Ethereum Teknik Raporu

YENİ NESİL BİR AKILLI SÖZLEŞME VE MERKEZİLEŞTİRİLMİŞ UYGULAMA PLATFORMU

Vitalik Buterin tarafından

Satoshi Nakamoto, Ocak 2009'da Bitcoin blok zincirini ilk kez harekete geçirdiğinde,
aynı anda iki radikal ve denenmemiş kavramı tanıtmak. İlki, merkezi olmayan bir "bitcoin"
Herhangi bir destek, gerçek değer veya merkezi yayıncı olmadan bir değeri koruyan eşler arası
çevrimiçi para birimi . Yani
Şu ana kadar, bir para birimi olarak "bitcoin", her ikisi de siyasi açıdan kamuoyunun dikkatini çekti.
Merkez bankası olmayan bir para biriminin özellikleri ve fiyattaki aşırı yukarı ve aşağı dalgalanmaları.
Bununla birlikte, Satoshi'nin büyük deneyinin aynı derecede önemli başka bir parçası daha var:
işlemlerin sırası üzerinde kamu anlaşmasına izin vermek için iş tabanlı blok zinciri. Bir uygulama
olarak Bitcoin
İlk dosyalama sistemi olarak tanımlanmalıdır: bir varlık 50 BTC'ye sahipse ve aynı anda aynı 50
BTC'yi
A ve B'ye, yalnızca ilk onaylanan işlem işlenecektir. Belirlemenin içsel bir yolu yoktur
daha önce gelen iki işlemden ve on yıllardır bu, ademi merkeziyetçi
dijital para birimi. Satoshi'nin blok zinciri, ilk güvenilir merkezi olmayan çözümdü. Ve şimdi dikkat
Bitcoin'in teknolojisinin bu ikinci kısmına ve blockchain konseptinin nasıl olabileceğine hızla
geçmeye başlaması
sadece paradan daha fazlası için kullanılır.
Yaygın olarak alıntı yapılan uygulamalar arasında, özel para birimlerini temsil etmek için blok zinciri
üzerinde dijital varlıkları kullanmayı ve
finansal araçlar ("renkli madeni paralar"), temelde yatan bir fiziksel cihazın sahipliği ("akıllı mülk"),
etki alanı adları ("Namecoin") gibi değiştirilemez varlıklar ve ayrıca
merkezi olmayan değişim, finansal türevler, eşler arası kumar ve blok zinciri üzerinde kimlik ve
itibar sistemleri. Bir diğer önemli sorgulama alanı da "akıllı sözleşmeler" dir - otomatik olarak
dijital varlıkları önceden belirlenmiş keyfi kurallara göre taşıyın. Örneğin, bir hazine sözleşmesi
olabilir
"A, günde en fazla X para birimi çekebilir, B günde en fazla Y, A ve B birlikte çekebilir
herhangi bir şeyi geri çekebilir ve A, B'nin geri çekilme yeteneğini kapatabilir ". Bunun mantıksal
uzantısı şudur:
merkezi olmayan özerk kuruluşlar (DAO'lar) - varlıkları içeren uzun vadeli akıllı sözleşmeler ve
bütün bir organizasyonun tüzüğünü kodlayın. Ethereum'un sağlamayı amaçladığı şey, yerleşik bir blok
zinciridir.
Kullanılabilecek "sözleşmeler" oluşturmak için kullanılabilen tam teşekküllü Turing-complete
programlama dili
keyfi durum geçiş işlevlerini kodlamak, kullanıcıların yukarıda açıklanan sistemlerden herhangi birini
oluşturmasına izin vermek için,
ve henüz hayal etmediğimiz diğer birçoklarının yanı sıra, sadece mantığı birkaç satır kodla yazarak.

İçindekiler

---

3. Sayfa

4. sayfa

Tarih
Merkezi olmayan dijital para birimi kavramı ve mülkiyet kayıtları gibi alternatif uygulamalar, onlarca yıldır etrafta. Çoğunlukla 1980'lerin ve 1990'ların anonim e-nakit protokolleri Chaumian körleme olarak bilinen kriptografik bir ilkele dayanan, yüksek dereceli bir para birimi sağladı
mahremiyet, ancak protokoller, merkezileştirilmiş bir
aracı. 1998'de, Wei Dai'nin b-money , yaratma fikrini ortaya koyan ilk teklif oldu.
hesaplamalı bulmacaların yanı sıra merkezi olmayan fikir birliği çözme yoluyla para, ancak teklif merkezi olmayan fikir birliğinin gerçekte nasıl uygulanabileceğine dair ayrıntılar konusunda yetersizdi. 2005 yılında Hal
Finney , b-money fikirlerini kullanan bir sistem olan "yeniden kullanılabilir çalışma kanıtları" kavramını tanıttı.
Adam Back'in hesaplama açısından zor Hashcash bulmacalarıyla birlikte bir
kripto para birimi, ancak bir arka uç olarak güvenilir hesaplamaya güvenerek bir kez daha idealin gerisinde kaldı.
Para birimi, işlem sırasının genellikle kritik olduğu ilk dosyalama uygulaması olduğundan merkezi olmayan para birimleri, merkezi olmayan fikir birliğine bir çözüm gerektirir. Ana barikat Bitcoin öncesi para protokollerinin karşı karşıya olduğu gerçeği,
uzun yıllar boyunca güvenli Bizans-hataya dayanıklı çok taraflı fikir birliği sistemleri oluşturmak, açıklanan protokoller sorunun yalnızca yarısını çözmekteydi. Protokoller, tüm katılımcıların sistem biliniyordu ve "N parti katılırsa,
sistem N / 4 kötü niyetli aktöre tahammül edebilir ". Ancak sorun anonim bir ortamda
bu tür güvenlik marjları, tek bir saldırganın binlerce kişi oluşturduğu sybil saldırılarına karşı savunmasızdır.
bir sunucu veya botnet üzerinde simüle edilmiş düğümler ve bu düğümleri tek taraflı olarak çoğunluk hissesini güvence altına almak için kullanır.
Satoshi tarafından sağlanan yenilik, çok basit bir merkezi olmayan fikir birliğini birleştirme fikridir. protokol, işlemleri her on dakikada bir "blok" halinde birleştiren düğümlere dayalı bir
düğümlerin hak kazandığı bir mekanizma olarak çalışma kanıtı ile sürekli büyüyen blok zinciri sisteme katılın. Büyük miktarda hesaplama gücüne sahip düğümler,
tüm ağdan daha fazla hesaplama gücü ile orantılı olarak daha fazla etki
bir araya getirilmesi, bir milyon düğümü simüle etmekten çok daha zordur. Bitcoin blockchain modeline rağmen
kabalık ve basitlik, yeterince iyi olduğu kanıtlandı ve önümüzdeki beş yıl içinde
dünya çapında iki yüzden fazla para birimi ve protokolün temeli.

5.Sayfa

Durum Geçiş Sistemi Olarak Bitcoin
Teknik bir bakış açısından, Bitcoin defteri, bir durum geçiş sistemi olarak düşünülebilir.
mevcut tüm bitcoinlerin sahiplik durumundan oluşan bir "durum" ve bunu alan bir "durum geçiş işlevi"
bir durum ve bir işlem ve sonuç olan yeni bir durum çıkarır. Standart bir bankacılık sisteminde Örneğin, durum bir bilançodur, bir işlem X $ 'ı A'dan B'ye taşıma talebidir ve durum
geçiş işlevi, A'nın hesabındaki değeri X $ azaltır ve B'nin hesabındaki değeri X $ artırır. Eğer A'lar hesabın ilk etapta X $ 'dan azı varsa durum geçiş işlevi bir hata döndürür. Bu nedenle, biri yapabilir resmi olarak tanımlayın:
UYGULA (S, TX)> S 'veya HATA
Yukarıda tanımlanan bankacılık sisteminde:
UYGULA ({Alice: $ 50, Bob: $ 50}, "Alice'den Bob'a 20 $ gönder") = {Alice: 30 $,

Bob: 70 dolar}
Fakat:
UYGULA ({Alice: $ 50, Bob: $ 50}, "Alice'den Bob'a 70 $ gönder") = HATA

Bitcoin'deki "durum", tüm madeni paraların (teknik olarak "harcanmamış işlem çıktıları" veya UTXO) toplanmasıdır.
her UTXO'nun bir mezhebi ve bir sahibi (bir
Aslında bir kriptografik genel anahtar olan 20 baytlık adres [1]). Bir işlem bir veya daha fazla girdi içerir,
her girdi, mevcut bir UTXO'ya bir referans ve cihaz tarafından üretilen bir kriptografik imza içerir.
sahibin adresiyle ilişkili özel anahtar ve bir veya daha fazla çıktı, her çıktı yeni bir
Eyalete eklenecek UTXO.

Sayfa 6

Durum geçiş işlevi UYGULAMA (S, TX)> S 'kabaca şu şekilde tanımlanabilir:
1.
TX'teki her giriş için:
ben.
Başvurulan UTXO S'de değilse, bir hata döndürün.
ii.
Sağlanan imza UTXO'nun sahibiyle eşleşmiyorsa, bir hata döndürün.
2.
Tüm UTXO girdilerinin değerlerinin toplamı, değerlerinin toplamından azsa
tüm UTXO çıktıları, bir hata döndürür.
3.
Tüm UTXO girişi kaldırılmış ve tüm UTXO çıkışı eklenmiş olarak S döndür.
İlk adımın ilk yarısı, işlem gönderenlerin mevcut olmayan coinleri harcamasını engeller, ikincisi
İlk adımın yarısı, işlem gönderenlerin diğer kişilerin paralarını harcamasını engeller ve ikinci adım
değerin korunmasını zorunlu kılar. Bunu ödeme için kullanmak için protokol aşağıdaki gibidir. Farz edin ki Alice istiyor
Bob'a 11,7 BTC göndermek için. İlk olarak Alice, sahip olduğu, en az toplamda en az bir sayıya ulaşan mevcut UTXO'yu arayacaktır.
11.7 BTC. Gerçekçi olarak, Alice tam olarak 11,7 BTC alamayacak; alabileceği en küçüğünün olduğunu söyle
6 + 4 + 2 = 12. Daha sonra bu üç girdi ve iki çıktıyla bir işlem yaratır. İlk çıktı 11.7 olacak
Bob'un adresini sahibi olarak BTC ve ikinci çıktı, kalan 0,3 BTC "değişim" olacaktır.
sahibi Alice'in kendisidir.
Madencilik
Güvenilir bir merkezi hizmete erişimimiz olsaydı, bu sistemi uygulamak önemsiz olurdu; o
tam olarak açıklandığı gibi kodlanabilir. Bununla birlikte, Bitcoin ile bir
merkezi olmayan para sistemi, bu nedenle devlet geçiş sistemini bir

Herkesin işlemlerin sırasını kabul etmesini sağlamak için fikir birliği sistemi. Bitcoin
Merkezi olmayan fikir birliği süreci, ağdaki düğümlerin sürekli olarak üretmeye çalışmasını gerektirir.
"bloklar" adı verilen işlem paketleri. Ağın her biri kabaca bir blok üretmesi amaçlanmıştır.
on dakika, her blok bir zaman damgası, bir nonce, bir referans (yani hash'i) içerir.

---

7. Sayfa

önceki blok ve önceki bloktan bu yana gerçekleşen tüm işlemlerin bir listesi.
Zamanla bu, sürekli olarak güncellenen kalıcı, sürekli büyüyen bir "blok zinciri" oluşturur.
Bitcoin defterinin en son durumu.
Bu paradigmada ifade edilen, bir bloğun geçerli olup olmadığını kontrol etmek için algoritma
aşağıdaki gibidir:
1.
Bloğun referans verdiği önceki bloğun var olup olmadığını ve geçerli olup olmadığını kontrol edin
2.
Bloğun zaman damgası önceki bloğun daha büyük olup olmadığını kontrol edin [2] ve yaklaşık 2'den daha düşük
geleceğe saatler.
3.
Bloktaki çalışma kanıtının geçerli olup olmadığını kontrol edin.
4.
S [0] bir önceki bloğun sonundaki durum olsun.
5.
TX'in n işlem içeren bloğun işlem listesi olduğunu varsayalım. 0 ... n-1'deki tüm i'ler için, setS [i + 1] =
UYGULA (S [i], TX [i]) Herhangi bir uygulama bir hata verirse, çıkın ve yanlış döndürün.
6.
Doğruya dönün ve S [n] 'yi bu bloğun sonundaki durum olarak kaydedin
Esasen, bloktaki her işlem, geçerli olan bir durum geçişi sağlamalıdır. Eyaletin
herhangi bir şekilde blokta kodlanmamış; doğrulama düğümü tarafından hatırlanması gereken tamamen bir soyutlamadır ve
herhangi bir blok için yalnızca (güvenli bir şekilde), oluşum durumundan başlayarak ve sırayla uygulayarak hesaplanabilir
her bloktaki her işlem. Ek olarak, madencinin işlemleri içerdiği sıranın
blok önemlidir; Bir blokta, B'nin A tarafından oluşturulan bir UTXO harcayacağı şekilde iki A ve B işlemi varsa,
o zaman blok, A B'den önce gelirse, aksi halde geçerli olmazsa geçerli olacaktır.
Blok doğrulama algoritmasının ilginç kısmı, "işin kanıtı" kavramıdır: koşul,
256 bitlik bir sayı olarak değerlendirilen her bloğun SHA256 hash değeri, dinamik olarak ayarlanmış bir hedeften daha az olmalıdır,
bu yazının yazıldığı tarih itibariyle yaklaşık 2.190 . Bunun amacı blok oluşturma yapmaktır
sayısal olarak "zor", böylece sybil saldırganlarının tüm blok zincirini kendi lehlerine yeniden oluşturmasını engelliyor.
SHA256, tamamen öngörülemeyen bir sözde rasgele işlev olarak tasarlandığından,
geçerli bir blok basitçe deneme yanılmadır, nonce'yi tekrar tekrar arttırır ve yeni hash'in eşleşip eşleşmediğine bakar.
Mevcut 2192 hedefinde bu, ortalama 264 deneme anlamına gelir; genel olarak hedef yeniden kalibre edilir.
ağda her 2016 blokta bir ağ, böylece ortalama olarak ağdaki bazı düğümler tarafından her onda bir yeni bir blok üretilir.
dakika. Madencileri bu hesaplama işi için telafi etmek için, her bloğun madencisinin hakkı vardır
kendilerine hiçbir yerden 25 BTC veren bir işlem içerir. Ek olarak, herhangi bir işlemin daha yüksek bir

girdilerindeki toplam değer, çıktılarından ziyade, fark aynı zamanda madenciye bir "işlem" olarak da gider.
Bu arada, bu aynı zamanda BTC'nin yayınlandığı tek mekanizmadır; oluşum durumu hiçbir
hiç bozuk para.

---

8. Sayfa

Madenciliğin amacını daha iyi anlamak için, kötü niyetli bir olay durumunda neler olduğunu inceleyelim.
saldırgan. Bitcoin'in temelindeki kriptografinin güvenli olduğu bilindiğinden, saldırgan,
Doğrudan kriptografi ile korunmayan Bitcoin sistemi: işlemlerin sırası. Saldırganın
strateji basit:
1.
Bazı ürünler karşılığında bir satıcıya 100 BTC gönderin (tercihen hızlı teslimat dijital
iyi)
2.
Ürünün teslimatını bekleyin
3.
Aynı 100 BTC'yi kendisine gönderen başka bir işlem gerçekleştirin
4.
Ağı, kendisine yaptığı işlemin ilk gelen işlem olduğuna ikna etmeye çalışın.
Adım (1) gerçekleştikten sonra, birkaç dakika sonra bazı madenciler işlemi bir bloğa dahil edecek.
270000 numaralı blok. Yaklaşık bir saat sonra, bu bloktan sonra zincire beş blok daha eklenecektir,
bu blokların her biri dolaylı olarak işleme işaret eder ve böylece işlemi "onaylar". Bu noktada
tüccar ödemeyi kesinleşmiş olarak kabul edecek ve ürünü teslim edecektir; çünkü bunun dijital olduğunu varsayıyoruz
iyi, teslimat anında. Şimdi, saldırgan 100 BTC'yi kendisine gönderen başka bir işlem oluşturur. Eğer saldırgan bunu vahşi ortamda serbest bırakır, işlem işlenmez; madenciler koşmaya çalışacak
UYGULA (S, TX) ve TX'in artık durumda olmayan bir UTXO kullandığını fark edin. Bunun yerine saldırgan
Blok zincirinin bir "çatalını" oluşturur, aynı şeyi gösteren 270000 bloğunun başka bir sürümünü madenciliği yaparak başlayarak
269999 bloğunu bir üst öğe olarak, ancak eskisinin yerine yeni işlemle. Çünkü blok verileri
farklı, bu çalışma kanıtının yeniden yapılmasını gerektirir. Ayrıca, saldırganın 270000 bloğunun yeni sürümünde bir
farklı hash, bu nedenle 270001 ila 270005 arasındaki orijinal bloklar ona "işaret etmez"; böylece, orijinal zincir ve
saldırganın yeni zinciri tamamen ayrıdır. Kural, bir çatalda en uzun blok zincirinin (yani bir
en büyük miktarda iş kanıtı ile desteklenir) gerçek olarak kabul edilir ve bu nedenle meşru madenciler üzerinde çalışacaktır.
270005 zinciri, yalnızca saldırgan 270000 zinciri üzerinde çalışırken. Saldırganın yapması için
blok zinciri en uzun olanıdır, ağın geri kalanından daha fazla hesaplama gücüne sahip olması gerekir.
yetişmek için birleştirilir (dolayısıyla, "% 51 saldırı").

Sayfa 9

Merkle Ağaçları
Sol: Bir dalın geçerliliğini kanıtlamak için bir Merkle ağacında yalnızca az sayıda düğüm sunmak yeterlidir.
Doğru: Merkle ağacının herhangi bir bölümünü değiştirmeye yönelik herhangi bir girişim, eninde sonunda bir yerlerde bir tutarsızlığa yol açacaktır.
zincir .

Bitcoin'in önemli bir ölçeklenebilirlik özelliği, bloğun çok seviyeli bir veri yapısında saklanmasıdır. "Hash"
bir bloğun yalnızca blok başlığının karmasıdır, kabaca 200 baytlık bir veri parçası
timestamp, nonce, önceki blok hash'i ve hepsini depolayan Merkle ağacı adı verilen bir veri yapısının kök hash'i
bloktaki işlemler.
Merkle ağacı, çok sayıda yaprak düğümü bulunan bir dizi düğümden oluşan bir tür ikili ağaçtır.
temel verileri içeren ağacın altında, her düğümün karma değeri olduğu bir dizi ara düğüm
iki çocuğu ve son olarak da iki çocuğunun karmasından oluşan tek bir kök düğüm
Ağacın "tepesi". Merkle ağacının amacı, bir bloktaki verilerin parça parça teslim edilmesine izin vermektir: bir düğüm
bir kaynaktan bir bloğun yalnızca başlığını, ağacın kendileriyle ilgili küçük kısmını ise

---

Sayfa 10

başka bir kaynak ve yine de tüm verilerin doğru olduğundan emin olun. Bunun işe yaramasının nedeni, hash'lerin
yukarı doğru yayılma: Kötü niyetli bir kullanıcı sahte bir işlemi Merkle ağacının altına takas etmeye çalışırsa,
bu değişiklik, yukarıdaki düğümde bir değişikliğe ve ardından bunun üstündeki düğümde bir değişikliğe neden olacaktır.
ağacın kökünü ve dolayısıyla bloğun karmasını değiştirmek, protokolün onu bir
tamamen farklı bir blok (neredeyse kesinlikle geçersiz bir çalışma kanıtı ile).
Merkle ağacı protokolü muhtemelen uzun vadeli sürdürülebilirlik için gereklidir. Bitcoin ağında bir "tam düğüm",
Her bloğun tamamını depolayan ve işleyen, Bitcoin'de yaklaşık 15 GB disk alanı kaplayan
ağ Nisan 2014 itibariyle ve ayda bir gigabayttan fazla büyüyor. Şu anda bu bazı masaüstü bilgisayarlar için uygundur
telefonlar değil bilgisayarlar ve daha sonra gelecekte sadece işletmeler ve hobiler katılabilecek.
"Basitleştirilmiş ödeme doğrulaması" (SPV) olarak bilinen bir protokol, başka bir düğüm sınıfının var olmasına izin verir.
Blok başlıklarını indiren "hafif düğümler", blok başlıkları üzerindeki çalışmanın kanıtını doğrular ve ardından
yalnızca kendileriyle ilgili işlemlerle ilişkili "şubeleri" indirin. Bu hafif düğümlere izin verir
Güçlü bir güvenlik garantisi ile herhangi bir Bitcoin işleminin durumunu ve mevcut durumlarını belirlemek için
denge, tüm blok zincirinin yalnızca çok küçük bir bölümünü indirirken.

Alternatif Blockchain Uygulamaları

Altta yatan blockchain fikrini alıp diğer kavramlara uygulama fikrinin de uzun bir geçmişi var. İçinde 2005, Nick Szabo bir belge olan " mal sahibi yetkisine sahip güvenli mülkiyet hakları " konseptiyle ortaya çıktı.
"çoğaltılmış veritabanı teknolojisindeki yeni ilerlemelerin" blockchain tabanlı bir sisteme nasıl izin vereceğini açıklayan
kimin hangi araziye sahip olduğuna dair bir kayıt depolamak, aşağıdaki gibi kavramları içeren ayrıntılı bir çerçeve oluşturmak
evde kalma, mülkiyet hakkı ve Gürcü arazi vergisi. Ancak maalesef etkili olmadı
o sırada mevcut olan çoğaltılmış veritabanı sistemi ve bu nedenle protokol pratikte hiçbir zaman uygulanmadı.
Ancak 2009'dan sonra, Bitcoin'in merkezi olmayan fikir birliği geliştirildikten sonra bir dizi alternatif uygulamalar hızla ortaya çıkmaya başladı:

- **Namecoin** - 2010 yılında yaratılan Namecoin , en iyi şekilde merkezi olmayan bir isim kaydı olarak tanımlanır.
veri tabanı. Tor, Bitcoin ve BitMessage gibi merkezi olmayan protokollerde bir yol olması gerekir

diğer kişilerin onlarla etkileşim kurabilmesi için hesapları tanımlama, ancak mevcut tüm çözümlerde sadece
tür
nın-nin
tanımlayıcı
mevcut
dır-dir
a
sözde rasgele
karma
like1LW79wp5ZBqaHW1jL5TCiBCrhQYtHagUWy. İdeal olarak, kişi bir
"george" gibi bir adı olan hesap. Ancak sorun şu ki, bir kişi bir
"george" adlı bir hesap varsa, başka biri aynı işlemi "george" için kayıt yaptırmak için kullanabilir.
kendileri de ve onları taklit ediyorlar. Tek çözüm dosyaya ilk paradigmadır.
ilk tescil ettiren başarılı olur ve ikincisi başarısız olur - Bitcoin mutabakatına mükemmel şekilde uyan bir problem
protokol. Namecoin, bir ad kaydının en eski ve en başarılı uygulamasıdır
sistem böyle bir fikir kullanarak.

- **Renkli madeni paralar** - renkli madeni paraların amacı, insanların kendi
kendi dijital para birimleri - veya bir para biriminin önemli önemsiz durumunda, dijital jetonlar,

---

Sayfa 11

Bitcoin blok zincirinde. Renkli madeni paralar protokolünde, bir kişi yeni bir para birimini halka açık bir şekilde
belirli bir Bitcoin UTXO'ya bir renk atamak ve protokol diğerinin rengini özyinelemeli olarak tanımlar.
UTXO, onları oluşturan işlemin harcadığı girdilerin rengiyle aynı olacaktır (bazıları
karışık renkli girişler için özel kurallar geçerlidir). Bu, kullanıcıların cüzdanları korumasına izin verir
yalnızca belirli bir renkte UTXO içerir ve bunları normal bitcoinler gibi etrafa gönderir,
aldıkları herhangi bir UTXO'nun rengini belirlemek için blok zincirinde geriye doğru izleme.

- **Metacoinler** - bir metacoin'in arkasındaki fikir, Bitcoin kullanarak, Bitcoin'in üstünde yaşayan bir protokole sahip olmaktır.
Metacoin işlemlerini depolamak için ancak farklı bir durum geçiş işlevine sahip Bitcoin işlemleri,
UYGULAMAK'. Metacoin protokolü, geçersiz metacoin işlemlerini
Bitcoin blok zincirinde göründüğünde, UYGULA '(S, TX) bir hata döndürürse,
protokol varsayılan olarak APPLY '(S, TX) = S şeklindedir. Bu, isteğe bağlı bir program oluşturmak için kolay bir mekanizma sağlar.
kripto para protokolü, potansiyel olarak içinde uygulanamayan gelişmiş özelliklerle
Bitcoin'in kendisi, ancak madencilik ve ağ oluşturmanın karmaşıklığından bu yana çok düşük bir geliştirme maliyetiyle
zaten Bitcoin protokolü tarafından işleniyor.
Bu nedenle, genel olarak, bir fikir birliği protokolü oluşturmaya yönelik iki yaklaşım vardır: bağımsız bir
ağ ve Bitcoin üzerine bir protokol oluşturma. Eski yaklaşım,
Namecoin gibi uygulamalarda uygulanması zordur; her bir uygulamanın ihtiyacı
bağımsız bir blok zincirini önyükleme, ayrıca gerekli tüm durum geçişlerini oluşturma ve test etme ve ağ kodu. Ek olarak, merkezi olmayan fikir birliği teknolojisi için uygulama setinin
Uygulamaların büyük çoğunluğunun, bunların garanti altına alınamayacak kadar küçük olduğu bir güç yasası dağılımını
kendi blok zincirine sahibiz ve özellikle merkezi olmayan uygulamaların büyük sınıfları olduğunu not ediyoruz.

birbirleriyle etkileşime girmesi gereken merkezi olmayan özerk kuruluşlar.
Öte yandan Bitcoin tabanlı yaklaşım, basitleştirilmiş ödemeyi miras almama kusuruna sahiptir.
Bitcoin'in doğrulama özellikleri. SPV, Bitcoin için çalışır çünkü blockchain derinliğini proxy olarak kullanabilir.
geçerlilik; bir noktada, bir işlemin ataları yeterince geriye gittikten sonra,
devletin meşru bir parçası. Öte yandan, blok zinciri tabanlı meta protokoller, blok zincirini zorlayamaz.
kendi protokolleri kapsamında geçerli olmayan işlemleri dahil etmemek. Bu nedenle, tamamen güvenli
SPV meta protokol uygulamasının Bitcoin'in başlangıcına kadar geriye doğru taraması gerekir.
belirli işlemlerin geçerli olup olmadığını belirlemek için blockchain. Şu anda, tüm "hafif" uygulamaları
Bitcoin tabanlı meta protokoller, verileri sağlamak için güvenilir bir sunucuya güveniyor, muhtemelen oldukça yetersiz bir sonuç
özellikle bir kripto para biriminin temel amaçlarından biri güven ihtiyacını ortadan kaldırmak olduğunda.

Komut dosyası oluşturma

Herhangi bir uzantı olmasa bile, Bitcoin protokolü aslında "akıllı" kavramının zayıf bir versiyonunu kolaylaştırır.
Bitcoin'deki UTXO sadece bir genel anahtara değil, aynı zamanda daha karmaşık bir komut dosyasına da sahip olabilir.
basit bir yığın tabanlı programlama dilinde ifade edilir. Bu paradigmada, bunu harcayan bir işlem UTXO, komut dosyasını karşılayan verileri sağlamalıdır. Aslında, temel açık anahtar sahiplik mekanizması bile

---

Sayfa 12

bir komut dosyası aracılığıyla uygulanır: komut dosyası, girdi olarak eliptik bir eğri imzası alır, bunu işlem ve UTXO'ya sahip olan adres ve doğrulama başarılıysa 1 ve 0 değerini döndürür
aksi takdirde. Diğer, daha karmaşık komut dosyaları, çeşitli ek kullanım durumları için mevcuttur. Örneğin, biri
Doğrulamak için verilen üç özel anahtardan ikisinden imza gerektiren bir komut dosyası oluşturun ("multisig"),
kurumsal hesaplar, güvenli tasarruf hesapları ve bazı tüccar emanet durumları için kullanışlı bir kurulum. Kodlar
aynı zamanda hesaplama problemlerinin çözümü için ikramiye ödemek için de kullanılabilir ve hatta bir komut dosyası bile oluşturulabilir
"Bu Bitcoin UTXO, bir Dogecoin gönderdiğinize dair bir SPV kanıtı sağlayabilirseniz
bu değerin bana aktarımı ", esasen merkezi olmayan çapraz kripto para birimi değişimine izin veriyor.
Bununla birlikte, Bitcoin'de uygulanan komut dosyası dilinin birkaç önemli sınırlaması vardır:

- **Turing tamlığı eksikliği** - yani büyük bir hesaplama alt kümesi varken
Bitcoin betik dilinin desteklediği, neredeyse her şeyi desteklemiyor. Ana
eksik olan kategori döngülerdir. Bu işlem doğrulama sırasında sonsuz döngülerden kaçınmak için yapılır;
teorik olarak, herhangi bir döngü simüle edilebildiğinden, komut dosyası programcıları için aşılması gereken bir engeldir.
temeldeki kodu bir if ifadesiyle birçok kez tekrarlayarak, ancak komut dosyalarına yol açar
bu alan çok verimsiz. Örneğin, alternatif bir eliptik eğri imzası uygulamak
algoritma muhtemelen 256 tekrarlanan çarpma turu gerektirecektir.
kodu.
- **Değer körlüğü** - bir UTXO betiğinin,
çekilebilecek miktar. Örneğin, bir oracle sözleşmesinin güçlü kullanım durumlarından biri,

A ve B'nin 1000 $ değerinde BTC koyduğu ve 30 gün sonra betiğin 1000 $ gönderdiği riskten korunma sözleşmesi
BTC'nin değeri A'ya ve geri kalanı B'ye. Bu, 1 BTC'nin değerini belirlemek için bir oracle gerektirecektir.
USD, ancak o zaman bile güven ve altyapı gereksinimi açısından büyük bir gelişme.
şu anda mevcut olan tamamen merkezi çözümler. Ancak, UTXO ya hep ya hiç olduğu için,
bunu başarmanın tek yolu, değişen birçok UTXO'ya sahip olmanın çok verimsiz bir şekilde kesilmesidir.
adlandırmalar (örn., bir UTXO 2 k 30'a kadar her k) ve Oracle çekme sahip olan UTXO için
A'ya ve hangisini B'ye gönder.

- **Devlet eksikliği** - UTXO harcanabilir veya harcanmayabilir; çok aşamalı fırsat yok
bunun ötesinde herhangi bir diğer iç durumu tutan sözleşmeler veya komut dosyaları. Bu, yapmayı zorlaştırır
çok aşamalı opsiyon sözleşmeleri, merkezi olmayan değişim teklifleri veya iki aşamalı kriptografik taahhüt
protokoller (güvenli hesaplama ödülleri için gereklidir). Ayrıca UTXO'nun yalnızca
basit, bir kereye mahsus sözleşmeler oluşturmak ve merkezi olmayanlar gibi daha karmaşık "duruma bağlı" sözleşmeler yapmak
organizasyonlar ve meta protokollerin uygulanmasını zorlaştırır. İkili durum birleştirilmiş
değer körlüğü aynı zamanda başka bir önemli uygulamanın, para çekme limitlerinin imkansız olduğu anlamına gelir.
- **Blockchain körlüğü** - UTXO, nonce ve önceki gibi blok zinciri verilerine kördür.
karma. Bu, kumar ve diğer bazı kategorilerdeki uygulamaları ciddi şekilde sınırlar.
potansiyel olarak değerli bir rastgelelik kaynağının komut dosyası dili.

Sayfa 13

Bu nedenle, kripto para biriminin yanı sıra gelişmiş uygulamalar oluşturmak için üç yaklaşım görüyoruz: yeni bir
Bitcoin üzerinde komut dosyası kullanarak ve Bitcoin'in üstünde bir meta-protokol oluşturan blok zinciri. Yeni inşa etmek
Blockchain, bir özellik seti oluşturmada sınırsız özgürlüğe izin verir, ancak geliştirme süresi ve maliyeti
önyükleme çabası. Komut dosyası kullanmak, uygulamak ve standartlaştırmak kolaydır, ancak
yetenekler ve meta protokoller, kolay olsa da, ölçeklenebilirlikteki hatalardan muzdariptir. Ethereum ile inşa etmeyi planlıyoruz
Her üç paradigmanın avantajlarını aynı anda sağlayabilen genelleştirilmiş bir çerçeve.

Ethereum

Ethereum'un amacı, komut dosyası oluşturma, altcoin ve zincir içi kavramları bir araya getirmek ve geliştirmektir.
meta protokoller ve geliştiricilerin, aşağıdaki özelliklere sahip keyfi fikir birliğine dayalı uygulamalar oluşturmasına olanak tanır.
bunların sunduğu ölçeklenebilirlik, standardizasyon, özellik tamlığı, geliştirme kolaylığı ve birlikte çalışabilirlik
aynı anda farklı paradigmalar. Ethereum bunu, esasen nihai olanı inşa ederek yapar.
soyut temel katman: yerleşik bir Turing-complete programlama diline sahip bir blok zinciri,
kendi keyfi olarak oluşturabilecekleri akıllı sözleşmeler ve merkezi olmayan uygulamalar yazan herkes
sahiplik kuralları, işlem biçimleri ve durum geçiş işlevleri. Namecoin'in çıplak kemik versiyonu olabilir
iki satır kodla yazılabilir ve para birimleri ve itibar sistemleri gibi diğer protokoller,

yirmi. Değer içeren ve yalnızca belirli koşullar geçerliyse kilidini açan akıllı sözleşmeler, kriptografik "kutular"
bir araya geldi, ayrıca Bitcoin betiklerinin sunduğundan çok daha fazla güçle platformumuzun üzerine inşa edilebilir.
Turing-bütünlüğünün, değer farkındalığının, blok zinciri farkındalığının ve devletin eklenen güçleri nedeniyle.

Ethereum Hesapları

Ethereum'da durum, her hesabın 20 baytlık bir adrese sahip olduğu "hesaplar" adı verilen nesnelerden oluşur.
ve durum geçişleri hesaplar arasında doğrudan değer ve bilgi aktarımıdır. Bir Ethereum hesabı
dört alan içerir:

- **Nonce** , bir karşı emin, her bir işlem sadece bir kez işlenebilir hale getirmek için kullanılan
- Hesabın mevcut **eter bakiyesi**
- Varsa, hesabın **sözleşme kodu**
- Hesabın **deposu** (varsayılan olarak boştur)

"Eter", Ethereum'un ana dahili kripto yakıtıdır ve işlem ücretlerini ödemek için kullanılır. Genel olarak var
iki tür hesap: harici olarak sahip olunan, özel anahtarlarla kontrol edilen hesaplar ve kontrol edilen sözleşmeli hesaplar
sözleşme kodlarına göre. Dışarıdan sahip olunan bir hesabın kodu yoktur ve bir kişi bir
bir işlem oluşturarak ve imzalayarak dışarıdan sahip olunan hesap; bir sözleşme hesabında, her seferinde

---

Sayfa 14

sözleşme hesabı, kodunun etkinleştirdiği bir mesajı alarak dahili depolamayı okuyup yazmasına izin verir ve
sırayla başka mesajlar gönderin veya sözleşmeler oluşturun.

Mesajlar ve İşlemler

Ethereum'daki "Mesajlar", Bitcoin'deki "işlemlere" biraz benzer, ancak üç önemli
farklılıklar. İlk olarak, bir Ethereum mesajı harici bir varlık veya bir sözleşme tarafından oluşturulabilirken,
Bitcoin işlemi yalnızca harici olarak oluşturulabilir. İkincisi, Ethereum mesajları için açık bir seçenek var
veri içermek için. Son olarak, bir Ethereum mesajının alıcısı, eğer bir sözleşme hesabı ise,
bir yanıt döndürmek; bu, Ethereum mesajlarının işlevler kavramını da kapsadığı anlamına gelir.
"İşlem" terimi, Ethereum'da bir mesajın saklandığı imzalı veri paketini ifade etmek için kullanılır.
dışarıdan sahip olunan bir hesaptan gönderilmiştir. İşlemler mesajın alıcısını, bir imzayı içerir
gönderenin, eter miktarının ve gönderilecek verilerin yanı sıra STARTGAS adlı iki değerin belirlenmesi ve
GAZ FİYATI. Kodda üstel patlamayı ve sonsuz döngüleri önlemek için, her işlemin
Her iki ilk mesaj da dahil olmak üzere, kod yürütmenin kaç hesaplama adımını ortaya çıkarabileceğine dair bir sınır
ve yürütme sırasında ortaya çıkan ek mesajlar. STARTGAS bu sınırdır ve GASPRICE
hesaplama adımı başına madenciye ödenecek ücret. İşlem yürütme "benzini biterse", tüm durum değişir
geri alma - ücretlerin ödenmesi dışında ve işlemin bir miktar gaz kalmasıyla durması halinde
ücretlerin kalan kısmı gönderene iade edilir. Ayrı bir işlem türü de vardır ve
bir sözleşme oluşturmak için ilgili mesaj türü; bir sözleşmenin adresi aşağıdakilere göre hesaplanır:
hesap nonce ve işlem verilerinin hash değeri.

Mesaj mekanizmasının önemli bir sonucu, Ethereum'un "birinci sınıf vatandaş" mülküdür -
mesaj gönderme yeteneği de dahil olmak üzere sözleşmelerin harici hesaplara eşdeğer yetkilere sahip olduğu fikri ve
başka sözleşmeler oluşturun. Bu, sözleşmelerin aynı anda birçok farklı role hizmet etmesini sağlar: örneğin,
merkezi olmayan bir kuruluşun üyesi olabilir (bir sözleşme) emanet hesabı olabilir (başka bir sözleşme)
özel kuantuma dayanıklı Lamport imzaları kullanan paranoyak bir birey (üçüncü bir sözleşme) arasında ve
güvenlik için beş anahtara sahip bir hesap kullanan bir birlikte imzalayan varlık (dördüncü bir sözleşme). Gücü
Ethereum platformu, merkezi olmayan organizasyonun ve emanet sözleşmesinin önemsemesine gerek olmamasıdır.
sözleşmenin her bir tarafının ne tür bir hesap olduğu hakkında.

Sayfa 15

Ethereum Durum Geçiş İşlevi
Ethereum durum geçiş işlevi, UYGULA (S, TX) -> S 'aşağıdaki gibi tanımlanabilir:
1.
İşlemin doğru yapılıp yapılmadığını (yani doğru sayıda değere sahip olup olmadığını), imzanın geçerli olup olmadığını kontrol edin,
ve nonce, gönderenin hesabındaki nonce ile eşleşir. Değilse, bir hata döndürün.
2.
İşlem ücretini STARTGAS * GASPRICE olarak hesaplayın ve gönderim adresini buradan belirleyin.
imza. Ücreti gönderenin hesap bakiyesinden çıkarın ve gönderenin hesap bakiyesini artırın.
nonce. Harcamak için yeterli bakiye yoksa bir hata döndürün.
3.
GAS = STARTGAS'ı başlatın ve bayt başına ödeme yapmak için bayt başına belirli bir miktarda gazı çıkarın.
işlem.
4.
İşlem değerini gönderenin hesabından alıcı hesaba aktarın. Eğer alıcı
hesap henüz mevcut değil, oluşturun. Alıcı hesap bir sözleşmeyse, sözleşmenin kodunu çalıştırın
ya tamamlanana kadar ya da infazın gazı bitene kadar.
5.
Gönderenin yeterli parası olmadığı için değer aktarımı başarısız olduysa veya kod yürütme
Benzin bitti, ücretlerin ödenmesi dışındaki tüm durum değişikliklerini geri alın ve ücretleri
madenci hesabı.
6.
Aksi takdirde, kalan tüm gazın ücretlerini göndericiye iade edin ve gaz için ödenen ücretleri gönderin
madenci için tüketildi.

Sayfa 16

Örneğin, sözleşmenin kodunun şu olduğunu varsayalım:
if! contract.storage [msg.data [0]]:
Contract.storage [msg.data [0]] = msg.data [1]
Gerçekte sözleşme kodunun düşük seviyeli EVM kodunda yazıldığını unutmayın; bu örnek Serpent'te yazılmıştır,
yüksek seviyeli dilimiz, netlik için ve EVM koduna göre derlenebilir. Sözleşmenin
depolama boş başlar ve 10 eter değeri, 2000 gas, 0.001 eter gasprice ile bir işlem gönderilir ve

iki veri alanı: [2, 'CHARLIE'] [3] . Bu durumda durum geçiş işlevi için süreç şu şekildedir:
1.
İşlemin geçerli ve doğru yapılıp yapılmadığını kontrol edin.
2.
İşlem göndereninin en az 2000 * 0,001 = 2 etere sahip olup olmadığını kontrol edin. Öyleyse, 2 eteri çıkarın
gönderenin hesabından.
3.
Gazı başlat = 2000; işlemin 170 bayt uzunluğunda ve bayt ücretinin 5 olduğunu varsayarak çıkarın 850 yani 1150 gaz kaldı.
4.
Gönderenin hesabından 10 eter daha çıkarın ve sözleşmenin hesabına ekleyin.
5.
Kodu çalıştırın. Bu durumda, bu basittir: sözleşmenin endeks 2'deki depolamasının kullanılıp kullanılmadığını kontrol eder,
öyle olmadığını fark eder ve bu nedenle dizin 2'deki depolamayı CHARLIE değerine ayarlar. Bunun sürdüğünü varsayalım
187 gaz, yani kalan gaz miktarı 1150 - 187 = 963
6.
Gönderenin hesabına 963 * 0.001 = 0.963 ether ekleyin ve ortaya çıkan durumu geri getirin.
İşlemin alıcı tarafında herhangi bir sözleşme yoksa, toplam işlem ücreti basitçe
sağlanan GASPRICE ile işlemin bayt cinsinden uzunluğunun çarpımına eşittir ve gönderilen veriler işlemin yanında alakasız olacaktır. Ek olarak, sözleşmeyle başlatılan mesajların atayabileceğini unutmayın.
ortaya çıkardıkları hesaplama için bir gaz limiti ve alt hesaplamanın gazı biterse, geri döndürülür.
sadece mesaj araması noktasına. Dolayısıyla, işlemler gibi, sözleşmeler de sınırlı
hesaplama kaynakları oluşturdukları alt hesaplamalara katı sınırlar koyarak.

Sayfa 17

Kod Yürütme

Ethereum sözleşmelerindeki kod, düşük seviyeli, yığın tabanlı bir bayt kodu dilinde yazılmıştır. "Ethereum sanal makine kodu" veya "EVM kodu". Kod bir dizi bayttan oluşur ve burada her bayt bir işlemi temsil eder. Genel olarak, kod yürütme, tekrar tekrar gerçekleştirmekten oluşan sonsuz bir döngüdür.
mevcut program sayacındaki işlem (sıfırdan başlar) ve ardından programı artırma
kodun sonuna ulaşılıncaya veya bir hata veya STOP veya RETURN talimatı tespit edilene kadar sayacı birer birer.
işlemlerin, verilerin depolanacağı üç tür alana erişimi vardır:

- **Yığın** , bir son giren ilk çıkar kap 32 bayt değerleri itilir ve attı edilebildiği
- **Bellek** , sonsuz genişletilebilir bir bayt dizisi
- Sözleşmenin uzun vadeli **deposu** , anahtarların ve değerlerin her ikisinin de 32 olduğu bir anahtar / değer deposu

bayt. Hesaplama bittikten sonra sıfırlanan yığın ve belleğin aksine, depolama uzun süre devam eder terim.
Kod ayrıca gelen mesajın değerine, gönderenine ve verilerine erişebilir ve başlık verilerini engelleyebilir,
ve kod ayrıca çıktı olarak bir bayt veri dizisi döndürebilir.

EVM kodunun resmi yürütme modeli şaşırtıcı derecede basittir. Ethereum sanal makinesi,
çalışan, tam hesaplama durumu tuple (block_state, transaction, message, code,
bellek, yığın, bilgisayar, gaz), burada block_state tüm hesapları içeren ve bakiyeleri içeren küresel durumdur
ve depolama. Her yürütme turunda, mevcut talimat pc-inci kod baytı alınarak bulunur ve
her komutun, demeti nasıl etkilediğine ilişkin kendi tanımı vardır. Örneğin, ADD iki öğeyi çıkarır
yığın ve toplamlarını iter, gazı 1 azaltır ve pc'yi 1 artırır ve SSTORE ilk ikisini iter
yığından çıkarır ve ikinci kalemi sözleşmenin deposuna birinci tarafından belirtilen dizinde ekler
ve aynı zamanda gazı 200'e kadar düşürüp pc'yi 1 artırıyor.
Tam zamanında derleme yoluyla Ethereum'u optimize edin, Ethereum'un temel bir uygulaması birkaç adımda yapılabilir
yüz satır kod.

Sayfa 18

Blockchain ve Madencilik
Ethereum blok zinciri, birçok yönden Bitcoin blok zincirine benzer, ancak bazılarına sahip
farklılıklar. Blockchain mimarisi ile ilgili olarak Ethereum ve Bitcoin arasındaki temel fark,
Bitcoin'den farklı olarak, Ethereum blokları hem işlem listesinin hem de en son durumun bir kopyasını içerir.
Bunun yanı sıra, diğer iki değer, blok numarası ve zorluk da blokta saklanır. Blok
Ethereum'daki doğrulama algoritması aşağıdaki gibidir:
1.
Referans verilen önceki bloğun var olup olmadığını ve geçerli olup olmadığını kontrol edin.
2.
Bloğun zaman damgasının referans verilen önceki bloktan daha büyük olup olmadığını kontrol edin ve geleceğe 15 dakikadan az kaldı
3.
Blok numarası, zorluk, işlem kökü, amca kökü ve gaz limitinin (çeşitli düşük seviyeli
Ethereum'a özgü kavramlar) geçerlidir.
4.
Bloktaki çalışma kanıtının geçerli olup olmadığını kontrol edin.
5.
S [0], önceki bloğun STATE_ROOT'u olsun.
6.
TX, n işlem içeren bloğun işlem listesi olsun. 0 ... n-1 içindeki tümü için, setS [i + 1] =
UYGULA (S [i], TX [i]). Herhangi bir uygulama bir hata verirse veya blokta tüketilen toplam gaz
bu nokta GASLIMIT'i aşana kadar bir hata döndür.
7.
S_FINAL'in S [n] olmasına izin verin, ancak madenciye ödenen blok ödülünü ekleyin.
8.
S_FINAL'in STATE_ROOT ile aynı olup olmadığını kontrol edin. Öyleyse, blok geçerlidir; aksi takdirde geçerli değildir.

Sayfa 19

Yaklaşım ilk bakışta oldukça verimsiz görünebilir, çünkü tüm durumu her biriyle birlikte depolaması gerekir.
blok, ancak gerçekte verimlilik Bitcoin ile karşılaştırılabilir olmalıdır. Nedeni, devletin depolanmasıdır.
ağaç yapısı ve her bloktan sonra ağacın sadece küçük bir kısmının değiştirilmesi gerekir. Bu nedenle, genel olarak,

iki bitişik blok arasında ağacın büyük çoğunluğu aynı olmalıdır ve bu nedenle veriler
bir kez depolanır ve işaretçiler kullanılarak iki kez başvurulur (yani, alt ağaçların karmaları). Olarak bilinen özel bir ağaç türü
"Patricia ağacı", bunu başarmak için kullanılır; Merkle ağacı konseptinde yapılan bir değişiklik de dahil olmak üzere,
etkin bir şekilde değiştirilmekle kalmayıp, eklenecek ve silinecek düğümler. Ek olarak, çünkü tüm devlet
bilgi son bloğun bir parçasıdır, tüm blok zinciri geçmişini saklamaya gerek yoktur - eğer
Bitcoin'e uygulanabilir, mekanda 5-20 kat tasarruf sağlayacak şekilde hesaplanabilir.

Başvurular

Genel olarak, Ethereum'un üzerinde üç tür uygulama vardır. İlk kategori finansal uygulamalardır, kullanıcılara paralarını kullanarak sözleşmeleri yönetmek ve imzalamak için daha güçlü yollar sağlamak. Bu
alt para birimlerini, finansal türevleri, riskten korunma sözleşmelerini, tasarruf cüzdanlarını, vasiyetnameleri ve nihayetinde
tam ölçekli iş sözleşmelerinin bazı sınıfları. İkinci kategori yarı finansal uygulamalardır.
işin içinde para var ama aynı zamanda yapılanın parasal olmayan ağır bir yanı da var; mükemmel bir örnek
hesaplama sorunlarına çözümler için kendi kendini uygulayan ödüller. Son olarak çevrimiçi gibi uygulamalar var
hiçbir şekilde finansal olmayan oylama ve ademi merkeziyetçi yönetişim.

Token Sistemleri

On-blockchain token sistemleri, bu tür varlıkları temsil eden alt para birimlerinden değişen birçok uygulamaya sahiptir.
Şirket hisse senetlerine USD veya altın olarak, akıllı mülkiyeti temsil eden bireysel belirteçler, takas edilemez kuponlar elde etme
ve hatta geleneksel değerle hiçbir bağı olmayan, teşvik için puan sistemleri olarak kullanılan jeton sistemleri.
Token sistemlerinin Ethereum'da uygulanması şaşırtıcı derecede kolaydır. Anlaşılması gereken en önemli nokta, tümünün
para birimi veya belirteç sistemi, temelde tek işlemli bir veritabanıdır: A'dan X birimini çıkarın ve
(İ) X'in işlemden önce en az X birime sahip olması ve (2) işlemin
A. tarafından onaylandı. Bir token sistemini uygulamak için gereken tek şey, bu mantığı bir sözleşmeye uygulamaktır.

Sayfa 20

Serpent'te bir token sistemi uygulamak için temel kod aşağıdaki gibidir:

```
gönderen = msg.sender
to = msg.data [0]
değer = msg.data [1]

Contract.storage [from]> = değer ise:
    Contract.storage [from] = contract.storage [from] değeri
    Contract.storage [to] = contract.storage [to] + value
```

Bu, esasen, daha ayrıntılı olarak açıklanan "bankacılık sistemi" durum geçiş işlevinin gerçek bir uygulamasıdır.
bu belgede yukarıda. Dağıtımın ilk adımını sağlamak için birkaç ekstra kod satırı eklenmesi gerekir
ilk etapta para birimleri ve birkaç diğer uç durum ve ideal olarak, izin vermek için bir işlev eklenecektir.
diğer sözleşmeler bir adresin bakiyesini sorgular. Ama hepsi bu kadar. Teorik olarak, Ethereum tabanlı
Alt para birimleri olarak hareket eden token sistemleri, potansiyel olarak zincir üzerindeki başka bir
önemli özelliği içerebilir.

Bitcoin tabanlı meta para birimleri eksiktir: işlem ücretlerini doğrudan o para biriminde ödeme yeteneği. Bu şekilde
uygulanacaktı, sözleşmenin eter bakiyesini geri ödeyeceği bir eter bakiyesini koruyacaktı.
gönderene ücret ödüyordu ve bu bakiyeyi kendi para birimlerini toplayarak yeniden dolduruyordu.
ücretleri alır ve sürekli çalışan bir açık artırmada yeniden satar. Bu nedenle, kullanıcıların kendi web sitelerini "etkinleştirmeleri" gerekir.
Ether ile hesaplar, ancak eter bir kez oradayken yeniden kullanılabilir çünkü sözleşme her birini geri ödeyecektir.
zaman.

Finansal türevler ve Sabit Değerli Para Birimleri

Finansal türevler, bir "akıllı sözleşmenin" en yaygın uygulamasıdır ve en basitlerinden biridir.
kodda uygulamak. Finansal sözleşmelerin uygulanmasındaki temel zorluk, bunların çoğunun
harici bir fiyat şeridine referans verilmesini gerektirir; örneğin, çok istenen bir uygulama, akıllı bir sözleşmedir.
ABD doları karşısında eterin (veya başka bir kripto para biriminin) oynaklığına karşı koruma sağlar, ancak bunu yapmak
sözleşmenin ETH / USD değerinin ne olduğunu bilmesini gerektirir. Bunu yapmanın en basit yolu bir "veri"
belirli bir tarafın (ör. NASDAQ) sürdürdüğü feed "sözleşmesi, ilgili tarafın
sözleşmeyi gerektiği gibi güncelleyin ve diğer sözleşmelerin bir mesaj göndermesine izin veren bir arayüz sağlayın
bu sözleşme ve fiyatı sağlayan bir yanıt alın.
Bu kritik bileşen göz önüne alındığında, riskten korunma sözleşmesi aşağıdaki gibi görünecektir:
1.
A tarafının 1000 eter girmesini bekleyin.
2.
B tarafının 1000 eter girmesini bekleyin.
3.
Depolamadaki veri besleme sözleşmesini sorgulayarak hesaplanan 1000 eter USD değerini kaydedin, şunu söyleyin
x $.
4.
30 gün sonra, x $ değerinde eter göndermek için A veya B'nin sözleşmeye "ping" yapmasına izin verin (şu hesapla hesaplanır:
yeni fiyatı) A'ya ve geri kalanını B'ye almak için veri feed'i sözleşmesini tekrar sorgulamak.

Sayfa 21

Böyle bir sözleşme, kripto ticaretinde önemli bir potansiyele sahip olacaktır. Hakkında alıntı yapılan ana sorunlardan biri
kripto para birimi, değişken olduğu gerçeğidir; birçok kullanıcı ve tüccar güvenlik isteyebilir ve
kripto varlıklarla uğraşma kolaylığı nedeniyle, çoğu kişi% 23'ünü kaybetme olasılığıyla yüzleşmek istemiyor.
tek bir günde fonlarının değeri. Şimdiye kadar, en yaygın olarak önerilen çözüm,
ihraççı tarafından desteklenen varlıklar; buradaki fikir, bir ihraççının ihraç etme hakkına sahip olduğu bir alt para birimi yaratmasıdır.
ve birimleri iptal edin ve onlara (çevrimdışı) tek bir birim sağlayan herkese para biriminin bir birimini verin.
belirtilen dayanak varlık (ör. altın, USD). İhraççı daha sonra, temeldeki bir birim sağlamayı taahhüt eder.
kripto varlığın bir birimini geri gönderen herkese varlık. Bu mekanizma herhangi bir kriptografik olmayan
ihraç edene güvenilebilmesi koşuluyla, bir kriptografik varlığa "yükseltilecek" varlık.

Ancak uygulamada, kart çıkaranlar her zaman güvenilir değildir ve bazı durumlarda bankacılık altyapısı da güvenilir değildir.
bu tür hizmetlerin var olamayacağı kadar zayıf veya düşmanca. Finansal türevler bir alternatif sunar. Burada, a yerine
bir varlığın yedeklenmesi için fon sağlayan tek bir ihraççı, merkezi olmayan bir spekülatör pazarı,
Bir kriptografik referans varlığının fiyatı artacaktır, bu rolü oynar. İhraççıların aksine, spekülatörlerin riskten korunma sözleşmesi fonlarını emanette tuttuğu için pazarlığın kendi tarafında temerrüt. Bunu unutmayın
yaklaşım tam olarak merkezi olmayan bir yaklaşım değildir, çünkü fiyat şeridini sağlamak için hala güvenilir bir kaynağa ihtiyaç vardır,
Yine de tartışmasız olsa da bu, altyapı gereksinimlerinin azaltılması açısından büyük bir gelişmedir (bir ihraççı olmanın aksine, bir fiyat feed'i yayınlamak lisans gerektirmez ve muhtemelen konuşma özgürlüğü olarak kategorize edilebilir)
ve dolandırıcılık potansiyelini azaltmak.

Sayfa 22

Kimlik ve İtibar Sistemleri
En eski alternatif kripto para birimi olan Namecoin, aşağıdakileri sağlamak için Bitcoin benzeri bir blok zinciri kullanmaya çalıştı.
Kullanıcıların diğer verilerle birlikte herkese açık bir veritabanına adlarını kaydedebilecekleri ad kayıt sistemi.
Belirtilen başlıca kullanım örneği, "bitcoin.org" (veya Namecoin's içinde) gibi alan adlarını
eşleyen bir DNS sistemi içindir.
durumda, "bitcoin.bit") bir IP adresine. Diğer kullanım durumları arasında e-posta kimlik doğrulaması ve potansiyel olarak daha fazlası bulunur
gelişmiş itibar sistemleri. Namecoin benzeri bir isim kayıt sistemi sağlamak için temel sözleşme Ethereum'da:

```
if! contract.storage [tx.data [0]]:
    Contract.storage [tx.data [0]] = tx.data [1]
```

Sözleşme çok basit; hepsi Ethereum ağına eklenebilen, ancak eklenemeyen bir veritabanıdır.
değiştirildi veya kaldırıldı. Herkes bir değeri olan bir adı kaydedebilir ve bu kayıt daha sonra yapışır sonsuza dek. Daha karmaşık bir ad tescil sözleşmesi, diğerlerine izin veren bir "işlev maddesine" de sahip olacaktır.
sorgulama sözleşmelerinin yanı sıra, bir adın "sahibi" (yani ilk kaydedici) için bir mekanizma
veri veya sahipliği aktarın. Üstüne itibar ve güven ağı işlevselliği bile eklenebilir.

Merkezi Olmayan Dosya Depolama
Geçtiğimiz birkaç yıl içinde, bir dizi popüler çevrimiçi dosya depolama başlangıcı ortaya çıktı.
Kullanıcıların sabit disklerinin bir yedeğini yüklemelerine ve hizmete sahip olmalarına izin vermek isteyen Dropbox olarak öne çıkmaktadır.
yedeği depolayın ve kullanıcının aylık bir ücret karşılığında ona erişmesine izin verin. Ancak bu noktada dosya
depolama pazarı bazen nispeten verimsizdir; çeşitli mevcut çözümlere üstünkörü bir bakış
şunu gösterir:

özellikle ne ücretsiz kotaların ne de kurumsal düzeyde indirimlerin olmadığı "tekinsiz vadi" 20-200 GB düzeyinde
temel dosya depolama maliyetleri için aylık fiyatlar, şu şekildedir:
bir ayda tüm sabit disk. Ethereum sözleşmeleri, merkezi olmayan bir
Bireysel kullanıcıların kendi paralarını kiralayarak küçük miktarlarda para kazanabilecekleri dosya depolama ekosistemi
sabit diskler ve kullanılmayan alan, dosya depolama maliyetlerini daha da düşürmek için kullanılabilir.
Böyle bir cihazın temel dayanak noktası, "merkezi olmayan Dropbox" olarak adlandırdığımız şey olacaktır.
Sözleşme ". Bu sözleşme şu şekilde çalışır. İlk olarak, istenen veriler bloklara bölünür ve her bir bloğu şifreler.
gizlilik için ve ondan bir Merkle ağacı oluşturur. Daha sonra biri, her N blokta,
sözleşme, Merkle ağacında rastgele bir dizin seçer (önceki blok özetini kullanarak,
bir rastgelelik kaynağı olarak sözleşme kodu) ve X ether'i ilk varlığa bir işlem ile bir işlem tedarik etmesi için verin.

Sayfa 23

basitleştirilmiş ödeme doğrulaması benzeri bloğun ağaçtaki söz konusu endekste sahipliğinin kanıtı. Zaman
kullanıcı dosyasını yeniden indirmek istiyorsa, bir mikro ödeme kanalı protokolü kullanabilir (ör. 32 başına 1 szabo öde
kilobayt) dosyayı kurtarmak için; En düşük maliyetli yaklaşım, ödeyen için işlemi şu tarihe kadar yayınlamamaktır:
son olarak, bunun yerine işlemi biraz daha kazançlı bir işlemle, her seferinde aynı nonce ile değiştirir. 32 kilobayt.
Protokolün önemli bir özelliği, pek çok rastgele düğüme güveniyor gibi görünse de,
dosyayı unutmaya karar vermek için, dosyayı birçok parçaya bölerek bu riski sıfıra yakın bir seviyeye indirebilir.
gizli paylaşım ve her bir parçayı görmek için sözleşmeleri izlemek hala bazı düğümlerin mülkiyetindedir. Bir sözleşme ise
Hala para ödüyor, bu da birisinin hala dosyayı depoladığına dair kriptografik bir kanıt sağlıyor.

Merkezi Olmayan Otonom Kuruluşlar
"Merkezi olmayan bir kuruluş" genel kavramı, belirli bir kümeye sahip sanal bir varlık kavramıdır.
Muhtemelen% 67 çoğunluk ile kuruluşun fonlarını harcama hakkına sahip üyeler veya hissedarlar ve kodunu değiştirin. Üyeler, örgütün fonlarını nasıl tahsis etmesi gerektiğine toplu olarak karar vereceklerdi.
Bir DAO'nun fonlarını tahsis etme yöntemleri, ödüllerden, maaşlardan daha egzotik mekanizmalara kadar değişebilir.
işi ödüllendirmek için dahili bir para birimi gibi. Bu, esasen geleneksel bir hukukun yasal tuzaklarını taklit eder.
şirket veya kar amacı gütmeyen kuruluş, ancak uygulama için yalnızca kriptografik blok zinciri teknolojisini kullanıyor. Şimdiye kadar çoğu
DAO'lar etrafındaki konuşma, "ademi merkeziyetçi bir özerk şirket" (DAC) "kapitalist" modeli etrafında olmuştur
temettü alan hissedarlar ve alınıp satılabilir hisseler ile; bir alternatif, belki bir
"merkezi olmayan özerk topluluk", tüm üyelerin kararda eşit bir paya sahip olmasını sağlayacaktır.
mevcut üyelerin% 67'sinin bir üye eklemeyi veya çıkarmayı kabul etmesini gerektiriyor. Birinin şartı kişi yalnızca bir üyeliğe sahip olabilir, bu durumda grup tarafından toplu olarak uygulanmalıdır.
Bir DO'nun nasıl kodlanacağına ilişkin genel bir taslak aşağıdaki gibidir. En basit tasarım, basitçe kendi kendini değiştiren bir parçadır
Üyelerin üçte ikisi bir değişiklik üzerinde anlaşırsa değişen kod. Kod teorik olarak değişmez olsa da,

kod parçalarını ayrı sözleşmelerde bulundurarak bunun üstesinden kolayca gelebilir ve fiili değişkenliğe sahip olabilir,
ve hangi sözleşmelerin çağrılacağı adresinin değiştirilebilir depoda depolanmış olması. Basitçe Böyle bir DAO sözleşmesinin uygulanması, verilerle ayırt edilen üç işlem türü olacaktır.
işlemde sağlanan:

- [0, i, K, V] depolama indeksi K'deki adresi şu şekilde değiştirmek için i indeksi ile bir teklif kaydetmek için
değer V
- [0, i] teklif lehine oy vermek için i
- [2, i] yeterli oylama yapıldıysa teklifi sonuçlandırmak için i

Daha sonra sözleşmede bunların her biri için hükümler olacaktır. Tüm açık depolamanın bir kaydını tutacaktır.
onlara oy verenlerin bir listesi ile birlikte değişiklikler. Ayrıca tüm üyelerin bir listesi de olacaktır. Ne zaman herhangi bir depolama

---

Sayfa 24

değişiklik için oy veren üyelerin üçte ikisine ulaşırsa, bir sonlandırma işlemi değişikliği gerçekleştirebilir. Bir daha
gelişmiş iskelet ayrıca bir işlem gönderme, ekleme gibi özellikler için yerleşik oylama yeteneğine sahip olacaktır.
üyeler ve üyelerin çıkarılması ve hatta Sıvı Demokrasi tarzı oy delegasyonu sağlayabilir (örn.
herkes kendisine oy vermesi için birini atayabilir ve atama geçişlidir, bu nedenle A, B'yi ve B, C'yi atarsa
daha sonra C, A'nın oyunu belirler). Bu tasarım, DO'nun merkezi olmayan bir şekilde organik olarak büyümesine izin verecektir.
topluluk, insanların sonunda kimin uzmanlara üye olduğunu filtreleme görevini devretmesine izin verir,
ancak "mevcut sistem" den farklı olarak uzmanlar zaman içinde kolayca ortaya çıkıp çıkabilirler.
bireysel topluluk üyeleri uyumlarını değiştirir.
Alternatif bir model, herhangi bir hesabın sıfır veya daha fazla hisseye sahip olabileceği merkezi olmayan bir şirket içindir ve
Karar vermek için hisselerin üçte ikisi gereklidir. Tam bir iskelet, varlık içerir
yönetim işlevselliği, hisse satın alma veya satma teklifinde bulunma ve teklifleri kabul etme yeteneği (tercihen sözleşmenin içinde bir sipariş eşleştirme mekanizması ile). Delegasyon da Sıvı olacaktır.
Demokrasi tarzı, "yönetim kurulu" kavramını genelleştiriyor.
Gelecekte, kurumsal yönetişim için daha gelişmiş mekanizmalar uygulanabilir; bu işte
merkezi olmayan bir kuruluşun (DO) merkezi olmayan bir özerk olarak tanımlanmaya başlayabileceğine dikkat edin
organizasyon (DAO). DO ve DAO arasındaki fark belirsizdir, ancak genel bölme çizgisi,
yönetişim genellikle politik benzeri bir süreç veya "otomatik" bir süreç aracılığıyla gerçekleştirilir; iyi bir sezgisel
test "ortak dil yok" kriteridir: iki üye hiç konuşmadıysa kuruluş hala çalışabilir mi?
aynı dil? Açıkçası, basit bir geleneksel hissedar tarzı şirket başarısız olur, oysa bunun gibi bir şey Bitcoin protokolünün başarılı olma olasılığı çok daha yüksektir. Robin Hanson'un futarşi, bir mekanizma
tahmin piyasaları aracılığıyla kurumsal yönetişim, gerçekten "otonom" yönetişimin ne olduğuna iyi bir örnektir.
gibi görünebilir. Tüm DAO'ların tüm DO'lardan üstün olduğu varsayılmaması gerektiğini unutmayın; otomasyon
basitçe, belirli yerlerde çok büyük faydalara sahip olması muhtemel bir paradigmadır ve olmayabilir

diğerlerinde pratiktir ve birçok yarı DAO da mevcut olabilir.

Diğer Uygulamalar

1. Tasarruf cüzdanları . Alice'in parasını güvende tutmak istediğini, ancak kaybedeceğinden endişelendiğini veya
birisi onun özel anahtarını kıracak. Eter'i bir banka olan Bob ile şu şekilde bir sözleşme imzaladı:

- Alice tek başına günlük fonların en fazla% 1'ini çekebilir.
- Bob tek başına günde fonların en fazla% 1'ini çekebilir, ancak Alice,
Bu yeteneği kapatan anahtarı ile işlem.
- Alice ve Bob birlikte her şeyi çekebilir.

Normalde, Alice için günde% 1 yeterlidir ve Alice daha fazla para çekmek isterse yardım için Bob ile iletişime geçebilir. Eğer
Alice'in anahtarı ele geçirilir, parayı yeni bir sözleşmeye taşımak için Bob'a koşar. Anahtarını kaybederse, Bob alacak
sonunda fonlar tükendi. Bob'un kötü niyetli olduğu ortaya çıkarsa, geri çekilme yeteneğini kapatabilir.

Sayfa 25

2. Mahsul sigortası . Biri kolayca finansal türevler sözleşmesi yapabilir, ancak hava durumunun veri beslemesini kullanarak
herhangi bir fiyat endeksi yerine. Iowa'da bir çiftçi,
Iowa'da yağış, o zaman bir kuraklık varsa, çiftçi otomatik olarak para alacak ve varsa
Yeterince yağmur yağar, çiftçi mutlu olur çünkü mahsulleri iyi gelir.

3. Merkezi olmayan bir veri beslemesi . Fark için mali sözleşmeler için, aslında merkezden uzaklaşmak mümkün olabilir
veri " SchellingCoin" adlı bir protokol aracılığıyla beslenir . SchellingCoin temelde şu şekilde çalışır: N parti
sisteme belirli bir verinin değeri (örneğin, ETH / USD fiyatı), değerler sıralanır ve herkes
25. ve 75. yüzdelik dilim arasında, ödül olarak bir jeton alır. Herkesin sağlamak için teşviki vardır.
diğer herkesin sağlayacağı cevap ve çok sayıda oyuncunun gerçekçi bir şekilde sağlayabileceği tek değer
üzerinde anlaşmak bariz varsayılan şeydir: gerçek. Bu, teorik olarak sağlayabilen merkezi olmayan bir protokol oluşturur
ETH / USD fiyatı, Berlin'deki sıcaklık ve hatta belirli bir sonucun sonucu dahil olmak üzere herhangi bir sayıdaki değer
zor hesaplama.

4. Akıllı çoklu imza emaneti . Bitcoin çoklu imzalı işlem sözleşmelerine izin verir, örneğin,
verilen beş anahtardan üçü fonları harcayabilir. Ethereum daha fazla ayrıntıya izin verir; örneğin, dört çıkış
beşte biri her şeyi harcayabilir, beşte üçü günde% 10'a kadar harcayabilir ve beşte ikisi,
Günde% 0,5. Ek olarak, Ethereum multisig eşzamansızdır - iki taraf imzalarını şuraya kaydedebilir:
farklı zamanlarda blok zinciri ve son imza işlemi otomatik olarak gönderecektir.

5. Bulut bilişim . EVM teknolojisi, doğrulanabilir bir bilgi işlem ortamı oluşturmak için de kullanılabilir,
kullanıcıların başkalarından hesaplamalar yapmalarını istemelerine ve daha sonra isteğe bağlı olarak
hesaplamaların kanıtlarını istemelerine olanak sağlar.
rastgele seçilen belirli kontrol noktaları doğru şekilde yapıldı. Bu, bir bulut bilişimin oluşturulmasına izin verir
herhangi bir kullanıcının masaüstü, dizüstü bilgisayarları veya özel sunucuları ile katılabileceği pazar ve nokta kontrolü
güvenlik depozitoları ile birlikte sistemin güvenilir olmasını sağlamak için kullanılabilir (yani düğümler

karlı hile). Böyle bir sistem tüm görevler için uygun olmasa da; yüksek seviye gerektiren görevler Örneğin süreçler arası iletişim, büyük bir düğüm bulutunda kolayca yapılamaz. Diğer görevler, ancak paralelleştirmek çok daha kolaydır; SETI @ home, fold @ home ve genetik algoritmalar gibi projeler,
böyle bir platformun üzerine kolaylıkla uygulanabilir.
6. Eşler arası kumar . Frank Stajano ve
Richard Clayton's Cyberdice , Ethereum blok zincirinde uygulanabilir. En basit kumar
protokol aslında bir sonraki blok karmasındaki fark için bir sözleşmedir ve daha gelişmiş protokoller, sıfıra yakın ücretle hile yapma yeteneği olmayan kumar hizmetleri oluşturarak oradan kurulur.
7. Tahmin piyasaları . Bir oracle veya SchellingCoin sağlandığında tahmin piyasalarının da uygulanması kolaydır,
ve tahmin piyasaları, SchellingCoin ile birlikte,
merkezi olmayan kuruluşlar için bir yönetişim protokolü olarak futarşi .
8. Kimlik ve itibar sistemini temel olarak kullanan, **zincir üzerinde merkezi olmayan pazar yerleri** .

Çeşitli ve Endişeler
Değiştirilmiş GHOST Uygulaması
"Greedy Heaviest Observed Subtree" (GHOST) protokolü, ilk olarak Yonatan tarafından sunulan bir yeniliktir.
Sompolinsky ve Aviv Zohar, Aralık 2013'te. GHOST'un arkasındaki motivasyon,
Şu anda onay süreleri, yüksek bayat oranı nedeniyle düşük güvenlikten muzdariptir - çünkü bloklar bir ağda yayılması için belirli bir süre, eğer madenci A bir bloğu mayınlarsa ve sonra madenci B madene gelirse
madenci A'nın bloğu B'ye yayılmadan önce başka bir blok, madenci B'nin bloğu boşa gidecek ve katkıda bulunmayacaktır.
ağ güvenliğine. Ayrıca, bir merkezileştirme sorunu var: Madenci A,% 30 ile bir madencilik havuzuysa hashpower ve B'nin% 10 hashpower'ı vardır, A, zamanın% 70'inde bayat blok üretme riskine sahip olacaktır (çünkü
A'nın son bloğu ürettiği sürenin diğer% 30'u ve bu nedenle madencilik verilerini hemen alır), oysa B, % 90 oranında bayat blok üretme riski. Bu nedenle, blok aralığı, bayat oranı için yeterince kısaysa yüksek olduğunda, A sadece boyutu nedeniyle önemli ölçüde daha verimli olacaktır. Bu iki efekt birleştirildiğinde,
Hızlı blok üreten blok zincirlerinin, yeterince büyük bir madencilik havuzuna yol açma olasılığı çok yüksektir.
madencilik süreci üzerinde fiili kontrole sahip olmak için ağ hashp gücünün yüzdesi.
Sompolinsky ve Zohar tarafından açıklandığı gibi GHOST, ilk ağ güvenliği kaybı sorununu çözer.
hangi zincirin "en uzun" olduğu hesaplamasında eski bloklar; yani, sadece ebeveyn ve daha fazlası değil
bir bloğun ataları, ama aynı zamanda bloğun atalarının eski çocukları (Ethereum jargonunda, "amcalar")
hangi bloğun onu destekleyen en büyük toplam çalışma kanıtına sahip olduğu hesaplamaya eklenir. İkinci sorunu çözmek için
merkezileştirme önyargısı nedeniyle, Sompolinsky ve Zohar tarafından tanımlanan protokolün ötesine geçiyoruz ve ayrıca bayatların
bir blok ödülü almak için ana zincire kaydolmak: eski bir blok, temel ödülünün% 93,75'ini alır,
ve bayat bloğunu içeren yeğen kalan% 6,25'i alır. Ancak işlem ücretleri
amcalara ödüllendirildi.
Ethereum, GHOST'un yalnızca beş seviye aşağı inen basitleştirilmiş bir sürümünü uygular. Özellikle bayat
blok yalnızca ebeveyninin 2. ila 5. nesil çocuğu tarafından bir amca olarak dahil edilebilir ve herhangi bir blok dahil edilemez.

daha uzak bir ilişki (örneğin, bir ebeveynin 6. nesil çocuğu veya bir büyük ebeveynin 3. nesil çocuğu). Bu
çeşitli nedenlerle yapıldı. İlk olarak, sınırsız GHOST,
belirli bir blok için hangi amcaların geçerli olduğunun hesaplanması. İkincisi, kullanıldığı gibi tazminat ile sınırsız GHOST
Ethereum'da bir madencinin bir halkın zinciri yerine ana zincir üzerinde madencilik yapma teşviki kaldırıyor
saldırgan. Son olarak, hesaplamalar, teşvikli beş seviyeli GHOST'un, bir
15 sn blok süresi ve% 25 hashpower ile madenciler% 3'ten daha az merkezileşme kazancı gösteriyor.

Sayfa 27

Ücretler
Çünkü blok zincirinde yayınlanan her işlem, ağa ihtiyaç duyma maliyetini getirir.
indirin ve doğrulayın, genellikle işlem ücretlerini içeren bazı düzenleyici mekanizmalara ihtiyaç vardır.
suistimali önlemek. Bitcoin'de kullanılan varsayılan yaklaşım, madencilerin harekete geçmesine güvenerek tamamen gönüllü ücretlere sahip olmaktır.
kapı bekçileri olarak ve dinamik minimumlar belirledik. Bu yaklaşım,
Bitcoin topluluğu özellikle madenciler arasında arz ve talebe izin veren "piyasa temelli" olduğu için
ve işlem gönderenler fiyatı belirler. Bu akıl yürütme çizgisindeki sorun, bununla birlikte,
işlem yapmak bir piyasa değildir; işlem sürecini yorumlamak sezgisel olarak çekici olsa da
madencinin gönderene sunduğu bir hizmet olarak, gerçekte bir madencinin içerdiği her işlem gerekli olacaktır.
ağdaki her düğüm tarafından işlenecek, böylece işlem işleme maliyetinin büyük çoğunluğu karşılanacaktır.
üçüncü şahıslar tarafından ve dahil edilip edilmeyeceğine karar veren madenci tarafından değil. Bu nedenle,
ortak trajedi sorunlarının ortaya çıkması çok muhtemeldir.
Bununla birlikte, piyasa temelli mekanizmadaki bu kusur ortaya çıktığı gibi, belirli bir yanlış verildiğinde
varsayımı basitleştirmek, sihirli bir şekilde kendi kendini ortadan kaldırır. Argüman aşağıdaki gibidir. Farz et ki:
1.
Bir işlem, k işlemine yol açar ve ödül kR'yi, R'nin dahil olduğu herhangi bir madenciye sunar.
gönderen tarafından belirlenir ve k ve R (kabaca) madenci tarafından önceden görülebilir.
2.
Bir işlemin herhangi bir düğüm için C işlem maliyeti vardır (yani tüm düğümler eşit verimliliğe sahiptir)
3.
Her biri tam olarak eşit işlem gücüne sahip N madencilik düğümü vardır (yani toplamın 1 / N'si)
4.
Madencilik dışı tam düğüm mevcut değildir.
Bir madenci, beklenen ödül maliyetten büyükse bir işlemi gerçekleştirmeye istekli olacaktır. Böylece
Madencinin bir sonraki bloğu işleme şansı 1 / N olduğundan beklenen ödül kR / N'dir ve işlem
madenci için maliyet basitçe kC'dir. Bu nedenle, madenciler, kR / N> kC veya R> NC olan işlemleri içereceklerdir. Unutmayın ki R
gönderen tarafından sağlanan işlem başına ücrettir ve bu nedenle, gönderenin elde ettiği fayda için daha düşük bir sınırdır
işlemden elde edilir ve NC, bir işlemi işlemenin birlikte tüm ağa olan maliyetidir. Bu nedenle,
madenciler, yalnızca toplam faydacı faydanın değeri aşan işlemleri dahil etme teşviğine sahiptir.
maliyet.
Bununla birlikte, gerçekte bu varsayımlardan birkaç önemli sapma vardır:

1.
Madenci, işlemi gerçekleştirmek için diğer doğrulama düğümlerinden daha yüksek bir maliyet öder, çünkü
ekstra doğrulama süresi, blok yaymayı geciktirir ve böylece bloğun
bayat olmak.
2.
Madencilik dışı tam düğümler var.

Sayfa 28

3.
Madencilik enerji dağıtımı, pratikte radikal bir şekilde adaletsiz hale gelebilir.
4.
Yararlılık işlevi halkın zarar görmesini içeren spekülatörler, siyasi düşmanlar ve çılgınlar.
ağ mevcuttur ve maliyeti maliyetten çok daha düşük olan sözleşmeleri akıllıca kurabilirler.
diğer doğrulama düğümleri tarafından ödenir.
Yukarıdaki 1. Nokta, madenciye daha az işlem dahil etme eğilimi sağlar ve 2. nokta NC'yi artırır; dolayısıyla, bu iki etki en azından kısmen birbirini ortadan kaldırır. 3. ve 4. noktalar ana konudur; çözmek için
bunlara sadece kayan bir sınır koyarız: hiçbir blok, BLK_LIMIT_FACTOR katından daha fazla işleme sahip olamaz.
uzun vadeli üstel hareketli ortalama. Özellikle:
blk.oplimit = floor ((blk.parent.oplimit * (EMAFACTOR - 1) + floor (parent.opcount * BLK_LIMIT_FACTOR)) /
EMA_FACTOR)
BLK_LIMIT_FACTOR ve EMA_FACTOR, şimdilik 65536 ve 1.5 olarak ayarlanacak sabitlerdir, ancak
daha fazla analizden sonra muhtemelen değiştirilecektir.
Hesaplama ve Turing-Tamlık
Önemli bir not, Ethereum sanal makinesinin Turing-tamamlanmış olmasıdır; bu, EVM kodunun sonsuz döngüler de dahil olmak üzere gerçekleştirilebilecek herhangi bir hesaplamayı kodlar. EVM kodu döngüye izin verir
iki şekilde. İlk olarak, programın bir önceki noktaya atlamasına izin veren bir JUMP talimatı vardır. kod ve koşullu atlama yapmak için bir JUMPI komutu, while x <27: x = x * 2 gibi ifadelere izin verir. İkincisi, sözleşmeler diğer sözleşmeleri çağırabilir ve potansiyel olarak özyineleme yoluyla döngüye izin verebilir. Bu doğal olarak
bir soruna yol açar: Kötü niyetli kullanıcılar madencileri ve tam düğümleri girmeye zorlayarak kapatabilir
sonsuz bir döngü içine mi? Sorun, bilgisayar bilimindeki durma sorunu olarak bilinen bir sorundan kaynaklanmaktadır:
Genel durumda, belirli bir programın durup durmayacağını söylemenin bir yolu yoktur.
Durum geçişi bölümünde açıklandığı gibi, çözümümüz bir maksimum belirleme işlemi gerektirerek çalışır.
atılmasına izin verilen hesaplama adımlarının sayısı ve yürütme daha uzun sürerse hesaplama geri alınır
ancak ücretler hala ödeniyor. Mesajlar aynı şekilde çalışır. Çözümümüzün arkasındaki motivasyonu göstermek için düşünün
aşağıdaki örnekler:

- Bir saldırgan, sonsuz bir döngü çalıştıran bir sözleşme oluşturur ve ardından bir işlem gönderir bu döngüyü madenciye aktive etmek. Madenci, sonsuz döngüyü çalıştırarak işlemi işleyecek, ve gazının bitmesini bekleyin. İnfazın gazı bitip yarı yolda durmasına rağmen
üzerinden, işlem hala geçerlidir ve madenci yine de saldırganın her biri için ücret talep etmektedir. hesaplama adımı.

- Bir saldırgan, madenciyi devam ettirmeye zorlamak amacıyla çok uzun sonsuz bir döngü oluşturur.
o kadar uzun bir süre hesaplama ki, zaman hesaplama bittiğinde birkaç blok daha
çıkar ve madencinin ücreti talep etmesi için işlemi dahil etmesi mümkün olmayacaktır. Ancak,

---

Sayfa 29

saldırganın STARTGAS için hesaplama sayısını sınırlayan bir değer göndermesi gerekecektir.
yürütmenin atabileceği adımlar, böylece madenci hesaplamanın bir süre alacağını önceden bilecektir.
çok fazla adım.

- Saldırgan, send (A, contract.storage [A]) gibi bir kod içeren bir sözleşme görür;
Contract.storage [A] = 0 ve yalnızca ilk adımı çalıştırmak için yeterli gaz içeren bir işlem gönderir,
ancak
ikinci (yani, para çekme ama bakiyenin düşmesine izin vermeme). Sözleşme yazarı,
bu tür saldırılara karşı korunma konusunda endişelenmeniz gerekir, çünkü yürütme,
değişiklikler geri alınır.
- Bir finansal sözleşme, dokuz özel veri beslemesinin medyanını alarak çalışır.
riski en aza indirin. Bir saldırgan, veri akışlarından birini devralır ve bu veri akışlarından birini
DAO'larla ilgili bölümde açıklanan değişken adresli çağrı mekanizması ve bunu bir
sonsuz döngü, böylece finansal sözleşmeden fon talep etme girişimlerini
benzini bitmek. Bununla birlikte, finansal sözleşme, bunu önlemek için mesaja bir gaz limiti koyabilir.
sorun.

Turing tamlığına alternatif, JUMP ve JUMPI'nin bulunmadığı Turing-eksikliğidir ve
herhangi bir zamanda çağrı yığınında her sözleşmenin yalnızca bir kopyasının bulunmasına izin
verilir. Bu sistem ile ücret
sistem tanımlandı ve çözümümüzün etkinliğiyle ilgili belirsizlikler gerekli olmayabilir, çünkü
Bir sözleşmeyi yürütmenin maliyeti, yukarıda boyutuna göre sınırlandırılacaktır. Ek olarak, Turing-
eksiklik değil
o kadar büyük bir sınırlama bile; dahili olarak tasarladığımız tüm sözleşme örneklerinden şimdiye
kadar sadece biri
bir döngü gerektiriyordu ve bu döngü bile tek satırlık bir kod parçasının 26 tekrarı yapılarak
kaldırılabilirdi.

Turing tamlığının ciddi sonuçları ve sınırlı faydası göz önüne alındığında, neden basitçe bir
Tamamlanmamış turing dili? Gerçekte, ancak, Turing-eksikliği, sorunun çözümü için temiz bir çözüm
olmaktan çok uzaktır.
sorun. Nedenini görmek için aşağıdaki sözleşmeleri göz önünde bulundurun:

C0: çağrı (C1); çağrı (C1);
C1: çağrı (C2); çağrı (C2);
C2: çağrı (C3); çağrı (C3);
...
C49: çağrı (C50); çağrı (C50);
C50: (bir programın bir adımını çalıştırın ve depodaki değişikliği kaydedin)

Şimdi, A'ya bir işlem gönderin. Böylece, 51 işlemde, 2 50 hesaplama alan bir sözleşmemiz var.
adımlar. Madenciler, her birinin yanında bir değer koruyarak bu tür mantık bombalarını önceden tespit
etmeye çalışabilirler.
alabileceği maksimum hesaplama adımı sayısını belirten ve bunu hesaplayan sözleşme
diğer sözleşmeleri yinelemeli olarak çağıran sözleşmeler, ancak bu, madencilerin oluşturan
sözleşmeleri yasaklamasını gerektirecektir.
diğer sözleşmeler (yukarıdaki 50 sözleşmenin tamamının oluşturulması ve yürütülmesi kolayca tek bir
sözleşmeye dönüştürülebildiğinden)
sözleşme). Bir diğer sorunlu nokta, bir mesajın adres alanının bir değişken olmasıdır, bu nedenle genel
olarak

Belirli bir sözleşmenin vaktinden önce hangi diğer sözleşmeleri arayacağını söylemek bile mümkün değildir. Sonuç olarak, biz
şaşırtıcı bir sonuca var: Turing bütünlüğünün yönetilmesi şaşırtıcı derecede kolaydır ve

---

Sayfa 30

Tam olarak aynı kontroller yapılmadıkça, turing tamlığının yönetilmesi de şaşırtıcı derecede zordur - ama bu durumda neden protokolün Turing-tamamlanmış olmasına izin vermiyorsunuz?

Para Birimi ve İhraç
Ethereum ağı, kendi yerleşik para birimi olan ether'i içerir ve bu iki amaca hizmet eder.
çeşitli dijital varlık türleri arasında verimli alışverişe izin veren birincil likidite katmanı ve daha fazlası daha önemlisi, işlem ücretlerinin ödenmesi için bir mekanizma sağlanması. Rahatlık ve gelecekten kaçınmak için
argüman (Bitcoin'deki mevcut mBTC / uBTC / satoshi tartışmasına bakın), mezhepler önceden etiketlenecektir:

- 1: wei
- 10 ^ 12: szabo
- 10 ^ 15: finney
- 10 ^ 18: eter

Bu, "dolar" ve "sent" veya "BTC" ve "satoshi" kavramlarının genişletilmiş bir versiyonu olarak alınmalıdır. Yakın gelecekte, "eter" in olmasını bekliyoruz.
olağan işlemler için, mikro işlemler için "finney" ve ücretler ve protokolle ilgili teknik tartışmalar için "szabo" ve "wei" kullanılır
uygulama.
İhraç modeli aşağıdaki gibi olacaktır:

- Ethereum, BTC başına 1337-2000 eter fiyatından döviz satışıyla piyasaya sürülecek.
Ethereum organizasyonunu finanse etmeyi ve yapılan geliştirme için ödeme yapmayı amaçlayan mekanizma
başka bir dizi kriptografik platform tarafından başarıyla kullanılmaktadır. Daha önceki alıcılar daha büyük
indirimler. Satıştan alınan BTC, tamamen maaş ve ikramiyeleri ödemek için kullanılacaktır.
kripto para ekosistemindeki geliştiriciler, araştırmacılar ve projeler.
- Satılan toplam miktarın 0,099 katı, katılan erken katkıda bulunanlara tahsis edilecektir.
BTC finansmanı veya finansman kesinliği sağlanmadan önce gelişme ve başka bir 0.099x
uzun vadeli araştırma projelerine tahsis edilmiştir.
- Satılan toplam miktarın 0,26 katı, bu noktadan sonra sonsuza kadar madencilere her yıl tahsis edilecektir.

Sayfa 31

İhraç Dağılımı
Kalıcı doğrusal arz büyüme modeli, bazılarının aşırı zenginlik olarak gördüğü riski azaltır.
Bitcoin'de yoğunlaşma ve şimdiki ve gelecek çağlarda yaşayan bireylere edinme şansı verir.
para birimleri, aynı zamanda eterin amortismanını caydırıyor çünkü "arz büyüme oranı"

bir yüzde, zaman içinde hala sıfır olma eğilimindedir. Ayrıca, madeni paraların her zaman zamanla kaybolması nedeniyle
dikkatsizlik, ölüm vb. ve madeni para kaybı, yıllık toplam arzın bir yüzdesi olarak modellenebilir. Dolaşımdaki toplam para arzı aslında nihayetinde yıllık ihracata eşit bir değerde istikrar kazanacaktır. kayıp oranına bölünür (ör.% 1 kayıp oranında, arz 26X'e ulaştığında 0,26X çıkarılacak ve
Her yıl 0,26 kat kaybederek bir denge yaratır).

Grup
Öğle yemeğinde
1 yıl sonra
5 yıl sonra
Para birimleri
1.198X
1.458X
2.498X
Alıcılar
% 83.5
% 68.6
% 40.0
Erken katılımcı dağılımı
% 8.26
% 6.79
% 3.96
Uzun vadeli bağış
% 8.26
% 6.79
% 3.96
Madenciler
% 0
% 17.8
% 52.0
Doğrusal para ihracı olmasına rağmen, tıpkı Bitcoin'de olduğu gibi zaman içinde arz büyüme oranı yine de eğilimlidir.
sıfıra.

Sayfa 32

Madencilik Merkezileştirme
Bitcoin madenciliği algoritması temelde madencilerin SHA256'yı biraz değiştirilerek hesaplamasını sağlayarak çalışır.
Blok başlığının sürümleri milyonlarca kez tekrar tekrar, ta ki sonunda bir düğüm bir
hash değeri hedefin altında olan sürüm (şu anda 2.190 civarında). Ancak, bu madencilik algoritması iki merkezileştirme biçimine karşı savunmasız. Birincisi, madencilik ekosistemine ASIC'ler hakim oldu
(uygulamaya özel entegre devreler), bilgisayar çipleri için tasarlanmış ve dolayısıyla binlerce kat daha fazlası
Bitcoin madenciliğinin özel görevinde verimli. Bu, Bitcoin madenciliğinin artık yüksek bir
Etkili bir şekilde katılmak için milyonlarca dolarlık sermaye gerektiren merkezi olmayan ve eşitlikçi bir arayış.
İkinci olarak, çoğu Bitcoin madencisi yerel olarak blok doğrulaması yapmaz; bunun yerine, bir
blok başlıklarını sağlamak için merkezi madencilik havuzu. Bu sorun tartışmalı olarak daha kötüdür: şu an itibariyle
Yazılı olarak, en iyi iki madencilik havuzu, Bitcoin ağındaki işlem gücünün kabaca% 50'sini dolaylı olarak kontrol eder,

Bu, madencilerin bir havuz veya koalisyon durumunda diğer madencilik havuzlarına geçebilmeleri gerçeğiyle hafifletilmektedir.
% 51'lik bir saldırı girişiminde bulunur.
Ethereum'daki mevcut amaç, rastgele benzersiz bir hash oluşturmaya dayalı bir madencilik algoritması kullanmaktır.
Her 1000 nonce için işlev, yeterince geniş bir hesaplama aralığı kullanarak
özel donanım. Böyle bir strateji, merkezileşme kazancını kesinlikle sıfıra indirmeyecektir, ancak gerek yok. Her bir kullanıcının, kendi özel dizüstü bilgisayarında veya masaüstünde belirli bir miktar gerçekleştirebileceğini unutmayın.
madencilik faaliyeti neredeyse ücretsiz, sadece elektrik maliyetleri ödüyor, ancak bunların% 100 CPU kullanımı noktasından sonra
bilgisayar ek madenciliği, hem elektrik hem de donanım için ödeme yapmalarını gerektirecektir. ASIC madencilik şirketleri
ilk hash'den başlayarak elektrik ve donanım için ödeme yapmanız gerekir. Dolayısıyla, merkezileşme kazancı olabilirse
bu oranın (E + H) / E altında tutulursa, ASIC'ler yapılsa bile sıradan madenciler için hala yer olacaktır.
Ek olarak, madencilik algoritmasını madenciliğin tamamına erişim gerektirecek şekilde tasarlamayı planlıyoruz.
Blockchain, madencileri tüm blok zincirini depolamaya ve en azından her işlemi doğrulamaya zorlar.
Bu, merkezi madencilik havuzlarına olan ihtiyacı ortadan kaldırır; madencilik havuzları hala meşru bir role hizmet edebilir
Ödül dağılımının rastlantısallığını ortadan kaldırmak için bu işlev, eşler arası tarafından eşit derecede iyi bir şekilde sunulabilir.
merkezi kontrolü olmayan havuzlar. Ayrıca, içindeki tam düğüm sayısını artırarak merkezileşme ile mücadeleye yardımcı olur.
ağ, böylece çoğu sıradan kullanıcı ışığı tercih etse bile ağın makul ölçüde merkezi olmayan kalması için
müşteriler.

Sayfa 33

Ölçeklenebilirlik
Ethereum ile ilgili ortak bir endişe, ölçeklenebilirlik konusudur. Bitcoin gibi, Ethereum da kusurdan muzdarip
her işlemin ağdaki her düğüm tarafından işlenmesi gerekir. Bitcoin ile akımın boyutu
Blockchain, saatte yaklaşık 1 MB büyür ve yaklaşık 20 GB'tır. Bitcoin ağı Visa'ları işleyecek olsaydı
Saniyede 2000 işlem, üç saniyede 1 MB büyür (saatte 1 GB, yılda 8 TB).
Ethereum'un, birçok uygulama olacağı gerçeğiyle daha da kötüleşen benzer bir büyüme modeline sahip olması muhtemeldir.
Bitcoin'de olduğu gibi sadece bir para birimi yerine Ethereum blok zincirinin üstünde, ancak
Ethereum tam düğümlerinin tüm blok zinciri geçmişi yerine sadece durumu depolaması gerektiği gerçeği.
Böylesine büyük bir blok zinciri boyutundaki sorun, merkezileştirme riskidir. Blockchain boyutu artarsa, örneğin,
100 TB ise, olası senaryo, yalnızca çok az sayıda büyük işletmenin tam olarak çalışacağıdır.
hafif SPV düğümlerini kullanan tüm normal kullanıcılar ile düğümler. Böyle bir durumda, potansiyel endişe ortaya çıkar.
tam düğümler bir araya gelebilir ve hepsi karlı bir şekilde hile yapmayı kabul edebilir (ör. bloğu değiştirme
ödül, kendilerine BTC verin). Işık düğümlerinin bunu hemen algılamanın bir yolu yoktur. Tabii ki, en az bir dürüst tam düğüm muhtemelen mevcut olacak ve birkaç saat sonra dolandırıcılıkla ilgili bilgiler damlayacaktı
Reddit gibi kanallar aracılığıyla, ancak bu noktada çok geç olacaktır: Sıradan kullanıcıların

Verilen blokları kara listeye almak için bir çaba organize edin, bu büyük ve olasılıkla gerçekleştirilemez bir koordinasyon problemidir.
% 51'lik başarılı bir saldırıyı gerçekleştirmeye benzer bir ölçek. Bitcoin söz konusu olduğunda, bu şu anda bir sorundur,
ancak Peter Todd tarafından önerilen ve bu sorunu hafifletecek bir blok zinciri değişikliği var .
Yakın vadede, Ethereum bu sorunla başa çıkmak için iki ek strateji kullanacak. İlk olarak, çünkü Blockchain tabanlı madencilik algoritmaları, en azından her madenci tam bir düğüm olmaya zorlanacak ve
tam düğüm sayısına bağlıdır. İkincisi ve daha da önemlisi, bununla birlikte, bir ara
her işlemi işledikten sonra blok zincirindeki ağaç kökünü durum. Blok doğrulama merkezileştirilmiş olsa bile,
dürüst bir doğrulama düğümü olduğu sürece, merkezileştirme sorunu bir doğrulama yoluyla çözülebilir.
protokol. Bir madenci geçersiz bir blok yayınlarsa, bu blok ya kötü biçimlendirilmiş olmalı ya da S [n] durumu
yanlış. S [0] 'nın doğru olduğu bilindiğinden, S [i-1] olduğu yerde yanlış olan bazı ilk S [i] durumu olmalıdır.
doğru. Doğrulama düğümü, alt kümeden oluşan bir "geçersizlik kanıtı" ile birlikte indeks i'yi sağlayacaktır.
UYGULAMA işlemesi gereken Patricia ağaç düğümlerinin sayısı (S [i-1], TX [i]) -> S [i]. Düğümler bu düğümleri kullanarak
hesaplamanın bu bölümünü çalıştırın ve üretilen S [i] 'nin sağlanan S [i] ile eşleşmediğine bakın.
Daha karmaşık bir başka saldırı, kötü niyetli madencilerin tamamlanmamış bloklar yayınlamasını içerecektir.
blokların geçerli olup olmadığını belirlemek için tam bilgi bile mevcut değildir. Bunun çözümü bir sınama-yanıt protokolü: doğrulama düğümleri, hedef işlem endeksleri biçiminde "zorluklar" yayınlar, ve bir düğüm alındığında hafif bir düğüm bloğu, madenci veya madenci olsun, başka bir düğüme kadar güvenilmez olarak değerlendirir.
başka bir doğrulayıcı, geçerlilik kanıtı olarak Patricia düğümlerinin bir alt kümesini sağlar.

Sayfa 34

Hepsini Bir Araya Getirmek: Merkezi Olmayan Uygulamalar
Yukarıda açıklanan sözleşme mekanizması, herkesin esasen bir komut satırı oluşturmasına izin verir tüm ağ üzerinde fikir birliği ile yürütülen bir sanal makinede çalışan uygulama,
küresel olarak erişilebilir bir durumu "sabit disk" olarak değiştirebilir. Ancak çoğu insan için komut satırı arayüzü
yani işlem gönderme mekanizması, ademi merkeziyetçiliği bir
çekici ana akım alternatifi. Bu amaçla, eksiksiz bir "merkezi olmayan uygulama" her ikisinden de oluşmalıdır.
Düşük seviyeli iş mantığı bileşenleri, tamamen Ethereum'da uygulanmış olsun, aşağıdakilerin bir kombinasyonunu kullanarak:
Ethereum ve diğer sistemler (ör. Şu anda birisinin eklenmesi planlanan bir P2P mesajlaşma katmanı) Ethereum istemcileri) veya tamamen diğer sistemler ve üst düzey grafik kullanıcı arayüzü bileşenleri.
Ethereum istemcisinin tasarımı bir web tarayıcısı olarak hizmet etmektir, ancak bir "eth" Javascript API nesnesi için destek içerir,
İstemcide görüntülenen hangi özel web sayfalarının Ethereum blok zinciri ile etkileşim için kullanılabileceği.
"Geleneksel" web açısından bakıldığında, bu web sayfaları tamamen statik içeriktir, çünkü blok zinciri ve diğer merkezi olmayan protokoller, sunucunun amacına uygun olarak tam bir ikame görevi görecektir.
kullanıcı tarafından başlatılan isteklerin işlenmesi. Sonunda, merkezi olmayan protokoller, umarım kendileri bir şekilde
Ethereum kullanmak, web sayfalarını saklamak için kullanılabilir.

Sonuç
Ethereum protokolü, başlangıçta bir kripto para biriminin yükseltilmiş bir versiyonu olarak tasarlandı.
Blockchain üzerinde emanet, para çekme limitleri ve finansal sözleşmeler, kumar gibi gelişmiş özellikler
piyasalar ve benzerleri oldukça genelleştirilmiş bir programlama dili aracılığıyla. Ethereum protokolü
Uygulamalardan herhangi birini doğrudan "destekleyin", ancak bir Turing-complete programlama dilinin varlığı
herhangi bir işlem türü veya uygulaması için teorik olarak keyfi sözleşmelerin oluşturulabileceği anlamına gelir. Nedir
Ethereum hakkında daha ilginç olan ise, Ethereum protokolünün sadece para biriminin çok ötesine geçmesidir.
Merkezi olmayan dosya depolama, merkezi olmayan hesaplama ve
Merkezi olmayan tahmin piyasaları, bu tür düzinelerce diğer kavramlar arasında, önemli ölçüde
hesaplama endüstrisinin verimliliğini artırmak ve diğer eşler arası için büyük bir destek sağlamak
ilk kez ekonomik bir katman ekleyerek protokoller. Son olarak, önemli bir dizi de vardır.
parayla hiçbir ilgisi olmayan uygulamalar.
Ethereum protokolü tarafından uygulanan rastgele durum geçiş işlevi kavramı,
benzersiz potansiyele sahip platform; kapalı uçlu, tek amaçlı bir protokol olmaktan ziyade,
Veri depolama, kumar veya finans alanında belirli uygulamalar dizisi, Ethereum tasarım gereği açık uçludur ve biz
her ikisinin de çok büyük bir kısmı için temel bir katman olarak hizmet etmek için son derece uygun olduğuna inanıyorum.
Önümüzdeki yıllarda mali ve mali olmayan protokoller.

Sayfa 35

Notlar ve Ek Okumalar
Notlar
1.
Deneyimli bir okuyucu, aslında bir Bitcoin adresinin eliptik eğrinin karması olduğunu fark edebilir.
genel anahtar değil, genel anahtarın kendisi. Bununla birlikte, aslında tamamen meşru kriptografik terminolojidir.
pubkey hash değerini bir genel anahtar olarak adlandırın. Bunun nedeni, Bitcoin'in kriptografisinin bir
özel dijital imza algoritması, burada genel anahtar, ECC pubkey karma değerinden, imzadan oluşur
ECC imzasıyla birleştirilmiş ECC pubkeyinden oluşur ve doğrulama algoritması şunları içerir:
Genel anahtar olarak sağlanan ECC pubkey hash ile imzadaki ECC pubkey'in kontrol edilmesi ve ardından
ECC imzasını ECC yayın anahtarına göre doğrulamak.
2.
Teknik olarak, önceki 11 bloğun medyanı.
3.
Dahili olarak, 2 ve "CHARLIE" her ikisi de sayıdır, ikincisi big-endian taban 256'dır.
temsil. Sayılar en az 0 ve en fazla 2 ^ 256-1 olabilir.
Daha fazla okuma
1.
İçsel değer: https://tinyurl.com/BitcoinMag-IntrinsicValu e
2.
Akıllı mülk: https://en.bitcoin.it/wiki/Smart_Property
3.
Akıllı sözleşmeler: https://en.bitcoin.it/wiki/Contracts

4.
B-para: http://www.weidai.com/bmoney.txt
5.
Yeniden kullanılabilir çalışma kanıtları: http://www.finney.org/~hal/rpow/
6.
Mülkiyet başlıklarını mal sahibi yetkisiyle güvence altına
alın: http://szabo.best.vwh.net/securetitle.html
7.
Bitcoin teknik raporu: http://bitcoin.org/bitcoin.pdf
8.
Namecoin: https://namecoin.org/
9.
Zooko'nun üçgeni: http://en.wikipedia.org/wiki/Zooko's_triangle
10.
Renkli madeni paralar teknik raporu: https://tinyurl.com/coloredcoin-whitepaper
11.
Mastercoin teknik raporu: https://github.com/mastercoin-MSC/spec
12.
Merkezi olmayan özerk şirketler, Bitcoin Magazine: https://tinyurl.com/Bootstrapping-DAC s
13.
Basitleştirilmiş ödeme
doğrulaması: https://en.bitcoin.it/wiki/Scalability#Simplifiedpaymentverification
14.
Merkle ağaçları: http://en.wikipedia.org/wiki/Merkle_tree
15.
Patricia ağaçları: http://en.wikipedia.org/wiki/Patricia_tree
16.
HAYALET: http://www.cs.huji.ac.il/~avivz/pubs/13/btc_scalability_full.pdf
17.
StorJ ve Otonom Ajanlar, Jeff Garzik: https://tinyurl.com/storj-agent s
18.
Turing Festival'de Akıllı Mülkiyet hakkında Mike Hearn :
http://www.youtube.com/watch?v=Pu4PAMFPo5Y

Sayfa 36

19.
Ethereum RLP: https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-RLP
20.
Ethereum Merkle Patricia ağaçları : https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-Patricia-Tree
21.
Merkle toplam ağaçlarında Peter Todd : http: //sourceforge.net/p/bitcoin/mailman/message/31709140/