ブロックチェーンを使用したサプライチェーン管理システムの試作及びプライバシー保護の動向

大場 義洋
キオクシア株式会社
メモリ技術研究所 システム技術研究開発センター
2020年9月25日

KIOXIA

アウトライン

第一部：ブロックチェーンを使用したサプライチェーン管理システムの試作

- ユースケース
- アーキテクチャ
- 試作システム実装
- 性能評価

第二部：プライバシー保護と非中央集権型識別子

- ブロックチェーンとプライバシー保護
- プライバシー保護関連法律
- DID (Decentralized Identifiers)
- プライバシー保護自動化における課題

ブロックチェーンを使用したサプライチェーン管理システムの試作

Reference: Zi Ee Lee, Raphael Liang Hui Chua, Sye Loong Keoh, Yoshihiro Ohba: Performance Evaluation of Big Data Processing at the Edge for IoT-Blockchain Applications. IEEE GLOBECOM 2019 [1]

背景

- IoTシステムは、対象物の制御・監視にセンサーやモバイル端末を使用

- 細粒度データ収集により、精度の高い制御、結果予測、異常検知が可能となる

＜課題＞ データ量増大（処理効率性が必要）、データ完全性（改ざん防止）

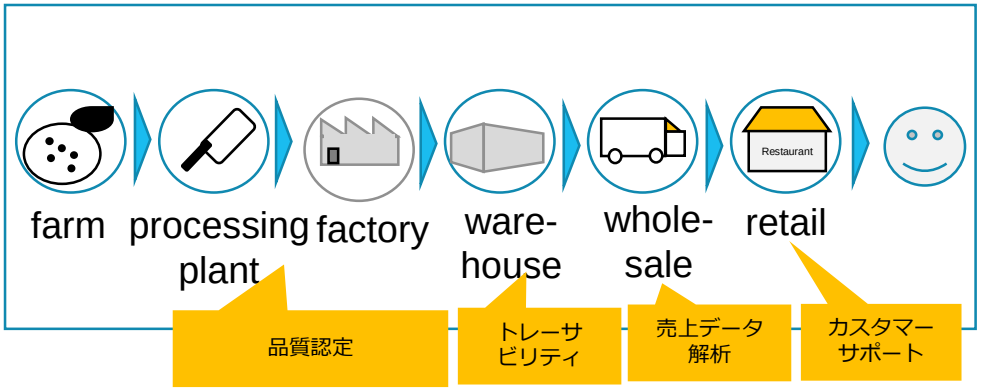
ターゲットユースケース： 食品サプライチェーン管理

- 目的：流通過程全体にわたる食品の安全性とトレーサビリティを提供

  例：ハラル認証：「農場からフォークまで」

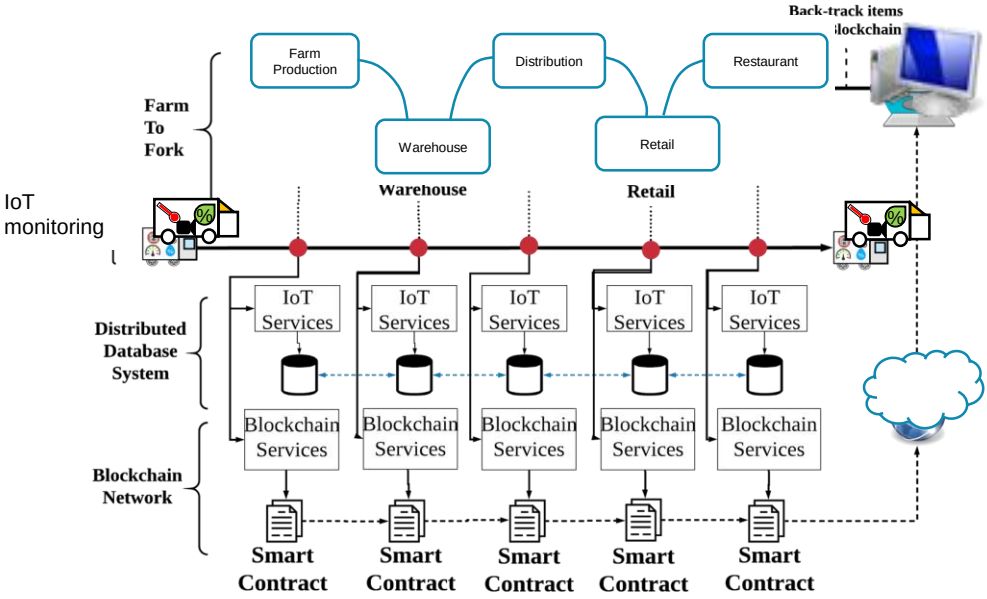
- プレイヤー：生産者、卸業者、運送業者、小売店、消費者
- サプライチェーンのエンドツーエンドでIoTセンサーによる食品状態の常時監視

  → 商品すり替え、食品管理条件違反等を検知



アーキテクチャ

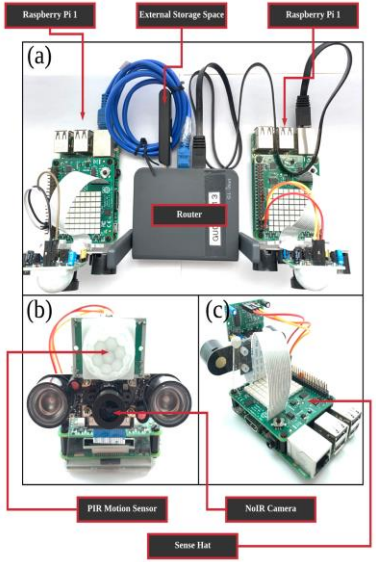
- IoTモニタリングサービス
  - 各流通拠点で商品状態監視
- IoTサービス
  - センサーデータ収集
- ブロックチェーン(BC)サービス
  - 流通トランザクション（センサーデータを含む）をスマートコントラクトにより実行し、分散台帳に記録
- エッジコンピューティングノード:
  - 流通各拠点での{IoT,BC}サービス提供
- ストレージ
  - {IoT,BC}サービスのデータを保存



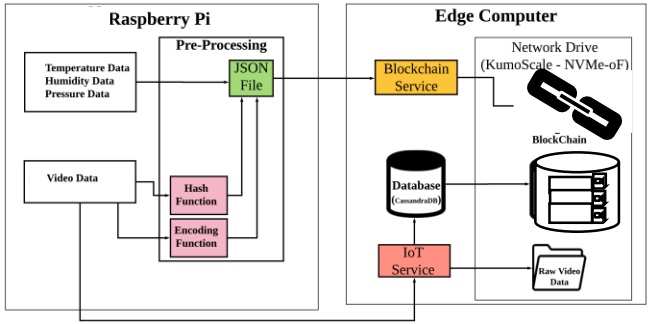
(Fig. 1 [1]より, ©2019 IEEE)

試作システム実装: IoTモニタリングサービス

- IoTモニタリングデバイス：Raspberry Pi
  - 温度センサー、湿度センサー、圧力センサー、赤外線センサー、GPS、ビデオカメラ
- IoTモニタリングサービス
  - エッジノードにセンサーデータをW-iFi®で送信
  - 定期的 and/or 配送車到着時
- ２台のモニタリングデバイスにより冗長化



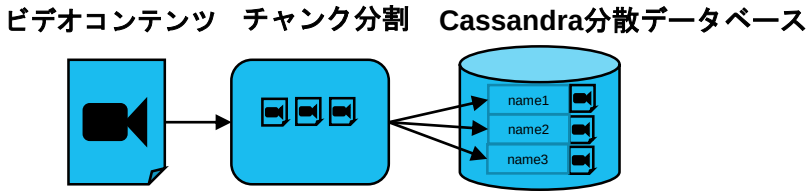
(Fig. 3 [1]より, ©2019 IEEE)



(Fig. 2 [1]より, ©2019 IEEE)

試作システム実装: IoTサービス

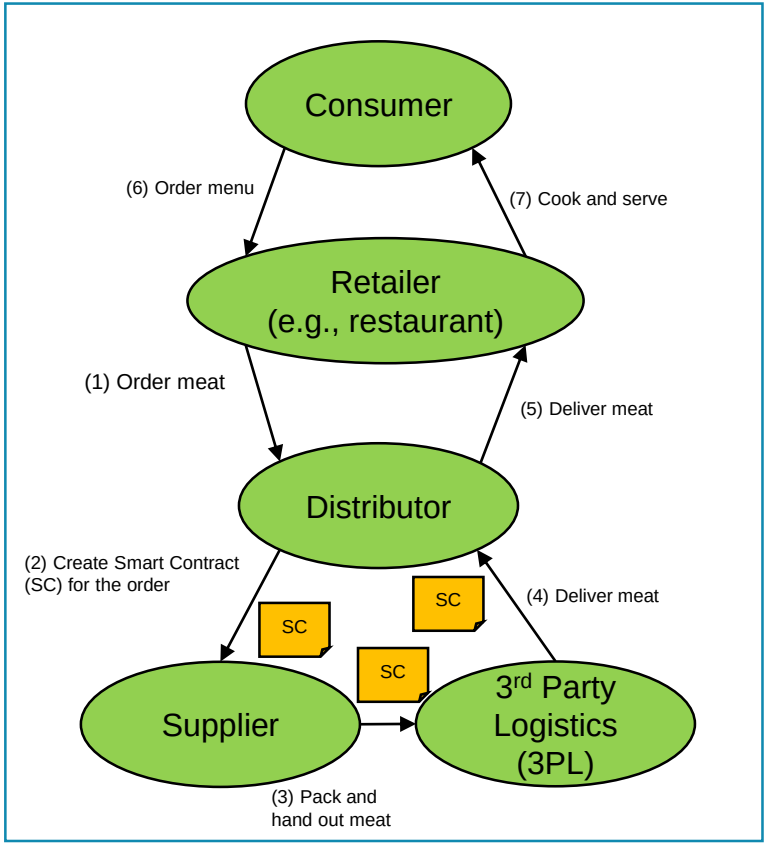
- ビデオコンテンツのメタデータとその他のセンサーデータは**オンチェーンデータ**として保持
  - ブロックチェーンによるレプリケーション
- ビデオコンテンツはチャンクに分割して**オフチェーンデータ**として保持
  - Apache Cassandra® 分散DBによるレプリケーション



試作システム実装：ブロックチェーンサービス

- ブロックチェーンプラットフォーム：
  Hyperledger® Sawtooth
  - スマートコントラクト: **Seth** transaction family

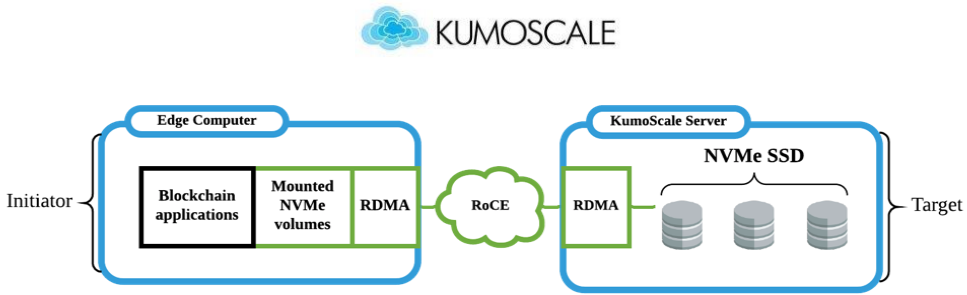
- 軽量コンセンサスアルゴリズムProof-of-Elapsed-Time (PoET) を使用
  - ***PoET :*** **各ノードがトラステッドプラットフォームIntel® SGX (Software Guard Extensions)上でランダム時間待った後に応答し、最も早く応答したノードが生成したブロックを選択**



試作システム実装：ネットワークストレージ(*KUMOSCALE™ for NVMe-oF*)

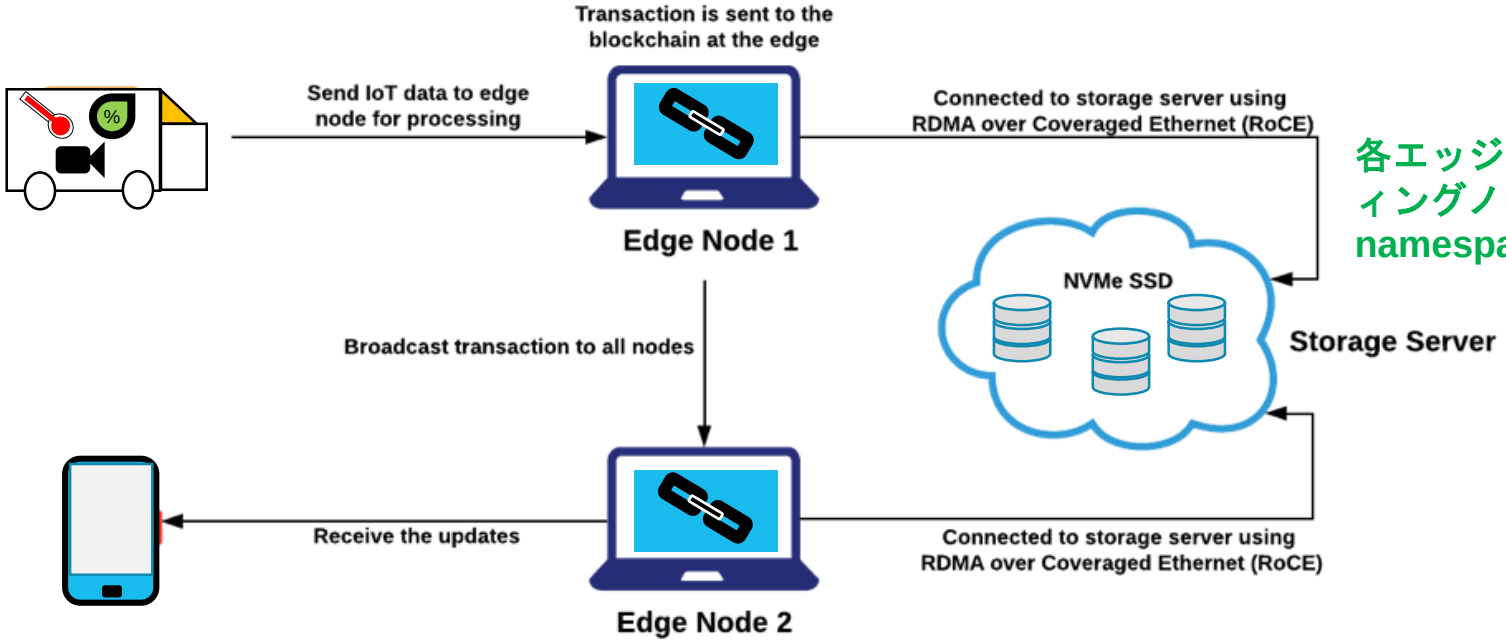
- NVMe-oF ™ (NVMe™ over fabrics) ：複数のコンピューティングノードがNVMeストレージに高速ネットワーク(e.g., 100GbE)上でリモート接続するための通信プロトコル
  - NVMeの**バーストI/O**機能により、低遅延＆高スループットI/Oが可能

- NMVe-oF管理ソフトウェア: KUMOSCALE™
  - **Namespace**による論理ドライブ割り当てにより、複数のエッジノードに対し動的かつ柔軟なストレージ領域の割当て



RDMA: Remote Direct Memory Access
RoCE: RDMA over Converged Ethernet

性能評価システムの構成



各エッジコンピューティングノードに個別のnamespaceを割当てる

想定シナリオ：同一物流拠点内の建屋毎にエッジサーバを配置

性能評価

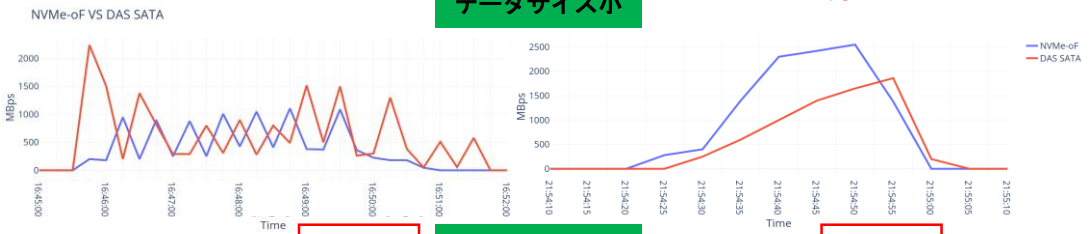
1. 2種類のI/Oインターフェースについて、I/O性能を比較
   - NVMe-oF (KUMOSCALE使用)
   - 直接接続SATA
2. ブロックチェーントランザクション実行時のCPU性能

- オフチェーンデータ(ビデオデータ)
- オンチェーンデータ:
  - センサーデータ(温度、湿度、圧力、時刻、現在位置、etc.)
  - ビデオデータのID及びハッシュ値
- エッジノード: CPU: Intel® Xeon® Gold 5115 @ 2.40GHz

評価結果(I/O性能)

青：NVMe-oF
赤：SATA

データサイズ小



(a) DiskIO of Cassandra Write (5.39GB)　(b) DiskIO of Cassandra Read (5.39GB)

データサイズ大

(c) DiskIO of Direct Access Storage Write (26.98GB)　(d) DiskIO of Direct Access Storage Write (26.98GB)

データサイズ小

TABLE I: Performance Comparison of CassandraDB Write/Read – 5.39GB

Storage	Max Write	Max Read	Write Mean	Read Mean
NVMeOF	1.109 GBps	2.587 GBps	328 MBps	600 MBps
SATA	2.242 GBps	1.861 GBps	452 MBps	207 MBps

データサイズ大

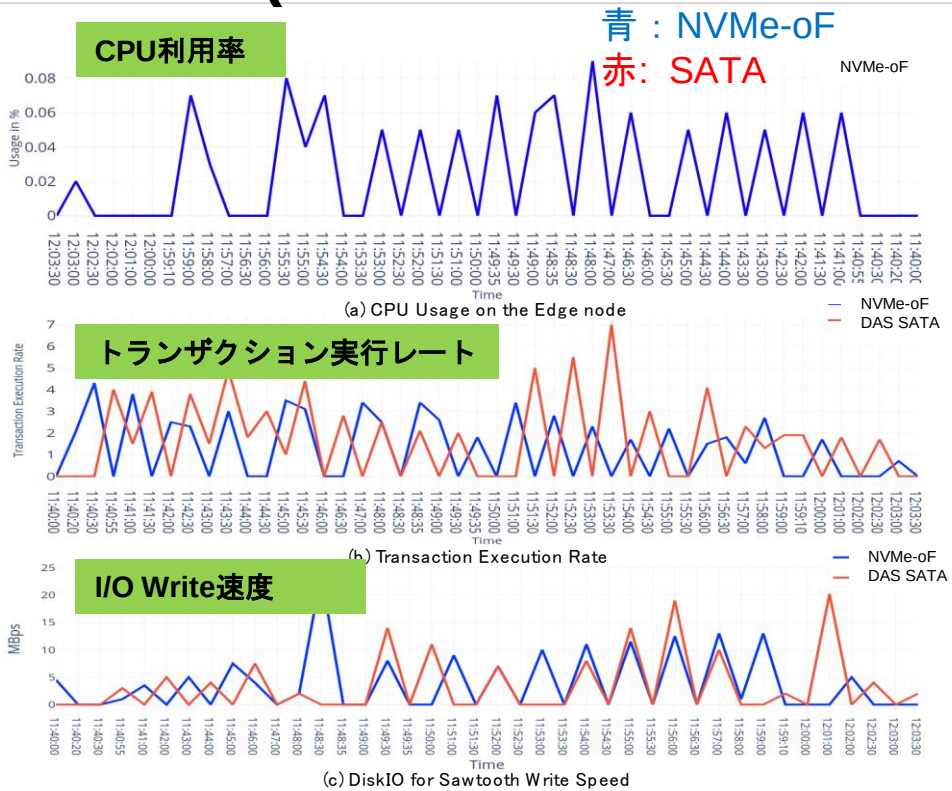
TABLE II: Performance Comparison of Direct Write/Read – 26.98GB

Storage	Max Write	Max Read	Write Mean	Read Mean
NVMeOF	8.596 GBps	6.715 GBps	2.080 GBps	2.076 GBps
SATA	4.250 GBps	4.062 GBps	1.500 GBps	1.499 GBps

(Fig. 5, Table 1, Table 2 [1]より, ©2019 IEEE)

- データサイズが小さい場合（1MB)、NVMe-oFとSATAは同等の性能
  - 理由：NVMe-oFのネットワーク通信オーバーヘッド
- データサイズが大きい場合(26.98GB)、NVMe-oFの性能はSATAを上回る
  - NVMe-oFの最大書込みスループットはSATAの2倍

評価結果(ブロックチェーントランザクション実行性能)



(Fig. 6 [1]より, ©2019 IEEE)

- エッジノードでのCPU利用率は10%以下
  - クラウドモデルの場合、
    - タスク集中実行のためCPU利用率が上昇
    - ネットワーク遅延も増大
    - → トランザクション実行レートが低下すると予想

- NVMe-oF使用時の平均トランザクション実行レートはSATA使用時を上回る(1.26 tr./s vs 1 tr./s)

- Write速度はほぼ同じ(ave. 2MBps)

第一部まとめ

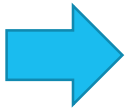
- NVMe-oFをストレージに用いたIoTブロックチェーンシステムを試作
- サプライチェーンユースケースにおいて、 NVMe-oFは高いI/O性能を示す
- エッジコンピューティングモデルによるIoTブロックチェーンシステムは、クラウドモデルに対して優位になりうる

プライバシー保護と非中央集権型識別子

ブロックチェーンとプライバシー保護ー
個人情報保護自動化のユースケースと重要個人情報

(自動化：スマートコントラクト自動実行と連動)

- 企業ユース（サプライチェーン管理も含む）
  - 顧客情報
  - 従業員情報
  - リクルート対象学生の個人情報
- 教育ユース
  - 学生の個人情報
- 医療ユース
  - 患者の個人情報
- 行政ユース
  - 住民の個人情報



ブロックチェーンを個人情報保護自動化用途に使用する場合、ブロックチェーン台帳に個人情報が含まれ得るため、以下を含む対策が必要

- 個人データの秘匿化
- 個人データの匿名化 ← 本パートの主題

個人情報保護に関する法律

<欧州> GDPR (General Data Protection Regulation, EU一般データ保護規則)

- https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.119.01.0001.01.ENG&toc=OJ:L:2016:119:TOC

<日本> 個人情報保護法（基本理念）＋ セクター毎（民間、国、独法、自治体）の法律に細分化

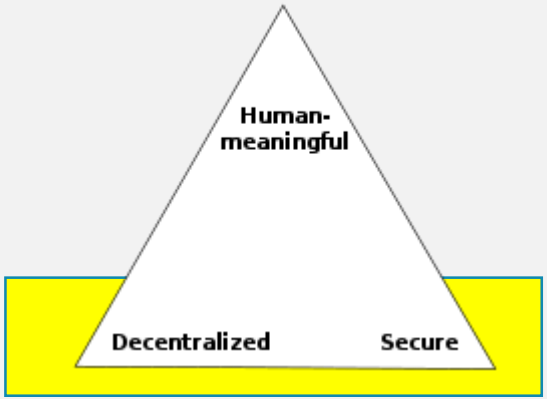
→「個人情報保護法制２０００個問題」(https://www8.cao.go.jp/kisei-kaikaku/suishin/meeting/wg/toushi/20161115/161115toushi01.pdf)

全般的に、個人情報は保護される方向にある

- データ収集・利用目的に関する本人同意取得に関する義務
  - 日本では第三者への情報提供時の本人同意はオプトアウト可
- データ消去に関する義務（忘れられる権利）
  - GDPRでは本人が求めれば直ちに消去可能
- データ漏洩時の通知に関する義務
  - GDPRでは72時間以内に監督機関に報告義務

DID (Decentralized Identifiers:非中央集権型識別子)

- DID: 非中央集権型のデジタルID
- W3C®（World Wide Web Consortium）にて定義
- DIDのユースケース及び要件：https://www.w3.org/TR/did-use-cases/
- DIDの仕様: https://w3c.github.io/did-core/
  - 以下を定義
    - A) DIDの一般的なシンタックス
    - B) DIDに紐づいたメタデータ(DIDドキュメント)に対するCRUD操作に必要な一般要件
      - CRUD: Create, Read, Update, Deactivate
- DIDのシンタックス： `did`:method:method-specific-id
  - `did`: DIDに対するURLスキーム識別子
  - method: DIDメソッドの識別子
  - did-method-specific-id: DIDメソッド固有識別子
- DIDの例: “did:ethr:0xb9c5714089478a327f09197987f16f9e5d936e8a”
- 現在の使用例：各種仮想通貨、gitのコミット署名, etc.



Zooko's Triangle
(https://en.wikipedia.org/wiki/Zooko%27s_triangle
を元に著者が黄色枠を付加)

DID導入の利点

- 企業ユース：サプライチェーン管理
  - 背景：現在、納税者（利用者）識別番号、取引主体識別子、企業・事業所識別コードなどいろいろなIDがあるが、どれも対応する発行団体にIDを割り当ててもらう必要があり、自身でIDを割り当てることができない
  - DID導入の利点：DIDにより、政府のID管理に要する負担が軽くなり、デジタル署名にも使用できるようになる

- 教育ユース：在籍・卒業・成績証明
  - 背景：従来のIDでは公開鍵を更新する手間がかかる（認証局とのやりとりが必要）
  - DID導入の利点：公開鍵を更新する際の手間が最小化できる（認証局不要）

- 医療ユース：プライバシーに配慮した診断と処方
  - 背景：人に知られたくない症状でオンライン診断を受け、オンラインで処方箋をもらいたい場合がある
  - DID導入の利点：匿名で受診、処方が可能

- 行政ユース：自治体単位のID管理
  - 背景：個人情報保護法制２０００個問題 (自治体ごとに個人データの種類も扱いも異なりうる)
  - DID導入の利点：ID管理の自由度増大

DIDの４要件

1. Decentralized	• 名前を集中管理するエンティティを要しないこと
2. Persistent	• 名前を維持するのに継続的なオペレーションを要しないこと
3. Cryptographically Verifiable	• 名前に対する制御の正当性を暗号的に証明できること
4. Resolvable	• 名前からそのメタデータを発見できること

DIDの３エンティティ

- **サブジェクト**：DIDにより参照されるエンティティ
  - 人、モノ、組織、etc. なんでもよい。

- **コントローラ**: DIDを生成、制御するエンティティ
  - サブジェクトとコントローラが同一エンティティである場合もそうでない場合もある

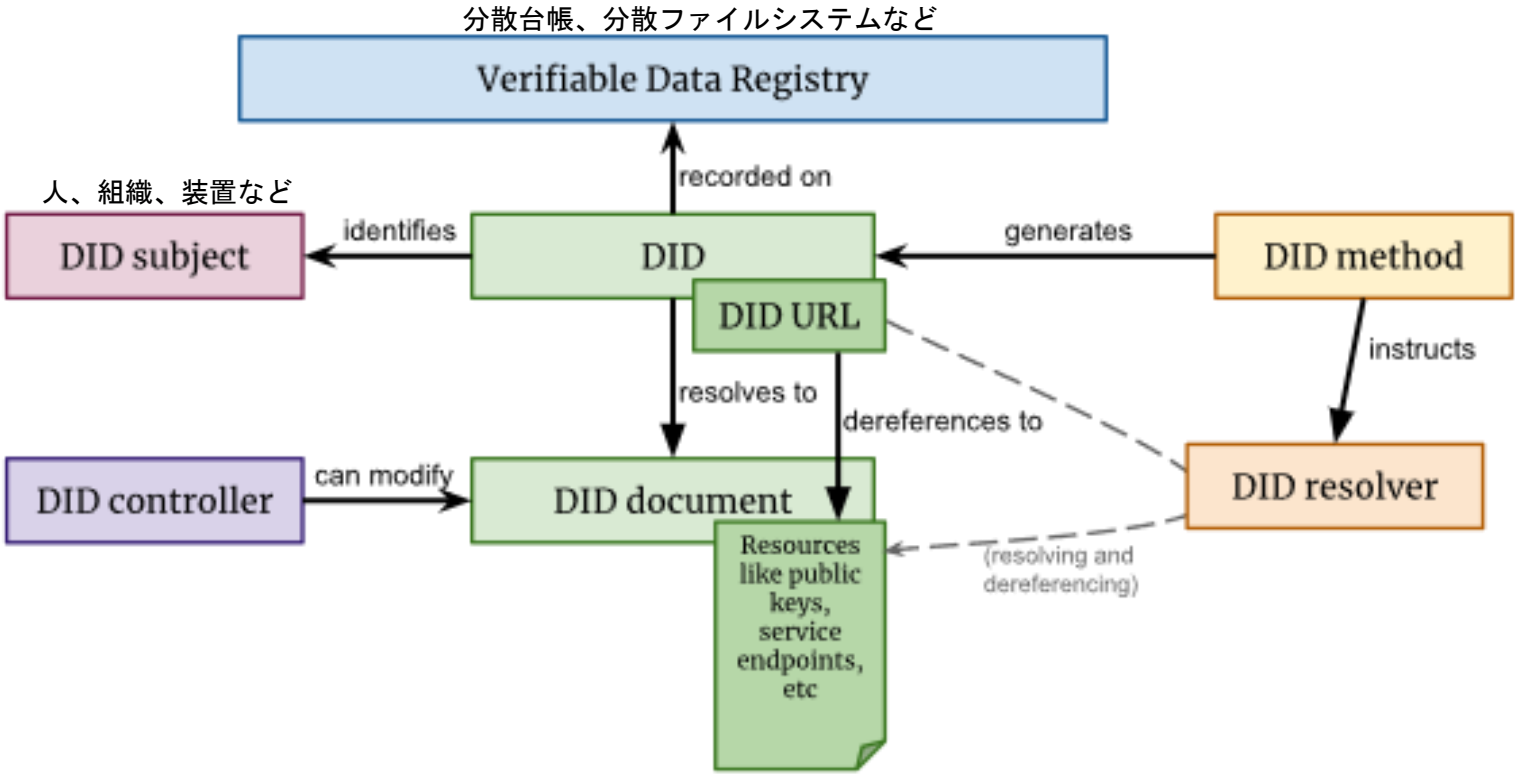
- **要求パーティー**：DIDを使用してサブジェクトに関係する何らかのやりとりを行うエンティティ

DIDには「Identity Provider」がない

用語

- DIDドキュメント：DIDに関するメタデータを記述したドキュメント
- DID解決：DID（と解決前メタデータ)を入力し、DIDドキュメント(と解決済メタデータ)を出力する機能
- DIDリゾルバ: DID解決により DIDからDIDドキュメントを出力するソフトウェア
- DIDメソッド：DID及びDIDドキュメントの生成/検証/更新/無効化方法に関する仕様
  - 現在定義されているDIDメソッド: https://www.w3.org/TR/2020/NOTE-did-spec-registries-20200618/#did-methods
- 検証可能データレジストリ: DID及びDIDドキュメントの生成/検証/更新/無効化を補助するシステム
  - 例えば、ブロックチェーン台帳を検証可能データレジストリとして使用可能

DIDアーキテクチャ



© 2020 W3C <http://www.w3.org/TR/2020/WD-did-core-20200907/>
(上記URLのFigure 1に著者が日本語補足を付加)

DIDドキュメントの例(JSONフォーマット)


```
{
 "@context": ["https://www.w3.org/ns/did/v1", "https://w3id.org/security/v1"],
 "id": "did:example:123456789abcdefghi",
 ...
 "verificationMethod": [{
  "id": "did:example:123#_Qq0UL2Fq651Q0Fjd6TvnYE-faHiOpRlPVQcY_-tA4A",
  "type": "JwsVerificationKey2020",
  "controller": "did:example:123",
  "publicKeyJwk": {
   "crv": "Ed25519",
   "x": "VCpo2LMLhn6iWku8MKvSLg2ZAoC-nlOyPVQaO3FxVeQ",
   "kty": "OKP",
   "kid": "_Qq0UL2Fq651Q0Fjd6TvnYE-faHiOpRlPVQcY_-tA4A"
  }
 },
（右列に続く）
```

```
{
  "id": "did:example:123456789abcdefghi#keys-1",
  "type": "Ed25519VerificationKey2018",
  "controller": "did:example:pqrstuvwxyz0987654321",
  "publicKeyBase58":
"H3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
 }, {
  "id": "did:example:123456789abcdefghi#keys-2",
  "type": "Secp256k1VerificationKey2018",
  "controller": "did:example:123456789abcdefghi",
  "publicKeyHex":
"02b97c30de767f084ce3080168ee293053ba33b235d7116a3263d29f1450936b71"
 }],
 ...
}}
```


 http://www.w3.org/TR/2020/WD-did-core-20200907/
(上記URLのEXAMPLE 13を著者が2列に分割、赤色付加)

個人情報保護ユースにブロックチェーン及びDIDを使用する場合の課題

オンチェーンデータに含まれる個人データの匿名化・秘匿化

- 個人データがブロックチェーン台帳に含まれる場合、匿名化もしくは暗号化が必要
- オンチェーンデータの匿名化や暗号化のための標準APIが必要

個人データ利用ポリシーの自動制御

- 個人データを第三者に自動提供する場合、データ利用ポリシーの記述言語とその処理系が必要
- 例：ポリシー自動制御用オープンソースポリシーツール: **Open Policy Agent** (https://www.openpolicyagent.org/)
  - ポリシー記述言語：Rego (Prolog系言語)
  - コンテナアクセス、HTTPアクセス、SSHログインのポリシー制御に使用される
  - 個人データの指定方法等が課題

第二部まとめ

- 個人情報保護を自動化する場合に有用な技術として、DIDを紹介

- 個人情報保護ユースにブロックチェーン及びDIDを使用する場合の２つの課題を抽出

  - オンチェーンデータに含まれる個人データの匿名化・秘匿化

  - 個人データ利用ポリシーの自動制御

KIOXIA