

# 実践エンタープライズブロックチェーン

システム設計・運用における課題とそのソリューション

日本オラクル株式会社  
クラウド事業戦略統括  
クラウドプラットフォームソリューション部  
中村 岳

**Modern Cloud Day Tokyo**

次世代クラウドが変える日本のビジネス



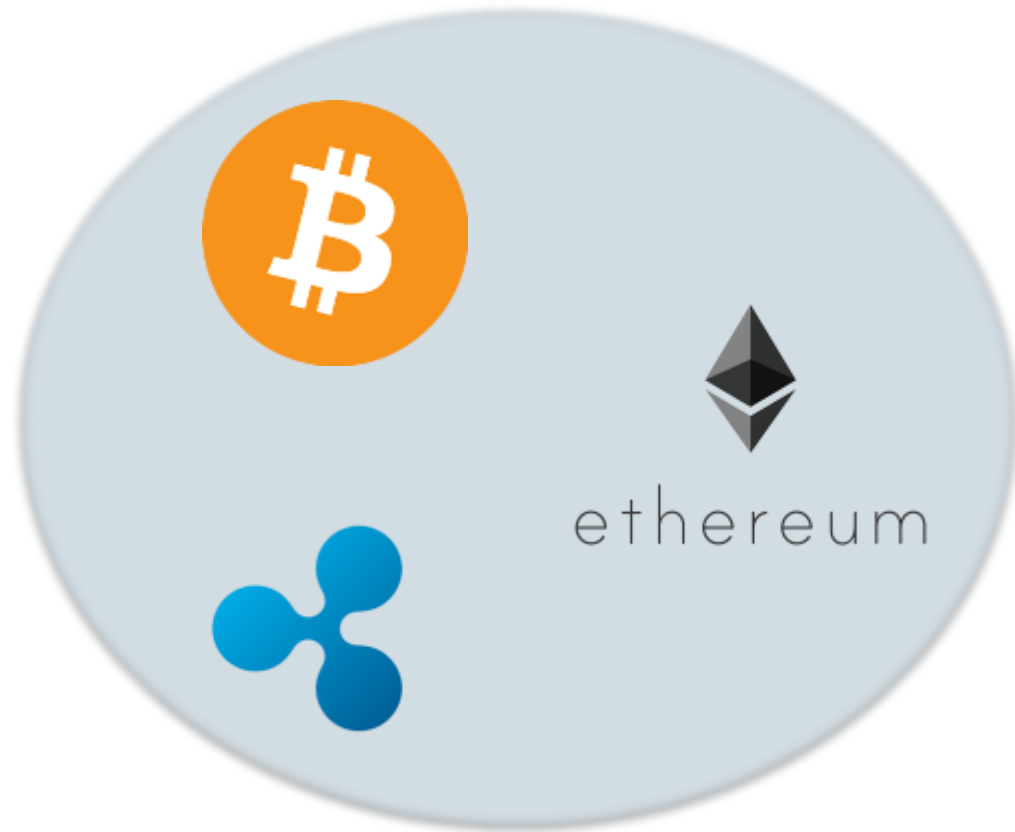
以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント（確約）するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

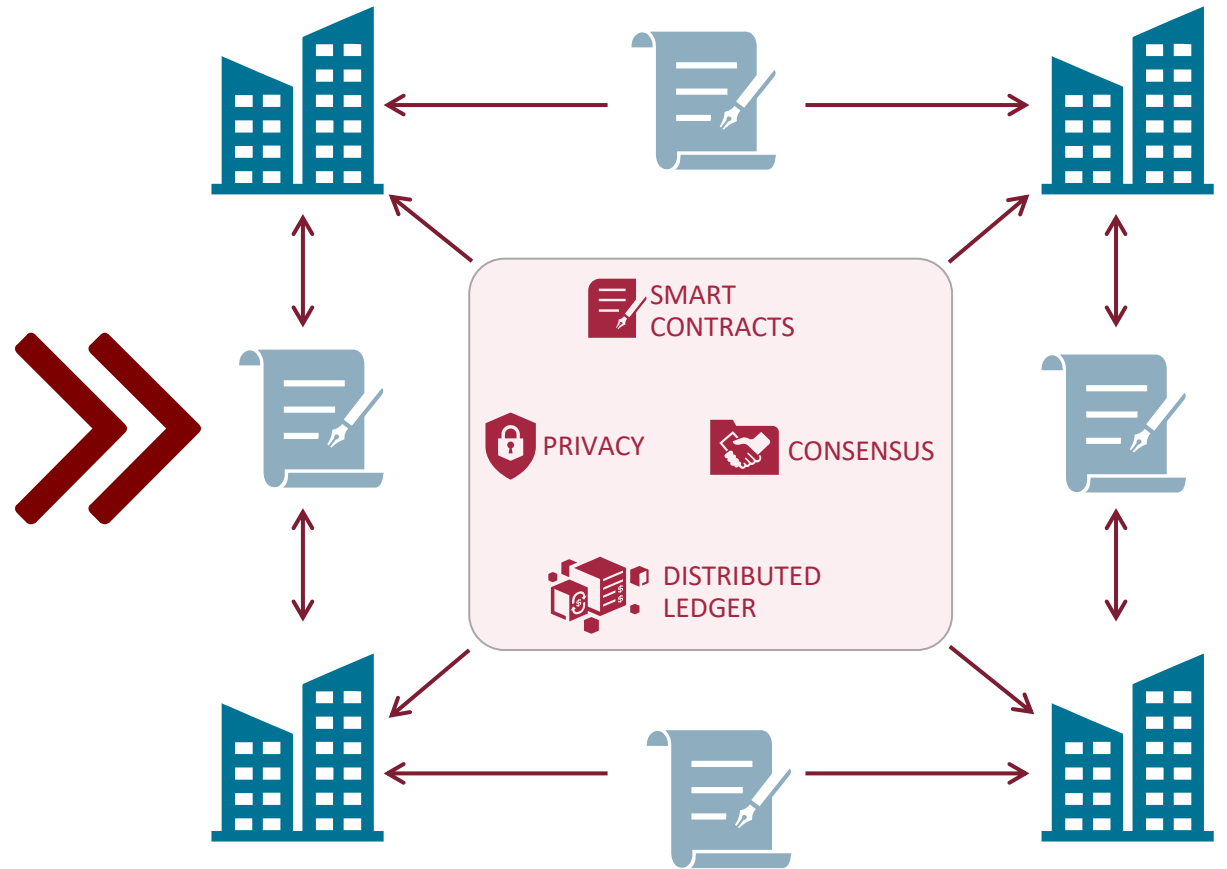
# Program Agenda

- 1 ▶ エンタープライズブロックチェーンシステム化における課題
- 2 ▶ Oracle Blockchain Platformによるソリューション
- 3 ▶ NRI様での取り組みのご紹介
- 4 ▶ 振り返り & まとめ

# ブロックチェーン技術は エンタープライズでの利用の段階へ



Wild excitement  
over Crypto Currencies .....



Enterprise Adoption !!

# エンタープライズブロックチェーン システム化における課題

**ブロックチェーン  
基盤の適切な選定**

**システム化実現への  
迅速性**

**アプリケーションと  
データの  
インテグレーション**

**独特な技術特性への  
対応**

# ブロックチェーン基盤の適切な選定

パーミッションレスorパーミッションドのカテゴリ選択



## パーミッションレス a.k.a. パブリック

- 誰でもネットワークに参加しノードを持てる  
(公開制、オープン)
- 経済的インセンティブによって不正を抑止する  
比較的低速なトランザクション処理

## パーミッションド a.k.a. プライベート

- 招待されたメンバーのみが参加しノードを保持  
(許可制、クローズド)
- 許可制でメンバーが信頼できることに依拠した  
高速なトランザクション処理



# ブロックチェーン基盤の適切な選定

パーミッションレスorパーミッションドのカテゴリ選択

## パーミッションレス a.k.a パブリック

- それぞれに異なる特長、課題、適用領域があることを理解する
- どちらが優れている／劣っている、ではなく、**自身のユースケースに合っているのはどちらか**、を考える
- 必ずしもエンタープライズ=パーミッションドではないが、現状ではパーミッションドの利用が多い

- 許可制でメンバーが信頼できることに依拠した  
**高速なトランザクション**処理

# カテゴリの中でのブロックチェーン基盤の比較選定

## 代表的なブロックチェーン／DLT基盤

	Bitcoin	Ethereum	Corda	Enterprise Ethereum	Hyperledger Fabric
用途	送金	汎用	金融機関間取引	汎用	汎用
主要ガバナンス	開発者コミュニティ	開発者コミュニティ	R3社	EEA	Linux財団
参加形態	Permission-less	Permission-less	Permissioned	Permissioned	Permissioned
NW内情報共有範囲	公開	公開	限定	限定可能	限定可能
コンセンサス	PoW	PoW	当事者間	方式選択可能	方式選択可能
ベース暗号通貨	BTC	ETH	なし	なし	なし
スマートコントラクト	なし	Solidity、 Vyper	Kotlin、Java	Solidity、 Vyper	GO、Node.js、 Java (&EVM)
備考	—	PoSへの移行予定	金融以外の分野での活用も進む	実装はQuorumなど複数あり	安定LTS版1.4系、最新機能の2.0系を両輪でサポート

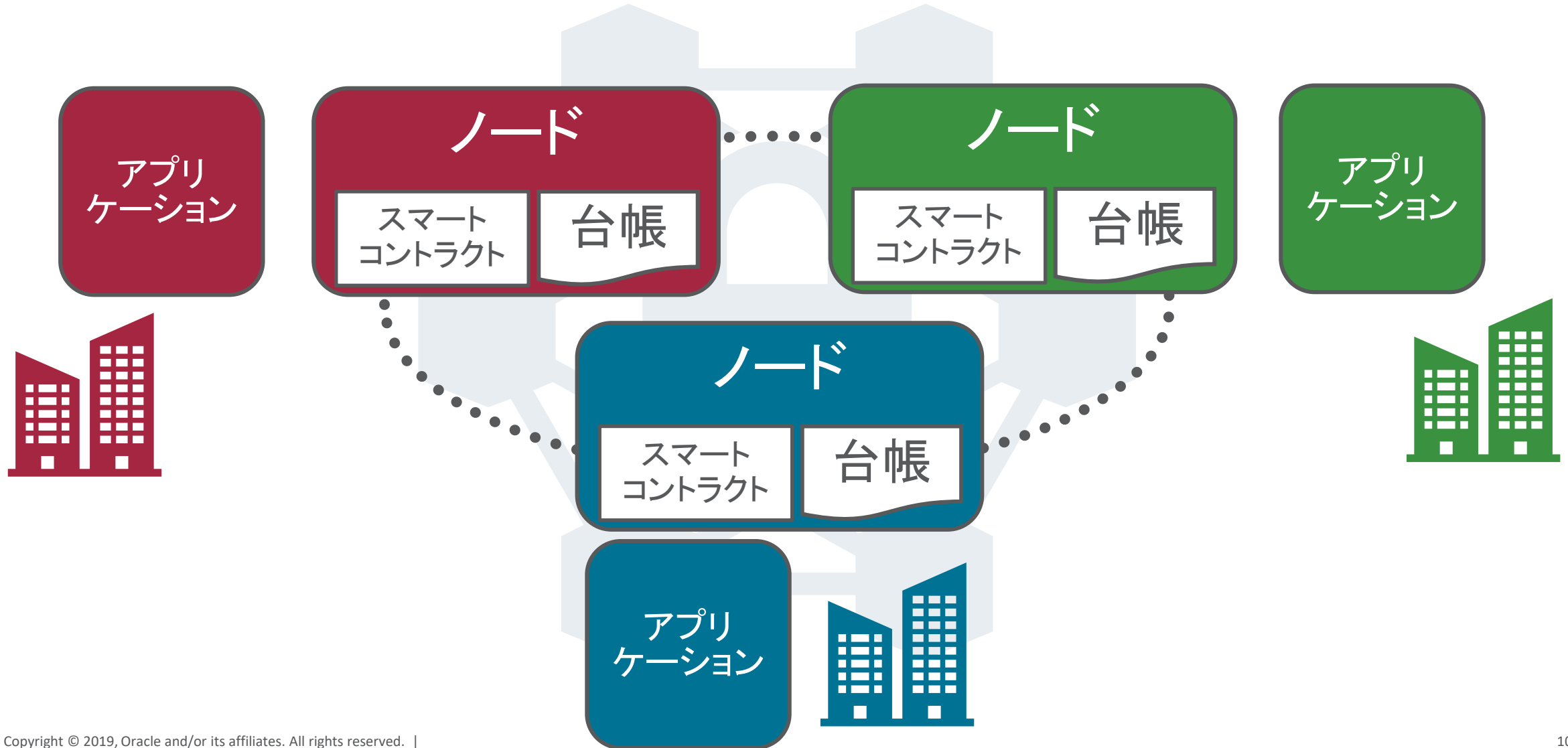


# カテゴリの中でのブロックチェーン基盤の比較選定

## 代表的なブロックチェーン／DLT基盤

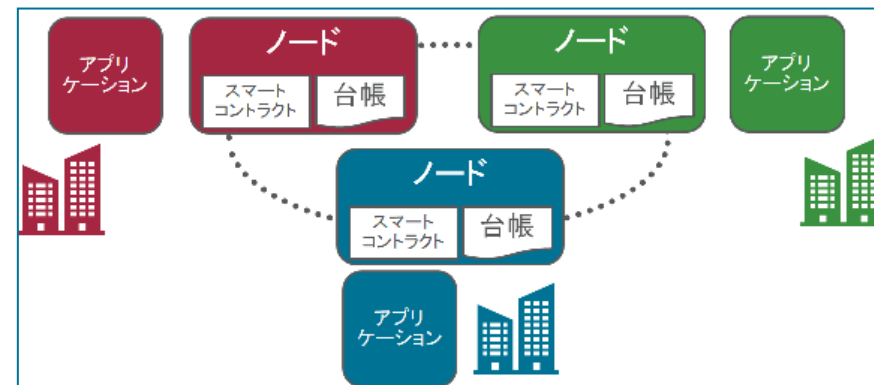
	Bitcoin	Ethereum	Corda	Enterprise	Hyperledger
<ul style="list-style-type: none"> <li>• <b>自分のユースケースに合っているものはどれか</b>を考える</li> <li>• <b>似たユースケースでの実績</b>を参考にする</li> <li>• 機能面だけを見るのではなく、総合的な判断が必要             <ul style="list-style-type: none"> <li>• 基盤側：開発体制、サポート、公開情報、実績、将来性</li> <li>• 自組織側：スキルギャップ、対応要員の確保</li> </ul> </li> </ul>					
備考	—	PoSへの移行予定	金融以外の分野での活用も進む	実装はQuorumなど複数あり	安定LTS版1.4系、最新機能の2.0系を両輪でサポート

# (コンソーシアム型の) ブロックチェーンネットワーク



# システム化実現への迅速性

学習、準備が必要な多くの事項に効率的に対応する



## ①基盤・ネットワーク構築関連

–OS、ネットワーク、VM/Docker、暗号鍵/署名…

## ②スマートコントラクト・アプリケーション開発関連

–言語、API/フレームワーク/SDK、開発ツール…

## ③システム&コンソーシアム運用・管理関連

–セキュリティ、ガバナンス、システム監視、障害対応…

# Hyperledger Fabricの学習要素

赤字：理解が必須の基礎知識  
青字：知っていると便利、応用編  
それ以外：必要に応じて学ぶ

- Organization
  - Peer
    - Endorser / Committer
    - Gossip
      - Anchor / Leader
      - Intra Org / Inter Org
- Ordering Service
  - Type: Solo / Kafka
  - Block Generation Settings
- MSP
- Ledger
  - State DB / Blockchain
- Endorsement Policy
  - Chaincode Level / Key Level
- Transaction Flow
- Channel
  - Capability Requirements
- Private Data
  - Auto Purge / Delete
- Identity Mixer
- Peer Event Service
  - Transaction/Network/  
Channel / Custom
- Service Discovery
- Chaincode Lifecycle
- Chaincode Techniques
  - Get State / Put State
  - Ranged Query / Rich Query
  - Transient Map
  - Custom Event
  - Attribute-based AC
  - Key History / Transaction History
  - Inner CC Invocation
    - Intra / Inter Channel
  - Ethereum VM (Burrow)
- Fabric SDK

# BaaSの利用

Blockchain as a Serviceを活用してプロジェクトを迅速化、効率化

## ①基盤・ネットワーク構築関連

–OS、ネットワーク、VM/Docker、暗号鍵/署名…

## ②スマートコントラクト・アプリケーション開発関連

–言語、API/フレームワーク/SDK、開発ツール…

## ③システム&コンソーシアム運用・管理関連

–セキュリティ、ガバナンス、システム監視、障害対応…

# BaaSの利用

Blockchain as a Serviceを活用してプロジェクトを迅速化、効率化

## ①基盤・ネットワーク構築関連

–OS、ネットワーク、VM/Docker、暗号鍵/署名…

## ② Blockchain as a Serviceと呼ばれる

– ブロックチェーン基盤一式をマネージド環境として提供するサービス、  
③ テンプレートを用いて構築を容易に実現するサービスが出現

– インフラ構築にかかる手間や学習時間を省力化し、  
ビジネス価値を生むアプリケーション開発、データ活用、  
コンソーシアム運営に集中

# BaaSの利用

Blockchain as a Serviceを活用してプロジェクトを迅速化、効率化

## ①基盤・ネットワーク構築関連

–OS、ネットワーク、VM/Docker、暗号鍵/署名…

## ②スマートコントラクト・アプリケーション開発関連

–言語、API/フレームワーク/SDK、開発ツール…

## ③システム&コンソーシアム運用・管理関連

–セキュリティ、ガバナンス、システム監視、障害対応…

- スマートコントラクト、アプリケーション開発に関連する知識とスキルの学習、習得はある程度まで必須
- システム運用・管理やコンソーシアム運営についても予め検討が必要で、イチから構築するのはかなりの労力が必要

① →BaaSによってはこれらについても独自拡張機能や補助ツールなどを通じてサポート（BaaSの付加価値）

## ②スマートコントラクト・アプリケーション開発関連

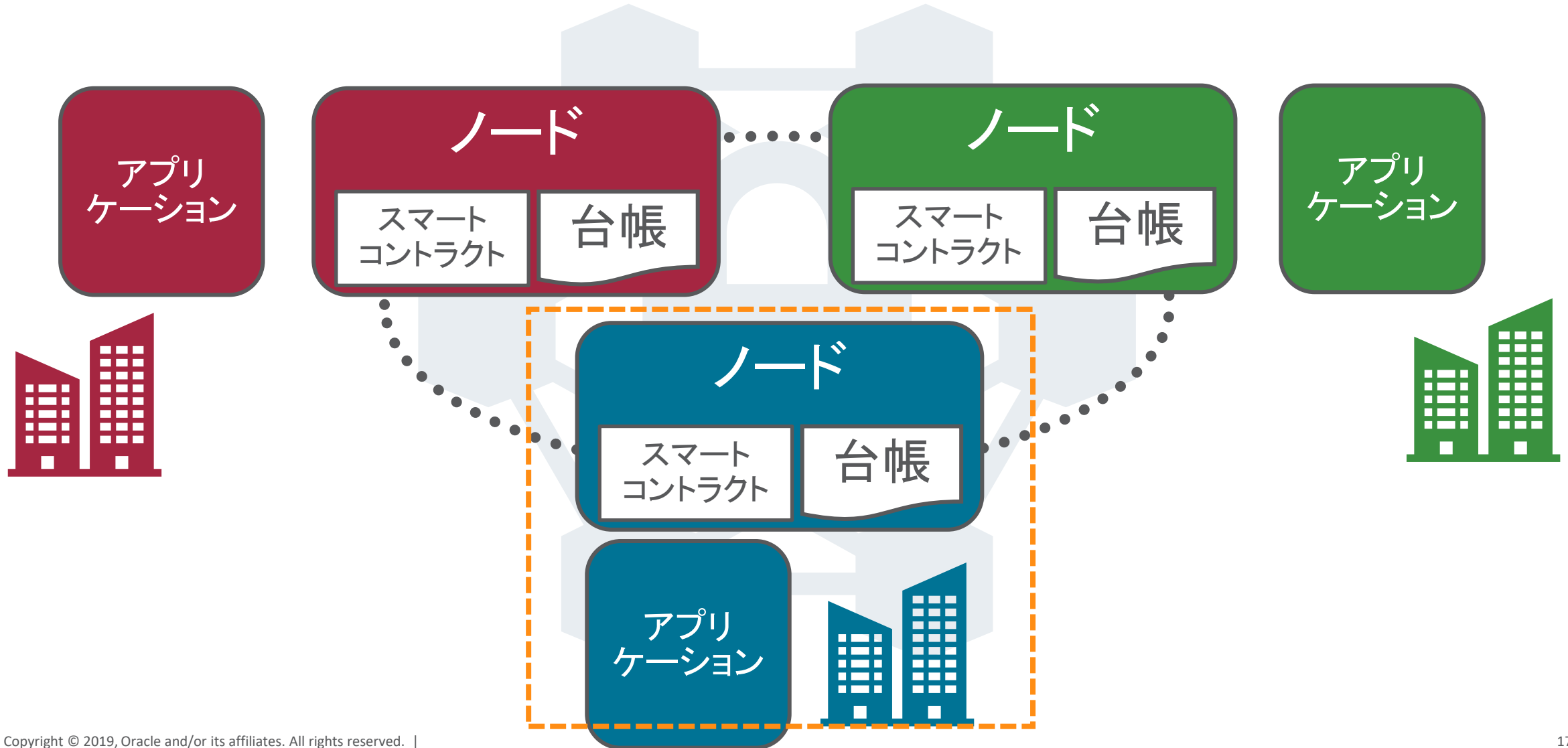
–言語、API／フレームワーク／SDK、開発ツール…

## ③システム&コンソーシアム運用・管理関連

–セキュリティ、ガバナンス、システム監視、障害対応…

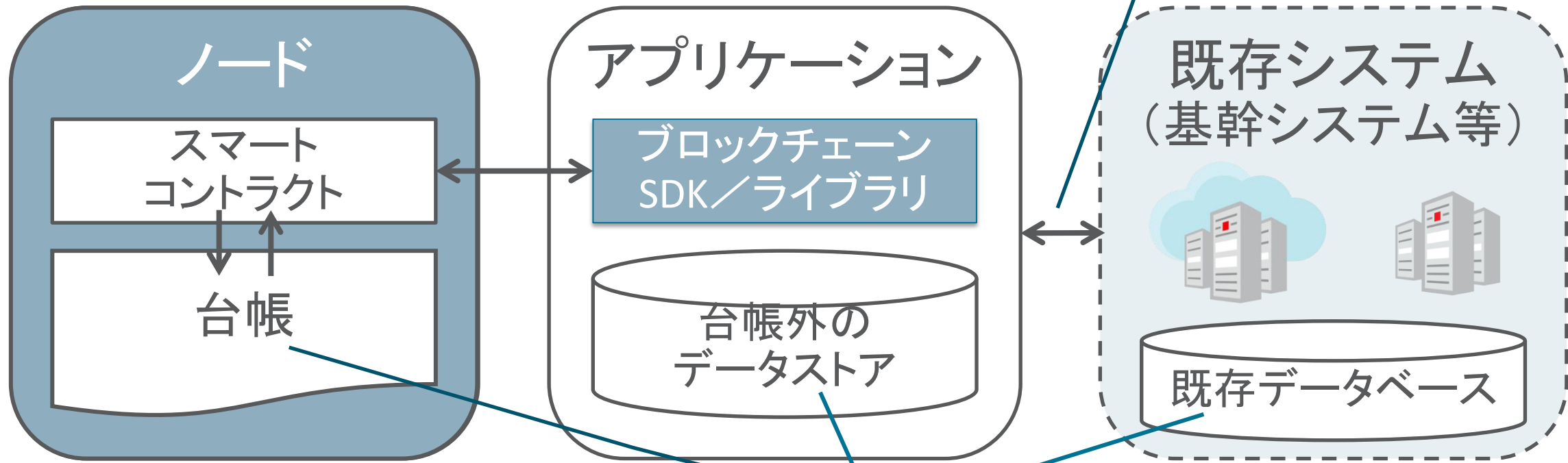


# (コンソーシアム型の) ブロックチェーンネットワーク



# 「アプリケーション」とひとくちに言っても

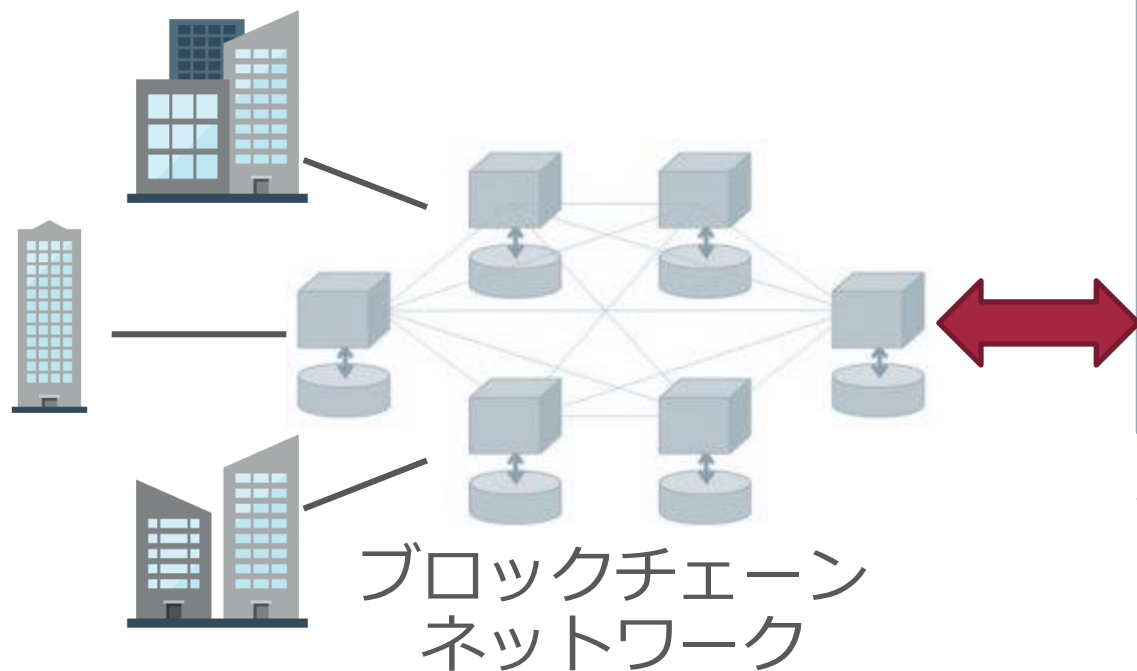
- ・連携するI/Fは？
- ・APIを作り込む？



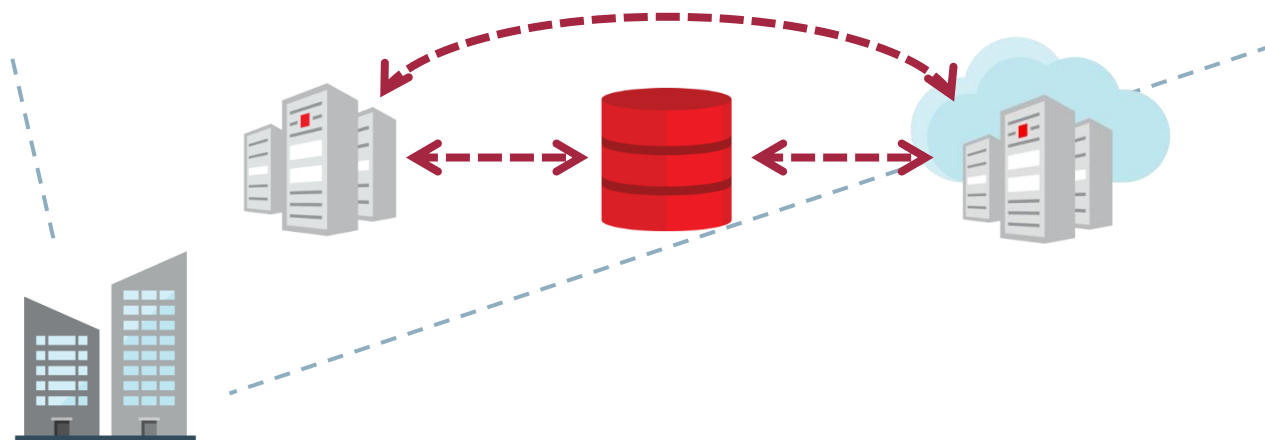
- ・データの整合性はどう保護する？
- ・バラバラなデータを統合して扱いたい

# インテグレーションを見越したシステム化検討

## 他システムとの連携をどう効率的に行うか



- ブロックチェーンアプリケーションと基幹システムなどの既存システムとの  
**アプリケーションインテグレーション**
- ブロックチェーン内外のデータ間、複数データストア間の  
**データインテグレーション**



## 独特の技術特性への対応

- 更新、クエリや集計の**将来も含めた性能を考慮**した設計
  - 運用とともに**台帳データの量**は単調増加（蓄積）していく
  - コンソーシアム**参加者、ノード、トランザクション数**は将来的な増大
  - 一般にオンチェーンでの**集計・分析**などのバッチ処理は苦手
- **データベースとは異なるトランザクションモデル、特性**
  - 分散台帳は一般にACID特性ではなく**BASE特性**
  - トランザクションロールバック不能→**2PC／3PC利用不能**
  - それぞれの**ブロックチェーン基盤**ごとの不得手、制約に留意
    - ファイナリティ、非一決定論的挙動、Dirty Read／Phantom Read、などなど



# Oracle Blockchain Platform Cloud Service



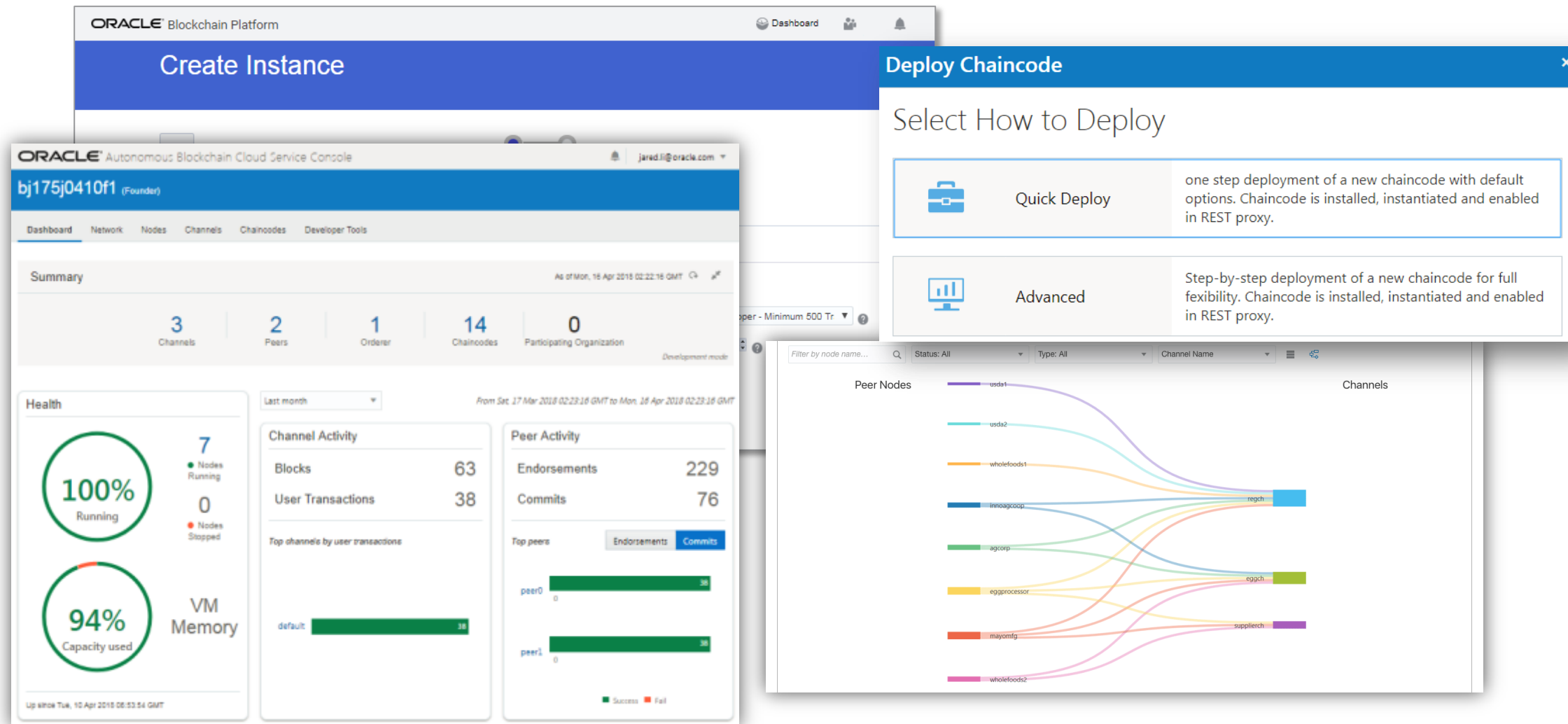
**Hyperledger Fabric**をベースにエンタープライズ利用向けPaaSとして提供

- **数ステップで構築**完了、**GUIコンソール**で管理・運用も容易
- **エンタープライズグレード**の耐障害制、堅牢性
- Oracle以外のHyperledger Fabricとも**オープンに連携**
- **Oracle独自の付加価値：**
  - 多機能な**REST API**：スマートコントラクトの利用を容易に
  - **台帳のデータをRDBにレプリケーション**：大量照会、分析、データ統合
  - **State DBとしてBerkeley DBを利用**：パフォーマンスとクエリ利便性向上、Hyperledger Fabricの**Phantom Read問題**に係る制約も解消

# Hyperledger Fabric

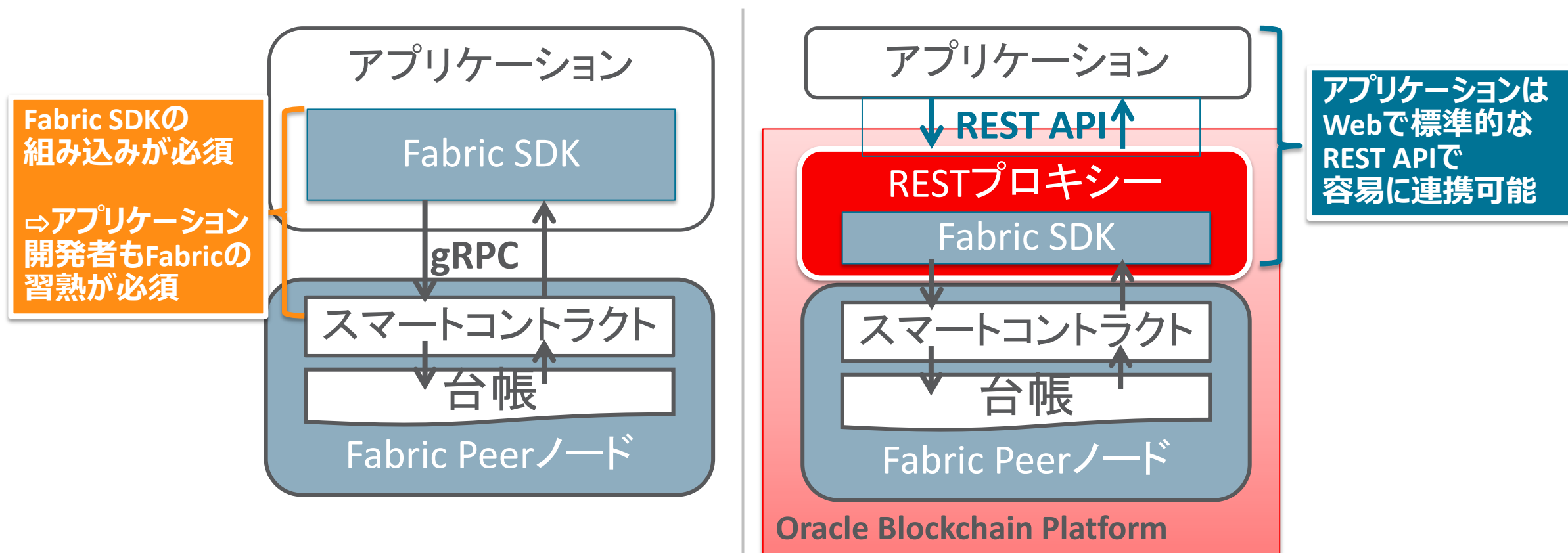
- **Hyperledger** :  
Linux財団がホストするオープンソースコミュニティで開発
- **Fabric** :
  - エンタープライズ用途の**パーミッションドブロックチェーン**
  - 金融からサプライチェーンまで**汎用的なユースケースをサポート**
  - **機密性/プライバシー**を確保したデータ共有機能
  - **スケーラブル**、プラガブルな設計
  - エンタープライズ領域での**本番利用の実績多数**

# GUIコンソールで構築・運用・管理を迅速、便利に



# RESTプロキシでアプリケーション開発を容易に

- 付属のRESTプロキシを利用することでREST APIでのスマートコントラクト実行が可能
- 通常必須となるFabric SDKへの習熟およびアプリケーションへの組み込みが不要に





# リッチヒストリーデータベースでデータ活用を容易に

- 台帳のデータをブロックチェーン外部のリレーショナルデータベースに複製



- 単純な参照処理、大規模検索処理（集計、分析）はデータベース側！
  - ブロックチェーンはそもそも集計、分析などのバッチ処理が苦手
  - 標準的なデータベースのスキルを活用して容易にデータを活用可能
  - 数多く存在するBIツールの利用もできる

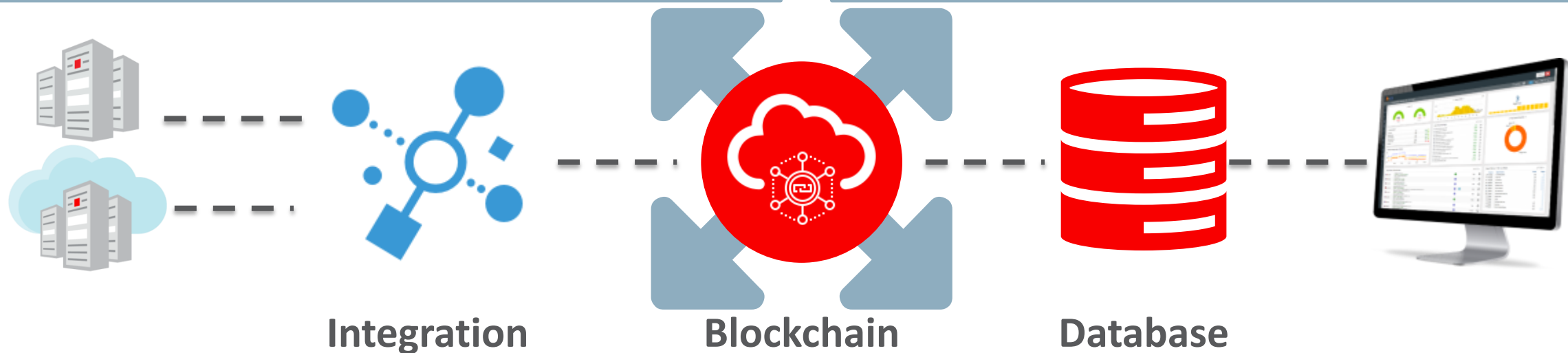
# 他システムとの連携を容易に

## Application Integration

- **REST API**が利用可能なため、Oracle Integration Cloudなどの**iPaaS製品**を利用し**ERP、SCMなど様々なシステムと容易に連携**

## Data Integration

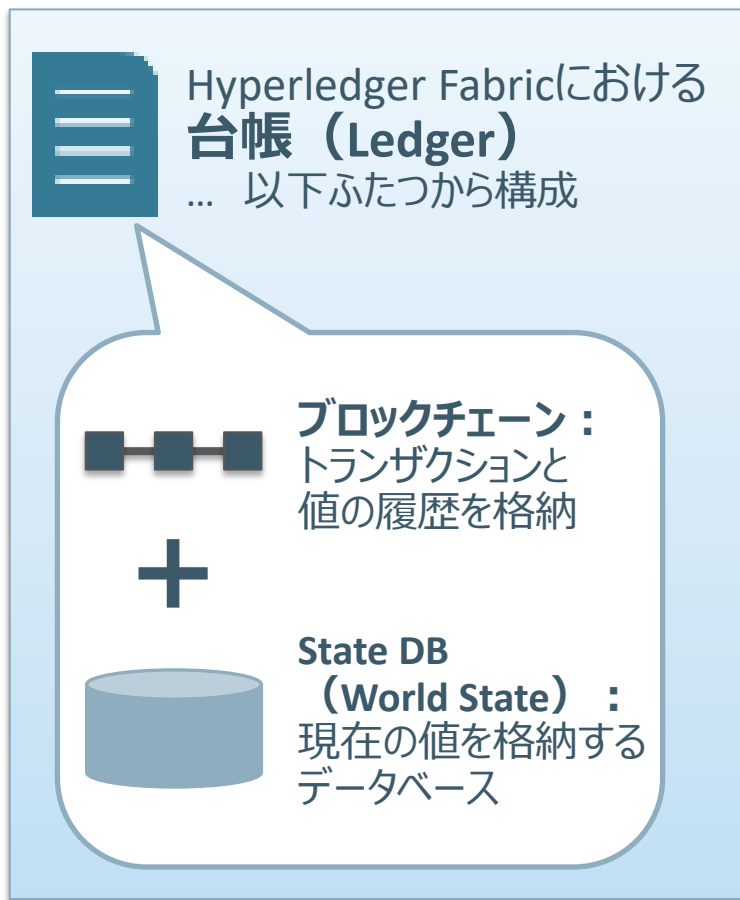
- **リッチヒストリーデータベース**により複製されたリレーショナル・データベース上のデータは他システムのデータとの統合も容易



# State DBとしてBerkeley DBを採用

※標準のHLFではLevelDB or CouchDB

- **パフォーマンスの向上**
  - スマートコントラクトの同時実行性の強化
- **SQLリッチクエリ**のサポート
  - JSONの属性を指定してのSQL-Likeクエリが可能
  - CouchDB形式のJSONクエリへの互換性も保持
- 技術的な制約の解消
  - Fabricにおける重大な制約につながっているCouchDB & リッチクエリ使用時の**Phantom Read検知不能問題を解消**





# Oracle Blockchain Platform cloud

Hyperledger Fabricをベースにエンタープライズ利用向け

- 数ステップで構築完了、**GUIコンソール**で管理・運用も容易
- **エンタープライズグレード**の耐障害制、堅牢性
- Oracle以外のHyperledger Fabricとも**オープンに連携**
- **Oracle独自の付加価値：**
  - 多機能な**REST API**：スマートコントラクトの利用を容易に
  - **台帳を外部のRDBにレプリケーション**：大量照会、分析、データ統合
  - **State DBとしてBerkeley DBを利用**：パフォーマンスとクエリ利便性向上、Hyperledger FabricのPhantom Read問題に係る制約も解消

Now  
Available  
in TOKYO



こんな時、かけこむ会社が増えています。



ビジネスプロセスを  
改善したい!



今のシステムは  
使いにくい!



システムコストを  
下げたい!



パフォーマンスを  
良くしたい!



経営分析を  
したいのだが...



どんなソリューションが  
あるの?



見積りはどれくらい  
なんだろう?



楽に管理を  
したい!

**Oracle Digitalは、オラクル製品の導入をご検討いただく際の総合窓口。  
電話とインターネットによるダイレクトなコミュニケーションで、どんなお問い合わせにもすばやく対応します。  
もちろん、無償。どんなことでも、ご相談ください。**



**お問い合わせは電話またはWebフォーム**

**☎ 0120-155-096**

受付時間 月～金 9:00-12:00 / 13:00-17:00  
(祝日および年末年始休業日を除きます)

**<http://www.oracle.com/jp/contact-us>**

ORACLE®