



AUTODESK Platform Services

APS Online Training – AutoCAD Automation

伊勢崎 俊明

Developer Advocacy & Support、Autodesk Developer Network

アジェンダ – ご紹介する内容

- はじめに
- 理由と仕組み
- AutoCAD Automation の理解
- デベロッパキーの取得
- サーバーの作成
- 基本 UI
- プラグイン（アドイン）の作成
- Activity（アクティビティ）の定義
- WorkItem（ワークアイテム）の実行



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



RESTful API
GET PUT POST DELETE



Autodesk Platform Services – 主要 API

Core APIs



Model
Derivative



Automation



Viewer



Authentication



Webhooks

Data APIs



AEC
Data Model



Data
Management



Manufacturing
Data Model



Data
Exchange

Product APIs



Forma



Fusion



Reality
Capture



ACC

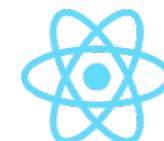


BIM 360



Tandem

RESTful API
GET PUT POST DELETE

 React

 ANGULARJS

 Microsoft .NET

 GraphQL

HTML  CSS  JavaScript 

 node.js™

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

The screenshot shows the Autodesk Platform Services website. At the top, there is a navigation bar with links for Solutions, Getting Started, Documentation (which is currently selected), Success Stories, Community, Support, Pricing, and App Store. A search bar is located above the navigation bar. On the left side, there is a large banner with the text "APIs to design anything" and a subtext: "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." Below the banner, there is a dropdown menu with options "APIs & SDKs" and "Code Samples", with a blue arrow pointing to "APIs & SDKs". On the right side of the page, there is a photograph of a woman holding a 3D-printed mechanical part, with colorful lines overlaid on the image to represent data flow or design elements.

APS サンプル – GitHub リポジトリ

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

The screenshot shows the Autodesk Platform Services website. At the top, there is a navigation bar with links: Solutions, Getting Started, Documentation, Success Stories, Community, Support, Pricing, and App Store. The Documentation link is currently selected, indicated by a dropdown menu that appears when the mouse hovers over it. This menu contains two items: APIs & SDKs and Code Samples, with a blue arrow pointing to the latter. Below the navigation bar, the main content area features a large image of a woman looking down at a screen, with the text "APIs to design" overlaid. In the bottom right corner of this content area, there is another blue arrow pointing to the "Code Samples" link in the footer navigation. The footer itself contains several sections: a logo for Autodesk, links to Company overview, Careers, Investor relations, and Newsroom under the "Solutions" section; links to AEC Data Model, Autodesk Construction Cloud, Autodesk Fusion, BIM 360, and BuildingConnected API under the "Documentation" section; links to Resources like Get Help, API Status, Blog, FAQ, and Code Samples under the "Resources" section; and links to About APS, Pricing, Success Stories, Certified Partners, and Partnerships under the "About" section. The "Code Samples" link in the footer is highlighted with a blue arrow.

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

学習リソース : Learn APS Tutorial (英語のみ)

<https://tutorials.autodesk.io/>

Get the tools 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

学習リソース : Learn APS Tutorial

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

The screenshot shows the Autodesk Platform Services website with a black header. The header includes the Autodesk logo, a search bar, and a 'Sign in' button. Below the header is a navigation menu with links: Solutions, Getting Started, Documentation, Success Stories, Community, Support, Pricing, and App Store. On the left side, there is a sidebar with a list of tutorials under 'Tutorials'. The 'Design Automation' section is highlighted with a blue bracket. The main content area shows the 'Design Automation' tutorial page. The breadcrumb navigation shows: Home > Tutorials > Design Automation. The main title is 'Design Automation'. A sub-section titled 'Introduction' is shown with the following text: 'This tutorial will guide you through creating a webapp that can upload an input file, change its width and height parameter and save the output file. The UI will let you define AppBundle & Activity (as an initial configuration) and execute Workitems, with an output result.' Below this, another section states: 'Both input and output files are saved in OSS Buckets, you can use Simple Viewer tutorial to view them.' At the bottom, there is a screenshot of a browser window showing a form to upload an input file with fields for 'Width' (set to 80) and 'Height'. The status message in the browser console says 'Uploading input file... Workitem started: 48eaa0bdec344d598e347844fed26642 { "status": "success", ... }'.

- Getting Started
- Environment Setup
- Tutorials**
- Simple Viewer
- Hubs Browser
- Dashboard
- Design Automation**
- Create Server
- Basic UI
- Create Plugin
- Define Activity
- Execute Workitem

Home > Tutorials > Design Automation

Design Automation

Introduction

This tutorial will guide you through creating a webapp that can upload an input file, change its width and height parameter and save the output file. The UI will let you define AppBundle & Activity (as an initial configuration) and execute Workitems, with an output result.

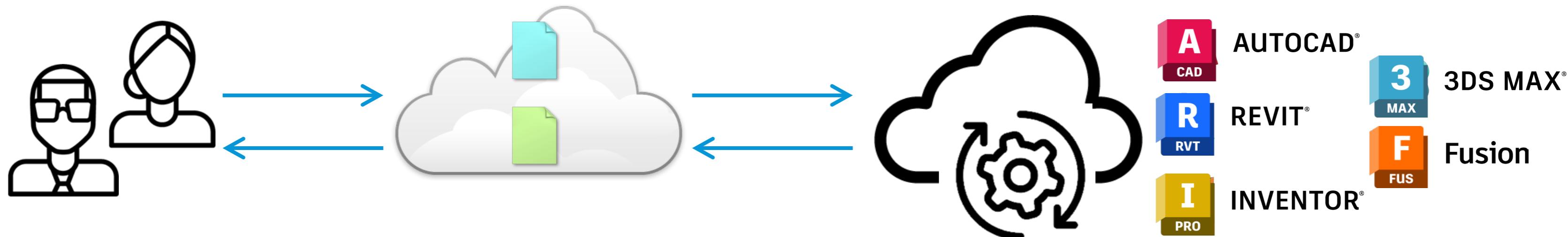
Both input and output files are saved in OSS Buckets, you can use Simple Viewer tutorial to view them.

A screenshot of a browser window titled 'Autodesk Forge - Design Automation'. The address bar shows 'localhost:3000'. The page content includes a form with 'Width:' set to 80 and 'Height:' set to 0. A status message in the browser's developer tools console indicates a successful upload: 'Uploading input file... Workitem started: 48eaa0bdec344d598e347844fed26642 { "status": "success", ... }'.

2025年7月1日（日本標準時） - 6月30日（米国太平洋標準時）～ Design Automation API 名称変更

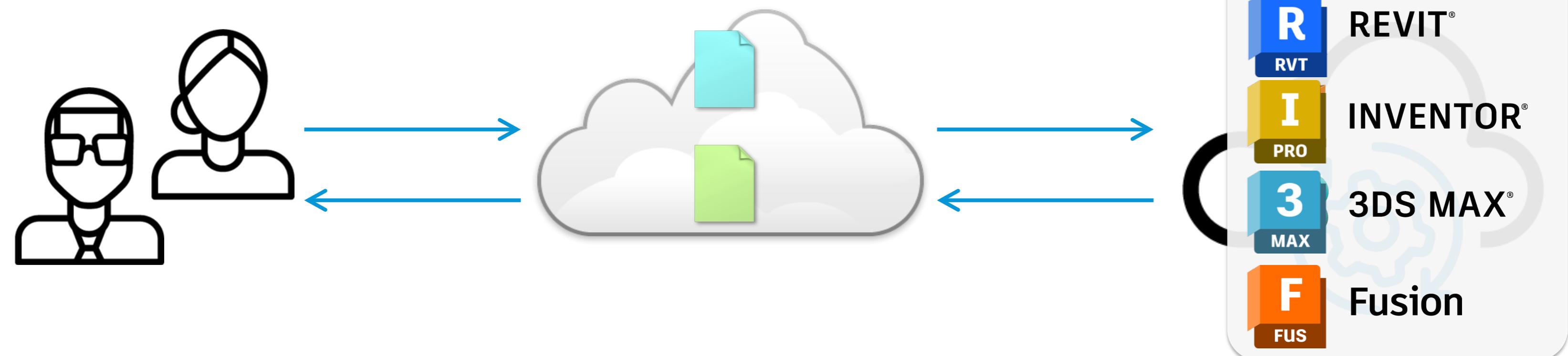
- Design Automation API → Automation API

旧呼称（エンジン別）	新呼称（エンジン別）
Design Automation API for AutoCAD	AutoCAD Automation API
Design Automation API for Revit	Revit Automation API
Design Automation API for Inventor	Inventor Automation API
Design Automation API for 3ds Max	3ds Max Automation API
Design Automation API for Fusion	Fusion Automation API



自動化ソリューション

- **Automation API (旧名 Design Automation API)**
- CAD エンジンでアドインを実行する仕組み
 - 対面操作を意図した‘製品’ではない
 - UI を持たないためオーバーヘッドなし
 - 製品一般規約による制限事項は不適用

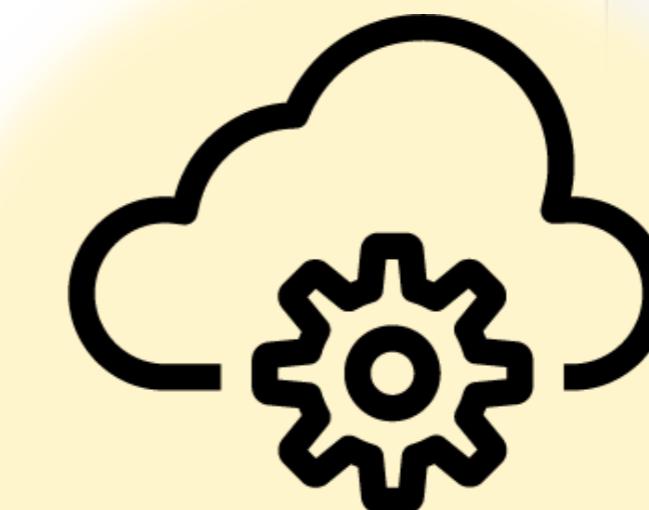


自動化ソリューション

- Automation API (旧名 Design Automation API)
- CAD エンジンでアドインを実行する仕組み
 - 製品一般規約による制限事項は不適用
 - UI を持たないためオーバーヘッドなし



Data Management API
データアクセス



Automation API
アドイン自動処理

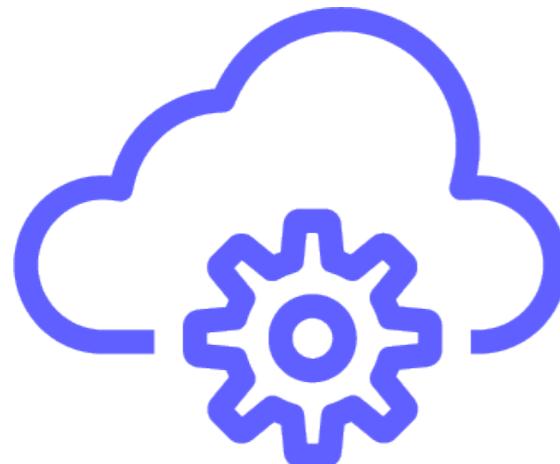


Automation API

Web API + デスクトップ API の知識が必要

APS 開発

 **AUTODESK**
Platform Services



製品別アドイン開発



Forge SDK : 2016 年 ~

- エンドポイントをラップした環境毎の‘ライブラリ’
 - Node.js、.NET、PHP、Java、(Python) ...
 - Design Automation API v2 (AutoCAD のみ) 対応
 - Design Automation API v3 対応は別 SDK で対応
- その後の暫定処置
 - Node.js、.NET のみ
 - Detect-to-S3 アプローチ 対応
 - Authentication v2 対応
- 現在、GitHub リポジトリは‘アーカイブ’状態
 - 今後の機能面の変更は後続の APS SDK のみで対応予定

APS SDK : 2024 年～

- 本 Design Automation Tutorial では未使用
- エンドポイントをラップした環境毎の‘ライブラリ’
 - 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>



2016年 Forge 正式リリース

2019年 Design Automation v3 リリース

Forge SDK

Forge Design Automation SDK - v3用

APS SDK

2016 2017

... ...

2023 2024

APS SDK ドキュメント

The screenshot shows the 'Developer's Guide' section of the Autodesk Platform Services documentation for the Model Derivative API. The page title is 'About the Model Derivative API'. On the left, there is a sidebar with navigation links: 'Developer's Guide' (selected), 'Overview', 'API Basics', 'GDPR Compliance', 'Webhooks', 'Field Guide', 'Supported Translations', 'Rate Limits', 'How-to Guide', 'Code Samples', 'Reference Guide' (highlighted with a red dashed box), and 'Change History'. The main content area shows a flowchart illustrating the Model Derivative API architecture. It starts with 'Source Files' (represented by a document icon) which feeds into 'Input' (represented by a folder icon). From 'Input', three arrows point to 'REST API Reference', '.NET SDK Reference', and 'TypeScript SDK Reference' (each with a blue arrow pointing to its respective link). From 'Input', another arrow points to a yellow box containing 'Geometry Extraction' (represented by a cube icon) and 'Data Extraction' (represented by a bar chart icon). A large blue bracket on the right side groups the three SDK references under the heading 'エンドポイントリファレンス' (Endpoint References).

AUTODESK Platform Services

Solutions Getting Started Documentation Success Stories Community Support Pricing App Store

Developer's Guide

Documentation / Model Derivative API / Developer's Guide

About the Model Derivative API

Overview

> API Basics

GDPR Compliance

Webhooks

Field Guide

Supported Translations

> Rate Limits

How-to Guide

Code Samples

Reference Guide

> REST API Reference

> .NET SDK Reference

> TypeScript SDK Reference

Change History

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

Geometry Extraction

Data Extraction



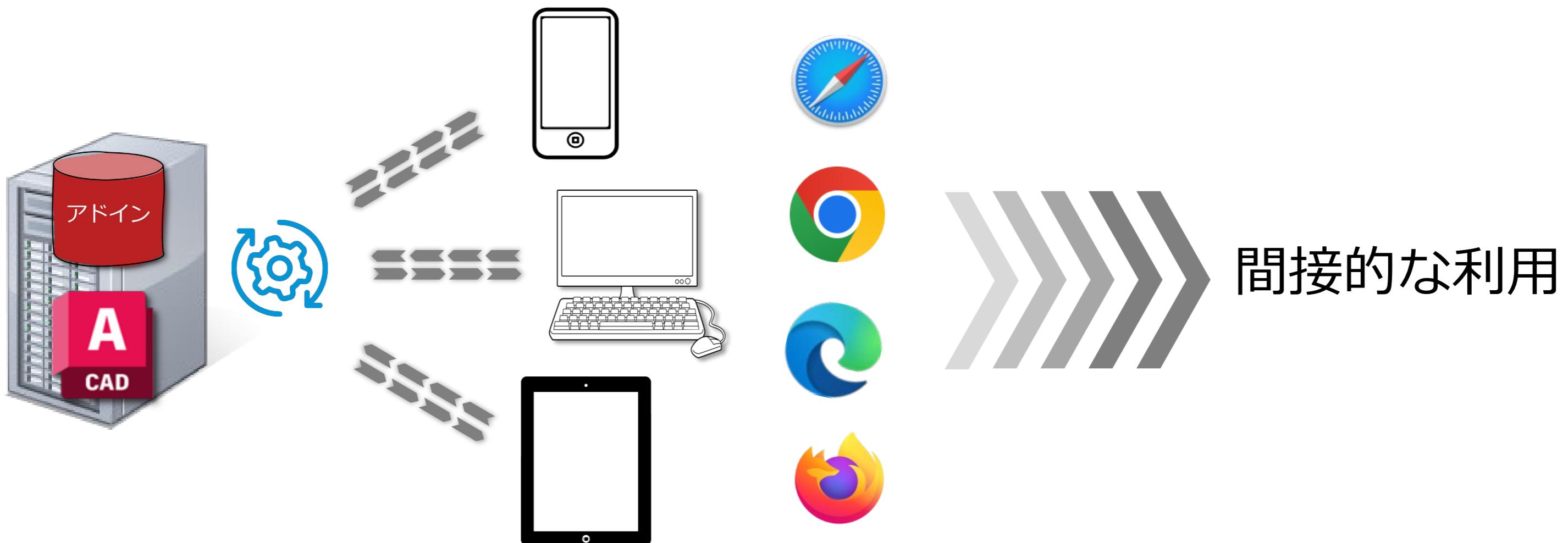
はじめに～理由と仕組み

AutoCAD アドインの目的？

- 作図補助機能の提供
 - 反復タスクの自動化の提供
 - 図面内情報を利用した外部システムの連携
- 
- 生産性の向上

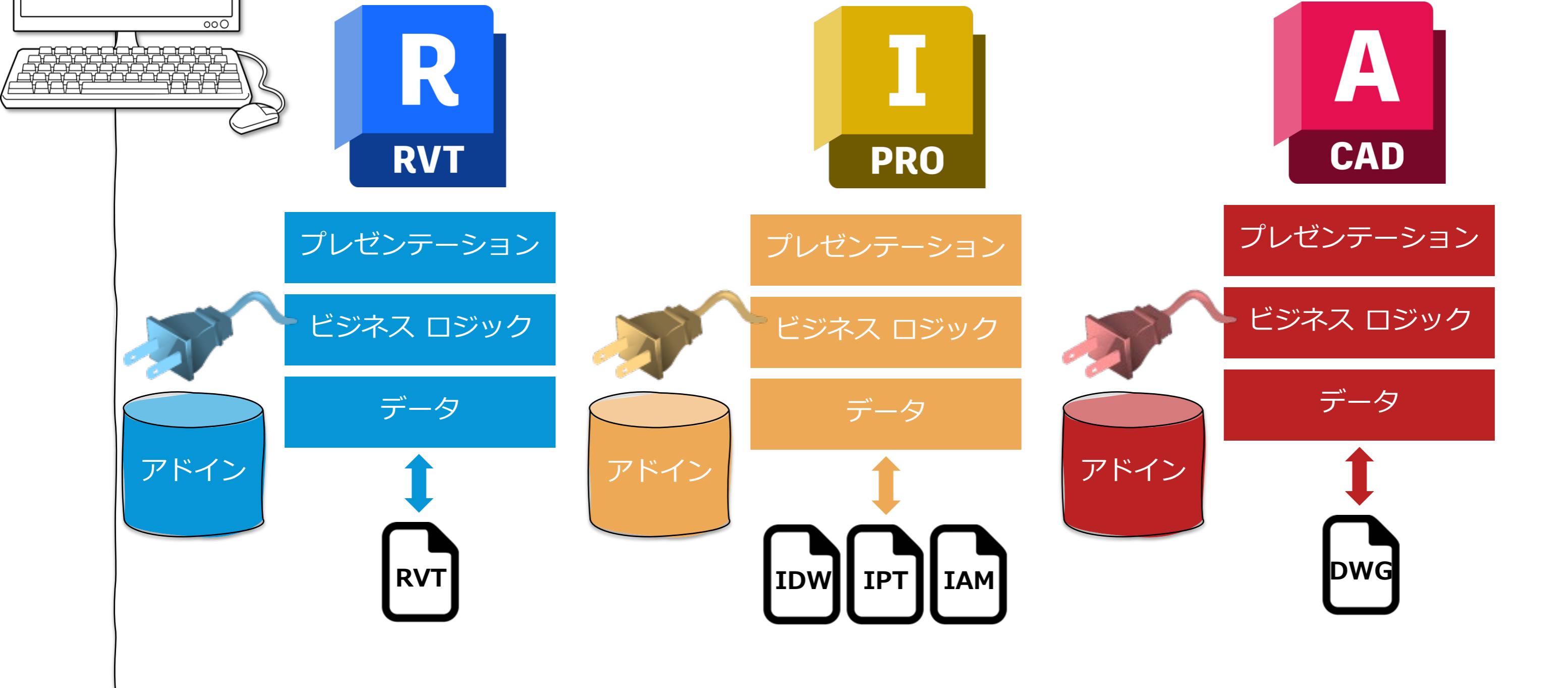
AutoCAD アドイン運用の多様化

- インターネット - クラウドの普及にともなう要求の変化



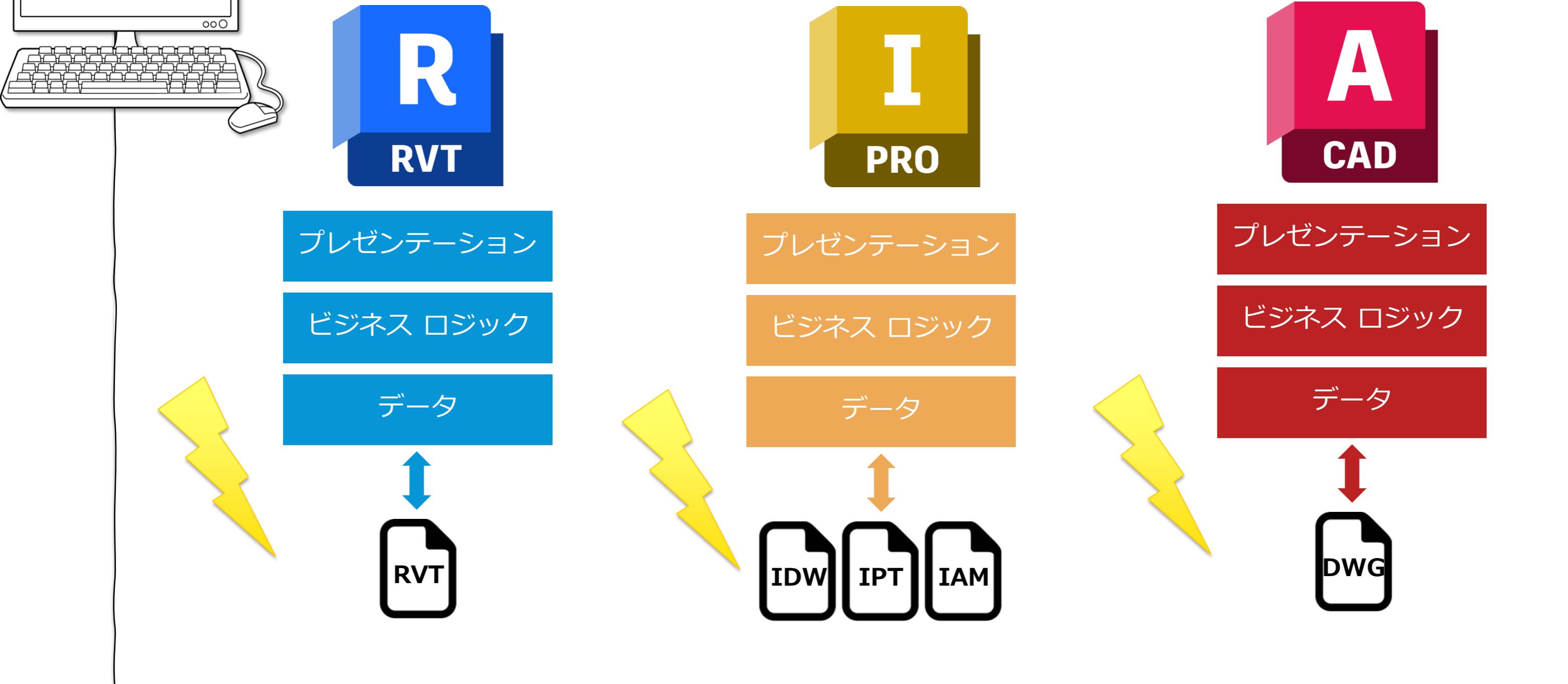
オートデスク デスクトップ製品の API で出来ること

デスクトップ製品内の自動化を実現

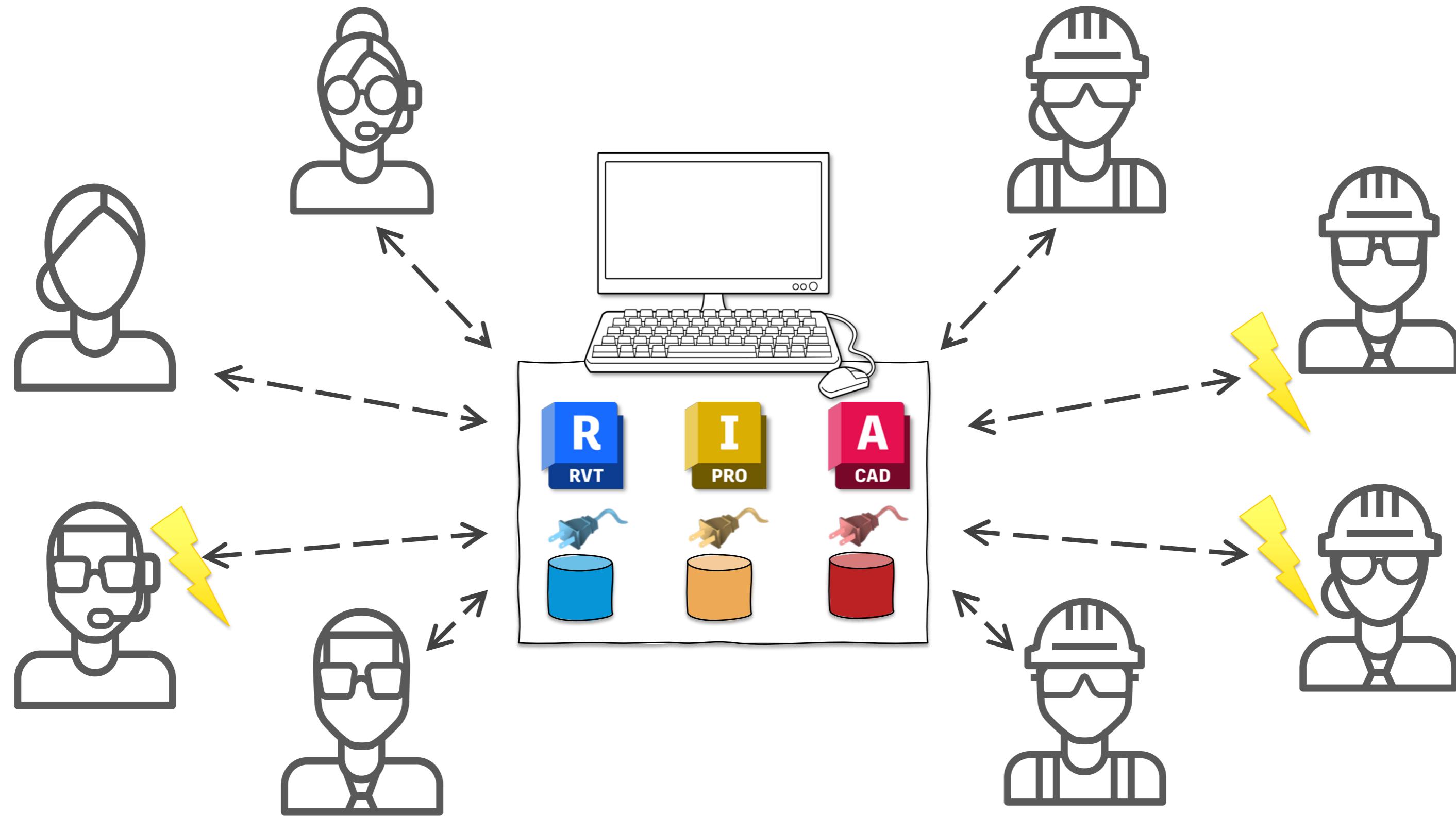


Windows の API で出来ること

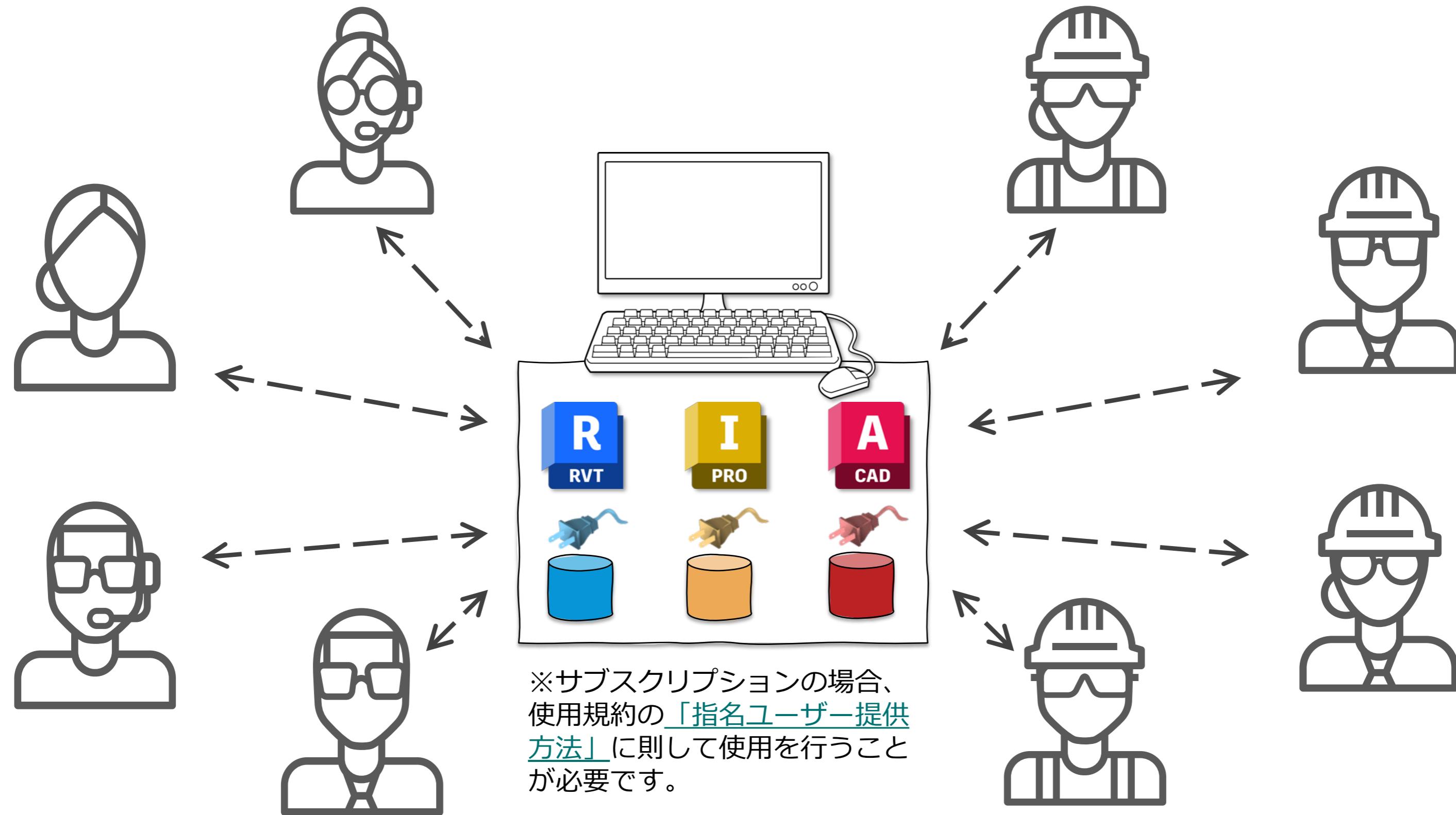
デスクトップ製品の外部からの起動を実現



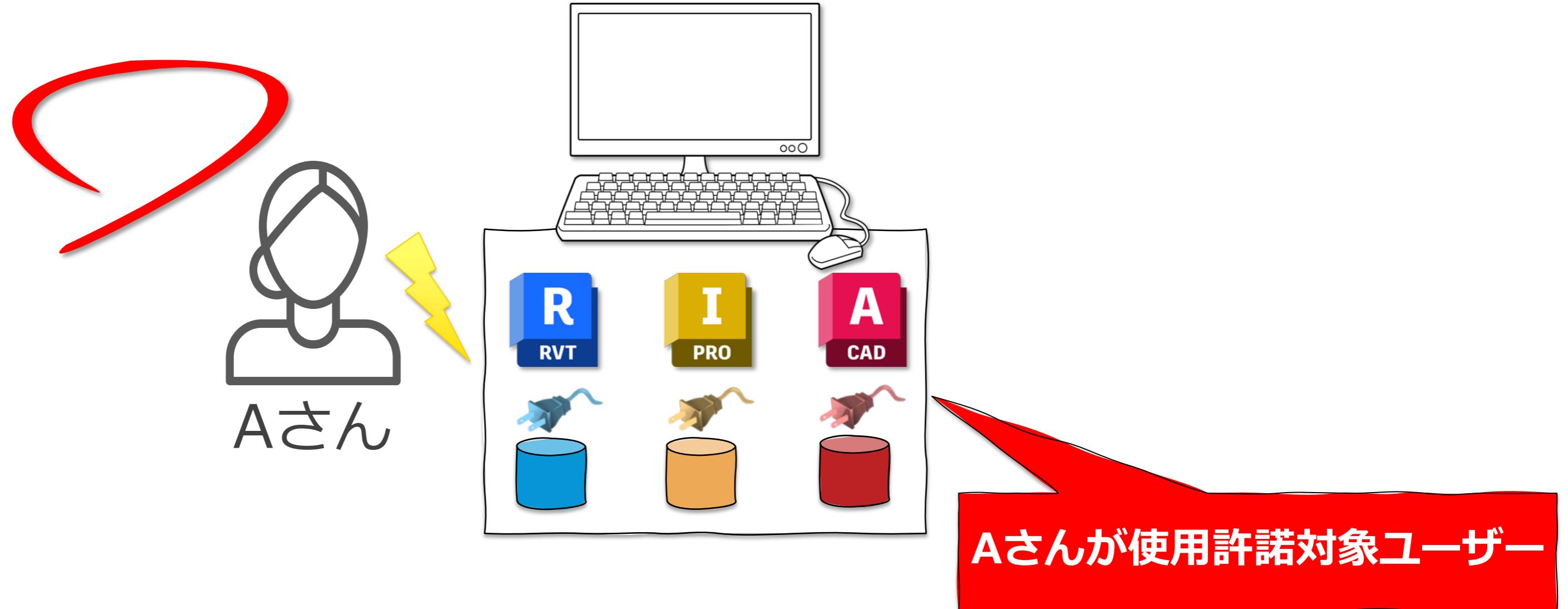
デスクトップ製品を利用した Web 利用の可否？



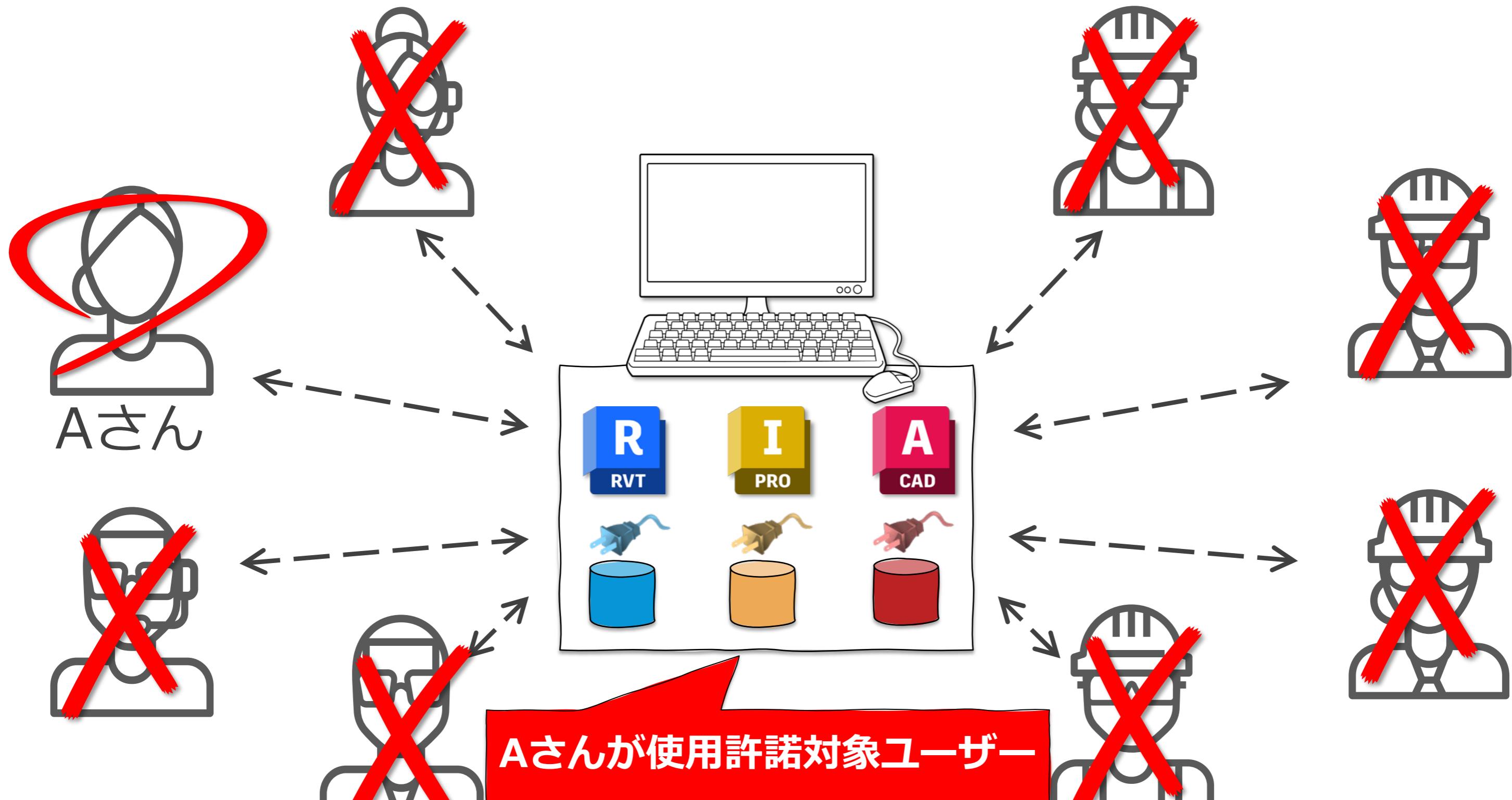
デスクトップ製品は誰にライセンスされているか？

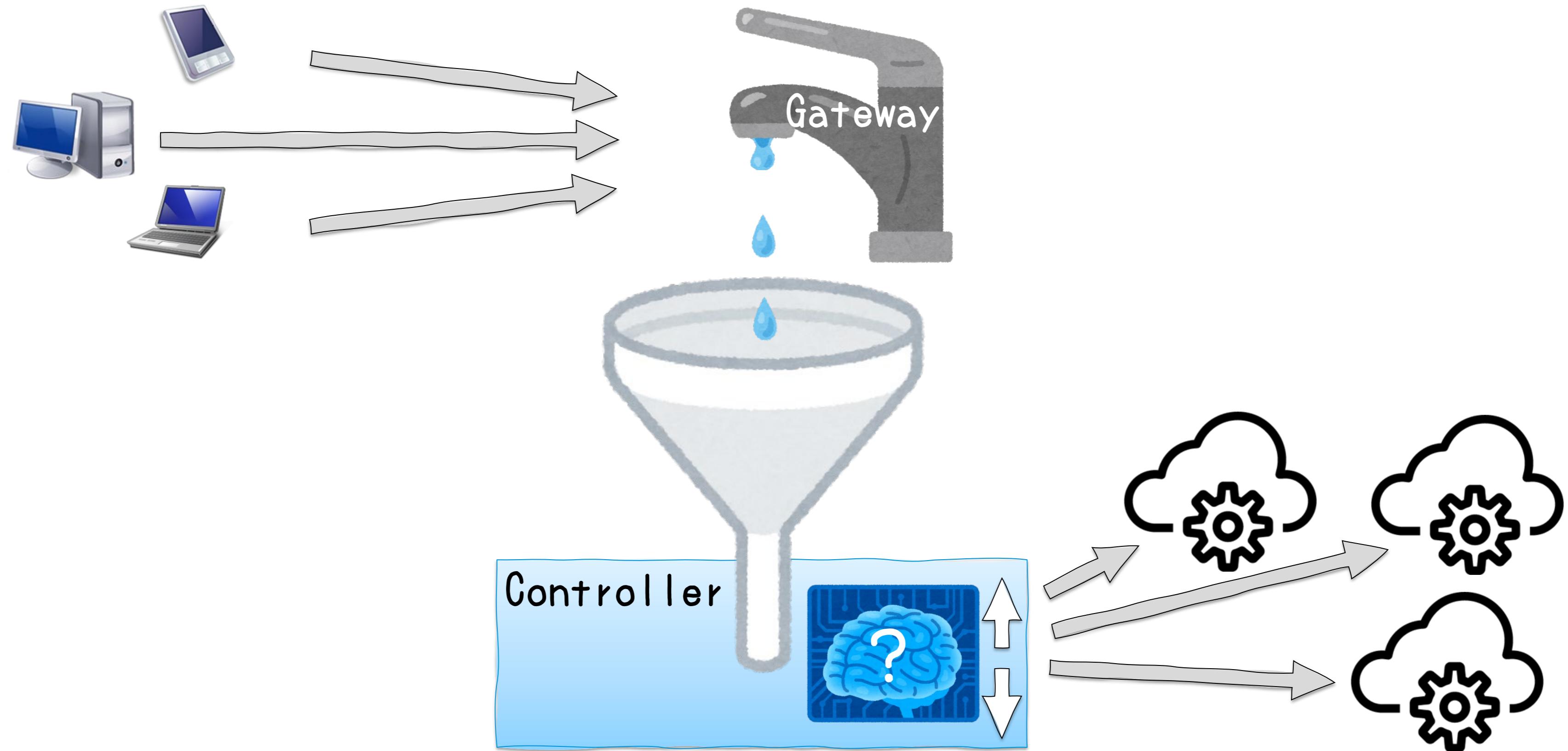


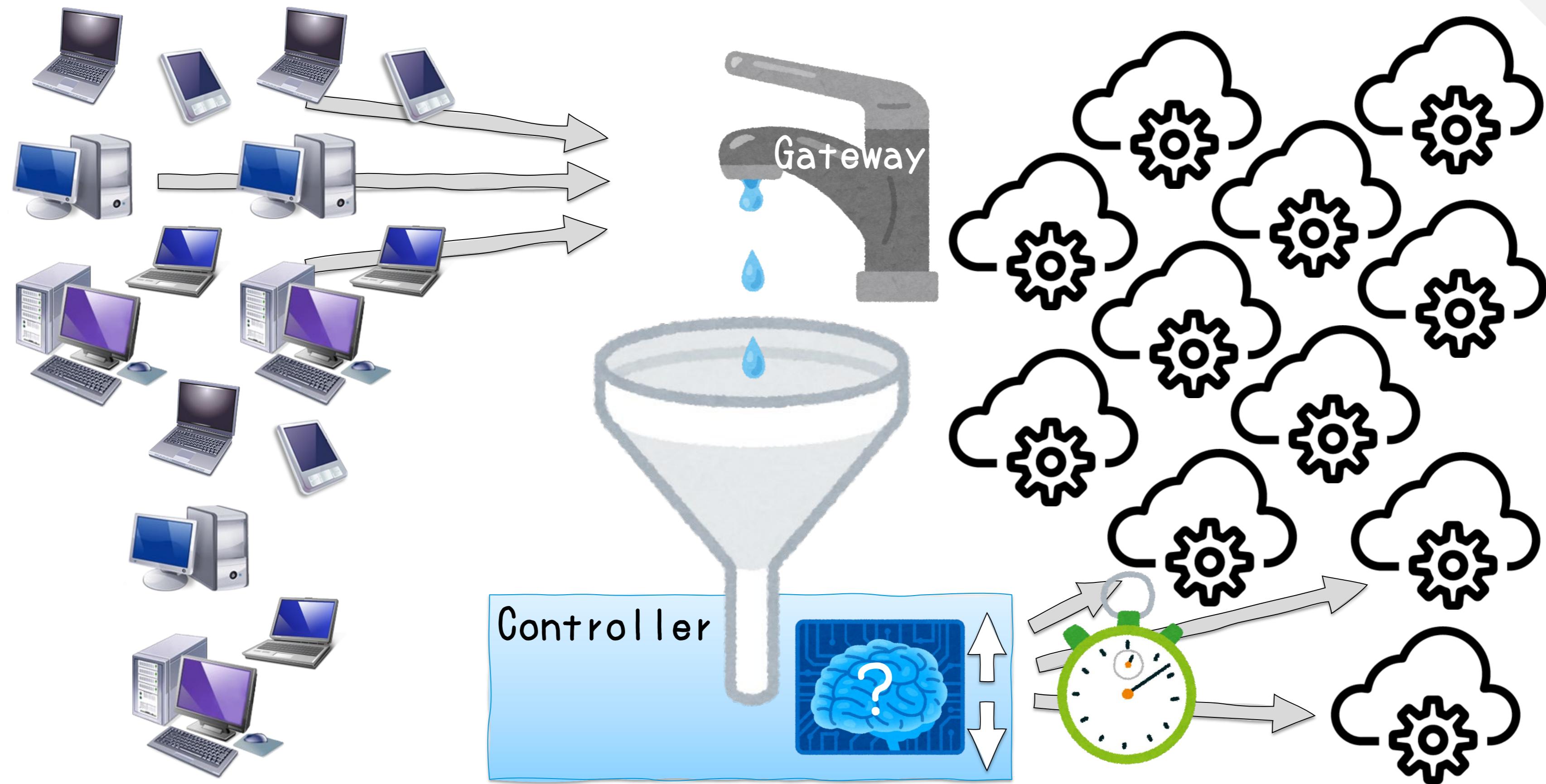
デスクトップ製品は誰にライセンスされているか？



デスクトップ製品は誰にライセンスされているか？



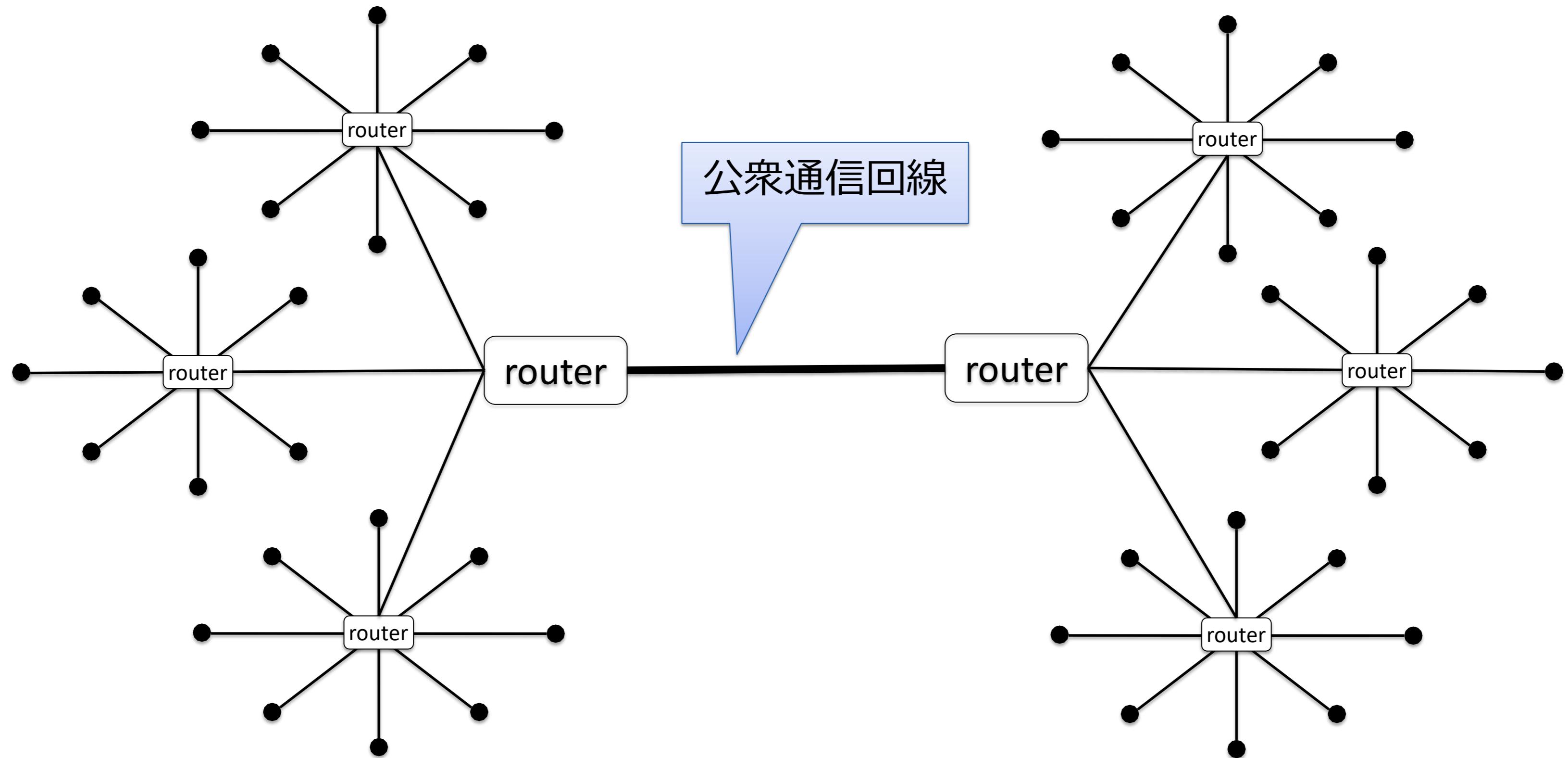




Elastic Computing ? : Automation API の場合

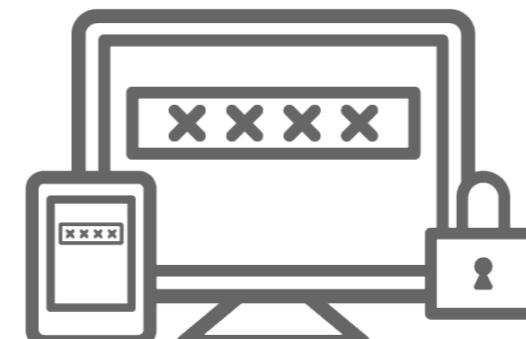


インターネット通信網の利用



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



Rate Limit 厳守のお願い

- 1分間あたりのエンドポイント呼び出し数制限
 - 429 レスポンス ステータス受信時に適切に対応する必要あり
 - 例) Automation API WorkItem 実行時のポーリング





AutoCAD Automation の理解

クラウド上でデスクトップ[®]製品アドイン実行環境を提供

- クラウド上の CAD コアエンジンをリモートで起動
- 起動したコアエンジンにアドインをロードして実行
- 成果物（デザインファイルや関連ファイル）を作成
- 成果物をダウンロードして利用

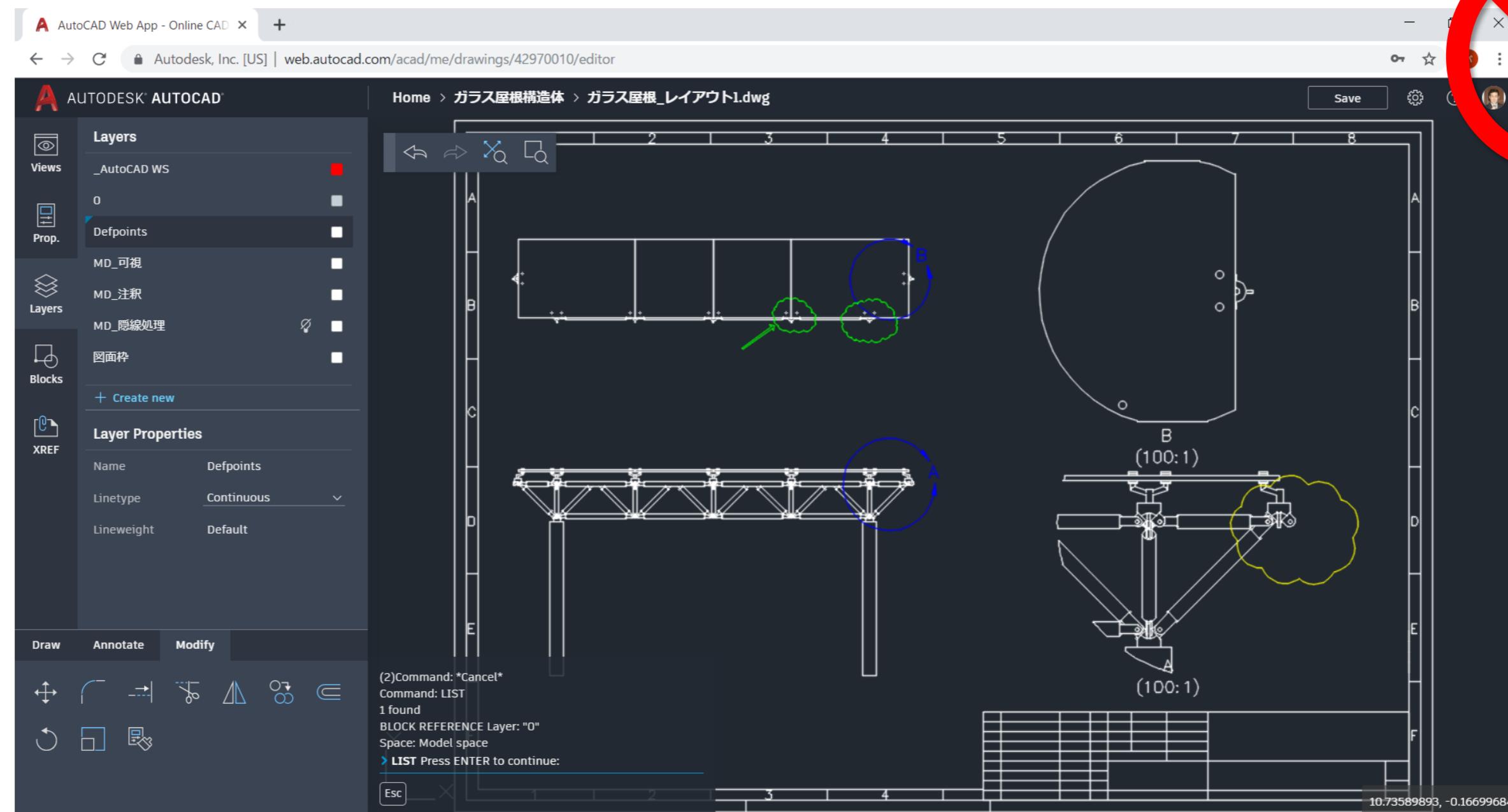
➤ Automation API

Automation API の誤った認識

- 製品を対話/対面操作するシンクライアント環境ではない
 - リモートデスクトップのような仕組みはありません
- Web ブラウザ上で CAD を実現する SDK ではない
 - フロント UI となる Web ページは HTML で実装が必要です
 - APS Viewer は Automation API には含まれません
- 処理時の利用ファイルはローカル PC との間で直接入出力不可
 - パブリック クラウド ストレージの利用が必須です
 - Automation API 仮想環境へダウンロード/からアップロードします

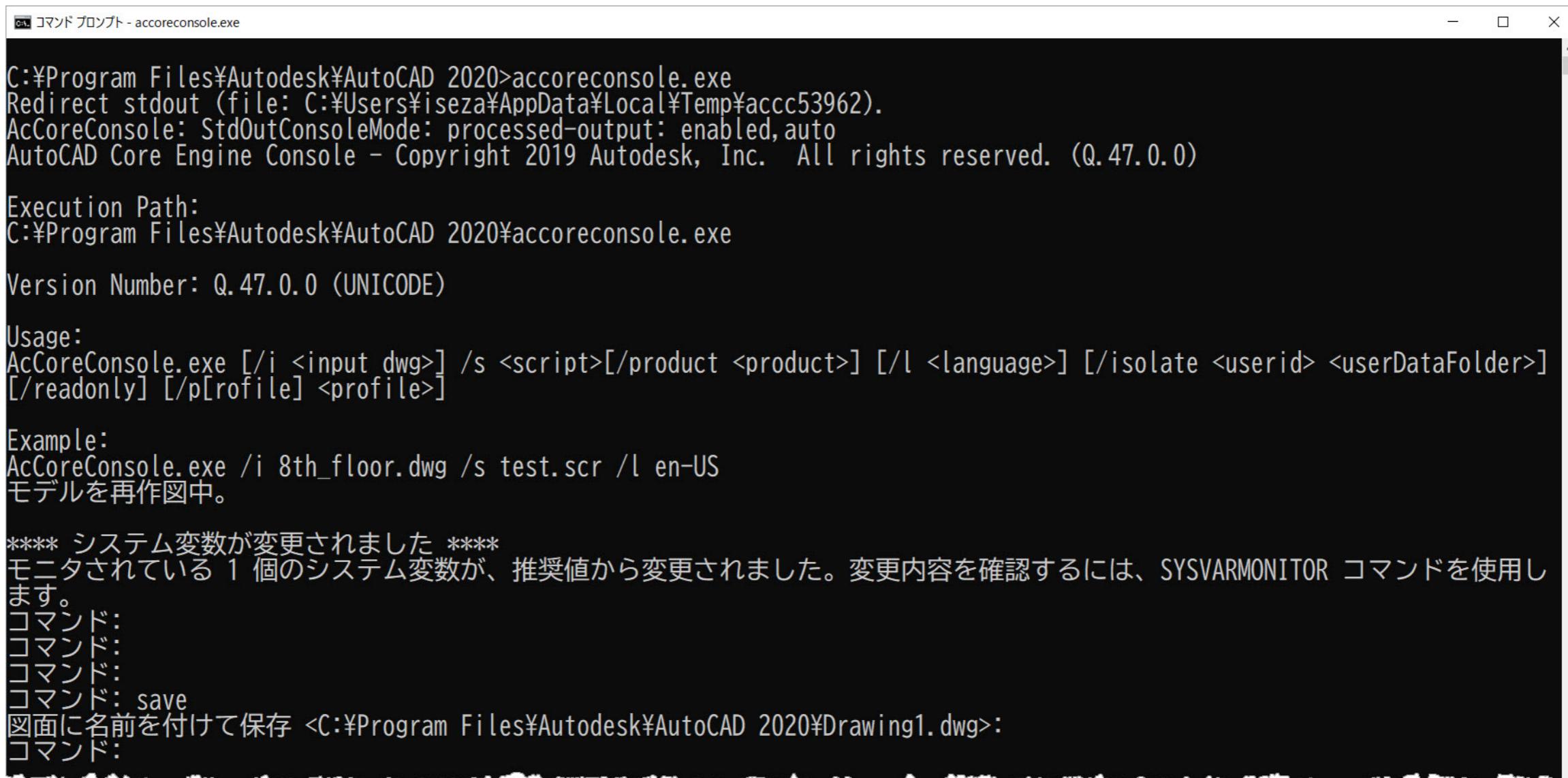
Automation API の間違った理解

- AutoCAD Web のアドイン開発環境ではありません



AutoCAD コアエンジン

- AutoCAD に同梱されているコマンド プロンプト版 AutoCAD
- **AcCoreConsole.exe** : 一切の GUI を省略



```

C:\Program Files\Autodesk\AutoCAD 2020>accoreconsole.exe
Redirect stdout (file: C:\Users\iseza\AppData\Local\Temp\accc53962).
AcCoreConsole: StdOutConsoleMode: processed-output: enabled,auto
AutoCAD Core Engine Console - Copyright 2019 Autodesk, Inc. All rights reserved. (Q.47.0.0)

Execution Path:
C:\Program Files\Autodesk\AutoCAD 2020\accoreconsole.exe

Version Number: Q.47.0.0 (UNICODE)

Usage:
AcCoreConsole.exe [/i <input_dwg>] /s <script>[/product <product>] [/l <language>] [/isolate <userid> <userDataFolder>]
[/readonly] [/p[rofile] <profile>]

Example:
AcCoreConsole.exe /i 8th_floor.dwg /s test.scr /l en-US
モデルを再作図中。

**** システム変数が変更されました ****
モニタされている 1 個のシステム変数が、推奨値から変更されました。変更内容を確認するには、SYSVARMONITOR コマンドを使用します。
コマンド:
コマンド:
コマンド:
コマンド: save
図面に名前を付けて保存 <C:\Program Files\Autodesk\AutoCAD 2020\Drawing1.dwg>:
コマンド:

```

Automation API を正しく理解しましょう

対話的な表示/編集機能はありません

- AutoCAD Web のようなものではありません

ビューア機能はありません

- 必要に応じて APS Viewer の利用を検討出来ます

エンドユーザ向けのサービスではありません

- 開発者向けのサービスです

サーバー モジュールではありません

- オンプレミス(プライベート)サーバー版はありません

Automation API 正しい理解のために

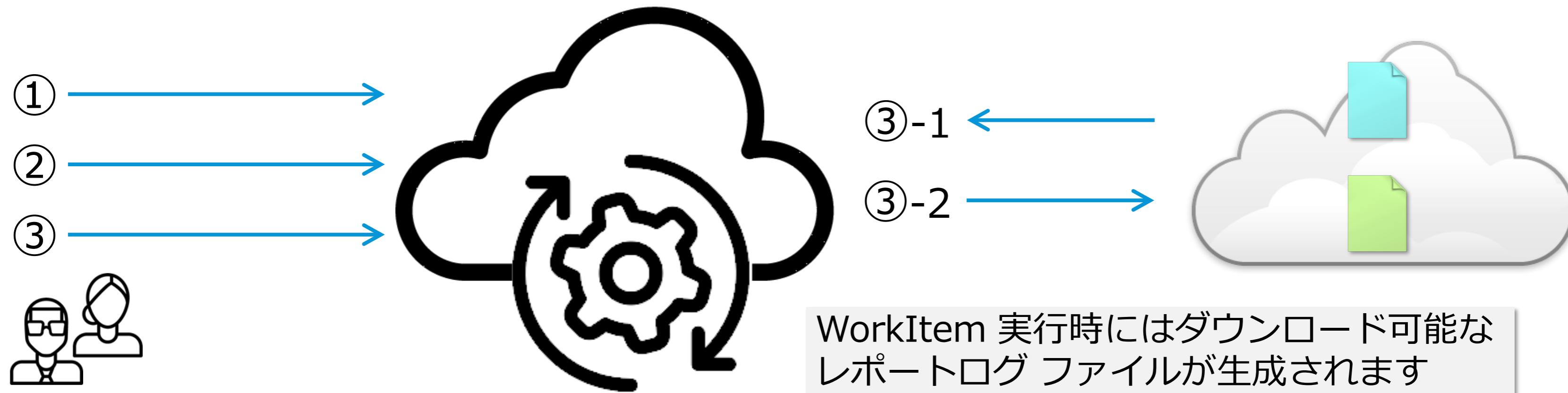
- 利用する用語 :
- AppBundle (アップバンドル/アプリバンドル)
- Activity (アクティビティ)
- WorkItem (ワークアイテム)
- Engine (エンジン、または、コア エンジン)

Automation API とは？

- **起動したコアエンジンにアドインをロード/実行**
 - アドイン+自動ロード定義を ZIP 圧縮したもの = AppBundle
 - AppBundle は WorkItem 実行前に登録が必要
- **成果物としてデザインファイルや関連ファイルの作成してダウンロードして利用**
 - Activity で入出力ファイル/パラメータを宣言
 - Activity は WorkItem 実行前に登録が必要
 - WorkItem (ファイル/パラメータ渡しを実行)
- **クラウド上の CAD コアエンジンをリモートで起動**
 - WorkItem (実際の処理実行)

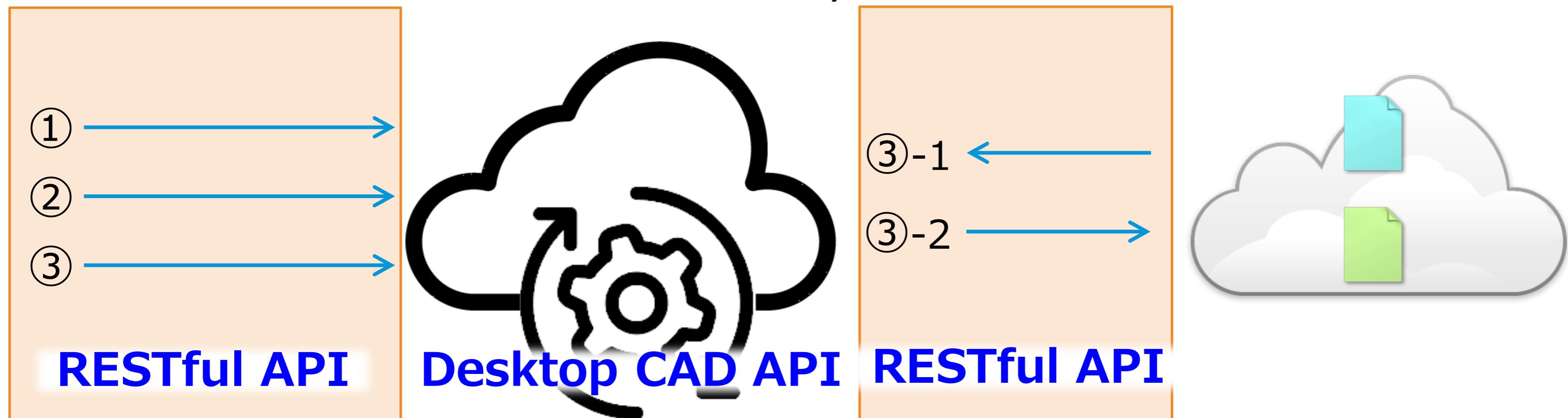
Automation API とは

- クラウドのコアエンジン上でアドインを実行する仕組みを提供
- ① AppBundle(アドイン本体+自動ロード定義) のアップロード&登録
 - ② Activity を登録 (AppBundle が使用する入出力ファイルを指定)
 - ③ WorkItem を指定、実行 (Activity と各種パラメータを指定)



Automation API とは

- クラウドのコアエンジン上でアドインを実行する仕組みを提供
 - ① AppBundle(アドイン本体+自動ロード定義) のアップロード&登録
 - ② Activity を登録 (AppBundle が使用する入出力ファイルを指定)
 - ③ WorkItem を指定、実行 (Activity と各種パラメータを指定)



Automation API で利用可能なコアエンジン バージョン

- デスクトップ製品と同期したコアエンジン バージョン
 - AppBundle はエンジン バージョンに合わせた作成が必須
 - コアエンジン ID の形式はコアエンジン毎に異なる
 - 最新 + 過去 3 バージョンがサポート対象

“Autodesk.AutoCAD+24_2” ⇒ AutoCAD 2023]

.NET Framework 4.8

“Autodesk.AutoCAD+24_3” ⇒ AutoCAD 2024]

.NET Framework 4.8

“Autodesk.AutoCAD+25_0” ⇒ AutoCAD 2025]

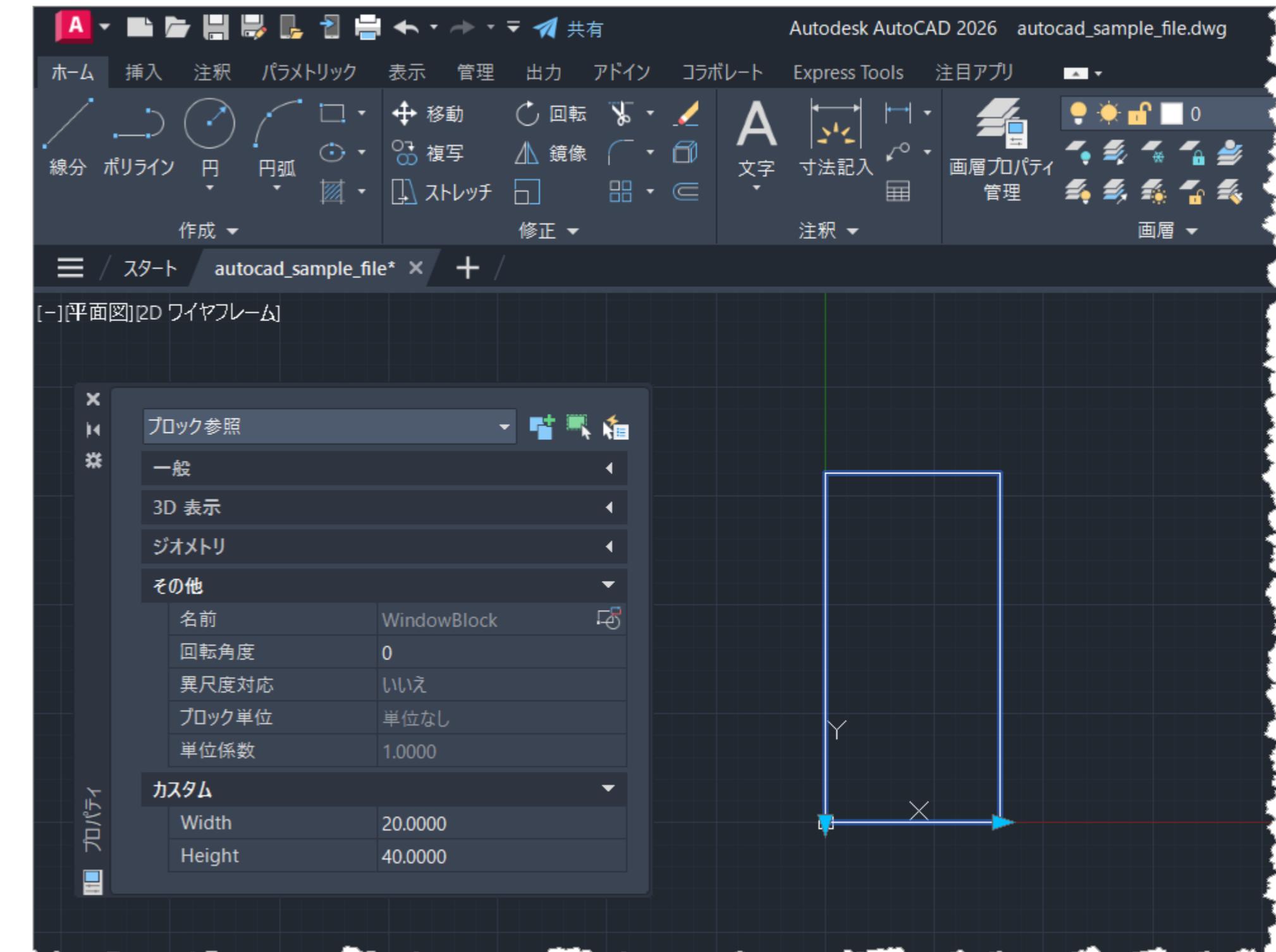
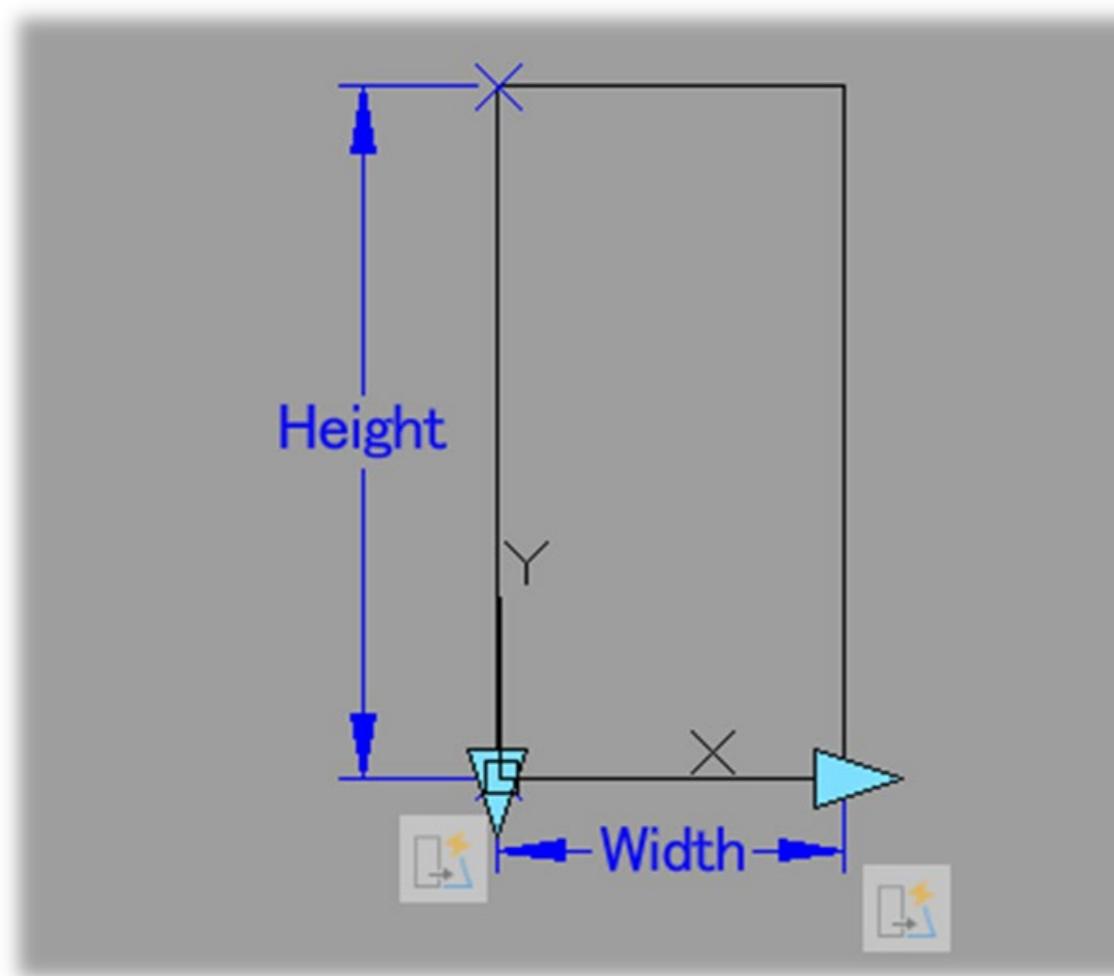
.NET 8 (.NET Core)

“Autodesk.AutoCAD+25_1” ⇒ AutoCAD 2026]

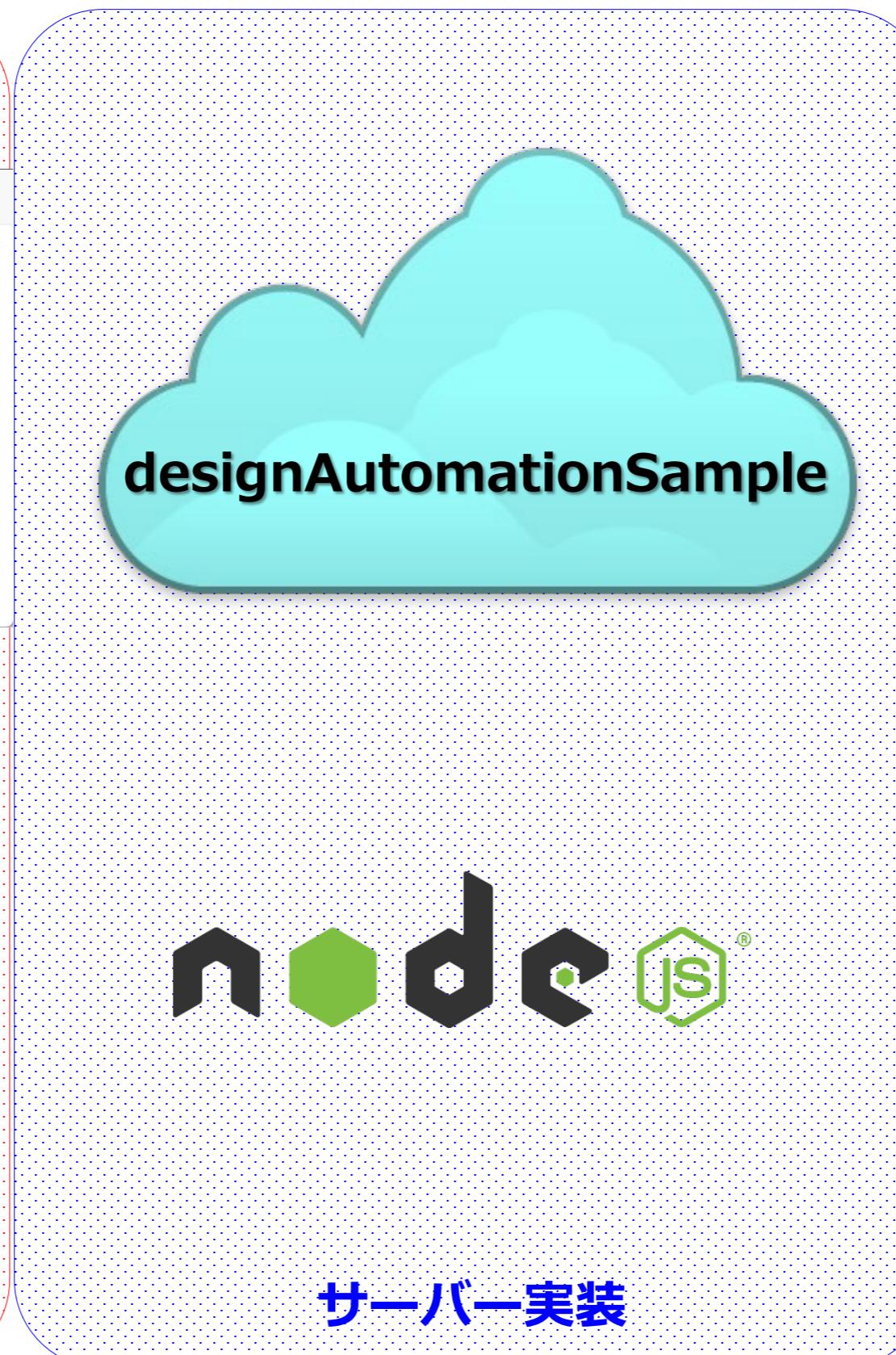
- AppBundle と Activity の登録時に指定

処理内容：ダイナミックブロック パラメータの変更

- 直線状パラメータ
 - Height
 - Width



Design Automation Tutorial ワークフロー





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



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)

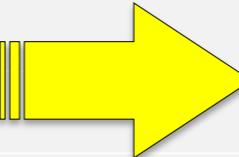
My applications からのアプリ登録

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。

 **AUTODESK**
Platform Services

Search

[Developer hubs](#) 

Solutions ▾ Getting Started Documentation ▾ Success Stories Community ▾ Support ▾ Pricing App Store ▾

[< Back to Applications](#)

designAutomationSample

App settings Token usage API usage

Client Credentials

The Client ID and Client Secret are used to obtain access tokens, which you must use to authenticate API calls.

Client ID
 

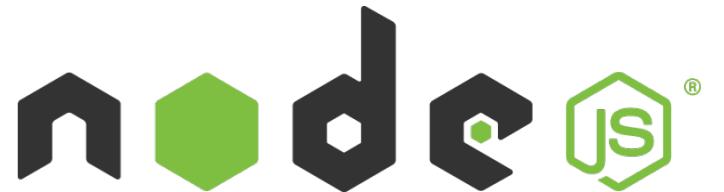
Client Secret
.....  

[Regenerate](#)

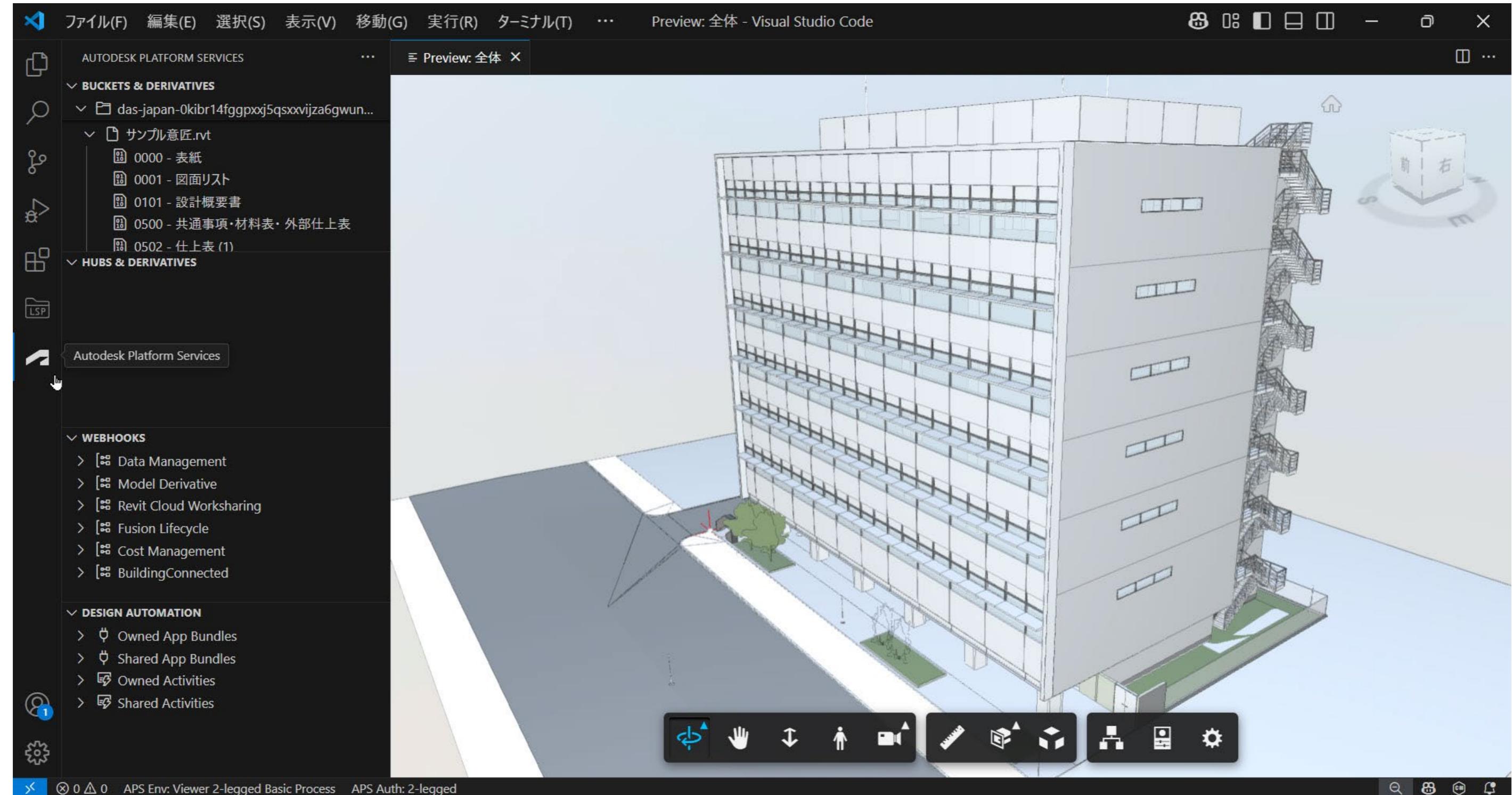
今回使用する開発環境

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**



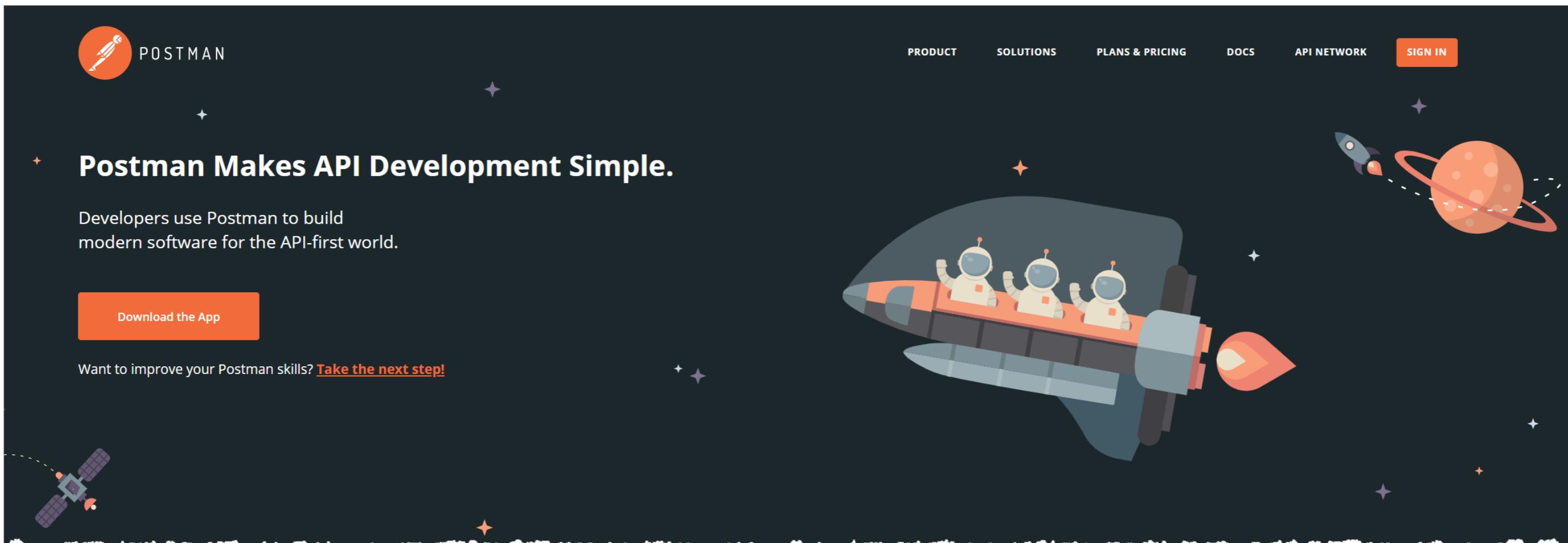
VS Code - APS エクステンション



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



The screenshot shows the official Postman website homepage. The header features the Postman logo (an orange circle with a white pen icon), navigation links for PRODUCT, SOLUTIONS, PLANS & PRICING, DOCS, API NETWORK, and a red SIGN IN button. The main headline reads "Postman Makes API Development Simple." Below it, a sub-headline states "Developers use Postman to build modern software for the API-first world." A prominent orange "Download the App" button is visible. To the right, there's a whimsical illustration of three astronauts in a rocket ship launching from Earth towards a planet with a ring, with a satellite in the background. At the bottom left, there's a link to "Take the next step!"



サーバーの作成

サーバーの作成 – Create Server

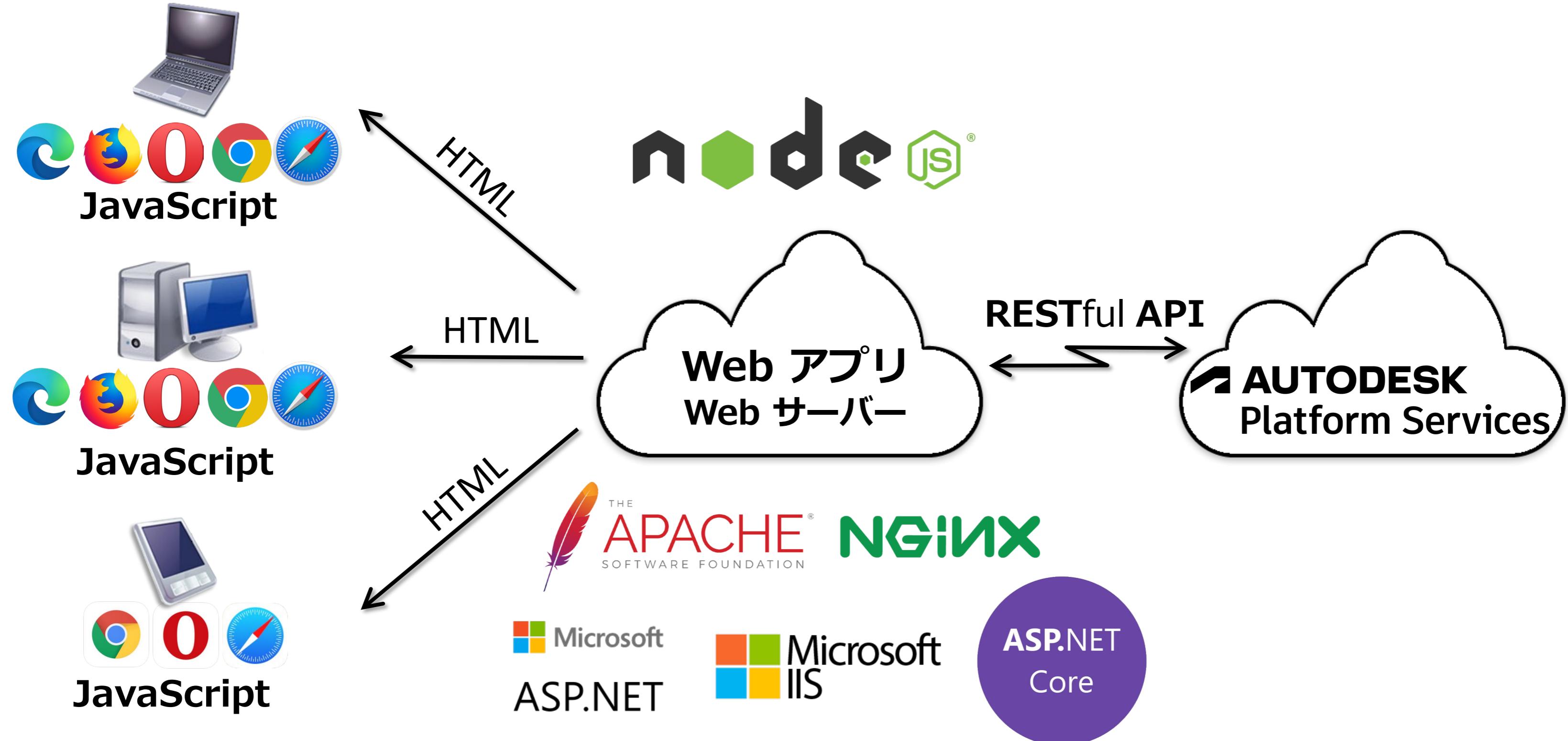
The screenshot shows the Autodesk Platform Services website with a dark header. The header includes the Autodesk logo, a search bar, and a 'Sign in' button. Below the header is a navigation bar with links: Solutions, Getting Started, Documentation, Success Stories, Community, Support, Pricing, and App Store.

The main content area has a breadcrumb navigation: Home > Tutorials > Design Automation > Create Server. The main title is 'Create Server'. A text block explains that Client ID & Secret should be protected and kept confidential, and it demonstrates how to prepare a local development server. It also suggests reviewing the Environment Setup section for required software.

On the left sidebar, under 'Tutorials', there are links: Getting Started, Environment Setup, Simple Viewer, Hubs Browser, and Dashboard. Under 'Design Automation', there are links: Create Server, Basic UI, Create Plugin, Define Activity, and Execute Workitem. The 'Create Server' link is highlighted in blue.

In the bottom right corner of the main content area, there are three options: 'Setup Project', 'Application Config', and 'Server Setup'. The 'Setup Project' option is currently selected, indicated by a red dashed border around the 'Node.js & VSCode' link. Other options include '.NET & VSCode' and '.NET & VS2022'.

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）の一部：
 - **autodesk.forge.designautomation**
 - cookie-session
 - express
 - express-formidable
 - **forge-apis**
 - socket.io

Forge Design Automation SDK - v3用

Forge SDK

コマンドプロンプト/VS Code 上の操作手順

1. プロジェクト フォルダの作成と Node.js の初期化
2. Node.js パッケージをインストール、**package.json** ファイルを確認
3. VS Codeでプロジェクトフォルダを開き **routes**、**wwwroot** フォルダを作成
4. Node.js 構成 を追加、**launch.json** ファイルに Client ID/Secret を指定
5. ルートフォルダに **start.js** ファイルの作成と実装
6. ルートフォルダに **server.js** ファイルの作成と実装
7. ルートフォルダに **socket.io.js** ファイルの作成と実装
8. ルートフォルダに **config.js** ファイルの作成とサーバー実装記述の追加
9. routes/common/**oauth.js** ファイルの作成と実装

Create Server | Autodesk Platform Services

get-started.aps.autodesk.com/tutorials/design-automation/create-server

Solutions Getting Started Documentation Success Stories Community Support Pricing App Store

Getting Started Environment Setup Tutorials Simple Viewer Hubs Browser Dashboard Design Automation Create Server

Home > Tutorials > Design Automation > Create Server

Create Server

Your Client ID & Secret should be protected and kept confidential as all your files will be bound to your account. For a web application, keep it on your server. This section demonstrate how to prepare create a local development server.

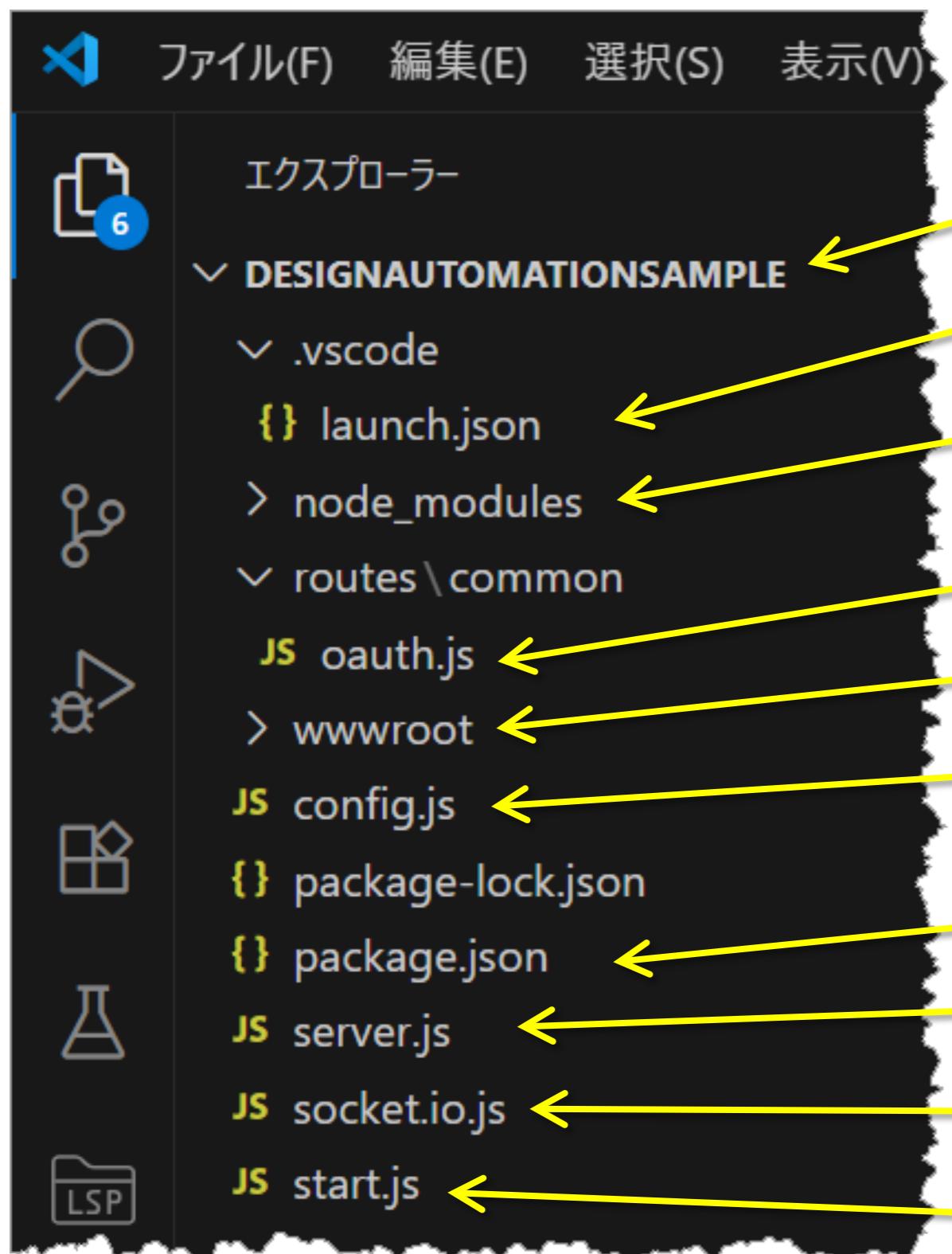
Please review Environment Setup section for required software.

Setup 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:

ここまでプロジェクト構成



サーバー実装

プロジェクト フォルダ

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

2-legged 認証実装

クライアント構成（未実装）

スコープ等認証情報情報

Node.js パッケージ依存関係

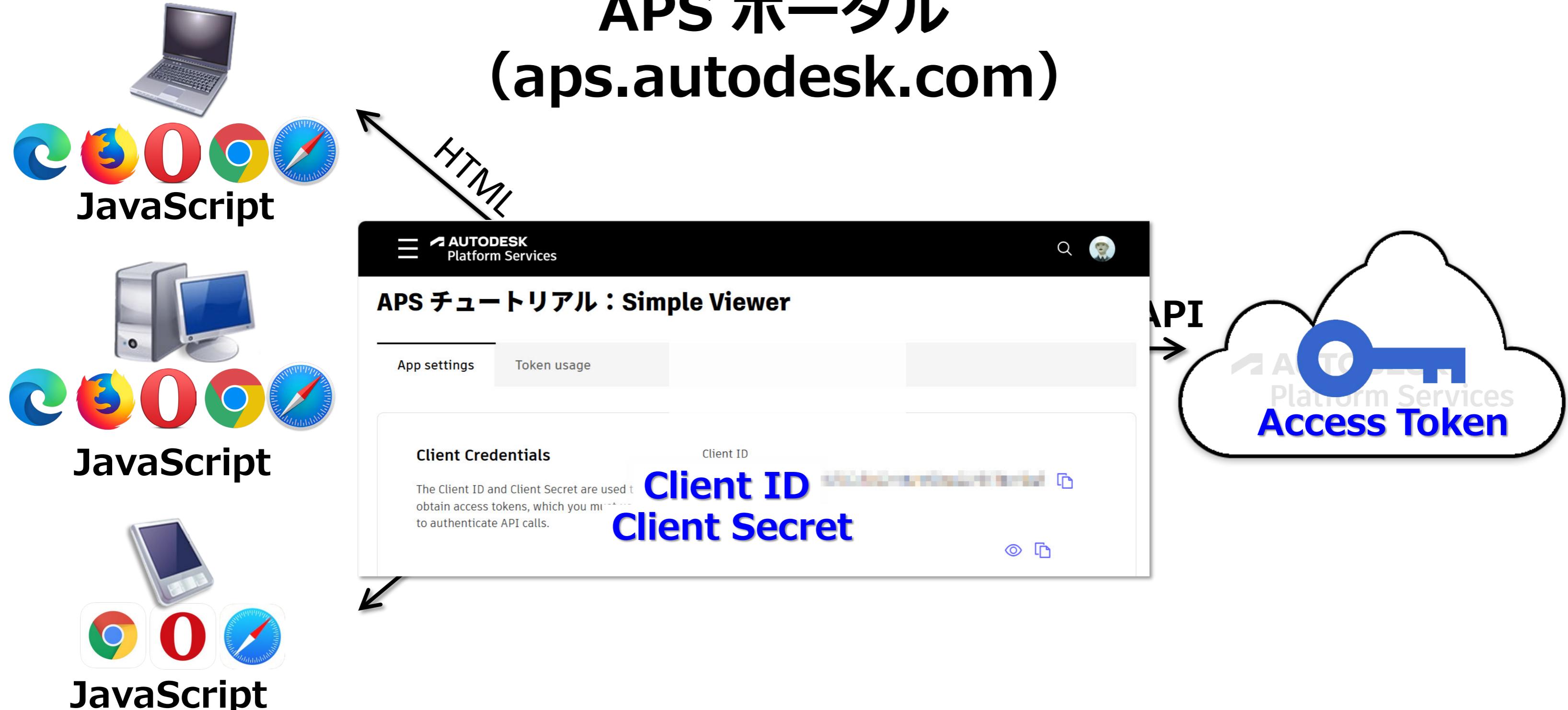
セッション クッキー等情報

ソケット通信接続/切断実装

Node.js サーバーの起動構成

デベロッパキーとアクセストークン

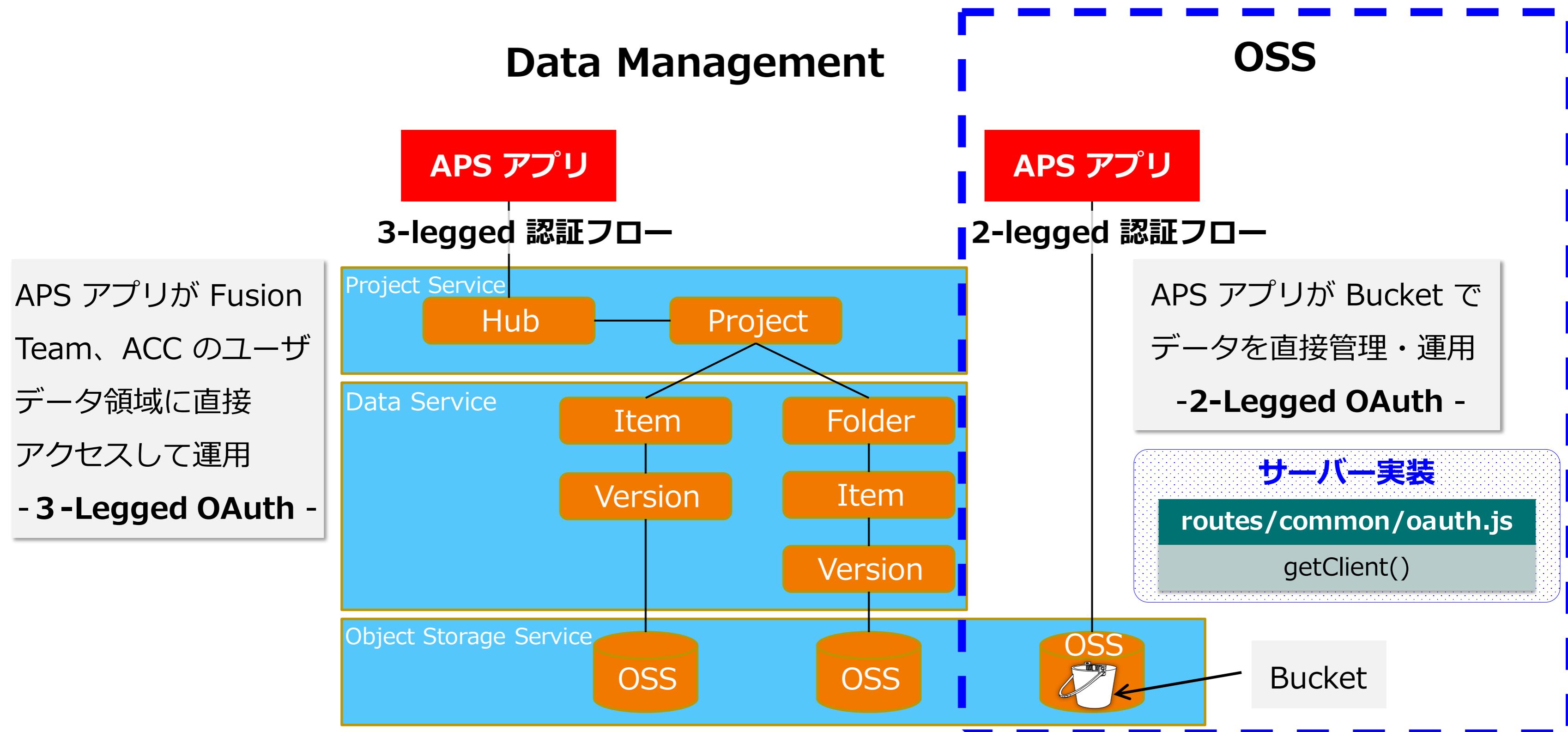
APS ポータル (aps.autodesk.com)



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

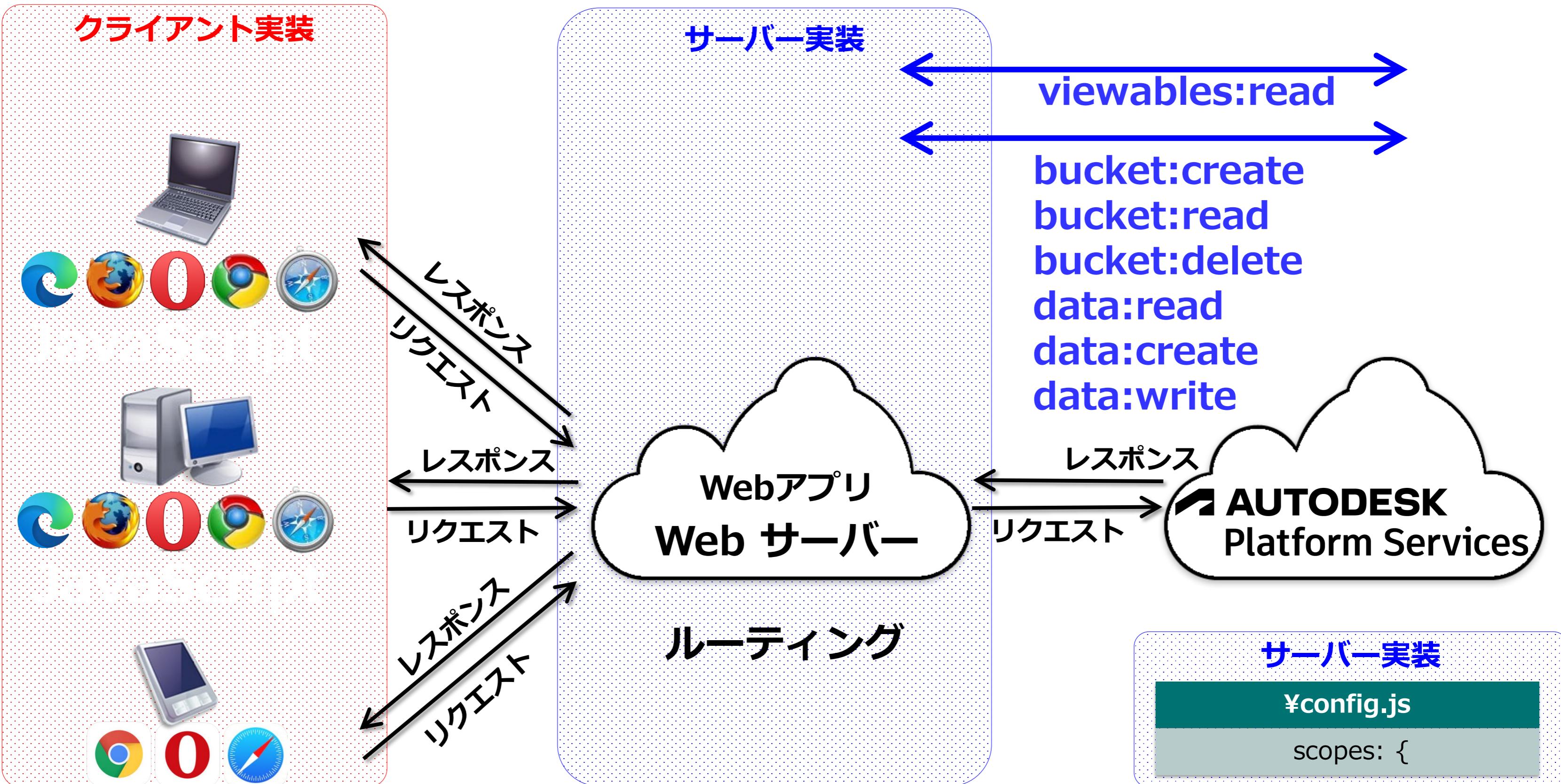
Design Automation Tutorial の 認証フロー



Design Automation Tutorial が使用する Scope

Scope 文字列	意味
user-profile:read	プロファイル (Autodesk ID) の表示
user:read	プロファイル (Autodesk ID) の読み取り
user:write	プロファイル (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:delete	Bucket の削除
bucket:update	Bucket の更新
code:all	コードの生成または実行 (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

スコープによるトークンを使い分け





基本 UI

Basic UI - 基本 UI

The screenshot shows a dark-themed web page for Autodesk Platform Services. At the top left is the Autodesk logo and "Platform Services". A search bar is at the top center, and a "Sign in" button is at the top right. The top navigation bar includes links for "Solutions", "Getting Started", "Documentation", "Success Stories", "Community", "Support", "Pricing", and "App Store". On the left sidebar, under "Tutorials", there are links for "Getting Started", "Environment Setup", "Simple Viewer", "Hubs Browser", and "Dashboard". Under "Design Automation", there are links for "Create Server", "Basic UI" (which is highlighted in blue), "Create Plugin", "Define Activity", and "Execute Workitem". The main content area has a breadcrumb navigation: Home > Tutorials > Design Automation > Basic UI. To the right of the breadcrumb is a file path: ApsDesignAutomation.js. The main title is "Basic UI". Below the title, a paragraph explains that the interface is based on vanilla HTML5 & JavaScript, mentioning differences in the Websocket implementation using socket.io or SignalR. It instructs to start with an index.html file in the wwwroot folder. Below this text are three buttons for "Node.js & VSCode", ".NET & VSCode", and ".NET & VS2022". At the bottom of the page, a dark bar displays the URL "wwwroot/index.html" and the opening tag "<!DOCTYPE html>".

AUTODESK
Platform Services

Solutions Getting Started Documentation Success Stories Community Support Pricing App Store

Getting Started Environment Setup

Tutorials

Simple Viewer
Hubs Browser
Dashboard

Design Automation

Create Server
Basic UI
Create Plugin
Define Activity
Execute Workitem

Home > Tutorials > Design Automation > Basic UI

ApsDesignAutomation.js

Basic UI

The interface is based on vanilla HTML5 & JavaScript. It essentially the same for any server-side, but there are a few differences: the Websocket implementation uses socket.io (Node.js) or SignalR (.NET 6). Let's start with the UI (HTML and JavaScript) files. Under the wwwroot folder, create an index.html file with the following content:

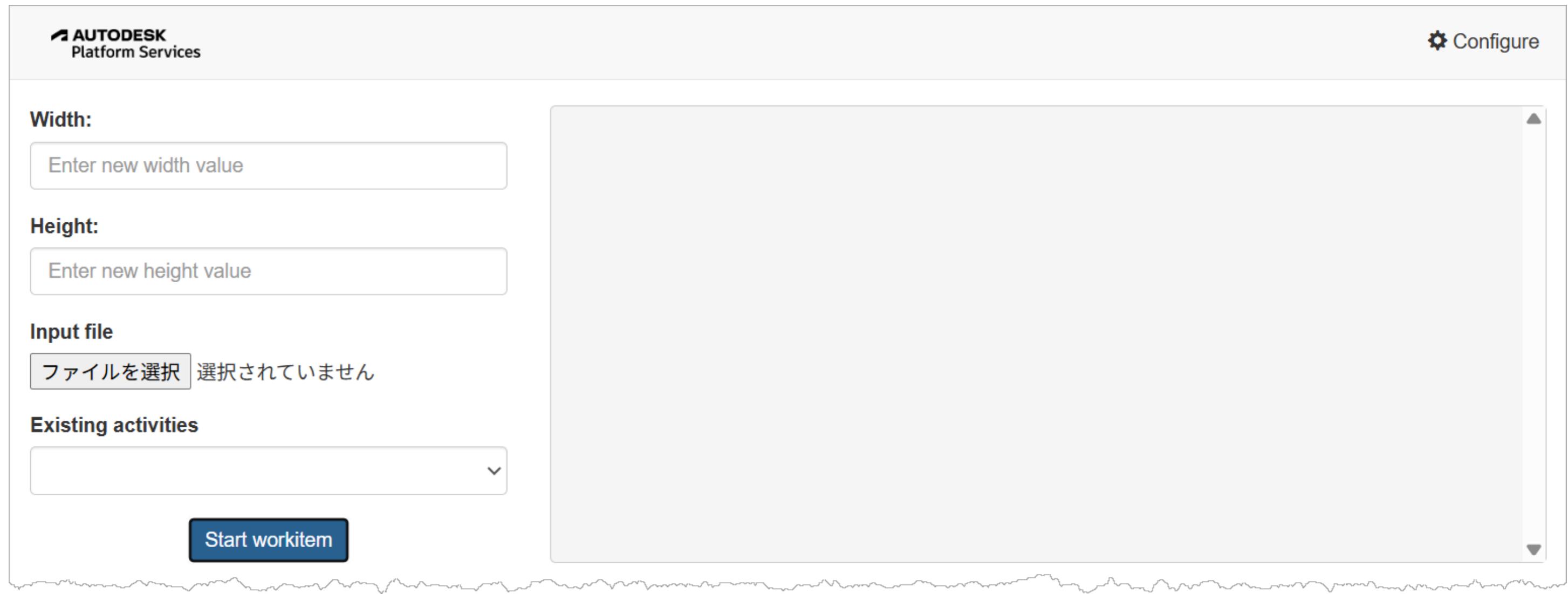
[Node.js & VSCode](#) [.NET & VSCode](#) [.NET & VS2022](#)

```
wwwroot/index.html
```

```
<!DOCTYPE html>
```

VS Code 上の操作手順

1. **wwwroot** フォルダに **index.html** ファイルを作成して実装
2. **js** サブフォルダを作成、**ApsDesignAutomation.js** ファイルを作成して実装
3. デバッグ実行 >> **localhost:8080**



Basic UI | Autodesk Platform Services X +

get-started.aps.autodesk.com/tutorials/design-automation/basic-app-ui

AUTODESK Platform Services Sign in

Solutions Getting Started Documentation Success Stories Community Support Pricing App Store

Getting Started Environment Setup

Tutorials

- Simple Viewer
- Hubs Browser
- Dashboard

Design Automation

- Create Server
- Basic UI**
- Create Plugin
- Define Activity
- Execute Workitem
- ACC Administrator
- ACC Issues (beta)

Home > Tutorials > Design Automation > Basic UI

ApsDesignAutomation.js

Basic UI

The interface is based on vanilla HTML5 & JavaScript. It essentially the same for any server-side, but there are a few differences: the Websocket implementation uses socket.io (Node.js) or SignalR (.NET 6). Let's start with the UI (HTML and JavaScript) files. Under the wwwroot folder, create an index.html file with the following content:

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

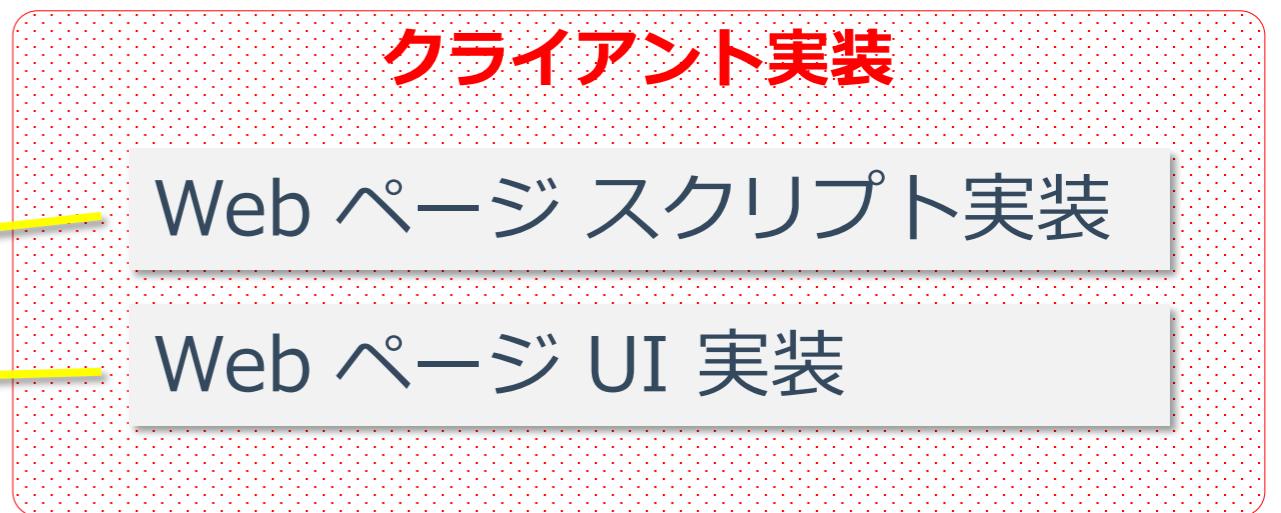
```
wwwroot/index.html
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>APS - Design Automation</title>
    <meta charset="utf-8" />
    <link rel="shortcut icon" href="https://cdn.autodesk.io/favicon.ico" />
```

ここまでプロジェクト構成

The screenshot shows the file explorer in VS Code with the following project structure:

- DESIGNAUTOMATIONSAMPLE
 - .vscode
 - launch.json
 - node_modules
 - routes\common
 - oauth.js
 - wwwroot
 - js
 - ApsDesignAutomation.js
 - index.html
- config.js
- package-lock.json
- package.json
- server.js
- socket.io.js
- start.js





プラグインの作成

プラグインの作成 – Create Plugin

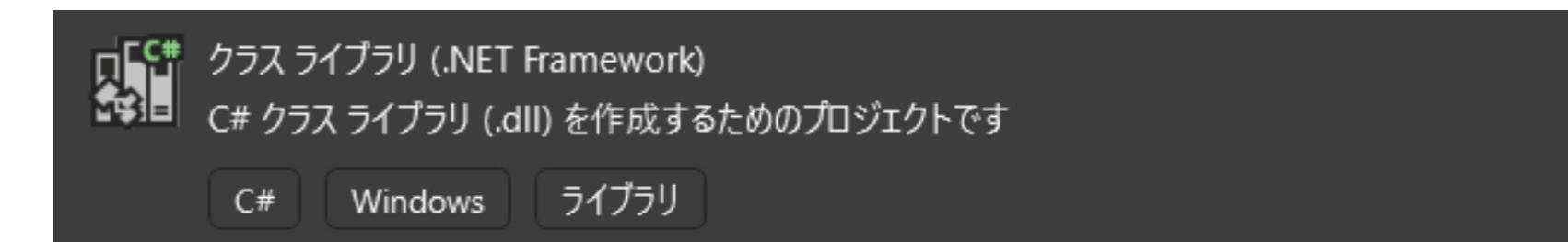
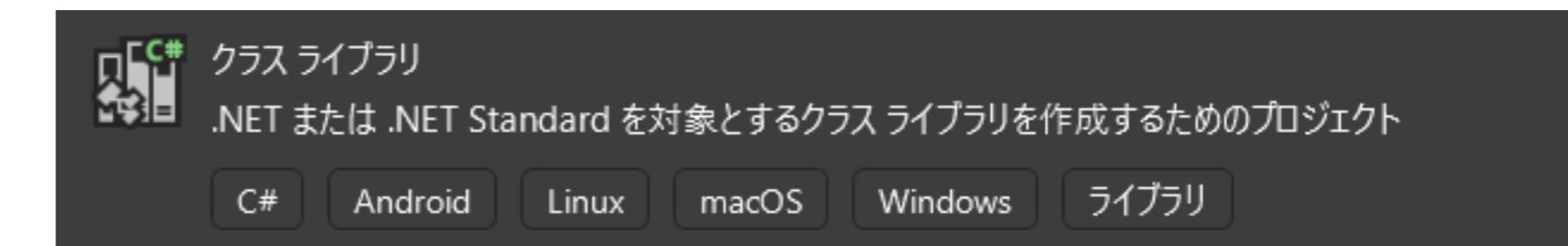
The screenshot shows the Autodesk Platform Services website with a dark header bar. The header includes the Autodesk logo, a search bar, and a 'Sign in' button. Below the header is a navigation menu with links: Solutions, Getting Started, Documentation, Success Stories, Community, Support, Pricing, and App Store.

The main content area has a breadcrumb navigation: Home > Tutorials > Design Automation > Create Plugin. The title 'Create Plugin' is prominently displayed. To the right of the title are three links: 'Prerequisites', 'Additional prerequisites', and 'Choose the engine'. A sidebar on the left under 'Design Automation' lists several options: Create Server, Basic UI, Create Plugin (which is highlighted in blue), Define Activity, and Execute Workitem.

The 'Choose the engine' section contains five buttons: AutoCAD Plugin (which is highlighted with a red dashed border), Inventor Plugin, Revit Plugin, and 3ds Max Plugin. Below this section is a descriptive text: 'This step will help you create a basic AutoCAD plugin for Design Automation. For more information, please visit [My First AutoCAD Plugin](#) tutorial.'

UpdateDWGParam アドインの作成

- Visual Studio 2022 Professional を使用可
 - AutoCAD アドイン開発で主流 (AutoCAD .NET API)
 - 使用する Microsoft Visual Studio のエディション(.NET)
 - クラスライブラリプロジェクト
 - エンジンバージョンによって .NET/.NET Framework 差に注意
 - AutoCAD .NET Wizard でプロジェクト作成も利用可
 - NuGet パッケージの導入
 - Newtonsoft.json
 - AutoCAD.NET.Model
 - AutoCAD.NET.Core



AutoCAD アドインの Automation API 化

- 標準的な AutoCAD .NET API アドインのアセンブリ参照
 - **AcCoreMgd.dll**、**AcDbMgd.dll**、**AcMgd.dll**
- ユーザ インタフェース表示コードの削除
 - ダイアログ、パレット、ツールバー
 - ユーザ入力値はパラメータとして渡す必要
 - パラメータ >> **JSON**
- Automation の標準的なアドインのアセンブリ参照
 - **AcCoreMgd.dll**、**AcDbMgd.dll**、**Newtonsoft.Json.dll**

Web ページからアドインへのパラメータの受け渡し

- **JSON (ジェイソン) : JavaScript Object Notation**
 - 軽量で単純なデータ交換フォーマット

- {} 括弧で囲んでデータを表現

```
{
  'width': 999,
  'height': 555
}
```



- Web 環境でデータ交換に多用
- APS でも JSON を多用
 - 入力パラメータを渡す
 - 要 Activity 記述

AUTODESK Platform Services

Width:
Enter new width value

Height:
Enter new height value

Input file
ファイルを選択 選択されていません

Existing activities
UpdateDWGParamActivity+dev

Start workitem

コアエンジン用アドイン？

- GUI 表示をしない実装が必須
- サポートされる API

- ObjectARX

- .NET API

- AutoLISP

- .NET API の場合の参照アセンブリ

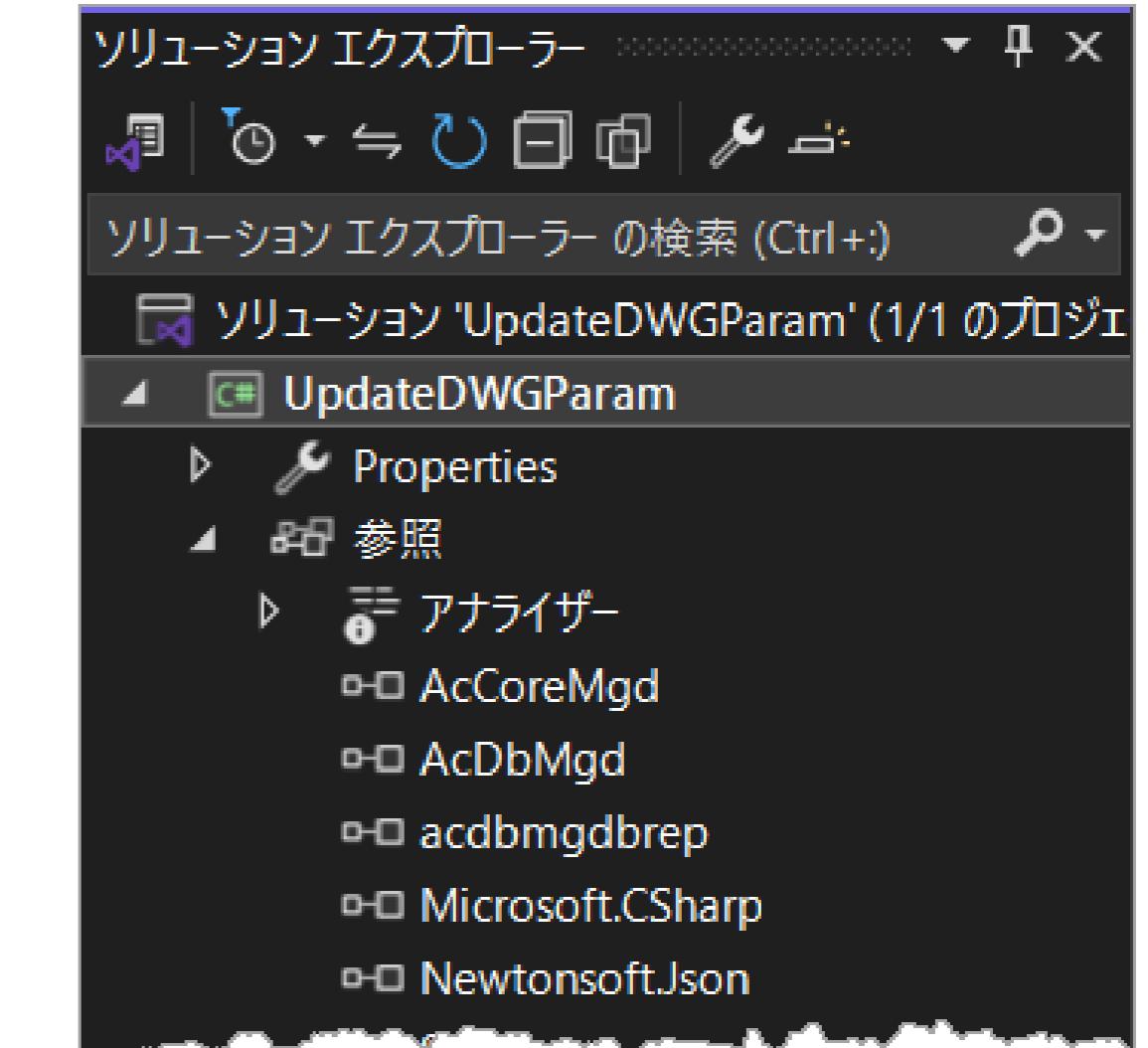
- AcCoreMgd.dll

- AcDbMgd.dll

推奨

必須
AcMgd.dll 参照不可

**NuGet から入手可能
(JSON ファイル操作で利用)**



コアエンジンへのアドイン ロード

- 有効なアドイン タイプ
 - AutoLISP アプリケーション
 - .NET アプリケーション
 - CRX アプリケーション

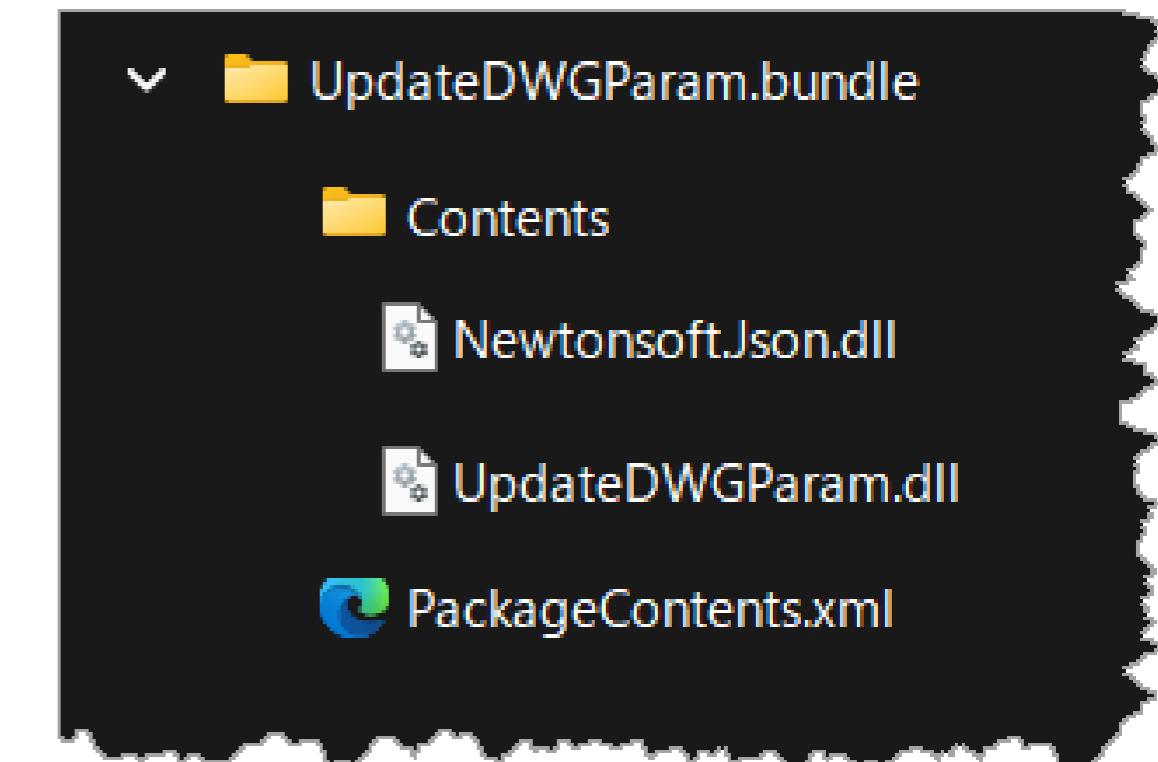


+

- アドインをロードさせる 自動ローダー 定義
 - PackageContents.xml とコンテンツを含むフォルダ構成
- ↓
- ZIP 圧縮したもの (パッケージバンドル)
> AppBundle

VS Professional 上の操作手順

1. クラス ライブラリ プロジェクトの作成
2. NuGet パッケージ マネージャーで必要パッケージをインストール
 - AutoCAD.NET.Model
 - AutoCAD.NET.Core
 - Newtonsoft.Json
3. Class1.cs の Commands.cs の名前変更と実装
4. アセンブリをビルド
5. パッケージ バンドルの作成
 - UTF-8 形式で ZIP 圧縮 (AppBundle)
 - UpdateDWGParam.zip を bundles フォルダへ移動



Create Plugin | Autodesk Platform Services +

get-started.aps.autodesk.com/tutorials/design-automation/prepare-plugin

AUTODESK Platform Services Sign in

Solutions Getting Started Documentation Success Stories Community Support Pricing App Store

Getting Started Environment Setup **Tutorials** Simple Viewer Hubs Browser Dashboard **Design Automation** Create Server Basic UI **Create Plugin** Define Activity Execute Workitem ACC Administrator ACC Issues (beta)

Home > Tutorials > Design Automation > Create Plugin

Create Plugin

Design Automation uses `.bundle` just like the Autodesk App Store, meaning you need to create a `PackageContents.xml` and a ZIP with the DLL (and other required files). For detailed information on how to create them, please visit Autodesk App Store Developer Center.

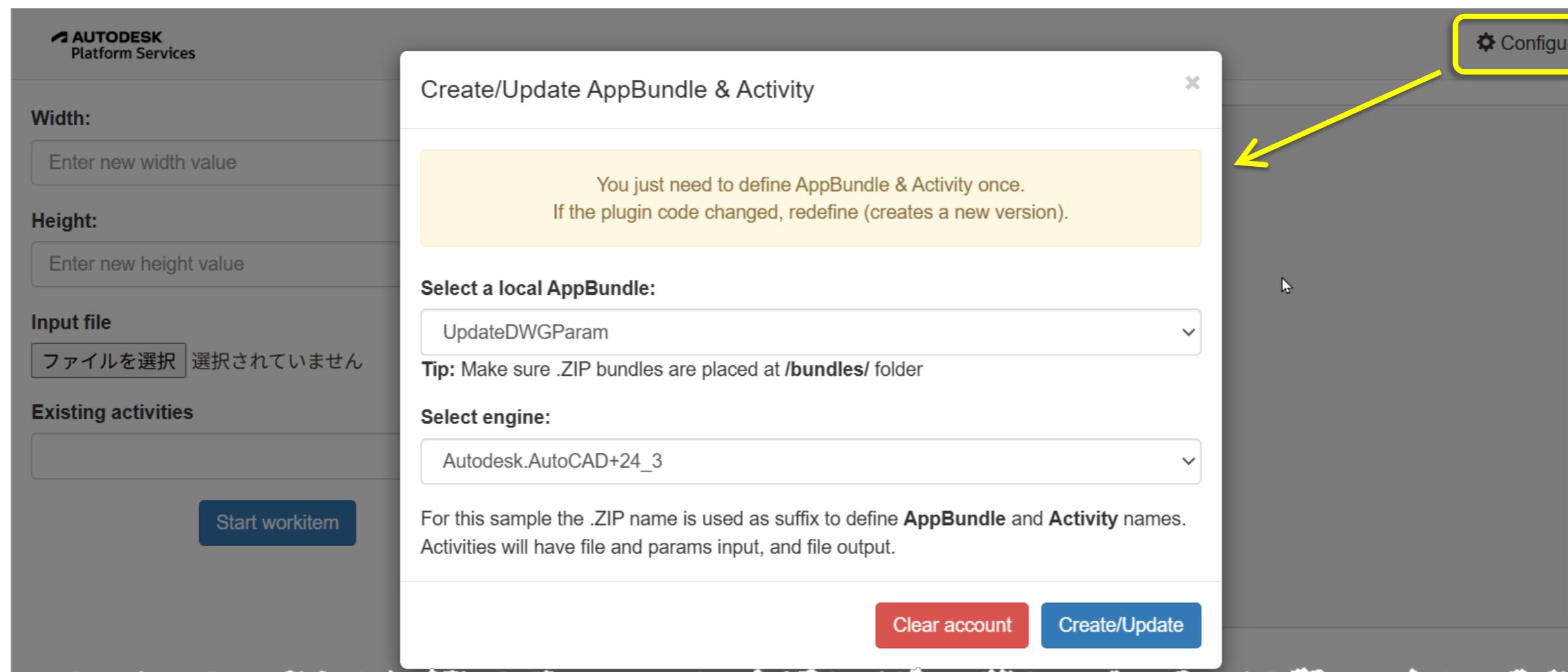
At this section we will create a basic plugin that update width and height parameter and save the resulting file. Also the supporting files (`PackageContents.xml`) and the folder structure to place them. Finally create a .ZIP file ready to upload to Design Automation.

In the root folder, create a `bundles` folder.

```
DESIGNAUTOMATIONSAMPLE
  .vscode
  launch.json
  bundles
  node_modules
  routes/common
    oauth.js
  wwwroot
```

VS Code 上の操作手順

1. ルートフォルダに **bundles** フォルダを作成
2. UpdateDWGParam アドイン パッケージを作成 または ダウンロード
3. **routes** フォルダ下に **DesignAutomation.js** ファイルを作成して実装
4. デバッグ実行 >> localhost:8080



Create Plugin | Autodesk Platform Services +

get-started.aps.autodesk.com/tutorials/design-automation/prepare-plugin

AUTODESK Platform Services Sign in

Solutions Getting Started Documentation Success Stories Community Support Pricing App Store

Getting Started Environment Setup **Tutorials** Simple Viewer Hubs Browser Dashboard **Design Automation** Create Server Basic UI **Create Plugin** Define Activity Execute Workitem ACC Administrator ACC Issues (beta)

Home > Tutorials > Design Automation > Create Plugin

Create Plugin

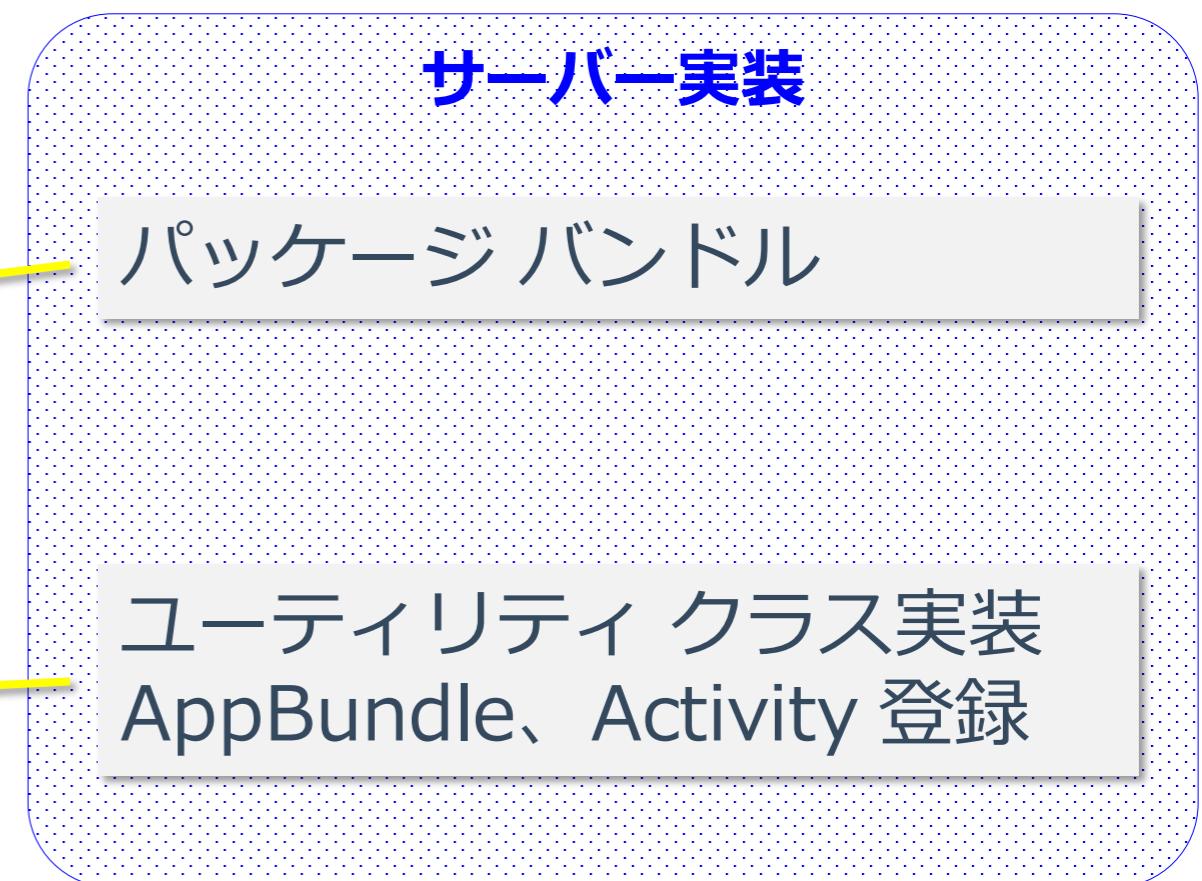
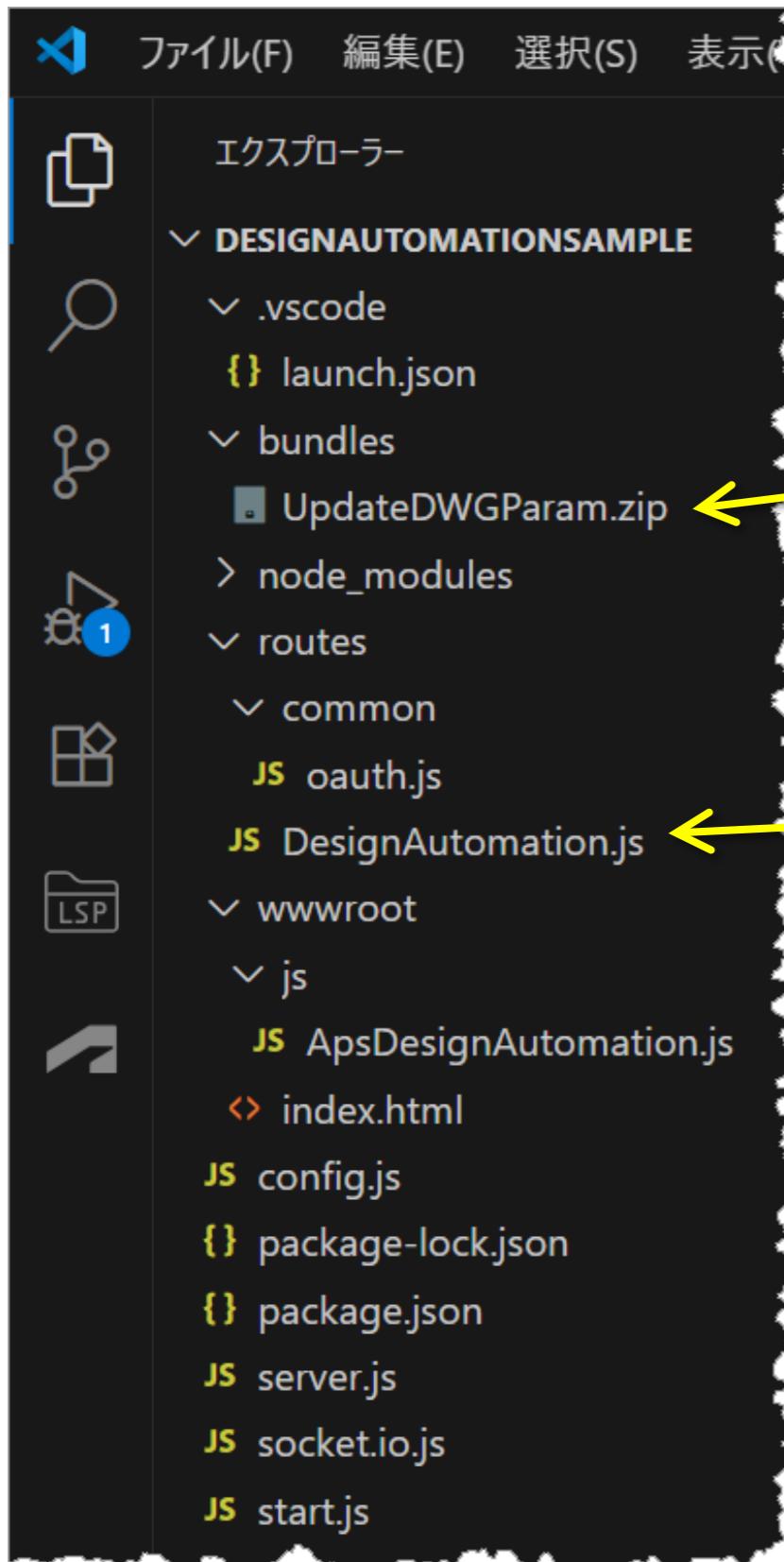
Design Automation uses `.bundle` just like the Autodesk App Store, meaning you need to create a `PackageContents.xml` and a ZIP with the DLL (and other required files). For detailed information on how to create them, please visit Autodesk App Store Developer Center.

At this section we will create a basic plugin that update width and height parameter and save the resulting file. Also the supporting files (`PackageContents.xml`) and the folder structure to place them. Finally create a .ZIP file ready to upload to Design Automation.

In the root folder, create a `bundles` folder.

```
DESIGNAUTOMATIONSAMPLE
  .vscode
  launch.json
  bundles
  node_modules
  routes/common
    oauth.js
  wwwroot
```

ここまでプロジェクト構成





Activity の定義

Activity の定義 - Define Activity

The screenshot shows the Autodesk Platform Services website. The top navigation bar includes links for Solutions, Getting Started, Documentation, Success Stories, Community, Support, Pricing, and App Store. A search bar and a sign-in button are also present. The left sidebar contains links for Getting Started, Environment Setup, Tutorials (Simple Viewer, Hubs Browser, Dashboard), Design Automation (Create Server, Basic UI, Create Plugin), Define Activity (the current page), and Execute Workitem. The main content area displays the 'Define Activity' tutorial, which explains what an activity is and provides a sample with 2 inputs and 1 output. It lists supported environments: Node.js & VSCode, .NET & VSCode, and .NET & VS2022. The 'Define Activity' section is currently selected.

AUTODESK Platform Services

Solutions Getting Started Documentation Success Stories Community Support Pricing App Store

Getting Started Environment Setup

Tutorials

- Simple Viewer
- Hubs Browser
- Dashboard

Design Automation

- Create Server
- Basic UI
- Create Plugin

Define Activity

Execute Workitem

Getting Started > Tutorials > Design Automation > Define Activity

Define Activity

Activity is the specification of an action that can be executed using a specified engine. It specifies the number of input and output files, and the AppBundle and entry-point to use.

In this tutorial sample, the activity has 2 inputs (file & JSON data) and 1 output (file).

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

- Activity

Now we will write endpoints for creating new activity and getting the existing

Activity – アクティビティ

1. WorkItem が処理する入出力ファイル/パラメータを宣言
2. WorkItem が実行するコマンド名を指定
 - Activity の種類
 - **パブリック共有 Activity**
 - オートデスクが提供する Activity
 - **カスタム Activity (独自の Activity)**
 - アドインによるカスタム コマンド
 - 標準コマンド (アドイン不要 = AppBundle 不要)
 - スクリプト

VS Code 上の操作手順

1. DesignAutomation.js ファイルに Activity 定義の登録を実装
2. デバッグ実行 >> **localhost:8080**



Define Activity | Autodesk Platform Services +

get-started.aps.autodesk.com/tutorials/design-automation/define-activity

AUTODESK Platform Services  Sign in

Solutions Getting Started Documentation Success Stories Community Support Pricing App Store

Getting Started Environment Setup  Tutorials Design Automation Define Activity

Tutorials

Simple Viewer
Hubs Browser
Dashboard

Design Automation

Create Server
Basic UI
Create Plugin

Define Activity

Execute Workitem 

ACC Administrator
ACC Issues (beta)

Learn More <<

Define Activity

Activity is the specification of an action that can be executed using a specified engine. It specifies the number of input and output files, and the AppBundle and entry-point to use.

In this tutorial sample, the activity has 2 inputs (file & JSON data) and 1 output (file).

[Node.js & VSCode](#) [.NET & VSCode](#) [.NET & VS2022](#)

- Activity

Now we will write endpoints for creating new activity and getting the existing activities, copy the following code into `DesignAutomation.js` file before the last line `module.exports = router;`

```
routes/DesignAutomation.js
```

```
//> <summary>
```

AppBundle と Activity の登録

Create/Update AppBundle & Activity

You just need to define AppBundle & Activity once.
If the plugin code changed, redefine (creates a new version).

Select a local AppBundle:

UpdateDWGParam

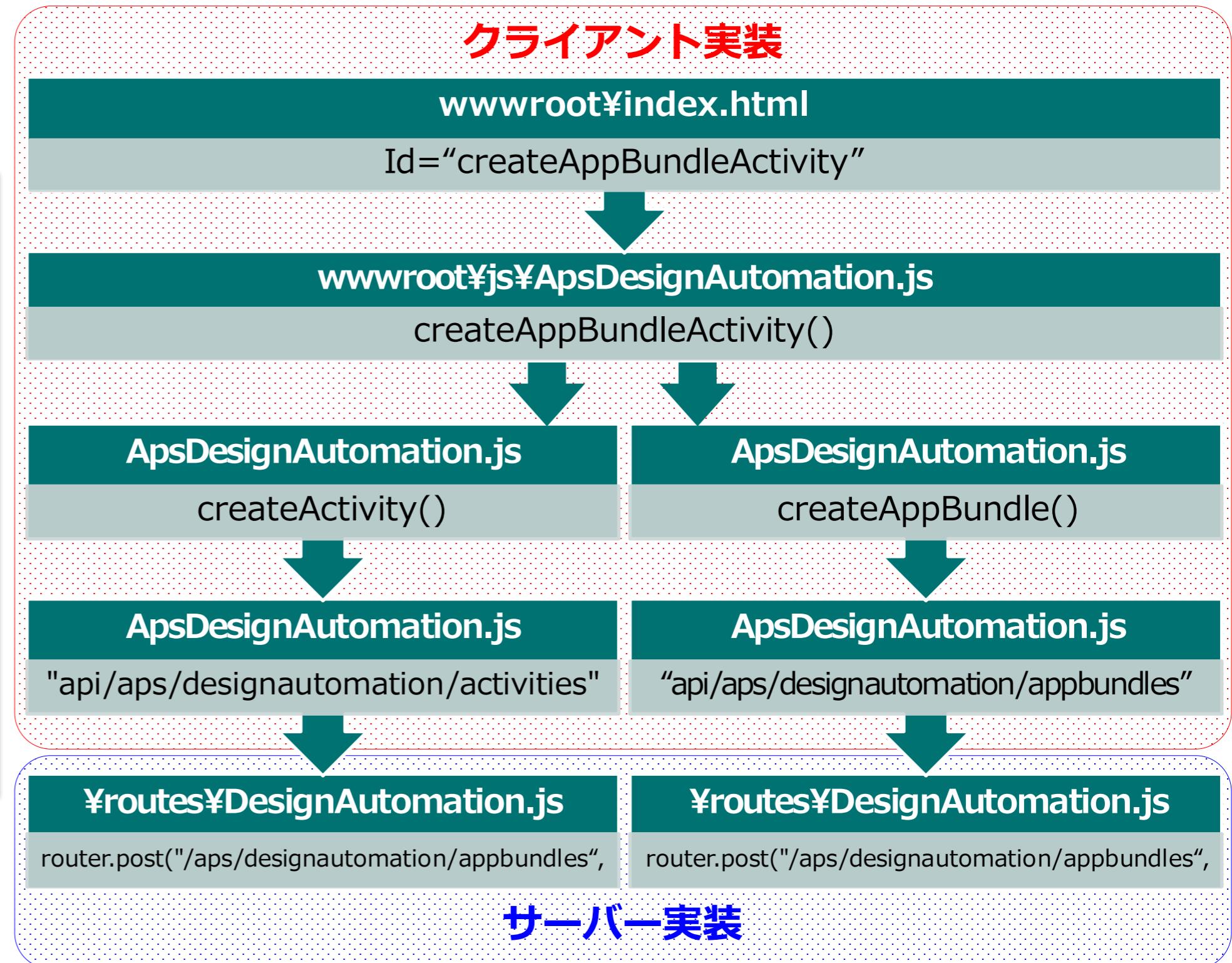
Tip: Make sure .ZIP bundles are placed at /bundles/ folder

Select engine:

Autodesk.AutoCAD+24_3

For this sample the .ZIP name is used as suffix to define **AppBundle** and **Activity** names.
Activities will have file and params input, and file output.

Clear account **Create/Update**

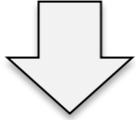


Automation API で扱う ID

- AppBundle と Activity で利用
 - Client ID に関連付けられる
 - 例)


```
"nqpwqsDLFGkSO6LgA2mvaSXy5AeH5VSJ.UpdateDWGParam+dev"
```
 - App 名
 - AppBundle と Activity 登録時に ID に指定
 - Alias 名
 - AppBundle と Activity バージョン識別に利用
 - 例) prod、dev
- WorkItem は作成時に個別 ID を割り当てる

ID と Nickname (ニックネーム)

- Client ID に関連付けられた AppBundle と Activity の ID
 - 例)
"nqpwqsDLFGkS06LgA2mvaSXy5AeH5VSJ. UpdateDWGParam+dev"
- Nickname のマッピング登録が可能 (任意)
 - 例)
“nqpwqsDLFGkS06LgA2mvaSXy5AeH5VSJ” を“DAS_Japan_AutoCAD”

"DAS_Japan_AutoCAD.UpdateDWGParam+dev"
- 注意：Nickname 登録後 Client ID 名指定は使用不可に

AppBundle と Activity の共有

- AppBundle/Activity のパブリック共有は不可
- 共有先 Client ID を指定した場合のみ可
 - [POST activities/:id/aliases](#)、[POST appbundles/:id/aliases](#)
 - リクエスト ボディに “receiver” 値を指定

```
{  
  "version": 1,  
  "id": "dev",  
  "receiver": [  
    "mDyW9ps2613NZ8AIaQVrlFYueRi6dwS",  
    "AjFukUWeRk05eA9XpH8Nnh62BzPD60mg"  
  ]  
}
```

- 配列指定時
 - 少なくとも 1 つの AppBundle/Activityリソース登録が必要



WorkItem の実行

WorkItem の実行 - Execute Workitem

The screenshot shows the Autodesk Platform Services website with a dark header. The header includes the Autodesk logo, a search bar, and a 'Sign in' button. Below the header is a navigation bar with links: Solutions, Getting Started, Documentation, Success Stories, Community, Support, Pricing, and App Store. On the left side, there is a sidebar with links for Getting Started, Environment Setup, Tutorials (Simple Viewer, Hubs Browser, Dashboard), Design Automation (Create Server, Basic UI, Create Plugin, Define Activity, Execute Workitem), and a Run & Debug section with links for Using the sample and Troubleshooting.

Execute Workitem

A job that executes a specified Activity, using specified input files and generating appropriate output files.

The relationship between an Activity and WorkItem can be thought of as a “function definition” and “function call”, respectively. The Activity specifies the AppBundle(s) to use, which in turn specify the Engine to use. The Workitem is then called to execute those.

In this tutorial sample, the workitem specifies the input file URL, the input JSON data with the new parameter values, and the destination URL for the output file. This sample will upload the input file to a OSS bucket before starting the workitem.

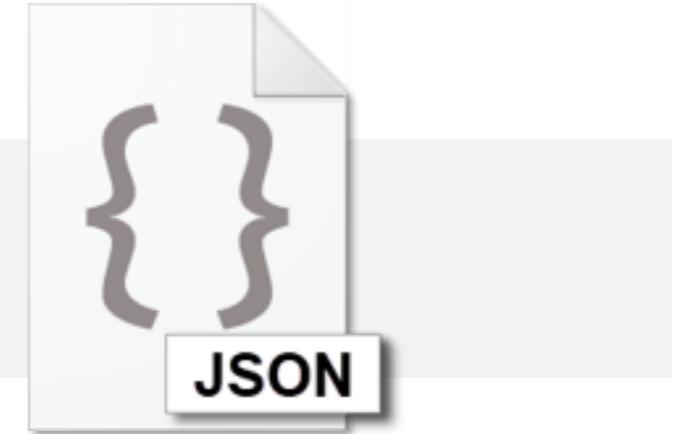
WorkItem – ワークアイテム

- 実際にクラウド上で処理されるタスクの単位
- 定義済 Activity を参照する WorkItem を作成
- Activity の応じた入出力パラメータ値の指定
 - 入力パラメータ
 - 入出力ファイル
- WorkItem の進捗状態の監視と処理終了の検出

WorkItem 実行時に起こること

- AppBundle (アドイン) は実行時に作成の仮想環境に展開
- Automation API は WorkItem で指定された値を利用
 - 指定のクラウドストレージから DWG をダウンロード (任意)
 - **指定の値を持つ JSON ファイルを作成**

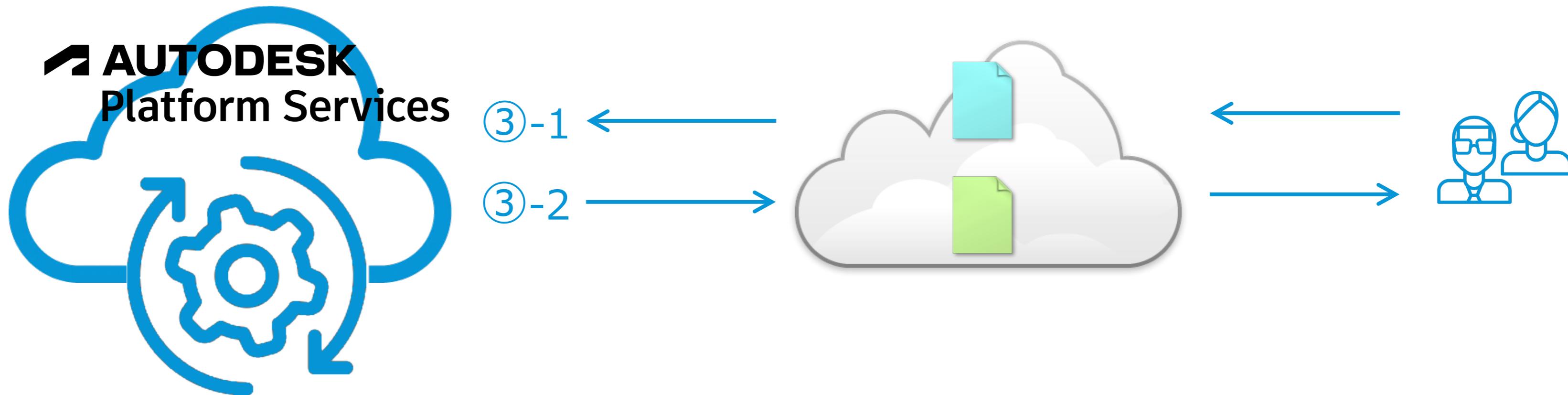
```
{  
  'width':999,  
  'height':555  
}
```



- 成果ファイルを指定のクラウドストレージへ保存 (アップロード)
 - DWG や PDF など
- WorkItem 終了後に仮想環境は自動削除 (含む使用ファイル)
 - キャッシュして後日の WorkItem で使用することは不可

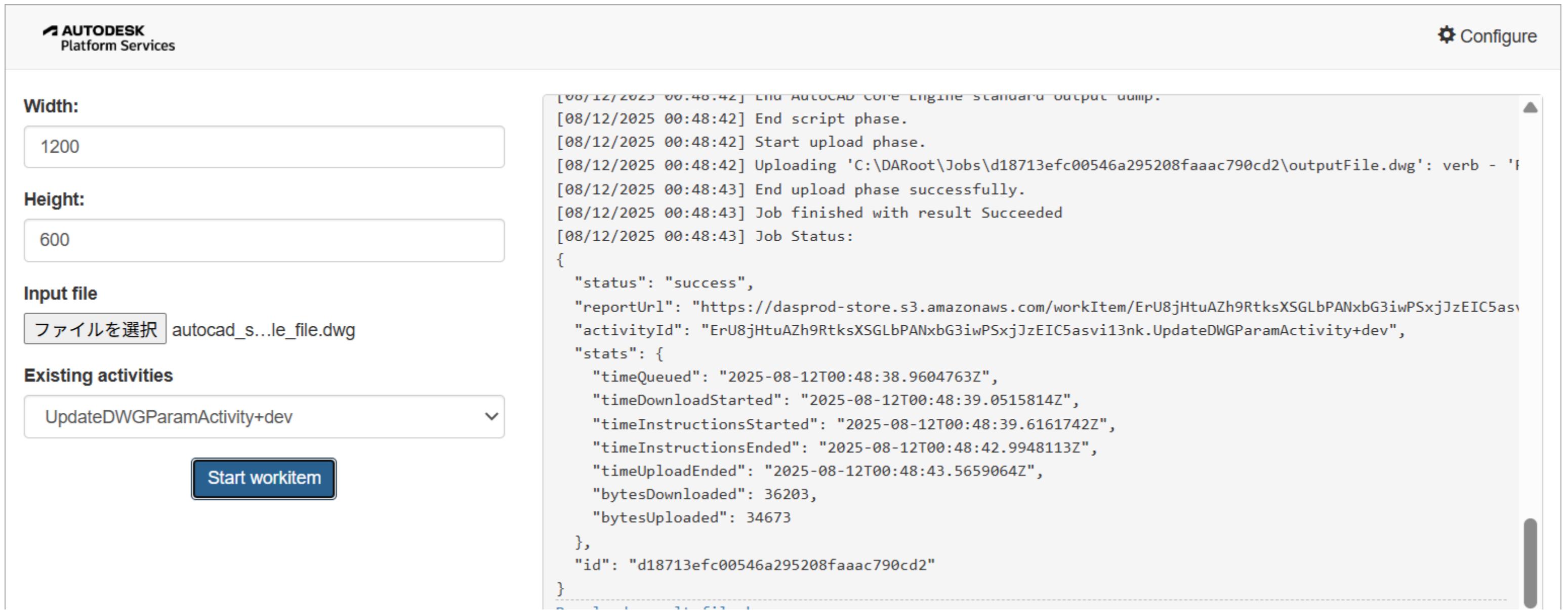
入出力ファイル指定は Direct-to-S3 か署名付き URL

- Automation API 用 Direct-to-S3 指定
 - OSS Bucket の場合：
`urn:adsk.objects:os.object:<Bucket name>/<Object key>`



VS Code 上の操作手順

1. DesignAutomation.js ファイルに WorkItem 実行処理の実装
2. デバッグ実行 >> localhost:8080



The screenshot shows the Autodesk Platform Services interface for configuring a work item. On the left, there are input fields for 'Width' (1200) and 'Height' (600). Under 'Input file', a file named 'autocad_sample_file.dwg' is selected. In the 'Existing activities' dropdown, 'UpdateDWGParamActivity+dev' is chosen. A large button at the bottom labeled 'Start workitem' is visible. On the right, a terminal window displays the execution log:

```
[08/12/2025 00:48:42] End script phase.
[08/12/2025 00:48:42] Start upload phase.
[08/12/2025 00:48:42] Uploading 'C:\DARoot\Jobs\d18713efc00546a295208faaac790cd2\outputFile.dwg': verb - 'F
[08/12/2025 00:48:43] End upload phase successfully.
[08/12/2025 00:48:43] Job finished with result Succeeded
[08/12/2025 00:48:43] Job Status:
{
    "status": "success",
    "reportUrl": "https://dasprod-store.s3.amazonaws.com/workItem/ErU8jHtuAZh9RtkXSGLbPANxbG3iwPSxjJzEIC5asvi13nk.UpdateDWGParamActivity+dev",
    "activityId": "ErU8jHtuAZh9RtkXSGLbPANxbG3iwPSxjJzEIC5asvi13nk.UpdateDWGParamActivity+dev",
    "stats": {
        "timeQueued": "2025-08-12T00:48:38.9604763Z",
        "timeDownloadStarted": "2025-08-12T00:48:39.0515814Z",
        "timeInstructionsStarted": "2025-08-12T00:48:39.6161742Z",
        "timeInstructionsEnded": "2025-08-12T00:48:42.9948113Z",
        "timeUploadEnded": "2025-08-12T00:48:43.5659064Z",
        "bytesDownloaded": 36203,
        "bytesUploaded": 34673
    },
    "id": "d18713efc00546a295208faaac790cd2"
}
```

Execute Workitem | Autodesk Platform X +

get-started.aps.autodesk.com/tutorials/design-automation/execute-workitem

AUTODESK Platform Services Sign in

Solutions Getting Started Documentation Success Stories Community Support Pricing App Store

Getting Started Environment Setup **Tutorials** Simple Viewer Hubs Browser Dashboard **Design Automation** Create Server Basic UI Create Plugin Define Activity **Execute Workitem** ACC Administrator ACC Issues (beta)

Run & Debug Using the sample Troubleshooting

Home > Tutorials > Design Automation > Execute Workitem

Execute Workitem

A job that executes a specified Activity, using specified input files and generating appropriate output files.

The relationship between an Activity and WorkItem can be thought of as a “function definition” and “function call”, respectively. The Activity specifies the AppBundle(s) to use, which in turn specify the Engine to use. The Workitem is then called to execute those.

In this tutorial sample, the workitem specifies the input file URL, the input JSON data with the new parameter values, and the destination URL for the output file. This sample will upload the input file to a OSS bucket before starting the workitem.

CAUTION

Please note that executing an activity in the Design Automation service has a cost associated with it, see the [Pricing](#) page for more details. We recommend that you use a non-expired trial subscription when following this tutorial.

WorkItem の実行

AUTODESK
Platform Services

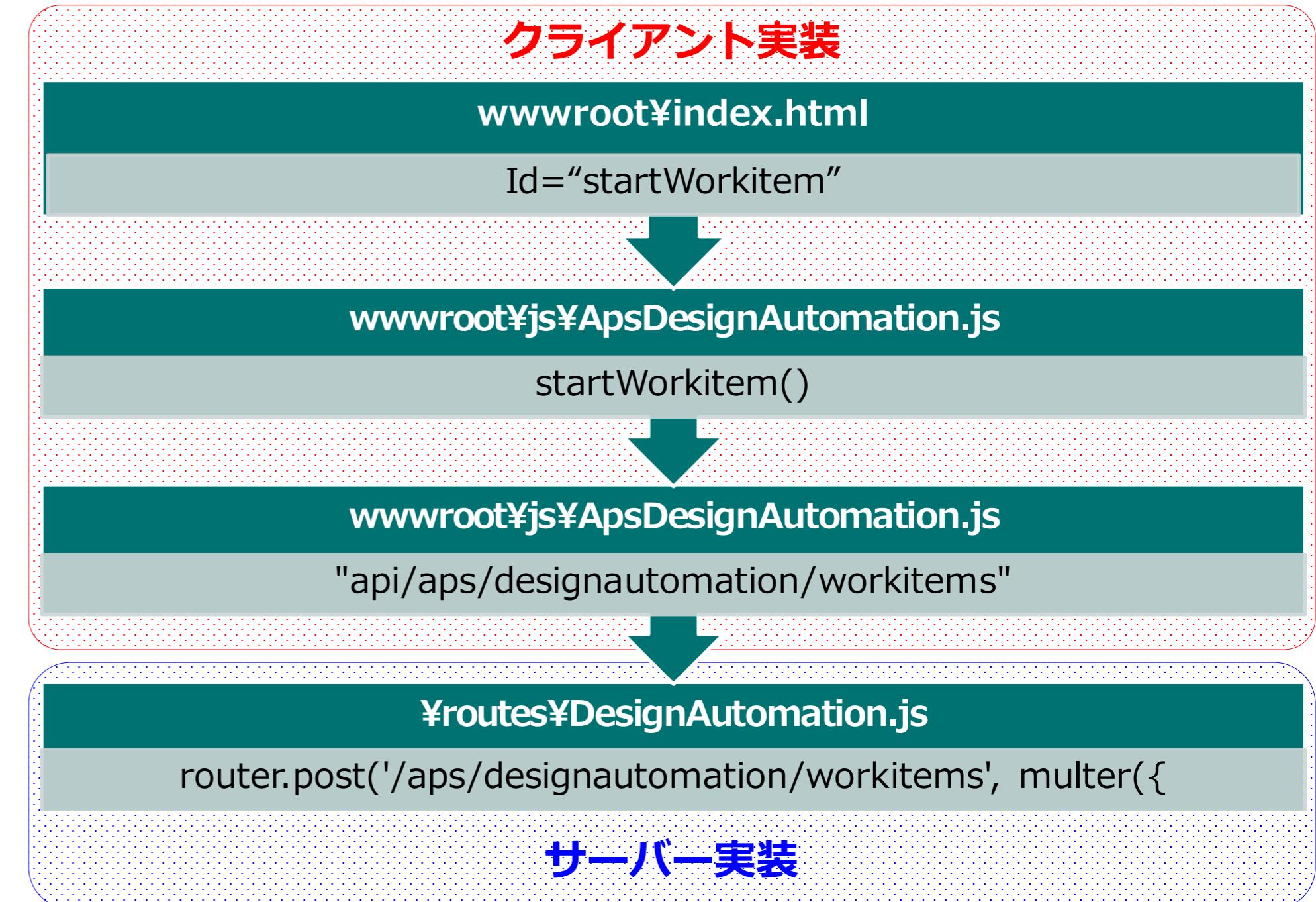
Width:
550

Height:
240

Input file
ファイルを選択 autocad_s...le_file.dwg

Existing activities
UpdateDWGParamActivity+dev

Start workitem



WorkItem 処理終了判定と処理レポートについて

- POST workitems エンドポイント呼び出し
 - 非推奨：150 rpm (150 回/1 分間) が上限
 - レスポンス ボディの **reportUrl** : WorkItem 毎に作成されるレポートログ
- OnComplete コールバックの実装
 - 推奨：要サーバー実装、ただし、クライアント↔サーバー間トラフィックが低減
 - 開発時のクライアント実装では ngrok ツール等の利用が必要
- Design Automation Tutorial の場合
 - クライアント↔サーバー実装はソケット通信
 - サーバー実装が POST workitems 呼び出し (&サーバーからプッシュ)

サーバー実装

```
¥socket.io.js & ¥routes¥DesignAutomation.js>autodesk.forge.designautomation パッケージ>bundle.js
```

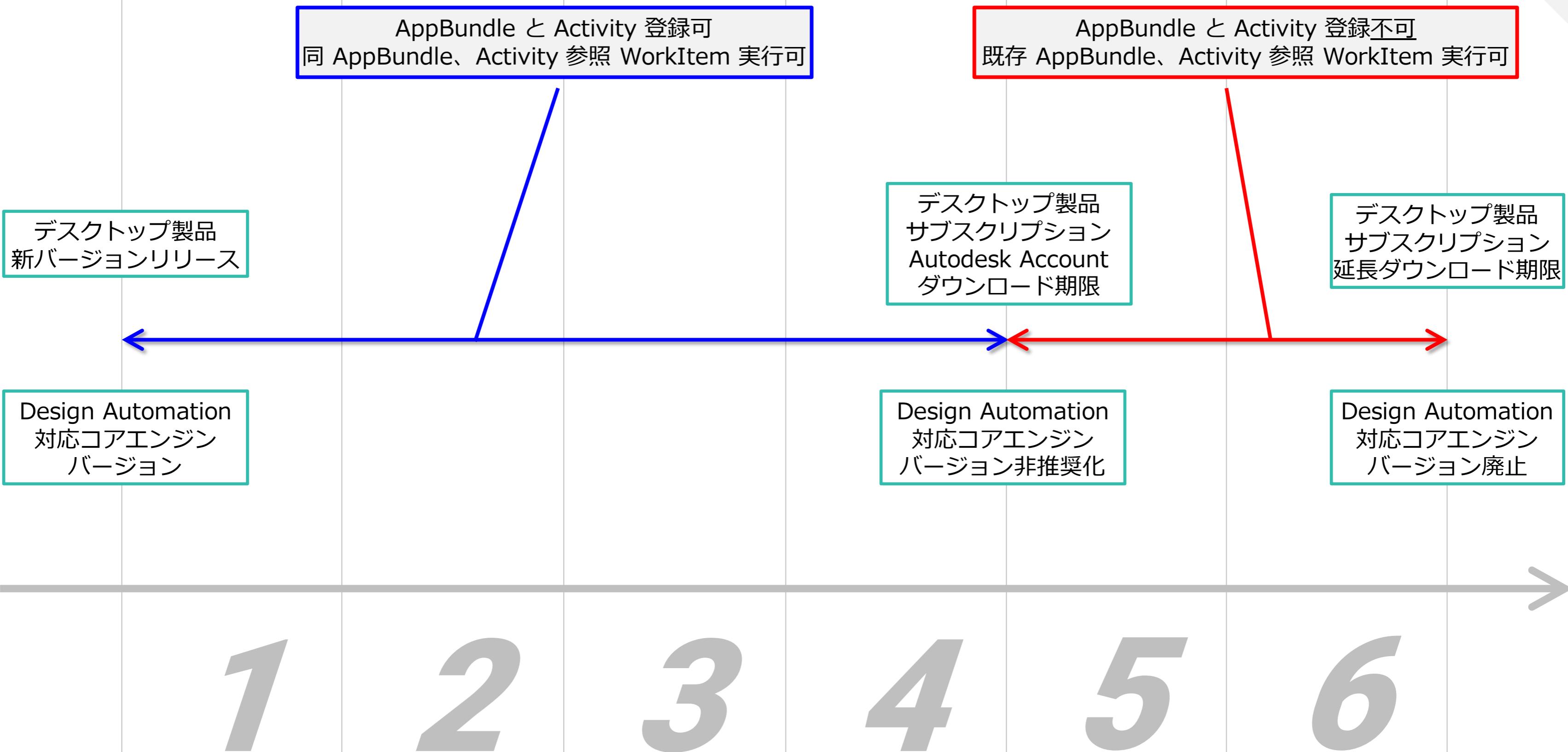
```
.getWorkitemStatus()
```



Automation API サポート ポリシー

Automation API サポートポリシー

1. コアエンジンは、対応するデスクトップ製品の最初のリリースから 4 年間サポートされます。
2. 4 年後にコアエンジンバージョンは非推奨となります、さらに 2 年間は利用可能で、その後削除されます。
3. 非推奨のエンジンバージョンでは、新しい AppBundle や Activity の登録は出来なくなりますが、登録済の AppBundle/Activity を参照する WorkItem は機能し続けます。
4. 削除されたコアエンジンバージョンを参照する WorkItem は実行出来ません。



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 スコープ 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インテナンスが必須
- 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

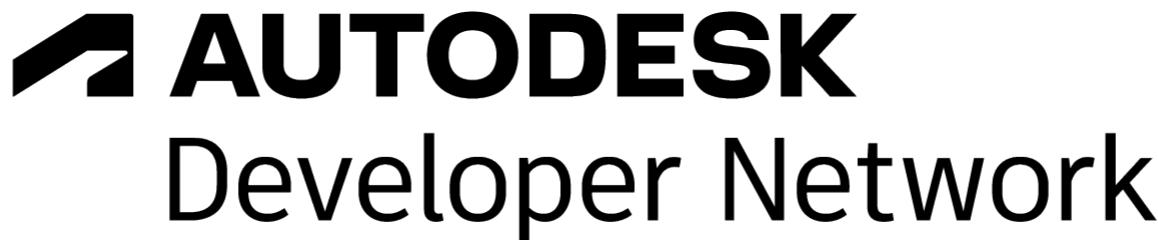
プレミアムAPIとサービス	コスト	情報
Fusion Automation API	3.0	Flex トークン / 処理時間
Automation API (その他)	2.0	Flex トークン / 処理時間
Flow Graph Engine API	1.0	Flex トークン / 処理時間
Model Derivative API	0.5	Flex トークン / コンプレックスジョブ
	0.1	Flex トークン / シンプルジョブ
Reality Capture API	1.0	50枚の写真毎のFlexトークン

- Autodesk Flex とは？

- <https://www.autodesk.com/jp/buying/flex>



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 (英語)
 - <https://APS.autodesk.com/en/support/get-help>
- ブログ :
 - <https://APS.autodesk.com/blog> (英語)
 - [Design Automation API for AutoCAD まとめ](#) (日本語)



Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2025 Autodesk. All rights reserved.