ThemiStruct
テミストラクト

利用者の意思に基づく API連携、データー・機能の利活用

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）
IdentityPlatform

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。
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。

現在の認証基盤の主要なユースケース

企業/企業グループ内の業務向けの「**社内統合認証基盤**」を構築する

顧客向けサービスサイトの「**共通ID基盤**」を構築する

オープンAPIを提供するための「**API連携認証システム**」を構築する

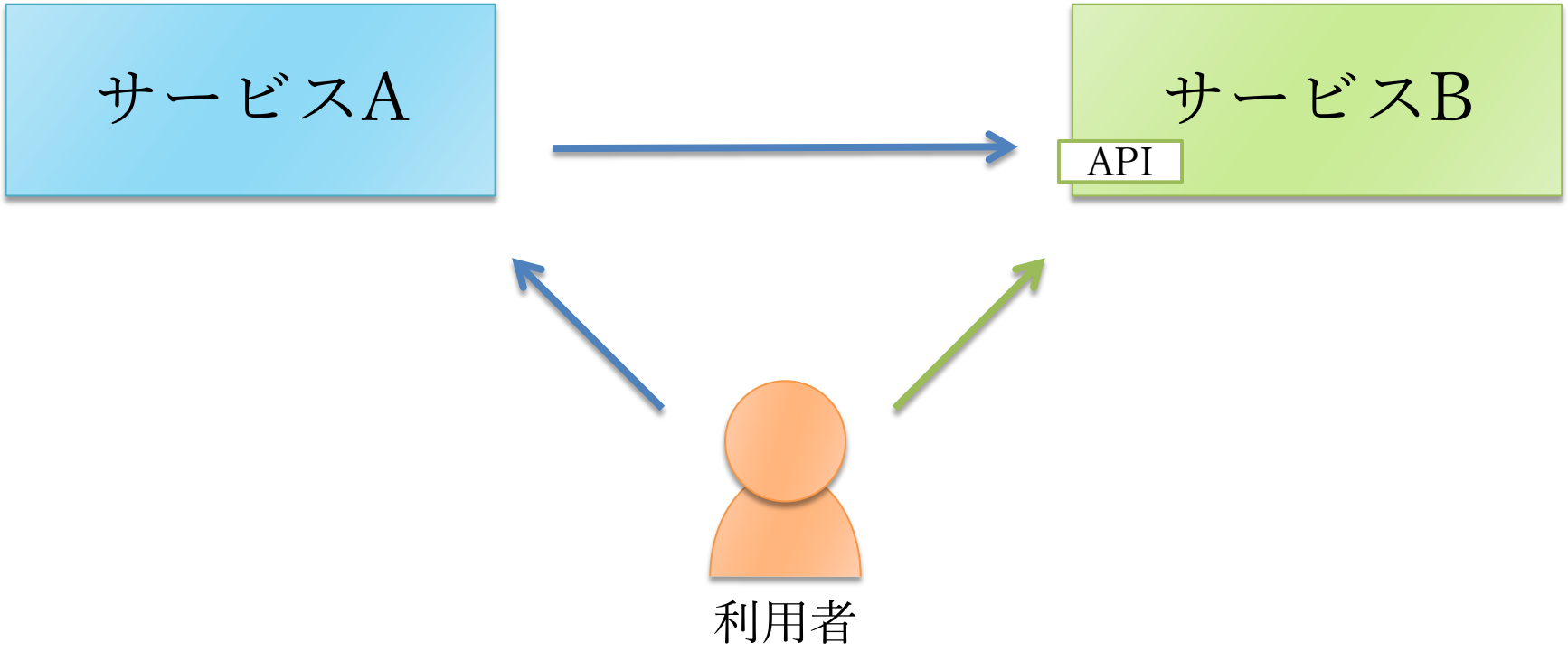
利用者の意思に基づき
データー・機能を提供する

X-Tech による新しい価値創出が進む

ホテル・旅行代理店	×	配車サービス	=	• ホテル予約と同時に配車手配 • 観光地に通じた運転手を指名
会計サービス	×	銀行	=	• 口座情報参照 • 振り込み
自動車	×	損害保険	=	• テレマティクス保険 • 事故時の迅速なフォロー

利用者 と サービス と サービス間のAPI連携

利用者のデーター、利用者が使用する機能を、
APIを使って、サービス間で連携させることが不可欠

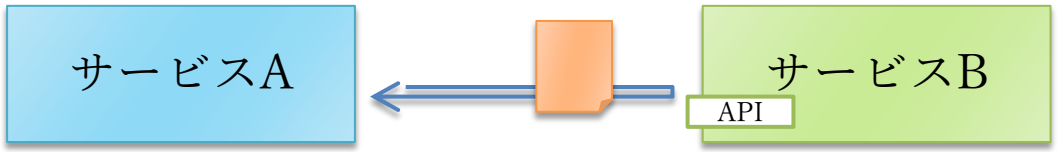


サービスとサービスを連携するパターン

❶

AとBの処理・手続きを一連の流れで提供する。

❷



利用者のBのデーターを提供してもらうことでAが付加価値サービスを提供する。

❸

利用者のBのデーターを継続的に提供してもらうことでAが付加価値サービスを提供する。

❹

Aは利用者の代わりにB上で処理を（自動的に）行なう。

サービスを選ぶときの利用者の心理

便利そう!

お得そう!

みんな使ってるから自分も!

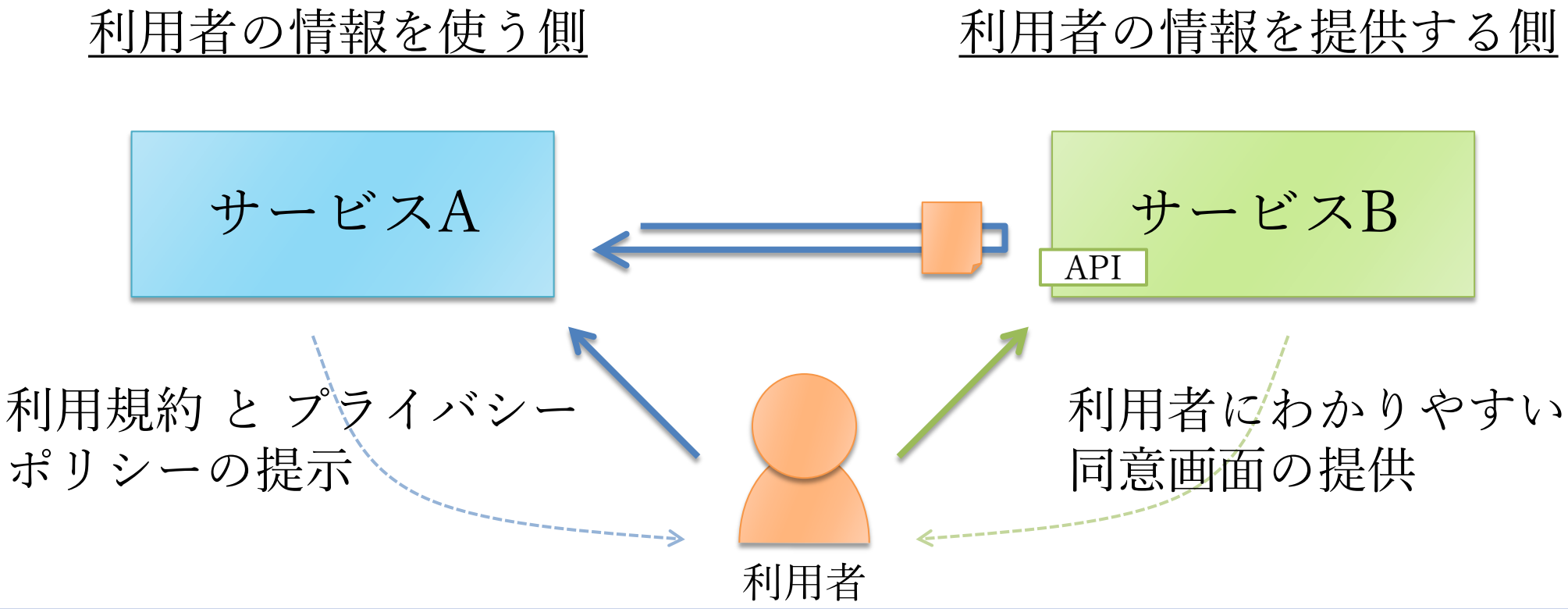
何の情報が渡される？

その情報は何に使うの？

情報は安全に管理される？

情報の使用と管理について適切に利用者に伝える

- 利用規約 と プライバシーポリシー
- 情報取得時点での利用者への提示と確認（同意）



利用者に何を伝えるか

ロ 法令・ガイドライン、業界団体等での検討結果、などを踏まえ、デザインしていく。

（例）「オープン API のあり方に関する検討会報告書」
3.4 利用者保護原則　3.4.2 説明・表示、同意取得

c 銀行は、トークン発行に当たって、少なくとも以下の点について、わかりやすく画面表示のうえ、利用者の同意を求めることが必要である。

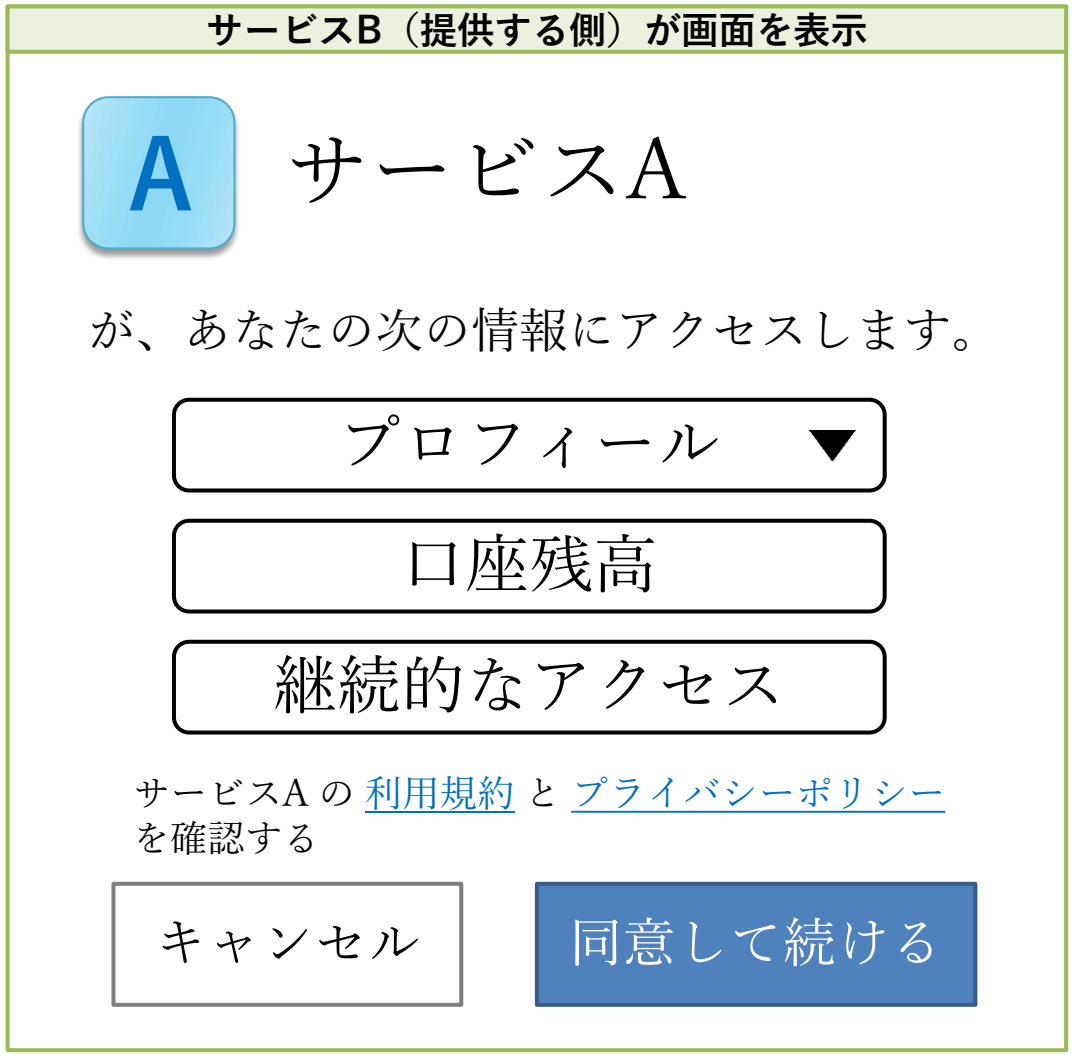
- アクセス権限を付与するAPI接続先の名称
- API 連携するサービス等の名称
- 付与する権限の内容・範囲
- 付与する権限の有効期限
- 付与した権限の削除、解除方法
- その他注意喚起が必要な事項

d API 接続先は、サービス提供に当たって、少なくとも以下の点について、わかりやすく画面表示のうえ、利用者の同意を求めることが必要である。

- 個人情報保護法にもとづく取得情報の利用目的、共有範囲（第三者提供の有無）
- 取得した情報の削除に関する事項
- サービス利用上の制限
- その他注意喚起が必要な事項

引用元: https://www.zenginkyo.or.jp/fileadmin/res/abstract/council/openapi/openapi_report_1.pdf

同意画面の構成例



どのサービスに情報を提供するか が視覚的に確認できる。

何の情報を提供するかを、利用者が理解しやすい表現で提示する。

1回だけの提供か、継続的な提供かを確認できる。

サービスA の利用規約とプライバシーポリシーへの導線を作り、利用者が確認しやすい環境を提供する。

情報提供を拒否することができる。

同意画面の構成例

APIを提供するサービスサイトの特性・デザインに合わせてUIを作る必要がある。

どのサービスに情報を提供するか
が視覚的に確認できる。

何の情報を提供するかを、利用者が
理解しやすい表現で提示する。

1回だけの提供か、継続的な提供か
を確認できる。

サービスA の利用規約とプライバシーポリシーへの
導線を作り、利用者が確認しやすい環境を提供する。

情報提供を拒否することができる。

情報アクセス前に 使用目的、管理方法 を利用者に提示

サービスA（使う側）が画面を表示

○○機能を提供するために、サービスBのあなたの
「プロフィール情報」
「口座残高」
情報を使用します。

これらの情報に継続的にアクセスします。

利用規約 と プライバシーポリシー を確認する

キャンセル　　続ける

サービスB（提供する側）が画面を表示

A サービスA

が、あなたの次の情報を要求しています。

プロフィール ▼
口座残高
継続的なアクセス

サービスA の 利用規約 と プライバシーポリシー を確認する

キャンセル　　同意して続ける

同意をいつ取得するか？

□ 最初に一度だけ取得する

- ➢ 条件が変わらない限りは同意画面は出ない。
- ➢ 確認せずに同意・処理を継続するクセを利用者につけない。

□ 情報・機能にアクセスするごとに取得する

- ➢ 重要な情報・機能にアクセスするときに利用者に確認する。
  - ● 同意画面としてではなく、利用する側のサービス（サービスA）側で確認画面を提供する方が良いだろう。

Web API を活用する際の API連携認証システムの必要性

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

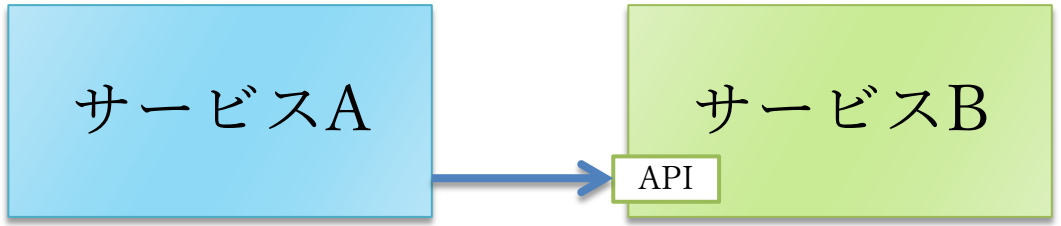
ロ オープンAPI

- 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

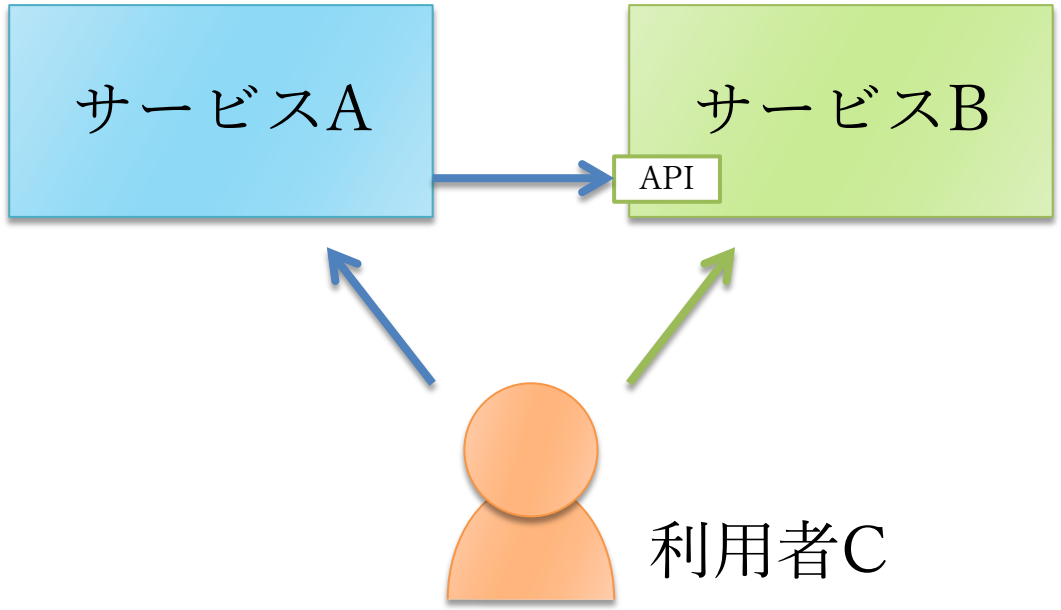
2者間でのAPI利用

- 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

- 利用者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、利用者Cの意図により、利用者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 利用者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。**

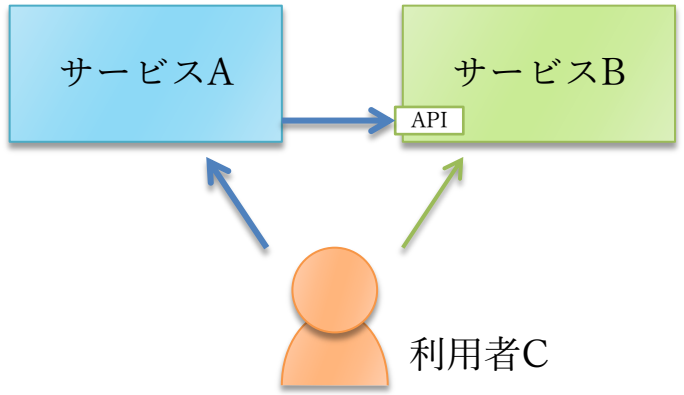
3者間でのAPI利用時の認証

ロ 利用者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。

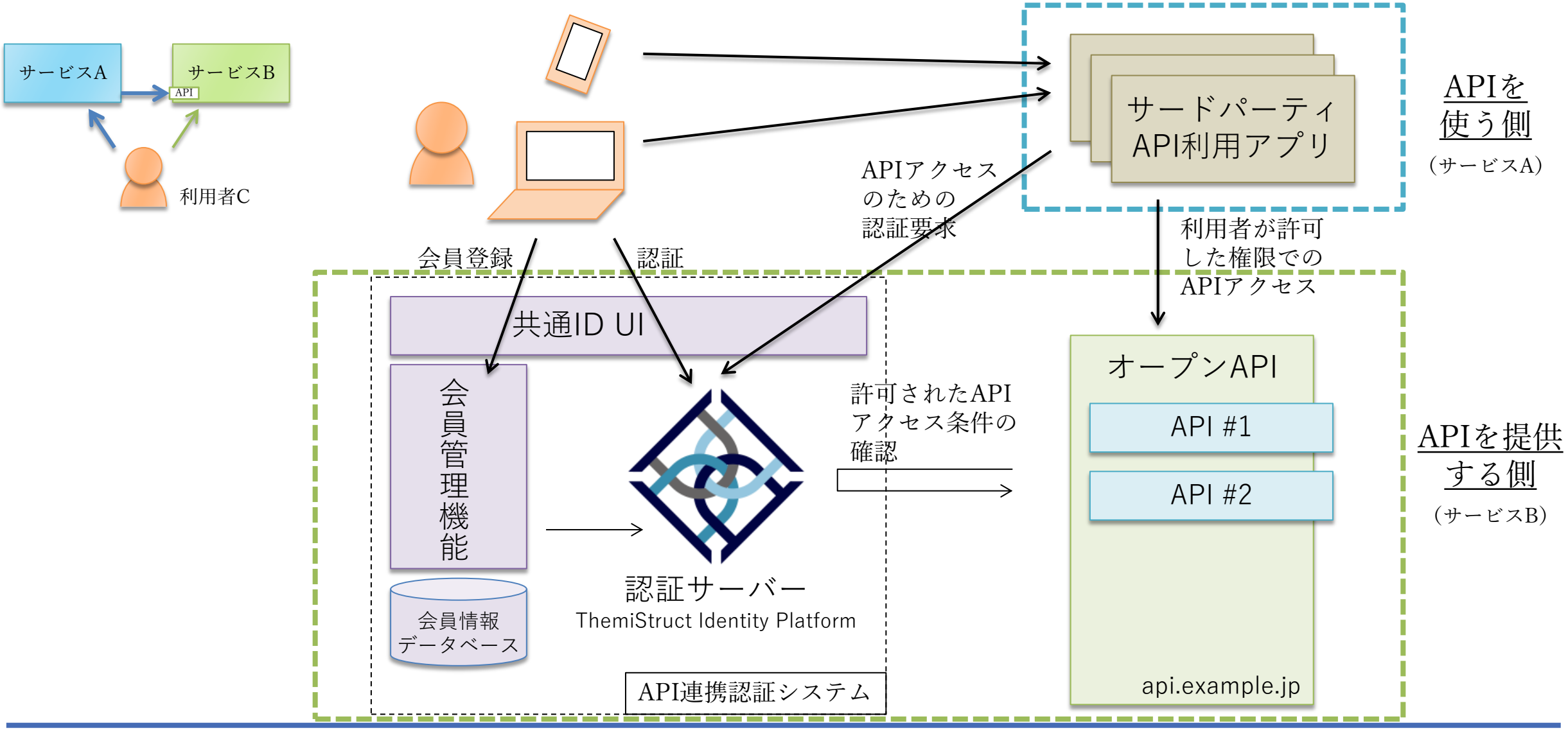
- ユーザーIDとパスワードなど、クレデンシャル情報を渡す方式では、全権限をサービスAに与えてしまう。

ロ 現時点では OAuth 2.0 の利用が唯一の選択肢

- 利用者C の同意に基づき、利用者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



「API連携認証システム」の構築イメージ



対応すべき OAuth 2.0 仕様の概略（最小限）

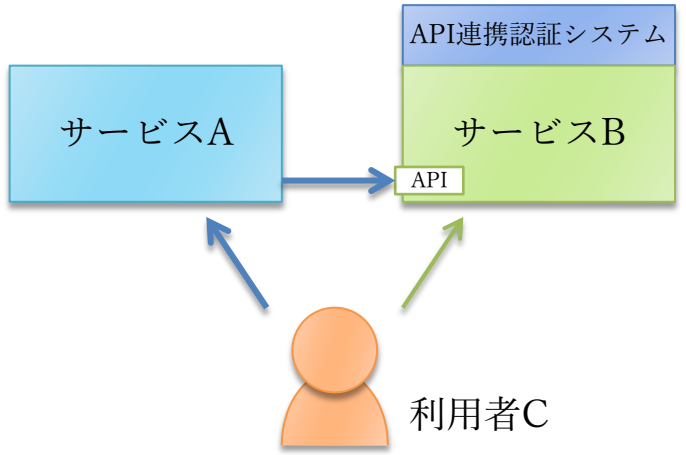
サービス開発者、API開発者も対応が必要

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]	○	—	○
.1	Authorization Code Grant	○	—	○
.2	Implicit Grant	○	—	○
.3	Client Credentials Grant	○	—	○
②	The OAuth 2.0 Authorization Framework: Bearer Token Usage [RFC 6750]	—	○	○
③	OAuth 2.0 Token Revocation [RFC 7009]	○	○	○
④	Proof Key for Code Exchange by OAuth Public Clients [RFC 7636]	○	—	○
⑤	OAuth 2.0 Token Introspection [RFC 7662]	○	○	—
⑥	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）

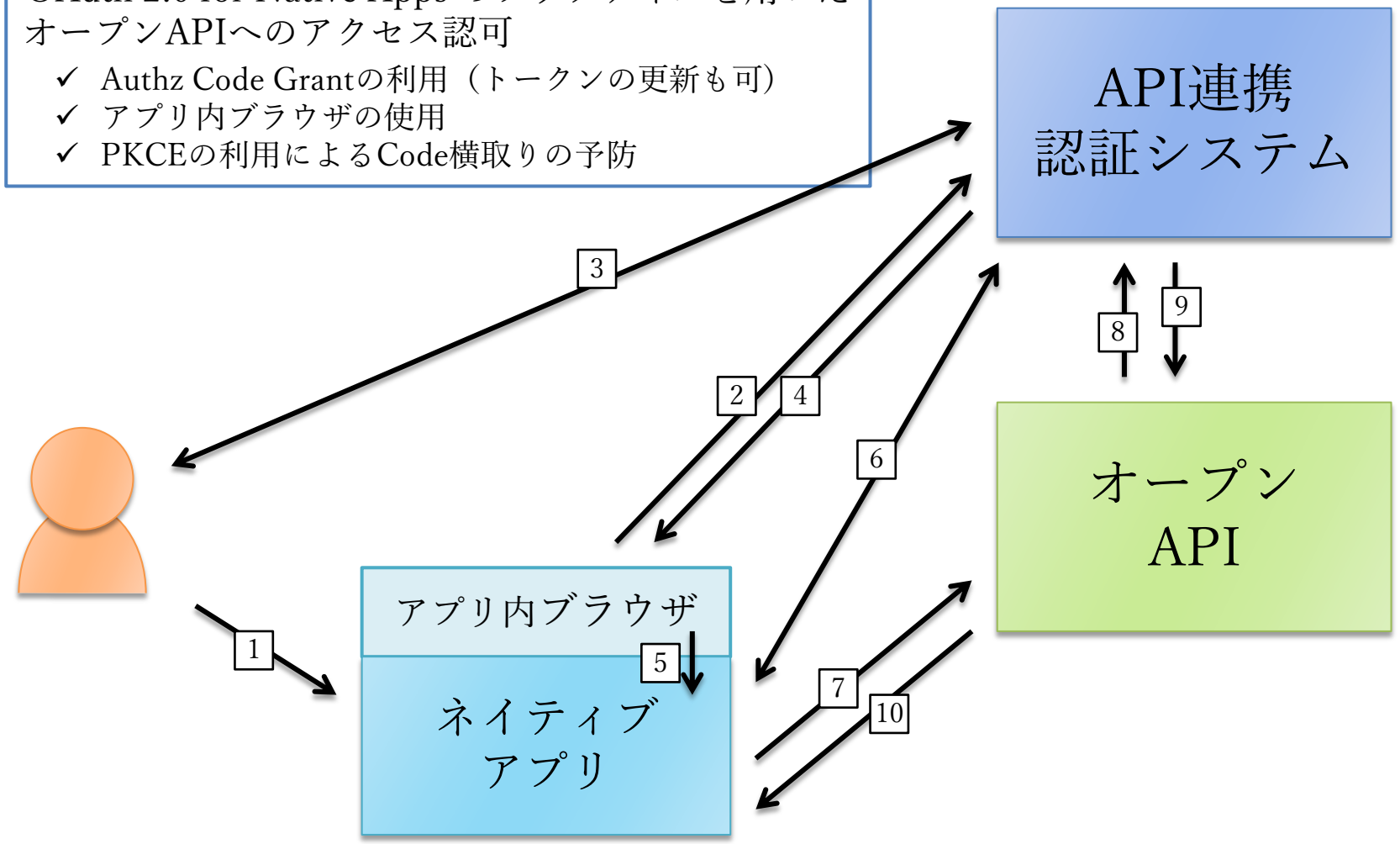
アプリ形態別、対応すべき OAuth の方式

#	利用者への提供形態	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。

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。

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-grade 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。**
  - JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)
  - Client Initiated Backchannel Authentication Profile.

Part 1: Read-Only API Security Profile ①

Draft-5の内容を抜粋

#	Authorization Server への要求事項	Draft-4差分
1	コンフィデンシャルクライアントをサポートする。(SHALL)	—
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. [MTLS] の 2章 で指定されている OAuthクライアント認証のためのTLS相互認証 2. [OIDC] の 9章 で指定されている client_secret_jwt もしくは private_key_jwt	—
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	[RFC7636] の S256コードチャレンジ方式 を使用する。(SHALL)	変更
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)	—

- **OAuth 2.0 [RFC6749] の基本仕様に加え、OAuthシリーズの追加仕様、OpenID Connectで定義された仕様などを併用するように整理されている。**
- **鍵やトークンの長さ、パラメータの使い方、リクエストやデータの検証方法など、より具体的な指定がなされている。**
- **PKCEへの対応が必須に。ネイティブアプリはカスタムスキームではなく App Links, Universal Links への対応が必要に。**

Part 1: Read-Only API Security Profile ②

Draft-5の内容を抜粋

#	Authorization Server への要求事項	Draft-4差分
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)	—
13	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ことが望ましい。(SHOULD)	変更
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	[RFC6749] の 10.10節 にあるように、攻撃者が生成されたトークンを推測する確率が 2^(-160) 以下となるよう、 エントロピーが最低128ビットである不透明な推測できないアクセストークンを発行する。(SHALL)	変更
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	2つ以上の方法でクライアント識別子を送信することを許可するクライアント認証方法では、提供されたクライアント識別子が一致していない場合、[RFC6749] の 5.2節 で定義された invalid_client エラーを返す。(SHALL)	追加
20	httpsスキームを使うリダイレクトURIを使用する。(SHALL)	追加

- **OAuth 2.0 [RFC6749] の基本仕様に加え、OAuthシリーズの追加仕様、OpenID Connectで定義された仕様などを併用するように整理されている。**
- **鍵やトークンの長さ、パラメータの使い方、リクエストやデータの検証方法など、より具体的な指定がなされている。**
- **PKCEへの対応が必須に。ネイティブアプリはカスタムスキームではなく App Links, Universal Links への対応が必要に。**

Part 1: Read-Only API Security Profile ③

Draft-5の内容を抜粋

#	Authorization Server への要求事項（認証されたユーザーの識別子を応答するときの追加要求事項）	Draft-4差分
1	[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)	—
2	[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)	—
3	[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)	—
4	[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)	—
5	[OIDC] 3.1.3.2節にあるとおりトークン要求を検証する。(SHALL)	—
6	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 2.0 [RFC6749] の基本仕様に加え、OAuthシリーズの追加仕様、OpenID Connectで定義された仕様などを併用するように整理されている。**
- **鍵やトークンの長さ、パラメータの使い方、リクエストやデータの検証方法など、より具体的な指定がなされている。**
- **PKCEへの対応が必須に。ネイティブアプリはカスタムスキームではなく App Links, Universal Links への対応が必要に。**

Part 2: Read and Write API Security Profile ①

Draft-5の内容を抜粋

#	Authorization Server への要求事項	Draft-4差分
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	クライアントが（認可要求の中で）state 値を提供した場合は、state 値を保護するため、IDトークンに state 値のハッシュである s_hash を含める。(SHALL) 認可エンドポイントから返される IDトークン に s_hash が存在する場合は、トークンエンドポイントから返されるIDトークンでは (s_hashを) 省略できる。	変更
5	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のみを発行する。(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への対応、オプションとして定義されている高度な仕様への対応**
- **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）**
- **策定中のOAuth追加仕様 (OAUTB, MTLS) の実装**
- **多要素認証の対応**

Part 2: Read and Write API Security Profile ②

Draft-5の内容を抜粋

#	Authorization Server への要求事項	Draft-4差分
10	リクエストもしくは request_uri を通じて渡された全てのパラメータは、署名されたリクエストオブジェクト中に存在しなければならない。(SHALL)	追加
11	7章で説明する リクエストオブジェクトエンドポイント をサポートしても良い。(MAY)	追加
12	パブリッククライアントをサポートするときは、[RFC7636] の S256コードチャレンジ方式 はパブリッククライアントのみで使用する。(SHALL)	追加
13	リクエストオブジェクトは exp クレームを含んでいなければならない。(SHALL)	追加
14	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) （この項目はFAPI Part 1 の 5.2.2.4節 の内容を上書きしている） 1. [MTLS] の 2章 で指定されている OAuthクライアント認証のためのTLS相互認証 2. [OIDC] の 9章 で指定されている private_key_jwt	追加

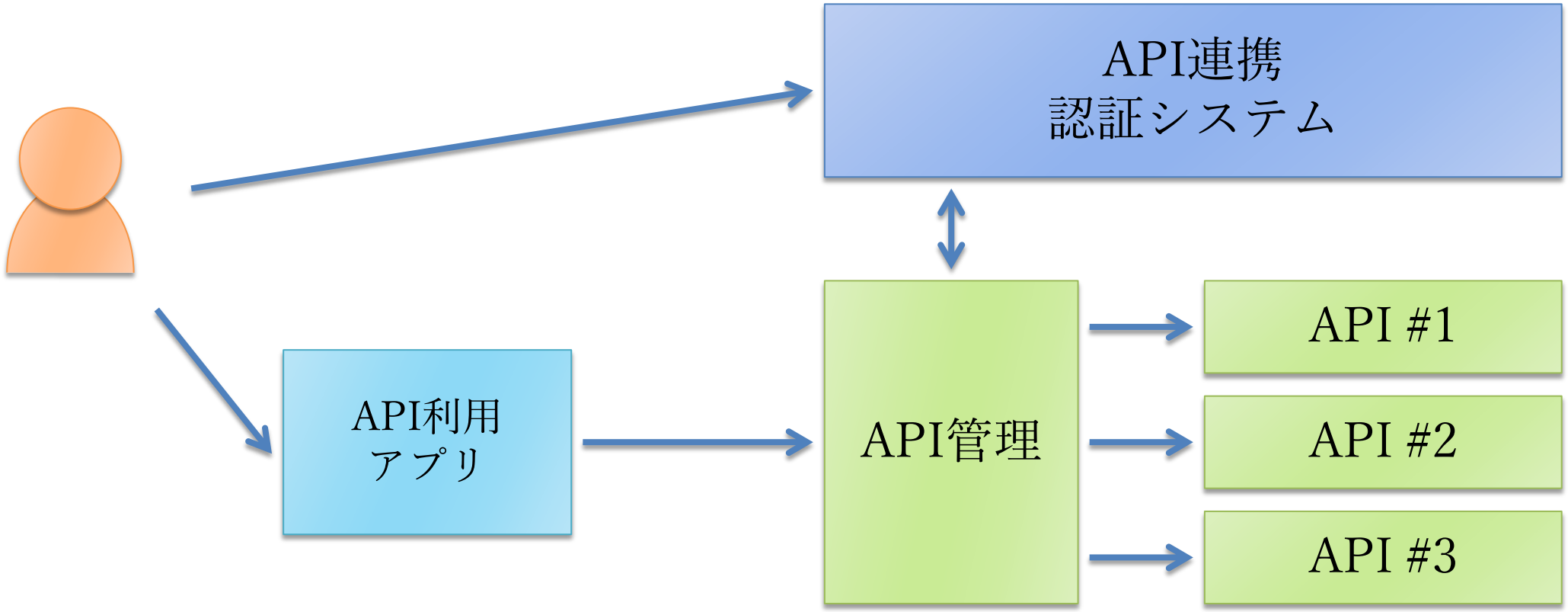
- **OpenID Connectへの対応、オプションとして定義されている高度な仕様への対応**
- **IDトークンを使った認可レスポンスへの署名（at_hash, c_hash, s_hashへの対応）**
- **策定中のOAuth追加仕様 (OAUTB, MTLS) の実装**
- **多要素認証の対応**

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	○	—	○
.1	[OAUTB], [MTLS] を使った記名式トークン、認可コード発行への対応	○	○	○
.2	署名、暗号化されたIDトークンのサポート [Sec.2][Sec.16.14]	○	○	○
.3	長期間有効なトークン発行時の適切な同意画面の表示	○	—	—
②	OAuth 2.0 トークン発行時の scope クレームの応答への対応	○	—	○
③	Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants [RFC 7521]	○	—	○
④	JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants [RFC 7523]	○	—	○
⑤	OAuth 2.0 Token Binding [OAUTB] [draft-ietf-oauth-token-binding-07]	○	○	○
⑥	OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens [MTLS] [draft-ietf-oauth-mtls-11]	○	○	○
⑦	JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)	○	—	○

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



すべてのAPIが金融向けのグレードである必要はない

- FAPI の Profile は主に金融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

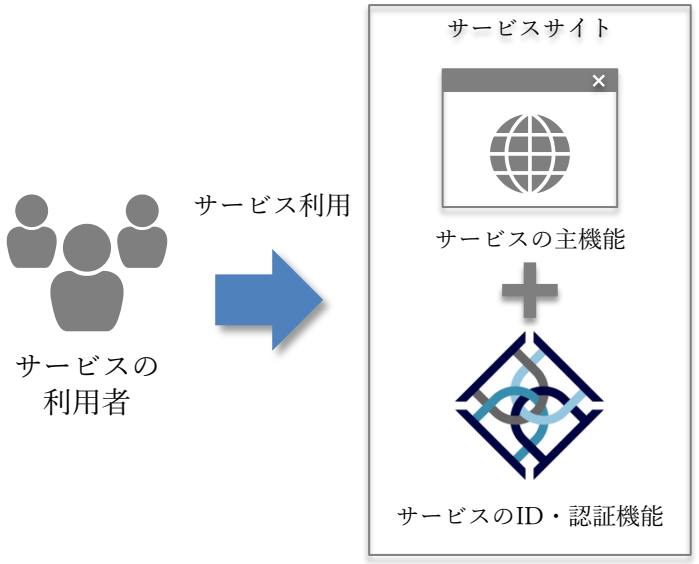
ThemiStruct（テミストラクト）
IdentityPlatform

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

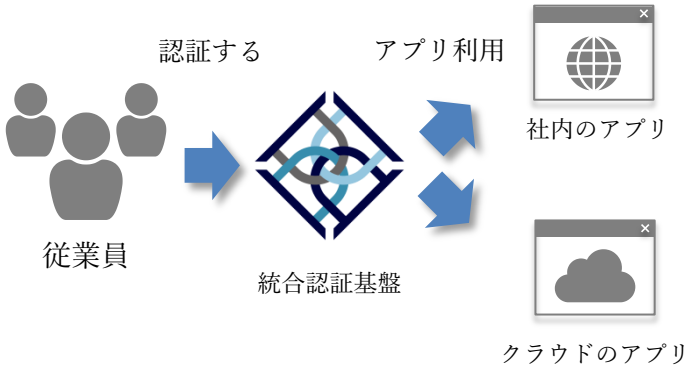
ThemiStruct Identity Platform は、ITシステム利用者のアイデンティティ管理と認証の機能を提供します。顧客向けにサービスを展開したり、従業員向けにアプリケーションを展開する際に必要となる、共通ID基盤の構築に活用いただけます。

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

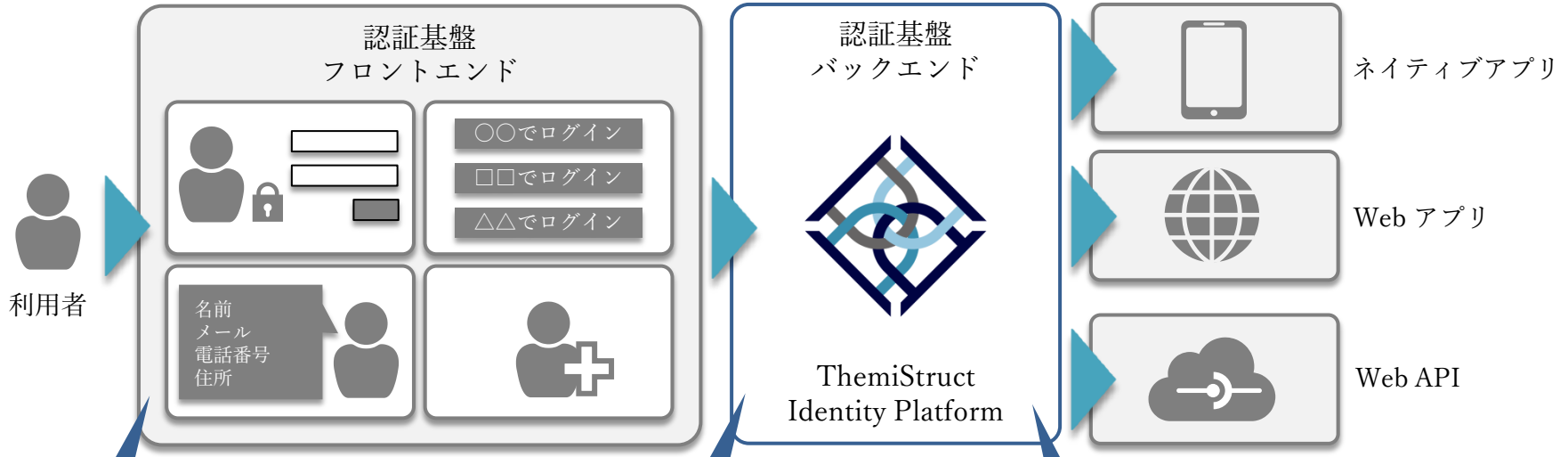
エンタープライズ領域に活用



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

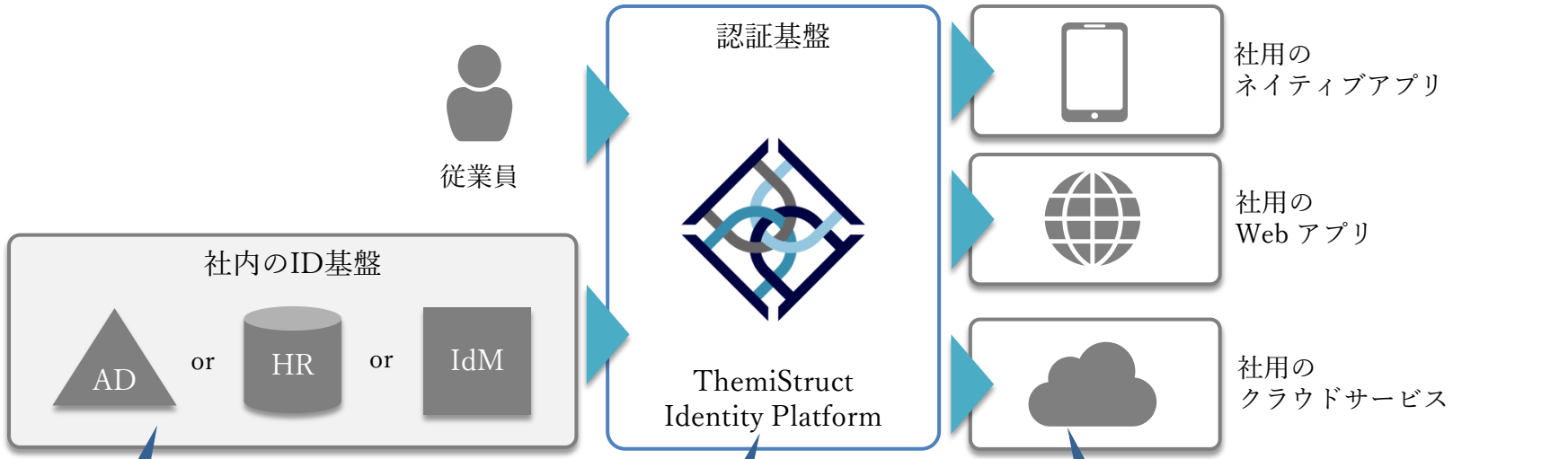
- OAuth 2.0 の技術仕様に基づいた認証・アクセス認可機能を提供します。OAuth 2.0 を使ったAPI連携認証システムが実現でき、モバイルアプリ、オープンAPIを安全に公開できます。

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

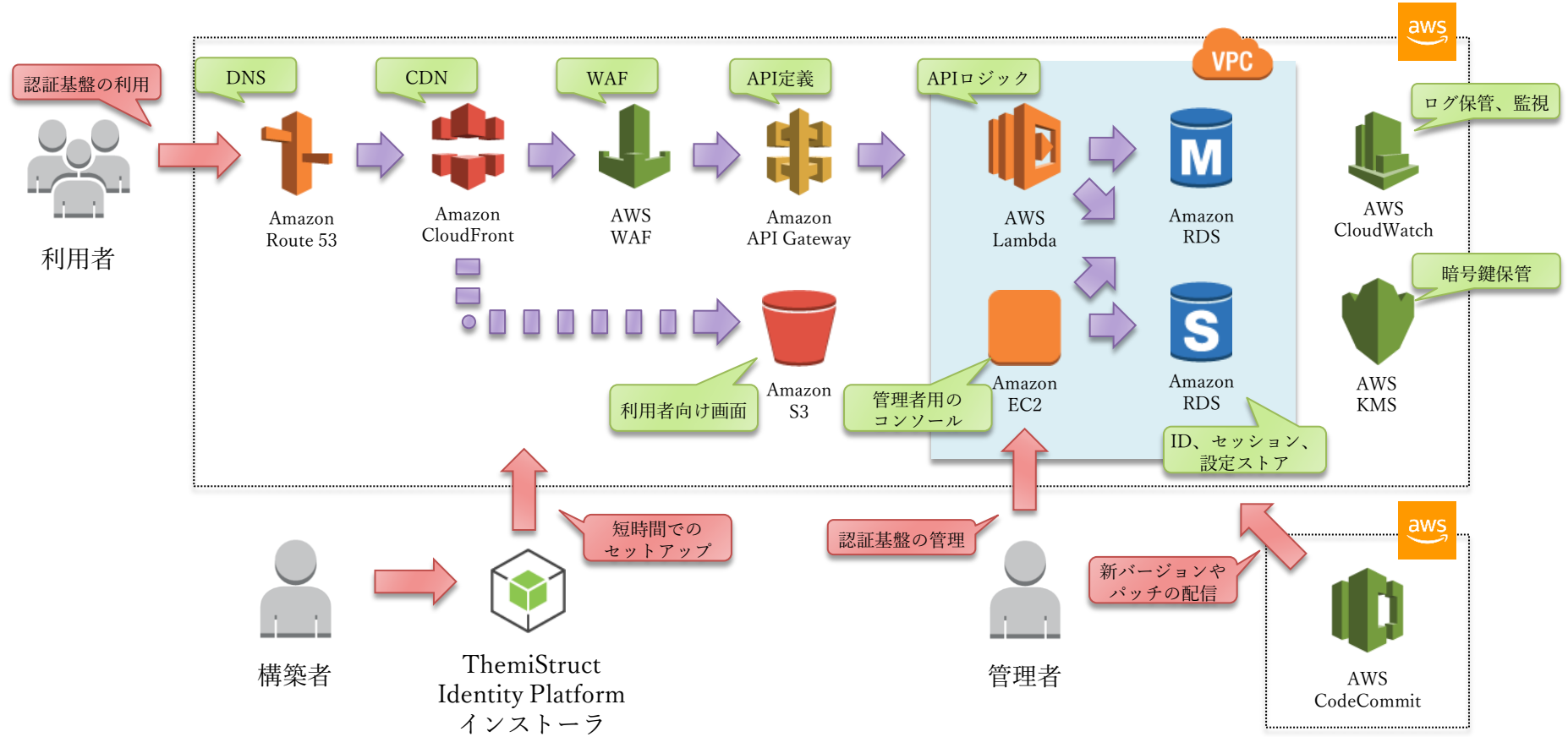
- ユーザーの属性や状態、利用するアプリケーションに応じて、柔軟な認証ポリシーをデザインすることができます。例えば、社内ネットワークからのアクセスの場合、IDとパスワードの認証を提供し、外出先からのアクセスの場合、追加でワンタイムパスワード認証を提供するといったポリシーを展開することができます。

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

- APIの技術調査・検討から運用までのフルカバー
- API利用者のニーズを捉えたビジネスの継続的な進化を実現

お客様API取組状況	課題	ご提供ソリューション
企画・計画	• デジタルビジネスを企画したいが、技術面を担当できる要員がいない • APIの取り組みが初めてで、方法論や技術がわからない	API導入コンサルティング
開発	• 「APIの設計、実装」など技術面でのサポートが欲しい • API公開を迅速に進めたい	API公開支援ソリューション(構築) API管理製品
運用	• 「APIの運用」の技術面でのサポートが欲しい • 公開するAPIに脆弱性がないか確認したい	API公開支援ソリューション(運用) API脆弱性診断サービス
活用	• 作成したAPIを使って、モバイルやAIスピーカーなどクライアントを構築したい	API活用アプリケーション開発

まとめ

まとめ

- 認証基盤のユースケースは、顧客向けサービス向け、オープンAPI向けへと広がり、重要性がより高まっている。
- 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。
- サービス間の連携、利用者の情報の提供にあたっては、利用者の意思を反映することが必要であり、適切なUIが作成・提供できることが必要である。
- 当社では統合認証パッケージ ThemiStruct Identity Platform を提供し、共通ID基盤、API連携認証システム の構築ニーズに対応していく。

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