

ブロックチェーンを用いたDIDの実装調査 及びそれを用いたシステムの構成

筑波大学大学院 システム情報工学研究群

博士前期課程1年 内堀 紘徳

目次

1. はじめに
2. 背景
3. IONの利用
4. DIDの活用可能性と活用方法の検討
5. インターンシップの感想

目次

1. はじめに
2. 背景
3. IONの利用
4. DIDの活用可能性と活用方法の検討
5. インターンシップの感想

インターンシップテーマ

「ブロックチェーンを用いたDIDの実装調査及びそれを用いたシステムの構成」

- ・ ブロックチェーンの応用例の一つとして分散型識別子（DID）が挙げられる
- ・ 現在あるブロックチェーンを用いたDIDの今の概況を調査し、ブロックチェーンとDIDを用いた具体的なシステムを検討・構成する

概要

- ・ 近年、BitcoinやEthereumをはじめとする暗号資産が世界中で注目されている
- ・ 暗号資産を支えるブロックチェーン技術は暗号資産以外の様々な応用分野への活用可能性が期待されている
- ・ 今回はその中でもDIDという技術に焦点を当ててインターンシップに取り組んだ
- ・ DIDはPoC段階の技術で課題も多く残されているが、活用可能性は十分にある分野だと感じた

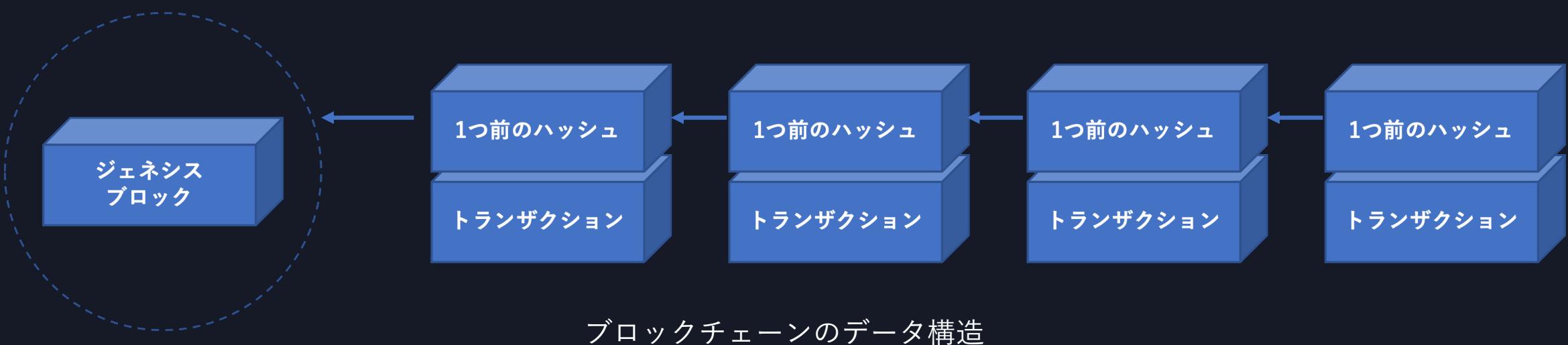


目次

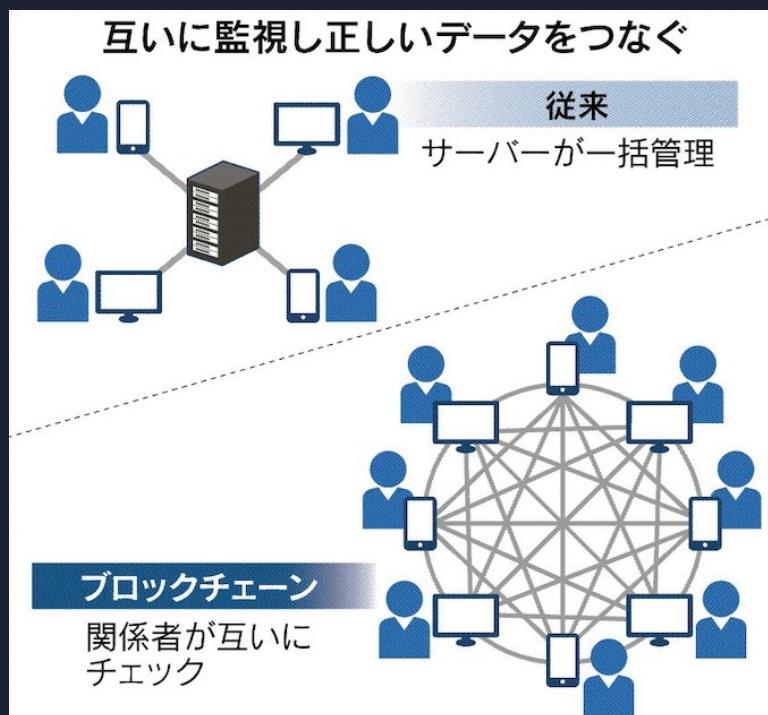
1. はじめに
2. 背景
3. IONの利用
4. DIDの活用可能性と活用方法の検討
5. インターンシップの感想

ブロックチェーンとは

ネットワーク内で発生した取引の記録を「ブロック」と呼ばれる記録の塊に格納し、個々のブロックには取引の記録に加えて、1つ前に生成されたブロックの内容を示すハッシュ値と呼ばれる情報などを格納したデータ構造



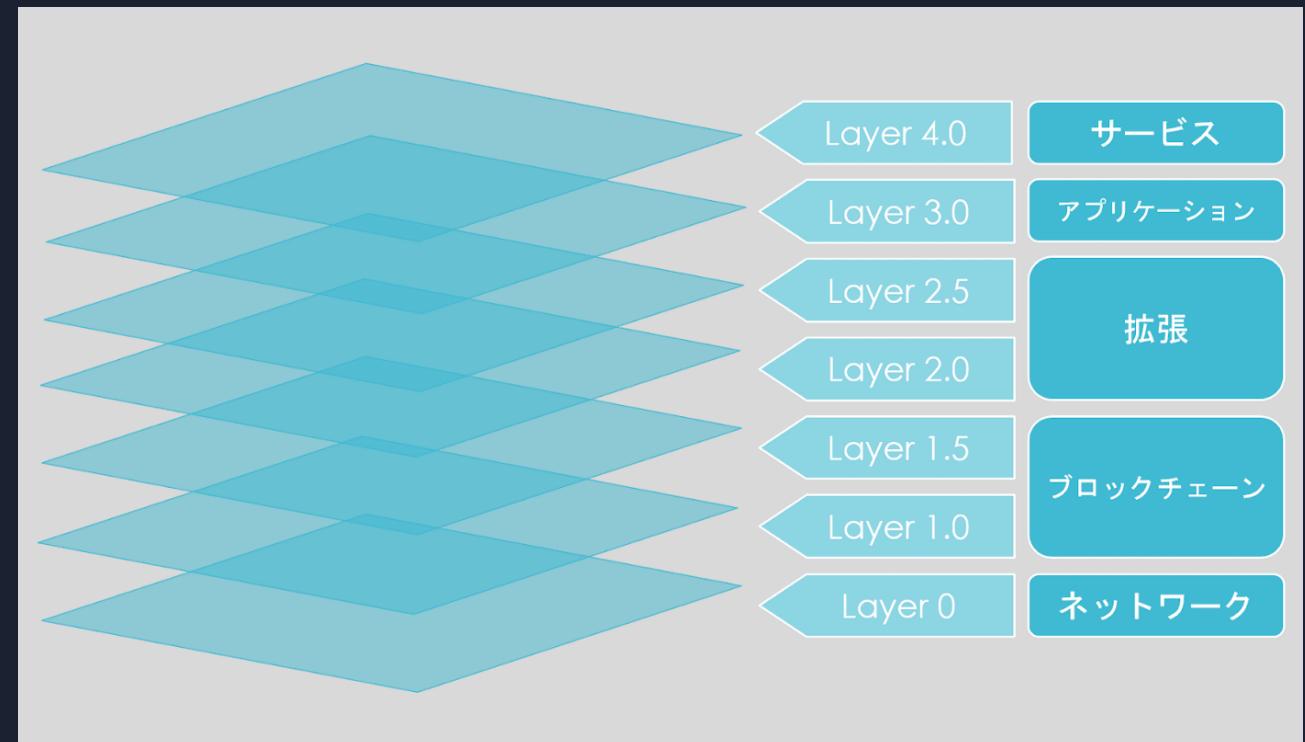
ブロックチェーンの主な特徴



1. 単一障害点のない分散システム
 - ノード同士がP2P (Peer to Peer) 通信でつながっている
 - 隣接ノード同士で情報を交換し合い、システム全体で同じ情報を保有 (分散台帳)
 - 一つのノードがダウンしても全体は影響を受けない
2. 耐改ざん性
 - データはブロックチェーンに保存
 - 改ざんするためにはそれ以降の全てのノードを変更しなければ矛盾が生じる
(ブロックには前のブロックのハッシュを格納するため)

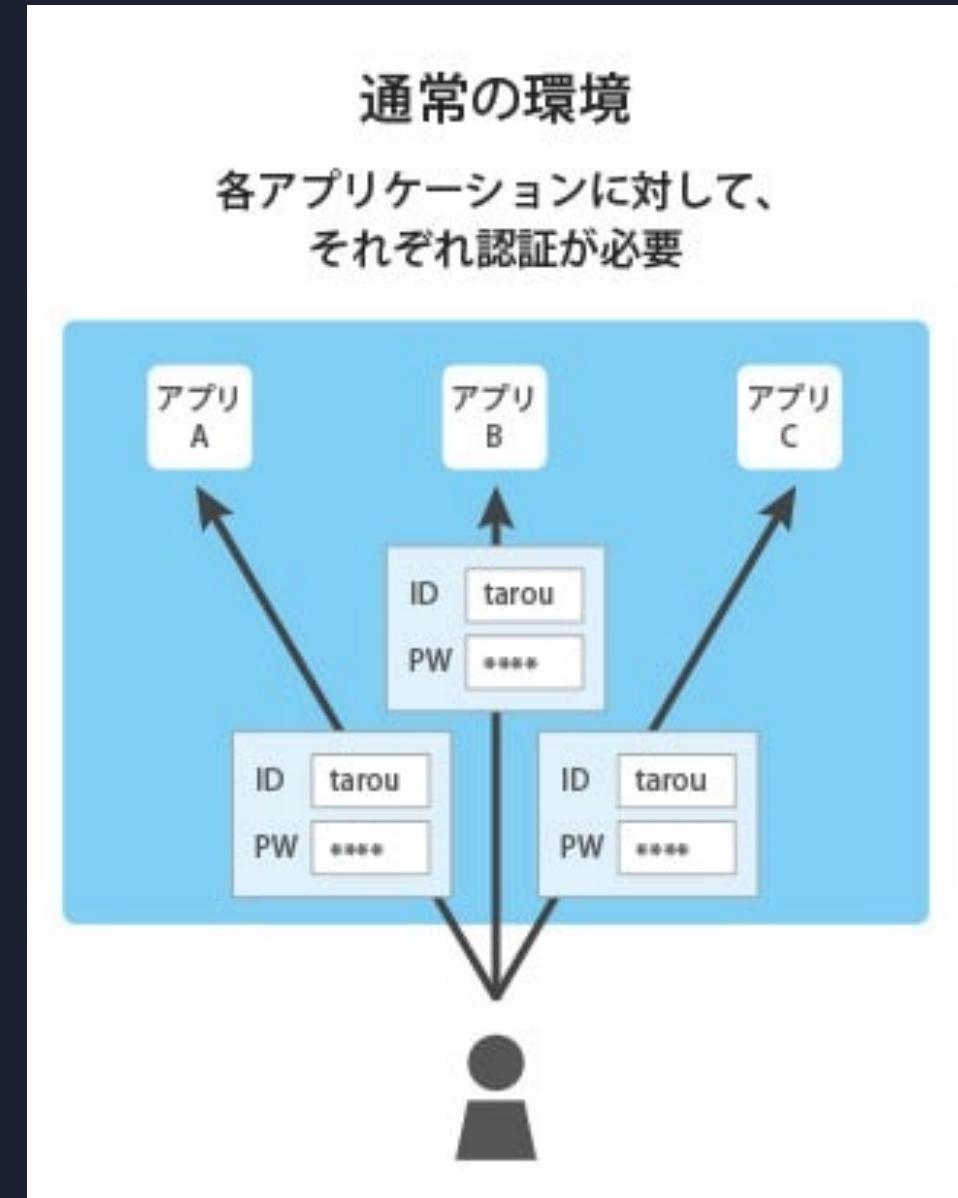
ブロックチェーンの レイヤ構造

- ・ ブロックチェーン技術はレイヤ構造で捉えられる
- ・ ブロックチェーンの分散台帳や耐改ざん性といった特徴から暗号資産以外の応用分野への活用が期待されている
- ・ 今回利用するDIDはLayer 2の「拡張領域」にあたる技術



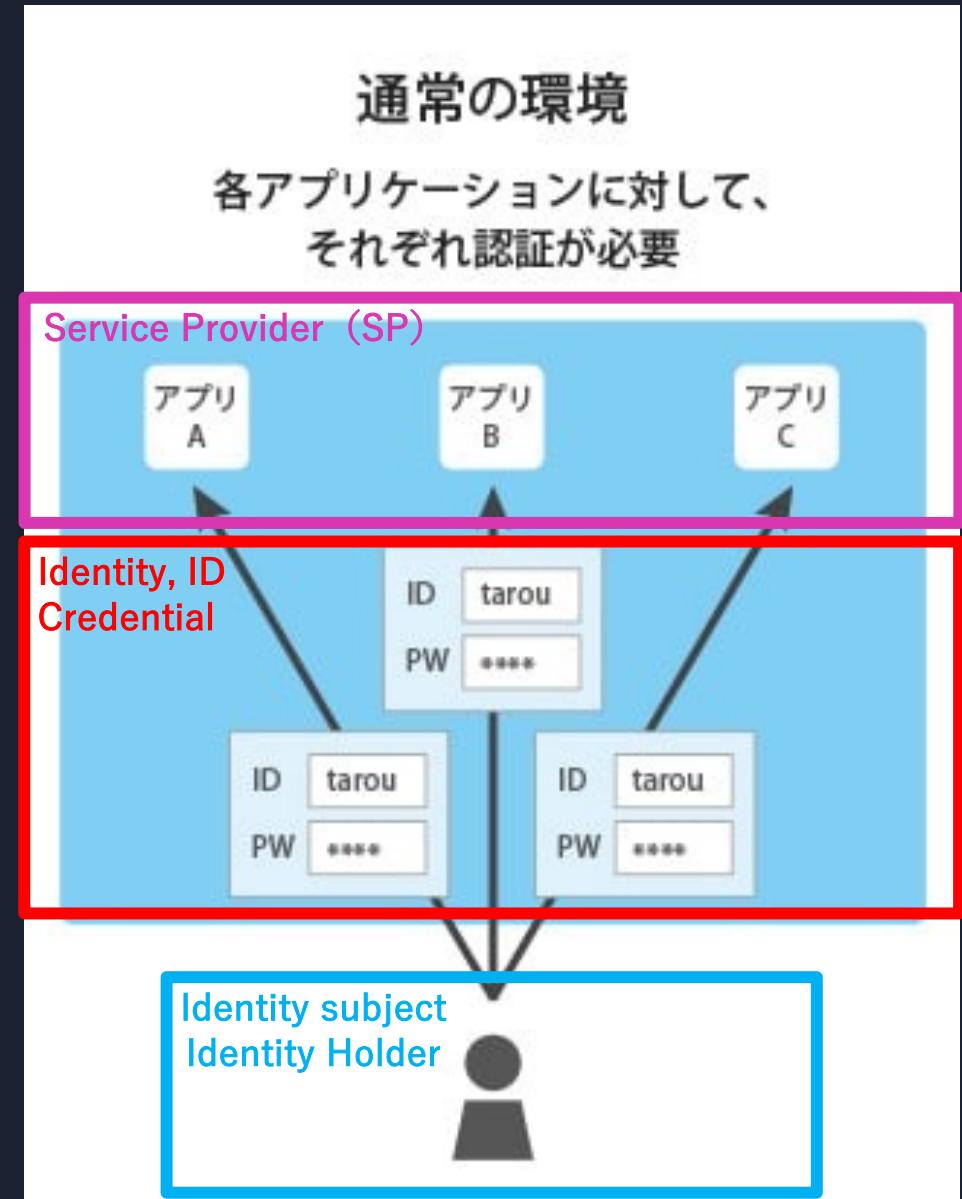
一般的なインターネットサービスの認証（1/2）

- ユーザは各サービスやシステムごとにIDやパスワードを保持している
- これらの情報を入力して認証を受け、成功した場合はログインできる



一般的なインターネットサービスの認証（1/2）

- ユーザは各サービスやシステムごとにIDやパスワードを保持している
- これらの情報を入力して認証を受け、成功した場合はログインできる



メールアドレスでログイン

 メールアドレス

 パスワード

パスワードを表示する

ログインする >

▶ [パスワードをお忘れの方](#)

アカウントをお持ちでない方

メールアドレスで会員登録する >

外部IDでログイン

▶ [外部ID/ニコニコでログインする前に](#)



ニコニコでログイン



LINEでログイン



Facebookでログイン



Twitterでログイン



dアカウントでログイン



Googleでログイン



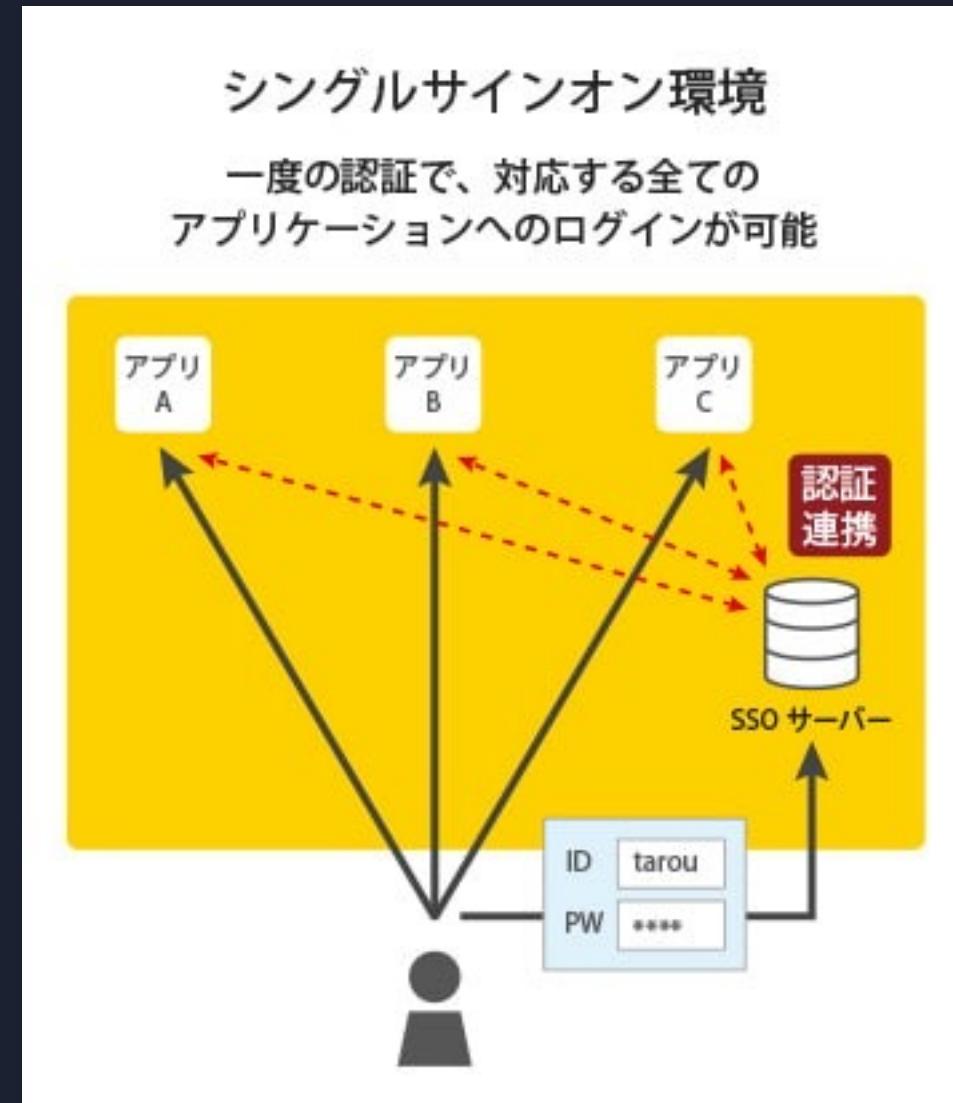
Yahoo! JAPAN IDでログイン



Appleでサインイン

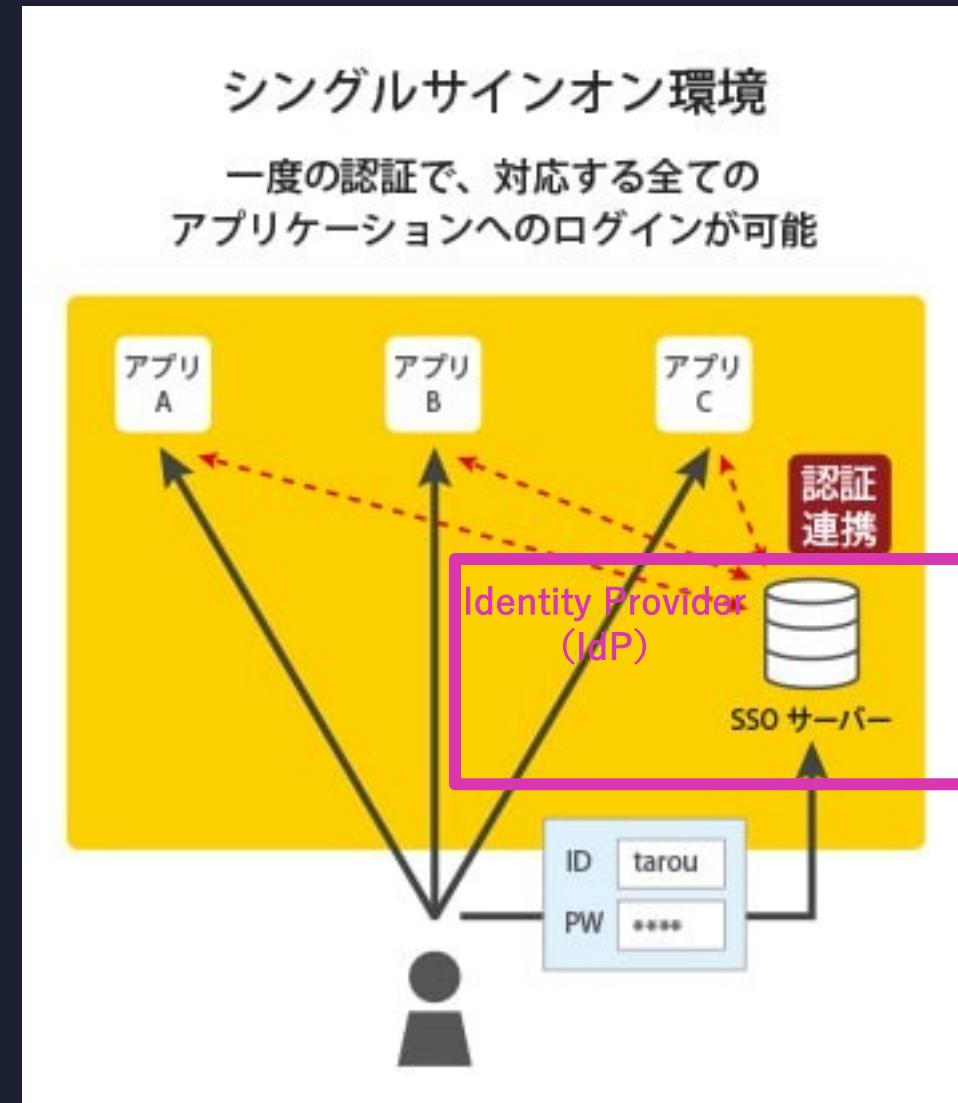
一般的なインターネットサービスの認証（2/2）

- ・ シングルサインオン
(SSO : Single Sign On)
- ・ 一つのユーザ認証で複数のシステムが利用できる仕組み
- ・ サービスの認証部分を外部のサービスに依頼する



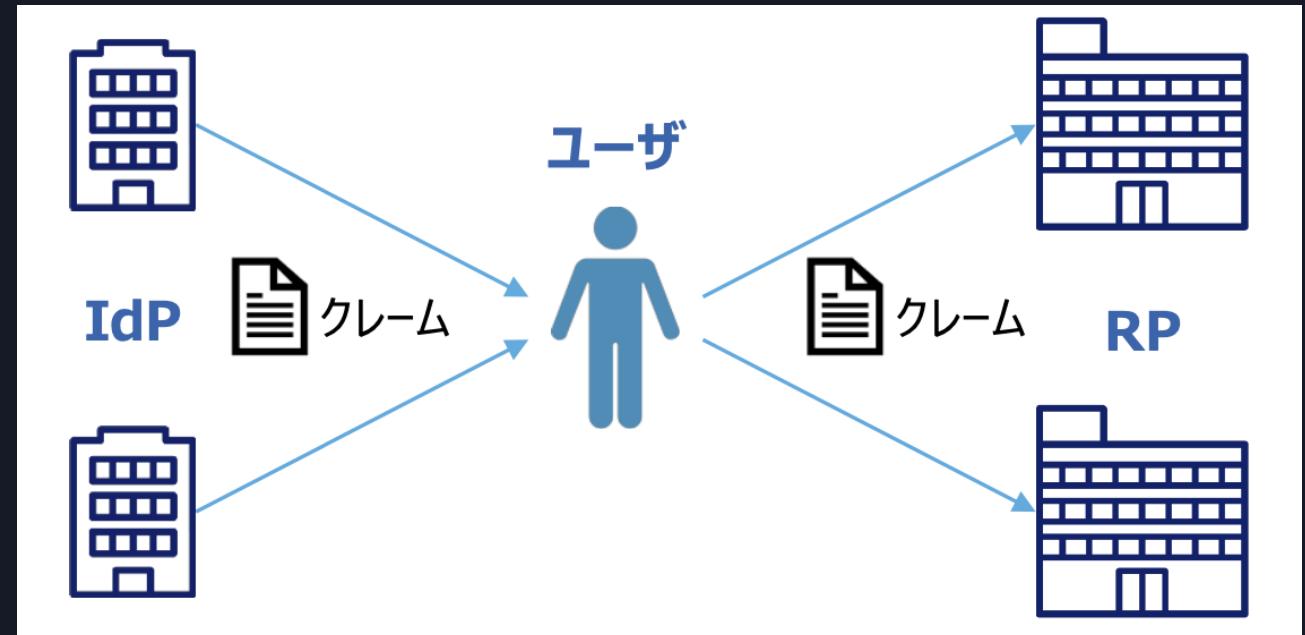
一般的なインターネットサービスの認証（2/2）

- ・ シングルサインオン
(SSO : Single Sign On)
- ・ 一つのユーザ認証で複数のシステムが利用できる仕組み
- ・ サービスの認証部分を外部のサービスに依頼する



自己主権型アイデンティティ (SSI) とは

- 「誰にも依存せずに個人が自分自身のアイデンティティをコントロールできるようにする」という考え方
- ユーザが自分の属性情報（クレーム）を自分で管理する
- IdPはアイデンティティの作成・発行のみを行い、認証情報の保持はない
- 信頼できる組織から発行された本人の属性情報を取得し、ユーザが許可した範囲で提示する



<アクター>

IdP: アイデンティティプロバイダ
アイデンティティを作成する

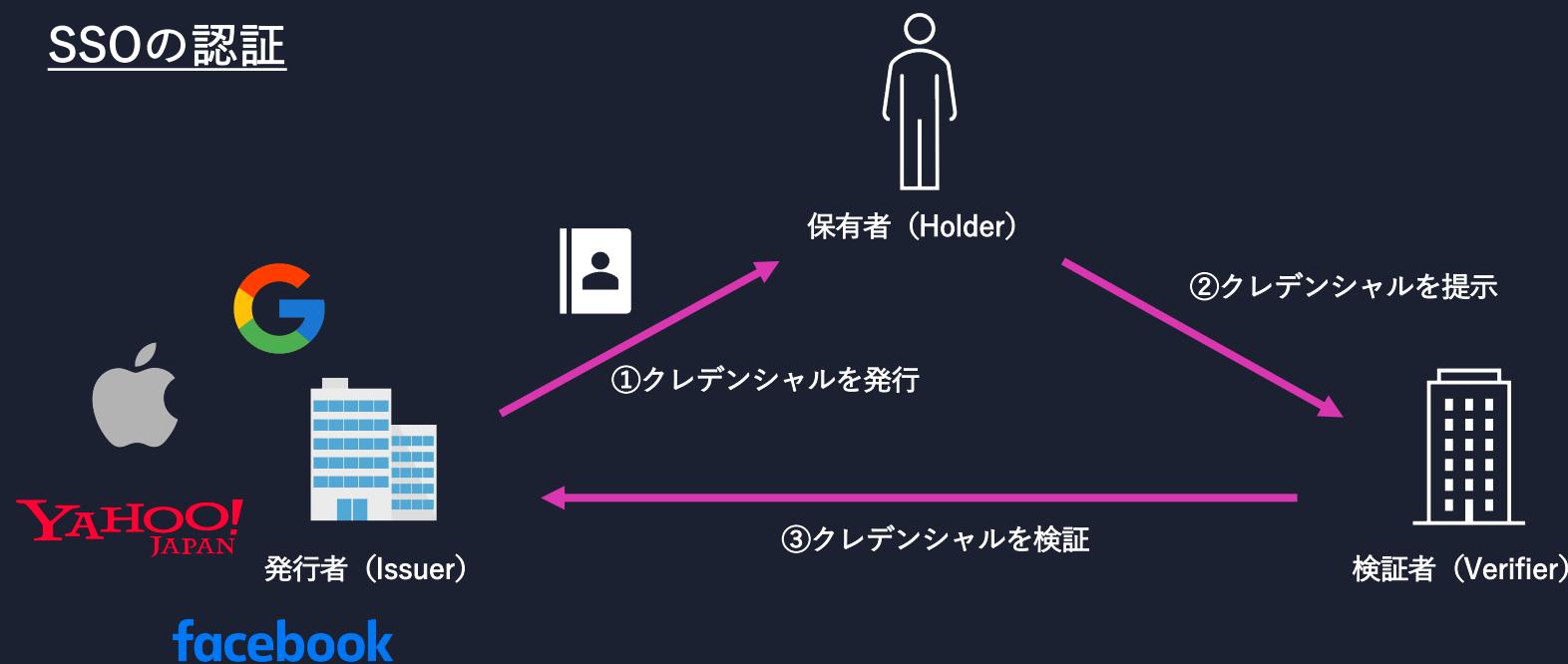
RP: リライングパーティ
検証を行う

Verifiable Credentialとは

- ・ 検証可能な資格証明書
- ・ 定義：「内容の検証がオンラインで可能なデジタル個人情報」
- ・ 例：分散型アイデンティティ

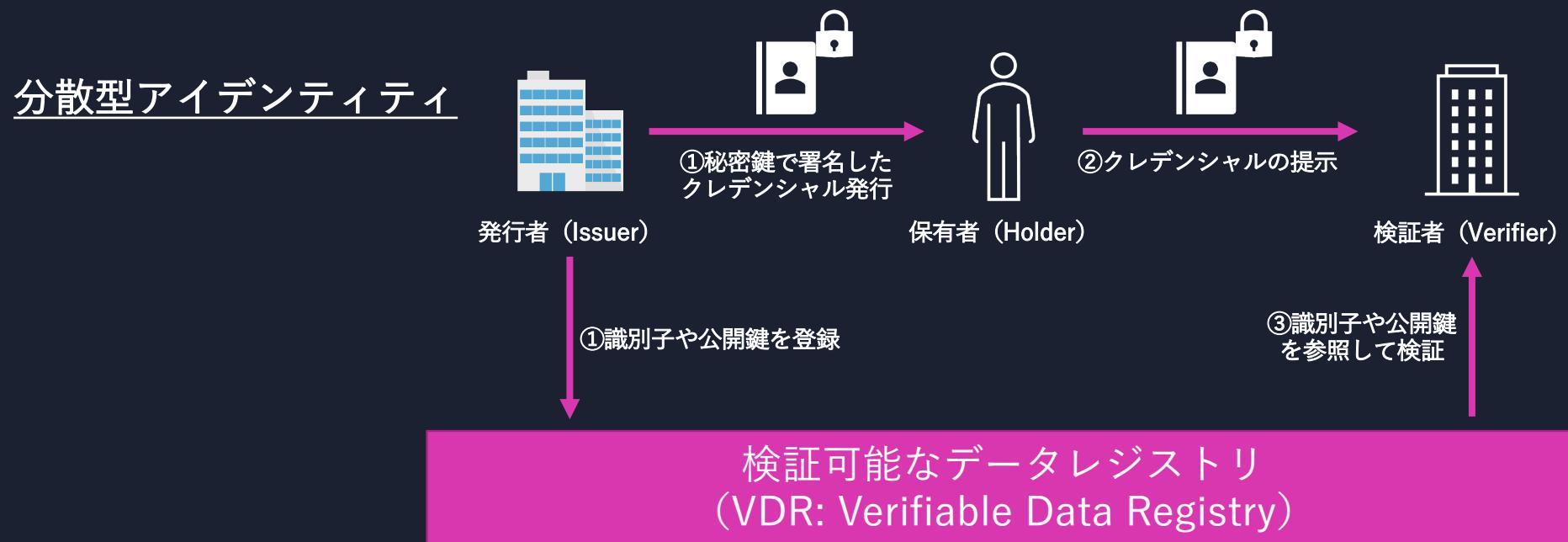
分散型アイデンティティ (1/3)

- 「ユーザのアイデンティティがIdPに依存しない」というSSIの考え方に基づき、分散システムを利用してこれを実現する



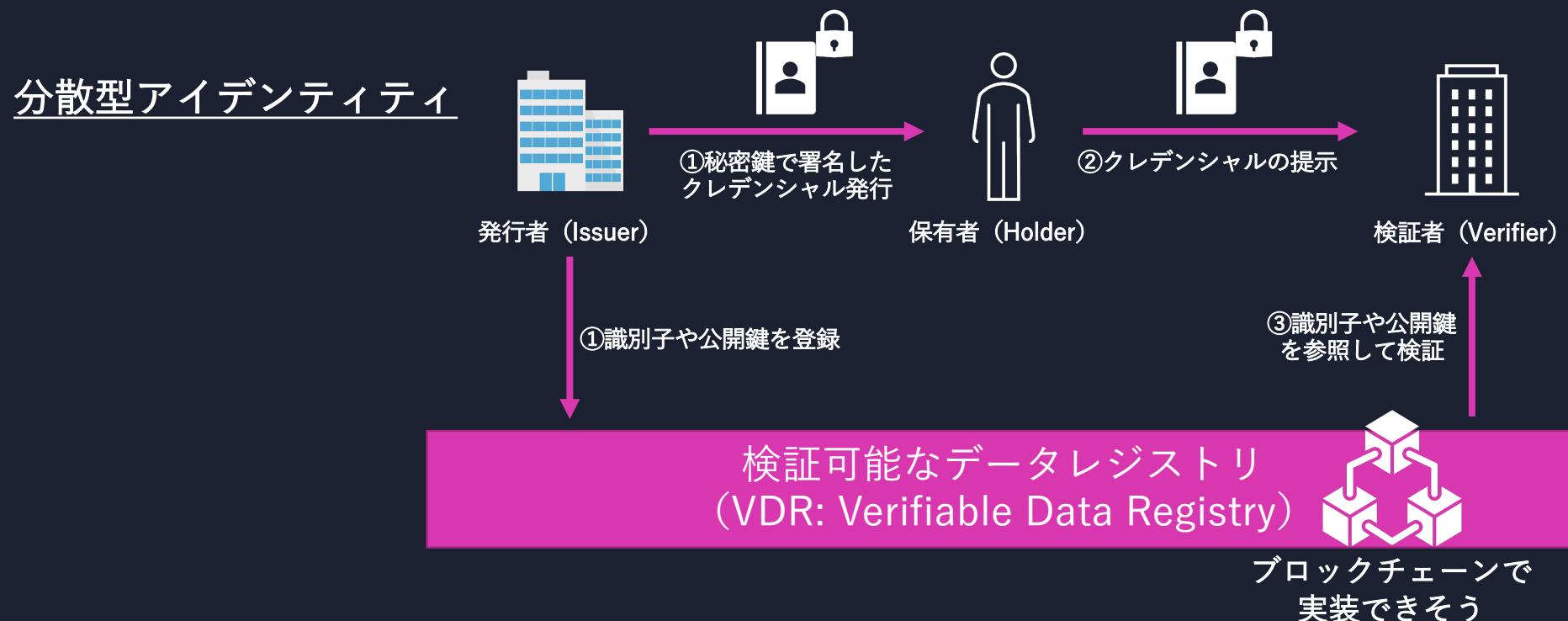
分散型アイデンティティ (1/3)

- 「ユーザのアイデンティティがIdPに依存しない」というSSIの考え方に基づき、分散システムを利用してこれを実現する



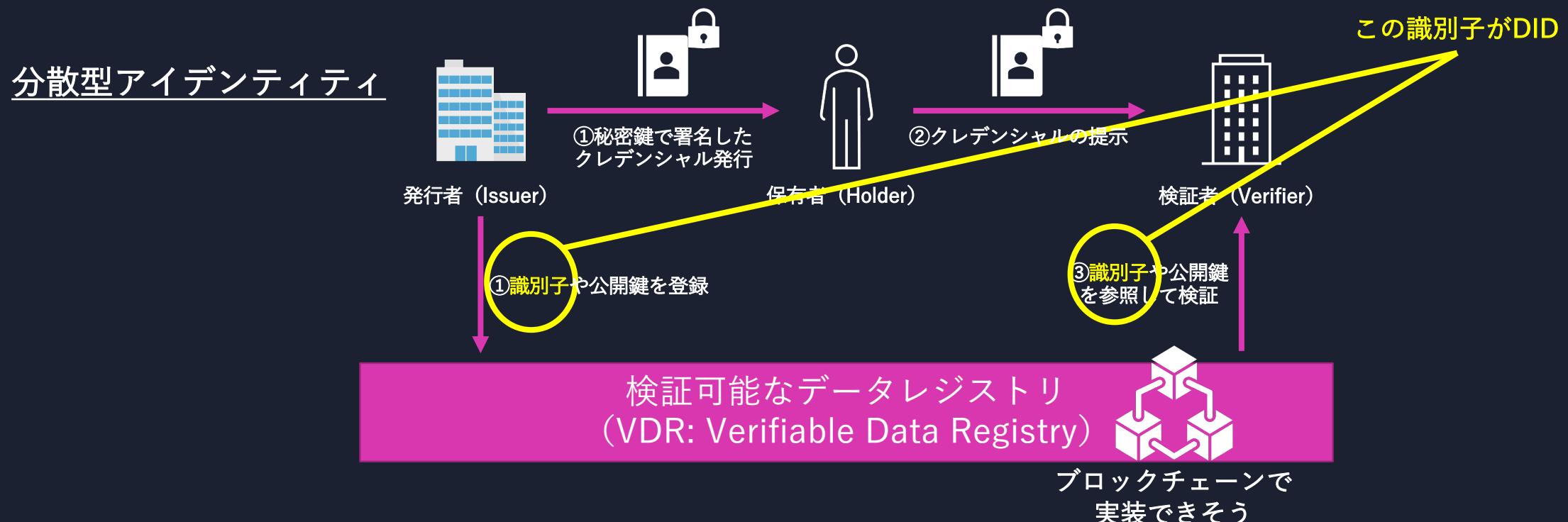
分散型アイデンティティ (2/3)

- 「ユーザのアイデンティティがIdPに依存しない」というSSIの考え方に基づき、分散システムを利用してこれを実現する



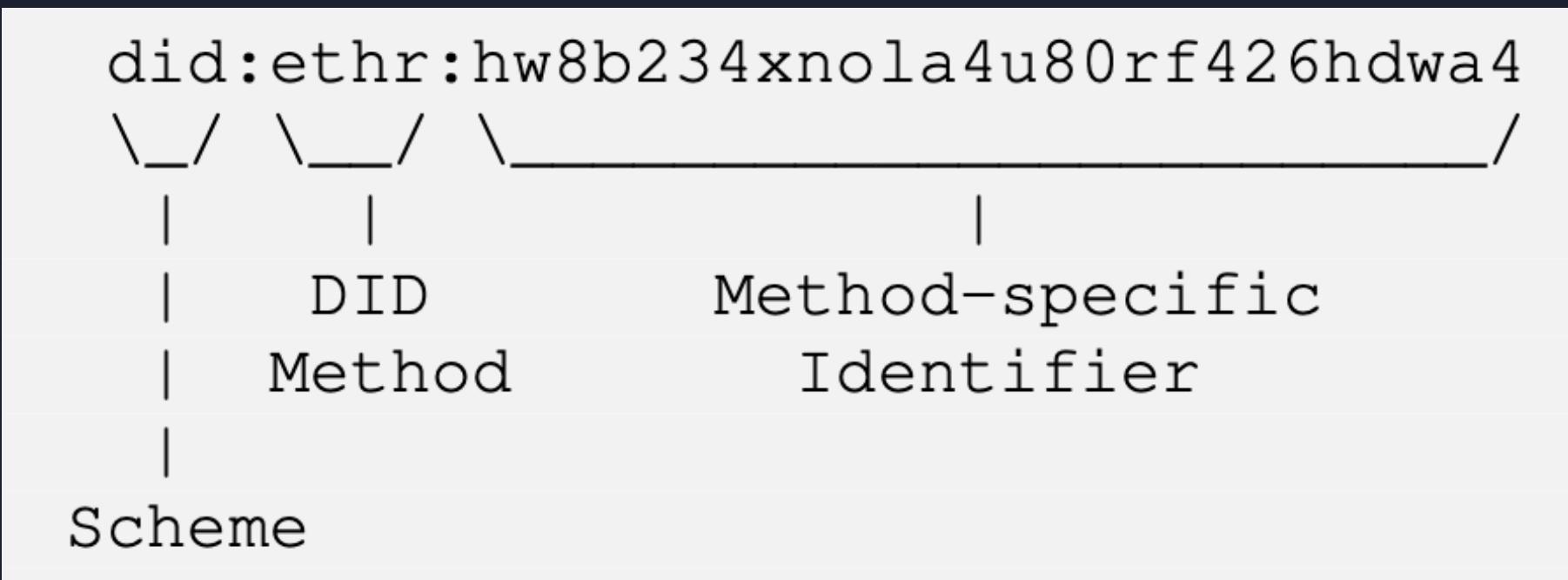
分散型アイデンティティ (3/3)

- 「ユーザのアイデンティティがIdPに依存しない」というSSIの考え方に基づき、分散システムを利用してこれを実現する



DID（分散型識別子）とは

- IssuerとHolderを世界規模で一意に参照するために、識別子の概念がある



DIDとResolve (1/2)

- DIDはW3Cにより標準化されている
- 検証者（Verifier）はDIDに紐づいたDID Documentの内容を調べることで検証を行うことができる。

The screenshot shows a web browser window with the URL `localhost:3000/identifiers/did:ion:test:EiCLWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw`. A callout box highlights the URL bar with the text "指定方式" (指定方式) and "スキーム:DIDメソッド:メソッド内で一意の識別子" (Scheme:DID Method:Unique identifier within the method). The main content area displays a JSON object representing a DID Document.

```
{"@context": "https://w3id.org/did-resolution/v1", "didDocument": {"id": "did:ion:test:EiCLWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw", "@context": ["https://www.w3.org/ns/did/v1", {"@base": "did:ion:test:EiCLWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw"}], "verificationMethod": [{"id": "#sign", "controller": "did:ion:test:EiCLWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw", "type": "EdDSAsecp256k1VerificationKey2019", "publicKeyJwk": {"crv": "secp256k1", "kty": "EC", "x": "x2ZoT2yFRMyIHKUgxCjQfpGr0jpdMy7I1qTccN8THQ4", "y": "UGtsr59itS44XgMVI62geeZr52ZQ6LFCZquLDqlK1YM"}]}, "authentication": [{"id": "#sign"}], "didDocumentMetadata": {"method": {"published": true, "recoveryCommitment": "EiBksRY4HzocVV_WCpQXJ90_FTp0fRm0Ju8kSY77Bs0kzA", "updateCommitment": "EiApril1a1Dg0wy7UtLVGKMxARd05Bo0y0-YCSVU-QHx3Ig"}, "canonicalId": "did:ion:test:EiCLWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw"}]}
```

DIDとResolve (1/2)

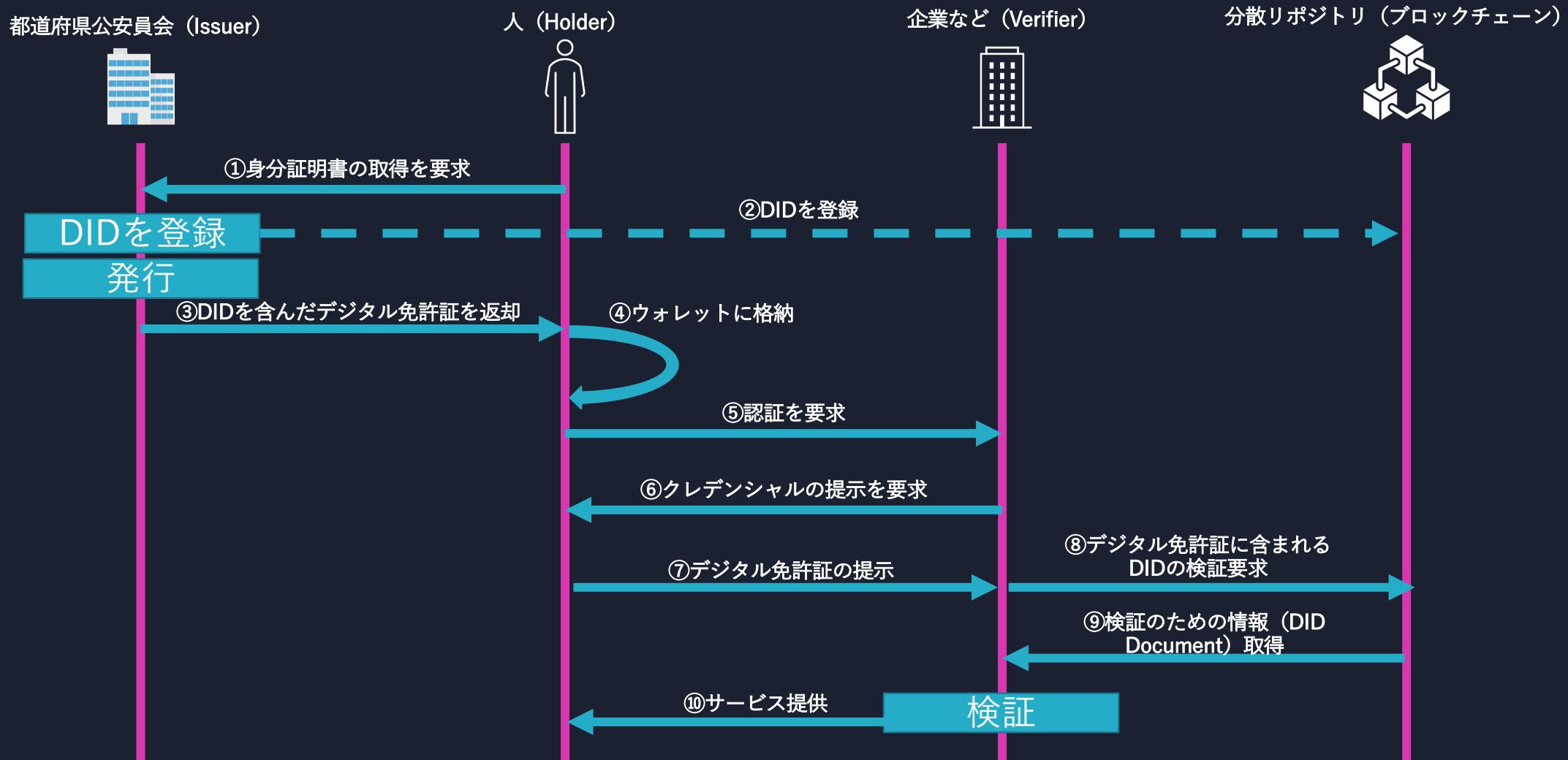
- DIDはW3Cにより標準化されている
- 検証者（Verifier）はDIDに紐づいたDID Documentの内容を調べることで検証を行うことができる

DID Document
DIDの検証に必要なメタデータ.
公開鍵による検証方法などが含まれている.

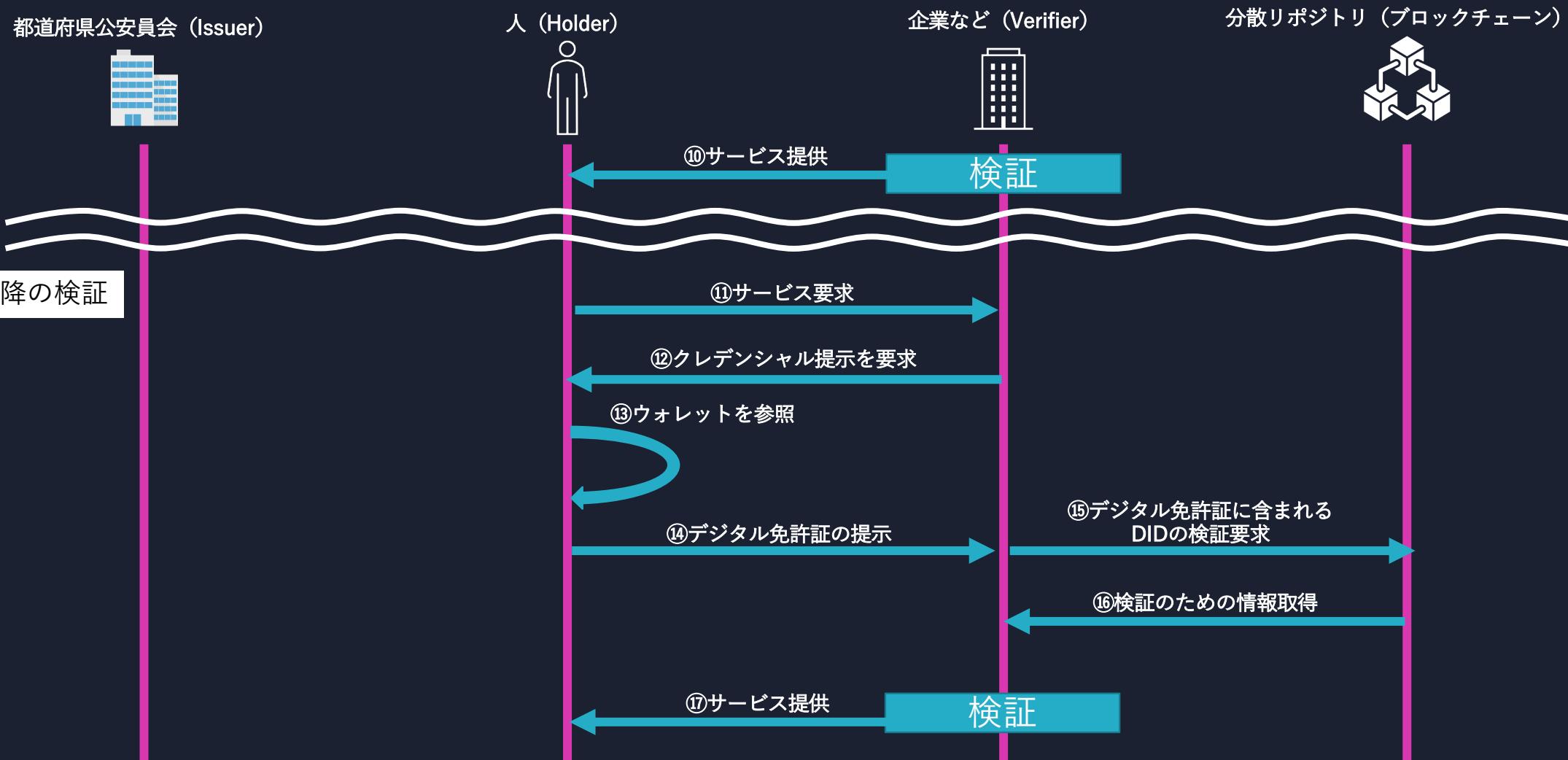


```
{"@context": "https://w3id.org/did-resolution/v1", "didDocument": {"id": "did:ion:test:EiCLWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw", "@context": ["https://www.w3.org/ns/did/v1", {"@base": "did:ion:test:EiCLWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw"}], "verificationMethod": [{"id": "#sign", "controller": "did:ion:test:EiCLWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw", "type": "EcdsaSecp256k1VerificationKey2019", "publicKeyJwk": {"crv": "secp256k1", "kty": "EC", "x": "x2ZoT2yFRMyIHKUgxCjQfpGr0jpdMy7I1qTccN8THQ4", "y": "UGtsr59itS44XgMVI62geeZr52ZQ6LFCZquLDqlK1YM"}]}, "authentication": [{"id": "#sign"}], "didDocumentMetadata": {"method": {"published": true, "recoveryCommitment": "EiBksRY4HzocVV_WCpQXJ90_FTp0fRm0Ju8kSY77Bs0kzA", "updateCommitment": "EiApril1a1Dg0wy7UtLVGKMxARd05Bo0y0-YCSVU-QHx3Ig"}, "canonicalId": "did:ion:test:EiCLWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw"}]}
```

ユースケース：運転免許証による身分証明の例（1/2）



ユースケース：運転免許証による身分証明の例（2/2）



インターンシップテーマ

「ブロックチェーンを用いたDIDの実装調査及びそれを用いたシステムの構成」

- ・ ブロックチェーンの応用例の一つとして分散型識別子（DID）が挙げられる
- ・ 現在あるブロックチェーンを用いたDIDの今の概況を調査し、ブロックチェーンとDIDを用いた具体的なシステムを検討・構成する

DIDの主なプロダクト

- SpruceID
6つのブロックチェーン (Ethereum, polygon, TEZOS, …) で
シームレスに利用可能
- ION
Microsoftが中心となって開発
基盤のブロックチェーンはBitcoin
- Hyperledger Indy
Hyperledgerが提供するOSS
基盤のブロックチェーンはHyperledger



DIDの主なプロダクト

- SpruceID
6つのブロックチェーン (Ethereum, polygon, TEZOS, …) で
シームレスに利用可能
- ION
Microsoftが中心となって開発
基盤のブロックチェーンはBitcoin
 - Bitcoinはパブリックなブロックチェーン
 - 長く利用され十分に安全性がテストされている
 - 世界中に多くのノードがある
- Hyperledger Indy
Hyperledgerが提供するOSS
基盤のブロックチェーンはHyperledger



Layer 2 DID Network

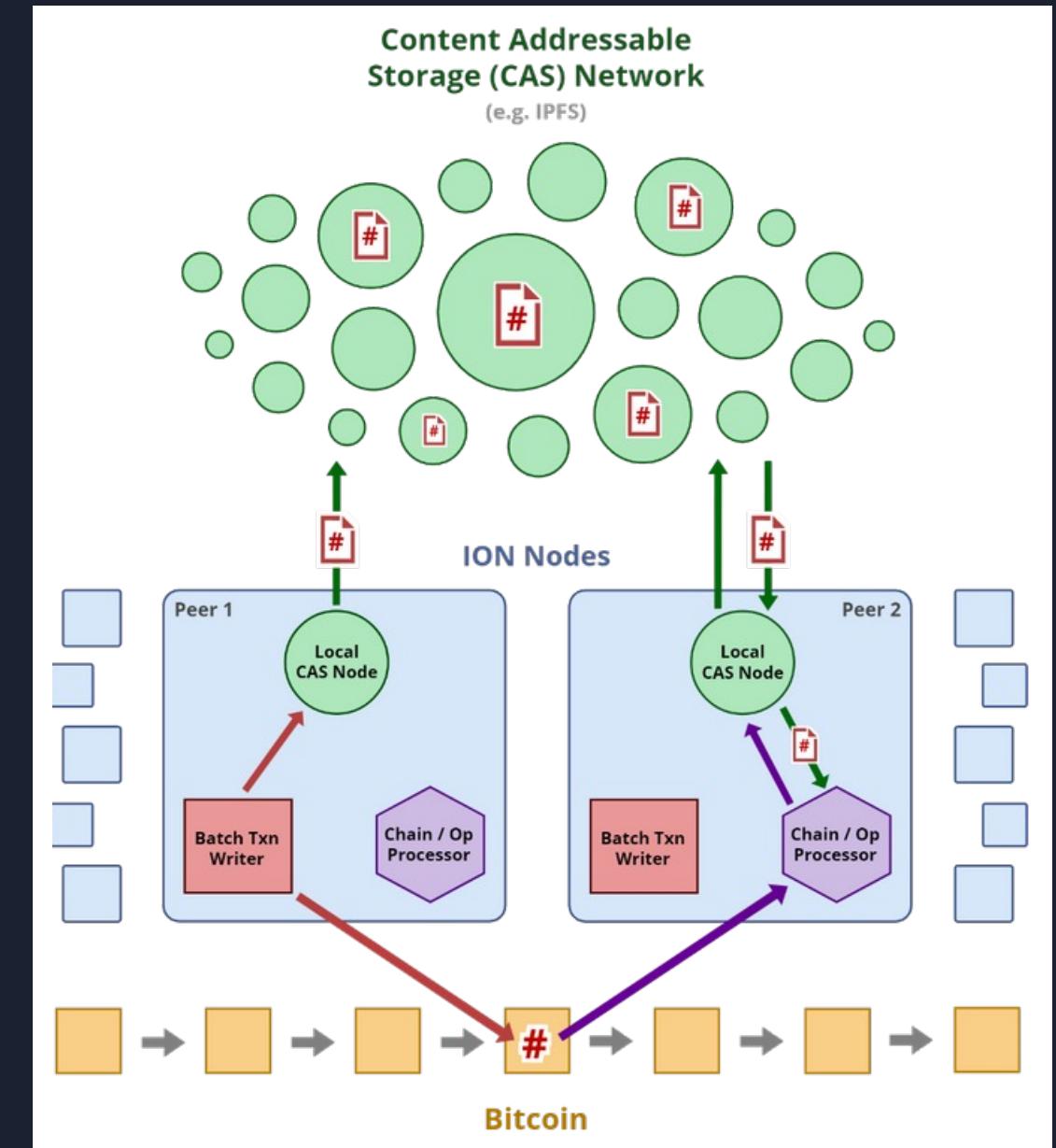
ION (Identity Overlay Network) とは

- DIDプロダクトの一つ
- 基盤のブロックチェーンにBitcoinを採用
- Layer 2（拡張領域）で実装
- Microsoftが中心となって開発を進めている

IONのアーキテクチャ

全体のアーキテクチャは3層構造

1. 基盤となる分散型台帳 (Bitcoin)
2. セカンドレイヤーとなるION
3. CAS (Content-Addressable Storage) ネットワークのIPFS



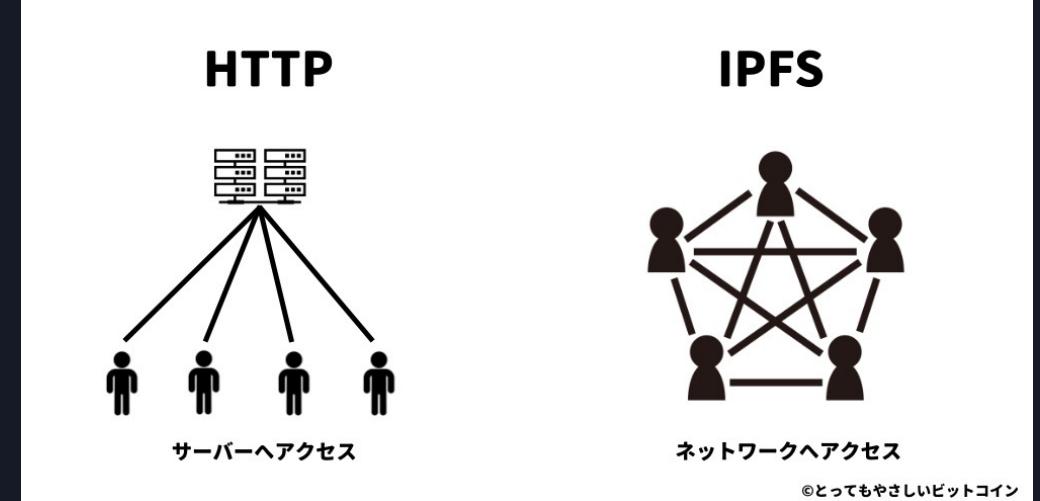
IONの実装を支える技術

- IPFS
 - P2Pで構成された分散ファイルシステム
 - DID Documentを格納する
 - コンテンツアドレス (CID) によりファイルを一意に識別
- MongoDB
 - ドキュメント指向データベース (NoSQLに分類される)
 - 今回はローカルデータの保存に利用



IPFS (InterPlanetary File System)

- HTTPを補完/置換するプロトコルとして位置付けられている
 - 中央集権的なHTTPとは異なり、P2Pネットワークでファイルを保持する分散システム
 - 特徴
 - 負荷分散
 - 耐検閲性
 - 耐障害性
 - 耐改ざん性
- } ブロックチェーンとの親和性高い



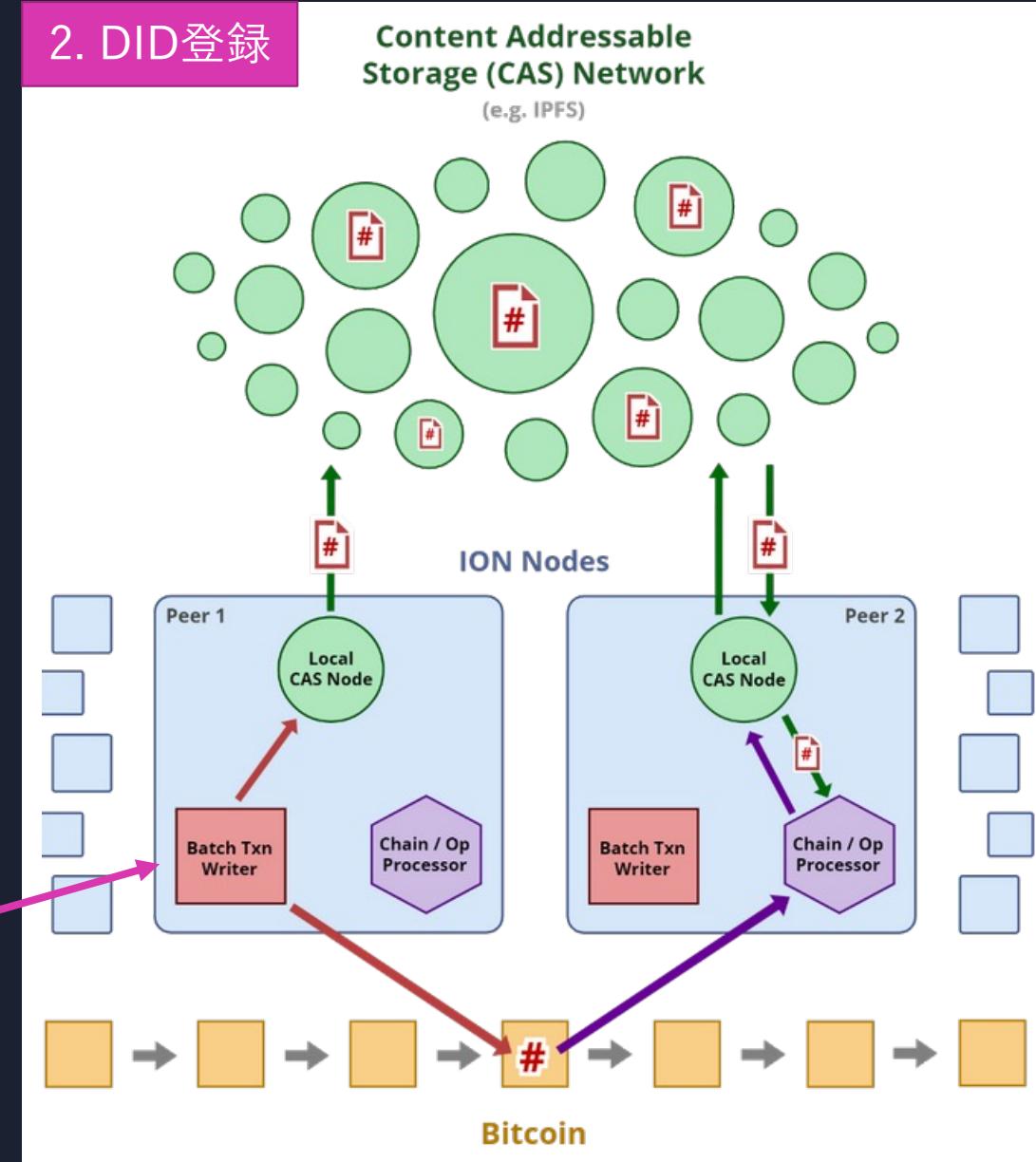
	HTTP	IPFS
特徴	中央集権型	分散型 (P2P)
ファイル指定	URL (ファイルの位置)	CID (ファイルのハッシュ値)
プロトコル	ロケーション指向	コンテンツ指向

2. 背景

- IssuerはIONノードで
- DID登録命令 (Create)
 - DID Documentを作成する



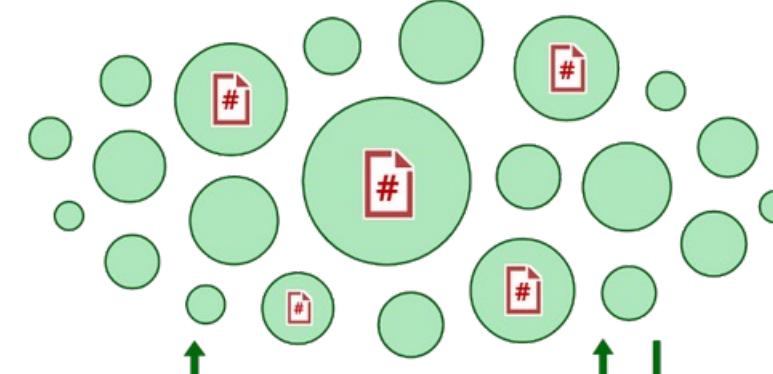
Issuer



Microsoft 「ION – Booting up the network」

<https://techcommunity.microsoft.com/t5/identity-standards-blog/ion-booting-up-the-network/ba-p/1441552>

2. DID登録

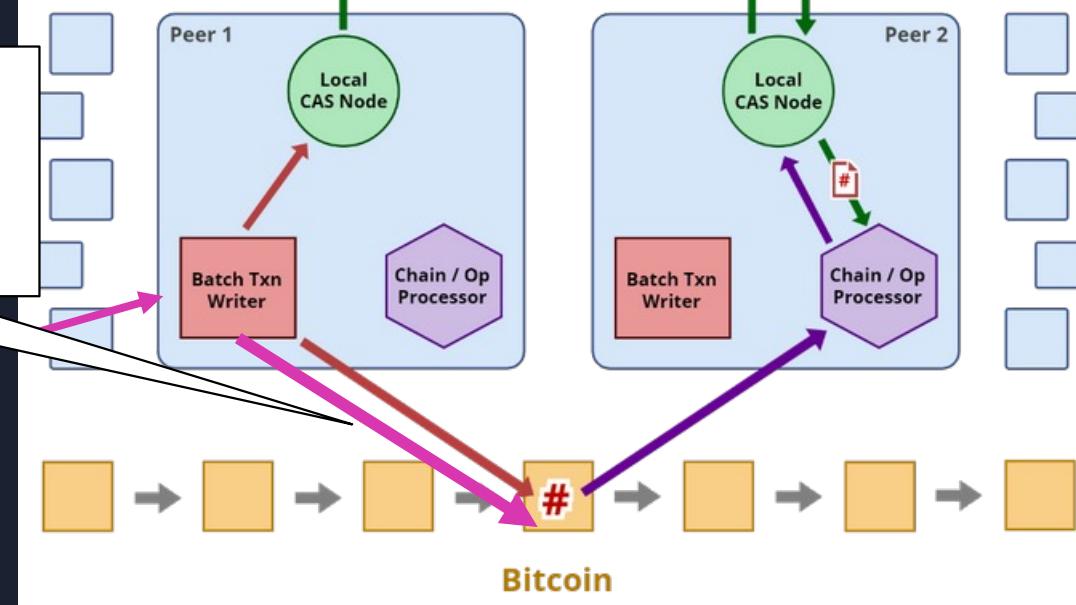
Content Addressable Storage (CAS) Network
(e.g. IPFS)

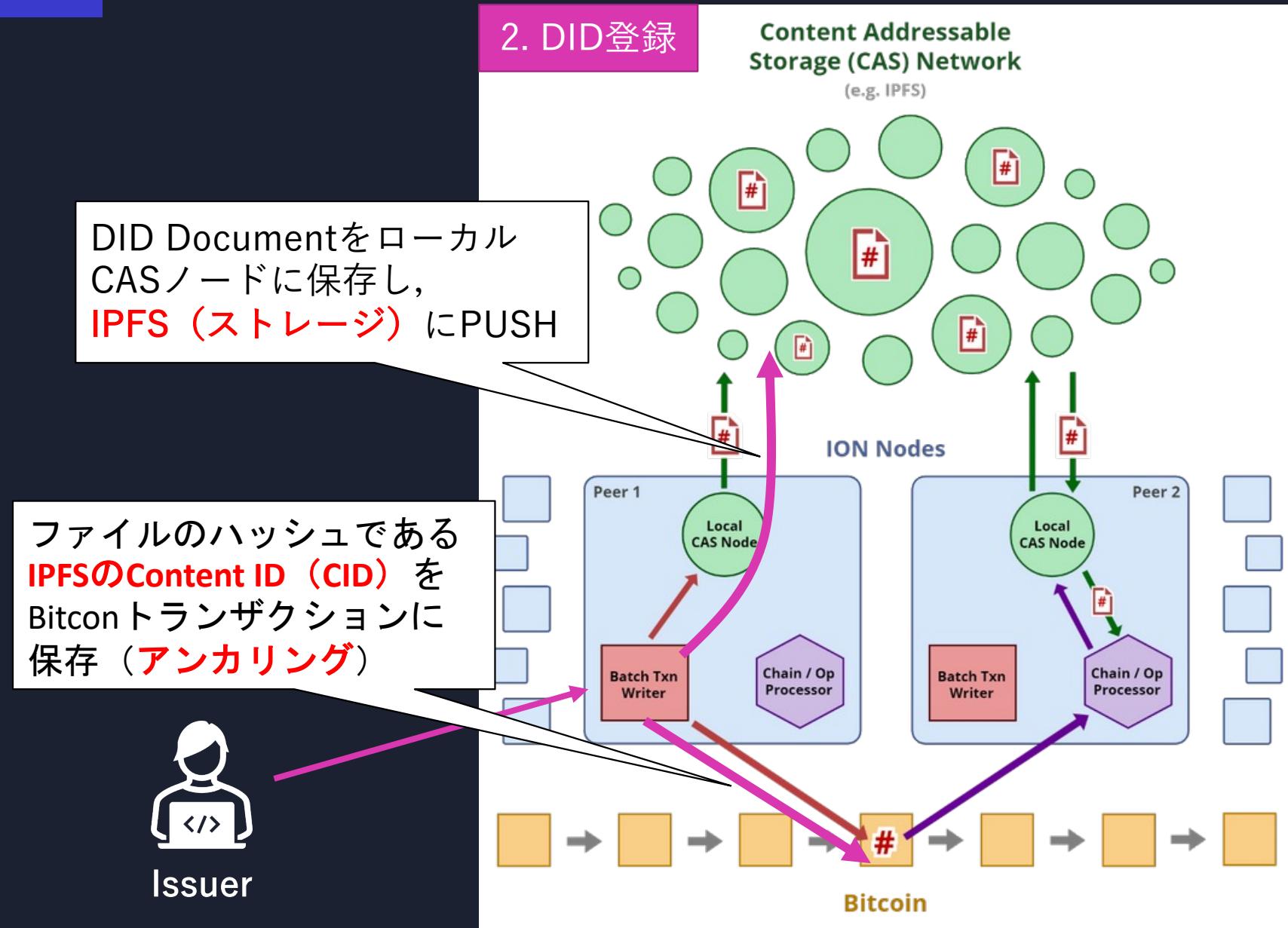
ION Nodes

ファイルのハッシュである
IPFSのContent ID (CID) を
Bitcoin トランザクションに
保存（アンカリング）

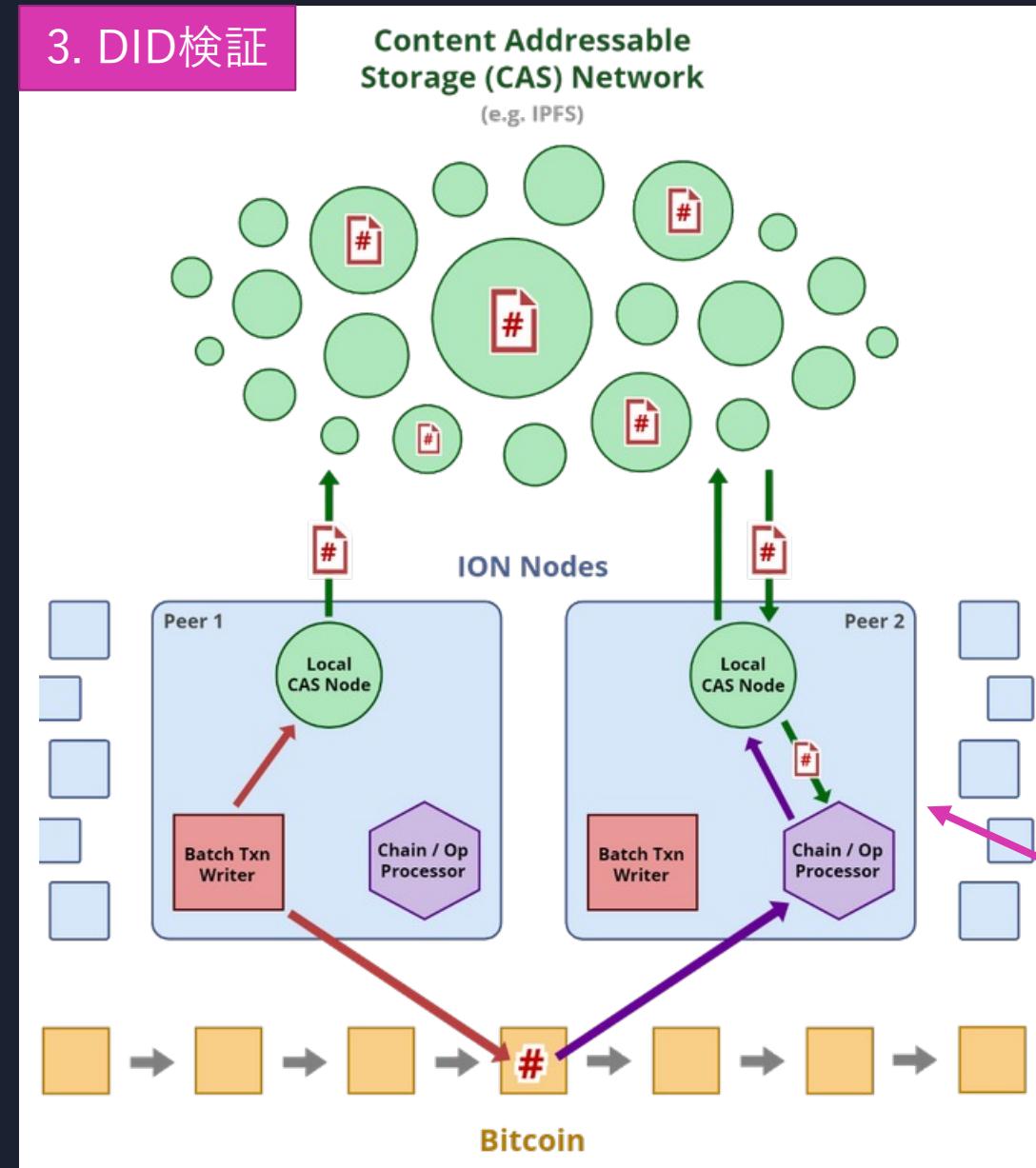


Issuer

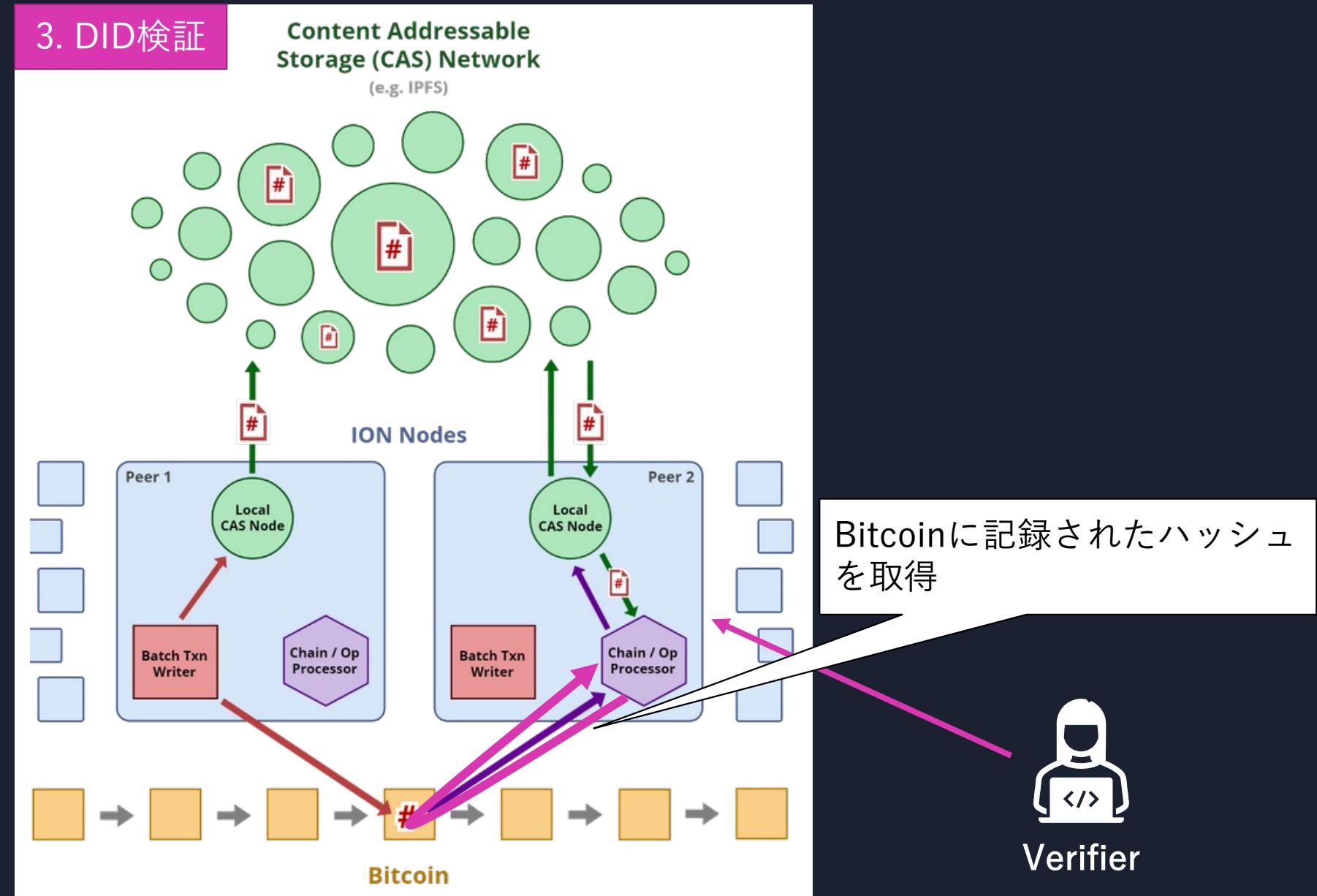




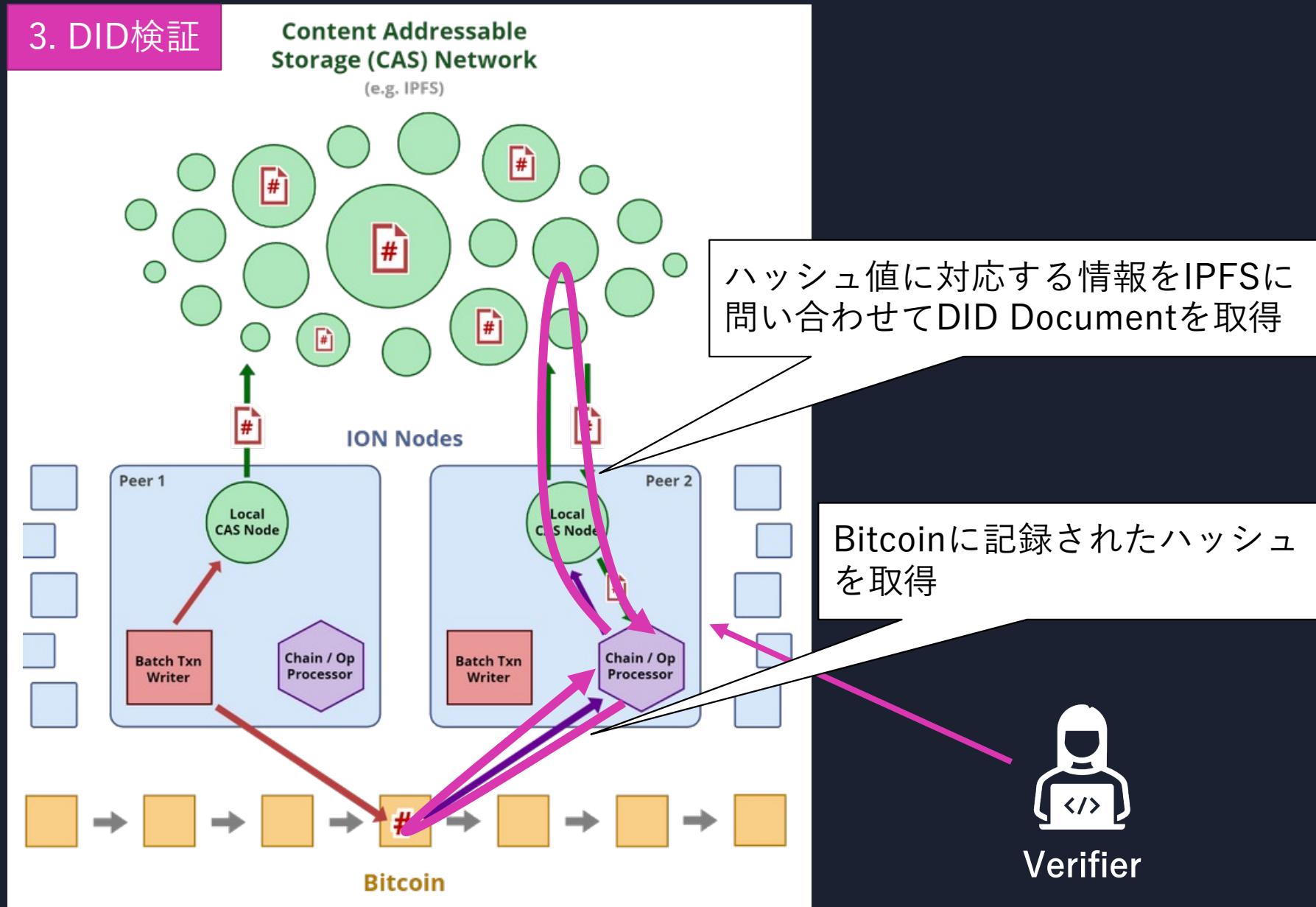
Issuer

Issuer


Issuer



目次

1. はじめに
2. 背景
3. IONの利用
4. DIDの活用可能性と活用方法の検討
5. インターンシップの感想

IONの利用

- IONを実際に利用し、DIDの仕組みを確かめた
- 構築した環境の仕様

サーバ	種別	AWS EC2
	OS	Ubuntu 22.04
	リージョン	東京
	アーキテクチャ	i386, x86_64
	メモリ	4GB
	ストレージ	1TB (メインネットの場合)
Bitcoin クライアント		Bitcoin Core 0.18.0
IPFS (Kubo)		v0.15.0
MongoDB		server-6.0
ION		1.0.2, 1.0.4

IONを実際に利用している様子

エクスプローラー … { } testnet-bitcoin-config.json ~/ion_config { } testnet-bitcoin-config.json ~/.../json X { } testnet-bitcoin-versioning.json

> 開いているエディター ion > json > { } testnet-bitcoin-config.json > ...

UBUNTU [SSH: ION_AWS]

- > .bitcoin
- > .cache
- > .config
- > .ipfs
- > .local
- > .mongodb
- > .npm
- > .ssh
- > .vscode-server
- > blockchain
- > hasegawa
- > ion
- > ion_config
- { } testnet-bitcoin-co...
- { } testnet-bitcoin-ver...
- > ion_core_config
- > ion-1.0.2
- > ion-did
- > kubo
- > snap
- > uchibori
- ≡ .bash_history
- \$.bash_logout
- \$.bash_profile
- \$.bashrc
- ≡ .dbshell
- ≡ .lessht
- JS .mongorc.js
- \$.profile
- ≡ .sudo_as_admin_su...
- ≡ .viminfo
- ≡ .wget-hsts
- \$.xsessionrc
- ≡ ipfs-desktop-0.23.0...

> アウトライン

> タイムライン

問題 出力 デバッグ コンソール ターミナル ポート 6

ok: 1

}

No queued operations to batch.

Event emitted: sidetree_batch_writer_loop_success: {"batchSize": 1}

End batch writing. Duration: 14 ms.

Waiting for 60 seconds before writing another batch.

Handling resolution request for: did:ion:test:intern2022:EiClWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw...

Resolving DID unique suffix 'EiClWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw'...

CommandSucceededEvent {
 connectionId: 'localhost:27017',
 requestId: 4371,
 commandName: 'find',
 duration: 4,
 reply: {
 cursor: { firstBatch: [Array], id: 0, ns: 'ion-testnet-core.operations' },
 ok: 1
 }
} DID Document found for DID 'did:ion:test:intern2022:EiClWZ1MnE8PHjH6y4e4nCKgtKnI1DK1foZiP61I86b6pw'...
Fetching Sidetree transactions from blockchain service...
Fetching URI 'http://127.0.0.1:3002/transactions?since=10086919673217032&transaction-time-hash=00000000000000002b6ee43309d51f38c0b1d475f7d604dc20b357c4e949d1a7'...
Fetch response: 200'.
Fetched 0 Sidetree transactions from blockchain service in 63 ms.
Successfully kicked off downloading/processing of all new Sidetree transactions.
Processing previously unresolvable transactions if any...
CommandSucceededEvent {
 connectionId: 'localhost:27017',
 requestId: 4428,
 commandName: 'find',
 duration: 0,
 reply: {
 cursor: {
 firstBatch: [],
 id: 0,
 ns: 'ion-testnet-core.unresolvable-transactions'
 },
 ok: 1
 }
} Fetched 0 unresolvable transactions to retry in 1 ms.
Event emitted: sidetree_observer_loop_success
Waiting for 60 seconds before fetching and processing transactions again.

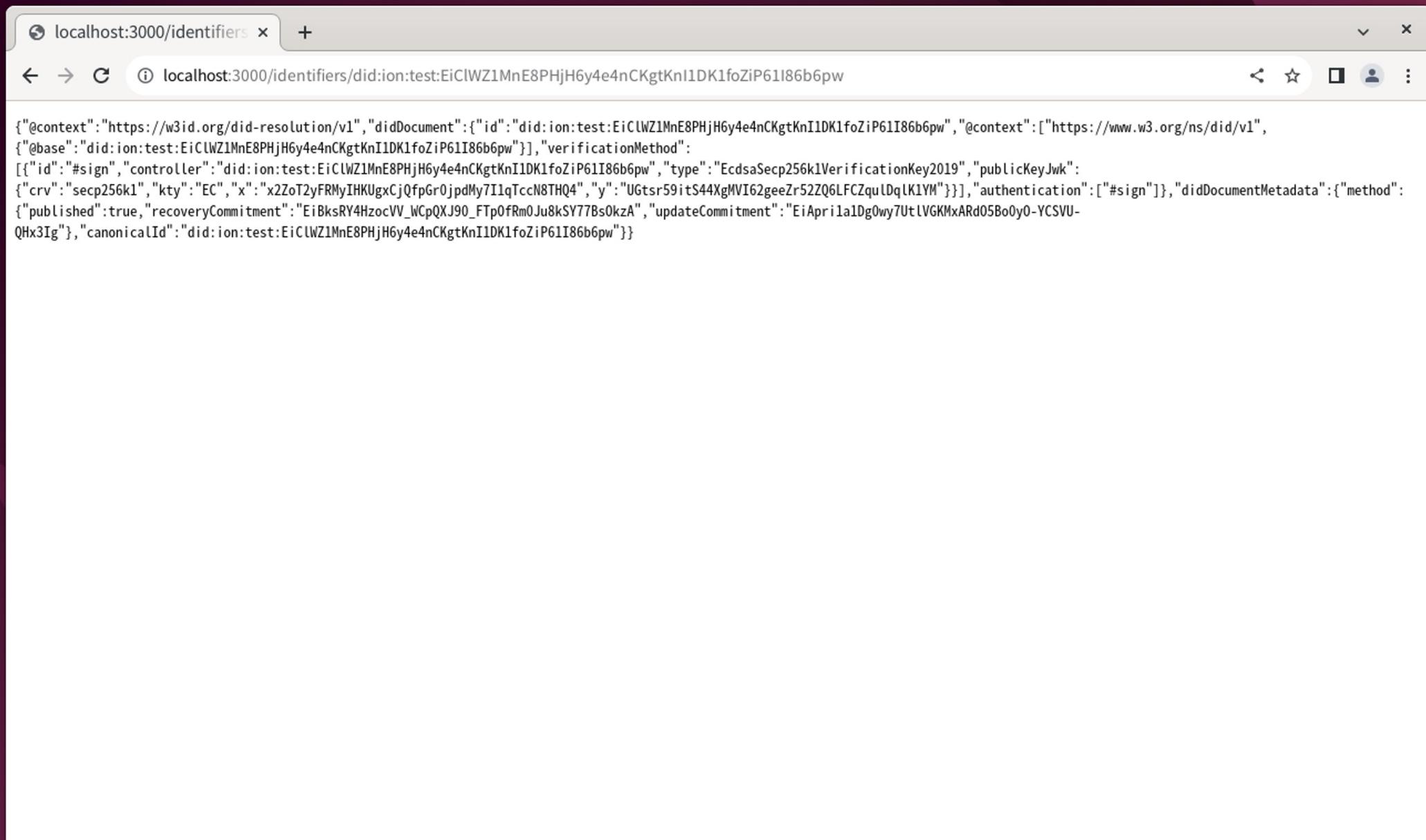
DIDの解決 (resolution) に成功した時のIONのログ

+ ▲ ×

bash
./bitcoin-0.18.0/bin/bitcoind...
npm ion-1.0.2
npm ion

行 1、列 1 スペース: 2 UTF-8 LF {} JSON 🔍 🔍

42



IONに触れてみて

- ・ 環境構築中にさまざまなエラーが発生してトラブルシューティングに時間を要した
- ・ 開発段階でドキュメントが揃っていない
- ・ DIDの利用には時間的・経済的なコストがかかる
 - Bitcoinの初回同期に非常に時間がかかる
(今回利用したテストネットでは2, 3時間。メインネットではさらにかかる。)
 - IONの最小要件となるスペックが高い(メインネットではストレージ1TBなど)



誰もが利用できるサービスとはいえ、気軽に導入できる段階には至っていない

目次

1. はじめに
2. 背景
3. IONの利用
4. DIDの活用可能性と活用方法の検討
5. インターンシップの感想

PoC事例 - 大阪府豊能町

- ・ 2022年7月、日本初のDIDの商用サービス
- ・ デジタル商品券利用時に活用する「とよのんウォレット」など行政から個人が町内で利用するサービス全てにおいて「MyDID」での管理を可能にする
- ・ 基盤はパーミッション型ブロックチェーンHyperledger Iroha



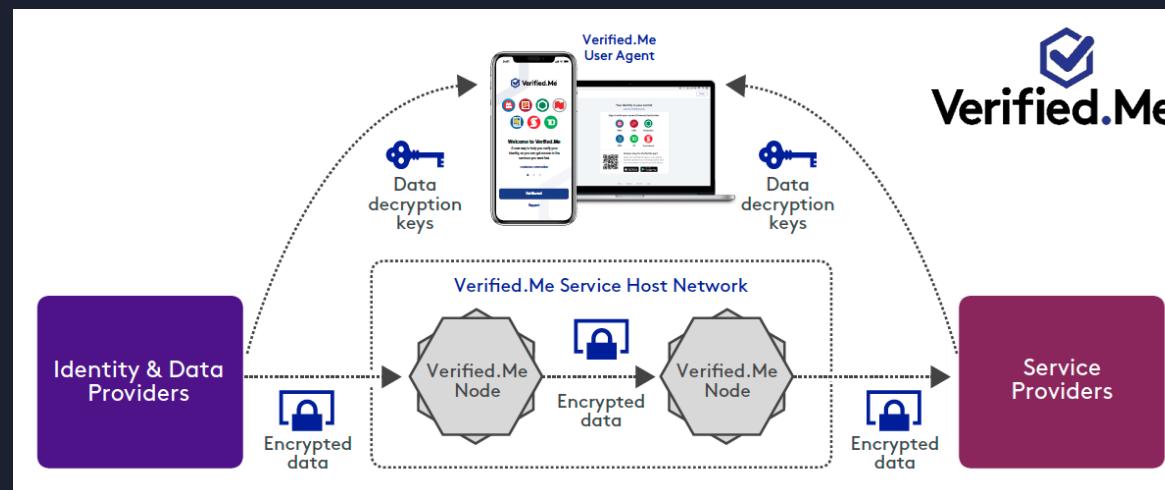
PoC事例 – 慶應義塾大学

- 2020年10月、慶應義塾大学がMicrosoftなどと連携して実証実験を開始
- 学生証、卒業証明書、学割チケットなどをスマホアプリに移行
- 就職活動を行う学生に対してスマホアプリでの卒業見込証明を発行し、採用企業に成績証明書や卒業見込証明書を提供
- 基盤のブロックチェーンに関する具体的な記述はなし



活用事例 – カナダ主要7銀行

- ・ 2019年5月からサービス提供
- ・ カナダの主要7銀行がコンソーシアムを組んでVerified.Meというサービス実現
- ・ 生命保険加入時に本人情報を連携でき、手続きが簡素化される
- ・ 基盤のブロックチェーンはHyperledger Fabric



DIDの活用/普及に向けた課題

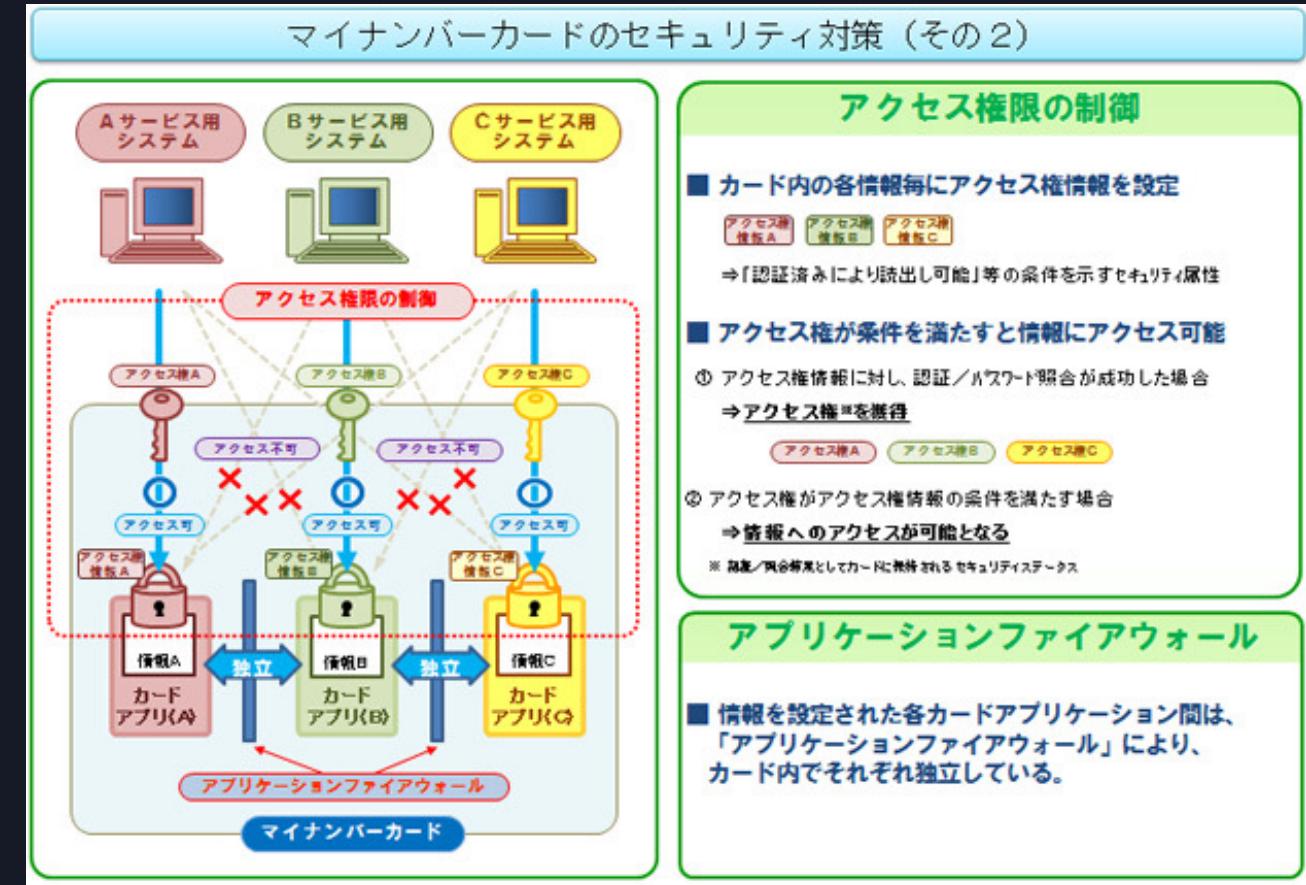
技術	法律	標準化
<ul style="list-style-type: none">• DIDを利用するまでの手順が多く、負担大• 技術者はブロックチェーンの同期、IPFSやMongoDBの立ち上げなど行わなければならぬ• エンドユーザはクレデンシャルを保管するウォレットの管理が必要• 普及のためにはUIの複雑さ解消が必要	<ul style="list-style-type: none">• 分散型アイデンティティに法的効力を持たせるための法整備	<ul style="list-style-type: none">• 異なるDIDプロダクト間での相互運用性• 相互運用性の低さがDID活用の妨げとなる

DIDの活用可能性の検討

- マイナンバーカードのデジタル化
- SSIの考え方
 「Holderが自分の属性情報のコントロール権を確保し、信頼できる組織から発行された本人の属性情報を取得し、ユーザの許可した範囲でVerifierに提示」



マイナンバー制度とSSI/DIDはセキュリティ面で考え方が近い



目次

1. はじめに
2. 背景
3. IONの利用
4. DIDの活用可能性と活用方法の検討
5. インターンシップの感想

インターンシップの感想

- ・これまで触れてこなかったブロックチェーンの拡張領域（Layer 2）の技術を体験し、ブロックチェーン技術について理解が深まった。
- ・実証段階の分野ということもあり、ドキュメントや情報の不足、一部のサービスの終了などでDIDの発行ができなかった。
- ・古いバージョンのソフトをインストールしなければならないなど、バージョンによる縛りも大きく、エラー処理に苦労した
- ・社員の方のお話や施設見学、定例ミーティングへの参加を通じて、セコムIS研究所の取り組みや研究分野、研究の進め方などを知ることができた
- ・IS研究所で現地参加したい気持ちもあったが、つくば市からの参加だったのでオンラインで参加させて頂けてありがたかった。

参考文献

- [1]日本経済新聞「ブロックチェーンとは データ改ざんリスク低く」2020年3月9日
<https://www.nikkei.com/article/DG XKZ056546400Y0A300C2NN1000/>
- [2]小笠原寿仁「レイヤー構造から見たブロックチェーン」2018年12月27日. <https://onl.sc/6Cfuwxw>
- [3]NTTドコモ「シングルサインオンとは」<https://www.ntt.com/bizon/glossary/e-s/sso.html>
- [4]野村総合研究所「ブロックチェーン技術等を用いたデジタルアイデンティティ の活用に関する研究」
https://www.fsa.go.jp/policy/bgin/ResearchPaper_NRI_ja.pdf
- [5]Microsoft 「ION – Booting up the network」 <https://techcommunity.microsoft.com/t5/identity-standards-blog/ion-booting-up-the-network/ba-p/1441552>
- [6]Digital Platformer 株式会社「Digital Platformer、大阪府豊能町で日本初分散型ID（DID）の商用サービス開始とデジタル商品券第2弾」
<https://prtmes.jp/main/html/rd/p/000000020.000059855.html>
- [7]慶應義塾大学「慶應義塾大学、次世代デジタルアイデンティティ基盤の実証実験を開始」<https://www.keio.ac.jp/ja/press-releases/files/2020/10/26/201026-1.pdf>
- [8]DIACC 「DIACC Identity Networks Paper」 https://diacc.ca/wp-content/uploads/2020/05/DIACC-Identity-Networks-Paper-Self-Assessment_SecureKey-VerifiedMe.pdf
- [9]総務省「マイナンバーカード」https://www.soumu.go.jp/kojinbango_card/03.html
- [10] Web3 PRESS 「IPFSとは？ブロックチェーンや仮想通貨・Web3.0との関連性、事例、今後について初心者にもわかりやすく解説！」
<https://tottemoyasashiibitcoin.net/entry/2021/11/11/2104>
- [11] Yıldız, H., Küpper, A., Thatmann, D., Göndör, S., & Herbke, P. (2022). A Tutorial on the Interoperability of Self-sovereign Identities. arXiv preprint arXiv:2208.04692.