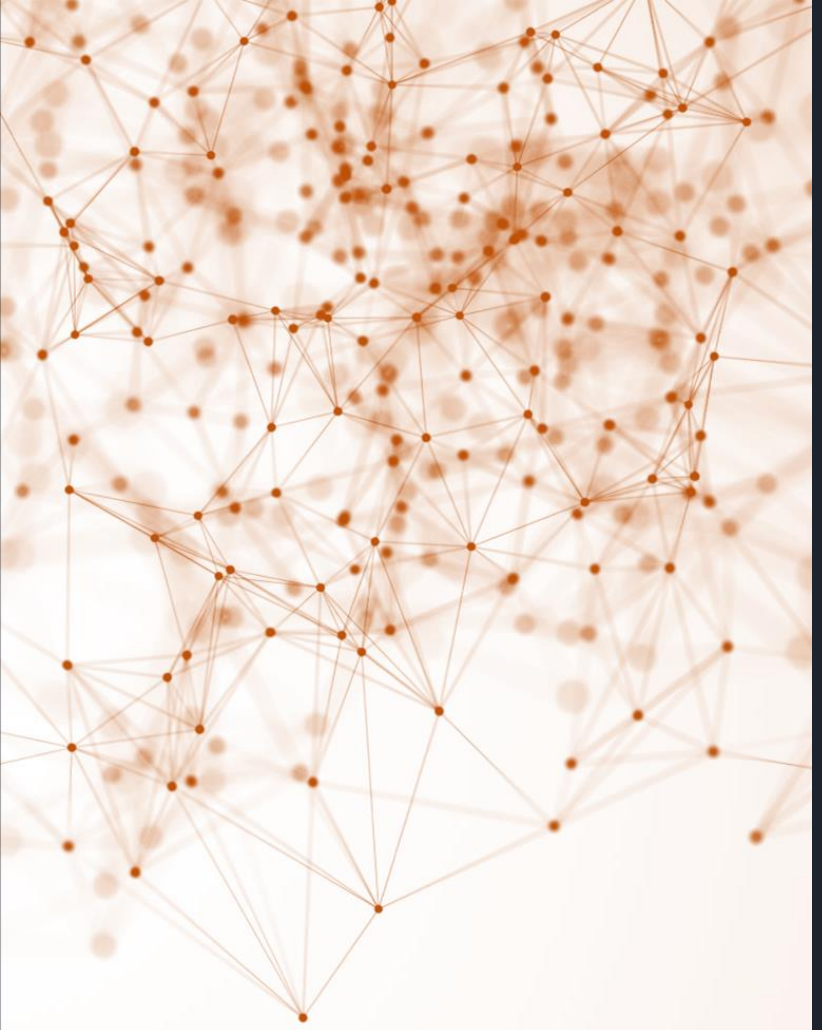


# 30分で理解するブロックチェーン(基礎編)

KUMOPRO
BLOCKCHAIN

Lesson 01

1. ブロックチェーンの特徴
2. ブロックチェーンの中身
3. コンセンサスアルゴリズム
4. スマートコントラクト

# 1. ブロックチェーンの特徴

# ブロックチェーンを一言で表すと

「あらゆる資産を、管理者不要かつ
改ざんできない状態で管理することのできる技術」

# 管理できる資産

# 資産の種類

| | |
|---|---|
| 流動資産 | 現金 |
| | 預金 |
| | 受取手形 |
| 固定資産 | 土地 |
| | 美術品 |
| | 著作権 |

ブロックチェーンで管理できるもの

「データ化できる」 かつ
「価値があると認められる」もの

# ブロックチェーン応用例

- 仮想通貨の取引履歴
- 著作権の証明
- サプライチェーンの可視化
- 学歴の証明

# 改ざんできない理由

# ブロックチェーンの通信方式

- Peer to Peer方式(PtoP、P2P)
- ブロックチェーンを管理するノードが直接通信
- 各ノードが同一のデータを保持

# P2P方式

単一障害点がない

一つのノードが故障しても、ネットワーク全体へ影響を及ぼさない。

処理の即時性に欠ける

全ノードが同一のデータを保持する必要があるため。

分散ネットワークのため、
全ノードが同一のデータを保持している

全てのノードが管理するブロックチェーンを
改ざんする必要がある

# 管理者不要の理由

# 分散ネットワークにおける意思決定方法

## 全ノードが一致した意思決定を下すことが必要

- ビザンチン将軍問題の解決

◆ ビザンチン将軍問題

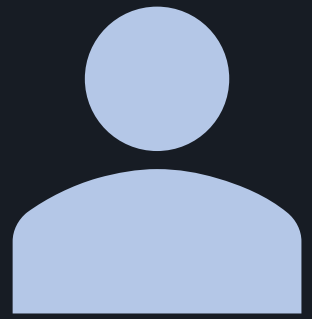
特定多数の意思決定者が存在する状況下で、どのようにすれば合理的で最適な意思決定を下すことができるか、という思考事例のこと

# ビザンチン将軍問題

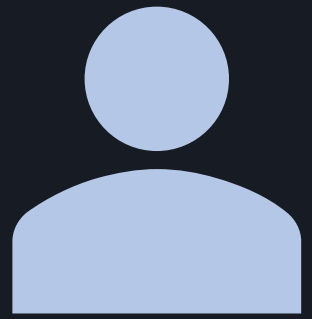
- 登場人物
  - 将軍A：ビザンチン帝国のスパイ
  - 将軍B：オスマン帝国
  - 将軍C：オスマン帝国
- 前提
  - 全ての将軍が一致団結しなければ
  - ビザンチン帝国は攻略不可能

ビザンチン帝国

将軍A

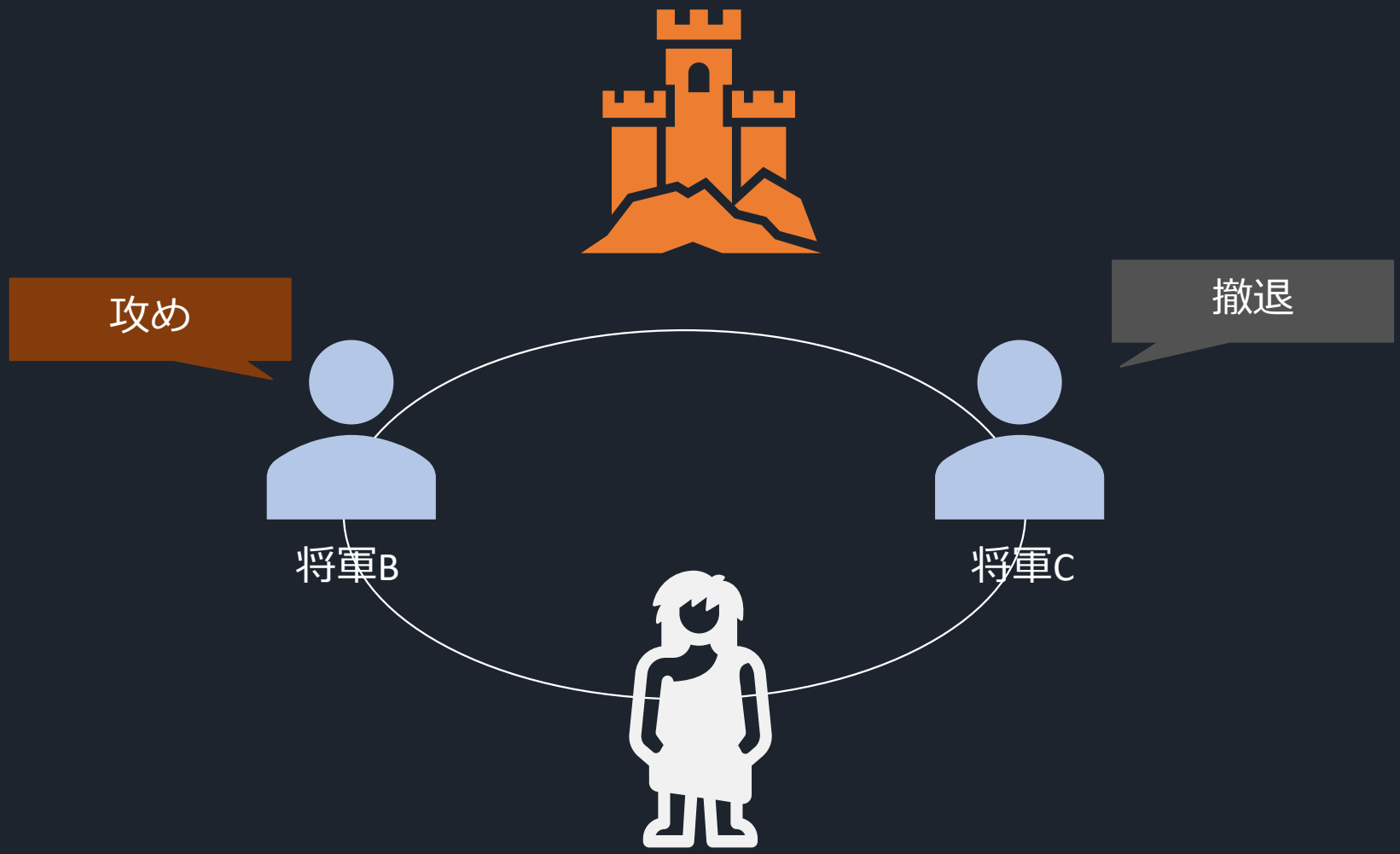


将軍B



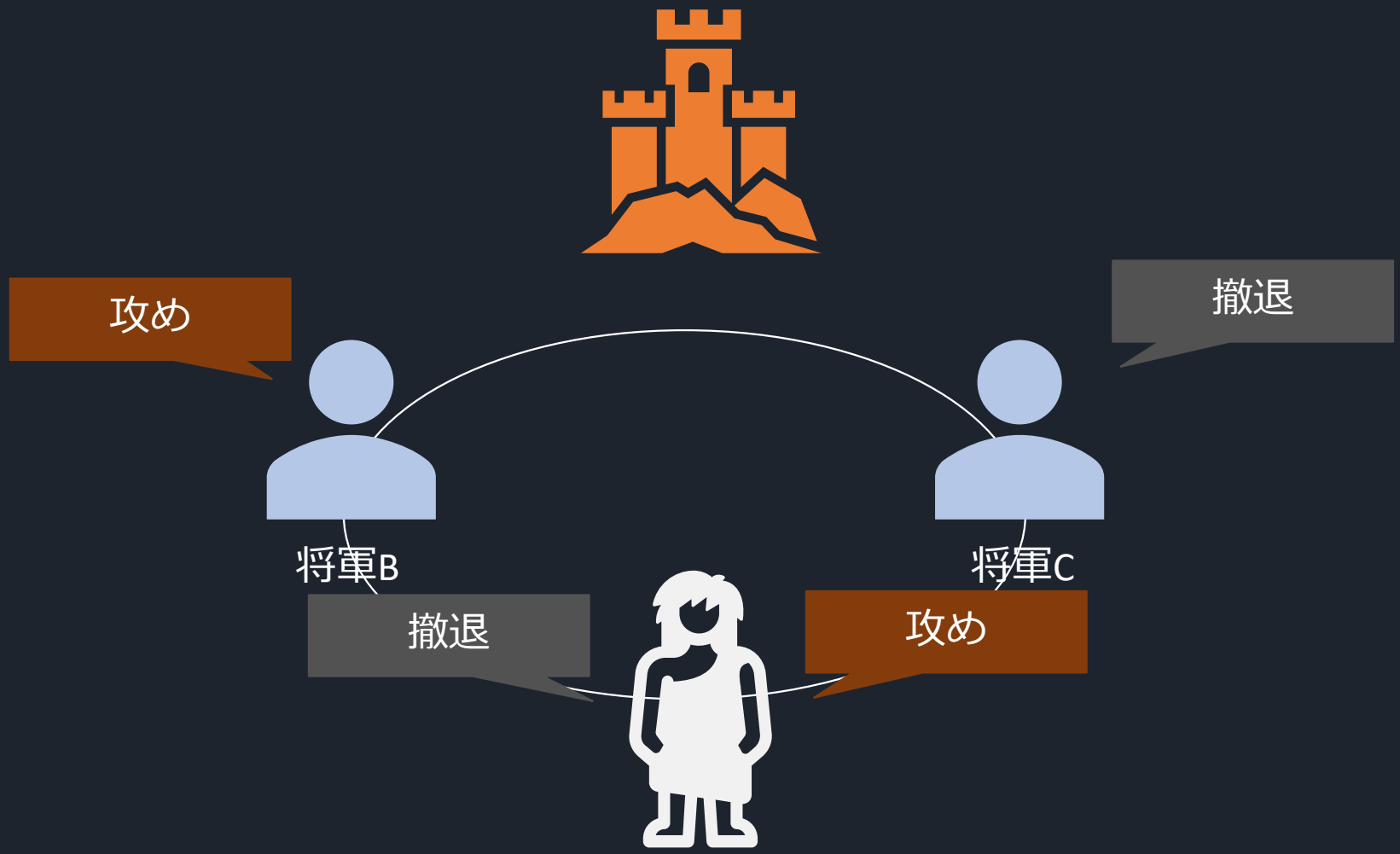
将軍C

# ビザンチン将軍問題



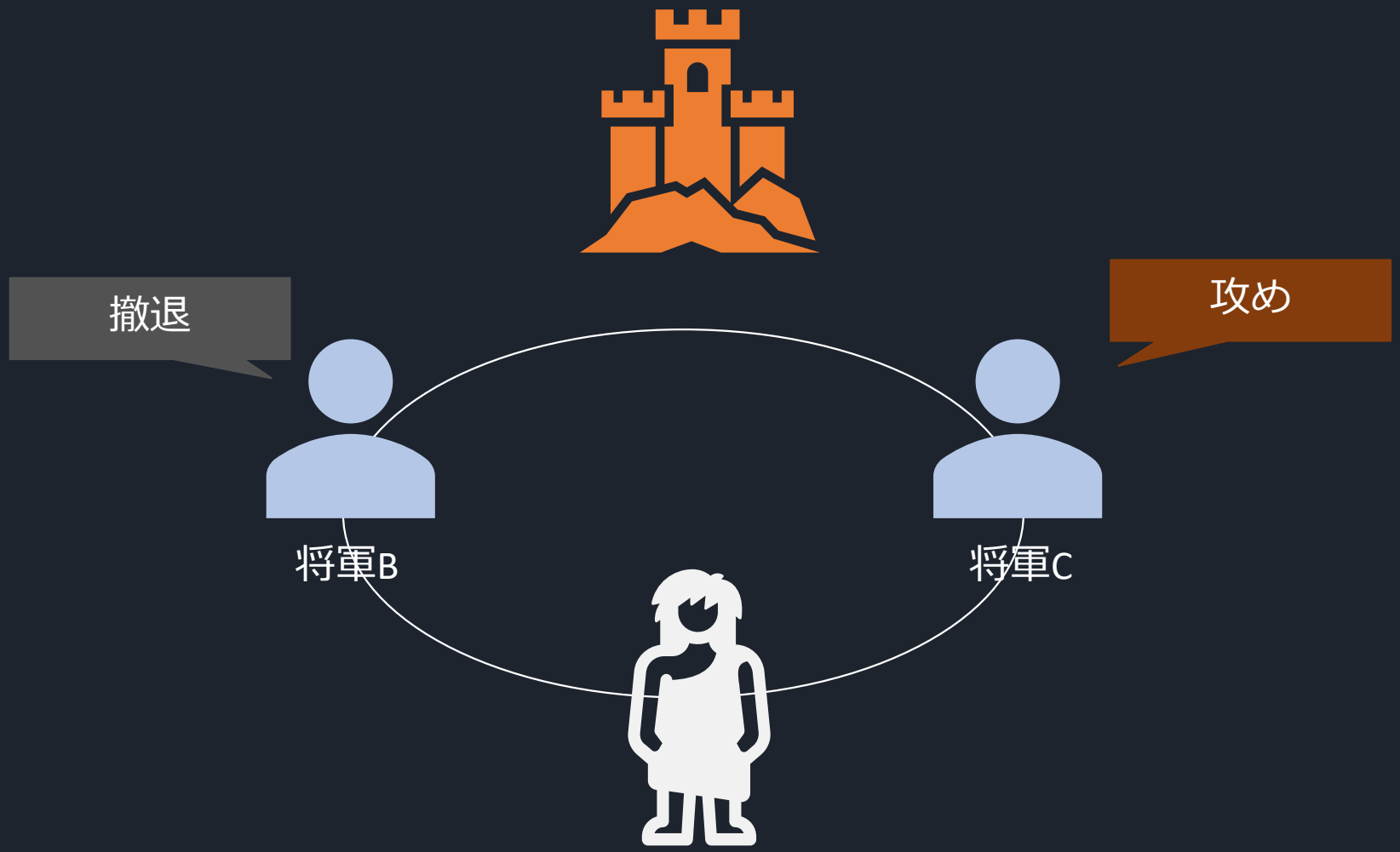

1人が「攻め」、1人が「撤退」の意見を表明し
残り1人の意見によって全将軍の行動が決定される状況。

# ビザンチン将軍問題



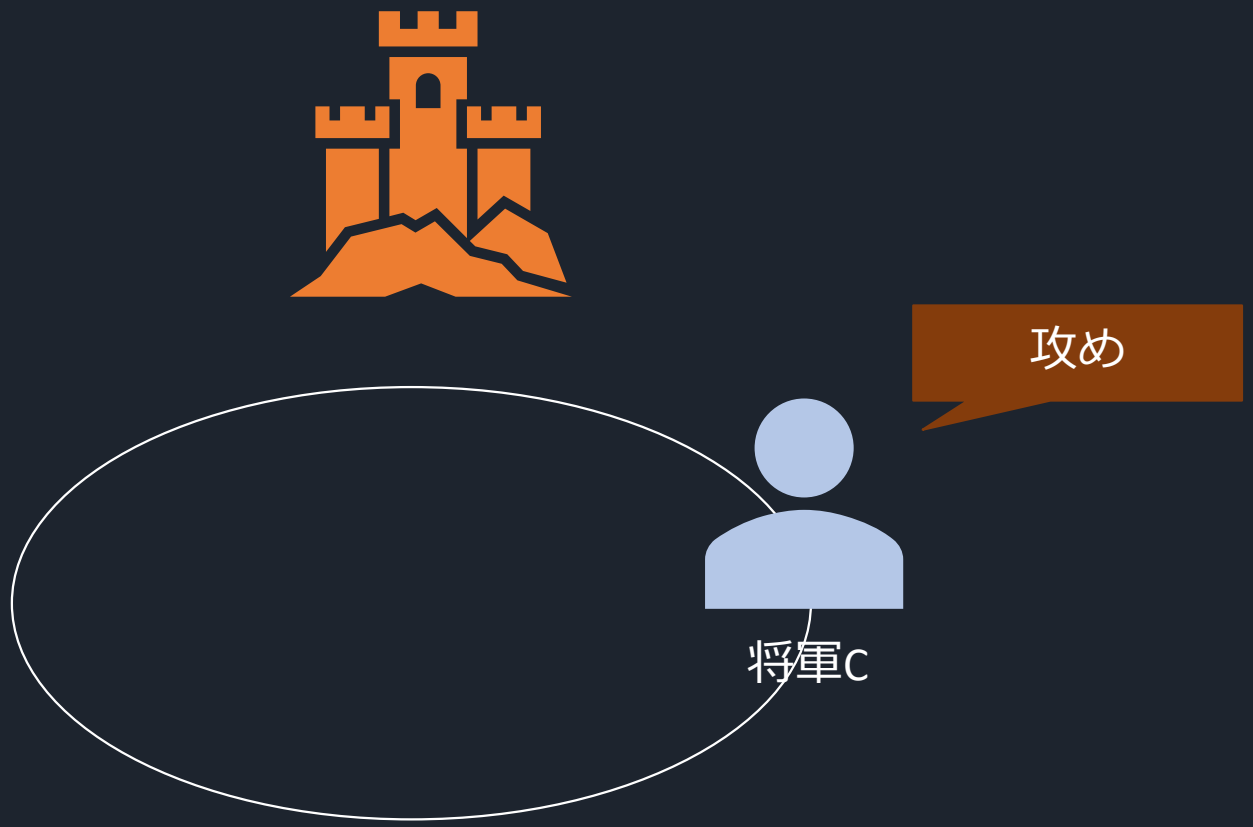

スパイは「攻め」を表明している将軍へ「撤退」を伝達し
「撤退」を表明している将軍へ「攻め」を伝達

# ビザンチン将軍問題




将軍Bは「攻め」を表明していたが多数決により「撤退」
将軍Cは「撤退」を表明していたが多数決により「攻め」を選択

# ビザンチン将軍問題




将軍cのみビザンチン帝国を攻めることとなり失敗に終わる

# ブロックチェーンによるビザンチン将軍問題の解決方法

## 経済的合理性の提示

最適な意見を表明するとインセンティブを付与する

▼

コンセンサスアルゴリズム

## デメリットの提示

ブロックチェーンが破壊され、インセンティブを取得できない

改ざんするためには時間や機器等のリソースを膨大に消費する必要がある

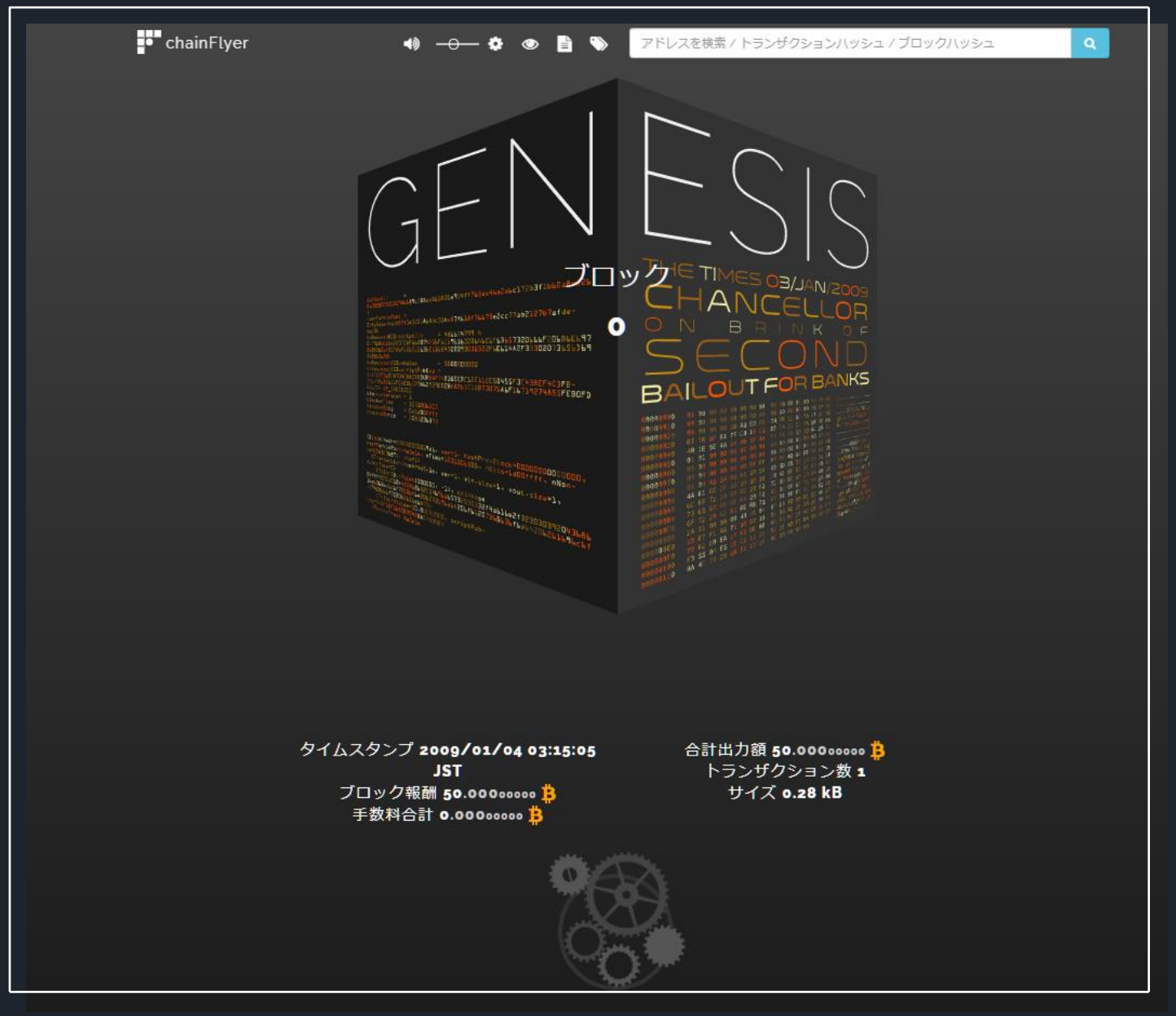
# 2. ブロックチェーンの中身(取引履歴)

# chainFlyer

- ブロックチェーン可視化ツール
- 仮想通貨取引所を運営するbitFlyerが提供

# GENESIS

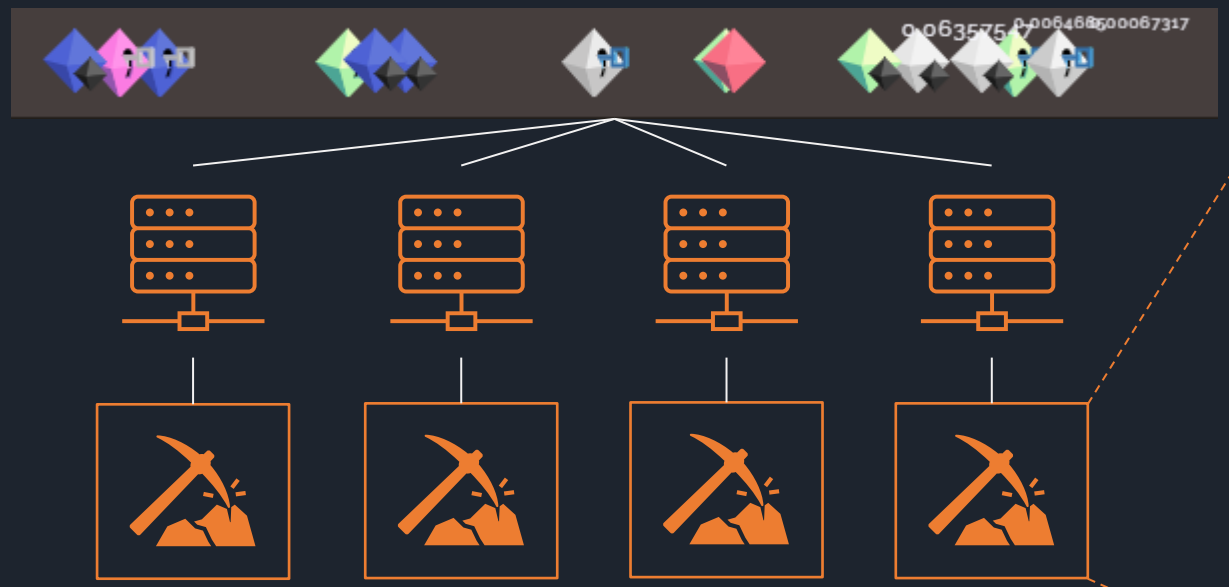
- ブロックチェーンの最初のブロック



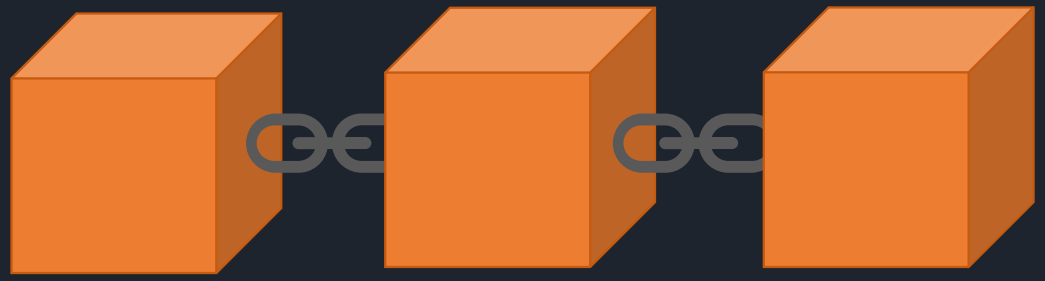
# ブロック

- ①ブロックナンバー
- ②ハッシュ値
- ③トランザクション数
- ④取引情報　etc…

# 取引履歴をブロックチェーンへ格納する流れ①

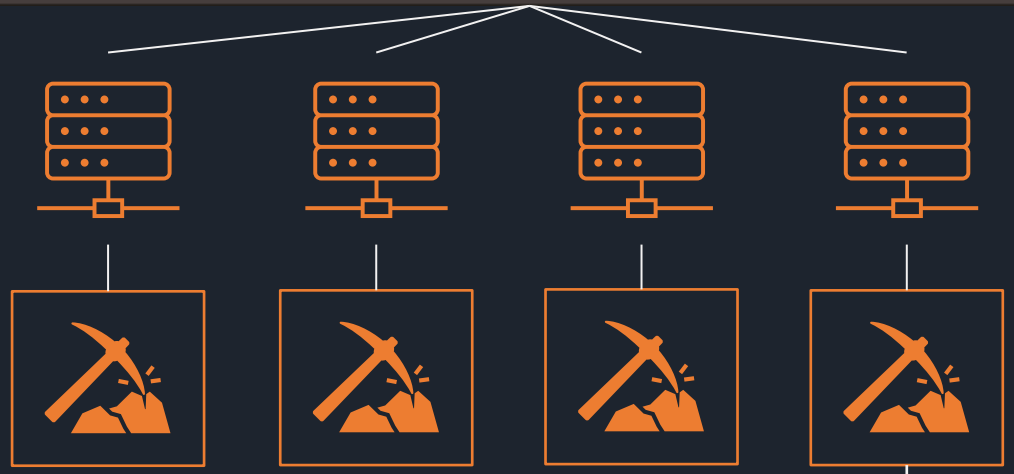



ブロックチェーン



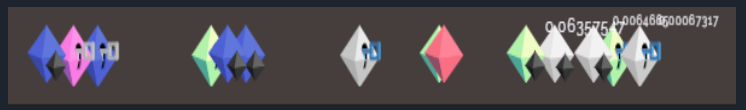
※ナンス
　Number used once(一度だけ使用される使い捨ての数字)。32ビットの数値で、取引データと合わせてハッシュ関数に通す。

# 取引履歴をブロックチェーンへ格納する流れ②



ブロックの中身

前ブロックのハッシュ値



ブロックのハッシュ値

ナンス

ブロックチェーン

# 3. コンセンサス アルゴリズム

# コンセンサスアルゴリズムとは

各ノードが保持するデータの正当性を
担保する仕組み

# コンセンサスアルゴリズムの種類

- Proof of Work(PoW)
- Proof of Stake(PoS)
- Proof of Importance(PoI)
- Proof of Consensus(PoC)

# Proof of Work(PoW)

| | |
|---|---|
| 概要 | 仕事量で正当性を担保。一番早く計算問題を解いた人にブロックを提案する権利を付与<br>提案したブロックが他のノードに承認された場合、ブロックを追加する<br>追加出来たら新たに発行されたコイン(Bitcoin等)を受け取ることができる |
| 代表例 | Bitcoin |
| メリット | 非中央集権で、公平性が保たれる |
| デメリット | 電力消費が高く、環境に悪い<br>悪意を持ったノードが過半数を占める場合、ブロックチェーンが改ざんされる |

# Proof of Stake(PoS)

| | |
|---|---|
| 概要 | 暗号資産の保有量によって、マイニングができる確率が高まる方法 |
| 代表例 | Ethereum |
| メリット | 無駄な電力消費を抑えることができる |
| デメリット | 暗号資産の保有量が多い人が、よりお金持ちとなり不公平 |

# Proof of Importance(PoI)

| | |
|---|---|
| 概要 | 保有量だけでなく、取引数や取引量、取引相手の信用スコアなど総合的に判断してブロックを提案できるノードを決める仕組み |
| 代表例 | Nem |
| メリット | 総合評価のためPoSより公平性が担保されている |
| デメリット | 最低限の保有量は必要なため、お金持ちがお金持ちになることを完全に解決できていない |

# Proof of Consensus(PoC)

| | |
|---|---|
| 概要 | あらかじめブロックを承認するノード(バリデーター)を定め、バリデーターの80%が承認するとブロックが追加できる仕組み |
| 代表例 | Ripple |
| メリット | 承認するノードが定められているため、ブロック追加のスピードが速い |
| デメリット | バリデーターが結託して情報操作を行う可能性がある |

# 4. スマート
# コントラクト

# スマートコントラクトとは

スマート(自動的な) ＋ コントラクト(契約)

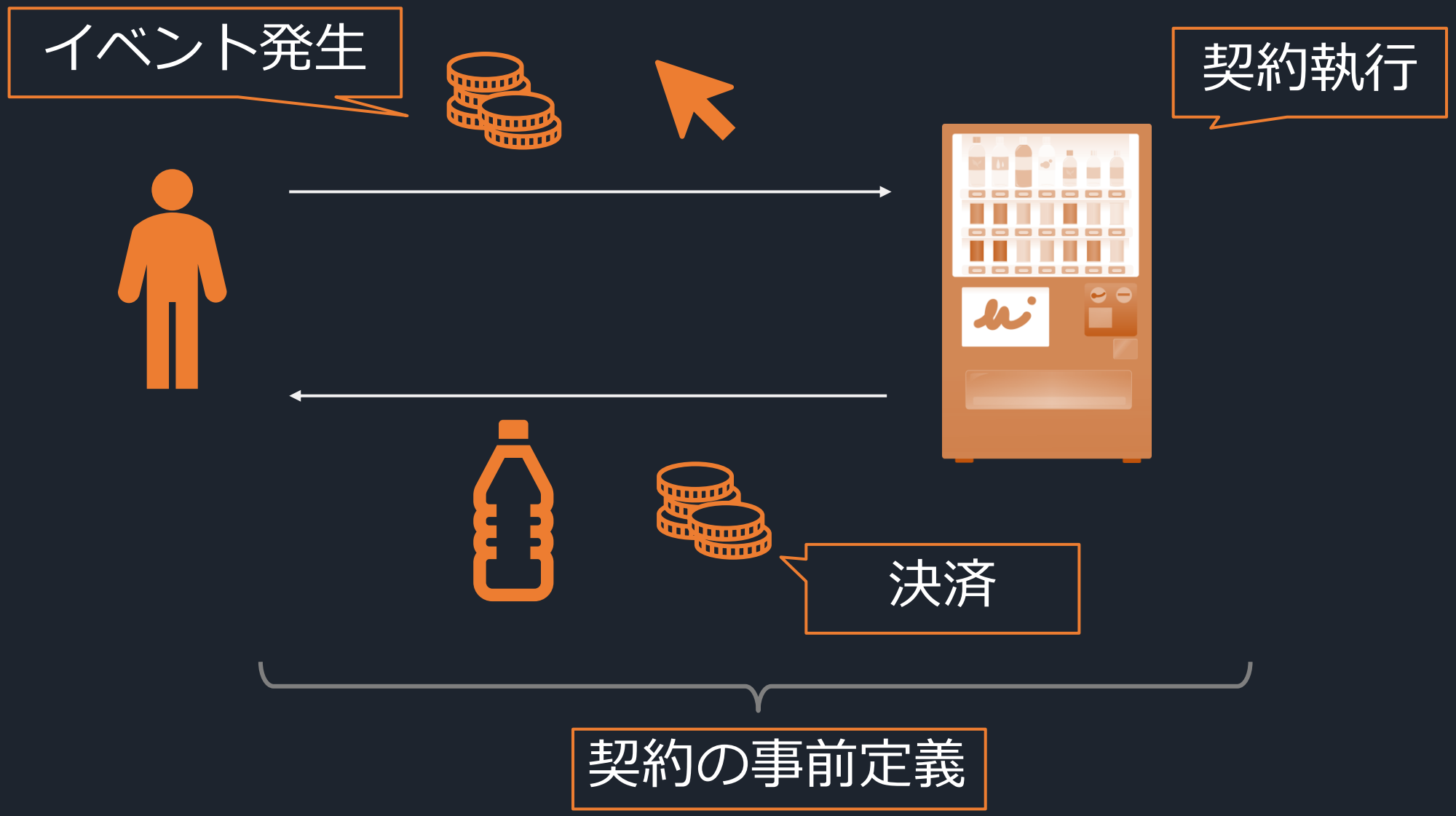
契約のスムーズな検証、執行、実行、交渉を意図したコンピュータプロトコル

第三者を介さずに信用が担保されたトランザクションを処理可能

1994年にニック・スザボにより提唱された

# スマートコントラクトの流れ

# スマートコントラクトの代表例



Bitcoinを筆頭に初期のブロックチェーンは、
取引が発生すると無条件にブロック格納対象としている。

イベントが発生してからブロックへ格納するまでの
条件と処理を記述したプログラムのことを
スマートコントラクトと呼ぶ。

# KUMOPRO

## BLOCKCHAIN