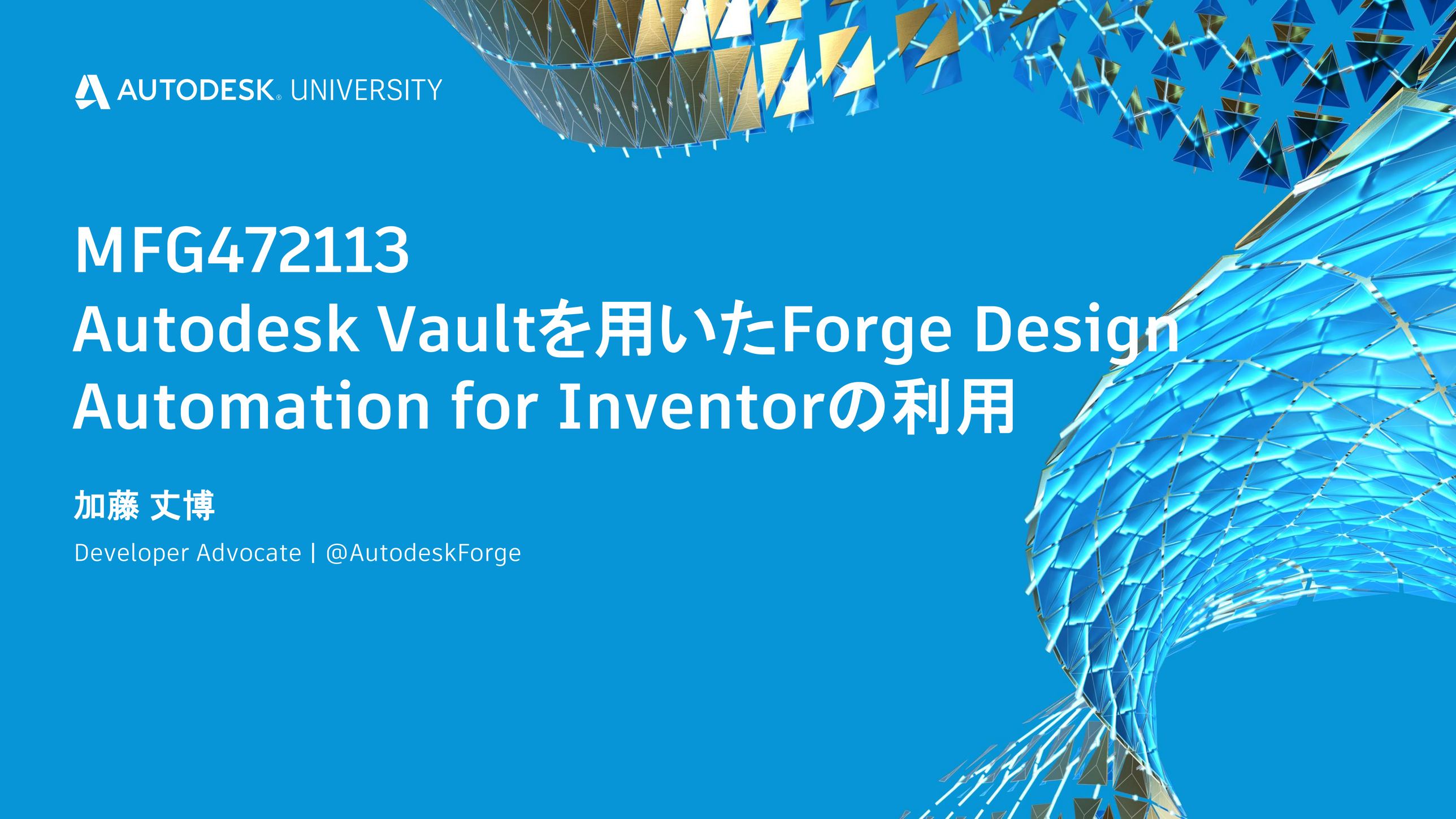
AUTODESK. UNIVERSITY

MFG472113
Autodesk Vaultを用いたForge Design Automation for Inventorの利用

加藤 丈博

Developer Advocate | @AutodeskForge



このセッションの内容

- Vault APIを用いたVaultからのファイルの取得および、取得したファイルをDesign Automation for Inventorへの入力として利用する方法の学習
- クラウド環境にあるDesign Automation for Inventorが、どのようにInventorのカスタム Plug-inを実行するかの学習
- カスタムアプリケーションを通じて、APIデスクトップアプリとWebサービスAPIという異なる種類のAPIを組み合わせて利用する方法を学習

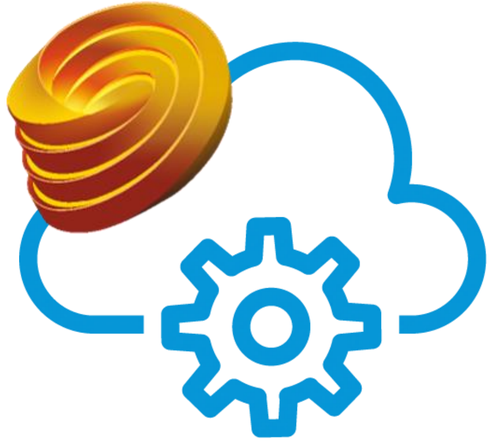
加藤 丈博

Developer Evangelist, Developer Technical Services
Forge Partner Development

流通業向けシステム開発、電気設計向けのPDM/PLMパッケージソフトウェアの開発を経て、2020年からオートデスクにて、InventorやAutoCAD API のアドイン開発のサポート及び、Forge プラットフォーム API のエバンジェリストとして活動

Autodesk VaultとForge Design Automation API

VAULT

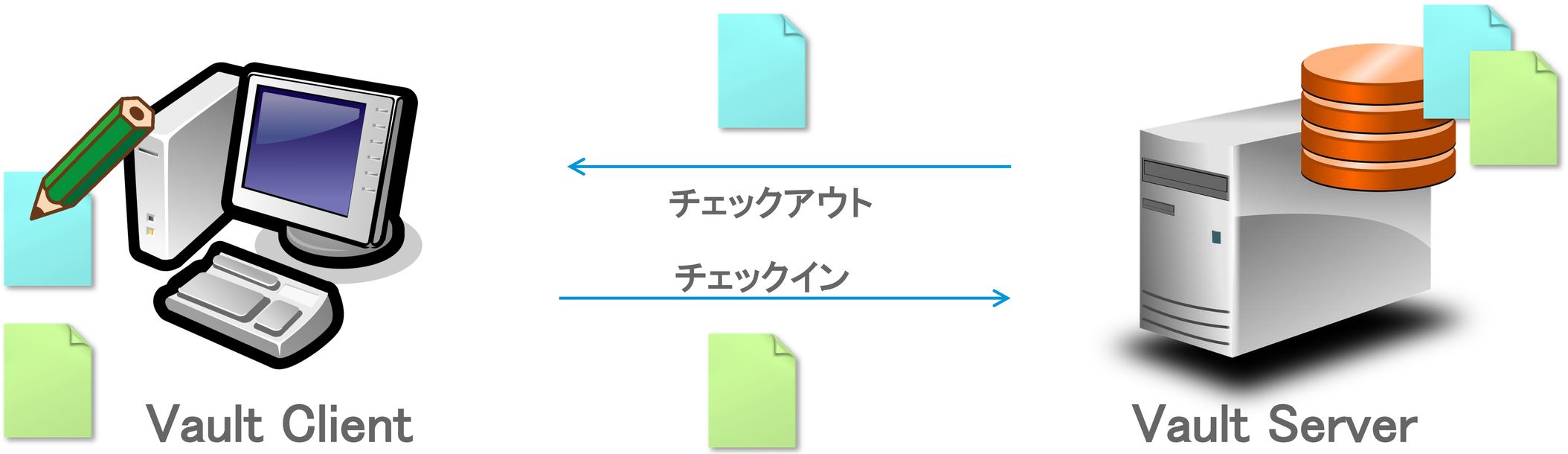


Autodesk のPDM（製品データ管理）ソフトウェア。プロジェクトの関係者全員が一元管理されたデータを使って作業を進めることで、効率的なコラボレーション、エラーの削減、時間の節約が実現。

AutodeskのWebサービス APIである Forge のうちの一つ。クラウド上で稼働するコアエンジン（v2 では AutoCAD のみ、v3 では AutoCAD、Inventor、Revit、3ds MAX ）にアドイン（プラグイン）をロードさせて、反復タスクを自動化するバッチ処理を実現。

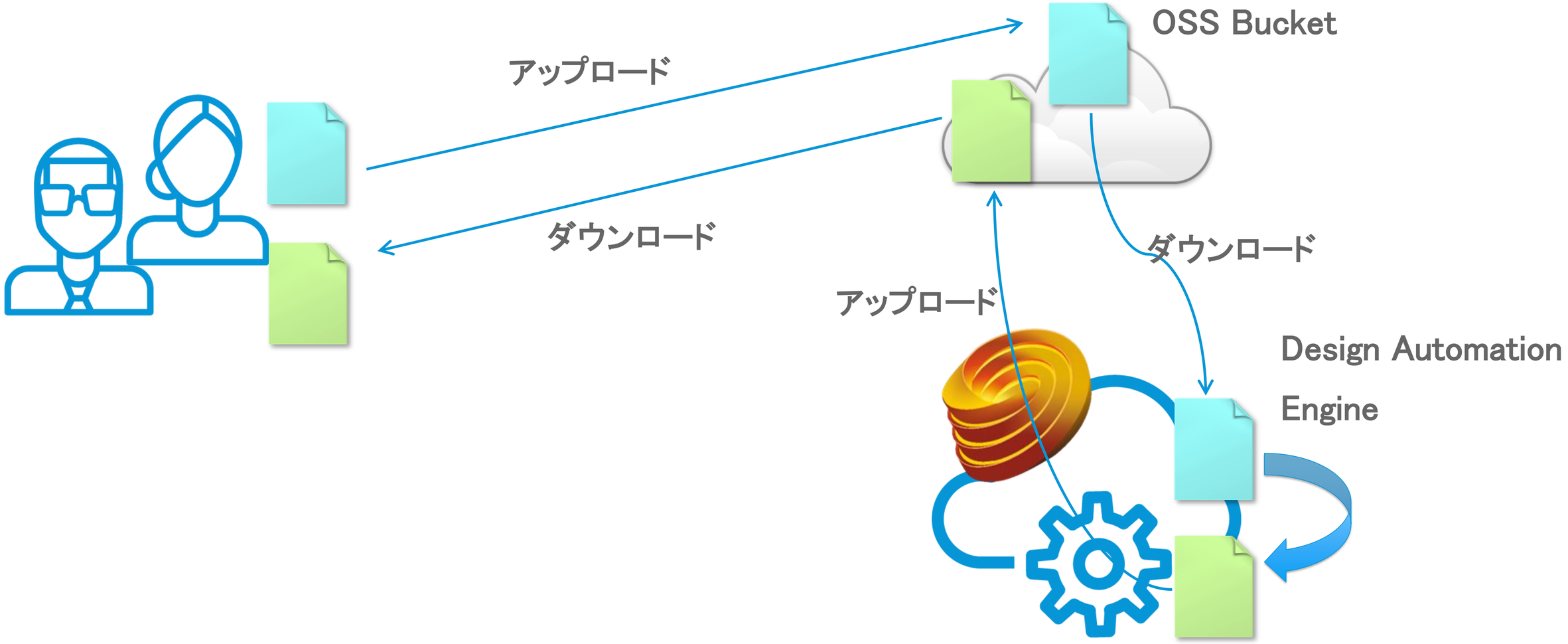
Autodesk Vault

デザインデータは、Vault Client↔Server間でやり取り



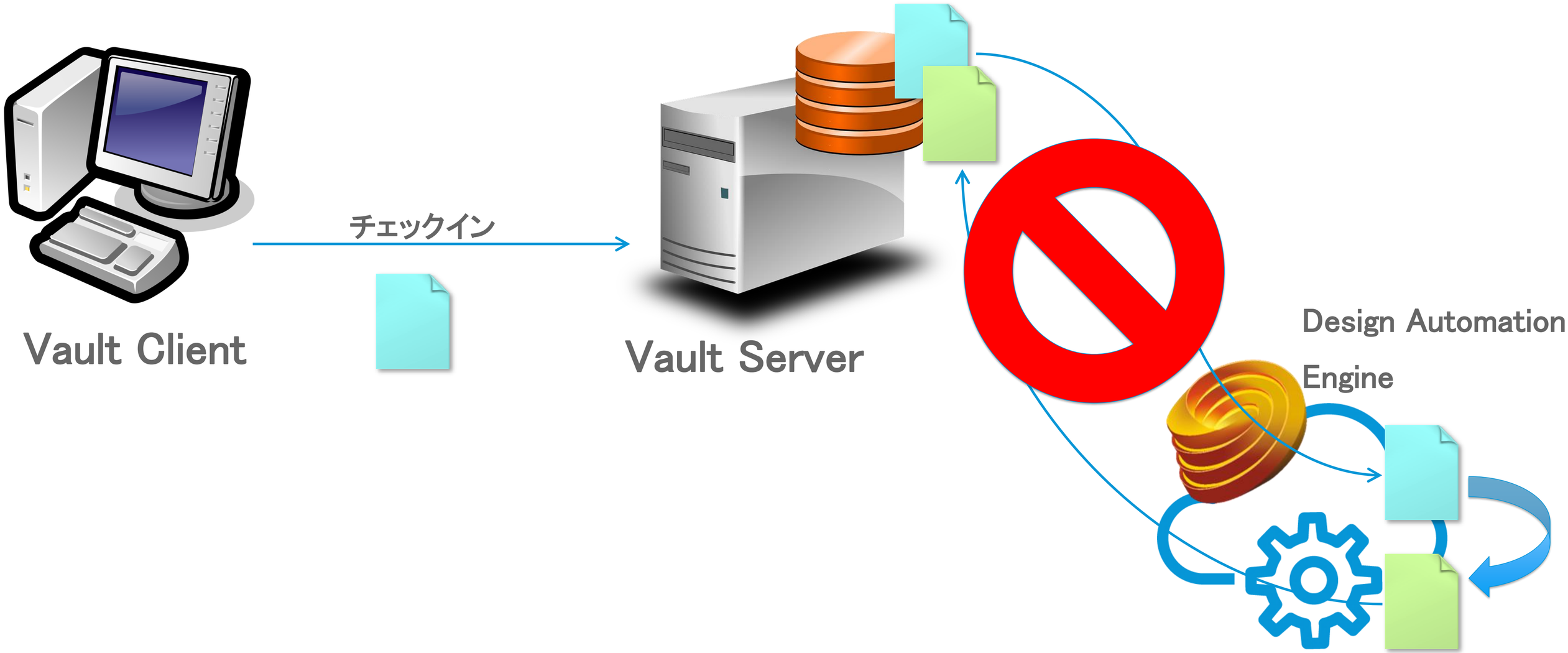
Design Automation API

デザインデータはクラウドストレージにUpload/Downloadが必要



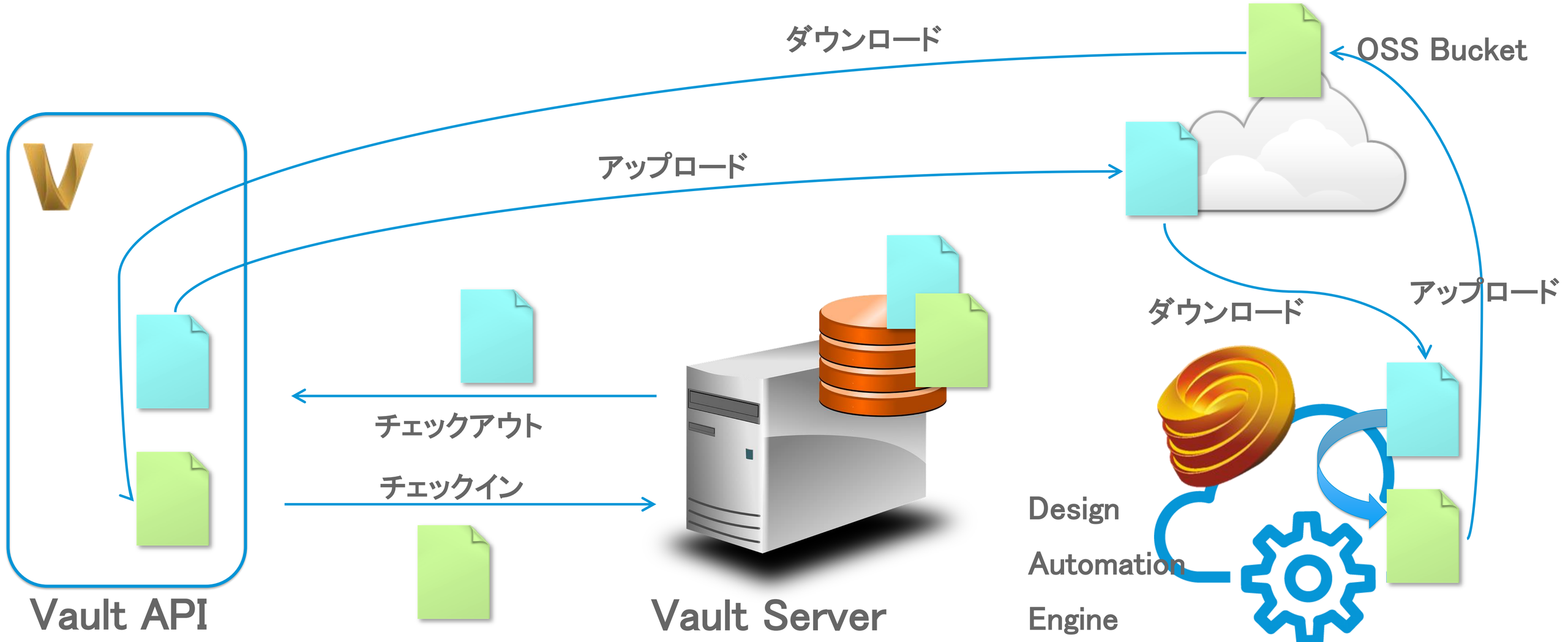
Design Automation APIとAutodesk Vault

Vaultサーバのデザインデータに、直接Design Automationの実行はNG



Design Automation APIとAutodesk Vault

Vault APIを使って、Design Automationを自動的に実行



デモ

デモアプリのソースコード

https://github.com/sajith-subramanian/Inventor_Design_Automation_with_Vault

master · 1 branch · 0 tags · Go to file · Code

sajith-subramanian Add files via upload — 746b171 on 27 Jun 2019 — 4 commits

UpdateiProp	Add files via upload	15 months ago
VaultInvDA	First commit	17 months ago
.gitignore	First commit	17 months ago
LICENSE	Create LICENSE	17 months ago
README.md	Create README.md	17 months ago
VaultInvDA.sln	First commit	17 months ago

README.md

Inventor Design Automation with Vault

platform Windows | .Net Framework 4.7.2 | Design Automation v3

license MIT | Inventor 2019 | Vault 2019

This sample is a .NET console app that demostrates the use of Inventor Design Automation on files in Vault. It launches a Vault UI that allows the user to select a file in UI, checks out the file, changes an iProperty of the file using Design Automation, and checks the file back in to Vault. The code used for the bundle can be found in the UpdateiProp project.

About

Inventor Design Automation for files in Vault

Readme

MIT License

Releases

No releases published

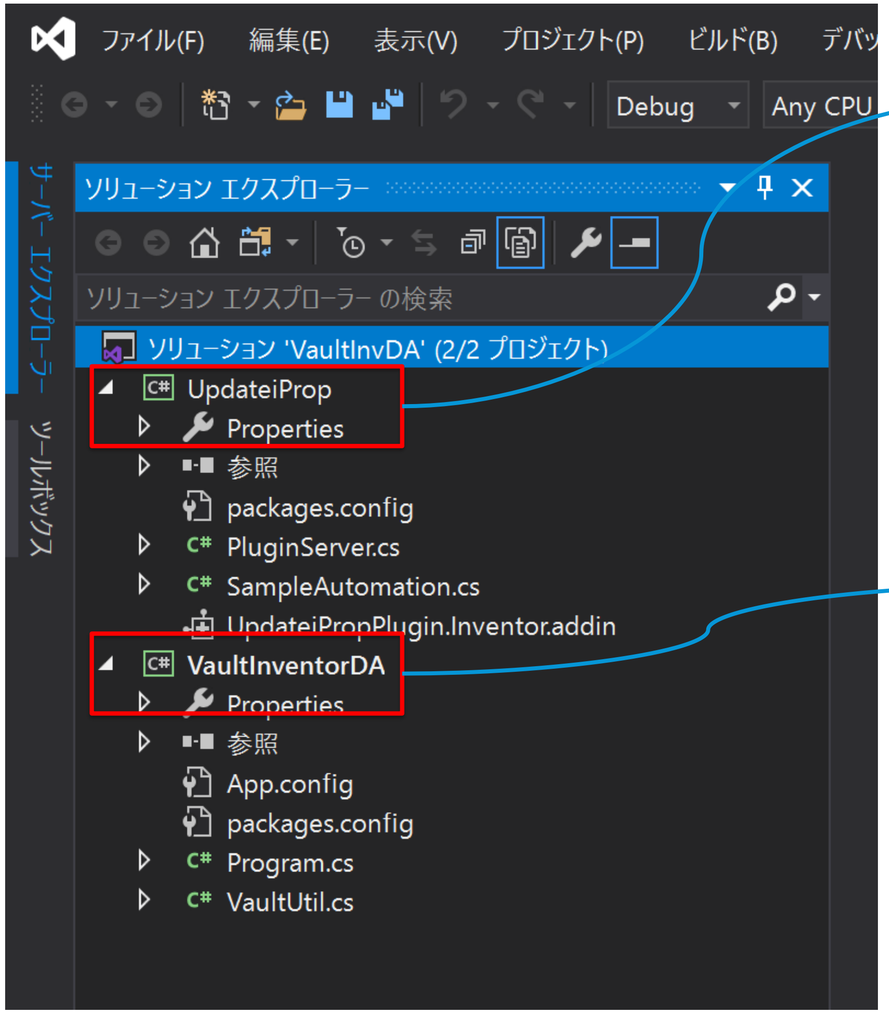
Packages

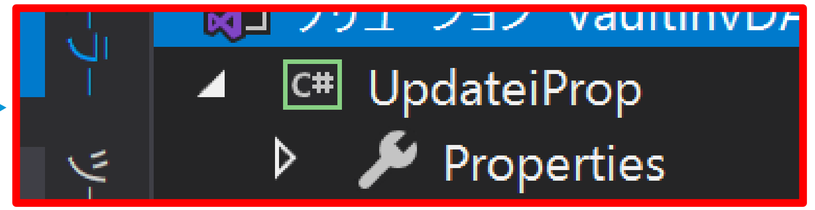
No packages published

Languages

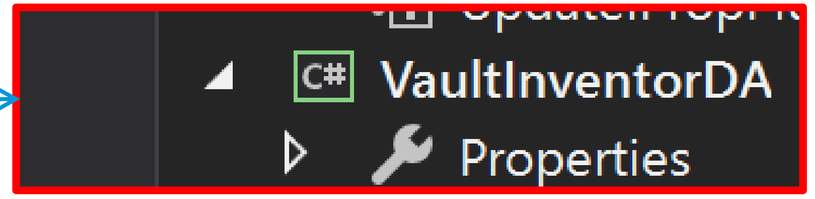
C# 100.0%

ソリューション内の2つのプロジェクト





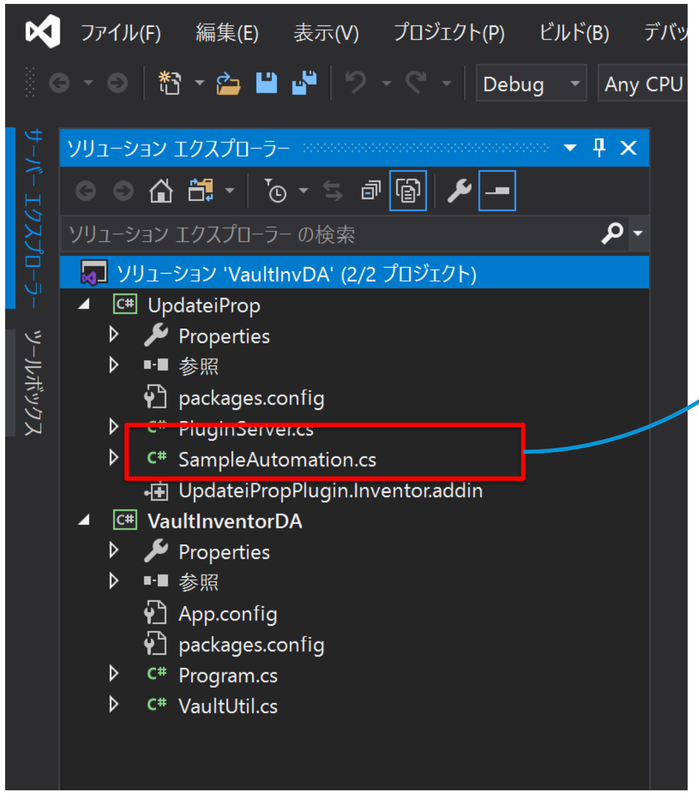
Design Automation for Inventorで実行される
Plug-inプロジェクト



Vault APIによるVaultからのデータのチェックイン・アウト、Design Automation for Inventorの起動をおこなうアプリケーションのプロジェクト

Inventor Plug-inの処理

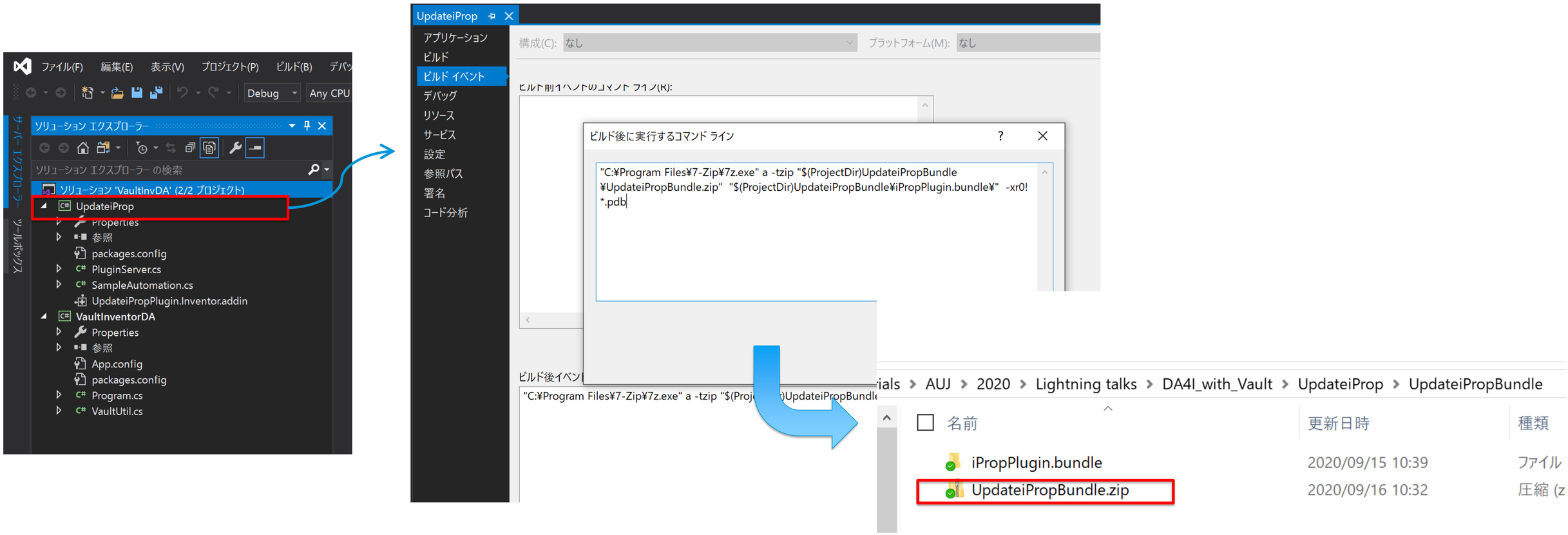
Design Automation for Inventer実行時にRun()または、RunWithArguments()メソッドが実行される




```
0 個の参照
public void Run(Document doc)
{

    Trace.TraceInformation("Running with no Args.");
    NameValueMap map = m_inventorServer.TransientObjects.CreateNameValueMap();
    RunWithArguments(doc, map);

}

1 個の参照
public void RunWithArguments(Document doc, NameValueMap map)
{
    try
    {
        StringBuilder traceInfo = new StringBuilder("RunWithArguments called with ");
        traceInfo.Append(doc.DisplayName);
        Trace.TraceInformation(map.Count.ToString());
        // values in map are keyed on _1, _2, etc
        for (int i = 0; i < map.Count; i++)
        {
            traceInfo.Append(" and ");
            traceInfo.Append(map.Value["_" + (i + 1)]);
        }
        Trace.TraceInformation(traceInfo.ToString());

        #region Update iProperty

        Trace.TraceInformation("Updating iProperty.");
        PropertySet invPropSet = doc.PropertySets["Inventor Summary Information"];
        // Get the 'Comments' property and set a value.
        Property invCommentsProperty = invPropSet["Comments"];
        invCommentsProperty.Value = "Comments added using Design Automation";
        doc.Update();
        doc.Save();

        #endregion
    }
    catch(Exception ex)
    { Trace.TraceInformation(ex.Message); }
}
```

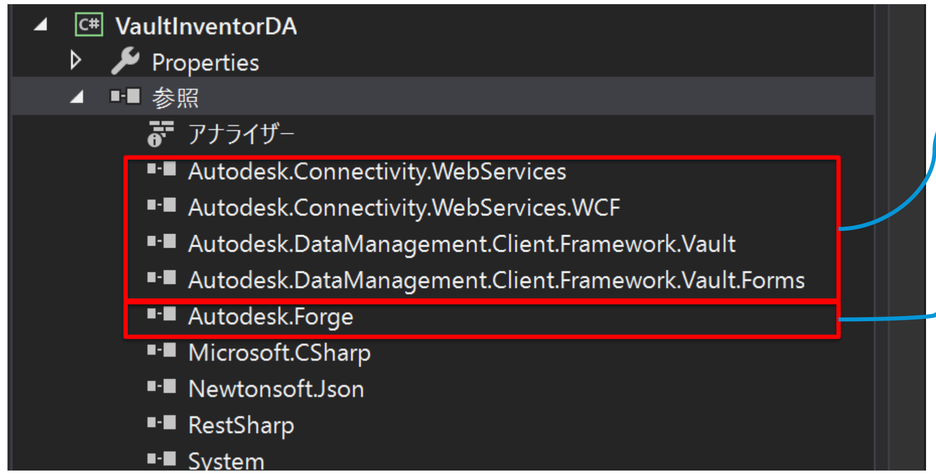
AppBundleファイルの作成

UpdateiPropのビルド後イベントで、Design Automation for Inventorに登録するAppBundle（.zipファイル）を出力



VaultInventorDAプロジェクトの参照設定

Vault SDKのDLLとForge .Net SDK



Vault SDKディレクトリ配下のDLLファイル

C:¥Program Files¥Autodesk¥Autodesk Vault 2021 SDK¥bin¥x64

Forge .Net SDK

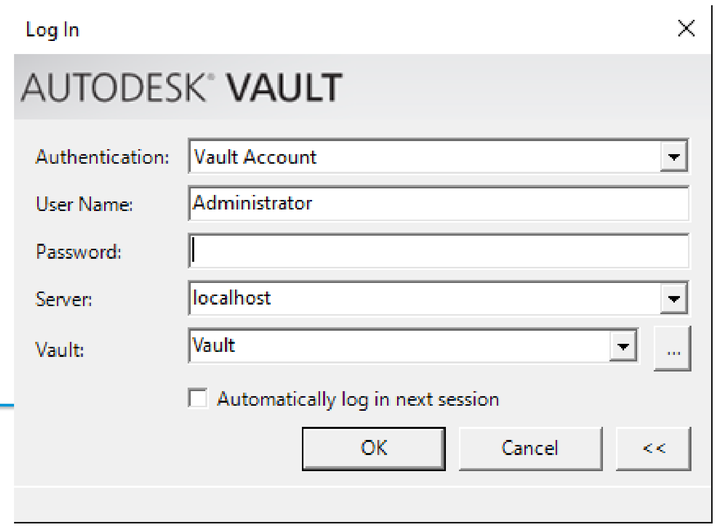
.Netプログラムからの、ForgeのREST API（OAuth, Data Management, Model Derivative, and Design Automation）の呼び出しを容易にするライブラリ

→https://github.com/Autodesk-Forge/forge-api-dotnet-client

Vaultサーバへの接続

‘Autodesk.DataManagement.Client.Framework.Vault.Forms’のLoginメソッドによりVaultサーバにログイン

```
//connection object to Vault
private static VDF.Vault.Currency.Connections.Connection _connection { get; set; }

// This shows the dialog and returns the connection and instantiates the _connection
_connection = VDF.Vault.Forms.Library.Login(null);
```



参考：GUIを表示せずにログインをする場合

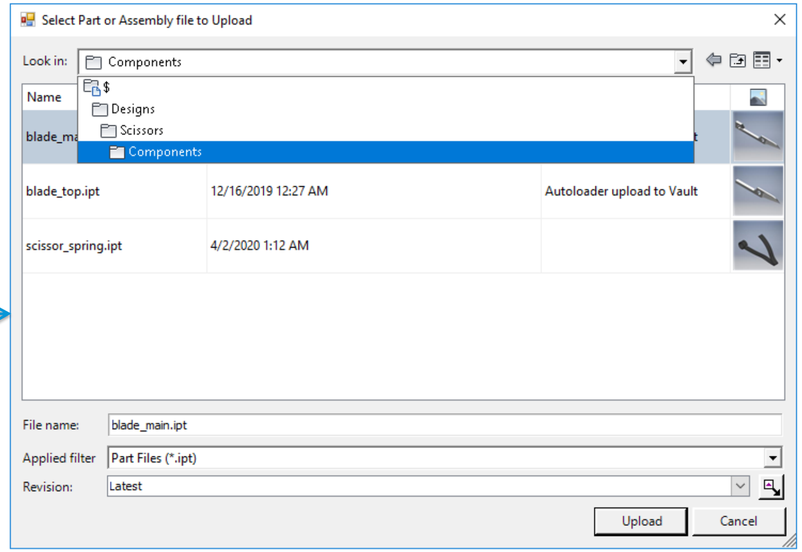
```
VDF.Vault.Results.LogInResult results = VDF.Vault.Library.ConnectionManager.LogIn(
                                "ServerName", "VaultName", "UserName", "Password",
                                 VDF.Vault.Currency.Connections.AuthenticationFlags.Standard, null);

// get the connection object to Vault
_connection = results.Connection;
```

ファイル選択ダイアログの表示

VDF.Vault.Forms.Library.SelectEntityメソッドにより、Vaultサーバから、チェックアウト対象のファイルを選択するGUIを表示

```
private static Autodesk.Connectivity.WebServices.File SelectFilefromUI()
{
    VDF.Vault.Currency.Entities.FileIteration fileIter = null;
    _connection = VDF.Vault.Forms.Library.Login(null);
    if (_connection.IsConnected)
    {
      VDF.Vault.Forms.Settings.SelectEntitySettings settings =
      new VDF.Vault.Forms.Settings.SelectEntitySettings();
      VDF.Vault.Forms.Settings.SelectEntitySettings.EntityRegularExpressionFilter[
       filters = new
       VDF.Vault.Forms.Settings.SelectEntitySettings.EntityRegularExpressionFilter[]
       {
         // Filter for only Part (ipt) files
         new VDF. Vault.Forms.Settings.SelectEntitySettings.
         EntityRegularExpressionFilter("Part Files (*.ipt)", ".+ipt",
         VDF.Vault.Currency.Entities.EntityClassIds.Files)
       };
         VDF.Vault.Forms.Controls.VaultBrowserControl.Configuration initialConfig = new
         VDF.Vault.Forms.Controls.VaultBrowserControl.Configuration(_connection,
                                    settings.PersistenceKey, null);

       // configure the columns that need to be shown
       initialConfig.AddInitialColumn(
              VDF.Vault.Currency.Properties.PropertyDefinitionIds.Server.EntityName);

       initialConfig.AddInitialColumn(
              VDF.Vault.Currency.Properties.PropertyDefinitionIds.Server.CheckInDate);

      initialConfig.AddInitialColumn(
              VDF.Vault.Currency.Properties.PropertyDefinitionIds.Server.Comment);

      initialConfig.AddInitialColumn(
              VDF.Vault.Currency.Properties.PropertyDefinitionIds.Server.ThumbnailSystem);
```

```
      // configure on how the items should be sorted in the dialog
      initialConfig.AddInitialSortCriteria(
             VDF.Vault.Currency.Properties.PropertyDefinitionIds.Server.EntityName, true);

      settings.DialogCaption = "Select Part or Assembly file to Upload";

      settings.ActionableEntityClassIds.Add("FILE");

      settings.MultipleSelect = false;

      settings.ConfigureActionButtons("Upload", null, null, false);

      settings.ConfigureFilters("Applied filter", filters, null);

      settings.OptionsExtensibility.GetGridConfiguration = e => initialConfig;

      Console.WriteLine("Launching Vault Browser...");

      VDF.Vault.Forms.Results.SelectEntityResults results =
      VDF.Vault.Forms.Library.SelectEntity(_connection, settings);

      if (results!=null)
      {
        fileIter = results.SelectedEntities.FirstOrDefault() as
              VDF.Vault.Currency.Entities.FileIteration;
      }
    }
    return fileIter; // return the selected File
}
```



Forge Authenticate APIの実行

2 Legged 認証によるTokenの取得

```
private async static Task<dynamic> GetInternalAsync()
{
      if (InternalToken == null || InternalToken.ExpiresAt < DateTime.UtcNow)
      {
        InternalToken = await Get2LeggedTokenAsync(new Scope[] { Scope.BucketCreate,
                        Scope.BucketRead, Scope.DataRead, Scope.DataCreate});
        InternalToken.ExpiresAt =
                        DateTime.UtcNow.AddSeconds(InternalToken.expires_in);
       }
     return InternalToken;
}


private async static Task<dynamic> Get2LeggedTokenAsync(Scope[] scopes)
{
       TwoLeggedApi oauth = new TwoLeggedApi();
       string grantType = "client_credentials";
       dynamic bearer = await oauth.AuthenticateAsync(
              ConsumerKey, // Client ID
              ConsumerSecret, // Client Secret
              grantType,
              scopes);
       return bearer;
}
```

Design Automation入力ファイル用のOSS Bucketの作成

Design Automation for Inventor実行時の入力ファイル（.iptファイル）を格納するOSS Bucketを作成

```
private async static Task<dynamic> CreateBucket()
{
    // bucket keys must be of the form [-_.a-z0-9]{3,128}
    string bucketKey = "inventorio" + Guid.NewGuid().ToString("N").ToLower();
    // transient bucket persist for 24 hours
    PostBucketsPayload postBucket = new PostBucketsPayload(bucketKey, null,
            PostBucketsPayload.PolicyKeyEnum.Transient);
    BucketsApi bucketsApi = new BucketsApi();
    // pass in the token that we got from the previous step
    bucketsApi.Configuration.AccessToken = InternalToken.access_token;
    dynamic newBucket = await bucketsApi.CreateBucketAsync(postBucket);
   return newBucket;
}
```

入力ファイルをOSS BucketにUpload

VaultからDownloadしたファイル（.iptファイル）を作成したOSS BucketにUpload

```
private async static Task<dynamic> UploadIptFile(string bucketKey)
{
    ObjectsApi objects = new ObjectsApi();
    objects.Configuration.AccessToken = InternalToken.access_token;
    dynamic uploadedObj = null;
    using (StreamReader streamReader = new StreamReader(new
                                            MemoryStream(filebytes)))
    {
        uploadedObj = await objects.UploadObjectAsync(bucketKey,
            inputFileName, (int)streamReader.BaseStream.Length,
            streamReader.BaseStream, "application/octet-stream");

    }
   return uploadedObj;
}
```

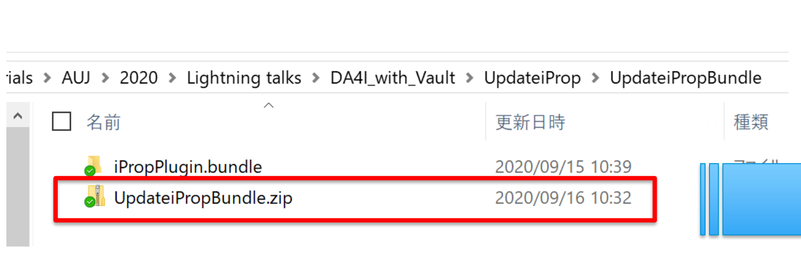
AppBundle、Activityの作成 ～ AppBundleの登録

Desing Automation for Inventorを用いて自動化する処理のひな型を定義（AppBundle=プラグインDLLを含むアプリケーションの定義と、Activity=入出力パラメータや実行するAppBundle等の定義）を作成

```
private static async Task<dynamic> CreateActivity()
 {
   Bearer bearer = (await Get2LeggedTokenAsync(new Scope[] { Scope.CodeAll
                       })).ToObject<Bearer>();
   string nickName = ConsumerKey;

   AppBundlesApi appBundlesApi = new AppBundlesApi();
   appBundlesApi.Configuration.AccessToken = bearer.AccessToken;
   PageString appBundles = await appBundlesApi.AppBundlesGetItemsAsync();
   string appBundleID = string.Format("{0}.{1}+{2}", nickName, APPNAME, ALIAS);


   if (!appBundles.Data.Contains(appBundleID))
    {
      if (!System.IO.File.Exists(LocalAppPackageZip)) return new
            Output(Output.StatusEnum.Error, "Bundle not found at " +
            LocalAppPackageZip);
       // create new bundle
      AppBundle appBundleSpec = new AppBundle(APPNAME, null, EngineName,
                                    null, null, APPNAME, null, APPNAME);
      AppBundle newApp = await
                           appBundlesApi.AppBundlesCreateItemAsync(appBundleSpec);
      if (newApp == null)
       {
          return new Output(Output.StatusEnum.Error, "Cannot create new app");
        }
      // create alias
      Alias aliasSpec = new Alias(1, null, ALIAS);
      Alias newAlias = await
      appBundlesApi.AppBundlesCreateAliasAsync(APPNAME, aliasSpec);
```

```
// upload the zip bundle
RestClient uploadClient = new
RestClient(newApp.UploadParameters.EndpointURL);
RestRequest request = new RestRequest(string.Empty, Method.POST);
request.AlwaysMultipartFormData = true;
foreach (KeyValuePair<string, object> x in
    newApp.UploadParameters.FormData)
    request.AddParameter(x.Key, x.Value);
request.AddFile("file", LocalAppPackageZip);
request.AddHeader("Cache-Control", "no-cache");
var res = await uploadClient.ExecuteTaskAsync(request);
}
```

UpdateiPropビルド時に作成されたzipを、AppBundleとしてForgeに登録



AppBundle、Activityの作成 ～Activityの作成

Desing Automation for Inventorを用いて自動化する処理のひな型を定義（AppBundle=プラグインDLLを含むアプリケーションの定義と、Activity=入出力パラメータや実行するAppBundle等の定義）を作成

```
    ActivitiesApi activitiesApi = new ActivitiesApi();
    activitiesApi.Configuration.AccessToken = bearer.AccessToken;
    PageString activities = await activitiesApi.ActivitiesGetItemsAsync();
    string activityID = string.Format("{0}.{1}+{2}", nickName, ACTIVITY_NAME, ALIAS);
    if (!activities.Data.Contains(activityID))
    {
     // create activity
     string commandLine =
                              string.Format(@"$(engine.path)\\inventorcoreconsole.exe /i
                             $(args[InputIPT].path) /al $(appbundles[{0}].path)", APPNAME);
     ModelParameter iptFile = new ModelParameter(false, false,
                            ModelParameter.VerbEnum.Get, "Input Ipt File", true, inputFileName);
     ModelParameter result = new ModelParameter(false, false,
                             ModelParameter.VerbEnum.Put, "Resulting Ipt File", true,
                             inputFileName);

    Activity activitySpec = new Activity(new List<string> { commandLine },
                                 new Dictionary<string, ModelParameter>()
                                 {{ "InputIPT", iptFile },{ "ResultIPT",result},
                                 EngineName,
                                 new List<string>() { string.Format("{0}.{1}+{2}", nickName, APPNAME
                                 null,ACTIVITY_NAME, null, ACTIVITY_NAME
                                );
    Activity newActivity = await activitiesApi.ActivitiesCreateItemAsync(activitySpec);
    Alias aliasSpec = new Alias(1, null, ALIAS);
    Alias newAlias = await activitiesApi.ActivitiesCreateAliasAsync(ACTIVITY_NAME, aliasSpe
  }
  return new Output(Output.StatusEnum.Sucess, "Activity created");
}
```

WorkItemの作成と実行

作成したActivityに具体的なパラメータを指定して、WorkItemを作成し実行

```
private static async Task<dynamic> CreateWorkItem(String bucketkey)
{
string nickName = ConsumerKey;
Bearer bearer = (await Get2LeggedTokenAsync(new Scope[] { Scope.CodeAll
                    })).ToObject<Bearer>();
  // path to the input ipt file
string downloadUrl = string.Format("https://developer.api.autodesk.
                  com/oss/v2/buckets/{0}/objects/{1}", bucketkey, inputFileName);
 // path to output file.
string uploadUrl = string.Format("https://developer.api.autodesk.
                com/oss/v2/buckets/{0}/objects/{1}", bucketkey, inputFileName);
JObject iptFile = new JObject {
                        new JProperty("url", downloadUrl),
                        new JProperty("headers",
                                new JObject{
                                    new JProperty("Authorization", "Bearer " +
                                        InternalToken.access_token)
                                    })
                       };
JObject resultIpt = new JObject  {
                                    new JProperty("verb", "put"),
                                    new JProperty("url", uploadUrl),
                                    new JProperty("headers",
                                    new JObject{
                                            new JProperty("Authorization", "Bearer " +
                                            InternalToken.access_token)
                                             })
                                   };
```

```
WorkItem workItemSpec = new WorkItem(
            null, string.Format("{0}.{1}+{2}", nickName, ACTIVITY_NAME, ALIAS),
            new Dictionary<string, JObject>()
            {{ "InputIPT",  iptFile },{ "ResultIPT", resultIpt  }}, null);
WorkItemsApi workItemApi = new WorkItemsApi();
workItemApi.Configuration.AccessToken = bearer.AccessToken;
WorkItemStatus newWorkItem = await
workItemApi.WorkItemsCreateWorkItemsAsync(null, null, workItemSpec);

 for (int i = 0; i < 1000; i++)
 {
    System.Threading.Thread.Sleep(1000);
    WorkItemStatus workItemStatus = await
    workItemApi.WorkItemsGetWorkitemsStatusAsync(newWorkItem.Id);
    if (workItemStatus.Status == WorkItemStatus.StatusEnum.Pending ||
        workItemStatus.Status == WorkItemStatus.StatusEnum.Inprogress) continue;
    break;
 }
 // check in the output file back into Vault.
 await CheckintoVault(uploadUrl);
 return new Output(Output.StatusEnum.Sucess, "Activity created");
}
```

Design Automationの実行結果をVaultにチェックイン

Design Automationの実行結果を取得して、Vaultにチェックイン

```
public static async Task<dynamic> CheckintoVault(string url)
 {
     IRestClient client = new RestClient("https://developer.api.autodesk.com/");
     RestRequest request = new RestRequest(url, Method.GET);
     request.AddHeader("Authorization", "Bearer " + InternalToken.access_token);
     request.AddHeader("Accept-Encoding", "gzip, deflate");
     IRestResponse response = await client.ExecuteTaskAsync(request);
     if (response.StatusCode != System.Net.HttpStatusCode.OK)
     {
      return new Output(Output.StatusEnum.Error, "Not able to download to local drive");
     }
     else {
      VaultUtil.CheckinFileStream(response.RawBytes);
      return new Output(Output.StatusEnum.Sucess, "Checked into Vault
                    successfully");
      }
  }
```

まとめ

- Vault APIとForge Design Automationを組み合わせるカスタムアプリケーションについて、処理の手順と使用しているAPIを解説