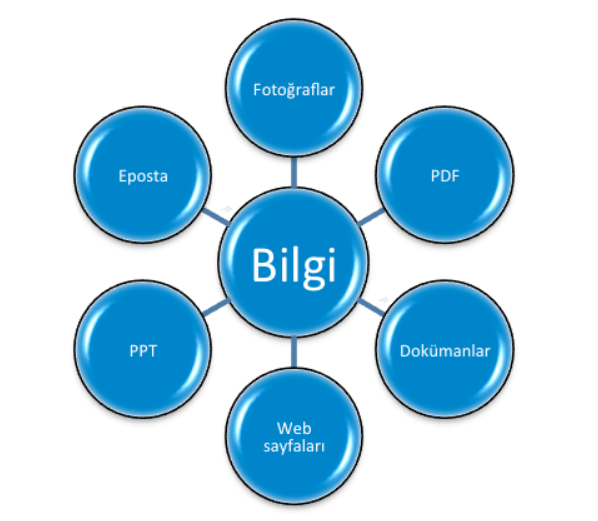
**Blok zinciri teknolojisi ve Bitcoin**

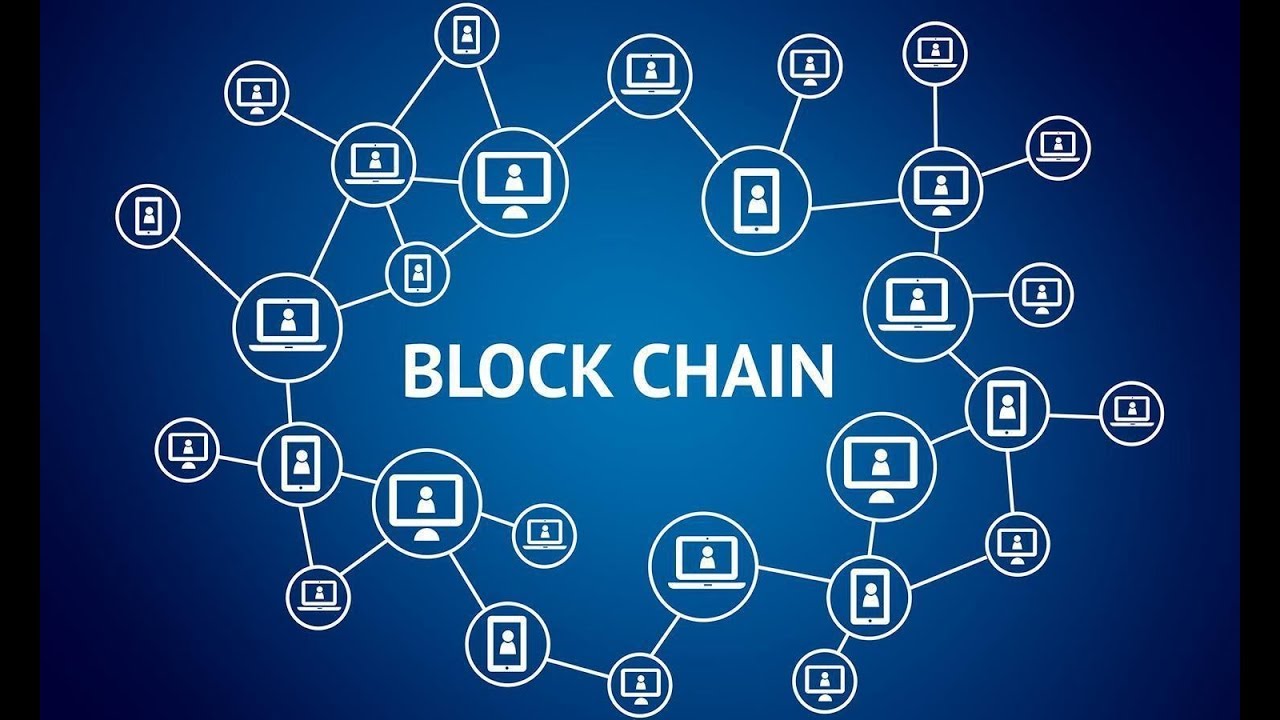
yer, kedi, iç mekan içeren bir resim

Açıklama otomatik olarak oluşturulduBlok zinciri 2008 yılında ortaya atılmış, 2009 yılında ise Bitcoin sanal para birimi ile birlikte tanınmaya başlamıştır. Blok zinciri, dağıtık, paylaşılan, şifrelenmiş, geri dönüşü olmayan ve bozulmayan bir bilgi deposudur. Blok zinciri, ağ yardımı ile sistemi kullanan kullanıcılar arasındaki işlemlerin tümünü doğrulayarak saklayan bir sistemdir. Bu yüzden bütünlüğüne güvenilir bloklar ve bu blokları oluşturan sorgulanabilir işlemlerden oluşan bir veri tabanı olarak tanımlanmaktadır.

Blok zinciri sisteminde işlemler bloklar halinde tutulur ve bu bloklar birbirine bağlanarak zincir oluşturulur. Belli kurallar çerçevesinde oluşturulan bloklar sisteme yazılmaktadır. Daha sonra blok tüm dağıtık kayıt defterlerine yayılır ve eklenir. Yeni blok oluşturmada bir önceki bloğa ait özet alınır ve ikinci blok üretilerek zincire ekleme yapılmaktadır.

Bu yapı tüm blokları birbirine bağlayan ve bir önceki bloğun özeti ile beraber olacak biçimde devam eden bir yapı ile sürdürülür. Bir işlem gerçekleştiğinde mevcut ağ üzerinden yayınlanır ve şifreleme algoritmaları ile bu işlem doğrulanarak blok oluşturulur. Sisteme dahil olan her düğüm, sistemdeki herhangi iki kişi tarafından yapılan bu işlemi onaylayarak kaydını tutar. Bu sayede blok doğrulanır, sonrasında bu bilgi asla değiştirilemez veya silinemez.

Her blok birbirine zincirlenerek eklenmeye devam eder. Böylece başka biri onları hiçbir zaman değiştiremez. Sistemde bulunan her bir kullanıcı bir düğümü ifade eder. Sisteme katılan her düğüm, kendi başına bir blok zinciri kopyasına yani kayıt defterine bir başka deyişle veri tabanına sahiptir. Bu defter bir uçtan uca protokolü kullanılarak diğer düğümlerle senkronize edilir. Bu sayede aracı ortadan kaldırılmakta ve merkezi bir otorite zorunluluğu da gerekmemektedir. Bir düğüm başarısız olur veya işlevini durdurursa, kalan düğümler arızalı yerin yokluğunda tüm işlem ayrıntılarını muhafaza eder. Bu şekilde sistem gerçek zamanlı bilgi sağlamakta ve işlemlerin hata ya da başarısızlık oranlarını azaltmaktadır. Birden fazla tarafın bulunduğu Blok zinciri sisteminde, sisteme eklenmesi istenen herhangi bir işlemin doğrulanabilmesi için genel kabul görmüş kurallara uygunluğunun kontrol edilmesi gerekmektedir.



**Blok zinciri sistemi temel bazı kriterlere sahip olup bu standartlar üzerine kurulur. Bu özellikler aşağıda kısaca açıklanmıştır.**

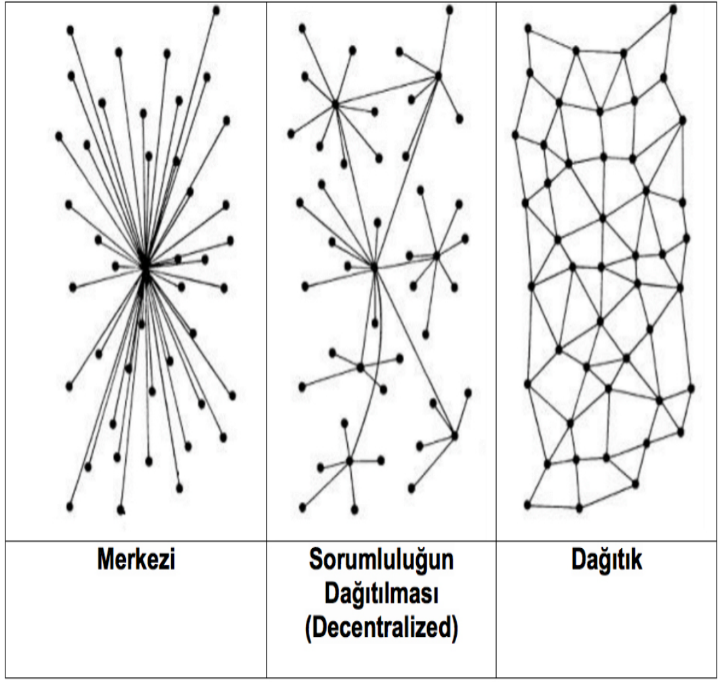
**Dağıtık:** Blok zincirinin en temel özelliği verilerin tek bir yerde tutulmaması, dağıtılabilir şekilde kaydedilmesi, depolanması ve güncellenmesidir.

**Şeffaf:** Blok zinciri sistemi ile verilerin kaydı her düğüme şeffaf olup veriler geriye dönük olarak doğrulanabilir. Bu nedenle blok zinciri güvenilir kabul edilmektedir.

**Bağımsız:** Mutabakat yapısı sayesinde blok zincir sistemindeki her düğüm verileri güvenli bir şekilde aktarabilir, merkezi bir sisteme ihtiyaç duyulmamaktadır.

**Değiştirilemez:** Blok zincirine eklenen kayıt hiçbir şekilde güncellenemez, silinemez ve kayıt kalıcı olarak saklanır. Kaydın değiştirilebilmesi çeşitli saldırı tipleri dışında mümkün değildir.

**Kimlik gizliliği:** Blok zinciri sisteminde düğümler kimlik belirtmeden veri aktarımı gerçekleştirebilir. Bu işlem için kişinin blok zinciri adresinin bilinmesi yeterlidir.



**Blok Zinciri Çeşitleri ise** (Types of Blockchain) Blok zinciri izin mekanizmasına göre üç çeşitten oluşmaktadır:

* **Açık**
* **Özel ve**
* **Konsorsiyum blok zinciri.**

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Blok Zinciri Mimarisi ve Çalışma Prensipleri (Architecture and Working Principles of Blockchain)**

Blok zinciri işlemleri bloklar içinde saklanmakta ve bloklar ile zincir oluşturmaktadır. Bu bloklar ile zincir belirli özelliklere sahip olup belirli kurallar ile bir araya gelmektedir. Aşağıda bununla ilgili temel kavramlar açıklanmıştır.

**Blok (Block):** Blok zinciri sisteminde verilerin saklanması blok olarak isimlendirilir. Bloklar zincir biçiminde sıralanır. Bu zincir yapısı içerisinde yer alan ilk blok başlangıç bloğu olarak isimlendirilir. Her blok başlık ve gözde kısımlarından oluşur.

**Dijital imza (Digital signature):** Her kullanıcının sahip olduğu özel anahtar (private key) ve açık anahtar (public key) olmak üzere iki farklı anahtar mevcuttur. İşlemleri imzalamak için özel anahtar kullanılmaktadır. Dijital imzalanmış tüm işlemler ağ içerisinde herkes tarafından ulaşılabilecek biçimde açık anahtarlar kullanılarak erişilmektedir. Dijital imza imzalama ve doğrulama olmak üzere iki aşamadan oluşmaktadır.

**Mutabakat mekanizması (Consensus mechanism):** Blok zinciri ağındaki verinin tüm kayıt defterlerinde senkronize biçimde bulunması gerekir. Bu durumun oluşması için ağ genelinde mutabakat sağlanması gerekir. Blok zinciri sisteminde blokların güvenilir olduğu bilgisini sağlamak için Kriptografik Özetleme (Cryptographic Hashing) ve zaman bilgisi kullanılmaktadır.

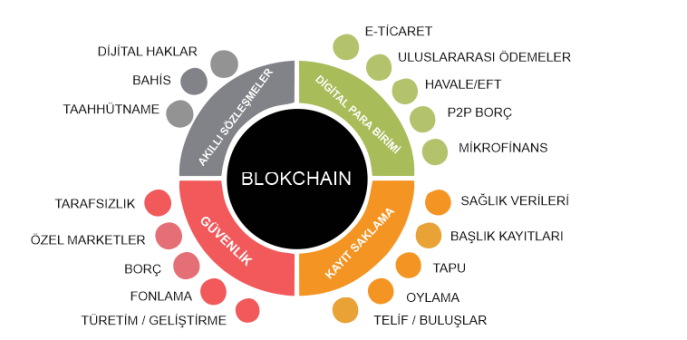
**Çatallaşma işlemi (Fork):** Çatallaşma, herhangi bir kaynak kodu ele alıp uyarlama yapıp farklı ve yeni bir çalışma ortaya koyma işlemidir. Bu duruma açık kaynak kodlu projelerde çok sık rastlanmaktadır. Blok zinciri yazılımının yeni sürümü yayınlandığında, uzlaşma kuralındaki yeni yazılım düğümlere dağıtılır. Blok zinciri yapısı dağıtık olduğundan ve merkezi kontrollü olmadığından eğer değişiklik isteği büyük bir grup tarafından kabul görürse bu andan itibaren yeni kurallar belirlenerek yeni çatallaşma duyurusu yapılır [14]. Böyle bir durumda dört olay oluşmaktadır:

* Yeni düğümler, eski düğümler tarafından gönderilen blok işlemine katılırlar.
* Yeni düğümler, eski düğümler tarafından gönderilen blok işlemine katılamaz.
* Eski düğümler, yeni düğümler tarafından gönderilen blok işlemine katılırlar.
* Eski düğümler, yeni düğümler tarafından gönderilen blok işlemine katılamazlar.

**Akıllı sözleşmeler (Smart contracts):** Akıllı sözleşme kullanıcıların iletişimini daha güvenli bir biçimde yürütebilmek amacıyla 1994 yılında ortaya çıkmış bir kavramdır. Fikrin en önemli örneği Bitcoin’de kısmen bulunmakla birlikte, kod yazarak sözleşme yazmayı insanlara gösteren ve 2014 yılında ortaya çıkan Ethereum'dur.

**Blok Zinciri Kullanım Alanları (Usage Areas of BlockChain)**

Blok zinciri finans sektörü ile birlikte yayılsa da birçok alanda kullanılmaktadır. Bitcoin ile tanınan teknoloji daha sonra birbirinden farklı sektörlerde kullanılmaya başlanmıştır.



**Bitcoin’i Blok zinciri ile proje geliştirilmesi**

Bitcoin Blockchain'de üç ana bölümün varlığını fark edeceksiniz:

* Madencilik süreci aracılığıyla işlem bloklarının doğrulanması ile iç çekirdeği.
* Blockchain üzerindeki işlemlerin yerel yönetimi.
* Blockchain'in Eşler Arası ağ iletişimi.

Bu hikayede, işleyişi Bitcoin'e eşdeğer olacak bir Blockchain'in iç çekirdeğinin nasıl oluşturulacağını öğrenmeyi öneriyorum. JavaScript programlama dilini kullanacağım ancak sunulan kavramlar diğer dillerde kolayca uygulanabilir.

**Bir İşlem Bloğunun Modellenmesi**

Bitcoin Blockchain, SHA-256 imzalarıyla birbiri ardına bağlanan bloklardan oluşur. Her blok, kullanıcıları tarafından yapılan işlemleri içerir. Şu an için işlem yönetimi kısmını uygulamamayı seçtiğimiz için Blockchain'imizin blokları string veri olacak.

Bir blok aşağıdaki bilgileri içerecektir:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bir Bloğun SHA-256 İmzasının Hesaplanması

Blok zincirimizin bloklarının hash'i, tüm detayları wikipedia'da bulabileceğiniz kriptografik hash algoritması SHA-256'ya dayanarak hesaplanacaktır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Artık bir bloğu girdi olarak alan ve JS-256 kod kitaplığındaki sha256 işlevini kullanarak SHA-256 imzasını bir dize olarak döndüren bir hesaplaHash işlevi oluşturabilirim:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bir Blok Madenciliği

Bloğumuz için geliştirme çalışmalarını neredeyse bitirdik. Bitcoin Blockchain tarafından kullanılan Proof-of-Work konsensüs algoritmasının kalbinde yer alan madencilik sürecini hala uygulamamız gerekiyor.Madencilik, blok için farklı SHA-256 imza değerleri denenerek bu imzanın tanımlanmış sayıda sıfırla başlamasından oluşur. Buradaki sıfır sayısı, Blockchain'imizin bloklarını çıkarmanın zorluğunu temsil ediyor. Bitcoin Blockchain'de olduğu gibi, bu zorluk zamanla değişebilir. Dolayısıyla bu zorluğu oluşturacağımız mineBlock fonksiyonunun parametresi olarak geçmeliyiz. Block sınıfının içindeki bu fonksiyon içinde, verilen kısıtlamalar dahilinde bloğun madenciliği yapılmadan önce yapılan denemelerin sayısını temsil eden nonce değişkenini 0 olarak ayarlayacağım. Ardından, bu hash, zorluk giriş parametresine eşit sıfır sayısını başlatana kadar bloğun ve döngünün oluşturulmasında hesaplanan ilk hash'i alacağım. Her yinelemede, istenen sayıda sıfırla başlayan karma bulunana kadar yeni bir SHA-256 imzası hesaplanacaktır.

Bu bize Block sınıfı içinde aşağıdaki işlevi verir:

Block.prototype.mineBlock = function(difficulty) {

this.nonce = 0;

var zeros = "0".repeat(difficulty);

while (this.hash.substring(0, difficulty) != zeros) {

this.nonce++;

this.hash = calculateHash(this);

}

};

**Blockchain Modelleme**

Blockchain, verilerinin bütünlüğünü sağlamaktan sorumlu dağıtılmış bir veritabanıdır. Bunu temsil etmek için, bu Blockchain üzerindeki madencilik bloklarının zorluğunun bir T anında görüntüsü olacak bir zorluk özelliğine sahip bir Blockchain sınıfı tanımlayacağım. Ayrıca Blockchain içinde saklanan farklı blokları depolamak için kullanılacak bir dizi tanımlayacağım. Bütün bunlar bize Blockchain sınıfı için aşağıdaki kodu verir:

class Blockchain {

constructor(difficulty) {

this.difficulty = difficulty;

this.blocks = [];

}

}

Yeni Bloklar Oluşturma ve Madencilik

Geliştirmeyi öğrendiğiniz Blockchain, Blockchain'in mevcut zorluklarını hesaba katarak yeni blokların oluşturulmasına ve madenciliğine izin verecektir. Bunu yapmak için, girdi olarak string formatında veri alan ve Blockchain için yeni bir blok çıkaran bir newBlock fonksiyonu ekliyorum. Bu yeni oluşturulan blok, öncekiHash özelliği aracılığıyla Blockchain'in son bloğunun hash'ine bağlanır. Ardından Blockchain'e bir blok eklemekten sorumlu bir addBlock işlevi tanımlıyorum. Blok eklemeden önce, eklenen bloğun mineBlock yöntemi çağrılarak madencilik işlemi gerçekleştirilecektir. Son olarak, genesis bloğunun oluşturulabilmesi için Blockchain sınıfının yapıcı kodunun güncellenmesi gerekmektedir. Bu bloğun 0'da bir indeksi ve boş bir öncekiHash özelliği olacaktır.

Bu, bu noktada Blockchain sınıfı için aşağıdaki kodu verir:

class Blockchain {

constructor(difficulty) {

this.difficulty = difficulty;

this.blocks = [];

// Add Genesis Block

var genesisBlock = new Block(0, null, Date.now(), "Genesis block");

genesisBlock.mineBlock(this.difficulty);

this.blocks.push(genesisBlock);

}

}

Blockchain.prototype.newBlock = function (data) {

var latestBlock = this.blocks[this.blocks.length - 1];

return new Block(latestBlock.index + 1, latestBlock.hash,

Date.now(), data);

}

Blockchain.prototype.addBlock = function(block) {

block.mineBlock(this.difficulty);

this.blocks.push(block);

}

**Blockchain'in Geçerliliğini Kontrol Etme**

Bitcoin Blockchain, kullanıcılarına geçerliliğini sağlamak için sakladığı blokların verilerinin bütünlüğünü garanti eder. Blockchain'imizin bütünlüğünü kontrol etmek için, bunu kontrol etmek için Blockchain sınıfında üç yöntem tanımlayacağım:

İlk blok geçerli

Yeni bir blok, önceki bloğa göre geçerlidir

Blockchain geçerl Bu yöntemlerin uygulanabilmesi için göz önünde bulundurmamız gereken geçerlilik kriterlerinin tanımlanması gerekmektedir. Bu nedenle, ilk blok, indeksi 0'a eşitse, önceki bir önceki Hash'e sahip değilse ve son olarak da hash'i doğruysa geçerlidir. Yeni bir blok, indeksi önceki bloğun indeksine göre 1 artırılırsa, önceki Hash alanı önceki bloğun hash'i ile doldurulursa ve son olarak hash'inin kendisi tutarlı ise, Blockchain'in önceki bloğunun yanında geçerlidir. Son olarak, bir Blok Zinciri geçerlidir ve ilk blok geçerli olduğunda ve onu oluşturan her blok kendinden önceki bloğun yanında geçerli olduğunda verilerinin bütünlüğü garanti edilir. Bütün bunlar bize Blockchain sınıfı için aşağıdaki tam kodu verir:

class Blockchain {

constructor(difficulty) {

this.difficulty = difficulty;

this.blocks = [];

// Add Genesis Block

var genesisBlock = new Block(0, null, Date.now(), "Genesis block");

genesisBlock.mineBlock(this.difficulty);

this.blocks.push(genesisBlock);

}

}

Blockchain.prototype.newBlock = function (data) {

var latestBlock = this.blocks[this.blocks.length - 1];

return new Block(latestBlock.index + 1, latestBlock.hash,

Date.now(), data);

}

Blockchain.prototype.addBlock = function(block) {

block.mineBlock(this.difficulty);

this.blocks.push(block);

}

Blockchain.prototype.isFirstBlockValid = function() {

var firstBlock = this.blocks[0];

if (firstBlock.index != 0)

return false;

if (firstBlock.previousHash != null)

return false;

if (firstBlock.hash == null ||

calculateHash(firstBlock) != firstBlock.hash)

return false;

return true;

}

Blockchain.prototype.isValidBlock = function (block, previousBlock) {

if (previousBlock.index + 1 != block.index)

return false;

if (block.previousHash == null ||

block.previousHash != previousBlock.hash)

return false;

if (block.hash == null ||

calculateHash(block) != block.hash)

return false;

return true;

}

Blockchain.prototype.isBlockchainValid = function() {

if (!this.isFirstBlockValid())

return false;

for (var i = 1; i < this.blocks.length; i++) {

var block = this.blocks[i];

var previousBlock = this.blocks[i - 1];

if (!this.isValidBlock(block, previousBlock))

return false;

}

return true;

}

Blockchain.prototype.display = function() {

for (var i = 0; i < this.blocks.length; i++) {

var block = this.blocks[i];

var str = "Block #" + block.index + " [" +

"previousHash: " + block.previousHash + ", " +

"timestamp: " + block.timestamp + ", " +

"data: " + block.data + ", " +

"hash: " + block.hash + "]";

console.log(str);

}

}

Son olarak, Blockchain sınıfının görüntüleme yöntemine yapılan çağrı, içinde bulunan blokları ve bunların bağlantılarını görselleştirmemize izin verecektir.

var blockchain = new Blockchain(4);

var block1 = blockchain.newBlock("Second Block");

blockchain.addBlock(block1);

var block2 = blockchain.newBlock("Third Block");

blockchain.addBlock(block2);

var block3 = blockchain.newBlock("Fourth Block");

blockchain.addBlock(block3);

console.log("Blockchain Validity: " + blockchain.isBlockchainValid());

blockchain.display();

Executing this code in a browser such as Brave Browser produces the following result:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Blockchain'imizin Bozulması**

Blockchain'in geçerliliğini kontrol etme yöntemimizi daha fazla doğrulamak için, bozuk bir bloğu entegre ederek Blockchain'imizi bozacağız. Bu ekleme yapıldıktan sonra geriye Blockchain sınıfının isBlockchainValid yöntemini çağırmak kalır:

var blockchain = new Blockchain(4);

var block1 = blockchain.newBlock("Second Block");

blockchain.addBlock(block1);

var block2 = blockchain.newBlock("Third Block");

blockchain.addBlock(block2);

var block3 = blockchain.newBlock("Fourth Block");

blockchain.addBlock(block3);

console.log("Blockchain Validity: " + blockchain.isBlockchainValid());

blockchain.display();

var block4 = new Block(12, "falseHash", Date.now(), "Block Invalid");

blockchain.addBlock(block4);

console.log("Blockchain Validity: " + blockchain.isBlockchainValid());

blockchain.display();

Bu kodu Brave Browser gibi bir tarayıcıda çalıştırmak aşağıdaki sonucu verir:

Bu nedenle Blockchain'imiz bozuk olup olmadığını iyi tespit edebilir.metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Blokzincir Teknolojisinin Fayda ve Sakıncaları**

**Blokzincir Teknolojisinin Faydaları**

Merkezi olmayan dağınık bir veri tabanına sahip olması, verilerin herkesin erişimine açık olması ve ancak belli orandaki kullanıcının onaylanması halinde veri tabanına kaydedilebilmesi bu sistemin şeffaflığını göstermektedir. Bu özellik herhangi bir merkez tarafından bu sistemin kapatılmaması veya hacklenememesi gibi faydaları da sağlamaktadır.

Bu şeffaflık güvenilirliğe ve aracı kurumlara duyulan ihtiyacın da ortadan kalkmasına sebep olmaktadır. Söz konusu sistem gerek dijital imzalar gerekse kullanıcıların doğrulama yapma özelliklerinden dolayı güvenli ortam yaratmaktadır.

Söz konusu sistemin diğer bir özelliği de mahremiyettir. Sistemin sağladığı anonimlik özelliği sayesinde kişiler bu şekilde açık ve şeffaf bir sistemde dahi gizlilik ve mahremiyet sorunu yaşamadan anonim şekilde finansal veya hukuksal işlemleri rahatlıkla yapabilmektedirler.  
   
**Blokzincir Teknolojisinin Sakıncaları**

Blokzincir teknolojisinin faydaları olduğu kadar birçok sakıncası da bulunmaktadır.

Blokzincir   teknolojisinin sadece 14 yıllık bir geçmişi olduğu için kullanımı halen çok yaygınlaşmamıştır. Henüz bu sisteme tam güvenmeyen kişiler bulunmaktadır. Bu güvensizliğin nedeninin Bitcoin takas merkezlerinde yaşanan hırsızlık olaylarından kaynaklandığı düşünülmektedir.

Ancak meydana gelen bu hırsızlık olaylarının genellikle asıl sebebinin kişilerin çoklu imza hususuna gerekli hassasiyeti göstermemeleri ve çevrimiçi cüzdan ile entegrasyonu çevrimdışı cüzdanlara tercih etmeleridir. Bu sistemin diğer bir sakıncası ise Blokzincir siteminin işlem süresinin uzunluğu ve iş hacminin düşüklüğünden kaynaklanan gecikmeli işlemlerdir. Blokzincir teknolojisindeki işlemler algoritmaların çözümüyle doğrulandığından ve bunlar pek karışık olduğundan ticari ve finansal hayatın aradığı hızı yakalayamamıştır. Blokzincir teknolojilerinde anonimlik unsuru da söz konusu verilerin gerçek kişi olarak kime ait olduğunun tespitini zorlaştırmakta hatta imkânsız hale getirmektedir. Diğer bir ifadeyle sisteme yüklenen verilerin, uçtan uca şifresi kırılarak bir veya birkaç kişi tarafından erişilebilir hale gelmesi durumunda bile veri sahibi taraf ya da taraflar anonim olarak işlem yapabilmektedirler. Bu kişilerin tespiti yapıldığında ise sadece kullanıcı adı olarak seçilen isimleri gözükecek gerçek isimleri ve kimlikleri tespit edilemeyecektir. Bu durum da söz konusu sistemin başka bir sakıncasıdır.

**Sonuç ve öneriler**

Blok zinciri teknolojisi son dönemde popülaritesini artırmış güncel bir teknolojidir. Çok çeşitli alanlarda büyük bir dönüşüm potansiyeline sahiptir. Diğer yeni teknolojilerde olduğu gibi Blok zincir teknolojisinde de yeni projelerle denemeler yapılmaktadır. Yapılan her yeni projede açıklar kapanarak sistem geliştirilmektedir. Hızla gelişen teknolojinin doğru şekilde büyümesi ve yaygınlaşması, yenilik ve gelişimini ihlal etmeden, sistemin yönetim, güvenlik ve esnekliğini sağlamak için doğru denge üzerinde durmak gerekecektir. Blok zincirinde başarının en önemli sebeplerinden birisi etkili yönetimdir. Sistemin gizlilik ve siber güvenlik risklerine karşı dayanıklılığını artırmaktadır. Blok zinciri teknolojisindeki son gelişmeler yapay zekâ uygulamaları için yeni fırsatlar yaratmaktadır. Yapay zekâ teknolojileri birçok Blok zinciri zorluğunun çözülmesine yardımcı olacağı öngörülmektedir. Blok zinciri teknolojisinin vadettiği güvenli ve etkin veri aktarımı geleneksel sistemlerle zorlu bir yarışa gireceğinin göstergesidir. Mevcut sisteme sahip şirketlerin bu alanda Ar-Ge çalışmalarını hızlandırdıkları bilinmektedir. Mevcut sistemlerin yanında Blok zincirine yatırım yapmak dışında açacağı yeni alanlara da yatırımlar yapılması büyük önem kazanmaktadır. Bu alanlarda yapılan yatırımlar arttıkça Blok zinciri teknolojisinin daha hızlı gelişmesi ve yaygınlaşması mümkündür.

**Kaynakça**

<https://blokzincir.bilgem.tubitak.gov.tr/blok-zincir.html>

<https://www.ozgunlaw.com/makaleler/blokzincir-teknolojisi-nedir-ne-degildir-uygulama-alanlari-nelerdir-808>

<https://www.teknolojidenbihaber.com/teknoloji-devriminin-son-dalgasi-blok-zincir/>

TANRIVERDİ, M., UYSAL, M., & ÜSTÜNDAĞ, M. T. (2019). Blokzinciri teknolojisi nedir? ne değildir?: alanyazın incelemesi. Bilişim Teknolojileri Dergisi, 12(3), 203-217.‏

|  |
| --- |
|  |
| Gökhan, Ü. N. A. L., & ULUYOL, Ç. (2020). Blok zinciri teknolojisi. Bilişim Teknolojileri Dergisi, 13(2), 167-175.‏  Avunduk, H., & Hakan, A. Ş. A. N. (2018). Blok zinciri (blockchain) teknolojisi ve işletme uygulamaları: Genel bir değerlendirme. Dokuz Eylül Üniversitesi İktisadi İdari Bilimler Fakültesi Dergisi, 33(1), 369-384.‏ |
| https://en.wikipedia.org/wiki/SHA- |  |