

デジタル庁御中

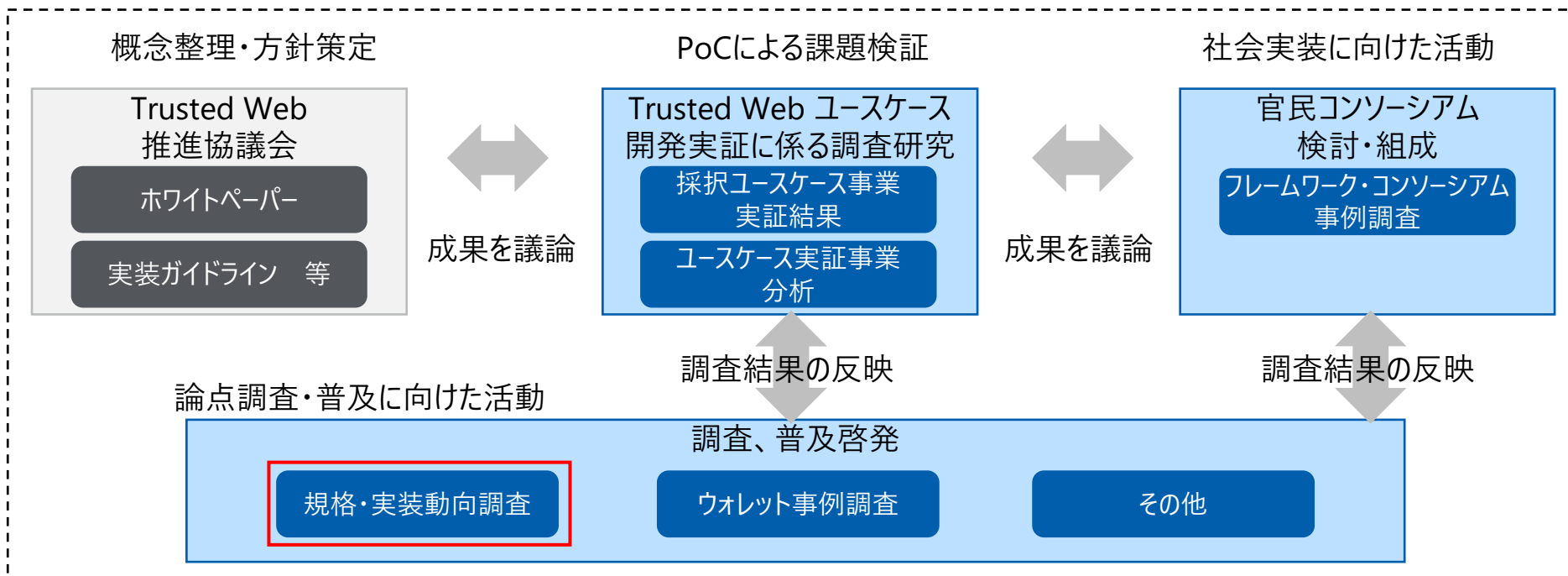
令和4年度補正
Trusted Web 開発等推進事業に係る調査研究
(規格・実装動向調査)

令和6年3月
TOPPAN株式会社

本書の位置づけ

- 本事業は、昨年度事業である13件のユースケースの開発実証等や、内閣官房デジタル市場競争本部事務局において活動を進めている「Trusted Web 推進協議会」におけるTrusted Web ホワイトペーパー策定等の活動、他検討結果を踏まえて、デジタル庁の委託のもと以下の業務を実施
 - ① Trusted Web ユースケース開発実証に係る調査研究
 - ② 官民コンソーシアム検討・組成
 - ③ 調査・普及啓発
- 本報告書は、③調査・普及啓発の中で、規格・実装にかかる動向調査をとりまとめたものである

 : 本事業で実施する業務
 : 本事業の成果報告書
 : 本報告書



目次

1. 背景・目的	P.3
2. 調査アプローチ	P.4
3. 調査まとめ	P.5
4. 規格実態調査	
4.1. 規格全体感	P.6
4.2. 各規格の概要・代表的な仕様	P.7
5. 規格詳細調査	
5.1. 調査対象	P.16
5.2. 規格・実装方式の詳細	P.17
6. 実装方式マッピング	
6.1. サービス	P.33
6.2. ライブラリ	P.44

1. 背景・目的

背景

- 近年、アイデンティティにかかるデータ主権を個人に帰属させることや、単一障害点の排除等の観点で、分散型アイデンティティが注目されており、W3C(World Wide Web Consortium)で規格化されたVC(Verifiable Credentials)やDIDs (Decentralized Identifiers)の技術を活用したサービスの実装検討が増えてきている
- これらの規格は、DIF(Decentralized Identity Foundation)や、Hyperledger、OpenID Foundation等の国際団体で標準化に向けた取組が進められている。また、EU・カナダ等の国・地域では、アイデンティティサービス提供にかかるフレームワーク等を策定してVC/DIDsを活用したサービスの相互運用性を高める取組が進められている
- ただし、現時点ではこれらの実装方法は多様であり、どのように収束していくかを把握するのが課題である
- 今後本邦においても相互運用性が確保されたサービスを提供していくためにも、現時点の規格動向や、規格を普及させるための取組がどうなっていて、各主要な団体がどのような実装を検討されているかを把握することが求められる

目的

- 本調査では、VC/DIDs等を活用したアイデンティティサービスにかかる技術スタックにおける規格の動向を調査し、実装パターンがどの程度あるかを把握する※
 - 規格実態
 - 技術スタック詳細
 - 実装パターンの抽出

※ 本調査は2024年3月までの既知の情報を元に記載している。また、実装パターンを紹介しているが本領域は発展途上の段階であり、各技術および組合せに関する安全性、信頼性を担保していない

※ 本調査はデジタル庁の意見を表明しているものではない

2. 調査アプローチ

	規格実態調査	技術スタック詳細調査	実装パターンの抽出
調査で明らかにしたい論点	<ul style="list-style-type: none"> VC/DIDs等を活用したアイデンティティサービスにどのような規格があるか 	<ul style="list-style-type: none"> 左記規格の技術スタックにはどのような特徴があるか 	<ul style="list-style-type: none"> VC/DIDs等を活用したアイデンティティサービスについてどの程度の実装パターンがあげられるか
調査内容	<ul style="list-style-type: none"> VC/DIDs等を活用したアイデンティティサービスにかかる規格の概要調査 <ul style="list-style-type: none"> 各レイヤの技術スタック概要 代表的な技術仕様 	<ul style="list-style-type: none"> VC/DIDs等を活用したアイデンティティサービスにかかる技術スタックの詳細調査 <ul style="list-style-type: none"> 技術スタック概要 (複数手法がある場合) 比較整理 VCモデルとmDLモデルの比較 	<ul style="list-style-type: none"> サービスの概要調査・実装パターン整理 ライブラリの概要調査・ライブラリ活用パターン整理
調査内容	デスクトップ調査		

3. 調査まとめ

規格実態調査

- DIFで整理されている「Interoperability Mapping Exercise」をもとにレイヤ別(「Credential Layer」、「Agent Layer」、「Public Trust Layer」の「Vertical / Cross-cutting」)にデータフローのイメージ、規格概要、代表的な規格を取りまとめた
- 業界ごとのVCデータモデルの検討状況を整理した
- ISOで検討されているmDL関連規格(18013シリーズ/23220シリーズ)の各項目で検討されている規格概要・規格ステータスを整理した

技術スタック
詳細調査

- Credential Layerでは、証明書フォーマット・証明書の検証モデル・証明書に記載されている情報の選択的開示に関する規格の概要・比較整理を行った
- Agent Layerでは、通信プロトコル・デバイス連携に関する規格の概要・比較整理を行った
- Public Trust Layerでは、DID Documentを活用した場合/しなかった場合の証明書の検証方法、DID Documentを活用した場合のストレージ比較、名前解決方法、データ格納庫等について整理した
- mDL・VCの特徴(データモデル・アーキテクチャ・検証フロー)の整理を行った

実装パターンの
抽出・事例調査

【サービス・フレームワーク】

- 実際に普及しているサービス・ガイドラインのサービス概要・実装されている技術スタック等を調査し、その中から現時点で考える実装パターンを分析した

【ライブラリ】

- 実際に普及しているライブラリの概要・提供形態・対応範囲等を調査し、事業者が実装する場合にどのライブラリを活用すべきかの分析を行った

4.1. 規格全体観

- DIFで整理されている「Interoperability Mapping Exercise」をもとに技術スタックの概要や代表的な規格を取りまとめた

Application Layer : ユーザアプリケーション

Credential Layer : 証明書フォーマット、交換、バインディング

証明書
フォーマットCredential
Format証明書
正当性証明Credential
Proofing

証明書失効

Credential
Revocation証明書(VP)の要求と
提示用のフォーマットCredential
Exchange証明書
バインディングCredential
Binding

Agent Layer : 通信プロトコル、ストレージ、鍵アクセス

包装
(P2P通信)Envelope
(P2P-Communication)

伝送

Transport

制御リカバリ

Control Recovery

鍵操作

Key Operations

メタデータの
運搬性Meta data
Portability

Public Trust Layer : トラストアンカー、データレジストリ

DIDドキュメント/
スケーリングDID Document
/DID Scaling

DIDメソッド

DID Method

アンカータイプ

Anchor Types

DID解決

DID Resolution

アンカーサービス

DID-Anchored
Svc, EDV

Vertical / Cross-cutting

認証/認可

Authentication/A
uthorization選択的開示
/ゼロ知識証明Disclosure
/ZKP

コンプライアンス

Compliance

ストレージ

Storage

データフォーマット

Data Formats

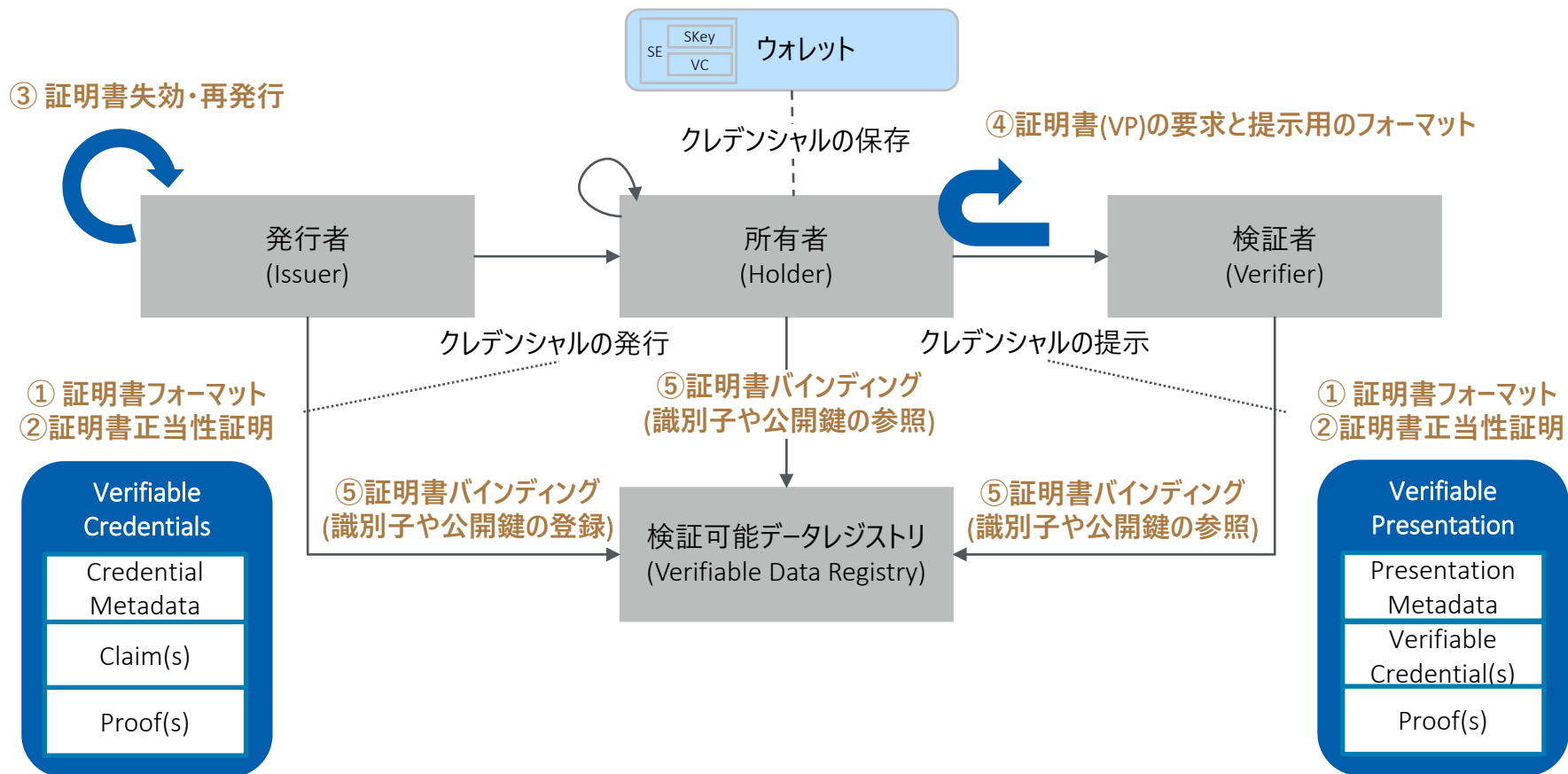
基本的な暗号方
式

Crypto Primitives

*出所 : <https://github.com/decentralized-identity/interoperability/blob/master/assets/interoperability-mapping-exercise-10-12-20.pdf>

4.2. 各規格の概要・代表的な仕様 - (1) Credential Layer (1/2)

- Credential Layerは、主に、証明書のフォーマット、失効、交換方式に関する規格群が定義されている
- 証明書連携フローについてCredential Layerで行われていること概要を図示すると以下ようになる



4.2. 各規格の概要・代表的な仕様 - (1) Credential Layer (2/2)

■ (続き)

分類	概要	(参考)代表的な技術仕様のURL
①Credential Format	<ul style="list-style-type: none"> 検証可能な資格情報のデータモデルを定義 代表的な規格としてW3Cが策定したVerifiable Credential、Verifiable Presentationや、OpenID Foundationが策定し、一般的な認証システムでも使用されるOpenID Connect ID Tokenが挙げられる 	Verifiable Credentials / Presentation (W3C) (Ver.1.1) (Ver.2.0) OID4VCI (OpenID Foundation) OID4VP (OpenID Foundation)
②Credential Proofing	<ul style="list-style-type: none"> 証明書の正当性を示すデータ形式を定義、選択的的属性開示とも関連が深い 代表的な規格としてVC JSON-LD Proofs、SD-JWT VC、AnonCreds等が挙げられる 	VC-JWT (W3C/DIF) VC JSON-LD Proofs (W3C) SD-JWT VC (IETF) AnonCreds (Hyperledger)
③Credential Revocation	<ul style="list-style-type: none"> 証明書の失効管理方法について定義 代表的な規格としてx.509のOnline Certificate Status ProtocolやCertificate Revocation List、VC Status Revocation List、AnonCreds Revocation Status List v1/v2等が挙げられる 	X.509 (IETF) VC Status Revocation List (W3C) AnonCreds(Hyperledger)
④Credential Exchange	<ul style="list-style-type: none"> 証明書を所有者が検証者に提示する際の交換方式を定義 クレデンシャル(VP)の要求と提示に関するフォーマット 代表的な規格としてDIFのPresentation Exchange、W3CのVerifiable Presentation RequestやHyperledgerで検討しているPresent Proof Protocol等が挙げられる 	VP Request (W3C) Presentation Exchange (DIF) OIDC Credential Provider (OpenID Foundation) Aries Present Proof v2
⑤Credential Binding	<ul style="list-style-type: none"> 証明書等に記載されている署名情報を検証する公開鍵と特定の識別子が関連していることを表す情報群を定義 (例えば、証明書の真正性を保証するための署名を検証する公開鍵がどこにあるか示すものや、ユーザのアイデンティティを証明する情報群を提供) 代表的な規格としてW3Cで定義しているDecentralized IdentifiersやOpenID ConnectのID Token等が挙げられる。 	DID (W3C) OpenID Connect id token (OpenID Foundation) Link Secrets(CL-RSA)








(参考) VCデータモデルの標準策定状況

■ 学歴証明、製品の品質保証、ワクチン証明等でVCのデータモデル整備が進んでいる

VCのユースケース*

大学卒業証明	<ul style="list-style-type: none"> 大学卒業資格を定義 (学士・博士等の課程や学位) 	原油取引	<ul style="list-style-type: none"> 原油の品質や運搬・貿易にかかる記録等を定義
認定ミル	<ul style="list-style-type: none"> 金属等を用いて作られた製品が特定の規格に準拠した品質で生産されていることを保証された証明書の定義 	Covid-19	<ul style="list-style-type: none"> Covid-19の接種証明書の定義

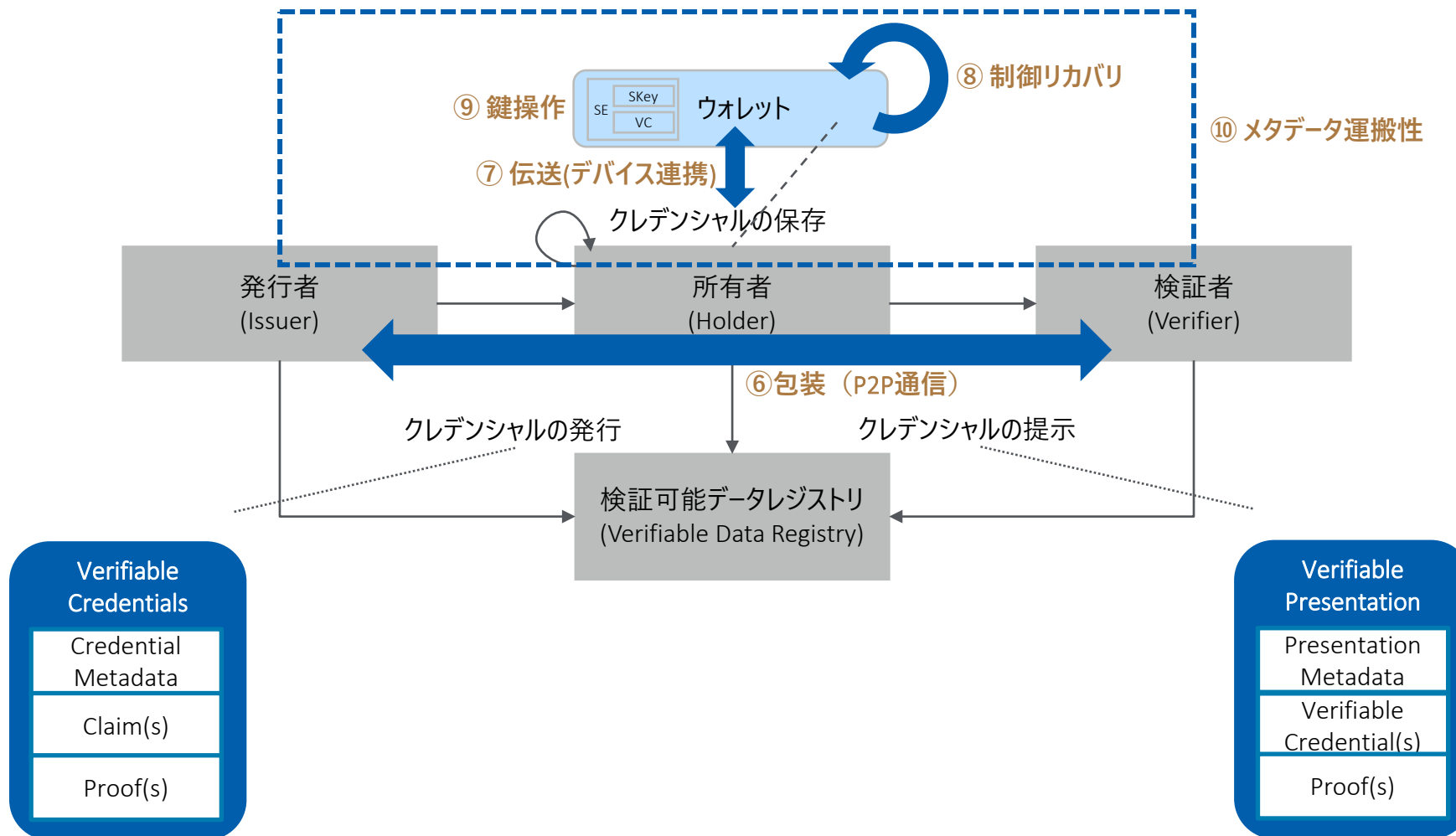
その他検討されているユースケース**

 Education <ul style="list-style-type: none"> Digital transcript Taking a test Transferring schools Online classes 	 Retail <ul style="list-style-type: none"> Address verification Adult beverages Fraud detection 	 Finance <ul style="list-style-type: none"> Reuse know your customer Money transfer Closing account Trying out a new service New bank account from home 	 Healthcare <ul style="list-style-type: none"> Prescribing Online pharmacy Insurance claim Traveling illness Proving Legal Disability Status
 Professional Credentials <ul style="list-style-type: none"> Find a doctor Busy doctor Bad university New employer Social authority Job applicant 	 Legal Identity <ul style="list-style-type: none"> Digital driving license Seamless immigration Speedy air travel Refugee crisis 	 Devices <ul style="list-style-type: none"> Devices during manufacturing Devices during delivery Devices setup for operating autonomously 	

出所 * <https://w3c-ccg.github.io/vc-examples/>** <https://www.w3.org/TR/vc-use-cases/>

4.2. 各規格の概要・代表的な仕様 - (2) Agent Layer (1/2)

- Agent Layerは、主に、エージェント間のセキュアなPeer-to-Peerコミュニケーションを実現するための証明書の検証方法や連携方法、および鍵情報へのアクセス方式等の規格群が定義されている
- 証明書連携フローについてAgent Layerで行われていること概要を図示すると以下のようなになる



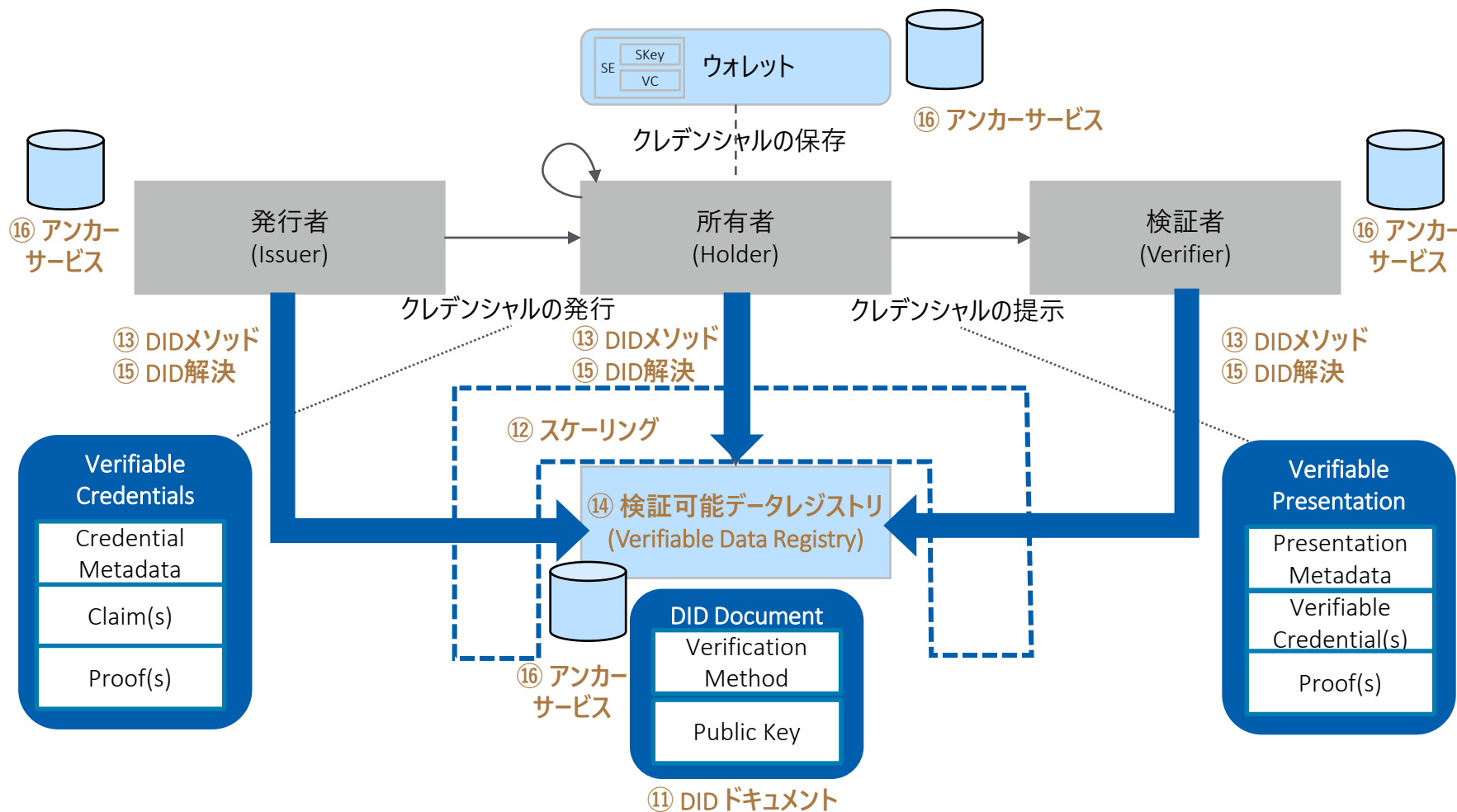
4.2. 各規格の概要・代表的な仕様 - (2) Agent Layer (2/2)

■ (続き)

分類		(参考)代表的な技術仕様のURL
⑥ Envelope (P2P- Communication)	<ul style="list-style-type: none"> エージェント間で証明書を連携する際の通信プロトコル、エンコード方式等を定義 代表的な規格としてIETFが定義しているJSON Web Message、DIFで定義しているDID CommやSelf-Issued OpenID Connect Provider DID Profile v0.1等が挙げられる 	<ul style="list-style-type: none"> • DIDComm Messaging v2 • Self-Issued OpenID Provider v2(SIOPv2)
⑦ Transport	<ul style="list-style-type: none"> 異なるデバイス間で証明書やアイデンティティ情報の格納先などを連携する方法を定義 代表的な規格として近距離無線通信規格のNFCやBluetooth、QRコード、HTTP等が挙げられる 	<ul style="list-style-type: none"> • OAuth 2.0 Device Flow • OpenID Connect CIBA Flow • OID4VP(Cross Device Flow) • SIOPv2(Cross-Device Self-Issued OP) • CTAP v2.2(Hybrid transports)
⑧ Control Recovery	<ul style="list-style-type: none"> 証明書やそれに関連する鍵情報などを紛失した場合のリカバリ方法について定義 代表的な規格としてブロックチェーンのウォレットのバックアップ用として用いられるBIP-39や生体認証と組み合わせてバックアップするHorcrux Protocol等が挙げられる 	<ul style="list-style-type: none"> • BIP-39 • DKMS • Horcrux Protocol • Universal Wallet 2020
⑨ Key Operations	<ul style="list-style-type: none"> 鍵情報をセキュアに格納する方式について定義 代表的な規格としてハードウェアに鍵を格納するCloud/Local HSMやTEE Chips等が挙げられる 	<ul style="list-style-type: none"> • HSM(cloud/local) • Secure Element • TEE Chips • Smart Cards
⑩ Meta data Portability	<ul style="list-style-type: none"> エージェント間でデータを運搬する方式について定義 代表的な規格としてW3Cで定義しているUniversal Wallet 2020が挙げられる 	<ul style="list-style-type: none"> • Universal Wallet 2020

4.2. 各規格の概要・代表的な仕様 - (3) Public Trust Layer (1/2)

- Public Trust Layerは、主に分散型IDのトラストの基盤となるプラットフォームや情報の格納形式、アクセス方式が定義されている
- 証明書連携フローについてPublic Trust Layerで行われていること概要を図示すると以下ようになる



4.2. 各規格の概要・代表的な仕様 - (3) Public Trust Layer (2/2)

■ (続き)

分類	概要	(参考)代表的な技術仕様のURL
⑪ DID Document	<ul style="list-style-type: none"> 証明書の発行元を証明する署名検証のための各種情報を格納したデータモデルやそれを活用した検証方法、インターフェイスを定義 代表的な規格としてDID Documentが挙げられる。 	Decentralized Identifiers (DIDs) v1.0
⑫ DID Scaling	<ul style="list-style-type: none"> データレジストリに用いられるブロックチェーンのスループット改善の方式等を定義 代表的な規格としてDIFのSidetree ProtocolやKERI (Key Event Receipt Infrastructure) が挙げられる 	Sidetree v1.0.1 KERI
⑬ DID Method	<ul style="list-style-type: none"> 分散型アイデンティティシステムの相互運用性を高めるための、各方式を識別する方法を定義 代表的な規格としてW3Cが定義しているDID Methodが挙げられる。 DID Methodにて各方式ごとに定義されており、Ethereumを用いるdid:ethr、Hyperledgerで検討されているSovrin Networkを用いるdid:sov、データレジストリを挟まずP2Pで直接やり取りするdid:key等がある 	Decentralized Identifiers (DIDs) v1.0 DID Specification Registries
⑭ Anchor Types	<ul style="list-style-type: none"> 各エンティティから信頼され、耐改ざん性があるデータレジストリの具体的な実装基盤を定義 代表的な規格としてHyperledger Indyを用いるプライベートチェーンであるSovrin Networkや、EthereumやBitcoinなどのパブリックチェーン等が挙げられる 	Hyperledger Indy (Sovrin Network等) ION Network (Bitcoin) Element (Ethereum)
⑮ DID Resolution	<ul style="list-style-type: none"> DID Methodを参照してデータレジストリを特定するための名前解決の仕組みを定義 代表的な規格としてW3Cが定義しているDecentralized Identifier Resolution (DID Resolution) v0.3やDIFで定義しているUniversal Resolverが挙げられる 	Decentralized Identifier Resolution (DID Resolution) v0.3 Universal Resolver
⑯ DID-Anchored SvcS	<ul style="list-style-type: none"> 鍵情報やDID Documentを格納するためのセキュアなストレージサービスを定義 代表的な規格としてDIFが定義しているIdentity HubやDigital Bazaarが定義しているEncrypted Data Vaults (EDV) が挙げられる 	Identity Hub Encrypted Data Vaults 0.1 Decentralized Web Node

4.2. 各規格の概要・代表的な仕様 - (4) Vertical / Cross-Cutting

■ Vertical / Cross-cuttingでは、暗号方式や選択的開示方式等のレイヤーを横断する要素技術・取組等について記載されている

分類	概要	(参考)代表的な技術仕様のURL
Authentication /Authorization	<ul style="list-style-type: none"> 分散型アイデンティティと連携する認証/認可システムを示す 代表的な規格としてW3CのWebAuthnやOpenID Foundationの、OpenID Connect、SIOP等が挙げられる 	Web Authentication
		User-Managed Access (UMA) Profile of OAuth 2.0
		Self-Issued OpenID Provider v2
Disclosure /ZKP	<ul style="list-style-type: none"> 所有者が検証者に提供する証明書を必要最低限の情報にすることでプライバシーを担保するため、証明書から開示する情報を選択できる方式を示す 代表的な規格としてHyperledger で定義しているAnonCreds ZKPsや、W3Cで規定しているBBS+ Signature等が挙げられる 	Anoncreds v1 ZKPs (Camenisch-Lysyanskaya)
		BBS+ signatures
Compliance	<ul style="list-style-type: none"> 分散型アイデンティティの実現する上で前提となる国や組織のコンプライアンスを示す 代表的な規格として欧州のGDPRやeIDAS等が挙げられる 	eIDAS
		GDPR
Storage	<ul style="list-style-type: none"> 各種データを格納するためのデータベース、分散型ストレージ等を示す 代表的な規格としてデータベースのMySQL、CouchDB、MongoDBや分散型ストレージのIPFS等が挙げられる 	Confidential Storage 0.1
		On-Chain (Public/Private)
		Web server
Data Formats	<ul style="list-style-type: none"> 各種データを表現するデータフォーマットを示す 代表的な規格としてJSON、XML、CBOR等が挙げられる 	JSON
		CBOR
Crypto Primitives	<ul style="list-style-type: none"> 分散型アイデンティティで使用される証明書の署名方式や暗号方式を示す 代表的な規格としてECDSA、EdDSAやsecp256k1が挙げられる。 	ECDSA
		EdDSA
		secp256k1

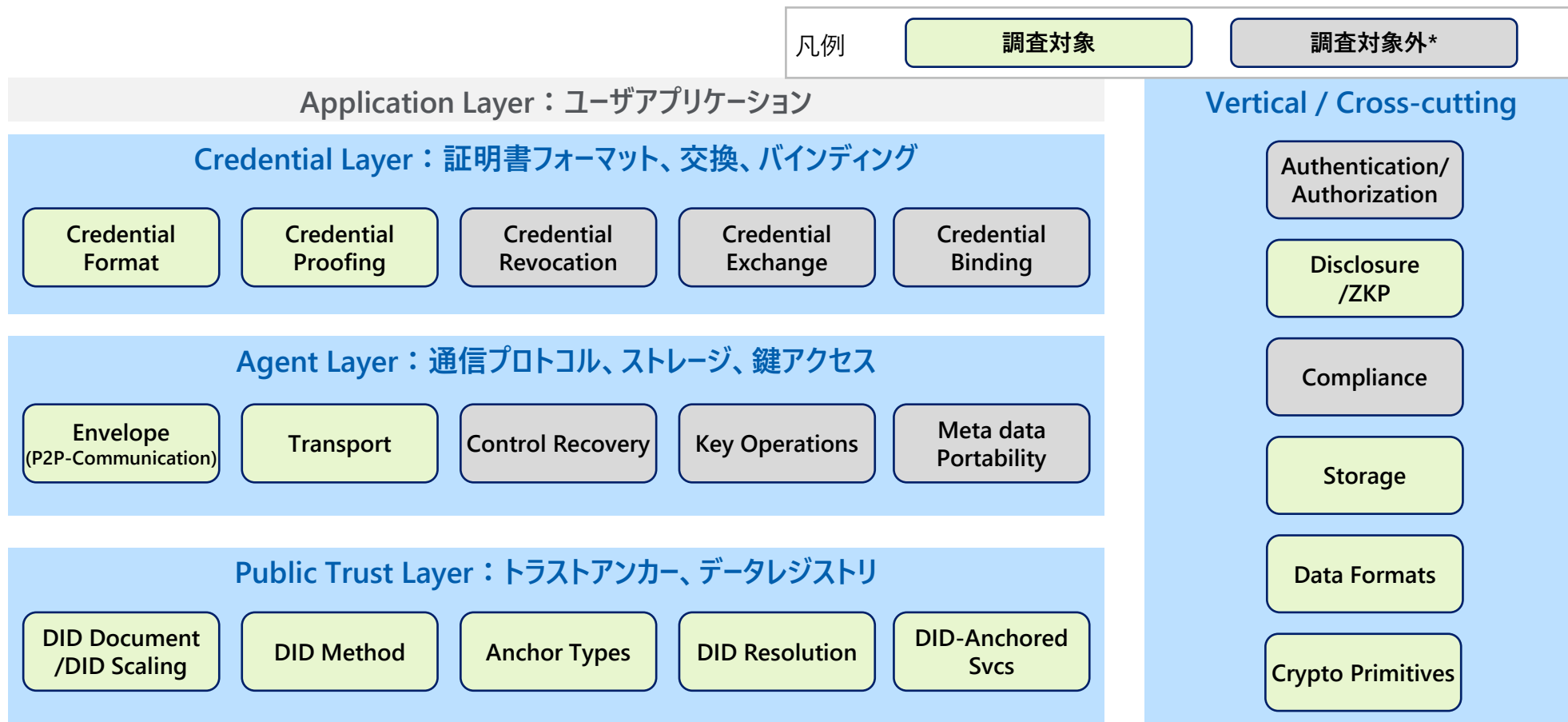
(参考) mDLに関する標準策定状況

- mDLはISO/IEC18013シリーズ、23220シリーズで標準化が進んでおり、一部規格は策定途中である

標準化項目			概要	ステータス
ISO/IEC18013 series (運転免許証にかかる標準化)	18013-1	Physical characteristics and basic data set	免許証の物理特性と基本的なデータセットを定義	Published
	18013-2	Machine-readable technologies	免許証の機械読取部分のデータ構造や読取方法を定義	Published
	18013-3	Access control, authentication and integrity validation	機械読取部分のアクセス制御、整合性検証方法を定義	Published
	18013-4	Test methods	適合性テストに使用される方法を定義	Published
	18013-5	Mobile driving licence (mDL) application	mDL実装のためのインターフェイス仕様を定義	Published
	18013-6	mDL test methods	mDLのテスト方法について定義	Under development
	18013-7	Mobile driving licence (mDL) add-on functions	mDLのアドオン機能を定義	Under development
ISO/IEC23220 series (デジタルIDにかかる標準化)	23220-1	Generic system architectures of mobile eID systems	モバイルID管理のアーキテクチャとライフサイクルを定義	Published
	23220-2	Data objects and encoding rules for generic eID systems	汎用IDのデータ構造とエンコード規則を定義	Under development
	23220-3	Protocols and services for issuing phase	発行フェーズの Protokol とサービスを定義	Under development
	23220-4	Protocols and services for operational phase	運用フェーズの Protokol とサービスを定義	Under development
	23220-5	Trust models and confidence level assessment	信頼モデルと信頼度評価を定義	Under development
	23220-6	Mechanism for use of certification on trustworthiness of secure area	セキュアエリア (=SE) の信頼度に関する認証利用の仕組み (日本から提案)	Under development

5.1. 調査対象

- 4章で整理した技術スタックのうち、現時点で仕様がある程度固まっているかつ実装パターンに影響が出る範囲を調査対象とした*
- 各技術要素で、実装手法が複数選択肢がある場合は主要なものの特徴を比較・整理した



*調査対象外とした理由

Credential Layer (Revocation / Exchange / Binding) : 仕様が十分に定まっていないため調査対象外

Agent Layer (Control Recovery / Key Operations / Meta data Portability) : ハードウェア仕様がまだ定まっていないため不確定要素が大きいため調査対象外

Authentication / Authorization : アプリレイヤーのため、相互運用性に影響ないため調査対象外

Compliance : 技術スタックとは関係ないため調査対象外

Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document / DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure / ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (1) 証明書を表現する際に利用されるデータフォーマットの規格

- コンピュータ黎明期には低容量で高速なバイナリが一般的であったが、次第に可読性の高いテキストに置き換わっていった
- 証明書の利用としてはJSON、JSON-LDがWebアプリで多用されるJavaScriptとの親和性が高く利用されている。また、リソースの小さいIoT機器等で利用する場合にはCBORの導入が進んでいる

	ASN.1	XML	JSON	JSON-LD	CBOR
名称	Abstract Syntax Notation One	eXtensible Markup Language	JavaScript Object Notation	JavaScript Object Notation Linked Data	Concise Binary Object Representation
概要	1988年にITUとISO共同で策定された通信プロトコル用のデータフォーマット	SGMLの拡張言語で、タグで挟むことで、データを表す。データ構造はXML Schemaで定義	JavaScriptのオブジェクト表記法のためフロントエンドでシームレスに利用できる。構造はJSON Schema等で定義	JSONを利用したデータ形式で、Web上のデータを構造化し、データ定義や関連性を明確に表現できる。	JSONと互換性があり、データフォーマット。認証器とクライアントを接続するプロトコルにも用いられる。構造はCDDLで定義
利用実態	(低データ量で処理が高速な)バイナリデータが特徴、黎明期において利用されていた	ASN.1よりは可読性が高く、JSON普及前に一般的に利用されていた	可読性が高く、フロントエンドとの接続と相性が良く、現在主流のフォーマット	Webページのデータを構造化することで、SEO対策に利用されている (Google検索で推奨)	IoT領域やスマートフォンのセキュリティ領域等低レイヤへの書き込みに利用されている
可読性	独自色が強く、バイナリ形式のため可読性は低い	テキスト形式であり、項目名のタグで囲むため、可読性が高いが容量が大きくなりやすい	テキスト形式であり、項目と値をコンパクトに定義できるため、可読性が高く容量が小さい	テキスト形式であり、特定のデータの意味付けや関係性を定義できるため、JSONと比較して表現度が高い	バイナリ形式のため可読性は低い、コンパクトにまとめることが出来るためリソースの小さいIoT等に適する
可用性	通信プロトコル等に使用されることが多く対応言語を選ぶ	ほぼ全ての言語でサポート	ほぼ全ての言語でサポート	JSON-LD固有のライブラリが必要、多くの言語で対応されている	サポートは増えつつあるが、現時点では中程度
パフォーマンス	高速に処理可能 (バイナリエンコードの場合)	構造解析に時間がかかる	バイナリ形式に劣るがXMLよりも高速に処理可能	JSONと比較して構造が複雑化しやすいがパフォーマンス的な差異は小さい	高速に処理可能
利用ケース	通信プロトコル(SMTP、LDAP)、データ交換	複雑な文書、SOAP、設定ファイル	Web API、設定ファイル、データ交換	検索エンジン、データ連携、セマンティックWeb	IoTデバイス、WebAuthn、CTAP
証明書利用	×	×	○ (主にVCで活用)	○ (主にSEO対策,VCで活用)	○ (主にmDLで活用)

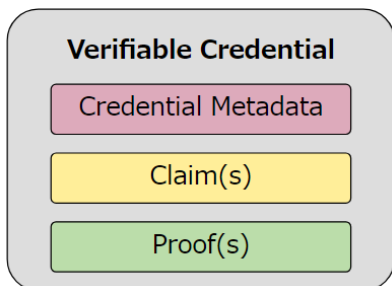
Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document /DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure /ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (2) アイデンティティサービスを構築する場合に基準とされる証書モデルの規格(1/2)

- Verifiable Credential (VC) と Verifiable Presentation (VP) は、デジタルアイデンティティを確認し、信頼性を担保するための概念で、これらはW3C (World Wide Web Consortium) により仕様が標準化*されている

VC (Verifiable Credential)

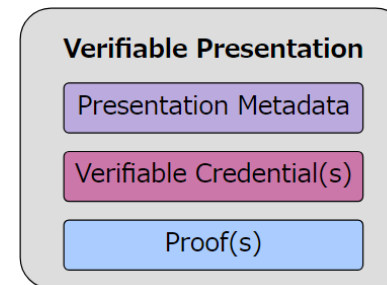
特定の主張（例えば、個人の名前や年齢、組織の所在地等）について、それが信頼できる発行者から発行されたものであることを証明するデジタル証明書的一种



W3Cで定義している
VCの基本構成

VP (Verifiable Presentation)

一つまたは複数のVCを保持し、それを他のエンティティに提示するためのパッケージ。VPは、提示者(通常はHolder)によってデジタル署名され、その署名で提示者とVCの真正性が確認できる



W3Cで定義している
VPの基本構成

- OpenID Foundationで策定が進められている、OpenID for Verifiable Credential (OID4VC) は以下の3つの仕様で構成されている**。W3Cで定義されている仕様はエンティティ間のプロトコルまで定義されておらず、実装者にゆだねられているため、プロトコルの標準化を進めている

OpenID for Verifiable Credential Issuance(OID4VCI)	<ul style="list-style-type: none"> 検証可能な資格情報（VC）を発行するためのAPIと対応するOAuthベースの認定メカニズムを定義
OpenID for Verifiable Presentations(OID4VP)	<ul style="list-style-type: none"> OAuth2.0上にプロトコルフローの一部として検証可能な資格情報の形式でクレームを定義できるようにするメカニズムを定義
Self-Issued OpenID Provider v2	<ul style="list-style-type: none"> 自己主権型アイデンティティとしてエンドユーザーが自分で管理するOpenIDプロバイダーを使用できるようにする

Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document / DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure / ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (2) アイデンティティサービスを構築する場合に基準とされる証明書モデルの規格(2/2)

- 現時点でアイデンティティサービスを構築する場合、基準とされる証明書モデルの規格は、W3Cで標準化されたVC/VPと、OpenID Connectの通信/認証プロトコルに対応しているOID4VCI/OID4VPの2つに大別される

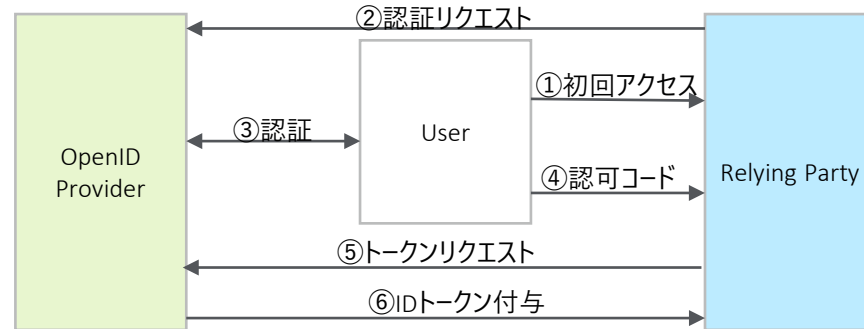
証明書モデル比較

規格主体		W3C	OpenID Foundation
規格・特徴	VC	W3C-VC <ul style="list-style-type: none"> 運転免許証や学歴証明書、資格証明書、その他の機密データなどの物理的に存在する個人の属性を表す情報をデジタル化し、オンライン上で検証可能にした証明書 	OID4VCI <ul style="list-style-type: none"> 既存のOAuth 2.0実装とOpenID Providerのサービスを拡張し、Verifiable Credentialを発行することが可能(新規フローなので標準化に時間を要す)
	VP	W3C-VP <ul style="list-style-type: none"> Verifiable Credentialからのデータを含み、検証者に共有するデータ形式 VCの保持者が相手に情報の内容まで知られたくない場合にゼロ知識証明などによって選択的な開示が可能 	OID4VP <ul style="list-style-type: none"> OAuth2.0及びOpenID Connectのプロトコルフロー上において、検証可能なプレゼンテーションの形式で要求を提示するメカニズムを定義 既存のOIDCのフローに近く、標準化の検討が進んでいる

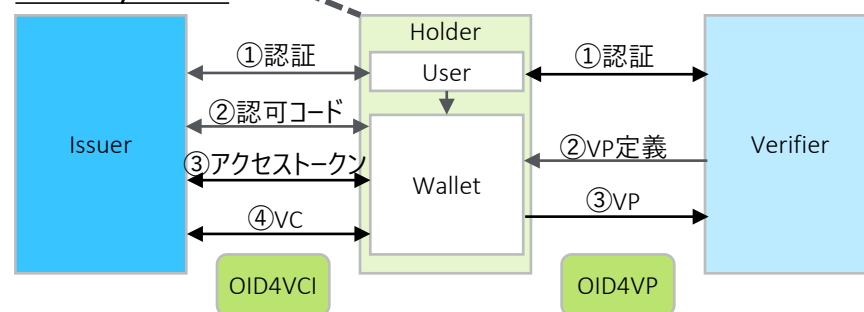
OID4VCI/OID4VPの拡張イメージ

OID4VPは既存のOpenID Connectのフローと類似しており(アクセストークン付与のプロセスにVPを追加するイメージ)、標準化が進んでいるが、OID4VCIは新規フローであるため、比較的時間を要している

従来のOpenID Connect



OID4VCI/OID4VP



Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document / DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure / ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (3) 選択的開示含む証明書検証のための規格

- 証明書のデータフォーマットとしては、JSONが大部分を占めており（一部JSONをバイナリ形式も存在）選択的開示を可能にする拡張仕様が複数規格が存在している

証明書規格	データフォーマット	通常採用される署名アルゴリズム*	特徴	選択的属性能開示	
				データ最小化	ゼロ知識証明
JWT VC (JSON Web Token)	JSON JSON-LD	ECDSA EdDSA	<ul style="list-style-type: none"> OAuth2.0およびOpenID Connet等のフレームワークで認証トークンとして広く使用されているJWTをベースにしているため、開発者が実装する上で必要なライブラリやツールが豊富に存在する プレーンJSONを用いたJWTベースのVCは暗号化とセキュリティのために既存のJOSEフレームワークを使用しており、コンテキストをうまく表現できないため、選択的属性能開示機能の実装が困難 	×	×
SD-JWT VC	JSON JSON-LD	ECDSA	<ul style="list-style-type: none"> 選択的開示を実現、実データにソルトを加えて生成したハッシュ群に対し署名を行うことで、実データ自身を送付せずに、署名検証できる 	○	×
LDP-VC (Linked Data Proofs)	JSON-LD	BBS+	<ul style="list-style-type: none"> プレーンJSONでは表現が困難なコンテキストを定義可能な形式。Linked Dataによりデータの記述に使用する用語（URI）を一意に特定できる 選択的開示はBBS+署名・ゼロ知識証明で実現する 	○	○
AnonCreds	AnonCreds (JSON)	CL	<ul style="list-style-type: none"> Hyperledger AnonCredsプロジェクトで検討している形式 選択的開示はゼロ知識証明+CL(Camenisch-Lysyanskaya)署名を使用する 	○	○
mDL	CBOR	ECDSA	<ul style="list-style-type: none"> 運転免許証の文書形式(mdoc)を拡張して、他ユースケースでも活用できるようにしたもの 	○	△**

*出所：<https://github.com/vcstuff/credential-profile-comparison?tab=readme-ov-file>**mDLにゼロ知識証明を活用した事例は現在確認例が少なく一般的でないが、CYBERNETICA社の取組を確認している https://cyber.ee/uploads/Zero_Knowledge_Proofs_report_89d6bc5438.pdf

Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document / DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure / ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (4) 選択的開示方式の規格

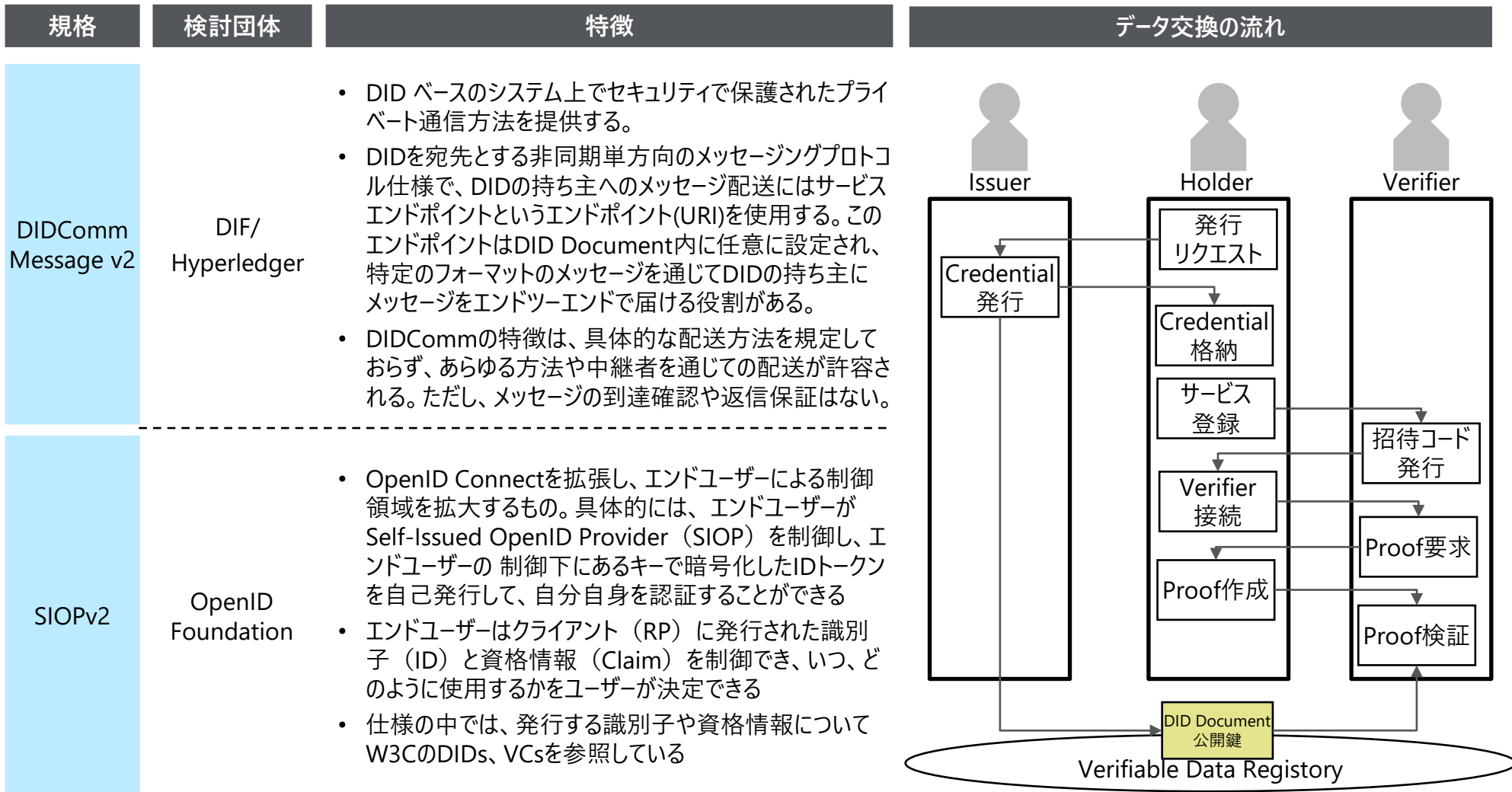
- 選択的属性開示手法として適用のし易さを重視したSD-JWT、厳密性と拡張性を重視したJSON-LD + BBS+署名が代表的

	SD-JWT	JSON-LD + BBS+署名
概要	<ul style="list-style-type: none"> 実データにソルトを加えて生成したハッシュ群に対し署名したSD-JWTと、実データを格納したSVCを用意 SVCはそのままHolderが保管し、Verifierに送付する際にSVCのClaimを選択し、Holderの署名をつけて開示する SD-JWT-Rのデータのハッシュ値と、SD-JWTに格納された値の同一性を確認することで真正性を検証 	<ul style="list-style-type: none"> マルチメッセージのデジタル署名の一種。通常、秘密鍵を使用してメッセージ全体を署名するが、マルチメッセージシステムでは、秘密鍵で署名されたメッセージをより小さな属性に分割して共有および検証できる。 そのため、Holderは署名されたデータを項目ごとに切り出して開示することが可能。ゼロ知識証明との組み合わせが可能。
イメージ	<p>分割単位にハッシュ化</p> <p>署名(Issuer)</p> <p>SVC</p> <p>元データを別に切り出し</p> <p>C1, S1 C2, S2 C3, S3</p> <p>SD-JWT-R</p> <p>C1, S1 C3, S3</p> <p>署名(Holder)</p> <p>開示項目のみ切り出して署名</p>	<p>定義+項目単位に変換</p> <p>@context + C1 @context + C2 @context + C3</p> <p>署名(Issuer)</p> <p>項目をグループ化して署名</p> <p>項目単位に切り出しても検証可能</p> <p>Holderにより創出された署名(Issuer)'</p> <p>署名(Holder)</p> <p>元のIssuerのProofは開示されない。</p>
評価	<ul style="list-style-type: none"> ユーザーが毎回同じ署名を利用するため、複数の検証処理を照合することで利用者特定が可能であり、Unlinkabilityを担保できない Webシステムのトークン仕様として、広く普及しているフォーマットのため、システム実装者の理解度、既存システムへの適用性が高い 	<ul style="list-style-type: none"> ゼロ知識証明と組み合わせることで、署名検証の都度新しい証明を作成できるため、Unlinkabilityを担保できる 新しいフォーマット仕様であるため、普及には時間がかかる可能性が高い

Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document /DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure /ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (5) エンティティ間で証明書データを連携する規格

- DIDCommは一般的なメッセージングの仕様を定めており、SIOPv2は認証プロセス全体を規定している



Credential Layer		Agent Layer		Public Trust Layer					Vertical / Cross-cutting			
Credential Format	Credential Proofing	Envelope	Transport	DID Document /DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure /ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (6) デバイス連携のための規格

- デバイス連携の手法としては、QRコード/NFC/Bluetooth (BLE) 等の複数デバイス間 (Cross Device) の連携と、同一デバイス内 (Same Device) に大別される
- QRコードは付属カメラ等で情報を取り込むため特別な拡張は不要。BLEやNFCは通信プロトコルレベルの拡張が必要になる。従って、具体的な実装には拡張仕様もしくは業者が独自に設計する必要がある (OID4VPにおいては、VP交換のBluetooth拡張機能が存在)。同一デバイス内であればアプリ間での標準的な遷移で対応可能だが、どの方式においてもプライバシーを保護する適切な制御が必要となる。

OID4VP	1	Same Device Flow	<ul style="list-style-type: none"> OID4VPを実行するソフトウェアとウォレットが同一デバイス上に存在する場合にアプリケーション間でリダイレクトする方式を定義
	2	Cross Device Flow	<ul style="list-style-type: none"> リダイレクトの代わりにQRコードを使用して、両デバイス間を連携 QRコードからURI取得後、検証者とウォレット間の通信はインターネット経由のため、検証側はHTTPSリクエストを受信する機能が必要
	3	OpenID for Verifiable Presentations over BLE	<ul style="list-style-type: none"> Bluetooth Low Energy (BLE)を活用することで、一方または双方のエンティティがインターネット機能がない場合でも、VPを要求し受信することが可能
SIOPv2	4	Same-Device Self-Issued OP	<ul style="list-style-type: none"> クライアントアプリケーション(RP)とOpenID Provider(OP)が同一端末で動作するフローを定義しており、RPとOP間の連携にリダイレクトを使用 v2からDIDが利用可能
	5	Cross-Device Self-Issued OP	<ul style="list-style-type: none"> OPが別デバイス (通常の認可サーバや別端末) で動作するフローを定義

		ベース機能		拡張機能	
複数デバイス間でのデータ交換	QRコード				業者が既存仕様をもとに設計・実装
	NFC	2	5		
	Bluetooth			3	
同一デバイス内でのデータ交換		1	4		業者が既存仕様をもとに設計・実装

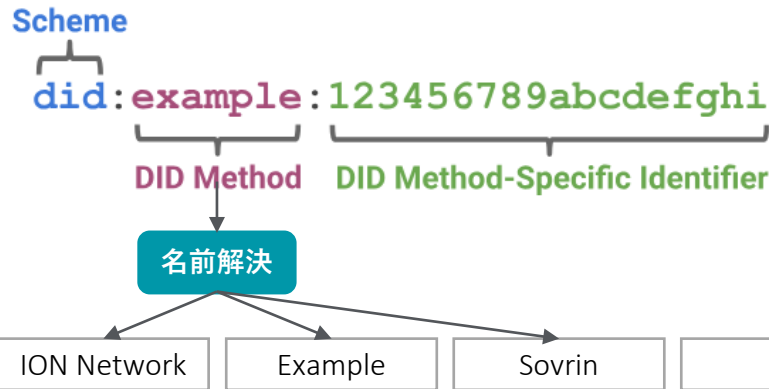
Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document / DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure / ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (7) DID Documentを活用した証明書の検証方法

- DID DocumentとVCの紐付けはDIDからシステム内のDID Documentを特定し、VC内のProofセクションに指定された公開鍵で検証を行う

DID

スキーマ この情報がDIDであることを示す	DIDメソッド DID Documentを格納したシステムを示すもの(did:ion, did:sov, did:web)	DIDメソッド特有のID DIDメソッドで示されたシステム内における一意なID
---------------------------------	---	---



DID Document

```

{
  "id": "did:example:123456789abcdefghi",
  "authentication": [
    {
      "id": "did:example:123456789abcdefghi#keys-1",
      "type": "Ed25519VerificationKey2018",
      "controller": "did:example:123456789abcdefghi",
      "publickeyBase58": "H3C2AVvLMv6gmmBam3uVAjZpfkCjCwDwnZn.."
    }
  ]
}
    
```

Verifiable Credential(VC)

※VPはVCから必要情報のみ取り出す

```

{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ],
  "id": "http://example.edu/credentials/3732",
  "type": [
    "VerifiableCredential",
    "UniversityDegreeCredential"
  ],
  "issuer": "https://example.edu/issuers/565049",
  "issuanceDate": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "did:example:123456789abcdefghi",
    "degree": {
      "type": "BachelorDegree",
      "name": "Bachelor of Science and Arts"
    }
  },
  "proof": {
    "type": "Ed25519Signature2018",
    "created": "2022-02-25T14:58:42Z",
    "verificationMethod": "did:example:123456789abcdefghi#keys-1",
    "proofPurpose": "assertionMethod",
    "proofValue": "z3FXQjecWufY46yg5abdVZsXqLhxhueuSoZgNSARiKBk..."
  }
}
    
```

証明書の定義情報

証明書のIDと種別

発行体と発行日

VCの主要情報

検証用の署名

VCと紐付け

公開鍵を特定

検証

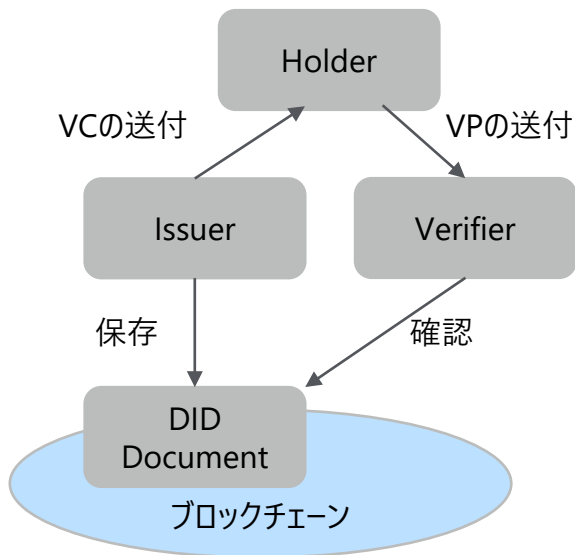
Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document /DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure /ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (8) DID Documentの格納場所

- DID Documentの格納し情報共有する方式は、大きく分類して①ブロックチェーン、②WEBサーバ、③P2P通信の3通りある

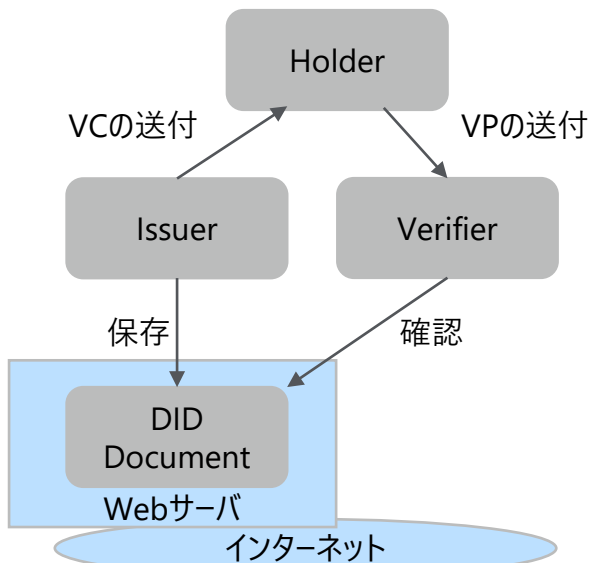
①ブロックチェーン

- Issuerが特定のブロックチェーン上にDID Documentを保存し、Verifierはそこを参照してVCの真正性を確認
- ブロックチェーンが耐改ざん性、可用性を担保してくれるため、Data Registryとしては一般的な方式（安全性・機密性の観点からPermissionedなブロックチェーンを用いられることが多いが、スケーラビリティに課題がある）



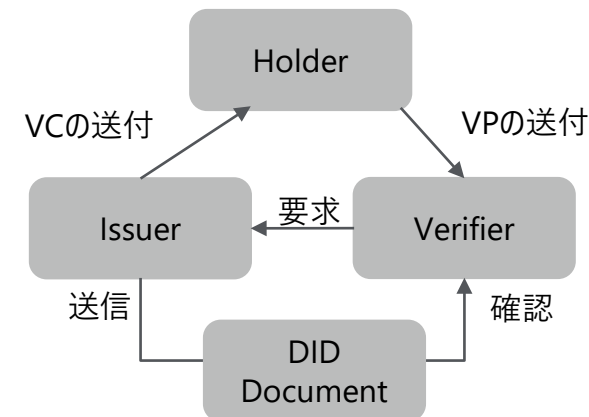
②WEBサーバ

- Issuerが特定のWebServerに公開鍵を含むDID Documentを保存し、Verifierはそこを参照してVCの真正性を確認
- Webサーバに格納されたDID Documentの耐改ざん性や可能性の担保が課題



③P2P通信

- Verifierの要求に応じてIssuerがDID Documentを直接送信
- 直接送付するため秘匿性は高い（DID Documentに秘匿性の高い情報を載せる場合等には有効）
- 送信方法にも依存するが、方法は複数存在し確立していない（メール、FTP...）

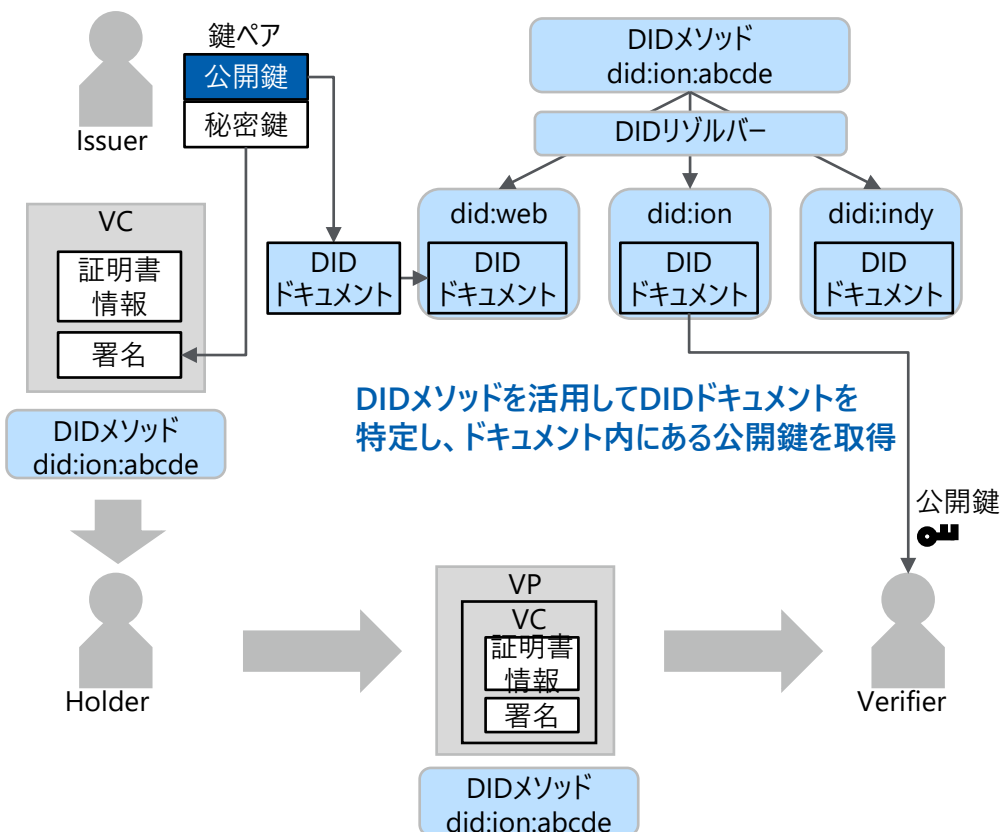


(参考)DID Documentを活用しない方式

- DIDは複数のプラットフォーム間で、一意にIDを識別する仕組みのため、複数プラットフォーム連携ニーズが無い場合、VCはDIDおよびDID Documentを適用しなくても成立する

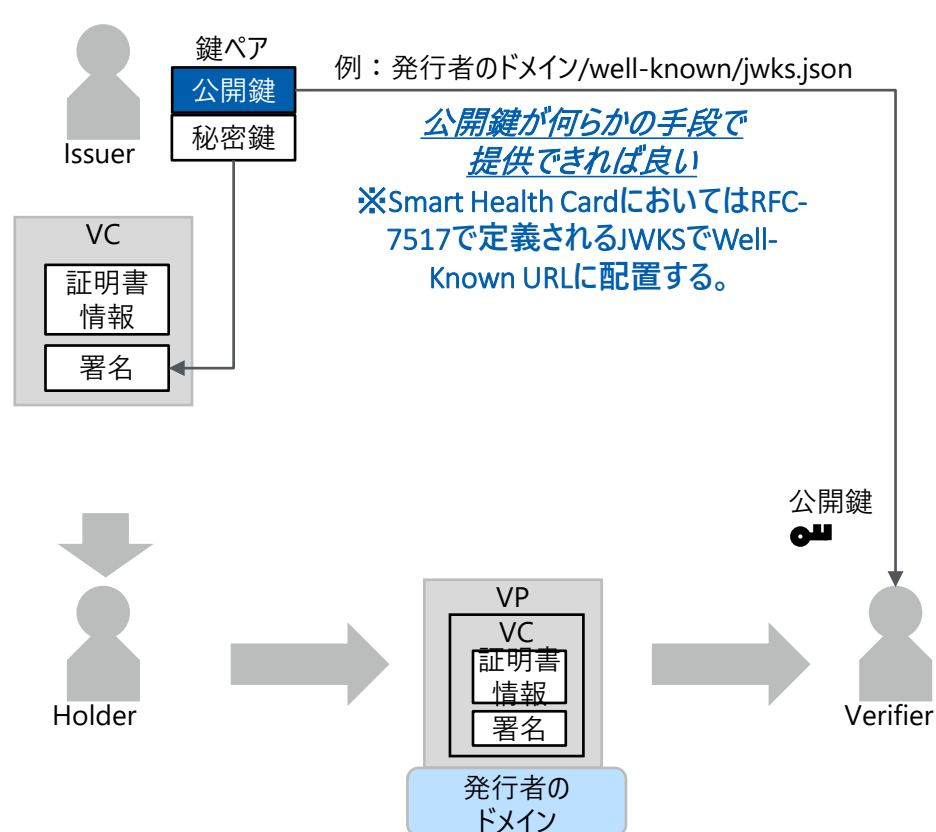
①DIDを活用

DIDは他者と信頼できる情報（公開鍵等）を適切に交換することを目的とするため、DIDにはプラットフォームを識別するDIDメソッド情報が含まれている。このDIDメソッドをDIDリゾルバーによって名前解決を行い、プラットフォームを特定し、DIDドキュメントを取得する（WebにおけるDNSのような役割）



②DIDを活用しないケース

検証可能な証明書としてVCモデルを採用する場合においても、証明書を発行する側が中心的な役割を果たす場合には、DIDは必ずしも不要となる。（例えば日本における新型コロナワクチン接種証明で採用されたSmart Health Cardにおいては、VCモデルに準拠しているもののDIDは利用されていない）



Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document /DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure /ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (9) ブロックチェーン比較

- 分散型アイデンティティに使用されるブロックチェーンは不特定ユーザとの検証が前提となるため、全公開が基本となる。ブロック作成、検証行為のみが許可制となる場合がある

ブロックチェーンマッピング

		Validator の許可レベル			
		Permissionless		Permissioned	
データアクセス 制御	Public	Bitcoin	Ethereum	Hyperledger Indy	
	Private			Hyperledger Fabric	R3 Corda

アイデンティティ管理等で活用する主要ブロックチェーン比較

	Ethereum	Hyperledger Indy	Hyperledger Fabric
VC活用事例	Civic Pass	BC Digital Trust/NB Orbit	Interac Verification Service
総評 (弊社)	参加制限がなくデータは全公開のため、秘匿情報を別に扱うオフチェーンの仕組みが必要。アカウント失効等の実装が課題になり標準化が停滞している状況。近年ではSBTに統合されて仕様検討が進んでいる。自己管理の考え方が強く、NFTやDAO等他のWeb3サービスとの連携に活用可能性がある。	分散型アイデンティティ管理に特化しており、クレデンシャル定義や失効レジストリ等の機能を有する。ネットワークへの参加は許可制であるが、情報は全公開であり、クライアント用のツールキットのAriesで、エンティティ間のメッセージ交換や格納を制御する。信頼性と運用性のバランスが取り易いが、コンソーシアム運営が課題。	エンタープライズ利用を前提とした汎用プラットフォーム。チャンネルやプライベートトランザクション等機能が搭載されている。秘匿情報をブロックチェーン上で一元的に扱える利点があるが、アクセス制御の組合せが多いとデータ容量・運用面の設計の複雑さや、Indyと同様コンソーシアム運営などが課題。
参加者制限	×	○	○
ノード保有者データアクセス制御	×	×	○
スマートコントラクト	○	×	○
スループット[TPS]	10~15	100	400
コンセンサスアルゴリズム	Proof of Stake	Plenum	Raft & Endoring-Ordering-Validation
出所	https://ethereum.org/ https://github.com/ethereum	https://www.hyperledger.org/projects/hyperledger-indy	https://www.hyperledger.org/projects/fabric

(参考)ブロックチェーン領域で活用される秘匿化技術

- 機密性やプライバシーレベルが高い情報を秘匿化する技術（PET：privacy-enhancing technologies/techniques）は多数存在し、大きくは3つに分類される。また、パブリックチェーン / プライベートチェーンで活用される技術は異なる。

	共有先制御型PET	非可読化型PET	関係性隠匿型PET
説明	各参加者がネットワーク上の全取引の一部にしかアクセスできないようにする手法	暗号化技術を用いることで、第三者が取引情報を解釈できないようにする手法	台帳に記録された送金者・受領者情報から、第三者が取引当事者を特定することを困難にする手法
パブリック/ プライベート チェーン双方 で活用され る技術	ペイメントチャネル オフチェーンで取引することで、秘匿性を強化する仕組み。参加者は個々の取引をネットワーク全体にブロードキャストすることがない	ゼロ知識証明（ZKP） データを公開することなく、そのデータの真実性のみを証明する。VCでは、BBS + 署名と組み合わせて取引情報の秘匿化が行われる 準同型暗号 データを暗号化したまま計算可能な暗号方式	秘密分散 データを断片化して一定以上の断片が揃わないと復元できない仕組み リング署名 複数人が記載された公開鍵リストがあり、その中に署名者が存在することを保証する(誰であるかは特定できない) グループ署名（BBS+署名/CL署名） あるグループに属する者の署名からはそのグループに属することしかわからず、グループ管理者だけが特定できる。VCでは、ゼロ知識証明と組み合わせて取引情報の秘匿化が行われる
プライベート チェーンで 主に活用さ れる技術	Flow Framework（Corda） 取引当事者間のみデータを共有するトランザクションを発行可能 Channel（Hyperledger Fabric） グループ単位にブロックチェーンを分割する方式。トランザクション送信時にチャンネルを指定することで、任意のグループのみデータを共有可能	Private Transaction（Quorum） トランザクションに秘匿情報を含まないハッシュ等のみを格納（アンカリング）し、データの実体は指定ノードのみ共有する方式 Private Data Collection（Hyperledger Fabric） トランザクションに秘匿情報を含まないハッシュ等のみを格納し、データの実体は指定ノードのみ共有する方式。チャンネルより細かい単位で共有範囲を指定	—

<https://www.boj.or.jp/paym/fintech/rel200212a.htm>をもとにTOPPAN作成

Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document /DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure /ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細- (10) Universal Resolver

- DID Methodを識別し、名前解決を行う仕組みとして実質的な標準となっているのがUniversal Resolver。DID MethodをサポートするためにプラグインとなるDriverを追加する

対応ドライバリスト

- W3Cが制定するDID Core 1.0およびDID Resolution仕様に基いてさまざまなDIDメソッドに対応した名前解決やDIDの登録を行う
- Dockerイメージで提供されREST APIを利用して動作する。ドライバー形式を採用しており、特定のDIDメソッドに対応したドライバーを組み込むことでサポートするメソッドを拡張できる
- 現時点で60個以上のドライバに対応しており、標準化はまだされていないものの、事実上のデファクトスタンダードとなっている
- 現在はDIF Identifiers & Discovery Working Groupの作業項目となっている

Drivers

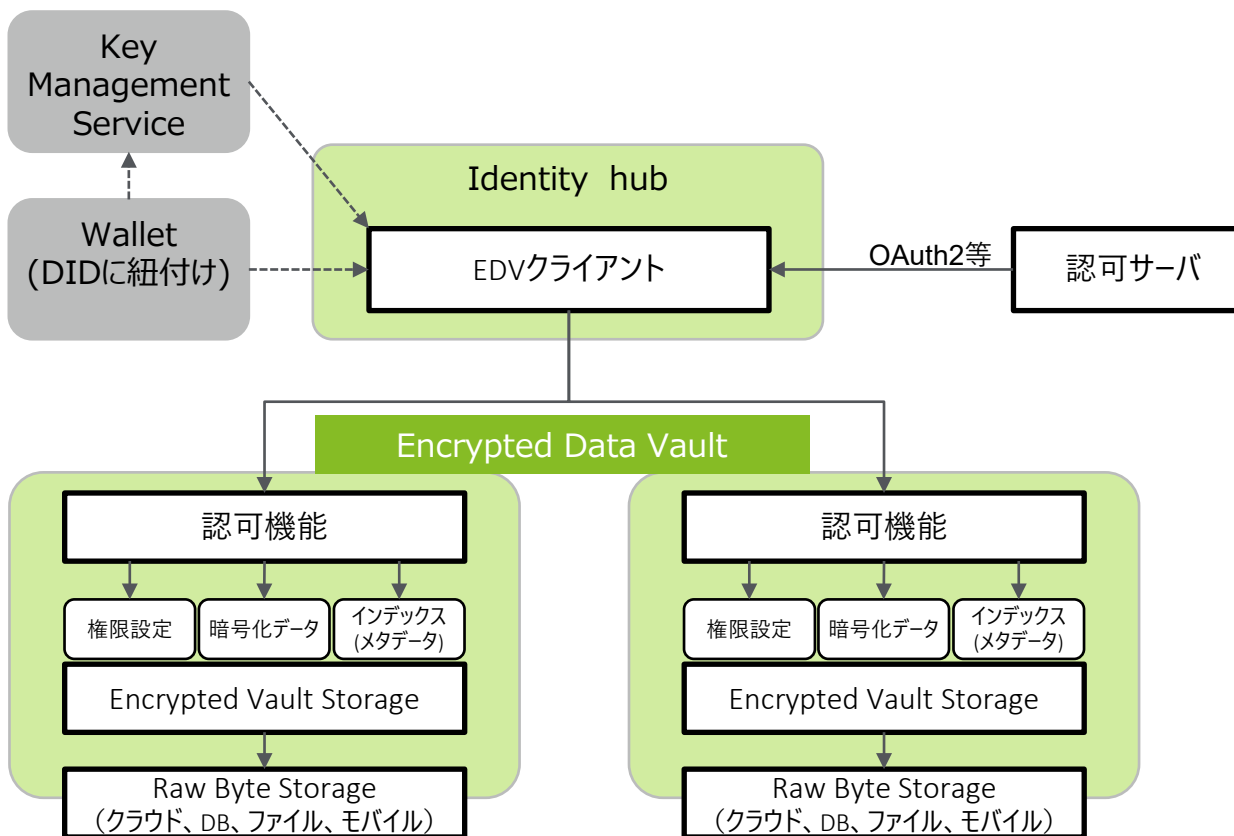
Are you developing a DID method and Universal Resolver driver? Click [Driver Development](#) for instructions.

Driver Name	Driver Version	DID Method Spec Version	Docker Image or URL	Description
did-btcr	0.1-SNAPSHOT	0.1	universalresolver/driver-did-btcr	Bitcoin Reference
did-sov	0.1-SNAPSHOT	0.1	universalresolver/driver-did-sov	Sovrin public ledger
did-stack	0.1	1.0	universalresolver/driver-did-stack	
did-dom	0.1-SNAPSHOT	(missing)	universalresolver/driver-did-dom	
did-ethr	4.3.0	9.1.0	uport/uni-resolver-driver-did-uport	Ethereum addresses or secp256k1 publicKeys
did-ens	4.3.0	0.1.1	uport/uni-resolver-driver-did-uport	ENS names
did-web	4.3.0	3.0.0	uport/uni-resolver-driver-did-uport	Domain name
did-peer	4.3.0	1.0-draft	uport/uni-resolver-driver-did-uport	Peer DID
did-eosio	0.1.3	0.1	gimlyblockchain/eosio-universal-resolver-driver	EOSIO blockchain platform
did-v1	0.1	1.0	veresone/uni-resolver-did-v1-driver	Veres One Blockchain
did-jolo	0.1	0.1	jolocomgmbh/jolocom-did-driver	Jolocom identity management
did-hacera	0.1	(missing)	hacera/hacera-did-driver	HACERA autonomous data exchange network

Credential Layer		Agent Layer		Public Trust Layer				Vertical / Cross-cutting				
Credential Format	Credential Proofing	Envelope	Transport	DID Document /DID Scaling	DID method	Anchor Types	DID Resolution	DID-Anchored Svcs	Disclosure /ZKP	Storage	Data Formats	Crypto Primitives

5.2. 規格・実装方式の詳細 - (11) 証明書を安全に格納するサービス規格

- 暗号化データ保管庫（EDV：Encrypted Data Vault）は個人や企業が所有(契約)するクラウドストレージやモバイルストレージ等のデータを安全に格納、インデックス、共有するためのメカニズムを定義する仕組みでありW3Cで検討されている
- ユーザや企業（エンティティ）はストレージプロバイダーに内容を知られることなくデータを格納可能で、かつプロバイダーの管理者がアクセスできなくなる仕組みを提供する。クライアントはDIDに紐付けられたキーを使って独自の暗号化・復号を行うため、クライアントが参照先を完全に管理できる



5.3. mDLとVCの比較

- mDLとVCは両規格とも所有者が自身の資格情報を制御できるが、mDLはデータフォーマットや通信プロトコルまで厳密に定義されており、資格情報管理も発行機関が一元的に管理できる選択肢を残している点がかつとも大きな相違点としてあげられる

	mDL	VC
サマリ	各エンティティの役割、処理フロー、データモデル（基本項目）からハードウェア、通信規格まで厳密に決められており、最低限のサービスの互換性を担保	制御に必要な必要最低限のデータ項目のみ定義されており、ハードウェアやソフトウェアの詳細仕様が、明確に定義されていないため個別に解釈して独自実装するか、関連団体(OIDFやDIF等)に移譲
アーキテクチャ	<ul style="list-style-type: none"> 発行機関のサーバから資格情報を取得することも可能であるが、その場合、発行者はいつ・誰によって使用されたか知ることができる（事前同意は行う） オプションとして証明性の検証用にVICALによるルート証明書を配布できる(失効確認) 	<ul style="list-style-type: none"> VDRに検証のためのメタ情報を格納し、所有者のデバイスに資格情報を格納することが前提
データモデル	<ul style="list-style-type: none"> データモデルとして規定された用途(運転免許)がある フォーマットはCBOR(※)またはJSON 基本名前空間と拡張用名前空間を持つ 	<ul style="list-style-type: none"> 汎用的なモデルとして定義されている 基本的な資格項目は存在しない JSONLDまたはJSONが標準であるが、他の形式でも利用可能
通信プロトコル	<ul style="list-style-type: none"> mDLとリーダー、発行機関とリーダーとのI/Fを明確に定義(デバイス接続：NFCまたはQRを使用、通信I/F：WiFi、BLE、NFCを使用) サーバアクセスはAPI/OIDCのエンドポイント使用 	<ul style="list-style-type: none"> 資格情報の交換に必要なインターフェイスは定義されていない
セキュリティ	<ul style="list-style-type: none"> セッション暗号化（AES） MSO(Mobile Security Object)に格納された情報に付与したデジタル署名を検証 	<ul style="list-style-type: none"> 特定形式のデジタル署名は未定義 全てのエンティティがVDRを信頼する必要がある。ただし、VDRに必要な条件は規定されていない
共通事項	<ul style="list-style-type: none"> 発行した資格情報を所有者の管理下にあるデバイスまたはレジストリに格納する 資格情報は完全に所有者に制御されて、検証者に提供する判断は所有者にある 発行者は所有者の同意なしに、認証情報を検証者に直接公開できない 所有者がいつ・どこでmDLを使用するか発行者・検証者は認識できない 	

(参考) Unlinkabilityとは

<p>背景</p>	<ul style="list-style-type: none"> ISO/IEC 29100で定めている個人識別情報(PII)の処理に関わるアクターに適用されるプライバシー原則の中の1つに「収集の制限」があるが、現在は、インターネットサイトがサービスへのアクセス時に必要以上の情報を収集することが一般的となっている。 例えば、WebサイトがPIIの主体が特定の年齢以上であることのみを確認したいときに、ユーザーの永続的識別子などの不必要な情報まで取得していることが挙げられる。これによって同一のPII主体による異なるサイトへの訪問や、同一サイトへの二回以上の訪問をリンクすることが可能となっており「収集の制限」の原則が達成されていない状況である。 収集の制限の原則に従うためには、上記のケースのサイトはPII主体による二回以上の訪問をリンクさせないタイプのエンティティ識別子を使用すべきで、二つのトランザクションが行われた場合、それらのトランザクションが同一ユーザーによるものか、異なる二人のユーザーによるものかを区別できないこと(Unlinkabilityの確保)が求められる。 属性ベースのリンク不可能なエンティティ認証(ABUEA：attribute-based unlinkable entity authentication)は、PII主体が、Unlinkabilityを確保したうえで、身元属性情報の真正性を確立する手段を提供することができこの関連技術標準をまとめたものがISO/IEC 27551(Information security, cybersecurity and privacy protection — Requirements for attribute-based unlinkable entity authentication)で規定されている。
<p>VC/mDLを活用する際に留意すべき事例</p>	<p>【VC】</p> <ul style="list-style-type: none"> 証明書の署名値を同一のものを使用した証明書をHolderがVerifierに証明書の提示を行うと、Verifierの結託によってどこに情報提示したか類推される(P.21参照) <p>【mDL】</p> <ul style="list-style-type: none"> Server Retrievalで証明書の失効問い合わせをIssuerにすると、Issuerは、特定のHolderとVerifierがやり取りしたかが類推できる

※Unlinkabilityについて詳細を確認したい場合、[ISOサイト\(ISO/IEC27551\)](#)から購入すること

6. 実装パターンの抽出

6.1. サービス – 6.1.1. サービス実装パターン (1) JWT-VC・SD-JWT・LDP-VC

<p>総評</p>	<p>Credential Layerの証明書フォーマットおよび署名アルゴリズムの暗号方式は下位レイヤに影響しないため、組合せのバリエーションが多い。検証者に提供する証明書の項目を選択的に開示するため既存フォーマットで対応するか、より高機能かつ柔軟性の高い組合せにするかで異なる</p>	<p>112</p>
<p>Credential Layer</p>	<p>証明書モデル 検証者に提供する証明書モデル Credentialフォーマット 利用される暗号方式 (選択的的属性開示含)</p> <p>W3Cに準拠しつつ独自に実装するパターンとOIDFで規定しているOID4VPに対応する2パターンに大別される。証明書のフォーマットは選択的開示対応を既存フォーマットで行うか、厳密に構造化された新しいフォーマットで行うかで別れる。証明書と署名方式の組み合わせは事例ベースで確認*</p> <p>VC Data Model(v1.0 / v1.1 / v2.0) Verifiable Presentations OID4VP JWT-VC SD JWT-VC LDP-VC ECDSA EdDSA BBS+</p> <p>2 7*</p>	<p>14</p>
<p>Agent Layer</p>	<p>エンティティ間の通信プロトコル 通信I/F(デバイス連携) 鍵管理関連</p> <p>JWT-VC、SD-JWT、LDP-VCは別レイヤの実装に影響しないものの、DIFで規定するDIDCommか、OIDFの自己発行を規定したSIOPに大別される。デバイス連携・鍵管理については明確に定義されていないため、自前で実装が必要。</p> <p>DIDcomm (v1 / v2) SIOP v2 連携なし QR NFC BLE ウォレット等を実装するハードウェアの種類によって対応すべきパターンが異なる ※HSM/TEE/SmartCard等がある</p> <p>2 4 N/A</p>	<p>8</p>
<p>Public Trust Layer</p>	<p>システムアクセスのための識別子 DIDメソッド DID Document格納方式 (ストレージ) 名前解決</p> <p>JWT-VC、SD-JWT、LDP-VCは別レイヤの実装に影響しないため、本レイヤにおける組合せはプラットフォームごとに異なり、複数存在するため、パターンを絞り込むことが出来ない。</p> <p>※didを活用しない DIDメソッド did : indy did : web did : sov ※didメソッドで格納方式は一意に決定されることが多い B/C活用 B/C非活用 Hyperledger Indy ION Network Ethereum P2P Web Server IPFS Universal Resolver</p> <p>100以上 N/A</p>	<p>N/A</p>

* https://docs.google.com/spreadsheets/d/1X93ptJcmfX1NZEo5E7ElnqJ-knDS4Dj6JOYSJ_2PsUw/edit#gid=1590639334

6. 実装パターンの抽出

6.1. サービス – 6.1.1. サービス実装パターン (2) AnonCreds

<p>総評</p>	<p>AnonCredsはHyperledgerプロジェクトで検討されている。レガシー仕様ではHyperledger Indyを前提に検討されていたが、近年では様々なプラットフォームに対応することが検討されている。ただし、前提となるバリエーションはそれほど多くない。</p>	<p>64</p>																														
<p>Credential Layer</p>																																
<p>証明書モデル</p> <p>検証者に提供する証明書モデル</p> <p>Credentialフォーマット</p> <p>利用される暗号方式 (選択的属性開示含)</p>	<p>当初はCL署名のみ対応していたが、AnonCreds v2において、PS署名に対応し、BBS+署名もサポートしている。また、ポスト量子オプションも検証している。 https://github.com/hyperledger/anoncreds-v2-rs</p>	<table border="1"> <tr> <td colspan="2">VC Data Model(v1.0/ v1.1 / v2.0)</td> <td></td> </tr> <tr> <td colspan="2">Verifiable Presentations</td> <td>2</td> </tr> <tr> <td colspan="2">AnonCreds</td> <td>2</td> </tr> <tr> <td>CL</td> <td>BBS+</td> <td></td> </tr> </table>	VC Data Model(v1.0/ v1.1 / v2.0)			Verifiable Presentations		2	AnonCreds		2	CL	BBS+																			
VC Data Model(v1.0/ v1.1 / v2.0)																																
Verifiable Presentations		2																														
AnonCreds		2																														
CL	BBS+																															
<p>Agent Layer</p>																																
<p>エンティティ間の通信プロトコル</p> <p>通信I/F(デバイス連携)</p> <p>鍵管理関連</p>	<p>証明書を安全に伝達するためのプロトコルとしてDIDCommが定義、デバイス連携は対象外、鍵管理は以下で検討 https://github.com/hyperledger/aries-askar</p>	<table border="1"> <tr> <td colspan="4">DIDcomm (v1 / v2)</td> <td>1</td> </tr> <tr> <td>連携なし</td> <td>QR</td> <td>NFC</td> <td>BLE</td> <td>4</td> </tr> <tr> <td colspan="4">Aries-askar</td> <td>1</td> </tr> </table>	DIDcomm (v1 / v2)				1	連携なし	QR	NFC	BLE	4	Aries-askar				1															
DIDcomm (v1 / v2)				1																												
連携なし	QR	NFC	BLE	4																												
Aries-askar				1																												
<p>Public Trust Layer</p>																																
<p>システムアクセスのための識別子</p> <p>DIDメソッド</p> <p>DID Document格納方式 (ストレージ)</p> <p>名前解決</p>	<p>DIDメソッドはAnonCreds Methods Registryにて定義されている、レガシー仕様としてHyperledger Indyを利用する場合にはdid:indyが推奨されているが、新規としては4種類定義されている。HTTPはdid:webとほぼ同一 https://hyperledger.github.io/anoncreds-methods-registry/</p>	<table border="1"> <tr> <td colspan="5">AnonCreds Methods Registry</td> </tr> <tr> <td colspan="2">※DIDを活用しない</td> <td colspan="3">DIDメソッド</td> </tr> <tr> <td>HTTP</td> <td>cheqd</td> <td>Cardano</td> <td>それ以外</td> <td>did:indy</td> </tr> <tr> <td colspan="3">B/C活用</td> <td colspan="2">B/C非活用</td> </tr> <tr> <td>Hyperledger Indy</td> <td>cheqd</td> <td>Cardano</td> <td colspan="2">Web Server</td> </tr> <tr> <td colspan="5">Universal Resolver</td> </tr> </table> <p>基本 4</p>	AnonCreds Methods Registry					※DIDを活用しない		DIDメソッド			HTTP	cheqd	Cardano	それ以外	did:indy	B/C活用			B/C非活用		Hyperledger Indy	cheqd	Cardano	Web Server		Universal Resolver				
AnonCreds Methods Registry																																
※DIDを活用しない		DIDメソッド																														
HTTP	cheqd	Cardano	それ以外	did:indy																												
B/C活用			B/C非活用																													
Hyperledger Indy	cheqd	Cardano	Web Server																													
Universal Resolver																																

6. 実装パターンの抽出

6.1. サービス – 6.1.1. サービス実装パターン (3) mDL

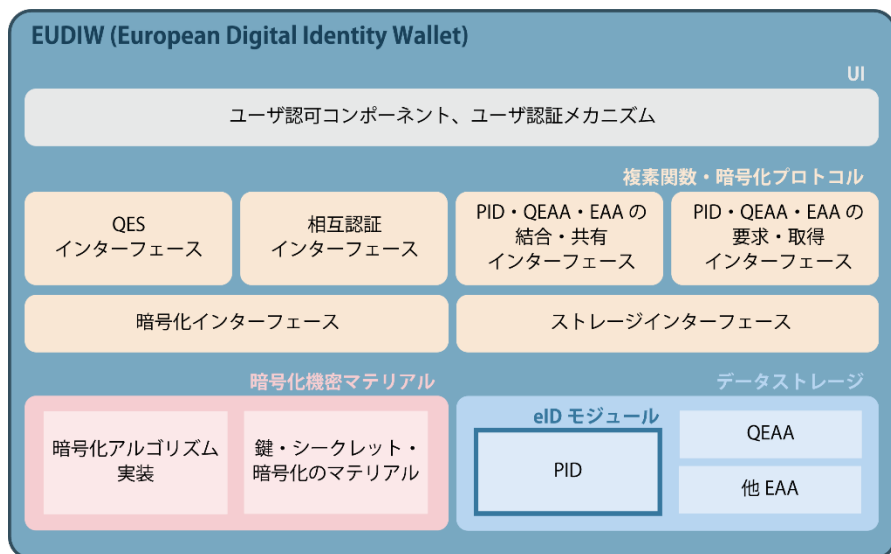
<p>総評</p>	<p>mDLはISOにて厳密に処理方式からフォーマット、デバイス連携、エンティティ間のデータの受け渡し方法等を明確に定義しているため、ISO内ではそれほどバリエーションがないが、仕様そのものは特定のフォーマットや実装に依存しないため、複数の仕様に適用させることが可能。</p>	<p>144</p>	
<p>Credential Layer</p>			
<p>証明書モデル</p>	<p>ISO/IEC 18013-5ではデジタル運転免許証の提示と検証に関する一連の Protokol とデータ交換フォーマットを定義されており、証明書モデルは実装や具体的なアプリケーションに依存する。署名アルゴリズムはECDSAおよびEdDSAが規定されている。</p>	<p>mDL</p>	<p>12</p>
<p>検証者に提供する証明書モデル</p>		<p>ISO/IEC 18013-5 Verifiable Presentations OID4VP 3</p>	<p>=</p>
<p>Credential フォーマット</p>		<p>mdoc ×</p>	<p>×</p>
<p>利用される暗号方式 (選択的屬性開示含)</p>		<p>ES256 ES384 ES512 EdDSA 4</p>	<p>×</p>
<p>Agent Layer</p>			
<p>エンティティ間の通信プロトコル</p>	<p>ISOで定めた通信プロトコルがあるものの、他の方式での実装も可能。デバイス連携も想定されているため、組合せは多い。</p>	<p>ISO/IEC 18013-5 SIOP</p>	<p>4</p>
<p>通信I/F(デバイス連携)</p>	<p>https://www.iso.org/standard/69084.html</p>	<p>Server retrieval (Same device) Device retrieval (NFC) Device retrieval (QR) Device retrieval (BLE) 4</p>	<p>×</p>
<p>鍵管理関連</p>		<p>なし N/A</p>	<p>×</p>
<p>Public Trust Layer</p>			
<p>システムアクセスのための識別子</p>	<p>mDLにおいては、Issuer Authority がHolderに対し証明書を直接送信するか、またはサーバに格納するか2つの選択肢が存在する。</p>	<p>※DIDを 活用しない</p>	<p>3</p>
<p>DIDメソッド</p>			
<p>DID Document格納方式 (ストレージ)</p>	<p>ストレージはローカルかサーバか選択可能だが、サーバの派生でB/Cに格納することも特定条件をクリアすることで可能(通信暗号化、JWS署名対応、アクセス制御)</p>	<p>B/C活用 B/C非活用 特定条件をクリアすれば可能 (プライベートチェーン必須) サーバ ローカルストレージ 3</p>	
<p>名前解決</p>		<p>なし</p>	

6. 実装パターンの抽出

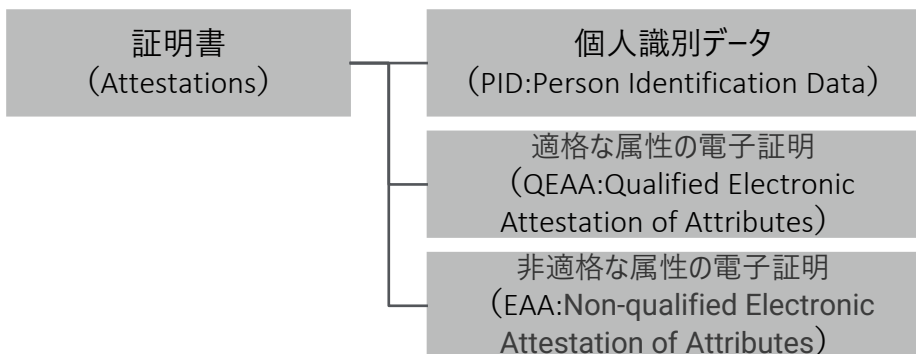
6.1. サービス – 6.1.2. サービス実装詳細 (1) EU DIW (EU ARFで策定されているもの)

- ARFはその目的としてEUDIウォレット中心の仕様、エコシステムの各アクターの役割、ウォレットの機能要件および非機能要件等を定義している。

EUDIウォレットの機能構成



EUDIウォレットに格納される証明書構成



実装技術スタック詳細

Credential Layer	証明書モデル	Verifiable Credential	mDL
	Credential フォーマット	SD-JWT (JSON)	LDP-VC (JSON) mdoc (CBOR)
	選択的開示の実現方式	(SD-JWT ゼロ知識証明はオプション)	
	署名方式	ECDSA/EdDSA	
	検証者に提供する証明書モデル	OID4VP	
Agent Layer	エンティティ間の通信プロトコル	SIOPv2/OID4VP	
	通信I/F (デバイス連携)	NFC, Bluetooth, QRコード	
Public Trust Layer	システムアクセスのための識別子	N/A	
	DID Document格納方式 (ストレージ)	Holder: スマホストレージ (SE等) Issuer: N/A	
	名前解決	N/A	

6. 実装パターンの抽出

6.1. サービス – 6.1.2. サービス実装詳細 (2) Lissi

- Lissiは、EUDIWのARFの技術標準(SD-JWT、OID4VC系等)と、IDUnionネットワークで構想している技術仕様(Hyperledger AnonCreds / Indy、DIDComm等)の双方に対応している

サービスの特徴	
開発元	Lissi GmbH
サービス概要	<ul style="list-style-type: none"> • eIDAS2.0に準拠しているウォレットを提供 • IDunion(ドイツ)のIDunionネットワークをサポート
利用事例	<ul style="list-style-type: none"> • 顧客の会員証 • 従業員の資格証 • 従業員・顧客のIDアクセスマネジメント ※30ほどのユースケース事例があると記載されている
関与/参照している主要規格団体・フレームワーク	<ul style="list-style-type: none"> • eIDAS2.0 • IDUnion network • W3C (W3C VC, SD-JWT) • DIF (DIDComm) • OpenID Foundation (OID4VP, SIOPv2) • Hyperledger Foundation (AnonCreds / Indy / Aries) • ISO/IEC 18013シリーズ 等
今後の動向	N/A

実装技術スタック詳細			
Credential Layer	証明書モデル	ISO/IEC18013-5:2021, W3C VC	
	Credential フォーマット	CBOR+MSO、SD-JWT VC、JSON-LD+LD-Proofs	
	選択的開示の実現方式	SD-JWT	
	署名方式	ECDSA	
Agent Layer	検証者に提供する証明書モデル	ISO/IEC18013-5:2021, OID4VP	
	エンティティ間の通信プロトコル	ISO/IEC18013-5:2021, SIOP v2, DIDComm	
Public Trust Layer	通信I/F (デバイス連携)	QRコード, e-mail	
	システムアクセスのための識別子	N/A	didi:indy
	DID Document格納方式 (ストレージ)	スマホストレージ (SE等)	Hyperledger Indy
	名前解決	N/A	

出所：<https://www.lissi.id/eidas-2-0>
https://idunion.org/wp-content/uploads/2023/07/2023_06_05_TDI_Framework_for_Walletsecurity.pdf

6. 実装パターンの抽出

6.1. サービス – 6.1.2. サービス実装詳細 (3) Microsoft Entra Verified ID

- Microsoft Entra Verified IDは、Webサーバ、ブロックチェーン(ion network)のストレージに対応した証明書発行・検証サービスを提供している

サービスの特徴	
開発元	Microsoft
サービス概要	<ul style="list-style-type: none"> 従業員・顧客等の資格情報の発行・検証が可能なサービスを提供 Microsoft Entra ID のサービスに含まれており、Identity Access Management 関連のサービスの拡張で、追加費用なしで利用可能
利用事例	<ul style="list-style-type: none"> NHS(National Health Service) 医療従事者の資格管理 ロイヤルメルボルン工科大学 成績証明書としての活用 イギリス教育省 成績証明書としての活用(実証)
関与/参照している主要規格団体・フレームワーク	<ul style="list-style-type: none"> W3C (VCデータモデル、JWT-VC等) IETF (JWT-VC) DIF (Presentation Exchange v1.0 / Well Known DID Configuration 等に準拠) OIDF (SIOPv2 / OID4VC)
今後の動向	N/A

実装技術スタック詳細			
Credential Layer	証明書モデル	Verifiable Credentials Data Model v1.1	
	Credential フォーマット	JWT – VC	
	選択的開示の実現方式	(選択的開示は実装検討中)	
	署名方式	EdDSA	
Agent Layer	検証者に提供する証明書モデル	OID4VP	
	エンティティ間の通信プロトコル	SIOPv2	
Public Trust Layer	通信I/F (デバイス連携)	QRコード	
	システムアクセスのための識別子	did:web	did:ion
	DID Document格納方式 (ストレージ)	Webサーバ	ion network (bitcoinレイヤ2)
	名前解決	Universal Resolver	

出所：<https://www.microsoft.com/ja-jp/security/business/identity-access/microsoft-entra-verified-id>

6. 実装パターンの抽出

6.1. サービス – 6.1.2. サービス実装詳細 (4) AnonCreds

- AnonCredsはHyperledger Foundationがサポート、現在はOID4VC系への対応がないが今後対応が期待される

技術スタック

開発元	Hyperledger Foundation
サービス概要	<ul style="list-style-type: none"> VCの検証に重要なプライバシー保護(ゼロ知識証明)機能を基本機能として追加されていることが特徴
利用事例	<ul style="list-style-type: none"> 他Hyperledger Indy等を活用しているサービス
関与/参照している主要規格団体・フレームワーク	<ul style="list-style-type: none"> W3C Hyperledger プロジェクト(Indy / Aries)
今後の動向	<ul style="list-style-type: none"> AnonCreds v2に向けた取り組み (BBS+署名への対応) OpenID Foundation系の規格への対応

実装技術スタック詳細

Credential Layer	証明書モデル	Verifiable Credentials Data Model v1.1
	Credential フォーマット	AnonCreds
	選択的開示の実現方式	AnonCred ZKPs
	署名方式	CL(Ver.1.0) BBS+ (Ver.2.0)
	検証者に提供する証明書モデル	Verifiable Presentations
Agent Layer	エンティティ間の通信プロトコル	DIDComm V2
	通信I/F (デバイス連携)	N/A
Public Trust Layer	システムアクセスのための識別子	did:indy (+ link secrets)
	DID Document格納方式 (ストレージ)	Hyperledger Indy
	名前解決	Universal Resolver

出所：<https://www.hyperledger.org/projects/anoncreds>

6. 実装パターンの抽出

6.1. サービス – 6.1.2. サービス実装詳細 (5) BC Digital Trust

- BC Digital TrustはHyperledger Indyを活用して、カナダブリティッシュコロンビア州のサービス(弁護士資格確認・大学学生資格確認)を提供している

サービスの特徴	
開発元	Province of British Columbia (Canada)
サービス概要	<ul style="list-style-type: none"> VCを発行・検証するためのソフトウェア、オープンソースのBCウォレットを提供、検証者が発行者に問い合わせることなく証明書の検証を行うので高いプライバシーレベルを確保 BC州で登録された組織に関する信頼できる情報を提供するデジタル資格情報を使用する検索可能な公開ディレクトリを提供
利用事例	<ul style="list-style-type: none"> 州弁護士の資格確認 州大学の学生資格確認
関与/参照している主要規格団体・フレームワーク	<ul style="list-style-type: none"> W3C (W3C VC, JSON-LD) Hyperledger Foundation (Indy /Aries) Trust Over IP Foundation 等
今後の動向	N/A

実装技術スタック詳細		
Credential Layer	証明書モデル	Verifiable Credentials Data Model v1.1
	Credential フォーマット	JSON-LD + LD-Signatures
	選択的開示の実現方式	LD-Signature+BBS+、ZKP
	署名方式	EdDSA、BBS+
	検証者に提供する証明書モデル	Verifiable Presentations
Agent Layer	エンティティ間の通信プロトコル	DIDcomm V2
	通信I/F (デバイス連携)	QRコード
Public Trust Layer	システムアクセスのための識別子	did:indy
	DID Document格納方式 (ストレージ)	OrgBook BC (Hyperledger Indy)
	名前解決	Universal Resolver

出所：<https://digital.gov.bc.ca/digital-trust/about/about-bc-wallet/>
<https://www.hyperledger.org/blog/bc-digital-trust-leveraging-hyperledger-tools-for-digital-trust>

6. 実装パターンの抽出

6.1. サービス – 6.1.2. サービス実装詳細 (6) Nothern Block

- Nothern Blockは、法人向け・個人向けのウォレットサービスを提供、EU ARFを受けて、直近OID4VC系の対応とそれに合わせたCredential フォーマット(JWT-VC、JSON-LD)の追加がされた

サービスの特徴	
開発元	Nothern Block
サービス概要	<ul style="list-style-type: none"> 企業向けのデジタル証明書管理・発行プラットフォーム・Webベースのウォレットサービス(Orbit Enterprise)と個人向けウォレットサービス(Orbit Edge Wallet)を提供
利用事例	<ul style="list-style-type: none"> 鉱業関連 (鉱業データ・鉱山事業資格検証)
関与/参照している主要規格団体・フレームワーク	<ul style="list-style-type: none"> W3C (W3C VC) Hyperledger Foundation (AnonCreds / Indy / Aries) 等 OIDF (OID4VC)
今後の動向	N/A

実装技術スタック詳細		
Credential Layer	証明書モデル	Verifiable Credentials Data Model v1.1
	Credential フォーマット	JWT-VC、JSON-LD、AnonCred
	選択的開示の実現方式	AnonCred ZKPs
	署名方式	CL(Ver.1.0) BBS+ (Ver.2.0)
	検証者に提供する証明書モデル	Verifiable Presentations / OID4VP
Agent Layer	エンティティ間の通信プロトコル	DIDcomm V2 / SIOPv2
	通信I/F (デバイス連携)	QRコード
Public Trust Layer	システムアクセスのための識別子	did:indy (+ link secrets)
	DID Document格納方式 (ストレージ)	Hyperledger indy
	名前解決	Universal Resolver

出所：<https://northernblock.io/orbit-enterprise/>
<https://northernblock.io/blog/interoperability-update-addition-of-openid4vc-to-northern-block-products/>

6. 実装パターンの抽出

6.1. サービス – 6.1.2. サービス実装詳細 (7) Dock Certs

- Dock Certsは独自のブロックチェーンを活用して資格証明書発行・検証にかかるノーコードプラットフォームサービスを提供しており、学歴やスキル証明サービスを提供している事業者に対してサービス提供を行っている。

サービスの特徴	
開発元	Dock Certs
サービス概要	<ul style="list-style-type: none"> 組織が検証可能な資格情報を効率的かつ安全に発行、検証、管理、および取り消すことを可能にする、ユーザーフレンドリーなノーコードプラットフォームを事業者向けに提供
利用事例	<ul style="list-style-type: none"> BurstIQ 従業員の身元情報・健康情報・業績等を証明書にして管理・データ分析するサービスを提供 Gravity 高所作業にかかる資格証明・健康情報証明書検証サービスを提供 SEVENmile デジタル卒業証明書発行サービスの提供 (オーストラリアニューサウスウェールズ州教育局と提携して2024年までに1,500の高校に導入予定)
関与/参照している主要規格団体・フレームワーク	<ul style="list-style-type: none"> W3C/DIF/IETFの仕様に対応
今後の動向	2024年第2四半期にウォレット型SDKを提供予定

実装技術スタック詳細		
Credential Layer	証明書モデル	Verifiable Credentials Data Model v1.0
	Credential フォーマット	LDP – VC
	選択的開示の実現方式	BBS+
	署名方式	Ed25519Signature2018、BBS+
	検証者に提供する証明書モデル	Verifiable Presentations
Agent Layer	エンティティ間の通信プロトコル	DIDcomm V2
	通信I/F (デバイス連携)	QRコード
Public Trust Layer	システムアクセスのための識別子	did:dock / did:key
	DID Document格納方式 (ストレージ)	Substrate (独自チェーン)
	名前解決	Universal Resolver

出所：<https://www.dock.io/>

6. 実装パターンの抽出

6.1. サービス – 6.1.2. サービス実装詳細 (8) Blockcert

- Blockcertは、ブロックチェーンを台帳として活用した資格証明書管理サービスを提供しており、主に学歴証明・専門職の資格証明に関連したサービスを提供している

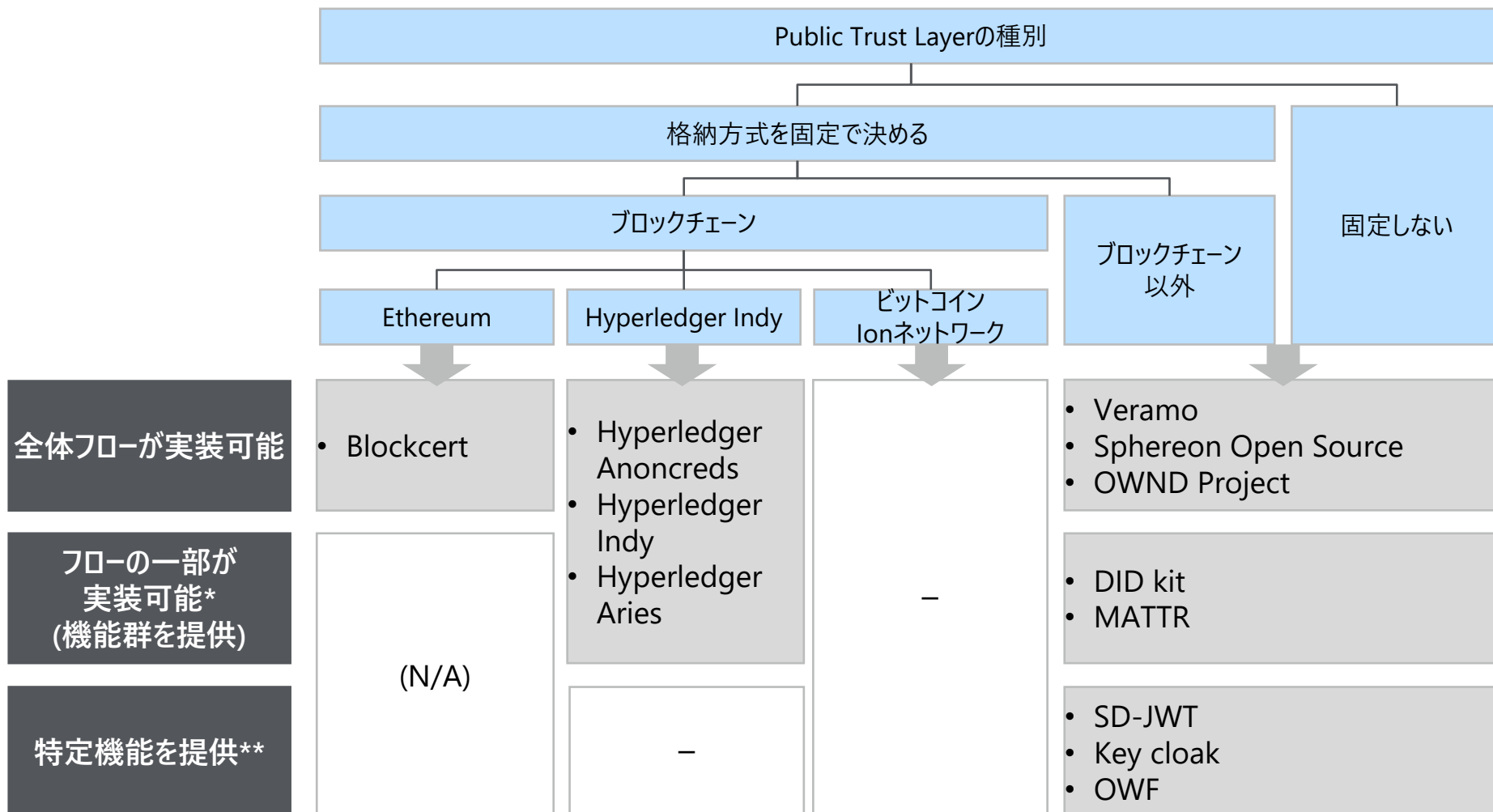
サービスの特徴	
開発元	MIT Media Lab、Learning Machine（現在 Hyland Credential）が共同開発
サービス概要	<ul style="list-style-type: none"> ブロックチェーンベースの公式記録を発行および検証するアプリを構築するためのオープンスタンダード、主に学歴証明書で活用されている
利用事例	<ul style="list-style-type: none"> MIT等の教育機関 学歴証明書として活用 マルチ 生涯にわたる学習履歴を証明書として1か所に保存、教育機関・企業に対して学習記録を提示するサービスを政府として提供 Federation of State Medical Boards(FSMB) 医師資格の資格証明として活用
関与/参照している主要規格団体・フレームワーク	<ul style="list-style-type: none"> W3Cの仕様に準拠 (Verifiable Claims / Linked Data Signatures / Rebooting Web of Trust Decentralized Identifiers)
今後の動向	<ul style="list-style-type: none"> 対応ブロックチェーン基盤の追加 失効モデルの柔軟性向上 (発行者の失効権限の分散化等)

実装技術スタック詳細			
Credential Layer	証明書モデル	Verifiable Credentials Data Model v1.1	
	Credential フォーマット	LDP – VC	
	選択的開示の実現方式	(選択的開示は実装検討中)	
	署名方式	ECDSA	
	検証者に提供する証明書モデル	Verifiable Presentations	
Agent Layer	エンティティ間の通信プロトコル	DIDcomm	
	通信I/F (デバイス連携)	QRコード	
Public Trust Layer	システムアクセスのための識別子	did:web	
	DID Document格納方式 (ストレージ)	Ethereum	Bitcoin
	名前解決	Universal Resolver	

出所：<https://www.blockcerts.org/>

6.2. ライブラリ – 6.2.1 ライブラリ全体観

- DID/VCに関わるライブラリ、プラットフォームはVCモデル等の一連の処理全体をサポートするものから、証明書発行など各エンティティが提供する一部の機能をサポートするもの、選択的開示機能等特定領域をサポートするものに大別している
- ライブラリの大部分はプラットフォーム（基盤部分を担うPublic Trust Layer）への依存度が高く、特定のブロックチェーンを前提としたもの、複数のブロックチェーンに汎化したもの、ブロックチェーンを使用しないケースも考慮したものに分類できる



6. 実装パターンの抽出

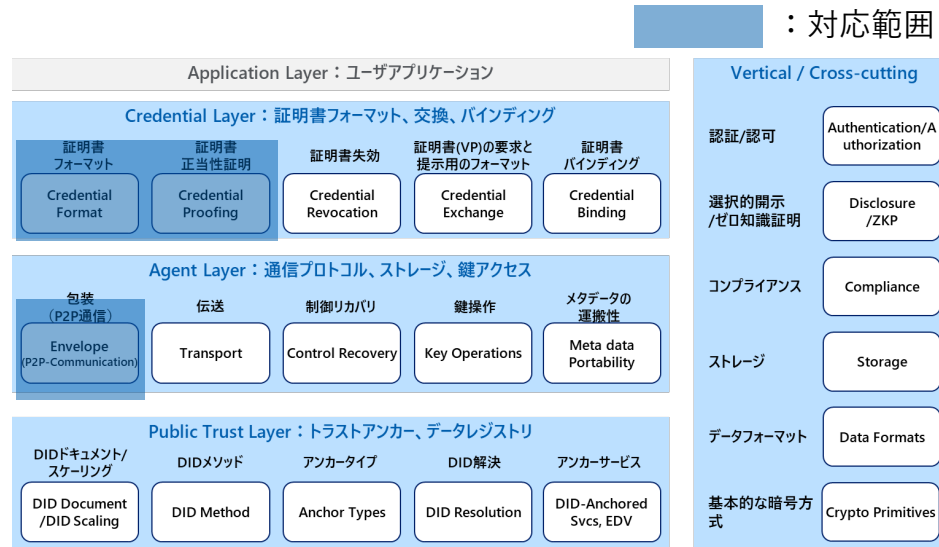
6.2. ライブラリ – 6.2.2. ライブラリ詳細 (1) Blockcert

- ブロックチェーンベースの公式記録を発行および検証するアプリを構築するためのオープンスタンダード
- オープンソースのライブラリ、ツール、モバイル アプリで構成

サービスの特徴

公開元	MIT Media Lab、Hyland-credentials
ライブラリ名/内容	<ul style="list-style-type: none"> • cert-schema スキーマと仕様、スキーマとJSON-LDを検証するためのpythonライブラリ • cert-issuer BitcoinまたはEthereum上で証明書を発行するためのPythonライブラリ • cert-verifier-js Node.jsアプリやブラウザで使用する検証用のJavascriptライブラリ • blockcerts-verifier スタンドアローンの検証ツール • wallet-iOS iOS用ウォレット実装 • wallet-android Android用ウォレット実装
提供形態	ソースコード
ドキュメント	https://www.blockcerts.org/guide/
ライセンス	MIT License

技術スタック

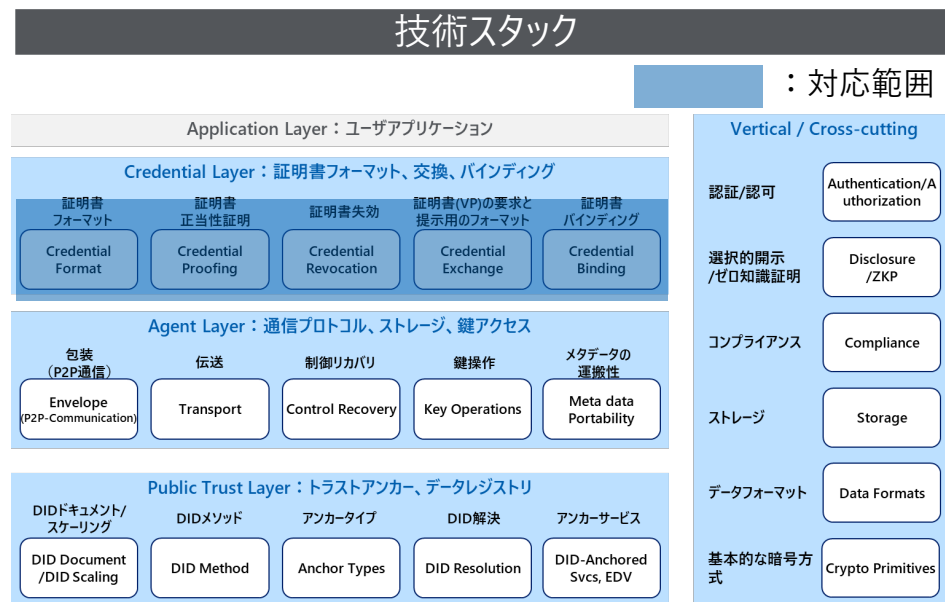


6. 実装パターンの抽出

6.2. ライブラリ – 6.2.2. ライブラリ詳細 (2) Hyperledger AnonCreds

- 1985年から検討が続いている仕様であるが、2016年以降はHyperledgerで検討が続けられている
- AnonCreds v1.0はCL署名に基づいており、AnonCreds v2.0はBBS+署名に基づいた実装となっている

サービスの特徴	
公開元	Linux Foundation (Hyperledger Project)
ライブラリ名/内容	<ul style="list-style-type: none"> • anoncreds-rs 重要なプライバシー保護機能であるZKP (ゼロ知識証明) 機能をコア VC 保証に追加する VC の一種。台帳やクライアントに依存せず、Hyperledger IndyやAries とは独立して動くことを想定して設計されているため、他の検証可能なデータレジストリ/台帳および検証可能な資格情報クライアントとともに使用できる
提供形態	ソースコード
ドキュメント	https://hyperledger.github.io/anoncreds-spec/
ライセンス	<ul style="list-style-type: none"> • Apache License Version 2.0

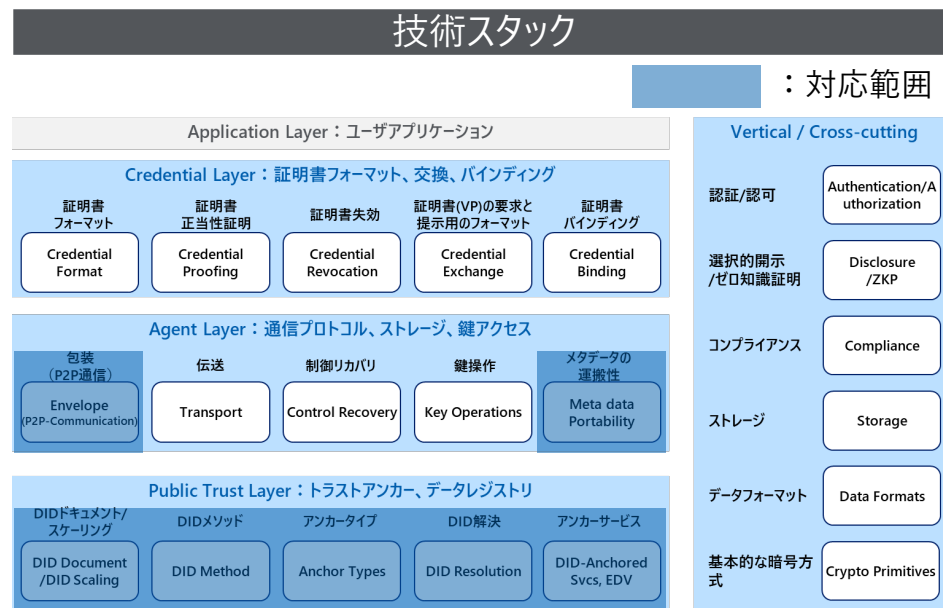


6. 実装パターンの抽出

6.2. ライブラリ – 6.2.2. ライブラリ詳細 (3) Hyperledger Indy

- 分散型アイデンティティ専用のブロックチェーン基盤であるHyperledger Indyに接続するライブラリ群
- Hyperledger Indyの公式SDKであったIndy SDKが非推奨となり、個別の実装に置き換わりつつある状況

サービスの特徴	
公開元	Linux Foundation (Hyperledger Project)
ライブラリ名/ 内容	<ul style="list-style-type: none"> • Indy-VDR (Verifiable Data Registry) Indy Nodeとの接続モジュール • Aries Askar Wallet機能の実装 • indy-cli-rs Indyのコマンドラインインターフェイス
提供形態	ソースコード
ドキュメント	https://hyperledger.github.io/indy-did-method/
ライセンス	<ul style="list-style-type: none"> • Apache License Version 2.0 • MIT License

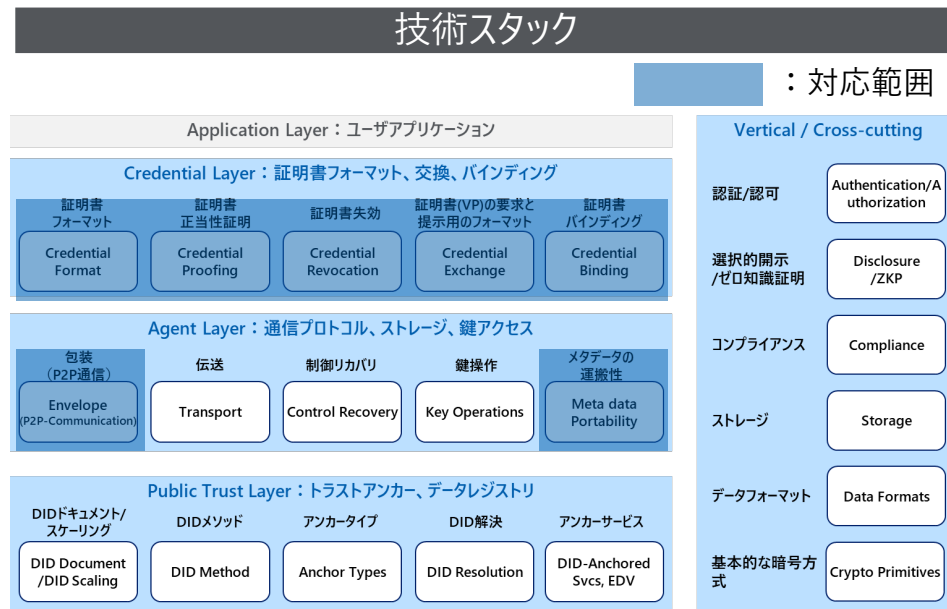


6. 実装パターンの抽出

6.2. ライブラリ – 6.2.2. ライブラリ詳細 (4) Hyperledger Aries

- エンティティ間で相互に信頼できるP2P接続を行うコンポーネント群（ユーザエージェント、DID通信、キー管理、プロトコル）で構成されている
- 複数のプロジェクトが立ち上がっており、現在も活発に活動中。ただし、更新を停止し、アーカイブに格納されたプロジェクトも多数存在(モバイル用途に開発された.NET版やRuby版、テストフレームワークなど)
- 共有ライブラリとして、ストレージプラグインを含む鍵管理、データリポジトリと接続するインターフェイス、各種ユーティリティ等を提供するCライブラリを公開予定（現在はフレームワークごとに個別実装）

サービスの特徴	
公開元	Linux Foundation (Hyperledger Project)
ライブラリ名/内容	<ul style="list-style-type: none"> • Hyperledger Aries Cloud Agent Python (ACA-Py) Pythonで実装された非モバイル環境を想定したアプリケーション構築基盤 • Aries Framework Go (AFG) ACA-pyのGo実装 • Static Agent Library (SAL) IoT利用を想定
提供形態	ソースコード
ドキュメント	https://hyperledger.github.io/indy-did-method/
ライセンス	• Apache License Version 2.0

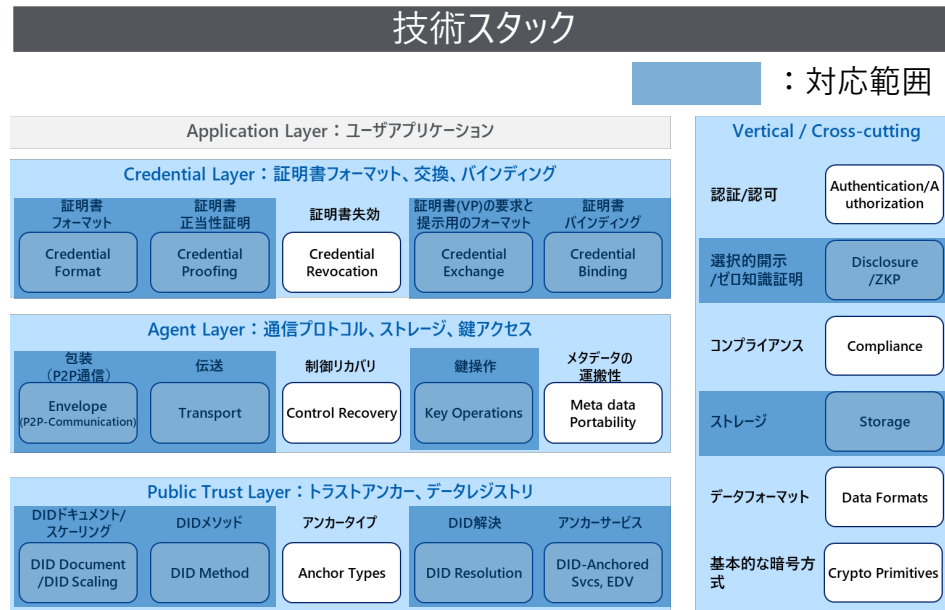


6. 実装パターンの抽出

6.2. ライブラリ – 6.2.2. ライブラリ詳細 (5) Veramo

- Ethereumを利用した分散型デジタルIDサービス「uPort」から分割して設立されたプロジェクト。W3CとDIFと協力して開発されており、DIFのGithubにソースコードが存在する。更新頻度は高い。
- 選択的開示は開発中でSD-JWT、JSON-LD BBS+が検討されている（SD-JWTはソースコードが存在）。

サービスの特徴	
公開元	Veramo
ライブラリ名/内容	<ul style="list-style-type: none"> • Veramo Veramo は、検証可能なデータとSSIのためのJavaScript フレームワーク。柔軟なモジュール式に設計されており、プラグインによって拡張可能。以下の機能に対応されている <ul style="list-style-type: none"> - 署名と暗号化のためのキーの作成と管理 - DIDの作成と管理 - VCとVPの発行 - 選択的開示による資格情報の提示 - DIDCommによるエージェント間の通信データの受信、フィルタリング、保存、提供
提供形態	ソースコード
ドキュメント	https://veramo.io/docs/basics/introduction
ライセンス	<ul style="list-style-type: none"> • Apache License Version 2.0

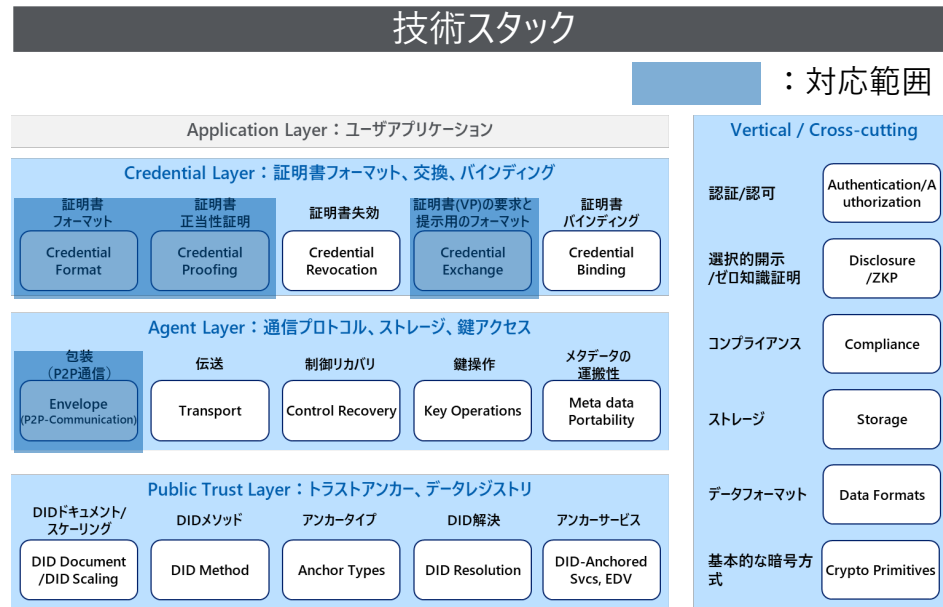


6. 実装パターンの抽出

6.2. ライブラリ – 6.2.2. ライブラリ詳細 (6) Sphereon Open Source

- 行政、医療、臨床試験、モビリティ、教育、その他の業界向けのデータ交換ソリューションであり、ウォレット機能、SIOPv2に準拠したデータ交換、eIDASに準拠した署名クライアント、SSI SDK等を提供している。
- 選択的開示機能はまだ実装されていないものの、BSS+署名ベースの検討はされている模様。更新頻度は高い

サービスの特徴	
公開元	Sphereon Open Source
ライブラリ名/内容	<ul style="list-style-type: none"> • SSI-SDK • OID4VCI • SIOP-OID4VP • ssi-mobile-wallet • eidas-signature-client • Presentation Exchange v1 and v2 TypeScript Library
提供形態	ソースコード
ドキュメント	https://github.com/Sphereon-Opensource/SSI-SDK
ライセンス	<ul style="list-style-type: none"> • Apache License Version 2.0



6. 実装パターンの抽出

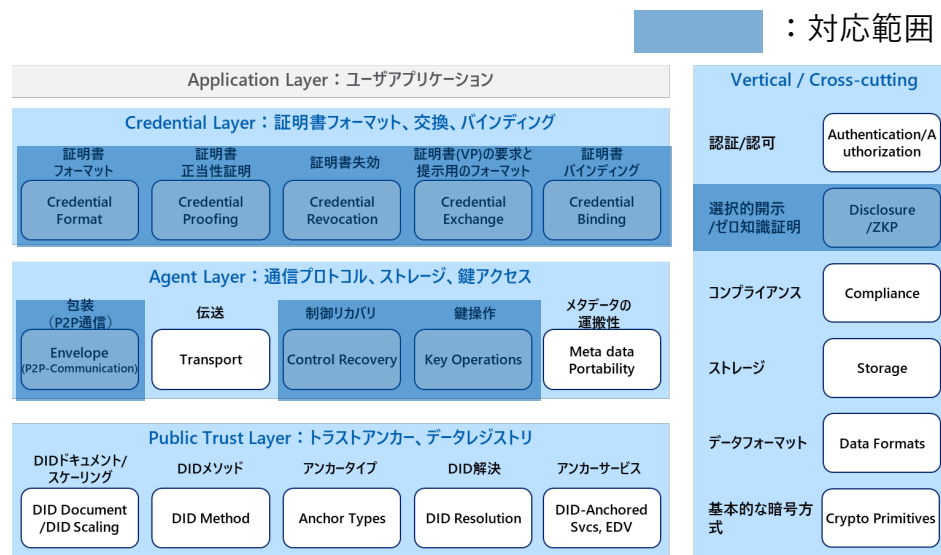
6.2. ライブラリ – 6.2.2. ライブラリ詳細 (7) OWND Project

- 国際標準技術に準拠したホワイトラベルのデジタルアイデンティティウォレット。OID4VCI / OID4VP / SIOP v2に対応しており、iOS用とAndroid用が提供されている。選択的開示機能としてSD-JWTに対応。JSON-LD ZKP with BBS+も対応予定
- OID4VCIに準拠したデジタルアイデンティティ発行サービス（OWND Project VCI）とマイナンバーカード情報、従業員ID、イベント参加証明書の3つのWebアプリケーションが含まれる
- OWND walletを用いてアイデンティティを管理できるE2E暗号化に対応したメッセージングアプリケーションを提供している（サーバ機能、クライアント機能、React SDK）

サービスの特徴

公開元	OWND Project
ライブラリ名/内容	<ul style="list-style-type: none"> OWND Wallet ホワイトラベルのデジタルアイデンティティウォレット <ul style="list-style-type: none"> OWND-Wallet-iOS OWND-Wallet-Android OWND-Project-VCI デジタルアイデンティティ発行サービスとサンプルコード <ul style="list-style-type: none"> OWND-Project-VCI OWND Messenger アイデンティティ管理可能なメッセージングアプリ <ul style="list-style-type: none"> OWND-Messenger-Server OWND-Messenger-Client OWND-Messenger-React-SDK
提供形態	ソースコード
ドキュメント	https://github.com/OWND-Project/whitepaper
ライセンス	<ul style="list-style-type: none"> MIT License

技術スタック

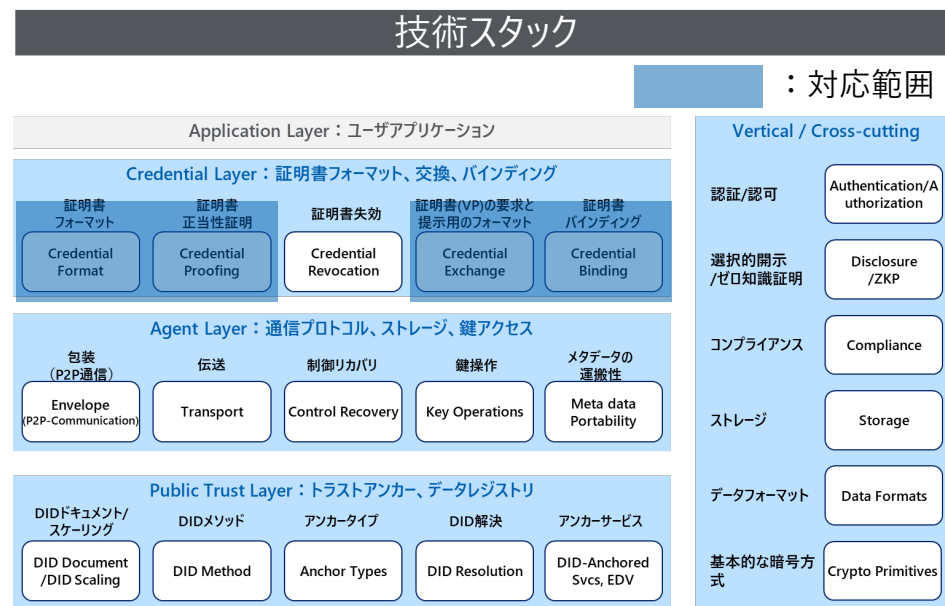


6. 実装パターンの抽出

6.2. ライブラリ – 6.2.2. ライブラリ詳細 (8) DID Kit

- W3C検証可能クレデンシャルの署名・検証、多言語ライブラリサポート、HTTP/HTTPSサーバー提供、Linked Data ProofsとJOSEトークン間の変換、多様なW3C分散型識別子の処理、およびオブジェクト機能モデルに基づく認証トークンの発行と使用が可能なツール。VC-APIに対応したサーバ用コンテナも提供する。
- 更新頻度は比較的高く、コマンドライン、HTTPサーバおよびモバイル環境としてC, Java, AndroidおよびFlutterでの利用が想定されている。

サービスの特徴	
公開元	Spruce Systems Inc.
ライブラリ名/ 内容	<ul style="list-style-type: none"> • DIDKit
提供形態	ソースコード
ドキュメント	https://www.spruceid.dev/didkit/didkit
ライセンス	<ul style="list-style-type: none"> • Apache License Version 2.0

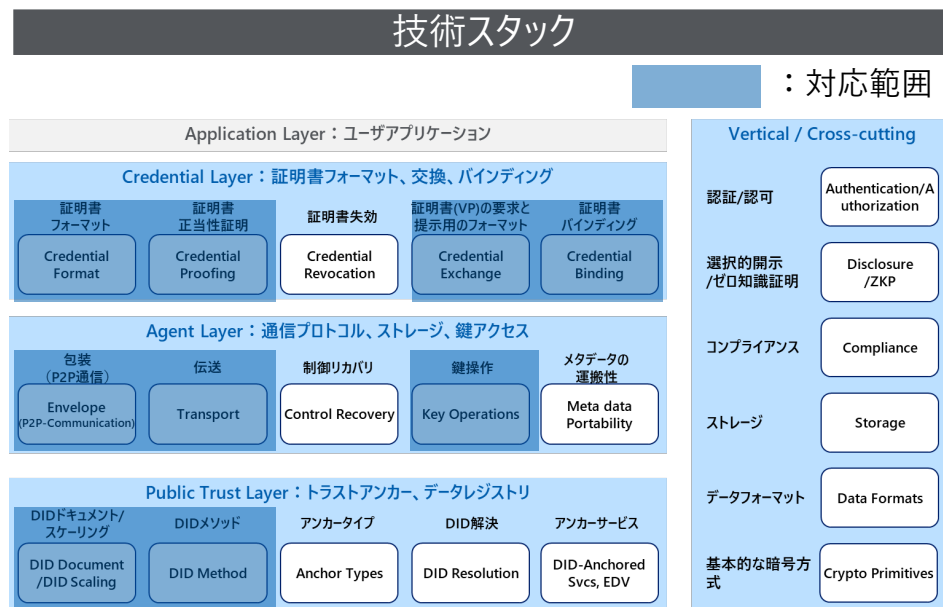


6. 実装パターンの抽出

6.2. ライブラリ – 6.2.2. ライブラリ詳細 (9) MATTR

- 検証可能なデータと検証可能な資格情報と連携し、情報を安全に共有、保持、検証するための製品群を提供している。
- 製品はサービス（REST API）、SDKで提供されており、ソースコードでの提供ではないが、JSON-LD+BBS+署名などのサンプルコードは公開されている。

サービスの特徴	
公開元	MATTR Platform
ライブラリ名/ 内容	<ul style="list-style-type: none"> • MATTR VII : 検証可能な認証情報を生成・管理するためのAPIと各種機能を備えたクラウドプラットフォーム • MATTR Pi : ウォレットと検証機能を開発するためのSDK <ul style="list-style-type: none"> - キーと分散型識別子 (DID) の作成と管理 - OpenID資格情報プロバイダー - 検証機能 - BBS+署名による選択的開示 - DIDベースのメッセージング - カスタマイズ可能なユーザーエクスペリエンス - iOSとAndroidで利用可能な共通コード • MATTR GO : 独自のブランディング、色、タイポグラフィなどを使用してユーザーエクスペリエンスをカスタマイズできるプラットフォーム
提供形態	SDK + SaaSサービス
ドキュメント	https://learn.mattr.global/docs/
ライセンス	• MATTR



6. 実装パターンの抽出

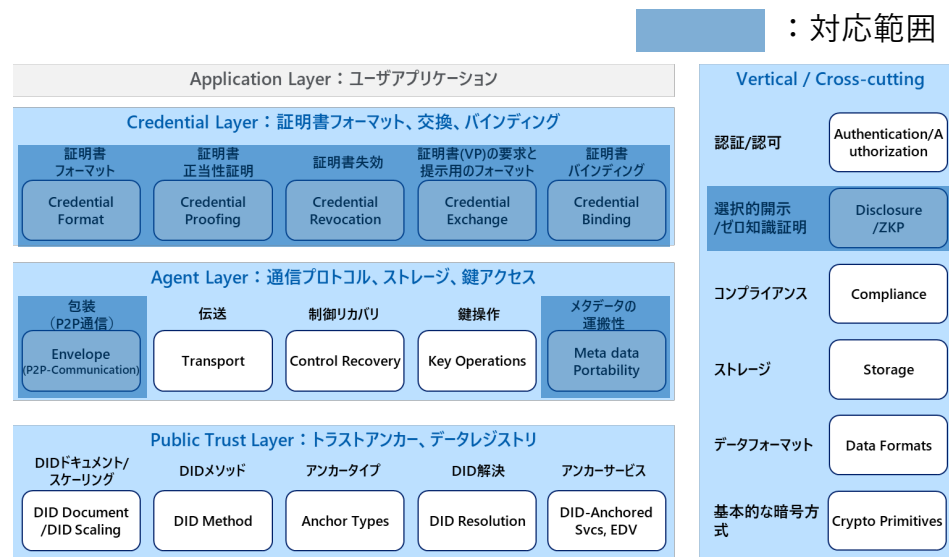
6.2. ライブラリ – 6.2.2. ライブラリ詳細 (10) OpenWallet Foundation

- デジタル ID ウォレットを構築するためのオープン フレームワーク。OID4VCおよびSD-JWTをサポートする。また、OID4VC/OID4VP / SIOP v2に対応している。
- Hyperledgerで取り組まれていたAries Framework .NETおよびAries Framework Java ScriptがOpenWallet Foundationに移管されて移行作業中である

サービスの特徴

公開元	OpenWallet Foundation
ライブラリ名/ 内容	<ul style="list-style-type: none"> • Wallet Framework for .NET • Aries Framework - JavaScript(AF-JS)
提供形態	ソースコード
ドキュメント	https://aries-cloud-agent-python.readthedocs.io/en/latest/
ライセンス	• Apache License Version 2.0

技術スタック

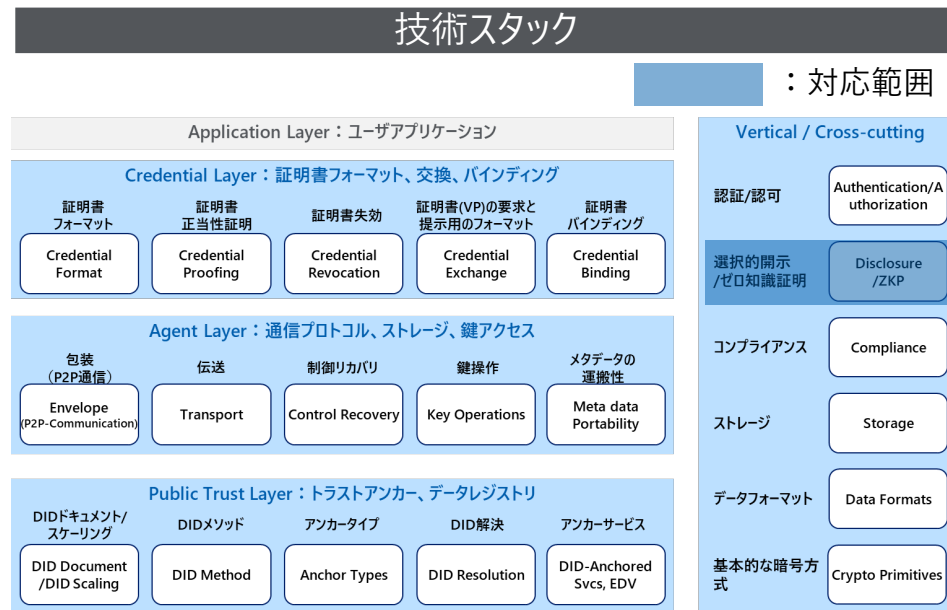


6. 実装パターンの抽出

6.2. ライブラリ – 6.2.2. ライブラリ詳細 (11) SD-JWT (OpenWallet Foundation)

- IETF SD-JWT 仕様の例を生成するために使用され、SD-JWT を実装するための他のプロジェクトでも使用可能、複数の言語に対応

サービスの特徴	
公開元	OpenWallet Foundation
ライブラリ名/ 内容	<ul style="list-style-type: none"> • SD-JWT Python Reference Implementation • SD-JWT Rust Reference Implementation • SD-JWT Implementation in JavaScript (TypeScript) • SD-JWT-DotNet • SD-JWT Implementation in Kotlin
提供形態	ソースコード
ドキュメント	https://github.com/openwallet-foundation-labs
ライセンス	• Apache License Version 2.0

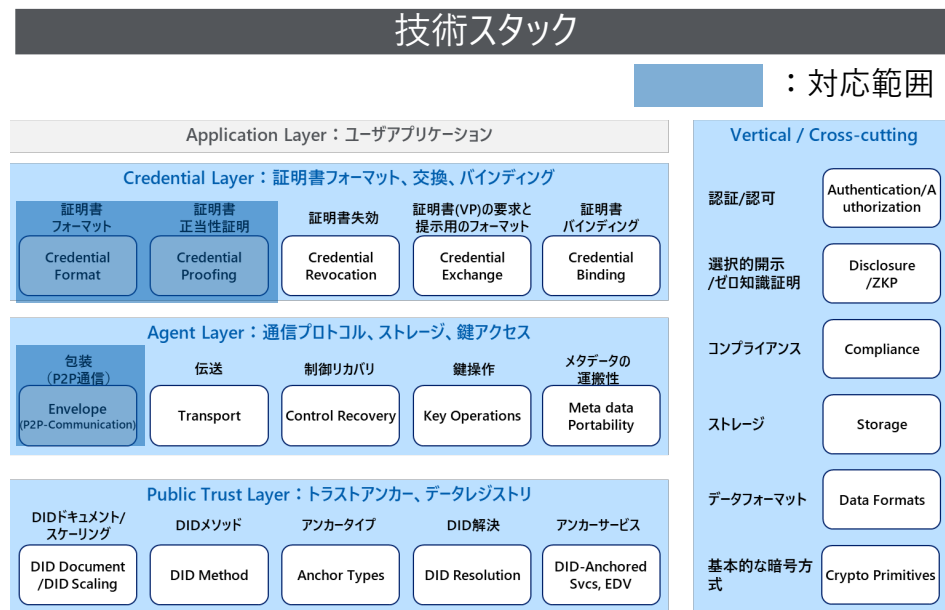


6. 実装パターンの抽出

6.2. ライブラリ – 6.2.2. ライブラリ詳細 (12) Keycloak

- KeycloakはOIDCに準拠したアイデンティティ管理ミドルウェアであり、SIOP-2 / OIDC4VP クライアントをサポートし、OIDC4VCI プロトコルを通じて 準拠ウォレットにVerifiable Credentials を発行するためのKeycloakのプラグインである。
- FIWAREは、欧州連合（EU）のICTプロジェクトとして、2011年からの5年間に実施された次世代インターネット官民連携プログラム（FI-PPP）において開発されている。V8.4.0から認証ミドルウェアのKeycloakにVC機能を持たせたモジュールが提供されている。

サービスの特徴	
公開元	Keycloak (Cloud Native Computing Foundation)
ライブラリ名/ 内容	<ul style="list-style-type: none"> • keycloak-vc-issuer
提供形態	ソースコード
ドキュメント	https://www.keycloak.org/guides
ライセンス	<ul style="list-style-type: none"> • Apache License Version 2.0



すべてを突破する。
TOPPA!!!
TOPPAN