

第1回 エリア・データ連携基盤 技術セミナー (Kong Gateway / FIWARE Orion編)

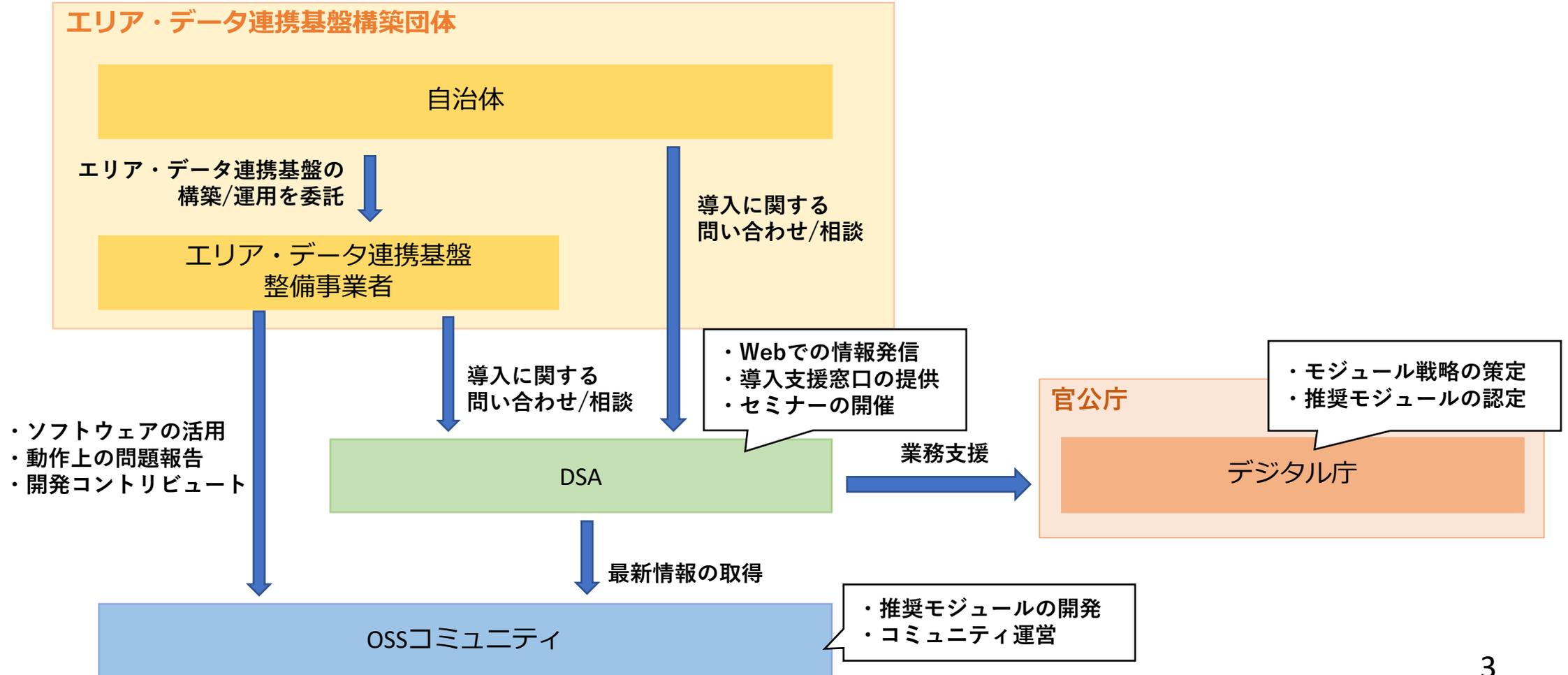
2023年1月20日



- DSAの活動について
- 関連する技術知識 / 用語について
- エリア・データ連携基盤のシステム構成要素
- Kong Gatewayについて
- FIWARE Orionについて
- 構築デモ
- 運用上の注意点
- まとめ

エリア・データ連携基盤に関するDSAの活動と位置づけ

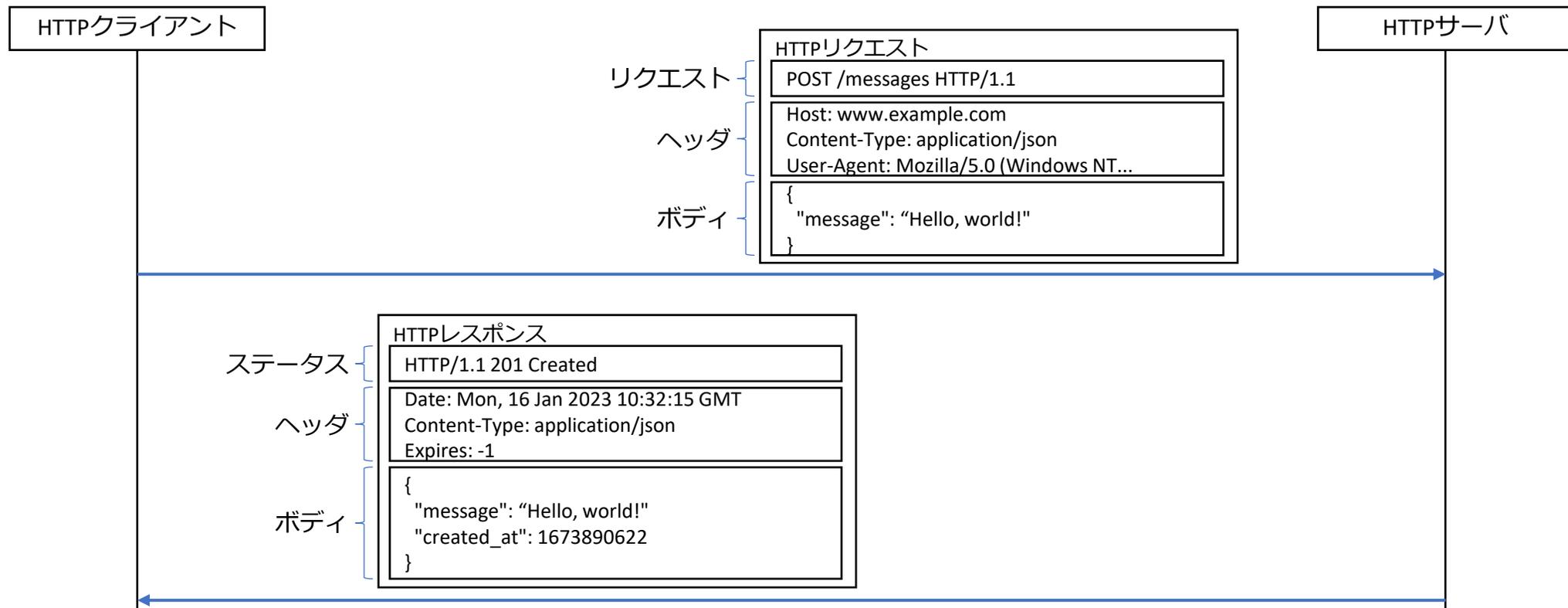
- DSAは、デジタル庁が認定した推奨モジュールに関する情報発信やセミナー開催などの普及促進活動を担うとともに、デジタル庁のモジュール戦略策定や推奨モジュールの認定に関する支援を行います。



エリア・データ連携基盤の構築に 関連する技術知識 / 用語

- HTTP / HTTPS
 - API / Web API / REST API / NGSIv2
 - REST API
 - OpenAPI仕様
 - APIゲートウェイ
 - コンテキストブローカー
 - NGSIv2
 - コンテナ / Kubernetes
-
- 参考 : APIを使うアプリケーションの設計

- クライアント～サーバ間のコネクション（HTTPSの場合はSSL/TLSにより暗号化されている）上でメッセージをやりとりするプロトコル。Webブラウザをはじめとして、インターネットを介した通信を行う多数のアプリケーションが採用しています。



- 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。Web API、REST API、NGSIv2は下図のような包含関係にあります。
- TwitterやFacebookなどサービスが提供するAPIは、REST APIとして設計されるのがデファクトスタンダードとなっています。

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、URLでリソースを、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設計思想です。近年公開されているWebサービスのAPIは、ほとんどがREST APIに準拠した設計となっています。
- GETのパラメータはURL内のクエリパラメータで、それ以外のパラメータはBodyに記載するのが一般的です。

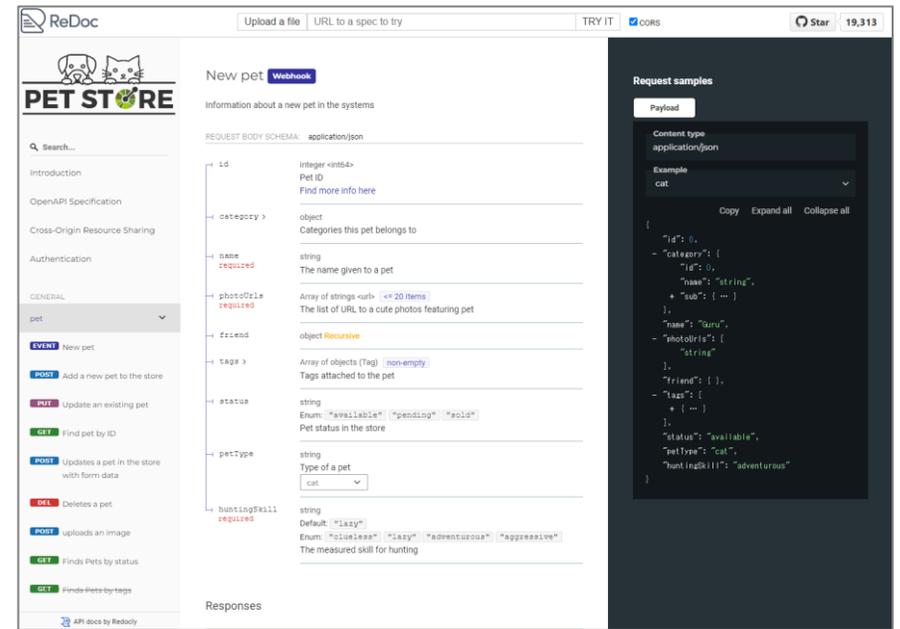
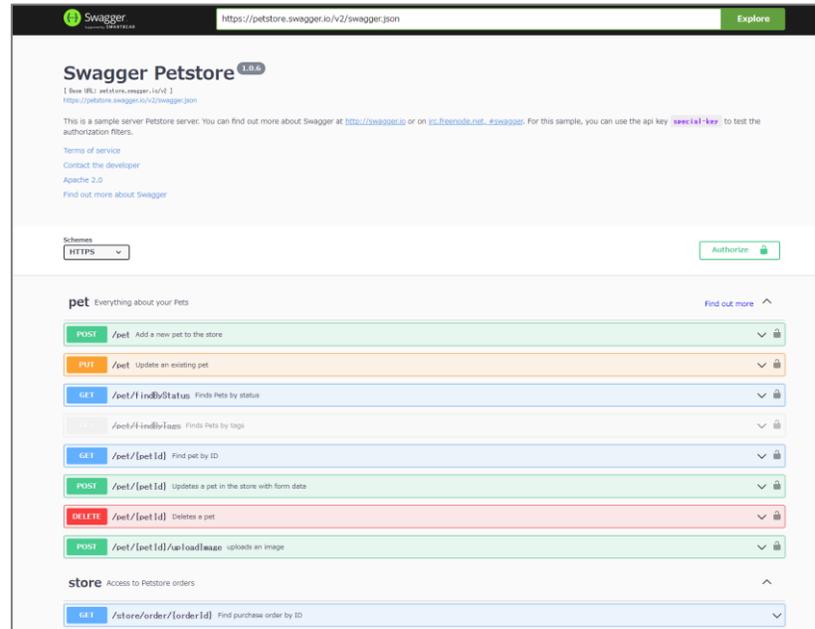
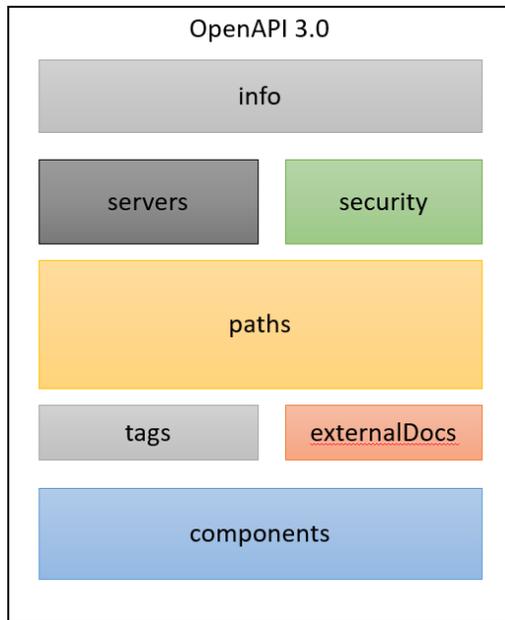
▼ 記事とコメントが管理できるサービスのAPI例

用途	Method	Path
記事一覧の取得	GET	/articles
記事の登録	POST	
記事の参照	GET	
記事の更新	PUT	/articles/{articleId}
記事の部分更新	PATCH	
記事の削除	DELETE	
コメント一覧の取得	GET	/articles/{articleId}/comments
コメントの登録	POST	
コメントの参照	GET	/articles/{articleId}/comments/{commentId}
コメントの更新	PUT	
コメントの部分更新	PATCH	
コメントの削除	DELETE	

▼ HTTPメソッドとCRUDの対応

	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

- REST APIの設計は、OpenAPI Specification（API設計書の書き方の仕様）に準拠した形で記載するのがデファクトスタンダードとなっています。
- OpenAPI Specファイルは、SwaggerUIやRedoc等の外部ツールに読み込ませることでWebブラウザで見やすく表示することができます。

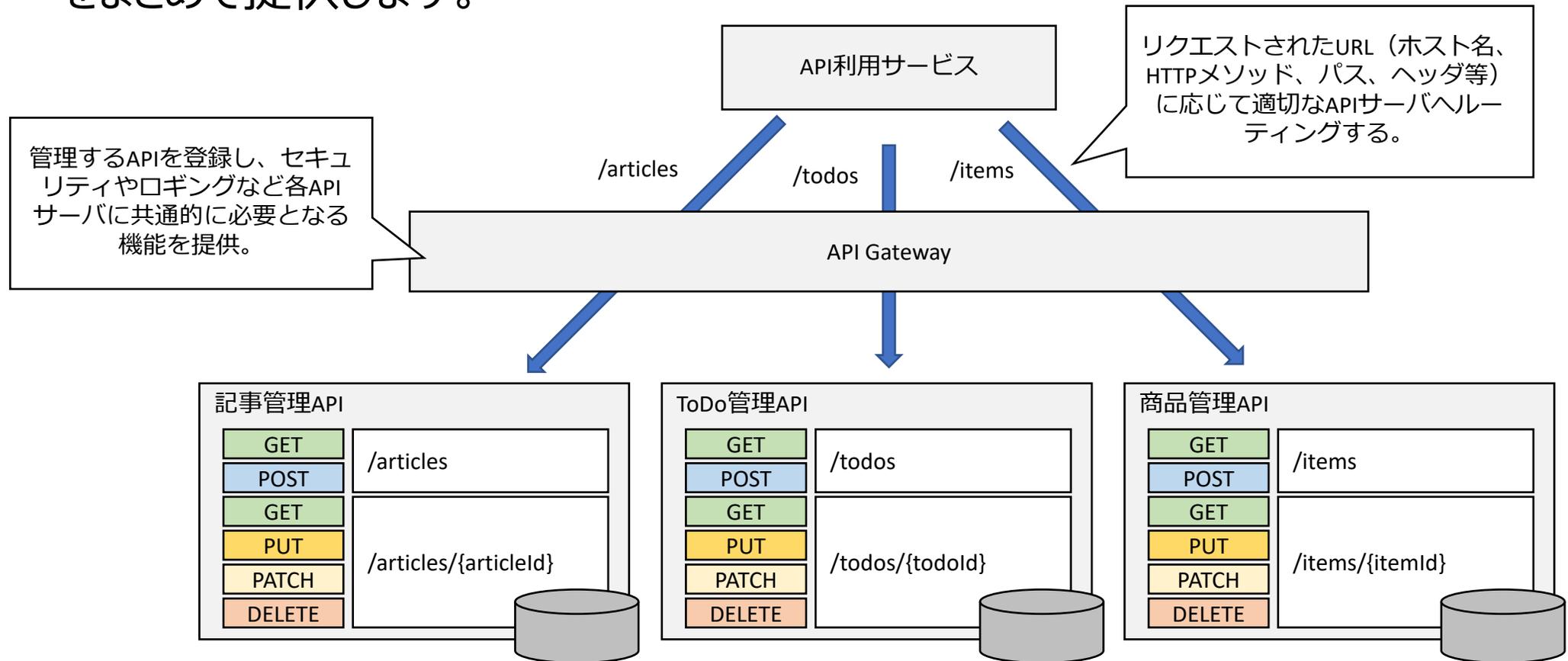


▲ OpenAPI Specの構造。
実際はYAMLかJSONで記載される。

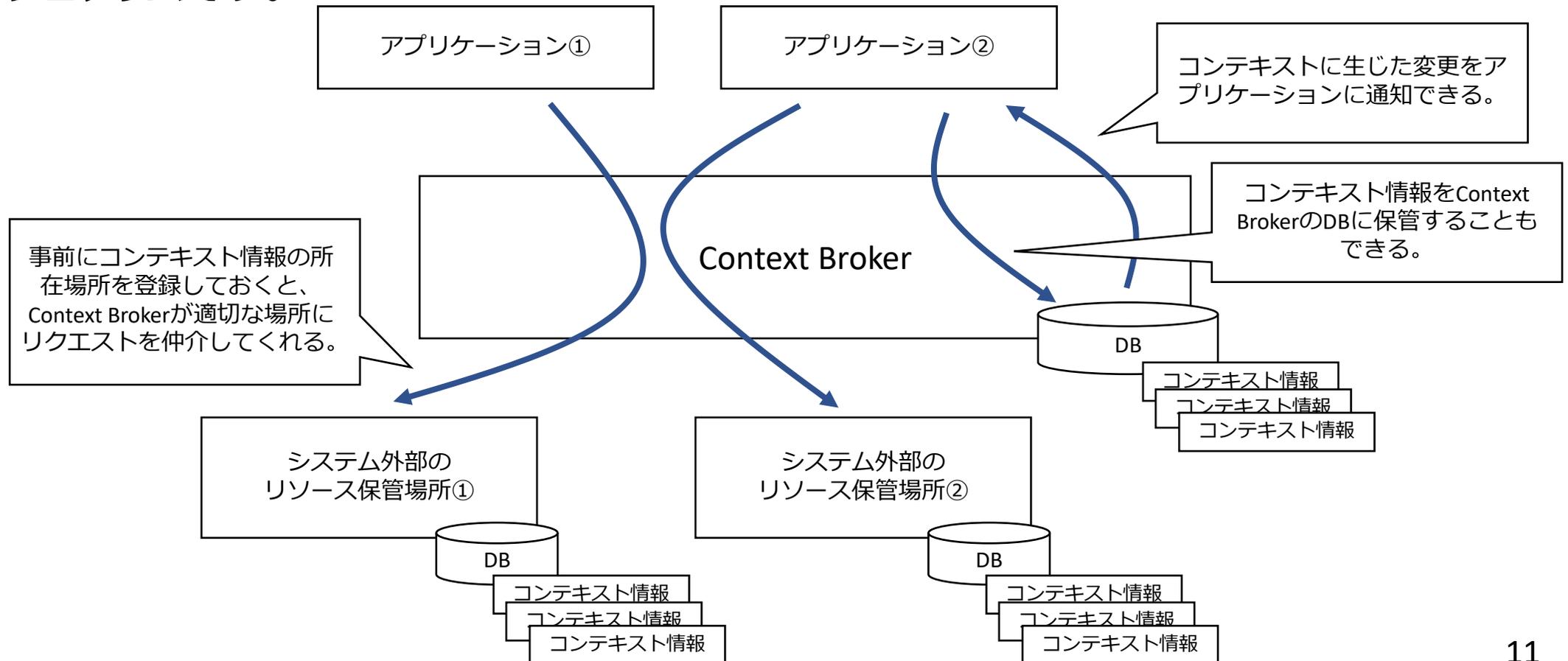
▲ SwaggerUIでAPI仕様を表示した例。

▲ RedocでAPI仕様を表示した例。

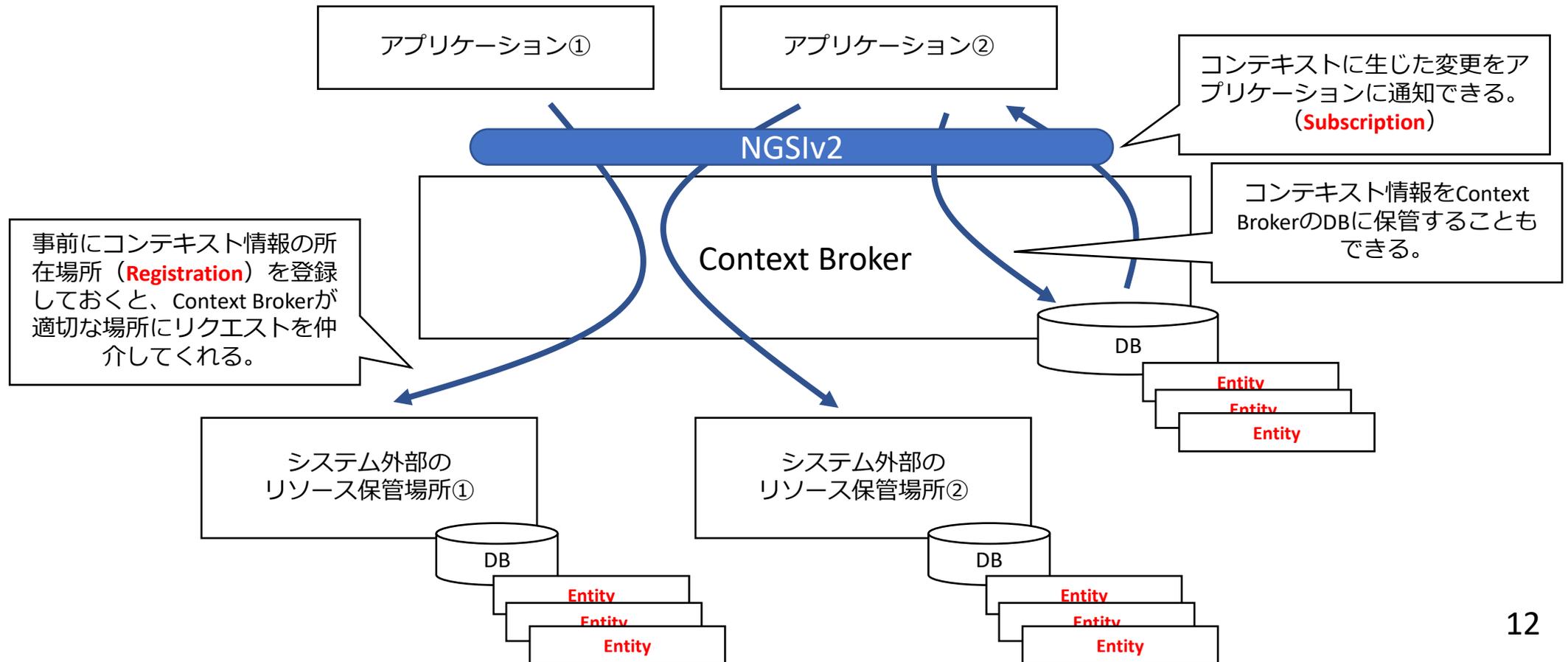
- APIゲートウェイは、複数のAPIサーバが提供されるとき、そのリクエストの受け口をまとめるリバースプロキシです。各APIで共通に必要なセキュリティやロギング等の機能をまとめて提供します。



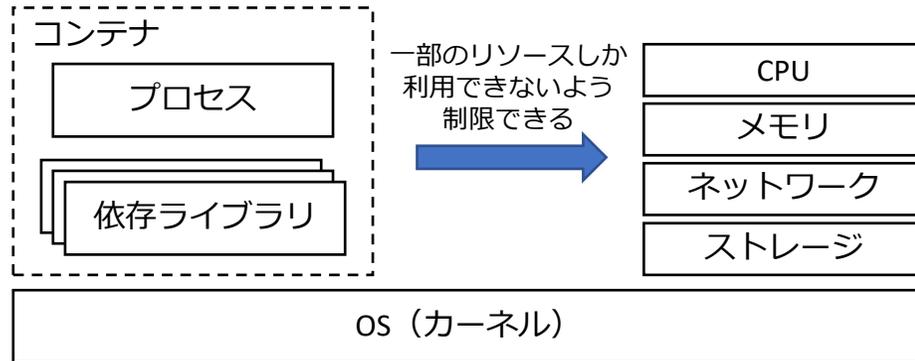
- コンテキストブローカーは、様々な場所に保存されたコンテキスト情報を管理し、複数のプログラムやサービスからのリクエストをデータ提供元へ仲介、利用できるようにするプログラムです。



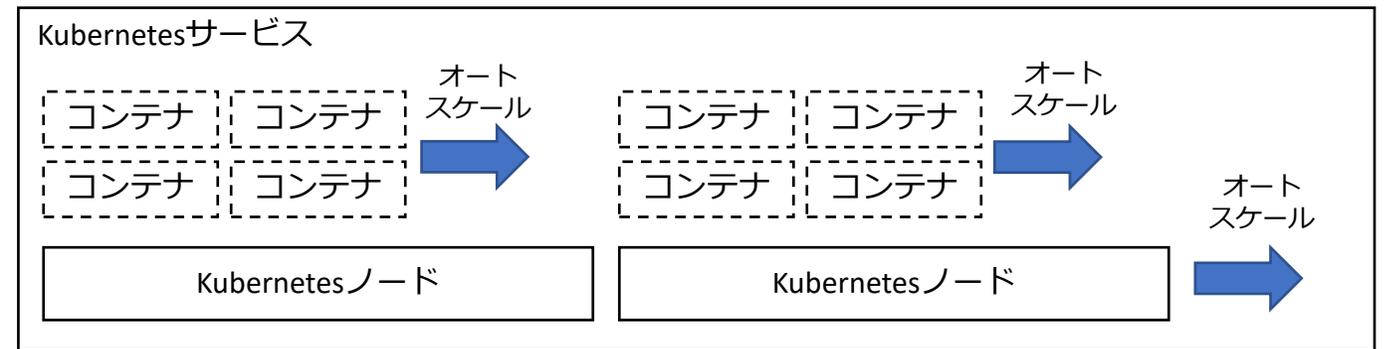
- NGSiv2は、コンテキストブローカーの1つであるFIWARE Orionが採用しているAPI仕様です。Entity、Registration、Subscriptionの3種類の主要リソースを組み合わせて情報が管理されています。



- コンテナは、プログラム本体とその依存ライブラリをまとめ、アクセスできるリソースを限定したプロセスです。コンテナ間も名前空間が分離されています。OSのブートが不要なため起動が早く、メンテナンス性のよいシステムを構築できます。
- Kubernetes（主要なパブリッククラウドでマネージドサービスとして提供されています）を使うとコンテナ運用の一部を自動化できます。
- データの不揮発化が必要な場合はマネージドサービスの利用を検討します。

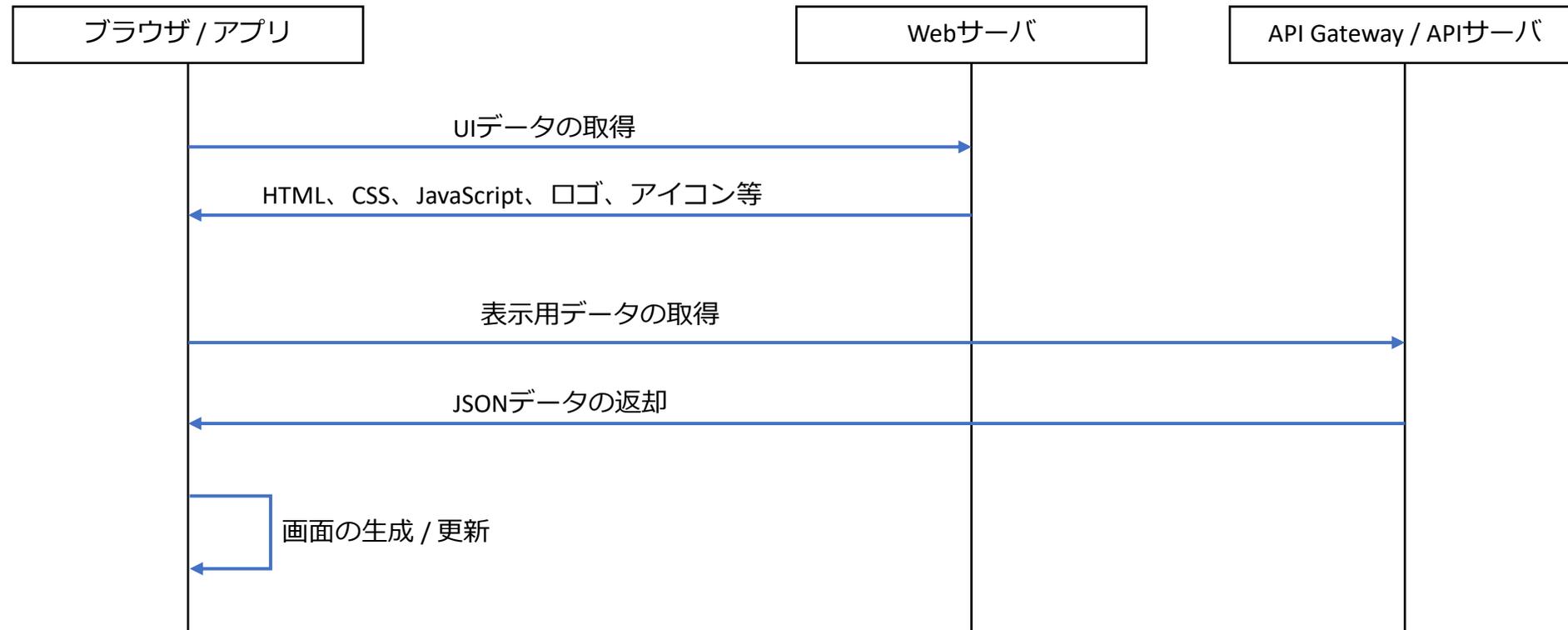


▲ コンテナはカーネルを共有しているためVMと比べて起動速度が早い。コンテナ内のプロセスからは独立した空間で稼働しているように見える。



▲ Kubernetes上にコンテナをデプロイして利用するイメージ。各コンテナはKubernetesノード上にデプロイされ、プロセスがクラッシュすると自動で新しいコンテナが作成されたり、負荷に応じてオートスケールさせることができる。

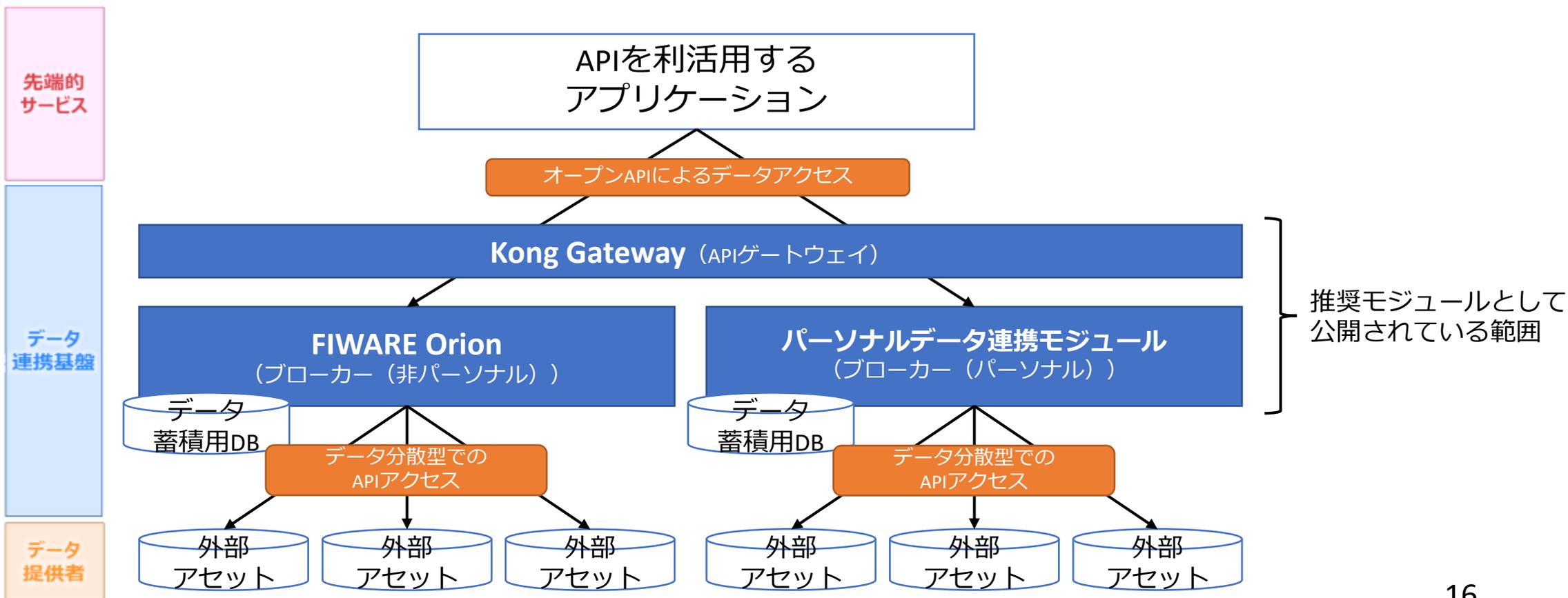
- API自体でやりとりするデータはJSONが主体となるので、入力ボックスやボタンなどのUIはアプリケーション側でコードを持っておきます。ブラウザ上で動くものはSPA (Single Page Application) と呼ばれます。



エリア・データ連携基盤の システム構成

エリア・データ連携基盤のシステム構成例 (推奨モジュールに関連する領域)

- APIを活用するアプリケーションは、APIゲートウェイを通してブローカー（非パーソナル）またはブローカー（パーソナル）にアクセスし、必要なデータを取得します。この際、APIゲートウェイ部分で必要な認証・認可やロギングなどの制御が行われます。



Kong Gateway (APIゲートウェイ)

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Webサーバ「Nginx」をベースとして開発された、複数のAPIを管理できるAPIゲートウェイです。プラグインで機能拡張な設計で、複雑な要件に対しては自作したプラグインを組み合わせて利用可能です。

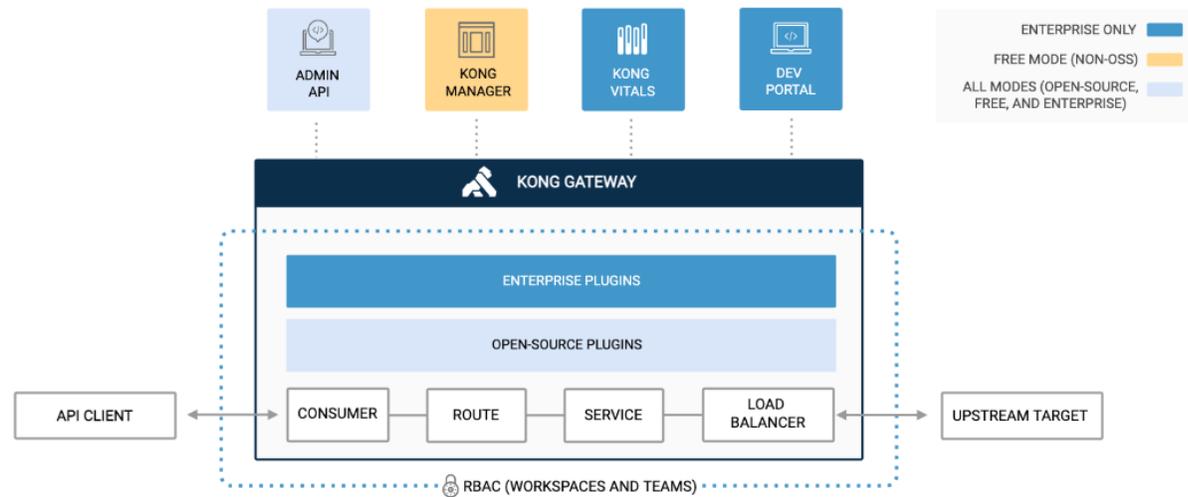
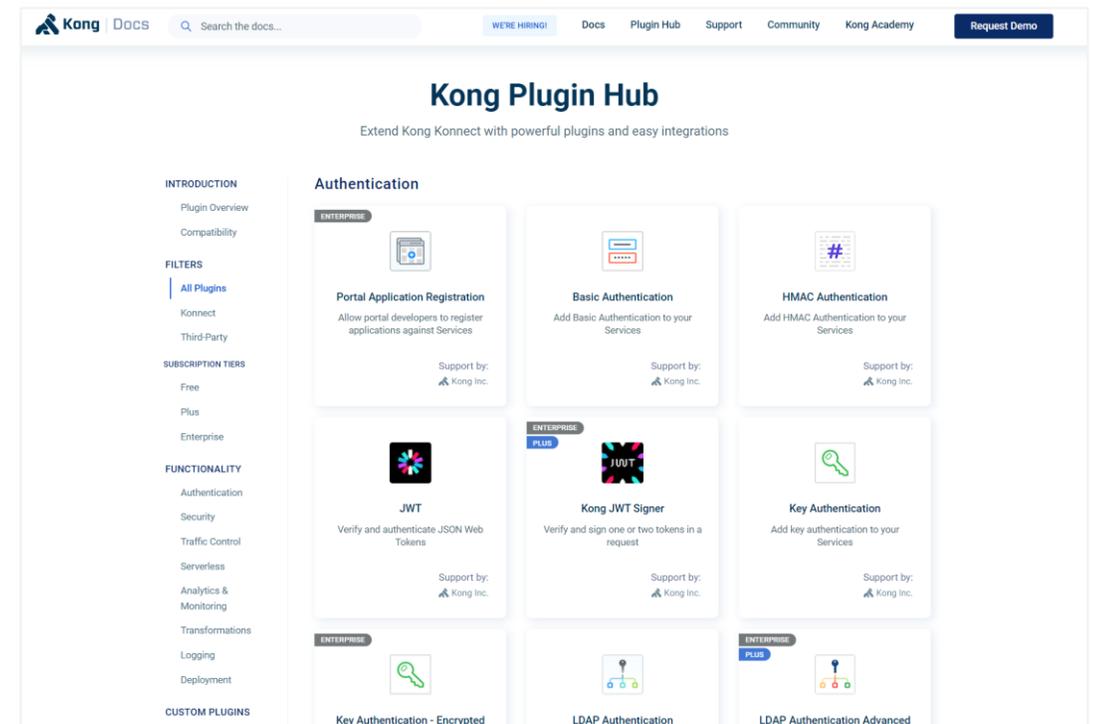


Figure 1: Diagram of Kong Gateway modules and how they relate to the foundational Gateway components. Requests flow from an API client into the Gateway, are modified and managed by the proxy based on your Gateway configuration, and forwarded to upstream services.

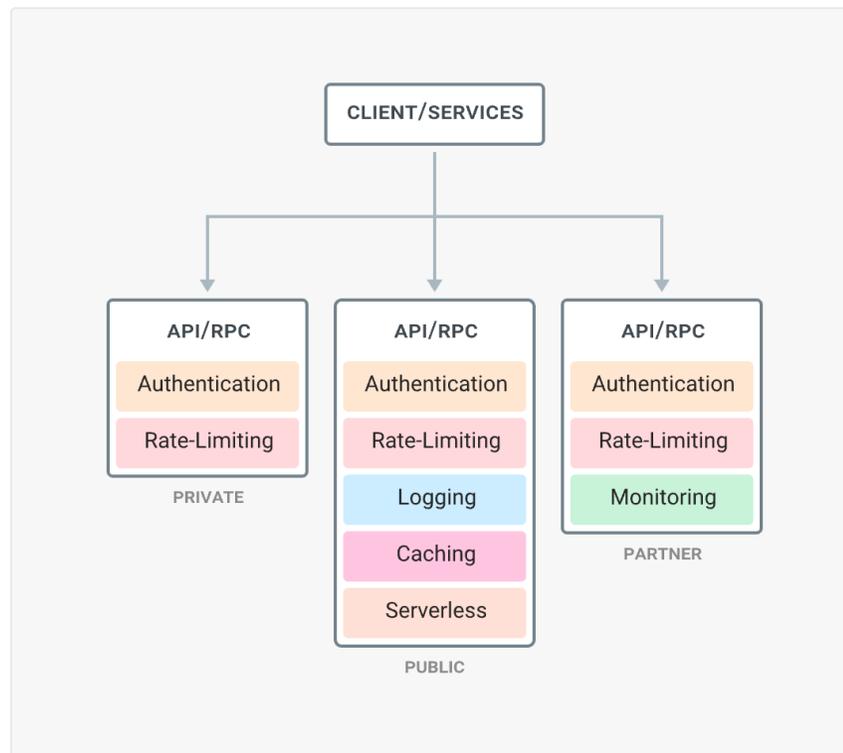
Kongドキュメントより引用
<https://docs.konghq.com/gateway/latest/>



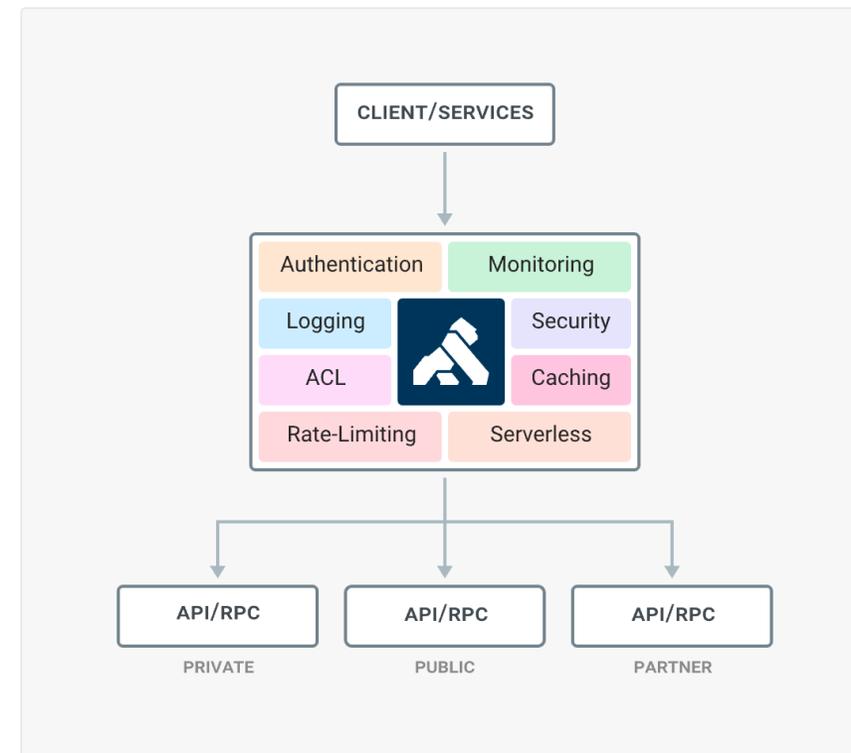
Kongプラグイン一覧ページ
<https://docs.konghq.com/hub/>

- 複数のAPIが異なる事業者によって開発・提供される場合でも、その接続要件やレート制限のルール、API呼び出しのログ取得など共通的に必要となる機能をまとめて提供します。提供するAPIの均質化と、API開発コストの低減に貢献します。

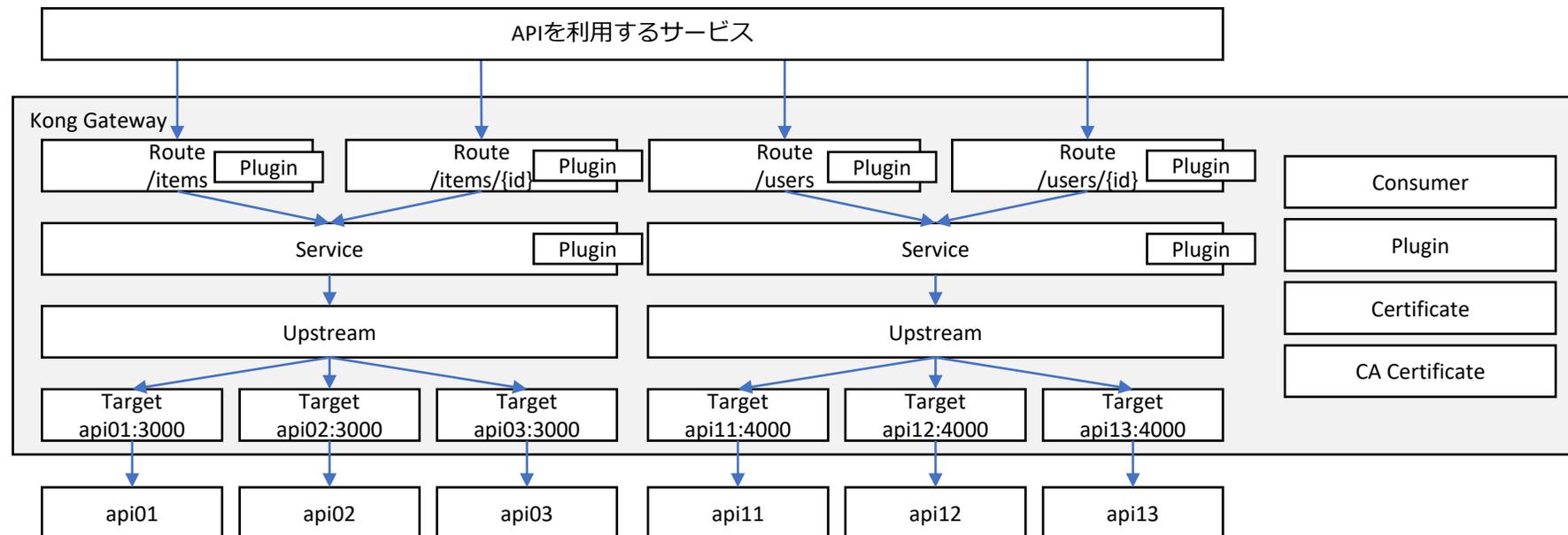
The Redundant Old Way



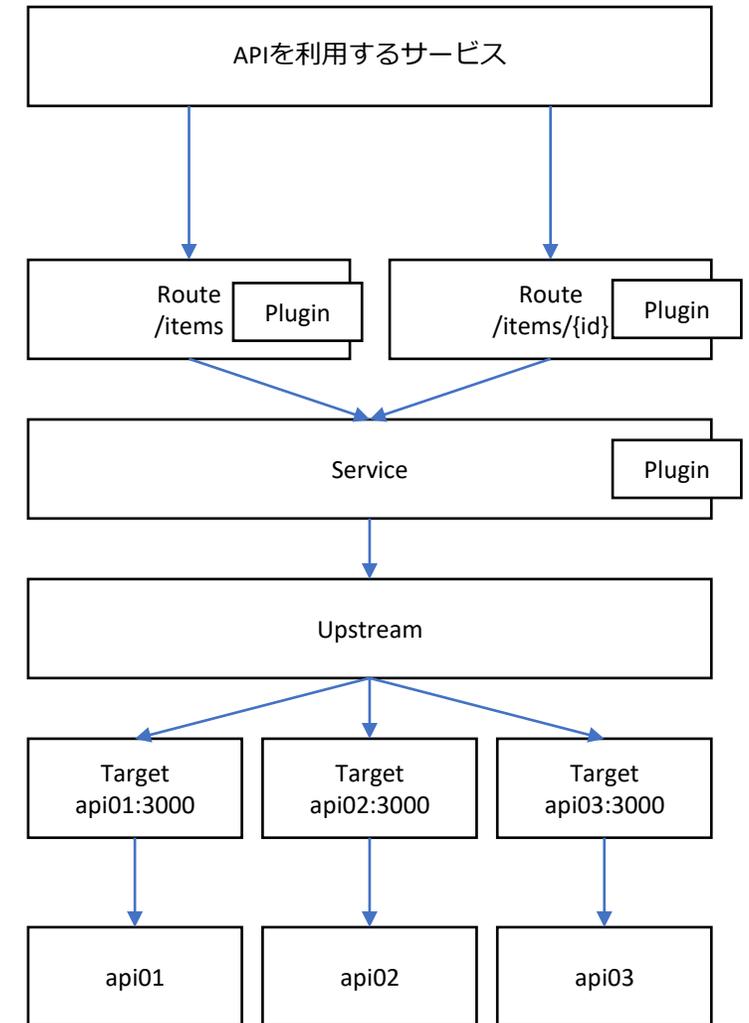
The Kong Way



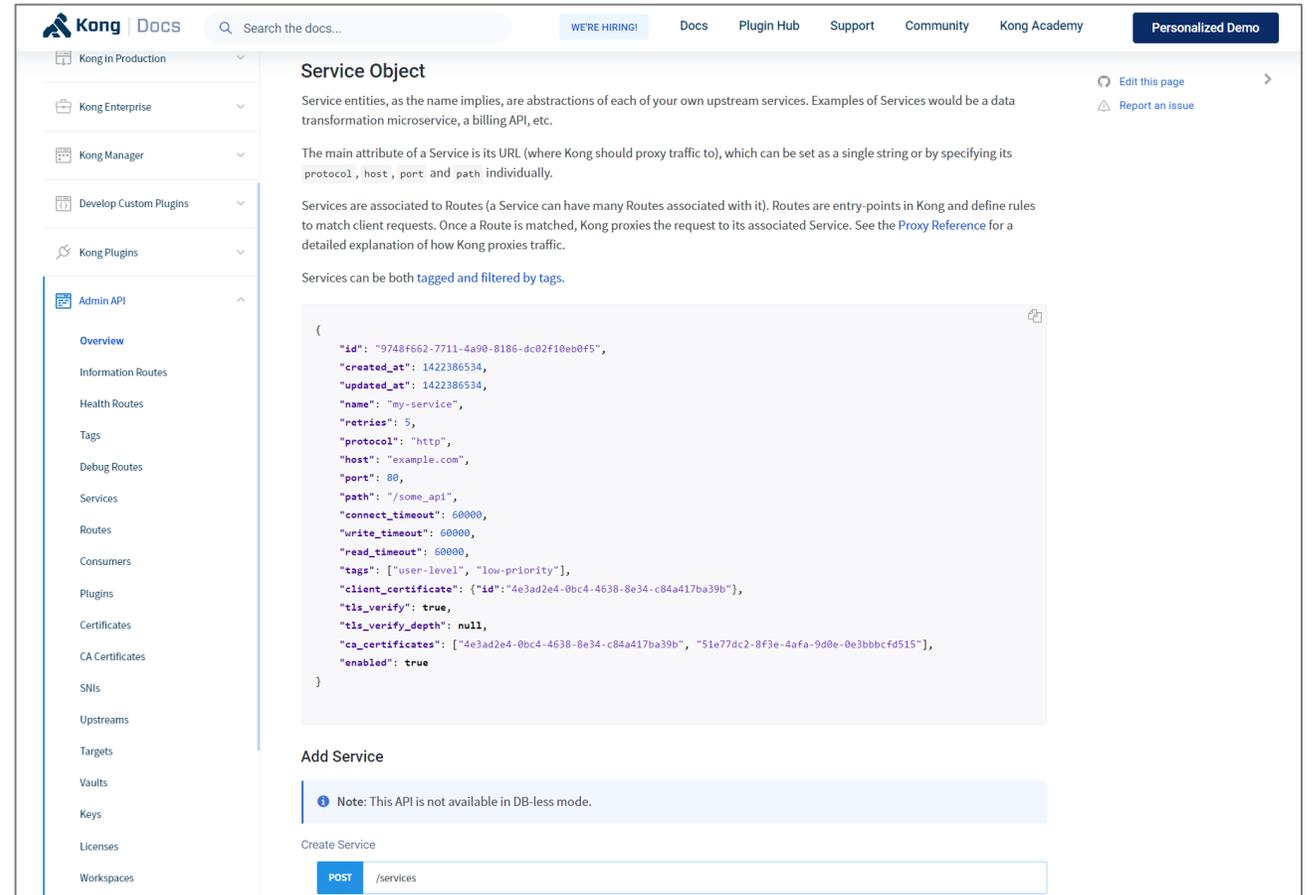
- Kongは、NginxをベースにLuaで拡張されたオープンソースのAPI Gatewayです。API Gatewayのコア機能に特化し、設定は管理用のREST APIで行います。
- 外部から受け付けたHTTPリクエストは、Route、Service、Upstream、Targetとよばれる内部リソースを辿ってAPIサーバヘルレーティングされます。RouteやServiceを通過する際にリクエストをフックし、プラグイン処理を加えることができます。



- 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本体にはGUIが付属しておらず、操作はAPIで実施することになります。
- Kong公式サイトには、取り扱い可能なリソースと操作方法が確認できるAPIリファレンスが公開されています。



Service Object

Service entities, as the name implies, are abstractions of each of your own upstream services. Examples of Services would be a data transformation microservice, a billing API, etc.

The main attribute of a Service is its URL (where Kong should proxy traffic to), which can be set as a single string or by specifying its protocol, host, port and path individually.

Services are associated to Routes (a Service can have many Routes associated with it). Routes are entry-points in Kong and define rules to match client requests. Once a Route is matched, Kong proxies the request to its associated Service. See the [Proxy Reference](#) for a detailed explanation of how Kong proxies traffic.

Services can be both [tagged and filtered by tags](#).

```
{
  "id": "9748f662-7711-4a90-8186-dc02f10eb0f5",
  "created_at": 1422386534,
  "updated_at": 1422386534,
  "name": "my-service",
  "retries": 5,
  "protocol": "http",
  "host": "example.com",
  "port": 80,
  "path": "/some_api",
  "connect_timeout": 60000,
  "write_timeout": 60000,
  "read_timeout": 60000,
  "tags": ["user-level", "low-priority"],
  "client_certificate": {"id": "4e3ad2e4-0bc4-4638-8e34-c84a417ba39b"},
  "tls_verify": true,
  "tls_verify_depth": null,
  "ca_certificates": ["4e3ad2e4-0bc4-4638-8e34-c84a417ba39b", "51e77dc2-8f3e-4afa-9d0e-0e3bbcf6515"],
  "enabled": true
}
```

Add Service

Note: This API is not available in DB-less mode.

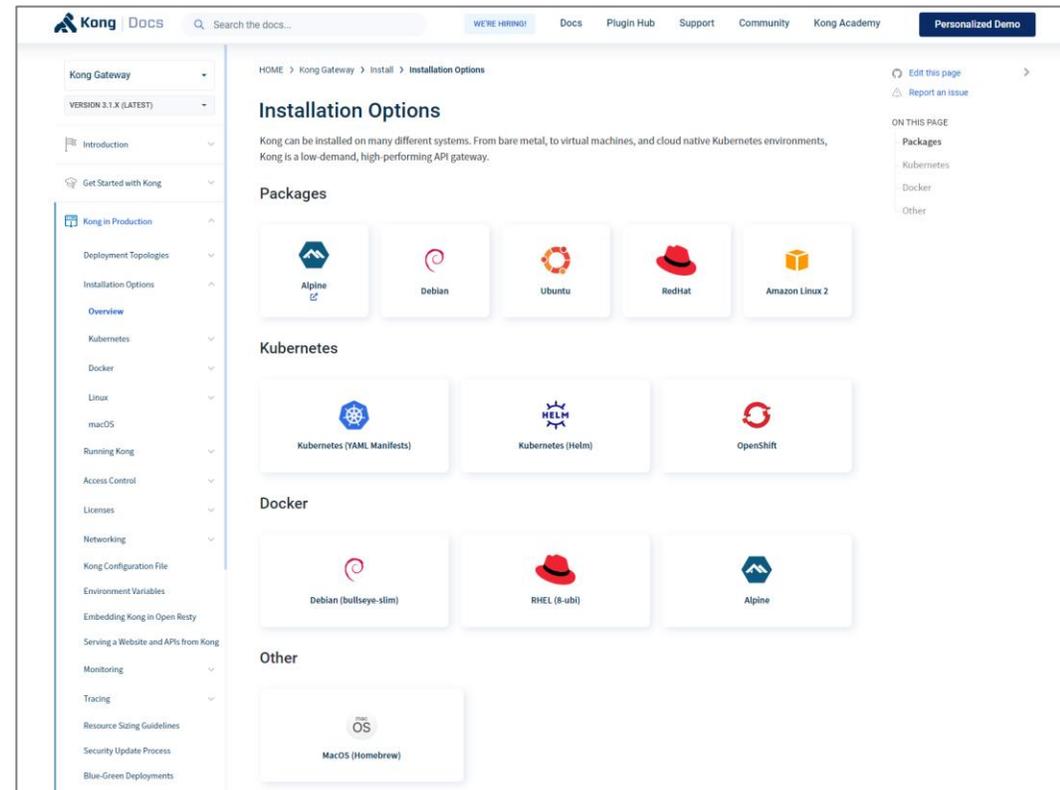
Create Service

POST /services

Kong Admin APIリファレンス
<https://docs.konghq.com/gateway/3.1.x/admin-api/>

- Kong Gatewayは、Linuxサーバに直接パッケージをインストールして利用する方法、Dockerコンテナを使ってコンテナホスト上で稼働させる方法、またはKubernetesのIngress Controllerとして動作させることができます。
- PostgreSQLに依存しているため、パブリッククラウドサービスが提供するマネージドサービスを利用する等して用意が必要です。
 - DB-less mode と呼ばれる構成を採用することで、DBを用意しないでKongを動かすことも可能です（一部機能制約があることに注意）。

<https://docs.konghq.com/gateway/latest/production/deployment-topologies/db-less-and-declarative-config/>



<https://docs.konghq.com/gateway/latest/install/>

FIWARE Orion

(非パーソナルデータ向けのコンテキストブローカー)

FIWARE Orion : ブローカー (非パーソナル) の推奨モジュール

- FIWARE Orionは、データブローカーとして国内/国外のスマートシティ事業において多数の実績があるOSSです。データ蓄積とデータ分散双方の機能を具備し、提供される接続インターフェースとしてNGSIv2というオープンAPIで実装されています。

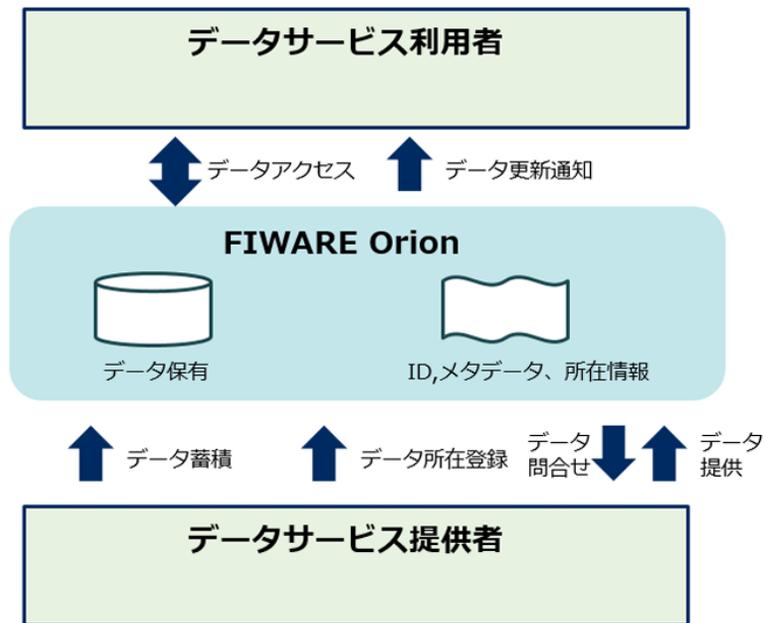
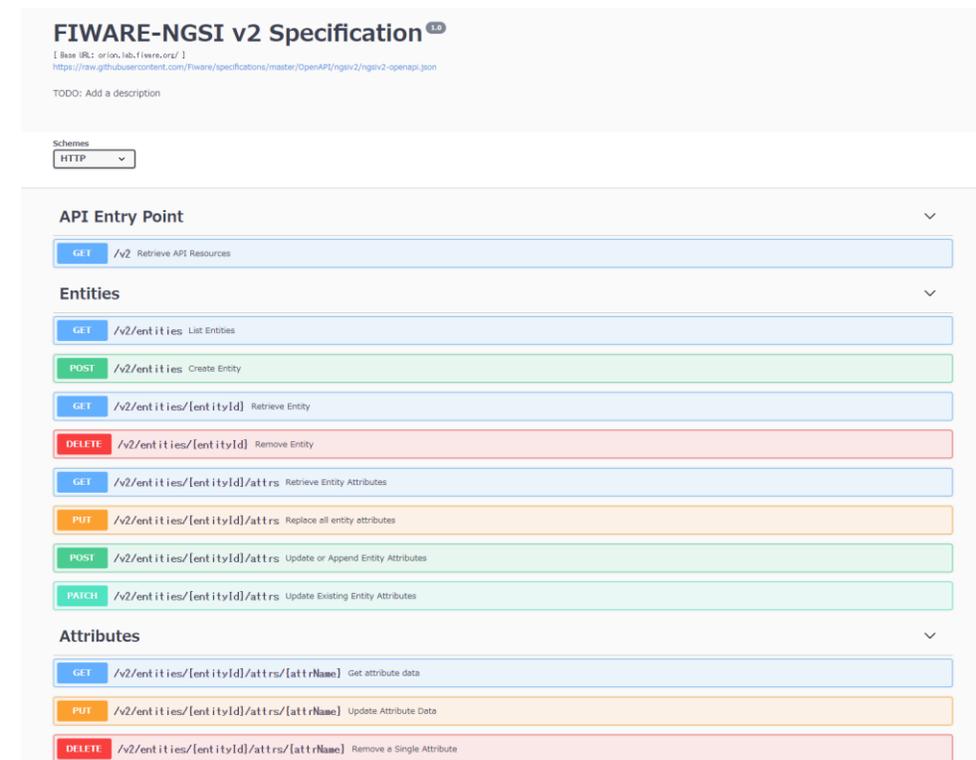


図 4-2-1-2 FIWARE Orion データフロー図

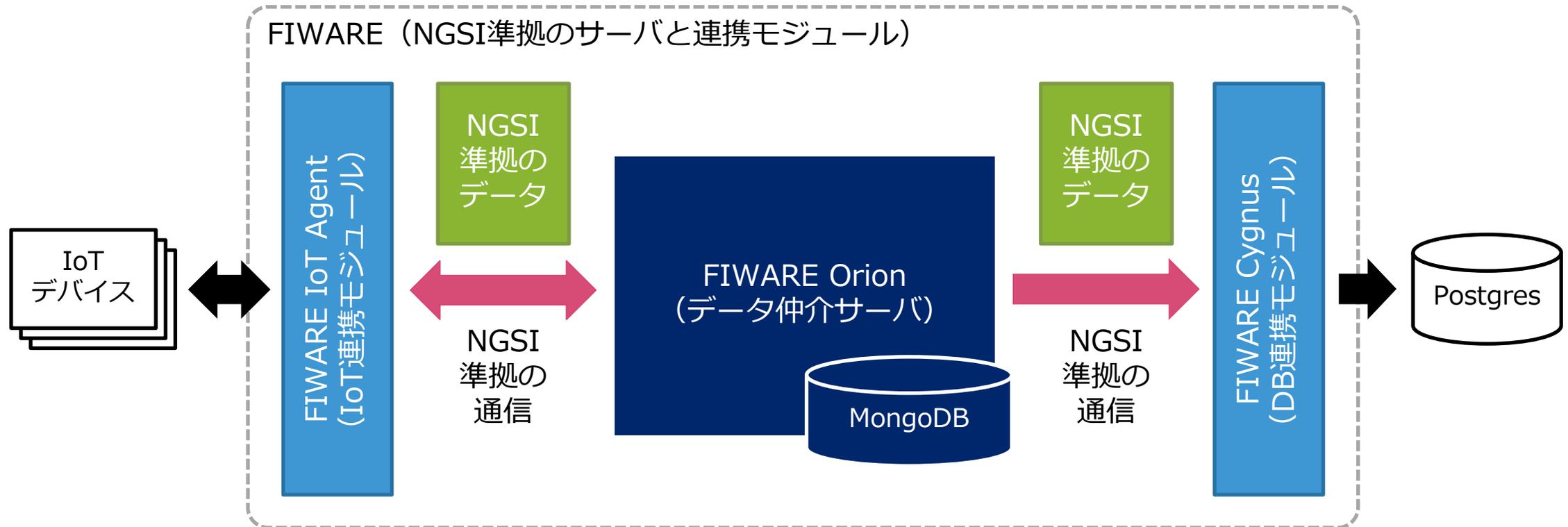
2021年度 生活用データ連携に関する機能等に係る調査研究 調査報告書より引用



FIWARE-NGSI v2 Specification (公開されているAPI仕様ページ)
<https://swagger.lab.fiware.org/>

FIWARE Orion : ブローカー (非パーソナル) の推奨モジュール

- NGSI = データ形式 + 通信方法のルール
- FIWARE = NGSI準拠のデータ仲介サーバ※1 + 連携モジュール群※2



※1 : NGSIv2に準拠したFIWARE Orionがよく利用され実績が多い。

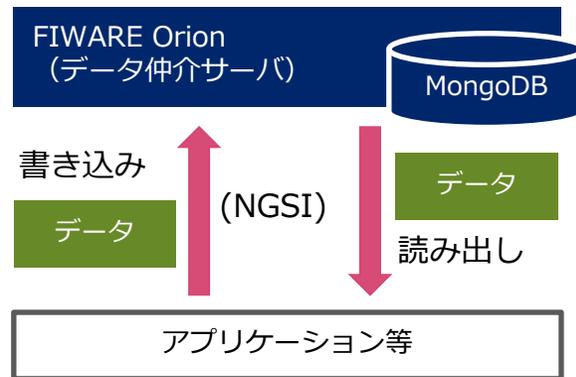
※2 : IoTデバイスやDBとの連携以外にも多数の連携モジュールが存在する。

<https://github.com/FIWARE/catalogue/releases/>

- 大きく分けて (A) データの管理、(B) データ所在情報の管理、(C) データ更新情報の購読 の3つの機能が利用できます。

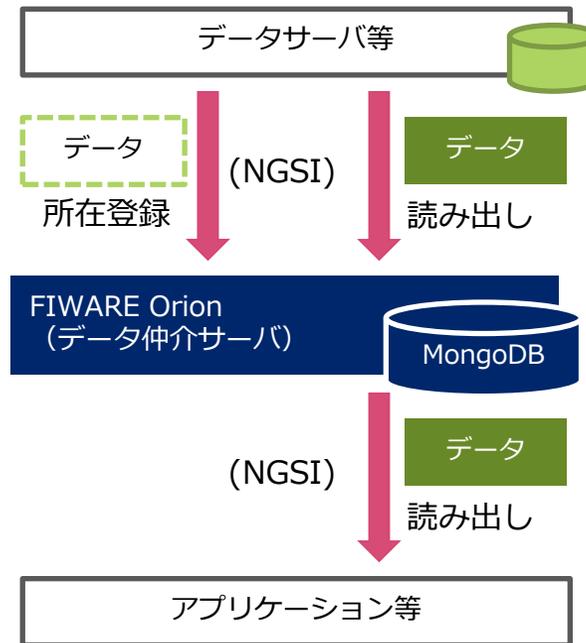
(A) データの管理

データを溜め込むDBとして利用する。



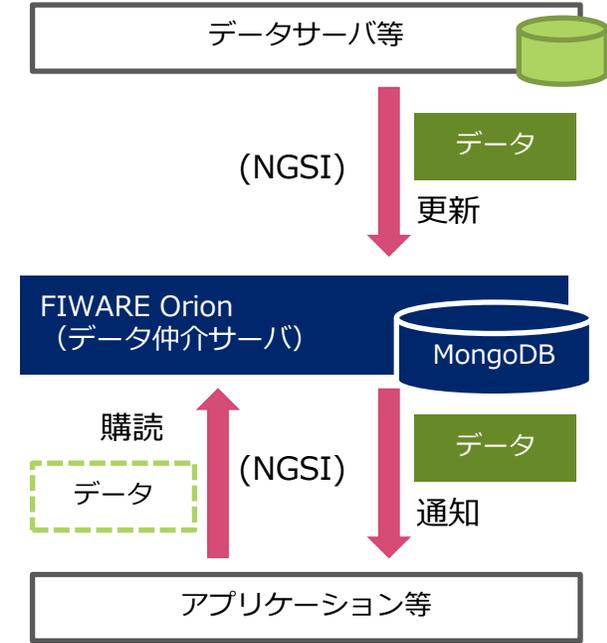
(B) データ所在情報の管理

多数のデータソースへのアクセスを1つのエンドポイントにまとめる。

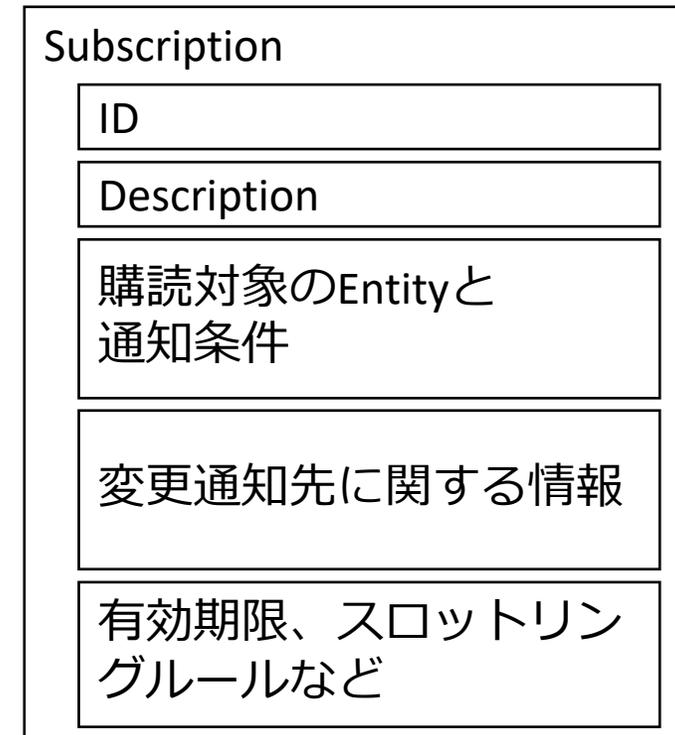
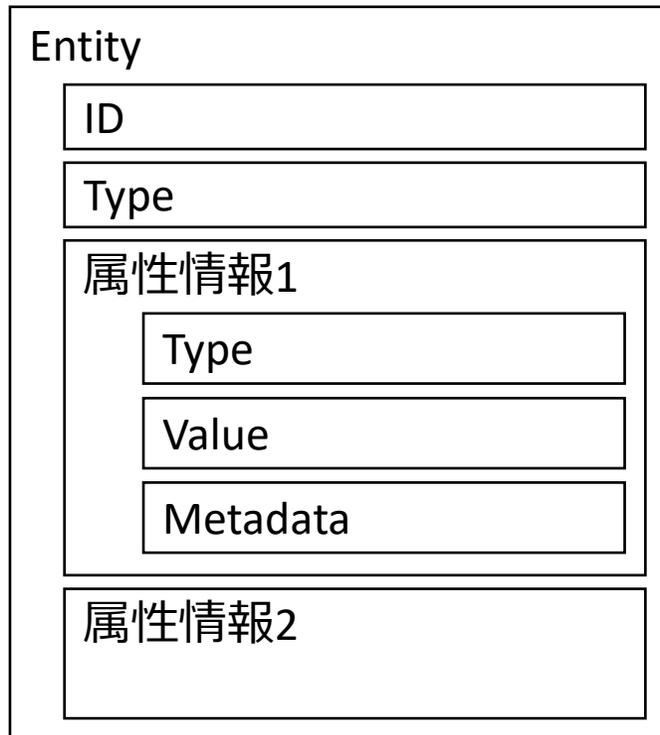


(C) データ更新情報の購読

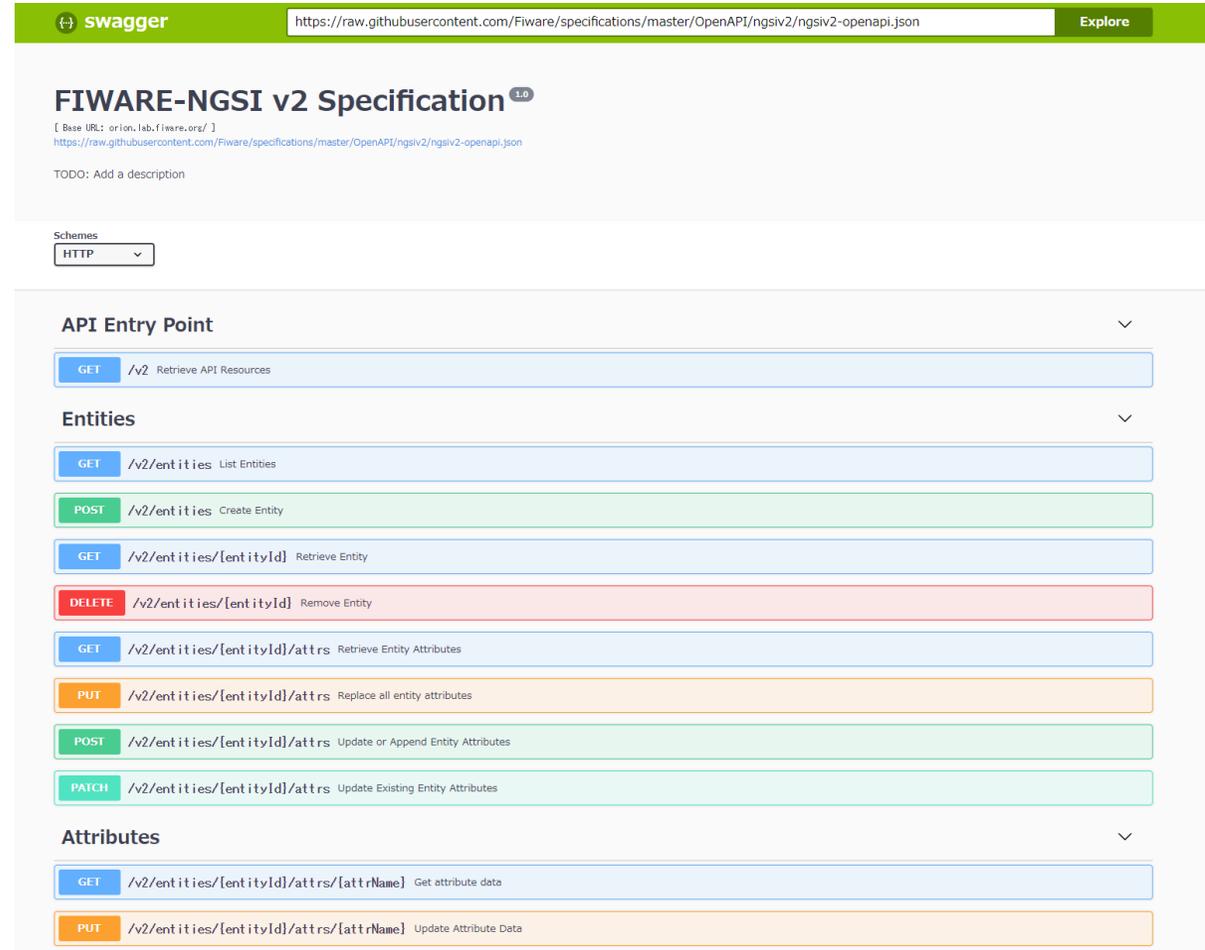
データが更新されたら通知してもらう。



- Entity : コンテキスト情報1つに相当
- Registration : コンテキスト情報の所在場所のルール1つに相当
- Subscription : コンテキスト情報の変更購読ルール1つに相当



- NGSIv2に対応しています。
利用可能なAPIは右のサイトで公開されています。
- 外部データを所在登録する場合、そのデータ取得もNGSIv2（POST /op/query）に対応している必要があります。
- 情報更新を購読する場合、その通知の受信もNGSIv2（POST /op/notify）に対応している必要があります。

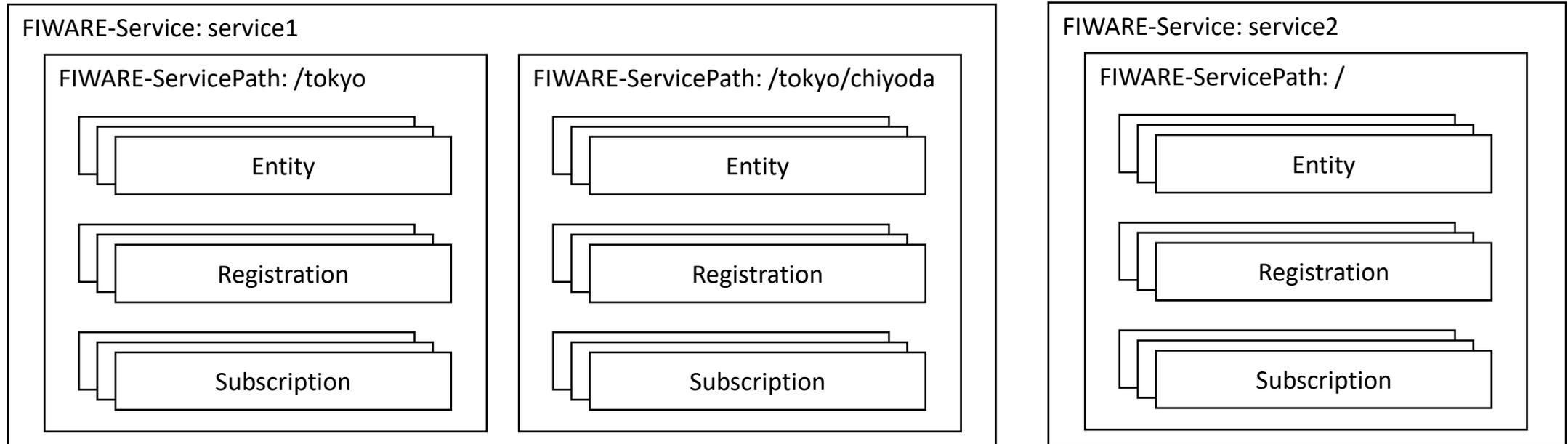


The screenshot displays the Swagger UI for the FIWARE-NGSI v2 Specification (1.0). The base URL is `https://raw.githubusercontent.com/Fiware/specifications/master/OpenAPI/ngsiv2/ngsiv2-openapi.json`. The API is categorized into three main sections:

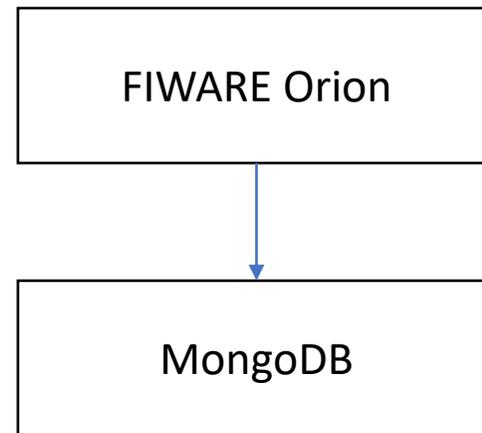
- API Entry Point:**
 - GET `/v2` Retrieve API Resources
- Entities:**
 - GET `/v2/entities` List Entities
 - POST `/v2/entities` Create Entity
 - GET `/v2/entities/{entityId}` Retrieve Entity
 - DELETE `/v2/entities/{entityId}` Remove Entity
 - GET `/v2/entities/{entityId}/attrs` Retrieve Entity Attributes
 - PUT `/v2/entities/{entityId}/attrs` Replace all entity attributes
 - POST `/v2/entities/{entityId}/attrs` Update or Append Entity Attributes
 - PATCH `/v2/entities/{entityId}/attrs` Update Existing Entity Attributes
- Attributes:**
 - GET `/v2/entities/{entityId}/attrs/{attrName}` Get attribute data
 - PUT `/v2/entities/{entityId}/attrs/{attrName}` Update Attribute Data

<https://swagger.lab.fiware.org/>

- FIWARE Orionでは、Entity、Registration、Subscriptionを保存するネームスペースとしてFIWARE ServiceとFIWARE ServicePathが用意されています。それぞれHTTPリクエストのヘッダに記載して設定します。
- FIWARE Orionに認証や認可に関する機能はありませんが、API Gateway側でHTTPリクエストヘッダごとに利用可能なユーザを制御する管理は可能です。

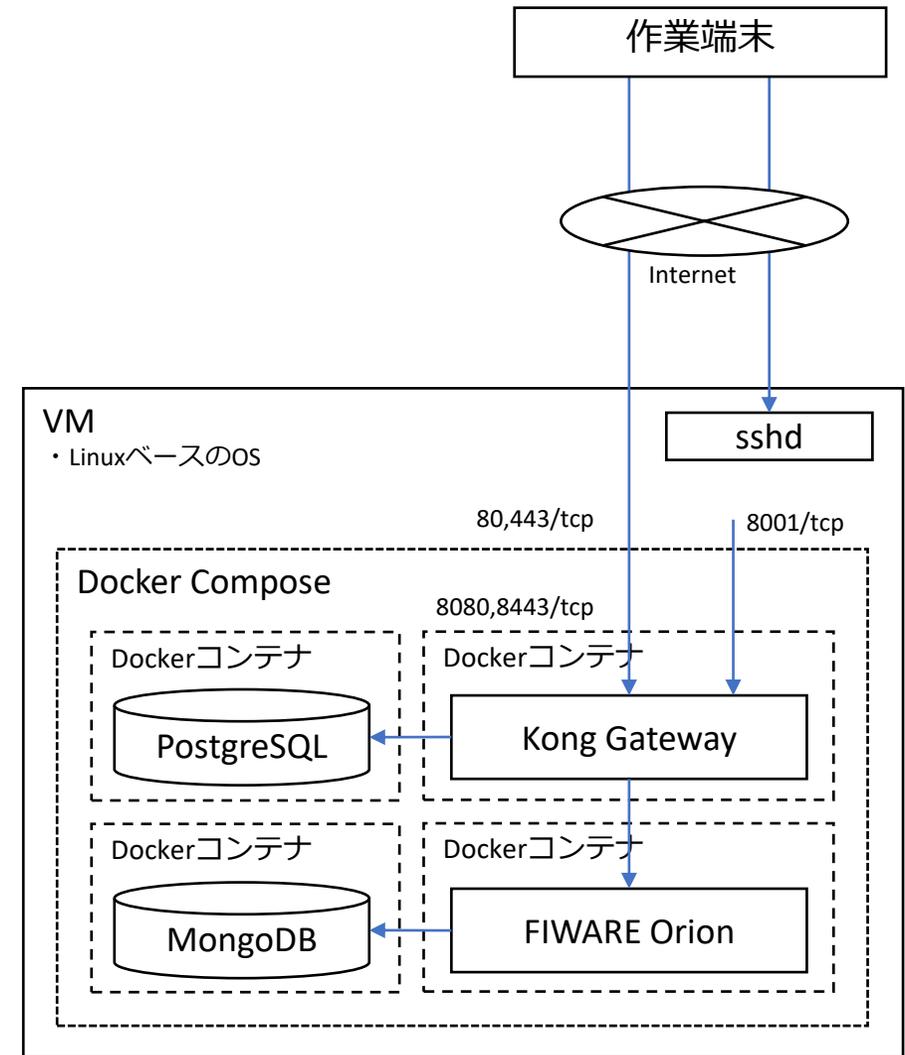


- Dockerhubにコンテナイメージが用意されています。更新が不定期なので古いイメージを利用する場合は脆弱性に注意してください。
- MongoDBに依存しています。MongoDB互換のマネージドサービス（Azure Cosmos DBやAmazon DocumentDB）は、FIWARE Orionのバージョンによってサポートされていない可能性があるため検証が必要です。
- 認証やセキュリティに関する機能は備えていないため、別の仕組みで用意が必要です。



Docker Composeによる 構築 / 設定デモ

- Docker ComposeでKong GatewayおよびFIWARE Orionのコンテナを立て、APIによる設定を行います。
 - Docker環境はインストールされているものとします。
 - DBは本来マネージドサービスを払い出したものに接続しますが、今回は簡単のためコンテナで動作させるものとします。
- 次の操作を行います。
 - Docker Composeファイルの作成
 - コンテナイメージの取得
 - DBコンテナの起動と初期化
 - Kong / FIWARE Orionコンテナの起動
 - KongとFIWARE Orionの連携設定
 - Kongプラグインによる認証設定



参考 : デモで利用した Docker Compose ファイル

FIWARE Orion と MongoDB

```
version: '3.8'
services:

  mongo:
    image: mongo:4.4
    restart: always
    command: --bind_ip_all --quiet
    environment:
      TZ: Asia/Tokyo
    volumes:
      - ./data/mongodb:/data

  orion:
    image: fiware/orion:3.7.0
    restart: always
    command: -dbhost mongo -logLevel INFO
    environment:
      TZ: Asia/Tokyo
    depends_on:
      - mongo
    ports:
      - 1026:1026
```

これらはデモ目的で用意したものですので、セキュリティ設定や冗長化など本番環境で必要な構成についてはそれぞれ別途設定が必要です。参考程度に利用ください。

Kong Gateway と PostgreSQL

```
version: '3.8'
services:

  kong:
    image: kong:2.8.3
    restart: always
    environment:
      TZ: Asia/Tokyo
      KONG_DATABASE: postgres
      KONG_PG_HOST: db
      KONG_PG_USER: kong
      KONG_PG_PASSWORD: password
      KONG_PG_DATABASE: kong
      KONG_PROXY_ACCESS_LOG: /dev/stdout
      KONG_ADMIN_ACCESS_LOG: /dev/stdout
      KONG_PROXY_ERROR_LOG: /dev/stderr
      KONG_ADMIN_ERROR_LOG: /dev/stderr
      KONG_ADMIN_LISTEN: 0.0.0.0:8001, 0.0.0.0:8444 ssl
      KONG_PROXY_LISTEN: 0.0.0.0:80, 0.0.0.0:443 ssl
    ports:
      - 80:80
      - 443:443
      - 8001:8001
      - 8444:8444
    depends_on:
      - db

  db:
    restart: always
    image: postgres:13
    environment:
      TZ: Asia/Tokyo
      POSTGRES_PASSWORD: password
      POSTGRES_USER: kong
      POSTGRES_DB: kong
    volumes:
      - ./data/postgres:/var/lib/postgresql/data
```

- コンテナイメージの取得
 - docker compose pull
- Kong Gatewayコンテナの起動
 - docker compose up -d db
 - docker compose run kong kong migrations bootstrap
 - docker compose up -d
- FIWARE Orionコンテナの起動
 - docker compose up -d

- Kongのサービス登録
 - `curl -X POST -d "name=fiware-orion" -d "url=http://172.17.0.1:1026/" localhost:8001/services`
 - `curl -X POST -d "name=all" -d "paths[]=/" localhost:8001/services/fiware-orion/routes`
- Consumer登録とAPIキーの作成
 - `curl -X POST -d "username=app01" localhost:8001/consumers`
 - `curl -X POST localhost:8001/consumers/app01/key-auth`
- APIキー認証の設定
 - `curl -X POST -d "name=key-auth" localhost:8001/routes/all/plugins`

- Entityの登録
 - `curl -X POST -H "Content-Type: application/json" -H "apikey: xxx" -d '{"id": "Room001", "type": "Room", "temperature": {"type": "float", "value": 25.5, "metadata": {}}}' localhost/v2/entities`
- Entityの取得
 - `curl -H "apikey: xxx" localhost/v2/entities/Room001`
- Entityの更新
 - `curl -X PUT -H "Content-Type: text/plain" -H "apikey: xxx" -d "23.5" localhost/v2/entities/Room001/attrs/temperature/value`
- Entityの削除
 - `curl -X DELETE -H "apikey: xxx" localhost/v2/entities/Room001`

運用上の注意点

- Kubernetesクラスタの定期アップグレード
- SSL証明書の更新作業
- コンテナイメージの更新
- FIWARE Orionにおけるバイナリの取り扱い方法
- FIWARE Orionで取り扱うEntityの重複に注意
- NGSIv2実装ノートを確認する

Kubernetesクラスタの定期アップグレード

- Kubernetesクラスタを利用している場合は、クラウドサービス側でサポートされているバージョンのサポート期限が設定されているケースがほとんどです。定期的なクラスタアップグレードが必要なケースは計画的な実施が必要です。

AKS Kubernetes リリース予定表

過去のリリース履歴については、[Kubernetes](#) に関するページを参照してください。

ⓘ 注意
アスタリスク (*) は、日付が確定されていないことを示します。このため、以下のタイムラインは変更される可能性があります。引き続きリリース カレンダーで更新プログラムを確認してください。

K8s バージョン	アップストリームのリリース	AKS プレビュー	AKS GA	サポート終了
1.21	2021年4月8日	2021年5月	2021年7月	1.24 GA
1.22	2021年8月4日	2021年9月	2021年12月	1.25 GA
1.23	2021年12月	2022年1月	2022年4月	1.26 GA
1.24	2022年4月22日	2022年5月	2022年7月	1.27 GA
1.25	2022年8月	2022年10月	2022年12月	1.28 GA
1.26	2022年12月	2023年1月	2023年2月	1.29 GA
1.27	2023年4月	2023年5月	2023年6月	1.30 GA
1.28	*	*	*	1.31 GA

- ▲ Microsoft Azureで提供される Kubernetesサービスのリリース予定
<https://learn.microsoft.com/ja-jp/azure/aks/supported-kubernetes-versions?tabs=azure-cli>

Amazon EKS Kubernetes リリースカレンダー

ⓘ 注記
月と年のみの日付はおおよその日付であり、確定後に正確な日付で更新されます。

Kubernetes バージョン	アップストリームのリリース	Amazon EKS リリース	Amazon EKS のサポート終了
1.24	2022年5月3日	2022年11月	2024年1月
1.23	2021年12月7日	2022年8月11日	2023年10月
1.22	2021年8月4日	2022年4月4日	2023年5月
1.21	2021年4月8日	2021年7月19日	2023年2月
1.20	2020年12月8日	2021年5月18日	2022年11月1日
1.19	2020年8月26日	2021年2月16日	2022年8月1日

- ▲ Amazon Web Servicesで提供される Kubernetesサービスのリリース予定
https://docs.aws.amazon.com/ja_jp/eks/latest/userguide/kubernetes-versions.html#kubernetes-release-calendar

リリース チャンネルのスケジュール

★ 注: リリースチャンネルに登録されているクラスタは、次のスケジュールのアップグレード列で指定された日付以降に自動的にアップグレードされます。バージョンは有効期限が終了するまでリリースチャンネルで使用できます。チャンネルのデフォルトバージョンよりも古いバージョンを選択する場合は、メンテナンスの除外を使用して、クラスタがサポート終了まで自動的にアップグレードされないようにできます。

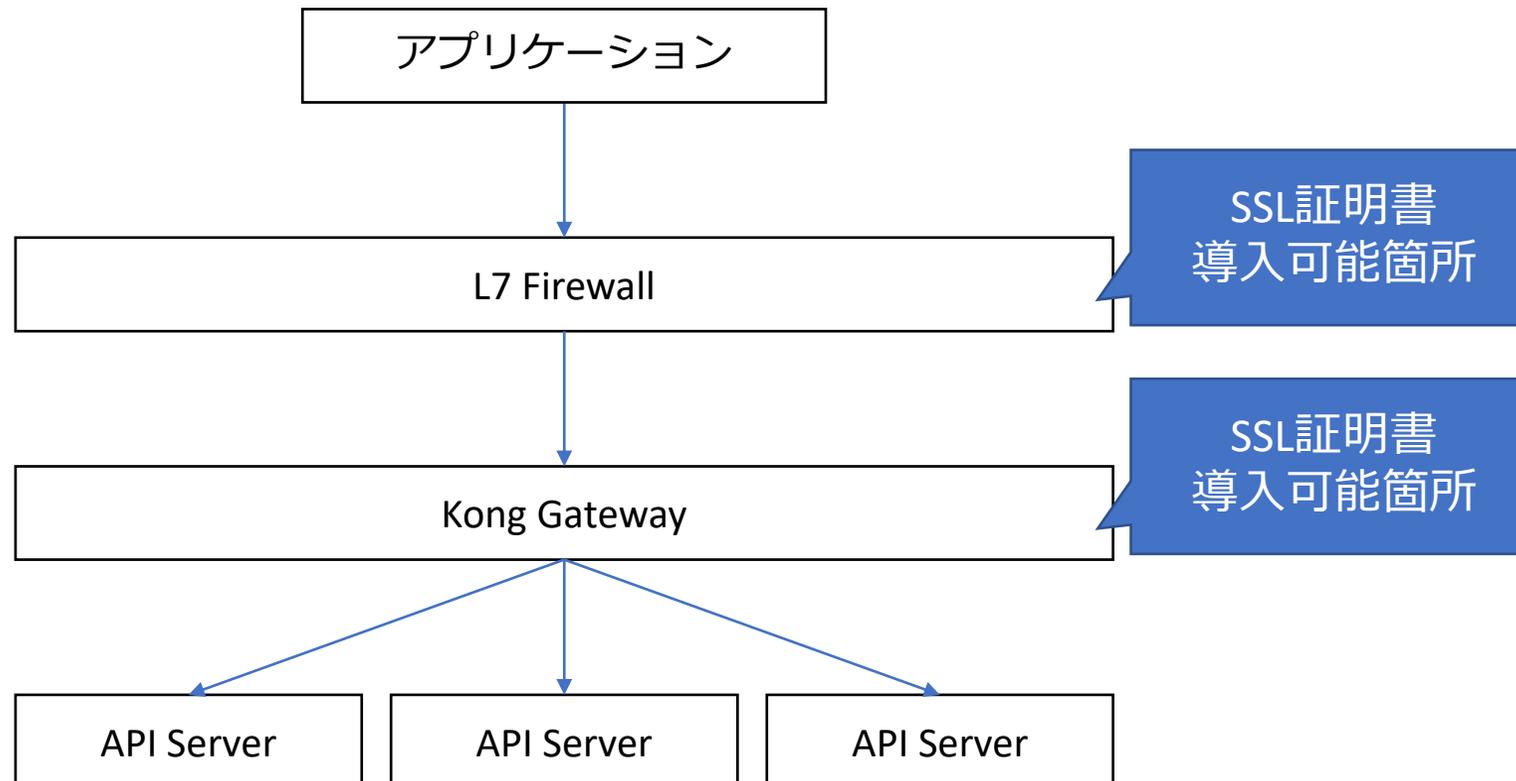
リリースチャンネルに登録されているクラスタは、以下のアップグレードスケジュールに従います。

Kubernetes のバージョン	Kubernetes のリリース日	迅速性と		Regular		Stable		サポート終了 ³
		使用可能 ¹	アップグレード ²	使用可能 ¹	アップグレード ²	使用可能 ¹	アップグレード ²	
1.21	2021-04-08	2021-06-06	2021-09-02	2021-10-01	2022-01-07	2021-12-18	2022-04-29	2023-01-31
1.22 ⁴	2021-08-04	2021-10-12	2022-04-05	2022-01-26	2022-05-27	2022-05-06	2022-09-23	2023-04-30
1.23	2021-12-14	2022-01-27	2022-05-27	2022-04-29	2022-10-20	2022-07-22	2023-01-09	2023-07-31
1.24	2022-05-03	2022-06-03	2022-08-12	2022-08-19	2023-01-09	2022-11-30	2023-03	2023-10-31
1.25 ⁵	2022-08-23	2022-09-13	2023-01-09	2023-01-09	2023-03	2023-03	2023-Q2	2024-02-29
1.26	2022-12-06	2023-01-16	2023-03	2023-03	2023-Q2	2023-Q2	2023-Q2	2024-05-31

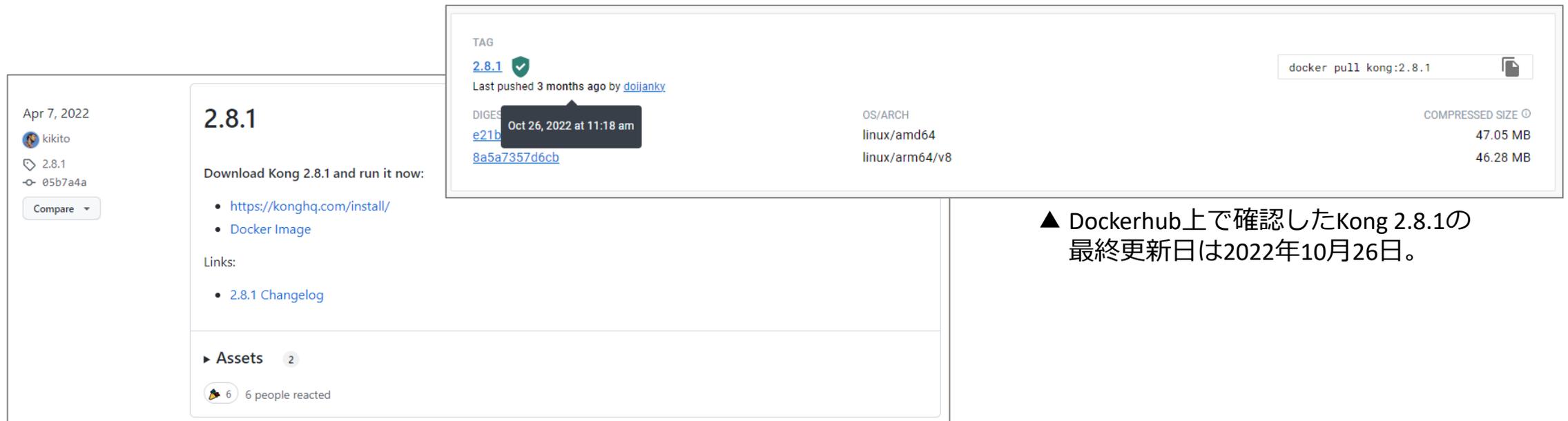
ⓘ 注意: GKE バージョン 1.20 以前はサポート終了となり、サポートされなくなりました。

- ▲ Google Cloud Platformで提供される Kubernetesサービスのリリース予定
<https://cloud.google.com/kubernetes-engine/docs/release-schedule?hl=ja>

- 基本的にすべての操作はAPIを介して実施するため、APIエンドポイントをSSL終端する場合は証明書の期限が切れるとすべてのアクセスが失敗する可能性があります。テストの上で計画的に証明書の更新が必要です。



- Dockerhubで公開されているコンテナイメージにはバージョンタグが設定されていますが、同じバージョンタグでもイメージ自体が更新されているケースがあります。アプリケーションのバージョンは同じで、ライブラリが更新されているケース等があります。
 - たとえば、Kong 2.8.1は2022年4月7日リリースですが、コンテナイメージの最終更新日は2022年10月26日となっており、リリース当初からイメージが更新されています。

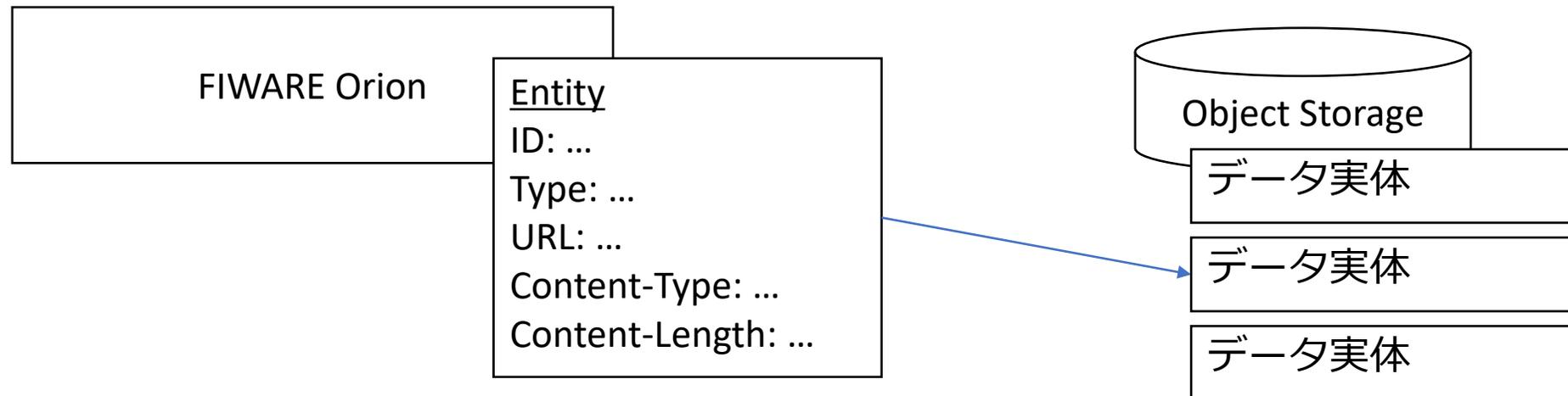


The screenshot shows two side-by-side views of the Kong 2.8.1 release. On the left is the GitHub repository page, which shows the release date as 'Apr 7, 2022'. On the right is the Docker Hub page for the same version, which shows a 'Last pushed' date of 'Oct 26, 2022 at 11:18 am'. A callout box highlights this date on the Docker Hub page. The Docker Hub page also displays technical details for the image, including OS/ARCH (linux/amd64 and linux/arm64/v8) and compressed sizes (47.05 MB and 46.28 MB).

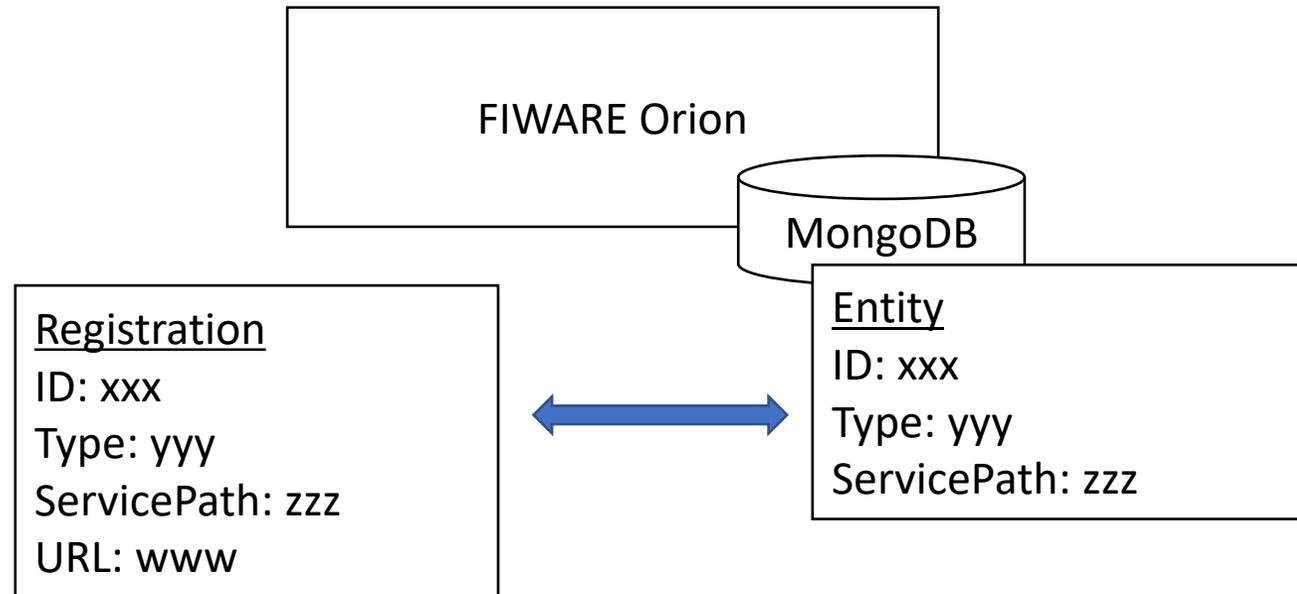
▲ GitHub上で確認できるKong 2.8.1のリリース日は2022年4月7日。

▲ Dockerhub上で確認したKong 2.8.1の最終更新日は2022年10月26日。

- FIWARE OrionではJSONデータのみ取り扱い可能です。そのため、画像、動画、PDF、Excelファイル等のデータはURLやサイズ等のメタデータのみを登録し、データ本体は別の場所（オブジェクトストレージ等）に格納する必要があります。



- EntityはFIWARE Service/ServicePath/ID/Typeで識別しますので、これらが重複しないよう設計する必要があります。また、FIWARE Orionに蓄積されているEntityと、Registrationで所在登録のみされているEntityで上記が重複しないよう注意する必要があります。



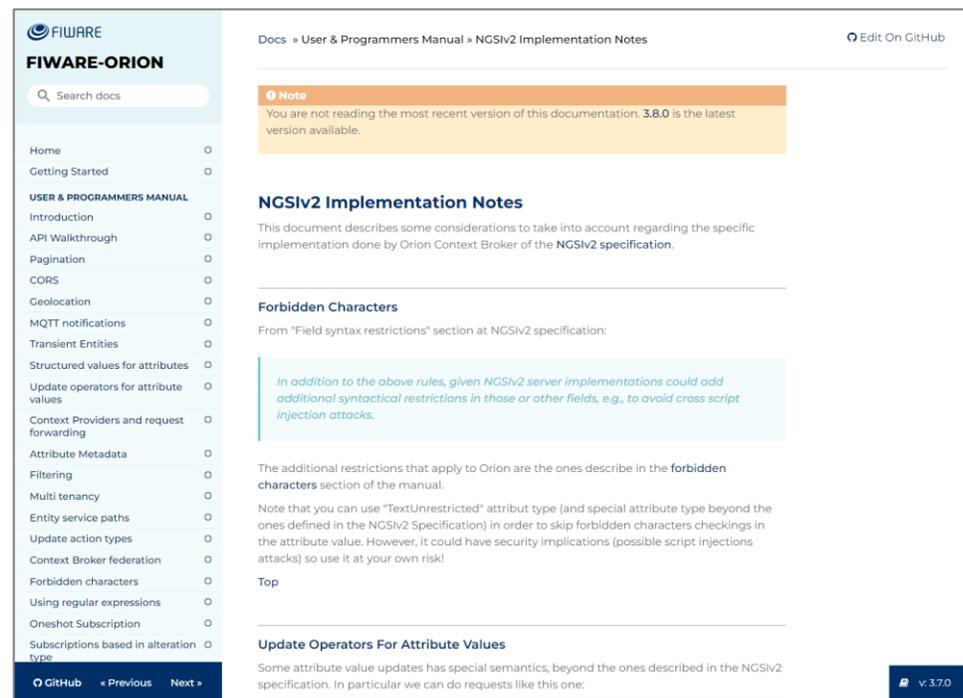
- NGSiv2実装ノートには、FIWARE Orionで利用できない文字、日付の形式、リクエストペイロードの検証項目など、FIWARE Orionを利用する上での留意事項がまとまっています。一読した上で動作確認することをおすすめします。

https://fiware-orion.letsfiware.jp/3.7.0/user/ngsiv2_implementation_notes/

https://fiware-orion.readthedocs.io/en/3.7.0/user/ngsiv2_implementation_notes.html



The screenshot shows the FIWARE Orion documentation website. The main content area is titled "NGSiv2 実装ノート (NGSiv2 Implementation Notes)". It lists various restrictions and considerations for NGSiv2 implementation, such as forbidden characters, date formats, and request payload validation. The left sidebar contains a navigation menu with categories like "ホーム", "入門", "ユーザープログラマ・マニュアル", and "NGSiv2 実装ノート".



The screenshot shows the FIWARE Orion documentation website. The main content area is titled "NGSiv2 Implementation Notes". It includes a note stating: "You are not reading the most recent version of this documentation. 3.8.0 is the latest version available." Below this, the page content is partially visible, showing sections like "Forbidden Characters" and "Update Operators For Attribute Values". The left sidebar contains a navigation menu with categories like "Home", "Getting Started", "USER & PROGRAMMERS MANUAL", and "NGSiv2 Implementation Notes".

まとめ

- 本セミナーでは、非パーソナルデータを対象として、次の項目について説明しました。
 - エリア・データ連携基盤の構築に関連する技術知識と用語
 - Kong Gatewayの概要と内部動作
 - FIWARE Orionの概要と内部動作
 - Docker Composeを使ったテスト環境の構築と、curlコマンドによる設定、動作確認
 - 運用上の注意点

問い合わせ方法等について

- 自治体等からの問い合わせは「エリア・データ連携基盤お問合せフォーム」よりお問合せください。別途担当者よりご連絡させていただきます。

DSAエリア・データ連携基盤お問合せフォーム

<https://data-society-alliance.org/area-data/contact/>

- なお、DSAのHPには推奨モジュールの導入を支援するために、導入ガイドライン（ビルド手順書 / 構築手順書）や利用ガイドライン（利用手順書 / 連携手順書）を公開しており、推奨モジュールのソースコードはOSSとしてGitHubに公開されています。

Kong Gateway <https://github.com/Kong/kong>（外部サイト）

FIWARE Orion <https://github.com/telefonicaid/fiware-orion>（外部サイト）

- また、ハンズオンセミナーへの参加を希望される場合は、DSAへの会員登録をお願いいたします。

質疑応答