



# FORGE - BIM 360 活用セミナー 2020

## BIM 360 Docs と Design Automation API for AutoCAD の自動化

伊勢崎俊明

Developer Advocacy & Support



# アジェンダ – ご紹介する内容

- § Design Automation API とは
- § DA4A の仕組みと使用例
- § DA4A を用いた PDF 生成
- § 成果ファイルの BIM 360 Docs への保存
- § Web Hooks API を使った拡張可能性

# Design Automation API とは



# Design Automation API とは？

**クラウドで稼働するオートデスク製品コアエンジン  
（3ds Max、AutoCAD、Inventor、Revit）に  
開発したアドイン・プラグインをロードさせて  
処理を自動実行させる Forge API の1つ**

**クラウド ストレージからファイルを読み込んで（ダウンロード）  
アドイン・プラグインによる自動処理後  
生成ファイルをクラウド ストレージに保存（アップロード）が可能**



# Design Automation API の誤った認識

- § **製品を対話・対面操作するシンククライアント環境ではありません**
  - § 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 ではありません**
  - § フロント UI となる Web ページは HTML で実装が必要です
  - § Forge Viewer は Design Automation API には含まれません
- § **処理に利用するファイルはローカル PC から直接入出力出来ません**
  - § パブリック クラウド ストレージの利用が必須です
  - § DA API の作業領域へ/からアップロード/ダウンロードします

# Design 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、または、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）

# Design 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

§ AppBundle（事前登録が必要）

## § 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（事前登録が必要）

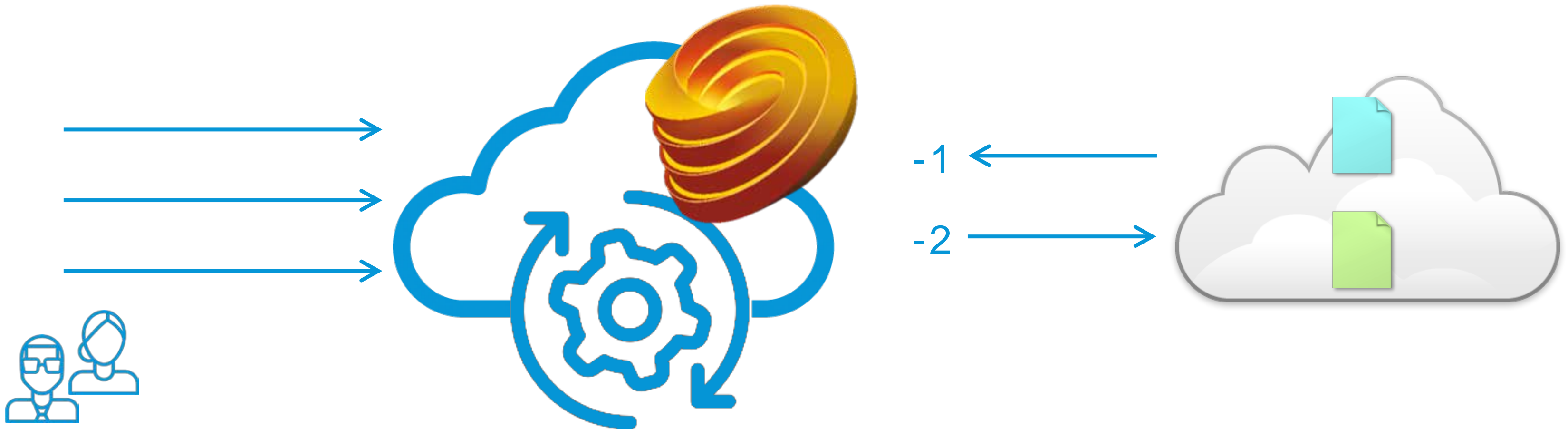
§ WorkItem（ファイル/パラメータ渡しを実行）

## § 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）

# Design Automation API とは

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





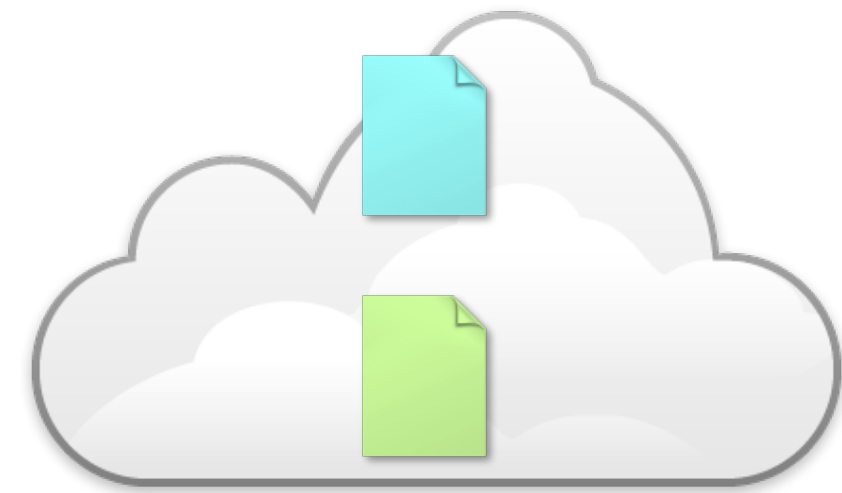
# Design Automation API とは

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

RESTful API

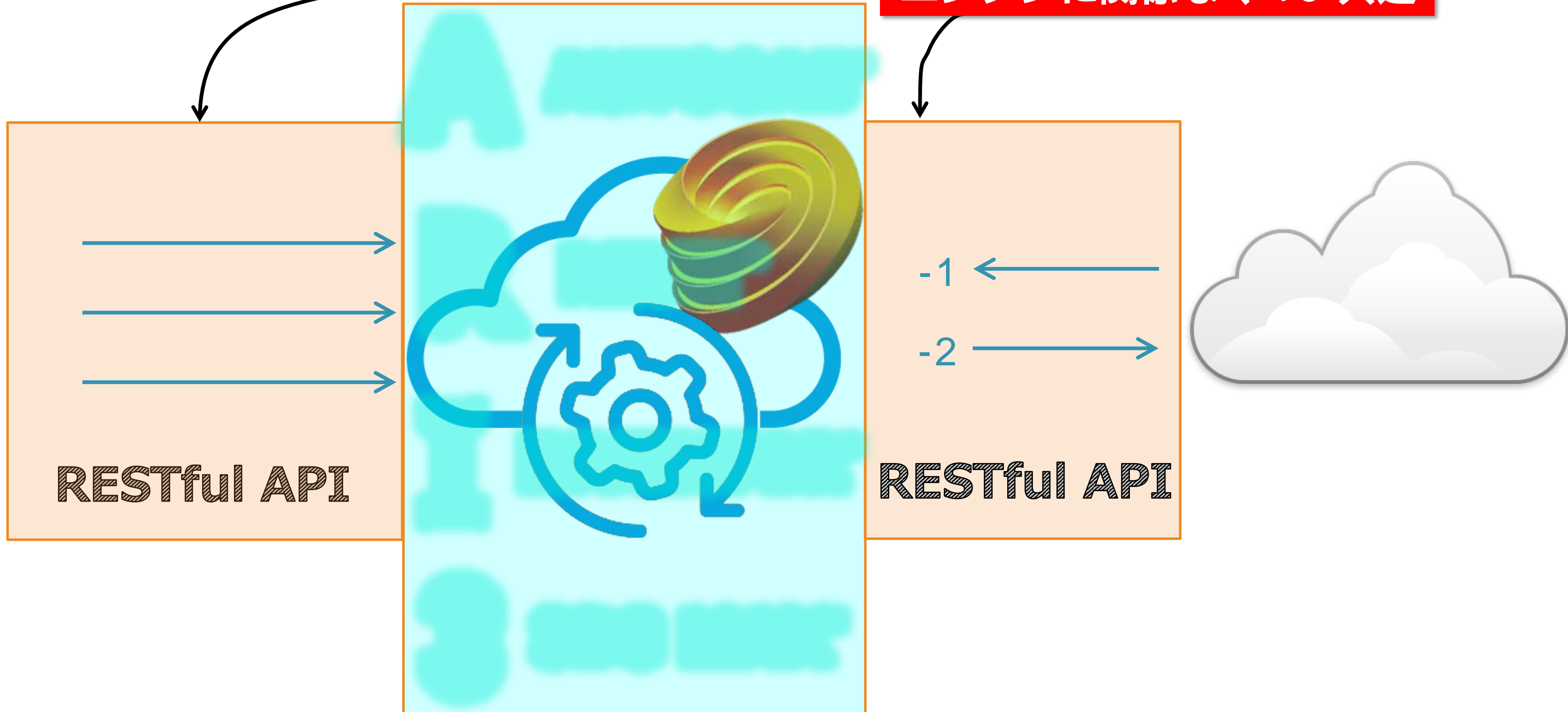
Desktop CAD API

RESTful API



# Design Automation API v3 は

エンジンに関係なく v3 共通

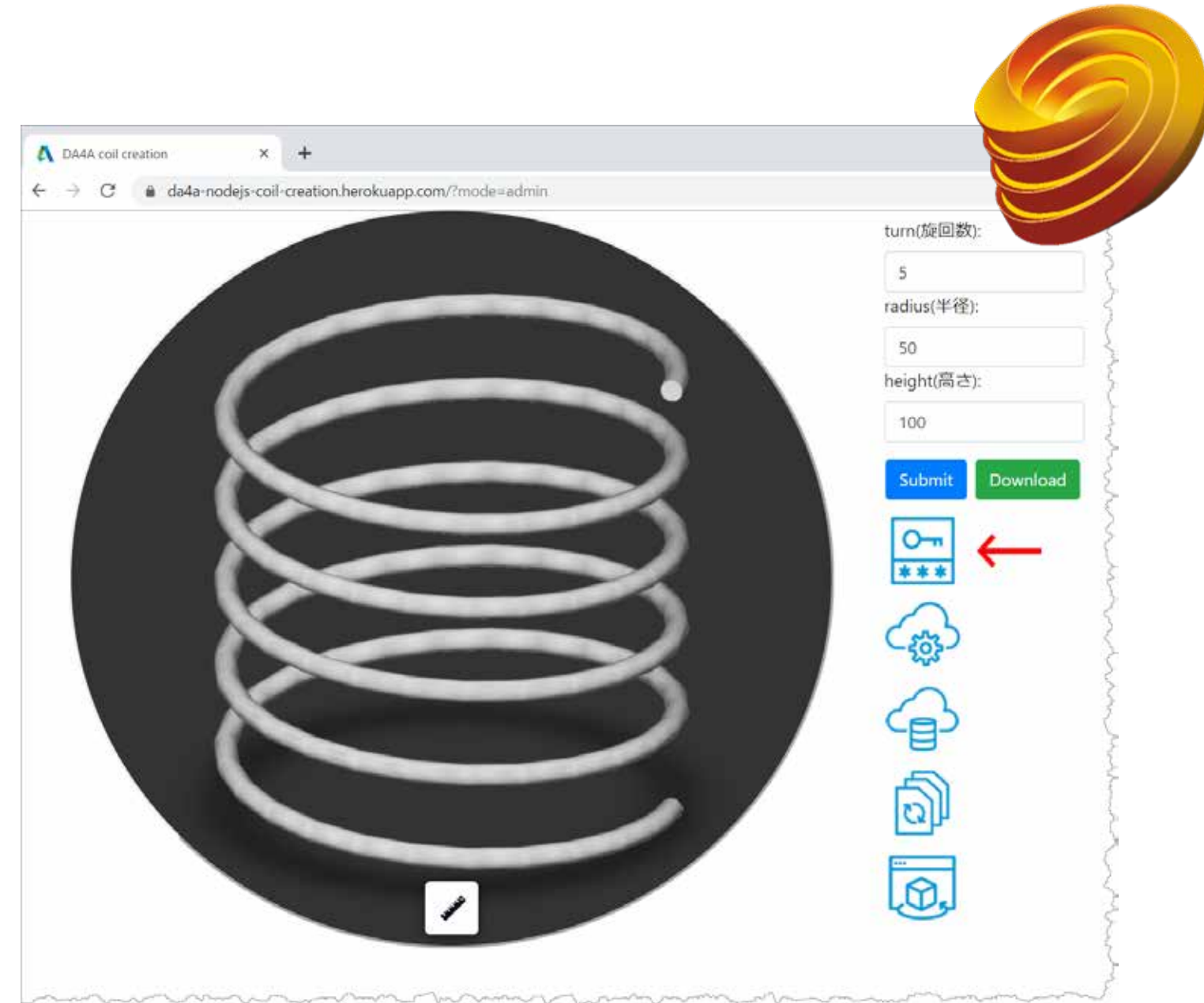
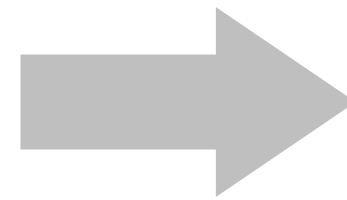
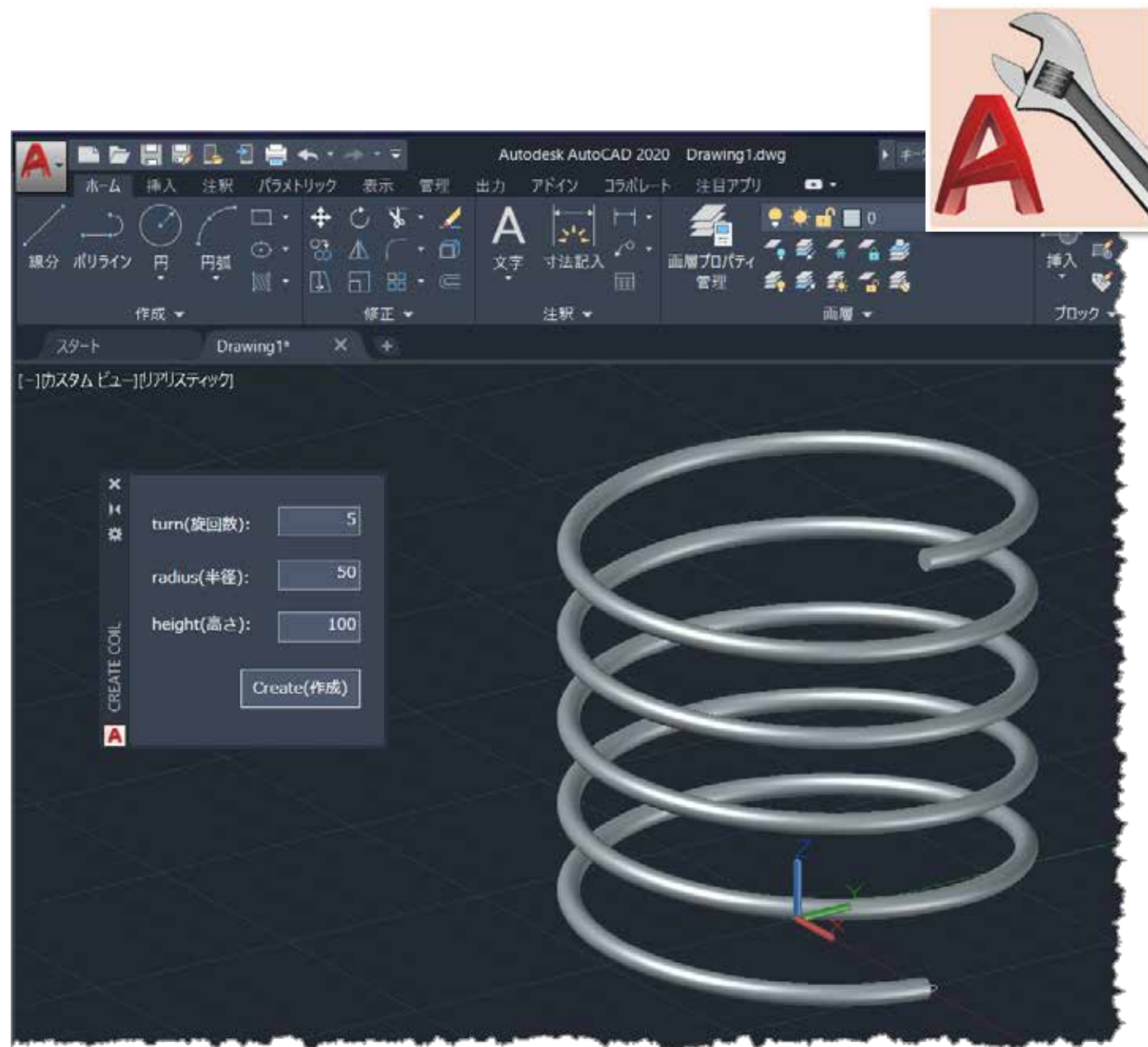


Design Automation API for AutoCAD

DA4A の仕組みと使用例



# 例) Design Automation API for AutoCAD





# AppBundle - App バンドル

## § アドイン/プラグイン ファイル



+

## § アドインをロードさせる 自動ローダ 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)



# Activity - アクティビティ

- § アクティビティの種類

- § **事前定義**（Pre-Defined）**アクティビティ**

  - § オートデスクが提供するアクティビティ

- § **カスタム アクティビティ：独自のアクティビティ**

  - § カスタム コマンドやスクリプトを実行

- § 入出力パラメータを宣言

- § URL 引数によって提供

- § アクティビティは共有またはプライベート化が可能

# WorkItem - ワークアイテム

- § 設定によってワークアイテムを作成:
- § 関連付けられた 'アクティビティ'
- § 入力パラメータ (URL)
- § 出力パラメータ (URL)
- § ワークアイテムの進捗状態:
- § 'ステータス' ポーリングによる進捗状態と処理終了の検出
- § コールバック URL の提供による処理終了の検出

# 疑問：各種パラメータの受け渡しは？

§ DA API で実行するアドインへの値の受け渡し

§ カスタム アクティビティ（カスタム コマンド）実行時に必須

The diagram illustrates the flow of parameter passing. On the left, a dark-themed 'CREATE COIL' dialog box contains three input fields: 'turn(旋回数):' with the value '5', 'radius(半径):' with the value '50', and 'height(高さ):' with the value '100'. A 'Create(作成)' button is at the bottom. A large grey arrow points from this dialog to the right. On the right, a light-themed form displays the same three parameters: 'turn(旋回数):' with value '5', 'radius(半径):' with value '50', and 'height(高さ):' with value '100'. Below these fields are two buttons: a blue 'Submit' button and a green 'Download' button.

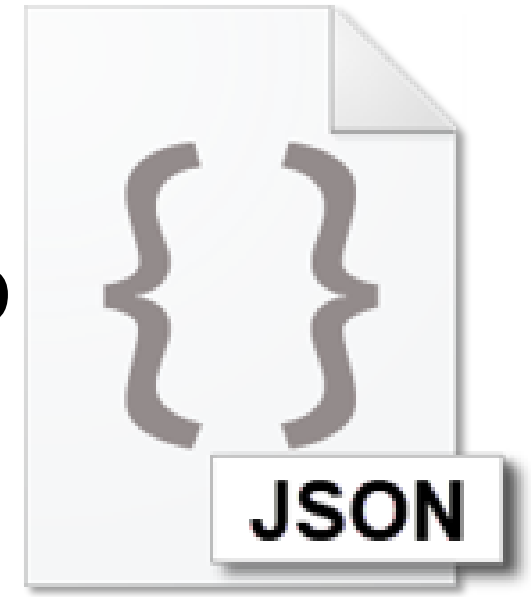
§ DA API で扱うファイルの指定（図面、パーツ、BIM etc）

§ コアエンジンに開かせる図面ファイル

§ カスタム アクティビティ（カスタム コマンド）の成果ファイル

# 回答：パラメータは JSON 経由で

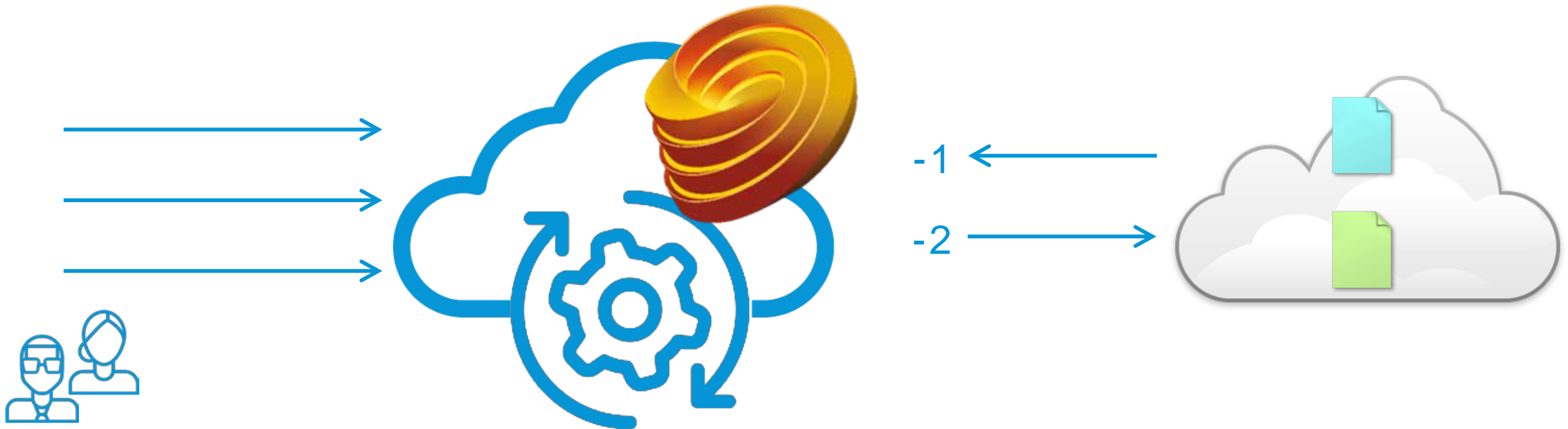
- § JSON（ジェイソン）：JavaScript Object Notation
- § 軽量で単純なデータ交換フォーマット
- § {} 括弧で囲んでデータを表現
- § JavaScript の派生ながら Web 全体でデータ交換に利用されている
- § Forge でも JSON を多用
  - § データを Forge サーバーに渡す
  - § Forge サーバーからデータを返す
  - § AppBundle 登録時の JSON 例



```
{  
  "id": "CreateCoil",  
  "engine": "Autodesk.AutoCAD+23_1",  
  "description": "Create a coil"  
}
```

# 疑問：入力ファイルと出力ファイルの扱い

- § プログラムでローカル PC のファイル操作の自動化は不可
  - § ローカル PC から DA API 作業領域に参照ファイルを保存
  - § DA API 作業領域から成果ファイルをローカル PC に保存
- § Web セキュリティ上の制限：ユーザ操作が必要





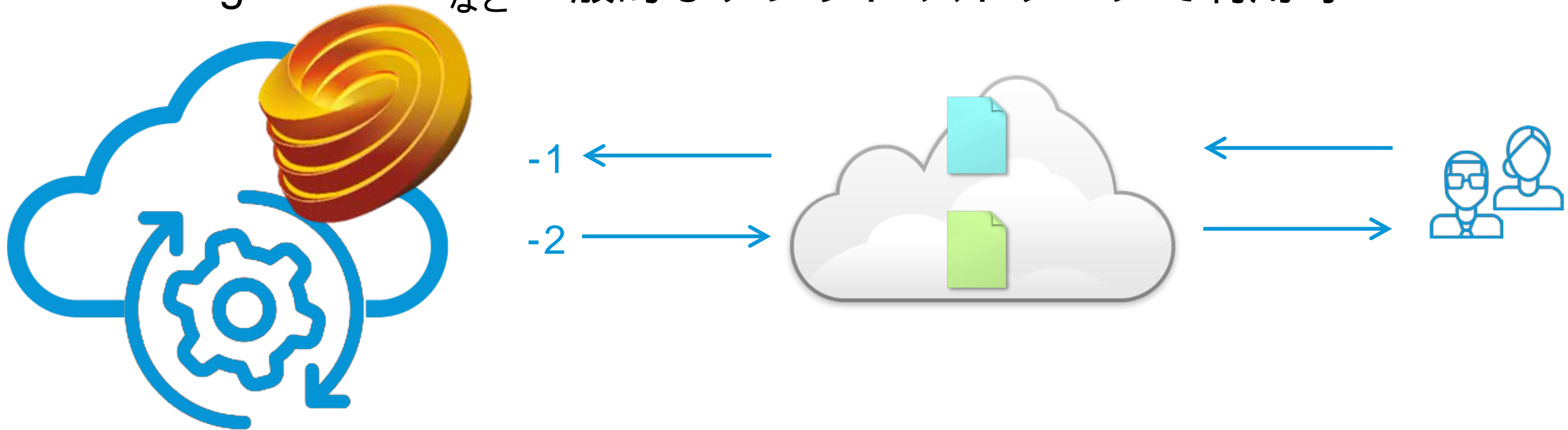
# 回答：入出力ファイル指定は**署名付き URL** で

§ ユーザにリソース アクセス権を有効期限付きで一時的に提供

§ クラウド ストレージから DA API 作業領域に参照ファイルを保存

§ DA API 作業領域から成果ファイルをクラウド ストレージに保存

§ OSS Bucket、A360/Fusion Team、BIM 360 Docs、AWS S3、Google Cloud など一般的なクラウド ストレージで利用可



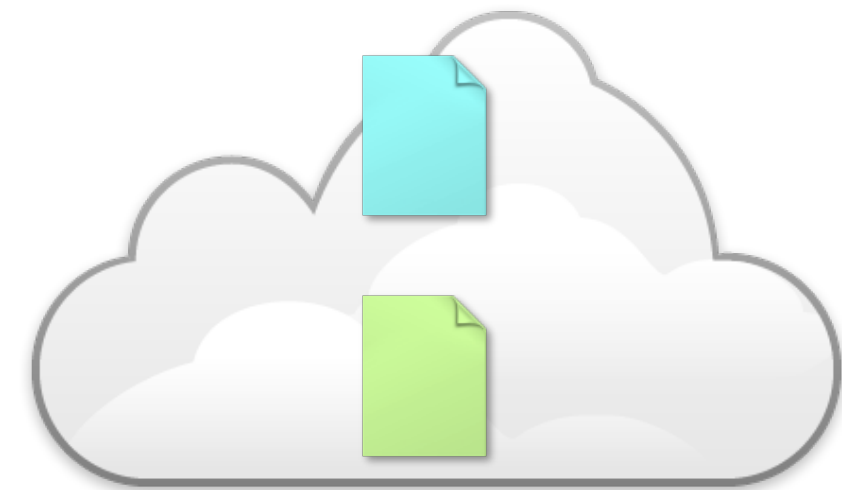
# 再度、Design Automation API とは

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

RESTful API

Desktop CAD API

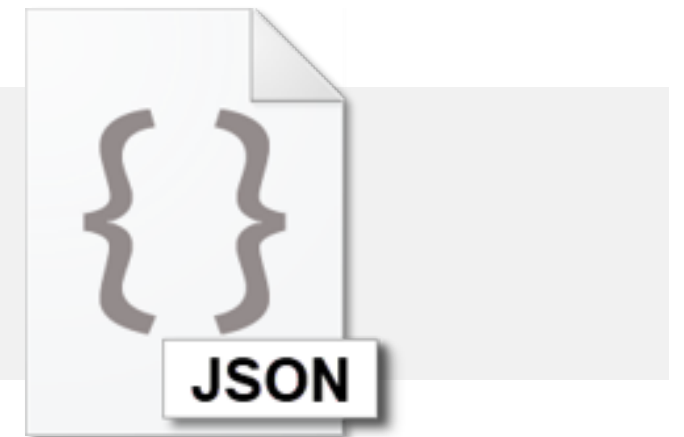
RESTful API



# ワークアイテム実行時に起こること

- § AppBundle（アドイン）は実行時に作成される作業フォルダに保存
- § DA 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

```
{  
  "turn" : "17",  
  "radius" : "70",  
  "height" : "250"  
}
```



- § 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

# Activity ( アクティビティ ) 登録時の JSON 例

## § パラメータ値の宣言

```
{
  "id": "CreateCoil",
  "commandLine": [ "$(engine.path)¥¥accoreconsole.exe /i $(args[DWGInput].path) /al
$(appbundles[CreateCoil].path) /s $(settings[script].path)" ],
  "parameters": {
    "DWGInput": {
      "zip": false,
      "ondemand": false,
      "verb": "get",
      "description": "Template drawing",
      "required": true,
    },
    "Params": {
      "zip": false,
      "ondemand": false,
      "verb": "get",
      "description": "Input parameters to create coil",
      "required": true,
      "localName": "params.json"
    },
  },
}
```

# Activity ( アクティビティ ) 登録時の JSON 例 ~ 続き

## § パラメータ値の宣言

```
"DWGOutput": {
  "zip": false,
  "ondemand": false,
  "verb": "put",
  "description": "Created drawing",
  "required": true,
  "localName": "result.dwg"
},
"settings": {
  "script": {
    "value": "CreateCoil¥n"
  }
},
"engine": "Autodesk.AutoCAD+23_1",
"appbundles": ["nqpwqsDLFGkS06LgA2mvaSXY5AeH5VSJ.CreateCoil+dev"],
"description": "Create a coil solid to new drawing"
}
```

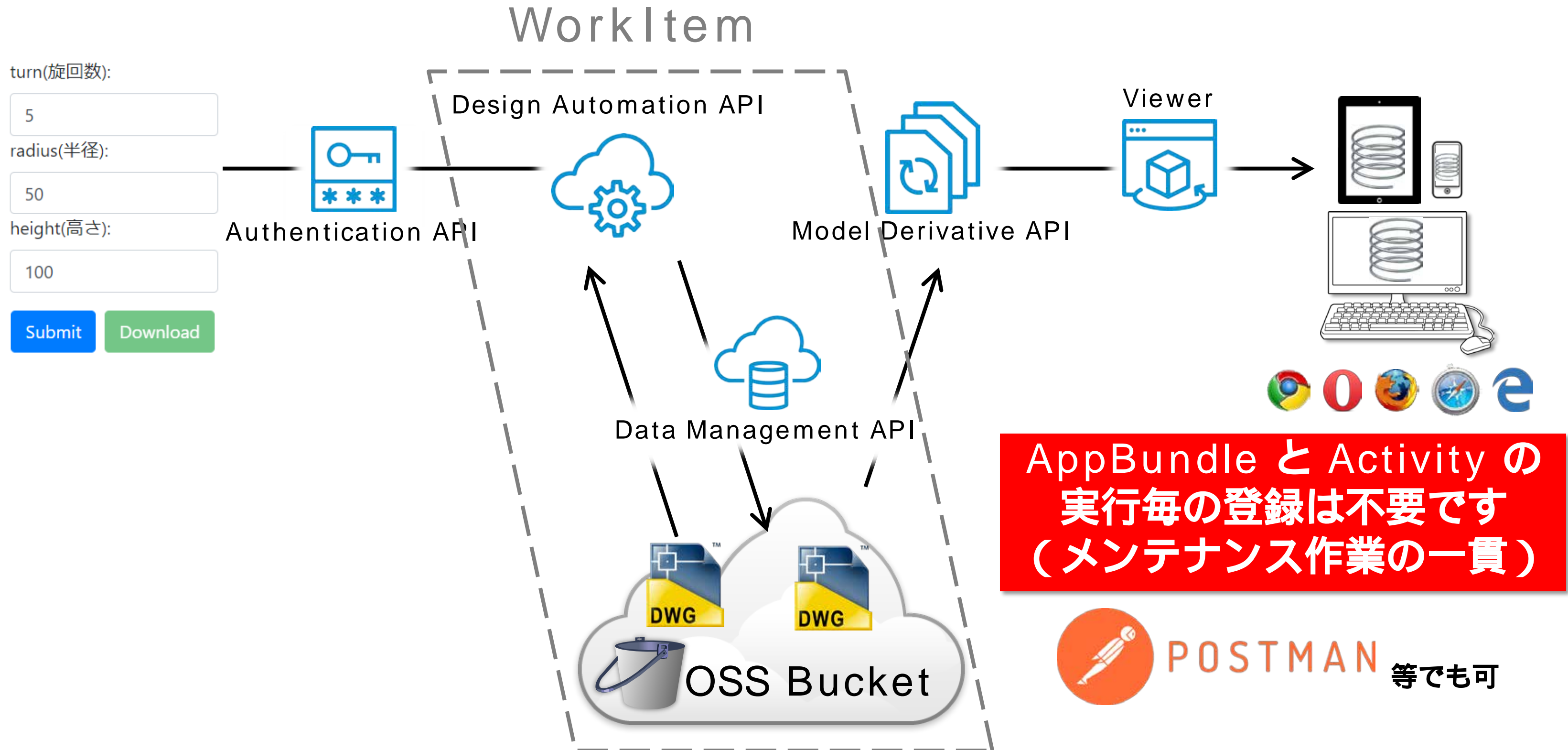


# WorkItem（ワークアイテム）実行時の JSON 例

## § パラメータ宣言に基づいて実際の値を指定

```
{
  "activityId": "nqpwwsDLFGkS06LgA2mvaSXY5AeH5VSJ.CreateCoil+dev",
  "arguments": {
    "DWGInput": {
      "url": "https://developer.api.autodesk.com/oss/v2/signedresources/xxxxx?region=US",
      "headers": {
        "Authorization": "Bearer xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "Content-type": "application/octet-stream"
      },
      "verb": "get"
    },
    "Params": {
      "url": "data:application/json,{¥\"turn¥\":¥\"17¥\", ¥\"radius¥\":¥\"70¥\", ¥\"height¥\":¥\"250¥\"}"
    },
    "DWGOutput": {
      "url": "https://developer.api.autodesk.com/oss/v2/signedresources/xxxxx?region=US",
      "headers": {
        "Authorization": "Bearer xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "Content-type": "application/octet-stream"
      },
      "verb": "put"
    }
  }
}
```

# ‘Create Coil’ サンプル実行時のワークフロー

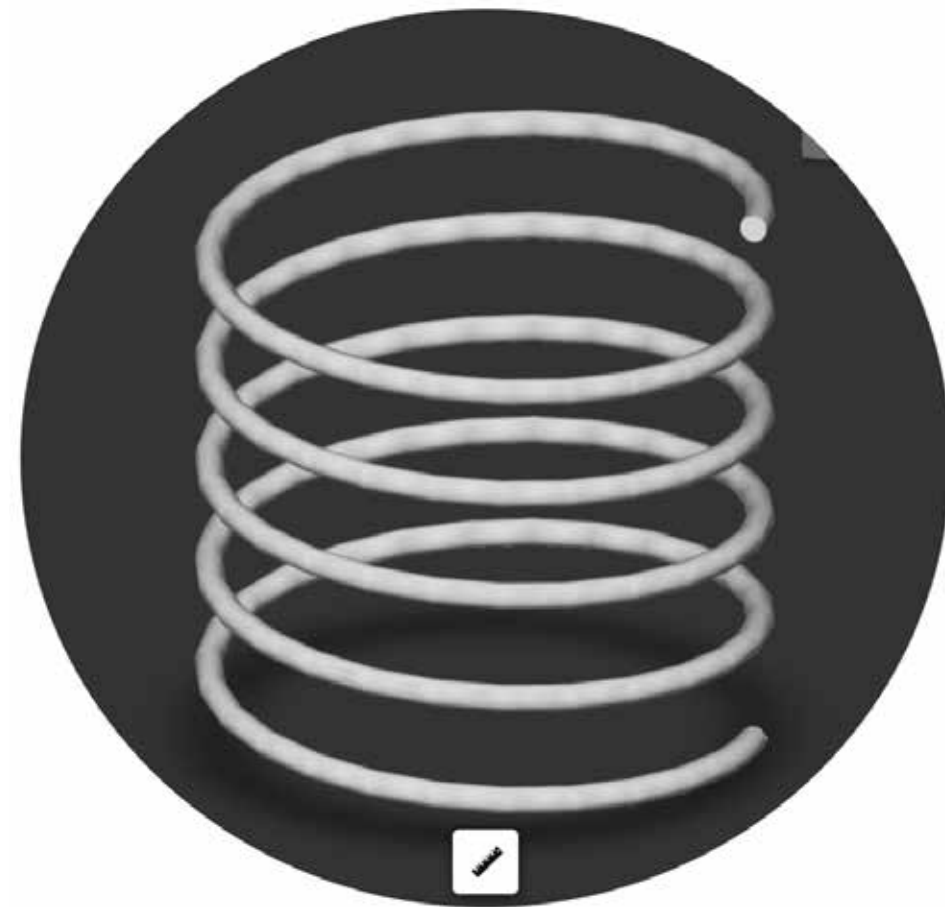


# CreateCoil サンプル改造 ~ その1

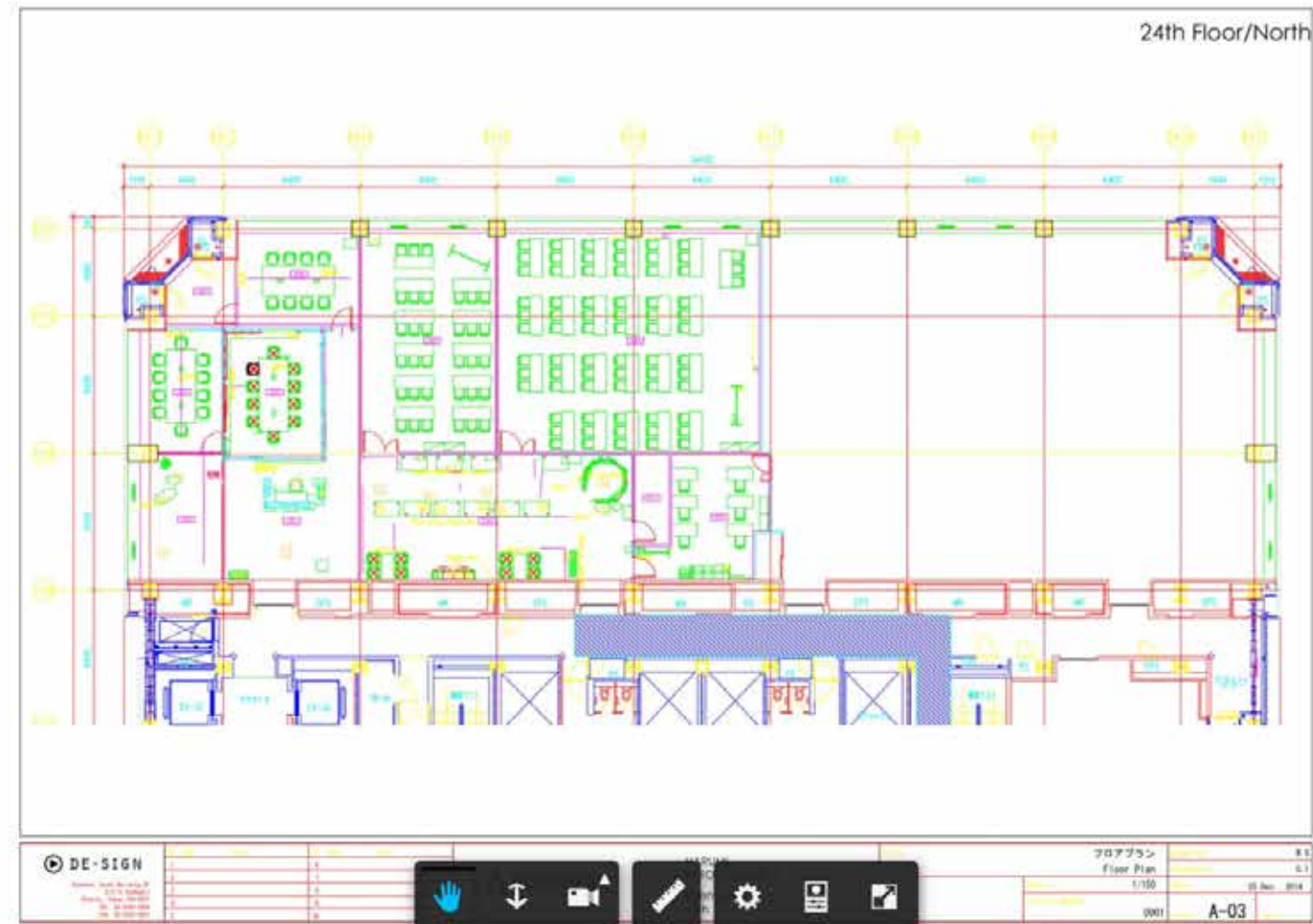
## DA4A を用いた PDF 生成



- § Forge Viewer 表示コンテンツの 2D 化
- § 入力ファイル名の可変化
- § 実行コマンドの変更



turn(回転数):  
  
radius(半径):  
  
height(高さ):



Upload

Download



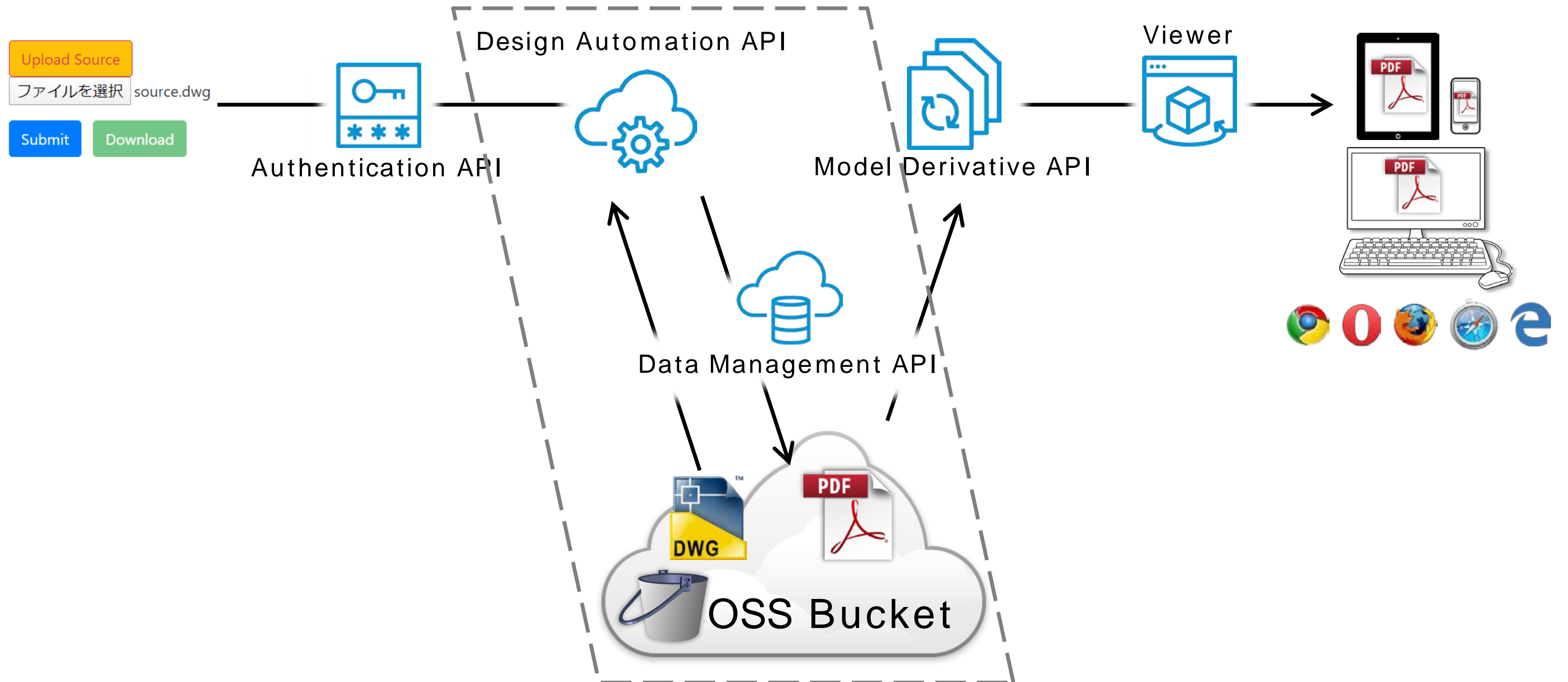
Delete Activity

Register Activity



# ‘PDF 出力’ への改造（ワークフローの考察）

WorkItem





# Activity ( アクティビティ ) 登録時の JSON 例

## § パラメータ値の宣言

```
{
  "id": "PDFPlot",
  "commandLine": [ "$(engine.path)¥¥accoreconsole.exe /i $(args[DWGInput].path) /s $(settings[script].path)" ],
  "parameters": {
    "DWGInput": {
      "zip": false,
      "ondemand": false,
      "verb": "get",
      "description": "Source drawing",
      "required": true
    },
    "PDFOutput": {
      "zip": false,
      "ondemand": false,
      "verb": "put",
      "description": "putput PDF drawing",
      "required": true,
      "localName": "result.pdf"
    }
  }
},
```

# Activity（アクティビティ）登録時の JSON 例 ～ 続き

## § パラメータ値の宣言

```
"settings": {  
  "script": {  
    "value": "_tilemode 0 -export _pdf _all result.pdf¥n"  
  }  
},  
"engine": "Autodesk.AutoCAD+23_1",  
"appbundles": [],  
"description": "PDF output"  
}
```

§ 実行コマンドは標準コマンド/システム変数の利用も可能です

§ script 指定で記述

§ AppBundle の利用は必須ではありません

§ AppBundle 不使用の場合は [] で指定

# WorkItem（ワークアイテム）実行時の JSON 例

## § パラメータ宣言に基づいて実際の値を指定

```
{
  "activityId": "nqpwqsDLFGkS06LgA2mvaSXY5AeH5VSJ.PDFPlot+dev",
  "arguments": {
    "DWGInput": {
      "url": "https://developer.api.autodesk.com/oss/v2/signedresources/xxxxx?region=US",
      "headers": {
        "Authorization": "Bearer xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "Content-type": "application/octet-stream"
      },
      "verb": "get"
    },
    "PDFOutput": {
      "url": "https://developer.api.autodesk.com/oss/v2/signedresources/xxxxx?region=US",
      "headers": {
        "Authorization": "Bearer xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "Content-type": "application/octet-stream"
      },
      "verb": "put"
    }
  }
}
```

# 補足：日本語 TrueType フォントについて

## § DA4A 稼働環境：

§ Amazon Machine Image でデプロイされた英語 Windows 環境

§ 日本語フォントの扱い

§ 日本語シェイプ フォントはコアエンジン付帯で利用可能

§ 日本語 TrueType フォントは一部利用可能

§ MS ゴシック、MS P ゴシック、MS UI ゴシック

MS 明朝、MS P 明朝、游 明朝、メイリオ、メイリオ UI

§ 上記以外の利用：

AppBundle への同梱、または、

WorkItem 指定で実行時ダウンロードが必須

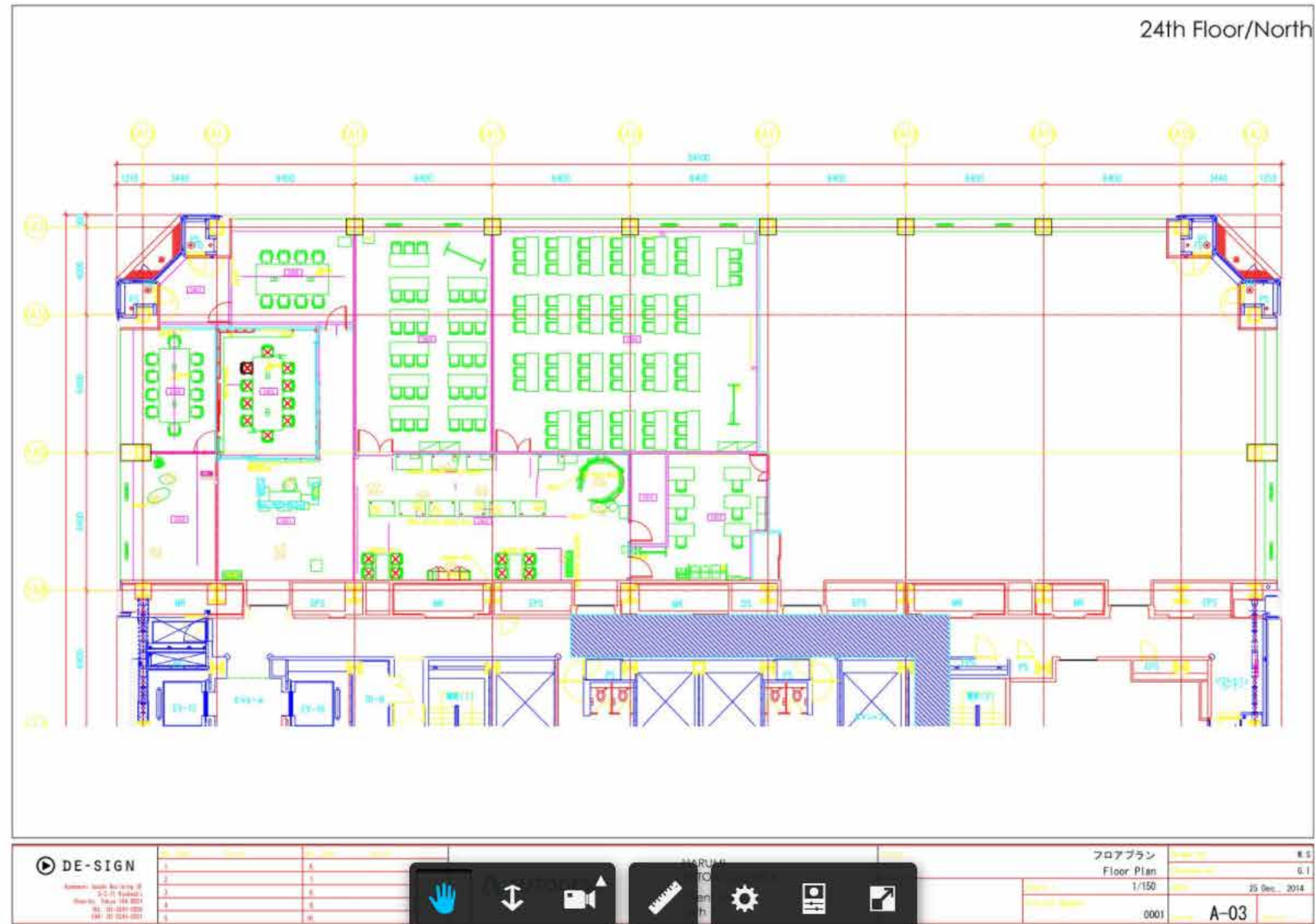
§ フォントが未解決の場合は FONTALT システム変数を採用

CreateCoil サンプル改造 ~ その2

成果ファイルの BIM 360 Docs への保存

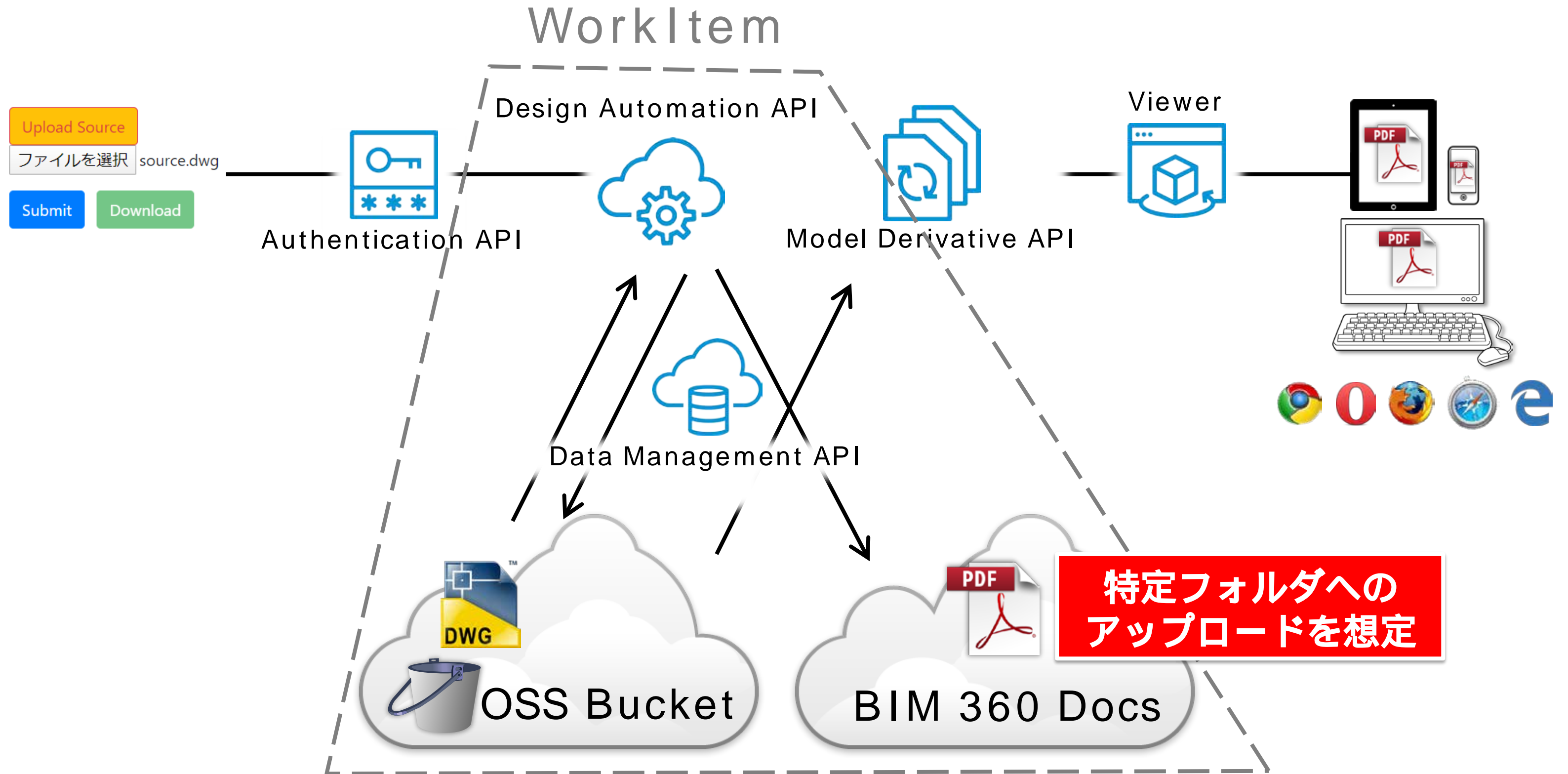


## § BIM 360 Docs フォルダへの成果 PDF 保存





# ‘PDF 出力’ への改造（ワークフローの考察）





# DA4A 作業領域から BIM 360 Docs へのアップロード

§ WorkItem で指定

§ BIM 360 Docs 用 signed URL 作成不可

§ Bucket 名 wip.dm.prod と Object 名が必要

§ Object 名は Storage Object の作成で動的に生成される

§ **運用シナリオに依存**

§ 3-legged Access Token : ユーザ レベル

§ 2-legged Access Token : スーパーユーザ レベル



# BIM 360 Docs ファイル アップロード（登録）

## § Version **登録が必須**

§ ファイル アップロードだけでは認識されない

## § OnComplete コールバックの利用

§ WorkItem の完了を通知

§ 次のアクションへの実装に最適

§ 2-legged/3-legged Access Token 利用可

§ バージョン登録で利用可能！

§ Forge サーバー（外部）からの通知を受信

§ DNS 解決出来ない為 localhost:3000 を呼び出せない！



# Activity ( アクティビティ ) 登録時の JSON 例

## § パラメータ値の宣言 ( 抜粋 )

```
"parameters": {  
  "DWGInput": {  
    "zip": false,  
    "ondemand": false,  
    "verb": "get",  
    "description": "Source drawing",  
    "required": true  
  },  
  "PDFOutput": {  
    "zip": false,  
    "ondemand": false,  
    "verb": "put",  
    "description": "Output PDF drawing",  
    "required": true,  
    "localName": "result.pdf"  
  },  
  "PDFOutputToBIM360": {  
    "zip": false,  
    "ondemand": false,  
    "verb": "put",  
    "description": "Upload PDF drawing to BIM 360 Docs",  
    "required": false,  
    "localName": "result.pdf"  
  }  
}
```

# ローカル開発環境では



§ 通知コールバック用ローカル URL のトンネル化が必要

§ <https://ngrok.com/download>

§ 例 ) `http://localhost:3000`      <http://de9feb1c.ngrok.io>

§ コマンド プロンプト上で `ngrok.exe http <ポート番号>`

A screenshot of a Windows command prompt window. The title bar shows the file path 'C:\Users\iseza\OneDrive\デスクトップ\イベント\Forge - BIM 360 活用セミナー\ngrok.exe - ngrok http 3000'. The terminal output shows the ngrok service is online and forwarding traffic from a public URL to localhost:3000. It also displays connection statistics and the start of an HTTP request log.

```
C:\Users\iseza\OneDrive\デスクトップ\イベント\Forge - BIM 360 活用セミナー\ngrok.exe - ngrok http 3000
ngrok by @inconshreveable

Session Status      online
Account             ToshiakiIsezaki (Plan: Free)
Version             2.3.35
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://de9feb1c.ngrok.io -> http://localhost:3000
Forwarding           https://de9feb1c.ngrok.io -> http://localhost:3000

Connections         ttl    opn    rt1    rt5    p50    p90
                   0      1      0.00   0.00   0.00   0.00

HTTP Requests
-----
POST /onComplete
```

# WorkItem（ワークアイテム）実行時の JSON 例

## § パラメータ宣言に基づいて実際の値を指定

```
{
  "activityId": DA4A_FQ_ID,
  "arguments": {
    :
    "DWGInput": {
      :
    },
    "PDFOutput": {
      :
    }
  }
  "PDFOutputToBIM360": {
    "url": "https://developer.api.autodesk.com/oss/v2/buckets/wip.dm.prod/objects/" + name,
    "headers": {
      "Authorization": "Bearer xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "Content-Type": "application/octet-stream"
    },
    "verb": 'put'
  },
  "onComplete": {
    "verb": "post",
    "url": "http://de9feb1c.ngrok.io/onComplete"
  }
}
```

# BIM 360 Docs ファイル アップロード詳細

Activity で成果ファイルのアップロードを宣言  
成果ファイルの保存先 BIM 360 Docs フォルダに Storage を作成  
WorkItem にアップロードする URL ( Storage ID ) を指定  
WorkItem の処理終了を OnComplete コールバックで検出  
OnComplete コールバック内でバージョンを登録

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

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

	アップロード者	アップロード先	アップロード日
result.pdf		Plans/Japan RUG sample/DA4A	2020年4月11日 17:47
result.pdf	Toshiaki Isezaki	Plans/Japan RUG sample/DA4A	2020年4月11日 17:33

# Forge Webhooks API とは

## クラウド上で発生する様々なイベントを Forge アプリに通知するメカニズムを提供

- § BIM 360、A360、Fusion Team ストレージのイベントを監視
  - § フォルダ/ファイル/バージョンの追加、削除、移動、コピー、名前変更
- § Model Derivative API での変換経過/完了通知
- § Revit Cloud Worksharing イベント通知
- § Fusion Lifecycle イベント通知





## 1. Select folder:

## 2. Select events:

## 3. Contact information:

- Augusto Goncalves
- Augusto Goncalves
    - 3D Print
    - 3d Viewer - EMWorks
    - Augusto's First Project
      - Augusto's First Project
    - Autodesk 360 Experience
    - Forge
    - Jim's First Project
    - test2
  - Autodesk Forge Partner Development
  - DevTech AEC

### Files & Versions

- ☐ Added
- ☐ Modified
- ☐ Copied
- ☐ Moved
- ☐ Deleted

### Folders

- ☐ Added
- ☐ Modified
- ☐ Copied
- ☐ Moved
- ☐ Deleted

### SMS notification



(201) 555-0123

### Email notification



Enter email

### Slack notification



https://hooks.slack.com/services/

e.g. T000000000/B000000000/XXXXXXXXXXXXXX

Create notification

<https://bim360notifier.autodesk.io/>

§ A360/BIM 360 Docs/Fusion Team イベントの監視/通知

§ プロジェクト内のフォルダ、ファイル/バージョンの監視

§ 外部 Web API との連携

§ SMS への通知 >>  twilio

§ <https://twilio.kddi-web.com/>

§ Email への送信（通知）>>  Postmark

§ <https://postmarkapp.com/developer/api/email-api>

§ Slack チャンネルへの通知 >>  slack API

§ <https://api.slack.com/web>



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 and services offerings, and specifications and pricing 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.