APIエコノミー実践：API公開に必要なプロセスとプラットフォーム

株式会社オージス総研
サービス事業本部 クラウドインテグレーションサービス部
齋藤 伸也（Saito_Shinya@ogis-ri.co.jp）

目次

- APIエコノミーの概要
- API公開プロセス
- APIの認証・認可
- API公開のプラットフォームの活用
- オージス総研のAPI関連ソリューション

オージス総研のAPIへの取り組み

- 10年以上にわたり、システム連携にコミット
- 2012年からAPI案件に取り組み開始
- すでに多数のAPI開発・公開案件を実施
  （EC、インターネットサービス、金融、エネルギー、医療、製造、メディア等）

	取り組み
2001	SOA、連携基盤に関する技術開発を開始
2007	システム連携基盤構築サービス提供開始
2012	EC向けAPI連携プラットフォームサービス提供開始
2013	EC関連を中心にAPI取り組みを強化
2014	APIゲートウェイ技術開発開始
2015	データ簡単API化サービストライアル提供開始、IoT系API案件の増加
2016	社外向けAPI公開案件の増加

オージス総研

APIエコノミーの概要

デジタルビジネス時代の到来

オージス総研

□ デジタルビジネス = 自社の提供価値（既存事業）× 他社の提供価値 × IT

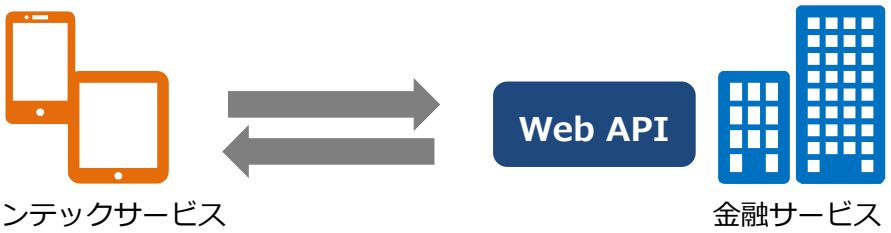
配車サービス × ホテル・旅行代理店	• ホテルの場所に迅速な配車サービス • 観光地に通じた運転手を手配
会計サービス × 銀行	• リアルタイムの会計情報で与信・融資 • すぐれた会計サービスのUIから企業間の支払い
損害保険 × 自動車	• 安全な運転に応じて保険料を変動 • 事故時の迅速なフォロー

□ これら新しいビジネスの背後には「API」を通じたサービス、データの連携がある

□ より「オープン」な形で、より「ビジネス」に直結するAPIが公開され、APIを通じてビジネスを行う「APIエコノミー」の世界が急速に広がってきている

FinTechとAPI

- APIによって、新しい金融サービスを提供するITベンチャー企業と既存の金融企業を結びつける
- 進む環境整備



  - 改正銀行法可決、成立 (2017/5/26)
  - 経済産業省：「クレジットカードデータ利用に係るAPI連携に関する検討会」を開催（2017.3.31）
  - 全銀協に「オープンAPIのあり方に関する検討会」を設置（2016.10.21）

- 金融機関の取り組み

  - 三井住友銀、アプリで入出金照会 2社と連携 (2017/7/27)
  - 千葉銀、アプリ運営会社と提携 (2017/7/27)
  - 富山第一銀、本部横断のフィンテック専門チーム (2017/7/11)
  - マーケット情報を瞬時にスマホ配信 QUICKが新サービス (2017/6/28)
  - 常陽銀、個人通帳のアプリ開発　年内にも提供へ (2017/6/13)
  - みずほ銀など、資産管理アプリと口座を連動　安全性より高く (2017/5/22)
  - メガバンクが「更新系API」を提供開始、マネーフォワードが経費精算振込で連携（2017.3.31）
  - 三菱UFJが「API」開放へ（2017/3/12）
  - メガバンク各行、ハッカソンを開催　(2016年)

IoTとAPI

- IoT（Internet of things）では、APIはモノとクラウドを繋ぐ役割
- 自社製品（H/W、スマホアプリ）の付加価値向上 ⇒ 製品のサービス化
  - 製品（センサー）が生み出すデータをもとに付加価値サービスを提供
  - 社外のサービスやデバイスとの情報連携により、顧客に対して新しいサービス体験を提供
- API活用例
  - 製造業など：機器の故障やメンテナンス時期を把握し、迅速・的確なアフターサービスを提供
  - 大手百貨店でのスマホへのセール情報送信（店舗内のセンサーで得意客を検知し、セールなどお得情報を通知）
  - エネファームIoT

[出所]
http://media.amazonwebservices.com/jp/summit2016/1B_02.pdf

APIによるビジネスモデル（期待効果）

無料	無償提供によるユーザ拡大	一部のデータや機能をAPIとして公開し、顧客ベースを拡大する
開発者支払	有料API	提供するデータ、サービスそのものの価値に課金
開発者に支払	アフィリエート	APIは新しい販売チャネル。APIを通じて得られた収益を開発者に一部還元する
間接収益 （売上増）	プレミアムサービスとしてAPI提供	上位サービスの差別化により、顧客単価増を狙う ・上位サービスのみに提供されるデータ、機能 ・上位サービスのみAPIでアクセス可能（顧客システムへ組み込み可能）
	パートナーとのビジネス拡大	・チャネルとして活用（パートナー経由で、自社データ、サービスを販売） ・企業連携による新規サービス・ビジネスの展開
	APIによる開発期間短縮効果 間接収益 （生産性向上、費用削減）	・新規アプリケーション、新規サービス立ち上げ時の工数削減

APIエコノミーの具体例

オージス総研

- ビジネスの変革と売上への貢献
  - 寺田倉庫（minikura API）
    → 「モノを管理する」倉庫システムと「モノを動かす」物流システムの機能をAPIで公開
      - 倉庫、物流を必要とする企業とのコラボレーションが実現、協業企業のエンドユーザー（新規顧客）の動向を収集できるように
  - 米ピツニーボウズ
    → 郵便料金計器の製造販売から、グローバルEコマース等のサービス事業へシフト（3年間で200API公開）（日経コンピュータ 2017.4.27)
  - 日本医療データセンター様
    →医療統計データの新サービス提供で、個人向けの新たなサービスの創出を推進

オージス総研

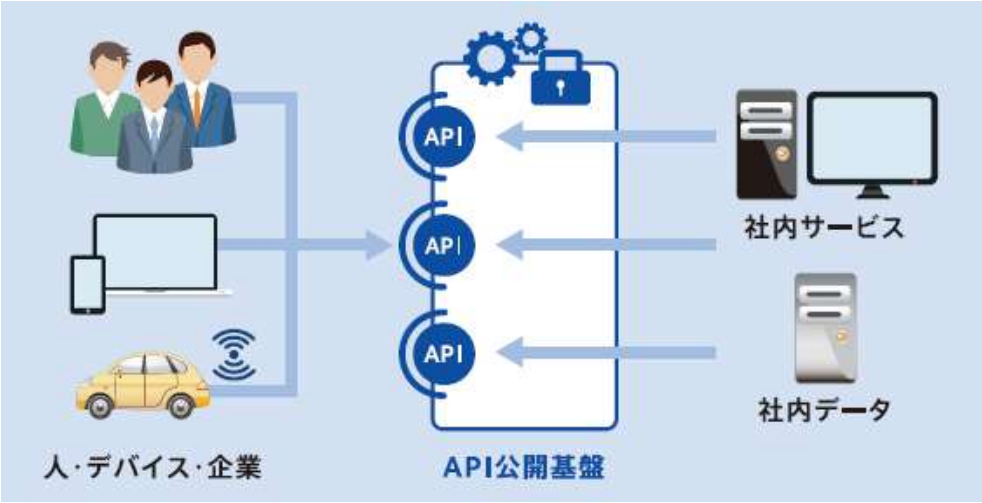
API公開プロセス

API公開を成功させるには変化への素早い対応が重要

- デジタルビジネスは変化が激しく予測が難しい

- APIに求められる変化

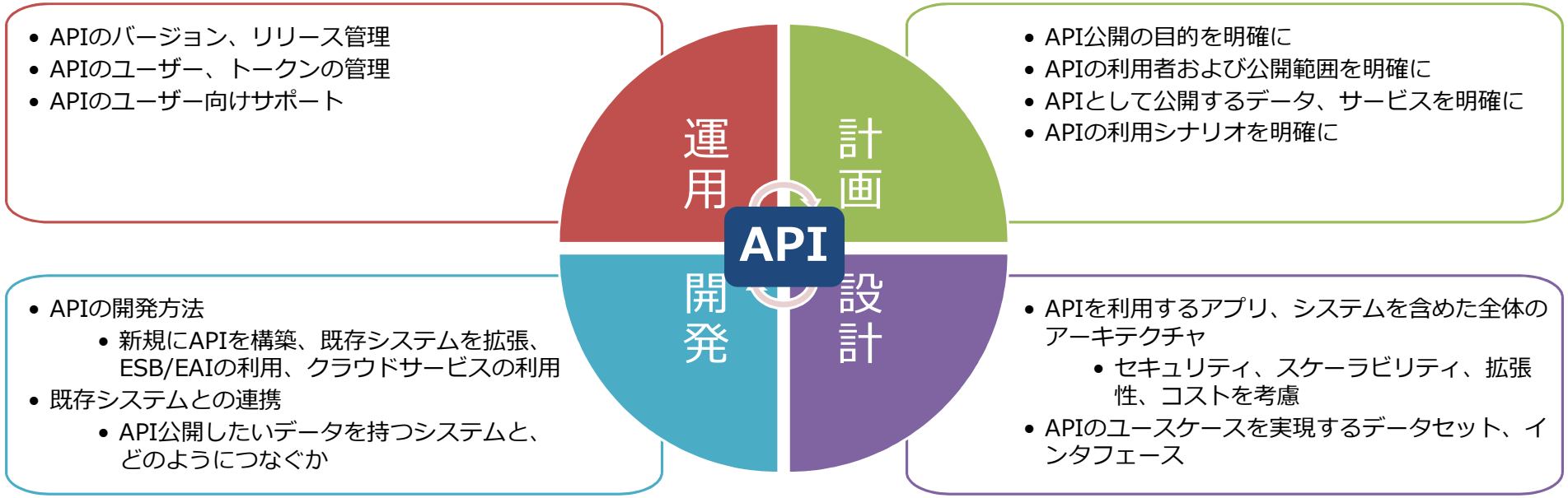
  - 既存のデータやサービスを拡張
  - 新しいデバイスや新しいビジネスパートナーの増加



APIライフサイクル

- APIライフサイクルは、一度きりの取り組みではない
- デジタルビジネスの成長、変化にあわせAPIを改修し、バージョンアップすることが必要

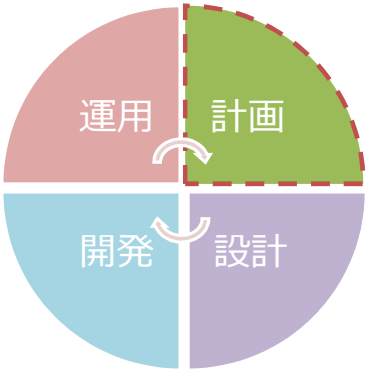
  → ライフサイクル管理が大切



API公開のライフサイクル:計画

API公開の計画で重要になるポイント

- API公開の目的を明確にする
- APIの利用者および公開範囲を明確にする
- APIとして公開するデータ、サービスを明確にする
- APIの利用シナリオを明確にする



API公開範囲の種類について

プライベート

メリット： 様々なクライアントから共通的に利用可能なモジュールを提供できる。

例: モバイル向けのバックエンドAPI

パートナー

メリット: パートナーとの新規協業、立ち上げの迅速化ができる。

例: 取引先、代理店向けのカタログAPI

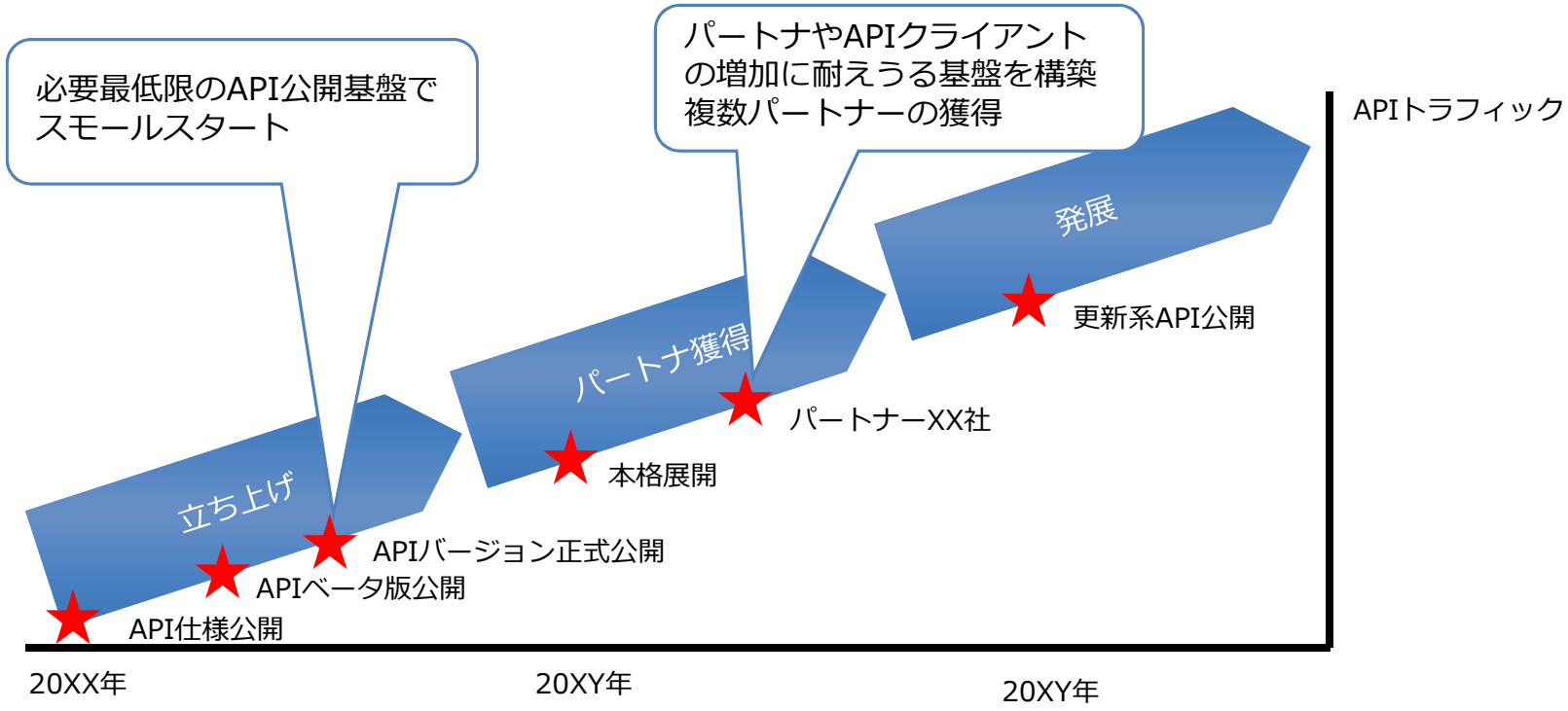
パブリック

メリット: ビジネスをプラットフォーム化することを実現できる。

例: オープンに公開されているMap API

例：製造業様API公開ロードマップ

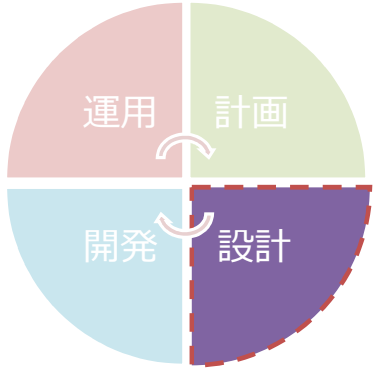
- IoT対応した製品を中心としたエコシステムを構築することを上位目標にすえ、APIをエコシステム拡大の手段としてロードマップを作成



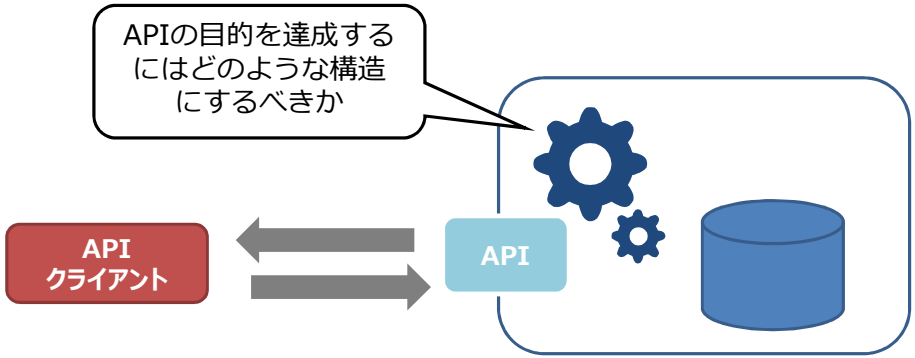
APIライフサイクル:設計

オージス総研

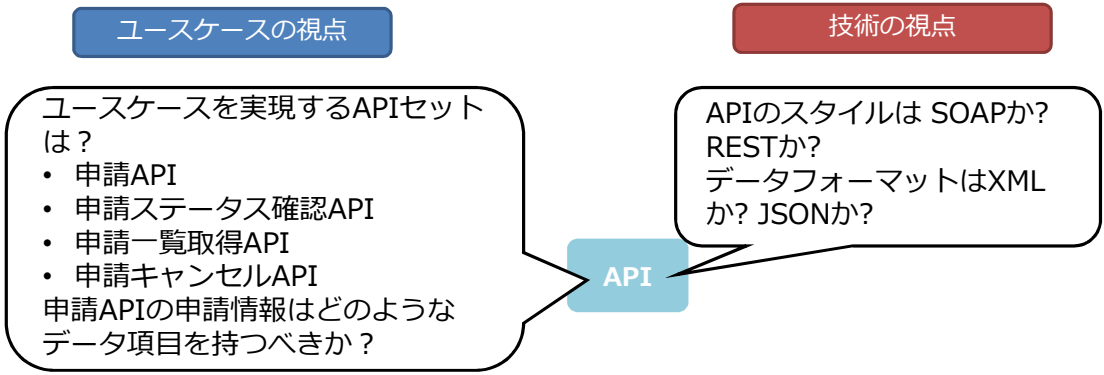
- APIライフサイクルの設計で重要になるポイント
  - APIを利用するアプリ、システムを含めた全体のアーキテクチャ
    → セキュリティ、スケーラビリティ、拡張性、コストを考慮する
  - APIのユースケースを実現するデータセット、インタフェース
    → ユーザ視点のデータセット、標準的なAPIスタイルなどユーザの利用しやすさを考慮する



アーキテクチャ設計

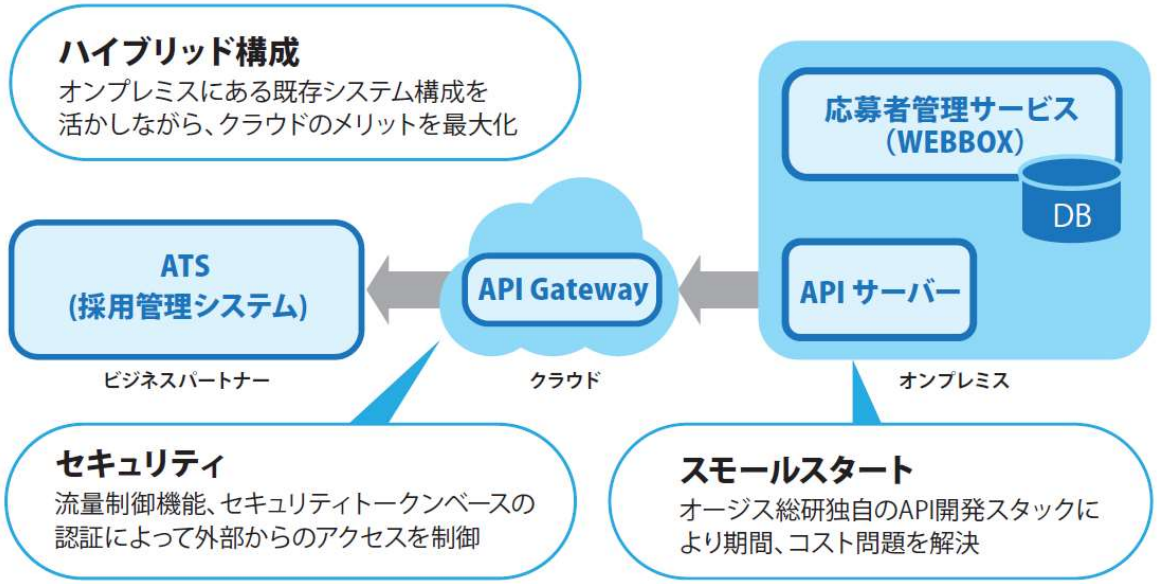


インタフェース設計



事例：パーソルキャリア様APIアーキテクチャ

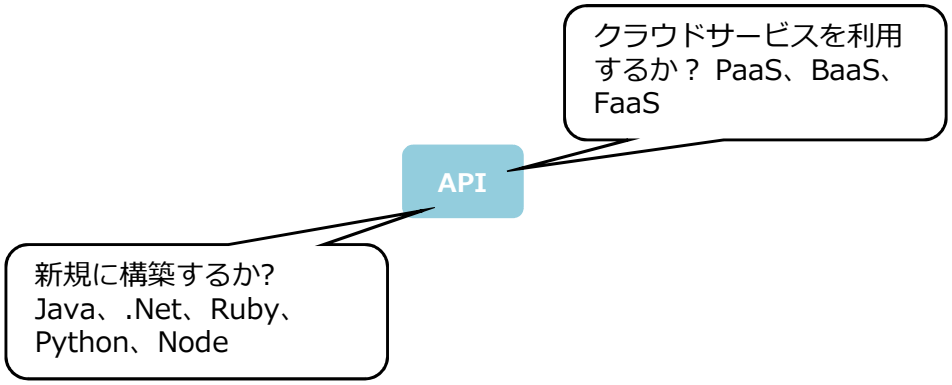
- パーソルキャリア株式会社(旧名:株式会社インテリジェンス)様
- アルバイト求人情報サービス「an」の応募情報へのアクセスをAPI化し、法人サービスの利便性を向上



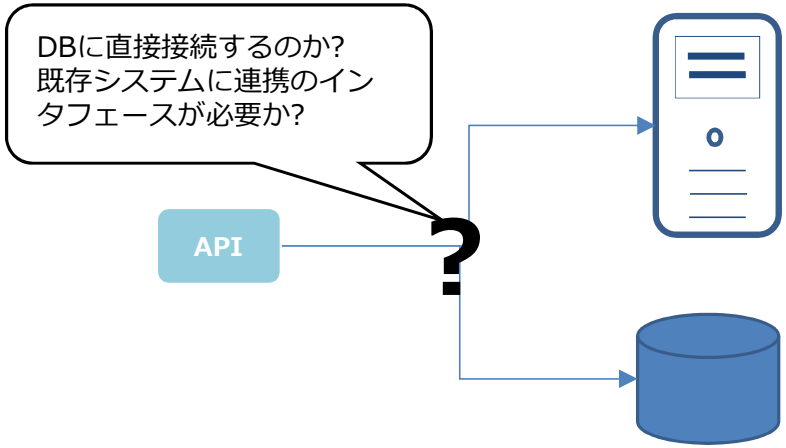
API公開のライフサイクル:開発

- API公開の開発で重要になるポイント
  - APIの開発方法
    - → 新規にAPIを構築、既存システムを拡張、ESB/EAIなどの連携ミドルウェアの利用、クラウドサービスの利用
  - 既存システムとの連携
    - → API公開したいデータを持つシステムと、どのようにつなぐか

何をつかって開発する？



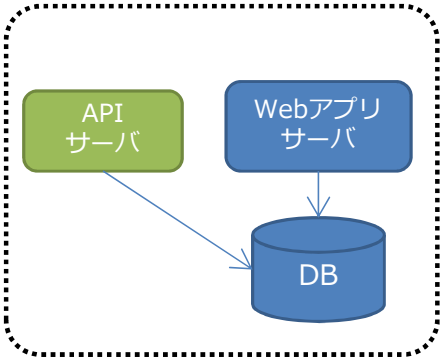
どうやって連携する？



事例:パーソルキャリア様

- APIをどのように実装するか。
  - **既存DBを利用しAPIを新規開発**
- APIの内部処理
  - HTTP リクエスト/レスポンスのハンドリング
  - メッセージバリデーション
  - データベースアクセス
  - キャッシュ
  - 参照系APIのポイント
    → フィルター、ソート、ページネーション
  - 更新系APIのポイント
    → 再送信対策

既存のWebアプリへの影響をできるだけ小さくするために、別途APIサーバを立てる形でAPIを実装

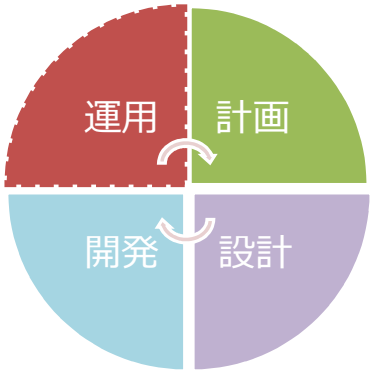


API公開のライフサイクル:運用

オージス総研

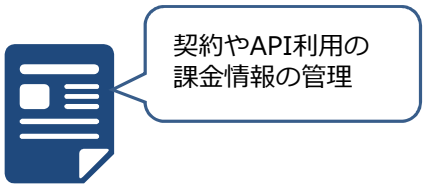
API公開の運用で重要になるポイント

- APIのバージョン、リリース管理
- APIのユーザ、契約管理
- APIのユーザ向けサポート
- APIの監視、障害対応



APIの機能追加やデータ項目
変更などの管理する

	バージョン	ライフサイクル
ユーザプロファイル 変更API	1.2	公開中
サービス取得API	0.1	開発中



オージス総研

APIの認証・認可

APIの認証・認可の種類 1

種類	説明
X509クライアント証明書	APIサーバが発行した秘密鍵でリクエストに署名を付けて呼び出す。 例. 電力小売スイッチングAPI
API Key （ユニーク文字列）	APIサーバが発行するユニークな文字列をリクエスト(ヘッダー、クエリー、ボディ等)に含めて呼び出す。 例. Google Map API
リクエスト署名 (HMAC、HKDF)	API Key(id+secret)を利用してリクエストの署名を作成し、その署名をリクエストに含めて呼び出す。リクエストの改ざん防止、リクエストの有効期限の設定など目的がある。 例. AWS API(Amazon Signature v4)
Basic認証	リクエストのAuthorization ヘッダーにユーザ:パスワードをBase64エンコードして呼び出す。 例. Github API(GithubはOAuth推奨だがBasic認証もサポート)

OpenAPI-Specification 3.0の Security Scheme を参考に作成

APIの認証・認可の種類 2

詳細	説明
Bearer(JWT)	RFC6750 に定めされている Bearer トークンを汎用的なHTTP 認証・認可に用いる。よく使われるトークンはJWT(Json Web Token)である
OAuth2 Authorization Code	クライアントは認可サーバがクライアントとリソースオーナー(エンドユーザー)の仲介となって発行する認可コードを利用し、アクセストークンを取得し、アクセストークンを含めてリクエストを行う 例. Google AdWords APIなどの様々なWeb API
OAuth2 Implicit	Authorization Codeフローを単純化したものであり、認可コードの代わりに直接アクセストークンを取得する 例. Fitbit API などの様々なWeb API
OAuth2 RO Password	クライアントはリソースオーナーのパスワードクレデンシャルを使ってアクセストークンを取得する。リソースオーナーとクライアントの間で高い信頼があるケースで利用することが推奨されている
OAuth2 Client Credentials	クライアントのクレデンシャルを使ってアクセストークンを取得する。対象のリソースがクライアントの管理対象であるケースで利用する

OpenAPI-Specification 3.0の Security Scheme を参考に作成

何を使えばよいのか?

- 提供しているAPIは、クライアントに対してアクセス制御したいのか、リソースオーナー(エンドユーザー)に対してアクセス制御したいのか。

クライアントに対するアクセス制御

ユーザに依存しない特定クライアント向けのデータや機能
例えば、取引先のシステムに提供する、カタログ情報参照API

エンドユーザーに対するアクセス制御

ユーザーが承認したデータや機能
例えば、サードパーティアプリに提供する、口座情報参照API

何を使えばよいのか?

- クライアントに対してアクセス制御するケースでは、セキュリティレベルや実装コストのバランスを検討し、選択する必要がある。

種類	標準仕様	トークン/クレデンシャル有効期限	リクエストの改ざん防止	スコープによるアクセス制御
X509クライアント証明書	RFC3647	有り	有り	なし
API Key（ユニーク文字列)	なし	実装によるがないケースが多い	ない(SSL/HTTPSと組み合わせて実現)	なし
リクエスト署名(HMAC、HKDF)	なし	実装によるがリクエストの有効期限が設定される	有り	なし
Basic認証	RFC7235	ない	ない(SSL/HTTPSと組み合わせて実現)	なし
OAuth2 Client Credentials	RFC6749	有り	ない(SSL/HTTPSと組み合わせて実現)	有り

何を使えばよいのか?

- エンドユーザーに対してアクセス制御するケースでは、OAuth2 がデファクトスタンダードである。クライアントの種類や必要なセキュリティレベルに応じて、プロファイルを選択する必要がある。

#	ユーザーへの提供形態	バックエンド	OIDC/OAuthのフロー	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 Flow	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 Flow	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 Flow	（① と同じ）
④		なし	Authz Code Flow 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 Flow	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。

モバイル、オープンAPIを活用するための認証基盤、より引用

Scopeとは

- 認可サーバから発行されたアクセストークンでアクセスできる範囲を表す。基本的にスコープ名には何を使っても良いが、リソース名とアクション名を使うケースが多い

- 例. Facebook
  - read_insights
  - user_actions.news
- 例. Github
  - read:org
  - write:org

- 例. Google
  - https://www.googleapis.com/auth/gmail.readonly
  - https://www.googleapis.com/auth/gmail.send

OAuth2 認証・認可サーバの実装と運用

オージス総研

- APIの認証・認可の仕組みとしてOAuth2は様々なケースに対応可能なフレームワークであるが、その実装と運用は大変

大量の関連仕様

RFC6749 OAuth2.0
RFC6750 Bearer Token
RFC7009 Token Revocation
RFC7636 PKCE
RFC7662 Token Introspection
RFC7515 JWS
RFC7518 JWT

などなど

どこまで対応が必要？
正しく実装できるか？

管理・運用

- ユーザのクレデンシャルの管理
- 開発者の管理
- クライアントの管理
- トークンの管理
- スコープの管理

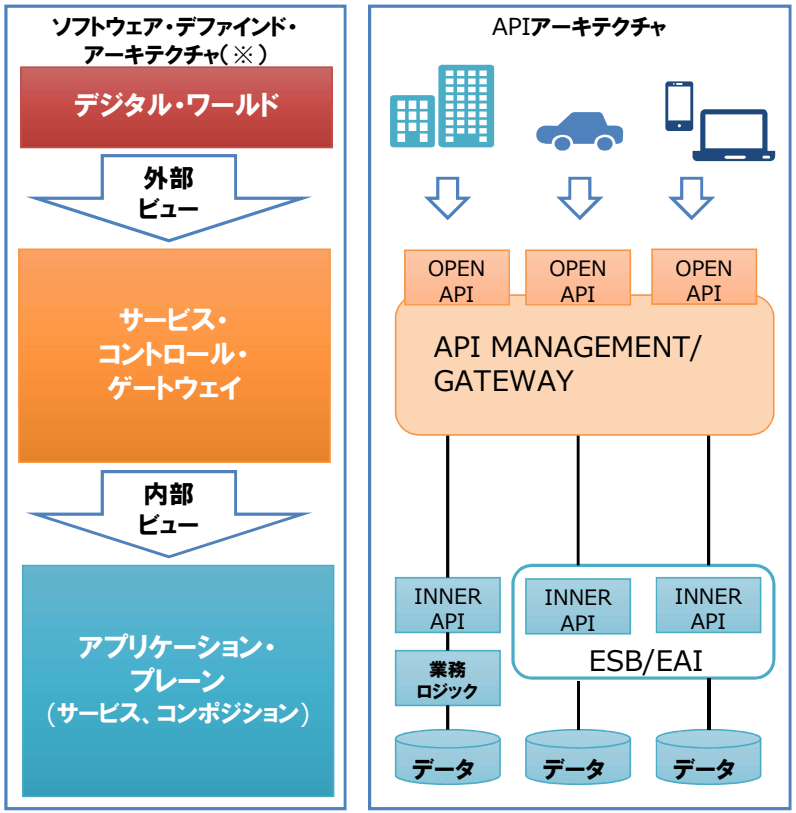
これらの項目の管理はツール/サービスのサポートは必須

オージス総研

API公開プラットフォームの活用

APIアーキテクチャのベストプラクティス

□ APIに共通的に必要になる機能を集約し、効率的にAPI公開を実現する。



(※) 出典「デジタル・ビジネスのアプリケーション向けソフトウェア・デファインド・アーキテクチャ/Gartner」

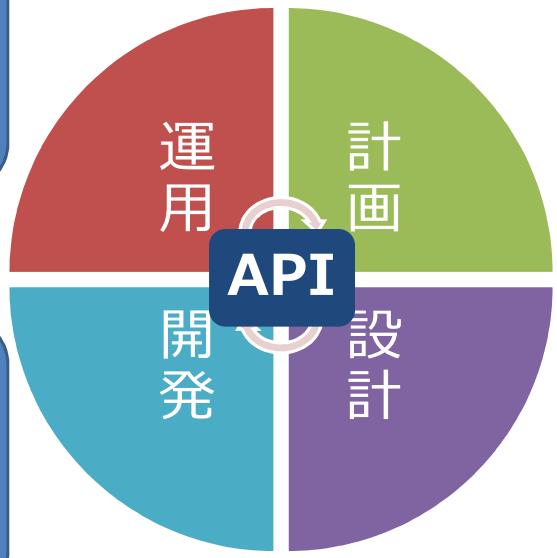
- デジタルワールド
  - モノ、アプリケーション(Web/モバイル)、サービス
- サービス・コントロール・ゲートウェイ
  - API MANAGEMENT/GATEWAY
    - API公開に必要な認証/認可、流量制御、バージョン管理、モニタリングなどの機能を追加
  - OPEN API
    - モノ、アプリケーション、サービスより、利用可能なAPI
    - 外部とのビジネスの変化に柔軟に対応する
- アプリケーション・プレーン
  - 内部システム(組織内/F/W内)より、再利用可能な業務ロジック+データをマイクロサービス(API)化
  - 自社のビジネスの変化に柔軟に対応する

APIの公開のライフサイクルを支援するAPI管理製品

- APIの公開のライフサイクルの中の設計、開発、運用をより効果的に実践できる。

運用を効率的に実行できる

高速にAPIの開発行うことができる



設計のベストプラクティスを適用できる

APIゲートウェイ/管理製品

オージス総研

- API公開(特にオープンAPI)に必要な機能を提供するプラットフォーム

提供機能

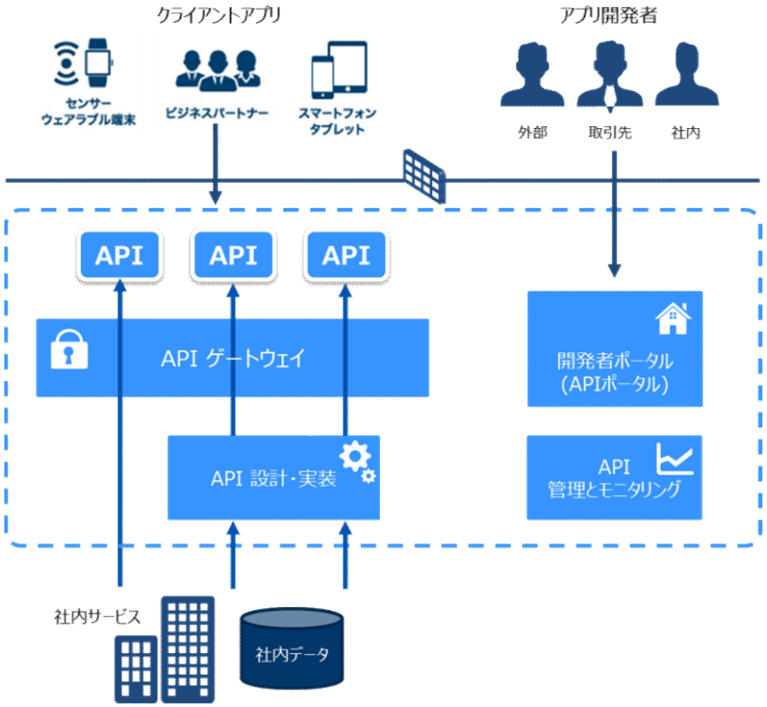
- APIゲートウェイ(プロキシ)
  - セキュリティゲートウェイ、流量制御
- API管理機能
  - （認証・認可や流量制御設定等）
  - モニタリング（ログ、アクセス統計情報等）
- API設計・実装支援ツール
- 開発者ポータル

代表的なサービス/ソフトウェア

- クラウドサービス
  - Amazon API Gateway
  - Azure API Management
- パッケージ/ハイブリッド
  - IBM API Connect
  - Apigee Edge
  - CA API Management
  - Mulesoft Anypoint Platform
- OSS(オープンソースソフトウェア)
  - Kong
  - Zuul

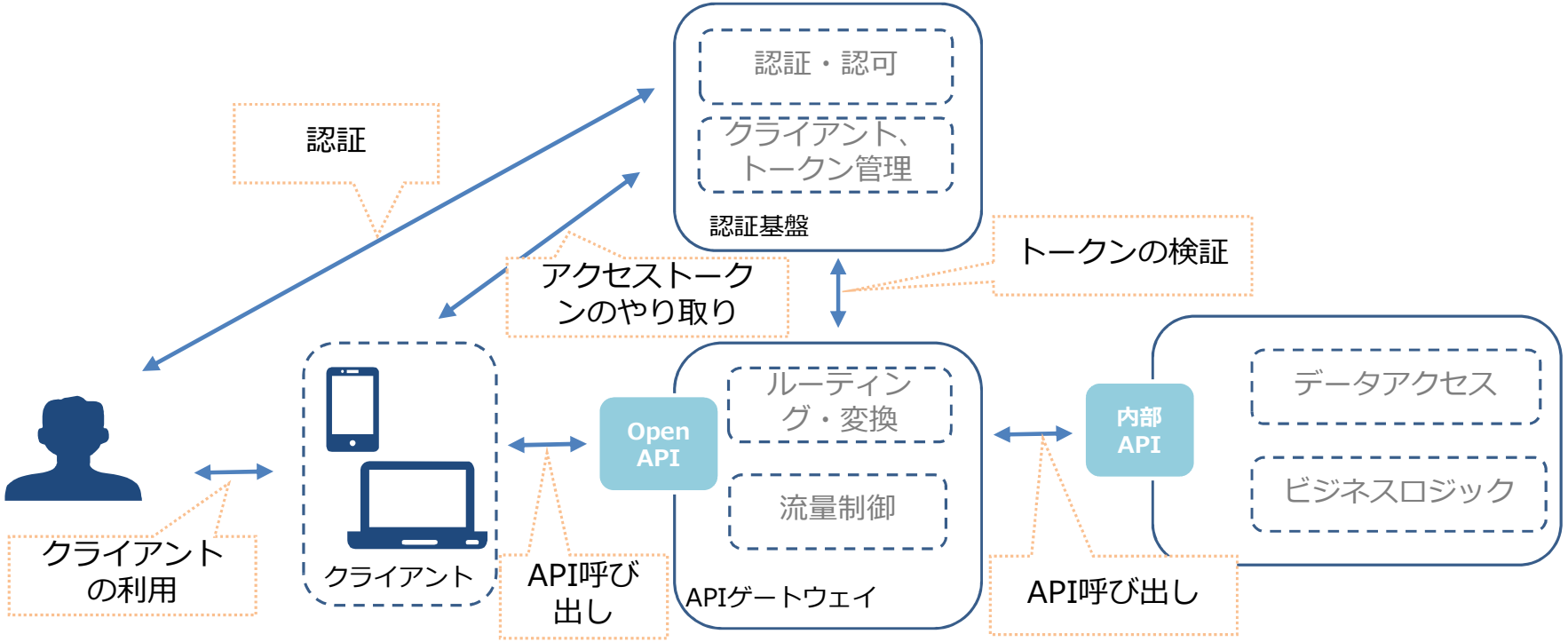
API管理製品選定のポイント

- API管理製品：個々のAPI実装から、共通に管理すべき機能・非機能要件を分離
- 公開しているAPIの全貌を一括して管理可能
  - API管理製品の標準的な構成
    - → APIゲートウェイ
    - → API管理機能（認証・認可や流量制御設定等）と モニタリング（ログ、アクセス統計情報等）
    - → API設計・実装
    - → 開発者ポータル
- 製品選定のポイント
  - ゲートウェイで必要最小限のアクセス管理をするか？
  - API管理製品（スイート）でライフサイクル全体を管理するか？
  - クラウドサービスかオンプレか？
  - 社内既存システムとの連携の重要度は？
  - API実装フレームワークを製品で共通化するか？



APIゲートウェイ/管理製品+認証基盤

- APIゲートウェイ/管理製品はそれ自体にAPIの認証・認可機能を持っているが、OAuh2/OpenId Connectへの対応が部分的であり、求める要件に対して十分でないケースがある。その場合、認証基盤と組み合わせて実現する。

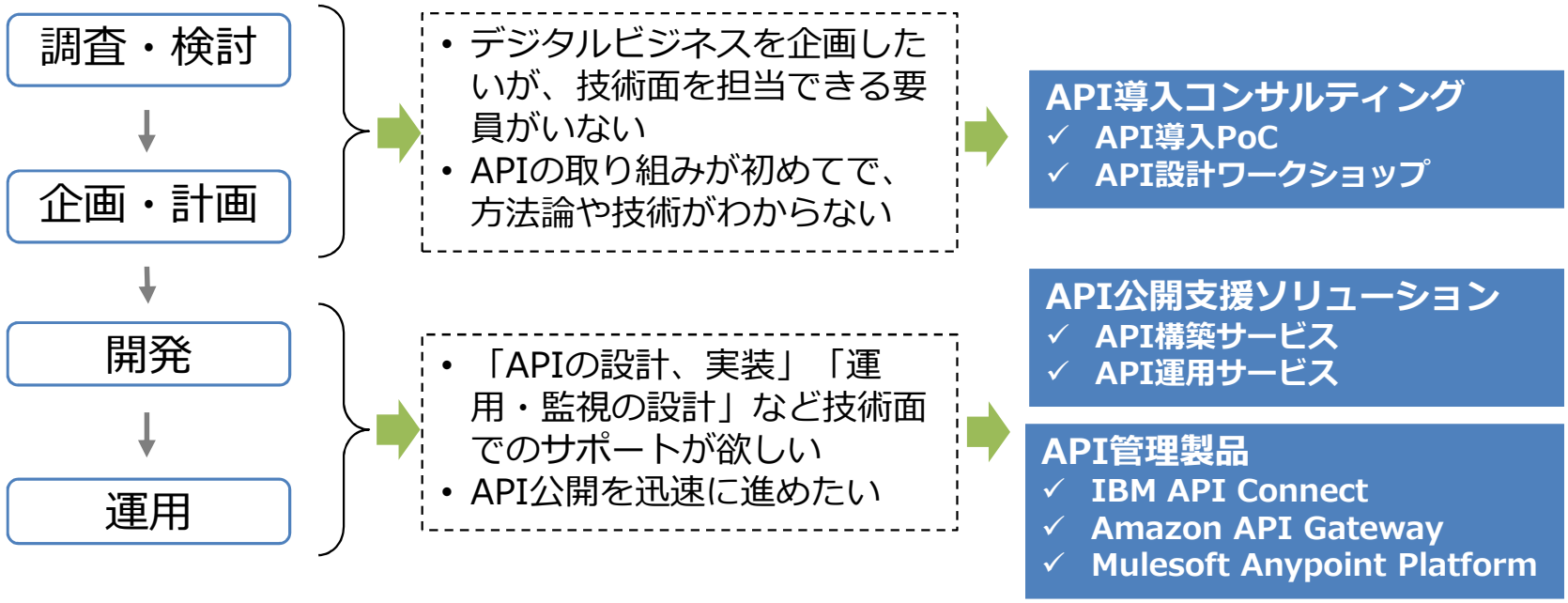


オージス総研

オージス総研のAPI公開支援ソリューション

オージス総研APIソリューションの全体像

- 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する



まとめ

オージス総研

- APIエコノミーとは、API自体がビジネス価値を持ち、他社と**つながる**ことで生まれる新しいサービスやイノベーションを生みだすことである

- API公開のライフサイクルのポイント
  - 計画：APIの目的および**ロードマップ**を明確にする
  - 設計：**API Gatewayパターン**のアーキテクチャ、使いやすい標準的なAPI仕様
  - 開発：開発、テスト、デプロイの効率的な実践
  - 運用：APIと利用者の管理モデル

- APIの認証・認可
  - **認証・認可の主体**がクライアントなのか、エンドユーザーなのかによって認証認可モデルが異なる
  - エンドユーザーを対象にした認証・認可は**OAuth2が標準**である
  - OAuth2の実装や運用は大変

- APIプラットフォーム
  - 設計のベストプラクティス
  - 管理を効率化するためのツール
  - **API管理+認証基盤**でセキュアなAPIプラットフォームを実現できる