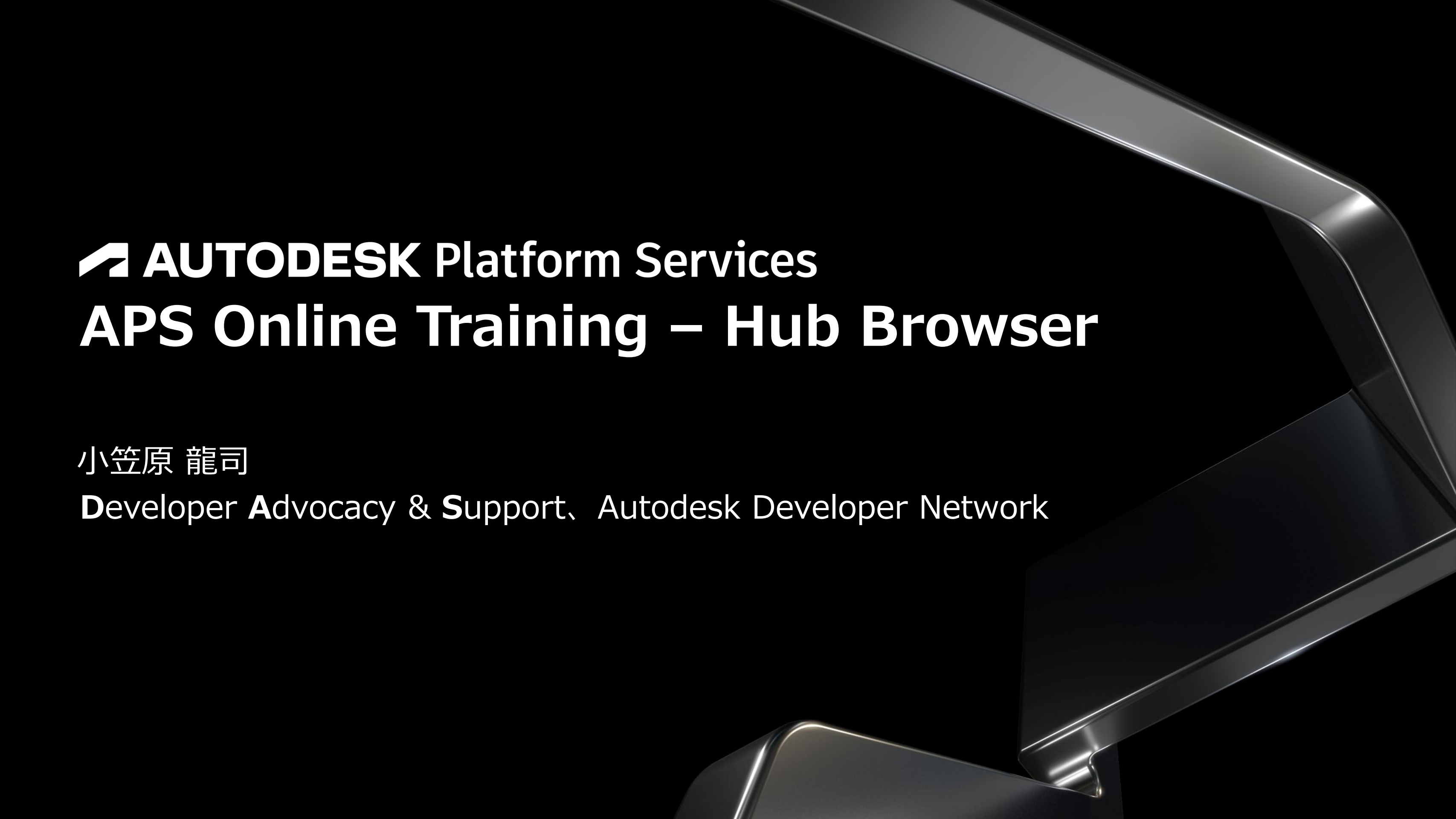




AUTODESK Platform Services **APS Online Training – Hub Browser**

小笠原 龍司

Developer **A**dvocacy & **S**upport、Autodesk Developer Network

アジェンダ – ご紹介する内容

- はじめに – Hub Browser
- デベロッパキーの取得
- ACC カスタム統合の設定
- セットアップ
- 認証と認可
- データのブラウジング
- Viewer と UI



はじめに

インターネットで各社提供 API への接続が可能な時代

- 業界標準 & オープンソース API テクノロジ

<HTML>
HTML



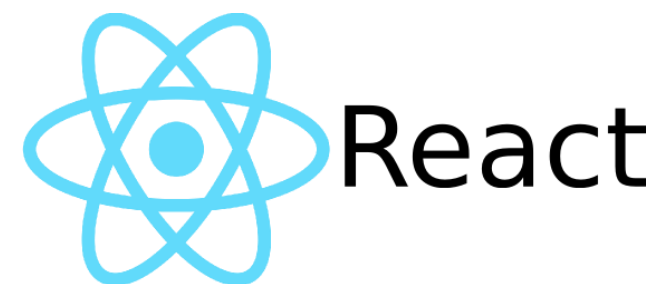
{CSS}
CSS



(JS)
JavaScript



RESTful API
GET PUT POST DELETE

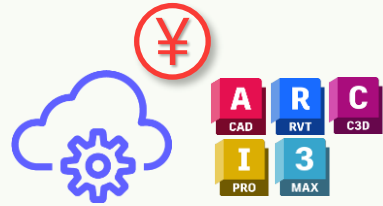


Autodesk Platform Services – 主要 API

Core APIs



Model
Derivative



Design
Automation



Viewer



Authentication



Webhooks

Data APIs



AEC
Data Model



Manufacturing
Data Model



Data
Management



Data
Exchange

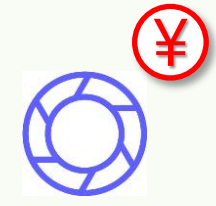
Product APIs



Forma



Fusion



Reality
Capture



ACC



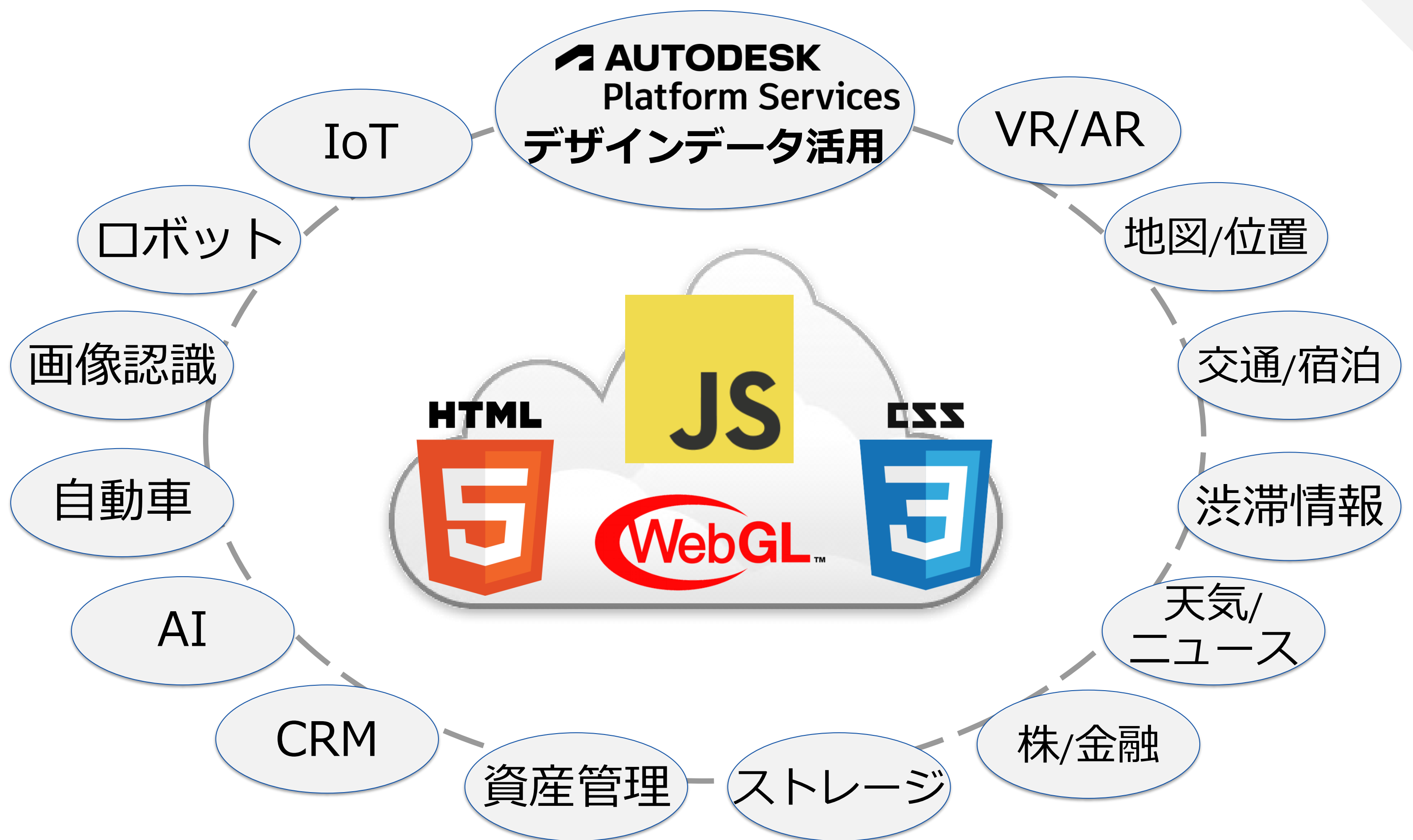
BIM 360



Tandem

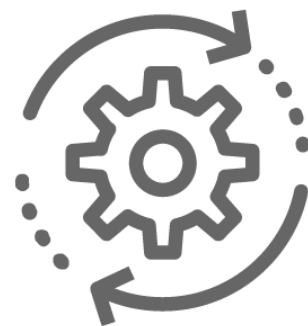
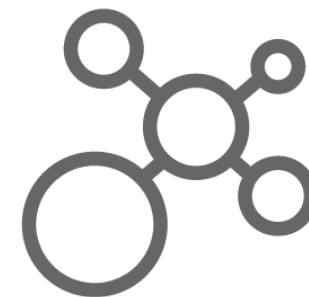
RESTful API
GET PUT POST DELETE





デスクトップ製品のアドイン開発者の方へ

- デスクトップ開発と Web 開発の違いをご理解ください
 - Web アプリの特性 – セキュリティ視点
 - 通信経路
 - 呼び出し数制限
 - 非同期処理
 - 仮想環境
 - オンプレミスとの違い

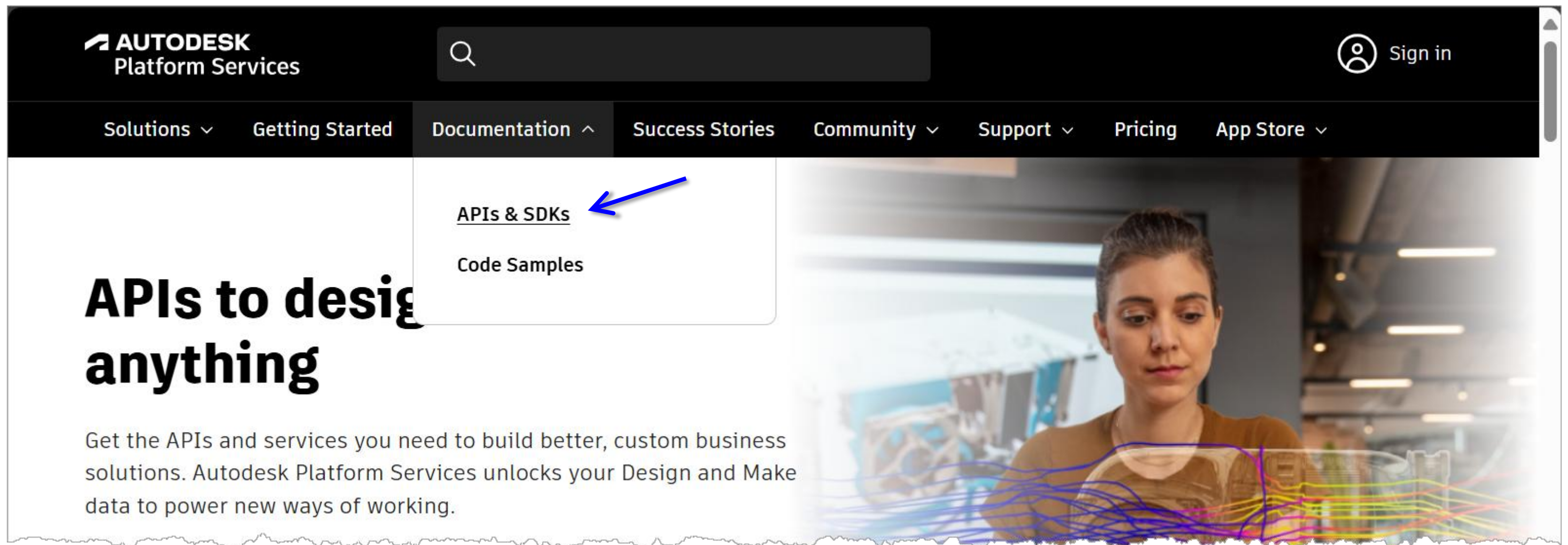


関連ブログ記事

公式ドキュメント（英語のみ）

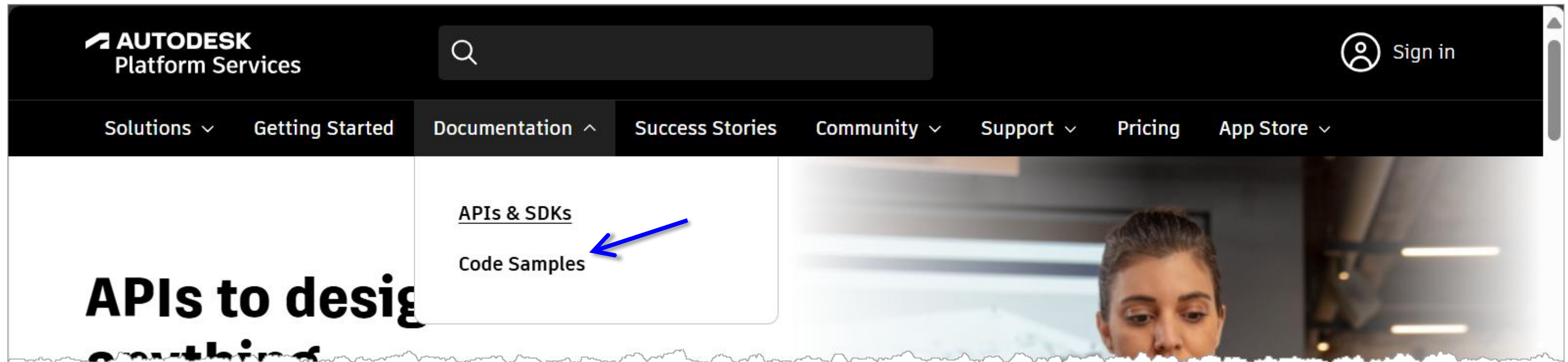
<https://aps.autodesk.com/developer/documentation>

- APS ポータル (<https://aps.autodesk.com/>) に記載



APS サンプル – GitHub リポジトリ

<https://aps.autodesk.com/code-samples>



 AUTODESK	Solutions	Documentation	Resources	About
Company overview	AEC Data Model	AEC Data Model	Get Help	About APS
Careers	Autodesk Construction Cloud	Authentication	API Status	Pricing
Investor relations	Autodesk Fusion	Autodesk Construction Cloud	Blog	Success Stories
Newsroom	BIM 360	BIM 360	FAQ	Certified Partners
	BuildingConnected API	Data Exchange	Code Samples	Partnerships
			ADN Member Sign-in	

学習リソース : Learn APS Tutorial (英語のみ)

<https://get-started.aps.autodesk.com/>

Get the APIs and services you need to build better, custom business solutions. Autodesk Platform Services unlocks your Design and Make data to power new ways of working.

Sign up to try Autodesk Platform Services



APS Tuesdays

Join a free weekly webinar to get to know APS



Learn APS Tutorial

Step by step guide on GitHub

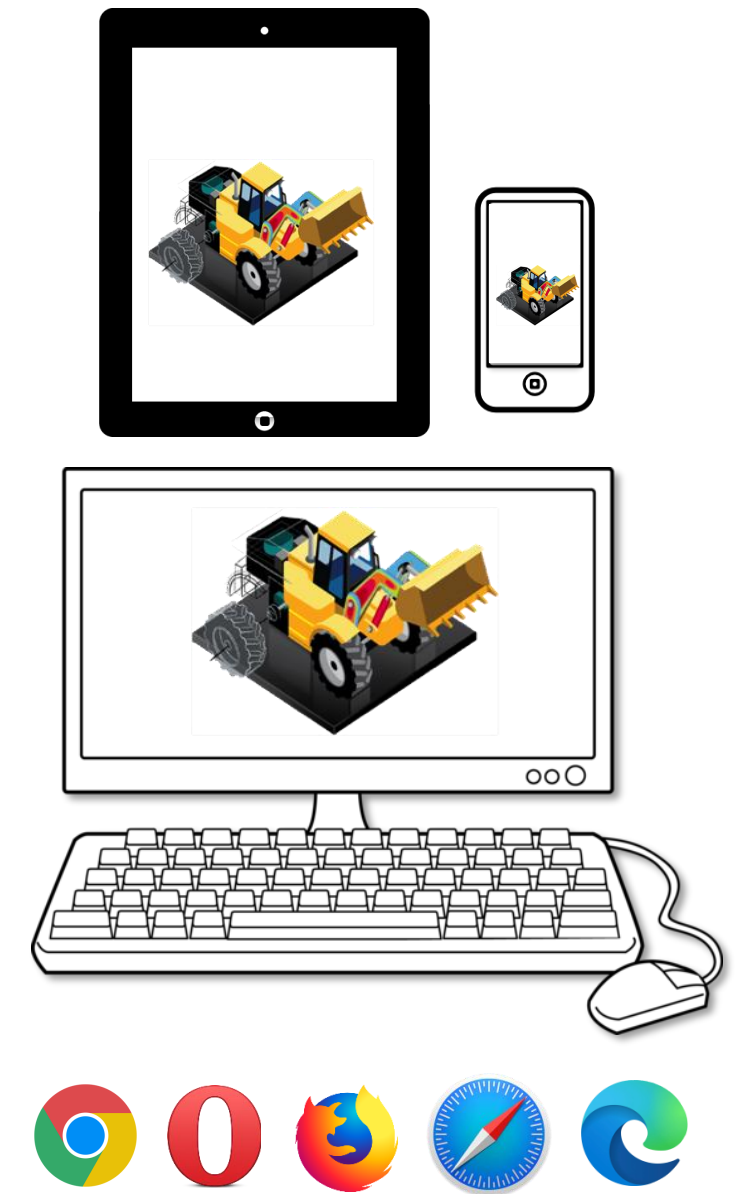
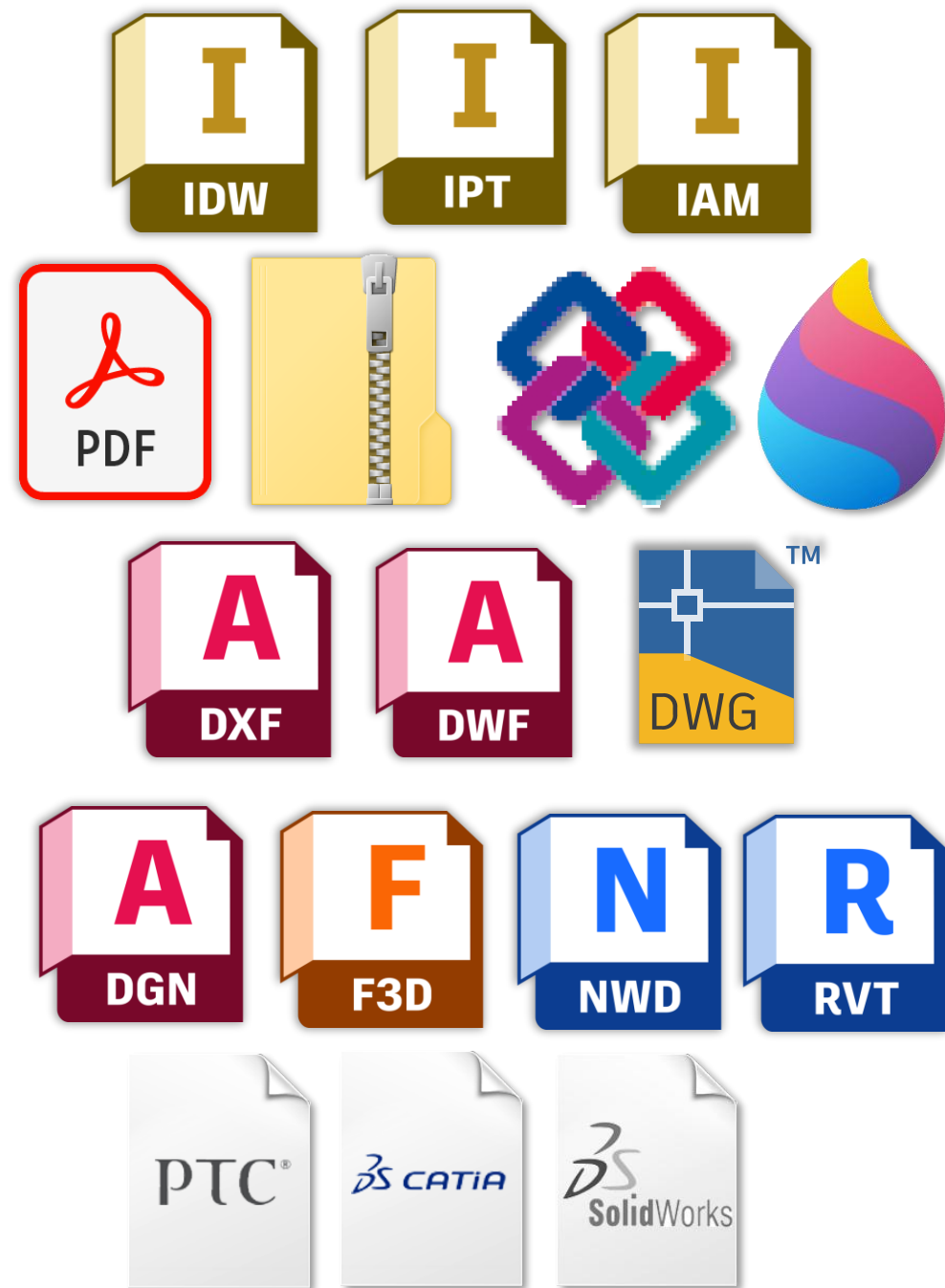


Developer Newsletter

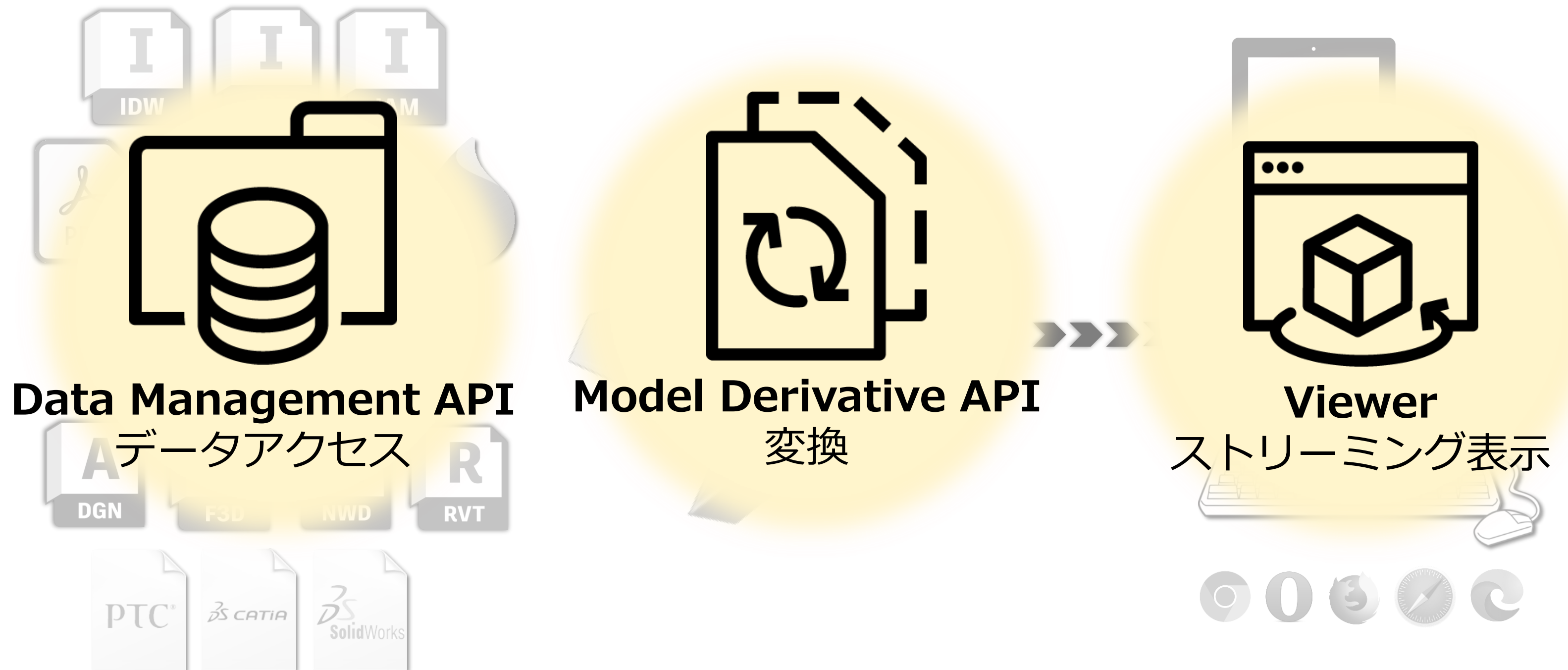
Stay up to date with the developer community



Viewer ソリューション

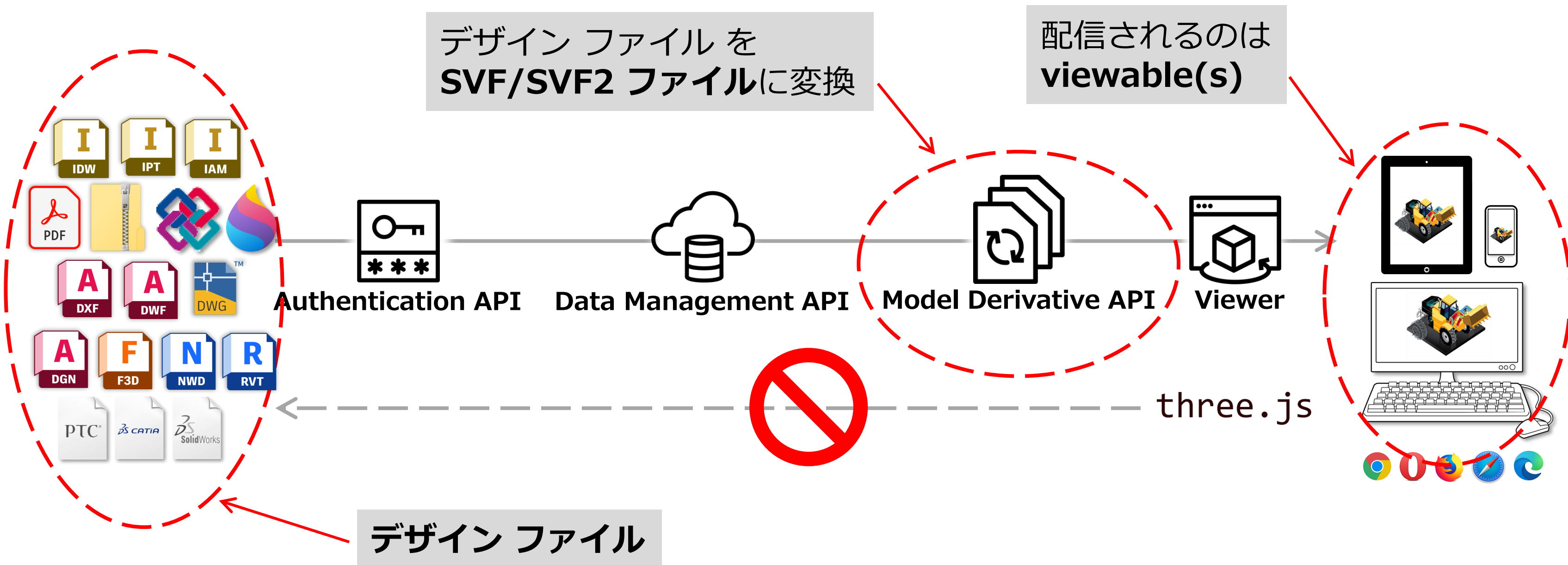


Viewer ソリューション (OSS)

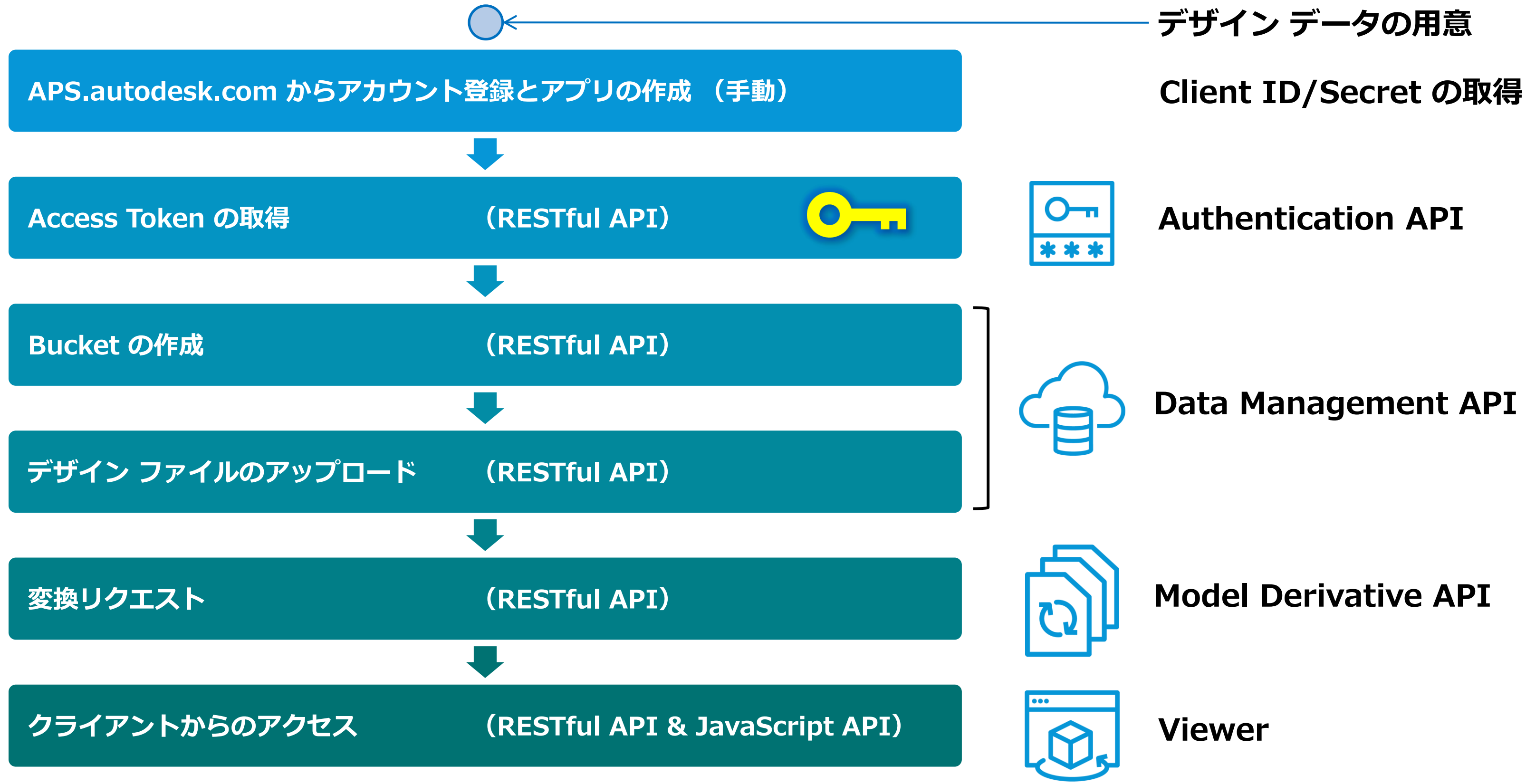


APS Viewer ソリューションのながれ (OSS)

- APS Viewer からのオリジナルデータへの反映は不可

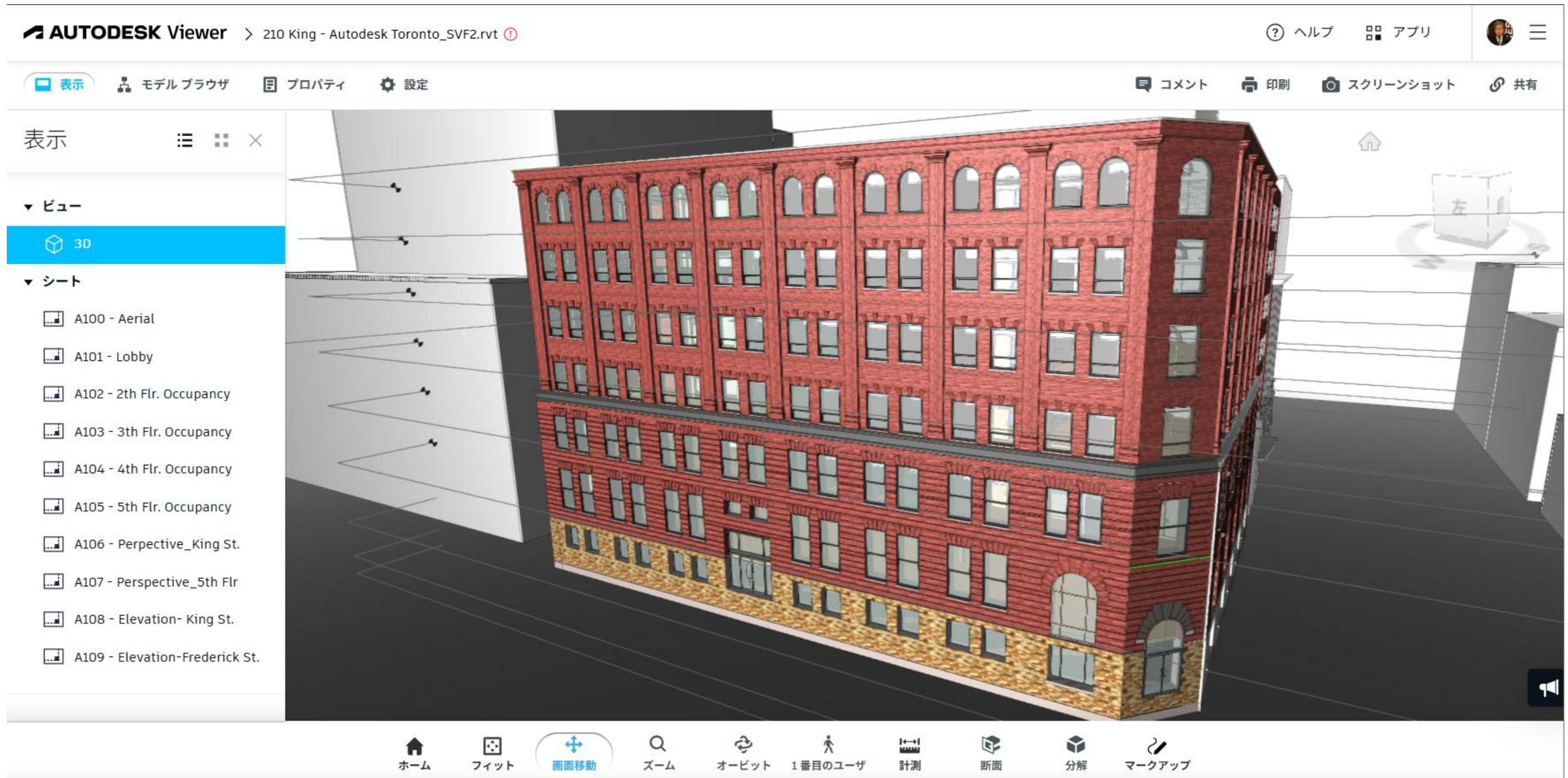


2-legged OAuthでの Viewer 利用手順の理解

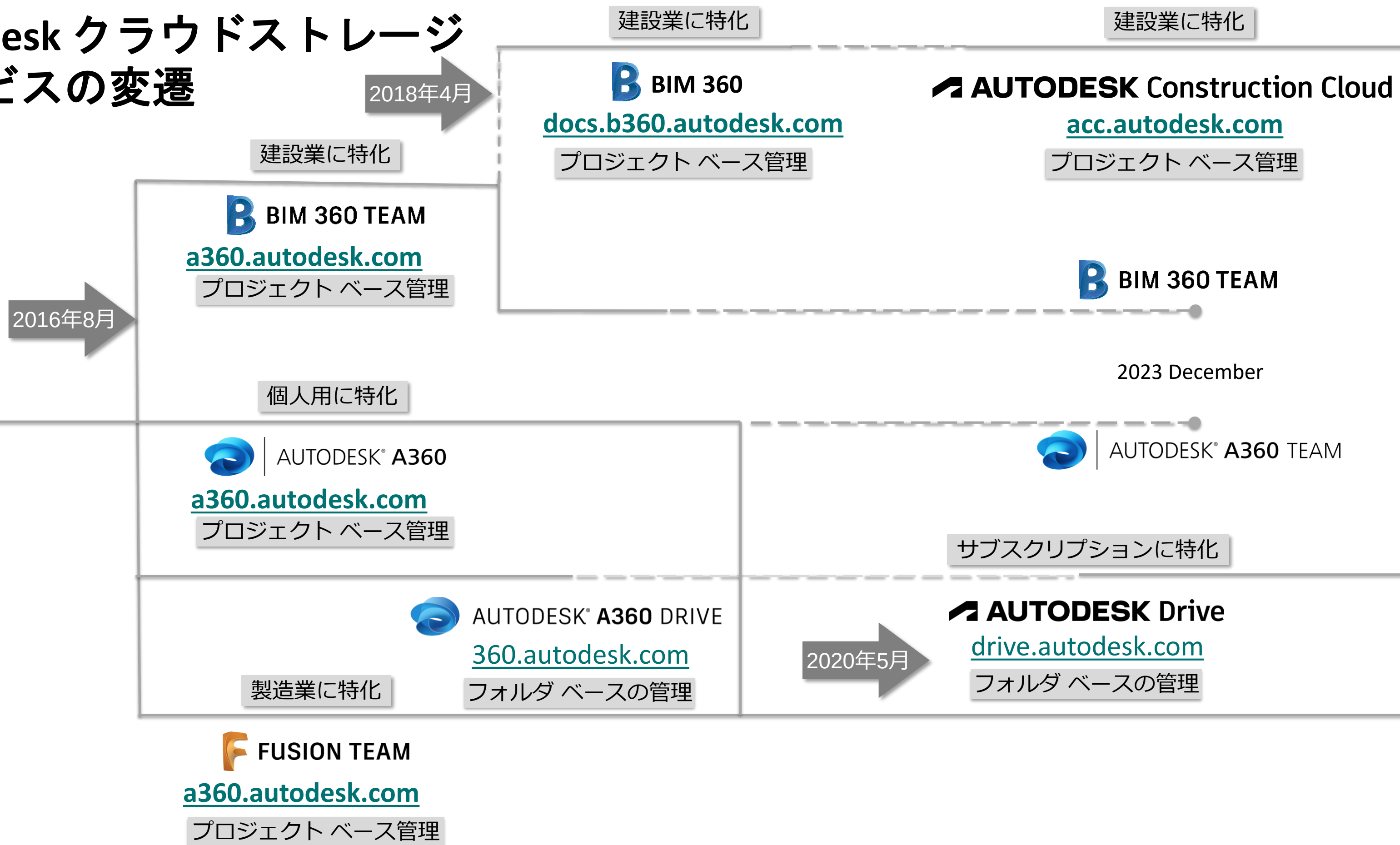


Autodesk Viewer

<https://viewer.autodesk.com/>



Autodesk クラウドストレージ サービスの変遷

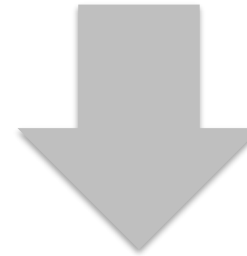


B AUTODESK®
BIM 360® COORDINATE

B AUTODESK®
BIM 360® DESIGN

B AUTODESK®
BIM 360® BUILD

B AUTODESK®
BIM 360™ DOCS



C AUTODESK
BIM Collaborate

T AUTODESK
Takeoff

B AUTODESK
Build

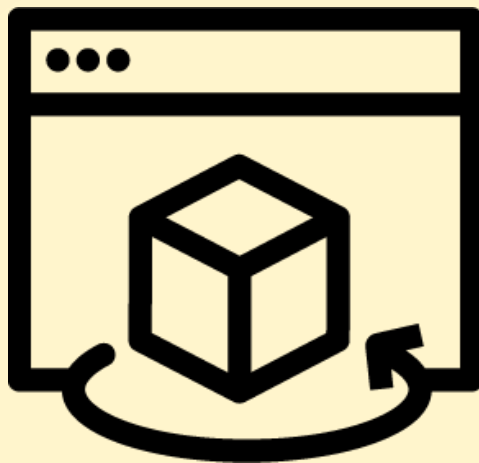
D AUTODESK®
DOCS
DOC

 **AUTODESK** Construction Cloud

ストレージ統合ソリューション (ACC)



ACC API
ACC 固有機能



Viewer
ストリーミング表示



Data Management API
データアクセス

ACC モジュール（機能）別 API

Account Admin

Automate project setup, managing member and partner directories, and syncing data with external systems.

[→ API reference](#)

Assets

Create, manage, and search assets, improving their organization and workflow efficiency within Autodesk Construction Cloud Assets.

[→ API reference](#)

AutoSpecs

Access draft submittal logs, extracted from construction project specification documents.

[→ API reference](#)

Cost Management

Manage budget and cost changes in Autodesk Construction Cloud by accessing, exporting, and importing data to and from external systems.

[→ API reference](#)

Data Connector

Retrieve data from several Autodesk Construction Cloud services for local analysis, reporting, and business intelligence.

[→ API reference](#)

Files

Access, upload, and share 2D plans, 3D intelligent models, and any other project documents to maximize collaboration.

[→ API reference](#)

Forms

Access data stored in the Forms module, which enables your team to securely fill out, review, and manage project forms.

[→ API reference](#)

Issues

Create and update project-associated issues in Autodesk Construction Cloud for tracking and management of tasks and problems.

[→ API reference](#)

Locations

Configure a hierarchy of building areas in projects, to identify the location of Assets, Issues, Photos, Forms, and more.

[→ API reference](#)

Model Coordination

Manage issues when combining 3D models from different design disciplines into unified project coordination spaces.

[→ API reference](#)

Model Properties

Build and query indexes from properties in intelligent models uploaded to Autodesk and BIM 360 Docs, and calculate changes (diffs) between consecutive model versions.

[→ API reference](#)

Photos

Access data stored in the Photos module, a single unified place to view and manage photos and videos in Autodesk Construction Cloud.

[→ API reference](#)

Relationships

Easily create, retrieve, and delete links between entities across all domains in Autodesk Construction Cloud.

[→ API reference](#)

RFIs

Create, track, and update RFIs, assigning members, transitioning states, and adding comments.

[→ API reference](#)

Takeoff

Retrieve and update takeoff project components like settings, packages, classifications, and content views.

[→ API reference](#)

Sheets

Manage and distribute sheets for field use. You can upload, publish, and organize sheet versions and sets.

[→ API reference](#)

Submittals

Access the submittal items a user has permission to view in a project.

[→ API reference](#)

ACC : Hub(Account), Project

Hub/Account

Projects

プロジェクト

アクティブ アーカイブ済み

+ プロジェクトを作成

名前	番号	プラットフォーム	メンバー	会社	ステータス	使用するテン
Ryuji ACC 2022 Test Project			1	1	アクティブ	-
Ryuji ACC 2023 Admin API Test Project	HP-0002		0	1	アクティブ	-
Ryuji ACC 2023 Test Project			2	1	アクティブ	-
Ryuji ACC 2024-2 Admin API Test with T...	HP-0002		0	0	アクティブ	Ryuji ACC 20
Ryuji ACC 2024 Admin API Test Project	HP-0002		1	1	アクティブ	-
Ryuji ACC 2024 Admin API Test with Te...	HP-0002		0	0	アクティブ	Ryuji ACC 20
Ryuji ACC 2024 Test Project			1	1	アクティブ	-
Ryuji ACC 2025 Test Project			2	1	アクティブ	-
Ryuji ACC Design Collaboration Devit 2			1	1	アクティブ	-

1 ~ 19/19 の表示中

Autodesk Docs : Folder, Item

The screenshot shows the Autodesk Docs interface for a project named "Ryuji ACC 2025 Test Project". The interface is divided into several sections:

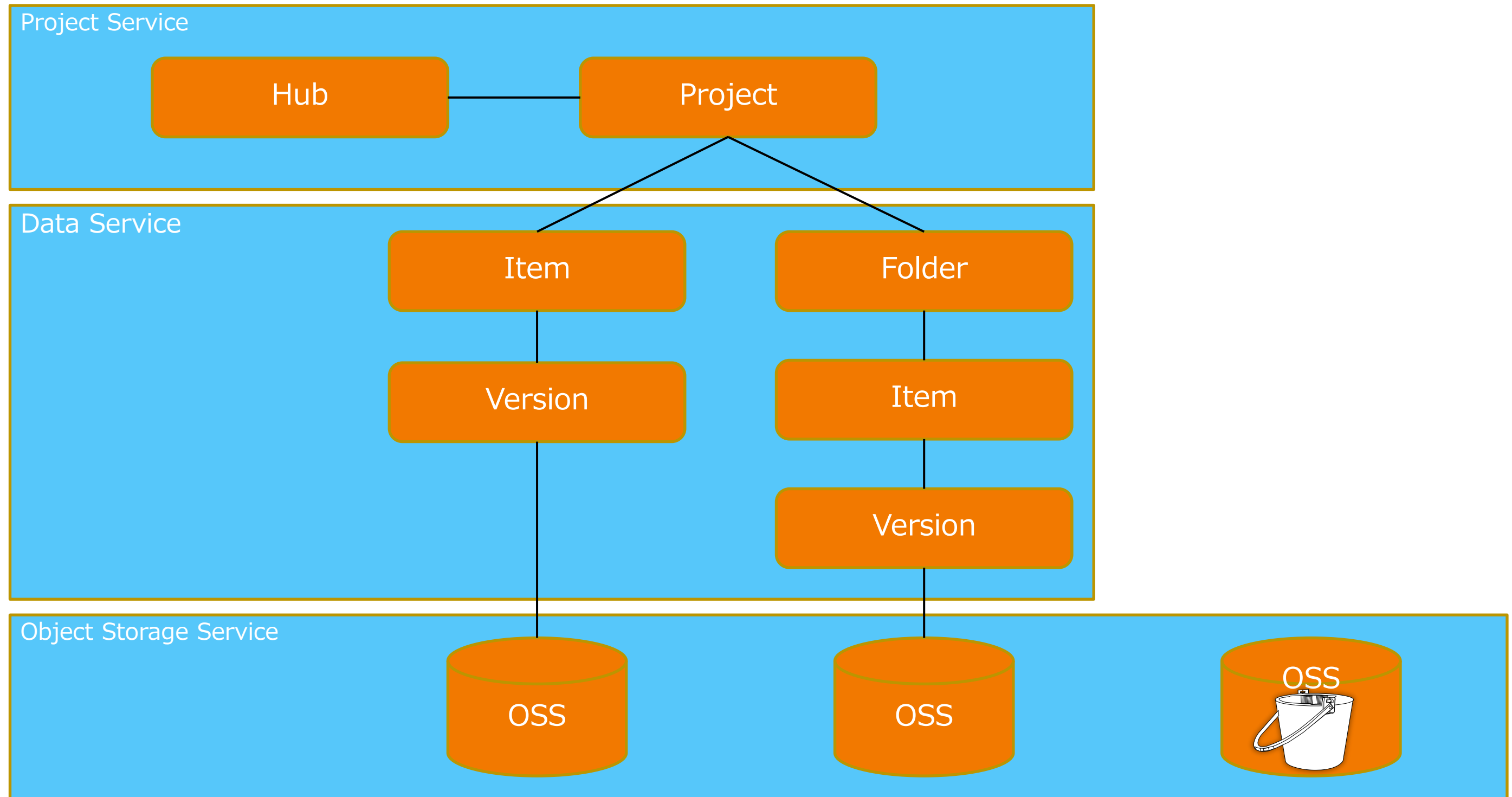
- Left Sidebar:** Contains navigation links for "Docs", "ファイル" (Files), "仕様" (Specifications), "レビュー" (Reviews), "ファイル転送" (File Transfer), "指摘事項" (Issues), "ブリッジ" (Bridge), and "設定" (Settings).
- Top Bar:** Displays the project name "Ryuji ACC 2025 Test Project" and a user profile icon.
- Main Content Area:** Shows a list of files under the "プロジェクト ファイル" (Project Files) folder. The list includes a file named "サンプル意匠.rvt" (Sample Design.rvt) with a version "V1".

Annotations with arrows point to specific elements:

- Autodesk Docs:** Points to the "Docs" link in the left sidebar.
- [プロジェクト ファイル]フォルダ (Folder):** Points to the "プロジェクト ファイル" folder in the main content area.
- ユーザー作成フォルダ (Folder):** Points to the "意匠" (Design) folder in the main content area.
- アイテム/ファイル (Item):** Points to the "サンプル意匠.rvt" file in the list.
- アイテム バージョン (Version):** Points to the "V1" version label next to the file.

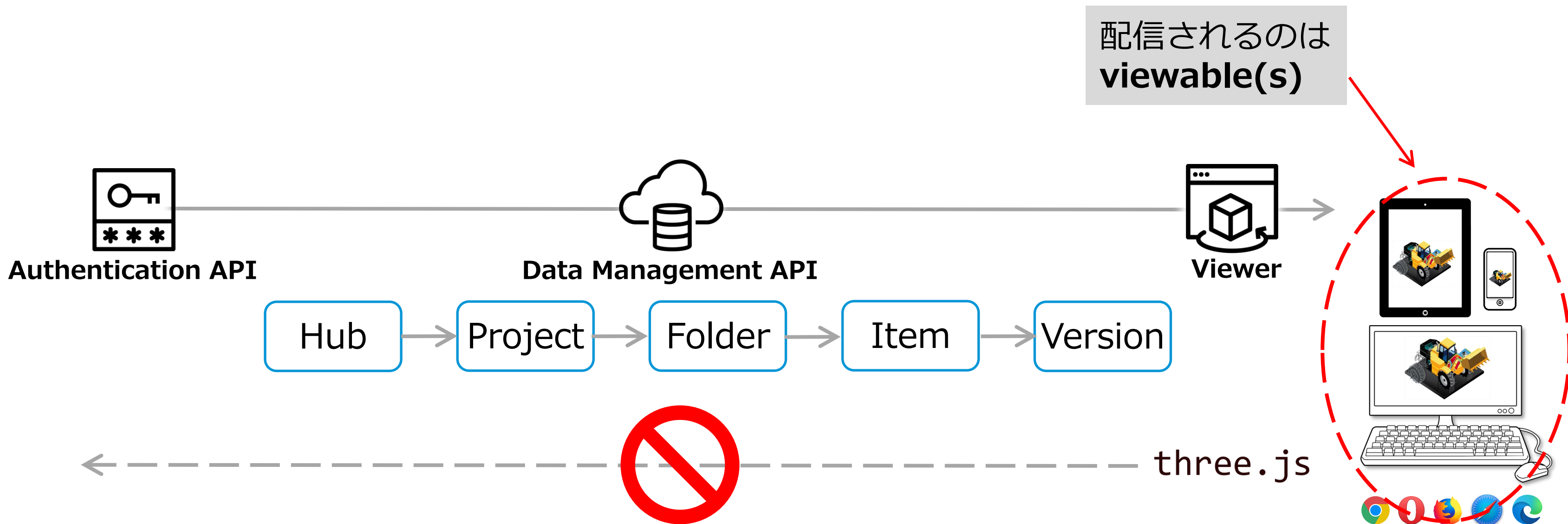
At the bottom of the interface, it says "1 個の項目を表示中" (Showing 1 item).

Autodesk クラウドストレージ サービスの構造

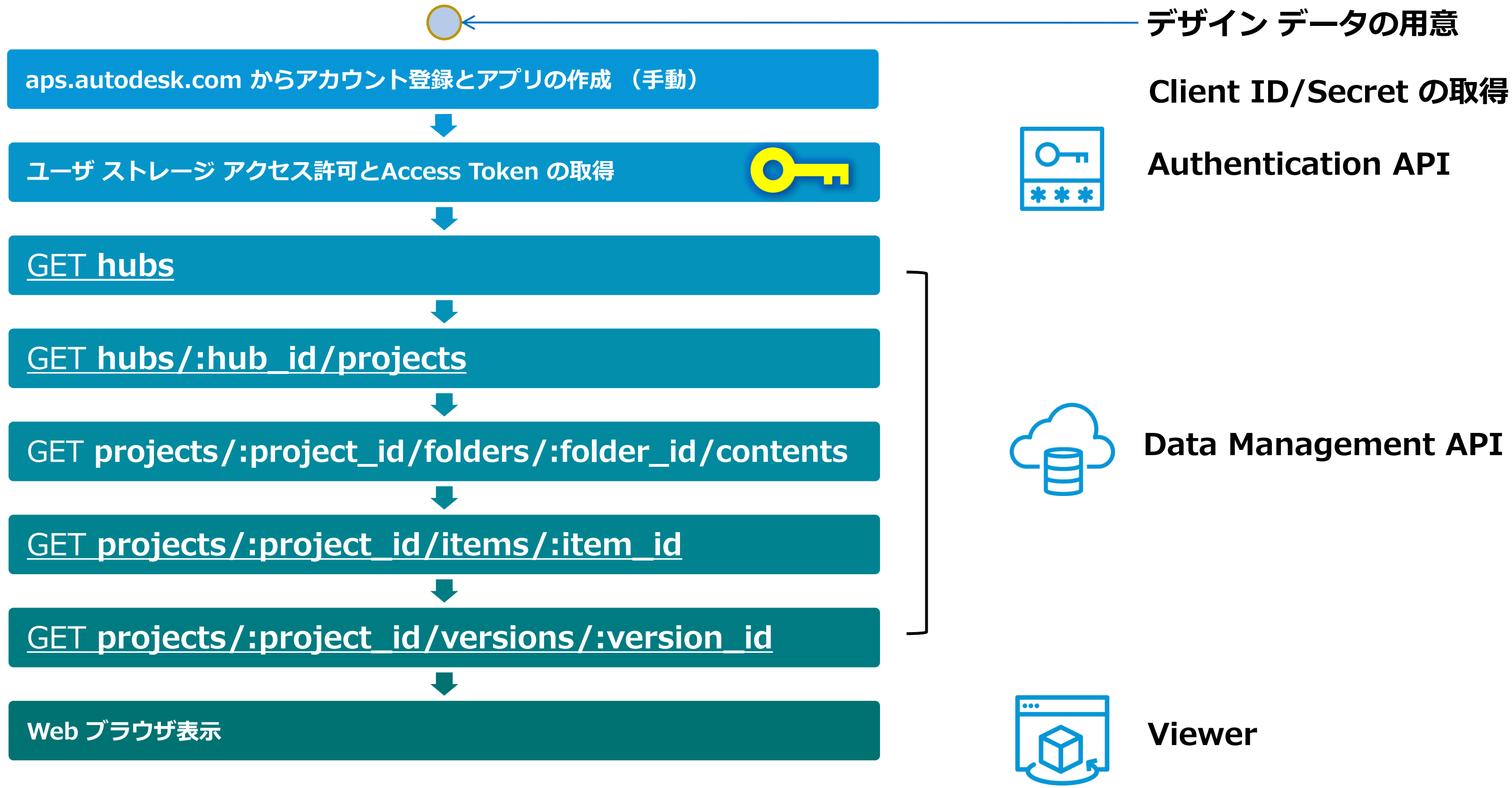


APS Viewer ソリューションの流れ (Hub)

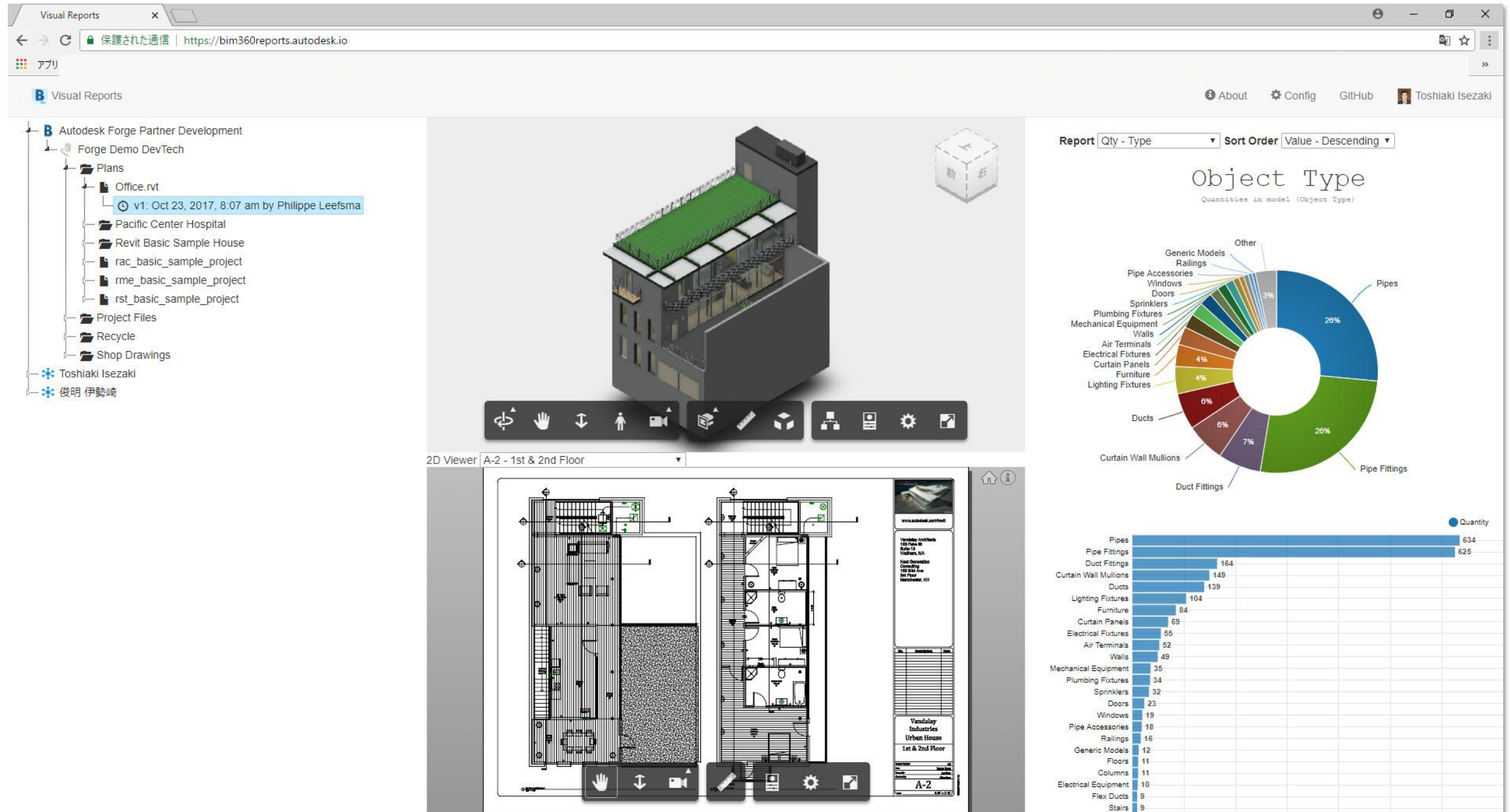
- ACC などクラウドサービスは自動的に変換を実行→変換結果を利用可
 - Model Derivative API で変換する必要はありません。



3-legged OAuthでの Viewer 利用手順の理解



何ができるか? – ACC/BIM 360 統合 ダッシュボード



APS SDK : Hub Browser でも利用

- エンドポイントをラップするサーバー用途パッケージ
 - エンドポイントをラップ
 - SDKManager と API 毎に分割したパッケージで提供
 - .NET と Node.js (TypeScript) を用意
- **.NET SDK**
 - NuGet パッケージとして提供
 - <https://www.nuget.org/profiles/AutodeskPlatformServices.SDK>
- **Node.js SDK (TypeScript SDK)**
 - Node.js パッケージとして提供
 - <https://www.npmjs.com/~aps.sdk>



SDK ドキュメント

28



Developer's Guide

Overview

API Basics

Field Guide

Pagination

Filtering

Retention Policies

Commands

▸ Rate Limits

How-to Guide

Reference Guide

▸ REST API Reference

▸ .NET SDK Reference (OSS)

▸ .NET SDK Reference (Data Management)

▸ TypeScript SDK Reference (OSS)

▸ TypeScript SDK Reference (Data Management)

Documentation / Data Management API / Developer's Guide

Data Management API

▸ REST API Reference

▸ .NET SDK Reference (OSS)

▸ .NET SDK Reference (Data Management)

▸ TypeScript SDK Reference (OSS)

▸ TypeScript SDK Reference (Data Management)

multiple versions.

Schema Service

This allows your application to understand the structure and semantics of the data.

Object Storage Service (OSS)

This allows your application to download and upload raw files (such as images, videos, etc.)

エンドポイント リファレンス

.NET SDK リファレンス

Node.js リファレンス

製品機能が全てカバーされているわけではありません

- ACC 自身を拡張するカスタマイズは出来ません
 - Web アプリに ACC/BIM 360 機能を組み込み/利用が可能
- モジュール毎にサポートされている API は異なります
- 理由：さまざま考察が必要
 - 例)
 - 公開 API 利用時のパフォーマンス低下を抑止する必要あり
 - 1. 機能により多くの API（通信）呼び出しが集中する可能性
 - 2. ACC 自体の処理速度や応答が低下してしまう懸念
 - 3. 呼び出し数制限の設定を検討...
 - 4. API の変更、廃止、新規追加などに随時対応が必要

例) [ACC Project Admin API: プロジェクト作成とユーザー管理](#)





デベロッパキーの取得

APS を始めるには?

- まずは <https://aps.autodesk.com/> へ

1



2



3

Autodesk ID で
サインイン

APS アプリ作成

開発

- 目的はデベロッパ キーの取得
 - **Client ID** (別名 : Consumer Key)
 - **Client Secret** (別名 : Consumer Secret)

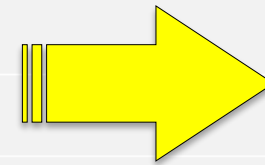
My applications からのアプリ登録

どのAPIを使うか指定

API Access

These are the APIs your app is allowed to access. It can be changed anytime.

- ☒ AEC Data Model API
- ☒ Application Management API
- ☒ Autodesk Construction Cloud API
- ☒ BIM 360 API
- ☒ BuildingConnected API
- ☒ Data Exchange API
- ☒ Data Management API
- ☒ Design Automation API
- ☒ Flow Graph Engine API
- ☒ Manufacturing Data Model API
- ☒ Model Derivative API
- ☒ Parameters API
- ☒ Premium Reporting API
- ☒ Reality Capture API
- ☒ Tandem Data API
- ☒ Token Flex Usage Data API
- ☒ Webhooks API



My applications からのアプリ登録

課金対象のチーム

Team Assignment

Assign an [Autodesk team](#) to pay for any charges incurred by this application

Toshiaki Isezaki - 2013



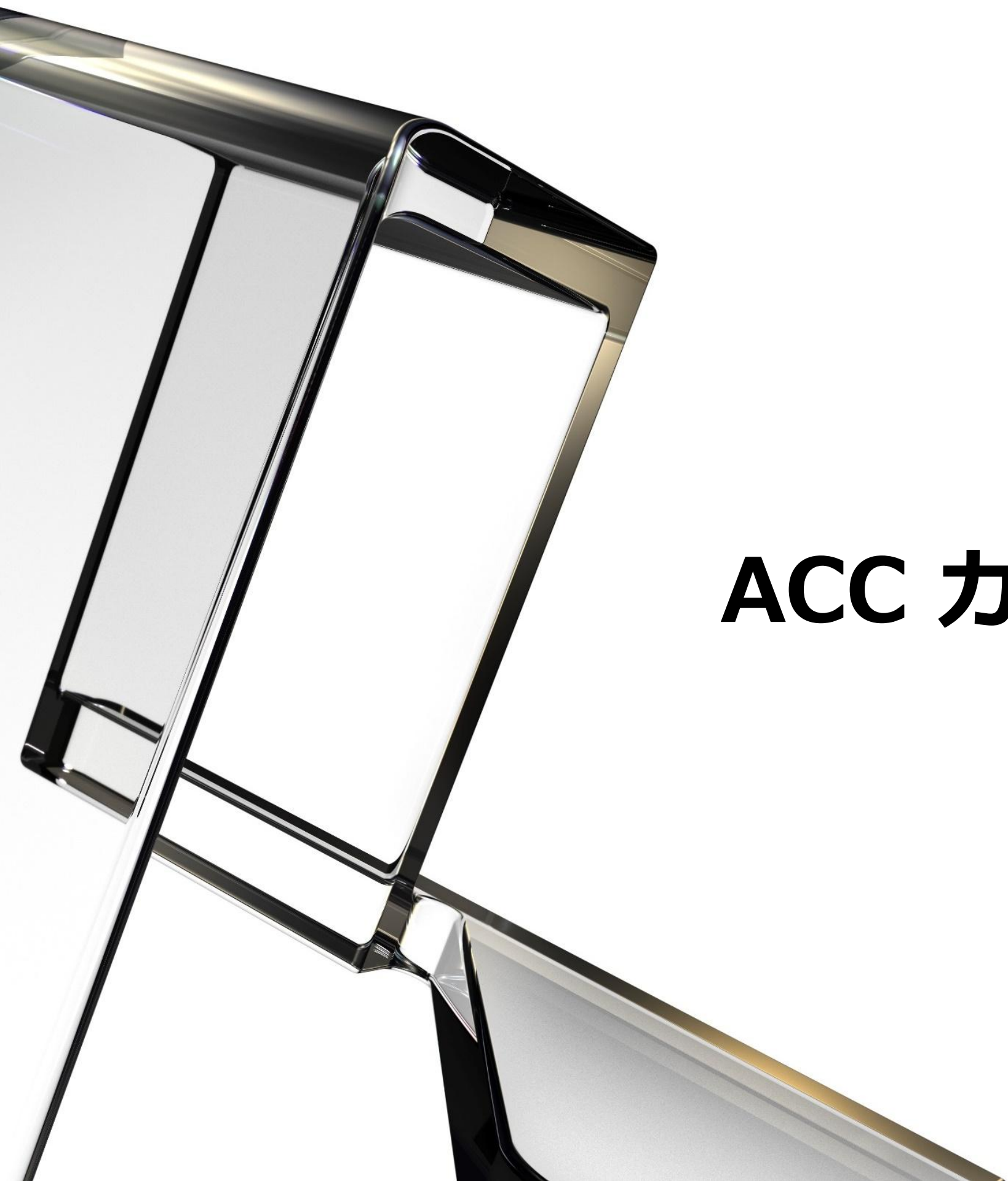
Assign

アカウントで複数のチームを編成している場合には、プライマリ管理者、または、セカンダリ管理者に適切なチームを確認してください。通常はアカウントに 1 つのチームしか関連付けられていないので、表示されるチームを割り当て（Assign）ます。

3-legged 認証利用時の必須項目

- Callback URL
 - 今回のチュートリアルでは、`http://localhost:8080/api/auth/callback`

The screenshot shows the Autodesk Platform Services configuration interface. The top navigation bar includes the Autodesk logo and links for Solutions, Getting Started, Documentation, Success Stories, Community, Support, Pricing, and App Store. The main content area displays configuration options for an application, including Application Type (Traditional Web App), Grant Type, and Authorization Code and Client Credentials. The Callback URL field is highlighted with a red box and contains the value `http://localhost:8080/api/auth/callback`. Below this field is a button labeled '+ Add URL'.



ACC カスタム統合の設定

ACC 統合は「カスタム統合」処理が前提

- APS アプリの ACC アクセスを許可する設定



- Account Admin アカウントによる事前設定が必須
- BIM 360 時代から存在



AUTODESK Construction Cloud

Account Admin ▾ AEC DM Developer Advocacy Support ▾

プロジェクト

メンバー

会社

役割

プロジェクトテンプレ...

ライブラリ

アプリ

カスタム統合

設定

BIM 360 管理者

プロジェクト

アクティブ アーカイブ済み

+プロジェクトを作成 ⓘ

名前	番号	タイプ	メンバー
AEC DM Bootcamp Project	012024		11
AEC DM DAS Test Project	-		9
Construction : Sample Project - Seaport Civic Ce...	-		2
DAS Japan Project	007		2
DAS Project	-		5



カスタム統合 ⓘ

+ カスタム統合機能を追加



名前	電子メール	ステータス	追加日	
AEC Data Model JP test - Toshiaki	[REDACTED]	アクティブ	2023/11/21	⋮
AEC Data Model Sample Zhong ZkKbTg...	[REDACTED]	アクティブ	2023/11/28	⋮
AEC Data Model Test -Carol	[REDACTED]	アクティブ	2023/11/23	⋮
AEC Data Model Test Ryuji	[REDACTED]	アクティブ	2023/12/05	⋮
AEC DM Explorer APP	[REDACTED]	アクティブ	2023/11/10	⋮

カスタム統合機能を追加



アカウント ID



統合から API にアクセスする場合は、アカウント ID が必要です。続行する前にアカウント ID を確認してください。後でアカウント ID を参照することもできます [アカウント設定](#)。

Autodesk Platform Services クライアント ID * ⓘ

Autodesk Platform Services クライアント ID

<<Client ID>>

カスタム統合名 *

カスタム統合名を入力

説明

カスタム統合機能の説明を入力

0/255

キャンセル

追加



セッティング

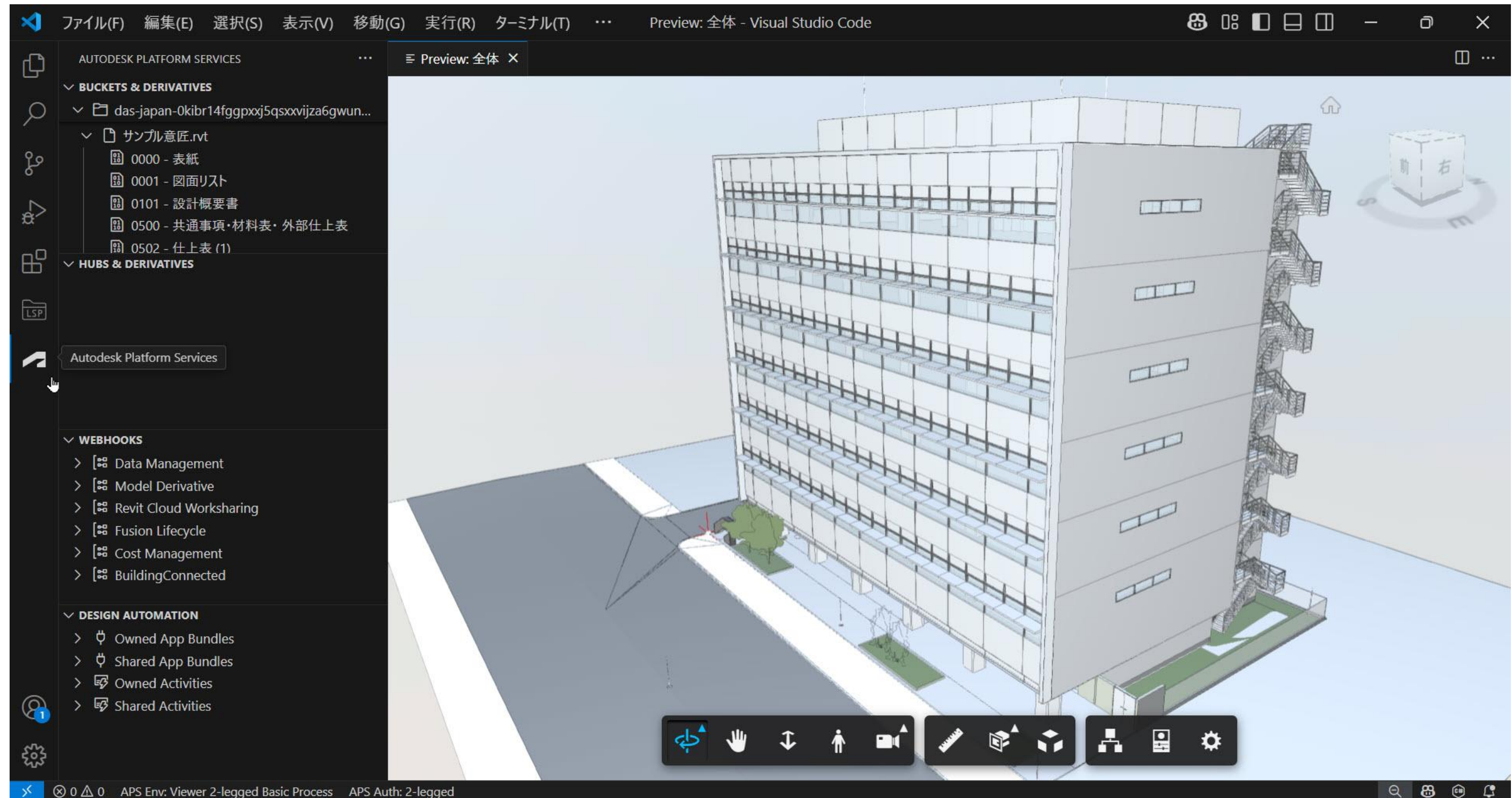
今回使用する開発環境

これに限定するものではありません

- Web ブラウザ : **Google Chrome**
- HTML/JavaScript エディタ : **VS Code**
- Web サーバー実装 : **Node.js**
- テスト ツール : **Postman**
- リポジトリ ユーティリティ : **git for Windows**



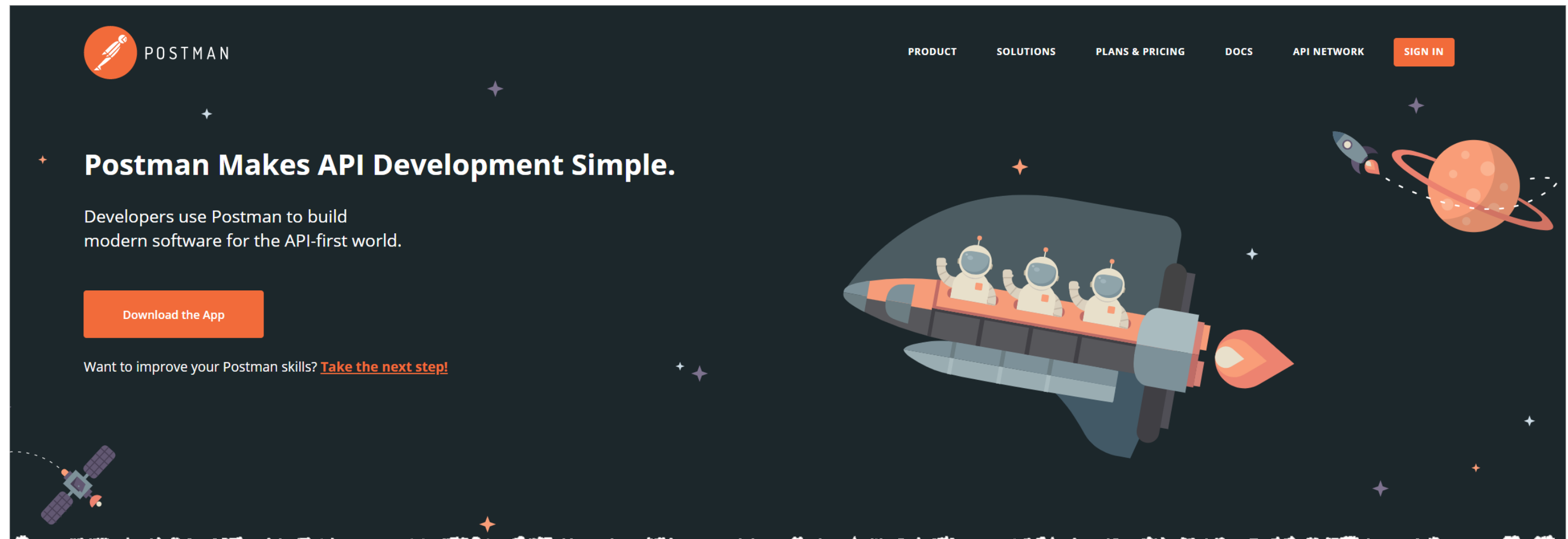
VS Code - APS エクステンション



RESTful API のテストツール



- Postman
 - <https://www.getpostman.com/>
 - コードを書かずに RESTful API のテストが可能



 **AUTODESK**
Platform Services

[Solutions](#) [Getting Started](#) [Documentation](#) [Success Stories](#) [Community](#) [Support](#) [Pricing](#) [App Store](#)

[Getting Started](#)
[Environment Setup](#)
[Tutorials](#)
[Simple Viewer](#)
[Hubs Browser](#)
[Application Setup](#)
[Authentication](#)
[Data Browsing](#)
[Viewer & UI](#)
[Dashboard](#)
[Design Automation](#)
[ACC Administrator](#)
[Learn More](#)

[Home](#) > [Tutorials](#) > [Hubs Browser](#) > **Application Setup**

Application Setup

Let's start by creating a new project, installing required dependencies, and setting up a basic server application.

Create project

Node.js & VSCode

.NET & VSCode

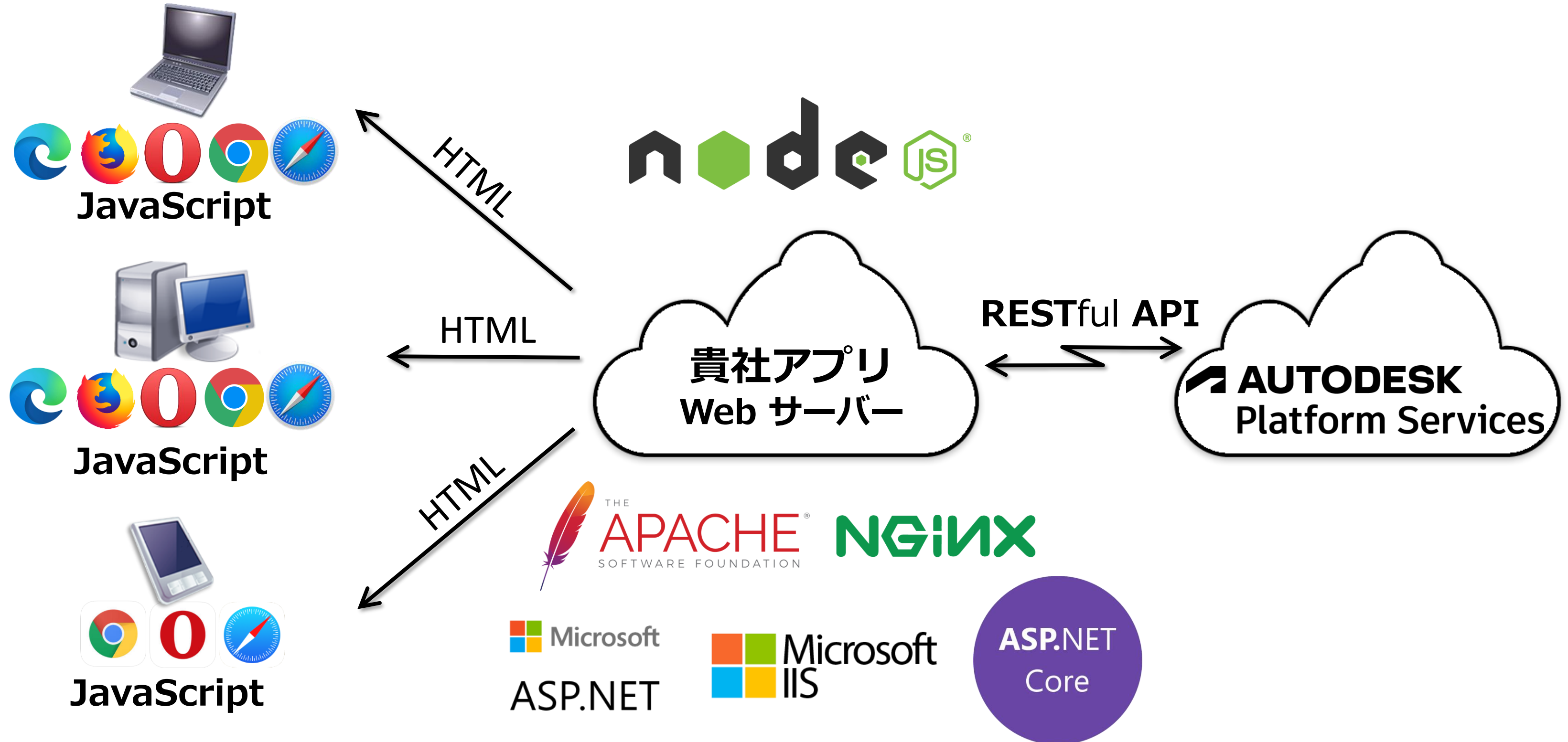
.NET & VS2022

Create a new folder for your project, navigate to it in the command line, and initialize a new Node.js project:

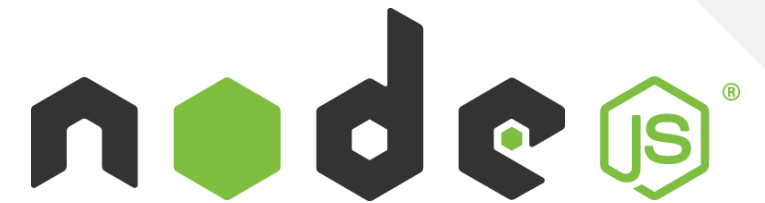
```
npm init -y
```

Next, install all the Node.js dependencies we're going to use. In this case it will be [dotenv](#) (utility for

一般的な Web アプリの構成



Node.js とは



- オープンソース
- JavaScript をサーバー上で実行するための環境
- Google V8 JavaScript Engine が使用
- Node Package Manager で拡張可能
- 今回使用しているパッケージ(ミドルウェア) :
 - @aps_sdk/...
 - dotenv
 - express
 - cookie-session

チュートリアル: VS Code 上の操作手順

1. プロジェクト フォルダの作成、VS Code でフォルダを開く
2. VS Code のターミナルで Node.js プロジェクトを初期化
3. ターミナルでAPS Node.js SDK パッケージのインストール
4. **package.json** ファイル内 “**dependencies**” の記述追記
5. **wwwroot**、**routes**、**services** サブフォルダの作成
6. **config.js** ファイルの作成、実装
7. **.env** ファイルの作成と Client ID/Secret の指定
8. **.vscode** フォルダ下に **launch.json** ファイルを作成、実装
9. **server.js** ファイルの作成とサーバー実装記述の追加
10. **package.json** ファイル内の “**scripts**” の記述変更
11. デバッグ実行 >> localhost:8080

Application Setup

Let's start by creating a new project, installing required dependencies, and setting up a basic server application.

Create project

- Node.js & VSCode
- .NET & VSCode
- .NET & VS2022

Create a new folder for your project, navigate to it in the command line, and initialize a new Node.js project:

```
npm init -y
```

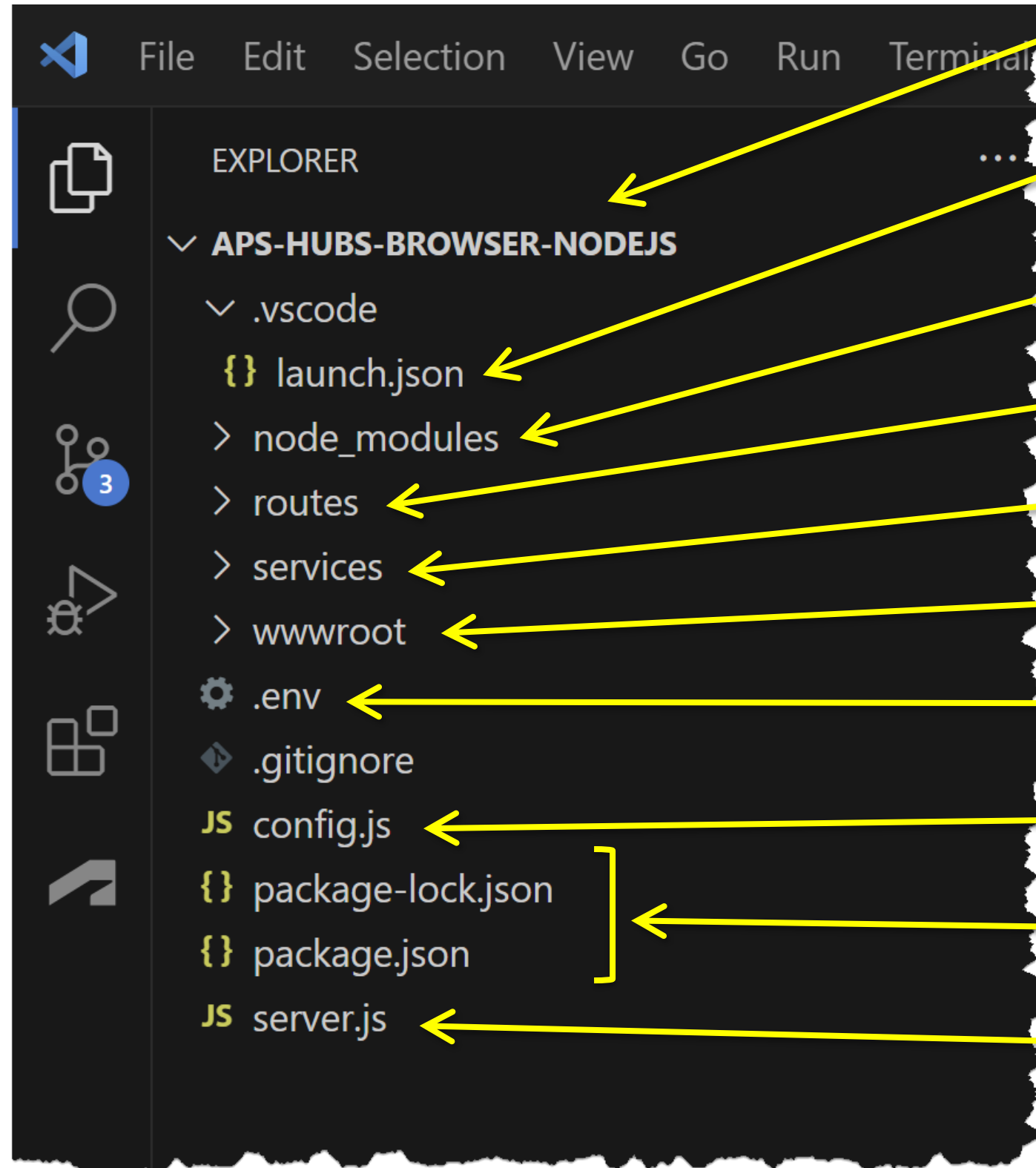
Next, install all the Node.js dependencies we're going to use. In this case it will be [dotenv](#) (utility for initializing environment variables from a file), [Express.js](#) (web framework), and [cookie-session](#) for handling cookie-based sessions, and finally the required APS SDK modules:

```
npm install --save dotenv express cookie-session
npm install --save @aps_sdk/authentication @aps_sdk/data-management
```

The "dependencies" in your `package.json` file should now look something like this (potentially with slightly

- Create project
- Application config
- Create basic server
- Try it out

ここまでのプロジェクト構成



プロジェクト フォルダを
Hub Browserとした場合

VS Code デバッグで使用

依存関係からインストール
された Node.js パッケージ

ルーティング実装一式

サーバー実装一式

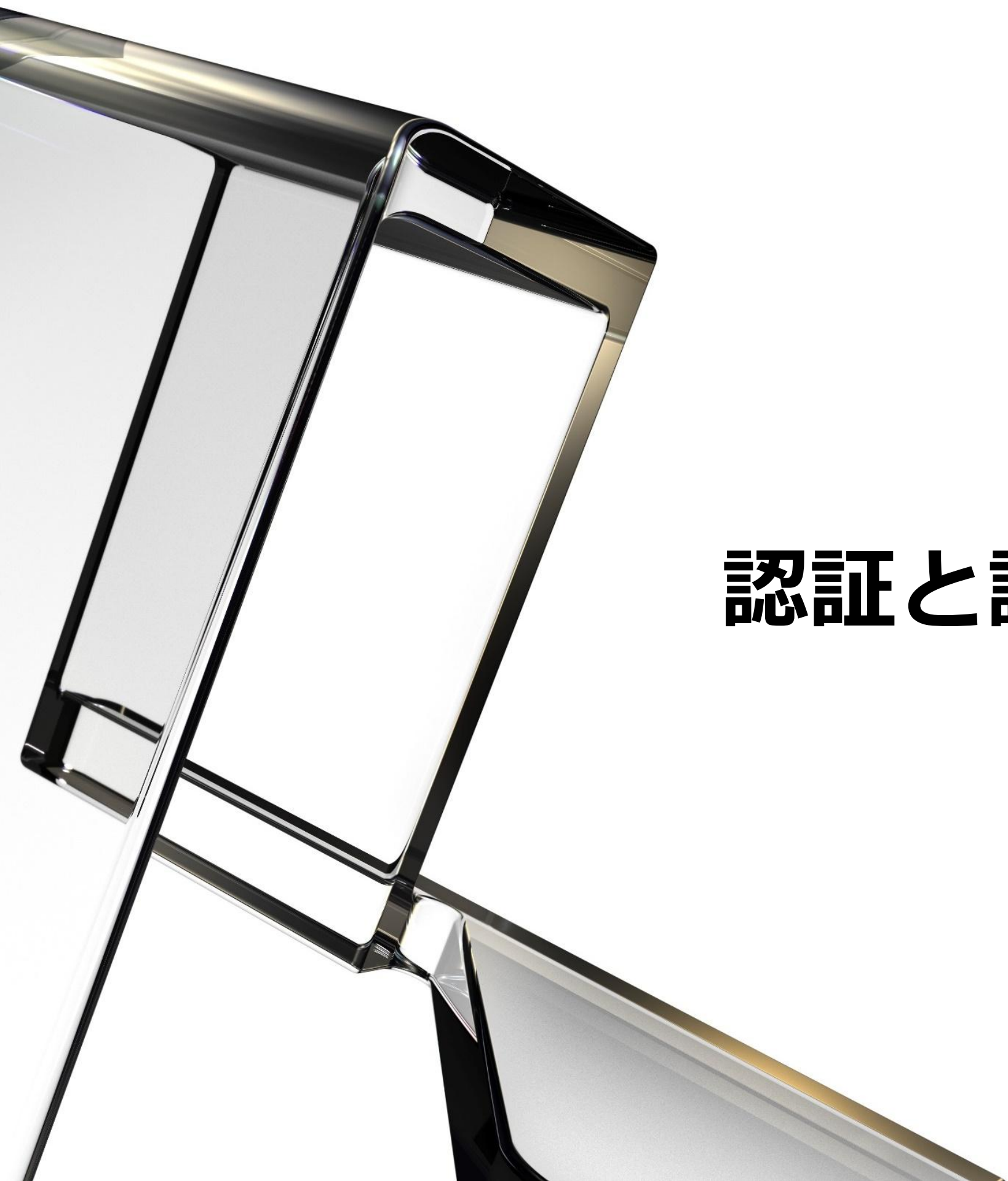
クライアント構成一式

Client ID、Secret を記述

Node.js の起動時情報指定

Node.js パッケージ依存関係

Node.js サーバーの起動構成

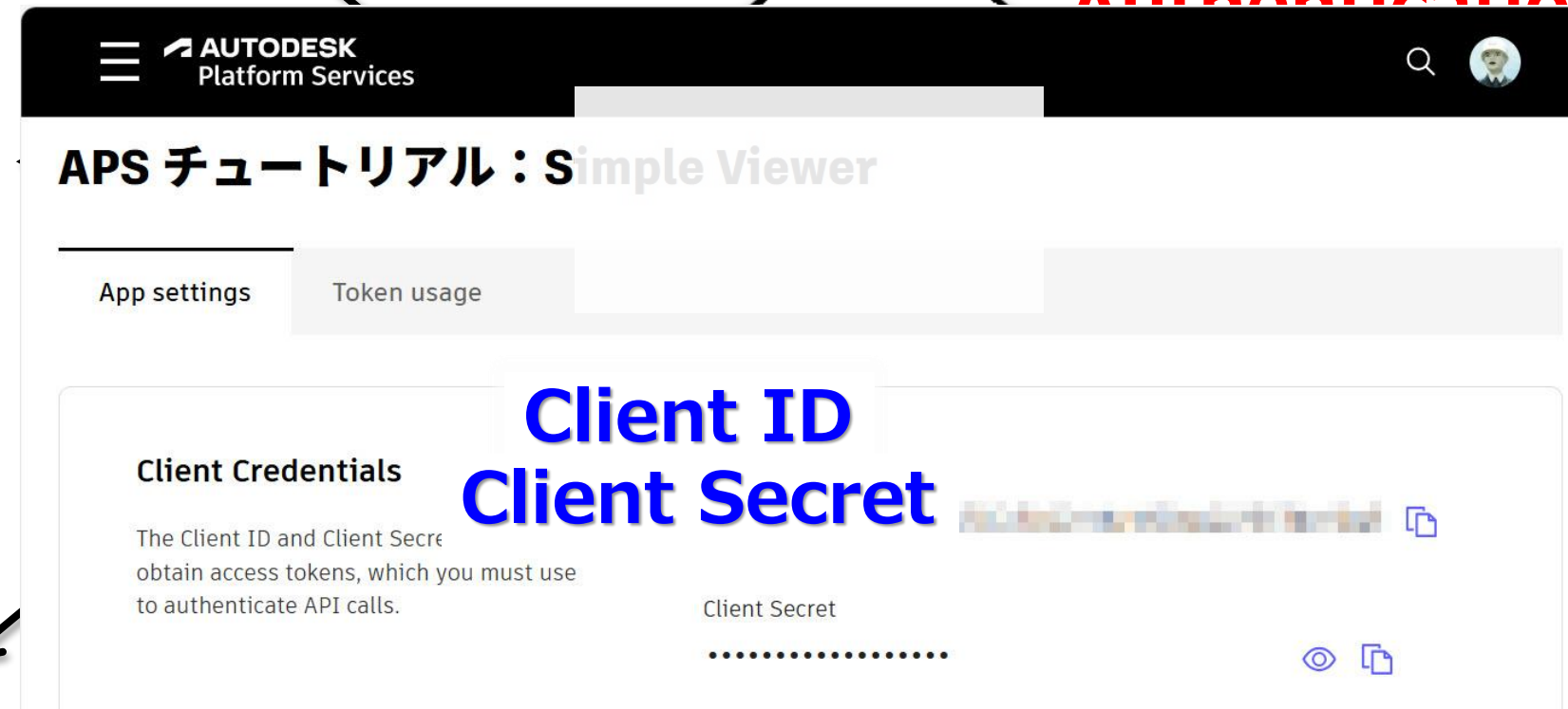
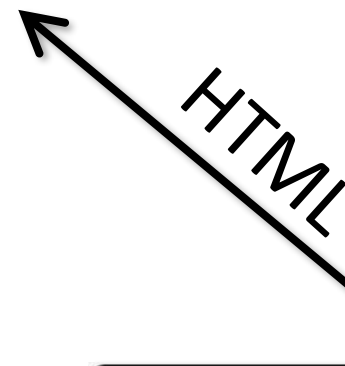


認証と認可

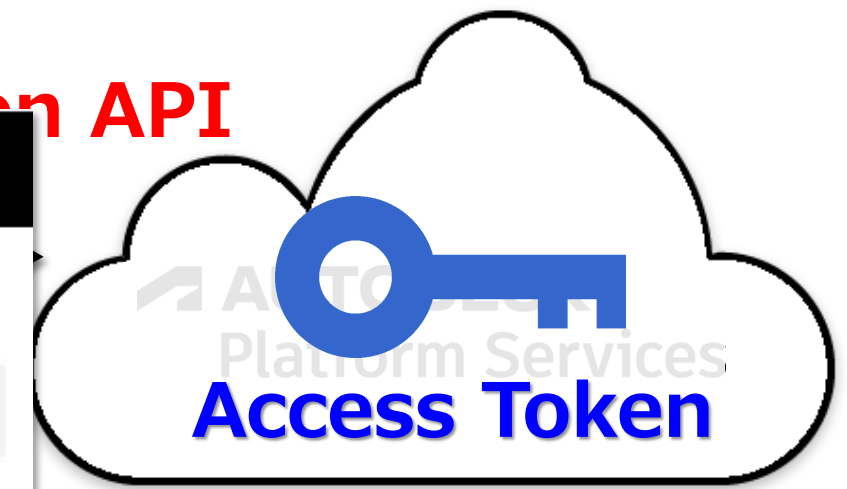
デベロッパ キーとアクセス トークン



APS ポータル (aps.autodesk.com)



Authentication API

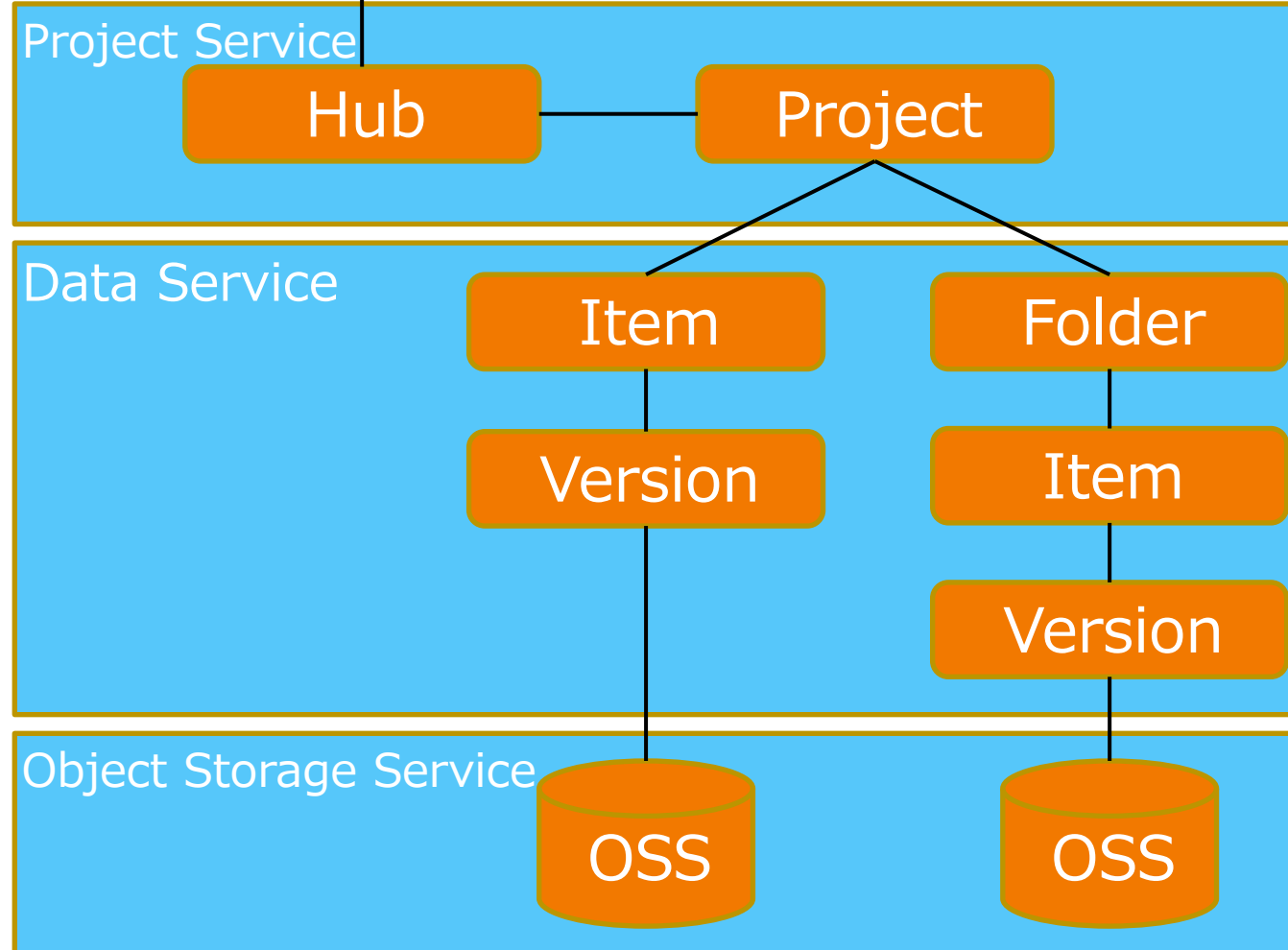


シナリオ/目的に応じた認証フロー

SDK/
Data Management

APS アプリ

3-legged / (2-legged) OAuth



APS アプリが Fusion Team、ACC のユーザーデータ領域に直接アクセスして運用
- 3 - Legged OAuth -

APS アプリ

SDK/OSS

2-legged OAuth

APS アプリが Bucket でデータを直接管理・運用
- 2 - Legged OAuth -



Bucket

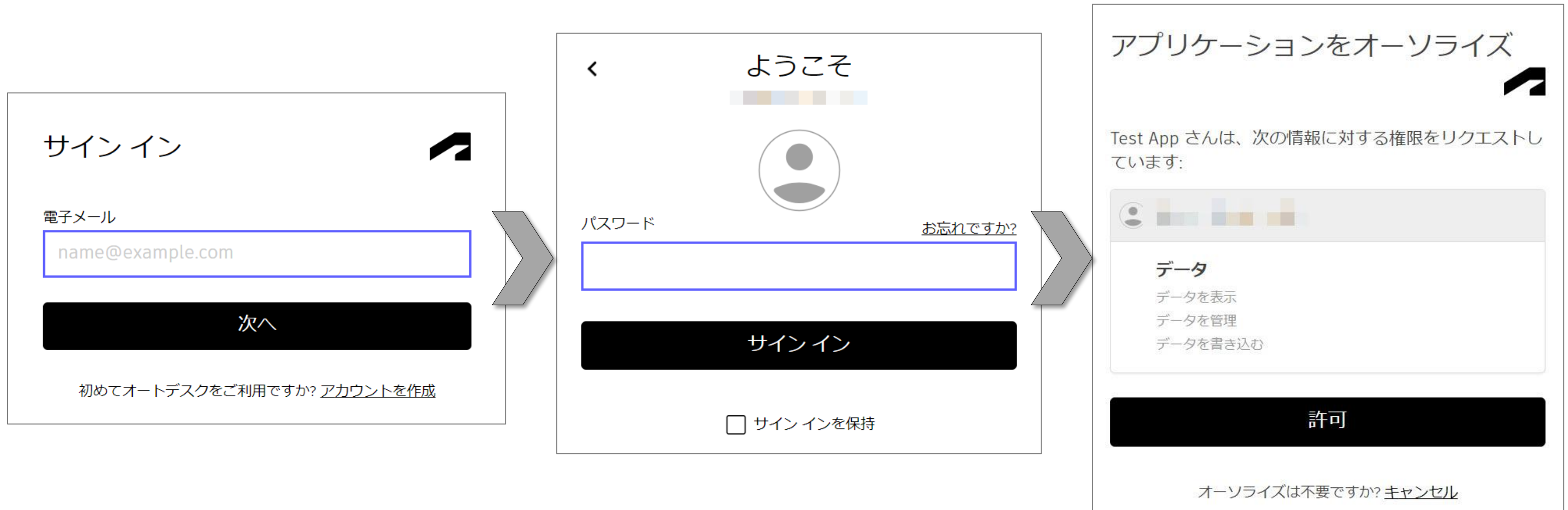
アクセス トークン (Access Token) とは

- クラウド リソース へのアクセス権限をチェックする仕組み
 - 有効期限が設定される
- 生成に必要なもの
 - デベロッパキー
 - **Client ID** と **Client Secret** のペア
 - APS ポータルでアプリ (App) 作成時に取得可能
 - スコープ (Scope)
 - リソースへのアクセス権限を指定
 - 使用するリソースによって適切に使い分けが必要
 - 複数の Scope 文字列を半角スペースで結合して指定

クラウドサービスに応じた使い分け

- Fusion Team
 - エンドユーザーのデータにアクセスするには、APS アプリに **3-legged 認証フローで得た** アクセス トークンをサーバーに提供する必要があります。
- Autodesk Docs (ACC)
 - 事前に Account Admin が Autodesk Construction Cloud (ACC) で「**カスタム統合**」を使って APS アプリを登録する必要があります。
 - APS アプリは、**3-legged 認証フローまたは 2-legged 認証フローのいずれかで得た** アクセス トークンを使用してデータにアクセスできます。

3-legged 認証フロー：エンドユーザーによる認可



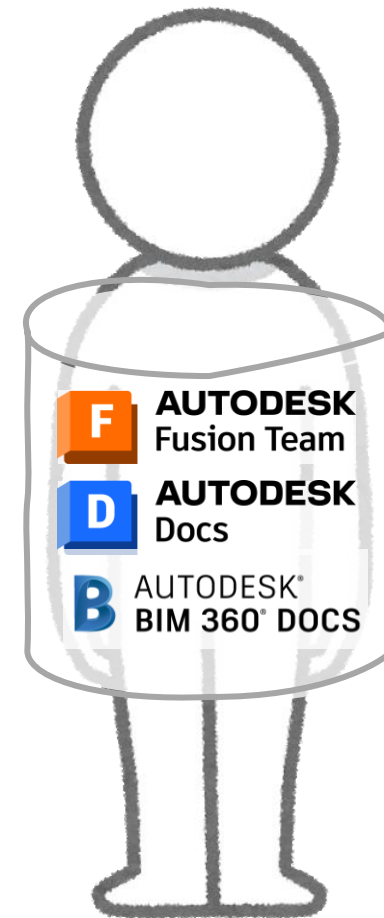
① APS アプリにアクセスすると
デザインデータの所有者である
エンドユーザーにサインインを促す
ユーザ名(Autodesk ID)画面に遷移

② 同じくパスワード入力画面に遷移

③ サインインが完了した時点で
APS アプリが指定した Scope に
沿って APS アプリからのアクセス
を認可するか確認する画面に遷移

3-legged 認証フローのイメージ

データ リソースのオーナー
(SaaS アプリのユーザー)



APS を利用する Web アプリ



APS サーバー (リソース)

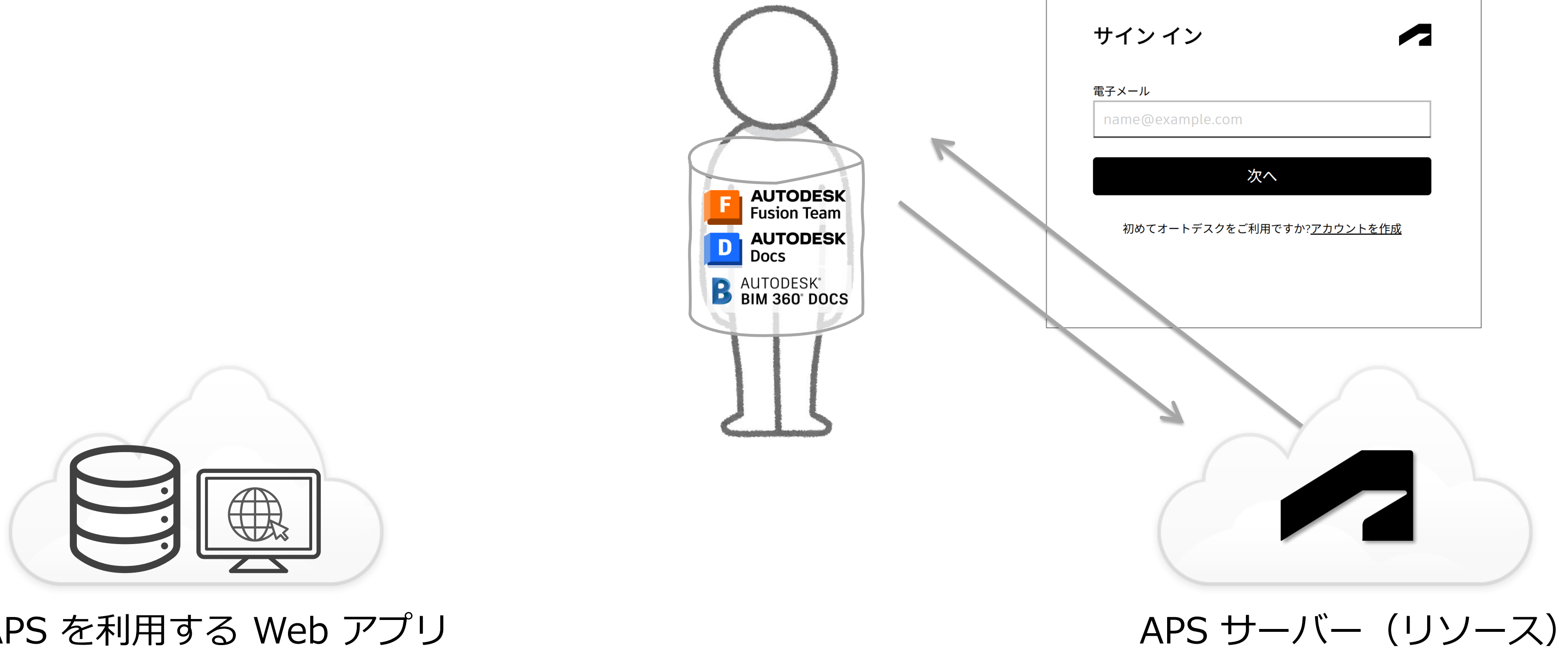
3-legged 認証フローのイメージ

- APS とユーザ データ リソースアクセスをリクエスト



3-legged 認証フローのイメージ

- データ リソース オーナーへのログインを求める



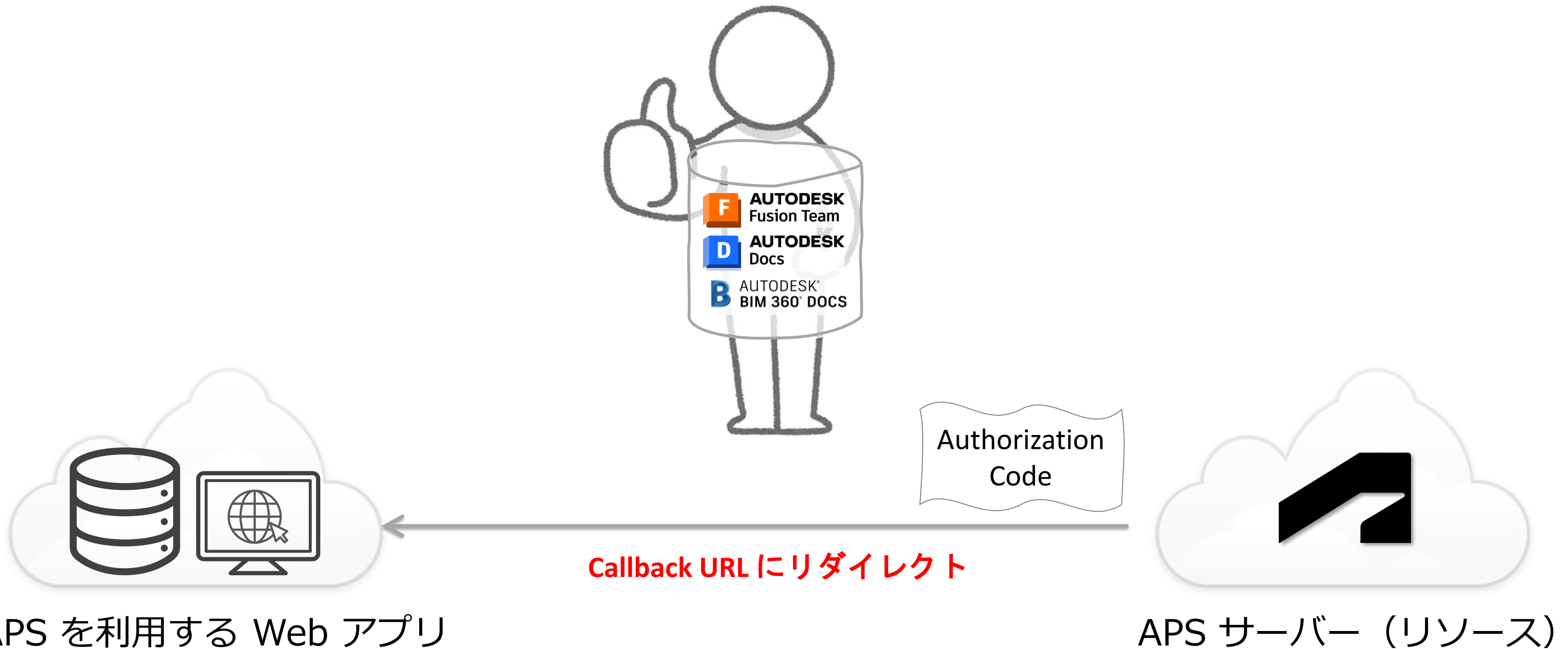
3-legged 認証フローのイメージ

- データ オーナーがスコープで指定された範囲でのデータアクセスを認可



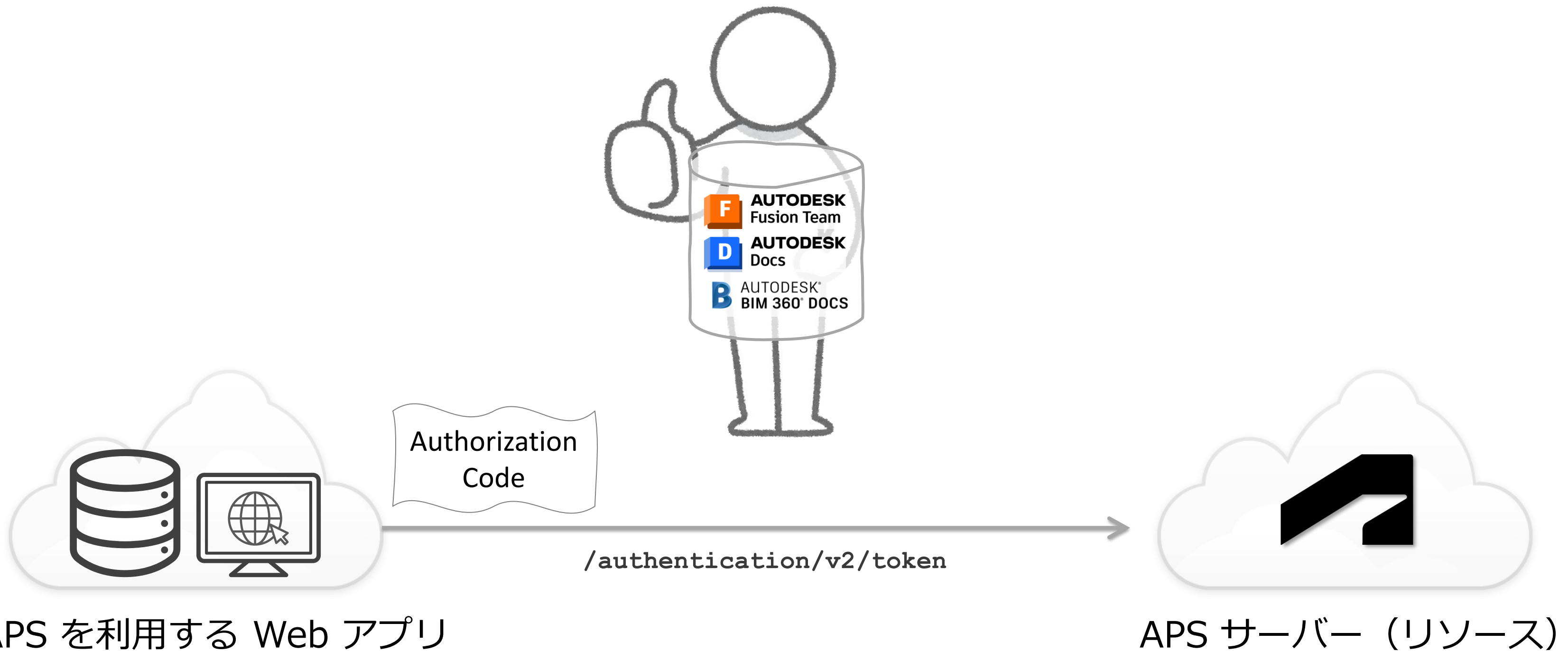
3-legged 認証フローのイメージ

- データ オーナーのデータアクセス許可を明示



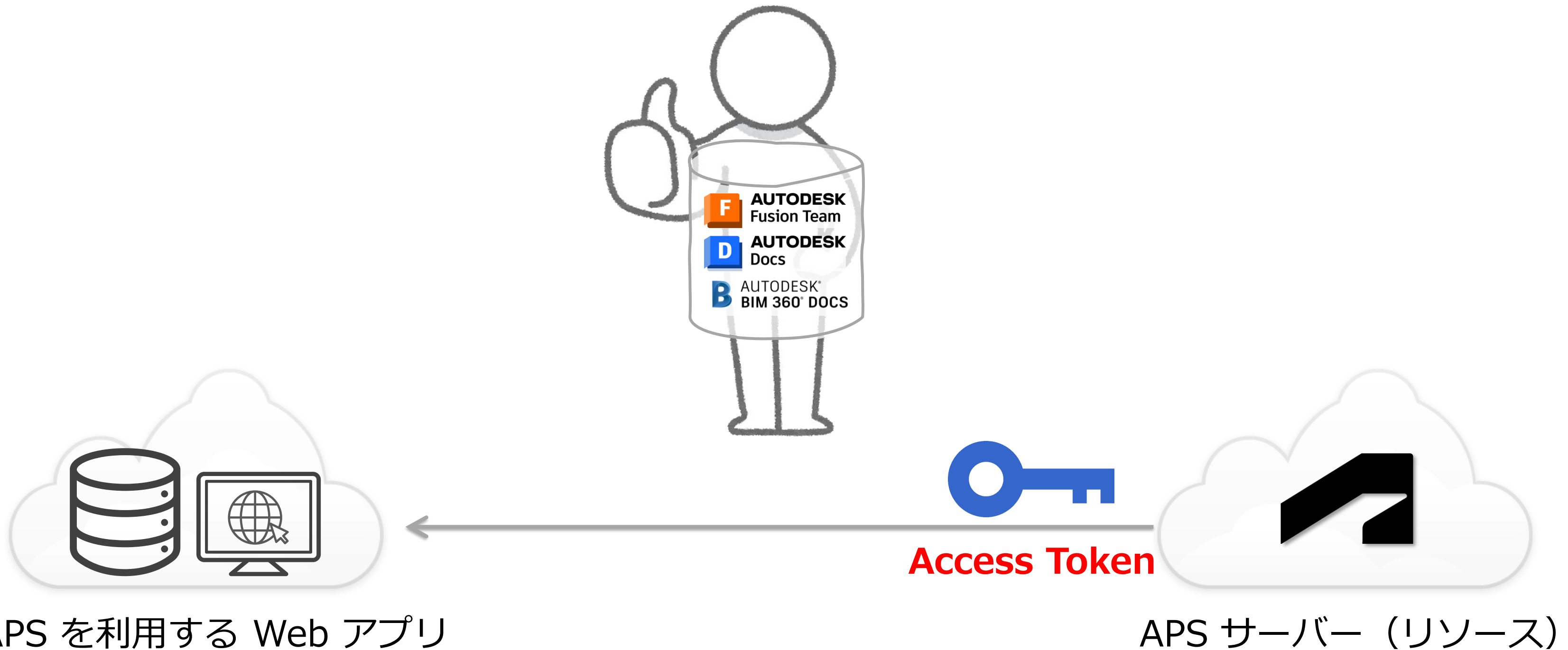
3-legged 認証フローのイメージ

- データ オーナーのデータアクセス許可を明示



3-legged 認証フローのイメージ

- APS アプリを**認証**してアクセストークンを発行



Scope（スコープ）とは

Scope 文字列	意味
user-profile:read	プロフィール（Autodesk ID）の表示
user:read	プロフィール（Autodesk ID）の読み取り
user:write	プロフィール（Autodesk ID）の書き込み
viewables:read	変換後のデザインデータ（SVF）の読み取り（表示）
data:read	ストレージデータの読み取り
data:write	ストレージデータの書き込み（編集）
data:create	ストレージデータの作成
data:search	ストレージデータの検索
bucket:create	新しい Bucket の作成
bucket:read	Bucket の読み取り
bucket:update	Bucket の更新
bucket:delete	Bucket の削除
code:all	コードの生成または実行（Design Automation API）
account:read	アプリやサービス アカウントの読み取り
account:write	アプリやサービス アカウントの書き込み

適切な OAuth スコープ

- リファレンスで エンドポイント 毎に必要なスコープをチェック！

GET buckets

This endpoint will return the buckets owned by the application. This endpoint supports pagination.

Resource Information

Method and URI	GET https://developer.api.autodesk.com/oss/v2/buckets
Authentication Context	app-only
Required OAuth Scopes	bucket:read
Data Format	JSON

bucket:read

POST buckets

Creates a bucket. Buckets are arbitrary spaces that are created by applications and are used to store objects for later retrieval. A bucket is owned by the application that creates it.

Resource Information

Method and URI	POST https://developer.api.autodesk.com/oss/v2/buckets
Authentication Context	app-only
Required OAuth Scopes	bucket:create
Data Format	JSON

bucket:create

2-legged or 3-legged

- エンドポイント毎に必要とされるコンテキストは異なります。

GET

projects/:project_id/items/:item_id

Retrieves metadata for a specified item. Items represent word documents.

Notes:

- The tip version for the item is included in the `included` array.
- To retrieve metadata for multiple items, see the [ListItems](#) command.

New! Autodesk Construction Cloud platform (ACC). Note that this endpoint is only available to the bucket creator. For more information about the new Construction Cloud APIs, see the [Autodesk Construction Cloud documentation](#).

Resource Information

Method and URI	GET https://developer.api.autodesk.com/oss/v2/projects/:project_id/items/:item_id
Authentication Context	user context optional
Required OAuth Scopes	data:read

GET

users/me

Retrieves the profile information about an end user, including the user's name and email address. However it does not currently provide full information about a user's permissions, such as *create*, or *full control* permissions.

New! Autodesk Construction Cloud platform (ACC). Note that this endpoint is only available to the bucket creator. For more information about the new Construction Cloud APIs, see the [Autodesk Construction Cloud documentation](#).

Resource Information

Method and URI	GET https://developer.api.autodesk.com/oss/v2/users/me
Authentication Context	user context required
Required OAuth Scopes	account:read
Data Format	JSON

GET

buckets/:bucketKey/objects

List objects in a bucket. It is only available to the bucket creator.

New! Autodesk Construction Cloud platform (ACC). Note that this endpoint is only available to the bucket creator. For more information about the new Construction Cloud APIs, see the [Autodesk Construction Cloud documentation](#).

Resource Information

Method and URI	GET https://developer.api.autodesk.com/oss/v2/buckets/:bucketKey/objects
Authentication Context	app-only
Required OAuth Scopes	data:read
Data Format	JSON

< JUMP TO

2-legged 認証とユーザー情報の組み合わせ

POST `projects/:project_id/items`

Headers

Authorization * Must be `Bearer <token>` , where `<token>` is obtained via either a two-legged or three-legged OAuth flow.
string

Content-Type * Must be `application/vnd.api+json`
string

x-user-id
string

In a two-legged authentication context, the app has access to all users specified by the administrator in the SaaS integrations UI. By providing this header, the API call will be limited to act on behalf of only the user specified.

Note that for a three-legged OAuth flow or for a two-legged OAuth flow with user impersonation (`x-user-id`), the user must have permission to upload files to the specified parent folder (`data.attributes.relationships.parent.data.id`).

例) アプリが 2-legged 認証フローでアップロード

- アップロードしたユーザーは誰？

**2-legged Token 使用時に
x-user-id リクエストヘッダー未指定時**

フォルダ パッケージ

✓ フォルダ プロジェクト ファイル...

フォルダ A Project

☐ 名前 ↑	最終更新日	更新者	バージョンの追加者	レビ
☐ result.pdf	2024年11月18日 17:31	ACC システム	ACC システム	
☐ result2.pdf	2024年11月18日 18:00	Toshiaki Isezaki Developer Advocacy ...	Toshiaki Isezaki Developer Advoca...	

**3-legged Token 使用時、または
2-legged Token 使用時に
x-user-id リクエストヘッダー指定時**

ACC 統合アプリの運用のまとめ

- プロジェクト データのオーナーは「ユーザー」
- 原則、3-legged 認証フローの利用を推奨

認証フロー	ACC	Fusion Team
2-legged 認証	○	×
2-legged 認証 x-user-id	○	×
3-legged 認証	○	○

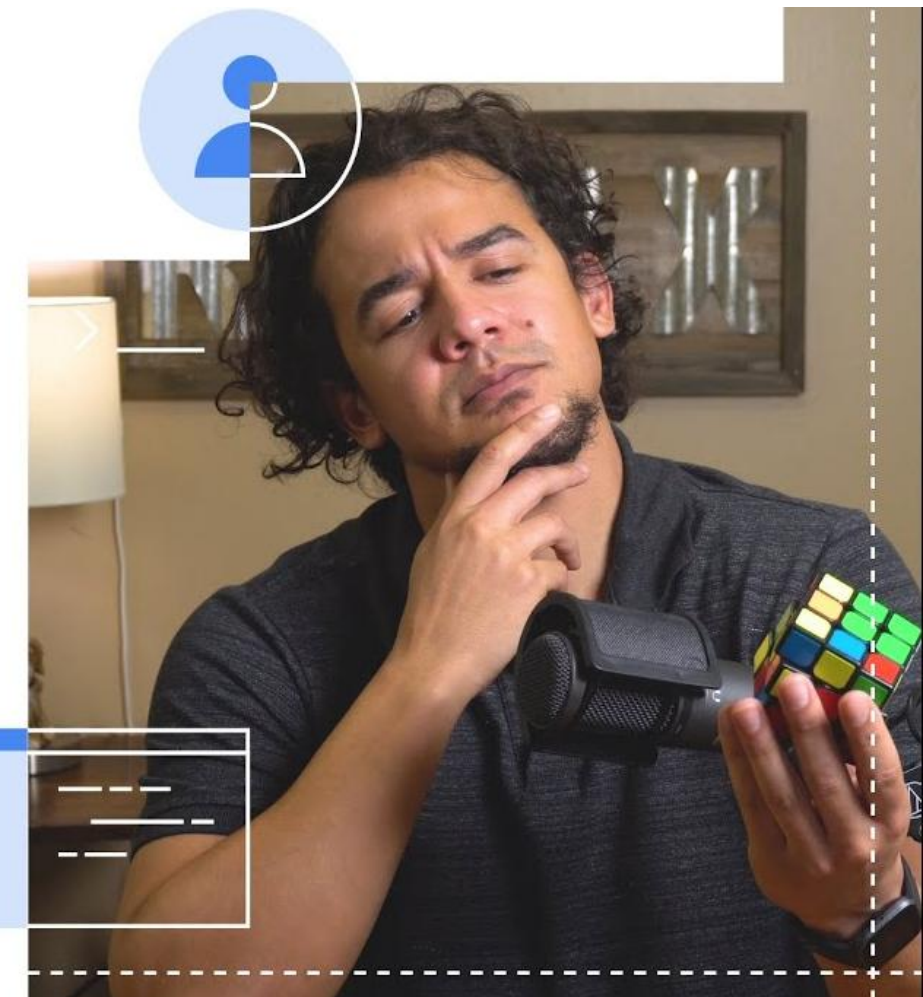
- x-user-id は [GET OIDC User Info](#) sub フィールド値
- 2-legged 認証フローはアプリケーション コンテキスト時のみ

Secure Services Accounts (セキュア サービス アカウント)

- 「スコープ付き」サーバー間統合の重要性を理解
- 「時々強力すぎる」 2-Legged の必要性を低減/排除を目的
- 「3-legged 認証」の必要な API でもセキュアに統合可能に

Service Accounts
Basics

https://adndevblog.typepad.com/technology_perspective/2025/04/introducing-secure-service-accounts-ssa-now-in-public-beta.html



チュートリアル：認証

70

 **AUTODESK**
Platform Services

 Sign in

Solutions ▾Getting StartedDocumentation ▾Success StoriesCommunity ▾Support ▾PricingApp Store ▾

Getting Started
Environment Setup
Tutorials
Simple Viewer
Hubs Browser
Application Setup
Authentication
Data Browsing
Viewer & UI
Dashboard
Design Automation
ACC Administrator
ACC Issues (beta)
Learn More

 > Tutorials > Hubs Browser > Authentication

Authentication

In this step we're going to extend the server implementation so that it can authenticate itself to the APS platform, guide the user through a [3-legged OAuth workflow](#), and generate access tokens for various needs.

 **TIP**
It is a good practice to generate an "internal" token with more capabilities (for example, allowing the owner to create or delete files in the Data Management service) that will only be used by the server, and a "public" token with fewer capabilities that can be safely shared with the client-side logic.

Access tokens

Node.js & VSCode.NET & VSCode.NET & VS2022

Create an `aps.js` file under the `services` folder. This is where we will be implementing all the APS logic that will be used in different areas of our server application. Let's start by adding the following code to the file:

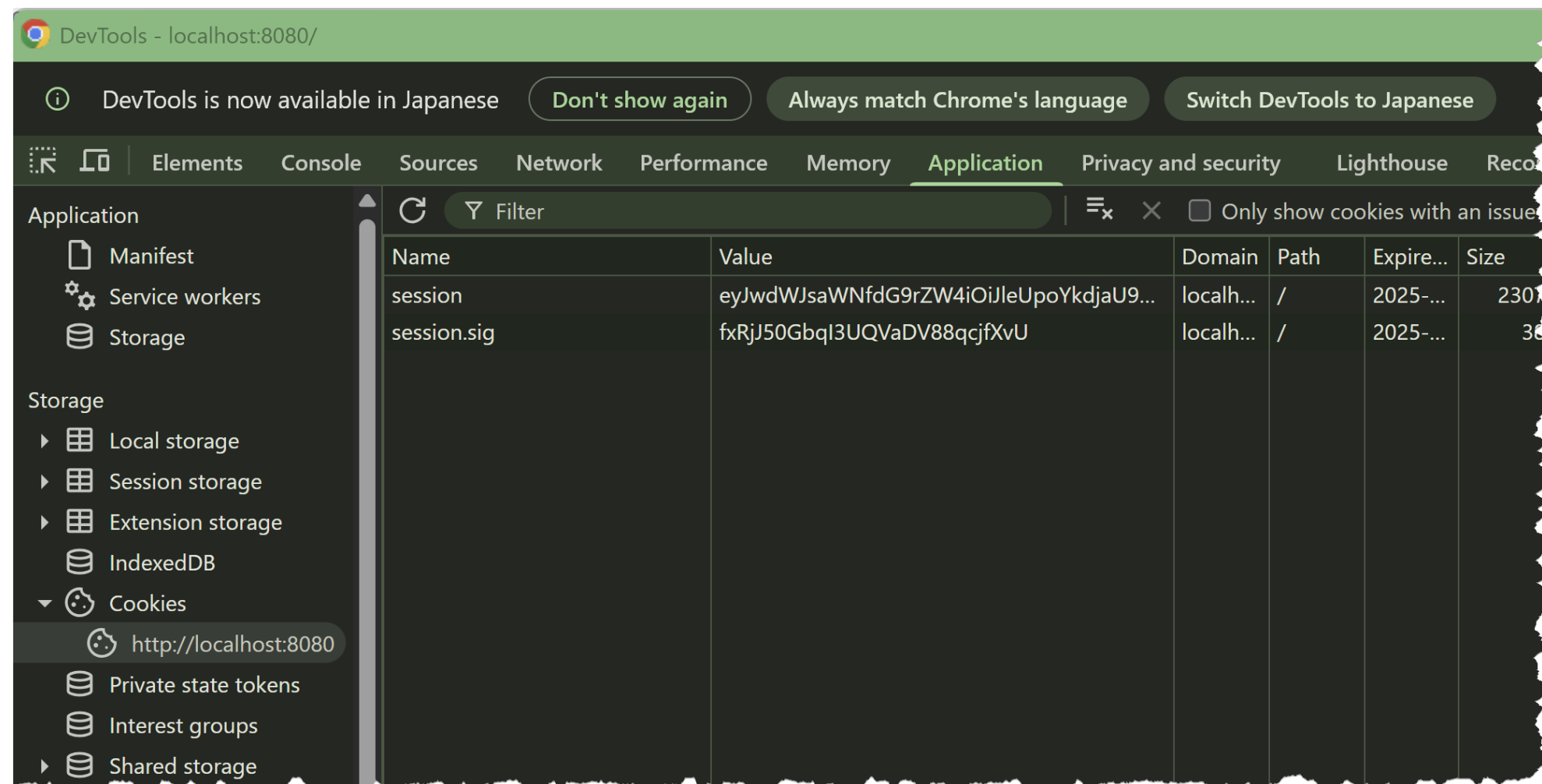
services/aps.js

```
const { AuthenticationClient, ResponseType } = require('@aps_sdk/authentication');
```

Access tokens
Server endpoints
Try it out

チュートリアル： VS Code 上の操作手順

1. services フォルダ下に aps.js ファイルを作成して実装
2. routes フォルダ下に auth.js ファイルを作成して実装
3. server.js ファイルに auth.js をマウント
4. デバッグ実行 >> <http://localhost:8080/api/auth/login>



FileEditSelectionViewGoRunTerminalHelp

Hub Browser

72

EXPLORERHUB BROWSER

.vscode

launch.json

node_modules

routes

services

wwwroot

.env

config.js

package-lock.json

package.json

server.js

15 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities

PS C:\Users\ogasawr\Hub Browser> npm install --save @aps_sdk/authentication @aps_sdk/data-management

added 44 packages, and audited 122 packages in 10s

18 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities

PS C:\Users\ogasawr\Hub Browser>

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS

powershell

Show All CommandsCtrl + Shift + P

Go to FileCtrl + P

Open ChatCtrl + Alt + I

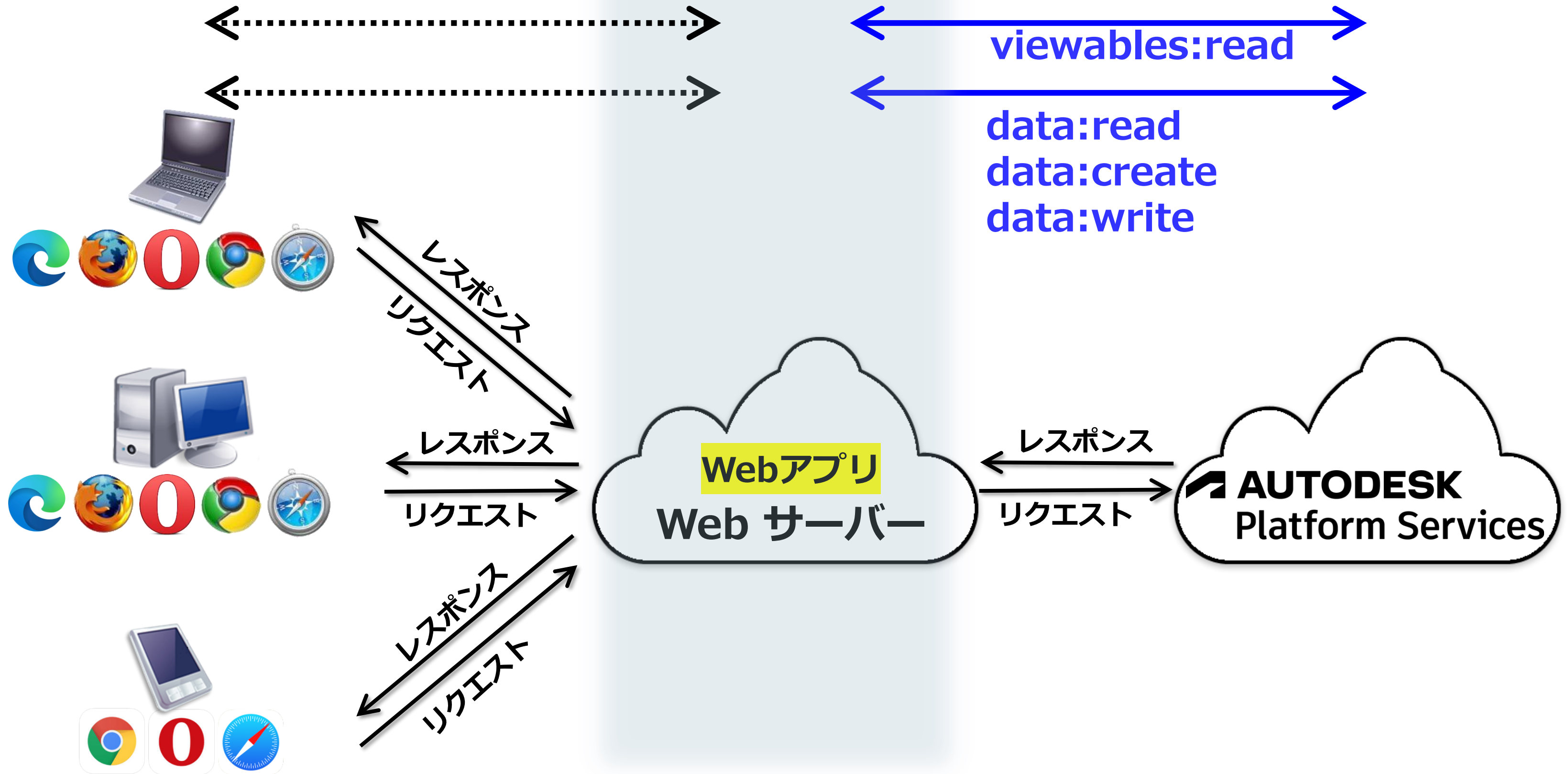
Find in FilesCtrl + Shift + F

Toggle TerminalCtrl + @

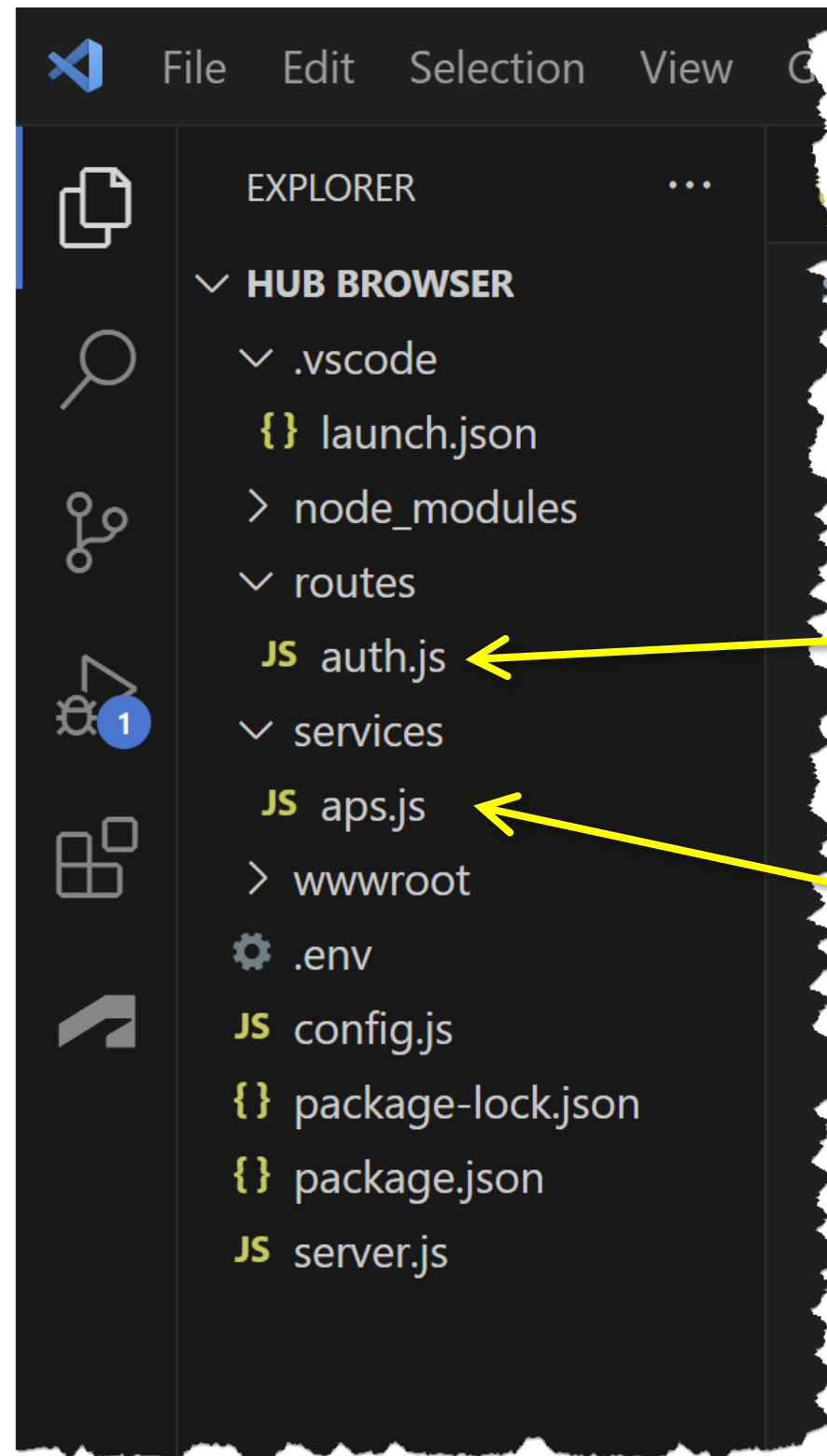
<00Launch Server (Hub Browser)

00

スコープによるトークンを使い分け



ここまでのプロジェクト構成

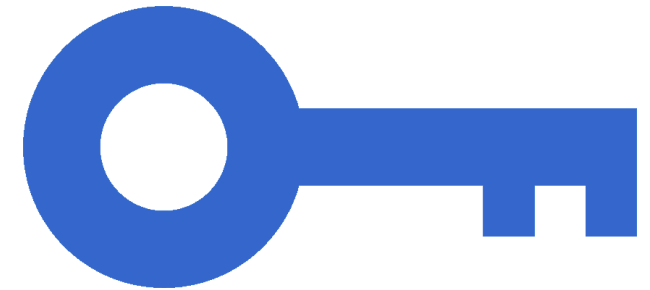


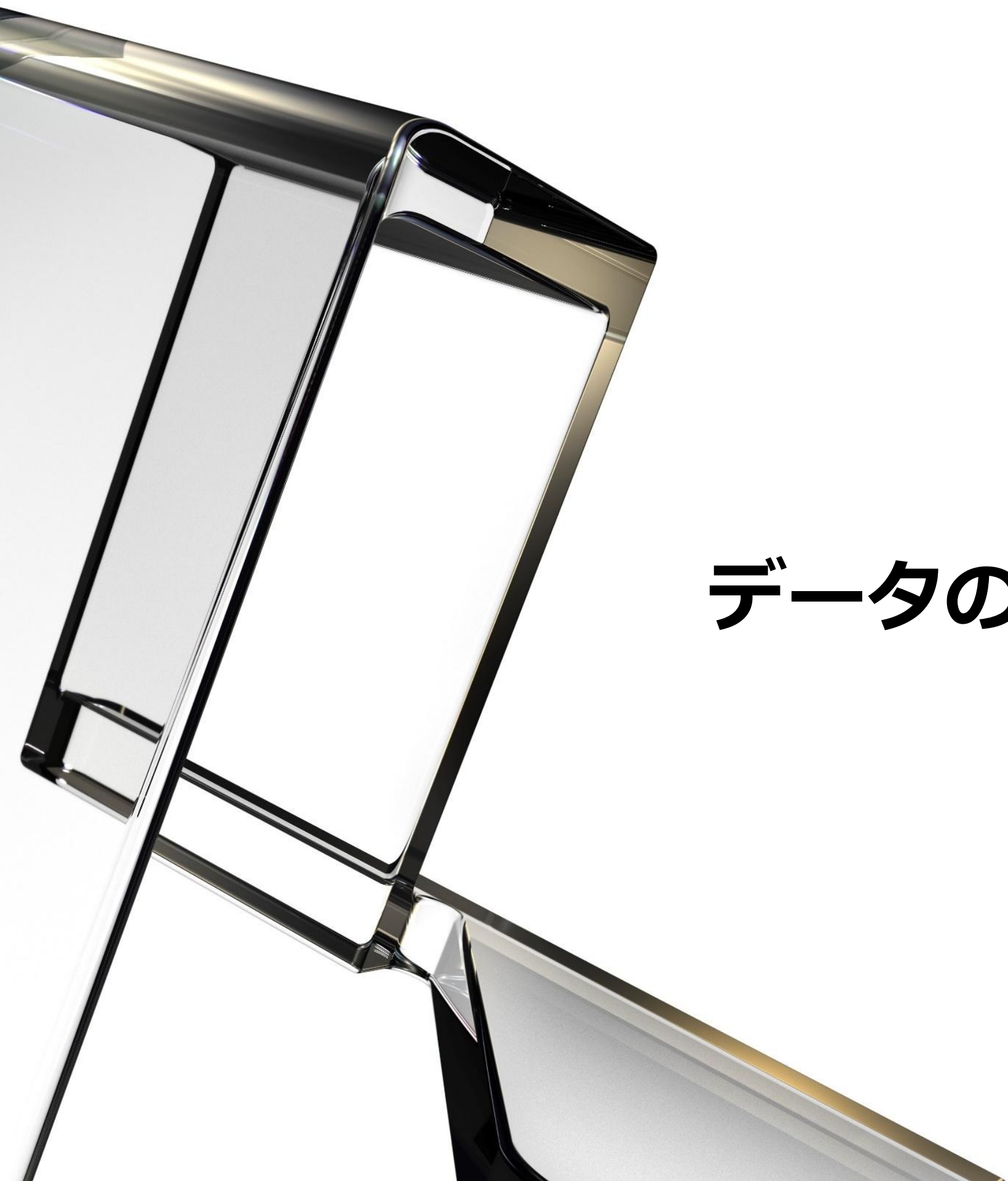
クライアントが呼ぶ出す
エンドポイントを実装
(Client ID/Secret を隠蔽し
てアクセストークンを取得)

サーバー実装からアクセス
トークンを取得する関数定義

その他、3-legged 認証単体の評価

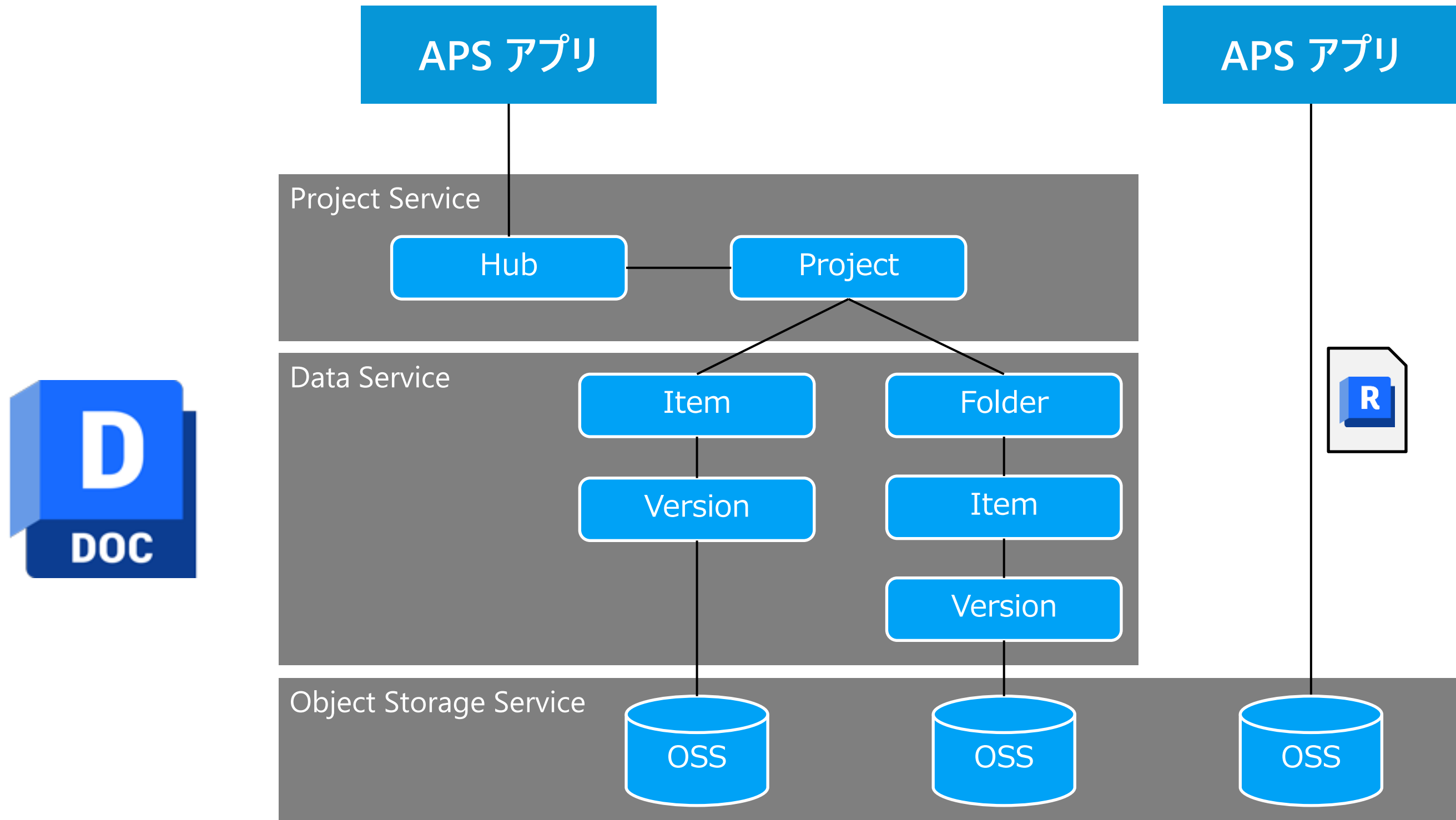
- アクセストークンの取得
- Postman
 - [Postman を使った 3-legged 認証・認可\(OAuth v2\)](#)
- VS Code
 - [新 Node.js SDK と VS Code で 3-legged アクセストークンを取得](#)



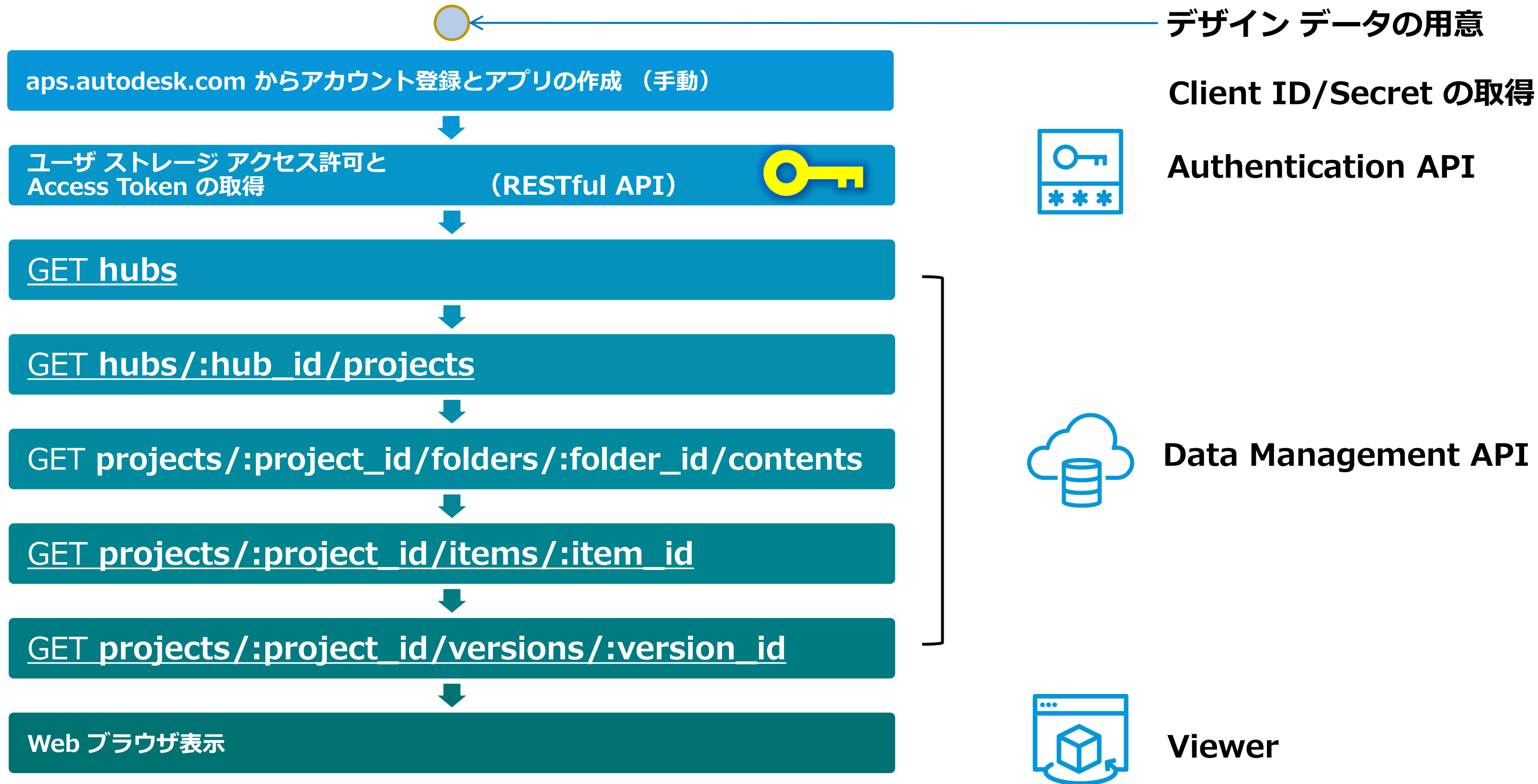


データのブラウジング

シナリオ/目的に応じた使い分けが必要



3-legged OAuthでの Viewer 利用手順の理解



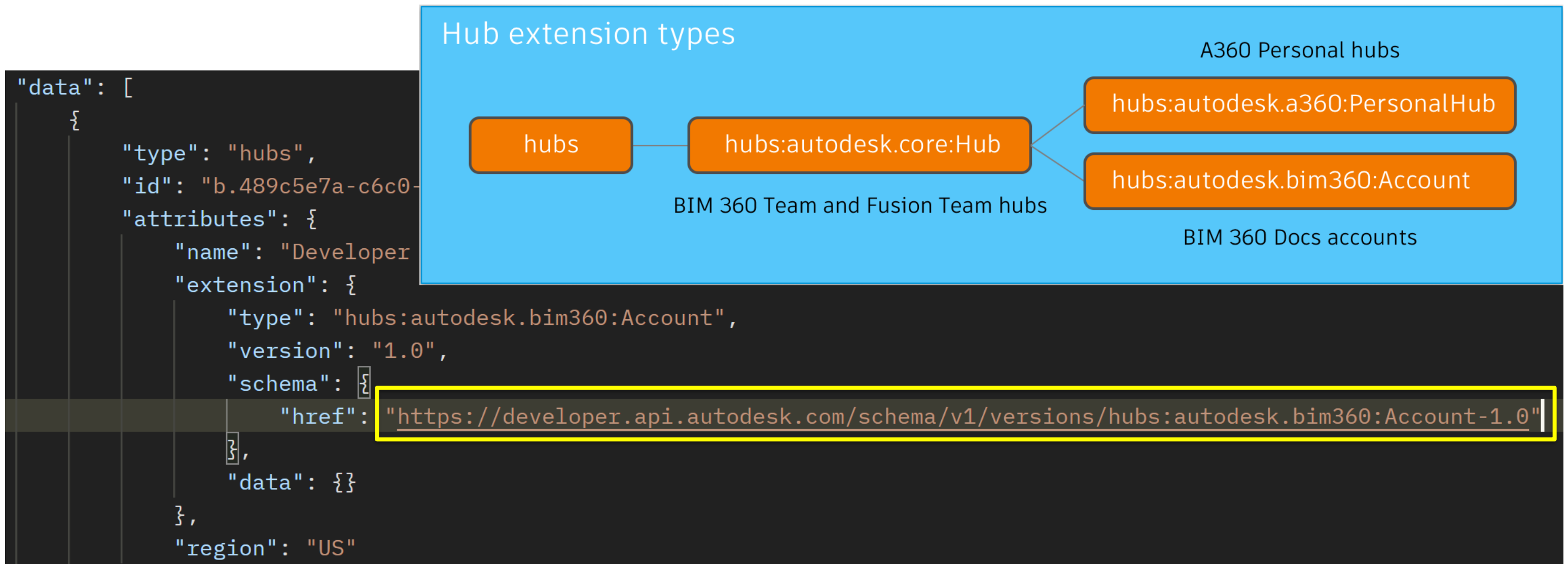
Hubs – ACC/BIM 360 とそれ以外の Hub の判別

- “Hubs” は、Autodesk Docs/BIM 360 Docs の “Account” に相当。
- Autodesk Docs/BIM 360 Docs の Hub と Project の Id は、b.xxx で始まり、それ以外のプロダクトは a.xxx で始まります。

```
"data": [  
  {  
    "type": "hubs",  
    "id": "b.489c5e7a-  
    "attributes": {  
      "name": "Developer Advocacy Support",  
      "extension": {  
        "type": "hubs:autodesk.bim360:Account",  
        "version": "1.0",  
        "schema": {  
          "href": "https://developer.api.autodesk.com/schema/v1/versions/hubs:autodesk.bim360:Account-1.0"  
        },  
        "data": {}  
      },  
      "region": "US"  
    },  
  ],  
]
```

Hubs – タイプの違い

- *:autodesk.core:*ではなく、*:autodesk.bim360:* の拡張タイプを使用
詳細: https://aps.autodesk.com/en/docs/data/v2/developers_guide/basics/



Projects – ACC/BIM 360 の判別

- プロジェクトが BIM 360 か ACC か判別するには、“projectType” 属性を参照します。

GET hubs/:hub_id/projects

```
"type": "projects",
"id": "b.4fae[REDACTED]",
"attributes": {
  "name": "Ryuji ACC Test Project ",
  "scopes": [
    "b360project.4fae[REDACTED]",
    "02tenant.[REDACTED]"
  ],
  "extension": {
    "type": "projects:autodesk.bim360:Project",
    "version": "1.0",
    "schema": {
      "href": "https://developer.api.autodesk.com/schema/v1/versions/projects:autodesk.bim360:Project-1.0"
    },
    "data": {
      "projectType": "ACC"
    }
  }
},
```

data.attributes.extension.data.projectType
ACC | BIM360

Folders

- [プロジェクト ファイル]フォルダが起点

The screenshot displays the Autodesk Construction Cloud interface. The top navigation bar includes the 'Docs' dropdown and the project name 'Ryuji ACC 2023 Test Project'. The left sidebar contains a list of navigation items: 'ファイル' (Files), 'レビュー' (Review), 'ファイル転送' (File Transfer), '指摘事項' (Issues), 'レポート' (Reports), 'メンバー' (Members), 'ブリッジ' (Bridge), and '設定' (Settings). The 'ファイル' section is active, showing a list of folders and files. The 'プロジェクト ファイル' (Project Files) folder is highlighted in the left sidebar. The main area shows a list of files and folders, including 'Shared', 'データ交換', '処理中', '干渉チェックテスト...', '意匠', '構造', and '設備'. The '意匠' folder is expanded, showing a list of files. The table below shows the details of the files and folders.

名前	バージョン	インジケータ	サイズ	最終更新日	更新者	マーク
名前 ↑						
Consumed	--		--	2023年10月15日 16:39	Ryuji Ogasawara	--
サンプル意匠.rvt	V8		117.4 MB	2023年10月18日 23:21	Ryuji Ogasawara	

Folders - “Project Files” Folder Id の取得

- SDK の場合、下記のメソッドで簡単に[プロジェクト ファイル]フォルダの ID を取得できます。

```
dataManagementClient.getProjectTopFolders(hubId, projectId, {accessToken});
```

- REST API の場合、

GET hubs/:hub_id/projects/:project_id/topFolders

- レスポンスデータの中から name : “Project Files” のフォルダを探す

GET projects/:project_id/folders/:folder_id/contents

- rootFolder Id を指定して、その配下のコンテンツのデータから探す

Projects – プロジェクトのルートフォルダの取得

GET hubs/:hub_id/projects

```
"relationships": {
  "hub": {
    "data": {
      "type": "hubs",
      "id": "b.489c5[REDACTED]"
    },
    "links": {
      "related": {
        "href": "https://developer.api.autodesk.com/project/v1/hubs/b.489c5[REDACTED]"
      }
    }
  }
},
"rootFolder": {
  "data": {
    "type": "folders",
    "id": "urn:adsk.wipprod:fs.folder:co.5N7Me[REDACTED]"
  },
  "meta": {
    "link": {
      "href": "https://developer.api.autodesk.com/data/v1/projects/b.4faef[REDACTED]/folders/urn:adsk.wipprod:fs.folder:co.5N7Me[REDACTED]"
    }
  }
},
}
```

レスポンスの relationships 配下のデータから、プロジェクトのルートフォルダの ID を取得できます。

Folders - “Project Files” フォルダ配下のコンテンツ

GET projects/:project_id/folders/:folder_id/contents

- 直下のみ。再帰処理が必要です。

“Project Files” FolderId

The screenshot displays the Autodesk Construction Cloud interface. At the top, the header shows 'AUTODESK Construction Cloud' and a dropdown menu for 'Ryuji ACC 2023 Test Project'. A blue callout box points to this dropdown, labeled '“Project Files” FolderId'. The left sidebar contains navigation options: 'Docs', 'ファイル' (Files), 'レビュー' (Review), 'ファイル転送' (File Transfer), '指摘事項' (Issues), 'レポート' (Reports), 'メンバー' (Members), 'ブリッジ' (Bridge), and '設定' (Settings). The main content area is titled 'ファイル' (Files) and shows a folder tree under 'プロジェクト ファイル' (Project Files). A yellow box highlights the '意匠' (Design) folder, which is expanded to show subfolders: 'Shared', 'データ交換' (Data Exchange), '処理中' (In Progress), '干渉チェックテスト...' (Interference Check Test...), '意匠' (Design), '構造' (Structure), and '設備' (Equipment). The right side of the interface features a table of files and folders. The table has columns for '名前' (Name), 'バージョン' (Version), 'インジケータ' (Indicator), 'サイズ' (Size), '最終更新日' (Last Updated), '更新者' (Updater), and 'マークアップ' (Markup). The table lists two items: 'Consumed' and 'サンプル意匠.rvt' (Sample Design.rvt). The 'サンプル意匠.rvt' file is highlighted with a blue box and has a 'V8' version indicator.

名前	バージョン	インジケータ	サイズ	最終更新日	更新者	マークアップ
Consumed	--		--	2023年10月15日 16:39	Ryuji Ogasawara	--
サンプル意匠.rvt	V8		117.4 MB	2023年10月18日 23:21	Ryuji Ogasawara	

Folder と Item の識別、Version Id の取得

GET projects/:project_id/folders/:folder_id/contents

The screenshot shows the Autodesk Construction Cloud interface for a project named 'Ryuji ACC 2023 Test Project'. The left sidebar contains navigation options: Docs, ファイル (Files), レビュー (Review), ファイル転送 (File Transfer), 指摘事項 (Issues), レポート (Reports), メンバー (Members), ブリッジ (Bridge), and 設定 (Settings). The main area displays a folder tree on the left and a list of items on the right. A blue callout box points to the '意匠' (Architecture) folder in the tree, stating 'type 属性で識別' (Identified by type attribute) and listing 'folders' and 'items'. The item list on the right has columns for '名前' (Name), 'バージョン' (Version), 'インジケータ' (Indicator), 'サイズ' (Size), '最終更新日' (Last Updated), '更新者' (Updated By), and 'マークアップ' (Markup). The 'Consumed' folder is highlighted with a yellow box, and the 'サンプル意匠.rvt' file is highlighted with a green box. The version 'V8' for the file is highlighted with a red box, with a blue callout box stating 'Items の included 配下に type : versions のデータも' (Data for type : versions is also included under Items' included). The user 'Ryuji Ogasawara' is logged in.

type 属性で識別

- folders
- items

Items の included 配下に
type : versions のデータも

Items と Versions

- 詳細な情報は、Items / Versions の各エンドポイントで取得可能。

The screenshot displays the Autodesk Construction Cloud interface for a project named 'Ryuji ACC 2023 Test Project'. The left sidebar shows a navigation menu with options like 'Docs', 'レビュー', 'ファイル転送', '指摘事項', 'レポート', 'メンバー', 'ブリッジ', and '設定'. The main area is titled 'ファイル' (Files) and shows a list of files under the 'プロジェクト ファイル' (Project Files) folder. The file list has columns for '名前' (Name), 'バージョン' (Version), 'インジケータ' (Indicator), 'サイズ' (Size), '最終更新日' (Last Updated), '更新者' (Updated By), and 'マークアップ' (Markup). A file named 'サンプル意匠.rvt' is highlighted with a green box, and its version 'V8' is highlighted with a red box.

名前	バージョン	インジケータ	サイズ	最終更新日	更新者	マークアップ
Consumed	--		--	2023年10月15日 16:39	Ryuji Ogasawara	--
サンプル意匠.rvt	V8		117.4 MB	2023年10月18日 23:21	Ryuji Ogasawara	

Learn More

[Home](#) >
 [Tutorials](#) >
 [Hubs Browser](#) >
 [Data Browsing](#)

Version

Try it out

チュートリアル: VS Code 上の操作手順

1. **services** フォルダ下の **aps.js** ファイルに実装を追加
 - Data Management API
2. **routes** フォルダ下に **hubs.js** ファイルを追加して実装
3. **server.js** ファイルに **hubs.js** をマウント
4. デバッグ実行 >> <http://localhost:8080/api/hubs>

FileEditSelectionViewGoRunTerminalHelp

Hub Browser

90

EXPLORER

HUB BROWSER

.vscode

launch.json

node_modules

routes

auth.js

services

aps.js

wwwroot

.env

config.js

package-lock.json

package.json

server.js

services > JS aps.js > ...

26 service.authRefreshMiddleware = async (req, res, next) => {
37 };
38 const publicCredentials = await authenticationClient.refreshToken(internalCredentials.refresh_token, APS_CLIENT_ID, {
39 clientSecret: APS_CLIENT_SECRET,
40 scopes: PUBLIC_TOKEN_SCOPES
41 });
42 req.session.public_token = publicCredentials.access_token;
43 req.session.internal_token = internalCredentials.access_token;
44 req.session.refresh_token = publicCredentials.refresh_token;
45 req.session.expires_at = Date.now() + internalCredentials.expires_in * 1000;
46 }
47 req.internalOAuthToken = {
48 access_token: req.session.internal_token,
49 expires_in: Math.round((req.session.expires_at - Date.now()) / 1000),
50 };
51 req.publicOAuthToken = {
52 access_token: req.session.public_token,
53 expires_in: Math.round((req.session.expires_at - Date.now()) / 1000),
54 };
55 next();
56 };
57
58 service.getUserProfile = async (accessToken) => {
59 const resp = await authenticationClient.getUserInfo(accessToken);
60 return resp;
61 };

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

powershell

15 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities
● PS C:\Users\ogasawr\Hub Browser> npm install --save @aps_sdk/authentication @aps_sdk/data-management

added 44 packages, and audited 122 packages in 10s

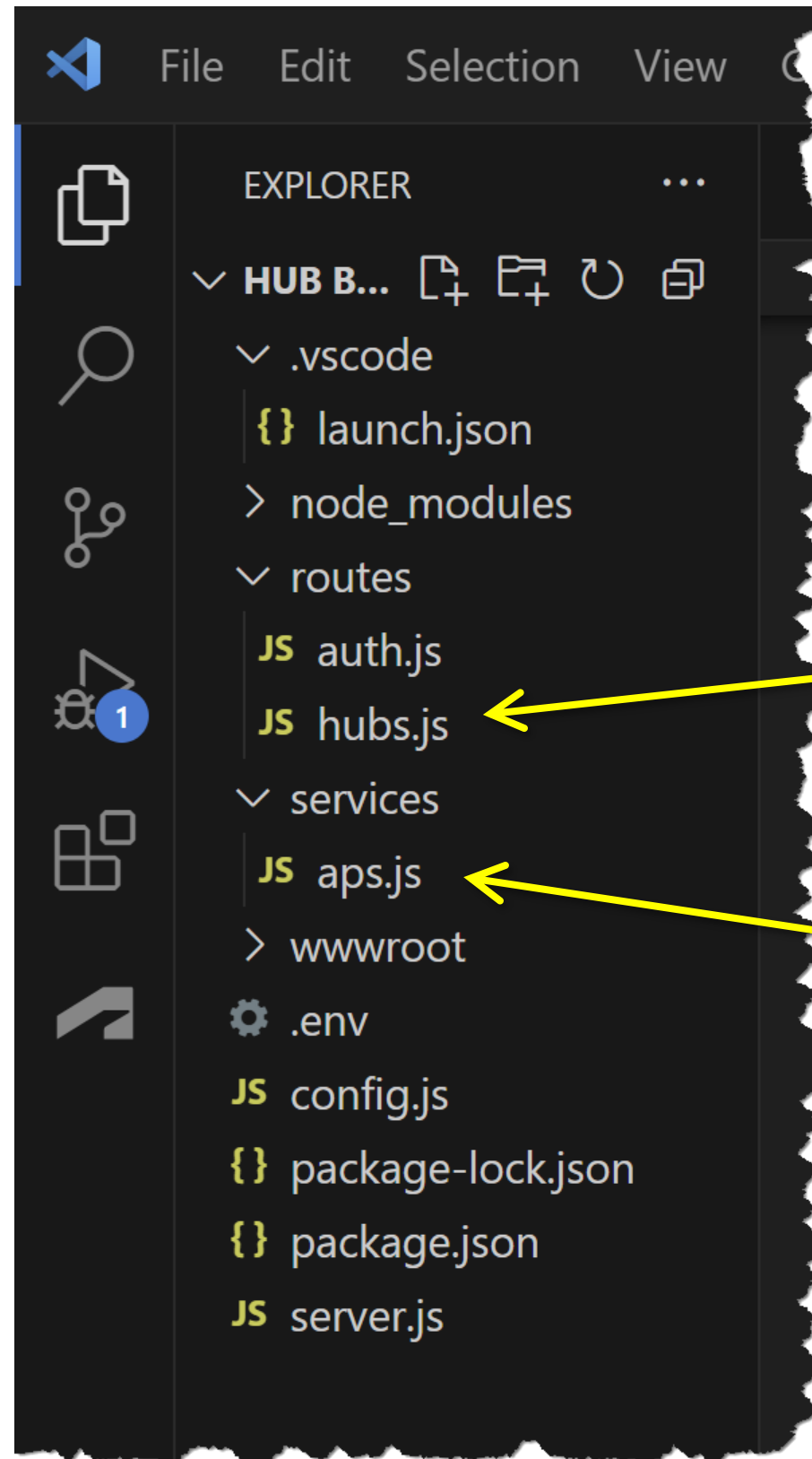
18 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities
● PS C:\Users\ogasawr\Hub Browser>

< 0 0 Launch Server (Hub Browser)

Ln 61, Col 3 Spaces: 4 UTF-8 CRLF {} JavaScript

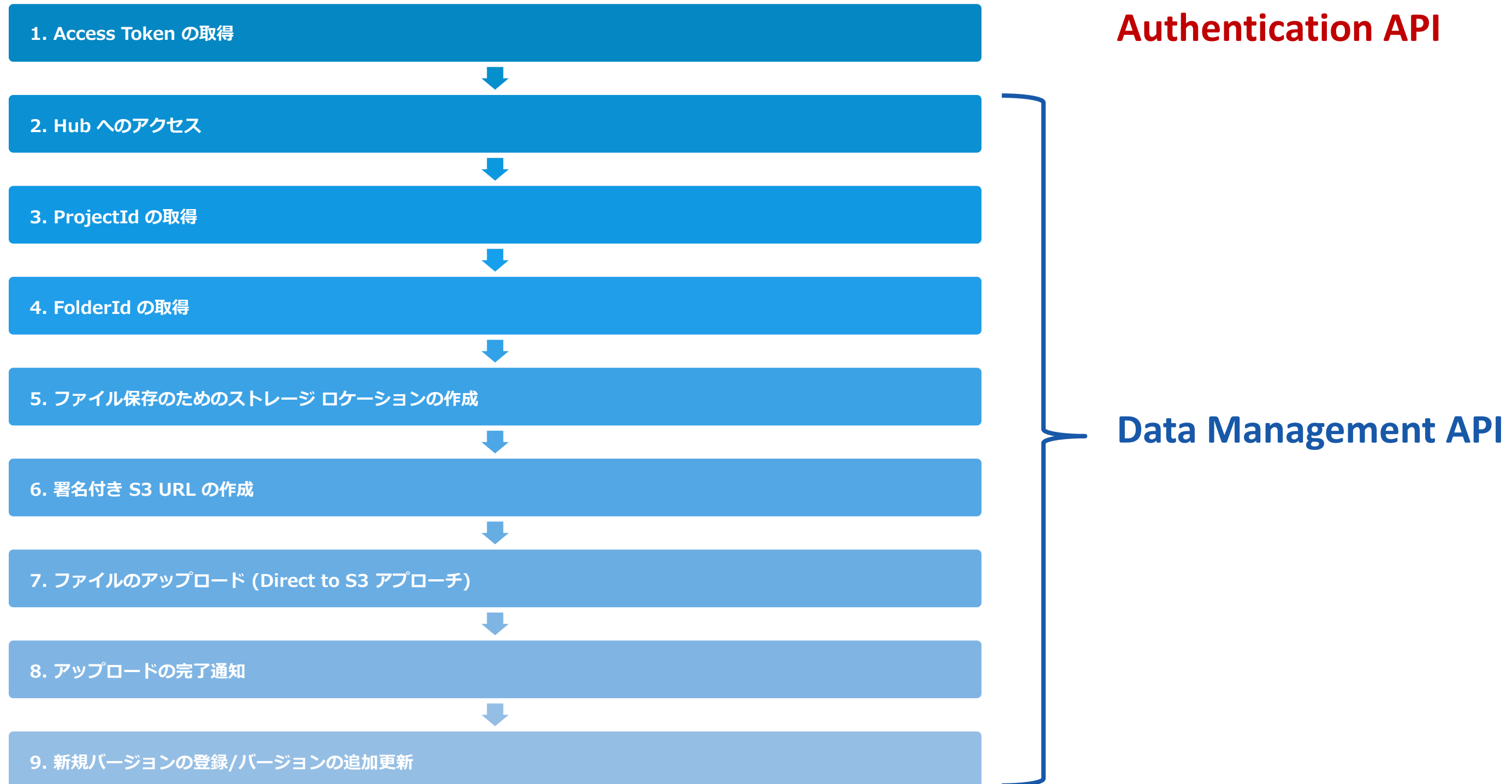
ここまでのプロジェクト構成



クライアントが呼ぶ出す
エンドポイントを実装

Autodesk Docs の情報を取得するロジックを定義

ファイルアップロードのフロー



チュートリアル: <https://aps.autodesk.com/en/docs/data/v2/tutorials/upload-file/>

SDK 利用方法: https://adndevblog.typepad.com/technology_perspective/2024/10/uploading-a-file-to-acc-using-the-net-sdk.html

ストレージ ロケーションの作成

- プロジェクトにアップロードするためのストレージロケーションを作成。

- REST API

POST projects/:project_id/storage

- SDK

createStorage()

Operation: Create a Storage Location in OSS

```
curl -v 'https://developer.api.autodesk.com/data/v1/projects/:project_id/storage' ¥  
-X 'POST' ¥  
-H 'Authorization: Bearer AuIPTf4KYLTyGVnOHQ0cuolwCW2a' ¥  
-H 'Content-Type: application/vnd.api+json' ¥  
-d '{  
  "jsonapi": {  
    "version": "1.0"  
  },  
  "data": {  
    "type": "objects",  
    "attributes": {  
      "name": "drawing.dwg"  
    },  
    "relationships": {  
      "target": {  
        "data": {  
          "type": "folders",  
          "id": "urn:adsk.wipprod:fs.folder:co.mgS-lb-BThaTdHnhiN_mbA"  
        }  
      }  
    }  
  }  
}
```

Direct-S3 アプローチによるアップロード

1. 署名付き S3 URL の作成

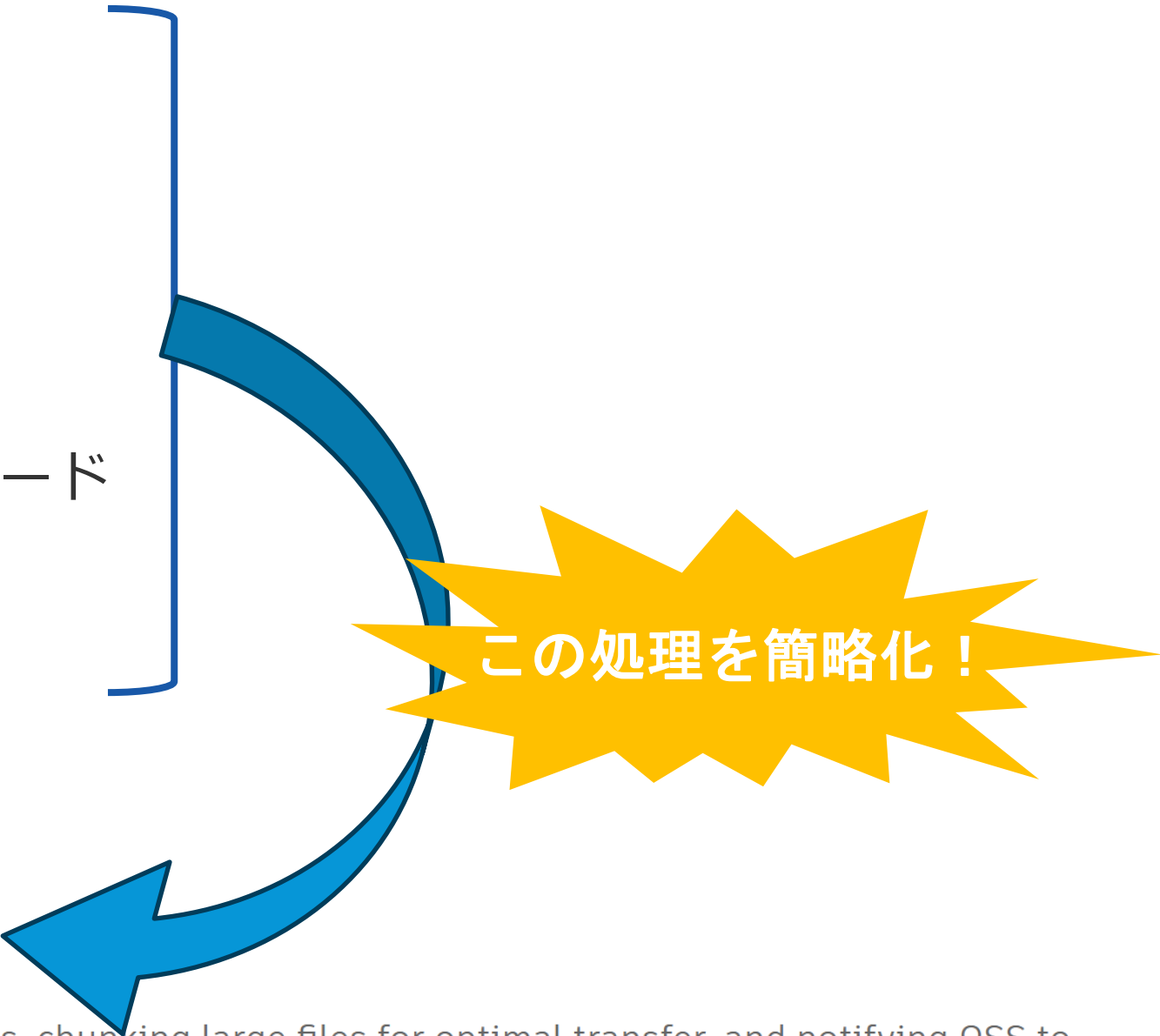
- アップロードするファイルのパート数を算出
- チャンクアップロードのための URL (最大25) を生成
 - bucketKey と objectKey、uploadKey

2. ファイルのアップロード

- パート ファイルを、対応する署名付き URL にアップロード

3. アップロードの完了

- uploadKey を使用してアップロードを確定させる



この処理を簡略化！

SDK

uploadObject()

Operation: Uploads a file by transparently handling operations like obtaining signed upload URLs, chunking large files for optimal transfer, and notifying OSS to assemble the uploaded parts.

```
uploadObject( bucketKey , objectKey , sourceToUpload , optionalArgs ?): Promise < ObjectDetails >
```

新規バージョンの登録/バージョンの追加更新

- アップロード完了しただけでは、Autodesk Docs 上に表示されません。
- 初回アップロード時は、Item の新規登録が必要です。
- 2回目以降は、バージョンを更新します。

createItem()

Operation: Create an Item

createItem(projectId , itemPayload , optionalArgs): Promise < CreatedItem >

createVersion()

Operation: Create a Version

createVersion(projectId , versionPayload , optionalArgs): Promise < CreatedVersion >

📁 ファイルをアップロード 🔍 検索				
☐ 名前へ	説明	バージ...	マーク...	サイズ
☐ 📁 ファミリタイプ選択用	--	--	--	--
☐ 🔄 サンプル設備.rvt		V2	🔗 👑	82.7 MB
☐ 📄 サンプル意匠.rvt		V1	🔗 👑	62.3 MB
☐ 📄 サンプル構造.rvt		V1	🔗 👑	48 MB

アップロード後の SVF/SVF2 変換と完了通知

WebHooks API

Model Derivative API

1. Autodesk Docs にアップロード完了

dm.version.added
dm.version.modified

2. 自動的に SVF に変換

extraction.updated
extraction.finished

変換の完了

GET:Manifest
Polling

3. SVF 変換後、自動的に SVF 2 に変換

extraction.updated
extraction.finished

- pending
- inprogress
- success
- failed
- timeout



Viewer と UI

Viewer と UI

98

 **AUTODESK**
Platform Services

 Sign in

Solutions ▾Getting StartedDocumentation ▾Success StoriesCommunity ▾Support ▾PricingApp Store ▾

Getting StartedEnvironment Setup**Tutorials**
Simple Viewer**Hubs Browser**
Application SetupAuthenticationData Browsing**Viewer & UI**
DashboardDesign AutomationACC AdministratorACC Issues (beta)Learn More

 > Tutorials > Hubs Browser > Viewer & UI

Viewer & UI

Finally, we're ready to build the client-side piece of our application.

 TIP

If you're developing with Node.js, you can use TypeScript definitions for the Viewer. Run

`npm install --save-dev @types/forge-viewer`

in your terminal to add the TypeScript definition file to your project.

Viewer logic

Let's start by implementing the Viewer functionality for our application. Create a `viewer.js` file under the `wwwroot` subfolder with the following code:

Viewer logicSidebar logicApplication logicUser interfaceTry it out

Viewer: 初期化

```
<div id="preview"></div>
```

```
document.getElementById('preview')
```

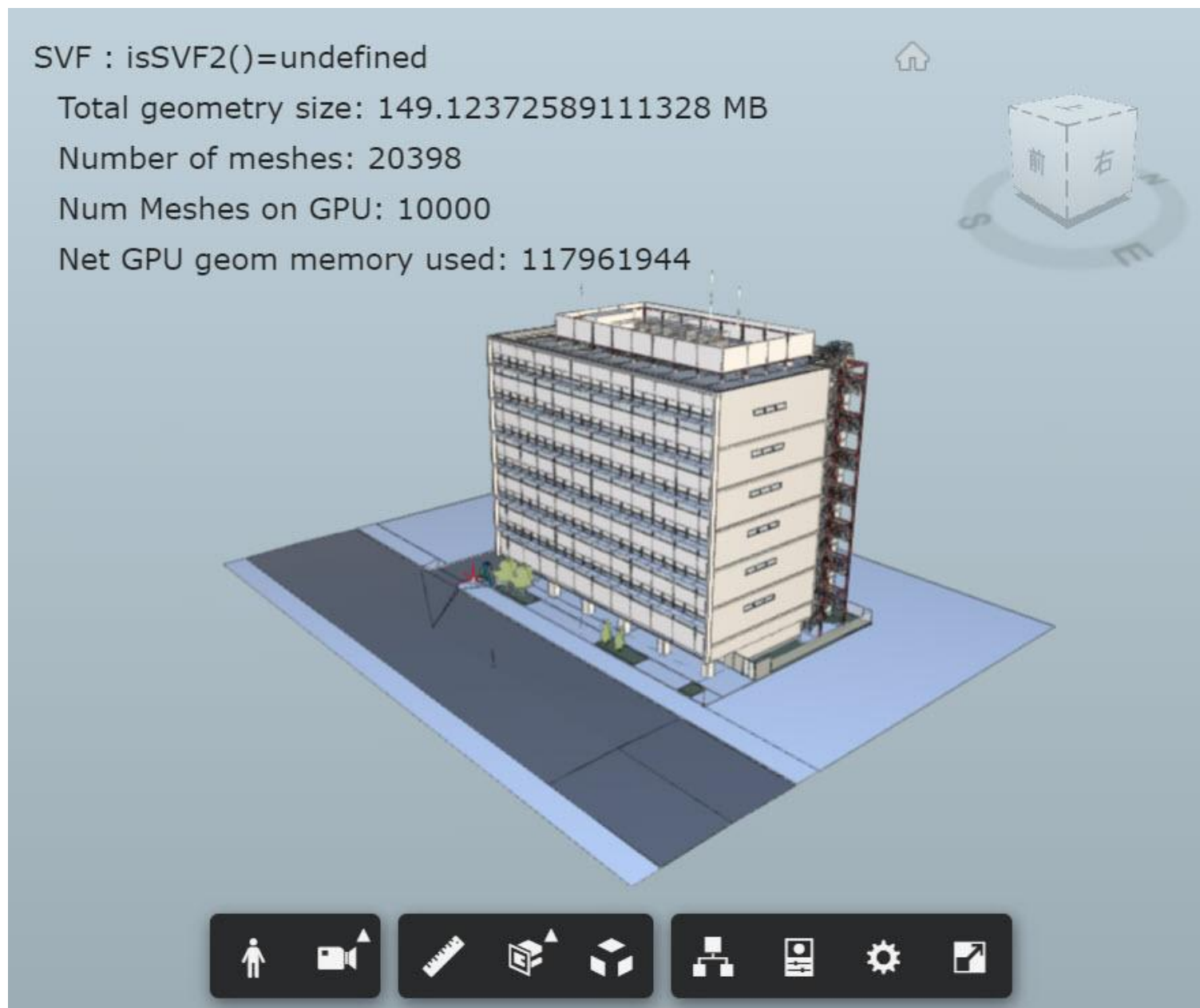
```
export function initView(container) {  
  return new Promise(function (resolve, reject) {  
    Autodesk.Viewing.Initializer({ env: 'AutodeskProduction', getAccessToken }, function () {  
      const config = {  
        extensions: ['Autodesk.DocumentBrowser']  
      };  
      const viewer = new Autodesk.Viewing.GuiViewer3D(container, config);  
      viewer.start();  
      viewer.setTheme('light-theme');  
      resolve(viewer);  
    });  
  });  
}
```

```
async function getAccessToken(callback) {  
  try {  
    const resp = await fetch('/api/auth/token');  
    if (!resp.ok)  
      throw new Error(await resp.text());  
    const { access_token, expires_in } = await resp.json();  
    callback(access_token, expires_in);  
  } catch (err) {  
    alert('Could not obtain access token. See the console for more details.');
```

**Scope
viewables:read**

```
    console.error(err);  
  }  
}
```

フォーマット: SVF vs SVF2



https://adndevblog.typepad.com/technology_perspective/2020/11/svf-and-svf2.html

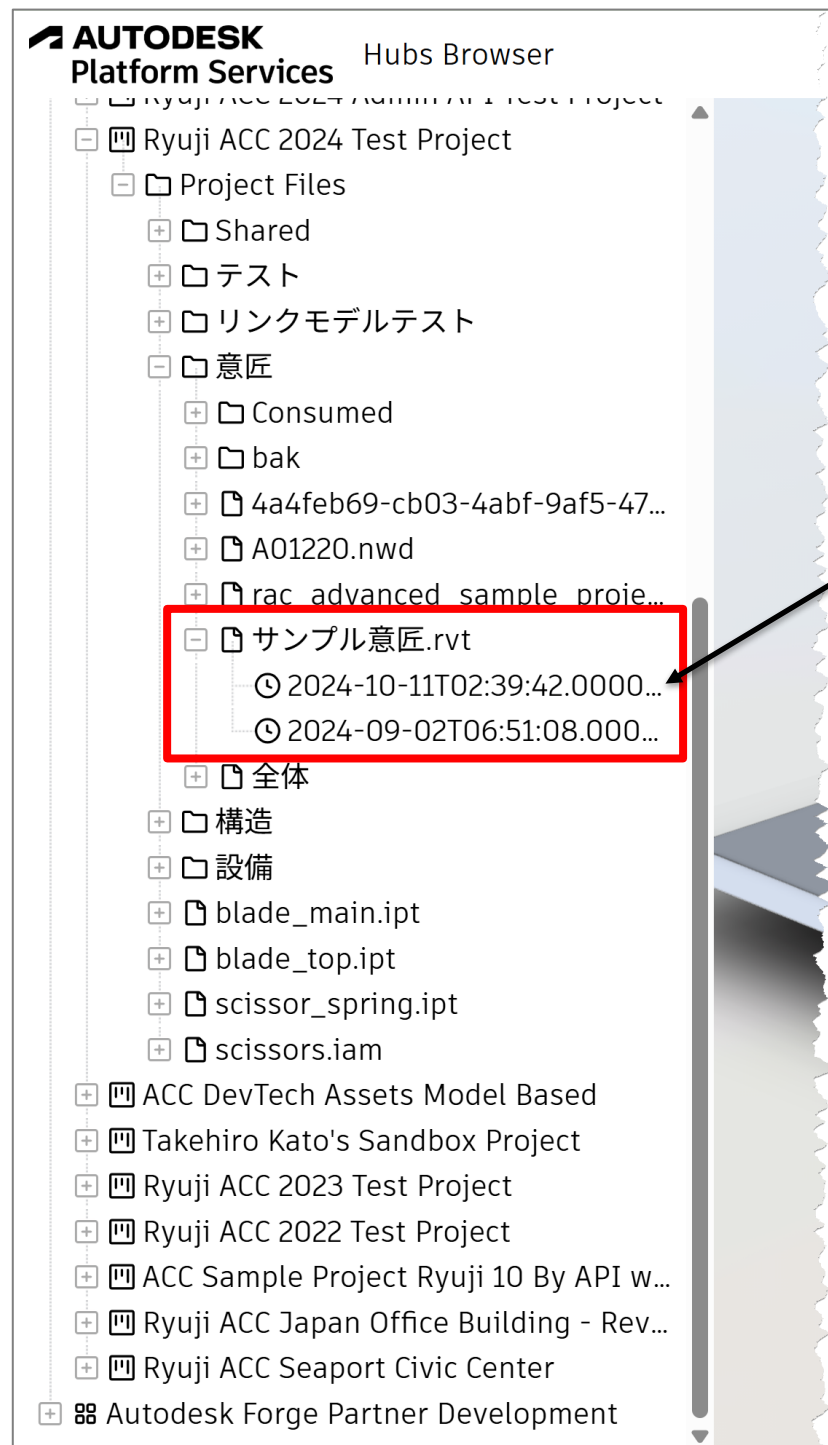
https://adndevblog.typepad.com/technology_perspective/2021/12/svf2-and-metadata-api-endpoints.html

Viewer: 初期化時の設定

```
var options = {  
  env: 'AutodeskProduction2',  
  api: 'streamingV2', // for models uploaded to EMEA change this option to 'streamingV2_EU'  
  getAccessToken: function(onTokenReady) {  
    var token = 'YOUR_ACCESS_TOKEN';  
    var timeInSeconds = 3600; // Use value provided by APS Authentication (OAuth) API  
    onTokenReady(token, timeInSeconds);  
  }  
};
```

Parameter	SVF	SVF2
env	AutodeskProduction	AutodeskProduction2
api	derivativeV2 (for US)	streamingV2 (for US)
	derivativeV2_EU (for EU)	streamingV2_EU (for EMEA)
	derivativeV2_AUS (for AUS)	streamingV2_AUS (for AUS)

Viewer: URN の取得と Base64 エンコード



- チュートリアルでは、ツリーコンポーネントにて versionId が保持されています。
- main.js 内で window.btoa() メソッドを使用。

```
<li class="collapsed selectable draggable drop-target leaf"
  data-uid="version | urn:adsk.wipprod:fs.file:vf.省略?version=2">
  <div class="title-wrap">
    <a class="title icon icon-version" tabindex="1" unselectable="on">
      2024-10-11T02:39:42.0000000Z
    </a>
  </div>
</li>
```



```
dXJuOmFkc2sud2lwcHJvZDpmcy5maWxlOnZmLlZHZUpyN
HFwUWJTekZPX0dRLWh1M0E/dmVyc2lvbj0y
```


Viewer: モデルのロードと AECModelData

```
export function loadModel(viewer, urn) {  
  function onDocumentLoadSuccess(doc) {  
    doc.downloadAecModelData( () => viewer.loadDocumentNode(doc, doc.getRoot().getDefaultGeometry()));  
  }  
  function onDocumentLoadFailure(code, message) {  
    alert('Could not load model. See console for more details.');    console.error(message);  
  }  
  Autodesk.Viewing.Document.load('urn:' + urn, onDocumentLoadSuccess, onDocumentLoadFailure);  
}
```

- Doc.downloadAecModelData()
 - IFC ファイルの場合は、ifcLevelsEnabled: true を設定すると、レベルの情報が抽出されます。
 - AECModelData は、レベル、通芯、フェーズ、ビューポートなどの情報を保持しています。



Viewer: AECModelData の取得方法

```
// From document
await doc.downloadAecModelData()
```

```
// From bubble
await Autodesk.Viewing.Document.getAecModelData(viewer.model.getDocumentNode())
```

[illegible]

チュートリアル：VS Code 上の操作手順

1. **wwwroot** フォルダ下に **viewer.js** ファイルを作成して実装
2. **wwwroot** フォルダ下に **sidebar.js** ファイルを作成して実装
3. **wwwroot** フォルダ下に **main.js** ファイルを作成して実装
4. **wwwroot** フォルダ下に **index.html** ファイルを作成して実装
5. **wwwroot** フォルダ下に **main.css** ファイルを作成して実装
6. デバッグ実行 >> localhost:8080

FileEditSelectionViewGoRunTerminalHelp

Hub Browser

06

EXPLORER

HUB BROWSER

.vscode

launch.json

node_modules

routes

auth.js

hubs.js

services

aps.js

wwwroot

.env

config.js

package-lock.json

package.json

server.js

JS aps.js

JS hubs.js

JS auth.js

JS server.js

JS config.js

routes > JS hubs.js > router.get('/api/hubs/:hub_id/projects/:project_id/contents') callback

```
17 router.get('/api/hubs/:hub_id/projects', async function (req, res, next) {
18     try {
19         const projects = await getProjects(req.params.hub_id, req.internalAuthToken.access_token);
20         res.json(projects.map(project => ({ id: project.id, name: project.attributes.name })));
21     } catch (err) {
22         next(err);
23     }
24 });
25
26 router.get('/api/hubs/:hub_id/projects/:project_id/contents', async function (req, res, next) {
27     try {
28         const entries = await getProjectContents(req.params.hub_id, req.params.project_id, req.query.folder_id, req.internalAuthToken.access_token);
29         res.json(entries.map(entry => ({ id: entry.id, name: entry.attributes.displayName, folder: entry.type === 'folders' })));
30     } catch (err) {
31         next(err);
32     }
33 });
34
35 router.get('/api/hubs/:hub_id/projects/:project_id/contents/:item_id/versions', async function (req, res, next) {
36     try {
37         const versions = await getItemVersions(req.params.project_id, req.params.item_id, req.internalAuthToken.access_token);
38         res.json(versions.map(version => ({ id: version.id, name: version.attributes.createTime })));
39     } catch (err) {
40         next(err);
41     }
42 });
43
44 module.exports = router;
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

powershell

15 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities

PS C:\Users\ogasawr\Hub Browser> npm install --save @aps_sdk/authentication @aps_sdk/data-management

added 44 packages, and audited 122 packages in 10s

18 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities

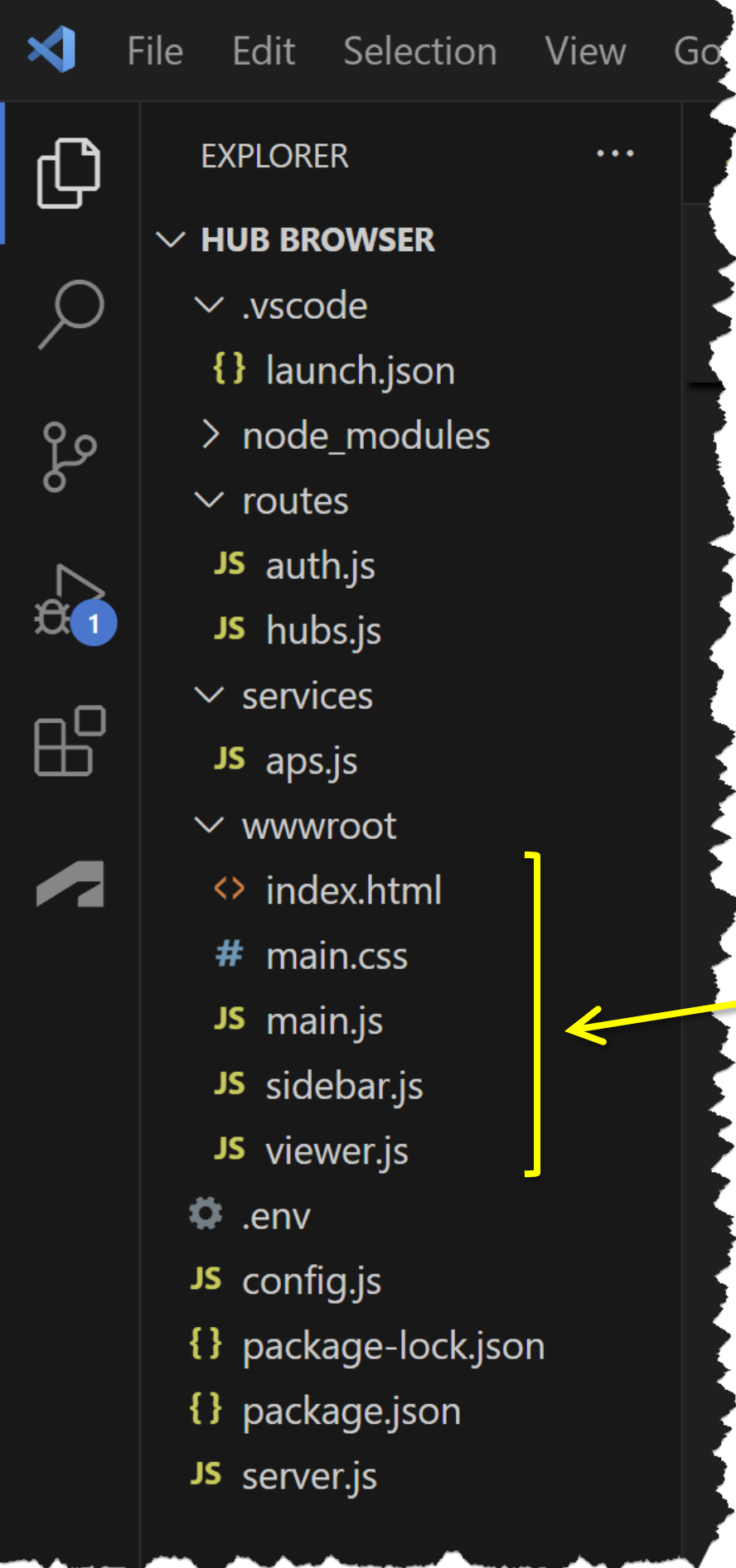
PS C:\Users\ogasawr\Hub Browser>

OUTLINE

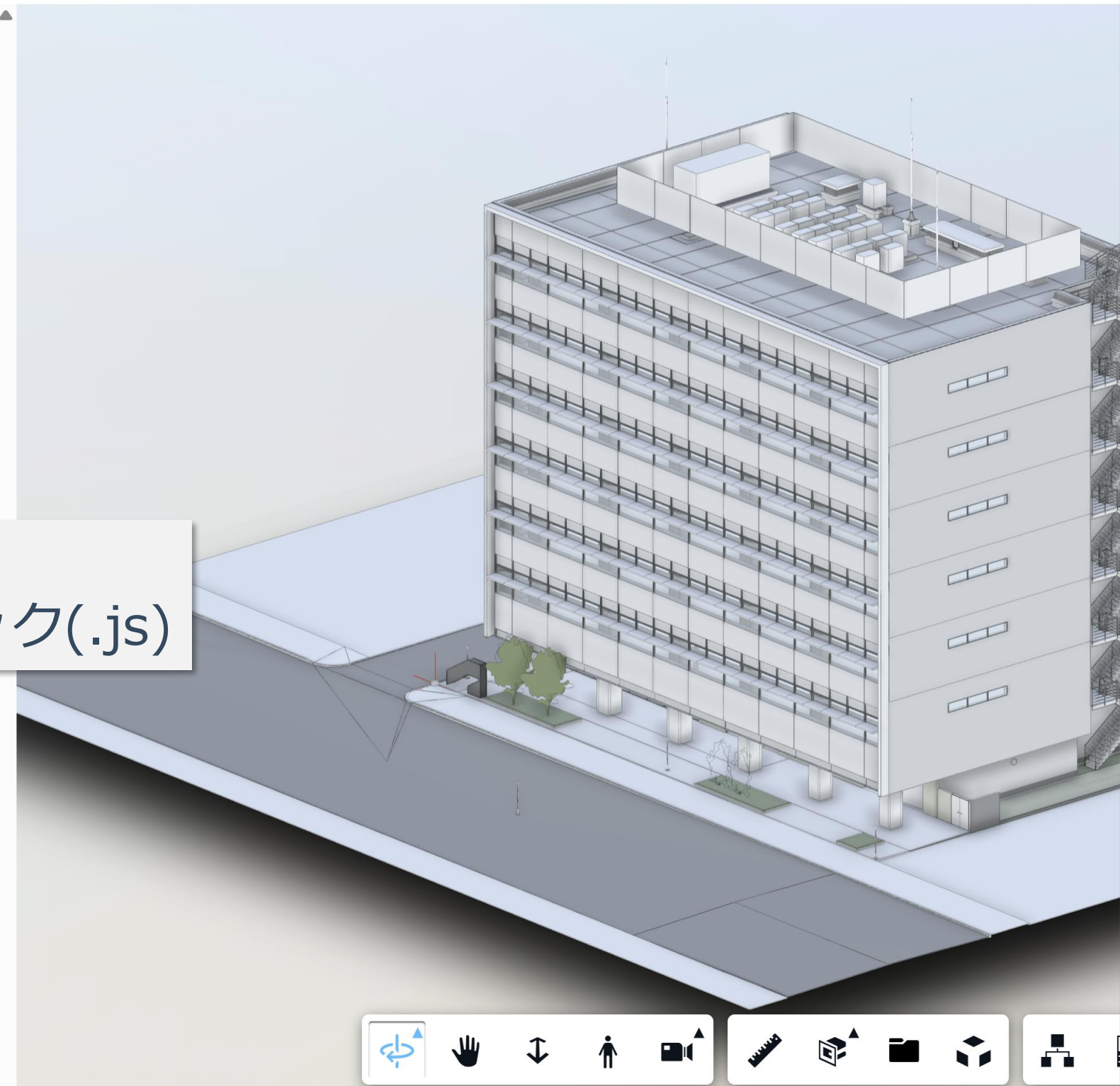
TIMELINE

Ln 27, Col 10 Spaces: 4 UTF-8 CRLF {} JavaScript

ここまでのプロジェクト構成

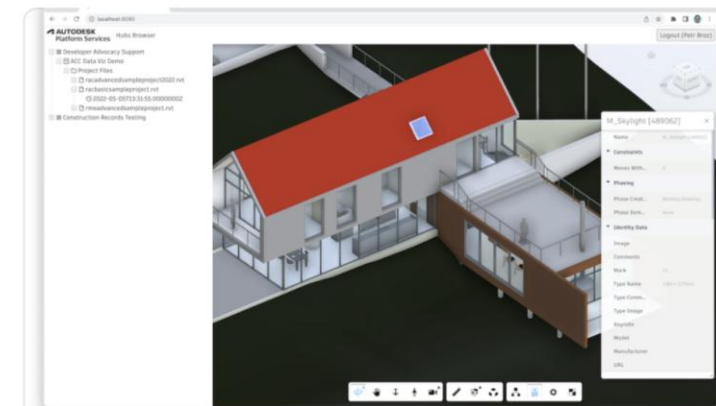


Web ページ定義一式
UI(.html、.css)とロジック(.js)



Hub Browser GitHub リポジトリ

- Code Sample : Hub Browser
 - Node.js ベース
 - APS SDK を使用
 - 3-legged 認証
 - Autodesk クラウドストレージにアクセス
 - Viewer 表示



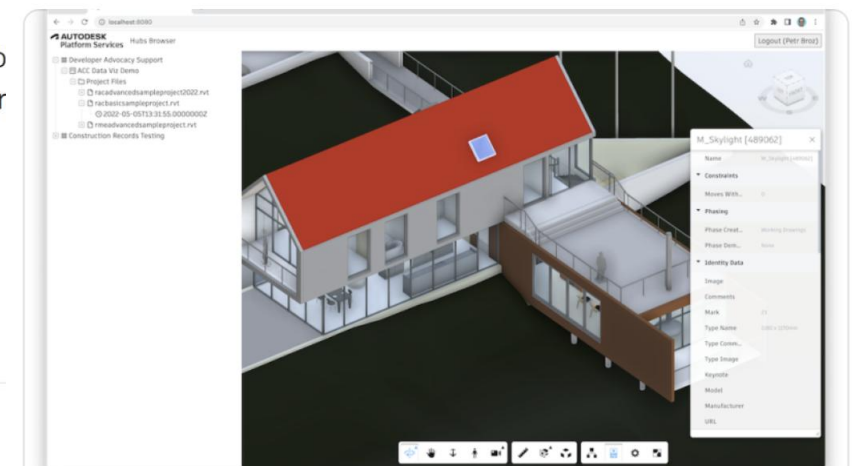
Hubs Browser (.NET)

Autodesk Platform Services applicatio
following the Hubs Browser tutorial fr

Data Management, Viewer, C#

[View in GitHub](#)

[Demo](#)



Hubs Browser (Node.js)

Autodesk Platform Services application built by
following the Hubs Browser tutorial from https

Data Management, Viewer, JavaScript

[View in GitHub](#)

[Demo](#)

Tips: Viewer Extension (機能拡張)

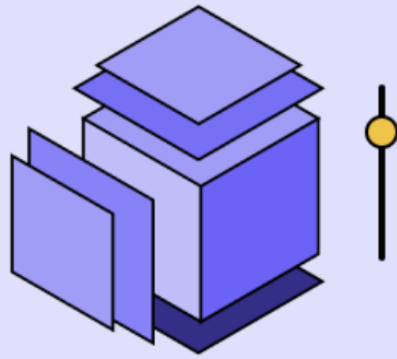
```
14 export function initView(container) {
15     return new Promise(function (resolve, reject) {
16         Autodesk.Viewing.Initializer({ env: 'AutodeskProduction', getAccessToken }, function () {
17             const config = {
18                 extensions: ['Autodesk.DocumentBrowser', 'Autodesk.AEC.Minimap3DExtension', 'Autodesk.AEC.LevelsExtension']
19             };
20             const viewer = new Autodesk.Viewing.GuiViewer3D(container, config);
21             viewer.start();
22             viewer.setTheme('light-theme');
23             resolve(viewer);
24         });
25     });
26 }
27
```

- Autodesk 公開の Extension をロード
 - <https://aps.autodesk.com/en/docs/viewer/v7/reference/Extensions/>
- 独自に Extension を作成してロード
 - <https://get-started.aps.autodesk.com/tutorials/dashboard/>
 - [Writing an Extension](#)

- + 88 Ryuji Ogasawara
- 88 Developer Advocacy Support
 - + 88 Ryuji ACC Worksharing 2024
 - + 88 Ryuji ACC Japan Office Building
 - + 88 ACC Sample Project Ryuji 8 By API wi...
 - + 88 Component Library Project 024d6467...
 - + 88 Sample Project
 - + 88 ACC Sample Project Ryuji 11 By API w...
 - 88 Ryuji ACC 2025 Test Project
 - 88 Project Files
 - 88 意匠
 - 88 サンプル意匠.rvt
 - 🕒 2025-04-16T01:30:14.0000...
- + 88 Ryuji ACC Test Project
- + 88 ACC-point-clouds
- + 88 Ryuji ACC Design Collaboration Revit ...
- + 88 Ryuji ACC 2024 Template Project
- + 88 Ryuji ACC 2024 Template
- + 88 ACC Sample Project Ryuji 9 By API wi...
- + 88 ACC Sample Project Ryuji 12 By Manu...
- + 88 Ryuji ACC 2024 Admin API Test Project
- + 88 Ryuji ACC 2024 Test Project
- + 88 ACC DevTech Assets Model Based
- + 88 Takehiro Kato's Sandbox Project
- + 88 Ryuji ACC 2023 Test Project
- + 88 Ryuji ACC 2022 Test Project
- + 88 ACC Sample Project Ryuji 10 By API w...
- + 88 Ryuji ACC Japan Office Building - Rev...
- + 88 Ryuji ACC Seaport Civic Center
- + 88 Autodesk Forge Partner Development

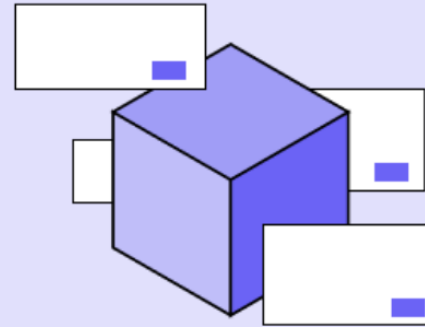


Interactive Examples



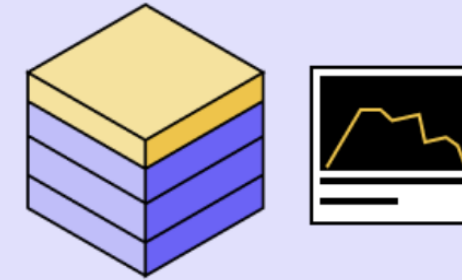
Controlling Viewer states

Learn to programmatically control Viewer states, like exploding a model.



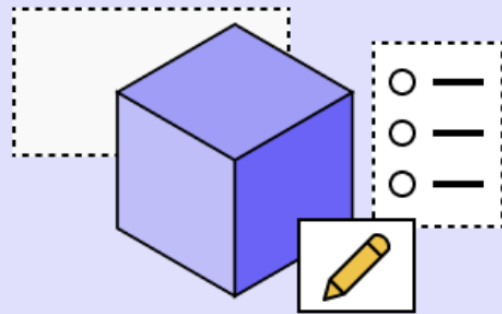
Handling Viewer events

Learn to respond to users' actions and perform tasks based on specific triggers.



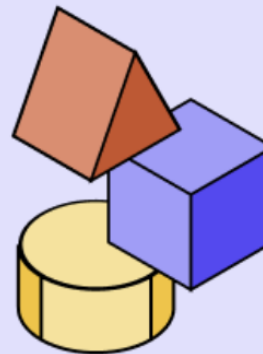
Querying model properties

Add functionality to search for specific model property values and display results in a dialog window.



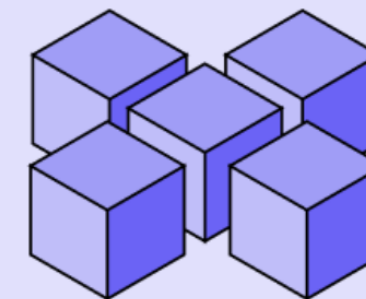
Customizing Viewer UI

Create custom UI elements for unique interactions, like a button that displays a dialog window when clicked.



Customizing UI scene

Create and interact with custom three.js geometry that integrates like standard Viewer models.



Model aggregation

Combine a Revit model and a Fusion 360 model in the same view.

Tips: Revit モデル固有の要素の表示

- 「部屋」「スペース」「ゾーン」を 3D ビュー上で表示できます。ただし、デフォルトでは非表示。
- ACC / BIM 360 では、**generateMasterViews** を true に設定して変換されます。
 - Revit モデルで作成したフェーズ毎に、「部屋」などの要素を含むマスタービュー（3D ビュー）が生成されるようになります。
- マスタービューの表示名は、既定値として Revit モデルのフェーズ名になります。

AUTODESK Platform Services Hubs Browser Logout (Ryuji Ogasawara)

モデル ×

検索

- ▶ ドア (129)
- ▶ カーテン パネル (104)
- ▼ 部屋 (113)
 - 事務室 1 [52842126]
 - EPS 2 [52842128]
 - 階段室 3 [52842130]
 - ELVホール 5 [52842134]
 - WC女 6 [52842136]
 - EPS 7 [52842138]

倉庫 130 [54248667] ×

Name	倉庫 130 [54248667]
▼ 識別情報	
番号	130
名前	倉庫
仕上表グループ名	Varies
イメージ	
コメント	
用途	
部署	

Tips: Revit モデル固有の要素の表示

- マスタービューの検索に Special Id を使用できます。
 - '08f99ae5-b8be-4f8d-881b-128675723c10'

```
28 export function loadModel(viewer, urn) {
29   function onDocumentLoadSuccess(doc) {
30     const root = doc.getRoot();
31     const viewables = root.search({'type': 'geometry', 'role': '3d'});
32     console.log('Viewables:', viewables);
33     const phaseViews = viewables.filter(v => v.data.name === v.data.phaseNames &&
34       | v.getViewableRootPath().includes('08f99ae5-b8be-4f8d-881b-128675723c10'));
35     console.log('Master Views:', phaseViews);
36     viewer.loadDocumentNode(doc, phaseViews[0], { skipHiddenFragments: false });
37   }
38   function onDocumentLoadFailure(code, message) {
39     alert('Could not load model. See console for more details.');
```

```
40     console.error(message);
41   }
42   Autodesk.Viewing.Document.load('urn:' + urn, onDocumentLoadSuccess, onDocumentLoadFailure);
43 }
```

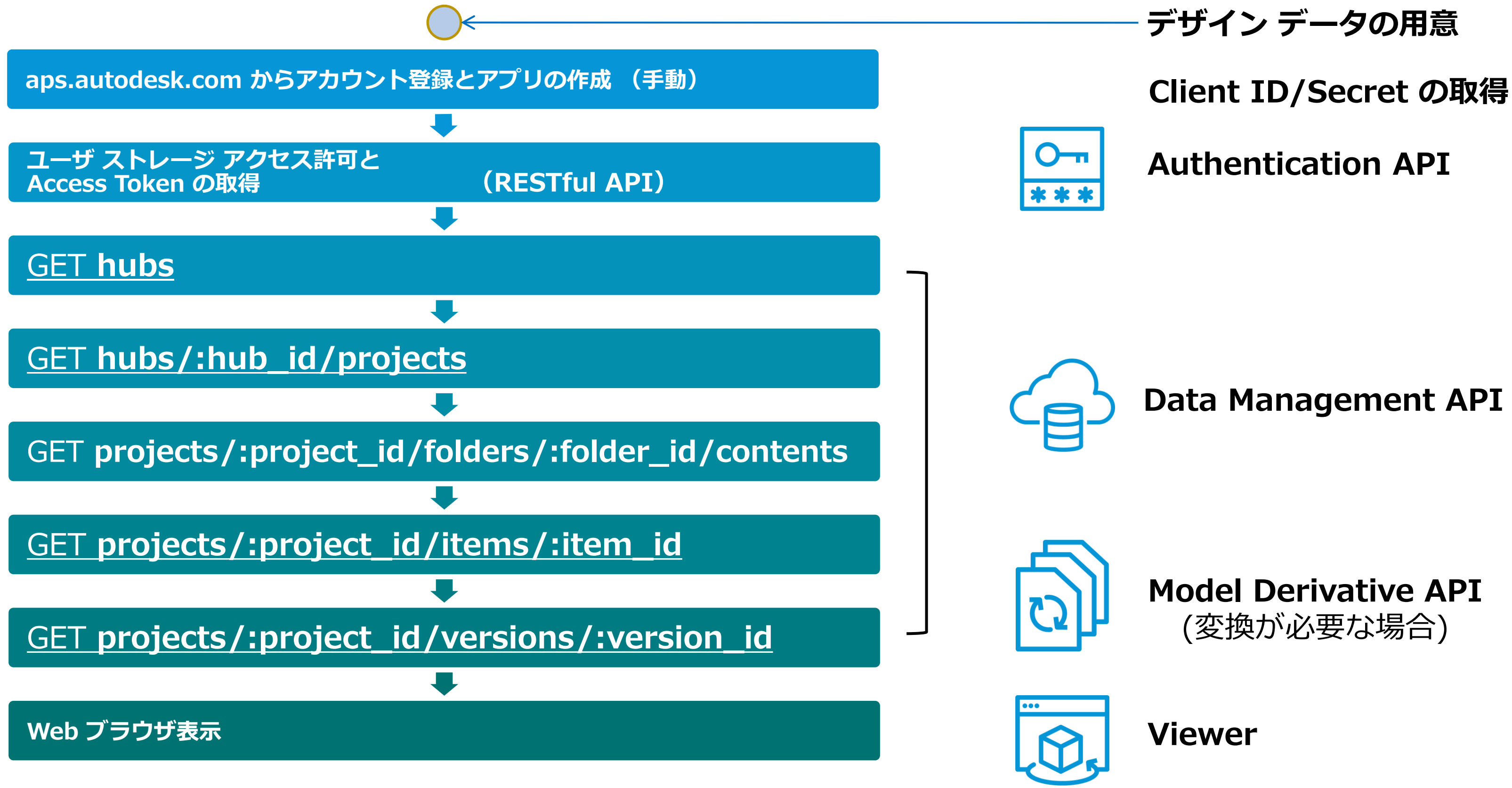
Tips: 要素の識別

```
{
  "objectid": 1706, = dbId
  "name": "11030_FAN_消音ボックス付送風機 [17299672]",
  "externalId": "706c1fe4-e098-4258-8c15-25674aab130c-0107f8d8", = Revit の場合は Element.UniqueId
  "properties": {
    "拘束": {
      "風量": "400.000 CMH",
      "機外静圧": "118.000 Pa",
      "基準レベル": "1F",
      "基準レベルからの高さ": "3327.500 mm",
      "ホスト": "レベル : 1F",
      "ホストからのオフセット": "3327.500 mm",
      "既定の高さ": "0.000 mm",
      "番手": "#1 1/4"
    },
    "文字": {
      "☑給気/☐排気": "Yes",
      "特殊仕様": "",

```

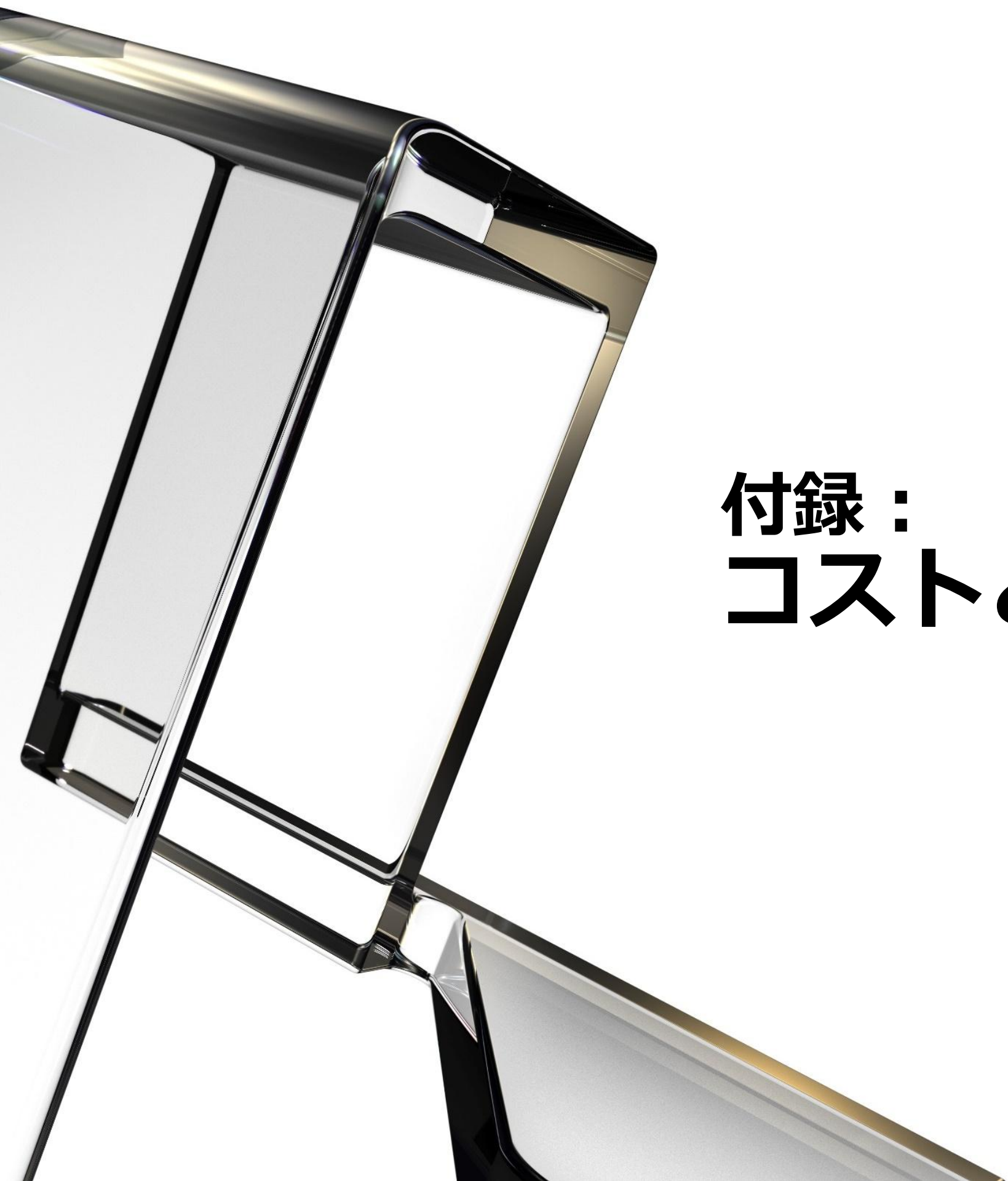
- objectid / dbId
APS Viewer が要素を識別するための ID。
特定の要素のプロパティにアクセスする際などにも利用。
- externalId
Revit の UniqueId に保持されている GUID と同じ値。

3-legged OAuthでの Viewer 利用手順の理解



ご注意：進化し続ける APS・開発作業に終わりはなし クラウドサービス

- APS の機能向上によって動作/仕様が変わる可能性
 - 例) [Design Automation API への OAuth スcope code:all 適用](#)
 - APS Viewer は特定バージョン指定も可能
- クラウド開発にはメンテナンスが必須
- デスクトップ製品のアドイン開発とは異なります！
- 開発ベンダーとのサブスクリプション？契約が必要！！



付録： コストとサポート

APS 利用に対する '課金' とは？

<https://aps.autodesk.com/pricing>

- 特定の APS API 利用に **Autodesk Flex** による従量課金

プレミアムAPIとサービス

コスト 情報

Design Automation API

2.0

Flex トークン / 処理時間

Flow Graph Engine API

1.0

Flex トークン / 処理時間

Model Derivative API

0.5

Flex トークン / コンプレックスジョブ

0.1

Flex トークン / シンプルジョブ

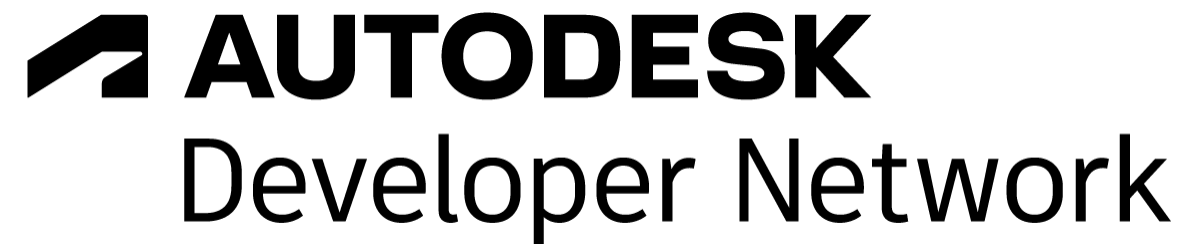
Reality Capture API

1.0

50枚の写真毎のFlexトークン



Autodesk Developer Network (ADN) とは



- **年間契約の開発サポートプログラム（有償）です**
 - 新製品/バージョン、API、テクノロジーの啓蒙と流布
 - 無償ブログ公開、ウェビナー等イベント開催、その他
 - ADN 加入メンバーへの API 開発サポートの提供
 - API の問題点、不具合、要望の米国本社へのフィードバック
 - Autodesk App Store アプリの公開審査とサポート

ADN 特典

1. ADN Extranet（会員専用サイト）アクセスを提供
2. オートデスク API についてのお問合せ窓口提供
3. 開発用途の製品ライセンスを提供
4. マーケティング用ロゴの提供
5. ユーザー数に応じた Autodesk Flex の提供
6. 開発者向け有償イベントへの優待価格提供

今後のリソース

- サンプル コード : GitHub
 - <https://github.com/Developer-Autodesk>
 - <https://github.com/Autodesk-APS>
- サポート : StackOverflow
 - <https://APS.autodesk.com/en/support/get-help>
- ブログ : APS Community Blog
 - <https://APS.autodesk.com/blog>

