ThemiStruct
テミストラクト

B2B・B2Cビジネスにおける Webサービスのモデルケースと認証基盤の役割

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
金井　敦

自己紹介

金井　敦

株式会社オージス総研

統合認証ソリューション担当部門　マネジャー

主戦場

金融・保険業界を中心としたB2B/B2C向けの
Identity and Access Management基盤の構築、
プロジェクトマネジメント、ソリューション開発

OpenIDファウンデーション・ジャパン　KYC WG

オージス総研について

会社情報

社名	株式会社オージス総研
代表者	代表取締役社長　中沢　正和
設立	1983年6月29日
資本金	4.４億円　（大阪ガス株式会社100%出資）
事業内容	システム開発、プラットフォームサービス、コンピュータ機器・ソフトウェアの販売、コンサルティング、研修・トレーニング
売上実績	402.4億円（単体）　参考　733.2億円（グループ単純合計）(2018年度)
従業員数	1,439名（単体）　参考　3,456名（グループ合計）（2019年3月31日現在）

オフィス

本社	大阪府 大阪市西区千代崎3-南2-37 ICCビル
東京本社	東京都 品川区西品川1-1-1 住友不動産大崎ガーデンタワー20階
千里	大阪府 豊中市新千里西町1-2-1
うつぼ本町	大阪府 大阪市西区靭本町1-10-24 三共本町ビル10階
名古屋	愛知県 名古屋市中区錦1-17-13 名興ビル
豊田	愛知県 豊田市小阪本町1-5-10 矢作豊田ビル４階

関連会社

さくら情報システム株式会社、株式会社宇部情報システム、
株式会社アグニコンサルティング、株式会社システムアンサー、
OGIS International, Inc.、上海欧計斯軟件有限公司（中国）

ダイバーシティ関連認定

統合認証ソリューションを15年以上やってまいりました

自社オリジナル商品
（主に大規模サービス、B2B/B2C向け）

ThemiStruct（テミストラクト）
Identity Platform

OAuth 2.0 に対応した
APIエコノミー時代に求められる
統合認証パッケージ

オープンソースベース商品
（主に社内・グループ内でのエンプラ利用向け）

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

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連携認証基盤**」を構築する

本日のテーマ

エンタープライズ向け

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

本セッションでの取り扱いテーマ

B2B・B2C 向け

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連携認証基盤**」を構築する

認証基盤が果たすべき役割とは？

認証基盤が果たすべき役割として、本セッションでは下記３つの観点で整理を行いたいと思います。

◆ サービス利用時のUX向上に寄与できる基盤

- ユーザが簡単に手間なくスピーディにサービスを利用できる

◆ 安全にサービスを利用・提供できる基盤

- ユーザがサービスを信頼できる、安心して情報を提供できる
- サービスの安全性が適切に維持され、安定的に提供できる

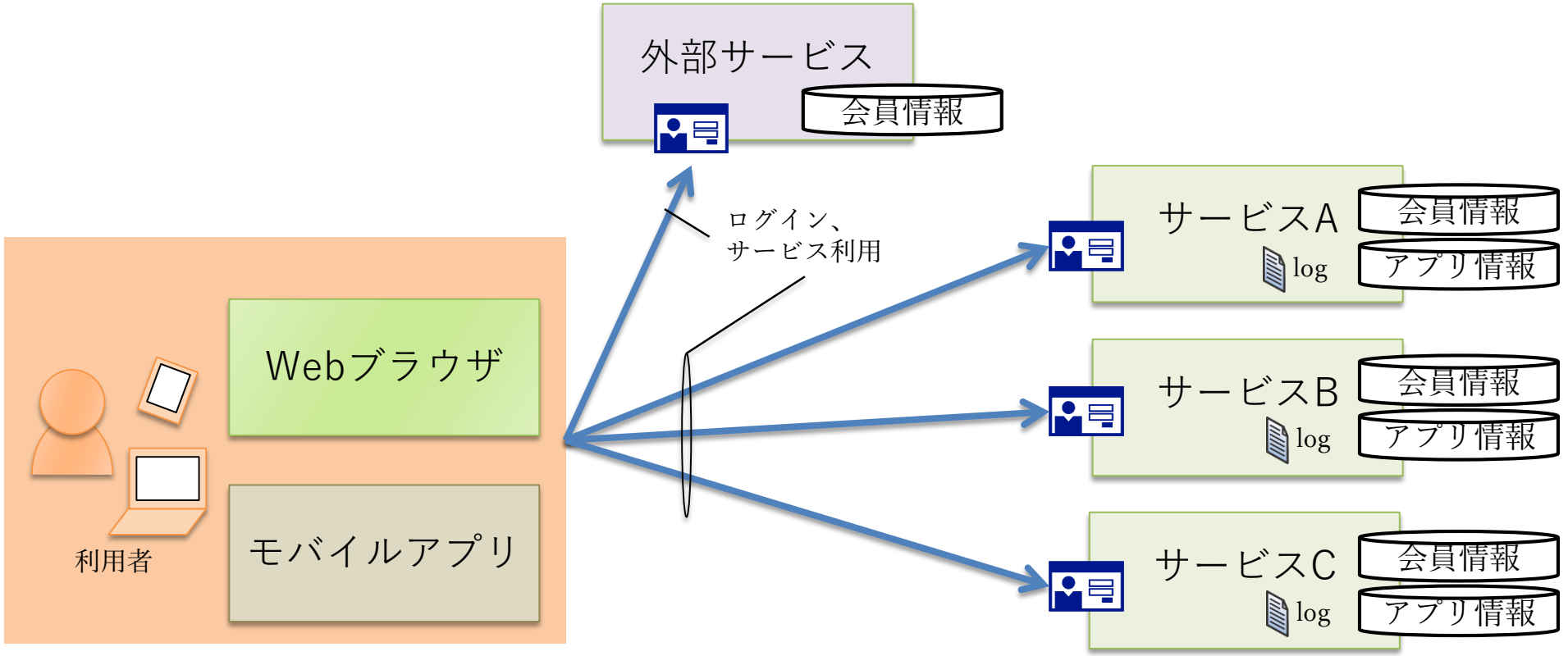
◆ サービスの開発や運営に貢献できる基盤

- 開発資源をメインとなるサービスの機能拡張、改善に集中できる
- システム運用における作業や時間を抑制できる

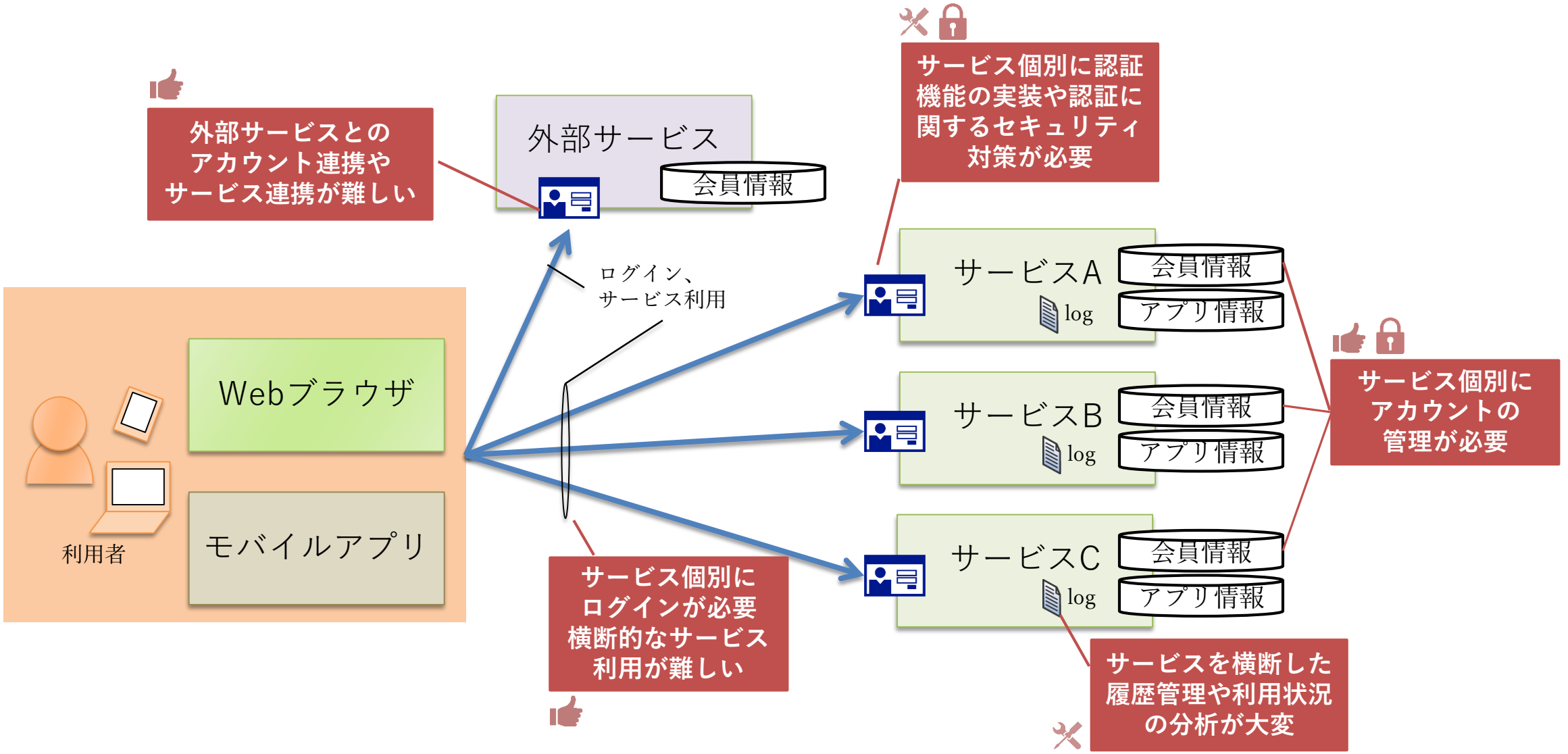
B2B・B２Cビジネスにおける
従来型のWebサービスモデルの課題

従来型のWebサービスモデルケース①

- Webアプリケーションサーバ上で提供される各種サービスにユーザ登録を行い、Webブラウザもしくはモバイル上のアプリケーションからログインしてサービスを利用する。
- アカウントやアプリケーション上のデータ等は個々のサービス毎に運用管理される。



従来型のWebサービスモデルケース①における課題



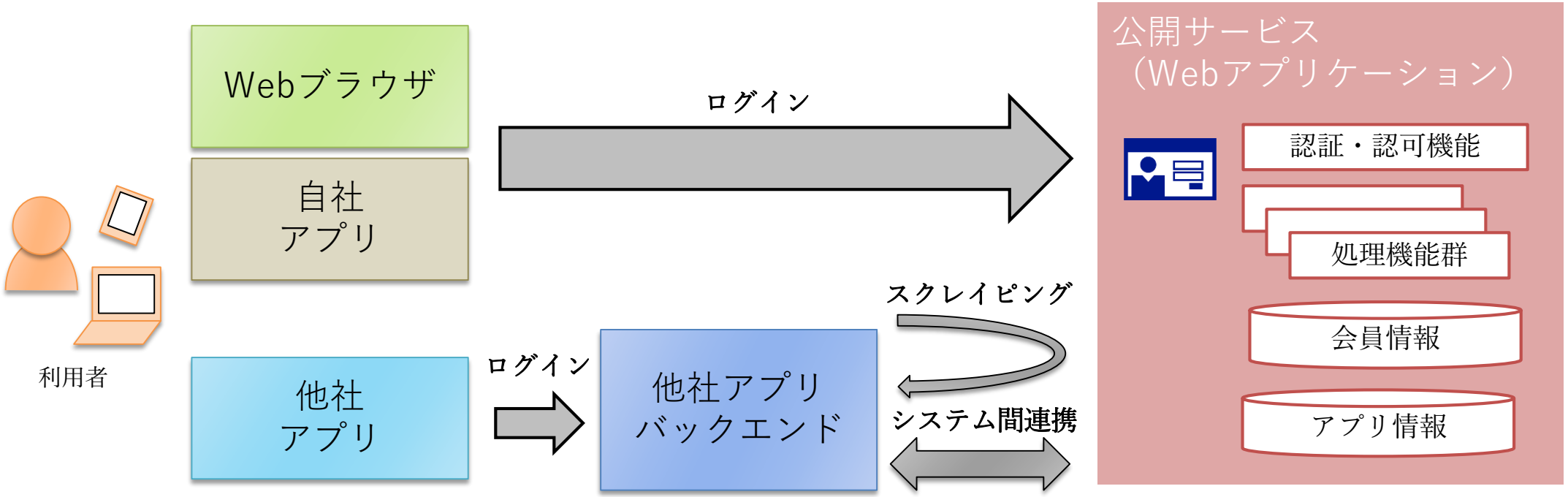
従来型モデルケースの課題整理

✓ ユーザのアカウントを統合してサービス間で共通化したい			
✓ ユーザのログインを統合して横断的にサービス利用できるようにしたい			
✓ 外部サービスのアカウントを利用してサービス利用できるようにしたい			
✓ 外部サービスのアカウントと連携して会員登録できるようにしたい			
✓ サービス個別に認証機能の実装やセキュリティ対策を行いたくない			
✓ サービスの利用状況を横断的に把握・分析できるようにしたい			

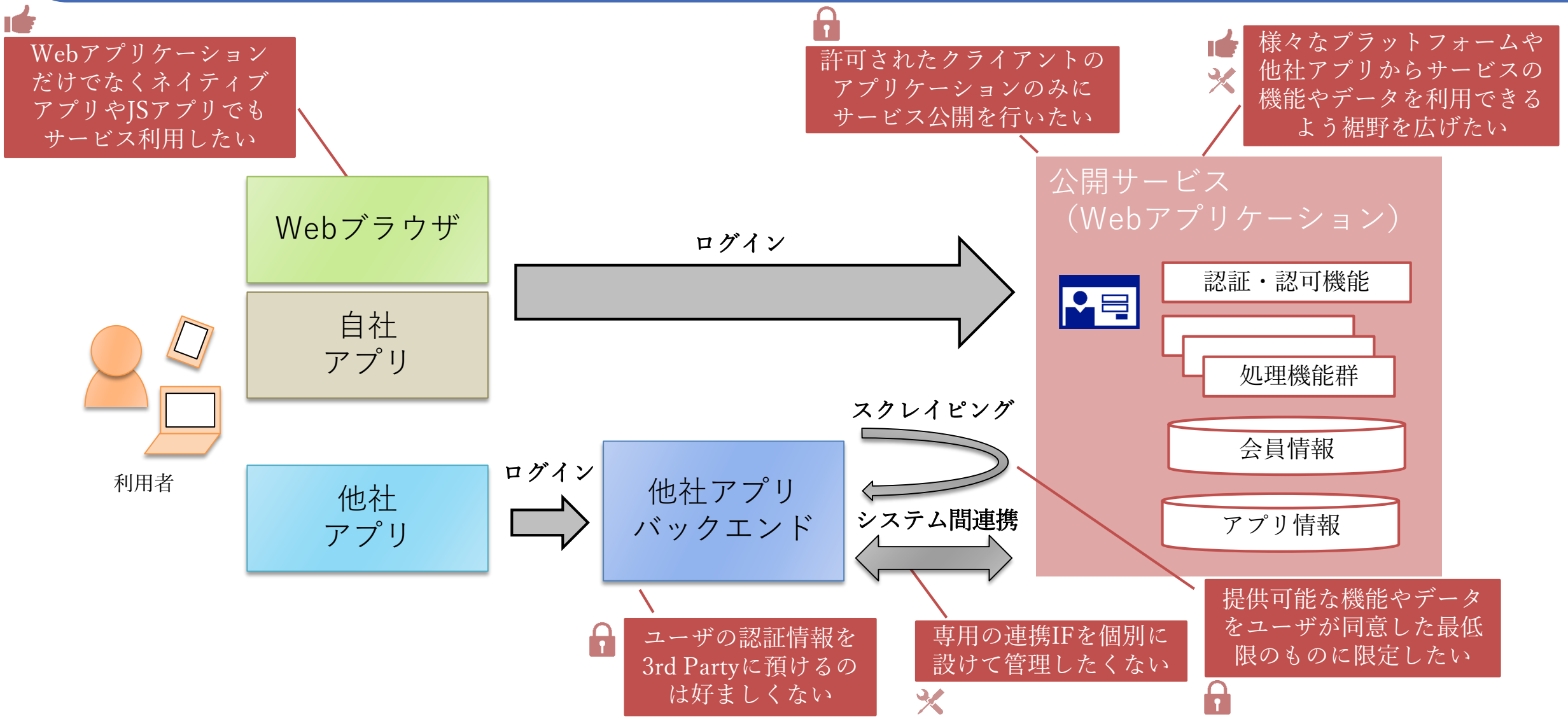
共通ID基盤	◆共通のユーザレポジトリによるアカウント一元管理 ◆共通の認証画面を通じたシングルサインオン機能 ◆外部サービスとの認証連携機能 ◆最新の標準技術仕様に適合した認証機能

従来型のWebサービスモデルケース②

- 利用者向けに提供される各サービス上の機能やデータの利用はWebアプリケーションサーバへのログインを経てのみ利用することができる。（もしくはシステム間連携で行われるシステムの認証を経て）
- 外部サービスとの連携を行うには、システム間連携のための機能を双方のアプリケーションに実装するか、ユーザの認証情報を外部サービスに提供して、外部サービスからログインさせる必要がある。



従来型のWebサービスモデルケース②における課題



従来型モデルケースの課題整理

✓ ネイティブアプリやJSアプリでもサービス利用できるようにしたい			
✓ 3rd Partyアプリに認証情報を保存させずにサービス利用したい			
✓ 外部サービスにWebAPIを公開して安全に利用できるようにしたい			
✓ WebAPIを公開したいが利用できるアプリケーションは制限したい			
✓ WebAPIを通じて提供する機能やデータについてユーザの同意に基づき制限したい			
✓ 外部サービスとの連携のために個別の連携仕様をサービスに実装したくない			

API連携認証基盤	◆WebAPIに対するトークンを利用した認証・認可機能 ◆認可権限設定、権限委譲への同意確認機能 ◆発行されたトークンの検証・失効などの管理機能 ◆最新の標準技術仕様に適合した安全なトークン連携機能

顧客向けサービスサイトの

「共通ID基盤」の構築を考える

「共通ID基盤」の役割

認証統合

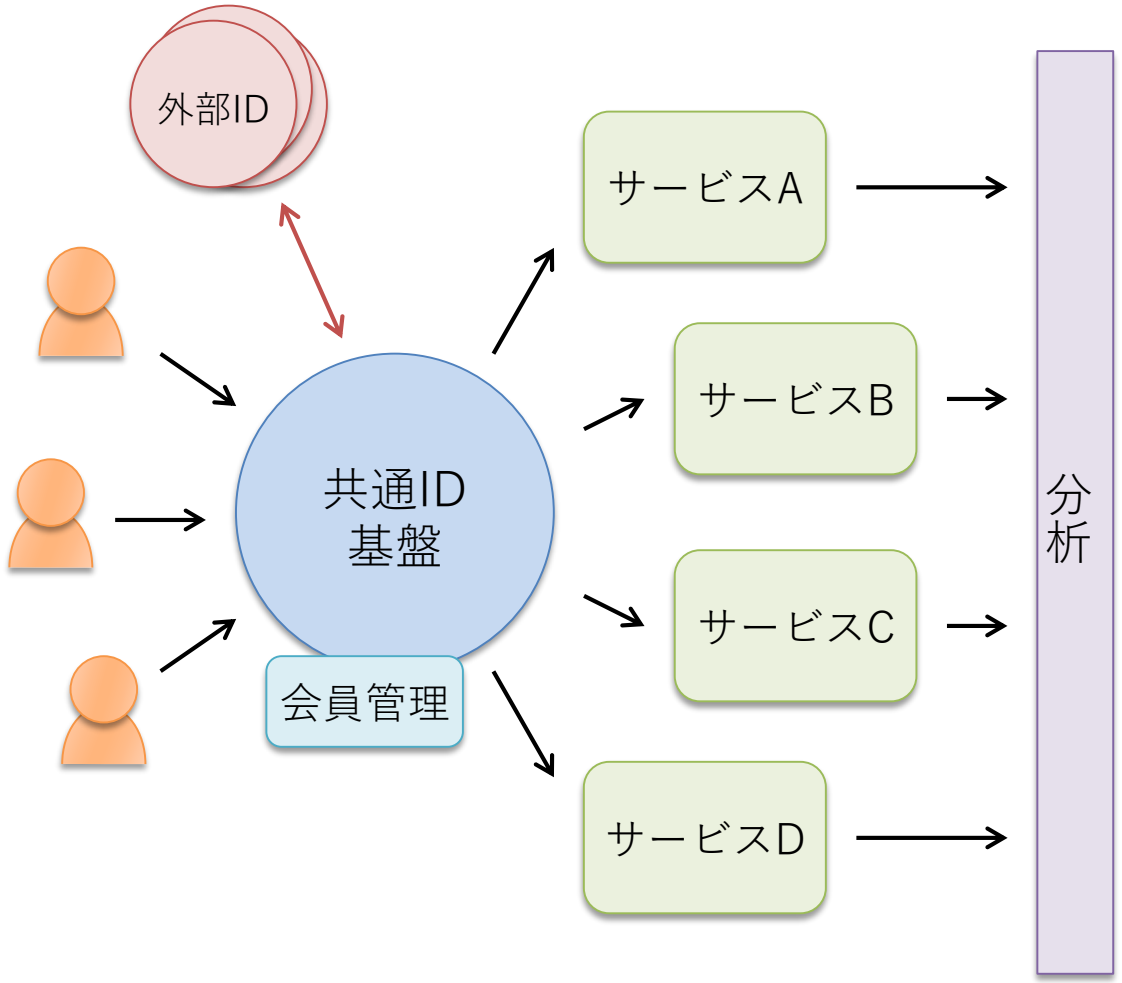
- ◆ 自社サービスの会員IDを共通化
- ◆ 自社サービスのログインを共通化
- ◆ サービス横断での利用状況の把握・分析

認証連携

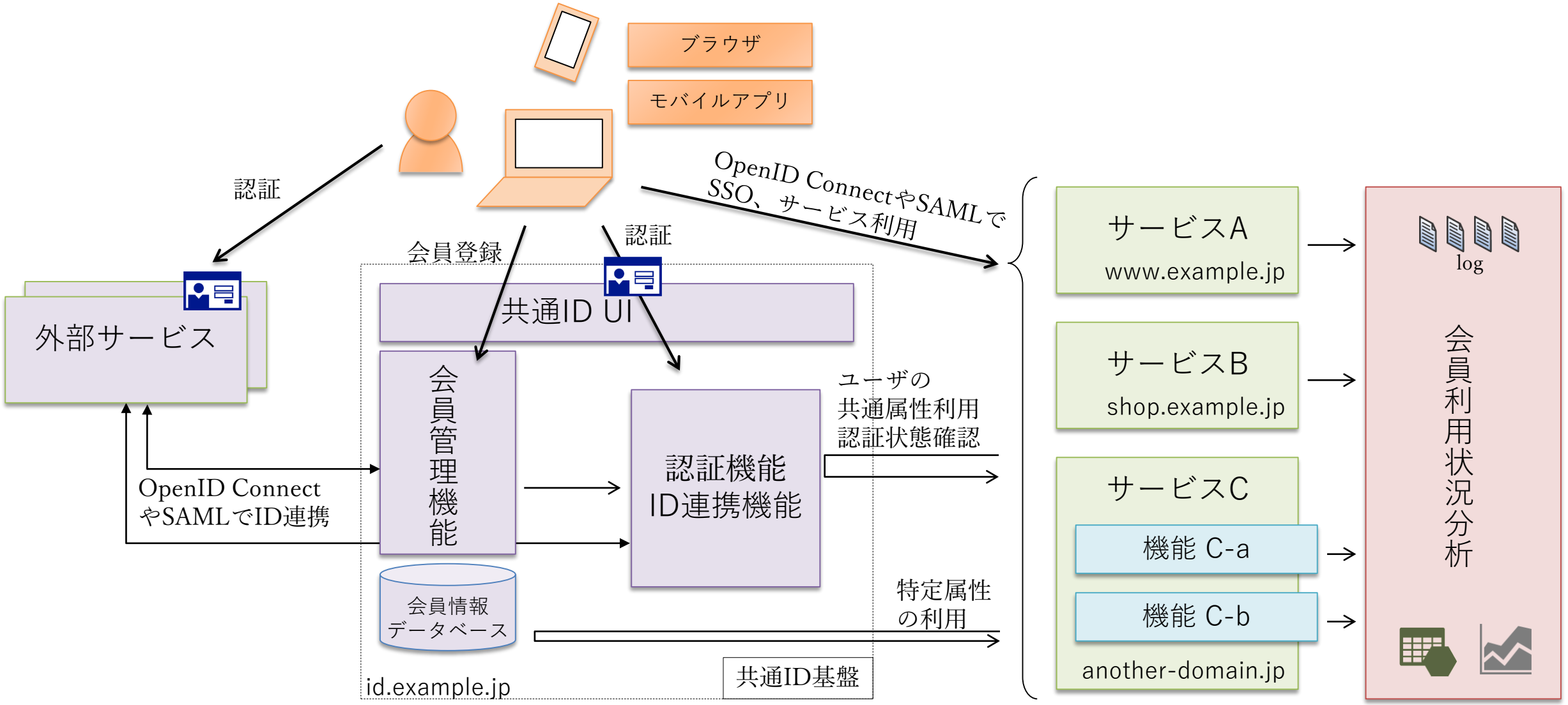
- ◆ 標準技術仕様に適合したシステム間連携の実現
- ◆ 外部IDと連携した会員登録の実現
- ◆ 外部IDと連携したログインの実現

安全策

- ◆ 変わりゆく認証方式への対応
- ◆ 多要素認証による認証強化



「共通ID基盤」のモデルケース



外部IDを利用した会員登録やログインの実現

◆ 外部IDとの認証連携によるメリット

- 外部サービスのIDを使ってログインができ、サービス利用時に**サービス毎のID/パスワードを思い出す必要がなく、アカウントロックやパスワード忘れの発生頻度も削減**される
- **外部サービスが提供する多要素認証の仕組み**をユーザが利用できる
- **新規会員登録の際に外部サービスから**取得できる氏名やメールアドレスなどの**情報を入力フォームに自動補完**できる

◆ 外部IDとの認証連携時の注意点

- 外部IDを利用したログインを行うためには、**事前に自社サービス上のIDと外部サービスのIDを紐付け登録する必要**がありそのための機能が必要
- 利用登録だけでなく、ユーザが任意のタイミングで外部IDの**利用を解除できる機能も必要**
- 標準的なID連携技術に対応していても、外部サービスによって実際の**APIの仕様やセキュリティ対策の状況、利用可能な認証方式、取得可能な情報などはそれぞれ異なる**、また外部サービス側の仕様変更が突然発生する可能性も

認証方式の強化

多段階認証

認証強度の問題

多段階だが多要素ではない

生体認証

ユーザの心理的抵抗感

ユーザへの展開負担

ワンタイムパスワード認証

カバレッジや手間の問題

ユーザへの展開負担

リスクベース認証

精度や環境制約の問題

誤認証の可能性

クライアント証明書認証

証明書発行管理の負担

クライアントへの展開負担

ICカード認証

ICカード発行管理の負担

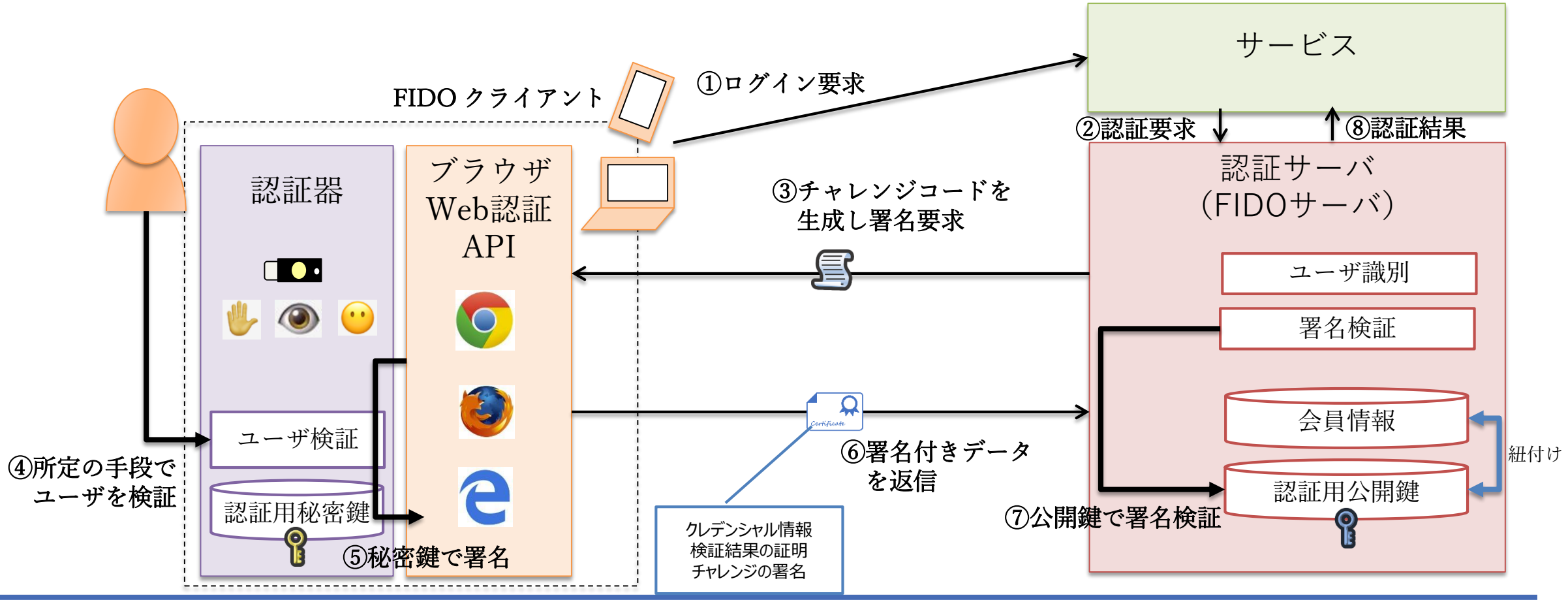
ユーザへの展開負担

負担増やコスト増の懸念から特定業種以外では積極的には採用が進んでこなかった

変わりゆく認証方式

◆ FIDO2の登場

- 公開鍵認証方式を応用してオンライン経由で認証を行う仕組み（≠生体認証、パスワードレス）



変わりゆく認証方式

◆ FIDO2のメリット

- 認証サーバに**パスワード情報や生体情報を保持する必要がない**
- パスワード情報や生体情報が**ネットワーク上を流れない**
- 上記情報がないため、認証サーバが**狙われない、不正ログインできない**
- FIDO対応のブラウザと認証器を持つデバイスがあれば**多要素認証が簡単に実現できる**
- パスワードを使わず認証することもでき、**パスワード管理から解放される**
- FIDO2対応のUSBやBluetooth、NFCの**外部認証器を利用することもできる**

変わりゆく認証方式

◆ FIDO2における注意点

- 利用するには事前にアカウントに対する**認証器の登録作業が必要**
- 認証器を内蔵するデバイスを複数利用しているなど、複数の認証器をサービスで併用したいときは、**認証器毎に事前登録が必要**（１つの外部認証器を複数のデバイスで利用することは可能）
- （現時点では） **Safariは利用できない**
- **一部ブラウザやバージョンが古いブラウザでは利用できない**
- 認証器を紛失した場合の**アカウント復旧方法は別途考慮が必要**
- （当たり前だが）認証器が**手元に無いとログインできない**
- B2CにおいてFIDO認証を**全ユーザ必須にするのは現時点で現実的でない**

「共通ID基盤」構築による効果

◆ サービス利用時のUX向上に寄与できる基盤

- 会員IDやログイン処理の**統合によるユーザ負担の軽減、UXの向上**
- 外部IDとの連携による**集客効果、心理的ハードルの抑制、パスワード管理負担の軽減**

◆ 安全にサービスを利用・提供できる基盤

- **認証方式の強化**や**新たな技術仕様への追従**が容易
- 高い安全性が認められている**標準技術仕様に適合したサービス提供**が可能

◆ サービスの開発や運営に貢献できる基盤

- 開発要員やコストを認証部分に割くことなく、**サービス自体の機能開発に注力**
- ユーザ管理、システム監査、サービス利用分析などの**運用業務の負担軽減や効率化**

オープンAPIを提供するための

「API連携認証基盤」の構築を考える

Web API を公開する際の課題

◆ 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

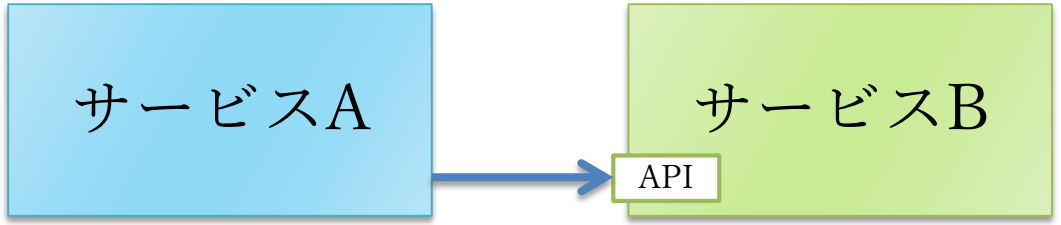
◆ オープンAPI

- 組織の境界の外側の**第三者へ積極的に公開**する Web API
- 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

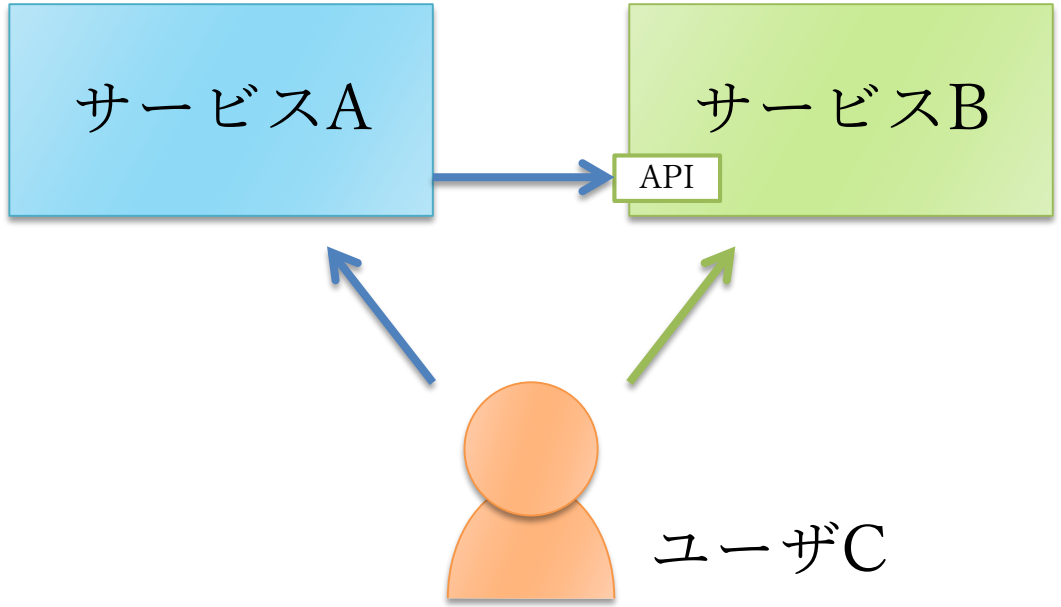
2者間でのAPI利用

- 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。



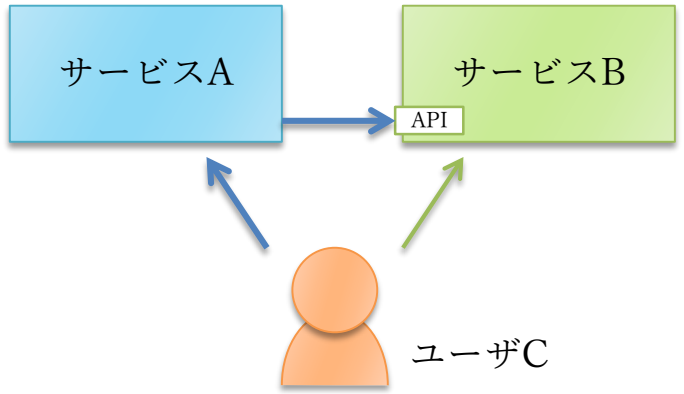
3者間でのAPI利用

- 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Cの意図により、ユーザ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C 向けの機能・データを提供する。



3者間でのAPI利用時の認証

- **ユーザC は サービスA に、アクセスできるデータや操作を限定して、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。
- **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



オープンAPIの提供には「API連携認証基盤」が不可欠

- **OAuth 2.0を利用するには「認可サーバ」の機能が必要**
- **認可（アクセス権の委譲）の前提としてユーザが適切な方法で認証されている必要がある**

- **オープンAPIを提供するためには、API利用者の認証・認可ができる仕組みを構築しないといけない**
  - ユーザを認証する機能（OpenID Connectなどが別途必要）
  - OAuth2.0 認可サーバの機能

API連携認証基盤を構築することに他ならない

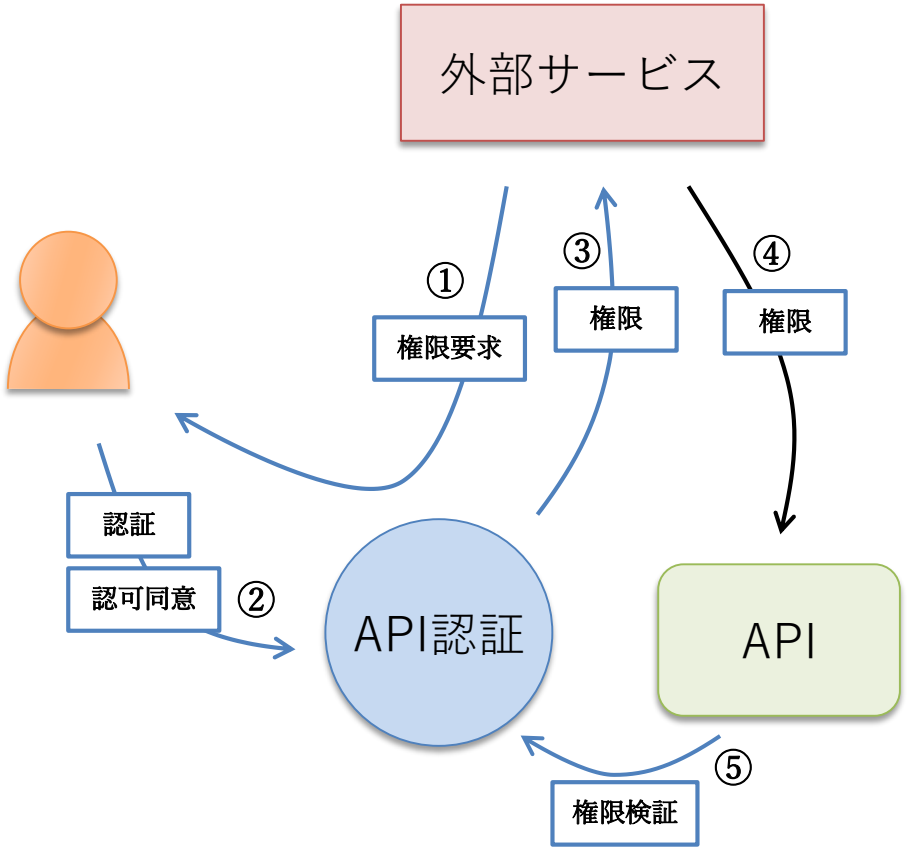
「API連携認証基盤」（OAuth認可サーバ）の役割

API認証

- 外部サービスへのAPI提供、そのための安全な認証機能の実現
- 再ログインを伴わない安全なサービス利用継続の実現
- サービス利用可能なプラットフォームの拡充

権限管理

- 利用者の同意に基づくAPI提供・アクセス許可の実現
- APIアクセス可能なクライアントの制限、アクセス権の適用
- ＋ 共通ID基盤の特性



OAuth 2.0 仕様の概略（最小限）

OAuth 2.0の構成要素

◆ Resource Owner（RO）

- リソースの所有者でありアプリケーションに権限を委譲する人やモノ、主にID/パスワードなどでユーザ認証する

◆ OAuth Client（CL）

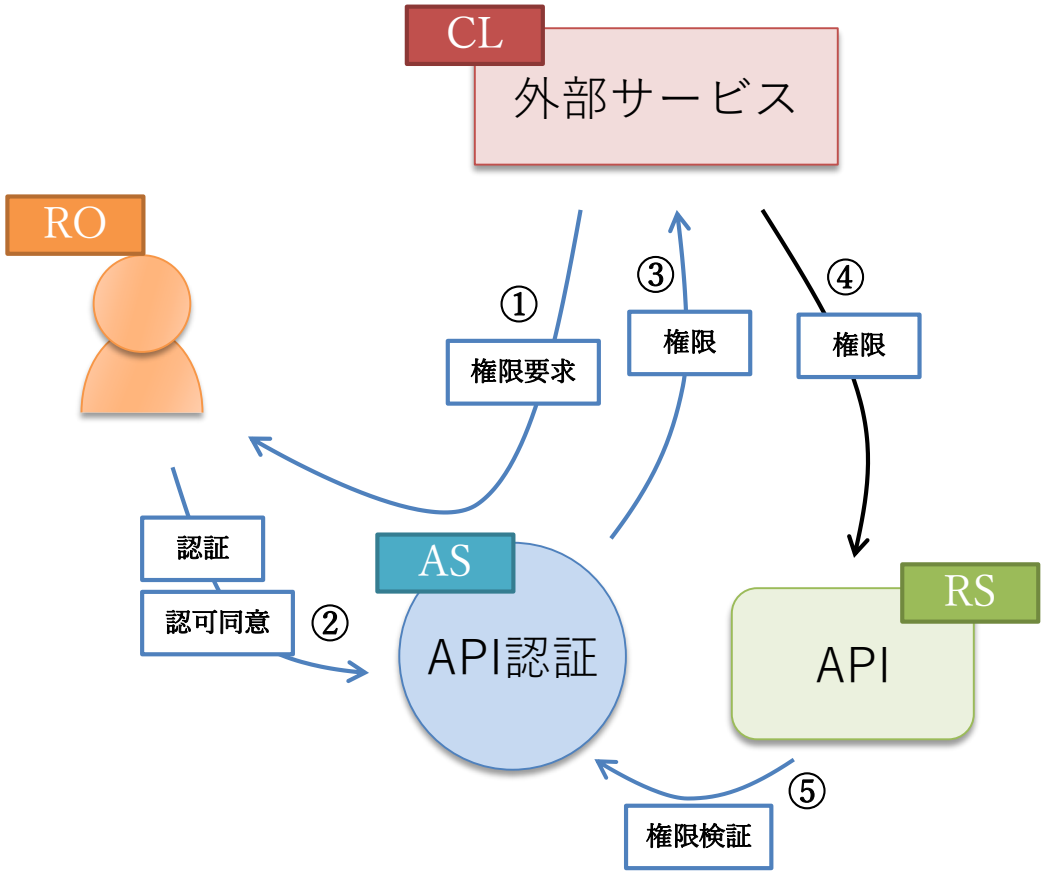
- ROからリソースへのアクセス権限委譲を受けるアプリケーション、ClientID/Secretでクライアント認証することが望ましい

◆ Authorization Server（AS）

- ROがアプリに対して行う権限委譲の仲介をするシステム、認証サーバを兼任してユーザ認証機能を提供することもできる

◆ Resource Server（RS）

- ROが所有するリソースを管理するシステム（オープンAPI）



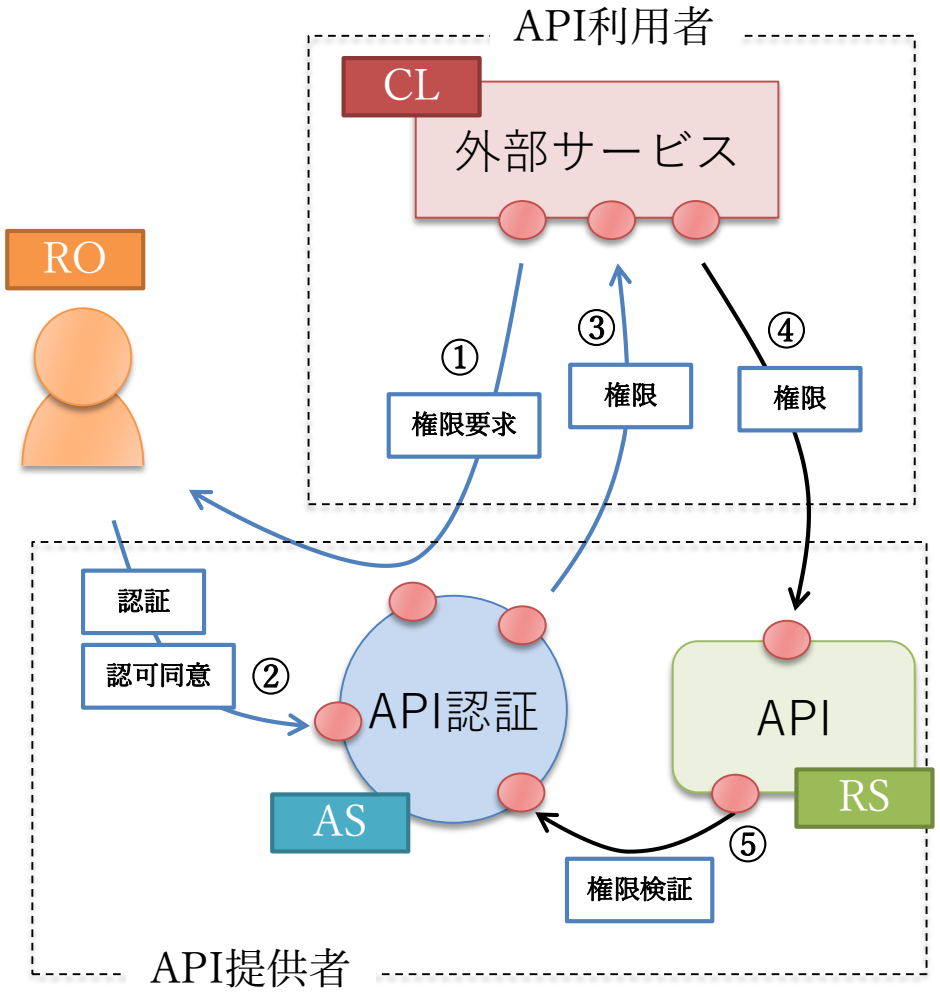
APIの提供者、APIの利用者において必要な対応

◆ オープンAPIの提供者

- API連携認証システム（AS）の実装
- オープンAPI（RS）のOAuth利用に必要な実装

◆ オープンAPIの利用者

- アプリケーション（CL）のOAuth利用に必要な実装
  - フロントエンドアプリケーション
  - バックエンドアプリケーション



対応が必要となる OAuth の具体的仕様類

#	仕様	AS	RS	CL
①	The OAuth 2.0 Authorization Framework [RFC 6749] ASがCLの認可を行いトークン発行するための仕様	○	―	○
②	The OAuth 2.0 Authorization Framework: Bearer Token Usage[RFC 6750] CLがトークンを利用してRSにアクセスするための仕様	―	○	○
③	OAuth 2.0 Token Revocation [RFC 7009] CLに発行したトークンを失効させるための仕様	○	△	○
④	Proof Key for Code Exchange by OAuth Public Clients [RFC 7636] CLがトークンをASから安全に取得するための対策	○	―	△
⑤	OAuth 2.0 Token Introspection [RFC 7662] RSがトークンを検証するために必要な情報をASに要求するための仕様	○	○	―
⑥	OAuth 2.0 for Native Apps [RFC 8252] CLをネイティブアプリで実装する際のベストプラクティス	―	―	△

※　上記以外にも実装にあたりセキュリティ面で考慮すべき事項についてOAuth 2.0 Threat Model and Security Considerations [RFC 6819]として公開されている。

API利用時の構成パターン

◆ バックエンドあり（Confidential Client）

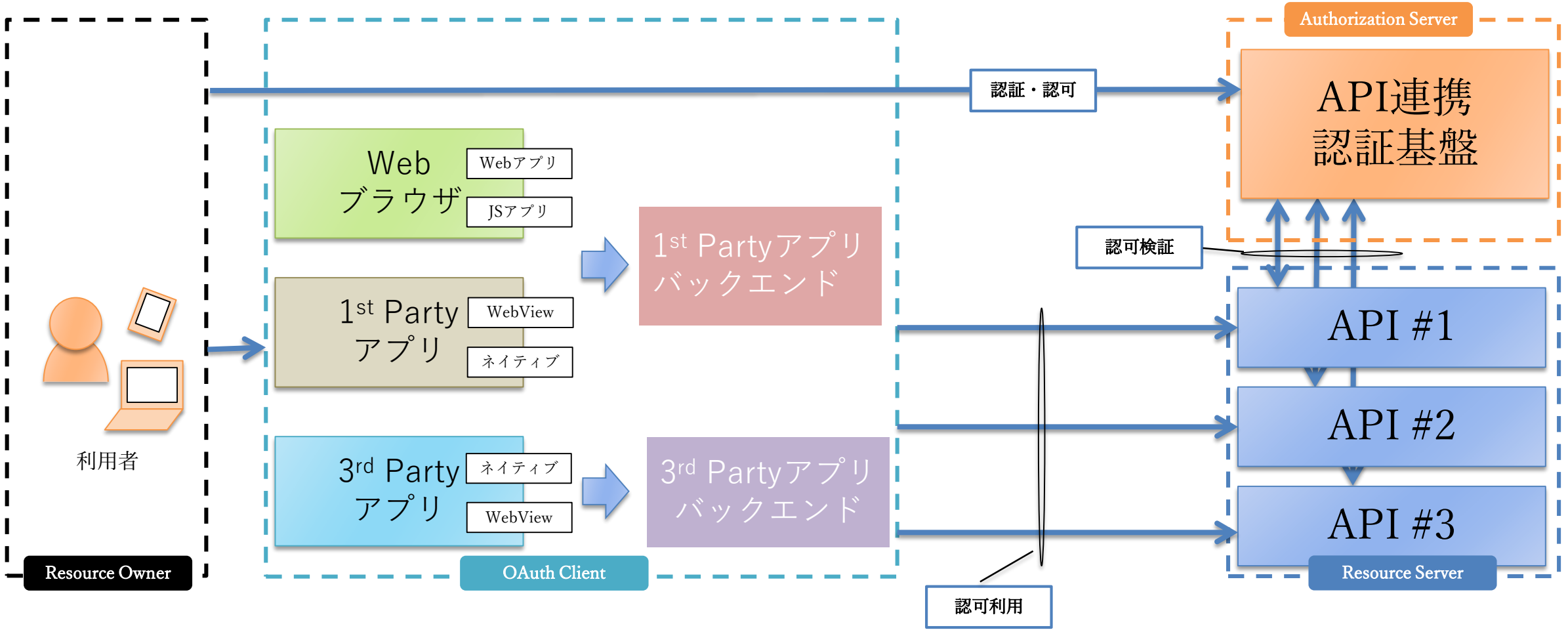
◆ バックエンドなし（Public Client）

アプリ形態別、対応すべき OAuth の方式（RFC 6749）

#	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 ➡ Authz Code Grant w/ PKCE に向かう	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）	Authz Code Grant	Webアプリケーションサーバがオープン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。

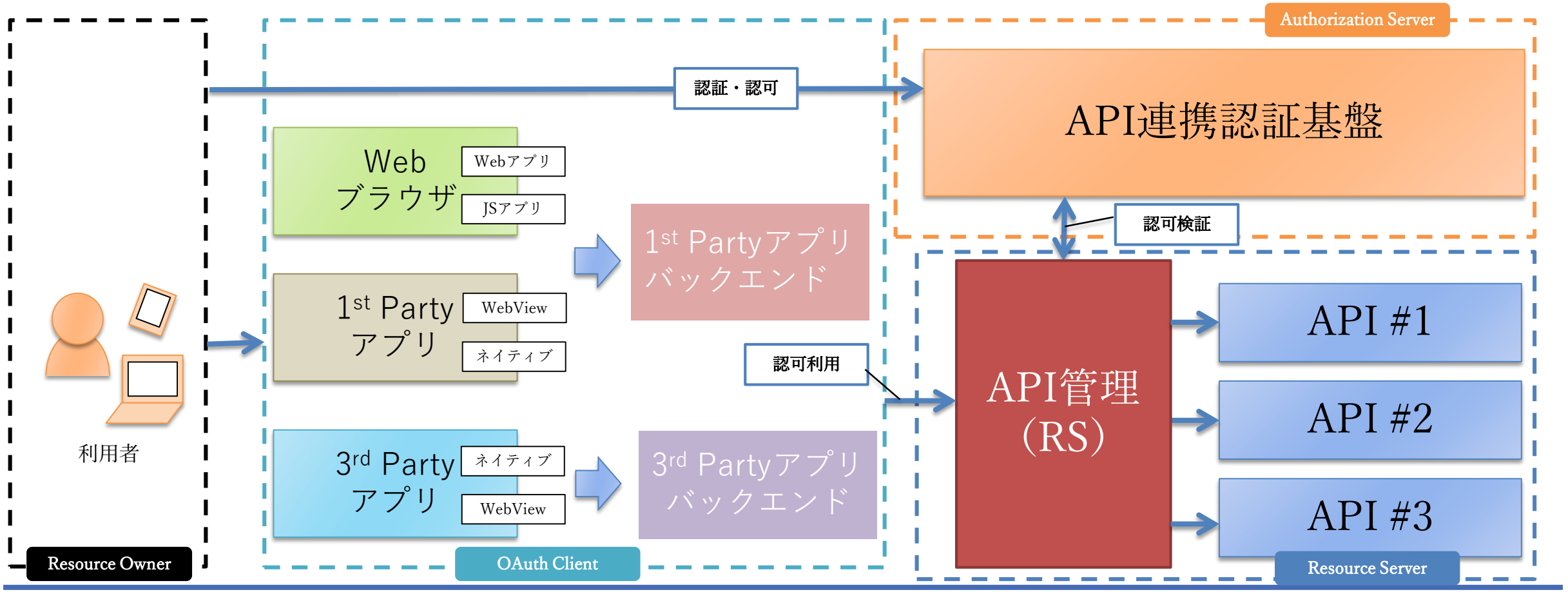
「API連携認証基盤」のモデルケース

◆ API連携認証基盤を導入したオープンAPIの提供モデル



API管理のソリューションの導入も効果的

- 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



「API連携認証基盤」構築による効果

◆ サービス利用時のUX向上に寄与できる基盤

- **再認証を伴わないサービス利用継続**による利便性向上
- **様々なプラットフォーム上**で同じサービスをユーザの状況に応じて利用できる

◆ 安全にサービスを利用・提供できる基盤

- 公開APIに対する**認証や範囲を限定した認可、権限設定が可能**
- **認証情報**自体をクライアント側に**保持させることなく期限を限定**し公開APIを利用許可
- 公開APIを通じたサービス連携に対して**利用者自身による同意や失効手続きが可能**

◆ サービスの開発や運営に貢献できる基盤

- 外部サービスへの連携により**新たな価値、新しいサービス領域を創造できる可能性**

まとめ

まとめ

- **認証基盤のユースケースは、社内統合認証にとどまらず、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ます。**
- **認証基盤を導入することでUX向上や安全かつ効率的なサービス提供に貢献することができます。**
- **API公開や外部アプリケーションとの安全な連携のためにはOAuthやOpenID Connectといった最新の認証連携に関する標準技術仕様群の追加更新に継続的に対応していく必要があります。**
  - セッション②【認証標準技術の必要性と最新動向】において最新の技術動向をご紹介致します。
- **当社では統合認証ソリューションを提供。それらを組み合わせて共通ID認証基盤、API連携認証基盤の構築ニーズに対応できます。**
  - セッション③【「共通ID基盤のスピード導入」と「ソーシャルサイトを対象としたアイデンティティ連携」の実現方法】において当社認証ソリューションをご紹介致します。

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