



BIM & Forge セミナー 2019

Forge Design Automation API for Revit での自動化

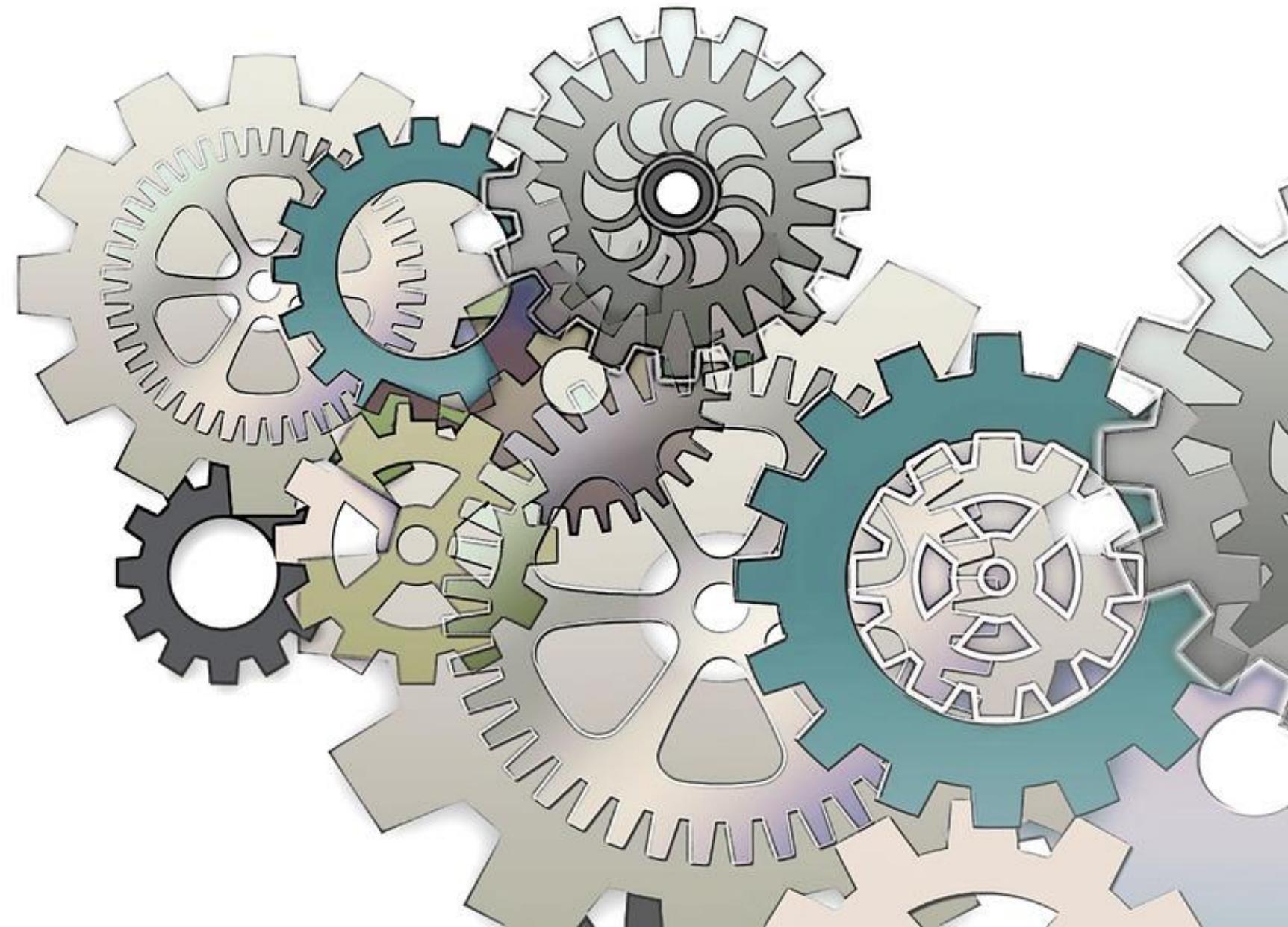
小笠原 龍司

Autodesk Developer Network, Forge Partner Development



Forge Design Automation API

- Design Automation API は、クラウド上で CAD エンジンのコア API を利用する環境を提供します。
- Forge プラットフォームを活用して、自動化されたジョブを、大規模かつ効率的に実行することができます。
 - 繰り返し行う処理
 - 頻度の高い処理
 - 大規模な演算処理能力を必要とする処理



Design Automation API Version 3

- Design Automation API で実行可能なコアエンジンに、従来の AutoCAD に加えて、Inventor、3ds Max、そして Revit が加わります。

Design Automation API
Version 1 & 2

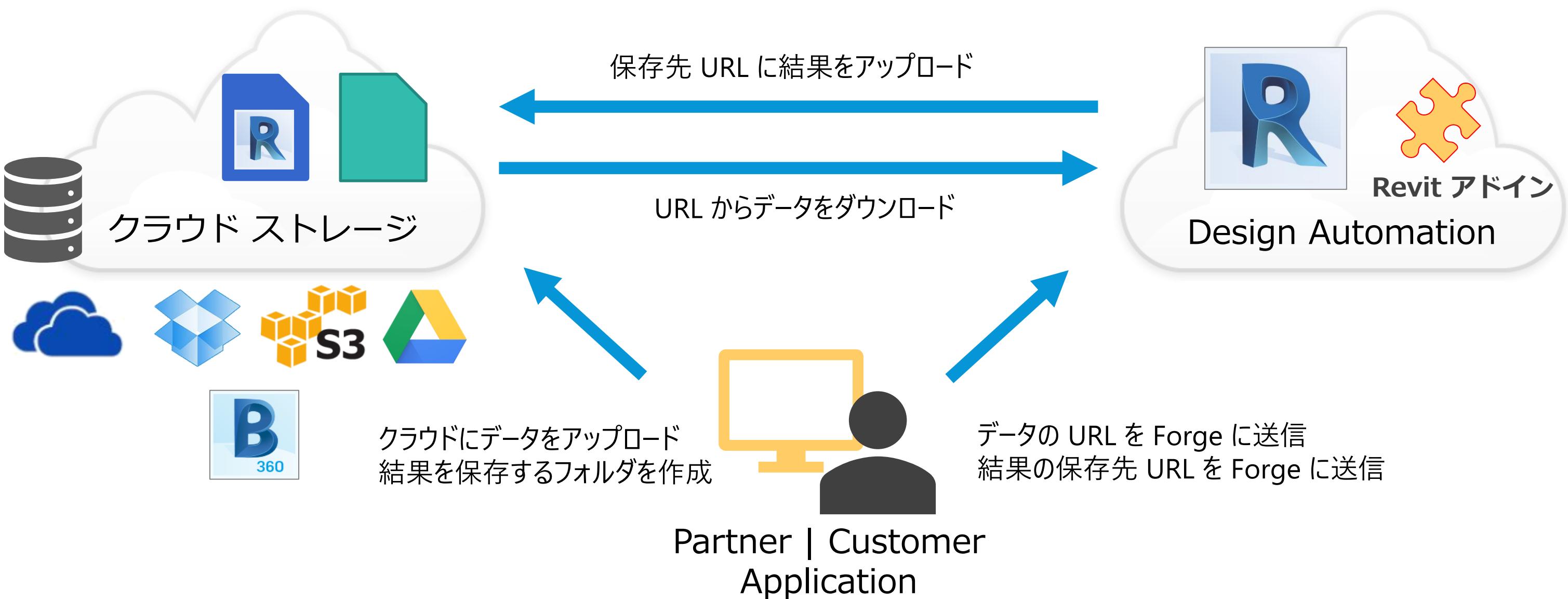


Design Automation API
Version 3

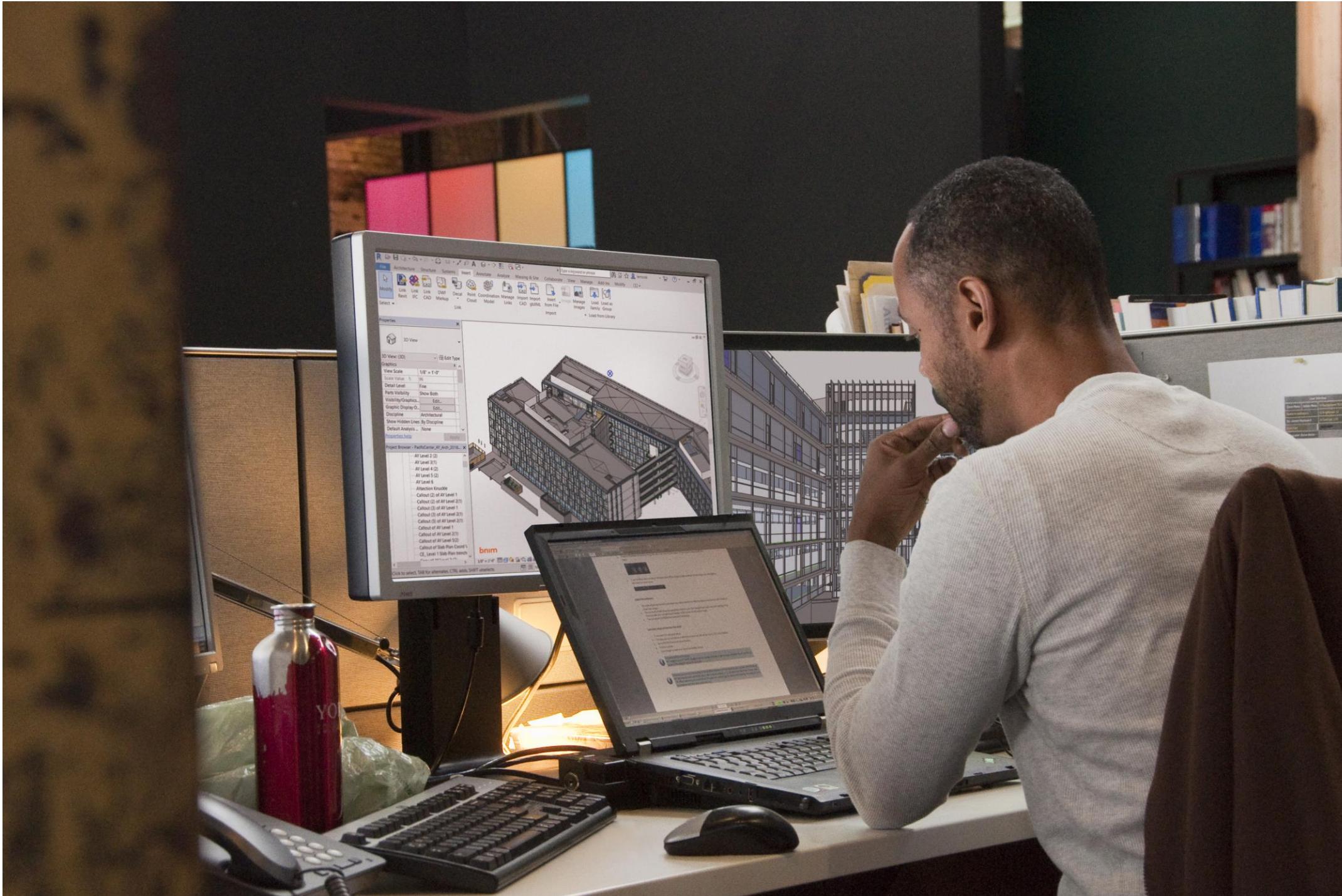


オンデマンドで Revit データをクラウド処理

- Revit がローカル環境になくても、クラウドサービスを通じて Revit アドインにフルアクセス（**ライセンス不要**）
- Revit API を利用して開発した Revit アドインを実行
- どこからでもデータを読み取り、どこにでもデータを保存



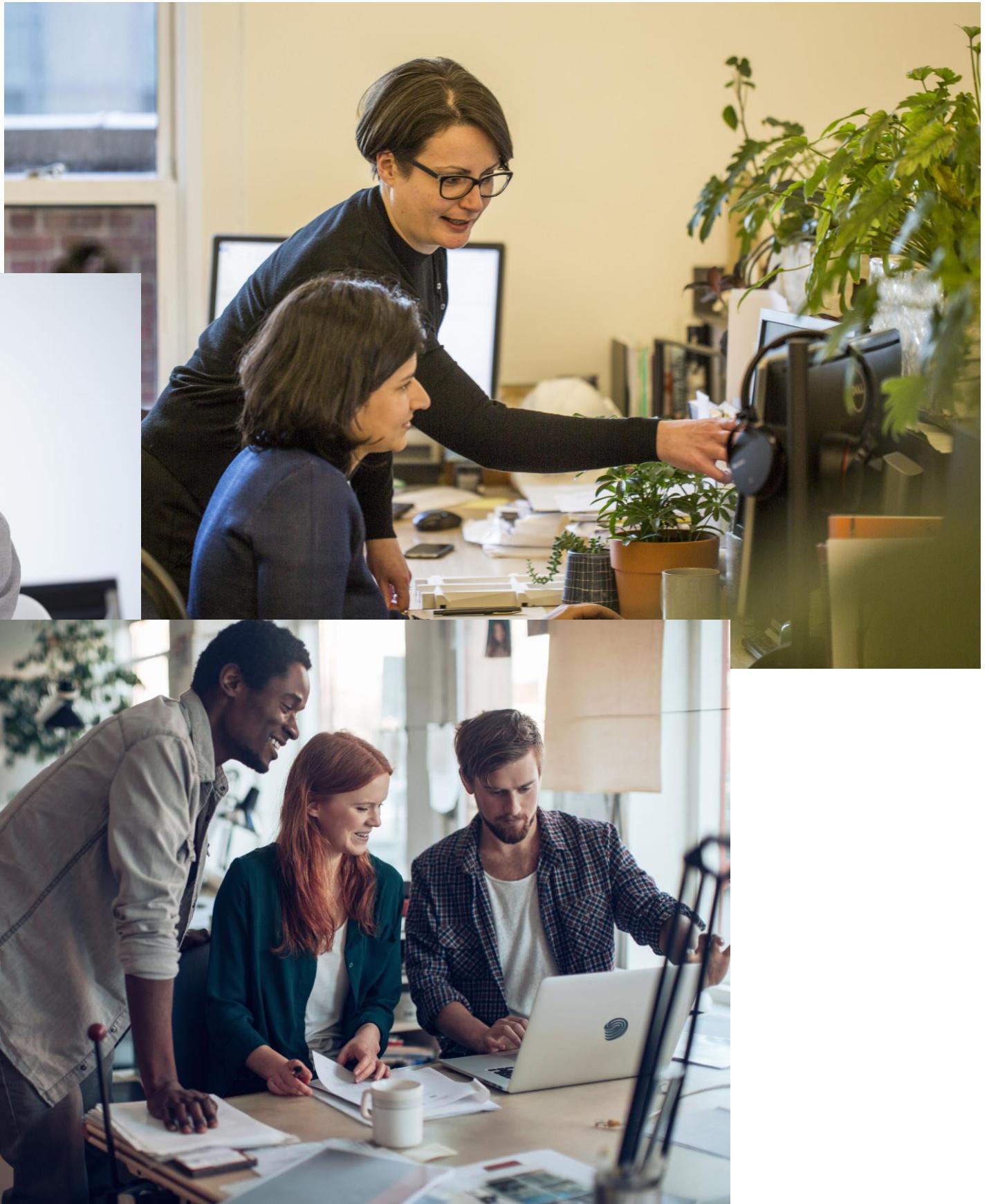
設計者のためのタスク自動化サービス



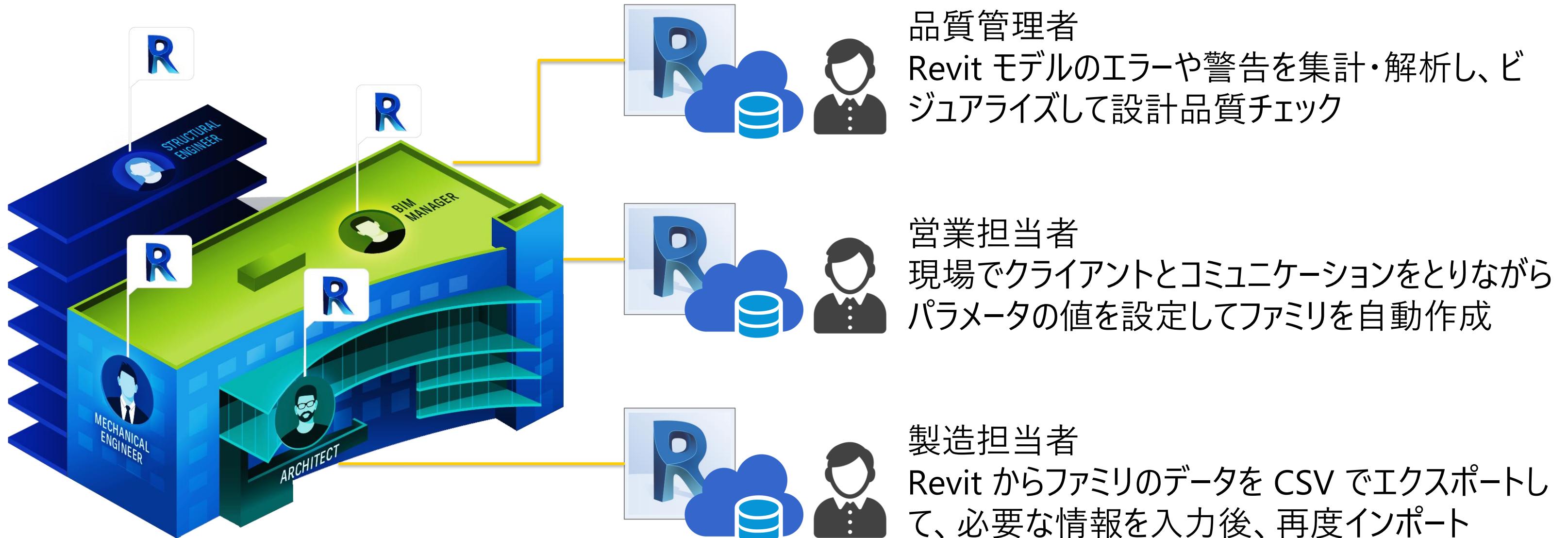
建築設計
MEP（機械、電気、配管）
構造エンジニアリング

幅広いユーザーが利用できる Revit サービス

- オーナー、クライアント
- 営業担当者
- ドキュメント管理者
- コスト管理者
- 品質管理者
- 設備管理者
- エネルギー管理者
- 施設管理
- 保守・メンテナンス

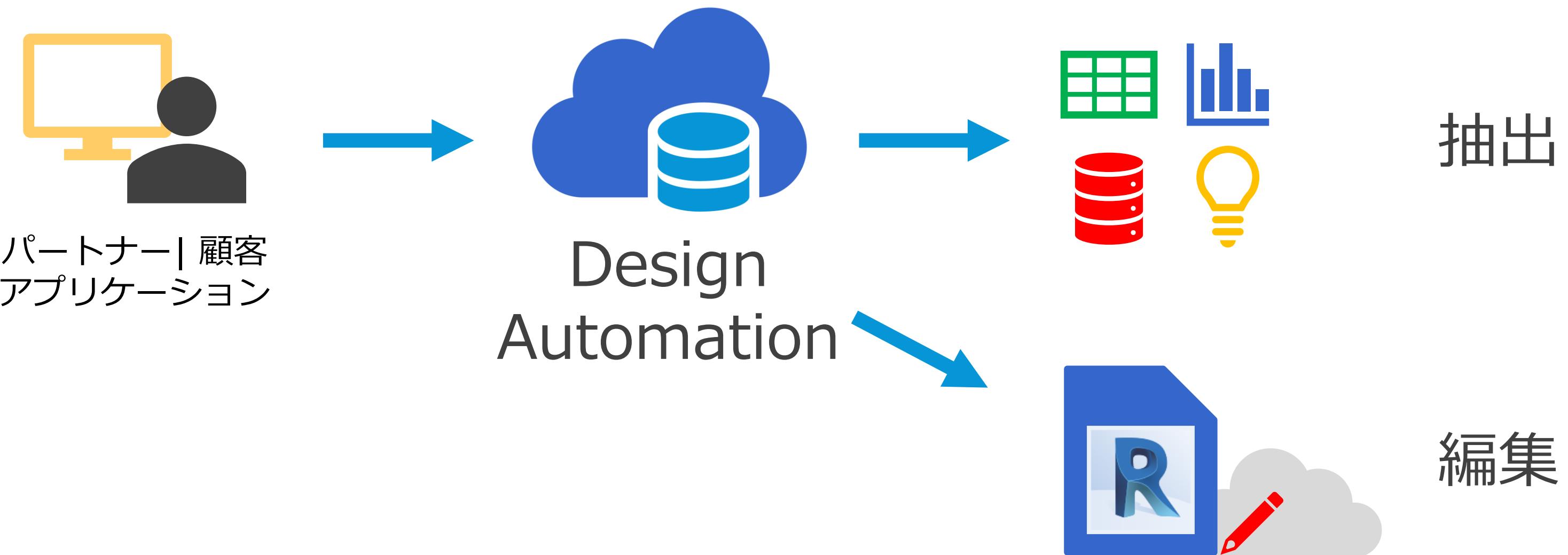


Revit のワークフローを拡張して、特定の業務課題を解決



- Design Automation API は、開発者向けプラットフォームです。
- 個々にソリューションを提供するわけではありません。

3つのワークフロー

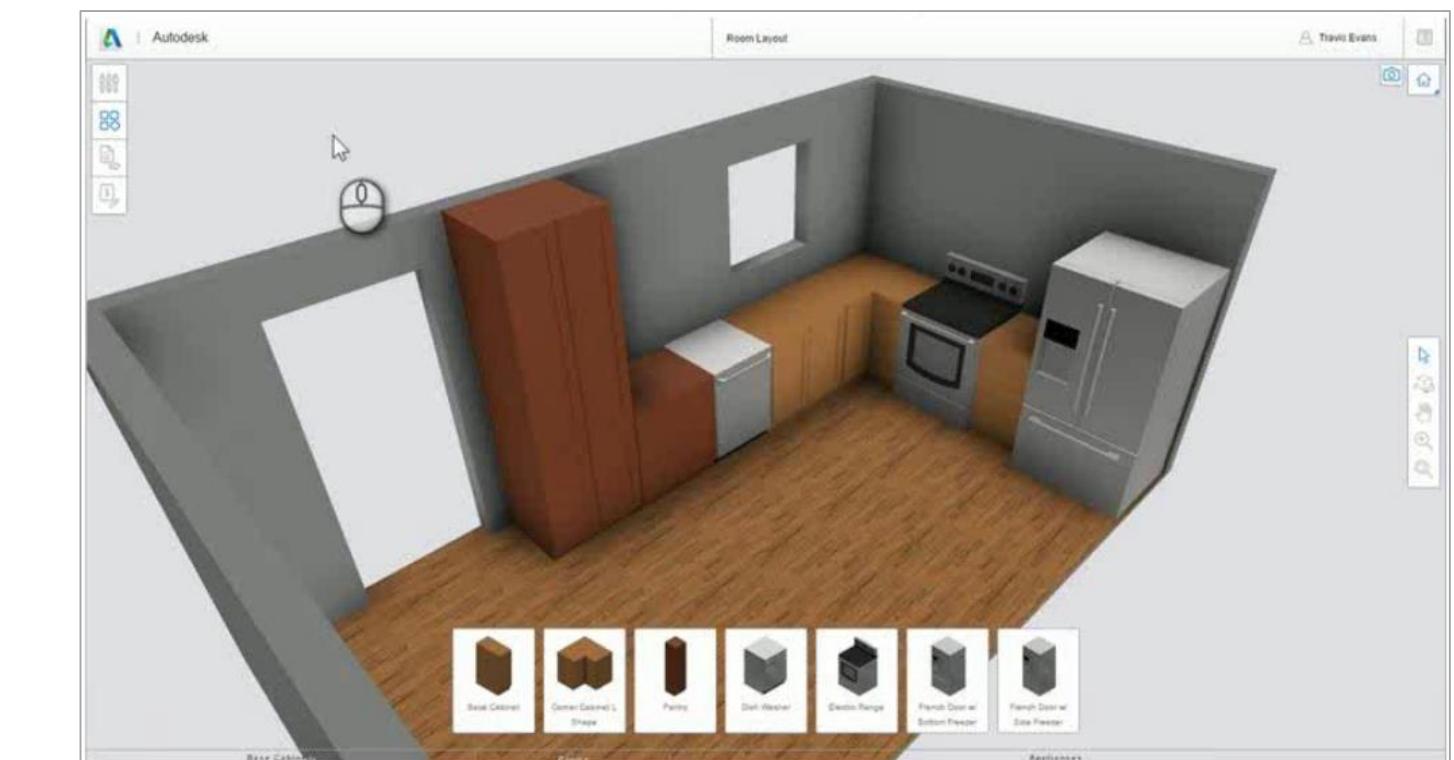
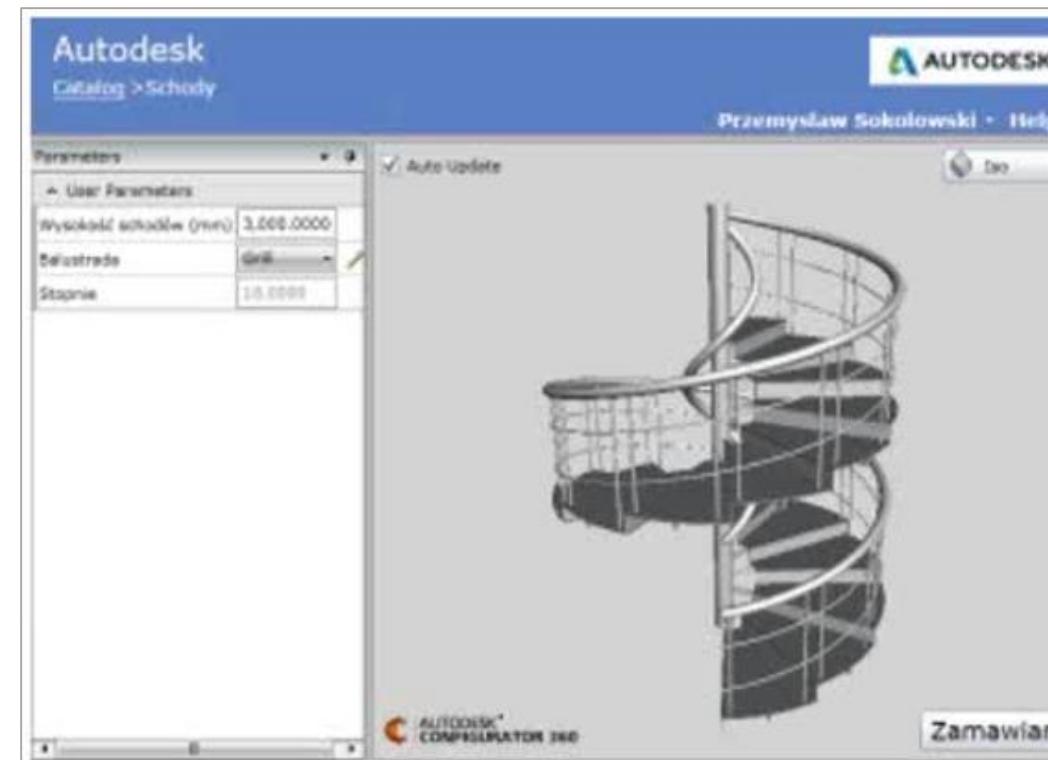


ワークフロー：作成

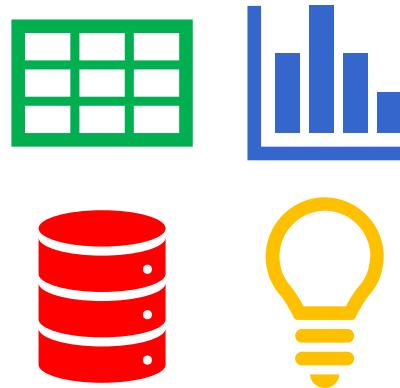


- ファミリ（RFA）を作成
- レイアウトアプリケーションから RVT プロジェクトを作成
- サードパーティのフォーマットから RVT/RFA に変換
- 仕様書から設計図書を自動生成

想定イメージ

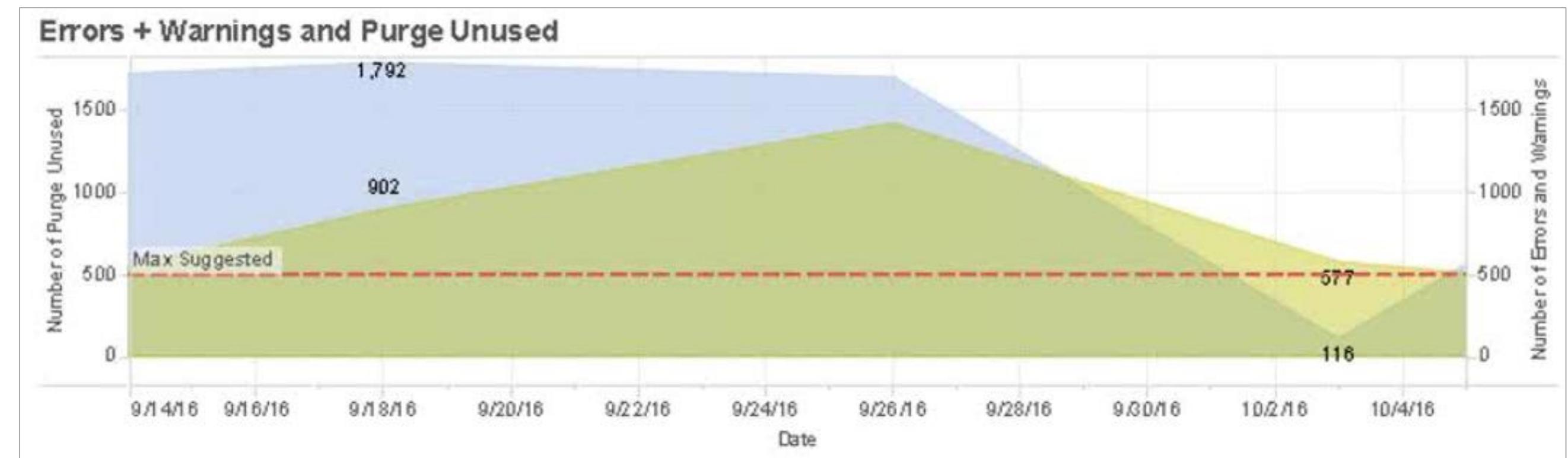


ワークフロー：抽出



- カスタムの抽出処理 (CSV、外部データベース、未サポートフォーマットなど)
- 抽出の自動実行
- モデルの品質保証/品質管理 (QA/QC) チェック
- Revit を所有していないコンサルタントとモデルの情報を共有

想定イメージ



ワークフロー：編集



- モデリングや設計図書の誤りを修正
 - Upgrade/Update のバッチ実行
 - 古いデータを自動的に最新版に更新
 - それ以外の編集処理
-
- 将来サポート予定
 - 定期的に Dynamo スクリプトを実行
 - BIM360 Design (C4R)モデル上でタスクを実行
 - デザインオプションの作成



想定イメージ



AUTODESK®
FORGE

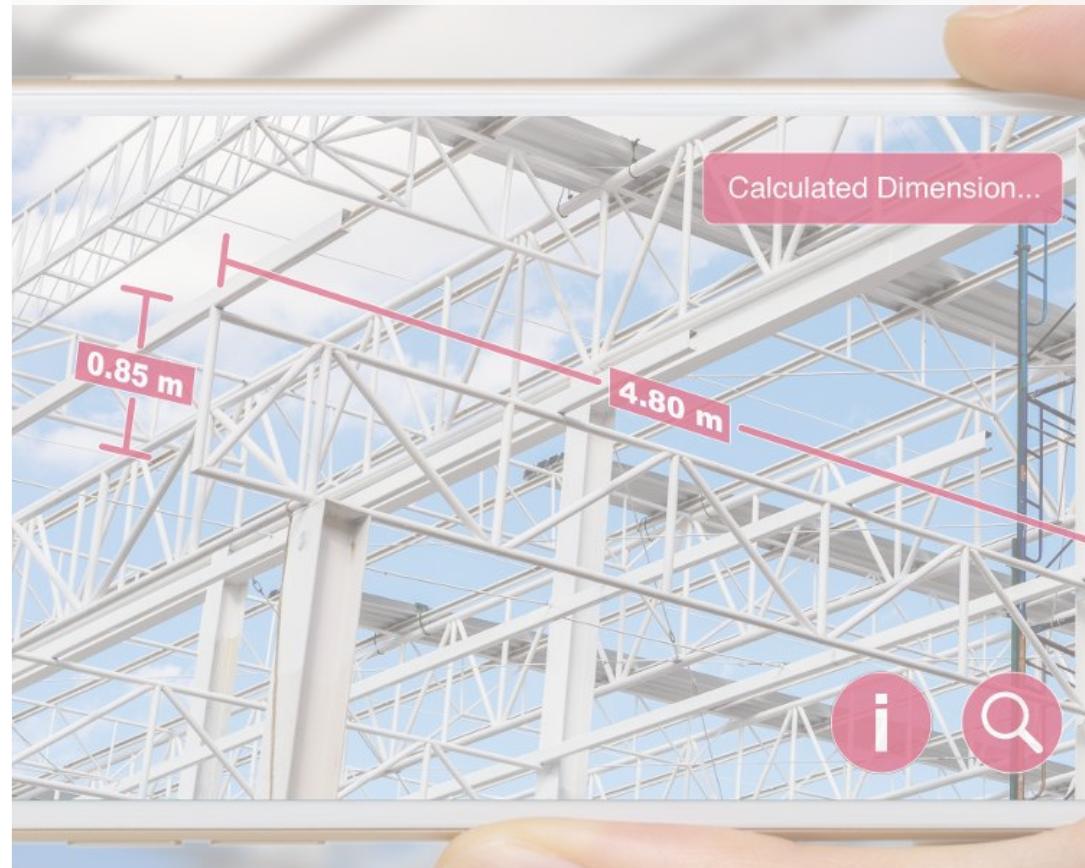
Design Automation API for Revit

クラウドサービスで Revit データを活用



作成

- カスタムコンテンツの生成
- モデル作成の自動化



抽出

- モデルデータの検索・分析
- レポートの自動生成



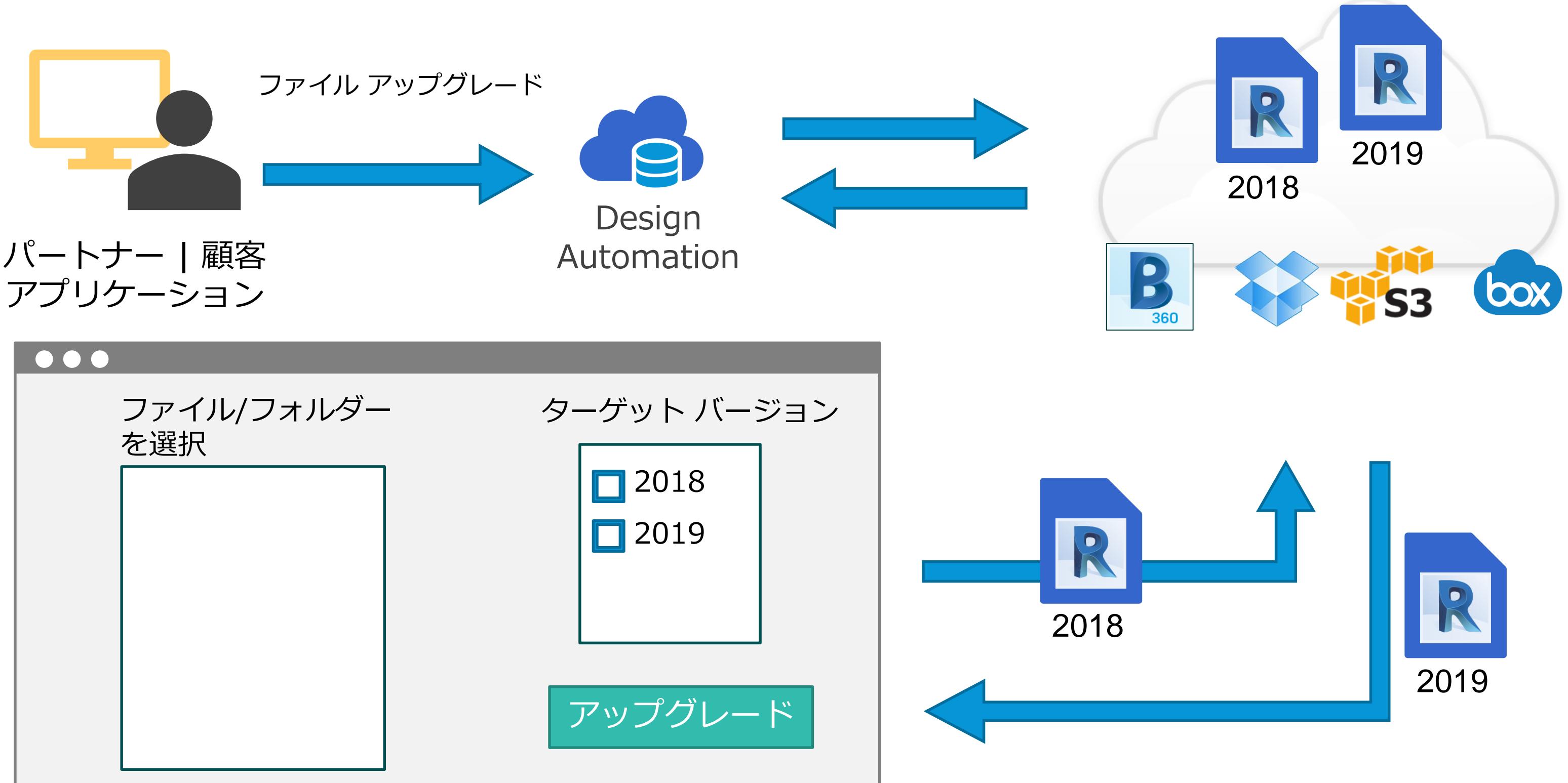
編集

- 企業の基準を維持
- 設計図書の自動生成

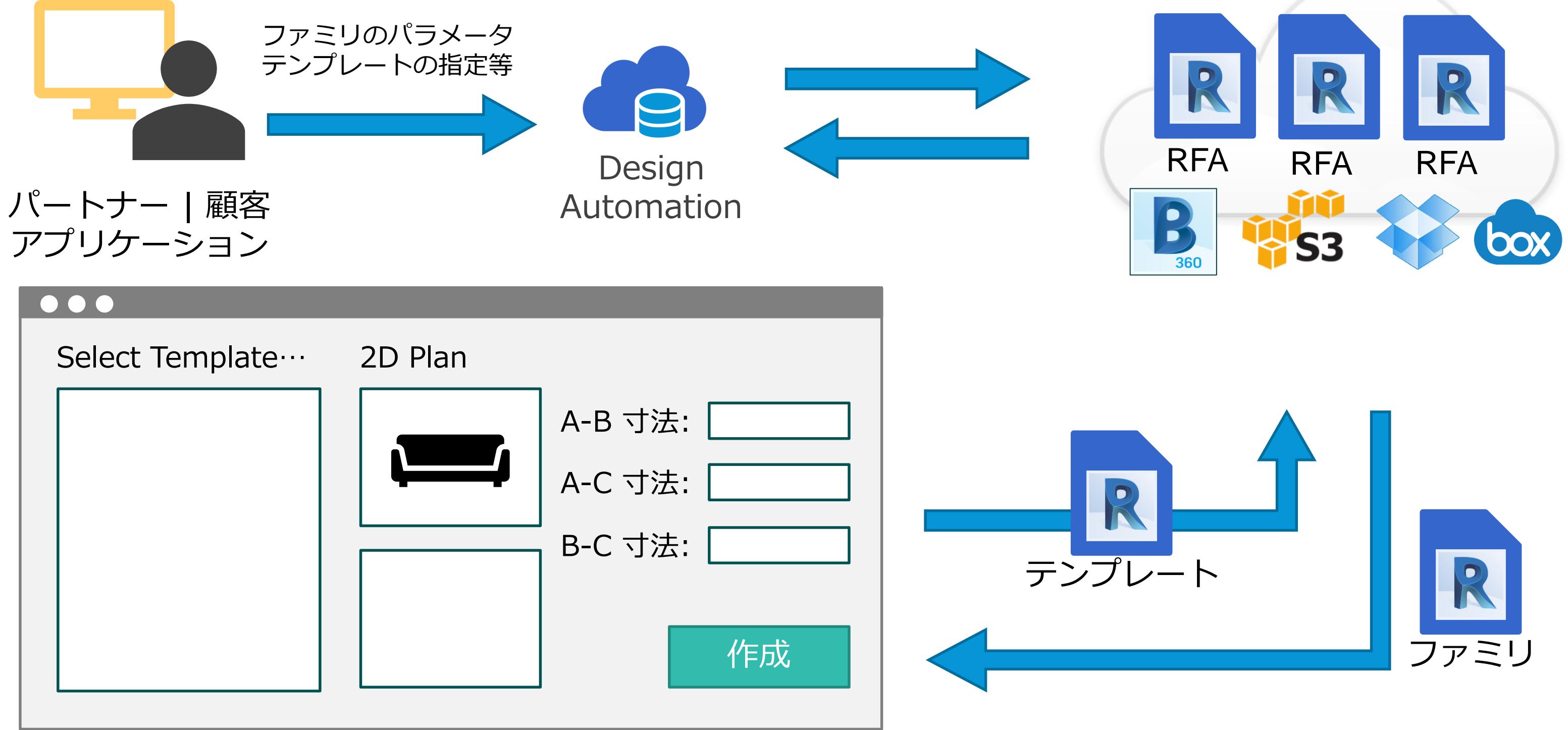
利用のシナリオ



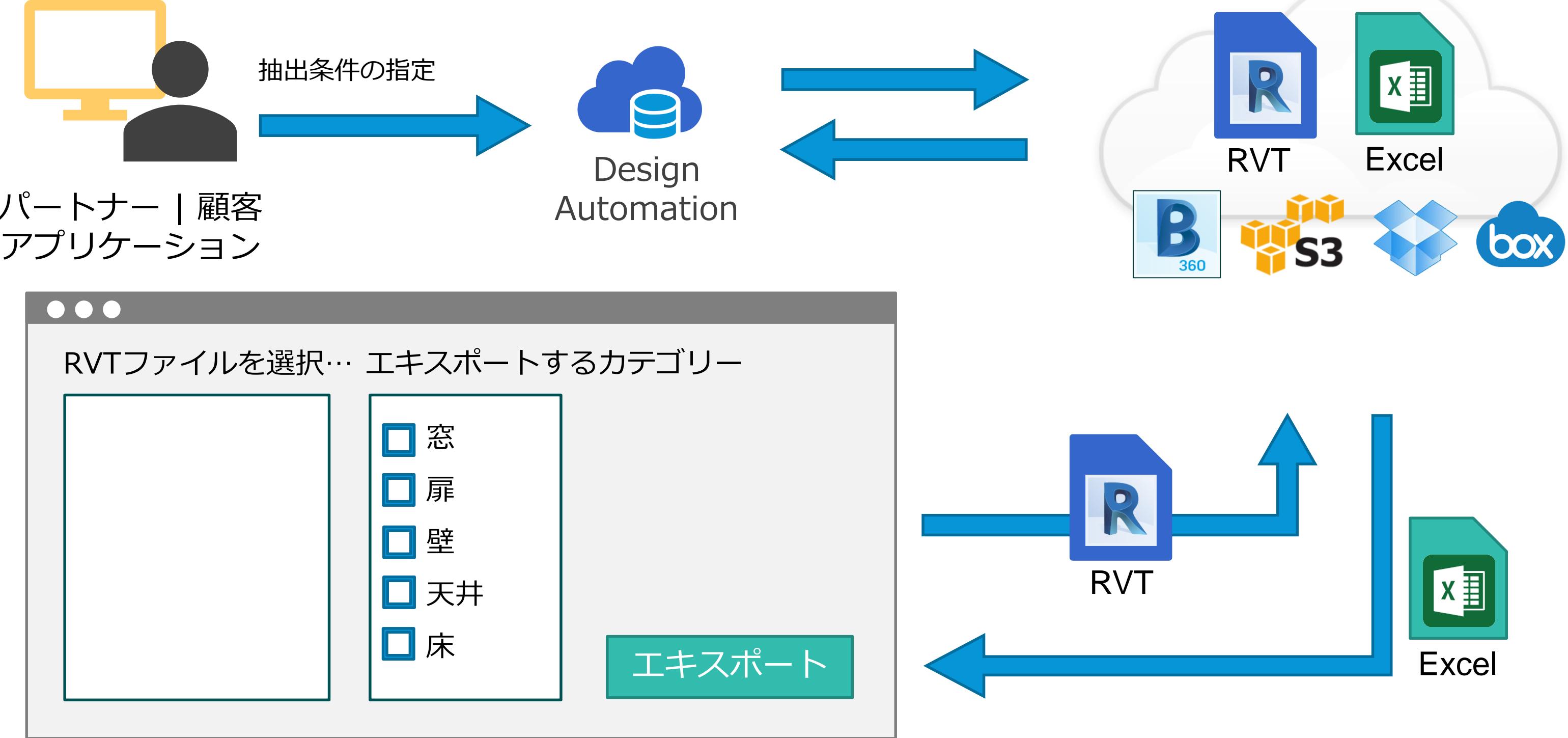
Revit ファイル アップグレーダー



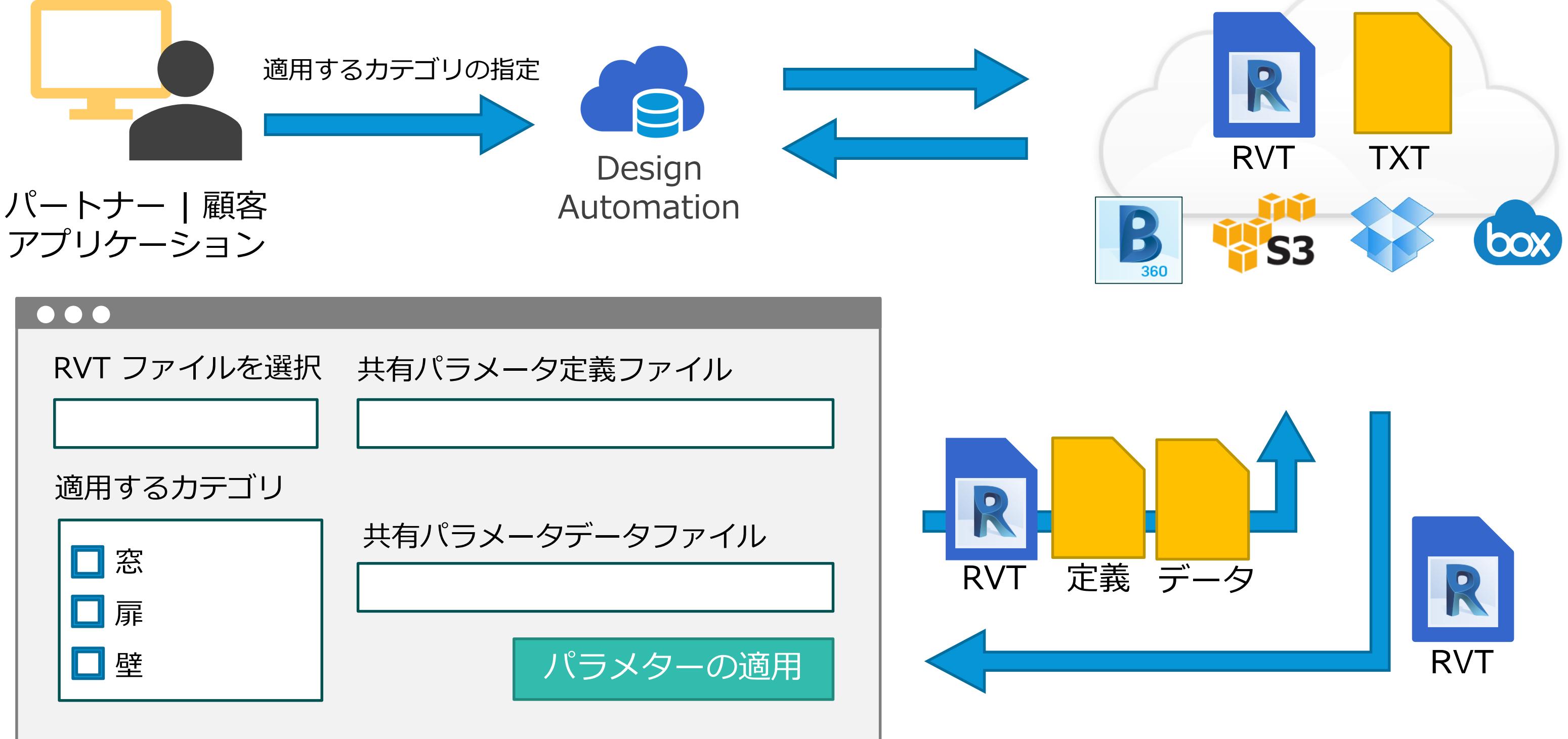
ファミリ (RFA) ジェネレーター



エクセルシート エクスポート



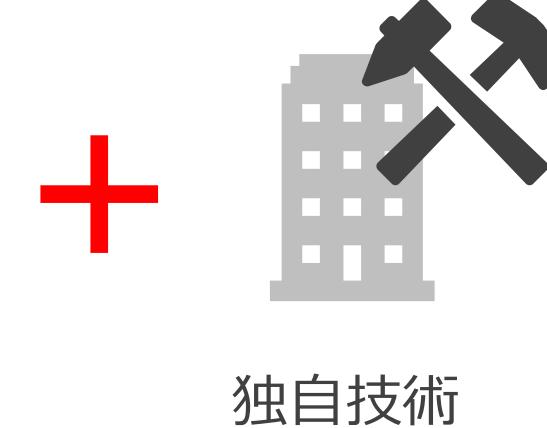
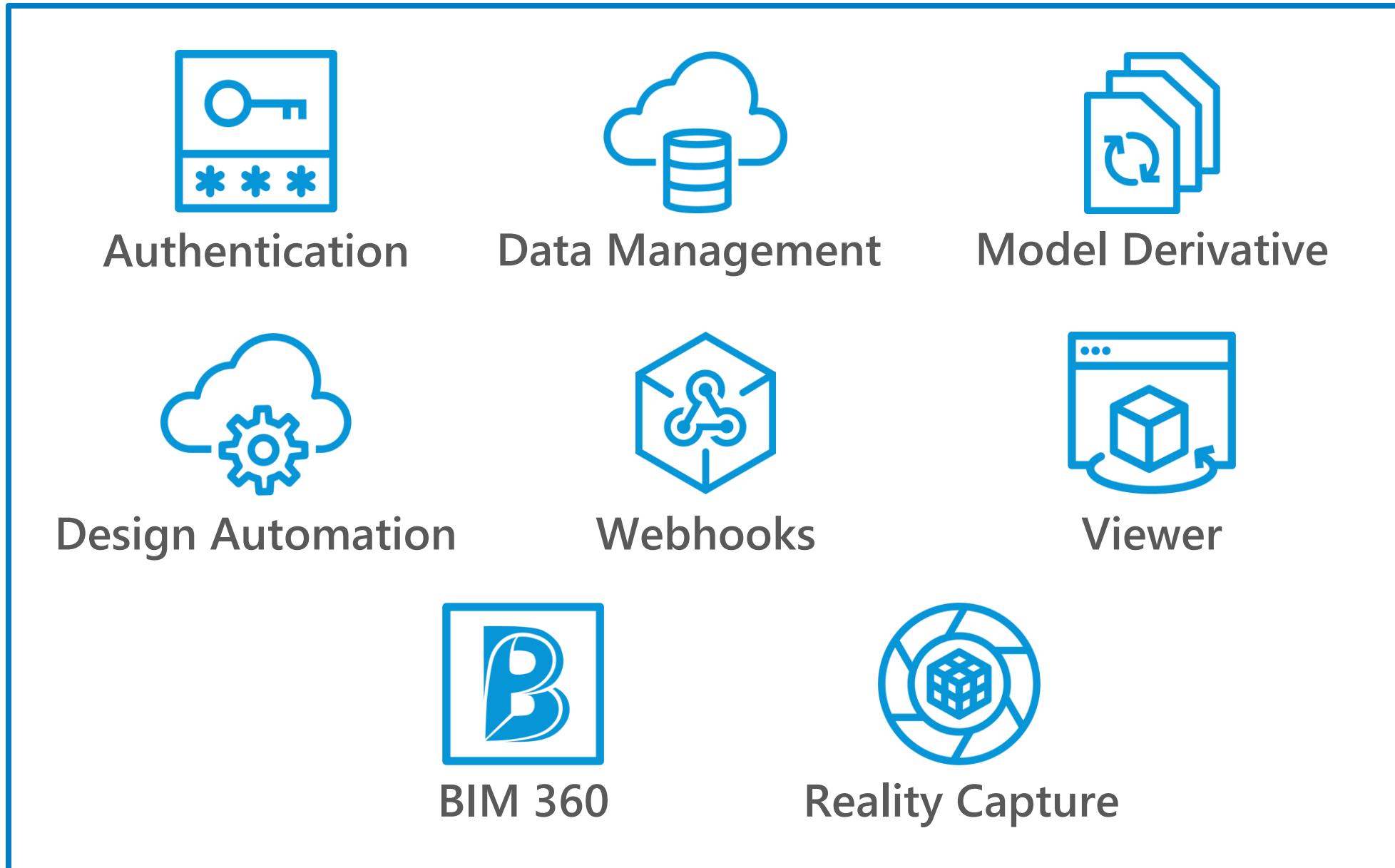
共有パラメータの一括更新



独自のクラウドサービスを作ることができます



AUTODESK® FORGE

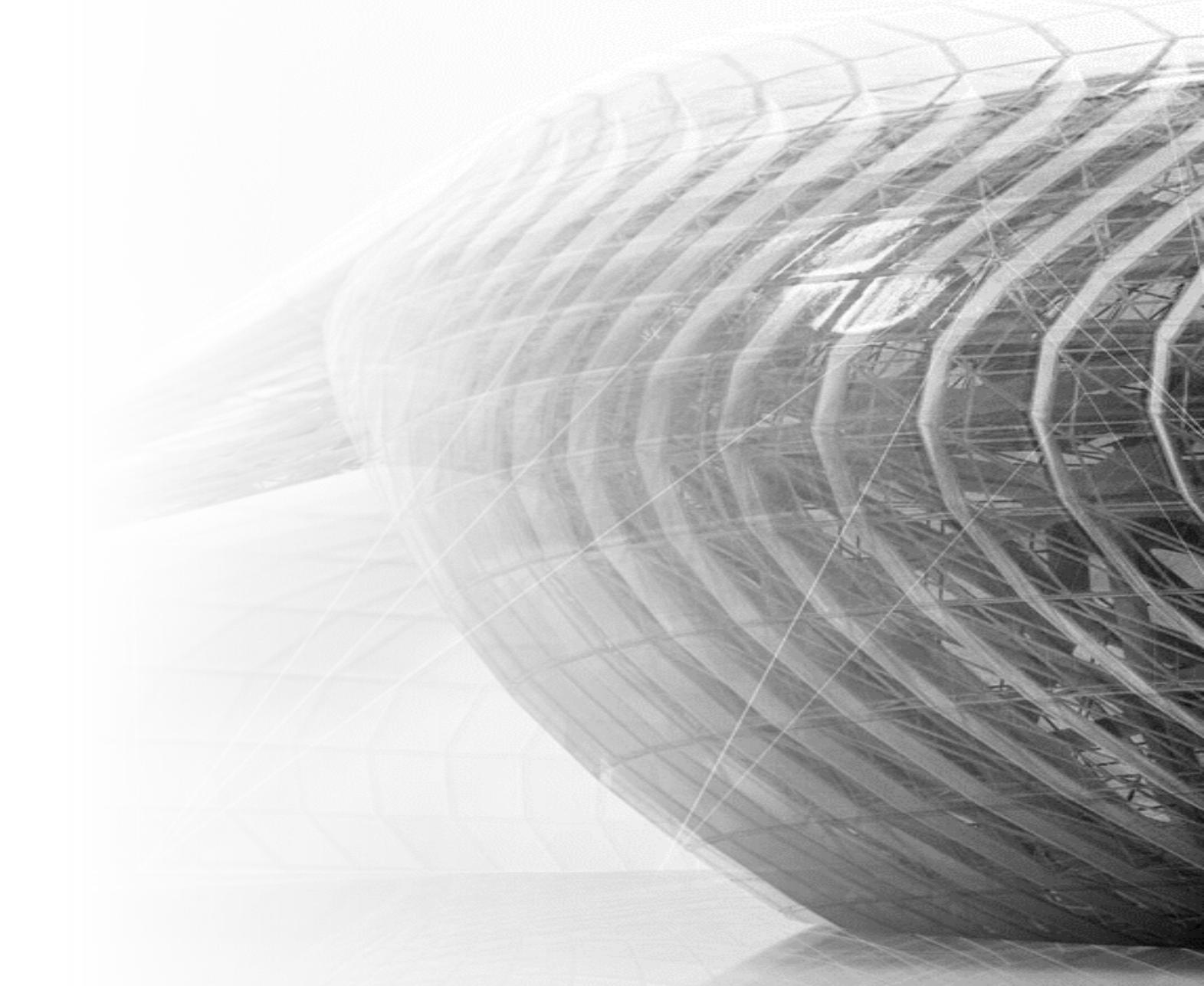


Web サービス API



社内データベース

API の仕組みと開発要件



3つのステップ

1. Revit アドインを作成

2. Revit アドインを Design Automation にアップロードして登録する Forge アプリケーションを作成

3. Revit モデルとその他リソースをクラウドストレージにアップロードして、Forge アプリケーションから Design Automation にジョブを依頼



1. Revit アドインを作成

- Revit API
 - C#、VB.NET、マネージ C++、その他

- .NET Framework
 - Revit 2019 -> .NET Framework 4.7
 - Revit 2018 -> .NET Framework 4.6

- Microsoft Visual Studio
 - 各バージョンに一致するアセンブリ出力が可能なもの

- 参照ライブラリ
 - RevitAPI.dll
 - Revit 2018: DesignAutomationBridge.dll
 - Revit 2019: DesignAutomationBridge.dll

```
using System;
using System.IO;
using Autodesk.Revit.ApplicationServices;
using Autodesk.Revit.DB;
using DesignAutomationFramework;

namespace DeleteWalls
{
    using Autodesk.Revit.Attributes.Regeneration(Autodesk.Revit.Attributes.RegenerationOption.Manual);
    using Autodesk.Revit.Attributes.Transaction(Autodesk.Revit.Attributes.TransactionMode.Manual);

    public class DeleteWallsApp : IExternalDBApplication
    {
        public ExternalDBApplicationResult OnStartup(Autodesk.Revit.ApplicationServices.ControlledApplication app)
        {
            DesignAutomationBridge.DesignAutomationReadyEvent += HandleDesignAutomationReadyEvent;
            return ExternalDBApplicationResult.Succeeded;
        }

        public ExternalDBApplicationResult OnShutdown(Autodesk.Revit.ApplicationServices.ControlledApplication app)
        {
            return ExternalDBApplicationResult.Succeeded;
        }

        public void HandleDesignAutomationReadyEvent(object sender, DesignAutomationEventArgs e)
        {
        }
    }
}
```

ソリューションエクスプローラー
DeleteWalls.csproj
DeleteWalls
DeleteWalls.cs
DeleteWalls.addin
DeleteWalls.cs
PackageContents.xml

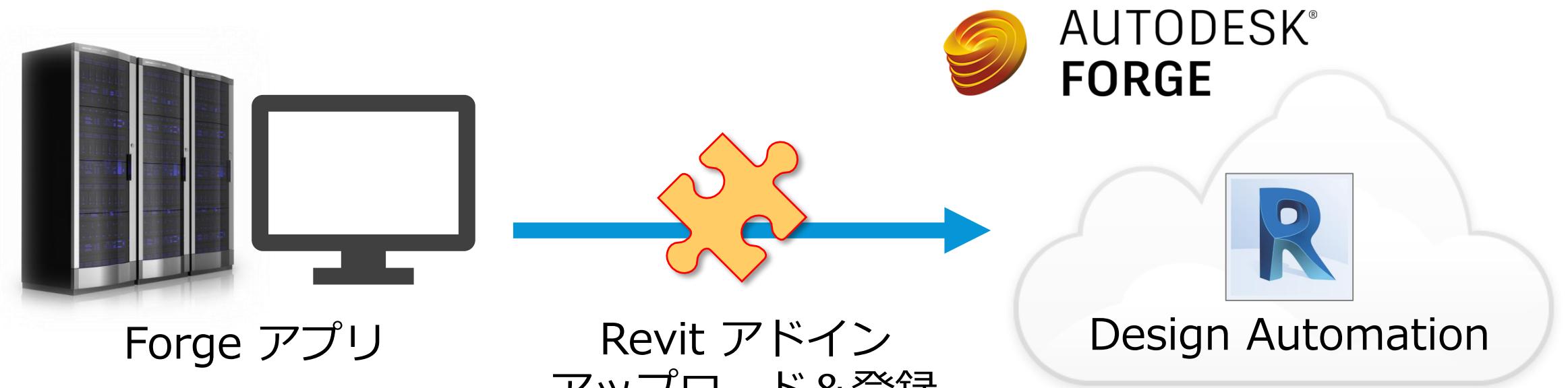
Revit アドイン



ローカルの Revit アドイン開発

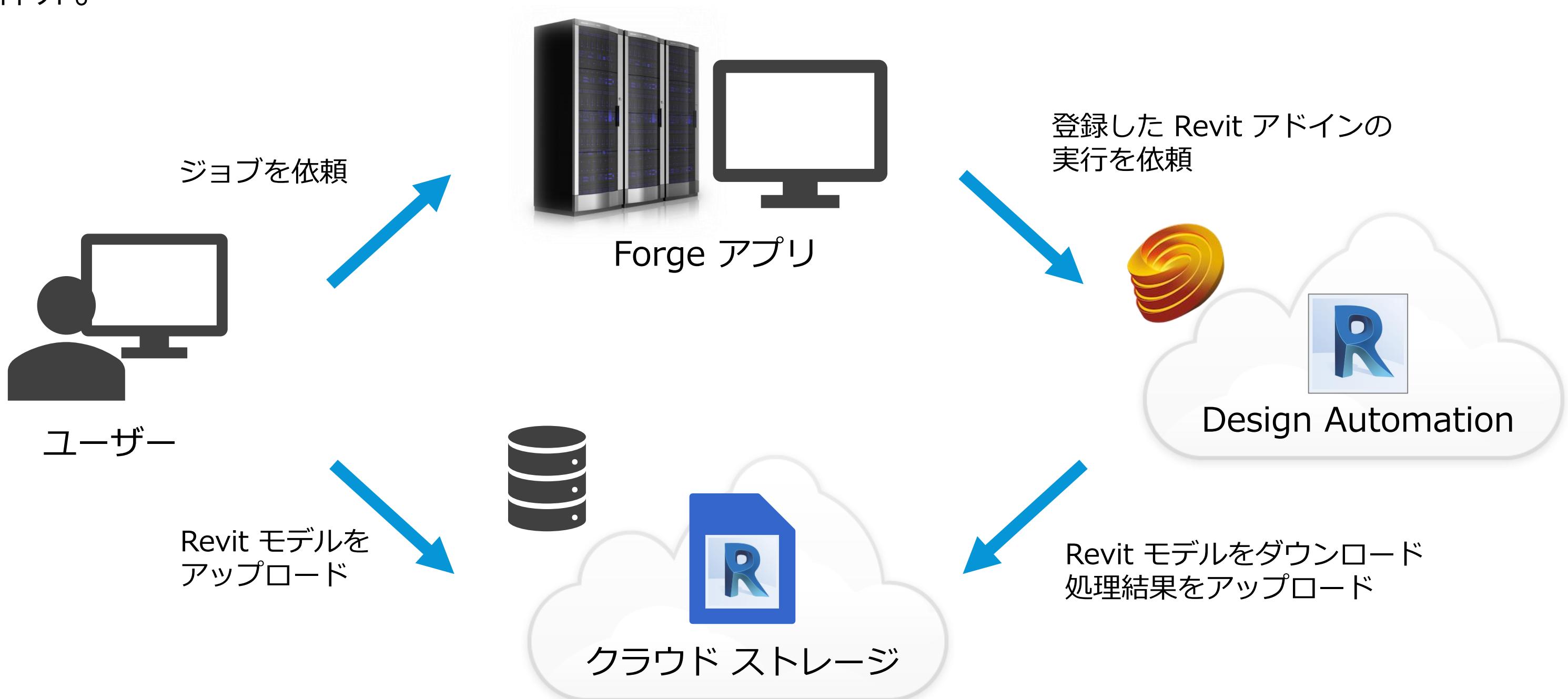
2. Revit アドインをアップロードして登録する Forge アプリケーションを作成

- Forge Platform の “Web API”。HTTP REST API、いわゆる RESTful API
 - “Authentication API”, “Data Management API”, “Design Automation API” など
- 開発言語
 - JavaScript, PHP, C#, Python, JAVA など
- Web サーバー
 - NodeJS, ASP.NET など、それぞれの開発言語に応じて選定
- 開発エディタ
 - 原則自由
- Web ブラウザ
 - WebGL 対応ブラウザ
 - Google Chrome を推奨



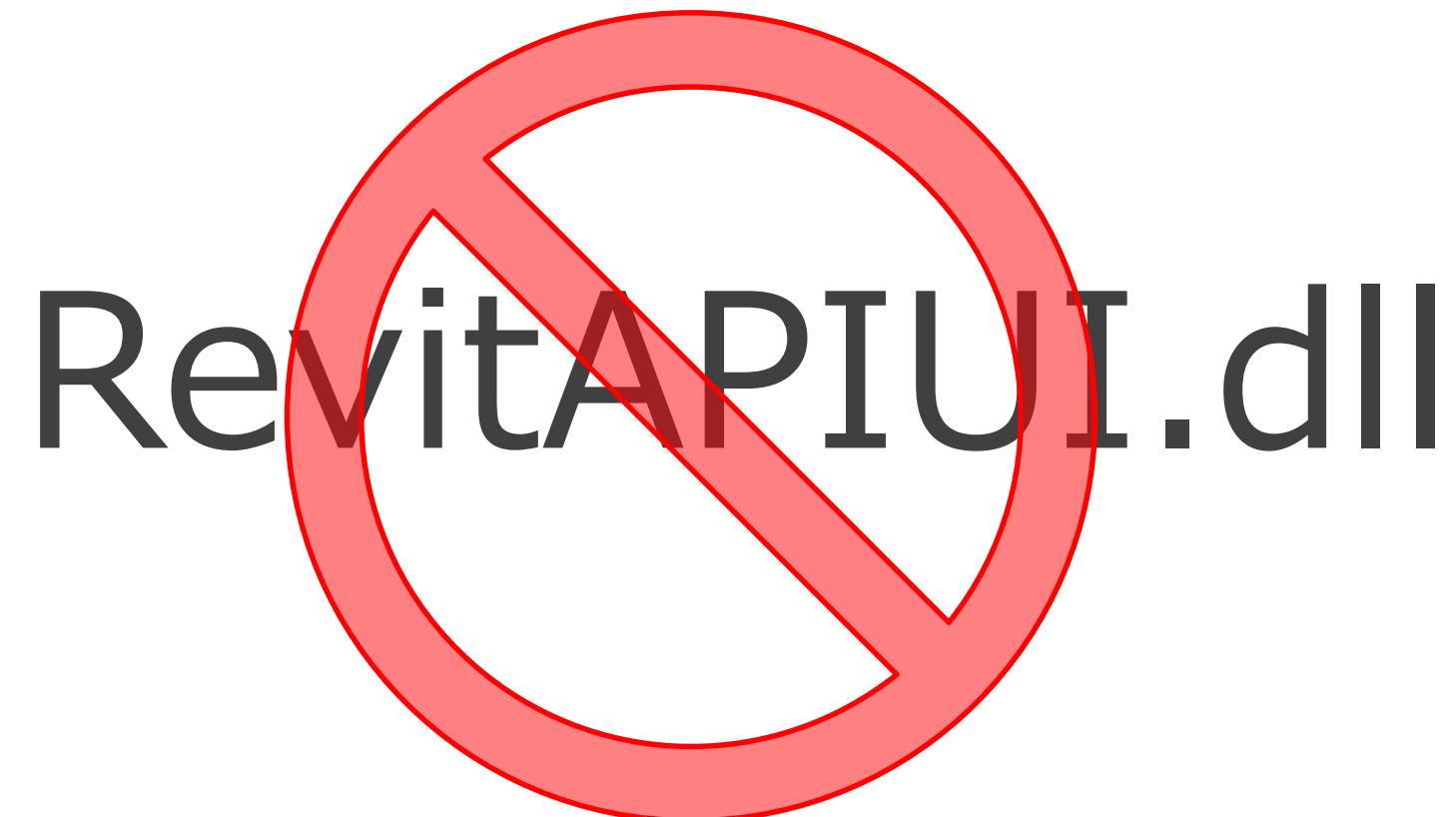
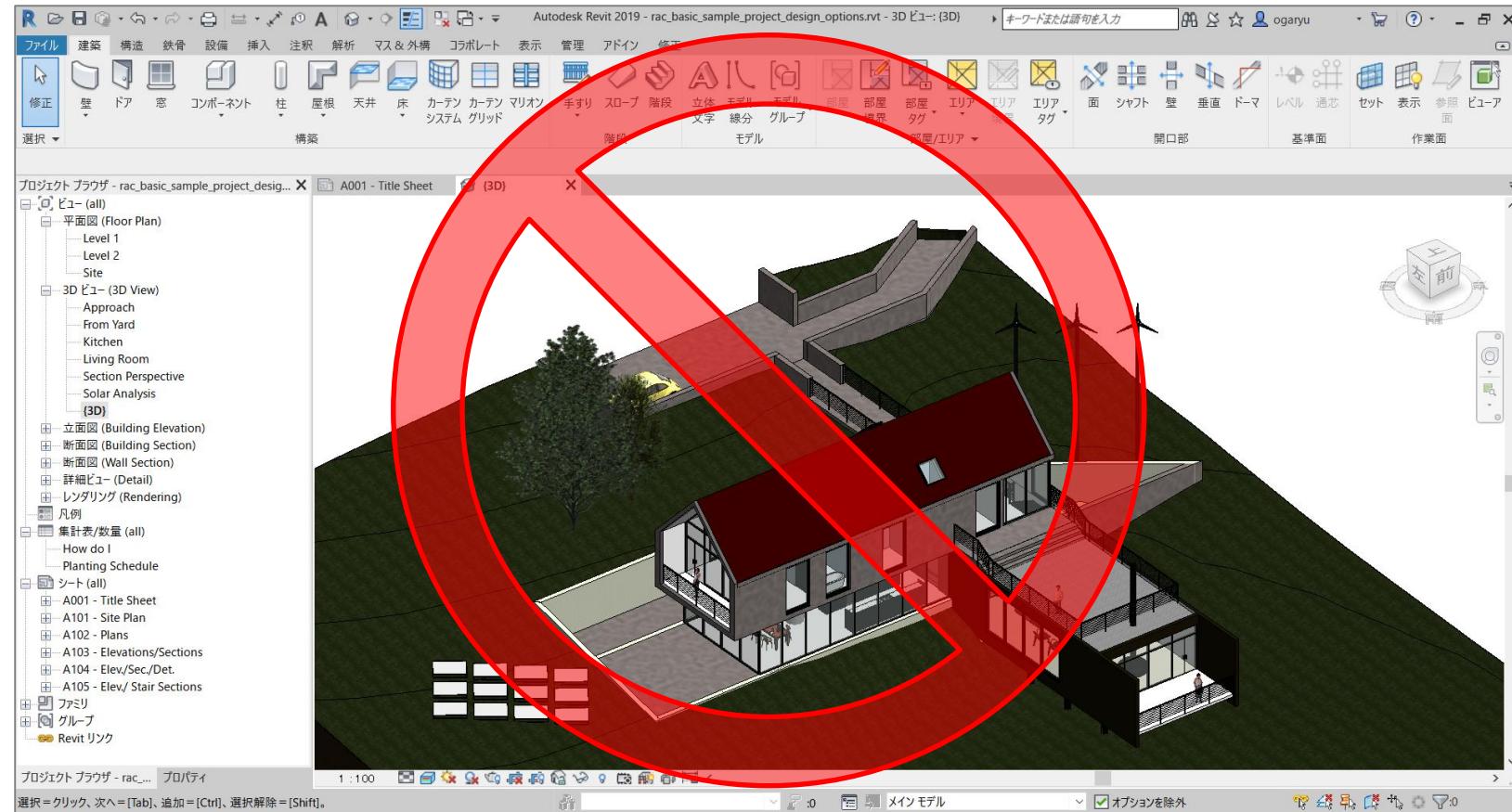
3. Forge アプリケーションから Design Automation にジョブを依頼

- Forge アプリは、Forge Platform の RESTful API を利用して、クライアントと Design Automation を仲介。



Revit アドイン開発時の制約事項

- [x] Revit UI 名前空間へのアクセス、アセンブリ参照はできません。
- [x] REST API で直接 Revit のデータにアクセスできません。
- [x] 複雑なセッション管理は想定されていません。（バッチ処理を想定）
- [x] アドインからネットワークにアクセスすることはできません。
- [x] ユーザーとのインタラクションが発生する処理はサポートされておりません。



Revit アドインのリソースを再利用

- 既存の Revit アドインの開発コード資産を再利用できます。

- ただし、**RevitAPIUI.dll** は参照に含めることはできません。
- Autodesk.Revit.UI 名前空間のクラスやメソッド、プロパティにアクセスしないようにコードを修正してください。
- 最低限必要な名前空間は以下の通りです。

```
using Autodesk.Revit.ApplicationServices;
using Autodesk.Revit.DB;
using DesignAutomationFramework;
```



Design Automation 実行時のデータのセキュリティ

- Design Automation は、クライアント/Forge アプリが指定した任意のクラウドストレージのデータにアクセスして、Forge の一時的なデータ領域（ワーキングディレクトリ）にダウンロードします。
- ユーザー/Forge アプリは、クラウドストレージ上のデータの URL を発行する際に、署名付き URL を作成することで、URL の有効期間を指定したり、アクセストークンによる制御を行うことができます。
 - 署名付き URL の作成方法は、使用するクラウドストレージによって異なります。

Google Cloud Storage の場合

```
StringToSign = HTTP_Verb + "\n" + Content_MD5 + "\n" +
Content_Type + "\n" + Expiration + "\n" +
Canonicalized_Extension_Headers + Canonicalized_Resource
```

- Forge の一時的なデータ領域のデータは、Design Automation の実行後に破棄されます。

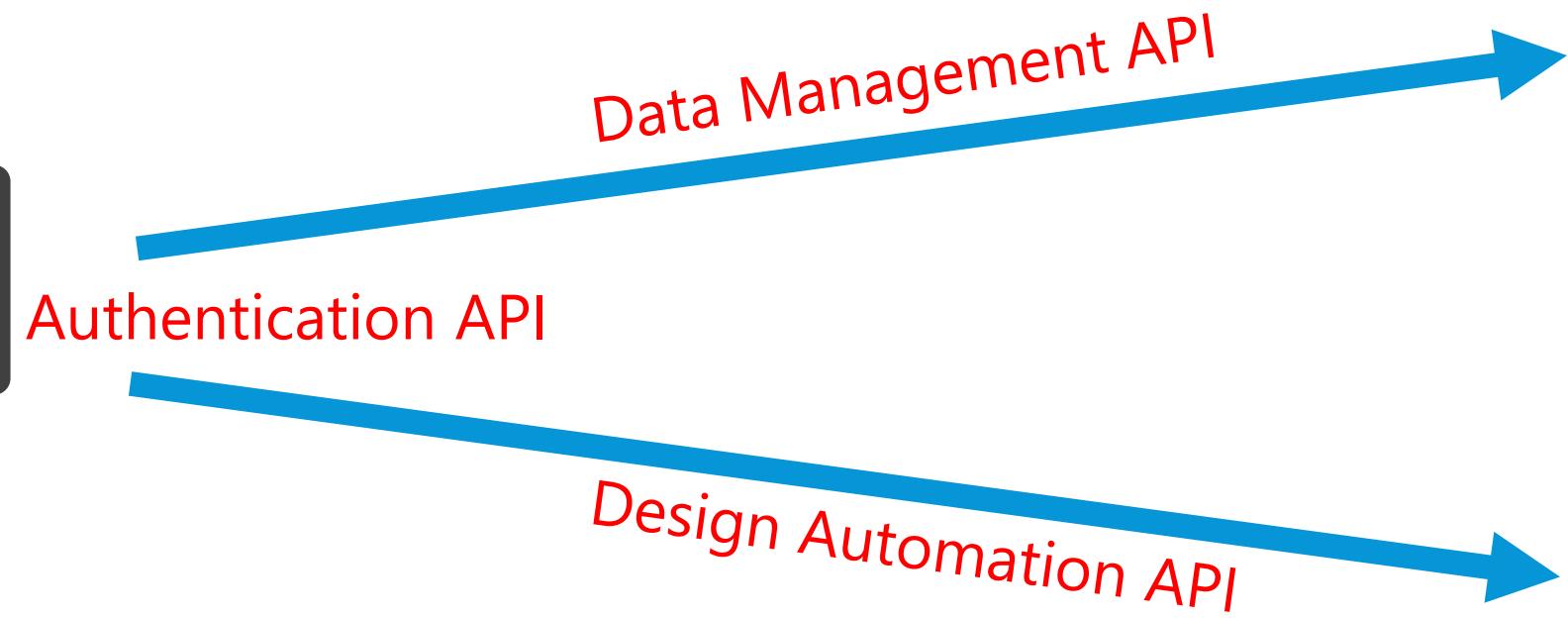
BIM 360 Docs の利用

- クラウドストレージとして BIM 360 Docs を利用。

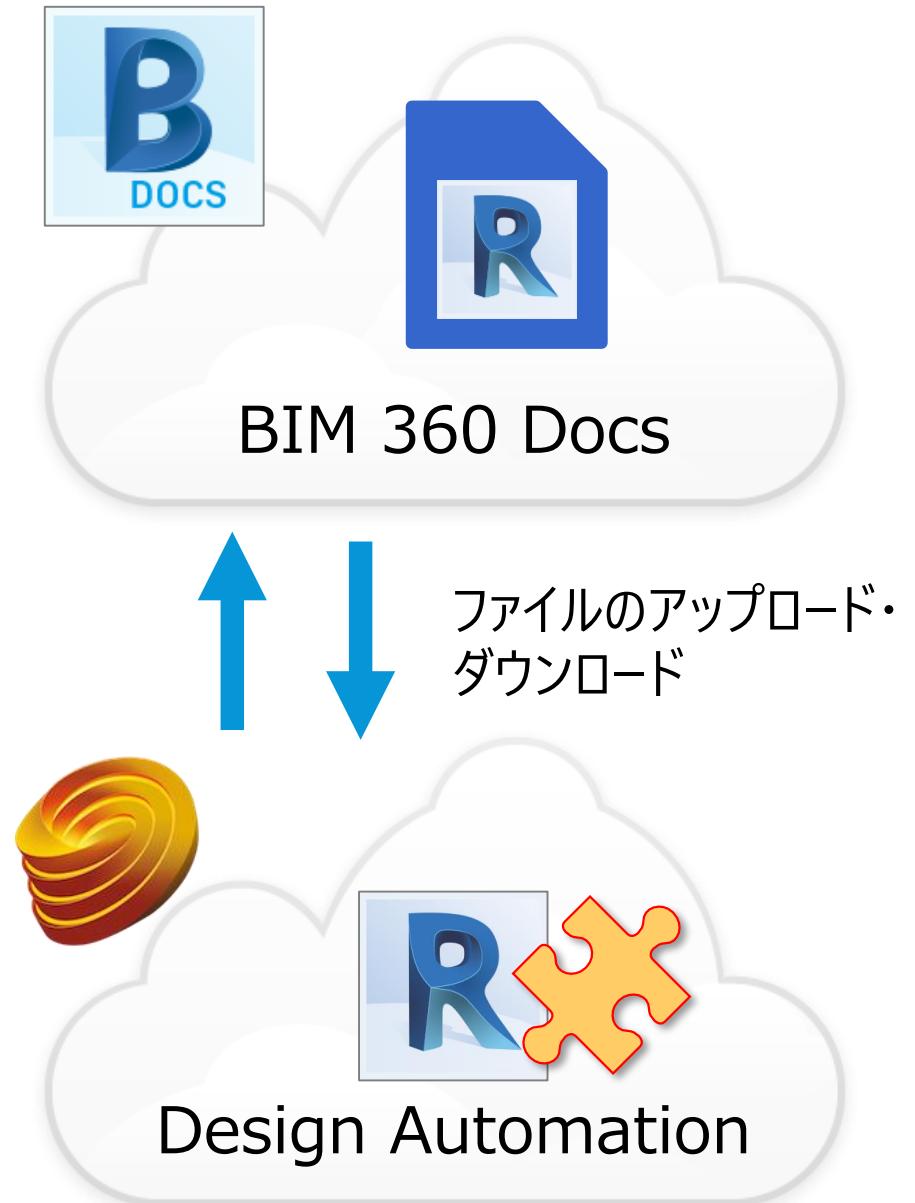
- フォルダ・ファイルのデータにアクセス
- Revit モデルのアップロード・ダウンロード
- Cloud Workshared Revit モデルをパブリッシュ



Forge アプリ

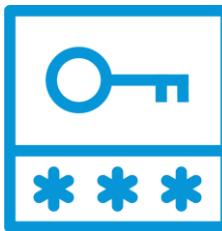


- Revit アドインをアップロード
- ジョブを依頼
- 処理結果を取得



Forge API との連携例

1. 認証・認可



Authentication

2. Revit モデルをアップロード



Data Management

Forge アプリ



5. バージョン追加の通知



Webhooks



BIM 360 Docs



4. 処理結果をアップロード



Design Automation



Model Derivative



Design Automation

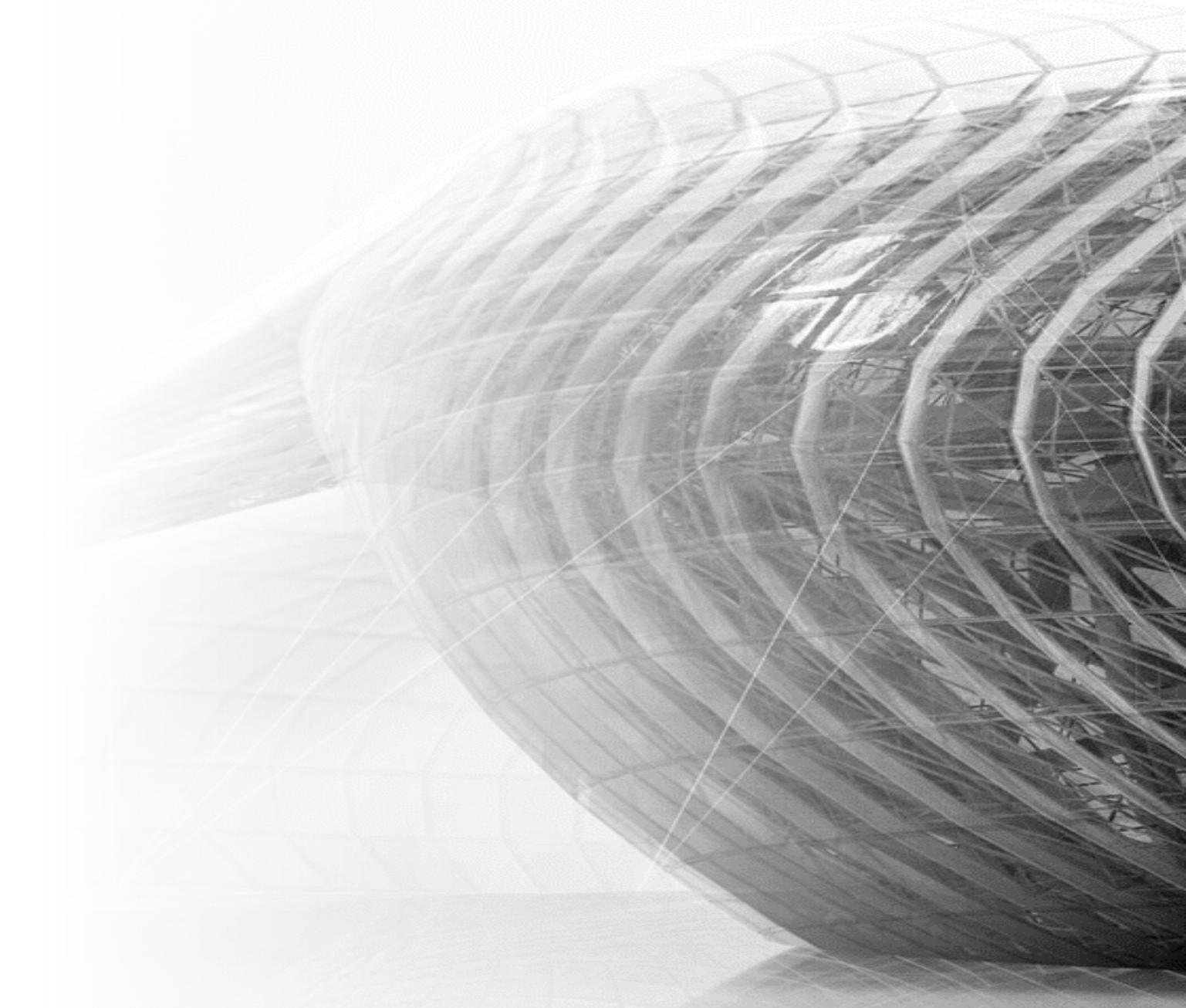
3. Revit アドインのジョブを依頼



Viewer

7. Revit モデルを表示

開発の流れ

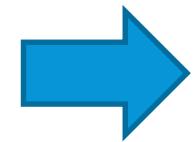


開発の流れ 1: アドインの変換とバンドルパッケージの作成



Revit アドインの変換

- ① .NET C# or VB で実装したアドインプロジェクトを用意
- ② DesignAutomationBridge.dll を参照に追加
※ RevitAPIUI.dll は参照を削除
- ③ IExternalApplication または、
IExternalCommand を
IExternalDBApplication に変更
- ④ エラーハンドリングを実装
- ⑤ ローカルの Revit で動作確認



バンドルパッケージの作成

.bundle フォルダ構成のパッケージを ZIP ファイルに圧縮

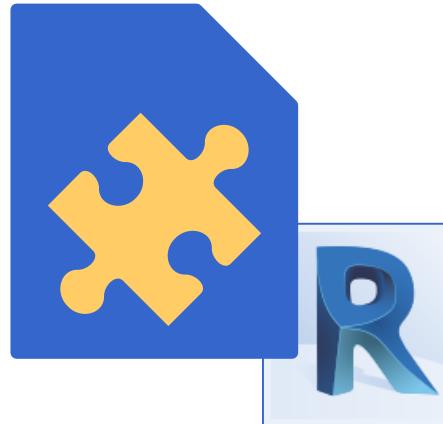
```
DeleteWallsApp.zip
|-- DeleteWalls.bundle
|   |-- PackageContents.xml
|   |-- Contents
|       |-- DeleteWalls.dll
|       |-- DeleteWalls.addin
```

IExternalDBApplication の実装

```
1  using System;
2  using System.IO;
3  using Autodesk.Revit.ApplicationServices;
4  using Autodesk.Revit.DB;
5  using DesignAutomationFramework;
6
7  namespace DeleteWalls
8  {
9      [Autodesk.Revit.Attributes.Regeneration(Autodesk.Revit.Attributes.RegenerationOption.Manual)]
10     [Autodesk.Revit.Attributes.Transaction(Autodesk.Revit.Attributes.TransactionMode.Manual)]
11     public class DeleteWallsApp : IExternalDBApplication
12     {
13         public ExternalDBApplicationResult OnStartup(Autodesk.Revit.ApplicationServices.ControlledApplication app)
14         {
15             DesignAutomationBridge.DesignAutomationReadyEvent += HandleDesignAutomationReadyEvent; イベントハンドラの登録
16             return ExternalDBApplicationResult.Succeeded;
17         }
18
19         public ExternalDBApplicationResult OnShutdown(Autodesk.Revit.ApplicationServices.ControlledApplication app)
20         {
21             return ExternalDBApplicationResult.Succeeded;
22         }
23
24         public void HandleDesignAutomationReadyEvent(object sender, DesignAutomationEventArgs e)
25         {
26             e.Succeeded = true;
27             DeleteAllWalls(e.DesignAutomationData); ← カスタム処理を呼び出し
28         }
29     }
30 }
```

用語の解説

AppBundle

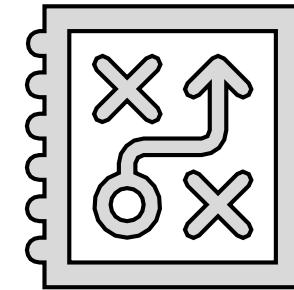


Revit アドインのバイナリパッケージ

Id : CustomApp
Engine : 2019
Description: Test custom app
Package: Storage URL

Revit の .NET API で作成したアセンブリや関連ファイルを ZIP 圧縮してアップロードし、AppBundle として登録する。

Activity

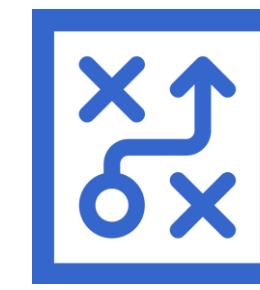


実行されるアクションの定義

Id: UpdateParamActivity
Input Parameter : RVT, TXT
Output Parameter: RVT
AppBundle: CustomApp

カスタム処理の雛型を定義する。
.NET アセンブリ内でどんなデータを入力して、どんなデータを出力するか定義する。

WorkItem



指定のアクションを呼び出すジョブ

Id: 返却される文字列
Activity : UpdateParamActivity
Input Parameter : File URL
Output Parameter: Storage URL

REST API でリクエストするジョブ。
対象のモデルやテキストデータ、出力先の URL と、実行する Activity を指定する。

開発の流れ 2: AppBundle と Activity の登録

AppBundle を登録

1. 新しい AppBundle を作成
2. バンドルパッケージをアップロード
3. エイリアスとバージョンを設定

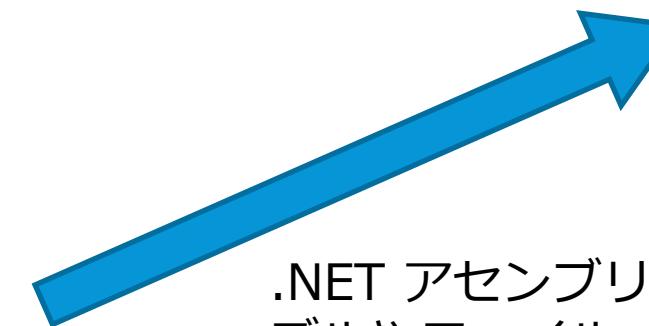


Revit エンジンのバージョン
AppBundle の名前



Activity を登録

1. 新しい Activity を作成
2. エイリアスとバージョンを設定
3. 必要があればロケールを設定



Design
Automation

.NET アセンブリ内で参照するモ
デルやファイル（入出力）や、パ
ラメータなどの雛型を定義

用語の解説

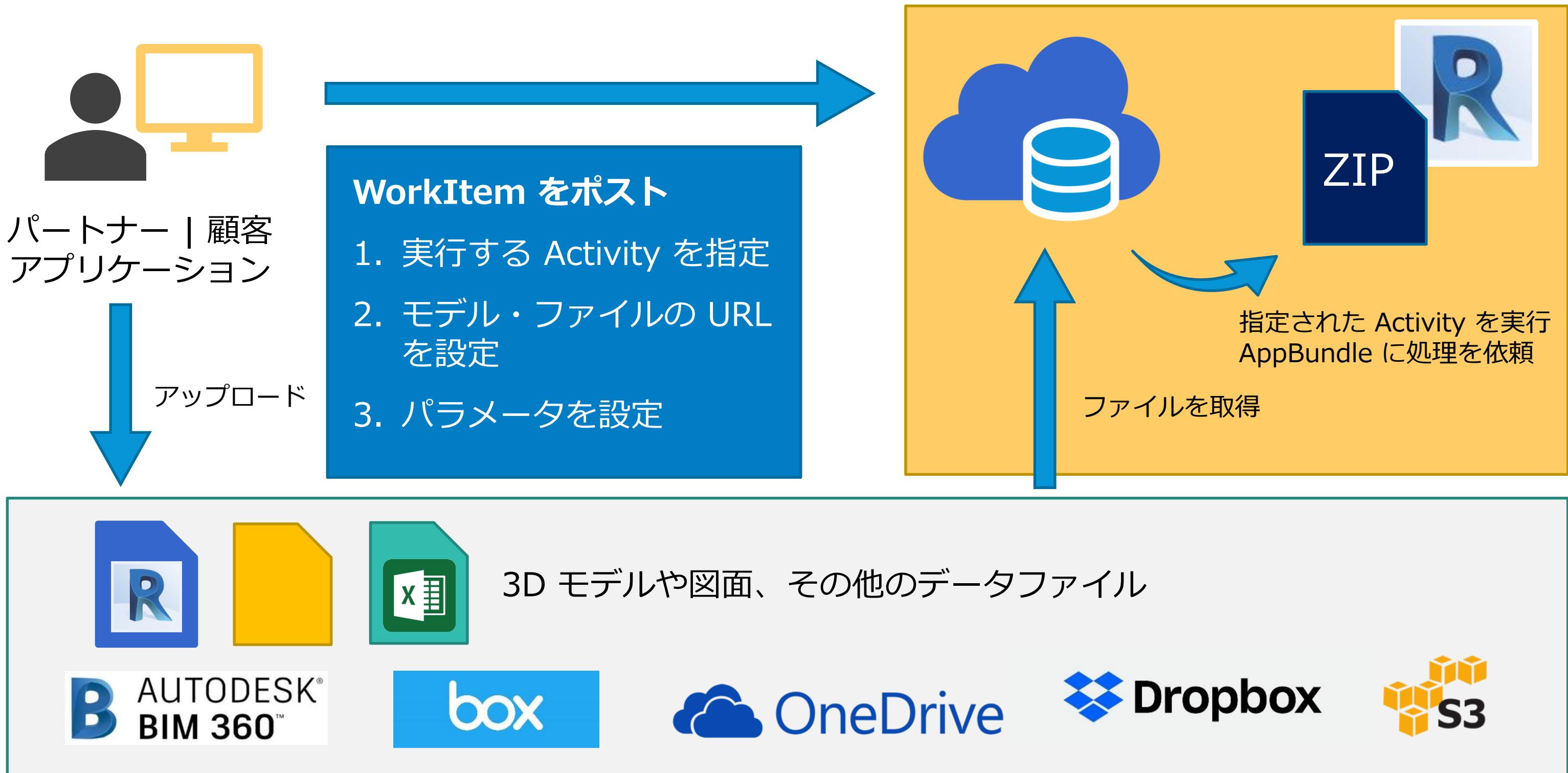
- AppBundle と Activity は、それぞれバージョンを追加していくことができます。
- 特定のバージョンに任意のラベルで名前をつけてエイリアスを作成できます。

AppBundle	Version	Alias
	1	
	2	Production
	3	
	4	Test
	5	Development

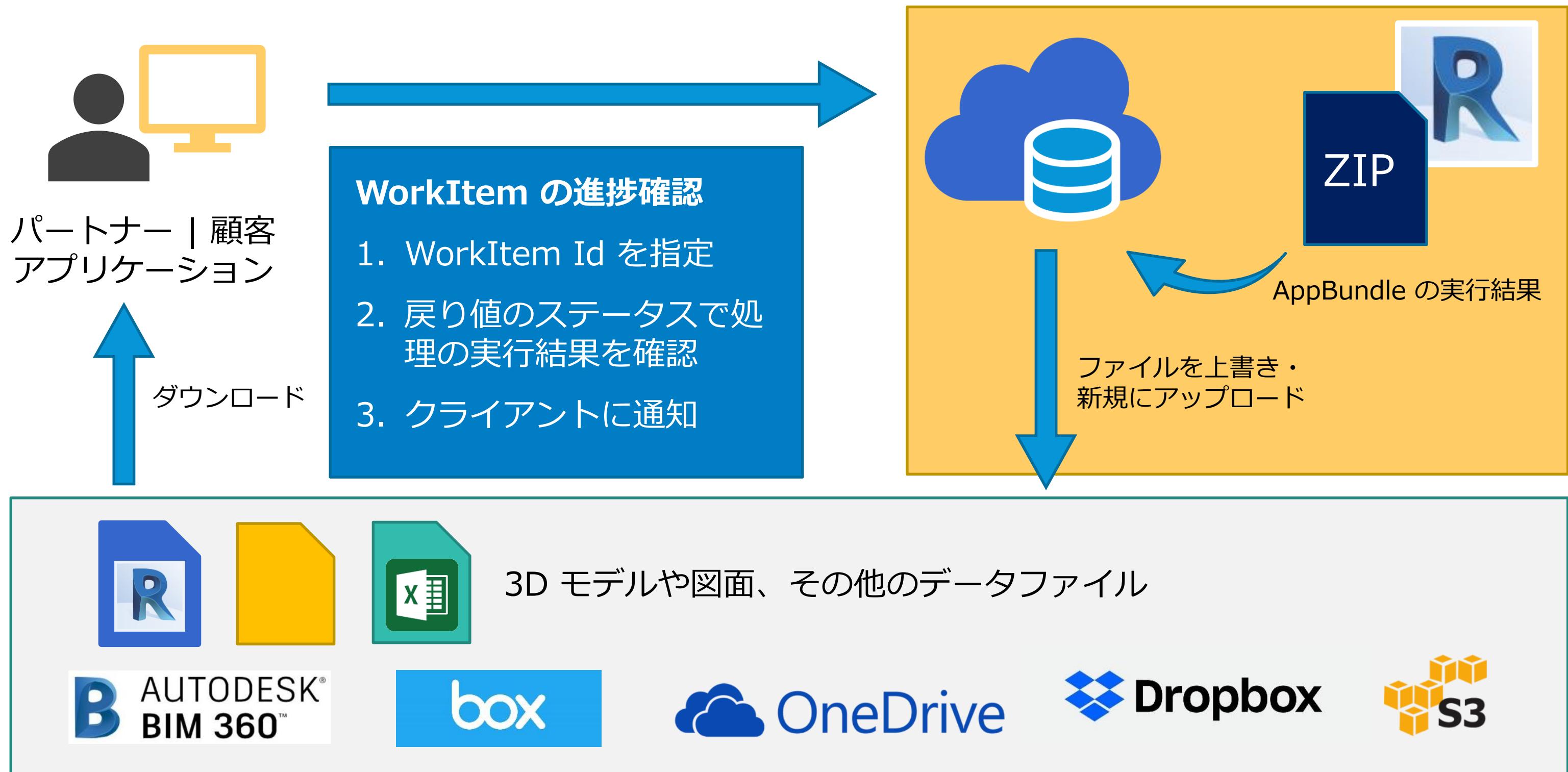
例えば、Activity のテストを行う際に、Test エイリアスを作成してテストを実行する。ただし、AppBundle は、稼働中の Production のエイリアスを呼びだす、といった開発・テスト用の使い方ができます。

Activity	Version	Alias
	1	
	2	
	3	
	4	
	5	Production
	6	
	7	Test
	8	Development

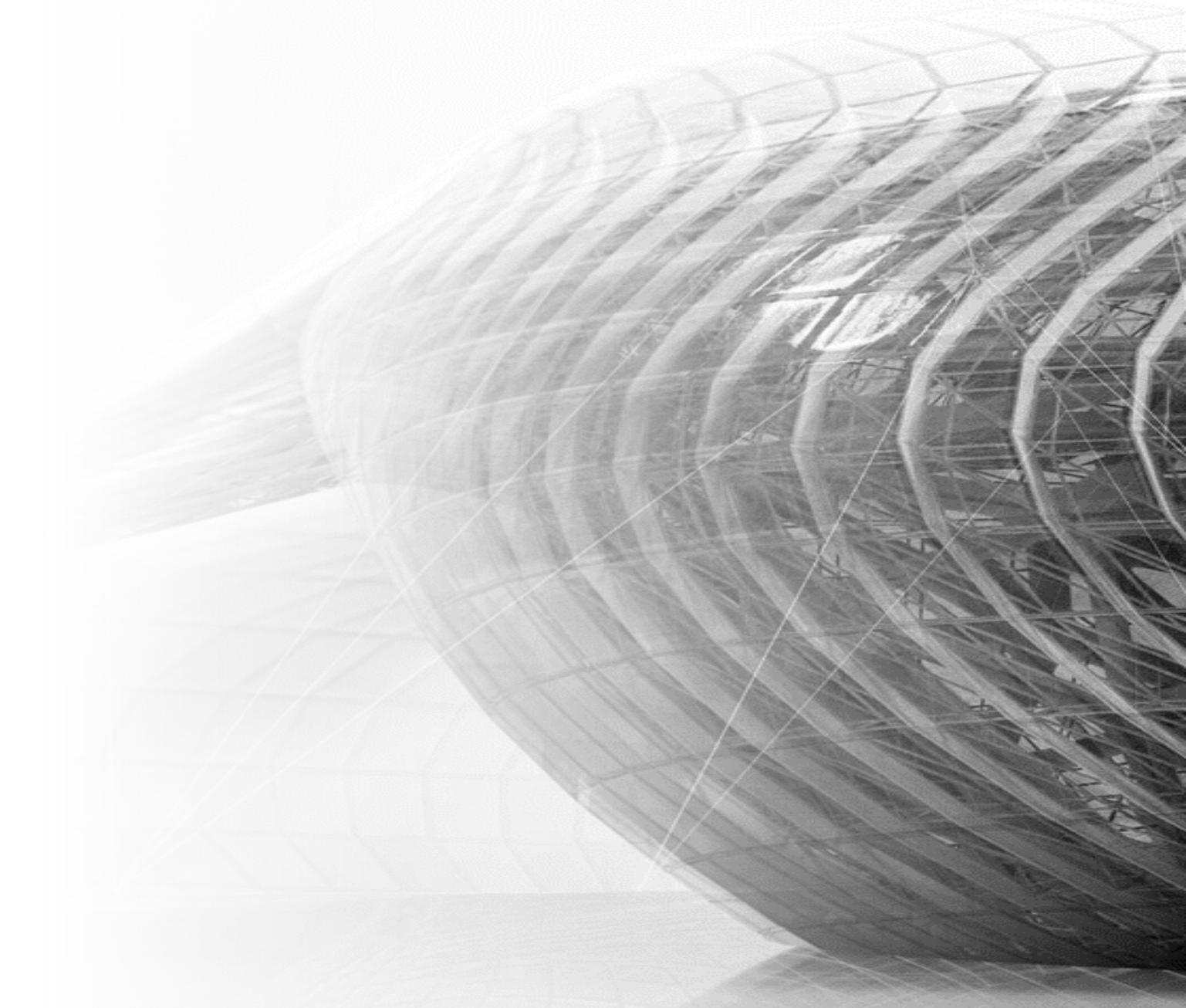
開発の流れ 3: WorkItem をポストする



開発の流れ 4: WorkItem の進捗チェック



デモ動画



Forge DevCon 2018 セッション

- Manage Revit Digital Content using Forge Design Automation API for Revit.
 - Marc Durand, Digital Disruption Director, WS Atkins & Partners Overseas
 - Mustafa Salaheldin, Data Science Manager, WS Atkins & Partners Overseas

The slide features a blue and white abstract background graphic at the bottom. At the top, the title 'Managing Digital Content with Design Automation API v3 for Revit' is displayed in large blue text. Below the title, two speakers are listed: Mustafa Salaheldin (Data Science Manager) and Marc Durand (Digital Disruption Director). In the bottom left corner, the Autodesk University logo is present.

Managing Digital Content with
Design Automation API v3 for Revit

Mustafa Salaheldin
Data Science Manager

Marc Durand
Digital Disruption Director

AUTODESK®
UNIVERSITY

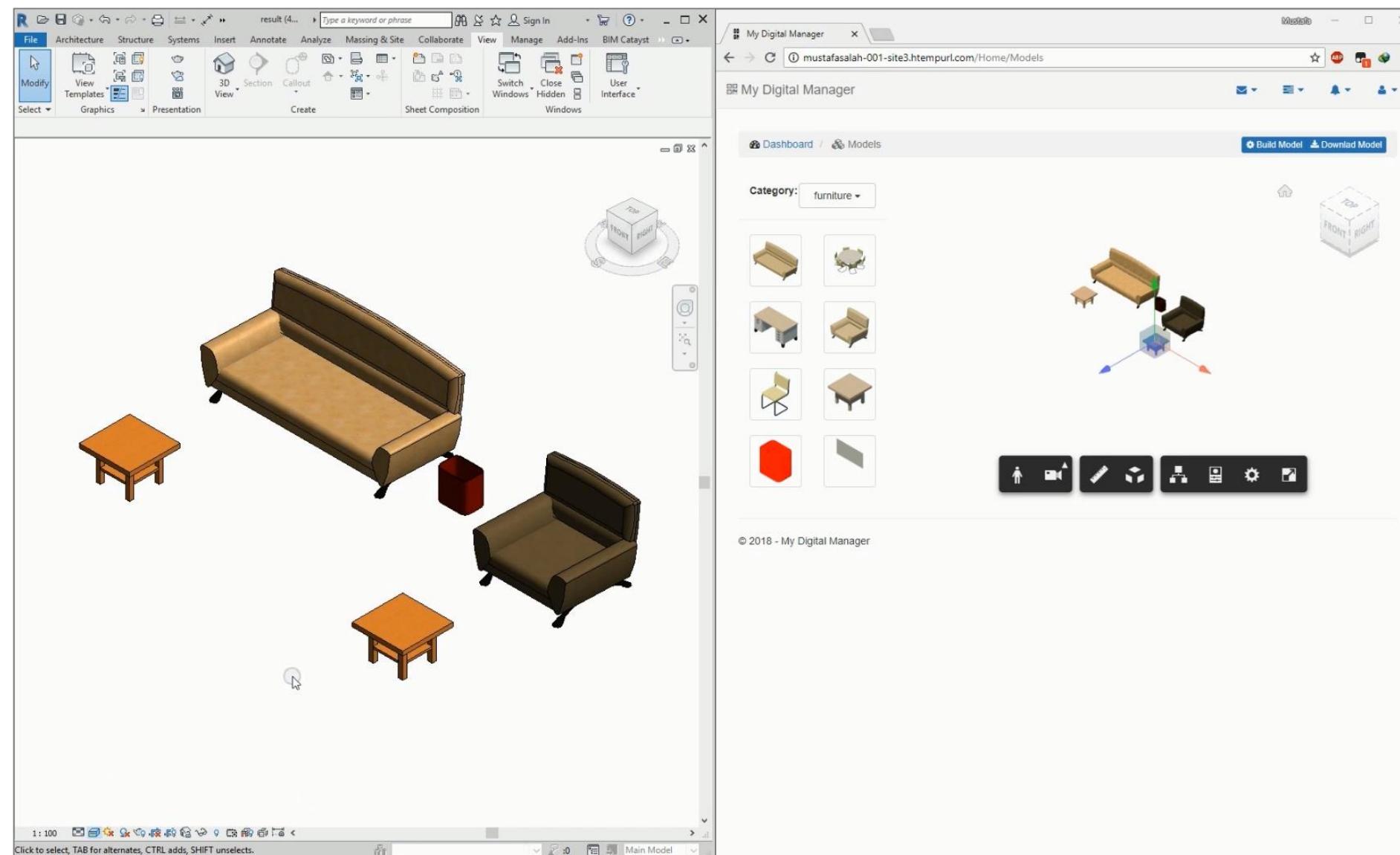
ATKINS

Member of the SNC-Lavalin Group

<https://www.youtube.com/watch?v=WpCm4o2Ynj0&list=PLyobA0wOWp-qhdSSbIroQaf8ZFSXEVF2K>

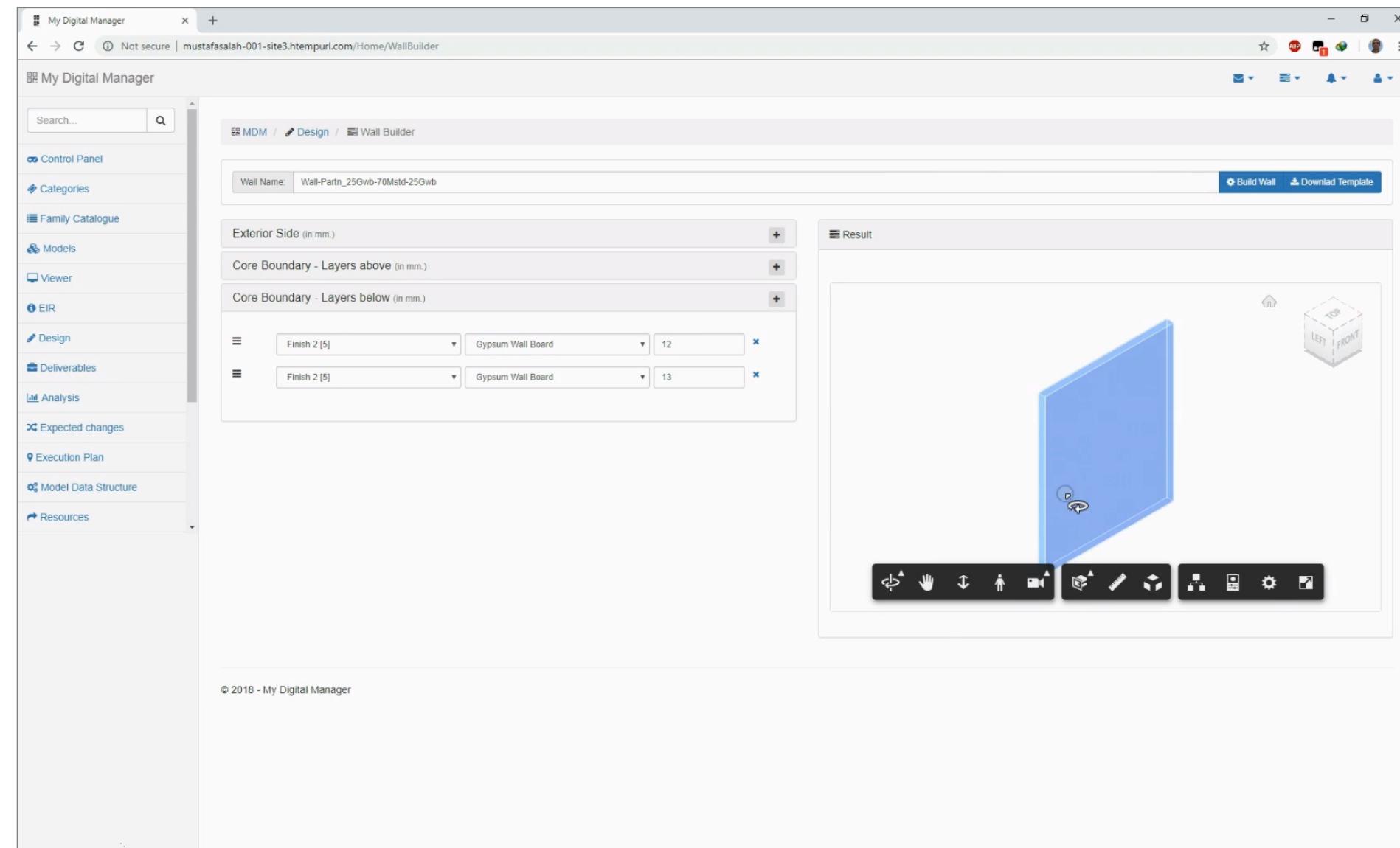
ロード可能なファミリをプロジェクト上に配置してレイアウト調整

- 企業で利用しているファミリを Revit プロジェクトに配置してダウンロードするサンプル。
- Forge Viewer で新規プロジェクトのビューを開いて、ドラッグ & ドロップでファミリを配置できます。
- ファミリの位置や方向を操作して、レイアウトを作成します。



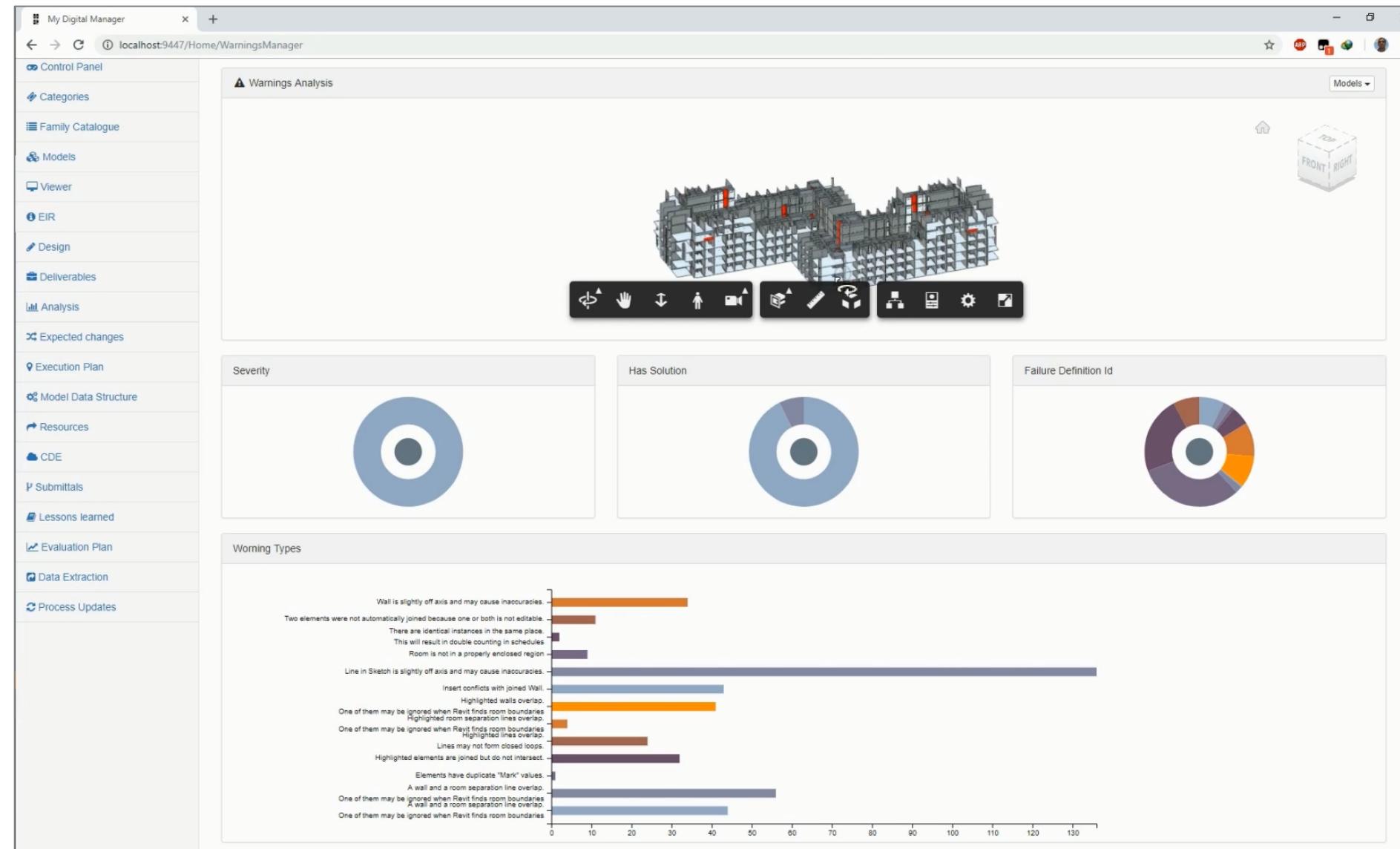
システムファミリの新規ファミリタイプを作成

- 新規にシステムファミリのタイプを作成して、Revit プロジェクトに配置してダウンロードするサンプル。
- 壁、床、天井、屋根をサポート。
- デモでは壁の複層構造の機能、マテリアル、厚さ、順序などをそれぞれ指定して壁タイプを作成。



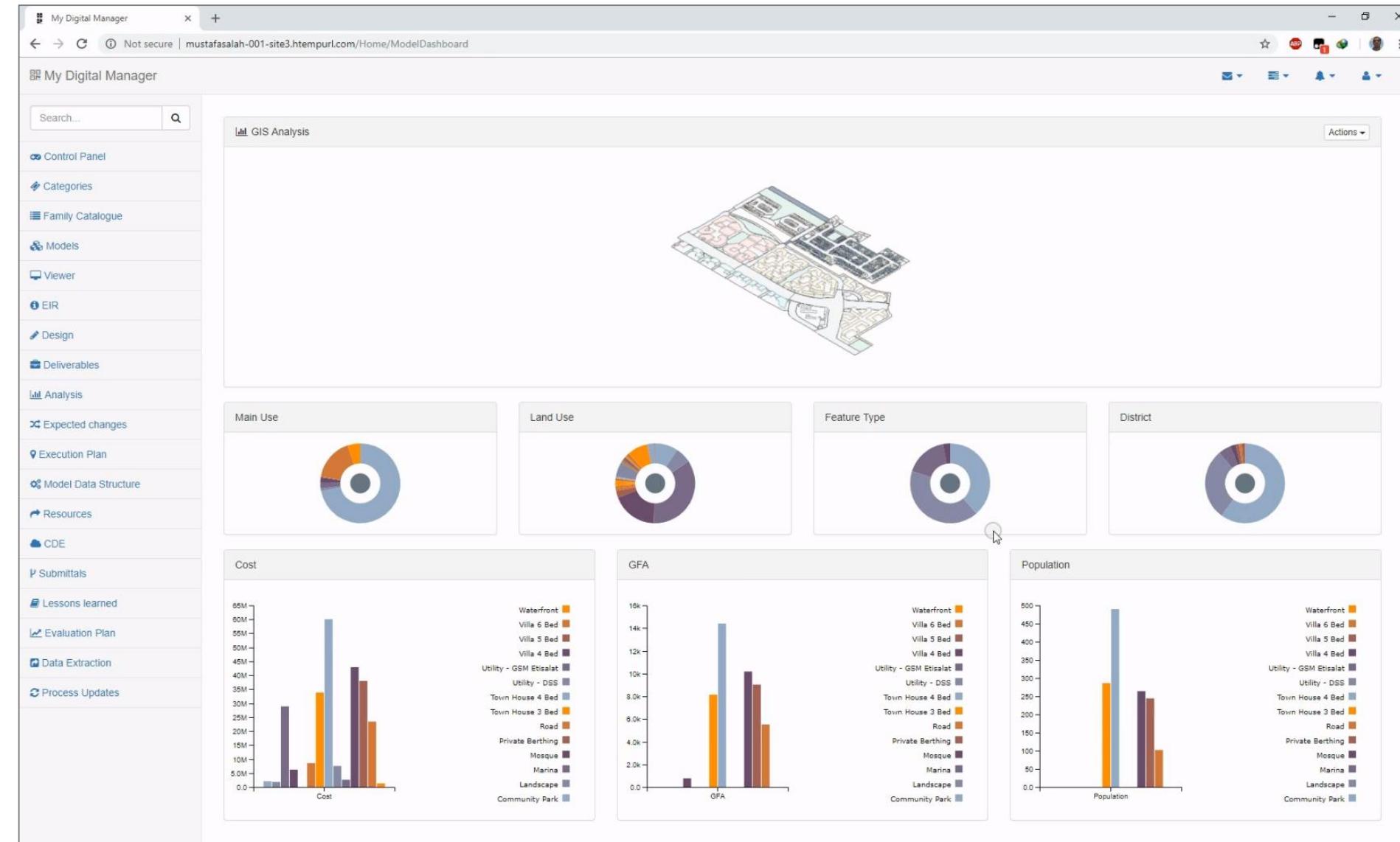
Revit モデルの警告・エラーを取得してヘルスダッシュボードに表示

- Revit モデルで発生している警告・エラーをチェックして、その集計結果を様々なタイプのグラフで表示。
- 機械学習プログラムが、その問題の原因を解析し、解決方法のサジェストionsを一覧表示。



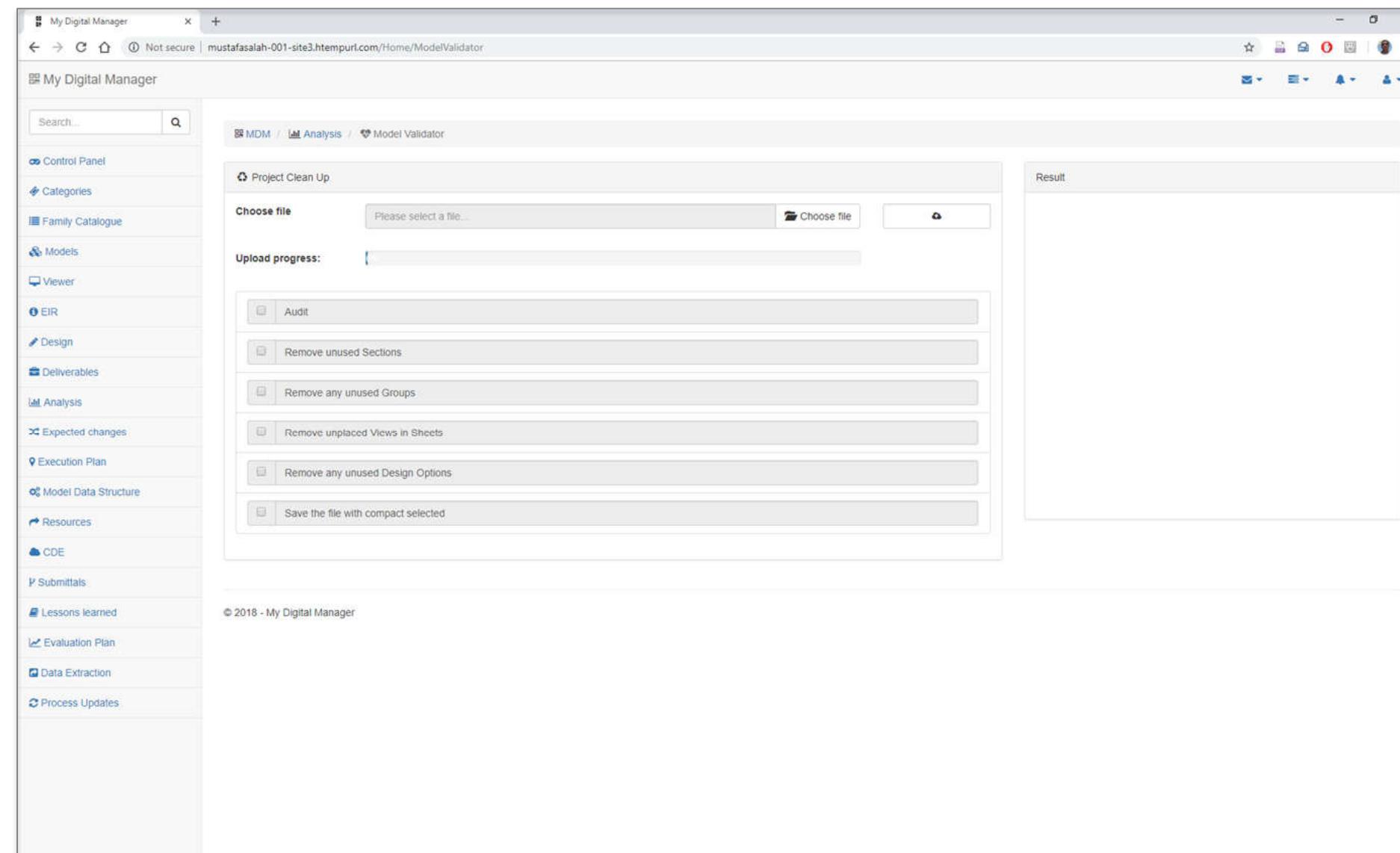
Revit モデルのデータを抽出してインサイトダッシュボードに表示

- マスター・プランのプロジェクトで、GIS データの解析結果をビジュアライズして意思決定に活用。
- Design Automation API で Revit モデルからデータを抽出して、インサイトダッシュボードを構築。

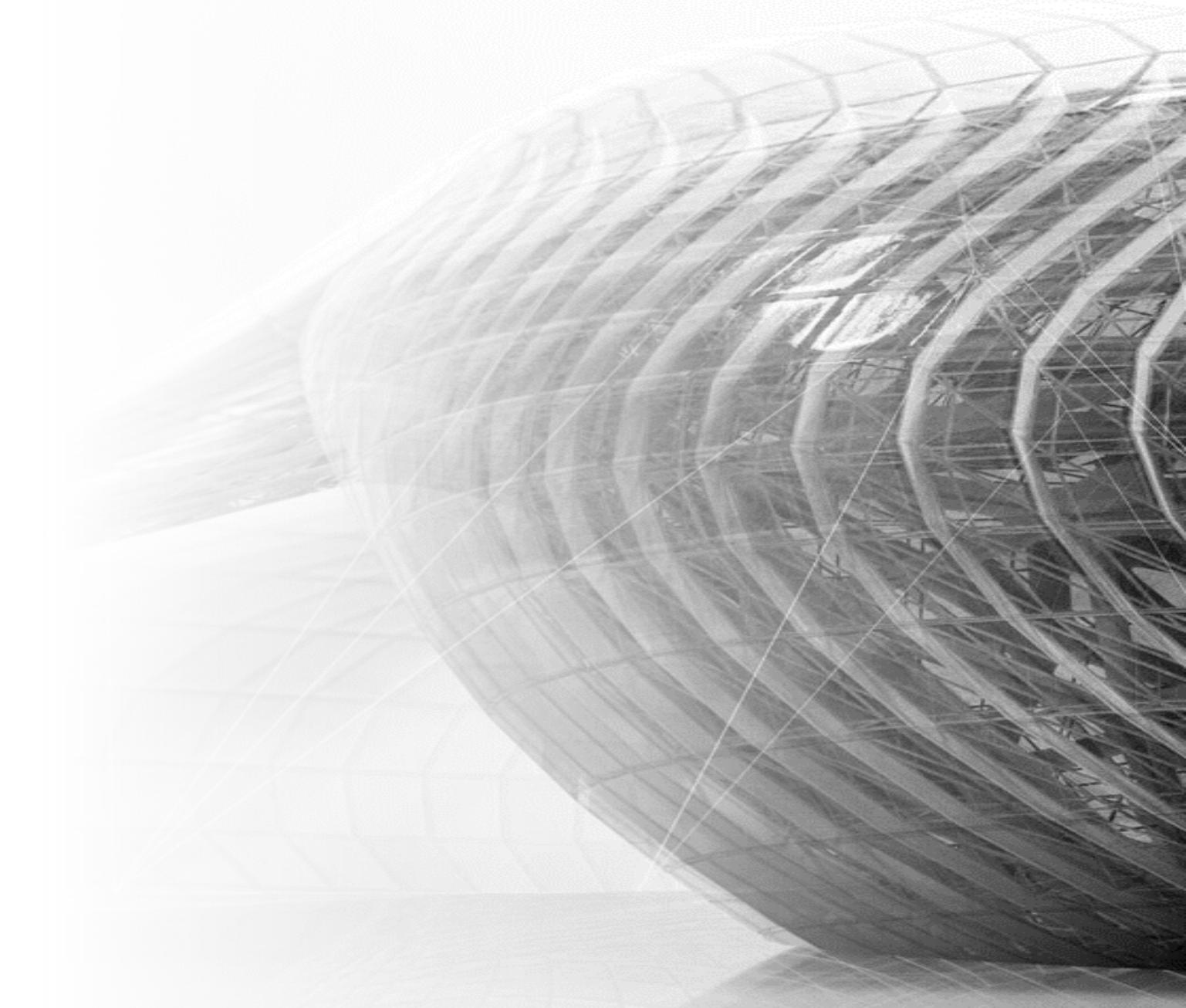


デリバリ後の Revit モデルのメンテナンス

- プロジェクトのデリバリ後、プロジェクトの整合性や堅牢性を維持するために、企業の基準を満たしているかチェックするアプリケーション。
- 評価基準のチェックリストに応じて、未使用のビューやシート、未配置の部屋やグループを削除して保存。



補足事項



Revit アドインのエラーハンドリング

- Revit アドインで発生するエラーは、トランザクションのコミット時、あるいはロールバック時に呼び出されます。
- DesignAutomationBridge は、デフォルトのエラーハンドラを搭載しており、すべての警告を抑止し、エラーの場合は、それを解決しようと試みます。解決したらコミットします。
- デフォルトの解決方法が「要素の削除」となる場合は、エラーハンドラは要素を削除せずに、ロールバックします。
- 開発者は、カスタムのエラーハンドラを実装して、デフォルトのエラーハンドラを上書きすることができます。

```
public void HandleDesignAutomationReadyEvent(object sender, DesignAutomationReadyEventArgs e)
{
    // Hook up the CustomFailureHandling failure processor.
    Application.RegisterFailuresProcessor(new CustomFailureHandlingProcessor());

    // Run the application logic.
    SketchItFunc(e.DesignAutomationData);

    e.Succeeded = true;
}
```

サポートされている入力の引数: JSON データの埋め込み

```
"parameters": {  
    "rvtFile": {  
        "zip": false,  
        "ondemand": false,  
        "verb": "get",  
        "description": "Input Revit model",  
        "required": true,  
        "localName": "$(rvtFile)"  
    },  
    "countItParams": {  
        "zip": false,  
        "ondemand": false,  
        "verb": "get",  
        "description": "CountIt parameters",  
        "required": false,  
        "localName": "CountItParams.json"  
    },  
    "result": {  
        "verb": "put",  
        "url": "https://myWebsite/signed/url/to/result.txt"  
    }  
}
```

Activity

```
"arguments": {  
    "rvtFile": {  
        "url": "https://myWebsite/CountIt.rvt"  
    },  
    "countItParams": {  
        "url": "data:application/json, {  
            'walls': false,  
            'floors': true,  
            'doors': true,  
            'windows': true}"  
    },  
    "result": {  
        "verb": "put",  
        "url": "https://myWebsite/signed/url/to/result.txt"  
    }  
}
```

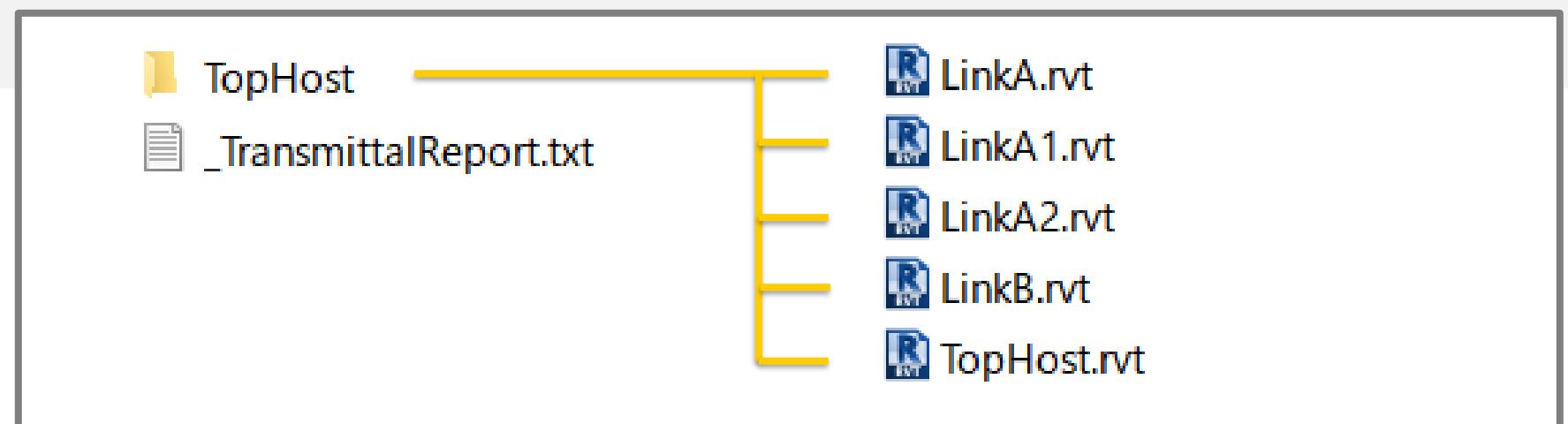
WorkItem

JSONデータをリクエストのパラメータに埋め込むと、
Revit アドインから JSON ファイルとして取得可能。

サポートされている入力の引数: eTransmit ファイル

- eTransmit for Revit で出力した ZIP ファイルをサポート。

```
"arguments": {  
    "rvtFile": {  
        "url": "https://s3-us-west-2.amazonaws.com/revitio-dev/test-data/TopHost.zip"  
    },  
    "countItParams": {  
        "url": "data:application/json,{\"walls\": true, \"floors\": true, \"doors\": true, \"windows\": true}"  
    },  
    "result": {  
        "verb": "put",  
        "url": "https://myWebsite/signed/url/to/result"  
    }  
}
```



サポートされている入力の引数: Revit リンクモデル



※ サブフォルダのリンクモデルもサポート

サポートされている入力の引数: ZIP ファイル

- アップロード速度を向上させるために、Revit モデルを ZIP 圧縮して送信することができます。
- “pathInZip” オプションで、メインの Revit モデルのパスを指定します。

```
"arguments": {  
  "rvtFile": {  
    "url": "https://path/to/zip/file.zip",  
    "pathInZip": "RevitFile.rvt"  
  },  
}
```

ログ出力

- WorkItem の進捗リクエストのレスポンスには、**report.txt** の URL が含まれます。
- Revit アドインのプログラム内で、**Console.WriteLine()** メソッドを呼び出せば、任意のタイミングで、report.txt にカスタムのログを出力することができます。

```
Starting work item 8829b13e8a704450a8e5fee5b085bcd8
Start download phase.
Start preparing AppPackage SharedParameter_ApplyParameter.
Start downloading file
// 省略
End download phase.
// 省略
===== Revit is running: revitcoreconsole =====
Current Exe path:
T:\Aces\AcesRoot\19.0\coreEngine\Exe\revitcoreconsole.exe
// 省略
Initializing RevitCoreEngine...
Selected Revit\RCE install Path: (from app.config) C:\Revit2019
Resolving location of Revit/RevitCoreEngine installation...
Running user application....
Found an addIn for registration: SharedParameter_ApplyParameter.addin
Initializing RCE....
Initializing RevitCoreEngine (and possibly running stub installer)...
Language not specified, using English as default
// 省略
Finished running. Process will return: Success
===== Revit finished running: revitcoreconsole =====
Process exit code: 0
End Revit Core Engine standard output dump.
End script phase.
Start upload phase.
// 省略
End upload phase.
Job finished with result Succeeded
```

Revit ローカライズ版のエンジンを指定可能

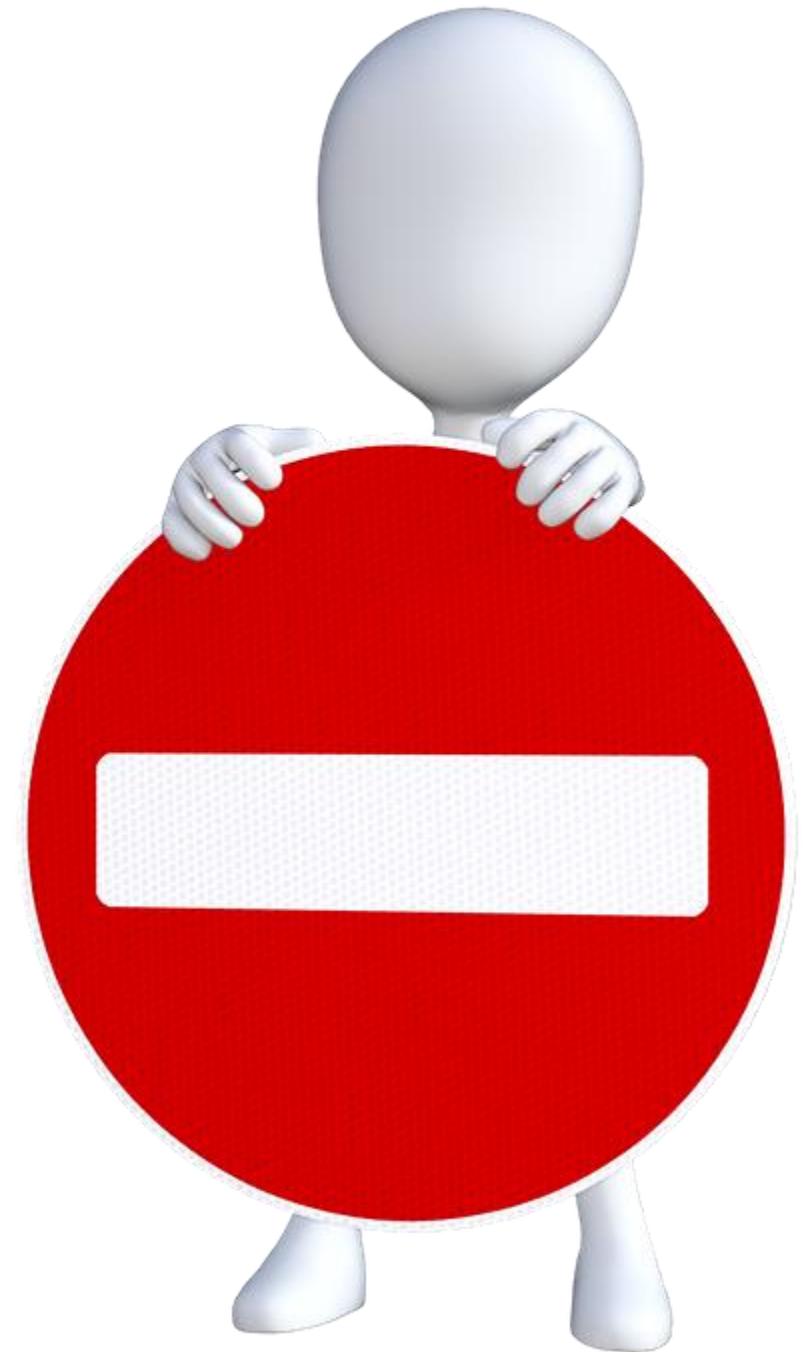
- 起動する Revit のエンジンを、英語以外の言語も指定することができます。
 - /I JPN を指定。
- 日本語版で起動すれば、すべての Revit の出力も日本語版で実行されます。
- 日本語の名称を含んだビルトインの要素やタイプをサポートする場合に利用します。



```
{  
  "commandLine": [  
    "$(engine.path)¥¥¥revitcoreconsole.exe /i $(args[rvtFile].path) /al $(apps[DeleteWallsApp].path) /I DEU"  
  ],  
  ...  
}
```

その他の制約事項

- [x] プリント出力・PDF 出力
- [x] IFC のインポート・エクスポート
- [x] Navisworks ファイルのエクスポート
- [x] Desktop Connector 未サポート
- RFA ファイルは、Model Derivatives API で
未サポートのため、Viewer でも表示できません。





AUTODESK®
FORGE

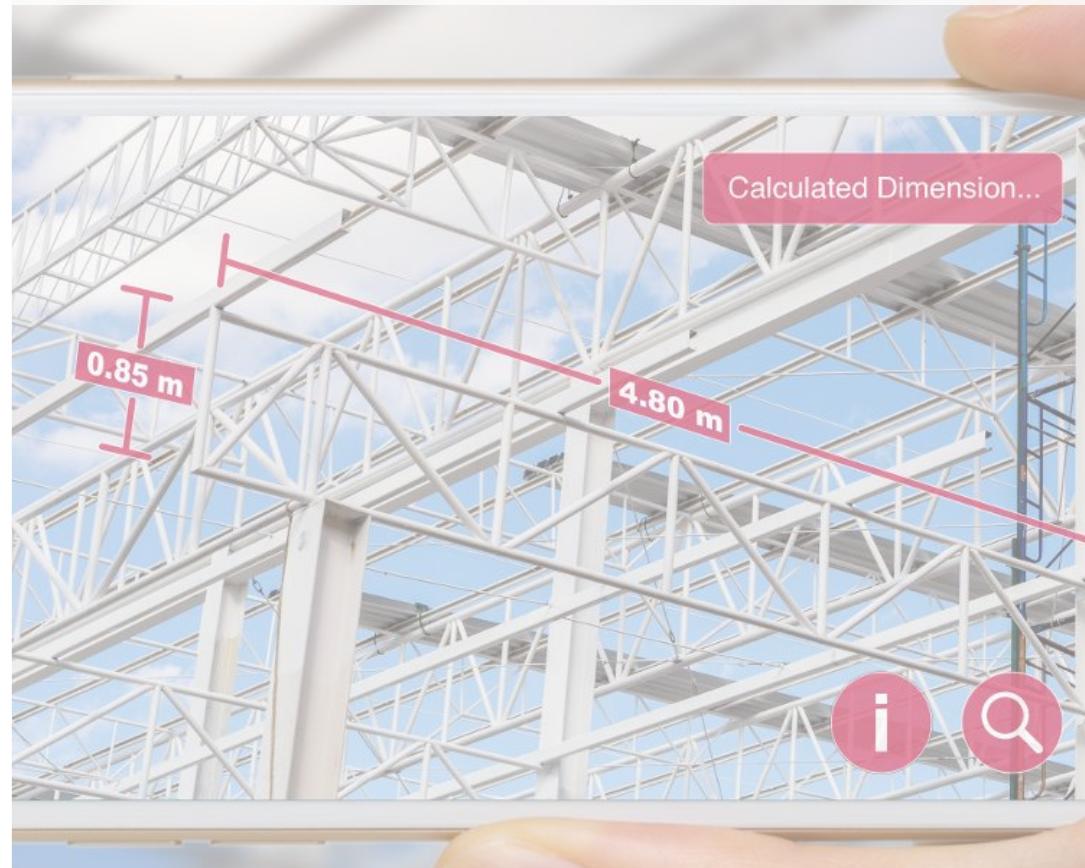
Design Automation API for Revit

クラウドサービスで Revit データを活用



作成

- カスタムコンテンツの生成
- モデル作成の自動化



抽出

- モデルデータの検索・分析
- レポートの自動生成



編集

- 企業の基準を維持
- 設計図書の自動生成

Forge DevCon 2018 セッション動画とマテリアル

- Revit Data on Forge – How Can the Design Automation for Revit API Help Me?
 - <https://www.autodesk.com/autodesk-university/class/Revit-Data-Forge-How-Can-Design-Automation-Revit-API-Help-Me-2018>
 - Sasha Crotty, Revit Product Manager, Autodesk
- Manage Revit Digital Content using Forge Design Automation API for Revit.
 - <https://www.autodesk.com/autodesk-university/class/Manage-Revit-Digital-Content-using-Forge-Design-Automation-API-Revit-2018>
 - Marc Durand, Digital Disruption Director, WS Atkins & Partners
 - Mustafa Salaheldin, Data Science Manager, WS Atkins & Partners
- Design Automation: Revit and Inventor – Better Together
 - <https://www.autodesk.com/autodesk-university/class/Design-Automation-Revit-and-Inventor-Better-Together-2018>
 - Cyrille Fauvel, Forge Developer Advocate, Autodesk
 - Adam Nagy, Forge Developer Advocate, Autodesk

ロードマップ

招待者向けベータ版



ドキュメント修正
フィードバック収集
作成と抽出のワークフローを
サポート

パブリックベータ版



すべての Forge パートナーに
ベータ版を公開
フィードバック収集
スケーラビリティの検証

パブリックリリース版



TBD
パブリック版の公開
プロダクションレベルのサ
ポート

リリース後



Revit Cloud Worksharing
の サポート
Dynamo スクリプトサポート
SOC2 認証の取得

パブリック ベータ版公開のご案内

2019年 1月 28日（米国時間）に公開予定





AUTODESK®

Make anything.

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.

© 2017 Autodesk. All rights reserved.