

Cordaインフラ構築の留意点

～ the Corda Network構成の利用に関して～

株式会社シーエーシー
ビジネス統轄本部 デジタルITプロダクト部
ブロックチェーン推進グループ
山田祐司



株式会社 **シーエーシー**

目次

- Cordaのネットワーク構成概要
 - BC Post Tradeについて
 - Cordaのネットワーク構成について・比較
- the Corda Network(tCN)利用の流れ
 - the Corda Network(tCN)について
 - tCN側のサービス稼働状況確認
 - サポートについて
- まとめ

BC Post Tradeについて

The screenshot shows the SBI Holdings website with a news article titled "外国為替コンファメーションシステム「BCPostTrade」リリースに関するお知らせ ~ブロックチェーン/分散型台帳技術を採用したシステムを実用化~". The article is dated 2020年4月16日 and is from SBIホールディングス株式会社 and SBIリクイディティ・マーケット株式会社. The diagram below the article illustrates the transition from a traditional email-based confirmation process to a Corda-based system. The "Before" state shows two parties exchanging data via email, with a list of drawbacks: difficult to search for evidence, inconsistent methods, data silos, and risk of email misdelivery. The "After" state shows a Corda-based process where data is shared on a web platform, allowing for common confirmation data, web-based input/viewing, and shared data between related parties, which is easier and more secure.

注文 → **コンファメーション** → 決済

Before

- 証拠が残りにくい、探しにくい
- 会社ごとに方法がばらばら
- データは各社分断されている
- メール誤送信等ミスの発生リスク

After

- コンファメーションデータの共有
- WEB上でデータの入力、閲覧可能
- 関係者間でデータ共有されるので改ざんしにくい

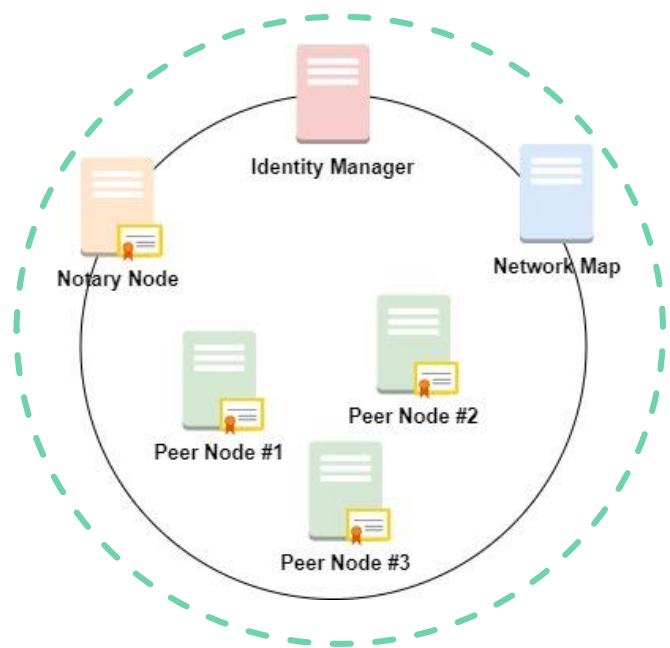
イメージ概要

https://www.sbigroup.co.jp/news/2020/0416_11934.html

企業間取引ではパブリック型ブロックチェーンのようにビジネスネットワーク内の全ての参加者へ情報を伝搬することが相応しくないケースが多く見受けられますが、Cordaを採用することで情報を共有する範囲を限定(プライバシー)した取引が複数実現可能となり、取引範囲間における価値の移動(インターオペラビリティ)も行うことができます。さらにブロックチェーン技術の1つの特徴であるUTXO(Unspent Transaction Output)モデルを採用しつつもコンセンサスアルゴリズムを情報の共有相手とノタリーと呼ばれる2重支払防止用ノードのみでファイナリティを与える仕組みとなっておりますので、パフォーマンスを維持しながらの参加者の追加が容易な構造となっております(スケーラビリティ)、本リリースではCorda Network Foundationが提供するthe Corda Networkサービスの利用により既存の参加者ノードからの利用も可能となっております。また、Cordaはkotlinで開発されており、アプリケーション開発もJavaまたはkotlinを使用できますので、開発はもとより保守運用に関しても生産性が高いと言えます(プロダクティビティ)。

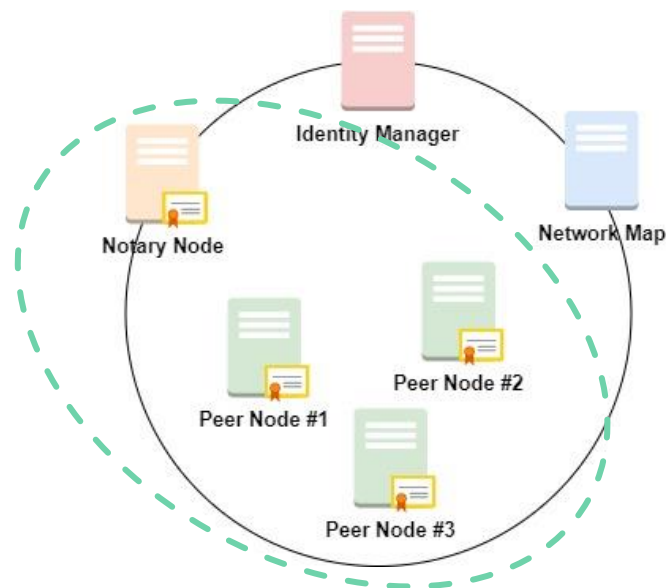
Cordaのネットワーク構成

Private Network



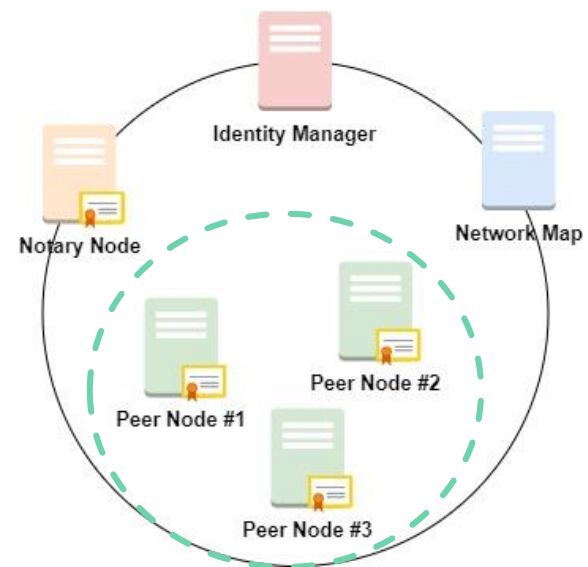
ネットワーク運営：独自
Notary運営：独自

Segregated Network



ネットワーク運営：Corda Network Foundation
Notary運営：独自

the Corda Network(tCN)



ネットワーク運営：Corda Network Foundation
Notary運営：Corda Network Foundation

..... 構築範囲

Cordaのネットワーク構成

各コンポーネントの取り扱い

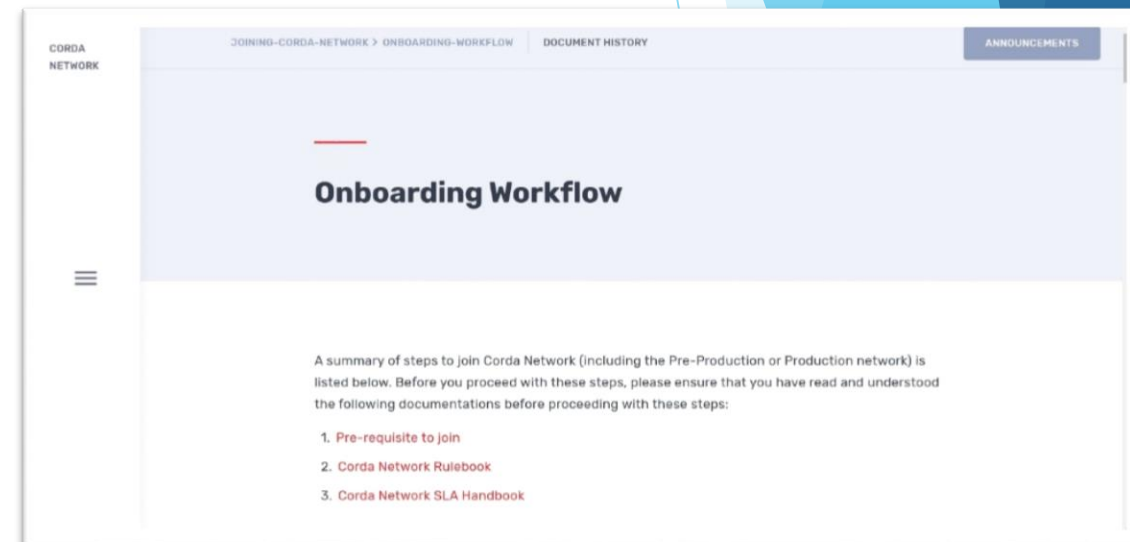
	Private Network	Segregated Network	the Corda Network(tCN)
管理	必要	Notaryのみ必要	不要
運用	必要	Notaryのみ必要	不要
コントロール	可能	Notaryのみ可能	不可
外部NWとの接続	無し(構成次第)	tCN側との接続あり (CordaFW、WAF併用)	tCN側との接続あり (CordaFW、WAF併用)

tCNを利用するメリット

⇒Trust Root/Identity Manager/Network Map Serviceに関してはNetwork構成管理が主であるため、その部分を自前での管理・運用が不要(tCNに委譲)となる。

the Corda Network(tCN)について

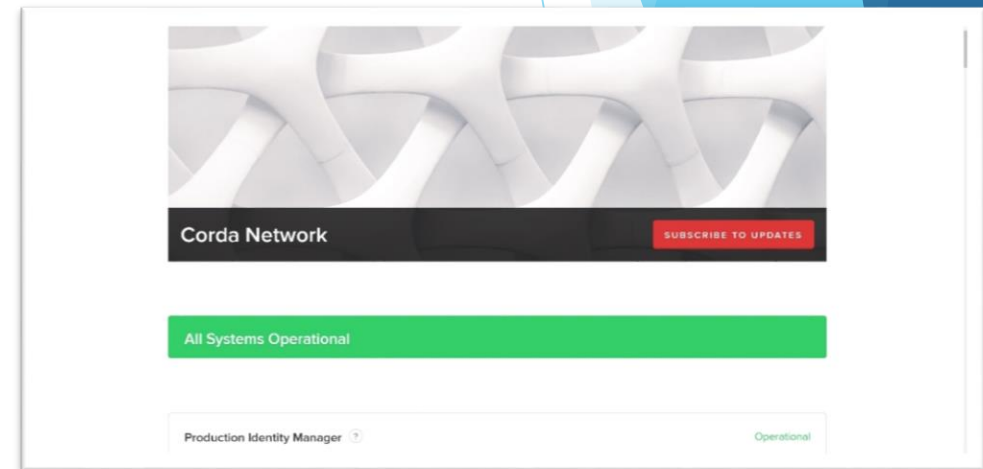
- ▶ 環境 . . . Pre-production/productionの2面
- ▶ 参加手順 . . . ドキュメント手順に従って登録作業を行う
⇒申請・事前承認後、ノード登録作業の流れ
- ▶ 実作業としての所感
 - ▶ CENM(IdentityManager等)の利用と同様
 - ▶ 証明書発行はCorda Network Foundation側の承認作業が必要
⇒CSR後少し待つ必要あり、途中接続切れても再接続可
※IDファイル(certificate-request-id.txt)は確保
 - ▶ HSM利用の場合には事前にセットアップ
⇒登録時にキー作成される為その前に用意するとスムーズ
 - ▶ CordaFW利用の場合は、後からセットアップでも可
 - ▶ 最終的に接続される宛先をnode.confに設定
 - ▶ CRLとの接続はNodeとFW(Float、Bridge)の全てから接続あり
 - ▶ node.confにメールアドレスが必須



<https://corda.network/>

tCN側のサービス稼働状況確認

- ▶ ステータスページで計画的なメンテ連絡(サービスへの影響等)
 - ▶ 開始、終了等もメールによる通知
- ▶ メンテ例
 - ▶ メンテにおいては機能的な影響は基本的に無し
 - ▶ IM、NMSの一部機能が一時的に利用不可のケースはあり。
(事前に作業予定連携有り。機能的には影響無し)
 - ▶ Notaryが利用不可となったことは無し。
⇒Notaryに関してはヘルスチェックツールも用意されている



<https://corda-network.statuspage.io/>

SBI R3 Japanサポート

- ▶ サポートデスク経由での照会
 - ▶ 技術的サポート
 - ▶ 対外的な照会とりまとめ
 - ▶ R3
 - ▶ Corda Network Foundation
 - ▶ Cordaの新バージョンのリンク連携
- ▶ 緊急の場合は電話で連絡



<https://support.sbir3japan.co.jp/hc/ja>

まとめ

- ▶ tCNを利用することでネットワーク管理/運営が不要
- ▶ 導入/利用はドキュメントが整備されている（使用感もCENMと同じ）
- ▶ 運用についてもサポート受けながら実施

Corda RPC APIの再接続

株式会社シーエーシー
ビジネス統轄本部 デジタルITプロダクト部
ブロックチェーン推進グループ
劉 昊



株式会社 **シーエーシー**

目次

- RPC接続中断
 - ▶ 原因
- CordaのRPC再接続機能
 - ▶ 仕組み
 - ▶ 必要な処理
 - ▶ コードサンプル
- まとめ

RPC接続は中断される可能性がある(1/2)

- ノードとの接続はCorda RPC Clientを使用
- RPC接続は長生き：
 - ▶ 接続自体を1回作成して、使い回し
 - ▶ trackByでvaultの更新を監視する

RPC接続は中断される可能性がある(2/2)

- しかし、接続は途中で切断（瞬断）される可能性がある
 - ▶ ノードが止まった
 - ▶ TCP接続が中断された
 - ▶ . . .
- その場合再接続する必要がある

GracefulReconnectで再接続

GracefulReconnectは自動的に再接続する仕組み：

- ▶ 切断時にObservableは自動的に再接続する
- ▶ 再接続中に新しいRPC callをブロック

以下の設定が可能：

- ▶ maxAttempts: 最大の接続リトライ回数
- ▶ onDisconnect : 接続切断時のコールバック
- ▶ onReconnect : 再接続成功時のコールバック

再接続の場合

Side effect有無次第で対応方法は異なる

- ▶ Side effect : vaultに影響

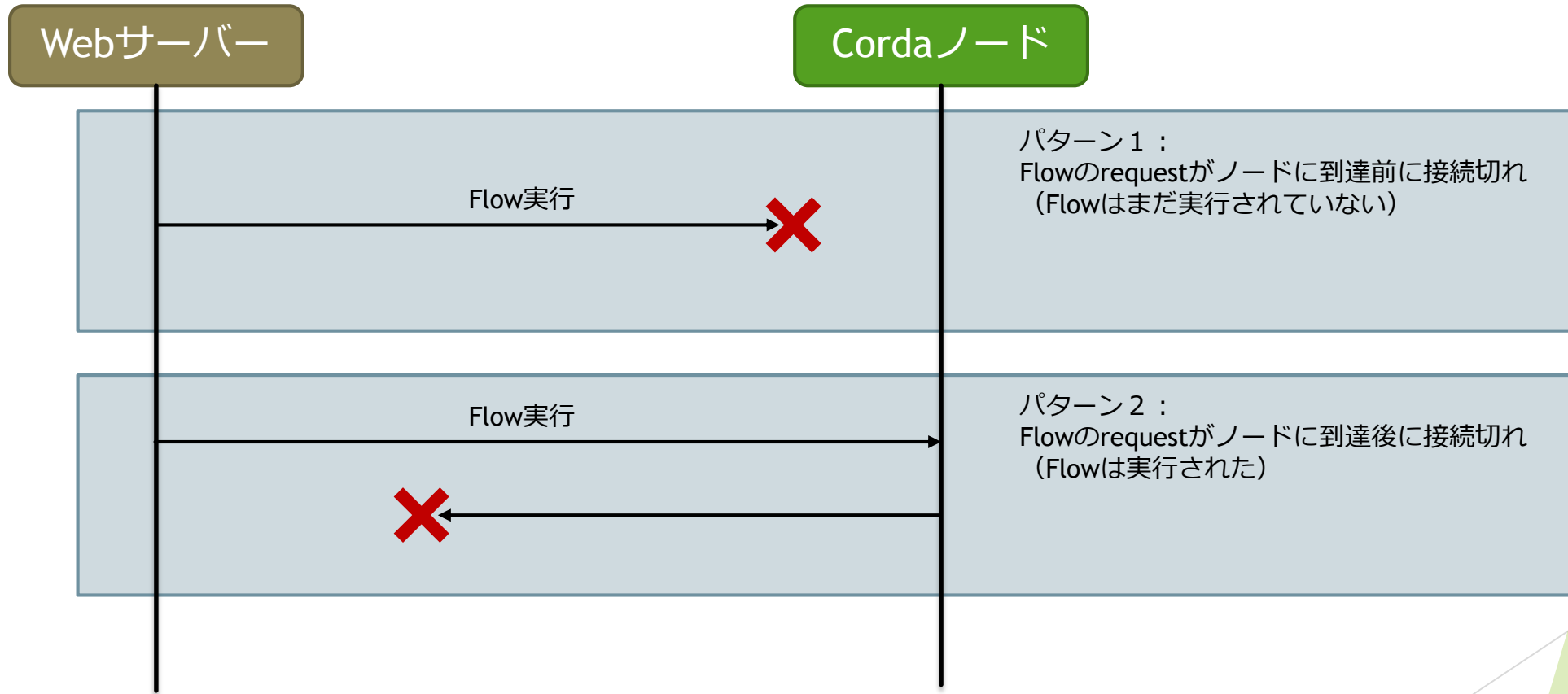
Side effect無し : nodeの情報取得、stateのクエリ等

- ▶ 特に対応する必要がない

Side effect有り : Flowの実行等

- ▶ 先ずはFlow実行の状況を確認

Flow実行時接続切れのパターン（2つ）



- ▶ WebサーバーからFlow実行されたか検知出来ないため、Flow実行の状況を確認する必要がある

Flow実行状況の確認

Corda 4.6から、Flow実行にclient IDを指定することが可能

- ▶ 同じIDでFlow複数回の実行は不可能
- ▶ 後からIDでFlowの実行状況（成功・失敗）を確認出来る
- ▶ IDはClient側で管理する

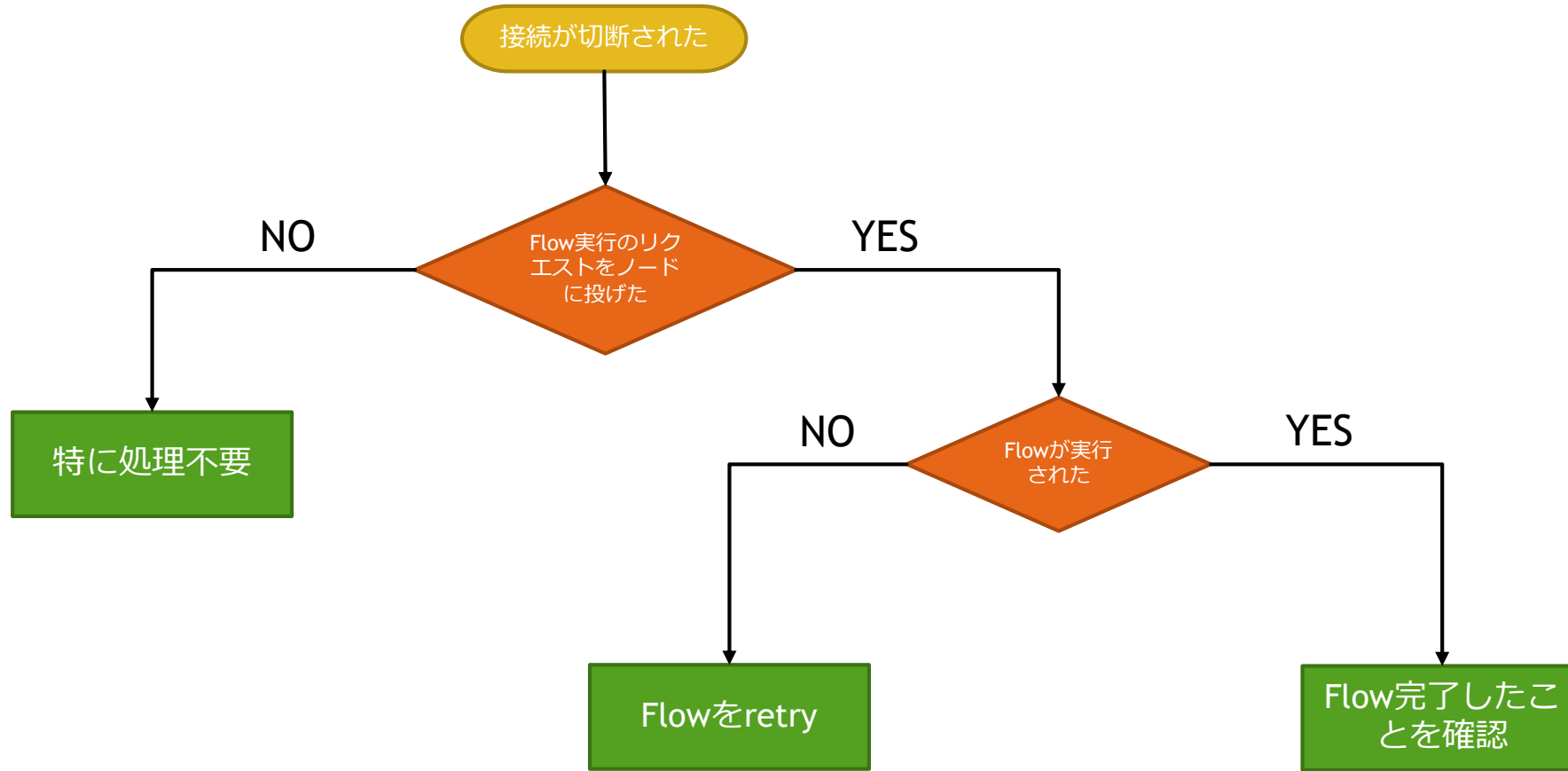
- ▶ `startFlowDynamicWithClientId` でFlowを実行するだけ：

```
String clientId = UUID.randomUUID().toString();  
FlowHandleWithClientId flowHandleWithClientId = proxy.startFlowDynamicWithClientId(clientId, Flow.class);
```

再接続処理サンプル

```
private void onDisconnect() {  
    // 接続切断時の処理。ログ出力等  
}  
  
private void onReconnect() {  
    List<String> unfinishedFlowClientIds = getClientIdsFromDatabase(); // DBから保存したClient IDを取得  
  
    // 各Flowの実行状況を確認し、未実行のFlowを再実行  
    unfinishedFlowClientIds.forEach(clientId -> checkFlowResultByClientIdWithRetry(clientId));  
}  
  
void callFlow(Flow flow) {  
    GracefulReconnect gracefulReconnect = new GracefulReconnect(this::onDisconnect, this::onReconnect, 3);  
    CordaRPCClient cordaClient = new CordaRPCClient(nodeRpcAddress);  
    CordaRPCConnection proxy = cordaClient.start(rpcUserName, rpcUserPassword, gracefulReconnect);  
  
    String clientId = UUID.randomUUID().toString();  
    persistClientId(clientId); // DB等にClient IDを保存する  
  
    FlowHandleWithClientId flowHandleWithClientId = proxy.startFlowDynamicWithClientId(clientId, Flow.class);  
  
    removePersistedClientId(clientId); // 実行成功の場合、Client IDをDBから削除  
}
```

まとめ



CordaのRPC再接続機能を利用して、より頑強なシステムを作りましょう