ThemiStruct
テミストラクト

Web API の保護に係る技術動向と、API連携認証システムのあり方

株式会社オージス総研
事業開発本部 テミストラクトソリューション部
八幡 孝

自己紹介

八幡 孝

株式会社オージス総研
統合認証ソリューション担当

OpenAMコンソーシアム
副会長

OpenIDファウンデーション・ジャパン
Enterprise Identity WG
リーダー

統合認証ソリューションを15年以上やってきました

ThemiStruct-WAM　シングルサインオン 認証基盤ソリューション

ThemiStruct-IDM　ID管理ソリューション

ThemiStruct-CM　電子証明書発行・管理 ソリューション

ワンタイムパスワードソリューション
ThemiStruct-OTP

システム監視ソリューション
ThemiStruct-MONITOR

ThemiStruct（テミストラクト）
Identity Platform AWS対応版

APIエコノミー時代の 統合認証パッケージ

認証基盤のユースケースが拡大

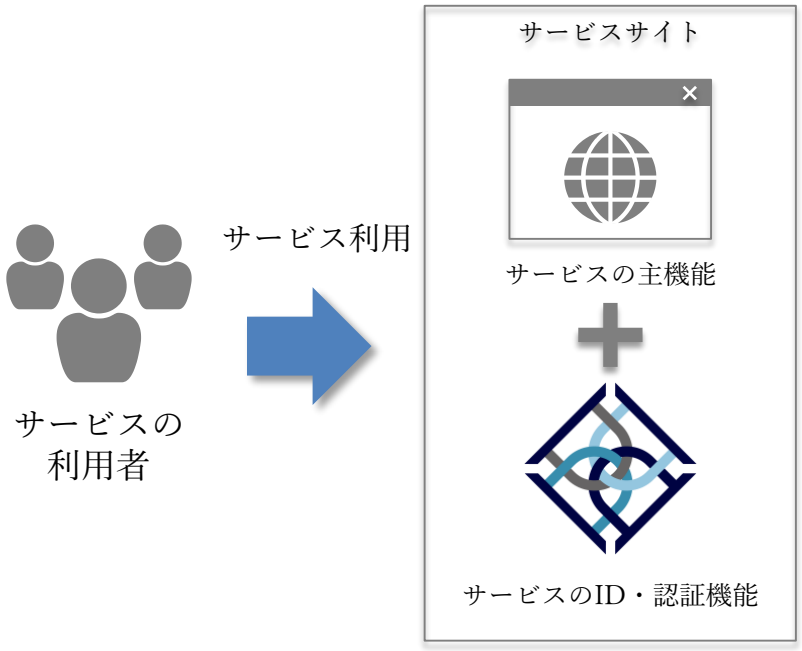
ユースケース	狙い・特長
社内システム利用のガバナンス強化	認証処理の一元化、人事システム等と連動したタイムリーなIDメンテナンス。
取引先へのシステム提供	取引先ユーザーの確実な認証。IPアドレスや電子証明書の併用。
クラウドサービス利用時、スマホ・タブレット利用時の認証強化	社外からの利用の制限。社外での利用時の追加の認証の実施。社用端末の識別。 クラウドサービスのIDメンテナンス。
顧客（一般消費者）向けの情報提供、サービス提供	SSOによる顧客への利便性の提供。複数アプリへの展開。収集した属性の活用。他社サービスとの連携。

これからは モバイルアプリ と オープンAPI対応 が必要に

ユースケース	狙い・特長
社内システム利用のガバナンス強化	認証処理の一元化、人事システム等と連動したタイムリーなIDメンテナンス。
取引先へのシステム提供	取引先ユーザーの確実な認証。IPアドレスや電子証明書の併用。
クラウドサービス利用時、スマホ・タブレット利用時の認証強化	社外からの利用の制限。社外での利用時の追加の認証の実施。社用端末の識別。 クラウドサービスのIDメンテナンス。
顧客（一般消費者）向けの情報提供、サービス提供	SSOによる顧客への利便性の提供。複数アプリへの展開。収集した属性の活用。他社サービスとの連携。
モバイルアプリ化、オープンAPI活用によるアプリ機能、サービスの高度化	• アプリ内にパスワード保存不要な安全な認証方式、SSOに対応できる認証方式への対応。 • 利用者同意に基づく必要最少権限でのデータ連携を実現する認証・認可方式への対応。

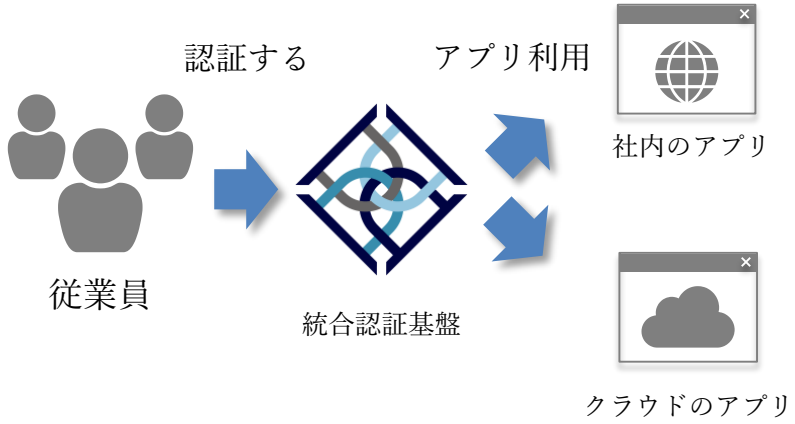
統合認証パッケージ ThemiStruct Identity Platform

顧客向けサイト領域に活用



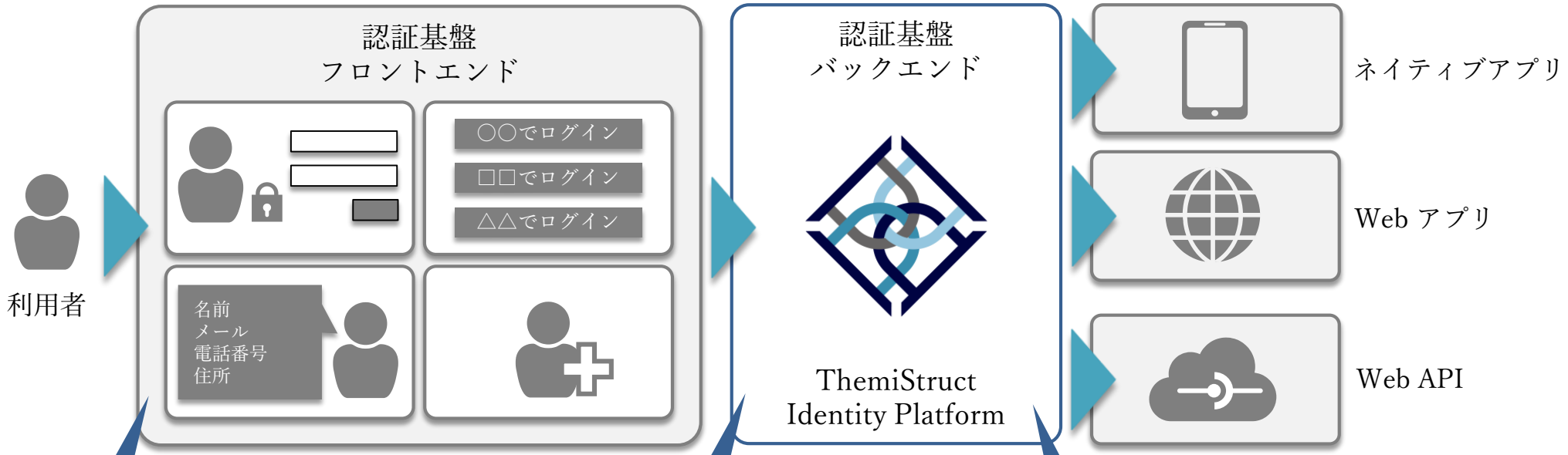
サービスサイトの
ID、認証の共通機能として利用

エンタープライズ領域に活用



エンタープライズアプリ群の
SSOやアクセス制御の基盤として利用

ThemiStruct Identity Platform を『顧客向けサイト』に活用



IDと認証、アクセス認可の機能をアプリに組み込む

利用者のID情報の管理や認証などの共通機能を、サイトの統一されたデザインで組み込むことができます。

Web APIの認証・アクセス認可に対応

OAuth 2.0 を使ったAPI連携認証システムが実現でき、モバイルアプリ、オープンAPIを安全に公開できます。

短期間での導入が可能

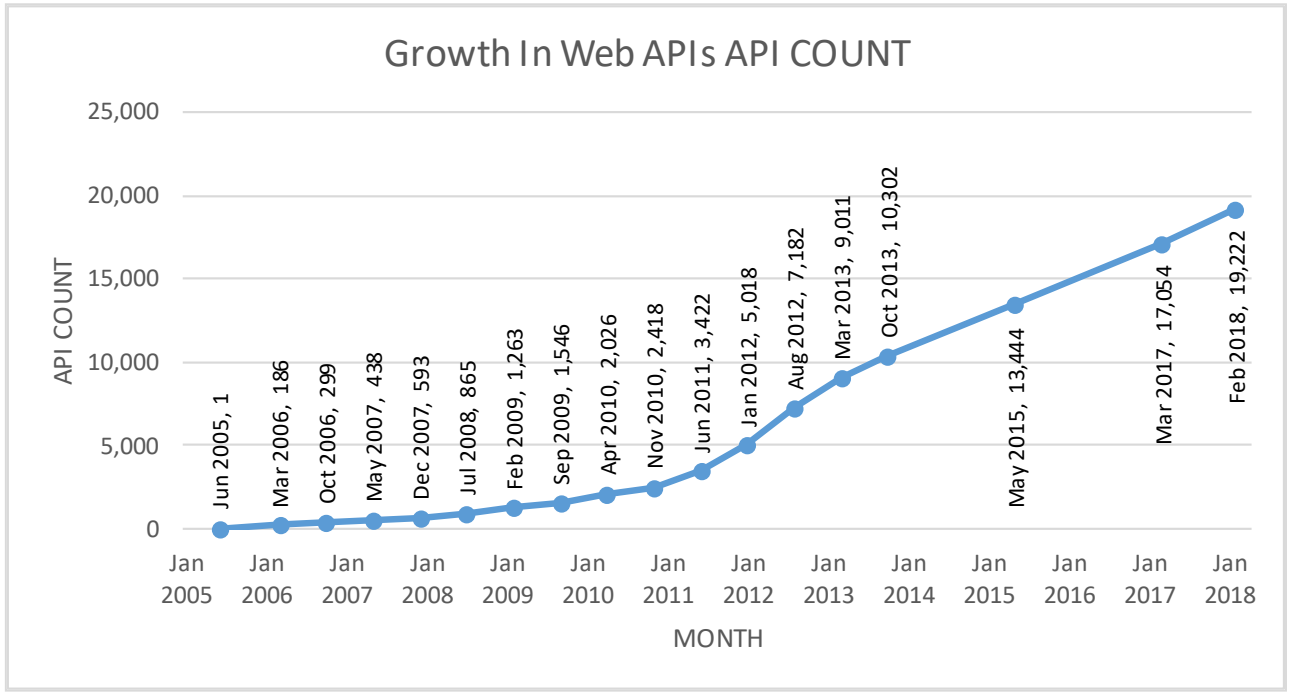
スモールスタート、突発的アクセス集中に対応できる伸縮可能かつ事業継続性を考慮した認証基盤が、2時間でセットアップできます。

Web API を活用する際の API連携認証システムの必要性

Web API の利用が広がっている

増加を続ける Web API（公開型）

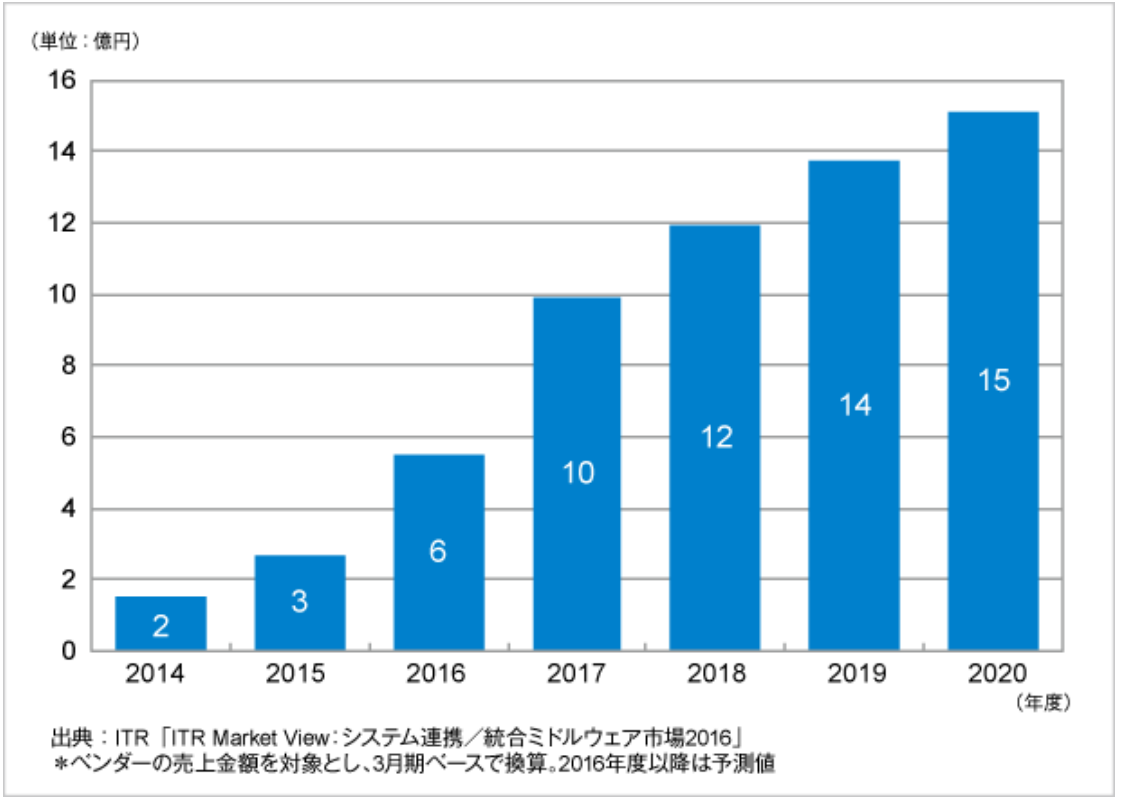
ProgrammableWebの情報を基に当社で加筆・グラフ化



引用元: https://www.programmableweb.com/api-research

「API管理市場売上金額推移および予測」

ITR社プレスリリースより



引用元: https://www.itr.co.jp/company/press/161213PR.html

Web API は xTECH の中核技術である

> APIの公開、およびそれに伴うデータの公開による新たな事業の創出への期待が需要の喚起を加速させ、デジタルイノベーションやデジタルトランスフォーメーションを実現するコア技術のひとつとして導入が進む

ITR社「**ITRがAPI管理市場規模推移および予測を発表**」(2016年12月)より
引用元: https://www.itr.co.jp/company/press/161213PR.html

> オープンAPIは［中略］他の事業者等と金融機関が協働して、それぞれの保有する情報やサービスを組み合わせ、あるいはお互いに知恵を絞り、オープン・イノベーションを実現していくためのキー・テクノロジーの一つと位置づけられる。

「**オープン API のあり方に関する検討会報告書**」(2017年7月)より
引用元: https://www.zenginkyo.or.jp/fileadmin/res/abstract/council/openapi/openapi_report_1.pdf

Web API を公開する際の課題

ロ クローズドAPI

- 組織内での Web API の利用
- 組織の境界内からのみアクセス可能

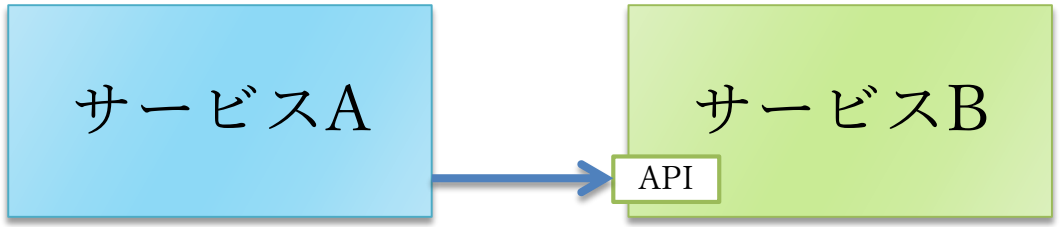
ロ オープンAPI

- 組織の境界の外側の第三者への Web API の公開
- Web API 利用者の**認証**、機能やデータへの**アクセス権管理**が不可欠

Web API 利用の形態

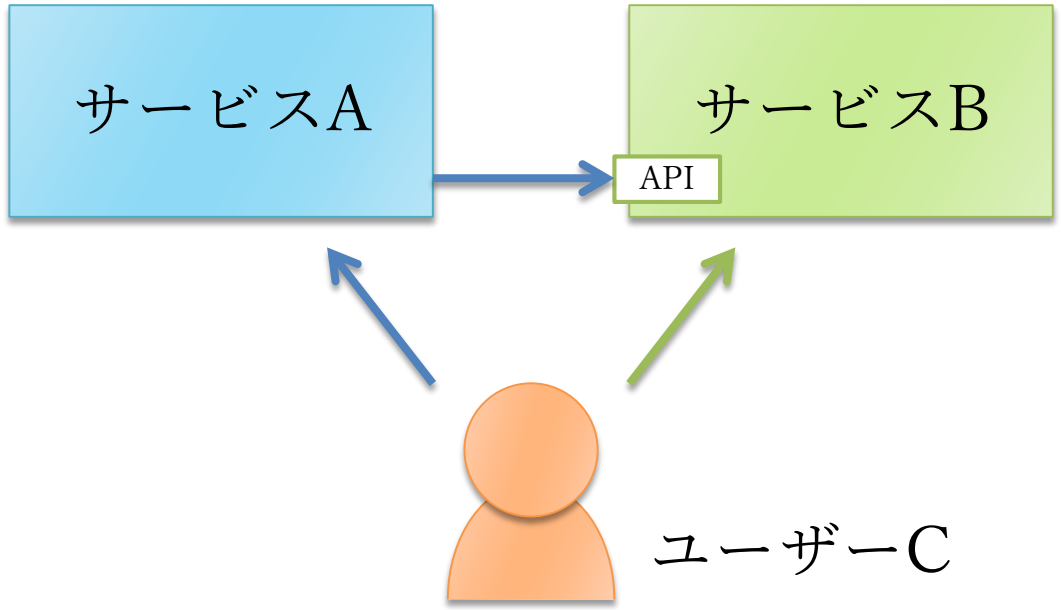
2者間でのAPI利用

- サービスA が サービスB の機能を利用するために API にアクセスする。
- サービスB は サービスA 向けの機能・データを提供する。



3者間でのAPI利用

- ユーザーC は サービスA、サービスB の利用者である。
- サービスA は、ユーザーCの意図により、ユーザーC に代わり サービスB の API にアクセスする。
- サービスB は ユーザーC 向けの機能・データーを提供する。



2者間でのAPI利用時の認証

- **API のアクセス元を認証する。**
  - APIキー
  - BASIC認証（IDとパスワード）
  - クライアント証明書
  - OAuth 2.0 Client Credentials Grant
  - …
- **アクセス元が認証できれば、API はアクセス権を判断できる。**

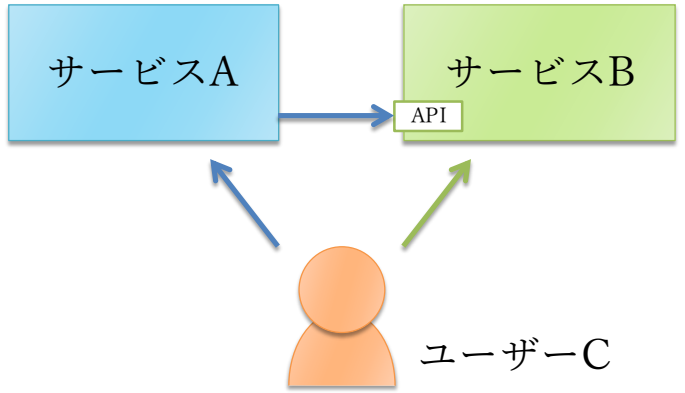
3者間でのAPI利用時の認証

ロ ユーザーC は サービスA に、アクセスできるデーターや操作を限定して、サービスB の API へアクセスさせたい。

- ユーザーIDとパスワードなど、クレデンシャル情報を渡す方式では、全権限をサービスAに与えてしまう。

ロ 現時点では OAuth 2.0 の利用が唯一の選択肢

- ユーザーC の同意に基づき、ユーザーC が意図した範囲の限定された権限で API へのアクセスを許可する方式。
- OAuth 2.0 Authorization Code Grant
- OAuth 2.0 Implicit Grant



オープンAPIの提供にはAPI連携認証システムが不可欠

- OAuth 2.0 を使うには「認可サーバー」の機能が必要である。
- 認可（アクセスの許可）の前提としてユーザーが適切な方法で認証されている必要がある。

- オープンAPI提供者は「API連携認証システム」を構築する
  - ユーザーを認証する機能
  - OAuth 2.0 認可サーバーの機能

Identity Platform を構築することに他ならない

対応すべき OAuth 2.0 仕様の概略（最小限）

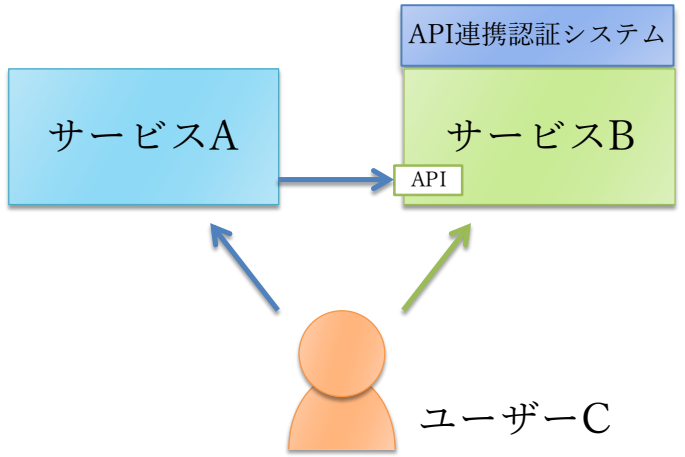
サービス開発者、API開発者も対応が必要

ロ オープンAPIの提供者（下図サービスB）

- API連携認証システム（Authorization Server）実装者
- API開発者

ロ オープンAPIの利用者（下図サービスA）

- アプリケーション開発者
  - フロントエンド開発者
  - バックエンド開発者



対応が必要となる OAuth の仕様類

#	仕様	AS 実装者	API 開発者	API利用者 アプリ開発者
①	The OAuth 2.0 Authorization Framework [RFC 6749]	○	—	○
②	Authorization Code Grant	○	—	○
③	Implicit Grant	○	—	○
④	Client Credentials Grant	○	—	○
⑤	The OAuth 2.0 Authorization Framework: Bearer Token Usage [RFC 6750]	—	○	○
⑥	OAuth 2.0 Token Revocation [RFC 7009]	○	○	○
⑦	Proof Key for Code Exchange by OAuth Public Clients [RFC 7636]	○	—	○
⑧	OAuth 2.0 Token Introspection [RFC 7662]	○	○	—
⑨	OAuth 2.0 for Native Apps [RFC 8252]	—	—	○

API利用時の構成パターン

ロ バックエンドあり（Confidential Client）

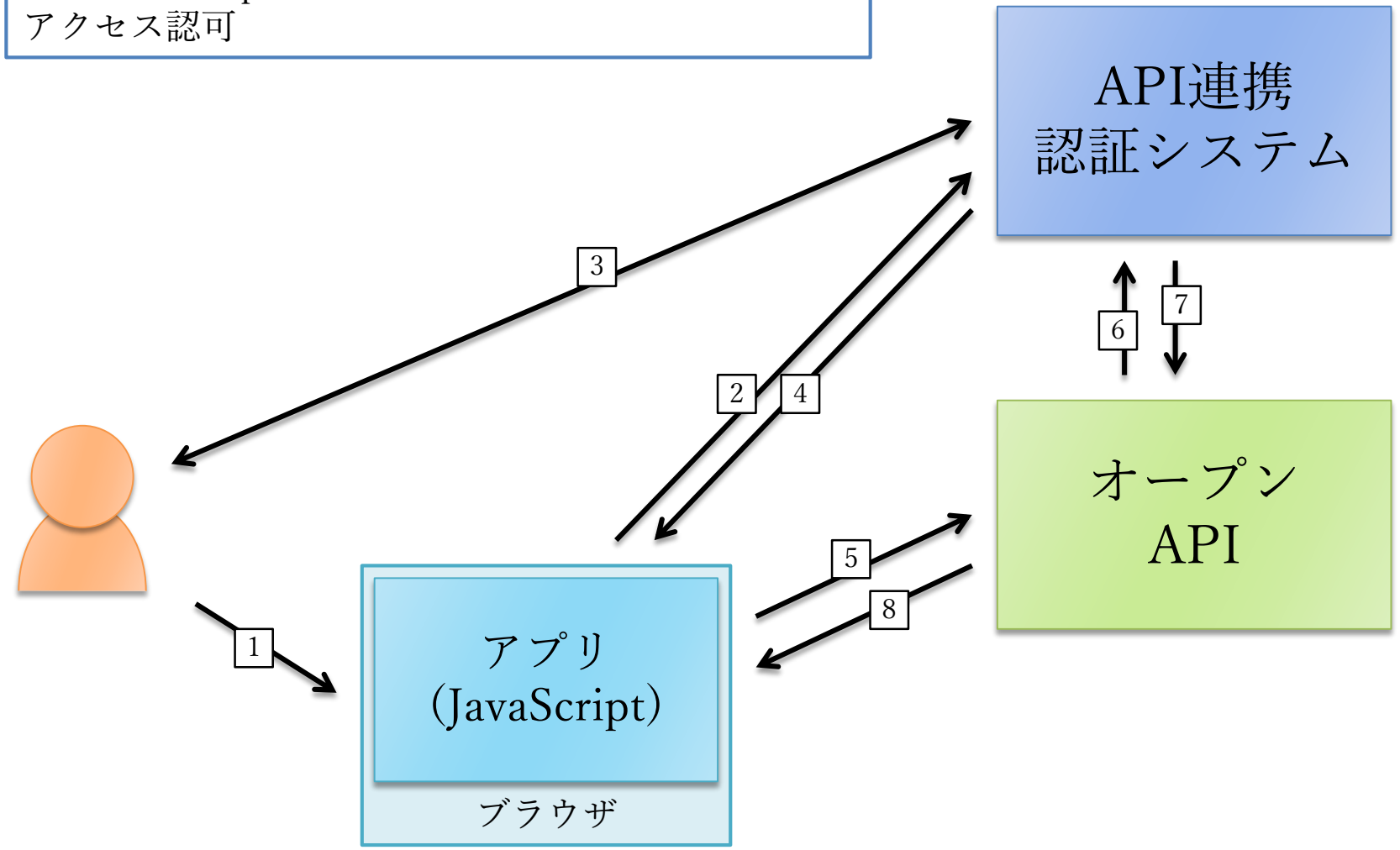
ロ バックエンドなし（Public Client）

アプリ形態別、対応すべき OAuth の方式

#	ユーザーへの提供形態	バックエンド	OAuthの方式	ポイント
①	**ブラウザで動作するJavaScriptアプリ** （モバイルウェブ、SPA、など）	あり	Authz Code Grant	バックエンドがオープンAPIへアクセス。バックグラウンドでのAPIアクセスのため、トークン自動更新が必要な場合も。
②		なし	Implicit Grant	JSアプリがオープンAPIへアクセス。トークン更新が必要な場合はブラウザを介して行なえる。
③	**ネイティブアプリ**	あり	Authz Code Grant	（① と同じ）
④		なし	Authz Code Grant w/ PKCE	ネイティブアプリがオープンAPIへアクセス。バックグラウンドでのAPIアクセスのため、トークン自動更新が必要な場合も。
⑤	**Webアプリ**	あり （Webアプリサーバー）	Authz Code Grant	WebアプリケーションサーバーがオープンAPIへアクセス。バックグラウンドでのAPIアクセスのため、トークン自動更新が必要な場合も。

JavaScriptアプリでのAPIアクセス認可

OAuth 2.0 Implicit Grant を用いたオープンAPIへのアクセス認可

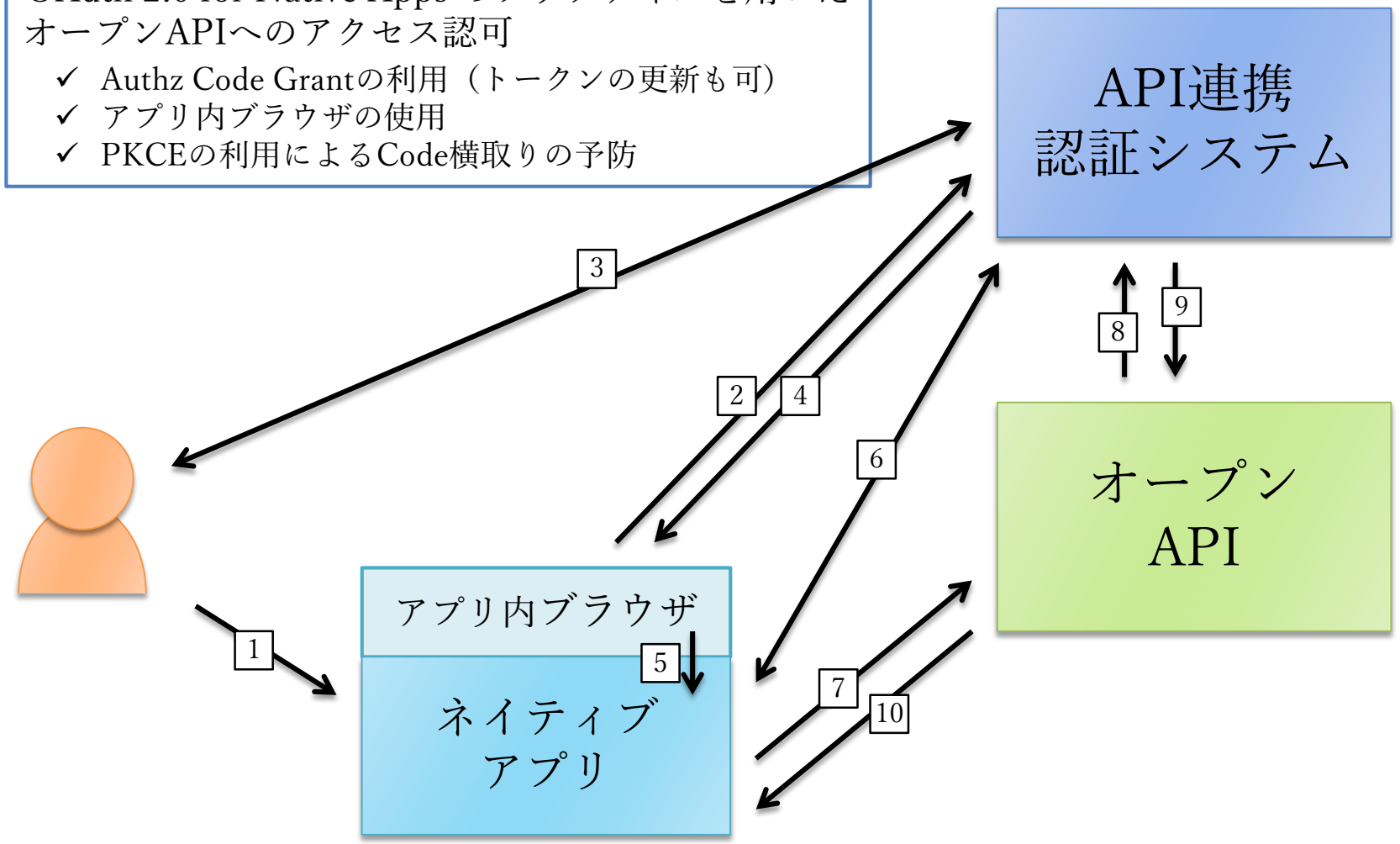


1. ブラウザでJSアプリを起動、ログイン開始
2. リダイレクトによりOAuth認可を要求
3. ユーザー認証を実行
4. オープンAPIアクセス用のアクセストークンを返却
5. アクセストークンをつけてオープンAPIにアクセス
6.~7. アクセストークンの情報を確認（ユーザー、Scope、有効期限、など）
8. オープンAPIが情報を使ってJSアプリが動作

ネイティブアプリでのAPIアクセス認可

OAuth 2.0 for Native Apps のプラクティスを用いたオープンAPIへのアクセス認可

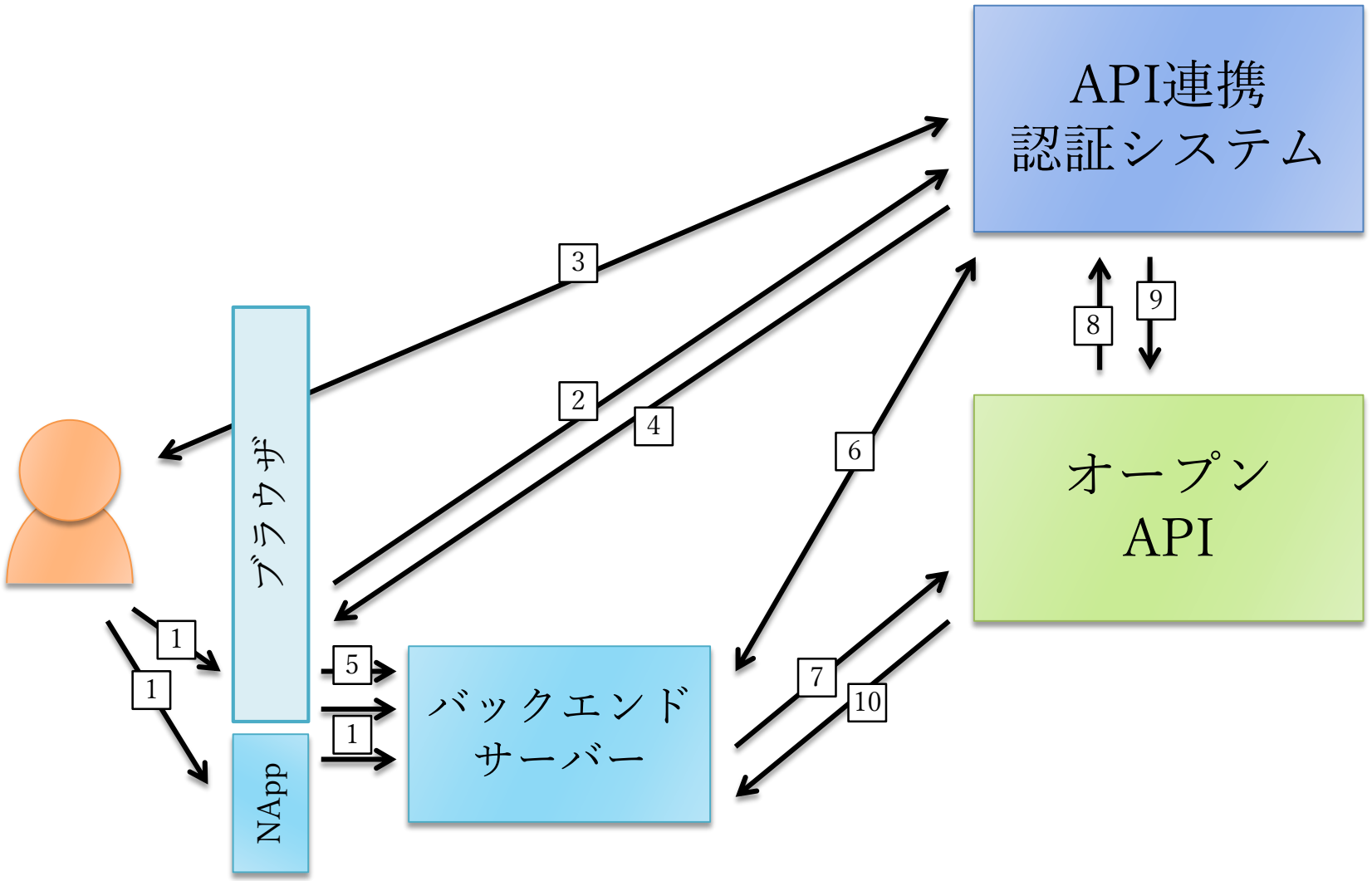
- ✓ Authz Code Grantの利用（トークンの更新も可）
- ✓ アプリ内ブラウザの使用
- ✓ PKCEの利用によるCode横取りの予防



1. ネイティブアプリを起動、ログイン開始
2. アプリ内ブラウザを使いOAuth認可を要求（PKCEを併用）
3. ユーザー認証を実行
4. アクセストークンを取得するためのCodeを返却
5. アプリ内ブラウザからネイティブアプリにCodeを返却
6. Codeをアクセストークンに交換（PKCEを併用）
7. アクセストークンをつけてオープンAPIにアクセス
8. ~9. アクセストークンの情報を確認（ユーザー、Scope、有効期限、など）
10. オープンAPIが情報を使ってネイティブアプリが動作

オープンAPIの振る舞いは、JSアプリ、ネイティブアプリで共通。

バックエンドサーバーを持つアプリでのAPIアクセス認可



1. アプリを起動、ログイン開始
2. バックエンドはネイティブアプリ（Napp）、ブラウザを通じてOAuth認可を要求
3. ユーザー認証を実行
4. アクセストークンを取得するためのCodeを返却
5. ブラウザから（NAppの場合はNAppを経由して）バックエンドサーバーにCodeを返却
6. Codeをアクセストークンに交換
7. アクセストークンをつけてオープンAPIにアクセス
8. ~9. アクセストークンの情報を確認（ユーザー、Scope、有効期限、など）
10. オープンAPIが情報を使ってバックエンドサーバー上のアプリが動作

オープンAPIの振る舞いは、JSアプリ、ネイティブアプリと共通。

今後対応が必要となってくる仕様
（OAuth 2.0, OpenID Connect 関連）

金融API向けにOAuth実装仕様の策定が進行中

- OpenID Foundation (Global) の Financial API WG が策定
  - https://openid.net/wg/fapi/
- OAuthの実装仕様について、以下の2つがすでに Implementer's Draft となり、公開されている。
  - Part 1: Read Only API Security Profile (Implementer's Draft).
  - Part 2: Read & Write API Security Profile (Implementer's Draft).
- 以下の仕様についても検討が進行中。
  - Client Initiated Backchannel Authentication Profile.

Part 1: Read Only API Security Profile ①

#	Authorization Server への要求事項	特に注目したい箇所
1	コンフィデンシャルクライアントをサポートすることが望ましい。(SHOULD)	—
2	パブリッククライアントをサポートすることが望ましい。(SHOULD)	—
3	（署名や暗号化に）対象鍵を使う場合は [OIDC] の 16.19節 の要件に準拠した client_secret を提供する。(SHALL)	—
4	トークンエンドポイントでは以下のいずれかの方法を用いてコンフィデンシャルクライアントを認証する。(SHALL) 1. TLS相互認証 [MTLS] 2. [OIDC] の 9章 で指定されている client_secret もしくは秘密鍵を用いた JWS クライアントアサーション	□
5	クライアント認証にRSA暗号アルゴリズムを用いる場合は 2048ビット 以上の長さの鍵を用いる。(SHALL)	—
6	クライアント認証に楕円曲線暗号アルゴリズムを用いる場合は 160ビット 以上の長さの鍵を用いる。(SHALL)	—
7	パブリッククライアントをサポートする場合は、[RFC7636] の S256コードチャレンジ方式 をサポートする。(SHALL)	—
8	リダイレクトURIは事前登録する。(SHALL)	—
9	認可要求のパラメータに redirect_uri を必要とする。(SHALL)	—
10	（認可要求の）redirect_uri の値は、事前に登録されたリダイレクトURIの一つに完全一致しなければならない。(SHALL)	—
11	[X.1254] に定義された LoA 2 以上のユーザー認証を要求する。(SHALL)	□
12	以前に承認されていない scope の要求を承認する場合には、ユーザーの明示的な同意を必要する。(SHALL)	—

- OAuth と OpenID Connect を併用（OIDCの仕様に基づきOAuthの認可要求を行なう）
- Confidential Client の認証をより強度が高い方式の利用が指定される
  - 事実上の OAuth 2.0 の追加仕様 RFC 7521, RFC 7523 への対応
- アイデンティティ情報更新時の再認証など信頼性を確保する仕組みの導入

Part 1: Read Only API Security Profile ②

#	Authorization Server への要求事項	特に注目したい箇所
13	可能であれば、認可コード（[RFC6749] 1.3.1節）が以前に使用されてないことを検証する。(SHALL)	—
14	[RFC6749] 4.1.4節に沿ったトークンレスポンスを返す。(SHALL)	—
15	発行されたアクセストークンとともに許可されたスコープのリストを返す。(SHALL)	□
16	[RFC6819] 5.1.4.2.2節に定義されているように最少128ビットの不透明な推測できないアクセストークンを発行する。(SHALL)	—
17	[OIDC] 16.18節にあるとおり認可中に長期間の許可を与えようとしていることユーザーに明示することが望ましい。(SHOULD)	□
18	[OIDC] 16.18節にあるとおりエンドユーザーがクライアントに付与されたアクセストークンやリフレッシュトークンを失効する仕組みを提供することが望ましい。(SHOULD)	□
19	[OIDC] 3.1.2.1節にあるとおりの認証要求をサポートする。(SHALL)	—
20	[OIDC] 3.1.2.2節にあるとおりの認証要求の検証を行なう。(SHALL)	—
21	[OIDC] 3.1.2.2節および3.1.2.3節にあるとおりユーザーを認証する。(SHALL)	—
22	[OIDC] 3.1.2.4節および3.1.2.5節にあるとおり認証結果に応じて認証応答を行なう。(SHALL)	—
23	[OIDC] 3.1.3.2節にあるとおりトークン要求を検証する。(SHALL)	—
24	要求された scope に openid が含まれている場合は、[OIDC] 3.1.3.3節にあるとおり、トークン応答の中で認証されたユーザーに対応するsub値を持つ（オプションでacr値も含めた）IDトークンを発行して応答する。(SHALL)	—

- OAuth, OpenID Connect を併用（OIDCの仕様に基づきOAuthの認可要求を行なう）
- 長時間有効な認可（Access Token, Refresh Token発行）には明示的な同意が必要に。
- 発行された Access Token の scope を明示的に応答することが必要に。

Part 2: Read & Write API Security Profile

#	Authorization Server への要求事項	特に注目したい箇所
1	[OIDC] 6章にあるとおり、request あるいは request_uri パラメータにJWS署名のJWTを使用する。(SHALL)	□
2	response_type の値には、code id_token もしくは code id_token token を使用する。(SHALL)	□
3	認可リクエストに対するレスポンスの分離署名として IDトークンを応答する。(SHALL)	—
4	state値を保護するため、IDトークンにはstateのハッシュ値 s_hash を含める。(SHALL)	□
5	書き込み操作の用途の場合は、記名式 (Holder of Key) の認可コード、アクセストークン、リフレッシュトークンを発行する。(SHALL)	□
6	記名式 (Holder of Key) の仕組みには [OAUTB] あるいは [MTLS] を用いる。(SHALL)	□
7	[X.1254] に定義された LoA 3 以上のユーザー認証を要求する。(SHALL)	□
8	署名付きのIDトークンをサポートする。(SHALL)	—
9	署名付きかつ暗号化されたIDトークンをサポートすることが望ましい。(SHOULD)	□

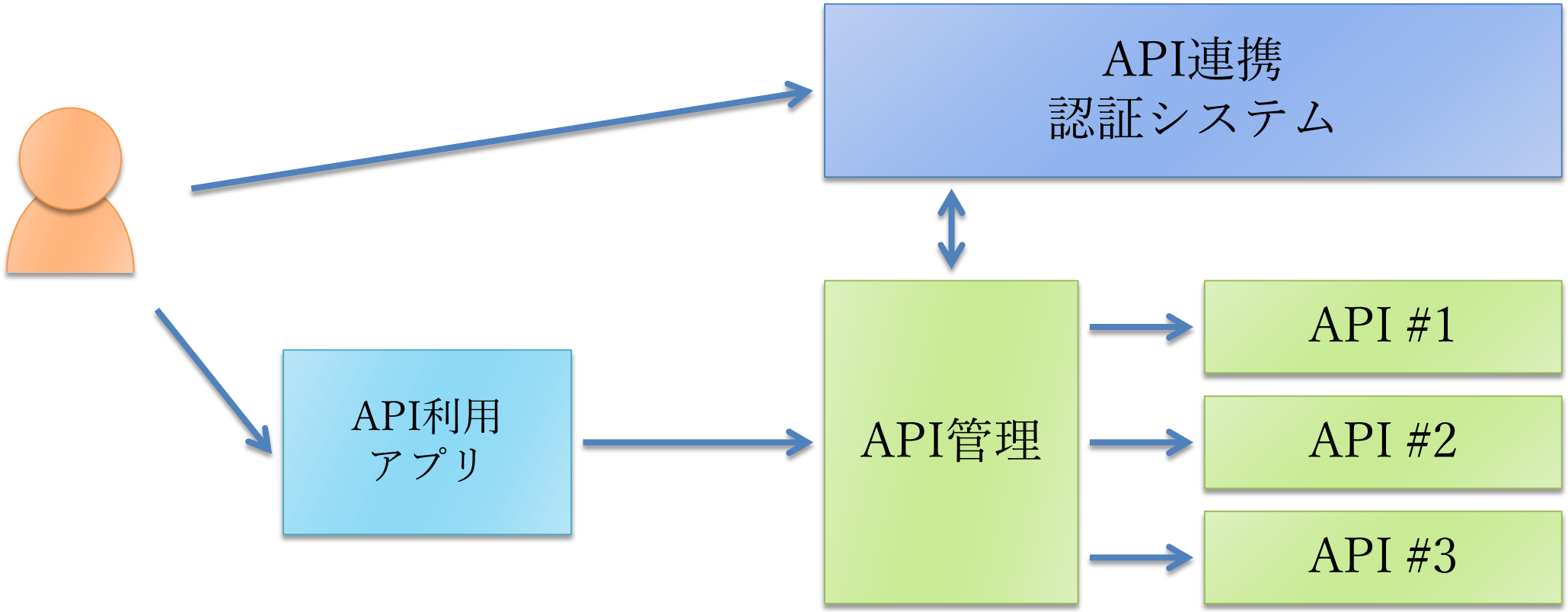
- OpenID Connect のより深いレベルの仕様の実装
  - JWTリクエストパラメーター、JWSとJWEによる署名と暗号化されたIDトークン
- IDトークンを使った認可レスポンスへの署名 (at_hash, c_hash, s_hashへの対応)
- 記名式トークン、記名式認可コード発行への対応
- OAuth の策定中仕様 (OAUTB, MTLS) の実装
- 多要素認証への対応

現時点で対応を検討しておくべき仕様

#	仕様	AS 実装者	API 開発者	API利用者 アプリ開発者
①	OpenID Connect Core 1.0	○	—	○
②	[OAUTB], [MTLS] を使った記名式トークン、認可コード発行への対応	○	○	○
③	署名、暗号化されたIDトークンのサポート [Sec.2][Sec.16.14]	○	○	○
④	長期間有効なトークン発行時の適切な同意画面の表示	○	—	—
⑤	OAuth 2.0 トークン発行時の scope クレームの応答への対応	○	—	○
⑥	Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants [RFC 7521]	○	—	○
⑦	JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants [RFC 7523]	○	—	○
⑧	OAuth 2.0 Token Binding [OAUTB] [draft-ietf-oauth-token-binding-05]	○	○	○
⑨	OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens [MTLS] [draft-ietf-oauth-mtls-07]	○	○	○

API管理のソリューション の位置づけはより重要に

- 策定される新しい仕様への追従を個々の API で進めるのは困難
- API管理ソリューションを使った集中管理が必要になっていく



金融APIでなければ、金融向けのグレードである必要はない

- FAPI の Profile は金融API向け
- APIが提供する機能、情報のリスクを評価してどこまで対応するかを決める
- 対応策は、FAPI Profile が参考になる

R/O Prof. LoA 2

R/W Prof. LoA 3

- 不便, 苦痛, または社会的地位やレピュテーションの毀損
- 経済的損失または機関の負債
- 組織や組織のプログラム、公共の利益への損害
- 許可されていないセンシティブ情報の公開
- 個人の安全
- 民事または刑事上の違反

Table 6-2 – Potential impact at each level of assurance

Possible consequences of authentication failure	Potential impact of authentication failure by LoA			
	1	2	3	4
Inconvenience, distress or damage to standing or reputation	Min*	Mod	Sub	High
Financial loss or agency liability	Min	Mod	Sub	High
Harm to the organization, its programs or public interests	N/A	Min	Mod	High
Unauthorized release of sensitive information	N/A	Mod	Sub	High
Personal safety	N/A	N/A	Min Mod	Sub High
Civil or criminal violations	N/A	Min	Sub	High

* Min=Minimum; Mod=Moderate; Sub=Substantial; High=High.

引用元: ITU-T Recommendation X.1254 Entity authentication assurance framework, p7 (2012年9月)

当社ソリューションのご紹介

統合認証ソリューション ThemiStruct を提供しています

ThemiStruct-WAM　シングルサインオン 認証基盤ソリューション

ThemiStruct-IDM　ID管理ソリューション

ThemiStruct-CM　電子証明書発行・管理 ソリューション

ワンタイムパスワードソリューション
ThemiStruct-OTP

システム監視ソリューション
ThemiStruct-MONITOR

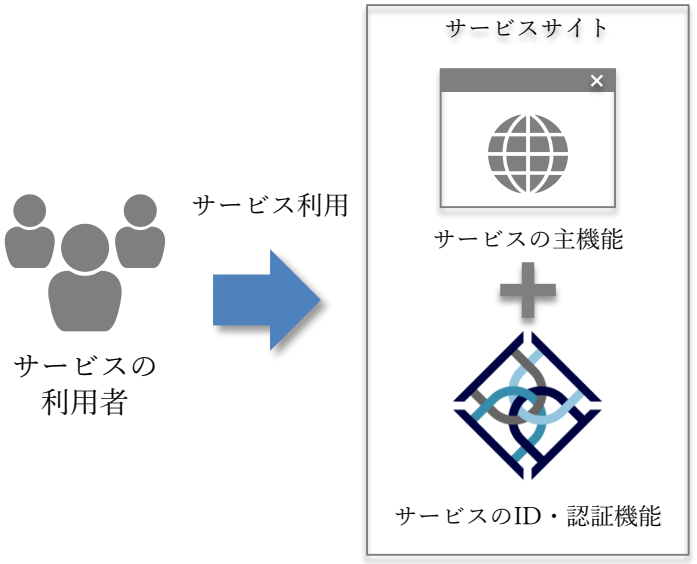
ThemiStruct（テミストラクト）
Identity Platform AWS対応版

APIエコノミー時代の 統合認証パッケージ

統合認証パッケージ ThemiStruct Identity Platform

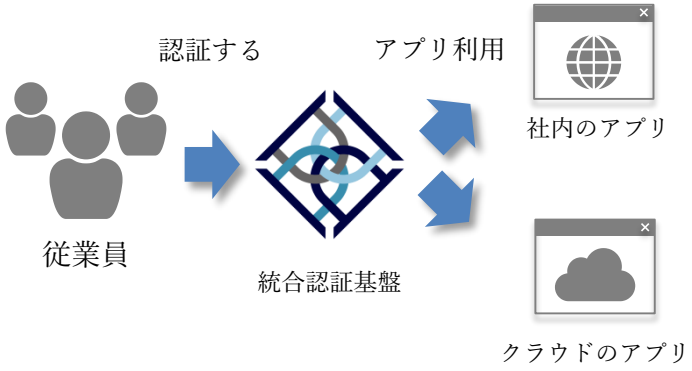
ThemiStruct Identity Platform は、ITシステム利用者のアイデンティティ管理と認証の機能を提供します。顧客向けにサービスを展開したり、従業員向けにアプリケーションを展開する際に必要となる、共通ID基盤の構築に活用いただけます。

顧客向けサイト領域に活用



サービスサイトの
ID、認証の共通機能として利用

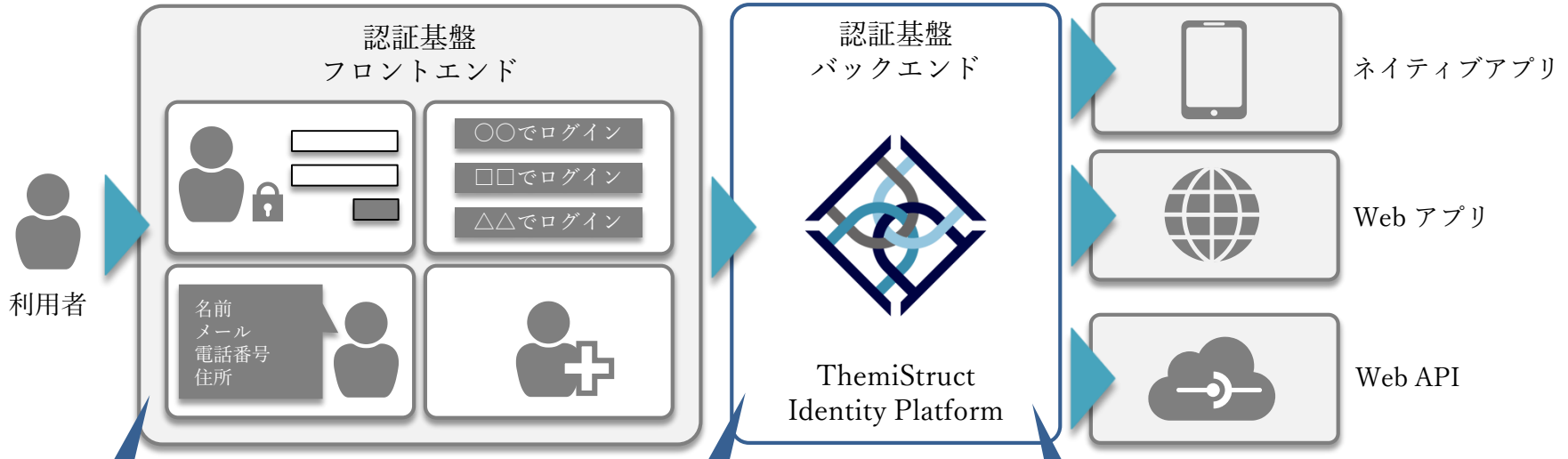
エンタープライズ領域に活用



エンタープライズアプリ群の
SSOやアクセス制御の基盤として利用

ThemiStruct Identity Platform を『顧客向けサイト』に活用

ThemiStruct Identity Platform を『顧客向けサイト』環境に適用した場合、サービスの利用者IDの管理と認証の機能を担うバックエンドサービスとして稼働します。これらの機能のUIにあたるアプリケーションの実装を支援し、実際のサービスアプリと接続するためのインタフェースを提供します。



IDと認証、アクセス認可の機能をアプリに組み込む

利用者のID情報の管理や認証などの共通機能を、サイトの統一されたデザインで組み込むことができます。

Web APIの認証・アクセス認可に対応

OAuth 2.0 を使ったAPI連携認証システムが実現でき、モバイルアプリ、オープンAPIを安全に公開できます。

短期間での導入が可能

スモールスタート、突発的アクセス集中に対応できる伸縮可能かつ事業継続性を考慮した認証基盤が、2時間でセットアップできます。

『顧客向けサイト』に向けた3大特長

ロ IDとユーザー認証、アクセス認可の機能をアプリに組み込む

- 利用者のID情報の管理や認証など利用者IDに関連する共通機能の実装を支援する『フレームワーク』と『Web API』を提供します。これらを活用し、貴社のサービスサイトに認証機能やパスワード変更機能、アプリケーションへのID連携機能などを組み込むことができます。

ロ Web APIの認証・アクセス認可に対応

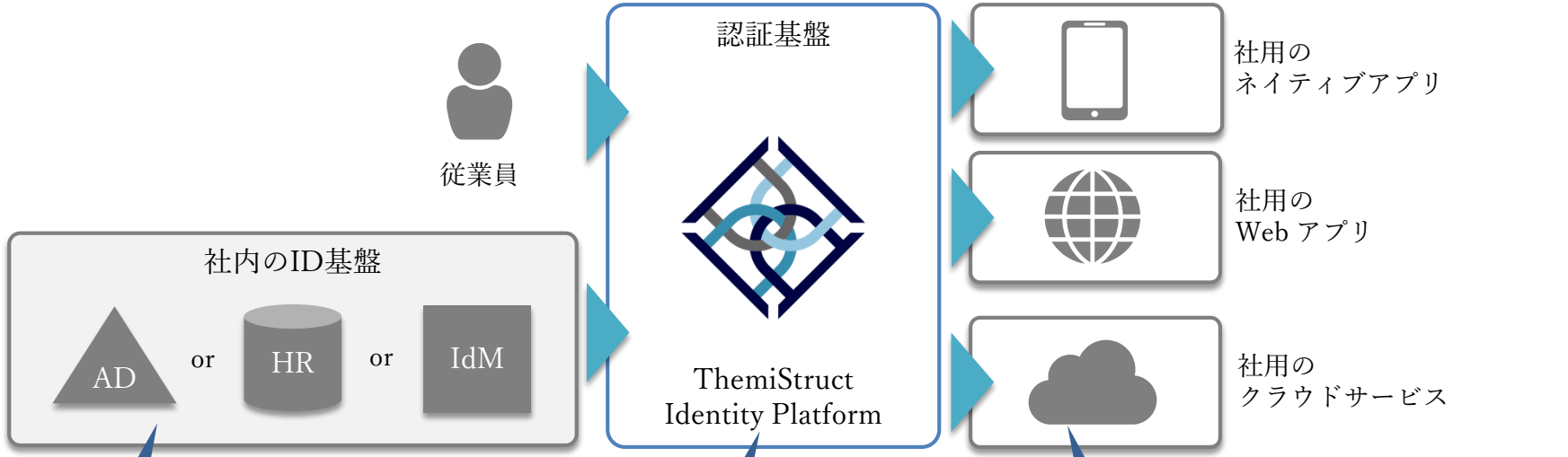
- OAuth 2.0 の技術仕様に基づいた認証・アクセス認可機能を提供します。OAuth 2.0 を使ったAPI連携認証システムが実現でき、モバイルアプリ、オープンAPIを安全に公開できます。

ロ 短期間での導入が可能

- AWSマネージドサービスのコンポーネントを活用し、導入時におけるキャパシティやアベイラビリティプラニングから解放します。スモールスタート、突発的アクセス集中に対応できる伸縮可能かつ事業継続性を考慮した認証基盤が、2時間でセットアップできます。

ThemiStruct Identity Platform を『エンタープライズ』に活用

ThemiStruct Identity Platform を『エンタープライズ』環境に適用した場合、社内外のアプリケーションにシングルサインオン可能な認証基盤を提供できます。また、企業のポリシーにあった認証機能の設定や既存のIDとの統合をすることができます。



既存のID基盤と統合

社内で既に展開されたIDに関連する仕組みとの統合を実現できます。

企業ポリシーの適用

多要素、多段階認証に対応し多様なセキュリティ要件に応じた認証機能を提供します。

シングルサインオン

社内、社外問わず、様々なアプリケーションにシングルサインオンできます。

『エンタープライズ』に向けた3大特長

ロ　シングルサインオン

- OpenID Connect などの標準技術仕様を用いたシングルサインオンに対応しています。社内、社外問わず、様々なアプリケーションにシングルサインオンできます。

ロ　企業ポリシーの適用

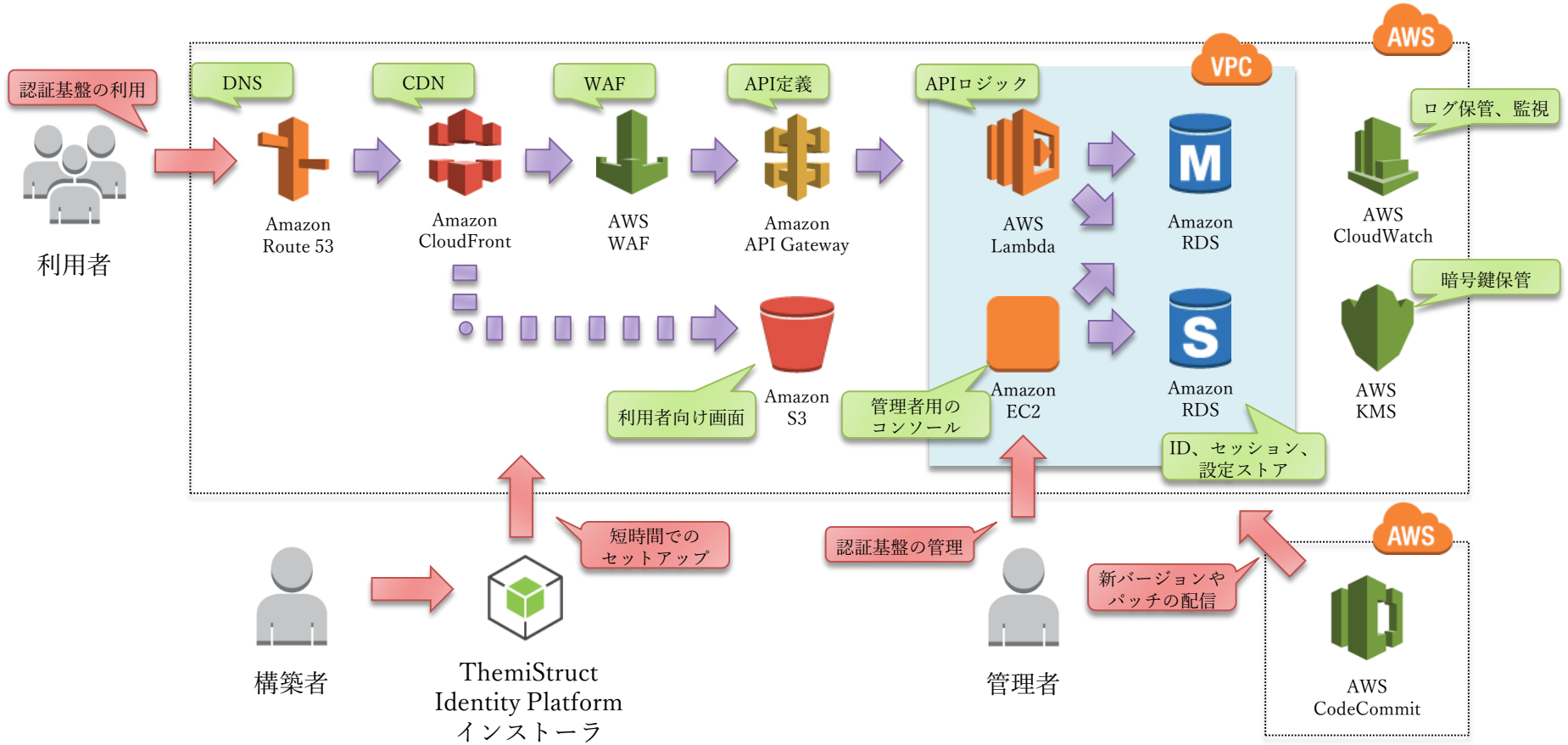
- ユーザーの属性や状態、利用するアプリケーションに応じて、柔軟な認証ポリシーをデザインすることができます。例えば、社内ネットワークからのアクセスの場合、IDとパスワードの認証を提供し、外出先からのアクセスの場合、追加でワンタイムパスワード認証を提供するといったポリシーを展開することができます。

ロ　既存のID基盤と統合

- 既存のID基盤と統合に向けたアカウント管理APIを提供しています。これにより、社内で既に展開されたIDに関連する仕組みとの統合を実現できます。

サーバーレスアーキテクチャを採用

ThemiStruct Identity Platform は Amazon Web Services のマネージドサービス上で稼働するため、一定のアベイラビリティ、スケーラビリティを実現することができます。また、以下の構成のセットアップを約2時間で完了できるインストーラを提供しています。



オージス総研のAPIソリューション

- オージス総研のクラウドとAPI開発・運用の知見を集めたサービス
  - FintechやIoTなどのデジテルビジネスに必要不可欠なAPI公開を実現
  - API導入のための技術調査・検討からAPIの公開・運用までフルカバー

■API導入コンサルティングサービス

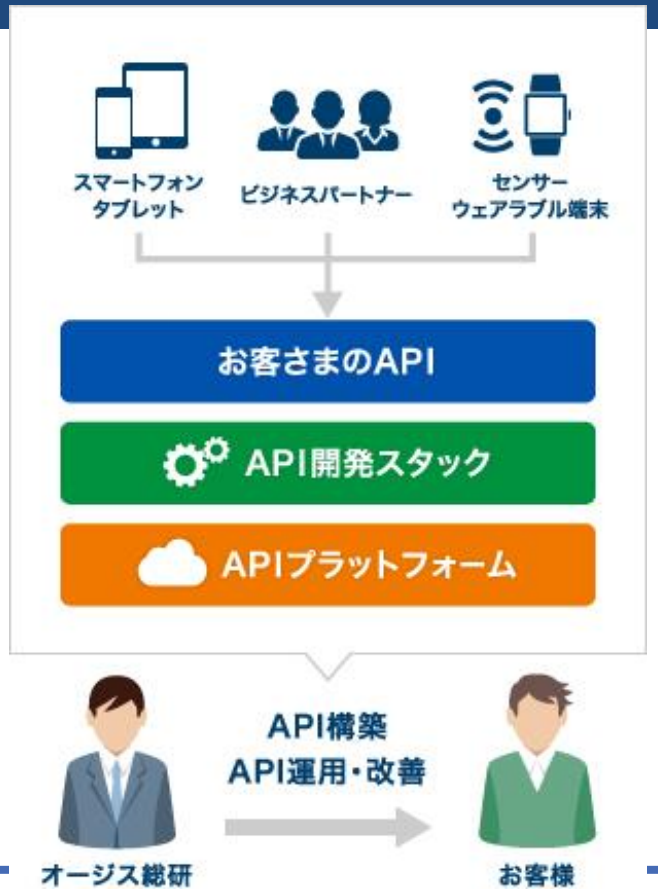
✓ ロードマップの策定援や、初期システムアーキテクチャ策定、API導入PoC等によりAPI導入を支援

■API構築サービス

✓ ベストプラクティスを組み込んだAPI開発スタックを使い、お客様の環境に合ったAPIプラットフォーム、APIを迅速かつ効率的に開発

■API運用サービス

✓ 継続的なAPIの改善・バージョンアップ
✓ 安定稼働のためのAPIプラットフォームの監視・障害対応
✓ API利用者向けサポートサービス



まとめ

まとめ

- オープンAPIを提供するためには、API連携認証システムの実現が必要となっている。
- API連携認証システムは、ユーザー認証とアクセス認可の機能を必要とし、Identity Platform を構築することに他ならない。
- OpenID Connect, SAMLといったID連携の標準技術に加え、OAuth 2.0 の標準仕様群に継続的に対応していく必要がある。
- 当社では ThemiStruct Identity Platform を開発・提供し、API連携認証システム の構築ニーズに対応。

AWS SUMMIT TOKYOに出展します

【セッション】

5月31日(木)　15:00～15:40　パミール1F 暁光

「統合認証基盤をサーバーレスアーキテクチャで構築する狙い、メリット、考慮点」

事業開発本部 テミストラクトソリューション部 マネジャー 上席アーキテクト　八幡 孝

【展示】

パミール1F 北辰　ブースNO.**G16**

・OAuth2.0に対応した統合認証プラットフォーム「ThemiStruct(テミストラクト)Identity Platform」
・API公開・管理基盤の導入、コンサルティング「API公開支援ソリューション」
・デバイス/センサー、ネットワークからクラウドまでパックでご提供「IoTかんたんパック」

ご清聴ありがとうございました

【お問い合わせ先】
株式会社オージス総研
TEL: 03-6712-1201 / 06-6871-8054
mail: info@ogis-ri.co.jp