



# Corda Enterprise アーカイブサービスの紹介

---

2021.11.25

**TIS株式会社**

産業公共事業本部

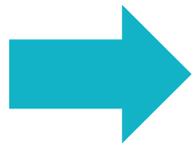
産業公共事業推進部 Blockchain推進室

- 背景
- 概要
- メリット
- アーカイブできるパターンの比較
- 利用手順
- 考察

# 背景

---

- ブロックチェーンの課題としてトランザクション数の増加によるデータ容量の膨大化があります。
- それに伴う、台帳への書き込み速度の低下やデータの取得速度の低下が引き起ります。



アーカイブサービスはこのような問題を改善、緩和できる可能性があります。

# 概要

---

- アーカイブサービスはCE4.7で追加された機能です。
- アーカイブサービスはすでに終了したトランザクションやattachmentのアーカイブを作成し、それを必要な時に復元することができます。
- アーカイブはトランザクションチェーン単位で実施します。
- データはzipファイル等にアーカイブできます。

# メリット

---

アーカイブすることにより、データベースの容量を確保することができます。

→ノードのデータベースへの負荷を軽減し、データの取得や書き込み速度の向上が期待できます。

```

$ java -jar corda-tools-archive-service-1.0.jar -t list-items
? Starting
? Reading configuration
? Loading filters
? Executing filters
? Done
Number of archivable transactions: 176
Number of archivable attachments: 0
Approximate size of archivable transactions: 989KB
Approximate size of archivable attachments: 0B
    
```

アーカイブする  
トランザクション数

アーカイブする  
バイト数

# アーカイブできるパターンの比較

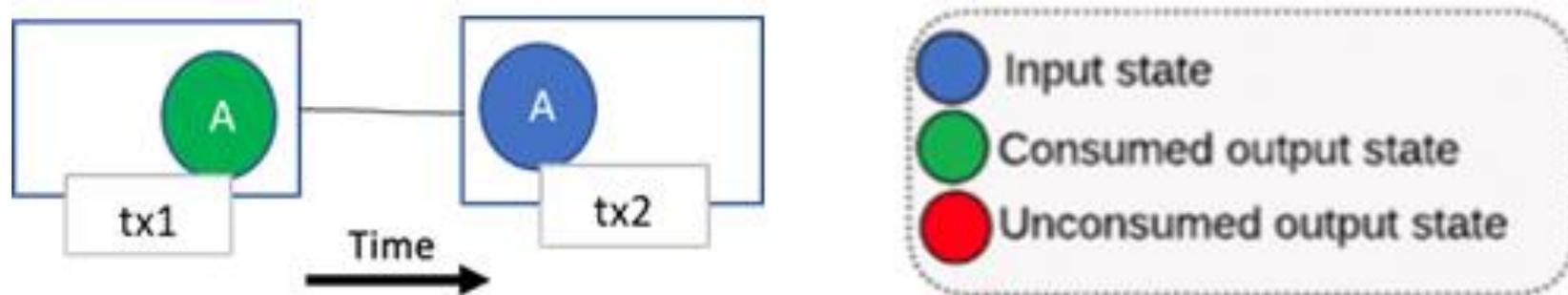
---

分岐や合流、他ノードを含むトランザクションにおいても以下の2つの条件に当てはまればアーカイブが可能になります。

- 対象のトランザクションチェーン内に未消費のStateがないこと。
- チェーンの終端、全ての関係者であること。

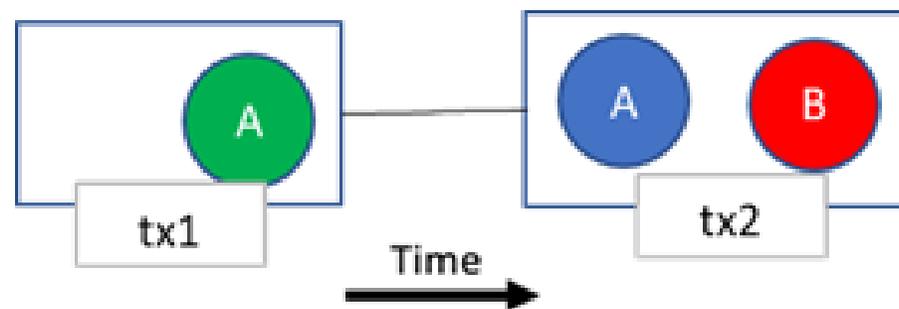
例を示して解説します。

円に書かれたアルファベットは所有者(ノード)を示します。tx1でOutput Stateを作成し、tx2でそのStateを消費して終了するトランザクションチェーンです。このトランザクションチェーンは未消費のStateがないため**アーカイブ可能**になります。



tx1でOutput Stateを作成し、tx2でそのStateを消費してOutput Stateを作成します。

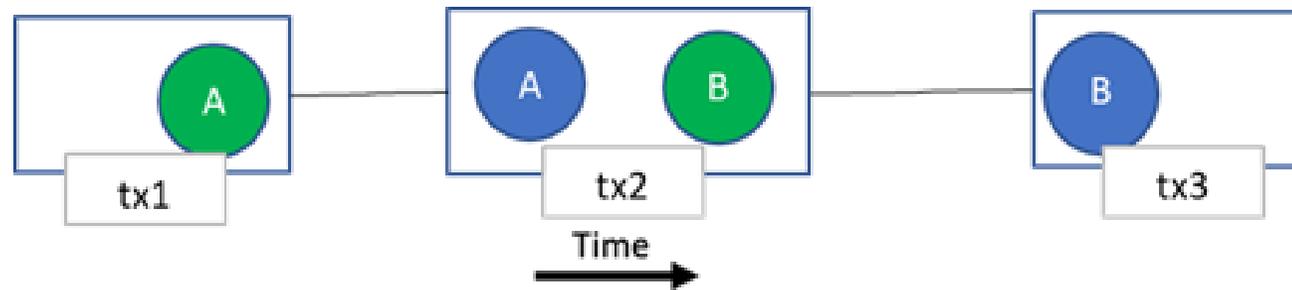
tx2で作成されたOutput Stateがまだ消費されていないためこのトランザクションチェーンは**アーカイブ不可**になります。



このトランザクションチェーンでは**Bのみアーカイブ可能**になります。

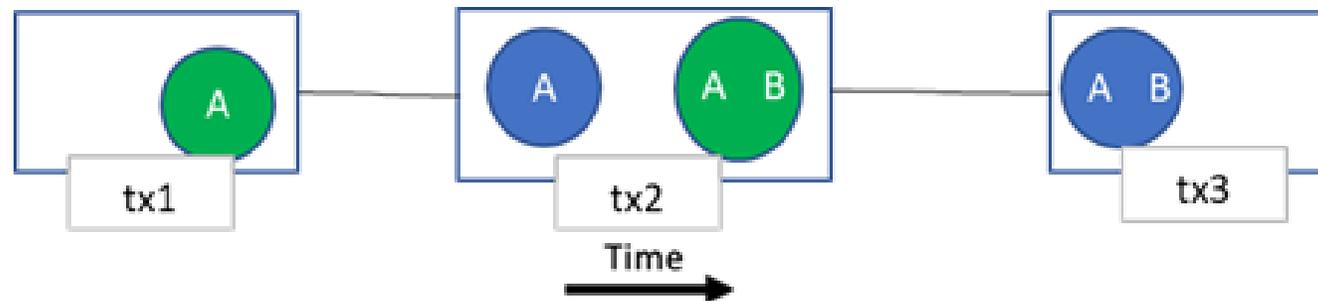
トランザクションチェーンに未消費のStateがありませんが、A側ではアーカイブできません。

Aはチェーンの終端のtx3の関係者ではないためアーカイブすることはできません。



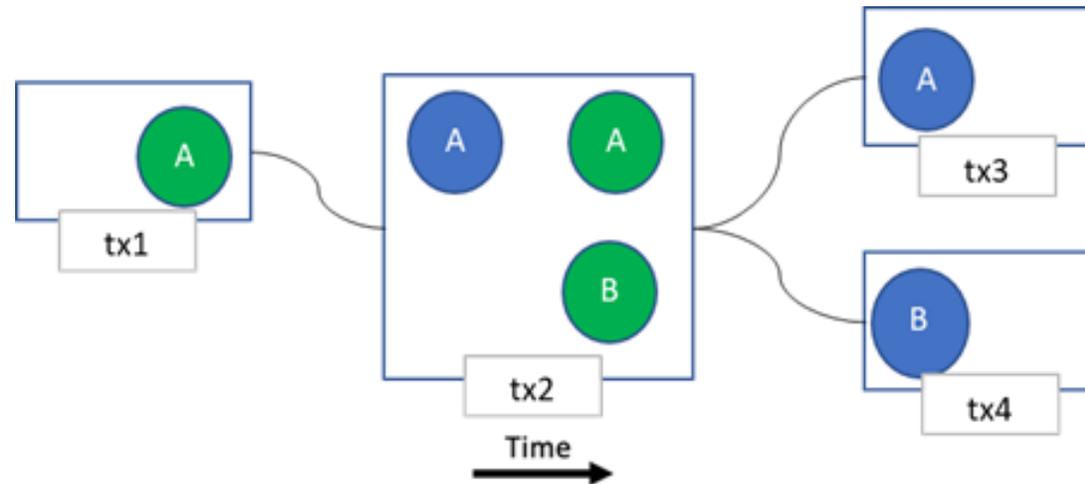
このトランザクションチェーンではA、Bともにアーカイブ可能になります。

A、Bともにチェーンの終端のtx3の関係者であるためアーカイブすることができます。

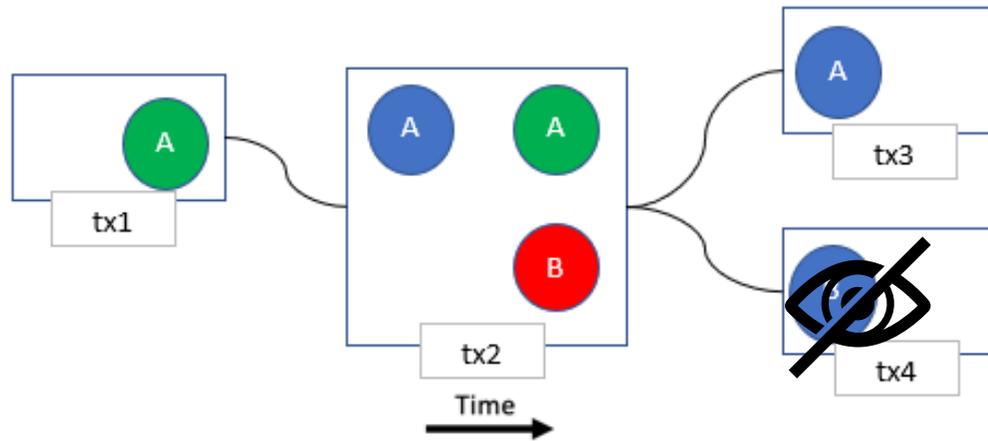


このトランザクションチェーンではA、Bともにアーカイブ不可になります。

このチェーンの終端はtx3とtx4になり、AとBはそれぞれ一方にしか関係していないためどちらもアーカイブすることができません。

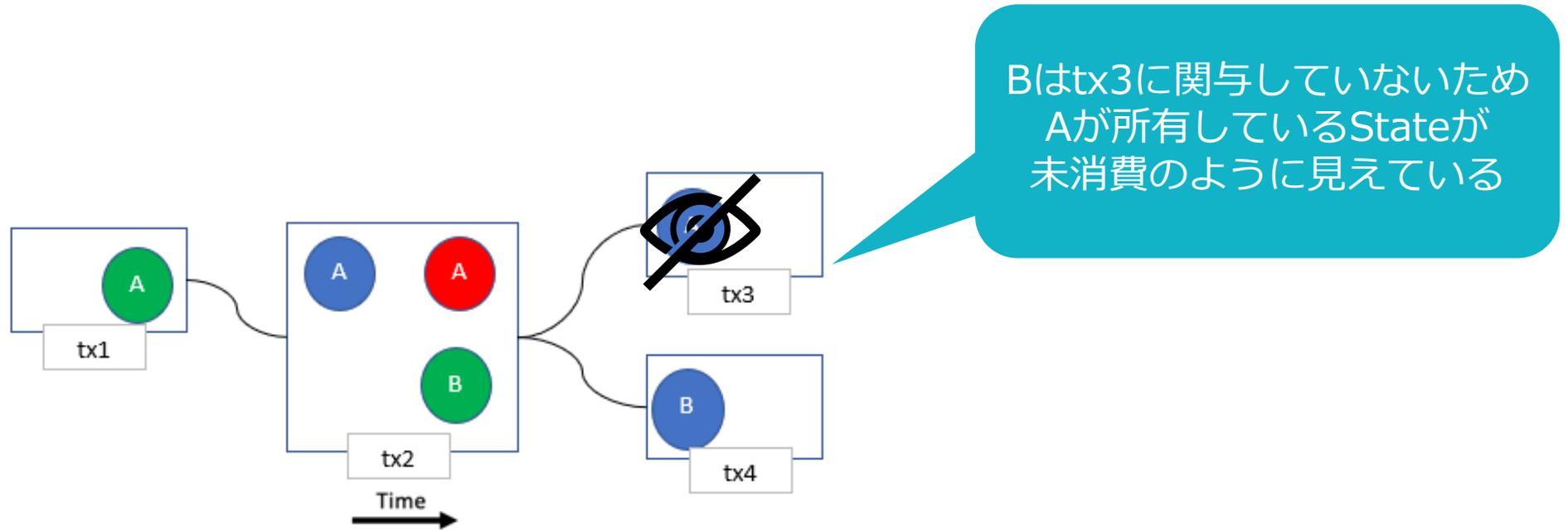


## ・ NodeAの見え方



Aはtx4に関与していないため  
Bが所有しているStateが  
未消費のように見えている

## ・ NodeBの見え方



# 利用手順

---

利用手順は2つに分けて解説します。

- 事前準備
- 実施手順

- 1 ノードのcordapps直下に以下のjarを配置します。
  - archive-service-1.0.jar
  - ledger-graph-1.2.jar
- 2 設定ファイルを配置します。
  - archive-service-1.0.conf
- 3 アーカイブサービスコマンドラインツールを配置します。
  - corda-tools-archive-service-1.0.jar

→APIを利用する場合は必要ありません。

## ※補足

アーカイブサービスにはアーカイブ中に障害が起きてもバックアップスキーマを用意することでデータを復元できる機能があります。

必須ではありませんが、安定運用を考えた場合、導入することをお勧めします。

アーカイブ実施時はバックアップスキーマにアーカイブ対象の全データを書き出し、一時的にデータ容量が増えるため注意が必要です。

## アーカイブ実施手順

1. 進行中のjobの確認
2. アーカイブできるトランザクションの数と情報を表示
3. アーカイブjobの作成
4. トランザクションをアーカイブする
5. vaultを削除する  
(アーカイブをキャンセルする)
6. バックアップスキーマの削除

## アーカイブデータの復元

アーカイブ実施と同じコマンドラインツールでオプションを指定して実施できます。

復元時、`node_transactions`や`vault_states`の`timestamp`は復元時間が挿入されるので注意が必要です。

# 考察

---

アーカイブ可否がある以上、アーカイブサービスに適した Cordapps を作成する必要があります。

- 短いトランザクションチェーンで完結する、もしくはトランザクションチェーンを定期的に切断し State を再発行するような機能を有していること。
- ビジネスロジック上使用されなくなった State についても償還できる機能や処理を有していること。

- Transactionが生成した複数のOutput Stateを複数のTransactionのInput Stateとして使用されない仕組みになっていること。

まとめとしてアーカイブサービスを利用するには「トランザクションチェーンが複雑に分岐せず、様々な人が交互に関係しないもの」や「再発行機能を実装したもの」が適しています。

THANK YOU

ITで、社会の願い叶えよう。



**TIS INTEC**  
Group