



Fusion Development Workshop

Power Apps + Azure

Ayumu Inaba
Cloud Solution Architect
Microsoft Japan

Agenda

はじめに

- Module 0 フュージョン開発アプローチ
- Module 1 Power Apps とカスタムコネクタ
- Module 2 市民開発者のセルフサービス開発
- Module 3 アプリと API の開発プロセス
- Module 4 開発者ポータルのセットアップ

まとめ

- Appendix A オンプレミス資源の利活用
- Appendix B Azure AD 認証によるセキュア API

はじめに

Power Platform を活用した市民開発者によるアプリケーション開発が進むほど、より広範なデータや API を活用する需要が高まってくる

代表的な SaaS サービスであれば多数のコネクタが既に提供されているが、企業が持つ独自のシステムでは再利用可能な形で API が提供されていないことも多い

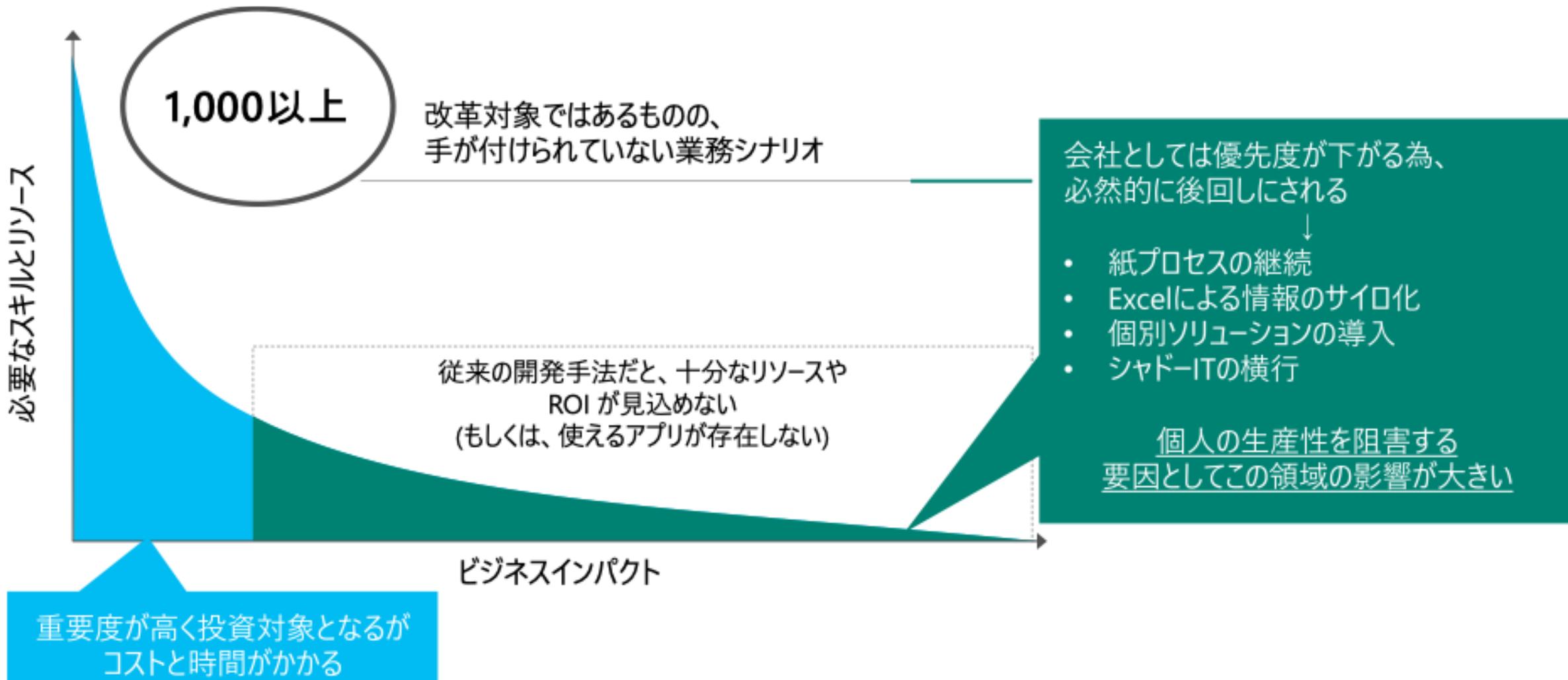
本資料ではこういった「既存システムが管理する重要な業務データを Power Apps から活用する」ために必要な方法などを整理・紹介する

Module 0

フュージョン開発

従来型アプリ開発の課題

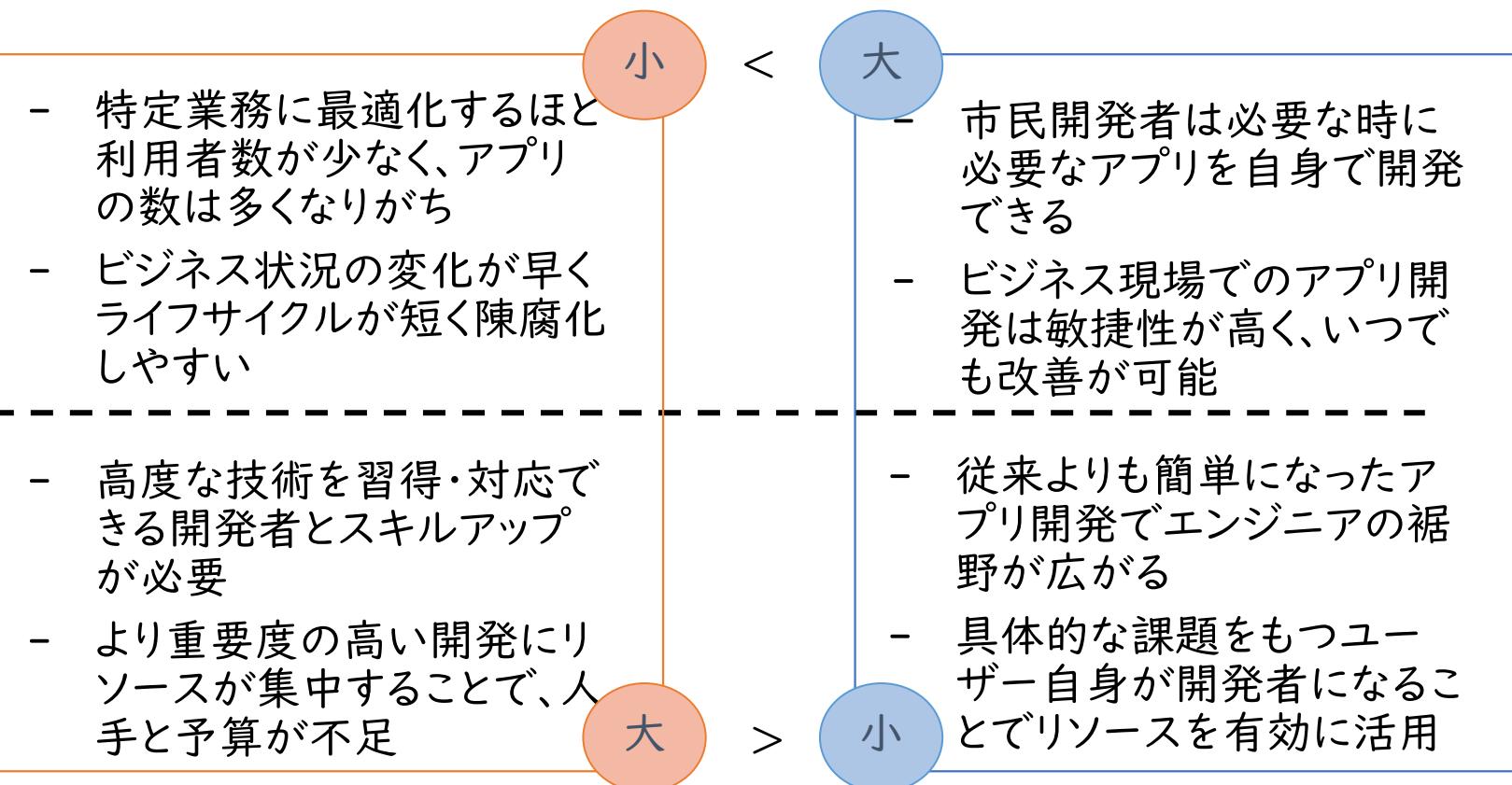
開発リソースの制約から全ての課題に対応できない



ロングテールへの対応

ROI の課題と市民開発やローコードツールへの期待

$$ROI = \frac{\text{効果} \\ (\text{アプリの質と量})}{\text{リソース} \\ (\text{お金と時間})}$$

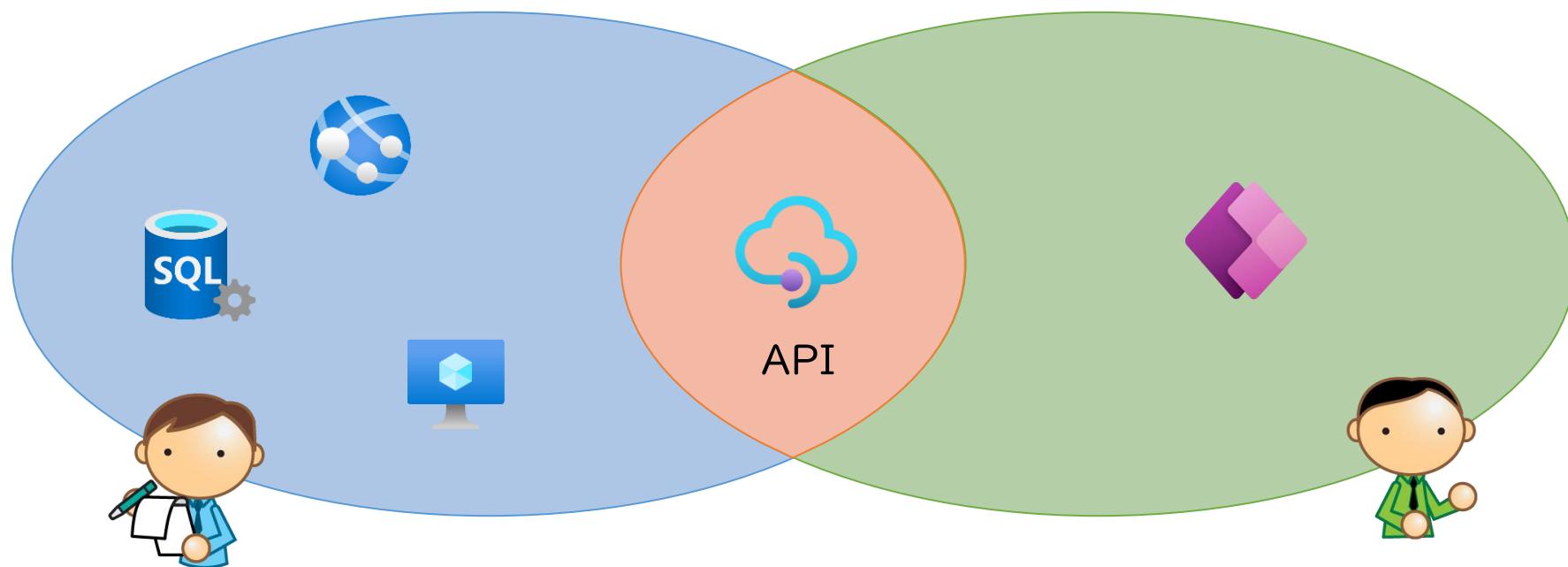


フュージョン開発のアプローチ

とはいえた市民開発者だけで解決できる課題は限定的なため、プロ開発者の必要性は依然として高い

既存システムとの連携、高度なデータ構造、複雑な業務ロジック、etc...

市民開発者とプロ開発者のギャップを埋めるための API が重要になってくる

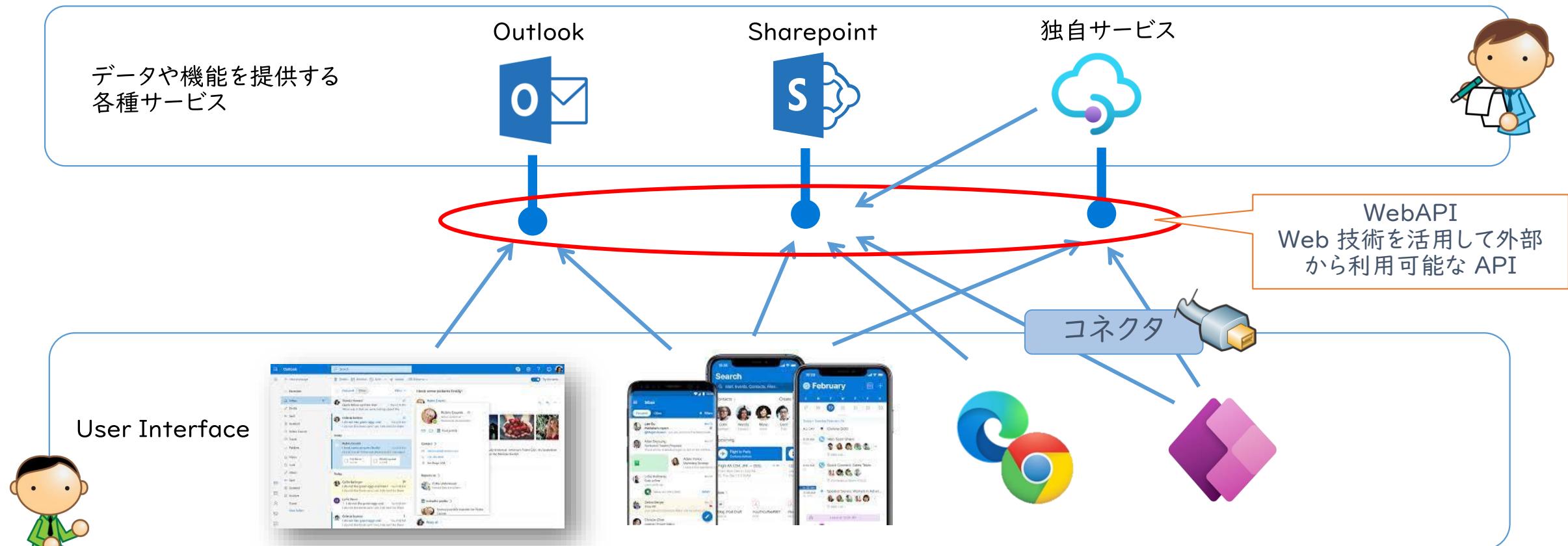


高度な IT の知見を持つプロ開発者 + リアルなビジネスの知見を持つ市民開発者

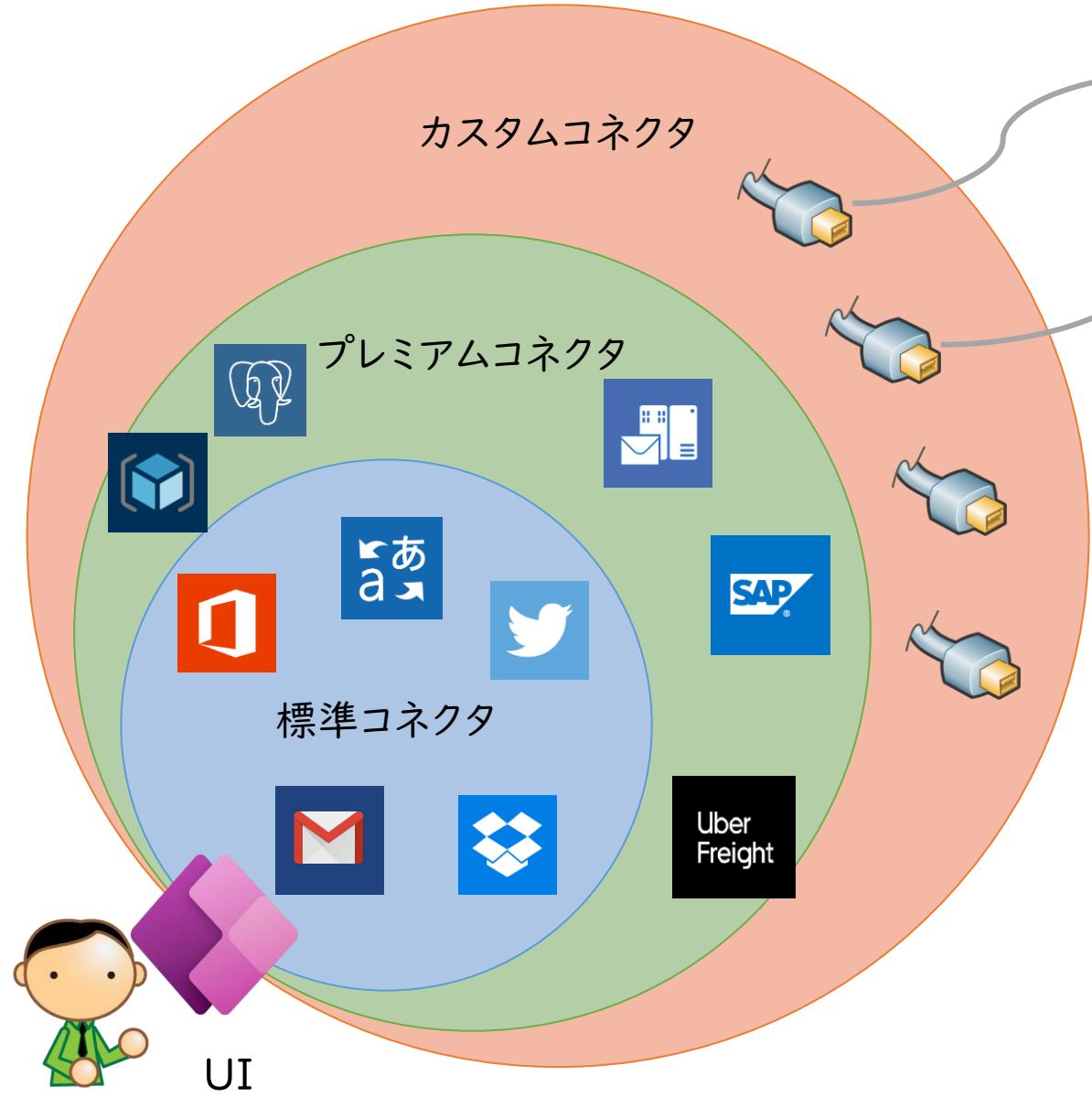
Application Programming Interface ?

各種のアプリケーションと「再利用する部品」あるいは部品間で情報を取り交換する仕様を定めたもの

利用できる API が多いほどアプリケーションは作りやすく実現可能な範囲も広がっていく
Power Apps からは「コネクタ」と呼ばれる機能を経由して Web API を呼び出すことができる



API を利用した Power Apps の拡張



オンプレミス社内システムや
クラウド上で稼働する自社 API



既成の標準・プレミアムコネクタだけでも多くのことができるが、
Microsoft / 3rd Party から多種多様なコネクタが
コネクタの一覧

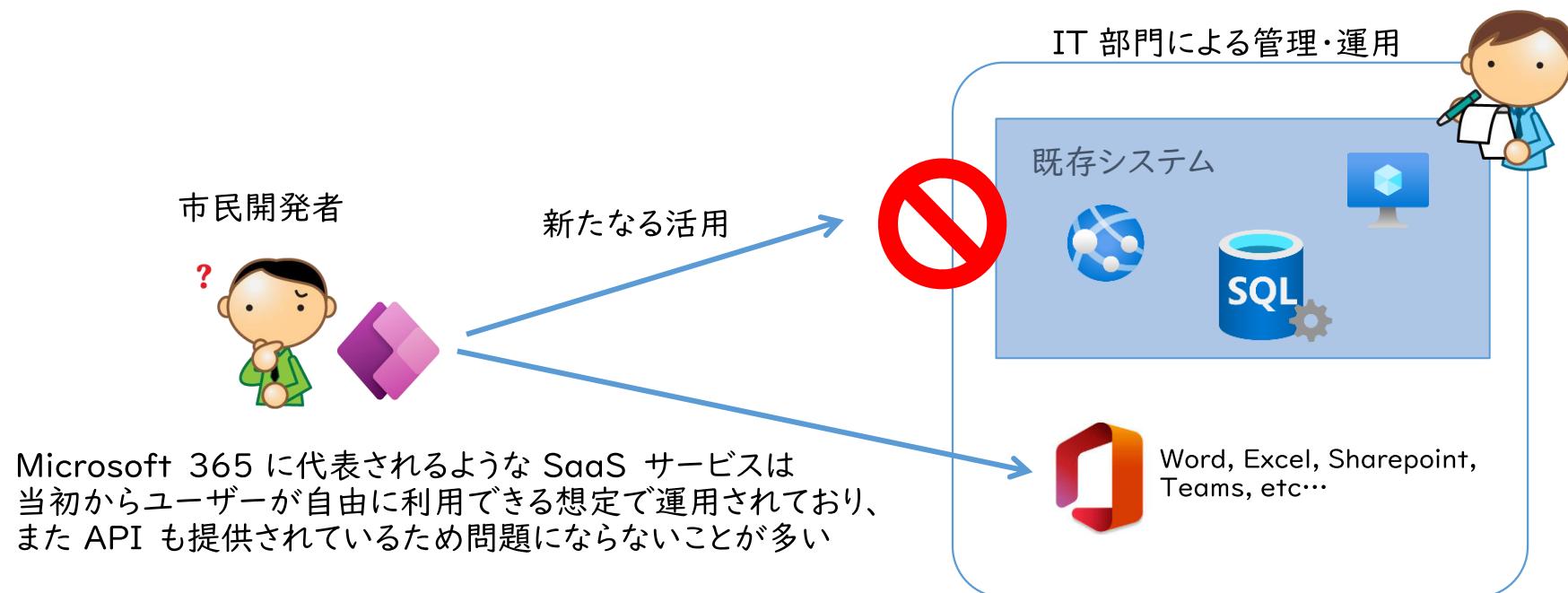
カスタムコネクタを独自開発すると
さらに多くの API が活用できる
オンプレミス、各種クラウドでホストする自社製 API
コネクタが提供されていない SaaS API
存在しない API はプロ開発者に作ってもらう

カスタム API の必要性

既存システムが元々 API による利活用を想定して設計・実装されていない限り Power Apps からは利用できない

当該システムが専用のクライアントや Web アプリなどのインターフェースしか許可できない場合は RPA ソリューションが必要になってくるが…

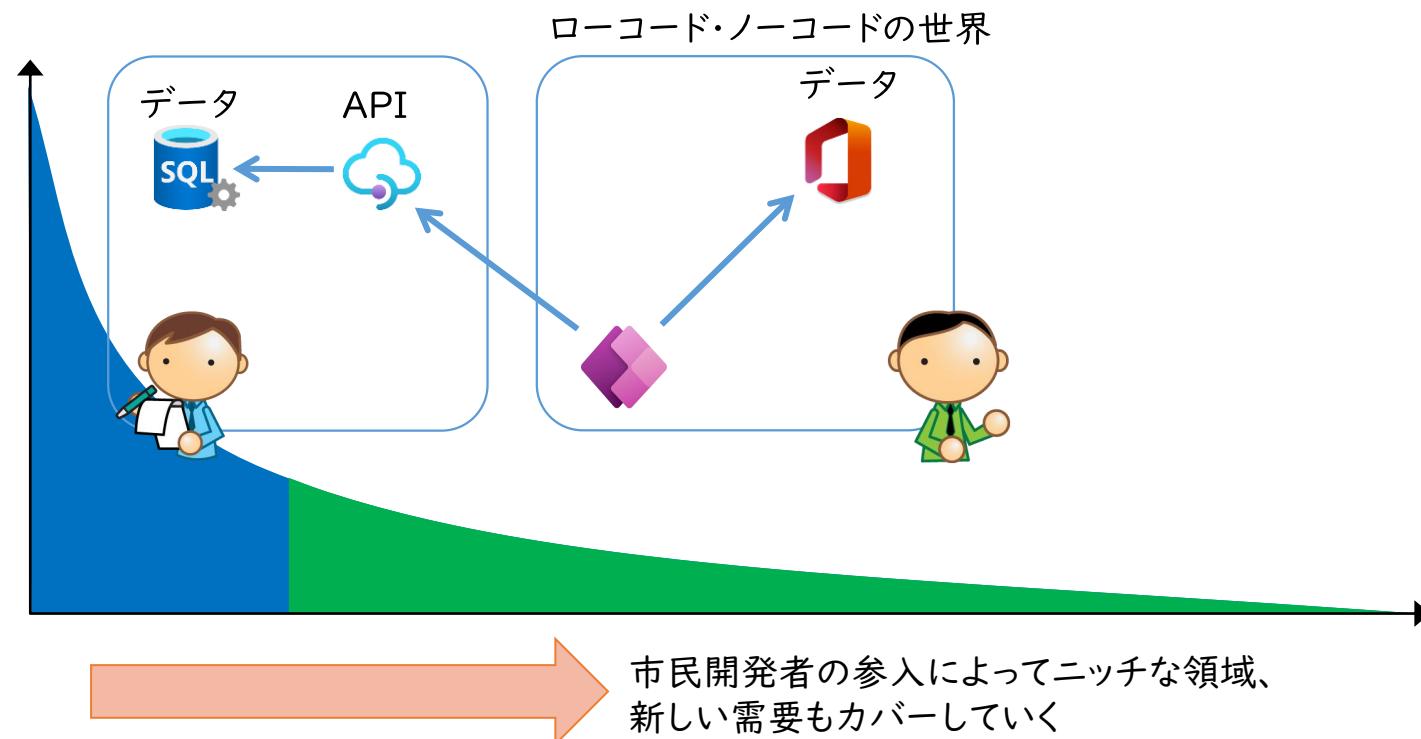
IT 部門がオフィシャルな API を提供することで、ユーザーが様々なアプリケーションを自身で開発することが可能になり、データの利活用による業務改善の促進が期待できる



スモールスタートで始める既存資産の利活用

既存システムに API がなければ新規構築が必要になるが、最初から全ての機能及びデータを抜け漏れなく提供する必要はない

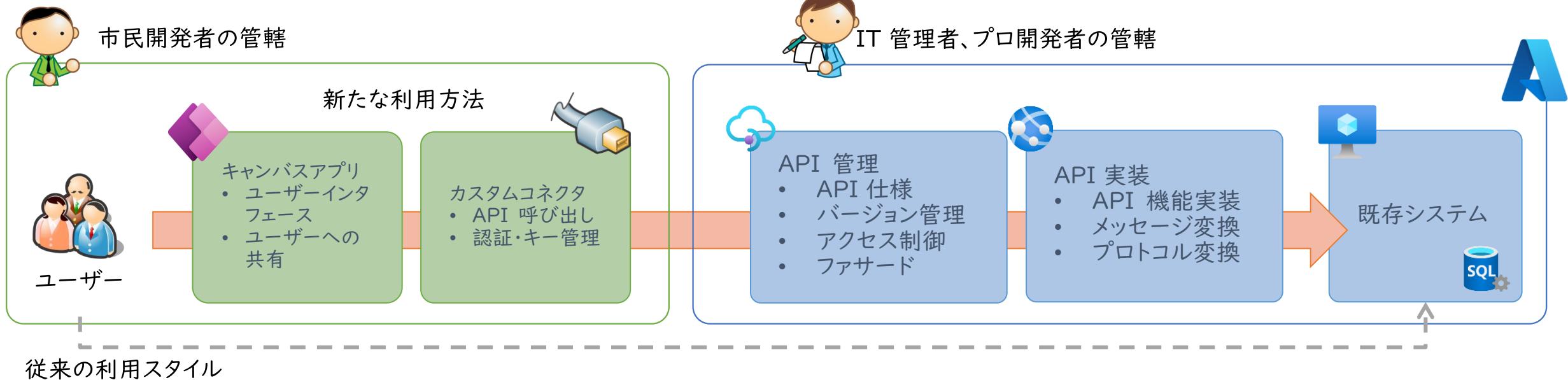
ユーザーの当面の業務目的を達成できる必要最低限のサブセットが提供できれば良い
システムを保護する意味でも IT 部門がサポートするオフィシャルな API であることが重要



アーキテクチャ

既存システムに使いやすい API がない場合には以下のような追加コンポーネントが必要になってくる

- | | |
|--------|---------------------------------------|
| API 実装 | : 既存システムが提供しているインターフェースをWeb API に変換する |
| API 管理 | : API の仕様、バージョン管理、利用状況などを集中管理する |
| キャンバス | : ユーザーが必要とする機能やデータを使いやすい「画面」として提供する |
| コネクタ | : API の定義情報と認証情報を管理し、画面から呼びだすための部品 |



Develop faster than ever before

Azure + Power Platform = 高速なアプリケーション開発

Power
Platform



Power Apps



Power Automate



Power Virtual Agents



Power BI

全ての開発者
(ローコード)

Azure
services



API
Management



Azure
Functions



AKS



Logic
Apps



Cognitive
Services



Bot
Services



Analysis
Services

プロ開発者
(Code First)

Azure / Office
Data Services



Microsoft
Graph



SQL Azure



Azure Synapse
Analytics



Cosmos DB



Visual
Studio



VS
Code



GitHub

Power Apps: a low-code approach to building apps



ウェブアプリやモバイルアプリを簡単に構築
フル機能のローコード/ノーコードプラット
フォーム



400以上の既製コネクタとカスタムコネクタを
使用して既存のデータに接続



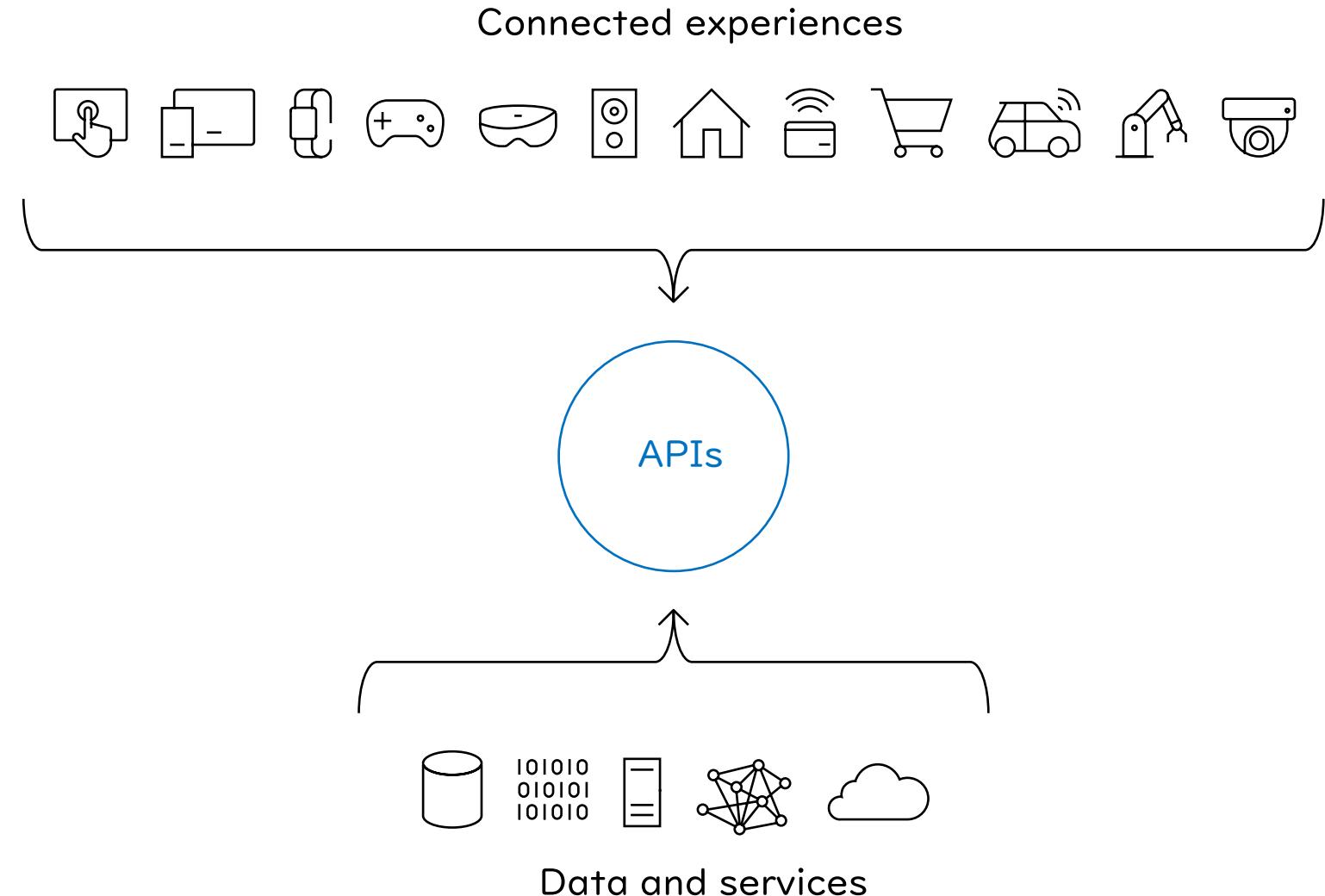
強力なエンタープライズガバナンスと
セキュリティ



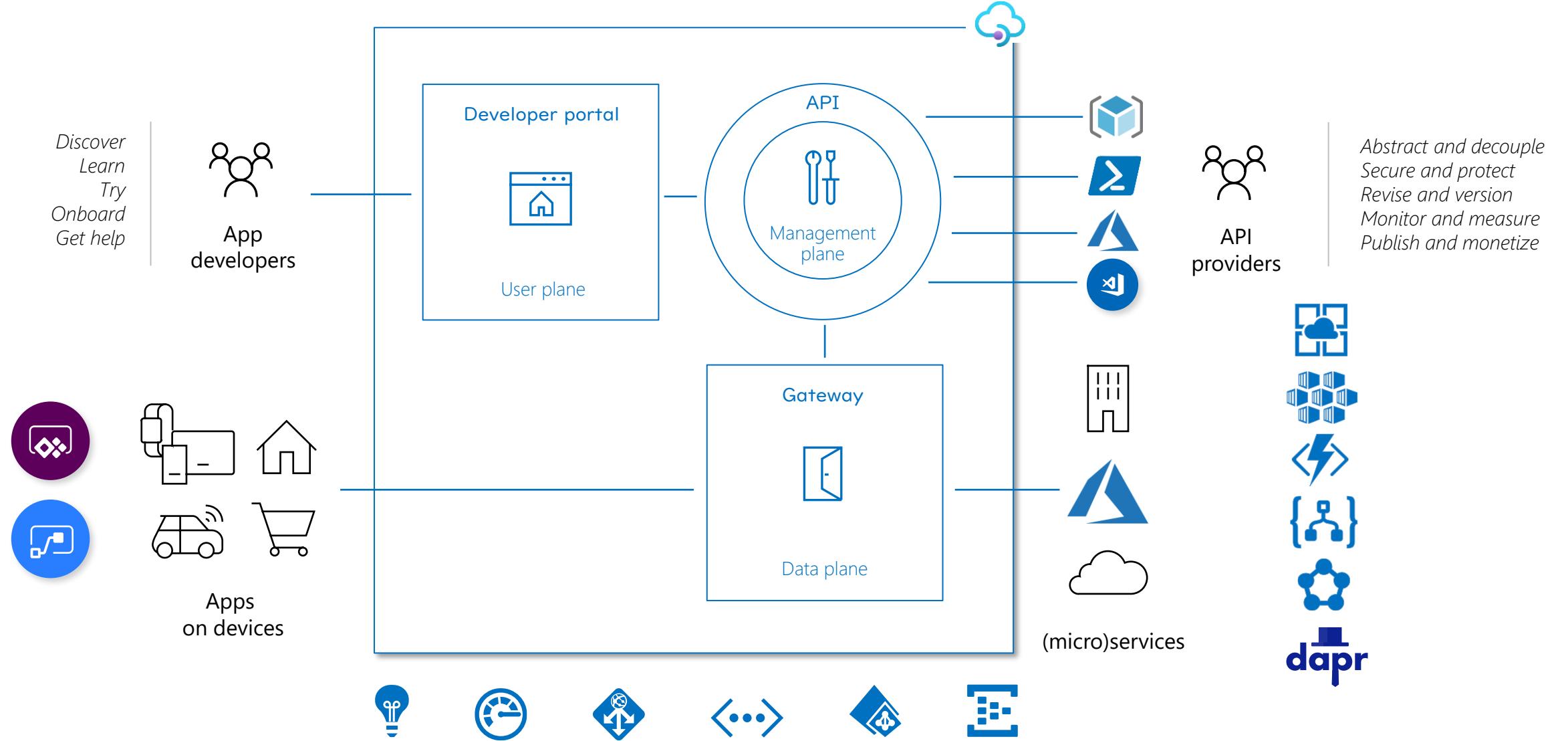
Pro-devの拡張性を実現
"無制限開発"



Azure API Management

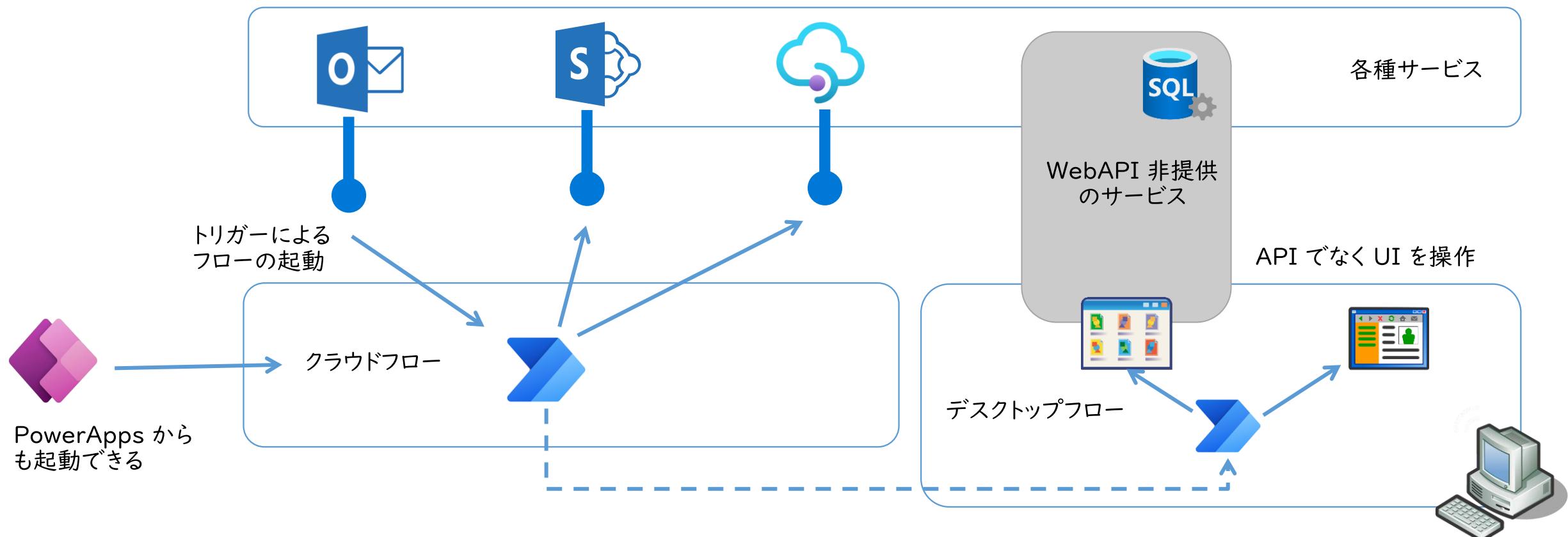


Facade, front door, and frictionless consumption



補足：Automate によるワークフロー自動化

クラウドフロー : 複数の API の呼び出しを無人操作
デスクトップフロー : UI の操作を自動化 (RPA)



カスタムコネクタ利用時のライセンス上の留意事項

Power Apps からカスタムコネクタを作成・利用するにはスタンダードアロンのライセンス(Per App/Per User)が必要

Power Apps から Power Automate のフローを呼び出す場合には、フロー内でもカスタムコネクタを利用することができる

Power Automate 単独でカスタムコネクタを利用したい場合にはやはりスタンダードアロンのライセンス(Per Flow / Per User)が必要になる

[Power Automate ライセンスの種類 - Power Platform | Microsoft Docs](#)

PowerApps for Microsoft 365 ライセンスを使用してカスタムコネクタを作成・利用する場合には以下の制限に注意

アプリやフローが Teams コンテキストで実行されている必要がある

カスタムコネクタの呼び出し先が Azure API Management であること

[Power Apps と Power Automate のライセンスに関するよくあるご質問 - Power Platform | Microsoft Docs](#)

[Known issues and limitations for Dataverse for Teams - Power Apps | Microsoft Docs](#)

ただしこの場合は Power Automate からのカスタムコネクタ利用ができない

Module I

Power Apps とカスタムコネクタ

本モジュールの目的

内容

ここではカスタムコネクタを作成し、キャンバスアプリから利用する方法を習得する
既存の資源を API として利用できることの価値と可能性を体験する

前提

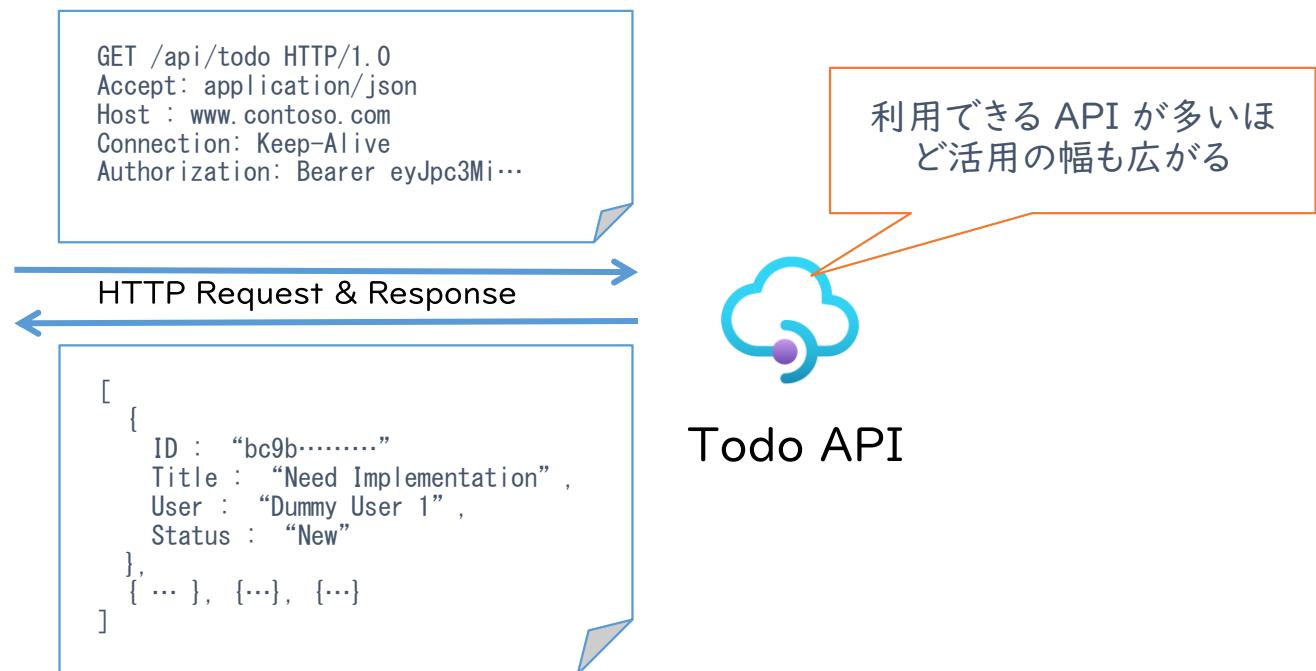
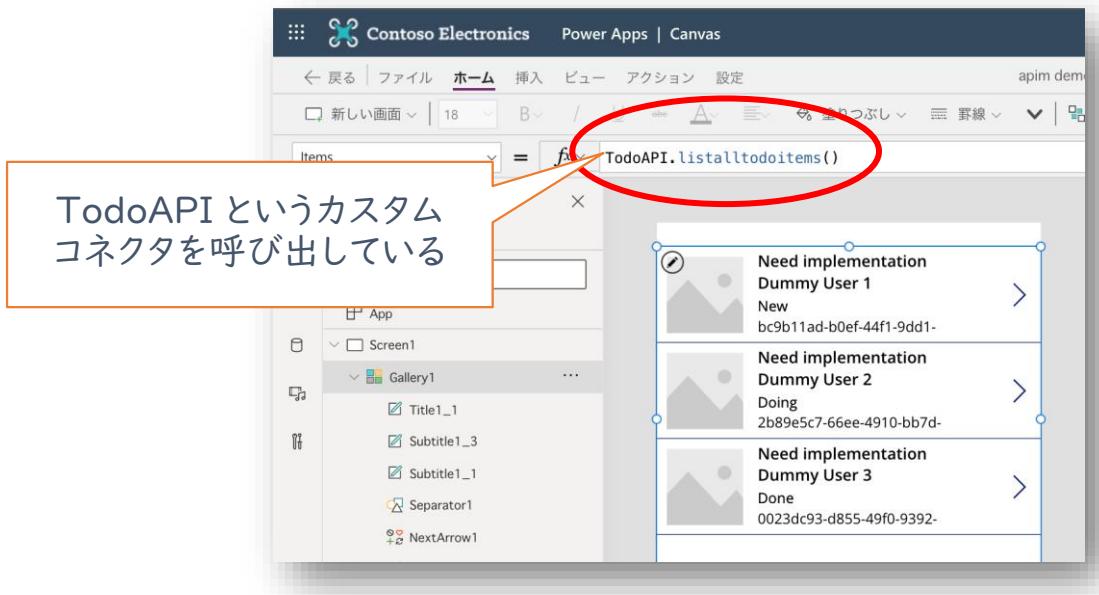
キャンバスアプリの作成に関する基礎知識が必要
API は既に用意されているものを利用する

Power Apps とカスタムコネクタ

API を呼び出すためにはコネクタが必要だが、一般提供されていないコネクタは自分で作ればよい

コネクタ = Web API を呼び出すためのラッパーで、Power Apps アプリからは 関数 として利用することが出来る

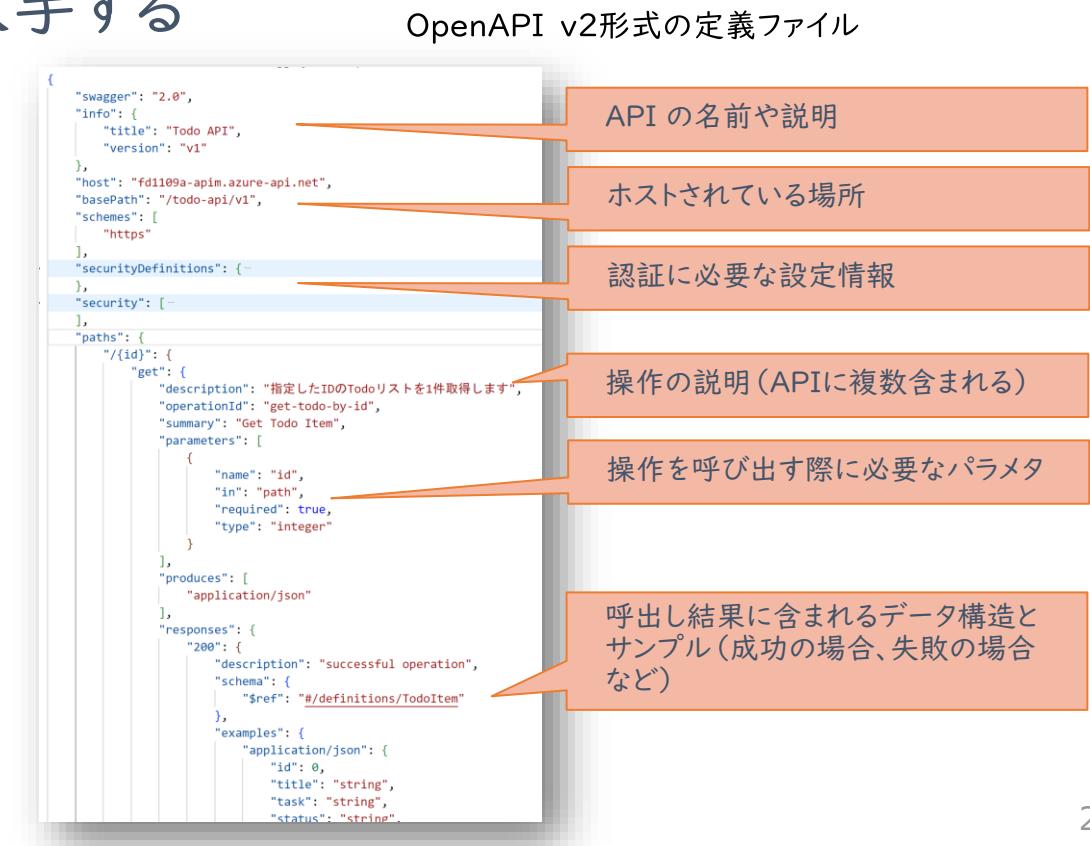
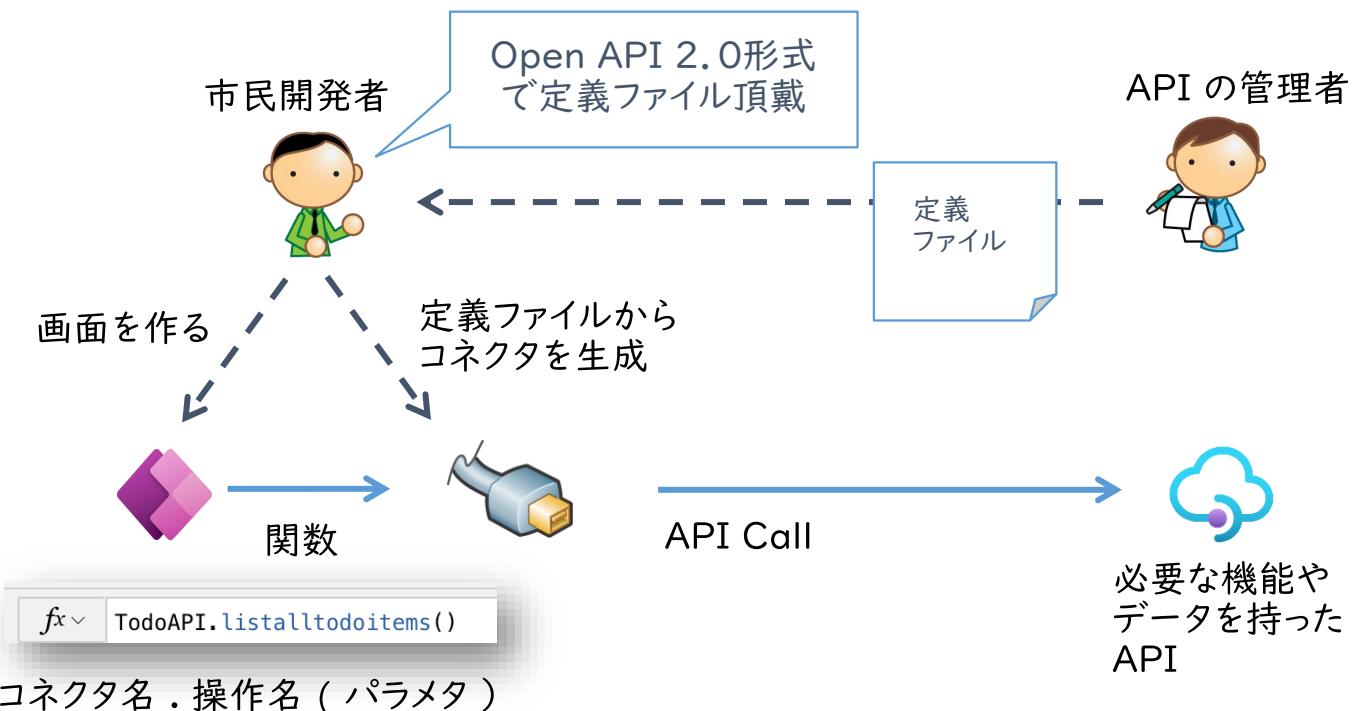
Web API の呼び出しのお作法やデータの送受信はコネクタ内部で行われるため、利用者は詳細を意識する必要なく、アプリの機能を拡張することが出来る



カスタムコネクタの作り方

カスタムコネクタの作り方はいくつかあるが、OpenAPI 定義から自動生成してしまうのが手っ取り早い

OpenAPI 定義 = Web API を呼び出すためのお作法が記述されたファイル
API の管理者に依頼して Open API 2.0 形式で入手する

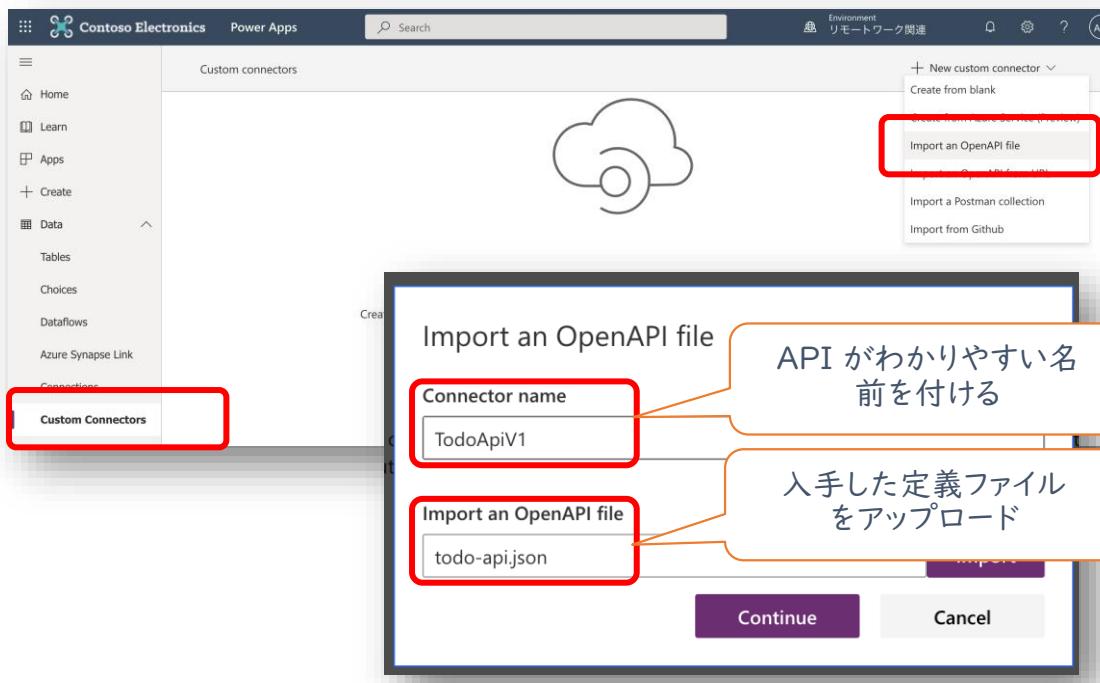


カスタムコネクタの作り方

Power Apps ポータルを使用して API 仕様を基にカスタムコネクタを作成する

<https://make.powerapps.com>

Open API 仕様がしっかり作りこまれていれば
ほとんどカスタマイズする必要はない



API で利用できる操作

Actions (4)

Actions determine the operations that users can perform. Actions can be used to read, create, update or delete resources in the underlying connector.

- 1 list-all-todo-item...
- 2 new-todo-item...
- 3 update-todo-item...
- 4 get-todo-item-by-id...

New action

References (2)

References are reusable parameters used by both actions and triggers.

- 1 ToDoltem
- 2 ArrayOfTodoltem

Policies (1)

Policies are used to change the behavior of actions and triggers through configuration. You can use one or more policies from a set of predefined templates.

- 1 Define api ver...

New policy

Policy details

Name *

Define api version

Template * Learn more

Set query string parameter

Adds or updates value of request query string parameter

Operations

List of actions and triggers to which the policy will apply to. If no operation is selected, this policy will apply to all operations.

list-all-todo-items, new-todo-item, update-todo-item, get-todo-item-by-id

Query parameter name *

Specifies the name of the query parameter to be set.

api-version

Query parameter value *

Specifies the value of the query parameter to be set.

v1

Action if query parameter exists

Specifies what action to take when the query parameter is already specified.

override

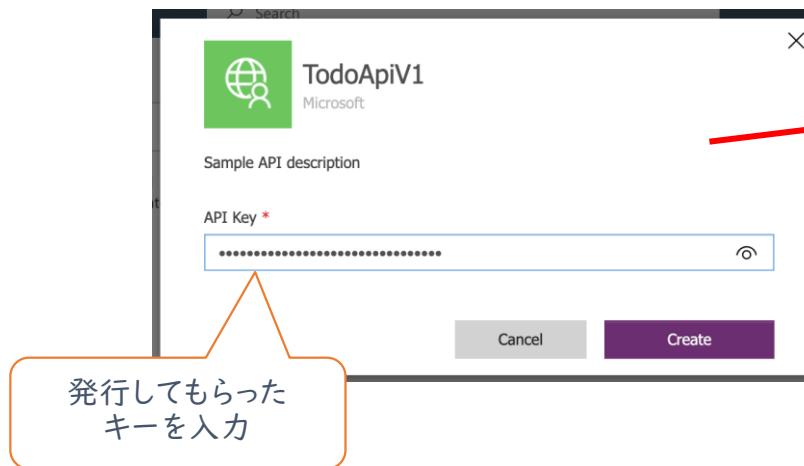
Security

Code (Preview) →

カスタムコネクタの動作確認

作成したコネクタから「接続」を作成し、動作確認を行う
接続の作成時に API を呼び出すキーが必要なため、こちらも管理者に発行してもらうとよい

接続(Connection)を作成



The screenshot shows the 'Test operation' and 'Connections' sections of the Azure portal. In the 'Connections' section, a connection named 'TodoApiV1_userkey' is selected. A red arrow points from the 'Selected connection' dropdown to a callout bubble containing the text '生成した接続を利用して' (Use the generated connection). In the 'Test operation' section, under 'Operations (4)', the 'list-all-todo-items' operation is selected. A callout bubble points to the 'Test operation' button with the text 'テスト実行' (Run test). The 'Request' tab shows a status code of 200, and the 'Response' tab displays a JSON response body:

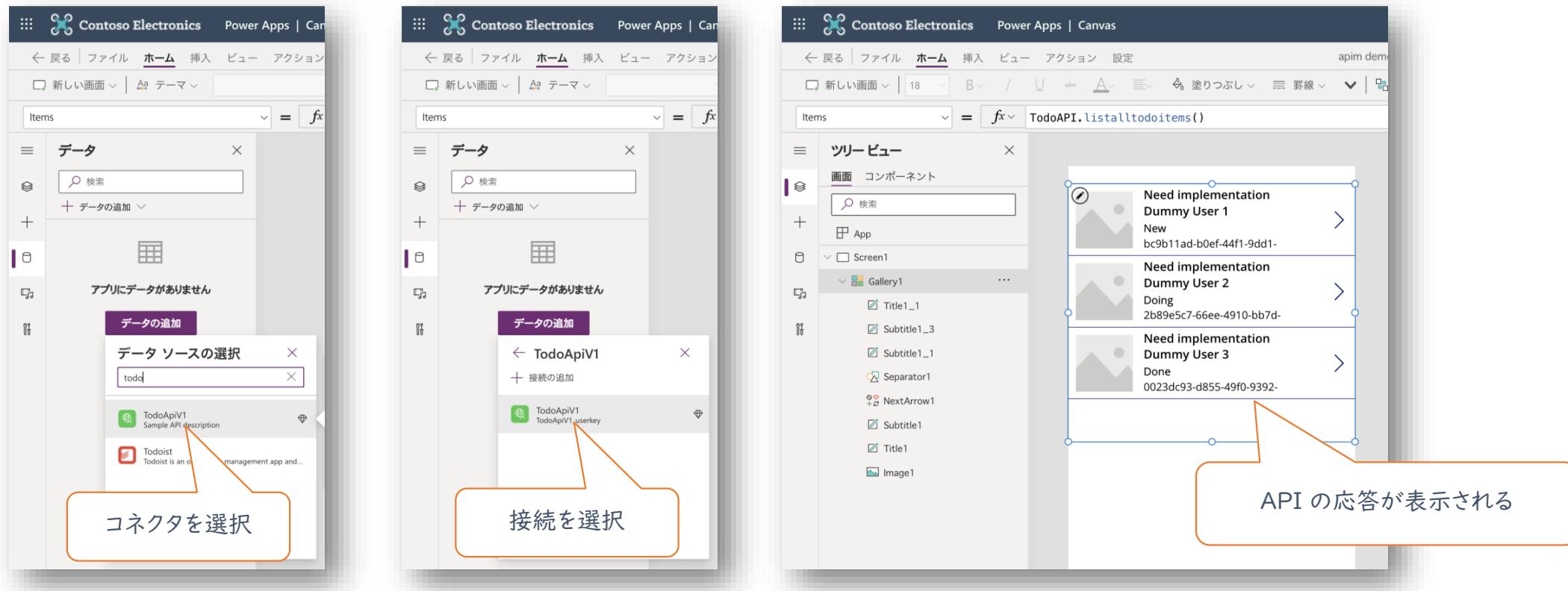
```
content-encoding: "gzip"
content-type: "application/json; charset=utf-8",
date: "Thu, 18 Nov 2021 09:42:07 GMT",
request-context: "appId=cid-v1:4494a9f3-76fd-4655-b614-c46762f42981",
vary: "Accept-Encoding,Origin",
"x-ms-apihub-cached-response": "true",
x-powered-by: "ASP.NET"

[{"id": "969292fd-dada-487b-91c4-3ced21d8fec", "Title": "Need implementation", "Owner": "Dummy User 1", "Status": "New"}, {"id": "4accd4d7-04e0-4260-9261-d27a0eaedcd", "Title": "Test item", "Owner": "Test User", "Status": "In Progress"}]
```

キャンバスアプリの開発

作成したコネクタと接続を使用してキャンバスアプリから呼び出せる
ことを確認する

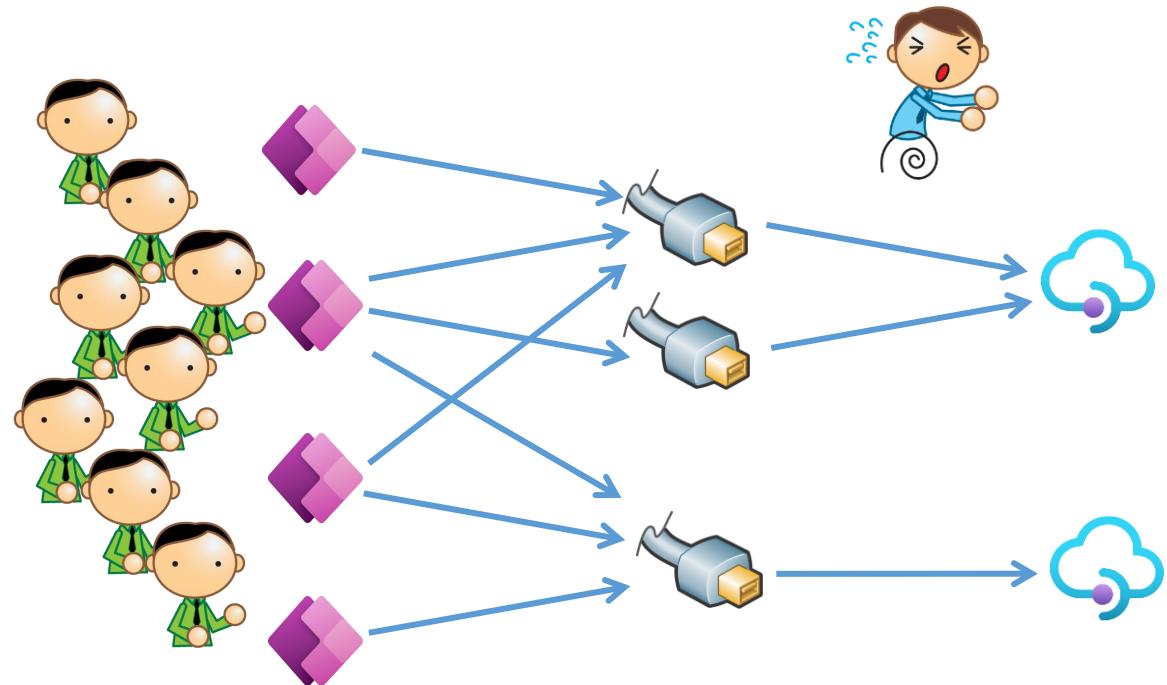
OpenAPI 定義に記載されていた操作(Operation) がそのまま Power Apps の関数になる
この例では Todo の一覧が変わってくるため、ギャラリーを使用して一覧表示している



API エコシステム

開発者とアプリと API が多いほど利活用が促進されるが、以下の課題も発生する

- そもそも API はどうやって発見する？
- 必要な API が無い場合はどうする？
- 呼び出すための認証情報はどこで取得する？
- API が乱立するとセキュリティや管理は？
- 作成したアプリの保守は誰が行うのか？
- アプリ、API、コネクタのバージョン管理は？



以降ではこれらの問題と対応方針について解説する

補足：API Management とカスタムコネクタ

前述の方法は OpenAPI 定義が取得できる API であれば、同じ操作でカスタムコネクタを作成できる

Power Apps から利用する Web API が Azure で実行されている必要はない

若干手間はかかるが汎用性が高いのでこの方法がおススメ

API Management から作成・更新することも可能だが以下の制限があるため、Azure を利用するプロ開発者が作成した API を自分でテストする場合などに活用するとよい

Power Apps の開発者が API Management へのアクセス権 (RBAC) も保有している必要がある

Azure サブスクリプションと同一の Azure AD テナントで管理されている Power Apps 環境にのみ作成可能

API Management 以外の Functions や App Service 等で提供されている API には対応できない

The image shows two screenshots illustrating the process of creating a custom connector in Azure API Management and its corresponding representation in the Power Apps environment.

Left Screenshot (Microsoft Azure API Management):

- The title bar says "fd1109a-apim | Power Platform API Management サービス".
- The left sidebar includes "Settings", "APIs", and "Power Platform".
- The main area is titled "コネクタの作成" (Connector Creation).
- Under "API", it says "Power Platform に接続する API を選択します。" and shows a dropdown menu with "Todo API v1" selected.
- Under "Power Apps", it says "API を発行する Power Apps 環境を選択し、Power P..." and shows a dropdown menu with "ESLZ admin's Environment (org35d2d81a)" selected.
- A large orange callout box highlights the "Power Apps 環境" dropdown, with the text "コネクタを作成する Power Apps 環境が表示される" (The Power Apps environment where the connector is created is displayed).
- A blue arrow points from the "作成・更新" (Create/Update) button at the top right of the Azure interface towards the Power Apps interface.

Right Screenshot (Power Apps):

- The title bar says "ESLZ admin's Environment..."
- The left sidebar includes "ホーム", "詳細", "アプリ", "作成", "Dataverse", "テーブル", "選択肢(複数)", "データフロー", "Azure Synapse Link", and "接続".
- The main area is titled "カスタム コネクタ" (Custom Connector).
- A table lists one item: "Todo API Auto Generate Inaba Ayumu" with an "アイコン" (Icon) column showing a globe icon.
- A blue arrow points from the "作成・更新" (Create/Update) button at the top right of the Azure interface towards the Power Apps interface.

補足：Power Apps の「コネクタ」と「接続」



API の呼び出しに必要なパラメータ（パラメータやデータ構造など）はコネクタで定義されている
カスタムコネクタの場合は Open API 定義等から生成されたもの
標準コネクタや 3rd Party 製品の場合は参照出来ない

実際に API を呼び出す際に必要な認証情報は
「接続」で管理されている

カスタムコネクタの動作確認の際には実際に呼出しが必要なので「接続」を作成している

同じコネクタを利用する複数のアプリやユーザーであっても、使用している接続は異なることが多い

Module 2

市民開発者のセルフサービス開発

本モジュールの目的

内容

本モジュールでは開発者自身が必要とする API を発見し利用する方法を習得する
API の利用をセルフサービスとすることで社内資源の利活用を促進する

前提

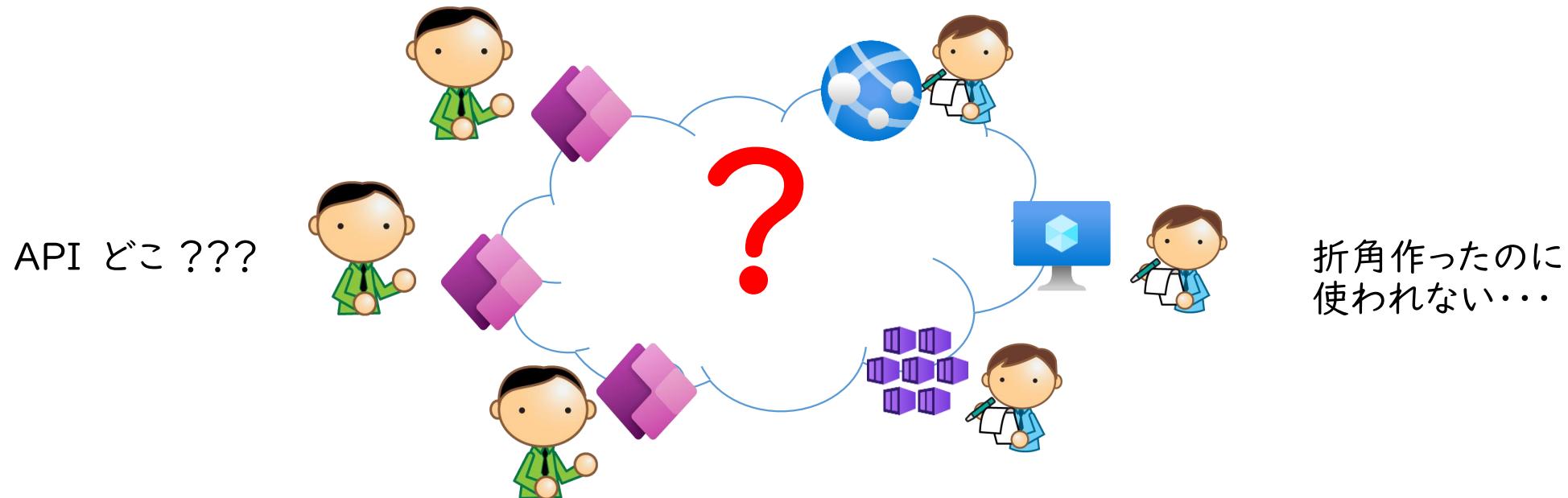
キャンバスアプリの作成に関する基礎知識が必要
API は既に用意されているものを利用する

API が見つからない問題

前述のようにカスタムコネクタを作り API を呼び出すこと自体はそれほど難しくはないが、..

そもそも API が存在するのか？ 存在するとして利用するために必要な手続きは？ 具体的にどんな機能とデータが提供されるのか？

標準コネクタであればインターネットで様々な情報が探せるが、社内システムなどクローズドな API は専用のポータルなどが必要になってくる

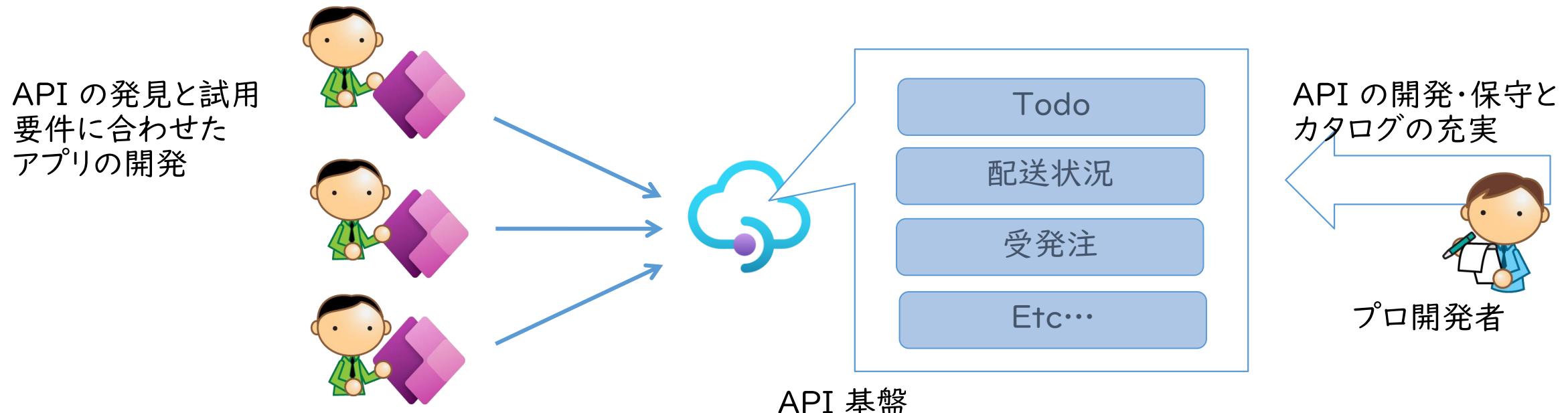


開発者セルフサービスによる API の活用

市民開発者が任意のタイミングで API 仕様の確認、キーの発行、テスト実行ができることが望ましい

API は特定の市民開発者の特定のキャンバスアプリだけではなく、さまざまなアプリやフローから広範に活用されることでその価値を増す

プロ開発者が API を呼び出すユーザーや認証情報を管理するのではなく、市民開発者自身が必要な時に必要な API を自由に呼び出せるプラットフォームを用意する



Azure API Management 開発者ポータル

API Management 付属の開発者ポータルを利用すると、市民開発者は自身で API を発見、必要な情報を入手できる

開発者ポータルは既定では有効になっていないため、有効化の設定が必要（後述）

プロ開発者も開発・保守している API を API Management に登録しておくだけでポータルに自動的に反映され、利用促進の効果が期待できる

The image displays three side-by-side screenshots of the Azure API Management developer portal for the organization "contoso".

- Left Screenshot (Sign in page):** Shows the sign-in interface with fields for "Email *" (contoso@contoso.com) and "Password". A yellow "Sign in" button is highlighted. At the bottom, there's a link to "Forgot your password?" and a "Azure Active Directory" button.
- Middle Screenshot (Products page):** Shows the "Products" section with a search bar and a table of products:

Name	Description
API for Power Platform	Power Apps などから呼び出す API 群です
Starter	Subscribers will be able to run 5 calls/minute up to a maximum of 100 calls/week.
Unlimited	Subscribers have completely unlimited access to the API. Administrator approval is required.
- Right Screenshot (User profile page):** Shows the "User profile" section with "Account details" and "Subscriptions".

Subscription details		Product	State	Action
Name	subscribe-from-1206	API for Power Platform	Active	Cancel
Started on	12/06/2021			
Primary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	Show Regenerate		
Secondary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	Show Regenerate		

初回利用時はサインアップ

初めて開発者ポータルを利用する場合はサインアップ[®](利用登録)
が必要になる

利用者情報と登録するとメールで確認が届くため、受信可能なメールアドレスを利用すること
既にサインアップ済みの場合は、登録したメールアドレスとパスワードでサインインできるようになる

Sign in

Not a member yet? [Sign up.](#)

Email *

Password *

Sign in

[Forgot your password?](#)

Azure AD 認証が使
えるとより望ましい

Sign up

Already a member? [Sign in.](#)

Email *

Password *

Confirm password *

First name *

Last name *

Enter the characters you see.

New | Audio



Please confirm your new fd1126b API account

"fd1126b" <apimgmt-noreply@mail.windowsazure.com> <apimgmt-noreply@mail.windowsa [✉](#)

14:02

宛先: ayumu.inaba@live.com

Dear Ayumu Inaba,

Thank you for joining the fd1126b API program! We host a growing number of cool APIs and strive to provide an awesome experience for API developers.

First order of business is to activate your account and get you going. To that end, please click on the following link:

<https://fd1126b-apim.devapphosting.azurewebsites.net/api/auth/confirm?userId=6385925b217d2440&ticketId=6385925b217d2440&code=1e92>

If clicking the link does not work, please copy-and-paste or re-type it into your browser's address bar and hit "Enter".

Thank you,

fd1126b API Team

fd1126b-apim.developer.azure-api.net

API の発見とサブスクリプション

開発者ポータルにサインイン出来たら、試したい API を含む
製品(Products)を探す

製品をみつけたら Subscribe することで API を利用するためのキーが払い出される

製品名

Starter

Subscribers will be able to run 5 calls/minute up to a maximum of 100 calls/week.

Your subscriptions

You don't have subscriptions yet.

subsc1

Subscribe

APIs in the product

Search APIs

Name Description

Echo API

試したい API

Todo API - v1

This screenshot shows a developer portal interface. On the left, there's a product page for 'Starter' with a red box around the product name. It includes a summary of usage limits, a section for 'Your subscriptions' (empty), and a 'Subscribe' button. Below that is a section for 'APIs in the product' with a search bar and a table showing 'Echo API'. A red box highlights the 'Todo API - v1' link. On the right, there's a 'User profile' section with account details like email and name, and a 'Subscriptions' section showing a single entry for 'subsc1' with red boxes around the 'Primary key' and 'Secondary key' fields.

User profile

Account details

Email: ayumu.inaba@live.com
First name: Ayumu
Last name: Inaba
Registration date: 11/29/2022

Change name Change password Close account

Subscriptions

Subscription details

Name	Started on	Product	State	Action
subsc1	11/29/2022	Starter	Active	Cancel

Primary key: XXXXXXXXXXXXXXXXXXXXXXXXXX
Secondary key: XXXXXXXXXXXXXXXXXXXXXXXXXX

Show | Regenerate

This screenshot shows a developer portal interface. On the left, there's a 'User profile' section with account details and a 'Subscriptions' section showing a single entry for 'subsc1' with red boxes around the 'Primary key' and 'Secondary key' fields. On the right, there's a 'Subscriptions' table with columns for Name, Started on, Product, State, and Action. The first row shows a subscription for 'subsc1' started on 11/29/2022, associated with the 'Starter' product, in an 'Active' state, with a 'Cancel' link in the Action column. There are also 'Show | Regenerate' links for the subscription keys.

API の試用と動作の確認



開発者ポータルでは仕様確認だけでなく呼び出してみることも可能

機能やデータが役立ちそうであれば、カスタムコネクタを実装するための定義ファイルをダウンロードする

The screenshot shows the Azure API Management developer portal for the 'contoso' tenant. The 'Todo API - v1' is selected. A callout box points to the 'API definition' dropdown menu, which is open and shows options: 'Open API 3 (YAML)', 'Open API 3 (JSON)', 'Open API 2 (JSON)', and 'WADL'. The 'Open API 2 (JSON)' option is highlighted with a red box. Another callout box points to the 'Try it' button, which is also highlighted with a red box. The overall interface includes sections for 'Search operations', 'Group by tag', and various API endpoints like 'List all todo items' and 'Request'.

The screenshot shows the Azure API Management test interface. It displays the 'Parameters' and 'Headers' sections, with a specific 'Ocp-Apim-Subscription-Key' header highlighted with a red box and a callout box stating '自分で発行したキーを利用して呼びだす' (Use the key you issued to call it). Below this is the 'HTTP response' section, which shows a successful 'HTTP/1.1 200 OK' response with the following JSON data:

```
HTTP/1.1 200 OK
content-encoding: gzip
content-type: application/json; charset=utf-8
date: Mon, 06 Dec 2021 02:51:49 GMT
request-context: appId=cid-v1:4850bdc-0980-4af3-84b1-70ac0ef4bbc0
strict-transport-security: max-age=31536000
transfer-encoding: chunked
vary: Accept-Encoding
x-powered-by: ASP.NET

[{"id": "d943ce17-ed5e-4e96-9335-3d2db587e1ce", "Title": "Need implementation", "Owner": "Dummy User 1", "Status": "New"}, {"id": "3adbca32-393e-429f-b717-0813871eb459", "Title": "Need implementation", "Owner": "Dummy User 2", "Status": "Doing"}, {"id": "c3643377-2dca-4a8e-b7fe-448a7f04981f", "Title": "Need implementation", "Owner": "Dummy User 3", "Status": "Done"}]
```

カスタムコネクタの開発と API の呼び出し

開発者ポータルから Open API 仕様と API キーが取得できれば、前述の手順に従って Power Apps に組み込めばよい

Actions (4)

Actions determine the operations that users can perform. Actions can be used to read, create, update or delete resources in the underlying connector.

- 1 list-all-todo-item...
- 2 new-todo-item...
- 3 update-todo-item...
- 4 get-todo-item...

New action

References (2)

References are reusable parameters used by both actions and triggers.

- 1 ToDoltem
- 2 ArrayOfTodoItem

Policies (1)

Policies are used to change the behavior of actions and triggers through configuration. You can use one or more policies from a set of predefined templates.

Define api ver...

New policy

Policy details

Name * Define api version

Template * Learn more Set query string parameter

Adds or updates value of request query string parameter

Operations

List of actions and triggers to which the policy will apply to. If no operation is selected, this policy will apply to all operations.

list-all-todo-items, new-todo-item, update-todo-item, get-todo-item-by-id

Query parameter name *

Specifies the name of the query parameter to be set.

api-version

Query parameter value *

Specifies the value of the query parameter to be set.

v1

Action if query parameter exists

Specifies what action to take when the query parameter is already specified.

override

← Security Code (Preview) →

Contoso Electronics Power Apps | Canvas

← 戻る ファイル ホーム 挿入 ビュー アクション 設定 apim dem

新しい画面 v 18 B / U A 塗りつぶし 範囲 線

Items = fx TodoAPI.listalltodoitems()

シリービュー

画面 コンポーネント 検索

App

Screen1

Gallery1

Title1_1
Subtitle1_3
Subtitle1_1
Separator1
NextArrow1
Subtitle1
Title1
Image1

Need implementation Dummy User 1 New bc9b11ad-b0ef-44f1-9dd1-

Need implementation Dummy User 2 Doing 2b89e5c7-66ee-4910-bb7d-

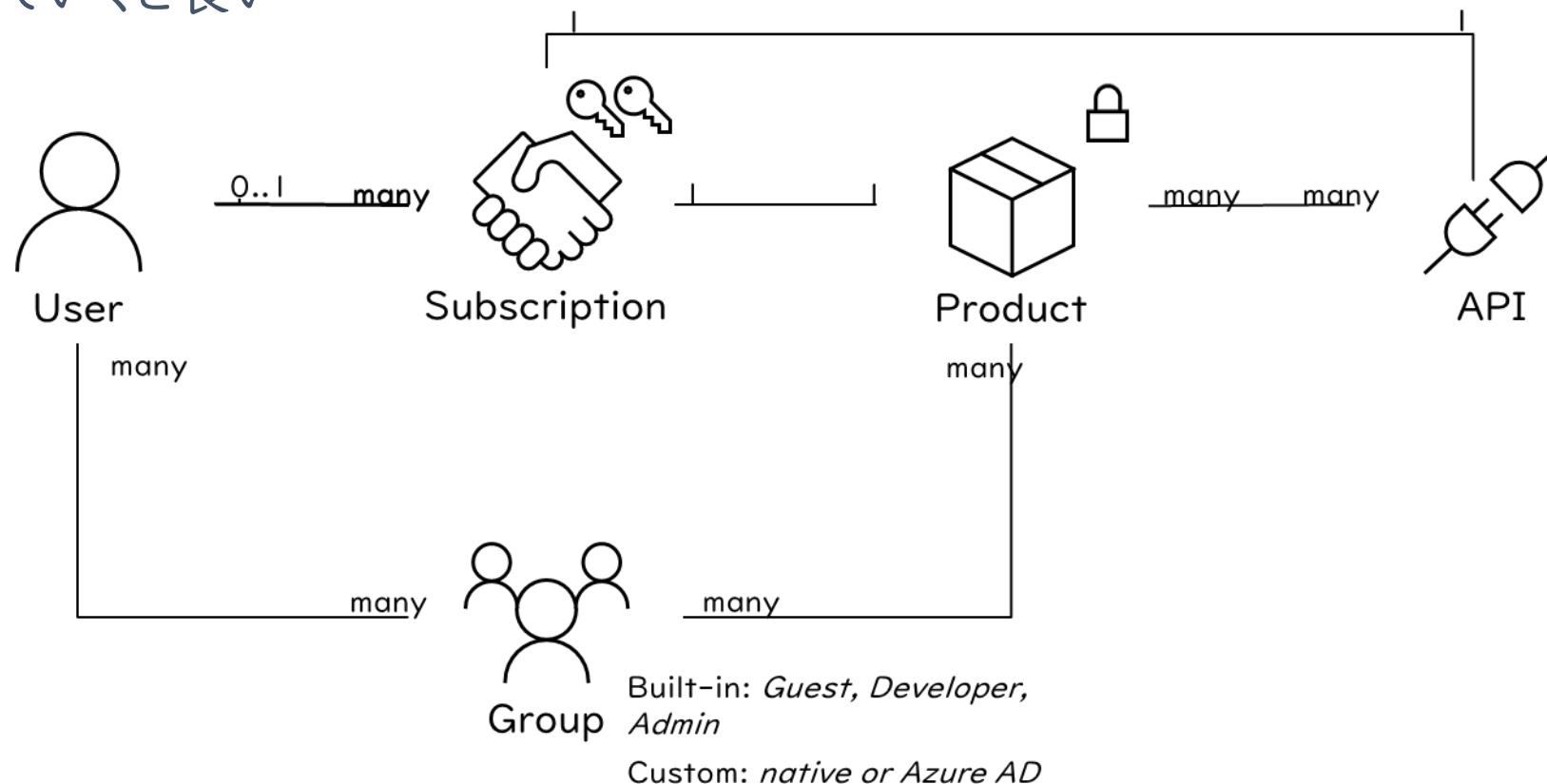
Need implementation Dummy User 3 Done 0023dc93-d855-49f0-9392-

[補足] API Management における製品と API

ユーザーはアクセス許可された「製品」を利用することができます

製品は1つ以上のAPIを束ねたもので、開発者はこの製品の単位でサブスクリプションを作成する
(同じ製品に含まれるAPIは同一キーで利用できる)

制限の異なる「製品版」と「試用版」、利用対象を分類した「一般向け」と「管理者向け」といった粒度で製品を管理していくと良い



Module 3

アプリと API の開発プロセス

本モジュールの目的

内容

必要な API が存在しない場合の開発プロセスを確認する
開発者間での合意事項や役割分担を確認する

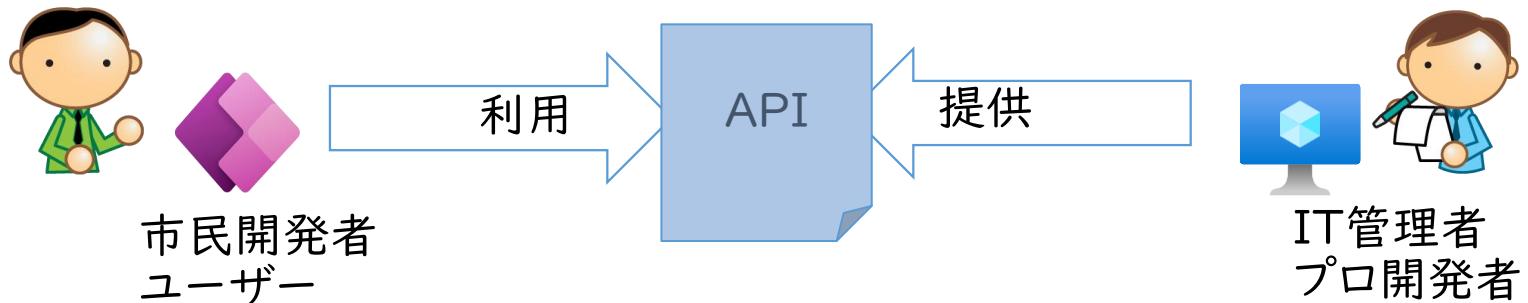
前提

Power Apps キャンバス アプリ及びカスタム コネクタの開発が出来ること
API Management や Web API 開発の基礎知識

API が存在しない問題

既存のシステムが管理している機能やデータを利活用したくとも、API がなければ使うことが出来ない

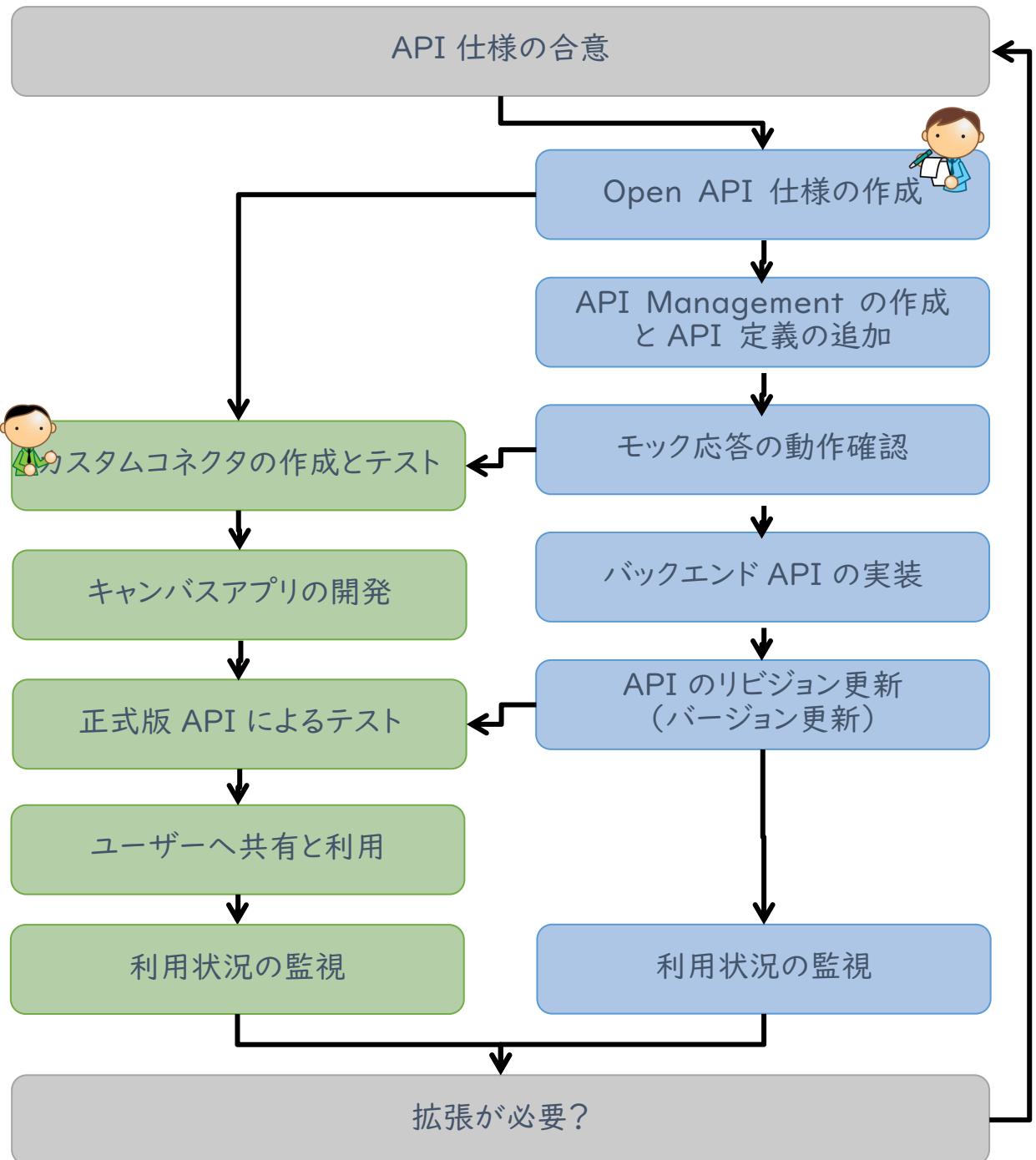
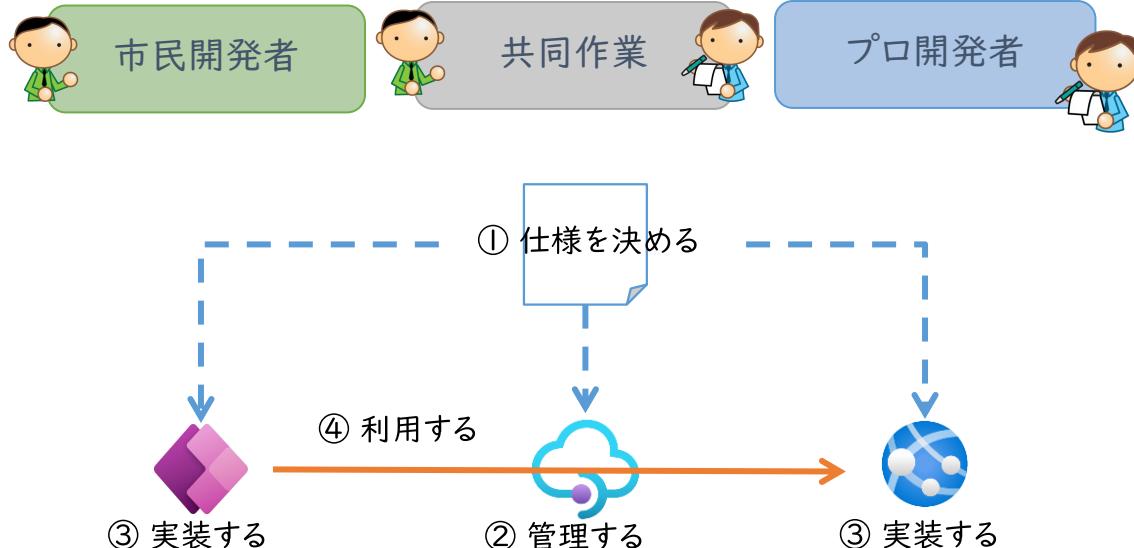
API 自体が存在しても市民開発者から直接利用されるシナリオが想定されていない場合や、データ構造や機能の粒度がアプリの実装には適していない場合もある
こういったケースではシステムを保守・運用するプロ開発者側と、API として利用したい市民開発者側の間で合意形成と新規開発が必要になる



開発の流れ

開発作業の大まかな流れは
右記のようになる

最初に API 仕様を決める コントラクト
ファースト アプローチを採用
ビジネス状況とユーザー要望に応じた継
続的な改善と拡張のため、バージョン管理
の戦略が重要になってくる

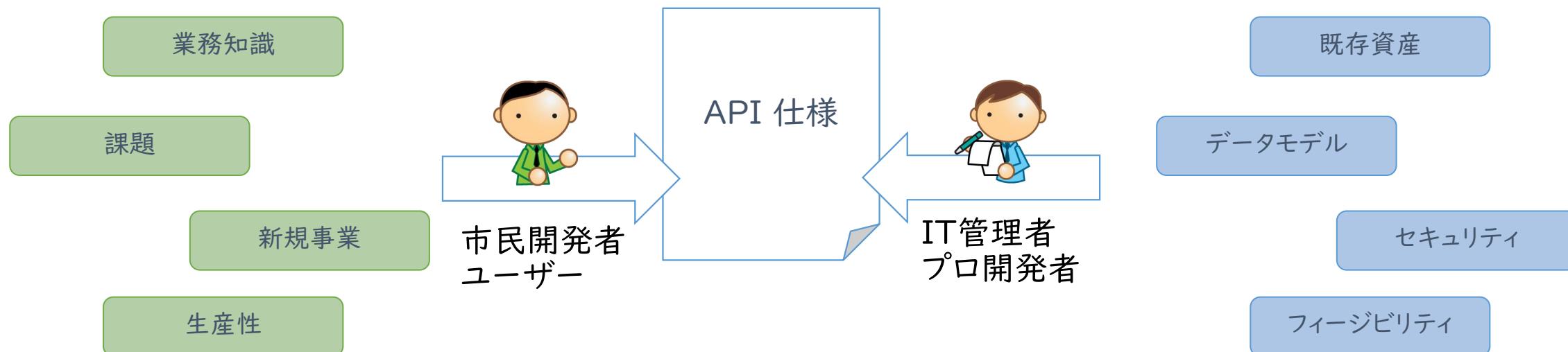


まずは API 仕様の合意

市民開発者やユーザーが必要とする機能やデータを API 仕様として定義する

提供する操作、リクエストの方法、レスポンスの種類、やりとりするデータの型、サンプルなどを定義
最終的には Open API 仕様に従ったファイルとしてまとめる(JSON/YAML形式)

JSON/YAML で手書きするのは難解なので [Swagger Editor](#) などのツールを活用するとよい
Azure API Management を利用して作成することも可能



Open API 仕様書の例

API として提供するリソース、操作、入出力データなどを仕様書として落とし込んでいく

API 設計のベストプラクティスなどは[こちら](#)が参考になる

```
openapi: 3.0.1
info:
  title: Todo API
  description: ''
  version: v1
servers:
  - url: https://fd1107-apim.azure-api.net/todos/v1
paths:
  /{id}:
    get:
      summary: Get Todo Item
      description: 指定したIDのTodoリストを1件取得します
      operationId: get-todo-by-id
      parameters:
        - name: id
          in: path
          required: true
          schema:
            type: integer
      responses:
        '200':
          description: successful operation
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/TodoItem'
              example:
                id: 0
                title: string
                task: string
                status: string
                dueDate: '2022-07-21'
                updatedOn: '2022-07-21T17:32:28.000000+00:00'
    put:
      summary: Update Todo Item
      description: 指定したIDのTodoリストを1件更新します
      operationId: update-todo-by-id
      parameters:
        - name: id
          in: path
          required: true
          schema:
            type: integer
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/TodoItem'
            example:
              id: 0
              title: string
              task: string
              status: string
              dueDate: '2022-07-21'
              updatedOn: '2022-07-21T17:32:28.000000+00:00'
```

Swagger Editor

```
openapi: 3.0.1
info:
  title: Todo API
  description: ''
  version: v1
servers:
  - url: https://fd1107-apim.azure-api.net/todos/v1
paths:
  /:
    get:
      summary: List All Todo Item
      description: 登録されているTodoリストを取得します
      operationId: list-all-todo-item
      responses:
        '200':
          description: successful operation
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/TodoList'
              example:
                [
                  {
                    "id": 0,
                    "title": "string",
                    "task": "string",
                    "status": "string",
                    "dueDate": "2022-07-21",
                    "updatedOn": "2022-07-21T17:32:28.000000+00:00"
                  },
                  {
                    "id": 1,
                    "title": "string",
                    "task": "string",
                    "status": "string",
                    "dueDate": "2022-07-21",
                    "updatedOn": "2022-07-21T17:32:28.000000+00:00"
                  }
                ]
```

VS Code Extension

[補足] Contract First or Code First

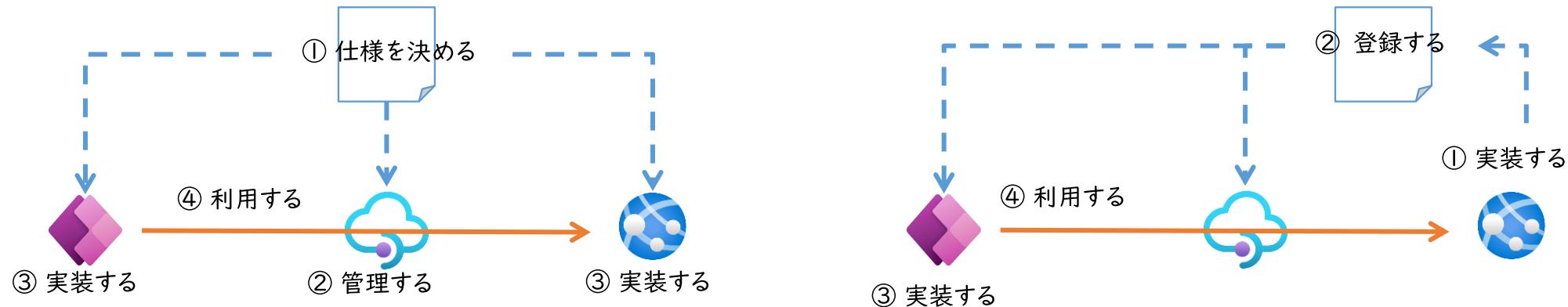
API 開発には大まかに2つのアプローチが考えられる

Contract First (Design First)

最初に API の仕様を決め、その仕様を主として API やクライアントアプリを実装する
バージョン管理の基点となる API の定義が明確に管理しやすく非互換問題も起こしにくく
開発者双方に OpenAPI 仕様の理解が必要となるため敷居が高く、比較的開発に時間がかかる

Code First (API First)

最初に各種プログラミング言語を使用して API を実装し、そこから OpenAPI 定義を生成する
プロ開発者が慣れ親しんだ技術を利用出来るため生産性が高く、すぐに動くものが手に入る
バージョン管理や実装が API 開発側の事情に影響されるため、必ずしも使いやすい API とはならない





API Management で API を管理する

必要な API の仕様が定まつたら Azure API Management にインポートする

この際に今後の API の仕様変更に備えてバージョニングを有効にしておくこと

ainaba-apim-1115 | API

API Management サービス

検索 (Cmd+)

開発者ポータル

概要

アクティビティ ログ

アクセス制御 (IAM)

タグ

問題の診断と解決

イベント (レビュー)

APIs

API

製品

サブスクリプション

名前付きの値

Backends

API タグ

Power Platform

開発者ポータル

ポータルの概要

Define a new API

HTTP

Manually define an HTTP API

WebSocket

Streaming, full-dup communication with server

+ Add API

All APIs

Echo API

Todo API

Create from definition

OpenAPI

Standard, language-agnostic interface to REST APIs

作成しておいた仕様
書を読み込ませる

Create an HTTP API

Basic Full

Display name Custo API demo

Name custo-api-demo

Description API の説明

Web service URL e.g. http://httpbin.org

URL scheme HTTP HTTPS Both

API URL suffix /custom-api-demo

Base URL https://ainaba-apim-1115.azure-api.net/custom-api-demo

Tags e.g. Booking

Products No products selected

To publish the API, you must associate it with a product. Learn more.

Gateways Managed

Version this API?

Version identifier v1

Versioning scheme Query string

Version query parameter api-version

Usage example https://ainaba-apim-1115.azure-api.net/custom-api-demo/[operation]

バックエンド API
がまだ存在しない
ので空で良い

バージョニングを
有効にする



API-M で API を定義する

API Management は API ゲートウェイだけではなく、API を定義するための「仕様書の作成ツール」にも使える

後述の手順で API を定義すると Open API 仕様書をエクスポートすることが出来る
仕様を事前にきっちり決めず、PoC として実装と再定義を繰り返すならこの方法がおススメ

API-M で仕様を定義する場合

API を作成
(手順は後述)

REVISION 1 | CREATED Nov 16, 2021, 2:45:

Design Settings Test Re

Search APIs Filter by tags Group by tag

+ Add API

All APIs

Echo API

Todo API

All operations

GET Get todo item by... v1

...

Clone Add revision Add version Import Export

Create Power Connector

API の定義



初期バージョン(ここでは v1)の API 仕様を作成していく

API が提供する Operation を追加して名前や URL を定義

URL テンプレート、Query、Header、Request ボディなどの入力パラメータを定義していく

応答の種類が複数ありうる場合には HTTP レスポンスコードごとに Content-Type、ボディ、ヘッダーを定義していく

(別途作成してインポートした場合は内容を確認)

The screenshot shows the 'ainaba-apim-1115 | API' management interface. On the left, there's a sidebar with 'Search APIs' and 'Filter by tags' buttons. Below that is a 'Group by tag' section with a checkbox. A red box highlights the '+ Add operation' button under the 'All APIs' section. In the main area, 'Todo API' is expanded, showing four operations: 'GET Get todo item by...', 'GET List all todo items...', 'POST New todo item...', and 'PATCH Update todo item...'. Each operation has a '...' button next to it.

The screenshot shows the 'Todo API > Add operation' dialog. It has tabs for 'Frontend' (selected), 'Template' (highlighted with a red box), 'Query', 'Headers', and 'Responses'. Under 'Frontend', fields are filled: 'Display name' (List Todo Items of specific types), 'Name' (list-todo-items-of-specific-types), 'URL' (GET /todo/{itemType}). Under 'Template parameters', there's one entry: 'itemType' with 'todo item type' in the description and 'string' in the type. A red box highlights the 'Template' tab.

The screenshot shows the 'Responses' tab for the Todo API operation. It has tabs for 'Template', 'Query', 'Headers', and 'Responses' (highlighted with a red box). Under 'Responses', there are two entries: '200 OK' and '204 No Content'. The '200 OK' entry has a 'Description' field and a 'Representations' table with a single row for 'application/json'. The '204 No Content' entry has an empty 'Description' field. Below these are 'Headers' and 'Definition' sections.

レスポンス系の定義

リクエスト系のパラメタ定義



データ型の定義

POSTリクエストやレスポンスボディで複合型を使う場合はデータ型の定義を追加する

複数の操作で同じデータ型を利用する場合には、共通のデータ型として定義しておくと良い

Representations

CONTENT TYPE SAMPLE DEFINITION DELETE

+ Add representation

+ New definition

Create new definition

Definition for Todo API > Response > Status code 200

* Definition name TodoItem1

Sample (JSON)

```
1 {  
2   "id": "e6086bc3-26a8-489d-a77a-511cf57c0acf",  
3   "Title": "Define all ",  
4   "Status": "New",  
5   "Owner": "Ayumu Inaba"  
6 }
```

Generate payload from sample Auto-generate payload from sample

* Payload

```
1 {  
2   "type": "object",  
3   "properties": {  
4     "id": {  
5       "type": "string"  
6     },  
7     "Title": {  
8       "type": "string"  
9     },  
10    "Status": {  
11      "type": "string"  
12    }  
13  }  
14 }  
15 
```

Representations

CONTENT TYPE SAMPLE DEFINITION

+ Add representation

DEFINITION

Select definition + New definition ArrayOfTodoItem id-GetRequest id-PatchRequest ToDoItem

Headers

NAME	DESCRIPTION	TYPE	VALUES
No headers to display.			

Design Settings Test Revisions Change log

Search definitions

+ Add definition

ToDoItem

Definition

* Name ToDoItem

Definition body

Schema Example

```
1 {  
2   "type": "object",  
3   "properties": {  
4     "id": {  
5       "type": "string",  
6       "format": "uuid"  
7     },  
8     "Title": {  
9       "type": "string"  
10    },  
11     "Owner": {  
12       "type": "string"  
13     },  
14     "Status": {  
15       "type": "string"  
16     }  
17   }  
18 }
```

Operations Definitions Save Discard



モック応答の有効化

この段階ではバックエンド API の実装が無く動作確認もできないため、モック応答を有効化しておく

Inbound processing policy として mock-response を追加しておくと、バックエンド API の代わりに応答してくれるようになる

各 Operation で設定した「サンプル」がダミーの応答として使用される（ない場合は自動生成）

The screenshot shows two panels of an API management tool. The left panel is titled 'Design' and displays the 'Frontend' section with operations like GET /, GET Get todo item by..., and GET List all todo items. The 'Responses' section shows a green '200 OK' button. The right panel is titled 'Inbound processing' and shows a flowchart where requests pass through 'base' and then 'mock-response'. A red box highlights the 'mock-response' policy. The right panel also shows the 'Test' tab selected, displaying the 'Todo API > List all todo items > Console' results. The results show a successful HTTP 1.1 200 OK response with a JSON payload containing three todo items.

```
HTTP/1.1 200 OK
content-length: 434
content-type: application/json
date: Thu, 18 Nov 2021 08:38:49 GMT
ocp-apim-apid: todo-api
ocp-apim-operationid: list-all-todo-items
ocp-apim-subscriptionid: master
ocp-apim-trace-location: https://apimstuskr60vx1lbidgyb1.blob.core.windows.net?sr=&sig=Hfehz4t7zJ0kpaTj1EYN0P%2F7Mb%2BWhNlmpie1B0LijM%3D&se=2021-11-18T08:38:49Z
request-context: appId=cid-v1:4494a9f3-76fd-4655-b614-c46762f42981
vary: Origin
[
  {
    "id": "e6086bc3-26a8-489d-a77a-511cf57c0acf",
    "Title": "Define all",
    "Owner": "Ayumu Inaba",
    "Status": "New"
  },
  {
    "id": "d1e8f68b-95d7-4d59-b7f1-f7cb7caeaa88",
    "Title": "Define all",
    "Owner": "Akira Koike",
    "Status": "Doing"
  },
  {
    "id": "d1e8f68b-95d7-4d59-b7f1-f7cb7caeaa88",
    "Title": "Define all",
    "Owner": "Mitsuhiko Takagi",
    "Status": "Done"
  }
]
```



API 仕様のエクスポート

API Management で管理している API 定義をエクスポートして市民開発者に渡す

2022年11月時点では Power Platform のカスタムコネクタは Open API 2.0 までしか対応していないので注意すること

The screenshot shows the Azure Portal's API Management interface. On the left, there's a sidebar with search and filter tools, and a main area for managing APIs. In the center, a specific API named 'Todo API' is selected. A dropdown menu is open over the API's name, showing options like 'Clone', 'Add revision', 'Add version', 'Import', 'Export', and 'Create Power Connector'. The 'Export' option is highlighted with a red box. At the bottom of the interface, there are 'Delete' and '...' buttons.

Azure Portal からエクスポート

The screenshot shows the 'Export API' dialog in the Azure Portal. It lists several options for exporting API definitions:

- OpenAPI v3 (YAML)
- OpenAPI v3 (JSON)
- OpenAPI v2 (JSON) (This option is highlighted with a red box.)
- </> (WADL)
- </> (WSDL) (This option has a purple diagonal banner labeled 'SOAP API only').

Each item has a brief description below it.

※ 開発者ポータルがセットアップされていればこの手順(=プロ開発者側の個別対応)は不要になる



テスト用の API キーの発行

市民開発者が API Management でホストされている API を呼び出すにはキー情報が必要になる

通常は開発者ポータルを用意しておいて、市民開発者に自身で発行してもらうことになるが、この方式に関しては後述する

ここでは簡易的に管理者側でサブスクリプションとキーを生成して伝達することとする

表示名	主キー	2 次キー	スコープ
Built-in all-access subscription	*****	*****	製品: Starter
test subscription 1	*****	*****	サービス
Temporary Subscription for citizen	ec538996536d4a13b3f7a7... (copy)	cde6c1f67492582a719... (copy)	API: Todo API v1

※ 開発者ポータルがセットアップされていればこの手順(=プロ開発者側の個別対応)は不要になる



カスタムコネクタの作成

市民開発者は Power Apps ポータルを使用して API 仕様を基にカスタムコネクタを作成する

指定すべき内容は Open API 仕様に
ほぼ記載されているが

特定の API バージョンを呼び出すように
ポリシーを追加する必要がある

The screenshot shows the 'Import an OpenAPI file' dialog in the Power Apps portal. It includes fields for 'Connector name' (set to 'TodoApiV1') and 'Import an OpenAPI file' (set to 'todo-api.json'). A callout points to the 'Connector name' field with the text 'コネクタ名にはバージョン番号を含めるとよい'. Another callout points to the 'Import an OpenAPI file' field with the text 'エクスポートしておいた Open API 2.0 の API 仕様ファイル'.

The screenshot shows the 'Policy details' configuration screen. It includes sections for 'Actions (4)', 'References (2)', and 'Policies (1)'. A callout points to the 'Actions (4)' section with the text 'Open API 仕様がしっかり出来上がっているとカスタマイズする必要はない'. Another callout points to the 'Set query string parameter' field with the text 'バージョン番号の指定方法'. A third callout points to the 'Operations' section with the text '対象の操作(全て指定)'.



カスタムコネクタのテスト

作成したコネクタから「接続」を作成し、動作確認を行う

接続の作成時に API を呼び出すキーが必要なため、IT 管理者に発行してもらうとよい

接続(Connection)を作成

発行してもらったキーを入力

Test operation

Selected connection *

TodoApiV1_userkey (Created at 2021-11-18T08:53:46.8838105Z)

+ New connection

Operations (4)

list-all-todo-items

Test operation

Status (200)

Headers

```
content-encoding: "gzip"
content-type: "application/json; charset=utf-8",
date: "Thu, 18 Nov 2021 09:42:07 GMT",
request-context: "appId=cid-v1:4494a9f3-76fd-4655-b614-c46762f42981",
vary: "Accept-Encoding,Origin",
"x-ms-apihub-cached-response": "true",
x-powered-by: "ASP.NET"
```

Body

```
[{"id": "969292fd-dada-487b-91c4-3ced21d8fecf", "Title": "Need implementation", "Owner": "Dummy User 1", "Status": "New"}, {"id": "4accd4bd-78dc-4260-9261-d27a0eaedcd1", "Title": "Test item", "Owner": "Test User", "Status": "In Progress"}]
```



キャンバスアプリの開発

作成したコネクタと接続を使用してキャンバスアプリから呼び出せることを確認する

Open API 仕様で定義した Operation がそのまま Power Apps の関数になっている
この例では Todo の一覧が返ってくるため、ギャラリーを使用して一覧表示している

The figure consists of three screenshots of the Microsoft Power Apps canvas editor interface, showing the steps to create a gallery control displaying a list of todo items from an API.

- Step 1: Selecting the Connector**
The first screenshot shows the "Data" screen with a modal dialog titled "Data Source Selection". A search bar contains "todo". Below it, a list includes "TodoApiV1 Sample API description" and "Todoist Todoist is an open source task management app and...". An orange callout box points to the "Todoist" item with the text "コネクタを選択" (Select connector).
- Step 2: Selecting the Connection**
The second screenshot shows the same "Data Source Selection" screen, but now the "Todoist" connector is selected. The modal now lists "TodoApiV1" and "TodoApiV1 userkey". An orange callout box points to the "TodoApiV1" item with the text "接続を選択" (Select connection).
- Step 3: Resulting Canvas App**
The third screenshot shows the completed canvas app. The "List" screen displays a gallery control containing three items, each representing a todo item. The items are:
 - Need implementation Dummy User 1 New bc9b11ad-b0ef-44f1-9dd1-
 - Need implementation Dummy User 2 Doing 2b89e5c7-66ee-4910-bb7d-
 - Need implementation Dummy User 3 Done 0023dc93-d855-49f0-9392-An orange callout box points to the bottom right of the gallery with the text "API Management のモック応答が表示されるはず" (Mock response from API Management should be displayed).

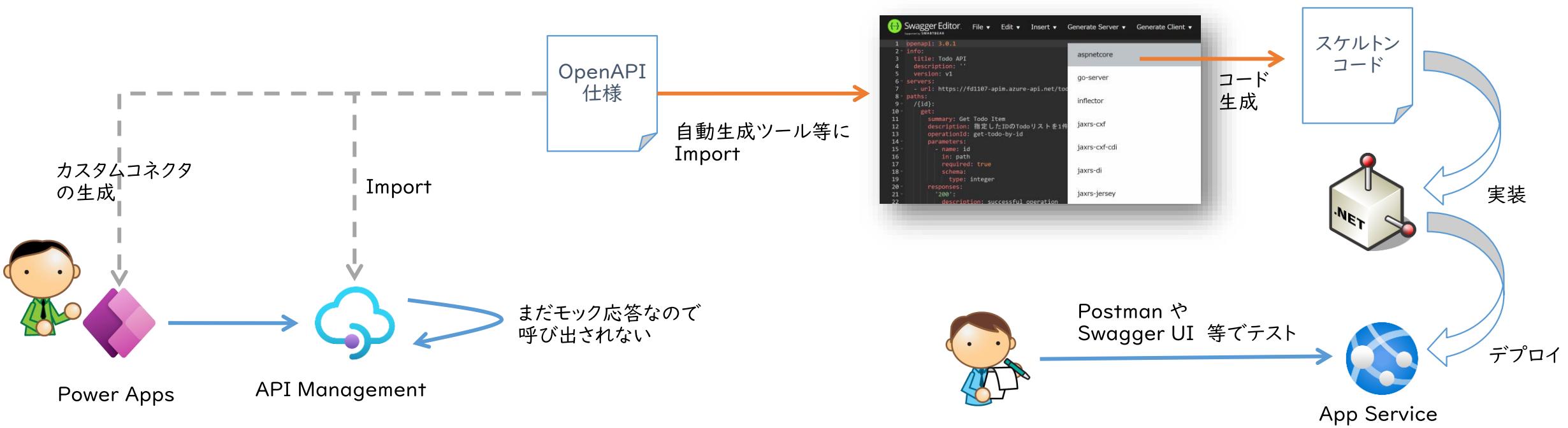


バックエンド API の実装

Open API 仕様を満たすバックエンド API を実装し、API Management から呼び出し可能な位置に配置する

ここではバックエンド API の動作環境として Azure App Service を想定

開発者間で合意した「仕様」からの差分が発生しないように、バックエンド API のスケルトンコードを自動生成すると良い (Swagger Editor 等)





[参考] ASP.NET Core API の実装(NSwag)

NSwag を使用すると Open API 仕様から ASP.NET core API のスケルトンコードを生成することができる
実際にやり取りするデータ構造や URL 等のインターフェースがズれないことが重要

実行ランタイムを選択

コントローラーを生成させる

API Management からエクスポートした Open API 仕様を貼り付け

Input: OpenAPI/Swagger Specification

Runtime: Net60

Default Variables ('foo=bar.baz=bar'), usage: \${foo}

TypeScript Client CSharp Client CSharp Controller

```

15 namespace MyNamespace
16 {
17     using System;
18     using System.Collections.Generic;
19     using System.Threading.Tasks;
20     using System.Threading.Tasks.Task;
21     using System.Threading.Tasks.Task<System.Collections.Generic.ICollection<ToDoItem>>;
22     using System.Threading.Tasks.Task<System.Collections.Generic.ICollection<ToDoItem>> ListAllTodoItemsAsync();
23     using System.Threading.Tasks.Task<System.Collections.Generic.ICollection<ToDoItem>> GetTodoItemByIdAsync(string id);
24     using System.Threading.Tasks.Task<System.Collections.Generic.ICollection<ToDoItem>> UpdateTodoItemAsync(ToDoItem body);
25     using System.Threading.Tasks.Task<System.Collections.Generic.ICollection<ToDoItem>> CreateTodoItemAsync(ToDoItem body);
26     using System.Threading.Tasks.Task<System.Collections.Generic.ICollection<ToDoItem>> DeleteTodoItemAsync(string id);
27     using Microsoft.AspNetCore.Mvc;
28     using Microsoft.AspNetCore.Mvc.Controllers;
29     using Microsoft.AspNetCore.Mvc.ModelBinding;
30     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders;
31     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result;
32     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.Result;
33     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType;
34     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType;
35     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType;
36     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType;
37     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType;
38     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
39     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
40     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
41     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
42     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
43     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
44     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
45     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
46     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
47     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
48     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
49     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
50     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
51     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
52     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
53     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
54     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
55     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
56     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
57     using Microsoft.AspNetCore.Mvc.ModelBinding.Binders.Result.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType.ResultType;
58 }

```

Generate Outputs Generate Files

コマンドラインで実行する場合

PowerShell

```
PS> nswag.exe openapi2cscontroller `
/invoke:...yaml...todo-api-spec.json `
/classname:Todo `
/namespace:FusionDev.Samples.TodoApi.Controllers `
/output:Controllers/TodoController.cs `
/UseActionResultType:true `
/UseLiquidTemplates:true `
/AspNetNamespace:"Microsoft.AspNetCore.Mvc" `
/ControllerBaseClass:"Microsoft.AspNetCore.Mvc.ControllerBase" `
/ControllerStyle:partial `
/ResponseArrayType:IEnumerable
```



[参考] バックエンド API の実装

NSwag でスケルトンコードを生成した場合には API として動作させるためにいくつか作業が必要になる

- スタートアップ コードで NSwag ミドルウェアを登録する ([NSwag と ASP.NET Core の概要](#))
- 生成されたインターフェースを実装する (ビジネスロジック実装、DBアクセス、外部システム連携など)
- 作成したクラスのインスタンスを DI フレームワーク渡すためのサービス登録をする

Open API 仕様に応じたインターフェースが生成される

C#

```
public class TodoControllerImpl : IToDoController
{
    public async Task<ToDoItem> GetTodoItemByIdAsync(string id)
    {
    }

    public Task<ICollection<ToDoItem>> ListAllTodoItemsAsync()
    {
    }

    public Task<ToDoItem> NewTodoItemAsync(ToDoItem body)
    {
    }

    public Task<ToDoItem> UpdateTodoItemAsync(ToDoItem body)
    {
    }
}
```

各 Operation を実装

```
# コントローラコード抜粋
public partial class ToDoController : Microsoft.AspNetCore.Mvc.ControllerBase
{
    private IToDoController _implementation;

    public ToDoController(IToDoController implementation)
    {
        _implementation = implementation;
    }

    # スタートアップコード抜粋
builder.Services.AddSingleton(
    typeof(aspNetapi.Controllers.IToDoController),
    new aspNetapi.Controllers.TodoControllerImpl());
}

var app = builder.Build();
```

コンストラクタで注入される
ように構成されているので

実装したクラスのインスタンスが
自動注入されるように指定



API リビジョンの更新

バックエンド API の実装が完了したら API Management がモック応答ではなく実際の API の呼び出しを行うように修正する

いきなり API 定義を書き換えるのではなく、新規リビジョンの作成 > 新しいリビジョンの定義を修正 > 新しいリビジョンをアクティブにする (make current) の流れで作業すること

The screenshot shows the Azure API Management portal interface. It displays two versions of the Todo API:

- REVISION 1 (Top):** Created Nov 16, 2021, 2:45:54 PM. Shows two revisions:
 - ID 2, Created Nov 17, 2021, 12:00:39 PM, Description: convert mock to backend api, URL: /todo;rev=2, Online: checked, Current: checked.
 - ID 1, Created Nov 16, 2021, 2:45:54 PM, Description: Echo API, URL: /todo, Online: checked, Current: checked.A red box highlights the "+ Add revision" button.
- REVISION 2 (Bottom):** Created Nov 17, 2021, 12:00:39 PM. Shows two revisions:
 - ID 2, Created Nov 17, 2021, 12:00:39 PM, Description: convert mock to backend api, URL: /todo, Online: checked, Current: checked.
 - ID 1, Created Nov 16, 2021, 2:45:54 PM, Description: Echo API, URL: /todo;rev=1, Online: checked, Current: checked.A red box highlights the "Design" tab of Revision 2.

Annotations in orange boxes explain the process:

- "リビジョン2を追加してもまだ1が Current" (Even if Revision 2 is added, 1 is still Current) points to the Revision 1 table.
- "リビジョン2の定義を修正する(1を修正すると利用中のユーザーに影響が出るため)" (Modify Revision 2's definition (Modifying 1 will affect active users)) points to the "Design" tab of Revision 2.
- "API 実装にルーティング" (Route to API implementation) points to the "Web service URL" field in Revision 2, which is highlighted with a red box.
- "リビジョン2の定義とテストが終わったら Current を切り替え" (Switch to Current after Revision 2's definition and tests are completed) points to the Revision 2 table.

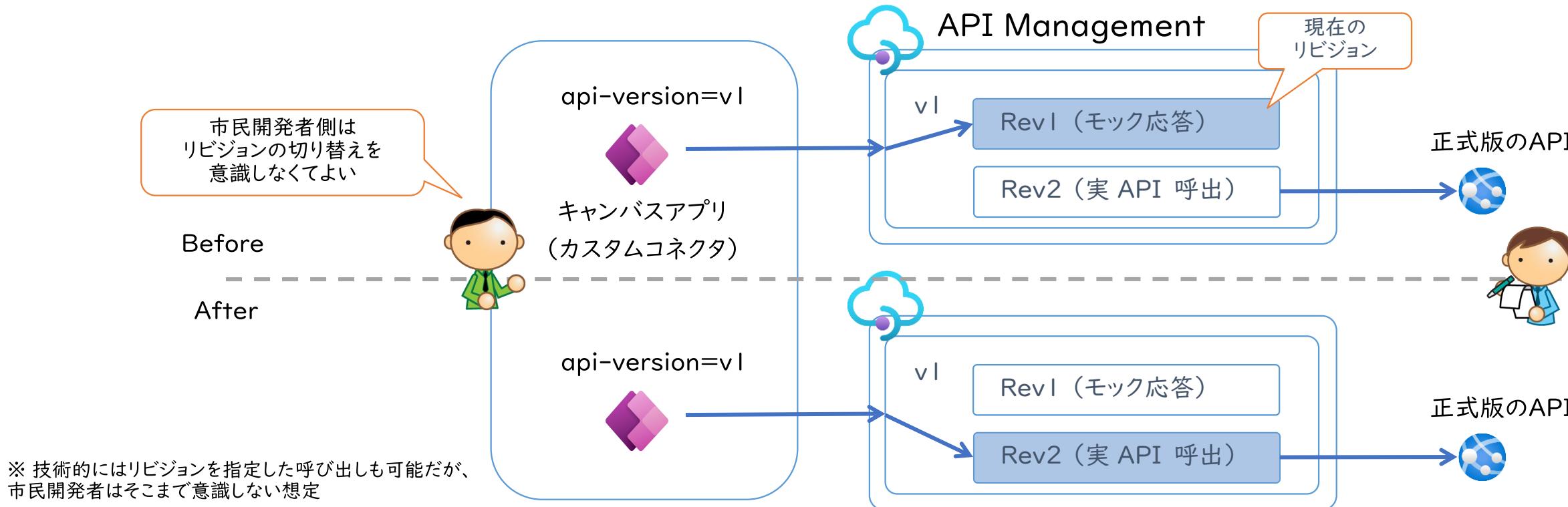


正式版 API によるテスト

リビジョンの更新が行われると、Power Apps 側も正式な API 実装を利用した開発・テストができるようになる

カスタムコネクタ内でバージョンは明記(v1)したが、リビジョンは指定していない

このため API Management が自動的に現在リビジョンにリクエストをルーティングする





ユーザーへの共有

開発が完了したアプリは他のユーザーに共有することで利用可能になる

共有されたアプリを利用するユーザーは、指定されたコネクタの「接続」が必要になる
このため各ユーザーが利用する API キーの払い出しプロセスが別途必要になる





ユーザーへの接続の共有

市民開発者側はアプリと共に「接続」も共有することができる

ユーザーはわざわざ API キーの発行・管理の手間がなくなる

全ユーザーが同じキーを使用することになるため、データや API のアクセス制御の観点で問題がないかは整理すること



市民開発者側で作成・管理するキーが埋め込まれた接続を共有する

Share TodoApiV1_developer-key

Enter names, email addresses, or user groups

+ Add everyone in my org

Shared with

Name	Email	Permission
Joni Sherman	JoniS@M365x861838.O...	Can use
Administrator M...	admin@M365x861838.o...	Owner

Cancel Save



共有された接続(=ユーザーは作らない)をアプリが利用することを許可するだけでよい

もう少しで終了します...

papp-apim-demo-1117 は、次を使用するためにアクセス許可を必要とします。続行するには、アクセス許可を付与してください。

TodoApiV1
TodoApiV1_developer-key
サインイン済み

アカウントの切り替え

許可 許可しない

API キーを共有する際の注意

API Management のキーをコネクタに埋め込んでしまわないよう注意すること

カスタムコネクタ作成時にHTTP ヘッダーなどでキーを記述してしまうことは技術的に可能だがここで指定した値は暗号化されず、コネクタを開発している「環境」にアクセス可能なユーザーであれば参照可能なため漏洩リスクがある
アプリやコネクタを他の環境へエクスポートする際にも漏洩する可能性がある

The screenshot shows the Azure API Management Policy Editor interface. A policy is being defined with the following sections:

- アクション (5)**: Describes actions like list-all-todo-items, new-todo-item, update-todo-item, get-todo-item-by-id, and get-todo-items.
- 参照 (4)**: Lists references such as ToDoItem, ArrayOfToDoItem, ClaimDto, and HelloDataTrans.
- ポリシー (2)**: Describes policies for existing headers, with one entry for 'API Key' marked with a warning icon.

The main configuration area includes:

- 名前 ***: API Key
- テンプレート * 詳細情報**: Set HTTP header
- Operations**: A dropdown containing list-all-todo-items, new-todo-item, update-todo-item, get-todo-item-by-id, and get-todo-items.
- Header name ***: Ocp-Apim-Subscription-Key
- Header value ***: APIを呼び出すためのキー
- Action if header exists**: override
- Run policy on ***: Request

← セキュリティ

コード (プレビュー) →

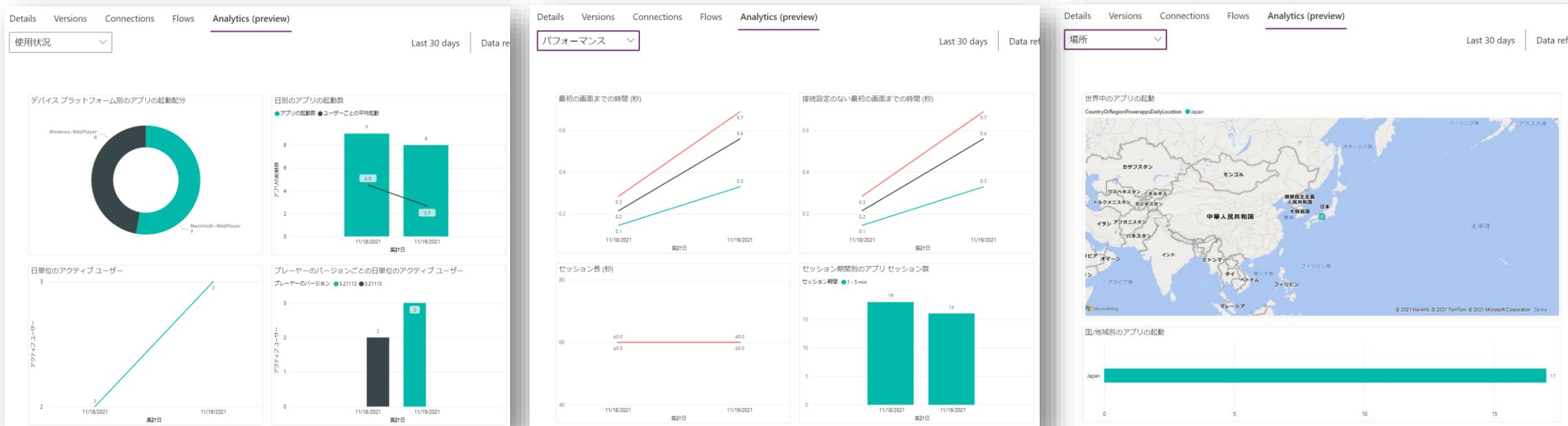


アプリ利用状況の監視

市民開発者は自身が作成したアプリの利用状況を確認することができる

有用性が高く広く使われるようになったアプリは市民開発者個人の手から離れ、組織として予算をつけて正式な保守・運用のプロセスの下で管理すべき

作っては見たものの利用頻度が低いアプリであれば無理して保守を継続する必要もなく、不要になれば破棄してしまうことで市民開発者への負担も下げられる





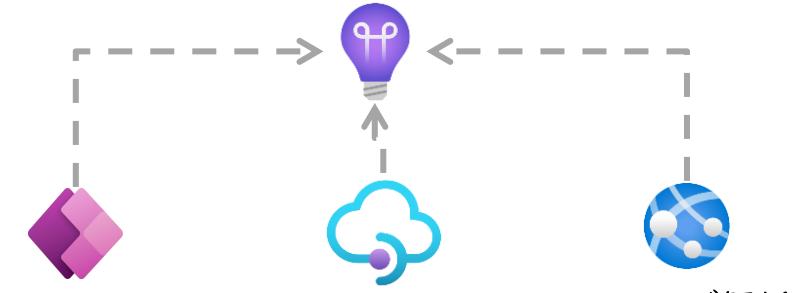
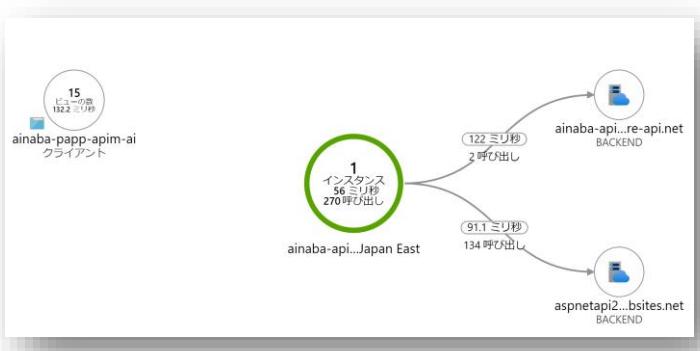
API 利用状況の監視

IT管理者側は API Management や バックエンド API の実行状況を Application Insights で監視するとよい

Power App の利用状況も監視可能(分散トレースには対応していない)

多数のアプリから高頻度に呼び出される API はそれだけ重要性も高い

アプリケーションマップ⁹



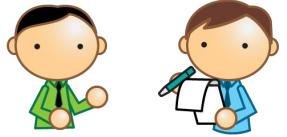
```
1 pageViews  
2 | summarize count() by client_OS, client_Browser, client_City
```

結果 グラフ | 列 ▼ | 時刻の表示 (UTC+00:00) ▼ | 列のグループ化

完了. 過去 3 日からの結果を表示しています。

client_OS	client_Browser	client_City	count_
Mac OS X 10.15	Edg 95.0	Saitama	2
Mac OS X 10.15	Edg 95.0	Singapore	4
Windows 10	Edg 95.0	Shinjuku	10

日々のアプリ保守と API 保守



アプリが日々利用されると様々な問題点や改善点がでてくるので、継続的な更新が必要になってくる

Power Apps 側はどのバージョンをユーザーに利用してもらうか「公開」操作で制御する
API 側は前述のリビジョン機能を利用して設定を分離、コネクタからの呼び出しを制御する

Details	Versions	Connections	Flows	Analytics (preview)
① It's only possible to restore app versions that were created in the last six months. Learn more				
Version	Modified	Modified by	Power Apps release	Published
Version 10	... 2021/11/19 19:1...	Administrator M...	3.21112.22	
Version 9	... 2021/11/19 19:1...	Administrator M...	3.21112.22	
Version 8	... 2021/11/19 14:1...	Administrator M...	3.21112.22	Live
Version 7	... 2021/11/19 14:0...	Administrator M...	3.21112.22	
Version 6	... 2021/11/19 14:0...	Administrator M...	3.21112.22	
Version 5	... 2021/11/18 20:4...	Administrator M...	3.21112.22	
Version 4	... 2021/11/18 19:5...	Administrator M...	3.21112.22	
Version 3	... 2021/11/18 19:3...	Administrator M...	3.21112.22	
Version 2	... 2021/11/18 19:2...	Administrator M...	3.21112.22	
Version 1	... 2021/11/18 19:0...	Administrator M...	3.21112.22	

REVISION 1 CREATED Nov 16, 2021, 2:45:54 PM

Design Settings Test Revisions Change log

Revisions

ID	CREATED	DESCRIPTION	URL	ONLINE	CURRENT
2	Nov 17, 2021, 12:00:39 PM	convert mock to backend api	/todo;rev=2	✓	
1	Nov 16, 2021, 2:45:54 PM		/todo	✓	✓

+ Add revision

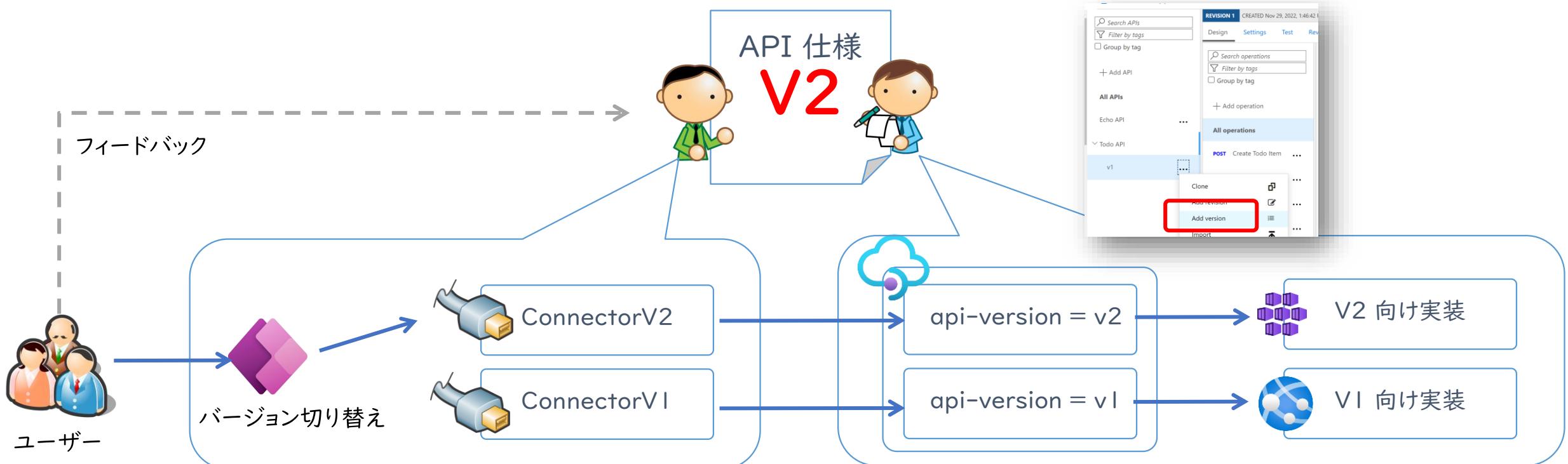


破壊的な変更

API 仕様が変わるような大きな変更が必要な場合は、更新ではなく新規開発と並行稼働を行う

Power App アプリ側の改修も必要となる可能性があるため、特に複数のアプリから共有される API のバージョンアップは並行稼働が極めて重要になる

API の互換性が無くカスタムコネクタが使い回せなくなるため、カスタムコネクタも新規バージョンとして作り直す（コネクタ名にバージョン番号を含めると良い）



補足：コネクタは誰が作る？

コネクタと API のバージョンを一致させる運用ならば、プロ開発者側で API とコネクタの両方を作ってしまっても良い

新しいバージョンの API 公開とともにカスタムコネクタも作成・テスト・共有してしまえば、市民開発者は API の存在など意識せずに用意されたコネクタを使うだけでよい

特に後述の Azure AD 認証を使う場合などは設定が非常に難解になるため、コネクタまで作ってしまう方が現実的な場合もある

役割分担はともかくとして市民開発者もカスタムコネクタ開発方法や は知っておいて損はない

社内でプロ開発者が管理していない外部の API などを利用したい場合は、結局自分でカスタムコネクタを作らざるを得ない

Power Apps 以外の API クライアントを使用したいケースも鑑みると、コネクタは作らないにしても API とその利用方法は理解できている方が応用範囲は広がる

Module 4

開発者ポータルのセットアップ

本モジュールの目的

内容

市民開発者のセルフサービスを実現するために開発者ポータルを利用可能にする
開発者ポータルを準備しておくことで API の利用促進と API 管理の負担を軽減できる

前提

API Management や Web API 開発の基礎知識



API Management 開発者ポータルの有効化

開発者ポータルは既定で有効になっていないため、まず公開する必要がある

これまで Power Apps などから呼び出していた API Management のゲートウェイ部分とは別に、「開発者ポータル」と呼ばれる独立した Web アプリケーションが用意される

ゲートウェイ URL: <https://api-management-name.azure-api.net>

開発者ポータル : <https://api-management-name.developer.azure-api.net>

ホーム > リソース グループ > papp-apim-demo-rg > ainaba-apim-1124

ainaba-apim-1124 | ポータルの概要

概要 検索 (Ctrl + /) 開発者ポータル 非推奨の開発者ポータル

概要 リビジョン

開発者ポータルは自動的に生成される、完全にカスタマイズ可能な Web サイトであり、お客様の API のドキュメントがそこには記載されます。API の使用者はこれらで API を検出し、使用方法を学習し、アクセスを要求できます。[開発者ポータルの詳細を調べるか、チュートリアルを使ってカスタマイズします。](#)

従来の開発者ポータルは非推奨になったため、セキュリティ更新プログラムのみを受け取ります。2023 年 10 月に廃止されすべての API Management サービスから削除されるまでは、引き続き使用することができます。新しい開発者ポータルへの移行方法をご確認ください。

開発者ポータルを公開すると、変更とカスタマイズを閲覧者が利用できるようになります。この操作は、非推奨のポータルには影響せず、新しい開発者ポータルにのみ適用されます。

開発者ポータルがまだ公開されていません

開発者ポータルを有効にする

CORS を有効にする

API を有効にする

Azure AD を有効にする

ID プロバイダーを手動で構成します (例: Azure AD B2C)

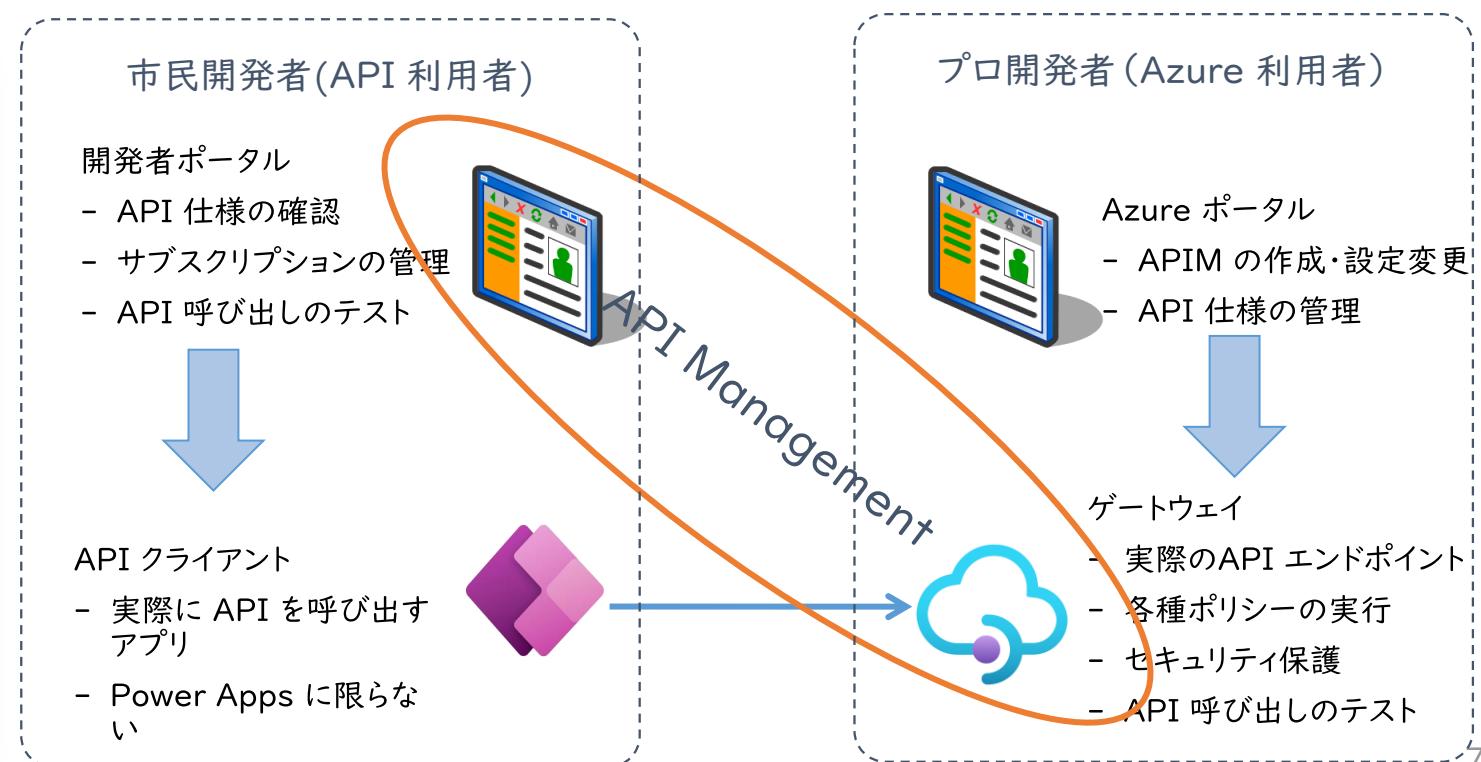
API Management サービス

概要 アクティビティログ アクセス制御 (IAM) タグ 問題の診断と解決 イント (プレビュー)

Settings プロパティ

APIs API 製品 サブスクリプション 名前付きの値 Backends API タグ Power Platform

開発者ポータル ポータルの概要 ユーザー グループ コード 委任 OAuth 2.0 + OpenID Connect 問題 (非推奨)



開発者ポータルからの API テストを有効にする



The screenshot shows the 'Developer Portal' section of the Azure API Management service settings. It includes a search bar, navigation links for 'Developer Portal' and 'Non-privileged Developer Portal', and tabs for 'Overview' and 'Regions'. A note states that the developer portal is automatically generated and can be customized. Below this, there's a 'Portal Public' section with a 'Public' button, a 'Previous Version' section showing a revision from December 6, 2021, and a 'CORS' section with a warning about the origin https://ainaba-apim-1124.developer.azure-api.net not having CORS configured. It also mentions Azure Active Directory integration and Azure AD enablement.

開発者ポータルに対して CORS を有効化することで、API 呼び出しのテストが可能になる

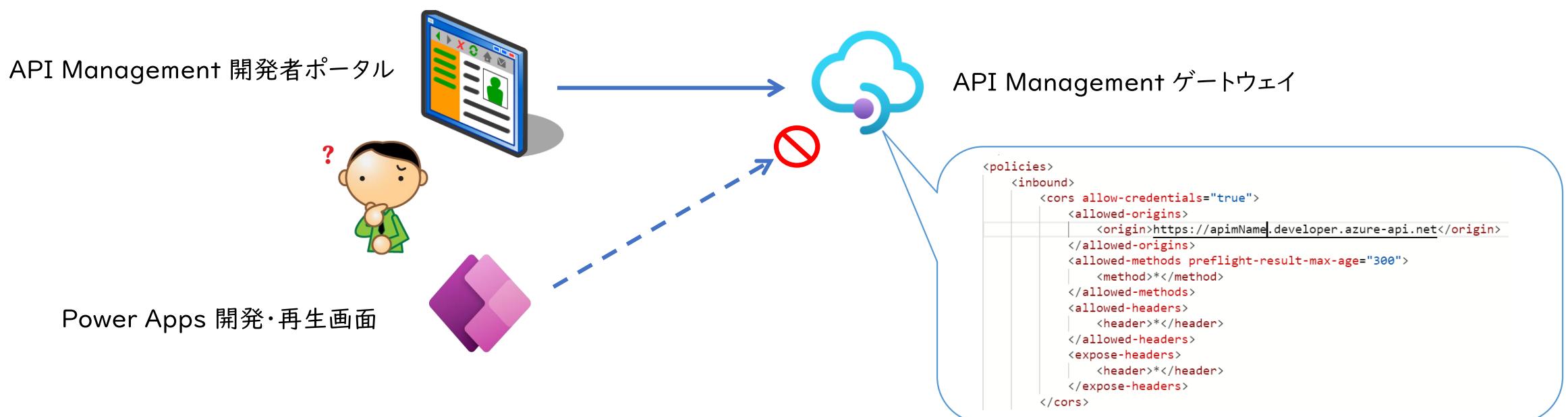
開発者ポータルが クロスドメインポリシーに登録されるようになる

この設定した直後は Power Apps 側からの呼び出しが出来なくなることに注意（回避策は後述）

The screenshot shows the 'API' section of the Azure API Management service settings. It includes a search bar, navigation links for 'Overview' and 'Developer Portal', and tabs for 'Design' and 'Settings'. Under 'Inbound processing', there's a note about modifying requests before sending them to the backend service. In the 'Policies' section, a 'cors' policy is listed under 'Frontend'.

開発者ポータルと CORS ポリシー

CORS ポリシーは API Management の開発者ポータルだけを許可しているため、PowerApps からは呼び出せない
つまり Power Apps 開発や利用中に発生する CORS プリフライトが成功するように明示的に許可してやる必要がある



Power Apps 向け CORS の登録

API を呼び出す各画面の Origin を CORS ポリシーに登録すると動作するようになる

コネクタ開発時のテスト実行

<https://flow.microsoft.com>

アプリ開発画面のプレビュー

<https://make.powerapps.com>

<https://jp.create.powerapps.com>

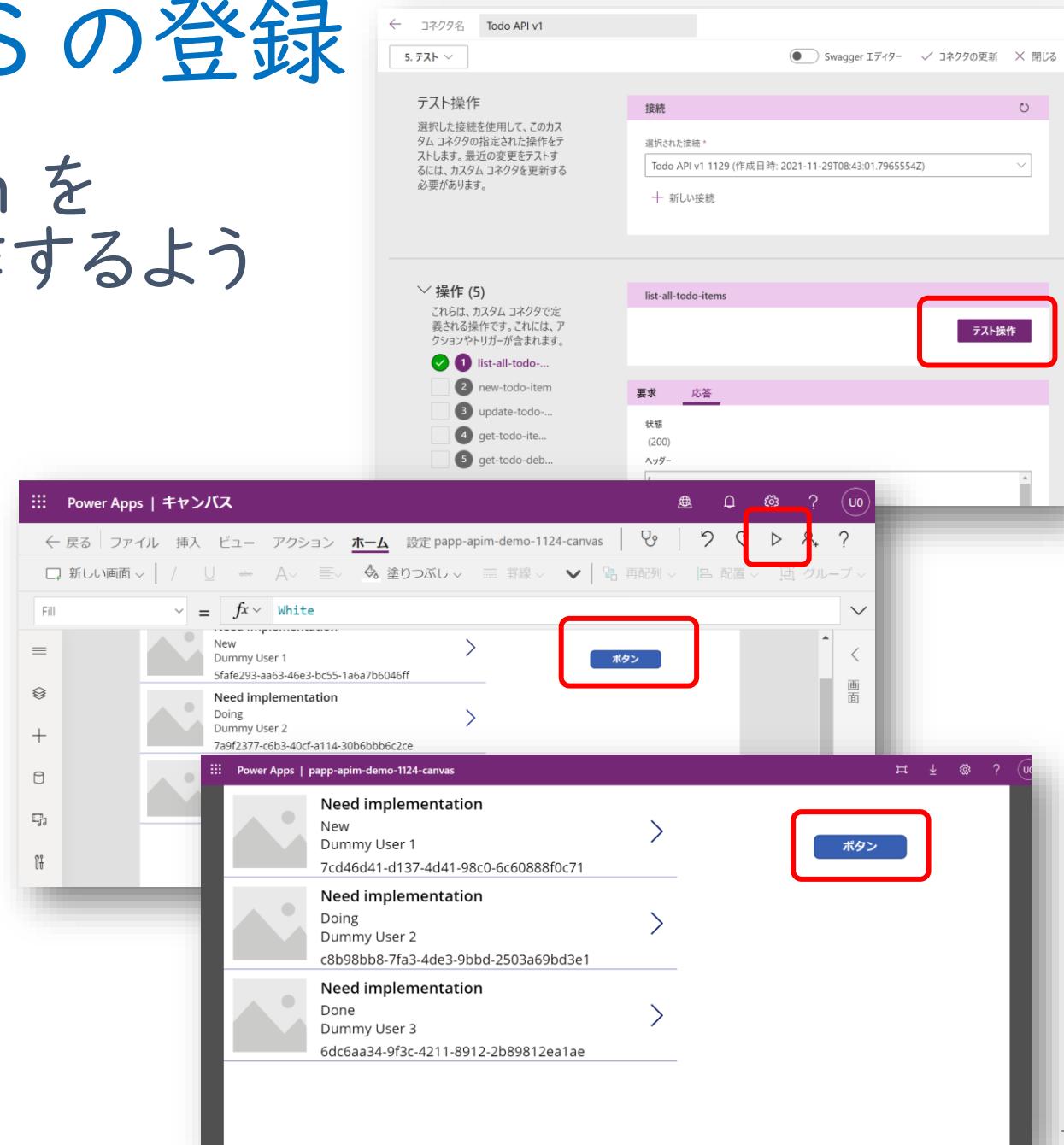
<https://authoring.jp>

<https://il101.gateway.prod.island.powerapps.com>

発行後アプリの再生画面

<https://apps.powerapps.com>

```
<inbound>
  <cors allow-credentials="true">
    <allowed-origins>
      <origin>https://ainaba-apim-1124.developer.azure-api.net</origin>
      <origin>https://flow.microsoft.com</origin>
      <origin>https://make.powerapps.com</origin>
      <origin>https://jp.create.powerapps.com</origin>
      <origin>https://apps.powerapps.com</origin>
    </allowed-origins>
  </cors>
</inbound>
```



補足：登録が必要な Origin の確認

The screenshot shows the Microsoft Edge developer tools Network tab. A preflight request for an API invoke call is selected. The Headers section shows the following key headers:

```
authority: japan.api.powerapps.com
method: OPTIONS
path: /api/invoke
scheme: https
accept: */*
accept-encoding: gzip, deflate, br
accept-language: ja
access-control-request-headers: authorization,cache-control,x-ms-diagnostics
access-control-request-method: POST
cache-control: no-cache
origin: https://apps.powerapps.com
pragma: no-cache
```

本資料の作成時点で公式に登録が必要な Origin が記載されたドキュメントが存在しない

このため現時点ではブラウザの開発者ツールを使って実際に発生している preflight 要求をキャプチャして確認するのが手っ取り早い

例えば jp.create.powerapps.com のような環境固有の値が含まれるケースは1つ1つ対応していく必要がある

The screenshot shows the Power Platform Management Center. Under the Environment section, it displays the following details:

環境 URL	状態
origin: https://apps.powerapps.com	Ready

Under the Details section, it shows:

地域	更新の頻度
日本	高頻度

Under the Policies section, it shows:

種類	セキュリティ グループ
Developer	割り当てられていません

Under the Groups section, it shows:

組織 ID
d3ee7cab-f0c0-4e0d-9f3a-0a2a2a2a2a2a

補足：セキュリティと再利用性のトレードオフ

API の利用促進という観点では、呼び出し元となるアプリを Power Apps に限定しない方が良いという考え方もある

Power Apps 以外のローコードツール、プロ開発者が作る SPA アプリ、外部 SaaS への API 提供など様々な呼び出し元が考えられる

新しいアプリやツールが増えるたびに前述の CORS 設定を書き換えていくのは煩雑なため、ワイルドカードドメインを設定して任意のクライアントを許可することもできる

セキュリティリスクが高まる面は否めないため、API キーの管理やユーザー認証はしっかりと設定すること

任意のドメインを許可

```
<inbound>
  <cors allow-credentials="false">
    <allowed-origins>
      <origin>*</origin>
    </allowed-origins>
```

部分的なワイルドカードドメインは指定できない

開発者ポータルと Power Apps だけを許可

```
<inbound>
  <cors allow-credentials="true">
    <allowed-origins>
      <origin>https://ainaba-apim-1124.developer.azure-api.net</origin>
      <origin>https://flow.microsoft.com</origin>
      <origin>https://make.powerapps.com</origin>
      <origin>https://jp.create.powerapps.com</origin>
      <origin>https://apps.powerapps.com</origin>
    </allowed-origins>
```



Azure AD 認証の有効化

開発者ポータルを有効化後、Azure AD 認証も有効化するとよい

市民開発者は Power Apps アプリの開発時に Azure AD ユーザーで認証されているはず
API Management の開発者ポータルも同じ ID でシングルサインオンできた方が便利
加えてユーザー名とパスワードによる認証、および、匿名ユーザーアクセスも無効化する

ainaba-apim-1124 | ポータルの概要

概要 リビジョン

開発者ポータルは自動的に生成される、完全にカスタマイズ可能な Web サイトを学習し、アクセスを要求できます。開発者ポータルの詳細を調べるか、コードを表示する

従来の開発者ポータルは非推奨になったため、セキュリティ更新プログラムのみ適用することができます。新しい開発者ポータルへの移行方法をご確認ください。

開発者ポータルの公開

ポータルを公開すると、変更とカスタマイズを閲覧者が利用できるようになります。

直前の発行

リビジョン 202112060146 は 2021/12/6 10:46:20 に作成され、2021/12/6 10:46:20 に公開されました。

CORS を有効にする

クロスオリジンリソース共有は、Web ページにあるリソースを別のドメインから、つまりページで対話型コントロールを使用するには CORS が必要であり、カスタムドメインのドメインビューに移動してください。詳細情報

https://ainaba-apim-1124.developer.azure-api.net オリジンに対して CORS を有効にする

手動でグローバル レベルに適用

Azure Active Directory を使用したユーザーのサインインを有効にする

Azure Active Directory フラッシュコンソールを監視的にプロビジョニングし、ご使用ください

Azure AD を有効にする

ID プロバイダーを手動で構成します (例: Azure B2C)

ainaba-apim-1124 | ユーザー

開発者ポータル

ポータルの概要

プロバイダーの種類

Azure Active Directory

ユーザー名とパスワード

削除

ainaba-apim-1124 | ユーザー

開発者ポータル

ポータルの概要

プロバイダーの種類

Azure Active Directory

API Management サービス

ユーザー サインアップの利用規約

サインアップ ページの利用規約の表示

同意が必要

匿名ユーザー

匿名ユーザーをサインイン ページにリダイレクト



Azure AD 認証の有効化

開発者が開発者ポータルに Azure AD 認証でサインインできるようになるためには管理者の同意が必要

この設定後に一般的なユーザーが開発者ポータルにアクセスすると、普段しようしている Azure AD ユーザー アカウントでサインインできるようになる

初回利用時にサインアップを行うと、API Management 側でも利用ユーザーとして登録され把握できるようになる

ainaba-apim-1124 | API のアクセス許可

検索 (Ctrl+ /) < 最新の情報に更新 フィードバックがある場合

概要 クイック スタート 統合アシスタント

管理 ブランド 認証 証明書とシークレット トークン構成 API のアクセス許可 API の公開 アプリ ロール 所有者 ロールと管理者 | プレビュー マニフェスト サポート + トラブルシューティング トランザクション 新しいサポート リクエスト

構成されたアクセス許可

API / アクセス許可の名前 種類 説明 管理者の同意が必要 状態

Azure Active Directory Graph (2)

API / アクセス許可の名前	種類	説明	管理者の同意が必要	状態
Directory.Read.All	アプリケーション	Read directory data	はい	ainabaeslz に付与されて...
User.Read	委任済み	サインインとユーザー プロファイルの読み取り	いいえ	...

Microsoft Graph (2)

API / アクセス許可の名前	種類	説明	管理者の同意が必要	状態
Directory.Read.All	アプリケーション	Read directory data	はい	ainabaeslz に付与されて...
User.Read	委任済み	Sign in and read user profile	いいえ	...

アクセス許可とユーザーの同意を表示および管理するために、[エンタープライズ アプリケーション](#)をお試しください。

ainaba-apim-1124 | アクセス許可

最新の情報に更新 アクセス許可の確認 フィードバックがある場合

概要 デプロイ計画 管理

プロバイダー 所有者 ロールと管理者 (プレビュー) ユーザーとグループ シングル サインオン プロビジョニング アプリケーション プロキシ セルフサービス カスタム セキュリティ属性 (プレビュー)

管理者はこのナビゲート内のすべてのユーザーに代わって同意を付与できます。これにより、エンド ユーザーはアプリケーションを直接起動して、データを操作できるようになります。

ainabaeslz に管理者の同意を与える

管理者の同意 ユーザーの同意

アクセス許可の検索

API 名	権限	種類	付与方法	許可元
Microsoft Graph	Sign in and read user profile	Delegated	管理者の同意	1 名の管理者
Windows Azure Active Directory	Sign in and read user profile	Delegated	管理者の同意	1 名の管理者

※ この操作には Azure AD の管理者権限が必要になる



市民開発者向けの製品を発行する

API Management に登録した API を利用してもらうためには、以下の手続きを行う

利用してもらうための製品（1つ以上の API を束ねたもの）を作成し、API を追加
利用者が所属するグループにアクセス権を付与

The screenshot shows the 'API for Power Platform' product page. The left sidebar has a 'Products' section with 'API for Power Platform' selected. The main area shows basic product information: name 'API for Power Platform', protection 'Requires subscription', status 'Published', and 1 subscription. Below this, under 'API', is a table with 'Todo API' listed. At the bottom, under 'Administrators', 'Developers' is selected. The 'Products' tab is highlighted in red.

既定では認証済みユーザーは全て
Developer グループに所属する
API の提供範囲に要件がなければ
まずは誰でも試せるように Developers
グループに対して製品を公開するとよい

API 利用者の管理

開発ポータルにサインアップした市民開発者の利用者情報は Azure Portal で管理することが出来る
ユーザーのロックや削除、パスワードの再発行、サブスクリプションの削除など

The screenshot displays three main windows from the Azure Portal:

- User Management Window:** Shows a list of users under the "fd1126b-apim" service. The "User" section is selected in the sidebar. A search bar at the top is empty. Below it, there are buttons for "追加" (Add), "招待" (Invite), and "列" (Columns). The list includes "Administrator", "Ayumu Inaba", and "ESLZ admin".
- User Details Window:** Shows detailed information for the user "Ayumu Inaba". The "Basic" tab is selected. It shows the following details:
 - 完全名: Ayumu Inaba
 - 状態: アクティブ
 - 電子メール: ayumu.inaba@live.com
 - ユーザー: Basic
 - グループ: Developers
- Subscription Management Window:** Shows a list of subscriptions for the user "Ayumu Inaba". The "サブスクリプション" section is selected in the sidebar. A search bar at the top is empty. Below it, there are buttons for "サブスクリプションの追加" (Add subscription) and "最新の情報に更新" (Update latest information). The table lists one subscription:

表示名	主キー	2 次キー	スコープ	状態
subsc1	[REDACTED]	[REDACTED]	製品: Starter	アクティブ

補足：2つの“ユーザー”メニュー

(ローカライゼーションの問題だが) Azure Portal では「ユーザー」という名前のメニューが 2 つ表示される

1 つは API の利用ユーザーを管理するための画面

もう1つはユーザーの認証方式を設定する画面

目的とする機能が異なるため混乱しないように気を付ける

The screenshot displays two side-by-side views of the Azure Portal's 'User' management interface for the service 'fd1126b-apim'.

Left View (User Management): This view shows the 'User' management screen. The left sidebar includes 'Power Platform', 'Portal Overview', 'Portal Settings' (which is highlighted with a red box), 'User' (which is also highlighted with a red box), 'Group', and 'User'. The main area lists users with columns for 'Name' and 'Email'. One user, 'Administrator', has the email 'fd1126b@fd1126b.local'. Another user, 'Ayumu Inaba', has the email 'ayumu.inaba@live.com'. A third user, 'ESLZ admin', has the email 'eslzadmin@ainabaezl.onmicrosoft.com'.

Right View (ID View): This view shows the 'ID View' screen for managing developer portal user authentication methods. The left sidebar includes 'Power Platform', 'Portal Overview', 'Portal Settings', 'User' (highlighted with a red box), 'Group', and 'Assignment'. The main area lists authentication providers: 'Azure Active Directory' and 'User Name and Password'. A note at the top states: 'ID ビューでは、開発者ポータルのユーザーの認証方法を管理できます。既定では、'.

補足：その他の Tips

要求のスロットリング

市民開発者の増加および API 活用の促進に応じて呼び出し回数が増えると、バックエンドシステム側に過剰な負荷がかかり、予期せぬ障害を引き起こす可能性がある

製品や API の単位で quota や rate-limit ポリシーを付与して、システムとして許容可能な程度に収まるようにゲートウェイ側で要求のスロットリングを行うとよい

[Azure API Management を使用した高度な要求スロットル | Microsoft Docs](#)

[Azure API Management のアクセス制限ポリシー | Microsoft Docs](#)

開発者ポータルのカスタマイズ

既定の状態の開発者ポータルは汎用的に作られているため、一部機能のカスタマイズが必要になる場合があるため、必要時応じてカスタマイズするとよい

ex) サインアップ機能を使用しないためナビゲーションリンクを除去するなど

[チュートリアル - 開発者ポータルへのアクセスとそのカスタマイズ - Azure API Management | Microsoft Docs](#)

[Azure API Management の開発者ポータルの概要 - Azure API Management | Microsoft Docs](#)

まとめ

Azure AI と Power Apps を活用、 市民開発とプロ開発・業務のプロが連携し 線路設備の異常検知ソリューションを東京メトロが開発

■課題

- 人が地道に歩き、視認して、目にした状況をメモ帳に書きつける。この作業が線路点検の基本。しかし少子高齢化による労働力不足という状況に直面し、生産性の改善が大きな課題となっている
- 労働力不足の現状に加えてコスト面を考えても、開発環境をスリム化しながらより良いものを目指さなければならない
- 東京メトロの全路線には膨大な数の締結装置があり、現地に赴いて全路線の点検を終えるには、通常約 1 年かかる

■選定ポイント

- IT 部門の担当者に、Microsoft Azure（以下、Azure）の AI サービスである Azure Cognitive Services と Microsoft Power Apps（以下、Power Apps）を利用するのが最適ではないかと提案された

■効果

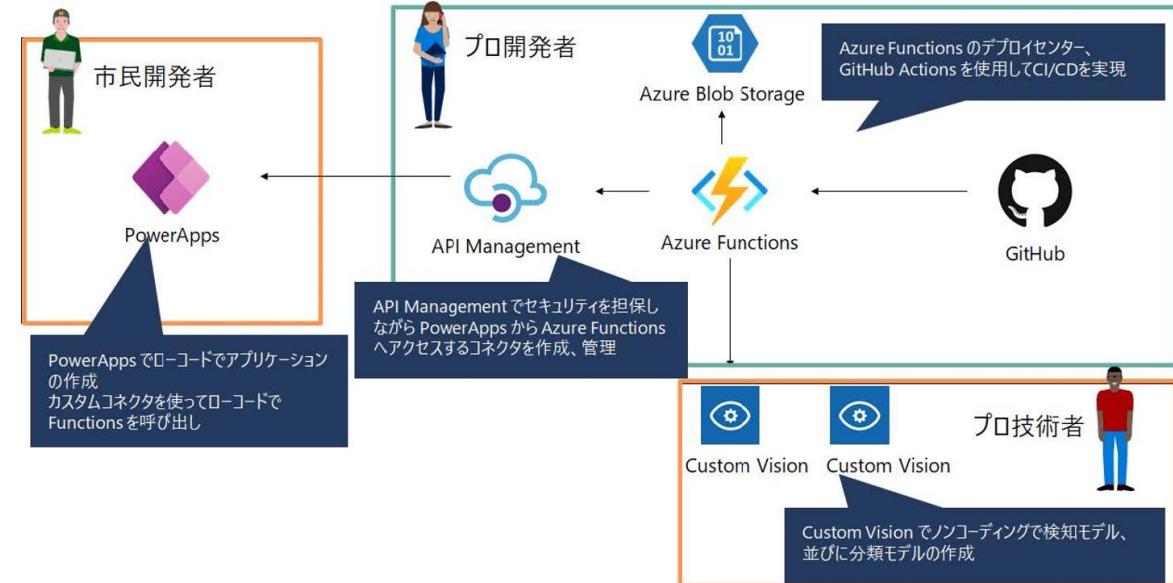
- 自分たちが主役となって AI を創り出す“AIの民主化”という言葉を実感した
- Custom Vision を使いながら、90 %を大きく超える精度の検出・分類モデルを作り上げた
- 今後の点検業務の高品質化・効率化に加えて、新技術の導入によってこれまで頼ってきたベテランの“勘所”が正しかったことを確認でき、その技を解明して次代につないでいくためのツールにもなる

■今後の展望

- 蓄積されるデータを機械学習で活用し、BI ツールで可視化することで、予防保全や点検の判断に活かすことが最終的に目指す地点となる

[<事例へのリンク>](#)

■市民開発者とプロ開発者、プロ技術者が連携



Power Apps に触れるのは初めてでしたが、Microsoft PowerPoint のスライド作成と似た感覚で開発ができ、目的通りのアウトプットを出せるアプリが出来上がったことにとても感動しました。実用化についてはまだこれからの課題ですね。改良しなければならない部分があるし、精度ももっと高めていきたい。ただ、今回の取り組みでそれらを実現できるという手応えは十分に得られました。

東京地下鉄株式会社
鉄道本部 工務部 軌道課 主任
工藤 浩之 氏

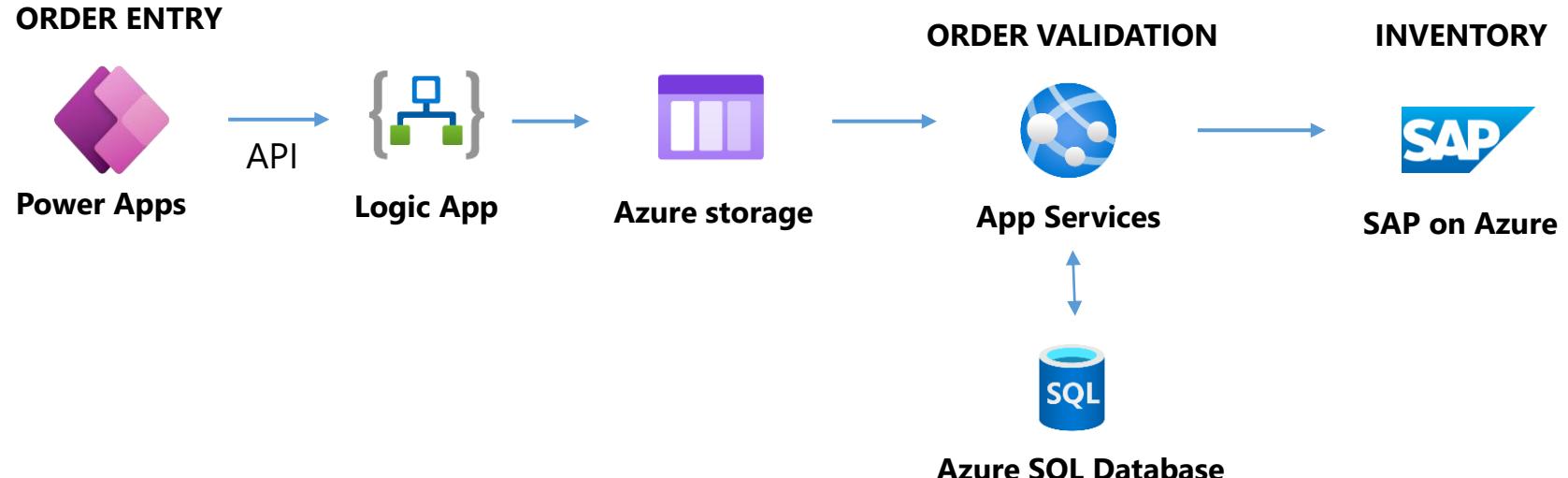
注文処理の自動化

ローコードで UI を開発し Azure で既存システムと接続箇所を開発



“オープンソースの選択肢も検討しましたが、これらのプラットフォームは常に変化し続けています。Power Platformのローコードのシンプルさと、Azureとの統合性が気に入っています。現在では、当社のすべてのERPアーリストに好まれるプラットフォームとなっています。”

Tommy Ammons
Mobile Computing Manager,
Coca-Cola UNITED



Situation

Coca-Cola UNITEDは、米国で3番目に大きいコカ・コーラ製品のボトラーです。同社は、顧客からのオンデマンドの出荷要求（または「強制出荷」）をより迅速に処理する方法を求めていました。以前は、アカウント担当者が手作業で注文を出し、在庫を確認しなければなりませんでしたが、これはしばしばエラーや出荷の遅れにつながりました。

Solution

Power AppsとAzureサービスを使って構築された新しいソリューションは、強制的な出荷プロセスを自動化します。現場の担当者は、注文の詳細を記したファイルをモバイルアプリに直接入力します。このデータはAzure App ServicesのWebJobに送られ、SAP on Azureの在庫データと照らし合わせてオーダーを検証し、さらにローカルのSQLデータベースをオーダーで更新します。

Impact

自動化されたソリューションにより、注文処理は数時間から数秒になりました。現在では、10倍の数の強制出荷オーダーを処理しており、そのすべてが、より良いトラッキング、より少ないエラー、より早い納期、より高い顧客満足度を実現しています。市場投入までの時間が短縮されたことで、売上も増加しました。

まとめ

Power Apps によるデータ活用を推進する上では独自の API を呼び出すカスタムコネクタ開発が重要になる

既成の SaaS およびコネクタで出来ることも十分に幅広いが、社内 IT などクローズド環境のデータ活用にはカスタム API が必要になる

API の乱立を避けるために Azure API Management を利用してオフィシャルな API プラットフォームを提供するとよい

カスタム API を一元管理し、バージョニング等のライフサイクルを制御できるようにする
開発者ポータルを使用してプロ開発者／市民開発者を問わずに API が使い易くなる

Power Apps / Azure のどちらも Microsoft 365 と同じ Azure AD を認証基盤とするメリットが活用できる

データを扱う上ではセキュリティは極めて重要だが、利便性や生産性とのバランスが重要
Azure AD を活用することで負荷のかからない認証・アクセス制御が構成可能

Next Step

オンプレミスの既存資産の再利用と Azure AD 認証に関しては
Appendix を参照

Power Apps / API Management / バックエンド API 各々の DevOps は別ワークショップにて紹介(計画中)

市民開発者の参画



データ活用と
業務改善

Github / Azure DevOps / Visual Studio

Power
Apps

Azure
API Management

Microsoft 365

Azure
Compute & Data

ON-PREMISE
API & DATA



プロ開発者の
生産性向上

Azure Active Directory

Appendix A

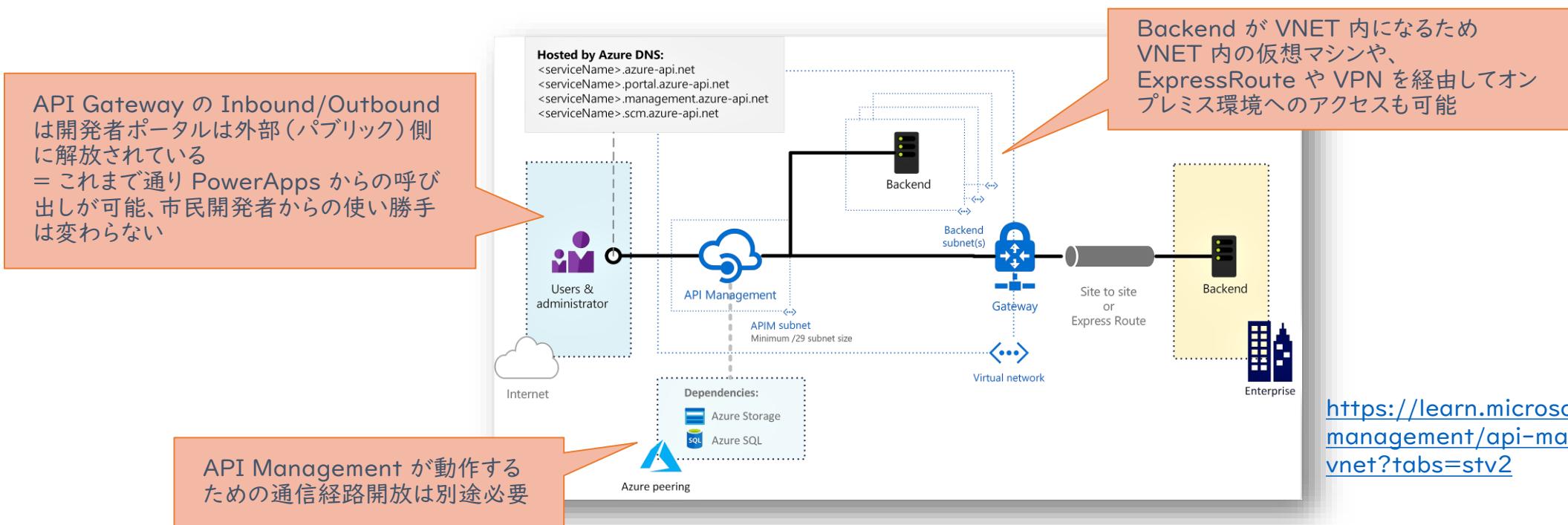
オンプレミス資源の利活用

オンプレミス データの利活用

アクセスしたいデータがオンプレミス データセンター や Azure VNET 内に存在する場合は、通信経路の確保が必要

これまで Power Apps + API Management + App Service であったため、Azure 内部とはいえパブリック ネットワーク経路での接続が行われていた

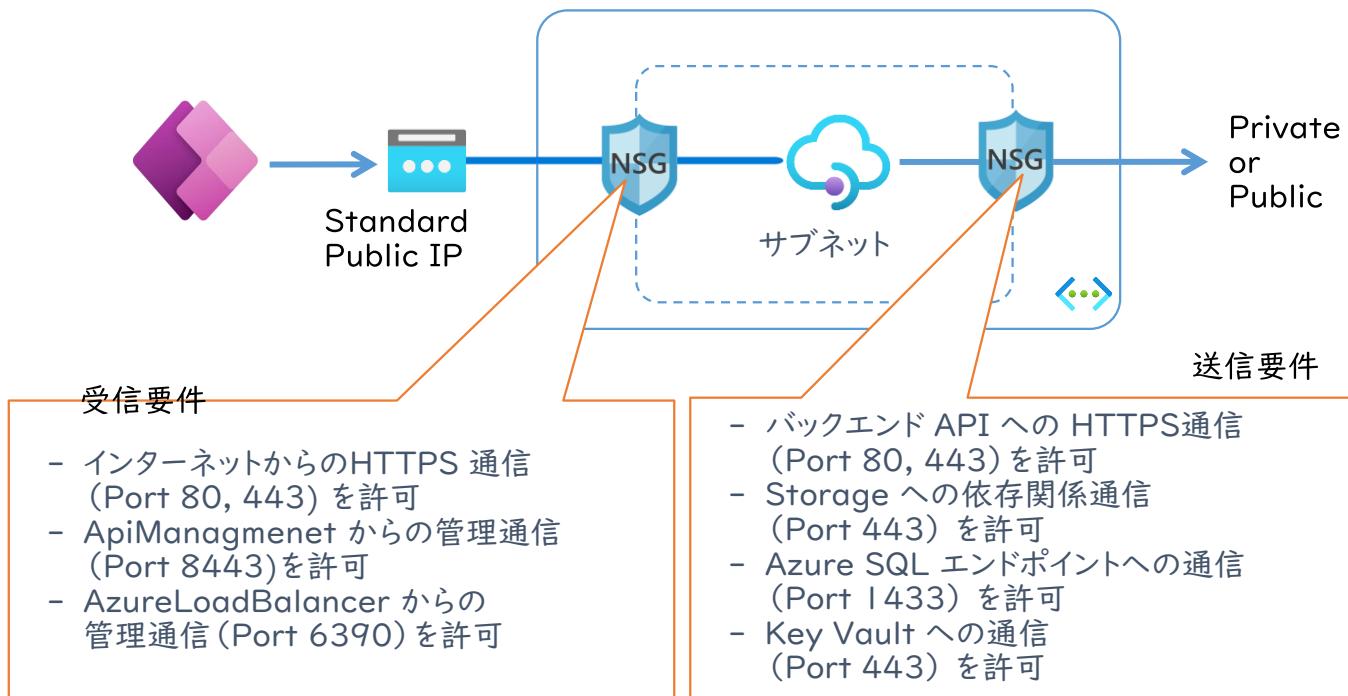
通信経路をプライベート ネットワークに引き込むためには API Management の外部ネットワークモードを利用するとよい



API Management 外部ネットワークモード

VNET に接続することで API Management に対する送受信を利用者自身のポリシーで制御できるようになる

裏を返せば利用者は API Management が正常に動作するための通信要件を満たす責務がある
VNET に接続していなかったときは意識しない各種管理系の通信が阻害されて動かなくなりがち
[公式ドキュメント](#)を確認して必要な通信が可能になるように設定すること



場所	種類	名前	状態
Japan East	ApiManagement Co...	https://fdsample-apim.manageme...	成功
Japan East	ApplicationInsightsIn...	dc.services.visualstudio.com	成功
Japan East	外部キヤッショ	https://login.windows.net	成功
Japan East	Azure Active Directory	https://apikv-kgfq9happhi1pvd...	成功
Japan East	BLOB ストレージ	apimstpl08nqrto9ipziblob.c...	成功
Japan East	CaptchaEndpoint	https://partner.prod.remp...	成功
Japan East	「ファイル」ストレージ	animstr108nqrto9ipzifile.cor...	成功

動作に必要な要件が満たされているか Azure Portal で確認可能

補足：API Management 内部ネットワークモード

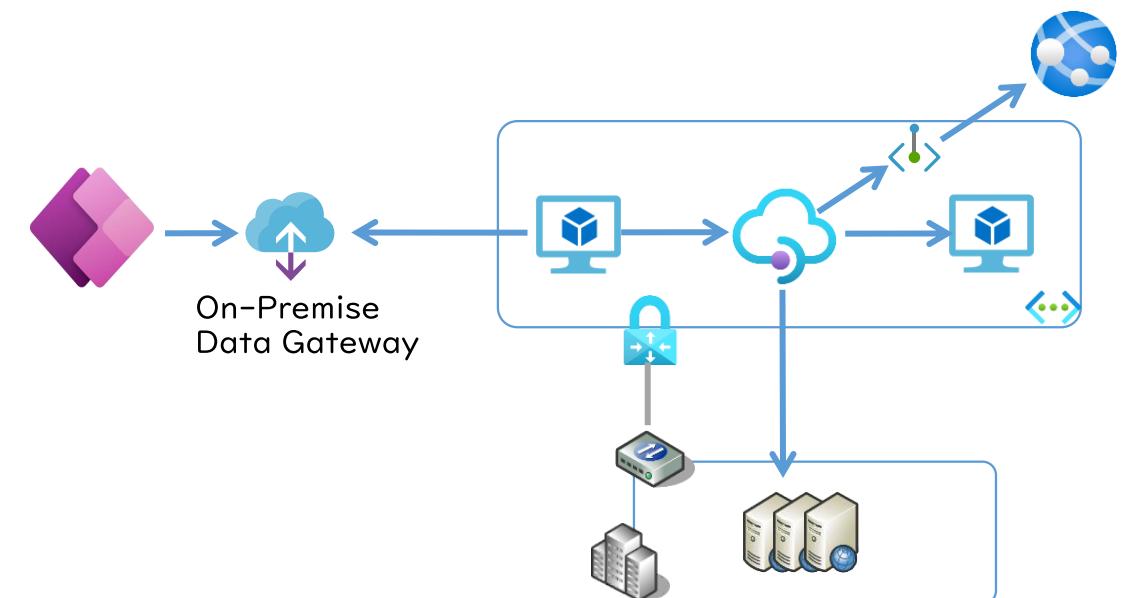
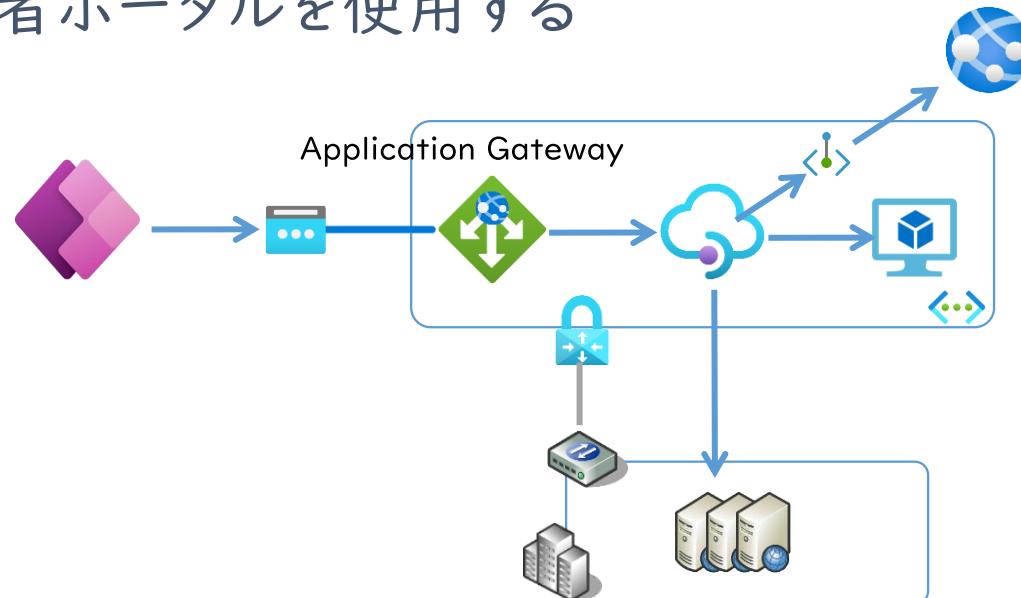
内部ネットワークモードを使用すると開発者ポータルや API Gateway も VNET 内でホストすることが出来る

Power Apps 等のパブリックネットワーク環境からアクセスするクライアントに対する通信経路の確保が別途必要になることに注意

例) [Application Gateway](#) を利用する

例) [オンプレミスデータゲートウェイ](#)を使用する

内部モードでは Azure Portal から API のテストが出来なくなるため、VNET 内から開発者ポータルを使用する



Appendix B

Azure AD 認証によるセキュア API

認証方式の課題

Power Apps キャンバスアプリ

キャンバスアプリの利用そのものには Azure AD 認証が必須であり、共有されたアプリしか利用できないためアクセス制限もできている

ただし「接続を共有する」方法であるためユーザーが特定できない、呼び出し回数も制限されやすいという課題がある

バックエンド API (App Service)

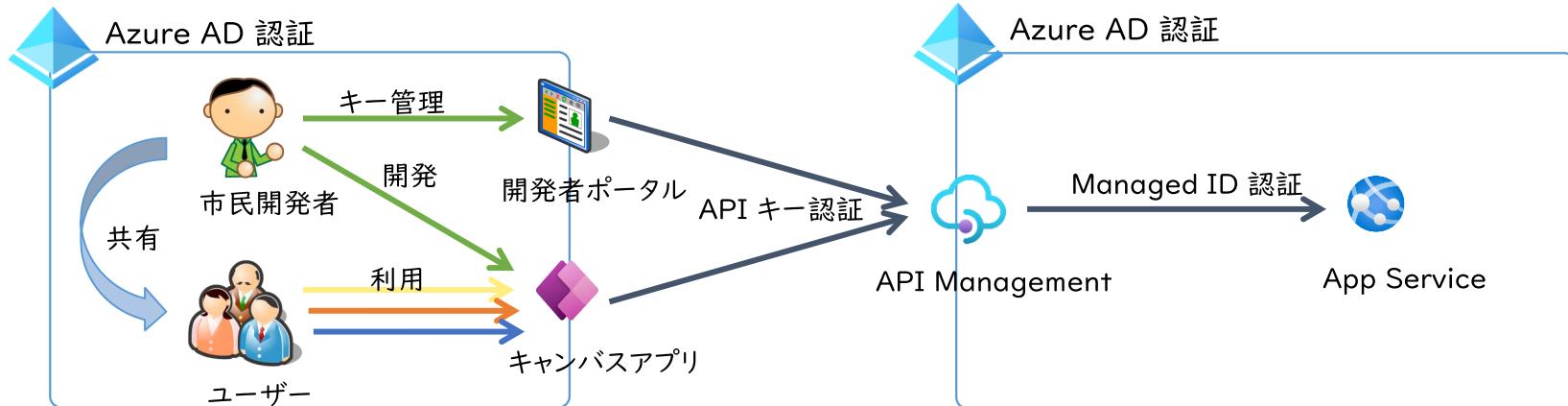
App Service などの Azure PaaS は明示的に認証を有効化しないとインターネット上の任意のユーザーから API の呼び出しが可能

直接の呼び出し元となる API Management を認証する、あるいは Power Apps を操作しているエンドユーザーを認証する方法が考えられる

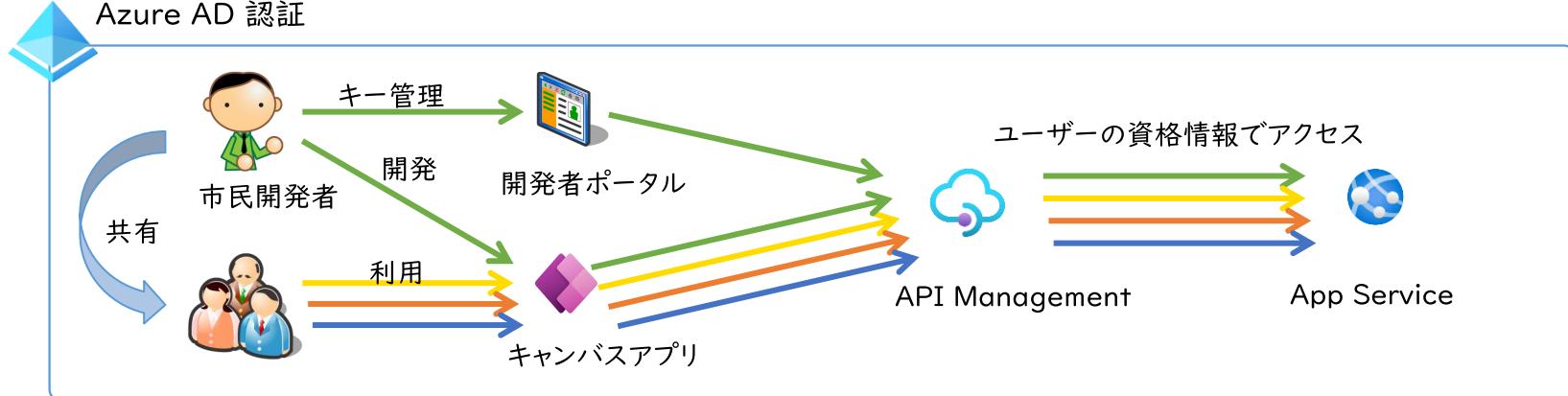
API の認証方式の整理

認証方式にはいくつかの組み合わせが考えられるが、ここで代表的な2つのパターンを紹介する

アプリケーション信頼型の
認証モデル



End to End の
ユーザー認証モデル



アプリケーション信頼型の認証モデル

バックエンド API を不特定多数のユーザーやアプリから呼び出されないように保護する

構成が比較的容易だが、ユーザーが特定できずにスロットリングも共有してしまう問題は改善しない
比較的小規模な利用想定の場合、あるいは暫定対策として考慮するとよい

アプリケーション信頼型の認証モデル

API Management は API キーを持つアプリからの呼び出しを許可

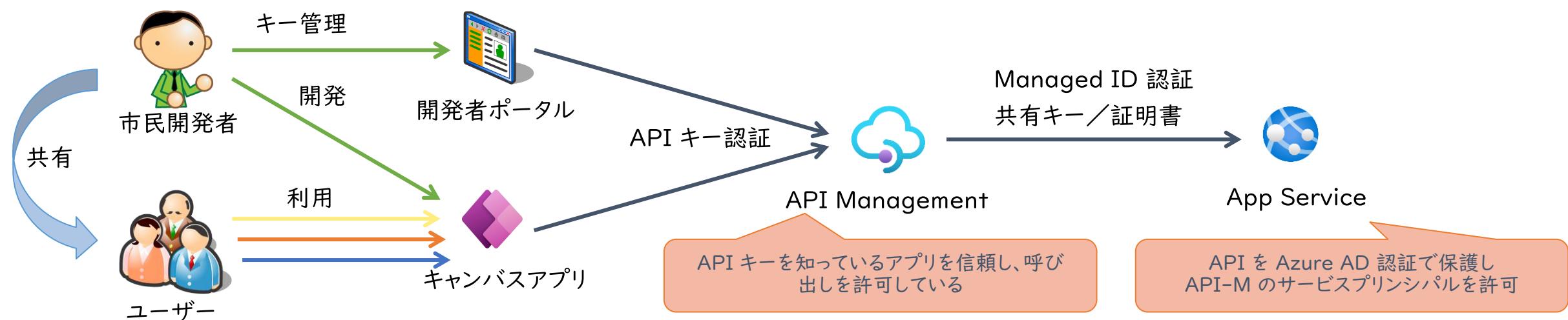
各ユーザーは API キーを所有しておらず、市民開発者側から接続を共有されている
あるいは、一般ユーザーも開発者ポータル等を使用して API キーを払い出すことが出来る
この部分はすでに実装済みのため割愛

バックエンド API は API Management からの呼び出しを許可

バックエンド API が App Service でホストされている場合にはその AAD 認証機能を使用し、API Management の Managed ID を認証させる構成が容易（後述）

バックエンド API が AAD 認証が使用できない場合は共有キー認証や証明書認証も考えられる

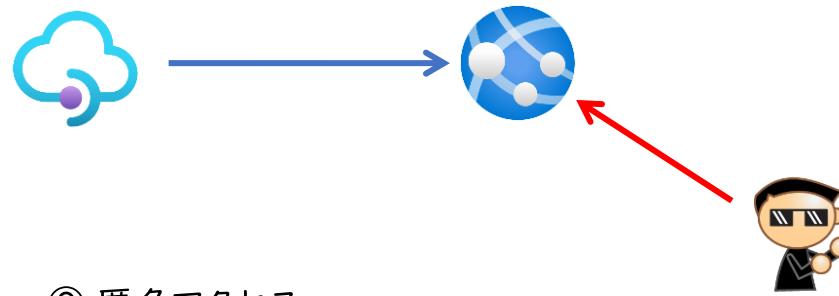
Azure Functions などの API キーによるアクセス制御が可能なプラットフォームであればそちらの機能を利用しても良い



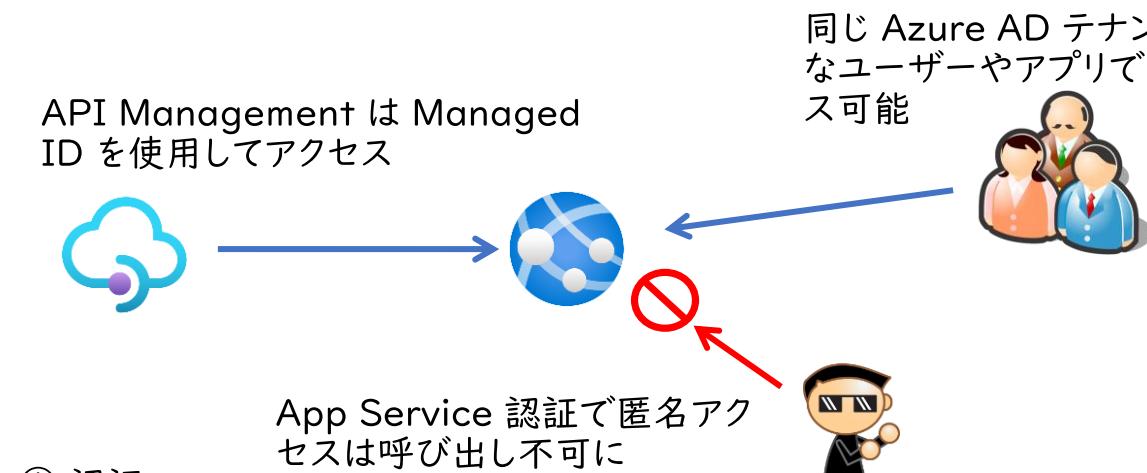


アプリケーション信頼型の認証モデル

初期状態では App Service は匿名アクセスが可能



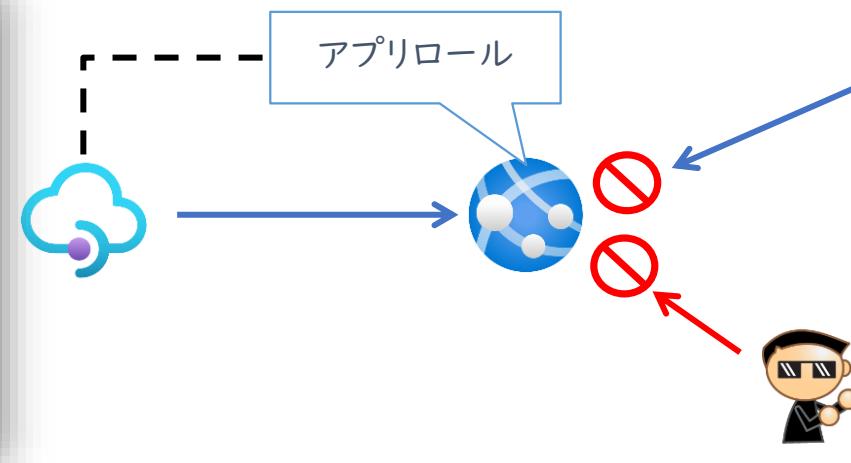
API Management は Managed ID を使用してアクセス



② 認証+アクセス制御

API Management は「割り当てられている」のでアプリロールに所属するクレームを提出できる

Screenshot of the Azure portal showing API permissions for an API Management application named 'ainaba-apim-1124'. The 'Access許可' (Permissions) section is highlighted with a red box. It shows a table with one row for 'ainaba-apibe-1124' with the role 'Daemon' and type 'Application'. The 'API名' (API Name) column has a dropdown menu with 'ainaba-apibe-1124' selected.



認証されていても明示的に「割り当てられていない」ユーザー やアプリケーションはアクセス不可

Screenshot of the Azure portal showing assigned roles for a service principal named 'ainaba-apim-1124'. The 'User & Group' section is highlighted with a red box. It shows a table with one row for 'ainaba-apim-1124' with the role 'Daemon' assigned. The 'Role' column has a dropdown menu with 'Daemon' selected.

API Management と App Service の AAD 認証



前述の手順で構築された API で認証を有効にして保護する

App Service 上でホストされる API はそのままでは匿名アクセスが可能なのでまずは Easy Auth を有効にして Azure AD 認証を要求する

API Management 側では Managed ID を有効にし、ポリシー機能を使用して呼び出し時の認証ヘッダーにバックエンド API アクセスの認可を表すトークンを組み込む

The screenshot shows the Microsoft Azure API Management service interface. On the left, the navigation bar includes 'ainaba-apim-1124' and 'マネージド ID'. The main area displays a 'Todo API > All operations > Policies' screen. A red dashed box highlights a specific policy element in the code editor:

```
1<!--  
2  IMPORTANT:  
3  - Policy elements can appear only within the <inbound>, <outbound>  
4  - To apply a policy to the incoming request (before it is f  
5  - To apply a policy to the outgoing response (before it is  
6  - To add a policy, place the cursor at the desired insertio  
7  - To remove a policy, delete the corresponding policy state  
8  - Position the <base> element within a section element to i  
9  - Remove the <base> element to prevent inheriting policies  
10 - Policies are applied in the order of their appearance, fr  
11 - Comments within policy elements are not supported and may  
12 -->  
13 <policies>  
14   <inbound>  
15     <authentication-managed-identity resource="b2b6a36a-ecf4-4d9d-9a44-6...>  
16   </inbound>  
17   <backend>  
18     <base />  
19   </backend>  
20 </policies>
```

An orange callout box at the bottom points to the 'authentication-managed-identity' element with the text: "App Service 側のアプリケーション ID を転記".

The screenshot shows the Microsoft Azure App Service configuration interface for 'ainaba-apibe-1124'. The '認証' (Authentication) section is highlighted with a red box. It shows the following settings:

ID プロバイダー	Microsoft (ainaba-apibe-1124)
アプリ (クライアント) ID	b2b6a36a-ecf4-4d9d-9a44-6...

※ この段階では「認証が必要」なだけでアクセス制御はできていない



呼び出し元のアクセス制御

[保護された Web API のアプリの登録](#)

API を呼び出せるアプリケーションを制限するために「割り当て」によるアクセス制御を行う

App Service の認証を有効化した際に Azure AD に登録された アプリケーションの設定画面にて「アプリ ロール」を定義する

同時に作成されているサービスプリンシパルの画面で「割り当てが必要」に設定することで、明示的にアクセスを許可されたアプリケーションのみ API を利用可能とする

App Service の認証設定画面

The screenshot shows the 'Authentication' section of the App Service configuration. It lists providers: 'App Service' (selected), 'OpenID Connect' (disabled), and 'Facebook' (disabled). Under 'App Service', it shows 'Enabled' status, 'HTTP 401 (認可されていない) を返す' (Return 401 (Unauthorized)), and '有効' (Enabled) for 'Tokens issued'. A red box highlights the 'Microsoft (ainaba-apibe-1124)' entry under 'ID Provider'.

Azure AD に登録されたアプリケーション

The screenshot shows the 'App roles' section of the Azure AD application registration. It lists a role named 'Daemon' with the description 'API access'. A red box highlights the 'API access' role definition.

登録されたアプリケーションのサービスプリンシパル

The screenshot shows the 'Delegated permissions' section of the service principal configuration. It lists the 'API access' role assigned to the application. A red box highlights the 'Assigned to this app' checkbox under the 'Role assignments' section.



呼び出し元のアクセス制御

API Management の Managed ID を、API アプリのロールに割り当てる

現在この設定は Azure Portal で実施できないため、Power Shell で設定を行う
New-AzureADServiceAppRoleAssignment の各パラメタは以下の値を利用する

ResourceId : API アプリのサービスプリンシパルのオブジェクト ID

Id : 割り当てるロールのオブジェクトID

ObjectId : API Management のサービスプリンシパル のオブジェクト ID

PrincipalId : ObjectId と同じ値を使用する

The screenshot shows the Azure Portal interface with the URL 'ainaba-apim-1124 | マネージド ID' in the top navigation bar. On the left, there's a sidebar with '検索 (Ctrl+/' and 'ポータルの概要'. Under 'API Management サービス', there's a section for 'システム割り当て済み' (Assigned via system) which says 'システム割り当てマネージド ID は 1 つまでしか付与することができます。マネージド ID' (A managed ID can only be assigned once). Below it, there's a 'オブジェクト (プリンシパル) ID' field containing '409b56b3-3c46-4053-9dbf-de288d962631' with a red box around it.

The screenshot shows a PowerShell window with the following command:

```
PS > Connect-AzureAD  
PS > New-AzureADServiceAppRoleAssignment  
    -ResourceId b69676ea-2ed8-4a3c-b701-0a8cb86285b4  
    -Id 8585681c-6b5e-4d48-8663-957a70a46eeef  
    -ObjectId 409b56b3-3c46-4053-9dbf-de288d962631  
    -PrincipalId 409b56b3-3c46-4053-9dbf-de288d962631
```

The parameters for ResourceId, Id, ObjectId, and PrincipalId are highlighted with red boxes.

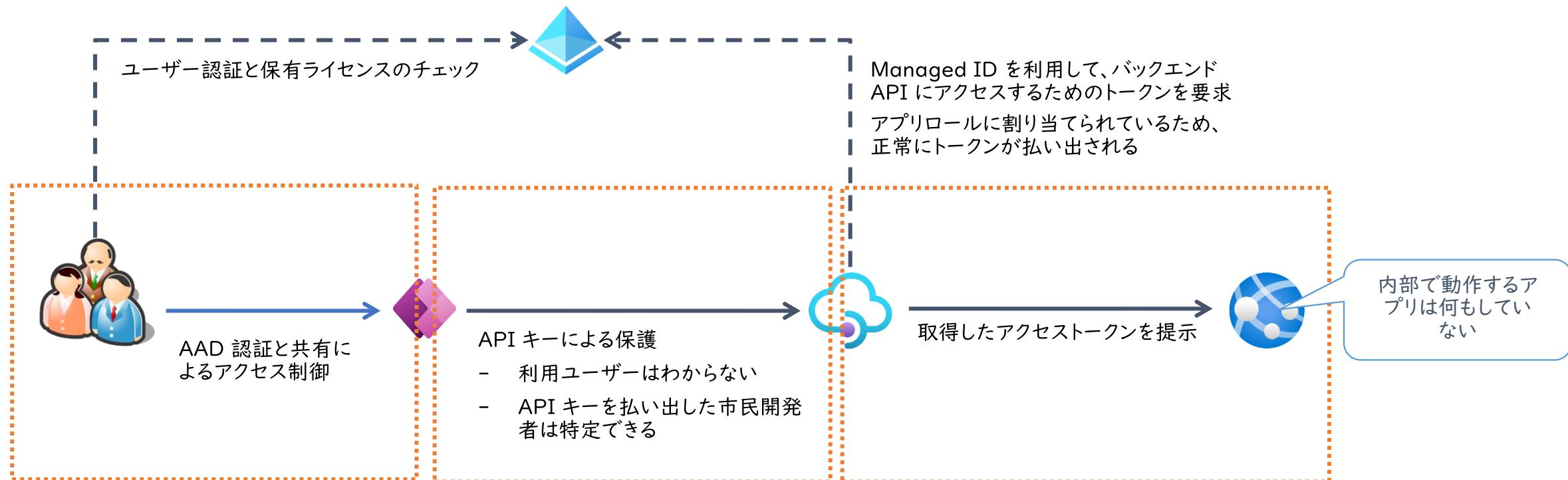
The screenshot shows the Azure Portal with the URL 'ainaba-apibe-1124 | プロパティ' (Properties). It displays the application role assignment settings. The 'オブジェクト ID' field is highlighted with a red box and contains 'b69676ea-2ed8-4a3c-b701-0a8cb86285b4'. The '割り当てが必要ですか?' (Is assignment required?) checkbox is checked ('はい'). The 'ユーザーに表示しますか?' (Show user) checkbox is also checked ('はい'). A detailed description of the application role is shown below, along with a table mapping display names to principal IDs.

表示名	説明	許可されたメンバーの種類	値
Daemon	API access	アプリケーション	Daemon

補足：認証処理の挙動と整理

ここで紹介した方式は3種類のアクセス制御方式を組み合わせて使っている

各サービス間の通信単位での認証とアクセス制御になっており、それぞれは独立した構成となる



補足：Azure AD に自動登録されるアプリ

開発者ポータルの AAD 認証と Managed ID の有効化によって2つの似たようなオブジェクトが AAD に作成される

The screenshot shows the Azure Active Directory overview page. In the search bar at the top, the text 'ainaba-apim-1124' is entered. Below the search bar, there are four search fields: 'ユーザー', 'グループ', 'デバイス', and 'アプリの登録'. Under the 'アプリの登録' field, two results are listed: 'ainaba-apim-1124' and 'ainaba-apim-1124'. An orange callout box points to the search bar with the text: 'Azure AD の概要画面から API Management の「名前」で検索すると複数のオブジェクトが取得できる'.

Azure AD の概要画面から API Management の「名前」で検索すると複数のオブジェクトが取得できる

The screenshot shows the developer portal's application registration details for 'ainaba-apim-1124'. The left sidebar includes sections like 'クイックスタート', '統合アシスタント', '認証', '証明書とシークレット', 'トークン構成', 'API のアクセス許可', 'API の公開', 'アプリ ロール', and '所有者'. The main content area displays basic information such as the application name ('ainaba-apim-1124'), client ID ('940c8847-db9d-429e-9ed7-bbccf3ecdfb2'), and redirect URIs ('https://localhost:4244/api/v1/auth'). A note at the bottom states: '2020年6月30日以降、Azure Active Directory 関連ライブラリ(ADAL)および Azure AD Graph に新しい機能はもう追加されません。テクニカルサポートとセキュリティ更新プログラムは今後提供されますが、機能更新プログラムは提供されません。アプリケーション ID、Microsoft 認証ライブラリ(MSL)および Microsoft Graph にアップグレードする必要があります。詳細情報'.

The screenshot shows the Azure portal's properties for the service principal 'ainaba-apim-1124'. The 'プロパティ' tab is selected. It displays the application ID ('940c8847-db9d-429e-9ed7-bbccf3ecdfb2'), object ID ('e049894e-d2a1-42d2-b147...'), and other metadata. An orange callout box points to the application ID with the text: '開発者ポータルをあらわすアプリケーションとサービスプリンシパル'.

開発者ポータルをあらわす
アプリケーションとサービスプリンシパル

The screenshot shows the Azure portal's properties for the service principal 'ainaba-apim-1124'. The 'プロパティ' tab is selected. It displays the application ID ('940c8847-db9d-429e-9ed7-bbccf3ecdfb2'), object ID ('e049894e-d2a1-42d2-b147...'), and other metadata. An orange callout box points to the application ID with the text: 'ゲートウェイに割り当てられた
Managed ID が使用するサービスプリ
ンシパル(アプリなし)'.

ゲートウェイに割り当てられた
Managed ID が使用するサービスプリ
ンシパル(アプリなし)

End to End ユーザー認証モデル

バックエンド API がユーザーを認証するモデル

コネクタや API Management の「裏に隠れた」バックエンド API がユーザーを認証するというのは違和感があるかもしれないが、ユーザー個人に紐付く情報を扱う API であればむしろ必須の構成になる

例) Outlook は各々のユーザー専用のメールボックスを扱う

例) Sharepoint で共有されたデータには許可されたユーザーのみがアクセス可能

構成としては若干複雑だが、いくつかのメリットが考えられる

API Management アクセスのための API キーの管理が不要になる

ユーザーは日常的に行うユーザー認証だけで利用可能な API が出来上がる

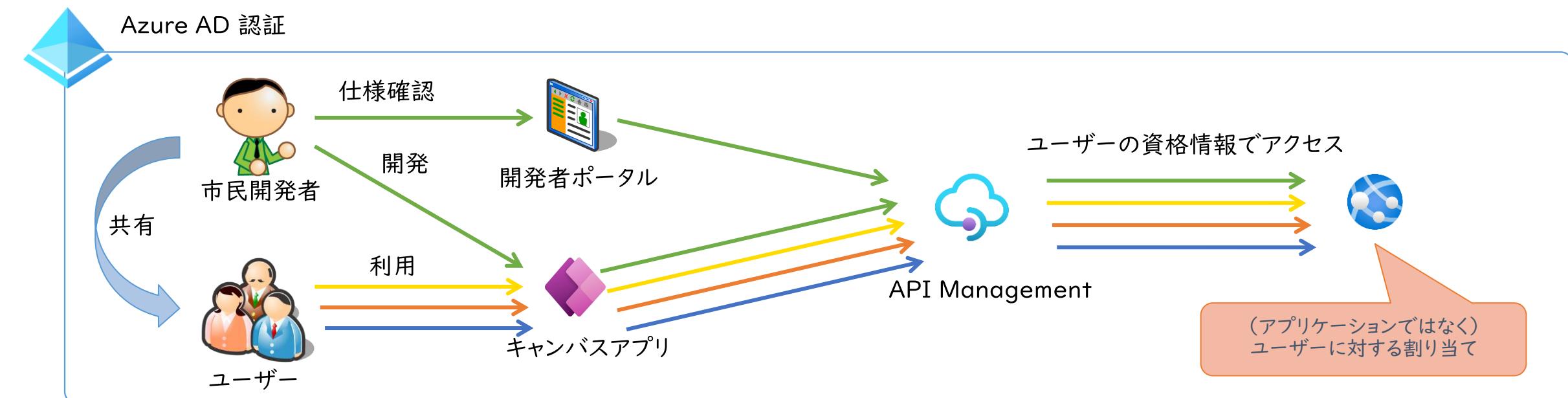
バックエンド API ではユーザーのロールやグループ情報などを用いたきめ細かなアクセス制御が可能

Etc…

End to End ユーザー認証モデル

フロントエンドアプリはユーザーの代理としてバックエンド API にアクセスする
ここではフロントエンドアプリは開発者ポータルとカスタムコネクタの 2 つが存在することに注意
各々のアプリはユーザーの同意に基づいてバックエンド API へのアクセスが許可される

バックエンド API はユーザー情報に基づいた柔軟なアクセス制御が可能
ユーザー個人に紐付く情報を扱う API であれば、API Management ではなくユーザーを認証できる必要がある
Outlook や Sharepoint といった API と同じ仕組みが実現できる
API Management はユーザーの認証情報を受け渡すだけになる（トークンの簡易的なチェックは可能）

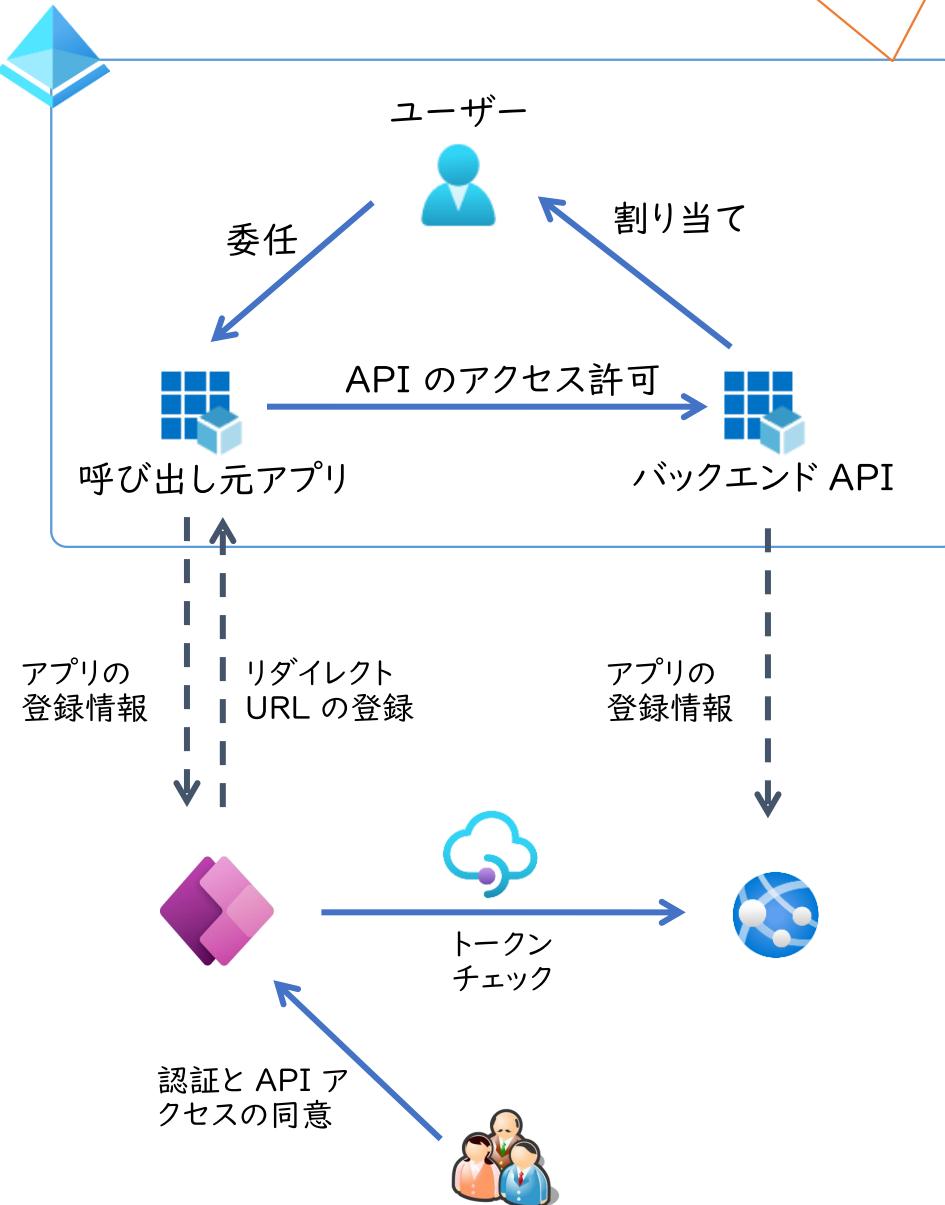


End to End ユーザー認証モデル

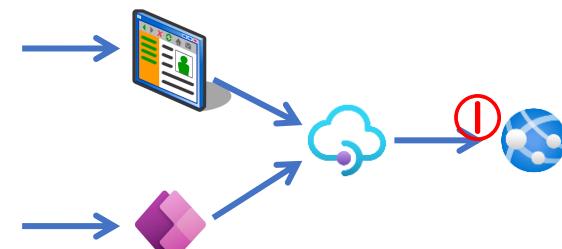
以降で紹介する設定内容が極めて煩雑なので、何をやっているか俯瞰しておく

認証フローに登場するアプリを AAD に登録する
バックエンド API を 1つ登録し、API を公開する(スコープ)
ユーザーに対してバックエンド API を割り当てる
呼び出し元のアプリはバックエンド API に委任アクセスを行う
各アプリの実体に OAuth の設定を行っていく
バックエンド API となる App Service
呼び出し元アプリである Power Apps Connector
呼び出し元アプリとなる API-M 開発者ポータル
API-M のゲートウェイは認証には直接かかわらない
無制限呼び出しを防ぐためにトークンチェックだけ行う

呼び出し元となるアプリは2つあるため2セットの構成が必要
- Power Apps コネクタ
- API-M 開発者ポータル



ユーザーを認証するバックエンド API



Azure AD にバックエンド API を表すアプリを登録する

App Service で Azure AD 認証を有効にするとアプリが自動登録される

登録されたアプリが公開する API スコープを定義

アクセスを許可するユーザーへの割り当てを行う

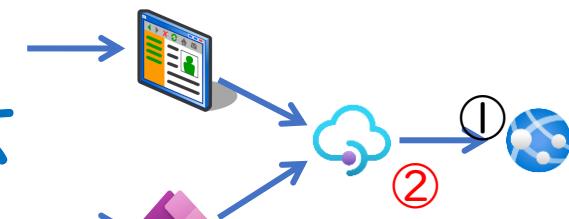
App Service 認証の有効化

API のスコープ

割り当ての強制

API を利用できるユーザーの割り当て

API Management アクセス許可の除去



前述のアプリケーション信頼モデルの構成をしていた場合は
その設定を除去する

バックエンド API に対する authentication-managed-identity ポリシーの除去
API Management に対して割り当てたアプリロールの除去

REVISION 2 CREATED Nov 24, 2021, 5:56:17 PM

Design Settings Test Revisions Change log

Search operations Filter by tags Group by tag Add operation

All operations

GET Get todo item by ...
GET List all todo items ...
POST New todo item ...
PATCH Update todo item ...

Todo API > All operations > Policies

```
<!--  
IMPORTANT:  
- Policy elements can appear only within the <inbound>, <outbound>, <backend>, <base>, <choose>, <foreach>, <msc>, <repeating-entities>, <script>, <trycatch>, and <trycatchall> elements.  
- To apply a policy to the incoming request (before it is forwarded to the application), place the policy within the <inbound> element.  
- To apply a policy to the outgoing response (before it is sent back to the client), place the policy within the <outbound> element.  
- To add a policy, place the cursor at the desired insertion point and click the Insert Policy button on the toolbar.  
- To remove a policy, delete the corresponding policy statement.  
- Position the <base> element within a section element to inherit policies from that section.  
- Remove the <base> element to prevent inheriting policies.  
- Policies are applied in the order of their appearance, from top to bottom.  
- Comments within policy elements are not supported and may cause parsing errors.  
-->  
<policies>  
  <inbound>  
    <base />  
    <authentication-managed-identity resource="b2b6a36a-e1f1-43d9-8e00-1a2a2a2a2a2a" />  
  </inbound>  
  <backend>  
    <base />  
  </backend>
```

ainaba-apibe-1124 | ユーザーとグループ ...

概要 テプロイ計画 管理

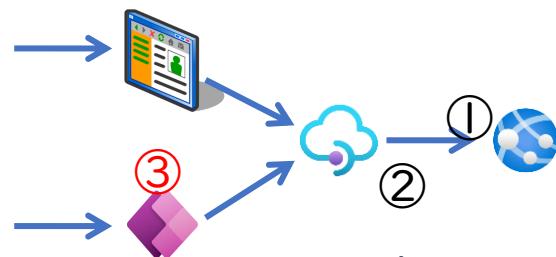
+ ユーザーまたはグループの追加 編集 削除 資格情報の更新 フィードバックがある場合

アドバイス: アプリケーションは、割り当てられたユーザーのマイ アプリ内に表示されます。これを表示しないようにするには、プロパティの中で [ユーザー] に表示しないように設定してください。

最初の 200 件を表示しています。すべてのユーザーとグループを検索するには、表示名を入力してください。

表示名	オブジェクトの種類
AI ainaba-apim-1124	ServicePrincipal
EA ESLZ admin	ユーザー
U0 user 01	ユーザー
AL ainaba live	ユーザー

カスタムコネクタのアプリ登録



カスタムコネクタを表すアプリケーションを Azure AD に登録し、バックエンドAPI のアクセス許可を要求するように設定

カスタムコネクタがユーザーのフリをしてバックエンド API にアクセスする（ユーザーの権限をアプリに委任する）必要がある

アプリケーションの登録 ...

* 名前
このアプリケーションのユーザー向け表示名(後で変更できます)。
ainaba-powerapp-connector-1210

サポートされているアカウントの種類
このアプリケーションを使用したりこの API にアクセスしたりできるのはだれですか?
 この組織ディレクトリのみに含まれるアカウント (ainabaeslz のみ - シングル テナント)
 任意の組織ディレクトリのアカウント (任意の Azure AD ディレクトリ - マルチテナント)
 任意の組織ディレクトリ内のアカウント (任意の Azure AD ディレクトリ - マルチテナント) と個人の Microsoft アカウント (Skype、Xbox)
 個人用 Microsoft アカウントのみ

選択に関する詳細...

リダイレクト URI (省略可能)
ユーザー認証が成功すると、この URI に認証応答を返します。この時点での指定は省略可能で、後ほど変更できますが、ほとんどの認証システムは、
Web 例: https://example.com/auth ✓

ホーム > ainabaeslz > ainaba-powerapp-connector-1210
ainaba-powerapp-connector-1210 | API のアクセス許可

概要 クイックスタート 統合アシスタント

管理 ブランド 認証 証明書とシークレット トーカン構成 API のアクセス許可 API の公開 アプリ ロール 所有者 ロールと管理者 | プレビュー マニフェスト サポート + トラブルシューティング

API アクセス許可の要求

このアプリの API
ainaba-apibe-1124
api://b2b6a36a-ecf4-4d9d-9a44-6d4c38254b28

アプリケーションに必要なアクセス許可の種類

委任されたアクセス許可
アプリケーションは、サインインしたユーザーとして API にアクセスする必要があります。

アクセス許可を選択する

+ アクセス許可の追加 ✓ ainabaeslz に管理者の同意が必要

API / アクセス許可の名前 種類 説明

Microsoft Graph (1)

User.Read 委任済み Signature

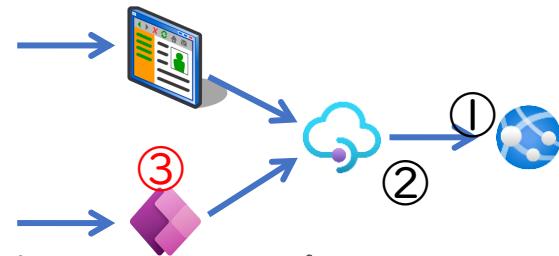
アクセス許可とユーザーの同意を表示および管理するために、この API にアクセスするには、管理者の同意が必要です。

いいえ 管理者の同意が必要

API で定義したスコープ

user_impersonation ① Access ainaba-apibe-1124

カスタムコネクタのアプリ登録



Azure AD に登録した情報をカスタムコネクタから利用するため以下の情報を控えておく

- アプリケーション(クライアント)ID : アプリ登録時点で決定される GUID
ディレクトリ(テナント)ID : 登録した Azure AD テナントを表す GUID
クライアントシークレット : 新しく作成した際に発行される

ainaba-powerapp-connector-1210

検索 (Ctrl+ /)

削除 エンドポイント プレビュー・機能

概要 クイック スタート 統合アシスタント

管理 ブランド 認証 証明書とシークレット トーカン構成 API のアクセス許可

へ 基本

表示名 : ainaba-powerapp-connector-1210

アプリケーション(クライアント)ID : 0980b2f0-9783-4f01-b632-26c743e3b445

オブジェクト ID : 2c595bb7-5beb-49fb-a2b9-81bca749f1a8

ディレクトリ(テナント)ID : f52c30c4-5504-48e5-b3d8-f22bfd4ce170

サポートされているアカウント... : 所属する組織のみ

新しく強化されたアプリの登録へようこそ。アプリの登録 (レガシ) からの変更点を確認する

2020 年 6 月 30 日以降、Azure Active Directory 認証ライブラリ (ADAL) および Azur

ainaba-powerapp-connector-1210 | 証明書とシークレット

検索 (Ctrl+ /)

フィードバックがある場合

概要 クイック スタート 統合アシスタント

管理 ブランド 認証 証明書とシークレット トーカン構成 API のアクセス許可 API の公開 アプリ ロール

アプリケーション登録証明書、シークレット、フェデレーション資格情報は、下のタブにあります。

証明書 (0) クライアント シークレット (1) フェデレーション資格情報 (0)

トーカンの要求時にアプリケーションが自身の ID を証明するために使用する秘密の文字列です。アプリケーション パスワードと呼ばれます。

+ 新しいクライアント シークレット

説明	有効期限	値
Secret for custom connector v1	2022/6/10	m50*****

カスタムコネクタの認証を構成

認証タイプ
API によって実装される認証の種類を選びます *

OAuth 2.0

編集

OAuth 2.0

ID プロバイダー
Azure Active Directory

Client id *
0980b2f0-9783-4f01-b632-26c743e3b445

Client secret *
.....

Login URL
https://login.windows.net

Tenant ID
f52c30c4-5504-48e5-b3d8-f22bfd4ce170

Resource URL *
api://b2b6a36a-ecf4-4d9d-9a44-6d4c38254b28

Enable on-behalf-of login
true

スコープ
スコープ

リダイレクト URL
https://global.consent.azure-apim.net/redirect

編集

カスタムコネクタの認証タイプを OAuth 2.0 に変更する

ID プロバイダー	: Azure Active Directory
Client Id	: アプリ登録時に控えた値
Client secret	: アプリ登録時に控えた値
Tenant ID	: アプリ登録時に控えた値
Resource URL	: バックエンド API アプリの URI

ainaba-apibe-1124

検索 (Ctrl +/)

削除 エンドポイント プレビュー機能

概要

クイック スタート

統合アシスタント

管理

ブランド

認証

証明書とシークレット

トークン構成

API のアクセス許可

API の公開

基本

表示名
ainaba-apibe-1124

アプリケーション (クライアント) ID
b2b6a36a-ecf4-4d9d-9a44-6d4c38254b28

オブジェクト ID
ae640380-9121-4516-a2c4-7a09c60d72b9

デイレクトリ (テナント) ID
f52c30c4-5504-48e5-b3d8-f22bfd4ce170

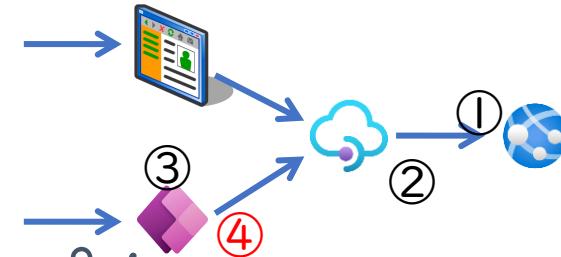
サポートされているアカウントの種類
所属する組織のみ

クライアントの資格情報
0 証明書、1 シークレット

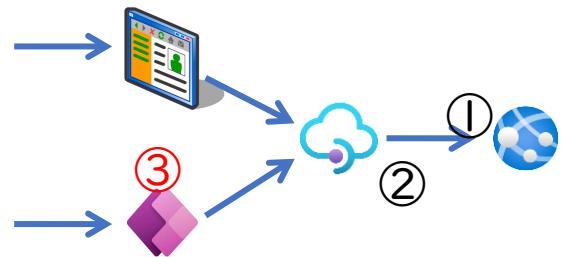
リダイレクト URL
リダイレクト URL を追加する

アプリケーション ID の URI
api://b2b6a36a-ecf4-4d9d-9a44-6d4c38254b28

ログイン リフレッシュ令牌のマップ ノード ノード ノード
ainaba-apibe-1124



カスタムコネクタのアプリ登録



カスタムコネクタの認証設定を保存するとリダイレクト URL が発行されるため、先ほど登録したカスタムコネクタを表すアプリに設定する

ホーム > ainabaeslz > ainaba-powerapp-connector-1210 | 認証

ainaba-powerapp-connector-1210 | 認証

検索 (Ctrl + /) 保存 破棄 フィードバックがある場合

概要 クイック スタート 統合アシスタント

管理 ブランド 認証 証明書とシークレット トークン構成 API のアクセス許可 API の公開 アプリ ロール 所有者 ロールと管理者 | プレビュー マニフェスト

プラットフォーム構成

このアプリケーションが対象としているプラットフォームまたはデバイスによっては、リダイレクト URI、特定の認証設定、ブレイブルにたゞ追加構成が必要となる場合があります。

+ プラットフォームを追加

サポートされているアカウントの種類

このアプリケーションを使用したりこの API にアクセスしたりできるのはだれですか?

この組織ディレクトリのみに含まれるアカウント (ainabaeslz のみ - シングル テナント)

任意の組織ディレクトリ内のアカウント (任意の Azure AD ディレクトリ - マルチテナント)

判断に役立つヘルプの表示...

⚠️ サポートされている機能が一時的に異なるため、既存の登録に関して個人用 Microsoft アカウントを有効にしないでください。アカウントを有効にする必要がある場合、マニフェスト エディターを使用して有効にできます。これらの制限に関する詳細

サポート + トラブルシューティング 詳細設定

Web の構成

リダイレクト URI

ユーザーの認証またはサインアウトに成功した後に認証応答 (トークン) を返すときに宛先として受け入れる URL。ループ URL とも呼ばれます。リダイレクト URI と削除の詳細情報

https://global.consent.azure-apim.net/redirect

フロントチャネルのログアウト URL

ここでは、アプリケーションがユーザーのセッション データをクリアするように要求を送信します。これは、シングルサインアウトが正常に動作するために必要です。

例: https://example.com/logout

暗黙的な許可およびハイブリッド フロー

承認エンドポイントから直接トークンを要求します。アプリケーションにシングルページ アーキテクチャ (SPA) があり、承認コード フローを使用していない場合、または JavaScript で Web API を起動する場合は、アクセストークンと ID トークンの両方を選択します。ハイブリッド認証を使用する ASP.NET Core Web アプリや他の Web アプリでは、ID トークンのみを選択します。トークンの詳細情報。

承認エンドポイントによって発行してほしいトークンを選択してください。

アクセス トークン (暗黙的なフローに使用)

ID トークン (暗黙的およびハイブリッド フローに使用)

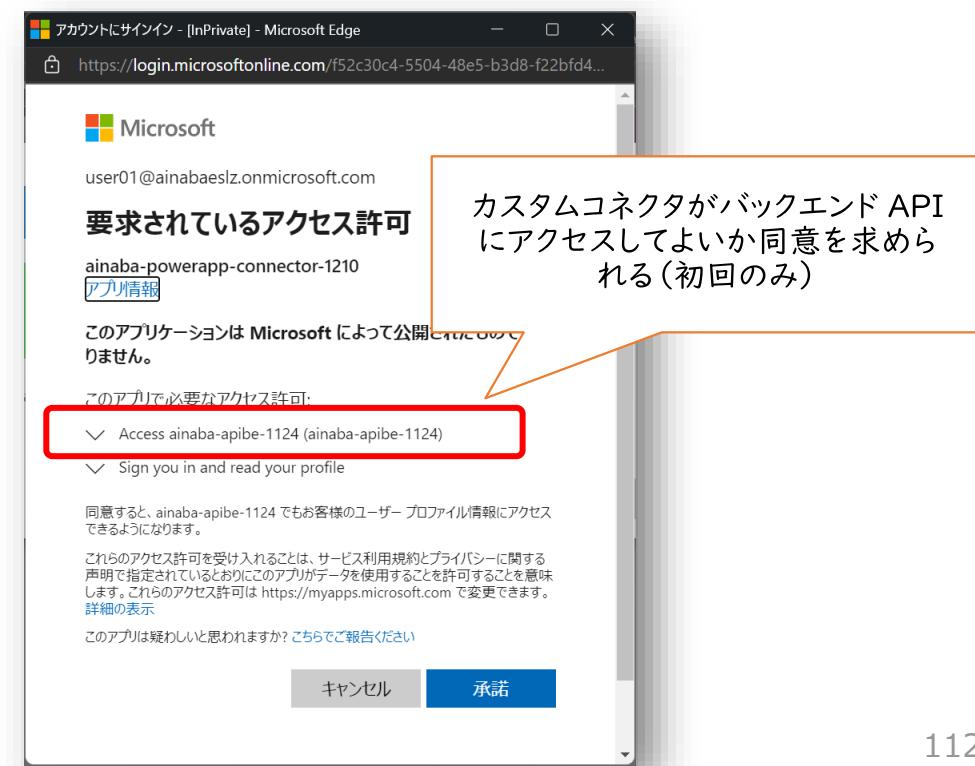
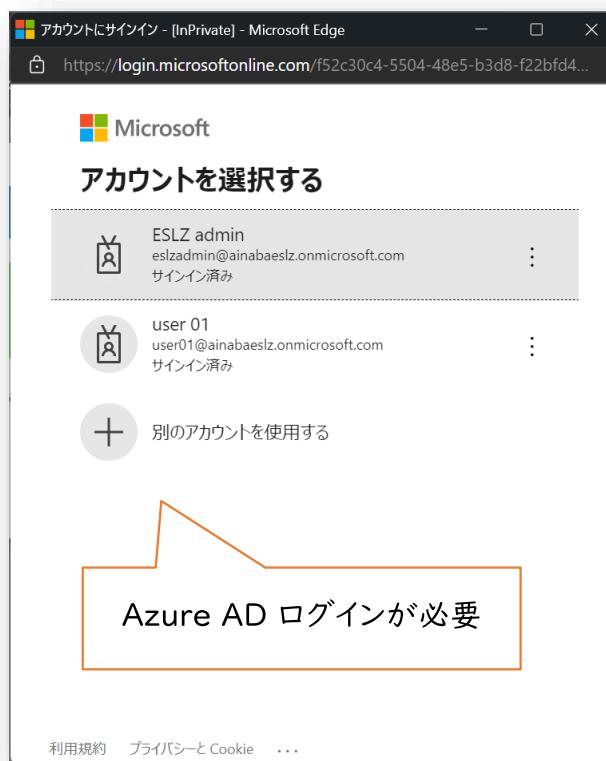
カスタムコネクタの接続を作成する



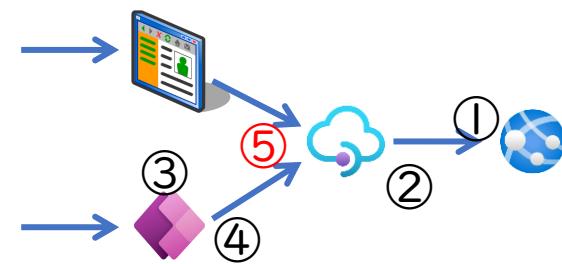
コネクタの認証設定を変更したため接続を再度作り直す

変更前は接続の作成時に API Management にアクセスするためのキーが必要だったが、設定変更によりユーザーの Azure AD 認証情報が使用できるようになっている

この方法で作成した接続は**他のユーザーに共有しないこと**



API Management の保護設定を変更



カスタムコネクタから API Management のキーが送信されなくなるため、キーなしでもアクセスできるように変更する

単にキーを不要にするだけでは任意のユーザーが API Management を呼び出せてしまうため、新たに送信されるようになったトークンのチェックを行うようにする

REVISION 3 CREATED Nov 29, 2021, 4:59:47 PM

Design Settings Test Revisions Change log

Subscription

Subscription required チェックを外す

Header name: Ocp-Apim-Subscription-Key

Query parameter name: subscription-key

Security

User authorization: None

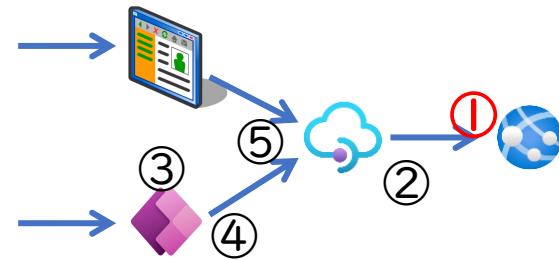
```
<inbound>
  <base />
  <validate-jwt header-name="Authorization" require-scheme="Bearer"
    failed-validation-httpcode="401"
    failed-validation-error-message="Unauthorized. Access token is missing or invalid." >
    <!--
    <openid-config
      url="https://login.microsoftonline.com/f52c30c4-5504-48e5-b3d8-
f22bf4ce170/v2.0/.well-known/openid-configuration" />

    <audiences>
      <audience>b2b6a36a-ecf4-4d9d-9a44-6d4c38254b28</audience>
    </audiences>
  </validate-jwt>
</inbound>
```

トークンチェック テナントID

バックエンド API のアプリケーションID

Azure AD 認証エンドポイントの変更



validate-jwtが参照するメタデータエンドポイントが v2.0 になっているため、要求するアクセストークンのバージョンもそろえておく

バックエンド API を表す Azure AD アプリケーションのマニフェスト設定において、`accessTokenAcceptedVersion` を 2 に設定する(規定値は null)

ainaba-apibe-1124 | マニフェスト

検索 (Ctrl+ /) 保存 破棄 アップロード ダウンロード フィードバックがある場合

概要 クイック スタート 統合アシスタント

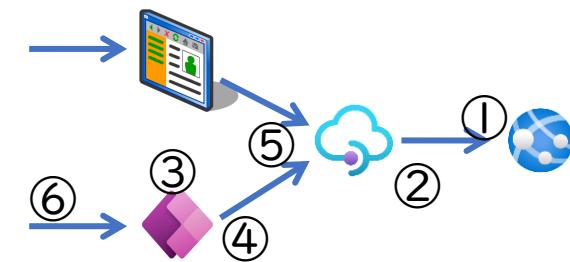
管理 ブランド 認証 証明書とシークレット トクン構成

```
{  
  "id": "ae640380-9121-4516-a2c4-7a09c60d72b9",  
  "acceptMappedClaims": null,  
  "accessTokenAcceptedVersion": 2,  
  "addIns": [],  
  "allowPublicClient": null,  
  "appId": "b2b6a36a-ecf4-4d9d-9a44-6d4c38254b28",  
  "appRoles": [  
    {  
      "allowedMemberTypes": [  
        "Application"  
      ]  
    }  
  ]  
}
```

メタデータ

```
# v2 メタデータエンドポイントのURLと記載されている Issuer  
# https://login.microsoftonline.com/tenantid/v2.0/.well-known/openid-configuration  
{  
  "issuer": "https://login.microsoftonline.com/tenantid/v2.0"  
}  
  
# v1 メタデータエンドポイントのURLに記載されている Issuer  
# こちらでアクセストークンが発行されてしまうと validate-jwt が通らなくなる  
# https://login.microsoftonline.com/tenantid/.well-known/openid-configuration  
{  
  "issuer": "https://sts.windows.net/tenantid"  
}
```

アプリ共有時の挙動



アプリを共有されたユーザーはコネクタの初回利用時に接続の作成を求められる

接続の作成には普段から使用している(=アプリを共有された)ユーザー アカウントで Azure AD 認証が“できれば”良い

バックエンド APIへのアクセス許可(=ユーザーの割り当て)がされていると、アクセストークンが取得できるため接続が作成され、実際にアプリの利用が可能になる

もう少しで終了します...
papp-apim-demo-1124-canvas は、次を使用するためにアクセス許可を必要とされています。続行するには、アクセス許可を付与してください。

Todo API v1
user 01 による接続

サインイン

許可 許可しない

Todo API v1
user 01 による接続

作成 許可しない

Microsoft Todo API v1 Sample API description

アカウントを選択する

- ESLZ admin
esladmin@ainabaeslz.onmicrosoft.com
サインイン済み
- Azure AD 管理者
aad-admin@ayuina20211005.onmicrosoft.com
サインイン済み
- Ayumi Inaba
ainaba@microsoft.com
Windows に接続済み
- Joni Sherman
JoniS@M365B61838.OnMicrosoft.com
Windows に接続済み

要求されているアクセス許可

ainaba-powerapp-connector-1210
アカウント情報

このアプリーションは Microsoft によって公開されたものではありません。

このアプリに必要なアクセス許可:

- ✓ Access ainaba-apibe-1124 (ainaba-apibe-1124)
- ✓ Sign in and read your profile
- 相手の代理として同意する

同意すると、ainaba-apibe-1124 でもお客様のユーザー プロファイル情報をアクセスするようになります。

このアプリのアクセス許可を授けられたことは、サービス利用規約に基づいてプライバシーに関する通知を表示する必要があります。このアプリがデータを使用することを許すことを確認してから、このアプリのアクセス許可を承認してください。

https://myapps.microsoft.com で変更できます。詳細の表示

このアプリは便利だと思いませんか？ こちらで報告ください

キャンセル 承認

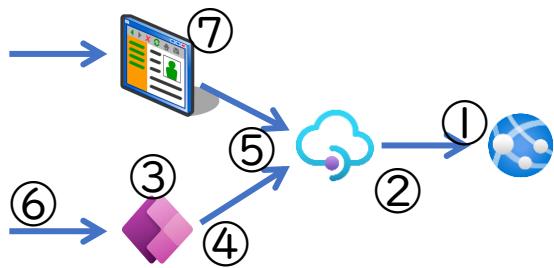
もう少しで終了します...
papp-apim-demo-1124-canvas は、次を使用するためにアクセス許可を必要とされています。続行するには、アクセス許可を付与してください。

Todo API v1
esladmin@ainabaeslz.onmicrosoft.com
サインイン済み

アカウントの切り替え

許可 許可しない

開発者ポータルからの API 呼び出し

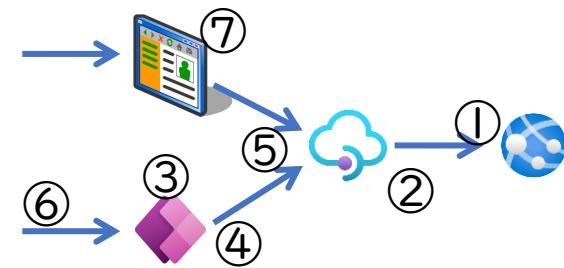


Power App カスタムコネクタと同様に開発者ポータル側にも
バックエンド API 呼び出し時の委任アクセス許可を与える

開発者ポータルに対する Azure AD ユーザー認証を有効にした段階でアプリ登録自体は
されている

The screenshot shows the 'API のアクセス許可' (API Access Permissions) section of the Azure API Management portal. It displays a list of permissions for the API 'ainaba-apibe-1124'. A red box highlights the '委任されたアクセス許可' (Delegated permissions) section, which contains the 'user_impersonation' permission under 'Azure Active Directory Graph' and 'Microsoft Graph'. An orange callout box points to this section with the text: '開発者ポータルがユーザーに代わって API アクセスを許可する設定' (Setting to allow the developer portal to grant API access on behalf of the user).

開発者ポータルからの API 呼び出し



開発者ポータルを表すアプリに関する以下の情報を控えておく

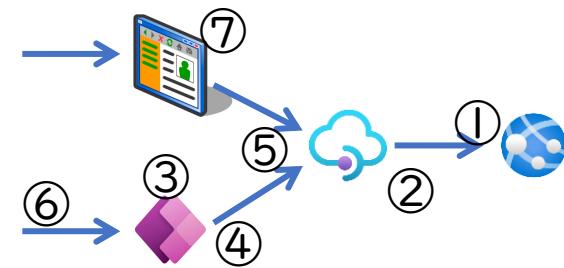
承認エンドポイントの URL、トークン エンドポイントの URL、テナント ID、クライアント ID、クライアントシークレット

クライアントシークレットは AAD 認証有効化時に自動発行されて AAD 側からは取得できないため、新規作成するか API Management の認証プロバイダ設定画面から取得するとよい

This screenshot shows the 'Endpoints' section of the Azure API Management service 'ainaba-apim-1124'. It displays several OAuth endpoints and their URLs. The 'Basic' section includes fields for 'Display Name' (ainaba-apim-1124), 'Client ID' (127d9544-84cd-48bf-a832-71f84e8a0290), and 'Tenant ID' (f52c30c4-5504-48e5-b3d8-f22bfd4ce170). The 'Microsoft ID' section lists various Microsoft identity endpoints.

This screenshot shows the 'ID Provider Update' page for the 'ainaba-apim-1124' user. It is configured for 'Azure Active Directory' with 'Client ID' (940c8847-db9-429e-9ed7-bbccf3ecdfb2) and 'Client Secret' (redacted). Other sections include 'Redirect URLs' (https://ainaba-apim-1124.developer.azure.net, https://ainaba-apim-1124.portal.azure.net) and 'Well-known URLs'.

開発者ポータルからの API 呼び出し



控えておいたアプリ情報を開発者ポータルに登録して OAuth 2.0 を有効化

POST メソッドを使用した承認コードフローを設定していく

クライアント登録ページは構成しないので <http://localhost> などの値を設定

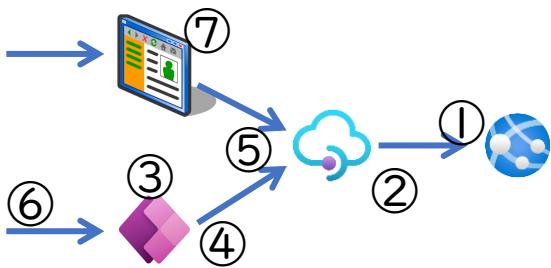
「既定のスコープ」には委任アクセス許可を設定したバックエンド API のスコープを転記

The screenshot shows three main windows illustrating the configuration process:

- Left Window (Developer Portal Overview):** Shows the navigation menu and the "OAuth 2.0 + OpenID Connect" configuration page for the "ainaba-apim-1124" service.
- Middle Window (OAuth Configuration):** Displays the "委任" (Delegated) OAuth configuration. It includes fields for "クライアント登録ページの URL" (Client registration page URL) set to "http://localhost", "承認コードの種類" (Authorization code type) checked for "承認コード" (Authorization code), and the "既定のスコープ" (Default scope) field highlighted with a red box containing the value "api://b2b6a36a-ecf4-4d9d-9a44-6d4c38254b28/user_impersonate".
- Right Window (Azure API Management API Definition):** Shows the "Scope" section of the API definition for "ainaba-apibe-1124". It lists the scope "api://b2b6a36a-ecf4-4d9d-9a44-6d4c38254b28/user_impersonate" with the status "有効" (Enabled).

A large red arrow points from the "既定のスコープ" field in the middle window to the corresponding scope entry in the right window, indicating the mapping between the developer portal configuration and the Azure API Management API definition.

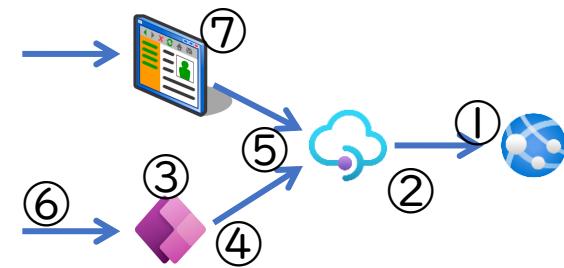
開発者ポータルからの API 呼び出し



前頁の設定で認可コード付与のリダイレクトURLが表示されているため、Azure AD アプリに追記しておく

The screenshot shows the Azure portal interface for managing an Azure AD application named 'ainaba-apim-1124'. The left sidebar has a '認証' (Authentication) section selected. In the main content area, under the 'Web' section, there is a 'リダイレクト URI' (Redirect URI) configuration. A red box highlights the first two entries: 'https://ainaba-apim-1124.developer.azure-api.net/signin' and 'https://ainaba-apim-1124.portal.azure-api.net/signin-aad'. A second red box highlights the third entry: 'https://ainaba-apim-1124.developer.azure-api.net/signin-oauth/code/callback/developer-console-oauth2'. A large orange callout points to this third URL with the text '認可コード付与フローの URL' (Authorization code flow URL). Another orange callout points to the first two URLs with the text '上2つはユーザー認証を有効化した際のもの' (The first two are for when user authentication is enabled).

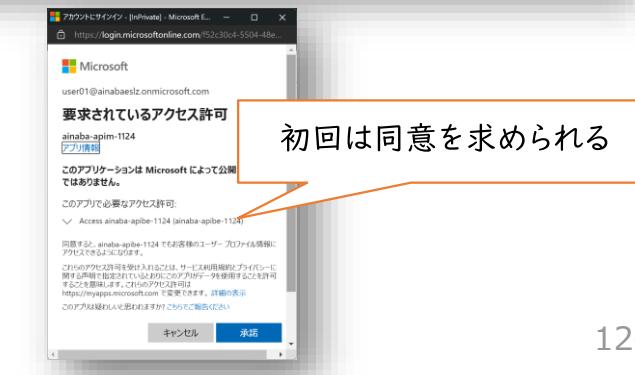
開発者ポータルからの API 呼び出し



バックエンド API の呼び出しに必要な OAuth サーバーを設定すると、開発者ポータルのテスト時にユーザーのアクセストークンを埋め込めるようになる

This screenshot shows the Azure API Management service configuration for an API named 'Todo API'. In the 'Security' section, 'User authorization' is set to 'OAuth 2.0' and the 'OAuth 2.0 server' is configured to '開発者ポータルの委任' (Delegate to developer portal). A red box highlights this configuration.

This screenshot shows the 'Todo API / v1 / Debug Info' page. The 'Authorization' header is set to 'bearer eyJ0eXAiOiJKV1QiLC...' with a red box highlighting it. The response status is '200 OK'.





Microsoft

Microsoft Confidential

- 本資料は情報提供のみを目的としており、本資料に記載されている情報は、本資料作成時点でのマイクロソフトの見解を示したもので。状況等の変化により、内容は変更される場合があります。本資料に特別条件等が提示されている場合、かかる条件等は、貴社との有効な契約を通じて決定されます。それまでは、正式に確定するものではありません。従って、本資料の記載内容とは異なる場合があります。また、本資料に記載されている価格はいずれも、別段の表記がない限り、参考価格となります。貴社の最終的な購入価格は、貴社のリセラー様により決定されます。マイクロソフトは、本資料の情報に対して明示的、黙示的または法的な、いかなる保証も行いません。

© 2020 Microsoft Corporation. All rights reserved.

Microsoft, Windows, その他本文中に登場した各製品名は、Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他、記載されている会社名および製品名は、一般に各社の商標です。