ブロックチェーンの課題と解決策の進展に基づく展望の考察

ccatak / https://cc-res.com**管理人**
Email: ccatak@cc-res.com
Website: https://cc-res.com

講演資料の公開場所：https://cc-res.com/dsgws_ccatak_materials_20181119

ブロックチェーン？

- ❑ 一般人のイメージとして、検索されているワードの一部を見てみる。
- ❑ 今回のプレゼンで少しでも関連しそうなワードを主観で赤字にした。

Googleサジェスト

- ブロックチェーン、**ブロックチェーンとは**、ブロックチェーン技術、ブロックチェーンゲーム、ブロックチェーン　応用、ブロックチェーンウォレット、ブロックチェーン　本、ブロックチェーン　プログラミング、ブロックチェーン　メリット、ブロックチェーンインフォ

ちょっと気になる系

- wiki、英語、**仕組み**、中国語、使い方、使いどころ、**定義**、とは、何ができる、何がすごい、**特徴**、比較、平野、未来、向いている、**メリット**、**問題点**、**問題**、有名人、用途、用語、利点、リスク、歴史、レボリューション、わかりやすく、わかりやすい、画像、技術、限界、雑誌、図解、図、図解pdf、動画、**デメリット**、ビットコイン、ブログ、バズワード

真面目に勉強系

- 研修、講座、資格、スクール、セミナー、入門、本、**論文**、学校、学習、大学、勉強会、勉強

詳しく知りたい系

- **ウォレット**、エネルギー、クラウド、クラウド 違い、コンソーシアム、**スマートコントラクト**、**セキュリティ**、**タイムスタンプ**、多数決、**追跡**、トークン、**ナンス**、認証、**ネットワーク**、**ノード**、**nonce**、**ハッシュ**、**非中央集権**、**秘密鍵**、**ヘッダ**、マイニング、マイニングとは、ユースケース、ライトニング、量子コンピュータ、**ルール**、**レイヤー**、**合意形成**、脆弱性、ゼロダウンタイム、台帳、**difficulty**、distributed、distributed ledger、データベース、電力、分散台帳、パブリック プライベート、**パブリック**、**ピアツーピア**、**ピア**、プラットフォーム、peer、**pow**、poc

イベント系

- イベント、展示会、フェスティバル

応用/事例/取組系

- （ざっくり）応用 、応用事例 、活用、サービス 、サービス例、**事例**、実用例、導入、導入事例、ビジネス、ビジネス活用
- （分野/商品/事例） AI、アプリ、IOT、医療、音楽、決済、広告、サプライチェーン、**送金**、製造業、著作権、チケット、**猫**、農業、不動産、ヘルスケア、保険、マーケティング、Miyabi、mijinメディア、野菜、薬、World State、銀行、軍事、**ゲーム**、在庫管理、在庫、物流、貿易、貿易 NTT、貿易金融、BAAS、ペット、ポイント
- （企業）Amazon、Qiita、NEC、協会、企業、ソフトバンク、富士通、MUFG、モバイルファクトリー、NTTデータ、NRI、NHK、グノシー、グーグル、Google、Zoom、ZOZO、象印、大和、団体、ベンチャー

政府系

- 経済産業省、総務省、法律

開発系

- （開発）開発、作り方、ハッカソン、ライブラリ、Ruby、rust、hello world、言語、go言語、実装、パイソン、**プログラミング**、プログラミング言語
- （求人）エンジニア、求人、転職、技術者

投資/投資管理系

- 仮想通貨、市場規模、スマホ、ニュース、仮想通貨　盗まれる、ハッキング、銘柄、無料、ランキング、ログイン、税金、バブル、pin、ペアリング

謎

- winny、宅配ボックス、指輪、ロゴ、ワイン

※ データはhttps://www.related-keywords.com/より。「ブロックチェーン」で検索して現れた関連ワードをひらがな順で上位3つを抽出して分類。分類は個人的な主観に基づく。（2018/11/18時点）

Satoshiは始まりの論文で何を語ったか

- 「信頼できる第三者機関」を介することによる不可逆性の問題と、「信頼できる第三者機関」がいないと発生してしまう二重支払い問題を同時に克服する電子決済システムを提案した。
- 暗号理論を基盤にP2Pネットワークが処理する機構とすることで、「信頼できる第三者機関」を排する。

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

Satoshiの問題意識

「信頼できる第三者機関」を介する電子決済は、だいたいうまくいくが、信頼ベースのモデルの弱点＝取引の不可逆性がないことの問題に悩まされている。例えば…
- 少額決済ができない。
- 顧客に多くの情報を求める。
- 一定割合の詐欺は不可避である。

電子署名を利用することで一部解決できるものもあるが、二重支払い防止のために「信頼できる第三者機関」が間に入ってしまうと、結局信頼ベースモデルの弱点が露顕する。

Satoshiの提案

「信頼できる第三者」を介さず、「暗号理論の裏付け」に基づいて二者間の直接取引が可能で、かつ二重支払いも防ぐ電子決済システム。
⇒信頼ベースモデルの弱点を克服しつつ二重支払いも防ぐ。

P2Pネットワークがトランザクションをハッシュ化し、タイムスタンプを押してプルーフオブワークのチェーンに記録する。プルーフオブワークをやり直さない限り記録を変更することはできない。
さらに最長のチェーンが正であるので、ハッシュパワーの半数以上を良心的なノードが掌握している限り、良心的なノードの行うプルーフオブワークの速度が攻撃者のプルーフオブワークの速度を上回るので攻撃者は改竄できない。

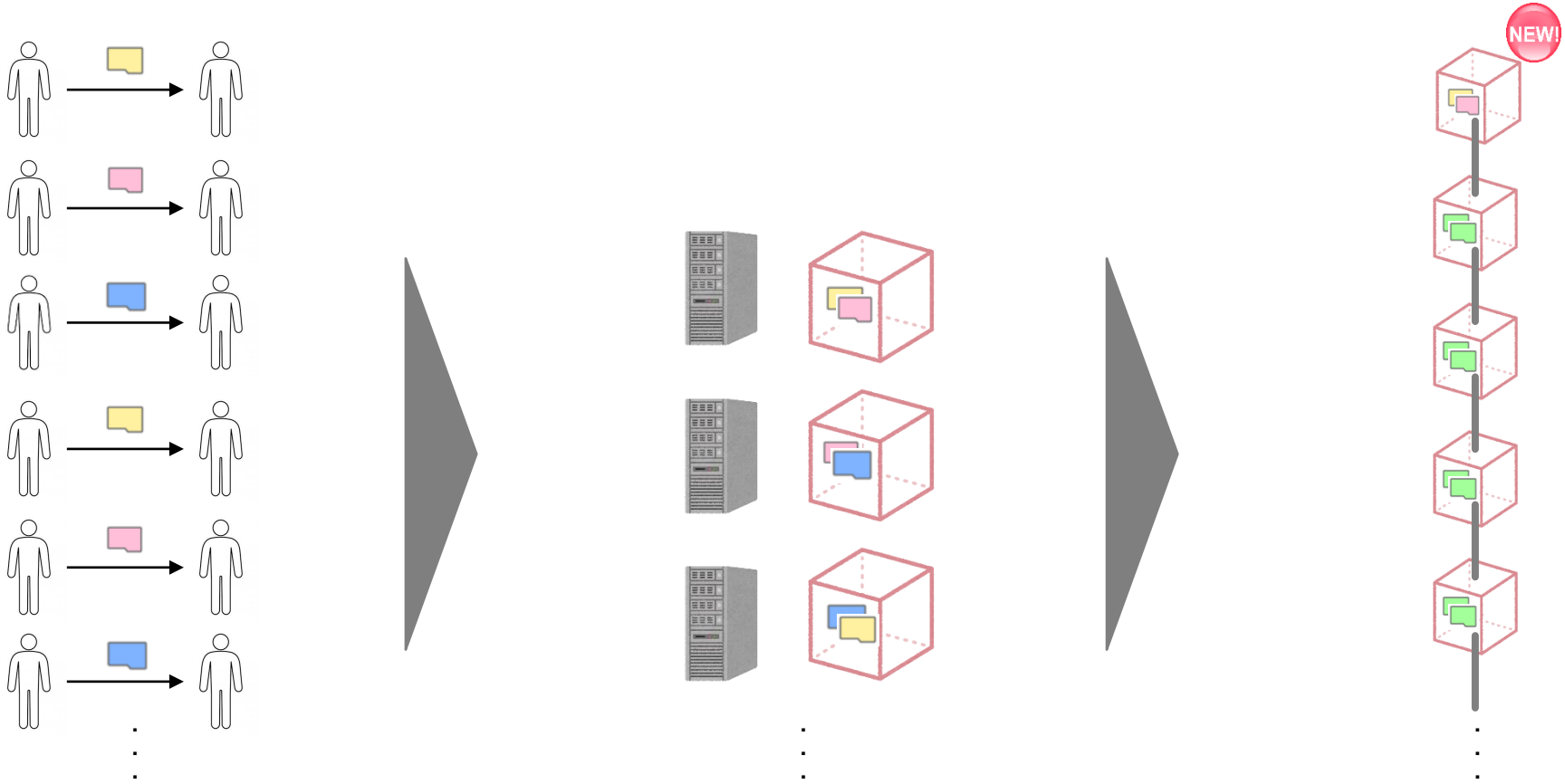
用語の定義

❑ 押さえておきたい主な用語を以下に整理。噛み砕いて表現した「ざっくり理解」のイメージだけ持っていればひとまず困ることはない。（注：以降、しばらくビットコインのブロックチェーンについて見ていきます。）

	用語	本稿での定義	ざっくり理解
1	ブロックチェーン	✓ ビザンチン障害を含む不特定多数のノードを用い、時間の経過とともにその時点の合意が覆る確率が0へ収束するプロトコル、またはその実装	✓ ブロックの積み重ね ✓ 改竄が難しい性質を持つ
2	ブロック	✓ 自身と直前のブロックを説明する情報並びにいくつかのトランザクションを主な成分として含む構造体	✓ トランザクションの集まり
3	トランザクション	✓ あるユーザから他ユーザへの価値の移転を主な成分として含む構造体	✓ 何らかの移動（今回は特に送金）
4	P2Pネットワーク	✓ 不特定多数の「ノード」からなる ネットワーク	✓ ノードがつながったもの
5	ノード	✓ ブロックチェーンにおいてブロックの生成やメッセージの伝達等を含む複数の役割を担うもの	✓ 縁の下の力持ち、便利屋さん
6	ビザンチンなノード	✓ 作為か不作為かによらず偽の情報を伝達する可能性のあるノード	✓ 体調不良の便利屋、悪徳便利屋
7	コンセンサスアルゴリズム	✓ 複数のノードで構成されるP2Pネットワークがある値について合意するためのルール	✓ 大切な手続き

ブロックチェーンに1つブロックがつながるまでのざっくりした流れ

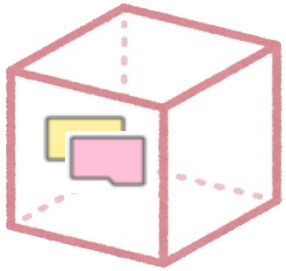
- ❑ たくさんのユーザがトランザクションを発行する。
- ❑ 複数のノードがそれらのトランザクションを各々自由に見繕ってブロックにまとめる。
- ❑ そのうちの1つだけが有効とされてブロックチェーンにつなげられる。



理解：なるほど！トランザクションを発行するとブロックにまとめられて、そのうちの1つがブロックチェーンにつながるんだね！
疑問1：あれ？でも、「つながる」ってどういうことだろう。URLのリンクでも貼ってるのかな？（⇒p6）
疑問2：あれ？どうやってたくさんあるブロックの中から1つを決めるのかな？（⇒p7）

ブロックの構造

- ブロックの構造を表に整理。
- ブロックヘッダ内にある「親ブロックのハッシュ」により、親ブロック（前のブロック）につながる。



＝

これが前のブロックの参照。
つまり、前ブロックの表2の情報
を要約したデータを持っている。 →

表1 ブロックの構造

フィールド名	説明	データサイズ
マジックナンバー	0xD9B4BEF9固定、ネットワークを識別	4バイト
ブロックサイズ	ブロックヘッダ～トランザクションまでのブロックサイズ	4バイト
ブロックヘッダ	ブロックのヘッダ情報	80バイト
トランザクションカウンタ	ブロック内のトランザクション数	1-9バイト
トランザクション	トランザクションのリスト	可変

表2 ブロックヘッダの構造

フィールド名	説明	データサイズ
バージョン	ブロックのバージョン	4バイト
親ブロックのハッシュ	親ブロックのブロックヘッダをハッシュ化したもの	32バイト
マークルルートハッシュ	ブロック内の全トランザクションをハッシュ化したもの	32バイト
タイム	ブロック生成時のタイムスタンプ	4バイト
ビット	ブロック生成時のプルーフオブワークの難易度	4バイト
ナンス	0から始まる32ビットの数字	4バイト

※ハッシュとは、あるデータを不可逆的に要約したもの。少しでも元データが変わると、ハッシュも全く異なるものになる。右にSHA-256でハッシュ化した例を示す（戻り値は32バイト→64文字）。

SHA-256(ccatak) ⇒BEBEFB94BE7D44DE87492B704DDCD2B0AD80C900C0E53E815918424175C9B37B
SHA-256(cca_tak) ⇒584BB788DAF7CD8E3AC91C080B239470F300D125F13D3CFB9F3BF84BDF930888
SHA-256(cc_atak) ⇒A39C2D004DA56DA16DAE065075393855CFEC8035BCA82105E85F8D74D64DA421
SHA-256(https://cc-res.com/) ⇒7CCCEBBB93B9096DF9B9B45657889D12C4414CE4788BEAC6264FC1FC93253D6A

理解：なるほど！前ブロックのブロックヘッダを要約したデータを持つことでつながるんだね！元のブロックヘッダの情報を直接持つよりもデータサイズはより小さくなる(80バイト⇒32バイト)し、エコだね！

ブロックの選び方(プルーフオブワーク)

❑ ブロックヘッダ内にある「親ブロックのハッシュ」、「マークルルートハッシュ」、「ナンス」の3つをハッシュ化して予め決められている値より小さな値を見つける競争を行う。最初に見つけたノードがブロックを生成できる。

ネットワーク

ネットワーク：
0が最初から17個続くハッシュを見つけたノードを勝ちとする。

表2 ブロックヘッダの構造

	フィールド名	説明	データサイズ
全ノード共通 →	バージョン	ブロックのバージョン	4バイト
全ノード共通 →	**親ブロックのハッシュ**	親ブロックのブロックヘッダをハッシュ化したもの	32バイト
ノードごとに異なる →	**マークルルートハッシュ**	ブロック内の全トランザクションをハッシュ化したもの	32バイト
	タイム	ブロック生成時のタイムスタンプ	4バイト
	ビット	ブロック生成時のプルーフオブワークの難易度	4バイト
試行回数ごとに異なる(1ずつ増やしていく) →	**ナンス**	0から始まる32ビットの数字	4バイト

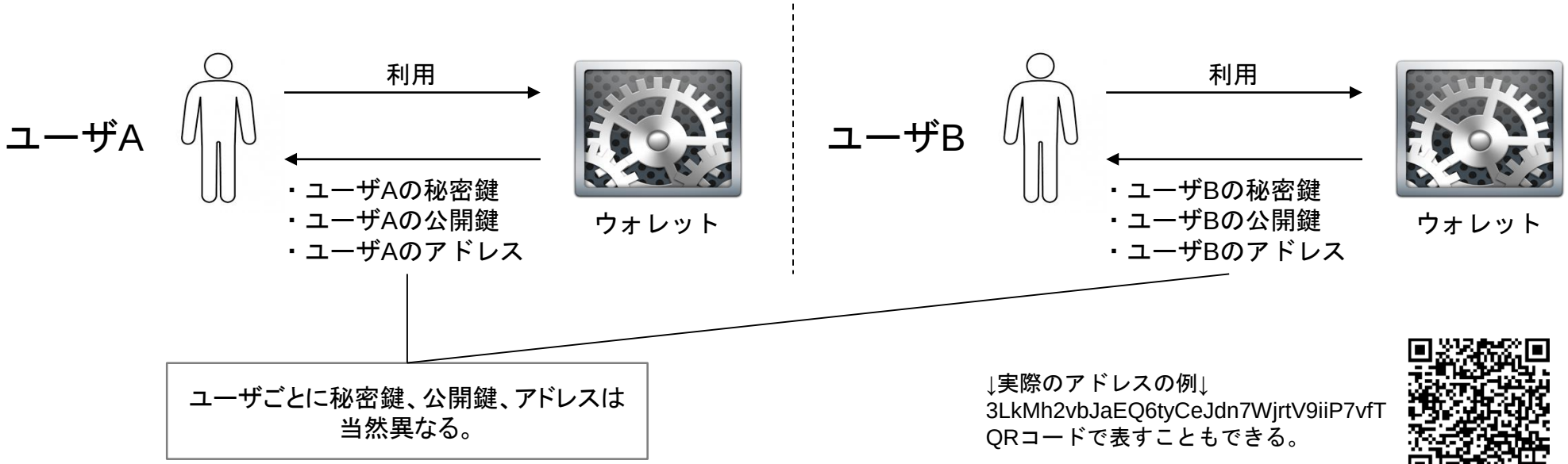
	Proof-of-Work	1回目	2回目	…	740825940回目
ノードA	親ブロックハッシュ	DFA73B59D0…	DFA73B59D0…		DFA73B59D0…
	マークルルートハッシュ	3D3BEED2BA…	3D3BEED2BA…		3D3BEED2BA…
	ナンス	0	1		740825940
	結果	14045AA295DD2… ✖	954088627AAD… ✖		00000000000000000F… ○
ノードB	親ブロックハッシュ	DFA73B59D0…	DFA73B59D0…		DFA73B59D0…
	マークルルートハッシュ	859DD5DB85…	859DD5DB85…		859DD5DB85…
	ナンス	0	1		740825940
	結果	A7845095F0090… ✖	0008DDFAFACC… ✖		96540FCCDBB… ✖
ノードC	親ブロックハッシュ	DFA73B59D0…	DFA73B59D0…		DFA73B59D0…
	マークルルートハッシュ	9AC21989CD	9AC21989CD		9AC21989CD
	ナンス	0	1		740825940
	結果	004357FFADCD… ✖	54829DDF44C8D… ✖		ADF000890470… ✖

※実際の計算速度はハッシュパワー依存なので同じ時間当たりの計算回数はノードごとに異なります。Aが100回計算している間にBは10000回計算しているということもあります。

理解:なるほど！最初に条件を満たす小さなハッシュを見つけたノードだけがブロックを生成できるんだね！
疑問3:あれ？でも、トランザクションってどうやって相手に届くの？ブロックに取り込まれて、ブロックチェーンに刻まれたけど、まだ相手に届いてないよね。というかそもそも相手ってどうやって分かるんだっけ？(⇒p8)

ユーザの識別：アドレス

- ユーザはアドレスで識別する。
- アドレスは公開鍵をもとに作られる。公開鍵は秘密鍵から作られる。
- ウォレットというソフトウェアを使うと秘密鍵・公開鍵のペアとアドレスを作ってくれる。



（補足）
かなりテクニカルになるが、仕組みを熟知していれば自分でアドレスを作ることも可能。
・1より大きく2の256乗（厳密にはもう少し小さいが）より小さな正の整数を秘密鍵として選ぶ。十分に大きな数を推奨。
・楕円曲線Secp256k1($y^2 = x^3 + 7 \mod 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$)上の座標である生成元G（Gx, Gy）を秘密鍵のスカラー倍して公開鍵(x, y)を演算。
・公開鍵(x, y)をもとにして、SHA-256でハッシュ化し、その結果をRIMEPD-160でハッシュ化し、その結果をBase58エンコードするとアドレスが完成。
※ウェブで検索するとコード付きで作り方が載っていたりするので、作ってみたい方は調べてみてね！

理解：なるほど！アドレスでユーザが分かるんだね！
疑問4：あれ？でも、そのユーザが持っている所持金ってどうやって分かるの？それが分からないと送金できないよね。銀行みたいに、アドレスが口座のような役割を果たすの？（⇒p9）

トランザクションの構造

- アウトプットトランザクション（Tx-out）では、金額と、相手先の情報と相手が利用するための条件（ScriptPubKey）を記述する。
- インプットトランザクション（Tx-in）では、自分宛てに送られた未使用のTx-outを指定し、そこに記載された利用条件を満たすよう情報を提示する（ScriptSig）。

トランザクションの構造

フィールド名	説明	データサイズ
バージョンナンバー	バージョン	4バイト
フラグ	あれば0001で、Wウィットネスがあることを示す	あれば2バイト、なければ0バイト
インプットカウンタ	Tx_inの数	1-9バイト
インプットリスト	Tx_inのリスト	Tx_inの数による
アウトプットカウンタ	Tx_outの数	1-9バイト
アウトプットのリスト	Tx_outのリスト	Tx_outの数による
ウィットネス	あれば1インプットにつき1ウィットネス	可変
ロックタイム	トランザクションのロックが解除されるブロック高かタイムスタンプ、ロックなしの場合は0	4バイト

インプットトランザクション(Tx_in)の構造

フィールド名	説明	データサイズ
前Tx_outのハッシュ	このインプットが参照している前Tx_outのハッシュ	32バイト
前Tx_outのインデックス	このインプットが参照している前Tx_outのインデックス	4バイト
入力者のスクリプト長	入力者の署名スクリプトの長さ	1-9バイト
入力者の署名スクリプト（ScriptSig）	入力者を確認するためのスクリプト、入力者の秘密鍵による電子署名と公開鍵を含む	可変
シーケンス	送信者が定義するトランザクションのバージョン	4バイト

アウトプットトランザクション(Tx_out)の構造

フィールド名	説明	データサイズ
値	トランザクションの送金額	8バイト
出力先のスクリプト長	出力先のスクリプトの長さ	1-9バイト
出力先のスクリプト（ScriptPubKey）	出力先の公開鍵を用いて値を引き出すための条件を記述したスクリプト	可変

（補足）
①あるユーザAがユーザBに送金するとき、Tx-outに送金額を含むユーザBの情報を記述する。Tx-inに過去に誰かからユーザAに送られた未使用のTx-outを指定して送金に十分な金額を引き出す。
②トランザクションも別のトランザクションを参照するような構造をとっている。各ユーザのその時点の所持金の計算は、過去のトランザクションを遡って行う。
③履歴を辿り計算するのではなく、銀行口座のようにアカウントを設けて管理しているブロックチェーンもある。

理解：なるほど！トランザクションもブロックみたいに参照構造を持っていて、それを使って送金したり、履歴を辿って計算して所持金を求めたりするんだね！
疑問5：あれ？それなら既存の決済システムは全部やめちゃってブロックチェーンに乗り換えればいいんじゃないの？（⇒p10）

スケーラビリティの問題

- ブロックチェーンの主な課題の一つがスケーラビリティ（スケーラビリティ問題）。これはビットコインに限った話ではない（ので、これ以降は多くのブロックチェーンを対象とする話です）。
- ブロックチェーンの初歩的な設計で捌けるトランザクションの数は少ない。最大限処理できるトランザクション量も多くないので、ユーザの数が増え、トランザクションの数が増えれば増えるほど、捌けずに残るトランザクションが多くなってしまう。結果、実用的でなくなる。

	比較対象	平均スループット	最大スループット	備考
ブロックチェーン	ビットコイン	2.6tps	8.3tps	平均スループットは2017/10/22～2018/10/21のデータについての平均値。 最大スループットはブロックサイズ1MB、平均トランザクションサイズ200B、ブロックタイム600秒での試算値。
ブロックチェーン	イーサリアム	8.2ps	20tps	平均スループットは、2017/10/22～2018/10/21のデータについて平均を算出。 最大スループットは[1]による。
非ブロックチェーン	paypal	240tps	不明	平均スループットは2017年に関するpaypal発表による。
非ブロックチェーン	VISA	1736.1tps	56000tps	平均スループットはVISA HPのデータをもとに算出した値。 最大スループットはVISAがIBMと実施した2014年の実験値による。

※[1] https://www.ethnews.com/the-raiden-network-could-allow-instant-transactions-in-ethereum

例えばビットコインとVISAを比べてみると、平均スループットで約670倍、最大スループットで約6700倍の差がある。
つまり…
・VISAが1秒で捌く平均的なトランザクション量をビットコインは約10分かかって捌いている。
・VISAがフルパワーで稼働しなければならないような状況下で考えると、VISAが1秒で捌くトランザクション量をビットコインは約2時間かけて捌くことになる。

理解：なるほど！確かに現在の状況ではユーザ数の増加やトランザクションの増加に対応しきれず困ってしまうね！
疑問6：なら、どうしたら解決（スケーリング）することができるだろう？ブロックチェーンの性能を上げるように設計すればいいのかな？（⇒p11、12）

スケーリング

- ブロックチェーンのスペックを上げる方法（オンチェーンスケーリング）とブロックチェーンの仕事を切り出す方法（オフチェーンスケーリング）がある。

単純な構図としては…

ブロックチェーンが処理できる量 

ブロックチェーンに入ってくる仕事量

ブロックチェーンが処理できる量  ブロックチェーンに入ってくる仕事量

となっていればよい。

従って解決策としては…

①ブロックチェーン自体が処理できる量をなんとかして増やす＝オンチェーンスケーリング
②ブロックチェーンに入ってくる仕事量をなんとかして減らす（ユーザの数を減らすのでは元も子もないので、仕事をアウトソースするイメージ）＝オフチェーンスケーリング

の2方向がある。

スケーリング

- オンチェーンスケーリングはノードのリソースを勘案する必要がある。オフチェーンスケーリングはそもそも技術自体を作る必要がある。

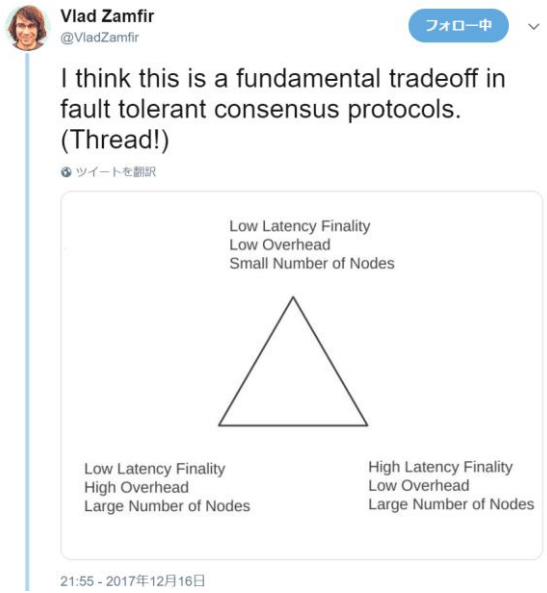
用語	オンチェーンスケーリング	オフチェーンスケーリング
考慮しないといけないこと	✓ ブロックチェーンを維持してくれているノードの回線、ストレージ、CPU、メモリ等のリソースを圧迫する。そのためこの方法によるスケーリングは限界があるし、段階的なものになりやすい。	✓ ブロックチェーンを切り出した先で処理する技術自体を開発する必要がある。 ✓ ブロックチェーンだけでなく、その技術が前提とするやり方やセキュリティ要件などの条件を受け入れる必要がある。
手法の一例	✓ ブロックサイズを増やす。 ブロックサイズを2倍にすれば平均的に2倍のトランザクションを含めることができるようになる。 ✓ ブロックタイム（ブロックの生成時間）を減らす。 ブロックタイムを1/2にすれば、1/2の時間で同数のトランザクションを処理できるようになる。 ✓ コンセンサスアルゴリズムを速いものに変更する。 ✓ Etc…	✓ ペイメントチャネル（ステートチャネルの応用） ブロックチェーン外に開設した2者間のチャネルでやり取りを行うもの。基本的にその2者間に限っては、ペイメントチャネルを開くときと、ペイメントチャネルで行ったやりとりを反映するときの2回のトランザクションだけをブロックチェーンで処理すればよい。 ✓ サイドチェーン（別のブロックチェーンを作って、そちらで諸々の処理を代替させるもの） 双方向にやりとり可能な（2way-pegの）ブロックチェーンを作る。新しく作るブロックチェーンは、pegされるブロックチェーンの設計に拘らずに設計できるので自由度が高い。スケーリングの観点から、サイドチェーンでは高速なコンセンサスアルゴリズムが用いられやすい。 ✓ Etc…

理解：なるほど！スケーリングはブロックチェーンの中でやる方法とか外でやる方法があって、一長一短があるんだね！
疑問7：あれ？でも、それなら初めから高速コンセンサスアルゴリズムを使えばいいんじゃないの？（⇒p13）

高速コンセンサスアルゴリズムのトレードオフ

- 単純に処理速度を追求する場合、高速コンセンサスアルゴリズムを採用するというのは全く間違っていないし、むしろトレンドでもある。数千tpsを記録するブロックチェーンも出てきている。
- ただし、高速コンセンサスアルゴリズムを採用することで、ネットワークの負荷が増えたり、中央集権的になったりするなど、トレードオフが存在する。
- いい方向に考えれば、利用するブロックチェーンを目的や好みに応じて「選択できる」ほどに充実してきたということでもある。

Vlad Zamfir氏



出典：https://twitter.com/VladZamfir/status/942271978798534657
の一連のツイートの冒頭。

だいたいの解釈

- 多数のノードにより非中央集権性を確保しつつネットワーク負荷を減らすと、ファイナリティは遅くなる（合意までに時間がかかる）
- ファイナリティを高速化してネットワーク負荷も減らそうとすれば、必然的にノードの数を少なくせねばならず非中央集権性が失われる。
- ファイナリティを高速化しつつ非中央集権的であろうとすれば、ネットワークが行わなくてはならない仕事量は膨れ上がる。

頑張って追求できるのは2つまで

理解：なるほど！高速化目的で速いコンセンサスアルゴリズムを採用してもトレードオフがあって別の部分で問題が発生したりするかもしれないんだね！
疑問8：いろいろなブロックチェーンがあるのは分かったけど、結局どういうところに利用するのさ？（⇒p14）

ブロックチェーンの特徴（主に機能面）

❑ 機能面の主な特徴として、下記に挙げるようなものが考えられる。

特徴	説明
1 記録のテクノロジであるが物理的に何かを生成する技術ではない	✓ ブロックチェーンは基本的には何らかのデータ、並びにデータの変化を記録する技術 ✓ 物理的な何かを生成する技術の代替は不可（当然車やPC等を制作することはできない）
2 電子データの生成やロジックの記述ができるが大きなデータや重たいロジックは扱えない	✓ スマートコントラクトを利用して電子的なデータの生成やデータの処理ロジックを記述可能 ✓ ノード側のリソースの問題から大量のデータを書き込んだり重いロジックを実行することは困難
3 透明性と追跡性があるがプライバシーの心配がある	✓ トランザクションは全て記録され、誰でも閲覧できる状態に置かれており高い透明性と追跡性があり不正がないことを証明可能 ✓ ひと度現実のアイデンティティと紐づくとブロックチェーンの解析によりかなりのプライバシー情報を推測可能
4 耐障害性があるが設計に注意が必要	✓ ノードにブロックチェーン自体を持たせることで、ネットワークに障害が発生しても正常に動作 ✓ ブロックチェーンを維持するノードのふるまいやネットワーク分裂時の対応などに関して繊細な設計が必要
5 入力を正しいものとして処理するが入力自体の正しさを保証しない	✓ 入力を正しいものとして処理し、記録し、正しく処理されたことを保証 ✓ 入力自体が正しいかどうかについては保証不可（例えばX⇒Yの送金トランザクションがあったとして、本当に送信者がXさんか分からない）
6 改竄耐性を持つがロールバックができない	✓ コンセンサスアルゴリズムに基づいてファイナライズによる高い改竄耐性 ✓ トランザクションの内容に間違いがあったり後から変更を加えたい場合でもブロックに取り込まれると後戻り不可
7 非中央集権により検閲耐性を得るが多くのステークホルダーが絡む	✓ ブロックチェーンは中央で管理する主体を持たず、プロトコルに基づきネットワークが維持 ✓ プロトコルの更新等を行う場合、開発者だけでなくネットワークに参加しているノードの合意を得る必要があるなど、多数のステークホルダーが関係し改修が容易でない側面がある。

メリットかデメリットかは文脈依存

＝

メリットがデメリットを上回る箇所で使えるとよい(界隈としては現在模索中といったところ)

※より汎用的かつ抽象的に特徴をまとめるならば、最近のブロックチェーンには**「電子データの生成とロジックの記述」「データとデータ遷移の記録」「データとデータ遷移の追跡性」**が備わっている。これらはかなり普遍的なので様々な分野に応用できるポテンシャルを秘めている。

例：CryptoKitties

- Ethereumブロックチェーン上に作られたゲーム。ネコの収集、育成、配合、譲渡（売却／購入）などができる。
- ネコの所有権の移転や新たな権利の作成等がスマートコントラクトにコーディングされている。トランザクションのin-outが分かるので、いま誰にネコの所有権があるのか、これまで誰から誰に移ってきたのか、ネコの売買が正しくなされたのかなど全て改竄なく把握できるなどのメリットがある。
- 逆に、ブロックチェーンを利用しているので、ユーザが熱狂するなどして大量にトランザクションが発行されるとネットワークを圧迫するという問題もある（というか問題になっている）。

CryptoKittiesってなんだにゃー？

CryptoKittiesは育成、収集、CryptoKittiesと呼ぶ超かわいい生き物を中心としたゲーム！それぞれのネコはオリジナルで、あなたが100%所有！複製・排除・破壊をすることはできないよ。

↓ブロックチェーン上のゲームCryptoKittiesの↓
コントラクト（ソースコード）の一部

```
contract KittyCore is KittyMinting {

    address public newContractAddress;

    function KittyCore() public {
        paused = true;
        ceoAddress = msg.sender;
        cooAddress = msg.sender;
        _createKitty(0, 0, 0, uint256(-1), address(0));
    }

    function setNewAddress(address _v2Address) external onlyCEO whenPaused {
        newContractAddress = _v2Address;
        ContractUpgrade(_v2Address);
    }

    function() external payable {
        require(
            msg.sender == address(saleAuction) ||
            msg.sender == address(siringAuction)
        );
    }

}
```

↓左記ソースコード実行のために発行されたトランザクションの一部↓

TxHash	Block	Age	From		To	Value	[TxFee]
0x589ba976ec58e6...	(pending)	7 secs ago	0x818372c5cd6e18...	IN	0x06012c8cf97bead...	0 Ether	(Pending)
0x8186c836c47331...	(pending)	16 secs ago	0xb358bf9c59c3f83...	IN	0x06012c8cf97bead...	0 Ether	(Pending)
0x4ddd771c49e92fb...	(pending)	11 mins ago	0x9df67a393fc8502...	IN	0x06012c8cf97bead...	0.00893726749856492 Ether	(Pending)
0xf59d5c1674cfb66...	(pending)	14 mins ago	0x52203a1aab008b...	IN	0x06012c8cf97bead...	0 Ether	(Pending)

↓左記ソースコードから発行されたトランザクションの一部↓

ParentTxHash	Block	Age	From		To	Value
0x4c96baae6f65396...	6652197	4 mins ago	0x06012c8cf97bead...	→	0x000007222caeb2...	0.008 Ether
0x6214cc30ae9a60...	6652194	5 mins ago	0x06012c8cf97bead...	→	0xe48048be88e725f...	0.008 Ether
0x9dd90fdd326e807...	6652191	6 mins ago	0x06012c8cf97bead...	→	0xe45ec6b57d9bf1f...	0.008 Ether
0xaa7995c6ae06eb...	6652174	10 mins ago	0x06012c8cf97bead...	→	0xc7af99fe5513eb6...	0.001 Ether

※CryptoKittiesのコントラクトのアドレスは「0x06012c8cf97BEaD5deAe237070F9587f8E7A266d」

CryptoKittiesの例のように、一つの方向性としてブロックチェーンは権利関係の扱いに長けているのではと考えられている（p14の②ロジックの記述、③透明性＆追跡性、⑥改竄耐性などの特徴と相性がよい）。

（補足）
その他、サーバメンテが要らないこと、ユーザの個人情報を管理しなくてよいこと等はメリットと言えるだろう。一方で、イーサリアムは一度デプロイすると変更がきかない（ブロックチェーンに刻まれるので）こともあり、バグには細心の注意を払う必要がある点はデメリットと言える。

まとめ

ブロックチェーンについて

- 原義的には暗号理論を基盤にP2Pネットワークを組合わせて不可逆性を得ることで「信頼できる第三者機関」を排する決済システムだった。（Nakamoto論文）
- ブロックの生成に”労力”をかけさせること、そのブロックを次々に参照していく構造をとること、最長のチェーンを正とすることで改竄を困難にする。

スケーラビリティ問題について

- ブロックチェーンの初期設計では増加するユーザやトランザクションに対応できない側面があった。
- オンチェーン・オフチェーンのそれぞれのスケーリング手法で対応している。
- 高速コンセンサスアルゴリズムを採用したサイドチェーン（オフチェーンスケーリング）によりトレードオフはあるが既存の決済システムと遜色ないスループットをマークすることもできるようになった。
- ブロックチェーンにも多様性が生まれている。

特徴と展望

- 「データ生成とロジック記述」「記録・保存」「透明性・追跡性」といった特徴は普遍的で様々な分野に応用できるポテンシャルがある。
- 例として、権利関係は相性がよさそうである。

ブロックチェーンの課題と解決策の進展に基づく展望の考察

ccatak / https://cc-res.com**管理人**
Email: ccatak@cc-res.com
Website: https://cc-res.com

講演資料の公開場所：https://cc-res.com/dsgws_ccatak_materials_20181119

おわり