AUTHLETE

OAuth 2.0 と OpenID Connect の細かい話

Authlete ナレッジベースから覗くプロファイリングの深淵

工藤達雄

Authlete, Inc.

お知らせ

- セッションはレコーディングされます
- ご質問、コメントは、Zoom Webinar の QA に投稿ください。可能な限り回答いたします

本日の内容

- OAuth 2.0 / OpenID Connect (OIDC) のプロファイリング（詳細仕様化）にまつわるあれやこれやをお話します
- プロファイリングの具体例として主に Authleteのナレッジベース (https://kb.authlete.com/ja/) の記事を紹介します

アジェンダ

- はじめに
- トークン発行のコンテキスト
- 発行するトークンの特性
- 発行後のトークン操作
- クライアントへのユーザー属性情報提供

AUTHLETE

はじめに

プロファイリング?

- 実際のユースケースへの OAuth / OIDC 仕様の適用
- 仕様が示す選択肢から選択
- 仕様の範囲外にも決めごとがいろいろ

1.2. Protocol Flow

```
     +--------+                                +---------------+
     |        |--(A)- Authorization Request ->|   Resource    |
     |        |                               |     Owner     |
     |        |<-(B)-- Authorization Grant ---|               |
     |        |                               +---------------+
     |        |                               +---------------+
     |        |--(C)-- Authorization Grant -->| Authorization |
     | Client |                               |     Server    |
     |        |<-(D)----- Access Token -------|               |
     |        |                               +---------------+
     |        |                               +---------------+
     |        |--(E)----- Access Token ------>|    Resource   |
     |        |                               |     Server    |
     |        |<-(F)--- Protected Resource ---|               |
     +--------+                               +---------------+
```

Figure 1: Abstract Protocol Flow

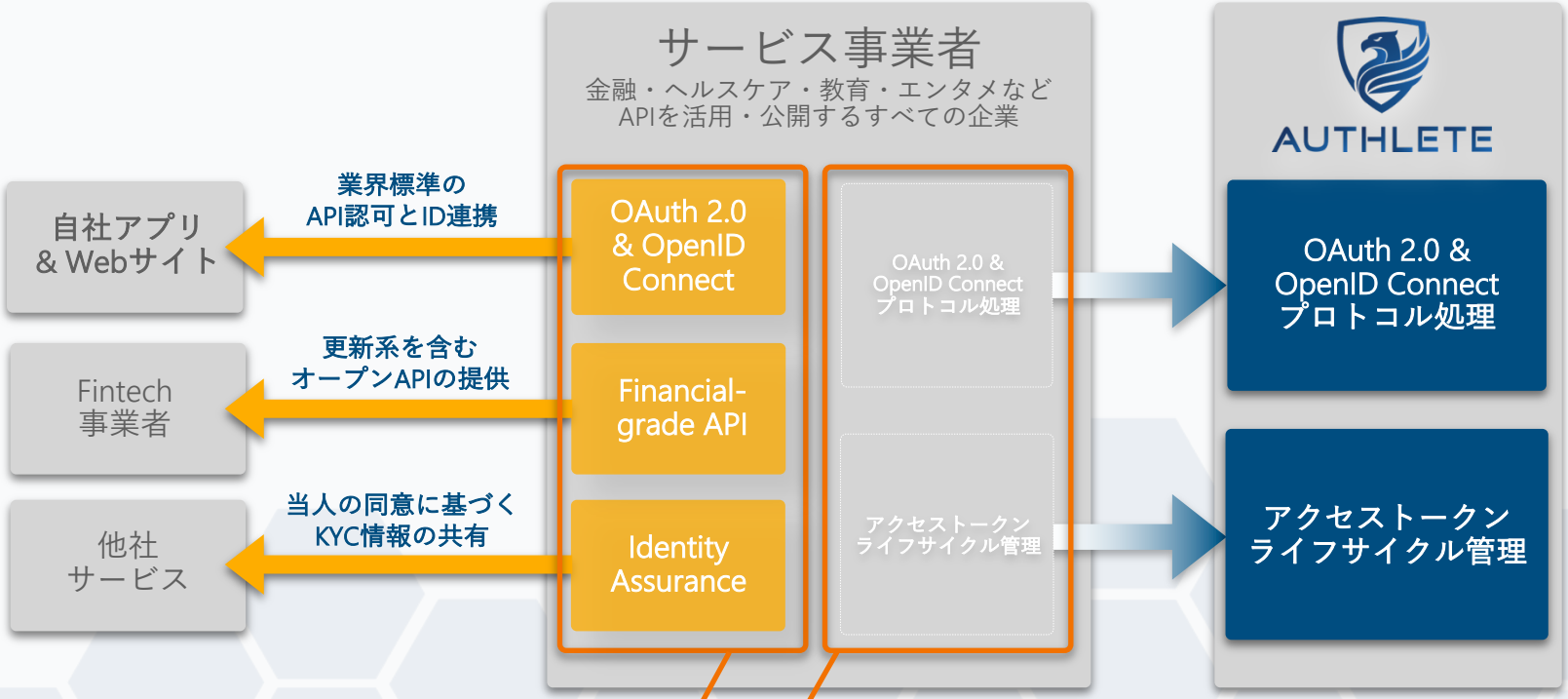
Source: RFC 6749 1.2. Protocol Flow https://tools.ietf.org/html/rfc6749

Q. こんなことを考えた経験はありますか?

(Yes or No)

1. クライアントからのトークンリフレッシュ要求に対して認可サーバーが返却する、リフレッシュトークン (RT) の有効期限はどうする?
2. リソースオーナーの同意の下にクライアントに発行されたアクセストークン (AT1) が有効な状態で、さらにそのクライアントに AT2 を発行する際、すでに発行されていた AT1 はどう取り扱う?

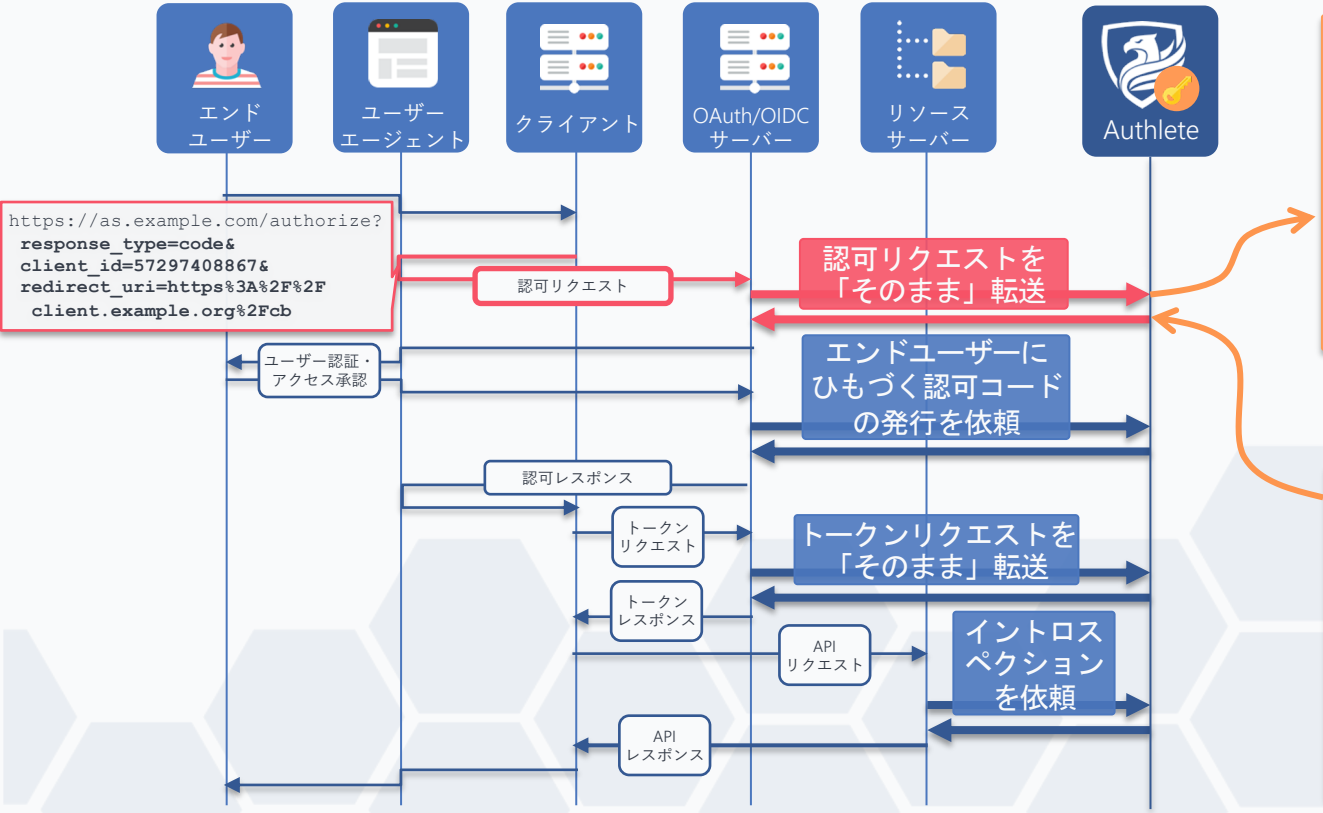
Authlete: OAuth/OIDCを実装するための実装



最先端の業界標準仕様に準拠したセキュアなAPI提供を実現

OAuth 2.0 & OpenID Connect に関する複雑な実装・運用を外部化
特定ソリューションに依存せず自由にUX設計が可能

AuthleteとOAuth/OIDCサーバーの連携フロー



POST /auth/authorization

```
{ "parameters":
"response_type=code&
 client_id=57297408867&
 redirect_uri=https%3A%2F%2F
  client.example.org%2Fcb" }'
```

OAuth/OIDC サーバーが次にすることを Authlete API が指示

```
{
  "action": "INTERACTION",
  "ticket": "c4iy3TWUzV9axH-9Q"
  ...
}
```

さまざまなプロファイリングの実装をサポート

AUTHLETE　ニュース・Blog　導入事例　開発者　サービス体系　各種資料　お問い合せ　無料トライアル

Spec Sheet

仕様

機能	Version	説明
標準仕様	1.1~	• RFC 6749: The OAuth 2.0 Authorization Framework • RFC 6750: The OAuth 2.0 Authorization Framework: Bearer Token Usage • RFC 7009: OAuth 2.0 Token Revocation • RFC 7636: Proof Key for Code Exchange by OAuth Public Clients • RFC 7662: OAuth 2.0 Token Introspection • OAuth 2.0 Multiple Response Type Encoding Practices • OAuth 2.0 Form Post Response Mode • OpenID Connect Core 1.0 • OpenID Connect Discovery 1.0
	2.0~	• RFC 7523: JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants • RFC 8705: OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens • Financial-grade API - Part 1: Read-Only API Security Profile • Financial-grade API - Part 2: Read and Write API Security Profile • UK Open Banking Security Profile
	2.1~	• RFC 7591: OAuth 2.0 Dynamic Client Registration Protocol • RFC 7592: OAuth 2.0 Dynamic Client Registration Management Protocol • RFC 8628: OAuth 2.0 Device Authorization Grant • OpenID Connect Dynamic Client Registration 1.0 • OpenID Connect Client Initiated Backchannel Authentication Flow - Core 1.0 • Financial-grade API: JWT Secured Authorization Response Mode for OAuth 2.0 (JARM) • Financial-grade API: Client Initiated Backchannel Authentication Profile
	2.2~	• RFC 8707: Resource Indicators for OAuth 2.0 • RFC 9101: The OAuth 2.0 Authorization Framework: JWT-Secured Authorization Request (JAR) • OAuth 2.0 Pushed Authorization Requests • OAuth 2.0 Rich Authorization Requests • OAuth 2.0 Demonstration of Proof-of-Possession at the Application Layer (DPoP) • OpenID Connect for Identity Assurance 1.0 • OAuth 2.0 Authorization Server Issuer Identifier in Authorization Response • Australia Consumer Data Right Security Profile • Open Banking Brasil Financial-grade API Security Profile
クライアント認証方式	1.1 ~	• none • client_secret_basic (RFC 6749) • client_secret_post (RFC 6749)
	2.0 ~	• client_secret_jwt (RFC 7523) • private_key_jwt (RFC 7523) • tls_client_auth (MTLS) • self_signed_tls_client_auth (MTLS)

エンドポイント	1.1 ~	• Authorization Endpoint (RFC 6749) • Discovery Endpoint (OIDC Discovery 1.0) • Introspection Endpoint (RFC 7662) • JWK Set Endpoint (RFC 7517) • Revocation Endpoint (RFC 7009) • Token Endpoint (RFC 6749) • Userinfo Endpoint (OIDC Core 1.0)
	2.1 ~	• Backchannel Authentication Endpoint (CIBA Core 1.0) • Device Authorization Endpoint (RFC 8628)
	2.2 ~	• Pushed Authorization Request Endpoint (PAR)
認可種別 (Grant Types)	1.1 ~	• authorization_code (RFC 6749) • implicit (RFC 6749) • password (RFC 6749) • client_credentials (RFC 6749) • refresh (RFC 6749)
	2.1 ~	• urn:openid:params:grant-type:ciba (CIBA) • urn:ietf:params:oauth:grant-type:device_code (RFC 8628)
応答種別 (Response Types)	1.1 ~	• code (RFC 6749) • token (RFC 6749) • id_token (Multiple Response Type) • code token (Multiple Response Type) • code id_token (Multiple Response Type) • id_token token (Multiple Response Type) • code id_token token (Multiple Response Type) • none (Multiple Response Type)
応答形式 (Response Modes)	1.1 ~	• query (Multiple Response Type) • fragment (Multiple Response Type) • form_post (Form Post Response Mode)
	2.1 ~	• jwt (JARM) • query.jwt (JARM) • fragment.jwt (JARM) • form_post.jwt (JARM)
署名アルゴリズム	1.1 ~	• HS256 • HS384 • HS512 • RS256 • RS384 • RS512 • ES256 • ES384 • ES512 • PS256 • PS384 • PS512 • none
暗号アルゴリズム	1.1 ~	• RSA1_5 • RSA-OAEP • RSA-OAEP-256 • A128KW • A192KW • A256KW • dir • ECDH-ES • ECDH-ES+A128KW • ECDH-ES+A192KW • ECDH-ES+A256KW • A128GCMKW • A192GCMKW • A256GCMKW • PBES2-HS256+A128KW • PBES2-HS384+A192KW • PBES2-HS512+A256KW
暗号方式	1.1 ~	• A128CBC-HS256 • A192CBC-HS384 • A256CBC-HS512 • A128GCM • A192GCM • A256GCM
Authlete 固有	1.1 ~	• Client ID 別名 • Extra Properties • リフレッシュトークン再発行ポリシー • サブジェクト毎のユーザアクセストークン • エラー説明の省略 • エラー URI の省略 • スコープ付与状態認証機能 *1 • PKCE 強制
	2.0 ~	• スコープ属性 • Mutual TLS 認証時の PKI 証明書チェーン検証 (MTLS)
	2.1 ~	• コードチャレンジのメソッド S256 強制 (PKCE) • JWT 形式アクセストークン • 許容可能時刻差異 • FAPI コンテキストにおけるバインディングメッセージ強制 (FAPI-CIBA) • リフレッシュトークン再発行ポリシーの拡張
	2.2	• ID トークンのヘッダーへの独自クレーム追加
トークン有効期間設定	1.1 ~	• サービス毎のアクセストークン有効期間設定 • サービス毎のリフレッシュトークン有効期間設定 • サービス毎の ID トークン有効期間設定
	2.0 ~	• スコープ毎のアクセストークン有効期間設定 • スコープ毎のリフレッシュトークン有効期間設定
	2.1 ~	• クライアント毎のアクセストークン有効期間設定 • クライアント毎のリフレッシュトークン有効期間設定 • サービス毎のバックチャネル認証 ID 有効期間設定 (CIBA) • サービス毎の認可レスポンス JWT 有効期間設定 (JARM) • サービス毎の検証コード有効期間設定 (RFC 8628)
	2.2 ~	• サービス毎のリクエスト URI 有効期間設定 (PAR)

*1 : エンタープライズプランのみ

OpenID Certification

OpenID 認証	バージョン	カテゴリー
OpenID Provider	1.1 ~	• Basic OP • Implicit OP • Hybrid OP • Config OP
	2.1 ~	• Dynamic OP • Form Post OP

FAPI OpenID Provider	2.1 ~	• Financial-grade API (FAPI) 1.0 Second Implementer's Draft
		  ◦ FAPI R/W OP w/ MTLS
		  ◦ FAPI R/W OP w/ Private Key

	2.2 ~	• Financial-grade API (FAPI) 1.0 Final
		  ◦ FAPI Adv. OP w/ MTLS
		  ◦ FAPI Adv. OP w/ MTLS, PAR
		  ◦ FAPI Adv. OP w/ Private Key
		  ◦ FAPI Adv. OP w/ Private Key, PAR
		  ◦ FAPI Adv. OP w/ MTLS, JARM
		  ◦ FAPI Adv. OP w/ Private Key, JARM
		  ◦ FAPI Adv. OP w/ MTLS, PAR, JARM
		  ◦ FAPI Adv. OP w/ Private Key, PAR, JARM
		• UK Open Banking (Based on FAPI 1 Advanced Final)
		  ◦ UK-OB Adv. OP w/ MTLS
		  ◦ UK-OB Adv. OP w/ Private Key
		• Australia CDR (Based on FAPI 1 Advanced Final)
		  ◦ AU-CDR Adv. OP w/ Private Key
		  ◦ AU-CDR Adv. OP w/ Private Key, PAR
		• Brazil Open Banking (Based on FAPI 1 Advanced Final)
		  ◦ BR-OB Adv. OP w/ MTLS
		  ◦ BR-OB Adv. OP w/ Private Key
		  ◦ BR-OB Adv. OP w/ MTLS, PAR
		  ◦ BR-OB Adv. OP w/ Private Key, PAR
		  ◦ BR-OB Adv. OP w/ MTLS, JARM
		  ◦ BR-OB Adv. OP w/ Private Key, JARM
		  ◦ BR-OB Adv. OP w/ MTLS, PAR, JARM
		  ◦ BR-OB Adv. OP w/ Private Key, PAR, JARM
		  ◦ BR-OB Adv. OP DCR
		• Financial-grade API (FAPI) 1.0 Second Implementer's Draft
		  ◦ FAPI R/W OP w/ MTLS
		  ◦ FAPI R/W OP w/ MTLS, PAR
		  ◦ FAPI R/W OP w/ Private Key
		  ◦ FAPI R/W OP w/ Private Key, PAR
		  ◦ UK-OB R/W OP w/ MTLS
		  ◦ UK-OB R/W OP w/ Private Key
		  ◦ AU-CDR R/W OP w/ Private Key
		  ◦ AU-CDR R/W OP w/ Private Key, PAR

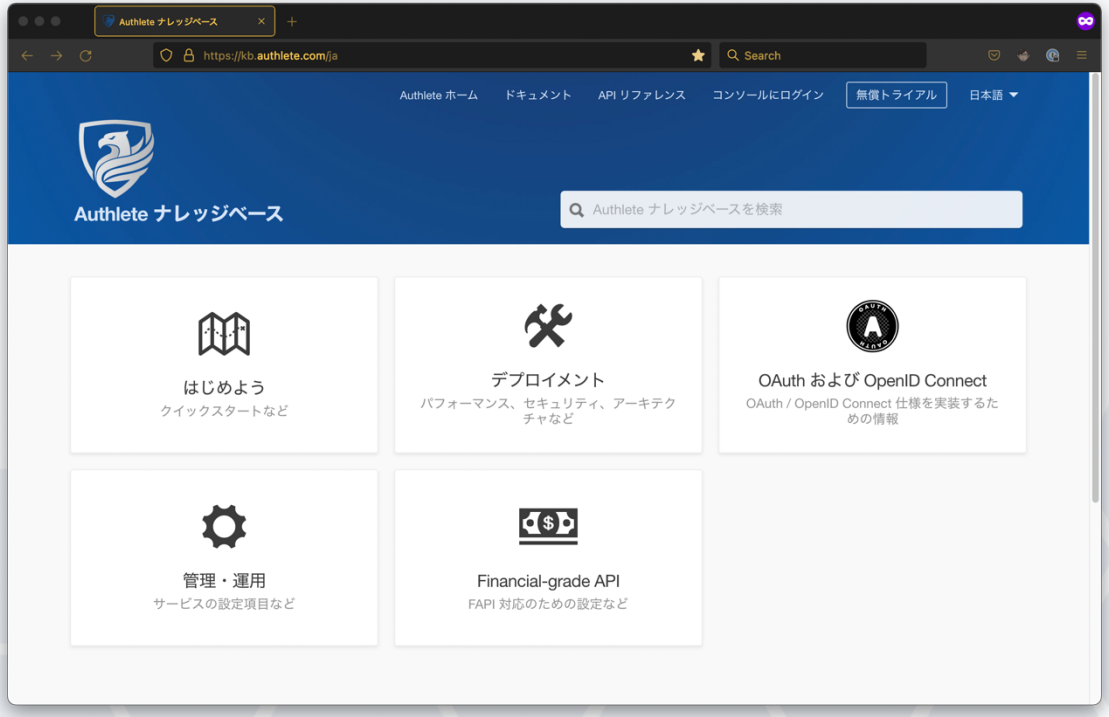
FAPI-CIBA Profile OpenID Provider	2.1 ~	• FAPI-CIBA OP Poll w/ MTLS • FAPI-CIBA OP Poll w/ Private Key • FAPI-CIBA OP Ping w/ MTLS • FAPI-CIBA OP Ping w/ Private Key

Last Updated at August 2, 2021 by Authlete, Inc.

See https://www.authlete.com/ja/legal/spec_sheet/ for details

Authleteナレッジベース https://kb.authlete.com/ja/

- Authleteの使いこなしかたを解説
- Authleteを使わない場合でもOAuth / OIDCサーバーを設計・実装する際のヒントとして有用（だと思います）



AUTHLETE

トークン発行のコンテキスト

トークンのライフサイクル

- 発行・利用・更新
- 失効・変更
- 無効化・削除

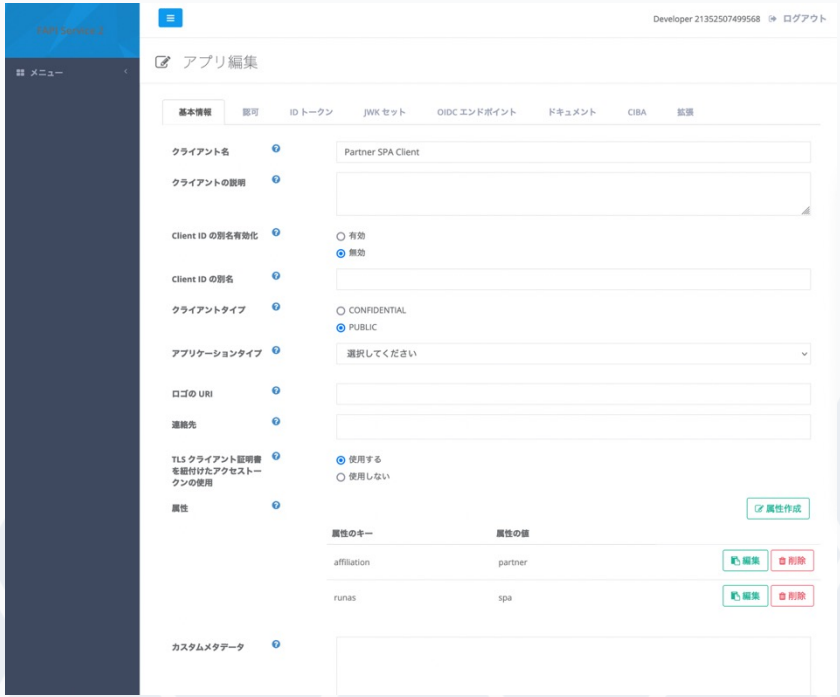
トークン発行の経緯（コンテキスト）

- クライアントの特性
- 対象リソース
- エンティティとメッセージの「確からしさ」
- リソースオーナーの意図

クライアントの特性

誰のものか? 何をする? どこに存在する? 作りは?

- 認可サーバー側で管理
  - 1st / 3rd パーティ
  - コンフィデンシャル / パブリック
  - Web Server App / Native App / SPA / …
- クライアント属性



対象リソース

要求対象のスコープ

- 静的
  - スコープ属性
- 動的
  - スコープのパラメーター化
  - Rich Authorization Requests (RAR)
  - “Lodging Intent Pattern”

対象リソース

スコープのパラメーター化

- 「パラメーター化されたスコープ」の利用

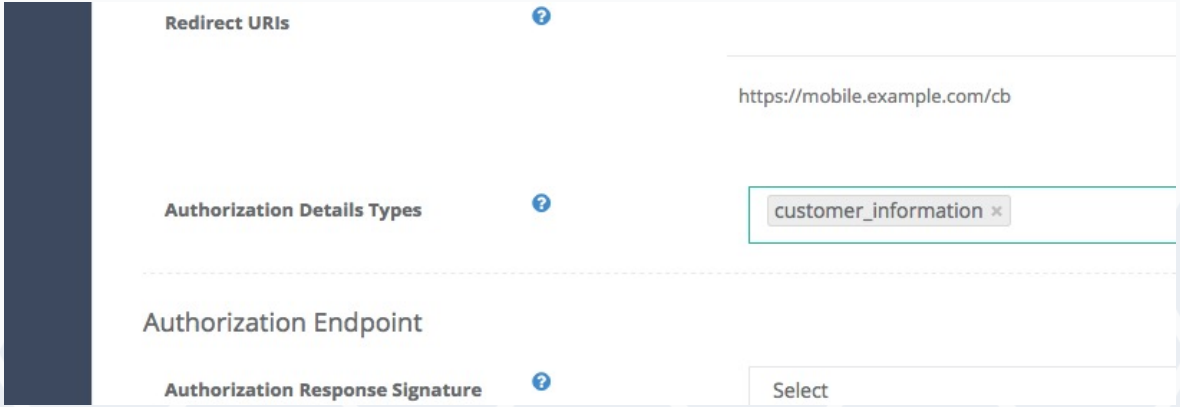
https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/parameterized-scopes

対象リソース

Rich Authorization Requests (RAR)

- Rich Authorization Requests

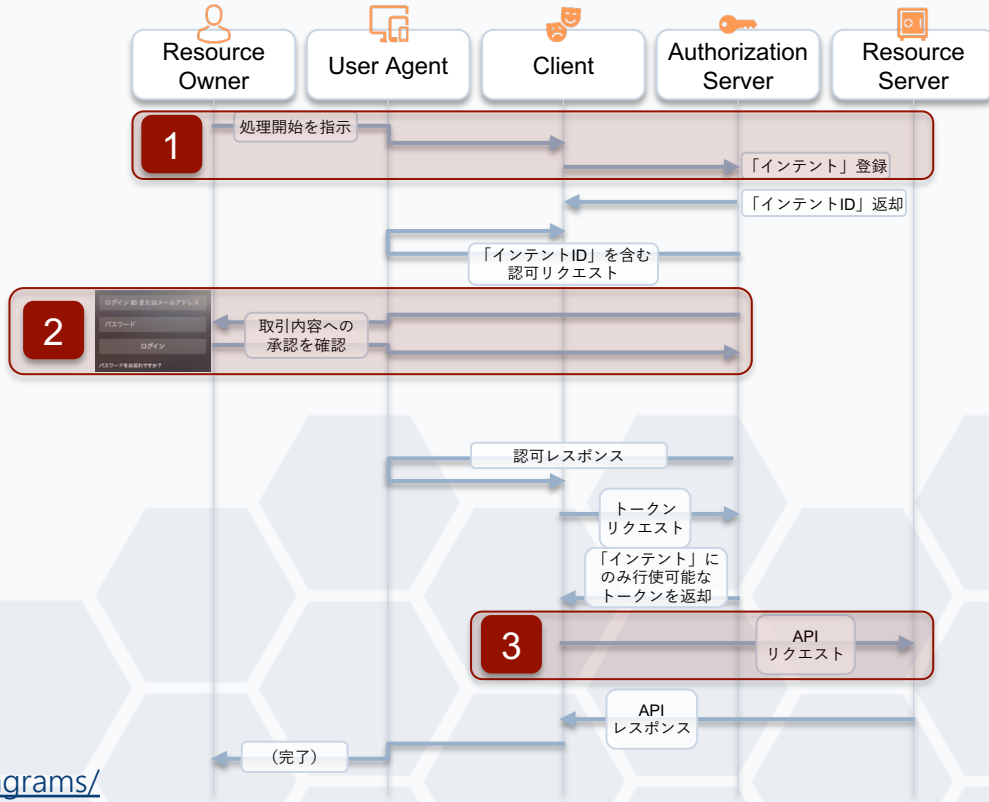
  https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/rar



対象リソース

“Lodging Intent Pattern”

1. サードパーティが銀行の API に口座情報取得や決済指図伝達などの「インテント」を事前登録
2. 登録された内容を、利用者が銀行のWebサイトやアプリにて確認・承認
3. 承認後、サードパーティが「インテント」の内容の実行を銀行のAPIにリクエスト



cf. https://www.authlete.com/ja/resources/templates/sequence-diagrams/

エンティティとメッセージの確からしさ

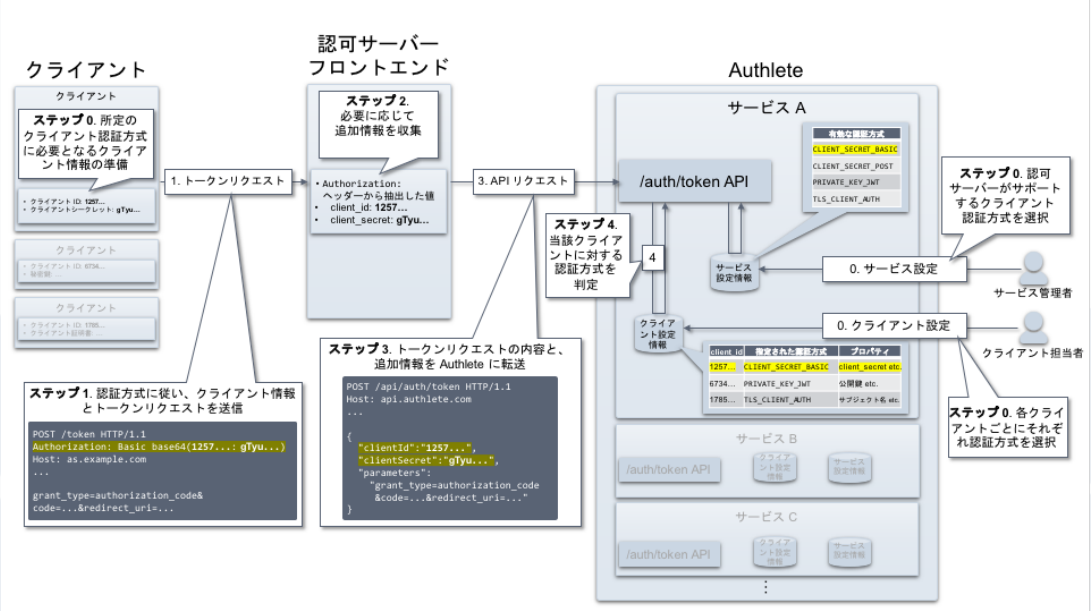
- クライアント認証
- リクエスト・レスポンスの保護

エンティティとメッセージの確からしさ

クライアント認証

- クライアント認証の設定

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/client-authentication

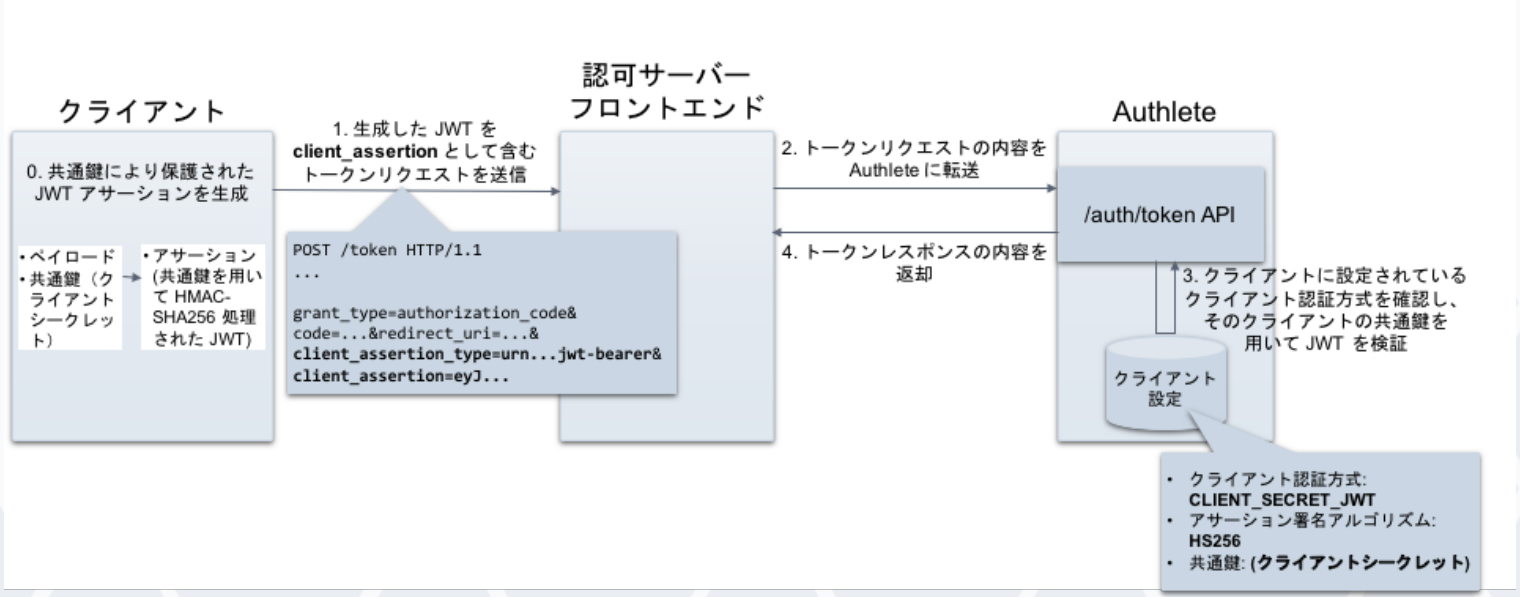


エンティティとメッセージの確からしさ

クライアント認証

- client_secret_jwt によるクライアント認証

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/client-auth-client-secret-jwt

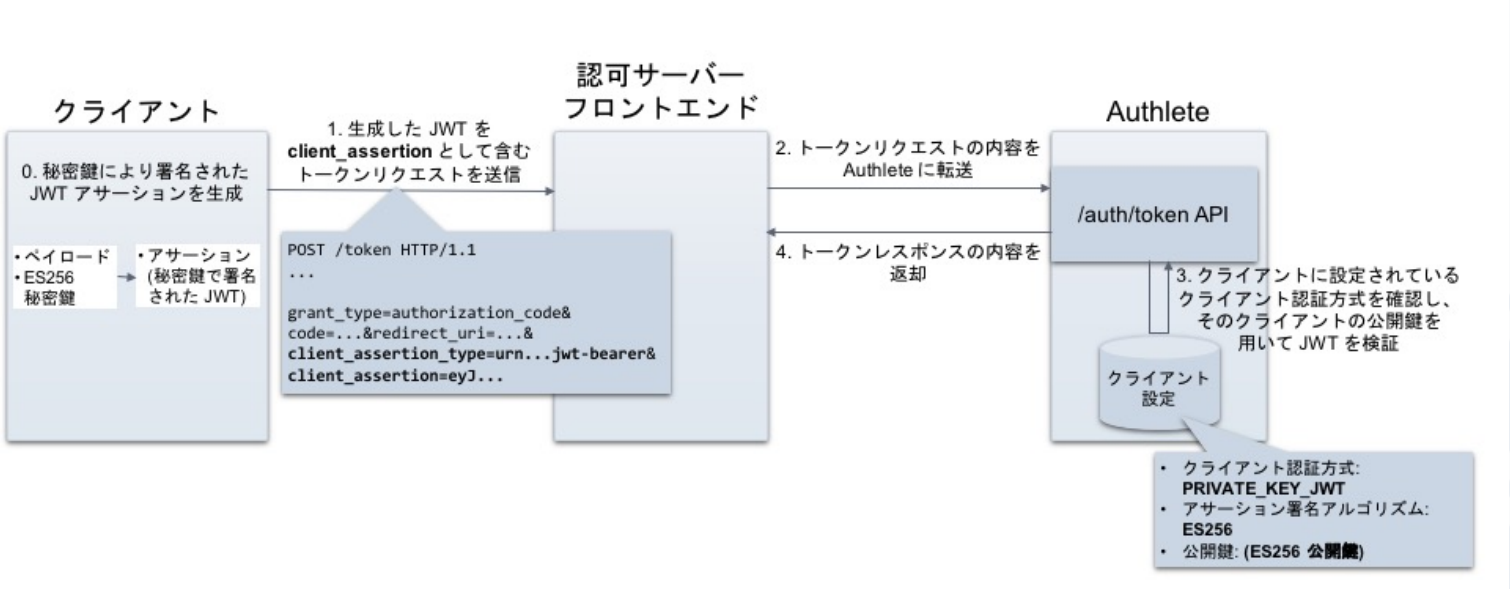


エンティティとメッセージの確からしさ

クライアント認証

- private_key_jwt によるクライアント認証

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/client-auth-private-key-jwt

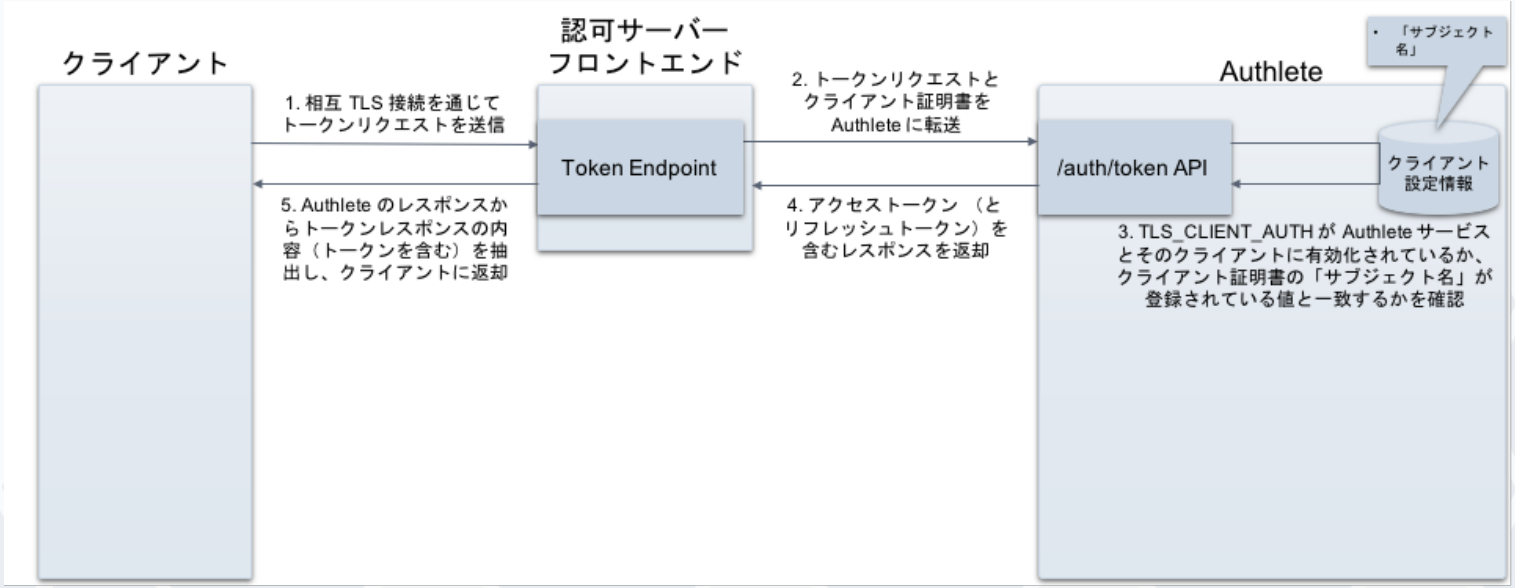


エンティティとメッセージの確からしさ

クライアント認証

- tls_client_auth によるクライアント認証

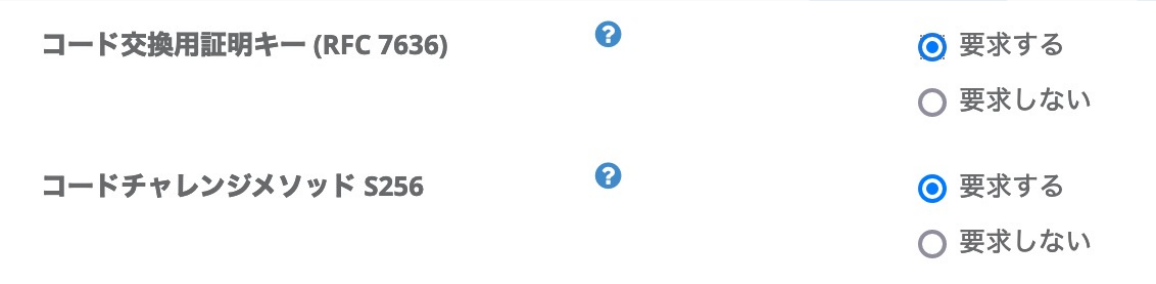
https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/tls-client-auth



エンティティとメッセージの確からしさ

PKCE (Proof Key for Code Exchange by OAuth Public Clients)

- 認可リクエストにおける PKCE 利用の強制化
  https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/requiring-pkce
- 認可リクエストでの PKCE 利用における "S256" 指定の強制化
  https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/requiring-s256

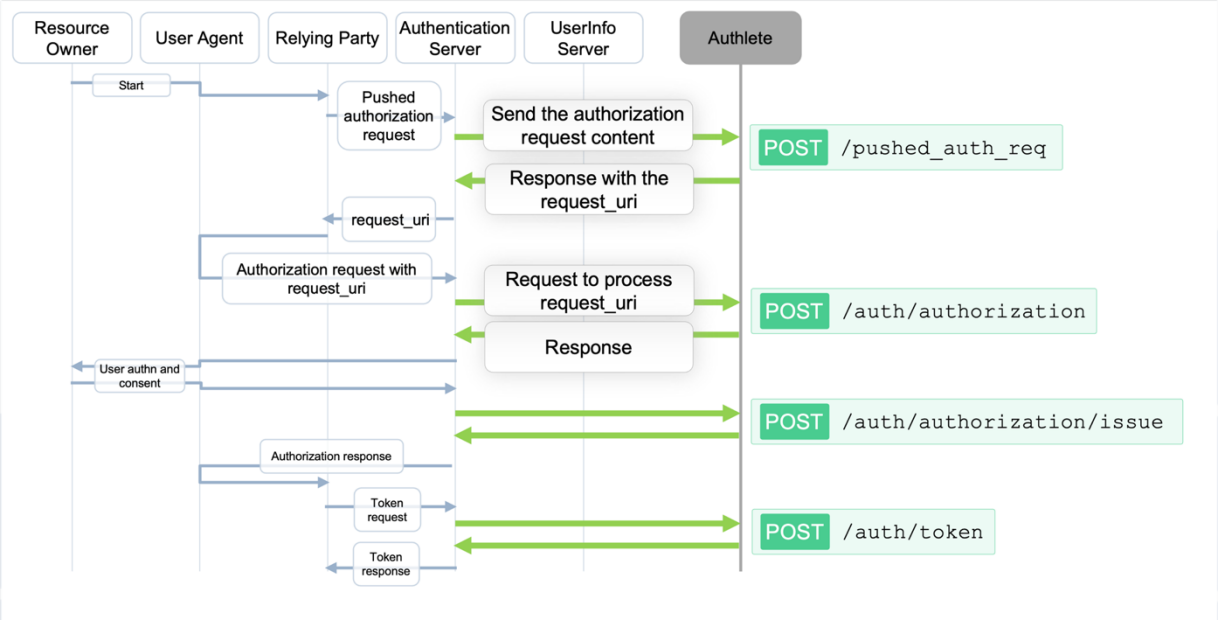


エンティティとメッセージの確からしさ

Pushed Authorization Requests (PAR)

- Pushed Authorization Requests (PAR)

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/pushed-authorization-requests-par



エンティティとメッセージの確からしさ

JARM

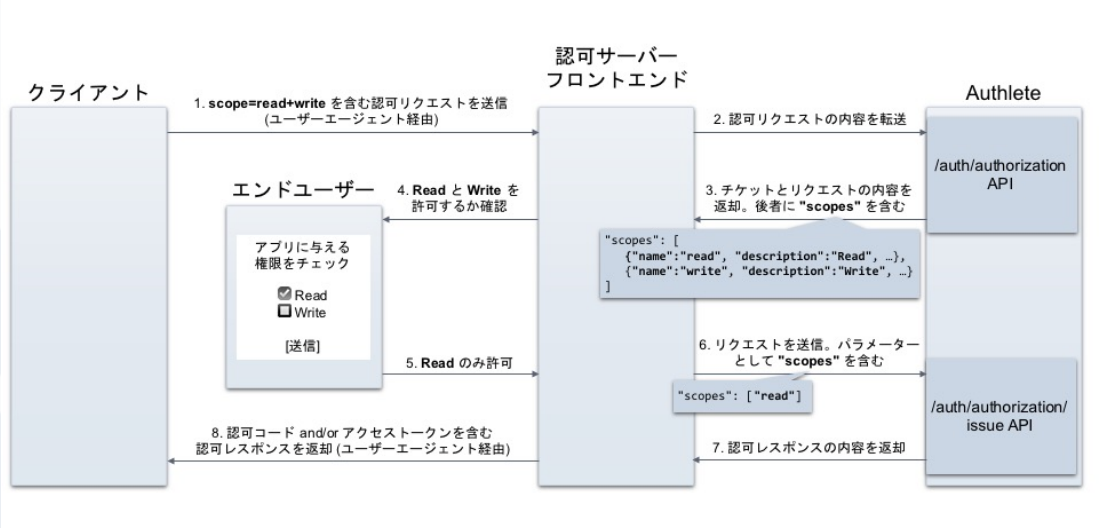
- JARM の有効化

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/enabling-jarm

リソースオーナーの意図

- エンドユーザーに認可するスコープを選択させる方法

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/letting-resource-owner-choose-scopes



AUTHLETE

発行するトークンの特性

トークンの特性

- 有効期間
  - サービス / スコープ / クライアント / トランザクション
- トークン属性
  - 識別子型 / 内包型
- Proof of Possession (PoP)
  - MTLS / DPoP

有効期間

- どの単位で設定するか
  - 認可サーバー
  - スコープ
  - クライアント
  - さらに細かく?

有効期間

サービス・スコープ・クライアント単位の設定

- トークンの有効期間の計算ロジック

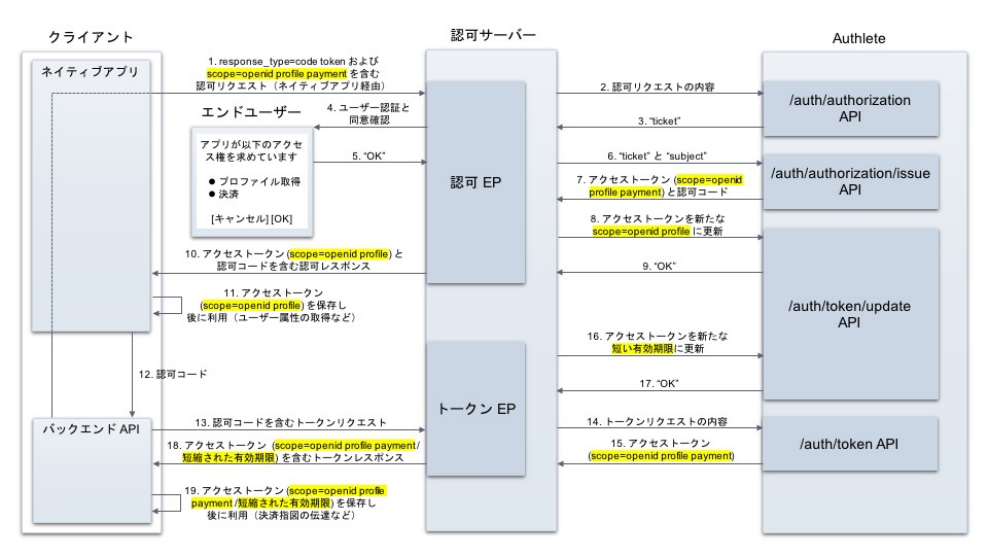
https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/how-to-calculate-token-duration

有効期間

トランザクション単位での有効期間設定

- ハイブリッドフローによる 2 つのアクセストークンの発行

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/hybrid-flow-access-tokens



トークン属性

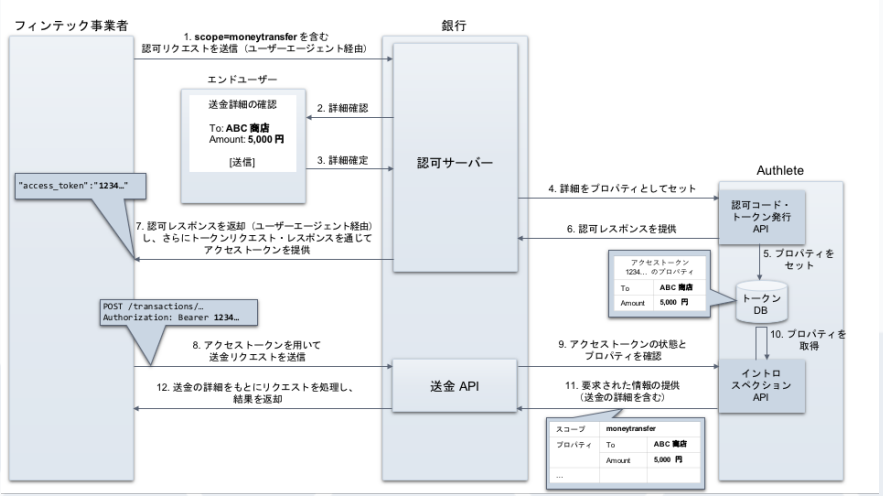
- 一般的にリソースサーバーはアクセストークンが有効かどうか以外の情報も必要とする
  - トークン発行のコンテキスト
    - 例: クライアントID、ユーザー識別子、スコープ、...
  - そのコンテキストに付随する情報
    - 例: クライアント区分、ユーザーのロール、ユーザーが承認した内容、...

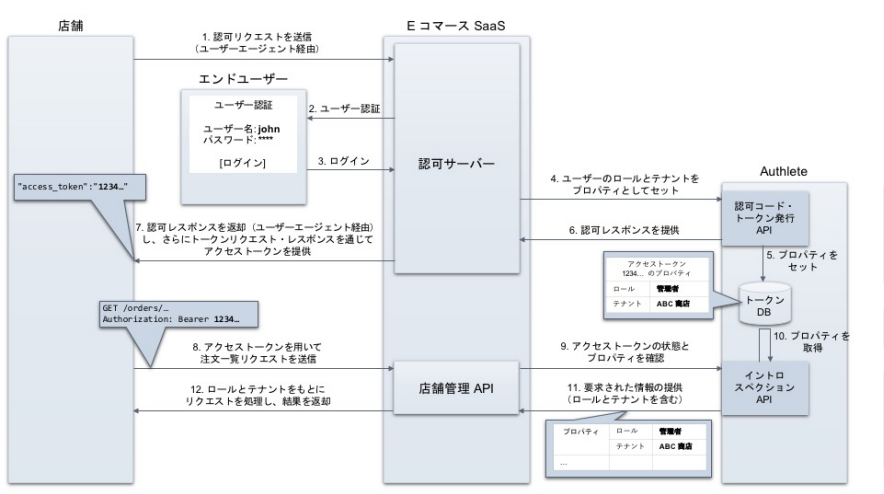
トークン属性

アクセストークンと属性とのひもづけ

- トークンに任意のプロパティをひもづける方法

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/extra-properties





トークン属性

リソースサーバーへの属性の渡しかた

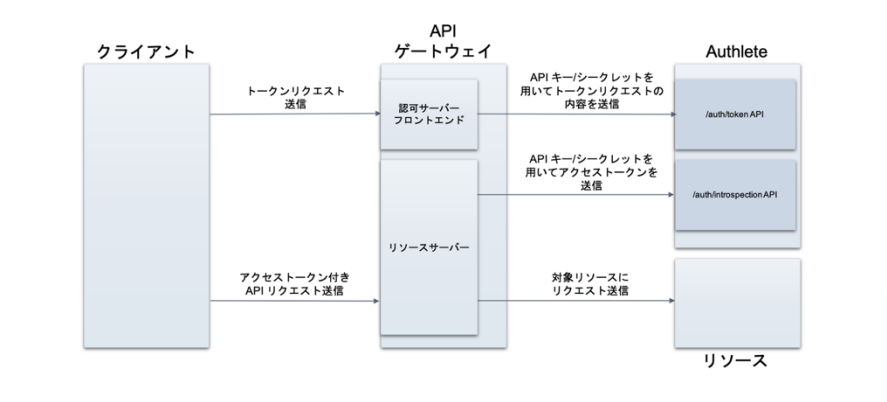
- 識別子型アクセストークン
  - イントロスペクションの結果として属性情報を提供
- 内包型アクセストークン
  - 属性情報をトークンに埋め込んで提供
- ハイブリッド型アクセストークン
  - 識別子型・内包型の両方の方法を使用

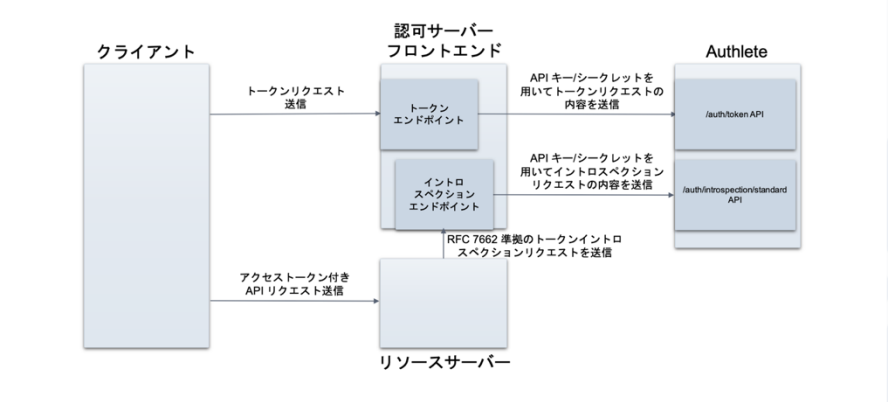
トークン属性

イントロスペクションによる属性情報提供

- 2 種類のイントロスペクション API の使い分け

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/two-introspection-apis



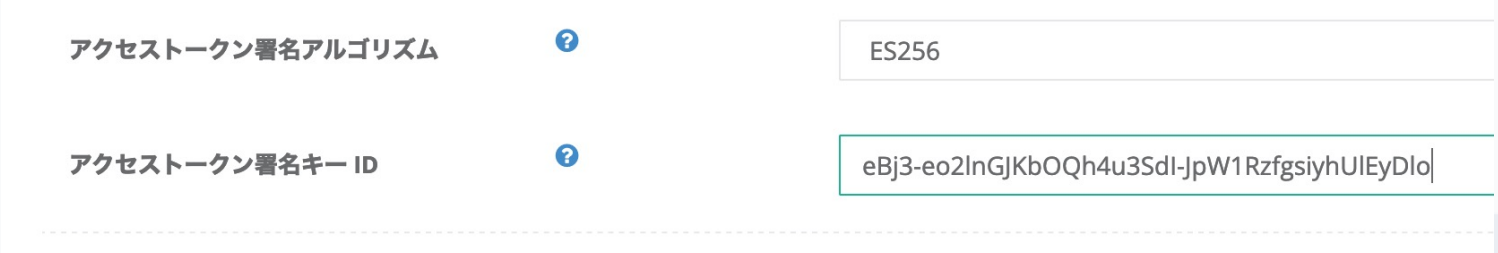


トークン属性

トークンへの属性情報の埋め込み

- 「JWT ベースのアクセストークン」の有効化

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/jwt-based-access-token

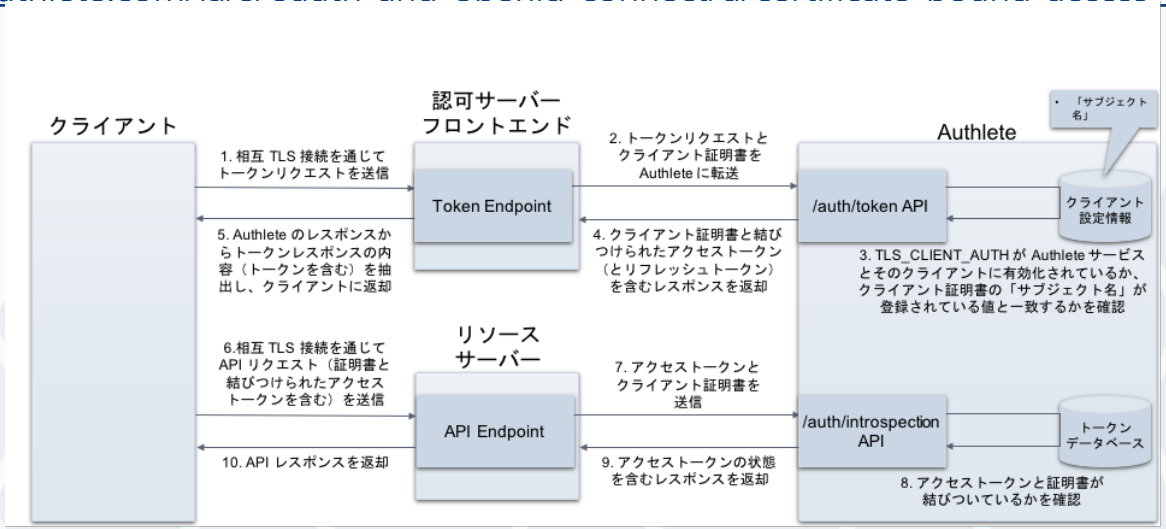


Proof of Possession (PoP)

MTLS

- TLS クライアント証明書を結びつけたアクセストークンの発行

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/certificate-bound-access-tokens



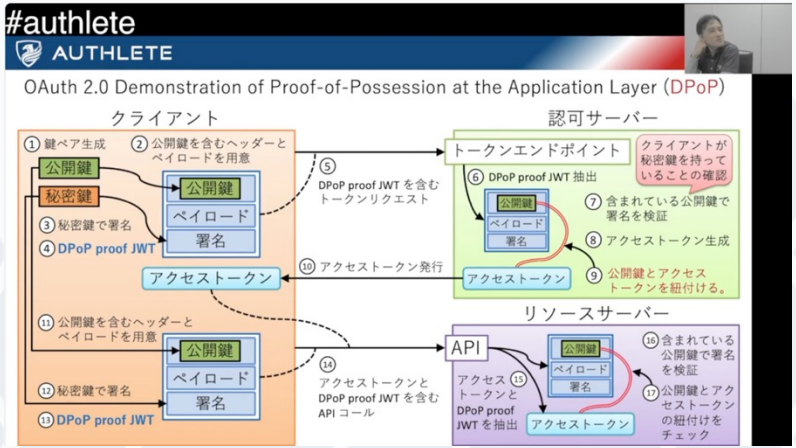
Proof of Possession (PoP)

DPoP

- OAuth & OIDC 勉強会 【アクセストークン編】

  5. Proof of Possession

  https://www.authlete.com/ja/resources/videos/20200422/05/

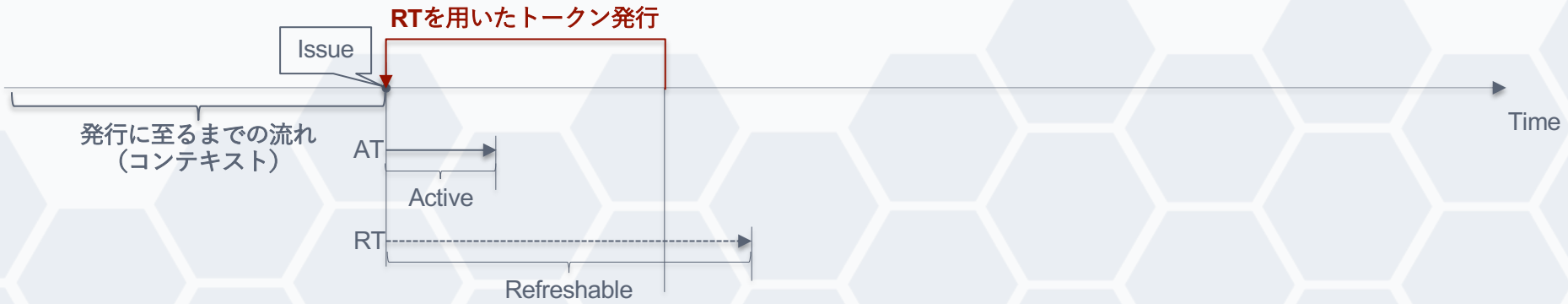


AUTHLETE

発行後のトークン操作

リフレッシュされたトークンの特性は?

- トークンの値: 引き継ぐ or 更新する
- 有効期間: 引き継ぐ or リセットする



リフレッシュされたトークンの特性

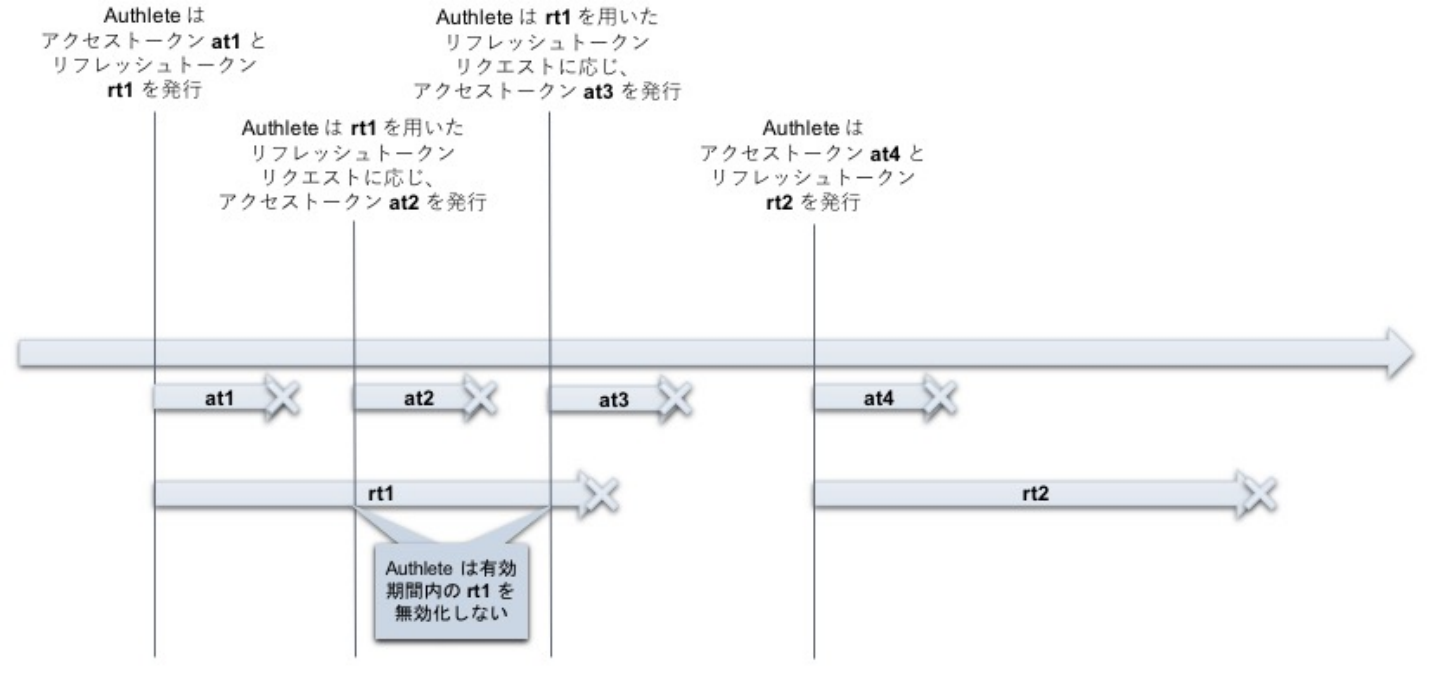
トークンリフレッシュのポリシー

- リフレッシュトークンの継続利用設定

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/refreshing-refresh-tokens

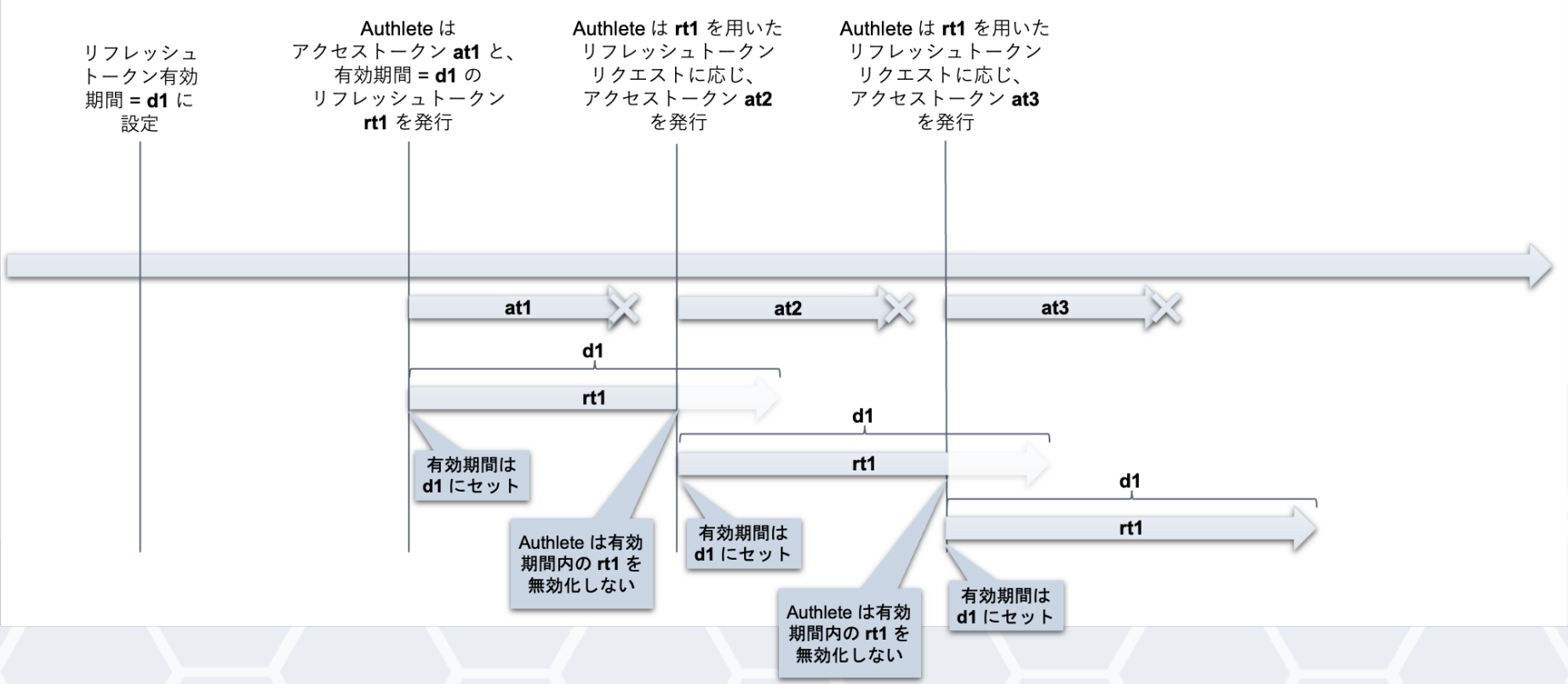
リフレッシュされたトークンの特性

1. 継続使用する & 有効期間をリセットしない



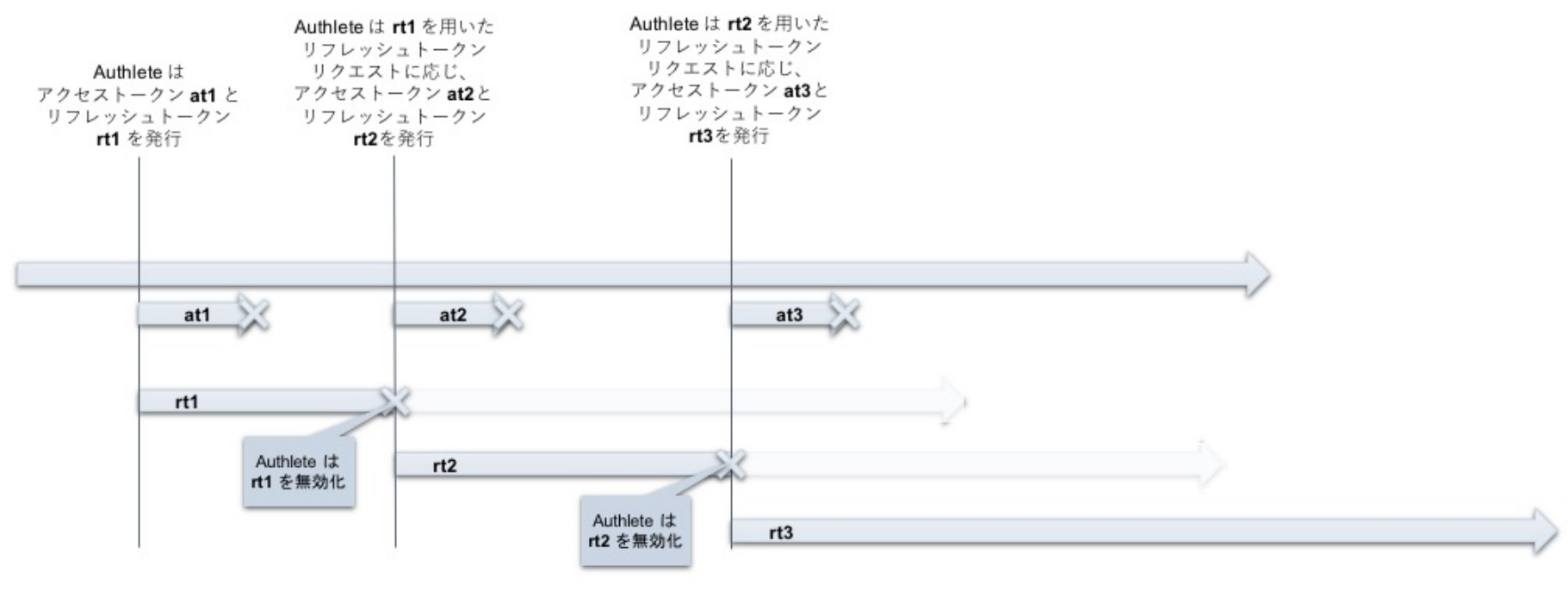
リフレッシュされたトークンの特性

2. 継続使用する & 有効期間をリセットする



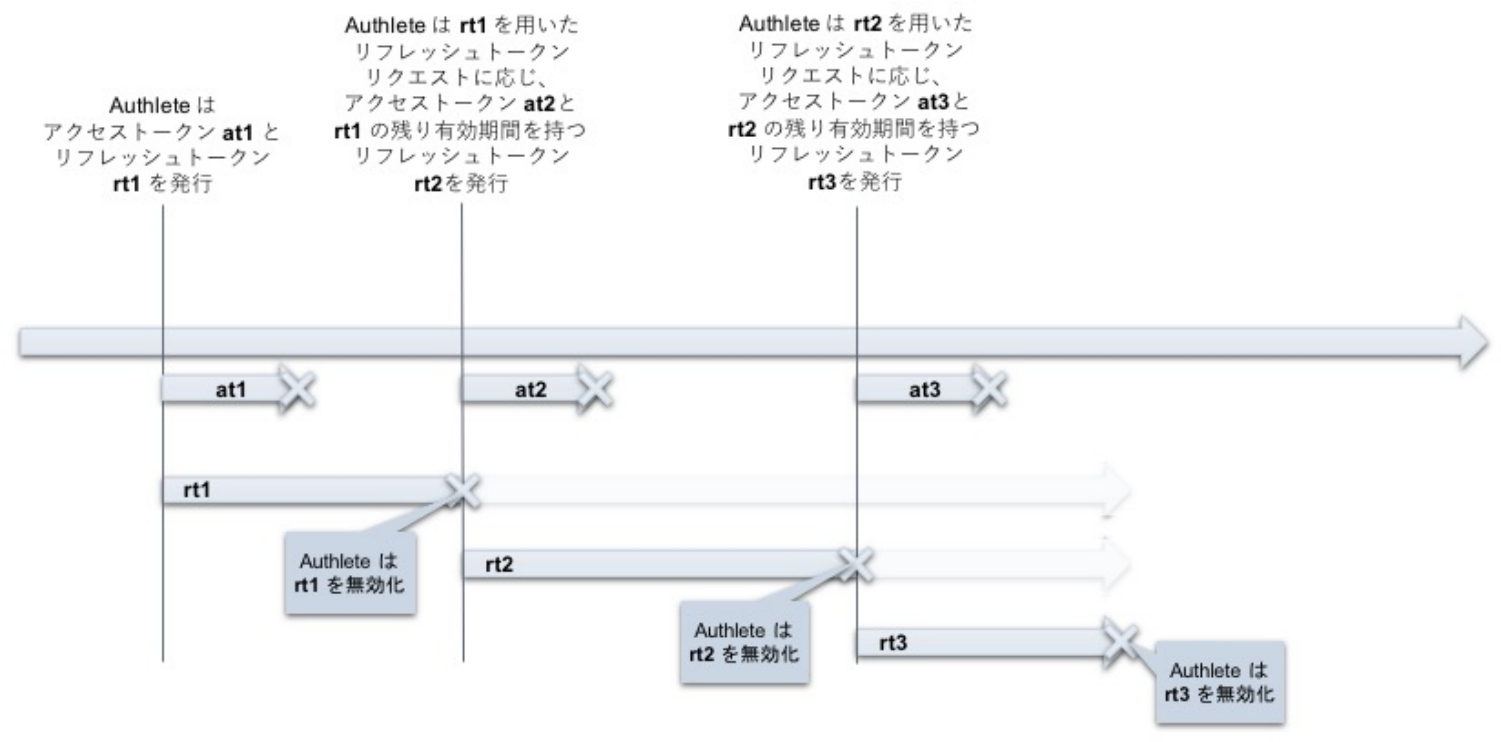
リフレッシュされたトークンの特性

3. 継続使用しない & 有効期間を引き継がない



リフレッシュされたトークンの特性

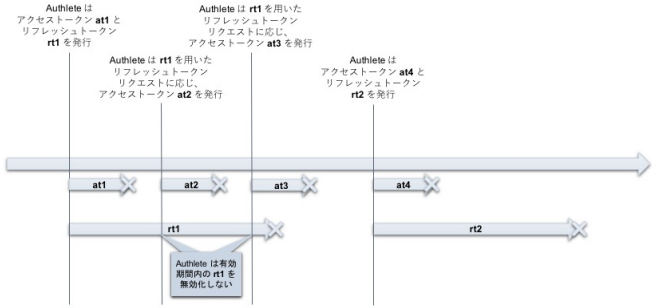
4. 継続使用しない & 有効期間を引き継ぐ



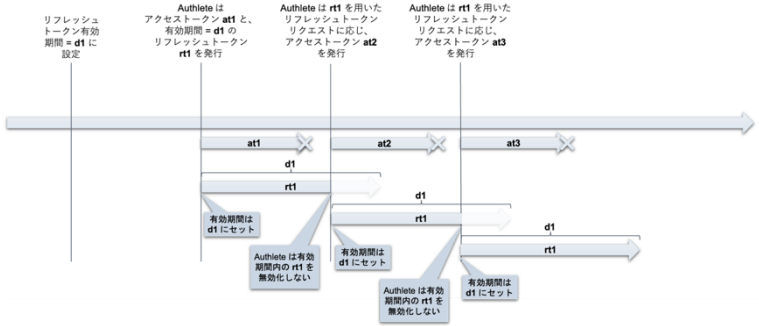
Q. どれを選びますか?

(1 or 2 or 3 or 4)

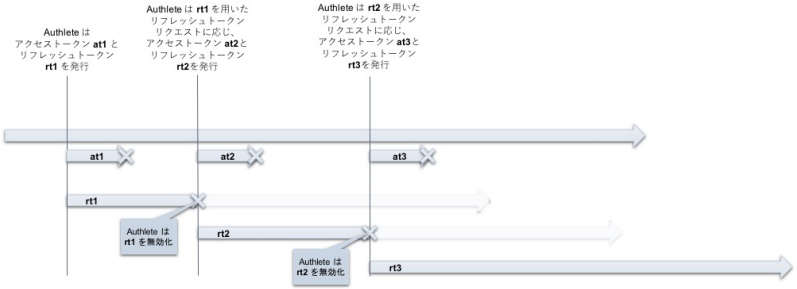
1. 継続使用する & 有効期間をリセットしない



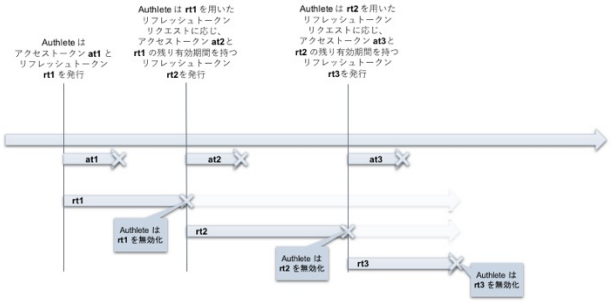
2. 継続使用する & 有効期間をリセットする



3. 継続使用しない & 有効期間を引き継がない



4. 継続使用しない & 有効期間を引き継ぐ



コンテキストが同等な場合にどう発行する?

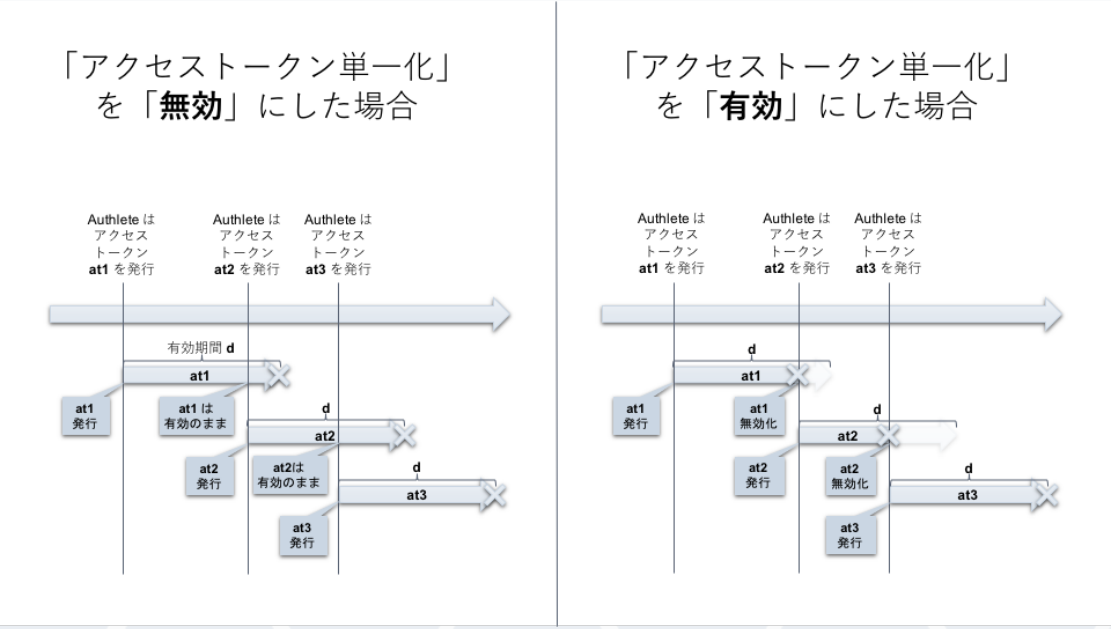
- とくに「同一のクライアント・リソースオーナーの組み合わせ」にトークンを発行する場合
- 発行済みの ATとRT は無効化するのか、そのままにするのか

コンテキストが同等な場合

トークン単一化ポリシー

- アクセストークンの単一化

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/single-access-token-per-subject



誰が失効処理の起点となるか?

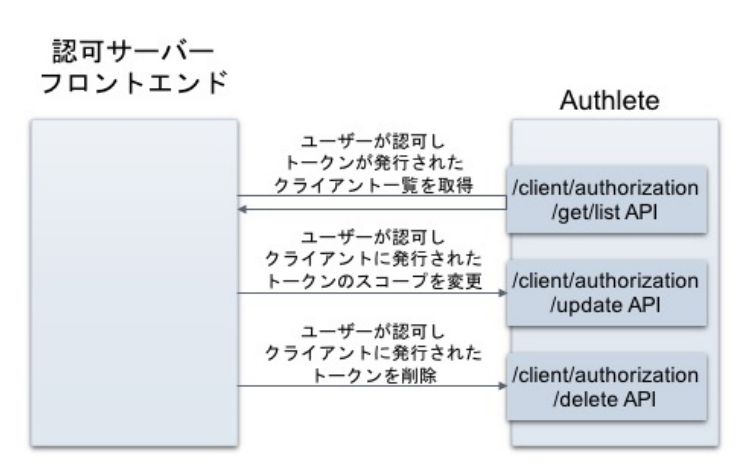
- クライアント
  - RFC 7009 OAuth 2.0 Token Revocation
- リソースオーナー
  - あるクライアントに関連するすべてのアクセストークンを失効
- 認可サーバー
  - あるリソースオーナーに関連するすべてのアクセストークンを失効

失効処理の起点

リソースオーナー起点のトークン失効

- ユーザーがクライアントに付与した認可の取得・変更・取り消し

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/managing-issued-tokens-granted-by-each-user



クリーンアップ

- 使用されていないトークンに関する Authlete の削除ポリシー

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/authletes-policy-on-sweeping-unused-tokens

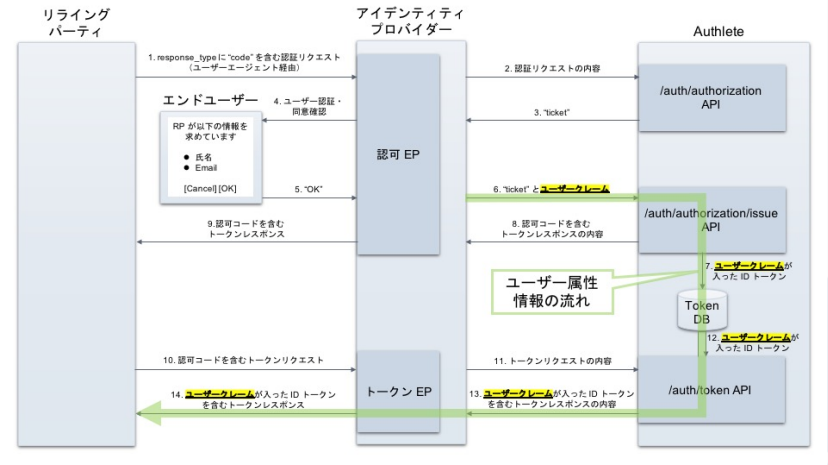
AUTHLETE

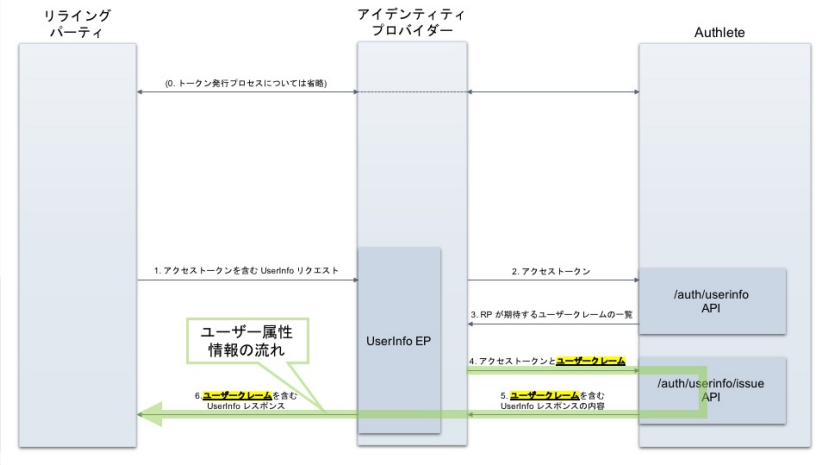
クライアントへのユーザー属性情報提供

ID トークンと UserInfo の使いわけ

- ユーザー属性情報の提供方式と、各方式における Authlete の処理方法

https://kb.authlete.com/ja/s/deployment/a/providing-user-attributes

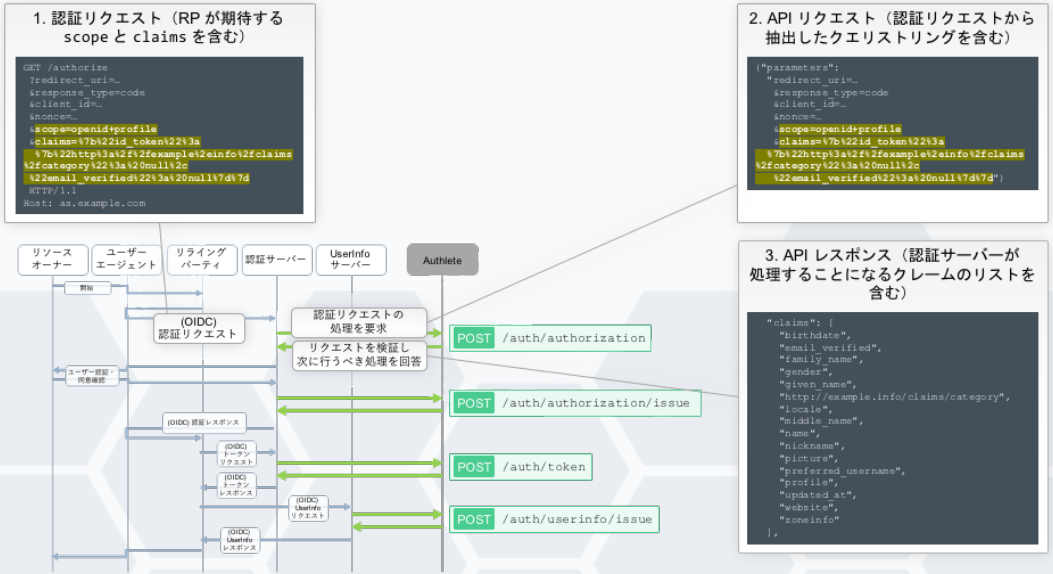




提供するユーザー属性情報の決定

- IDトークンに追加する「クレーム」の判別

https://kb.authlete.com/ja/s/oauth-and-openid-connect/a/claims-in-id-token



AUTHLETE

まとめ

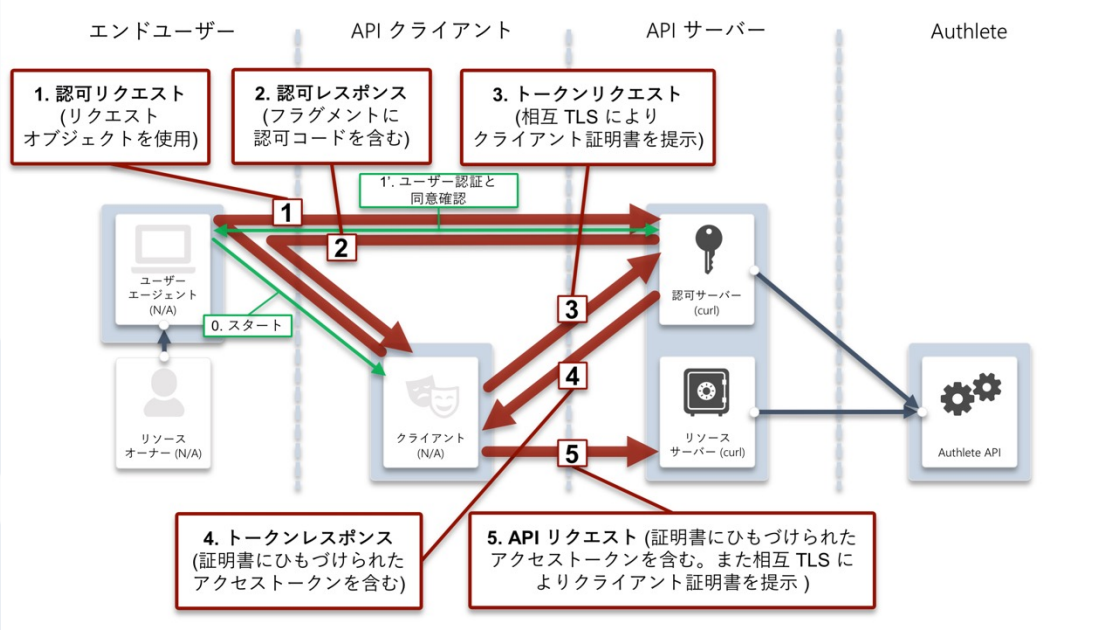
OAuth / OIDC プロファイリングの要点

- どのような場合にどのようなトークンを発行するか
  - 場合: 認可のコンテキスト
  - トークン: APIの実行に必要な情報
- ライフサイクル管理とセキュリティ対策
  - 更新・失効・無効化のポリシー
  - 拡張仕様・プラクティス・セキュリティプロファイル

セキュリティプロファイルの適用

- Financial-grade API (FAPI) Basics

https://www.authlete.com/ja/developers/tutorial/fapi/



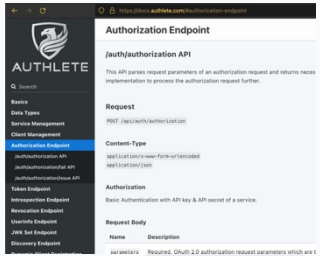
参考情報

- シーケンス図テンプレート

  https://www.authlete.com/ja/resources/templates/sequence-diagrams/

- Authlete API Reference

  https://docs.authlete.com/



- authlete-java-common

  https://authlete.github.io/authlete-java-common/

AUTHLETE

Thank You

www.linkedin.com/in/tatsuokudo

www.authlete.com