



ビットコインの使い方としくみ

ウォレット、トランザクション、ブロックチェーン

暗号通貨ユーザー会 村田 智英



村田智英(むらた ともひで)

- ソフトウェアエンジニア



@Derorisan



<https://www.facebook.com/tomohide.murata.1>

主にWEB系のバックエンドのエンジニアです。

好きなもの: D言語, perl, emacs, fedora

嫌いなもの: PHP, Oracle

CoinTip作りました

 <https://cointip.jp/>

- ツイッターアカウント宛にビットコインを送る送金サービス
- ビットコインとツイッターアカウントを持っていればOK
- 勉強会のスピーカーへのお礼や、有名人などへの投げ銭に使う目的



それでは本題

ビットコイン使ったことある人？



アジェンダ

- ビットコイン概要
- 使い方
 - ウォレットの作り方
 - 送金の仕方
- しくみ
 - ウォレット
 - トランザクション
 - ブロック(チェーン。)
 - マイニング

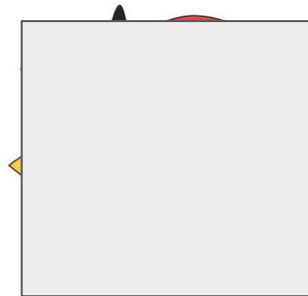
ビットコインとは

中央集権でない「通貨」

- P2Pネットワークによって構成されている。
 - ということはだれもメインのサーバにはならない。
- ただの通貨だと思ってok
 - 使う側から見たらUSDドルやユーロなどと同じ。
 - 為替レートもある。
 - 印刷すれば紙やコインにもなる。
 - 違うのはどこかの国が発行したものではなく、コントロールしている人もいない。

オープンソース

- 2008年の終わりにサトシ・ナカモトによってホワイトペーパー(論文的な)がネットに出る。
 - こんな感じのしくみにすれば中央集権でない通貨が作れるんじゃないか？的なもの。
 - 分散コンピューティングの未解決問題の現実的な解決方法 (ビザンチン将軍問題)
 - proof of workで中央機関を必要としないしくみ
- 実際のホワイトペーパー—————
 - <https://bitcoin.org/bitcoin.pdf>



2009年には稼働し始めた。

- 2009-01-03 18:15:05 に初のブロックが作られた。
 - <https://blockchain.info/ja/block-height/0>
- リファレンス実装はすべてgithubで公開されている。
 - <https://github.com/bitcoin/bitcoin>
 - 最初のころはサトシさんもコントリビューターだったがもう引退済み
 - ちなみにサトシさんはマジで誰だかわからない。
 - 2014年より前はsvn sourceforge
 - C++
 - この実装を参考にして java や js, python などで実装する人も！

単位

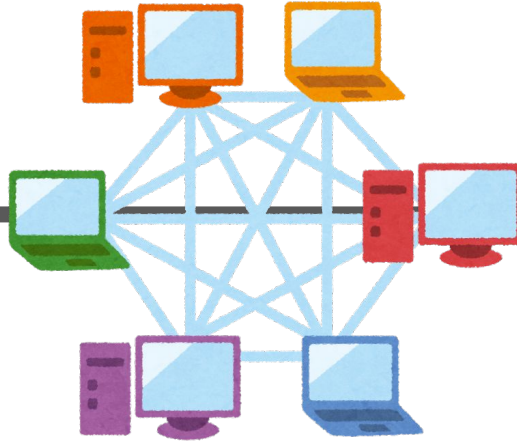
- 「円、銭」とか「ドル、セント」とかみたいに単位がある。
- ビットコインでは
 - satoshi (最小単位)
 - bits(μ BTC) = 100 satoshi
 - mBTC = 100,000 satoshi
 - BTC = 100,000,000 satoshi (一億サトシ)

送金のフロー

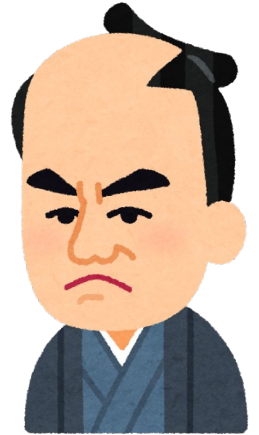
アリスさん



いんたーねっと。



ボブさん



とりあえず使うのに必要なもの

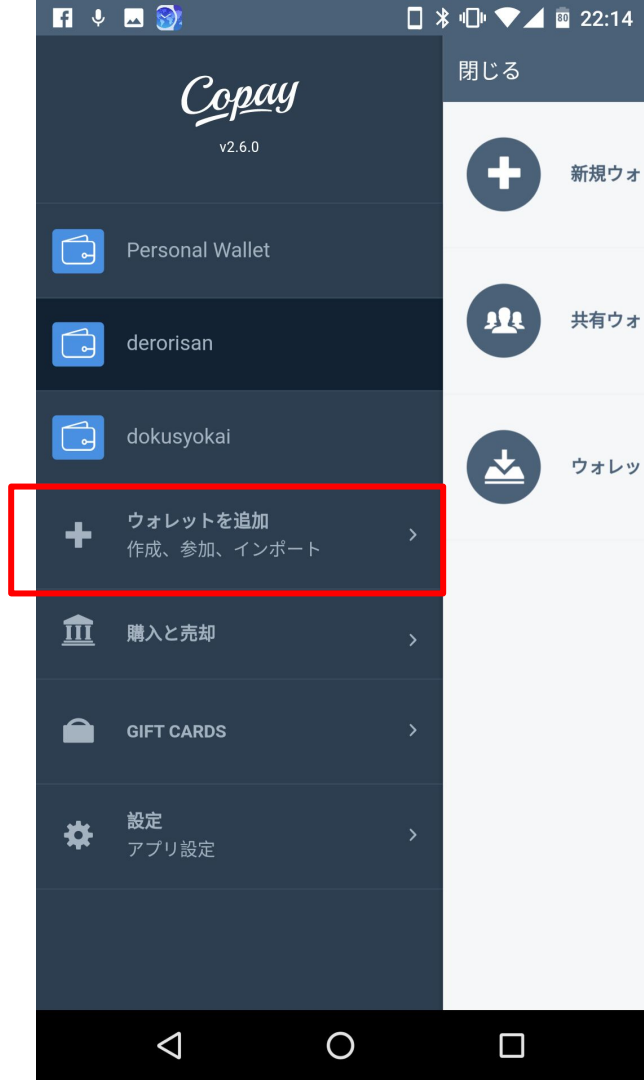
- ビットコインウォレット
 - 自分のビットコインの財布
 - アカウントみたいなもの
- ウォレットに入れるビットコイン
 - 誰かからもらう
 - 取引所などで円で買う
 - マイニングがんばる。

ウォレットを作るときのユーザーフロー

立ち上げて

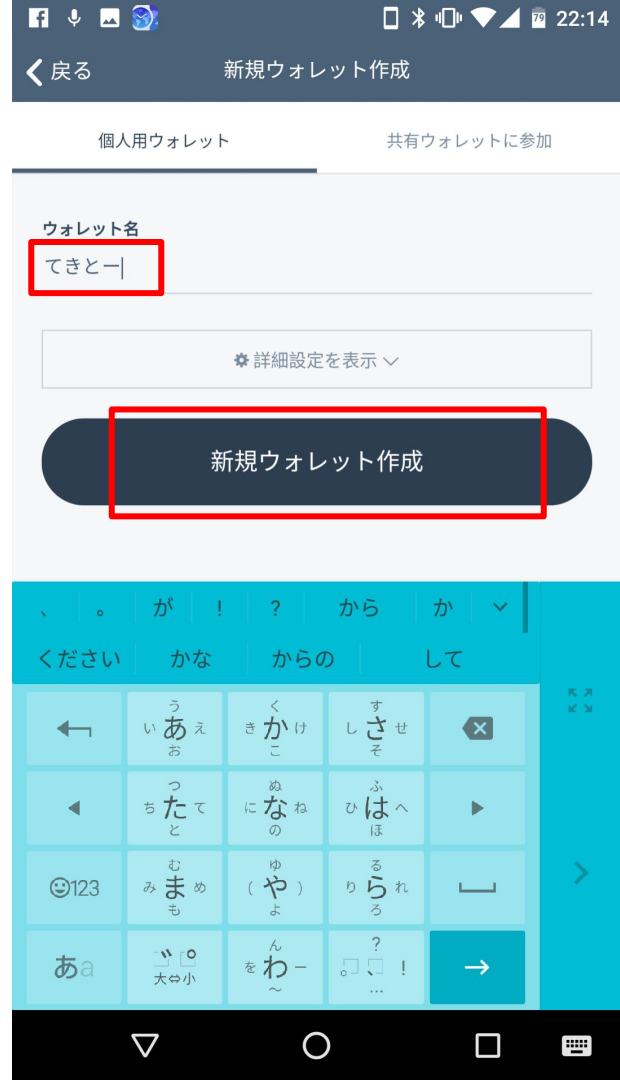
メニューから「ウォレットを追加」

画面はAndroid版copayです。



新規ウォレット作成

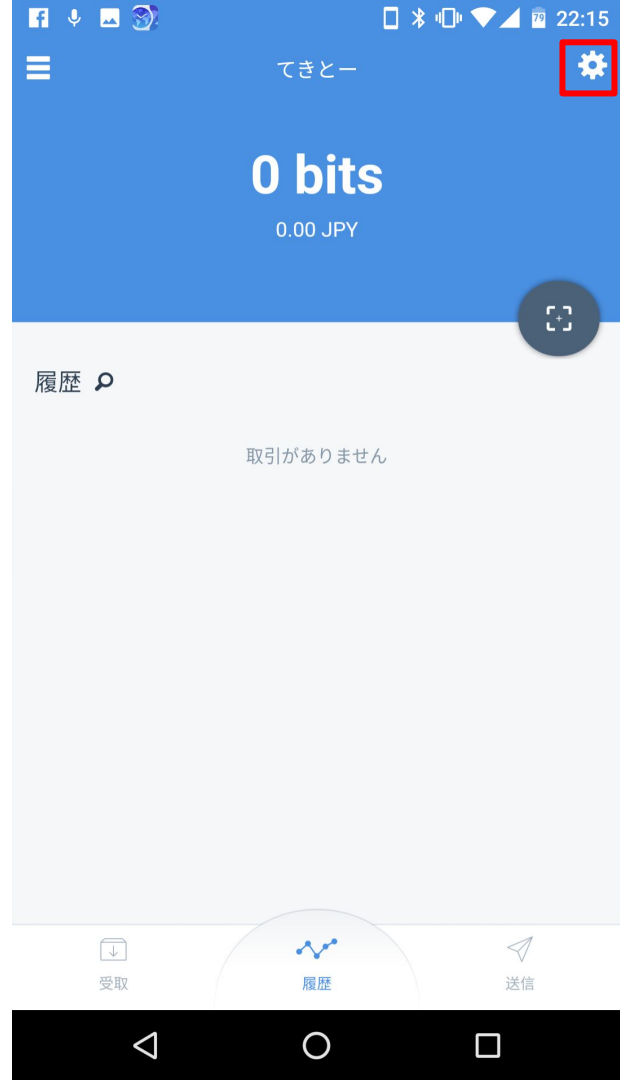
- ウォレット名を適当に入れる。
- 公開されるものではないので自分がわかるものを。



できあがり

ウォレット作るだけならタダだぜ！

いくらでもどうぞ。



バックアップを取りましょう

右上の歯車から

「バックアップ」



パスフレーズを書き留める

- スクリーンショットは避ける(画像が流出したら終
- この単語の組み合わせでこのウォレットの証明書に
 - ということは漏れると他の人が同じウォレットを持てる
- 使われる単語はBitcoinで決まっているよ
 - 間違えにくい単語になっている

てきとー



復元フレーズを書き留めて下さい

この個人用ウォレットを復元するには復元フレーズが必要です。紙などに書き留めておき、安全な場所で保管して下さい。

さんそ あやまる しらせる まんきつ
まとも せいじ ひたる だっこ つねづね
ひみつ げどく おおう

続ける

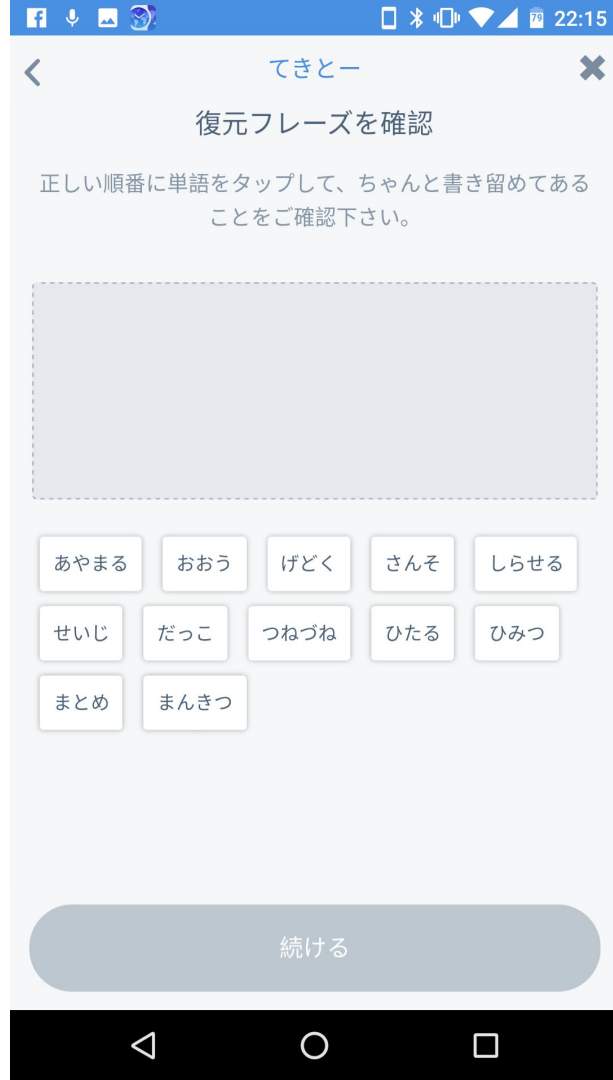
正しいか確認！

メモしたものが間違っていたら

アプリを消してしまったときなどの復元できない！

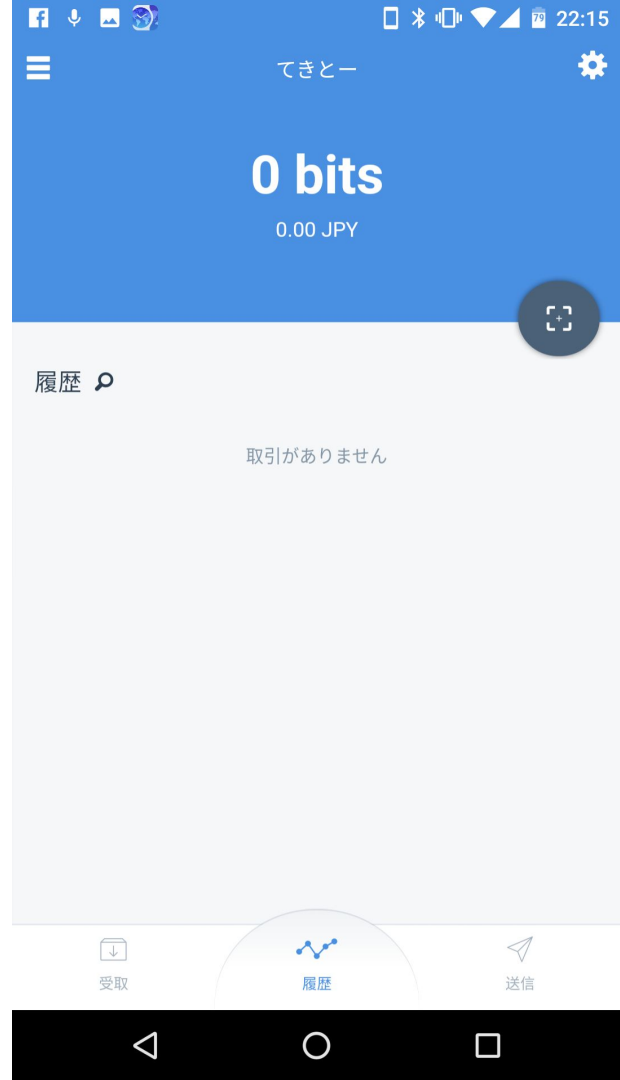
画面はcopayのものなので

別のウォレットでは入力させる場合も。



ほいきた

財布を買うためにお金はいらない。



これだけ。

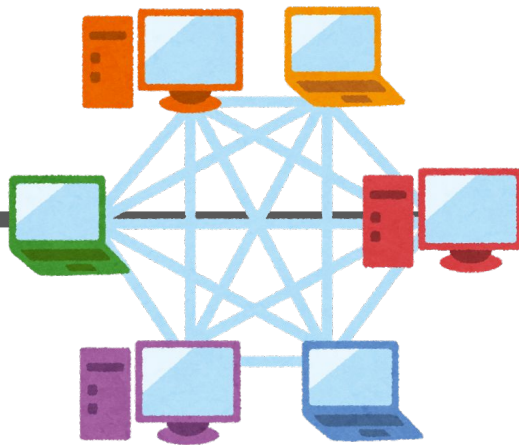
ウォレット

アリスさん



ウォレット

いんたーねっと。



ボブさん



ウォレット

送金するときは？

立ち上げて

あらかじめ少し入れておきました。

手に入れる方法は後ほど

下の「送信」



宛先と金額

宛先(後述)と金額を入れます。

- 宛先は送金先のビットコインアドレス
- 金額はこの場合はbits
 - ウォレットによってBTCで入れたりYenが使えたりします。
 - 最終的にはビットコイン換算 (satoshi)になります
 - **YENは直接送れませんよ。(当たり前かw)**



宛先(ビットコインアドレス)

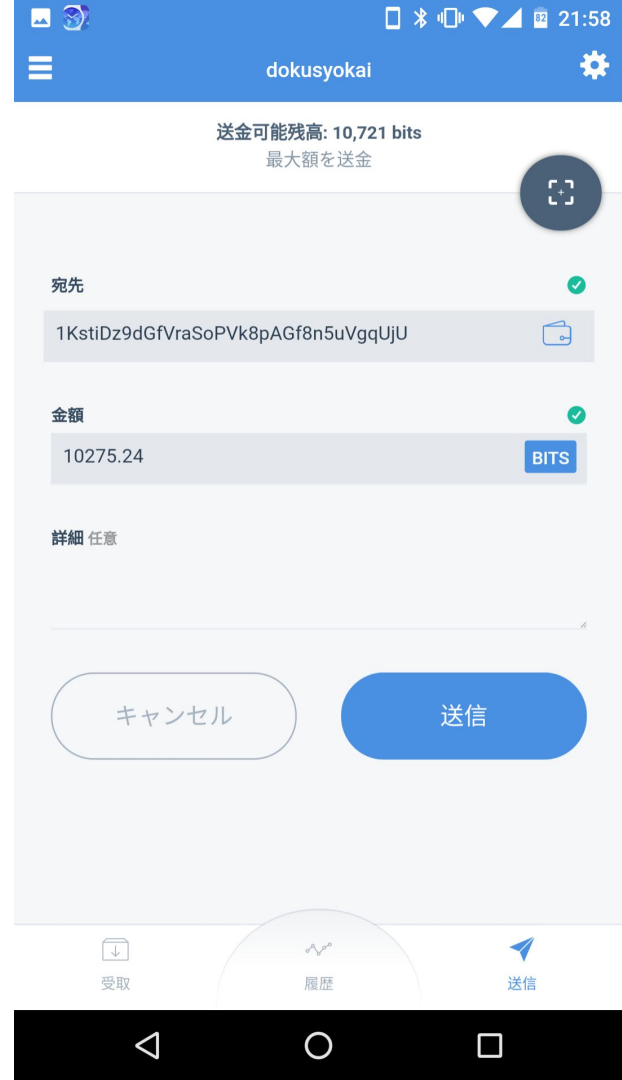
- 赤線で囲んである部分
- ウォレットに紐づく宛先のアドレス
- 1つのウォレットでいくらでも作れる。
- 使い回すこともできるし毎回作ることもできる。
- QRコードで表示させてあげるのがよく見るかたち
- URIのフォーマットが決まっている。
 - bitcoin:<address>[?amount=<amount>][?label=<label>][?message=<message>]



送金画面に戻ります。

- 宛先と
- 金額を

入れて「送信」

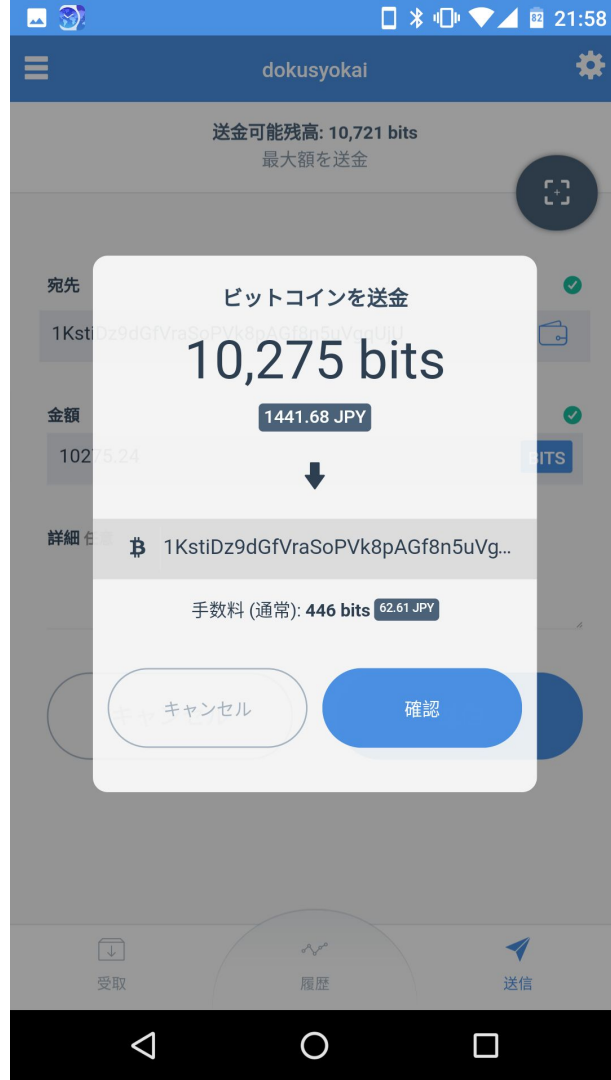


確認画面

手数料とは

- トランザクション手数料(transaction fee)
- マイナー(後述)に入る
- 高ければ高いほどブロック(後述)に入りやすい

「確認」を押して終わり。



これだけ。

手に入れ方

- 取引所で購入して、ウォレットに送金
 - 取引所にもウォレット的な機能があるので、BTC送金するときは必ずビットコインアドレスを入れる。
- 誰かにもらう
 - 自分のビットコインアドレスを教えてそこに送金してもらいましょう。
- マイニングがんばる
 - 難しいし電気代の元が取れないのでやめておきましょう。

ここからしくみ

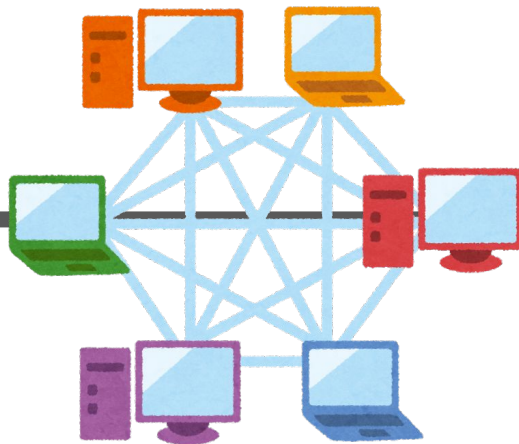
ウォレット

アリスさん



ウォレット

いんたーねっと。



ボブさん



ウォレット

ノード

ノードとはビットコインネットワークに参加しているコンピュータを指します。

ノードはいくつかの役割に別れています。主なものが以下の4つです。4つすべてを行っているのはフルノードと呼ばれたりします。

- ウォレット
- マイナー
- ブロックチェーンDB
- ルーティング

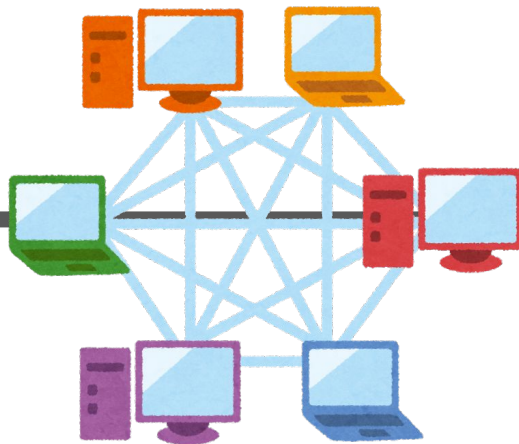
ウォレット

アリスさん



ウォレット

いんたーねっと。



ボブさん



ウォレット

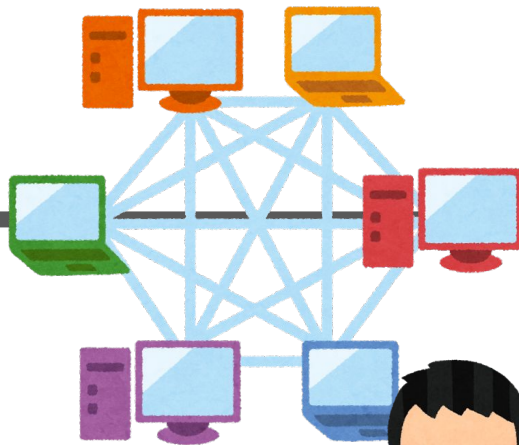
マイナー

アリスさん



ウォレット

いんたーねっと。



マイナー

ボブさん



ウォレット

ブロックチェーンDB

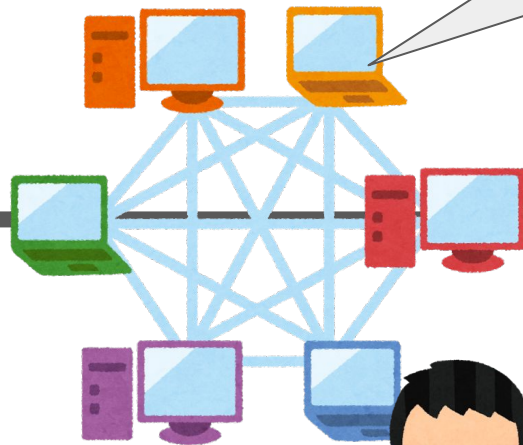
アリスさん



ブロックチェーンDB

ウォレット

いんたーねっと。



ブロックチェーンDB

ブロックチェーンDB



マイナー

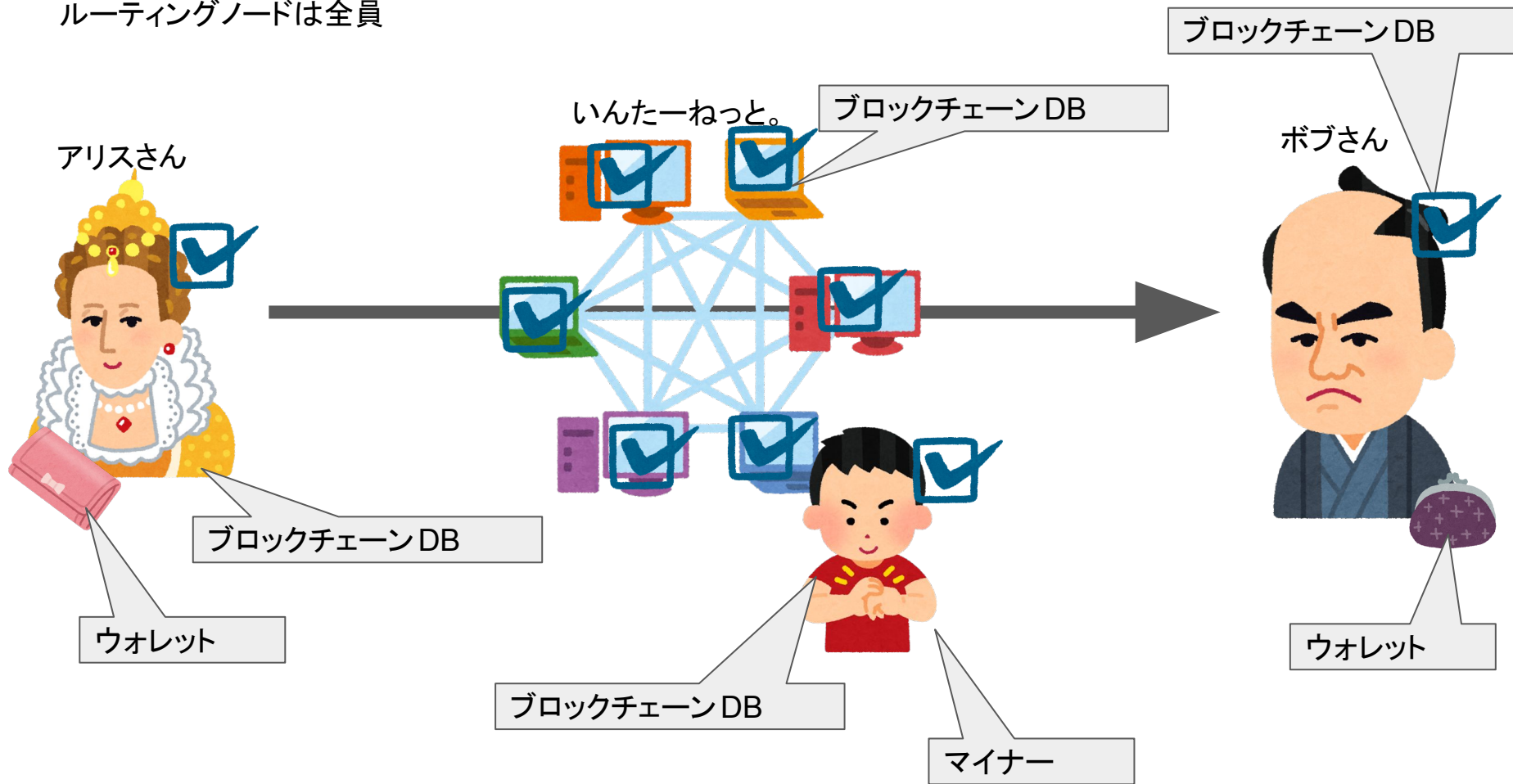
ブロックチェーンDB

ボブさん



ウォレット

ルーティングノードは全員



ウォレット

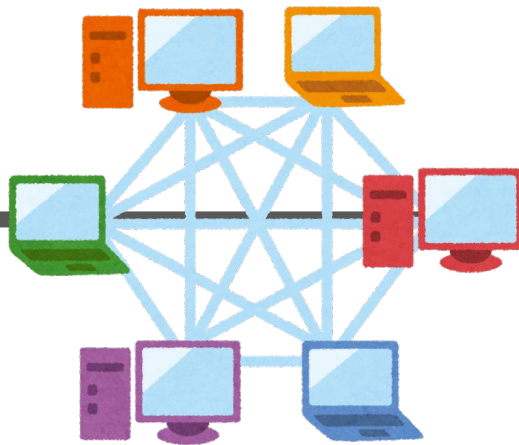
ウォレット

アリスさん



ウォレット

いんたーねっと。



ボブさん



ウォレット

ウォレット

- いわゆるお財布
- Bitcoinを使う人が持っているもの
- 基本的にはブロックチェーン、ウォレット、ネットワークの機能を持つ
- bitcoind(リファレンス実装)ではBerkeley DBを利用(昔からあるKV型のDBM)
- 保持している主なもの
 - 秘密鍵と公開鍵のペア
 - 入出金に使用するビットコインアドレス
- ウォレットの種類
 - apps(Copay, IndieSquare, BreadWalletなど)
 - Web(Blockchain.info, Coinbase, Bitgoなど)
 - ハードウェア(Trezer, Ledger, 紙に印刷など)

ウォレット

アリスさん

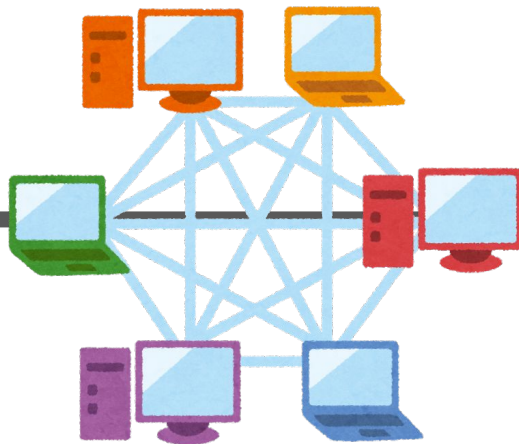


ウォレット

- 1EZBqbJSHFKSkVPNKzc5v26HA6nAHiTq6
- 1E4nqRyFxE5dVZCL83YR2y68tJqJurYhcR

ビットコインアドレスはいっぱい持ってる。(もしくは必要なときに作る)

いんたーねっと。



ボブさん



ウォレット

- 1L75eRMgeCwAxEjD1oWXjLgud9jxwxm34u
- 1EZBqbJSHFKSkVPNKzc5v26HA6nAHiTq6

トランザクション

トランザクション

- 誰から誰にいくら送るかのレコード
- インプットとアウトプット(ビットコインアドレスと金額が入る)
- ウォレットの鍵で署名
- ブロック内に蓄積されていく
- リファレンス実装ではlevelDB(google製のKVS)
- UTXO(unspent transaction output)をインプットとアウトプットに入る
- UTXOとはコインそのもの

ウォレット

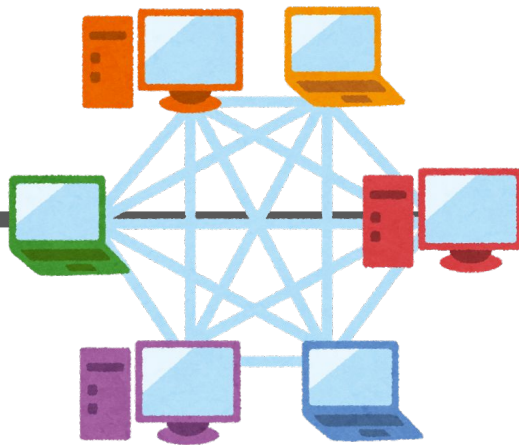
アリスさん



ウォレット

- 1BTC持ってる
- 1ALi2xx

いんたーねっと。



ボブさん



ウォレット

0.6BTC送りたい

•1BOB1xx

アリスさんからボブさんに0.6BTC送りたい

- こんな感じのデータが入っています。
- UTXOから1BOB1xxに0.6BTC送る
- UTXOから1AL2xxに0.4BTC送る
- それを署名

ここで重要なのが

おつりという概念

- なんで0.4BTCをアリスさんに戻さねばならないか。
- 1回使いきり。分割することはできない。
- これがインプット(UTXO)

P2SHとP2PKH

- P2SH(pay to script)
 - 3から始まるアドレス
 - トランザクションの中のスクリプトに認証方法を記載する。
 - 現在の実装ではマルチシグが使われている。
 - よりセキュア
- P2PKH(public key hash)
 - 1から始まるアドレス
 - はじめから使われている
 - トランザクションアウトプットの中に相手先の公開鍵が入っている。

ブロック

ブロックチェーンDB

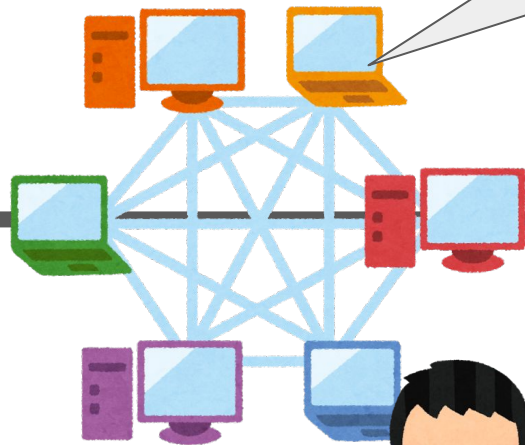
アリスさん



ブロックチェーンDB

ウォレット

いんたーねっと。



ブロックチェーンDB

ブロックチェーンDB



マイナー

ブロックチェーンDB

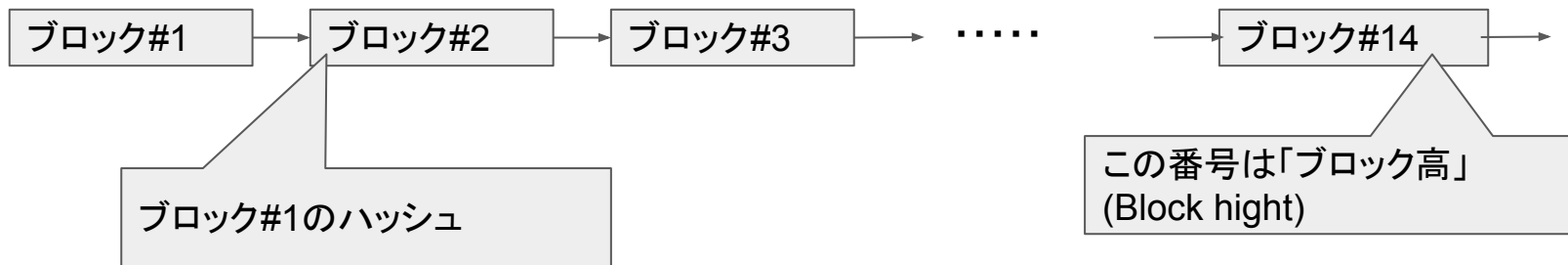
ボブさん



ウォレット

ブロック

- 一定量のトランザクションデータの塊
- 10分間のトランザクションで1つのブロックとなる。
- ブロック内に一個前のブロックのハッシュ値を入れてつなげる。
 - ブロックがチェーンしていくのでブロックチェーン
 - ブロックに連番をつける&前のブロックのハッシュを入れることで改ざんを防ぐ
 - このデータはすべてのブロックチェーン DBノードで同じものを共有

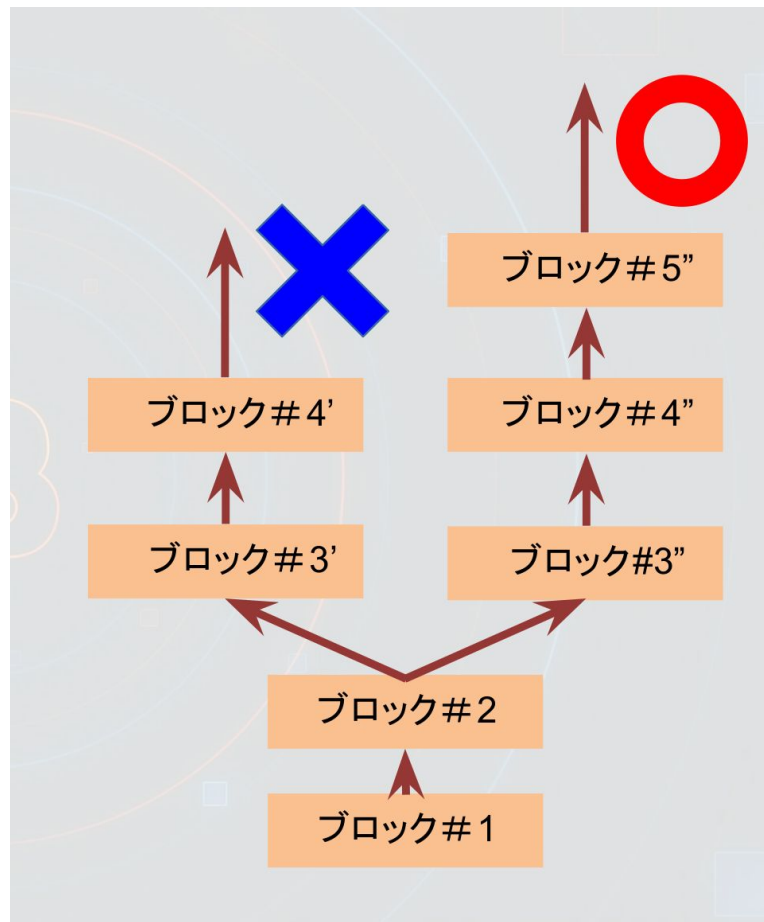


ブロックの分岐

いつ、誰がブロックを生成するか分からない。
異なるノードが同時にブロックを生成すると、
ブロックチェーンは「分岐」してしまう。

どのブランチを「正しい」とすればいい？

ビットコインでは、最も長いブロックチェーン
のブランチを採用することになっている。
(※厳密には、最もハッシュパワーがつぎ込まれた
ブランチが採用される。)



ブロック(チェーン)に関して

- マークルツリー
- ビザンチン将軍問題(誰が正しいのか問題)

とかとかあります。時間がなくなりそうなので割愛

マイニング

マイニング

- 新しいブロックを作ること。
- 新しいトランザクションをブロックに入れてnonceを変えながらsha256ハッシュを計算していった答えが出たらブロックをチェーンできる。

答えが合った人に報酬をゲットできる。(bitcoinで。)

- 50BTCから210,000ごとに半分になっていて、今は12.5BTC
- +トランザクション手数料

誰でもブロックを作れちゃうと困るので一番計算できた人が勝ちというしくみ。

これがProof of Work(「仕事したこと」の証明という)(PoW)

プルーフ・オブ・ワーク(PoW)

勝手にブロックを生成できるのでは、ダメ。

参加ノード全員でくじ引きで決める？

- これならば、参加者の大多数が善良であれば改ざんはできない
- では、どのように投票券を分配すればいい？
- IPアドレスごとに一票だと、大量のIPアドレスを確保できる人がいるので不公平になってしまう

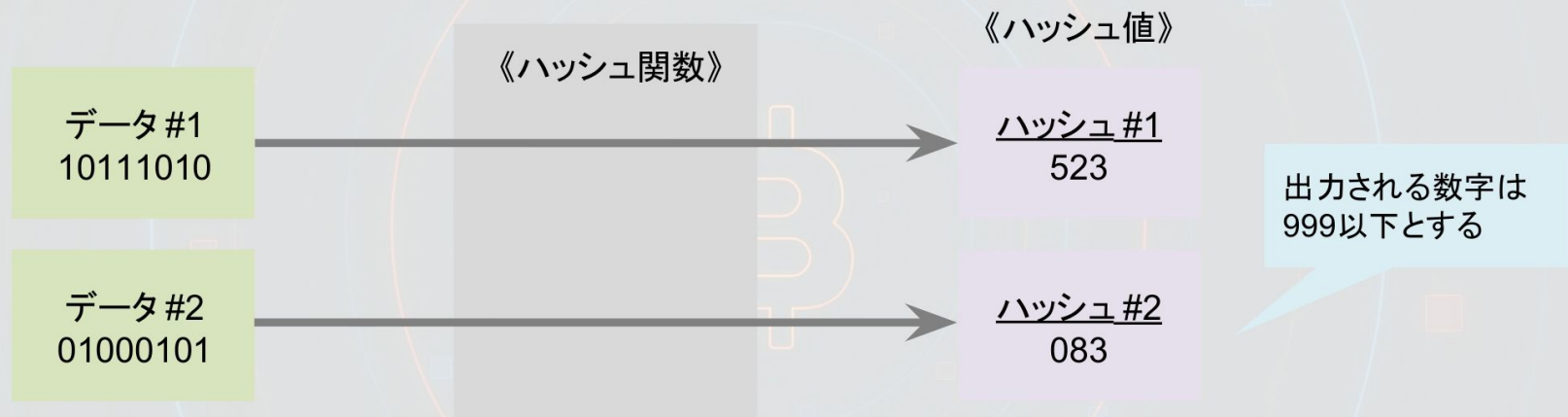
⇒ CPU一個につき、一票にしよう！（提案）

- 大量のCPUを独占するためには、大量の電気代とハードウェア代が必要
- IPアドレスを確保するよりは難しそう

▶ プルーフ・オブ・ワーク(Proof of Work; PoW = 働いたことの証明)

ハッシュ関数

「あるデータが与えられた場合にそのデータを代表する数値を得る操作、または、その様な数値を得るための関数のこと」(Wikipedia)



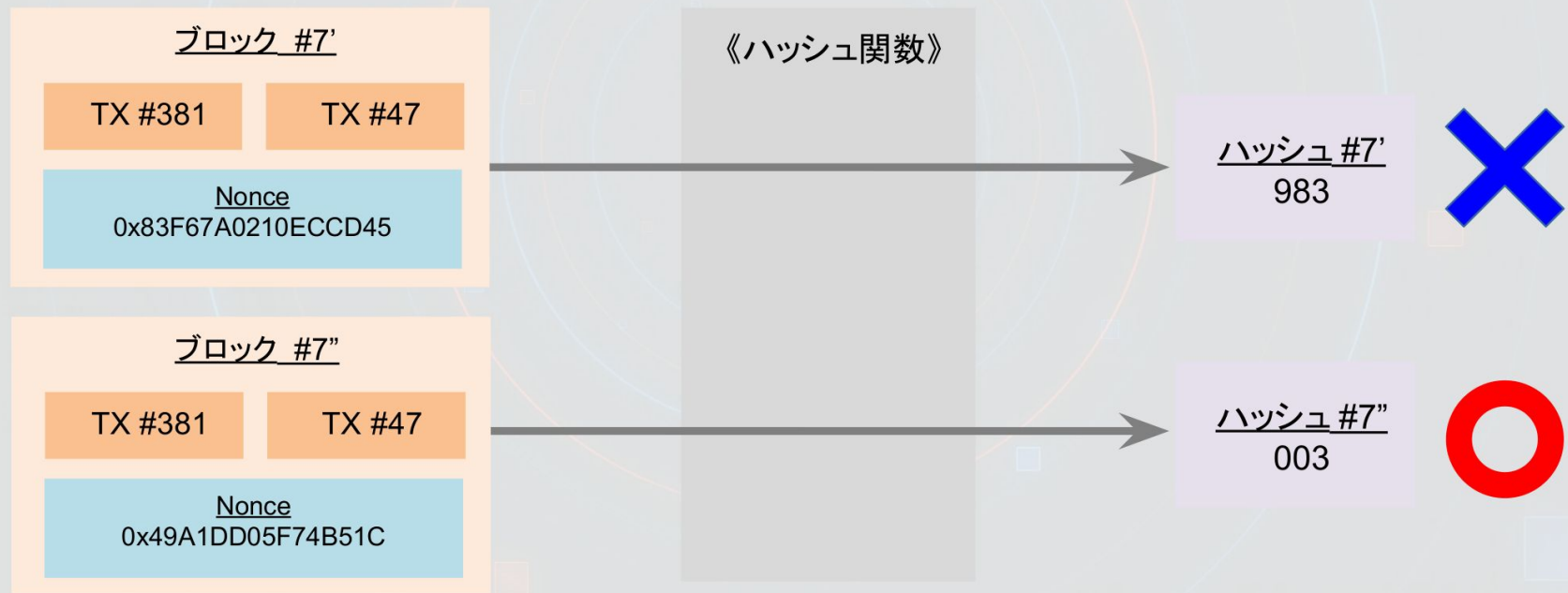
出力されるハッシュ値は、入力の値が変わるとランダムに変化する
(ように見える)

厳密には「暗号学的ハッシュ関数」という

ブロック生成と組み合わせる

ブロックに、ハッシュ値調整用のデータ (nonce、ナンス) を付加する。

ハッシュ値がある値以下となる nonce を発見できたら、ブロックが生成(採掘)される。



採掘の難しさ

Bitcoin はハッシュ関数として SHA-256 を採用している。

(とりうる範囲: $0 \sim 2^{256} \doteq 1.2 \times 10^{77}$)

例えば、ブロック高 #448,705 (2017/01/18 8:55:58 JST) のハッシュ値は、

0x000000000000000021ecb68b5f8ae6f2e4f3c36d58be50b2bd9c1704c4e43f7

であり、およそ

$1,446,996,272,793,502,613,504 \text{ 回} \div 1.45 \times 10^{21} \text{ 回} = 1.45 \text{ ゼタ回}$

のハッシュ計算を行うと、一つのブロックを採掘可能。

おまけ:京コンピュータで採掘するとどうなる？

京は約10Pflopsなので、ざっくり1,000flopsあたり一回ハッシュ計算可能と見積もると、一つのブロックを採掘するのに約4.6年(!)かかる。

ビットコインのこれから

取引量増加問題

- 現在のしくみでは1秒間に7回程度のトランザクションしか処理できない
 - ちなみにVISAカードは世界で47,000/秒も処理できちゃうゾ

改善策として下記のようなものが模索されています。

- ブロックサイズの上限を上げる(今は1MB)
- ブロック以外のデータベースを持ってそこにデータを逃がす
 - いろんな方法が提唱されています。

詐欺に注意！

- 仮想通貨(暗号)通貨に関する詐欺が横行しています。
- 「100%儲かる！」や「買った人みんなが儲かる！」みたいなおいしい話には注意してください。

ビットコインとか勉強会/暗号通貨輪読会

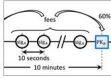
月に1回～2回程度勉強会を開催しています。



暗号通貨読書会 / 勉強会
暗号通貨(cryptocurrency)に関連する最新技術動向の論文読書会 / 勉強会
主催: 「暗号通貨読書会 / 勉強会」運営委員会

イベント メンバー 資料 [Bitcoin](#) [Google+](#) [Facebook](#) [Twitter](#) [グループのメンバーです](#)

イベント (12件)

イベント	日時	場所	参加人数
 【暗号通貨輪読会#7】 Bitcoin-NG	2017/03/06 (月) 19:30～	東京品川区西五反田2-8-1 (五反田ファースビル 9F)	43/67人
 ビットコインとか勉強会#5	2017/02/28 (火) 19:30～	東京都渋谷区恵比寿1-19-15 (ワノサワ東急ビル5階)	81/105人
 【暗号通貨輪読会#6】 Tumblebit	2017/02/07 (火) 19:20～	東京都港区港南 2-16-3 品川グランドセントラルタワー	61/55人
 ビットコインとか勉強会#4	2017/01/10 (火) 19:30～	東京都渋谷区東1-2-20 (住友不動産渋谷ファースタワー7F)	83/85人