KUMOPRO-BLOCKCAHIN

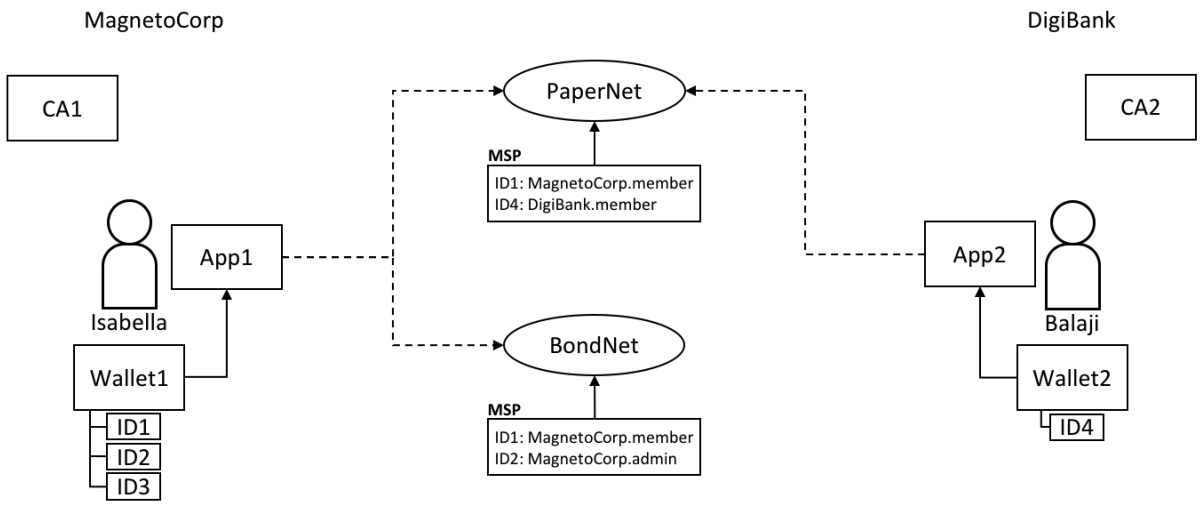
ウォレットの概要

2021 年 5 月 30 日

Wallet

ウォレットは、ユーザのアイデンティティを集めたものです。アプリケーションは、その中のアイデンティティを一つ選んでチャネルに接続します。チャネルへのアクセス権は、子のアイデンティティをMSPの組み合わせで判断されます。

なぜウォレットが重要なのか



アプリケーションは、ネットワークチャネルに接続するとき、それを行うためのユーザアイデンティティ(ID1等)を選択します。チャネルMSPは、ID1と組織の中の役割を関連付けます。この役割によって、最終的にアプリケーションのチャネル資源に対する権限が決まります。

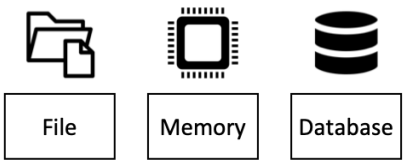
例えば、ID1は台帳に対する読み書きのできるメンバーとしてユーザを識別したり、ID2はコンソーシアムにあたらあしい組織を追加できる管理者として識別したりします。

上図から以下のことがわかります。

- ウォレットは、一人のユーザーに対して複数のアイデンティティを持つことができ、各アイデンティティは別々のCAによって発行されることもある。
- MSPはアイデンティティを発行しているCAに基づいて、IsabellaがMagnetoCorp組織のメンバーであり、BalajiはDigiBank組織のメンバーとして判断します。（一つの組織が複数のCAを用いることも、一つのCAが複数の組織に対応することも可能）
- ID1をPaperNetとBondNetに接続するときに使うことができる。どちらのネットワークでもID1はMagnetoCorpのメンバーとして認識される。
- ID4を用いてBondNetに接続することはできない。

ウォレットの種類

どこにアイデンティティを格納するかによって、ウォレットにはいくつかの種類があります。



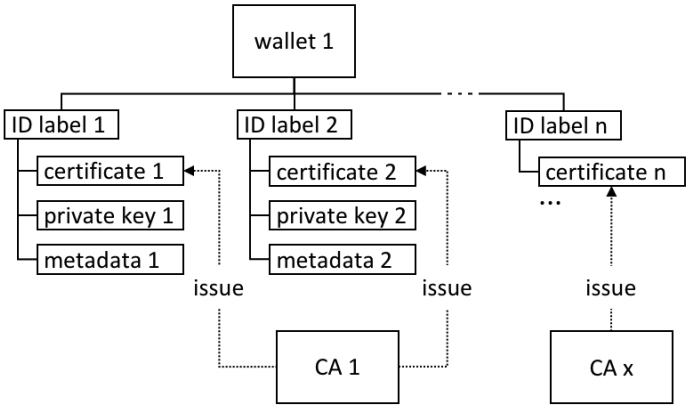
- ファイルシステム
    - もっとも一般的なウォレットの格納先である。
    - ネットワーク経由でマウントすることもでき、ウォレットのデフォルトとして良い選択肢です。
- インメモリ
    - アプリケーションないに格納されるウォレットです。

    - ブラウザのような制限された環境でアプリケーションが動いている場合に、この種類のウォレットを使用する。
    - 揮発性であり、アプリケーションが終了したりクラッシュするとアイデンティティは失われる。
- CouchDB
    - 使われることが最もまれな形のウォレットの格納先
    - データベースのバックアップと復元機構を使いたい場合に便利な選択肢となる。

ウォレットを作成するには、Walletsクラスが提供するファクトリ関数を使用してください。

ウォレットの構成

一つのウォレットは、複数のアイデンティティを持つことができ、それぞれのアイデンティティは認証局（CA）によって発行されています。各アイデンティティは、ラベル、公開鍵を含むX.509証明書、秘密鍵、Fabric特有のメタデータからなる標準的な構造を持っています。



ウォレットとアイデンティティの簡単なクラスメソッドがいくつかあります。

```
const identity: X509Identity = {

    credentials: {

        certificate: certificatePEM,

        privateKey: privateKeyPEM,

    },

    mspId: 'Org1MSP',

    type: 'X.509',

};
await wallet.put(identityLabel, identity);
```

メタデータOrg1MSP、証明書certificateと秘密鍵privateKeyを持つアイデンティティが作られているのがわかります。また、wallet.put()がこのアイデンティティを、identityLabelというラベルでウォレットに追加しているのがわかります。

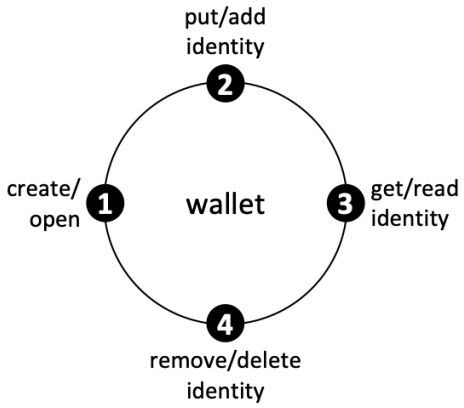
Gatewayクラスは、アイデンティティについてmspIdとtypeメタデータ(上記の例では、それぞれOrg1MSPとX.509)が設定されていることだけを要求します。Gatewayは、例えば、特定の通知に対する処理が要求された時などに、MSPIDの値をコネクションプロファイルのピアを識別するのに使用します。

以下の例だと、networkConnection.yamlから、Org1MSPの通知がpeer0.org1.example.comに関連付けられることがわかります。

```
organizations:

  Org1:

    mspid: Org1MSP

    peers:
      - peer0.org1.example.com
```

ウォレットの操作

全ての種類のウォレットは、共通のWalletインターフェースを実装しており、アイデンティティ管理のための標準的なAPIを提供しています。アプリケーションは、実際のウォレットの格納機構とは独立に作ることができます。



ウォレットは、ライフサイクルに従い、新たに作成したり既存のウォレットを開いた位することができます。また、アイデンティティを読み出すことや、追加、削除も可能です。

ウォレットは、既存の物を開いたり新たに作成した後、アイデンティティを追加・更新・読み取り・削除することができます。