T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DOKTORA TEZİ

MOTORLU ARAÇLAR İÇİN BLOKZİNCİR TABANLI KİMLİK DOĞRULAMA VE SİCİL KAYIT YÖNETİM SİSTEMİ

Suat ONUR

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Akıllı Ulaşım Sistemleri ve Teknolojileri Doktora Programı

HAZİRAN 2025

T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DOKTORA TEZİ

MOTORLU ARAÇLAR İÇİN BLOKZİNCİR TABANLI KİMLİK DOĞRULAMA VE SİCİL KAYIT YÖNETİM SİSTEMİ

Suat ONUR

DANIŞMAN
Prof. Dr. Mehmet TEKTAŞ
II. DANIŞMAN
Doç. Dr. İlyas ÖZER

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Akıllı Ulaşım Sistemleri ve Teknolojileri Doktora Programı

HAZİRAN 2025

BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
DOKTORA TEZ ÇALIŞMASI
ETİK BEYANI

Bandırma Onyedi Eylül Üniversitesi Lisansüstü Eğitim Enstitüsü tez yazım kılavuzuna göre hazırlamış olduğum "**Motorlu Araçlar İçin Blokzincir Tabanlı Kimlik Doğrulama ve Sicil Kayıt Yönetim Sistemi**" adlı tezimin özgün bir çalışma olduğunu, tez hazırlanırken tüm aşamalarda bilimsel etik ilkelerine uygun davrandığımı, tez kapsamında sunulan tüm verileri bilimsel etik ilkelerine uygun elde ettiğimi, tezde faydalandığım tüm eserlere atıf yaptığımı ve kaynaklar kısmında bu eserleri gösterdiğimi beyan ederim. 30/06/2025

ÖNSÖZ

Uzun yıllar kullanmakta olduğumuz motorlu araçların sahibiyiz, kendi araçlarımızla ilgili yapılan işlemlerin şahidiyiz, ancak yapılan işlemlere ait tutulan kayıtların sahibi değiliz ve istediğimiz her bilgiye istediğimiz zaman kolayca ulaşamıyoruz. Çünkü bilgiler birçok kurumun kendi merkezi bilgi işlem sistemlerinde tutulmakta ve kısıtlı miktarda bilgi paylaşımı yapılabilmektedir. Eğer kullanılmış araç satın aldıysak, aracın geçmişine ait yeterli ve doğru bilgiye ya hiç ulaşamıyoruz ya da zorluk yaşıyoruz. Oysa, sürekli gelişmekte ve değişimlerin öncülüğünü yapmakta olan bilgi iletişim teknolojileri, istenilen bilgilere dünyanın herhangi bir yerinden kolayca ulaşabilmesini sağlamaktadır. İnternet teknolojilerindeki gelişmenin bir ileri boyutu olarak ortaya çıkan blokzincir teknolojisi merkezi yapılanmayı ortadan kaldırarak ortak ve dağıtık bir platform üzerinde kurumların birlikte bilgi paylaşımı ve yönetimi yapabilmesini, tutulan kayıtların değişmezliğini garanti ederek, kurumlar arası iş birliğinin güvenli, güvenilir, şeffaf ve denetlenebilir olmasını sağlamıştır.

2009'dan beri kullanılan Bitcoin'in temelini attığı blockchain teknolojisi, finansal aracılara ihtiyaç duymadan güvenli dijital para transferi sağlamaktadır. Bu teknolojinin evrimi olan Ethereum ve akıllı sözleşmeler, işbirliğini ve dijital/fiziki varlık kayıtlarının güvenliğini yeni bir seviyeye taşımıştır. Bu gelişmelerden ilham alarak, motorlu araçların bir varlık olarak kabul edildiği ve tüm işlem kayıtlarının yönetildiği dağıtık bir uygulama geliştirme fikri bu tezin hazırlanmasında başlangıç noktası olmuştur.

Doktora eğitimi ve tezin hazırlanması süreci boyunca ilk günden itibaren bana destek çıkan, açık sözlülüğü, net duruşu ve engin tecrübesi ve fikirleriyle bana her zaman yardımcı olan Prof. Dr. Mehmet Tektaş hocama saygı ve teşekkürlerimi sunarım.

Tez hazırlık sürecinde öneri ve eleştirileriyle bana yol gösteren sayın Doç. Dr. Ufuk Çelik, Doç. Dr. İlyas Özer ve Doç. Dr. Emrah Dönmez hocalarıma değerli zamanlarını ayırıp bana destek oldukları için çok teşekkür ederim. Tezin simülasyon uygulaması için bilgisayarın kurulumunda ve makale hazırlıklarında bana yardımcı olan sayın Dr. Öğr. Üyesi Caner Pense hocama teşekkür ederim.

Son olarak her zaman yanımda olan, sabırla doktora sürecinin bitmesini bekleyen, manevi destekleriyle bana moral ve güç veren eşim Kadriye, kızlarım Elif ve Begüm’e çok teşekkür ederim.

ÖZET

DOKTORA TEZİ

MOTORLU ARAÇLAR İÇİN BLOKZİNCİR TABANLI KİMLİK DOĞRULAMA VE SİCİL KAYIT YÖNETİM SİSTEMİ

Suat ONUR

Bandırma Onyedi Eylül Üniversitesi
Lisansüstü Eğitim Enstitüsü
Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Danışman: Prof. Dr. Mehmet TEKTAŞ
İkinci Danışman: Doç. Dr. İlyas ÖZER
HAZİRAN 2025, 120 sayfa

Motorlu kara taşıtları sahiplerinin yasal olarak araçlarıyla ilgili yapmak zorunda oldukları birçok yükümlülükleri vardır. Bu yükümlülükler hem diğer sürücülerin ve yayaların güvenliğini sağlamayı hem de devletin düzenleyici işlevini yerine getirmesini amaçlar. Motorlu taşıtlar vergisi, zorunlu araç sigortası, araç muayenesi, egzoz muayenesi, çeşitli tescil kayıtları, köprü, otoyol geçiş ödemeleri, trafik ihlal ve cezaları, kaza ve onarım takibi gibi birçok işlemlerin yapılmasında araç sahipleri sorumludur. Bu işlemler, birçok resmi ve özel kurumun kendi bilgi işlem ve veri yönetim sistemleri üzerinden yapılmaktadır. Mevcut araç veri yönetim sistemleri hem araç sahiplerinin hem de araçla ilgili çeşitli bilgi ve belgelere ihtiyaç duyan kurumların verilere erişimine doğrudan açık değildir. Kurumlar, merkezi yapıda olmaları nedeniyle tek nokta hatası sorunlarıyla karşı karşıya kaldıkları gibi siber saldırılara karşı da savunmasızdır. Ayrıca, araçlarla ilgili verilerin farklı kurumlarda olması, verilere erişimin kısıtlı ve şeffaf olmaması, verilerin suistimale ve manipülasyona açık olması, eksik ve tutarsız verilerin varlığı, çeşitli hile ve dolandırıcılıkların yapılmasına fırsat vermekte, ikinci el araç alım satım süreçlerinde güvensizliğe sebep olmaktadır.

Bu çalışmada, araçlarla ilgili mevcut sorunların çözümüne katkı sağlayabilecek birçok kuruluşun işbirliğinin sağlandığı, güvenli, güvenilir ve şeffaf bir şekilde veri yönetimine imkân veren Blokzincir Tabanlı Araç Kimlik ve Sicil Sistemi (BAKSİS) adında yeni bir modeli öneriyoruz. Bu model ayrıca işlemlere ait belge ve dosyaların güvenli bir şekilde, bütünlüğünün korunarak depolanması için özel IPFS dağıtık dosya depolama sistemine sahiptir. Önerilen model de Türkiye'de araç sicil kayıt ve tescil işlemlerinde rol alan kuruluşların katılımından oluşan özel ve özelleştirilmiş blockchain ağı, Hyperledger Fabric blockchain platformu kullanılarak oluşturulmuştur. Tasarlanan blockchain ağında katılımcı olan kurum ve kuruluşların rollerine ve yetkilerine göre yazılmış akıllı sözleşmeler sayesinde, güvenilir bir iş birliği oluşturularak, araç sicil bilgileri için eksiksiz bir kayıt geçmişi sağlayan değişmez bir defter üzerinde kaydedilmesi sağlanmaktadır. Modelin tek bilgisayarda ve sanal makine uygulaması üzerinde çalıştırılan simülasyon uygulamasında yapılan performans testleri sonucunda sistemin 200tps'lik iş yükü altında saniyede 70tps işlem hacmine sahip olduğunu ve işlemlerin maksimum onay gecikme süresinin 33.07s ulaştığı belirlenmiştir. Sistemde 10 farklı istemci üzerinden kademeli olarak 1000tps ye kadar artırılarak gönderilen okuma-sorgu işlemlerinde %97 oranında verim elde edilmiştir.

Anahtar kelimeler: Araç Yaşam Döngüsü, Araç Veri Yönetimi, Blokzincir, Dağıtık Depolama, Hyperledger Fabric, IPFS.

ABSTRACT

Ph.D. THESIS

BLOCKCHAIN-BASED AUTHENTICATION AND REGISTRATION MANAGEMENT SYSTEM FOR VEHICLES

Suat ONUR

Bandırma Onyedi Eylül University
School of Graduate Studies
Department of Intelligent Transportation Systems and Technologies

Supervisor: Prof. Dr. Mehmet TEKTAŞ
Co-Supervisor: Doç. Dr. İlyas ÖZER
June 2025, 120 pages

Owners of motor vehicles are legally required to fulfill numerous obligations related to their vehicles. These obligations aim to ensure the safety of other drivers and pedestrians as well as to fulfill the regulatory functions of the state. Vehicle owners are responsible for various procedures such as motor vehicle tax, mandatory vehicle insurance, vehicle inspection, exhaust inspection, various registration records, bridge and highway toll payments, traffic violations and fines, and accident and repair tracking. These procedures are carried out through the information processing and data management systems of various public and private institutions. Current vehicle data management systems do not provide direct access to data for both vehicle owners and institutions that require various information and documents related to vehicles. Due to their centralized structure, institutions face single point of failure issues and are vulnerable to cyber attacks. Furthermore, the fact that vehicle-related data is stored in different institutions, access to data is limited and non-transparent, data is susceptible to misuse and manipulation, and the existence of incomplete and inconsistent data provide opportunities for various fraud and deception, causing insecurity in second-hand vehicle purchase and sale processes.

In this study, we propose a new model called the Blockchain-Based Vehicle Identification and Registration System (BAKSİS), which enables secure, reliable, and transparent data management through the collaboration of many organizations that can contribute to solving existing problems related to vehicles. This model also has a special IPFS distributed file storage system for the secure storage of documents and files related to transactions while maintaining their integrity. The proposed model has been created using the Hyperledger Fabric blockchain platform, which is a special and customized blockchain network consisting of organizations involved in vehicle registration and licensing processes in Turkey. Thanks to smart contracts written according to the roles and authorities of the institutions and organizations participating in the designed blockchain network, a reliable collaboration is established, and vehicle registration information is recorded on an immutable ledger that provides a complete record history. Performance tests conducted on a simulation application running on a single computer and a virtual machine application revealed that the system achieved a transaction volume of 70 tps under a workload of 200 tps, with a maximum transaction confirmation delay of 33.07 seconds. In read-query operations sent from 10 different clients, gradually increased up to 1000 tps, a 97% efficiency rate was achieved.

Keywords: Vehicle Lifecycle, Vehicle Data Management, Blockchain, Distributed Storage, Hyperledger Fabric, IPFS.

İÇİNDEKİLER

ŞEKİL LİSTESİ

Şekil **Sayfa**

TABLO LİSTESİ

Tablo **Sayfa**

SİMGE VE KISALTMA LİSTESİ

Simgeler	**Açıklama**
sn	: Saniye
tps	: Saniyedeki İşlem Sayısı

Kısaltmalar	**Açıklama**
API	: Application Programming Interface (Uygulama Programlama Arayüzü)
ARTES	: Araç Tescil Sistemi
BAKSİS	: Blokzincir Tabanlı Araç Kimlik ve Sicil Sistemi
BFT	: Byzantine Fault Tolerance (Bizans Hata Toleransı)
CA	: Certificate Authority (Sertifika Otoritesi)
CFT	: Crash Fault Tolerance (Çökme Hata Toleransı)
CID	: Content Identifier (İçerik Tanımlayıcı)
CLI	: Command Line Interface (Komut Satırı Arayüzü)
DLT	: Distributed Ledger Technology (Dağıtık Defter Teknolojisi)
DHT	: Distributed Hash Table (Dağıtık Karma/Özet Tablo)
EVM	: Ethereum Virtiual Machine (Ethereum Sanal Makinesi)
Hash	: Özet, Karma Değer
HGS	: Hızlı Geçiş Sistemi
HLF	: HyperLedger Fabric
IPFS	: InterPlanetary File System (Gezegenler Arası Dosya Sistemi
JSON	: JavaScript Object Notation (Javascript Nesne Notasyonu)
MSP	: Membership Service Provider (Üyelik Servis Sağlayıcısı)
MTV	: Motorlu Taşıtlar Vergisi
ORG	: Organization (Kuruluş)
P2P	: Peer to Peer (Eşten Eşe Ağ)
pBFT	: Practical Byzantine Fault Tolerance (Pratik Bizans Hata Toleransı)
PKI	: Public Key Infrastructure (Açık Anahtar Altyapısı)
PoS	: Proof of Stake (Hissenin İspatı)
PoW	: Proof of Work (İşin İspatı)

SDK	: Software Development Kit (Yazılım Geliştirme Kiti)
SHA	: Secure Hash Algorithm (Güvenli Özet Algoritması)
SPOF	: Single Point of Failure (Tek Hata Noktası)
TLS	: Transport Layer Security (Taşıma Katmanı Güvenliği)
TPS	: Transactions Per Second (Saniyedeki İşlem Sayısı)
TÜİK	: Türkiye İstatistik Kurumu
URL	: Uniform Resource Locator (Tekdüze Kaynak Konumu)
UTTS	: Ulusal Taşıt Tanıma Sistemi
VIN	: Vehicle Identification Number (Araç Kimlik Numarası)

1. GİRİŞ

Bireysel hareketliliğin ve ticari faaliyetlerin vazgeçilmez bir parçası olan motorlu kara taşıtlarının (araç) sayısı, nüfus artışı, kentleşme, ekonomik büyüme, refah seviyesinin yükselmesi ve araç sahibi olmanın kolaylaşması gibi nedenlerle hem dünya genelinde hem de Türkiye'de yıllar içerisinde sürekli bir artış göstermektedir. Türkiye İstatistik Kurumu (TÜİK) 2025 Mayıs ayı sonu verilerine göre, toplam kayıtlı motorlu taşıt sayısı 32 milyonu aşmıştır [1]. 2010 yılından itibaren her yıl ortalama iki milyona yakın yeni araç tescil kaydının yapıldığı, araç mülkiyet devir işlemlerinin ise özellikle son yıllarda 10 milyonun üzerinde gerçekleştiği görülmektedir. Araç sayılarının düzenli bir şekilde artış göstermesi, araçlarla ilgili yapılan işlem hacmini de önemli ölçüde artırmakta, bazı zorluk ve problemleri beraberinde getirmektedir.

Dünyada birçok ülkede ve Türkiye'de, araç sahiplerinin uzun yıllar kullandıkları araçlarıyla ilgili bazı sorumlulukları düzenli olarak yerine getirmeleri zorunludur. Bunlar; Araç muayenesi, trafik sigortası, Motorlu Taşıtlar Vergisi (MTV), egzoz gazı emisyon ölçümü, trafik ihlal cezaları, köprü otoyol otomatik geçiş ödemeleri (HGS, OGS vb.), çeşitli araç tescilleri, ruhsat ve plaka bilgi güncellemeleri gibi işlemlerdir. Yeni nesil elektrikli, bağlantılı, akıllı ve otonom araçların daha özel hizmetlere ait çeşitli işlemleri de bulunmaktadır. Bu nedenle, araçlarla ilgili gerçekleştirilen çeşitli işlem kayıtlarının uzun bir süre güvenle saklanması, takip ve yönetiminin verimli bir şekilde yapılması daha önemli hale gelmektedir.

Araçlarla ilgili tutulması gereken kayıtlar sadece periyodik olarak tekrarlanan ve zorunlu olarak yapılması gereken işlemlerle sınırlı değildir. Araçlarla ilgili işlemlerde rol alan resmi ve özel birçok kurum kendi hizmet alanlarına göre araçlara ait; üretim bilgileri, mülkiyet devri ve tescil yenileme, teknik muayene kayıtları, kullanım durumları, sigorta ve kasko bilgileri, bakım onarım kayıtları, akaryakıt tüketim veya batarya kullanım takibi, kaza ve hasar kayıtları, trafik ihlal ve ceza kayıtları, vergi ve çeşitli ödemeler gibi birçok konu ve kapsamda verilerin saklanması ve yönetilmesini sağlamaktadır. Devletler ve kurumlar araçlar üzerinde çeşitli nedenlerle daha fazla kontrol ve denetim sağlamak istedikleri için, araçlarla ilgili yapılan işlem çeşitliği ve kayıt altına alınması gereken veri miktarı da son zamanlarda önemli oranda artmıştır.

Birçok resmi ve özel kurum kendi hizmet ve amaçlarına göre, araçlarla ilgili gerçekleştirilen çeşitli işlemler için kendi merkezi yapıdaki veri takip ve yönetim sistemlerini kullanmaktadır. Ayrıca, mevcut durumda kurumların kendi merkezi sistemleri üzerinden işlemleri gerçekleştirmesi, kurumlar arasındaki güvensizlikler, kişisel verilerin gizliğini sağlama, siber güvenlik konusunda sorumluluktan kaçınma gibi nedenlerle, bilgi alışverişi ve paylaşımının kısıtlı

olduğu görülmektedir [2]. Ortak paydası araçlar olan bu kuruluşlar için araçların doğru tanımlanması, kimliklendirilmesi ve araç verilerinin tutarlı ve güvenilir olması oldukça önemlidir. Her kuruluş kendi sorumluluk alanındaki verilerin yönetimiyle ilgilendiği için dijital ortamlarda kurumlar arası bilgi paylaşımı tam anlamıyla yapılamamakta bu da kâğıt tabanlı belge kullanımının devam etmesine neden olmaktadır. Örneğin Noterler üzerinden gerçekleştirilen araç alım-satım ve tescil işlemleri için, halen çeşitli kâğıt tabanlı ve ıslak imzalı belge ve bilgiler istenerek, manuel doğrulama ve kontroller yapılarak işlemler tamamlanabilmektedir [3].

Akıllı Ulaşım Sistemleri, disiplinler arası iş birliğiyle dijital teknolojileri aktif bir şekilde kullanarak araçlarla ilgili verilerin kayıt edilmesi ve yönetilmesini sağlayan yeni teknolojileri geliştirmekle birlikte, kullanıcı-araç-altyapı-merkez arasındaki veri alışverişini yöneten, takip ve kontrolünü sağlayan sistemlerin geliştirilmesinde öncülük etmektedir. Araçlarla ilgili ve bağlantılı birçok verinin farklı kurumlar tarafından kaydedilmesi, paylaşılması ve yönetilmesi yerine tek bir merkezi kurum tarafından yönetilmesi, takip ve kontrollerinin sağlanması gerektiğini vurgulayan önemli çalışmalar yapılmıştır. Tektaş vd. [4] tarafından önerilen çalışmada Ulaşım Kontrol Merkezinin kurularak ulaşımdaki her türlü birimlerden gelecek verilerin yönetilmesi ve analiz edilmesi için merkezi bir sistem oluşturulmasının önemine işaret edilmektedir. Benzer bir başka çalışmada, Mahmutoğlu vd. [5] "Trafik Sorununa Bir Çözüm Önerisi: Trafik İzleme Başkanlığı" başlıklı makalesinde, bilişim teknolojilerini kullanarak, Trafik İzleme Başkanlığı kurumu adında merkezi bir sistem oluşturularak araçlarla ilgili her türlü verinin işlenmesi ve yönetilmesinin gerektiği ve bu sayede tek merkezde toplanan verilerin analizi ile birçok probleme çözüm bulunabileceği detaylı bir şekilde ele alınmaktadır.

Kurumlar arası iş birliği genelde güvenilir üçüncü bir tarafla sağlanır. Her türlü bilgi üçüncü tarafın gözetiminde kanıt oluşturacak şekilde paylaşılır. Bu durum aynı zamanda merkezi bir yapı oluşmasına da neden olmaktadır. E-devlet uygulamaları, çok sayıda kurum arasında iş birliği ve güvenin sağlanması, belge ve bilgilerin doğrulanması vb. konularda güvenilen üçüncü bir taraf rolündedir. Ancak bu durum, birçok süreçte tek hata noktası riskleri ve yoğun işlem koşullarında darboğaz oluşturmaktadır. Kurumsal bilgi paylaşımında kullanılan ağ sistemi, genellikle sunucu-istemci bağlantı modelinde çalışan merkezi bir sistemdir ve bilgiye erişim bir bağlantı (URL, Evrensel Kaynak Konumu) aracılığıyla İnternet üzerindeki tek sunucu tarafından sağlanır ve kontrol edilir. Bu nedenle, merkezi yapılarda tutulan veriler tek noktadan kaynaklanan arızalara karşı hassastır. Çeşitli siber saldırıların, özellikle hizmet reddi saldırılarının potansiyel bir hedefidir [6]. Kurumların merkezi sistemiyle ilişkili bir diğer önemli risk ise kurum personelinin yetkilerini kötüye kullanma ve verileri manipüle etme potansiyelidir.

Uzun yıllar kullanımda olan araçlarla ilgili kayıt altına alınması, izlenmesi ve yönetilmesi gereken bilgi çeşitliğinin çokluğuyla birlikte, bu bilgileri kullanarak hizmet veren bir çok resmi ve özel kurumun birbirinden bağımsız merkezi veri yönetim sistemlerine sahip olması, kurumlar arası veri paylaşımında ve doğrulanmasında yaşanan zorluklar, kayıtlar arasındaki tutarsızlıklar birçok problemin oluşmasına zemin hazırlamaktadır. Bu problemler içerisinde özellikle, ikinci el araç piyasasında aracın geçmişine ait bilgilerin öğrenilmesinde zorluklar, aracın daha önceki kaza bilgileri ile bakım-onarım kayıtlarının yetersizliği, sigorta dolandırıcılıkları, hileli işlemler ve veri manipülasyonu gibi sorunlar ortaya çıkmaktadır. Bu sorunlar nedeniyle otomobil sektörünün birçok hizmet alanındaki taraflar arasında güvensizlik hakimdir.

Literatürde yapılan çalışmalar incelendiğinde, çoğunlukla ele alınan problemlerin; ikinci el araç pazarında yaşan güvensizliklerin temel nedeni olarak araca ait geçmiş kayıtların olmaması yada yetersizliği [7], [8], araçların bakım onarım kayıtlarına ve kaza geçmişlerine erişimdeki zorluklar [9], [10], hurdaya ayrılması gereken ağır hasarlı araçların yolsuzluk yapılarak tekrar satışa çıkarılması [11], merkezi yapıda ve birbirinden bağımsız çalışan araç tescil kuruluşlarının kağıt tabanlı belgelerle sahte tescil işlemlerinin yapılma olasılığının olması, şasi ve motor numarası değiştirilen çalıntı araçların yeniden tescil edilip satılması [2], [12], araçların kilometre sayacı üzerinde yapılan veri manipülasyonları ve sahtekarlıklar [13], [14], [15], araçlarla ilgili işlemlerde kurumlar arasındaki yetersiz veri paylaşımı, veri kalitesi, verilerin güvenle saklanması, güvenirliği konusundaki sorunlar [16], araç sigorta işlemleri üzerinde yapılan dolandırıcılıklar [17], [18] gibi konular üzerinde yoğunlaştığı görülmektedir. Bu problemlerin ortaya çıkmasındaki en önemli etkenler; çok sayıda kuruluşun kendi merkezi veri yönetim sistemlerini kullanması nedeniyle tek nokta hatası sorunlarının olması, şeffaflığın olmaması, bütünlüğü korunmuş doğru ve detaylı kayıtların tutulmasındaki zorluklar, kurumlar arası güvensizlikler nedeniyle veri paylaşımının çok kısıtlı ya da hiç olmaması, verilere erişimin zorluğu, kayıt altına alınan verilere olan güvensizlikler olarak sıralanmaktadır.

Araçlarla ilgili kapsamlı kayıtların tutulmasında özel olarak tanımlanmış benzersiz Araç Kimlik Numaraları (VIN) ile motor ve şasi numaralarının önemi büyüktür. Araç kimlik bilgilerinin, araçların yaşam döngüsü boyunca izlenebilmesi, çalıntı araçların tespitinde, hurdaya ayrılmış araçların onarılıp sahte bilgi ve belgelerle manipülasyon yapılarak yeniden tescil edilmelerinin engellenmesinde önemli bir rolü vardır. Bilgi işlem sistemlerinde, araçla ilgili tüm bilgi, belge ve kanıtlara erişmede araç kimlik numara ve bilgileri kullanılmaktadır. Ancak görevini suistimal eden bir kurum çalışanının merkezi yapıdaki veritabanı üzerinde veri manipülasyonu yapabilme riski ve potansiyeli önemli bir sorundur [19].

Literatürde, yukarıda bahsi geçen problemlerin çözümü için, merkeziyetsiz, dağıtık veri kaydı, şeffaflık, değiştirilemezlik, akıllı sözleşme ve mutabakat mekanizmalarıyla birbirine güvenmeyen taraflar arasında iş birliğini mümkün hale getiren özellikleri ile dikkatleri üzerine çeken blokzincir teknolojinin kullanımını öneren birçok çalışma yapılmaktadır. Ancak yapılan çalışmaların birçoğunda, ortaya çıkan problemlerin ayrı ayrı ele alınarak, sadece probleme özgü blokzincir teknolojisinin kullanımına yönelik çözümler önerildiği görülmektedir. Bu tez çalışmasında problemlerin büyük çoğunluğunu ele alan kapsamlı bir blokzincir tabanlı çerçeve tasarlanmış, kurumların ortak bir platform üzerinden veri yönetimini yapabilmelerine imkân sağlanmıştır.

Blokzincir teknolojisi, birbirine güvenmeyen taraflar arasındaki işlemlerde güvenilir bir aracıya olan ihtiyacı ortadan kaldıran, merkeziyetsiz bir ortamda mutabakat içinde iş birliği yapılmasına imkân veren, mevcut iş modelleri ve iş akış yöntemlerini kökten değiştirebilecek yeni bir teknoloji olarak görülmektedir. Blokzincir teknolojisi başlangıçta kripto para ve finans sektöründe aracı banka ya da kurum olmaksızın para transferlerinde ve ödemelerde kullanılmış, akıllı sözleşme ve mutabakat yöntemlerinde yapılan geliştirmelerle birçok sektörde kullanılabilecek potansiyelde olduğu görülmüştür. Son zamanlarda kimlik ve sertifika yönetimi, gayrimenkul, tedarik zinciri, muhasebe denetimi, pazarlama, turizm, enerji, sağlık, otomotiv, tarım, akıllı şehir, akıllı ulaşım ve kamusal hizmetler gibi daha birçok alanda kullanılabileceğine yönelik öneriler ve uygulamalar geliştirilmiştir [20]. Sektörlerin iş süreçlerinde iyileştirmelerin yapılmasında ve problemlerinin çözülmesinde etkili olacağı düşünülen ve halen geliştirilmeye devam eden blokzincir teknolojisi, merkezi sistemlere göre daha fazla güvenlik ve güvenilirlik sağlamakta, hesap verebilirlik, değişmezlik, yedeklilik, şeffaflık gibi temel özellikleriyle daha fazla benimsenmektedir. Böylece maliyetlerin düşürülmesi, işlemlerin daha kısa zamanda tamamlanması, verimliğin artırılması, hile ve dolandırıcılıkların engellenmesi ve siber güvenlik konularında önemli iyileştirilmeler sağlama potansiyelinde olduğu için birçok sektörde blokzincir teknolojisinin kullanılabilirliği araştırılmakta ve önemli yatırımlar yapılmaktadır.

Bu tez çalışmasında, akıllı ulaşım sistemlerinin önemli unsurlarından biri olan araç veri yönetiminde, blokzincir teknolojisinin potansiyel faydaları araştırılmış, verilerin güvenli bir şekilde depolanması, birçok kurumun iş birliğiyle veri paylaşımı ve yönetimini sağlayan ortak bir platformun modellenmesi ve uygulaması gerçekleştirilmiştir.

1.1 TEZİN AMACI VE KAPSAMI

Bu tez çalışmasındaki amacımız; motorlu araçların yaşam döngüsü içerisinde gerçekleştirilen işlemlere ait gerekli bilgi ve belgelerin, silinemez, değiştirilemez ve güvenli bir şekilde, hurdaya ayrılıp kullanım dışı kalana kadar uzun bir süre boyunca kayıt altına alınmasını, bu bilgilere ihtiyaç duyan özel ve resmî kurumlar ile araç sahipleri tarafından kolayca erişilebilmesini sağlayacak, blokzincir teknolojisinin kullanıldığı bir platform oluşturmaktır. Bu platform, araçlarla ilgili çeşitli işlemlerde rol alan özel ve resmî kurumlar arasında iş birliğinin daha güvenli ve güvenilir olmasını ve çevrimiçi blokzincir ağı içerisinde bilgi ve belgelerin kronolojik olarak tahrif edilemez, değiştirilemez, şeffaf ve denetlenebilir olmasını sağlar. Ayrıca, aracın geçmişine ait değiştirilemez kayıtların ve kanıtların varlığı, tasarlanan sistemin mahremiyeti ve gizliliği koruyan yapısı, verilerin kaydı ve erişiminde izne ve yetkiye göre kriptografik yöntemlerle kontrol sağlanması sayesinde, kullanılmış araçların alım satım süreçlerindeki güvensizliklerin ortadan kaldırılmasına, çeşitli konularda denetim ve takip işlemlerinin kolaylaşmasına, olası sahtecilik ve dolandırıcılık gibi problemlerin engellenmesine katkı sağlar.

1.1.1 Hedefler ve Katkılar

- Araçlara ait, şasi, motor ve tescil tarihi bilgilerinden üretilecek hash kodu ile benzersiz bir araç kimlik numarası (VIN) tanımlayarak, araçların yaşam döngüsü boyunca gerçekleştirilen sınıflandırılmış işlem bilgilerinin araç kimlik numarası ile ilişkilendirilerek kayıt altına alınmasını sağlamak.
- Araç sahipleri ile birlikte, ilgili resmi ve özel kuruluşların araçlarla ilgili işlemleri, blokzincir tabanlı ortak bir platform üzerinde, tanımlanan yetki ve rollerine göre şeffaf, güvenli ve denetlenebilir bir şekilde iş birliği yapabilmelerine imkân sağlamak.
- İzinli Hyperledger Fabric blokzincir platformunun yapısal ve yönetimsel özellikleriyle, sonradan ihtiyaç duyulan hizmetler için yeni düğümler, kanal ve kayıt defterleriyle akıllı sözleşmeler tasarlanarak mevcut yapıyı bozmadan ölçeklenebilirliği sağlamak.
- Araçlarla ilgili yapılan işlemlerde kullanılan ve kanıt hükmünde olan kağıt tabanlı belge, resim ve video formatındaki bilgilerin özel ve izinli IPFS dağıtık dosya sisteminde bütünlük koruması, erişim kontrolü ve gerektiğinde şifrelenerek depolanmasını sağlamak.

1.1.2 Tezin sonraki bölümleri ve organizasyonu

Bu tez çalışmasında sonraki bölümler aşağıda belirtilen konular ile organize edilmiştir:

- Bölüm 1'de araçların yaşam döngüsü içerisinde kayıt altına alınması gereken veri çeşitliliği, mevcut araç veri yönetimi sistemlerinde problemli konuların neler olduğu incelenmiş, blokzincir teknolojisinin kullanıldığı çözüm önerisinin amacı, kapsamı ve katkılarından bahsedilmiştir.
- Bölüm 2'de tez çalışmasının alt yapısını oluşturan, blokzincir teknolojisinin mimarisi, bileşenleri, türleri ve temel özellikleri kavramsal olarak açıklanmış, blokzincir teknolojisinin kullanım alanları ve sağladığı faydalardan bahsedilmiştir.
- Bölüm 3'de önerilen BAKSİS blokzincir ağ modeli için kullanılan Hyperledger Fabric blokzincir platformunun mimari yapısı ve yapılandırılması ile temel özellikleri hakkında detaylı bilgiler verilmiş, önerilen modelde kullanılan ve geliştirilen akıllı sözleşmelerin algoritması ile kullanıcı arayüzlerinin yapısı ve işlevleri anlatılmıştır.
- Bölüm 4'de BAKSİS modeli için hazırlanmış simülasyon uygulamasının kullanıcı ekran arayüzleri
- Bölüm 5'de önerilen modelin Hyperledger Fabric blokzincir platformu üzerindeki uygulaması üzerinde, Hyperledger Caliper test aracı ile performans ölçümleri yapılarak sonuçlar değerlendirilmiştir.
- Bölüm 6'de sonuçlar, öneriler ve tasarlanan modelin ve uygulamanın kısıtları verilmiştir.

1.2 LİTERATÜR TARAMASI

1.2.1 Blokzincir teknolojisinin kullanıldığı araç sigortası konusunda yapılan çalışmalar

Demir vd. [21] araç sigorta kayıtlarının şeffaf ve detaylı bir şekilde paylaşılması için blokzincir teknolojisinin kullanımının önerildiği çalışmalarında sürücüler, devlet kurumları ve sigorta şirketleri gibi taraflar arasında, blokzincirin bir iletişim aracı olarak kullanılabileceği, bilgi ve belgelerde önemli olan şeffaflığın sağlanabileceği gösterilmektedir. Ayrıca blokzincir teknolojisinin özellikleri ve kısıtları değerlendirilmektedir.

Yadav vd. [22] kazaların raporlanmasın, önemli ölçüde insan çabası gerektiren sigorta işlemlerinin kolaylaştırılması, taraflar arasındaki güven ve güvenlik endişelerinin giderilmesi için Hyperledger Fabric blokzincir platformunun kullanıldığı, sigorta talepleri, kaza ve onarım

süreçlerinin yönetildiği bir çerçeve önermişlerdir. Blokzincir teknolojisinin kullanımının hesap verebilirliği artırdığı ve riskleri azalttığının vurgulandığı çalışmalarında, eksiklik olarak blokzincir teknolojisinin ölçeklenebilirlik problemlerine sahip olduğu, daha fazla işlem yükü altında kaldıkça blokzincir kayıt defterinin daha büyük hale geleceği, bu nedenle veri arama işlemlerinin daha verimli hale getirilmesi gerektiği önerilmiştir.

Dorri vd. (Dorri vd., 2017) bağlantılı akıllı araçlar üzerinde uygulanan çeşitli hizmetlerin, erişim kontrolünün, veri iletişim ve depolama güvenliğinin sağlanmasında, olası saldırı senaryolarına karşı önlem alınmasında blokzincir teknolojisinin kullanımının önerildiği çalışmada sigorta şirketlerinin bir kaza durumunda veriler üzerinde manipülasyon yapılma riskine karşı araca ait verilerin blokzincir teknolojisi sayesinde güvenli, şeffaf ve güvenilir bir şekilde alınabileceği vurgulanmaktadır.

Roriz vd. [18] sigorta şirketlerinin merkezi yapılarından kaynaklanan belirli dolandırıcılık türlerine karşı blokzincir kullanımının kurumsal işlemleri iyileştirme potansiyelini araştırdıkları çalışmalarında Solidity dili ile geliştirildikleri akıllı sözleşmelerle paydaşları sigorta şirketleri olan Ethereum tabanlı bir uygulama geliştirmişlerdir. Çifte sigorta dolandırıcılıkların engellenmesi için araç sigorta bilgilerinin ortak bir platformda paylaşılmasını sağladıkları pilot uygulamayı test edip çalıştığını kanıtlamışlardır.

Bader vd. [23] yaptıkları çalışmalarında, araç sigorta taleplerinin ve sigorta olaylarının işlenmesinde kurcalamaya karşı dayanıklı sensörlere dayalı otomatik hasar tespitinden faydalandıkları, böylece müşteri ve sigorta şirketi arasındaki güven eksikliğinin giderildiği, maliyet verimliliği, veri güvenliği ve müşteri gizliliği sağlamak için Ethereum tabanlı akıllı sözleşmelerle yönetilen CAIPY isimli bir blokzincir çerçevesi önermişlerdir.

Oham vd. [24] yaptıkları çalışmasında bağlantılı ve otomatik araçlar için bölümlenmiş ve izinli blokzincir tabanlı Otomatik Sigorta Talepleri ve Karaları (B-FICA) ismini verdikleri bir çerçeve geliştirmişlerdir. Sistemin güvenliğini tehlikeye atabilecek birçok güvenlik saldırısına karşı dirençli olan sistem, sensör verilerinin ve diğer varlıklar arasındaki etkileşimlerin dijital olarak imzalanması ile yürütme, uygunluk ve güvenilirlik kanıtı sağlamaktadır. Bu kanıtların tahrif edilmesini engellemek için dinamik ve sade bir mutabakat ve doğrulama mekanizması kullanmaktadır.

Hombalimath vd. [25], blokzincir teknolojisinin araç sigorta işlemlerinde kullanımının hangi yönlerden faydalı ve gerekli olduğunun sunulduğu çalışmalarında, otomatik hasar ödemeleri, müşterilerin doğrulanabilir kimliği, sahte bilgi ve belgelerle yapılan taleplerin

belirlenmesi ve engellenmesi, maliyetlerin düşürülmesi, şeffaflığın ve izlenebilirliğin artırılması gibi konularda detaylı bilgi verilmektedir.

Xiao vd. [26] çalışmalarında, Nesnelerin İnterneti Teknolojisinin (IoT) kullanılarak araç yolculuğu ve sürücü davranışına ait bilgilere göre sigorta prim ödemelerinde dinamik fiyatlandırma yapılabilmesi için hem izinli (Hyperledger) hem de izinsiz ve halka açık (Ethereum) blokzincirlerin koordineli bir şekilde kullanıldığı bir uygulama önerilmiştir. Dinamik sigorta prim fiyatlandırmasında kullanılacak verilerin IoT üzerinden gönderim performansı için çeşitli testler yapılmıştır. Amazon Web Servisi bulut ortamında oluşturulan prototip üzerinde verim analizi, gecikme süreleri ve kaynak kullanımı açısından kapsamlı bir değerlendirme yapılarak, ortalama 1000 aracın günlük yolculukları için veri gönderim sıklığı saate 950 veri gönderimi sağladığı belirlenmiş ve sistemin kapasitesinin ise saatte 1.260.000 yolculuk veri gönderimini yönetebilir olduğu belirtilmiştir.

Nizamuddin vd. [27] yaptıkları çalışmada, sigorta sektörünün veri toplama, veri yönetimi, bütünlük, güvenlik ve genel altyapısını iyileştirmek, araç sigortası sektörünün işleyişini, hasar yönetimini, ödeme düzenlemesini ve hasar onaylarını düzenlemeye yardımcı olmak için Ethereum blockchain destekli bir çözüm önermişlerdir. Büyük verilerin ve dosyaların depolanması için dağıtık (gezegenler arası) dosya depolama sistemi (IPFS) faydalanılmıştır. Önerdikleri blokzincir sisteminin IPFS ile birleştirilmesi sayesinde tüm belge ve işlemlerin tahrifata karşı dayanıklı, doğrulanabilir olduğu ve araç sigorta işlemlerinde güveni artırdığı, şeffaflık sağladığı, veri ve kimlik yönetimini geliştirdiği sonucuna varılmıştır.

Vo vd. [28] sürücülerinin sadece seyahat ettikleri süreler için, kiralık araç kullanan sürücülerin kullandıkları kadar sigorta primi ödemelerine imkan veren kurcalamaya karşı dayanıklı, güvenli ve şeffaf mikro sigorta veri yönetimi ve analizi için blokzincir tabanlı bir çözüm sunmuşlardır.

Alessandria vd. [17] araç kazalarından sonra, kazayı aydınlatmada fayda sağlayacak bilgilerin hızlı bir şekilde veri manipülasyonuna maruz kalmadan aktarmanın ve depolamanın çözümü olarak blokzincir teknolojisinin kullanımını araştırmışlardır. Aracın kara kutusu ve sensörleri ile belirlenen tüm hassas verilerin öz egemen kimlik doğrulama yöntemiyle (SSI, Self Sovereign Identity) blokzincir üzerinde kaydedilmesini sağlayarak olası dolandırıcılıkları azaltmayı amaçlayan bir model önermişlerdir.

Qi vd. [29], araçlarda kullanıma dayalı sigorta (UBI, Used-based Insurance) primi hesaplaması yapılarak sigorta ödemelerinin daha ekonomik ve kişiye özel olmasını sağlamak için araç sürüş verilerinin güvenliğine odaklandıkları çalışmasında, blokzincir ve sıfır bilgi kanıtı

teknolojilerini kullanarak periyodik olarak sürüş verilerinin akıllı sözleşmeler tarafından değerlendirilerek kayıt altına alınması, doğrulanması ve hesaplanan istatistiklerden sigorta priminin hesaplanmasını sağlayan, DUBI adını verdikleri Ethereum tabanlı bir uygulama geliştirmişlerdir. Benzer çalışmalarla yapılan kıyaslamalarda, kanıt oluşturma hesapları için 7ms ve depolama işlemi için 30ms süreli gecikmelerle benzer önerilerden yedi kat daha verimli olduğu sonucuna ulaşmışlardır.

1.2.2 Kullanılmış araçlar ve araç yaşam döngüsü üzerinde yapılan çalışmalar

Dayı vd. [10] Türkiye'de kullanılmış araçların fiyatları üzerinde etkili olan mikro faktörleri incelediği çalışmada, 6262 adet ikinci el araç üzerinde çeşitli bağımsız değişkenlerle lojistik ve çoklu regresyon yöntemleriyle analizler gerçekleştirmiş ikinci el otomobil fiyatları üzerinde marka ve çekiş türünün fiyatların belirlenmesi üzerinde daha fazla etkili olduğu, ek donanım, mülkiyet değişim sayları ve hasar durumunu gibi faktörlerin fiyatlar üzerinde etkili olmadığı ancak Türkiye'de ikinci el araç piyasasında, alıcı ile satıcı arasında bilgilerin gizlendiği ve asimetrik bilginin var olduğu belirlemiştir. İkinci el araç piyasasının suistimale açık olduğu, bu nedenle güven unsurunun önemli bir faktör olduğunu belirtmektedir. Özellikle alıcı ve satıcı arasında araca ait bilgilerin ve kusurların paylaşımı üzerinde eksik bilgilendirme, gizleme, yanıltıcı beyanda bulunma gibi aracın değeri üzerinde satılması ve satın alım kararı üzerinde etkili olabilecek faktörlerin olması, belirsizliklerin, güvensizliklerin temel sebebi olduğu vurgulanmıştır.

Tanrıverdi [30] yaptığı çalışmada, ikinci araç satışlarında önemli olan, aracın önceki teknik servis bakım kayıtlarının dağıtık ve şeffaf blokzincir ağı üzerinde saklanması ve erişim yönetimi için MultiChain blokzincir platformu üzerinde Araç Cüzdanı uygulaması geliştirilmiştir. Özel servislerin her birinin blokzincir ağında verilerin toplanması, depolanması ve denetlenmesini sağlama görevinde olduğu uygulamada her araç için bir motor şasi numarası kimlik olarak kabul edilerek cüzdan oluşturulmakta ve verilerin bu cüzdan hesabı ile bağlantılı olması sağlanarak depolanmaktadır. Böylece ikinci el araç alım satım işlemlerinde aracın geçmişine ait bilgiye erişimde yaşanan zorluklara karşı çözüm sunulmaktadır.

Jiang vd. [31] yaptıkları çalışmada, Tayvan'daki kullanılmış araç piyasasında güvenilir bir araç veri kaynağı oluşturmak için Go-Etehereum istemci uygulaması ile özel bir blokzincir platformu üzerinde uygulama geliştirmişlerdir. Aracın durumu ile ilgili bilgilerin girilmesi ve güncellenmesi için sadece güvenilir devlet kurumu ve bakım tesislerine yetki verilmiş ve ikinci el araç alım satım işlemlerinde sorun olan güvensiz bilgilerin önüne geçilmesi sağlanmıştır.

Oluşturulan arayüz üzerinden müşterilerin araç bilgilerini ücretsiz bir şekilde sorgulaması sağlanarak şeffaflık ve güven oluşturulmuştur.

Abhale vd. [32] yaptıkları çalışmada, Hindistan'da ikinci el araç pazarının güvensizliğine vurgu yaparak, çalıntı araçların kolay bir şekilde tekrar pazara sunulması nedeniyle araç satın alanların sıklıkla yasal zorluklarla karşılaştıklarını, aracın geçmişine ait hiçbir bilgiye erişilememesinin önemli bir sorun olduğu belirtmişlerdir. Çözüm olarak araçla ilgili bilgileri arayabilecekleri blokzincir tabanlı Android uygulaması geliştirmişlerdir.

Baunmann vd. [7] yaptıkları çalışmada, araçların geçmişlerine ait bilgilerin şefaf olmasının ikinci el araç satıcıları üzerindeki etkisini anket ve görüşmeler yaparak ve SWOT analizleri değerlendirmiştir. Sonuçlar, güvenilir araç verilerinin ikinci el araç satıcılarının yaşadığı sorunları çözebileceği ve bilgi asimetrisini azaltabileceğini göstermiştir.

Butera vd. [33] yaptıkları çalışmada, kullanılmış araç piyasasında alım-satım işlemleri ve araçların takibi için blokzincirin sağladığı avantajlarla, para ve mülkiyet transferlerini de içine alan NFT tabanlı bir çözüm sunmuşlarıdır. Ethereum, NFT, IPFS gibi teknolojilerin birlikte kullanıldığı bir mimari yapı tasarımı ile birlikte geliştirdikleri uygulama üzerinde, maliyet, performans analizi, güvenlik analizi değerlendirmesi ile Ethereum blokzincirin diğer blokzincir platformlarına göre yüksek maliyete sahip olduğunu belirlemişlerdir.

Zavolokina vd. [34] ikinci araç piyasasında bilgi asimetrilerinin temel sebebinin, satıcının satılan araç hakkında alıcının sahip olmadığı bilgilere sahip olmasından kaynaklandığını [35] vurguladığı çalışmasında, blokzincir platformunun bilgi asimetrisini önlemede ve güveni sağlamadaki etkisini deneylerle araştırmıştır. Deney sonuçlarında araç hakkında bilgi sağlamanın bilgi asimetrisini azalttığı ancak bilginin niteliğinin önemli olduğu bu nedenle bilgilerin doğruluğunun garanti edilmesi gerektiği vurgulanmıştır.

Bauer vd. [36] ikinci el araç pazarında yaşanan bilgi asimetrisi ve güven sorunlarına karşı blokzincir tabanlı bilgi sistemlerinin nasıl etkisi olabileceğini araştırmıştır. Niceliksel ve niteliksel ölçüm yöntemleriyle yaptıkları keşif çalışması sonunda blokzincir teknolojisi ile sağlanan kaliteli araç geçmişi bilgilerinin alıcı ve satıcılar açısından ayrı ayrı değerli olduğu bilgi asimetrisini azalttığı sonucuna ulaşmışlardır.

Barreto vd. [37] araç verilerinin bütünlüğünün korunmasının çeşitli dolandırıcılık yöntemlerinden korunmak için gerekli olduğunu belirtikleri çalışmalarında, veri manipülasyonundan, alıcı ve satıcının kimlik tahrifatlarına kadar uzanan dolandırıcılara karşı, blok zincirine dayalı bir sistemin uygulanmasını önermişler ve mevcut sorunların çözümünde blokzincir teknolojisinin neden uygun olduğunu incelemişlerdir.

Lopez-Pimentel vd. [38] NFT-Araç ismini verdileri dağıtık bir Ethereum NFT tabanlı dağıtık uygulama ile üretici, araç sahibi, resmi kurum, sigorta sağlayıcı, bakım hizmetleri ve alıcı gibi paydaşlar arasında akıllı mülkiyet fikrine dayalı dağıtık bir mimari üzerinden aracın yaşam döngüsü boyunca işlemlerin blokzincirine kaydedilmesini sağlayan bir model ve uygulama geliştirmişlerdir. Üretici tarafından her araç için NFT varlık jetonu (token) oluşturulması ve satma, satın alma, mülkiyet değişimi, bakım hizmetleri gibi işlemlerin akıllı sözleşmeler üzerinden gerçekleştirilmesi sağlanmıştır. Hesaplanan Ethereum Gas maliyetlerinin sunulan avantajalar göre aşırı olmadığı belirtilmiştir. Bu uygulama sayesinde araç satın alma işlemlerinde yaşanan dolandırıcılık gibi sorunların çözümüne katkı sağlandığı ifade edilmiştir.

Syed vd. [39] araç yaşam döngüsü takibi için blokzincir tabanlı kapsamlı bir çözüm sundukları çalışmalarını 4 modül içerisinde organize etmişlerdir. Bu modüller; (1) mülkiyet değişimini yeni ve kullanılmış araçlar için ayrı ayrı işleme almak, (2) yol sürüş kaydı ve kontrolü, ihlal ve kaza yönetimi ile birlikte bakım kayıtlarını oluşturmak, (3) kullanılmış araçlar için fiyat tahmini gerçekleştirmek, (4) aracın hurdaya ayrılma işlem kayıtlarını oluşturmak şeklinde düzenlenmiştir. İzinli Hyperledger Fabric platformu üzerinde geliştirdikleri uygulamada, sürücü, araç sahibi, sigorta ve bakım şirketlerinden oluşan roller belirlemişlerdir. Fiyat tahmini ile ilgili bazı süreçlerin zincir dışında gerçekleşmesini sağlamışlardır.

Brousmiche vd. [11] bir aracın yaşam döngüsüne ait işlemleri yönetmek ve bilgileri kayıt etmek için hem klasik veritabanı hem de blokzincirin kullanıldığı hibrit depolama yöntemini önermişlerdir. Paydaşlar arasındaki iş birliğini ve şeffaflığı sağlamak için sertifikalı, dijital bir bakım defteri oluşturulmasını önermişlerdir. Bakım kaydı, otomatik kilometre kaydı, araç satış işlemleri ve sigorta işlemlerinin nasıl gerçekleştiğine ait detay verilmemiştir.

Masoud vd. [40] CarChain adını verdikleri kullanılmış araçlar için blokzincir tabanlı bir sistem çerçevesi önermiştir. Ara verileri önbelleğe almak için geleneksel bir veritabanı kullanıldıkları önerilerinde sigorta şirketleri, tamir şirketleri ve araç sahipleri gibi paydaşlar bulunmaktadır. Dört farklı akıllı sözleşme ile işlem kayıtlarını ve güncellemelerini kontrol etmişlerdir. Uygulama hem blokzincir tabanlı hem de dağıtık, eşten eşe bir ağ olarak tasarlanmıştır.

Benarous vd. [2] merkezi yapıdaki araç tescil kuruluşlarında, olası sahte tescil işlem kayıtlarının, çifte kayıtların ve çalıntı araçların yeniden kaydedilmesinin engellenmesi için blokzincir teknolojisinin kullanımına yönelik bir çerçeve önermişlerdir. Önerdikleri ve uygulamasını yaptıkları sistemde olası sahte işlem enjeksiyonuna karşı, saldırı ağacı yöntemi ile güvenlik analizi gerçekleştirerek %10 dan daha az sahte işlem gerçekleşme olasılığın olduğunu

belirlemişlerdir. Öneri ve uygulamalarında kullandıkları blokzincir platformu hakkında detaylı bilgi verilmemiştir.

Florez vd. [41], Kolombiya'da sürücü ve araç kayıt sistemlerinde yaşanan sorunların çözümü için araç tescil belgeleriyle sürücü belgelerinin dijital belge olarak tanımlanması ve dijital imza ile doğrulanması işlemlerinde Blokzincir tabanlı bir sistemin benimsenmesi için bir model önermişler ve prototip uygulama geliştirmişlerdir

Distefano vd. [42] araç bilgi sistemini uygulamak ve araç yaşam döngüsü içerisindeki bilgilerin güvenli ve değişmez kaydını tutmak için sadece meta verilerin depolandığı Multichain platformu ve Mongodb veritabanı üzerinde çalışan bir uygulama tasarlamışlardır

Barolli vd. [43] çalışmasında, araç yaşam döngüsü yönetimi ve sigorta poliçelerine yönelik mevcut sistem ve platformları analiz ederek belirlenen sorunların çözümünde Blokzincir teknolojisinin ve yeni iş modellerinin kullanımına yönelik öneriler sunmuşlardır

1.2.3 Araçlarda kilometre sayacı sahtekarlığı ve veri manipülasyonlarının engellenmesi üzerine yapılmış çalışmalar

Uluslararası Otomobil Federasyonu (FIA) ve üyelerinin [44] Avrupa Parlamentosu Üyelerine yaptıkları çağrı metninde, Avrupa'nın çoğu ülkesinde yaygın olduğu bildirilen kilometre sayacı sahtekarlığının ikinci el araç pazarında tüketiciler için ciddi olumsuzluklar ve zararlar oluşturduğu bildirilmiştir. İkinci el araç piyasasında %30'lara ulaştığı tahmin edilen sahtekarlıkların, Avrupa tüketicilerine yıllık yaklaşık 5,6 ila 9,6 milyar avro maliyete sebep olduğunu bildirilen çağrı metninde, bu olumsuz durumun engellenmesi için gerekli araştırma ve eylemlerin acilen yapılması gerektiği vurgulanmıştır.

Abbade vd. [13] araçların kilometre sayaçları üzerinde yapılan sahtekarlıkların önlenmesi için iki bölümden oluşan bir API sistemi önermişlerdir. Araçlarla IoT ve 3G kablosuz teknolojilerle iletişim kurularak, HTTP protokolü üzerinden JSON formatında verilerin blokzincir tabanlı sisteme aktarılmasını ve zaman damgalı değiştirilemez kayıtların depolanmasını ve izlenmesini sağlayacak İş İspatı (POW) algoritmasına dayalı bir prototip uygulama geliştirmişlerdir.

Chanson vd [14] araç kilometre sayacı bilgilerinin halka açık Ethereum blokzincirde depolanmasını önermişlerdir. Araçlardaki kilometre ölçerler ile araçtaki OBD-II arayüzü üzerinden, aracın kilometre sayacı değerlerini ve araç kimliğini verilerini dizüstü bilgisayara gönderdiği bir deney düzeneği sunmuşlardır. Veriler daha sonra Node.js uygulaması ile Web3

kullanarak Ethereum ağına gönderilmiştir. Araç verileri blokzincir üzerinde depolanmış ve kullanıcı erişimine kapalı tutulmuştur.

Borkowski [45] Avrupa Birliğinden (AB) elde edilen kanıtlar göre kilometre sayacının tahrif edilmesinin olukça sık rastlanan bir sorun olduğunu, araçların değerleri arasındaki farkın engellendiğini belirtmişlerdir. Farklı ülkelerde kilometre sayacı tahrifatıyla oluşan maliyetlerin karılaştırmalarını ve değerlendirmelerini yaptıkları çalışmalarında, tahrifatın engellenmesinin eski yeni tüm araçlar için uygulanmasının önemli olduğu vurgulanmıştır.

Samuel vd. [46] yaptıkları çalışmada kilometre sayacı sahtekarlığı sorununu çözmeye odaklanmıştır. Araçların kilometre sayacı verilerini bir konsorsiyum blok zincirine gönderdiği bir mimari önermişlerdir. Bu mimari cüzdan tabanlı ve cüzdan tabanlı olmayan iki yaklaşımla tasarlanmıştır.

Bu tez çalışması, araç yaşam döngüsü üzerinde farklı problemlerin ele alınarak çözüm önerilerinin ve uygulamaların anlatıldığı yukarıdaki çalışmaların büyük çoğunluğundan farklı olarak, tüm araç yaşam döngüsünü izlemek ve takibini yapmak noktasındaki eksikliği gidermektedir. Bu tez, önerdiğimiz modelin modüler ve ölçeklenebilir olması, sonradan yeni modüllerin eklenebilmesine ve güncellemelerin yapılabilmesine imkân vermesi yanında kapsamlı bir araç veri yönetim modeli sunması yönüyle mevcut çalışmalardan ayrılmaktadır.

2. ARKA PLAN VE KULLANILAN TEKNOLOJİLER

2.1 BLOKZİNCİR TEKNOLOJİSİ

Bu bölümde blokzincir teknolojisinin yapısı, bileşenleri, türleri ve genel özellikleri üzerinde kavramsal bilgiler verilerek tez kapsamında önerilen blokzincir ağı için temel olan konu ve kavramlar açıklanmıştır.

2.1.1 Blokzincir Teknolojisi Tanımı ve Tarihçesi

Blokzincir, 2009 yılından itibaren adını duymaya başladığımız ve son yıllarda birçok gelişime ve değişime uğramış, bir çok alanda kullanılabilirliği keşfedilmiş yeni bir kavramdır. Uzun zamandır kullanılan kriptografi (Şifreleme bilimi), dağıtık defter teknolojisi (DLT), eşler arası ağ (P2P), mutabakat mekanizmaları ve merkle ağacı veri yapısı gibi eski teknolojilerin entegrasyonuyla standartlaştırılmaya çalışılan yeni bir teknoloji olarak da tanımlanmaktadır. Aşağıda tez kapsamında kullandığımız bu teknolojiye ait kavramsal açıklamalar verilmiştir.

2.1.1.1 Blokzincir Teknolojisi Tanımı

Blokzincir teknolojisi özünde bir veri depolama yöntemidir. Mevcut veri depolama ve veritabanı sistemlerinden bazı güçlü özellikleri ve getirdiği yenilikler nedeniyle ayrılmaktadır. Blokzincir teknolojisi, merkezi bir kontrol olmadan, tarafların (düğüm) kendilerine ait genel ve özel kriptografik anahtarlarıyla iletişim ve etkileşim kurması, işlemlerin akıllı sözleşmelerle otomatik olarak yürütülmesi, işlem kayıtlarının özdeş kopyalarının dağıtık yapıdaki madenci düğümlerde fikir birliği sağlanarak depolanması için, uzun süredir var olan çeşitli kriptografik ve dağıtık sistem ilkelerinin bir araya getirilmesiyle ortaya çıkarılmış yenilikçi bir veri depolama ve transfer yöntemidir [47]. Blokzincirde, verileri bloklar halinde, kronolojik ve kriptografik olarak birbirine bağlanarak bir zincir oluşturacak şekilde depolanır. Ayrıca güvenli bir işlem altyapısı ile merkezi bir otoriteye gerek olmadan dağıtık bir ortamda değiştirilemez, silinemez, şeffaf ve güvenli bir şekilde paylaşılır.

Dijital ortamlarda verilerin depolanması ve paylaşılması için geliştirilmiş olan birçok teknik ve yöntem uzun zamandır kullanılmaktadır. Blokzincir teknolojisi bu yöntemlere farklı bir yaklaşım getirmektedir. Bu farklılar şu şekilde özetlenebilir;

- **Merkeziyetsizlik:** Blokzincirde veriler tek bir merkezi sunucu yada otorite kontrolünde değildir. Blokzincir ağındaki tüm katılımcılar verilerin bir kopyasına sahip olduğu gibi yeni

verilerin eklenmesinde, fikir birliği içinde doğrulama yapmak zorundadır. Bu durum tek nokta hatası (SPOF, Single Point of Failure) ve saldırı noktası sorununu ortadan kaldırır ve merkeziyetsizlik sağlar. Klasik veri tabanlarında ise merkezi bir sunucu yada sunucu kümesinde çalışır ve tüm kontrol bu merkezi otoriteye aittir.

- **Değişmezlik:** Klasik veri tabanlarında veriler üzerinde; Oluşturma, Okuma, Güncelleme ve Silme (CRUD, Creat, Read, Update, Delete) işlemleri yapılabilirken, blokzincir teknolojisinde sadece veri oluşturma ve okuma işlemi yapılabilmektedir. Bir kez blokzincire eklenen veri/işlem kriptografik olarak birbirine bağlanan bloklar sayesinde değiştirilemez ve silinemez bir yapıdadır.
- **Veri bütünlüğü ve güven:** Blokzincir teknolojisinde, işlemlerin tutulduğu blok yapıları birbirleriyle kriptografik olarak bağlı olduğu için verilerin bütünlüğü de korunmuş olur. Dolayısıyla klasik veri tabanlarında kullanılan bütünlük koruma yöntemlerinden yedekleme, kısıtlama gibi tekniklere ihtiyaç yoktur.
- **Şefaflık:** Blokzincir ağında bulunan tüm katılımcılar tüm işlem ve verileri görebilir, işlem sahiplerinin kimlikleri ise gizli tutulur (anonimlik). Bu durum şeffaflık ve denetlenebilirlik sağlar. Klasik veri tabanlarında ise veriler bir otoritenin kontrolü altındadır.
- **Güven ve Güvenilirlik:** Blokzincir teknolojisinde işlemlerin doğrulanması ve verilerin kaydedilmesinde bir aracıya ihtiyaç yoktur. Ağda bulunan tüm katılımcılar mutabakat mekanizmaları ile işlemleri doğrular ve kaydeder. Bu durum taraflar arasında güven oluşturur. Klasik veri tabanlarında merkezi otorite ve güvenilir bir aracıya tarafından işlemlerin doğrulaması yapılır ve kayıt edilir. Güvenlik şifre yada erişim kontrolü ilse sağlanır bu durum tek saldırı noktası hatası oluşturur.

2.1.1.2 Blokzincir Teknolojisi Tarihçesi

Blokzincir kavramı, ilk kez 2008 yılında Satoshi Nakamoto'nun yazdığı “Eşler Arası Elektronik Bir Ödeme Sistemi, Bitcoin” (Bitcoin: A Peer-to-Peer Electronic Cash System) [48] makalesinde ortaya çıkmış ve ilk uygulaması 2009'da Bitcoin kripto para işlemleri için yapılmıştır. Blokzincir teknoloji ve Bitcoin, bankalar gibi finansal kuruluşlara ihtiyaç duymadan, değiştirilemez ve zaman damgalı Bitcoin kripto para işlem kayıtlarını dağıtık bir ağda depolamak için tasarlanmıştır. Tasarlanan sistemde, işlem yapanların gizliliğini korunmakta, çifte harcama önlenebilmekte ve verilerin bütünlüğünü iş kanıtı (PoW, Proof-of-Work) mutabakat algoritmasıyla sağlanmaktadır. Bitcoin’in kullandığı blokzincirin temel yapısında yer alan bazı

özelliklerin oluşturduğu ölçeklenebilirlik, bloklarda sınırlı işlem, işlemlerdeki gecikme süreleri, kısıtlı blok boyutu ve depolama kapasitesi ile düşük verimlilik gibi kısıtlamalar, darboğaz oluşturan işlem adımları, iş kanıtı (POW) mutabakat mekanizmasının yoğun iş yükü nedeniyle kullandığı enerji ve çevresel etkileri, sektörlerin ve kullanım alanlarının ihtiyaçlarının karşılanmasındaki zorluklar gibi konular üzerinde yenilikler getirilmiş, düzenlemeler yapılmış ve böylece birçok blokzincir platformu ortaya çıkarılmıştır. Ethereum, Vitalik Buterin tarafından 2015 yılında tanıtılarak hayata geçirilmiş ve blokzincir teknolojisine yepyeni bir boyut kazandırmıştır [49]. Vitalik Buterin, Ethereum EVM platformu üzerinde akıllı sözleşme program kodlarının kullanıldığı uygulamalar geliştirilmesini sağlayarak, blokzincirin finans alanı dışında birçok sektörde kullanılabilmesinin yolunu açmıştır. EVM üzerinde, akıllı sözleşmelerin geliştirilmesi için Solidty programlama dili tasarlanmış ve dünyaya tanıtılmıştır [49]. Akıllı sözleşmelerin gücüyle birlikte, kullanıcıların dijital varlıklarına ait işlemleri güvenle kaydedebildiği ve dağıtık uygulamalar geliştirebildiği bu platform, birçok dağıtık uygulamanın geliştirilmesi için ilham kaynağı olmuş ve blokzincir teknolojinin potansiyelini önemli ölçüde artırmıştır.

Akıllı sözleşme kavramı ilk olarak, Nick Szabo tarafından 1994 yılında yapılan çalışmasında ortaya konmuştur. Szabo, çalışmasında, taraflar arasındaki sözleşmenin bilgisayar ortamında bir protokol ile nasıl yapılacağı ve kriptografik olarak nasıl doğrulanacağından bahsetmiştir [50].

Bitcoin'den sonra, farklı özellikler sunan diğer kripto paraların da temelini oluşturan blokzincir teknolojisi geliştirilerek, çeşitli mutabakat algoritmaları ve akıllı sözleşme yöntemleri kullanan birçok blokzincir platformu ortaya çıkmıştır. Günümüzde yaygın olarak kullanılan blokzincir platformları arasında Ethereum, Corda, MultiChain ve Hyperledger Fabric gibi platformlar yer almaktadır.

2.1.1.3 Blokzincir Teknolojisinin Kullanım Alanları

Blokzincir teknoloji, kripto para uygulamaları ve finans alanındaki çeşitli işlemler haricinde, güvenliğin, şeffaf ve güvenilir işlemlerin ve en önemlisi gizliliğin önemli olduğu birçok alanda da başarıyla uygulanmıştır. Akıllı sözleşmelerin blokzincir teknolojisinde kullanımı ile fiziksel ve dijital varlıkların blokzincirde tanımlanması, bunlarla ilgili işlemlerin kayıt altına alınması ve otomatikleştirilmesini sağlayan yeni bir yöntem keşfedilmiştir. Blokzincirin

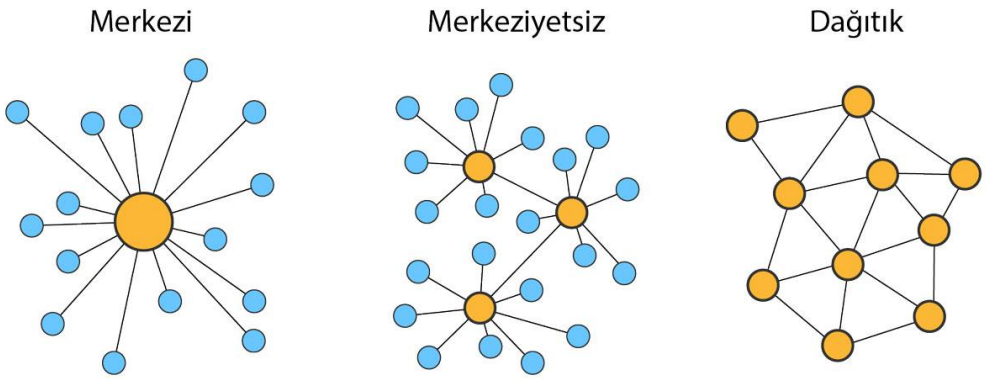
yapısındaki, merkeziyetsizlik, şeffaflık, denetlenebilirlik, kriptografik güvenlik, akıllı sözleşmelerle yürütülen işlemler, mutabakata dayalı onay süreçleri gibi temel özellikleriyle birçok sektörde kullanılabileceği fark edilmiş, birçok araştırma yapılmış ve sektörlerin iş akış yöntemlerinin iyileştirilmesi veya kökten değiştirilmesi potansiyeli üzerine uygulamalar geliştirilmiştir [51]. Blokzincir teknolojisi, tedarik zinciri yönetiminde ürünlerin takibini kolaylaştırarak şeffaflık ve güven sağlarken, sağlık sektöründe hasta kayıtlarının güvenli bir şekilde saklanması ve paylaşılması için ideal bir çözüm sunmuştur. Ayrıca, fikri mülkiyet haklarının korunması, oylama sistemlerinin şeffaflığı ve hatta nesnelerin interneti (IoT) cihazları arasındaki güvenli iletişim gibi birçok alanda da kullanılabilirliği üzerinde çok sayıda çalışma yapılmış ve uygulamalar geliştirilmiştir.

2.1.2 Blokzincir Teknolojisinin Bileşenleri

Bu tez çalışmamızda kullanılan blokzincir teknolojisine ait temel bileşen sırasıyla aşağıdaki gibidir. Çalışmamız için gerekli olmayan bileşenler ele alınmamıştır.

2.1.2.1 Dağıtık Defter Teknolojisi (DLT)

Blokzincir merkezi yapıdaki sistemlere karşı alternatif bir sistem olan dağıtık ve aynı zamanda merkeziyetsiz bir ağ yapısına sahiptir. Merkeziyetsizlik, bir otoriteye bağlı olmadan sistemin çalışabilmesi bir aracıya güven ihtiyacının olmaması anlamına gelmektedir. Merkeziyetsizlik için dağıtık bir ağ üzerinde, ağdaki tüm düğümler tarafından verilerin kontrolünün sağlanması gerekir [52]. Dağıtık bir ağda, düğümler tüm işlem kayıtlarının bir kopyasını üzerinde tutarak ve kopyaların tutarlığını sürekli kontrol edip garanti altına alarak merkezi yapılara ait tek nokta hatası sorunlarına, sorumluların görev ve yetkilerini suistimal edip hileli işlemler yapmalarına, siber saldırılara karşı bir çözüm sunmaktadırlar. Şekil 2.1'de merkezi, merkezi olmayan ve eşten eşe (P2P) dağıtık ağlar için basit bir topoloji şekli verilmiştir.



Şekil 2-1: Merkezi, merkeziyetsiz ve dağıtık ağ yapısı [52]

Merkeziyetsiz bir yapı şeffaflık ve güven sağlasa da mahremiyet ve gizlilik kaygıları nedeniyle her zaman tercih edilmez. Bazen uygulamaların hedef ve ihtiyaçlarına, verimlilik ve güvenlik durumlarına göre merkezi bir otoritenin kontrolünde olmak daha önemli olabilir.
Blokzincir teknolojisinde dağıtık bir yapının olması tarafların bir güvenli aracıya, otoriteye ihtiyaç olmadan doğrudan birbirleriyle güvenle işlem ve paylaşım yapabilme imkanına sahip olmaktadır. Bu durum şeffaflığın sağlanması, sansürlerin ortadan kalkması, hesap verebilirliğin ve denetimin sağlanması ile birlikte siber saldırılara ve veri manipülasyonlarına karşı güvenli ve güvenilir bir ortam oluşturur. Dağıtık ağlarda eşten eşe iletişim yöntemi uygulanır. Bu yöntemde ağdaki düğümler birbirleriyle yapacakları iletişimler için bir sunucu düğüme ihtiyaç duymazlar. Ağdaki her düğüm diğer düğümlere ait erişim bilgilerine sahip olduğu için her düğüm istediği diğer düğümle doğrudan iletişim kurabilir. Böylece eşten eşe bağlantı iki düğüm arasında aracısız bir şekilde sağlanmış olur.

Blokzincir teknolojisinde veriler bloklar içerisinde, bloklarda birbirlerine kriptografik özet değerleri ile birbirine bağlı bir zincir yapısında tutulmaktadır. Bu zincir yapısı ağdaki tüm düğümlere dağıtıldığı için dağıtık defter (Distributed Ledger) adını alır. Dağıtık ağ bir iletişim ve veri paylaşım ortamı sağlarken, dağıtık defter ise dağıtık ağ ortamında verilerin güvenli ve değiştirilemez şekilde dijital kayıtlarının tutulmasını sağlamaktadır. Geleneksel merkezi sistemlerde bulunan veri tabanlarından farklı olarak dağıtık defterler tek noktadan yönetilmezler. Defter kaydında yapılan bir değişiklik ağ üzerinde tüm düğümlere eş zamanlı olarak dağıtılır [53]. Blokzincir ağlarının en önemli özelliği olan Dağıtık Defter Teknolojisi (DLT), verilerin güvenle depolanması, yedeklenmesi, işlenmesi, paylaşılması ve aktarılması gibi işlemlerde oldukça önemli bir role sahiptir Bu teknolojiye ait temel fikirler 1990'lı yılara kadar uzanmaktadır. Haber vd. [54] ait çalışmada dijital belgelerin zaman damgalı imzalanması, Anderson [55] 1996 tarihindeki,

merkezi olmayan veri depolama sistemi önerisi, Schneier vd. [56] çalışmalarında, hassas ve önemli verilerin güvenilmeyen makineler arasında kriptografik şifrelenmesi üzerine yaptıkları çalışmalar DLT'nin ve blokzincirin temel fikirlerini oluşturmaktadır. Dağıtık ağ yapısı, mutabakat mekanizması, kriptografik şifreleme algoritmaları, DLT'yi güçlendiren temel özelliklerdir. Bu özelliklerin birlikte kullanılması nedeniyle blokzincir teknolojisi çoğu zaman dağıtık defter teknolojisi olarak da isimlendirilmektedir. DLT'ye ait temel özelliklerin oluşturduğu kriptografik şifrelemeler, mutabakat onayları ve verilerin ağda paylaşımı gibi işlemler, işlem ve zaman maliyeti ortaya çıkarmaktadır. Merkezi yapılarla kıyaslandığında DLT'deki işlem hızı ve verimi düşük kalmaktadır. Ancak güvenlik ve güvenilirlik noktasında önemli faydalar sağlamaktadır. Blokzincire benzer bir şekilde, Tangle teknolojisi de, DLT'yi altyapısında kullanan başka bir teknolojidir.

2.1.2.2 İşlemler

Bitcoin blokzincir platformunda, elektronik dijital para sahibinin, alıcının açık anahtar kriptografik özet bilgisi ile göndereceği para miktarı ve bir önceki duruma ait kriptografik özet değeri ile oluşturulan bilgiyi kendi özel anahtarıyla kriptografik olarak imzalayarak yeni alıcı sahibine göndermesine işlem adı verilir [48]. Ağdaki tüm düğümlere dağıtılan işlemler, madenci düğümler tarafından bir bloğa dahil edilip mutabakat ile onaylandıktan sonra blokzincire kalıcı olarak kaydedilir.

Ethereum blokzincir platformunda işlem, sadece kripto para (ETH) transferi değil, bir değer ya da varlık transferi olabileceği gibi, Ethereum Sanal Makinesi (EVM) platformunda bulunan akıllı sözleşmelerle etkileşimi ve kriptografik olarak imzalanmış bir talimatı temsil edebilir. Ethereum ağında işlemlerin akıllı sözleşmelerde yürütülmesi için bir "GAS" ücretlendirmesi yapılır. GAS, işlem için gereken hesaplama kaynaklarının kullanımına ait bir ölçüdür.

Diğer blokzincir platformlarında işlem, platformların tasarımına, kullanım amacına ve sunduğu özelliklere göre değişebilmekle birlikte genelde akıllı sözleşmelerle olan etkileşimi, kriptografik olarak imzalanmış ağdaki düğümler tarafından doğrulanıp blokzincire kaydedilen bir eylem ya da durum değişikliğini ifade etmektedir.

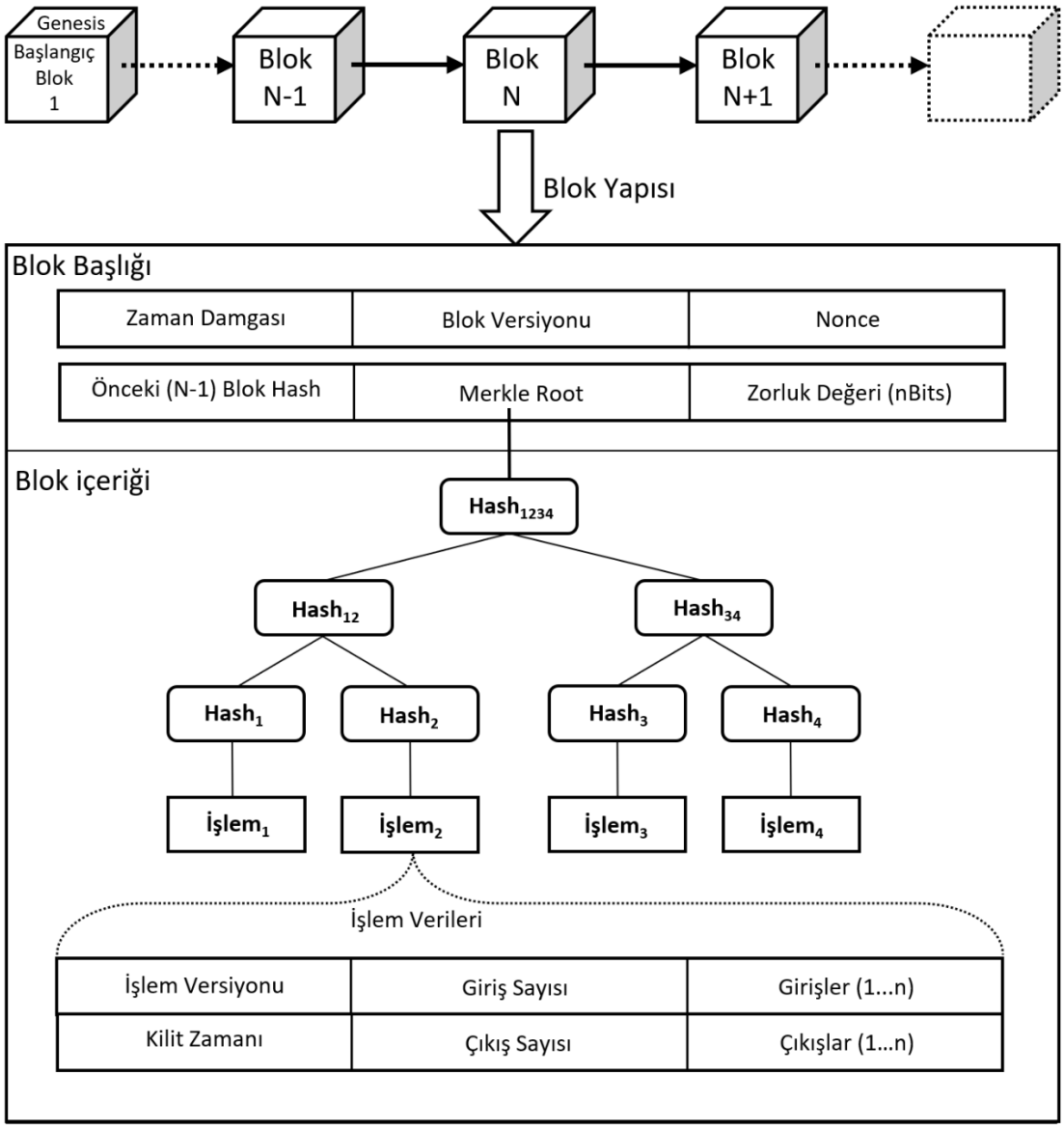
Blokzincirlerin en önemli özelliği, blok içerisine alınıp zincire eklenen her işlemin silinememesi ve üzerinde değişiklik yapılamasıdır. Blokzincirde kullanılan kriptografi yöntemleri değişmezliğin

garantisini sağlar. Bu yönüyle diğer merkezi veritabanı yöntemlerinde var olan, tek nokta hatası, siber saldırı, veri kaybı ve manipülasyon gibi risklere karşı blokzinciri daha üstün hale getirir.

2.1.2.3 Blok

Blokzincir ağlarında gerçekleştirilen işlemlerin kayıtları bloklar içerisinde tutulur ve bloklar birbirlerine bağlanarak zincir oluşturur. Teknolojinin adı bu yapısal özelliğinden gelmektedir. Zincirin en başındaki ilk blok başlangıç bloğu anlamında "Genesis" olarak isimlendirilmiştir. Blok içerisinde, blok meta verileri ile birlikte, blok içerisindeki sıralı işlem verileri ve sayısı bulunmaktadır. Blok meta verileri; Şekil 2.2'de gösterildiği gibi bloğun oluşturulduğu tarih ve saati temsil eden zaman damgası, bloğu oluşturan düğüme ait dijital imza ve açık anahtar bilgileri, blok numarası ve versiyonu, Zorluk değeri (nBits), sıralanmış işlemlerin merkle kök özet değeri, bloğun özet değeri ve önceki bloğun özet değeri bilgileri yer almaktadır [57].

İş Kanıtı (PoW) mutabakat mekanizmasının kullanıldığı blokzincir ağında yayınlanmış olan işlemler, madenciler tarafından blok içerisine dahil edilerek, zorlu bir matematiksel hesaplama ile çeşitli kombinasyonların denenmesiyle elde edilen "Nonce" değerinin bulunmasından sonra zincire eklenir. Yeni oluşturulan blok içerisinde; işlem verileri, işlem verilerinden elde edilen Merkle kök değeri, zaman damgası, Nonce değeri, bir önceki bloğa ait özet değeri ve bloğun mevcut tüm bilgilerinden hesaplanan özet değeri bulunmaktadır [48]. Nonce değeri, bloğun özet değerinin, ağda belirtilen zorluk derecesine göre hesaplanmasında kullanılan ve değiştirilebilen bir sayıdır. Şekil 2.2'de Bitcoin blokzincirine ait blok ve zincir yapısı verilmiştir [58].



Şekil 2-2: Bitcoin blokzinciri blok ve işlem veri yapısı

Blok içerisindeki herhangi bir verinin değişmesi bloğun özet değerini de değiştirecektir. Bu değişim sonraki bloklarla uyumsuzluk oluşturacağı için diğer düğümler tarafından hemen fark edilecektir. Değişikliğin yapıldığı zincirin bütünlüğünün sağlanması ve diğer düğümler tarafından kabul edilebilmesi için, sonraki bloklarda bulunan tüm özet değerlerinin de yeniden hesaplanarak değişmesi gerekir. Ancak bu her blok için iş kanıtının yeniden yapılmasını gerektirdiği için yoğun bir hesaplama yükü oluşturmakta ve ağdaki en uzun zincirin önüne geçilmesi gerekmektedir. Ağda dürüst düğümlerin varlığı ve çokluğu bu durumun oluşmasını pek mümkün hale getirmeyecektir. Ancak ağdaki düğümlerin %51'inin kontrol edilmesi halinde mümkün olabilir [48].

2.1.2.4 Düğüm ve Madenci

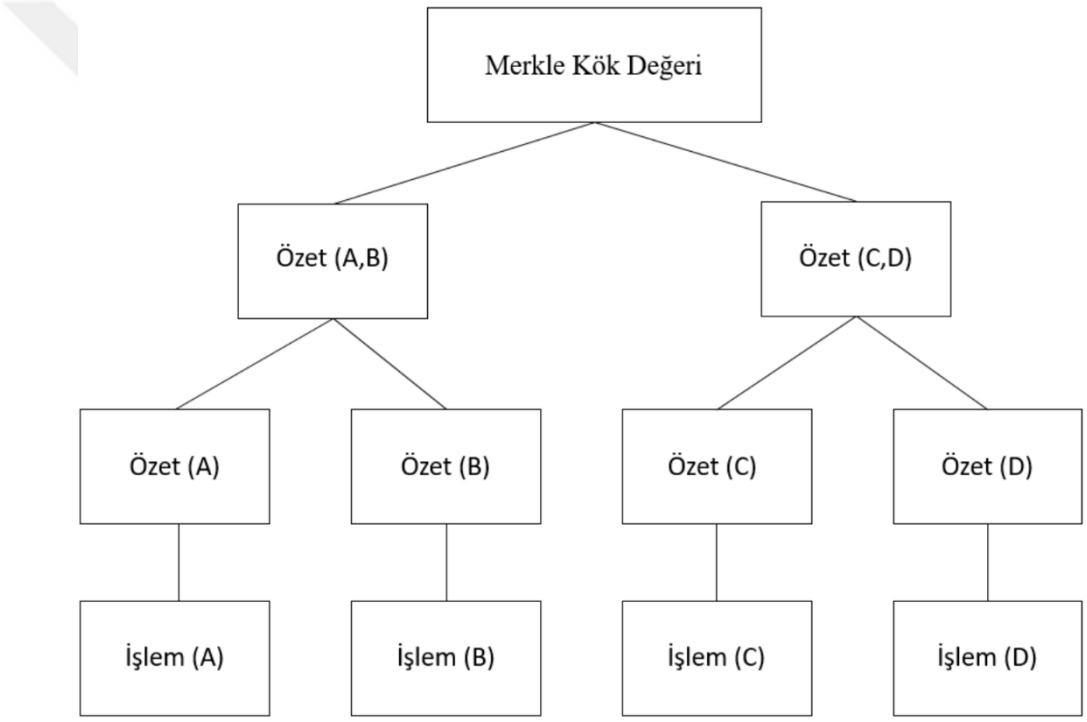
Blokzincirin temel bileşeni olan düğümler, eşler arası (P2P) bağlantı yöntemiyle veri alış-verişi yapan bilgisayar ve sunucu gibi cihazlardır. Ağın protokol yazılımını ve akıllı sözleşme kodlarını çalıştıran, işlemleri ve blokları doğrulama, geçerliliğini kontrol etme görevini yerine getirip mutabakat mekanizması içerisinde oy kullanan düğümlerin sayısının çokluğu, blokzincir ağının ayakta kalması ve sürdürülebilir olmasında oldukça önemlidir. Ancak her düğüm blok oluşturma ve doğrulama görevine sahip olmayabilir. Bazı düğümler sadece zincirin bir kopyasını tutmaktan sorumlu olabilir. Ağdaki her düğüm genel ve özel kriptografik anahtara sahiptir. Düğümler sahip oldukları anahtarlar ile imza ve şifreleme işlemlerini kullanarak birbirleri arasında bilgi alışverişi yapar ve doğrular. Böylece ağ üzerinde gerçekleştirilen işlemlerde verilerin tutarlığı, inkâr edilemezliği ve değiştirilemezliği güvence altına alınarak, ağın güvenli ve güvenilir olması sağlanır. İş kanıtı (PoW) mutabakat mekanizmasını kullanan Bitcoin ve Ethereum gibi ağlarda bulunan özel düğümlere Madenci (miner) adı verilmektedir. Madenciler, ağda yayınlanan işlemlerden yeni bloklar oluşturmak ve blokları zincire eklemekten sorumludur. Bloğu oluşturmak için belirli periyotlarda ağda oluşturulan zorluk derecesine bağlı olarak karmaşık matematiksel problemi ilk çözen madenci düğüm ağ tarafından belirlenen bir ödülü (yeni üretilmiş kripto para ya da işlem ücretleri) kazanır. Madenci düğümlerin çokluğu, varlığı ve ödül ile ağda kalmalarının teşvik edilmesi hem ağ güvenliği hem de ağın sürekliği için oldukça önemlidir. Ancak madencilerin zorlu matematiksel kombinasyon hesaplamalarını belirlenen süre içerisinde diğer düğümlerden önce yaparak yeni blok oluşturabilmesi için yüksek işlem gücüne sahip bilgisayarlar kullanmaları ve yüksek miktarda enerji harcamaları gerekmektedir. 2018-2020 yılları arasına ait bir araştırmada, Bitcoin ağında kullanılan elektrik enerjisinin, Avusturya veya İrlanda'nın kullandığı elektrik enerjisine yakın olduğu ifade edilmiştir [59].

Bitcoin platformunda bulunan düğümler, istedikleri zaman ağa katılıp ağdan ayrılabilmektedirler. Ağa katılan her düğümün ağda o anda var olan en uzun zinciri doğru kabul ederek bu zinciri uzatmaya, yeni blokları bu zincire eklemeye çalışması gerekir. Böylece ağda birden fazla farklı zincir oluşumu ve çatallanma engellenmiş olmaktadır.

2.1.2.5 Merkle Ağaç Yapısı

Blok içerisine alınan çok sayıdaki işlemlerin ayrı ayrı doğrulanması yerine bütün işlemleri içine alacak şekilde tek bir kriptografik özetin oluşturulmasını sağlayan yöntemdir. Bir ağaç

yapısına benzetilerek oluşturulan Merkle ağaç yapısı; Şekil 2.3'de verilen ağaç yapısındaki diyagramda, yaprakları temsil edilen ikili işlemlerden dalları oluşturan özetlerin ve ikili dalların özetlerinden köke doğru uzanan ve en sonunda kök özetin elde edilmesini sağlayan bir yapıdır. Merkle ağaç yapısı, ağdaki düğümlerde blokların daha hızlı, verimli ve güvenli bir şekilde doğrulanmasını kolaylaştırmakta ve doğrulama işleminin büyük miktarlardaki veriler için bile, daha hızlı yapılmasını sağlamaktadır. Bloktaki herhangi bir işlemde yapılacak en küçük bir değişiklik işlemin özet değerini ve takip eden özet değişiklikleriyle birlikte merkle kök özet değerini değiştirecektir. Böylece blok içerisindeki işlemlerde herhangi bir veri manipülasyonun yapılıp yapılmadığı kolayca tespit edilebilir. Merkle kök değeri blok başlığında yazılarak sadece bütünlük kontrol işlemlerinde kullanılmaktadır.



Şekil 2-3: Merkle ağacı

2.1.2.6 Akıllı Sözleşmeler

Akıllı sözleşmeler, blokzincir ağlarında düğümler üzerinde saklanan ve belirli koşullar sağlandığında, tanımlanan kurallara göre otomatik olarak çalışan program kodlarıdır. Geleneksel sözleşmelerde taraflar arasındaki anlaşmaların ve belirlenen sözleşme kurallarının dijital ortamdaki karşılıklarının, güvenilir bir otoriteye, aracıya gerek kalmadan otomatik olarak çalıştırılması olarak da tanımlanabilir. Akıllı sözleşmelerin girdileri, ağda dağıtılan işlemlerdir.

Düğümler işlemlerin önceden belirlenmiş ve kodlanmış koşulları sağlayıp sağlamadıklarını akıllı sözleşme program kodlarına göre yürüterek sonuçlarına göre işlemlerin onaylanmasını sağlar. Akıllı sözleşme kodları blokzincir ağında düğümlere bir kez dağıtıldığında değiştirilemez. Böylece kurcalamaya karşı koruma sağlanmış olur.

2.1.2.7 Mutabakat Mekanizmaları

Blokzincirlerin altyapısında yer alan DLT içerisinde, oluşturulan yeni blokların onaylanması ve geçerliliğinin doğrulanması sürecine mutabakat denilmektedir. Bir düğüm tarafından oluşturulan ve zincire eklenen bir blok, ağdaki diğer düğümler tarafından da kontrol edilmekte, blok ve içerisinde bulunan işlemlerin doğrulaması yapılarak geçerliliği üzerinde fikir birliğinin sağlanıp sağlanmadığına bakılmaktadır. Bloğun oluşturulması ve ağda dağıtılmasında mutabakat sürecinin hem güvenli hem de hızlı olmasını sağlamak için birçok algoritma geliştirilmiştir. Sistemin güvenliği ve güvenilirliği için mutabakat algoritmaları, ağda dürüst olmayan düğümlerin, tüm düğümlerin çoğunluğunu kontrol altına almasını zorlaştırılması, DLT sisteminin dürüst olmayan düğümler tarafından ele geçirilmesini engellemesi gerekmektedir.

Bitcoin ve Ethereum blokzincir platformlarında, ağda kimliği bilinmeyen anonim düğümlerden bir bloğun oluşturulması ve doğrulanması için yoğun hesaplama gücü gerektiren bir matematiksel kombinasyon hesabı ve ayarlanabilir zorluk derecesi ile yaklaşık olarak 10 dakikada bir bloğun oluşturulmasını sağlayacak, bloğu oluşturan dürüst düğümlere dürüst kalmaya devam etmelerini teşvik etmek için ödüllendirerek İş Kanıtı (PoW) mutabakat algoritması kullanmaktadır [48].

Bitcoin ve Ethereum gibi bazı platformlarında, madenci düğümlerden %51'inin kontrol altına alınması durumunda mutabakat mekanizmasının manipüle edilmesi riski vardır. %51 atağı olarak tanımlanan bu saldırı ile; madenci düğümlerin kontrol altına alınması ile önceki geçmiş işlemler üzerinde değişikliklerin yapılarak yeni bir zincir çatallanması yapılabilir ve çatallanma sonrası oluşan dalın daha uzun olması sağlanarak ağın tümü için geçerli hale getirilebilir. Çifte harcamalar yapılarak kontrol altındaki düğümlerden çoğunluk onaylarını alabilir ve ağın durdurulmasını sağlayabilir [60]. Blokzincir platformlarının bu riskleri önlemek için güçlü mutabakat mekanizmalarını kullanması, madencilik gücünün adil ve güvenilir bir şekilde dağıtılması, katmanlı yapılanma gibi önlemler alması gerekmektedir.

2.1.2.8 Kriptografi

Blokzincir teknolojisinin güvenli ve güvenilir olmasının temelinde kriptografinin kullanılması yer almaktadır. Blokzincir de kullanılan çeşitli kriptografik tekniklerle, düğümler arası iletişimde güvenlik, işlemlerde doğruluk, verilerde bütünlük ve gizlilik sağlanmaktadır. Kriptografik yöntemler kullanım amaçlarına göre, çeşitli matematiksel algoritma ve şifreleme tekniklerine sahiptir. Blokzincir teknolojisinde kullanılan başlıca kriptografik yöntemler ve kullanım amaçları aşağıda listelenmiştir:

- **Simetrik Şifreleme Yöntemi:** Simetrik şifreleme yönteminde, tek anahtar kullanılarak bu anahtarla şifreleme ve şifre çözme işlemi yapılmaktadır. Simetrik şifreleme yönetimi ile yapılan iletişimlerde tarafların aynı anahtarı paylaşarak kullanmaları gerekir. İletişimi dinleyen bir saldırgan anahtara sahip olmadığı için şifrenmiş verileri çözemez. Simetrik şifreleme daha hızlı çalıştığı için büyük miktarlardaki verilerin şifrelenmesinde basitliği ve daha az işlemci gücü gerektirdiği için daha verimlidir. AES, DES, RC4 vb. gibi algoritmaları vardır.
- **Asimetrik Şifreleme Yöntemi:** Asimetrik şifreleme yönteminde, anahtar çifti (özel ve genel) ile şifreleme ve şifre çözme işlemleri yapılmaktadır. Anahtar çiftleri matematiksel olarak birbirleriyle bağlantılıdır. Yani, genel anahtarla şifrelenmiş bir veri ancak o genel anahtarın eş çifti özel anahtarla doğrulanabilir. Genel anahtar ağdaki herkesle paylaşılabilir, ancak özel anahtar sadece sahibi tarafından kullanılmalı ve iyi muhafaza edilmelidir. Bu yöntem genelde dijital imza, kimlik doğrulama ve güvenli iletişimde kullanılmaktadır. Blokzincir platformlarında çoğunlukla asimetrik şifreleme dijital imzalar için kullanılır. Örneğin blokzincir ağında, düğümler arası iletişimde veri gönderen düğüm, özel anahtarını kullanarak veriyi imzalar ve alıcı düğüm, göndericinin genel anahtarını kullanarak imzanın doğruluğunu kontrol edebilir. İşlemlerde veri bütünlüğü ve gönderenin kimliğinin doğrulanmasında sıklıkla kullanılır. Asimetrik şifreleme de daha fazla matematiksel işlem yapıldığı ve daha fazla işlemci gücü kullanıldığı için yavaştır ancak özel anahtar gizli tutulduğu sürece daha güvenlidir. Yaygın olarak kullanılan RSA, ECDSA, DSA gibi algoritmaları vardır.
- **Merkle Ağacı:** Eski bir veri doğrulama ve bütünlük kontrol yöntemi olan Merkle Ağacı, verilerin ağaç yapısında hiyerarşik olarak saklanmasında kullanılan bir veri yapısıdır [61]. Büyük miktarlardaki verinin güvenli ve verimli bir şekilde doğrulanmasında kriptografik özet/hash fonksiyonlarından faydalanılır. Blokzincirde yaygın olarak blok yapısı içerisinde

kullanılan Merkle Ağacı için veriler işlemlerden oluşmaktadır. Her işlemin özet değeri Merkle ağacının yaprak düğümlerini oluşturmaktadır. Ara düğümlerin belirlenmesi için her iki yaprak düğüm özetlerinden tekrar özet değer oluşturulur. Tek kalan yaprak düğüm en son ara düğüm olarak kabul edilir. Her ara düğüm çiftlerinden bir alt ara düğüm oluşturma işlemi kök düğüme kadar tekrarlanır. Yapılan birleştirme ve özet değer elde etme işlemleri sonrası tek özet değere ulaşılır ve son elde edilen bu değere Merkle kökü adı verilir. Böylece blok içerisindeki tüm sıralı işlemlerin kriptografik özeti elde edilmiş olur.

- **Özet (Hash) Fonksiyonu:** Bu kriptografik fonksiyon verinin sabit uzunlukta karakterden oluşan karmasına dönüştüren matematiksel bir fonksiyondur. Aynı veri girişi her zaman aynı çıktıyı oluşturur. Bu fonksiyon tek yönlü olarak çalışır, özet değerden tekrar veriye ulaşılamaz. Veride ki en küçük bir değişiklik özet değeri tamamen değiştirmektedir. Verinin büyük ya da küçük boyutlu olduğu fark etmeksizin sabit karakter sayısı uzunluğunda bir çıktı elde edilir. Genelde verilerin bütünlüğünün kontrolünde kullanılmaktadır. Blokzincir platformlarında Merkle Ağaç yapısında kullanıldığı gibi bloklar arası bağlantıyı sağlamak için de kullanılmaktadır. Ayrıca düğümler arasında paylaşılan işlem ve kayıt defterinin bütünlük kontrolü ve doğrulaması için kullanılır. Blokzincir ağlarında çoğunlukla SHA-256 (Secure Hash Algorithm 256-bit) özetleme algoritması kullanılmaktadır.

2.1.3 Blokzincir Türleri

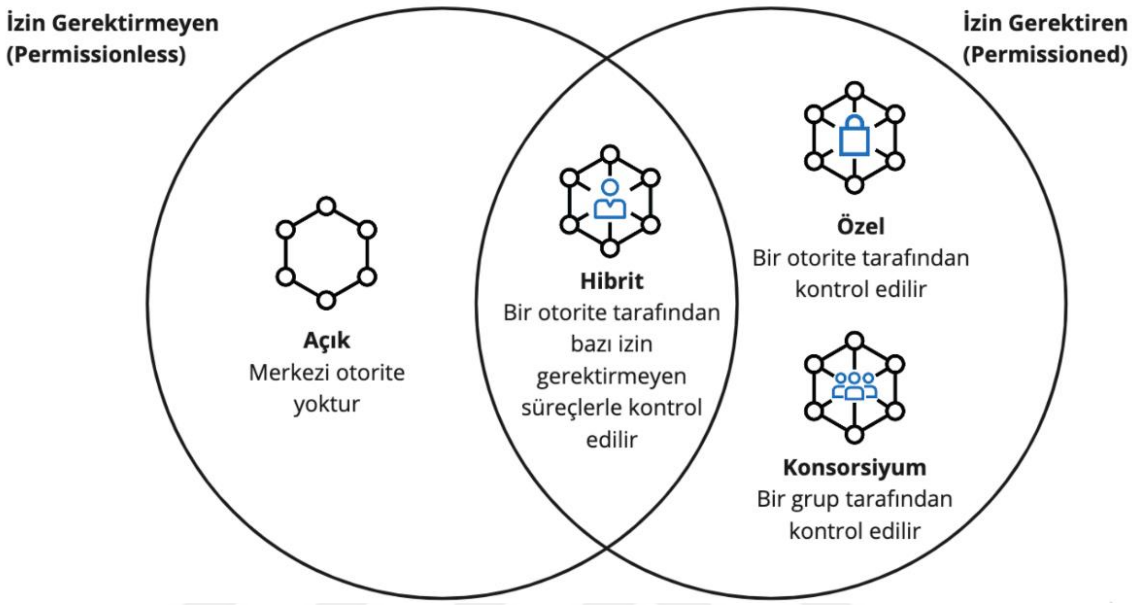
Blokzincir sistemleri, yönetim modeli, erişim izinleri ile dijital ya da fiziki varlık tanımlarına göre çeşitli türlerde geliştirilmiş farklı kullanım amaçlarına göre özelleştirilmiştir. Genelde izinli, izinsiz şeklinde sınıflandırılan blokzincir ağları Şekil 2.4'de gösterildiği gibi ortak özellikleriyle birlikte çeşitli kullanım senaryolarına göre konsorsiyum ve hibrit olarak da tasarlanabilmektedir.

- **İzinsiz (Genel) Blokzincir:** İzin gerektirmeyen blokzincir ağlarında, herhangi bir düğüm mutabakat sürecine katılabilmektedir. İşlemler şeffaf olup her düğüm tarafından görülebilir. Tam bir merkeziyetsizlik özelliğine sahiptir. Ağdaki düğümlerin kimliklerinin bilinmesine ihtiyaç yoktur, dolayısıyla anonimdirler. Düğümlerin ağa katılımı izne tabi olmadığı gibi istedikleri zaman ağdan çıkılabilir. Ağdaki düğüm sayısının çok olması ağın güvenliği için önemli olmasına rağmen işlemlerin ve blokların ağ üzerinde dağılımı ve

mutabakat onay süreçlerinde gecikme süreleri yüksektir. Ağdaki düğümlere güvenilmediği için bu türdeki ağlarda genelde İş kanıtı (PoW) veya hisse kanıtı (PoS) mutabakat mekanizmaları tercih edilmektedir [62]. Bitcoin ve Ethereum gibi popüler blokzincir platformları izin gerektirmeyen blokzincirlerdir.

- **İzinli (Özel) Blokzincir:** İzin gerektiren blokzincir ağlarında, ağa katılım izne tabi olduğu gibi mutabakat süreçlerine katılım diğer düğümlerin onaylarına, merkezi bir otoritenin iznine bağlı olarak, belirlenen yetkilere göre işlemlerin yapılmasıyla tüm süreçler kontrol altındadır. Dolayısıyla bu blokzincir türünün merkeziyetsiz olduğu söylenemez. Merkeziyetsizlik özelliği olmamasına rağmen, izin verilen ve güvenilen düğümlerle iş birliği yapıldığı için mutabakat süreçleri daha hızlı ve gecikmeler daha az olmakta, verimlilik artmaktadır. Bu tür ağlarda düğümlerin kimlikleri bilindiği ve güvenildiği için Raft veya Pratik Bizans Hata Toleransı (PBFT) gibi daha hızlı mutabakat mekanizmaları kullanılabilmektedir. Ağ yöneticileri ve diğer düğümlerin onayları ile katılım sağlanması, işlemlerin kontrol altında yürütülmesi, kısmen gizlilik ve mahremiyet sağlaması nedeniyle bazı sektörlerde kullanımı önemli görülebilir ve tercih edilebilir. Bu türün örnekleri arasında Hyperledger ve MultiChain gibi blokzincir platformları bulunmaktadır. Ayrıca Ethereum blokzincirin kurumsal kullanımlar için izinli ve özel ağ tasarımlarında kullanılabilmesi de mümkündür.
- **Konsorsiyum Blokzincir:** Belirli bir grup ya da kurumlar arasındaki iş birliğinin sağlanması ve yönetilmesi için oluşturulan blokzincir ağlarıdır. Ağa katılım ve işlem yetkileri, ağdaki birden fazla düğüm tarafından önceden tanımlanmış kurallara göre verilmektedir. Ağda paylaşılan işlemleri hangi düğümlerin görebileceği, işlemleri gerçekleştirebileceği ve şeffaflık düzeyi, düğümlerin bir kısmından oluşturulan konsorsiyum grubu tarafından belirlenmektedir. Bu yönüyle, bu blokzincir türünün yarı merkeziyetsiz olduğu söylenebilir. Birden fazla kuruluşun iş birliği yaptığı, paylaşımlarının gizlilik ve mahremiyetini kontrol altına almak isteyen özellikle finans alanında, lojistik ve tedarik zinciri gibi uygulamalarda tercih edilmektedir. Bu türde yer alan örnek blokzincir platformları arasında Hyperledger Fabric, R3 Corda ve Quorum gibi platformlar bulunmaktadır.
- **Hibrit Blokzincir:** İzinli ve izinsiz blokzincir ağlarının bazı temel özelliklerinin birlikte kullanıldığı ve birleştirildiği blokzincir ağlarıdır. Bu ağda belirli işlemlerin herkese açık olması sağlandığı gibi bazı işlemlerin sınırlı ve kontrollü erişimle yönetilmesi amaçlanır. İşlemlerin şeffaflığı ile gizliliği ağ yöneticilerinin kontrolünde olduğu için tam bir

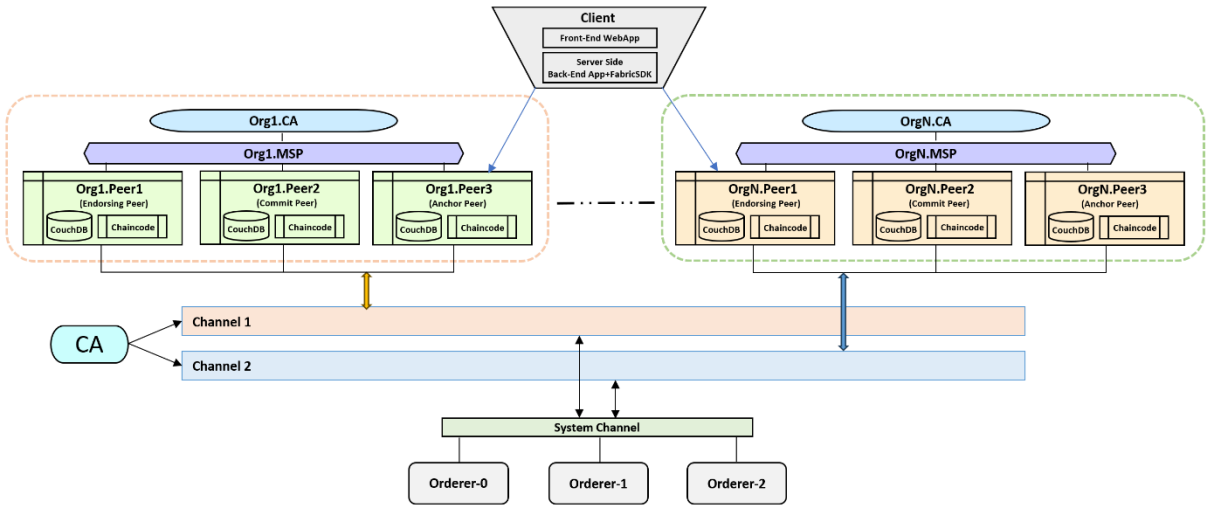
merkeziyetsiz yapıda olduğu söylenemez. Ağın yönetim kuralları ile merkeziyetsiz olma derecesi değişkenlik gösterebilir. Hibrit yapıya sahip blokzincir ağları farklı kullanım senaryolarında bazı faydalar sağlayabilecek potansiyeldedir.



Şekil 2-4: Blokzincir türleri [63]

2.2 HYPERLEDGER FABRIC (HLF) BLOKZİNCİR PLATFORMU

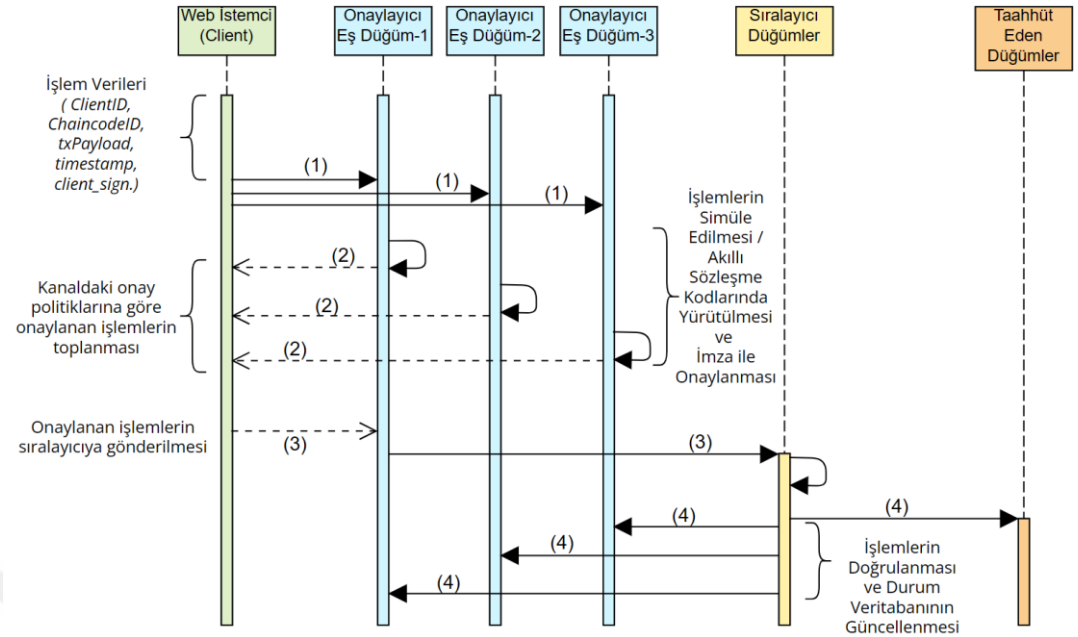
Hyperledger, Linux Foundation çeşitli blokzincir platformlarını ve araçlarını geliştiren bir projesidir. Blokzincir platform projelerinden en popüler olan, modüler bir yapıda ve akıllı sözleşmelere odaklanan açık kaynak kodlu Hyperledger Fabric'tir [64]. HLF, IBM şirketi ile Linux Foundation işbirliği ile geliştirilmiştir. HLF 'te kripto para birimi yoktur, temel amaç gerçek dünya iş senaryolarını çalıştırabilen bir blokzincir çerçevesi geliştirmektir. Çeşitli sektörler için izinli bir ağ yapısında DLT 'in kullanımını destekler. Geliştirilen blokzincir çözümlerinin dayanıklılığını ve esnekliğini daha üst düzeye çıkarmak için tasarlanmıştır. Şekil 2.5'de HLF blokzincir mimarisinin bileşenleri ve bağlantıları verilmiştir.



Şekil 2-5: Çok kanallı ve çok kuruluşlu HLF mimarisi [65]

HLF modüler, ölçeklenebilir ve genel amaçlı olacak şekilde geliştirilmiştir. Diğer birçok blokzincir platformlarında çalışma aşamaları Sırala-Yürüt (Order-Execute) şeklinde iken HLF'de Yürüt-Sırala-Doğrula (Execute-Order-Validate) şeklindedir [66]. Diğer birçok blokzincir platformlarından farklı bir yönü de ağda bulunan her düğümün blok oluşturamaması ve blok oluşturma ve işlemleri doğrulama işini yapan özel düğümlerin olmasıdır. Tüm doğrulayıcı düğümler diğer eş düğümlerle doğrudan iletişim kurdukları için HL Fabric eşler arası bir sistem olarak görülemez.

HLF mimarisinde işlemlerin gönderilme süreci hariç tutulduğunda, temelde üç aşamada işlenmektedir. İlk aşama, işlemin dağıtılmış akıllı sözleşme kodu (chaincode) ile yürütülmesi, ikinci aşama işlemlerin sıralanması ve son aşama ise işlem doğrulama ve taahhüt aşamasıdır. Bu mimari yapı, HLF 'in performansının optimize edilmesi ve doğrulama süreçlerinin daha hızlı yapılmasına yardımcı olmaktadır [67]. HLF'de işlemleri, Yürüt-Sırala-Doğrula (Execute-Order-Validate) adı verilen bir mimari yapıda işler. HLF'deki bu mimari yapının aşamaları ve sıralı akış diyagramı Şekil 2.6'da verilmiştir.



Şekil 2-6: HLF'de yürüt-sırala-doğrula aşamaları için sıra diyagramı [68]

Yürütme/Onaylama Aşaması (Endorsment):

- Bir istemci uygulaması üzerinden oluşturulan işlem teklifi, kanaldaki onay politikasına göre belirlenen onaylayıcı eş düğümlere gönderilir. İşlem teklifi içerisinde, işlem detayları ve içerik (txPayload) ile birlikte tanımlayıcı meta veriler; istemci kimliği (clientID) ve dijital imzası, akıllı sözleşme (chaincodeID), zaman damgası (timestamp) gibi bilgiler bulunur. Şekil 2.6'da (1) ile gösterilmiştir.
- Onaylayıcı eş düğümler, akıllı sözleşmeyi çalıştırarak işlem teklifinin geçerliliğini kontrol eder. Geçerlilik kontrolünde, istemcinin yetkisi ve imzası, işlem verilerinin sözleşme koşullarına uygunluğu ve varsa diğer dijital imzalar kontrol edilir.
- İşlem teklifinin geçerliliği onaylanırsa, onaylayıcı eş düğümler kendi dijital imzalarıyla onaylanmış işlemleri istemciye bir onay yanıtı (endorsement) mesajıyla gönderir. Şekil 2.6'da (2) ile gösterilmiştir.
- Kanalda tanımlı onay politikasında (endorsement policy) belirtilen gerekli ve yeterli onay sayısı elde edilirse, teklif edilen işlemin, iş kurallarına ve politikalarına uygun olarak doğrulandığı ve onaylandığı anlamına gelir ve mutabakatın ilk aşaması tamamlanmış olur.

Sıralama Aşaması (Ordering):

- İstemciyle bağlantılı eş düğüm de toplanan yeterli sayıda onaylanmış işlemler, onay ve işlem verileriyle bilirlikte sıralama hizmet düğümlerine (bir veya birden fazla sıralayıcı düğüm olabilir) gönderilir. Şekil 2.6'da (3) ile gösterilmiştir.
- Sıralayıcı düğüm kendisine farklı istemci uygulamalarından gelen tüm onaylı ve geçerli işlemleri kronolojik olarak sıralayıp blok içerisine koyar. Sıralayıcı düğümler bir bloğa belirli bir sayıda işlem geldiğinde, bloğun boyutu belirli bir sınır değere ulaştığında veya belirli bir süre geçtiğinde yeni blok oluştururlar. Bu süre ve değerlere ait sınırlılıklar konfigürasyon dosyasında tanımlanır. Performansın optimize edilmesinde ayarlanabilir.
- Sıralanmış işlemlerin olduğu bloklar ilgili kanaldaki tüm eş düğümlere dağıtılır. Bu aşamada tüm eş düğümlerin işlem ve blok doğrulamalarının ve zincir tutarlılıklarının mevcut mutabakat algoritmasına (Kafka, Raft, PBFT gibi) göre fikir birliğine varılarak aynı kayıt defterine sahip olmaları sağlanır.

Doğrulama ve Taahhüt Aşaması (Validation and Commitment):

- Sıralayıcı düğümlerden gelen yeni blokları alan tüm taahhüt edici (committing peers) eş düğümler ve onaylayıcı eş düğümler (endorsing peers) blokları kendi yerel kayıt defterine eklemeden önce, her işlem için onay politikasına göre yeterli onayı aldığını ve dijital imzaların geçerliliğini kontrol eder.
- Ayrıca, işlemlerde bir çakışmanın olup olmadığını İşlem teklifi ile sıralama aşaması arasındaki süreçte, işlem üzerinde bir değişikliğin olup olmadığını kontrol ederek, işlemlerin geçerli yada geçersiz olduğunu işaretleyerek kayıt defterine yazılmasını sağlar.
- Geçerliliği kabul edilmiş ve doğrulanmış işlemlerin yerel dünya durumu veritabanında işlenmesi ve bloğun yerel kayıt defterine eklenmesini sağlar.
- Her taahhüt eden eş düğümler işlemlerin başarılı bir şekilde kaydedildiğini yada geçersiz olduğunu istemci uygulamaya olay (Event) bildirimi ile duyurur. İstemci uygulamalar olay bildirimlerini takip ederek işlem sonuçlarını öğrenir. Şekil 2.6'da (4) ile gösterilmiştir.

2.2.1 Kuruluşlar (Organizasyonlar)

HLF 'te kuruluş (Org) temel bir bileşen olmakla birlikte mantıksaldır. Gerçek hayattaki şirket yada kurumları temsil eder. HLF ağında, her kuruluşun bir birimi yada şubesini temsil eden birden fazla eş düğümü olabilir. Sadece sıralayıcı düğümler bir kuruluşa ait değildir. HLF ağındaki düğümlerin mantıksal olarak gruplanarak, ağdaki kimliklerinin, yeteneklerinin ve

operasyonlarının tanımlanması olarak da düşünülebilir. Her kuruluş kendi düğümlerinden ve istemcilerinden sorumludur. HLF ağında her kuruluşun kendi düğümlerinin kimliklerini tanımlayabilmeleri ve işlemlerdeki yetki ve sınırlılıklarının belirlenebilmesi için soyut bir Üyelik Servis Sağlayıcı (MSP) ve Sertifika Yöneticisi (Org-CA) bileşenlerine sahip olması gereklidir. MSP sayesinde kuruluşun; eş düğüm, istemci, yönetici, kullanıcı gibi tüm üyelerinin kriptografik kimlikleri (x509 sertifikaları) tanımlanır ve gerektiğinde doğrulanır. Ağda oluşturulan onay politikaları ile Erişim Kontrol Listeleri (ACL) hangi kuruluşun hangi işlemleri yapma yetkisine sahip olduğu, kanal oluşturma ve zincir kodu dağıtımı gibi işlemlerdeki rolleri ve yetkileri belirlenir.

2.2.2 Eş Düğümler (Peers)

HLF'de eş düğümler, blokzincir ağının defter kaydını tutan, bu defter kaydı üzerinde, yazma yada sorgulama işlemlerini gerçekleştirmek için üzerinde zincir kodu uygulamasını barındıran temel bir bileşendir. Eş düğümler, ağdaki yapılanmaya göre farklı görevler ve roller üstlenebilirler. HLF 'in yapısal özelliğine göre temelde iki tür eş düğüm vardır, Bunlar; (1) ağın politikasına göre tanımlanan ve değişebilen Onaylayıcı Düğümler (Endorsing Peers) veya Taahhüt Eden Düğümler (Committing Peers). (2) Sadece sıralama ve mutabakata göre blok oluşturma görevini yerine getiren Sıralayıcı (Orderer) düğümler. HLF ağında tanımlı her kuruluşun yönetiminde birden fazla eş düğüm olabilmektedir. Kuruluşlara ait birden fazla eş düğüm, farklı görevleri yerine getiren birimler yada şubeleri temsil edebilir. Kuruluşa ait yada kuruluş dışındaki eş düğümlerle olan etkileşim ve iletişimleri sağlamada, bazı düğümlere tanımlanan Çapa Düğüm (Anchor Peer) ve Lider Düğüm (Leader Peer) gibi ek özel roller de bulunmaktadır. Eş düğüm türleri ve rolleri hakkında detaylı bilgiler aşağıda sunulmuştur [68].

- **Eş Düğümler (Peers):** Temelde, işlemlerin doğrulanması ve kayıt defterinin saklanması görevini yerine getirirler. Ağdaki eş düğümlerin hepsi işlemleri doğrulama sürecine katılmaz. İşlemlerin doğrulanması sürecine, bağlı bulunulan kanaldaki onay politikasına göre belirlenen eş düğümler onaylayıcı eş düğüm (Endorsing Peers) yetkisini alarak katılabilir. Diğer düğümler (Committing Peer) ise sıralayıcı düğüm tarafından gönderilen yeni blokları onaylanıp kayıt defterine ekleme görevini yapar.
- **Çapa Eş Düğüm (Anchor Peer):** Ağdaki farklı kuruluşlara ait eş düğümlerin tespit edilmesini ve iletişim kurulmasını organize eder. Her kuruluşun en az bir tane Çapa Eş Düğümü olması gereklidir. Çapa eş düğüm dedikodu (Gossip) protokolü üzerinden diğer

kuruluşların Çapa eş düğümleriyle iletişim kurar. Çapa eş düğümler kendi kuruluşlarında bulunan eş düğümler hakkındaki bilgileri birbirleriyle paylaşarak işlemlerin ve oluşturulan yeni blokların ağda yayılmasında, eş düğümler arasında doğrudan bağlantıların kurulmasında aracılık yapar.

- **Leader Eş Düğüm (Leader Peer):** Her kuruluş içinde Lider düğüm seçimi manuel yada dinamik olarak yapılır, eş düğümler hangi düğümün Lider düğüm olduğunu bilir ve gerektiğinde Lider düğüm seçimi tekrarlanarak iletişimdeki hata olasılığı azaltılır. Lider eş düğümler, Sıralayıcı düğümlerden aldıkları yeni blokların kuruluş içerisindeki taahhüt eden düğümlere Gossip protokolü üzerinden dağıtır. Böylece her eş düğümün kayıt defterinin güncel kalmasını sağlar. Ağdaki her düğümün Sıralayıcı düğümlere bağlanarak yeni blokları alması yerine, Lider düğümler üzerinden yeni blokların dağıtılması Sıralayıcı düğümlerin yükünü azaltır, ağın verimliliğini artırır.
- **Sıralayıcı Düğüm (Orderer Peer):** Ağda bir veya birden fazla kanalla bağlantılı bir veya birden fazla sıralayıcı düğüm her kanala ait işlemlerin ayrı ayrı sıralanması, blok içerisine alınarak paketlenmesi ve blokların ağda ve aynı kanalda bulunan diğer türdeki eş düğümlere dağıtılmasını sağlayan düğümlerdir. Sadece sıralayıcı düğümler blok oluşturabilir. Ağda birden fazla sıralayıcı düğümün olması tek nokta hatalarına karşı daha iyi olacaktır. Sıralayıcı düğümler yapısal olarak herhangi bir kuruluşa ait değildir, ancak ağın kurulumu ve yönetiminden sorumlu kuruluşun kontrolündedir.

Sıralayıcı düğüm ile eş düğümlerin farklı görevleri yerine getirmesi, işlemlerin onay ve kayıt sürecinde, sadece sıralayıcı düğümlerle diğer eş düğümler arasında bağlantının olması nedeniyle, HLF tam olarak merkeziyetsiz olarak kabul edilmediği gibi eşler arası iletişim yöntemine sahip (p2p) bir yapıda olduğu da söylenemez. Çünkü sıralayıcı düğümler, işlemleri sıralayarak blokların oluşturulmasında merkezi bir rol üstlenmektedirler. Ancak, farklı bir bakış açısıyla, eş düğümler arası özel veri paylaşımlarının olması ve işlemlerin doğrulama süreçlerine eş düğümlerin de katılmaları, HLF 'in kontrollü ve izinli bir eşler arası modele sahip olduğu söylenebilir.

2.2.3 Akıllı Sözleşme Zincir Kodu (Chaincode)

HLF'de akıllı sözleşmeler yerine zincir kodu (chaincode) tabiri kullanılır. Zincir kodu tüm düğümlere dağıtılmıştır ve ağdaki tüm kuruluşların ortak etkileşimlerini yöneten iş mantık ve

kurallarını içerir. Zincir kodları eşler arası işlemlerin gerçekleştirilmesi, istemci uygulamadan gönderilen ve onaylanan işlemlerin kayıt defterine ve durum veri tabanına yazılmasını yada durum veri tabanından sorgulamaların yapılarak defterle özdeş bilgilerin okunmasını sağlar. Ayrıca HLF çekirdek yazılımında ağ bileşenlerinin kurulması, başlatılması, güncellenmesi işlemleriyle birlikte ağın yönetiminde Sistem Zincir Kodu kullanılmaktadır. HLF'de zincir kodu programı yazmak için Java, Go ve Javascript dilleri desteklenmektedir.

2.2.4 Özel Veriler

HLF 'in özel ve izinli yapısı, özel verilere erişim için esnek bir kimlik sistemi, zincir kodu ve sağlam bir gizlilik politikası sağlar. Bazı iş senaryolarında veri gizliliği önemliyse, ağdaki bazı kuruluşlar arasında gizli kalması gereken veriler ve işlemler varsa, bu durumda Özel veri Koleksiyonu yöntemi kullanılabilmektedir. Bu yöntemde özel verilerin kayıt defterine doğrudan yazılması yerine özel verilerden elde edilen özet değeri (hash) yazılır ve asıl veri ilgili kuruluşların kendi özel durum veri tabanlarında tutulur. Böylece özel verilerin gizliliği sağlanırken bütünlükte korunmuş olur. Üyelik hizmeti sağlayıcısı tarafından verilen kimliklerle yalnızca yetkili kullanıcıların özel verilere erişebilmesini sağlanarak erişim kısıtlanabilir.

2.2.5 Onay Politikası (Endorsement Policies)

Blokzincir ağında ve aynı kanalda bulunan eş düğümlerden hangilerinin işlem onayı gerçekleştirebileceğini ve hangi zincir kodlarını onaylayabileceğini, okuma/yazma işlemlerindeki yetkileri belirleyen kurallardır. Onaylayıcı eş düğümler, çalıştırılan zincir kod sonuçlarına göre işlemlerin geçerliliğini belirler ve dijital imzasıyla onaylar. Bu politikalar, dijital kimliklerin özelliklerinden faydalanarak ağdaki kaynaklara erişimi kısıtlamak için kullanılır. Erişim Kontrol Listeleri (ACL) yapılandırmasıyla bir zincir kodu fonksiyonlarını kimlerin kullanabileceği belirlenir. Bu yapılandırmalar ağ başlatılmadan, kanal oluşturulmadan önce "configtx.yaml" ağ konfigürasyon dosyasında tanımlanır. Onay politikaları, mantıksal AND, OR ifadeleri ile tanımlanır. Örneğin, OR(Org1.peer, Org2.peer) ifadesi, işlem onay sürecinde, Org1 yada Org2 den sadece bir eş düğümün onayının yeterli olduğunu tanımlar. OutOf(2, 'Org1.peer, 'Org2.peer, 'Org3.peer) şeklindeki bir onay politikası, 3 kuruluştan en az ikisinin onayının gerekli olduğunu tanımlar.

2.2.6 Kanal (Channel)

HLF ağında yer çok sayıdaki alan kuruluş ve düğümlerin kendi aralarında grup oluşturması ve grup üyelerinin kendi arasında iletişim kurması için çoklu kanal yapılandırmasına desteklemektedir. Bu yönüyle birçok blokzincir platformlarından ayrılan HLF, her kanal için farklı politikaların uygulanması ve farklı kayıt defterleri tutulmasını sağlar. Kuruluş ve eş düğümlerin birden fazla kanala üye olması da mümkündür. Ancak, akıllı sözleşme kodları her kanal için farklı kayıt defterleri tutulduğu için sadece tanımlı olduğu kanalda çalıştırılabilir [68]. Akıllı sözleşme kodlarının diğer kanal kayıt defterlerine doğrudan erişimi yoktur. Ancak, zincir dışı (off-chain) bağlantı ve uygulamalarla diğer kanallar arasında veri alışverişi yapılması mümkündür.

2.2.7 Varlık (Asset)

HLF de varlık, temsil edilen yada yönetilen herhangi bir soyut yada somut bir değeri ifade eder. Bu değer, otomobil, ev, gayrimenkul, meyve-sebze gibi bir mal ya da fikri mülkiyet hakları gibi soyut kavramlar olabilir. HLF de zincir kodları ile varlıklar oluşturulur ve yönetilir. HLF de varlık, anahtar-değer (key-value) çifti olarak temsil edilir ve hem kayıt defterinde, hem de durum veritabanında bu şekilde depolanır. Anahtar (key), varlığın benzersiz bir tanımlayıcısı olabilir. (Örneğin bir arabanın şasi numarası, bir ürünün seri numarası vb.) Değer (value), varlığın özellikleri veya durumu ile ilgili bir değişikliği ifade edebilir. Değer ifadesi bir JSON, XML veri yapısı, ikili bir veri bloğu yada başka bir tanımlanabilir formatta olabilir. Kayıt defterleri sadece varlıkla ilgili anahtar-değer kaydını tutar ve kayıtlar silinemez. Varlıklarla ilgili kayıt defterinde sadece okuma ve yazma işlemleri yapılabilir. Silme ve değiştirme işlemleri yapılmaz. Ancak silinmesi ve değiştirilmesi gereken bir varlık için kayıt defterine en son girilen işlem bilgisi önceki varlığın nitelik ve özellik olarak durumunun değiştiğini bildirebilir ve önceki kaydın geçersizliğine kanıt olabilir. Kayıt defterleri, varlıklara ait işlem kayıtlarını kronolojik olarak tutmakta ve varlıkların yaşam döngüsünün yönetilmesinde kullanılmaktadır.

2.2.8 İstemci Uygulaması (Client Application) ve Kullanıcılar:

İstemci uygulaması, HLF blokzincir ağına dışarıdan erişmek ve etkileşim kurmak için kullanılan yazılımlardır. İstemci yazılımları Node.js, Go, Java, Python gibi programlama dillerinde ,web, mobil yada tasarlanmış özel bir program olarak yazılabilir ve Fabric SDK üzerinden

blokzincir ağıyla etkileşime girebilir [68]. İstemci yazılımlarının, HLF ağıyla iletişim kurabilmesi için geçerli bir kimlik ve sertifika sunması gerekir. Bu kimlik ve sertifika iletişim kurulacak eş düğümün ve organizasyonun Üyelik Servis Sağlayıcı (MSP) bileşeni tarafından verilir ve geçerliliği belirli bir süreye göre ayarlanır ve istenildiğinde iptal edilebilir. Genelde sunulan kimlik ve sertifika bir son kullanıcıyı (end user/client user) tanımlar. Kullanıcılar HLF ağında sadece sundukları sertifikaları ile bilinir, dolayısıyla gerçek kimlikleri gizli kalır.

2.2.9 Kayıt Defteri (Ledger)

Varlıklarla ilgili kayıtların, zaman damgalı, sıralı, değişmez, kurcalamaya karşı dayanıklı, kriptografik olarak birbirine bağlı blok ve işlem kaydının tutulduğu dosyayı ifade eder. Ağdaki tüm düğümlerin aynı kayıt defterine sahip olması için çeşitli senkronizasyon ve fikir birliği mekanizmaları kullanılmaktadır. Ağdaki güven ve şeffaflık kayıt defterinin her düğümde aynı ve değişmez olmasındandır. Kayıt defterleri, varlıklara ait işlem kayıtlarını kronolojik olarak tutmakta ve varlıkların yaşam döngüsünün yönetilmesinde kullanılmaktadır. HLF de kayıt defterinde yapılan işlemin son güncel durumu durum veritabanı (state database) adı verilen (dünya durumu (world state) olarak da isimlendirilir) bir bileşende de tutulmaktadır. HLF de her kanal için ayrı bir kayıt defteri tutulmaktadır. Sadece kanal üyeleri kendi kanalında tutulan kayıt defterine erişim sağlayabilir [68].

2.2.10 Durum Veritabanı

Kayıt defterine kaydedilen her işlemin mevcut durumuna ait anahtar - değer çifti şeklinde kaydının tutulduğu veri tabanıdır. HLF'de sadece LevelDB ve CouchDB kullanımı desteklenmektedir. Ağın kurulumunda hangisinin kullanılacağı seçilmesi gereklidir. LevelDB eş düğüm içerisinde gömülü olarak yer alır, ancak CouchDB için ayrı bir yapılandırma yapılması gerekir. HLF test ağlarında Docker konteyner olarak kullanılır. LevelDB üzerinde tutulan kayıtlarda, anahtar değeri, belirli bir aralıktaki kayıtlar ve bileşik anahtar gibi sorgulama seçeneklerini destekler. CouchDB farklı olarak, işlem kayıtlarının JSON veri yapısında depolanmasını sağladığı gibi daha kapsamlı sorgulamalar yapılmasını sağlar. Daha büyük veri kümleri içerisinde, dizin ve index özellikleri kullanılarak daha verimli sorgulama gereken uygulamalarda tercih edilir.

2.2.11 Sertifika Yöneticisi (CA) ve Üyelik Servis Sağlayıcı (MSP)

HLF ağında her katılımcının; ister bir kuruluş, ister bir ağ düğümü (peer/orderer) ya da bir son kullanıcı olsun, dijital bir kimliğe sahip olması şarttır. Bu kimlikler, X.509 sertifikaları veya Identity Mixer (Idemix) teknolojisi aracılığıyla sağlanır. Bu sertifikalar, Fabric ağındaki kullanıcıların ve düğümlerin yetkilerini ve özelliklerini tanımladığı için son derece önemlidir. Bir sertifikanın güvenilirliği, o sertifikayı veren sertifika otoritesinin (CA) güvenilirliğiyle doğrudan ilişkilidir. HLF mimarisinde sertifika üretimi ve dağıtımı için genellikle Fabric CA tercih edilir. Ancak, Fabric esnek ve modüler bir yapıda olduğu için OpenSSL veya diğer güvenilir sertifika otoriteleri (Trusted CA'lar) gibi farklı araçlar da kullanılabilir. Bu kimliklerin ve yetkilendirmelerin yönetimi ise Membership Service Provider (MSP) bileşeni tarafından yapılır. Her kuruluşun kendi MSP'si bulunur ve bu MSP, kuruluşun üyelerine ait sertifikaları doğrular, yetkilendirme politikalarını uygular ve ağdaki güvenliği sağlar [68].

2.2.12 Zincir Dışı (Off-Chain) ve Zincir İçi (On-Chain) Veriler

Zincir içi veriler, kayıt defterinde saklanan işlem verilerini ifade eder. Zincir dışı veriler ise, blokzincir ağı dışında depolama, veri taşıma ve işleme durumlarını ifade eder. Akıllı sözleşmeler, zincir dışı verilerle Oracle adı verilen bir yöntemle etkileşime girebilir ve veri alışverişi yapabilir. Zincir dışı işlemlerle, Oracle yöntemi ve DLT aracılığıyla IPFS ağları yada başka bir zincir defterlerine veri aktarımı yapılabilir [69].

2.2.13 Hyperledger Fabric Özellikleri

- HLF, İzinli türde bir blokzincir platformudur. Herhangi bir düğüm yada kullanıcının ağa katılabilmesi ve okuma yada yazma işlemlerinin yapılabilmesi için kimlik ve sertifika yetkisi gereklidir.
- Belirli düğümlerin belirli kanallar üzerinden etkileşim kurmasına izin verildiği ve kanala ait kayıt defterini yalnızca kanaldaki yetkili düğümler görebildiği için işlemlerin gizlilik ve mahremiyeti korunmuş, erişim sınırlandırılmış olur.
- HLF modüler ve yapılandırılabilir bir mimariye sahiptir. Mutabakat mekanizması, sertifika yöneticisi, takılabilir ve değiştirilebilir özelliktedir. Yeni organizasyon ve düğümlerin eklenip çıkarılabilmesi, zincir kodlarının güncellenebilmesi gibi esnekleri vardır [68].

- Durum veritabanı olarak LevelDB ve CouchDB desteği vardır. Kayıt zincirinin dünya durumu olarak sürekli güncellenen CouchDB ile daha geniş kapsamlı ve verimli bir şekilde sorgulamalar yapılabilir. Böylece HLF 'in sorgu ve okuma performansı artırılabilmektedir.
- HLF ölçeklenebilir bir yapıya sahiptir. İşlem yükü eş düğüm ve sıralayıcı düğümlere paylaştırılmıştır. Eş düğümler yürütme, doğrulama ve defter kaydı tutma işlemlerini yerine getirirken, işlemleri sıralama ve blok oluşturma ile fikir birliği süreçleri sıralayıcı düğümler tarafından yapılır. Böylece performans artışı sağlanarak daha yüksek işlem hacmi sağlanabilmektedir.
- HLF'de madenci düğüm kavramı yoktur. Blok oluşturma sıralayıcı düğümler tarafından yapılır. Böylece işlemlerin onaylanması ve doğrulanması süreçlerinde işlem gecikme süreleri azalmış ve sistem daha hızlı hale gelmiş olur. Mevcut izinli blokzincir türleri arasında en hızlı platform olarak kabul edilir.
- Akıllı sözleşmelerin programlanması için Java, Go ve Node.js/Javascript gibi farklı programlama dilleriyle akıllı sözleşme kodları yazılabilmektedir [68]. Özel bir dil gereksinimi yoktur.
- İzinsiz ve halka açık blokzincir platformlarından farklı olarak, HLF'de kripto para birimi kullanılmaz [69].

2.3 IPFS, DAĞITIK DOSYA DEPOLAMA

IPFS (Interplanetary File System), Merkezi olmayan, eşler arası (P2P) iletişim yöntemi ile dağıtık bir ağ ortamında dosyaların paylaşılması ve depolanmasını sağlayan bir protokoldür. 2015 yılında Juan Benet'in öncülüğünde Protocol Labs tarafından proje olarak geliştirilmeye başlanmıştır [Juan Benet 2014]. İnternetin daha güvenli, sansüre karşı korumalı ve hızlı olmasını sağlamak amaçlanmıştır. Bu nedenle Tablo 2.1'de gösterildiği gibi, internet ortamında kullanılan

URL tabanlı adresleme yerine İçerik Kimliği (CID, Content Identifier) ile adresleme yöntemini geliştirilmiştir. Böylece, internet ağ ortamındaki bir dosyanın bulunduğu konuma göre değil dosya içeriğinden oluşturulan kriptografik özet (hash) değerine göre erişim sağlanmaktadır. URL de kullanılan DNS (Domain Name System) sistemine benzer IPNS (InterPlanetary Naming System) ad hizmeti ile dosyanın yeni versiyonlarına da erişilebilir. Dosya IPFS ağına yüklendiğinde dosya içeriğinden benzersiz bir CID değeri oluşturulmakta, bu değer hem dosya adı hem de dosyaya erişim amacıyla kullanılmaktadır [70]. Bu yöntem aynı dosyanın yeniden yüklenmesini ve ayrı bir dosya gibi birden fazla kopyası olarak depolanmasını engellediği gibi, dosya içeriğinde meydana gelecek bir değişiklik CID değerini de değiştireceği için bütünlüğün korunması ve kontrolünü de kolaylaştırmaktadır.

Tablo 2-1: Dosya erişiminde URL ve CID yöntemlerinin karşılaştırılması

URL :	http://www.example.com/resim.jpg
CID :	ipfs:// QmX7G4zYQHhMgTn6t9V9JxwktCyuwjHmbBUgW4fxy977Ah yada örnek bir gateway ile; https://ipfs.io/ipfs/QmX7G4zYQHhMgTn6t9V9JxwktCyuwjHmbBUgW4fxy977Ah

IPFS de dosyalar küçük ve yönetilebilir 256 Kb'lık bölümlere ayrılır, özet değeri elde edilir, diğer bölümlerle bağlantı bilgileri oluşturularak IPLD (InterPlanetary Linked Data) nesneleri halinde düzenlenir. Her bölümün içeriğine ait özet (hash) değeri Merkle DAG (Directed Acyclic Graph) adında bir veri yapısında ilgili bölüme ait meta verileriyle birlikte saklanır. Merkle DAG içindeki bölümler arasındaki bağlantı için UnixFS (Unix File System on IPFS) adı verilen bir veri biçimi kullanılmaktadır. Bu veri biçimi dosya, dizin, boyut ve bağlantı gibi çeşitli diğer meta bilgilerini tutar. UnixFS de tutulan veriler, dosyanın veri bölümlerinin nasıl düzenlendiğini ve birleştirilerek orijinal dosyanın yeniden oluşturulacağını tanımlar [71]. IPFS de dosyanın bölümlere ayrılması veri bütünlüğünün kontrol edilmesinde ve dosyanın farklı versiyonlarının arasındaki farklıkların belirlenmesi ve takibinde oldukça fayda sağlamaktadır.

IPFS de bir dosyaya erişim sağlanmak istendiğinde aşağıdaki adımlar takip edilir:

- ***CID ile dosya talebi:*** Yerel IPFS düğümü talep edilen ve CID'si bilinen dosya bölümleri eğer kendisinde varsa doğrudan dosyaya erişim sağlanır. Yerel düğümde dosya yoksa, bir IPFS istemcisi ya da IPFS ağ geçidi (gateway) ile dosyanın CID bilgisi ile erişim talebi yapılır.

- ***DHT ((Distributed Hash Table) sorgusu:*** IPFS ağı ile bağlantılı yerel IPFS düğümü, aranan CID ye sahip düğümleri bulmak için, IPFS ağındaki düğümlerin hangi içerikleri depoladığının indeks kaydını tutan DHT'ye sorgu gönderir.
- **DHT Cevabı:** Ağdaki DHT sorgulanan CID bilgisine sahip uzak düğümlerin ağ adres bilgilerini tespit ederek, sorgulamanın yapıldığı yerel IPFS düğümüne cevap bilgisi olarak gönderir.
- ***Uzak düğüm bağlantısı ve dosya talebi:*** Adresi öğrenilen uzak düğümlerle doğrudan bağlantı kurularak istenen CID'ye ait veri bölümleri talep edilir. Dosya bölümleri birden fazla düğümde olabileceğinden, en hızlı yada en yakın olanlardan veriler alınmaya çalışılır.
- ***Uzak düğümden dosya bölümlerinin alınması:*** Uzak düğüm istenen CID'ye ait sahip olduğı dosya bölümlerine yerel IPFS düğümüne gönderir.
- ***Yerel düğümde dosya bölümlerinin toplanması ve birleştirilmesi:*** Yerel IPFS düğümü talep ettiği dosyaya ait bölümleri farklı uzak ağ düğümlerinden almış olabilir. Merkle DAĞ'da kayıtlı tüm bölümlerin kontrollerini yaparak topladığı bölümleri birleştirir ve meta verileri ile birlikte dosyaya erişim sağlanmış olur.

IPFS, internet ortamında çalıştırılmak için oluşturulmuş olsa da özel bir ağ ortamında da çalıştırılabilmektedir. Böylece depolanan dosyaların yalnızca ağdaki belirli bir grup tarafından erişilebilir olması sağlanabilir. Ortak bir gizli anahtar üzerinden IPFS düğümlerinin birbirleriyle iletişim kurması sağlanabilir. Genel IPFS ağında kullanılan MDNS (Multicast DNS) keşif hizmeti devre dışı bırakılıp, başka IPFS ağlarından izole edilerek, ortak özel anahtar ve belirli düğümlerle bağlantı izni gibi yapılandırmalar yapılarak özel bir IPFS ağı kurulabilir. Özel IPFS ağlarında her düğümün kimliği diğer düğümlere bildirilerek bağlantı kontrolü ve güvenlik sağlanabilmektedir.

Blokzincir platformları, blok yapıları ve kriptografik işlem doğrulama süreçleri nedeniyle boyutu büyük işlemler için kullanımında performans kaybı ve iletişim sorunları oluşturabileceği için uygun değildir. Bu nedenle IPFS, DLT ile entegre edilerek zincir dışı şifreli depolama amacıyla kullanılabilir. IPFS teknolojisinin sahip olduğu özelliklerine rağmen, gizlilik endişeleri vardır. Halka açık IPFS çözümlerinde, kaydedilen bir içeriğe erişilebilir ve alınabilir. Bu nedenle, IPFS ağına yüklenecek bilgilerin önceden şifrelenmesi, şifrelenmiş içeriği ait CID bilgisi ile şifre çözme anahtarlarının blokzincir ağlarında saklanması, akıllı sözleşmelerle erişimin kontrollü ve güvenli olması sağlanabilir. Bir başka çözüm de, özel IPFS ağı ile sadece ağdaki tarafların erişiminin denetimli olarak sağlanması ile gizlilik ve mahremiyet güvence altına alınabilir.

3. BLOKZİNCİR TABANLI ARAÇ KİMLİK VE SİCİL SİSTEMİ (BAKSİS) MODEL ÖNERİSİ

Bu bölümde tez çalışmasında önerilen Blokzincir Tabanlı Araç Kimlik ve Sicil Yönetim Sistemi (BAKSİS) ile araç kimlik ve sicil bilgilerinin güvenli bir ortamda depolanmasını ve yönetilmesini amaçlayan bir sistem tasarlanmış ve simülasyon uygulaması geliştirilmiştir. Tasarlanan modelin bileşenleri ve simülasyon uygulaması için, önce arka plan teknoloji ve kavramlar hakkındaki bilgiler aşağıdaki gibi açıklanmıştır. Sonrasında önerilen BAKSİS modelinin detayları ve simülasyon uygulamaları adım adım açıklanmıştır.

3.1 ÖNERİLEN BAKSİS MODELİ

Tez kapsamında ele alınan araç yaşam döngüsü içerisinde, her aracın kimliklendirilmesi ve sicil kayıtlarının güvenle depolanmasını ve paylaşılmasını sağlamak için, devlet otoritesinin önemli olduğunu, birçok merkezi yapıya sahip özel şirketlerle işbirliğinin denetimli ve kontrollü olması gerektiğini düşündüğümüz için, özelleştirilebilir birçok özellikleriyle birlikte tak çalıştır ve modüler bir yapıda olması nedeniyle HLF blokzincir platformu seçilmiştir.

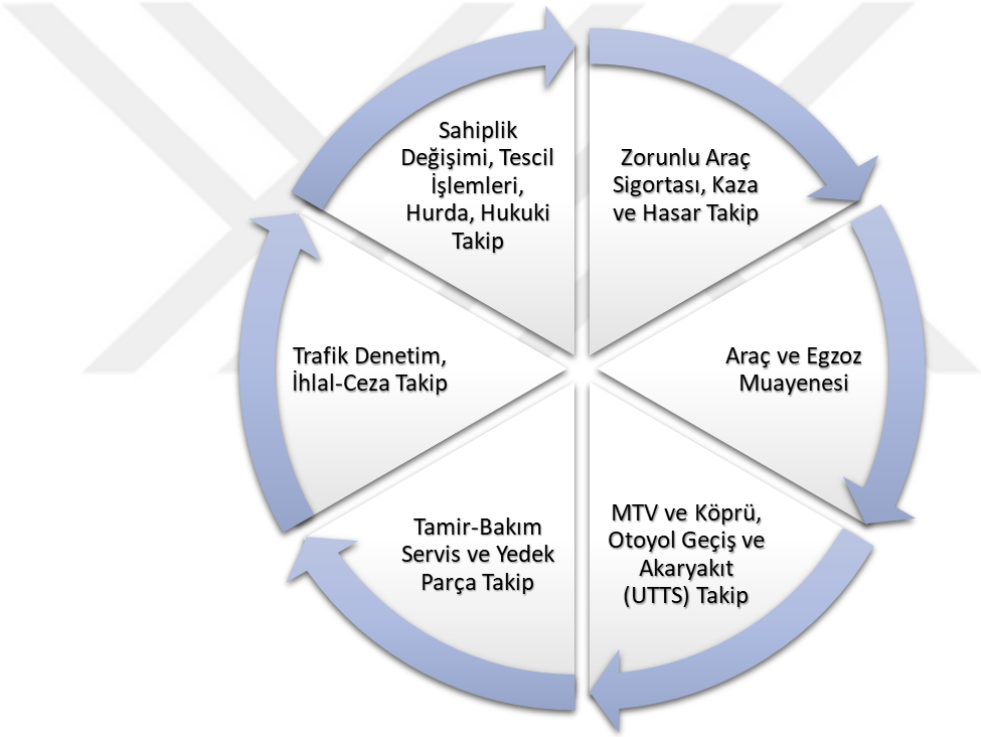
Açık kaynaklı olması nedeniyle herhangi bir maliyet getirmeyen Fabric'te, genel amaçlı programlama dilleriyle zincir kodu (chaincode) adı verilen akıllı sözleşmelerin yazılabilmesi esnekliğine sahiptir. Fabric 'in diğer avantajları da, çoklu kullanıcı desteği olan organizasyonların olması, birden fazla kanal oluşturulabilmesi, kontrol edilebilir erişim ve organizasyonlar arası iletişimin veri izolasyonu ve gizliliği ile sağlanabilmesine imkan veren özel veri koleksiyonu özellikleridir. Tez kapsamında hazırlanan akıllı sözleşme kodları ve istemci uygulaması geliştirmek için Node.js programlama dili kullanılmıştır.

HLF blokzincirin temel bileşenlerinden olan, varlık (asset), organizasyon ve düğümler, kullanıcılar, varlık üzerinde gerçekleştirilen işlemler için akıllı sözleşmeler, onay politikaları ve sistem gereksinimleri belirlenmesi üzerine çalışmalar yapılmıştır. Sonraki bölümlerde detayları verilen bileşenlerin tezin hedeflerine ulaşmak için hangi özellik ve hizmetleri yerine getirmesi gerektiği anlatılmaktadır.

Blokzincir platformlarında soyut yada somut bir değerin yani dijital olarak tanımlanan bir varlığın üzerinde yapılan işlemlerin kayıt altına alınması esastır. Kripto para çeşitleri değerli bir varlık olarak kabul edilerek, transfer ve ödeme gibi işlemlere ait kayıtlar blokzincir ağlarında depolanmaktadır. Önerilen blokzincir ağında, somut bir değeri olan motorlu kara taşıtları (araçlar)

varlık olarak kabul edilmiştir ve araçların tanımlanan benzersiz kimlikleriyle yapılan işlemler akıllı sözleşmelerle yönetilerek blokzincir ağında depolanmaktadır.

Araçların varlık olarak kabul edilmesinden sonra, araçlarla ilgili işlem çeşitliliği, işlemleri kimlerin (kullanıcı) yapabileceği ve yapılan işlemlerin blokzincir ağında hangi düğümler tarafından onaylanıp kayıt altına alınacağının belirlenmesi gerekmektedir. Öncelikli olarak işlem çeşitliğinin araştırılması ve bu işlemlerin yapılmasında rol alan tarafların belirlenmesi için araştırmalar yapılmış, elde edilen bulgulardan, araçların yaşam döngüsü boyunca yapılan işlemleri sınıflandırılarak Şekil 3.1'de gösterildiği gibi özetlenmiştir.



Şekil 3-1: Araç yaşam döngüsünde sınıflandırılmış işlemler

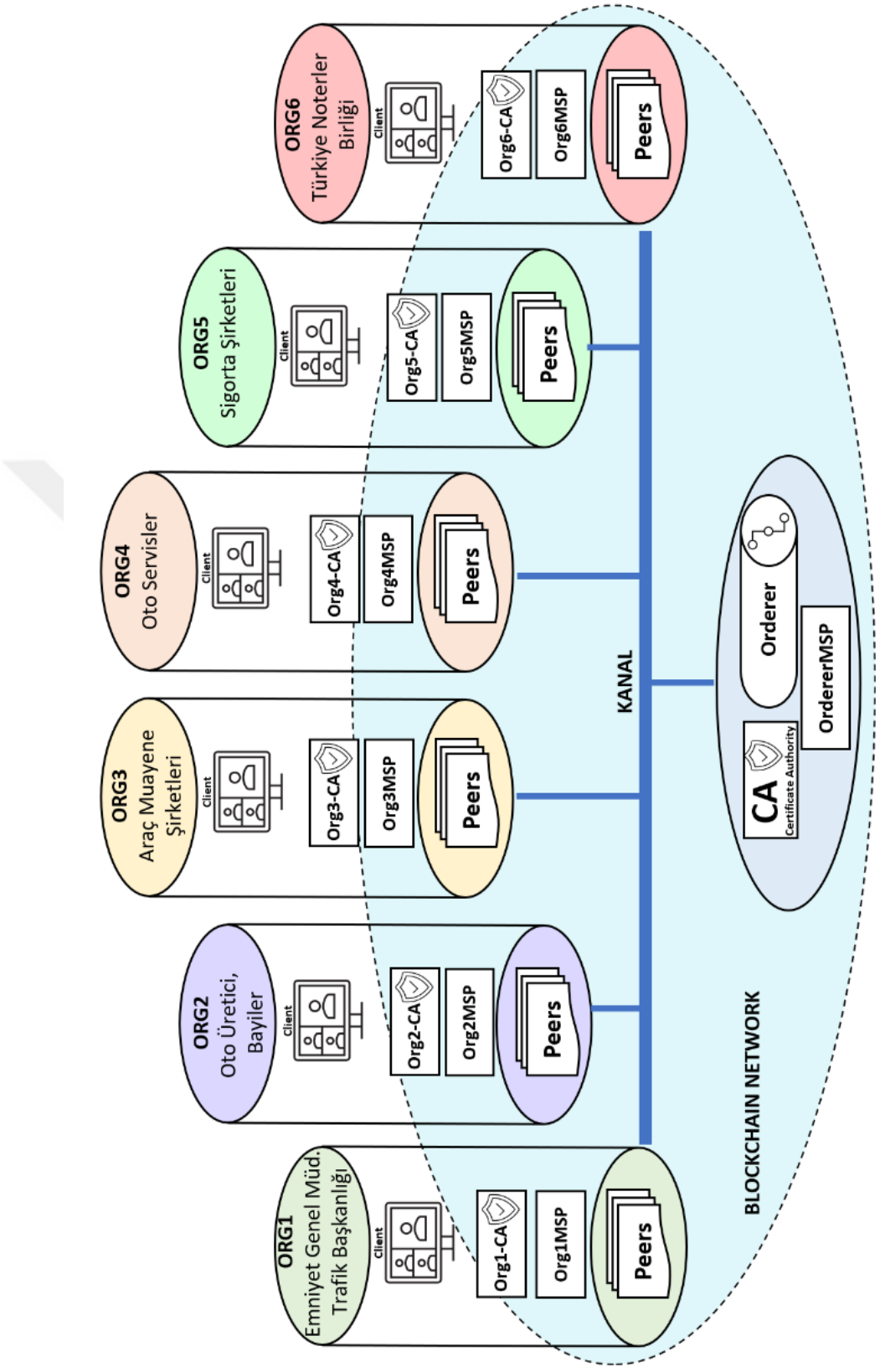
Araçların ilk tescil tarihinden itibaren, kullanılmaz hale gelip hurdaya ayrıldığı tarihe kadarki yaşam döngüsü içerisinde gerçekleştirilen işlemlerde rol alan, mevcut resmi ve özel kurumlar belirlenerek, yetki ve sorumluluklarıyla birlikte Tablo 3.1' de verilmiştir.

Tablo 3-1: Önerilen modeldeki kuruluşlar ve görevleri

Kurum Adı	**Yetkili ve Sorumlu Olduğu İşlemler**
Emniyet Genel Müdürlüğü Trafik Başkanlığı	- Araç tescil, ruhsat ve plaka işlemleri - Denetim, yasal ve hukuki takip - Kaza ve suç takibi - Trafik ceza ve vergi borç takibi
Türkiye Noterler Birliği	- Araç tescil, ruhsat işlemleri - Tadilat ve dönüşüm tescil işlemleri - Hurda takip - Yasal ve hukuki işlem takip - MTV, ÖTV vergi ve Ceza borç takip
Türkiye Sigorta Birliği, Sigorta Şirket ve Acenteleri	- Zorunlu trafik sigortası poliçe işlemleri - Kasko sigortası poliçe işlemleri (varsa) - Ekspertiz ve hasar takip, - Kaza ve Sigorta ödemeleri
Araç Üretici ve Bayileri	- Üretim bilgileri, teknik veri sağlama - Geri çağırma işlem takipleri - Araç garanti takip ve servis hizmetleri - Araç ilk tescil işlemleri
Oto Tamirhane ve Özel Servisler	- Bakım - onarım ve tamir hizmetleri - Yedek parça değişimi
Araç Muayene Şirketleri	- Araç muayenesi ve uygunluk kontrolü - Trafik ceza ve vergi borç sorgulamaları - Egzoz Muayenesi

Araçlarla ilgili işlemlerde tabloda verilenler haricinde doğrudan yada dolaylı birçok kuruluş ve kullanıcı rol alabilmektedir. Tez kapsamında en temel işlemlere ait roller belirlenerek tasarım yapılmıştır. HLF ağının ölçeklenebilirlik özelliklerinden olan sonradan yeni organizasyon ve düğümlerin eklenebilmesi, ihtiyaç olduğunda ağın genişletilebilmesine, yeni işlem ve hizmetlerin eklenebilmesine imkan vermektedir.

HLF mimarisine ve bileşenlerine uygun olarak Şekil 3.2'de verilen model tasarlanmıştır. Bu modelde yer alan bileşenlere ait detaylar sonraki bölümlerde verilmektedir.



Şekil 3-2: Önerilen BAKSİS blokzincir ağ modeli

3.2 BAKSİS MODELİNİN BİLEŞENLERİ VE İŞLEVLERİ

3.2.1 Hyperledger Fabric Blokzincir Ağı

Hyperledger Fabric genelde kurumsal blokzincir uygulama ve projelerinde tercih edilen açık kaynak kodlu bir DLT platformudur. Diğer blokzincir platformlarından birçok farklı özelliklere sahiptir. Modüler ve konfigüre edilebilir bir yapıda olan Fabric, izinli bir yapıya sahip olmasıyla devlet otoritesinin önemli olduğu uygulamalarda, tarafların kimliklerinin biliniyor olmasıyla daha güvenli ve denetlenebilir bir işbirliğinin sağlanmasını kolaylaştırmaktadır [72].

3.2.1.1 Kuruluşlar ve Eş Düğümler

Tez çalışmasında araçlarla ilgili işlemlerde rol alan çok sayıda kuruluş olsa da, temel işlemler için 6 adet kuruluş belirlenmiştir. HLF ağı hem esnek hem de modüler bir yapıya sahip olduğu için sonradan ağa yeni kuruluşlar ve eş düğümleri eklenebilmektedir. Her kuruluş kendi eş düğümlerini, istemcilerini ve kullanıcılarını yönetmektedir. Dolayısıyla her kuruluşun kendi belirlediği bir sertifika sağlayıcısı tarafından, eş düğümleri, istemcileri ve kullanıcıları için yönetilebilir bir dijital sertifika ve dijital imza tanımlamalarını yapması ve bu bilgileri güvenli bir ortamda saklaması oldukça önemlidir.

Ağın konfigürasyonunda, onay ve erişim politikalarının belirlenmesinde, yeni kuruluşların eklenmesi, kanal ve sıralayıcı düğümlerin yapılandırılması gibi yönetim işlemlerinde yönetici kuruluşun belirlenmesine ihtiyaç vardır. Önerilen Tasarımda bu kuruluş resmi bir otoriteyi temsil eden Emniyet Genel Müdürlüğü Trafik Başkanlığı olarak belirlenmiştir. Diğer kuruluşlar ve işlevleri aşağıdaki listede verilmiştir.

Kuruluşlar:

- ***ORG1 : Emniyet Genel Müdürlüğü Trafik Başkanlığı:*** Devlet otoritesini temsil eden bu resmi kuruluş, ağın genel yapılandırması ve yönetiminden sorumlu olmakla birlikte, araç tescil ve plaka işlemleri, trafik kontrol, ihlal ve ceza takip işlemleri, kaza ve suç takip işlemleri gibi faaliyetleri organize eder.
- ***ORG2:Araç Üreticileri ve Bayileri:*** Araçların ilk satış ve tescil işlemleriyle birlikte, garanti ve servis hizmetlerinden sorumludur.

- ***ORG3:Araç Muayene Şirketleri:*** Araç muayenesi, egzoz muayenesi ve uygunluk kontrol ve onay işlemlerinden sorumludur.
- ***ORG4:Oto Servis ve Tamirhaneler:*** Bu kuruluş birçok oto tamirhane ve özel servis kullanıcılarının üyelik, yetki ve sorumluklarını düzenleyen işlemleri yerine getirmekle birlikte, araçlar üzerinde yapılan her türlü bakım onarım faaliyetlerinin kayıtlarını sisteme girmekten sorumludur.
- ***ORG5:Türkiye Sigorta Birliği:*** Zorunlu trafik sigortaları ile kasko sigortası kapsamında oluşturulan poliçe işlemlerinde, ekspertiz ve hasar takip ve ödeme işlemlerinde sorumludur.
- ***ORG6:Türkiye Noterler Birliği:*** Araçların satış, devir, tescil işlemlerinde sorumludur.

Eş düğümler (Peers):

Kuruluşların ağda en az bir eş düğüme sahip olması gereklidir. Eş düğümler sayısı kuruluşların ihtiyaçlarına bağlı olarak belirlenebilir. Her ne kadar eş düğüm sayısının fazlalığı HLF ağları için performansı düşüren bir faktör olsa da ağın daha güvenli ve yedekli olması, yük dengelemesinin sağlanması amacıyla çok sayıda düğüm kullanılması tercih edilebilir. Ülke genelinde faaliyet gösteren yukarıda isimleri listelenen kuruluşların ihtiyaç ve gereksinimlere bağlı olarak il veya bölge kapsamında yeterli sayıda düğüme sahip olması, bu düğümlerden hangilerinin onaylayıcı düğüm hangilerinin sadece kayıt defterini tutan taahhüt edici düğüm olduğu belirlenerek ağın performansı optimize edilebilir.

Tez kapsamında gerçekleştirilen simülasyon uygulamasında her kuruluş için 3 adet eş düğüm tanımlanmış ve toplamda 18 eş düğümden oluşan bir ağ oluşturulmuştur. Her kuruluşun ilk eş düğümü istemci uygulamalarıyla bağlantı kurup etkileşime girecek çapa eş düğüm olarak, ağın temel konfigürasyon dosyasında tanımlanmıştır. Her eş düğüm onaylayıcı ve taahhüt edici rollerine sahip olup akıllı sözleşme zincir kodları her eş düğüme dağıtılmıştır.

Sıralayıcı Düğüm (Orderer):

Sıralayıcı düğümler HLF ağının temel bir bileşeni olup herhangi bir organizasyona ait değillerdir. Birden fazla sıralayıcı düğümün olması olası arızalara ve tek nokta hatalarına karşı ve yedeklilik sağlaması açısından önerilmektedir. Sıralayıcı düğümler, Çökme Hata Toleransı (Crash Fault Tolerance, CFT) protokolü olan RAFT mutabakat protokolünü kullanmaktadır. Ağın gereksinimlerine göre sıralayıcı düğüm sayısı artırılabilir. Bu durumda lider sıralayıcı düğüm seçimi RAFT mutabakat algoritmasına göre belirlenecektir. Önerilen modelde sıralayıcı

düğümlerin güvenliği oldukça önemli olduğundan kontrol ve yönetim ağın genel yönetiminden sorumlu Emniyet Genel Müdürlüğü Trafik Başkanlığı (ORG1) kuruluşuna verilmiştir.

3.2.1.2 Kullanıcı tanımı ve sertifikalar

HLF izinli bir blokzincir platformu olduğundan, diğer blokzincir platformlarından farklı olarak ağın güvenliği, gizliliği ve işleyişini sağlamak için tüm katılımcılarının belirli kimlikleri ve rolleri bulunmaktadır. Son kullanıcı olarak da tanımlanan ve istemci uygulamaları üzerinden blokzincir ağıyla etkileşim kuran, işlem verilerini ve işlem teklifini akıllı sözleşmelere gönderen ve sorgulamalar yapan kişi veya sistemlerin geçerli bir sertifika ve kimliğe sahip olması zorunludur. Her kuruluş kendi kullanıcılarının kimlik tanımlamalarını ve sertifika geçerlilik sürelerini bağlantılı olduğu kendi sertifika yöneticisi (CA_ORGn) kurumlar aracılığıyla sağlamaktadır.

Tez kapsamında tanımlanan her kuruluş, istemci uygulamalarını kullanacak her personeline sertifika ve genel ve özel anahtardan oluşan bir dijital imza tanımlaması gerekmektedir. Bu tanımlamaya cüzdan adı verilir. Uygulamada kullanılan örnek bir sertifika ve dijital imza dosya içeriği Tablo 3.2'de verilmiştir.

Tablo 3-2: Kullanıcı için tanımlanan örnek sertifika ve özel anahtar

{"credentials":{"certificate":"-----BEGIN CERTIFICATE-----
\nMIICljCCAj2gAwIBAgIUZQOw/h0ugLitY4yM8zXjO+4+FlwwCgYIKoZIzj
0EAwIw\naDELMAkGA1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENh
cm9saW5hMRQwEgYDVQQK\nEwtIeXBlcmxlZGdlcjEPMA0GA1UECxMG
RmFicmljMRkwFwYDVQQDExBmYWJyaWMt\nY2Etc2VydmVyMB4XDTI
1MDUyMjAzMTAwMFoXDTI2MDUyMjAzMTUwMFowXTELMAkG\nA1
UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDV
QQKEwtIeXBl\ncmxlZGdlcjEPMA0GA1UECxMGY2xpZW50MQ4wDAYD
VQQDEwV1c2VyMTBZMBMGByqG\nSM49AgEGCCqGSM49AwEHA0IA
BNNMJEGPLwEa5tHe8EX5IDAp7CBBeq9t/V516hKz\nrjRp8O659jWRmNEl
NqPCx4WBNVA+9IANisP7hKsnq7JuC+6jgc8wgcwwDgYDVR0P\nAQH/BA
QDAgeAMAwGA1UdEwEB/wQCMAAwHQYDVR0OBBYEFIu0o1UAD1jB
i81qvoP+\nZ3+Q6vMKMB8GA1UdIwQYMBaAFBdnNyklY1rnx/Xar7lJwWb
58HpGMBIGA1UdEQQL\nMAmCB2F1cy1sYWIwWAYIKgMEBQYHCAEE
THsiYXR0cnMiOnsiaGYuQWZmaWxpYXRp\nb24iOiIiLCJoZi5FbnJvbGxtZ
W50SUQiOiJ1c2VyMSIsImhmLlR5cGUiOiJjbGll\nbnQifX0wCgYIKoZIzj0E
AwIDRwAwRAIgDB22r7Y3++WaLDJLKiwXuV1LQXpCN4+M\n3hof6cO+
qrUYIFXr3whVxONj42eb4sSFMpNjFgoxu1eumk1qiCmNK4eC\n-----END
CERTIFICATE-----\n","privateKey":"-----BEGIN PRIVATE KEY-----
\nMIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQg
RPJX8PunleJcXPf+\notwyF9llAWtJEn25OOXiauOfvkGhRANCAATTTCRBj
y8BGubR3vBF+SAwKewgQXqv\nbf1edeoSs640afDuufY1kZjRJTajwseFgTV
QPvSADYrD+4SrJ6uybgvu\n-----END PRIVATE KEY-----
\n"},"mspId":"Org1MSP","type":"X.509","version":1}

Kuruluşlara ait görevleri ve işlemleri yerine getirmekten sorumlu personel, istemci web uygulamasına girişte, kullanıcı adı, şifresi ve sahip olduğu cüzdan verilerini kullanır. Bu veriler zincir dışı güvenli bir veritabanında tutulabilir. Zincir dışı (off-chain) gerçekleştirilmesi gereken kullanıcı ve kimlik yönetimi tez kapsamı dışında olduğundan bu konuda detaya girilmemiştir.

İstemci uygulamalarını kimlerin kullanacağı kuruluşların görev ve hizmetlerine göre değişmektedir. Her kuruluşun kendi personeli bir kullanıcı olarak tanımlandığı gibi bazı kuruluşların kullanıcıları personel haricinde sadece veri sağlama ve sorgulama hizmetlerini

kullanma yetkisi verilerek sınırlandırılabilecektir. Örneğin, araç sahipleri "Org1:Emniyet Genel Müdürlüğü Trafik Başkanlığı" kuruluşu tarafından verilen kimlikle ve ilgili web arayüzünü kullanarak kendi araçlarıyla ilgili sorgulama ve izleme yapabilmektedir. Bakım-onarım ve tamir hizmetlerini yürüten çok sayıdaki özel şirket sorumlu personeli, "Org4:Oto Servis" kuruluşu kapsamında kullanıcı yetkisi alarak istemci uygulamasına giriş yapabilmekte ve sadece araç sahibi tarafından izin verilen araçlara ait bakım, onarım ve tamir bilgi girişi yapabilmektedir.

3.2.1.3 Kanal ve Onay Politikası:

Tez kapsamında önerilen modelde tüm kuruluşların tek kanalda birbirleriyle iletişim kurmasının yeterli olduğu kararlaştırılmıştır. Şekil 3.4'de görüldüğü gibi, altı organizasyon ve eş düğümleri ile birlikte tek sıralayıcı düğüm tek kanal üzerinde birbirleriyle etkileşim kurabilecek şekilde yapılandırılmıştır. Birden fazla kanal kullanımı, her kanalda farklı kuruluşların kendi aralarında özel bir iletişim halinde olmasını gerektirecek bir veri paylaşımı durumunda tercih edilmektedir. Önerilen modelde araçlarla ilgili her türlü işlem verisi tüm kuruluşlarla paylaşılmakta ve şeffaflık oluşturulmaktadır. İşlem kayıtlarında, araç sahiplerinin ve kullanıcıların cüzdan ve sertifika bilgileri kullanılarak anonim hale getirilmesiyle, tek kanal kullanımında daha fazla gizliliğin sağlanarak paylaşımda bulunulması da mümkündür. Tek kanal kullanımı mimari yapının daha sade ve kolay yönetilebilir olmasını da sağlamaktadır. Ayrıca kanal üzerinde tanımlanan onay ve erişim politikalarıyla birlikte akıllı sözleşme kodları ve istemci uygulaması üzerinde yapılan sınırlamalarla, kullanıcıların sadece ihtiyaç duydukları verileri görmelerini de sağlayarak gizliliğin korunmasına katkı sağlanmaktadır.

Bir akıllı sözleşme zincir kodu tanımı yapılırken, onay politikası da tanımlanır. Bu politika, mevcut kanalda istemci uygulaması üzerinden teklif edilen işlemlerin onaylanması ve kayıt defterine yazılabilmesi için hangi kuruluşların onayının alınması gerektiğini bildirir. Sonraki bölümlerde, önerilen modelin simülasyon uygulaması için yapılandırma parametreleri içerisinde kullanılan onay politikası ve tanımı verilmiştir.

3.2.1.4 Durum Veritabanı

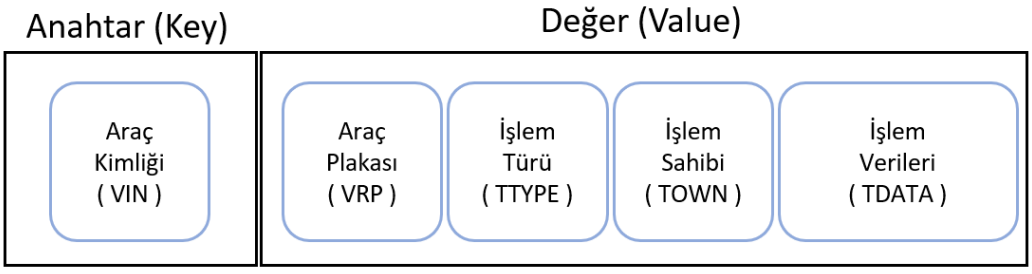
HLF blokzincir mimarisinde kayıt defterine ait özdeş mevcut durum bilgileri durum veritabanında saklanmaktadır. Bu akıllı sözleşme zincir kodu ile verimli okuma ve sorgulamalar için kolaylık sağlamaktadır. HLF ağlarında LevelDB ve CouchDB veritabanları

desteklenmektedir. Önerilen model de CouchDB kullanımı seçilmiştir. CouchDB, JSON veri yapısında daha detaylı sorgulamalar yapılabildiği için tercih edilmiştir. CouchDB kullanımı LevelDB ye göre sistem performansını düşürücü bir etkiye sahiptir. Ancak birçok kuruluşun daha detaylı sorgulama gereksinimi olduğu için kullanımı daha önemli olmaktadır.

3.2.1.5 İşlemler ve Veri Yapısı

Tez çalışmamızın hedeflerine ulaşmada en önemli kısım, akıllı sözleşmelerle verilerin kayıt defterine saklanması ve sorgulamalarla kayıt defterindeki verilerin okunmasının organize edilmesidir. Kuruluşların görev ve sorumlulukları kapsamında gerçekleştirilen işlemlerin çeşitliliği her ne kadar çok olsa da temel mantık ve kurallar benzerdir. Yetkili ve görevli personelin kullanıcı olarak istemci uygulamaları üzerinden girdikleri verilere göre işlemler öncelikli olarak sınıflandırılmaktadır. Bu sınıflandırmaya ve işlem çeşitliliğine göre isimlendirilmiş ve programlanmış zincir kodları tek bir akıllı sözleşme program dosyasında birleştirilmektedir. Akıllı sözleşme zincir kodlarının programlanmasında Node.js ve JavaScript programlama dili kullanılmıştır. Tasarlanan akıllı sözleşme programlarına ait algoritmalar ve işlevleri sonraki bölümde verilmiştir.

Veri Yapısı: Varlık olarak tanımlanan araçların üzerinde gerçekleştirilen işlem ve veri çeşitliliğinin çokluğu nedeniyle verilerin kayıt defterinde depolanması ve sorgulamaların kolayca yapılması ve anahtar-değer (key-value) çifti şeklinde verilerin depolanması için, anahtar alanına araç kimliği (VIN) yazılırken, değer alanına yazılacak verilerin sınıflandırılması yöntemi tercih edilmiştir. Buna göre aşağıdaki Şekil 3.3'de verilen bir veri yapısı oluşturulmuştur.



Şekil 3-3: Önerilen modelde işlem veri yapısı

Araç Kimliği (VIN): Blokzincir ağında varlık olarak tanımlanan araçların benzersiz dijital kimliği, araç üreticileri tarafından verilen motor ve şasi numaraları ile ilk tescil tarihlerinin birleştirilerek, kriptografik özet fonksiyonuyla (SHA256) elde edilen karması ile oluşturulmaktadır. Şekil 3.4'de araç kimliğinin nasıl tanımlandığı verilmiştir.

Şekil 3-4: Araç kimlik tanımı

Araç Plakası (VRP, Vehicle Registration Plate): Araç plakası, araçların yasal tescilini ve trafikteki kimliğini belirleyen anonim bir değer olarak araçların ve araç sahiplerinin kimliklerine ulaşmada kullanılsa da geçici bir değerdir. Çünkü araç plakaları gerektiğinde yada istenildiği zaman zaman değiştirilebilir. Ancak günümüzde yaygın olarak trafikte araçların tanımlanmasında, denetim ve takip işlemlerinde, araç sahibine ulaşmada kullanılmaktadır.

İşlem Türü (TTYPE, Transaction Type): İşlem türü, tez çalışmamızda işlem ve veri çeşitliğini tanımlamak ve sınıflandırmak amacıyla kullandığımız bir değerdir. Farklı kuruluşlar tarafından gerçekleştirilen işlemleri ve içeriğinde bulunan veri çeşitliğini belirlemekle birlikte sorgulamalarda filtreleme yapılarak verilere daha kolay ulaşılmasını sağlamak amaçlanmıştır.

İşlem Sahibi (TOWN, Transaction Owner): İstemci uygulaması üzerinden giriş yapan personel yada kullanıcının sistemdeki kimlik bilgisini ifade eder. Kayıt altına alınan her işlemin hangi kullanıcı tarafından oluşturulduğunun kanıtını oluşturur.

İşlem Verileri (TDATA, Transaction Data): İşlem çeşidine göre, kayıt altına alınması gereken bilgilerin JSON veri yapısındaki veri dizisidir.

3.2.1.6 Akıllı Sözleşme Kodları

Akıllı sözleşmeler, Node.js ve JavaScript programlama dili ile yazılmıştır. HLF 'de akıllı sözleşmelere zincir kodu adı verilmekte ve bir alt program olarak tanımlanmaktadır. Kanalda bulunan her düğüm, tanımlanan onay politikalarına göre onaylayıcı düğüm yetki ve izni alabileceği için aynı zincir kodlarına sahiptir. Dolayısıyla tüm zincir kodları tek bir dosyada

birleştirilerek kanaldaki tüm düğümlere dağıtılmıştır. Akıllı sözleşme zincir kodlarında kullanılan veri alanları ve açıklamaları Tablo 3.3'de verilmiştir.

Tablo 3-3: Akıllı sözleşme veri alanları ve açıklamaları

Veri Alanı	Açıklama
KEY	Kayıt Defteri (Ledger) ve Durum Veri Tabanında, Anahtar-Değer ikilisi olarak tutulan işlem kayıtlarında anahtar değeri ifade eder, Uygulamada, Araç Kimliği (VIN) anahtar değerdir. 64 karakter SHA256 özet (hash) değerinden oluşur.
VIN (Vehicle Identification Number)	İstemci uygulamalarından gönderilen ve üzerinde işlem yapılan Araç Kimliği (VIN) değeridir ve aynı zamanda anahtar değer olarak kullanılır. 64 karakter SHA256 özet (hash) değerinden oluşur.
VRP (Vehicle Registration Plate)	İstemci uygulamalarından gönderilen Araç Plaka değeridir.
TTYPE (Transaction Type)	İstemci uygulamasında yapılan işlem çeşidine göre uygulama içesinde tanımlanan işlem tipi değerdir. (Örnek: T01.001: İlk Tescil Kaydı, T02.001: Mülkiyet değişimi, S01.001: Sigorta Policesi vb.)
TOWN (Transcation Owner)	İstemci uygulaması üzerinden işlem teklifini gönderen yetkili personel kimlik tanımıdır. İşlem verilerinin kim tarafından girildiğinin kanıtını sağlar.
TDATA (Transaction Data)	İstemci uygulamalarında girilen ve yapılan işleme ait detaylı bilgileri içerisinde barındıran JSON veri yapısında bir dizi değerdir. Akıllı sözleşme kodları aynı yapıda tasarlanmış olup değişen kayıt verileri bu alanda tanımlanmaktadır.

Araçlarla ilgili çeşitli işlemlerin kayıt defterine yazılabilmesi için, önce araç kimliği ve araca ait tüm detay bilgilerin kayıt defterine yazılmış olması gereklidir. Kayıt defterinde kimliği olmayan bir araç için farklı türlerde işlem kayıtları yapılamaz. Bu nedenle, aracın ilk tescil kaydının Algoritma-1'de verilen sözde kod algoritmasına göre hazırlanmış zincir kodu ile yapılması gerekmektedir. Aracın ilk tescil kaydı Org1:EGM Trafik Başkanlığı ve Org6: Türkiye Noterler Birliği kuruluşlarının yetkili personeli tarafından yapılabilir.

Algoritma-1'de 2-6 arası satırlarda verilen giriş verileri ve tipleri diğer zincir kodlarında aynı amaçla kullanılmış olup, JSON veri yapısındaki "islemData" içeriği yapılan işlem tipine göre değişmektedir. Algoritma-2'de Sigorta Poliçe bilgilerinin işlem kaydını göstermektedir.

Algoritma 1. İlk Tescil Kaydı

```
 1: Giriş:
 2:     aracKimlik – Araç Kimliği,
 3:     aracPlaka – Araç Plakası,
 4:     islemTipi – İşlem Türü
 5:     islemSahip – İşlem Sahibi,
 6:     JSON(islemData) – JSON Veri Yapısında Araç Bilgileri [motorNo, saseNo,
Marka, Tipi, ticariAdi, modelYili, aracSinifi, Cinsi, Rengi, netAgirlik, AYAgirlik,
RYAgirlik, koltukSay, yolcukSay, silindirHacmi, motorGucu, yakitCinsi, GucOran,
kullanimAmaci, aracDurum …]

 7: Çıkış:
 8:     Return:  Onay / Hata
 9: FUNCTION ilkTescilKayit
10. IF (Ara_Ledger(VIN) = True)
11. THEN
12.       Return "Hata";
13. ELSE
14.     KEY ← aracKimlik;
15.     VRP ← aracPlaka;
16.     TTYPE ← islemTipi;
17.     TOWN ← islemSahip;
18.     Array(TDATA) ← JSON(islemData);
19.     RDATE ← date();
20.     IF ( kayit_Ledger(KEY, VRP, TTYPE, TOWN,TDATA,RDATE) = True)
21.           THEN Return "Onay";
22.     ELSE
23.          Return "Hata ";
24.     END IF
25. END IF
```

Algoritma 2. Sigorta Poliçe İşlem Kaydı

```
1: Giriş:
2:      aracKimlik – Araç Kimliği,
3:      aracPlaka – Araç Plakası,
4:      islemTipi – İşlem Türü
5:      islemSahip – İşlem Sahibi,
6:      JSON(islemData) – JSON Veri Yapısında Araç Bilgileri [sirketKodu, basTarih,
bitTarih, tramerNo, belgeNo, marka, model, tescilTarihi, TGrub, motorNo, sasiNo,
ABMaddi, KBMaddi, ABTedavi, KBTedavi…]

7: Çıkış:
8:      Return:  Onay / Hata
9: FUNCTION sigortaPoliceKayit
10. IF (ara_Ledger(VIN) = True)
11. THEN
12.       Return "Hata";
13. ELSE
14.     KEY ← aracKimlik;
15.     VRP ← aracPlaka;
16.     TTYPE ← islemTipi;
17.     TOWN ← islemSahip;
18.     Array(TDATA) ← JSON(islemData);
19.     RDATE ← date();
20.     IF ( kayit_Ledger(KEY, VRP, TTYPE, TOWN,TDATA,RDATE) = True)
21.             THEN Return "Onay";
22.     ELSE
23.           Return "Hata ";
24.     END IF
25. END IF
```

4. BAKSİS SİMÜLASYON UGULAMASI VE ÇALIŞMA PRENSİBİ

4.1 BLOKZİNZİR AĞ KURULUMU

Önerilen blokzincir modelinin simülasyon uygulamasını gerçekleştirmek için, Bandırma Onyedi Eylül Üniversitesi Akıllı Ulaşım Sistemleri ve Teknolojileri bilgisayar laboratuvarında bulunan sunucu özellikli bir bilgisayar kullanılmıştır. Sunucu ile uzaktan bağlantı kurularak simülasyon uygulaması geliştirilmiştir. Önerilen modelde belirtilen bileşenlerin yapılandırılması ve blokzincir ağ simülasyonu için HLF test ağına ait Docker konteyner uygulamaları ile Fabric ikili betik programları kullanılmıştır. Simülasyon uygulamasının kurulumu ve çalıştırılması için gerçekleştirilen işlemler adım adım sonraki bölümlerde detaylı bir şekilde verilmektedir.

4.1.1 Kullanılan Donanım ve Yazılımlar

Önerilen modelinin simülasyon uygulaması için Tablo 4.1'de özellikleri ve kullanım amaçları verilen donanım ve yazılımlar kullanılmıştır.

Tablo 4-1: Simülasyon uygulamasında kullanılan yazılım ve donanımlar

Donanım / Yazılım	Özellik / Versiyon	Kullanım Amacı
Sunucu Donanımı :	Intel Xeon 8x-CPU, 4,6GHz, 32 GB RAM, 2 TB Sabit Disk	Önerilen modelinin simülasyon uygulaması için Hyperledger Fabric Blokzincir ağının kurulumu ve çalıştırılması
İşletim Sistemi :	Ubuntu 22.04.6 LTS	Blokzincir Ağı ve İstemci / Kullanıcı Arayüz uygulamalarının çalıştırıldığı işletim sistemi
Hyperledger Fabric :	2.5.9	Hyperledger çekirdek yazılımları ve konteyner yapılandırmaları
Hyperledger Fabric-CA :	1.5.6	Hyperledger Fabric sertifika ve kriptografik dijital imza sağlayıcısı
Hyperledger Caliper :	0.6.0	Hyperledger Fabric performans ölçümleri
Hyperledger Explorer :	0.6.0	Hyperledger Fabric blok ve işlem takibi
Docker :	26.1.3	Hyperledger Fabric ağının bileşenleri olan; "Eş Düğümler", "Sıralayıcı Düğüm", "Durum Veritabanları", "Sertifika Yöneticileri" için sanal makine konteyner uygulamalarının çalıştırılması.
Docker Compose :	1.29.2	Docker konteyner uygulamalarının komut satırından yönetimi
Portainer :	2.19.5	Docker konteyner uygulamalarının Web Arayüzü üzerinden yönetimi

4.1.2 Blokzincir ağının konfigürasyonu

Tez kapsamında önerilen modelde 6 kuruluş (Org) ve her kuruluşa ait 3'er adet eş düğüm (peers) yapılandırması için gerekli olan parametreler Tablo 4.2'de gösterildiği gibi belirlenmiştir. peer0.Org1 isimli eş düğüme ait konteyner yapılandırma dosyası (docker_files.yaml) içeriği Şekil 4.1'de gösterildiği gibi tanımlanmıştır. Tüm eş düğüm ve sıralayıcı (orderer) düğümlerde Şekil 4.1'de ki gibi konfigürasyon tanımları, Tablo 4.2' de verilen parametre değerlerine göre ayrı ayrı yapılmıştır.

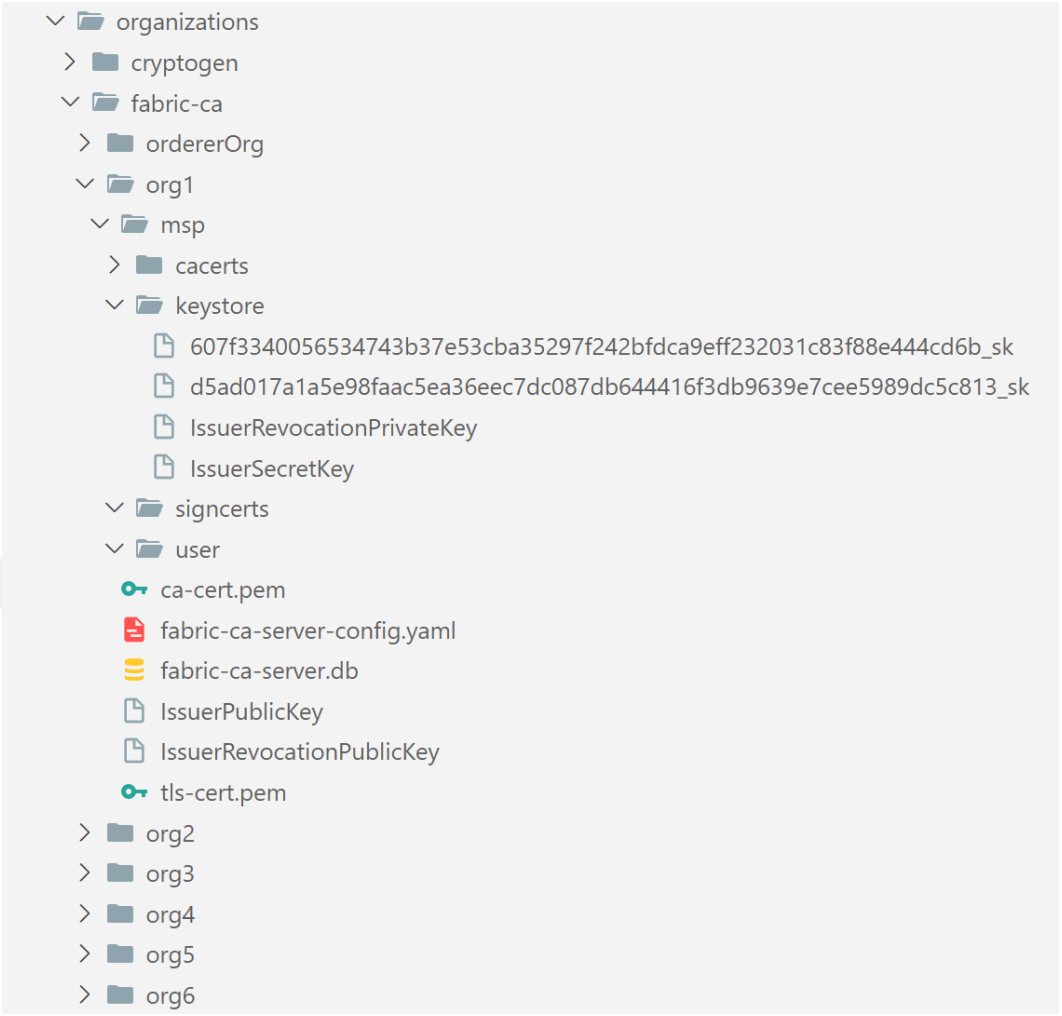
Tablo 4-2: Eş düğümler için planlanan konfigürasyon parametre değerleri

Org. No	Peer Address	Peer Address Port	ChainCode Address Port	Peer Local MSPID	CA Name	CA Port	Database (DB) Name	DB Port	Docker Container Name
ORG 1	peer0.org1.EGM_Trafik.gov	7101	7102	Org1MSP	ca_org1	7001	peer0_org1_DB	9010	peer0.org1
	peer1.org1.EGM_Trafik.gov	7111	7112				peer1_org1_DB	9011	peer1.org1
	peer2.org1.EGM_Trafik.gov	7121	7122				peer2_org1_DB	9012	peer2.org1
ORG 2	peer0.org2.Otomotiv.com	7201	7202	Org2MSP	ca_org2	7002	peer0_org2_DB	9020	peer0.org2
	peer1.org2.Otomotiv.com	7211	7212				peer1_org2_DB	9021	peer1.org2
	peer2.org2.Otomotiv.com	7221	7222				peer2_org2_DB	9022	peer2.org2
ORG 3	peer0.org3.TuvTurk.com	7301	7302	Org3MSP	ca_org3	7003	peer0_org3_DB	9030	peer0.org3
	peer1.org3.TuvTurk.com	7311	7312				peer1_org3_DB	9031	peer1.org3
	peer2.org3.TuvTurk.com	7321	7322				peer2_org3_DB	9032	peer2.org3
ORG 4	peer0.org4.Servis.com	7401	7402	Org4MSP	ca_org4	7004	peer0_org4_DB	9040	peer0.org4
	peer1.org4.Servis.com	7411	7412				peer1_org4_DB	9041	peer1.org4
	peer2.org4.Servis.com	7421	7422				peer2_org4_DB	9042	peer2.org4
ORG 5	peer0.org5.Sigorta.com	7501	7502	Org5MSP	ca_org5	7005	peer0_org5_DB	9050	peer0.org5
	peer1.org5.Sigorta.com	7511	7512				peer1_org5_DB	9051	peer1.org5
	peer2.org5.Sigorta.com	7521	7522				peer2_org5_DB	9052	peer2.org5
ORG 6	peer0.org6.Noter.com	7601	7602	Org6MSP	ca_org6	7006	peer0_org6_DB	9060	peer0.org6
	peer1.org6.Noter.com	7611	7612				peer1_org6_DB	9061	peer1.org6
	peer2.org6.Noter.com	7621	7622				peer2_org6_DB	9062	peer2.org6

```
#===============  ORG1 PEER0 ==================
  peer0.org1.example.com:
    image: hyperledger/fabric-peer:2.5.9
    labels:
      service: hyperledger-fabric
    environment:
      - FABRIC_CFG_PATH=/etc/hyperledger/peercfg
      - FABRIC_LOGGING_SPEC=INFO
      - CORE_PEER_TLS_ENABLED=true
      - CORE_PEER_PROFILE_ENABLED=false
      - CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt
      - CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key
      - CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt
      - CORE_PEER_ID=peer0.org1.example.com
      - CORE_PEER_ADDRESS=peer0.org1.example.com:7101
      - CORE_PEER_LISTENADDRESS=0.0.0.0:7101
      - CORE_PEER_CHAINCODEADDRESS=peer0.org1.example.com:7102
      - CORE_PEER_CHAINCODELISTENADDRESS=0.0.0.0:7102
      - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org1.example.com:7101
      - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org1.example.com:7101
      - CORE_PEER_LOCALMSPID=Org1MSP
      - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/fabric/msp
      - CORE_OPERATIONS_LISTENADDRESS=peer0.org1.example.com:9910
      - CORE_METRICS_PROVIDER=prometheus
      - CHAINCODE_AS_A_SERVICE_BUILDER_CONFIG={"peername":"peer0org1"}
      - CORE_CHAINCODE_EXECUTETIMEOUT=300s
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}_test
    volumes:
      - /var/run/:/host/var/run/
      -
../organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/
msp:/etc/hyperledger/fabric/msp
      -
../organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/
tls:/etc/hyperledger/fabric/tls
      - peer0.org1.example.com:/var/hyperledger/production
      - ./peercfg:/etc/hyperledger/peercfg
      - ${DOCKER_SOCK}:/host/var/run/docker.sock
    working_dir: /root
    command: peer node start
    ports:
      - 7101:7101
      - 9910:9910
    networks:
      - baksis_network
```

Şekil 4-1: "docker_files.yaml" dosyasında, peer0.org1 için tanımlanan konfigürasyon

HLF eş düğüm yapılandırmalarında eş düğümlerin çevresel değişkenlerine ait değerlerin neler olduğu tanımlanmaktadır. Bu değerlerin içerisinde eş düğümlerin sertifika ve dijital imzalarının olduğu dosyaların hangi klasörlerde olduğu da belirtilmektedir. Her eş düğüm ve kullanıcılarının önceden dijital sertifika ve dijital imza bilgilerini içeren klasör ve dosyaların oluşturulması gereklidir. Sonraki bölümde ağın kurulum aşamasında sertifika dosyalarının oluşturulmasında kullanılan yöntem ve komutlar anlatılmaktadır. Şekil 4.2'de eş düğümlere ait kriptografik sertifika dosyalarının bulunduğu dizin yapısı verilmiştir.



Şekil 4-2: Organizasyon ve eş düğümlerin kriptografik sertifika dosyaları dizin yapısı

HLF, esnek ve modüler bir yapıya sahip olduğundan, ağ bileşenlerinin (peer'lar, orderer'lar, CA'lar, istemci uygulamaları) her birinin davranışını ve birbirleriyle nasıl etkileşim kurduğunu belirleyen çeşitli yapılandırma dosyaları bulunur. Bu dosyalar genellikle YAML (.yaml) formatındadır ve ortam değişkenleri ile değerleri sonradan değiştirilebilir.

configtx.yaml yapılandırma dosyası: Bu dosya ile Fabric ağının ve kanallarının ileri düzey yapılandırma ayarlarını belirlemek için kullanılır. Bu dosya, ağın ilk kurulumu ve başlatılması esnasında **configtxgen** aracı uygulaması tarafından okunarak ağın ilk bloğunun (genesis) ve kanal yapısının oluşturulmasında kullanılır. Ağın başlatılması ve yapılandırma araçlarının kullanımı ağ yöneticisi (admin) rolündeki kullanıcı tarafından yapılabilir.

Önerilen çalışmada Org1 kuruluşunun admin kullanıcısına ait sertifika ve kriptografik dosyalar kullanılarak ağın yapılandırılması gerçekleştirilmiştir.

configtx.yaml dosyası üzerinde yapılan yapılandırma ayarları şunlardır;

- **Organizasyon Tanımları (*Organizations*):** Her kuruluşun ve Sıralayıcı düğümün adı, MSP ID 'si ve kriptografik materyallerin dizin yolu ile kuruluş ve eş düğümlerin rol tanımları. (Şekil 4.3)
- **Yetkilendirme Politikaları (Policies):** Ağ içerisinde yeni kuruluş ekleme, sistem kanalı güncelleme ile kanal içinde üyelik değişimi ve zincir kodu güncellenmesinde hangi kullanıcıların hangi işlemleri yapabileceği ve onayının alınabileceğini belirleyen (ImplicitMeta, Signature) yetki politikalarının tanımlanması, (Şekil 4.3)
- **Sıralama Hizmeti Yapılandırma (Orederer):** Sıralama hizmeti için kullanılacak mutabakat algoritma türü (Kafka, Raft yada PBFT vb. gibi) blok boyutu (işlem sayısı ve byte cinsinden boyut), blok zaman aşımı ve sıralama hizmetindeki düğümlerin isimleri. (Şekil 4.4)
    - **Desteklenen Uygulama Yetenekleri (Application Capabilities):** Ağ ve kanalın desteklediği Fabric uygulama versiyonları ile eski ve yeni versiyonlarda hangi yeteneklerin destekleneceği. (Şekil 4.5)

```
Organizations:
    - &OrdererOrg
        Name: OrdererOrg
        ID: OrdererMSP
        MSPDir: ../organizations/ordererOrganizations/example.com/msp
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Writers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Admins:
                Type: Signature
                Rule: "OR('OrdererMSP.admin')"
        OrdererEndpoints:
            - orderer.example.com:7050
    - &Org1
        Name: Org1MSP
        ID: Org1MSP
        MSPDir: ../organizations/peerOrganizations/org1.example.com/msp
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('Org1MSP.admin', 'Org1MSP.peer', 'Org1MSP.client')"
            Writers:
                Type: Signature
                Rule: "OR('Org1MSP.admin', 'Org1MSP.client')"
            Admins:
                Type: Signature
                Rule: "OR('Org1MSP.admin')"
            Endorsement:
                Type: Signature
                Rule: "OR('Org1MSP.peer')"
        AnchorPeers:
            - Host: peer0.org1.example.com
              Port: 7101
```

Şekil 4-3: Organizasyon ve onay politikaları için parametre ve değer tanımları

```
Orderer: &OrdererDefaults
    OrdererType: etcdraft
    Addresses:
        - orderer.example.com:7050
    EtcdRaft:
        Consenters:
        - Host: orderer.example.com
          Port: 7050
          ClientTLSCert:
../organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt
          ServerTLSCert:
../organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt
    BatchTimeout: 2s
    BatchSize:
        MaxMessageCount: 50
        AbsoluteMaxBytes: 99 MB
        PreferredMaxBytes: 10 MB
```

Şekil 4-4: Sıralama Hizmeti (Orderer) düğümü için parametre ve değer tanımları

```
Capabilities:
    Channel: &ChannelCapabilities
        V2_0: true
    Orderer: &OrdererCapabilities
        V2_0: true
    Application: &ApplicationCapabilities
        V2_5: true
```

Şekil 4-5: Desteklenen uygulama yetenekleri için parametre ve değer tanımları

core.yaml dosyası üzerinde yapılan yapılandırma ayarları şunlardır;

- **Eş Düğüm Kimlikleri ve Ağ Ayarları (peer):** peer0.org1.example.com gibi isimlendirme ve tanımlar, eş düğümler arası iletişimde ağ dinleme port numaraları, gossip protokol ayarları, eş düğümün üye olduğu MSP ID ayarları, sertifika materyalleri erişim yolu gibi ayarlamalar,
- **TLS Ayarları (peer.tls):** TLS güvenlik protokolünün etkin olup olmadığı, TLS sertifika materyallerine erişim yolları,
- **Kayıt Defteri (Ledger) Ayarları (ledger):** Kayıt defteri ve durum veritabanı verilerinin depolandığı dizin, kullanılacak Dünya Durumu (LevelDB, CouchDB) türü ve bağlantı ayarları,
- **Zincir Kodu (ChainCode) Ayarları:** Zincir kodu yürütme zamanı ile zincir kodu başlatma zaman aşım süre ayarları
- **Metrik ve İzleme Ayarları (metrics):** Prometheus ve StatsD gibi izleme araçları için ayarlamalar

orderer.yaml dosyası üzerinde yapılan yapılandırma ayarları şunlardır;

- Genel Ayarlar (General): Sıralayıcı düğümle bağlantı için adres ve dinleme port numaraları, TLS güvenlik protokolünün etkin olup olmadığı, Sıralayıcı düğüme ait MSP ayarları, başlangıç blok (Genesis) bağlantı yolu
- Mutabakat Ayarları (Consensus): Kullanılacak mutabakat algoritması türü (Kafka, Raft vb.) ve özel ayarları

4.1.3 Ağın Kurulum Aşamaları

HLF test ağının kurulması ve çalıştırılması için belirli işlem adımlarının sırayla takip edilmesi gerekmektedir. Kurulumun yapılacağı bilgisayarda öncelikle, Git, Docker ve Docker Compose, Node.js ve NPM linux paketlerinin Ubuntu Linux işletim sistemi üzerinde kurulu ve çalışır halde olması gereklidir [68]. Daha sonra Hyperledger Fabric v2.5 ağ kurulum aşamalarına geçilebilir ;

1.Adım: Fabric ikili dosyaları ile Docker imajlarının indirilmesi

Fabric CLI araçlarının ve konfigürasyon dosyalarının indirilmesi için Fabric 'in github deposundan aşağıdaki komut ile Fabric ikili dosya (configtxgen, configtxlator, cryptogen, discover, idemixgen, orderer, osnadmin, peer, fabric-ca-client, fabric-ca-server) ve yapılandırma araçları indirilir [68].

```
Ubuntu> curl -sSLO
https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh
&& chmod +x install-fabric.sh
Ubuntu> ./install-fabric.sh --fabric-version 2.5 binary docker
```

2. Adım: Sertifika ve kripto materyallerinin oluşturulması

HLF ağında tüm katılımcı ve kullanıcıların kimlik doğrulaması için ortak anahtar altyapısı (PKI) kullanır. Her düğüm, yönetici ve kullanıcı için geçerli bir sertifika ve özel anahtar tanımlamalarının yapılması için Fabric sertifika otoritesi (Fabric_CA) ya da Cryptogen aracı kullanılabilir [68].

```
# Cryptogen aracı
Ubuntu> cryptogen generate \
            --config=./organizations/cryptogen/crypto-config-org1.yaml \
            --output="organizations"
Ya da;
# Fabric-CA aracı
Ubuntu> docker-compose up -f docker-compose-ca.yaml
Ubuntu> fabric-ca-client register
            --caname ca-org1 \
            --id.name peer0 \
            --id.secret peer0pw \
            --id.type peer \
            --tls.certfiles ./organizations/fabric-ca/org1/tls-cert.pem
```

3. Adım: Kayıt defteri ve Başlangıç Bloğu (Genesis) oluşturulması

Defter (Ledger) blokzincir kayıtlarının tutulduğu "genesis.block" dosyasının oluşturulması için "configtxgen" aracı kullanılır.

```
Ubuntu> configtxgen -profile TwoOrgsOrdererGenesis \
                    -channelID system-channel \
                    -outputBlock ./system-genesis-block/genesis.block
```

4. Adım: Kayıt defteri ve Başlangıç Bloğu (Genesis) oluşturulması

Gerekli sertifikalar ve sertifika dizin yapıları ile kayıt defteri, başlangıç blok oluşturulduktan sonra, blokzincir ağının temel aktörleri olan eş düğüm ve sıralayıcı düğüm gerekli olan yapılandırma parametreleri ayarlandıktan sonra, Docker sanal makine üzerinde çalışmaları sağlanır. Durum veritabanı olarak CouchDB kullanılacaksa, her eş düğüm için durum veritabanları konteyner olarak kurulabilir.

```
Ubuntu> docker-compose docker-compose-peers.yaml up -d
Ubuntu> docker-compose docker-compose-couchdb.yaml up -d
```

5. Adım: Kanal oluşturulması ve eş düğümlerin kanal üyeliklerinin yapılması

Fabric de kanal oluşturulması ve kanala tüm eş düğümlerin üye yapılması için bazı çevresel değişkenlerin önceden tanımlanması ve alt adımların takip edilmesi gerekir.

A- Kanal adına göre kanal yapılandırma dosyasının oluşturulması

```
Ubuntu> export CHANNEL_NAME= aracsicilsistem
Ubuntu> configtxgen -profile SixOrgsChannel
-outputCreateChannelTx ./channel-artifacts/${CHANNEL_NAME}.tx
-channelID $CHANNEL_NAME
```

B- Kanal oluşturulması ve sıralayıcı hizmeti tanımı

```
Ubuntu> export CHANNEL_NAME=aracsicilsistem
Ubuntu> peer channel create -o localhost:7050
                          -c $CHANNEL_NAME
                          -f ./channel-artifacts/${CHANNEL_NAME}.tx
                          --tls  --cafile
${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/ms
p/tlscacerts/tlsca.example.com-cert.pem
```

C- Eş düğümlerin kanala üye yapılması

Her eş düğüm için çevresel değişkenlerin tanımlanarak kanala üye yapılması gerekir, Aşağıdaki komutta peer0.org1 için tanımlanma yapılmıştır [68].

```
Ubuntu> export CHANNEL_NAME= aracsicilsistem
Ubuntu> export PATH=${PWD}/../bin:${PATH}
Ubuntu> export FABRIC_CFG_PATH=$PWD/../config/
Ubuntu> export CORE_PEER_TLS_ENABLED=true
Ubuntu> export CORE_PEER_LOCALMSPID="Org1MSP"
Ubuntu> export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/
                      org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
Ubuntu> export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/
                      org1.example.com/users/Admin@org1.example.com/msp
Ubuntu> export CORE_PEER_ADDRESS=localhost:7051

Ubuntu> peer channel join -b ./channel-artifacts/aracsicilsistem.block
```

4.1.4 Akıllı sözleşme zincir kodunun eş düğümlere dağıtımı

Zincir kod dağıtım aşamaları [68];

```
Ubuntu> peer lifecycle chaincode package ArTes.tar.gz
                   --path ArTes --lang javascript
                   --label ArTes_1

Ubuntu> export CHANNEL_NAME= aracsicilsistem
Ubuntu> export PATH=${PWD}/../bin:${PATH}
Ubuntu> export FABRIC_CFG_PATH=$PWD/../config/
Ubuntu> export CORE_PEER_TLS_ENABLED=true
Ubuntu> export CORE_PEER_LOCALMSPID="Org1MSP"
Ubuntu> export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/
                      org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
Ubuntu> export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/
                      org1.example.com/users/Admin@org1.example.com/msp
Ubuntu> export CORE_PEER_ADDRESS=localhost:7051

Ubuntu> peer lifecycle chaincode install ArTes.tar.gz

Ubuntu> peer lifecycle chaincode queryinstalled

Ubuntu> peer lifecycle chaincode approveformyorg -o localhost:7050 \
                     --ordererTLSHostnameOverride orderer.example.com --tls \
                     --cafile msp/tlscacerts/tlsca.example.com-cert.pem \
                     --channelID aracsicilsistem --name ArTes --version 1 \
                     --package-id ArTes_1:… --sequence 1 \
```

```
Ubuntu> peer lifecycle chaincode checkcommitreadiness \
                    --channelID aracsicilsistem --name ArTes \
                    --version 1 --sequence 1 --output json

Ubuntu> peer lifecycle chaincode approveformyorg -o localhost:7050 \
                    --ordererTLSHostnameOverride orderer.example.com \
                    --tls --cafile ./msp/tlscacerts/tlsca.example.com-cert.pem \
                    --channelID aracsicilsistem --name ArTes \
                    --version 1 --package-id ArTes_1:… --sequence 1

Ubuntu> peer lifecycle chaincode commit -o localhost:7050 \
                    --ordererTLSHostnameOverride orderer.example.com --tls \
                    --cafile ./msp/tlscacerts/tlsca.example.com-cert.pem \
                    --channelID kanalsigorta --name ArTes \
                    --peerAddresses localhost:7051 \
                    --tlsRootCertFiles ./peer0.org1.example.com/tls/ca.crt \
                    --peerAddresses localhost:9051 \
                    --tlsRootCertFiles ./peer0.org2.example.com/tls/ca.crt \
                    --version 1 --sequence 1

Ubuntu> peer lifecycle chaincode querycommitted \
                    --channelID aracsicilsistem --name ArTes
```

4.2 İSTEMCİ UGULAMASI VE KULLANICI WEB ARAYÜZLERİ

HLF istemci uygulaması son kullanıcı ve sistemlerin Fabric ağı ile etkileşim kurmasını sağlayan yazılımlardır. Bu uygulamalar ile ağda tanımlı akıllı sözleşmeler çağrılarak kayıt defterine veri girişi yapmak ve defterde bulunana bilgileri sorgulamak ve okumak gibi işlevler yerine getirilir.

Tez çalışmamızda önerilen modelde 6 adet kuruluş araçlarla ilgili işlemleri gerçekleştirmek, HLF ağında paylaşılan bilgilere yetki ve izinlerine göre erişmek için kendi istemci uygulamalarına sahiptir. Her kuruluş kendi merkezi veri yönetim uygulamaları ile istemci uygulamaları arasındaki bağlantıyı ve entegrasyonu Fabric SDK'ları üzerinden organize edebilir. Bu entegrasyon tez kapsamında ele alınmadığı için her kuruluşa ait web arayüzleri programlanarak yapılacak işlemlere ait örnek web sayfaları ve veri giriş form alanları tasarlanmıştır.

Her kuruluş kendi sorumlulukları çerçevesinde araçlar üzerinde birçok işlem yürütmektedir. Tez çalışmamızda araçların yaşam döngüsü içerisinde gerçekleştirilen işlemlerin önerilen blokzincir ağ modeliyle kayıt altına alınması ve yönetilmesi hedeflenmektedir. Bu hedef doğrultusunda yapılan işlemlerin çok çeşitli ve fazla olduğu gerçeği karşısında, benzer yöntem ve özelliklere sahip her işlemin anlatılması yerine seçilen işlemlerin ve senaryoların nasıl yapıldığı sonraki bölümlerde anlatılmıştır.

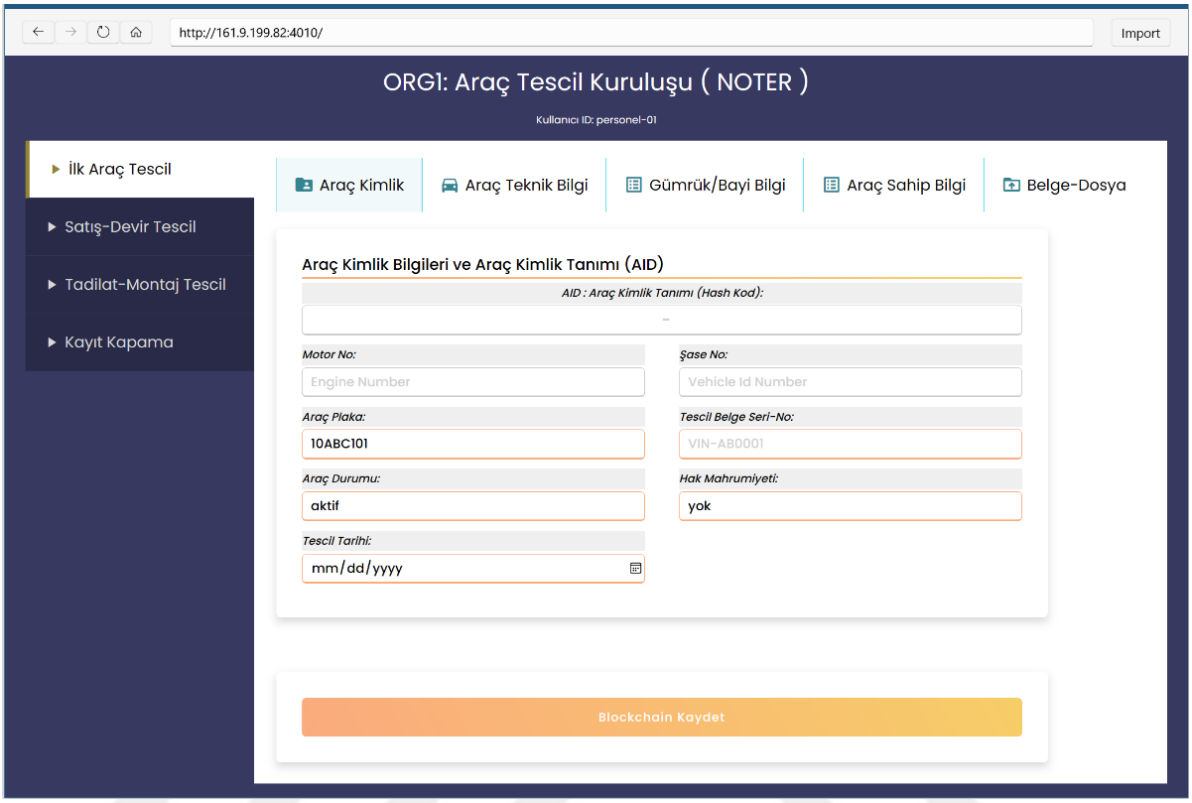
4.2.1 Araç Tescil İşlemleri

Ülkemizde resmi kurumlara ait olmayan araçların tescil işlemleri Emniyet Genel Müdürlüğü Trafik Başkanlığı ile Türkiye Noterler Birliği tarafından yapılmaktadır. Araçların ilk tescil işlemleri ve ikinci el veya mülkiyet değişimi tescil işlemleri arasında bazı farklılar vardır. Bunun yanında plaka değişikliği, araç üzerinde tadilat ve dönüşümler yapıldığında uygulanan tescil işlemleri de bulunmaktadır.

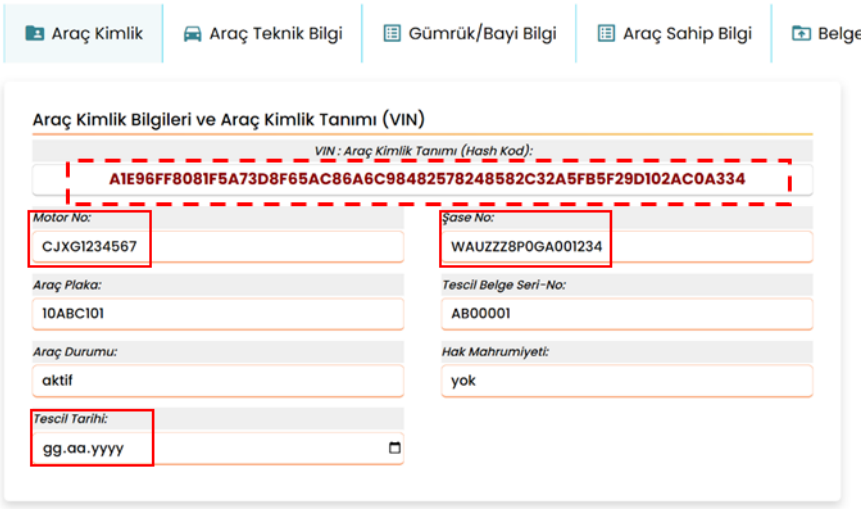
Aracın İlk Tescili ve Araç Kimlik Numarasının Belirlenmesi:

Uygulamamızda araçlarla ilgili tescil işlemlerinin yapılması için Şekil 4.6' da verilen web arayüzü kullanılmaktadır. İlk tescil işlemleri aracın kimlik (VIN) bilgisinin üretilmesi için gereklidir. Sonraki aşamalarda araçla ilgili tüm işlemlerde VIN değeri referans/anahtar değer olarak kullanılacaktır. Araç kimlik bilgisinin elde edilmesinde "Motor Numarası", "Şase

Numarası" ve "Tescil Tarihi" değerleri birleştirilerek SHA256 kriptografik özetleme (hash) fonksiyonu ile Şekil 4.7 'de gösterildiği gibi bir özet değer elde edilmektedir.



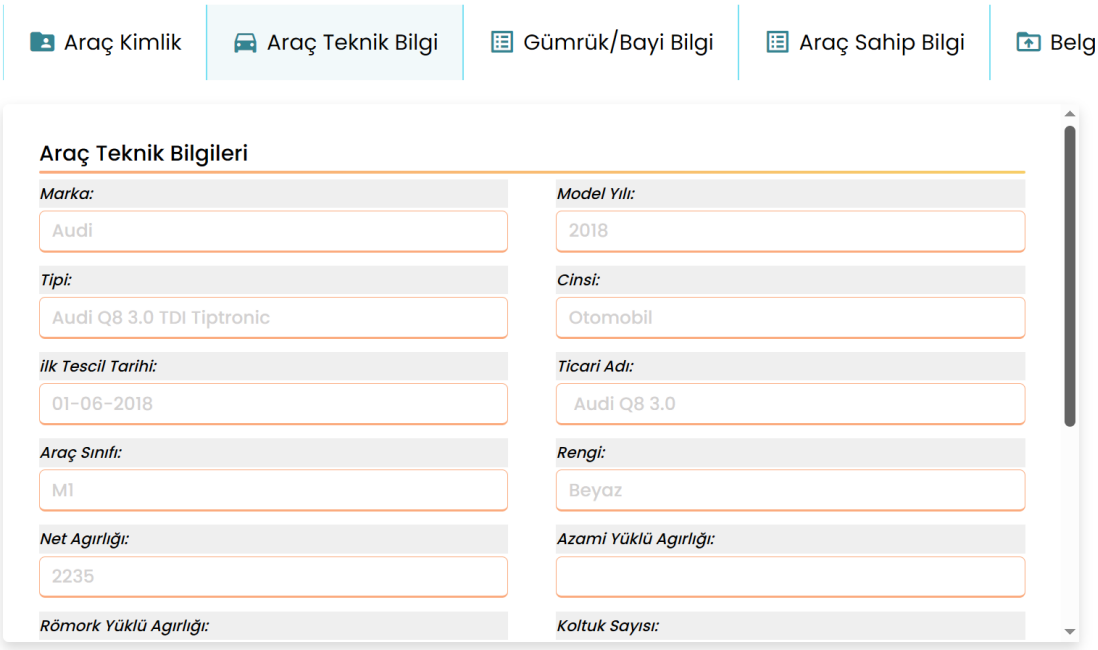
Şekil 4-6: Aracın ilk tescil kaydı için kullanılan ekran görüntüsü



Şekil 4-7: Aracın ilk tescil kaydında, kimlik numarası (VIN) belirlenmesi

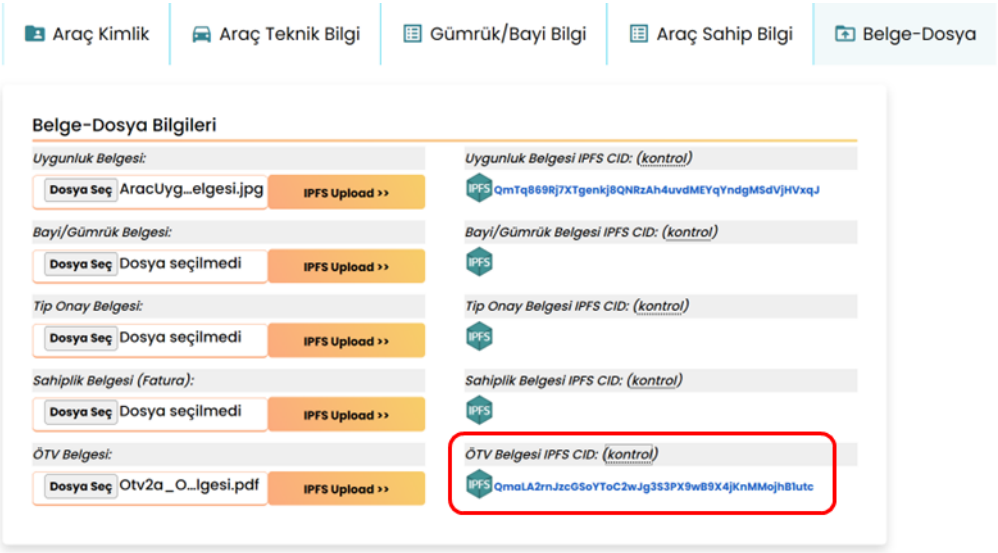
İlk araç tescil işlemlerinde araçla ilgili benzersiz (Şase no, Motor no, Tescil no vb.) tanımlama bilgileri ile birlikte aracın teknik bilgileri (Marka, Model Yılı, Tipi, Cinsi, Ticari Adı,

Araç Sınıfı, Rengi, Net Ağırlığı, Azami Yüklü Ağırlığı, Koltuk Sayısı, Silindir Hacmi, Yolcu Sayısı, Motor Gücü, Yakıt Cinsi, Kullanım Amacı vb.) bilgilerin girilmesi için Şekil 4.8'de verilen ekran kullanılmaktadır.

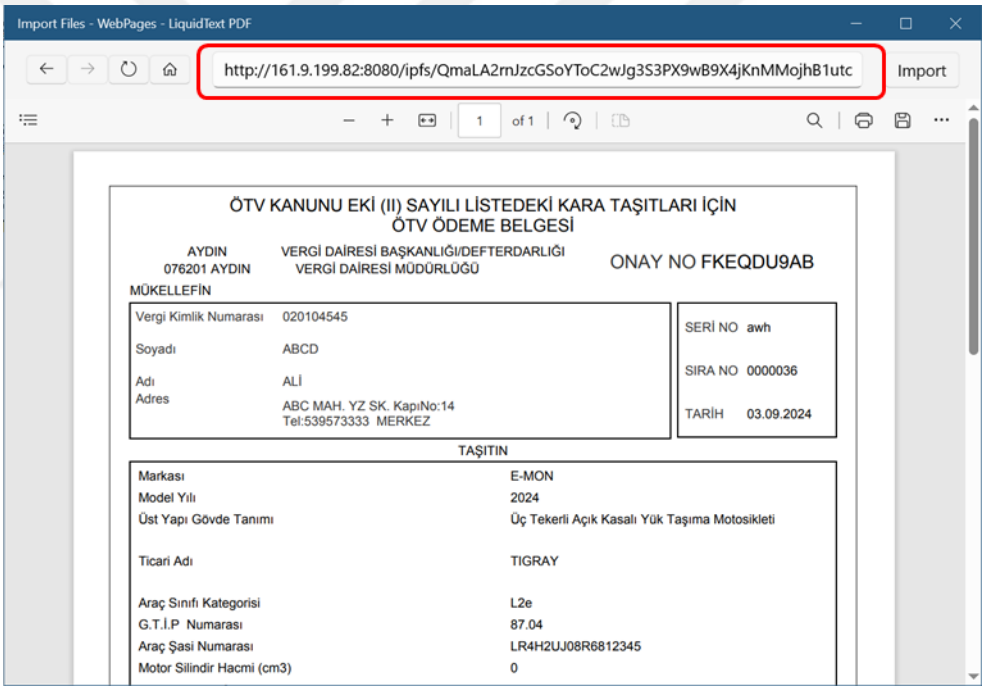


Şekil 4-8: Araç teknik bilgilerinin kaydı için kullanılan ekran görüntüsü

İlk araç tescil işleminin uygulanmasında araçla ilgili teknik bilgiler yanında araca ait belge ve dosyaların da IPFS dağıtık dosya depolama sistemine yüklenmesi gerekir. Şekil 4.9'de verilen ekran görüntüsünde IPFS ağına yüklenen dosyaların elde edilen CID değeri görülmektedir. CID değeri IPFS ağında depolanan dosyaya erişim amacıyla kullanıldığı gibi dosyanın kurcalamaya karşı korumalı olmasını, bütünlüğünün korunmasını sağlar. Bu CID değeri diğer araç verileri ile birlikte blok zincirinde kayıt defterinde saklanmakta Şekil 4.10'de gösterildiği gibi gerektiğinde CID bilgisi ile IPFS ağına ait bir düğüm üzerinden sorgulanarak dosyaya erişim sağlanabilmektedir.

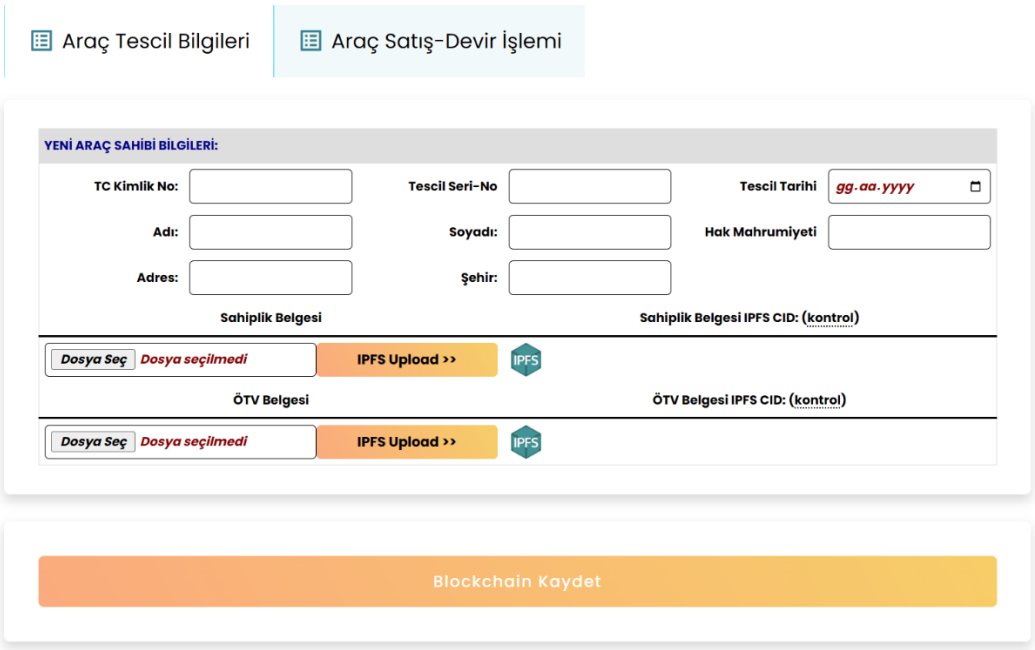


Şekil 4-9: Araca ait belge ve dosyaların IPFS ağına gönderilmesi



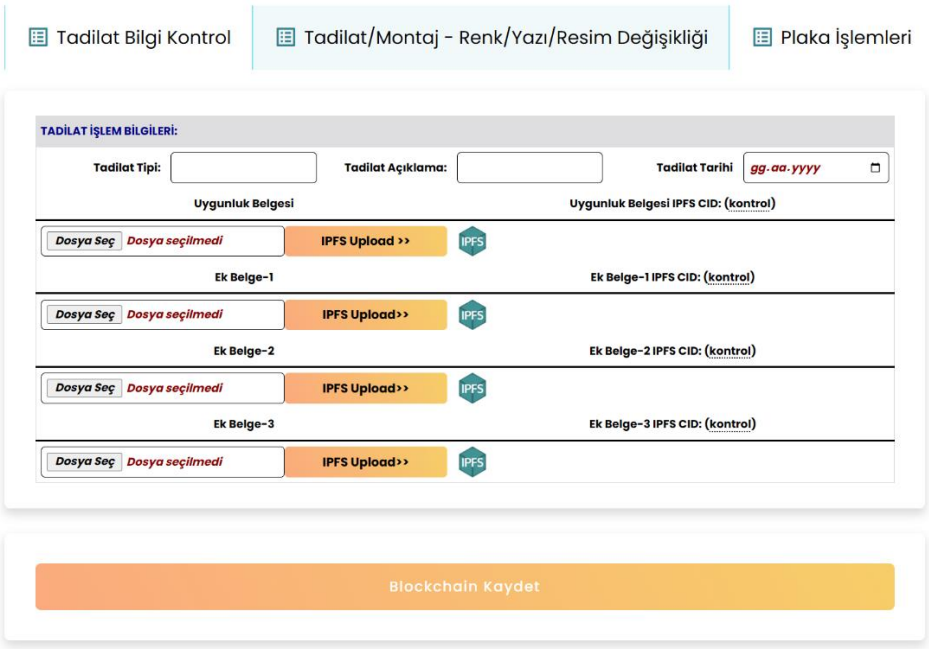
Şekil 4-10: IPFS ağındaki dosyanın CID bilgisi ile erişimi

Satış-Devir Tescili: İkinci el, kullanılmış araç alım satım işlemlerinde gerçekleştirilen mülkiyet devri için, önce aracın sistemde sorgulanarak, aracı tanımlayan bilgilerle birlikte araç sahibinin bilgileri ekrana getirilir. Şekil 4.11'de "Araç-Satış Devir İşlemi" sayfasında yer alan, aracın yeni sahibine ait gerekli bilgiler form alanlarına girilir, araca ait Satış Belgesi/Fatura, varsa ÖTV belgelerinin dijital dosyaları IPFS 'e gönderilerek dosyalara ait CID değerleri oluşturulur. Bilgiler blokzincir ağ sistemine gönderilerek kayıt işlemi tamamlanır.



Şekil 4-11: Araç satış ve devir işleminde kullanılan ekran görüntüsü

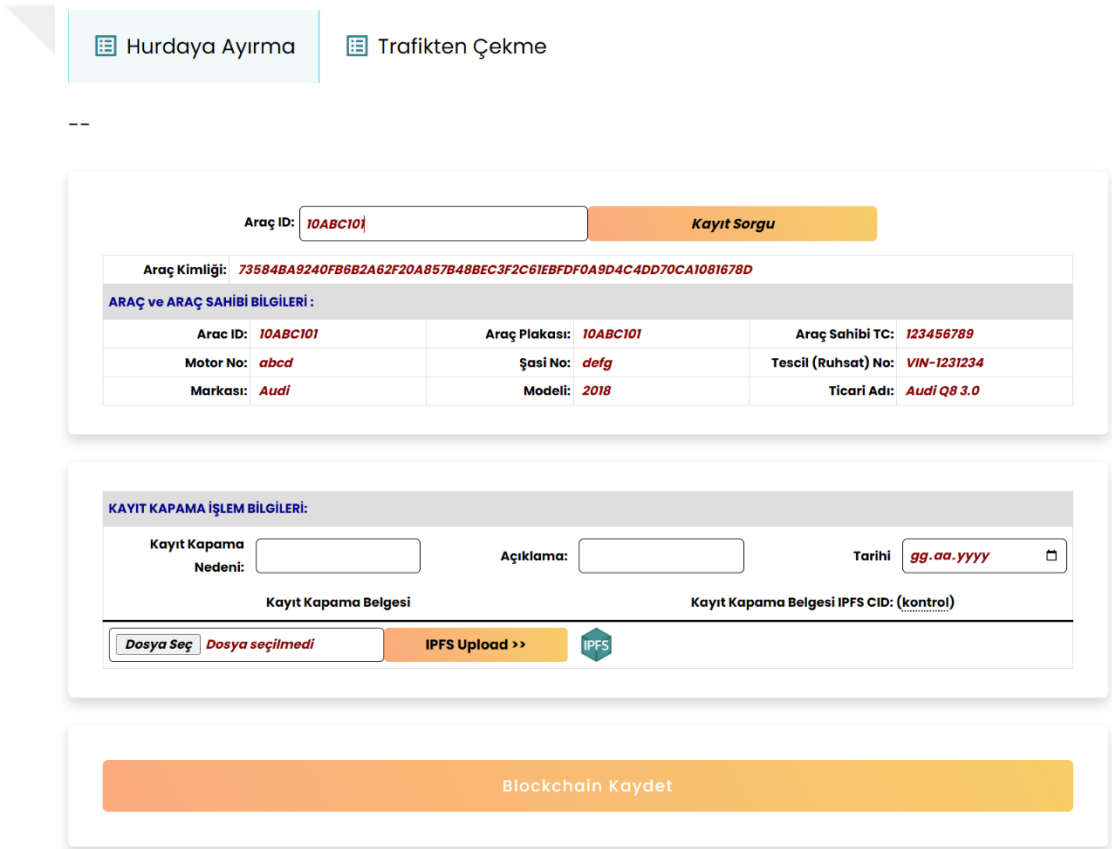
Diğer araç tescil işlemleri, araç üzerinde yapılan tadilat, modifiye, montaj, plaka değişikliği, aracın rengi, yazı ve resim ilaveleri gibi işlemlerinin kayıt altına alınması için yapılır. Şekil 4.12'de verilen ekran üzerinde, yapılan işlemlere ait uygunluk belgesi, proje dosyası ve resim gibi dijital kanıtlar IPFS dosya sistemine gönderilerek elde edilen CID bilgileri ile blokzincir ağ sistemine kaydı yapılır.



Şekil 4-12: Araç tadilat, montaj ve plaka değişimi tescil işlemleri

Kayıt Kapama ve Hurdaya Ayırma:

Araçların belirli bir süre trafiğe çıkamayacağının bildirilmesi, kaydının geçici olarak dondurulması gereken durumlarda aracın durumunu "pasif" olarak değiştirerek araç üzerinde başka bir işlemin yapılmasını engellemek için kullanılabilir. Şekil 4.13'de verilen ekran üzerinden "Trafikten Çekme ve Kayıt Kapama" işlemleri için gereken bilgi ve belgeler sisteme kaydedilir. Benzer şekilde araç ekonomik ömrünü tamamladıysa, ağı hasar nedeniyle bir daha kullanılması mümkün değilse, teknolojik olarak yetersiz hale geldiyse araç hurdaya ayrılarak kaydı kapatılır ve daha sonra araçla ilgili başka bir işlemin yapılması engellenir.



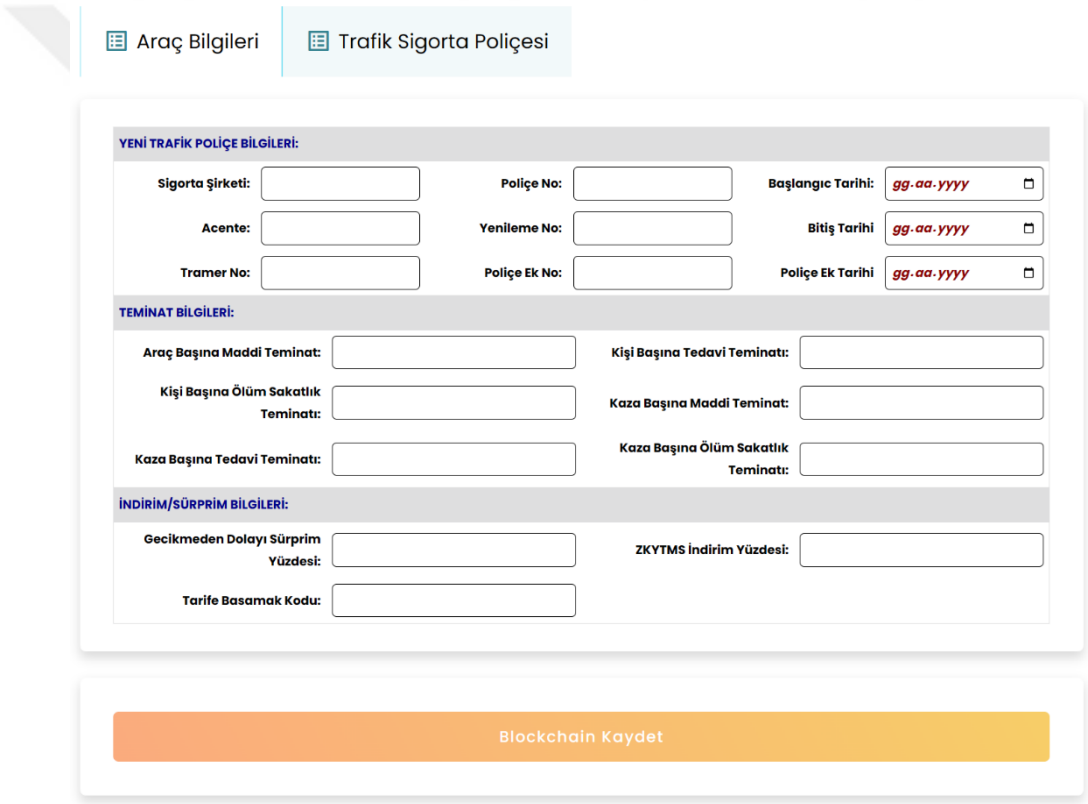
Şekil 4-13: Aracın trafikten çekilmesi yada hurdaya ayrılması

4.2.2 Zorun Trafik Sigortası:

Zorunlu Trafik Sigortası (Karayolları Motorlu Araçlar Zorunlu Mali Sorumluluk Sigortası), her motorlu araç sahibinin yaptırmakla yükümlü olduğu, trafik kazaları sonucu ortaya

çıkan maddi hasar ve bedeni zararların karşılanması için bir güvence sağlayan sigorta türüdür. Sigortasız araç kullanımı ve trafiğe çıkılması yasaktır. Trafik kazası durumunda kusurlu araçların diğer taraflara verdiği zararların karşılanmasında sürücülere destek sağlar.

Tez çalışmamızda, her yıl tekrarlanması gereken Zorunlu Trafik Sigortası bilgilerinin kolayca temin edilmesi ve takibi için blokzincir ağ sisteminde kayıt altına alınmasını sağlayan ekran arayüzü Şekil 4.14'de verilmiştir. Bu ekrana erişim Türkiye Sigortacılar Birliği olarak tanımlanmış olan kuruluşun eş düğümleri üzerinde yer alan istemci uygulamaları ile sağlanmaktadır. Yetkilendirilmiş personel tarafından araca ait önceki sigorta kayıtları görüntülenebildiği gibi yeni hazırlanan sigorta poliçe bilgiler de bu ekran üzerinde girilebilmektedir.



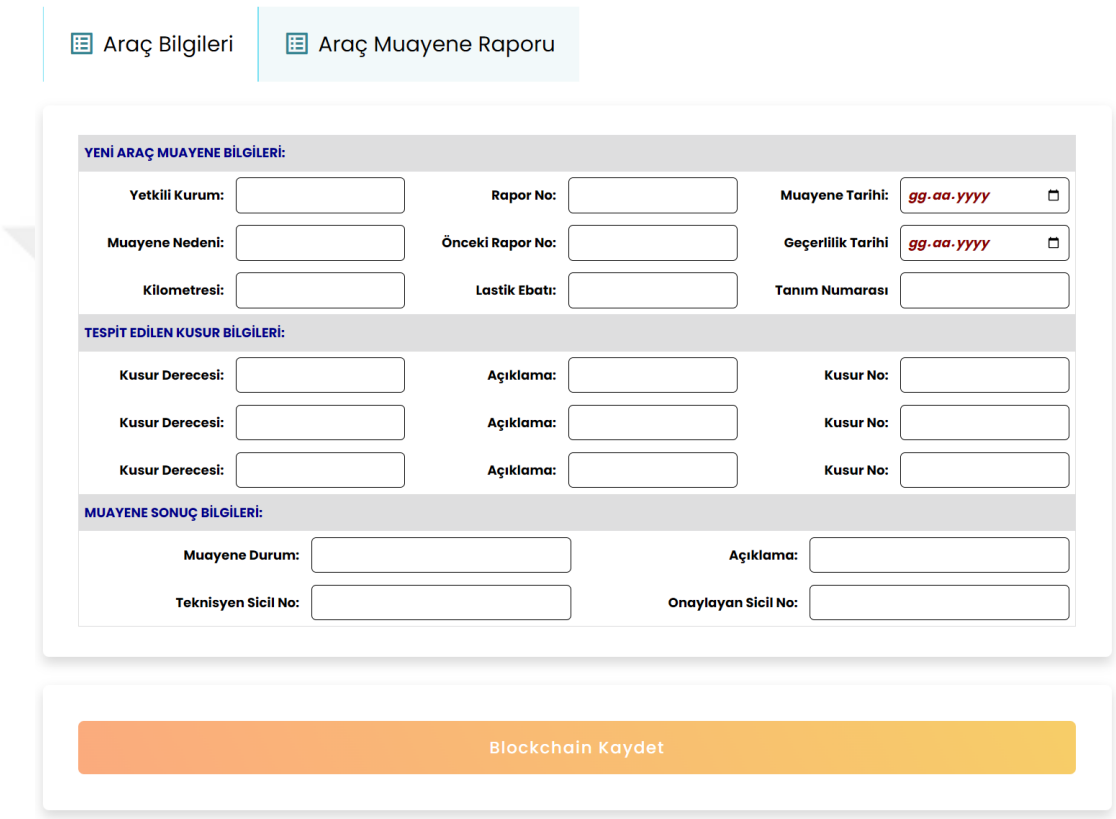
Şekil 4-14: Aracın trafik sigorta poliçe bilgilerinin girilmesi

4.2.3 Araç Muayene ve Egzoz Muayene

Araç ve egzoz muayene işlemleri, karayollarında seyreden araçların çevre sağlığı standartlarına uygunluğunu ve trafik güvenliğini kontrol etmek amacıyla periyodik olarak yapılan yasal bir zorunluluktur. Periyodik araç muayenesi, aracın genel olarak teknik yeterliliğini,

güvenliğini ve mevzuata uygunluğunu kontrol eden kapsamlı bir denetimdir. Türkiye'de bu hizmeti vermeye yetkili tek kuruluş TÜVTÜRK'tür.

Araç ve egzoz muayene işlemlerine ait kayıtların sisteme girilmesi için Şekil 4.15'de verilen ekran kullanılmaktadır. Bu ekran üzerinden araca ait gerekli teknik bilgilere ve geçmişte yapılan muayene raporlarına ulaşılabilmektedir. Yeni yapılan araç ve egzoz muayene bilgi ve belgeleri sisteme yüklenerek geçerlilik tarihi güncellenebilir.



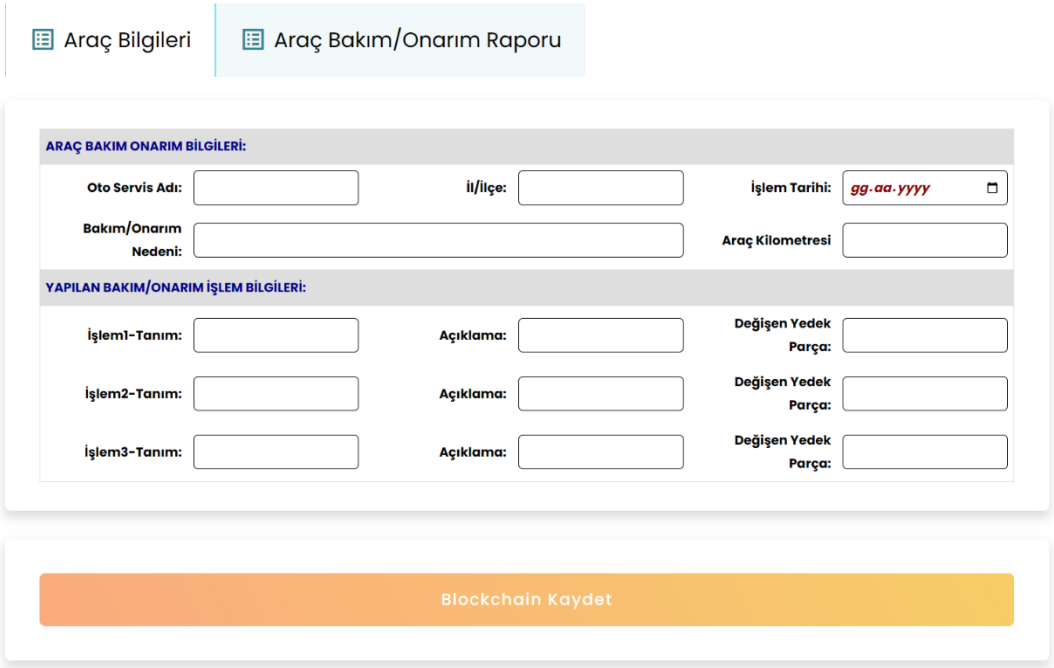
Şekil 4-15: Araç muayene raporu bilgileri giriş ekranı

4.2.4 Araç Tamir-Bakım ve Yedek Parça Değişimi

Araç tamir-bakım ve yedek parça değişimi bilgilerinin kayıt altına alınması oldukça önemlidir. Bu kayıtlar, hem araç sahipleri hem de ikinci el piyasası için şeffaflık, güvenilirlik ve araç değerinin doğru belirlenmesinde temel oluşturur.

Araç tamir-bakım ve yedek parça değişimi işlemlerine ait kayıtların sisteme girilmesi için Şekil 4.16'da verilen ekran kullanılmaktadır. Bu ekran üzerinden aracın daha önce yapılan tüm geçmiş onarım ve bakım bilgilerine ve geçmişte yapılan tadilat ve parça değişim işlem raporlarına

ulaşılabilmektedir. Yeni gerçekleştirilen bakım, onarım ve parça değişimi bilgi ve belgeleri sisteme yüklenebilir.



Şekil 4-16: Araca ait belge ve dosyaların IPFS ağına gönderilmesi

5. SİMÜLASYON UGULAMASI PERFORMANS TESTLERİ

Blokzincir sistemleri yapısal özellikleri nedeniyle yoğun hesaplama yükü ve ağ trafiği oluşturduğu için, sistemin performansı, altyapı kapasitesine, işlem hacmine ve hesaplama yükü yoğunluğuna göre önemli ölçüde değişebilmektedir. Sistemin hangi koşullarda nasıl bir kapasitede çalışabildiğini belirlemek ve sistemin kaynak kullanımını optimize edebilmek için sistemin izlenmesi ve darboğaz oluşturan noktaların belirlenmesi önemlidir.

Tez kapsamında önerilen blokzincir ağ modelinin simülasyon uygulamasında, sistemin performans değerlendirmesini yapmak için Hyperledger Caliper kıyaslama aracı kullanılmıştır. Caliper, Hyperledger çatısı altında geliştirilen Fabric, Besu gibi blokzincir projeleriyle birlikte EVM uyumlu Ethereum blokzincir projelerinin performans testlerini yapmak ve verimlilik analizlerini değerlendirmek için özel olarak tasarlanmış bir kıyaslama (benchmark) aracıdır [73].

Bu bölümde, HLF tabanlı simülasyon uygulamamızın farklı kullanım durumlarını değerlendirmek, ağdaki düğümlere dağıtılan akıllı sözleşmelerin, kademeli olarak artırılan işlem yükleri altındaki tepkilerini ölçmek için Caliper'de gerekli yapılandırmalar yapılmış ve iş yükü modülü programlanarak çeşitli testler uygulanmıştır. Test sonuçlarıyla elde edilen raporlar Excel'de analiz edilerek performans sonuçları değerlendirilmiştir.

5.1 HYPERLEDGER CALIPER TEST ARACI

Linux Vakfı (Linux Foundation) bünyesindeki Hyperledger projelerinden biri olan Caliper, 2018 yılından itibaren halen geliştirilmeye devam etmektedir. Caliper, çeşitli kullanım senaryolarına göre, çeşitli blokzincir sistemlerinin performansını değerlendirmek ve verimlilik ölçümleri yapmak ve platformlar arasında kıyaslama yapmak için kullanılan bir test programıdır. Bu program, işlemlerdeki gecikme süreleri, işlem hızı veya hacmi (TPS, saniyedeki işlem sayısını) CPU, RAM ve ağ kaynaklarının kullanımı gibi performans metriklerini ölçerek, blokzincir ağlarının ölçeklenebilirliğini, güvenilirliğini ve verimliliğini değerlendirmede oldukça önemlidir [73]. Çeşitli blokzincir platformlarının belirli gereksinimlere uygunluğunu değerlendirmek için kıyaslama testleri yapmak amacıyla sıklıkla kullanılmaktadır. Ayrıca akıllı sözleşmelerin farklı iş yükleri ve yoğunlukları altında nasıl performans gösterdiklerinin test edilmesi, sistemin kaynak kullanımı ve gereksinimlerini belirlenmesi, en uygun konfigürasyonların yapılması ve darboğaz oluşturan noktaların belirlenmesi noktasında yazılım geliştiricilere önemli bilgiler sağlamaktadır [74].

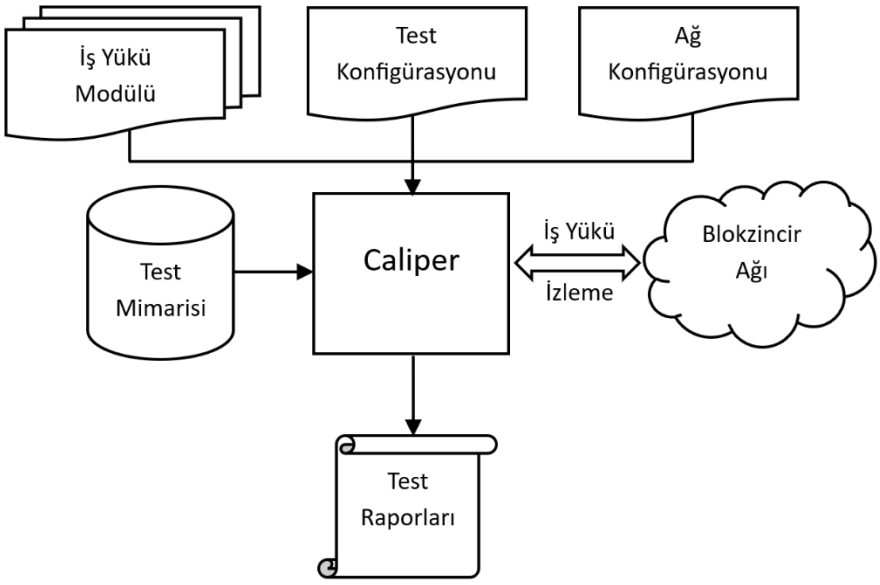
Blokzincir ağında test işleminin yapılabilmesi için, Şekil 5.1'de verilen bağlantı şemasındaki bileşenlerin ve bağlantıların oluşturulması gerekir. Caliper, önceden yapılandırılan test konfigürasyon, ağ konfigürasyon dosyaları ve iş yükü modülü ile birçok senaryonun tasarlanmasını imkan sağlayarak testler yapılmasını ve sonuç olarak, işlem/okuma verimi, işlem/okuma gecikmesi ile ağ, işlemci ve hafıza kaynak kullanımı gibi çeşitli performans metriklerini içeren bir rapor oluşturulmasını sağlar.

Test Yapılandırma Dosyası: Caliper test motorunun nasıl çalışacağının belirlenmesi için, yük oluşturacak istemci sayısının, toplam işlem sayısının ile saniyedeki işlem gönderme hızlarının, işlemi ve iş yükünü oluşturacak modül tanımlamalarının ve izleme ve ölçüm sonuçlarının parametrelerinin belirlendiği YAML tabanlı bir yapılandırma dosyasıdır.

Ağ Yapılandırma Dosyası: Bağlantı kurulacak ve test edilerek olan blokzincir ağ topolojisini tanımlayan ve ağdaki düğümler için kriptografik kimlik bilgilerinin erişim yolu ile düğüm üzerinde hangi akıllı sözleşmeyle etkileşimde bulunulacağının belirleyen YAML tipinde bir dosyadır.

Test Yöneticisi ve İş Yükü Modülü: Node.JS modüllerinden oluşmaktadır. İş yükü ve işlemlerin istenildiği gibi programlanmasına imkan verir. İşlem içeriklerinin oluşturulması ve konfigürasyonda belirtilen bilgi ve ayarlamalara göre akıllı sözleşmeler ile etkileşime girilmesini sağlar.

Test Raporları: Toplanan verilerin raporlanmasını tanımlar. HTML tipinde web sayfası dosyası olarak rapor sonuçlarının tablo ve grafik olarak sunulmasını sağlar.



Şekil 5-1: Caliper ile test edilen blokzincir ağı bağlantı şeması [75]

5.2 PERFORMANS METRİKLERİ

İşlem verimi (tps, saniyedeki işlem sayısı), işlem gecikmesi (sn, saniye) blokzincir ağ performans değerlendirmelerinde kullanılan temel performans ölçütleridir. Test edilen sistemin (SUT), işlem verimi, belirli bir zaman aralığında ve belirli bir sıklıkta gönderilen işlemlerin blokzinciri ağı içinde sonuçlandırılmasıdır. Bu metrik ile her düğümün tek tek faaliyetlerine değil, test edilen blokzincir ağındaki her düğümün toplam performansı ölçülmektedir. İşlem gecikmesi metriği, işlemin ilk gönderildiği andan sonucun ağ genelinde yayılarak erişilebilir olduğu ana kadar geçen zaman aralığının ölçüsüdür. Bu ölçüm, işlemin ağ üzerinde tamamlanması için yayılma süresi, mutabakat mekanizmasının etkinliği ile uzlaşma süresi gibi faktörleri de içine almaktadır [74].

Metrikler ve formülleri:

İşlem Verimi (TPS, Transaction Throughput):

Formül 4.2'de, saniye başına işlenen işlem sayısı (TPS, Transaction Throughput) ölçümünde, ilk işlem mesajının gönderildiği anda başlangıç zamanı ayarlanarak kaydedilir, son mesajın alınması ve blok dosyasına yazılması ile bitiş zamanı belirlenir. Başlangıç ve bitiş zamanı arasındaki geçerli ve başarılı kabul edilen işlemlerin toplam sayısı sayılarak belirlenir. Böylece İşlem Verimi, blokzincir defterine kaydedilen işlem sayısının geçen süreye bölünmesi ile elde edilir [74].

$$geçen\ süre = son\ işlem\ yazılma\ zamanı - \ ilk\ işlem\ gönderim\ zamanı \tag{4.1}$$

$$işlem\ verimi\ (TPS) = \frac{toplam\ onaylanan\ işlem\ sayısı}{geçen\ süre} \tag{4.2}$$

Gecikme (Latency):

Gecikme, işlem teklifinin gönderildiği zaman ile işlemle ilgili cevabın alındığı zaman arasındaki fark ile belirlenir.

$$gecikme = \text{işlem } onay\ zamanı\ x\ ağ\ eşiği - işlem\ gönderim\ zamanı \tag{4.3}$$

İşlem Başarı Oranı (Success Ratio):

İşlem başarı oranı, onaylayıcı eş düğümler tarafından geçerliliği doğrulanmış ve blok zincir defterine başarılı olarak eklenmiş işlemlerinin ölçülmesi için kullanılır. Geçersiz olan işlemlerde iş hacmi içerisinde yer almaktadır. Dolayısıyla başarı oranı geçerli işlemlerin sayısının toplam işlem sayısına bölünmesiyle hesaplanır.

$$Başarı\ Oranı = \frac{Geçerli\ işlem\ sayısı}{Toplam\ işlem\ hacmi} \tag{4.4}$$

Sistem Kaynak Kullanımı:

HLF blokzincir ağında genelde Docker konteynerleri aracılığıyla çalıştırılan, düğümler, sipariş hizmeti düğümü, durum veritabanı gibi Fabric bileşenlerinin sistem kaynak kullanımı Hyperledger Caliper tarafından Docker uygulaması ile etkileşim kurularak izlenebilmektedir. Sistem kaynak metrikleri olarak CPU Kullanımı, Bellek Kullanımı, Ağ Giriş/Çıkış veri miktarı, Disk Okuma/Yazma veri miktarı gibi değerler, maksimum, minimum ve ortalama hesaplamaları ile ölçülebilmektedir.

5.3 TEST ORTAMININ HAZIRLANMASI

5.3.1 Hyperledger Caliper Konfigürasyonu:

Hyperledger Caliper'in test için hazırlanmasında öncelikle test ve ağ yapılandırma dosyalarının ayarlanması gereklidir. Bu dosyalar test senaryosunu, ölçülecek metrikleri ve blokzincir ağına nasıl bağlanılacağını tanımlamada kullanılır.

Tez çalışmamızda hazırladığımız simülasyon uygulamasının testi için ".yaml" dosya türündeki bu yapılandırma dosyaları Şekil 5.2' ve Şekil 5.3'de gösterildiği gibi hazırlanmıştır.

"Test_Senaryo.yaml" dosyasında, Caliper'in hangi testleri çalıştıracağı, kaç işlem göndereceği ve hangi akıllı sözleşme zincir kodu için işlem yükü gönderileceği, testin toplam işlem hacmi ve metrik detayları gibi parametreler tanımlanır.

"Network.yaml" dosyası ile test edilecek olan blokzincir ağına nasıl bağlanılacağı, hangi eş düğümlerin, kanalların ve kullanıcı sertifika kimliklerinin kullanılacağı tanımlanır.

```yaml
name: ARAC TESCIL CALIPER TEST (ORG1)
version: "2.0.0"
caliper:
  blockchain: fabric
channels:
  - channelName: aracsicilsistem
    contracts:
    - id: ArTes
organizations:
  - mspid: Org1MSP
    identities:
      certificates:
      - name: 'User1'
        clientPrivateKey:
          path:
'./sertifikalar/peerOrganizations/org1.example.com/
users/User1@org1.example.com/msp/keystore/priv_sk'
        clientSignedCert:
          path:
'./sertifikalar/peerOrganizations/org1.example.com/
users/User1@org1.example.com/msp/signcerts/cert.pem
'
    connectionProfile:
      path:
'./sertifikalar/peerOrganizations/org1.example.com/
connection-org1.yaml'
      discover: true
```

Şekil 5-2: "network.yaml" konfigürasyon dosyası parametre ve değerleri

```yaml
test:
  # name: ArTes-chaincode-benchmark
  # description: Araç Tescil Kayıt
  workers:
    type: local
    number: 5
  rounds:
    - label: Kayıt (Rate:150tps)
      txNumber: 2000
      rateControl:
        # { type: "linear-rate", opts:
        # { startingTps: 25, finishingTps: 200 } }
        type: fixed-rate
        opts:
          tps: 150
      workload:
        module: test-senaryo/kayit_test.js

    # description: Araç Tescil Sorgu
    - label: Sorgu
      txDuration: 10
      rateControl:
        type: fixed-load
        opts:
          transactionLoad: 1
      workload:
        module: test-senaryo/sorgu_test.js
        arguments:
          assets: 1

# description: Docker; CPU, RAM, I/O, Network benchmarks
monitors:
  resource:
  - module: docker
    options:
      interval: 5
      cpuUsageNormalization: true
      containers:
      - all
```

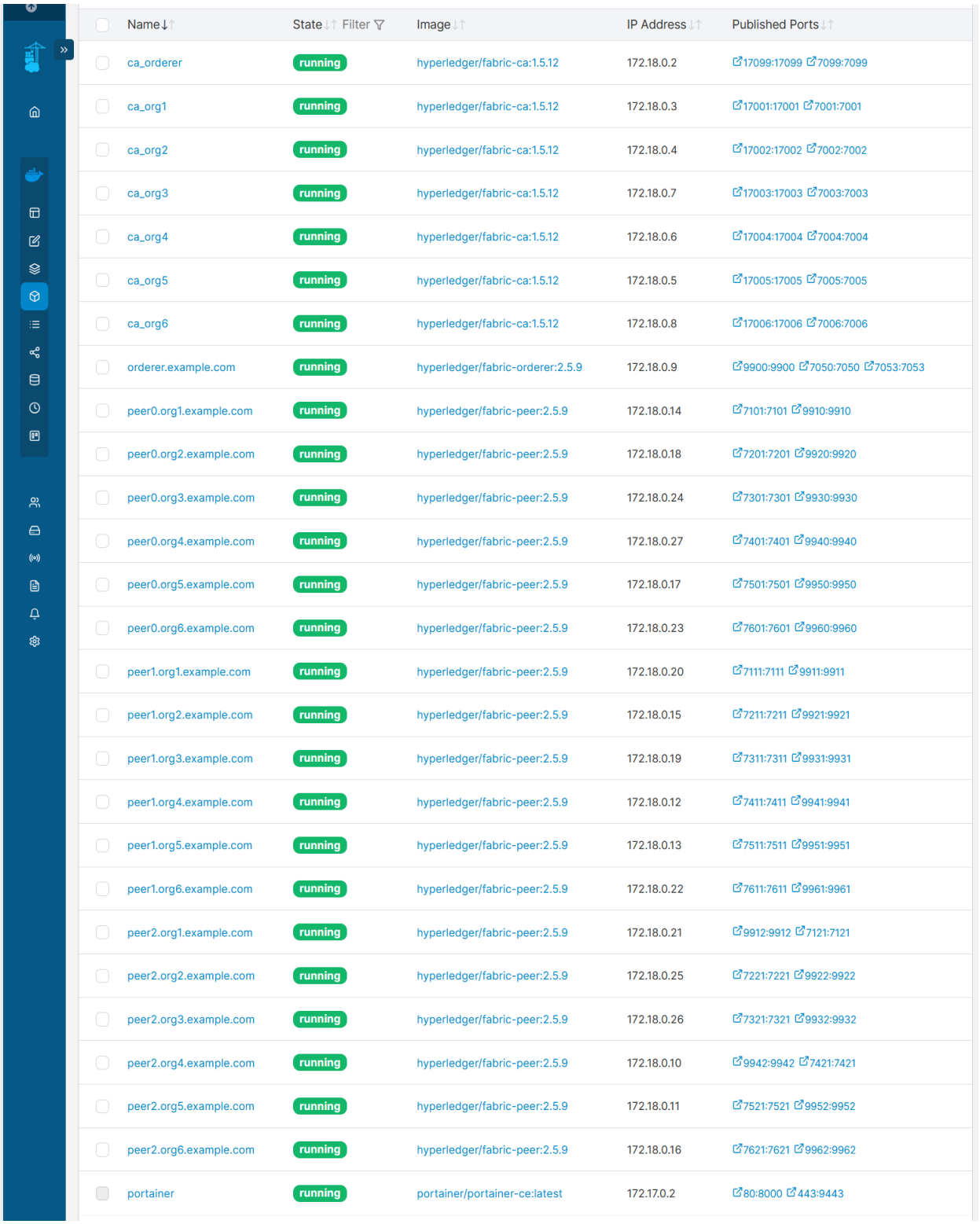
Şekil 5-3: "test_senaryo.yaml" konfigürasyon dosyası parametre ve değerleri

5.3.2 Blokzincir Ağ Konfigürasyonu

Simülasyon uygulamasının performans testinin gerçekleştirildiği ortam olarak sunucu bilgisayar kullanılmıştır. Intel Xeon-8x CPU, 4,6GHz, 32 GB RAM ve 2 TB kapasiteli bir sunucu üzerinde önerilen modelin genel yapılandırması 3. Bölüm de anlatıldığı gibi yapılmıştır. Blokzincir ağlarının geliştirilmesi ve test edilmesinde sıklıkla kullanılan Docker sanal makine uygulaması üzerinde, birbirinden ve ana sistemden izole edilmiş çok sayıda konteyner sanal makineleri ile blokzincir ağı oluşturulmuştur. Test edilen blokzincir ağında 6 kuruluş (Org) ve her kuruluş için üçer adet düğüm (Peer) olmak üzere toplamda tek kanal üzerinde 18 düğüm docker konteyner sanallaştırma yönetimi ile aynı ağda kullanılmaktadır. Her organizasyon ait birer adet sertifika Yönetici (CA) için de docker konteyner bulunmaktadır. Düğüm konteyner içerisinde durum veritabanı olarak LevelDB gömülü olarak bulunmaktadır ancak ağın konfigürasyonunda CouchDB durum veri tabanının kullanımı tercih edildiğinde her düğüm için ayrı ayrı CouchDB konteyner da docker sanal makinası içerisinde oluşturulabilmektedir. Ağda tek kanal üzerinde düğümler, sertifika yöneticileri, durum veritabanları ve bir adet sipariş hizmet düğümü ip adres ve port yapılandırmaları yapılmış bir şekilde Şekil 5.4'de gösterildiği test için hazır hale getirilmiştir.

Fabric Konfigürasyon Yapılandırmaları: HLF blokzincir tabanlı simülasyon uygulamamızda hangi konfigürasyon parametrelerinin blokzincir ağ performansı üzerinde etkisi olduğunu literatürde yapılan çalışmaları inceleyerek belirledik. Yapılan performans test çalışmalarında, Fabric ağlarının performansı üzerinde etkili olan parametrelerin hem donanım hem de yazılım özellikleri bakımdan sınıflandırıldığını ve temelde aynı konfigürasyon parametreleri üzerinde optimum değerlerin belirlenmeye çalışıldığını gözlemledik. Bu parametreler, yazılım için; blok boyutu, blok zamanı ve sıklığı, defter veritabanı, sıralama hizmeti, akıllı sözleşme programlama dili, TLS kullanımı, istemci sayısı, onaylayan eş sayısı, kuruluş sayısı ve onaylama politikası gibi ağ özellik ve bileşenlerinden oluşmaktadır. Donanım için ise; CPU sayısı ve hızı, ayrılan bellek, disk türü ve hızı, ağ hızı gibi bileşenlerden oluşmaktadır [74].

Tez çalışmamız için oluşturduğumuz simülasyon uygulaması tek sunucuda ve Docker sanal makinesinde gerçekleştirildiği için donanım özelliklerinin performans üzerindeki etkisini yapılan diğer çalışmalarla kıyaslamak pek doğru olmayacaktır. Oluşturduğumuz Fabric ağının konfigürasyon parametrelerinin performans üzerindeki etkisini ve en iyi performansı veren değerlerin belirlenmesi için çok sayıda test gerçekleştirdik. Şekil 5.5'de performans üzerinde etkiye sahip olan configtx.yaml dosyası parametreleri verilmiştir.



Name	State	Image	IP Address	Published Ports
ca_orderer	running	hyperledger/fabric-ca:1.5.12	172.18.0.2	17099:17099 7099:7099
ca_org1	running	hyperledger/fabric-ca:1.5.12	172.18.0.3	17001:17001 7001:7001
ca_org2	running	hyperledger/fabric-ca:1.5.12	172.18.0.4	17002:17002 7002:7002
ca_org3	running	hyperledger/fabric-ca:1.5.12	172.18.0.7	17003:17003 7003:7003
ca_org4	running	hyperledger/fabric-ca:1.5.12	172.18.0.6	17004:17004 7004:7004
ca_org5	running	hyperledger/fabric-ca:1.5.12	172.18.0.5	17005:17005 7005:7005
ca_org6	running	hyperledger/fabric-ca:1.5.12	172.18.0.8	17006:17006 7006:7006
orderer.example.com	running	hyperledger/fabric-orderer:2.5.9	172.18.0.9	9900:9900 7050:7050 7053:7053
peer0.org1.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.14	7101:7101 9910:9910
peer0.org2.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.18	7201:7201 9920:9920
peer0.org3.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.24	7301:7301 9930:9930
peer0.org4.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.27	7401:7401 9940:9940
peer0.org5.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.17	7501:7501 9950:9950
peer0.org6.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.23	7601:7601 9960:9960
peer1.org1.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.20	7111:7111 9911:9911
peer1.org2.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.15	7211:7211 9921:9921
peer1.org3.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.19	7311:7311 9931:9931
peer1.org4.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.12	7411:7411 9941:9941
peer1.org5.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.13	7511:7511 9951:9951
peer1.org6.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.22	7611:7611 9961:9961
peer2.org1.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.21	9912:9912 7121:7121
peer2.org2.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.25	7221:7221 9922:9922
peer2.org3.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.26	7321:7321 9932:9932
peer2.org4.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.10	9942:9942 7421:7421
peer2.org5.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.11	7521:7521 9952:9952
peer2.org6.example.com	running	hyperledger/fabric-peer:2.5.9	172.18.0.16	7621:7621 9962:9962
portainer	running	portainer/portainer-ce:latest	172.17.0.2	80:8000 443:9443

Şekil 5-4: Blokzincir ağ bileşenlerinin Docker sanal makine de çalıştırılması

```
Orderer: &OrdererDefaults
    OrdererType: etcdraft
    Addresses:
        - orderer.example.com:7050
    BatchTimeout: 2s
    BatchSize:
        MaxMessageCount: 50
        AbsoluteMaxBytes: 99 MB
        PreferredMaxBytes: 10 MB
```

Şekil 5-5: "configtx.yaml" konfigürasyon dosyası parametre ve değerleri

BatchTimeOut: Blok zaman aşımı değeri, blok oluşumunu başlatan ilk işlem geldikten sonra bloğun bitirilmesi için beklenecek saniye cinsinden süreyi tanımlar. Bu değer çok düşük ayarlanırsa, blokta kabul edilecek işlem sayısına ulaşılmadan bloğun bitirilmesi riski vardır. Ancak bu değerin çok yüksek ayarlanması da sipariş verenin blokları beklemesine ve genel performans ve gecikmenin düşmesine neden olabilir. Bu nedenle bu değerin en az; "maksimum işlem sayısı", "saniye başına maksimum işlem" sayısına oranı ile belirlenmesi daha iyi olacaktır.

Blok Parametre ve Değerleri (BatcSize):

- **MaxMessageCount:** Bir bloka dahil edilecek maksimum işlem sayısını ifade eder. 10 değeri blok içerisine en fazla 10 işlem verisi konulabileceğini her 10 işlem için bir blok oluşturulacağını belirlemektedir. Bu sayıyı artırmak performansın artırılması üzerinde etkili olabilir ancak performansı etkileyebilecek başka etkenlerde vardır. Blok boyutunun büyütülmesi daha fazla işlem gücü gerektirebilir, daha fazla hafıza ve ağ bant genişliği kullanılmasını gerektirebilir.
- **AbsoluteMaxBytes:** Sipariş hizmetinin (Orderer Service) bir bloğun oluşması için izin verilecek en büyük kapasite bayt değerini tanımlar. Hiçbir işlem Mutlak maksimum bayt değerinden daha büyük olamaz. Ayrıca blokların ağda dağıtılması için kullanılan grpc protokolü maksimum 100 Mb'ı desteklediği için, bu değerin 100 Mb'ın altında tanımlanması gerekir.
- **PreferredMaxBytes:** Tercih edilen maksimum bayt değeri ideal blok boyutunu belirler ve bu değer Mutlak maksimum bayttan küçük olmalıdır. Hiçbir onay içermeyen minimum işlem boyutu yaklaşık 1 KB'dir. Gerekli her onay için 1 KB eklenirse, basit bir işlem boyutu yaklaşık 3-4 KB olur. Bu nedenle, "Tercih edilen maksimum bayt" değerinin maksimum mesaj sayısı ile beklenen ortalama işlem boyutunun çarpımına yakın olacak şekilde ayarlanması önerilmektedir. Çalışma zamanında, mümkün olduğunda, bloklar bu boyutu

aşmayacaktır. Bloğun "mutlak maksimum bayt" boyutunu aşmasına neden olan bir işlem gelirse, blok kesilecek ve işlem yeni bir bloğa dahil edilecektir. İstisna olan böyle bir durumda yeni blok sadece tek bir işlemle kesilecek ve bu işlem de "mutlak maksimum bayt" boyutunu aşmayacaktır.

Performans üzerinde etkili olan diğer parametreler “Mutabakat algoritması” ve “Durum veritabanı” seçimidir. Uygulamada sıralayıcı hizmet düğümü mutabakat algoritması olarak RAFT kullanılmıştır. Durum veritabanı olarak LevelDB ve CouchDB kullanımları test edilmiş, CouchDB ’in sağladığı özelliklere ihtiyaç olmaması dikkate alınarak, performansı düşürdüğü bilindiği ve testlerde gözlemlendiği için LevelDB kullanımı tercih edilmiştir.

5.4 TEST SONUÇLARI VE PERFORMANS ANALİZİ

Hyperledger Caliper v0.6.0 test aracı ve Hyperledger Fabric v2.5.9 blokzincir platformunun kullanıldığı simülasyon uygulamasında gerekli yapılandırmalar ve konfigürasyon tanımlamaları yapıldıktan sonra Tablo 5.1’de verilen parametrelere bağlı olarak çok sayıda test gerçekleştirilmiştir. Farklı parametre kullanımlarında yapılan her test en az üçer kez tekrarlanarak ortalama değerler belirlenmiştir. Caliper test raporlarındaki değerler bir araya getirilip Excel tablosunda toplanarak bir veri seti oluşturulmuş ve istatistiksel grafikler elde edilmiştir.

Tablo 5-1: HLF performansı üzerinde etkili olan parametreler ve değerleri

Ağ Konfigürasyon Parametre ve Değerleri

Parametre	Sabit tutulan yada değiştirilebilen değerler
Kuruluş (Organizasyon) Sayısı	6
Kuruluş Başına Eş Düğüm Sayısı – Toplam Eş Düğüm Sayısı	3 - 18
Kanal Sayısı	1
Sıralama Hizmeti Düğüm Sayısı	1
Blok Boyutu (MaxMessageCount) (#TX)	10, 30, 50, 100, 150
Blok Zaman Aşımı (BatchTimeout) (sn)	0.5, 1, 1.5, 2
Durum Veritabanı (World State)	CouchDB, LevelDB
Mutabakat Algoritması	Solo, Raft
Onaylayıcı Eş Düğüm Sayısı (Onay Politikası: OR, AND, OutOf)	6, 4, 2
İşlem Hızı (Saniyede Gönderilen İş Yükü Sayısı) (TPS)	25, 50, 75, 100, 125, 150,175, 200
İstemci Sayısı (Workload)	2, 4, 8

Tablo 5.1'de verilen çeşitli yapılandırılabilir ağ bileşenleri ve parametrelerinin, en iyi verimi elde edebilmek için hangi değerlere sahip olması gerektiğini araştırdığımız testlerde, kuruluş sayısı, eş düğüm sayısı, sıralayıcı düğüm sayısı, kanal sayısı, durum veritabanı, mutabakat algoritması gibi bazı değerlerin sabit tutularak, blok boyutu, işlem hızı, istemci sayısı gibi bazı değerlerin değişimi ile elde ettiğimiz sonuçlar hem yazma hem de okuma işlemleri açısından analiz edilmiştir.

5.4.1 Test Sonuçları ve Değerlendirme

Yazma işleminde blok boyutunun etkisi: Blok boyutunun sistem performansı üzerindeki etkisini faklı işlem gönderme hızları (24tps ile 200tps arası) ile blok boyutunu (10tx ile 150tx arasında) değiştirerek testler gerçekleştirdik. Her blok boyutu değişiminde sistem sıfırlanarak tekrar başlatılmıştır. Testlerin her biri üçer kez tekrarlanarak ortalama değerleri alınmıştır.

Şekil 5.6'da verilen grafik ve Tablo 5.3 değerlendirildiğinde, işlem verimi blok başına 10, 30 ve 50tx değerlerinde küçük bir aralıkta sabit kaldığı gözlenmiştir. Ancak, blok başına 100tx ve 150tx değerlerinde, belirli bir eşik işlem hızı değer aralığına (75tps-100tps) kadar doğrusal olarak artığı, daha sonra bu eşik değerde sabit kaldığı gözlenmiştir. Blok boyutu 100tx ve 150tx iken, işlem gönderme hızı 200tps olduğunda en yüksek işlem hacmi (verimi) sırasıyla 60tps ve 70tps olarak elde edilmiştir. Şekil 5.7'de işlem yazma gecikmesinin test sonuçları görülmektedir. İşlem gecikmesi, işlem gönderim hızının artırılması ile belirli bir eşik değere kadar arttığı ve biraz düşüşle sabit kaldığı görülmektedir. Blok boyutunun artmasıyla sabit işlem hızlarında gecikmenin de bir miktar düştüğü gözlense de, işlem hızlarının yüksel olduğu durumlarda en fazla gecikmelerin yaklaşık olarak sabit kaldığı görülmektedir. Bu durum blok kapasitesinin ve blok zaman aşımı süresinin etkisiyle yeni blokların oluşmasını ve yeni oluşan bloklar içi gerçekleştirilen onay süreçlerinin gecikmeyi artırmasıyla açıklanabilir. Blok boyutunun artırılması, işlem hacmi açısından belirli bir doygunluk noktasına kadar artış sağlasa da belirli bir eşik değer aralığında sabit kalmakta, gecikme değerleri de bu doygunluk noktasında belirli bir aralıkta sabit kalmaktadır. Bu durum performansı etkileyen donanımsal ve yazılımsal diğer parametrelerden kaynaklanabilir.

Sonuç olarak 150 - 200tps işlem hızları ile elde edilebilen yoğun işlemlerde en fazla 33.07 saniye gecikmeye sahip 60 - 70tps aralığındaki işlem hacmi verimi, ağ konfigürasyon parametrelerinden "*MaxMessageCount"* blok boyutu değerinin 100tx olarak ayarlanmasının yeterli olacağını göstermektedir.

Tablo 5-2: Uygulanan testlerde sabit tutulan ve değiştirilen parametreler

Sabit Tutulan Parametreler

Parametre	Değer
Kuruluş Sayısı	6
Düğüm Sayısı (Toplam)	18
Kanal Sayısı	1
Sıralayıcı Düğüm Sayısı	1
Mutabakat Algoritması	Raft
Blok Zaman Aşımı	2 sn
Durum Veritabanı	LevelDB
Onay Politikası	AND(Org1,Org2,Org3,Org4,Org5,Org6)
İstemci Sayısı	2

Değiştirilen Parametreler

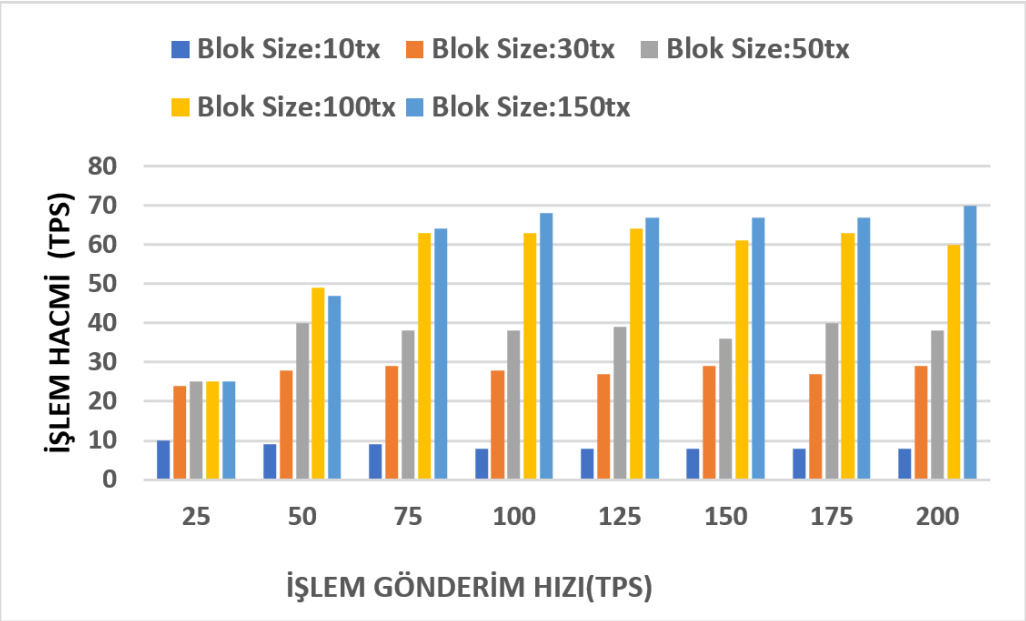
Parametre	Değer
Blok Boyutu	10, 30, 50, 100, 150, 200 (#tx)
Gönderilen İşlem Sayısı	25, 50, 75, 100, 125, 150,175, 200 (TPS)

Tablo 5-3: Yazma işlemi verimi (tps) değerler tablosu

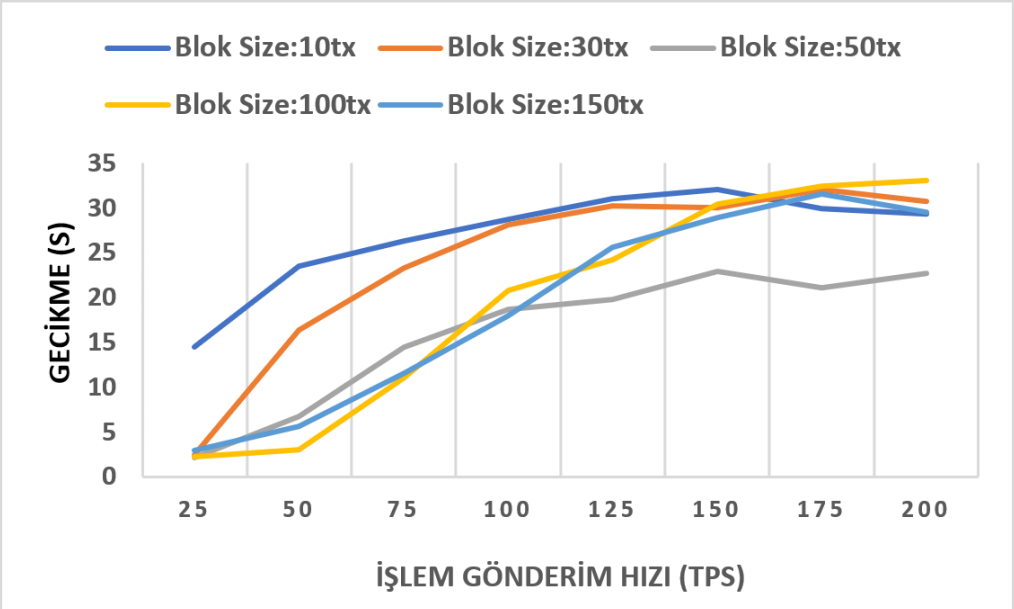
		Blok Boyutu (tx)				
		10	30	50	100	150
İşlem Hızı (tps)	25	10	24	25	25	25
	50	9	28	40	49	47
	75	9	29	38	63	64
	100	8	28	38	63	68
	125	8	27	39	64	67
	150	8	29	36	61	67
	175	8	27	40	63	67
	200	8	29	38	60	70

Tablo 5-4: İşlem gecikme değerler tablosu

		Blok Boyutu (tx)				
		10	30	50	100	150
Gecikme (s)	25	14.48	2.46	2.13	2.22	2.91
	50	23.46	16.4	6.69	3.05	5.67
	75	26.3	23.32	14.46	11.08	11.51
	100	28.71	28.1	18.64	20.75	17.95
	125	31.01	30.22	19.8	24.22	25.59
	150	32.03	30.06	22.94	30.42	28.92
	175	29.98	32.06	21.14	32.46	31.57
	200	29.35	30.74	22.71	33.07	29.53



Şekil 5-6: Yazma işlemi verim (tps) grafiği



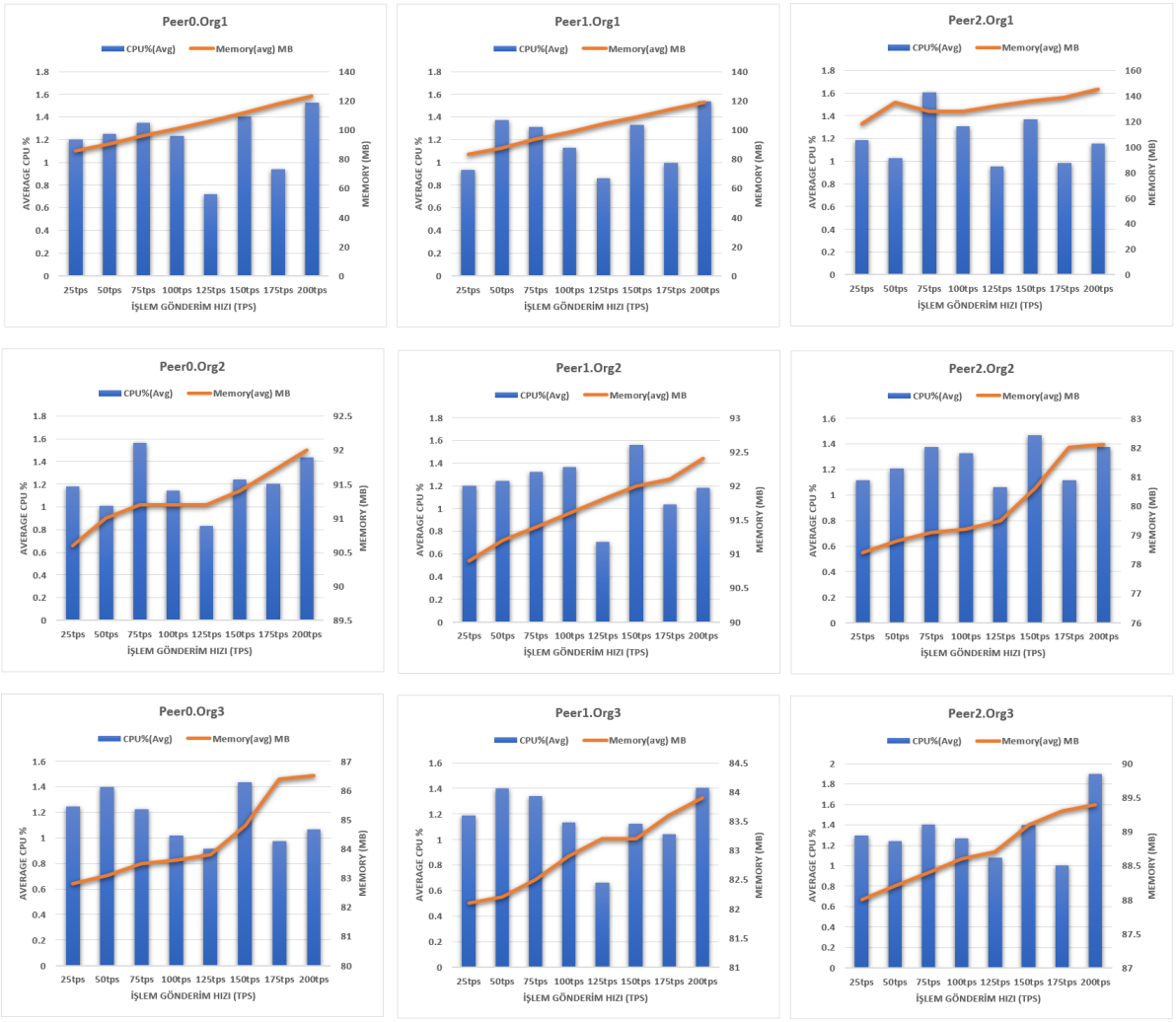
Şekil 5-7: Yazma işlemi gecikme (s) grafiği

Yazma işleminde donanım kaynak kullanımı: Blokzincir ağ bileşenleri tek sunucu bilgisayarda ve Docker sanal makine uygulaması üzerinde çalıştırılmaktadır. Caliper test aracı Docker üzerindeki tüm konteyner uygulamalarının donanım kaynak kullanım değerlerini istenilen zaman aralığında alabilmektedir. Simülasyon uygulamamızın Docker üzerindeki konteynerler ağ bileşenleri için yapılan her test için kaynak kullanım değerleri alınarak Excel uygulamasında birleştirilmiş ve bir veri seti oluşturulmuştur. Şekil 5.8'de verilen grafiklerde, elde ettiğimiz veri

seti ile her eş düğüme ait normalleştirilmiş CPU kullanım değeri, Hafıza kullanım değeri ağ giriş çıkış trafiği, disk yazma okuma değerleri gibi donanım kaynak kullanım bilgileri ile her eş düğüme için CPU ve Hafıza kullanımına ait istatistiksel grafikler oluşturulmuş ve analiz edilmiştir.

Şekil 5.8'de verilen grafiklerde üç kuruluşun üçer adet eş düğümlerine ait CPU ve Hafıza kullanımlarının kıyaslanması için yana yana verilmiştir. Diğer kuruluşların grafikleri de birbirine çok yakın ve benzerdir. CPU kullanım değerleri normalleştirilerek test raporlarına yazdırılmıştır. Bunun nedeni Docker sanal makinesinde CPU kullanımını vCPU ile paylaştırması bazen eş düğümlerin kendilerine ayrılan vCPU kullanımı %100 aşan değerlere ulaşmasıdır. Kıyaslamaların daha adil ve daha anlamlı yapılabilmesi için vCPU kullanımları normalleştirilmiştir.

Şekil 5.8'de verilen grafik değerlendirildiğinde, işlem gönderme hızları (25tp-200tps aralığında) artırıldığında, normalleştirilmiş vCPU kullanım değerlerinin her eş düğüm üzerinde 1 ile 1.5 aralığında dalgalı bir şekilde değişkenlik gösterdiği gözlenmiştir. Hafıza kullanımlarının işlem gönderme hız artışlarıyla birlikte doğrusal bir artış gösterdiği görülmektedir.



Şekil 5-8: Yazma işlemlerinde eş düğümlerin kaynak kullanım durumları

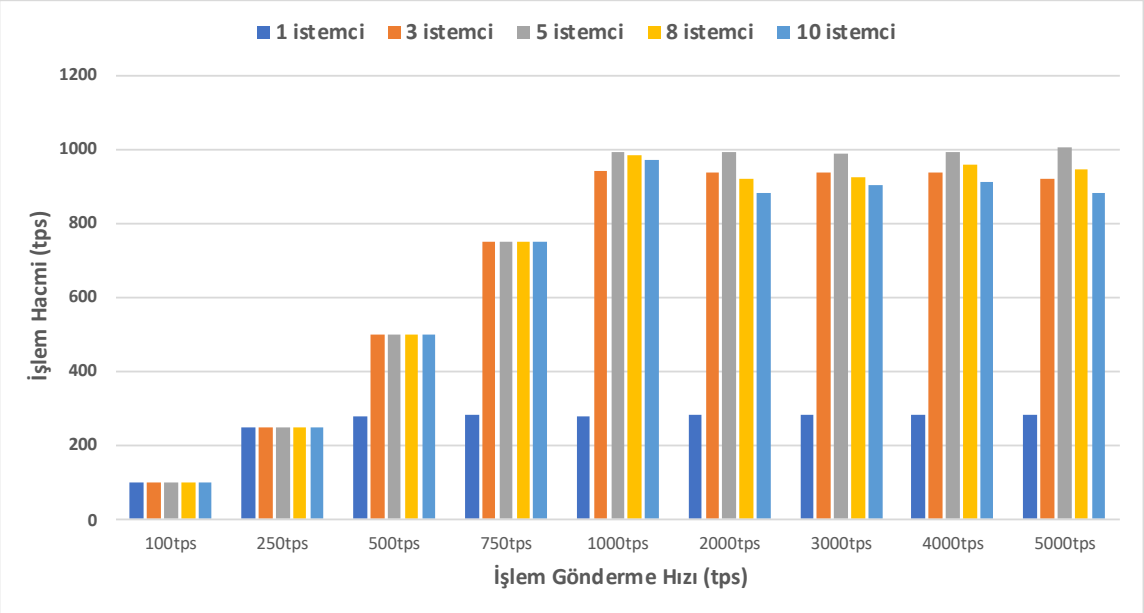
Okuma işleminde işlem gönderme hızı ve istemci sayılarının etkisi: HLF ağlarında işlem okuma/sorgulama işlemleri onaylama ve sıralama hizmeti süreçlerini gerektirmez. Dolayısıyla yazma işlemine kıyasla performans değerlemesi yapılmaz. Sorgulama işlemi kayıt defterinin mevcut durumunu sorgulama amacıyla kullanılır. HLF de durum veritabanları (LevelDB ve CouchDB) kayıt defterlerinin mevcut durumuna ait kayıtları anahtar-değer ikilisi ve JSON veri yapısında tutar. Sorgulamanın yapıldığı eş düğümde yer alan durum veritabanı üzerinden sorgulamalar yapıldığı için herhangi bir onay politikası süreci işletilmez ve sıralama hizmeti yapılmaz. Bu basitleştirilmiş süreç nedeniyle okuma işlemleri yazma işlemlerine göre genellikle daha hızlı gerçekleşir. Simülasyon uygulamamızda okuma işlem performansını ölçmek ve değerlendirmek için istemci sayısını (1, 3, 5, 8, 10 değerleri ile) ve işlem gönderme hızını (100tps ile 5000tps aralığında) kademeli artırarak sorgulama testleri yapılmıştır.

Şekil 5.9'da verilen grafik değerlendirildiğinde 1 adet istemci üzerinden gönderilen işlem hızı artırılsa bile gönderim hızının 285tps sevilerinde kaldığı dolayısıyla okuma veriminin de aynı

seviyede kaldığı gözlenmiştir. 3 istemci için de 945tps işlem hacmi üzerine çıkılamamıştır. Diğer istemci sayılarında 1000tps ve üzerindeki işlem gönderim artışına rağmen işlem hacmi 1000tps'nin altında kalmaktadır. Sonuç olarak, istemci sayısının 3 ve üzeri değerleri için sistemin sorgulama kapasitesinin 1000tps'nin altında olduğu söylenebilir.

Tablo 5-5: Okuma işleminde ölçülen değerler

		İstemci Sayıları				
		1 istemci	3 istemci	5 istemci	8 istemci	10 istemci
İşlem Hızı (tps)	100tps	100	100	100	100.1	100
	250tps	250	250	250	250	249.9
	500tps	278.2	500.1	500	499.3	499.6
	750tps	283.3	750.1	750	749.2	749.7
	1000tps	278.7	942.1	993	981.6	971.6
	2000tps	283.3	936.6	993	921	882.3
	3000tps	280.5	935	988.8	926	901.1
	4000tps	283.8	938.9	993.8	959.9	909.3
	5000tps	281.2	921.6	1006.8	943.8	880.3



Şekil 5-9: Okuma işlemi verim grafiği

Önerilen model için uygulanan simülasyon çalışmasında yapılan performans analiz çalışmalarının bazı kısıtlamalarını dikkate almak gerekir. Testler tek bilgisayarda ve sanal makine üzerinde gerçekleştirilmiştir. Gerçek dünya kullanım ortamlarında blokzincir ağındaki eş düğümler farklı coğrafi konumlarda veya bulut bilişim sistemlerinde dağıtılmış halde olabilir. Test

yazılım tabanlı bir ağ ortamında gerçekleştirilmekte ve ağ bileşenleri bilgisayar kaynaklarını ortak ve paylaşımlı bir şekilde kullanmaktadır. Fiziki bir ağ ortamı performans üzerinde etkili olabilecek farklı parametre ve sınırlamalara sahiptir. Bunun yanında eş düğümlerin gerçek donanım kaynaklarına sahip olması, işlemci ve hafıza kullanımları açısından performansı artıran bir ortam sunabilir. Dolayısıyla gerçekleştirilen performans testleri geliştirilen blokzincir ağındaki darboğaz noktalarını anlamamıza, konfigürasyon parametrelerinin performans üzerindeki etki düzeylerini fark etmemiz de bize fayda sağlamaktadır. Geliştirilen uygulamanın performans ölçümleri ve verim analizi, uygulamanın gerçek dünya kullanımına hazır olup olmadığı hakkında da bize bilgi vermesi acısından önemlidir.

Mevcut donanım ve yazılım koşullarında gerçekleştirilen performans test sonuçları, blok boyutu 150 işlem hacminde iken, saniyede 200tps işlem iş yükü altında, işlemlerin kayıt defterine yazılması saniyede 70tps işlem hacmine ulaşmaktadır. Aynı koşullarda işlemlerin onay gecikme süresi en fazla 33.07 saniye olarak ölçülmüştür. Okuma-sorgulama işlemlerinde ise 10 istemci üzerinden yapılan saniyede 1000tps işleme kadar ki iş yükü altında %99'a varan bir verimlilik elde edilmiştir Ancak iş yükü saniyede 1000tps işlemin üzerine çıktığında işlem gönderim hızı belirli bir eşik değerin üzerine çıkamadığı için verim düşmektedir. Bitcoin blokzincir platformundaki 1 MB blok boyutu, 10 dakikada bir blok oluşturulabilmesi, saniyede 7tps'lik işlemle sınırlı olması [76], ayrıca Ethereum blokzincir ağında bir işlemin onay süresinin 15 saniye olması ile kıyaslandığında yeterli olmasa da araçlarla ilgili yapılan işlem kayıtlarının saniyeler içinde yapılmasını sağlayabilecek kapasitededir.

6. SONUÇ VE ÖNERİLER

Bu tezde, her bir araç için benzersiz bir kimlik numarası tanımlanarak, trafiğe çıktığı ilk tescil tarihinden hurdaya ayrıldığı zamana kadarki yaşam döngüsü içerisinde gerçekleştirilen birçok işlem geçmişinin, Hyperledger Fabric blokzincir ağ platformu üzerinde kronolojik olarak zaman damgası ile kayıt altına alınması ve yönetilmesini sağlamak için "Blokzincir Tabanlı Araç Kimlik ve Sicil Sistemi (BAKSİS)" modeli önerilmiştir. Ayrıca, kayıt altına alınması gereken büyük boyutlu bilgi ve belgelerin güvenli bir şekilde depolanması, kontrollü bir şekilde erişilebilmesi için önerilen blokzincir tabanlı modelimiz, özel ve izinli bir IPFS dağıtık dosya depolama sistemiyle entegre edilmiştir. Bu entegrasyon sayesinde kanıt niteliğindeki bilgi ve belgelerin bütünlüğü ve gizliliği korunurken, blokzincirin sağladığı güvenli, güvenilir ve yedekli bir ağ ortamında depolanması sağlanmıştır. Önerilen modelin izinli ve özel bir blokzincir ağı olması, bilgiye erişim ve paylaşımın belirli yetki ve izinlere göre yapılmasını sağlayarak gizlilik ve mahremiyetin korunmasını da desteklemektedir.

BAKSİS modelinin simülasyon uygulamasında; EGM trafik başkanlığı, Türkiye Noterler Birliği, Türkiye Sigorta Birliği, Sigorta Şirket ve Acenteleri, Araç Üretici ve Bayileri, Oto Tamirhane ve Özel Servisler, Araç Muayene Şirketleri gibi kuruluşlar için üçer adet düğüm, toplamda 18 adet düğümden oluşan tek kanallı HLF blokzincir ağı, tek sunucu bilgisayarda ve Docker sanal makine uygulaması üzerinde oluşturulmuştur. Simülasyon uygulamasında, her kuruluş için, blokzincir ağı ile etkileşim kuracak olan istemci web arayüzleri zincir dışı çalışacak şekilde tasarlanmıştır. İstemci web arayüzleri üzerinden kuruluşun yetkilendirilmiş ve süreli sertifika ve dijital imzaya sahip kullanıcı rolündeki personeli, kuruluşların her biri için ayrı ayrı tanımlanmış işlemleri yapabilmektedir. Her yapılan yazma ve okuma işlemi için çok sayıda akıllı sözleşme kodları tasarlanmıştır. Ayrıca zincir dışı çalışan 4 düğümlü özel bir IPFS ağı Docker sanal makine uygulamasında çalıştırılarak blokzincir ağıyla entegrasyonu sağlanmıştır.

Önerilen BAKSİS modeli, araçlarla ilgili yapılacak olan işlemlerde rol alan resmi ve özel kurumlara, güvenilir bir kaynaktan ihtiyaç duydukları verileri alabilmesi ve doğrulama yapabilmesine olanak sağlandığı gibi, gerçekleştirdikleri işlemlere ait bilgileri, yetki ve izinlerine göre kayıt altına alıp paylaşabilmelerini de mümkün hale getirmektedir. Kurumlar, blokzincir ve IPFS ağında eş düğüm rolündeki sunucuları ile kendi merkezi veri yönetim sistemleri arasında entegrasyon sağlayarak araçlarla ilgili güncel bilgilere anında ulaşabileceklerdir. Böylece bilgilerin geçerliliğinin belirlenmesi ve doğrulanması için ıslak imzalı ve kâğıt tabanlı belge kullanımına gerek kalmamakta, kurumlar arası güvensizlikler de ortadan kalkmaktadır.

BAKSİS modelimiz, Hyperledger Fabric gibi yenilikçi ve özelleştirilmiş bir blokzincir teknolojisi altyapısına sahip olduğu için, sistemde yer alan kuruluşların iş süreçlerine yeni bir yaklaşım getirerek iyileştirilmesine ve hatta tamamen değiştirilmesine sağlayacak bir potansiyeldedir. Bu sistemin ve teknolojinin benimsenmesi, işlemlerin online ve daha hızlı yapılması, kurumların verimliliğini artırılabilecek, zamandan tasarruf sağlanarak uzun vadede maliyetlerinin önemli ölçüde azaltılmasına katkı sağlanabilecektir.

Blokzincir teknolojisinin kullanıldığı BAKSİS sisteminde işlemlerin değiştirilemez oluşu güvenilirliği artırmaktadır. Kayıt defterinde tutulan veriler üzerinde sonradan oynama yapılması veya silinmesi mümkün değildir. Kayıtlarda eş düğüm ve kullanıcıların dijital kimlikleri de yer aldığı için, inkâr edilemezlik sağlar ve sorumluların belirlenmesinde kanıt oluşturur. Her bir aracın ilk kaydında oluşturulan benzersiz ve değiştirilemez araç kimliği (VIN) araçların zaman damgalı geçmiş kayıtlarına erişmede kolaylık ve şeffaflık sağlamaktadır. Bu durum araçlar üzerinde çeşitli hile ve dolandırıcılıkların yapılmasını engelleyecektir. Araçlara ait bilgilerin ortak bir platform üzerinde blokzincir teknolojisinin getirdiği yeniliklerle yönetilmesi sayesinde, kaza, hasar ve poliçe bilgileri üzerinde oynama yapılarak gerçekleştirilen sigorta dolandırıcılıklarını, çalıntı araçların sahte belgelerle yeniden tescil edilmesini ve satılmasını, kilometre sayacı sahtekarlıklarını önleyebilecek bir potansiyeldedir. Araçların geçmişine ait, zaman damgalı bakım ve onarım kayıtlarının, yedek parça kullanım detaylarının, kaza ve hasar bilgilerinin kolayca erişilebilir ve şeffaf olması ikinci el kullanılmış araçlarda yaşanan güvensizlikleri de ortadan kaldıracaktır.

Önerilen BAKSİS modelinin performans analizi için Hyperledger Caliper test aracı kullanılmıştır. Modelin simülasyon uygulaması, sunucu özellikli, linux işletim sistemine sahip ve Docker sanal makine uygulaması ile donanımsal kaynakların ortak kullanıldığı ve paylaşıldığı bir bilgisayar da çalıştırılmıştır. Mevcut donanım ve yazılım koşullarında gerçekleştirilen performans test sonuçları, blok boyutu 150 işlem hacminde iken, saniyede 200tps işlem iş yükü altında, işlemlerin kayıt defterine yazılması saniyede 70tps işlem hacmine ulaşmaktadır. Aynı koşullarda işlemlerin onay gecikme süresi en fazla 33.07 saniye olarak ölçülmüştür. Okuma-sorgulama işlemlerinde ise 10 istemci üzerinden yapılan saniyede 1000tps işleme kadar ki iş yükü altında %99'a varan bir verimlilik elde edilmiştir Ancak iş yükü saniyede 1000tps işlemin üzerine çıktığında işlem gönderim hızı belirli bir eşik değerin üzerine çıkamadığı için verim düşmektedir. Bu test sonuçları, Bitcoin'in saniyede 7 TPS işlem hacmiyle, onay süresinin ve gecikmenin bir saati bulması, Ethereum'un 15-30 TPS aralığında işlem hacmine sahip olmasıyla kıyaslandığında, araçlarla ilgili işlem kayıtlarının yoğun işlem yükleri altında bile saniyeler içinde

gerçekleştirebilecek kapasitede olduğunu göstermektedir. Bu testler, geliştirilen blokzincir ağındaki darboğazları ve konfigürasyon parametrelerinin performans üzerindeki etkilerini anlamamızı sağlamıştır. Uygulamanın gerçek dünya kullanımına hazır olup olmadığına dair fikir vermesi açısından da bu performans ölçümleri ve verim analizi büyük önem taşımaktadır.

Kısıtlar:

Blokzincir teknolojisinin bilinirliği ve potansiyeli henüz tam anlamıyla anlaşılmış değildir. Finans uygulamalarıyla sınırlı olduğu düşünülen ve kripto para uygulamalarıyla kendini ispat eden bu teknolojisinin birçok sektörde kullanım alanına sahip olduğunu gösteren çalışmalar, henüz pilot uygulamalarla geliştirilmeye devam etmektedir. Yeni ortaya çıkan birçok teknolojide olduğu gibi blokzincir teknolojisinin de henüz çözümlenmemiş problemleri ve ölçeklenebilirlik sorunları vardır. Önerilen modelde tercih edilen Hyperledger Fabric blokzincir platformu ölçeklenebilirlik sorununa, modüler yapısı sayesinde, değiştirilebilir konsensüs mekanizmaları, yeni eş düğüm ve kanal eklemeleri, güncellenebilir akıllı sözleşme kodları ve durum veritabanı seçenekleri ile bazı çözümler getirmiştir. Böylece önerilen modelde olmayan sonradan ilave ve edilmesi gereken araçlarla ilgili yeni hizmetlerin veri yönetimi ve takip işlemleri kolayca sisteme entegre edilebilmektedir. Önerilen modelde kısıtlı da olsa ölçeklenebilirliğin olması, modelin ileriye yönelik uzun süre kullanımına ve ihtiyaca göre yeniden düzenlenebilmesine olanak sağlayacaktır.

Modelin simülasyon tabanlı performans analizlerinde bazı kısıtlamalar göz önünde bulundurulmalıdır. Testler tek bir bilgisayar ve sanal makine ortamında yapıldığından, gerçek dünya senaryolarındaki coğrafi olarak dağınık veya bulut tabanlı eş düğümlerin performans üzerindeki etkileri tam olarak yansıtılamamıştır. Ayrıca, test yazılım tabanlı bir ağ ortamında yapıldığı için fiziksel ağın kendine özgü parametreleri ve kısıtlamaları dışarıda kalmıştır. Gerçek donanım kaynaklarına sahip eş düğümlerin işlemci ve hafıza kullanımı açısından performansı artırabileceği de unutulmamalıdır.

Sonuç olarak, önerilen model, Hyperledger Fabric ve IPFS gibi yenilikçi teknolojileri entegre ederek, paydaşların rol ve sorumluluklarına göre tüm araç yaşam döngüsü geçmişini daha verimli ve güvenli bir şekilde izlemelerini ve yönetmelerini sağlamaktadır. Önerilen blokzincir tabanlı model, mevcut durumda birbirinden bağımsız faaliyet gösteren kuruluşların geleneksel iş modellerini önemli ölçüde dönüştürme potansiyeline sahiptir.

Öneriler:

Tez kapsamında önerilen modelin uygulanmasında bazı zorluklar bulunmaktadır. Önerilen blokzincir sisteminin gerçek hayat uygulamasının başarısı, araçlar üzerinde işlemleri gerçekleştiren küçük-büyük birçok özel şirketin ve ilgili devlet kurumlarının sistemi

benimsemelerine, iş birliğinin sağlayacağı faydalara inanmalarına ve veri girişlerinde gerekli hassasiyeti göstermelerine bağlıdır. Gerekli yasal düzenlemelerin yapılması, hukuki belirsizliklerin kaldırılması ve özendirici teşviklerle birlikte, araştırma, geliştirme ve altyapı faaliyetlerine yatırımların yapılmasıyla, önerilen modelin hayata geçirilmesi ve istenilen faydaları sağlaması mümkündür.

Devletin düzenleyici ve denetleyici işlevleri önerilen model ile daha kolay hale getirilmektedir. Araçlara ait tutulan kayıtlar hukuki konularda kanıt olarak kullanılabilir. Araç sahiplerinin yükümlüğünde olan ve periyodik olarak tekrarlanan araç muayenesi, egzoz muayenesi, motorlu taşıtlar vergisi, araç sigortası gibi işlemlerin geçerlilik tarihleri takip edilerek sürücü, araç ve trafik güvenliğinin iyileştirilmesi sağlanabilir. Devletin akaryakıt takibi için uygulamaya koyduğu Ulusal Taşıt Tanıma Sistemi (UTTS) ile önerilen BAKSİS modeli arasında entegrasyon sağlanarak akaryakıt kullanımı ve elektrikli araçlar için batarya kullanımı takip ve kontrol altına alınabilir.

BAKSİS modelinde kullanıcılar kuruluşların kendi personeli yada üyeleri olarak tanımlanmaktadır. Kullanıcılar her kuruluşun kendi istemci web arayüzünü kullanarak işlemleri gerçekleştirmektedir. Kullanıcıların üyelik yönetimi, sertifika geçerlilik süreleri ve yetkileri kuruluşların sorumluluğuna bırakılmıştır. Tez kapsamında kullanıcıların üyelik ve kimlik yönetimi ele alınmamıştır. Kimlik yönetiminin ve üyelik işlemlerinin e-devlet üzerinden yapılması güvenliğin ve güvenilirliğin sağlanması için oldukça önemlidir. Önerilen BAKSİS sisteminin en zayıf noktası kullanıcıların kimlik ve şifre bilgilerinin çalınması olacaktır. Bu nedenle istemci web arayüzlerine giriş işlemlerinin e-devlet üzerinden çok faktörlü doğrulama yöntemleriyle yapılması için gerekli entegrasyonların yapılmasına ihtiyaç vardır.

Modelimiz araç sahiplerinin doğrudan sistemle bağlantı kurmasına izin vermemektedir. Sistemde yer alan kullanıcılar çoğunlukla kuruluşların üyeleri ve personelidir. Araç sahiplerinin de kullanıcı olarak tanımlanması için EGM trafik başkanlığının kontrolünde ve yönetiminde e-devlet ile entegrasyonun yapılması, kimlik yönetimi ve araçla ilgili gerekli tüm bilgilere erişimin e-devlet üzerinden sağlanması daha güvenli olacaktır.

7. KAYNAKLAR

[1] https://data.tuik.gov.tr/Bulten/Index?p=Motorlu-Kara-Tasitlari-Mayis-2025-54041 (Erişim Tarihi: 22.06.2025)

[2] Benarous, L., Kadri, B., Bouridane, A., & Benkhelifa, E. (2022). Blockchain-based forgery resilient vehicle registration system. Transactions on Emerging Telecommunications Technologies, 33(10), e4237. https://doi.org/10.1002/ett.4237

[3] https://portal.tnb.org.tr/Artes/Sayfalar/TescilGerekliEvrak.aspx (Erişim Tarihi: 24.04.2025)

[4] Tektaş, M., Korkmaz, K., & Erdal, H. (2016). Akıllı Ulaşım Sistemlerinin Geleceği (Ekonomik ve Çevresel Faydaları). Balkan Sosyal Bilimler Dergisi, 561-577.

[5] Mahmutoğlu, A., & Çukurçayır, M. A. (2012). Trafik Sorununa Bir Çözüm Önerisi: Trafik İzleme Başkanlığı. Sayıştay Dergisi, 86,

[6] Elisa, N., Yang, L., Chao, F., & Cao, Y. (2023). A framework of blockchain-based secure and privacy-preserving E-government system. Wireless Networks, 29(3), 1005-1015. https://doi.org/10.1007/s11276-018-1883-0

[7] Baumann, J., Zavolokina, L., & Schwabe, G. (2021). Dealers of Peaches and Lemons: How Can Used Car Dealers Use Trusted Car Data to create value? Proceedings of the 54th Hawaii International Conference on System Sciences. http://hdl.handle.net/10125/71278

[8] Zafar, S., Hassan, S. F. U., Mohammad, A., Al-Ahmadi, A. A., & Ullah, N. (2022). Implementation of a Distributed Framework for Permissioned Blockchain-Based Secure Automotive Supply Chain Management. Sensors, 22(19), 7367. https://doi.org/10.3390/s22197367

[9] Akçi, Y. (2016). İkinci El Ototmobil: Tüketici Bakışıyla. Adıyaman Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 1(1), 329. https://doi.org/10.14520/adyusbd.68749

[10] Dayi, F., & Hasanoğlu, T. (2023). Türkiye'de İkinci El Otomobil Fiyatlarını Etkileyen Faktörlerin İncelenmesi. Süleyman Demirel Üniversitesi Vizyoner Dergisi, 14(38), 436-457. https://doi.org/10.21076/vizyoner.1193474

[11] Brousmiche, K. L., Heno, T., Poulain, C., Dalmieres, A., & Ben Hamida, E. (2018). Digitizing, Securing and Sharing Vehicles Life-cycle over a Consortium Blockchain: Lessons Learned. 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 1-5. https://doi.org/10.1109/NTMS.2018.8328733

[12] Das, D., Banerjee, S., Ghosh, U., Biswas, U., & Bashir, A. K. (2021). A decentralized vehicle anti-theft system using Blockchain and smart contracts. Peer-to-Peer Networking and Applications, 14(5), 2775-2788. https://doi.org/10.1007/s12083-021-01097-3

[13] Abbade, L. R., Ribeiro, F. M., Da Silva, M. H., Morais, A. F., De Morais, E. S., Lopes, E. M., ... & Rodrigues, J. J. (2020). Blockchain applied to vehicular odometers. IEEE Network, 34(1), 62-68. doi: 10.1109/MNET.001.1900162.

[14] Chanson, M., Bogner, A., Wortmann, F., & Fleisch, E. (2017). Blockchain as a privacy enabler: An odometer fraud prevention system. Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers, 13-16. https://doi.org/10.1145/3123024.3123078

[15] Holler, M., Barth, L., & Fuchs, R. (2019). Trustworthy Product Lifecycle Management Using Blockchain Technology—Experience from the Automotive Ecosystem. Içinde J. Stark (Ed.), Product Lifecycle Management (Volume 4): The Case Studies (ss. 13-19). Springer International Publishing. https://doi.org/10.1007/978-3-030-16134-7_2

[16] Schwabe, G. (2019). The role of public agencies in blockchain consortia: Learning from the Cardossier. Information Polity, 24(4), 437-451. https://doi.org/10.3233/IP-190147

[17] Alessandria, M. L., & Vizzarri, A. (2021). Self-Sovereign Identity and Blockchain applications for the automotive sector. 2021 AEIT International Conference on Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), 1-6. https://doi.org/10.23919/AEITAUTOMOTIVE52815.2021.9662861

[18] Roriz, R., & Pereira, J. L. (2019). Avoiding Insurance Fraud: A Blockchain-based Solution for the Vehicle Sector. Procedia Computer Science, 164, 211-218. https://doi.org/10.1016/j.procs.2019.12.174

[19] Xu, W., Song, Y., Liang, H., Sun, L., & Xie, Y. (2024). A Blockchain-Based Identity Control Scheme for Cross-Organizational Data Sharing. 2024 IEEE 10th Conference on Big Data Security on Cloud, 156-160. https://doi.org/10.1109/BigDataSecurity62737.2024.00035

[20] Özyürek, H. (2021). Blockchain Teknolojisinin Mevcut ve Muhtemel Kullanım Alanları. Anadolu Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 22(4), 31-50. https://doi.org/10.53443/anadoluibfd.988748

[21] Demir, M., Turetken, O., & Ferworn, A. (2019). Blockchain Based Transparent Vehicle Insurance Management. 2019 Sixth International Conference on Software Defined Systems (SDS), 213-220. https://doi.org/10.1109/SDS.2019.8768669

[22] Yadav, A. S., Charles, V., Pandey, D. K., Gupta, S., Gherman, T., & Kushwaha, D. S. (2023). Blockchain-based secure privacy-preserving vehicle accident and insurance registration. Expert Systems with Applications, 230. https://doi.org/10.1016/j.eswa.2023.120651

[23] Bader, L., Burger, J. C., Matzutt, R., & Wehrle, K. (2018). Smart Contract-Based Car Insurance Policies. 2018 IEEE Globecom Workshops (GC Wkshps), 1-7. https://doi.org/10.1109/GLOCOMW.2018.8644136

[24] Oham, C., Jurdak, R., Kanhere, S. S., Dorri, A., & Jha, S. (2018). B-FICA: BlockChain based Framework for Auto-Insurance Claim and Adjudication. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data, 1171-1180. https://doi.org/10.1109/Cybermatics_2018.2018.00210

[25] Hombalimath, A., Mangla, N., & Balodi, A. (2023). Designing A Permissioned Blockchain Network For The Insurance Claim Process Using Hyperledger Fabric And Composer. Informatica, 47(3). https://doi.org/10.31449/inf.v47i3.4158.

[26] Xiao, Z., Li, Z., Yang, Y., Chen, P., Liu, R. W., Jing, W., Pyrloh, Y., Sotthiwat, E., & Goh, R. S. M. (2020). Blockchain and IoT for Insurance: A Case Study and Cyberinfrastructure Solution on Fine-Grained Transportation Insurance. IEEE Transactions on Computational Social Systems, 7(6), 1409-1422. https://doi.org/10.1109/TCSS.2020.3034106.

[27] Nizamuddin, N., & Abugabah, A. (2021). Blockchain for automotive: An insight towards the IPFS blockchain-based auto insurance sector. International Journal of Electrical and Computer Engineering (IJECE), 11(3), 2443-2456. https://doi.org/10.11591/ijece.v11i3.

[28] Vo, H. T., Mehedy, L., Mohania, M., & Abebe, E. (2017). Blockchain-based Data Management and Analytics for Micro-insurance Applications. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2539-2542. https://doi.org/10.1145/3132847.3133172.

[29] Qi, H., Wan, Z., Guan, Z., & Cheng, X. (2021). Scalable Decentralized Privacy-Preserving Usage-Based Insurance for Vehicles. IEEE Internet of Things Journal, 8(6), 4472-4484. https://doi.org/10.1109/JIOT.2020.3028014.

[30] Tanrıverdi, M., Uysal, M., Üstündağ, M. T., & Ayaz, Z. (2021). Araç Cüzdanı: Motorlu Araçların Teknik Servis ve Bakım Kayıtlarının Blokzinciri Üzerinde Yönetilmesi. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 9(4), 1358-1373. https://doi.org/10.29130/dubited.904757.

[31] Jiang, Y.T., & Sun, H.-M. (2021). A Blockchain-Based Vehicle Condition Recording System for Second-Hand Vehicle Market. Wireless Communications and Mobile Computing, 2021(1), 6623251. https://doi.org/10.1155/2021/6623251.

[32] Abhale, A., Pathare, A., Indapure, A., Wadekar, J., & Shelke, Dr. P. (2023). Blockchain Based used Vehicle Tracking System. International Journal for Research in Applied Science and Engineering Tech., 11(5), 6986-6994. https://doi.org/10.22214/ijraset.2023.53333

[33] Butera, A., Gatteschi, V., Pratticò, F. G., Novaro, D., & Vianello, D. (2023). Blockchain and NFTs-Based Trades of Second-Hand Vehicles. IEEE Access, 11, 57598-57615. https://doi.org/10.1109/ACCESS.2023.3284676.

[34] Zavolokina, L., Schlegel, M., & Schwabe, G. (2021). How can we reduce information asymmetries and enhance trust in 'The Market for Lemons'? Information Systems and E-Business Management, 19(3), 883-908. https://doi.org/10.1007/s10257-020-00466-4.

[35] Akerlof, G. A. (1978). The market for "lemons": Quality uncertainty and the market mechanism. Uncertainty in economics - Elsevier, 235-251.

[36] Bauer, I., Zavolokina, L., & Schwabe, G. (2020). Is there a market for trusted car data? Electronic Markets, 30(2), 211-225. https://doi.org/10.1007/s12525-019-00368-5

[37] Barreto, J. L. S., Espinoza, R. A. B., & Dávila, G. A. (2022). Blockchain-Based System to Ensure the Integrity of Used Vehicle Purchase Transactions: Under Researchers' Perspective. Içinde R. Pereira, I. Bianchi, & Á. Rocha (Ed.), Digital Technologies and Transformation in Business, Industry and Organizations (C. 210, ss. 121-141). Springer International Publishing. https://doi.org/10.1007/978-3-031-07626-8_6

[38] López-Pimentel, J. C., Morales-Rosales, L. A., Algredo-Badillo, I., & Del-Valle-Soto, C. (2023). NFT-Vehicle: A Blockchain-Based Tokenization Architecture to Register Transactions over a Vehicle's Life Cycle. Mathematics, 11(13), 2801. https://doi.org/10.3390/math11132801

[39] Syed, T. A., Siddique, M. S., Nadeem, A., Alzahrani, A., Jan, S., & Khattak, M. A. K. (2020). A Novel Blockchain-Based Framework for Vehicle Life Cycle Tracking: An End-to-End Solution. IEEE Access, 8, 111042-111063. https://doi.org/10.1109/ACCESS.2020.3002170

[40] Masoud, M. Z., Jaradat, Y., Jannoud, I., & Zaidan, D. (2019). CarChain: A Novel Public Blockchain-based Used Motor Vehicle History Reporting System. 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), 683-688. https://doi.org/10.1109/JEEIT.2019.8717495.

[41] Florez, L., & Correal, D. (2023). Securing a National Driver and Vehicle Registration System with Blockchain. 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C), 239-245. https://doi.org/10.1109/ICSA-C57050.2023.00058

[42] Distefano, S., Giacomo, A. D., & Mazzara, M. (2021). Trustworthiness for Transportation Ecosystems: The Blockchain Vehicle Information System. IEEE Transactions on Intelligent Transportation Systems, 22(4), 2013-2022. https://doi.org/10.1109/TITS.2021.3054996.

[43] Barolli, E., & Zeneli, M. (2022). A Proposed Blockchain-Based Solution for a Data-Driven Vehicle Lifecycle Management. Sixth International Scientific Conference ITEMA Recent Advances in Information Technology, Tourism, Economics, Management and Agriculture 9-15. https://doi.org/10.31410/ITEMA.2022.9.

[44] https://www.fiaregion1.com/wp-content/uploads/2017/05/joint_appeal_against_odometer_manipulation_final.pdf (Erişim Tarihi: 14.05.2024)

[45] Borkowski, P. (2019). Reducing Odometer Fraud in the EU Second-Hand Passenger Car Market Through Technical Solution. İçinde G. Sierpiński (Ed.), Integration as Solution for Advanced Smart Urban Transport Systems (C. 844, ss. 184-194). Springer International Publishing. https://doi.org/10.1007/978-3-319-99477-2_17.

[46] Samue, C. N., Severine, G., David, B., Verdier, F., & Patricia, G.-O. (2020). Automotive Data Certification Problem: A View on Effective Blockchain Architectural Solutions. 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, 0167-0173. https://doi.org/10.1109/IEMCON51383.2020.9284886.

[47] Kardaş, S. (2019). Blokzincir Teknolojisi: Uzlaşma Protokolleri. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(2). https://doi.org/10.24012/dumf.426805.

[48] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[49] https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (Erişim Tarihi: 04.12.2024)

[50] N. Szabo,(1994). Smart contracts.

[51] Mendi, A. F. (2021). Blokzincir Uygulamaları ve Gelecek Öngörüleri. GSI Journals Serie C: Advancements in Information Sciences and Technologies, 4(1), 76-88.

[52] https://tr.linkedin.com/pulse/merkeziyetsizlik-decentralization-nihat-alparslan (Erişim Tarihi: 11.05.2025)

[53] Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., & Wang, J. (2018). Untangling Blockchain: A Data Processing View of Blockchain Systems. IEEE Transactions on Knowledge and Data Engineering, 30(7), 1366-1385. https://doi.org/10.1109/TKDE.2017.2781227.

[54] Haber, S., & Stornetta, W. S. (1991). How to Time-Stamp a Digital Document. Içinde A. J. Menezes & S. A. Vanstone (Ed.), Advances in Cryptology-CRYPTO' 90 (ss. 437-455). Springer. https://doi.org/10.1007/3-540-38424-3_32.

[55] Anderson, R., & Kuhn, M. (1996). Tamper resistance-a cautionary note. Proceedings of the second Usenix workshop on electronic commerce, 1-11.

[56] Schneier, B., & Kelsey, J. (1998). Cryptographic support for secure logs on untrusted machines. USENIX Security Symposium San Antonio, 53-62.

[57] Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. 2017 IEEE International Congress on Big Data, 557-564. https://doi.org/10.1109/BigDataCongress.2017.85.

[58] Mangrulkar, R. S., & Chavan, P. V. (2024). Blockchain Essentials, Core Concepts and Implementations. Apress. https://link.springer.com/book/10.1007/978-1-4842-9975-3

[59] Sezgin, S. (2024). Kripto Madenciliğinin Çevre Üzerindeki Etkileri. Akademik İzdüşüm Dergisi, 9(2), 285-312.

[60] Bamakan, S. M. H., Motavali, A., & Babaei Bondarti, A. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. Expert Systems with Applications, 154, 113385. https://doi.org/10.1016/j.eswa.2020.113385.

[61] Merkle, R. C. (1979). Secrecy, Authentication, and Public Key Systems. Doktora Tezi. Stanford University, https://www.proquest.com/dissertations-theses/secrecy-authentication-public-key-systems/docview/302984000/se-2?accountid=15410

[62] https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains (Erişim Tarihi: 26.04.2025)

[63] Yıldırım, H. (2018). Açık ve uzaktan öğrenmede blokzincir teknolojisinin kullanımı. Açıköğretim Uygulamaları Ve Araştırmaları Dergisi, 4(3), 142-153.

[64] Cachin, C. (2016). Architecture of the hyperledger blockchain fabric. In Workshop on distributed cryptocurrencies and consensus ledgers. 310(4). 1-4.

[65] Onur, S., Tektaş, M., Özer, İ., Çelik, U., vd. (2025). Motorlu kara taşıtlarında güvenli ve güvenilir veri yönetim modeli. Akıllı Ulaşım Sistemleri Ve Uygulamaları Dergisi, 8(1), 285-299. https://doi.org/10.51513/jitsa.1644011.

[66] Vukolic, M. (2016). The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. İçinde J. Camenisch & D. Kesdoğan (Ed.), Open Problems in Network Security (C. 9591, ss. 112-125). Springer International Publishing. https://doi.org/10.1007/978-3-319-39028-4_9.

[67] Helal, M. M., & Asghar, M. R. (2019). Towards preserving privacy and security in blockchain. Essentials of Blockchain Technology -Chapman and Hall/CRC. 99-120.

[68] https://hyperledger-fabric.readthedocs.io/en/latest/ (Erişim Tarihi: 22.12.2024)

[69] Ali, H., Ahmad, J., Jaroucheh, Z., Papadopoulos, P., Pitropakis, N., Lo, O., Abramson, W., & Buchanan, W. J. (2022). Trusted Threat Intelligence Sharing in Practice and Performance Benchmarking through the Hyperledger Fabric Platform. Entropy, 24(10), 1379. https://doi.org/10.3390/e24101379.

[70] Daniel, E., & Tschorsch, F. (2022). IPFS and Friends: A Qualitative Comparison of Next Generation Peer-to-Peer Data Networks. IEEE Communications Surveys & Tutorials, 24(1), 31-52. https://doi.org/10.1109/COMST.2022.3143147.

[71] Steichen, M., Fiz, B., Norvill, R., Shbair, W., & State, R. (2018). Blockchain-Based, Decentralized Access Control for IPFS. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 1499-1506. https://doi.org/10.1109/Cybermatics_2018.2018.00253.

[72] https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatis.html (Erişim Tarihi: 22.12.2024)

[73] https://hyperledger-caliper.github.io/caliper/0.6.0/ (Erişim Tarihi: 29.04.2025)

[74] Hang, L., & Kim, D.-H. (2021). Optimal blockchain network construction methodology based on analysis of configurable components for enhancing Hyperledger Fabric performance. Blockchain: Research and Applications, 2(1), 100009. https://doi.org/10.1016/j.bcra.2021.100009.

[75] https://hyperledger-caliper.github.io/caliper/0.6.0/getting-started/architecture/ (Erişim Tarihi: 21.06.2025)

[76] http://scet.berkeley.edu/wp-content/uploads/AIR-2016-Blokzincir.pdf (Erişim Tarihi: 21.06.2025)