

# ChatGPT



## Examples

"Explain quantum computing in simple terms" →



## Capabilities

Remembers what user said earlier in the conversation



## Limitations

May occasionally generate incorrect information



Corda5とは何ですか？



for a 10 year old's follow-up corrections harmful instructions or  
[ChatGPT Jan 30 Version](#). Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

意外とあってる・・・！！？

2022年初頭までしか学習していないはずなのに・・

# アジェンダ

- Web3将来像とCorda
  - ✓ 5つの論点
- Corda5
  - ✓ 概要
  - ✓ 5つの新アーキテクチャー
  - ✓ 技術スタックとCDSE
  - ✓ 情報ソース

# 自己紹介

## 生永 雄輔（いくなが ゆうすけ）

- ◆SBI Holdings ブロックチェーン推進室 室付部長
- ◆SBI R3 Japan ビジネス推進部長
- ◆SBI Traceability 取締役CTO

### <略歴>

兵庫県出身→大学から上京→ **IT** ⇄ **金融**

学生時代 : ゲーム理論・離散数学 (東大院修士@2004)

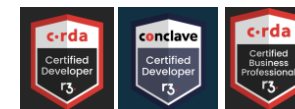
ITベンチャー : 2000-2004 (現像した**写真をネット**で共有)

2009-2013 (自動車/船舶/発電所むけ**大型図面管理**)

金融機関 : 2004-2009 (**メガバンク**で市場調査及び企画)

2013-2018 (**農林系金融機関**で投資ポートフォリオ運営企画)

2018年SBIへJoin



# Web3将来像とCorda

## 5つの論点

# 5つの重要な論点



## 1. 金融系の大規模商用の進展 = 想定取引量

- ✓ DTCC、Visa、Master等の動向
- ✓ 各国CBDCの動向（特に先進国動向）
- ✓ 銀行が提供するステーブルコインの動向

## 2. 規制 & 規則動向

- ✓ ID管理に関する規制、規則の進展
- ✓ トークン（NFT含む）への証券ルール
- ✓ 税制見直し
- ✓ 対ロシア／対中国関連規則（SCM関連 / AML関連）

## 5つの重要な論点（続き）



### 3. インシデント対応

- ✓ FTXに見られる不十分な監督に対する是正
- ✓ NFT周りの詐欺案件の露見

### 4. 金融×非金融ユースケースの進展

- ✓ 境界領域での新しいマーケットの実現

Ex. 貿易金融とeBLの連携／サプライチェーンファイナンス  
etc..

### 5. 異なるブロックチェーン間の接続

- ✓ 境界領域の実現に向けた技術的基礎
- ✓ 非機能要件のクリアに向けた技術発展が必要



**SBI<sup>r</sup>3.**  
Japan

**Corda5**

# c·rda



ビジネスのための基盤

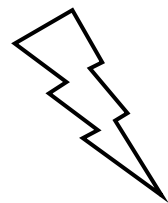


すべての業種に適用可能



異業種交流

# c.rda



パフォーマンスとスケーラビリティ



ID管理とプライバシー



企業間相互運用

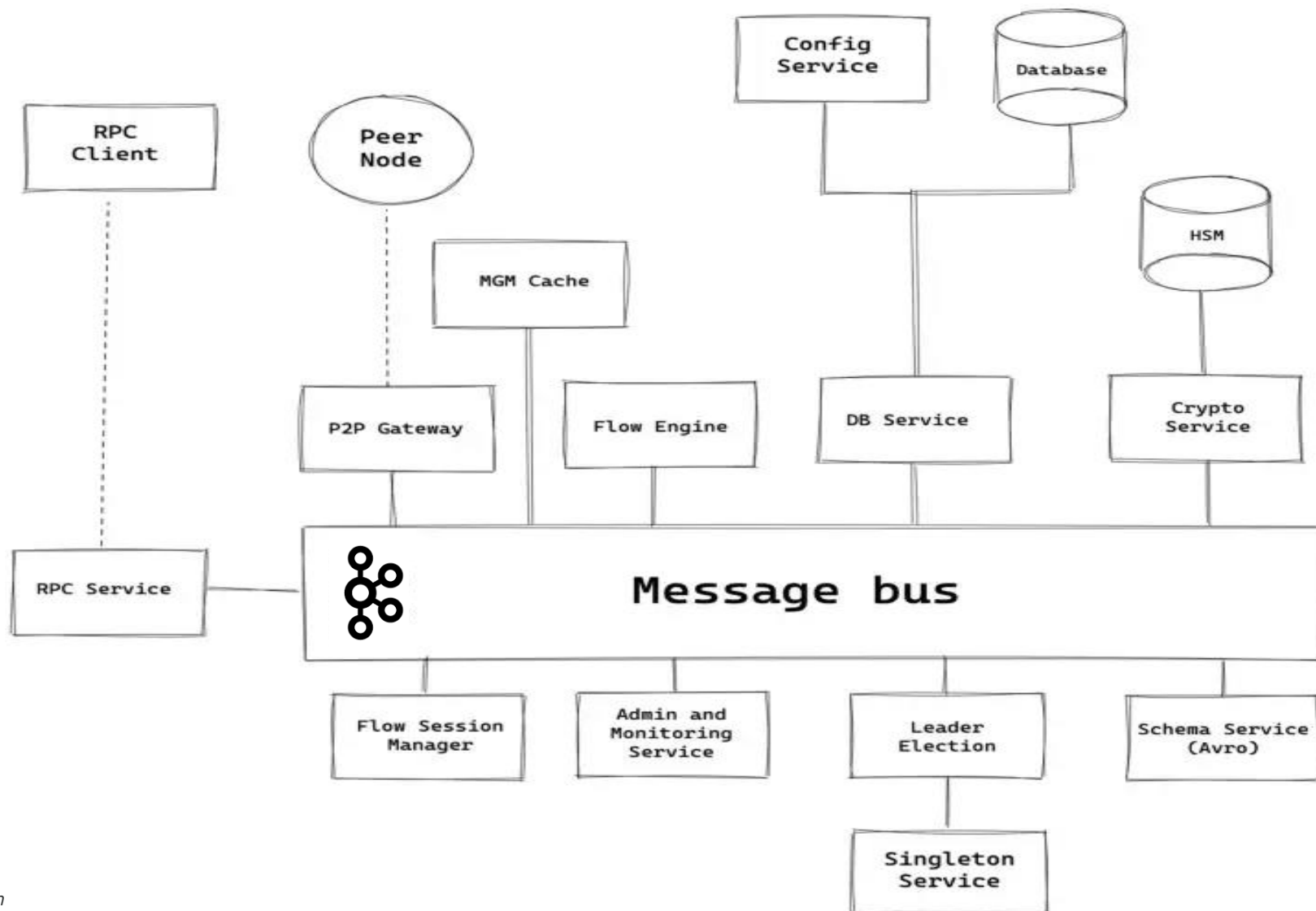
# Corda5

お客様のニーズに合わせて進化した新しいCorda

1. 高可用性
2. 水平方向のスケーラビリティ
3. より良い開発プロセス



# 新しいアーキテクチャ = K8s. + Kafka





# Corda5

5つの新アーキテクチャ

# 5つの新しいアーキテクチャ



## イベント駆動型worker

可用性とスケーラビリティを実現するCordaの新しい"コア"



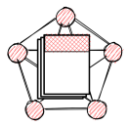
## 仮想Node

論理的にはCorda4と同等、1つのインスタンスに複数の論理Node配置



## モジュール式のAPI

JSON/REST over HTTP



## アプリケーションネットワーク

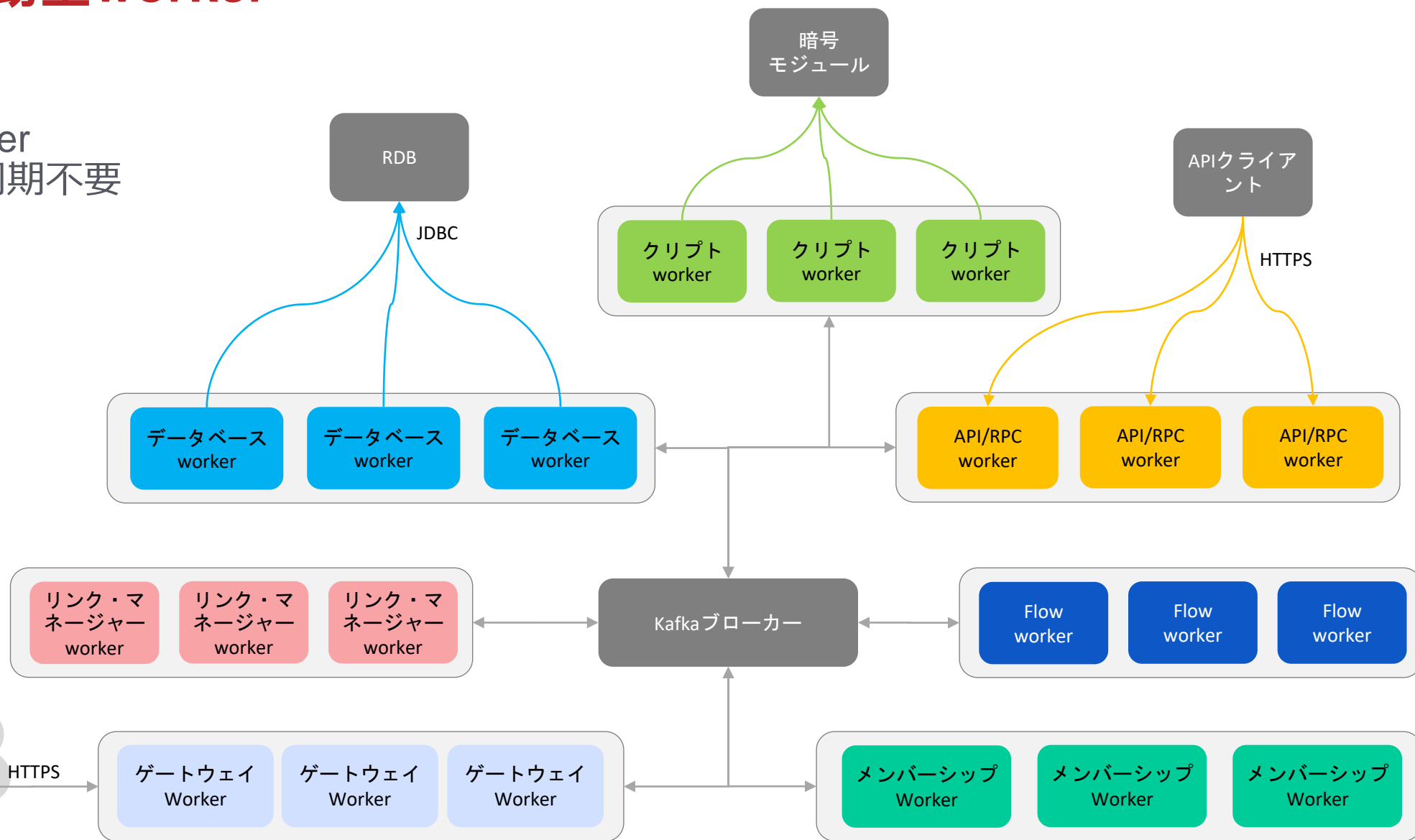
マルチパーティアプリケーション（MPA）に最適化されています。

?

???

# イベント駆動型worker

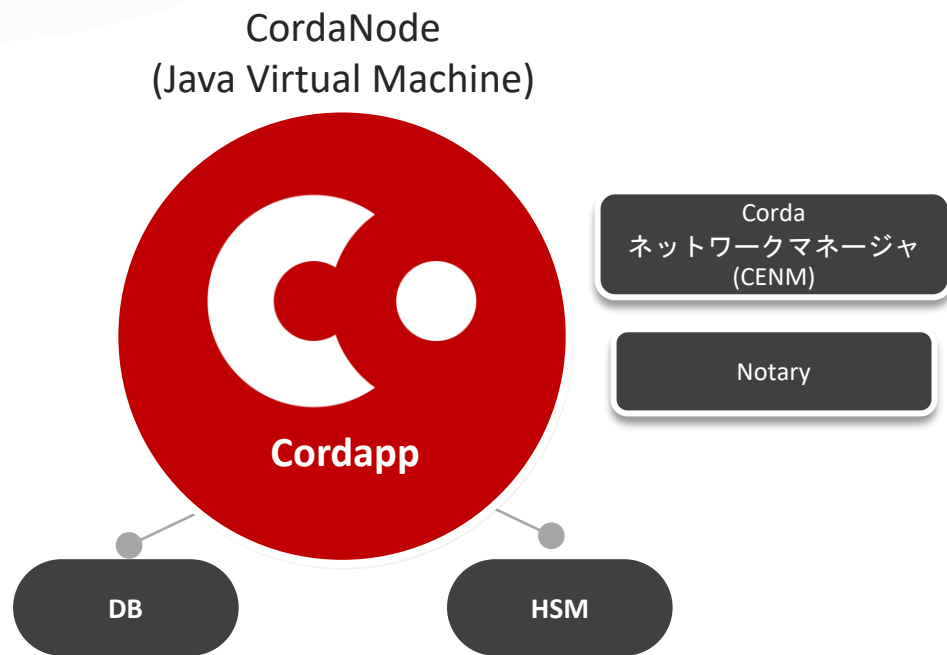
- Stateless worker  
⇒ Worker間同期不要



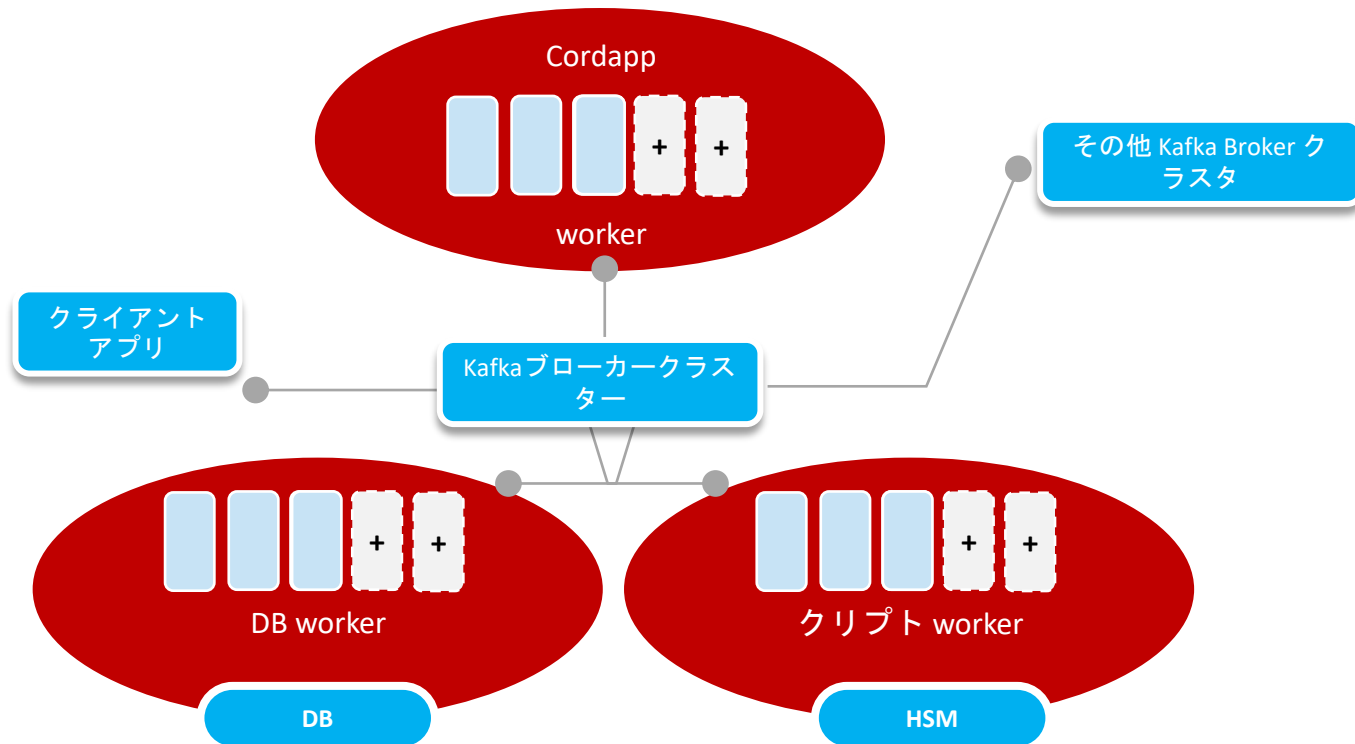
# 水平方向のスケラビリティ

## スケールアップ/ダウンが容易

### Corda4

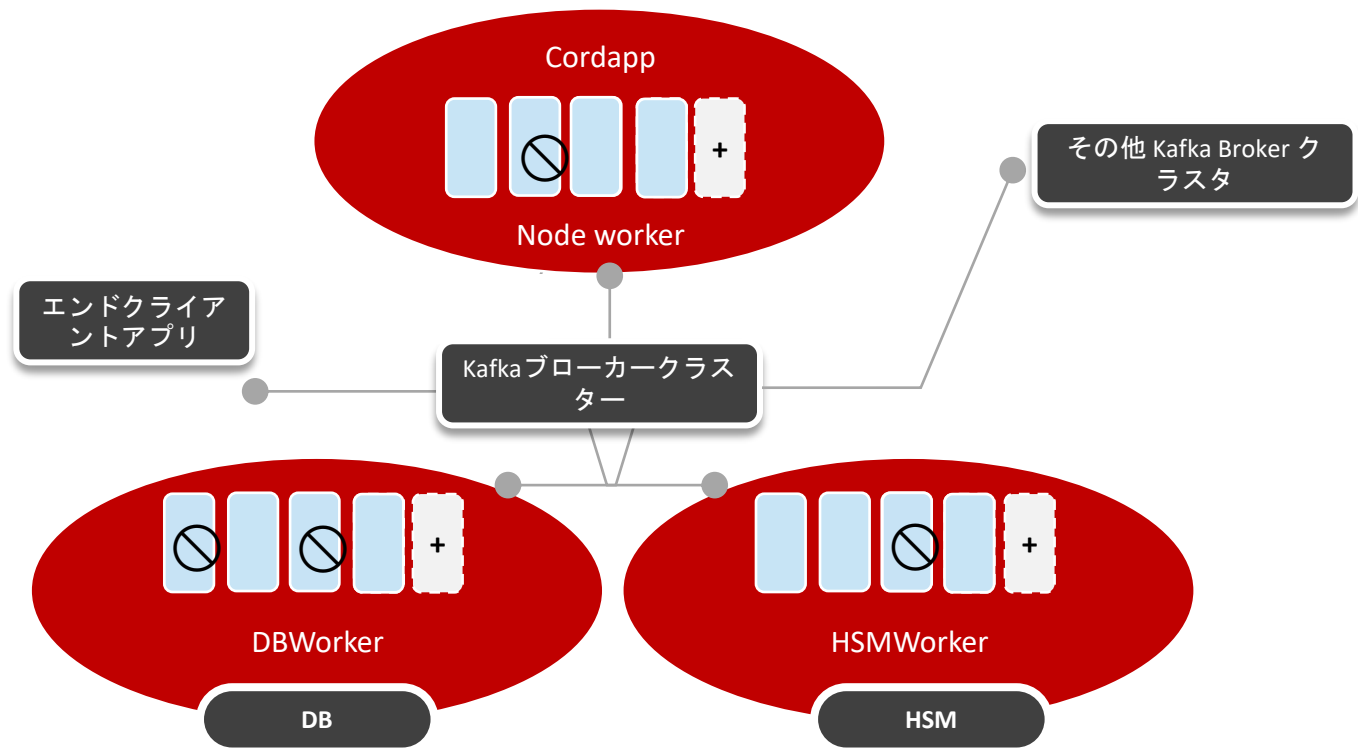


### 5



# 高可用性

ダウンタイムがほとんどない



# 5つの新しいアーキテクチャ



## イベント駆動型worker

可用性とスケーラビリティを実現するCordaの新しい"コア"



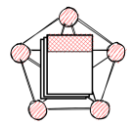
## 仮想Node

論理的にはCorda4と同等、1つのインスタンスに複数の論理Node配置



## モジュール式のAPI

JSON/REST over HTTP



## アプリケーションネットワーク

マルチパーティアプリケーション（MPA）に最適化されています。

?

???

# 仮想Node

- 論理的な「仮想」Node
  - プロセス、データ、アイデンティティのマッピング
- 1つの[workerクラスター] = 複数の[仮想Node]
- サンドボックス技術により、互いに分離・隔離

## Why ?

- 柔軟な運用モデル
- 可搬性
- 隔離性

Virtual nodes



Worker cluster



# 5つの新しいアーキテクチャ



## イベント駆動型worker

可用性とスケーラビリティを実現するCordaの新しい"コア"



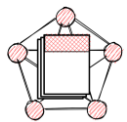
## 仮想Node

論理的にはCorda4と同等、1つのインスタンスに複数の論理Node配置



## モジュール式のAPI

JSON/REST over HTTP



## アプリケーションネットワーク

マルチパーティアプリケーション（MPA）に最適化されています。

?

???

# HTTP JSON/REST API + モジュール化API

## ✓ Netowrk Interface

安全なネットワーク構築とP2Pベースコミュニケーション

## ✓ Workflow

データにとらわれない、関係者間調整を実現するための分散API

## ✓ Shared Ledger

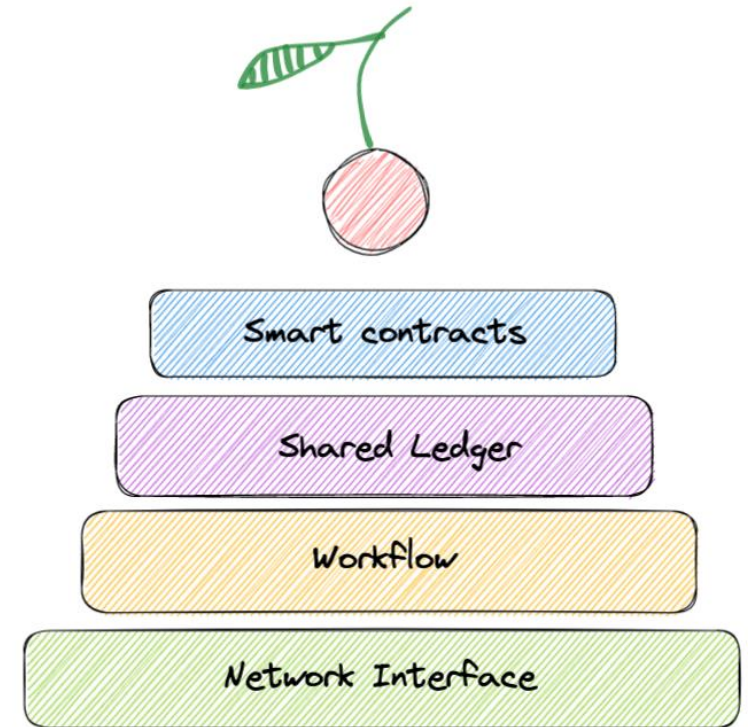
当事者間の信頼性の高いデータ合意（WYSIWIS）

## ✓ Smart Contract

ビジネスロジックの非中央集権的、プログラムによる実行

### Why?

- 柔軟な開発が可能  
（ワークフローとコントラクトロジックのどちらかだけ実装する等）
- 業界標準のプロトコルとツールの活用



# 5つの新しいアーキテクチャ



## イベント駆動型worker

可用性とスケーラビリティを実現するCordaの新しい "コア"



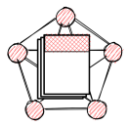
## 仮想Node

論理的にはCorda4と同等、1つのインスタンスに複数の論理Node配置



## モジュール式のAPI

JSON/REST over HTTP



## アプリケーションネットワーク

マルチパーティアプリケーション（MPA）に最適化されています。

?

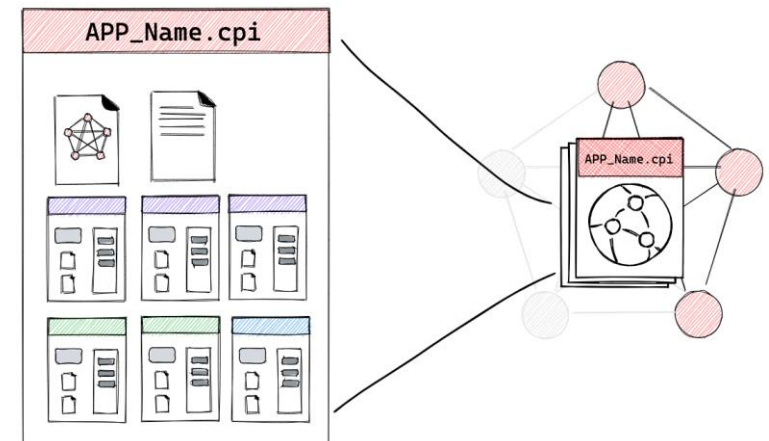
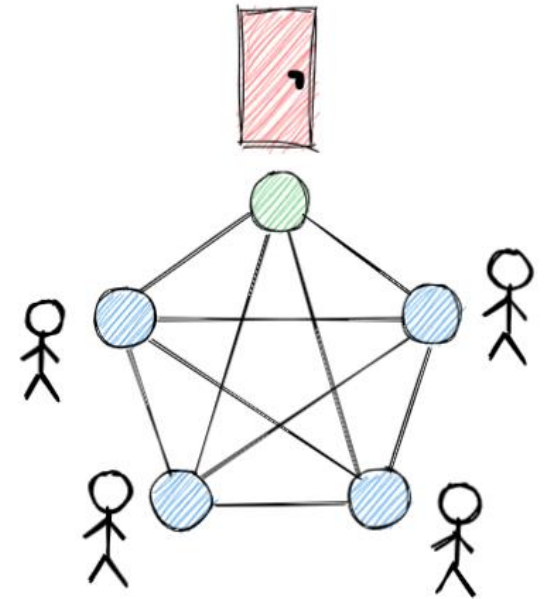
???

# アプリケーションネットワーク

- 仮想Nodeから構成されるネットワーク
- 個別の接続権限は、Node単位で管理
- ネットワークとアプリがワンセットで構成
- 単一のアーティファクト（インストーラ）による構成管理
  - アプリケーション
  - ネットワークの詳細
  - 参加者要件

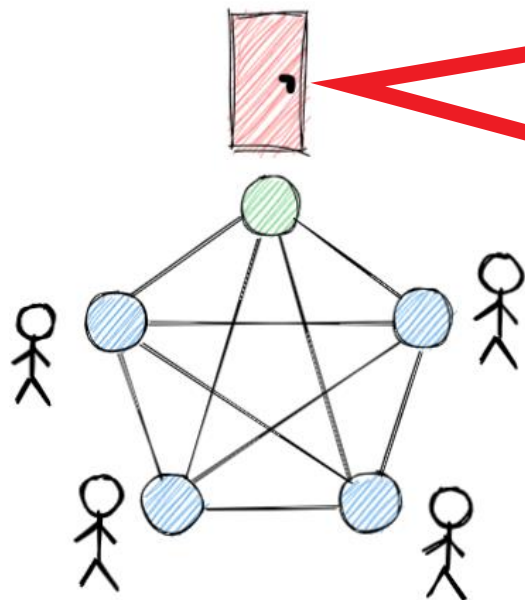
## Why?

- 事前調整の軽減
- よりシンプルなネットワーク管理

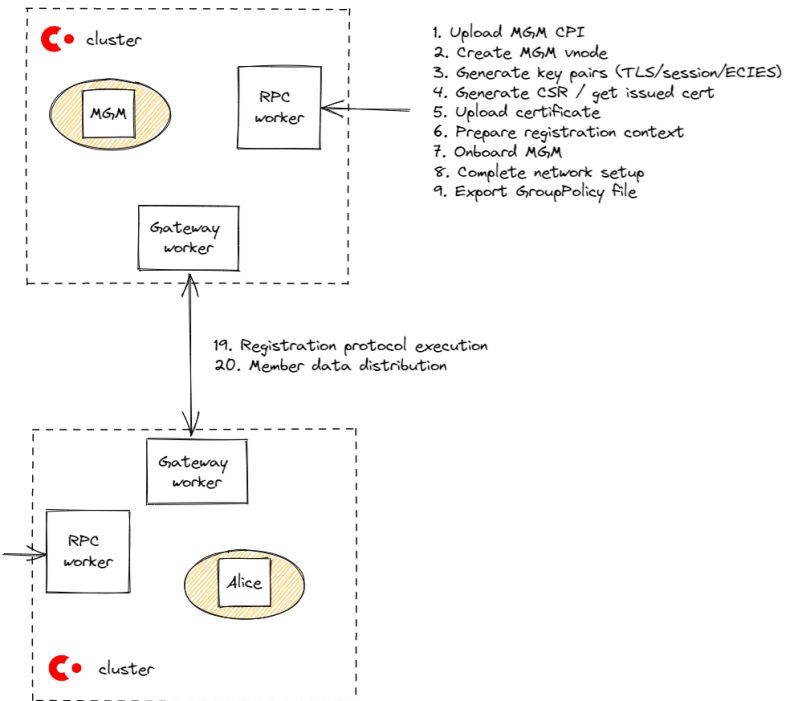


# アプリケーションネットワーク：ネットワーク構成

- メンバーシップグループマネージャー（MGM）が管理



10. Build member CPI
11. Upload member CPI
12. Create vnode (Alice)
13. Generate key pairs (identity/ledger/session)
14. Generate CSR / get issued cert
15. Upload certificate
16. Complete network setup
17. Prepare registration context
18. Register member



# 5つの新しいアーキテクチャ



## イベント駆動型worker

可用性とスケーラビリティを実現するCordaの新しい"コア"



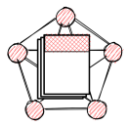
## 仮想Node

論理的にはCorda4と同等、1つのインスタンスに複数の論理Node配置



## モジュール式のAPI

JSON/REST over HTTP



## アプリケーションネットワーク

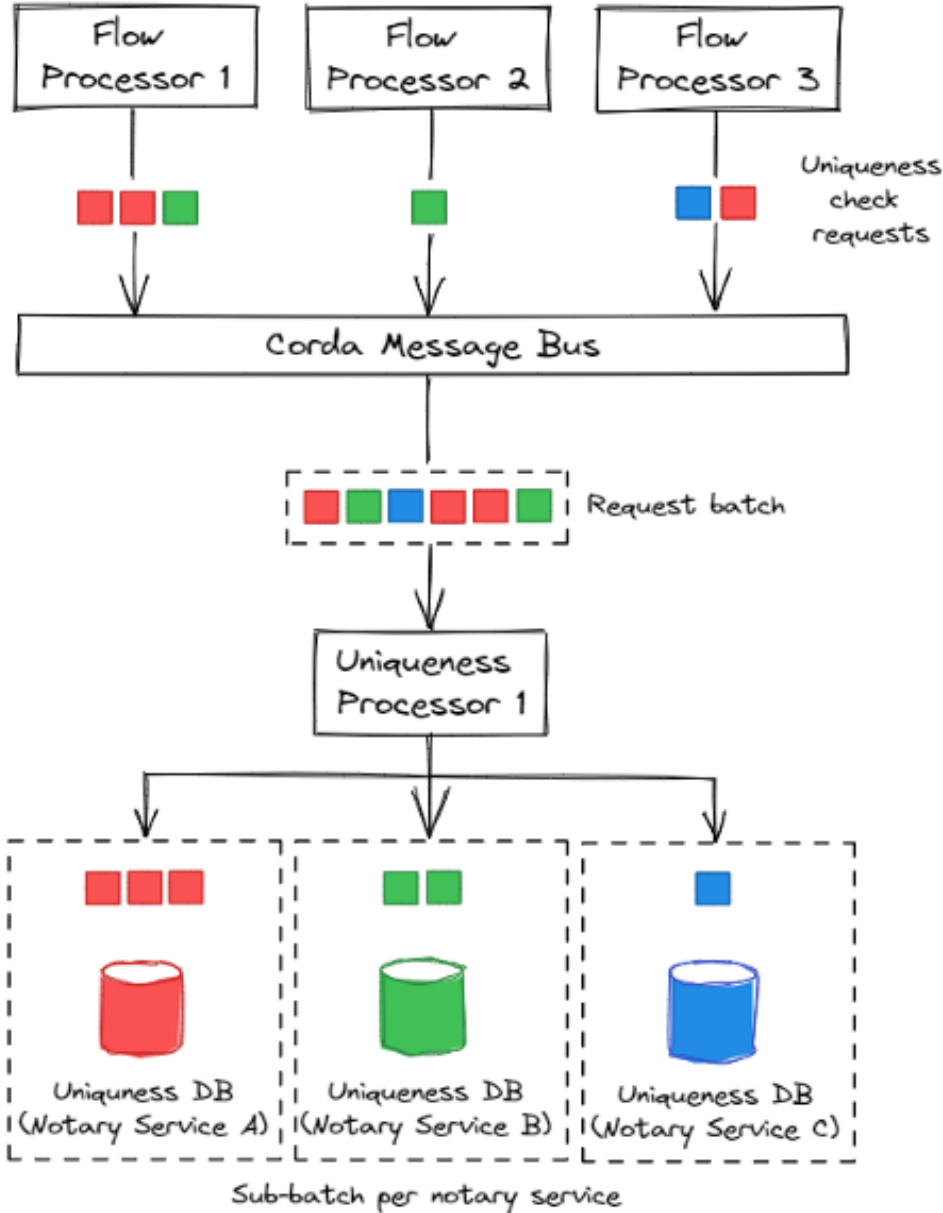
マルチパーティアプリケーション（MPA）に最適化されています。

## プラグブル Notary

# プラグブル Notary

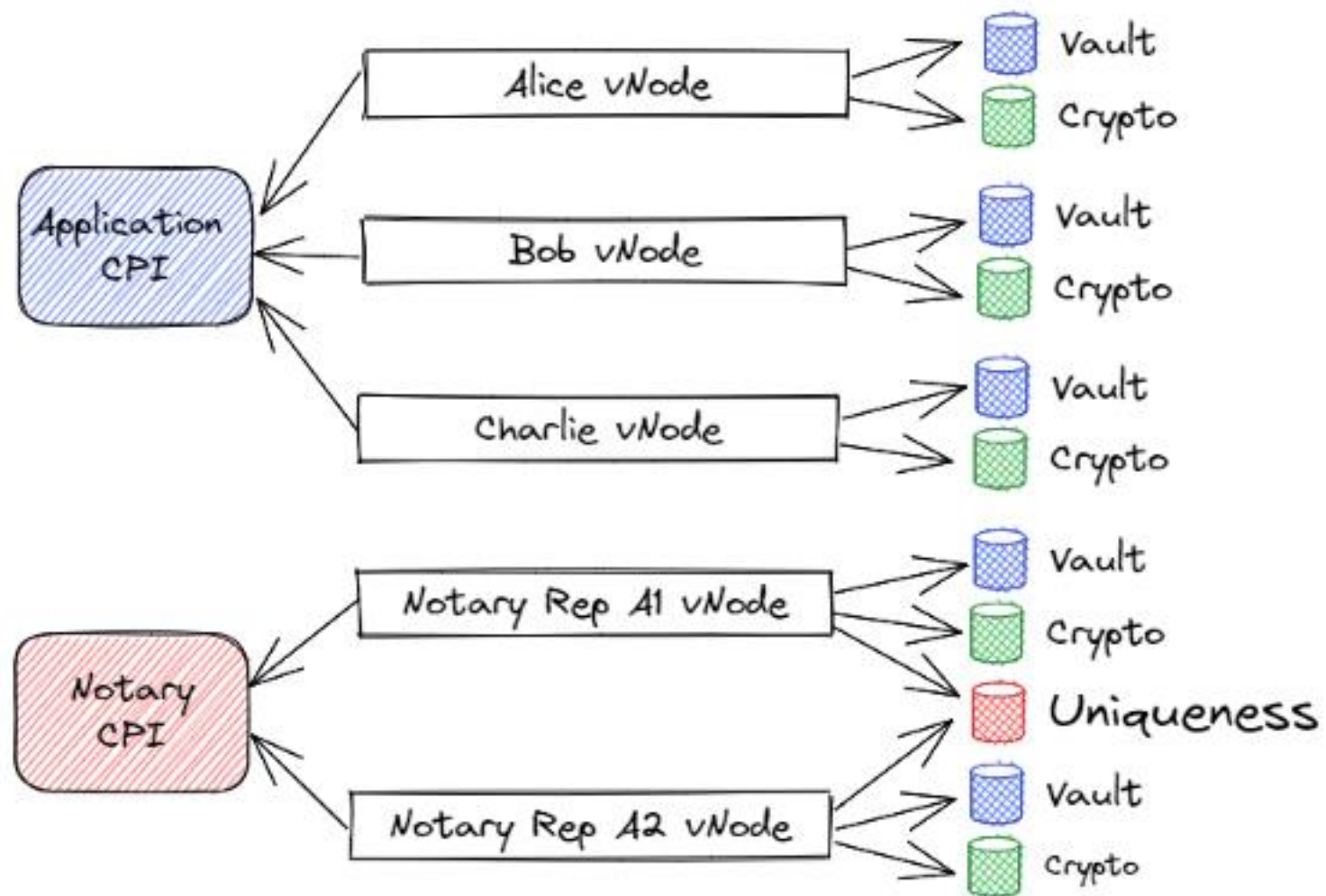
仮想Node + ユニークネス管理

= 仮想Notary



# プラグブル Notary

- Notary専用のFlowを設定可 (Notary CPI)
- Non-Validating Notary
- Validating Notary  
以外の実装も可能



# 技術スタックとCDSE

# インフラに関する技術スタック



- Kubernetes

- ✓ 複数サーバーにまたがる可用性プロセス（コンテナ）の維持



- PostgreSQL

- ✓ 永続化（Corda、CorDappsとも）



- Kafka

- ✓ worker間通信

ローカルでのテスト展開の場合。

- Docker + Minikube



docker



# Helm



- Kubernetesのパッケージマネージャ
- テンプレートに対応し、インストール時に一部項目を上書き可能

## •用語

- Chart - テンプレート構成とデフォルト値のコレクション
- リポジトリ-バージョン管理されたHelmチャートのコレクションをホスティング。

# Corda標準開発環境(CSDE)

## Corda Standard Development Environment

- 開発の起点となるCorDappプロジェクト
- Corda 5に必要なworkerの全てがテスト用に含まれているサンドボックス
- <https://github.com/corda/CSDE-cordapp-template-kotlin>

Software	Version
Operating systems	<ul style="list-style-type: none"><li>• Mac OS (intel and ARM)</li><li>• Windows 10/11</li><li>• Linux</li></ul>
Java	Azul Zulu JDK 11 (Other versions should work but have not been extensively tested.)
Intellij	~v2021.X.Y community edition
git	~v2.24.1
Docker	Docker Engine ~v20.X.Y or Docker Desktop ~v3.5.X

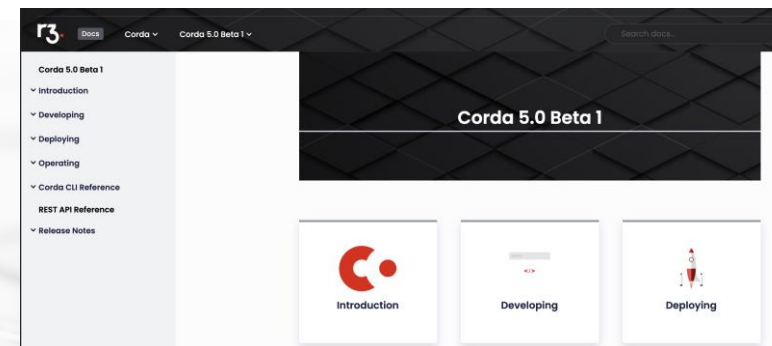


# 情報ソース

# 情報ソース

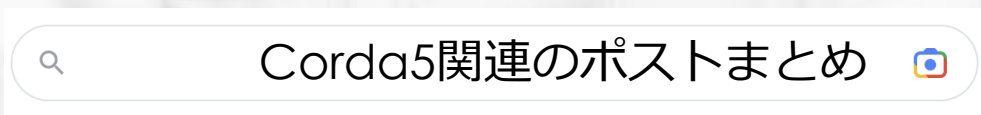
## 英語

- ・ 公式ドキュメント
- ・ <https://docs.r3.com/en/platform/corda/5.0-beta.html>



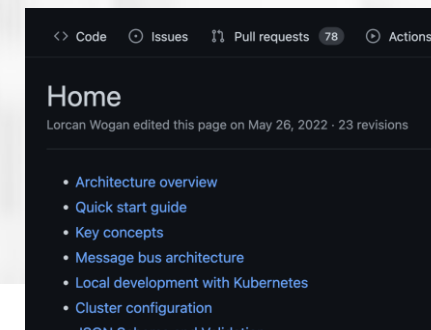
## 日本語

- ・ 概要説明 :



## ソースコード

- ・ <https://github.com/corda/corda-runtime-os/wiki>
- ・ <https://github.com/corda/corda-api>
- ・ <https://github.com/corda/corda-cli-plugin-host>
- ・ <https://github.com/corda/corda-dev-helm>
- ・ <https://github.com/corda/CSDE-cordapp-template-kotlin>



# アンケート

**春の人で使うDemo環境の希望を教えてください。**

# SBI r3.

## Japan

# アプリケーションのパッケージングとインストール

- **CPI = Corda Package Installer**
- **自己完結型のインストーラー**
- **アプリケーションとネットワークポリシーの設定に関する情報を含む**

## CPIメタデータ

- CPIフォーマット
- CPIバージョン
- CPI名
- CPIアップグレード (FlowDrain要否)

## ネットワークメタデータ

- 証明書のトラストルート
- MGM情報
- アイデンティティPKI
- アイデンティティキーポリシー
- サイファースイート

