



# Trusted Digital Ecosystems プロセス・アーキテクチャ

エンタープライズ・エージェント：発行者と検証者のワークフロー

# 目次

- トラスト・トライアングルにおけるエンタープライズ・エージェント
- VCの発行と検証
- エンタープライズ・エージェントの詳細
- 実例：Proven API
- フローの設計
- デモ



トラスト・トライアングルにおけるエンタープライズ・エージェント

# トラスト・トライアングル

VC（検証可能な資格情報）は、データを送受信しなければならないときに、最も価値が生まれる

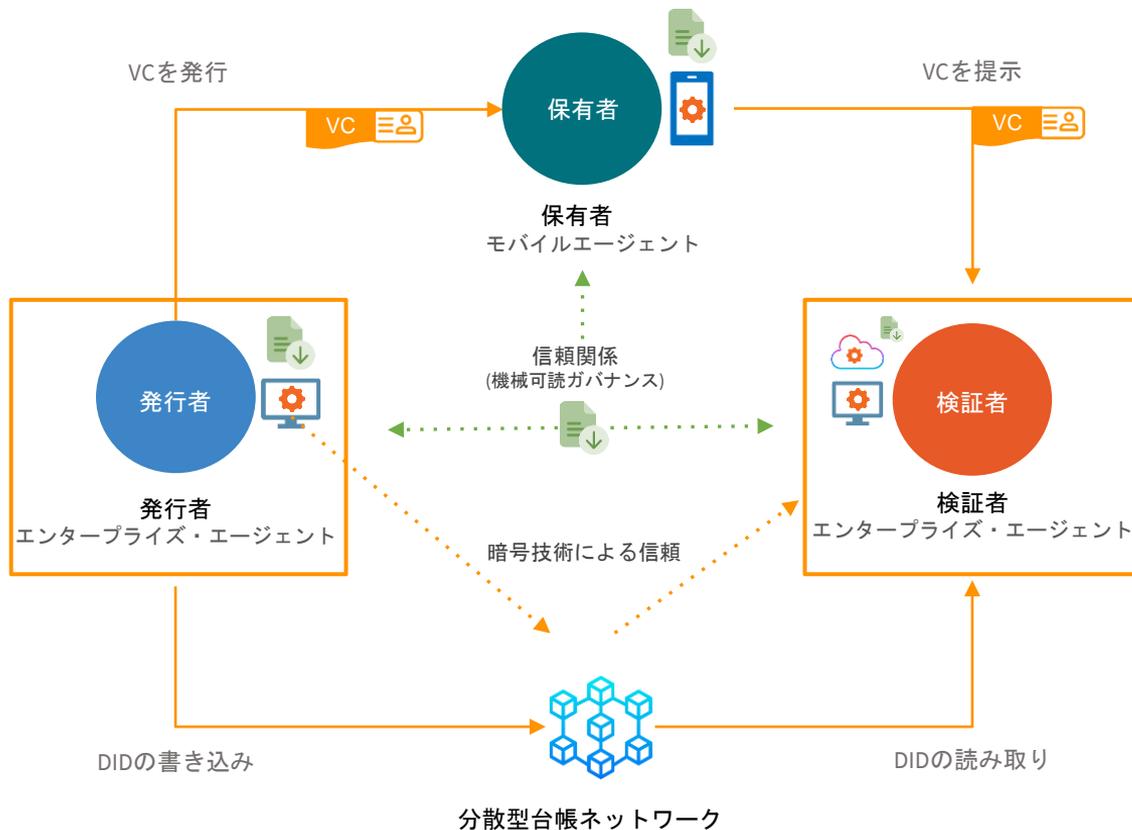
- 送信者は、どのようにしてデータを安全に共有し、法律を遵守できるのか？
- 個人またはデータ主体は、どのように自分のデータを管理できるのか？
- 受信側は、データが正確であることをどうやって確信できるのか？

# トラスト・トライアングル

エコシステムにおけるVCとは：

- 発行者は、データを永遠に保管する必要なく、安全にデータを共有できる。
- 保有者は、データを共有することに同意する。
- 検証者は、データの完全性を確信できる。
- すべてのエコシステム参加者は、困難なシステム統合を避け、即座にデータを活用できる。

# トラスト・トライアングルにおけるエンタープライズ・エージェント





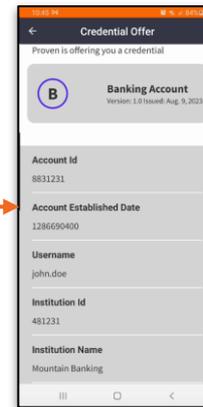
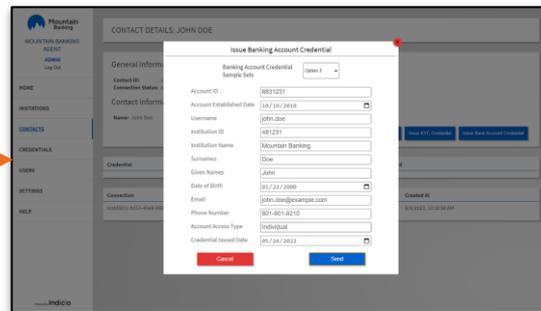
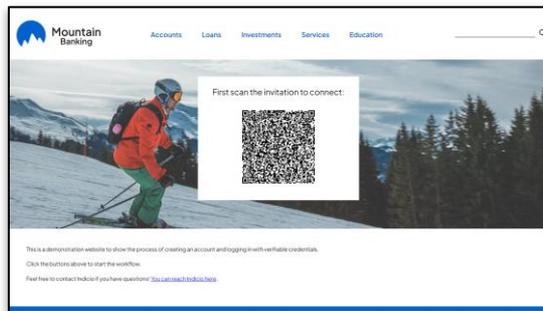
# VCの発行と検証

# VCの発行

1. QRコードや招待用URLを使用して、他のエージェントと接続する。
2. データソースと統合して、自動プログラムによって属性情報を入力する。

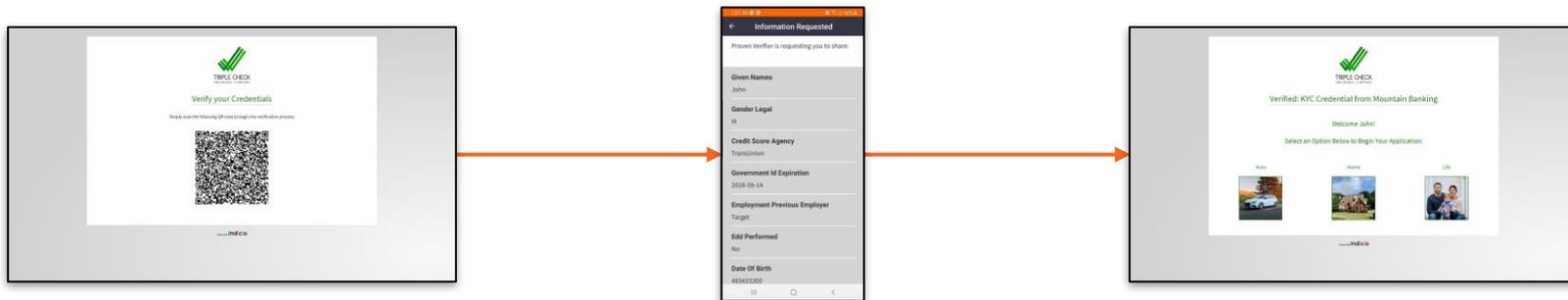
または

入力フォームや他のインターフェースを使用して、属性情報を手動で入力する。



# VCの検証

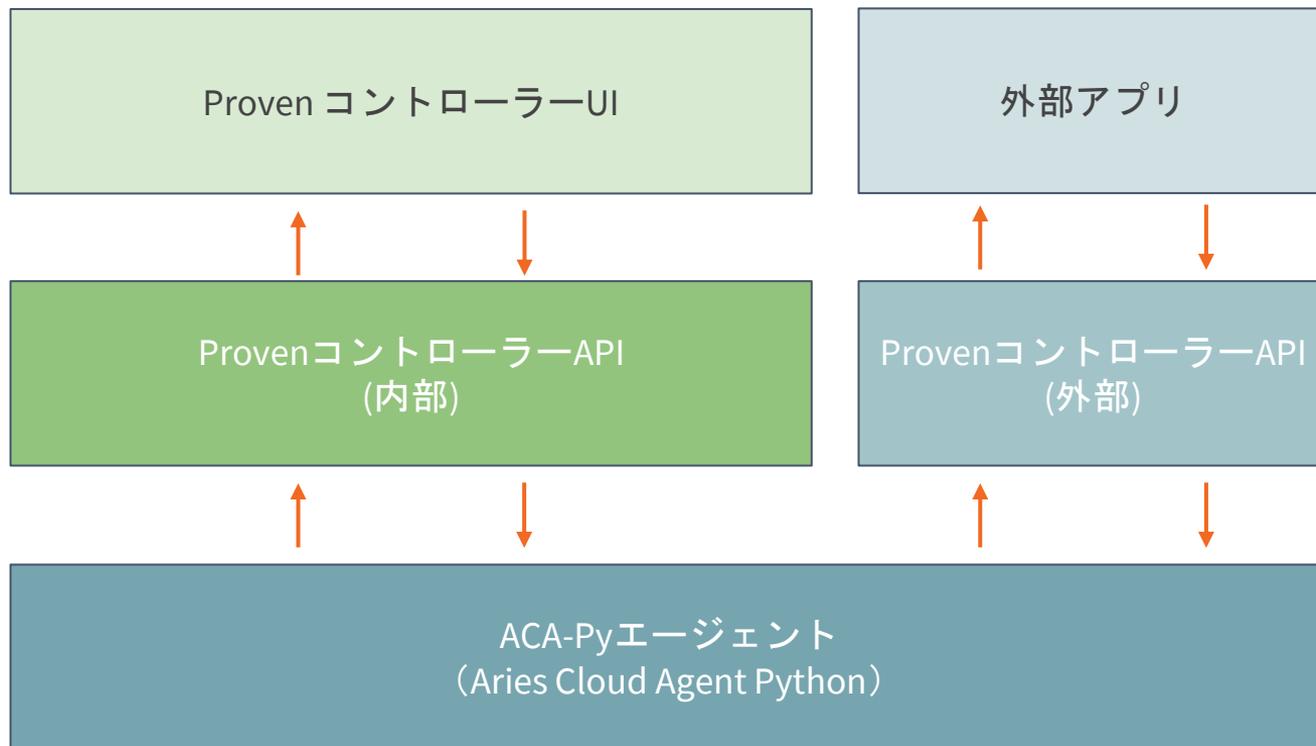
1. QRコードや招待用URLを使用して、他のエージェントに接続する
2. VPの提示をリクエストして、必要な属性情報の提供を依頼する
3. 完全開示または選択的開示、ゼロ知識証明を選べる
4. VPを受け取ったら、台帳上の情報とVPを使って、データを復号し検査する。
5. 最後に、真偽値のレスポンスが返ってくる。





# エンタープライズ・エージェントの詳細

# Indicioのエンタープライズ・エージェントの構造



# Indicio社のエンタープライズ・エージェントの構造

## ACA-Py (Aries Cloud Agent Python)

- PythonとPostgresで構築される
- よく整備されたオープンソース
- 機能が豊富
- Wallet機能とエージェント機能を提供
- APIを含む
- 接続やインタラクションのまとまった記録を作成しない
- ユーザーを管理しない
- UIを提供しない

# Indicio社のエンタープライズ・エージェントの構造

ProvenコントローラーAPI（内部および外部）

- Node.jsおよびExpress.js; Postgresデータベース
- データの永続性を管理する（接続先の情報）
- ユーザーとアクセスの管理
- 外部APIによって、WEBやレガシーアプリケーションがワークフローを推進し、エージェントの機能やデータにアクセス可能になる
- Indicio独自のコードベースだが、将来的にはオープンソースになる予定

# Indicio社のエンタープライズ・エージェントの構造

## ProvenコントローラーUI

- React.js
- 使いやすいUI
- システムを管理する
- Indicio独自のコードベースだが、将来的にはオープンソースになる予定

# エンタープライズ・エージェントソフトウェアのインストール

1. proven-docker のリポジトリを複製する
2. システム構成を更新する（一部のポートの設定を含む）
3. ファイルシステム内のいくつかのファイルとUIの設定を手動で行う
4. 作成者/承認者として設定する（DID、鍵、NYMトランザクションの台帳へ格納）
5. スキーマを格納する（通常は、すでに格納されているスキーマを使用）
6. CredDef（DIDをスキーマに連結するクレデンシャルの定義）を作成する
7. エージェントを使用開始！



# 实例：Proven API

# Proven APIの特徴

- Proven API は、エージェントの操作を外部アプリや既存システムから利用可能にする
- 重要なAPIは「Connect」「Issue」「Verify」の各エンドポイントにある
- 標準的なRESTful HTTPメソッドを使用して統合する
- 認証は、現在、API keyを介して行われている

# Proven API : Connect

コールのサンプル- /api/v1/invitations

```
{  
  "contact_id": "",  
  "alias": "API Invitation",  
  "invitation_type": "CV1",  
  "invitation_mode": "once",  
  "accept": "auto",  
  "public": false,  
  "invitation_role": "Holder",  
  "invitation_label": "CV1",  
  "invitation_status": "active",  
  "invitation_description": "Invitation created through API",  
  "invitation_active_starting_at": null,  
  "invitation_active_ending_at": null,  
  "uses_allowed": ""  
}
```

# Proven API : Connect

コールのサンプル- /api/v1/invitations

```
{  
  "contact_id" : "", // 基礎コースでは扱わない  
  "alias" : "API Invitation", // このinvitationを参照するための名称  
  "invitation_type" : "CV1", // invitationのプロトコル  
  "invitation_mode" : "once", // once または multi  
  "accept" : "auto", // auto または manual  
  "public": false, // true または false  
  "invitation_role" : "Holder", // 割り当てる役割  
  "invitation_label" : "CV1", // invitationとともに送られるラベル  
  "invitation_status" : "active", // Provenで設定されるフラグ (active, inactive, deleted)  
  "invitation_description" : "Invitation created through API", // <-- 作成者向け説明  
  "invitation_active_starting_at": null, // アクティブ期間の開始日  
  "invitation_active_ending_at": null, // アクティブ期間の終了日  
  "uses_allowed" : "" // invitationの複数利用の上限  
}
```

# Proven API : Connect

レスポンスのサンプル- /api/v1/invitations

```
{  
  "invitation_url" :  
  "https://dev.mediator.indiciotech.io?c_i=eyJAdHlwZSI6ICJkaWQ6c292OkJ6Q2JzTlloTXJqSGI  
xWkRUVUFTSGc7c3BIYy9jb25uZWNOaW9ucy8xLjAvaW52aXRhdGlvbilzICJAaWQiOiAiN2VmMWI  
1Y2UtZGNmMy00OTM4LThiZjltZmU2YWM2ZTNhMjU4liwgInNlcnZpY2VFbmRwb2ludCI6ICJod  
HRwczovL2Rldi5tZWRpYXRvci5pbmRpY2lvdGVjaC5pbylzICJsYWJlbiCI6ICJQcm92ZW4iLCAicmV  
jaXBpZW50S2V5cyI6IFsiRUROYmFEM3NNVnpKNWNYWVQ0N3Jwc2pCYUJEYmRZUUFaa0R2WH  
ptNFNhV2giXSgwInJvdXRpbmdLZXlzljogWyJDVkdwZ2M4N2FnSFRXVmt4eGFyelQzVWlxTjRmb  
29VRWpIN281eG9ISHBISiIsICI4aUtpU3NwcUtNOENQVUhfQjk3Z2dhMXBvR1l0OEpdU0p2VUdz  
aGpZRWRvll19"、  
  "invitation_id" : 1,  
  "contact_id" : ""  
}
```



# Proven API : Issue

コールのサンプル- /api/v1/credentials

```
{
  "invitation_id": 1,
  "contact_id": "",
  "schema_id": "Q7CyqfHss9RPK4hjwyZT32:2:Email:1.0",
  "attributes": [
    {
      "name": "local_part",
      "value": "john"
    },
    {
      "name": "domain",
      "value": "example.com"
    },
    ...
  ]
}
```

# Proven API : Issue

## コールのサンプル - /api/v1/credentials

```
{
  "invitation_id" : 1, // このinvitationに関連付けられた接続に提供する
  "contact_id" : "", // 招待時にcontact_idを使用した場合は、そのcontact_idを使用できる
  "schema_id" : "Q7CyqfHss9RPK4hjwyZT32:2:Email:1.0", // 提供するVCのスキーマ
  "attributes" : [ // 属性のリスト
    {
      "name" : "local_part", // 属性の名前
      "value" : "john" // 属性の値
    },
    {
      "name" : "domain",
      "value" : "example.com"
    },
    ...
  ]
}
```

# Proven API : Issue

レスポンスのサンプル - /api/v1/credentials

```
{  
  "success": "Credential was offered"  
}
```

```
{  
  "error": "contact_id and invitation_id can't both be null"  
}
```

# Proven API : Issue

レスポンスのサンプル - /api/v1/credentials

```
{  
  "success": "Credential was offered" // VC発行に成功  
}
```

```
{  
  "error": "contact_id and invitation_id can't both be null" // 失敗  
}
```

# Proven API : Verify

コールのサンプル - /api/v1/verifications

```
{
  "invitation_id": 1,
  "contact_id": "",
  "schemas": [
    {
      "schema_id": "Q7CyqfHss9RPK4hjwyZT32:2:Email:1.0",
      "schema_attributes": [
        "local_part",
        "domain",
        "address",
        "verified_at"
      ]
    }
  ],
  "timeout": "15",
  "rule": "no rule"
}
```

# Proven API : Verify

## コールのサンプル - /api/v1/verifications

```
{  
  "invitation_id" : 1, // このinvitationに関連付けられた接続にリクエストする  
  "contact_id" : "", // invitation_idの代わりにcontact_idも指定できる  
  "schemas" : [ // プレゼンテーションのリスト  
    {  
      "schema_id" : "Q7CyqfHss9RPK4hjwyZT32:2:Email:1.0", // VPのschema_id  
      "schema_attributes" : [ // 問い合わせるスキーマ属性  
        "local_part",  
        "domain",  
        "address",  
        "verified_at"  
      ]  
    }  
  ],  
  "timeout" : "15", // ユーザーの応答を待つ時間  
  "rule": "no rule" // 処理ルールの名称（将来の機能をサポートするために含まれている）  
}
```

# Proven API : Verify

## レスポンスのサンプル - /api/v1/verifications

```
[
  {
    "verification_id": 1,
    "connection_id": "682467a0-d0a4-4ab1-92b4-f26b1892d830",
    "contact_id": null,
    "invitation_id": 1,
    "schema_id": "Q7CyqfHss9RPK4hjwyZT32:2:Email:1.0",
    "schema_attributes": [
      "local_part",
      "domain",
      "address",
      "verified_at"
    ],
    "timeout": 15,
    "rule": "no rule",
    "meta_data": null,
    "complete": true,
    "result": true,
    "result_string": "Verified",
  }
]
```

# Proven API : Verify

## レスポンスのサンプル - /api/v1/verifications

```
[
  {
    "verification_id": 1, // この検証のID
    "connection_id": "682467a0-d0a4-4ab1-92b4-f26b1892d830", // このリクエストが送信された connection_id
    "contact_id": null, // 送信した内容の繰り返し
    "invitation_id": 1, // 送信した内容の繰り返し
    "schema_id": "Q7CyqfHss9RPK4hjwyZT32:2:Email:1.0", // このリクエストで要求したスキーマ
    "schema_attributes": [ // 要求した属性のリスト
      "local_part",
      "domain",
      "address",
      "verified_at"
    ],
    "timeout": 15, // 送信した内容の繰り返し
    "rule": "no rule", // 送信した内容の繰り返し
    "meta_data": null, // ルールに関する追加情報
    "complete": true, // この検証は完了したか？
    "result": true, // true または false VPは要件を満たしたか？
    "result_string": "Verified", // "result" と同じ意味のわかりやすいテキスト
  }
]
```

# Proven API : Verify

## レスポンスのサンプル - /api/v1/verifications

```
...
  "result_data": [
    {
      "name": "verified_at",
      "value": "1670296277"
    },
    ...
  ],
  "presentation_exchange_id": [
    "7729a152-5ac1-4664-8d9b-523a1377a261"
  ],
  "error": "",
  "created_at": "2023-08-08T08:38:39.597Z",
  "updated_at": "2023-08-08T08:38:51.040Z"
}
]
```

# Proven API : Verify

## レスポンスのサンプル - /api/v1/verifications

```
...
  "result_data": [// リクエストに応じて送信された属性のリスト
    {
      "name": "verified_at", // VPにおける属性名
      "value": "1670296277" // VPにおける属性の値
    },
    ...
  ],
  "presentation_exchange_id": [// この検証に関連付けられたVPを追跡するのに役立つ重要な ID
    "7729a152-5ac1-4664-8d9b-523a1377a261"
  ],
  "error": "", // エラーがあれば表示
  "created_at": "2023-08-08T08:38:39.597Z", // タイムスタンプ
  "updated_at": "2023-08-08T08:38:51.040Z" // タイムスタンプ
}
]
```

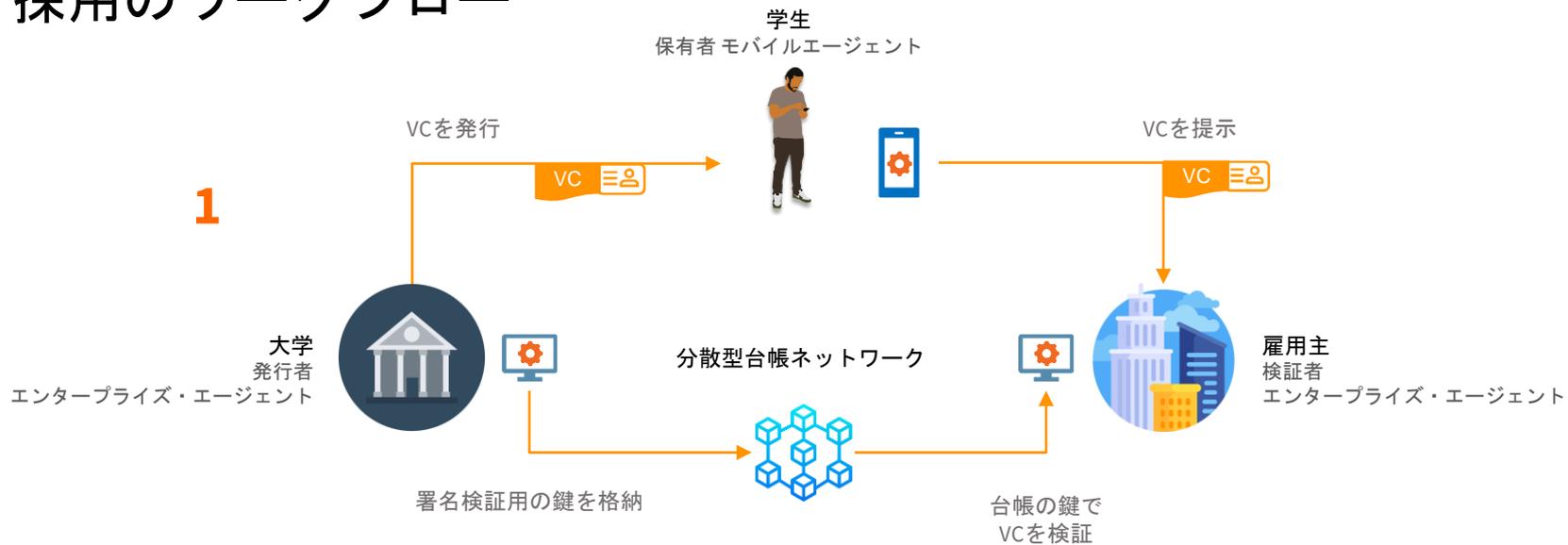


# フローの設計

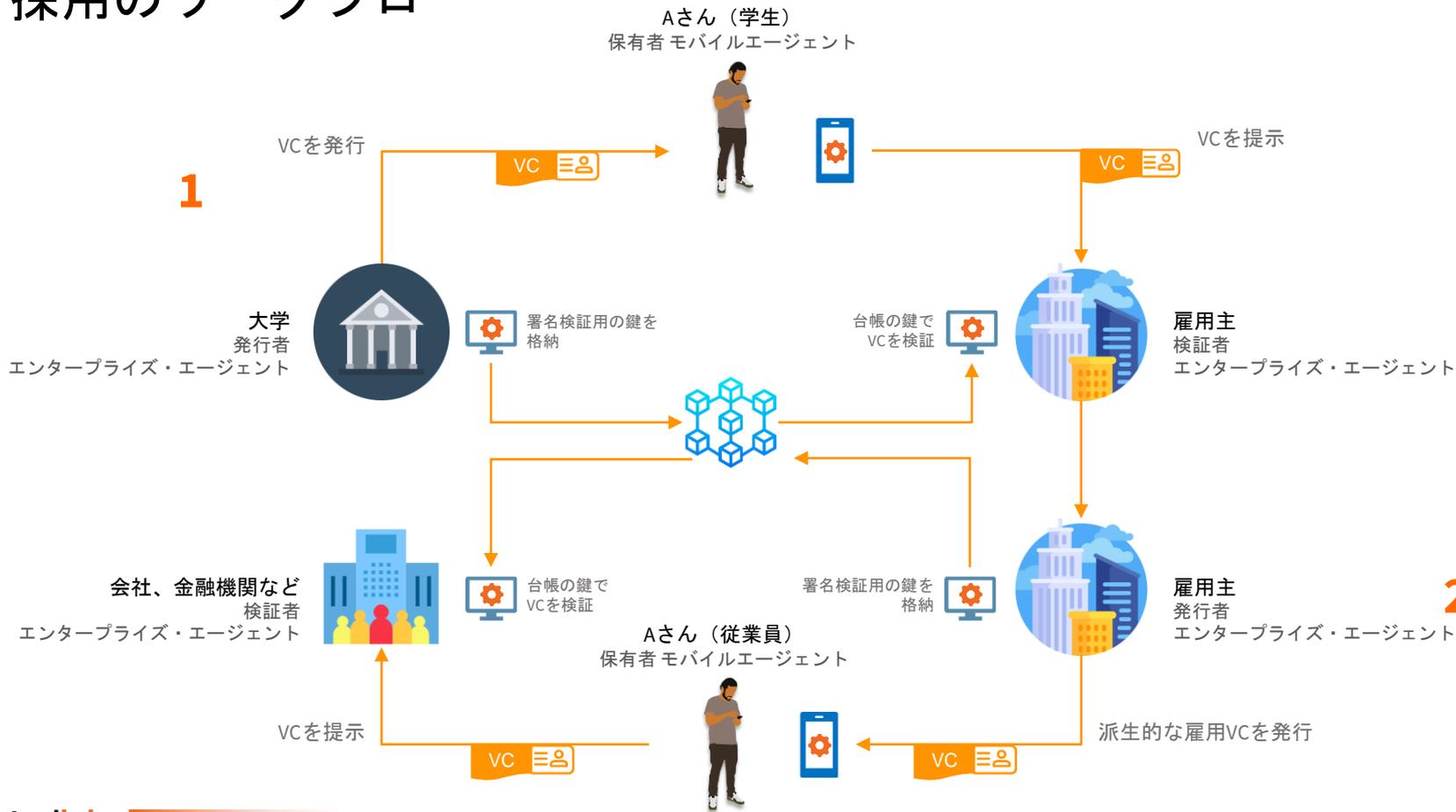
# フローの設計の視点

- 誰がデータを共有する必要があるのか？データの種類は？
- 誰がデータを保有するのか？
- 誰がデータを検証するのか？
- 検証してから発行するようなセットや仲介者はいるのか？
- エージェントとフローを表すために、矢印で関係図を描くことができる

# 採用のワークフロー



# 採用のワークフロー





デモ

# まとめ

- トラスト・トライアングルにおけるエンタープライズ・エージェント
- VCの発行と検証
- エージェントの詳細
- Proven API
- フローの設計
- デモ



# 質疑



ご清聴ありがとうございました