



# Cordaの勘所 ～トレーサビリティ編～

---

2021.05.14

**TIS株式会社**  
産業公共事業本部  
Blockchain推進室

## TIS株式会社 Blockchain推進室 エキスパート 中地 英樹（なかし ひでき）

1996年大学院前期課程修了（情報工学分野専攻）。  
同年、TIS株式会社に入社。

入社当時に流行していたオブジェクト指向開発ブームを背景にオブジェクト指向データベースや分散オブジェクト技術を利用したシステム開発などを多数経験。

2018年にTISにBlockchain推進室が新設されたことを機に部署異動。  
現在はエンタープライズ向けブロックチェーン技術に注目しており、  
実際にお客様と試行錯誤しながら各種PoC（実証実験）を推進中。

## 0. 前置き

### 1. データ全共有

### 2. データモデリング

### 3. データ検索

- 今回のお話の内容は、  
「**製造業におけるトレーサビリティ**」  
を**Corda**で実現するときの**Tips**です。
- **トレーサビリティ**と言っても色々あります。
  - ① **製品の工場出荷**→(物流)→卸→小売→消費者まで
  - ② **原材料**→部品→(組立)→半製品→製品まで
- ①は受発注などの**商流(カネの流れ)**もありますが、  
メインは**物流(モノの流れ)**です。(チェーントレーサビリティ)
- ②のメインは**製造工程の流れ**を**トレース**することです。  
(内部トレーサビリティ)

Agenda

1. データ全共有
2. データモデリング
3. データ検索

## ■ Cordaで実現する際の悩みどころ

1. 商流はプライバシーを保ちたい（個社間共有）が、物流は川上から川下まで把握させ（全共有）たい。
2. UTXO型でモノの流れをどう表現する？  
単一のモノの流れなら製品IDなどで追えるが、製造工程のようにモノ以外（梱包、検査など）が登場する場合は？
3. UTXO型で表現されたデータを、どうトレースする？

# 1. データ全共有

---

## ■ トレーサビリティのためにデータを全共有する方法は？

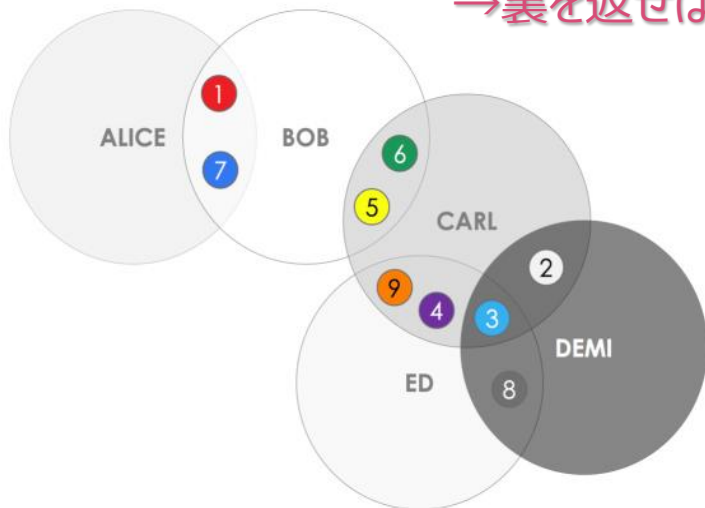
- ① 全ノードをStateの共有先に指定する
- ② NotaryをValidating-Notaryにする
- ③ Observerノードを導入する

## ①全ノードをStateの共有先に指定する

### ■ Cordaの特徴：プライバシーの確保

一般的なブロックチェーン製品は全てのNodeが同一の情報を持つが、Cordaの場合は**トランザクションごとにデータの検証と共有先を動的に指定**できる。

→裏を返せば、共有先を全ノードにすれば全共有できる



ALICE = { 1 7 }

BOB = { 1 5 6 7 }

CARL = { 2 3 4 5 6 9 }

DEMI = { 2 3 8 }

ED = { 3 4 8 9 }



## ①全ノードをStateの共有先に指定する

### ■全ノードとP2P通信を行うことになるので遅い

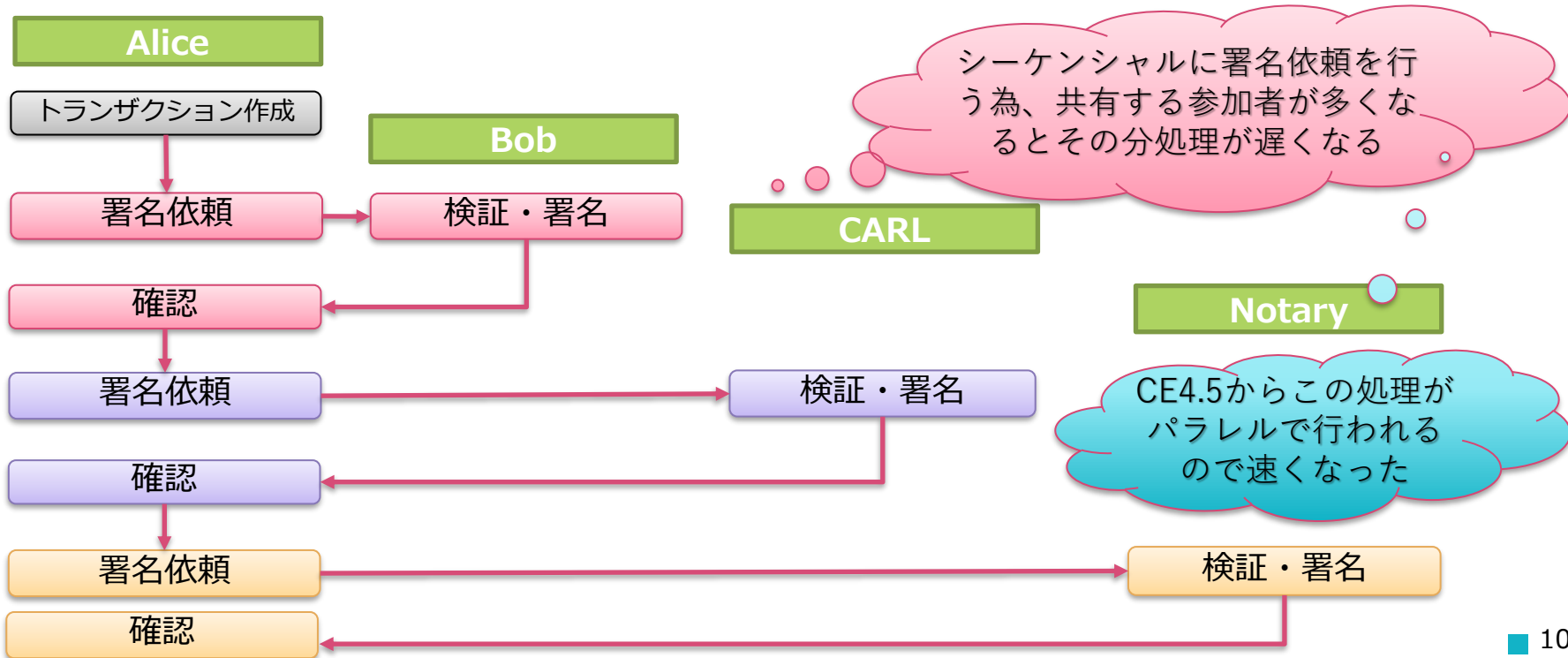
- ★CE4.4までは全ノードをシーケンシャルにP2P通信していたが、  
**CE4.5からは平行にP2P通信できるようになって、**  
速くはなったが・・・  
(Corda Enterprise版でないとはなりません)

### ■どこかのノードが1つダウンしただけで、 トランザクションが待ち状態になる

- ★ダウンしたノードが再起動するまでリトライを繰り返し、  
トランザクションは完了しない（エラーで破棄にもならない）

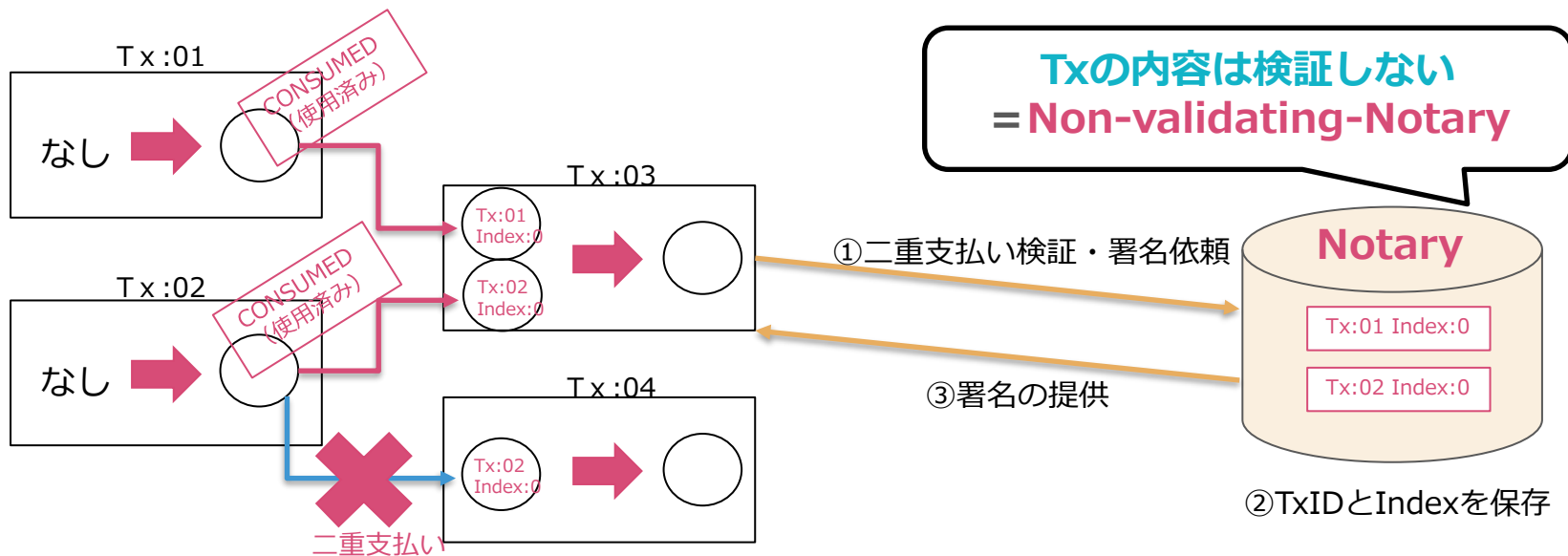
・ブロックチェーンって、全員でデータ共有するのが基本だけどCordaはできる？

➡ できるが、向かない。全員の検証・署名が必要になり、処理が遅くなる。



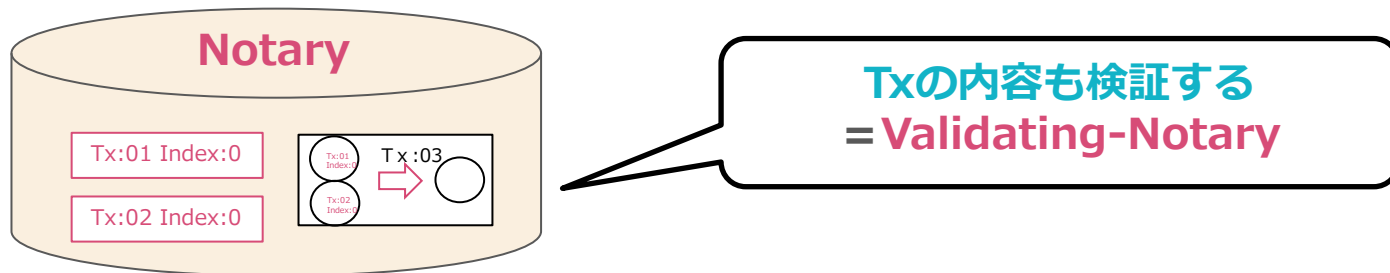
## ②NotaryをValidating-Notaryにする

NotaryとはOutputを使用済みかどうか（二重支払い）をチェックし、一意性コンセンサスを実現する



## ②NotaryをValidating-Notaryにする

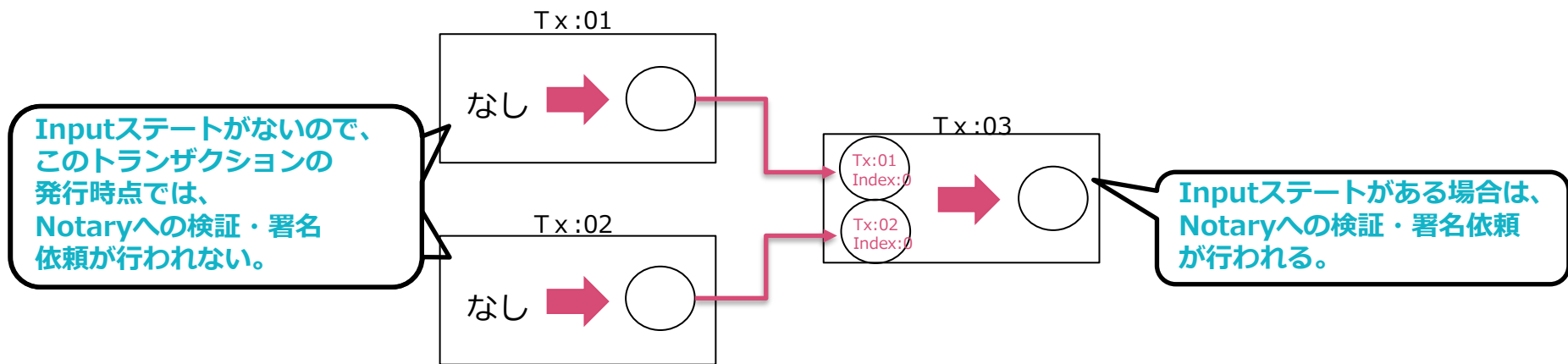
★通常のNotaryは、Non-validating-Notary。  
二重支払いのチェックはするが、トランザクションの検証はしないため、DBには使用済みのOutputステートのトランザクションID (=ハッシュ値) とインデックスしか情報をもたないが、**Validating-Notaryにすると、トランザクションの検証も行うためトランザクションデータもNotaryのDBにもつ。**



## ②NotaryをValidating-Notaryにする

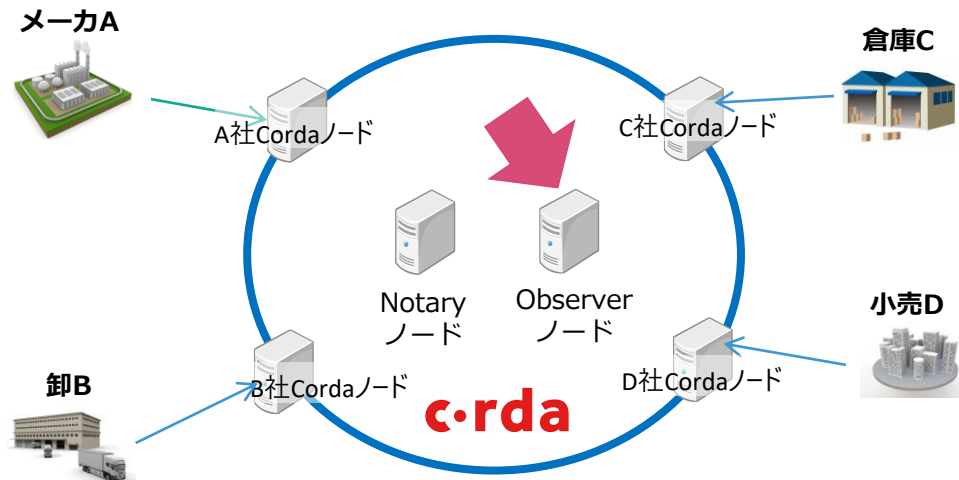
### ■ Inputステートのないトランザクションの場合はNG

★二重支払いチェックが不要なためNotaryへの通信は発生しない  
(=トランザクションの検証も保存もされない)



### ③Observerノードを導入する

★Observerノードは、本来、規制当局などが運用することを想定した監査ノードのこと。個社や個社間で発行したトランザクションを第三者機関が保持する目的で導入する。このObserverノードのトランザクションデータを参照すれば、物流／商流の川上から川下まで、全データをトレース可能。



## ③Observerノードを導入する

### ■ 現実解はこれ。

ただし、非中央集権性を求められる場合は要注意

★実際には全てのトランザクションをObserverノードに  
持たせるのではなく、皆でトレースしたい情報のみに絞るべき。  
(例) 物流データ (モノの流れ) はObserverにも保持  
商流データ (カネの流れ) は当事者間のみで保持  
など、使い分けすることが勘所。

## 2. データモデリング

---

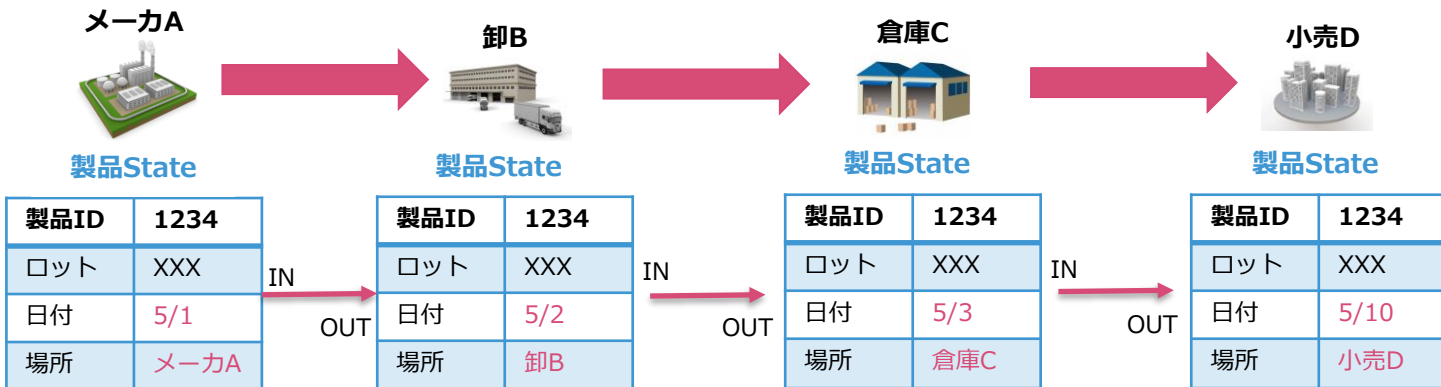


### ■ トレーサビリティのために何をステートと定義する？

- ① モノ（製品、部品、原材料）をステートとする
- ② 工程（加工、検査、梱包、出荷）をステートとする

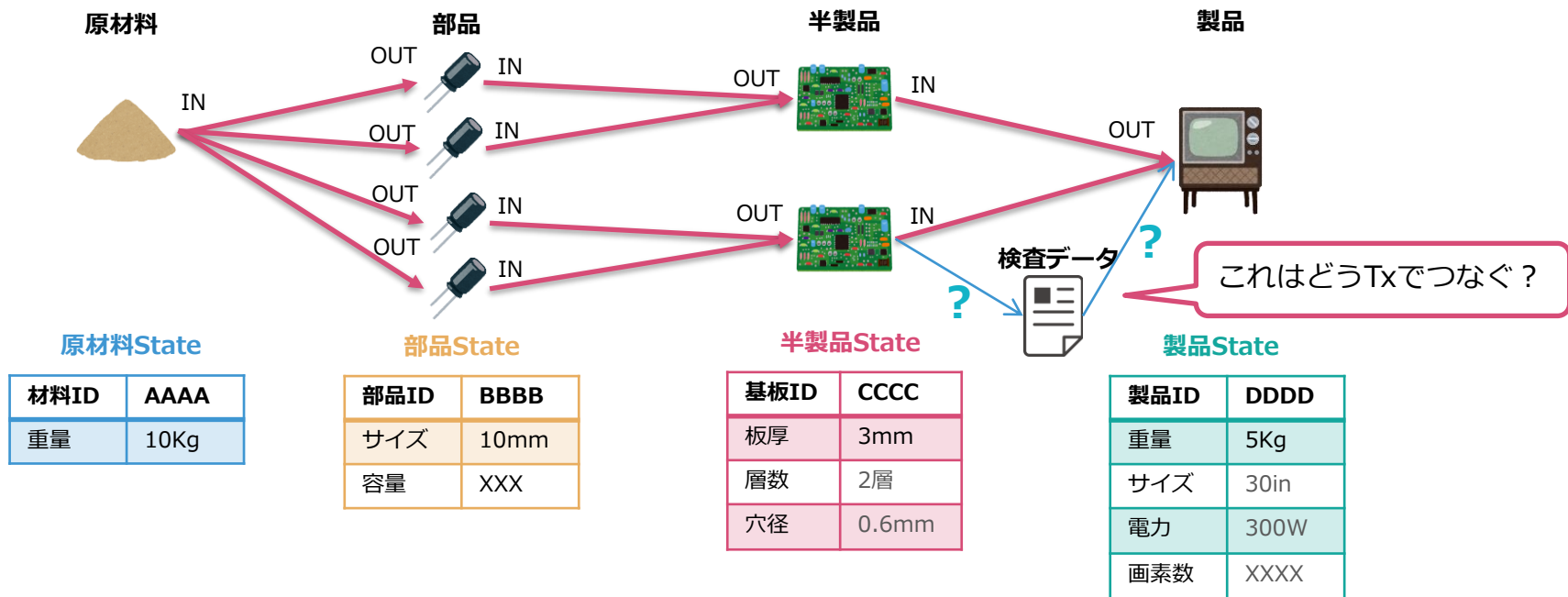
## ① モノ（製品、部品、原材料）をステートとする

- 流れていくモノだけが対象の場合は、モノとその付加情報（日付、場所など）をステートと定義し、TxのInput/Outputで繋げばよい。



★Stateの種別が同一だと実装も検索もラク！

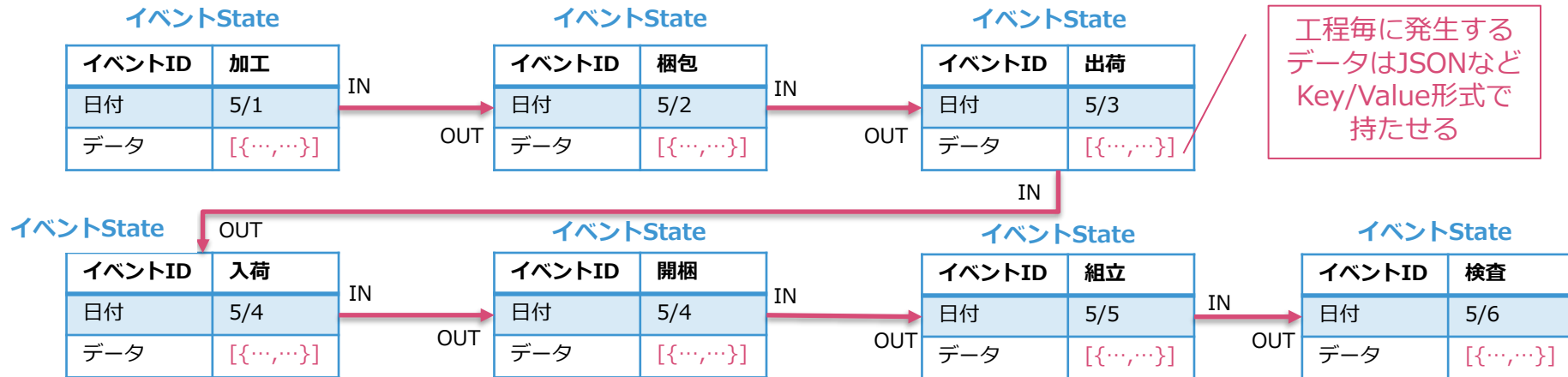
### ① モノ（製品、部品、原材料）をステートとする



★Stateの種別が異なるとVaultQueryでのステート検索が困難  
 ★梱包や検査などのデータはTxのIN/OUTではモノと繋ぎにくい

### ② 工程（加工、検査、梱包、出荷）をステートとする

- モノ以外のイベント（梱包、検査）がある場合は、モノをステートとせず、イベントをステートと定義し、TxのInput/Outputで繋げばうまく表現できる。



★ Stateの種別が同じなので実装／検索がラク

★ TxのIN/OUTを辿れば、工程もモノの流れもトレースできる

### 3. データ検索

---

## ■トレーサビリティのためにデータを検索する方法は？

① VaultQueryなどでステートを検索（一般的）

② serviceHubなどを使いトランザクションを検索

- `validatedTransactions.track().snapshot` 全件取得
- `validatedTransactions.getTransaction(id)` id指定で取得

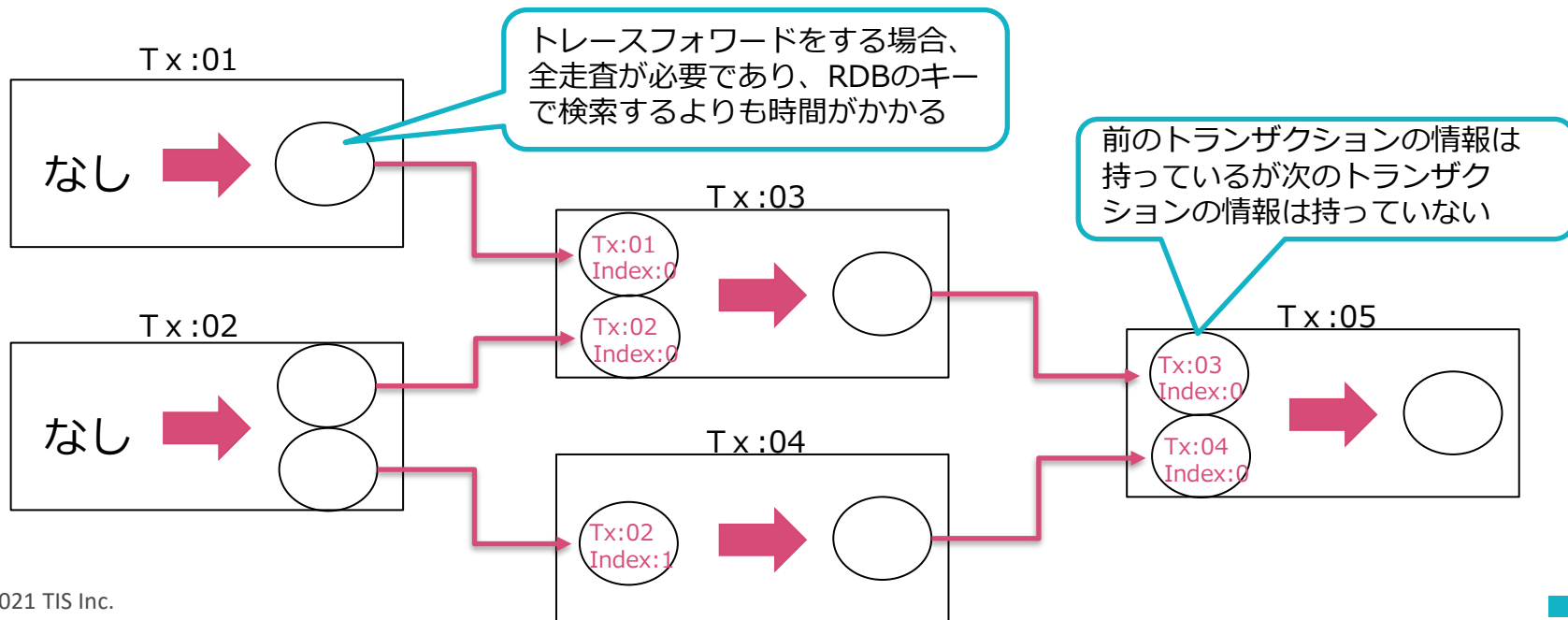
③ CordaのDBはRDBなのでSQLで検索する（番外編）

★ここでは②のトランザクション検索がテーマです。

## ② serviceHubなどを使いトランザクションを検索

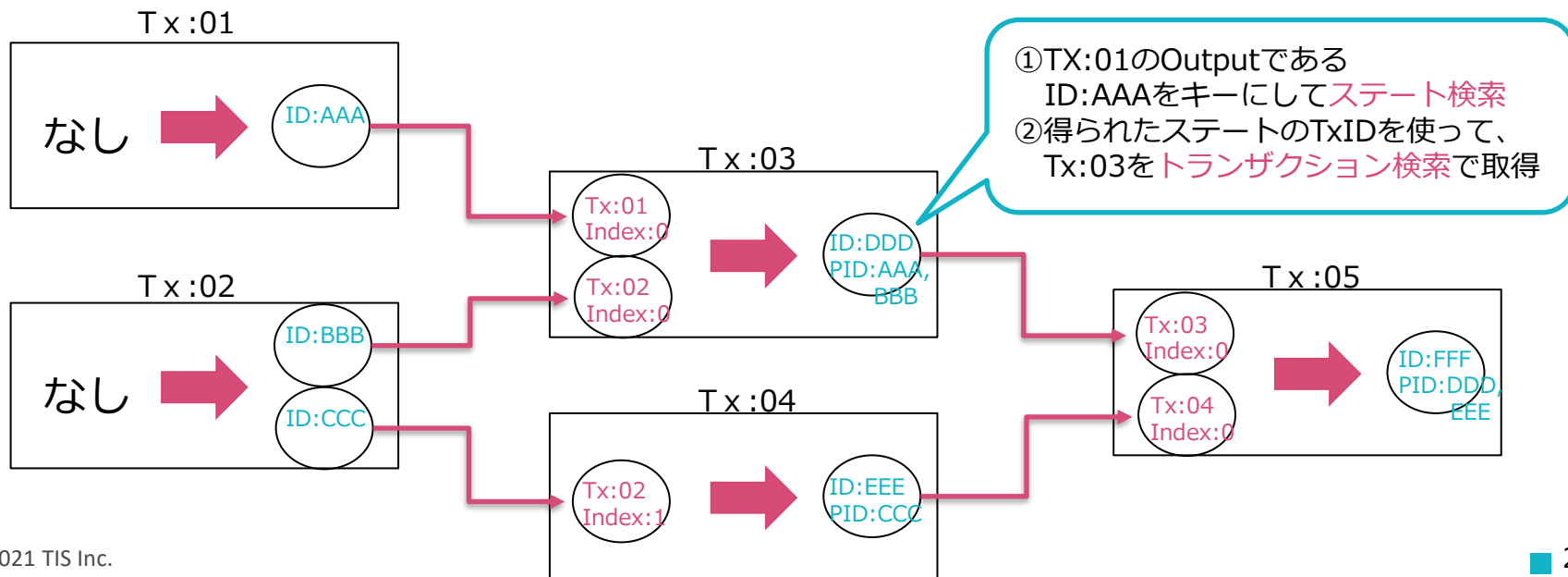
### ■ 直接TxのIN/OUTを辿るには便利

★ただしトレースバックは得意だが、トレースフォワードは苦手



## ② serviceHubなどを使いトランザクションを検索

★トレースフォワードのためにVaultQueryでの**ステート検索**を併用することが**現実解**。そのために、ステートを検索しやすい**キー**を適切に設定しておくことが**勘所**。





## ■ トレーサビリティのためにデータを全共有する方法は？

Observerノードの導入がおすすめ。  
ただし、全共有するデータは絞みましょう。

## ■ トレーサビリティのために何をステートと定義する？

モノではなく、製造工程（イベント）をステートと定義し、  
イベント間をUTXOのチェーンでつなげましょう。

## ■ トレーサビリティのためにデータを検索する方法は？

UTXOのチェーンはトランザクション検索で辿れます。  
ステート検索も併用できるようにキーを適切に設定しましょう。

# ITで、社会の願い叶えよう。



**TIS INTEC**  
Group

<本資料の取り扱いに関して>

本資料は、著作権法及び不正競争防止法上の保護を受けております。資料の一部あるいは全部について、TIS株式会社から許諾を得ずに、複写、複製、転記、転載、改変、ノウハウの使用、営業秘密の開示等を行うことは禁じられております。本文記載の社名・製品名・ロゴは各社の商標または登録商標です。