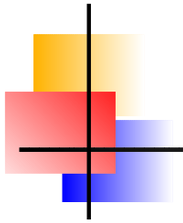
魔法を信じるな。

Web3 技術のリアリティ

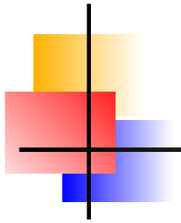
早稲田大学 大学院経営管理研究科
斉藤 賢爾



このスライドは

https://speakerdeck.com/ks91

に置かれています



このお話では

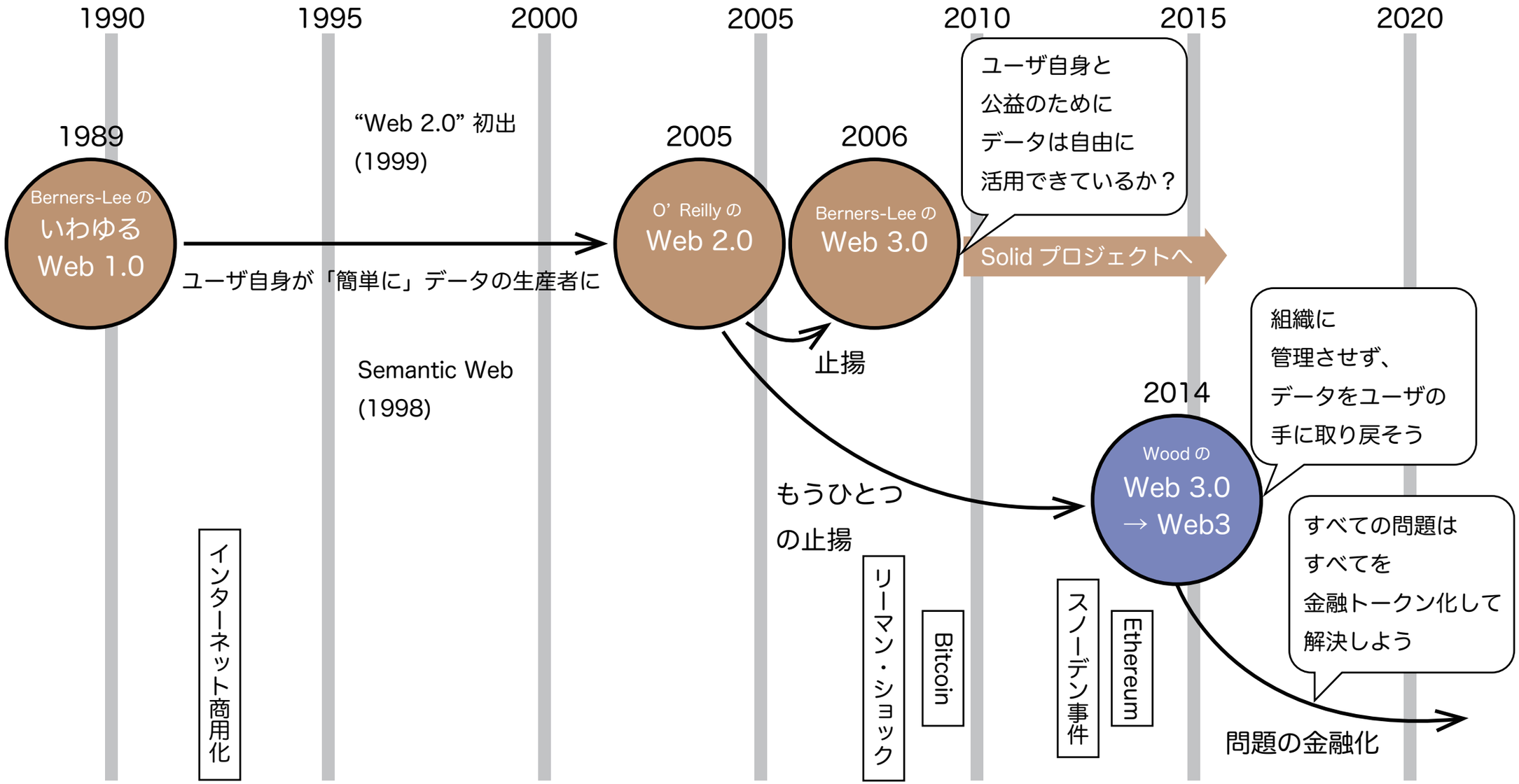
- Web3 という言葉がいまだに巷を賑わせています
- Web 3.0 を 語源とするこの言葉は、「World Wide Web を応用したビジネスの革新であった Web 2.0 が、副作用として生み出した諸問題」を解決するための、新しいパラダイムを表しているはずです
- その諸問題とは何で、新しいパラダイムはどんなもので、それによりどうして問題は解決されると考えられているのでしょうか
- そして、そのパラダイムで本当に問題は解決されるのでしょうか

- この講演では、Web3 を支えるとされる技術の現実を露わにします

Web3 編

- Web3 はどのような問題を解くとされているのでしょうか

Web 1-2-3


1990
1995
2000
2005
2010
2015
2020
1989
Berners-Lee の
いわゆる
Web 1.0
“Web 2.0” 初出
(1999)
ユーザ自身が「簡単に」データの生産者に
Semantic Web
(1998)
2005
O’ Reilly の
Web 2.0
2006
Berners-Lee の
Web 3.0
ユーザ自身と
公益のために
データは自由に
活用できているか？
Solid プロジェクトへ
止揚
もうひとつ
の止揚
2014
Wood の
Web 3.0
→ Web3
組織に
管理させず、
データをユーザの
手に取り戻そう
すべての問題は
すべてを
金融トークン化して
解決しよう
問題の金融化
インターネット商用化
リーマン・ショック
Bitcoin
スノーデン事件
Ethereum

Web 3.0 や *Web3* は何を解きたいか

- いわゆる Web 1.0 (Berners-Lee) : Read × Write
  - 研究者のための出版メディア → 基本的に全員が論文を書き、読むのだから最初から双方向
  - 「ユーザがデータを管理するが、出版は容易でない」
- Web 2.0 (O'Reilly) : Read × Write ← 以前 (Web 1.0 の時代) から
  - 「ユーザはデータを管理できないが、出版が容易」
- Web 3.0 (Berners-Lee)
  - 「ユーザがデータを管理し、かつ出版が容易」を目指す → Solid (Social linked data)
- Web 3.0 → Web3 (Wood)
  - Ethereum をウェブから使えるようにする ← web3.js, web3.py
- Web3 (Dixon) : Read × Write × ~~Own~~
  - 「私たちがオンラインで行うほぼすべてのことの内部に、トークンという形で金融資産を組み込む」(Bloomberg)
  - 他者へのトラストに依らずにトークンは所有できても、トークンが指したり包含するものは所有できない

体験したい方は

- 「実践！スマートコントラクトプログラミング」シリーズのスライド資料
  - 第 1 回 Ethereum 入門
    - Python 3 の新しめの版で問題がある場合は Python 3.9 (3.9.18) で
    - Ethereum 上のトランザクションを試すには、例えば Görli テストネットの ETH を取得する必要がありますが、メインネットに 0.001 ETH 以上の残高をもつアドレスが必要です
    - または “.brownie/network-config.yaml” の Ethereum の networks に以下を加えた上で Sepolia テストネットで

```
- chainid: 11155111
  explorer: https://api-sepolia.etherscan.io/api
  host: https://sepolia.infura.io/v3/$WEB3_INFURA_PROJECT_ID
  id: sepolia
  multicall2: '0x5BA1e12693Dc8F9c48aAD8770482f4739bEeD696'
  name: Sepolia (Infura)
  provider: infura
```

  - 第 4 回 Ethereum 演習 III
    - Adam byGMO (コントラクト：0xb30fC2D754C88c451275b743b6F530F19f643683) の事例をお試しください

ブロックチェーン編

- 概要
- Ethereum ブロックチェーン
- スマートコントラクトの実行メカニズム

ブロックチェーン：満たすべき性質

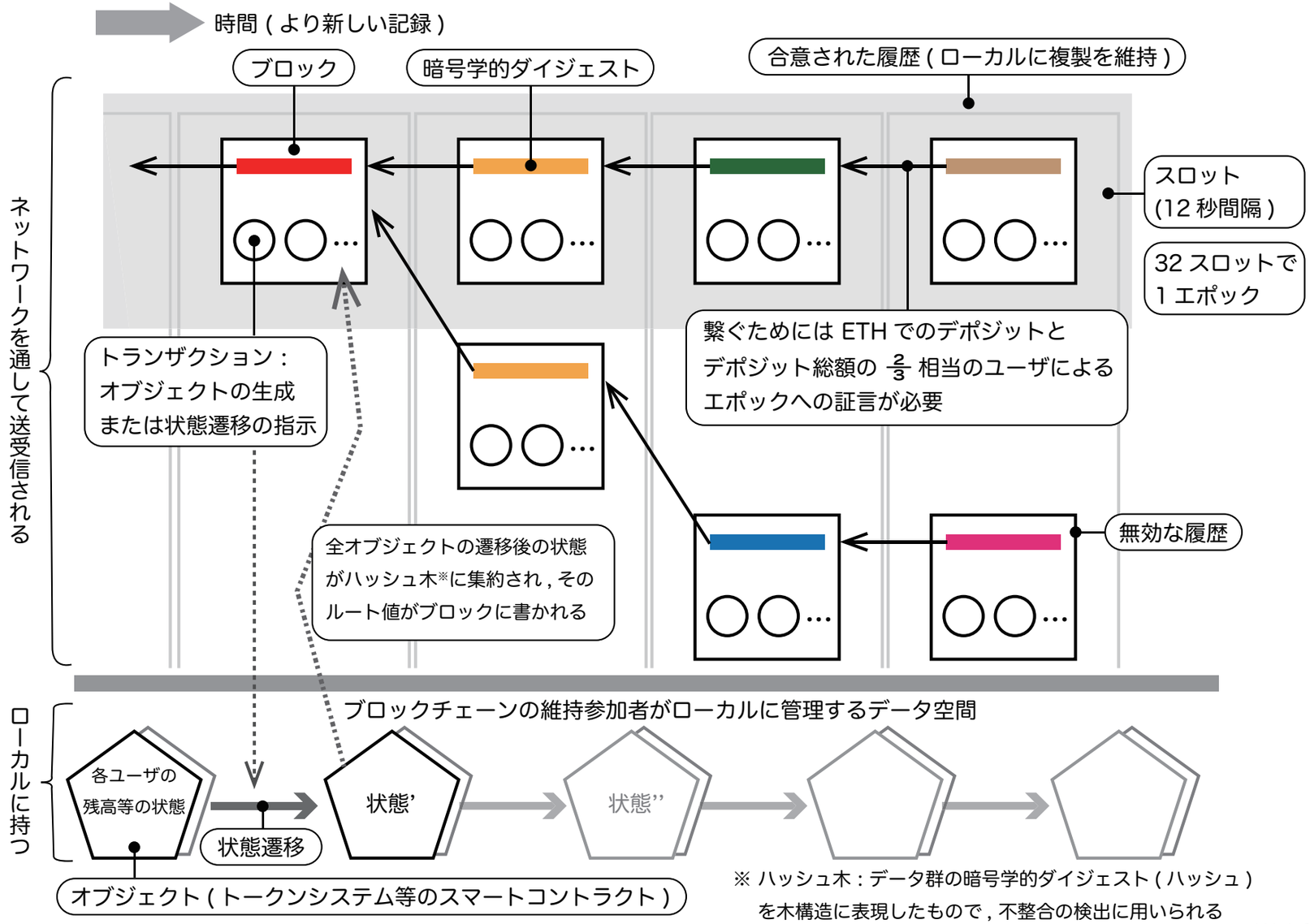
- 元々 Bitcoin を可能にするために発明され、
  「自分が持つコインを自分だけが自由に誰かに送るのを誰にも止めさせない」
  必要があるのだから . . .

- 自己主権性：ユーザ自身が意思決定して実行できる (例：アカウントを勝手に作れる)
- 耐検閲性 (狭義)：他者の意思で記録やその確認を妨げられない
- 耐障害性：故障によっても記録やその確認を妨げられない
- 耐改ざん性：過去の記録を抹消・改変・捏造できない

⇒ といった広義の「耐検閲性」が満たされる必要がある

- いかなる方法によっても記録の否定ができない
- 技術なので動作条件がある
  - ブロックチェーン特有の動作条件は「ネイティブ暗号資産の市場価格が十分に高い」

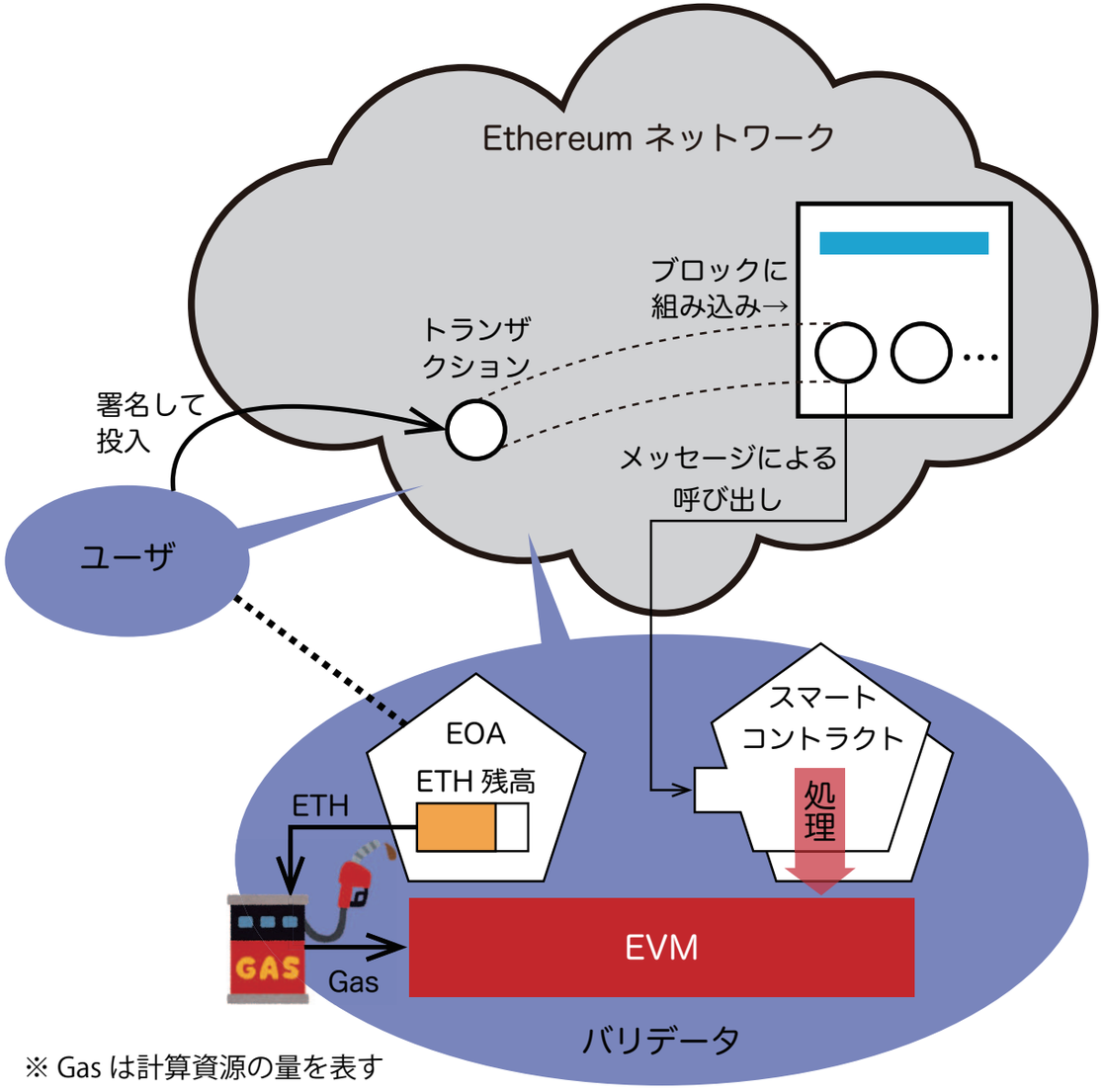
Ethereum ブロックチェーンの構造



※ ハッシュ木 : データ群の暗号学的ダイジェスト (ハッシュ)
を木構造に表現したもので , 不整合の検出に用いられる

- 大事なことは . . .
- ETH で報酬を得る
  バリデータたちが
  自発的に参加する
  - だから ETH の市場
    価格が十分に高い
    必要がある
- 各自がもつ状態マシンの
  状態を確実に一致させる
  - 等しい初期ブロック
    から始まる
  - ブロックを全員に
    コピー
  - ブロックの並びが
    等しい
  - 非決定でない処理

スマートコントラクトの実行メカニズム



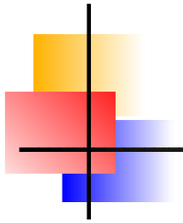
※ Gas は計算資源の量を表す

※ Gas 使用料を ETH で支払うユーザがトランザクションを投入しない限り
スマートコントラクトは動作しない

- 広義の耐検閲性を満たす台帳に
プログラムコードとデータを
書き込んで実行できるからこそ
スマートコントラクトは有用

- 大事なことは . . .
- Ethereum 仮想マシン (EVM) が
スマートコントラクトを実行する
- それを指示するユーザは ETH で
計算資源量を買わなければ
ならない
- 誰かが計算資源量を買わないと
スマートコントラクトは動かない

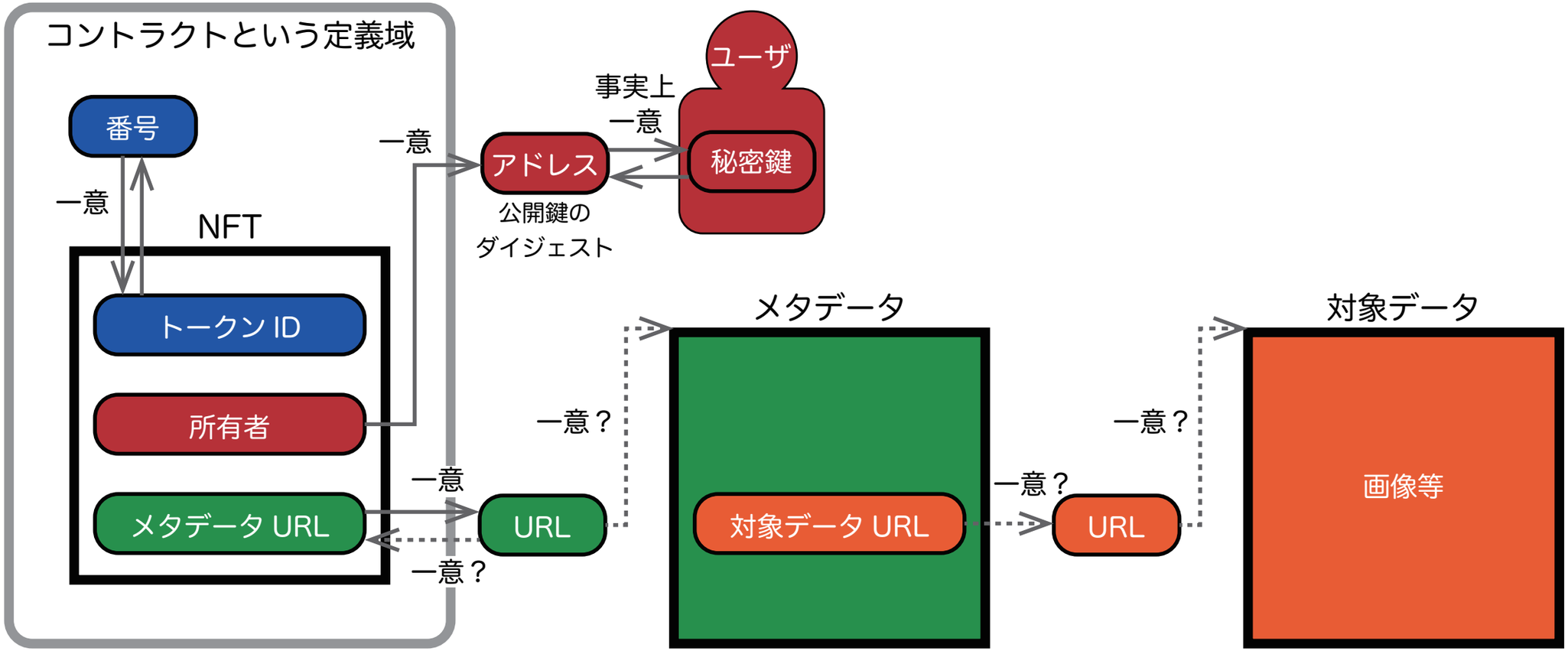
  ↑ DAO 実現への落とし穴



NFT (Non-Fungible Token) 編

- ファンジブルか、ノン・ファンジブルかの見分け方
  - 券を借りて、同じ数量を示す別の券ですぐ返した時、怒られるかどうか
    - 1 万円札は？ → ファンジブル (代替可能)
    - コンサートのチケットは？ → ノン・ファンジブル (代替不可能)
- 概念的に券 (チケット) であり、発行元をトラストすることにもとづきます
  - トラストできない発行元は同じ意味をもつ (とされる) NFT を無数に作れます
  - トラストできるかもしれないけど NFT をブロックチェーンに書き込まない発行元も多く存在します

ERC-721 仕様にもとづく NFT

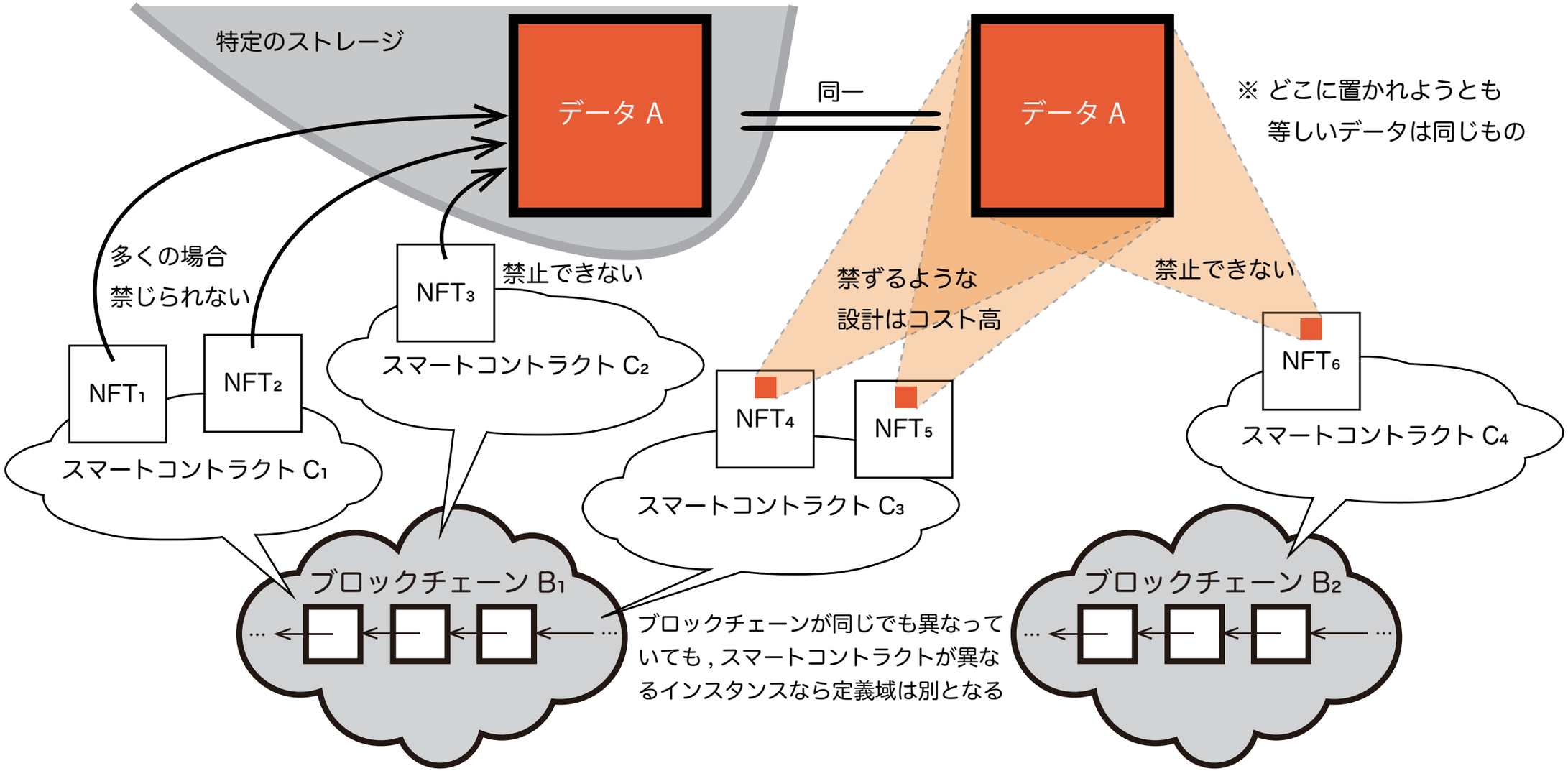


※ そのコントラクトという定義域の中で , 実線矢印で示した一意性を保証するに過ぎない

※ 破線矢印で示した一意性は , 実装・運用の工夫次第で保証できる

例えば , URL にそれが指すリソースの暗号学的ダイジェストを含めること (例 : IPFS の利用) により「URL → データ」の一意性を保てる (その場合でも「データ → URL」の一意性は無い)

NFT の唯一性にまつわる幻想を捨てよう



※ NFT がデータを指す方式とデータを格納する方式は , データの可用性は異なるとしても一意性に関わる性質は変わらない

NFT の実在性も、ときに疑える

- 未出庫の NFT という概念
  - Ethereum 上には存在しない NFT が売買されています
  - いくつかの NFT マーケットプレイスは、そのことを明記しています
    - 「出庫手続き」をしない限り、実際には Ethereum に書き込みません
    - Lazy Minting という手法であり、賛 (?) 否あります
    - チケットでこれをやったら無意味 (従来と変わらない) なのだから、トークンの売買が前提
  - 書き込みコスト (Gas 使用料) が高い (かつ従量制である) ため、このような方法がとられる

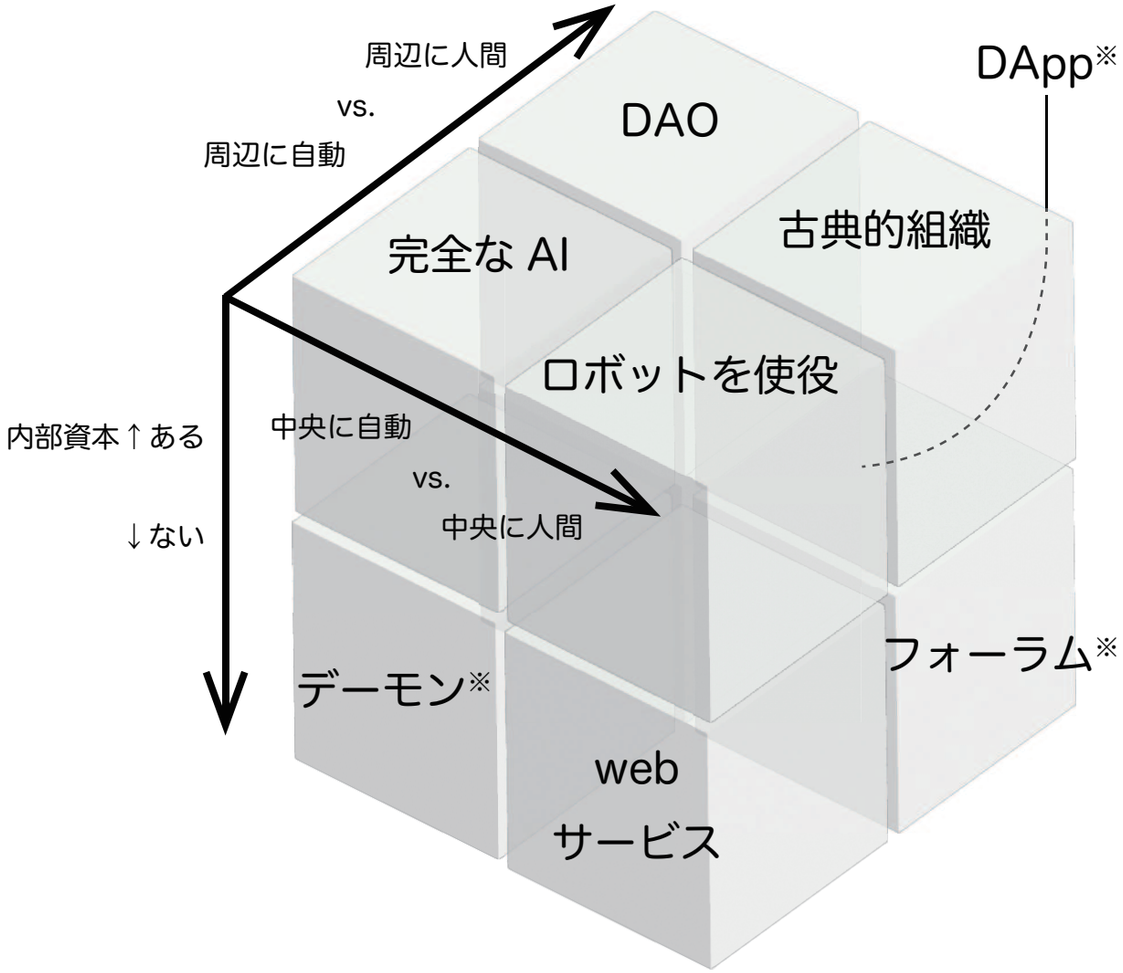
DAO (Decentralized Autonomous Organization) 編

- 分散型自律組織

  (私たちは分散の後に「型」とは付けないので、分散システム研究開発のコミュニティの外で扱われている概念だと分かる)

- 意思決定を自律的に行えるでしょうか
  - つまり、誰からもコントロールされないと本当に言えますか？

DAO と各種組織のキューブ



※ DApp：スマートコントラクトによるアプリケーション (Decentralized App)
※ フォーラム：人々が特定の話題について議論したり情報を交換したりする場
※ デーモン：バックグラウンドで稼働し，イベントに自動的に対応するプログラム

- 左のキューブには矛盾がありますが、それはさておき . . .
- Buterin による DAO の定義 (2014)
  - インターネット上に自律的に存在するが、自動システム自身にはできない特定のタスクを担うために、人間を雇うことに大きく依存している
  - そのため、内部に資本（報酬として使われ人間を駆動する）をもつ
  - 意思決定を自律的に行う
- 自律とは？
  - 外部から支配されない

DAO なのか、そうではないのか

- Bitcoin は DAO？
  → DAO
- Ethereum は DAO？
  → うん、まあ . . . DAO？ (人間がさっさと意思決定するけど) (特定の集団が支配してるっぽさがあります)
- スマートコントラクトで作ったものは DAO？
  → え？. . . あれれ？
  - 自律的には動いていない (呼び出されないと動かない)
    ↑ 総体としてみんなで動かしている応用システム、という解釈はできるかも
  - トークンの持ち分による投票に意思決定を依存している
    - 提案はスマートコントラクトのコードとして書かれる
    - それを全員が読めることが前提となる
  - 可決した提案を実行する (計算資源量を買う) のは誰？
    - 決まった誰かがやるなら、その人に拒否権があり実質的な支配者では？
    - 誰でもできるなら、真意を難読化した提案を用いて容易に攻撃できる ← 実例あり

まとめに代えて — 「所有」は何かの解決になるのか

- 「所有」の概念の浅さというリアリティ
  - なんでもトークン化すれば解決するのか？(しませんよね . . . します？)
    - 「ユーザがデータを管理し、かつ出版が容易」は実現できているのでしょうか
    - 「ユーザがデータを管理する」のがオーナーシップであれば Web 1.0 で出来ていますが？
  - 他者へのトラストに依らずにトークンは所有できても、トークンが指したり包含するものは所有できません
  - トークンの所持は証明でき、ブロックチェーンは広義に検閲できないので、誰にも邪魔されずに各自の意思表示はできます
    - したがって、トークンの所持量に応じた投票は実現できます
    - しかし投票は意思決定における最後の手段では？
      - 投票をする前にやるべきことがあると分かっている DAO もあるようですが、その部分、普通に Web 2.0 的に動いていたりしませんか？
- そもそも . . .
  「同じ志をもって物事を行う集団」は誰かが所有できるべきものなのでしょうか