T.C.

FENERBAHÇE ÜNİVERSİTESİ

MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLOKZİNCİRİ TEKNOLOJİSİNİN FİNANS DÜNYASI ÜZERİNDEKİ ETKİSİ VE UYGULAMASI

BİTİRME PROJESİ

Hazırlayan

CÜNEYT BALCI

190301019

DR. OSMAN SELVİ

EKİM – 2022

KABUL VE ONAY FORMU

Üniversite : Fenerbahçe Üniversitesi

Fakülte : Mühendislik ve Mimarlık Fakültesi

Bölümü : Bilgisayar Mühendisliği

Öğrenci Numarası : 190301019

Öğrenci Adı Soyadı : Cüneyt BALCI

Proje Başlığı : Blokzincir Teknolojisinin Finans Dünyası Üzerindeki Etkisi ve Uygulaması

Sınav Yeri :

Sınav Tarihi :

Jüri Görüşleri:

Bitirme projesi tarafımızdan okunmuş, kapsam ve kalite yönünden kabul edilmiştir.

Jüri Üyesi Ünvan Ad SOYAD	**Görüş (KABUL / RET)**	**İmza**

Yukarıdaki jüri kararı Bölüm Başkanlığının ...../...../..... tarih ve ..... sayılı kararı ile onaylanmıştır.

Unvan Ad ve Soyad

Bölüm Başkanı

AKADEMİK DÜRÜSTLÜK BEYANI

Bu projenin planlamasından yazımına kadar olan süreç içerisinde hiçbir şekilde bilimsel ahlak ve değerlere uygun olmayan davranışlarımın olmadığını, projenin içeriğinin yazım ilkelerine uygun bir biçimde hazırladığımı, proje ile elde edilmeyen bilgi ve yorumlara kılavuzda belirtilen kurallar ışığında kaynak göstererek projeyi tamamladığımı ve ayrıca projeyi herhangi bir patent ve telif hakkını ihlal etmeyecek bir biçimde hazırladığımı beyan ederim.

01/ 06 /2023

Cüneyt Balcı

TEŞEKKÜR

Bu tezin hazırlanmasında bana rehberlik eden ve değerli yorumlarıyla katkıda bulunan danışmanım Dr. Osman Selvi'ye sonsuz teşekkürlerimi sunarım. Tez sürecinde bana maddi ve manevi destek veren, sabrını ve sevgisini esirgemeyen aileme, özellikle annem Şennure Balcı'ya ve babam Akın Balcı'ya teşekkür ederim. Ayrıca, tez çalışmamda bana yardımcı olan, motivasyonumu artıran kız arkadaşım İrem Ceylan'a, arkadaşlıklarını paylaşan Samet Dağdevir'e ve Umut Talha Çelik'e şükranlarımı sunarım. Bu tez, bana verilen bu fırsatı değerlendirmeme ve akademik hayatımda bir adım daha ilerlememe olanak sağlayan Yükseköğretim Kurulu Başkanlığı'nın desteğiyle gerçekleştirilmiştir. Bu desteği sağlayan kuruma ve çalışanlarına teşekkür ederim.

CÜNEYT BALCI

İstanbul, 2022

ÖZET

Blokzincir, finans dünyasında yeni fırsatlar yaratan bir teknolojidir. Bu teknoloji, güvenli, şeffaf ve merkezi olmayan bir veri tabanı sağlayarak, finansal işlemleri kolaylaştırabilir ve maliyetlerini azaltabilir. Bu çalışmada, blokzincir teknolojisinin finans dünyasına etkisi incelenmiş ve Türkiye'de ekonomik sorunlar yaşandığı için artık bir problem haline gelen konut kiralama durumları için bir uygulama önerisi sunulmuştur. Solidity programlama dili kullanılarak akıllı sözleşme şeklinde bir kira kontratı geliştirilmiştir. Bu kontrat, Ethereum ağı üzerinde çalışmakta ve kiraların blokzincir ağında bulunan cüzdanlar tarafından ödenmesini sağlamaktadır. Ayrıca, kira ödenmemesi, fazla zam yapılması, sözleşmenin haksızca feshedilmesi gibi durumlara karşı akıllı sözleşmenin otomatik olarak yapılması gereken işlemleri uygulamasını içermektedir. Bu çalışma, blokzincir teknolojisinin finansal hizmetlerde nasıl kullanılabileceğine ve hangi avantajları sağlayabileceğine dair bir örnek teşkil etmektedir.

ANAHTAR KELİMELER:blokzincir, akıllı sözleşme, merkeziyetsiz veritabanı, ethereum, Solidity

ABSTRACT

Blockchain is a technology that creates new opportunities in the financial world. This technology can facilitate financial transactions and reduce their costs by providing a secure, transparent and decentralized database. In this study, the impact of blockchain technology on the financial world is analyzed and an application proposal is presented for the leasing sector, a sector that is experiencing economic problems in Turkey. A smart contract lease contract was developed using the Solidity programming language. This contract runs on the Ethereum network and ensures that rents are paid by wallets on the blockchain network. It also includes the automatic sanctioning of the smart contract against non-payment of rent, over-increase, unfair termination of the contract. This study is an example of how blockchain technology can be used in financial services and what advantages it can provide.

KEYWORDS: blockchain,smart contract,finance,decentralized database,ethereum,solidity

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ

Sayfa

SİMGELER VE KISALTMALAR LİSTESİ

Simgeler

& **:** Ve

Kısaltmalar

dApp **:** Decentralized Application
DEX **:** Decentralized Exchange
DeFi **:** Decentralized Finance
PoW **:** Proof of Work
PoS **:** Proof of Stake
DPoS **:** Delegated Proof of Stake
PoA **:** Proof of Authority
ETH **:** Ether
EVM **:** Ethereum Virtual Machine
NFT **:** Non-Fungible Token
IDE **:** Integrated Development Environment
JSON **:** JavaScript Object Notation
ABI **:** Application Binary Interface

1. GİRİŞ

Dünyamızda teknoloji ve finans sisteminin gelişimi, birçok alanda hızla ilerlemektedir. Özellikle, finansal teknolojiler alanında da blok zincir teknolojisi önemli bir gelişme kaydetmektedir.

Blokzincir bir veri yapısıdır ve merkezi olmayan bir ağ üzerinde birden fazla kişi tarafından paylaşılan bir veri kaydının tutulmasını sağlar. Bu veri kaydı, birçok bilgisayar tarafından onaylandıktan sonra değiştirilemez ve silinemez. Bu özellikleri nedeniyle, blokzincir özellikle para transferi işlemlerinde güvenli ve güvenilir bir seçenek olarak kullanılmaktadır. Blokzincir teknolojisi, çağımızda birçok alanda etkili bir şekilde kullanılmaktadır. Özellikle de finansal sektörde büyük bir etkiye sahiptir. Aynı zamanda veri güvenliği ve güvenliği sağlamak için de kullanılmaktadır. Örneğin, sağlık verileri ve kişisel bilgiler gibi özel bilgilerin saklanmasında blokzincir kullanılabilir. Ayrıca üretim ve lojistik alanlarında, ürünlerin kaynağını izlemek ve doğrulamak için kullanılmaktadır. Ayrıca, çevre ve enerji alanlarında, çevre dostu enerji üretimini ve takip etmeyi kolaylaştırmak için kullanılmaktadır. Blok zincir teknolojisinin en büyük kullanım alanı olan kripto paralar blok zincir teknolojisinin daha da tanınmasını ve yaygınlaşmasını sağlamıştır. Kripto paralar, merkezi bir otorite tarafından kontrol edilmeyen, güvenlik için kriptografi kullanan sanal para birimleridir. Kripto paralar sayesinde, para transferi işlemleri daha hızlı, daha güvenli ve daha düşük maliyetli hale gelebilir.

Ülkemizde son yıllarda konut fiyatları ve kira artışları önemli bir sorun haline gelmiştir. Kira sözleşmesi, kiraya verenin bir taşınmazın kullanımı veya kullanımla birlikte ondan yararlanılmasını kiracıya bıraktığı, kiracının da buna karşılık bir bedel ödemekle yükümlü olduğu bir sözleşmedir.(ANTALYA & ÇAVDAR, 2020) Konut fiyatlarındaki artışın nedenleri arasında arz-talep dengesizliği, enflasyon, kredi faizleri, döviz kurları, arsa maliyetleri ve spekülasyonlar sayılabilir. Kira artışlarındaki artışın nedenleri arasında ise konut fiyatlarındaki artışa paralel olarak ev sahiplerinin beklentileri, kiracıların talebi, enflasyon ve yasal düzenlemeler sayılabilir. Enflasyon oranındaki artış, insanların alım gücünü düşürmüştür. Bu da insanların kira ödemelerini karşılamakta zorlanmalarına neden olmuştur. Bu sebepler ötürü kiracılar

ödemelerini düzenli yapamamaya başlamış, ev sahipleri haksızca kiracıları evden çıkarmaya başlamış ve taraflar arası anlaşmazlıklar git gide artmaya başlamıştır.

Bu çalışmada da bu anlaşmazlıkları ve sorunları gidermek için blok zincir teknolojisi kullanılarak bir akıllı kontrat geliştirilip, genel geçer kullanımda olan kira kontratı bu akıllı kontratın üzerine entegre edilerek bir prototip uygulama geliştirilmiştir.

2. LİTERATÜR TARAMASI

“Blokzincir teknolojisi kullanıcı verilerini, dijital varlıkları yönetmemizi sağlayan dağıtık defter yapısında bir mimaridir. Blokzincir mimarisine dayalı bir yapı olarak önerilen kripto para sistemleri ile sanal ekonomi ve finans sistemleri giderek gelişmektedir.”(Parlar & Prof, 2022) “Blockchain (blokzincir) ilk defa 2009 yılında ortaya çıkan bir alt yapı teknolojisi olsa da finans dünyası tarafından keşfedilip araştırılmaya başlanması 2014 yılının ilk aylarını bulmuştur.”(YILMAZ TÜRKMEN & ERÖZEL DURBİLMEZ, 2019) “Bu sistem güvenli ve tutarlı bir veritabanı sistemi iddiasıyla başta finans sektörü olmak üzere birçok alanda yaygınlaşmaya başlamıştır. Bu durum geleneksel veritabanı yaklaşımları ile Blok zinciri sisteminin karşılaştırılarak başta güvenlik olmak üzere avantaj ve dezavantajlarının değerlendirilmesi ve ortaya konmasını zorunlu kılmaktadır.”(ULUYOL & ÜNAL, 2020). Ayrıca yine aynı sistemde işleyen ve Vitalik Buterin tarafından geliştirilen ethereum ise bir blokzincir ağıdır ve Bitcoin'in yanı sıra en popüler dijital para birimi olarak kabul edilir. Ethereum, Bitcoin gibi bir değişim aracı olarak kullanılabilir, ancak aynı zamanda diğer blokzincir uygulamaları için de bir platform sağlar. Ethereum, "smart contract"leri ve decentralized uygulamaları (dApps) oluşturmak için kullanılabilir. “Ether ise bu platform tarafından üretilen kripto para birimine verilen addır, tıpkı Blokzincir ve Bitcoin gibi. Etherum’un Solidity yazılım dilini kullanan, farklı sertifikalarla geliştirilmiş yapısı Ethereum üzerinde Smart Contracts adı verilen ve yazılımdan oluşan kod parçalarının yerleştirilmesine ve çalışmasına olanak sağlamıştır.”(TEVETOĞLU, 2021)

“Blokzincir teknolojisinin sahip olduğu potansiyelin bir parçası olan akıllı sözleşmel er özel teşebbüs tarafından tedarik zinciri yönetimlerinde denenmeye başlanmıştır.”(DURDU & GÖKÇE, 2022) Akıllı sözleşmeler, blok zincir teknolojisi kullanarak otomatikleştirilmiş veya gerçekleştirilmiş işlemleri ifade eden kod parçacıklardır. Bu sözleşmeler, belirli koşullar gerçekleştiğinde otomatik olarak yürürlüğe girer ve yürütülür. Örneğin, bir alıcı ve satıcı arasında bir ticaret yaparken, sözleşme tarafların tarafından belirlenen koşullar gerçekleştiğinde, para otomatik olarak alıcıya ve ürün otomatik olarak satıcıya gönderilir. “Smart Contracts kavramı ilk defa 1994 yılında bir hukuk profesörü olan Nick Szabo’nun yaptığı çalışmalar neticesinde ortaya çıkmış ve

zaman içinde şekillenerek bugünkü halini almıştır."(TEVETOĞLU, 2021) Akıllı sözleşmelerin faydalarından birisi ise otomatik işlemleri yerine getirdiği için daha hızlı olmaları ve hata oranını düşürmeleridir. Ayrıca blokzincir teknolojisi üzerine inşa edildiği için, işlemlerin güvenliği değiştirilmeye ve yanıltılmaya karşı korunmasını sağlar. Bir diğer faydası ise maliyetleri azaltmasıdır. Örneğin, bir banka, bir kredi işlemi gerçekleştirdiğinde otomatik olarak kredi skoru kontrol edebilir veya bir ödeme gerçekleştiğinde otomatik olarak fonların transferi gerçekleştirebilir. Akıllı sözleşmeler, tedarik zincirinde de birçok farklı noktada kullanılabilir. Tedarik zinciri mali işlemleri için, akıllı sözleşmeler, ödemelerin otomatik olarak gerçekleştirilmesini sağlar. Güvenliği sağlamak için, akıllı sözleşmeler, malın nerede ne zaman üretildiğini, nerede ne zaman teslim edildiğini ve nerede ne zaman alındığını takip edebilir. "Ancak akıllı sözleşmeler üzerinde çalıştıkları blok zincir dışından veri alamamaktadır. Bu problem gerçek yaşam problemlerinin çözümü adına projelerin ortaya çıkışını sınırlandırmaktadır. Örneğin döviz bilgileri ile ilgili bir uygulamada gerekli veri kaynağı blok zincir dışıdır. Tam bu noktada blok zincir ile dış dünya arasında bir köprüye ihtiyaç duyulmaktadır. Blok zincir ulakları blok zincir içi ve dışı verileri arasındaki ilet işim boşluğunu doldurmaktadır."(BEŞTAŞ, 2022)

2.1 BLOKZİNCİR TEKNOLOJİSİ

2.1.1 Blokzincir Nedir?

Blok zincir merkeziyetsiz kayıt defteri olarak tanımlanmaktadır ve birçok bloktan oluşur. Her blok, bir önceki bloğun hash değerini içerir ve ayrıca işlemleri veya diğer bilgileri içerebilir. Bu yapı, güvenli bir şekilde verilerin depolanmasını ve değiştirilmemesini sağlar. Bu veriler arasında Bitcoin gibi kripto paraların işlemleri de yer alabilir.

2.1.2 Blokzincir Yapısı

Blokzincirinin depolama alanları, blokzincirinin türüne ve kullanılan sürücülere bağlı olarak değişir. Blokzincirindeki her blok, belirli bir miktarda veri içerir ve sınırlı bir

depolama kapasitesine sahiptir. Blokzincirindeki veriler, ağa yayılmış bir dijital veritabanında saklanır. Bu veritabanı, merkezi olmayan ve değiştirilemez bir yapıdadır.

2.1.2.1 Bloklar

Blok zincirindeki bloklar, veri işlemlerini depolamak için kullanılır. Her blok, birçok işlemi içerebilir ve her işlem, bir gönderici ve bir alıcı arasında bir miktar para transferini temsil eder. Bu işlemler, blok zincirindeki diğer kullanıcılar tarafından onaylandıktan sonra, blok zincirinde kaydedilir. Bu onaylamalar, diğer kullanıcılar tarafından yapılan "madencilik" işlemleri ile gerçekleştirilir. Bu işlemler, işlemleri içeren bir blok oluşturmak için bilgisayarlar tarafından gerçekleştirilen zor bir matematik problemi çözmeye yönelik çalışmalardır. Bloklar arasındaki bu bağlantılar sayesinde, blok zincirindeki verilerin geriye dönük olarak değiştirilmemesi sağlanır.

2.1.2.2 Hash

Blok zincirinde, her blok bir hash değerine sahiptir. Hash, bir veri girişinden hesaplanan benzersiz bir değerdir. Bu değer, veri girişinin herhangi bir değişikliği halinde değişir. Dolayısıyla, bir bloğun içindeki veriler değiştirilmedikçe, o blok için hesaplanan hash değeri de aynı kalacaktır. Hash değerleri, blok zincirindeki bloklar arasındaki bağlantıları sağlar. Her blok, bir önceki bloktaki hash değerini bulundurur. Bu sayede, blok zincirindeki verilerin geriye dönük olarak değiştirilmemesi sağlanır.

2.1.2.3 Merkle Ağacı

Merkle ağacı, bir veri yapısıdır ve birçok veri bloğunun hash değerlerini içerebilir. Bu veri bloğları, işlemler veya diğer bilgiler olabilir. Merkle ağacının temel özelliği, veri blokları arasındaki bağlantıları sağlamak için kullanılan hash değerleridir.

Merkle ağacının yapısı, her veri bloğunun hash değerini içeren iki veri bloğunun hash değerlerini birleştirerek oluşur. Bu işlem, iki veri bloğunun hash değerlerinin hash

değerini oluşturur. Bu işlem, her veri bloğu için tekrar edilir ve sonunda, tüm veri bloğlarının hash değerlerini içeren tek bir hash değeri elde edilir. Bu değere de Merkle ağacının kökü denir.

Merkle ağaçları, blok zincirleri gibi kripto paraların işlemlerini depolamak için kullanılabilir. Özellikle çok sayıda işlem içeren bloklar için, Merkle ağacı kullanımı verimlilik sağlar. Ayrıca, Merkle ağaçları, veri güvenliğini sağlamak için kullanılabilir çünkü her veri bloğunun değiştirilmemesi durumunda, Merkle ağacının kök düğümünde oluşan hash değeri de değişmez.

2.1.2.4 Dağıtık Defter

Blok zincirinde, dağıtık defter, bir veri yapısıdır ve birçok kullanıcı tarafından paylaşılan verileri içerebilir. Bu veriler arasında işlemler, bireysel hesap bilgileri ve diğer bilgiler de yer alabilir. Blokzincirde saklanan bu veriler her kullanıcı tarafından kullanılabilir. Dağıtık defter, herhangi bir merkezi otorite olmaksızın işlemleri gerçekleştirmek için kullanılabilir. Her kullanıcı, blok zincirinde işlem yapabilir ve bu işlemler, diğer kullanıcılar tarafından onaylandıktan sonra blok zincirinde kaydedilir. Bu onaylamalar, diğer kullanıcılar tarafından yapılan "madencilik" işlemleri ile gerçekleştirilir. Bu işlemler, işlemleri içeren bir blok oluşturmak için bilgisayarlar tarafından gerçekleştirilen zor bir matematik problemi çözmeye yönelik çalışmalardır.

Dağıtık defter teknolojisi, blok zincir teknolojisi ile birlikte kullanılarak kullanılabilir. Blok zincir, verilerin geriye dönük olarak değiştirilmemesini sağlar ve dağıtık defter ise verilerin her kullanıcı tarafından paylaşılmasını sağlar. Bu teknolojilerin birlikte kullanımı, güvenli ve merkezi olmayan bir sistem oluşmasını sağlar.

2.2 Merkeziyetsizlik

Blokzincir teknolojisi merkeziyetsizlik ilkesine dayanmaktadır. Bu ilke blokzincirde bulunan verilerin, merkezi bir otoriteye bağlı olmadan tüm kullanıcılar tarafından paylaşılması ve doğrulanması gerektiği anlamına gelmektedir. Yani blokzincirde

merkeziyetsizlik, herhangi bir kontrol mekanizması olmadan kullanıcılar arasında dağıtık ve eşit bir şekilde sistemin oluşturulmasıdır.

2.3 Güvenlik

Blok zinciri, verileri güvence altına almak için gelişmiş şifreleme teknikleri kullanır. Blok zincirindeki her işlem, sadece sahibinin erişimine sahip olduğu özel anahtarla imzalanır. Bu, başka kimse tarafından veriye müdahale edilmesini veya manipüle edilmesini engeller. Bu, verilerin değişmez ve müdahale edilemez olmasını sağlar. İşlemi değiştirmek veya silmek için ağın mutabakatına ihtiyaç duyulur, bu da neredeyse imkansızdır.

2.4 Tamper-Proof

"Tamper proof" terimi, genellikle bir sistem veya teknolojinin yetkisiz veya hileli değişikliklere karşı dayanıklı olduğunu belirtmek için kullanılır. Blokzincir teknolojisi, tamper proof özelliğini benzersiz veri yapısı sayesinde sağlar.

Blokzincir, bir dizi işlemi bir arada gruplandıran ve bu grupları "bloklar" olarak adlandıran bir veri yapısıdır. Her blok, önceki bloğun bir özeti olan bir hash değeri içerir. Bu bloklar birbirine bağlanarak bir "zincir" oluşturur ve bu zincir, tüm işlem geçmişini içerir.

Bu yapı, blokzinciri tamper proof yapar çünkü herhangi bir bloktaki bir işlemi değiştirmeye çalışmak, o bloğun hash değerini değiştirecektir. Bu değişiklik, sonraki bloğun referansını geçersiz kılacak ve sonraki tüm blokların hash değerlerinin de değişmesine neden olacaktır. Bu nedenle, bir işlemi değiştirmeye çalışmak, tüm blokzincirin yeniden yazılmasını gerektirir.

2.5 Açıklık ve Şeffaflık

Açıklık ve şeffaflık blokzincir teknolojisinin en temel unsurlarındandır. Bu, blokzincir üzerindeki tüm işlemlerin şeffaf olduğu ve görmek isteyen herkes tarafından

incelenebileceği anlamına gelir. Şeffaflık blokzincirde kaydedilen işlemlerin herkes tarafından görüntülenebilme ve doğrulanabilme yeteneğine referans eder. Ayrıca, blokzincir teknolojisinin şeffaflığı, tüm işlemlerin güvenli ve değiştirilemez bir defterde kaydedilmesine yardımcı olur ve dolandırıcılık faaliyetlerini önlemeye yardımcı olur. Bu, bir işlemin blokzincirde kaydedildikten sonra değiştirilemeyeceği veya silinemeyeceği anlamına gelir ve ek bir güvenlik ve hesap verebilirlik katmanı sağlar.

2.6 Ağ Bütünlüğü

Blokzincir teknolojisinin ağ bütünlüğünü sağlamada temel yollarından biri, çoğu blokzincir ağındaki düğümlerin, işlemleri blok zincirine eklenmeden önce doğrulaması ve onaylamasıdır. Bu işlem, ağdaki tüm işlemlerin meşru olduğundan ve defterin zamanla doğru ve tutarlı olmasını sağlar. Ek olarak, birçok blokzincir ağı, blok zincirindeki verileri güvence altına almak için şifreleme tekniklerinden yararlanır. Genel olarak, ağ bütünlüğü, blok zinciri teknolojisinde depolanan verilerin doğru, tutarlı ve manipülasyon veya değiştirme girişimlerine karşı dirençli olmasını sağladığı için kritik bir özelliktir.

2.7 Gücün Dağıtımı ve Konsensüs Mekanizmaları

Bir blok zinciri ağındaki gücün dağıtımı genellikle bir fikir birliği mekanizması aracılığıyla gerçekleştirilir. Fikir birliği mekanizması, ağdaki düğümlerin blok zincirinin durumu konusunda nasıl anlaşacaklarını belirleyen bir dizi kuraldır.

2.7.1 Proof of Work (PoW)

Proof of Work (PoW), blockchain ağlarında konsensus sağlamak için kullanılan bir algoritma türüdür. Bu algoritma, madencilik olarak adlandırılan işlemleri gerçekleştiren bilgisayarlar tarafından kullanılır ve blokların doğrulama sürecinde önemli bir rol oynar. PoW sisteminde, madenciler karmaşık bir matematiksel problemin çözümü için yarışırlar ve doğru çözümü bulan ilk madenci yeni oluşturulan

kripto para ile ödüllendirilir ve aynı zamanda yeni işlemleri blok zincirine ekleyebilme yeteneğine sahiptir.

Madencilerin çözmeleri gereken matematiksel problem zor ve kaynak yoğun olacak şekilde tasarlanır, böylece madencilik süreci rekabetçi olur ve herhangi bir varlığın ağ üzerinde fazla kontrol sahibi olmasını engeller. Bir madenci doğru çözümü bulduktan sonra, çözümü ağa yayınlar ve diğer düğümler hızlı bir şekilde çözümün geçerli olup olmadığını doğrular. Çözüm doğrulandıktan sonra, madenci blok ödülü kazanır, bu genellikle sabit miktarda kripto paradır ve bekleyen tüm işlemler yeni bloğa dahil edilir.

Konsensüs mekanizmaları, blokzincir ağlarında gerçekleştirilen işlemlerin doğru ve güvenli bir şekilde onaylanması için kullanılan bir yöntemdir. Blokzincir ağları, merkezi olmadıkları için, işlemleri onaylamak için bir aracıya ihtiyaç duymazlar. Bunun yerine, konsensüs mekanizması ile tüm ağ katılımcıları, işlemlerin doğruluğunu ve geçerliliğini kontrol ederek, yeni blokların eklenmesi için anlaşmaya varırlar.

2.7.2 Proof of Stake (PoS)

Proof of Stake (PoS), blockchain ağlarında konsensus sağlamak için kullanılan bir algoritma türüdür. Proof of Stake, madencilik yerine "hisse sahipliği" veya "pay sahipliği" kavramını kullanır. Bu algoritma, blokların doğrulama sürecinde katılımcıların belirli bir miktar kripto varlık stake etmesini gerektirir. PoS sisteminde, düğümler, sahip oldukları kripto paranın miktarına göre işlemleri doğrulamak için seçilirler.

Proof of Work'ten farklı olarak, madencilerin matematiksel problemleri çözmek için yarıştığı PoW sistemlerinin aksine, PoS sistemleri katılımcıların doğrulama sürecine katılabilmek için ağda bir paya sahip olmalarını gerektirir. Pay, katılımcı tarafından yatırılmış bir kripto para mevduatı veya tutulan belli miktardaki kripto paradan oluşabilir.

PoS sistemlerinde işlemleri doğrulama seçimi, katılımcının payının büyüklüğü ve kripto parayı ne kadar süredir tuttuğu gibi çeşitli faktörlere bağlıdır. Genellikle, bir

katılımcının elindeki pay ne kadar yüksekse, işlemleri doğrulamak ve ödüller kazanmak için seçilme olasılığı o kadar yüksektir.

PoS'nin PoW'e göre avantajlarından biri, pahalı madencilik ekipmanına ihtiyaç duymaması nedeniyle çok daha az enerji yoğun olmasıdır. Ayrıca, PoS sistemleri, daha büyük pay sahiplerinin daha fazla kontrolü olsa da ağ üzerinde mutlak kontrolü önlemesi nedeniyle daha az merkezileşmeye eğilimli olabilir.

2.7.3 Delegated Proof of Stake (DPoS)

DPoS, EOS ve BitShares gibi bazı blokzincir ağları için kullanılan bir konsensüs mekanizmasıdır. Bu sistemde, ağın katılımcıları, belirli bir sayıda sağlayıcının seçilmesiyle blokları doğrulamak için oy kullanırlar.

2.7.4 Proof of Authority (PoA)

PoA, özel blokzincirlerin birinde kullanılan bir konsensüs mekanizmasıdır. Bu sistemde, birkaç otorite figürü, blok zincirinin doğruluğunu sağlamak için görevlendirilir.

2.8 Gizlilik

Blokzincirde gizlilik, verilerin kim tarafından gönderildiğini, kim tarafından alındığını ve ne kadar değer taşıdığını gizlemek için kullanılan tekniklerle sağlanır. Yapılan işlemlerde gerçek kimlikler yerine şifreli adresler kullanılır. Bu adresler rastgele harf ve rakamlardan oluşur ve her işlem için yeni bir adres oluşturulabilir. Bu sayede işlem yapan tarafların kimlikleri gizli kalır. Depolanan verilerin içeriği de şifreli bir şekilde saklanabilir. Bu sayede verilerin ne olduğu veya ne anlama geldiği sadece işlem yapan taraflar tarafından bilinir. İşlemlerde transfer edilen miktarlar da şifreli bir şekilde saklanabilir.Bu sayede işlemin değeri veya büyüklüğü sadece işlem yapan taraflar tarafından bilinir. Örneğin, Monero gibi bazı kripto paralar, işlem miktarlarını gizlemek için halka imza veya gizli işlem gibi teknikler kullanır.

2.9 Web 1.0

Web 1.0, internetin keşfinin ilk ve en ilkel dönemi olarak kabul edilir. Web 1.0, kullanıcıların interneti sadece görüntüleyebildiği ancak içerik oluşturamadığı bir web teknolojisiydi. Web 1.0'da web siteleri statik, tek yönlü ve salt okunurdu. Kullanıcılar web sayfalarındaki bilgilere ulaşabilir ancak katkıda bulunamaz veya etkileşime giremezlerdi. Web 1.0, internetin temelini oluşturduğu için internetin en eski sürümü olarak da adlandırılır.

2.10 Web 2.0

Web2, internetin gelişmiş bir versiyonunu ifade eder ve kullanıcıların etkileşimde bulunabileceği, paylaşım yapabileceği ve içerik oluşturabileceği bir web deneyimi sunar. Web2, web teknolojilerinin evrimiyle ortaya çıkmıştır. Web1, temel olarak web sitelerinin statik içerik sunmasını ve kullanıcıların pasif bir şekilde tüketmesini sağlayan bir yapıya sahipti. Ancak Web2, kullanıcıların daha etkileşimli hale geldiği bir döneme işaret eder. Web2 ile birlikte sosyal medya platformları, kullanıcıların birbirleriyle etkileşimde bulunabildiği ve içerik paylaşabildiği ortamlar haline geldi. Kullanıcıların kendi içeriklerini oluşturmasına ve paylaşmasına olanak tanır. Web uygulamaları, daha zengin ve etkileşimli hale geldi. Kullanıcıların web üzerinde alışveriş yapması, oyun oynaması, veritabanlarına erişmesi gibi işlemler mümkün hale gelmiştir. Web2, büyük veri analitiği ve kişiselleştirilmiş deneyimlere odaklanır. Kullanıcıların verileri toplanır ve bu veriler üzerinden kişiselleştirilmiş öneriler, reklamlar ve içerikler sunulur.

2.11 Web 3.0

Web3, internetin merkezi olmayan ve blok zinciri teknolojilerine dayalı bir sürümü için bir fikirdir. Web3, kullanıcıların internet üzerindeki veri ve içeriğe sahip olmasını ve bunları kripto para birimleri ve NFT'ler gibi araçlarla değerlendirmesini sağlamayı amaçlar. Web3, gücü şirketlerden ziyade bireylerin eline verir.

2.12 Hassas Veriler

Hassas veriler, kişisel kimlik bilgileri, finansal bilgiler, yasal belgeler ve tıbbi kayıtlar gibi özel veya gizli olan herhangi bir bilgiyi ifade eder. Blokzincir teknolojisi bağlamında, hassas verilerin kamuya açık bir blokzincirde saklanması bireylerin gizliliği ve güvenliği açısından risk oluşturabilir.

Bu nedenle, blokzincirde hangi veri türlerinin saklanacağına dikkat etmek önemlidir, özellikle de kamusal bir blokzincir üzerinde. Hassas bilgileri örtbas etmeye ama aynı zamanda doğrulanmasına izin verecek şifreleme ve karma teknikleri kullanmak bir yaklaşımdır. Ayrıca, ağa erişimi onaylanan katılımcılarla sınırlandırılan izinli blokzincirlerin kullanılması, hassas bilgilerin güvenli ve gizli kalmasını sağlayabilir.

2.12.1 Dijital Parmak İzi

Blok zincirinde dijital bir parmak izi genellikle bir hash olarak adlandırılır. Bir hash, kriptografik bir algoritma kullanılarak oluşturulan benzersiz bir dijital tanımlayıcıdır. Veriler bir algoritmaya girildiğinde, sabit uzunluklu karakter dizisi olan hash'i üretir.

Bloklar arasında bir bağlantı oluşturarak, her blok önceki bloğun hash'ini içerdiğinden blok zincirinde bu dijital parmak izi veya hash, depolanan verilerin bütünlüğünü ve gerçekliğini doğrulamanın bir yolu olarak hizmet eder. Belirli bir bloğun hash'ini orijinal hash'iyle karşılaştırarak, verilere yapılan herhangi bir değişiklik kolayca tespit edilebilir.

Ek olarak, kriptografik hash'lerin kullanımı, aynı hash'i üretmeyecek şekilde tasarlandığından, sahtekarlık ve manipülasyonlara karşı daha fazla güvenlik sağlar.

2.13 Bilgi Erişimi

Bloklar arasında, başka kişiler tarafından kullanılmak üzere izin verilen bilgi, belirli bir blok zinciri ve ilişkili kurallarına bağlıdır.

Örneğin Bitcoin gibi açık bir blok zincirinde, tüm işlemler ağdaki herkes tarafından görülebilir. Gösterilen bilgiler arasında gönderen ve alıcının adresleri, aktarılan miktar ve ödenen işlem ücreti yer alır.

Ancak bazı durumlarda, kullanıcılar belirli bilgileri gizli tutmak isteyebilirler. Örneğin, bir işletme veya organizasyon tarafından kullanılan yetkilendirilmiş bir blok zincirinde, katılımcılar rolü veya erişim seviyesine bağlı olarak yalnızca belirli verileri görüntüleyebilirler. Ayrıca birçok blok zinciri, kişisel tanımlama bilgileri veya ticari sırlar gibi hassas verileri yetkisiz erişimden korumak için şifreleme tekniklerini kullanır.

Sonuç olarak, bir blok zinciri sistemindeki gizlilik ve gizlilik düzeyi, geliştiricilerin tasarım seçimlerine ve kullanıcıların belirli ihtiyaçlarına bağlı olacaktır.

2.14 Cüzdanlar

Blok zincirinde, cüzdan, kullanıcıların kripto paralarını saklamak ve işlem yapmak için kullandıkları bir yazılımdır. Cüzdan, kullanıcının kripto paralarını saklamak için kullandığı özel anahtarları içerebilir. Bu anahtarlar, kullanıcının kripto paralarını güvenli bir şekilde işlem yapmasını sağlar.

Cüzdanlar, farklı türlerde olabilir. Örneğin, cüzdanlar, masaüstü veya mobil cihazlar için indirilebilir yazılımlar olarak mevcut olabilir, ayrıca internet tabanlı cüzdanlar veya fiziksel cüzdanlar da mevcut olabilir.

Blok zincirinde cüzdan kullanımı, kullanıcının kripto paralarını göndermek ve almak için kullanabileceği bir adres sağlar. Bu adres, kullanıcının özel anahtarı ile ilişkilidir ve kullanıcının kripto paralarını saklaması ve işlem yapması için gereklidir. Cüzdanlar genellikle ayrıca işlem geçmişi gibi diğer bilgileri de sunarlar.

Cüzdanların kullanımı, kullanıcının kripto paralarının güvenliğini sağlamak için önemlidir. Kullanıcının özel anahtarlarının güvende tutulması ve cüzdanın güncel olarak yedeklenmesi, kullanıcının kripto paralarının kaybını önlemek için önemlidir.

Blochchain'de hashing işlemi, herhangi bir giriş verisini bir kriptografik fonksiyon aracılığıyla sabit boyutlu bir çıktı değerine dönüştürme işlemidir. Bu çıktı değeri, genellikle bir hash veya mesaj özeti olarak adlandırılır. Hashing, blokzincir

teknolojisinin önemli bir yönüdür, çünkü sistemin bütünlüğünü ve güvenliğini sağlamaya yardımcı olur.

Blokzincir ağındaki her blok, önceki blok başlığının bir hash'ini ve kendi işlemlerinin bir hash'ini içerir. Bu, blokların bir zincir oluşturduğu anlamına gelir, her blok bir öncekinden bağlantılıdır ve kırılamayan bir sıra oluşturur. Bir blokla oynamaya çalışmak, o bloğun hash'ini ve zincirdeki tüm sonraki blokların hash'ini değiştirmeyi gerektirir. Bu da hesaplama açısından pahalı ve pratik olarak imkânsız olur.

2.14.1 Soğuk Cüzdan

Soğuk cüzdan, kripto varlıklarını internete bağlı olmayan bir ortamda saklamak için kullanılan bir cüzdan türüdür. Soğuk cüzdanlar, kullanıcıların özel anahtarlarını çevrimdışı ortamda depolayarak kripto varlıklarını çevrimdışı bir şekilde koruma sağlar. Soğuk cüzdanlar, kripto paralarınızı fiziksel olarak saklayabileceğiniz kâğıt veya donanım cihazları şeklinde olabilir. Soğuk cüzdanlar, uzun vadeli yatırımlar için tercih edilir. Soğuk cüzdanlarda, kripto paralarınıza erişmek için özel bir anahtar kodu gereklidir. Bu kodu kaybetmemek veya başkalarıyla paylaşmamak önemlidir.

2.14.2 Sıcak Cüzdan

Sıcak cüzdan, kripto varlıklarınızı internete bağlı bir ortamda saklamak için kullanılan bir cüzdan türüdür. Sıcak cüzdanlar, kullanıcılara kolaylık ve hız sağlar, çünkü çevrimiçi ortamda erişilebilir ve işlem yapılabilirler.

Sıcak cüzdanlar genellikle web tabanlı cüzdanlar, mobil cüzdanlar veya masaüstü cüzdanlar olarak sunulur. Bu cüzdanlar, internete bağlı bir cihaza kurulu olduğundan, kripto varlıklarınıza hızlı erişim sağlar ve işlemlerinizi kolaylıkla gerçekleştirebilirsiniz.

Sıcak cüzdanlar, genellikle günlük işlemler ve kullanım için tercih edilir. Örneğin, sıcak cüzdanlarla online alışveriş yapabilir, kripto varlıklarınızı diğer kullanıcılara transfer edebilir veya ticari işlemler gerçekleştirebilirsiniz. Bunlar, kullanım kolaylığı ve anlık erişim gerektiren durumlar için uygun olabilir.

2.14.3 Web3 Cüzdanı

Blockchainde web3 cüzdanı, merkezi olmayan web veya web3 olarak adlandırılan internetin bir sürümüne erişim sağlayan bir dijital cüzdandır. Web3 cüzdanı, kullanıcıların blok zinciri teknolojisi ile etkileşim kurmasına ve dijital varlıklarını yönetmesine olanak tanır.

Web3 cüzdanı, normal kripto cüzdanlarına benzer şekilde, kullanıcıların kripto para birimlerini saklamalarına, göndermelerine ve almalarına olanak tanır. Ancak, web3 cüzdanı aynı zamanda kullanıcıların akıllı sözleşmelerle etkileşim kurmasına, NFT'leri alıp satmasına, blok zinciri tabanlı uygulamaları kullanmasına ve web3 topluluklarına bağlanmasına da olanak tanır.

2.15 Düğümler

Blokzincir'de düğümler, işlemleri doğrulayan ve blokzincirin bir kopyasını depolayan bireysel bilgisayarlar veya cihazlardır. Her düğüm, sürekli olarak zincire eklenen yeni bloklarla güncellenen tam bir blokzincir kopyası tutar.

Düğümler, işlemleri doğrulamak ve blokzincirin durumu üzerinde uzlaşmaya varmak için birlikte çalışırlar. Yeni bir işlem ağına gönderildiğinde, tüm düğümlere yayılır. Düğümler daha sonra işlemi kontrol ederek geçerli olduğundan emin olur ve eğer geçerliyse, kendi blokzincir kopyalarına eklerler.

Bazı blokzincirlerde, tüm düğümlerin aynı yetkiye sahip olmadığı durumlar vardır. Örneğin, Bitcoin gibi bir proof-of-work blokzincirde, hesaplama gücü sağlayan düğümler (madenciler olarak bilinir) normal düğümlerden daha fazla yetkiye sahiptir. Ancak Ethereum gibi diğer blokzincirlerde, tüm düğümler aynı yetkiye sahiptir ve doğrulama sürecine daha demokratik bir şekilde katılırlar.

2.15.1 Kötü Niyetli Düğümler

Blokzincirde kötü niyetli düğümler, sistemi bozmak veya tehlikeye sokmak amacıyla ağa katılan bireysel bilgisayarlar veya cihazlardır. Bu düğümler, sistemdeki zayıf noktaları sömürmek isteyen hacker'lar, siber suçlular veya diğer kötü niyetli aktörler tarafından işletilebilir.

2.15.1.1 Sybil saldırıları

Bir sybil saldırısında, tek bir kullanıcı, ağı etkilemek ve kontrolünü ele geçirmek için birden fazla sahte kimlik veya düğüm oluşturur.

2.15.1.2 Çift harcama saldırıları

Çift harcama saldırısı, bir kullanıcının aynı dijital tokeni veya dijital para birimini iki veya daha fazla kez harcamaya çalıştığı bir durumu ifade eder. Bu, dijital paraların fiziksel paralardan farklı bir şekilde işlem gördüğü bir alandır. Fiziksel bir paranın bir kez harcanması durumunda, sahibinin elinden çıkar ve bir daha kullanılamaz. Ancak, dijital paralar bir bilgisayar dosyası olduğu için, kopyalanabilir ve tekrar tekrar kullanılabilir gibi görünebilirler.

2.15.1.3 Yüzde Saldırıları

Yüzde saldırısı, kötü niyetli bir düğüm grubunun ağın hesaplama gücünün yüzde 50'sinden fazlasını kontrol ettiği durumlarda gerçekleşir. Bu, blok zinciri geçmişini yeniden yazmalarına ve işlemleri geri alarak ağı ciddi şekilde zarara uğratmalarına olanak tanır.

2.15.1.4 Eclipse Saldırıları

Bir eclipse saldırısı, kötü niyetli bir düğümün başka bir düğümün gelen ve giden bağlantılarını ele geçirerek onu ağın geri kalanından izole etmesidir.

2.16 Dijital İmza

Bir dijital imza, elektronik belgeleri ve bilgileri güvence altına almak için kullanılan bir teknolojidir. İmzanın gerçek sahibinin kimliğini doğrulama yeteneği ve belgenin tamamının bütünlüğünü koruma yeteneği ile bir dijital imza, fiziksel bir imzanın dijital eşdeğeridir.

Blokzincir teknolojisi, dijital imzaları kullanarak verilerin doğruluğunu ve bütünlüğünü sağlar. Bir işlem bir blokzincir ağına gönderildiğinde, işlemi yapan kişi özel anahtarıyla işlemi dijital olarak imzalar. Bu imza, ağın gerçekten işlemi gönderen kişinin yaptığını doğrulamasına yardımcı olur.

Bir dijital imza aynı zamanda işlemin değiştirilmediğini doğrulamada da önemlidir. İşlem, imzalandıktan sonra değiştirilirse, dijital imza geçersiz hale gelir ve ağ, işlemi reddeder. Dijital imza daha sonra işleme veya mesaja eklenir ve ağa yayılır. Ağdaki diğer düğümler, göndericinin genel anahtarını kullanarak dijital imzayı doğrulayabilir ve işlemin veya mesajın değiştirilmediğinden emin olabilirler.

2.17 Dağıtılmış Veritabanı

Blokzincirde dağıtılmış bir veritabanı, her bir bilgisayarın (düğümün) tüm veritabanının bir kopyasına sahip olduğu bir ağ üzerinde yayılan bir veritabanı türüdür. Verilerin depolanmasına yönelik bu yaklaşım, geleneksel merkezi veritabanlarına kıyasla birçok avantaj sağlar.

Blokzincirde dağıtılmış bir veritabanının temel faydalarından biri artan şeffaflık ve güvenliktir. Ağdaki her düğümün tüm veritabanının bir kopyasına sahip olması nedeniyle, verilerin bütünlüğünü tehlikeye atabilecek tek bir zayıf nokta veya açıklık yoktur.

Blokzincirde dağıtılmış bir veritabanının diğer bir faydası, artan verimlilik ve ölçeklenebilirliktir. Geleneksel merkezi veritabanları ile karşılaştırıldığında, verilerin miktarı arttıkça, o verileri depolamak ve yönetmek için daha güçlü donanımlara ihtiyaç duyulur. Bunun aksine, blokzincirde dağıtılmış bir veritabanında, yeni düğümler ağa eklenebilir ve talebin artması durumunda ek donanım yükseltmeleri gerektirmeden işleme kapasitesini artırabilir.

2.18 Blokzincir Ağları

Blokzincir ağları, işlemleri kaydeden ve verileri güvenli ve şeffaf bir şekilde saklayan merkezsiz dijital defterlerdir. Bu ağlar, kripto para birimleri, dijital sertifikalar ve sözleşmeler de dahil olmak üzere çeşitli dijital varlıkları yönetmek ve takip etmek için kullanılabilir.

2.18.1 Kamusal Blokzincir Ağları

Herkesin katılmasına ve işlem yapmasına izin veren merkezsiz ve izinsiz bir ağdır. Bitcoin, Ethereum ve Litecoin gibi kamusal blokzincir örnekleri bulunur. Kamusal bir blokzincirde tüm işlemler şeffaftır ve herkes tarafından görülebilen blokzincire kaydedilir. Herhangi biri, ağın konsensüs mekanizmasına katılarak zincire yeni bir blok ekleyebilir.

2.18.2 Özel Blokzincir Ağları

Özel bir blokizncir, belirli kişilere veya kuruluşlara erişimi kısıtlamalı bir blokzincir ağıdır. Genellikle, tedarik zinciri yönetimi gibi özel amaçlar için tek bir kuruluş veya kuruluş konsorsiyumunda kullanılır. Özel bir blokzincirde katılımcılar, ağa katılabilmek için izin almak zorundadır ve genellikle kamusal blokzincir ağlarındaki katılımcılardan daha sıkı kurallar ve yönetmeliklere tabidirler.

2.18.3 Hibrit Blokzincir Ağları

Hibrit bir blokzincir, kamusal ve özel blokzincirlerin özelliklerini birleştirir. Belirli katılımcıların ağa sınırlı erişimine izin verirken, bazı derecede merkezsizlik ve şeffaflık sağlayabilir. Hibrit bir blokzincir, belirli verilerin özel olarak korunması gerektiği durumlarda kullanışlı olabilirken, diğer veriler ise ağda herkes tarafından paylaşılabilir.

2.18.4 Konsorsiyum Blokzincir Ağları

Konsorsiyum blokzincirler, kamusal ve özel blokzincirler arasında bir hibrittir ve birden fazla organizasyonun ortaklaşa paylaştığı bir blokzincir ağı sürdürmesine olanak tanır. Konsorsiyum blokzincirler, kamusal blokzincirlere göre daha kontrollü bir ortam sağlar, ancak yine de bazı derecede merkezsizlik korur.

2.18.5 Federatif Blokzincir Ağları

Federatif blokzincirler, konsorsiyum blokzincirlerine benzer, ancak tek bir organizasyondan birden fazla düğümün doğrulama sürecine katılmasına izin veren ek işlevselliğe sahiptir.

2.19 Ethereum Yapısı

Ethereum, bir blokzinciri platformudur ve Bitcoin'den sonra en popüler kripto para birimidir. Ethereum, Bitcoin gibi bir kripto parayı içerir ama aynı zamanda bir programlama dilinde yazılmış smart contractları (akıllı sözleşmeler) yürütmek için bir araç sağlar. Bu araçlar, kullanıcıların decentralized aplikasyonlar (dApps) veya decentralize bir otomatikleştirilmiş sistemler oluşturmasına olanak tanır.

Ethereum ağı, birçok düğümlerden oluşur. Bu düğümler, ağın işlemlerini gerçekleştirir ve blokları onaylar. Her işlem, bir Ether (ETH) ücreti karşılığında gerçekleştirilir. Bu ücret, işlemleri gerçekleştiren düğümler tarafından kabul edilir ve ağın sağlıklı çalışmasını sağlamak için kullanılır.

Ethereum ağı, dApps ve smart contractların yürütülmesi için "Ethereum Virtual Machine" (EVM) adı verilen bir yazılım kullanır. Bu yazılım, smart contractların yürütülmesi için gerekli olan işlem gücünü sağlar. EVM, smart contractların yürütülmesi için kullanılan Ether'lerin bir kısmını kabul eder.

Ethereum'un diğer önemli bir özelliği ise, onun "proof of stake" (poS) algoritmasını kullanmasıdır. Bu algoritma, madencilik işlemleri yerine, düğümlerin ağda sahip olduğu Ether miktarına dayanarak işlemleri onaylamayı sağlar. Ethereum 2.0 sürümünde ise, PoS algoritması kullanılacaktır. Bu, ağın enerji tüketimini azaltmayı ve işlem hızını arttırmayı amaçlamaktadır.

2.20 Akıllı Sözleşmeler

Akıllı sözleşmeler, bir kontratın hükümlerinin otomatik olarak uygulanmasını sağlayan bilgisayar programlarıdır. Bu programlar, belirli koşulların yerine getirilmesi durumunda belirli eylemleri otomatik olarak tetiklerler.

Akıllı sözleşmeler genellikle blok zincir teknolojisi ile birlikte kullanılır. Bu teknoloji, taraflar arasında güvenli ve şeffaf bir şekilde işlem yapmayı sağlar. Blok zinciri üzerinde çalışan akıllı sözleşmeler, manipülasyona karşı korunur ve bir kere uygulanmaya başlandığında durdurulamaz veya değiştirilemez.

Bu sistem, taraflar arasındaki işlemleri hızlandırabilir, maliyetleri azaltabilir ve verimliliği artırabilir.

2.20.1 Akıllı Sözleşmelerde Hız, Verimlilik ve Doğruluk

Akıllı sözleşmeler, işlemleri hızlandırır. Klasik bir sözleşmeyle karşılaştırıldığında, belirli bir durumda ne yapılması gerektiğini belirleyen bir dizi otomatik ve önceden programlanmış talimatları içerirler. Bu, bir anlaşmanın manuel olarak yönetilmesine kıyasla daha hızlıdır. İşlemlerin otomatikleşmesi sayesinde bürokrasiyi ve gereksiz işlemleri azaltabilir. Bu, genel verimliliği artırır ve operasyonel maliyetleri düşürür. Ayrıca, aracıları ortadan kaldırarak işlemleri daha verimli hale getirir. Manuel işlemlerde olabilecek hataları ve yanlış anlaşılmaları ortadan kaldırır. Sözleşme şartları kod olarak yazılır ve otomatik olarak uygulanır, bu da işlemlerin daha doğru ve tutarlı olmasını sağlar.

2.20.2 Akıllı Sözleşmelerde Güven ve Şeffaflık

Arada üçüncü bir taraf olmadığı ve işlem kayıtlarının şifreli olarak katılımcılar arasında paylaşıldığı için bilgilerin kişisel çıkar için değiştirilip değiştirilmediğini sormaya gerek yoktur.

2.20.3 Akıllı Sözleşmelerde Güvenlik

Akıllı sözleşmeler, genellikle blokzincir teknolojisi üzerine inşa edildikleri için, bu teknolojinin getirdiği bir dizi güvenlik özelliğinden yararlanırlar. Blokzincir, merkezi olmayan bir yapıda olduğu için, bir hükümet veya özel kuruluş tarafından kontrol edilmez ve tek bir hata noktası içermez. Bu, verilerin manipülasyon veya hileli faaliyetlere karşı korunmasına yardımcı olur.

2.20.4 Akıllı Sözleşmelerde Tasarruf

Akıllı sözleşmeler otomatiktir ve bu da işlem hızını önemli ölçüde artırır. Bu hızlılık, beklemek yerine anında sonuç almayı mümkün kılar, böylece zaman tasarrufu sağlar. Genellikle aracıları ortadan kaldırır çünkü taraflar doğrudan ve otomatik olarak birbirleriyle etkileşim kurar. Bu, genellikle aracılara ödenen ücretleri ortadan kaldırır ve böylece maliyet tasarrufu sağlar.

2.20.5 Akıllı Sözleşmelerde Kalıcılık

Akıllı sözleşmelerin kalıcılığı, yazılımın içerdiği kodun blokzincir üzerindeki dağıtık doğası sayesinde sağlanır. Bir akıllı sözleşme bir kez blok zincirine eklendiğinde, orada kalıcı olarak kalır ve değiştirilemez. Akıllı sözleşmenin hükümleri uygulandığında, işlem blok zincirine eklenir ve orada kalıcı olarak kalır. Bu nedenle, akıllı sözleşmeler geri dönüşümsüz ve değiştirilemezdir.

2.20.6 Akıllı Sözleşmelerde İnsan Faktörü

Akıllı sözleşmelerin hazırlanması, hataların önlenmesi için doğru bir şekilde yapılmalıdır. Bunun için, kodlama konusunda deneyimli ve yetenekli bir ekip tarafından hazırlanmaları gerekmektedir. Ayrıca, akıllı sözleşmelerin test edilmesi ve güvenliği konusunda uzman kişiler tarafından denetlenmesi de önemlidir.

Ayrıca, akıllı sözleşmelerin yönetimi de insan faktörünü dikkate almalıdır. Sözleşmelere erişim ve onay süreçlerinin kontrol edilmesi, hata yapma riskini azaltacaktır. Sözleşmelerin güncellenmesi ve yönetimi de doğru kişilerce yapılmalı, doğru prosedürler izlenmelidir.

Bu yüzden insan faktörü, hazırlık ve yönetim aşamalarında dikkatle ele alınması gereken bir faktördür. Doğru bir şekilde yönetildiğinde, akıllı sözleşmelerin hataları en aza indirgenir ve daha güvenli bir şekilde kullanılırlar.

2.20.7 Akıllı Sözleşme Türleri

Farklı türdeki akıllı sözleşmeler farklı amaçlara hizmet eder ve değişen düzeylerde karmaşıklığa sahiptir. Farklı akıllı sözleşme tiplerinin önemi, belirli iş ihtiyaçlarını ele alabilme ve belirli süreçleri otomatikleştirebilme yeteneklerinde yatar.

2.20.7.1 Kendini Uygulayan Sözleşmeler

Kendi şartlarını herhangi bir dış otoriteye veya mekanizmaya güvenmeden otomatik olarak uygulayan sözleşmelerdir. Bu sözleşmeler, önceden belirlenmiş koşullar yerine getirildikten sonra insan müdahalesi gerektirmez. Örneğin, bir akıllı sözleşme, ödeme blokzincir üzerinde onaylandığında otomatik olarak bir aracın sahipliğini alıcıya aktarabilir.

2.20.7.2 Kendini Yürüten Sözleşmeler

Bunlar, akıllı sözleşmeler ile geleneksel yasal sözleşmeleri birleştirerek daha esnek ve sağlam çözümler oluşturan sözleşmelerdir. Bu tür sözleşmeler, belirlenen şartlar karşılandığında otomatik olarak kendilerini yerine getiren, bilgisayar kodu ile yazılan

dijital sözleşmelerdir. Akıllı sözleşmeler genellikle bilgisayar koduyla yazılır ve blok zinciri ağı üzerinde çalışır, bu da onların yürütülmesinin şeffaf, değiştirilemez ve güvenli olmasını sağlar.

2.20.7.3 Ricardian Sözleşmeler

Doğal dil ile kodu birleştirerek hem yasal olarak bağlayıcı anlaşmalar oluşturan hem de makineler tarafından yürütülebilen şartları kodlayan sözleşmelerdir. Bu tür akıllı sözleşmeler hem insanlar hem de makinalar tarafından okunup anlaşılabilir. Ricardian sözleşmeler hukuki bir sözleşmeyi bilgisayar programıyla birleştirerek, mahkemelerde yasal olarak bağlayıcı olmalarını sağlar. Anlaşmanın koşullarını açıklayan insan okunabilir metinler ve yürütmesini otomatikleştiren makine okunabilir kodlar içerirler.

2.20.7.4 Hibrit Sözleşmeler

Blokzincirler bağlamında, hibrit sözleşmeler, zincir dışı ve zincir üstü mantığın öğelerini birleştiren akıllı sözleşmeleri ifade eder. Hibrit sözleşmeler, zincir üstü ve zincir dışı mantığın güçlü yönlerinden faydalanabileceği için kullanışlıdır. Örneğin, bir sözleşmenin bazı kısımları, performans sınırlamaları nedeniyle bir blok zinciri üzerinde yürütülemeyecek kadar karmaşık olabilir, ancak zincir dışında daha verimli bir şekilde yürütülebilir. Sözleşmenin diğer kısımları, şeffaflık ve denetlenebilirlik nedenleriyle zincir üstünde yürütülmelidir.

2.20.8 Akıllı Sözleşme Metrikleri

Akıllı sözleşme metrikleri, bir blok zinciri platformunda çalışan ve çeşitli işlem türlerini uygulayan yazılım programları olan akıllı sözleşmelerin kalitesini ve performansını ölçen yazılım metrikleridir.

2.20.8.1 Gaz Kullanımı

Gaz kullanımı, Ethereum platformunda bir akıllı sözleşmeyi dağıtmak ve yürütmek için gereken gaz miktarını (Ether cinsinden) ölçen bir akıllı sözleşme metriğidir. Gaz, akıllı sözleşme kodunu çalıştıran madencilerin hesaplama kaynaklarını telafi eden yürütme ücretine karşılık gelir. Gaz kullanımı, veri türleri, kontrol yapıları, işlev değiştiricileri ve harici çağrılar gibi geliştiriciler tarafından yapılan uygulama seçimlerine bağlıdır. Gaz kullanımı, bir akıllı sözleşmenin maliyetini ve performansını etkilediği için önemli bir metriktir.

2.20.8.2 Yürütme Süresi

Yürütme süresi, bir blok zinciri platformunda bir akıllı sözleşmeyi dağıtmak ve çalıştırmak için gereken süredir. Yürütme süresi, akıllı sözleşme kodunun karmaşıklığına, ilgili işlemlerin sayısına ve türüne, gaz limitine ve fiyatına ve akıllı sözleşmeyi yürüten düğümlerin performansına bağlı olarak değişebilir. Yürütme süresi önemli bir metriktir çünkü akıllı sözleşmenin maliyetini ve verimliliğini etkiler.

2.20.8.3 Kod Karmaşıklığı

Kod karmaşıklığı, bir akıllı sözleşme kodunu anlama, test etme ve sürdürme zorluğunu ölçmekle alakalıdır. Kod karmaşıklığı, kod satırı sayısı, fonksiyon sayısı, dal sayısı, bağımlılık sayısı ve tasarım kalıplarının kullanımı gibi faktörlerden etkilenebilir. Kod karmaşıklığı önemli bir ölçüttür çünkü akıllı bir sözleşmenin kalitesini ve güvenliğini etkiler. Yüksek kod karmaşıklığı akıllı sözleşmedeki hata, güvenlik açığı ve bug riskini artırabilir ve bu da blok zinciri ortamında ciddi sonuçlar doğurabilir. Bu nedenle, uygun kodlama uygulamaları, test araçları ve yeniden düzenleme teknikleri kullanılarak akıllı sözleşmelerin kod karmaşıklığının azaltılması arzu edilir.

2.20.8.4 Sözleşme Çağrıları

Akıllı sözleşme ölçümlerindeki sözleşme çağrıları, bir blokzincir platformundaki akıllı sözleşmeler arasındaki etkileşimidir. Bir sözleşme çağrısı, bir akıllı sözleşme tarafından başka bir akıllı sözleşmenin bir işlevinin ya da değişkeninin çağrılmasıdır.

Sözleşme çağrıları, çağrıyı yapan ve çağrıyı alanın aynı ya da farklı hesaplara ait olmasına bağlı olarak dahili ya da harici olabilir. Sözleşme çağrıları, bir akıllı sözleşmenin performansını ve güvenliğini etkilediği için önemli bir metriktir.

2.20.8.5 Sözleşme Bakiyesi

Sözleşme bakiyesi, bir akıllı sözleşmenin bir blok zinciri platformundaki hesabında tuttuğu kripto para miktarının ölçüsüdür. Bir sözleşme bakiyesi, diğer hesaplardan ya da sözleşmelerden ödeme alınarak artırılabilir ya da diğer hesaplara ya da sözleşmelere ödeme gönderilerek azaltılabilir. Sözleşme bakiyesi önemli bir ölçüttür çünkü bir akıllı sözleşmenin işlevselliğini ve güvenliğini etkiler. Yüksek bir kontrat bakiyesi, bir akıllı kontratın popüler, karlı olduğunu ya da değerli bir hizmet sunduğunu gösterebilir.

2.20.8.6 Sözleşme Hacmi (Volume)

Sözleşme hacmi, akıllı sözleşmenin bir blok zinciri platformunda gerçekleştirdiği işlem miktarının bir ölçüsüdür. Sözleşme hacmi, bir akıllı sözleşme işlevini ya da değişkenini çağıran işlemlerin sayısının sayılmasıyla ya da bu işlemler tarafından aktarılan kripto para biriminin değerinin toplanmasıyla hesaplanabilir. Yüksek bir sözleşme hacmi, bir akıllı sözleşmenin yaygın olarak kullanıldığını, değerli bir hizmet sunduğunu ya da çok fazla gelir elde ettiğini gösterebilir.

2.20.8.7 Sözleşme Hacmi (Volume)

Sözleşme değeri, akıllı sözleşmeyle ilişkili kripto para biriminin ya da jetonun piyasa değeri, akıllı sözleşme tarafından elde edilen gelir ya da kar, akıllı sözleşmenin sosyal etkisi ya da faydası ya da akıllı sözleşmenin itibarı ya da güvenilirliği gibi çeşitli yöntemlerle tahmin edilebilir[12]. Sözleşme değeri önemli bir ölçüttür çünkü bir akıllı sözleşmenin başarısını, sürdürülebilirliğini ve çekiciliğini yansıtır. Yüksek bir sözleşme değeri bir akıllı sözleşmenin yenilikçi, rekabetçi ya da faydalı olduğunu gösterebilir

2.20.9 Akıllı Sözleşme Test Süreci

Blokzincirinde bir akıllı sözleşmeyi test etmek için, akıllı sözleşme dili ve platformuyla uyumlu bir test çerçevesi ve test aracı kullanmak gerekir. Örneğin, Ethereum platformu için Solidity'de bir akıllı sözleşme geliştiriliyorsa, test çerçevesi olarak Truffle'ı ve test aracı olarak Ganache'yi kullanılabillir. Bir test çerçevesi, akıllı sözleşmeniz için test senaryoları yazmak ve yürütmek için bir yapı ve bir dizi kütüphane sağlar. Bir test aracı ise gerçek kripto para harcamadan ya da ana ağı etkilemeden akıllı sözleşme konuşlandırılabilir ve onunla etkileşime geçebilecek simüle edilmiş bir blok zinciri ortamı sağlar. Blokzincirde bir akıllı sözleşmeyi test etmek için genel adımlar şunlardır:

- Test çerçevesini kullanarak akıllı sözleşme için test senaryoları yazılır. Birim testi, entegrasyon testi, işlevsel test, güvenlik testi vb. gibi farklı test türlerini kullanabilirsiniz. Ayrıca JavaScript, Python, Solidity gibi farklı diller kullanılabilir.

- Yerel bir blok zinciri örneği oluşturmak için test aracı çalıştırılır ve akıllı sözleşme bunun üzerine dağıtılır. Test aracı gaz limiti, gaz fiyatı, blok zamanı, hesaplar vb. gibi ağ parametrelerini özelleştirmek için yapılandırılabilir.

- Test çerçevesini ve test aracını kullanarak test senaryolarını yürütülür. Akıllı sözleşmeye işlem göndermek ve sonuçları kontrol etmek için test aracı kullanılır. Test aracı akıllı sözleşmenin ve blok zincirinin durumunu incelemek için de kullanılabilir.

- Test sonuçları analiz edilir ve akıllı sözleşme kodunda herhangi bir hata varsa düzeltilir. Test çerçevesi, test vakaların raporlarını ve günlüklerini oluşturmak için kullanılabilir. Akıllı sözleşme mantığındaki sorunları tespit etmek ve çözmek için hata ayıklama araçları da kullanılabilir.

- Akıllı sözleşmenin kalitesinden ve performansından memnun kalana kadar test süreci tekrarlanır.

2.20.10 Akıllı Sözleşme Güvenlik Açıkları

Akıllı sözleşmeler, blokzincir üzerinde çalışan ve farklı taraflar arasındaki anlaşmanın otomatik olarak yerine getirilmesini sağlayan bilgisayar programlarıdır. Akıllı sözleşmeler, güvenlik, şeffaflık, verimlilik ve yenilik gibi birçok avantaj sunabilir. Ancak, akıllı sözleşmeler de diğer yazılımlar gibi çeşitli türde saldırılara maruz kalabilir ve bunlar fon kaybı, veri bozulması veya sözleşme arızası gibi sonuçlara yol açabilir.

2.20.10.1 Tekrar Kayıt Sorunları

Blockchain, doğası gereği, verileri değiştirilemez ve kalıcı bir şekilde saklar. Bu, aynı işlemin iki kez kaydedilmesini veya bilgilerin çakışmasını önler. Ancak, akıllı sözleşmelerde işlemlerin otomatik doğası, belirli durumlarda tekrar kayıt sorunlarına yol açabilir. Örneğin, bir kullanıcı bir akıllı sözleşmeye yanlışlıkla iki kez etkileşimde bulunabilir ve bu, sözleşmenin iki kez uygulanmasına neden olabilir. Bu tür durumlar genellikle kullanıcı hatası veya yanıltıcı bir kullanıcı arayüzü sonucu ortaya çıkar ve akıllı sözleşme tasarımı ile doğrudan ilgili olmayabilir. Bu tür sorunların çözümü genellikle daha iyi kullanıcı eğitimi ve daha iyi kullanıcı arayüzleri gerektirir. Bunun yanında, akıllı sözleşme kodunda ek kontroller de bulunabilir.

2.20.10.2 Mantık Hataları

Bunlar, akıllı sözleşmenin kodunda veya tasarımında beklenmeyen veya istenmeyen sonuçlara yol açan hatalardır. Mantık hataları, bir programın beklenen şekilde çalışmamasına neden olan ve genellikle kodun doğru yazılmamasından kaynaklanan hatalardır. Bir akıllı sözleşmedeki mantık hatası, beklenmedik davranışlara veya hatta güvenlik açıklarına yol açabilir. Mantık hataları, saldırganların akıllı sözleşme mantığındaki boşlukları veya kontrolleri istismar etmelerine veya atlatmalarına izin verebilir.

2.20.10.3 Keyfi Adreslere Harici Çağrılar

Bu sorun, bir akıllı sözleşmenin doğrulanmamış veya geçerli olmayan bir adrese harici bir çağrı yapması durumunda ortaya çıkar. Bu, saldırganların akıllı sözleşmeyi zararlı bir sözleşmeyi çağırmaya kandırmasına ve zararlı kod çalıştırmasına veya fon çalmasına olanak tanır. Örneğin, saldırganlar sahte bir yönlendirici adresi kullanarak merkezi olmayan bir borsa (DEX) toplayıcısından fon boşaltabilir

2.20.10.4 Kontrol-Etki-Etkileşim Deseni İhlali

Güvenli akıllı sözleşme yazmak için kullanılan en iyi uygulamalardan biridir ve herhangi bir durum değişikliği yapmadan veya herhangi bir fon göndermeden önce tüm koşulları ve gereksinimleri kontrol etmeyi ve tüm etkiler bittikten sonra harici sözleşmelerle etkileşime girmeyi içerir. Bu deseni ihlal etmek, akıllı sözleşmeyi tekrar kayıt saldırılarına veya durum değişikliklerini veya fon transferlerini istismar edebilecek diğer saldırı türlerine maruz bırakabilir.

2.20.10.5 Eksik Doğrulama/Girdi Doğrulama İhlali

Bu sorun, bir akıllı sözleşmenin harici kaynaklardan aldığı girdileri veya parametreleri doğrulamadığı veya temizlemediği durumda ortaya çıkar. Bu, saldırganların akıllı sözleşmeye zararlı kod veya veri enjekte etmelerine veya akıllı sözleşmenin davranışını veya çıktısını manipüle etmelerine olanak tanır. Örneğin, saldırganlar büyük bir girdi değeri kullanarak bir tam sayı taşması veya eksikliğine neden olabilir, bu da yanlış hesaplamalara veya fon transferlerine yol açabilir.

2.20.10.6 Anlık Kredi Saldırısı

Bu, anlık kredilerin özelliklerinden yararlanan bir saldırı türüdür. Anlık krediler, aynı işlemde ödünç alınabilen ve geri ödenebilen teminatsız kredilerdir. Saldırganlar anlık kredileri, merkezi olmayan finans (DeFi) protokollerinin fiyatlarını veya likiditesini manipüle etmek ve fiyat farklarından yararlanmak veya kar elde etmek için kullanabilir. Örneğin, saldırganlar anlık kredileri büyük miktarda jeton ödünç almak, başka bir jetona takas etmek ve sonra geri takas etmek için kullanabilir.

2.21 Metamask

MetaMask, Ethereum tabanlı akıllı sözleşmelerle ve diğer Ethereum hizmetleriyle etkileşim kurmanızı sağlayan bir kripto cüzdanıdır. Bir tarayıcı eklentisi olarak çalışır ve kullanıcıların Ethereum adreslerini yönetmelerine, Ether ve diğer Ethereum tabanlı tokenleri saklamalarına ve transfer etmelerine olanak sağlar.

MetaMask ayrıca kullanıcıların Ethereum tabanlı dApp'ler ile etkileşim kurmasına olanak sağlar. Bu, kullanıcıların akıllı sözleşmelerle etkileşime girebilmesi, token satın alabilmesi veya DeFi hizmetlerini kullanabilmesi gibi şeyleri içerir.

Ayrıca, MetaMask bir "web3 enjektörü" görevi görür, bu da onun Ethereum tabanlı uygulamaları desteklemeyen normal bir web tarayıcısını, Ethereum ve akıllı sözleşmelerle etkileşim kurabilen bir tarayıcıya dönüştürdüğü anlamına gelir.

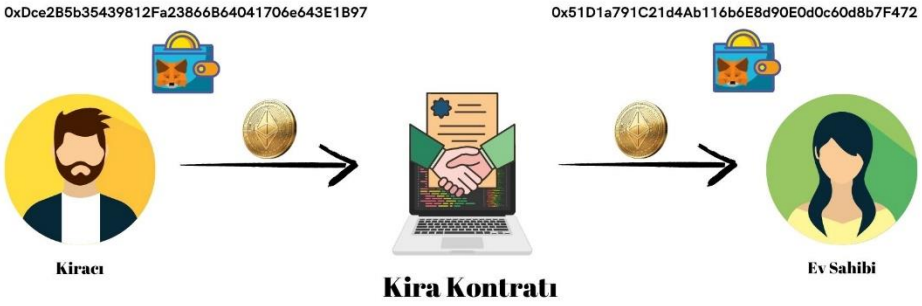
2.21.1 Goerli Test Ağı

Goerli test network, Ethereum tabanlı bir test ağıdır. Test ağları, geliştiricilerin ve kullanıcıların blok zinciri tabanlı uygulamaları güvenli ve ucuz bir şekilde denemelerine olanak tanır. Test ağlarında kullanılan Ether, ana Ethereum ağından farklıdır ve gerçek değeri yoktur.

Goerli test network ile MetaMask arasındaki bağlantı, MetaMask'in Goerli test network'ünü desteklemesi ve kullanıcıların MetaMask aracılığıyla Goerli test network'üne bağlanabilmesi ile sağlanır. Böylece, kullanıcılar MetaMask ile Goerli test network'ündeki uygulamaları deneyebilir ve test Ether'i alabilir veya harcayabilir

3. MATERYAL VE YÖNTEM

Projede blokzincir teknolojisi kullanılarak bir kira kontratı gerçeklenmeye çalışılmıştır. Bu blok zincir teknolojisinin kullanımı, blokzincirin bir parçası olan Ethereum ağında gerçekleşmiştir. Ethereum ağının sağladığı akıllı kontrat oluşturma özelliği sayesinde, akıllı kontratları oluşturmaya yardımcı olan Solidity yazılım dili ile bir kira kontratının prototip uygulaması yapılmıştır. Bu uygulama yapılırken Remix isimli web tabanlı IDE kullanılmıştır. Uygulamada tarafların kripto cüzdanları bulunmaktadır ve bu cüzdanlarının adresleri ile kontrata erişim sağlamaktadırlar. Bu cüzdanlar ise Metamask isimli web üzerinde eklentisi ve uygulaması bulunan, ethereum blokzinciri üzerinde iletişim kurmak için kullanılan sağlayıcı tarafından sağlanmaktadır. Metamask'taki Goerli Test Ağı içerisinde 2 adet cüzdan adresi oluşturulmuştur. Ardından bu cüzdanlara web üzerindeki Goerli Faucet isimli kripto para musluğundan bir miktar test ethereumu gönderilmiştir. Sonrasında bu cüzdan adresleri uygulamanın yazıldığı Remix Web IDE'sindeki "Account" kısmına eklenmiştir. Remix Web IDE'si ile uygulama yazılımının derlenmesi, sözleşmenin ağa dağıtılması ve işlemleri kontrol edilebilmesi gibi olanakları sağlanmıştır. IDE içerisine uygulamanın yazılımının yazılması ve derlenmesinin ardından "Deploy & Run Transactions" bölümünün arayüzü, sözleşmedeki tarafların adreslerinin girişini, ödenecek kiranın değeri, ödenecek depozitonun değeri ve her sene yenilenen kira artış değerlerinin kontrat oluşturulurken düzenlenmesi olanaklarını sağlamaktadır. Bu veriler kontrata işlendikten sonra kontrat oluşturucusu tarafından ağ üzerindeki bir sonraki bloğa gönderilir.



Şekil 3.1. Akıllı Kontrat ile Kiracı ve Ev sahibi arasındaki ödeme işlemi

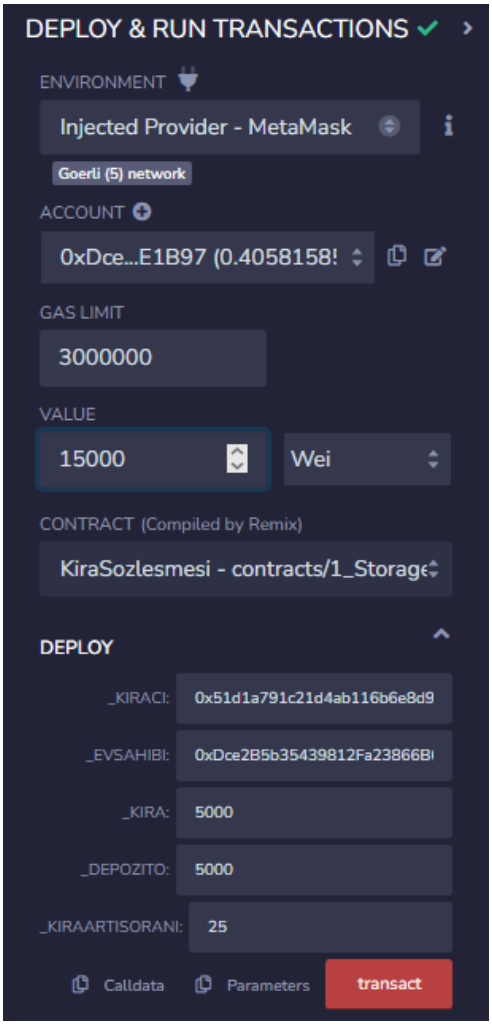
[1]

[1] Şekil Canva grafik tasarım platformu ile hazırlanmıştır.

4. UYGULAMA

Uygulama sırasında Remix Web IDE'sine giriş yapılmaktadır. Sonrasında eklenecek cüzdanlarımızın adreslerini "Injected Web3" kısmına giriyoruz. Metamask eklentisi açıldığında Remix'te bağladığımız cüzdanları görüntülediğimizde cüzdanın bağlanmış olduğunu doğruluyoruz. Sonrasında Remix arayüzünde kontrata dahil edilen değerler (kiraci, evSahibi, kira, depozito, kiraArtisOrani) girildikten sonra (Şekil 4.1) IDE üzerinde "Transact" kısmından kontratın ağa dağıtımı gerçekleşecektir.

[2]



Şekil 4.1. Kontratın veri girişi yapılan Remix üzerindeki arayüzü

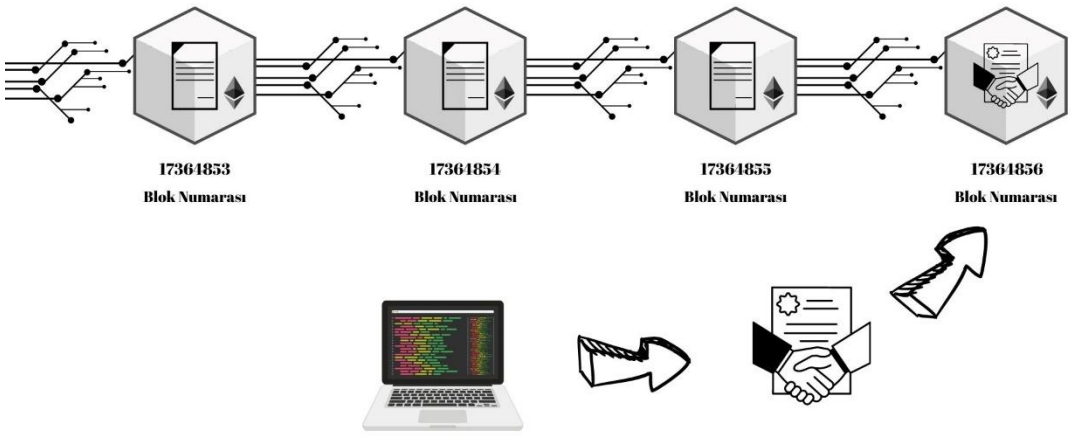
[2] Şekil Remix Web IDE'si üzerinden alınmıştır.

Bu dağıtım esnasında akıllı kontratın bytecode'unu içeren işlem, gönderildiği düğüm tarafından doğrulanır ve işlem havuzuna eklenir. İşlem havuzu, henüz onaylanmamış işlemlerin beklediği bir alandır.

İşlem havuzundaki işlemler, madenciler tarafından seçilir ve yeni bir blok oluşturmak için bir araya getirilir. Madenciler, işlemleri gas fiyatına göre sıralar ve en yüksek gas fiyatına sahip olanları seçer çünkü bu şekilde daha fazla gelir elde ederler. Madenciler ayrıca blok başına gas limitine de uymak zorundadır.

Madenciler, blok oluşturmak için proof-of-work mekanizmasında zorlu bir matematik problemini çözmeye çalışır. Problem, bloğun hash değerinin belirli bir hedefin altında olmasını gerektirir. Bu problemi çözen ilk madenci, bloğu ağa yayınlar ve diğer madencilerden onay ister.

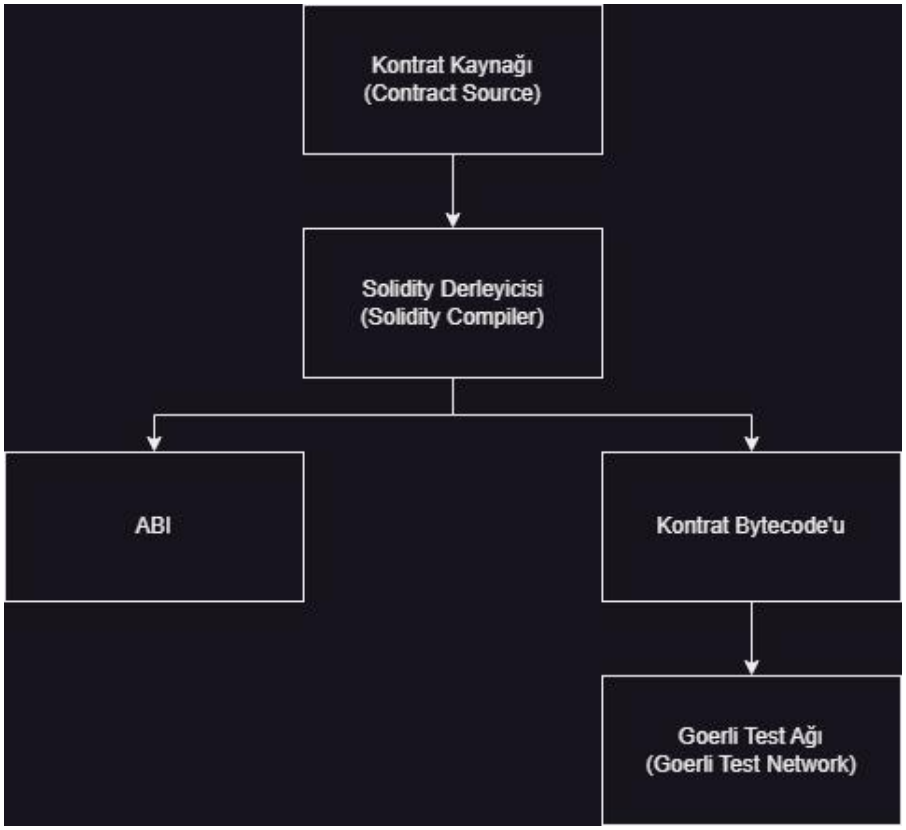
Diğer madenciler, bloğun geçerli olduğunu doğrular ve kendi blok zincirlerine ekler. Ayrıca yeni bir blok oluşturmak için çalışmaya devam eder. Blok zinciri, en uzun geçerli zincirdir. Blokların üzerine yeni bloklar eklendikçe, blok zinciri uzar ve daha güvenli hale gelir.



[3]Şekil 4.2 Akıllı kontratın, ethereum ağındaki bir sonraki bloğa eklenişi

[3] Şekil Canva grafik tasarım platformu ile hazırlanmıştır.

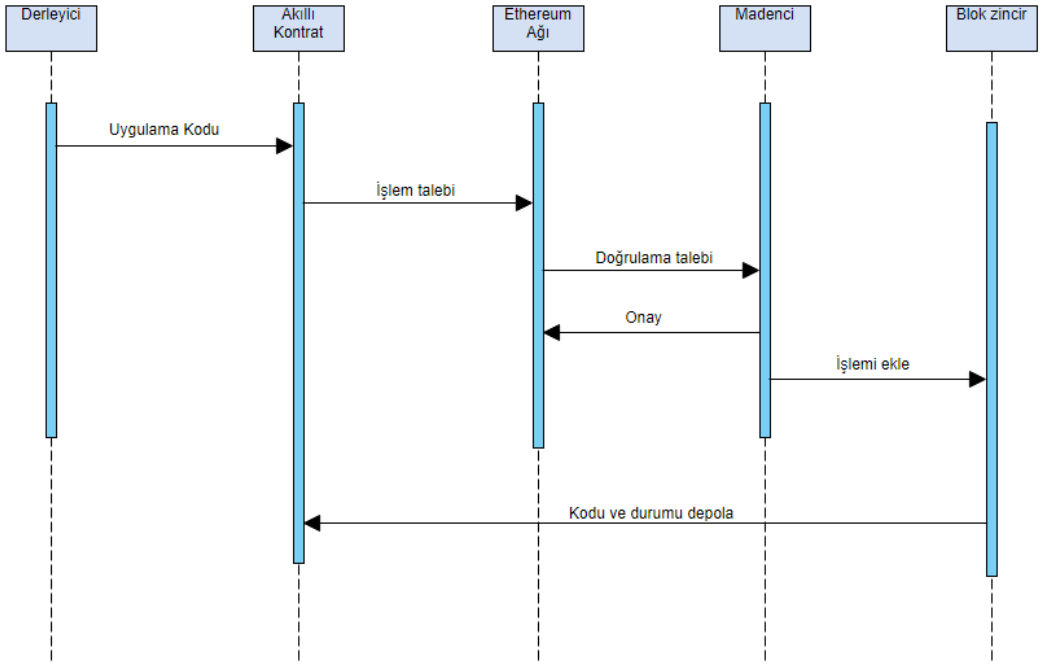
Akıllı kontratın bytecode'unu içeren işlem, blok zincirine dahil edildiğinde onaylanmış olur. Bytecode, akıllı kontratın makine diline çevrilmiş halidir. Bytecode, Ethereum Sanal Makinesi (EVM) tarafından çalıştırılabilir ve aynı zamanda makine diline derlenebilir durumdaki komut setinin isimlendirilmesinde kullanılmaktadır. Akıllı kontratlar bytecode olarak Ethereum ağına dağıtılır ve blokzincir üzerinde saklanır. Bu durumda, akıllı kontrat Ethereum ağına dağıtılmış olur. Akıllı kontratın adresi, işlemin hash değerinden türetilir.



Şekil 4.3 Kontratın ağa gönderildiği esnadaki işlemlerin basit gösterimi

Bytecode'un yanı sıra derleyiciden dağıtım sürecinde bir de ABI oluşturulur. ABI, akıllı sözleşmenin dış uygulamalar ve diğer akıllı sözleşmeler ile iletişim kurmasını ve etkileşimde bulunmasını sağlayan bir arayüzdür. Remix'te çalıştırırken MetaMask'tan adres enjekte ettiğinizde, remix arka planda ABI'yi kullanarak akıllı kontrat ile iletişim

kurmaktadır. Remix'in sağ üst köşesindeki "Deploy & Run transactions" sekmesinde, "Environment" seçeneğini "Injected Web3" olarak ayarlandığında, Remix MetaMask'tan gelen ABI'yi almıştır ve akıllı kontrat fonksiyonlarını görüntemiştir. Bu sayede remix üzerinden akıllı kontrat ile etkileşimde bulunmuştur.



Şekil 4.4 Kodun derleyiciden blokzincire kadar olan işlemlerinin sequence diyagramı

5. BULGULAR VE TARTIŞMALAR

5.1 Proje Bulguları

Şu anda mevcut kira ödemeleri sisteminde Türkiye'de ödemeler yazılı kira kontratı şeklinde yapılmaktadır. Bu sistemde kiracıların kiralarını geç ödemesi, eksik ödemesi, ödememesi; ev sahiplerinin kira dönemi bittiğinde belirlenen orandan fazla zam yapması, kiracıları haksız yere çıkarması, sözleşmeyi haksızca feshetmesi durumları sorunları ortaya çıkmaktadır. Geliştirilen sistemde bu problemler ortadan kaldırılmaya çalışılmıştır.

5.2 Geliştirilen Projenin Avantajları ve Kolaylıkları

Yazılı kira kontratına göre geliştirilen sistemin avantajları olarak kiraların süresinde ve eksiksiz ödenmesi sağlanacaktır. Blok zincir sistemine eklenen akıllı kontratın değiştirilemezliği sayesinde kira üzerindeki tüm şartlar başarılı bir şekilde sonuç verecektir. Akıllı kontrat üzerindeki şartlar kiracının ve ev sahibinin tüm haklarını korumak için yazılmıştır ve böylece aradaki kira ödenmemesi, eksik kira ödenmesi, kiranın zamanında ödenmemesi, sözleşmenin ev sahibi tarafından haksızca feshedilmesi, kira dönemi bitiminde ev sahibinin belirlenen orandan fazla zam yapması, kiracının eve zarar vermesi durumunda sözleşmenin devam etmesi gibi hukuksuz durumlar ortadan kaldırılacaktır. Sistemde sözleşme üçüncü bir tarafa ihtiyaç duyulmadan kiracı ve ev sahibi tarafından yapılmaktadır.

5.3 Geliştirilen Projenin Dezavantajları ve Zorlukları

Geliştirilen proje birçok probleme çözüm getirmiş olsa da halen dezavantajlara sahiptir. Bu dezavantajlar şunlardır ki bu akıllı kontratı uygulayacak herkesin telefon, bilgisayar, tablet vb. teknolojik alete sahip olması gerekmektedir. Her ne kadar teknoloji gelişmiş olsa da bütün bireylerin bu sistemi kullanabilmesinin yolu bütün bireylerin teknolojik alete sahip olmasından geçmektedir ve bu da ekonomik olarak refah düzeyinin en azından bir seviyede olmasını gerektirmektedir. Ayrıca henüz ülkemizde kripto para ile ödemeler yasal değildir ve bu sistem ancak bir yasalaşmadan sonra resmi bir şekilde kullanılabilecektir. Ödemeler ise ethereum kripto para birimi

cinsinden yapıldığı için ve Ethereum kripto para biriminin değeri istikrarsız olduğu için kira bedelinin sürekli değişmesi tarafların istemeyeceği durumlar ortaya çıkarabilir.

6. SONUÇ VE ÖNERİLER

Bu çalışmada, Türkiye'deki mevcut kira ödemeleri sisteminde yaşanan sorunlara çözüm getirmek amacıyla blok zincir teknolojisine dayalı bir akıllı kontrat sistemi geliştirilmiştir. Geliştirilen sistem, kiracı ve ev sahibi arasındaki kira sözleşmesini akıllı kontrat olarak blok zincir üzerinde kaydetmekte ve kira ödemelerini ethereum kripto para birimi ile gerçekleştirmektedir. Bu sayede, tarafların haklarını koruyan, güvenli, şeffaf ve değiştirilemez bir kira ödeme sistemi sunulmaktadır. Geliştirilen sistem, yazılı kira kontratına göre birçok avantaj sağlamakta ve kira ödemelerindeki hukuksuz durumları ortadan kaldırmaktadır. Bununla birlikte, geliştirilen sistemin bazı dezavantajları ve zorlukları da bulunmaktadır. Bunlar arasında teknolojik aletlere erişim, kripto para ile ödeme yapma alışkanlığı, ethereum kripto para biriminin değerindeki dalgalanma ve yasal düzenleme eksikliği sayılabilir. Bu dezavantajların ve zorlukların aşılması için ileride yapılacak çalışmalarda daha fazla araştırma ve geliştirme yapılması gerekmektedir. Bu çalışma, Türkiye'deki kira ödemeleri sisteminin blok zincir teknolojisi ile nasıl iyileştirilebileceğine dair önemli bir katkı sunmaktadır.

KAYNAKLAR

ANTALYA, O. G., & ÇAVDAR, P. (2020). İfa Güçsüzlüğüne Genel Bakış ve Onun Kira Sözleşmesindeki Görünümü. *Marmara Üniversitesi Hukuk Fakültesi Hukuk Araştırmaları Dergisi*, *26*(1), 219–237. https://doi.org/10.33433/maruhad.725794

BEŞTAŞ, M. (2022). Blok Zincir ve Ulak (Oracles) İle Parametrik İşlemlerin Gerçekleştrilmesi. *European Journal of Science and Technology*. https://doi.org/10.31590/ejosat.1093723

DURDU, A., & GÖKÇE, A. (2022). Blokzincir teknolojisi akıllı sözleşme uygulamalarının kamu alımlarında kullanımı. *Sakarya Üniversitesi İşletme Enstitüsü Dergisi*, *4*(2), 43–48. https://doi.org/10.47542/sauied.1019897

Parlar, T., & Prof, A. (2022). Uygulamaları [Blockchain Technology and Decentralized Finance Applications. In *Journal of Politics, Economy and Management* (Vol. 5, Issue 2).

TEVETOĞLU, M. (2021). ETHEREUM ve AKILLI SÖZLEŞMELER. *İnönü Üniversitesi Hukuk Fakültesi Dergisi*. https://doi.org/10.21492/inuhfd.852860

ULUYOL, Ç., & ÜNAL, G. (2020). Blok zinciri teknolojisi. *Bilişim Teknolojileri Dergisi*, 167–175. https://doi.org/10.17671/gazibtd.516990

YILMAZ TÜRKMEN, S., & ERÖZEL DURBİLMEZ, S. (2019). Blockchain Teknolojisi ve Türkiye Finans Sektöründeki Durumu. *Finans Ekonomi ve Sosyal Araştırmalar Dergisi*. https://doi.org/10.29106/fesa.509254

EK-1

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract KiraSozlesmesi {
    address public kiraci;
    address public evSahibi;

    uint256 public kira;
    uint256 public depozito;

    enum State {Active, Terminated}
    State public state;

    bool public kiraciViolation;
    bool public evSahibiRelocation;

    uint256 public kiraArtisOrani;
    uint256 public sonKiraArtisTarihi;
    uint256 public ilkOlusturmaTarihi=block.timestamp;
    uint256 public yeniOdemeTarihi;
    uint256 public aySayimi=1;

    constructor(address _kiraci, address _evSahibi, uint256 _kira, uint256 _depozito,
uint256 _kiraArtisOrani) payable {
        kiraci = _kiraci;
        evSahibi = _evSahibi;
        kira = _kira;
        depozito = _depozito;
        state = State.Active;
        kiraciViolation = false;
```

```
        evSahibiRelocation = false;
        kiraArtisOrani=_kiraArtisOrani;

        require(msg.sender == kiraci || msg.sender == evSahibi , "Yalnizca kiraci ya da ev sahibi sozlesmeyi olusturabilir");
        require(msg.value == kira * 2 + depozito, "Sozlesme basladiginda kiraci ilk ayin ve ikinci ayin kirasini ve depozitoyu odemelidir. ");

        payable(evSahibi).transfer(msg.value);

        yeniOdemeTarihi= ilkOlusturmaTarihi;
        sonKiraArtisTarihi=ilkOlusturmaTarihi;
    }

    function paykira() external payable {
        require(msg.sender == kiraci, "Yalnizca kiraci kirayi odeyebilir");
        require(msg.value == kira, "Kiraci kira bedelini tam olarak odemelidir");
        require(state == State.Active, "Sozlesme aktifligi");

         // Ev sahibine ödemenin transfer edilmesi
        payable(evSahibi).transfer(msg.value);
        yeniOdemeTarihi = yeniOdemeTarihi + 30 days;
        aySayimi++;
    }

    function terminateBykiraci() public payable {
        require(msg.sender == kiraci, "Kiraci sozlesmeyi feshedebilir");
        require(state == State.Active, "Sozlesme aktifligi.");

        state = State.Terminated;
```

```
        payable(kiraci).transfer(depozito + kira);
    }

    function terminateByevSahibiViolation(bool _kiraciViolation) public payable {
        require(msg.sender == evSahibi, "Ev sahibi sozlesmeyi feshedebilr");
        require(state == State.Active, "Sozlesme aktifligi");

        kiraciViolation = _kiraciViolation;

        require(kiraciViolation, "Sozlesmeyi feshetmek icin olaganustu durum
olusmalidir");

        state = State.Terminated;

        payable(kiraci).transfer(kira);
    }

    function terminateByevSahibiRelocation(bool _evSahibiRelocation) public
payable {
        require(msg.sender == evSahibi, "Ev sahibi sozlesmeyi feshedebilr");
        require(state == State.Active, "Sozlesme aktifligi");

        evSahibiRelocation = _evSahibiRelocation;

        require(evSahibiRelocation, "Sozlesmeyi feshetmek icin olaganustu durum
olusmalidir");

        state = State.Terminated;

        payable(kiraci).transfer(depozito + kira);
    }
```

```
    function evSahibiIthar() external payable {
        require(msg.sender == evSahibi, "Sadece ev sahibi ihtar cekebilir");
        require(state == State.Active, "Kontrat aktifligi");

        require(block.timestamp >= yeniOdemeTarihi + 30 days * 2 && msg.value < kira * 2, "Kiraci sonraki ayi odedi");

        uint256 sonTarih = block.timestamp + 30 days;

        if (block.timestamp >= sonTarih && msg.value < kira) {
            state = State.Terminated;
            payable(kiraci).transfer(depozito + kira);
        }
    }

    function setkiraArtisOrani() external {
        require(msg.sender == evSahibi, "Yalniz ev sahibi kira artis oranini belirleyebilir");
        require(state == State.Active, "Sozlesme aktif degil");
        require(yeniOdemeTarihi >= sonKiraArtisTarihi + 360 days, "Kira artisi her 12 ayda bir olacaktir");

        sonKiraArtisTarihi=yeniOdemeTarihi;
        kira = kira * (100 + kiraArtisOrani) / 100;
    }
}
```

ÖZGEÇMİŞ

Adı Soyadı : Cüneyt Balcı

Doğum Yeri ve Yılı : Kartal, 2000

Medeni Hali : Bekar

Yabancı Dili : İngilizce

Eğitim Durumu

Lise : Maltepe Final Temel Lisesi, 2018

Lisans : Fenerbahçe Üniversitesi, 2023

Mesleki Deneyim