ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

BLOK ZİNCİRİ TEKNOLOJİSİNİN NESNELERİN İNTERNETİ ORTAMINDA UYGULANMASI

Aydın ELBÜZ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2022

ÖZET

Yüksek Lisans Tezi

BLOK ZİNCİRİ TEKNOLOJİSİNİN NESNELERİN İNTERNETİ ORTAMINDA UYGULANMASI

Aydın ELBÜZ

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ömer Özgür TANRIÖVER

Nesnelerin interneti, bir ağ üzerinden birbirine bağlanan ve içerisinde algılayıcıların bulunduğu farklı tipteki cihazlar topluluğudur. İlk uygulaması 1982 yılında gerçekleştirilen bu teknoloji; günümüzde özellikle internetin yaygınlaşmasıyla beraber, tanımlama ve izleme teknolojileri, kablolu ve kablosuz algılayıcılar ve dağıtık bilişim sistemleri gibi birçok alanda sıklıkla kullanılmaktadır. Kullanım alanları genişledikçe, bu teknoloji veri toplama amacıyla da değerlendirilmeye başlanmıştır. Diğer yandan, işlenebilir verinin önemli ölçüde kıymet kazandığı bir çağda, nesnelerin interneti yoluyla veri toplamak sadece verinin ilintili olduğu alandaki araştırmacılar için değil, aynı zamanda bu veriden ticari menfaat elde etmek isteyenler için de mühim bir uğraş hâline gelmiştir. Bu durum, bu tür verilerin güvenilir ve sağlıklı bir şekilde pazarlanabilmesi ihtiyacını ortaya çıkarmıştır. Blok zinciri teknolojisi; şeffaflık, kayıtların değiştirilemezliği ve ademimerkeziyetçilik gibi özellikleri dolayısıyla bu ihtiyacın karşılanması noktasında etkili ve güvenli bir araç olarak öne çıkmaktadır.

Bu çalışmada, nesnelerin interneti cihazlarından toplanan verilerin pazarlanabileceği blok zinciri tabanlı bir veri ticaret platformu önerilmektedir. Önerilen bu platformun temel amacı kullanıcıların güvenilir ve sağlıklı bir şekilde ticaret yapması olup, platformun bileşenleri bu ihtiyaçlara göre belirlenmiş ve ihtiyaçlara çözüm yöntemleri sunulmuştur. Bileşenlerin, bileşenler arası ilişkilerin ve bileşenlerin karşıladığı gereksinimlerin belirlenmesi esnasında SysML kullanılarak modelleme yapılmıştır. Daha sonra önerilen bu veri ticaret platformunun gerçeklenebileceğini göstermek amaçlı web tabanlı bir platform oluşturulmuş ve olası senaryolarda çalıştırılarak platformun ihtiyaçlarının karşılanabildiği sonucuna ulaşılmıştır.

Ocak 2022, 58 sayfa

Anahtar Kelimeler: Blok zinciri, nesnelerin interneti, dağıtık kayıt defteri, veri ticareti

ABSTRACT

Master Thesis

APPLICATION OF BLOCKCHAIN TECHNOLOGY ON INTERNET OF THINGS

Aydın ELBÜZ

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Ömer Özgür TANRIÖVER

Internet of things (IoT) is a collection of various types of devices that has sensors and connected to each other over a network. This technology was first implemented in 1982 and today, it is frequently used in many areas such as identification and monitoring technologies, wired and wireless sensors and distributed information systems. As its usage areas have expanded, this technology has started to be used for data collection. On the other hand, in an era where workable data has gained significant value, collecting data via the IoT has become an important endeavor not only for researchers in the field of data, but also for those who want to gain commercial benefits from this data. This situation has revealed the need to market such data in a reliable and healthy way. Due to its features such as transparency, immutability and decentralization, blockchain technology stands out as an effective and safe method to meet this need.

In this study, a blockchain-based data trading platform is proposed where IoT data can be marketed. The main purpose of this platform is to allow users to trade data in a reliable and healthy way, so the components of the platform were determined according to these needs and solution methods were presented. SysML was used for components, component relations and requirements that the components met. Then, a web-based platform was created to show that this platform could be implemented, and it was concluded that the requirements could be met by running it with possible scenarios.

January 2022, 58 pages

Key Words: Blockchain, internet of things, distributed ledger, data trading

TEŞEKKÜR

Çalışmaya başladığımız ilk günden beri yardımlarını esirgemeyerek akademik ortamda gelişmeme katkıda bulunan danışman hocam sayın Dr. Öğr. Üyesi Ömer Özgür TANRIÖVER'e (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı) ve çalışmalarım süresince bilgi, öneri ve yardımını esirgemeyen değerli hocam sayın Dr. Murat OSMANOĞLU'ya (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı) en derin duygularla teşekkür ederim.

Çalışmalarım süresince maddi, manevi pek çok fedakârlık göstererek beni destekleyen ve her anımda yanımda olan aileme ve dostlarıma tüm içtenliğimle minnettarlığımı belirterek, en derin duygularla teşekkür ederim.

Aydın ELBÜZ
Ankara, Ocak 2022

İÇİNDEKİLER

KISALTMALAR DİZİNİ

CSS	Basamaklı Stil Sayfaları (Cascading Style Sheets)
EVM	Ethereum Sanal Makinesi (Ethereum Virtual Machine)
HTML	Köprü Metni Biçimlendirme Dili (HyperText Markup Language)
IoT	Nesnelerin İnterneti (Internet of Things)
SysML	Sistem Modelleme Dili (Systems Modeling Language)

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Nesnelerin interneti (Internet of things, IoT), bir ağ üzerinden birbirine bağlanan ve içerisinde sensörlerin bulunduğu cihazlar topluluğudur. İlk örneği 1982 yılında Carnegie Mellon Üniversitesi'nde geliştirilmiş; bir kola otomatının stok durumu ve içerisinde bulunan içeceklerin soğuk olup olmama bilgilerini ARPANET ağı üzerinden göndermesi sağlanmıştır. Daha sonraları Raji (1994), yayımladığı makalede bu konsepti "Ev aletlerinden fabrikalara kadar her şeyi bütünleştirmek ve otomatikleştirmek için küçük veri paketlerini büyük düğüm kümelerine taşımak." olarak tanımlamıştır. Nesnelerin interneti kavramı ise ilk olarak 1999 yılında Kevin Ashton tarafından kullanılmıştır (Anonymous 2021a). Günümüzde akıllı evler ve şehirlerin yanı sıra tarım ve sağlık sektöründe de sıklıkla kullanılan nesnelerin interneti teknolojisinin kullanım alanları daha da genişlemekte, bu durum da bu tür cihazların veri toplama amaçlı kullanılmasını sağlamaktadır. Toplanan bu veriler sağlık sektöründe tanı ve tedavilerin daha hızlı bulunmasını amaçlayan araştırmalarda veya tarım sektöründe üretilen ürünlerin rekoltesini artırmaya yönelik çalışmalarda kullanılabilmektedir. Bunların yanı sıra işlenebilir verilerin önemli ölçüde değer kazandığı günümüzde, bu veriler üzerinden ticari menfaat elde etmek isteyen kişiler için de veri toplama işlemi önemli bir uğraş hâline gelmiştir. Bu durum, bu tür verilerin güvenilir bir şekilde pazarlanabilmesi ihtiyacını doğurmuştur.

Blok zinciri teknolojisi, merkezî olmayışı ve verilerin değiştirilemez bir biçimde kaydedilebilmesi gibi özellikleri sayesinde bu ihtiyacı karşılayabilecek önemli bir araç olarak kullanılabilmektedir. Blok zinciri, dağıtık kayıt defterini gerçekleyen ve işlemlerin açık ve merkezî olmayan bir sistemde depolanmasını sağlayan bir teknolojidir. Blok zincirinde işlemler, sistemdeki kullanıcılar tarafından onaylandıktan sonra değiştirilemez ve silinemez bir şekilde kaydedilmektedir. Blok zinciri salt okunur bir veri tabanı sistemi olmakla birlikte, sahte veya hatalı bir işlemin zincire yazılmasına izin vermemektedir. Güvenilirliği, şeffaflığı ve veri değiştirilemezliğini sağlayan blok zinciri teknolojisi günümüzde tarımdan eğitime, sağlıktan sigortacılığa kadar birçok farklı alanda güncel problemlerin çözümünde de etkin bir biçimde kullanılmaktadır.

Bu çalışmada, nesnelerin internetine bağlı cihazlardan toplanan işlenebilir verilerin pazarlanabildiği blok zinciri tabanlı bir veri ticaret platformunun nasıl olması gerektiğinden, gereksinimlerinden ve bileşenlerinden bahsedilmektedir. Böyle bir platformun geliştirilmesi, beraberinde bazı gereksinimler ve problemler getirmektedir. Bu amaçla verilerin doğru bir şekilde yüklenebilmesi, alıcı ve satıcı arasındaki para ve veri transferinin güvenilir ve sorunsuz bir biçimde gerçekleştirilebilmesi, kötü niyetli alıcı veya satıcılara karşı alınan önlemler gibi bu ve daha birçok ihtiyaç ve problemin nasıl giderilebileceği bu çalışmada anlatılmaktadır. Başta platformun bileşenlerinin belirlenmesinde olmak üzere, bu bileşenlerin birbirleriyle olan ilişkilerinin belirlenmesi ve gereksinimlerin belirlenerek bu gereksinimlerin hangi bileşenler tarafından karşılanabileceğinin tespit edilmesi aşamalarında SysML yönteminden yararlanılmıştır.

Çalışmanın 2. Bölümünde bu alan üzerindeki kavramlar ve yapılan çalışmalar sunulmaktadır. 3. Bölüm'de, blok zinciri tabanlı veri ticaret platformunun gerçeklenebilmesi için karşılanması gereken gereksinimlerden, bileşenlerden, bileşenlerin içeriklerinden ve kullanılması gereken yöntem ve teknolojilerden bahsedilmektedir. 4. Bölüm'de ise, 3. Bölüm'de bahsedilen yöntemler ve teknolojiler kullanılarak bir kısmı gerçeklenen web tabanlı bir veri ticaret platformunun genel operasyonları ve akışı anlatılmaktadır. Bölüm 5'te, gerçeklenen bu platform sonucunda elde edilen bulgulardan bahsedilmekte ve bu bulgular, karşılanan ihtiyaçlar ile birlikte tartışılırken; Bölüm 6'da ise bu çalışmaların sonucunda elde edilen bilgiler toparlanarak genel bir çerçevede sunulmaktadır.

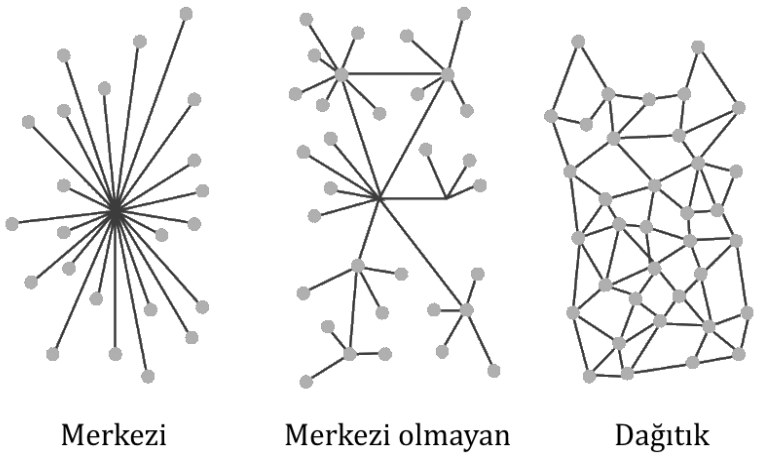
2. KURAMSAL TEMELLER VE LİTERATÜR ÖZETİ

Bu bölümde, bir blok zinciri tabanlı veri ticaret platformunun gerçekleştirilmesi sırasında gereken kuramsal temeller açıklanmakta; bununla birlikte literatürde yapılan çalışmalardan da bahsedilmektedir.

2.1 Kuramsal Temeller

2.1.1 Dağıtık kayıt defteri

Dağıtık kayıt defteri, işlemleri ve kayıtları depolamak veya saklamak için kullanılan merkezi olmayan bir veri tabanı sistemidir. Bu sisteme katılmış olan her kullanıcı bu kayıtlardaki işlemlere ulaşabilir ve bu kayıtların bir kopyasını kendine alabilir. Sistemde gerçekleştirilen her bir değişiklik veri tabanına yansıtılır. Örneğin deftere yeni bir kayıt yazılması gerektiği zaman, sistemdeki kullanıcılar bir uzlaşma (consensus) algoritması yardımıyla yeni gelen kaydın yazılıp yazılmayacağına karar verir. Kaydın yazılması kararlaştırıldığında, kayıt veri tabanına eklenir ve kayıt defteri güncelleştirilir. Ekleme ve güncelleştirme işlemi tamamlandıktan sonra kayıt defteri son hâlini almış olur. Sistemdeki diğer kullanıcılar ise, kayıt defterinin bu son hâlini kendilerine alarak kendilerindeki kopyayı güncellerler. Bu sayede her kullanıcı kayıt defterinin son hâlini kendine almış olur.



Şekil 2.1 Merkezi, merkezi olmayan (ademimerkezi) ve dağıtık sistemler (Baran 1964)

Dağıtık kayıt defterinin en önemli özelliği, tahrif edilmeye karşı dayanıklı olmasıdır. Şekil **2.1**'de de gösterildiği üzere merkezi bir yapıda, merkeze yapılan bir saldırı tüm sistemi etkilemektedir. Tek bir merkeze bağlı olmayan ancak küçük merkezlerin bulunduğu ademimerkeziyetçi bir yapıda ise, küçük merkezlerin birkaçına yapılabilecek saldırılar sistemin tamamını olmasa da belli bir parçasını etkileyecektir. Ancak dağıtık sistemlerde bir merkez bulunmadığı gibi; kötü niyetli bir kullanıcı kendi kopyası üzerinde değişiklik yapsa da, sistemdeki diğer kullanıcılarda kayıt defterinin orijinal hâli bulunduğu için yaptığı değişiklik kabul görmeyecektir. Ayrıca yeni kayıtların yazılması ve kayıt defterinin güncellenmesinde kriptografik özet fonksiyonları ve elektronik imzaların kullanılması, bu sistemi daha güvenli hâle getirmekte ve blok zinciri teknolojisinde kullanılmasının da önünü açmaktadır.

2.1.2 Blok zinciri

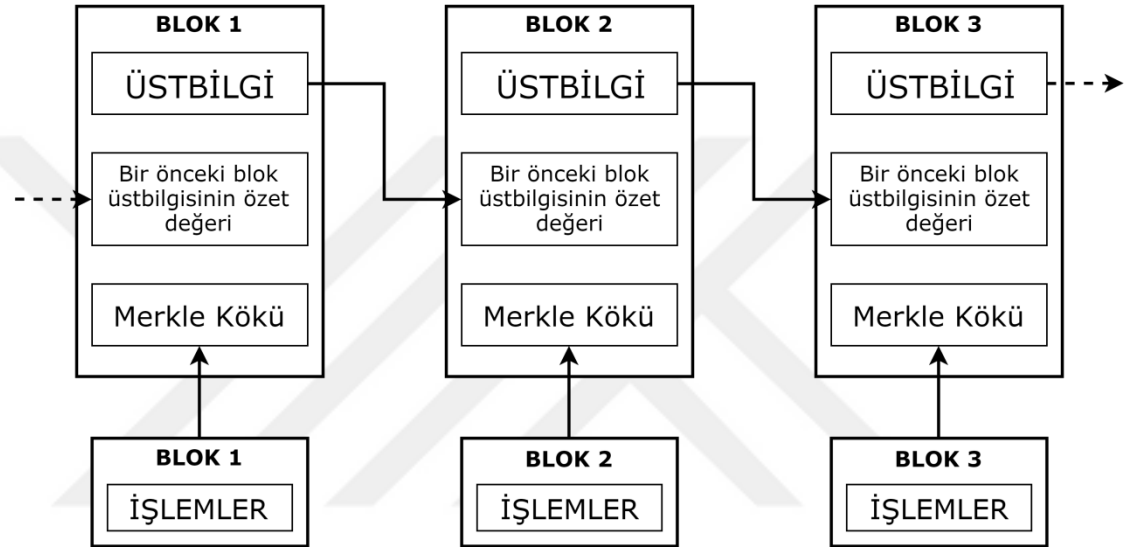
Blok zinciri, dağıtık kayıt defterinin gerçeklenmesini sağlayan değiştirilemez bir depolama teknolojisidir. İsmini, gerçekleştirilen işlemlerin yazıldığı bloklardan ve bu blokların bir zincir gibi arka arkaya sıralanmasından almaktadır.

İlk olarak Haber ve Stornetta (1991) tarafından dijital dokümanların zaman damgalarının değiştirilmesini önlemek amacıyla ortaya atılmıştır. Daha sonraları ise, Nakamoto (2008) tarafından Bitcoin isimli kripto para sisteminin ortaya atılmasıyla somutlaştırılmış ve kısa zamanda çok popüler bir teknoloji hâline gelerek birçok alanda kullanılmaya başlanmıştır.

Blok zincirinin sağladığı güvenilirlik, kullanıcı gizliliği ve veri değiştirilemezliği; bu teknolojinin kullanılma sebeplerinin başında gelmektedir. Blok zinciri sistemine dâhil olan her kullanıcı, sistemin kendine ürettiği blok zinciri adresiyle birlikte açık ve kapalı anahtarlar ile işlemlerini gerçekleştirir. Bu sayede kullanıcının blok zinciri adresi ortaya çıkmış olsa da kim olduğu bilinemeyeceği için kullanıcı gizliliği sağlanmış olur. Adres üretme işleminde açık ve kapalı anahtarlar ile birlikte kriptografik özet fonksiyonları da kullanıldığı için rastgelelik sağlanmış olur. Bu sayede kullanıcılar işlemlerini güvenle

yapar ve yapılan işlemleri onaylayarak bloklara yazarlar. Onaylanarak bloğa yazılan bir işlem kaydının bir daha düzenlenmesi veya silinmesi olanaksızdır.

Blok zincirindeki her işlem blok adı verilen küçük yapılarda tutulur ve bu bloklar kapasitelerini doldurduktan sonra ana zincirin en sonuna eklenir. Daha sonra yeni gelen işlemlerin yazılması için yeni bir blok oluşturulmaya başlanır ve bu süreç böyle devam eder.



Şekil 2.2 Blok zincirindeki blokların yapısı

Şekil **2.2**'de de gösterildiği üzere her blok üst bilgiden ve işlemlerden oluşmaktadır. Bloğun üst bilgisi içerisinde zaman damgası, zorluk derecesi ve versiyon gibi bilgilerin yanı sıra; bir önceki bloğa ait üst bilginin özet değeri ile mevcut bloğun işlemleriyle Merkle ağacı kullanılarak oluşturulmuş bir özet değeri mevcuttur. Dolayısıyla arka arkaya sıralanmış iki blok arasında bir doğrulama mekanizması bulunmaktadır. Blokta yapılacak en ufak bir değişiklik bile bloktaki üst bilginin özet değerini tamamıyla değiştireceği için bu, sonradan gelen bütün blokların da özet değerini değiştirecek ve zinciri bozacaktır. Bu sebeple kötü niyetli bir kullanıcı kendi blok zinciri kopyasındaki bir bloğu değiştirse bile, çoğunluk ve uzlaşma sağlanamayacağı için bu blokları gerçek ve ana zincirdeki bloklarla değiştiremeyecektir.

İşlemleri onaylayarak bloklara yazan kullanıcılara madenci (miner) adı verilir. Madenciler, işlemleri iş ispatı veya benzeri yöntemlerle bloklara yazarlar. Ardından bu yazma işlemi sonucunda belirli bir miktar sistem içi para ile ödüllendirirler. Bu sayede hem madencilerin madencilik yaparak blokları yazmaları teşvik edilir, hem de sistemin güvenliği sağlanmış olur.

Blok zincirleri, erişim durumlarına göre izinsiz (permissionless) ve izinli (permissioned) olmak üzere ikiye ayrılmaktadır. İzinsiz blok zincirleri çoğunlukla kripto para sistemlerinde kullanılmakta olup herhangi bir izin istemeden herkesin katılabileceği ağlardır. İşlemler ağdaki herhangi bir kullanıcı tarafından onaylanabilmekte ve sistemin güvenliği de yine bu kullanıcılar tarafından sağlanmaktadır. Öte yandan izinli blok zincirlerinde bir erişim kontrolü bulunmakta olup, sadece otoriteler tarafından izin verilen kullanıcılar sisteme katılabilmektedir. Bununla birlikte işlemler de sadece sistemde izin verilen kullanıcılar tarafından onaylanabilmekte ve bu kullanıcılar ağ sahibi tarafından kontrol edilebilmektedir. Geliştirilen sistemin ihtiyacına göre blok zinciri izinli veya izinsiz olabilmektedir.

2.1.3 Nesnelerin interneti (Internet of things)

Nesnelerin interneti, bir ağa bağlanabilir fiziksel cihazların oluşturduğu topluluklara verilen isimdir. Genel olarak bir cihazın nesne olarak nitelendirilebilmesi ve ağa bağlanabilmesi için bazı şartları sağlaması gerekmektedir. Bunlar; benzersiz bir kimliğe sahip olması, ağa bağlanabilir olması ve bir sensöre sahip olmasıdır.

Nesne olarak kabul edilecek her cihazın, genel veya özel fark etmeksizin bir ağa bağlanabilmesi gereklidir. Bu ağ üzerinde nesneler hem çevreyle etkileşim hâlinde olarak veri toplayacak, hem de ağ üzerindeki diğer cihazlarla iletişim hâlinde olacaklardır. Ağa bağlanacak her cihazın, IP veya MAC adresi gibi benzersiz bir kimliğe sahip olması gerekmektedir. Bu sayede ağa bağlı cihazlara erişim ve cihazların yönetimi kolay hâle gelecektir.

Nesnelerin interneti teknolojisi günümüzde akıllı şehir veya akıllı ev uygulamalarında, tarım, güvenlik, sağlık alanlarında ve enerji tüketimi yönetiminde sıklıkla kullanılmakla birlikte; akıllı saat ve akıllı gözlük gibi kişisel kullanımlarda da oldukça yaygındır.

2.1.4 Kriptografik özet fonksiyonu

Kriptografik özet fonksiyonu; gönderilen bir veriyi sabit uzunluktaki bir metne dönüştüren tek yönlü bir fonksiyondur. Bütün şifreleme mekanizmalarında, veri bütünlüğünün doğrulanması işlemlerinde ve daha birçok alanda kullanılmaktadır.

Özet fonksiyonları, aynı girdide her zaman aynı sonucu üretmektedir. Bununla birlikte bu fonksiyonlar tek yönlü oldukları için; bilinen bir özet değerinden girdi değerini çıkarmak, yani tersine işlem yapmak olanaksızdır. Gönderilen girdinin uzunluğu ne olursa olsun, kullanılan özet fonksiyonuna bağlı olarak çıktı uzunluğu 128 ile 512 bit arasında değişkenlik göstermektedir. Çizelge 2.1'de en çok kullanılan özet fonksiyonlarının çıktı uzunlukları gösterilmektedir.

Çizelge 2.1 Bazı kriptografik özet fonksiyonlarının bit ve karakter olarak çıktı uzunlukları

Özet Fonksiyonu	Çıktı Uzunluğu (Bit)	Çıktı Uzunluğu (Karakter)
MD5	128	32
RIPEMD-160	160	40
SHA-1	160	40
SHA-256	256	64

Özet fonksiyonlarındaki çıktılar on altılık sayı sisteminde (hexadecimal) oldukları için her karakter 4 bitten oluşmaktadır. Dolayısıyla B bit uzunluğunda çıktısı olan bir özet fonksiyonunun karakter olarak çıktı uzunluğu $B/4$'e tekabül etmektedir. Bununla birlikte her özet fonksiyonu, özetleme işlemi için farklı algoritmalar dizisi kullandığı için verilen aynı girdi için farklı çıktılar üretmektedirler. Çizelge 2.2'de aynı uzunlukta çıktılar veren özet fonksiyonlarının aynı girdide verdikleri farklı çıktılar gösterilmiştir.

Çizelge 2.2 Bazı kriptografik özet fonksiyonlarında verilen girdiye karşılık alınan çıktı sonuçları

Özet Fonksiyonu	**Çıktı Uzunluğu (Bit)**	**Girdi**	**Çıktı**
MD5	128	abcde	ab56b4d92b40713acc5af89985d4b786
RIPEMD-128	128	abcde	a0a954be2a779bfb2129b72110c5782d
RIPEMD-160	160	abcde	973398b6e6c6cfa6b5e6a5173f195ce3274bf828
SHA-1	160	abcde	03de6c570bfe24bfc328ccd7ca46b76eadaf4334

Özet fonksiyonu çıktılarının bu kadar uzun olmasının sağladığı faydalardan biri de özet değerlerinin çakışma ihtimalinin çok düşük olmasıdır. Verilen iki farklı girdi için aynı özet değerinin üretilme ihtimali çok düşüktür. Çıktı uzunluğu X bit olan bir özet fonksiyonunun verebileceği toplam çıktıların sayısı 2^X 'tir. Örneğin 160 bit çıktı uzunluğuna sahip SHA-1 fonksiyonunda toplam 2^{160} farklı çıktı üretilebilir.

Kriptografik özet fonksiyonlarının bir diğer özelliği ise girdide yapılan en ufak bir değişikliğin bütün çıktıyı etkilemesidir. Bu özelliğe çığ etkisi (avalanche effect) adı verilmektedir ve özellikle blok zincirinde üst bilgi veya işlemlerin üzerinde değişiklik yapılamamasının başlıca sebeplerinden biridir. Blok zinciri tarafında herhangi bir bloktaki herhangi bir işlemde yapılacak bir bitlik değişim o bloğun özet değerini tamamen değiştirecektir. Bunun sonucunda da bir sonraki bloğun özet değeri değişecek ve bu değişim olayı en son bloğa varıncaya dek sürecektir.
Çizelge **2.3**'te SHA-1 özet fonksiyonu için girdilerde yapılan değişikliklerin çıktıları nasıl değiştirdiği gösterilmiştir.

Çizelge 2.3 SHA-1 kriptografik özet fonksiyonuna verilen farklı girdiler sonucunda alınan çıktılar

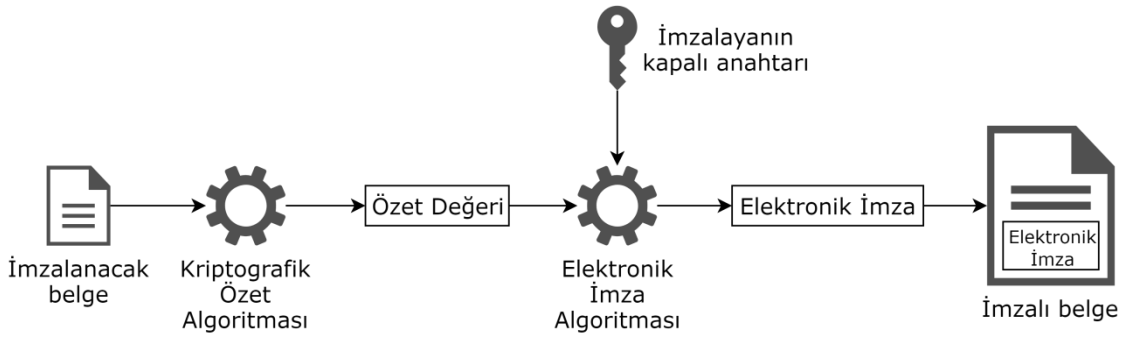
Girdi	**Çıktı**
Kriptografi	0e66ab0b5254de2aa8795d6713f38dc0affaa913
kriptografi	fb79ff9f4c0fdf257536101e29d79443a345033c
KRİPTOGRAFİ	8b1a632b1ae776d4d9668b3ccbcaa3f85ca531b2

2.1.5 Elektronik imza

Elektronik imza, bir veri veya belgenin doğruluğunu kanıtlamak amacıyla kullanılan matematiksel bir işlemdir. Bu işlem üç aşamadan oluşmaktadır ve işlem sonucunda aynı özet fonksiyonundaki gibi bir özet değeri çıkar. Bu aşamalar sırasıyla; anahtar üretimi, imzalama ve doğrulamadır.

Anahtar üretimi aşamasında RSA gibi bir açık anahtar algoritması kullanılarak kullanıcı için açık ve kapalı anahtarlar üretilir. Üretilen açık anahtar herhangi bir sorun teşkil etmeden karşıdaki kişiye gönderilebilir. Kapalı anahtar ise kişiye özel olup paylaşılmaması gerekmektedir. Üretilen açık ve kapalı anahtar ikilileri birbirleriyle bağlantılıdır ancak biri üzeriden diğerinin üretilmesi olanaksızdır.

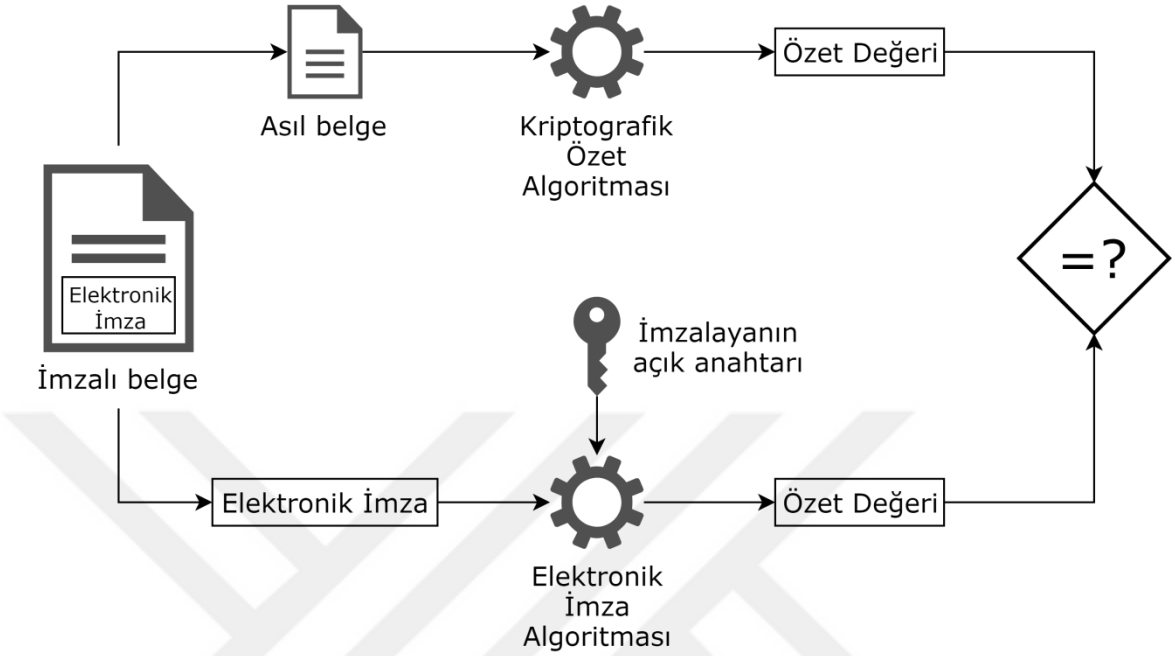
Şekil **2.3**'te de gösterildiği üzere, imzalama aşaması sırasında gönderilecek mesajın özet değeri, kriptografik özet fonksiyonu yardımıyla hesaplanır. Ardından gönderen kişi, bu özet değerini kendi kapalı anahtarıyla şifreler ve imzalama işlemi tamamlanmış olur. Mesajın yanına bu imzayla birlikte imzalayan kişinin açık anahtarı da eklenir ve alıcıya gönderilir.



Şekil 2.3 Elektronik imzalarda imzalama işlemi aşamaları

Şekil **2.4**'te ise doğrulama işleminin aşamaları gösterilmektedir. Doğrulama işlemi sırasında alıcı öncelikle göndericinin kullandığı aynı kriptografik özet fonksiyonunu kullanarak imzalanmamış mesajın özet değerini hesaplar. Ardından, gelen imzayı göndericinin açık anahtarı ile birlikte işleme sokarak ikinci bir özet değeri daha hesaplar. Hesaplanan bu iki özet değerinin aynı olup olmadığı karşılaştırılır. Bu iki

değerin aynı olması mesajın değiştirilmediğini ve mesajı imzalayanın göndericinin kendisi olduğunu kanıtlar; bunun sonucunda da doğrulama işlemi tamamlanmış olur.

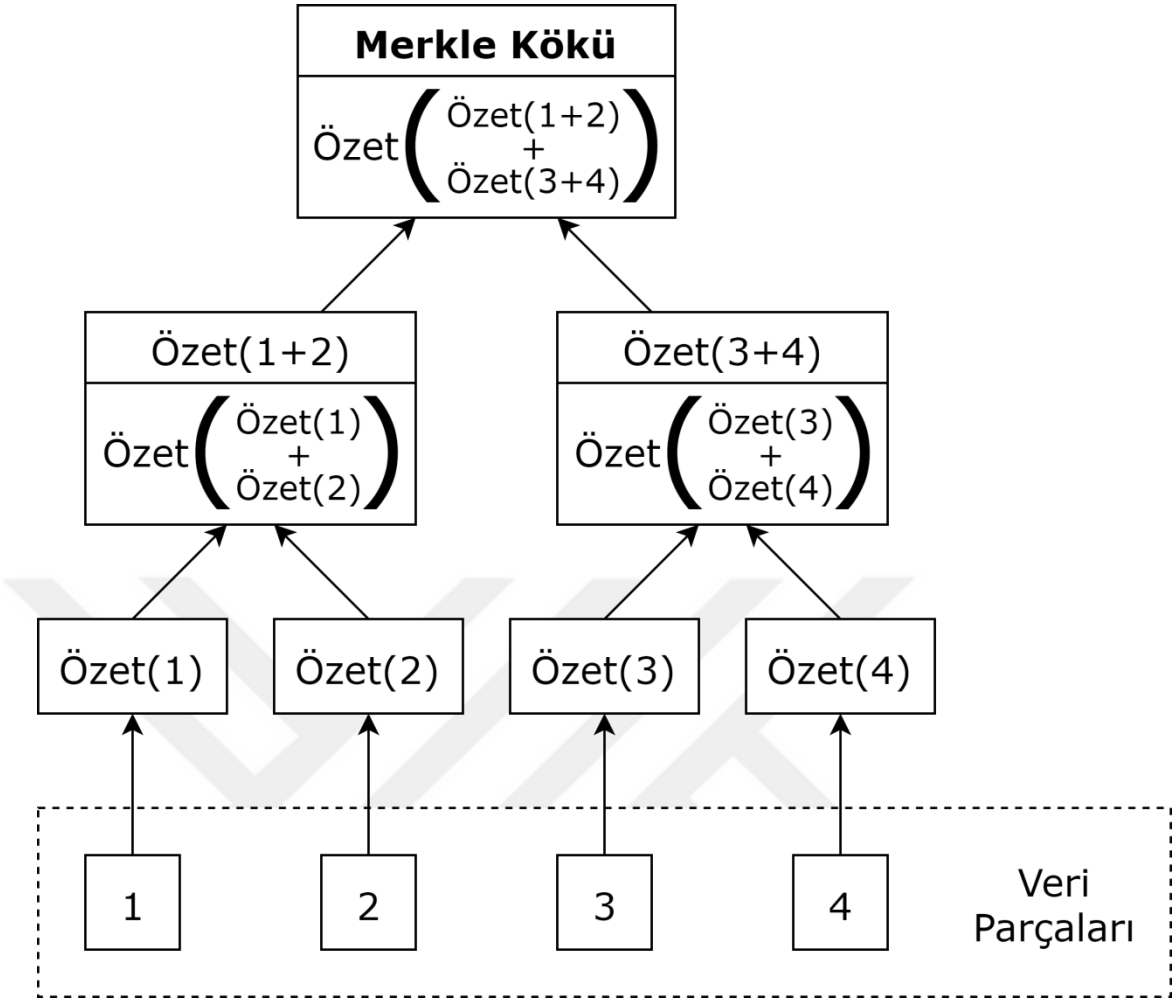


Şekil 2.4 Elektronik imzalarda doğrulama işlemi aşamaları

2.1.6 Merkle ağacı (Merkle tree)

Merkle ağacı, küçük parçalardan oluşan büyük verilerin doğrulanmasında ve güvenliğinde kullanılan bir özetleme yöntemidir. Bu yöntemin temel amacı, veriyi oluşturan küçük parçalar üzerinde düzenleme yapılmasını önleyerek verinin bütünlüğünü korumaktır. Bu işlem için kriptografik özet fonksiyonlarından faydalanılmaktadır.

Merkle ağacı oluşturulurken, öncelikle büyük veri bölünebilir küçük parçalara ayrılır. Ardından her parçanın özet değeri, kriptografik özet fonksiyonlarından biri vasıtasıyla hesaplanır. Hesaplanan bu özet değerleri ikişerli olarak birleştirilir ve bu özet değerlerinden tek bir özet değeri hesaplanır. Bu işlem ağacın köküne, yani tek bir özet değeri kalıncaya kadar devam eder. En son hesaplanan özet değerine Merkle kökü adı verilir. Şekil 2.5'te 4 parçadan oluşan bir veri için Merkle ağacının oluşturulması ve özetlerinin hesaplanarak Merkle kökünün elde edilme aşamaları gösterilmektedir.



Şekil 2.5 4 parçadan oluşan bir veri için özet değerlerinin hesaplanarak Merkle ağacının oluşturulması

Parçalarda meydana gelecek herhangi bir değişiklik parçanın özet değerini; parçanın özet değeri de zincirleme olarak üst dallardaki özet değerlerini ve dolayısıyla Merkle kökünü değiştireceği için parçalar üzerinde değişiklik yapmak olanaksızdır.

2.1.7 Akıllı sözleşme

İlk olarak Nick Szabo isimli bir bilgisayar bilimcisi tarafından 90'lı yıllarda orta atılan akıllı sözleşmeler, blok zincirinin arka tarafında otomatik olarak çalışan ve karşılıklı alışverişte kullanılan bir tür bilgisayar kodudur. Bu kod, içerisinde belirli kurallar ve doğrulamalar barındırır. Herhangi bir veri veya para aktarımında bu kodlar çalıştırılarak içerisindeki kurallar aktif olur ve bu kuralların sağlanıp sağlanmadığı kontrol edilir. Bütün kuralların sağlanması durumunda alışveriş onaylanarak aktarım sağlanır.

Akıllı sözleşmelerin en önemli özelliği, otomatik olarak çalıştırılarak üçüncü kişileri ortadan kaldırması ve işlem maliyetini düşürmesidir. Ayrıca güvenilirdir, başlayan bir sözleşme hiçbir şekilde durdurulamaz ve sisteme kurulan bir sözleşmenin üzerinde hiçbir değişiklik yapılamaz.

2.1.8 İş ispatı (Proof of work)

İş ispatı blok zinciri uygulamalarında ve özellikle kripto para sistemlerinde kullanılan; yapılan işlemlerin onaylanmasını ve blok zincirine yazılmasını, ayrıca yazımı tamamlanan bloğun ana zincire eklenmesini sağlayan bir çeşit matematiksel algoritmadır. Dwork ve Naor (1993) tarafından temel kavramları ortaya atılan bu yöntem, Jakobsson ve Juels (1999) tarafından iş ispatı adıyla anılmış ve resmîleştirilmiştir.

Yeni bir işlem yapıldığı zaman, bu işlemin onaylanması ve ardından da bloğa yazılması gerekmektedir. Bunun için sistemdeki kullanıcılar, belirli zorluktaki bir matematiksel problemi deneme yanılma yoluyla çözmeye çalışırlar. Bu işlemi yapan kullanıcılara madenci adı verilmektedir. Problemin çözümünü bulan madenci bir sonraki bloğu yazmaya hak kazanır ve belirli bir miktar sistem için para ile ödüllendirilir. Bu sayede hem sistem güvenliği ve sürdürülebilirliği sağlanır hem de kullanıcıların madencilik yapmaya devam etmeleri teşvik edilir.

Her yeni blok yazıldığında, matematiksel problemin zorluğu da gittikçe arttığı için madencilerin daha fazla olası çözüm ile deneme yapmaları gerekmektedir. Dolayısıyla iş ispatı yönteminde daha fazla olası çözümü deneyen, yani daha çok iş yapanın çözümü bulma ihtimali daha yüksektir.

2.1.9 Alan ispatı (Proof of space)

Kullanılan cihazdaki bellek alanına dayanan, ispatlayıcı ve onaylayıcı olarak adlandırılan iki kullanıcı arasında gerçekleştirilen bir protokoldür. Protokol,

ispatlayıcıya lokal belleğinde spesifik bir amaç için tahsis ettiği alanın belirlenen süre içerisinde sadece o amaç doğrultusunda değiştirilmeden korunduğunu ispat etme imkânı vermektedir. Protokol iki aşamadan oluşmaktadır. İlk aşamada ispatlayıcı N-bitlik bir veriyi lokal belleğine kaydeder ve ikinci aşamada kullanılmak üzere bu verinin bir özeti sayılabilecek veriyi lokal belleğinde sakladığına dair bir taahhüt değerini üretip onaylayıcıyla paylaşır. İkinci aşamada onaylayıcı ispatlayıcının ilgili veriyi orijinal haliyle tutup tutmadığını test etmek için rastgele bir sınama değeri oluşturur ve bu değeri ispatlayıcıyla paylaşır. Bu aşamada ispatlayıcı sınama değerine karşılık gelen ve veriyi orijinal haliyle sakladığını ispat eden bir kanıt değeri oluşturur ve bu kanıt değerini onaylayıcıyla paylaşır. Son olarak onaylayıcı ispatlayıcının ilk aşamada paylaştığı taahhüt değeriyle uyumlu bir kanıt değeri oluşturulup oluşturulmadığını kontrol ederek kanıt değerini kabul eder ya da reddeder. İspatlayıcı ve onaylayıcı arasında karşılıklı etkileşim gerektiren alan kanıtı yöntemi ilk olarak Dziembowski vd. (2015) tarafından önerilmiş bir uzlaşı algoritması olup, önerilen yöntemde kanıt oluşturma işlemlerinde Merkle ağaç yapısı kullanılmış ve bu sayede ikinci aşamada üretilen kanıt değerinin orijinal dosyaya göre oldukça düşük boyutta üretilmesi sağlanmıştır.

2.1.10 Ethereum

Ethereum, 2013 yılında Vitalik Buterin tarafından tasarlanmış açık kaynak bir blok zinciri sistemidir. Ethereum'un en temel özelliği, Bitcoin benzeri bir kripto para sistemi olmasının yanı sıra akıllı sözleşmeler ile programlanabilen bir teknolojiye sahip olmasıdır.

Ethereum içerisindeki akıllı sözleşmeler, Ethereum Sanal Makinesi (Ethereum Virtual Machine, EVM) üzerinde çalışmaktadır. EVM'nin temel amacı işlemleri ve akıllı sözleşmeleri çalıştıracak bir ortam sunarak dağıtık kayıt defterinin kullanımı sırasında karşılaşılan sınırlı işlem yeteneğini ortadan kaldırmak ve blok zincirine işlevsellik getirmektedir. Bu sanal makine bulunduğu ortamdan yalıtılmış ve bağımsız bir şekilde çalışmakta olup, dışarıdaki herhangi bir ağa veya dosya sistemine erişimi yoktur.

Ethereum'da üretilen ve kullanılan temel para birimi Ether olup, en küçük Ether birimi Wei olarak isimlendirilir ve 1 Wei 10^{-18} Ether'e eşittir. Ethereum ağı üzerinde gerçekleştirilen bütün işlemler belirli bir miktarda Ether kullanır ve bu kullanım için gereken miktara gaz (gas) adı verilir. Gazın temel amacı madencilere ödeme yapmak ve gerçekleştirilmek istenen işlemlerdeki iş miktarını sınırlandırmaktır. Gaz ücretleri (gas price) Gwei (giga-wei) birimiyle tanımlanır ve 1 Gwei 10^{-9} Ether'e eşittir. Dolayısıyla bu birimlerden, $1\ ETH = 10^{-9}\ Gwei = 10^{-18}\ Wei$ eşitliği çıkarılabilir.

Bir işlemin gerçekleştirilmesi sırasında gerekli olan gazı gönderecek kullanıcının tüketmek istediği maksimum gaz miktarına gaz limiti (gas limit) adı verilir ve standart bir transferde bu limit varsayılan olarak 21.000 birimdir. Bu limit işlem sırasında düşürülebilir veya yükseltilebilir. Gaz limitini yükselmek, belirli bir seviyeye kadar madencilerin ödülünü yükselteceği için gerçekleştirilen işlemin onay süresini kısaltacaktır. Ancak gaz limitini düşürmek, gerçekleştirilmek istenen işlemin yetersiz gaz miktarından ötürü durdurulmasına neden olabilir; bu durum da, madenciler işlemin onaylanması için gönderilen gazı harcadıklarından dolayı işlemin geri alınması esnasında kullanılan gazın işlem sahibine geri verilmemesine sebep olur. Gönderilecek gazı etkileyen bir diğer parametre ise gaz ücretidir. Gaz ücreti, gaz limitindeki birim başına düşen gazın ücretini belirtir ve yukarıda bahsedildiği üzere Gwei birimiyle belirlenir. Dolayısıyla gerçekleştirilecek işlem için ödenecek miktar, $gaz\ limiti * gaz\ ücreti$ denklemiyle hesaplanmaktadır. Örneğin gönderilecek bir işlem için gaz limitinin 21.000, gaz ücretinin ise 300 Gwei olduğu varsayılırsa, kullanıcının ödeyeceği toplam ücret $21.000 * 300 = 6.300.000\ Gwei$ yani 0,0063 Ether olarak hesaplanır. Gerçekleştirilen işlemler sonrasında gazın artması durumunda fazla gaz kullanıcının hesabına iade edilir. (Anonymous 2021b)

5 Ağustos 2021 yılında devreye giren Ethereum Londra güncellemesi ile birlikte ise, gaz ücretinin yerini temel ücret (base fee) ve bahşiş (tip) almıştır. Temel ücret, ağdaki her bir bloğun sahip olduğu ve o bloğa yazım için ödenmesi gereken minimum ücreti belirtirken bahşiş ise madencilere ödenecek işlem ücretidir. İşlemin gerçekleştirilmesi sonrasında temel ücret yanar, bahşiş ise madencilere gönderilir. Buradaki ödenecek toplam ücret de yukarıda belirtilen yönteme benzer olup, ödenecek toplam ücret

$gaz\ limiti * (temel$ ücret $+ bahşiş)$ denklemiyle hesaplanmaktadır. (ethereum.org, 2021)

2.1.11 SysML

SysML (Systems Modeling Language), geliştirilecek bir sistemin belirli seviyelerdeki tasarımlarının ve analizlerinin yapılabilmesi için kullanıcılara çeşitli diyagram yapıları sunan ve bu sayede sistemin detaylı olarak irdelenebilmesine olanak sağlayan bir tasarım dilidir. Bu dil, UML (Unified Modeling Language)'in bir uzantısı olarak 2007 yılında standart bir modelleme dili olarak literatüre sunulmuştur. Bu dilde sistem içerisindeki yapılarda kullanılacak elemanlar blok veya modül olarak isimlendirilmekte olup, bu blok veya modüller birbirlerine yönlü ve ilişkili oklarla bağlanmakta ve birbirlerine olan ilişkinin kuvveti okun türüne ve yönüne göre değişiklik göstermektedir. SysML, içerisinde barındırdığı farklı diyagramlar ile geliştirilecek sistemin bileşenlerine çeşitli açılardan bakılmasına imkân tanımaktadır. SysML sayesinde, tasarlanan sistemlerin ve sistemlerin iç yapısının paydaşlar arasında kolaylıkla anlaşılabilmesi sağlanmış ve sistem geliştirme yaşam döngüsünde karşılaşılabilecek hataların ve risklerin önceden görülmesi mümkün kılınmıştır.

2.2 Literatür Özeti

Blok zinciri yönteminin ortaya çıkışı 20. yüzyılın sonlarına dayansa da, tanınması ve etkin olarak kullanılması son 10 yıl içerisinde gerçekleşmiştir.

Blok zinciri teknolojisinin ilk örneği olarak kabul edilen çalışma, 1991 yılında gerçekleştirilmiştir. Haber ve Stornetta (1991) dijital dokümanların zaman damgalarının (timestamp) değiştirilmesini önlemek amacıyla kriptografik özet fonksiyonlarını da kullanan iki farklı çözüm önermişlerdir. Haber ve Stornetta'ya göre dijital bir dokümanı zaman ile damgalamak için iki farklı özelliğe ihtiyaç vardır. Bunlardan birincisi ortamın özelliklerinden bağımsız olarak bir belgenin kendisini zaman ile damgalamak, bu sayede değişiklik açıkça yapılmadığı sürece belgedeki bir bitin bile değiştirilmesini imkânsız hâle getirmektir. İkincisi ise bir belgenin farklı bir zaman ile damgalanmasını

önlemektir. Önerdikleri ilk çözüm, damgalanacak belgelerin özet değerlerinin bir zaman damgalama servisine gönderilmesi ve bu değerlerin birbirine bağlanması; ardından bu bağlantıları tutan bir sertifikanın sistemdeki diğer alıcılara dağıtılmasını öngören bir sistemdir. İkinci yöntem ise, her belgenin rastgele seçilen belirli kişiler tarafından damgalanmasını sağlayarak; bir belgenin zaman damgasında değişiklik yapılmaya çalışıldığında o belgeyi hangi kullanıcıların damgalayıp hangilerinin damgalamadığını bulmayı imkânsız hâle getirmektir.

Blok zincirine dair en büyük ve en ses getiren çalışmaların başında ise Nakamoto (2008) tarafından gerçekleştirilen Bitcoin isimli kripto para sistemi gelmektedir. Nakamoto'ya göre elektronik ödemelerde güvene dayalı modeller, sistemde içsel bir zayıflık oluşturabilmekle birlikte aracılar da bu sisteme ek maliyet getirmektedir. Buna göre ihtiyaç duyulan sistem, herhangi bir tarafın güvenilir bir üçüncü tarafa ihtiyaç duymadan karşı taraf ile işlem yapmasına izin veren ve güven yerine şifrelemeye dayanan bir elektronik ödeme sistemidir. Nakamoto bu sistemde elektronik imza ve kriptografik özet fonksiyonlarından faydalanmış ve iş ispatında da Back'in (2002) Hashcash mekanizmasını temel almıştır.

Bitcoin'in ortaya çıkışının ardından benzer kripto para sistemleri ortaya çıkmış ve sayıları da kısa zamanda artmıştır. Wood (2014), çıkardığı Ethereum platformu ile akıllı sözleşmeleri de destekleyen bir blok zinciri sistemi ortaya atmıştır. Wood'un sistemindeki temel amaçlardan biri, belirli zorluklar sebebiyle birbirine güvenmeyen iki kullanıcı arasındaki işlemlerin gerçekleştirilmesini kolaylaştırmaktır. Kullanılacak akıllı sözleşmelerin kodları, Merkezi olmayan Ethereum Sanal Makinesi'ni kullanarak çalıştırılmaktadır. Ethereum Sanal Makinesi, verilen işlem kodlarını (opcode) sırasıyla çalıştıran bir sistemdir. Akıllı sözleşmelerin yazıldığı Solidity gibi programlama dilleri EVM tarafından desteklenmediği için işlem kodu adı verilen alt seviye makine komutlarına dönüştürülmesi gerekmektedir (Hollander 2019).

2.2.1 Blok zinciri ile veri yönetimi

Blok zincirinin merkezî olmayan bir veri tabanı sistemi olmasından ötürü bu teknoloji verilerin depolanması ve yönetilmesinde de kullanılmıştır. Zyskind vd. (2015) kullanıcıların kendi verilerini almalarını ve kontrol etmelerini sağlayan ve merkezi olmayan bir kişisel veri yönetim sistemi önermişlerdir. Platform, blok zincirini otomatikleştirilmiş erişim denetleyicisine dönüştürerek üçüncü partilere olan ihtiyacı ortadan kaldırmıştır. Bununla birlikte işlemler, kripto para sistemlerindeki gibi finansal veri tutmamakta; onun yerine kişisel verilerin depolanması, sorgulanması ve paylaşılmasına dair komutları barındırmaktadır.

Azaria vd. (2016) sağlık sistemindeki elektronik tıbbi kayıtların yönetimini kolaylaştıracak, merkezi olmayan bir sistem önermişlerdir. Önerilen bu sistemde hastalara geniş kapsamlı ve değiştirilemez bir kayıt defteri sunulmuştur. Hastaların kişisel bilgilerini Ethereum adreslerine kaydetmişler ve sağlık verilerini kayıt altında tutmak için referanslar kullanarak bu referansları da blok zincirine eklemişlerdir. Sağlık sisteminde bulunan araştırmacıları madenci olarak kullanarak sistemin devamını sağlamışlar ve araştırmalarında kullanmaları için de anonim kullanıcı verileri ile ödüllendirmişlerdir.

Wilkinson vd. (2016) Storj adını verdikleri bir eşler arası (peer-to-peer) bulut depolama ağı tasarlamışlardır. Normalde eşler arası sistemler veri depolama için uygun değildir, çünkü bu sistemlerde verilerin depolanma sebebi hizmetten ziyade verinin popülerliğidir. Wilkinson vd. bu sistemde bir talep-yanıt sistemi oluşturarak, bu yöntem ile veri bütünlüğünü belirli zaman aralıklarında kontrol etmeyi planlamışlardır. Veri bütünlüğünü sağlayarak verileri koruyan kullanıcılara da ödül sunmayı amaçlamışlardır. Depolanacak verileri kriptografik özet fonksiyonu ile şifreledikten sonra belirli sayıda parçalara ayırmışlardır. Veri bütünlüğünü kontrol ederken ise Merkle ağacından faydalanmışlardır.

Liang vd. (2017) bulut sistemleri için blok zinciri tabanlı bir veri kanıtı sistemi önermişlerdir. Bulut verisi kanıtı, bir bulut verisinin oluşturulmasından itibaren üzerinde

gerçekleştirilen işlemlerin bütününü içeren bir üst bilgidir. Sistem; verilerin toplanması, depolanması ve doğrulanması olmak üzere 3 temel aşamadan oluşmaktadır. Önerilen blok zinciri tabanlı veri kanıtı, izinsiz değişikliklere karşı korumalı kayıtlar sağlamakta, veri sorumluluğunu daha şeffaf hâle getirmekte ve veri kanıtların gizliliği ile geçerliliğini artırmada yardımcı olmaktadır.

Goharshady vd. (2018) kullanıcıların kredi geçmişlerini güvenilir bir şekilde muhafaza etmek ve raporlamak için blok zinciri tabanlı bir platform önermişlerdir. Platform, kredi kayıt büroları gibi üçüncü tarafları kredi sürecinden kaldırarak bunları akıllı sözleşmelerle değiştirmektedir. Bir yandan müşterilerin kredi verilerinin bütünlüğünü, güvenilirliğini ve gizliliğini sağlarken diğer yandan da müşterilerin doğrudan borç verenlerle veya bankalarla etkileşime girmelerini sağlamaktadır. Kullanıcılara, kendi kredi raporlarını paylaşma izni verilmekte ancak değiştirme izni verilmemektedir. Bu sayede kredi verenlerin görebildiği kredi raporlarının tamamen doğru ve değiştirilemez olması sağlanmaktadır.

Huang vd. (2018) medikal sistemler için ilaç izlenebilirliği ve düzenliliği sağlayan blok zinciri tabanlı eşler arası mimariyi kullanan bir ilaç tedarik zinciri önermişlerdir. Sistemin verimliliğini en yüksek oranda tutmak için blok zinciri depolamasını ilaçların son kullanma tarihini göre azaltmışlardır. Önerilen sistem; servis sağlayıcıları, paydaşları ve hastaları ortak bir çevrede birleştirerek sürdürülebilir bir mimari sunmuştur.

2.2.2 Blok zinciri ve nesnelerin interneti

Blok zincirinin bilinirliği ve kullanım sıklığı arttıkça, bu teknoloji farklı alanlarda da kendine yer edinmiş ve son zamanlarda nesnelerin internetinde kullanılmaya başlanmıştır.

Biswas ve Muthukkumarasamy (2016) akıllı şehirlerdeki sistemlerin güvenilir bir şekilde iletişim kurabilmelerini sağlamak için blok zinciri teknolojisini kullanan bir güvenlik sistemi önermişlerdir. Akıllı cihazlarda; fiziksel katman, iletişim katmanı,

veritabanı katmanı ve arayüz katmanı bulunmaktadır. Biswas ve Muthukkumarasamy'e göre, kayıtları tutan veritabanı katmanı dağıtık kayıt defteri altyapısı kullanacak ve açık veya kapalı olabilecektir. Blok zincirinin açık olması, sistemi şeffaf ve sansürlenemez yapmasına rağmen uzlaşmaya olan erişim süresini uzatmakta ve sistemi anonim saldırılara açık hâle getirmektedir. Dolayısıyla ölçeklenebilirliği, performansı ve gerçek zamanlı güvenliği sağlamak için kapalı blok zincirinin kullanılması önerilmektedir.

Lee ve Lee (2017) IoT ortamındaki gömülü cihazlar için güvenilir bir aygıt yazılımı (firmware) güncelleme sistemi önermişlerdir. Bir aygıt yazılımının sürümünü güvenli bir şekilde kontrol etmek, yazılımın doğruluğunu onaylamak ve gömülü cihazlar için en son yazılımı indirmek için blok zinciri teknolojisi kullanmışlardır. Buna göre bir cihaz, kendi yazılımının güncel olup olmadığını kontrol etmek için ağdaki diğer düğümlere istek göndererek karşılığında bir yanıt almaktadır. Aygıt yazılımının güncel olmaması durumunda yazılımın son sürümü, düğümlerden oluşan eşler arası bir yazılım paylaşım ağından indirilmektedir. Yazılımın güncel olması durumunda ise yazılımın bütünlüğü, dolayısıyla yazılımın doğruluğu kontrol edilmekte ve bu sayede cihazın aygıt yazılımının değiştirilmediği sürece güncel olduğunu garanti edilmektedir.

Huh vd. (2017) Ethereum platformunu kullanarak IoT cihazlarını kontrol etmek için bir sistem önermişlerdir. Bir sayaç aracılığıyla gönderilen elektrik kullanım bilgileri ve akıllı telefon aracılığıyla gönderilen cihaz ayarları akıllı sözleşmeler aracılığıyla kaydedilmekte ve cihazlar bu verileri Ethereum üzerinden belirli aralıklarla okuyarak kendilerini güncelleyebilmektedir. İlk denemelerini ampul ve klima üzerinde yaparak ampulün yanıp yanmama durumu ile klimanın enerji kiplerini kontrol etmişlerdir.

Sagirlar vd. (2018) IoT için hibrit bir blok zinciri mimarisi önermişlerdir. Önerilen bu sistemde IoT cihazlarını blok zincirleri içinde gruplayarak alt blok zincirler oluşturmuşlardır. İş ispatını kullanan bu alt blok zincirlerini ise birbirleriyle iletişim kurar hale getirmişlerdir.

Dorri vd. (2019) akıllı evlerdeki IoT cihazları için katmanlı ve ölçeklenebilir bir blok zinciri mimarisi önermiştir. Önerilen mimaride iki katman bulunmaktadır. Alt katmanda

kısıtlı kaynaklara sahip cihazlar kapalı bir blok zincirini yönetmekte ve açık anahtarlar yardımıyla iletişimi sağlamaktadırlar. Üst katmanda ise kaynakları geniş cihazlar açık bir blok zincirini yöneterek uçtan uca (end-to-end) olan gizliliği ve güvenliği sağlamaktadırlar.

2.2.3 Blok zinciri ile veri ticareti

Nesnelerin interneti teknolojisinin kullanım alanları genişledikçe, bu cihazlar veri toplama amaçlı da kullanılmaya başlanmış, bu durum da işlenebilir verilerin pazarlanabilmesi ihtiyacını ortaya çıkarmıştır.

Bu amaçla Huang vd. (2017) IoT verilerinin pazarlanabilmesini sağlayan Ethereum blok zinciri tabanlı katmanlı bir mimari önermişler ve verileri işlemlerin kaydedildiği blok zinciri ile aynı katman içerisinde bulunan farklı bir depolama sistemine (bulut depolama, veri tabanı vb.) kaydederek performansın artırılmasını amaçlamışlardır. Bununla birlikte veri sahiplerinin, platforma veri yükledikleri sırada veriye ait üst bilgilerin de eklenmesini sağlayarak alıcıların veri hakkında bilgi sahibi olmalarını da sağlamışlardır.

Kiyomoto ve Fukushima (2018) taraflara adil pazarlama olanağı sunan bir sistem önermişlerdir. Sistem, veri almak isteyen veri analistleri ile bu verileri analistlere gönderecek olan veri aracıları arasında adil bir fiyatlandırma yapılmasını sağlamaktadır. Aracılar, ellerindeki verileri şifreli bir biçimde depolama alanına yüklemekte ve yüklenen bu veriler parçalara ayrılarak her bir parçanın doğrulama etiketi hesaplanmakta ve kaydedilmektedir. Analistler, almak istedikleri verileri analiz edebilmek maksadıyla veri setinden görmek istedikleri rastgele bir kaydı aracıdan istemekte ve gelen parçanın doğruluğunu, depolama alanından gelen ilgili doğrulama etiketi vasıtasıyla kontrol etmektedir. Ayrıca verinin fiyatlandırılması aşamasına geçilebilmesi, ancak bu işlemin belirli bir sayıda (en fazla veri setindeki kayıt sayısı kadar olmak üzere) yapılması sonucunda gerçekleştirilmektedir.

Guan vd. (2018) büyük verilerin ve bu verilerin istatistiksel bilgilerinin pazarlanabilmesini sağlayan bir sistem önermiştir. Sistemde gerçek zamanlı sağlık verileri kullanılmıştır. Sistem içerisinde veriler işlenmeden önce küçük parçalara ayrılmış ve her bir parçanın özet değeri hesaplanarak verilerin değiştirilmesinin önüne geçilmiştir. Ayrıca bu yolla alıcıların, satın aldıkları verileri özet değerleri vasıtasıyla kontrol edebilmelerine de imkân tanınmıştır.

Sabounchi vd. (2018) alıcılara, almak istediği verinin özelliklerini ve ödemek istediği miktarı içeren teklifler oluşturarak bu teklifleri ilgili satıcılara iletebilmesine olanak sağlayan bir sistem önermişlerdir. İletilen bu teklifler sonrasında akıllı sözleşmeler vasıtasıyla satıcın seçtiği tekliflerden biri üzerinden satış işlemi gerçekleştirilmektedir.

Wang vd. (2018) iki farklı blok zinciri kullanan bir veri paylaşım modeli önermişlerdir. Önerilen modelde blok zincirlerinden biri verilerin tamamını depolamada, diğeri ise yapılan işlemlerin saklanmasında kullanmıştır. Veriler öncelikle küçük parçalara ayrılarak her bir parçanın özet değeri hesaplanmış, ardından bu parçalar şifrelenerek kendi özet değerleri ile birlikte veri blokları hâlinde verilerin depolandığı blok zincirine kaydedilmiştir. Verilerin parçalara ayrılarak özet değerlerinin hesaplanması, verilerin değiştirilemez bir biçimde depolanmasına olanak sağlamış ve sistemin güvenilirliğini artırmıştır.

Papadodimas vd. (2018) ise sensör verisi olarak hava durumu bilgilerinin pazarlanabildiği izinli blok zinciri kullanan ve yukarıda bahsi geçen yapılara ek olarak, alışveriş esnasında kullanılabilecek sistem içi yeni bir para birimi öneren bir platform sunmuşlardır. Ethereum altyapısı kullanıldığı için sistem, sistem içi oluşturulan para ile Ethereum platformunda kullanılan Ether arasında değişiklik yapılabilmesini de desteklemektedir.

An vd. (2019) mobil cihazlardan elde edilen kitle yoğunluğu (crowdsensing) verilerinin pazarlanabildiği bir sistem önermişlerdir. Sistemde akıllı sözleşme tabanlı araçlar alıcılardan gelen talepleri satıcılara ileterek satıcılardan fiyat teklifi almaktadır. Alınan bu teklifler arasından, ters açık artırma yöntemi ile uygun satıcılar belirlenmekte ve

alıcılar ile iletişime geçirilmektedir. Satın alınan şifreli veriler bulut sunucuları üzerinden, şifreler de akıllı sözleşme tabanlı yardımcılar tarafından güvenli bir yol ile alıcılara gönderilmektedir.

Tzianos vd. (2019) tarafından önerilen izinsiz blok zinciri tabanlı sistemde; satıcılar ellerindeki sensörlerin bilgilerini, alıcılar da istedikleri veri tipini market üzerinden yayınlayabilmektedir. Pazarlama işlemi, istenen sensörün verilerine belirli bir süre ulaşabilme hakkı tanımaktadır. Alıcıların satın aldığı sensöre ait veriler bir depolama alanına belirli bir formatta şifreli olarak kaydedilmektedir. İstenildiği durumda bu alan üzerinden veri anahtarları vasıtasıyla şifresi açılarak verilere ulaşılabilmektedir.

Manzoor vd. (2019) de benzer bir sistemi Ethereum tabanlı izinli blok zincirini kullanarak oluşturmuşlar ve veri kaydı esnasında verileri ayrı bir depolama alanı olarak bulut depolamaya kaydetmişlerdir. Bunun yanı sıra verileri satıcının anahtarı ile şifreleyerek veri güvenliğini de sağlamışlardır. Verilerin satışı gerçekleştirildiğinde ise bu şifre açılarak bu sefer alıcının açık anahtarına göre şifreleme yapılmış ve verilere bulut depolama üzerinden alıcının kendi kapalı anahtarı ile erişebilmesi sağlanmıştır.

Chen vd. (2019) de araçların interneti (internet of vehicles) ağına bağlı cihazlardan elde edilen veriler için önermişlerdir. Alıcılar veri ihtiyaçlarını aracılara iletmekte, aracılar ise bu ihtiyaçları satıcılara ileterek fiyat teklifi almakta ve uygun fiyatlı teklifi tekrar alıcıya bildirmektedir. Aracılar akıllı sözleşme tabanlı olup, sadece veri ihtiyacının ve fiyat teklifinin karşı tarafa iletilmesinden sorumludur. Dolayısıyla paranın ve verinin takasında herhangi bir aracı taraf bulunmamakta, alıcı veriyi aldıktan sonra parayı direkt olarak satıcıya göndermektedir.

Dai vd. (2019) ise alıcının, almak istediği veriyi analiz algoritmaları vasıtasıyla ölçebildiği bir sistem önermişlerdir. Bu sistemde alıcı elindeki analiz algoritmalarını, satıcı da elindeki veriyi daha önceden belirlenmiş güvenilir kullanıcılara göndermektedir. Bu kullanıcılar analiz algoritmaları üzerinden elde ettikleri sonuçları tekrar alıcıya göndererek alıcının veri hakkında bilgi sahibi olmasını sağlamaktadırlar.

Elbüz vd. (2019), verilerin alıcılar tarafından incelenebilmesine olanak sağlayan ve izinsiz blok zinciri kullanan bir ticaret platformu önermişlerdir. Sistemde satıcılar, verilerini anlamlı küçük veri setlerine bölmekte ve bu veri setleri üzerinden bir özet değeri hesaplayarak blok zincirine yüklemektedir. Alıcılar ise satın almak istedikleri veriyi oluşturan veri setleri arasından rastgele bir sayı seçmekte ve bu sayıya denk gelen veri setini alıcıdan alıp inceleyebilmektedirler. Alışveriş işlemi, satın alınacak verinin özet değerinin blok zincirindeki özet değeriyle aynı olduğu durumda gerçekleştirilmekte ve bu sayede verinin değiştirilmediği garantilenmektedir. Kiyomoto ve Fukushima'nın çalışmasından farklı olarak bu çalışmada veriler herhangi bir depolama alanına yüklenmemekte, bütün veriler satıcıların kendi depolama sistemlerinde bulunmaktadır. Ayrıca satılacak verinin fiyatlandırılmasına dair de herhangi bir işlem yapılmamaktadır.

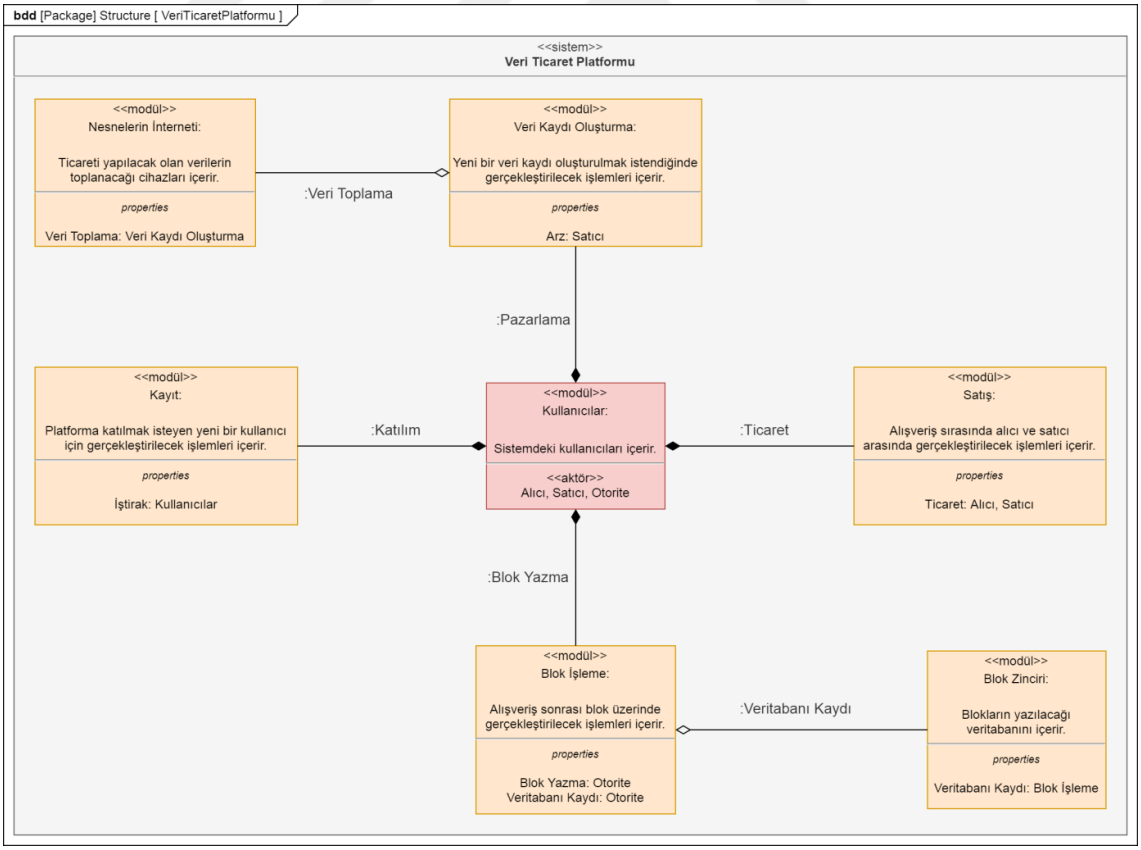
Zheng vd. (2020) ise başarılı alışveriş işlemleri sonrasında veri sahibine ödül verilmesini sağlayan Ethereum tabanlı kapalı bir blok zinciri sistemi önermişlerdir. Önerilen bu sistemde alıcı taraf veri ihtiyaçlarını yayınlamakta ve yukarıda bahsedilen diğer sistemlerden farklı olarak veri dağıtıcıları olarak adlandırılan kullanıcılar bu ihtiyaçlar doğrultusunda veri sahiplerinden aldıkları verileri alıcılara iletmektedir. Alıcı, satın aldığı verinin kullanım bilgilerini zincire kaydetmekte, bu sayede verilerin kalitesinin artmasına katkı sağlamaktadır. Bununla birlikte sistem, alıcının satın aldığı verinin kalitesine ve miktarına göre veri sağlayıcısına ödül vermektedir.

3. SİSTEM MODELİ VE SENARYOSU

Bu bölümde nesnelerin internetine bağlı cihazlardan toplanan verilerin alınıp satılabildiği blok zinciri tabanlı bir veri ticaret platformunun yapısından, bileşenlerinden ve böyle bir platformun geliştirilmesi esnasında kullanılan teknolojiler ile yöntemlerden bahsedilecektir.

3.1 Sistem Modeli

Bu bölümde, nesnelerin internetine bağlı cihazlardan toplanan verilerin ticaretinin yapılabileceği blok zinciri tabanlı bir veri ticaret platformunun SysML diyagramları yardımıyla detaylı açıklaması yapılacaktır. Platformda her bir bileşen ayrı bir modül olarak ele alınacak ve bileşenler arasındaki ilişkiler oklar yardımıyla belirtilecektir.



Şekil 3.1 Platformun genel yapısı

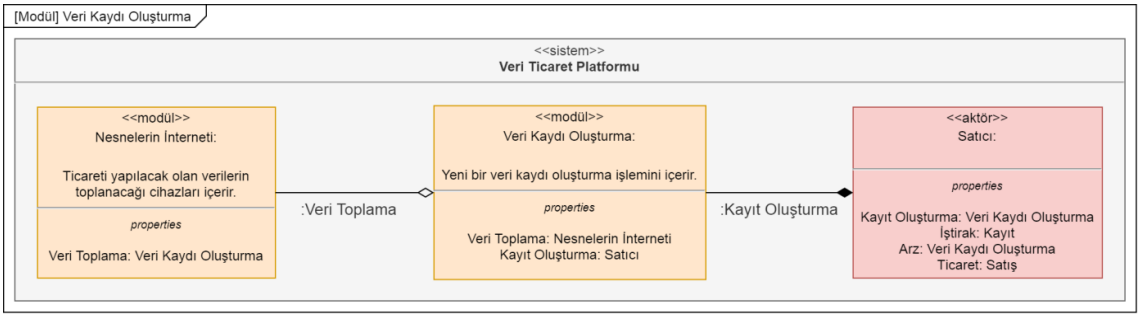
Şekil 3.1'de de gösterildiği gibi blok zinciri tabanlı veri ticaret platformu; **Kullanıcılar**, **Kayıt**, **Nesnelerin İnterneti**, **Veri Kaydı Oluşturma**, **Satış**, **Blok İşleme** ve **Blok Zinciri** olarak adlandırılan 7 farklı modülden oluşmaktadır. **Kullanıcılar** modülü platform içerisinde yer alacak rolleri belirlemekte olup alıcı, satıcı ve otorite olmak üzere 3 farklı rol içermektedir. Satıcılar, **Nesnelerin İnterneti** modülünden elde ettikleri verileri platform üzerinden pazarlayan ve satan kullanıcılardır. Alıcılar, platform üzerinden **Nesnelerin İnterneti** yoluyla toplanan verileri satın almak isteyen kullanıcılardır. Otoriteler ise, platformda paylaşılan işlemleri toplayan ve blok zincirine kaydeden kullanıcılardır.

3.1.1 Kayıt

Platform içerisinde alıcı, satıcı veya otorite olarak yer alacak her bir kullanıcının öncelikle **Kayıt** modülü üzerinden platforma kaydolması gerekmektedir. Bölüm 2'de de belirtildiği gibi blok zincirleri izinli ve izinsiz olmak üzere ikiye ayrılmakta olup; izinsiz blok zincirlerinde kullanıcılara kayıt için herhangi bir şart koşulmazken, izinli blok zincirlerinde sadece otoriteler tarafından izin verilen kullanıcılar sisteme kayıt olabilmektedir. Bu çalışmada, platform izinsiz blok zinciri üzerine inşa edilmiş bir yapı olarak ele alınmaktadır. Platform izinli blok zinciri kullanılarak hayata geçirilecekse bu modülün daha dikkatli ve detaylı tasarlanması gerekmektedir.

3.1.2 Veri kaydı oluşturma

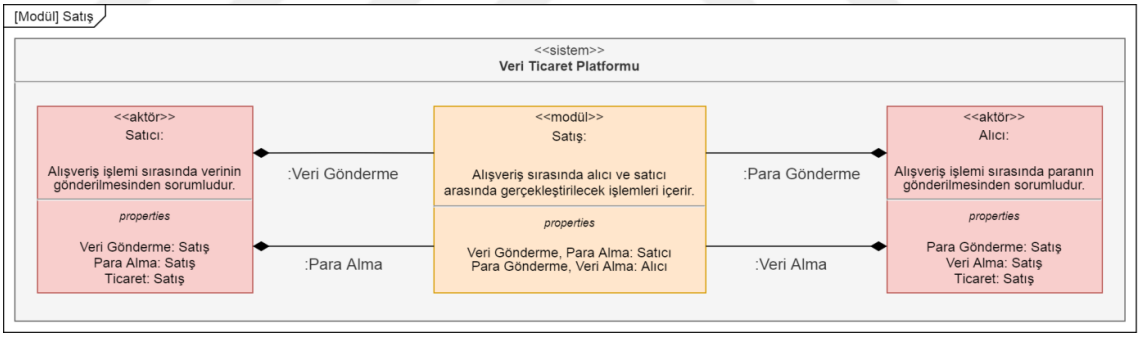
Satıcıların platform içerisinde veri satışı gerçekleştirebilmeleri için, öncelikli olarak ilgili verilerle alakalı veri kaydı oluşturmaları gerekmektedir. Bu doğrultuda satıcılar, Şekil 3.2'de de gösterildiği gibi **Nesnelerin İnterneti** modülü üzerinden elde ettikleri IoT verileri için **Veri Kaydı Oluşturma** modülü yoluyla, alıcıların veri ile ilgili gerekli bilgileri bulabileceği bir veri kaydı oluştururlar. Bu veri kayıtları ilgili verilerle alakalı gereken açıklamaları içermesinin yanında; ihtiyaca yönelik, örneğin Elbüz vd. (2019)'de olduğu gibi güvenli bir alışveriş için gerekli olan ekstra bilgiler de içerebilmektedir.



Şekil 3.2 "Veri Kaydı Oluşturma" modülü

3.1.3 Veri alışverişi

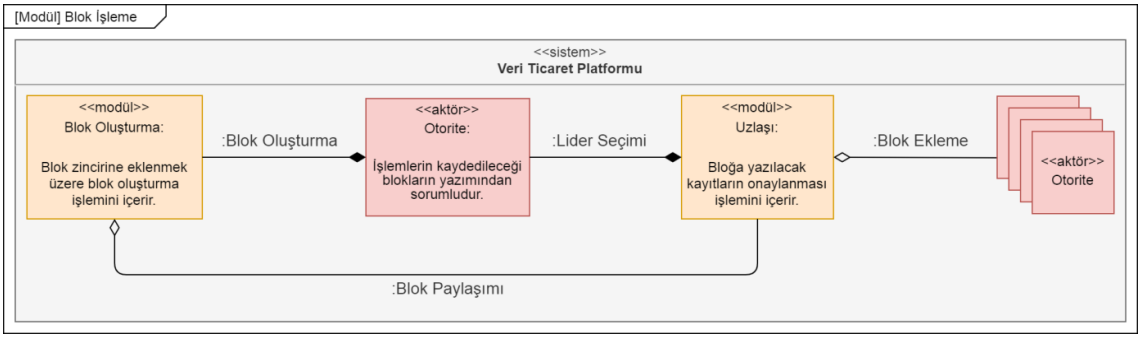
Alıcılar **Veri Kaydı Oluşturma** modülü üzerinden platforma kaydedilen verilerden birini satın almak istediklerinde, öncelikle **Satış** modülü vasıtasıyla ilgili satıcı ile irtibata geçerler. Taraflar alışverişin şartları üzerinde uzlaşmaya varırlarsa, Şekil 3.3'te de görüldüğü gibi, yine aynı modül üzerinden ilgili verinin ve uzlaşılan paranın takası gerçekleştirilir.



Şekil 3.3 "Satış" modülü

3.1.4 Blok işleme

Şekil 3.4'te resmedildiği gibi, tasarlanan modelde platformda paylaşılan işlemler periyodik olarak **Uzlaşı** modülü üzerinden belirlenmiş otoritelerce toplanarak yeni bloklara yazılmakta ve oluşturulan bu yeni bloklar platforma kayıtlı otoritelerin onayıyla zincire eklenmektedir.

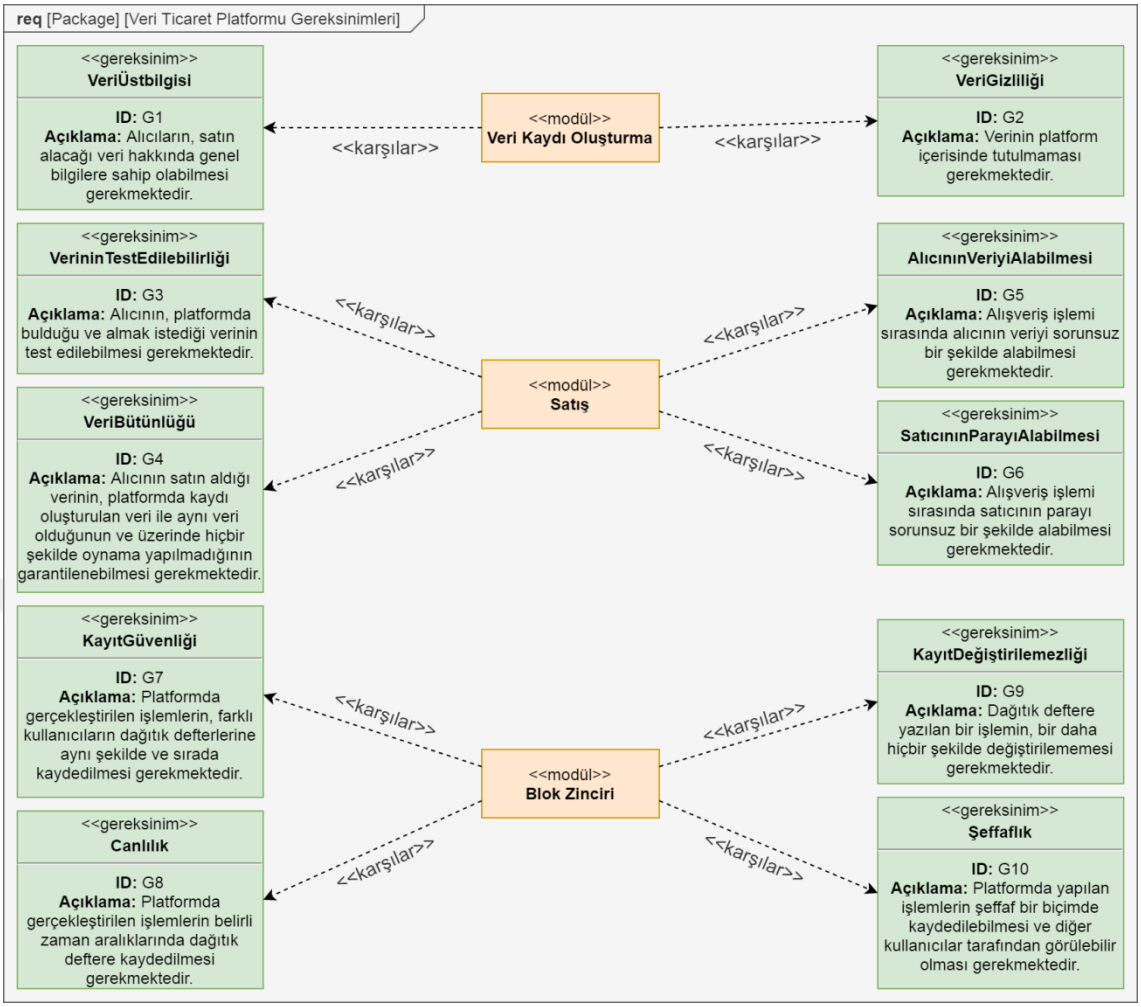


Şekil 3.4 "Blok İşleme" modülü

Biraz detaylandırmak gerekirse; bu aşamada, önce **Uzlaşı** modülü yoluyla bloğu oluşturacak lider belirlenir. Lider belirleme süreci platformdaki otoritelerin mutabık kaldığı belirli bir uzlaşı algoritması üzerinden yürütülmektedir. Uzlaşı algoritması üzerinden belirlenen lider, ilgili periyot esnasında platformda paylaşılmış olan işlemleri içeren yeni bir blok oluşturur. Sonrasında lider yeni bloğu platformdaki diğer otoritelerle paylaşır ve yine **Uzlaşı** modülü üzerinden diğer otoritelerce onaylanan bu blok kriptografik özet fonksiyonları yardımıyla değiştirilemez bir şekilde zincire eklenir.

3.1.5 Gereksinimler

Bu bölümde platformun doğru ve sorunsuz bir şekilde idame ettirilebilmesi için ihtiyaç duyulan gereksinimlerden bahsedilecektir. Karşılanması gereken bu gereksinimler Şekil 3.5'te bulunan gereksinim diyagramında gösterilmektedir. Diyagramda gereksinimler ve ilgili modüller arasındaki ilişki standart yönlü kesikli çizgiler vasıtasıyla belirtilmiştir.



Şekil 3.5 Veri ticaret platformu gereksinim diyagramı

Satışı yapılacak olan veriler platform içerisinde orijinal hâlleriyle tutulursa, sistem şeffaf olduğu için sistemdeki kullanıcılar para ödemeksizin bu verilere erişebilirler. Bundan dolayı verilerin sistem içerisinde gizli bir şekilde tutulması gerekmektedir. Bununla birlikte, veri kaydı oluşturulurken verilerin platformda daha rahat taranabilmesi ve alıcıların satın almak istedikleri verileri kolayca değerlendirebilmesi için, veriyle alakalı bir üst bilginin platformda paylaşılması gerekmektedir.

Kötü niyetli satıcıların, alıcıları kandırmak amacıyla platforma işe yaramaz veriler yükleme ihtimali bulunmaktadır. Bundan dolayı sistemin alıcılara, veri satın almadan önce bu verileri test edebilme imkânı sunması gerekmektedir. Bunun yanı sıra kötü niyetli bir satıcı, satış işlemi sırasında platforma yüklediği veri üzerinde değişiklik yaparak alıcının mağdur olmasına sebep olabilir. Dolayısıyla bu noktada veri

bütünlüğünün sağlanması gerekmektedir. Diğer bir ifadeyle, bir alıcının satın almak istediği verinin, başlangıçta platforma yüklenen veri ile aynı veri olduğunun ve üzerinde herhangi bir değişiklik yapılmadığının garantilenmesi gerekmektedir. Ayrıca, alıcı gerekli ücreti ödedikten sonra verinin tamamını; benzer şekilde satıcı da verinin tamamını alıcıya gönderdikten sonra paranın tamamını güvenilir ve sorunsuz bir şekilde alabilmelidir.

Dağıtık kayıt defteri, belirli bir ağ veya ağ grubu üzerinde dağıtılmış olan ortak bir veri tabanı sistemidir. Ağdaki her kullanıcı bu defterin lokal bir kopyasına sahip olabilir ve defter üzerinde yapılan değişiklikler uzlaşı protokolleri vasıtasıyla belirli aralıklarla bu kopyaların tamamına yansıtılmaktadır. Sağlam bir dağıtık kayıt defterinin güvenlik ve canlılık olmak üzere iki temel gereksinimi karşılaması beklenir. İlk gereksinim, deftere yazılacak işlemlerin ağdaki kullanıcılar tarafından ortak bir mutabakata varılarak yazılacağını belirtirken; ikincisi, doğru bir şekilde gerçekleştirilmiş bir işlemin ağdaki kullanıcılar tarafından er ya da geç onaylanarak deftere yazılacağını belirtmektedir. Bu çalışmada dağıtık kayıt defteri sisteminin blok zinciri teknolojisi üzerinden gerçekleştirildiği varsayılmaktadır.

Kötü niyetli bir kullanıcı kendi adına menfaat sağlamak için, dağıtık defter içerisinde yapılmış bir ticaretin kayıtları üzerinde oynayarak o ticaret hiç yapılmamış gibi gösterebilir. Benzer şekilde hiç yapılmamış bir ticaretin de, dağıtık defterdeki kayıtlar üzerinde oynanarak yapılmış gibi gösterilmesi mümkündür. Bu tür senaryolara karşı sistemi daha dirençli kılmak için dağıtık deftere yazılan bir işlemin, değiştirilemez bir şekilde saklanması gerekmektedir. Bu sayede kötü niyetli kullanıcıların sistemdeki kayıtları değiştirerek kendilerine menfaat sağlamasının önüne geçilecektir. Benzer şekilde kötü niyetli bir kullanıcı, otoritelerle anlaşarak işlemlerin eksik kaydedilmesini sağlayabilir ve bunun sonucunda alıcı veya satıcının itibarının zedelenmesine sebep olabilir. Bu sebeple platformda gerçekleştirilen işlemlerin şeffaf bir biçimde kaydedilebilmesi gerekmektedir. Bu sayede kaydedilen işlemler diğer kullanıcılar tarafından incelenebilir olacak ve platforma duyulan güven artacaktır.

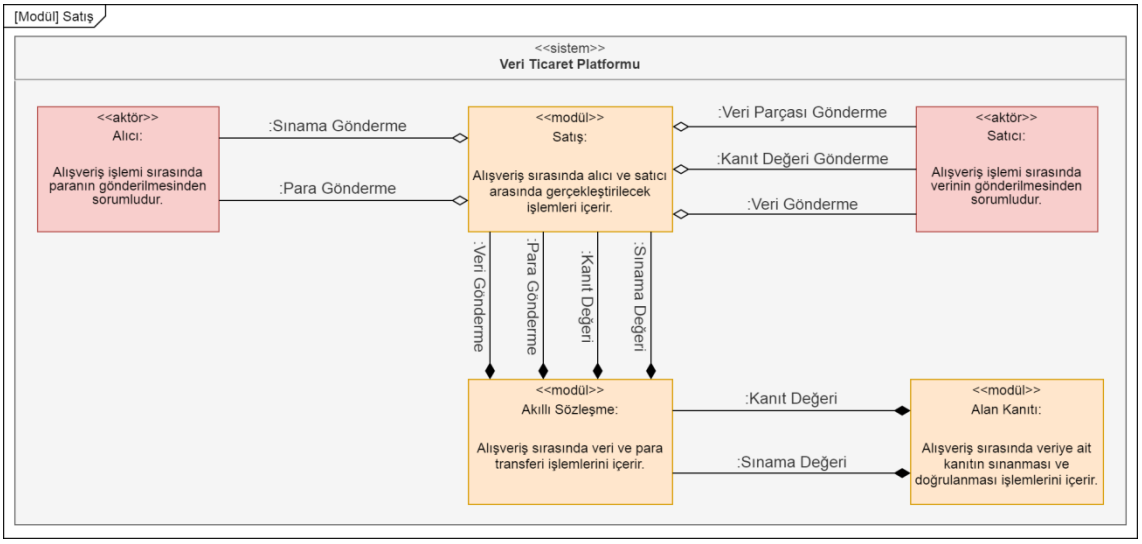
3.2 Senaryo

Bu bölümde yukarıda bahsedilen gereksinimlerin nasıl karşılanabileceği örnek bir senaryo üzerinden tartışılacaktır.

Sistemde satış yapmak isteyen satıcı; sistem içerisinde alıcıların verileri daha rahat araştırabilmesine ve değerlendirebilmesine imkân sağlamak için verilere ait bir üst bilgi hazırlar ve bu üst bilgi üzerinden verilere ait bir kayıt oluşturur. Gereksinimler kısmında da bahsedildiği gibi, blok zinciri şeffaf bir sistem olduğu için verilerin gizliliğini sağlamak adına satıcı verileri lokal depolama alanında saklayabileceği gibi şifreli olarak bir bulut depolama sisteminde de tutabilir.

Alıcılar platform üzerinde oluşturulan verilere ait kayıtları blok zinciri vasıtasıyla görüntüleyebilmektedir. Alıcılar bu verilerden birini satın almak istediğinde satış modülü üzerinden ilgili satıcıyla irtibata geçerler. Alışveriş işleminden önce sistemin alıcıya veriyi test edebilme imkânı vermesi gerekmektedir. Verinin test edilebilirliği diye adlandırdığımız bu gereksinim satıcının veriye ait küçük bir parçayı alıcıyla paylaşması yoluyla sağlanabilir. Bu noktada paylaşılan bu parçanın veriye ait olduğunun kanıtlanması gerekmektedir. Bununla birlikte satıcının verinin başlangıçta kaydı oluşturulan veri ile aynı olduğunu ve değiştirilmediğini, diğer bir ifadeyle verinin bütünlüğünün sağlandığını da alıcıya ispat etmesi gerekmektedir. Bu iki gereksinim sisteme entegre edilecek **Alan Kanıtı** modülü vasıtasıyla sağlanabilir.

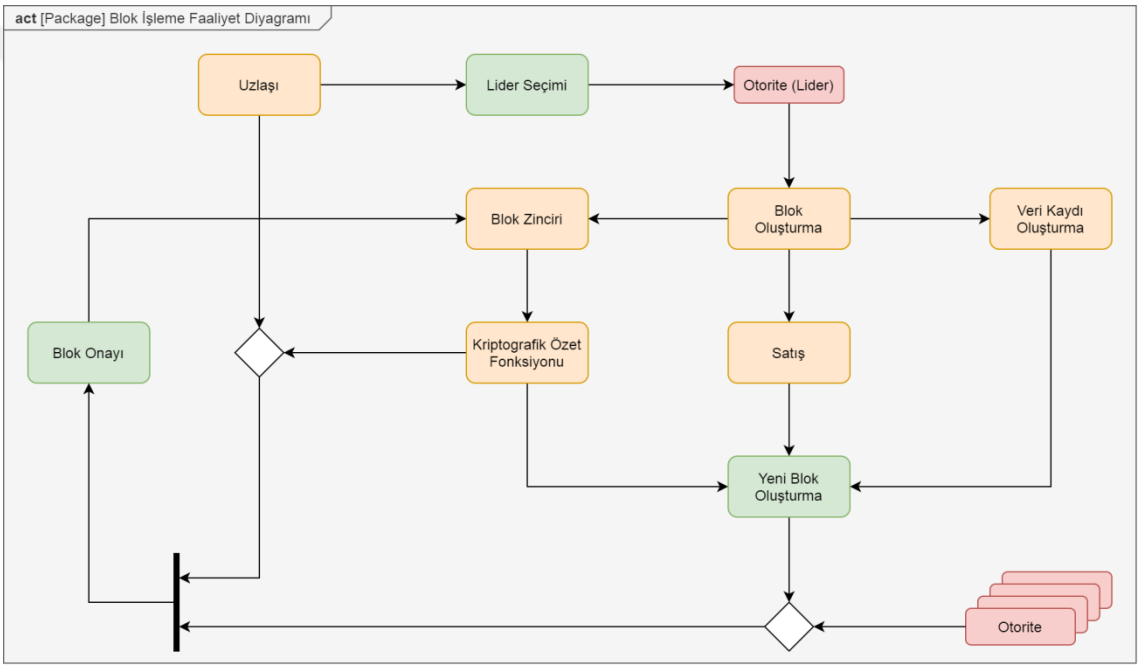
Bunların yanı sıra yukarıda da belirtildiği gibi Satış modülü ile ilintili karşılanması gereken alıcının veriyi ve satıcının parayı alabilmesi olarak adlandırdığımız gereksinimler sisteme entegre edilebilecek Akıllı Sözleşme modülü üzerinden karşılanabilirler. Alan kanıtı protokolünün uygulanmasına imkân sağlayacak **Alan Kanıtı** modülünün ve akıllı sözleşme çalıştırmamıza imkân sağlayacak Akıllı Sözleşme modülünün nasıl çalıştıklarına geçmeden önce Alan kanıtı protokolünü ve akıllı sözleşme kavramını kısaca açıklamak yerinde olacaktır.



Şekil 3.6 Örnek senaryo için "Satış" modülü

Yukarıda da ifade edildiği gibi bir önceki bölümde belirlenen ve Şekil 3.6'da gösterilen **Satış** modülü ile ilgili gereksinimler sisteme entegre edilen **Alan Kanıtı** ve **Akıllı Sözleşme** modülleri yardımıyla karşılanabilirler. **Alan Kanıtı** modülünde Dziembowski vd. (2015) tarafından önerilmiş Merkle ağaç yapısı uygulanarak gerçekleştirilen alan kanıtı protokolü kullanılmaktadır. Alan kanıtı protokolünün sağlıklı bir şekilde çalışabilmesi için satıcıların satmak istedikleri verilerle ilgili kayıt oluştururken alan kanıtında kullanılmak üzere verilere ait taahhüt değerini de veri üst bilgisine eklemeleri gerekmektedir. Bu noktada satıcılar ellerindeki veriyi önce n parçaya bölerler ve bu parçalar üzerinden bir Merkle ağacı oluştururlar. Sonrasında ilgili Merkle ağacının kökünü ve n sayısını taahhüt değeri olarak veri üst bilgisine eklerler. Alıcı platformdaki verilerden birini almak istediğinde; önce veriyi test edebilmek ve aynı zamanda veri bütünlüğünün ispatı için uygulanacak alan kanıtı protokolünde kullanılmak üzere, $\{1, \ldots, n\}$ kümesinden bir sayıyı rastgele bir şekilde sınama değeri olarak belirleyip satış modülü üzerinden ilgili satıcıya ve **Akıllı Sözleşme** modülüne gönderir. Sonrasında satıcı **Satış** modülü üzerinden sınama değerine karşılık gelen veri parçasını alıcıya ve veri parçasıyla beraber alan kanıtı protokolünde kullanılmak üzere oluşturduğu kanıt değerini **Akıllı Sözleşme** modülüne gönderir. Veri parçasını değerlendiren alıcı eğer veriyi almak isterse **Satış** modülü üzerinden **Akıllı Sözleşme** modülüne satış işlemi için gerekli olan parayı ve satıcıya bu isteğini gönderir. Satıcı bu isteğin karşılığı olarak veriyi **Akıllı Sözleşme** modülüne gönderir. Blok zinciri şeffaf bir sistem olduğundan

platform içerisinde verinin direkt paylaşılması veri gizliliği gereksinimi ile çelişeceğinden dolayı, satıcının veriyi ya da veri eğer bulut depolamada tutuluyorsa ilgili gizli anahtarı alıcının kimliğiyle şifreleyip ilgili şifreyi **Akıllı Sözleşme** modülüne göndermesi daha sağlıklı olacaktır. Diğer yandan, **Akıllı Sözleşme** modülü veri bütünlüğünün sağlanıp sağlanmadığını kontrol etmek için; **Alan Kanıtı** modülü üzerinden alıcıdan gelen sınama değeri, satıcıdan gelen kanıt değeri ve öncesinde zincire eklenmiş ilgili taahhüt değeri üzerinde alan kanıtı protokolünü çalıştırır. Eğer alan kanıtı protokolü verinin bütünlüğünü doğrularsa **Akıllı Sözleşme** modülü veri – para takasını gerçekleştirecektir.



Şekil 3.7 "Blok İşleme" faaliyet diyagramı

Platformda gerçekleştirilen işlemler periyodik olarak **Uzlaşı** modülü üzerinden belirlenen lider tarafından bloklara yazılmakta ve yine **Uzlaşı** modülü üzerinden çalıştırılan uzlaşı protokolü üzerinden onaylanarak zincire eklenmektedir. Şekil 3.7'deki faaliyet diyagramında da görüleceği gibi önce uzlaşı modülü üzerinden yeni bloğu oluşturacak lider seçilir. Sonrasında lider seçilen otorite ilgili zaman aralığında platformda paylaşılan işlemler üzerinden yeni bir bloğa oluşturur. Liderler ayrıca sağlam bir zincir elde edebilmek amacıyla, çakışmaya dirençli kriptografik bir özet fonksiyonunu zincire son eklenmiş blok üzerinde çalıştırarak bir özet değeri oluştururlar

ve bu değeri yeni bloğun başına yazarlar. Bu şekilde her bir bloğun çakışmaya dirençli kriptografik bir özet fonksiyonu üzerinden bir sonraki bloğa bağlanması sayesinde blokların kırılamaz bir zinciri elde edilmiş olmaktadır. Bu da bloklara işlenen kayıtların değiştirilemez bir biçimde saklanmasına imkân vermektedir.

Sonrasında liderler oluşturdukları blokları platformun diğer kullanıcılarıyla paylaşırlar. Platformun bu yapısı platformda gerçekleştirilen işlemlerin platforma kayıtlı bütün kullanıcılarca şeffaf bir biçimde görülebilmesine imkân tanıyacaktır. Yeni oluşturulan bloğu alan diğer otoriteler bloktaki işlemleri ve bloğun başına yazılan özet değerini doğruladıktan sonra ilgili bloğu zincire eklerler. Platformdaki diğer kullanıcıların bloğun doğruluğunu kontrol etme gibi bir sorumluluğu yoktur. Diğer kullanıcılar bu süreçten bağımsız olarak istediklerinde platformdaki otoritelerden biriyle irtibata geçerek ellerindeki zinciri güncellerler. İkinci bölümde de belirtildiği gibi platform izinsiz blok zinciri kullanılarak hayata geçirileceğinden, çok sayıda oyuncuyu tolere edebilen iş ispatı ya da hisse ispatı (proof of stake) gibi algoritmalar çalıştıran çekiliş tabanlı uzlaşı protokolleri kullanmak platformun verimliliğini artıracaktır. Ayrıca Garay vd. (2015) ve David vd. (2018) yukarıda zikredilen sırasıyla iş ispatı ve hisse ispatı tabanlı uzlaşı protokollerinin canlılık ve güvenlik gereksinimlerini karşıladığını formel olarak göstermişlerdir.

4. BLOK ZİNCİRİ TABANLI VERİ TİCARET PLATFORMU

Bölüm 3'te bahsedilen bileşenler ve bileşenler arasındaki ilişkiler kullanılarak blok zinciri tabanlı bir veri ticaret platformu ile işlenebilir verilerin güvenilir bir şekilde alım ve satımının yapılabilmesi problemine bir çözüm sunulmuştur.

4.1 Platformun Ön Gereksinimleri

Blok zinciri tabanlı bir veri ticaret platformunu gerçekleyebilmek için, platformun belli başlı kütüphanelerin ve teknolojilerin kullanılması gerekmektedir. Bu amaçla, platformun arayüz tasarımı, blok zincirinin kurulması, kullanılacak akıllı sözleşmelerin yazılması ve blok zinciri üzerinde derlenmesi, platformun blok zinciri ve akıllı sözleşmeler ile iletişiminin sağlanması gerekmektedir.

4.1.1 Akıllı sözleşmelerin yazılması

Platform üzerinde çalıştırılacak akıllı sözleşmeler, Ethereum platformunda kullanılan **Solidity** programlama dili ile yazılmaktadır.

Solidity, akıllı sözleşmelerin yazılması için kullanılan nesne yönelimli bir programlama dilidir. JavaScript, C++ ve Python dillerinden etkilenerek geliştirilen Solidity programlama dili, 2014 yılında Ethereum'un kurucularından olan Gavin Wood tarafından önerilmiştir. Şekil 4.1'de Solidity ile yazılmış basit bir akıllı sözleşme gösterilmektedir.

```
pragma solidity ^0.8.7;

contract ExampleContract {
    uint variable;

    function setVariable(uint _variable) public {
        variable = _variable;
    }

    function getVariable() public view returns (uint) {
        return variable;
    }
}
```

Şekil 4.1 Solidity dili ile yazılmış basit bir akıllı sözleşme

Akıllı sözleşme kodunun en üstünde, sözleşmenin yazıldığı Solidity sürümü belirtilmektedir. Devamında ise sözleşme üzerinde tanımlanan değişkenler ve metotlar bulunmaktadır. Sözleşme içerisindeki *setVariable* metodu, *variable* isimli işaretsiz tam sayı değişkeninin değerini değiştirmekte, *getVariable* metodu ise bu değişkenin mevcut değerini dönmektedir.

4.1.2 Blok zincirinin kurulması

Platformda oluşturulacak veri kayıtları ile bu kayıtlar üzerinden gelen veri taleplerinin ve başarıyla gerçekleştirilen alışverişlere ait işlemlerin depolanması için blok zinciri olarak **Ganache** kullanılmaktadır.

Ganache, yazılan akıllı sözleşmelerin üzerinde çalıştırılabildiği ve sözleşmelerin test edilebildiği Ethereum tabanlı bir blok zinciri emülatörüdür. Kullanıcı arayüzü de içeren Ganache'ın kullanılma sebepleri şu şekilde sıralanabilir:

- Basit ve hızlı bir şekilde lokal blok zinciri oluşturabilmek.
- Oluşturulan blok zincirindeki kullanıcı sayısını ve kullanıcıların başlangıçtaki Ether miktarlarını düzenleyebilmek.
- Ethereum gaz ücretini ve madencilik hızını düzenleyebilmek.

Kullanıcı arayüzü içeren Ganache CLI üzerinden yeni bir Ethereum blok zinciri oluşturabilmek mümkündür. Varsayılan olarak Ganache, lokal blok zinciri üzerinde her biri 100 Ether içeren 10 adet farklı kullanıcı tanımlamaktadır. Tanımlanan her bir kullanıcının Ethereum adresi ile birlikte kapalı anahtarları da görüntülenebilmektedir. Oluşturulan blok zinciri, bilgisayarın lokal adresi olan 127.0.0.1 adresi üzerindeki belirli bir port numarasından uzaktan yordam çağrısı (remote procedure call, RPC) sunucusu oluşturmaktadır. Oluşturulan bu sunucu sayesinde Ganache üzerindeki lokal blok zincirine bağlanılarak akıllı sözleşmeler çalıştırılabilmekte veya bir kullanıcı üzerinden işlemler yapılabilmektedir.

4.1.3 Akıllı sözleşmelerin blok zinciri üzerine yüklenmesi

Solidity programlama dili ile yazılan akıllı sözleşmelerin, hedeflenen şekilde çalışabilmeleri için üzerinde çalışacağı blok zincirine yüklenmesi gerekmektedir. Bu işlemi gerçekleştirebilmek için, **Truffle** geliştirme ortamı kullanılmaktadır.

Truffle; akıllı sözleşmelerin derlenebilmesi ve test edilebilmesi ile birlikte, sözleşmelerin çalışan bir blok zincirine yüklenebilmesini sağlayan bir geliştirme ortamıdır. Sözleşmelerin blok zincirine yüklenebilmesi için, Truffle ortamının bağlantı kuracağı blok zincirine ait IP adresi ve port numarasını bilmesi gerekmektedir. Bu bilgiler, oluşturulacak uygulamanın içerisinde yer alan Truffle yapılandırma dosyası içerisinde belirtilmektedir. Şekil 4.2'de, yapılandırma dosyasının blok zinciri ile bağlantı kuracağı kısmı içeren parametreler gösterilmektedir.

```
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1",
      port: 7545,
      network_id: "5777",
    },
  },
}
```

Şekil 4.2 Truffle yapılandırma dosyası içeriği

Yapılandırma dosyası içerisinde bulunan **"host"** ve **"port"** parametreleri, Truffle'ın bağlanacağı blok zincirinin IP adresi ve port numarasını belirtmektedir. **"network_id"** parametresi ise Ganache'ın ağ üzerinde kendini tanımlayacağı ağ kimliğini belirtmekte olup, bu değer de Ganache üzerinde varsayılan olarak tanımlıdır.

Truffle yapılandırması tamamlandıktan sonra Ganache üzerindeki blok zincirine akıllı sözleşmelerin yüklenebilmesi için öncelikle sözleşmenin Truffle üzerinde derlenmesi gerekmektedir. Bunun için **"truffle compile"** komutu kullanılarak var olan sözleşmelerin derlenmesi sağlanır. Herhangi bir hata alınmadığı durumda derlenen akıllı sözleşmenin blok zincirine yüklenme aşamasına gelinir. Yükleme aşamasında **"truffle migrate"** komutu kullanılarak belirli bir gaz ücreti karşılığında akıllı sözleşmenin blok zincirine yüklenmesi sağlanır. Truffle, akıllı sözleşmeyi varsayılan olarak Ganache üzerindeki blok zincirinde bulunan ilk kullanıcı üzerinden yüklemektedir. Başka bir kullanıcı üzerinden yüklenmek istendiğinde Şekil 4.2'de gösterilen yapılandırma dosyasına **"from"** alanı eklenerek bu alana kullanılacak olan kullanıcının adresi verilir.

4.1.4 Veri ticaret platformunun kurulması

Akıllı sözleşmelerin yazılması ve derlenmesiyle birlikte sözleşmelerin blok zinciri üzerine kurulumu sağlandıktan sonra, alıcı ve satıcıların veri alıp satabileceği bir veri ticaret platformunun oluşturulması gerekmektedir. Bu amaçla platformun ön uç (frontend) ve arka uç (backend) tasarımlarında farklı teknolojilerin kullanılması

mümkün olmakla beraber; bu çalışmadaki platformun geliştirilmesinde **HTML**, **CSS** ve **JavaScript** teknolojileri kullanılmaktadır.

HTML (HyperText Markup Language), bir internet sayfasının içeriğini oluşturma aşamasında kullanılan temel biçimlendirme dilidir. Sayfanın içeriğine başlıklar, paragraflar, linkler, listeler veya tablolar gibi birçok kontrolün kolay ve hızlı bir biçimde eklenmesini mümkün kılan bu dil; günümüzdeki internet sayfalarının neredeyse tamamında kullanılmaktadır.

CSS (Cascading Style Sheets), HTML üzerindeki kontrollerin başta renk, boyut ve yazı tipi olmak üzere birçok özelliğini stil olarak düzenlemek ve değiştirmek için kullanılan bir formatlama dilidir. Tasarım sırasında geliştiricinin işini kolaylaştırmasının yanı sıra sayfa üzerindeki kontrolleri de kullanıcı dostu hâle getirmekte ve internet sayfasının kolay bir şekilde kullanılmasına olanak sağlamaktadır.

JavaScript internet tarayıcılarında sıklıkla kullanılan bir programlama dili olup, temel amacı sayfaların ön ve arka taraflarındaki mantıksal ve dinamik işlemleri gerçekleştirmektedir. Bu dil, HTML sayfalarının içerisine rahatlıkla eklenebildiği için herhangi bir derleyiciye ihtiyaç duyulmadan internet sayfası yüklendiğinde kolaylıkla çalıştırılabilmektedir.

4.1.5 Veri ticaret platformunun blok zinciri ile bağlantısının yapılması

Geliştirilen veri ticaret platformunda, alıcıların veri kaydı, satıcıların da satın alma talebi oluşturmalarının ardından bu bilgilerin blok zincirine kaydedilmesi; daha sonra ise kaydedilen bu verilerin tekrar blok zincirinden çekilerek platformun arayüzünde kullanıcılara gösterilmesi gerekmektedir. Dolayısıyla HTML, CSS ve JavaScript ile tasarlanmış olan veri ticaret platformunun Ganache yardımıyla oluşturulan blok zincirine bağlantısının sağlanması gerekmektedir. Bu bağlantıyı sağlamak için **Web3.js** kullanılmaktadır.

Web3.js, Ethereum blok zinciri üzerinde çalışan akıllı sözleşmelerin internet sayfaları üzerinden çağrılmasına ve çalıştırılmasına olanak sağlayan bir JavaScript kütüphanesidir. Kütüphane sayesinde herhangi bir internet sayfası üzerinden JavaScript kullanılarak aktif olan bir blok zincirine bağlantı kurulabilmekte ve zincir üzerinde çalışmakta olan akıllı sözleşmelerin içerisindeki metotlar çağrılabilmekte veya değişkenlerine ulaşılabilmektedir. Bağlantı kurabilmek için, internet sayfasının yüklenmesi sırasında Şekil 4.3'te gösterilen JavaScript kodunun çalıştırılması gerekmektedir.

```
async function loadWeb3() {
  if (typeof web3 !== 'undefined') {
    App.web3Provider = web3.currentProvider;
    App.web3 = new Web3(web3.currentProvider);
  } else {
    window.alert('Please connect to Ethereum wallet.');
  }
  if (window.ethereum) {
    window.web3 = new Web3(ethereum);
    try {
      await ethereum.enable();
      web3.eth.sendTransaction({});
    } catch (error) {}
  } else if (window.web3) {
    App.web3Provider = web3.currentProvider;
    window.web3 = new Web3(web3.currentProvider);
    web3.eth.sendTransaction({});
  } else {
    console.log('Non-Ethereum browser detected.');
  }
}
```

Şekil 4.3 Platform üzerinden Web3.js bağlantısının kurulması

4.1.6 Platformda kullanılacak olan Ethereum cüzdanının bağlanması

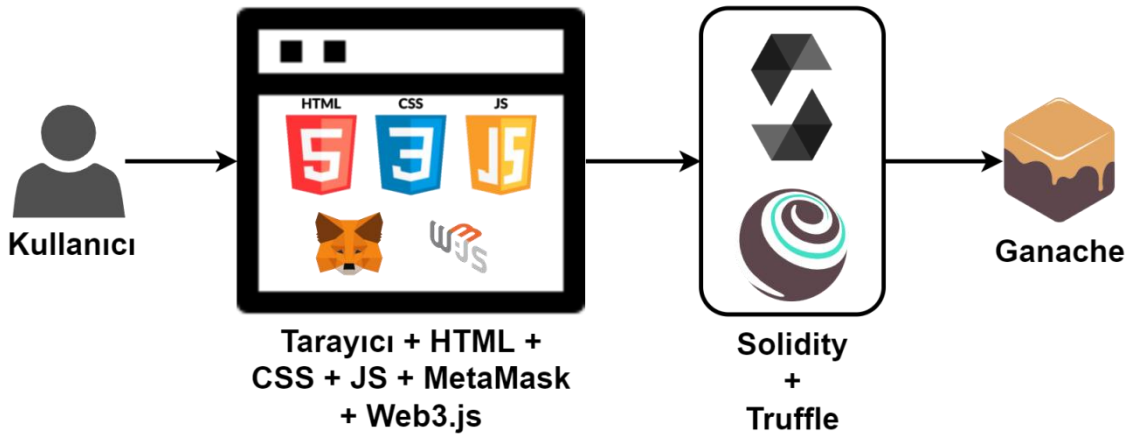
Platformdaki alıcıların, veri örneği isteği veya veri satın alma talebi göndermeleri esnasında platforma gereken gaz ücretini ödeyebilmeleri; ayrıca veri satın alımından

sonra da veri ücretinin satıcı tarafına gönderimini onaylayabilmeleri için platforma Ethereum cüzdanlarını bağlamaları gerekmektedir. Bu amaçla kullanılabilecek en uygun kripto para cüzdanı **MetaMask**'tır.

MetaMask, 2016 yılında piyasaya sürülen ve Ethereum tabanlı blok zincirleri için cüzdan hizmeti sunan bir uygulamadır. Masaüstü platformlarında tarayıcı eklentisi olarak yüklenebilen MetaMask sayesinde kullanıcılar Ethereum blok zincirini cihazlarına indirmelerine gerek kalmadan direkt olarak ekosistemdeki diğer kullanıcılara talep gönderebilmektedir. Bununla birlikte kullanıcılar uygulamaya Ethereum hesaplarını kapalı anahtarlarını girerek ekleyebilecekleri gibi, kapalı anahtarlarını oluşturan belli sayıdaki kelime grubunu (mnemonic) kullanarak da ekleyebilmektedirler.

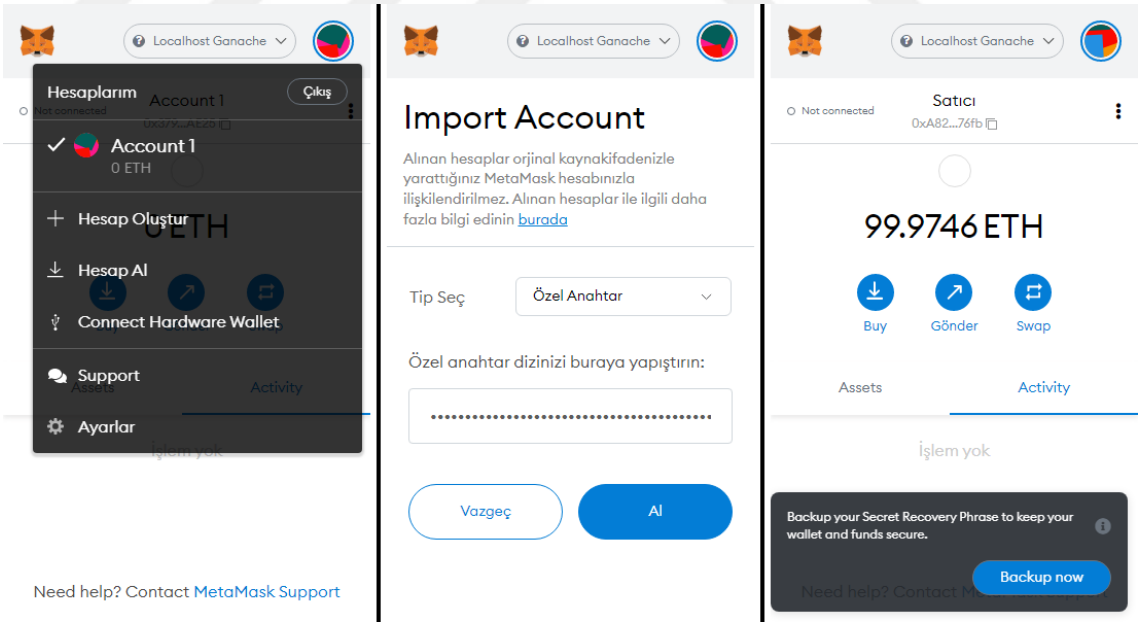
4.2 Platform Mimarisi

Veri ticaret platformunun mimarisi Şekil 4.4'te gösterilmektedir. Platforma farklı coğrafik konumlardaki çok sayıda kullanıcının kolay ve hızlı şekilde erişmesini mümkün kılmak adına platform bir web uygulaması olarak geliştirilmiştir. Kullanıcılar tarayıcıları üzerinden platforma erişebilmekte ve MetaMask eklentisine bağladıkları Ethereum cüzdanları ile işlem yapabilmektedirler. Web uygulamasının arka tarafında kullanılan Web3.js kütüphanesi, Solidity programlama dili ile yazılan ve Truffle ile derlenerek Ganache lokal blok zincirine yüklenen akıllı sözleşmelere erişerek operasyonel işlemleri gerçekleştirmektedir.



Şekil 4.4 Blok zinciri tabanlı veri ticaret platformu mimarisi

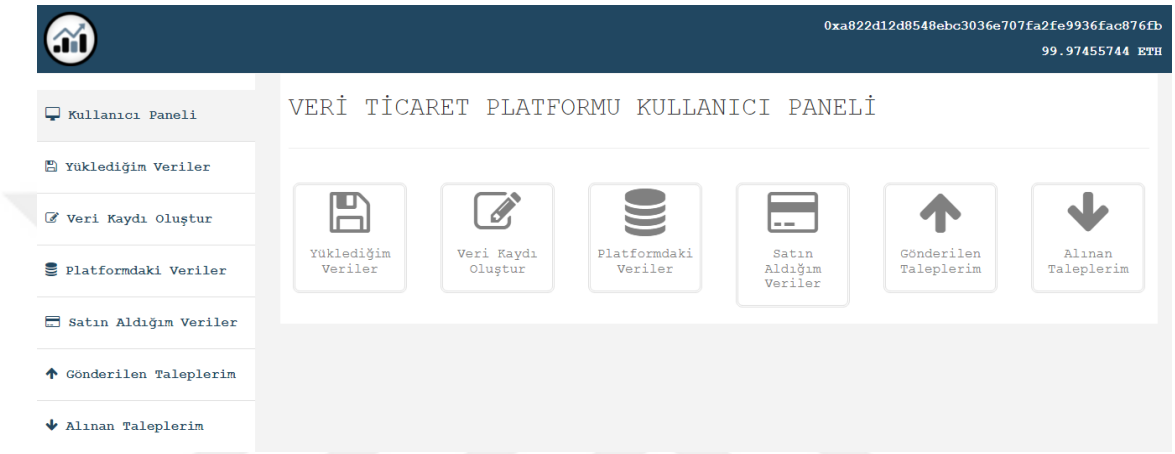
Kullanıcılar, Ethereum cüzdanlarını platforma bağlayarak veri kaydı oluşturma, satın alma talebi gönderme ve alışveriş işlemlerini gerçekleştirebilmektedir. Bunun için kullanıcılar tarayıcılarına eklenti olarak yükledikleri MetaMask'ı açarak Şekil 4.5'te gösterilen "Hesap Al" seçeneğine tıklarlar ve kapalı anahtarlarını içeri aktararak cüzdanlarını bağlarlar.



Şekil 4.5 MetaMask'a cüzdan ekleme aşamaları

Satıcı veya alıcılar platforma giriş yaptıktan sonra ilk olarak Şekil 4.6'da gösterilen kullanıcı paneline erişmektedirler. Satıcılar kullanıcı paneli vasıtasıyla yeni veri kaydı

oluşturabilmekte, oluşturdukları veri kayıtlarını görebilmekte ve gelen satın alma taleplerini görerek bu taleplere cevap verebilmektedirler. Alıcılar ise platforma yüklenen veri kayıtlarını görebilmekte, bu veri kayıtlarından istediklerine satın alma talebi gönderebilmekte ve gönderdikleri talepleri onaylayabilmektedirler. Platformun sağ üst köşesinde giriş yapan kullanıcıya ait Ethereum adresi ve Ethereum cüzdanlarındaki bakiye bilgisi gösterilmektedir.

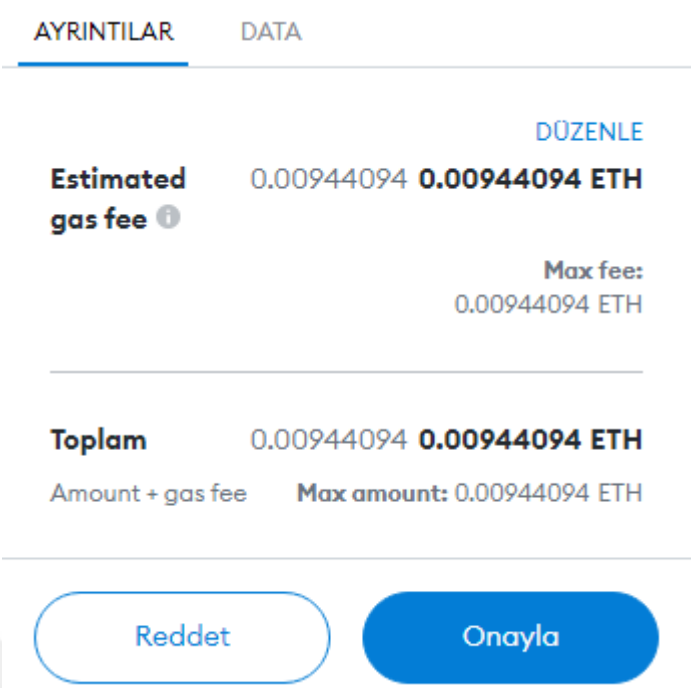


Şekil 4.6 Veri ticaret platformu kullanıcı paneli

Satıcılar, kullanıcı paneli üzerinden veri kaydı oluşturabilmekte ve oluşturdukları veri kayıtlarını görebilmektedirler. Yeni bir veri kaydı oluşturmak için, “Veri Kaydı Oluştur” sayfasına gidilir ve Şekil 4.7’de gösterilen bu sayfada ilgili alanları doldurarak yeni bir veri kaydı oluşturmaları sağlanır.

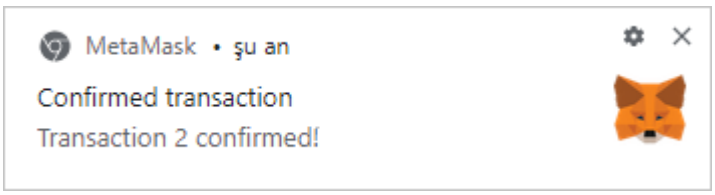
Şekil 4.7 Veri kaydı oluşturma arayüzü

"Üst Veri" alanı satışa sunulan veri hakkındaki genel bilgileri içermekte ve bu bilgiler, alıcıların platformda dolaşması esnasında satıştaki veriler hakkında bilgilenmelerine olanak sağlamaktadır. Verinin tamamını blok zincirine yüklemek blok zincirinin boyutunu çok fazla artıracağı ve dolayısıyla blok zincirinin performansını düşüreceğinden; veri kaydı oluşturma esnasında verinin kanıt değeri yüklenmektedir. Bu amaçla veri kaydı oluşturma arayüzünde satıcıdan, kanıtının hesaplanabilmesi için satışa sunacağı veriyi seçmesi istenmektedir. İlgili alanlar doldurulduktan sonra veri kaydı oluşturulabilmesi için bu kaydın blok zincirine eklenmesi gerekmekte, bu da belirli bir miktar Ethereum gazına mal olmaktadır. Blok zinciri operasyonları için gaza ihtiyaç duyulduğu durumda MetaMask Şekil 4.8'de gösterildiği gibi, kullanıcıdan tahsil edilecek gaz için onay talep eden bir onay penceresi açmaktadır.



Şekil 4.8 MetaMask veri kaydı oluşturma ücreti tahsilat onay ekranı

Satıcının onaylamasından sonra veri kaydı ve kayda ait işlem bilgisi blok zincirine yazılmakta ve Şekil 4.9'daki gibi işlemin onaylandığına dair bir bilgi ekranı gösterilmektedir.



Şekil 4.9 MetaMask onay işlemi sonrası bilgi ekranı

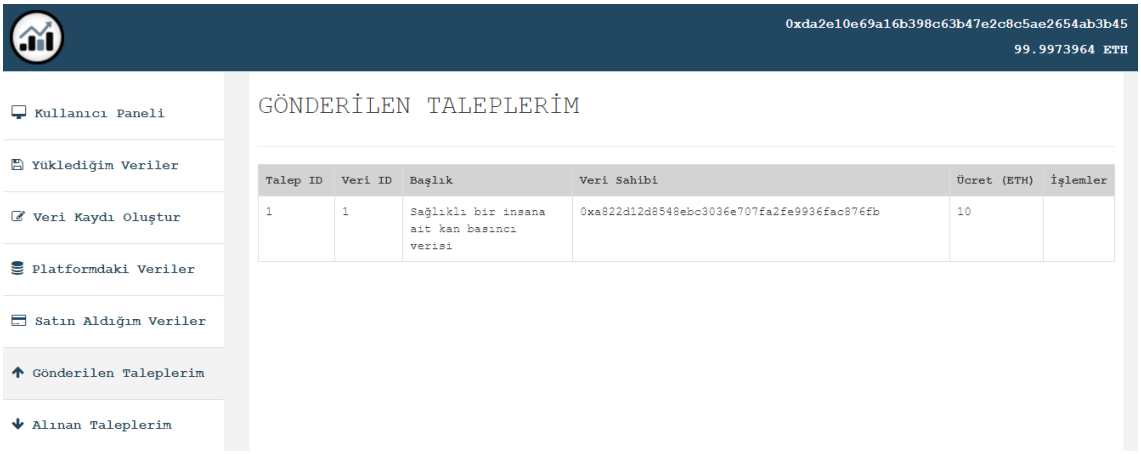
Satıcı, oluşturduğu veri kayıtlarını Şekil 4.10'da gösterilen "Yüklediğim Veriler" sayfasında görebilmektedir.

Şekil 4.10 Satıcının kendi oluşturduğu veri kayıtlarını görebileceği arayüz

Platform üzerinde oluşturulan bütün veri kayıtlarına kullanıcıların tamamı "Platformdaki Veriler" sayfası üzerinden erişebilmekte ve istedikleri bir veri kaydına satın alma talebi gönderebilmektedirler. Bu sayfa Şekil 4.11'de gösterilmektedir.

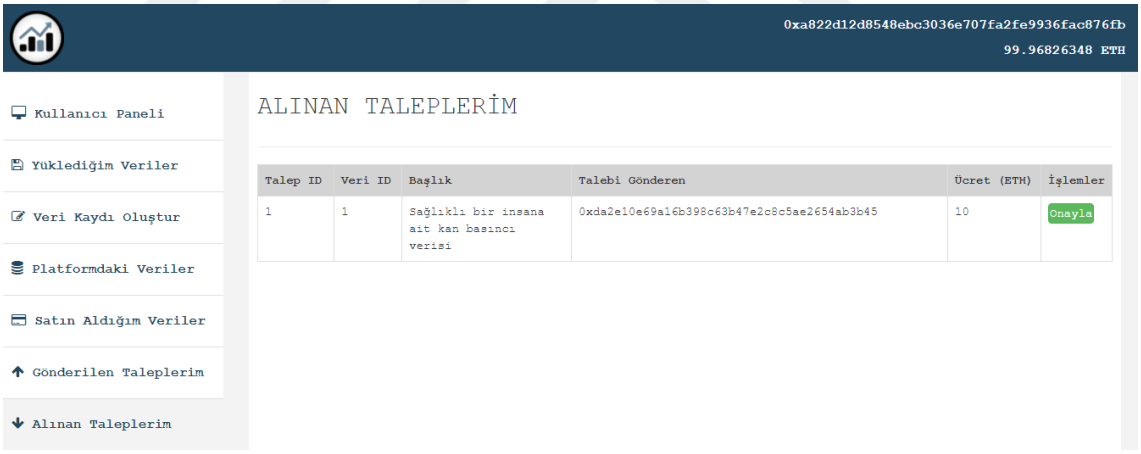
Şekil 4.11 "Platformdaki Veriler" arayüzü

Veri satın almak isteyen bir alıcının, satıcıya satın alma talebi göndermesi gerekmektedir. Bunun için "Platformdaki Veriler" sayfasında ilgili veri satırında bulunan "Satın Al" düğmesine basılır. Satın alma talebi gönderildikten sonra kullanıcı, gönderdiği bütün talepleri Şekil 4.12'de gösterilen "Gönderilen Taleplerim" sayfasından görebilmektedir.



Şekil 4.12 "Gönderilen Taleplerim" arayüzü

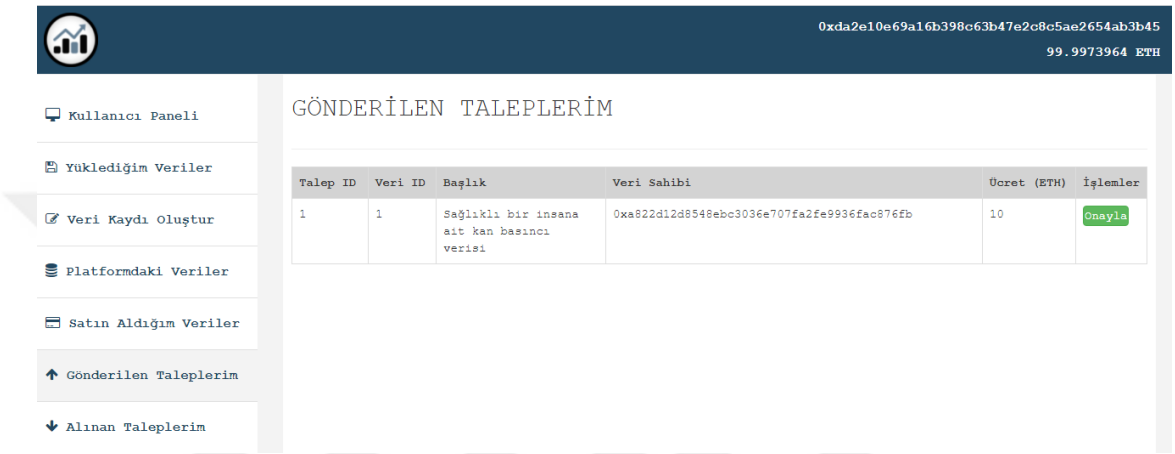
Alıcı bir satın alma talebi gönderdiğinde, bu talep satıcının kullanıcı panelindeki "Alınan Taleplerim" sayfasına düşmektedir. Bu sayede satıcı, kendisine gönderilen bütün satın alma taleplerini görebilmekte ve bu talepleri onaylayabilmektedir. "Alınan Taleplerim" arayüzü Şekil 4.13'te gösterilmektedir.



Şekil 4.13 "Alınan Taleplerim" arayüzü

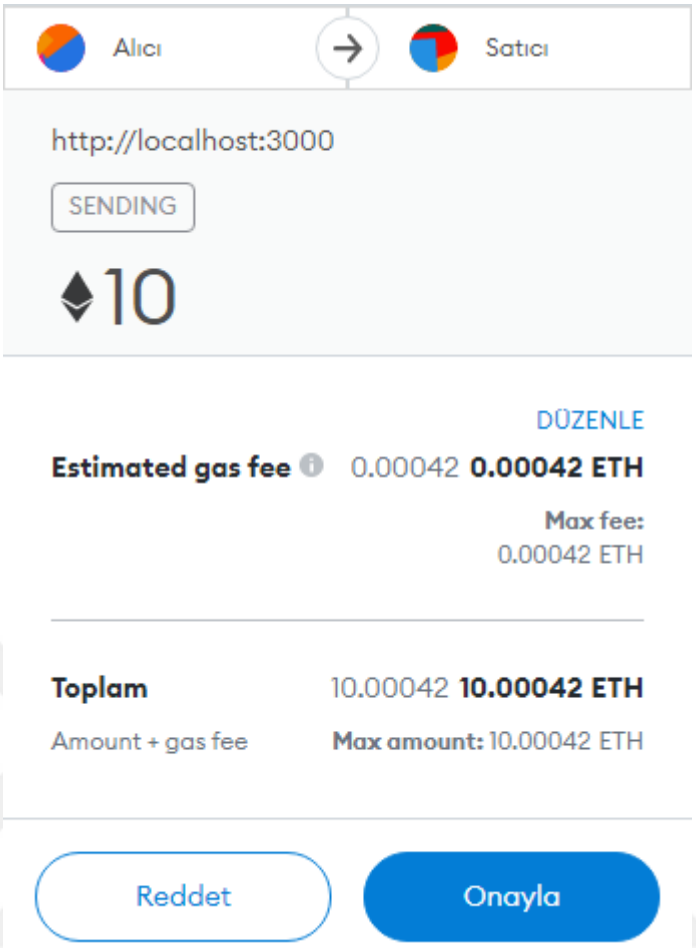
Satıcının bir satın alma talebini onaylaması durumunda Bölüm 3'te de bahsedildiği üzere veri kaydı oluştururken yüklediği verinin, göndereceği veri ile aynı olduğunun teyit edilmesi gerekmektedir. Bu nedenle bir satıcı kendisine gelen bir satın alma talebini "Onayla" düğmesine basarak onaylarken; aynı zamanda lokal depolama alanında bulunan veriyi seçerek kanıtının hesaplanmasını sağlayacaktır. Satıcının, kendisine gelen talebi onaylarken seçtiği verinin kanıtı ile blok zincirinde bulunan veri

kaydına ait kanıt değeri aynı ise satın alma talebi satıcı tarafından onaylanacaktır; aksi durumda satıcıya uyarı gösterilerek yüklediği verinin yanlış olduğu bildirilmektedir. Satıcı satın alma talebini onayladıktan sonra bir onay da alıcı tarafından beklenecek ve alıcının veri ücretini göndermesi sağlanacaktır. Bu amaçla yukarıda bahsedilen "Gönderilen Taleplerim" sayfasında Şekil 4.14'te olduğu gibi ilgili veri talebi satırında bir "Onayla" düğmesi bulunacak ve alıcının satın almayı onaylaması sağlanacaktır.



Şekil 4.14 "Gönderilen Taleplerim" arayüzünde alıcı onayının gerçekleştirilmesi

Alıcı, kendisinin gönderdiği ve satıcı tarafından onaylanmış olan satın alma talebini onaylamak için "Onayla" düğmesine basacaktır. Bu işlemin akabinde Şekil 4.15'te gösterilen MetaMask penceresi açılarak veri ücretinin alıcıya gönderilmesi için onay istenmektedir. Yukarıdaki şekilde de görüleceği gibi veri ücreti 10 ETH olduğu için MetaMask penceresinde gönderilecek veri ücreti ve ücretin gönderilmesi için gereken gaz miktarı toplamı 10,00042 ETH olmaktadır.



Şekil 4.15 MetaMask veri ücreti gönderme onay ekranı

Alıcı, satıcıya ilgili ücreti gönderdikten sonra ilgili miktar alıcının cüzdanından düşülerek satıcının cüzdanına eklenmektedir. Platformun gerçeklenmesi esnasında Ganache lokal blok zinciri kullanıldığı için Ganache arayüzü üzerinden kullanıcılar ve kullanıcı bakiyeleri takip edilebilmektedir. Şekil 4.16'da alışveriş işlemi sonucunda satıcı (ilk kullanıcı) ve alıcının (ikinci kullanıcı) bakiyeleri gösterilmektedir. Satıcının bakiyesinin 0,03 ETH eksik görünmesinin sebebinin akıllı sözleşmenin blok zincirine yüklenme, ilgili veri kaydını oluşturma ve satın alma talebini onaylama işlemleri sonucunda gereken gazlardan kaynaklandığına dikkat edilmelidir.

Şekil 4.16 Alışveriş işlemi sonucunda satıcı (üstte) ve alıcı (altta) bakiyeleri

Alışveriş işlemi sonrasında alıcı, satın aldığı verileri Şekil 4.17'de gösterilen "Satın Aldığım Veriler" sayfası üzerinden görebilmekte ve indirebilmektedir.

Şekil 4.17 "Satın Aldığım Veriler" arayüzü

5. BULGULAR VE TARTIŞMA

Bölüm 3'te önerilen blok zinciri tabanlı veri ticaret platformunun gerçeklenebilirliğini göstermek için örnek bir platform tasarlanarak platformun belli başlı gereksinimleri test edilmiştir. Bu amaçla Ganache ile 10 adet kullanıcı içeren ve her kullanıcıda 100 ETH bakiye bulunan lokal bir blok zinciri oluşturulmuştur. Blok zinciri üzerinde yapılacak işlemler için gereken gaz ücreti 20.000.000.000 Wei, yani 20 Gwei olarak belirlenmiştir.

Platformun kullanıcı ile etkileşime geçen ön yüzü bir web sayfası olarak tasarlanmış ve arka tarafında ise blok zinciri ile haberleşmenin sağlanabilmesi için Web3.js kütüphanesi kullanılmıştır. Ganache ile oluşturulan blok zincirindeki kullanıcıların platforma bağlanabilmeleri için MetaMask tarayıcı eklentisi kullanılmıştır. Platformdaki işlemlerin gerçekleştirilmesi için gereken akıllı sözleşmeler ise Solidity programlama dili ile yazılmış ve Truffle ile derlendikten sonra blok zincirine yüklenmiştir.

Tasarlanan platform üzerinde veri kaydı oluşturma, kullanıcının kendi oluşturduğu veri kayıtlarını görme, platforma yüklenen tüm veri kayıtlarını görme, satın alınmak istenen bir veri için satın alma isteği oluşturma, oluşturulan satın alma isteğinin satıcı ve alıcı taraflarında onaylanması ve en sonunda ise kullanıcıların satın aldıkları verileri görebilme işlemleri gerçekleştirilmiştir. Platformun arka tarafında ise veri kaydı oluşturma sırasında verinin kanıt değerinin oluşturulması ve blok zincirine kaydedilmesi, alışveriş sırasında satıcının göndereceği veriye ait kanıtın hesaplanarak blok zincirindeki kanıt ile aynı olduğunun kontrol edilmesi, bu iki kanıtın aynı olmadığı durumda alışverişin gerçekleştirilmemesi ve kanıtın aynı olduğu durumda ise alışveriş işleminin gerçekleştirilerek veri ücretinin alıcıdan satıcıya gönderilmesi işlemleri gerçekleştirilmiştir. Bu işlemlerin başarıyla gerçekleştirildiği görülmüş olup yukarıda önerilen blok zinciri tabanlı veri ticaret platformunun gerçeklenebilir olduğu gösterilmiştir. Bununla birlikte, platformdaki operasyonlar için yazılmış olan akıllı sözleşmeler içerisindeki metotların gaz gereksinimleri, web üzerinden sözleşmelerin yazılmasına, derlenmesine ve çalıştırılmasına olanak sağlayan Remix IDE uygulaması üzerinde çalıştırılarak hesaplanmıştır. Bu gaz gereksinimlerinin, varsayılan gaz ücreti

olan 20 Gwei üzerinden hesabı yapılarak Ethereum karşılıkları da hesaplanmıştır. Hesaplanan bu değerler Çizelge 5.1'de gösterilmektedir.

Çizelge 5.1 Akıllı sözleşme metotlarının gaz gereksinimleri

Akıllı Sözleşme Metodu	**Kullanım Amacı**	**Gaz Gereksinimi**	**Ethereum Karşılığı**
createDataRecord	Satıcı tarafından veri kaydı oluşturma	323.393	0,00646786
createDataPurchaseRequest	Alıcı tarafından veri satın alma isteği gönderme	134.761	0,00269522
approveBySeller	Satıcı tarafından satın alma isteğinin onaylanması	155.457	0,00310914
approveByBuyer	Alıcı tarafından satın alma isteğinin onaylanması	155.535	0,0031107
sendPriceToSeller	Veri ücretinin alıcıdan satıcıya transferi	31.810	0,0006362

Çizelgedeki değerlerden de görüleceği üzere, gönderilen bir veri satın alma isteğinin alıcı ve satıcı taraflarındaki onayı yaklaşık olarak aynı miktarda gaza mal olmaktadır. Bununla birlikte satın alma isteği gönderme metodu da bu iki metot gibi düşük miktarda gaz istemektedir. Platforma yeni bir veri kaydının yapılması ve bu veri kaydında başlık, üst veri, kanıt, fiyat vb. alanların bulunması, bu değerlerin blok zincirine yazılmasını gerektirdiği için daha yüksek miktarda gaza mal olmaktadır. Varsayılan gaz ücretinin 20 Gwei olduğu düşünülürse, yeni bir veri kaydı oluşturmanın kullanıcıya maliyeti yaklaşık olarak $323.393 * 20 * 10^{-9} \approx 0{,}0065$ Ethereum olmaktadır. Satıcıların satmakta olduğu veriler bu gaz maliyetinin katbekat üstünde olduğu için satıcı tarafında herhangi bir zarar durumu olmamaktadır. Veri ücretinin alıcıdan satıcıya gönderilmesi işlemi için gereken gaz miktarı ise beklendiği üzere diğer metotların gaz ihtiyaçlarına göre çok daha düşüktür. Çünkü para transferi gibi basit, hızlı ve kullanıcılar arasında çok sık kullanılacak bir işlemin gaz maliyetinin çok yüksek olmaması gerekmektedir.

6. SONUÇ

Bu çalışma kapsamında, nesnelerin interneti ağına bağlı olan cihazlardan toplanan verilerin güvenilir ve sağlıklı bir şekilde nasıl satılabileceği konusu tartışılmıştır. Literatürdeki diğer blok zinciri çalışmalarının yanı sıra, benzer blok zinciri tabanlı veri ticaret platformları da incelenerek bunlardan da bahsedilmiştir. Bunların ardından ise güvenilir bir blok zinciri tabanlı veri ticaret platformu önerilmiş, platformdaki bileşenler ve bileşenler arası ilişkiler SysML yardımıyla modellenerek platformda hangi gereksinimlerin olması ve bu gereksinimlerin hangi bileşenler tarafından karşılanması gerektiği yine SysML modelleri üzerinden gösterilmiştir. Önerilen bu platformda başta bu verilerin platforma sunulması olmak üzere, satın alma taleplerinin alınması, bu taleplerin onaylanması, alışveriş sırasında verinin doğruluğunun sağlanması ve veri karşılığında paranın güvenilir bir şekilde karşı tarafa iletilmesi gibi önemli gereksinimlerin nasıl karşılanabileceği gösterilmiştir. Bunların doğrultusunda ise bu gereksinimlerin büyük bir kısmını karşılayan ve Ethereum blok zinciri kullanan web tabanlı bir platform gerçeklenmiş ve farklı senaryolarda çalıştırılarak yukarıda bahsedilen ihtiyaçların karşılandığı gösterilmiştir. Bunun yanı sıra ihtiyaçların karşılanabilmesi için yazılan akıllı sözleşmelerin gaz ihtiyaçları da hesaplanarak, böyle bir platformun gerçeklenmesi sonucunda alıcı, satıcı ve madencilere maliyet açısından çok fazla yük bindirilmeyeceği ve işlemler sonucundaki ücretlerin veri satışından elde edilen ücretlere göre çok daha makul olduğu görülmüştür.

Çalışmada Ethereum platformu ve buna bağlı olarak akıllı sözleşmeler kullanıldığı için her ne kadar işlem ücreti olarak taraflara çok fazla yük binmese de bu ücretler diğer platformlara göre daha yüksektir. Ancak bu ücretleri daha da düşürebilmek için Avalanche platformu kullanılabilir. Avalanche, Ethereum platformu gibi akıllı sözleşmelere izin veren ve işlem ücretlerinin Ethereum'a göre oldukça düşük olduğu açık kaynak bir blok zinciridir.

Bunlarla birlikte yapılan bu çalışmada bir alıcının, satıcıdan veriyi aldıktan sonra aynı veriyi daha yüksek bir fiyata satışa çıkarması mümkün olup; bu durum serbest piyasa ekonomisi gereği pekâlâ gerçekleşebilir bir durumdur. Ancak bu konuda herhangi bir

kısıtlama getirme ihtiyacı duyulursa, satıcıya ait verinin içerisine verinin ilk sahibi bilgisi gömülerek bu durum belirli bir seviyeye kadar engellenebilir. Yine de veriyi satın alan alıcı, artık verinin yeni sahibi sayıldığı için veri üzerinde değişiklik yaparak (örneğin verinin sadece bir parçasını kullanarak) veya yapabiliyorsa ilk sahibi bilgisini veriden çıkarmak suretiyle bu veriyi tekrar platforma yüklemesi de oldukça olasıdır.

Bölüm 3'te de bahsedildiği üzere, veri kanıtının hesaplanması esnasında alan kanıtı protokolü uygulanmaktadır. Alan kanıtının uygulanması ve Merkle ağacının oluşturulabilmesi için, platformda satışa çıkarılacak verinin mantıklı parçalara ayrılabiliyor olması gerekmektedir. Bu aşamada satıcının elindeki verinin nasıl bir formatta olması gerektiği ve bu verinin nasıl bölünebileceği ileriki çalışmalarda belirlenecektir.

Bu çalışmada önerilen blok zinciri tabanlı veri ticaret platformu üzerinde, yeni bir veri kaydı oluşturulduğunda bu verilerin şifrelenerek bir bulut sunucusuna aktarılması ve satın alma işlemi sonrasında alıcıya verinin bu sunucu üzerinden gönderilmesi düşünülebilir. Bunlar haricinde bu platformun verinin dâhil olduğu bütün alanlarda, gerek veri ticaretinin gerekse verilerin yönetildiği bir sistemde, kullanılabilmesi mümkündür.

KAYNAKLAR

An, B., Xiao, M., Liu, A., Gao, G., Zhao, H. 2019. Truthful Crowdsensed Data Trading Based on Reverse Auction and Blockchain, International Conference on Database Systems for Advanced Applications, 22-25 April, 292-309, Chiang Mai, Thailand.

Anonymous. 2021a. Web Sitesi: https://en.wikipedia.org/wiki/Internet_of_things, Erişim Tarihi: 14.12.2021.

Anonymous. 2021b. Web Sitesi: https://ethereum.org/en/developers/docs/gas, Erişim Tarihi: 13.12.2021.

Azaria, A., Ekblaw, A., Vieira T., Lippman A. 2016. MedRec: Using Blockchain for Medical Data Access and Permission Management, 2nd International Conference on Open and Big Data (OBD), 22-24 August, 25-30, Vienna, Austria.

Back, A. 2002. Hashcash - A Denial of Service Counter-Measure. Web Sitesi: http://www.hashcash.org/hashcash.pdf, Erişim Tarihi: 13.12.2021.

Baran, P. 1964. On Distributed Communications Networks. IEEE Transactions on Communications Systems, 12(1), 1-9.

Biswas, K., Muthukkumarasamy, V. 2016. Securing Smart Cities Using Blockchain Technology, IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 12-14 December, 1392-1393, New South Wales, Australia.

Chen, C., Wu, J., Lin, H., Chen, W., Zheng, Z. 2019. A Secure and Efficient Blockchain-Based Data Trading Approach for Internet of Vehicles. IEEE Transactions on Vehicular Technology, 68(9), 9110-9121.

Dai, W., Dai, C., Choo, K.R., Cui, C., Zou, D., Jin, H. 2019. SDTE: A Secure Blockchain-Based Data Trading Ecosystem. IEEE Transactions on Information Forensics and Security, 15, 725-737.

David, B., Gaži, P., Kiayias, A., Russell, A. 2018. Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain, EUROCRYPT 2018, 29 April-3 May, 66-98, Tel Aviv, Israel.

Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P. 2019. LSB: A Lightweight Scalable Blockchain for IoT security and anonymity. Journal of Parallel and Distributed Computing, 134, 180-197.

Dwork, C., Naor, M. 1993. Pricing via Processing or Combatting Junk Mail, 12th Annual International Cryptology Conference (CRYPTO' 92), 16-20 August, 139-147, California, USA.

Dziembowski, S., Faust, S., Kolmogorov, V., Pietrzak, K. 2015. Proofs of Space, Advances in Cryptology - CRYPTO 2015, 16-20 August, 585-605, California, USA.

Elbüz, A., Osmanoğlu, M., Tanrıöver, Ö. 2019. Designing a Secure Blockchain-based Trading Platform for Internet of Things. Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, 61(1), 102-110.

Garay, J., Kiayias, A., Leonardos, N. 2015. The Bitcoin Backbone Protocol: Analysis and Applications, EUROCRYPT 2015, 26-30 April, 281-310, Sofia, Bulgaria.

Goharshady, A.K., Behrouz, A., Chatteriee, K. 2018. Secure Credit Reporting on the Blockchain, 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), 30 July-3 August, 1343-1348, New Scotland, Canada.

Guan, Z., Shao, X., Wan, Z. 2018. Secure Fair and Efficient Data Trading Without Third Party Using Blockchain, 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), 30 July-3 August, 1395-1401, Halifax, Canada.

Haber, S., Stornetta, W.S. 1991. How to time-stamp a digital document. Journal of Cryptology, 3(2), 99-111.

Hollander, L. 2019. The Ethereum Virtual Machine - How does it work? Web Sitesi: https://medium.com/mycrypto/the-ethereum-virtual-machine-how-does-it-work-9abac2b7c9e, Erişim Tarihi: 13.12.2021.

Huang, Y., Wu, J., Long, C. 2018. Drugledger: A Practical Blockchain System for Drug Traceability and Regulation, 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), 30 July-3 August, 1137-1144, New Scotland, Canada.

Huang, Z., Su, X., Zhang, Y., Shi, C., Zhang, H., Xie, L. 2017. A decentralized solution for IoT data trusted exchange based-on blockchain, 3rd IEEE International Conference on Computer and Communications (ICCC), 13-16 December, 1180-1184, Chengdu, China.

Huh, S., Cho, S., Kim, S. 2017. Managing IoT devices using blockchain platform, 19th International Conference on Advanced Communication Technology (ICACT), 19-22 February, 464-467, Pyeongchang, South Korea.

Jakobsson, M., Juels, A. 1999. Proofs of Work and Bread Pudding Protocols, Secure Information Networks: Communications and Multimedia Security, 20-21 September, 258-272, Leuven, Belgium.

Kiyomoto, S., Fukushima, K. 2018. Fair-trading protocol for anonymised datasets requirements and solution, 4th International Conference on Information Management (ICIM), 25-27 May, 13-16, Oxford, UK.

Lee, B., Lee, J.H. 2017. Blockchain-based secure firmware update for embedded devices in an Internet of Things environment. The Journal of Supercomputing, 73, 1152-1167.

Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., Njilla, L. 2017. ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability, 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), 14-17 May, 468-477, Madrid, Spain.

Manzoor, A., Liyanage, M., Braeke, A., Kanhere, S.S., Ylianttila, M. 2019. Blockchain based Proxy Re-Encryption Scheme for Secure IoT Data Sharing, IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 14-17 May, 99-103, Seoul, South Korea.

Nakamoto, S. 2008. Bitcoin: A peer-to-peer electronic cash system. Web Sitesi: https://bitcoin.org/bitcoin.pdf, Erişim Tarihi: 13.12.2021.

Papadodimas, G., Palaiokrasas, G., Litke, A., Varvarigou, T. 2018. Implementation of smart contracts for blockchain based IoT applications, 9th International Conference on the Network of the Future (NOF), 19-21 November, 60-67, Poznań, Poland.

Raji, R.S. 1994. Smart networks for control. IEEE Spectrum, 31(6), 49-55.

Sabounchi, M., Wei, J., Roche', R. 2018. Blockchain-Enabled Peer-to-Peer Data Trading Mechanism, 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), 30 July-3 August, 1410-1416, Halifax, Canada.

Sagirlar, G., Carminati, B., Ferrari, E., Sheehan, J.D., Ragnoli, E. 2018. Hybrid-IoT: Hybrid Blockchain Architecture for Internet of Things - PoW Sub-blockchains. Web Sitesi: https://arxiv.org/abs/1804.03903, Erişim Tarihi: 13.12.2021.

Tzianos, P., Pipelidis, G., Tsiamitros, N. 2019. Hermes: An Open and Transparent Marketplace for IoT Sensor Data over Distributed Ledgers, IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 14-17 May, 167-170, Seoul, South Korea.

Wang, Z., Tian, Y., Zhu, J. 2018. Data Sharing and Tracing Scheme Based on Blockchain, 8th International Conference on Logistics, Informatics and Service Sciences (LISS), 3-6 August, 1-6, Toronto, Canada.

Wilkinson, S., Boshevski, T., Brandoff, J., Prestwich, J., Hall G., Gerbes, P., Hutchins, P., Pollard, C. 2016. Storj: A Peer-to-Peer Cloud Storage Network. Web Sitesi: https://www.storj.io/storjv2.pdf, Erişim Tarihi: 13.12.2021.

Wood, G. 2014. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Web Sitesi: https://ethereum.github.io/yellowpaper/paper.pdf, Erişim Tarihi: 13.12.2021.

Zheng S., Pan L., Hu D., Li M., Fan Y. 2020. A Blockchain-Based Trading Platform for Big Data, IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 6-9 July, 991-996, Toronto, Canada.

Zyskind, G., Nathan, O., Pentland, A.S. 2015. Decentralizing Privacy: Using Blockchain to Protect Personal Data, IEEE Security and Privacy Workshops, 21-22 May, 180-184, California, USA.