

# トレーサビリティ機能を備えた サプライチェーンマネジメントアプリの紹介

2021/08/25

# 自己紹介



実家で飼っている猫。今年で**20歳**です。

## 中村 成 (なかむら たいら)

SBI R3 Japan マーケティング担当兼

SBIホールディングス ブロックチェーン推進室

(所属部署は用件に応じて転々としています。)



- 現在の業務は、HP運用やCordaニュースの配信、SCMアプリ管理、SBIグループ内ブロックチェーン活用会議の事務局等を担当。
- 趣味はスポーツ観戦(野球、サッカー、卓球)。毎日、大谷くんの活躍を楽しみにしています。

# 目次

## 1. サプライチェーンマネジメント(SCM)アプリの紹介(ビジネス編)

- 本アプリについて
- デモ実演

## 2. SCMアプリの技術紹介

- アプリの全体像
- データの流れの確認
- Cordapp実装の一部紹介

対象はCordaに初めて触れる方を対象にしています。

「サプライチェーンマネジメント」は以後「**SCM**」と略して表記します。

# **1. SCMアプリの紹介(ビジネス編)**

# 本アプリについて

---

- ✓ BC利用イメージを掴んでいただくためのサンプルアプリ
  - ✓ 鉱物、食品、素材、部品などの汎用的なSCMを想定
  - ✓ 「モノ」の生産者をバイヤー側からトレースできる。
  - ✓ 追加開発により実証実験の移行が容易
- (現時点は企業間データ連携部分のみのシンプルな実装)

# ブロックチェーンによるトレーサビリティの強み

製品に含まれる素材の産地や流通ルートを一瞬かつ正確に把握できる。

**Before**

情報が分断されているため、1社1社たどる必要がある。

素材会社

部品会社

自動車会社



**After**

川上から川下までトレース情報が最初から可視化されている。

素材工場

部品工場

自動車工場

共通アプリケーション



# 本アプリの機能

製造業のSCMを想定した3つの機能がある。

**製造**

保有している素材や部品を消費して新しい製品を製造

**購入**

他の企業が保有している製品を購入。

**トレースバック**

製品を作るために使用した部品や素材の情報を取得。

# SCMアプリ デモシナリオ

自動車工場で車を製造するための企業間取引の流れのデモを行う。

## PartyA.素材工場

ゴム  
ワイヤー



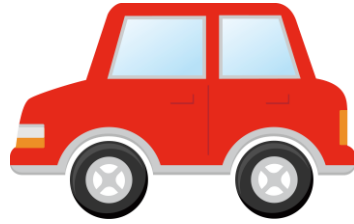
## PartyB.部品工場

タイヤ



## PartyC.自動車工場

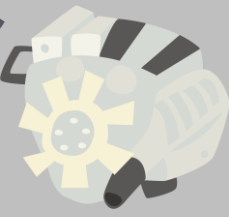
車



オイル  
鉄  
化学薬品



エンジン



※時間の都合でエンジンまで事前に作成しています

1. A(素材工場)はゴム、ワイヤー、オイル、鉄、化学薬品を生産し、B(部品工場)が購入。
2. B(部品工場)は素材をベースにタイヤ、エンジンを生産し、C(自動車工場)が購入。
3. C(自動車工場)はタイヤ、エンジンを元に車を製造。
4. C(自動車工場)は車の部品・素材に対する製造情報を自社のDBを見て確認。



# デモ実演

# デモシナリオ

デモはシナリオの詳細は以下の通り。製造に必要な素材は、事前にリストとして作成しておく。

1				2			
Party A (素材工場)				Party B(部品工場)			
Input		Output		Input		Output	
-	-	ゴム	40	→ ゴム	40	タイヤ	4
		ワイヤー	80	ワイヤー	80		
		オイル	10	オイル	10	エンジン	1
		鉄	20	鉄	20		
		化学薬品	30	化学薬品	30		
3							
Party C(自動車工場)							
Input		Output					
・	タイヤ	4 車	1				
・	エンジン	1					

## 2. SCMアプリの技術紹介

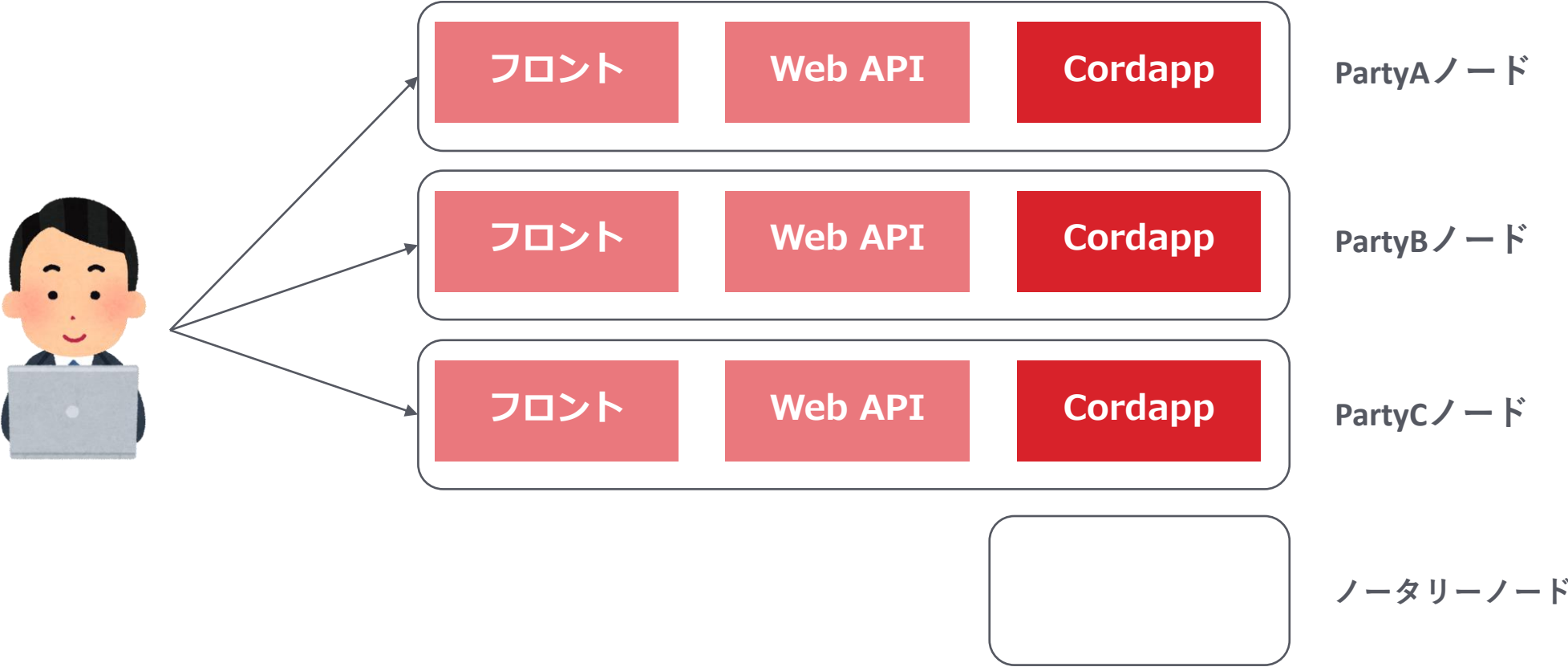
---

# アプリの全体像

---

# 本SCMアプリに含まれるもの

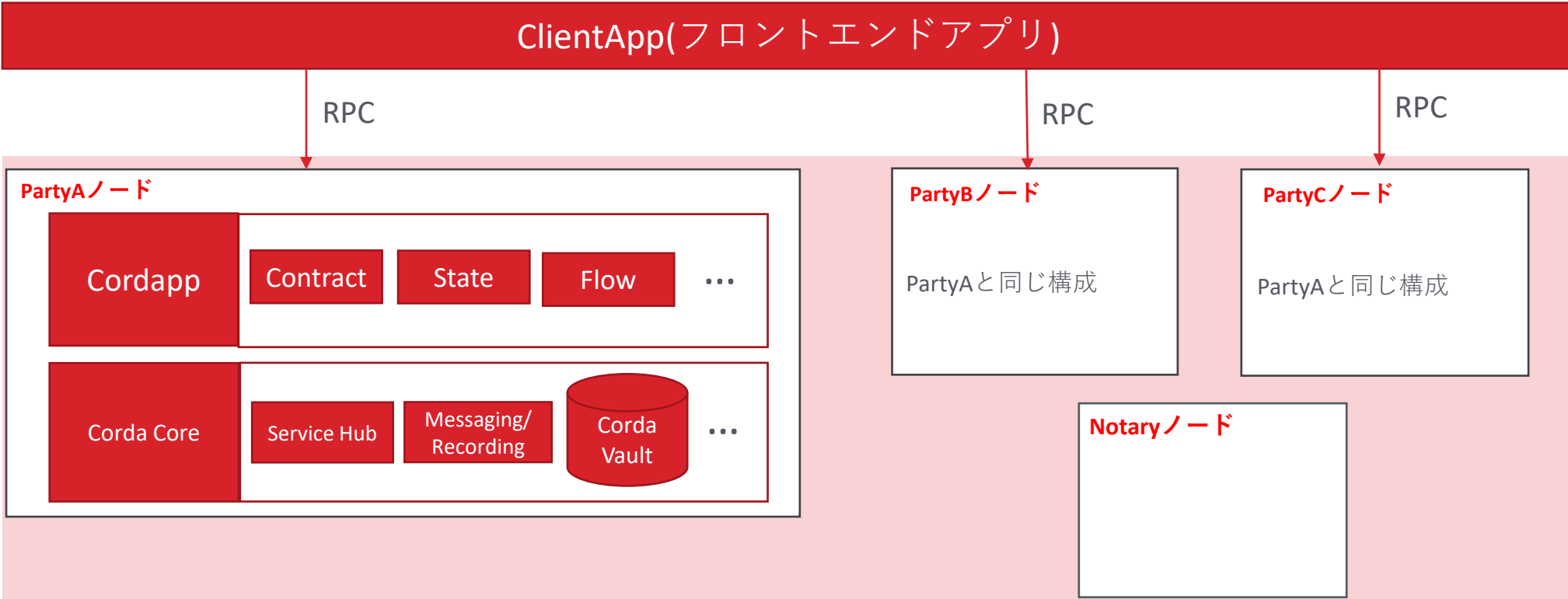
アプリはフロントエンド(React)、 WebAPI(Spring Boot)、 バックエンド (Cordapp\*)の3つの階層から成り立つ。



※Corda上に構築されたアプリケーションを「Cordapp」と呼んでいます

# システムの全体像

フロントエンドアプリからRPCで各ノードに指示を出す。各ノード間の通信はAMQP/TLS通信を行う。Cordappにデータに対する検証や操作の機能を実装する。



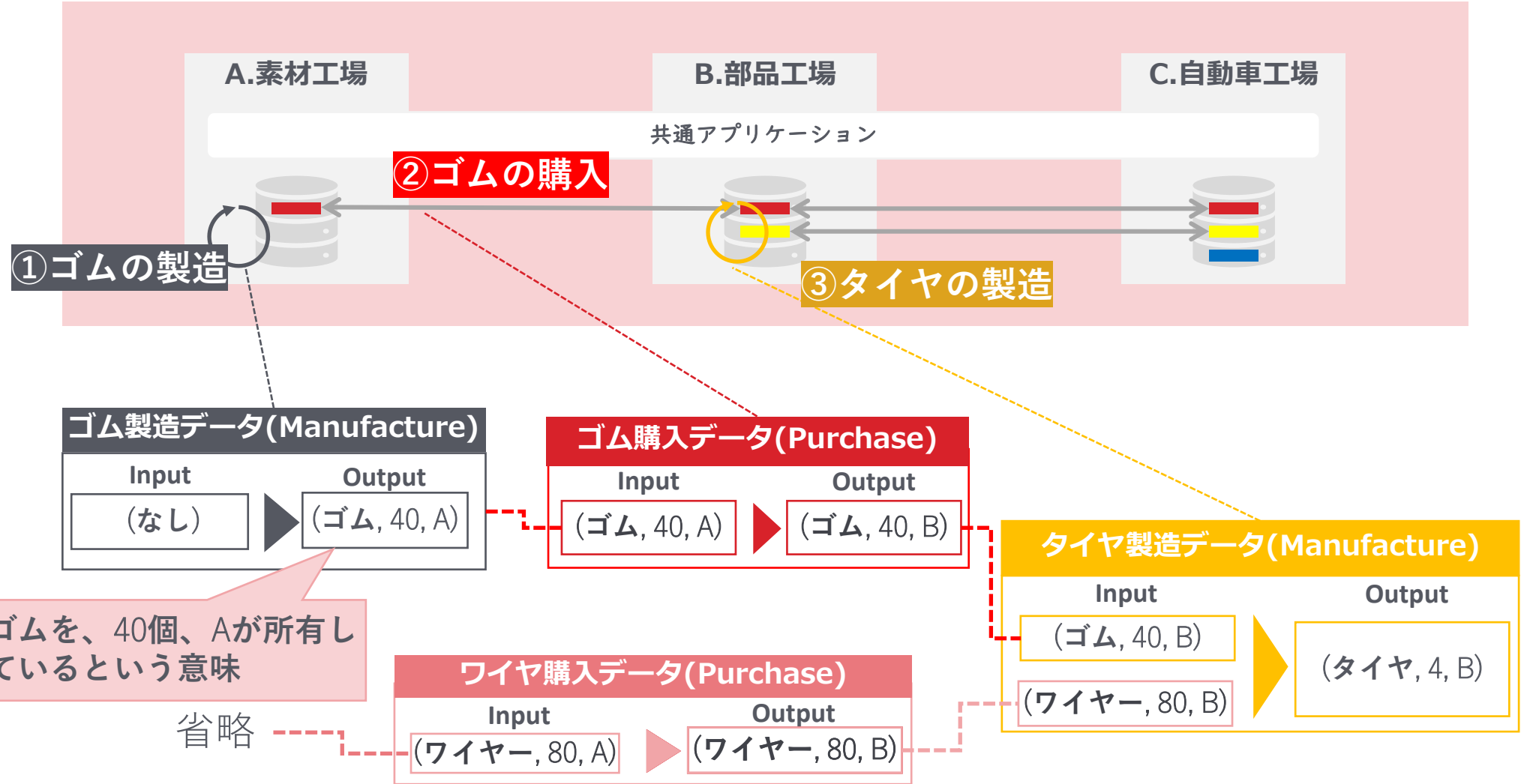
各ノード間はAMQP/TLS通信を行う。

# データの流れの確認

---

# データの流れ

CordaはInputに前のデータのOutputを使用(消費)するUTXOモデルを採用している。





# **Cordapp実装の一部紹介**

---

# Flow

本アプリでは主に3つのFlowを実装している。(今回は時間の都合でGetTraceBackFlowのみ詳細を触れます。)

## ManufactureFlow

➤ 保有している素材や部品を消費して**新しい製品を製造**する。

## TradeFlow

- 他Partyが保有している**製品を購入**する。
- 現状は通貨との交換は発生しない

## GetTraceBackFlow

➤ 製品を作るために使用した部品や素材の情報を取得。

# GetTraceBackFlowの概要

製品のトレース情報(TraceInfoクラス)を返す。

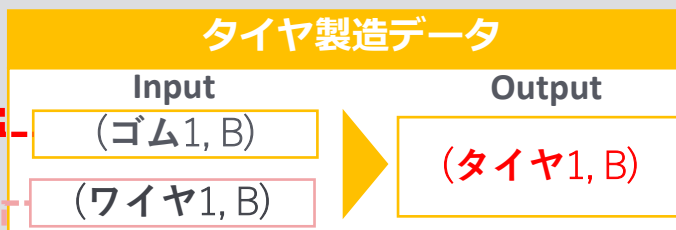
GetBackTraceFlowを実行



### TraceInfoクラス(トレース情報)

ロット番号：タイヤ1  
製品名：タイヤ  
製造者：B  
製造日：8/25  
子の製品情報：下記2つのTraceInfoクラス

TraceInfo	TraceInfo
ロット番号：ゴム1 製品名：ゴム 製造者：A 製造日：8/25 子の製品情報：null	ロット番号：ワイヤ1 製品名：ワイヤ 製造者：A 製造日：8/25 子の製品情報：null

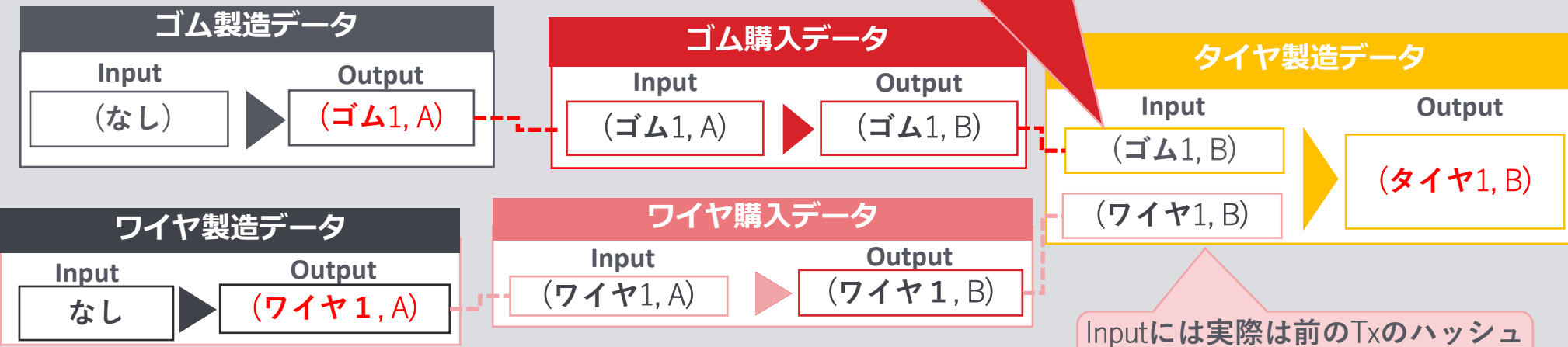


# GetTraceBackFlowの技術的実装

トレース情報を取得するためにトランザクション(以下Tx)を遡る。

Inputに含まれるから次の情報からTxを遡り、トレース情報を取得する。

- ・ 前のTxのハッシュ値
- ・ インデックス



Inputには実際は前のTxのハッシュ値とインデックスが含まれる

# まとめ

---

- 本SCMアプリは、製品の製造、購入を企業間の不整合なく行うことができる。
- 本SCMアプリは「モノ」をいつ、だれが製造したのかを、自社のDBを見るだけで把握できる。
- Cordappの“機能”はFlowに実装する。
- CordaはUTXOモデルを使用しているため、簡単にトレースバックが行える

**SCMアプリは現在は物流だけですが、様々な用途で拡張して活用することができます。**

**本アプリを追加開発するPoC等大歓迎ですので、  
ご興味ある方はお声掛けいただけますと幸いです。**

p.s. 個人的には野球、卓球談議也大歓迎です。

# PM・エンジニア募集中

- ブロックチェーンCordaを活用して、グループ内外のDX支援を行いたいエンジニア、PMを募集しています。
- Javaの開発経験またはプロジェクトマネジメントの経験がある方、大歓迎です。



こちらのQRコードからお申し込みください。

