ArkとSparkプロトコルを概観する

📖 ビットコイン研究所（日本ビットコイン産業株式会社）
押川 拓夢
2026/3/12

目次・全体の流れ

1. Ark・Spark 超概要

・何ができる?
・LNと違い・関係

2. トランザクションの仕掛け

・Bitcoin Script
・マルチシグ
・事前署名
・タイムロック

3. Arkプロトコル解説

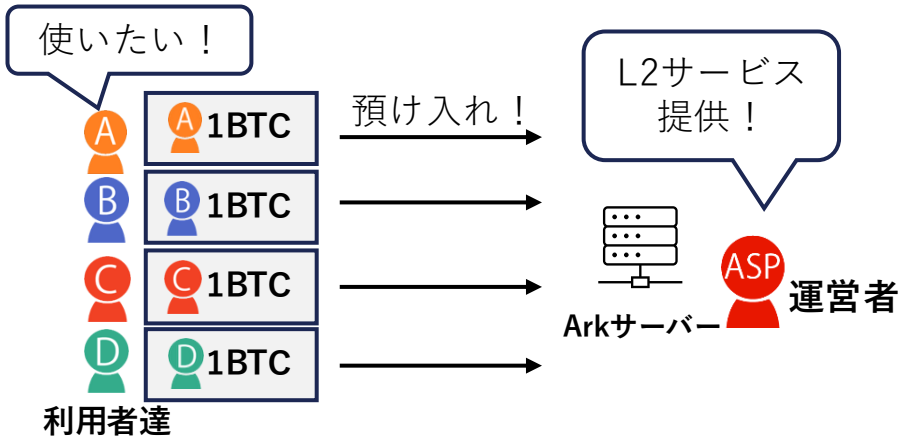
・(背景技術)コベナンツとは？
・コベナンツからArkの誕生まで
・Ark TXツリーの構成
・Arkにおける送金・退出方法
・Ark実装の紹介

4. Sparkプロトコル解説

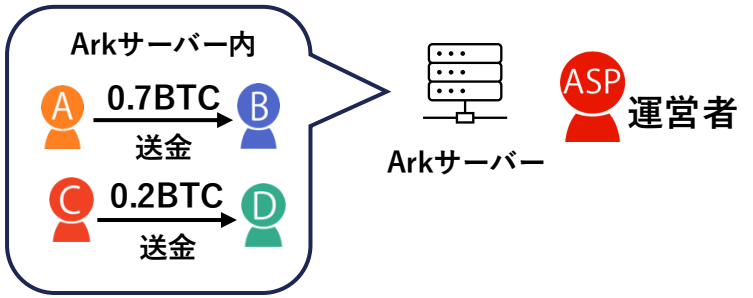
・(背景技術)ステートチェーンとは？
・Sparkにおける所有権・譲渡
・Sparkツリーの構成
・Lightsparkの現状

Ark / Sparkプロトコル超概要

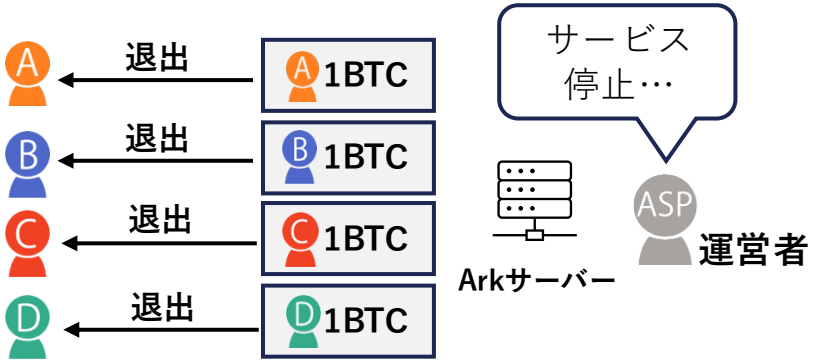
Ark / Spark L2決済サービスを運営
- L2取引サービス利用者を募集
- 利用者が資産を預け入れ



Ark / Sparkサービス上でオフチェーン決済
- 決済サービス上でオフチェーン決済が利用可能



運営の機能停止時や不正時
・利用者は出金してArkサーバーから退出
→ 運営が不正に利用者の資金を奪うことはできない



Ark/SparkとLNとの関係・違い

LNノード運用などの手間を省きL2決済サービス上で取引が可能

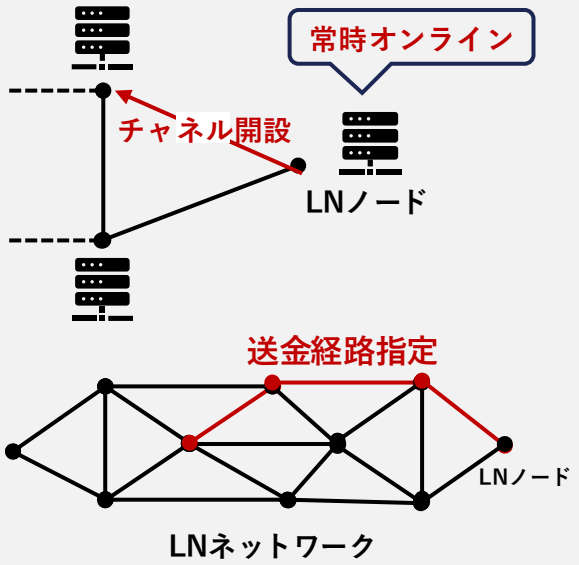
LN送金の問題

・LNはノード運用のコストが高い

・ノードは常にオンラインな必要あり

・チャネル管理、流動性管理、送金経路指定

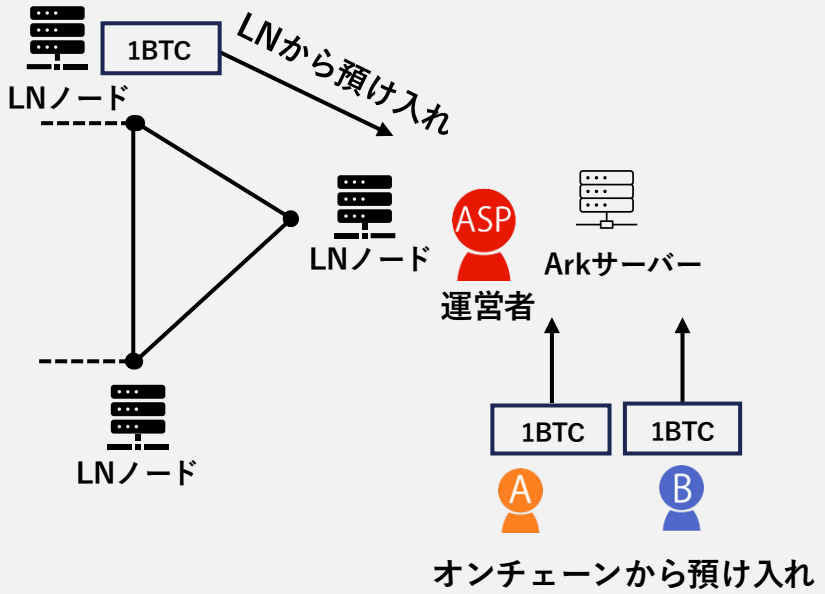
→ 難易度が高く決済UXは悪い



Ark Spark では

Ark / Sparkで解決

利用者は資産を預け入れるだけでL2決済を利用可能！
サービス上で高速決済！



Ark/SparkはLNの代替ではなく補完的な役割のL2プロトコル

Wallet of Satoshiで使われているのがLightspark社が提供するSpark実装の決済API

Ark / Sparkプロトコルの概要

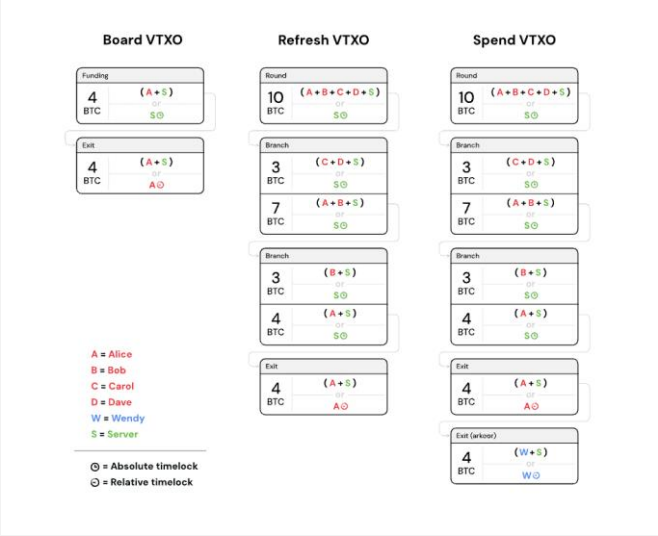
Ark / Sparkは類似点が多いが別のプロトコル

共通点	• サービスプロバイダーに資金を預け入れる • 一方的な出金によりユーザーは資金を守れる • ツリー構造でオフチェーンUTXOデータを管理

構成する技術が異なる

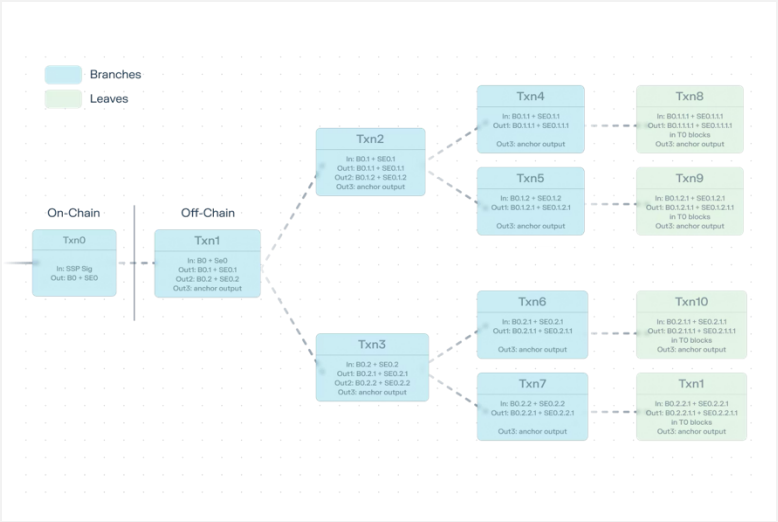
Ark

コベナンツ / 事前署名済みTX



Spark

ステートチェーン
閾値署名(FROST)

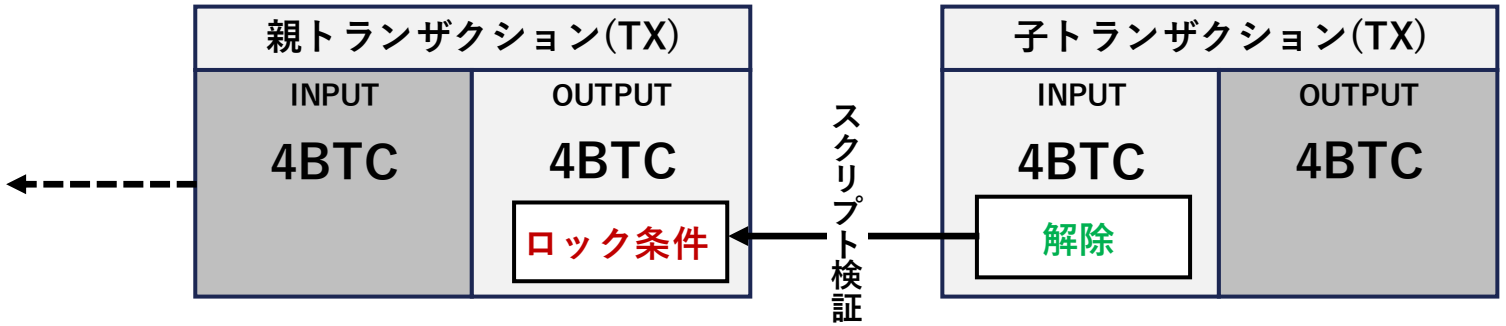


（初学者向け）トランザクションの仕掛け

Bitcoin Script / マルチシグ

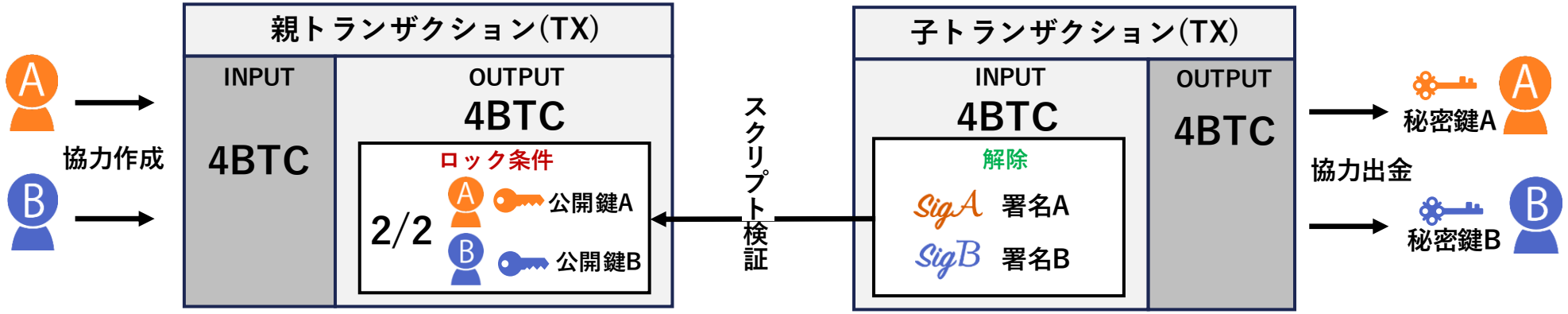
Bitcoin Script （≒ スマートコントラクト）

- 専用プログラムを用いてトランザクションの解除(使用)条件を設定することができる。
- オペコードと呼ばれるスクリプトを用いて記述 (OP_EQUAL, OP_IF, OP_CHECKSIGなど)



マルチシグ

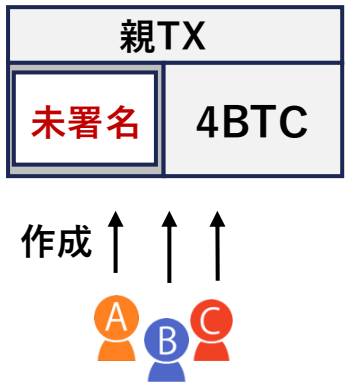
- 複数の鍵でトランザクションをロックし一定の数の署名が集まらないとトランザクション解除ができない。
- 2/2, 2/3, 3/5, 100/100など閾値の設定は自由。



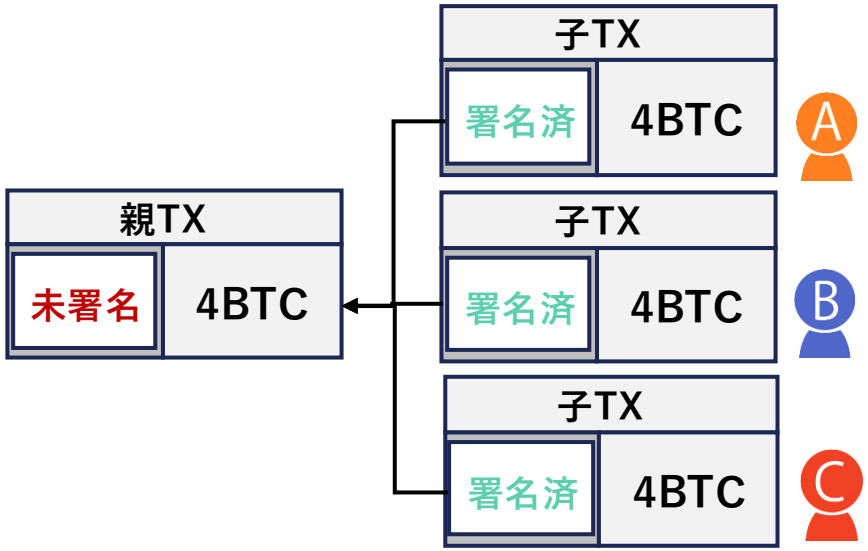
事前署名

事前署名済みトランザクション
署名済みのトランザクションを作成しておき、いつでも解除できる状態で保持
事前にユーザーにTXを渡し署名を繰り返すなどのコミュニケーションコストが高い

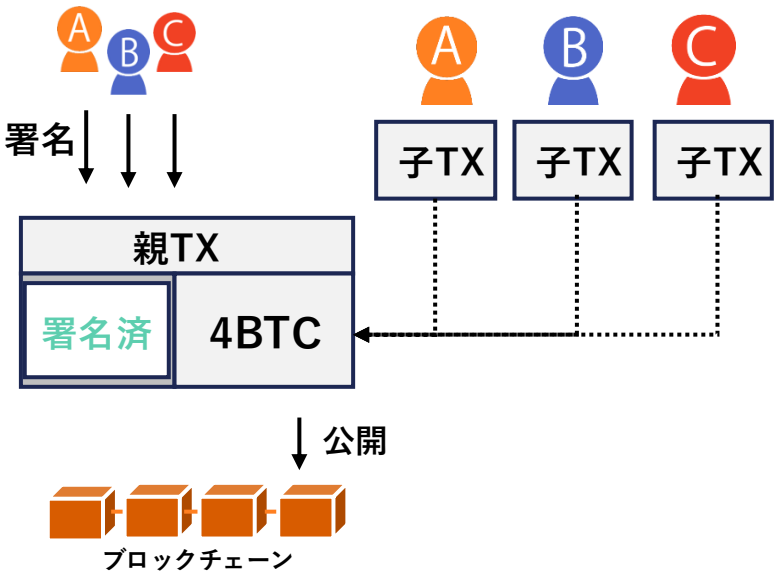
1. 未署名の親TX作成



2. 署名済み子TX先に作成



3. 親TXに署名しオンチェーン公開

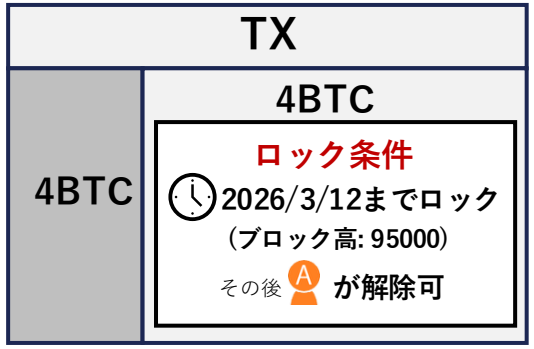


署名済み子TXをいつでも公開可能！

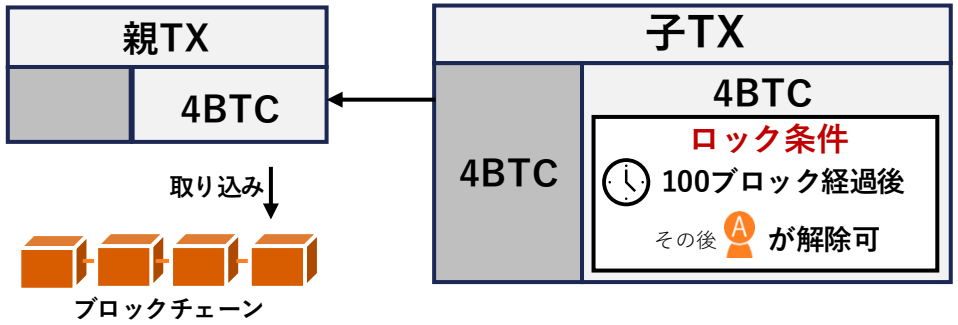
タイムロックとその応用

タイムロック：TXに対して一定期間のロック期間を設け、ロック期間が経過後でないと資金を利用できない

絶対タイムロック
ブロックに取り込まれるタイミングを時間指定

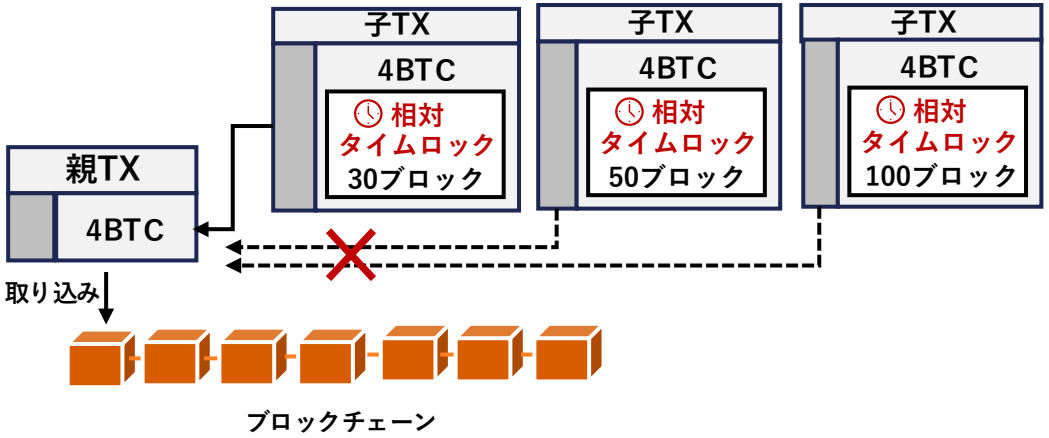


相対タイムロック
UTXOが有効になってから一定の待ち時間指定

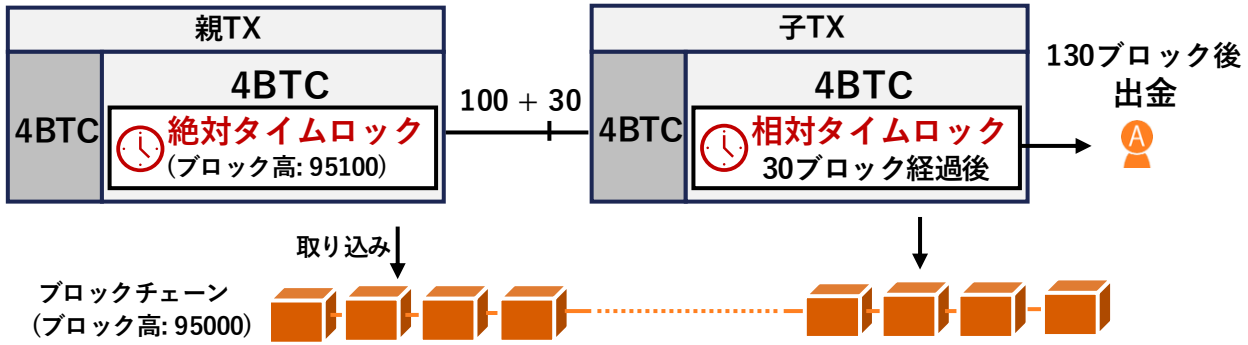


(応用) タイムロックを使ったトランザクションの順序付け

相対タイムロックが一番短いTXだけが有効



絶対タイムロック経過後、相対タイムロックが発動

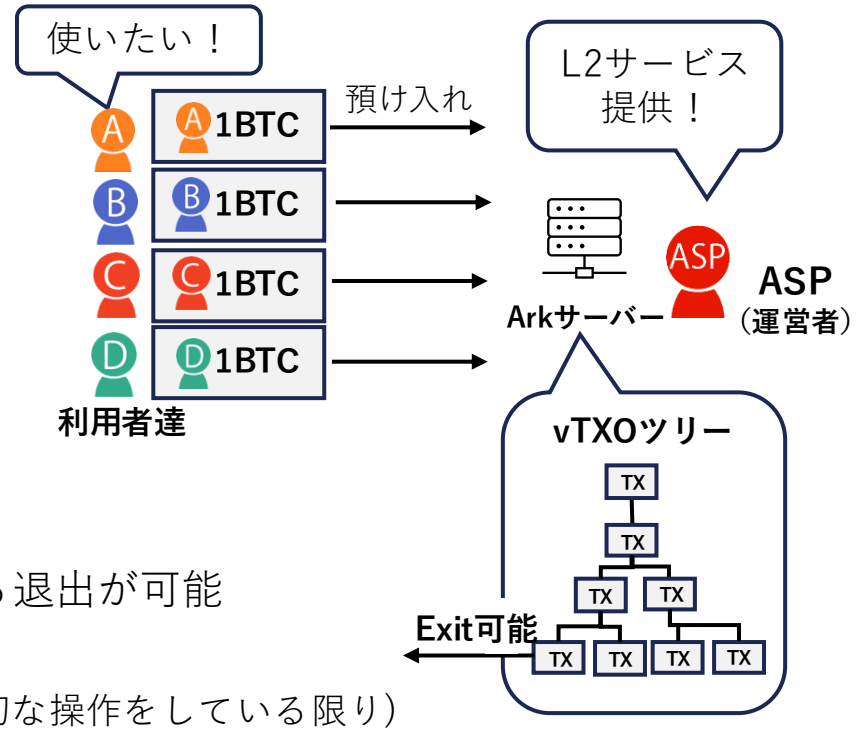


Arkプロトコルの概要

Arkプロトコル概要

運営による決済サービス提供

- **Ark Service Provider(ASP)**が運営者としてL2決済サービスを運営
- ユーザーは所有する資産を預けてArkサービス上でオフチェーン取引
- ASPはユーザー間の取引を二重支払いが起きないように取りまとめ



運営者（ASP）が機能停止 / 不正時の対策

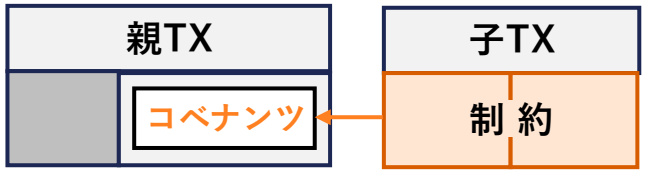
- 各ユーザーが保持する**Exitトランザクション**のオンチェーン公開による退出が可能
  - 強制出金が可能でユーザーの資産は守れる
  - 運営者はユーザーの資金を動かすことができない(ユーザーが適切な操作をしている限り)

Arkを構成する技術

- **コベナンツ(主にOP_CTV)** を用いてトランザクションのツリーを構成(vTXOツリー)
- コベナンツを使わず**事前署名済みTXを用いるコベナンツレスArk(clArk)**も登場

（背景技術）コベナンツとは？

Bitcoinに未導入だが期待されているアップデート



親トランザクションのアウトプットを使用する子トランザクションのインプットと**アウトプット**に制約をかける
例: 使用できるトランザクションを限定する、使用できるアドレスを制限するなど

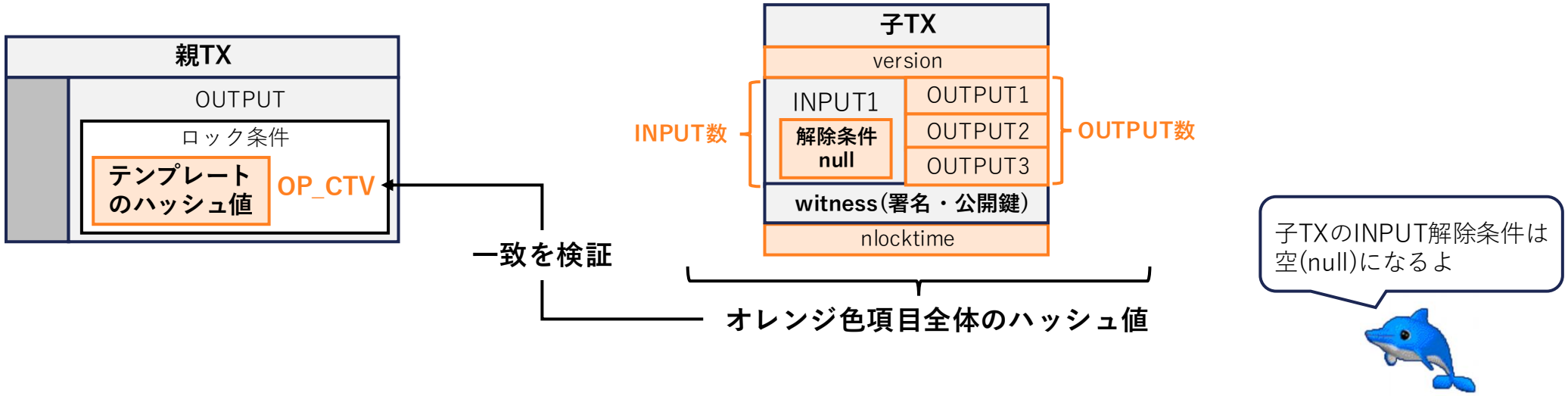
1. **OP_CTV** (BIP-119): 子トランザクションのテンプレートのハッシュ値を親トランザクションで設定（詳しくは次のページで後述）
2. **OP_CAT** (BIP-347):
   - スクリプト処理でのスタック結合を実行
   - コベナンツ専用のアップデートではないが高度なスクリプト構築が可能になり、コベナンツも実現可能
3. **OP_CSFS** (BIP-348): スタック上の任意のデータの署名検証を可能にする
4. **OP_VAULT** (BIP-345): Bitcoinの資金を「遅延付きでしか引き出せないようにする」ためのスクリプト機能

など

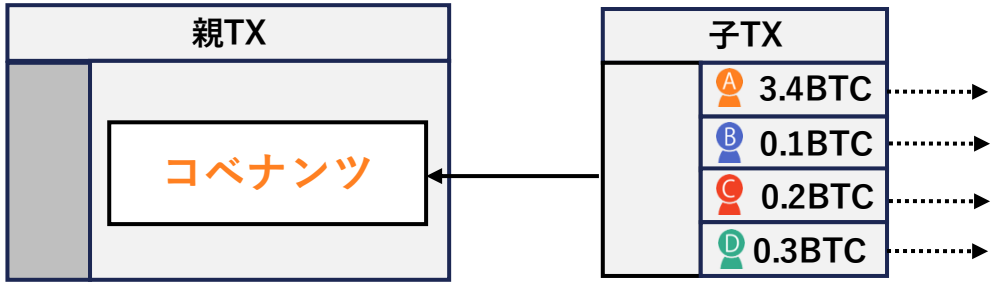
事前署名で類似することができるが対話コスト・トラストリスク・ストレージ要件を軽減可能

（背景技術）OP_CTV(OP_CHECKTEMPLATEVERIFY)とは？

- 親TXで「子TXのテンプレート」と「実際の子TXにおける特定の項目のハッシュ値」の一致を検証
- 「この形のトランザクションしか作れない」という**テンプレート**をロック条件に設定



子トランザクションで「誰に・いくら払う・そのロック条件は」(アウトプット)をあらかじめ未署名で限定できる



しかし、**コベナンツ単体では払い出ししかできない**

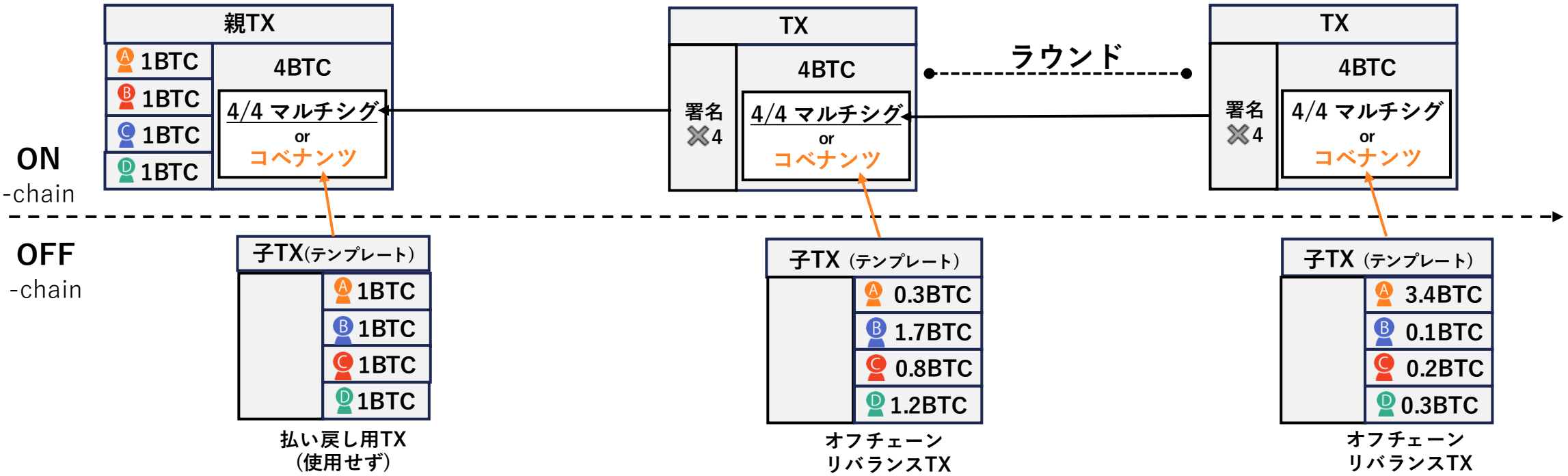
OP_CTVの活用(1) -コベナンツとマルチシグによるオフチェーン取引-

マルチシグでの解除条件を追加することで資金のオフチェーンでのリバランスが可能になる

1. 複数ユーザーが参加して資金をプール

2. 参加者同士で取引しオフチェーンで資金のリバランスを行う

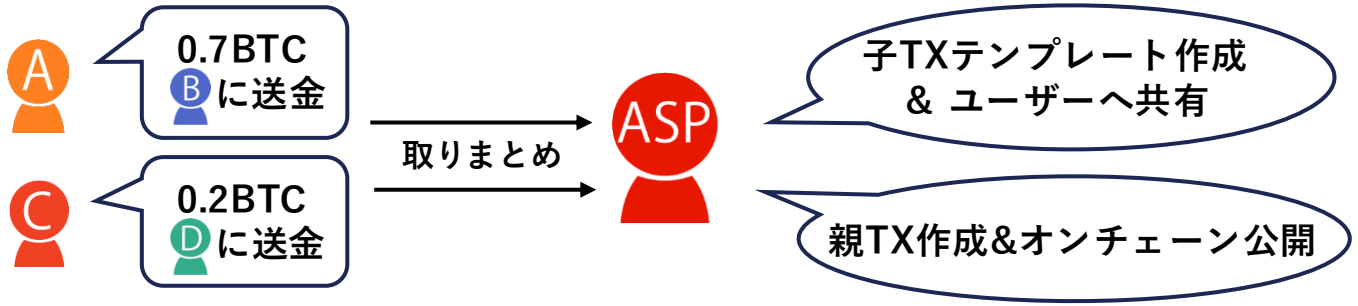
3. **一定間隔(ラウンド)でリバランスを行うことでオフチェーン取引実現**
→ オンチェーンTX数を削減可



OP_CTVの活用 (2) –コベナンツによるオフチェーン取引の問題-

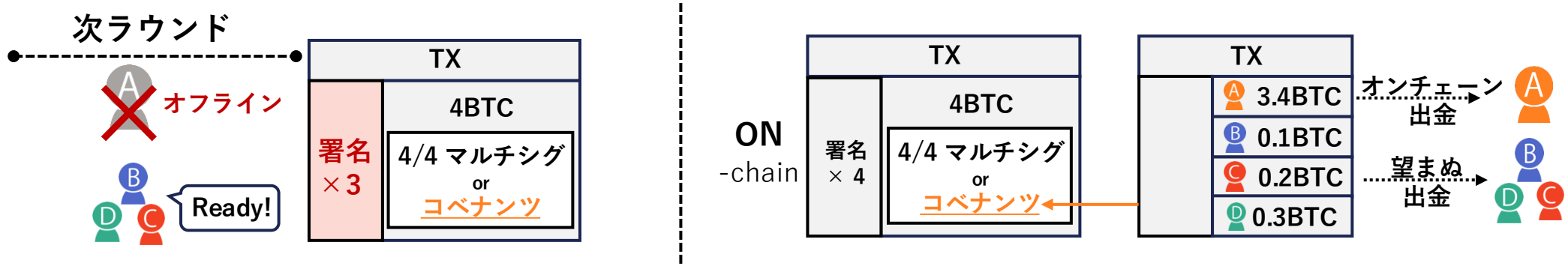
参加者の送金額をまとめてオンチェーン・オフチェーンTXを作成するのが大変…

→ **Ark Service Provider（ASP的な存在）のようなコーディネーターが必要になる**



大問題

参加者の一人でもオフラインorオンチェーン出金するとリバランスができず、全員強制オンチェーン払い出し



次ラウンドに不参加のメンバーがいてもリバランスを続けたい…

OP_CTVの活用からArk誕生へ -トランザクションツリーの構成

コベンナンツを使った2分木でトランザクションのツリーを作成

Round TX(オンチェーン)

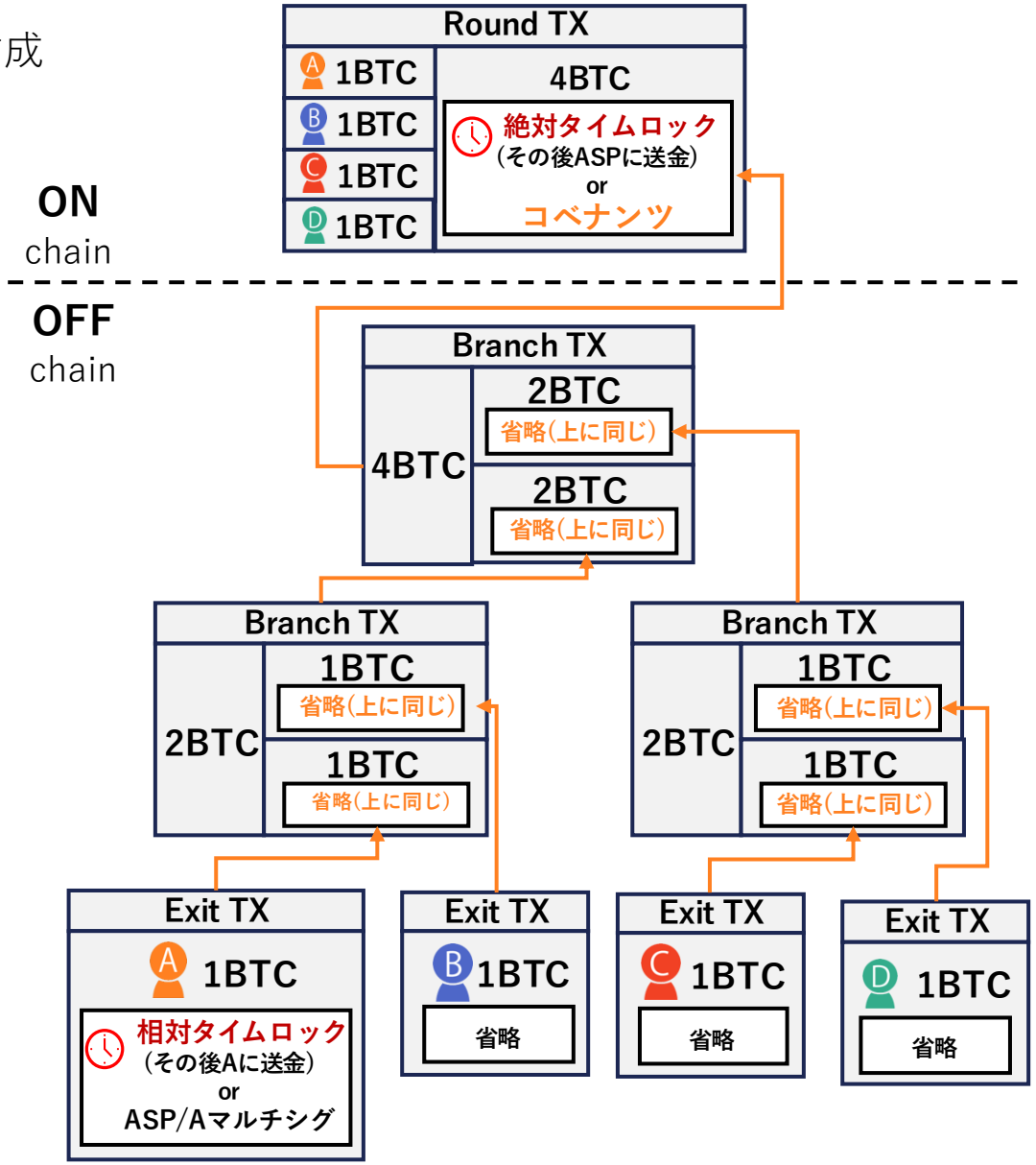
- ラウンドの期限を表す絶対タイムロックをセット
- ラウンドごとに新たに作成しオンチェーン公開

Branch TX (オフチェーン)

- Round TXから個人用Exit TXまで分割するための中間TX
- 参加人数が多くなるほどBranch TXの数は増える

Exit TX (オフチェーン)

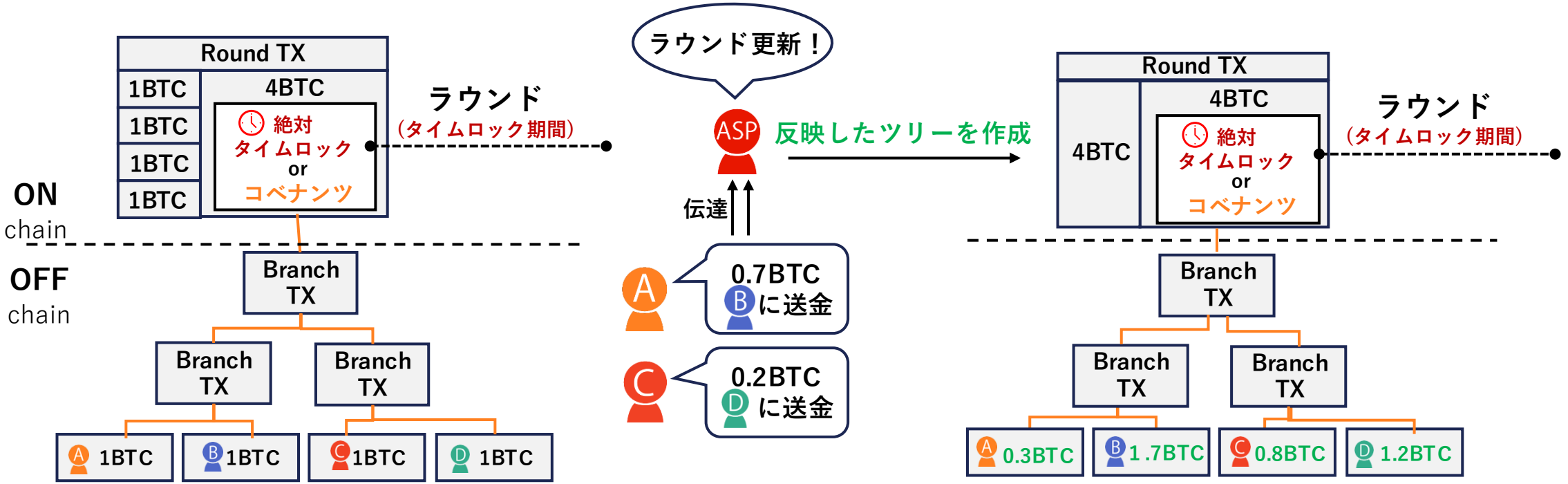
- ツリーの末端のリーフでユーザーごとに払い出し
- Exit実行の待ち時間である相対タイムロックをセット



Arkにおける２種の送金手法 –ラウンド送金-

ラウンド送金

- ラウンドごとに新しいArk TXツリーを作成し、ラウンド更新タイミングでASPが各ユーザーの資金額を調整
- 頻繁なラウンド更新(10分に一回とか)はコストが高すぎる
- **ほとんど使われない送金方法**
    - **トラストレスな送金は実現可能**



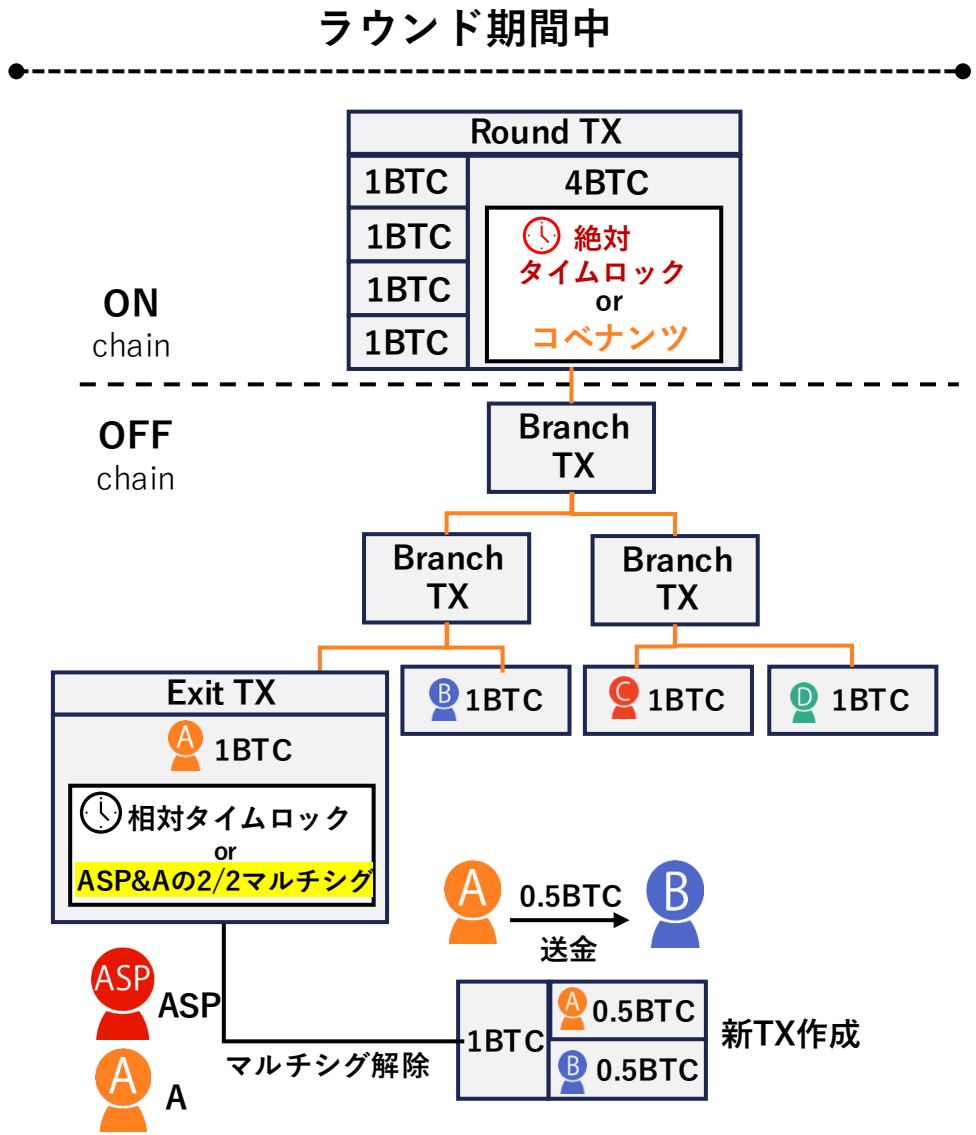
Arkにおける２種の送金手法 - OOR送金（ラウンド外送金）-

OOR送金(out-of-round送金,ラウンド外送金)

- Arkにおける主な送金手段
- **送金にあたってASPとの共同署名の協力が必要**
    - **ASPへのトラストが発生**
- ASPの協力によりいつでも高速送金が可能

送金手順

1. 送金ユーザーのUXTO(**vTXO**)を分割して新たなTXを作成
2. ASPは二重支払いが無いように調整し署名
3. 送金ユーザーは送金額に問題なければ署名



一方的退出方法(Exit)

ユーザーによるExitの実行

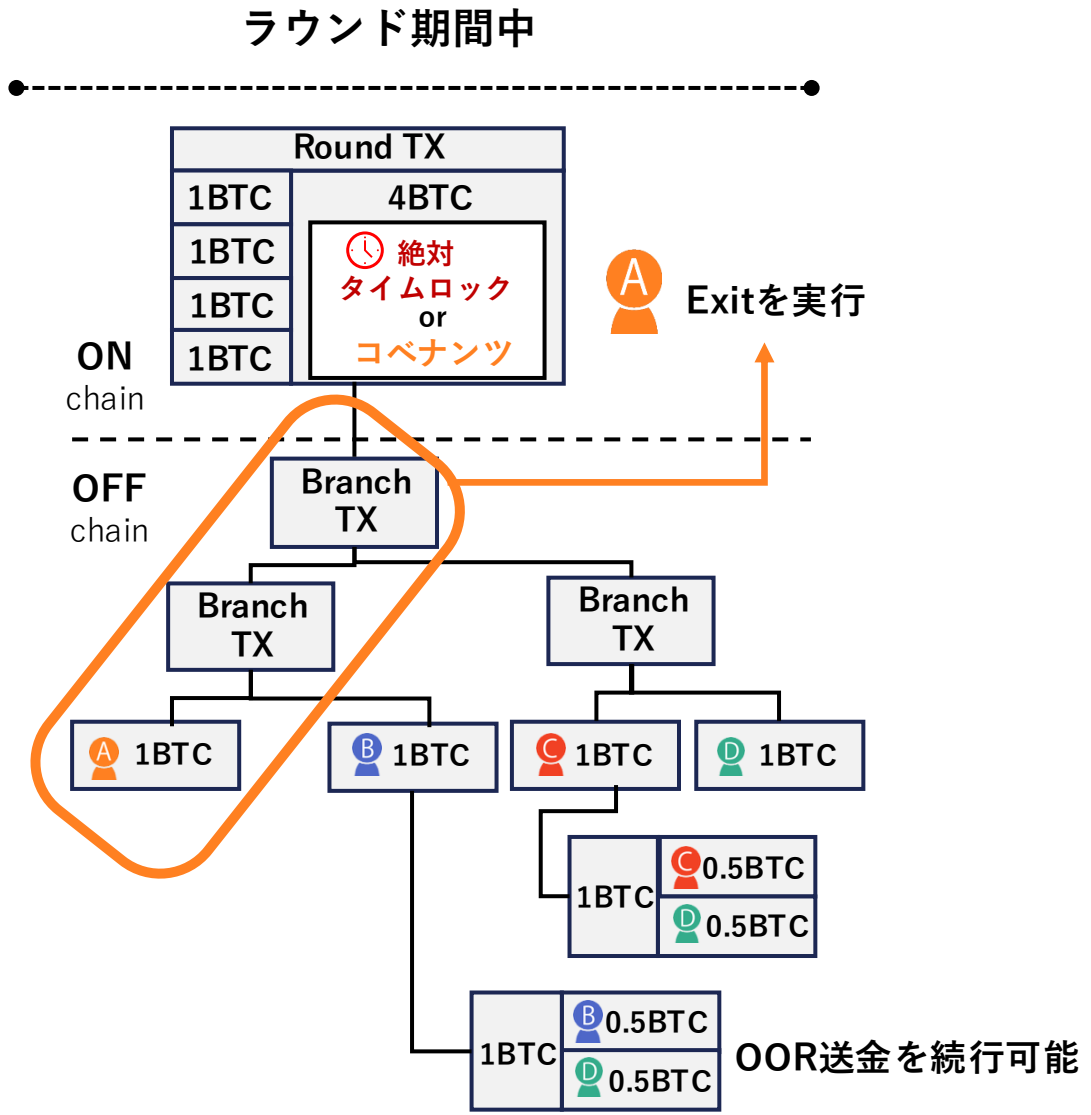
- ユーザーが自身のExit TXとその親のBranch TXを
  ユーザー自らオンチェーンに公開
- ユーザーによる回収には絶対+相対タイムロックが発生

他のユーザーへの影響

- 他のユーザーはOOR送金でオフチェーン取引を続行
- 退出済みユーザーなしで次のラウンドに移行

手数料が高くなる

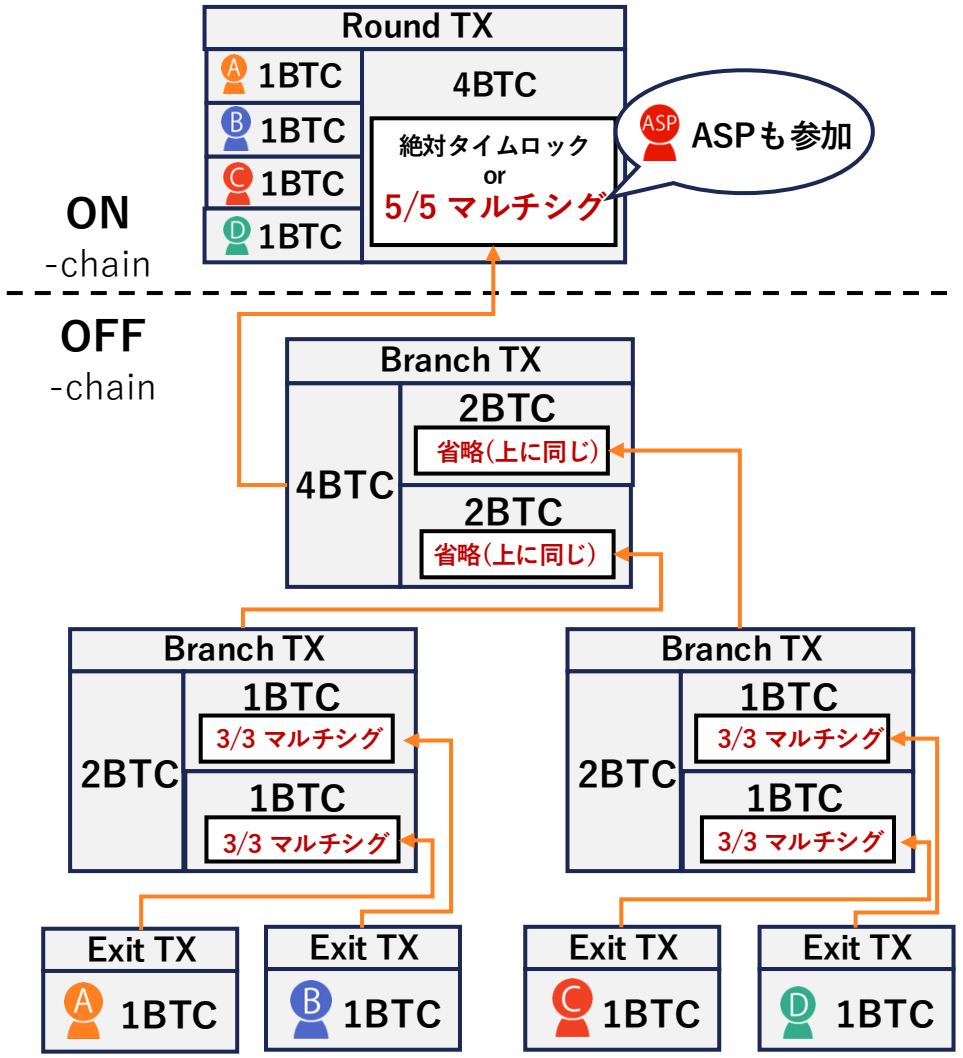
- Branch TXのオンチェーン手数料が必要
- アトミックスワップなどを用いてASPがユーザーに
  出金する**協力的退出も可能**



clArk（コベナンツレスArk）とは？

事前署名を用いてコベナンツなしでArkを実現

- **コベナンツの代わりに事前署名TXを協力して作成**
  - ユーザー + ASPのマルチシグ
- 各ユーザーが多数のTXへの署名が必要
  - コベナンツ有りArkと比較しコミュニケーションコスト大



Arkの現在地

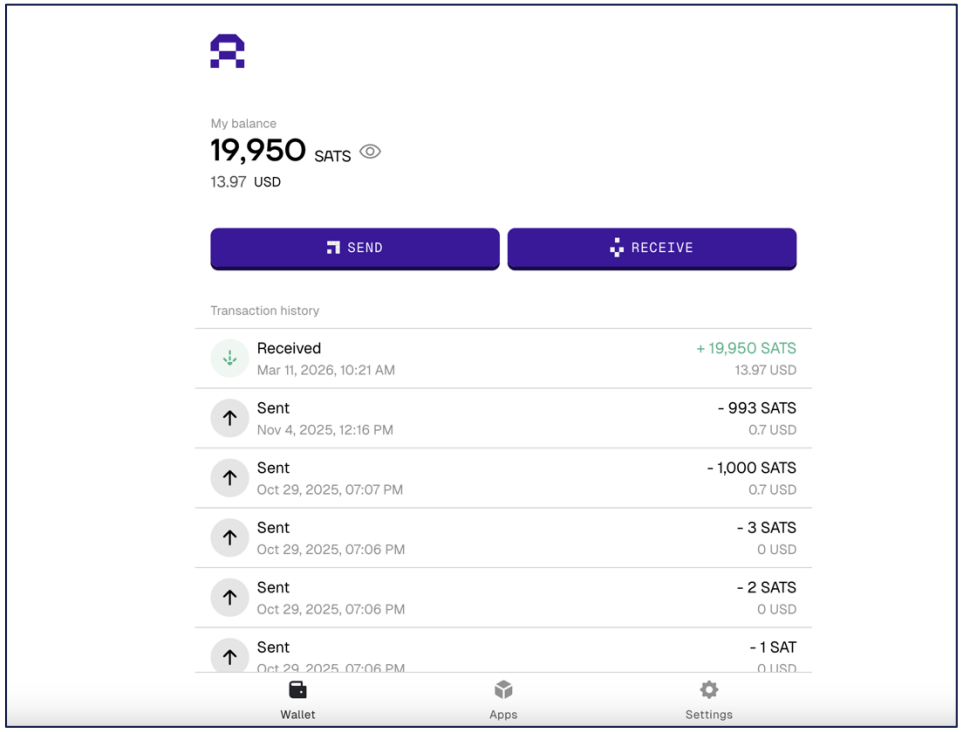
Second社のBarkとArk LabsのArkadeの２つの実装がOSSで公開

- Barkはシンプルなcl Ark実装でL2決済特化
- ArkadeはDefiの実現のためかなり複雑

	Second	Ark Labs
実装名	Bark	Arkade
目的	L2決済	Defi スマートコントラクト
Lightning決済 サポート	あり	部分的にあり (サブマリンスワップ対応)
特徴	決済に特化	・スマートコントラクト用 拡張Bitcoin Script (Arkade Script) ・独自トークン発行 (vTXOツリー上でのOpen Asset Protocolようなもの)

ブラウザWallet 「Arkade Wallet」でArkを体験可能

https://arkade.money/



Arkadeのエクスプローラーもある

https://arkexplorer.blockonomics.co/

Sparkプロトコルの概要

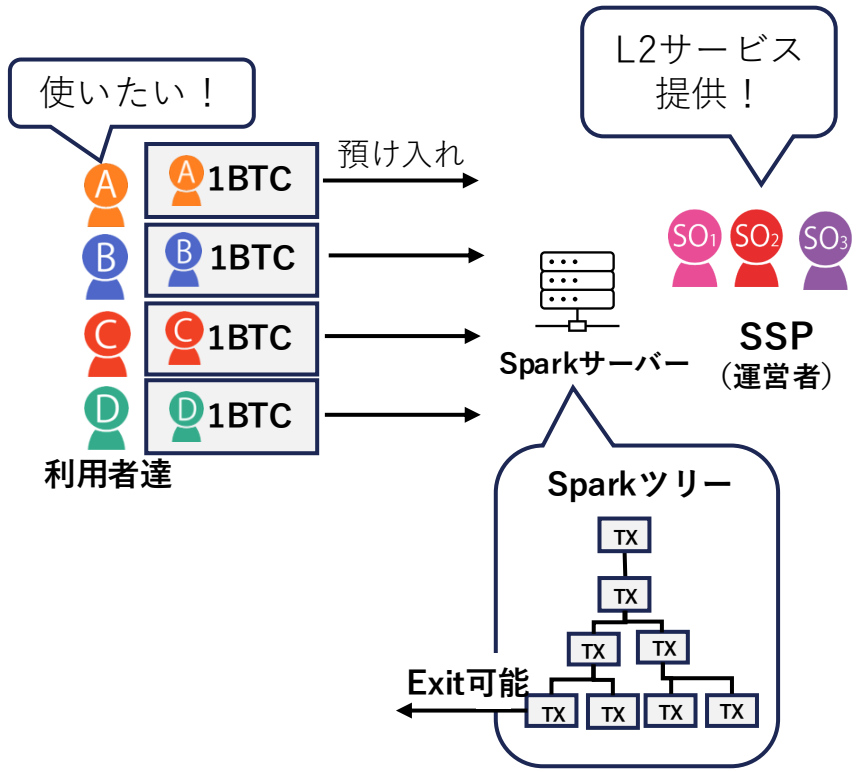
Sparkプロトコルの概要

Arkと同様の機能を実現可能

- 基本的にArkと同様で**Spark サービスプロバイダー (SSP)がL2決済サービスを運用**
- 利用者の資産をSparkサービスに預ける
- 運営の機能不全・不正発生時にユーザーによる**Exitトランザクションによる一方的退出**が可能

Arkとの構成技術の違い

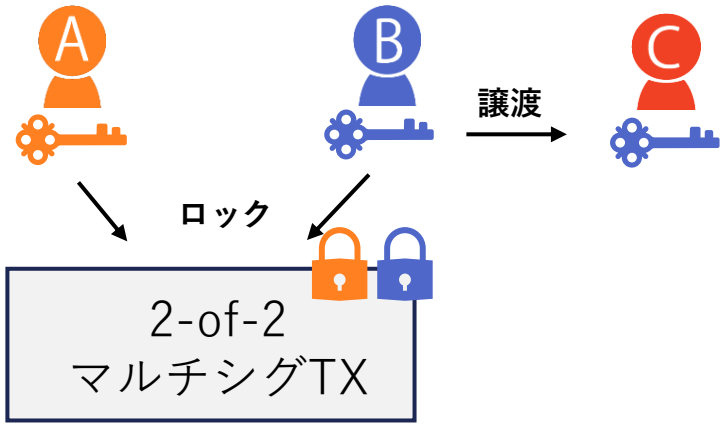
- ステートチェーンによるUTXOの所有権移転(送金)を実現
- 複数のSparkオペレーター(Sparkの運営者)間で
  閾値署名(FROST)を使い運営者の不正リスクを抑える
- ラウンドの概念はない



(背景技術) Statechainとは？

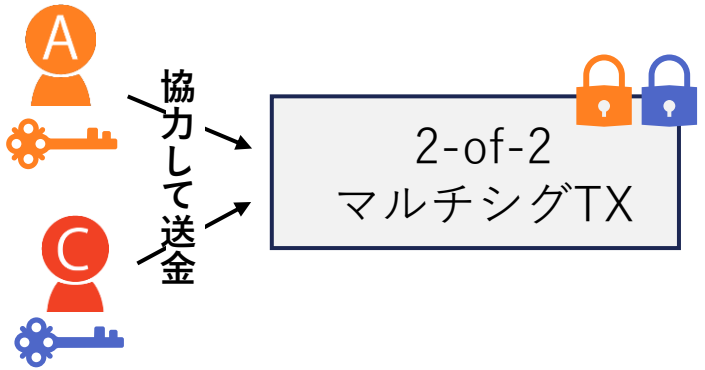
秘密鍵を渡すなどの方法で**UTXOを使用する権限を別のユーザーに移すことでオフチェーンでの資金の移動**を実現

2-of-2マルチシグで片方が秘密鍵を譲渡することで



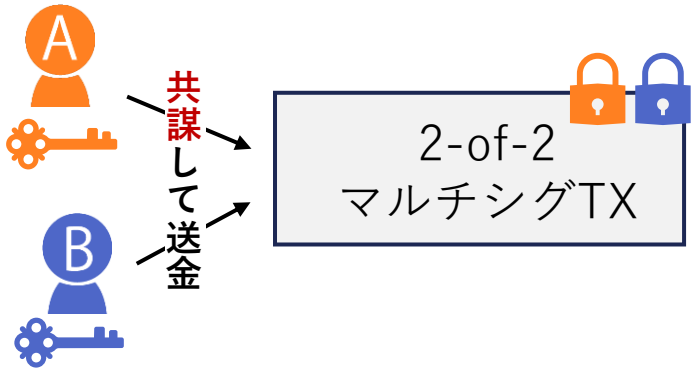
正常

譲渡された秘密鍵で送金



異常

前の秘密鍵所有者と共謀されること危険もある

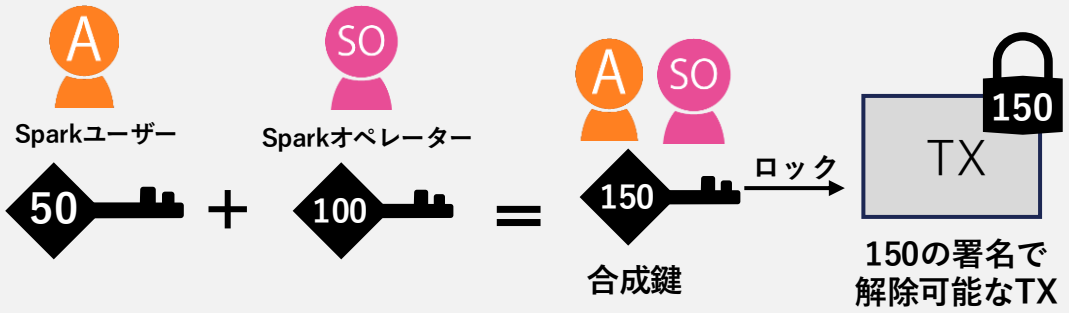


前の所有者

Sparkにおけるトランザクション譲渡(送金)の実現方法

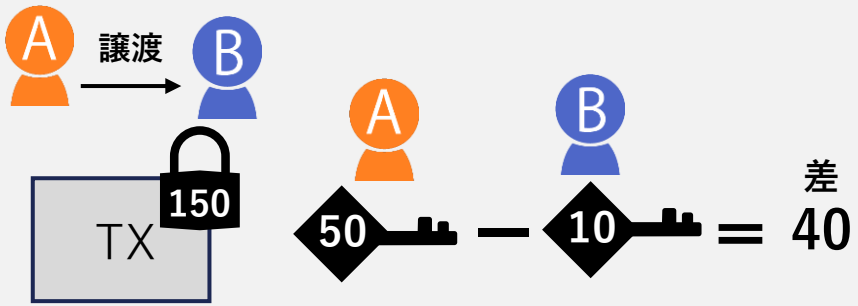
Sparkではトランザクションの所有権譲渡(送金)をSchnorrの合成鍵の譲渡を実現

1. それぞれの鍵を元に合成鍵の生成しUTXOをロック

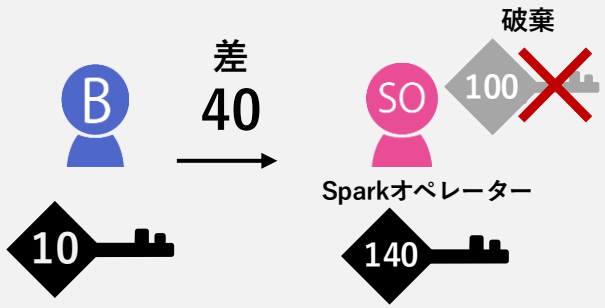


AからBに送金する場合

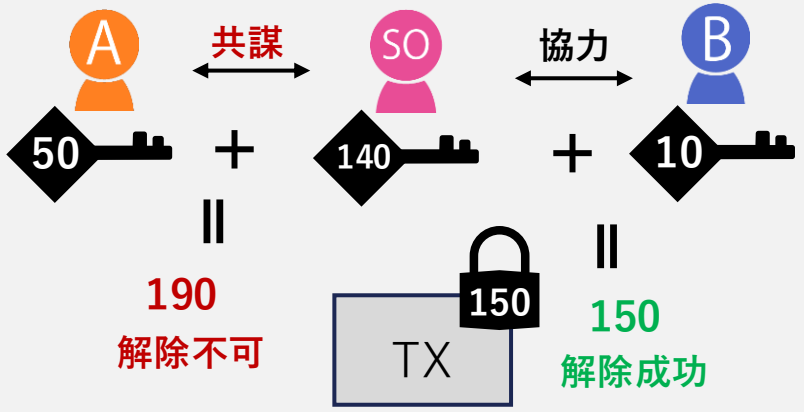
2. Bは自身の鍵を生成し、Aの鍵との差を計算



3. 差を元にオペレーターは自身の鍵を調整し、
新しい鍵を作成。オペレータは前の鍵を破棄。



4. 前のユーザーとオペレーターが共謀しても
資金のロック解除ができない。



Sparkのトランザクションツリーの構成

Sparkツリー

- Ark同様に一つのルートUTXOを分割してツリー構成
- 末端のExit TXで退出が可能なように中間TX作成

Branch TX

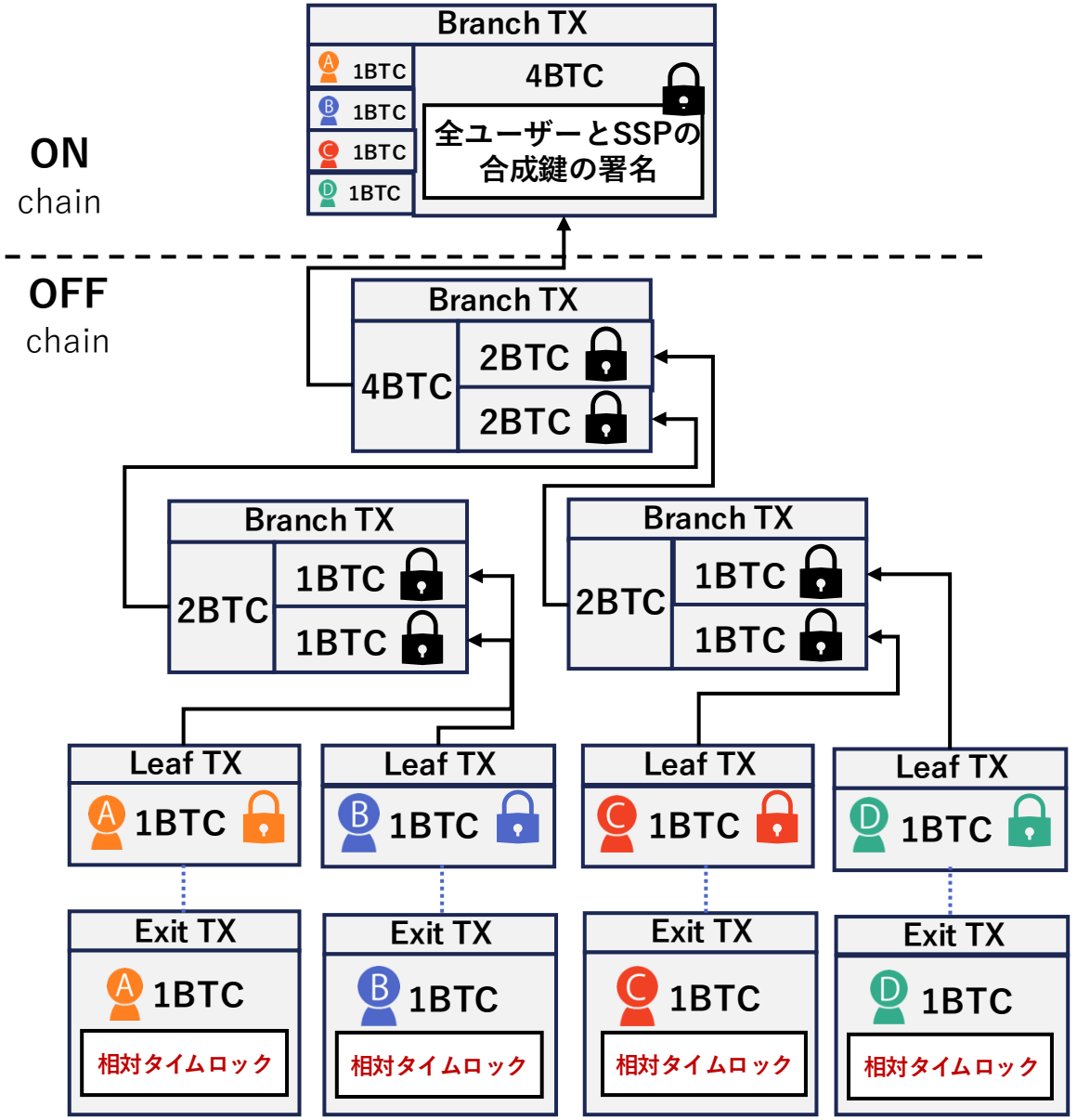
- ツリー構造に分割するための中間TX
- その下のすべての Leaf 鍵の合計鍵でロック

Leaf TX

- 前述したユーザー+Sparkオペレータの集約鍵でロック
- 送金の際はSparkオペレータの共同署名が必要

Exit TX

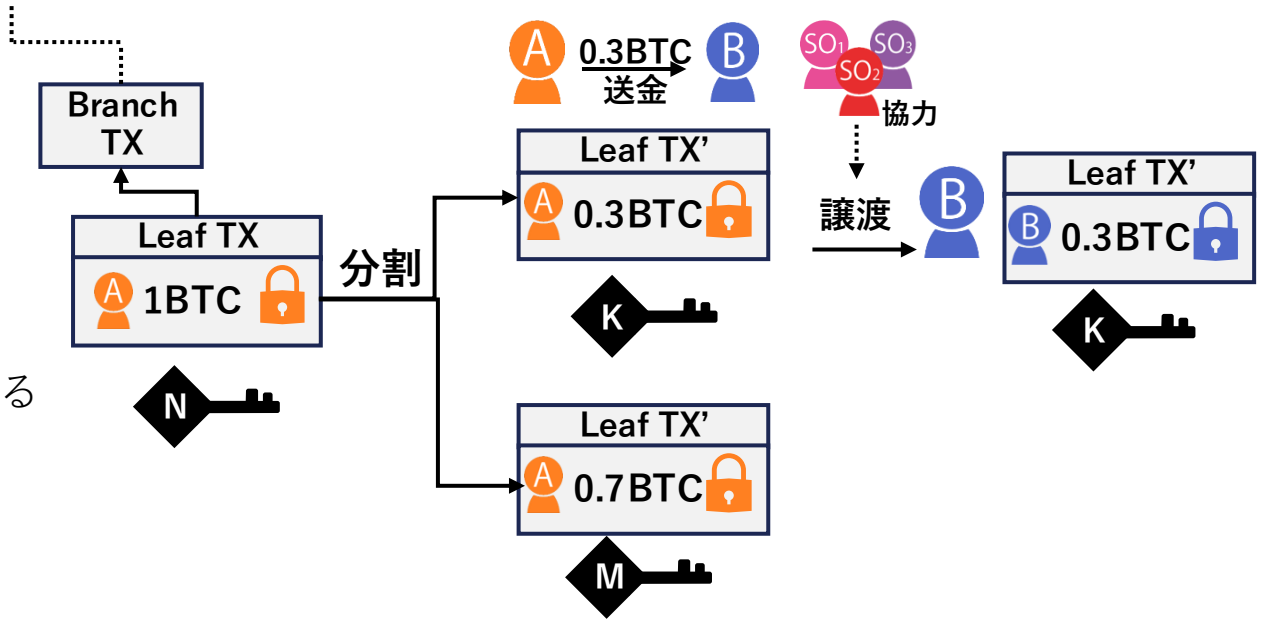
- Ark同様一方的退出用のTX
- 一方的退出は親TX(BranchとLeaf)を公開する必要あり
- ユーザーとオペレータが協力して出金TXを作る
  協力的退出方法ももある



Sparkの特性

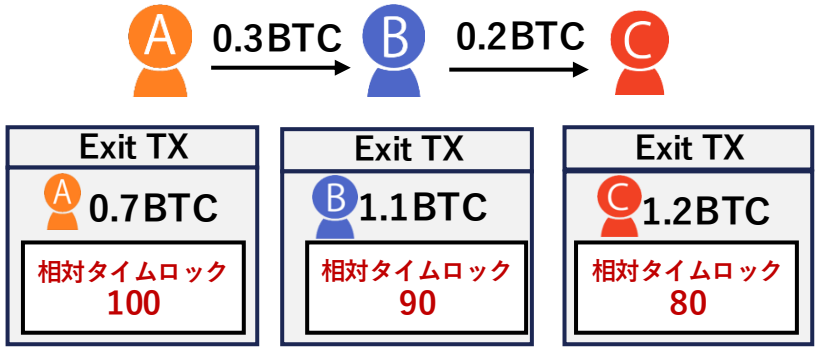
鍵とLeaf TXの分割による送金

- Schnorrの特性を利用した鍵を分割することが可能
- 送金時、Leaf TXを分割
- 同時に鍵を分割し譲渡(送金)
    - ここで前述した集約鍵譲渡のスキームを用いる



相対タイムロックによるExit TXの順序付け

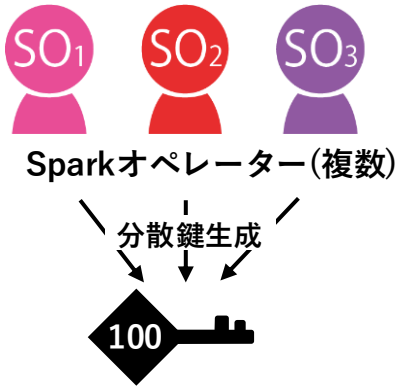
- 送金のたびに受金者のExit TXを更新
- 相対タイムロックを減少させることで最新のTXが先にオンチェーンで使用可
- **悪意のある古いExit TXが公開されないように抑制**



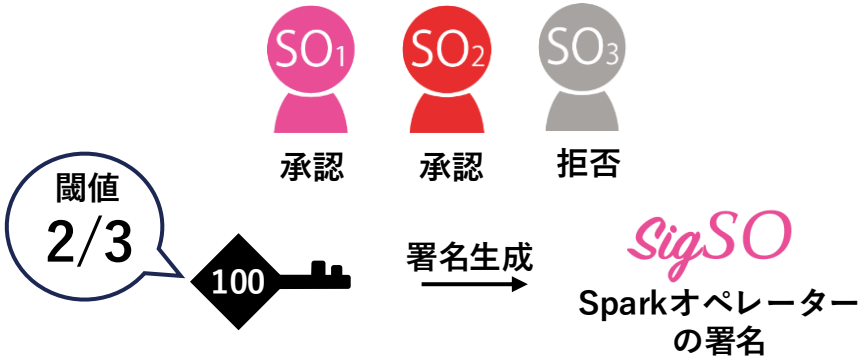
閾値署名を用いた運営の不正リスク抑制

閾値署名を用いることでSparkオペレーターの不正リスクを抑える

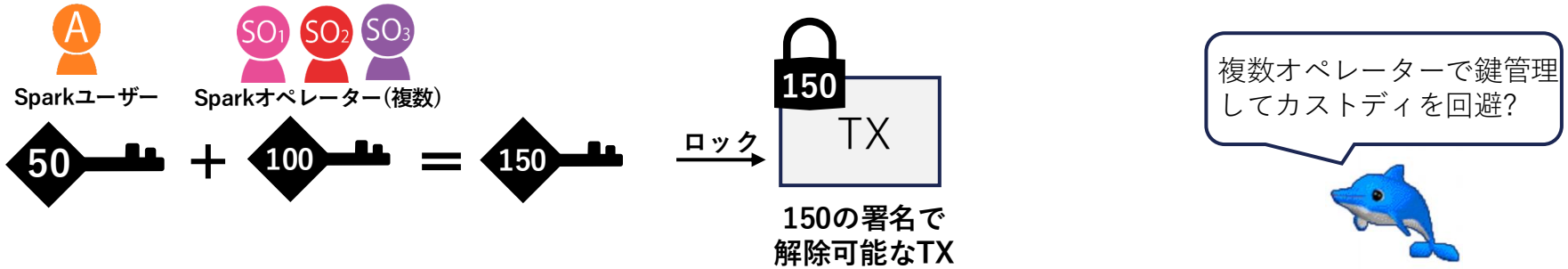
複数のSparkオペレーター同士で分散鍵生成で
合成秘密鍵を生成



閾値署名(FROST)で閾値を越えるSparkオペレーター
の承認がないと署名生成ができない



この仕組みと前ページのトランザクションのロック・アンロック・所有権移転を実現



Lightsparkとは？

- 企業向けSpark決済ネットワークを提供する企業
- Sparkプロトコルを設計しオープンプロトコル化
- 元Meta(リブラ)メンバーにより構成

採用状況: 続々と採用が進む

- Wallet of Satoshi
- Xverse
- LNBits
- その他Coinbaseなど金融機関での採用と記述あり

Transaction Type	Fee
L1 to Spark	On-chain fee paid by user
Spark to Spark	Free (small flat fee coming in 6-12 months)
Spark to Lightning	0.25% + routing fee
Lightning to Spark	0.15% (charged on routing nodes)
Exit to L1	L1 fee + SSP fee (formula: `sats_per_vbyte × (111 × 2 + tx_vbytes)`)

Lightsparkの手数料

懸念点

- Sparkオペレーターは現状**lightspark**と**Flashnet**の2者のみ
- **ウォレット向けSDK、APIには一方的退出手段が無い**
- LRC-20とかいう謎規格に準拠したSpark上で発行・取引可能なトークンBTNKを推す

まとめ・Ark/Sparkの課題

Ark / Sparkの課題

共通する課題

- ASP・SSPへのトラストが発生する
    - L2決済に承認が必要
    - ASP・SSPへのプライバシー問題
- Exitする際のブランチTXの高額オンチェーン手数料の支払いが必要
- 「プロトコル」と呼べるほどの定まった仕様がなくそれぞれが独自実装に近い
    - アトミックスワップを用いてオンチェーン・L2経由の送金ができるので問題はないが

Arkの課題

- タイムロック期間中にASPの資金はロックされるため、ASPは大量の資金が必要
- clArkでは事前署名が必要となるため、オンライン要件が発生
    - 署名の主体を委任するなどの方法を検討 (Delegated refreshes)

Sparkの課題

- Sparkオペレータが鍵を削除することをトラスト

そして、仕様は頻繁に変更がかかるため今日勉強した内容も半年後には全く別物な可能性あり

まとめ

ArkとSparkプロトコル まとめ

- LNノード運用の複雑さを解消するLNの補完的L2プロトコル。UTXOを預け入れるだけでオフチェーン決済可能
- サービス停止・不正時はExit TXによる一方的退出が可能で、運営者がユーザーの資金を奪うことはできない
- オフチェーンUTXOをツリー構造で管理し、末端のExit TXから退出できる設計
- Exit時のBranch TX手数料負担が大きく、仕様変更も頻繁で実用化に向けた課題が残る

Arkまとめ

- ASP（運営者）がコベナンツまたは事前署名でvTXOツリーを構成。コベナンツ不要なclArkも登場
- 主な送金手段はOOR送金（ラウンド外送金）で、ASPとの共同署名が必要なためASPへのトラストが発生する
- SecondのBark（決済特化）とArk LabsのArkade（Di-Fi向け）の2実装がOSSで公開済み

Sparkまとめ

- Schnorr鍵集約とステートチェーンで秘密鍵の差分を渡すことでオフチェーンの所有権移転を実現
- 複数Sparkオペレーター間のFROST閾値署名で単一オペレーターの不正リスクを分散
- LightsparkはオペレーターがlightsparkとFlashnetの2者のみで一方的退出手段もSDKに未実装と課題あり

参考文献

参考文献(Ark)

1. Somsen, R. (2021). Ark: A Scalable Second Layer for Bitcoin. Bitcoin Development Mailing List.
   https://gnusha.org/pi/bitcoindev/1300890009.1516890.1684742043892@eu1.myprofessionalmail.com/1-a.txt
2. Second. Ark Protocol Documentation.
   https://docs.second.tech/ark-protocol/intro/
3. Second. Second Blog.
   https://blog.second.tech/
4. Ark Protocol Project. Ark Protocol Documentation.
   https://ark-protocol.org/intro/index.html
5. Ark Labs. Ark Documentation.
   https://docs.arklabs.xyz/ark/
6. Arkpill. Ark Protocol Deep Dive. (Archived).
   https://web.archive.org/web/20240328181345/https://www.arkpill.me/deep-dive
7. Arkade. Arkade: Applications for the Ark Protocol.
   https://arkade.money/
8. Ceru, A. Awesome Ark Protocol. GitHub.
   https://github.com/aljazceru/awesome-ark-protocol
9. Somsen, R. (2023). Simplest Ark Explanation. GitHub Gist.
   https://gist.github.com/RubenSomsen/a394beb1dea9e47e981216768e007454
10. 加藤 規新. 「ビットコインで話題の新レイヤー2『Ark』はどのような技術なのか」,ビットコイン研究所.
    https://bitcoin-research.jp/how-ark-works-and-bitcoin-seoul/
11. 加藤 規新. 「Arkがライトニングの流動性問題を解決したと言われる理由は？」,ビットコイン研究所.
    https://bitcoin-research.jp/ark-vs-lightning-liquidity/

参考文献(Spark)

1. Spark. TL;DR: Spark Overview.
   https://docs.spark.money/learn/tldr
2. Lightspark. Spark Protocol Source Code. GitHub.
   https://github.com/buildonspark/spark
3. Spark. Spark: Bitcoin Layer-2 Payments Infrastructure.
   https://www.spark.money/
4. Spark. Spark Network Status.
   https://www.spark.money/status
5. Lightspark. Lightspark Official Website.
   https://www.lightspark.com/
6. 加藤 規新. 「Lightsparkが発明した新しいレイヤー2：Spark」,ビットコイン研究所.
   https://bitcoin-research.jp/spark-statechains-evolution/
7. 加藤 規新. 「SparkとXverseの提携で浮かび上がる、新種のビットコインエコシステム」,ビットコイン研究所.
   https://bitcoin-research.jp/xverse-spark/

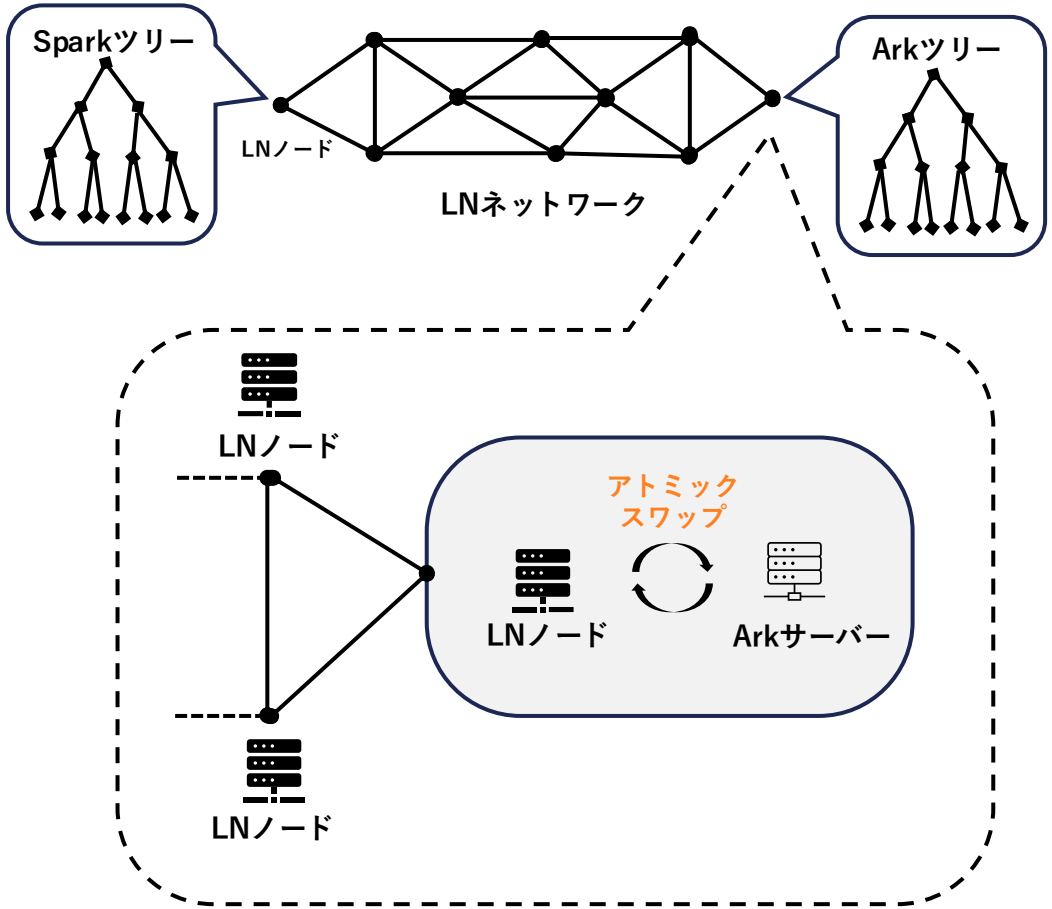
参考文献(その他)

1. Bitcoin Optech. Covenants. https://bitcoinops.org/en/topics/covenants/
2. Bitcoin Optech. OP_CHECKTEMPLATEVERIFY. https://bitcoinops.org/en/topics/op_checktemplateverify/
3. Rubin, J. (2020). BIP-119: OP_CHECKTEMPLATEVERIFY. Bitcoin Improvement Proposal. https://github.com/bitcoin/bips/blob/master/bip-0119.mediawiki
4. Bitcoin Core Developers. BIP-347: OP_CAT. Bitcoin Improvement Proposal. https://github.com/bitcoin/bips/blob/master/bip-0347.mediawiki
5. Bitcoin Core Developers. BIP-348: OP_CHECKSIGFROMSTACK. Bitcoin Improvement Proposal. https://github.com/bitcoin/bips/blob/master/bip-0348.md
6. Bitcoin Core Developers. BIP-345: OP_VAULT. Bitcoin Improvement Proposal. https://github.com/bitcoin/bips/blob/master/bip-0345.mediawiki
7. Bitcoin Covenants. Bitcoin Covenants Explained. https://bitcoincovenants.com/
8. Bitcoin Magazine. Bitcoin Layer-2 Statechains. https://bitcoinmagazine.com/technical/bitcoin-layer-2-statechains
9. 加藤 規新. 「LNを補完するStatechains、サイドチェーンの伏兵Drivechain」, ビットコイン研究所. https://bitcoin-research.jp/statechains-to-complement-lns-sidechain-foreshadowing-drivechain/

Appendix.

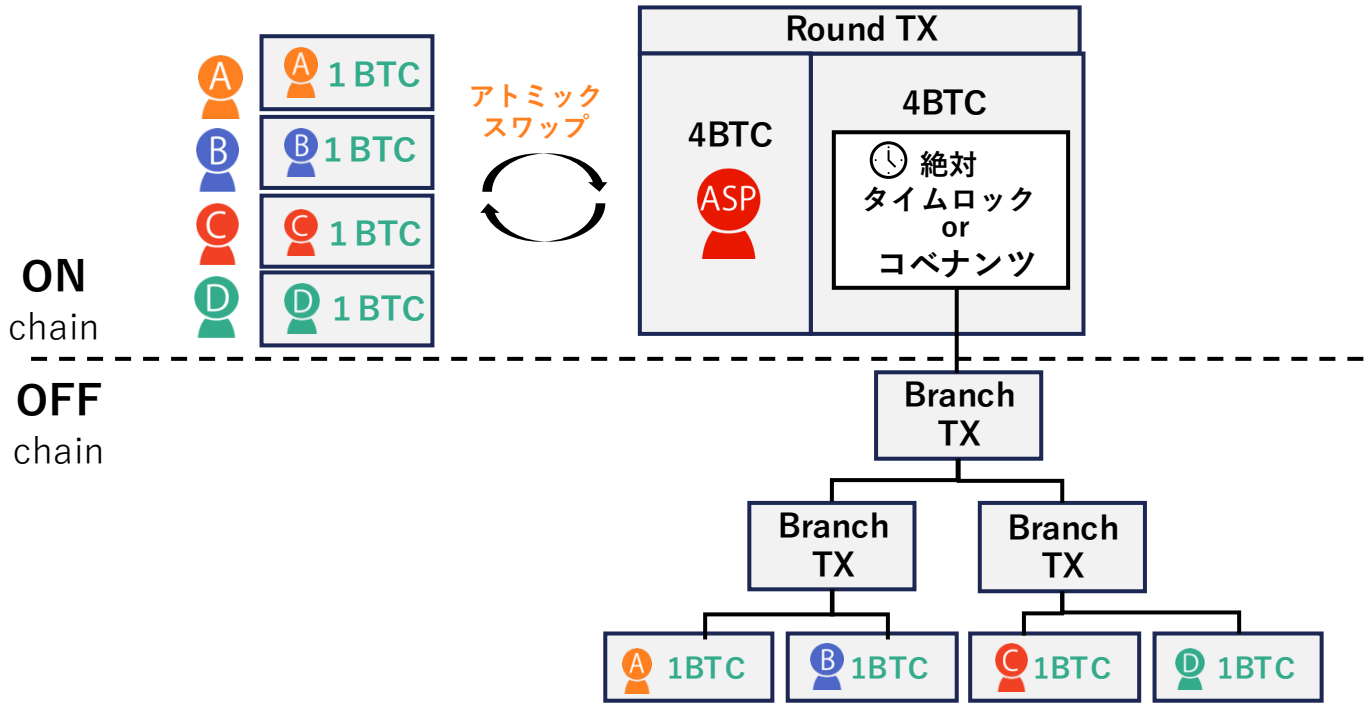
(Appendix.1) Ark・Spark外への送金

Ark・Sparkサービスへの預け入れ・出金ではサブマリンスワップなどのアトミックスワップを利用
Ark・Spark外のLNユーザーに送金する場合はサービスプロバイダーによるチャネル・流動性管理に依存



(Appendix.2) 実際のArkラウンドTXへの入金方法

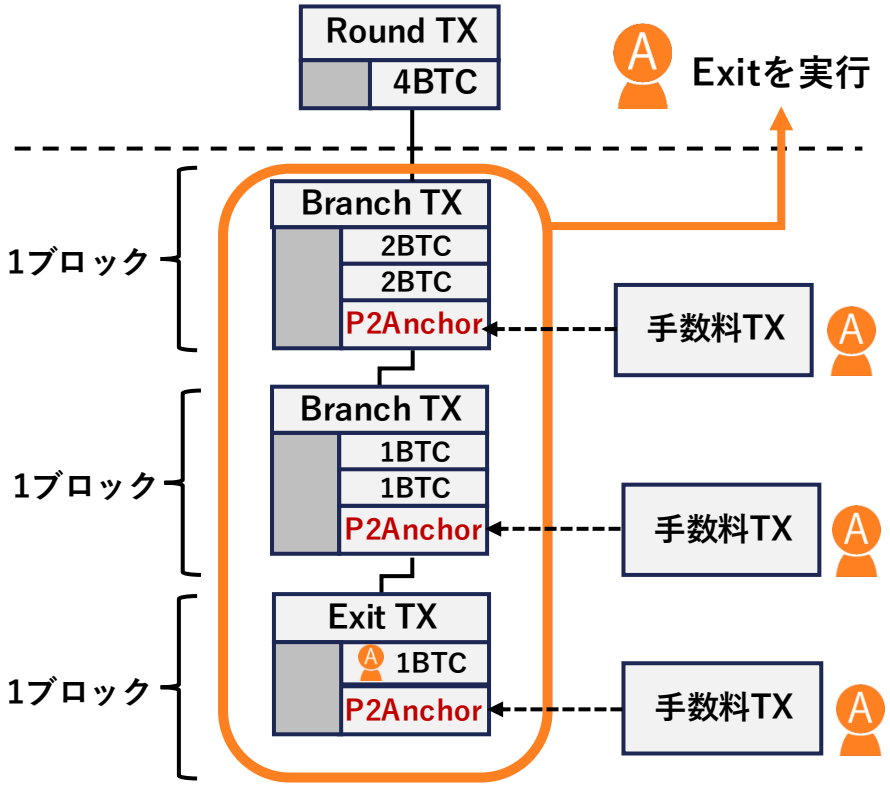
- ユーザーはArkへの入金時、Ark TXツリー上のvTXOとUTXOをアトミックスワップし入金
- ラウンドTXのInputはASPが支払う
  → 多くの資金が必要になる



(Appendix.3) Arkトランザクションとトランザクション手数料の問題

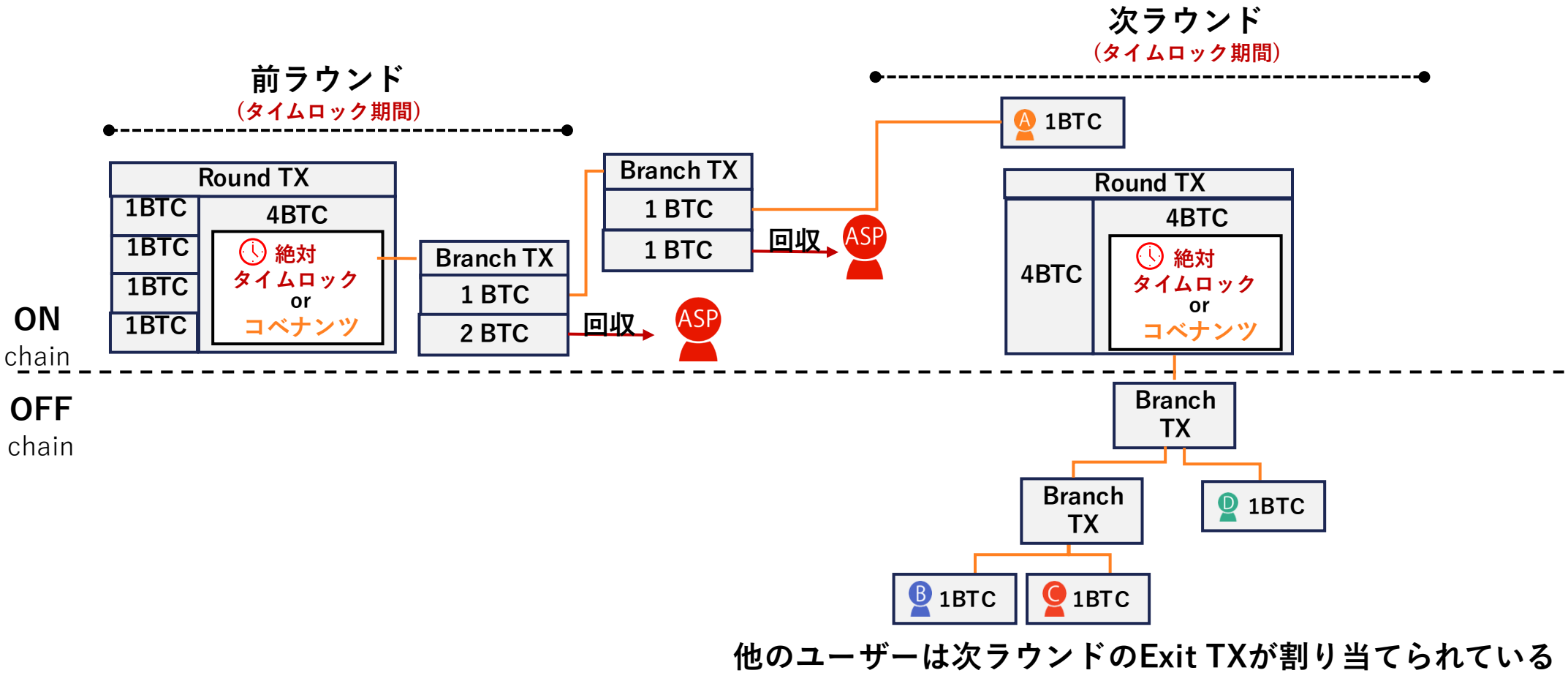
Branch TXを上から一個づつ公開が必要

- Branch TXの手数料はExitするユーザーが払う必要あり
- 各Branch TXにPay to Anchorアウトプットがある
    - ユーザーが手数料補填をする必要あり
- パッケージリレー等のアップデートが必要になる



(Appendix.4-1) Exit後のユーザー&ASPよる資金回収方法

1. ユーザーAがExit TXとBranch TXをオンチェーン公開し一方的退出(絶対＋相対タイムロック待ち)
2. ASPは次ラウンドのRound TXを公開し、TXツリーを作成し、他のユーザー用vTXO割り当て
3. ASPが絶対タイムロック後、Branch TXのアウトプット回収
4. 絶対＋相対タイムロック経過後、AがExit TX回収



他のユーザーは次ラウンドのExit TXが割り当てられている

（Appendix.4-2）より詳細な図