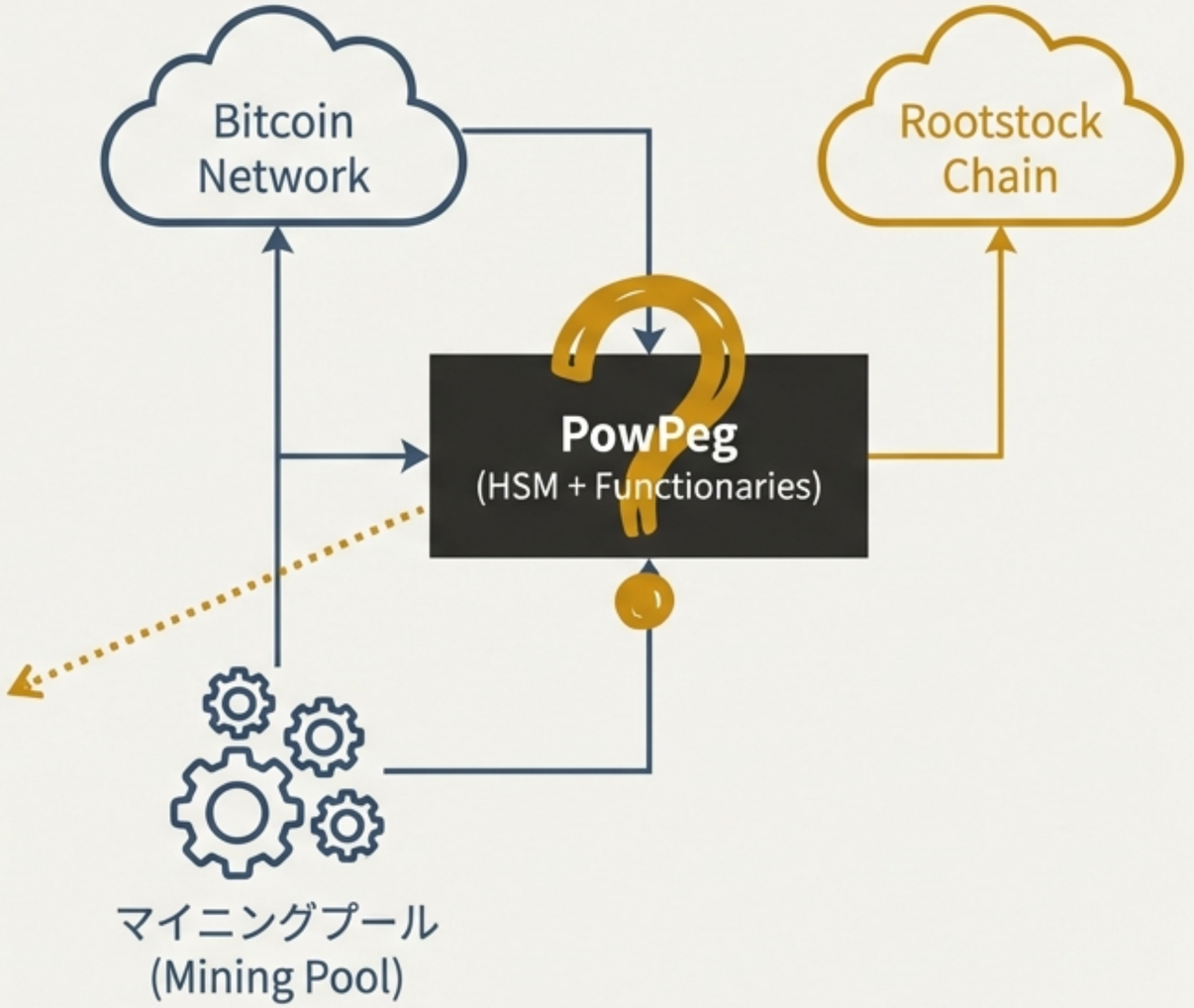
ほぼDrivechainを実現できるのか？

Rootstockブリッジ「Union」の仕組みと技術的挑戦

まずは現状の理解から：RootstockとPowPeg

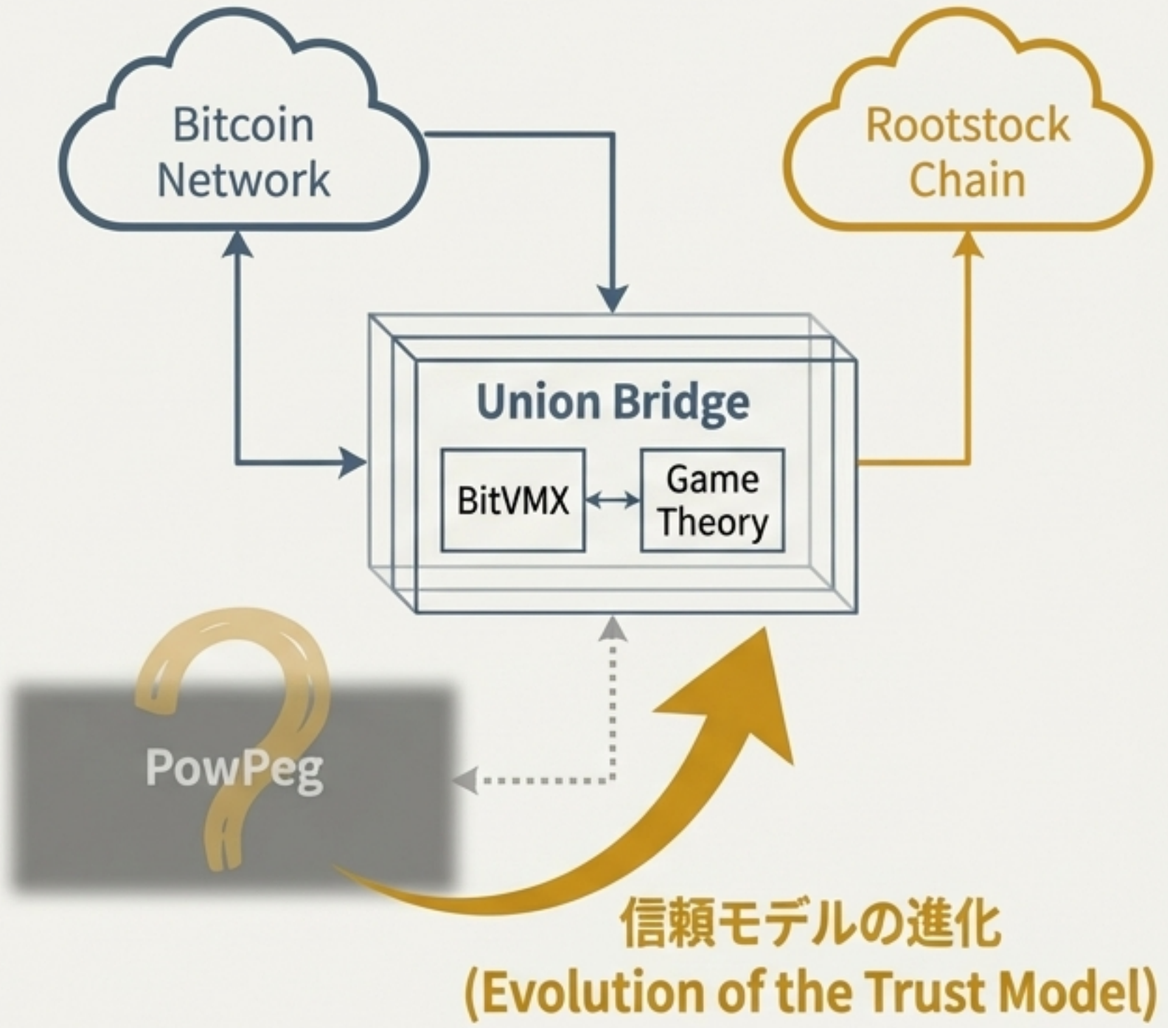
- **Rootstockとは？**
  - ビットコインとマージマイニングされるPoWベースのスマートコントラクト・サイドチェーン。
  - BTCをrBTCにブリッジし、EVM互換環境でDeFiなどを利用可能にする。
- **現在のブリッジ：PowPeg**
  - 2018年から運用されている2-way pegの仕組み。
  - マイナー、PowPegファンクショナリー、HSM（ハードウェアセキュリティモジュール）の多層構造でBTCを保護。
- **PowPegの課題：信頼の境界**
  - HSM内でプログラムが正しく実行されているかを外部から検証する手段がない。
  - ユーザー視点では、Blockstream Liquidのようなマルチシグと信頼モデルが大きく変わらない。
  - 「マイナーがHSM内の署名鍵を不正に利用できてしまうのではないか？」という根本的な信頼の仮定が残る。



新たな提案「Union」：ブリッジ層の再発明

- **Union**は新しいチェーンではない。
  「Bitcoin ↔ Rootstock間のBTCブリッジのコンセンサス」を再定義するL2の仕組み。

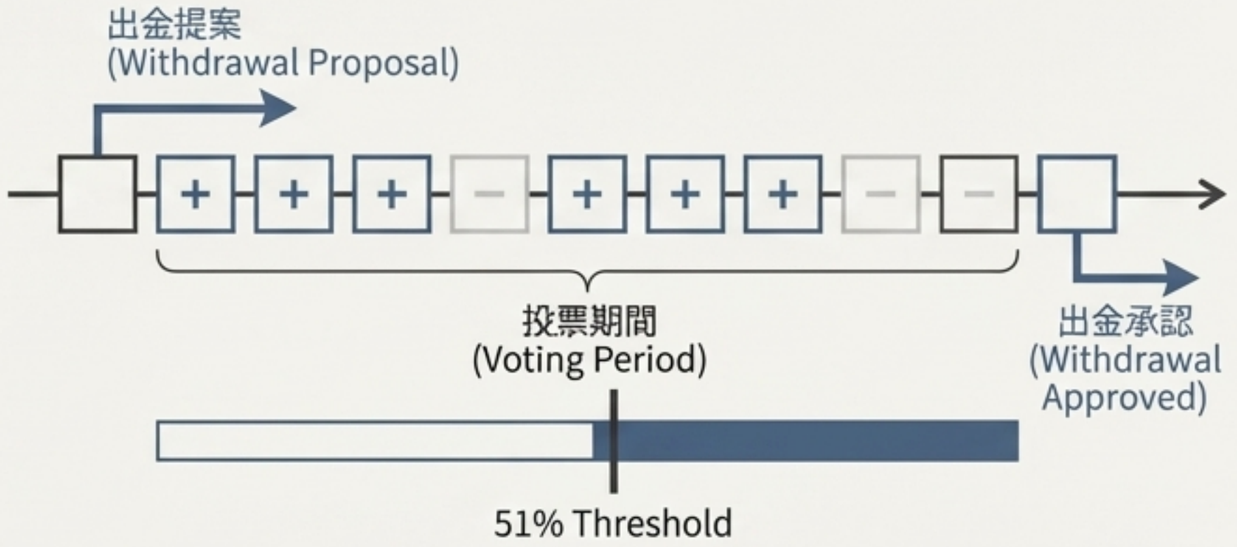
- **目的:** PowPegの信頼モデルを克服し、よりトラストを最小化したブリッジを構築する。
- **コアコンセプト:**
  1. Bitcoin側のBTCロック用UTXOを**BitVMX**で保護する。
  2. Rootstockチェーンの正当性を**SPV検証 + SNARK証明**を用いてBitcoin上で検証可能にする。
  3. 最終的に、BitcoinとRootstockの**51%ハッシュパワー**がサイドチェーンの正当性を決定する仕組みを目指す。



目指す姿：Drivechainのセキュリティモデル (BIP300/301)

Drivechainは、ビットコインのソフトフォークを前提としたサイドチェーン提案。その出金メカニズムは「マイナー投票」に依存する。

- **プロセス:**
  1. サイドチェーンからの出金提案がBitcoinのL1トランザクションとして公開される。
  2. マイナーは数ヶ月間にわたり、コインベーストランザクションにシグナルを埋め込む形で「賛成／反対」を表明。
  3. **51%以上のハッシュパワー**が賛成した提案のみが、最終的に有効な出金として承認される。
- **本質:** サイドチェーンの状態という「外部コンセンサス」を、Bitcoinマイナーの経済的インセンティブ（ハッシュパワー）によって決定する仕組み。



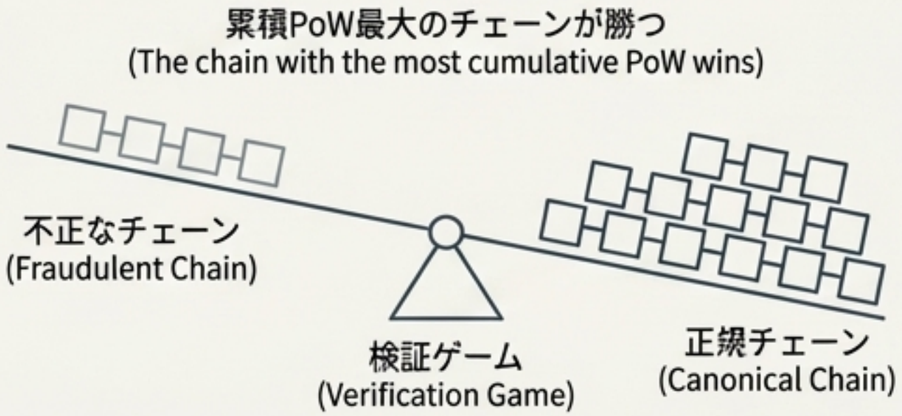
Unionの核心：投票ではなく「ゲーム」で51%ハッシュパワーを反映する

Unionはマイナーによる直接投票を採用しない。代わりに、正直な参加者が必ず勝つように設計された検証ゲームを構築する。

Unionのゲーム構造:

1. **前提:** Rootstockでは、マージマイニングによる「累積PoW（累積難易度）が最大のチェーン」が正規チェーンとなる。
2. **検証:** Bitcoin側では、`checkChain`という関数（後述）を通して提出されたRootstockチェーンのヘッダ列が正規かどうかをBitVMXで検証する。
3. **対抗:** もし不正なチェーン（例：攻撃者が作ったフォーク）が提出された場合、正直な参加者は`checkAltChain`を使い、「より累積PoWが大きい、本物のチェーン」をカウンターとして提出できる。

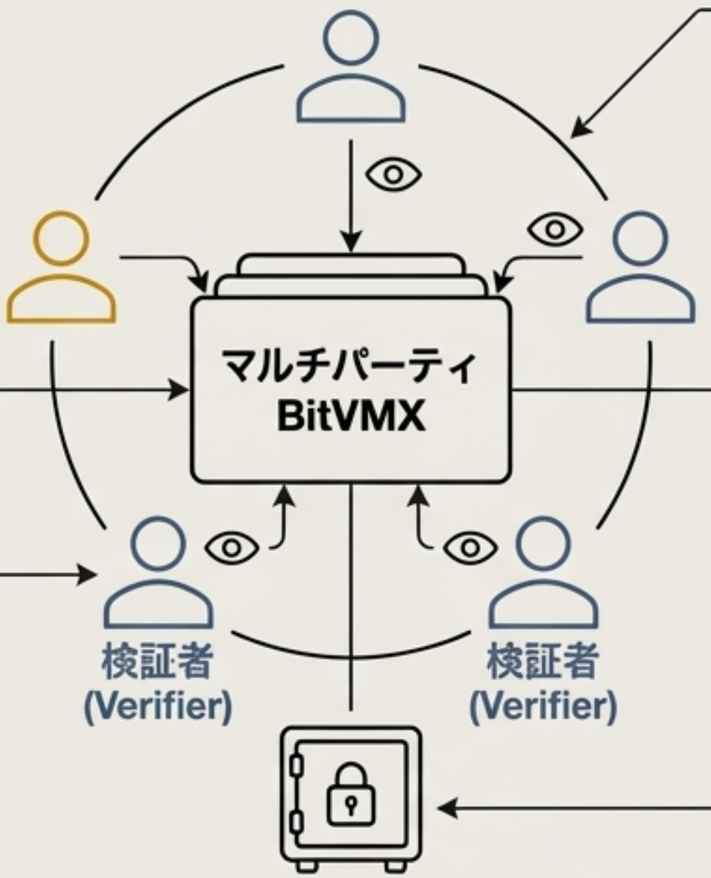
結論: このゲームでは、**最終的に累積PoW最大のチェーンだけが生き残る。**そのチェーン上の出金のみが承認されるため、実質質的に「51%ハッシュパワーが正規チェーンを決定する」というDrivechainの思想をエミュレートしている。



Unionのアーキテクチャ：登場人物の紹介

Bitcoin L1

Union Bridge



運用者 (Functionary) セット

ブリッジを運用する事業者の集合。**N人のうち1人でも正直であれば安全**という「1-of-N honest」モデルを前提とする (BitVM2と同様)。

Peg-out時に、ある運用者が「オペレータ」としてユーザーにBTCを立て替えて支払い、証明を提出する。

他の運用者は「検証者」として、その証明を監視し、不正があれば反証を提出する。

全運用者で巨大な事前署名済みトランザクションDAGに署名し、BTCをロックするVMXOを作成する。これにより、ゲームに勝たない限りBTCを引き出せない状態を作る。

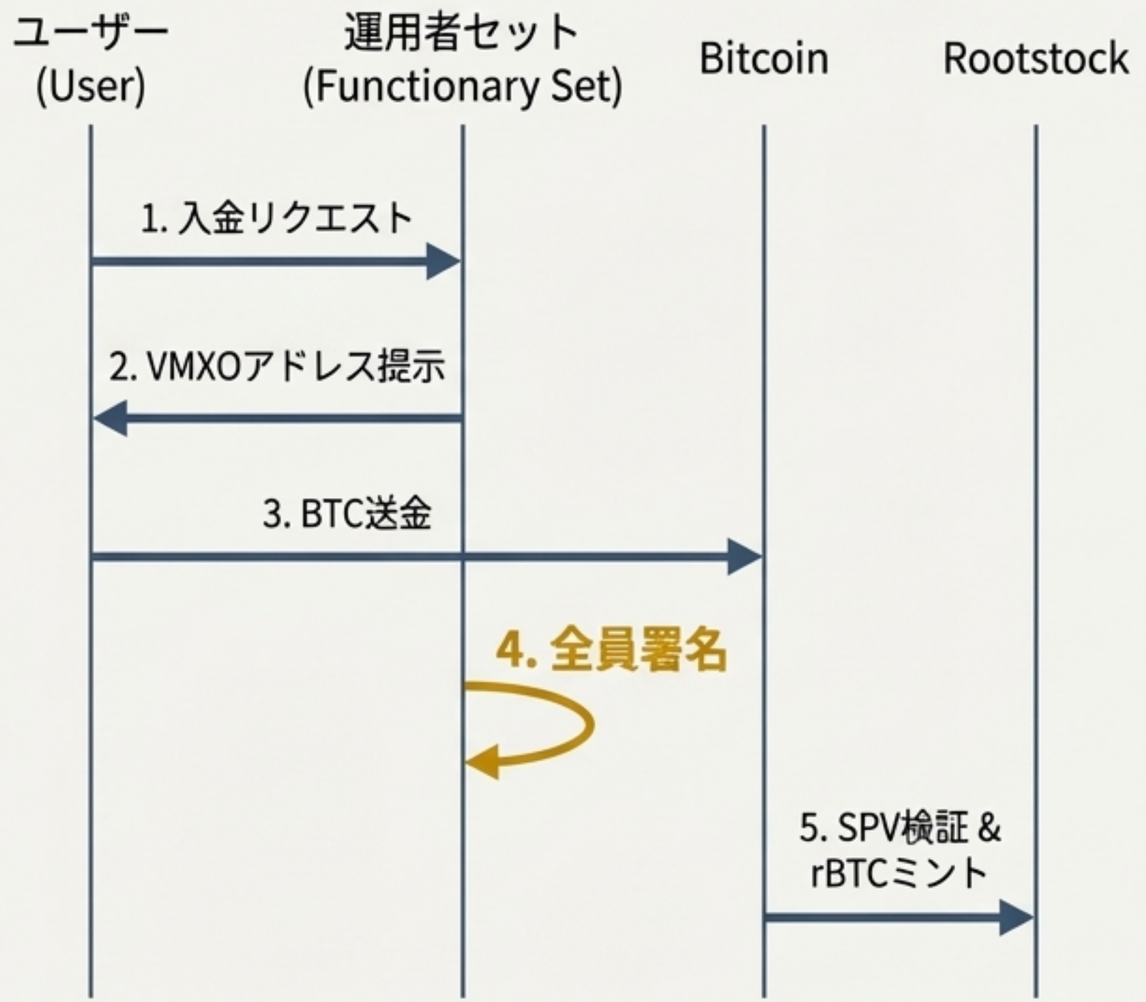
運用者は保証金（Security Deposit）をロックし、不正が証明されると没収（スラッシュ）される。

Rootstock L2

プロセス解説①：Peg-in (Bitcoin → Rootstock)

UnionのPeg-inは、運用者全員の協力が必要な許可制プロセスである。

1. **依頼:** ユーザーが「**X BTC**を入金したい」と運用者にリクエスト。
2. **アドレス生成:** 運用者セット全員が合意し、入金額に対応するVMXO用のBitcoinアドレスをユーザーに提示。
3. **送金:** ユーザーがそのアドレスにBTCを送金。
4. **全員署名:** 全運用者が、このデポジットを有効にするための署名を行う。
5. **ミント:** Bitcoin上でのトランザクションが確定後、Rootstock側がSPV検証を行い、対応するrBTCをミントする。

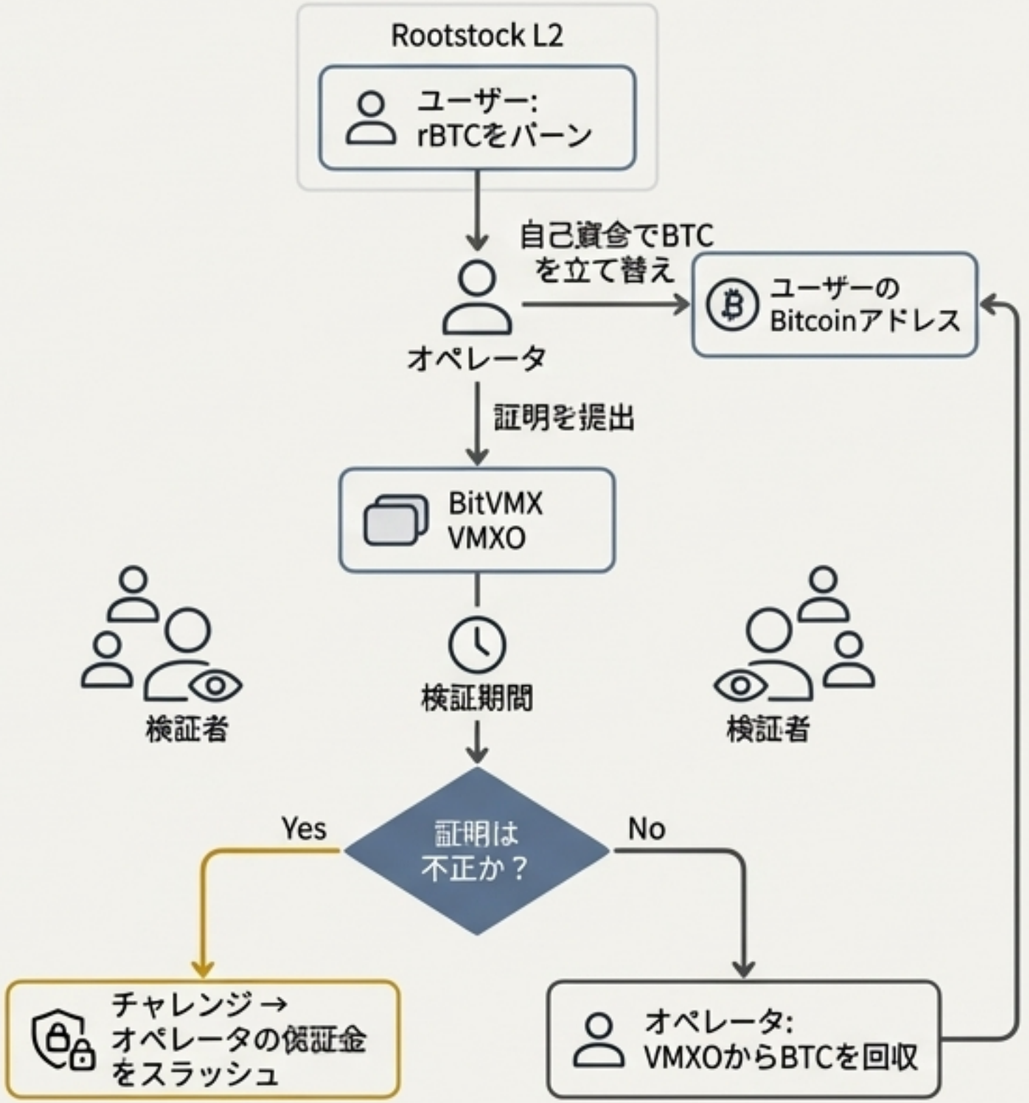


*** 注意点:** このプロセスは運用者全員の署名を必要とする。もし運用者が入金を拒否した場合、ユーザーが自力でBTCを回収できるかけは現時点では不明。UXと検閲耐性の観点でトレードオフが存在する。

プロセス解説②：Peg-out (Rootstock → Bitcoin)

Peg-outは、オペレータによる立て替えと、その後の検証ゲームによって進行する。

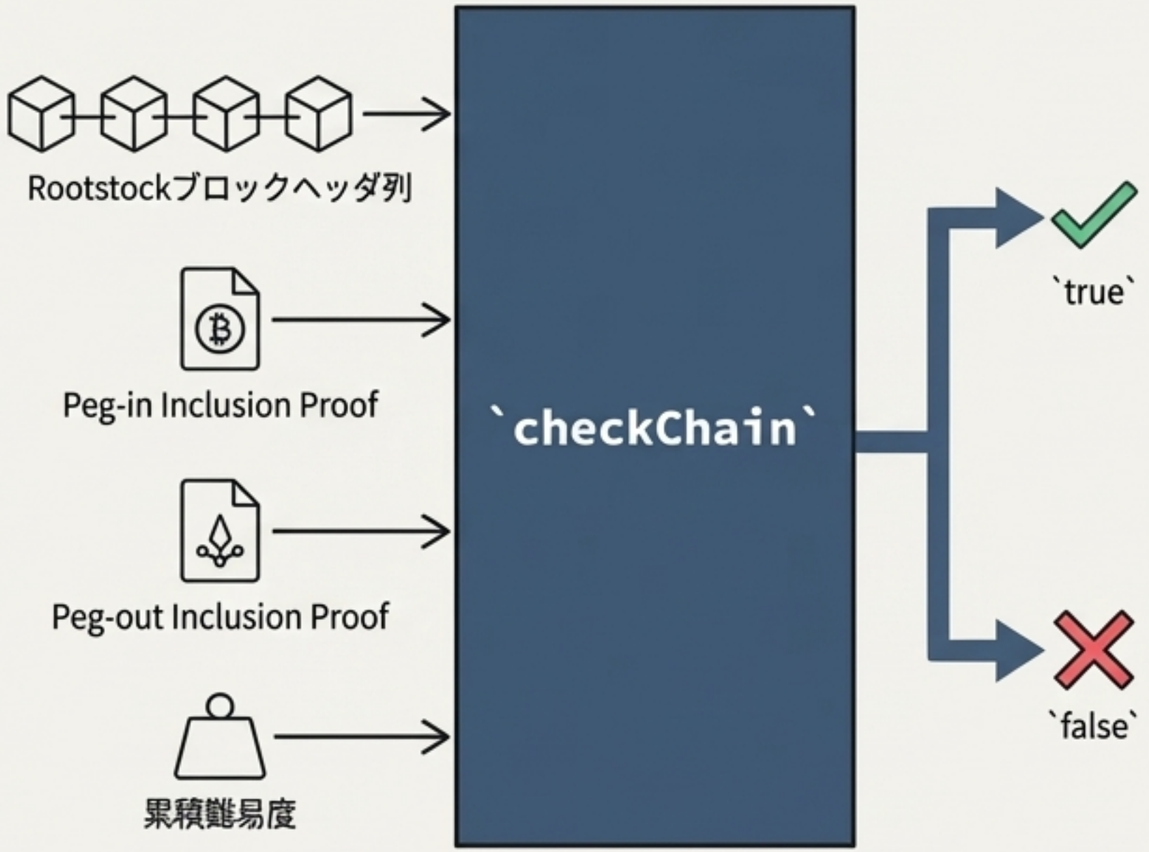
1. **申請:** ユーザーがRootstock上でpeg-outトランザクションを成し、rBTCをバーン（専用アドレスに送付）。
2. **オペレータの役割:** ある運用者が「オペレータ」となり、まず**自己資金から**ユーザーのBitcoinアドレスにBTCを立て替え送金する。
3. **証明の提出:** オペレータは、Rootstock上のpeg-outとBitcoin上の支払いを紐付ける証拠を作成し、BitVMXの検証プロセスを開始する(Kick-off Txを提出)。この証拠には`checkChain´の検証結果がSNARKで圧縮されて含まれる。
4. **検証期間:** 他の運用者（検証者）が、提出された証拠をチェックする。
5. **チャレンジ:** もし証拠が不正（例：偽のチェーンからの出金）であれば、検証者は反証を提出してチャレンジする。
6. **資金回収:** チャレンジがなければ、タイムロック期間後にオペレータはVMXOから立て替えたBTCを回収する。



Unionの心臓部①：`checkChain` — チェーンの正当性を証明する関数

`checkChain`は、提出されたRootstockのブロックヘッダ列が、ブリッジのルールに則っているかを検証する純粋な関数である。BitVMX上で実行される。

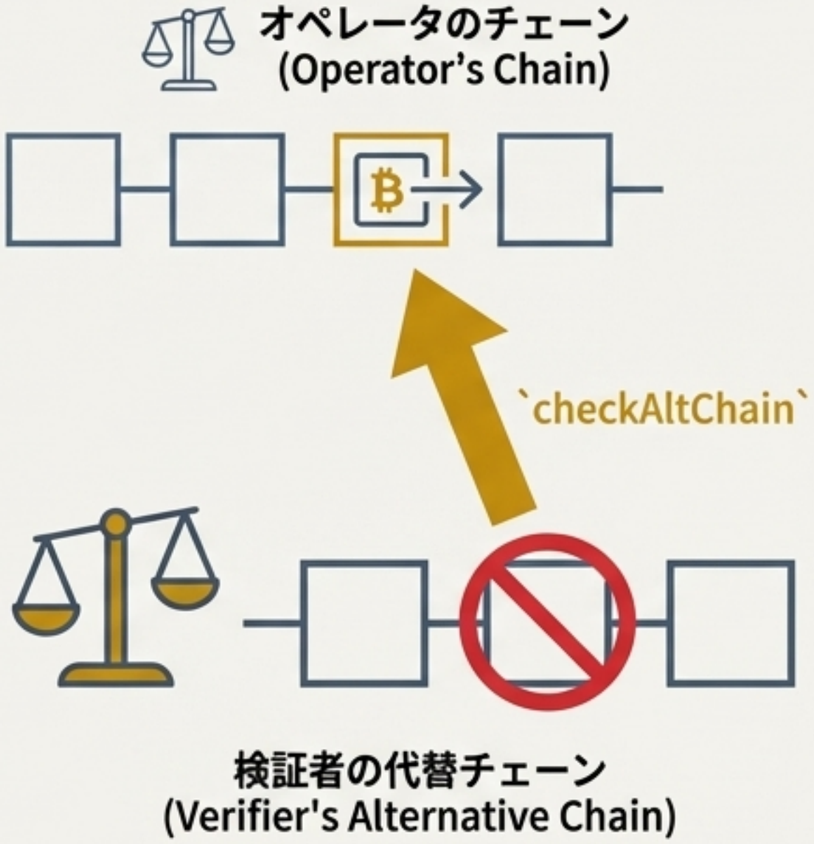
- 入力（Inputs）：
  - Rootstockのブロックヘッダ列
  - 対象となるPeg-inトランザクションのInclusion Proof (Bitcoin上の)
  - 対象となるPeg-outトランザクションのInclusion Proof (Rootstock上の)
  - ヘッダ列の末尾における累積難易度 (Cumulative Difficulty)
- 処理（Processing）：
  - ヘッダ列がRootstockのコンセンサスルール（PoWなど）を満たしているか？
  - 指定されたPeg-inがBitcoin上に実在し、このチェーンに正しく反映されているか？
  - 指定されたPeg-outがこのRootstockチェーン上に正しく含まれているか？
- 出力（Output）：
  - 全ての条件を満たしていれば`true`、そうでなければ`false`。



Unionの心臓部②：`checkAltChain` — より強いチェーンで反証する関数

`checkAltChain`は、オペレータが提出した`checkChain`の主張（「このチェーンは正しい」）に対して、「いや、こちらがより正当なチェーンだ」と反証するための関数である。

- **入力:**
  - オペレータが提示したものとは**別の**ブロックヘッダ列（代替チェーン）
  - **より大きな**累積難易度
  - Peg-inは含むが、問題のPeg-outトランザクションは**含まない**ことの証明
- **目的:**
  - 「こちら（代替チェーン）の方が累積PoWが大きく、より『本物』のチェーンである」
  - 「かつ、オペレータが主張するPeg-outはこの本物のチェーンには存在しない（つまり無効である）」
  - 上記2点を同時に証明する。
- **役割:** これにより、正直な検証者は悪意あるオペレータの不正な出金要求をBitVMXのゲーム内で打ち破ることができる。

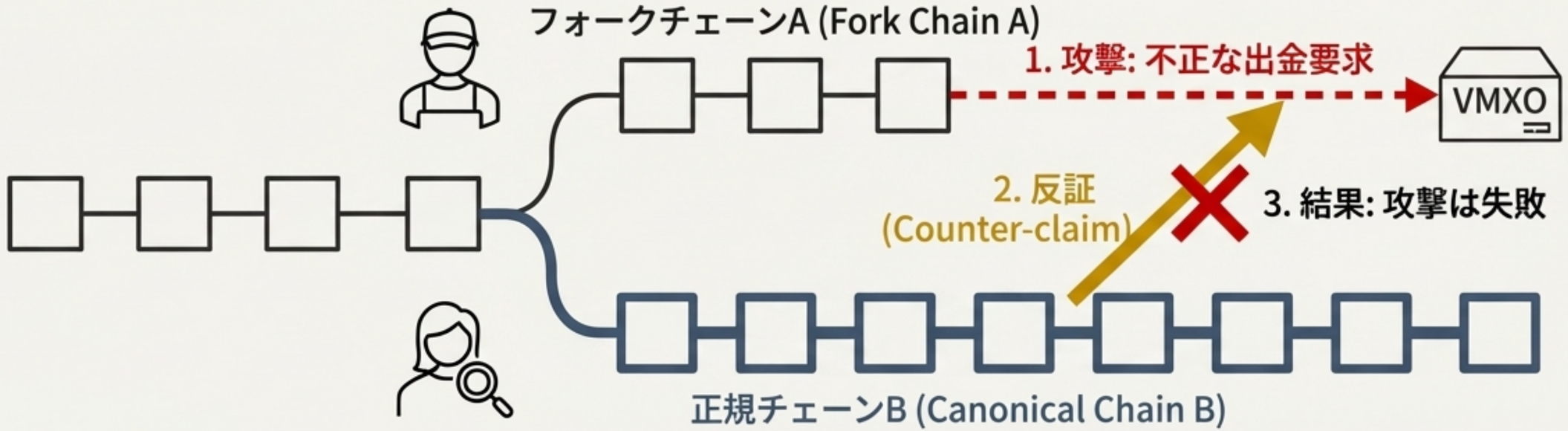


実践：ゲーム理論による不正防止の仕組み

悪意あるオペレータが、自分で採掘したマイナーフォークチェーンAを使い、不正にBTCを引き出そうとするシナリオを考える。

1. 攻撃 (オペレータ): `checkChain(チェーンA)` の証明を作成し、出金を要求。
2. 反証 (正直な検証者): `checkAltChain`を使い、「チェーンBはAより累積PoWが大きく、かつ問題のPeg-outを含まない」と反証。
3. 結果: BitVMXゲームは正直な検証者の勝利。オペレータの出金は却下され、保証金はスラッシュされる。

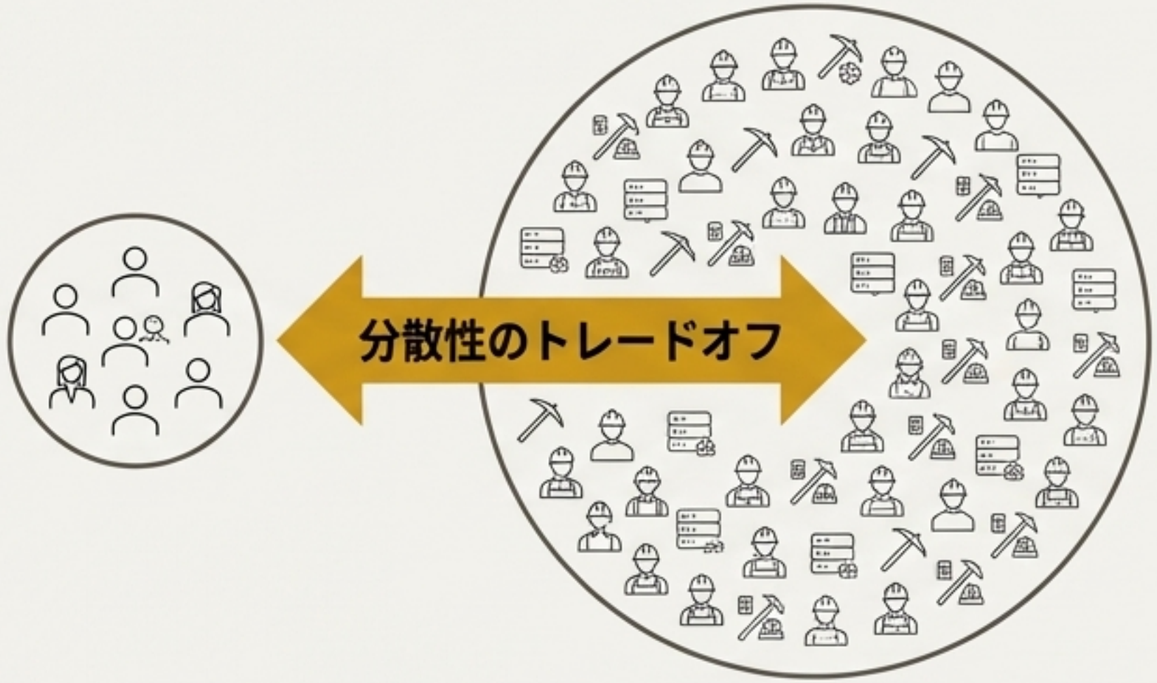
結論 : この設計により、「累積PoW最大のチェーン上にないPeg-outは、必ず誰かに反証される」という強力なセキュリティが実現される。



限界とトレードオフ①：有限な運用者セット

Unionは強力なモデルだが、現実的な制約も存在する。

* 課題：セットアップの複雑さ
  * マルチパーティBitVMXは、参加者全員で膨大な量の事前署名トランザクション（presigned Tx）を生成・共有する必要がある。
  * このセットアップコストは参加人数が増えるほど指数関数的に増大する。
* 結果として生じる制約:
  * 運用者セットは有限かつ比較的小規模にならざるを得ない。
  * 運用者の追加や交代も、再度重いセットアップをやり直す必要があり、簡単ではない。
* **Drivechainとの比較:**
  * Drivechainが「全マイナーが自動的に検証者として参加する」オープンなモデルを目指すのに対し、Unionは許可された小規模なグループに依存する。分散性の観点では明確なトレードオフとなる。

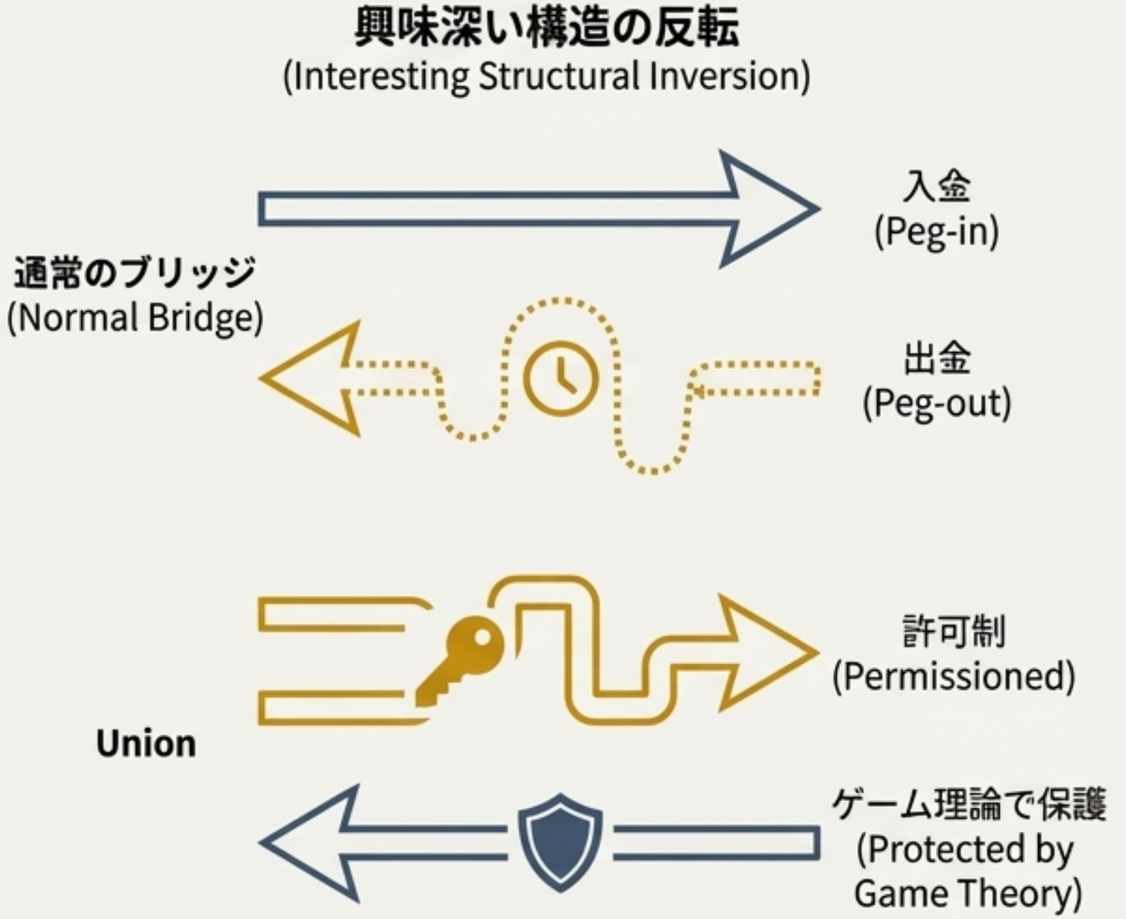


Union:
許可制の小規模セット

Drivechain:
オープンな全マイナー参加

限界とトレードオフ②：許可制Peg-inとUX

* 課題：Peg-inには全運用者の署名が必要
  * ユーザーが自由に、任意の金額をブリッジすることはできない。
  * 運用者セットが共謀すれば、特定のユーザーの入金を拒否（検閲）することも理論上は可能。
* UXへの影響:
  * 既存の多くのブリッジが提供する「アドレスに送るだけ」というシームレスな体験とは異なる。
  * ウォレットやSDKレベルでの高度な抽象化が、実用化の鍵となる。
* 興味深い構造の反転:
  * 通常のブリッジ: 入金は簡単（パーミッションレス）、出金は複雑で時間がかかる。
  * Union: 入金は複雑（許可制）、出金は（正直であれば）ゲーム理論に守られて比較的スムーズ。



まとめ：Unionが切り拓く可能性と残された課題

Unionの位置づけ

- マルチパーティBitVMXと1-of-N honestモデルにより、「誰か一人でも正直な運用者がいればBTCは安全」な構造を実現。
- `checkChain/`checkAltChain`のゲーム理論で、「累積PoW最大のチェーンが正義」であることをBitcoin L1上で強制する。
- これにより、ソフトフォーク無しで**Drivechainのセキュリティ思想をL2でエミュレートする挑戦。**

残されたトレードオフ

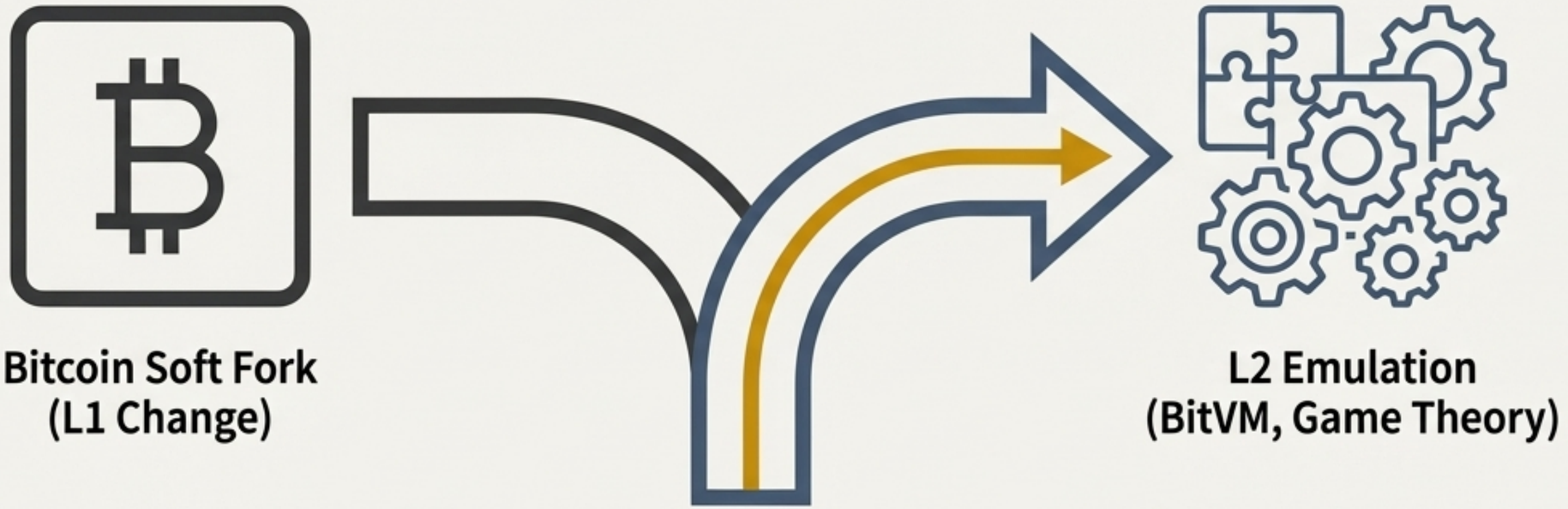
- 運用者セットの有限性（分散性）
- Peg-inの許可制（検閲耐性、UX）
- 運用者の登録・更新メカニズムの具体設計

我々エンジニアにとっての示唆

- Unionは「Bitcoin本体を変更せずに、L2/ブリッジレイヤーの工夫でどこまでトラストを最小化できるか」という技術的探求の最前線である。
- マイナーによるマージマイニングが十分に行われているか、ユーザー側で検証することの重要性が増す。

L1のコンセンサスを、L2でエミュレートする未来

Drivechainを巡る「L1に入れるべきか否か」という長年の議論とは別に、Unionは新たな問いを提示している。
「同様のセキュリティ仮定（51% Hashpower is King）を、L2側の暗号技術とゲーム理論だけでどこまで実現できるのか？」



Unionの試みは、Bitcoinの拡張性を巡る議論において、プロトコルレイヤーの変更だけでなく、オフチェーンの洗練されたメカニズム設計という、もう一つの重要な道筋を示唆している。

今後、実際に検証可能な実装が登場した際、その挙動と堅牢性をコミュニティで検証していくことが極めて重要になるだろう。