli ve düşük maliyetli uzaktan erişim çözümleri geliştirmektir. Bu doğrultuda, Orange Pi Zero tabanlı bir sistem üzerinde WireGuard kullanılarak, endüstriyel cihazlara güvenli erişim sağlayan bir IoT sistemi tasarlanmıştır. Teknolojik altyapı olarak yüksek güvenlik standartlarına sahip modern bir VPN çözümü olan WireGuard ve düşük maliyetli, yüksek performanslı bir geliştirme kartı olan Orange Pi Zero bulunmaktadır. Kablosuz Ağ Yönetimi için Node-RED üzerinde geliştirilen bir dashboard ekranı kullanılmaktadır.

1.1. Endüstride Uzaktan Erişim Modelleri

Endüstride uzaktan erişim modellerinin önemini giderek artmaktadır ve bu uzaktan erişim modelleri teknolojiye dayalı endüstrilerden yaygınlaşmaya başlayarak diğer sektörlerde de hızla yerini almaya başlamıştır. Uzaktan erişim çözümleri, masaüstü paylaşımı, uzaktan yönetim ve çevrimiçi toplantılar gibi amaçlarla kullanılmaktadır. Bu ihtiyaçları karşılayan yaygın olarak kullanılan bazı uzaktan erişim modelleri şunlardır:

TeamViewer, güvenli ve kullanıcı dostu arayüzüyle, çeşitli sektörlerde birçok uygulamada uzaktan erişim için sıkça tercih edilen bir platformdur [9]. Çalışanlar, müşterilerle etkileşimde bulunurken, teknik destek sağlarken veya sunumlar yaparken TeamViewer'ı kullanarak uzaktan bağlantı kurabilirler.

Yüksek performansı ve kolay kullanımıyla **AnyDesk**, hızlı ve güvenli uzaktan erişim sağlayan bir diğer popüler platformdur ve endüstriyel ihtiyaçları karşılamak için ideal bir çözüm sunmaktadır [10]. Endüstriyel ortamlarda, teknik destek ekipleri veya uzmanlar, AnyDesk'i kullanarak sorun giderme ve bakım işlemlerini uzaktan gerçekleştirebilirler.

Remote Desktop Connection, birçok endüstriyel ortamda kullanılan bir uzaktan erişim modelidir. Büyük ölçekli kurumsal ağlarda, çalışanlar uzak ofislere veya sunuculara erişmek için bu modeli kullanabilirler. Güvenli bir şekilde, kullanıcılar Windows tabanlı sistemlere uzaktan bağlantı kurabilirler [11].

Belirtilen platformlarla endüstriyel uygulamalarda masaüstü paylaşım yapılarak uzaktan bağlantı kurulumu sağlanabilmektedir. Bu platformlar yaygın şekilde kullanılmaya devam edilmektedir.

AnyDesk ve TeamViewer gibi ekran paylaşımlı uzak masaüstü çözümleri, bir bilgisayarı başka bir yerden kontrol etmek için kullanılır. Bu tür uygulamalar, özellikle teknik destek sağlama veya uzaktan eğitim gibi durumlar için idealdir. Ancak, bu tür araçlar genellikle bireysel bilgisayarlar veya küçük gruplar için tasarlanmıştır ve büyük ölçekli kurumsal ağlara erişim sağlamak için yeterli güvenlik ve yönetim özelliklerine sahip değildir. Geniş ölçekli ağ erişim ihtiyaçları için VPN'ler (Virtual Private Network, Sanal Özel Ağ) ortaya çıkmıştır. VPN teknolojisi, uzaktaki çalışanların kurumsal ağlara güvenli bir şekilde erişebilmelerini sağlamak amacıyla geliştirilmiştir. Özellikle birden fazla konumda çalışanları ve kaynakları olan büyük organizasyonlar için VPN'ler iş verilerinin güvenliğini sağlama ve ağ kaynaklarına uzaktan erişim sağlama konusunda kritik bir role sahiptir.
Popüler VPN çeşitleri genellikle kullanım amaçlarına göre temel olarak 2'ye ayrılır. Kurumsal VPN'ler ve kişisel VPN'ler. Her iki kategoride de sık kullanılan bazı popüler VPN çeşitleri aşağıda sıralanmıştır:

PPTP (Point-to-Point Tunneling Protocol) : PPTP VPN 1999 yılında duyurulmuştur. Daha eski bir VPN teknolojisi olup, kolay kurulumu ve geniş cihaz desteği nedeniyle popülerliğini sürdürmektedir ancak güvenlik açısından daha güncel VPN protokollerinden daha düşük olarak kabul edilmektedir [12].

SSL (Secure Sockets Layer): SSL VPN 1995 yılında duyurulmuştur. Web tarayıcıları aracılığıyla güvenli bir erişim sağlar ve genellikle kullanıcıların yalnızca bir web tarayıcısı yoluyla kurumsal uygulamalara erişimine izin verir.

IPsec (Internet Protocol Security) : IPsec VPN 1998 yılında duyurulmuştur. İnternet üzerinden güvenli bir şekilde veri aktarımı sağlamak için tasarlanmıştır. IPsec, veri paketlerini şifreleyerek güvenli bir iletişim kanalı oluşturur.

L2TP/IPsec (Layer 2 Tunneling Protocol with IPsec) : . L2TP/IPsec VPN 1999 yılında duyurulmuştur. L2TP ve IPsec protokollerinin bir kombinasyonudur. Güçlü güvenlik sağlamak için IPsec'in şifreleme yeteneklerini kullanır [12].

OpenVPN : OpenVPN 2001 yılında duyurulmuştur. Açık kaynaklı ve çok yönlü bir VPN çözümüdür. SSL/TLS kullanarak güvenlik sağlar ve hem "site-to-site" hem de "remote access" konfigürasyonları için kullanılabilir [12].

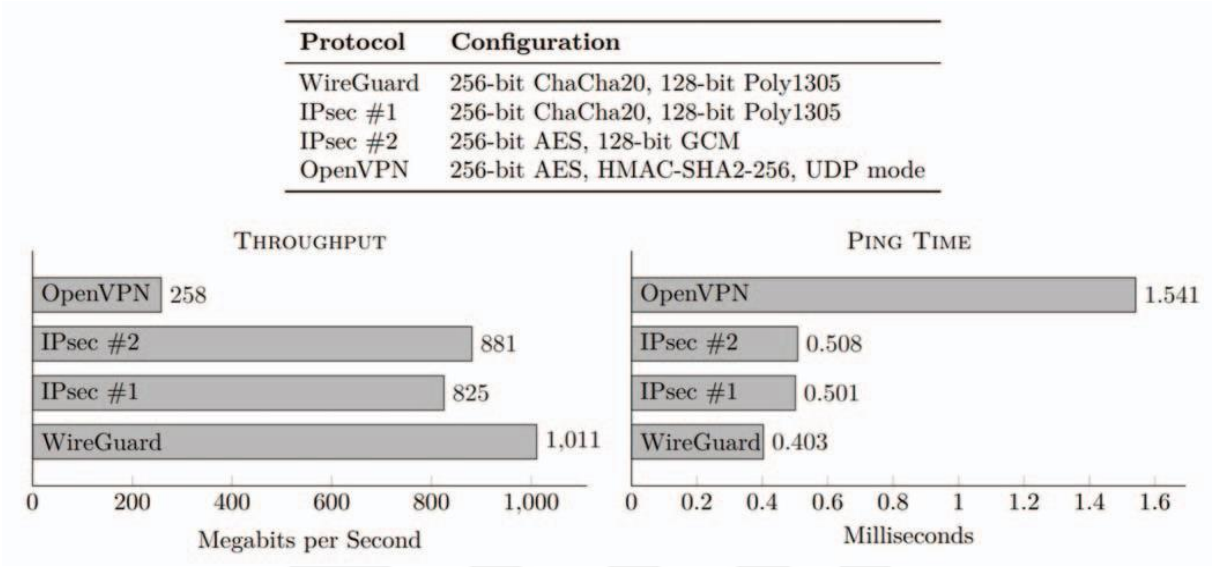
Wireguard: Wireguard 2020 yılında duyurulmuştur. Yeni ve hızlı gelişen bir VPN protokolüdür. Basitlik ve yüksek hızlar sunar ve modern şifreleme tekniklerini kullanır.

Daha önce açıklanan VPN çeşitlerinin birbirlerine göre farklı avantaj/dezavantaj durumları mevcuttur. Tablo 1.1'de VPN türlerinin kıyaslaması verilmiştir[14-18].

Tablo 1.1 Popüler VPN Türlerinin Karşılaştırılması

Parametreler	PPTP	L2TP/IPsec	OpenVPN	IKEv2/IPsec	SSL	WireGuard
Duyurulma Tarihi	1999	1999	2001	2005	1995	2020
Şifreleme Düzeyi	128-bit MPPE	AES-256	AES-256	AES-256	AES-256	ChaCha20
Kimlik Doğrulama	Parola tabanlı	Sertifikalar, PSK	Sertifikalar, PSK	Sertifikalar, PSK	Sertifikalar	Genel Anahtarlar
Veri Bütünlüğü	Zayıf	Güçlü	Güçlü	Güçlü	Güçlü	Güçlü
Bağlantı Hızı	Yüksek	Orta	Yüksek	Yüksek	Yüksek	Çok Yüksek
Gecikme	Düşük	Orta	Orta	Düşük	Orta	Çok Düşük
Bant Genişliği Kullanımı	Düşük	Orta	Orta	Düşük	Orta	Düşük
Kurulum Maliyeti	Düşük	Orta	Orta	Orta	Orta	Düşük
İşletme Maliyeti	Düşük	Orta	Orta	Orta	Orta	Düşük
Ölçeklenebilirlik	Düşük	Orta	Yüksek	Yüksek	Yüksek	Yüksek
Standartlara Uyum	Yok	Orta	Yüksek	Yüksek	Yüksek	Orta
Kurulum Kolaylığı	Kolay	Orta	Karmaşık	Orta	Orta	Kolay
Yönetim ve İzleme	Temel	Orta	Gelişmiş	Gelişmiş	Gelişmiş	Temel
Destek ve Dokümantasyon	Sınırlı	İyi	İyi	İyi	İyi	Sınırlı
Bağlantı Güvenilirliği	Orta	Yüksek	Yüksek	Yüksek	Yüksek	Yüksek

Sık kullanılan VPN türleri arasında OpenVPN ve IPsec öne çıkmaktadır. Bu tez çalışmasında kullanılan Wireguard, bu VPN türleri arasında karşılaştırılarak Şekil 1.1'de gösterilmiştir[26]. Karşılaştırmada, şifreleme türü, verimlilik ve ping gecikme süresi gibi parametreler incelenmiştir.



Şekil 1.1 VPN Türleri Arasında Performans Karşılaştırması[26]

Bu tez çalışmasında, endüstriyel ekipmanların uzaktan izlenebilirliğini ve yönetilebilirliğini artırmayı hedeflemekte olup, açık kaynaklı teknolojiler kullanarak güvenlik, esneklik ve maliyet açısından optimize edilmiş çözümler sunmayı amaçlamaktadır. Bu tez çalışmasının kapsamı şu şekildedir:

1. Bölümde çalışmanın amacı, kapsamı ve gereksinimleri belirtilmiştir. 2. Bölümde literatürde yer alan benzer çalışmalar ve ilgili teknolojiler incelenmiştir. 3. Bölümde sistemin tasarımı ve yazılım geliştirme basamakları hakkında bilgi verilmiştir. 4. Bölümde bu tez çalışmasının da ana amacı doğrultusunda uzaktaki donanımlara erişim senaryolarına yer verilmiştir. 5. Bölümde sistemden elde edilen sonuçlar ve değerlendirmeler belirtilmiştir.

2. BENZER ÇALIŞMALAR

Literatürde endüstriyel cihazların uzaktan yönetimi ve bakımı için çeşitli çalışmalar bulunmaktadır. Bu tez çalışmasının da kapsamına uygun olarak yapılan literatür taraması ,Web of Science veri tabanında gerçekleştirilmiştir. Tarama için "VPN Applications" kuralı oluşturulduğunda 919 çalışma çıkmaktadır. Ancak "Wireguard" kelimesi ile filtreleme yapıldığında sadece 5 tane çalışma kalmaktadır. Bunlar içinde 3 tanesi araştırma makalesi [1,4,13] ve 2 tanesi de konferans bildirisi [3,6] bulunmaktadır. Bu çalışmalara ek olarak bir başka tarama da "Wireguard VPN OR (Remote Device Access AND VPN Applications)" kuralı ile yapılmış ve çıkan sonuçlarda "Sorted by: Citations: highest first" seçimi yapılarak atıf sayısı yüksekten düşüğe doğru sıralanmıştır. Bu filtrelemelerden çıkan çalışmalardan bazıları incelenmiştir.

2.1. Literatürdeki Kaynak Araştırması

Mansouri ve arkadaşları, hibrit bulut ortamlarında dağıtık veri tabanlarının performans değerlendirmesine odaklanmıştır[1]. Bu çalışma, özel ve genel bulut kaynaklarını birleştirerek performansı iyileştirmeyi ve ön yatırım maliyetlerini azaltmayı amaçlayan hibrit bulutların otomatik bir uygulamasını sunar. Çalışmada, özel ve genel bulutlar arasında güvenli bir bağlantı sağlamak için Linux tabanlı bir VPN olan WireGuard ve altyapı kaynaklarını dağıtmak için Terraform kullanılmıştır. WireGuard, kolay kullanımı, esnekliği, güvenliği ve maliyetsiz olması nedeniyle tercih edilmiştir. Performans değerlendirmesi, yerel OpenStack ve Microsoft Azure'u kapsayan hibrit bulut üzerinde altı modern ve dağıtık veritabanı sistemi üzerinde gerçekleştirilmiştir. Sonuçlar, MongoDB ve MySQL Cluster'ın genel buluta kaynaklarını genişlettiklerinde daha verimli çalıştıklarını, buna karşın Cassandra, Riak, Redis ve CouchDB'nin performansının önemli ölçüde azaldığını göstermektedir. Bu çalışma, hibrit bulutların güvenli, esnek ve maliyetsiz bağlantılarla nasıl uygulanabileceğine ve bu ortamda dağıtık veri tabanlarının performansının nasıl değerlendirileceğine dair önemli bulgular sunmaktadır.

Haga ve arkadaşları, 5G ağ dilimi izolasyonu için WireGuard ve Açık Kaynaklı MANO'yu kullanarak VPNaaS (Hizmet Olarak VPN) bir kavram ortaya sunmaktadır[2]. Bu çalışma, 5G ağlarında çeşitli hizmetlerin aynı fiziksel altyapıda barındırılması amacını taşır ve ağ dilimlemeyi bu hedefe ulaşmada kilit bir araç olarak görür. Çalışmada, güvenli ağ tünelleme protokolü olan WireGuard, 5G ağında sanallaştırılmış ağ fonksiyonları (VNF) arasında VPNaaS işlevselliği sağlamak için uygulanmıştır. WireGuard etkin hizmetlerin 4 dakika 26 saniyede başlatılabileceği, önceden kurulmuş ve güncellenmiş WireGuard sürümüne sahip

görüntülerle bu sürenin 2 dakika 44 saniyeye kadar düşebileceği gösterilmiştir. Ayrıca, WireGuard'ın OpenVPN'e kıyasla 5,3 kat daha yüksek ağ verimi ve %41 daha düşük gecikme sağladığı ifade edilmiştir.

Pudelko ve arkadaşları, VPN geçitlerinin performans analizine odaklanmaktadır[3]. Bu çalışmada, OpenVPN, Linux IPsec ve WireGuard gibi farklı yazılım tabanlı VPN geçitlerinin performansları karşılaştırılmaktadır. Çalışma, WireGuard uyumlu bir VPN kıyaslama örnek uygulamasını üç farklı yazılım mimarisi ile geliştirerek bireysel etkileri ve optimizasyonları izole bir şekilde test etmiştir. Sonuçlar, WireGuard'ın mimari açıdan en umut verici VPN geçidi olduğunu göstermektedir. Özellikle DPDK üzerinde WireGuard'ın ardışık düzen mimarisi ile gerçekleştirilen uygulama, 6.2 Mbps ve 40 Gbit/s hızına ulaşarak en hızlı VPN uygulaması olmuştur. Çalışmada çok çekirdekli senkronizasyon ve veri yapılarının VPN'lerinin ölçeklenebilirliğindeki ana sorun olduğu ve ardışık düzen ve mesaj iletimi tabanlı bir mimarinin bu sorunu çözebileceği vurgulanmaktadır. Ayrıca VPN geçitlerinin performansının yüksek hızlı ağlarda nasıl iyileştirilebileceği ve özellikle WireGuard'ın bu alandaki potansiyeli gösterilmektedir.

Kjorveziroski ve arkadaşları, güvenli bir uç-bulut sürekliliği için tam ağ (full-mesh) VPN performans değerlendirmesine odaklanmışlardır[4]. Bu çalışma dağıtık Kubernetes kümeleri arasında güvenli ve verimli iletişim sağlamak amacıyla Headscale, Netbird ve ZeroTier gibi üç farklı tam ağ VPN çözümünü değerlendirmektedir. Çalışmada, her bir VPN çözümünün Kubernetes altyapısı üzerindeki performansı, TCP ve UDP veri aktarım hızı ve Kubernetes API yanıt süreleri açısından test edilmiştir. Sonuçlar, Netbird'ün doğrudan bağlantı senaryolarında en iyi performansı sunduğunu, ancak yüksek gecikme ve paket kaybı gibi olumsuz ağ koşullarında Headscale ve ZeroTier'in daha iyi performans gösterdiğini ortaya koymaktadır. Bu çalışma, tam ağ VPN çözümlerinin güvenli ve verimli bir şekilde uygulanabilirliği ve çeşitli ağ koşullarında performanslarını değerlendirmek için önemli bir rehber sunmaktadır.

Paillisse ve arkadaşları, WireGuard için bir kontrol düzlemi sunmaktadır[5]. WireGuard, basit yapılandırma, mobilite desteği, azaltılmış kod tabanı ve yüksek performans gibi avantajlarıyla dikkat çekmektedir. Ancak, WireGuard’ın kontrol düzlemi bulunmamakta, bu nedenle her bir düğüm manuel olarak diğer düğümlerin genel anahtarları ve IP adresleri ile yapılandırılmalıdır. Bu çalışmada, merkezi bir sunucu kullanılarak bu bilgilerin otomatik olarak dağıtılmasını sağlayan bir mimari önerilmektedir. Bu mimaride, ilk olarak, WireGuard tüneli manuel olarak merkezi sunucuya kurulur ve tüm düğümler genel anahtarlarını ve IP adreslerini bu sunucuda saklar. Düğümler, iletişim kurmak istedikleri diğer düğümlerin bilgilerini bu güvenli kanal

üzerinden talep ederler. Bu yaklaşım, Public Key Infrastructure (PKI) tabanlı anahtar dağıtım şemalarına göre daha basit bir çözüm sunmayı, düğümlere gönderilen genel anahtar sayısını sınırlamayı ve User Datagram Protocol (UDP) tabanlı bir protokol ile tünel kurulumu gecikmesini azaltmayı amaçlamaktadır. WireGuard'ın kontrol düzlemi olmadan kurulumunun zaman alıcı ve hata yapmaya açık olduğu göz önünde bulundurulduğunda, bu otomasyonun kurumsal veya Internet Service Provider (ISP) senaryolarında dağıtımı kolaylaştırabileceği belirtilmektedir. Ayrıca, çalışmada kontrol düzleminin tasarımı, performans metriklerinin analizi ve uygulanabilecek olası iyileştirmeler detaylandırılmaktadır.

Mackey ve arkadaşları, iki popüler VPN protokolü olan WireGuard ve OpenVPN'in performansını karşılaştırmaktadır[6]. WireGuard, hafif yapısı ve basit kurulumu ile dikkat çekmekte olup, çok çekirdekli makinelerde çoklu iş parçacığı desteği sayesinde OpenVPN'e kıyasla belirgin bir performans avantajı sağlar. Yaklaşık 4.000 satırlık bir kod tabanına sahip olan WireGuard, OpenVPN'in 60.000 satırlık karmaşık yapısına kıyasla daha hızlı kurulur ve daha az kaynak tüketir. Çalışmada, Amazon Web Services (AWS) ve yerel sanal makineler üzerinde yapılan testlerde, özellikle TCP bağlantılarında WireGuard'ın sürekli olarak OpenVPN'den daha iyi performans gösterdiği gözlemlenmiştir. Gelecek çalışmalarda, farklı platformlarda daha fazla test yapılması ve WireGuard'ın tek çekirdekli sistemlerdeki üstün performansının nedenlerinin araştırılması önerilmektedir. Bu bulgular, WireGuard'ın OpenVPN'e karşı önemli bir rakip haline geldiğini ve gelecekte daha yaygın olarak kullanılabileceğini göstermektedir.

Wu ve arkadaşları, Android platformunda güvenli anahtar depolama ve dinamik yetkilendirme mekanizmaları ile geliştirilmiş bir WireGuard versiyonu olan SeWG'yi tanıtmaktadır[7]. Bu çalışmada, TEE (Trusted Execution Environment) teknolojisi kullanılarak anahtarların güvenli bir şekilde saklanması ve kötü amaçlı yazılımların VPN tünelini kötüye kullanmasının önlenmesi amaçlanmıştır. Çalışmada, Qualcomm'un QSEE platformu üzerinde SeWG'nin uygulanabilirliği gösterilmiştir. Deneysel sonuçlar, SeWG'nin orijinal WireGuard'a kıyasla yüksek verimlilikle çalıştığını ve herhangi bir verimlilik kaybı olmadan güvenlik iyileştirmeleri sağladığını göstermiştir.

Yao ve arkadaşları, akıllı şebeke sistemlerinde kullanıcı verilerinin gizliliğini koruma ve güvenli veri iletimi sağlama gerekliliklerini ele almaktadır[13]. Önerilen sistem, Hyperledger Fabric ve WireGuard teknolojilerini entegre ederek dağıtık ve güvenli bir enerji ticaret platformu tasarlamaktadır. Hyperledger Fabric, enerji ticareti ve kullanıcı verilerini blok zincir üzerinde güvenli bir şekilde saklarken; WireGuard ise cihazlar arası güvenli veri aktarımı

sağlar. Çalışmada geliştirilen sistemde, kullanıcıların enerji tüketim ve üretim verilerini görüntüleyebilecekleri bir web uygulaması sunulmuş, akıllı cihaz simülatörleri ile veri görselleştirme ve performans değerlendirmeleri gerçekleştirilmiştir. Deney sonuçları, WireGuard'ın ağ altyapısına eklenmesinin bant genişliği ve gecikme açısından çok az kayba neden olduğunu, ancak güvenli bir iletişim sağladığını göstermiştir. Hyperledger Fabric ise akıllı şebeke sisteminde işlem tutarlılığı ve izlenebilirlik sağlamaktadır.

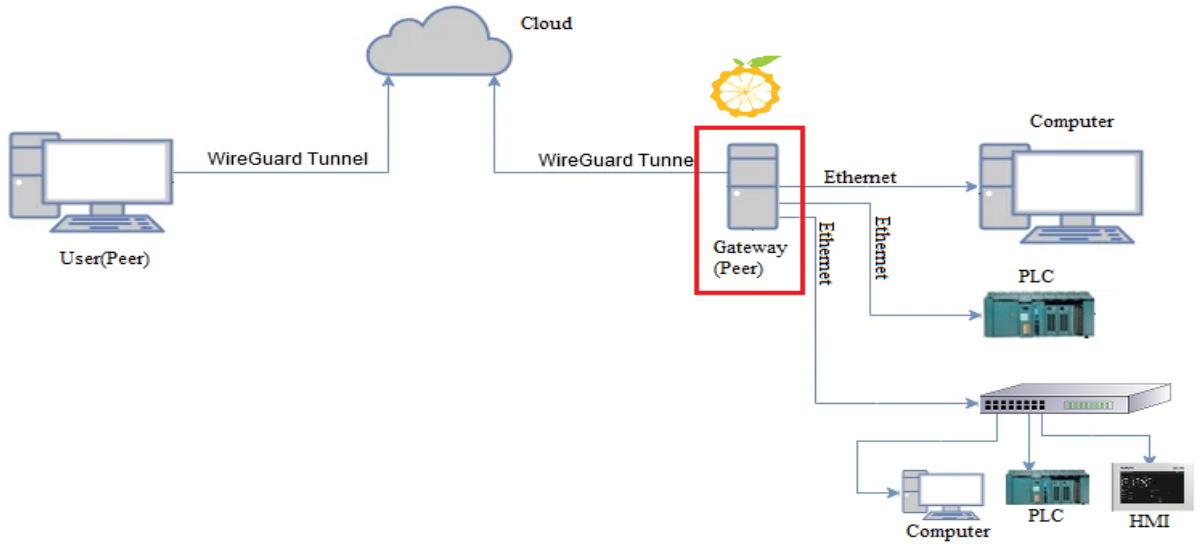
Geliştirilen uygulama, VPN teknolojilerinin endüstrinin uzaktan erişim ihtiyaçlarına nasıl cevap verdiğini ve mobil uygulamalar aracılığıyla kullanıcı dostu sistemler sağlamanın önemini vurgulamaktadır. VPN teknolojileri, özellikle endüstriyel sektörde uzaktan erişim sağlarken güvenlik ve gizliliği garantilemek için kritik öneme sahiptir. Bu teknolojiler, uzaktan çalışan mühendislerin veya operatörlerin fabrika ağlarına güvenli bir şekilde bağlanmalarını sağlar, böylece üretim süreçlerini kesintisiz bir şekilde izlemelerine olanak tanır. Endüstriyel VPN çözümleri, ağ geçitleri üzerinden şifreli kanallar kurarak veri sızıntılarına ve siber saldırılara karşı koruma sağlar. Bu sayede, hassas üretim verileri korunurken, sistemler arası etkileşim ve veri aktarımı güven içinde gerçekleştirilebilir.

Bu çalışma, özellikle günümüzün hızla dijitalleşen endüstriyel ortamlarında, uzaktan erişim çözümlerinin vazgeçilmezliğini ve bu çözümlerin kullanıcı deneyimini nasıl iyileştirdiğini vurgulamak için önemlidir. Bu bağlamda, bu tez çalışmasında geliştirilen hem donanım hem de yazılım içeren sistem, endüstriyel otomasyon ve IoT sistemleri için güvenli, etkili ve kullanıcı dostu bir uzaktan erişim çözümü sunarak literatüre önemli bir katkı sağlamaktadır.

3. AĞ GEÇİDİ (GATEWAY) CİHAZI GELİŞTİRME AŞAMALARI

Bu bölümde, geliştirilen gateway cihazının tasarım ve uygulama süreçleri adım adım ele alınmıştır. Bu süreçler, cihazın donanım seçiminden yazılım geliştirmeye kadar tüm aşamaları kapsamaktadır.

Sistemin temelinde Şekil 3.1'de de gösterilen yüksek güvenlik standartlarına sahip modern bir VPN çözümü olan WireGuard ve düşük maliyetli, yüksek performanslı bir geliştirme kartı olan Orange Pi Zero bulunmaktadır. Orange Pi Zero'ya WireGuard VPN sunucusu kurulmuş olup, uzaktan erişim sırasında veri güvenliği üst düzeyde sağlanmıştır. VPN üzerinden şifrelenmiş kanallar, yetkisiz erişim ve veri ihlallerine karşı etkili bir koruma sağlamaktadır.



Şekil 3.1 Sistem Mimarisi

3.1. Donanım Seçimi ve Kurulumu

Ağ geçidi cihazının performansı ve güvenilirliği, büyük ölçüde kullanılan donanım bileşenlerine bağlıdır. Bu çalışmada, düşük maliyetli ve yüksek performanslı bir çözüm olan Orange Pi Zero kullanılmaktadır.

Şekil 3.2'de gösterilen Orange Pi Zero, IoT ve gömülü sistemler için ideal olan kompakt, düşük maliyetli ve yüksek performanslı bir geliştirme kartıdır. Orange Pi Zero'nun temel özellikleri aşağıda sıralanmıştır:

- CPU: H2 Quad-core Cortex-A7 H.265/HEVC 1080P
- RAM: 512MB DDR3 (256MB seçeneği de mevcut)
- Depolama: microSD kart yuvası
- Ağ: 10/100 Ethernet, WiFi 802.11 b/g/n
- Bağlantılar: 26 pin GPIO header, USB 2.0, IR alıcı



Şekil 3.2 Orange Pi Zero

Orange Pi Zero'nun bu özellikleri, bir IoT çözümünü esnek hale getirmektedir. Bununla birlikte, benzer geliştirme kartlarıyla karşılaştırıldığında, farklı avantaj ve dezavantajları bulunmaktadır. Tablo 3.1'de Orange Pi Zero'nun farklı kartlarla karşılaştırılması gösterilmiştir.

Tablo 3.1 Orange Pi Zero ve Diğer Geliştirme Kartlarının Karşılaştırılması

Özellik	**Orange Pi Zero**	**Raspberry Pi Zero W**	**Banana Pi M2 Zero**
CPU	H2 Quad-core Cortex-A7	Broadcom BCM2835 ARM11	Allwinner H2+ Quad-core A7
RAM	512MB DDR3	512MB LPDDR2	512MB DDR3
Depolama	microSD	microSD	microSD
Ağ	10/100 Ethernet, WiFi 802.11 b/g/n	WiFi 802.11 b/g/n, Bluetooth 4.1	WiFi 802.11 b/g/n
Bağlantılar	26 pin GPIO, USB 2.0, IR	40 pin GPIO, mini HDMI, USB OTG	40 pin GPIO, USB OTG
Fiyat	10-15 USD	10 USD	15 USD
Güç Tüketimi	Düşük	Çok Düşük	Düşük

Orange Pi Zero kurulum aşamaları şu şekildedir:

İşletim Sistemi Kurulumu: Orange Pi Zero'ya Armbian işletim sistemi kurulmuştur. Armbian, Orange Pi serisi için optimize edilmiş olup, stabil ve performanslı bir çalışma ortamı sunar.

Sistem Güncellemeleri: Kartın yazılım ve sistem güncellemeleri yapılmıştır. Bu adım, sistemin en güncel ve güvenli sürümde çalışmasını sağlar.

Gerekli Bağımlılıkların Kurulumu: WireGuard VPN, Node-RED, Iptables ve diğer gerekli yazılım bağımlılıkları kurulmuştur. Bu bağımlılıklar, sistemin düzgün çalışmasını sağlayacak kütüphaneler ve araçlardır.

Wireguard'ın Kurulması : WireGuard, modern, hızlı ve güvenli bir VPN çözümüdür. Orange Pi Zero üzerinde WireGuard kurulumu ve yapılandırması, gateway cihazının güvenli ve şifrelenmiş bir ağ bağlantısı kurabilmesi için kritik bir adımdır. Aşağıda, WireGuard VPN, Orange Pi Zero'ya aşağıdaki komutlar kullanılarak kurulmuştur:

- sudo apt update
- sudo apt install wireguard

Bu komutlar, sistem paketlerinin listesini günceller ve WireGuard VPN yazılımını kurar.

Kernel Modülünün Yüklenmesi : WireGuard'ın çalışabilmesi için gerekli kernel modülleri aşağıdaki komut ile yüklenmiştir:

- sudo modprobe wireguard

VPN yapılandırma dosyaları oluşturma aşamaları şu şekildedir:

Anahtar Çiftlerinin Oluşturulması: WireGuard, her uç için bir genel ve özel anahtar çifti kullanır. Anahtarlar aşağıdaki komutlar ile oluşturulmuştur.

- wg genkey | tee privatekey | wg pubkey > publickey

Buradaki komutlarımızdan 'wg genkey' özel anahtarı oluşturur. 'tee privatekey' komutu özel anahtarı bir dosyaya kaydeder. 'wg pubkey' komutu ise özel anahtardan genel anahtarı türetir ve bir dosyaya kaydeder.

Sunucu Yapılandırma Dosyasının Yazılması: Sunucu tarafında, WireGuard VPN yapılandırma dosyası '/etc/wireguard/eliargateway.conf' olarak oluşturulmuştur. Dosya içeriği Konfigürasyon 3.1'deki gibidir:

```
[Interface]
PrivateKey = PrivateKey
ListenPort = 445
Address = 101.115.117.11/32

[Peer]
PublicKey = PublicKey
PresharedKey = PresharedKey
AllowedIPs = 101.115.117.0/24
Endpoint = Statik Ip : Port
PersistentKeepalive = 25
```

Konfigürasyon 3.1 Wireguard Sunucu Yapılandırması

Yukarıdaki konfigürasyon ağ geçidi cihazının kullanıcı bilgilerini içermektedir. Privatekey adı altında kullanıcı için özel anahtar mevcuttur. ListenPort parametresi oluşturulacak sanal tünelin paketleri dinleyeceği port belirlemesini sağlar. Bu konfigürasyonda 445 portundan dinleme yapmaktadır. Bu parametreyi boş bırakırsak wireguard kendi otomatik port ataması yapmaktadır. Tünel aktif iken port numarası sıklıkla değişmektedir. Bu da bazı durumlarda bağlantı problemi olarak karşımıza çıkmaktadır. Address kısmındaki ip adresi oluşturulan sanal tünelin ipv4 türünde ip adresini almaktadır. Bu kısımda ipv6 türü de destekleniyor olsa da ipv4 daha sıklıkla kullanılmasından dolayı tercih edilmiştir.
Peer kısmı ise gateway cihazın bağlantı kuracağı wireguard sunucu bilgilerini içermektedir. Buradaki PublicKey ve PresharedKey sunucu kısmının bilgilerini içermektedir. AllowedIPs kısmındaki 101.115.117.0/24 adresi sunucu tarafındaki 101.115.117.1-254 ip adresi aralığındaki kullanıcıların bu cihaza erişebileceği anlamına gelmektedir. Wireguard VPN de özellikle NAT arkasındaki cihazlara erişim sağlama durumunda bazen timeout durumu gerçekleşip bağlantıyı koparmaktadır. Buna çözüm olarak da *PersistentKeepalive* parametresindeki 25 değeri ile 25 saniye periyodunda sunucuya *keepalive* paketleri atarak bağlantı kopmasının engellenmesi amaçlanmıştır. Gateway cihaz kullanıcısının Wireguard sunucuya ulaşım sağlayacağı Statik IP adresi ya da public olarak erişilen domain girilen *Endpoint* parametresi bulunmaktadır. Burada cloud üzerinde bulunan sunucunun Statik IP adresi ve port bilgileri girilmektedir.

WireGuard VPN servisinin başlatılması şu şekildedir:

Sunucu ve istemci yapılandırma dosyaları oluşturulduktan sonra WireGuard servisi aşağıdaki komut ile başlatılmıştır:

- sudo wg-quick up eliargateway

Sistemin başlangıçta çalışması için Wireguard servisinin aşağıdaki komut ile ayarlaması yapılmıştır:

- sudo systemctl enable wg-quick@eliargateway

Bu adımlar, WireGuard VPN'in kurulumu ve yapılandırılmasını içermektedir. Wireguard VPN yazılımının orange pi zero geliştirme kartının üzerinde açılışta çalışacak şekilde ayarlamaları göstermektedir.

IP Yönlendirmesi : Ethernet soketin veya wifi ağına bağlı networklerin içerisindeki ağları Wireguard VPN tüneline bağlanması istenmektedir. Iptables aracılığıyla oluşturulan yönlendirmelerin aktifleşmesi için işletim sisteminde buna izin verilmesi gerekmektedir. Bu izin işlemi için /etc/sysctl.conf dosyasının içerisine aşağıdaki satır eklenmiştir [19].

- net.ipv4.ip_forward=1

Bu satır işlemi eklendikten sonra cihaz yeniden başladığında aktif olacaktır. Cihaz yeniden başlamadan aktifleştirmek için aşağıdaki kodu terminalde yazılabilir.

- sudo sysctl -w net.ipv4.ip_forward=1

3.2. Node-RED Uygulamasının Kurulumu ve Yapılandırması

Node-RED, görsel bir programlama arayüzü sunarak kullanıcıların çeşitli donanım cihazları, API'ler ve çevrimiçi hizmetler ile kolayca entegrasyon kurmasını sağlar. Bu çalışmada, Orange Pi Zero üzerinde Node-RED kullanarak ağ yönetimi ve diğer fonksiyonları sağlayan bir dashboard ekranı geliştirilmiştir. Aşağıda, Node-RED'in kurulumu ve yapılandırma adımları detaylı olarak açıklanmaktadır. Node-RED'in Orange Pi Zero üzerinde kurulumu aşağıdaki adımlarla gerçekleştirilmiştir:

Node.js ve npm Kurulumu: Node-RED, Node.js ve npm (Node Package Manager) üzerine kuruludur. İlk olarak, Node.js ve npm kurulumu gerçekleştirilmiştir.

- sudo apt update
- sudo apt install -y nodejs npm

Node-RED Kurulumu: Node.js ve npm kurulduktan sonra Node-RED, aşağıdaki komut ile global olarak kurulum işlemi yapılmaktadır.

- sudo npm install -g --unsafe-perm node-red

Node-RED'i Başlatma: Node-RED kurulum işlemi bittikten sonra terminale Node-RED yazarak çalıştırma işlemi yapılmaktadır. Node-RED varsayılan port numarası 1880 üzerinden çalışmaktadır. Tarayıcı üzerinden çalışan platforma erişim için 'http://<Ip_Adresi>:1880' ile erişim sağlanabilmektedir. IP adresi kısmına çalıştığı sistemin IP adresi girilmesi gerekmektedir.

Node-RED Servisinin Otomatik Başlatılması: Node-RED'in sistem yeniden başlatıldığında otomatik olarak başlaması için systemd servisi Konfigürasyon 3.2'deki gibi oluşturulmuştur [20].

```
[Unit]
Description=Node-RED
After=network.target

[Service] ExecStart=/usr/bin/env node-red
User=root
Group=root
WorkingDirectory=/root
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Konfigürasyon 3.2 Node-RED Sistem Servis Yapılandırması

Aşağıdaki komutları terminalde sırayla çalıştırarak servisi aktifleştirip Node-RED uygulamasını çalışması sağlanır.

- sudo systemctl daemon-reload
- sudo systemctl enable node-red
- sudo systemctl start node-red

Node-RED Modüllerinin Eklenmesi: Node-RED default yüklü uygulamalar ile çalışmaya başlayacaktır. Uygulama için modül yükleme işlemi 2 farklı şekilde yapılabilmektedir:

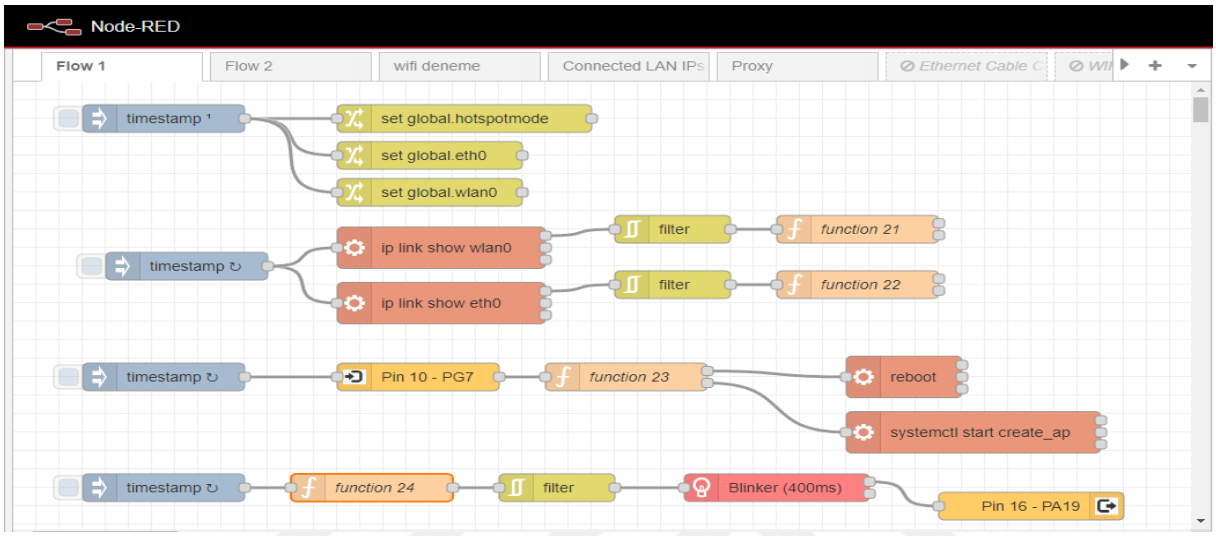
1. Web tarayıcı üzerinden açılan ekrandan sağ üst menü butonuyla *manage palette* sayfasını açarak istediğimiz modülleri aratıp kurma işlemi yapılmasına olanak sağlar.
2. İşletim sisteminde Node-RED kurulu olduğu dizine gelip kurmak istenilen modülleri *'nmp install modül-ismi'* koduyla kurulabilmektedir.

Bu tez çalışmasında kullanılan modüllerden bazıları şunlardır:

- npm install node-red-dashboard
- npm install node-red-contrib-os
- npm install node-red-contrib-opi-gpio
- npm install node-red-contrib-cpu

Geliştirilen cihazda ağ yapılandırması ve ilgili ayarların gerçekleştirilmesi gerekmektedir. Bu görevleri olay tabanlı bir yaklaşımla yürütmek amacıyla Node-RED platformunda yazılım

geliştirilmiştir. Geliştirilen bu yazılım, kullanıcıların kablosuz ağ ayarlarını yapılandırabilmesine, cihaz üzerinden erişilen IP adreslerine yönelik proxy ayarlarını yapmasına ve Ethernet portu üzerinden bağlanan cihazların IP adreslerini kullanıcıya aktarabilmesine imkan tanır. Şekil 3.3'de görüldüğü üzere, Node-RED uygulaması bu yapılandırmalar için bir arayüz sunmaktadır.



Şekil 3.3 Node-RED Uygulaması

Algoritma 3.1'de gösterilen Fonksiyon 21, wlan0 arayüzünün durumunu "Aktif" mi yoksa "Pasif" mi olduğunu belirlemek için kontrol eder. Duruma bağlı olarak, genel bir değişken ayarlar ve karşılık gelen eylemi tetikler.

```
const output = msg.payload;
let status = output.includes("state UP") ? 1 : 0;

if (status === 1) {
  msg.payload = status;
  global.set('wlan0',1);
  return [msg, null]; }
else {
  global.set('wlan0', 0);
  return [null, msg]; }
```

Algoritma 3.1 Wlan Bağlantısı Durum Kontrolü

Algoritma 3.2'de gösterilen Fonksiyon 22, eth0 arayüzünün durumunu "Aktif" mi yoksa "Pasif" mi olduğunu belirlemek için kontrol eder. Duruma bağlı olarak, genel bir değişken ayarlar ve karşılık gelen eylemi tetikler.

```
const output = msg.payload;
let status = output.includes("state UP") ? 1 : 0;
if (status === 1) {
  msg.payload = status;
  global.set('eth0',1);
  return [msg, null]; }
else {
  global.set('eth0', 0);
  return [null, msg]; }
```

Algoritma 3.2 Ethernet Bağlantısı Durum Kontrolü

Algoritma 3.3'de gösterilen Fonksiyon 23, cihaz üzerindeki butonun basılı kalma süresine bakarak işlem yapar. Butona basılma süresi 3-10 saniye aralığında basılı kalırsa cihaz yeniden başlatma görevi görecektir. 10 saniye ve üzeri basılma durumunda cihaz hotspot moduna geçecektir. Hotspot moduna geçme aşamasında wifi bağlantısı var ise bağlantısını kesip wlan arayüzü üzerinden ağ paylaşımı yapacaktır.

```
let lastState = flow.get('lastState') || 0;
let startTime = flow.get('startTime') || Date.now();
let hotspotmode = global.get('hotspotmode ');
if (lastState === 1 && msg.payload === 0)
{   let endTime = Date.now();
    let elapsedTime = endTime- startTime;
    flow.set('lastState', msg.payload);
    flow.set('startTime', Date.now());

    if (elapsedTime >= 3000 && elapsedTime <= 10000){
        return [msg, null];
   } else if (elapsedTime > 10000) {
        hotspotmode = 1;
        global.set('hotspotmode', hotspotmode);
        return [null, msg]; }
   } else if (msg.payload === 1 && lastState === 0){
        startTime = Date.now(); }
flow.set('lastState', msg.payload);
flow.set('startTime', startTime);
return [null, null];
```

Algoritma 3.3 Hotspot Modu Yönetimi

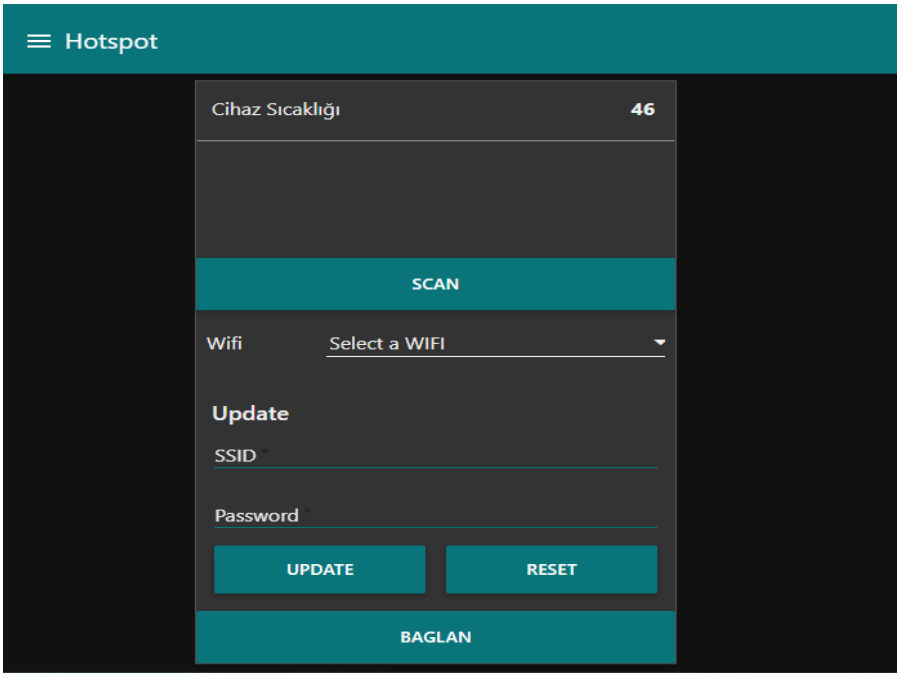
Algoritma 3.4'de gösterilen Fonksiyon 24, hotspot modunun aktif olup olmadığını kontrol eder. Hotspot modu aktif olma durumunda blink fonksiyonuyla pin 16 çıkışına çıkış vererek yeşil ledin flaşör şekilde yanmasını sağlayacaktır. Son kullanıcı, yeşil led aracılığıyla cihazın hotspot moduna geçtiğini anlayacaktır.

```
let hotspotmode = global.get('hotspotmode ');

if (hotspotmode === 1) {
   msg.payload = true; }
else if (hotspotmode === 0) {
   msg.payload = false; }
return msg;
```

Algoritma 3.4 Hotspot Modu Kontrolü

Önerilen sistemde, cihazın kablosuz ağ konfigürasyonları ve yönetimi için Node-RED üzerinde geliştirilen bir dashboard kullanılmaktadır. Şekil 3.4'de gösterilen bu web tabanlı dashboard, kullanıcılara Orange Pi Zero kartının bulunduğu yerdeki WiFi ağlarını gerçek zamanlı olarak tarayabilme, seçilen ağa bağlanabilme ve gerekli ağ ayarlarını yapabilme imkânı sunmaktadır. Kullanıcı arayüzü, teknik bilgileri olmayan son kullanıcıların dahi kolaylıkla etkileşimde bulunabilmesi için sade ve anlaşılır bir tasarıma sahiptir. Ağ geçidi cihazın anlık sıcaklığı ekranda verilmektedir.



Şekil 3.4 Node-RED Ağ Ayar Sayfası

3.3. Iptables Yapılandırması

Iptables, Linux çekirdeği üzerinde çalışan ve ağ paketlerinin filtrelenmesi için kullanılan bir güvenlik duvarı uygulamasıdır. Iptables, ağ trafiğini kontrol etmeyi ve istenmeyen erişimleri engellemeyi sağlar. Node-RED uygulaması için Iptables yapılandırması, güvenlik ve erişim kontrolleri açısından önemlidir.

Iptables Kurulumu: Çoğu Linux dağıtımında Iptables varsayılan olarak kurulu gelir. Ancak, Iptables'in kurulu olup olmadığını kontrol etmek ve kurmak için aşağıdaki komutlar kullanılır.

- sudo apt update
- sudo apt install iptables

Iptables Temel Komutları: Iptables, çeşitli zincirler (chains) ve kurallar (rules) kullanarak paket filtrelemesi yapar. Temel zincirler Input, Output ve Forward'dır [21]:

- Input: Gelen trafiği filtreler.
- Output: Giden trafiği filtreler.
- Forward: Yönlendirilen trafiği filtreler.

NAT ve MASQUERADE Kullanımı: NAT (Network Address Translation), iç ağdaki cihazların dış ağ (internet) ile iletişim kurmasını sağlamak için kullanılır. MASQUERADE ise dinamik IP adreslerine sahip ağ bağlantıları için kullanılır.

- **NAT (Network Address Translation):** NAT, iç ağdaki IP adreslerini dış ağa çevirir. Bu, ağdaki tüm cihazların tek bir dış IP adresi kullanarak internete çıkmasını sağlar. NAT yapılandırması için Iptables'ta aşağıdaki komutlar kullanılabilir:

Örn; sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

Bu komut, eth0 arayüzü üzerinden çıkan tüm trafiği NAT kullanarak maskeleyerek yönlendirir.

- **MASQUERADE:** MASQUERADE, dinamik IP adreslerine sahip ağ bağlantıları için NAT'ın özel bir türüdür. MASQUERADE kullanarak, iç ağdaki cihazların internete çıkması sağlanabilir. MASQUERADE yapılandırması için aşağıdaki komut kullanılabilir:

Örn; sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

Bu komut, eth0 arayüzü üzerinden çıkan tüm trafiği maskeleyerek yönlendirir. MASQUERADE, dinamik IP adreslerinde kullanıldığı için IP adresi değişse bile çalışmaya devam eder.

PostUP ve PostDown Komutları: WireGuard yapılandırmasında, bağlantı kurulurken veya sonlandırılırken çalıştırılacak komutlar için PostUP ve PostDown direktifleri kullanılır. Bu komutlar, belirli ağ ayarlarını yapılandırmak için kullanışlıdır.

- **PostUP:** VPN bağlantısı kurulduktan sonra çalıştırılan komutları belirtir. Örneğin, NAT ve MASQUERADE ayarlarını yapmak için aşağıdaki komut kullanılabilir:

PostUP = iptables -A FORWARD -i eth0 -o eliarvpn -j ACCEPT; iptables -A FORWARD -i eliarvpn -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

- **PostDown:** VPN bağlantısı sonlandırıldıktan sonra çalıştırılan komutları belirtir. NAT ve MASQUERADE ayarlarını geri almak için aşağıdaki komut kullanılabilir:

PostDown = iptables -D FORWARD -i eth0 -o eliarvpn -j ACCEPT; iptables -D FORWARD -i eliarvpn -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

Önceki sayfada verilen PostUp ve PostDown kısmında bulunan komutlar şu anlamlara gelmektedir:

iptables -A FORWARD -i eth0 -o eliarvpn -j ACCEPT: eth0 arayüzünden gelen tüm ağ trafiğini eliarvpn arayüzüne iletir.

iptables -A FORWARD -i eliarvpn -j ACCEPT: eliarvpn üzerinden gelen tüm trafiğe izin verir.

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE: eth0 arayüzünden çıkan tüm trafiği, kaynağın IP adresini gizlemek ve tek bir dış IP adresi üzerinden internete erişimini sağlamak için maskeleme işlemi yapar.

3.4. Hotspot Kurulumu

Bu bölümde, Orange Pi Zero üzerinde create_ap kullanarak bir WiFi hotspot'un nasıl kurulacağı ve yapılandırılacağı anlatılmaktadır. *Create_ap*, kullanıcıların cihazlarını kolayca bir WiFi erişim noktası (hotspot) olarak yapılandırmasına olanak tanıyan güçlü bir araçtır[25].

Gerekli Paketlerin Kurulumu : İlk olarak, create_ap'nin çalışabilmesi için gerekli olan bağımlılıkları kurmamız gerekmektedir. Aşağıdaki komutları kullanarak gerekli paketler yüklenir.

- sudo apt update
- sudo apt-get install git hostapd dnsmasq

Create_ap Araçlarının Kurulması : Hotspot görevi olan create_ap'yi kurmak için aşağıdaki komutları kullanarak git ile ilgili dosyaları klonlanır ve klonlanan dosya içerisine girerek yüklemesi yapılır.

- git clone https://github.com/oblique/create_ap
- create_ap
- sudo make install

Hotspot Konfigürasyonunun Yapılması : Create_ap komutunu kullanarak WiFi adaptörü ve ethernet adaptörü kullanarak ağ paylaşımı yapılabilmektedir. Aşağıdaki komut kullanılarak hotspot başlatılır.

- sudo create_ap -n wlan0 HotspotName HotspotPassword

Yukarıdaki komuta HotspotName ve HotspotPassword yerlerine istenilen ağ ismi ve şifresi koyularak orange pi zero cihazı ağ paylaşımını yapmaktadır. Hotspot kontrolü Node-RED platformunda yapılacak olmasından dolayı systemd servisi otomatik başlayacak şekilde konfigüre edilmemiştir.

3.5. Nginx Proxy Kurulumu ve Stream Yapılandırması

Bu bölümde, Orange Pi Zero üzerinde Nginx web sunucusunun nasıl kurulacağını ve yapılandırılacağını detaylı olarak anlatılacaktır. Nginx, yüksek performanslı, hafif ve açık kaynaklı bir web sunucusudur. Bu çalışmada, Nginx, Node-RED ve WireGuard VPN arasında bir proxy olarak kullanılacaktır.

İlk olarak, Nginx'i yüklemek için gerekli paketlerin kurulması gerekmektedir. Aşağıdaki komutları kullanarak gerekli paketler yüklenir:

- sudo apt update
- sudo apt-get install nginx

Nginx kurulumundan sonra, servisi başlatmak ve sistem yeniden başlatıldığında otomatik olarak başlamasını sağlamak için aşağıdaki komutları kullanılır:

- sudo systemctl start nginx
- sudo systemctl enable node-red

Nginx stream yapılandırması şu şekildedir:

Nginx, TCP ve UDP trafiğini yönlendirmek için stream modülünü kullanır. Bu modül, özellikle VPN ve diğer TCP/UDP tabanlı uygulamaların yönlendirilmesi için kullanışlıdır. Bu bölümde, stream modülünü kullanarak port üzerinden gelen trafiğin yönlendirilmesi gösterilmektedir.

Nginx Stream Konfigürasyon Dosyasının Düzenlenmesi : Nginx Stream konfigürasyonu için nginx aracın kurulu olduğu '/etc/nginx' klasöründe bulunan 'nginx.conf' dosyası açılıp aşağıdaki şekilde strema bloğu eklenir.

Konfigürasyon 3.3'de eklenen stream bloğu orange pi zero kartın networkünde bulunan 3000 portu ile 192.168.40.101 ip adresinin 3000 portuna proxy yapılması sağlanmıştır. Bu işlem ile grafana çalışan bir sisteme erişim amaçlanmıştır.

```
stream {
        server {
                listen 3000;
                proxy_pass 192.168.40.101:3000;
                }
```

Konfigürasyon 3.3 Nginx Port Proxy Yapılandırması

3.6. Wireguard Sunucu ve Arayüz Ekranı

Bu bölümde, Debian Bullseye işletim sistemi içeren bir cloud sunucusuna WireGuard VPN sunucusunun kurulumu ve yönetimi için WGDashboard arayüz ekranının kurulumu anlatılmaktadır[22].

Wireguard Kurulumu: Sunucunun bulunduğu işletim sisteminde Bölüm 3.2'de verilen işlemler ile wireguard kurulumu ve diğer kurulumlar yapılmalıdır. Bu işlemler sonucunda '/etc/wireguard/eliarvpn.conf' isimli dosyanın içeriği Konfigürasyon 3.4'deki gibi olacaktır.

```
[Interface]
Address = 101.115.117.1/32
ListenPort = 25000
PrivateKey = SunucuPrivateKey

PostUp = iptables -A FORWARD -i eth0 -o eliarvpn -j ACCEPT; iptables -A FORWARD -i eliarvpn -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

PostDown = iptables -D FORWARD -i eth0 -o eliarvpn -j ACCEPT; iptables -D FORWARD -i eliarvpn -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

[Peer]
PublicKey = Peer1PublicKey
PresharedKey = Peer1PresharedKey
AllowedIPs = 101.115.117.2/32
PersistentKeepalive = 25

[Peer]
PublicKey = Peer2PublicKey
PresharedKey = Peer2PresharedKey
AllowedIPs = 101.115.117.3/32
PersistentKeepalive = 25

[Peer]
PublicKey = Peer3PublicKey
PresharedKey = Peer3PresharedKey
AllowedIPs = 101.115.117.4/32
PersistentKeepalive = 25

[Peer]
PublicKey = Peer4PublicKey
PresharedKey = Peer4PresharedKey
AllowedIPs = 101.115.117.5/32
PersistentKeepalive = 2
```

Konfigürasyon 3.4 Wireguard Server Dosya Yapılandırması

Sunucu üzerinde 4 adet kullanıcı bilgileri (Peer 1, 2, 3, 4) paylaşılmıştır. Sunucu üzerinde aktif olarak bu şekilde 11 adet kullanıcı tanımlaması yapılmıştır. Bu bilgiler üzerinden Wireguard Peer kullanıcıları sunucuya Statik IP üzerinden 25000 portunu kullanarak VPN erişimi sağlanmaktadır. PostUP ve PostDown komutlarıyla iptables kuralları oluşturulmuştur.

WGDashboard Arayüz Ekranı : WGDashboard, WireGuard VPN sunucusunu yönetmek için kullanıcı dostu bir web arayüzü sağlar[22]. Aşağıdaki adımlar, WGDashboard'ın Debian Bullseye işletim sisteminde kurulum aşamalarını göstermektedir.

Arayüz ekran tasarımı için bazı kurulumlar yapılmaktadır. Bu kurulumlar kapsamında arayüz ekranı çalıştıracak programın github repository olarak mevcut olduğu için aşağıdaki komutlar "git" aracı kurularak githubdaki uygulamayı sunucuya indirme işlemi yapılmaktadır.

- sudo apt install git curl

Yukarıdaki aracın kurulumu yapıldıktan sonra aşağıdaki komut ile arayüz dosyaları sunucuya indirilmektedir.

- git clone https://github.com/donaldzou/WGDashboard.git

İndirimi tamamlanmış olan WGDashboard klasöründe bulunan uygulama kurulumu için bazı kurulumlar gerekmektedir. Bu kurulumlar için gereken araçlar şunlardır:

- sudo apt install python3
- sudo apt install python3-pip

Python3 kurulum işlemi tamamlandıktan sonra Python içinde uygulamanın kullandığı kütüphaneler bulunmaktadır. Kullanılan bu kütüphaneler şunlardır:

- pip3 install Flask
- pip3 install ifcfg
- pip3 install psutil
- pip3 install icmplib
- pip3 install Flask-QRcode
- pip3 install gunicorn
- pip3 install certbot

Buradaki kütüphanelerin kurulması gerekmektedir. Buradaki kurulumları toplu şekilde yapmak için /root/wgdashboard/src klasörünün içerisine girip aşağıdaki Python kodunu yazarak gerekli kütüphaneler kurulabilmektedir.

- pip3 install -r requirements.txt

Gerekli kurulumlar tamamlandıktan sonra klasör içinde bulunan ‘wg-dashboard.service’ dosyasının içeriği Konfigürasyon 3.5’deki gibi yapılmalıdır.

```
[Unit]
After=netword.service

[Service]
WorkingDirectory=/root/wgdashboard/src
ExecStart=/usr/bin/python3 /root/wgdashboard/src/dashboard.py
Restart=always

[Install]
WantedBy=default.target
```

Konfigürasyon 3.5 Node-RED Sistem Servis Yapılandırması

Burada servis dosyası, arayüz programının başlangıçta otomatik olarak çalışabileceği şekilde ayarlanmıştır. Bu dosya ‘/etc/systemd/system/’ klasörünün içine kopyalanmıştır. Bu işlemden sonra aşağıdaki komutlarla dosyaya izin verdikten sonra servisi başlatıp, işletim sistemin açılışında arayüz uygulamasının çalışmasını sağlamaktadır.

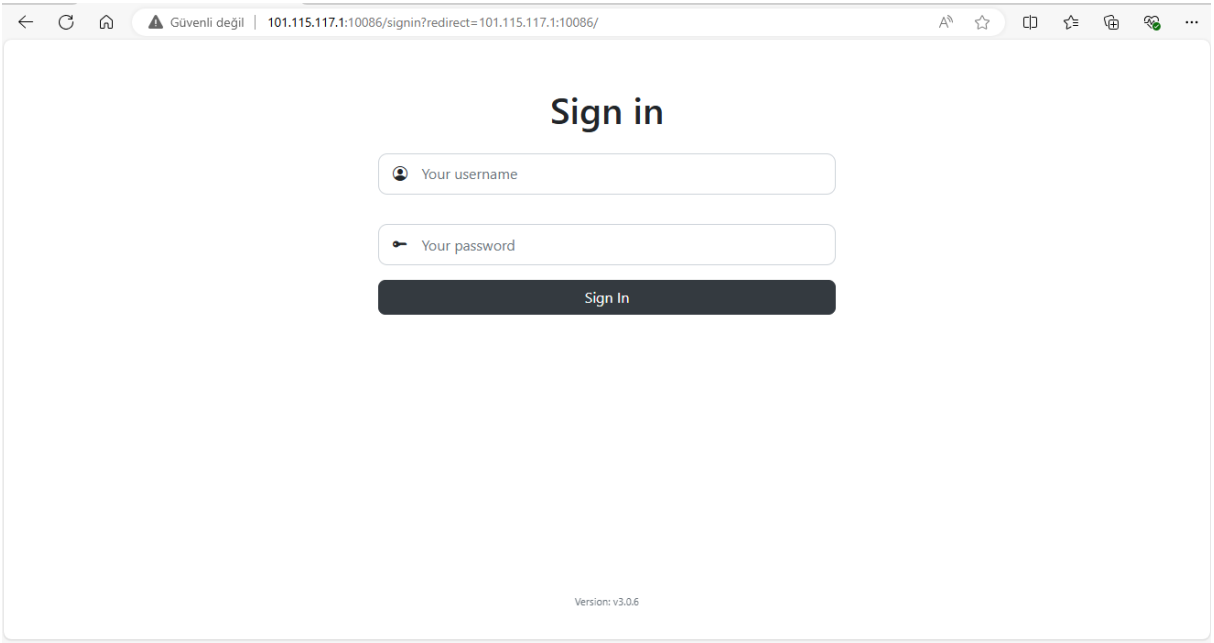
- sudo chmod 644 /etc/systemd/system/wg-dashboard.service
- sudo systemctl daemon-reload
- sudo systemctl enable wg-dashboard.service
- sudo systemctl start wg-dashboard.service

Yukarıdaki komut işlemlerinden sonra servisin çalışma durumunu kontrol etmek için *sudo systemctl status wg-dashboard.service* komutu kullanılmaktadır.

Kurulumla birlikte gelen ‘wg-dashboard.ini’ isimli konfigürasyon parametreleri içeren bir dosya bulunmaktadır. Bu dosya içerisinde arayüzün çalışacağı ip adresi ve portun belirlenmesi, kullanıcı adı ve şifresi gibi birçok ayarlama yapılmaktadır. Varsayılan şekilde gelen bu konfigürasyon ayarlamaları değişiklik yapılmadan çalıştırılmıştır. Bu varsayılan konfigürasyon ile 10086 portu üzerinden web tabanlı browser aracılığıyla arayüz ekrana erişim sağlanabilmektedir.

Yukarıdaki kurulum işlemleri yapıldıktan sonra Wireguard VPN tünel aktif olması durumunda sunucunun çalıştığı 101.115.117.1:10086 ip adresi ve portuyla bağlantı kurulumu Şekil 3.5’de gösterilmektedir. Şekil 3.5’de gösterilen ana sayfada kurulum esnasında klasör içerisinde

bulunan ‘wg-dashboard.ini’ dosyası içerisinde default kullanıcı adı ve şifresiyle erişim sağlanabilmektedir. Kullanıcı bilgilerinde değişiklik yapılması durumunda dosyada bulunan bölümden değişiklik yapıldıktan sonra systemd servisinin durdurulup tekrar başlatılması gerekmektedir.



Şekil 3.5 Wireguard Dashboard Kullanıcı Girişi Sayfası

3.7. Android Studio ile Mobil Uygulama Geliştirme

Bu bölümde, Android Studio kullanılarak Flutter ile Node-RED'e bağlanacak ve VPN bağlantısı kuracak bir mobil uygulamanın geliştirilmesi ele alınmıştır. Uygulama, Wi-Fi ağı üzerinden cihaza bağlanmakta ve ağ bağlantısı sağlandıktan sonra Node-RED'in 1880 portunda çalışan dashboard ekranını açmaktadır. Bu arayüz ekranında gerekli ayarlamaların yapılmasının ardından, cihazdan WireGuard VPN tünel bilgileri alınarak tünel içindeki sunucuda çalışan Python uygulamasına bağlantı sağlanmaktadır.

Android Studio, Google tarafından sağlanan Android uygulamaları geliştirmek için kullanılan yazılım geliştirme aracıdır (IDE). IntelliJ IDEA tabanlı olan Android Studio, Android uygulama geliştirme sürecini kolaylaştıran çeşitli araçlar ve özellikler sunar. Android Studio'nun temel özellikleri şunlardır [23]:

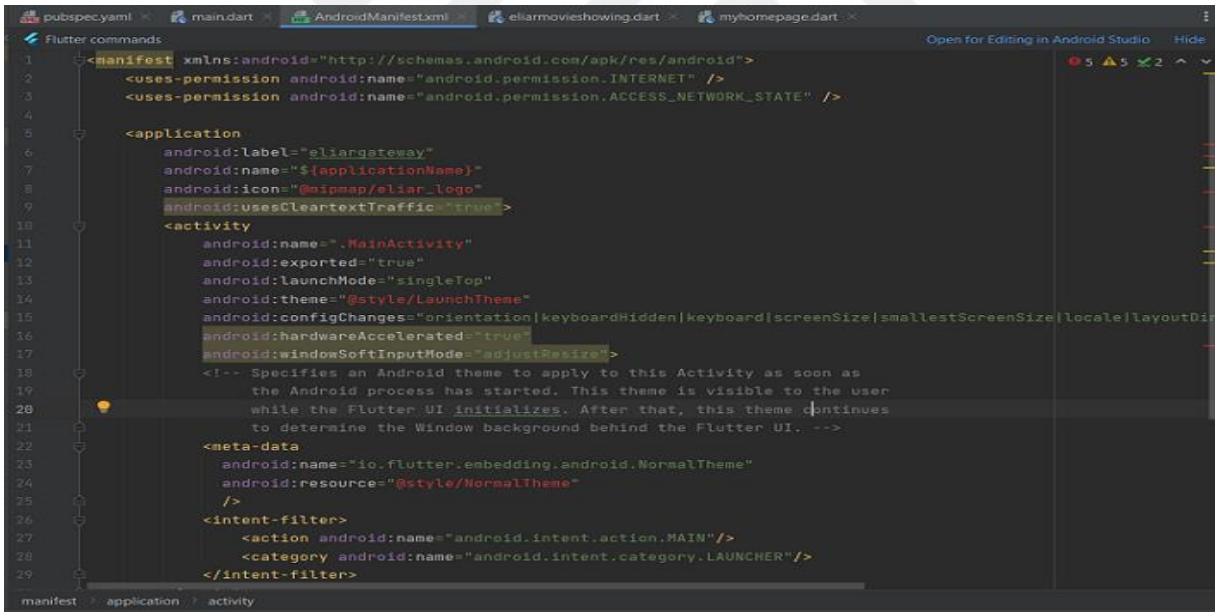
- Akıllı Kod Düzenleyici: Hata ayıklama ve kod tamamlama özellikleriyle geliştiricilere yardımcı olur.

- Emülatör: Geliştiricilerin Android uygulamalarını çeşitli cihazlarda ve Android sürümlerinde test etmelerini sağlar.
- Yerleşik Profil Araçları: Uygulamanın performansını izleme ve optimize etme araçları sunar.
- Gradle Tabanlı Yapı Sistemi: Projelerin derlenmesi, paketlenmesi ve dağıtılması için esnek bir yapı sistemi sağlar.
- UI Design Tools: Sürükle ve bırak arayüzü tasarım araçları ile kullanıcı arayüzlerini oluşturmayı kolaylaştırır.

Android Studio ile geliştirilen bu çalışmada uygulama için platform içerisinde bulunan 'AndroidManifest.xml' dosyasında bazı eklemeler yapılması gerekmektedir.

Şekil 3.6'da AndroidManifest dosyasında yapılan konfigürasyon işlemleri göstermektedir. Sistemimizde kullanacağımız internete erişim, network durumları gibi parametreleri girmemiz gerekmektedir. Oluşturulan mobil uygulamada interneti kullanacak olmasından dolayı uygulamaya gerekli izin şu parametreler ile verilmiştir:

- <uses-permission android:name="android.permission.Internet" />
- <uses-permission android:name="android.permission.Access_Network_State" />



Şekil 3.6 AndroidManifest Konfigürasyonu

Bu işlemler ile mobil uygulamanın, mobil cihazın internet bağlantısını kullanabilme yetkisi sağlanmıştır. Bu parametrelerin yanı sıra, mobil uygulama için isim, logo gibi ayarlamalar da şu şekilde yapılmıştır:

- android:label="eliargateway"
- android:icon="@mipmap/eliar_logo"
- android:usesCleartextTraffic="true">

Yukarıdaki konfigürasyon ile 1. adımda uygulama ismini ‘eliargateway’, 2. adımda ‘eliar_logo’ isimli görüntü de icon olarak belirlenmiştir. 3. adımdaki işlemle de uygulamanın http trafiğini kullanma izni verilmiştir.

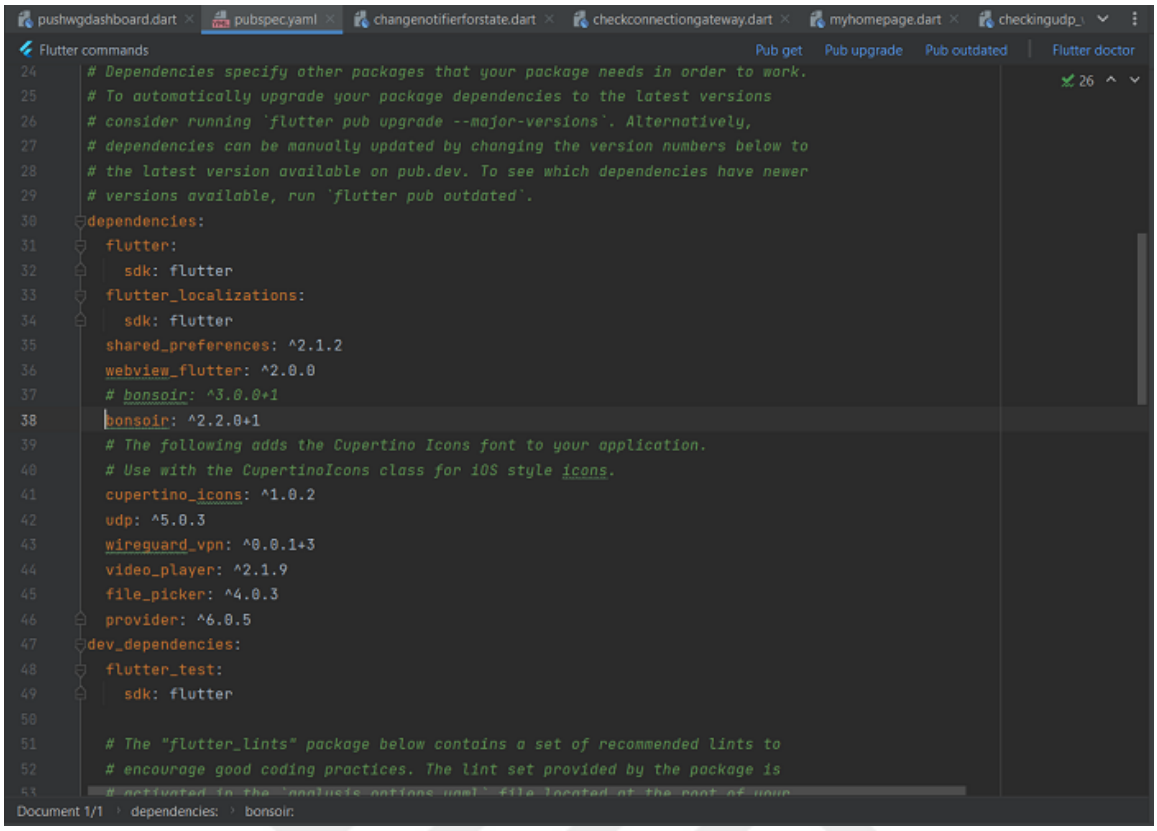
Android Studio ile Flutter kullanarak geliştirilen uygulamalar, iOS işletim sisteminde çalıştırılmak üzere Mac cihazlarda Xcode ile derlenebilmektedir. Bu nedenle, uygulamanın Flutter ve Dart programlama dili ile geliştirilmesi tercih edilmiştir.

Flutter, Google tarafından geliştirilen açık kaynaklı bir UI yazılım geliştirme kitidir (SDK). Flutter, çapraz platform geliştirme imkanı sunarak hem iOS hem de Android için natively compiled uygulamalar oluşturmayı sağlar [24]. Flutter'ın temel özellikleri şunlardır:

- Dart Programlama Dili: Flutter, Dart programlama dilini kullanır ve modern, reaktif bir framework sunar.
- Widget Tabanlı Yapı: Flutter, önceden oluşturulmuş widget'lar sunarak kullanıcı arayüzlerini kolayca oluşturmayı sağlar. Her şey bir widget olarak temsil edilir.
- Hot Reload: Geliştiricilerin kodda yaptığı değişiklikleri anında görmelerini sağlayan hızlı bir geliştirme döngüsü sunar.
- Yüksek Performans: Flutter, natively compiled kod üreterek yüksek performans sağlar.
- Platform Uyumu: iOS ve Android dışında web ve masaüstü platformları için de uygulama geliştirme imkanı sunar.

Flutter ile kodlama yapılması için çalışacak kodun bağımlılıkları içeren kütüphaneleri ‘pubspec.yaml’ isimli sayfada belirtilmesi gerekmektedir. Bu bağımlılıklar Şekil 3.7’de gösterilmiştir.
Şekil 3.7’de dart programlama dilinde kullanılacak kütüphanelerin tanımlaması yapılmıştır. Kullanılacak kütüphaneler şu şekildedir:

- webview_flutter: ^2.0.0
- bonsoir: ^2.2.0+1
- cupertino_icons: ^1.0.2
- udp: ^5.0.3
- wireguard_vpn: ^0.1.0+3
- video_player: ^2.1.9
- provider: ^6.0.5
- flutter_localizations:



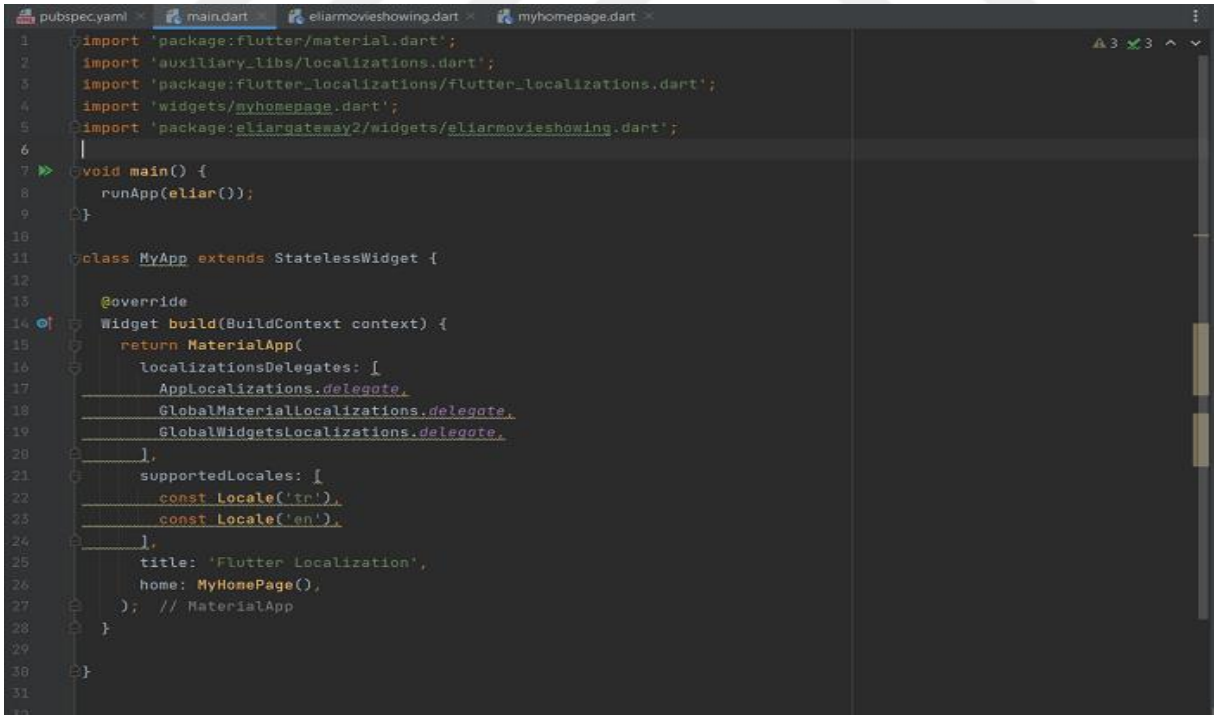
Şekil 3.7 Pubspec.yaml Konfigürasyonu

Yukarıdaki kütüphaneler dart programlamada kullanılacak kütüphane listesini içermektedir. Kullanılan kütüphanelerin kısaca görevleri aşağıda açıklanmıştır:

- webview_flutter: ^2.0.0: Flutter uygulamasında yerel bir WebView (web tarayıcı bileşeni) oluşturulmasını sağlar. Bu paket sayesinde uygulama içinde web sayfaları görüntülenebilir ve kullanıcıların web içeriğiyle etkileşime geçmesi sağlanabilir.
- bonsoir: ^2.2.0+1: Flutter uygulamasında yerel ağ (LAN) üzerinden hizmet keşfi ve yayınlaması yapmasını sağlar. Bu, uygulamaların aynı yerel ağda bulunan diğer cihazları bulmasına ve onlarla iletişim kurmasına olanak tanır. Örneğin, aynı ağdaki bir cihazda çalışan bir hizmeti keşfetmek için kullanılabilir.
- cupertino_icons: ^1.0.2 : Cupertino stilinde (iOS tarzında) ikonlar sağlar. Bu paket, iOS benzeri bir görünüm ve his sağlamak için iOS tasarım yönergelerine uygun ikonları içerir ve Flutter uygulamalarında kullanılabilir.
- udp: ^5.0.3 : Flutter uygulamasında UDP (User Datagram Protocol) ağ iletişimini sağlar. UDP, hızlı ve hafif bir iletişim protokolüdür ve genellikle gerçek zamanlı uygulamalar veya düşük gecikme süresi gerektiren durumlar için kullanılır.

- wireguard_vpn: ^0.1.0+3: Flutter uygulamasında WireGuard VPN tünelleri oluşturmasına ve yönetmesine olanak tanır.
- video_player: ^2.1.9 : Flutter uygulamalarında video oynatılmasına olanak tanır. Bu paket, video dosyalarını veya akışları yerel olarak veya internet üzerinden oynatmak için kullanılabilir. Video oynatma işlemleri için temel kontroller ve işlevler sağlar.
- provider: ^6.0.5 : Flutter uygulamalarında durum yönetimi için kullanılan popüler bir pakettir. Uygulamanın çeşitli bileşenleri arasında veri paylaşımını ve yönetimini kolaylaştırır. Provider, reaktif programlama ile değişiklikleri dinleyerek kullanıcı arayüzünü otomatik olarak günceller ve bağımlılık enjeksiyonu ile sınıflar arasında bağımlılıkları yönetir.
- flutter_localizations: Flutter uygulamalarında yerelleştirme (localization) desteği sağlayan bir pakettir. Bu paket, uygulamada farklı diller için uygun metin ve formatları gösterebilmesini sağlar. Örneğin, tarih ve sayı formatlarının yerel dilde doğru bir şekilde görüntülenmesi gibi işlevleri içerir.

Sıralanan bu kütüphanelerin eklenmesinden sonra dart programlamasının ilk sayfasındaki kodlar Şekil 3.8'de gösterilmektedir.



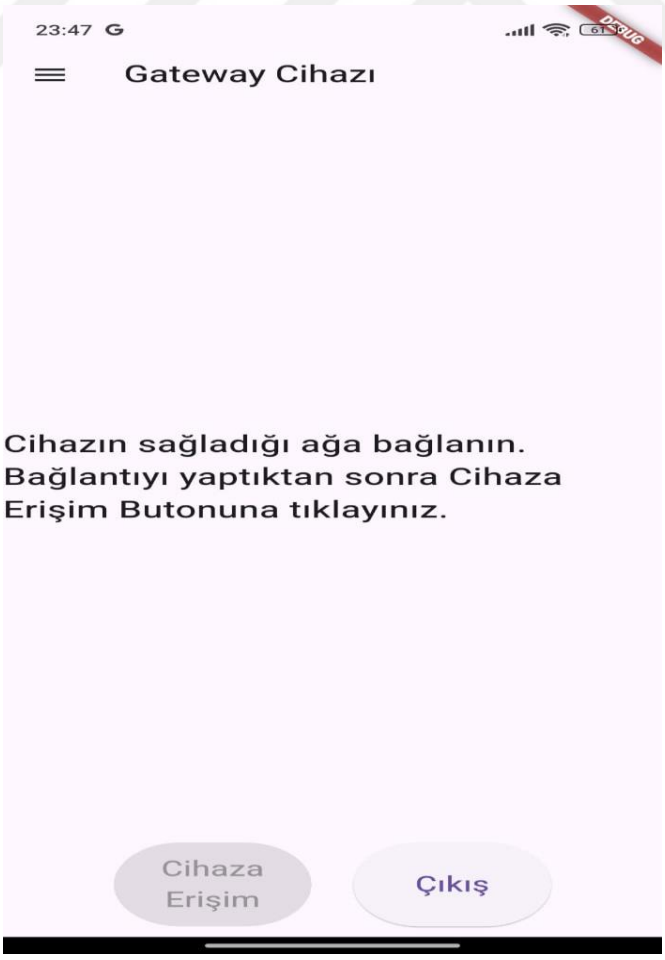
Şekil 3.8 Dart Programlamada Ana Sayfa

Şekil 3.8'de Dart programlama dili ile yazılmış ana sayfanın kodları gösterilmektedir. İlk bölümde, kullanılacak olan kütüphaneler ve diğer kodlama sayfaları import edilmektedir. Bu

import işleminden sonra, ana sayfada bir widget ekranı oluşturulmakta ve bu ekran, oluşturulan metinlerin localization ile İngilizce ve Türkçe desteğini sunmaktadır. Mobil uygulamanın aktif olarak kullandığı dile göre uygun gösterim sağlanmaktadır. Bu işlemlerin ardından, MyHomePage kodlama sayfası devreye girmekte ve diğer sayfalardaki kodlar çalıştırılmaktadır.

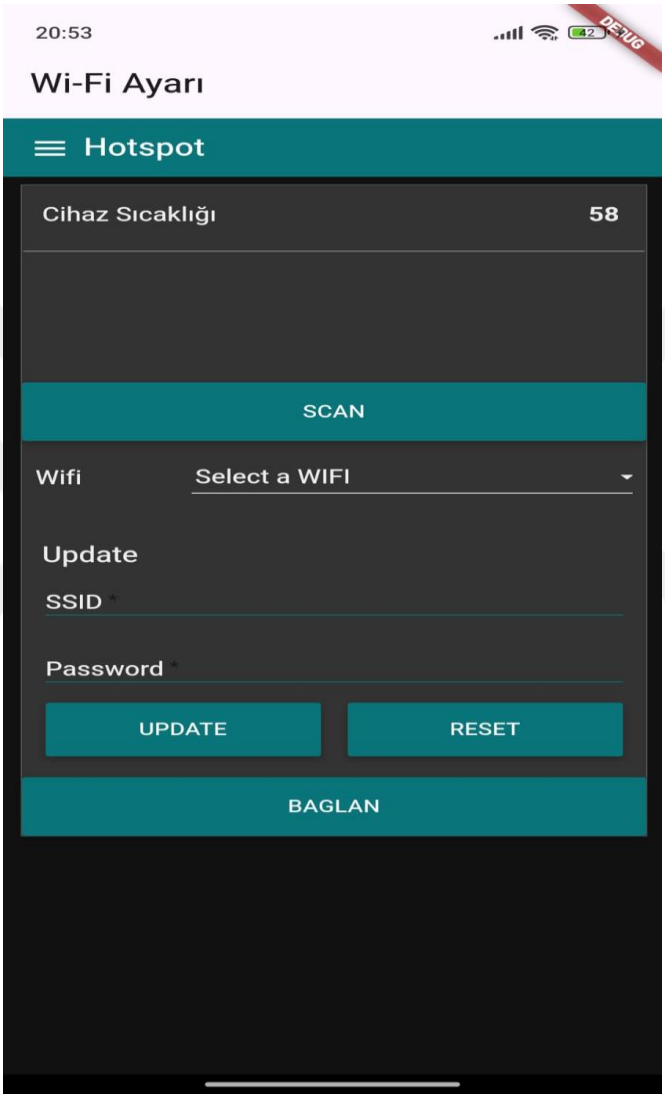
Android Studio ile Dart programlama dili kullanılarak geliştirilen uygulama, derleme işlemi tamamlandıktan sonra Android işletim sistemine sahip cihazlara yüklenmektedir. Bu bölümde, uygulama ile oluşturulan widgetların anlatımı yapılmıştır.

Şekil 3.9'da mobil cihaza yüklenmiş uygulama gösterilmektedir. Bu widget alanında, kullanıcıya mobil cihaz ile ağ geçidi cihazının aynı ağa bağlandıktan sonra "Cihaza Erişim" butonunu kullanarak cihaza bağlantı sağlayabileceği uyarısı verilmektedir. Mobil cihaz, ağ geçidi cihazının hotspot ile yaydığı ağa bağlandıktan sonra "Cihaza Erişim" butonu aktif olarak kullanılabilir hale gelecektir. Buton kullanım durumunda bağlantı gerçekleştirilir. "Çıkış" butonu ise uygulamadan çıkış işlemini gerçekleştirir.



Şekil 3.9 Mobil Cihaz Üzerinde Gateway Cihazına Erişim Sayfası

Şekil 3.10'da, önceki adımda "Access Device" butonuna basıldıktan sonra mobil cihaz ve gateway cihazı aynı ağa bağlanır ve birbirlerinin IP adreslerini öğrendikten sonra mobil cihaz üzerinden webview ile Node-RED dashboard ekranı açılır. Bu ekran üzerinde sol üstteki tab menüden "WiFi Ayarlama", "Proxy Yapma", "Ethernete Bağlı Cihazları Listeleme" ve "Cihaz Projesi" gibi ekranlara erişim sağlanmaktadır.

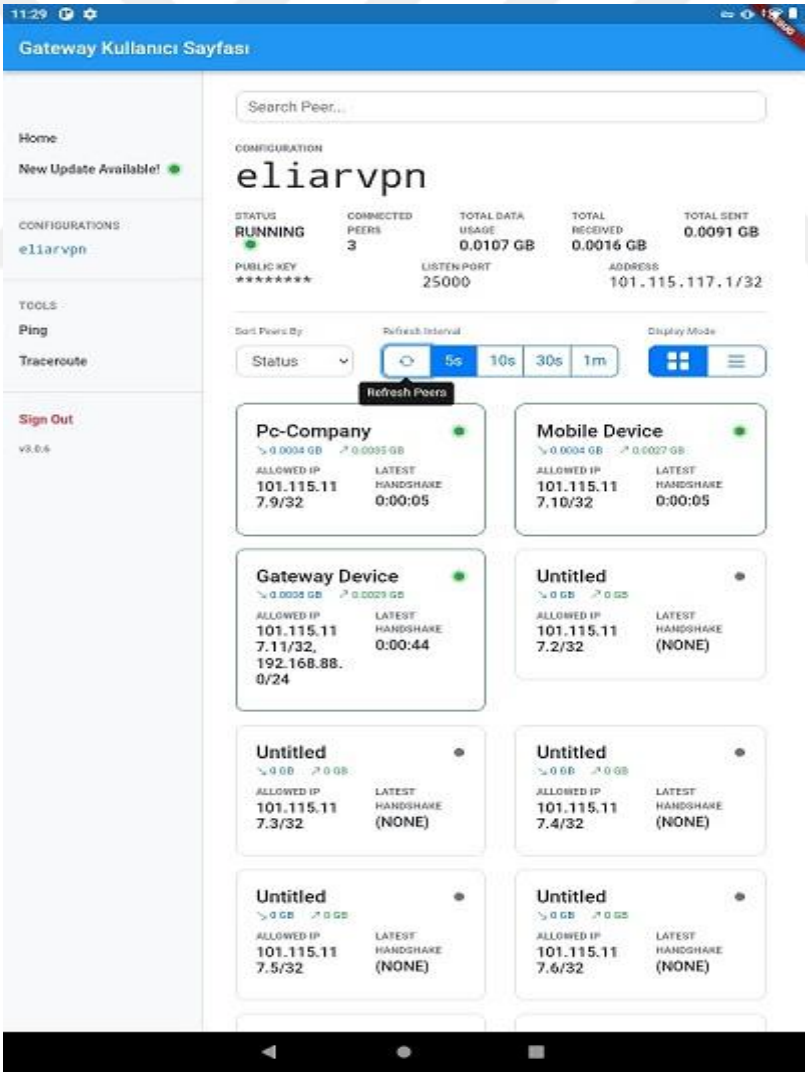


Şekil 3.10 Mobil Cihaz ile Node-RED Dashbord Ekranına Erişim

Ana ekran olarak gelen WiFi ekranında, gateway cihazının bağlantı kuracağı WiFi ağının adresi taranarak seçilir ve şifresi girilip "Bağlan" butonuna tıklanır. Bu işlem sonrası hotspot bağlantısı kapatılarak cihazın WiFi ağına bağlanması sağlanır. Butona tıklandıktan sonra, gateway cihazı üzerinden WireGuard VPN kullanıcı bilgisi mobil cihaza gönderilir ve bağlantı kurma işlemi başlatılır. Kullanıcıya, bağlantı ayarlarının girildiğine dair bilgi verilir ve mobil cihazın internete bağlanması ile VPN tünelinin aktif hale gelmesi beklenir.

Bu bekleme süresi 30 saniye olarak belirlenmiştir. Cihaz internete başarılı bir şekilde bağlanır ve VPN tüneli kurulur ise, aktif kullanıcıların olduğu Python ile yapılan dashboard ekranı mobil cihaz üzerinden açılır.

Şekil 3.11'de, önceki bölümdeki işlemlerin başarılı şekilde gerçekleşmesinden sonra açılan ekran gösterilmektedir. Bu ekranda, WireGuard VPN sunucusu üzerinde tanımlı 11 kullanıcıdan 3 tanesinin aktif olarak bağlı olduğu görülmektedir. Diğer kullanıcılar herhangi bir cihaza tanımlı olmadığı için "Untitled" olarak adlandırılmıştır. Mobil cihaz, gateway cihazı ve bilgisayar kullanıcısı aktif olarak görüntülenmektedir. Gateway cihazına 192.168.88.0/24 alt ağ adresi tanımlaması yapıldığı görülmektedir, bu da WireGuard VPN tüneline gateway cihazı üzerinden 192.168.88.1-254 IP adresli cihazlara erişim sağlanabileceği anlamına gelmektedir. Şeklin sol tarafında gösterilen ‘ping’ ve ‘traceroute’ araçları kullanılarak cihazların erişim testi yapılabilmektedir.

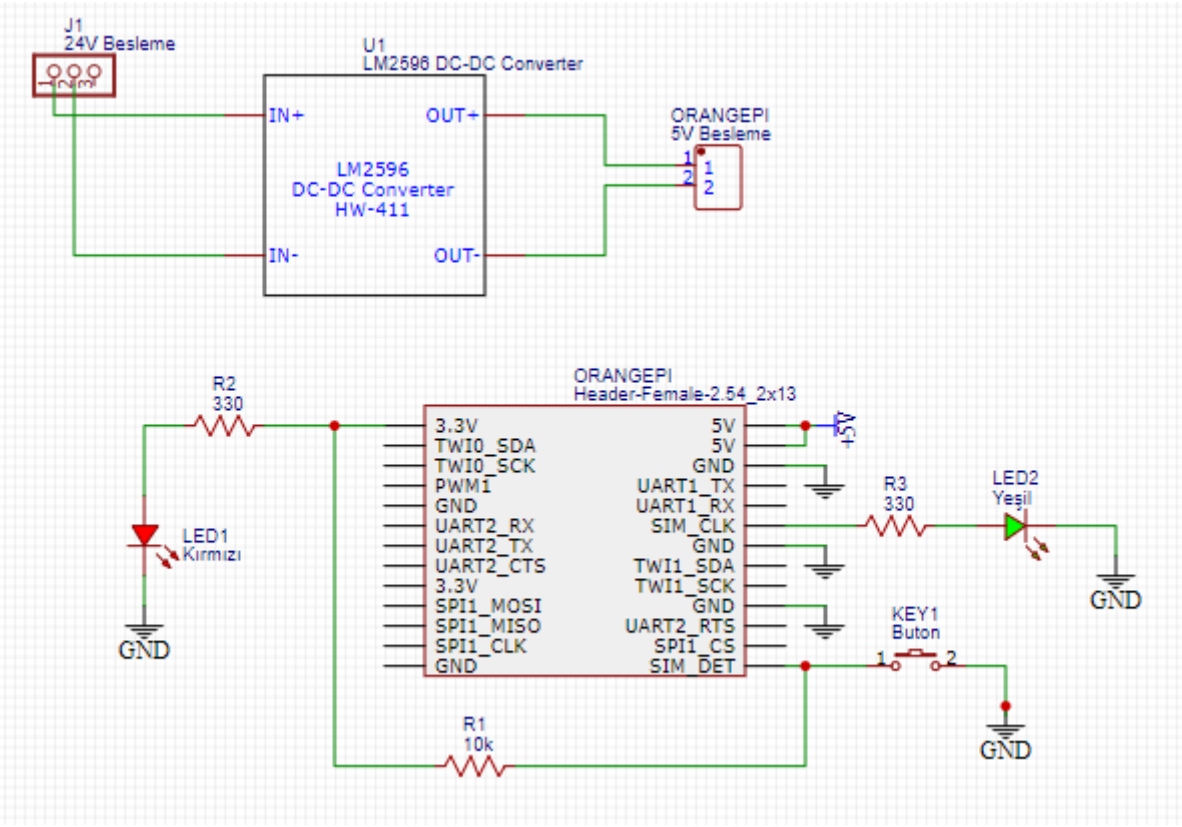


Şekil 3.11 Mobil Cihaz Üzerinde Aktif Kullanıcı Sayfasının Gösterimi

3.8. Donanım Tasarımı ve İşlevleri

Bu bölümde geliştirilen sistemin donanım tasarımına ilişkin bilgiler sunulmuştur. Öncelikle donanımın iç yapısı ve bileşenleri tanıtılmış; ardından, bu bileşenlerin çalışma prensipleri ve birbirleriyle etkileşimi açıklanmıştır.

Şekil 3.12'de donanıma ait açık devre şeması verilmiştir. Aşağıda kullanılan malzemelerin işlevleri açıklanmaktadır.

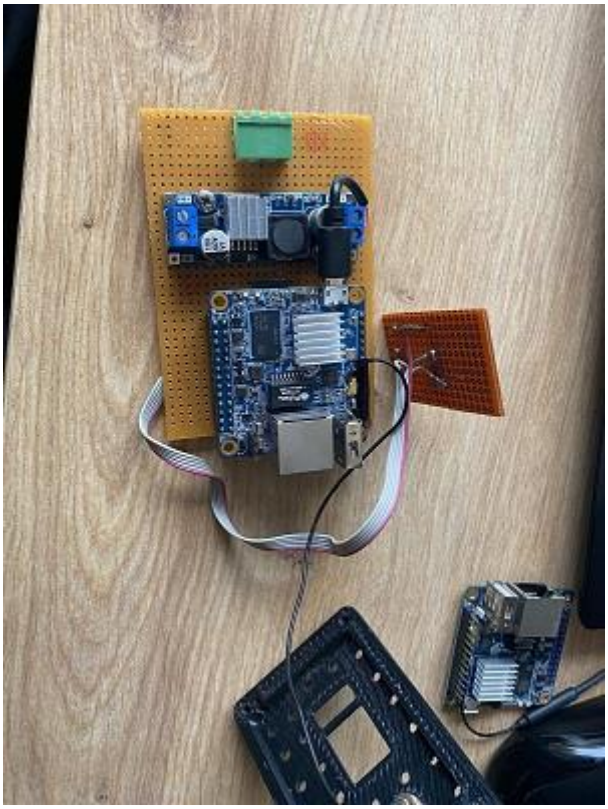


Şekil 3.12 Ağ Geçidi Cihazın Açık Devre Şeması

Cihazın endüstri ortamında çalışması için sıklıkla 24V enerji beslemesi gerekir. Bundan dolayı cihazın devre şemasında ‘J1’ isimli 24V besleme soket girişi bulunmaktadır. Cihazda kullanılan orange pi zero kartı 5V enerji seviyesinde çalışmaktadır. Bu yüzden 24V enerji beslemesini 5V seviyesine düşürecek DC-DC Converter kullanılmaktadır. ‘U1’ isimli dönüştürücüsü 24V enerji beslemesini orange pi zero için 5V enerji dönüştürecek şekilde bağlantıları gerçekleştirilmiştir. Cihazda 2 adet led ve 1 adet buton kullanılmıştır. Bu komponentlerin kullanım amaçları aşağıda açıklanmıştır:

- ‘LED1’ isimli Kırmızı led cihazda enerjinin bulunduğunu cihaz kullanıcısına belirtecek şekilde 3.3V enerji çıkışından 330Ω dirençle bağlantısı kurulmuştur.
- ‘LED2’ isimli Yeşil led cihazda bazı işlevleri gerçekleştirmesine yarayacak şekilde anlamlandırılması yapılmıştır. Yeşil led yanmaması internet bağlantısının olmadığı anlamını taşımaktadır. Flaşör şekilde yanma durumu, hotspot modunda cihaz konfigüre edilebilir anlamı taşımaktadır. Sabit yanma durumunda da cihaz aktif kullanılabilir anlamına gelmektedir.
- Devrede bulunan ‘KEY1’ isimli buton cihazı hotspot moduna geçirmek ya da cihazı yeniden başlatmak amacıyla kullanılmaktadır. Buton 5-10 sn aralığında basılma durumunda cihaz yeniden başlayacaktır. 10 saniyeden daha fazla basılma durumunda ise cihaz hotspot moduna geçmektedir.

Şekil 3.13'de Orange Pi Zero kartının dişi soketlere bağlantısı gösterilmektedir. Bu bağlantıda, kart üzerindeki Wi-Fi anteninin kablosu takılmış ve cihaz kapağına bağlanmış durumdadır. Şekil 3.14'de ise, bu yapı 3D yazıcı ile üretilen cihaz kutusuna monte edilmiş haliyle gösterilmektedir.



Şekil 3.13 Orange Pi Zero Kartının Montajı

Bu bölümde, önceki bölümde anlatılan kartın 3D yazıcı ile yapılan kutusuna monte edilmesi süreci ve bu kutulamanın işlevselliği hakkında bilgi verilmiştir.



Şekil 3.14 Tasarlanan Cihazın farklı yönlerden görünüşü a)Ön, b)Arka, c)Üst ve d)Alt

Şekil 3.14'de, cihazın dört ayrı yüzeyi bir harf ile temsil edilmiştir:

A harfi ile ön yüzey: Cihazın ön yüzü ve anten bağlantısı gösterilmektedir. Cihazın ön yüzünde, buton ve LED'ler bulunmaktadır. Kırmızı LED, cihazda enerjinin mevcut olduğunu gösterir. Yeşil LED'in flaşör şeklinde yanması, cihazın hotspot modunda olduğunu; sabit yanması ise WiFi ağ konfigürasyonunun yapıldığını belirtmektedir. Şekilde gözüken buton ile hotspot moduna geçilip ağ paylaşımı yapılabilmektedir. Cihazın yaydığı ağa mobil cihazlar, bilgisayarlar gibi WiFi kartına sahip elektronik cihazlar ile bağlantı kurulabilmektedir.

B harfi ile arka yüzey: Cihazın arka yüzeyi gösterilmektedir. Endüstride kullanılan panolarda bulunan raylara montaj aparatı, şekilde işaretli alanda belirtilmiştir. Bu aparat, cihazı panoya takıp kullanmak isteyen kullanıcılar için tasarlanmıştır.

C harfi ile üst yüzey: Cihazın üst yüzeyi gösterilmektedir. Üst yüzeyde, Ethernet bağlantı girişi, USB bağlantı girişi ve Wi-Fi adaptörünün daha iyi sinyal alabilmesi için anten bulunmaktadır.

D harfi ile alt yüzey: Cihazın alt yüzeyi gösterilmektedir. Endüstriyel panolarda kullanılma durumu için enerji besleme yeri bu alt yüzeye yerleştirilmiştir. Alt yüzey etrafındaki delikli alan, cihaz içerisinde hava akışına izin verilmesi amacıyla tasarlanmıştır.

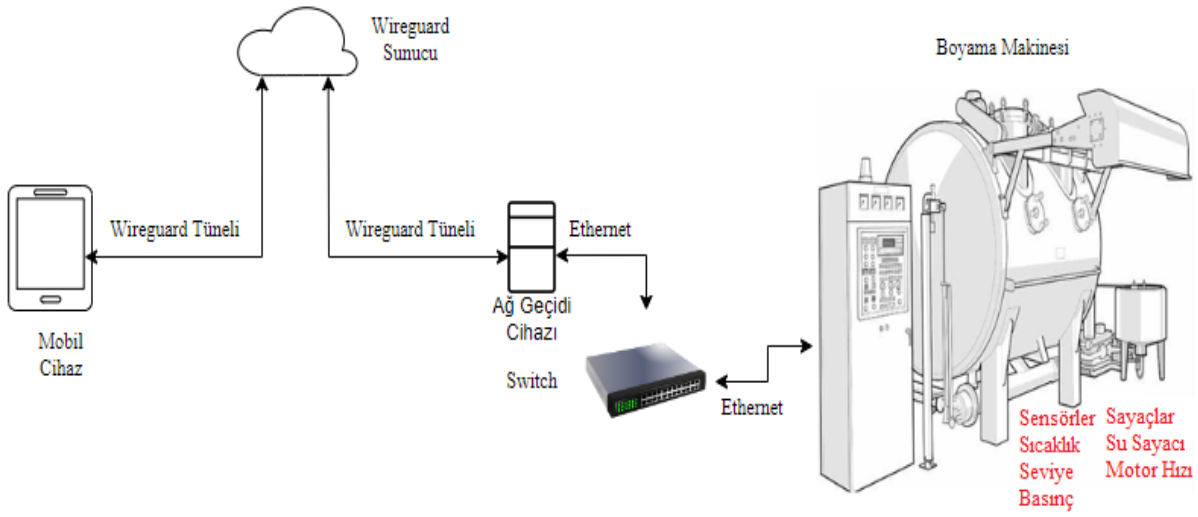
4. UZAKTAN İZLEME ve KONTROL

Önceki bölümlerde Ağ geçidi donanımın montajı ve yazılımsal ayarlamaları gerçekleştirilmiştir. Bu bölümde ise, hazır hale gelen cihazın mobil cihaz ile ağ yapılandırması yapıldıktan sonra cihazın internet bağlantısı etkinleştirilmiştir. İnternet bağlantısı aktif olmasıyla birlikte cihaz üzerinde bulunan VPN tüneli ayaklandırılmıştır. Bu aşamadan sonra cihaz uzaktan erişim sağlayacak hale gelmiş olup; sahada birkaç uygulama senaryosu için hazırdır. Oluşturulan mobil uygulama, bir telefon/tablet cihazına yüklenmiştir.

Tasarlanan donanım kullanılarak, bir uygulama senaryosu kurgulanmıştır. Senaryoda 3 farklı makineye erişim sağlanmaktadır. Sahada bulunan bu makinelere bağlantı gerçekleştirip Codesys tabanlı PLC cihazına uygulama yükleme ve yazılım güncellemesi yaptırma, InfluxDB Explorer ile veri tabanındaki verilere erişilmesi, Grafana'dan veri görselleştirmesinin yapılması ve HMI cihazının mobil cihaz ile kontrol yapılması sağlanmıştır.

4.1. Senaryo 1: Mobil bir cihaz üzerinden uzak makineye bağlantı

Bu makinelerden birincisi, tekstil sektöründe kullanılan bir kumaş boyama makinesidir. Kumaş boyama ve yıkama makinelerinde batch engine sistemine sahip kontrolör cihazları ile çalışmaktadır. Genel bir yapı içermesinden dolayı bu tür makinelerin devreye alım kısmında cihazda bulunan HMI panel üzerinden gerçekleşmektedir. Makineye göre cihazlar özelleştirilerek kullanıma sunulmaktadır.



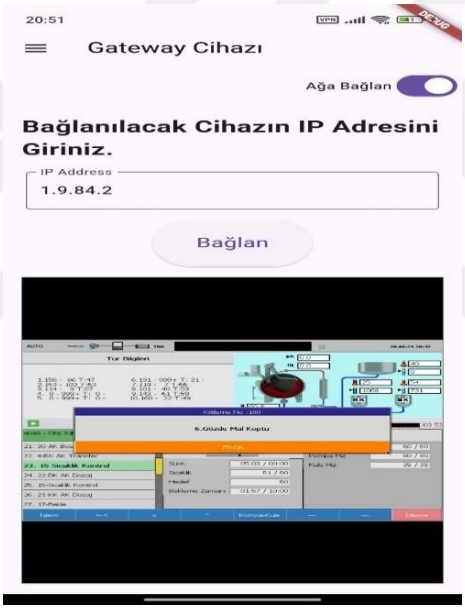
Şekil 4.1 Gateway Cihazının Boyama Makinesine Bağlantı Şeması

Şekil 4.1'de gateway cihazı bir tekstil şirketinde bağlantısı gösterilmektedir. Gateway cihazının şirket içerisinde bulunan makinelerin bağlı olduğu endüstriyel switch cihazına ethernet kablosu ile bağlantısı gerçekleştirilmiştir. Wifi bağlantısı ile de internet erişimi sağlanarak daha önceden

oluşturduğumuz Wireguard sunucumuza erişimi sağlanmıştır. Bu tekstil şirketinde takılan cihaz ile Şekil 4.2'deki mobil cihaz erişimi ve Şekil 4.3'deki InfluxDB editörüne bilgisayarla erişimi gerçekleştirilmiştir.

Şekil 4.2'de Makine HMI paneline erişimin mobil cihaz yardımıyla uzaktan erişim imkanı sağlanmıştır. Mobil cihaz ile boyama makinesine uzaktan erişim, kullanıcılara aşağıdaki faydaları sağlamaktadır:

- Mobil cihazı ile uzaktan devreye alma imkanı sağlamasıyla ulaşım, zaman ve personel giderlerinden tasarruf edilmiştir.
- Bir operatör başına 5-7 makine düştüğü göz önüne alınırsa, prosesi durduran alarmları daha hızlı müdahale etme imkanı sağlamıştır. Bu duruş süresini kısaltarak makinenin verimliliği artırmıştır.



Şekil 4.2 Mobil Cihaz ile Kontrolör Paneline Erişim

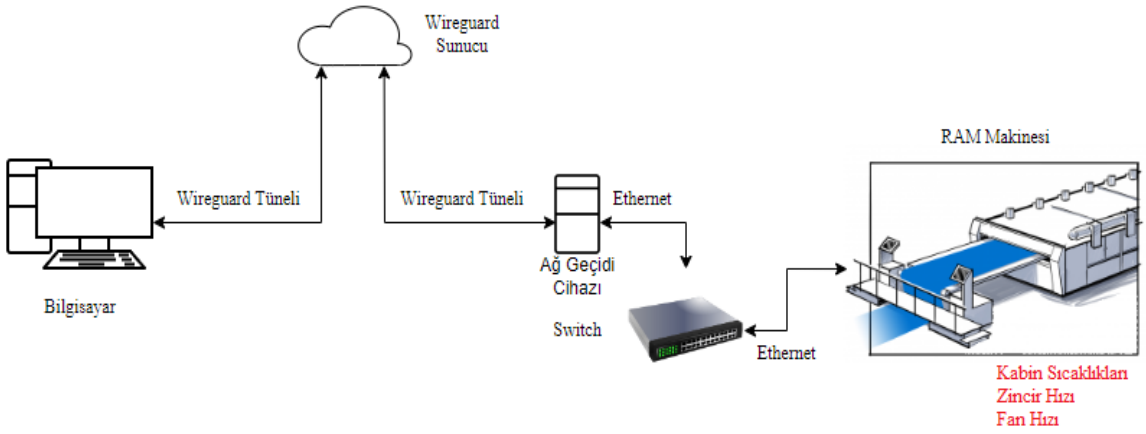
Bu kumaş boyama makinesinde boyama işlemi sırasında elektrik, su ve doğalgaz gibi kaynaklar tüketilmektedir. Doğal kaynakları tüketen bu tür sistemlere sahip şirketler, makineye ait geçmiş tüketim verilerine erişmek istemekte ve istenen veriler üzerinde filtrelemeler yaparak raporlar oluşturulmasını arzu etmektedir. Şekil 4.3'de gösterilen InfluxDB veritabanında, çeşitli filtreler uygulanarak soğuk su tüketim verilerine erişimi gösterilen senaryoya yer verilmiştir.



Şekil 4.3 InfluxDB Veritabanında Su Tüketimine Ait Grafik

4.2. Senaryo 2: Bir PC üzerinden uzak makineye bağlantı

Şekil 4.4'de ağ geçidi cihazının başka bir uzak makineye bağlantısı gösterilmektedir. Gateway cihazının şirket içerisinde bulunan makinelerin bağlı olduğu endüstriyel switch cihazına ethernet kablosu ile bağlantısı gerçekleştirilmiştir.



Şekil 4.4 Bilgisayar ile Ram Makinesine Uzaktan Erişim Şeması

Ram makineleri, kumaşları yüksek sıcaklıklara ulaşan kabinlerden geçirir ve kumaşlar, çeşitli işlemlerden geçerken kenarlarından zincirler ile tutulur. Bu yöntem, kumaş boyama işlemi görmüşse boyanın daha kalıcı olmasını sağlarken, kumaşın pürüzsüzlüğünü ve parlaklığını artırarak istenen uzunluğa erişmesini de sağlamaktadır. Bu makinede kabin sıcaklıklarının ve

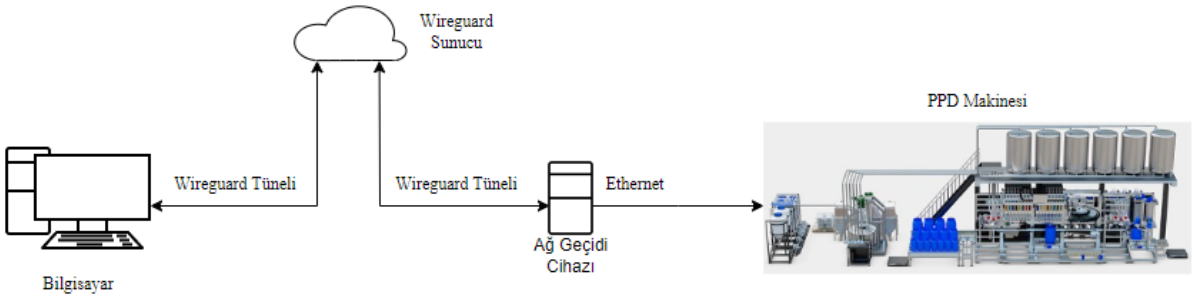
kumaş uzunluklarının belirli değerlerde tutulması gerektiğinden, sıcaklık ve uzunluk verilerine anlık olarak erişilmesi ve izlenebilmesi önemlidir. Makine numarası ve parti numarası gibi filtreleme parametreleriyle geçmiş iş emrindeki sıcaklık ve hız değişimlerinin takibi yapılmaktadır. Geçmiş parti numarasına göre analizler yapılarak, reçetelerin ona göre düzenleme işlemleri yapılabilmektedir. Ayrıca zaman filtresi kullanılarak raporlar oluşturabilmektedir. Bu sistem, bu senaryodaki gibi farklı bölgelerde fabrikası olan şirketler için önem kazanmaktadır. Şekil 4.5'te, Grafana platformunda ram makinesindeki kabinlerin anlık sıcaklıkları, kumaş zincir hızı ve fan hız bilgilerinin veri görselleştirmesine yer verilmiştir.



Şekil 4.5 Grafana Veri Görselleştirilmesi

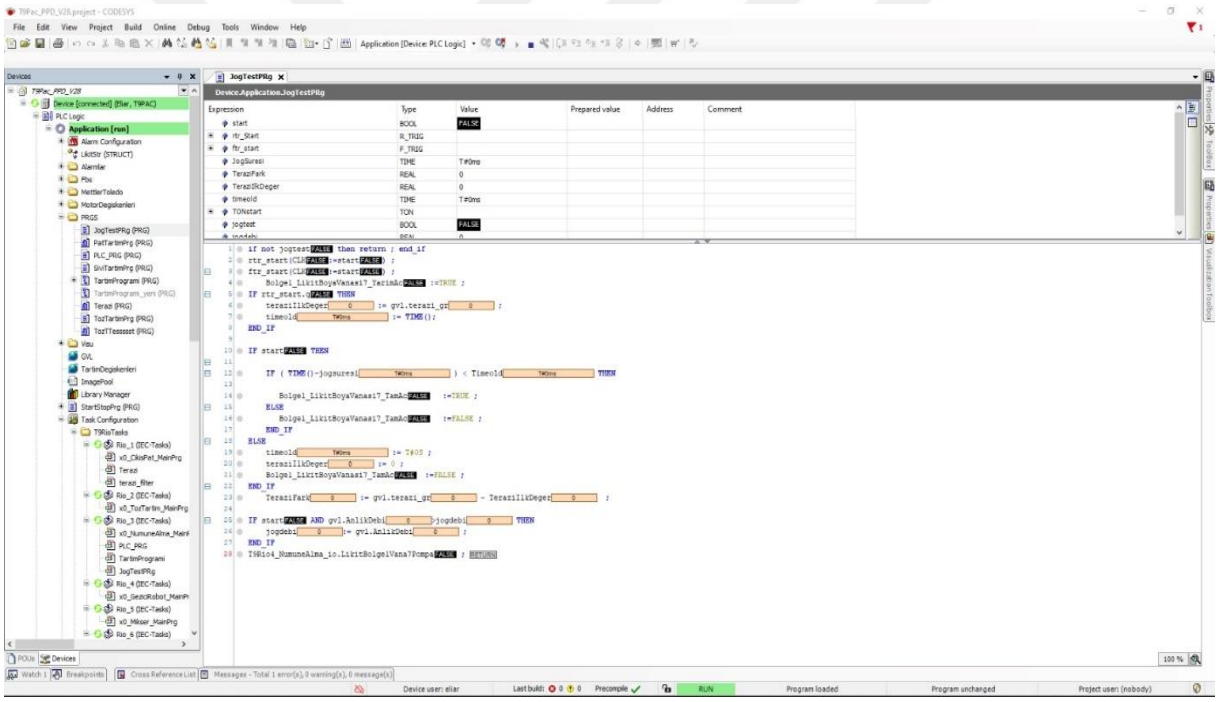
4.3. Senaryo 3: Bir PC üzerinden uzak makinede yazılım güncellemesi

Şekil 4.6'da Gateway cihazının PPD makinesine bağlantısı gösterilmektedir. Bu cihaz, PPD makinesinde bulunan plc cihazına direk bağlantısı yapılmıştır.



Şekil 4.6 Bilgisayar ile PPD Makinesine Uzaktan Erişim Şeması

PPD, tekstil endüstrisinde kullanılan baskı için kimyasalların hazırlanmasını tamamen otomatik hale getirir. Bu tür bir sistem, baskı işlemlerinde kullanılan boyaların, kimyasalların ve diğer bileşenlerin belirli oranlarda karıştırılması ve baskı işlemi için hazır hale getirilmesini sağlar. Kompleks bir yapıya sahip olan PPD makinesi birçok cihaz ile farklı haberleşme kullanarak entegreli çalışmaktadır. OPC UA, Modbus TCP ve EtherCAT haberleşme protokollerine sahip bu cihazların uzaktan konfigüre edilmesi sağlanmış, böylece sistemin daha hızlı bir şekilde kullanıma hazır hale gelmesine olanak tanımıştır. Böylelikle Otomasyon ekibi, makinenin mekanik kısmının üretildiği yere gitmeden birçok çalışmayı uzaktan yapabilmiştir. Şekil 4.7'de gösterilen Codesys tabanlı PLC programı, ağ geçidi cihaz aracılığıyla uzaktan bağlantı kurulmasına, yazılım yüklenmesine ve diğer konfigürasyon ayarlarının yapılmasına olanak sağlamaktadır. Bu senaryo Orijinal Ürün Üretici (OEM) firmalar için önem arz etmektedir.



Şekil 4.7 PPD Makine Plc Yazılımı

Bu uzaktan izleme ve kontrol çözümü, sanayide kullanılan makinelerin sürekli olarak izlenebilmesini ve yönetilmesini olanak sağlar. Sistemin güvenliği ve performansı, WireGuard VPN teknolojisi ile sağlanmaktadır. Bu entegre yaklaşım, modern sanayi ortamı için kapsamlı bir çözüm sunarak gelişmiş teknolojilerin, günümüzün bağlantılı dünyasının taleplerini karşılamasını sağlar.

5. SONUÇLAR

Bu tez çalışması, endüstriyel ortamlarda yer alabilen çeşitli makine ve cihazların uzaktan erişim, izleme ve kontrol ihtiyaçlarını karşılamak üzere güvenli ve yenilikçi bir çözüm sunmayı amaçlamıştır. IoT tabanlı cihazlar için geliştirilen bu sistemde güvenlik, veri bütünlüğü ve erişilebilirlik gibi temel unsurlar göz önünde bulundurulmuştur. Elde edilen sonuçlar, sistemin endüstriyel alanda geniş bir kullanım potansiyeline sahip olduğunu göstermektedir. Çalışmada kullanılan düşük maliyetli donanım bileşenleri ile modern VPN teknolojilerinin entegrasyonu, IoT cihazlarının güvenliğini sağlamak adına esnek ve etkili bir çözüm sunmaktadır. Bu çalışma, özellikle WireGuard VPN'in endüstriyel uygulamalarda kullanımına yönelik detaylı bir inceleme sunmakta olup, düşük gecikme süresi ve yüksek veri transfer hızı ile güvenlik açısından sağladığı avantajları ortaya koymaktadır.

Tez çalışmasında geliştirilen sistemin sahada yapılan uygulamalarla test edilmesi sonucunda güvenlik, verimlilik ve esneklik açısından önemli bulgular elde edilmiştir. Sistem, kullanılan VPN protokolü ile veri iletiminde yüksek güvenlik sağlamaktadır. Bu protokol, şifreleme teknikleri ile izinsiz erişimlerin önüne geçerek endüstriyel cihazların güvenli bir şekilde izlenmesi ve kontrol edilmesini mümkün kılmaktadır. Ayrıca, Node-RED tabanlı kontrol arayüzü, kullanıcıların kablosuz ağ bağlantılarını hızlı bir şekilde yapılandırmasına ve cihazları kolaylıkla yönetmesine olanak tanıyarak endüstriyel ortamlarda yüksek verimlilik sunmaktadır. Sistemin mobil uygulama ile uyumlu olması da, kullanıcıların farklı platformlardan erişim sağlayabilmesini sağlamaktadır. Bu esneklik, sahada hareket halindeyken cihazları yönetmek isteyen kullanıcılar için önemli bir avantaj teşkil etmektedir.

Bu bulgular ışığında, çalışma, hedeflerine ulaşmış ve sistemin sunduğu faydaları net bir şekilde ortaya koymuştur. Gelecek çalışmalarda sistemin ölçeklenebilirliği ve farklı endüstriyel alanlarda kullanılabilirliği üzerine araştırmalar yapılması önerilmektedir. Daha büyük ölçekli endüstriyel tesislerde sistemin performansını artırmak için yeni yazılım optimizasyonları yapılabilir ve gelişmiş veri analitiği teknikleri kullanılarak daha fazla veri toplama ve işleme kapasitesine sahip bir sistem oluşturulabilir.

Tablo 5.1 Endüstride Kullanılan Cihazlar ile Bu Cihazın Karşılaştırılması

Özellik	**Secomea**	**Weidmüller**	**Lenze**	**Bu Cihaz**
Kullanılan Teknoloji	Bilinmiyor	OpenVPN	Bilinmiyor	Wireguard
Konfigürasyon Şekli	USB Stick	Browser	USB Stick	Mobil Uygulama
Mobil Uygulama	Desteği Vardır	Desteği Yoktur	Desteği Yoktur	Desteği Vardır
Fiyat	~530 €	~250 €	~200 €	~70 €*

* : Sistemin maliyetini temsil etmektedir.

Endüstride benzer şekilde çalışacak sistemler bulunmaktadır. Secomea Site Manager, Weidmüller IoT Gateway ve Lenze IoT Gateway bunlardan bazılarıdır. Yapılan sistemin bu cihazlarla karşılaştırılması Tablo 5.1'de belirtilmiştir.

Sonuç olarak, bu çalışma, endüstriyel IoT cihazlarının güvenli ve verimli bir şekilde yönetilmesi için geliştirilmiş yenilikçi bir sistemi ortaya koymaktadır. Geliştirilen sistemin endüstriyel süreçlerde güvenlik ve etkinlik sağlama konularındaki katkıları dikkate alınarak, benzer sistemlerin geliştirilmesine ilham vermesi beklenmektedir.

6. KAYNAKLAR

[1] Mansouri Y.; Prokhorenko V.; Babar M. A., ”An automated implementation of hybrid cloud for performance evaluation of distributed databases”, Journal of Network and Computer Applications, vol. 167, 2020.

[2] Haga S.; Esmaeily A.; Kralevska K.; and Gligoroski D., ”5G Network Slice Isolation with WireGuard and Open Source MANO: A VPNaaS Proof-of-Concept”, IEEE Conference on Network Function Virtualization and Software Defined Networks, pp. 181-187, 2020.

[3] Pudelko M.; Emmerich P.; Gallenmüller S.; Carle G., ”Performance Analysis of VPN Gateways” IFIP Networking Conference and Workshops, pp. 325-333, 2020

[4] Kjorveziroski V.; Bernad C.; Gilly K.; Filiposka S., ”Full-mesh VPN performance evaluation for a secure edge-cloud continuum” Software-Practice and Experience, vol. 54, pp. 1543-1564, 2024.

[5] Paillisse J.; Barcia A.; Lopez A.; Rodriguez-Natal A.; Maino F.; and Cabellos A. ”A Control Plane for WireGuard” International Conference on Computer Communications and Networks (ICCCN), 2021.

[6] Mackey S.; Mihov I.; Nosenko A.; Vega F.; Cheng Y. ” A Performance Comparison of WireGuard and OpenVPN” Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy, Codaspy, pp. 162-164, 2020.

[7] Wu Y.; Shan YW.; Wang ZC.; Zhang PC.; He M.; Liu JH. ”SeWG: Security enhanced WireGuard for Android Based on TEE”, 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1711-1717, 2020.

[8] Arfaoui A.; Kribeche A.; Senouci SM.; Hamdi M. ”Game-Based Adaptive Remote Access VPN for IoT: Application to e-Health” IEEE Global Communications Conference, 2018.

[9] Pañeda, X. G.; Melendi, D.; Corcoba, V.; Pañeda, A. G.; García, R.; García, D. “Forensic Analysis of File Exfiltrations Using AnyDesk, TeamViewer and Chrome Remote Desktop” MDPI Electronics, vol. 13, no. 8, Article 1429, doi:10.3390/electronics13081429, 2024.

[10] Soni, N., Kaur, M., Bhardwaj, V. “A forensic analysis of AnyDesk Remote Access application by using various forensic tools and techniques” Forensic Science International-Digital Investigation, vol. 48, Article 301695, doi:10.1016/j.fsidi.2024.301695, 2024.

[11] Zhang, X. L.; Wu, X. Y.; Zhang, W. X.“Research and Implementation of RDP Proxy Proxy-based Audit System” Proceedings of the International Conference on Computer Networks and Communication Technology (CNCT 2016), vol. 54, 2016.

[12] Masuda, H; Nakanishi, M. “Secure wireless LAN service at a COOP cafeteria”, IEEE Pacific Rim Conference On Communications, Computers, and Signal Processing, doi: 10.1109/PACRIM.2003.1235878, 2003.

[13] Yao SX., Tian XJ., Chen JG., Xiong Y. ” Privacy Preserving distributed smart grid system based on Hyperledger Fabric and Wireguard” International Journal of Network Management, vol. 33 doi: 10.1002/nem.2193, 2023.

[14] Cloudwards, “VPN Protocol Breakdown: A Comprehensive Guide,” [Çevrimiçi]. Erişim adresi: https://www.cloudwards.net/vpn-protocol-breakdown/. Erişim tarihi: 01.07.2023.

[15] Privacy Journal, “VPN Protocols: Detailed Analysis and Comparison,” [Çevrimiçi]. Erişim adresi: https://www.privacyjournal.net/vpn-protocols/. Erişim tarihi: 01.07.2023.

[16] CyberGhost VPN, “VPN Protocols: Features and Benefits,” [Çevrimiçi]. Erişim adresi: https://www.cyberghostvpn.com/features/vpn-protocols. Erişim tarihi: 01.07.2023.

[17] Techopedia, “Best VPN Protocol: A Comprehensive Review,” [Çevrimiçi]. Erişim adresi: https://www.techopedia.com/vpn/best-vpn-protocol. Erişim tarihi: 01.07.2023.

[18] Security.org, “VPN Protocols: A Complete Guide,” [Çevrimiçi]. Erişim adresi: https://www.security.org/vpn/protocols/. Erişim tarihi: 01.07.2023.

[19] Linuxiac, “How to Set Up WireGuard VPN Server on Ubuntu,” [Çevrimiçi]. Erişim adresi: https://linuxiac.com/how-to-set-up-wireguard-vpn-server-on-ubuntu/. Erişim tarihi: 03.07.2024.

[20] Garfatta, I.; Souid, N. E., Klai, K. “Towards Formal Verification of Node RED-Based IoT Applications,” Verification and Evaluation of Computer and Communication Systems, pp. 90-104, doi:10.1007/978-3-031-49737-7_7, 2023.

[21] Diekmann, C.; Hupel, L.; Michaelis, J.; Haslbeck, M.; Carle, G. “Verified iptables Firewall Analysis and Verification”, Journal of Automated Reasoning, vol. 61, issue. 1-4, pp. 191-242. doi:10.1007/s10817-017-9445-1. 2018.

[22] D. Zou, “WGDashboard,” GitHub repository, [Çevrimiçi]. Erişim adresi: https://github.com/donaldzou/WGDashboard. Erişim tarihi: 03.07.2024.

[23] Novac, O. C.; Mureşan, I. I.; Novac, M. C., Ursu, M. P.; Oproescu, M.; Gordan, C.; Bujdoso G. “Development of a Weather Mobile Application in Android Operating System”, In Proceedings of the 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), doi: 10.1109/ECAI46879.2019.9042141, 2019.

[24] Kuzmin, N.; Ignatiev, K.; Grafov, D. “Experience of Developing a Mobile Application Using Flutter”, Information Science and Applications, vol. 621, pp. 571-575, doi: 10.1007/978-981-15-1465-4_56, 2019.

[25] Oblique, “create_ap,” GitHub repository, [Çevrimiçi]. Erişim adresi: https://github.com/oblique/create_ap. Erişim tarihi: 03.07.2024.

[26] Jumakhan H., Mirzaeinia A. “Wireguard: An Efficient Solution for Securing IoT Device Connectivity: Regular Research Paper (CSCI-RTMC)”. IEEE International Conference on Computational Science and Computational Intelligence (CSCI), 2023

ÖZGEÇMİŞ

Adı Soyadı **:** MEHMET BEZENK

Yabancı Dil **:** İngilizce

Öğrenim Bilgileri

Lisans **:** Sakarya Üniversitesi
Elektrik –Elektronik Mühendisliği (GPA : 3,33)

Kariyer

05.2021 – Devam E. Eliar Elektronik A.Ş. - Saha Uygulama ve Otomasyon Mühendisi

01.2019 – 04.2021 Pressan Madeni Eşya A.Ş. - Otomasyon Mühendisi