

事例で学ぶ、 「API公開と運用」の基本

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)

株式会社オービス総研

- **代表者：** 代表取締役社長 西岡 信也
- **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
- **主な事業所**
 - 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 港区港南2-15-1 品川インターシティA棟
 - 千里オフィス： 大阪府 豊中市新千里西町1-2-1
 - 名古屋オフィス： 愛知県 名古屋市中区錦1-17-13 名興ビル
- **売上実績：** 642.8億円（連結） 371.3億円（単体） （2016年度）
- **従業員数：** 3,284名（連結） 1,407名（単体）
- **関連会社：** さくら情報システム（株）、（株）宇部情報システム、（株）システムアンサー、
OGIS International, Inc.、上海欧計斯软件有限公司（中国）
- **オービス総研グループ[®] 売上構成比（連結）**



取得許可認定



- 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



- API公開プロセス
- API運用の課題
- API管理製品の活用
- オージス総研のAPI公開ソリューション

API公開プロセス

□ 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

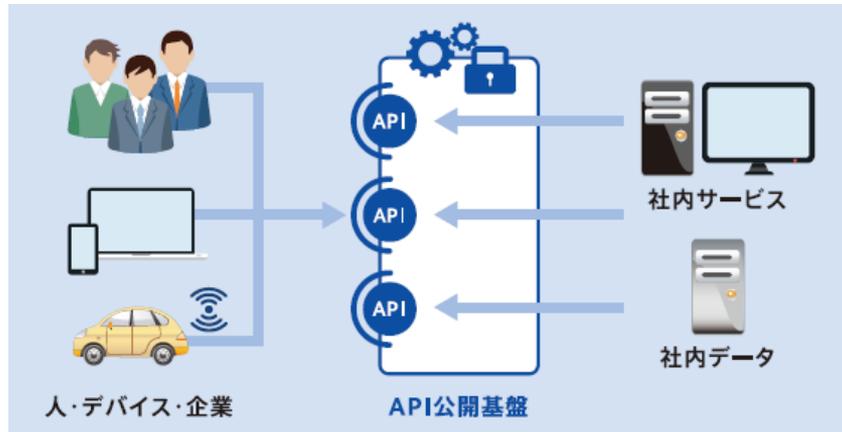
損害保険 × 自動車

- 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

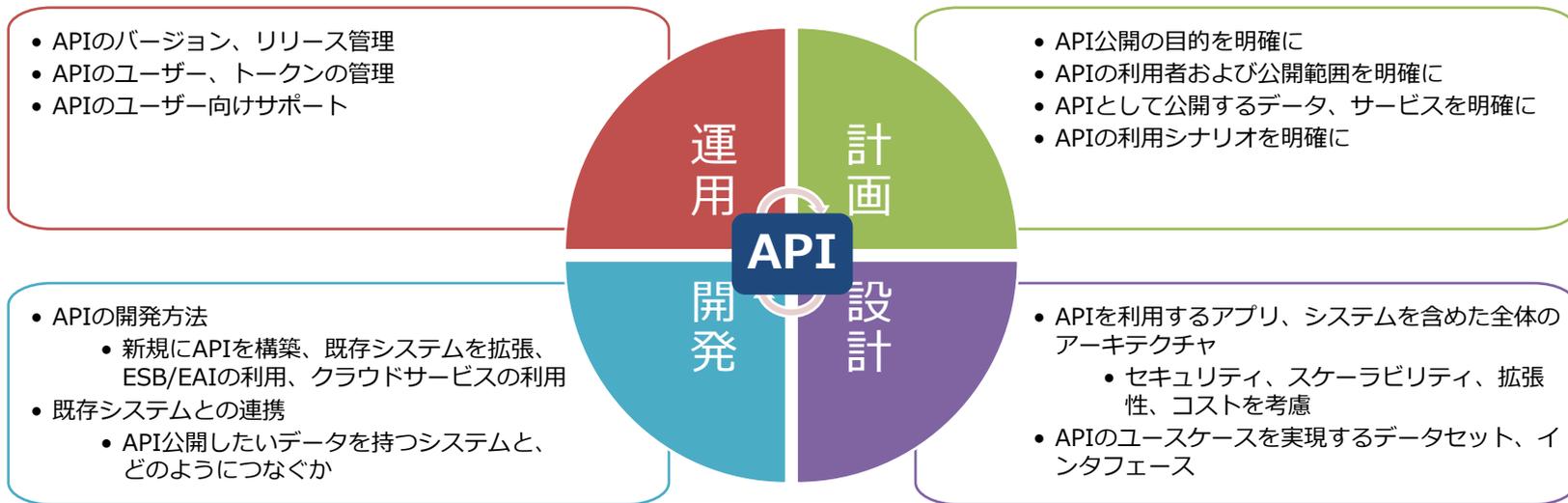
□ これら新しいビジネスの背後には「API」を通じたサービス、データの連携がある

□ 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

- 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

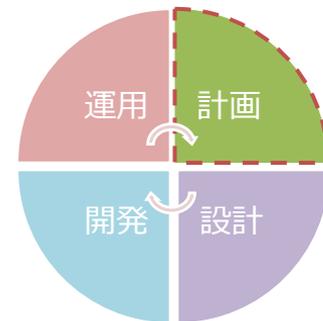


- API公開は、一度きりの取り組みではない
- デジタルビジネスの成長、変化にあわせAPIを改修し、バージョンアップすることが必要
→ ライフサイクルをしっかりと回していくことが重要



□ 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。

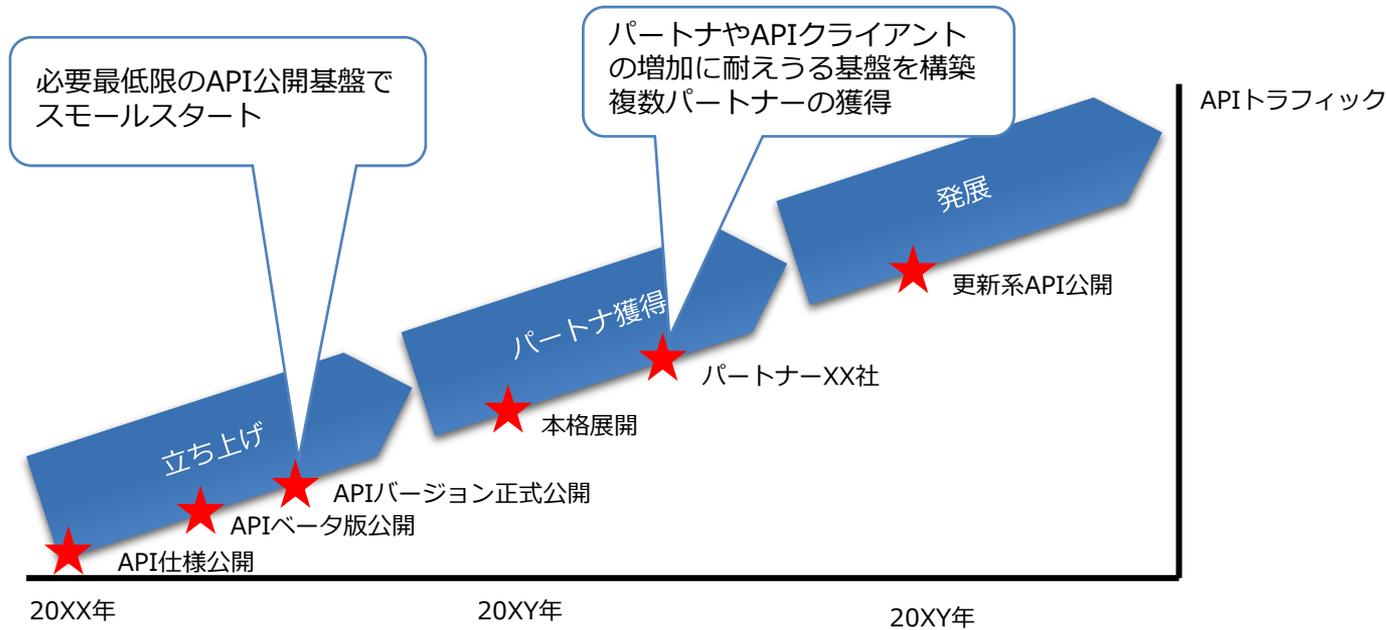
例：取引先、代理店向けのカタログAPI

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。

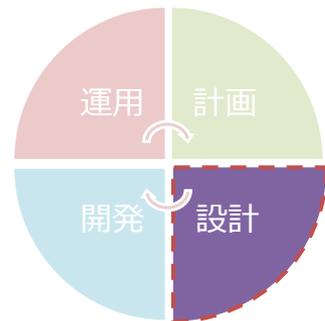
例：オープンに公開されているMap API

- 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

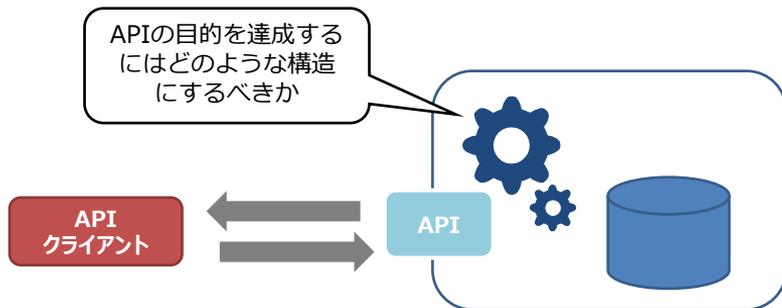


□ API公開の設計で重要になるポイント

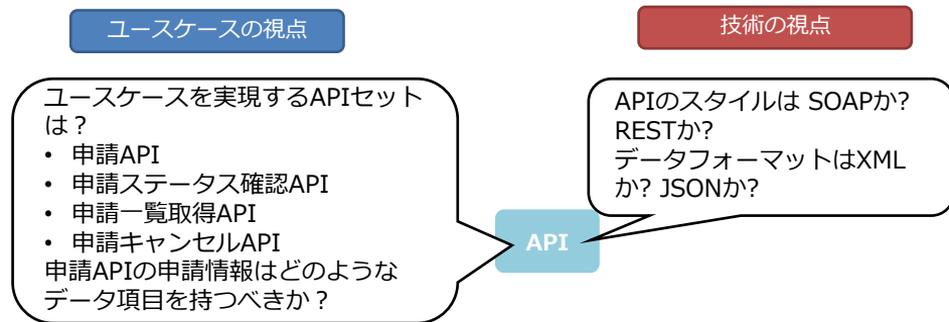
- APIを利用するアプリ、システムを含めた全体のアーキテクチャ
→ セキュリティ、スケーラビリティ、拡張性、コストを考慮する
- APIのユースケースを実現するデータセット、インタフェース
→ ユーザ視点のデータセット、標準的なAPIスタイルなどユーザの利用しやすさを考慮する



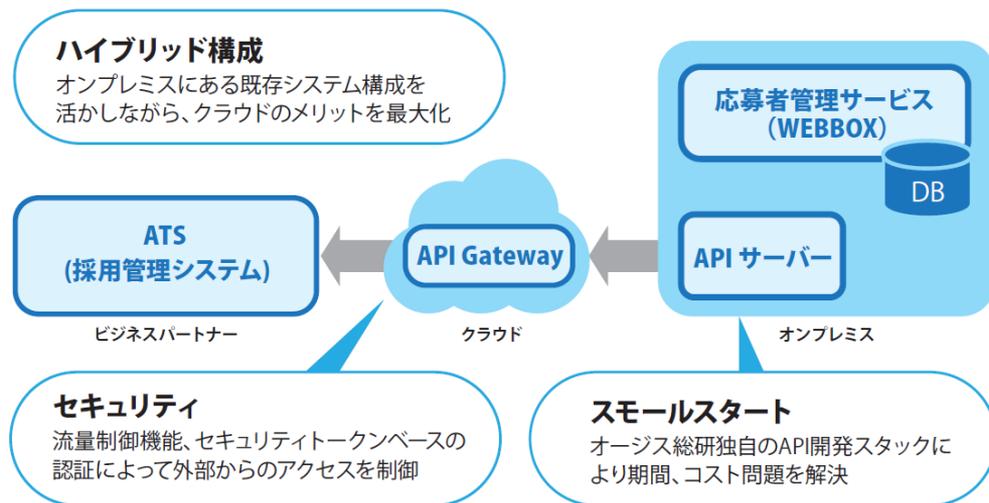
アーキテクチャ設計



インタフェース設計



- 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



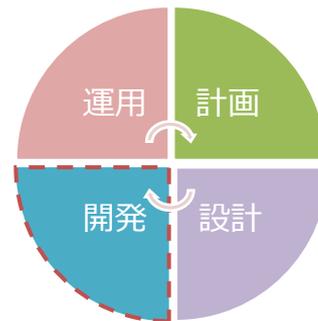
□ API公開の開発で重要になるポイント

- APIの開発方法

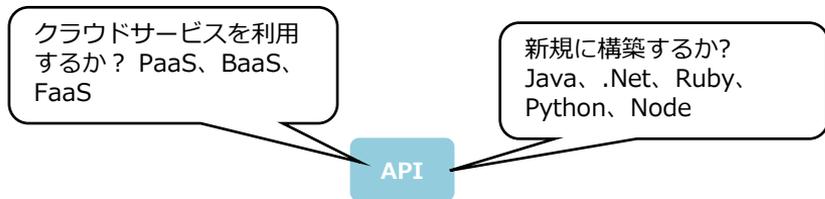
→ 新規にAPIを構築、既存システムを拡張、ESB/EAIなどの連携ミドルウェアの利用、クラウドサービスの利用

- 既存システムとの連携

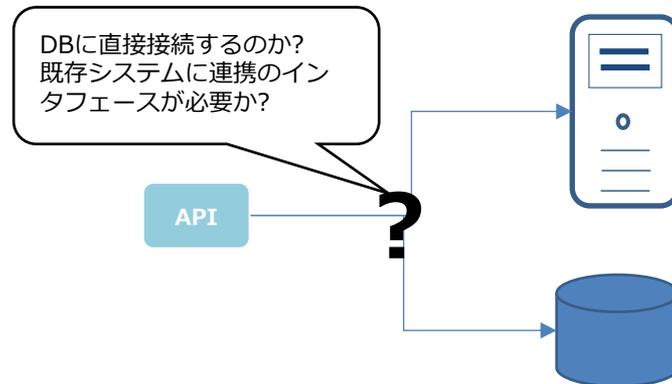
→ API公開したいデータを持つシステムと、どのようにつなぐか



何をつかって開発する？

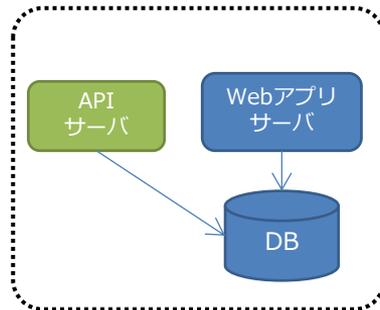


どうやって連携する？



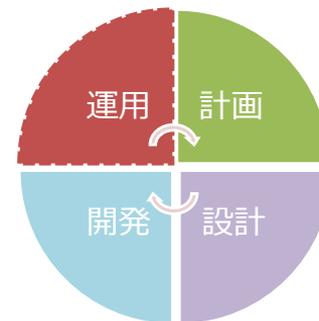
- APIをどのように実装するか。
 - 既存DBを利用しAPIを新規開発
- APIの内部処理
 - HTTP リクエスト/レスポンスのハンドリング
 - メッセージバリデーション
 - データベースアクセス
 - キャッシュ
 - 参照系APIのポイント
 - フィルター、ソート、ページネーション
 - 更新系APIのポイント
 - 再送信対策

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
ユーザプロフィール変更API	1.2	公開中
サービス取得API	0.1	開発中



ユーザ

APIユーザや管理
APIを利用するために
トークンの管理



API利用契約

契約やAPI利用の
課金情報の管理

APIの使い方を理解するための
ドキュメント、SDK



開発者

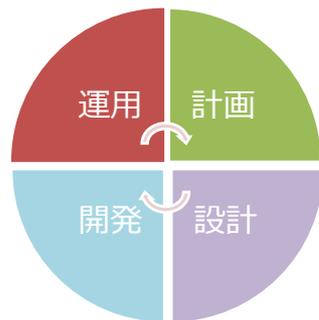
開発支援



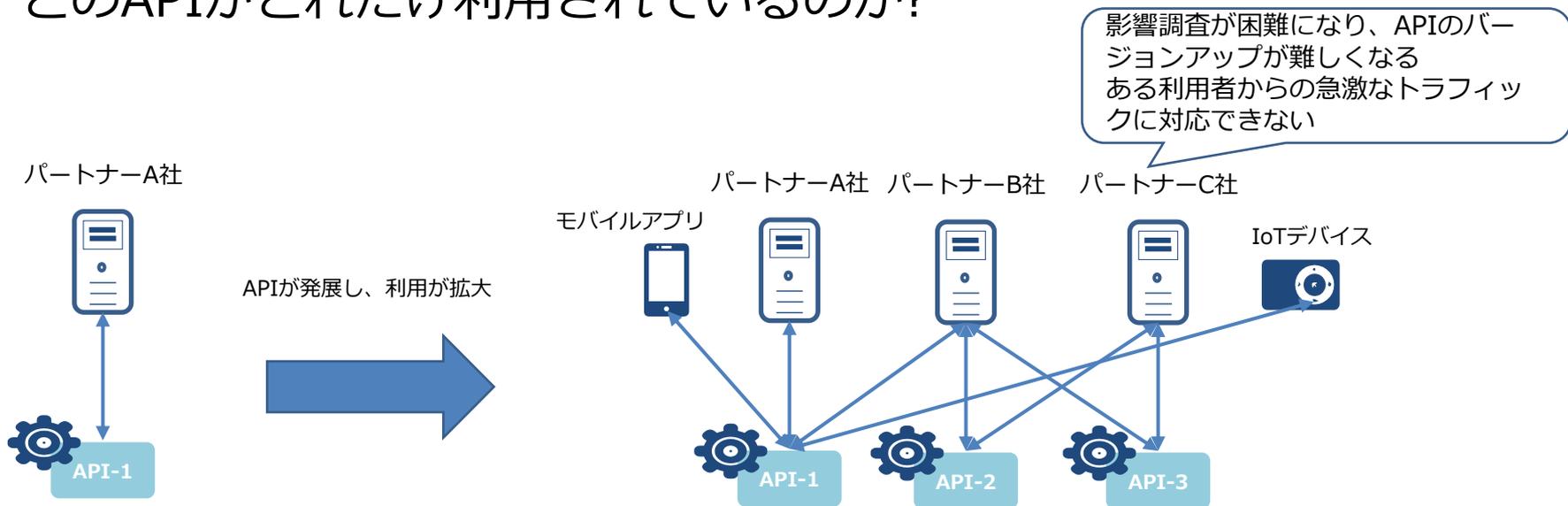
No	ユースケース	開発・運用切り分け		オペレーションマニュアル作成対象
		開発	運用	
1	公開APIの開発環境を作成する		○	○
2	公開APIの本番環境を作成する		○	○
3	内部APIをリリースする（バックエンドAPIのプロキシ）	○		
4	公開APIを新規作成する（パラメータ変換処理等を含む）	○		
5	公開APIを更新する（パラメータ変換処理等を含む）	○		
6	公開APIをリリースする		○	○
7	公開APIを利用会社に通知する		○	○
8	公開APIの利用状況を確認する		○	○
9	公開エンドポイントを死活監視する		○	
10	公開APIのパフォーマンスを監視する		○	
11	ユーザ(APIキーも含む)・ロールを追加・変更する		○	○
13	障害対応を行う（ログ取得、サポート問合せ）		○	○
13	設定情報をバックアップする		○	○

API公開の課題

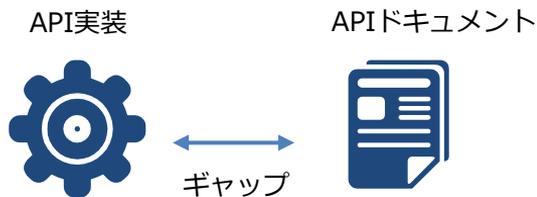
- APIは新しいビジネスや新しいパートナーに向けて公開されるため、常に変化が求められる
- APIは提供サービス・データの更新や接続先の増加に合わせて、変更や追加が発生する
 - 平均すると2ヶ月に1回ぐらいの頻度でAPIの変更や追加が発生(当社実施した案件の実績値より)
 - 短い場合は、1週間でAPIの改修、本番デプロイを実施
- いかに早くサイクルを回していくか、が大事になる



- APIは誰(どのAPIクライアント)から利用されているのか？
- どのAPIがどれだけ利用されているのか？



- APIを利用する開発者向けにAPIの仕様をドキュメントとして提供しなければならない
- WordやHTMLなどの静的なドキュメントでは、変化の多いAPI開発に追従させなければならないが、APIの実装と仕様の乖離が発生してしまうことがある



- APIを利用する開発者に向けて分かりやすい標準的な仕様にすることが重要になる。独自仕様は開発者の混乱を招き、API利用が進まない原因になる
- REST/JSONスタイルが採用されることが多い
 - RESTスタイルのメリットは一貫性と予測可能性(URIとHTTPメソッドからある程度何ができるAPIなのか類推できる)

REST/JSON スタイル

REST: 基本的な考えはHTTPの原理。URIはリソースを表す名詞

GET http://domain/api/items 商品一覧取得

POST http://domain/api/items 商品登録

JSON: シンプルで相互運用性が高い

```
{"user" : [  
  { "name" : "saito", "age" : "32" },  
  { "name" : "yamada", "age" : "25" },  
  { "name" : "kimura", "age" : "41" }  
]}
```

技術的な標準仕様

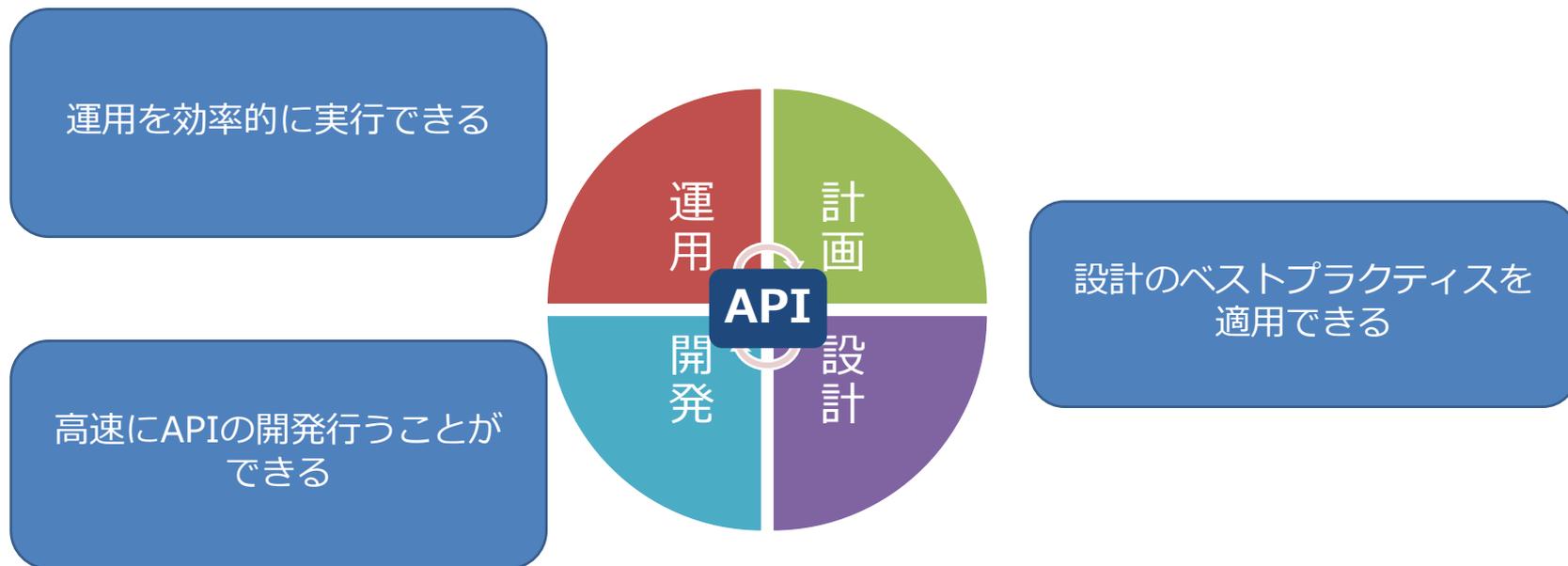
- OpenAPI Specification

業界向け標準仕様

- [金融] Fintech共通API、Open Banking Standard
- [スマートハウス] HEMSデータ利活用業者間API標準

API管理製品の活用

- API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



□ 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

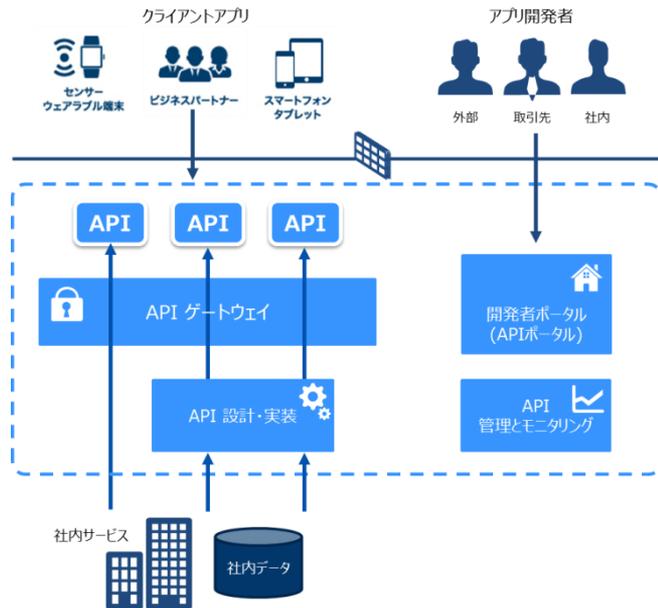
- ゲートウェイで必要最小限のアクセス管理をするか？

- 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？



作成

APIを楽々開発

- 既存のデータストアやサービスから容易にAPI開発を行えるツール
- ビルド、テスト、デプロイ
- StrongLoop のNode.jsを提供

実行

APIを安定・高速稼動

- APIの実行環境の管理
- 監視、スケーリング
- 膨大なAPIが稼動可能

万全のセキュリティ

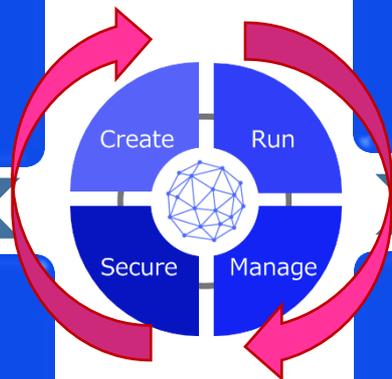
- 実績多数の堅牢なゲートウェイ
- API利用者の認証とアクセス制御
- アクセス数の流量制限

保護

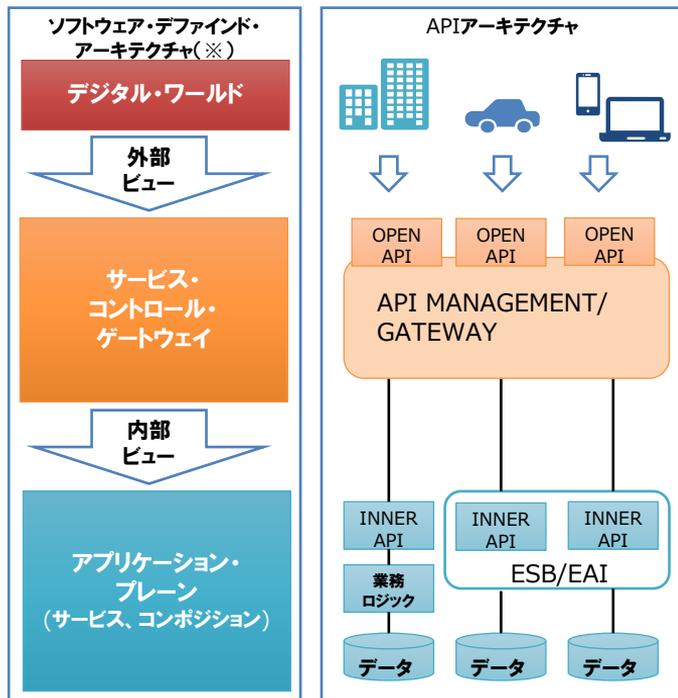
緻密な利用管理

- 公開APIの容易なポリシー定義
- 利用状況の分析、課金
- 開発者ポータルへの自動連携

管理



□ 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の開発

- HTTPリクエストレスポンスのハンドリング
- データリソースへのアクセス
- バリデーション
- 認証・認可
- ログ
- キャッシュ

□ APIのテスト

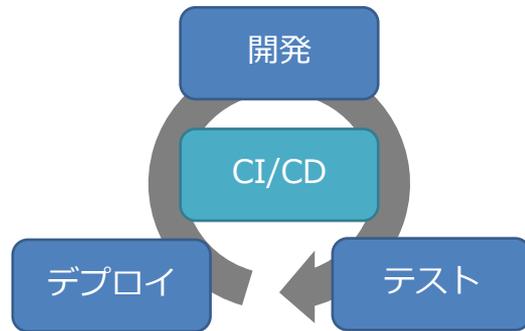
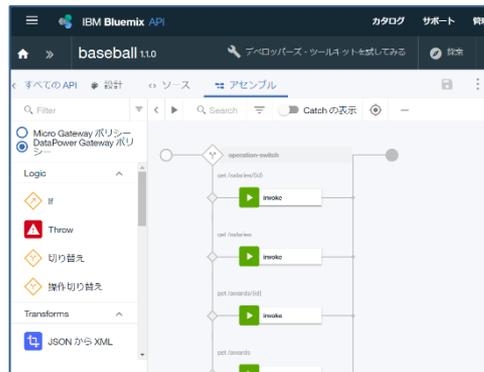
- HTTPリクエストを送信し、実際にAPIを叩くテスト

□ APIのデプロイ

- APIを実行環境に配置する

□ API Tool Kit

- API Designer
 - API をグラフィカルに開発
- API CLT
 - CI/CDに組み込みやすいツール



□ APIのグルーピング

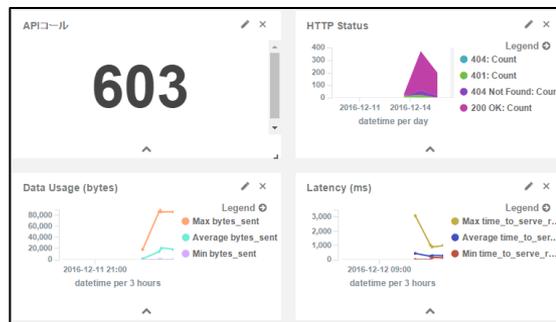
- APIのバージョン
- 環境(本番、開発)
- 利用プラン(無料、エンタープライズ)

□ APIランタイムの管理項目

- 死活監視
- トラフィック量
- ターンアラウンドタイム

□ API利用者の管理項目

- API開発者
- 開発者の組織
- APIクライアント
- APIキー



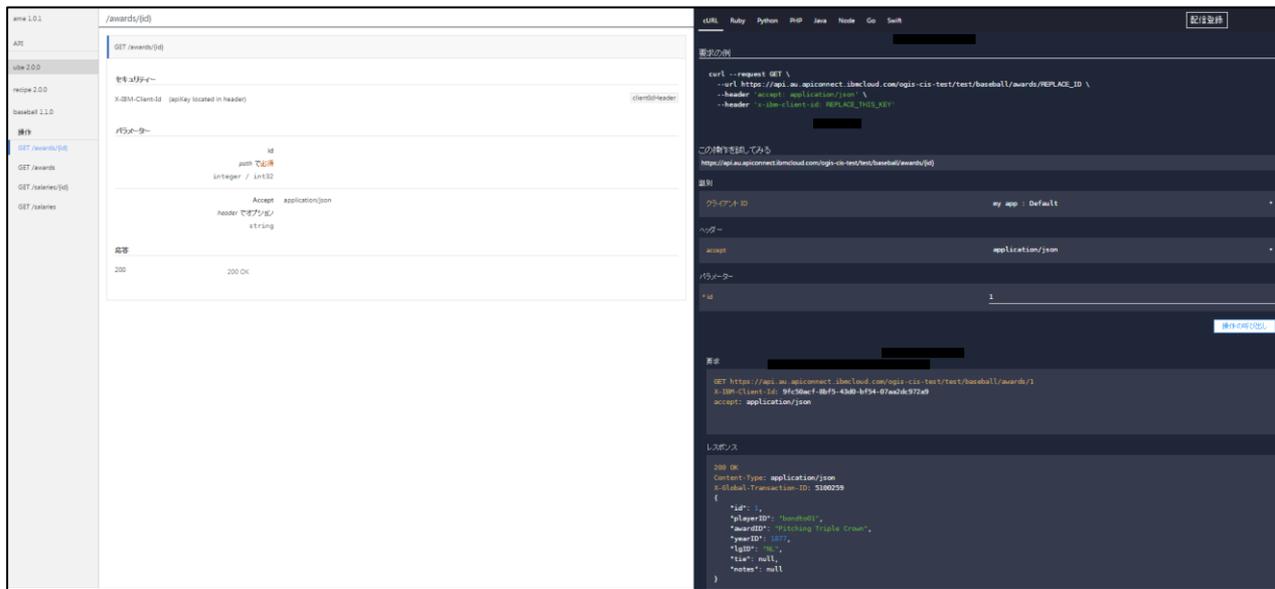
API Analytics



APIプランの管理

- APIをデプロイする環境を論理的に分離
 - 例1: 開発、ステージング、本番
 - 例2: API仕様のバージョン
- APIをグルーピング
 - 製品としてパッケージ化
 - クライアント登録、APIキー管理を製品単位でおこなう
- 契約条件ごとに機能レベルを変更
 - 例: APIコール回数制限あり、なし

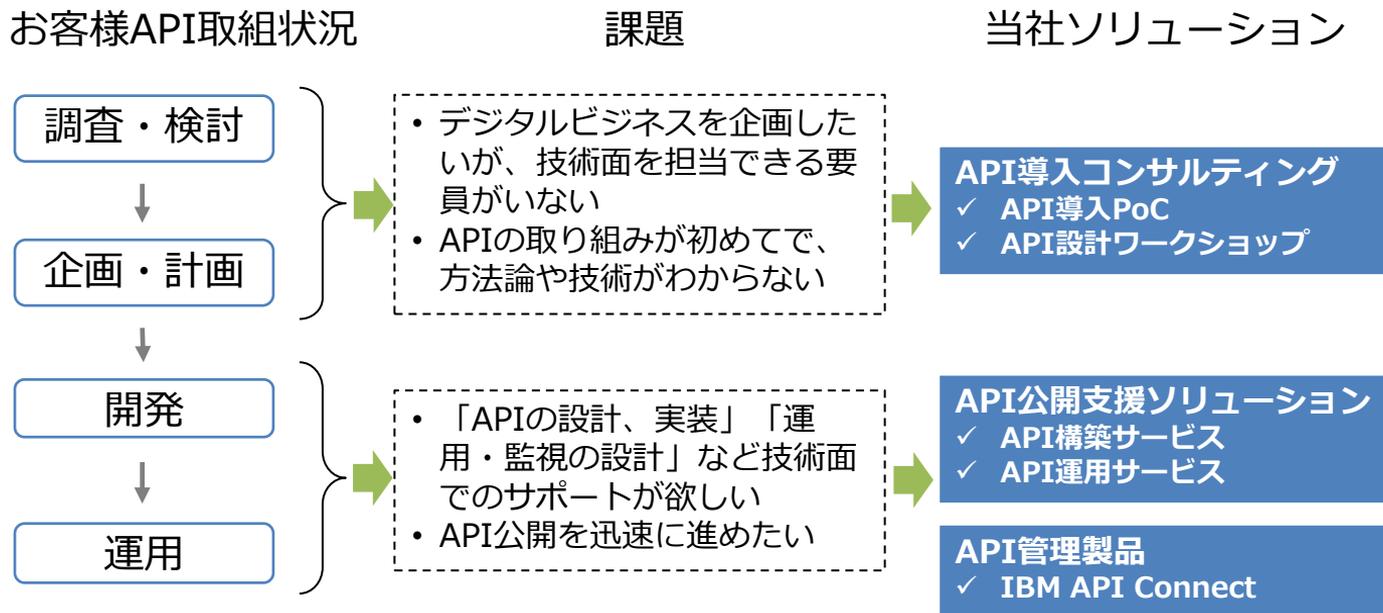
- API定義言語 Open API Spec(Swagger)を利用することで、インタフェース実装とドキュメントがリンクする
- APIのドキュメントを管理者ポータルで提供することで、APIコールも試すことができる



The screenshot displays an API management interface. On the left, a sidebar shows a list of endpoints, with 'GET /awards/0/' selected. The main area shows the endpoint details, including the method (GET), path (/awards/0/), and parameters (id: integer / int32). Below this, the '実行' (Execute) button is visible. On the right, the '実行結果' (Execution Results) panel shows the request details, including the URL (https://api.ogis-ri.com/ogis-cis-test/test/baseball/awards/0/), headers (X-IBM-Client-Id, X-IBM-Client-Id: REPLACE_THIS_KEY), and the response body (application/json). The response body is a JSON object with fields like id, name, and address.

オージス総研のAPI公開支援 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する



「API抽出とKPI設定」 ⇒ 「PoC実施」 ⇒ 「API評価、運用」
3ステップで、API導入の実践・評価

1. API抽出とKPI設定

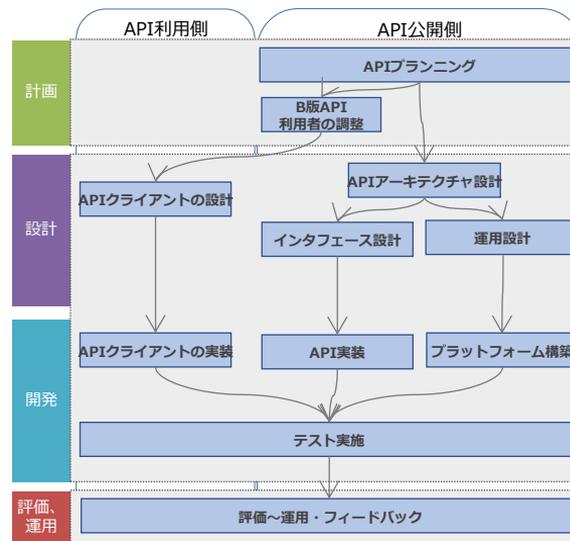
- ビジネスの現状と将来像を考慮したビジネスモデリング、API抽出、KPIの設定を実施します

2. PoC実施(概念検証)

- 最も重要なAPIを選定し、PoCを実施します

3. API評価、運用

- PoCの結果を評価し、API導入推進の意思決定を支援します
- また、PoC後の本格展開とともに必要となる追加APIの開発やプラットフォームの拡張も承ります



APIの公開に必要なベストプラクティスを体系的に会得できる

ベストプラクティスの知識だけでなく、実践的に考えるプロセスを学べる

講義資料例

どのように設計するのか

設計のポイント

- APIゲートウェイを使うか
 - 流量制御やキャッシュが必要か
 - 複数のAPIサービスが存在するか
- 開発者ポータルを使うか
 - クライアントやAPIキー管理をAPI提供者がおこなえるか
 - APIドキュメントを提供する方法があるか
 - 多数の開発者に使ってもらいたい
- APIゲートウェイでオーケストレーションや変換をすべきか
 - 一般的にはNo!
 - APIサービスの変更がAPIゲートウェイに影響する
 - テストしにくい
 - リリース時期やコストを優先するのであれば、負債になる覚悟を持つ

ワークショップ内容

イントロダクション	
ワークショップの説明	
参加者の自己紹介	
講義	
APIアーキテクチャをどのように設計していくのか	
APIアーキテクチャの構成要素、典型的なパターン	
APIの特徴を知る	
演習	
課題の説明	
要件の仮定	グループでアーキテクチャを検討しながら、明示されていない要件を仮定する
アーキテクチャ設計	グループでアーキテクチャ設計をする
振り返り・ディスカッション	

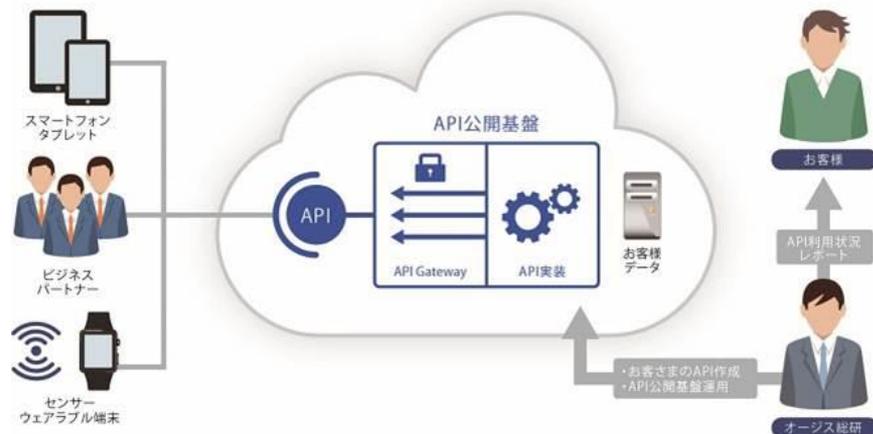
ワークショップの課題設定や内容はお客様の要望に合わせてカスタマイズ可能です。

基盤・作業をパッケージ化

- API公開をはじめる際に必要となるAPIインフラの用意、API作成作業、その後のシステム運用をまとめてご提供します

ビジネス規模に対応

- 公開するAPIの数、トラフィック量などの増大に応じたシステム拡張が可能、APIビジネスの規模に応じたスモールスタートに適しています



- API構築サービス

APIプラットフォーム構築	クラウド、ハイブリッド、オンプレミスに対応 要件にマッチした製品/サービスを利用してご提供
API開発	API開発のベストプラクティスを詰め込んだAPI開発スタック を使い、効率的にAPIを開発

- API運用サービス

APIアップデート	APIのデータ項目の追加などAPIのアップデート、リリースの実施
APIプラットフォームメンテナンス	定期的なAPIプラットフォームのセキュリティアップデート
APIプラットフォーム監視	APIプラットフォームの障害・異常検知および通知
API障害分析、対応	障害発生時の原因調査、切り分けおよび復旧対応
API利用状況レポート	APIの利用状況のレポートニング
APIユーザサポート	API利用者への問い合わせ対応、APIクライアント開発支援

- 様々なAPI案件を積み重ね得られたAPI公開プロセスのノウハウとプロセスを効率的に実践できるAPI管理製品により、お客様のAPI公開を実現いたします



API公開プロセス

+



IBM API Connect

- API公開を成功させるには**変化への素早い対応**が重要になる
- 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、デプロイを効率的な実践
 - 運用：APIと利用者の管理モデル
- API管理製品はAPI公開の**近道**
 - 設計のベストプラクティス
 - 生産性の高い開発ツール
 - 管理を効率化するためのツール

株式会社オービス総研

営業本部営業企画部

【 TEL 】 03-6712-1201（東京） / 052-209-9390（名古屋） / 06-6871-8054（大阪）

【 E-mail 】 info@ogis-ri.co.jp

【 URL 】 <http://www.ogis-ri.co.jp>