KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

BLOKZİNCİRİN TEDARİK ZİNCİRİNDE KULLANIMI İÇİN TASLAK BİR MİMARİ

YILMAZ DİKİLİTAŞ

KOCAELİ 2021

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

BLOKZİNCİRİN TEDARİK ZİNCİRİNDE KULLANIMI İÇİN TASLAK BİR MİMARİ

YILMAZ DİKİLİTAŞ

Prof.Dr. Ahmet SAYAR
Danışman, Kocaeli Üniversitesi …….
Do.Dr Hikmet Hakan GÜREL
Jüri Üyesi, Kocaeli Üniversitesi …….
Dr.Öğr. Üyesi Adnan ÖZSOY
Jüri Üyesi, Hacettepe Üniversitesi …….

Tezin Savunulduğu Tarih: 16.06.2021

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında blok zincir teknolojisinin tedarik zincirlerinde nasıl kullanabilineceğine ve geliştireceğine dair entegrasyon çalışması yapılmıştır. Hyperledger Fabric platformu ile tedarik zincirine uygulanabilecek geliştirmeler gösterilmiştir.

Yüksek lisans öğrenimim boyunca görüşleri ile çalışmalarıma katkıda bulunan, karşılaştığım zorluklarda desteğini esirgemeyen hocam Prof.Dr. Ahmet SAYAR'a sonsuz teşekkürlerimi sunarım.

Akademik çalışmalarım sırasında tecrübesi ve fikirleriyle desteklerini esirgemeyen Talha OKTAY'a teşekkürlerime sunarım.

Hayatımın her anında yanımda durarak, bana inanarak moral ve güç veren, her anımda sevinçlerimi ve üzüntülerimi paylaşan sevgili aileme teşekkürlerimi sunarım.

Haziran - 2021 Yılmaz DİKİLİTAŞ

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

API	: Application Programming Interface(Uygulama Programlama Arayüzü)
BFT	: Byzantine Fault Tolerance (Bizans Hata Toleransı)
BTC	: Bitcoin
CSS	: Cascading Style Sheets(Basamaklı Stil Şablonları)
DDT	: Dağıtılmış Defter Teknolojisi
DNS	: Domain Name System(Alan Adı Sistemi)
ETH	: Ether
FTP	: File Transfer Protocol(Dosya aktarım Protokolü)
HTML	: Hypertext Markup Language(Hiper Metin İşaretleme Dili)
HTTP	: Hypertext Transfer Protocol (Hiper Metin Transferi Protokolü)
IBM	: International Business Machines (Uluslararası İş Makineleri)
IOT	: Internet of Things(Nesnelerin İnterneti)
JSON	: JavaScript Object Notation(Javascript Obje Gösterimi)
MSP	: Membership Service Providers (Üyelik Servis Sağlayıcıları)
PBHT	: Pratik Bizans Hata Toleransı
POA	: Proof of Authority (Otorite Kanıtı)
POS	: Proof of Stake (Hisse Kanıtı)
POW	: Proof of Work (İşin Kanıtı)
TLS	: Transport Layer Security (Taşıma Katmanı Güvenliği)
XRP	: Ripple

BLOKZİNCİRİN TEDARİK ZİNCİRİNDE KULLANIMI İÇİN TASLAK BİR MİMARİ

ÖZET

Blok zincir her geçen gün popülerliği artan bir teknolojidir. Merkezi olmayan yapısı ile beraber endüstride kullanabilecek çok avantajlı özellikler getirmiştir. Dijital bir kayıt tutma teknolojisi olan blok zincir, tedarik zincirlerin de kullanılırsa çok büyük avantajlar getirebilir. Ürünlerin daha hızlı ve uygun maliyetli bir biçimde son tüketiciye gitmesi sağlanır. İzlenebilirliği artırması sayesinde organizasyonlar arasında koordinasyonu geliştirir. Bu çalışmada merkezi olmayan dağıtılmış bir defter olan blok zinciri teknolojisi kullanılarak, tedarik zincirlerinin güvenlik ve veri gizliliği sorunlarına çözüm bularak, geliştirilen çözümün tedarik zincirlerinde ürün hareketleri problemine bir çözüm olarak uygulanması hedeflenmekte olup simule edilmiştir. Ayrıca tedarik zinciri özelinde, ürünlerin hareketlerinin takibinde, gelen bilgi blok zincir (Blockchain) teknolojisi kullanılarak otomatik ve güvenli bir şekilde takip edilmesini sağlayacak bir sistemin geliştirilmesi ve bir benzetim ortamında gösteriminin yapılmıştır.

Anahtar Kelimeler :Blok Zincir, Hyperledger Fabric, Tedarik Zinciri.

A BLUEPRINT FRAMEWORK FOR USING BLOCKCHAIN IN SUPPLY CHAIN

ABSTRACT

Blockchain is a technology that is increasing in popularity day by day. With its decentralized structure, it has brought very advantageous features that can be used in the industry. Blockchain, which is a digital record keeping technology, can bring great advantages if it is used in supply chains. It improves the coordination between organizations by increasing traceability. In this study, it is aimed and simulated to find a solution to the security and data privacy problems of supply chains by using blockchain technology, which is a decentralized distributed ledger, and to apply the developed solution as a solution to the problem of product movements in supply chains. In addition, in the supply chain, a system was developed and demonstrated in a simulation environment, which will ensure that the incoming information is automatically and securely tracked using blockchain technology.

Keywords :Blockchain, Hyperledger Fabric, Supply Chain.

GİRİŞ

Bitcoin gibi kripto para ağlarının arkasındaki dijital kayıt tutma teknolojisi olan blok zincir, finans dünyasında potansiyel bir oyun değiştiricidir. Ancak büyük umut vaat ettiği bir diğer alan da tedarik zinciri yönetimidir. Blok zincir, ürünlerin daha hızlı ve daha uygun maliyetli teslimatını sağlayarak, ürünlerin izlenebilirliğini artırarak, ortaklar arasındaki koordinasyonu geliştirerek ve finansmana erişime yardımcı olarak tedarik zincirlerini büyük ölçüde iyileştirebilir.

Tedarik zinciri, belirli bir ürünü nihai alıcıya üretmek ve dağıtmak için bir şirket ve tedarikçileri arasında bir ağdır. Bu ağ farklı faaliyetleri, insanları, varlıkları, bilgileri ve kaynakları içerir.Tedarik zinciri aynı zamanda ürünü veya hizmeti orijinal durumundan müşteriye ulaştırmak için atılan adımları da temsil eder[1].

Bir tedarik zinciri, bir ürün veya hizmeti müşteriye ulaştırmak için bir dizi adım içerir. Adımlar, hammaddelerin taşınması ve bitmiş ürünlere dönüştürülmesi, bu ürünlerin taşınması ve son kullanıcıya dağıtılmasını içerir. Tedarik zincirinde yer alan kuruluşlar arasında üreticiler, satıcılar, depolar, nakliye şirketleri, dağıtım merkezleri ve perakendeciler bulunur.

Blok zincir, dağıtılmış veya merkezi olmayan bir defterdir. Birden fazla taraf arasındaki işlemleri doğrulanabilir, kurcalamaya karşı korumalı bir şekilde kaydetmek için dijital bir sistemdir. Defterin kendisi de işlemleri otomatik olarak tetikleyecek şekilde programlanabilir. Kripto para ağları için, blok zincirinin ana işlevi, sınırsız sayıda anonim tarafın merkezi bir aracı olmadan birbirleriyle özel ve güvenli bir şekilde işlem yapmasını sağlamaktır. Tedarik zincirleri için, sınırlı sayıda bilinen tarafın daha iyi performansı desteklerken ticari operasyonlarını kötü niyetli aktörlere karşı korumasına izin vermektir. Tedarik zincirleri için başarılı blok zinciri uygulamaları, yeni izin verilen blok zincirleri, bir bloktaki işlemleri temsil etmek için yeni standartlar ve sistemi yönetmek için yeni kurallar gerektirecektir ve bunların tümü geliştirilmenin çeşitli aşamalarındadır[2].

Ürün hareketinde üretici, satıcı, alıcı, ara kademelerdeki tüm noktalarda ve taşınan malların üzerinde blok zincir teknolojisi ile yönetiminin sağlanması, her işlemin günümüzde olduğu üzere, sürekli olarak farklı firma, ülkelere özel farklı teknolojiler ve manuel süreçler ile izlemesini ve doğrulamasını zorunlu kılmak yerine, hatasız,ispat edilebilir veya inkar edilemez, değiştirilemez, açık ve el ile işlem gerektirmeden, otonom, maliyet etkin olarak, uluslararası bir boyutta takip edilebilecektir.

Blok zincirinin benzersiz avantajı, değişiklik yapamama ve uçtan uca işlem şeffaflığı sağlama yeteneğidir. Defter "yalnızca ek" bir yapıdır, bu nedenle bir kez eklendikten sonra bir kayıt değiştirilemez veya silinemez. Kullanıcıların ürünü, kaynağından başlayarak tüm tedarik zinciri boyunca izlemesine olanak tanır.

Daha güvenli ve daha şeffaf işlemler, dahil olan tüm taraflar arasında güvenin artmasına yol açar. Daha az manuel çalışma ve evrak işlerinde gecikmeleri engeller. Akıllı sözleşmeler (bir programın koduna doğrudan yazılan anlaşma koşullarıyla kendi kendini yürüten sözleşmeler) sayesinde, tüm sözleşme koşulları herhangi bir insan müdahalesi olmadan makine tarafından yerine getirilebilir.

Blok zincir, tedarik zincirlerindeki bilgi hatalarını ortadan kaldırabilir. Bunun nedeni, deftere açık erişime ve tedarik zinciri boyunca öğelerin işlenmesini etiketleme ve kaydetme kullanımına dayanmaktadır. Blok zinciri, değişmez veri kaydı, dağıtılmış depolama ve kontrollü kullanıcı erişimleri kullanarak tedarik zinciri endüstrilerindeki şeffaflık sorunlarını büyük ölçüde iyileştirebilir. Blok zinciri teknolojisinin kullandığı merkezi olmayan dağıtılmış bir sistem, kullanım ömrü boyunca her bir ürünün temel ürün bilgilerini toplayacak, depolayacak ve yönetecektir.

Blok zincir teknolojisi şeffaflığı, ürün yolculuğunu üreticiden müşteriye teslim edildiği noktaya kadar kaydetme ve takip etme yeteneği sunar. Böylece, bilgi tedarik zincirinde yer alan herkes tarafından görülebilir. Bilginin şeffaflığı, tedarik zinciri ağındaki çeşitli taraflar arasındaki güveni artırabilir.

Blok zincir değişmezliği, tüm taraflara ağda kaydedilen tüm faaliyetlerin kötü niyetli aktörlerin elinden arınmış olduğuna dair güvence sağlar. Böyle güçlü bir güvenlik, kriptografi algoritması uygulanarak elde edilir. Tedarik zinciri şeffaflığı, oyuncak ürünlerinde tehlikeli kimyasalların bulunması gibi potansiyel skandalları ortadan

kaldırmanın bir yolu olabilir. Blok zinciri teknolojisinin kayıt ve denetleme işlevleri ve işlemin neredeyse gerçek zamanlı takibi ile mümkün olmaktadır. Sonuç olarak, dış denetçilere duyulan ihtiyaç en aza indirilebilir, dolayısıyla denetim maliyetini azaltabilir.

Blok zincir teknolojisinin uygulanması, özellikle izlenebilirlik bölümünde tedarik zinciri endüstrilerini büyük ölçüde iyileştirebilir. Blok zincir teknolojisini uygulayarak her aktör, hammadde kalitesi, tedarik zinciri boyunca malzeme yolculuğunun zaman damgaları, üretim ve dağıtımda yer alan çeşitli aktörler gibi tedarik zinciri boyunca tüm bilgileri uzaktan izleyebilir.

Blok zincir teknolojisi, tedarik zinciri sürdürülebilirliğini geliştirmek için büyük potansiyeller sunar. Blok zincir teknolojisi uygulaması, tehlikeli ürün ve malzemelerin etkin bir şekilde izlenmesini sağlar, tedarik zinciri boyunca çevresel uyum da iyi bir şekilde izlenebilir. Blok zinciri teknolojisi uygulamasının tedarik zincirini yeşillendirmeye yönelik diğer faydaları şunlardır: Birincisi, blok zincir teknolojisi uygulaması kusurlu malları tam olarak izleyebilir, böylece ürünlerin yeniden işlenmesi ve geri çağrılması ihtiyacını azaltabilir, bu da kaynak tüketimini ve israfı azaltacaktır; İkincisi, blok zincir teknolojisi uygulaması, elektriği uzun mesafelerde iletme ihtiyacını azaltabilir ve ardından uzun mesafeli iletimde boşa harcanan enerjinin büyük bir kısmını kurtarabilir.

Gerçekleştirdiğimiz çalışma kapsamında, tedarik zincirinin içerdiği düğüm noktaları (üretici, satıcı, ara kademeler vs.) ile bir benzetim ortamı oluşturulmuş, bilgilerin blok zincir ile dağıtımı ve kaydını sağlayan bir sistem oluşturularak gösterimi sağlanmıştır. Hyperledger Fabric ortamı ile üstte bahsedilen avantajları gözlemlenmeye çalışılmıştır. Bu kapsamda blok zincir teknolojisi tedarik zinciri göz önüne alınarak kaynak kod bazında değiştirilmiş, benzetim ortamında, lojistik düğümleri, ürünler ile çalışması sağlanmıştır. Söz konusu sistem gösteriminde; gelen her türlü verinin blok zincir ortamında, farklı, anlık olarak dağıtılacak, sergilenen ve sorgulanabilen ürün hareketlerinin tam bir otonomluk içinde takip edilebilmesi sağlanmıştır.

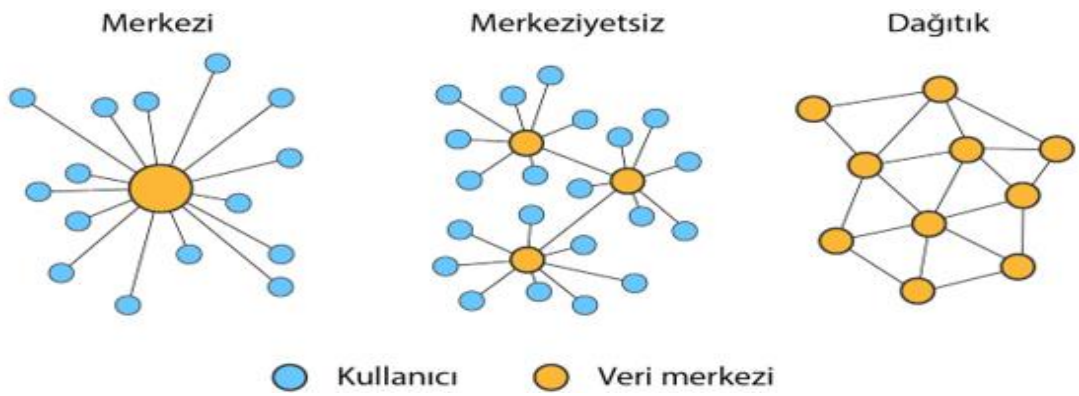
1.GENEL BİLGİLER

1.1. Blok Zincir Teknolojisi

Blok zincir teknolojisi ilk olarak 2008 yılında Satoshi Nakamoto'nun "Bitcoin:A peer-to-peer electronic cash system" adlı makale ile tanıtılmıştır.Bitcoin bu teknoloji üzerine geliştirilen ilk uygulamadır.Merkezi olmayan,bağımsız bir elektronik ödeme sistemidir.Bu uygulama ile blok zincir teknolojisi hayatımıza girmiştir.Her geçen gün daha çok uygulama geliştirilmektedir.

Blok zincir, kriptografi kullanılarak birbirine bağlanan, blok adı verilen ve büyüyen bir kayıt listesidir. Her blok, önceki bloğun kriptografik bir özetini(hash), bir zaman damgasını ve işlem verilerini içerir. Zaman damgası, işlem verilerinin, özete(hash) girmek için blok yayınlandığında var olduğunu kanıtlar. Bloklar, bir zincir oluşturan önceki bloğun özetini(hash) içerir ve her ek blok kendisinden öncekileri güçlendirir. Bu nedenle, blok zincirleri verilerinin değiştirilmesi imkansızdır, çünkü bir kez kaydedildikten sonra, herhangi bir bloktaki veriler, sonraki tüm blokları değiştirmeden geriye dönük olarak değiştirilemez. Şekil 1.1'de örnek blok zincir yapıları gösterilmiştir.

Blok zincir, bir ağ üzerinden dağıtılan halka açık bir defterdir. İşlemleri kaydeden (bir ağdan gönderilen mesajlar düğümden diğerine) ağ katılımcıları arasında yürütülür. Her işlem, blok zincirine eklenmeden önce çoğunluk fikir birliği mekanizmasına(consensus) göre ağ düğümleri tarafından doğrulanır.

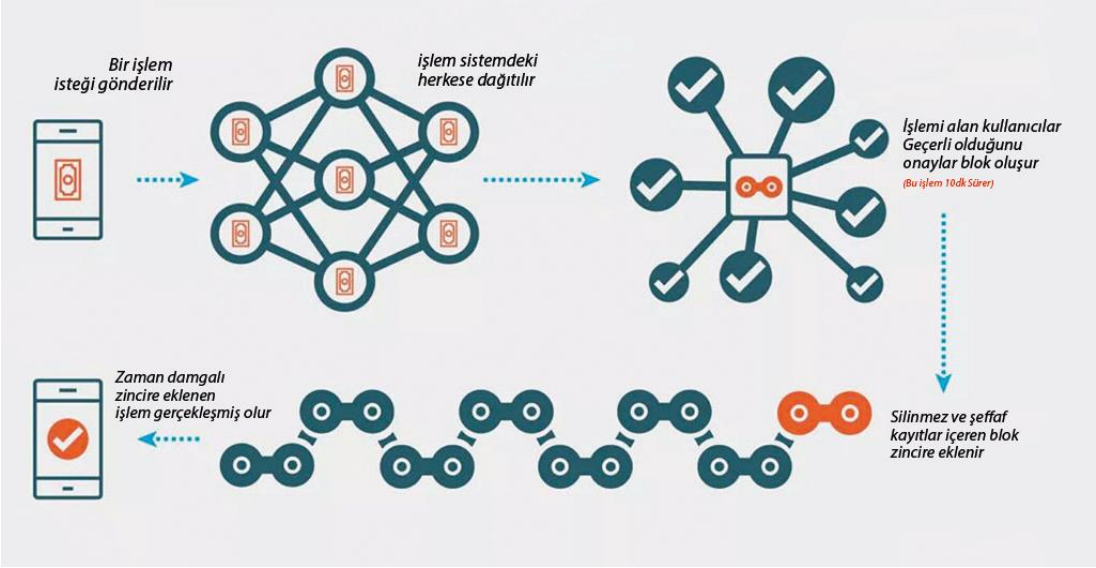


Şekil 1.1. Merkezi ve merkezi olmayan blok zincir yapıları[3]

Merkezi sistemlerin aksine blok zincirin her üye düğümü ağda bulunan tüm verilerin kopyasını elinde bulundurmaktadır (Şekil 1.1) .Kaydedilen bilgiler değiştirilemez veya silinemez ve her işlemin geçmişi herhangi bir zamanda yeniden oluşturulabilir.[4]

Blok zincir teknolojisi bazen uzun bir DNA zinciri olarak temsil edilir ve bu zincir yeni işlemler yapıldıkça sürekli artmaktadır. İşlemler bloklar halinde gruplandırılmıştır ("blok zinciri" adının geldiği yer), her biri sıralı bir şekilde öncekine bağlı olarak sıralanır. Zincir, işlemlerin geçerliliğini doğrulayan ve bunları madencilik ile yeni bloklara ekleyen bir düğüm ağı tarafından korunur.

Bir blok zinciri teknolojisi, uygulamalara defter ve akıllı sözleşme hizmetleri sağlayan teknik bir altyapıdır. Öncelikle akıllı sözleşmeler, daha sonra defterin kopyasına sabit bir şekilde kaydedildikleri ağdaki her eş düğüme dağıtılan işlemleri oluşturmak için kullanılır. Uygulamaların kullanıcıları, istemci uygulamaları veya blok zinciri ağ yöneticileri kullanan son kullanıcılar olabilir.



Şekil 1.2. Blok zincir çalışma örneği[5]

1.1.1. Blok zincir türleri

Blok zinciri sanal bir defter olarak düşünürsek, bu teknoloji bir çok farklı alanda kullanılabilir. Ama blok zincirin halka açık olan yapısı her şey için uygun olmayabilir. Blok zincir teknolojisi, açık blok zincirler, özel blok zincirler, konsorsiyum blok zincirleri olarak 3 türe ayrılır.

1.1.1.1. Açık blok zincirler

Herkese açık bir blok zincirin kısıtlamaları yoktur. İnternet bağlantısı olan herkes ağa erişebilir ve blokları doğrulamaya ve işlem göndermeye başlayabilir. Tipik olarak, bu tür ağlar, blokları doğrulayan kullanıcılar için bir tür teşvik sunma eğilimindedir.

Bu ağ işlemleri doğrulamak için iş ispatı veya hisse ispatı fikir birliği algoritmalarını kullanma eğilimindedir. Gerçek anlamda "açık" bir ağdır.Bu algoritmaları ileri bölümlerde anlatacağız.

Satoshi Nakamoto'nun 2009'da önerdiği model buydu[6]. Daha sonra, kurumsal şirketler blok zincir teknolojisine ilgi göstermeye başladı ve merkezi olmayan defterin doğasını değiştirdi ve özel blok zincirlerini tanıttı.

Açık blok zincir mimarisinde, protokolü istediğiniz zaman indirebilirsiniz ve hiç kimseden izin almanız gerekmez. Bu nedenle, tamamen merkezi olmayan bir sistemdir, ekosistemi tek bir kuruluş kontrol etmez[7].

1.1.1.2. Özel blok zincirler

Yalnızca doğrulanmış katılımcıların seçilen girişine izin veren özel bir blok zinciri çalıştırılması gerekiyorsa, özel bir blok zinciri uygulamasını tercih edebilirsiniz. Bir katılımcı, böyle özel bir ağa yalnızca gerçek ve doğrulanmış bir davet yoluyla katılabilir. Doğrulama, ağ operatörleri tarafından veya ağ tarafından uygulanan açıkça tanımlanmış bir dizi protokol tarafından da gereklidir[8].

Açık ve özel blok zincirleri arasındaki temel ayrım, özel blok zincirlerinin ağa kimlerin katılmasına izin verildiğini kontrol etmesi, madencilik haklarına ve ödüllerine karar veren uzlaşma protokolünü yürütmesi ve paylaşılan defteri tutmasıdır. Ağın sahibi, gerektiğinde blok zincirindeki gerekli girişleri geçersiz kılma, düzenleme veya silme hakkına sahiptir.

Gerçek anlamda, özel bir blok zinciri merkezi olmayan değildir ve kriptografi kavramlarına dayanan kapalı, güvenli bir veritabanı olarak çalışan dağıtılmış bir defterdir. Teknik olarak konuşursak, herkes özel blok zincirinde tam bir düğümü çalıştıramaz, işlem yapamaz veya blok zinciri değişikliklerini doğrulayamaz. Şekil 1.3’de blok zincir türlerinin farkları gösterilmiştir.

	Blockchain türü		
Herkese Açık	Özel	Konsorsiyum	
İzne dayalı mı?	Hayır	Evet	Evet
Kim okuyabilir?	Herkes	Yalnızca davet edilen kullanıcılar	Duruma göre değişir
Kim yazabilir?	Herkes	Onaylanmış katılımcılar	Onaylanmış katılımcılar
Sahiplik	Hiç kimse	Tek bir birim	Birden fazla birim
Katılımcılar bilinir mi?	Hayır	Evet	Evet
İşlem hızı	Yavaş	Hızlı	Hızlı

Şekil 1.3. Özel, Herkese Açık ve Konsorsiyum Blok Zincirlerin Farkları[9]

1.1.1.3. Konsorsiyum blok zincirler

Bir konsorsiyum blok zinciri, bir şekilde özel blok zinciri gibidir. Özel blok zincir gibi verimlilik sağlar ama buna rağmen özel blok zincir gibi tek düğüm tarafından kontrol edilmez. İşlemler düşük maliyetli ve hızlıdır çünkü her düğüm tarafından onaylanmasına gerek yoktur. Sonuç olarak, yeni blok oluşması için ağdaki tüm düğümler değil bazı düğümler tarafından onaylanması gerekir. Bu tarz blok zincirler kripto para gibi blok zincirlerde tercih edilmese de sektörde bir çok şirketin isteyebileceği tarz bir blok zincir türüdür.

1.1.2. Blok zincir yapısı

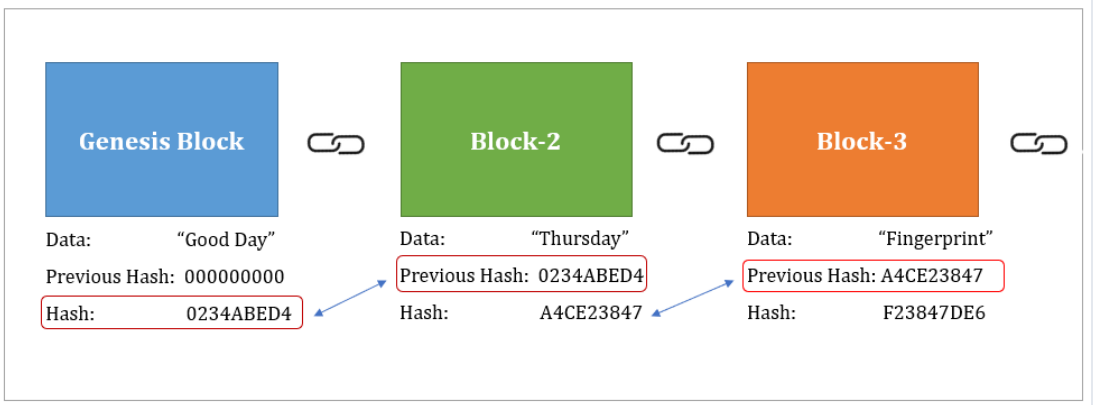
Bir blok zinciri, birçok bilgisayardaki işlemleri kaydetmek için kullanılan blok adı verilen kayıtlardan oluşan merkezi olmayan, dağıtılmış ve çoğu zaman halka açık bir dijital defterdir, böylece ilgili herhangi bir blok, sonraki tüm bloklar değiştirilmeden geriye dönük olarak değiştirilemez. Bir blok zinciri veritabanı, eşler arası bir ağ ve dağıtılmış bir zaman damgası sunucusu kullanılarak özerk olarak yönetilir. [10]

Mantıksal olarak, bir blok zinciri birkaç katmandan oluşuyor olarak görebiliriz. Bunlar:donanım, ağ, uzlaşma(consensus), veri ve uygulamadır.

1.1.3. Bloklar

Bloklar, karma hale getirilmiş ve bir Merkle ağacına kodlanmış geçerli işlem gruplarını tutar. Her blok, blok zincirindeki önceki bloğun kriptografik hashini içerir ve ikisini birbirine bağlar. Bağlı bloklar bir zincir oluşturur. Bu yinelemeli süreç, genesis bloğu olarak bilinen ilk bloğa kadar önceki bloğun bütünlüğünü doğrular. [11]

Bazen ayrı bloklar aynı anda üretilebilir ve geçici bir çatal oluşturabilir. Güvenli bir hash tabanlı geçmişe ek olarak, herhangi bir blok zincir, geçmişin farklı sürümlerini puanlamak için belirli bir algoritmaya sahiptir, böylece daha yüksek puana sahip olanı diğerleri üzerinde seçilebilir. Zincire dahil edilmek üzere seçilmeyen bloklara öksüz bloklar denir. [12] Veritabanını destekleyen eşler, zaman zaman geçmişin farklı sürümlerine sahiptir. Veritabanının yalnızca kendilerinin bildiği en yüksek puanlı sürümünü saklarlar.Her blok SHA256 kriptografik özet(hash) algoritması kullanılarak oluşturulan hash ile tanınabilir.Her blok kendinden bir önceki bloku işaret eder.[13]



Şekil 1.4. Blok Zincirde Teknolojisinde Blok Yapısı[14]

1.1.4. Dağıtık kayıt defteri teknolojisi

Dağıtılmış Defter Teknolojisi(DDT), merkezi olmayan bir dijital veritabanının güvenli çalışmasını sağlayan bir protokoldür. Dağıtılmış ağlar, manipülasyona karşı kontrol sağlamak için merkezi bir otoriteye olan ihtiyacı ortadan kaldırır. DDT, tüm bilgilerin kriptografi kullanılarak güvenli ve doğru bir şekilde depolanmasına izin verir. Aynısına "anahtarlar" ve kriptografik imzalar kullanılarak erişilebilir. Bilgi

depolandığında, değişmez bir veritabanı haline gelir ve ağın kuralları tarafından yönetilir. Dağıtılmış bir defter fikri tamamen yeni değildir ve birçok kuruluş verileri farklı yerlerde tutar. Bununla birlikte, her konum tipik olarak, her birini periyodik olarak güncelleyen bağlı bir merkezi sistem üzerindedir. Bu, merkezi veritabanını siber suçlara karşı savunmasız hale getirir ve merkezi bir organın uzaktaki her bir notu güncellemesi gerektiğinden gecikmelere eğilimli hale getirir[15].

Merkezi olmayan bir defterin doğası, onları bir siber suçtan muaf kılar, çünkü saldırının başarılı olması için ağda depolanan tüm kopyaların aynı anda saldırıya uğraması gerekir. Ek olarak, kayıtların eşzamanlı (eşler arası) paylaşılması ve güncellenmesi, tüm süreci çok daha hızlı, daha etkili ve daha ucuz hale getirir.

Girişimlere ek olarak, IBM ve Microsoft gibi birçok büyük şirket blok zincir teknolojisini deniyor. En popüler dağıtılmış defter protokollerinden bazıları Ethereum, Hyperledger Fabric, R3 Corda ve Quorum'dur[16].Bu protokollerden bazılarını ileride anlatacağız.

1.1.5. Fikir birliği(konsensus) mekanizması

Herhangi bir merkezi sistemde, bir ülkedeki sürücü ehliyetleri hakkında önemli bilgileri tutan bir veritabanı gibi, bir merkezi yönetici, veritabanını koruma ve güncelleme yetkisine sahiptir. Herhangi bir güncelleme yapma görevi belirli lisansları almaya hak kazanan kişilerin adlarını ekleme,silme,güncelleme gibi gerçek kayıtların tutulmasından tek sorumlu olan merkezi bir otorite tarafından gerçekleştirilir.

Merkezi olmayan, kendi kendini düzenleyen sistemler olarak çalışan açık blokzincirleri, tek bir otorite olmadan küresel ölçekte çalışır. Blok zincirinde ve blok madenciliği faaliyetlerinde meydana gelen işlemlerin doğrulanması ve doğrulanması üzerinde çalışan yüz binlerce katılımcının katkılarını içerirler.

Blok zincirinin böylesine dinamik olarak değişen bir durumunda, bu halka açık olarak paylaşılan defterlerin, ağ üzerinde gerçekleşen tüm işlemlerin gerçek olmasını ve tüm katılımcıların bir fikir birliği üzerinde anlaşmasını sağlamak için verimli, adil, gerçek zamanlı, işlevsel, güvenilir ve güvenli bir mekanizmaya ihtiyacı vardır. Bu çok önemli görev, blok zincirinin çeşitli katılımcılarının katkılarına karar veren bir dizi kural olan fikir birliği mekanizması tarafından gerçekleştirilir[17].

Farklı ilkeler üzerinde çalışan farklı fikir birliği mekanizması algoritmaları vardır.Bu algoritmaları bu bölümde anlatılacaktır.

1.1.5.1. İş kanıtı algoritması(Proof of work)

İş kanıtı (PoW), istenmeyen e-postalar göndermek veya hizmet reddi saldırıları başlatmak gibi bilgi işlem gücünün anlamsız veya kötü niyetli kullanımlarını caydırmak için önemsiz olmayan ancak makul miktarda çaba gerektiren bir sistemi tanımlar. Konsept daha sonra 2004 yılında Hal Finney tarafından SHA-256 hash algoritmasını kullanarak "yeniden kullanılabilir iş kanıtı" fikriyle dijital parayı güvence altına almaya uyarlandı[18].

2009 yılında piyasaya sürülmesinin ardından Bitcoin, Finney'nin PoW fikrinin yaygın olarak benimsenen ilk uygulaması oldu (Finney aynı zamanda ilk bitcoin işleminin alıcısıydı)[19]. İş kanıtı, diğer birçok kripto para biriminin de temelini oluşturur ve güvenli, merkezi olmayan bir fikir birliğine izin verir.

Bu açıklama, Bitcoin ağında çalıştığı çalışan iş kanıtı(PoW) üzerine odaklanacaktır. Bitcoin, "blok zinciri" olarak bilinen bir tür dağıtılmış defter tarafından desteklenen dijital bir para birimidir. Bu defter, tüm bitcoin işlemlerinin sıralı "bloklar" halinde düzenlenmiş bir kaydını içerir, böylece hiçbir kullanıcının sahip olduğu parayı iki kez harcamasına izin verilmez. Kurcalamayı önlemek için, defter halka açıktır veya "dağıtılmıştır"; değiştirilmiş bir sürüm diğer kullanıcılar tarafından hızla reddedilecektir.

Kullanıcıların hash değiştirmek için kurcalaması iş kanıtı olarak düşünülebilir.Belirli verileri kullanarak bir hash fonksiyonu koyunca bir hash oluşacaktır. Bununla birlikte, "çığ etkisi" nedeniyle, orijinal verilerin herhangi bir kısmındaki küçük bir değişiklik bile, tamamen tanınmayan bir hash ile sonuçlanacaktır. Orijinal veri kümesinin boyutu ne olursa olsun, belirli bir işlev tarafından oluşturulan hash aynı uzunlukta olacaktır. Hash tek yönlü bir işlevdir: orijinal verileri elde etmek için kullanılamaz, yalnızca hashi oluşturan verilerin orijinal verilerle eşleşip eşleşmediğini kontrol etmek için kullanılabilir[20].

1.1.5.2. Varlık kanıtı algoritması(PoS)

Varlık kanıtı (PoS) kavramı, bir kişinin blok işlemlerini kaç jeton(coin) tuttuğuna göre madencilik yapabileceğini veya doğrulayabileceğini belirtir. Bu, bir madencinin sahip olduğu daha fazla madeni parayla beraber daha fazla madencilik gücüne sahip olduğu anlamına gelir.

Varlık kanıtı, iş kanıtı (PoW) kavramına alternatif olarak, ikincisindeki doğal sorunları çözmek için yaratıldı. Şu anda, yalnızca altcoinler varlık kanıtı konseptini kullanıyor. Bir işlem başlatıldığında, işlem verileri maksimum 1 megabayt kapasiteli bir bloğa yerleştirilir ve ardından ağ üzerindeki birden çok bilgisayar veya düğümde kopyalanır. Düğümler, blok zincirinin yönetim organıdır ve her bloktaki işlemlerin meşruiyetini doğrular[21].

Doğrulama adımını gerçekleştirmek için, düğümlerin veya madencilerin, iş probleminin kanıtı olarak bilinen hesaplamalı bir bulmacayı çözmeleri gerekir. Her blok işlem probleminin şifresini çözen ilk madenci bir jetonla(coin) ödüllendirilir. Bir işlem bloğu doğrulandıktan sonra, halka açık şeffaf bir defter olan blok zincirine eklenir.

1.1.5.3. Pratik bizans hata toleransı – PBHT

Konsensüs modelleri, dağıtılmış blok zinciri sistemlerinin birincil bileşenidir ve kesinlikle işlevsellikleri için en önemlilerinden biridir. Kullanıcıların birbirleriyle güvensiz bir şekilde etkileşime girebilmelerinin bel kemiğidirler ve kripto para birimi platformlarına doğru şekilde uygulanmaları, olağanüstü potansiyele sahip yeni bir ağ yelpazesi yaratmıştır.

Amaç, bu kötü niyetli düğümlerin ağın doğru işlevi üzerindeki etkisini ve sistemdeki dürüst düğümler tarafından ulaşılan doğru fikir birliğini azaltarak yıkıcı sistem arızalarına karşı savunma yapmaktır[22].

Pratik Bizans Hata Toleransı (PHBT) bu optimizasyonlardan biridir ve Miguel Castro ve Barbara Liskov tarafından 1999 yılında “Practical Byzantine Fault Tolerance” başlıklı akademik bir makalede tanıtılmıştır[23].

Bu algoritma fikir birliği mekanizmalarını geliştirmeyi amaçlamaktadır ve bazı popüler blokzincir platformları da dahil olmak üzere birkaç modern dağıtılmış bilgisayar sisteminde uygulanmış ve geliştirilmiştir.

PHBT modeli öncelikle, bağımsız düğüm hataları , bağımsız düğümler tarafından yayılan manipüle edilmiş mesajlar olduğu varsayımıyla Bizans hatalarını (kötü niyetli düğümler) tolere eden pratik bir Bizans durumu makine replikasyonu sağlamaya odaklanır. Algoritma, eşzamansız sistemlerde çalışmak üzere tasarlanmıştır ve etkileyici bir ek yük çalışma süresi ve gecikmede yalnızca küçük bir artış ile yüksek performanslı olacak şekilde optimize edilmiştir.

1.1.5.4. Otorite kanıtı(Proof of authority)

Otorite kanıtı (PoA), blokzincir ağları (özellikle özel olanlar) için pratik ve verimli bir çözüm sunan itibara dayalı bir fikir birliği algoritmasıdır. Terim, 2017 yılında Ethereum[24] kurucu ortağı ve eski CTO Gavin Wood tarafından önerildi.

Otorite kanıtı fikir birliği algoritması, kimliklerin değerini kullanır; bu, blok onaylayıcıların madeni paraları değil, kendi itibarlarını ortaya koyduğu anlamına gelir. Bu nedenle, otorite kanıtı blok zincirleri, güvenilir varlıklar olarak rastgele seçilen doğrulama düğümleri tarafından güvence altına alınır.

Otorite kanıtı modeli sınırlı sayıda blok doğrulayıcıya dayanır ve bu onu oldukça ölçeklenebilir bir sistem yapan şeydir. Bloklar ve işlemler, sistemin moderatörü olarak görev yapan önceden onaylanmış katılımcılar tarafından doğrulanır[25].

Otorite kanıtı modeli çeşitli senaryolarda uygulanabilir ve lojistik uygulamalar için yüksek değerli bir seçenek olarak kabul edilir. Örneğin tedarik zincirleri söz konusu olduğunda, PoA etkili ve makul bir çözüm olarak kabul edilir.

Otorite kanıtı modeli, şirketlerin blok zincir teknolojisinin avantajlarından yararlanırken gizliliklerini korumalarını sağlar. Microsoft Azure, PoA'nın uygulandığı başka bir örnektir. Birkaç kelimeyle Azure platformu, madenciliğe gerek olmadığı için 'gas' gibi yerel para birimi gerektirmeyen bir sistemle özel ağlar için çözümler sunar[26].

1.1.5.5. Delege edilmiş hisse kanıtı(Delegated proof of stake)

Delege edilmiş hisse kanıtı fikir birliği algoritması, 2014 yılında Daniel Larimer tarafından geliştirilmiştir.

Delege edilmiş hisse kanıtı tabanlı bir blok zinciri, paydaşların çalışmalarını bir üçüncü tarafa yaptırdıkları bir oylama sistemi ile sayılır. Başka bir deyişle, ağı kendi adına güvence altına alacak birkaç delegeye oy verebilirler. Delegelerden tanık olarak da bahsedilebilir ve yeni blokların oluşturulması ve doğrulanması sırasında fikir birliğine varmaktan sorumludurlar. Oylama gücü, her kullanıcının sahip olduğu jeton sayısı ile orantılıdır. Oylama sistemi projeden projeye değişir, ancak genel olarak her delege oy isterken bireysel bir teklif sunar. Genellikle delegeler tarafından toplanan ödüller orantılı olarak seçmenleriyle paylaşılır[27].

Bu nedenle, delege edilmiş hisse kanıtı algoritması, doğrudan delegelerin itibarına bağlı olan bir oylama sistemi oluşturur. Seçilen bir düğüm hatalı davranırsa veya verimli bir şekilde çalışmazsa, hızla çıkarılır ve başka bir düğümle değiştirilir.

Performansla ilgili olarak, delege edilmiş hisse kanıtı blok zincirleri, PoW ve PoS ile karşılaştırıldığında saniyede daha fazla işlem işleyebildiğinden daha ölçeklenebilir.

1.1.6. Blok zincir platformları

Blok zincir platformları, blok zincir tabanlı uygulamaların geliştirilmesine izin verir. İzinli veya izinsiz olabilirler. Bitcoin,Ethereum, Hyperledger, R3, Ripple ve EOS, insanların blok zinciri üzerinde uygulamalar geliştirmesine ve barındırmasına olanak tanıyan blok zinciri platformlarından birkaç tanesidir.

1.1.6.1. Bitcoin

Bitcoin, Ocak 2009'da oluşturulan dijital bir para birimidir. Gizemli Satoshi Nakamoto tarafından ortaya konulan fikirleri takip eder.Teknolojiyi yaratan kişi veya kişilerin kimliği hala bir muammadır. Bitcoin, geleneksel çevrimiçi ödeme mekanizmalarından daha düşük işlem ücretleri vaat ediyor ve devlet tarafından verilen para birimlerinin aksine, merkezi olmayan bir otorite tarafından işletiliyor[28].

Bitcoin bir tür kripto para birimidir. Fiziksel bitcoin yoktur, yalnızca herkesin şeffaf erişime sahip olduğu halka açık bir defterde tutulan bakiyeler vardır. Tüm bitcoin

işlemleri, muazzam miktarda hesaplama gücü ile doğrulanır. Bitcoinler herhangi bir banka veya hükümet tarafından verilmez veya desteklenmez, bireysel bitcoinler bir emtia olarak değerli değildir. Yasal ihale olmamasına rağmen, Bitcoin çok popüler ve toplu olarak altcoin olarak adlandırılan yüzlerce başka kripto para biriminin piyasaya sürülmesini tetikledi. Bitcoin genellikle "BTC" olarak kısaltılır.

Bitcoin sistemi, tümü bitcoin kodunu çalıştıran ve blok zincirini depolayan bir bilgisayar koleksiyonudur ("düğümler" veya "madenciler" olarak da adlandırılır). Mecazi olarak, bir blok zinciri bir bloklar koleksiyonu olarak düşünülebilir. Her blokta bir işlem koleksiyonu bulunur. Blok zincirini çalıştıran tüm bilgisayarlar aynı blok ve işlem listesine sahip olduğundan ve bu yeni blokların yeni bitcoin işlemleriyle doldurulduğunu şeffaf bir şekilde görebildiğinden, kimse sistemi aldatamaz.

Bir Bitcoin "düğümü" çalıştırsın ya da çalıştırmasın, herkes bu işlemlerin canlı olarak gerçekleştiğini görebilir. Hileli bir eylemi başarmak için, kötü bir oyuncunun Bitcoin'i oluşturan bilgi işlem gücünün% 51'ini çalıştırması gerekir. Bitcoin'in Ocak 2021 itibariyle yaklaşık 12.000 düğümü var ve bu sayı artıyor ve böyle bir saldırıyı pek olası değil[29].

1.1.6.2. Ethereum

2015 yılında piyasaya sürülen Ethereum, kendi kripto para birimi olan ether için kullanılan açık kaynaklı, blok zincir tabanlı, merkezi olmayan bir yazılım platformudur. Akıllı kontratlar ve dağıtılmış uygulamaların herhangi bir kesinti, dolandırıcılık, kontrol veya üçüncü bir şahsın müdahalesi olmadan oluşturulmasını ve çalıştırılmasını sağlar[30].

Ethereum sadece bir platform değil, aynı zamanda bir blok zincir üzerinde çalışan bir programlama dilidir,geliştiricilerin dağıtılmış uygulamalar oluşturmasına ve yayınlamasına yardımcı olur.

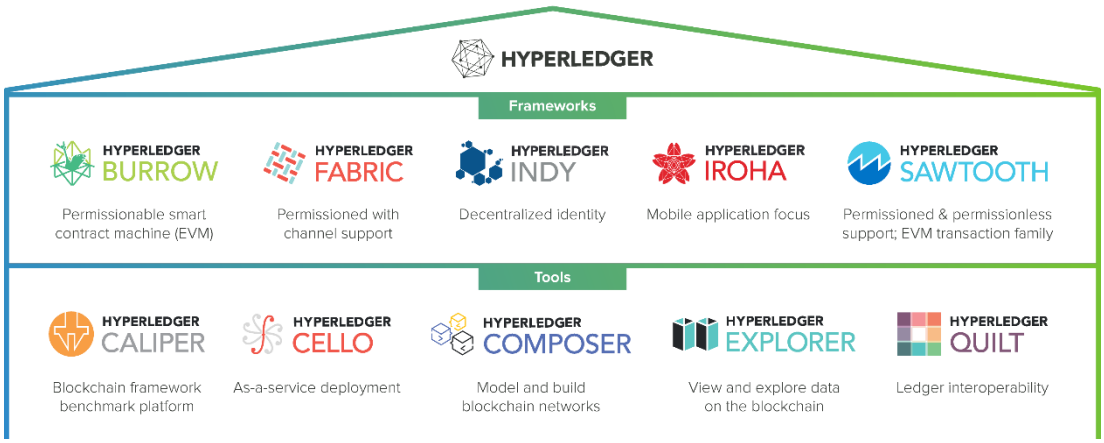
Ethereum ağına katılan herkesin defterin aynı bir kopyasını tutması ve tüm geçmiş işlemleri görmelerine izin vermesi anlamında dağıtılır. Ağın herhangi bir merkezi varlık tarafından çalıştırılmaması veya yönetilmemesi, bunun yerine tüm dağıtılmış defter sahipleri tarafından yönetilmesi nedeniyle merkezi olmayan bir sistemdir.

Blok zincir işlemleri, ağı güvende tutmak ve işlemleri doğrulamak için kriptografi kullanır. İnsanlar, ağdaki her bir işlemi onaylayan ve sistemin kalbindeki blok zincirine yeni bloklar ekleyen karmaşık matematiksel denklemleri "madencilik" yapmak veya çözmek için bilgisayarları kullanır. Katılımcılar kripto para bakiyeleri ile ödüllendirilir. Ethereum sistemi için bu bakiyelere Ether (ETH) denir.

Ether, Bitcoin gibi mal ve hizmet satın almak ve satmak için kullanılabilir. Ayrıca, son yıllarda fiyatında hızlı artışlar görüldü ve bu da onu fiilen spekülatif bir yatırım haline getirdi. Ancak Ethereum ile ilgili benzersiz olan şey, kullanıcıların bir bilgisayarda "çalışan" bir yazılım gibi blok zinciri üzerinde "çalışan" uygulamalar oluşturabilmesidir. Bu uygulamalar kişisel verileri depolayabilir ve aktarabilir veya karmaşık finansal işlemleri yönetebilir.

1.1.6.3. Hyperledger

Aralık 2015'te, Linux Vakfı, Hyperledger Projesi'nin oluşturulduğunu duyurdu. Şubat 2016'da projenin kurucu üyeleri ve on üye daha açıklandı ve yönetim kurulunun yapısı 29 Mart'ta açıklandı.Brian Behlendorf, 19 Mayıs'ta projenin icra direktörlüğüne atandı[31].



Şekil 1.5. Hyperledger Yapısı[33]

Projenin amacı, bu sistemlerin performansını ve güvenilirliğini (karşılaştırılabilir kripto para birimi tasarımlarına kıyasla) iyileştirmeye odaklanarak blok zincirleri ve dağıtılmış defterler geliştirerek sektörler arası işbirliğini ilerletmektir, böylece küresel ticari işlemleri destekleyebilirler. Proje, bağımsız açık protokolleri ve standartları, kendi fikir birliği ve depolama rutinlerine sahip blok zincirleri ve kimlik, erişim

kontrolü ve akıllı sözleşmeler için hizmetler de dahil olmak üzere kullanıma özel modüller için bir çerçeve aracılığıyla entegre ediyor. Önceleri Hyperledger'ın kendi Bitcoin tipi kripto para birimini geliştireceği konusunda bazı karışıklıklar vardı, ancak Behlendorf, Hyperledger Projesi'nin kendisinin asla kendi kripto para birimini oluşturmayacağını kayıtsız şartsız belirtti[32].

Hyperledger hizmet modeli, sunduğu hizmetleri dört ana kategoride değerlendirir. Bu; Kimlik hizmeti, hizmet politikası hizmeti, engelleme hizmeti, akıllı sözleşme hizmetidir. Ek olarak, ulusal iletişim standartları, hizmet grubunda iki yönlü operasyon yoluyla işbirliğine izin verir. Bu sistem çok karmaşıktır, ancak Hyperledger günümüzde birçok blok zinciri projesinde küresel iş ihtiyaçları için temelleri sağlamak için kullanılmaktadır.Şekil 1.5'te Hyperledger'ın yapısı verilmiştir.

1.1.6.4. Corda

Corda, merkezi olmayan bir ağ ortamını desteklemek için verileri işleyen ve kaydeden dağıtılmış defter yazılımıdır. Esas olarak Corda, finans sektörlerine yöneliktir. Corda R3 mimarisi temel olarak Braine, Bakshi ve Clack'in tanımına benzeyen akıllı sözleşmeleri destekler[34].

Corda R3 mimarisi içindeki akıllı sözleşme yapısı, yürütmenin bilgisayar koduna, ancak insan kontrolü ve girdisine bağlı olduğu bir anlaşmadır. Bununla birlikte, herhangi biri isterse veya ihtiyaç duyarsa, bu akıllı sözleşmeleri her zaman yasal olarak uygulayabilir. Corda özel blokzincirinin akıllı sözleşme özelliğinde herhangi bir adaletsizlikle karşılaşırsanız yasal işlem yapabilirsiniz.

Başlangıçta, R3 blok zincir konsorsiyum üyeleri, finans kurumlarının ihtiyaçları için Corda özel blokzincirini tasarladı. Ancak kullanım açısından çok daha geniş olduğu ortaya çıktı. Corda özel blokzincirinde tipik blok zinciri sisteminin büyük benzerliğini görebilirsiniz, ancak gerçek hayattaki durumlarla başa çıkamayacak kadar geleneksel bir tasarıma sahip değildir.

Esas olarak Corda R3 mimarisi, blokzincirin orijinal sürümünü düzenler ve bir yeni bir blokzincir sunar. Corda R3 mimarisinin yapı taşında her biri "durum nesneleri" olarak adlandırılır. Bu nesneler, gerçek dünya sözleşmelerinin bir bölümünü temsil eder veya buradaki sözleşmenin tamamını içerir.

Tipik blok zincirinden farklıdır çünkü sanal makinenin tamamı veya tüm defter sistemi, kullanıcıların fikir birliğine varmasına bağlıdır. Ancak Corda'da işler biraz farklı yapılır.

Corda R3 mimarisi, herkesin tek kaynak noktasına güvenebileceği küresel bir muhasebe sistemi fikriyle başlar. Ancak Corda R3 mimarisi tamamen öyle değildir. Tüm işlemleri bir defterde görülmez, bu nedenle küresel görünürlük Corda özel blokzincirinin özelliği değildir. İki kişi veya grup arasındaki herhangi bir işlem yalnızca onlar tarafından görülebilir ve başka hiç kimse tarafından görülmez. Ancak, fikir birliğine katılacak kullanıcılar, defterin iyiliği için doğrulamaları gerektiği için onları da görebilirler.

1.1.6.5. Ripple

Eskiden OpenCoin olarak bilinen Ripple, dağıtılmış bir defter veritabanının (XRP Ledger) üstüne bir ödeme ve değişim ağı (RippleNet) oluşturan özel bir şirkettir. Ripple'ın temel amacı, bankaları, ödeme sağlayıcılarını ve dijital varlık borsalarını birbirine bağlayarak daha hızlı ve düşük maliyetli küresel ödemeler sağlamaktır[34].

Ripple ilk olarak 2004 yılında, Ripple'ın ilk prototipini merkezi olmayan bir dijital para sistemi (RipplePay) olarak geliştiren Ryan Fugger tarafından geliştirildi. Sistem 2005 yılında hayata geçti ve küresel bir ağ içinde güvenli ödeme çözümleri sağlaması amaçlandı.

2012 yılında Fugger, projeyi Jed McCaleb ve Chris Larsen'e devretti ve birlikte ABD merkezli teknoloji şirketi OpenCoin'i kurdular. Bu noktadan itibaren Ripple, bankalar ve diğer finans kurumları için ödeme çözümlerine odaklanan bir protokol olarak inşa edilmeye başlandı. OpenCoin, 2013 yılında Ripple Labs olarak yeniden markalandı ve daha sonra 2015 yılında Ripple olarak yeniden isim verildi.

Ripple, kendi kripto-para birimine sahip olsa da (XRP) yapı itibari ile para birimlerinden bağımsız bir sisteme sahiptir, üzerinde her türlü para birimi hatta değer ifade eden herhangi bir birim ile işlem yapılabilmektedir.

Ripple platformu üzerinde kullanıcılar ,güvendikleri kullanıcıları ve bu güven yapısı içerisindeki işlem bilgilerini (limit vb.) tanımlamak zorundadırlar. Ripple platformu,

iki kullanıcı arasında yapılan bir para transfer işleminde öncelikli olarak bu iki kullanıcı arasında güvenli bir iletişim kanalı kurmaya çalışır. Direk bir iletişim kanalı oluşturamaması durumunda, kullanıcıların güvendikleri diğer yapıları kullanarak bu iletişimi oluşturmaya çalışır[32].

1.1.7. Blok zincirin avantajları ve dezavantajları

Çoğu blok zinciri, dağıtılmış bir dijital defter olarak işlev gören merkezi olmayan bir veritabanı olarak tasarlanmıştır. Bu blok zinciri defterleri, verileri kronolojik bir sırayla düzenlenmiş ve kriptografik kanıtlarla bağlanmış bloklar halinde kaydeder ve saklar. Blok zincir teknolojisinin yaratılması, çeşitli endüstrilerde birçok avantajı beraberinde getirdi ve güvenilir ortamlarda daha fazla güvenlik sağladı. Bununla birlikte, merkezi olmayan doğası da bazı dezavantajları da beraberinde getirir. Örneğin, geleneksel merkezi veri tabanlarıyla karşılaştırıldığında, blok zincirleri sınırlı verimlilik sunar ve daha fazla depolama kapasitesi gerektirir.

Dağıtık Yapı

Blok zinciri verileri genellikle dağıtılmış bir düğüm ağındaki binlerce cihazda depolandığından, sistem ve veriler teknik arızalara ve kötü niyetli saldırılara karşı oldukça dirençlidir. Her ağ düğümü, veritabanının bir kopyasını kopyalayabilir ve depolayabilir ve bu nedenle tek bir hata noktası yoktur: tek bir düğümün çevrimdışı olması ağın kullanılabilirliğini veya güvenliğini etkilemez.

Bunun aksine, birçok geleneksel veritabanı bir veya birkaç sunucuya dayanır ve teknik arızalara ve siber saldırılara karşı daha savunmasızdır[35].

Kararlılık

Onaylanmış blokların tersine çevrilmesi pek olası değildir, yani veriler blok zincirine bir kez kaydedildikten sonra onu kaldırmak veya değiştirmek son derece zordur. Bu, blokzinciri finansal kayıtların veya denetim takibinin gerekli olduğu diğer verilerin depolanması için harika bir teknoloji haline getirir, çünkü her değişiklik izlenir ve kalıcı olarak dağıtılmış ve halka açık bir deftere kaydedilir.

Örneğin, bir işletme, çalışanlarının dolandırıcı davranışlarını önlemek için blok zincir teknolojisini kullanabilir. Bu senaryoda, blok zinciri, şirket içinde gerçekleşen tüm

finansal işlemlerin güvenli ve istikrarlı bir kaydını sağlayabilir. Bu, bir çalışanın şüpheli işlemleri gizlemesini çok daha zor hale getirir[36].

Verimlilik

Merkezi bir makam aracılığıyla gerçekleştirilen işlemlerin tamamlanması birkaç gün sürebilir. Örneğin, Cuma akşamı bir çek bozdurmaya çalışırsanız, hesabınızda Pazartesi sabahına kadar parayı göremeyebilirsiniz. Finans kuruluşları iş saatlerinde, haftanın beş günü faaliyet gösterirken, blok zincir günde 24 saat, haftada yedi gün ve yılda 365 gün çalışıyor. İşlemler on dakika kadar kısa bir sürede tamamlanabilir ve birkaç saat sonra güvenli kabul edilebilir. Bu, özellikle saat dilimi sorunları ve tüm tarafların ödeme işlemlerini onaylaması gerektiği gerçeği nedeniyle genellikle daha uzun süren sınır ötesi ticaretler için kullanışlıdır.

Şeffaflık

Çoğu blok zinciri tamamen açık kaynaklı yazılımdır. Bu, kodu herkesin ve herkesin görebileceği anlamına gelir. Bu, denetçilere güvenlik için Bitcoin gibi kripto para birimlerini inceleme yeteneği verir. Bu aynı zamanda Bitcoin'in kodunu kimin kontrol ettiği veya nasıl düzenlendiği konusunda gerçek bir otorite olmadığı anlamına gelir. Bu nedenle, herkes sistemde değişiklik veya yükseltme önerebilir. Ağ kullanıcılarının çoğunluğu, yükseltme ile kodun yeni sürümünün sağlam ve değerli olduğu konusunda hemfikir olursa, Bitcoin güncellenebilir.

Dezavantajları

Teknoloji Maliyeti

Blok zincir, kullanıcıların işlem ücretlerinden tasarruf etmesine rağmen, teknoloji ücretsiz olmaktan çok uzaktır. Örneğin, Bitcoin'in işlemleri doğrulamak için kullandığı "iş kanıtı" sistemi, büyük miktarda hesaplama gücü tüketir. Gerçek dünyada, Bitcoin ağındaki milyonlarca bilgisayarın gücü, Danimarka'nın yıllık olarak tükettiğine yakındır[37].

Bitcoin madenciliği maliyetlerine rağmen, kullanıcılar blok zincirindeki işlemleri doğrulamak için elektrik faturalarını artırmaya devam ediyor. Bunun nedeni,

madencilerin Bitcoin blok zincirine bir blok eklediklerinde, zamanlarını ve enerjilerini değerli kılmak için yeterli Bitcoin ile ödüllendirilmeleridir. Ancak, kripto para kullanmayan blok zincirleri söz konusu olduğunda, işlemlerin doğrulanması için madencilere ödeme yapılması veya başka bir şekilde teşvik edilmesi gerekecektir.

Bu sorunlara bazı çözümler ortaya çıkmaya başlıyor. Örneğin, Bitcoin madenciliği çiftlikleri güneş enerjisi, kırma alanlarından gelen fazla doğal gaz veya rüzgar çiftliklerinden gelen gücü kullanmak üzere kurulmuştur.

Yasa Düzenlemeleri

Kripto alanındaki pek çok kişi, kripto para birimleri üzerindeki devlet düzenlemeleri hakkındaki endişelerini dile getirdi. Merkezi olmayan ağı büyüdükçe Bitcoin gibi bir şeyi bitirmek gittikçe zorlaşırken ve neredeyse imkansız hale gelirken, hükümetler teorik olarak kripto para birimlerine sahip olmayı veya ağlarına katılmayı yasadışı hale getirebilirler.

Depolama

Blok zincir kayıt defterleri zamanla çok büyük hale gelebilir. Bitcoin blok zinciri şu anda yaklaşık 200 GB alana ihtiyaç duyar. Blok zincir boyutlarındaki mevcut büyüme sabit belleklerdeki büyümenin üzerindedir ve kayıt defteri bireylerin indirip ve saklayabilmesi için fazla büyük hale gelirse ağ düğümü kaybetme riskiyle karşı karşıya kalabilir.

1.1.8. Blok zincir kullanım alanları

1.1.8.1. Ekonomi ve finans

Blok zincir teknolojisi, kripto para birimi adı verilen finansal işlemler için yaygın olarak kullanılmaktadır. Kripto para birimleri son yıllarda öne çıkan yazılım sistemleri haline geliyor.Kripto para, ürün ve hizmet satın almak ve satmak için kullanılabilen bir çevrimiçi ödeme şeklidir. Birçok işletme, işletmenin sunduğu mallar veya hizmetler için değiştirilebilen, jeton(coin) olarak bilinen kendi para birimlerini yarattı. Ürün veya hizmeti kullanmak için, kripto para birimi karşılığında gerçek para ticareti yapmanız gerekir.

Bir piyasa analizi web sitesi olan Coinmarketcap.com'a göre, 9,700'den fazla ayrı kripto para birimi halka açık olarak işlem görüyor[38]. Ve kripto para birimleri, fonları artırmak için kullanılmaya devam ederek popülerlik kazanmaya devam ediyor. CoinMarketCap'e göre, tüm kripto para birimlerinin toplam değeri 2021 Mayıs ayında 2,2 trilyon dolardan fazlaydı.En yaygın dijital para birimi olan tüm Bitcoinlerin toplam değeri 1 trilyon dolardan fazlaydı.

İşlem, kredi, ipotek ve ödeme hizmetleri temel bankacılığın bir parçasıdır. Bu hizmetlerin çoğu eski yürütme süreçlerine bağlıdır. Bireyler 30 ila 60 gün içinde bir ipotek alabilir ve küçük ve orta ölçekli işletmeler, bilgi onayına, kredi puanlamasına, kredi işleme ve dağıtım fonlarına bağlı olarak 60 ila 90 gün içinde iş kredisi alabilirler [39]. Blok zincir teknolojisi ile karşı taraf riskini azaltarak ve ihraç ve ödeme sürelerini azaltarak bankacılık ve borç verme hizmetlerini basitleştirir.

1.1.8.2. Sağlık hizmetleri

Sağlık hizmetleri açısından, özel ağlar artık kamusal ağlardan daha gelişmiş durumdadır. Blok zinciri teknolojisinin bir şeyleri değiştirme potansiyeline sahip olduğu bir yer burası. Hasta verilerinin güvenli şifrelenmesi ve salgın hastalıkların hafifletilmesi dahil olmak üzere çok çeşitli görevleri yerine getirebilir. Estonya, 2012 yılında sağlık hizmetlerinde blok zincir teknolojisini uygulayan bu alanda bir öncüdür. Şu anda, blok zincir tüm sağlık hizmetleri faturalandırmasını, sağlık kayıtlarının yüzde 95'ini ve ilaç ayrıntılarının yüzde 99'unu yönetmek için kullanılmaktadır [40]. Sağlık hizmetlerinde blok zincir uygulamalarıyla ilgili sorunlar; ağ altyapısı güvenliği, kimlik doğrulama ve elektronik sağlık bilgilerine erişim için kimlik doğrulama ve yetkilendirmedir.

Sağlık hizmeti veri tabanı yönetim sistemlerinin geleneksel yöntemlerine göre blokzincirleri kullanmanın faydaları arasında merkezi olmayan yönetim, değiştirilemez veritabanları, veri kaynağı, izlenebilir veriler, sağlam veriler, verilerin herhangi bir yetkili kullanıcı tarafından kullanılabilirliği ve yetkisiz kullanıcıların elinden uzak tutulması da vardır.

1.1.8.3. Akıllı sözleşmeler

Akıllı sözleşmeler, önceden belirlenmiş koşullar karşılandığında çalışan bir blokzincirinde depolanan programlardır. Genellikle bir anlaşmanın yürütülmesini basitleştirmek için kullanılırlar, böylece her iki taraf da herhangi bir aracıya ihtiyaç duymadan veya zaman kaybına gerek kalmadan sonuçtan hemen emin olabilir. Ayrıca, bu koşullar karşılandığında bir sonraki adıma başlayarak bir iş akışını otomatikleştirebilirler[41].

Akıllı sözleşmelerin işe yaraması için basit ifadeler bir blok zincirinde koda yazılır. Önceden belirlenmiş koşullar karşılandığında ve test edildiğinde, faaliyetler bir bilgisayar ağı tarafından yürütülür. Bu faaliyetler, uygun taraflara para transferini, bir arabayı kaydettirmeyi, bildirim göndermeyi veya bir bilet düzenlemeyi içerebilir. İşlem tamamlandığında, blok zinciri değiştirilir. Bu, işlemin tersine çevrilemeyeceği ve sonuçların yalnızca izin verilenler tarafından görülebileceği anlamına gelir.

Akıllı sözleşmelerin faydaları verimlilik, doğruluk, şeffaflık ve güvenliktir çünkü bu, blok zinciri operasyonlarına entegre etmek isteyen çoğu şirkete fayda sağlar.

1.1.8.5. Nesnelerin interneti(IOT)

Nesnelerin İnterneti (IoT) insanları, yerleri ve nesneleri birbirine bağlayarak değer yaratma ve yakalama fırsatları yaratır. Nesneler, her biri verileri IoT ağına ileten gelişmiş çipler, sensörler ve aktüatörlerle gömülüdür. Bununla birlikte, henüz çözülmemiş bir dizi teknolojik ve güvenlik sorunu vardır.

Bir siber saldırı durumunda, IoT ağı, veri işlemlerini farklı kuruluşların sahip olduğu ve yönettiği çeşitli cihazlar aracılığıyla işleyecek ve bu da herhangi bir veri sızıntısı kaynağının izlenmesini imkansız hale getirecektir. Dahası, IoT büyük miktarda veri üretir ve bu verilerin mülkiyeti, dahil olan çeşitli paydaşlar nedeniyle her zaman şeffaf değildir[42].

IoT verilerini depolamak için blok zinciri kullanmak, bilgisayar korsanlarının ağa erişmek için etrafından dolaşması gereken ek bir şifreleme katmanı sağlayacaktır. Blok zincir teknolojisi, çok daha yüksek bir şifreleme derecesi sunarak, mevcut veri kayıtlarının üzerine yazmayı neredeyse imkansız hale getiriyor. Blok zincir, ağa erişim

izni olan herkesin önceki işlemleri görüntülemesine ve izlemesine izin vererek şeffaflık sunar.Bu, herhangi bir veri parçasının kökünü tam olarak belirlemek ve anında düzeltici adımlar atmanın güvenilir bir yolu olabilir.

1.1.8.6. Tedarik zincirleri

Bir tedarik zinciri, tedarikçiden tüketiciye ürün veya hizmet alınmasında yer alan kuruluşlar, kişiler, faaliyetler, bilgiler ve kaynaklardan oluşan bir koleksiyondur. Nakliye sürecinde önemli ürünleri iyi durumda tutmak için yapılmıştır. Yolsuzluk, dolandırıcılık ve kurcalama, merkezi tedarik zinciri yönetim sistemleriyle ilişkili risklerdir[43]. Blok zincir, modern bir dağıtılmış bilgi sistemidir.Emtia akışlarının görünürlüğünün ve hesap verilebilirliğinin büyük zorluklar olduğu tedarik zincirinde yeni bir çözümü yansıtır.

Bir tedarik zincirinde blokzincirinin faydaları vardır. Bunlar gelişmiş koruma ve izlenebilirliktir. Hiçbir belge kaybedilemez, yok edilemez veya değiştirilemezdir. Ayrıca, blok zinciri tabanlı bir çerçeve, belgeleri veya ürünleri çalmak için yanlış tanımlayıcıların kullanılması olasılığını ortadan kaldırır. İşlemler daha güvenli ve daha basittir, bu da dahil olan tüm taraflar için artan güven ile sonuçlanır; daha az manuel çalışma ve evrak işlerinde gecikme yoktur. Akıllı sözleşmeler sayesinde tüm sözleşme koşulları hiçbir insan müdahalesi olmadan makine tarafından karşılanabilir.

1.2. Tedarik Zinciri ve Blok Zincir ile İlişkisi

Tedarik zinciri, belirli bir ürünü üretmek ve nihai alıcıya dağıtmak için bir şirket ile tedarikçileri arasında bir ağdır. Bu ağ, farklı etkinlikleri, kişileri, varlıkları, bilgileri ve kaynakları içerir. Tedarik zinciri aynı zamanda ürünü veya hizmeti orijinal halinden müşteriye ulaştırmak için attığı adımları temsil eder.

Şirketler, maliyetlerini düşürmek ve iş ortamında rekabetçi kalabilmek için tedarik zincirleri geliştirir. Tedarik zinciri yönetimi çok önemli bir süreçtir çünkü optimize edilmiş bir tedarik zinciri daha düşük maliyetler ve daha hızlı bir üretim döngüsü ile sonuçlanır.

Bir tedarik zinciri, müşteriye bir ürün veya hizmet sağlamak için bir dizi adım içerir. Adımlar, hammaddelerin taşınması ve bitmiş ürünlere dönüştürülmesi, bu ürünlerin taşınması ve son kullanıcıya dağıtılmasını içerir. Tedarik zincirinde yer alan kuruluşlar

arasında üreticiler, satıcılar, depolar, nakliye şirketleri, dağıtım merkezleri ve perakendeciler yer alırlar.

Bir tedarik zincirinin öğeleri, müşterinin talebini karşılamak için bir sipariş almakla başlayan tüm işlevleri içerir. Bu işlevler arasında ürün geliştirme, pazarlama, operasyonlar, dağıtım ağları, finans ve müşteri hizmetleri bulunur.

Tedarik zinciri yönetimi, iş sürecinin çok önemli bir parçasıdır. Bu zincirde beceri ve uzmanlık gerektiren birçok farklı bağlantı vardır. Tedarik zinciri yönetimi etkili olduğunda, bir şirketin genel maliyetlerini düşürebilir ve karlılığı artırabilir. Bir bağlantı bozulursa, zincirin geri kalanını etkileyebilir ve maliyetli olabilir.

Tedarik zincirlerinin evrimi ve artan verimliliği, enflasyonun kontrol altına alınmasında önemli bir rol oynadı. Ürünlerin A'dan B'ye taşınmasındaki verimlilik arttıkça, bunu yaparken maliyetler düşer ve bu da tüketiciye nihai maliyeti düşürür. Deflasyon genellikle olumsuz olarak görülürken, tedarik zinciri verimliliği deflasyonun iyi bir şey olduğu birkaç örnekten biridir.

Küreselleşme devam ederken, tedarik zinciri verimliliği daha optimize hale gelir ve bu da girdi fiyatları üzerindeki baskıyı sürdürür.

Temelde blok zincirler açık ve özel blok zincirler iki türde bulunur.Bu fark tedarik zincirinde kullanımı açısından önemlidir.

Çoğu durumda, bugünün tedarik zincirleri, blok zincir teknolojisi olmadan geniş ölçekte çalışmaktadır. Yine de bu teknoloji, tedarik zinciri dünyalarını için önemli gelişmelere önayak oldu. Ayrıca pek çok makaleye ilham kaynağı oldu ve girişimcileri gelecek vaat eden pilot projeleri başlatmaya teşvik etti.

Blok zincirin tedarik zincirlerine nasıl değerler katabileceği her geçen gün daha iyi gözlemlendi.Yavaş, manuel işlemlerin değiştirilmesi bu değerlerden biridir. Tedarik zincirleri halihazırda büyük, karmaşık veri kümelerini işleyebilse de, süreçlerinin çoğu, özellikle de alt tedarik kademelerindekiler, yavaştır ve nakliye endüstrisinde hala yaygın olduğu gibi, tamamen kağıda dayanmaktadır. İzlenebilirliğin güçlendirilmesi bu değerlerden başkasıdır. Provenans bilgisine yönelik artan yasal düzenleme ve tüketici talebi, halihazırda değişimi yönlendirdi. Ayrıca, izlenebilirliğin

iyileştirilmesi, geri çağırma, itibar hasarı veya kara veya gri borsa ürünlerinden kaynaklanan gelir kaybı gibi kalite sorunlarının yüksek maliyetlerini azaltarak değer katar. Karmaşık bir tedarik tabanını basitleştirmek, daha fazla değer yaratma fırsatı sunar. Bir diğer yararı ise tedarik zinciri bilişim işlem maliyetlerini düşürmektir. Bu aşamada, bu fayda gerçek olmaktan çok teoriktir. Bitcoin, insanlara her bloğu veya işlemi doğrulaması için ödeme yapar ve yeni bir blok teklif eden kişilerin tekliflerine bir ücret eklemesini gerektirir. Böyle bir maliyet, tedarik zincirlerinde büyük olasılıkla engelleyici olacaktır çünkü ölçekleri şaşırtıcı olabilir. Ayrıca, tüm bu işlemler birlikte, blok zincirinin dağıtılmış defter yaklaşımının önemli bir bileşeni olan veri depolama talebini önemli ölçüde artıracaktır. Ek olarak, veri setlerinin sayısız kopyasını oluşturmak ve sürdürmek, tedarik zinciri ortamında, özellikle de izinsiz blok zincirlerinde pratik olmayacaktır.

Tedarik zinciri için blok zincir teknolojisini benimseyen bir şirket, öncelikle inşa etmesi gereken blok zincir türüne karar vermelidir. Bitcoin yaklaşımının, bilinmeyen veya güvenilmeyen taraflarla doldurulmuş, izinsiz bir blok zinciri olduğunu hatırlamak gerekir. Açık alanda bulunur ve her blokta güven oluşturmak için bir fikir birliği doğrulama protokolü kullanır. Bu blok zincirlerinde merkezi bir veritabanı veya merkezi yönetim yoktur.

Tersine, çoğu tedarik zincirinde taraflar bilinir ve güvenilirdir. Dahası, tedarik zinciri dünyasının açık erişimi kabul etmesi pek olası değildir çünkü kullanıcıları talep, kapasiteler, siparişler, fiyatlar, marjlar gibi özel ayrıntıları, değer zincirinin tüm noktalarında bilinmeyen katılımcılara açıklamak istemezler. Bu, tedarik zinciri blok zincirlerinin çoğunun, erişimin merkezi olarak yönetildiği ve belirli veri segmentleriyle sınırlı olabilecek bilinen taraflarla sınırlı olduğu için izin alınması gerektiği anlamına gelir.

Teoride, bu yaklaşım önerilen her bloğun genel veya özel doğrulamasına izin verir. Ancak, tüm taraflar tanındığında tedarik zinciri dünyasında önerilen engellemelerin kamuya açık bir şekilde doğrulanmasını görmemizin olası olmadığına inanılır. Örneğin, nakliyatta, her bloğun doğrulanmasından sorumlu olan, zincirde taşıyıcılar, limanlar, gümrükler, nakliye hatları dahil olmak üzere yalnızca birkaç bilinen taraf

vardır. Güvenilir tarafların sayısı az olduğunda, kamusal alanda kullanılan fikir birliği protokollerini bağımsız olarak doğrulama ihtiyacı sınırlıdır.

Çoğu durumda, tedarik zincirleri halihazırda milyarlarca işlem ve veriyi genellikle gerçek zamanlı olarak hareket ettirir. Sistemler mükemmel değildir ve birçok tedarik zincirinde büyük veri bağlamında gruplanmış, farklı biçimlendirilmiş, erişilmesi zor veya görselleştirilmesi veya analiz edilmesi zor verilerle ilgili sorunlar vardır. Böyle olsa bile, iyi veri yönetimine sahip, iyi yönetilen merkezi veritabanları, tedarik zinciri görselleştirme ve analitik becerilerle birleştirildiğinde bugün büyük ölçekte elde edilebilir.

Bu çözümler, blok zincirinin ortaya çıkardığı bazı teknik karmaşıklıkların ek yükünü taşımazlar. Bu nedenle, genişletilmiş tedarik zincirlerindeki tüm taraflar tanındığında ve güvenilir olduğunda, bu bilinen ve güvenilir taraflara gerçeğin tek bir gerçek zamanlı versiyonunu sağlamak için güvenilebileceğinden, bir blok zinciri çözümüne muhtemelen ihtiyaç duyulmayacağını savunulur. Böyle bir durumda, bir bulut sunucusu veya merkezi olmayan eşler arası bağlantılar gibi merkezi çözümler yeterli olacaktır.

Katılımcıların bilinmediği veya güvenilmediği tedarik zincirleri için blok zincir teknolojisi güven, şeffaflık ve izlenebilirlik katabilir. Neredeyse tanım gereği, bu tedarik zincirleri karmaşıktır, çok kademelidir, birçok tarafı içerir ve daha yüksek düzeyde izlenebilirlik gerektiren düzenlenmiş bir ortamda çalışırlar.

Bununla birlikte, bilinen ve güvenilir oyunculara sahip tedarik zincirleri için, merkezi bir veritabanı yaklaşımı genellikle fazlasıyla yeterlidir. Bu, tüm bu tedarik zincirlerinin şu anda gerçek bir uçtan-uca yaklaşımı izlediği anlamına gelmez ve aslında birçoğunun yalnızca sınırlı izlenebilirliğe sahip verileri içeren veritabanları kullandığı anlamına gelmez. Bu nedenle, bu tedarik zincirlerinin çoğu, kendi başlarına veya ortaklarıyla yüksek hacimli işlemlerine daha uygun olan mevcut teknolojilerden yararlanabildiklerinden, bu tür sorunları çözmek için blok zinciri teknolojisine ihtiyaç duymazlar.

Tedarik zinciri dünyasında blok zincir teknolojisini çalıştırmanın maliyetlerini tahmin etmek ve bunları diğer teknolojilerle karşılaştırmak için henüz çok erkendir.

Bununla birlikte, değer önerisi açık olmalıdır. Dahili işlem verimlilikleri nelerdir? Son ürün arızalarında, geri çağırmalarda ve davalarda potansiyel maliyet nedir? Bir tüketici, tedarik zinciri boyunca şeffaflık sunan bir ürün için daha fazla ödeyecek mi? Tedarik zincirlerinde kullanım için blok zinciri düşünüldüğünde bu tür sorular sorulmalıdır.

2.MALZEME VE YÖNTEM

2.1. Kullanılacak Teknolojiler

2.1.1. Hyperledger

Hyperledger, çeşitli sektörlerde kullanılmak üzere açık kaynaklı blok zincirleri ve ilgili uygulamaları oluşturmak için gerekli çerçeveyi, standartları, yönergeleri ve araçları sunan küresel bir kurumsal blok zinciri projesidir. Hyperledger'in projeleri, ağ katılımcılarının birbirleri tarafından bilindiği ve bu nedenle fikir birliği oluşturma sürecine katılmaya içsel bir ilgiye sahip olduğu çeşitli kurumsal kullanıma hazır izinli blok zinciri platformlarını içerir[44].

Hyperledger, çeşitli işlerin verimliliğini, performansını ve işlemlerini artırmak için çeşitli endüstri sektörlerinde kullanılabilecek yüksek performanslı ve güvenilir blokzinciri ve dağıtılmış defter tabanlı teknoloji çerçevesi geliştirmek için endüstri çapında işbirliğini hızlandırmak amacıyla kuruldu.

Hyperledger, finans, bankacılık, nesnelerin interneti, tedarik zinciri yönetimi, üretim ve üretim ve teknoloji alanlarından önde gelen işletmeleri içeren küresel bir işbirliğidir.

2.1.2. Hyperledger fabric

Hyperledger Fabric, özel kuruluşlarda kullanılması amaçlanan tak ve çalıştır bileşenleri kullanarak blok zinciri tabanlı ürünler, çözümler ve uygulamalar geliştirmek için bir temel görevi gören modüler bir blok zinciri çerçevesidir.

Hyperledger Fabric, Digital Asset ve IBM tarafından başlatıldı ve şu anda Linux Vakfı tarafından barındırılan, işbirliğine dayalı bir sektörler arası girişim olarak ortaya çıktı[45].

Geleneksel blok zincir ağları, işletmeler için son derece önemli olan özel işlemleri ve gizli sözleşmeleri destekleyemez. Hyperledger Fabric, endüstriyel blok zinciri

çözümleri sunmak için modüler, ölçeklenebilir ve güvenli bir temel olarak buna yanıt olarak tasarlanmıştır.

Hyperledger Fabric, açık kaynaklı bir yazılımdır ve iş kullanım durumlarında blokzinciri değerlendirmek ve kullanmak için en önemli özelliklerle ilgilenir.

Özel endüstriyel ağlarda, bir katılımcının doğrulanabilir kimliği birincil gerekliliktir. Hyperledger Fabric, izne dayalı üyelikleri destekler; tüm ağ katılımcılarının bilinen kimlikleri olmalıdır. Sağlık ve finans gibi birçok iş sektörü, çeşitli katılımcılar ve bunların çeşitli veri noktalarına erişimiyle ilgili verilerin korunmasını zorunlu kılan veri koruma düzenlemelerine tabidir. Fabric, bu tür izin tabanlı üyeliği destekler.

Hyperledger Fabric'in modüler mimarisi, işlem işleme iş akışını üç farklı aşamaya ayırır: dağıtılmış mantık işlemeyi ve sistemin anlaşmasını, işlem siparişini ve işlem onaylama ve taahhüdünü içeren zincir kodu adı verilen akıllı sözleşmelerdir. Bu ayrım, birçok fayda sağlar.Bu faydalardan bazıları;ağı karmaşıklıktan uzaklaştırma,doğrulama, gelişmiş ağ ölçeklenebilirliği ve daha iyi performanstır.

Ek olarak, Hyperledger Fabric'in çeşitli bileşenlerin tak ve çalıştır desteği, mevcut özelliklerin kolayca yeniden kullanılmasına ve çeşitli modüllerin hazır entegrasyonuna olanak tanır. Örneğin, katılımcının kimliğini doğrulayan bir işlev zaten mevcutsa, kurumsal düzeydeki bir ağın aynı işlevi sıfırdan inşa etmek yerine bu mevcut modülü takıp yeniden kullanması yeterlidir.

Fabric, kısıtlı etki alanına sahip özel diller yerine Java, Go ve Node.js gibi genel amaçlı programlama dillerinde yazılmış akıllı sözleşmeleri destekleyen ilk dağıtılmış defter platformlarından biridir. Bu özellik, çoğu kuruluşun hali hazırda kullandığı ve teknolojilerine hakim olduğu programlama dilleriyle akıllı sözleşmeler geliştirebileceği, yeni bir programlama dili öğrenmesi gerekmediği anlamına gelmektedir.

2.2. Geliştirme Ortamı

2.2.1. Visual studio code

Visual Studio Code, Java, JavaScript, Go, Node.js, Python ve C ++ dahil olmak üzere çeşitli programlama dilleriyle kullanılabilen bir kaynak kodu düzenleyicisidir.

Bir proje sistemi yerine, kullanıcıların bir veya daha fazla dizini açmasına izin verir ve bunlar daha sonra tekrar kullanılmak üzere çalışma alanlarına kaydedilebilir. Bu, herhangi bir dil için dilden bağımsız bir kod düzenleyicisi olarak çalışmasına izin verir. Bir dizi programlama dilini ve dile göre değişen bir dizi özelliği destekler. İstenmeyen dosyalar ve klasörler ayarlar aracılığıyla proje ağacından çıkarılabilir. Birçok Visual Studio Code özelliği, menüler veya kullanıcı arabirimi aracılığıyla gösterilmez, ancak komut paleti aracılığıyla erişilebilir[46].

Visual Studio Code, merkezi bir depo aracılığıyla kullanılabilen uzantılar aracılığıyla genişletilebilir. Bu, editöre ve dil desteğine yapılan eklemeleri içerir. Dikkate değer bir özellik, yeni diller, temalar ve hata ayıklayıcılar için destek ekleyen uzantılar oluşturma, statik kod analizi gerçekleştirme kullanarak kod linterleri ekleme yeteneğidir.

Bu projede Visual Studio Code geliştirme ortamı kullanılacaktır. Yukarıda belirtilen kolaylıkların yanı sıra IBM şirketi tarafından blok zincir teknolojisi için geliştirilmiş IBM Blockchain Platform eklentisinin de Visual Studio Code içerisinde bulunması projenin Visual Studio Code üzerinde gerçeklenmesi için gerekli ortamı sağlamaktadır. Bu eklenti geliştiricilerin akıllı sözleşmeler oluşturmasına, test etmesine ve hatalarını ayıklamasına, Hyperledger Fabric ortamlarına bağlanmasına ve blok zinciri ağımızda işlem yapan uygulamalar oluşturulmasına yardımcı olmaktadır.

2.2.2. Sunucu

Sunucu, bir ağ üzerinden yapılan istekleri kabul eden ve bunlara yanıt veren bir yazılım veya donanım aygıtıdır. İsteği yapan ve sunucudan yanıt alan cihaza istemci adı verilir. İnternette "sunucu" terimi genellikle bir web dosyalarına yönelik istekleri alan ve bu dosyaları istemciye gönderen bilgisayar sistemini ifade eder.

Sunucular ağ kaynaklarını yönetir. Örneğin, bir kullanıcı bir ağa erişimi kontrol etmek, e-posta göndermek / almak, yazdırma işlerini yönetmek veya bir web sitesini barındırmak için bir sunucu kurabilir. Ayrıca yoğun hesaplamalar yapma konusunda da uzmandırlar. Bazı sunucular, genellikle adanmış olarak adlandırılan belirli bir göreve bağlıdır. Ancak günümüzde birçok sunucu, e-posta, DNS, FTP ve hatta bir web

sunucusu durumunda birden fazla web sitesinin sorumluluğunu üstlenen paylaşılan sunuculardır[47].

Bu proje özelinde oluşturulan üretici, nakliyeci, depocu ve orderer organizasyonların her biri için bulut sunucular kullanılmıştır. Kullanılan bu sunuculara işletim sistemi olarak Ubuntu 20.04 LTS işletim sistemi kurulmuştur. İşletim sistemi kurulumundan sonra Hyperledger Fabric için gerekli olan altyapıların kurulumları yapılmıştır. Kurulumlar tamamlandıktan sonra sunucuların iletişimi için Docker Swarm ile sunucular birbirlerine bağlanmıştır. Kullanılan sunucuların özellikleri;4 çekirdekli CPU,8GB Ram ve 160GB depolama yeridir.

2.3. Kullanılan Proglamlama Araçları

Hyperledger başlangıçta Go dili kullanılarak oluşturuldu ve Hyperledger ekibi, dile aşina olan geliştiricilerin yeni dil öğrenmek yerine zincir kod geliştirme üzerinde çalışmaya başlayabilmeleri için olabildiğince çok dili desteklemeyi hedefler. Chaincode Go, Nodejs, Java'da yazılabilir. Diğer iki dil ile karşılaştırıldığında, Node.js, daha anlaşılır ve kullanımı daha kolay bir dildir.

2.3.1. Go

Go veya Golang, açık kaynaklı bir programlama dilidir. Statik olarak yazılmıştır ve derlenmiş makine kodu ikili dosyaları üretir. Geliştiriciler, sözdizimi söz konusu olduğunda Google'ın Go dilinin yirmi birinci yüzyılın C'si olduğunu söylüyor. Bununla birlikte, bu yeni programlama dili, belleği güvenle kullanmanıza, nesneleri yönetmenize, çöpleri toplamanıza ve eşzamanlılıkla birlikte statik (veya katı) yazım sağlamanıza olanak tanıyan araçları içerir[48].

Dünya Go ile ilk kez 2009'da Google'ın Rob Pike, Robert Griesemer ve Ken Thompson sayesinde tanıtıldı. Go'yu oluşturmanın temel amacı, diğer programlama dillerinin en iyi özelliklerini birleştirmekti.

Go bir proje için kullanıyorsa çok büyük bir teknoloji yığınına ihtiyacı olmaz. Go'da oluşturulan uygulamalar aslında yerel makine koduna derlenir ve herhangi bir yorumlayıcıya veya sanal makineye ihtiyaç duymaz. Bu aynı zamanda Go uygulamalarının daha hızlı çalışacağı anlamına gelir.

Hyperledger başlangıçta Go dili kullanılarak oluşturulmuştur. Hyperledger ekibi, dile aşina olan geliştiricilerin yeni dil öğrenmek yerine zincir kod geliştirme üzerinde çalışmaya başlayabilmeleri için olabildiğince çok dili desteklemeyi hedeflemektedir. Hyperledger Fabric, çeşitli programlama dillerinde akıllı sözleşmeler (zincir kodu) geliştirmeyi desteklemek için bir dizi API sunmaktadır.

Akıllı sözleşme API'leri Go, Node.js ve Java için mevcuttur. Temel özellikleri ve daha iyi performansı uygulamak için hangi dilin seçilmesi gerektiği her zaman büyük tartışmaları beraberinde getirmektedir.

Blok zincir ağı geliştirilirken dil seçimi bazı faktörlere bağlı olarak değişiklik göstermektedir. Hyperledger Fabric'in akıllı sözleşmelerin geliştirilmesinde Java, Javascript ve Go yazılım dillerini desteklemesi, Go programlama dilinin blok zincirinde öncü programlama dili olması göz önüne alındığında oluşturacağımız sistemde GO programlama dili kullanarak akıllı sözleşmelerin programlanması gerçekleştirilmiştir.

2.3.2. Javascript

JavaScript, hem istemci tarafında hem de sunucu tarafında kullanılan ve web sayfalarını etkileşimli hale getirilmesine olanak tanıyan metin tabanlı bir programlama dilidir. HTML ve CSS'nin web sayfalarına yapı ve stil kazandıran diller olduğu durumlarda JavaScript, web sayfalarına kullanıcının ilgisini çeken etkileşimli öğeler sağlar. JavaScript'i dahil etmek, web sayfasının kullanıcı deneyimini statik bir sayfadan etkileşimli bir sayfaya dönüştürerek iyileştirir. Özetlemek gerekirse, JavaScript web sayfalarına davranış ekler[49].

JavaScript, esas olarak web tabanlı uygulamalar ve web tarayıcıları için kullanılır. Ancak JavaScript, Web'in ötesinde yazılımlarda, sunucularda ve gömülü donanım kontrollerinde de kullanılır.

Web geliştirme dünyasını yöneten JavaScript, ismini blokzinciri programlama dilleri arasında da kendini göstermektedir. Bunun nedeni, dilin ve Angular, React ve Node gibi çok çeşitli kitaplık ve çerçevelerin, eş zamansız eylemleri işleme kolaylığı sağlamasıdır. JavaScript, blok zinciri geliştiricilerinin birden çok düğüm arasındaki iletişimi zahmetsizce ele almasına yardımcı olur ve bu da tasarlanan çözümlere

ölçeklenebilirliğin gücünü getirir. Bu koşullar göz önüne alınarak bu projede Javascript kullanılacaktır.

2.3.3. CouchDB

CouchDB, birçok kişinin NoSQL çözümü olarak adlandırdığı çözümlerden biridir. Özellikle, CouchDB belge odaklı bir veritabanıdır ve her belge alanı içinde anahtar-değer haritaları olarak depolanır. Alanlar basit bir anahtar / değer çifti, liste veya harita olabilir.

Apache CouchDB, Erlang'da uygulanan açık kaynaklı, belge odaklı bir NoSQL veritabanıdır.Bu veritabanı, verilerini depolamak, aktarmak ve işlemek için birden çok format ve protokol kullanır. Verileri depolamak için JSON, MapReduce kullanarak sorgu dili olarak JavaScript ve bir API için HTTP kullanır.

İlişkisel bir veritabanının aksine, bir CouchDB veritabanı verileri ve ilişkileri tablolarda saklamaz. Bunun yerine, her veritabanı bağımsız belgelerden oluşan bir koleksiyondur. Her belge kendi verilerini ve kendi kendine yeten şemasını korur. Bir uygulama, biri kullanıcının cep telefonunda, diğeri bir sunucuda depolanmış gibi birden çok veri tabanına erişebilir. Belge meta verileri revizyon bilgilerini içerir, bu da veritabanlarının bağlantısı kesilirken meydana gelmiş olabilecek farklılıkları birleştirmeyi mümkün kılar[50].

Hyperledger Fabric, iki tür eş durum veri tabanını desteklemektedir. LevelDB, eş düğüme gömülü varsayılan durum veri tabanıdır. LevelDB, zincir kodu verilerini basit anahtar-değer çiftleri olarak depolar ve yalnızca anahtar, anahtar aralığı ve bileşik anahtar sorgularını desteklemektedir.

CouchDB, genel muhasebedeki verileri JSON olarak modellemenize ve anahtarlar yerine veri değerlerine karşı zengin sorgular yayınlamanıza olanak tanıyan isteğe bağlı, alternatif bir durum veri tabanıdır. CouchDB desteği, sorguları daha verimli hale getirmek ve büyük veri kümelerini sorgulamanızı sağlamak için dizinleri zincir kodunuzla dağıtmanıza da olanak tanımaktadır.CouchDB'nin avantajlarından, yani içerik tabanlı JSON sorgularından yararlanabilmek için verilerimizin JSON formatında modellenmiş olması gerekmektedir. Bir eşin LevelDB'yi kullanırken

CouchDB'ye geçmesi, veri uyumluluğu sorunları nedeniyle desteklenmemektedir. Bu sebeple veri tabanı tercihi Hyperledger Fabric ağı oluşturulurken yapılmalıdır.

Proje özelinde, oluşturduğumuz veri yapımız JSON formatında olması, Hyperledger Fabric'in CouchDB'yi desteklemesi ve CouchDB kullanmanın daha avantajlı olması nedeni ile CouchDB NoSQL veri tabanı kullanılmıştır.

2.3.4. Docker

Docker, uygulamaları geliştirmek, göndermek ve çalıştırmak için açık bir platformdur. Docker, yazılımlarınızı hızlı bir şekilde teslim edebilmeniz için uygulamalarınızı altyapınızdan ayırmanıza olanak tanır. Docker ile altyapınızı, uygulamalarınızı yönettiğiniz şekilde yönetebilirsiniz. Docker'ın kodu gönderme, test etme ve hızlı bir şekilde dağıtma metodolojilerinden yararlanarak, kod yazma ile onu üretimde çalıştırma arasındaki gecikmeyi önemli ölçüde azaltılabilir[51].

Docker, bir uygulamayı konteyner adı verilen gevşek bir şekilde izole edilmiş bir ortamda paketleme ve çalıştırma yeteneği sağlar. İzolasyon ve güvenlik, belirli bir ana bilgisayarda aynı anda birçok konteyneri çalıştırmanıza izin verir. Konteynerlar uygulamayı çalıştırmak için gereken her şeyi içerir, bu nedenle ana bilgisayarda şu anda yüklü olana güvenmeniz gerekmez. Çalışırken konteynerları kolayca paylaşabilir ve paylaşılan herkesin aynı şekilde çalışan aynı konteynera sahip olduğundan emin olunabilir.

Docker, geliştiricilerin, uygulamaları ve hizmetleri sağlayan yerel kapsayıcıları kullanarak standartlaştırılmış ortamlarda çalışmasına izin vererek geliştirme yaşam döngüsünü kolaylaştırır. Kapsayıcılar, sürekli entegrasyon ve sürekli teslim iş akışları için mükemmeldir.

Docker'ın kapsayıcı tabanlı platformu, yüksek düzeyde taşınabilir iş yüklerine olanak tanır. Docker konteynerleri, bir geliştiricinin yerel dizüstü bilgisayarında, bir veri merkezindeki fiziksel veya sanal makinelerde, bulut sağlayıcılarında veya çeşitli ortamlarda çalışabilir.

Docker'ın taşınabilirliği ve hafif yapısı aynı zamanda iş yüklerini dinamik olarak yönetmeyi, iş ihtiyaçlarının gerektirdiği şekilde uygulamaları ve hizmetleri neredeyse gerçek zamanlı olarak ölçeklendirmeyi veya parçalamayı kolaylaştırır.

Docker Swarm, Docker uygulamalarında çalışan bir orkestrasyon yönetim aracıdır. Son kullanıcıların bir Docker düğümleri kümesi oluşturmasına ve dağıtmasına yardımcı olmaktadır. Tüm Docker arka plan programları Docker API'sini kullanarak etkileşime girmektedir. Swarm içindeki her bir konteyner aynı kümenin düğümleri tarafından konuşlandırılabilir ve erişilebilir.

Docker Swarm, blok zincir teknolojisinin temel ilkeleri ile benzerlik gösterdiği için ve bu proje özelinde blok zincir ağındaki düğümlerin Docker Swarm ile birbirlerine bağlanmıştır.

2.3.5. Postman

Postman, başkaları tarafından yapılan RESTful API'leri incelemeye veya kendi yaptıklarınızı test etmeye çalışırken harika bir araçtır. Sadece bir API'nin işlevselliğini test etmek için bir sürü kod yazma zahmetine girmeden HTML isteklerinde bulunabileceğiniz şık bir kullanıcı arayüzü sunar. Bunun yerine, koleksiyon adı verilen test paketleri oluşturularak Postman'nin API ile etkileşim kurmasına izin verilmektedir. Bu araçta, herhangi bir geliştiricinin ihtiyaç duyabileceği hemen hemen tüm işlevler gömülüdür. Bu araç GET, POST, PUT, PATCH gibi çeşitli HTTP istekleri yapma ve API'yi JavaScript ve Python gibi diller için koda dönüştürme yeteneğine sahiptir[52].

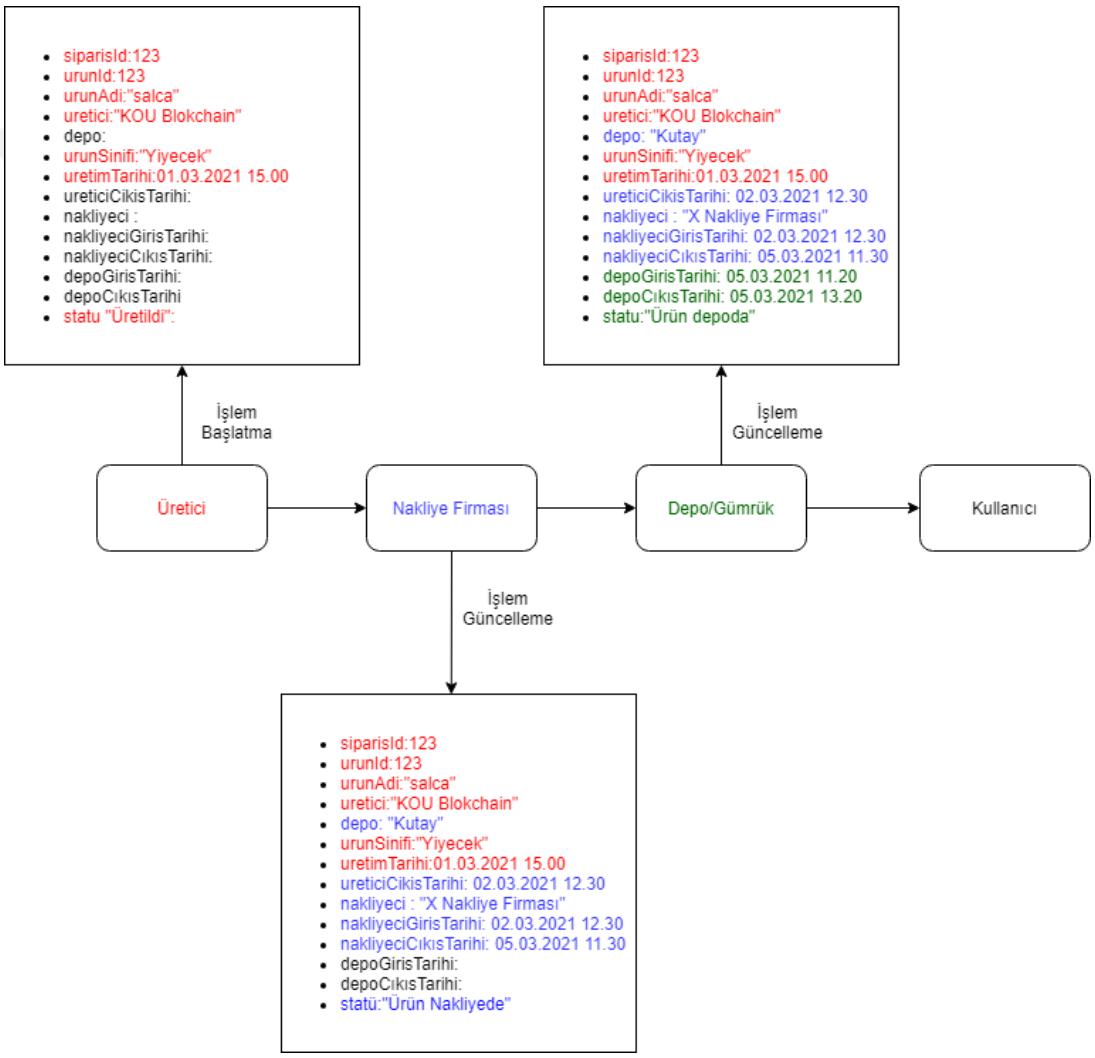
Hyperledger Fabric ağına bir işlem olarak GET ve POST edebilmek için Postman platformu tercih edilmiştir. Ayrıca API'leri test etme imkanı sunduğu için geliştirilen Hyperledger Fabric ağı Postman üzerinde test edilmiştir.

2.4. Uygulama

Bu proje kapsamında üretici, nakliyeci, depo ve satıcı olmak üzere dört tane organizasyon bulunmaktadır. Organizasyona kayıtlı kullanıcılar, ağ içerisinde işleme girecek olan ürünler üzerinde, kendilerine verilen yetkiler ile işlem gerçekleştirebilmektedir. Bu organizasyonların her biri farklı sunuculardan işlem yapmaktadır.

Docker Swarm sayesinde bu organizasyonlar birbiriyle haberleşir.Docker Network altında aynı ağda çalışırlar.Organizasyonlar tanımlı bir kullanıcı yaratarak süreçleri ise akıllı sözleşmelerle çalıştırırlar.

Üretici organizasyonunda ki kullanıcılar, ürünün üretildiği bilgisini sisteme bildirir.Bu bilgi sisteme iletildiğinde blok zincirde bu ürün için işlem başlatılacaktır.Bu işlemde ürüne ve siparişe ait bilgiler olacaktır.Şekil 2.1'de organizasyonun veri yapısı gösterilmiştir.



Şekil 2.1.Organizasyon veri yapıları

Nakliye organizasyonuna kayıtlı kullanıcılar, üreticiden aldığı ürünlerin nakliyede olduğu bilgisini nakliye araçlarına geçince sisteme bildirecektir. Sisteme iletilen bilgi ile üreticinin ürün özelinde başlatmış olduğu işlem güncellenecektir. Bu aşamada tedarik sürecinde birden fazla nakliye firması olabileceği için her nakliye firması için

tanımlayıcı bir numara sistem tarafından oluşturulacaktır. Güncelleme işleminde nakliyeci gerekli bilgileri güncelleyecektir.

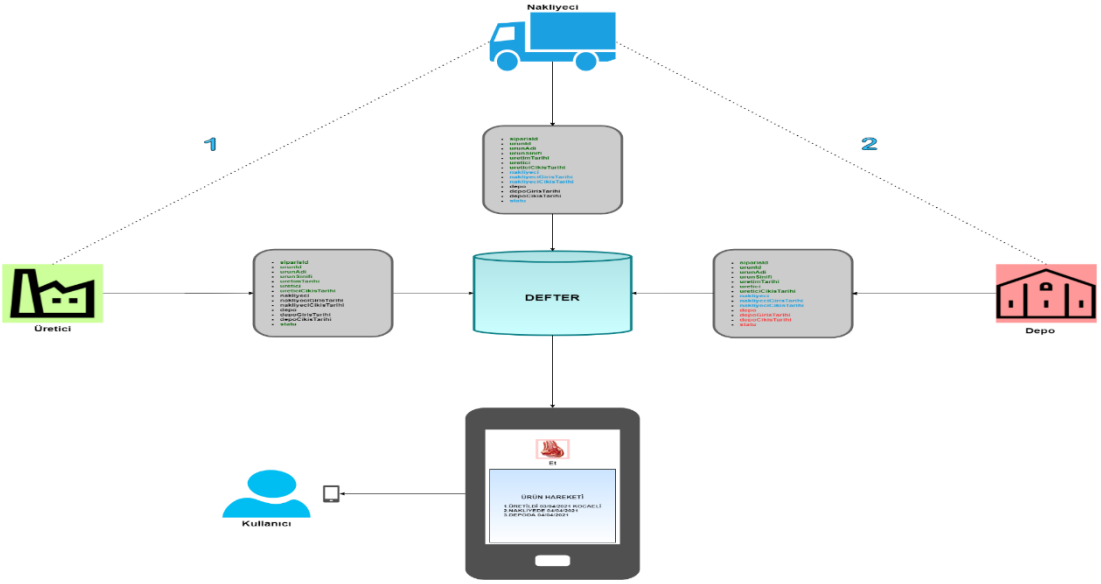
Depo organizasyonuna kayıtlı kullanıcılar, nakliyeci firmadan gelen ürünleri depo girişlerinde ürünün depoda olduğu bilgisini sisteme bildirecektir. Sisteme aktarılan bu bilgi ile ürün üzerinde devam eden işlem güncellenecektir. Güncelleme işleminde depo organizasyonuna kayıtlı kullanıcılar;

- Ürünün durum bilgisini,
- Depoya giriş tarihini,

otomatik bir şekilde güncelleyerek tedarik sürecini tamamlayacaktır. Tamamlanan bu tedarik süreci sayesinde proje kapsamında hedeflediğimiz, son kullanıcı için şeffaflık ve güven ilkelerinin sağlanması hedeflenmiştir.

2.4.1. Sistem mimarisi

Tedarik zinciri senaryosu kapsamında hazırlanan yapı Şekil 2.2.'de gösterilmiştir.Temel senaryo süreci üreticinin ürünün üretildiği bilgisini ağa yaymasıyla başlamaktadır.

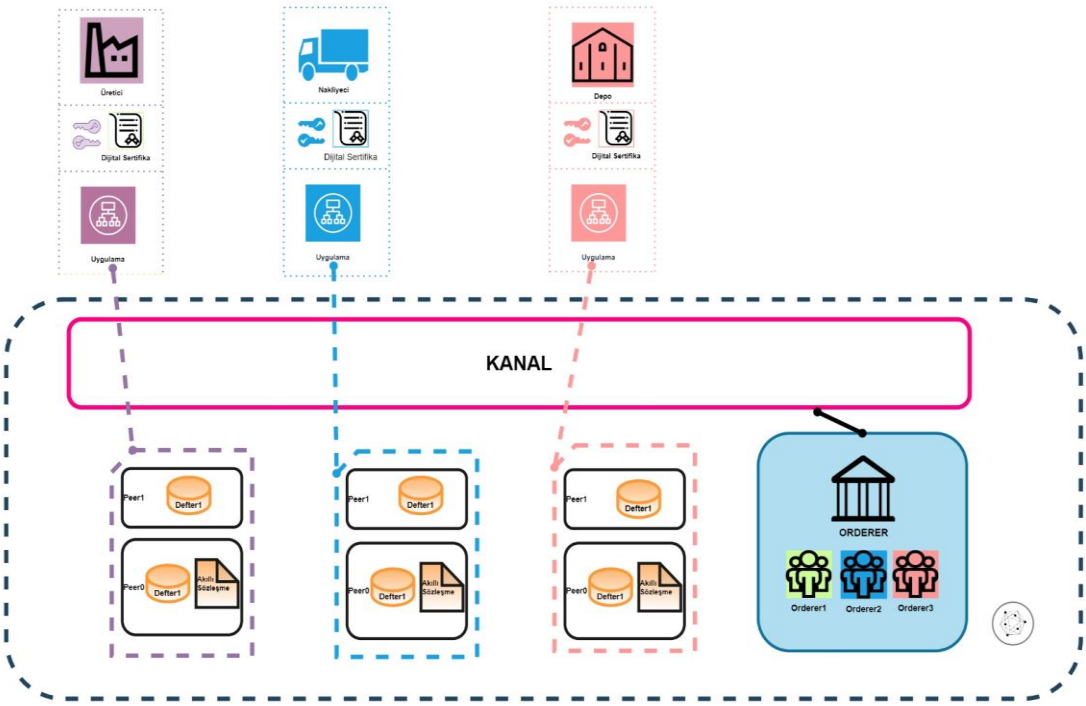


Şekil 2.2.Sistem Mimarisi

Nakliye firmasının ürünü bir depoya ulaştırması, ürünün depodan çıkarak tekrar nakliye firması aracılığıyla satıcıya ulaştırılması senaryo kapsamında hedeflenmektedir. Her bir noktada ürünün yolculuğu ağa kaydedilip güncellenmektedir. Nihai noktada bir kullanıcı ürünün üretiminden itibaren rafa kadar gelen süreçteki tüm yolculuğunu görüntüleyebilmektedir. Projenin ana hedeflerinden biri olan tedarik zincirinde şeffaflık ve güven unsurlarının tesisi bu şekilde sağlanmış olmaktadır.

2.4.2. Hyperledger ortamının kurulması

Hyperledger ağında 3 tane organizasyon oluşturulmuştur.Bunlar;üretici,nakliyeci ve depodur. Orderer organizasyonu oluşturularak akıllı sözleşmeler ve üyeliklerde ki kontroller sağlanmıştır.Orderer organizasyon bizans hata toleransı algoritması ile Hyperledger ortamında gerçekleştirilecek işlemlerin kontrolünü sağlanmıştır.Hyperledger Fabric'in kurulumu için script kullanılmıştır. Bu script ile Hyperledger Fabric imajları ve bir ağı kurmak için ihtiyacımız olan ikili dosyaları hem de bütün gereksinimleri sistemimize kurmuş oluyoruz.Gerekli olan tüm dosyalar yüklenmiş oluyor.Şekil 2.3'te Hyperledger Fabric ağı gösterilmiştir.



Şekil 2.3. Hyperledger Fabric Ağı

2.4.3. Docker swarm ağının kurulması

Docker Swarm ağında düğümler, manager ve worker düğümler olarak ikiye ayrılırlar. Manager düğümler, ağ üzerinde kontrole sahip olanlardır ve felaket durumunda yük dengeleme ve konteyner kurtarmadan sorumludurlar. Worker düğümleri, kapsayıcıları çalıştırmaya yardımcı olacak düğümlerdir.

Üç organizasyon da Docker Swarm ağına üç ayrı sunucu üzerinden katılacaklardır. Docker Swarm ağının oluşturulması ve bu ağa ”manager” ya da ”worker” olarak katılabilmek için ağa katılacak cihazda 2377, 7946, 4789 portlarının açık olması ve statik bir ip adresine sahip olması gerekmektedir.

Swarm ağı başlatılır. Ağı başlatan kişi manager rolündedir. Terminalde yazılan “docker swarm init –advertise-addr 192.168.0.20” komutuyla bize worker rolüyle ağa düğüm eklemek için bir token döndürmektedir. Bu token ile farklı bir düğüm ağımıza katılabimektedir. Ağa manager rolüyle düğüm eklemek için ise terminalde yeni bir komut çalıştırmamız gerekmektedir.

Dönen token kullanılarak çalıştırılan komut ile ağa eklenecek yeni bir düğümün manager rolünde olması sağlanmaktadır.”docker info” komutu ile Swarm ağının mevcut durumu görüntülenebilmektedir. Bu komut yalnızca manager rolündeki nodelar için bir çıktı döndürmektedir. Bu çıktıda ağda bulunan tüm nodeların; id, hostname, status, availability ve manager status gibi bilgilerinin bir çıktısı bulunmaktadır.

Komut ile worker rolündeki bir node, manager rolüne yükseltilebilmektedir. Bu yükseltme manager rolündeki bir node, çevrimdışına alındığında kullanılabilmektedir. ”docker swarm leave” komutu ile de bulunulan ağdan ayrılma işlemi gerçekleştirilmektedir.

2.4.4. Sertifika otoritelerinin oluşturulması

Sertifika Otoritesi, bir blok zinciri kullanıcılarına bir dizi sertifika hizmeti sağlar. Daha spesifik olarak, bu hizmetler, kullanıcı kaydı, blok zincirinde başlatılan işlemler ve kullanıcılar veya blok zincirinin bileşenleri arasındaki TLS korumalı bağlantılar ile ilgilidir.

Kayıt sertifikası otoritesi , yeni kullanıcıların blok zincir ağına kaydolmasına ve kayıtlı kullanıcıların bir kayıt sertifikası çifti talep etmesine olanak tanır. Bir sertifika veri imzalama, diğeri veri şifreleme içindir. Veri şifreleme anahtarı daha sonra kullanıcı tarafından ECIES (Eliptik Eğri Tümleşik Şifreleme Sistemi) tarzında kullanılmak üzere dönüştürülür.

Bir kullanıcı kaydolduğunda, işlem sertifikası otoritesinden işlem sertifikaları da talep edebilir. Bu sertifikalar, zincir kodu dağıtmak ve blok zincirde zincir kodu işlemlerini başlatmak için kullanılacaktır. Bu sertifika otoritesi Hyperledger Fabric kurulumunda gelen "hyperledger/fabric-ca" imajı aracılığıyla oluşturulur. Birden fazla işlem için tek bir işlem sertifikası kullanılabilse de, gizlilik nedeniyle her işlem için yeni bir işlem sertifikası kullanılması önerilir."fabric-ca-client enroll" komutuyla yönetici kaydedilir ve kripto materyaller alınır.

2.4.5. Kimlik sınıflandırma

```
NodeOUs:
  Enable: true
  # For each identity classification that you would like to utilize, specify
  # an OU identifier.
  # You can optionally configure that the OU identifier must be issued by a specific CA
  # or intermediate certificate from your organization. However, it is typical to NOT
  # configure a specific Certificate. By not configuring a specific Certificate, you will be
  # able to add other CA or intermediate certs later, without having to reissue all credentials.
  # For this reason, the sample below comments out the Certificate field.
  ClientOUIdentifier:
    # Certificate: "cacerts/cacert.pem"
    OrganizationalUnitIdentifier: "client"
  AdminOUIdentifier:
    # Certificate: "cacerts/cacert.pem"
    OrganizationalUnitIdentifier: "admin"
  PeerOUIdentifier:
    # Certificate: "cacerts/cacert.pem"
    OrganizationalUnitIdentifier: "peer"
  OrdererOUIdentifier:
    # Certificate: "cacerts/cacert.pem"
    OrganizationalUnitIdentifier: "orderer"
```

Şekil 2.4. Kimlik Sınıflandırma

Varsayılan MSP uygulaması, kuruluşların kimliklerine göre clients, admins, peers, and orderers olarak daha fazla sınıflandırmasına olanak tanır.

Şekil 2.4'de ki "NodeOUs: Enable: true" komutu ile kimlik sınıflandırma işlemi başlar."Certifacate" komutu ile sertifika otoritesi dosyasının konumu gösterilir.

- Bir kimlik, bir sipariş düğümüne aitse, "orderer" olarak sınıflandırılmalıdır.

- İşlemleri onaylayan veya taahhüt eden bir kimlik, “peer” olarak sınıflandırılmalıdır.
- Bir kimlik, bir kanala eş birleştirme veya bir kanal yapılandırma güncelleme işlemini imzalama gibi yönetim görevlerini yerine getiriyorsa, “admin” olarak sınıflandırılmalıdır.
- Bir kimlik, ağ üzerinde işlem yapıyorsa, “client” olarak sınıflandırılmalıdır.

Dört sınıflandırma birbiriyle ilişkili değildir. Kimlikler client veya peer olarak sınıflandırılmadan önce bazı kanal özelliklerinin etkinleştirilmesi gerekir. Kimliklerin admin veya orderer veren olarak sınıflandırılması için bir üst seviye kanal özelliğinin etkinleştirilmesi gerekir.

2.4.6. Kanal yapılarının oluşturulması

Hyperledger Fabric blok zinciri ağı için paylaşılan yapılandırma, kanal başına bir tane olmak üzere bir koleksiyon yapılandırma işleminde saklanır. Her yapılandırma işlemi genellikle daha kısa configtx adıyla anılır.

Hyperledger Fabric kanalı, özel ve gizli işlemlerin gerçekleştirilmesi amacıyla iki veya daha fazla belirli ağ üyesi arasındaki özel bir iletişim "alt ağı" dır. Bir kanal, üyeler (kuruluşlar), üye başına bağlantı eşleri, paylaşılan defter, zincir kodu uygulamaları ve sipariş hizmeti düğümleri tarafından tanımlanır. Ağdaki her işlem, her bir tarafın kimliğinin doğrulanması ve o kanalda işlem yapmak için yetkilendirilmesi gereken bir kanalda yürütülür. Bir kanala katılan her eşin, her eşin kendi kanal eşlerine ve hizmetlerine kimliğini doğrulayan bir üyelik hizmetleri sağlayıcısı (MSP) tarafından verilen kendi kimliği vardır.

Hyperledger Fabric üzerinde bir kanal oluşturabilmek için öncelikle kanalın konfigürasyonlarını belirten bir yapılandırma dosyası oluşturuldu. Kanal yapılandırması genellikle defterde tutulmaktadır ve sonradan eklenen bütün blokları yönetmektedir. Kanal yapılandırması, hangi organizasyonların kanal üyesi olduğunu, kanala yeni bloklar ekleyebilen düğümleri ve kanal güncellemelerini belirleyen politikaları belirtmektedir. Kanal konfigürasyonu sonradan güncellenebilmektedir. Yeterli sayıda organizasyon güncellemeyi onaylarsa, yeni bir kanal yapılandırma bloğu oluşarak kanalı yönetecektir.

Kanal yapılandırmasında yer alan en önemli bilgiler kanal üyesi olan organizasyonlardır. Her organizasyon bir üyelik hizmet sağlayıcısı ve üyelik hizmet sağlayıcısı kanalı ile oluşturulmaktadır. Üyelik hizmet sağlayıcısı kanal yapılandırmasında depolanmaktadır. Bir organizasyonun düğümlerini, uygulamalarını ve yöneticilerini tanımlamak için kullanılan sertifikaları içermektedir. Oluşturduğumuz kanal konfigürasyon dosyasının organizasyonlar bölümü, kanalın her üyesi için bir üyelik hizmet sağlayıcısı kanalını ve üyelik hizmet sağlayıcısı kimliği oluşturmak için kullanmaktadır.

```
Organizations:

    - &OrdererOrg
        Name: OrdererOrg
        ID: OrdererMSP
        MSPDir: ../../setup1/vm4/crypto-config/ordererOrganizations/example.com/msp

        Policies:
            Readers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Writers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Admins:
                Type: Signature
                Rule: "OR('OrdererMSP.admin')"

    - &Org1
        Name: Org1MSP
        ID: Org1MSP

        MSPDir: ../../setup1/vm1/crypto-config/peerOrganizations/org1.example.com/msp

        Policies:
            Readers:
                Type: Signature
                Rule: "OR('Org1MSP.admin', 'Org1MSP.peer', 'Org1MSP.client')"
            Writers:
                Type: Signature
                Rule: "OR('Org1MSP.admin', 'Org1MSP.client')"
            Admins:
                Type: Signature
                Rule: "OR('Org1MSP.admin')"
            Endorsement:
                Type: Signature
                Rule: "OR('Org1MSP.peer')"

        AnchorPeers:
            - Host: peer0.org1.example.com
              Port: 7051
```

Şekil 2.5. Organizasyon Ayarları

- ”Name” ile belirtilen alan organizasyonu tanımlamak için kullanılan resmi olmayan bir addır.
- ”ID” ile belirtilen alan organizasyonun üyelik hizmet sağlayıcısı kimliğidir.
- ”AnchorPeers” ile belirtilen alan kanala üye olan organizasyonun, kanal üzerinden diğer organizasyonlarla iletişime geçecek eşi tanımlamak için kullanılmaktadır.

- ”Policies” ile belirtilen bölüm, kanala üye olan organizasyonun imza ilkelerini tanımlamak için kullanılmaktadır.
- ”MSPDir” ile belirtilen alan organizasyon tarafından oluşturulan bir üyelik hizmet sağlayıcısı klasörünün yoludur. Üyelik hizmet sağlayıcısı kanalını oluşturmak için bu klasör kullanılmaktadır. Üyelik hizmet sağlayıcısı oluşturmak için kullanılan klasör yalnızca ortak sertifikalar içermektedir.

Varsayılan olarak, her kanal üyesi, kuruluşlarına referans veren bir dizi imza ilkesi tanımlar. Bir eşe bir teklif sunulduğunda veya sipariş düğümlerine bir işlem sunulduğunda, düğümler işleme eklenen imzaları okur ve bunları kanal konfigürasyonunda tanımlanan imza politikalarına göre değerlendirir. Her imza politikasının, imzaları politikayı karşılayabilecek kuruluşlar ve kimlikler kümesini belirten bir kuralı vardır. Aşağıdaki configtx.yaml'nin Şekil-10’da Org1 tarafından tanımlanan imza politikalarını görebilirsiniz.

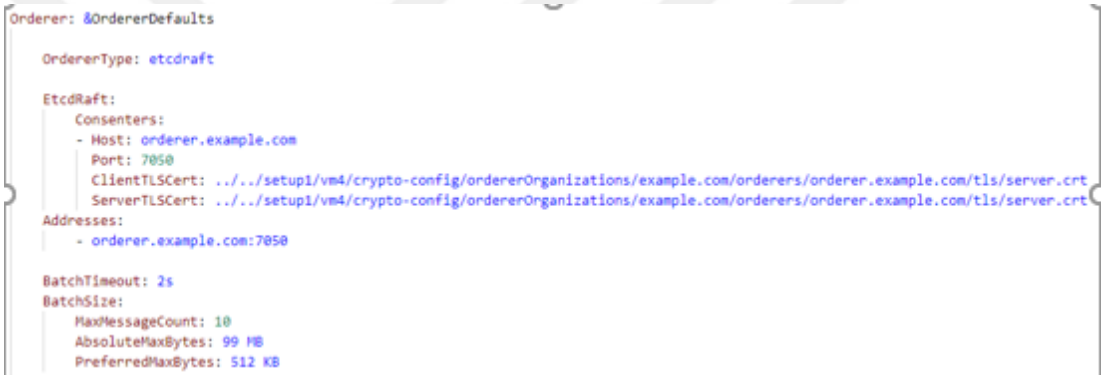
Şekil 2.5.’de ki tüm politikalar Org1'in imzalarıyla karşılanabilir. Bununla birlikte, her politika, organizasyon içinde politikayı karşılayabilecek farklı bir rol dizisini listeler. Yöneticiler politikası yalnızca “Admins” rolüne sahip bir kimlik tarafından gönderilen işlemlerle karşılanabilirken, yalnızca “peer” rolüne sahip kimlikler Onay politikasını karşılayabilir. Tek bir işleme eklenen bir dizi imza, birden çok imza ilkesini karşılayabilir. Örneğin, bir işleme eklenen onaylar hem Org1 hem de Org2 tarafından sağlandıysa, bu imza seti Org1 ve Org2'nin onay politikasını karşılayacaktır.

```
Policies:
    Readers:
        Type: ImplicitMeta
        Rule: "ANY Readers"
    Writers:
        Type: ImplicitMeta
        Rule: "ANY Writers"
    Admins:
        Type: ImplicitMeta
        Rule: "MAJORITY Admins"
    LifecycleEndorsement:
        Type: ImplicitMeta
        Rule: "MAJORITY Endorsement"
    Endorsement:
        Type: ImplicitMeta
        Rule: "MAJORITY Endorsement"
```

Şekil 2.6. ImplicitMeta Politikaları

Kanalınız varsayılan politikaları kullanıyorsa, her kuruluş için imza politikaları, kanal yapılandırmasının daha yüksek seviyelerinde ImplicitMeta politikaları tarafından değerlendirilir. Kanala gönderilen imzaları doğrudan değerlendirmek yerine, ImplicitMeta ilkeleri, kanal yapılandırmasında ilkeyi karşılayabilecek bir dizi başka ilke belirleyen kurallara sahiptir. Bir işlem, ilkenin referans aldığı temel imza ilkeleri kümesini karşılayabiliyorsa, bir ImplicitMeta ilkesini karşılayabilir.Şekil 2.6'da ImplicitMeta politikaları gösterilmiştir.

Tip olarak kullanılan "ImplicitMeta" politikasının önemli bir avantajı, kanala yeni bir yönetici organizasyonu eklendiğinde kanal politikasını güncellenmesini gerektirmemesidir.Bu da sisteme esneklik getirmektedir.



Şekil 2.7. Orderer Ayarları

Orderer , Raft konsensusu gibi orderer düğümler tarafından kullanılanları yönetmektedir ve kanal onaylayıcı kümesine ait olan düğümleri sıralayarak zincir kodu üzerinde çalıştırılabilir en düşük Hyperledger Fabric sürümünü ayarlamaktadır.

Configtx.yaml'nin Orderer bölümündeki ImplicitMeta politikaları, bir kanalın "orderer" düğümlerini "application" bölümünün eş kuruluşlarını yönetmesine benzer şekilde yönetir. ImplicitMeta ilkeleri, hizmet yöneticileri sipariş eden kuruluşlarla ilişkili imza ilkelerine işaret eder.

Her kanal yapılandırması, kanal onaylayıcısı kümesindeki "orderer" düğümleri içermektedir. Onaylayan kümesi, yeni blokların sisteme dahil edilmesi ve yeni blokları kanala katılan eşlere dağıtma yeteneğine sahip orderer düğümleri grubudur. Onaylayan kümesinin üyesi olan her orderer düğümünün uç nokta bilgileri kanal yapılandırmasında depolanmaktadır. "OrdererType" ile belirtilen alan konsensus türü

seçmek için kullanılmaktadır. Proje özelinde tercih edilen Raft konsensus algoritması bu alanda tanımlanmaktadır. Raft orderer hizmetleri, konsensus sürecine katılabilecek onay verenler listesi tarafından tanımlanmaktadır. Onaylayanlar listesindeki her orderer düğümü, bitiş noktasındaki adresleri, istemci ve sunucu TLS sertifikaları ile tanımlanmaktadır.Şekil 2.7'de ordererın ayarları gösterilmiştir.

- Batch size: Bu parametreler, bir bloktaki işlemlerin sayısını ve boyutunu belirler. Absolute_max_bytes büyüklüğünden daha büyük veya blok içinde max_message_count'tan fazla işlem bulunan hiçbir blok görünmeyecektir. Preferred_max_bytes altında bir blok oluşturmak mümkünse, o zaman bir blok erken kesilecek ve bu boyuttan daha büyük işlemler kendi bloklarında görünecektir.
- Batch timeout: Bir bloğu kesmeden önce ek işlemler için ilk işlem geldikten sonra beklenecek süre. Bu değeri düşürmek gecikmeyi iyileştirir, ancak çok fazla düşürmek bloğun maksimum kapasitesine kadar dolmasına izin vermeyerek verimi azaltabilir.

```
OrdererGenesis:
    <<: *ChannelDefaults
    Capabilities:
        <<: *ChannelCapabilities
    Orderer:
        <<: *OrdererDefaults
        OrdererType: etcdraft
        EtcdRaft:
            Consenters:
            - Host: orderer.example.com
              Port: 7050
              ClientTLSCert: ../../setup1/vm4/crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt
              ServerTLSCert: ../../setup1/vm4/crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt
            - Host: orderer2.example.com
              Port: 8050
              ClientTLSCert: ../../setup1/vm4/crypto-config/ordererOrganizations/example.com/orderers/orderer2.example.com/tls/server.crt
              ServerTLSCert: ../../setup1/vm4/crypto-config/ordererOrganizations/example.com/orderers/orderer2.example.com/tls/server.crt
            - Host: orderer3.example.com
              Port: 9050
              ClientTLSCert: ../../setup1/vm4/crypto-config/ordererOrganizations/example.com/orderers/orderer3.example.com/tls/server.crt
              ServerTLSCert: ../../setup1/vm4/crypto-config/ordererOrganizations/example.com/orderers/orderer3.example.com/tls/server.crt
        Addresses:
            - orderer.example.com:7050
            - orderer2.example.com:8050
            - orderer3.example.com:9050

        Organizations:
        - *OrdererOrg
        Capabilities:
            <<: *OrdererCapabilities
    Consortiums:
        SampleConsortium:
            Organizations:
            - *Org1
            - *Org2
            - *Org3
```

Şekil 2.8. Genesis Blok Ayarları

Orderer konfigürasyonu içerisinde onaylayan kümesini yöneten ilkeler tanımlanmaktadır. Oluşturulan blok zincir ağı, orderer yöneticilerinin çoğunluğunun

orderer düğümlerini, organizasyonların veya blok kesme parametrelerine bir güncelleştirme eklenmesini, kaldırılmasını veya onaylamasını gerektiren Hyperledger Fabric tarafından sağlanan varsayılan ilkeleri kullanmaktadır.

Hyperledger Fabric, kanal yapılandırması için Profiles bölümündeki kanal profillerini okumaktadır. Her profil diğer bölümlerden veri toplayabilmek için ”YAML” söz dizimini kullanmaktadır. Hyperledger Fabric bir uygulama kanalı oluşturmak veya genesis bloğu yazmak için bu yapılandırmayı kullanmaktadır.

Sistem kanalı, genesis bloğu oluşturmak için Profiles alanında bulunan OrdererGenesis bölümünü kullanmaktadır.Şekil 2.8’de genesis blok ayarları verilmiştir. Sistem kanalı orderer düğümlerini ve hizmet yöneticilerini sıralayan organizasyon kümesini tanımlamaktadır. Sistem kanalı, blok zincir konsorsiyumuna ait bir dizi eş organizasyon içermektedir. Konsorsiyumun her üyesinin kanal üyelik servis sağlayıcısı sistem kanalına dahil edilerek yeni uygulama kanalları oluşturmalarına ve kanala yeni konsorsiyum üyeleri eklemelerine olanak sağlamaktadır.

2.4.7. Kanala katılma işlemleri

Kanal oluşturulduktan sonra “peer” ile kanala katılabiliriz. Kanalın üyesi olan kuruluşlar, eş kanal getirme komutunu kullanarak kanal genesis bloğunu sipariş hizmetinden alabilir.

```
createChannel(){
    rm -rf ./channel-artifacts/*
    setGlobalsForPeer0Org1

    # Replace localhost with your orderer's vm IP address
    peer channel create -o 142.93.110.47:7050 -c $CHANNEL_NAME \
    --ordererTLSHostnameOverride orderer.example.com \
    -f ./../../artifacts/channel/${CHANNEL_NAME}.tx --outputBlock ./channel-artifacts/${CHANNEL_NAME}.block \
    --tls $CORE_PEER_TLS_ENABLED --cafile $ORDERER_CA
}

# createChannel

joinChannel(){
    setGlobalsForPeer0Org1
    peer channel join -b ./channel-artifacts/$CHANNEL_NAME.block

    setGlobalsForPeer1Org1
    peer channel join -b ./channel-artifacts/$CHANNEL_NAME.block

}

# joinChannel

updateAnchorPeers(){
    setGlobalsForPeer0Org1
    # Replace localhost with your orderer's vm IP address
    peer channel update -o 142.93.110.47:7050 --ordererTLSHostnameOverride orderer.example.com -c $CHANNEL_NAME -f ./../../artifacts/channel/${CORE_PEER_LOCALMSPID}anchors.tx --tls $CORE_PEER_TLS

}

# updateAnchorPeers

# removeOldCrypto

createChannel
joinChannel
updateAnchorPeers
```

Şekil 2.9. Kanal Oluşturma

Kuruluş daha sonra -peer channel join komutunu kullanarak eşi kanala birleştirmek için genesis bloğunu kullanabilir. Peer kanala katıldığında, eş kanaldaki diğer blokları sipariş hizmetinden alarak blok zinciri defterini oluşturacaktır.

Bir organizasyon, peerları kanala dahil ettikten sonra, bir anchor peer olmak için peerlardan en az birini seçmelidir. Özel veriler ve hizmet keşfi gibi özelliklerden yararlanmak için anchor peer gereklidir. Her kuruluş, yedeklilik için bir kanalda birden çok bağlantı peerı ayarlamalıdır.

Her kuruluşun bağlantı eşlerinin uç nokta bilgileri, kanal yapılandırmasına dahil edilir. Her kanal üyesi, kanalı güncelleyerek anchor peerları belirleyebilir. Kanal konfigürasyonunu güncellemek ve Organizasyon 1 ve Organizasyon 2 için bir anchor peer seçmek için configtxlator aracını kullanılır. Bir anchor peer ayarlama işlemi, diğer kanal güncellemelerini yapmak için gerekli olan adımlara benzer ve bir kanal yapılandırmasını güncellemek için yapılandırıcının nasıl kullanılacağına bir giriş sağlar. Ayrıca jq tool yerel makineye yüklenmesi gerekir.Şekil 2.9'da kanalı nasıl oluşturduğumuz gösterilmiştir.

Kanal yaratılmış durumda ise bunun için "createchannel" komutu kullanılmaz.Bunun yerine "fetchchannelblock" komutu kullanılır.

2.4.8. Akıllı sözleşmenin yapılandırmaları

Son kullanıcılar, akıllı sözleşmelere başvurarak blok zinciri defteriyle etkileşime girer. Hyperledger Fabric'te akıllı sözleşmeler, zincir kodu olarak adlandırılan paketlerde devreye alınır. İşlemleri doğrulamak veya defteri sorgulamak isteyen kuruluşların, meslektaşlarına bir zincir kodu yüklemeleri gerekir. Bir kanala katılan peerlara bir zincir kodu yüklendikten sonra, kanal üyeleri zincir kodunu kanala yerleştirebilir ve kanal defterinde varlıklar oluşturmak veya güncellemek için zincir kodundaki akıllı sözleşmeleri kullanabilir.

Zincir kodu, Kumaş zincir kodu yaşam döngüsü olarak bilinen bir işlem kullanılarak bir kanala dağıtılır. Kumaş zincir kodu yaşam döngüsü, birden fazla kuruluşun, işlemler oluşturmak için kullanılmadan önce bir zincir kodunun nasıl çalıştırılacağını kararlaştırmasına olanak tanır. Örneğin, bir onay politikası hangi kuruluşların bir işlemi doğrulamak için bir zincir kodu yürütmesi gerektiğini belirtirken, kanal

üyelerinin zincir kodu onay politikası üzerinde anlaşmak için Yapı zincir kodu yaşam döngüsünü kullanması gerekir.

```
type Product struct {
    ProductName  string `json:"productname"`
    ProductClass  string `json:"productclass"`
    Producer string `json:"producer"`
    ProductionDate  string `json:"productiondate"`
    ProducerCheckoutDate  string `json:"producercheckoutdate"`
    Transporter  string `json:"transporter"`
    TransporterEntryDate  string `json:"transporterentrydate"`
    TransporterCheckoutDate  string `json:"transportercheckoutdate"`
    Warehouse string `json:"warehouse"`
    WarehouseEntryDate string `json:"warehouseentrydate"`
    WarehouseCheckoutDate string `json:"warehousecheckoutdate"`
    Status  string `json:"status"`
```

Şekil 2.10. Ürün Bilgileri Değişkenleri

Projede zincir kodunun içeriği, Go içe aktarma ifadeleri ile başlamaktadır. Daha sonra ürünü tanımlamak, ürünün süreç için gereken değişkenlerini içererek başlar.

Projede ürün bilgisi ve durumu gibi özel detayların tutulduğu değişkenler bulunmaktadır. ”ProductId” değişkeni ürünün id bilgisini, ”Status” değişkeni ise ürünün durum bilgisini tutmaktadır.(Şekil 2.11.) Daha sonra zincir kodunun başlatma işlevi olarak kullanılacak olan ”Init” fonksiyonu eklenmiştir. Varsayılan olarak bu işlev asla çalıştırılmaz. ”Init” fonksiyonu sadece zincir kodunu başlatmak için kullanılmaktadır. Hyperledger Fabric, ”Init” fonksiyonunun diğer işlemlerden önce çağırılmasını ve yalnızca bir kez çağrılmasını sağlamaktadır. Bu fonksiyon, hangi kullanıcıların zincir kodunu başlatabileceği konusunda ek kontrol ve deftere ilk verileri ekleme yeteneği sağlamaktadır.

```
type ProductPrivateDetails struct {
    ProductId   string `json:"productid"`
    Status  string `json:"status"`
}
```

Şekil 2.11. Ürünün Özel Bilgileri

```go
func (s *SmartContract) Invoke(APIstub shim.ChaincodeStubInterface) sc.Response {

    function, args := APIstub.GetFunctionAndParameters()
    logger.Infof("Function name is:  %d", function)
    logger.Infof("Args length is : %d", len(args))

    switch function {
    case "queryProduct":
        return s.queryProduct(APIstub, args)
    case "createProduct":
        return s.createProduct(APIstub, args)
    case "queryAllProduct":
        return s.queryAllProduct(APIstub)
    case "changeProductStatus":
        return s.changeProductStatus(APIstub, args)
    case "getHistoryForAsset":
        return s.getHistoryForAsset(APIstub, args)
    case "queryProductByStatus":
        return s.queryProductByStatus(APIstub, args)
    default:
        return shim.Error("Invalid Smart Contract function name.")
```

Şekil 2.12. Çağırma Fonksiyonu

”Invoke” fonksiyonu, istemciler tarafından sorgu fonksiyonlarının çağrılması için kullanılmaktadır. Bu fonksiyon zincir kodunu, kanal defterindeki verileri okuma ve yazma için kullanılmasına olanak sağlamaktadır. Projede ürün yaratmak, ürün bilgilerini sorgulamak, defterdeki kayıtları sorgulamak gibi fonksiyonları çağırmak için kullanılmıştır.

- ”queryProduct” ile belirtilen fonksiyon, ürünün ile alakalı tüm bilgileri listeler.
- ”createProduct” fonksiyonu, bir ürün yaratarak yukarıda tanımlanan değişkenlerin içlerini doldurmak için kullanılır.
- ”queryAllProduct” ile belirtilen fonksiyon, oluşturulan tüm ürünleri getirmek için kullanılır.
- ”changeProductStatus” fonksiyonu, ürünün üreticiden tüketiciye kadar bulunduğu sürecin durum bilgisini güncellemek için kullanılır.
- ”getHistoryForAsset” fonksiyonu, sorgulanacak olan ürünün en güncel haline kadarki işlem geçmişini göstermek için kullanılır.
- ”queryProductByStatus” fonksiyonu, ürünün durum bilgisini sorgulamak için kullanılır.

2.4.9. Akıllı sözleşmelerin organizasyonlara yüklenmesi

Varlık transferi akıllı sözleşmeyi paketledikten sonra, zincir kodunu peerlara yükleyebiliriz. Zincir kodunun, bir işlemi onaylayacak her peera yüklenmesi gerekir. Onay politikasını hem Organizasyon 1 hem de Organizasyon 2'den onay gerektirecek şekilde ayarlayacağımız için, zincir kodunu her iki kuruluş tarafından işletilen peerlara yüklenir.

Zincir kodu, Hyperledger Fabric zincir kodu yaşam döngüsü olarak bilinen bir işlem kullanılarak proje özelinde oluşturulan kanala dağıtılmıştır. Hyperledger Fabric zincir kodu yaşam döngüsü, birden çok kuruluşun bir zincir kodunun işlem oluşturmak için kullanılmadan önce nasıl çalıştırılacağına olanak tanımaktadır.

Bir kuruluş zincir kodunu emsallerine yüklediyse, kuruluş tarafından onaylanan zincir kodu tanımına paket kimliğini eklemeleri gerekir. Paket kimliği, bir eşte kurulu zincir kodunu onaylanmış bir zincir kodu tanımıyla ilişkilendirmek için kullanılır ve bir kuruluşun işlemleri onaylamak için zincir kodunu kullanmasına izin verir. Eşinizi sorgulamak için eş yaşam döngüsü zincir kodu “queryinstalled” komutunu kullanarak bir zincir kodunun paket kimliği bulunur.

Zincir kodunun eşlere yüklenebilmesi için önce bir ”tar” dosyasında paketlenmesi gerekmektedir. Öncelikle ”packageChaincode()” fonksiyonu ile oluşturulan akıllı sözleşme ”tar.gz” uzantılı olarak sıkıştırılır. Zincir kodu paketi, hareketleri yürütecek ve onaylayacak her eşe yüklenmesi gerekmektedir. Zincir kodu eşe yüklendikten sonra zincir kodu oluşturulmakta ve eğer zincir kodu ile ilgili bir sorun ortaya çıkarsa bir yapı hatası döndürülmektedir. Sıkıştırma işlemi tamamlandıktan sonra oluşturulan ”tar.gz” uzantılı dosya ”installChaincode()” fonksiyonu ile organizasyona yüklenmektedir.

Zincir kodu bir zincir kodu tanımlayıcısı tarafından yönetilmektedir. Kanal üyeleri bir zincir kodu tanımını onayladığında, bu onay kabul ettiği zincir kodu parametrelerinde bir organizasyon tarafından oy olarak işlev görmektedir. Bu onaylanmış organizasyon tanımları, kanal üyelerinin bir kanalda kullanılmadan önce zincir kodu üzerinde onaylannmasını sağlamaktadır.”queryInstalled()” ve ”approveForMyOrg1()” fonksiyonları kullanılarak belirtilen onay işlemleri gerçekleştirilir. Yeterli sayıda kanal

üyesi bir zincir kodu tanımını onayladıktan sonra, kanal üzerinde bir organizasyon tanımlanabilmektedir. Tanımlama işlemi ilk olarak, organizasyonlar için onaylanan zincir kodunu sorgulayan ve organizasyonları tarafından onaylanmış kanal üyelerinin eşlerine gönderilmektedir. Bu işlem daha sonra orderer hizmetine gönderilmektedir. İşlem orderer hizmetine ulaştığında zincir kodu tanımı tamamlanmaktadır.”checkCommitReadyness()” ve ”commitChaincodeDefination()” fonksiyonları kullanılarak bu işlemler gerçekleştirilir.

2.4.10. API server

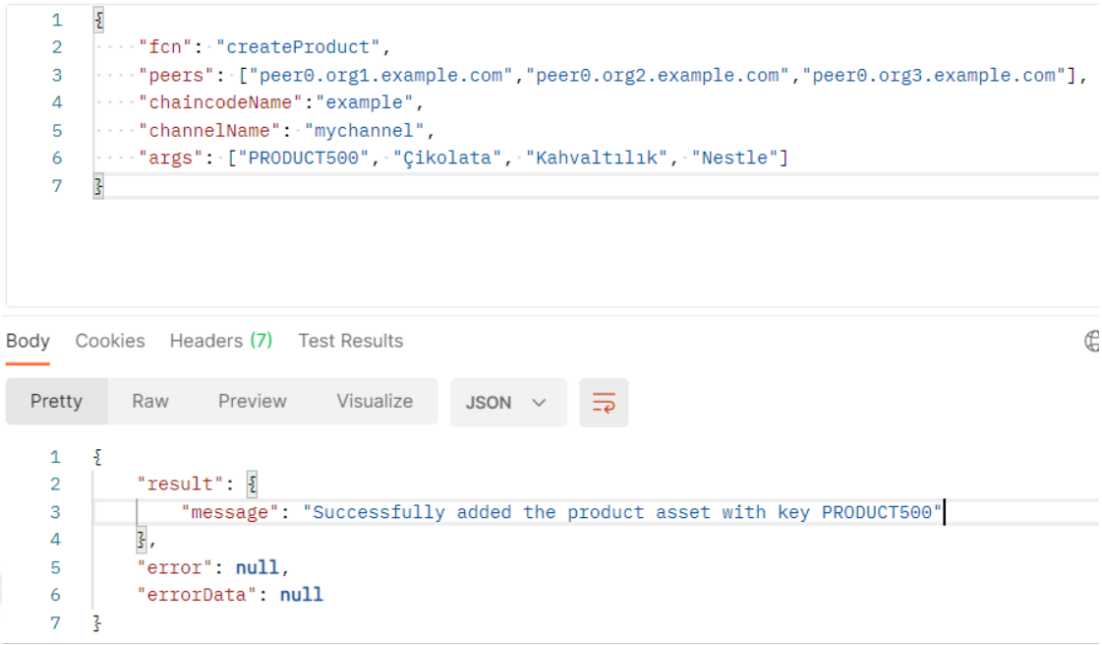
API server ile uygulamanın gerçekleşmesi için gereken veriler gönderilir.Bu postman uygulaması ile yapılır.Bunun sayesesinde kurduğumuz Hyperledger Fabric yapısıyla haberleşilir.

“App.js” dosyası ile Hyperledger Fabric ağına ulaşılır.Burada GET ve POST gibi metodlar ile serverın diğer dosyalarına erişilir.Gönderdiğimiz metodlar ile başarılı bir şekilde ağa yeni kayıtlar atabilir ve güncellenir. “invoke.js” kodu ise “POST” etiketli metotların akıllı sözleşmelerle etkileşim kurup, işlem isteğinin gönderildiği yerdir.

“App.js” tarafında tetiklemiş olduğumuz “invokeTransaction” metodu ile Hyperledger Fabric ağına işlem isteği gönderilir. Postman ara yüzünden gelen bilgiler ile burada kullanıcının organizasyon bilgisi, kimlik bilgileri gibi bilgiler doğrulanarak ağ üzerinde bir bağlantı açılır ve akıllı sözleşmelere erişilir. İşlem tamamlandıktan sonra “gateway.disconnect()” metodu ile bağlantı sonlandırılır.

2.5. Uygulamanın Gerçekleştirilmesi

Bütün kurulumlar tamamlandıktan sonra tedarik zinciri üzerinde ilerleyebilmek için öncelikle bir ürün üretilmesi gerekmektedir. Ürünün üretilebilmesi için öncelikle üretici organizasyonuna kayıtlı bir kullanıcı oluşturulmaktadır. Diğer organizasyonlarda da kullanıcı kaydetmek için aynı işlem gerçekleştirilmektedir. Bu işlemi gerçekleştirebilmek için Postman üzerinden akıllı sözleşmenin şartlarına uygun bir ”Json” dosyası ilgili organizasyona iletilmektedir. İşlem başarılı bir şekilde gerçekleştirildiği takdirde kayıt olan kullanıcı için sistemde bir token oluşturulmaktadır.



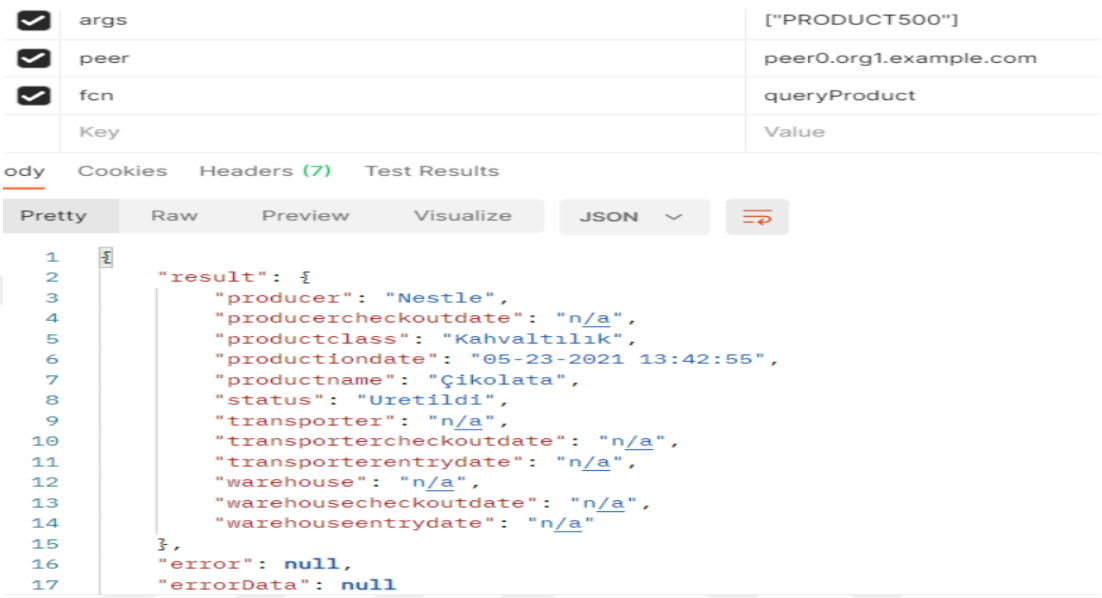
Şekil 2.13. Ürünün Üretilmesi

”fcn” ile belirtilen alanda ürünün üretildiği bilgisini sisteme iletmek için çalışacak fonksiyon belirtilmiştir. ”peers” ile belirtilen bölümde organizasyonların kanal ile haberleşmek için hangi eşler olduğu belirtilmektedir.

Şekil 2.13.’te görülen fonksiyon “createProduct” ile ürün yaratılır.”args” ile gönderilen JSON’da ürünün bilgileri gönderilir.Üretici organizasyonu bunla beraber ürünü yaratır.Daha sonra bu ürünün depoya gönderilmesi gerekir.Burada nakliyeci devreye girer.Bu yüzden ürünün durumunun değiştirilmesi gerekir.

Şekil 2.14. Ürün Durumunun Değiştirilmesi

Şekil 2.14'de görüldüğü gibi "changeProductStatus" fonksiyonu ile ürünün durumu değiştirilir.Ürünün hangi duruma geçtiğini "args" değişkeninde belirlenir.Ürünün nakliyede olduğu bu değişkene ürünün ismi yollanarak belirlenir.Daha sonra depoya geçmesi ise bu değişkene deponun ismi yollanarak geçirtilir.Depo aşamasından sonra ise yine nakliyeci organizasyonu ile son durağa gider.Burada ürün teslim olur.



Şekil 2.15. Ürün Sorgulama

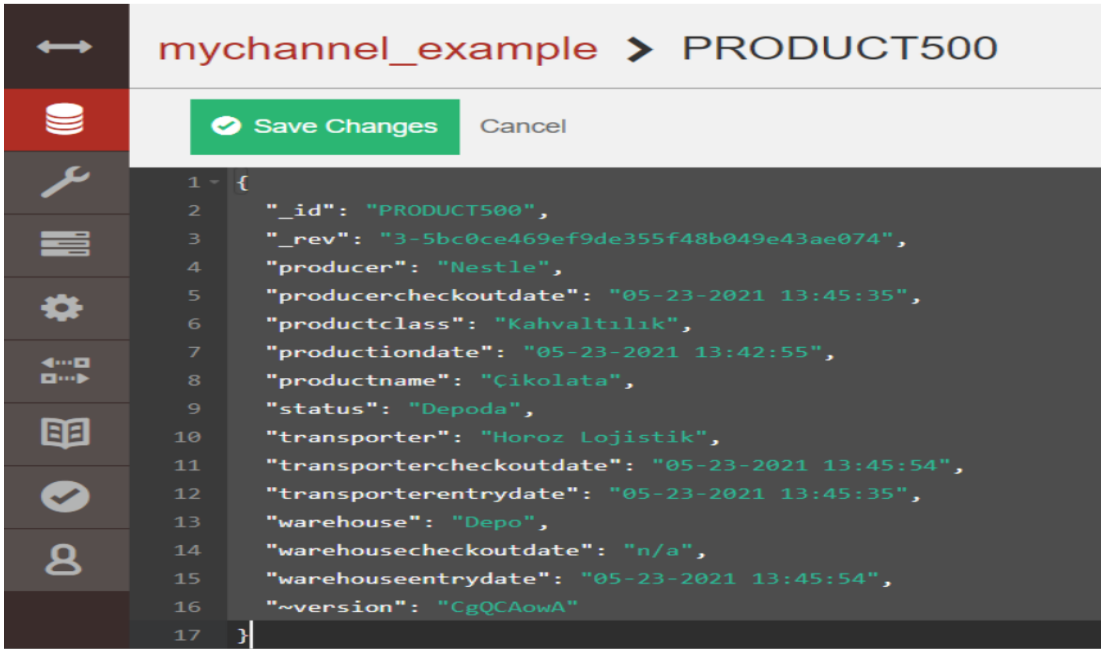
Ürünün durumunu öğrenmek için "queryProduct" fonksiyonu kullanılır.Ürünün ismi yollandıktan sonra ürünün o anki tüm bilgileri dökülür.

Ürünün o anda geçmiş tüm hareketlerini görmek için "getHistoryForAsset" fonksiyonunu kullanırız.Değişken olarak istenilen ürün ismi girildikten sonra o ürüne o ait o zamana kadar tüm bilgiler listelenir.Şekil 2.16'da bir ürüne ait tüm hareketler görülebilir.

Bütün işlemler başarılı bir şekilde gerçekleştirildiğinde Şekil 2.17'de belirtilen defter kaydı oluşturulmaktadır ve oluşan bu defter kaydı her aşamada güncellenmektedir. Belirtilen defter kaydındaki tarihler ve statü işlemler gerçekleştiği anda deftere basılmaktadır.

```
{
    "result": [
        {
            "TxId": "19464a8d389c766d974d7b4944316c890e0e1c570f03ca9067b0fcee5bb3654d",
            "Value": {
                "productname": "Çikolata",
                "productclass": "Kahvaltılık",
                "producer": "Nestle",
                "productiondate": "05-23-2021 13:42:55",
                "producercheckoutdate": "05-23-2021 13:45:35",
                "transporter": "Horoz Lojistik",
                "transporterentrydate": "05-23-2021 13:45:35",
                "transportercheckoutdate": "05-23-2021 13:45:54",
                "warehouse": "Depo",
                "warehouseentrydate": "05-23-2021 13:45:54",
                "warehousecheckoutdate": "n/a",
                "status": "Depoda"
            },
            "Timestamp": "2021-05-23 13:45:54.111 +0000 UTC",
            "IsDelete": "false"
        },
        {
            "TxId": "da2d88064827025 3d04ae1eccdd43838e62f76018b3671cd512519c67747a314",
            "Value": {
                "productname": "Çikolata",
                "productclass": "Kahvaltılık",
                "producer": "Nestle",
                "productiondate": "05-23-2021 13:42:55",
                "producercheckoutdate": "05-23-2021 13:45:35",
                "transporter": "Horoz Lojistik",
                "transporterentrydate": "05-23-2021 13:45:35",
                "transportercheckoutdate": "n/a",
                "warehouse": "n/a",
                "warehouseentrydate": "n/a",
                "warehousecheckoutdate": "n/a",
                "status": "Nakliyede"
            },
            "Timestamp": "2021-05-23 13:45:35.771 +0000 UTC",
            "IsDelete": "false"
        },
        {
            "TxId": "067c6b62e4480231ed5114489ad1fbe16bfe34a18f24a4179098bee42a39fd9d",
            "Value": {
                "productname": "Çikolata",
                "productclass": "Kahvaltılık",
                "producer": "Nestle",
                "productiondate": "05-23-2021 13:42:55",
                "producercheckoutdate": "n/a",
                "transporter": "n/a",
                "transporterentrydate": "n/a",
                "transportercheckoutdate": "n/a",
                "warehouse": "n/a",
                "warehouseentrydate": "n/a",
                "warehousecheckoutdate": "n/a",
                "status": "Uretildi"
            },
            "Timestamp": "2021-05-23 13:42:55.351 +0000 UTC",
            "IsDelete": "false"
        }
    ],
    "error": null,
    "errorData": null
}
```


Şekil 2.16. Ürünün Geçmişi



Şekil 2.17. CouchDB Görünümü

3.SONUÇLAR VE ÖNERİLER

Blok zincir teknolojisi günümüzde hızlı gelişmeye devam etmektedir.Bu gelişim sayesinde tedarik zincirine sahip şirketler bu teknolojiyi bu zincirin verimliliğini artırabilmek için entegre etmeye çalışmaktadır.

Çoğu durumda, bu şirketler tedarik zinciri görünürlüğüne ve ürün takibine yardımcı olmak için blok zinciri kullanabilir, ancak bazıları bunu işlemleri kolaylaştırmak ve bilgi, mal ve malzeme akışını hızlandırmak için bir araç olarak kullanabilir. Blok zincir, artan tedarik zinciri şeffaflığının yanı sıra tedarik zinciri boyunca düşük maliyet ve risk sağlayabilir. Blok zincir, tedarik zincirinde uçtan uca daha şeffaf ve doğru bir izleme sağlayabilir: Kuruluşlar, fiziksel varlıkları dijitalleştirebilir ve tüm işlemlerin merkezi olmayan değişmez bir kaydını oluşturarak, varlıkların üretimden teslimata veya son kullanıcı tarafından kullanılmasına kadar takip edilmesini mümkün kılar. Bu artan tedarik zinciri şeffaflığı, hem işletmelere hem de tüketicilere daha fazla görünürlük sağlar.

Blok zincir, yüksek değerli ürünler için sahtekarlığı azaltmaya yardımcı olmak için tedarik zinciri şeffaflığını artırabilir. Blok zincir, şirketlerin bileşenlerin ve bitmiş ürünlerin her bir alt yükleniciden nasıl geçtiğini anlamalarına ve kar kayıplarını azaltmalarına ve ayrıca sahte ürünlerin etkisini azaltarak veya ortadan kaldırarak son piyasa kullanıcılarına olan güveni artırmalarına yardımcı olabilir.

Tez de kullandığımı Hyperledger Fabric ile tedarik zinciri olan şirketler bunu kolayca sağlayabilir.Blok zincirin sağladığı tüm avantajları bu platform ile elde edilebilir.Akıllı sözleşmeler ile yerleştirdiğimiz mantığın dışına çıkması imkansızdır.Kurulabilecek yapılarla her organizasyonun tüm işlemleri görmesi mümkün değildir.Böylelikle zincirde bulunan tüm organizasyonlar için güvenli ortam oluşturulmuştur.Ayrıca blok zincir değiştirilemez olduğu için verilerde oynama yapılması imkansızdır.Bu da kurumsal güvenirliği artıracaktır.

Blok zincir, ilgili tedarik zincirindeki tüm taraflara aynı bilgilere erişim sağlayarak iletişim veya aktarım veri hatalarını potansiyel olarak azaltır. Verileri doğrulamak için

daha az zaman harcanabilir ve mal ve hizmetlerin sunulmasına daha fazla zaman harcanabilir.

Hyperledger Fabric platformu bu verileri yönetmek için kullandığı özellikler ile tedarik zincirine üstte bahsedildiği gibi bir çok özellik sağlar.TLS sertifikası sayesinde verilerin güvenliğini sağlar.

Geliştirme anlamında Postman ile veri güncelleme ve yeni veri gönderme işlemleri yerine daha modern olabilecek bir arayüz yapılabilir.Ayrıca tedarik zincirleri operasyonlarında akıllı sözleşmemizde yer alan fonksiyonlar dışında farklı fonksiyonlarda kullanabilir.

Sonuç olarak,tedarik zincirinde, üreticiden tüketiciye kadar olan bütün sürecin, blok zincir teknolojisi entegre edilerek takip edilmesi sağlanmıştır. Kullanıcıların güvenli, şeffaf ve konforlu bir şekilde bu süreçlerin takibini sağlayabileceği bir ortam hazırlanmıştır. Tezde kullanılacak verilerin, güvenilir ve doğrulanmış bir şekilde depolanması ve işlenmesi başarılı bir şekilde gerçekleşmiştir.Gelecekte blok zincire hızlı bir şekilde geçiş sağlanabilmesi amacıyla yeni bir standart belirlenebilir.Bu standart ile hazır geçiş sağlanarak daha hızlı bir adaptasyon süreci sağlanabilir.Akıllı kontratların güvenliği ihlal edilme ihtimaline karşın yeni güvenlik protokolleri geliştirilebilir.Tedarik zincirlerinde bunlar gibi yeni özellikler blok zinciri güvenlik açısından olabileceği en yüksek seviyeye çıkaracaktır.

KAYNAKLAR

[1] What's the Difference Between Blockchain and Supply Chain, https://supplychaingamechanger.com/whats-the-difference-between-blockchain-and-supply-chain/ , (Ziyaret Tarihi:10 Mayıs 2021)

[2] How Blockchain Will Redefine Supply Chain Management, https://knowledge.wharton.upenn.edu/article/blockchain-supply-chain-management/ ,(Ziyaret Tarihi:10 Mayıs 2021)

[3] Günen E., En Sade Anlatımla Blockchain Nedir, Nasıl Çalışır?, https://tr.cointelegraph.com/news/a-simple-explanation-of-what-is-blockchain-and-how-its-works,(Ziyaret Tarihi:10 Mayıs 2021)

[4] Gatteschi V., Lamberti F., Demartini C., Pranteda C., ve Santamaría V., To Blockchain or Not to Blockchain: That Is the Question, *IT Professional*,2019, **20**(2),62-74.

[5] Güran B., Blockchain nedir? Nerelerde ve Nasıl Kullanılır?,https://www.turktoyu.com/blockchain-nedir-nerelerde-ve-nasil-kullanilir,(Ziyaret Tarihi:10 Mayıs 2021)

[6] Nakamoto S.,Bitcoin: A Peer-to-Peer Electronic Cash System, , https://bitcoin.org/bitcoin.pdf/,(Ziyaret Tarihi:10 Mayıs 2021)

[7] Geroni D., What Is A Public Blockchain? Beginner's Guide, https://101blockchains.com/what-is-a-public-blockchain/,(Ziyaret Tarihi:10 Mayıs 2021)

[8] Seth S., Public, Private, Permissioned Blockchains Compared, https://www.investopedia.com/news/public-private-permissioned-blockchains-compared/,(Ziyaret Tarihi:10 Mayıs 2021)

[9] Özel, Herkese Açık ve Konsorsiyum Blockchainlerin Farkları, https://academy.binance.com/tr/articles/private-public-and-consortium-blockchains-whats-the-difference,(Ziyaret Tarihi:10 Mayıs 2021)

[10] The great chain of being sure about things, https://www.economist.com/briefing/2015/10/31/the-great-chain-of-being-sure-about-things,(Ziyaret Tarihi:10 Mayıs 2021)

[11] Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu.,A Survey on Ethereum Systems Security: Vulnerabilities, Attacks, and Defenses.

ACM Comput. Surv,2020,**53**(3),1-43.

[12] Bhaskar, Nirupama & Chuen, David., Bitcoin Mining Technology. *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*.,2015,**3**(2), 45-65.

[13] What is the Blockchain data structure?, https://cryptoticker.io/en/blockchain-data-structure/,(Ziyaret Tarihi:10 Mayıs 2021)

[14] Kasthala V., Blockchain key characteristics and the conditions to use it as a solution, https://medium.com/swlh/blockchain-characteristics-and-its-suitability-as-a-technical-solution-bd65fc2c1ad1,(Ziyaret Tarihi:10 Mayıs 2021)

[15] Sunyaev A. (2020) Distributed Ledger Technology. In: Internet Computing. Springer, Cham. https://doi.org/10.1007/978-3-030-34957-8_9

[16] Frankenfield J., Distributed Ledger Technology (DLT), https://www.investopedia.com/terms/d/distributed-ledger-technology-dlt.asp#:~:text=Distributed%20Ledger%20Technology%20(DLT)%20is,and%20accurate%20manner%20using%20cryptography.,(Ziyaret Tarihi:10 Mayıs 2021)

[17] What Is a Blockchain Consensus Algorithm?, https://academy.binance.com/en/articles/what-is-a-blockchain-consensus-algorithm,(Ziyaret Tarihi:10 Mayıs 2021)

[18] Hertig A.,What is Proof of Work,https://www.coindesk.com/what-is-proof-of-work,(Ziyaret Tarihi:10 Mayıs 2021)

[19] Finney H.,RPOW - Reusable Proofs of Work, https://nakamotoinstitute.org/finney/rpow/,(Ziyaret Tarihi:10 Mayıs 2021)

[20] Laurie, Ben; Clayton, Richard.,Proof-of-work proves not to work, https://www.cl.cam.ac.uk/~rnc1/proofwork2.pdf,(Ziyaret Tarihi:10 Mayıs 2021)

[21] Frankenfield J.,Proof of Stake(POS), https://www.investopedia.com/terms/p/proof-stake-pos.asp,(Ziyaret Tarihi:10 Mayıs 2021)

[22] Practical Byzantine Fault Tolerance,https://www.geeksforgeeks.org/practical-byzantine-fault-tolerancepbft/,(Ziyaret Tarihi:10 Mayıs 2021)

[23] Miguel Castro and Barbara Liskov., Practical Byzantine fault tolerance, *In Proceedings of the third symposium on Operating systems design and implementation* ,1999,**14**(5), 173–186.

[24] Welcome to Ethereum, https://ethereum.org/en/,(Ziyaret Tarihi:10 Mayıs 2021)

[25] Proof of Authority Nedir?, https://academy.binance.com/tr/articles/proof-of-authority-explained,(Ziyaret Tarihi:10 Mayıs 2021)

[26] Parity Ethereum,https://github.com/openethereum/parity-ethereum,(Ziyaret Tarihi:10 Mayıs 2021)

[27] Walters S.,Delegated Proof of Stake, https://www.coinbureau.com/education/delegated-proof-stake-dpos/,(Ziyaret Tarihi:10 Mayıs 2021)

[28] Guide: What is Bitcoin and how does it work?, https://www.bbc.co.uk/newsround/25622442,(Ziyaret Tarihi:10 Mayıs 2021)

[29] Global Bitcoin Nodes Distribution, https://bitnodes.io/#global-bitcoin-nodes-distribution,(Ziyaret Tarihi:10 Mayıs 2021)

[30] Rodeck D., What Is Ethereum And How Does It Work?, https://www.forbes.com/advisor/investing/what-is-ethereum-ether/,(Ziyaret Tarihi:10 Mayıs 2021)

[31] Founder of the Apache Software Foundation Joins Linux Foundation to Lead HyperledgerProject,https://www.hyperledger.org/announcements/2016/05/19/founder-of-the-apache-software-foundation-joins-linux-foundation-to-lead-hyperledger-project,(Ziyaret Tarihi:10 Mayıs 2021)

[32] USTA A.,DOĞANTEKİN S., "Blockchain 101 v2", https://bkm.com.tr/wp-content/uploads/2019/08/15082019_kitap.pdf,,(Ziyaret Tarihi:10 Mayıs 2021)

[33] Akıllı Sözleşme Platformları Listesi ve Detayları Hakkında**,** https://www.geliyoobilisim.com/akilli-sozlesme-platformlari-listesi-ve-detaylari-hakkinda/,(Ziyaret Tarihi:10 Mayıs 2021)

[34] Corda Platform, https://www.r3.com/corda-platform/,(Ziyaret Tarihi:10 Mayıs 2021)

[35] Strebko J., Romanovs A.,The Advantages and Disadvantages of the Blockchain Technology, *2018 IEEE 6th Workshop on Advances in Information*,2018,**10**(11),1-6.

[36] Blockchain – Hot or Not, https://softengi.com/blog/blockchain-hot-or-not/,(Ziyaret Tarihi:10 Mayıs 2021)

[37] Lee T., Bitcoin's insane energy consumption, explained, https://arstechnica.com/tech-policy/2017/12/bitcoins-insane-energy-consumption-explained/,(Ziyaret Tarihi:10 Mayıs 2021)

[38] Today's Cryptocurrency Prices by Market Cap**,** https://coinmarketcap.com/,(Ziyaret Tarihi:10 Mayıs 2021)

[39] Varma J.,Blockchain in Finance, *Vikalpa*, 2019,**44**(1),1-11.

[40] Sandner P., Blockchain in Healthcare, https://philippsandner.medium.com/blockchain-in-healthcare-fbbd2989a9dc,(Ziyaret Tarihi:10 Mayıs 2021)

[41] Alharby M., Moorsel A., *Blockchain-Based Smart Contracts : A Systematic Mapping Study*,2018

[42] Sharma, Vishal and N. Lal. "A Detail Dominant Approach for IoT and Blockchain with their Research Challenges." *2020 International Conference on Emerging Trends in Communication, Control and Computing* ,2020,1-6.

[43] Blockchain in Supply Chain and Transportation: Benefits, Use Cases, Limitations, and Opportunities, https://www.altexsoft.com/blog/blockchain-supply-chain/,(Ziyaret Tarihi:10 Mayıs 2021)

[44] What is Hyperledger?, https://www.hyperledger.org/,(Ziyaret Tarihi:10 Mayıs 2021)

[45] Hyperledger Fabric nedir?, https://www.ibm.com/tr-tr/topics/hyperledger,(Ziyaret Tarihi:10 Mayıs 2021)

[46] Visual Studio Code, https://code.visualstudio.com/,(Ziyaret Tarihi:10 Mayıs 2021)

[47] Sunucu Nedir?, https://www.turhost.com/sunucu-nedir/#serp,(Ziyaret Tarihi:10 Mayıs 2021)

[48] Go Programlama Dili Nedir? Google Go ve Go Dili ile Neler Yapılabilir?, https://wmaraci.com/nedir/golang,(Ziyaret Tarihi:10 Mayıs 2021)

[49] Javascript Nedir?, https://www.biltektasarim.com/blog/javascript-nedir,(Ziyaret Tarihi:10 Mayıs 2021)

[50] Introduction, https://docs.couchdb.org/en/stable/intro/index.html,(Ziyaret Tarihi:10 Mayıs 2021)

[51] Docker, https://www.docker.com/,(Ziyaret Tarihi:10 Mayıs 2021)

[52] Postman, https://www.postman.com/,Ziyaret Tarihi:10 Mayıs 2021)

KİŞİSEL YAYIN VE ESERLER

[1] **Dikilitaş Y**.,Toka K.O.,Sayar A.,Current Research Areas in Blockchain, *3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications*,Ankara,Türkiye,11-13 Haziran 2021

ÖZGEÇMİŞ

Lise öğrenimini Necip Fazıl Anadolu Lisesi'nde tamamladı. 2011 yılında girdiği İzmir Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümü'nden 2016 yılında mezun oldu. 2021 yılından itibaren Kocaeli Üniversitesi Bilgisayar Mühendisliği bölümünde Araştırma Görevlisi olarak görev yapmaya başladı. 2019 yılında Kocaeli Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine başladı. Yüksek lisans eğitiminde blok zincir teknolojileri konusunda çalışmaktadır.