

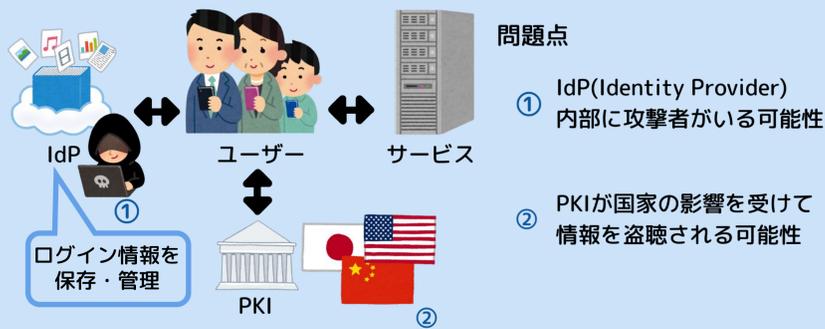
Cassis:カジュアルにDIDを扱うためのツール

開発駆動コース 仲山ゼミ 黒澤 佑太

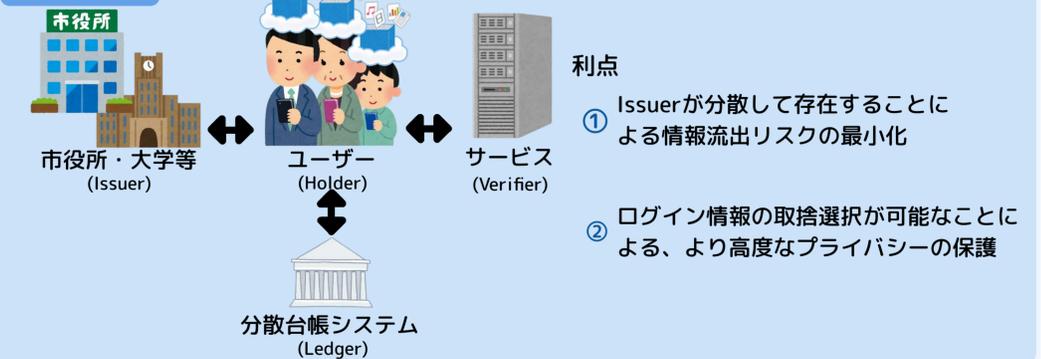
DIDとは？

DID(Decentralized Identifiers)=ユーザー主権型の新しいIDの形

今までのシングルサインオンの構成



DIDでの構成



DIDが普及した世界: 個人情報をも自分で、自信をもって「安全」と言えるようになる → さらに詳細な情報(自分の好み、位置情報の履歴等)もログインに利用できる!

DIDを使う上で面倒なこと

環境的面倒さ

- Issuer・Verifierのエージェントの用意
- 適切なアプリの用意
- 適切なLedgerの用意

手続き的面倒さ

- プロトコル(DIDComm)の理解
- Ledgerネットワークへのアクセス権の獲得
- エージェント間のやり取り

▶ 手軽に使えない!

DIDと仲良くなる

ユーザー視点で**カジュアル**に使いたい!

- ・ 煩雑な手続きの簡略化
- ・ ユーザーからの入力やパスワード等の覚えることを少なく
- ・ 最小限の時間で試すことができる
- ・ 「ぱっとダウンロードして、ぱっと使える」

Cassis = CAsual Self Sovereign Identity System

CassisはDIDをカジュアルに使うためのCLIツール: 必要なものをワンセットに

cassis all
Select:
build Agent

cassisコマンドの入力

```
digraph graph_name {
  //format
  //for Issuer
  //xlabel = "Label:YOUR_ISSUER_LABEL,seed:"
  Issuer1 [label = "Issuer", xlabel = "Label:issuer1,wallet-key:my_password"]
  Issuer2 [label = "Issuer", xlabel = "Label:issuer2,wallet-key:my_password"]
  Holder1 [label = "Holder"]
  Holder2 [label = "Holder"]
  Verifier [label = "Verifier"]
  Issuer1 --> Holder1
  Issuer1 --> Holder2
  Issuer2 --> Holder1
  Issuer2 --> Holder2
  Holder1 --> Verifier
  Holder2 --> Verifier
}
```

label属性で Agentの種類を指定
xlabel属性でその他の属性を指定
label: Agent名
wallet-key: walletのパスワード

DOT言語ファイル

Issuer・Verifierのエージェント
アプリ
クローズドなLedger
手続きの簡略化

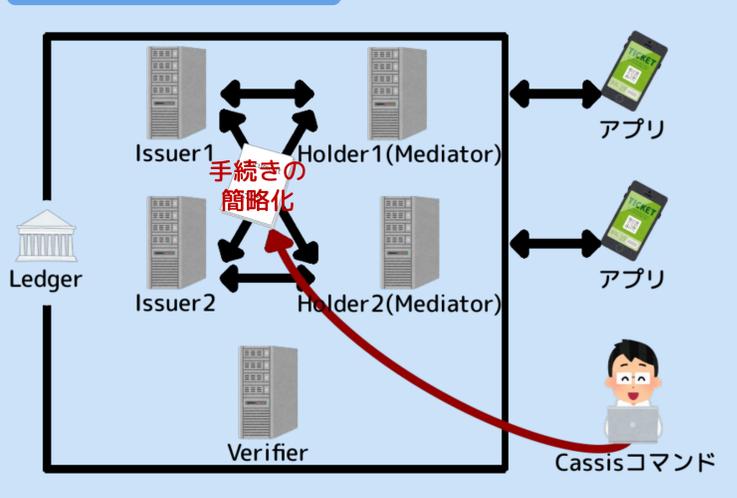
セットで提供

```
label:issuer1,wallet-key:my_password
```

```
label:issuer2,wallet-key:my_password
```

Graphvizによる可視化

構成されたDIDネットワーク



特徴

- DOT言語による可視化と構成ファイルの自動生成
 - グラフィカルに構成の変更をすることが容易
 - DIDに必要な処理を簡略化することが可能
- 対話型UIによる入力と自動検知
 - ユーザーの管理コストを減らし、ホスト環境への適応力を高める
 - 最短で、cassisコマンドを実行して2~3分待つだけでDIDを使った認証の準備が整う

Cassisで出来ること・出来ないこと

出来ること

- コマンドを数回打つだけで、DIDを使った認証が使用可能になる
- 事前の信頼関係がある個人や小規模なコミュニティ内での認証

出来ないこと

- 情報の厳密な精査が必要な、大規模コミュニティや行政から発行される情報での認証
- オープンなLedgerを使用した認証

実際には？

cassis all : エージェント全体への操作

```
~/tmp$ cassis all
Use the arrow keys to navigate: ↓ ↑ ← →
? Select:
  build Agent
  remove Agent

Successfully built 65000254dc78
Successfully tagged tmp_holder2:latest
WARNING: Image for service holder2 was built because it does not exist. To resolve this warning, create and build a container for holder2.
Creating tmp_issuer2_1 ... done
Creating tmp_holder2_1 ... done
Creating tmp_holder1_1 ... done
done!
```

構成ファイル生成ダウンロード
2~3分

build Agent コマンドを実行 構成完了!

cassis issuer : Issuerに対する操作

```
~/tmp$ cassis issuer
Dot file exists
  Issuer1
  schema
  originate schema...
  configure schema
  set attribute
  schema name: test-schema
  Type "end" to exit.
  attribute: number
  attribute: name
  attribute: end
  Do you want to originate now?: Y
schema: test-schema --> ledger originated!
```

QRコードの生成 schemaの登録

cassis holder : holderに対する操作

```
~/tmp$ cassis holder
tuple@tuple-Desktop-Ubuntu:~/tmp$ cassis holder
? Holder1
Use the arrow keys to navigate: ↓ ↑ ← →
? Select:
  build Agent
  accept invitation
  send proposal
  send request
  store credential
```

招待の受け取り ログイン情報の受け取り

より少ないコマンド + 約5分でDIDが使えるようになった!