JA

OAuth / OpenID Connect FAPI & IDA 超入門

小岩井 航介

OpenID Foundation

EN

OAuth / OpenID Connect FAPI & IDA for Dummies

Kosuke Koiwai
OpenID Foundation

JA

今日はこれだけ覚えて帰ってください。

- **FAPI**: Financial-grade API Security Profile
  - **OAuth2.0**のAPIを、より安全に利用するための仕様
  - 金融(Financial)**だけじゃない**

- **IDA**: OpenID Connect for Identity Assurance
  - **OpenID Connect**の仕組みを使い、本人確認済みのID情報を連携するための仕様
  - 本人確認プロセスに関するメタデータを定義

EN

TL;DR;

- **FAPI**: Financial-grade API Security Profile
  - A spec to use **OAuth2.0** APIs much safer
  - **NOT limited** to “Financial” use-cases

- **IDA**: OpenID Connect for Identity Assurance
  - A spec to transfer verified ID information using **OpenID Connect**
  - Defines meta-data about ID proofing processes

JA

FAPI: Financial-grade API Security Profile

OAuth2.0のAPIを、より安全に利用するための仕様

EN

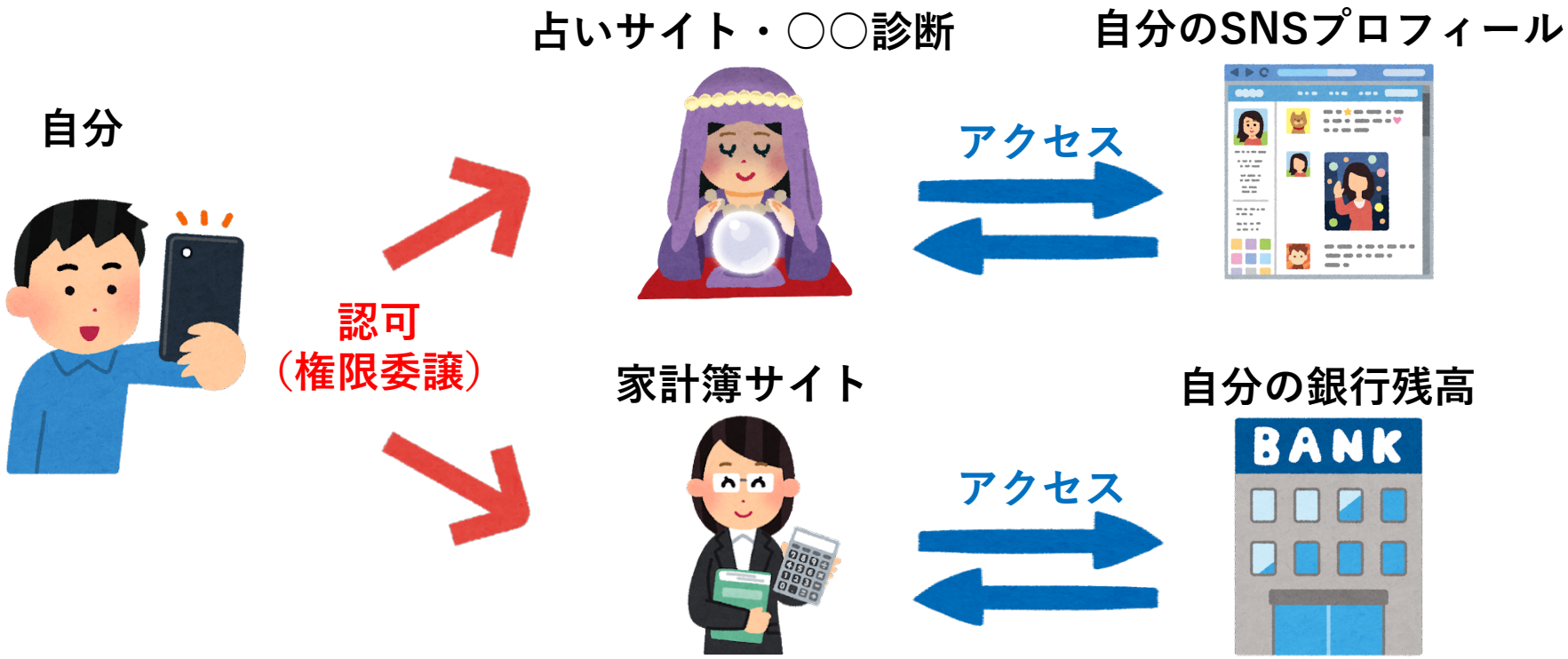
FAPI: Financial-grade API Security Profile

A spec to use **OAuth2.0** APIs much safer

JA

OAuth2.0 とは

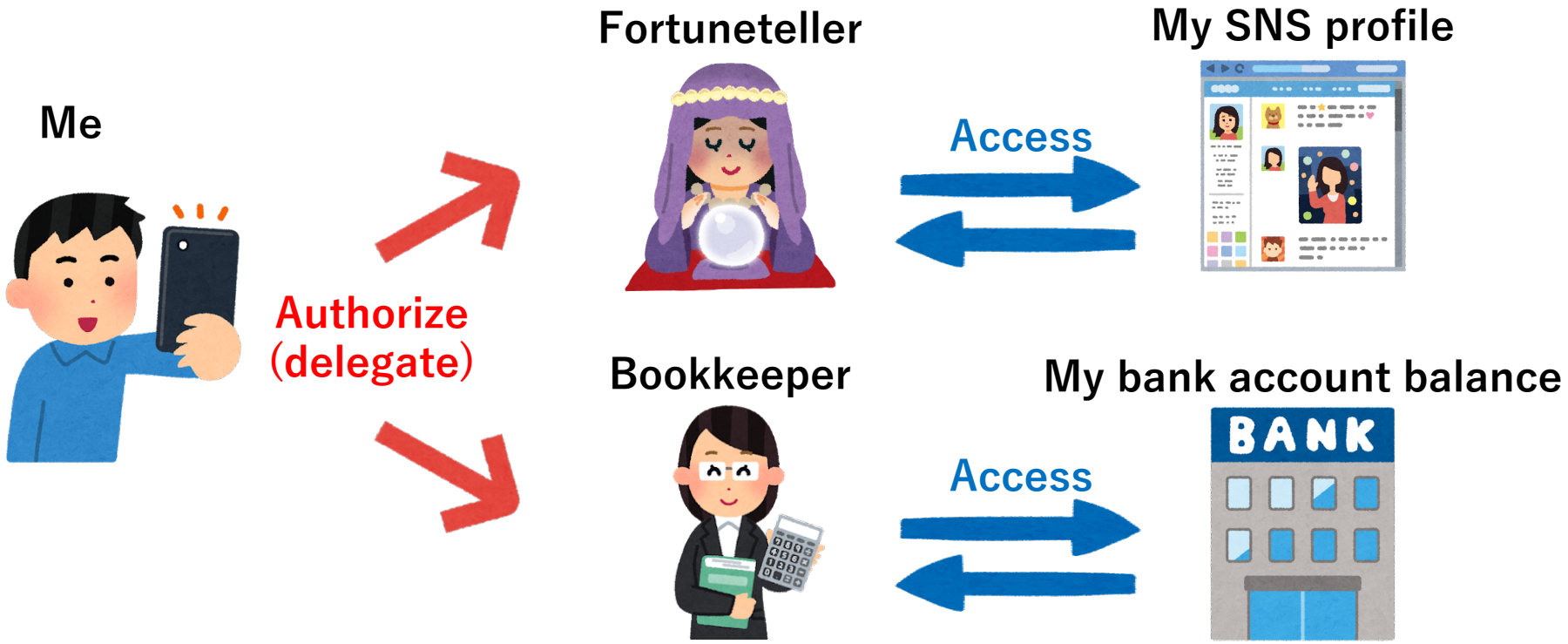
- API認可のためのフレームワーク
  - RFC6749 - The OAuth 2.0 Authorization Framework



EN

OAuth2.0 is

- API Authorization Framework
  - RFC6749 - The OAuth 2.0 Authorization Framework



JA

OAuth2.0 の課題

- 「フレームワーク」なので、様々な使い方が定義されている
- 銀行送金のように、重要な情報などをやりとりする場合、OAuth2.0 とだけ取り決めても、本当に安全かわからない

- **そこで、FAPI が爆誕**

EN

Issue in OAuth2.0

- “Framework” means there are many ways to use it
- Just requiring to use OAuth2.0 doesn’t mean it is safe for high-risk transactions such as Bank transfer

- **That is why FAPI is born**

JA

FAPI とは

- FAPI 1.0 が**2021年3月**に**Final**化。
- 安全なAPI認可、利用を実現するための、**OAuth2.0** のセキュリティプロファイル（設定値の一覧）
- 当初は金融API向けを想定していたが、途中で金融以外にも広く使われるプロファイルを目指すことに
  - Financial**-grade** API ：金融**グレード**API と命名を修正

EN

FAPI is

- FAPI 1.0 **Finalized** in **March 2021**
- **OAuth2.0** security profile (a set of configuration parameters) for a use and authorization of APIs
- Initially proposed for financial use cases, but the scope broadened
  - Name changed to Financial**-grade** API
  - Initially the spec included data models for Financial use cases

JA

FAPI 1.0の構成

- Part1 : Baseline
  - 当初は Read-only という名前だった
  - 情報の漏洩を防ぐのが主な目的

- Part2 : Advanced
  - 当初はRead&Write という名前だった。
  - Baselineに加え、情報の改ざんも防ぐ（メッセージ署名）のが主な目的

- Read-onlyでも重要なデータもあるし、Read&Writeでもそこまで重要じゃない場合もあるということで、Baseline/Advancedに改名

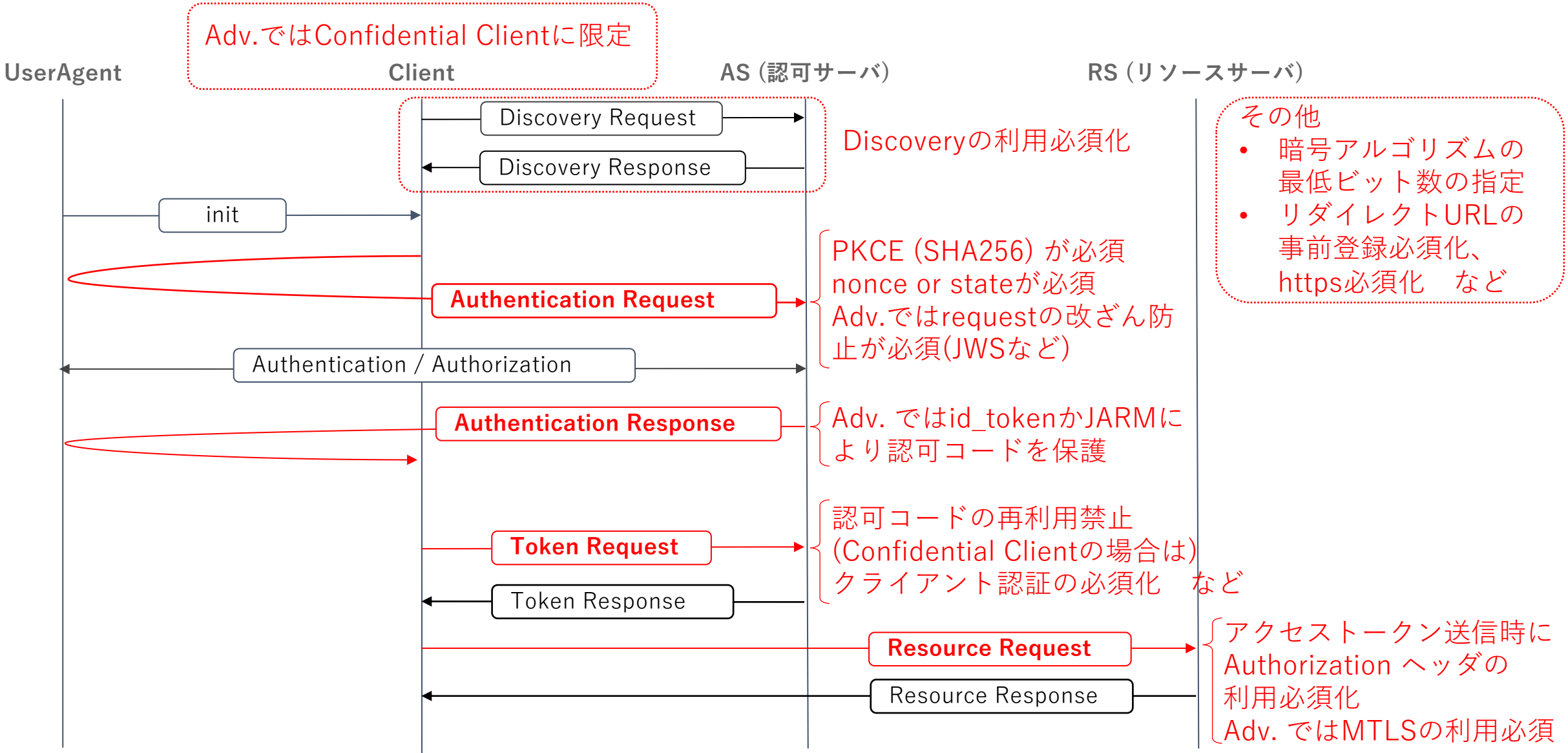
EN

FAPI 1.0 consists of

- Part1 : Baseline
  - Initially named “Read-only”
  - Main purpose is to prevent data leakage
- Part2 : Advanced
  - Initially named “Read&Write”
  - Main purpose is to prevent data tampering (message signing)
- The names were changed to Baseline/Advanced because R/O data can be more important than R/W, or vice versa.

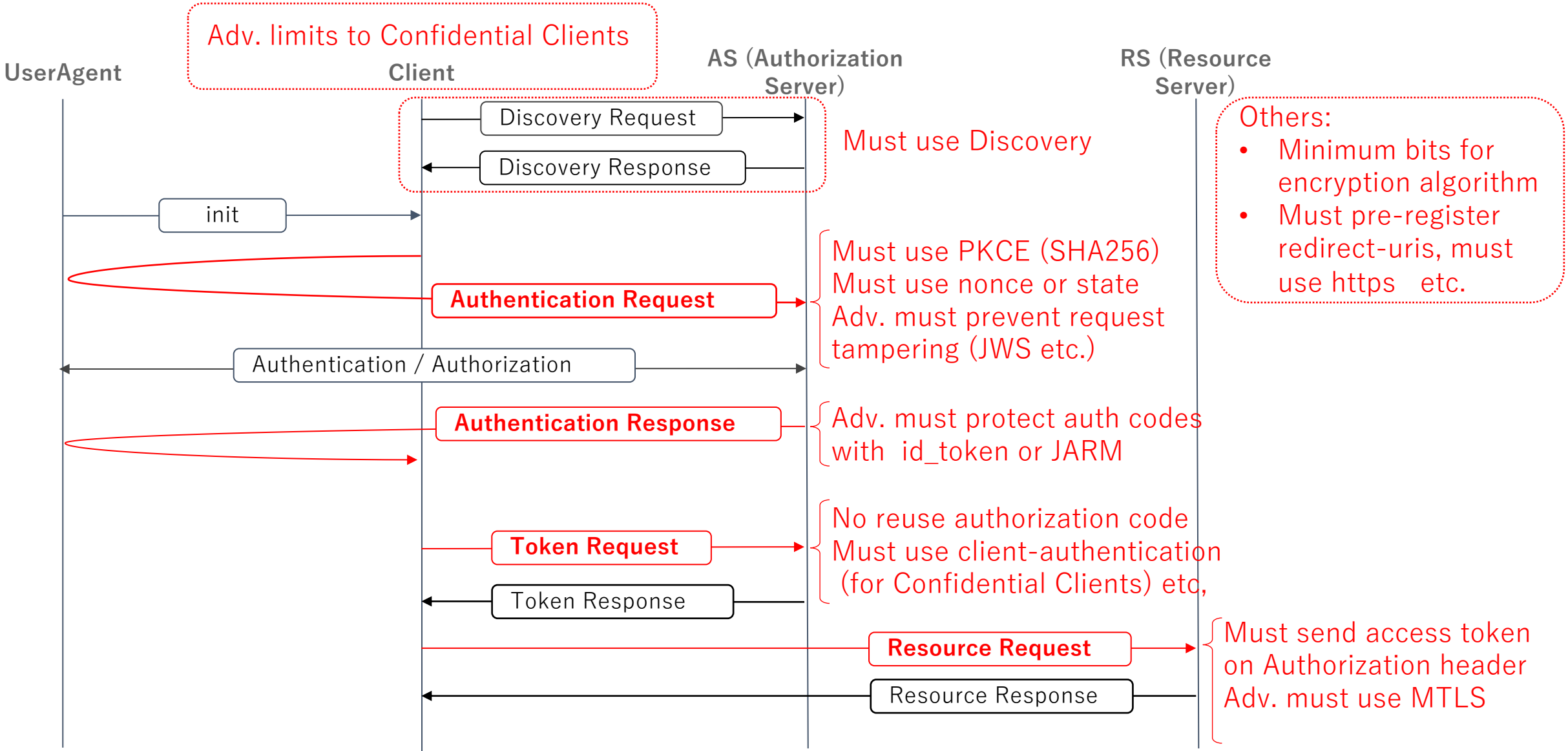
JA

OAuth2.0シーケンス概略とFAPI規定部分(の例)



EN

OAuth2.0 Flow and FAPI spec (a part of)



JA

Certificationが肝

- 仕様が万全でも、それが完璧に実装されていないと意味がない。
- OpenID Foundationでは、Certification Testを提供し、各社で実装が仕様通りかどうかをチェックすることができる。
- イギリス、オーストラリア、ブラジルと、Open BankingでFAPIを採用した国では、Certification Testへの合格を銀行に課している（RPに課している場合もある）
  - 国ごとに微妙に仕様が異なるので、OpenID Foundationではそれぞれのを開発している。

EN

Certification is the key

- Complete spec can’t be of no use without complete implementation
- OpenID Foundation provides Certification Tests so that implementers can check if implementation are aligned to the spec
- Countries adopted FAPI for Open Banking (UK, AU, BR) require banks to pass Certification Tests (BR requires RPs, too)
  - Each country has slightly different specs, so OpenID Foundation develops and maintains Certification Tests for each.

JA

IDA: OpenID Connect for Identity Assurance

OpenID Connectの仕組みを使い、本人確認済みの
ID情報を連携するための仕様

EN

IDA: OpenID Connect for Identity Assurance

A spec to transfer verified ID information using **OpenID Connect**

JA

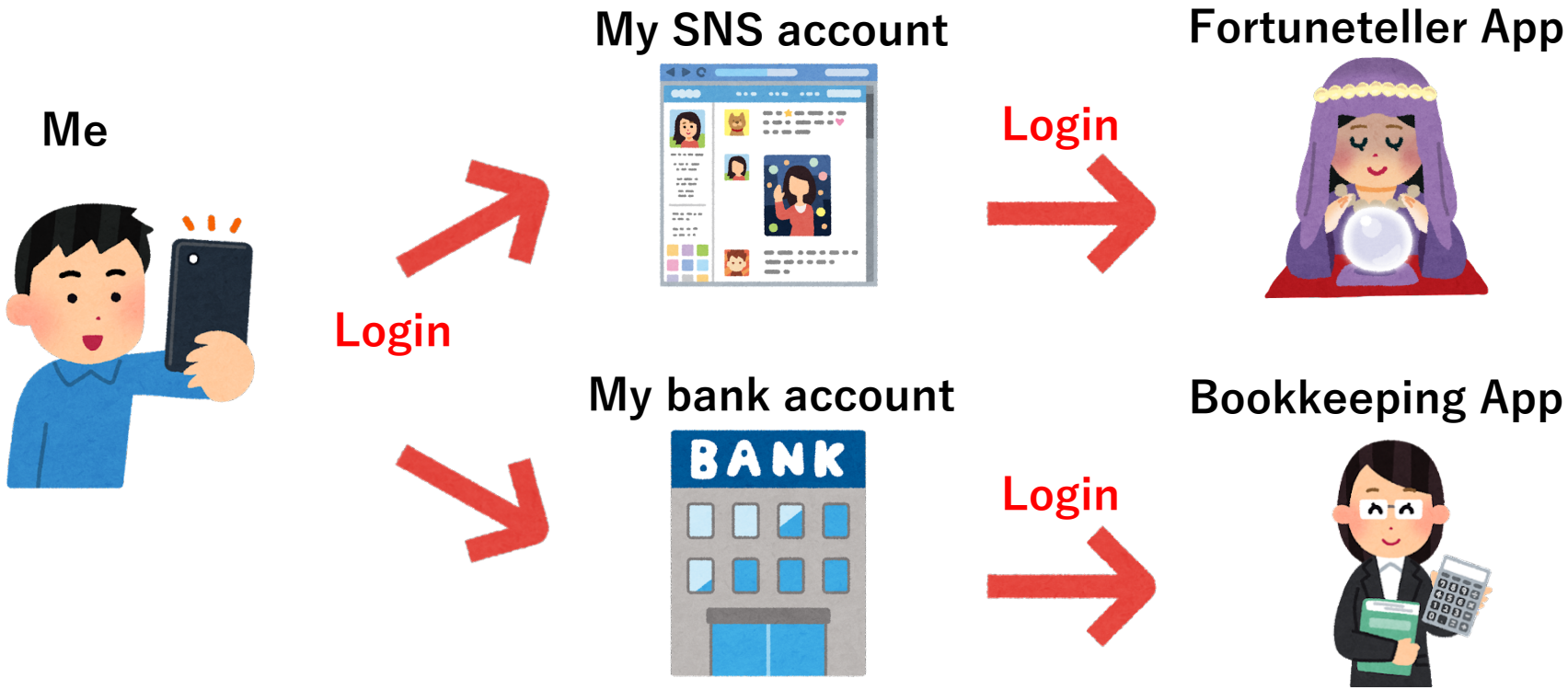
OpenID Connect とは

- 認証連携のためのOAuth2.0の拡張仕様

EN

OpenID Connect is

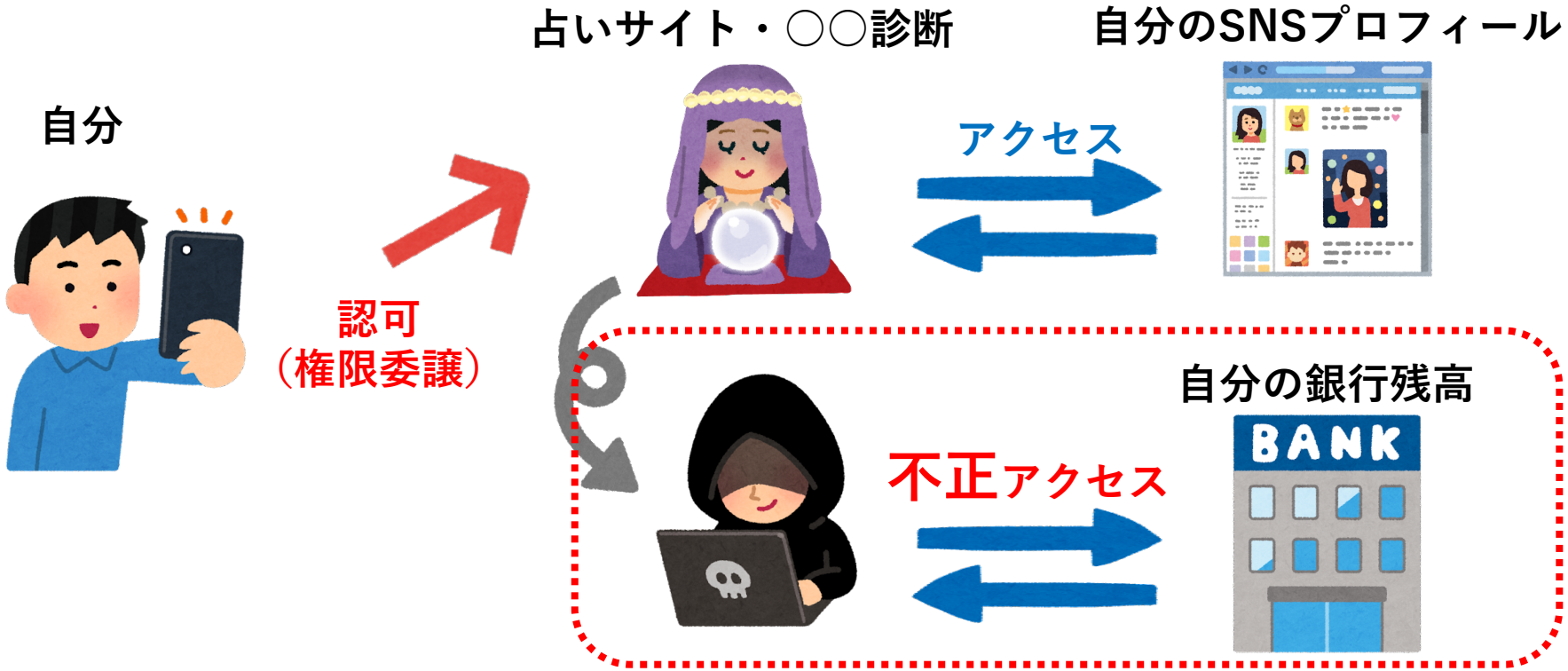
- An extension of OAuth2.0 for federated authentication



JA

OAuth2.0 を認証に使ってはいけない

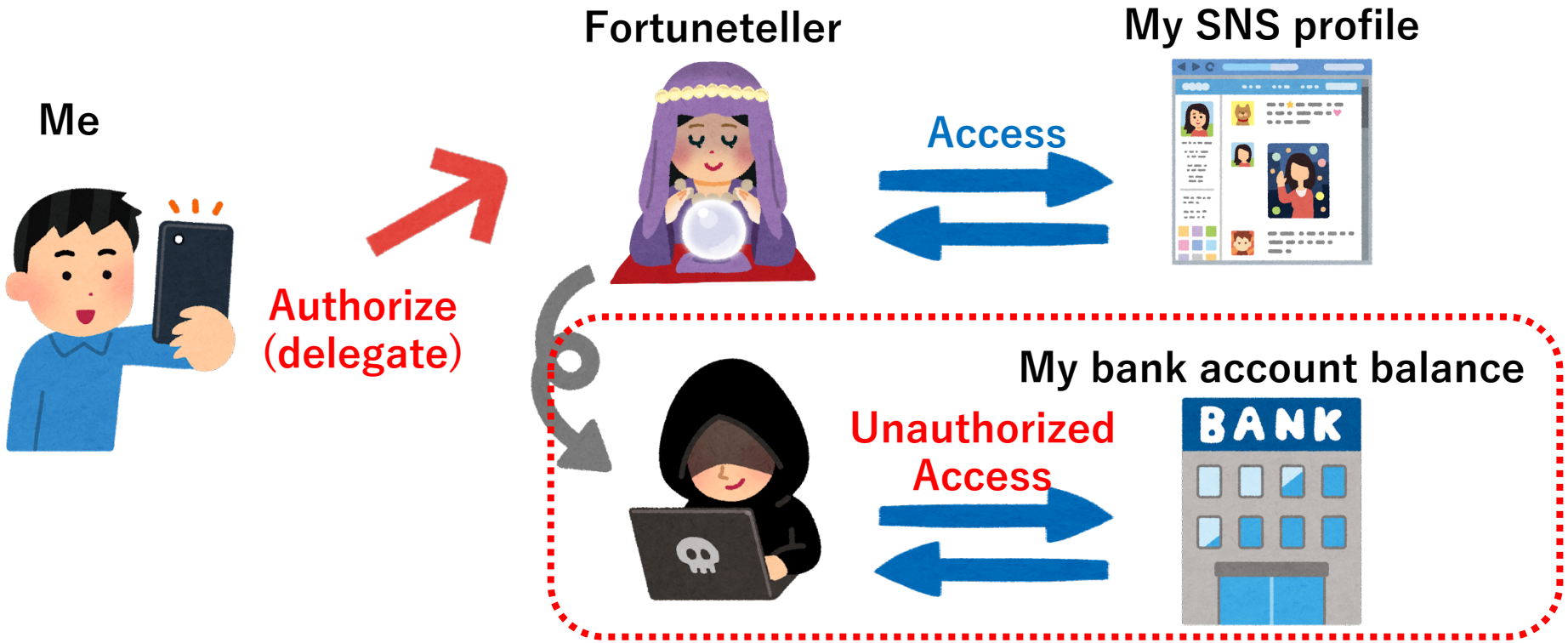
- OAuth2.0で得られるのは特定のAPIを使う権限であり、本人であることの証明ではない



EN

NEVER use OAuth2.0 for Authentication

- OAuth2.0 does NOT give a proof of the person's identity but just a right to use a specific API



JA

OpenID Connect の課題

- 「氏名」「メルアド」「生年月日」「住所」など、様々な個人情報を連携できる一方、その情報の確からしさは不明
- SNSプロフィールで20歳以上と書いてあっても、それを根拠にお酒を売っていいわけではない

- **そこで、IDA が爆誕**

EN

Issue in OpenID Connect

- Can convey various personal info (name, birthdate, email, etc.) but can't tell how certain this info is
- The person's SNS account says s/he is over 21 doesn't mean it is ok to sell Beer to that person

- **That is why IDA is born**

JA

IDA とは

- 本人確認済みのID情報(Identity Assurance) を連携するための、**OpenID Connect** の拡張仕様（Extension）
- 現在 3rd Implementer’s Draft
- OpenID Connectでサポートされている各種個人情報に対して、「誰が」「どのように」「いつ」「何を元に」確認したかを、メタデータとして付与することができる。

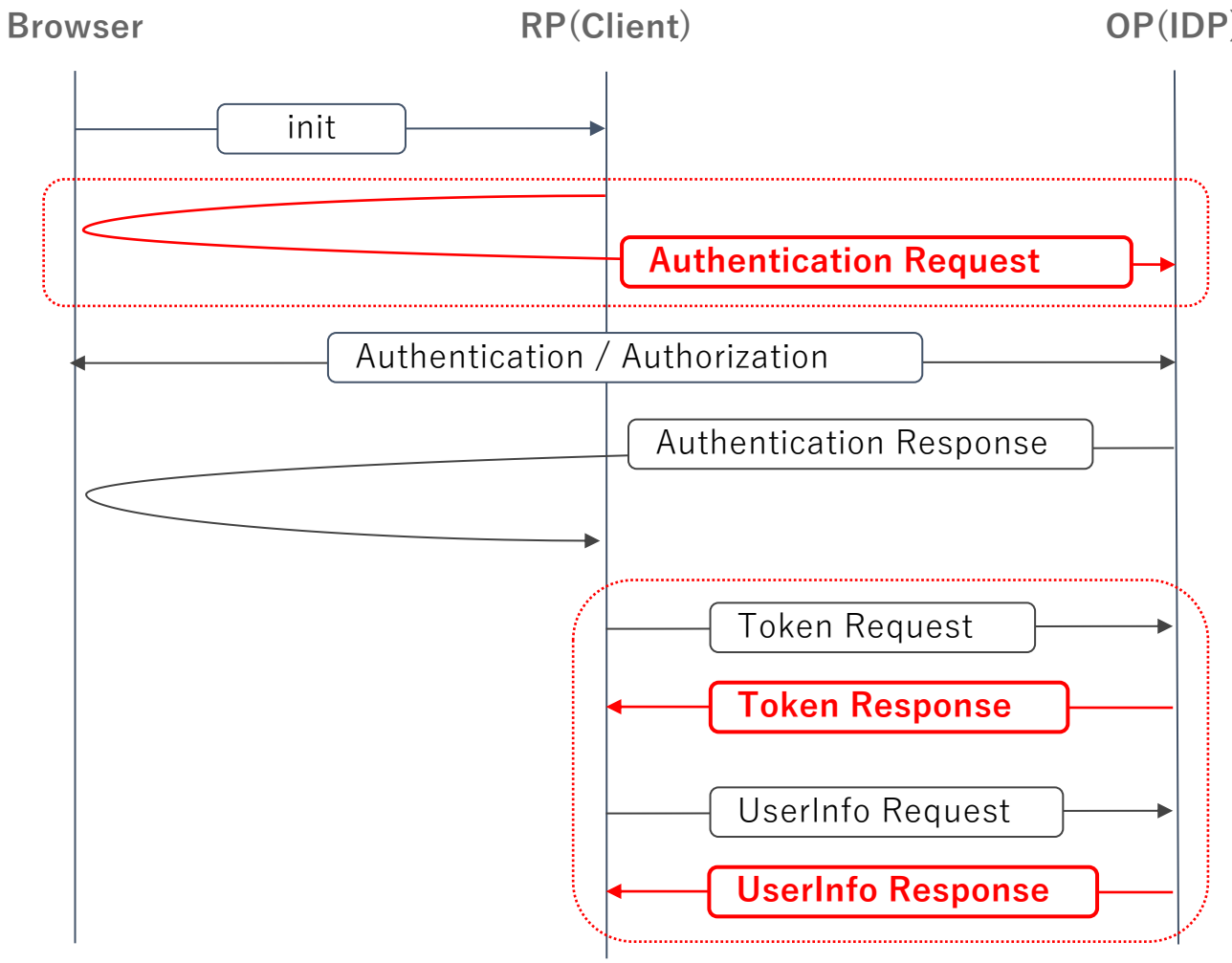
EN

IDA is

- An Extension of **OpenID Connect** to federate proofed/verified ID information
- Currently at 3rd Implementer's Draft status
- Can add Metadata to OpenID Connect-supported claims about "who," "how," "when" and "based on what" the claims are verified.

OIDCシーケンス概略とIDA拡張部分

JA

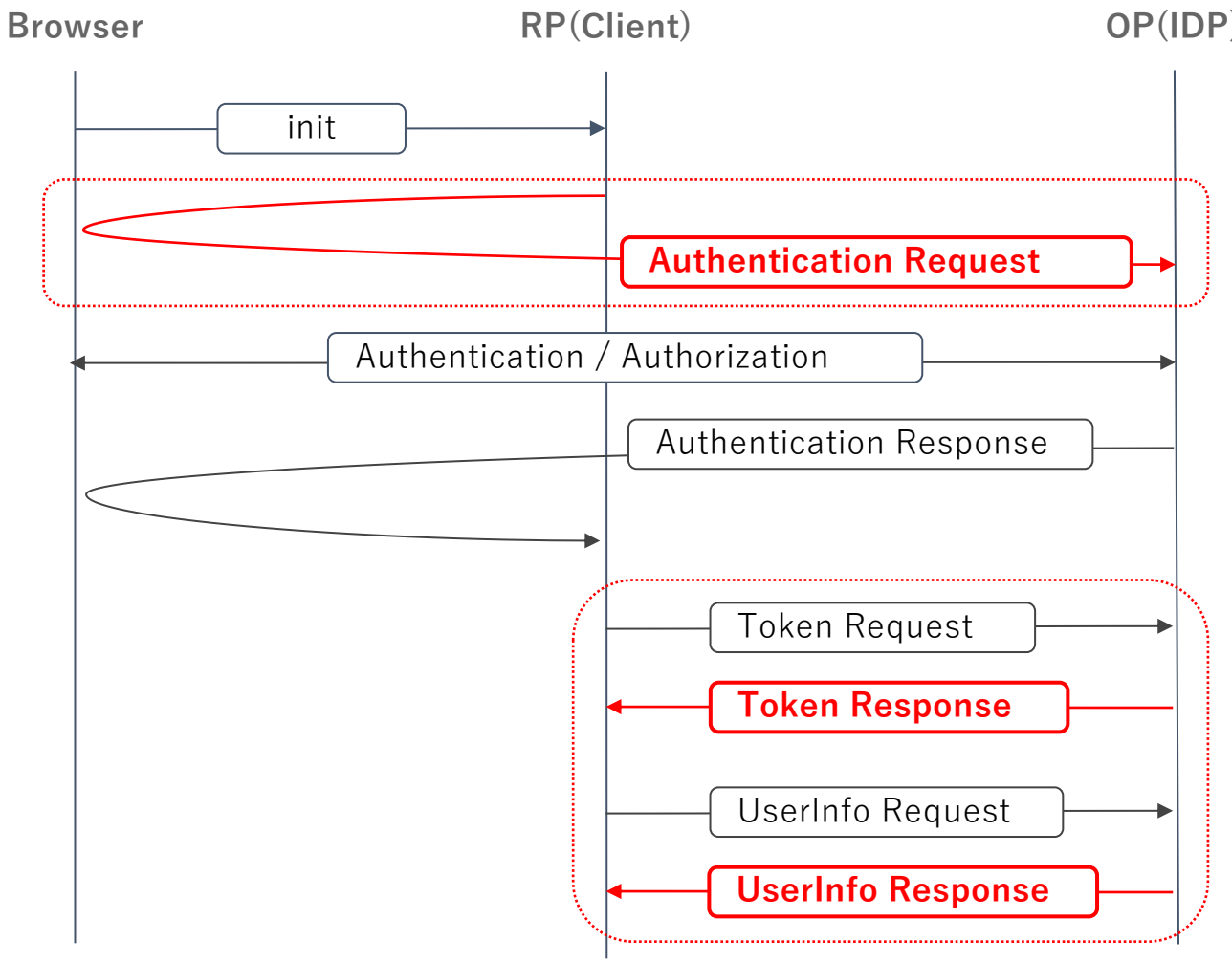


リクエスト時の”claims”
パラメータの拡張

id_token, UserInfo
レスポンスの拡張

OIDC flow and the parts IDA extends

EN



extended ”claims” parameter

extended “id_token”, “UserInfo” responses

JA

リクエストとレスポンスの1例

リクエスト

```
"claims": {
    "userinfo": {
        "verified_claims": [
            {
                "verification": {
                    "trust_framework": {
                        "value": "ja_aml"
                    },
                    "evidence": [
                        {
                            "type": {
                                "value": "document"
                            }
                        }
                    ]
                },
                "claims": {
                    "given_name": null,
                    "family_name": null,
                    "birthdate": null
                }
            }
        ]
    }
}
```

日本の犯収法に基づく確認結果を要求

何らかの文書を証跡として確認すること

必要なのは氏名と生年月日

レスポンス

```
{
    "verified_claims": {
        "verification": {
            "trust_framework": "ja_aml",
            "time": "2021-11-23T00:30Z",
            "evidence": [
                {
                    "type": "document",
                    "method": "pipp",
                    "document_details": {
                        "type": "jp_drivers_license",
                        "date_of_issuance": "2020-11-23",
                        "date_of_expiry": "2025-11-22"
                    }
                }
            ]
        },
        "claims": {
            "given_name": "国際",
            "family_name": "太郎",
            "birthdate": "1986-06-01"
        }
    }
}
```

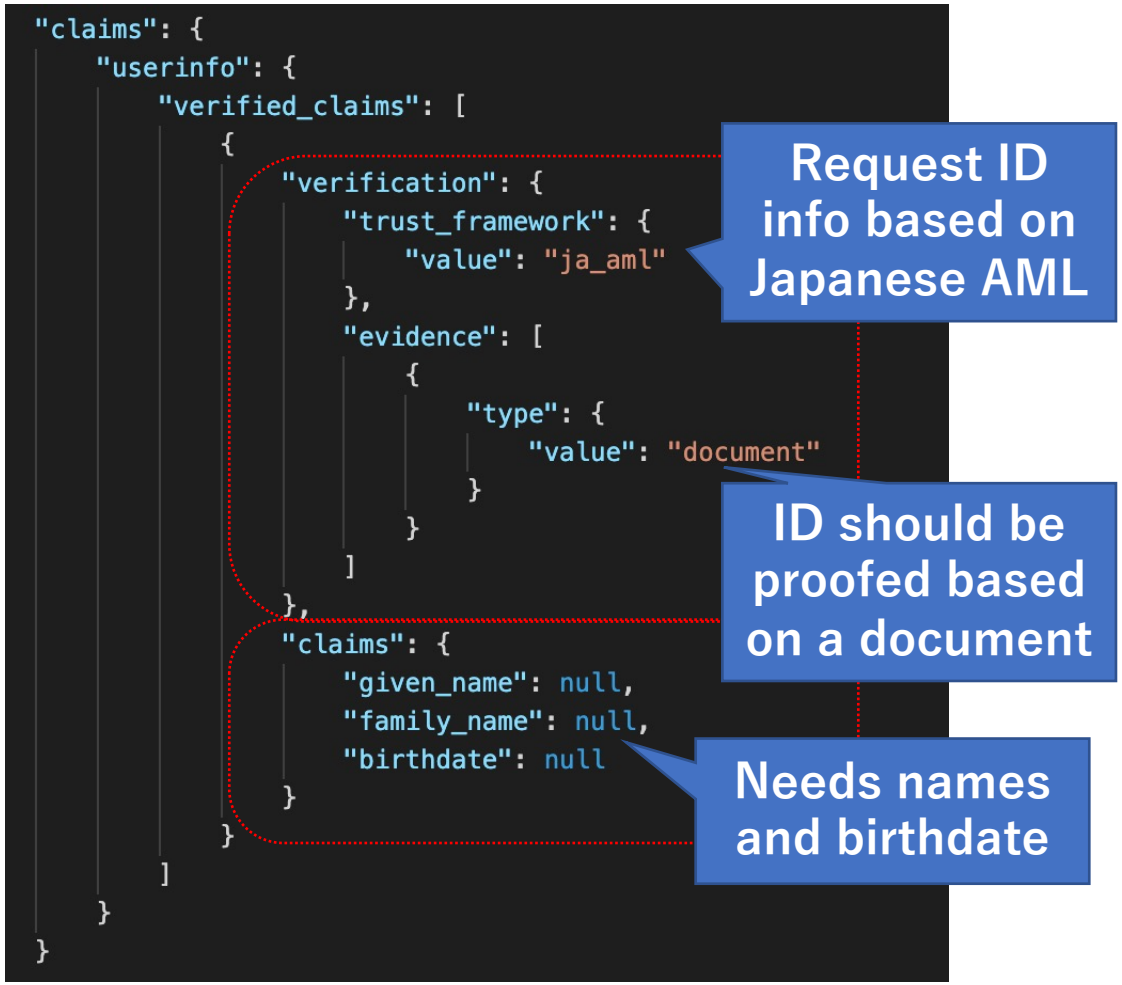
運転免許証を証跡として対面で確認

本人確認済みの個人情報

EN

Sample Request and Response

Request



Response

JA

今日はこれだけ覚えて帰ってください。

- **FAPI**: Financial-grade API Security Profile
  - **OAuth2.0**のAPIを、より安全に利用するための仕様
  - 金融(Financial)**だけじゃない**

- **IDA**: OpenID Connect for Identity Assurance
  - **OpenID Connect**の仕組みを使い、本人確認済みのID情報を連携するための仕様
  - 本人確認プロセスに関するメタデータを定義

EN

TL;DR;

- **FAPI**: Financial-grade API Security Profile
  - A spec to use **OAuth2.0** APIs much safer
  - **NOT limited** to “Financial” use-cases

- **IDA**: OpenID Connect for Identity Assurance
  - A spec to transfer verified ID information using **OpenID Connect**
  - Defines meta-data about ID proofing processes

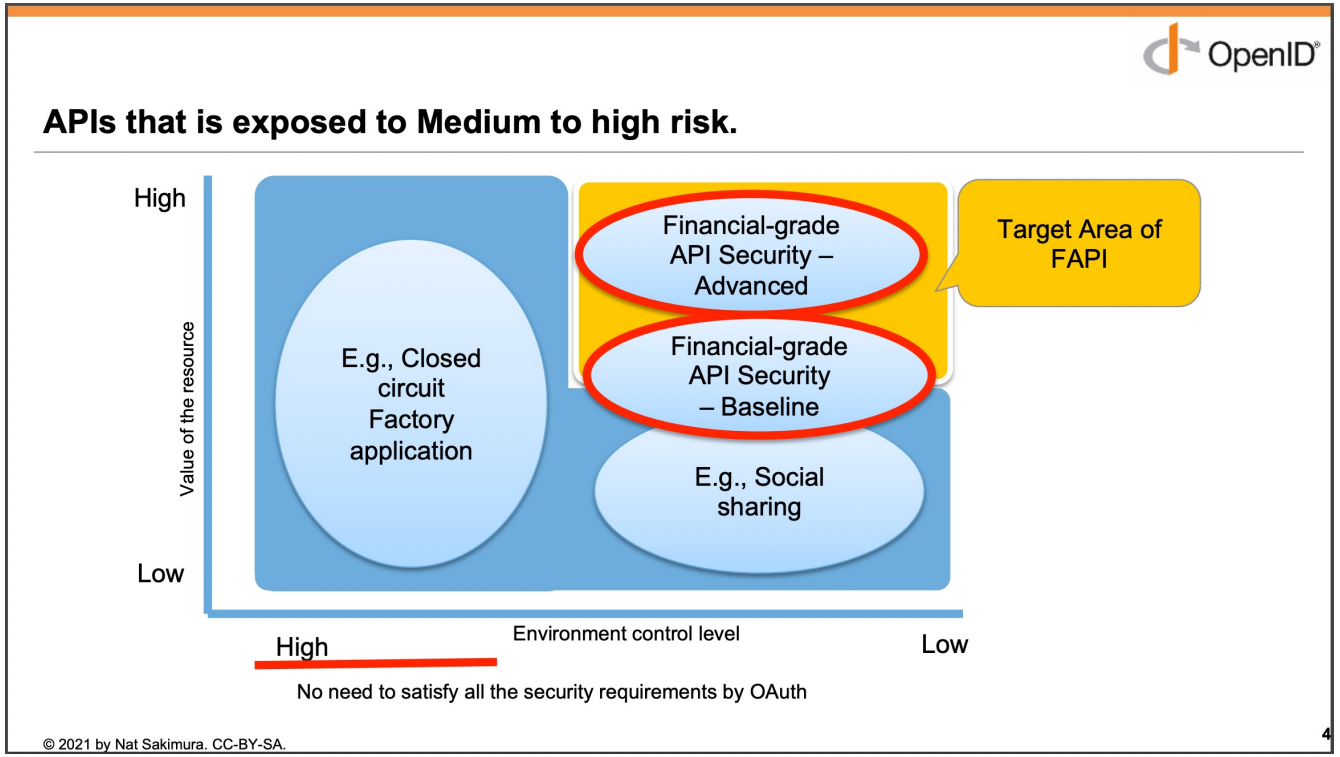
Appendix

またの名を没ネタ / also known as ”save for later”

JA

FAPIの用途

- OAuth2.0は拡張性高くフレキシブルな一方、設定次第で安全にも危険にもなる。
- インターネット上など信頼度が低い環境で重要な情報をやり取りする際に、利用すべきOAuth2.0の設定値を仕様化したもの。

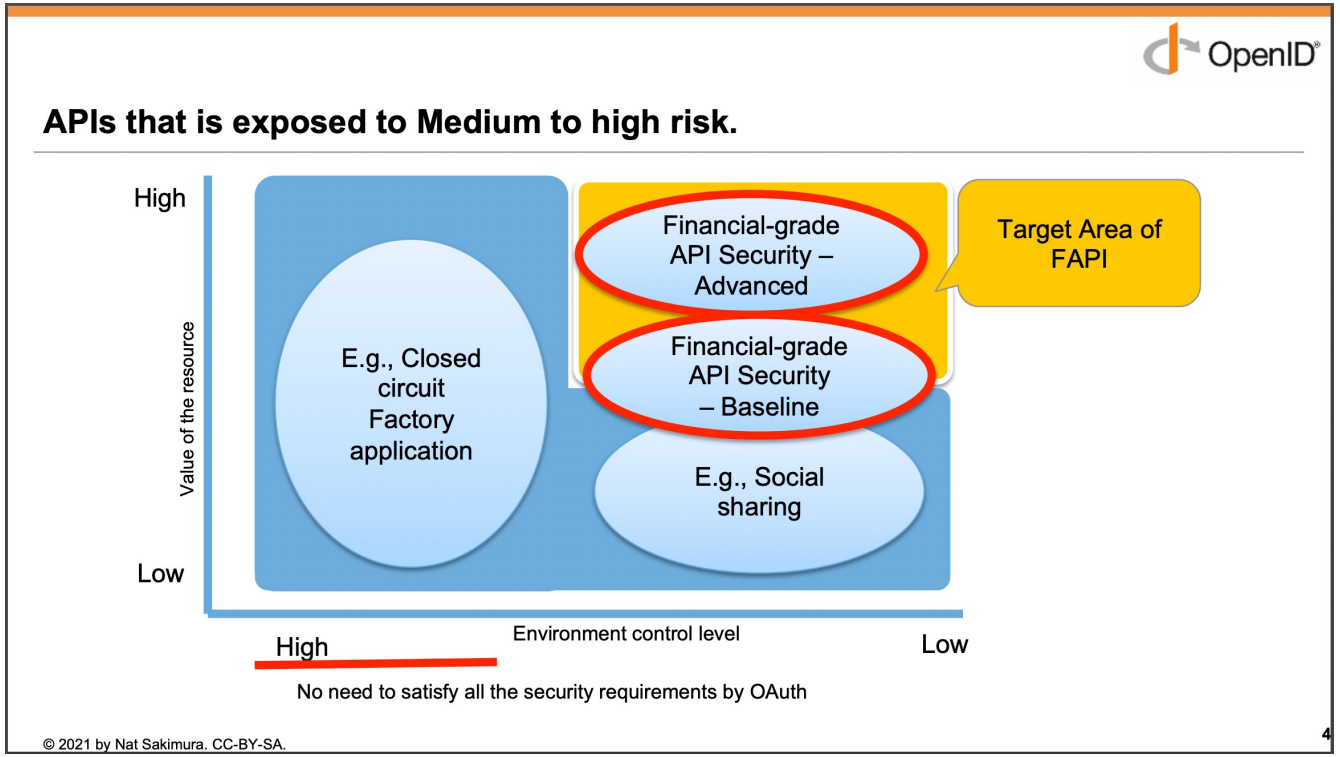


https://openid.net/wordpress-content/uploads/2021/09/OIDF_FAPI-WG-Updates_Nat-Sakimura-Torsten-Lodderstedt.pdf

EN

FAPI is used for

- OAuth2.0 is very flexible, which makes it vulnerable for misconfigurations
- FAPI standardized the set of OAuth2.0 parameters when exchanging valuable information over less-controlled environments such as over Internet.



https://openid.net/wordpress-content/uploads/2021/09/OIDF_FAPI-WG-Updates_Nat-Sakimura-Torsten-Lodderstedt.pdf

JA

FAPI 1.0 の中身抜粋: Baseline

- 暗号アルゴリズムの最低ビット数の指定
  - RSAなら2048bit以上、ECなら160bit以上
- PKCE(S256)が必須, redirect-uriはhttpsが必須
  - ネイティブアプリはAppLinksやUniversalLinksの利用が前提
- Discoveryの利用が必須（別の手段のやりとりだと、フィッシングなどで偽のエンドポイントに案内される恐れがある）
- Confidential Client (サーバ) の場合はprivate_key_jwtなど、鍵を生で送らない形式のクライアント認証が必要
- nonce (openidの場合)か state が必須
- リソースアクセス時のaccess_tokenはAuthorizationヘッダ利用必須

EN

Sneak peak of FAPI 1.0: Baseline

- Minimum bits for encryption algorithms
  - ≧2048bit for RSA, ≧160bit forEC
- Must use PKCE(S256), must use https for redirect-uri
  - Thus, AppLinks or UniversalLinks is required for Native Apps
- Must use Discovery (Otherwise, clients may be directed to a fake endpoints via phising attacks)
- Confidential Clients need to use client-authentication with no shared-secret over the network (e.g. private_key_jwt)
- Must have nonce (openid) or state
- Must put access-token in Authorization header in resource access

JA

FAPI 1.0 の中身抜粋: Advanced

- Baselineに加えて(一部例外はあり)・・・
- Publicクライアント（ブラウザやアプリ）は禁止
- JWS署名したrequestオブジェクトの利用 or request_uriの利用が必須（パラメータの改ざん防止）
- リソースサーバアクセス時にクライアント証明書によるクライアント認証が必須、さらにaccess_tokenは証明書と紐付け、盗まれても他人が利用できないこと。
- id_token かJARMを使って、codeやstateを改ざん防御すること
  - id_tokenの場合、c_hash/s_hashを含める。(s_hashはFAPI1で新設)
  - JARMはcode含むレスポンスをまるごとJWT化する仕様

Sneak peak of FAPI 1.0 : Advanced

EN

- In addition to Baseline (some exceptions) :
- No Public Clients (Browsers or Apps)
- Must use JWS-signed request object or request_uri
  - to prevent tampering request parameters
- Must use client-authentication with client-certificates, and must bind access_token to client so that a stolen token can be used by others
- Prevent code/state from tampering using id_token or JARM
  - If using id_token, include c_hash/s_hash (FAPI1 defines “s_hash”)
  - JARM is a spec to JWT-ize entire authorization response

JA

FAPI 2 への動き

- よりシンプルに

- より相互運用性を
  - 選択肢（optionality）を減らす方向
  - Pushed Authorization Request (PAR: RFC9126) の必須化

- 新たな要件の取り込み
  - Grant Management
  - Rich Authorization Request

EN

Movement towards FAPI 2

- Simpler
- More interoperability
  - Less optionality
  - Require Pushed Authorization Request (PAR: RFC9126)
- New use cases
  - Grant Management
  - Rich Authorization Request

JA

IDA: さらなる拡張

- Selective Abort/Omit
  - 例えば「日本の犯収法に基づかない結果であれば不要」など、結果を受領する条件を規定することができる
  - 手数料を徴求するようなビジネスモデルで有用
- Claims Transformation
  - 「20歳以上」「東京都か千葉県在住かどうか」など、個人情報本体ではなく、その条件をTrue/Falseで返却する
- いずれも、Advanced Syntax for Claimsという新たな仕様として策定中
  - OIDC単体に対する拡張としても利用可能だが、IDAと併用するとさらに有用

EN

IDA: Further Extensions

- Selective Abort/Omit
  - Set conditions to receive results, such as “do not return unless the claims is verified based on Japanese AML law”
  - Useful for business models to receive fees
- Claims Transformation
  - Return conditional results as True/False, not the claims itself, such as “over 20 or not” or “lives in Tokyo or Chiba prefecture”
- Under efforts to create new spec “Advanced Syntax for Claims”
  - Can be used for sole OIDC extension, but much more useful with IDA.