

デジタル田園都市国家構想実現のための生活用データ連携基盤推奨モジュール  
拡充・運用・自治体への運用支援等事業

## 推奨モジュールの役割と活用例

2024年10月

一般社団法人データ社会推進協議会



## 1. 都市OSの特徴とコアとなるシステム要素

- 用語解説
- 都市OSと推奨モジュールの関係

## 2. 都市OSの成長ステップとデータ連携基盤の基本構成

- スマートシティの成長に合わせた都市OSの成長ステップ
- 都市OSのコア要素（推奨モジュール）
- 非パーソナル向けデータ連携基盤で利用されるAPIとデータモデル
- パーソナル向けデータ連携基盤で利用されるAPIとデータモデル

## 3. データ連携基盤の構成例と機能要件例

- データ連携基盤構成例の表記
- 構成例(非パーソナル)
- 構成例(パーソナル)
- 非パーソナルとパーソナル両方適用の例

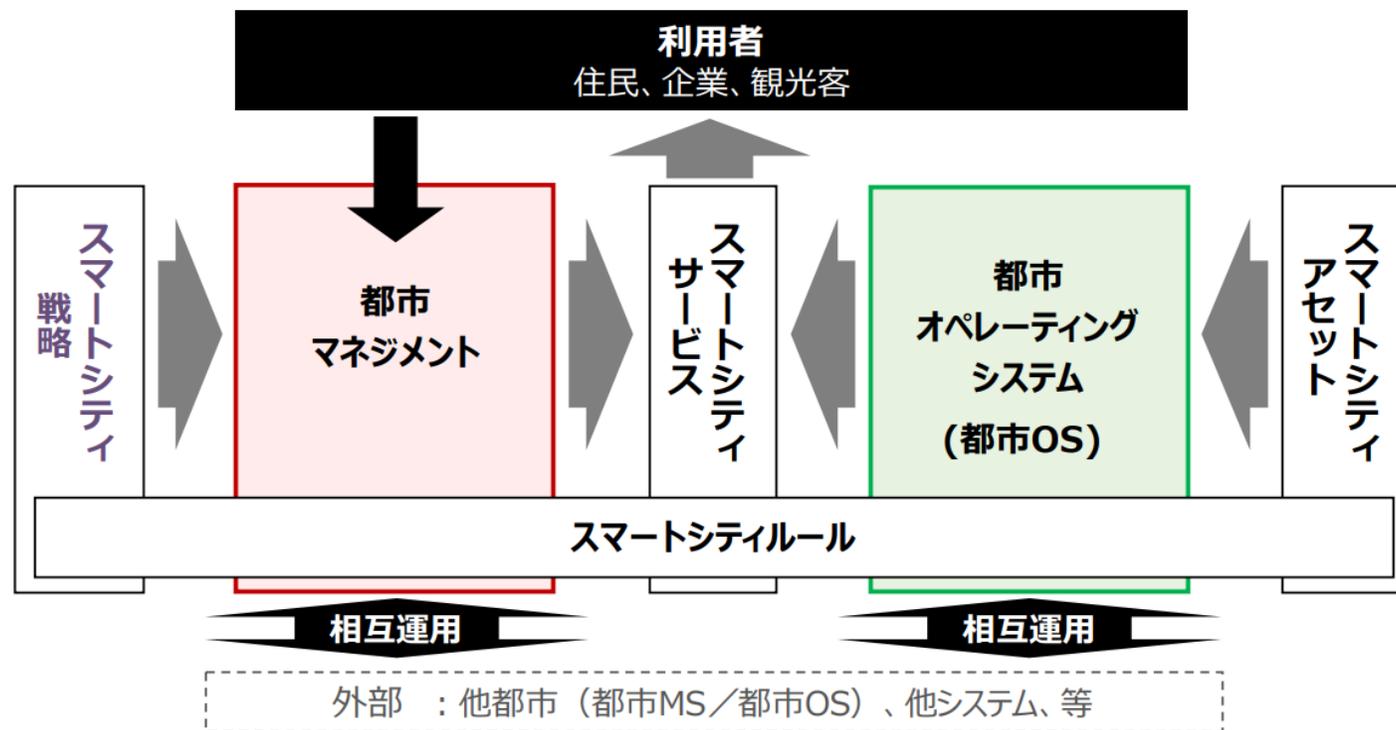
## ● Appendix

- FIWARE Orionのデータ管理について
- Kong Gateway:APIゲートウェイの推奨モジュール
- パーソナルデータ連携モジュールの概要と公開情報

# 1. 都市OSの特徴とコアとなるシステム要素

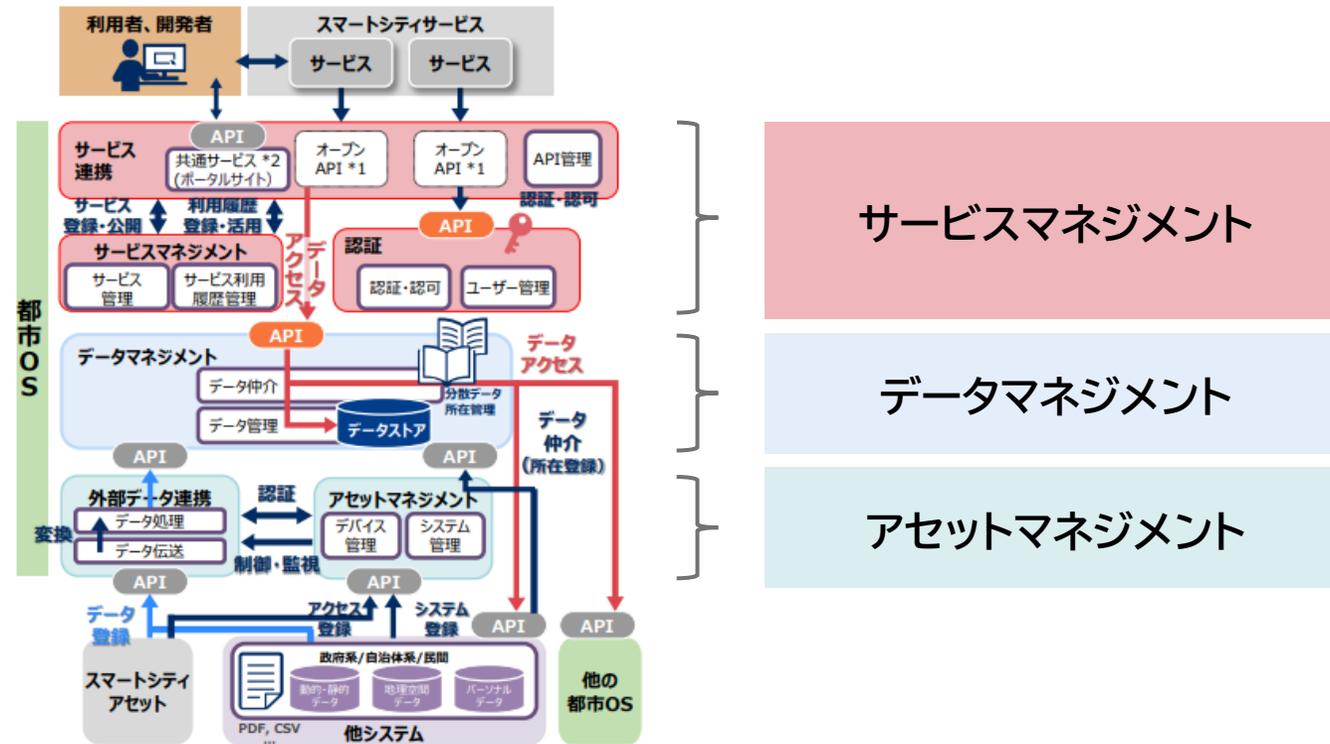
# スマートシティリファレンスアーキテクチャとは

- スマートシティリファレンスアーキテクチャは、都市の課題解決を検討する際に共通指針として参照すべきアーキテクチャ(設計思想、設計方法、実現方法)のこと。



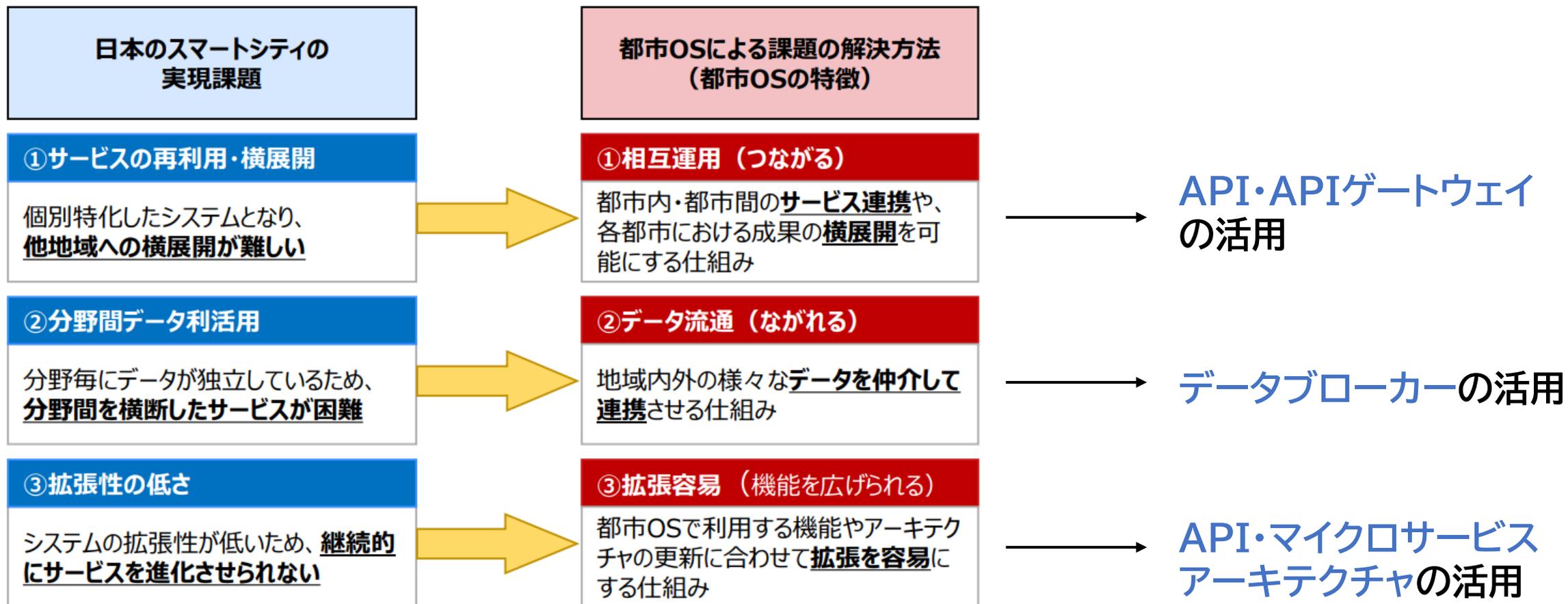
“スマートシティリファレンスアーキテクチャの使い方(導入ガイドブック)(第2版)” (p. 19).  
[https://www8.cao.go.jp/cstp/society5\\_0/smartcity/architecture.html](https://www8.cao.go.jp/cstp/society5_0/smartcity/architecture.html)

- 都市OSは、スマートシティの実現を目指す地域が運用するITシステムの総称。防災などのデータを収集/分析し、都市内外のサービスと連携する役割を担う。
- 主にサービス・データ・アセットの3つを管理することでサービスを提供している。



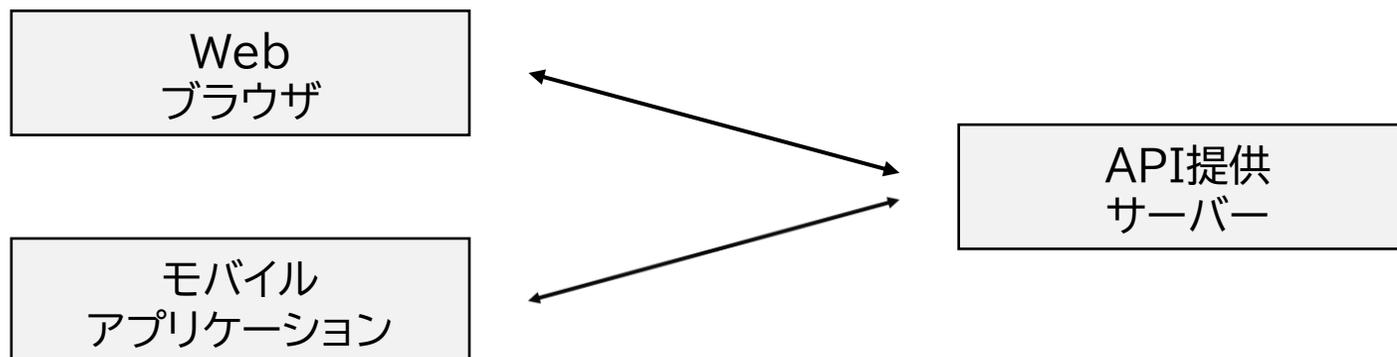
“スマートシティリファレンスアーキテクチャの使い方(導入ガイドブック)(第2版)” (p. 25).  
[https://www8.cao.go.jp/cstp/society5\\_0/smartcity/architecture.html](https://www8.cao.go.jp/cstp/society5_0/smartcity/architecture.html)

# 都市OSの特徴を実現するための技術要素



“スマートシティリファレンスアーキテクチャの使い方(導入ガイドブック)(第2版)” (p. 22).  
[https://www8.cao.go.jp/cstp/society5\\_0/smartcity/architecture.html](https://www8.cao.go.jp/cstp/society5_0/smartcity/architecture.html)

- APIは、プログラムから機能を利用するためのインターフェースの総称。  
APIのうち、特にWebを介して利用するものはWeb APIと呼ばれ、主にインターネット経由で接続可能なWebサーバーが提供している。



API : Application Programming Interface

# データブローカーとは

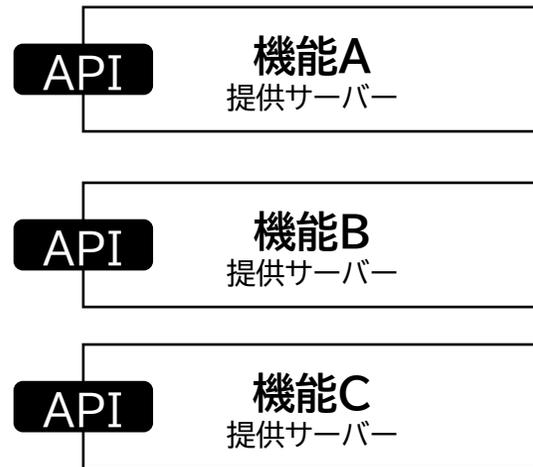
- データブローカーは、データ自体またはデータの所在情報を収集・管理し、これらを利用する内外のシステムにAPIを提供する役割を担う。
- 利活用したいデータの所有権によって2種類のブローカーを使い分けることが重要。



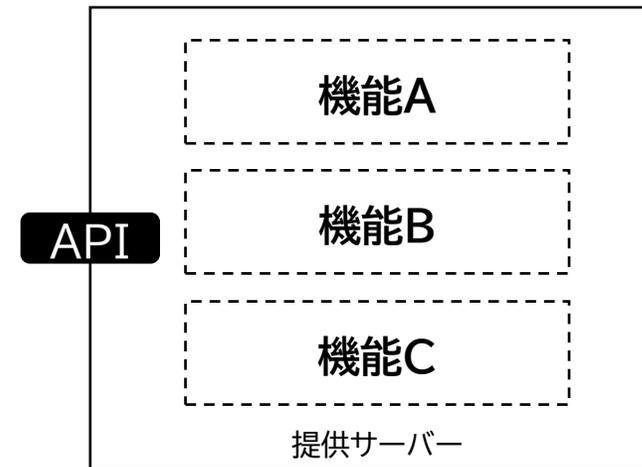
オープンデータの管理 : 非パーソナル用ブローカー  
 プライベートデータの管理 : パーソナル用ブローカー

# マイクロサービスアーキテクチャ（ビルディングブロック方式）とは

- マイクロサービスアーキテクチャは、アプリケーションを独立して動作する小さな機能ごとに分割し、これらを組み合わせて1つのシステムとして構築する手法のこと。これに対して複数の機能を一つのコンポーネントでまとめて構築する手法はモノリシックサービスと呼ばれる。
- マイクロサービスは、モノリシックサービスと比べて機能追加などの変更にも強く、状況に応じた柔軟なシステム運用が可能となる。



マイクロサービス



モノリシックサービス

# APIゲートウェイとは

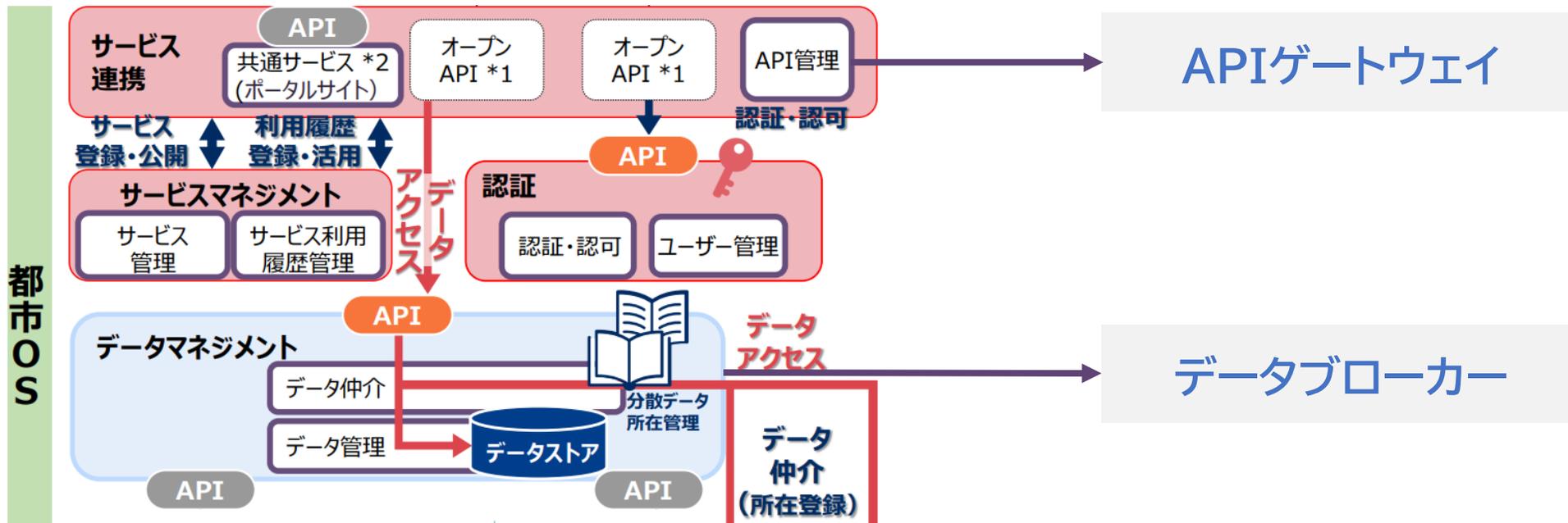
- APIゲートウェイは、マイクロサービスとして提供されたAPIを管理し、クライアントに提供する役割を担う（サービスと利用者の管理）。
- ブローカーはAPIゲートウェイで管理可能なAPI提供サーバーの一つで、都市OSではセットで利用することが一般的である。



データブローカーは、登録したデータ・データの所在情報を管理するAPI提供サーバーの一種。

# 都市OSと推奨モジュールの関係

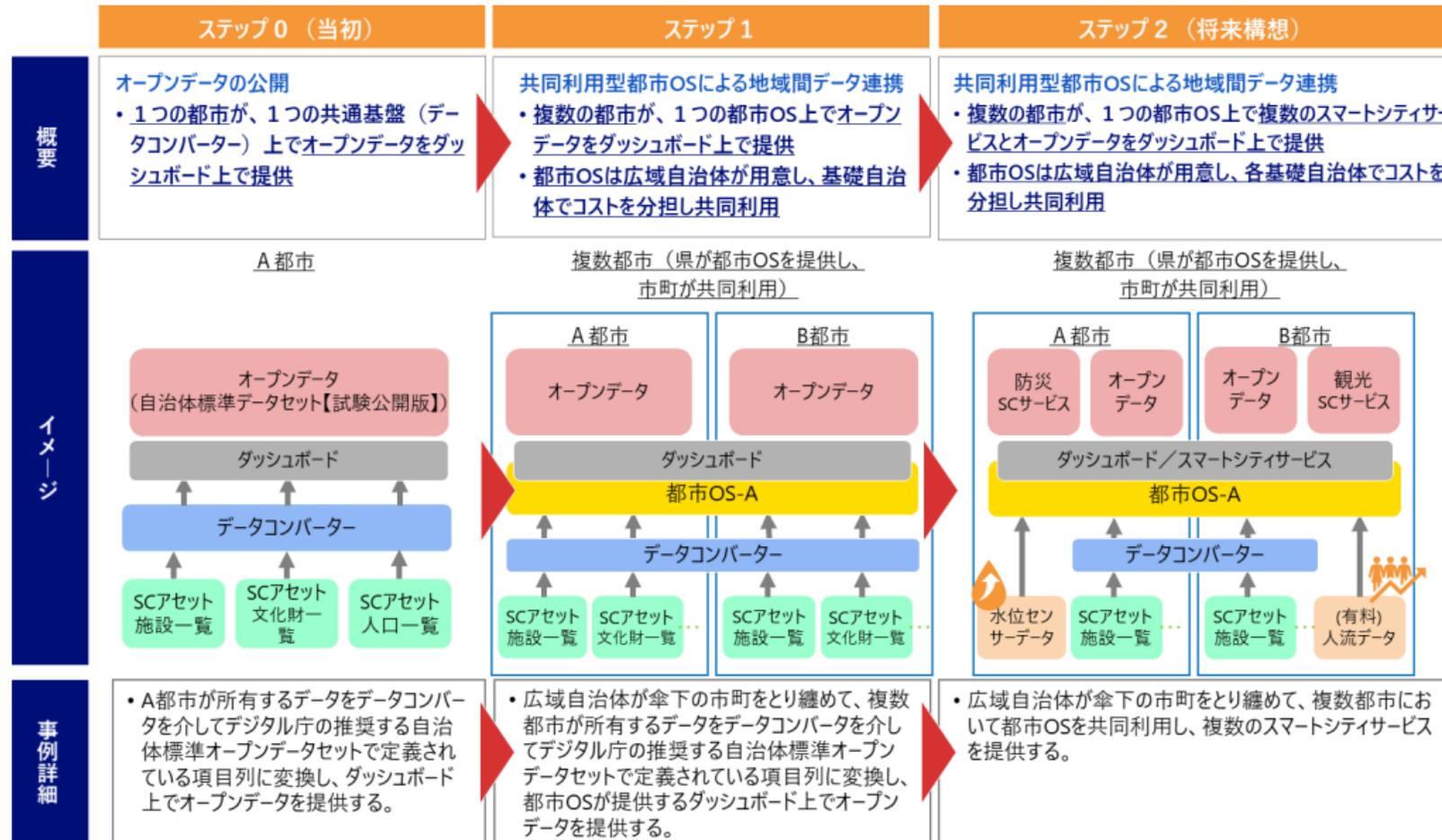
- 都市OSの構築で最低限必要となるコアモジュール(APIゲートウェイ、データブローカー)は推奨モジュールと定義されている。
- 「どんなデータを利活用したいのか」によって必要機能の拡充を検討することが重要である。



“スマートシティリファレンスアーキテクチャの使い方(導入ガイドブック)(第2版)” (p. 25).  
[https://www8.cao.go.jp/cstp/society5\\_0/smartcity/architecture.html](https://www8.cao.go.jp/cstp/society5_0/smartcity/architecture.html)

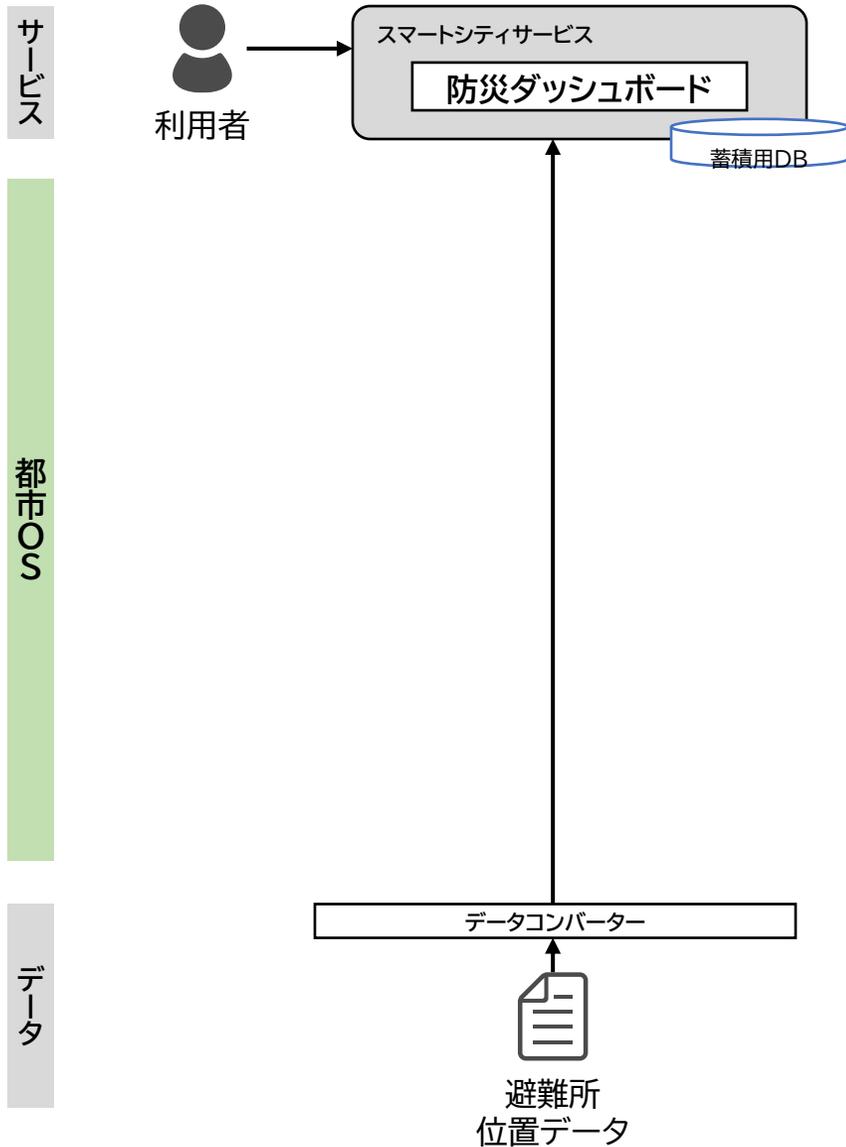
## 2. 都市OSの成長ステップとデータ連携基盤の基本構成

# スマートシティの成長に合わせた都市OSの成長ステップの例



“スマートシティリファレンスアーキテクチャ(ホワイトペーパー)(第2版)” (p. 148).  
[https://www8.cao.go.jp/cstp/society5\\_0/smartcity/architecture.html](https://www8.cao.go.jp/cstp/society5_0/smartcity/architecture.html)  
 “自治体標準オープンデータセット(正式版)”  
[https://www.digital.go.jp/resources/open\\_data/municipal-standard-data-set-test](https://www.digital.go.jp/resources/open_data/municipal-standard-data-set-test)

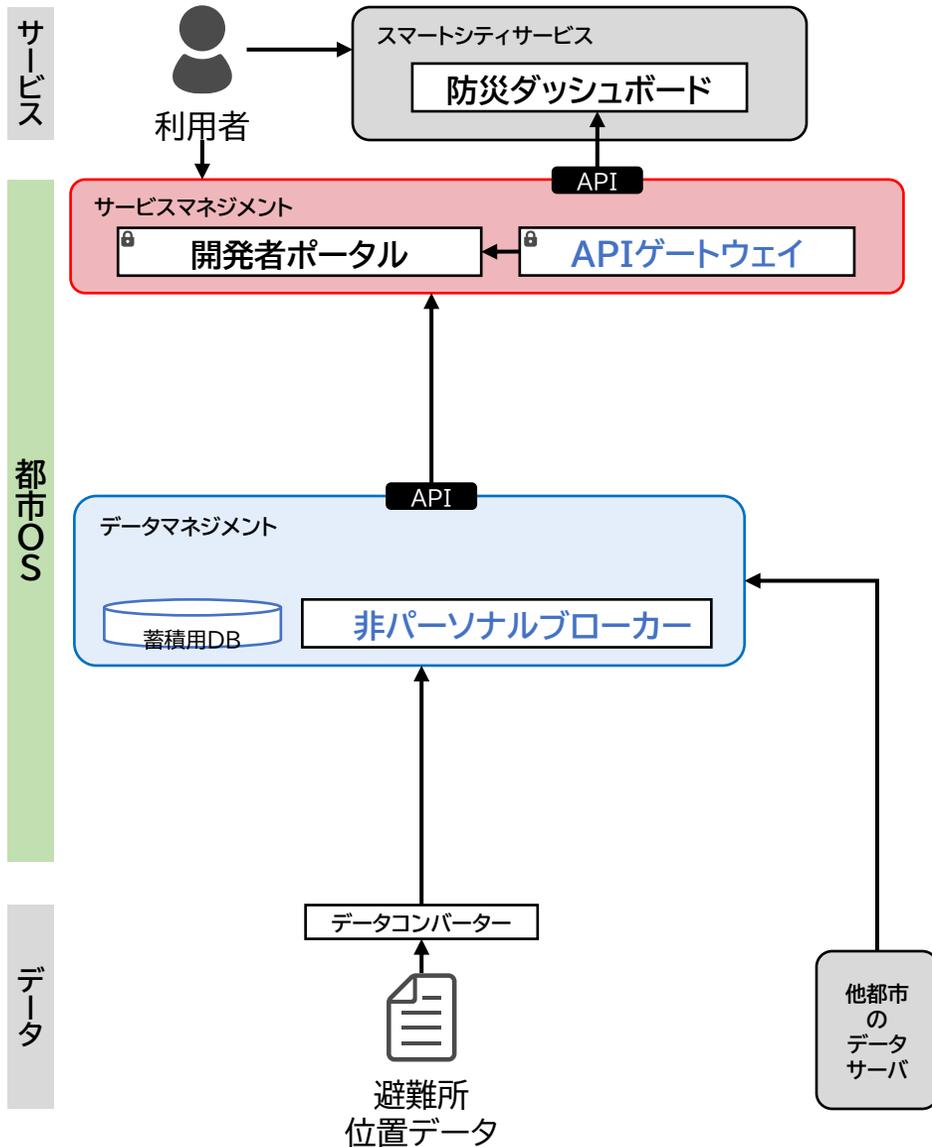
# ステップ0 オープンデータの公開（都市OS未導入）



オープンデータを専用のデータベースに登録し、ダッシュボードアプリケーションを公開している。

都市OSは利用していない状態

# ステップ1 地域間データ連携（都市OSの導入）

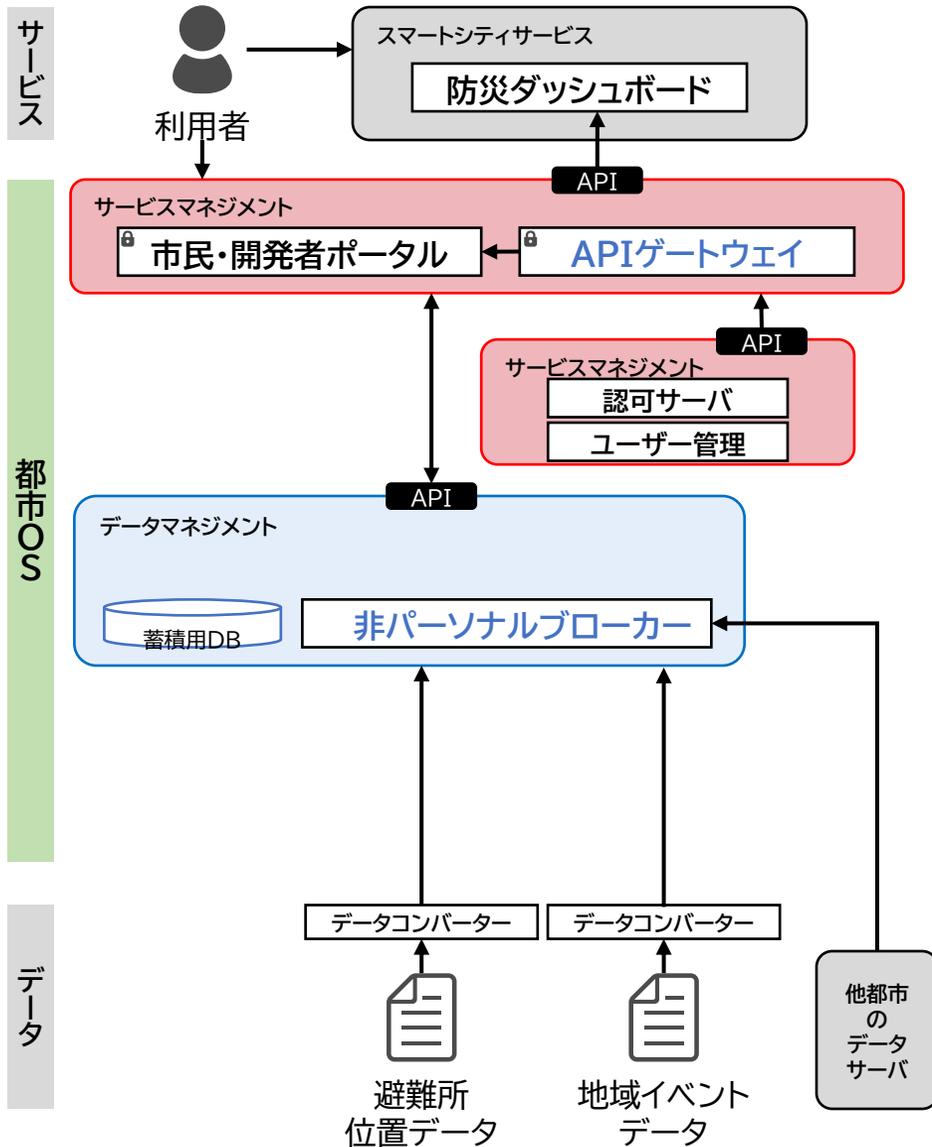


APIゲートウェイを導入し、データブローカーのAPIを含む複数のサービスが管理できる。

開発者向けのポータルサイトを導入し、新しいスマートシティサービスの内部検討が推進できる。

データブローカーを導入し、複数都市のデータが共同管理できる。

# システム拡充A 市民ポータルと認証連携の導入

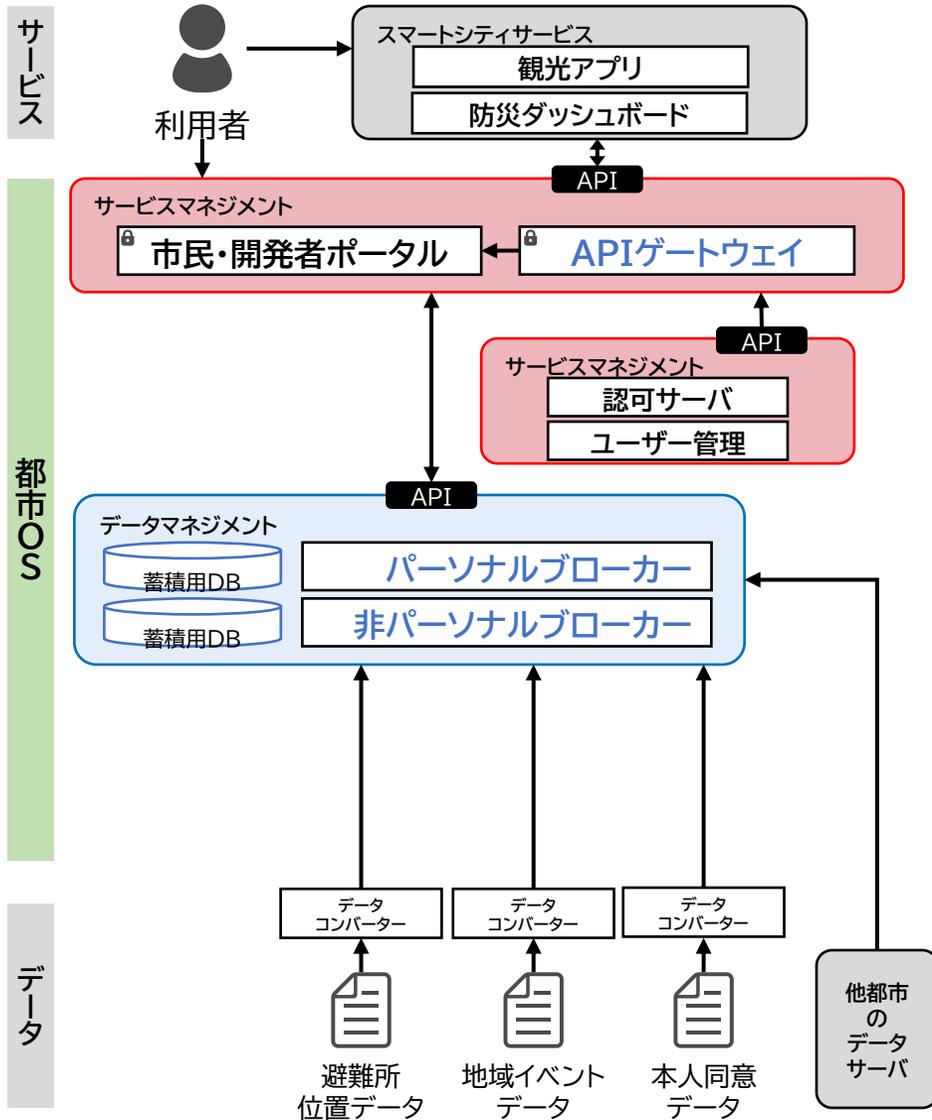


市民ポータルを導入し、ユーザへの地域イベントなどのレコメンドサービスが提供できる。

認可機能・ユーザー管理機能を導入し、スマートシティサービスに認証機能が利用できる。

データの形式や状態に応じたデータコンバーターを導入し、ブローカーが求める形式に変換できる。

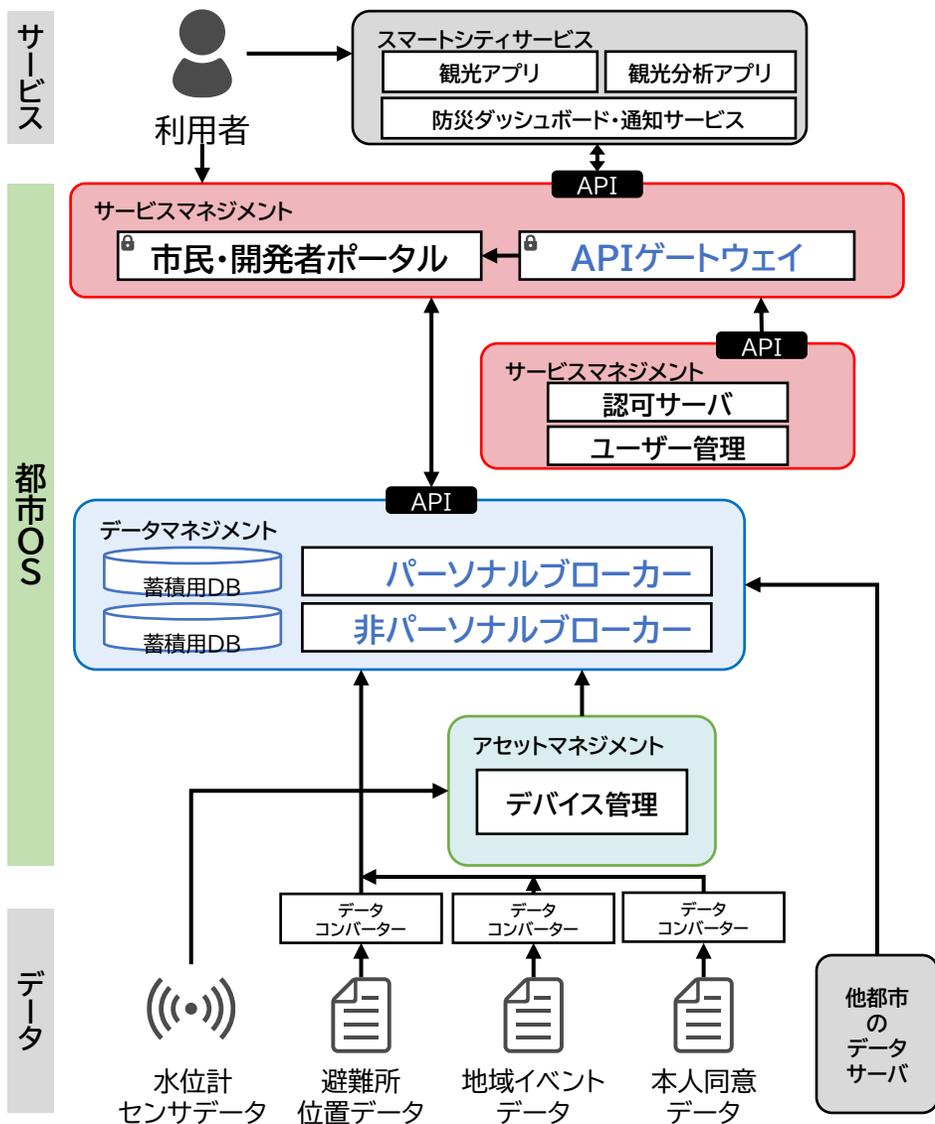
# システム拡充B パーソナルデータ連携機能の導入



本人同意や個人情報を活用した観光アプリサービス等の対話型サービスが提供できる。

パーソナルブローカーを導入し、本人同意データが共同管理できる。

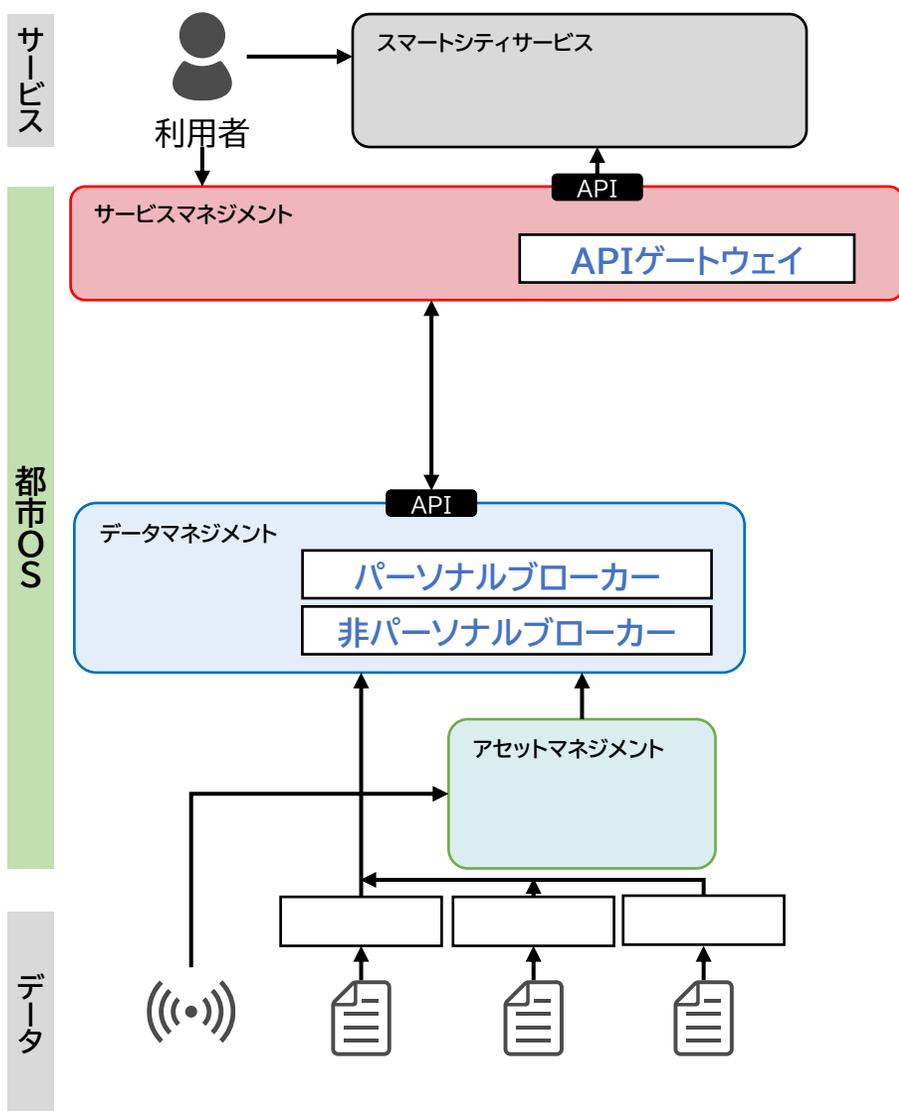
# ステップ2 広域データ連携（センサーアセットの導入など）



地域で提供されている複数のデータやAPIを活用し、複雑度が高い多くのサービスが提供できる。

デバイス管理機能を導入し、都市に設置されたIoTセンサデバイスのデータが活用できる。

# 都市OSのコア要素（推奨モジュール）

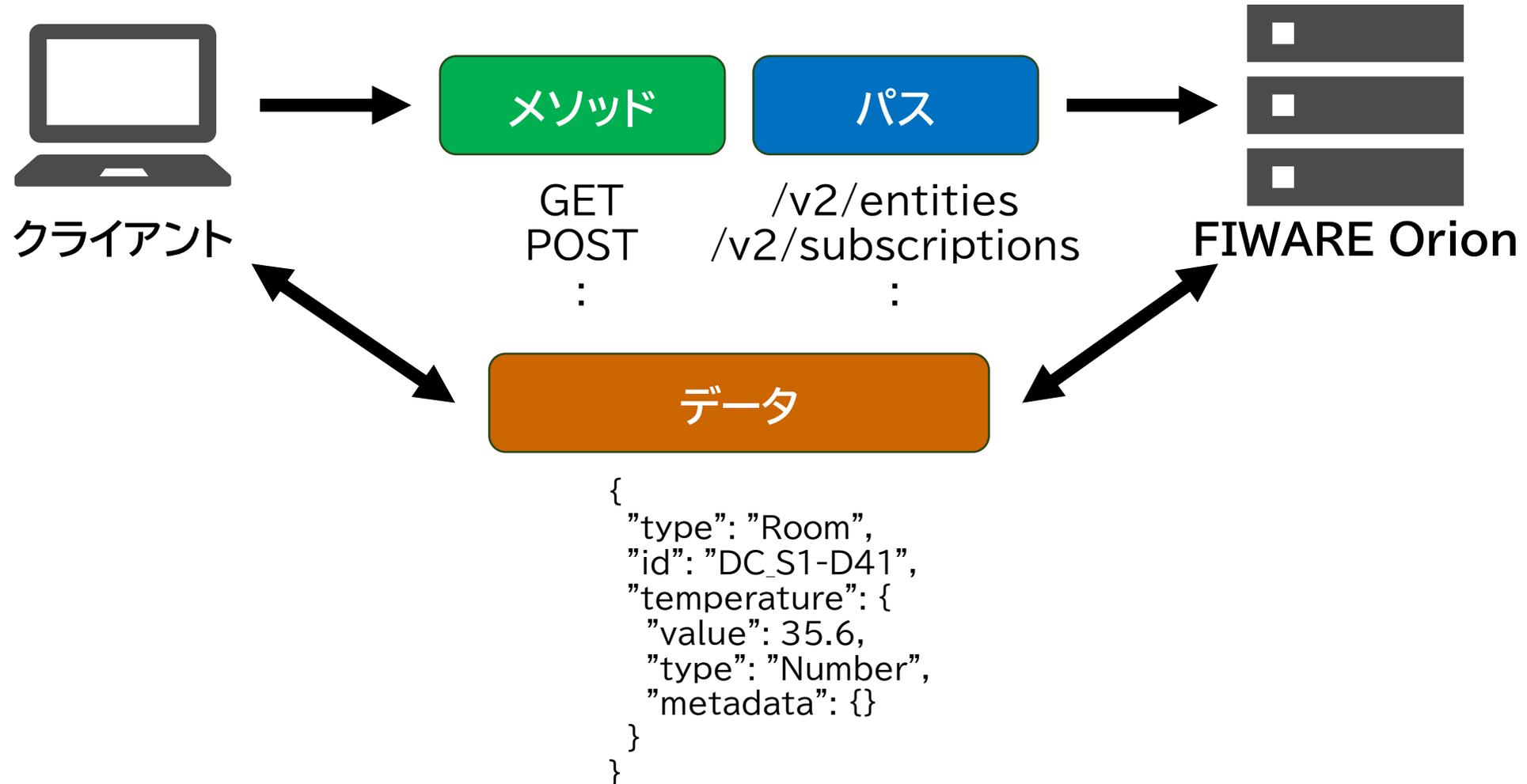


- 都市OSでは、**APIゲートウェイ**を中心とした **サービスマネジメント機能**を軸に、**API**を活用した **マイクロサービスアーキテクチャ**を構築することが重要である。
- データ連携基盤は、**データマネジメント機能**の **API**を提供する**データブローカー**を併用し、地域のデータ利活用促進を促す仕組みである。
- IoTデバイスなどの個別管理が必要な**アセット**がある場合は、**アセットマネジメント層**を設け、**データブローカー**と連携される必要がある。

# 非パーソナル向けデータ連携基盤で 利用されるAPIとデータモデル

# FIWARE OrionのAPIによるデータ授受

- FIWARE OrionのAPIでは、主にメソッド・パス・データを指定することで、クライアントとのデータ授受を実施している。このデータ授受のルールはNGSIv2と呼ばれている。



- API (Application Programming Interface) とは、プログラムから機能を利用するためのインタフェースの総称である。Web API、REST API、NGSIv2は下図のような包含関係にある。
- NGSIv2とは、FIWARE Orionが採用/提供するAPI仕様のことで、Orionとのデータ授受に関する各種ルールがREST APIの思想に準じて定義されている。

API: プログラムで機能を利用するためのインタフェースの総称。

Web API: APIのうち、Webを介して機能を利用するもの。

REST API: Web APIのうち、HTTPメソッドでリソースの登録/参照/更新/削除操作を表現するAPI設計思想。

NGSIv2: REST APIのうち、FIWARE Orionで採用されているコンテキスト情報を取り扱うためのAPI仕様。

▲ API / Web API / REST API / NGSIv2 の関係

# REST APIとは ~データ連携で用いられるAPI

- REST APIとは、リソースをパスで、アクションをHTTPメソッドで表現するAPI設計思想。近年のWebサービスのAPIは、多くがこれに準じた設計になっている。FIWARE Orionが採用するNGSIV2も同様である。
- FIWARE Orionでは連携するデータを**Entity**という単位で管理している。

▼ HTTPメソッドとその役割

	Method	用途
Read	GET	リソースの参照
	POST	リソースの登録
Write	PUT	リソースの更新
	PATCH	リソースの部分更新
	DELETE	リソースの削除

▼ FIWARE OrionのAPI仕様例 (抜粋)

用途	Method	Path
エンティティの一覧取得	GET	/v2/entities
エンティティの登録	POST	
エンティティの参照	GET	/v2/entities/{entityId}
エンティティの更新	PUT	
エンティティの部分更新	PATCH	
エンティティの削除	DELETE	
アトリビュートの取得	GET	/v2/entities/{entityId}/attrs/{attrName}
アトリビュートの更新	PUT	
サブスクリプションの一覧取得	GET	/v2/subscriptions
サブスクリプションの登録	POST	

# FIWARE Orionで用いられるデータモデル

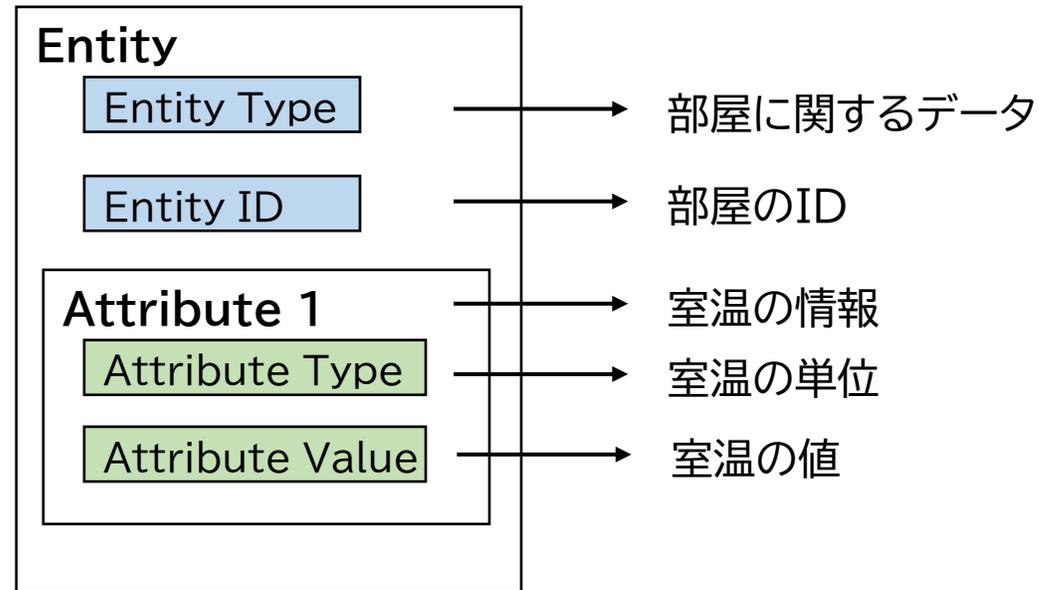
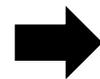
- FIWARE Orionによるデータ授受は、**JSON**形式で表記されたテキストデータのやりとりによって実現している。**JSON**形式では、データ全体を中括弧で囲み、キーと値のペアを使って情報を整理している。
- FIWARE Orionの**Entity**では、**Type**と**ID**というEntityの識別キーと、そのEntityでやり取りしたい詳細情報を**Attribute**をひとまとめにしたデータを取り扱っている。

```

{
  "type": "Room",
  "id": "DC_S1-D41",
  "temperature": {
    "value": 35.6,
    "type": "Number",
    "metadata": {}
  }
}

```

▲ JSON表記のEntityデータ例



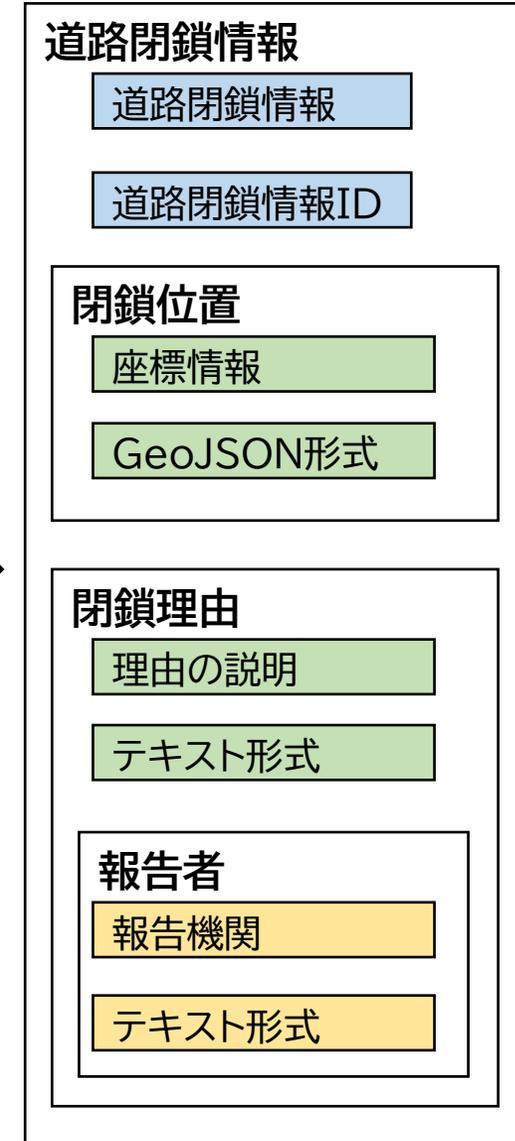
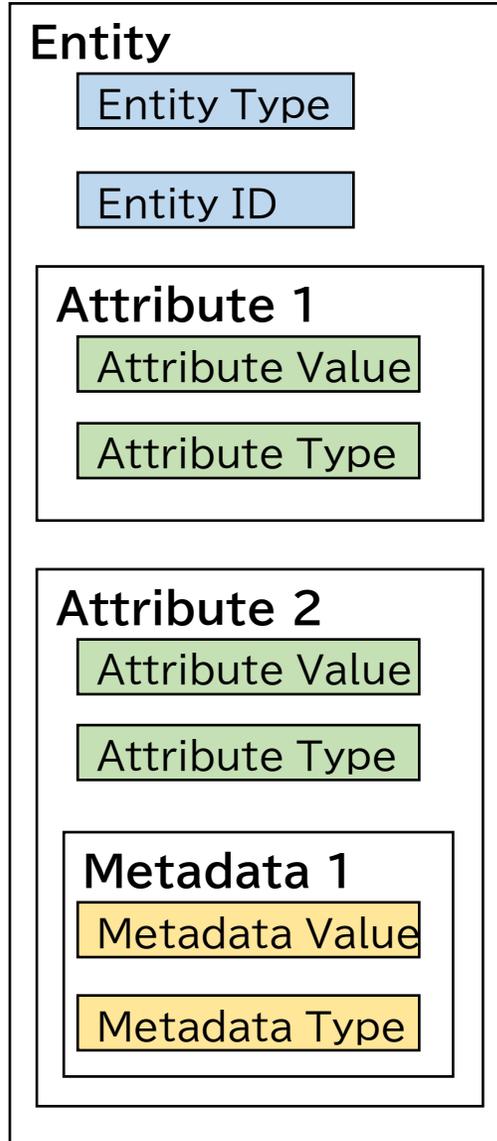
▲ Entityデータの構造

# FIWARE Orionで用いられるEntityデータの具体例

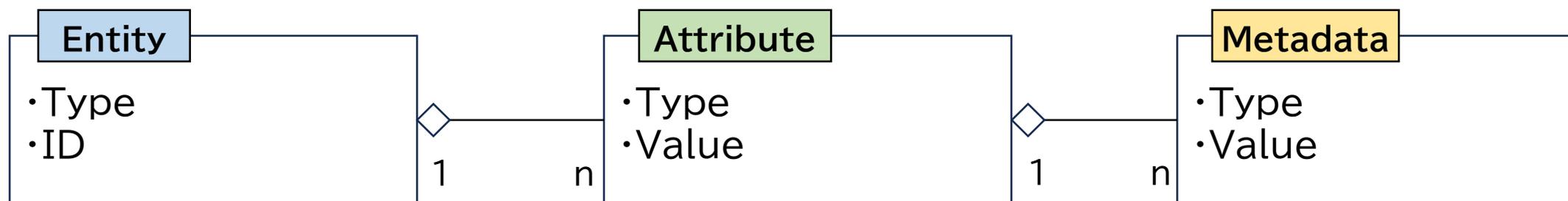
```

{
  "type": "RoadClosure",
  "id": "RoadClosure:12345",
  "location": {
    "value": {
      "type": "Point",
      "coordinates": [
        -3.68666,
        40.48108
      ]
    }
  },
  "type": "geo:json",
},
"closureReason": {
  "value": "Construction",
  "type": "Text",
  "metadata": {
    "reportedBy": {
      "value": "City Maintenance Department",
      "type": "Text"
    }
  }
}
}

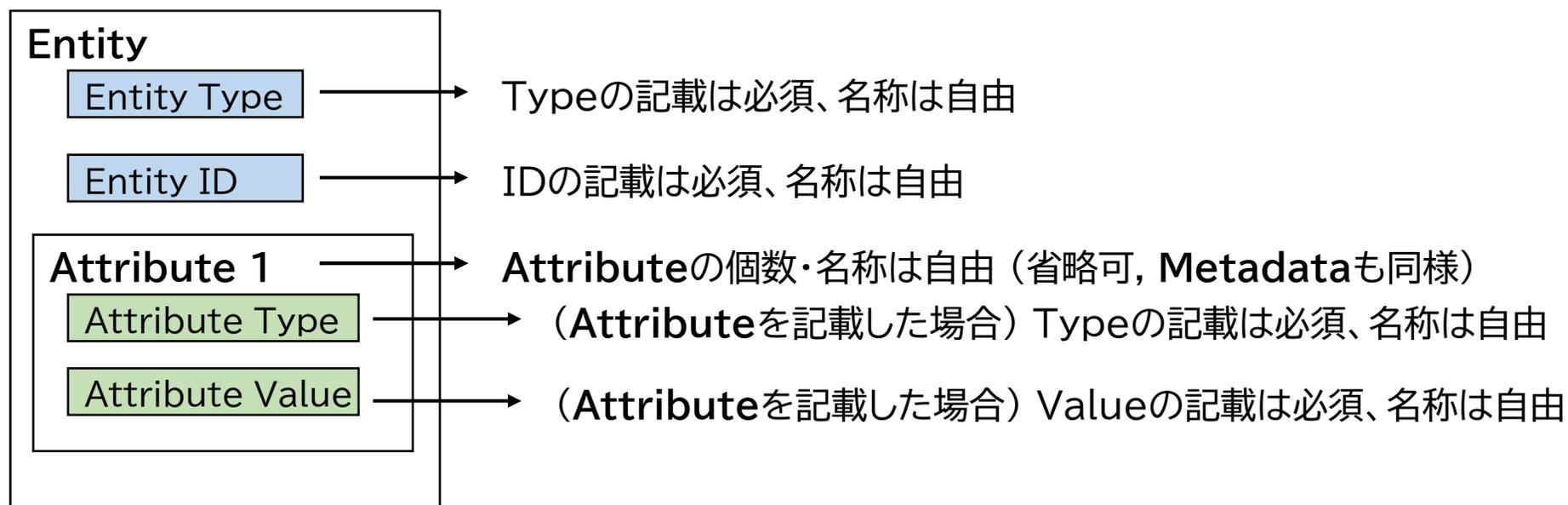
```



- FIWARE Orionの**Entity**には、複数の**Attribute**を設定することができ、**Entity**と**Attribute**の関係は1対多の関係である。さらに**Attribute**には、複数の**Metadata**を設定することができ、**Attribute**と**Metadata**の関係も1対多の関係である。
- 前ページの例のように「室温情報を有した部屋データ」を管理したり、「道路閉鎖の座標、閉鎖理由とその報告者を有した道路閉鎖データ」を管理したりすることが基本である。
- Entityデータの管理単位や命名規則に関する統一ルールは現状策定されていないため、データ連携基盤の発展によって検討が加速するものと考えられる。



- FIWARE Orionを中心としたシステムでは、**Entity-Attribute-Metadata**の構造に準拠したJSON形式のテキストデータを管理することができる。
- **Attribute**項目名は、一部のデータベース予約語(dataCreatedなど)を除き自由に指定することができる。
- **Entity**データの管理単位や命名規則に関するルールは現状策定されていないため、データ連携基盤の発展によって検討が加速するものと考えられる。

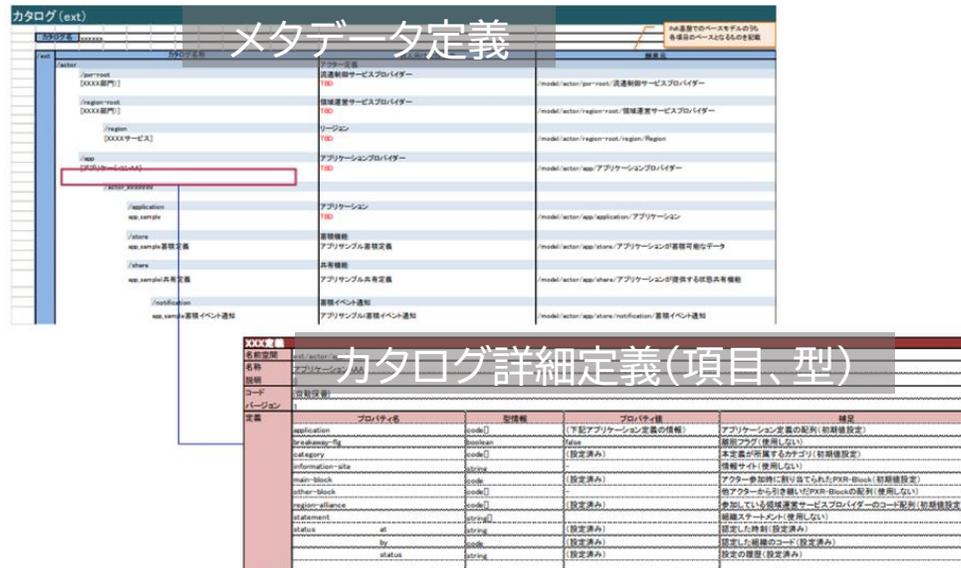


# パーソナル向けデータ連携基盤で 利用されるAPIとデータモデル

# パーソナルデータ連携基盤で用いられるデータモデル

- パーソナルデータ連携基盤では流通させたいデータモデルをデータカタログ (My-Condition-Book) として定義※。  
データカタログはJSON形式となっており、公開APIを用いてシステムにデータを投入する。

## カタログ定義シート



項目名	型	型情報	プロパティ値	補足
application	code[]	(下記アプリケーション定義の情報)	アプリケーション定義の配列(初期値設定)	
displayname_jg	boolean		画面フラグ(使用しない)	
category	code[]	(設定済み)	本定義が所属するカテゴリ(初期値設定)	
information_site	code[]	(設定済み)	情報サイト(使用しない)	
main_block	code[]	(設定済み)	アクター参加時に割り当てられたMyC-Book(初期値設定)	
author_block	code[]	(設定済み)	他アクターから引き継いだMyC-Bookの配列(使用しない)	
region_alliance	code[]	(設定済み)	参加している医療連携サービスプロバイダーのコード配列(初期値設定)	
statement	string	(設定済み)	個人ステートメント(使用しない)	
status	code[]	(設定済み)	設定したステータス(初期値設定)	
by	code[]	(設定済み)	設定した組織のコード(設定済み)	
status	string	(設定済み)	設定の理由(設定済み)	

## カタログ定義ファイル

JSON形式ファイル

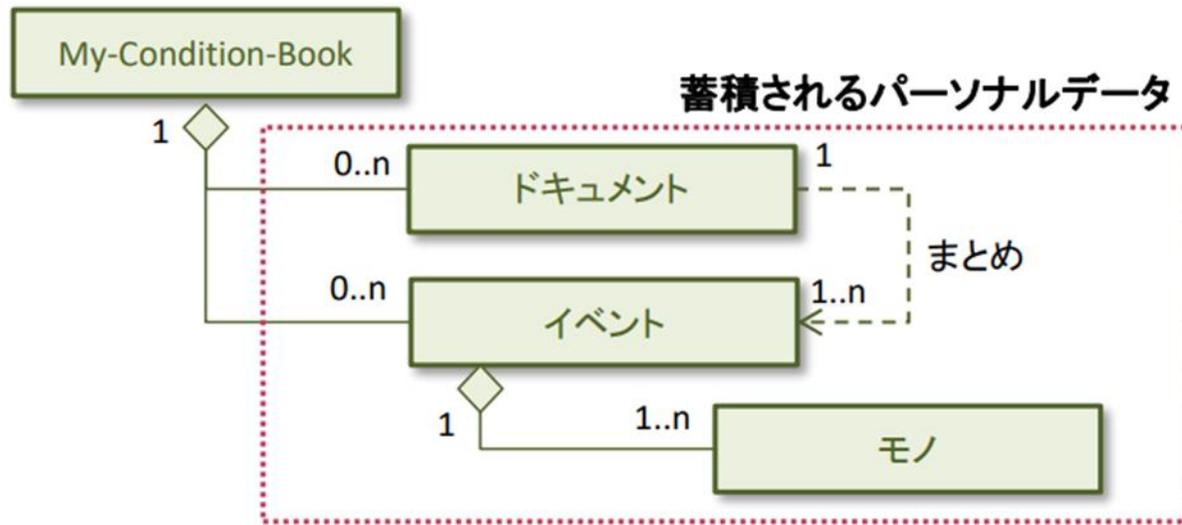
```
{
  "catalogItem": {
    "ns": "catalog/ext/nec-healthcare-consortium/document/actor_1001548",
    "name": "診察レポート",
    "_code": {
      "_value": 1001613,
      "_ver": 2
    },
    "inherit": {
      "_value": 52,
      "_ver": 1
    },
    "description": {
      "title": null,
      "section": [
        {
          "title": "診察レポート",
          "content": [
            {
              "sentence": "診察レポート"
            }
          ]
        }
      ]
    }
  }
}
```

公開API  
(カタログ管理  
サービス)

※カタログ管理については、セミナー資料を参照のこと  
[pxr-linkage/doc/セミナー/20221227 パーソナルデータ連携モジュール 説明会資料/パーソナルデータ連携モジュール 説明資料 v3.pdf](https://pxr-linkage/doc/セミナー/20221227%20パーソナルデータ連携モジュール%20説明会資料/パーソナルデータ連携モジュール%20説明資料%20v3.pdf) at main · Personal-Data-Linkage-Module/pxr-linkage · GitHub

# データカタログ(My-Condition-Book)の構造

- パーソナルデータ連携基盤では、カタログ管理にて、データ定義を管理。  
My-Condition-Bookの構造に従ってデータを定義する。



## My-Condition-Book

個人の基本属性とドキュメントおよびイベントを格納  
例)個人ID,氏名,性別,生年月日,住所,etc

## ドキュメント

個人が体験するシーンに紐づくイベントのまとめり+  
アドバイスなどのテキスト  
例)本日の診察レポート

## イベント

個人が体験するシーンを個人の基本動作に分解したもの  
例)診断、検体検査、負荷検査

## モノ

イベントの中で記録、測定、評価される項目  
例) 診断病名、検査項目-値

### 3. データ連携基盤の構成例と機能要件例

- 非パーソナル                   :交通・モビリティ  
                                      :防災
- パーソナル                     :ヘルスケア
  
- 非パーソナルとパーソナル両方適用の例



## 【交通】新規サービスのコンセプト(例)

### ● 解決したい地域課題

- 交通渋滞が深刻化し、地域経済に影響を与えている

### ● 実現したいサービス

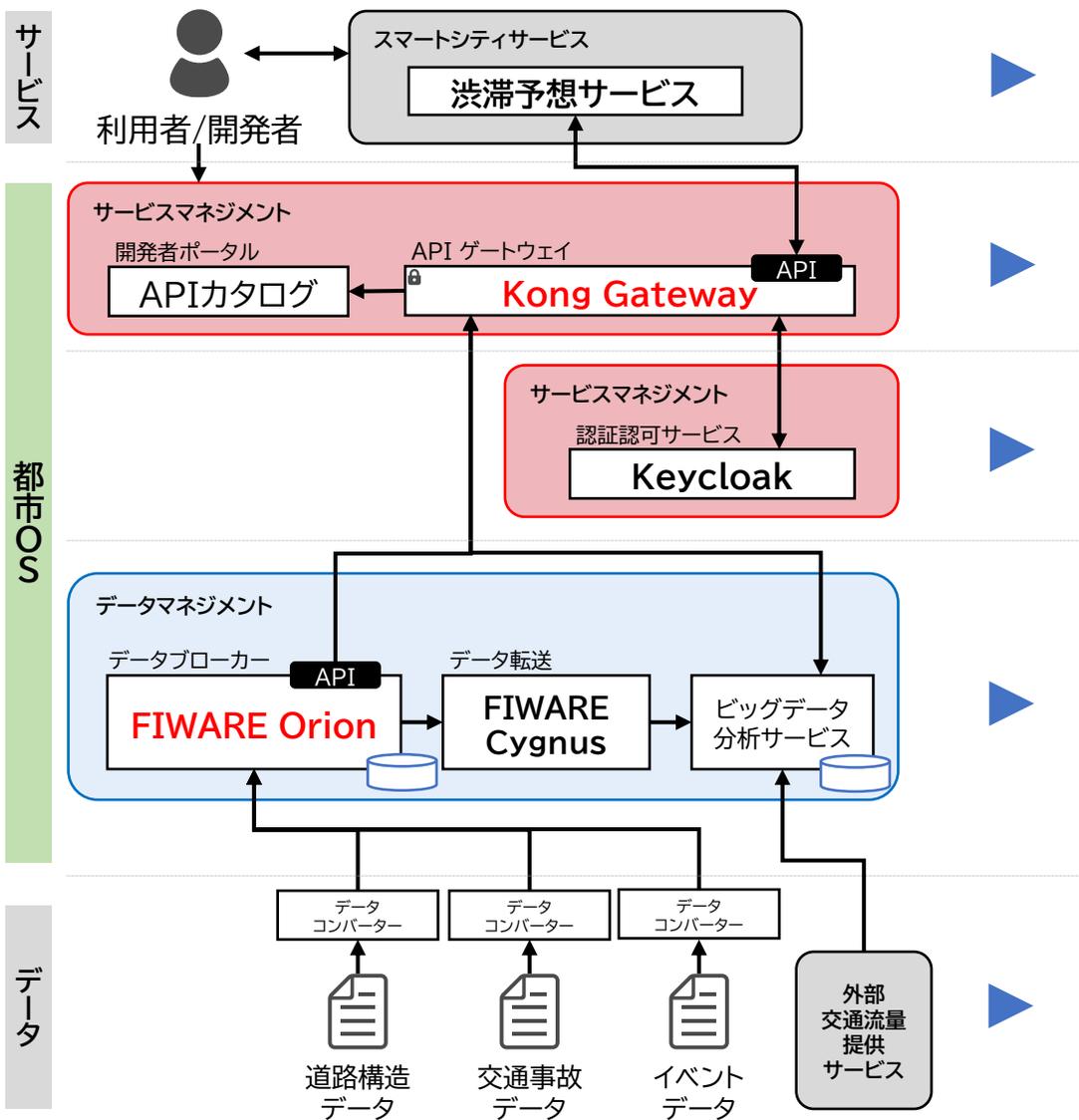
- 地域の公的機関が有するリアルタイムな情報を加味した渋滞予測情報を提供したい
- 地域の企業と連携して渋滞回避に向けた地域計画を検討したい

### ● 活用したいアセット・外部データ

- 道路データ
- 交通事故データ
- 地域イベント予定データ
- (外部提供)交通量データ

⇒ **非パーソナル型のデータ連携基盤**の活用を検討

# 【交通】システム構成例と要件例



赤字:推奨モジュール、黒字:OSSモジュール

## 道路渋滞予想サービスの提供

- ▶ 現在の渋滞位置と予想結果がユーザーに提供できること
- ▶ 利用者からのフィードバック情報が収集できること

## サービス管理機能/開発者向けポータル機能の提供

- ▶ サービス間/地域間でのデータ連携/機能共有が可能であること
- ▶ 開発者向けに本システムのAPI仕様書閲覧機能が提供できること

## 認証連携/課金請求機能の提供

- ▶ 認証認可サービスと連携し、API利用者の制限・管理が可能であること
- ▶ API利用者を特定し、利用量に応じた課金請求が可能であること

## データ管理機能の提供

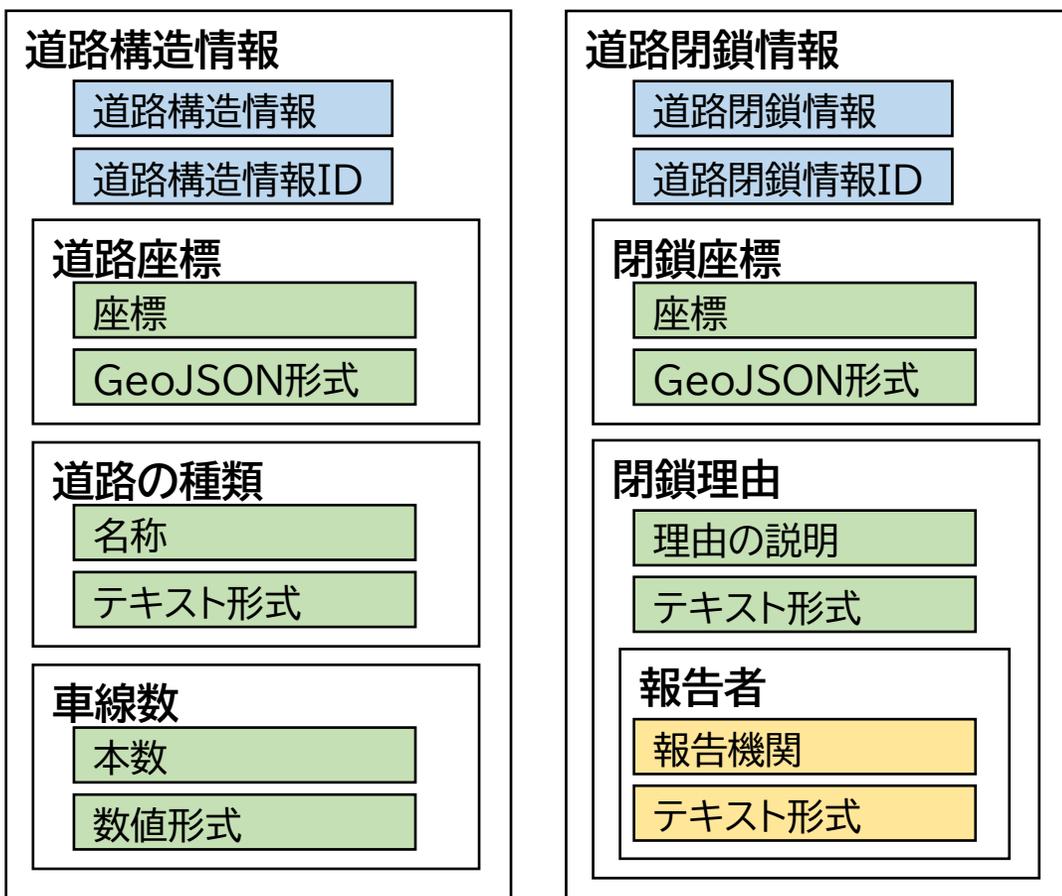
- ▶ APIによるサービス間/地域間でのデータ連携が可能であること
- ▶ リアルタイムデータ収集によるビッグデータ管理/予測が可能であること
- ▶ ビッグデータ分析サービスへのリアルタイムなデータ転送が可能であること

## データの収集・連携

- ▶ 道路情報、交通事故情報等の公的データが追加可能であること
- ▶ 外部の民間企業が提供する交通情報等の取得が可能であること

## データモデル

- JSON形式かつEntity - Attribute - Metadataの構造  
各種データを表現する。



## API

- FIWARE OrionのAPIをデータ授受機能として提供する。
- 一部のAPIにはKong Gatewayで認証認可を設け、管理者等の関係者のみが操作できるように構成する。
- サブスクリプション用のAPIを活用して、更新通知によるサービス間のデータ連携機能を提供する。

用途	Method	Path
エンティティの一覧取得	GET	/v2/entities
エンティティの登録 	POST	
エンティティの参照	GET	/v2/entities/{entityId}
サブスクリプションの一覧取得 	GET	/v2/subscriptions
サブスクリプションの登録 	POST	

## 【防災】新規サービスのコンセプト(例)

### ● 解決したい地域課題

- 自然災害による危機発生に備えた防災計画を支援したい

### ● 実現したいサービス

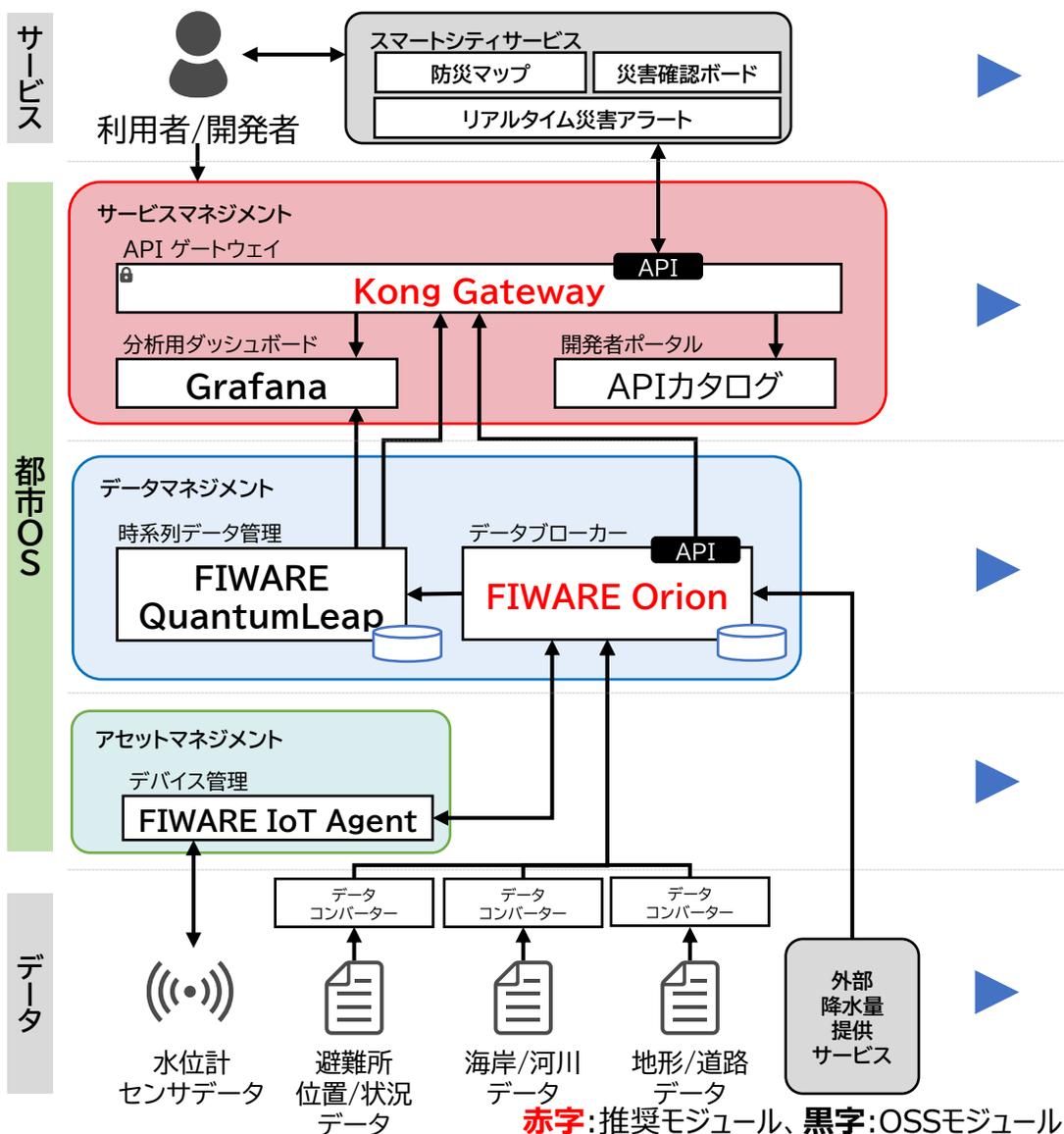
- 地域の公的機関が有する施設/避難所情報の情報提供や可視化が実現したい。
- 河川水位をリアルタイムに監視し、大雨等のリスク発生時にはアラームを発報したい。

### ● 活用したいアセット・外部データ

- 河川水位計データ
- 避難所/施設データ
- 地形/道路データ
- (外部提供)降水量データ

⇒ 非パーソナル型のデータ連携基盤の活用を検討

# 【防災】システム構成例と要件例



## 地域防災情報プラットフォームサービスの提供

- ▶ 防災関連機能のAPIを活用した各種サービスが提供できること
- ▶ 防災関連データに基づく災害通アラート機能が提供できること

## サービス管理機能/開発者向けポータル機能の提供

- ▶ サービス間/地域間でのデータ連携が可能であること
- ▶ キー認証によってAPIの利用者が制限できること
- ▶ 開発者向けに本システムのAPI仕様書閲覧機能が提供できること
- ▶ リアルタイムデータの可視化/分析機能が提供できること

## データ管理機能の提供

- ▶ サービス間/地域間でのデータ連携が可能であること
- ▶ センサデータ等の時系列データを蓄積し、API提供が可能であること

## アセット管理機能の提供

- ▶ 公的機関が所有する各種デバイスを管理・疎通し、データ収集が可能であること

## データの収集・連携

- ▶ 公的機関が所有する各種センサーのデータが収集可能であること
- ▶ 避難所などの施設情報、道路情報等の公的データが追加可能であること
- ▶ 外部の民間企業が提供する雨量等の気象情報等の取得が可能であること

## データモデル

- 水位計センサーが収集した水位情報と、その拠点の基準水深情報をそれぞれNGSIv2形式で記録する。



## API

- FIWARE OrionとFIWARE QuantumLeapのAPIをデータ授受機能として提供する。同一Path名のAPIがあるため、Kong Gatewayのルーティング設定で異なる名称を指定する。
- 一部のAPIにはKong Gatewayで認可を設け、管理者等の関係者のみが操作できるように構成する。
- サブスクリプション用のAPIを活用して、更新通知によるサービス間のデータ連携機能を提供する。

用途	Method	Path
エンティティの一覧取得	GET	/orion/v2/entities
エンティティの登録 	POST	
エンティティの参照	GET	/v2/entities/{entityId}
サブスクリプションの一覧取得 	GET	/v2/subscriptions
サブスクリプションの登録 	POST	
エンティティの時系列取得	GET	/quantumleap/v2/entities

## 【ヘルスケア】新規サービスのコンセプト(例)

### ● 解決したい地域課題

- 地理的要因で、医療施設への通院や医療従事者の訪問が困難な状況
- 住民の健康情報を基に運動習慣を促進して健康増進を図りたい

### ● 実現したいサービス

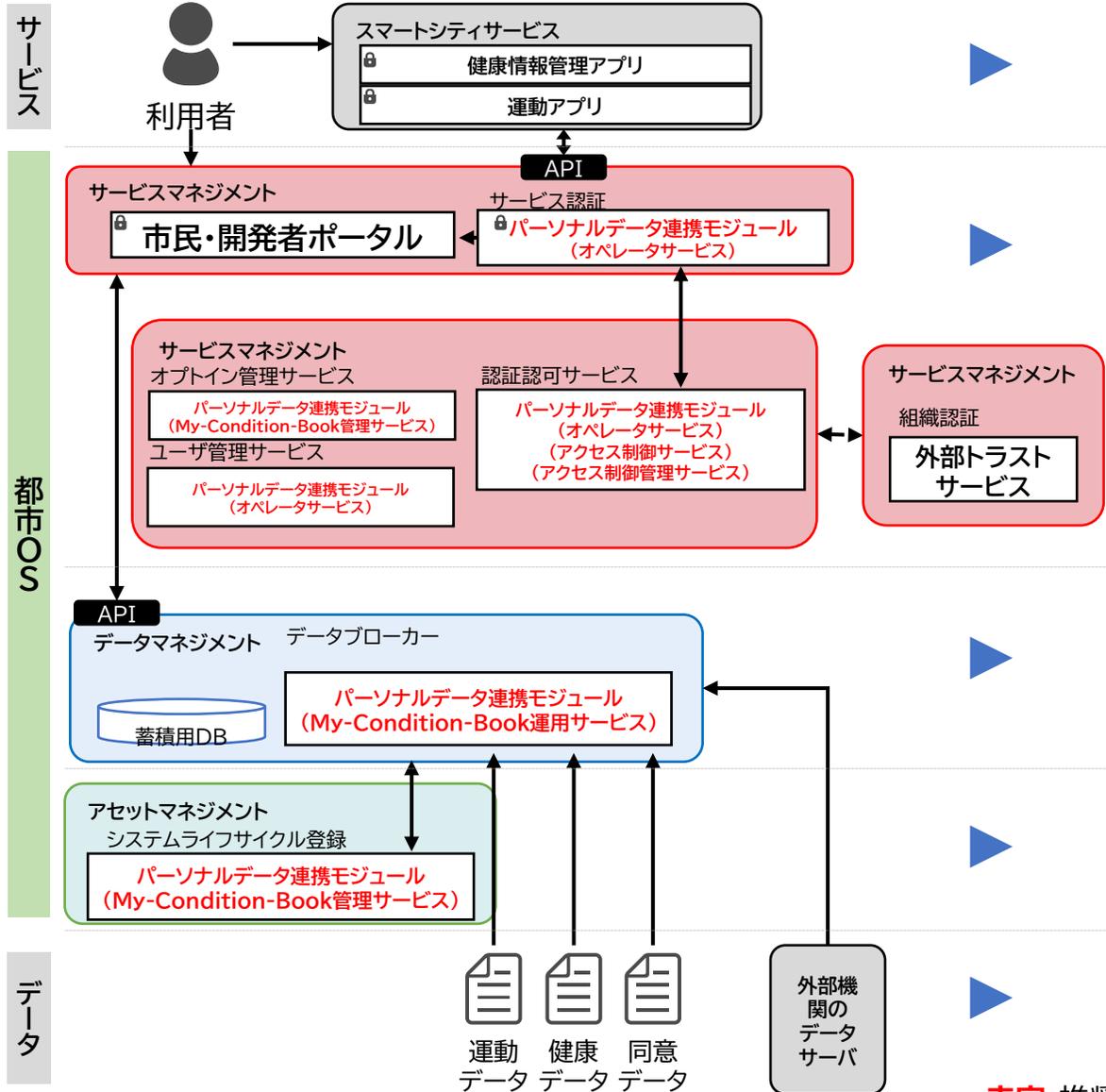
- 健康データと運動記録を踏まえた健康改善相談サービス

### ● 活用したいアセット・外部データ

- 運動レポートデータ
- (外部提供)外部機関が管理しているヘルスケアデータ (例:健康診断データ)

⇒データの扱いに本人同意が必要とするため、**パーソナル型のデータ連携基盤**の活用を検討

# 【ヘルスケア】システム構成例と要件例



## 地域のポータルサービスの提供

- 健康データと運動記録を踏まえた健康改善相談ができること

## サービス管理機能/開発者向けポータル機能の提供

- 地域で提供されている複数のデータやAPIを活用し、複数のサービスが提供できること。
- 認可サーバおよびIDプロバイダと連携し、ポータルやアプリにユーザログイン機能が提供できること。

## オプトイン管理機能の提供

- 住民が個人の判断で、個人のパーソナルデータの公開範囲を指定するための機能を提供できること。

## データ管理機能の提供

- 利用者の同意データに基づいてデータの蓄積、共有ができること。

## システムライフサイクル登録

- 連携する外部システムのの連携情報を管理できること。

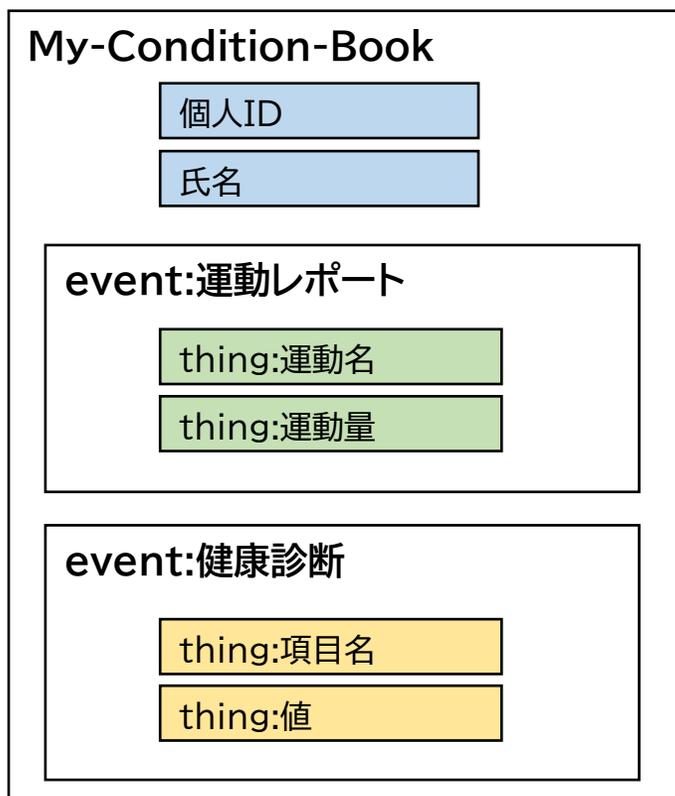
## データの収集・連携

- 外部機関が管理しているヘルスケアデータが連携できること。

赤字:推奨モジュール、黒字:OSSモジュール

## データモデル

流通させたいデータモデルをデータカタログ(My-Condition-Book)として定義。My-Condition-Bookに運動レポート及び、健康診断を保持。



## API

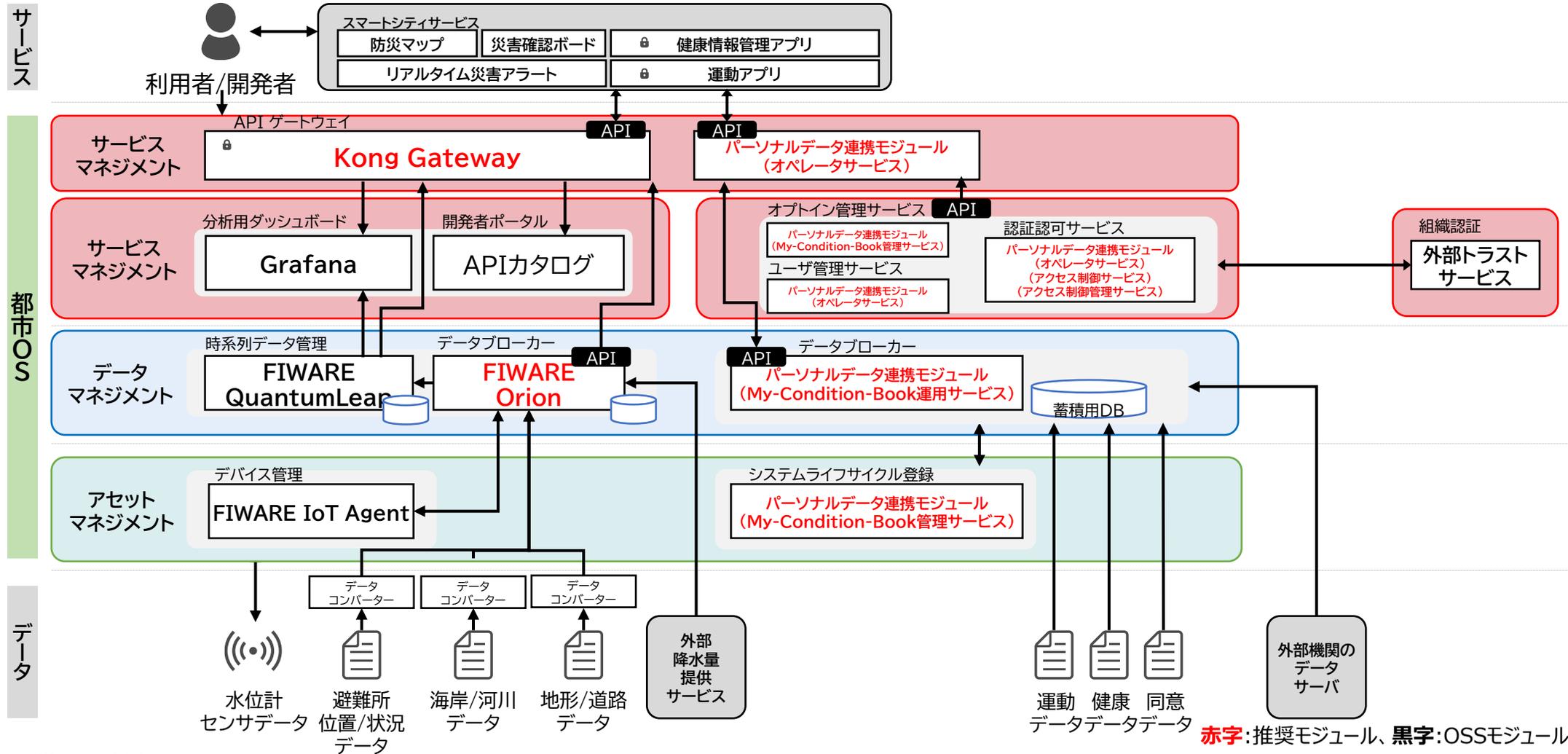
複数のアプリを連携させるサービスを定義して蓄積・共有するデータを登録したのち、定義したサービスに対する個人同意を取得する。

同意が取得されているデータのみ、蓄積・共有を可能とする。

用途	Method	Path
データの蓄積定義追加	POST	/settings/store
データの共有定義追加	POST	/settings/share
未同意の規約の取得	POST	/term_of_use/request/list
プラットフォーム利用規約同意	POST	/term_of_use/platform
データの蓄積(event)	POST	/event/{userId}
データの蓄積(thing)	POST	/thing/{userId}/{eventId}
データ共有のリクエスト	POST	/share

# 非パーソナルとパーソナル両方適用の例(防災×ヘルスケア)

- 一つのエリアにパーソナルと非パーソナルの基盤が存在する構成例を以下に示す。それぞれのデータを掛け合わせて横断型のサービスを提供するといった拡張性を備えることができる。



# Appendix

- FIWARE Orionのデータ管理について
- APIゲートウェイの役割・機能 / Kong Gatewayの特徴
- パーソナルデータ連携モジュールの概要と公開情報

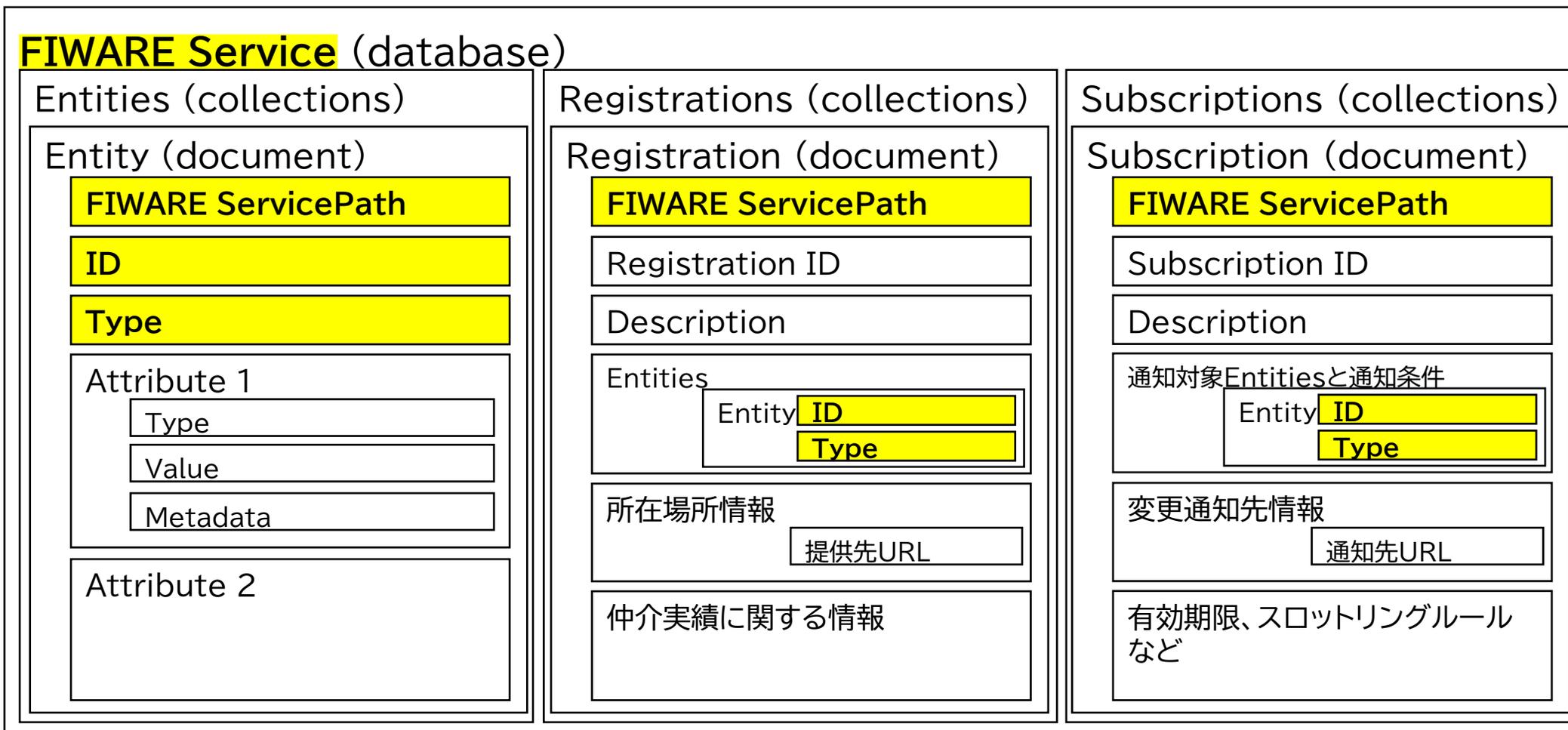
# FIWARE Orionのデータ管理について



- FIWARE Orionは蓄積用DBとして必ずMongoDBを利用する。
- MongoDBには大別して3種の情報記録し、このデータ群を元に機能提供している。
  - ①データ提供者が登録したEntityデータ (**Entities**コレクション)
  - ②データ提供者が登録したEntityデータ提供先情報 (**Registrations**コレクション)
  - ③Entityデータ更新通知の配信先情報 (**Subscriptions**コレクション)

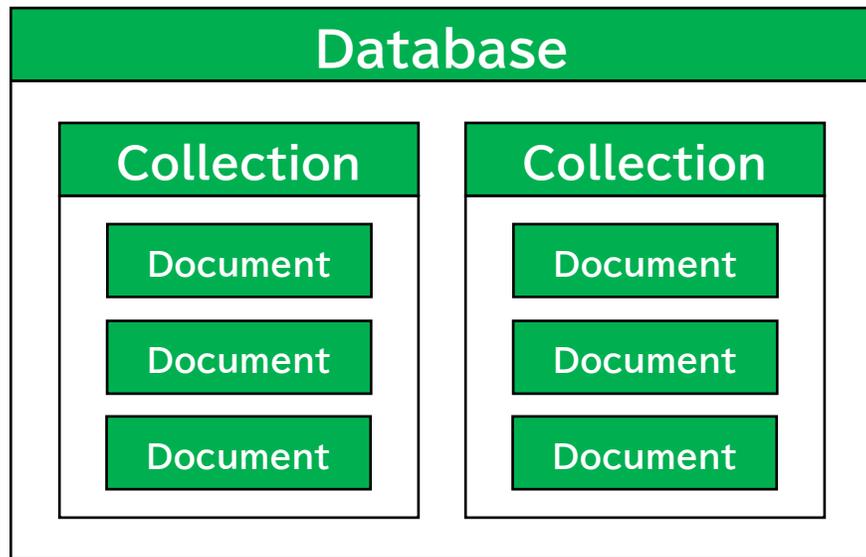


- データ利用時のクエリでは、**FIWARE Service**, **FIWARE ServicePath**, **Entity ID**, **Entity Type**の4項目の複合キーでレコードを検索し、ドキュメントを取得する。

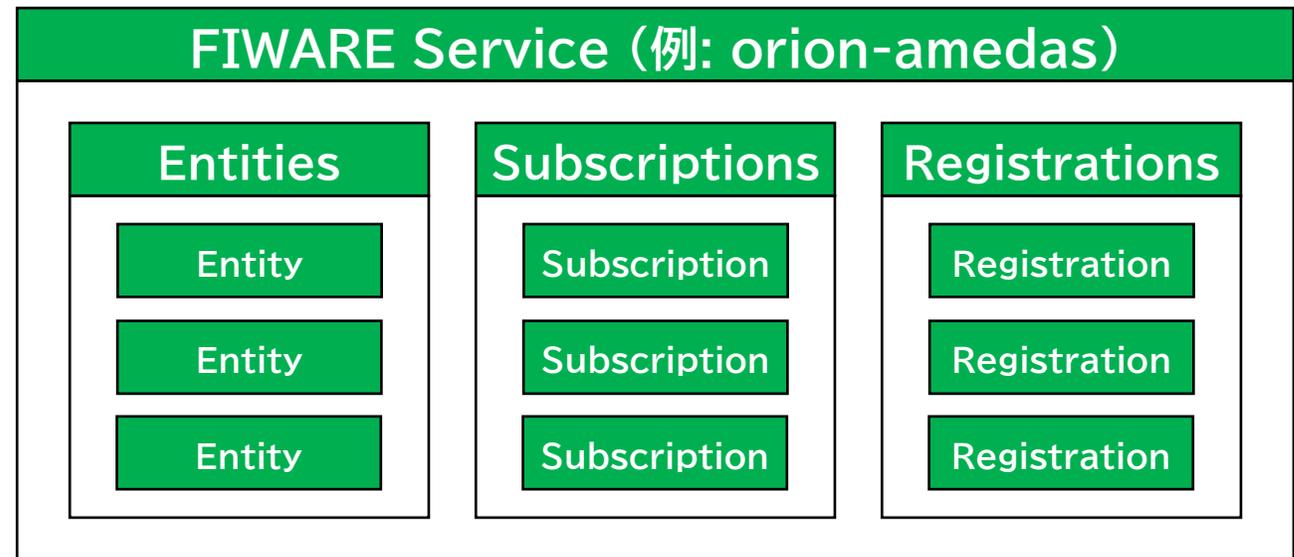


# MongoDBのデータベース構造とFIWARE Orionでの管理方法

- MongoDBではDocumentと呼ばれるJSON形式(BSON)のデータをCollection, Databaseという単位でまとめてデータを管理している。
- FIWARE Orionの3種の情報はCollection単位で管理され、1つのデータベースで情報が完結する。
- MongoDB内に複数のDatabaseを作成し、マルチテナント運用も可能である。



MongoDBのデータベース構造



FIWARE Orionのデータ管理構造と名称

# APIゲートウェイの役割・機能 / Kong Gatewayの特徴



# Kong Gateway: APIゲートウェイの推奨モジュール

- Kong Gatewayは、国内外で広く利用されている代表的なAPIゲートウェイである。コア機能はOSSで提供され、プラットフォームの制約なく利用することができる。プラグインによる機能拡張が可能な設計で、拡張性と柔軟性の高さにも優位性がある。

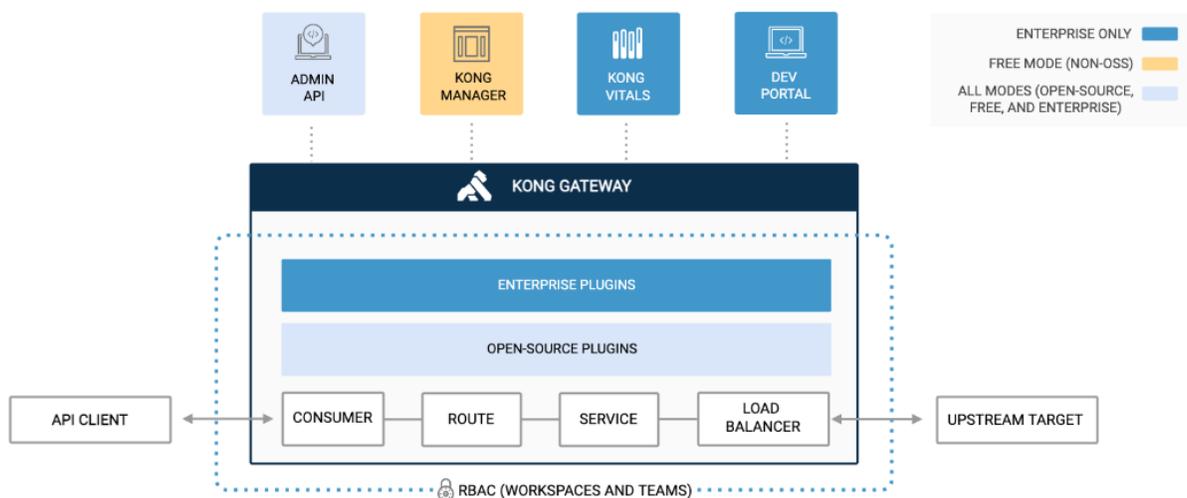
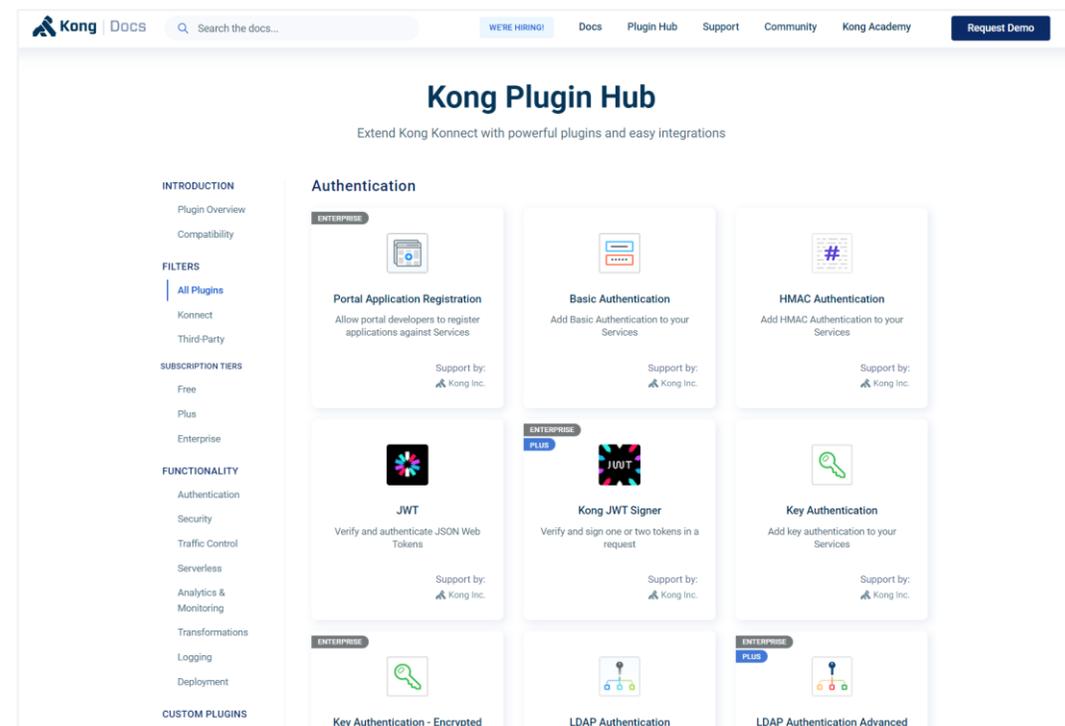


Figure 1: Diagram of Kong Gateway modules and how they relate to the foundational Gateway components.

Requests flow from an API client into the Gateway, are modified and managed by the proxy based on your Gateway configuration, and forwarded to upstream services.

Kongドキュメントより引用  
<https://docs.konghq.com/gateway/latest/>

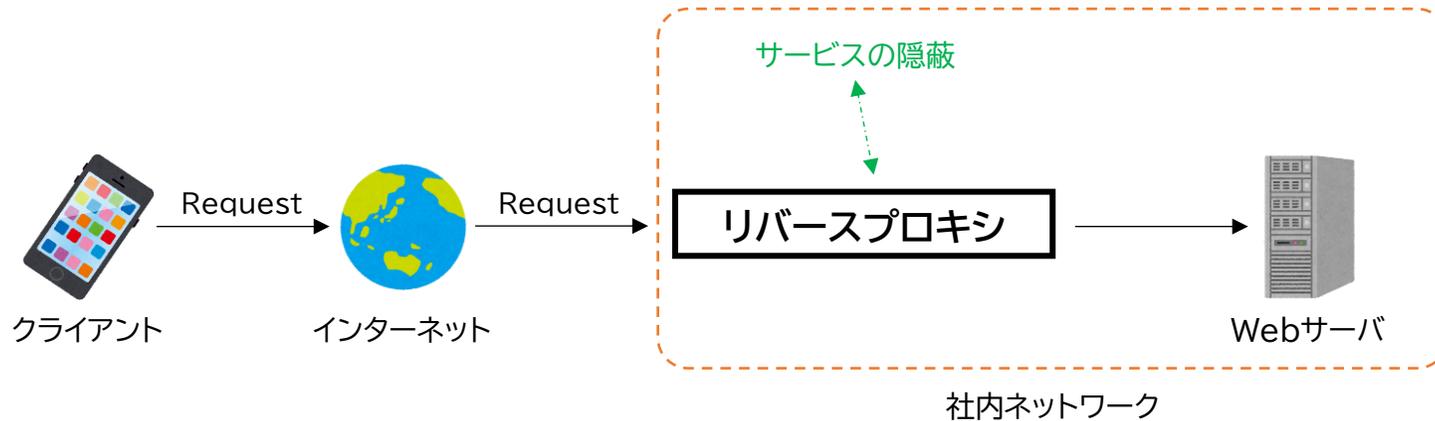


Kongプラグイン一覧ページ  
<https://docs.konghq.com/hub/>

- APIゲートウェイの役割を理解するには、類似する3つの製品を比較するとわかりやすい。
  1. リバースプロキシ
  2. ロードバランサー
  3. APIゲートウェイ

# リバースプロキシの役割

- リバースプロキシは、クライアントからのリクエストを受信し、内部サーバーに転送する。

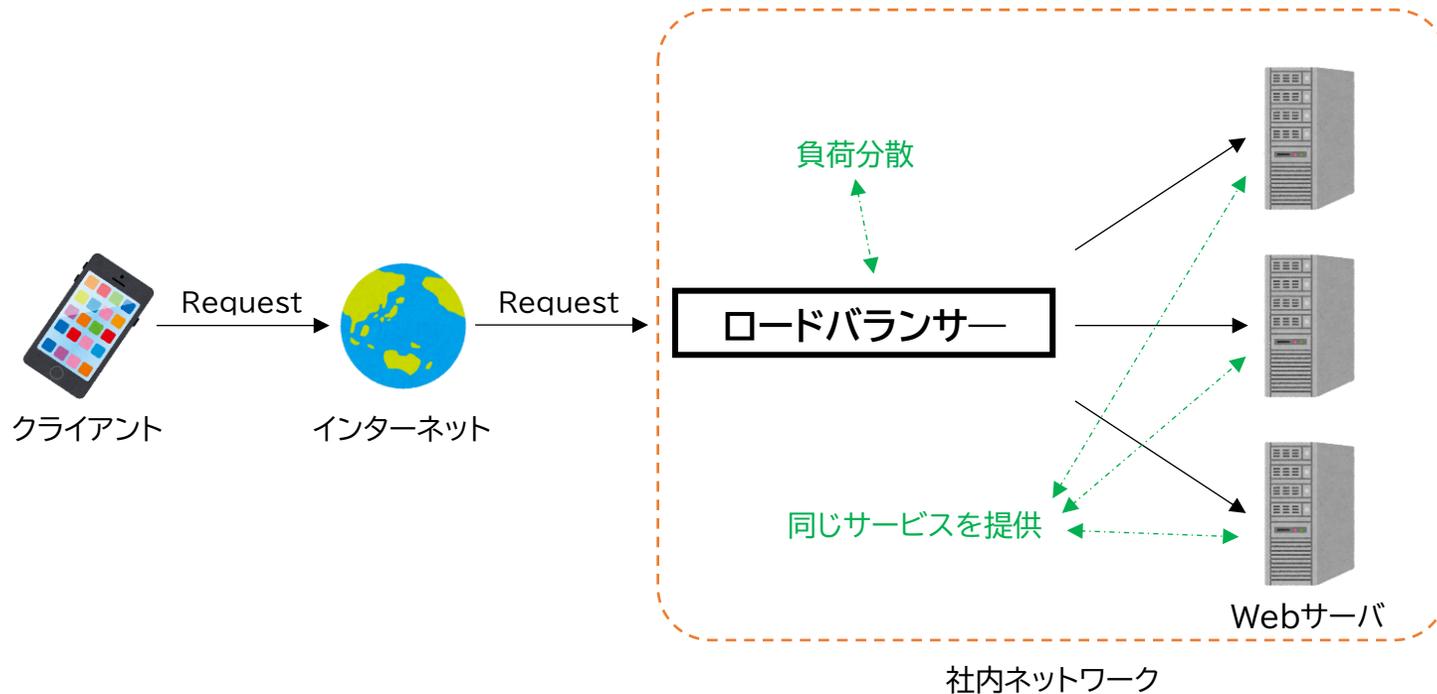


- 主な役割

- 攻撃からの保護
- SSL終端
- キャッシング

# ロードバランサーの役割

- ロードバランサーは、複数のサーバーにクライアントからのリクエストを分散させ、サーバー負荷の均一化を図る。



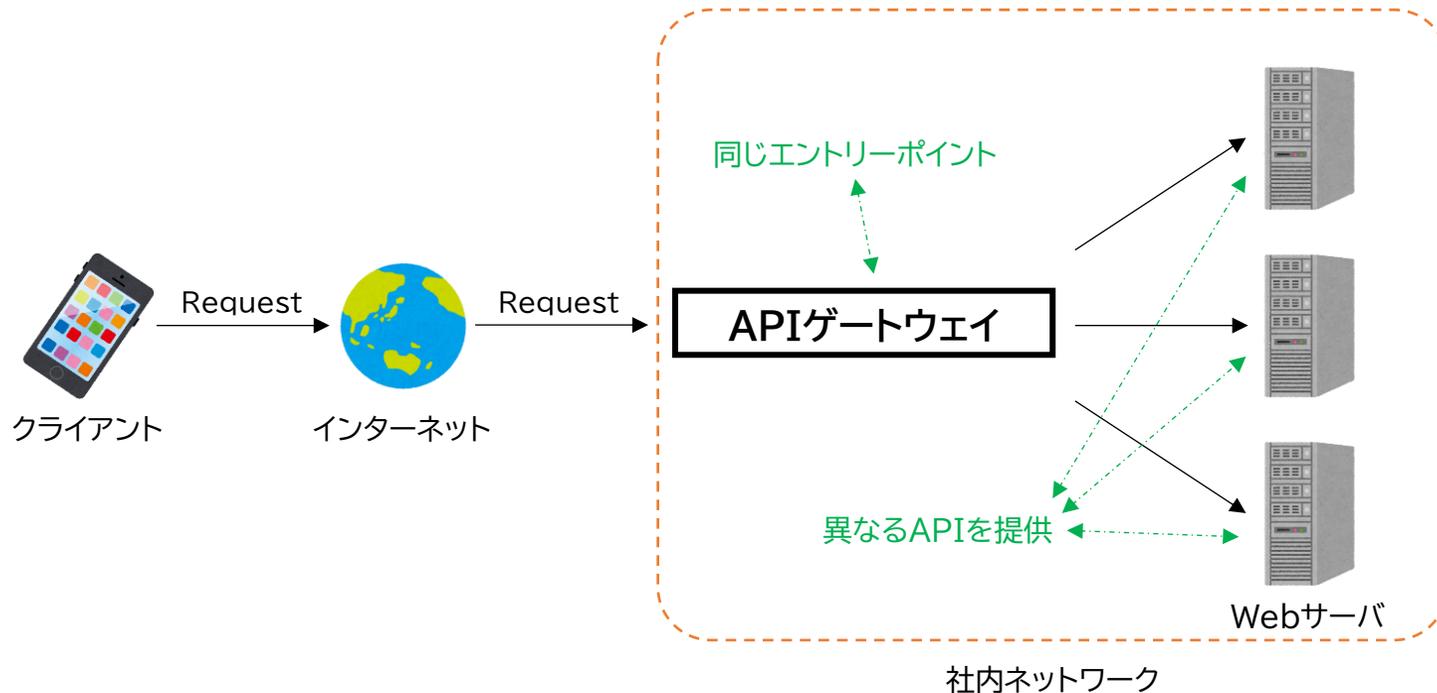
## ● 主な役割

- 負荷分散
- バックエンドのヘルスチェック
- セッション維持

⇒ ロードバランサーは、リバースプロキシの役割を兼ねることもできる。

# APIゲートウェイの役割

- APIゲートウェイは、複数の異なるAPIサーバーを管理対象として、APIに対するクライアントからのリクエストをコントロール/モニタリングする。



## ● 主な役割

- 認証/認可
- 流量制御
- モニタリング

⇒ APIゲートウェイは、リバースプロキシの役割を兼ねることもできる。

# APIゲートウェイの役割とは？

---

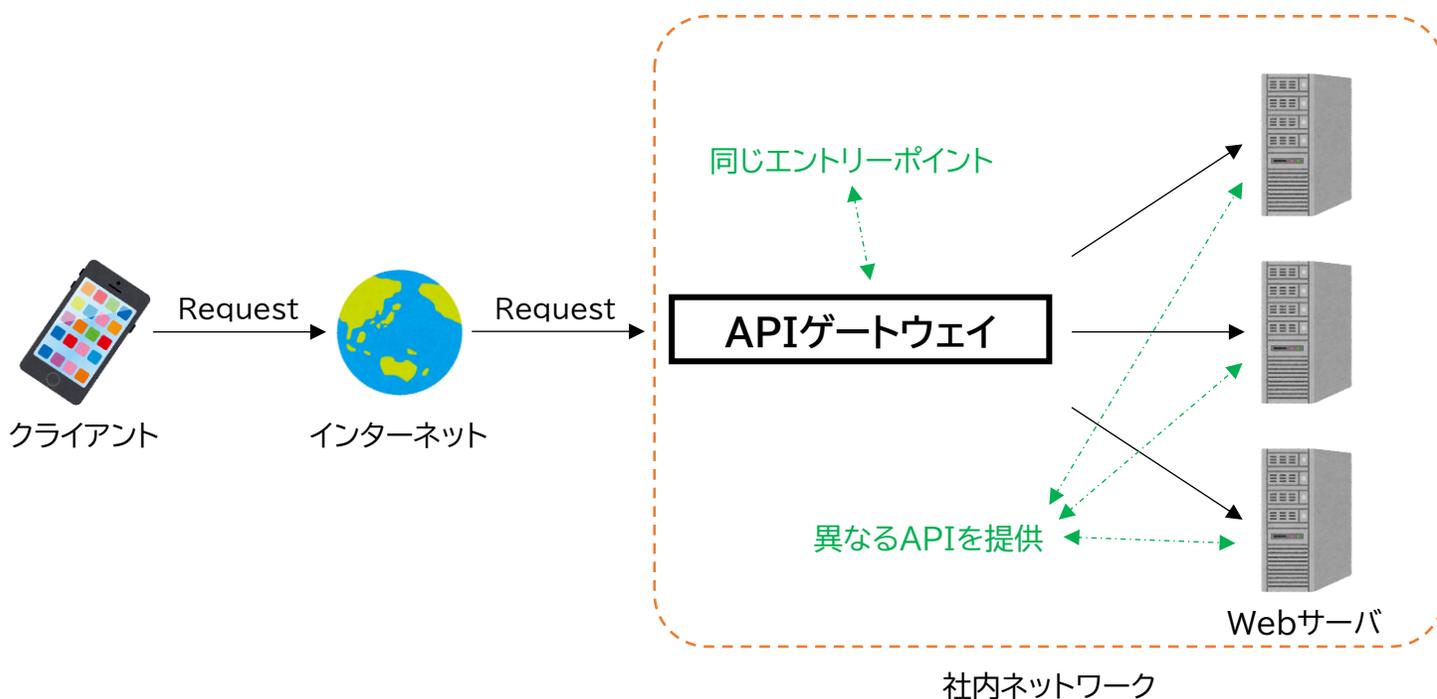
APIゲートウェイは、API管理機能を提供するWebサーバである。

1. リバースプロキシ：リクエストを仲介してセキュリティ強化等を図るWebサーバ
2. ロードバランサー：リクエストを分散させてシステムの可用性を向上させるWebサーバ
3. APIゲートウェイ：APIリクエストを仲介してシステムへの通信を管理するWebサーバ

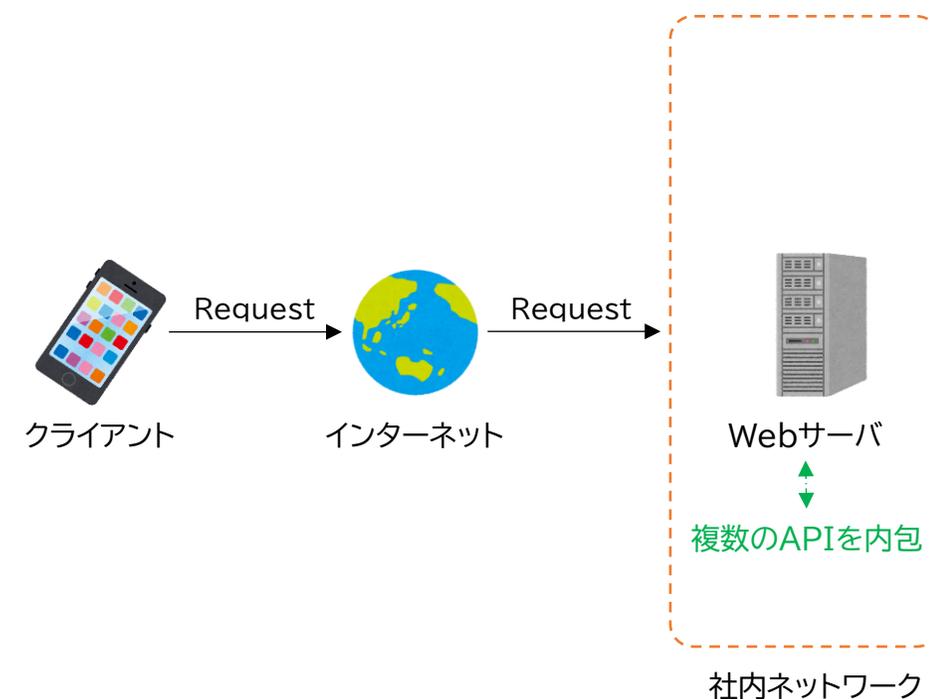
特に、マイクロサービスアーキテクチャを構築する場合によく利用される。

- システム構成要素間を疎結合にすることで、変更・改良に耐えうる設計が実現できる。
- 小規模システムで、無計画に利用すると全体構成が複雑化する恐れもある。

## マイクロサービス



## モノリシックサービス



- Kongを含むAPIゲートウェイは、クライアントから受信したリクエストに対して、特定のプログラムを実行した上で、適切なAPIサービスにリクエストを転送する。
- 各種プログラムは、kongではプラグインと呼ばれ稼働後も自由に着脱することができる。

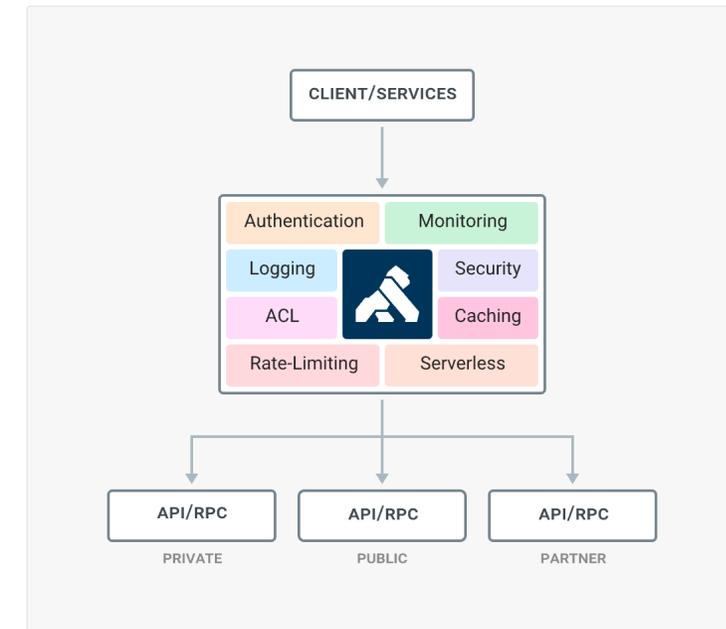
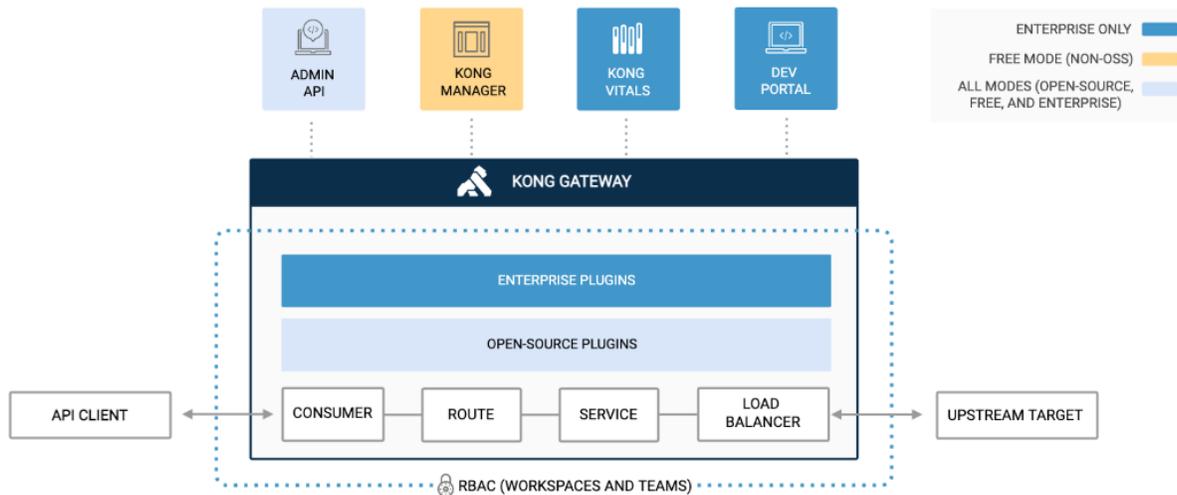
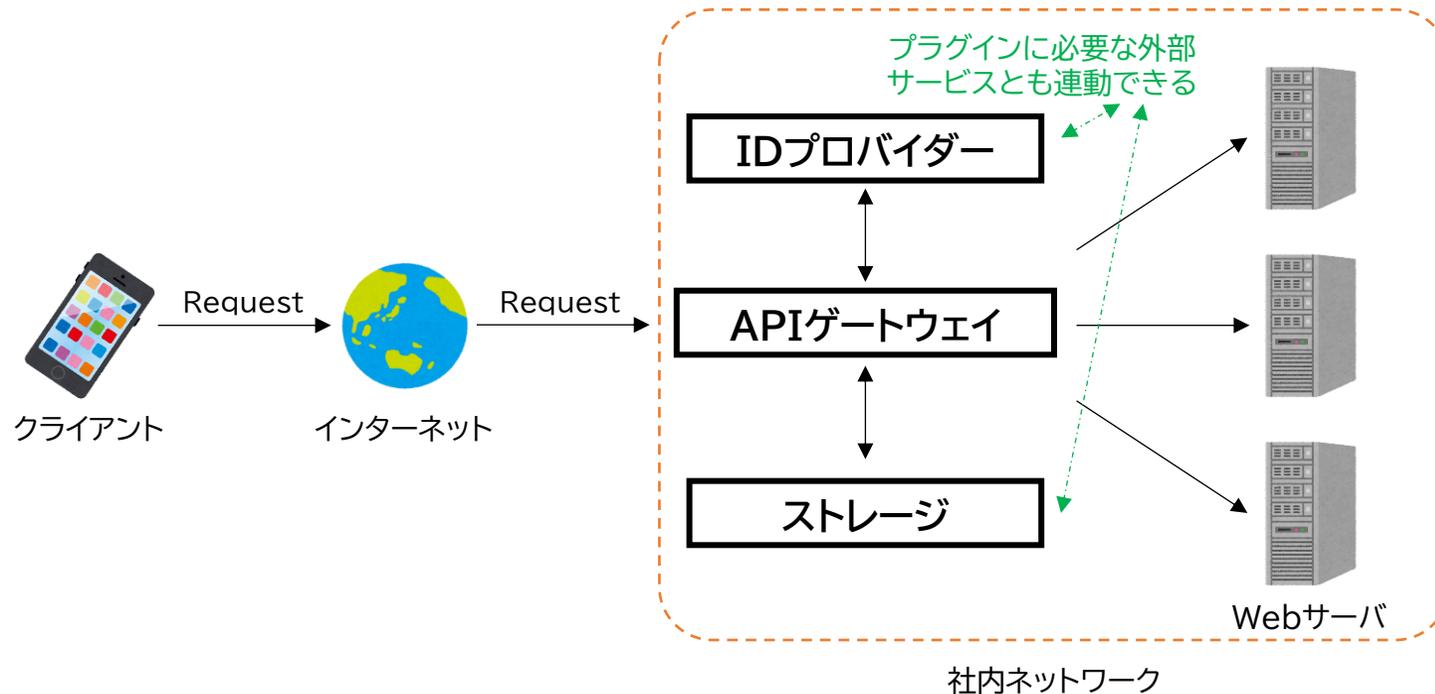


Figure 1: Diagram of Kong Gateway modules and how they relate to the foundational Gateway components. Requests flow from an API client into the Gateway, are modified and managed by the proxy based on your Gateway configuration, and forwarded to upstream services.

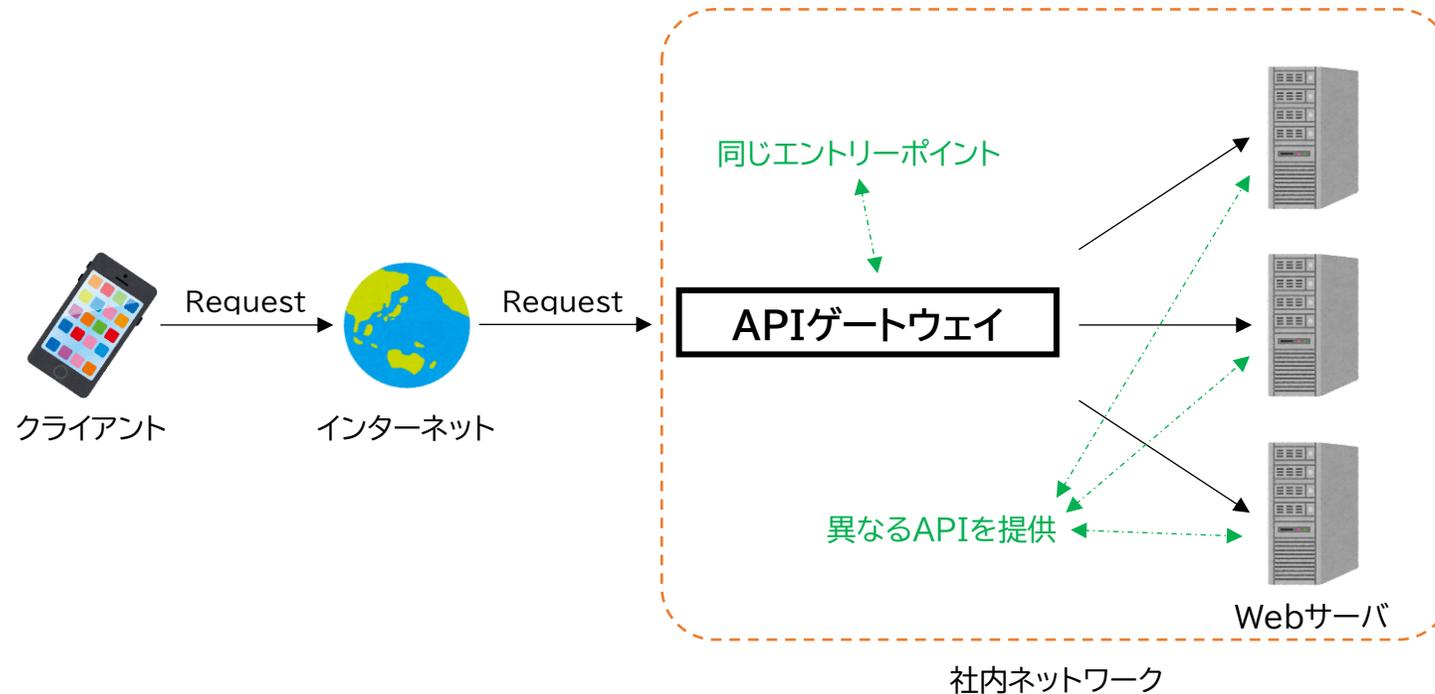
# APIゲートウェイが提供する代表的な機能

- ログイン: リクエストを見て、ログに記録して、転送する。
  - 認証: リクエストを見て、ID/PASSが適切であれば、転送する。
  - 流量制限: リクエストを見て、記録している既定回数以下であれば、転送する。
- …などプログラムさえ書けば技術的にはなんでも可能であるが、多くは簡易な共通機能に限られる。



# APIゲートウェイの機能① リクエストの集約

- クライアントからのリクエスト導線を一本化できる。
  - リクエストの全量が把握できる
  - APIに共通した機能(認証など)を同時追加できる
  - システム全体のセキュリティホールを最小限に抑えられる

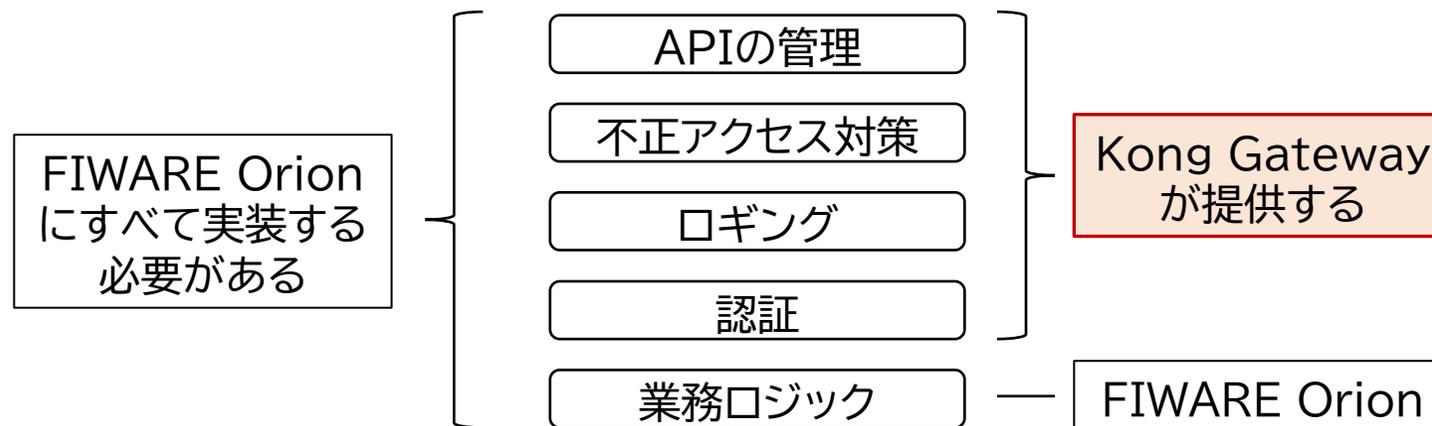


## APIゲートウェイの機能② APIへの機能追加

- API GatewayをAPIのフロントに配置することで、システム上必ず設定しなければならないセキュリティ機能や運用管理機能を追加することができる。
- APIゲートウェイは大型システムの管理だけではなく、機能が不足した個別サービスに対して機能拡充させるためにも利用することができる。

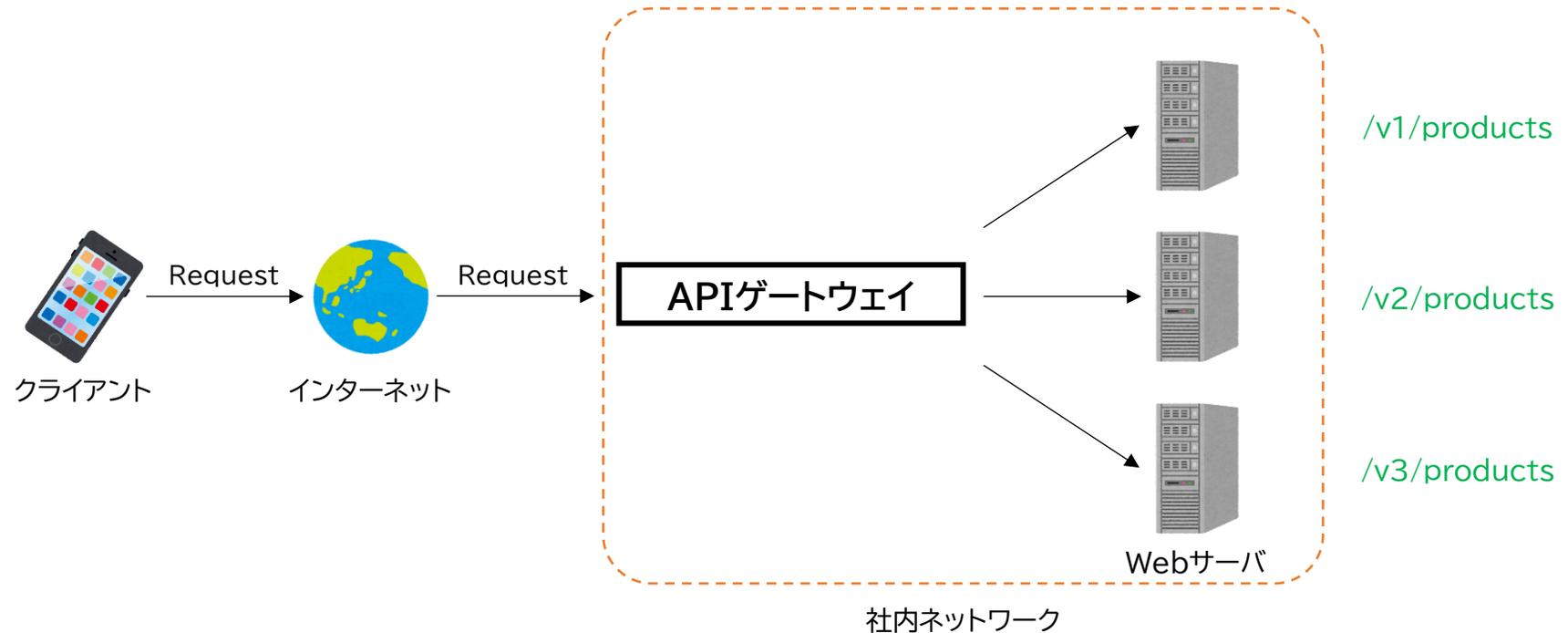
API Gatewayなしの場合

API Gatewayありの場合



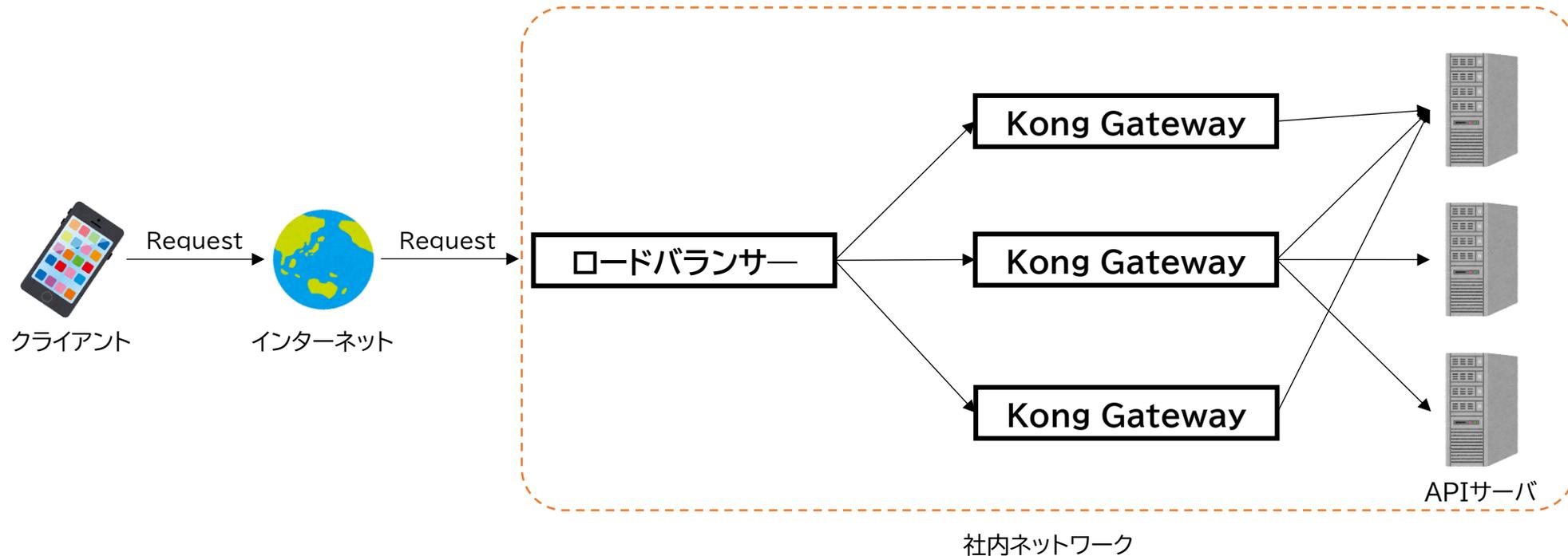
# APIゲートウェイの機能③ APIのバージョン管理

- 異なるバージョンのAPIを並列提供することができる。
  - v1のリクエストはv1のAPIサーバへ転送する。v2以降も同様。
  - APIがそれぞれ独立しているため無停止でバージョン以降が実施できる。



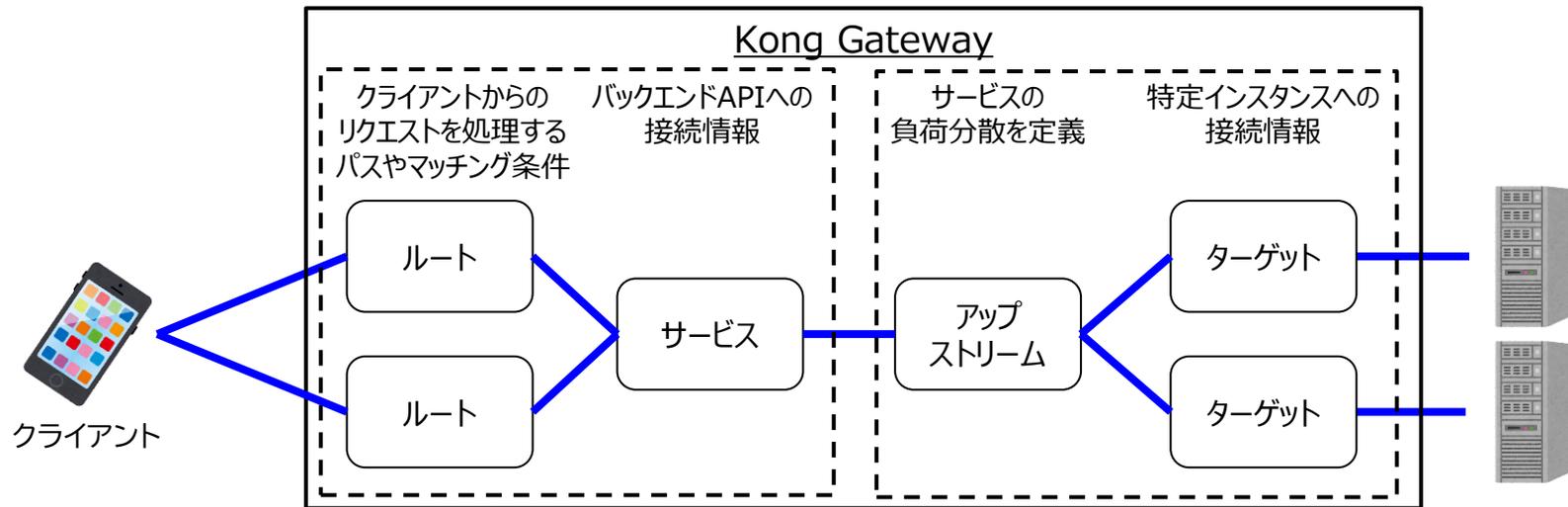
# Kong Gatewayの特徴① Kong自体も冗長化できる

- Kong Gatewayはコンテナ形式でも提供されていて、並列稼働することもできる。
  - ロードバランサー等と組み合わせることで、高可用性が確保できる。
  - OSSであるため自由にサイジングや稼働環境を設定することができる。
- 自力でホストしないといけない点は、クラウドサービスと比較した場合のデメリット。



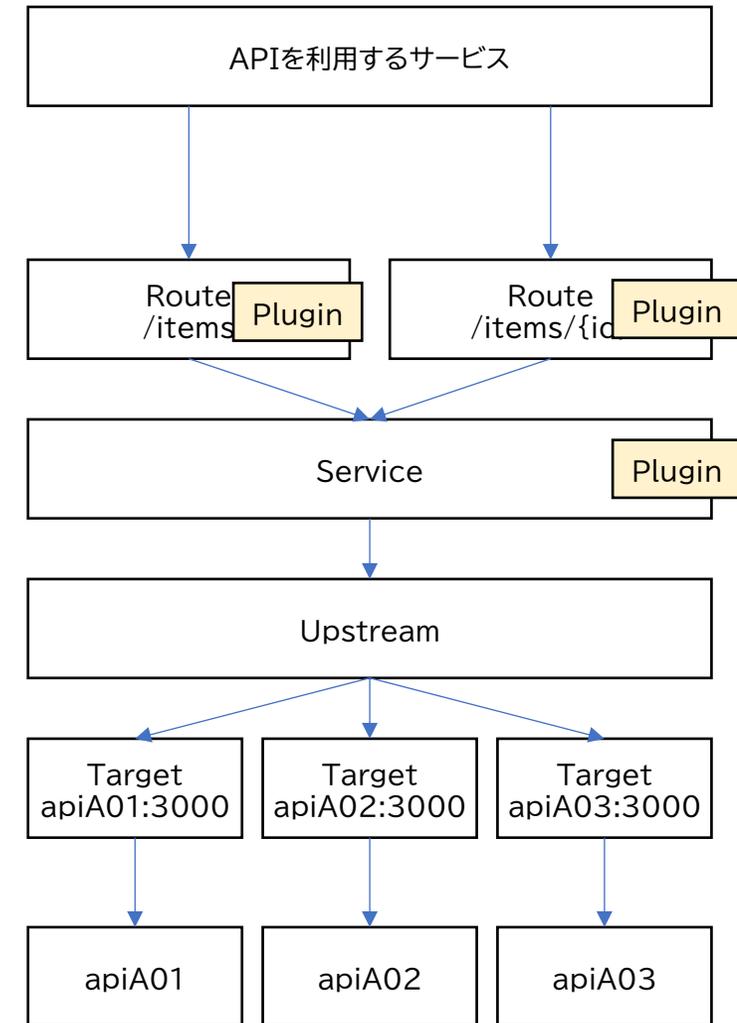
## Kong Gatewayの特徴② 柔軟にルーティング設定できる

- 稼働開始後も、Kong自体のAPIで設定を追加/変更/削除することができる。
- リクエストを受信するルーティングパスを複数かつ詳細に設定できる。
- リクエストを転送先の設定を複数かつ詳細に設定でき、APIサーバーに対して、ロードバランスやヘルスチェック(ダウン時の自動切り離し)も設定できる。(リバースプロキシとしてもロードバランサーとしてもふるまえることができる)



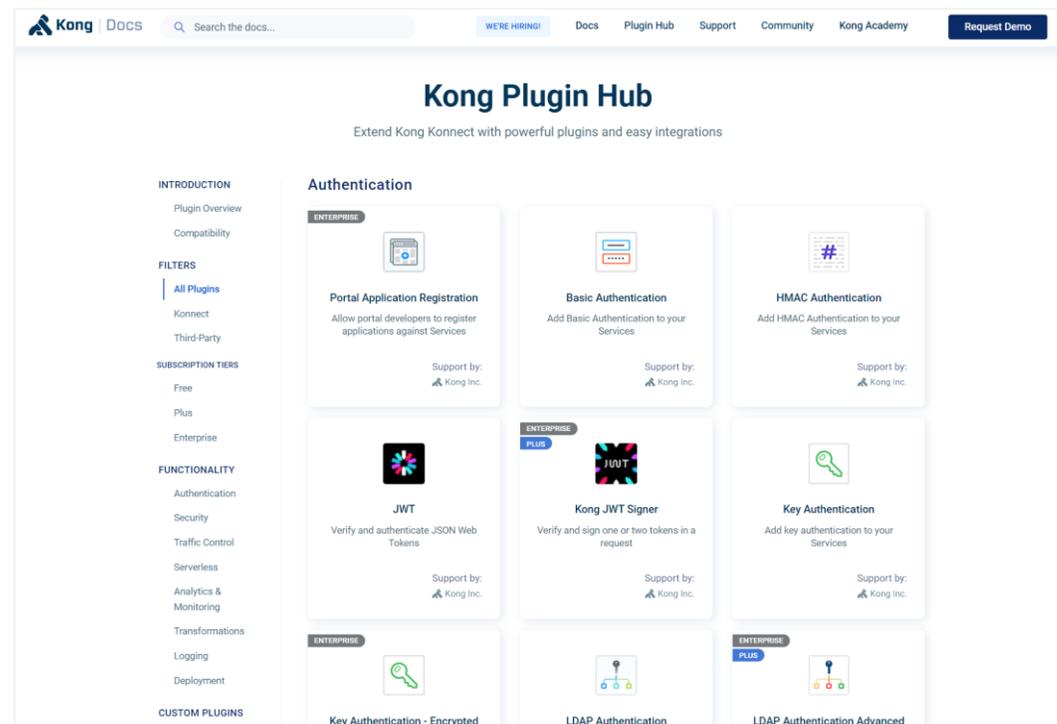
# [参考] Kong Gatewayの内部リソース用語集

- Service :APIサービス1つに相当
- Route :メソッドとパスの組に相当
- Upstream :Kongの内部ロードバランサ
- Target :APIサーバ1つに相当
- Consumer :API利用ユーザに相当
- Plugin :Service、Route、Consumer、  
またはグローバルに動作する  
リクエスト / レスポンスの追加処理



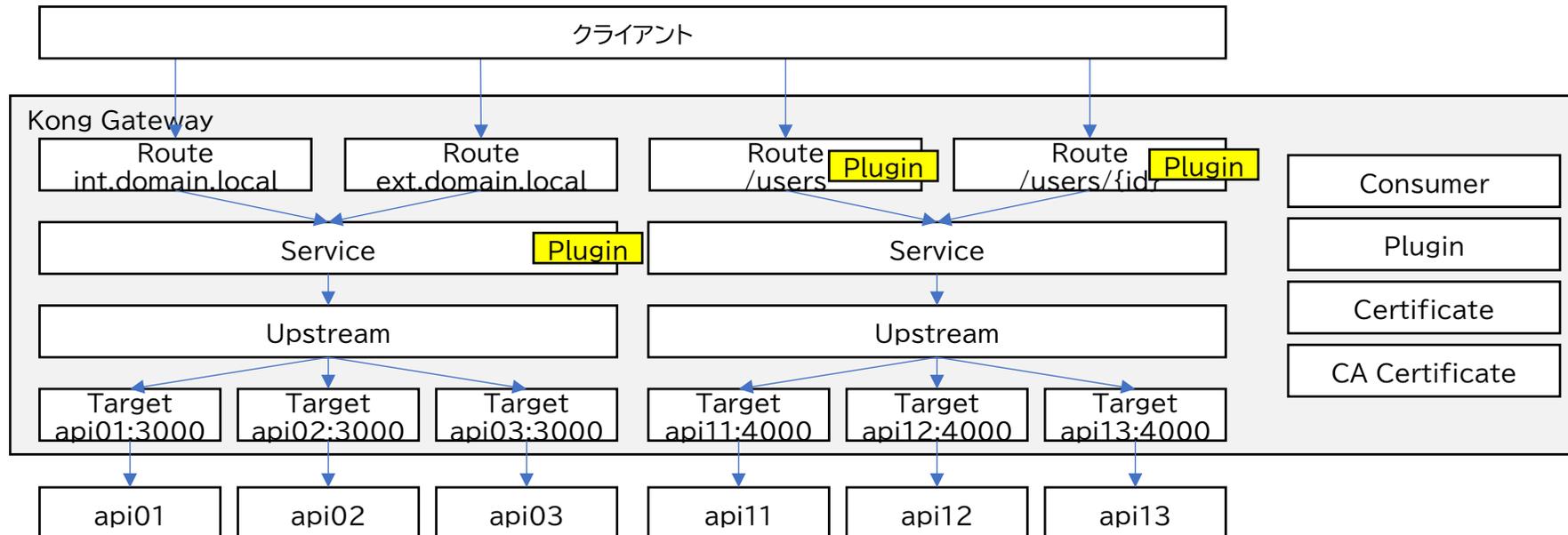
# Kong Gatewayの特徴③ 標準プラグインが豊富

- 一般的システムでよく設定する各種機能は、OSS版Kongにバンドルされている。
- プラグインは、APIゲートウェイ全体に適用することが可能な他、ルート単位、ターゲット単位、(認証の)ユーザー単位などきめ細やかに設定できる。稼働後もAPIで着脱できる。
- クラウドベンダー提供のサービスよりも、設定が直感的でわかりやすく学習コストが低い。



# [参考] Kong Gatewayの内部構成例

- システム上のコントロール状況を、モデル図のようにきれいにマッピングすることができる。
- システムに導入することで、セキュリティ面、運用管理面、監査面等で特にメリットを見出すことができる。
- Kongの内部処理のレイテンシはごくわずかであり、パフォーマンス面で優れている。

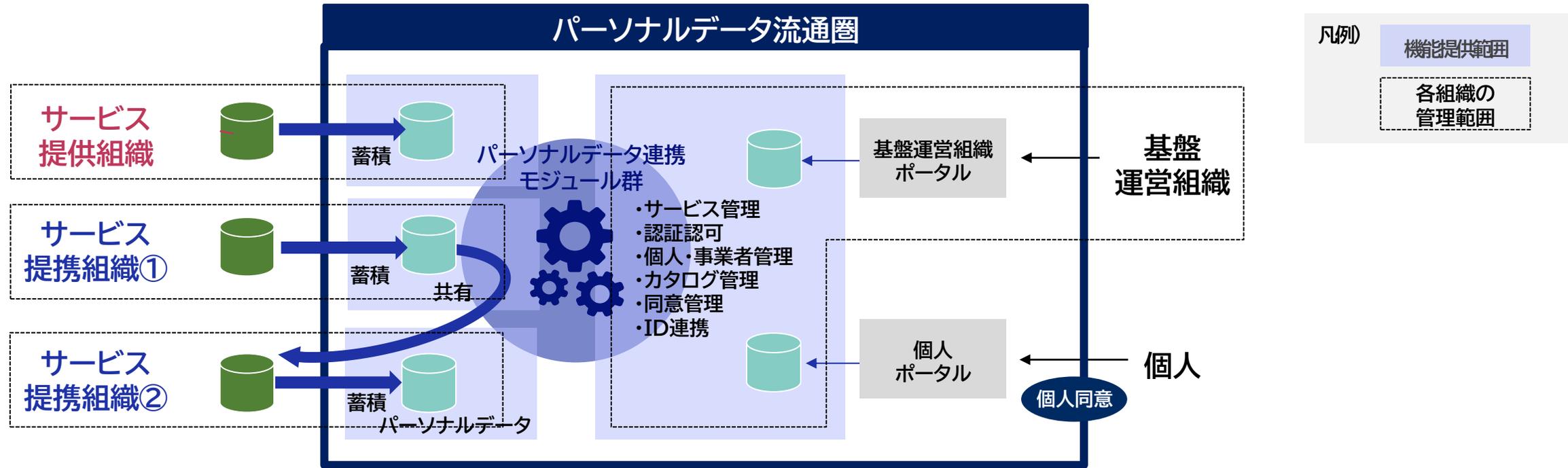


# パーソナルデータ連携モジュールの概要と公開情報



# パーソナルデータ連携モジュールの概要

- パーソナルデータ連携モジュールとは、サービス事業者が保有しているパーソナルデータを、個人同意に基づき組織をまたがって共有できる、トラスト層/データ連携層の機能群。

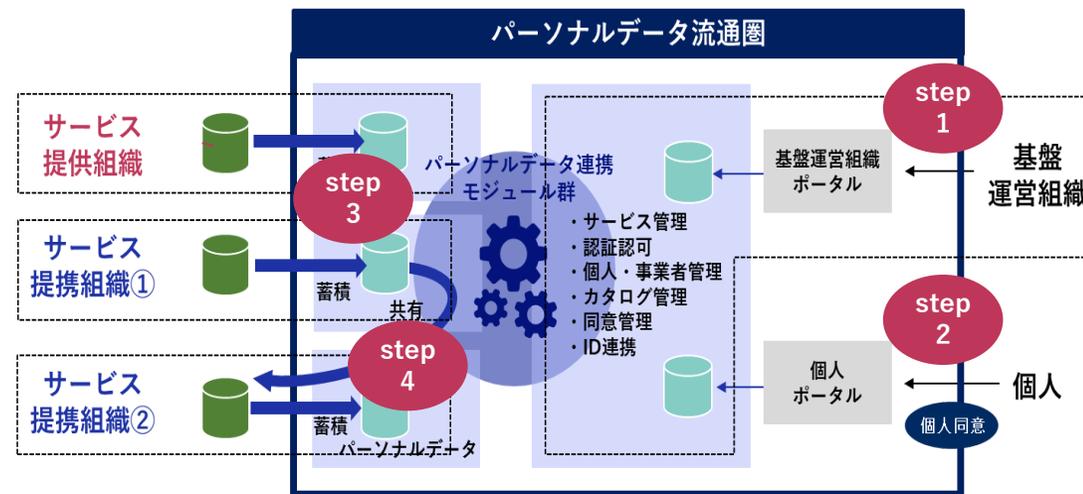
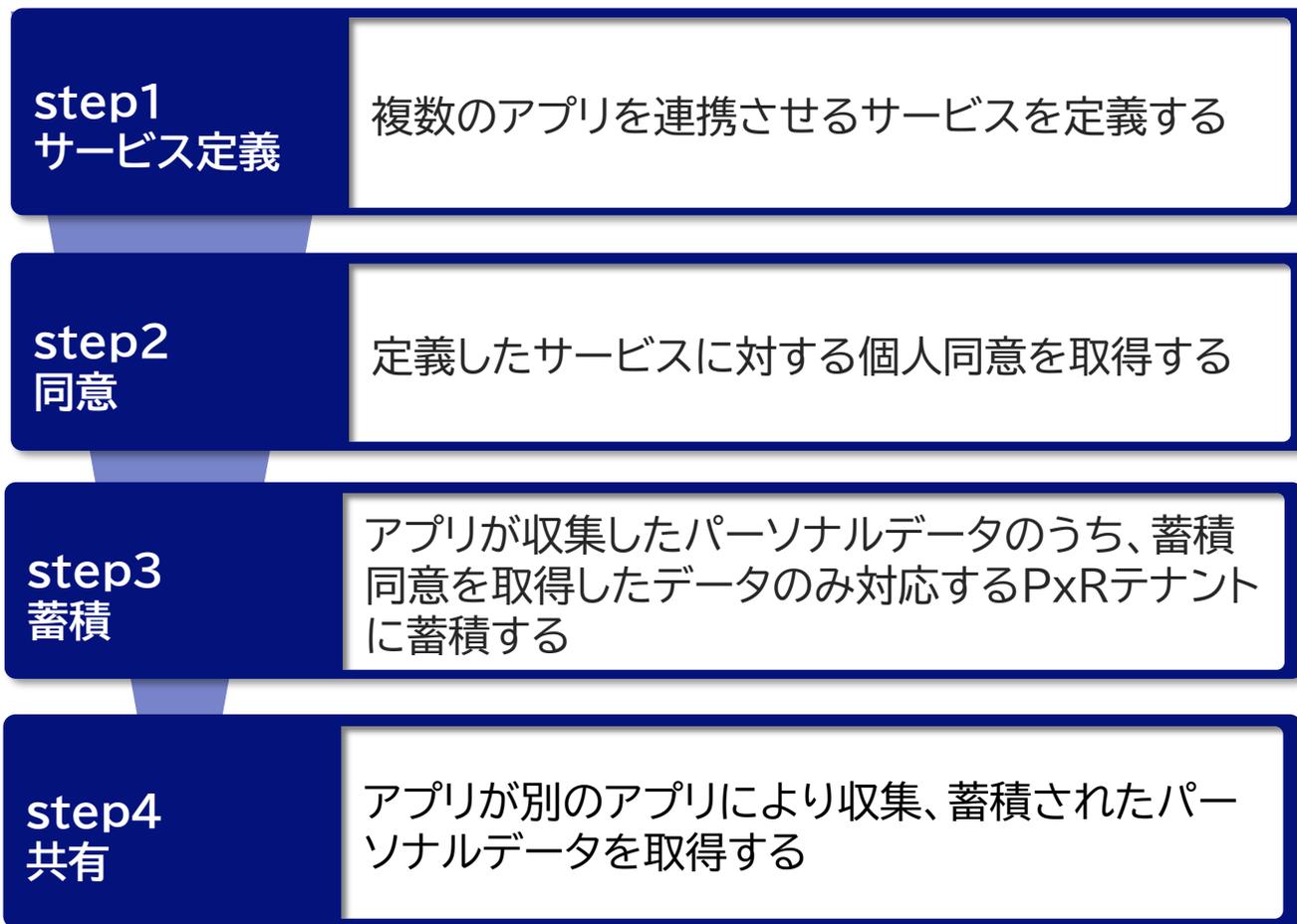


- 蓄積: 個人の同意に基づき、パーソナルデータを他組織からアクセス可能な状態にすること
- 共有: 個人の同意に基づき、他組織が蓄積したパーソナルデータにアクセスすること

詳細は[パーソナルデータ連携モジュール説明資料](#)を参照

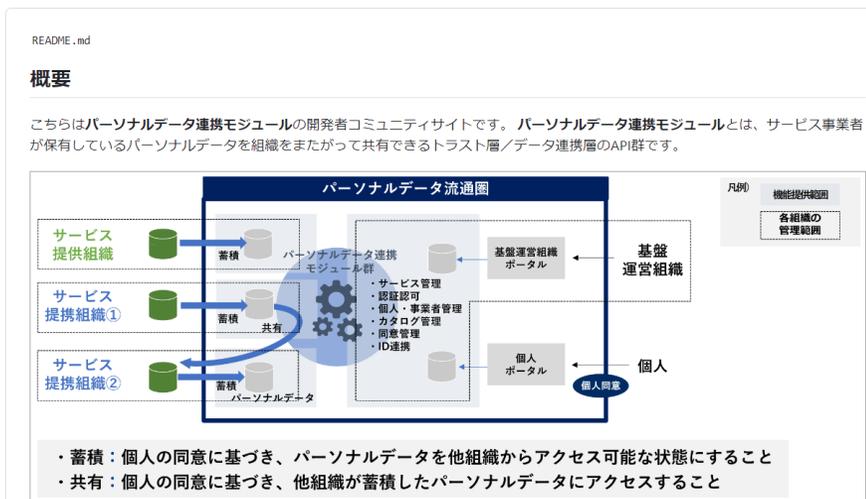
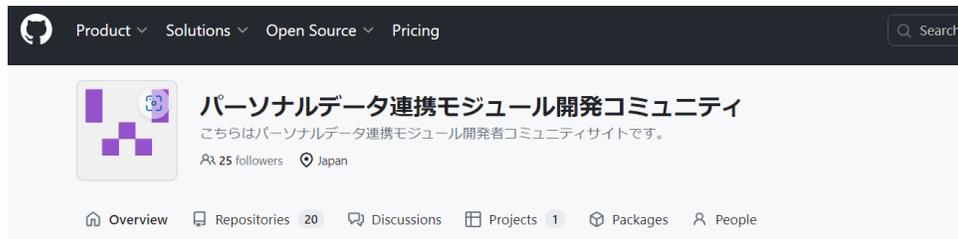
# パーソナルデータ流通の概観

- 4つのstepを通じて、各アクターとパーソナルデータ連携モジュール群(API群)が連携し協調動作することでパーソナルデータ流通圏を形成。



# これまでのコミュニティ活動の成果物について

- 過去に開催した技術セミナーは、DSAのホームページにアーカイブ動画が公開されている。
- モジュールの説明資料等はGitHubにも格納されている。



パーソナルデータ連携モジュールの概念図

### 活動情報

#### 開発活動

プロダクトの状況は、[メンテナンス用プロジェクト](#)で管理されています。バグや質問への対応状況を確認することができます。また、本コミュニティは、運営および開発メンバーを随時募集しています。メンバーとして参加を希望する方は、[メンバー加入希望](#)

[パーソナルデータ連携モジュール開発コミュニティ・GitHub](#)

### ◆概要資料:

[pxr-linkage/doc/セミナー/20221227 パーソナルデータ連携モジュール 説明会資料/パーソナルデータ連携モジュール 説明資料 v3.pdf at main · Personal-Data-Linkage-Module/pxr-linkage · GitHub](#)

### ◆自治体職員様向け動画、資料:

[【開催レポート】第2回 エリア・データ連携基盤 技術セミナー | 一般社団法人データ社会推進協議会\(DSA\) \(data-society-alliance.org\)](#)

### ◆構築事業者様向け動画、資料:

[【開催レポート】第3回 エリア・データ連携基盤 技術セミナー | 一般社団法人データ社会推進協議会\(DSA\) \(data-society-alliance.org\)](#)

### ◆接続事業者様向け動画、資料:

[【開催レポート】第4回 エリア・データ連携基盤 技術セミナー | 一般社団法人データ社会推進協議会\(DSA\) \(data-society-alliance.org\)](#)

### ◆その他関連情報DSA:

[推奨モジュールの概要 | 一般社団法人データ社会推進協議会\(DSA\) \(data-society-alliance.org\)](#)

