

D365FOのIF方式

目次

1. はじめに
2. Interface方式
 - 2-1. Flow
 - 2-2. Common data service (CDS)
 - 2-3. BYOD
 - 2-4. Recurring Integration
 - 2-5. その他ETLツール
3. Inbound Interface C#からのアクセス (Custom Service)
4. Outbound Interface

1. はじめに

1. はじめに

本番環境のデータベースはMicrosoft社管理の為
アクセスすることはできません。

ETLツール
PL/SQL、ストアドプロシージャ等による
ODBC接続



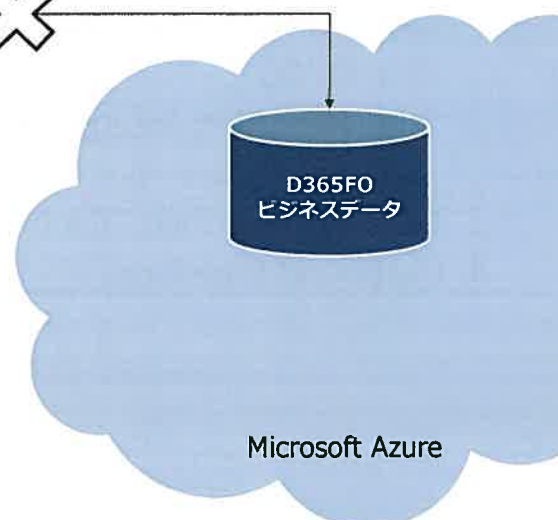
ユーザー様の要件として、以下の様な要件がある場合があります。

【入力】

- ・ 他システムで登録した顧客マスタ情報を連携
- ・ ハンディターミナルで登録した入在庫情報を連携

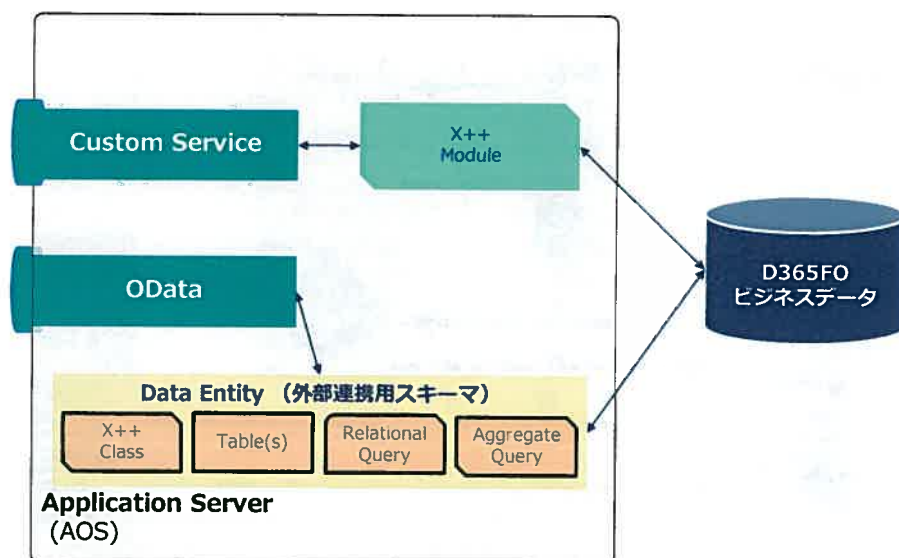
【出力】

- ・ 外部システムへD365の在庫情報を連携



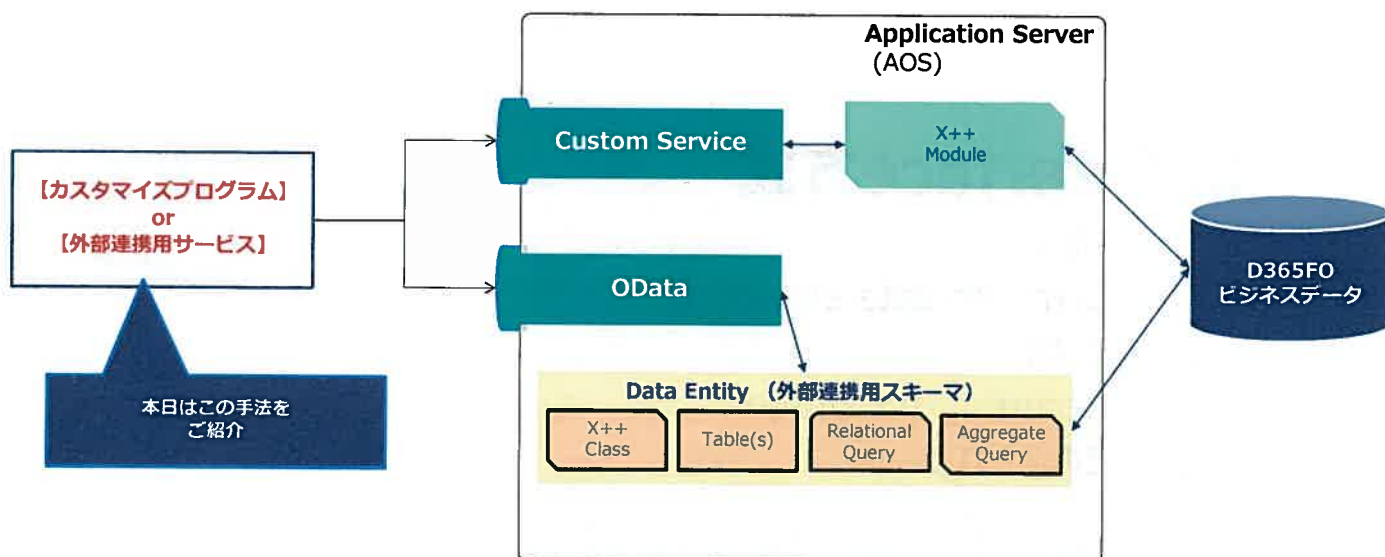
1. はじめに

D365FOのデータへアクセスする為に、
受口を用意しています。



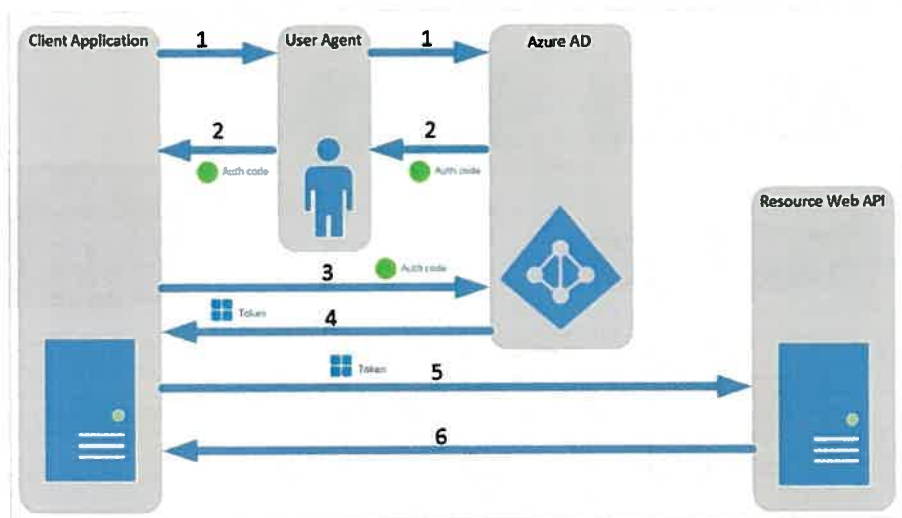
1. はじめに

その受口を介して、D365FOのビジネスデータへのアクセスを行います。
アクセス方法には様々な手法があります。それを紹介致します。



1. はじめに

D365FOのプログラムを実行するためには、Azure ADの認証認可が必要です。
具体的には、ユーザーID、パスワード、アプリケーションキー、（必要に応じてシークレットキー）が必要になります。

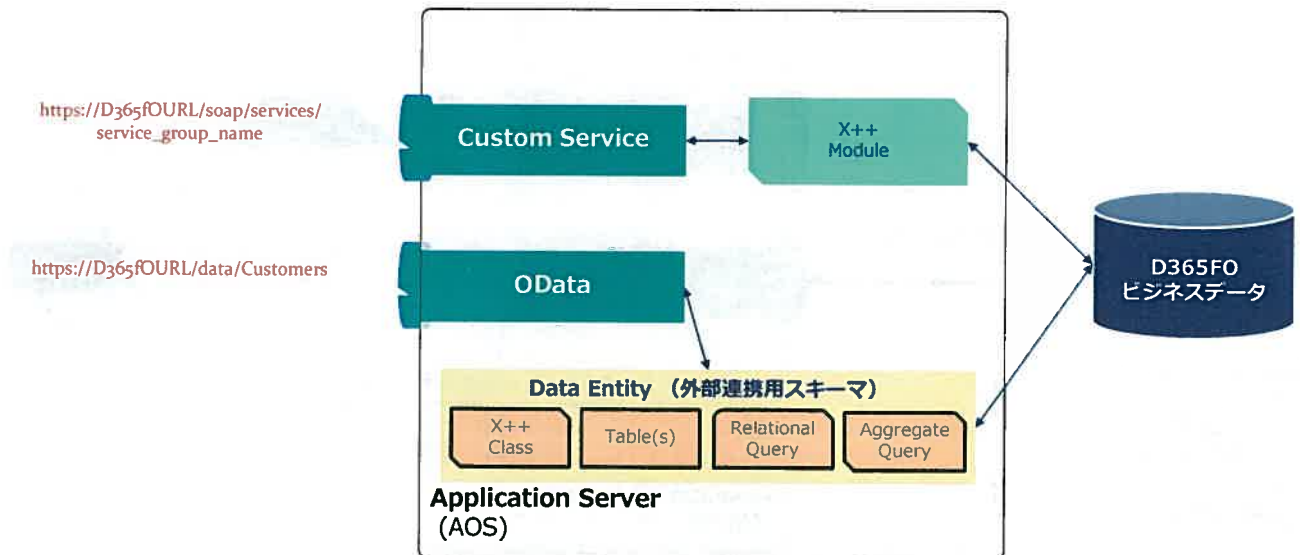


2. Interface方式

- 2-1. Flow
- 2-2. Common data service (CDS)
- 2-3. BYOD
- 2-4. Recurring Integration
- 2-5. その他ETLツール

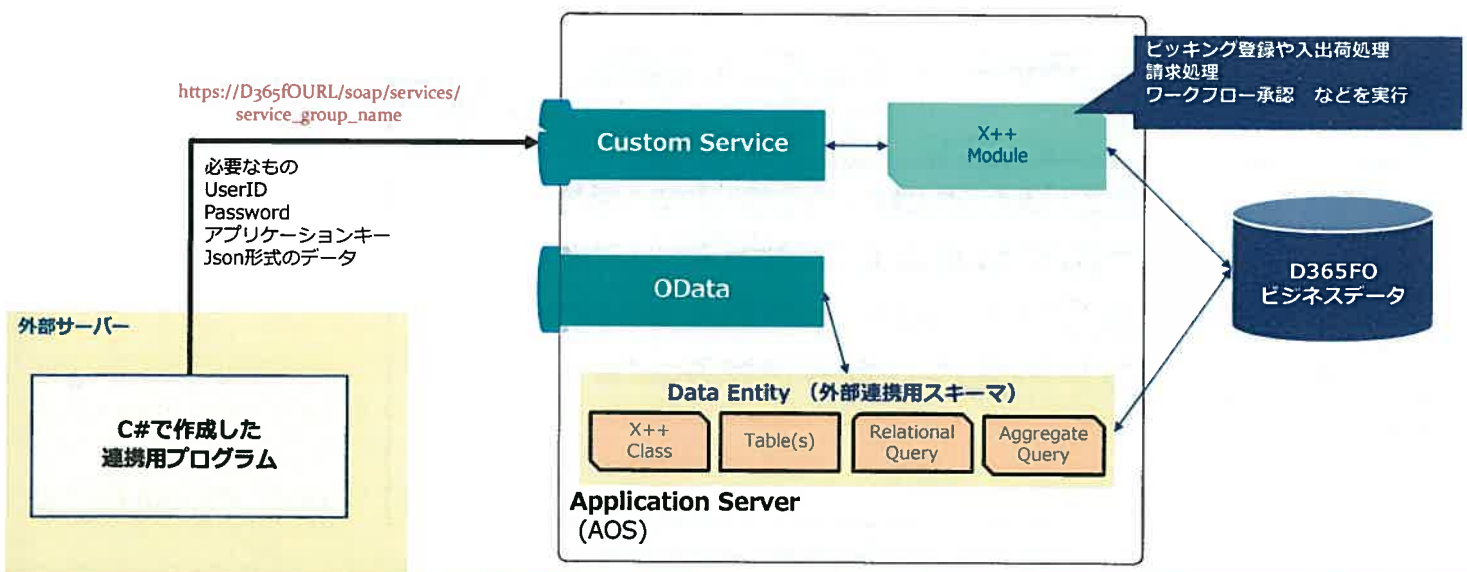
2. Interface方式

外部連携用の受口はURLで公開されています。
自分で作成する事も可能です。



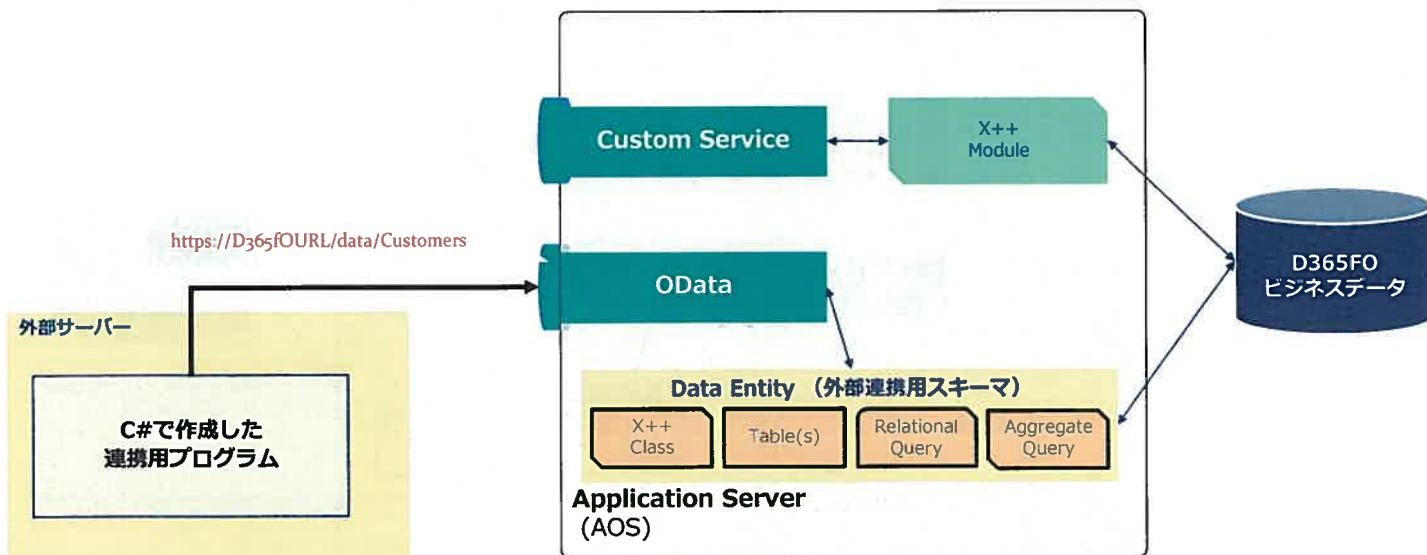
2. Interface方式

C#でアクセスが可能です。
従来のBusiness connectorと同じようなイメージとなります。



2. Interface方式

ODataのURLを指定すれば、jsonフォーマットでデータを取得する事ができます。



2. Interface方式

ODataのURLを指定すれば、jsonフォーマットでデータを取得する事ができます。



2. Interface方式

OData、Custom Serviceはそれぞれ以下の特徴があります。

OData

- ✓ X++のコーディング無しでCRUDが可能
- ✓ 提供されているData Entityが多数
- ✓ DIXFで利用するData Entityと同じで理解がしやすい
- ✓ 外部からの単純なCRUDに利用

Custom Service

- ✓ X++のコーディングを行う事で、自由に処理が可能
- ✓ 販売注文の転記処理（出荷、売上計上）などの実装が可能
- ✓ 外部からバッチ処理の実行等を利用

2-1. Flow

FlowとD365FOをつなげ、データの入出力が可能です。

Microsoft Flow アプリやサービス間のオートメーションを行うワークフロー



通知/Notification



ファイル同期



データ収集

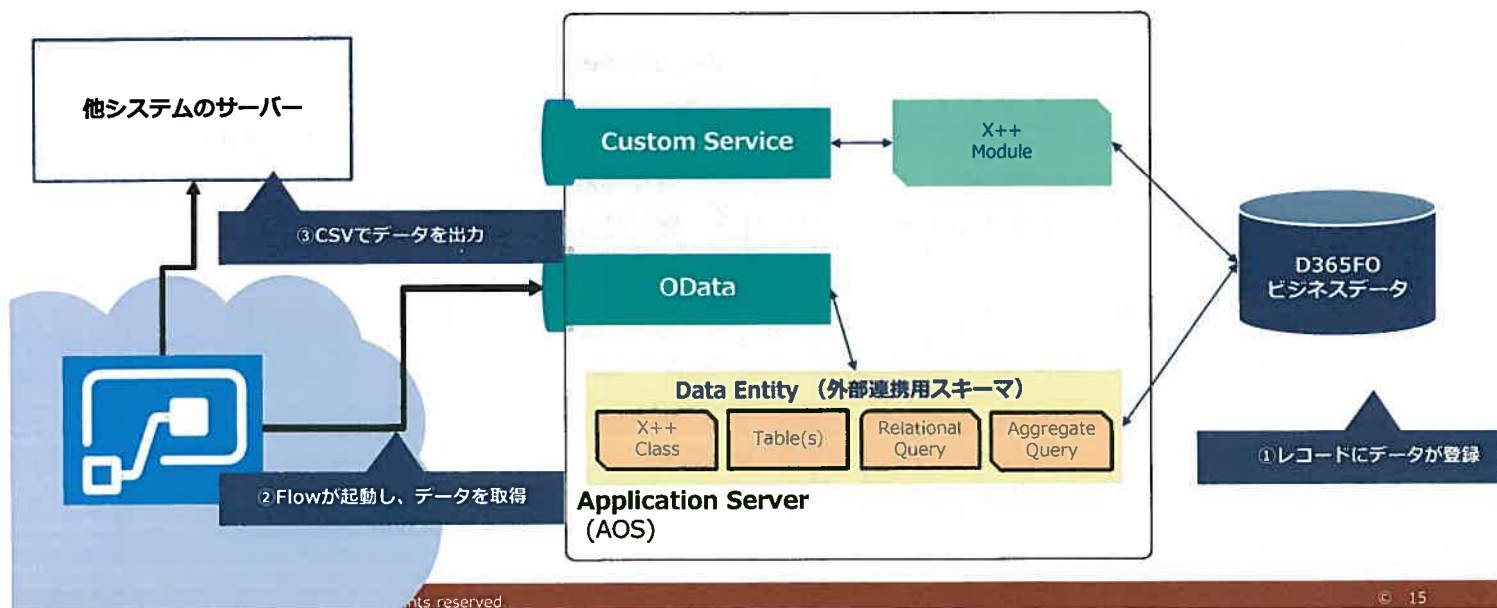


自動承認

<https://flow.microsoft.com/ja-jp/>

2-1. Flow

ODataのURLを指定すれば、jsonフォーマットでデータを取得することができます。



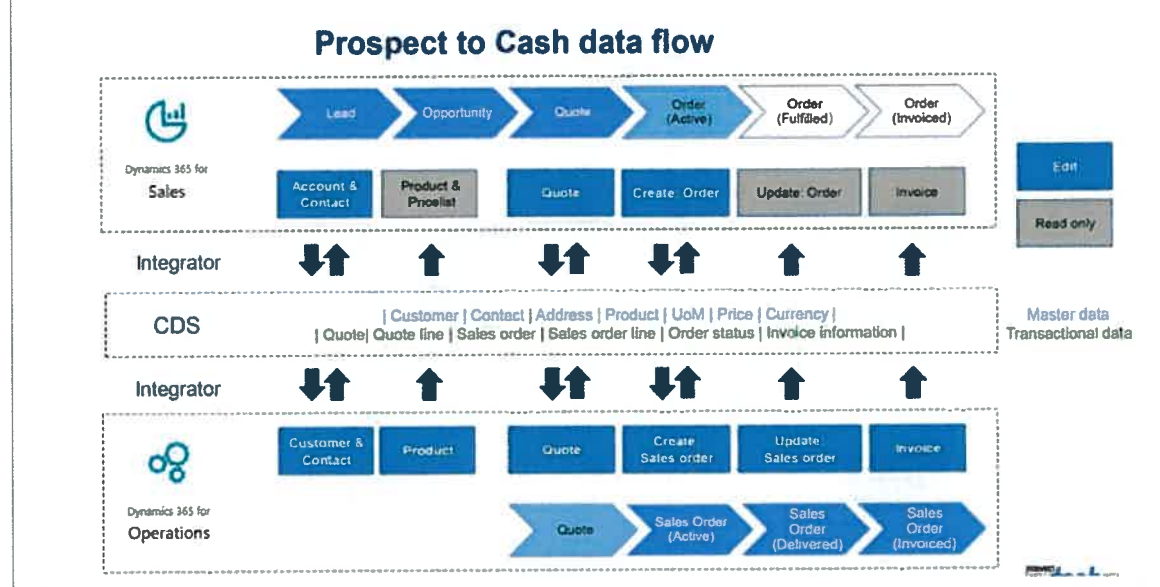
2-1. Flow

Flowから接続先インスタンスとエンティティを選択し、データをCSVに出力するなどがGUIからの実装が可能です。



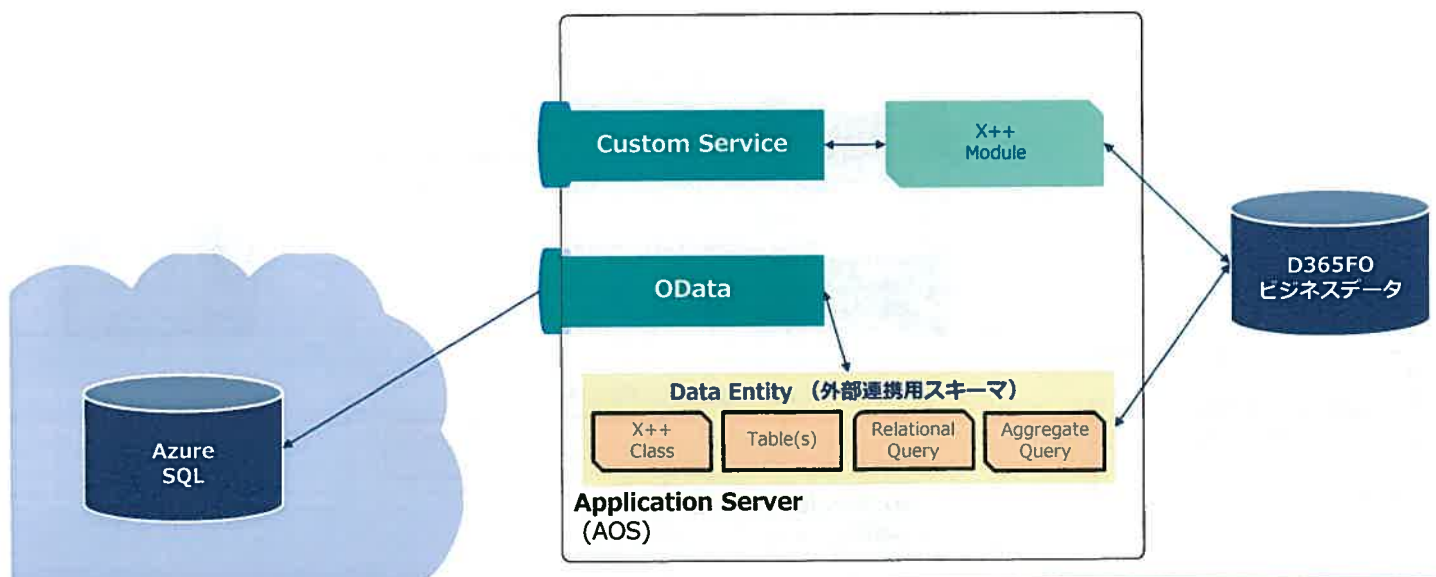
2-2. Common data service

Common data service

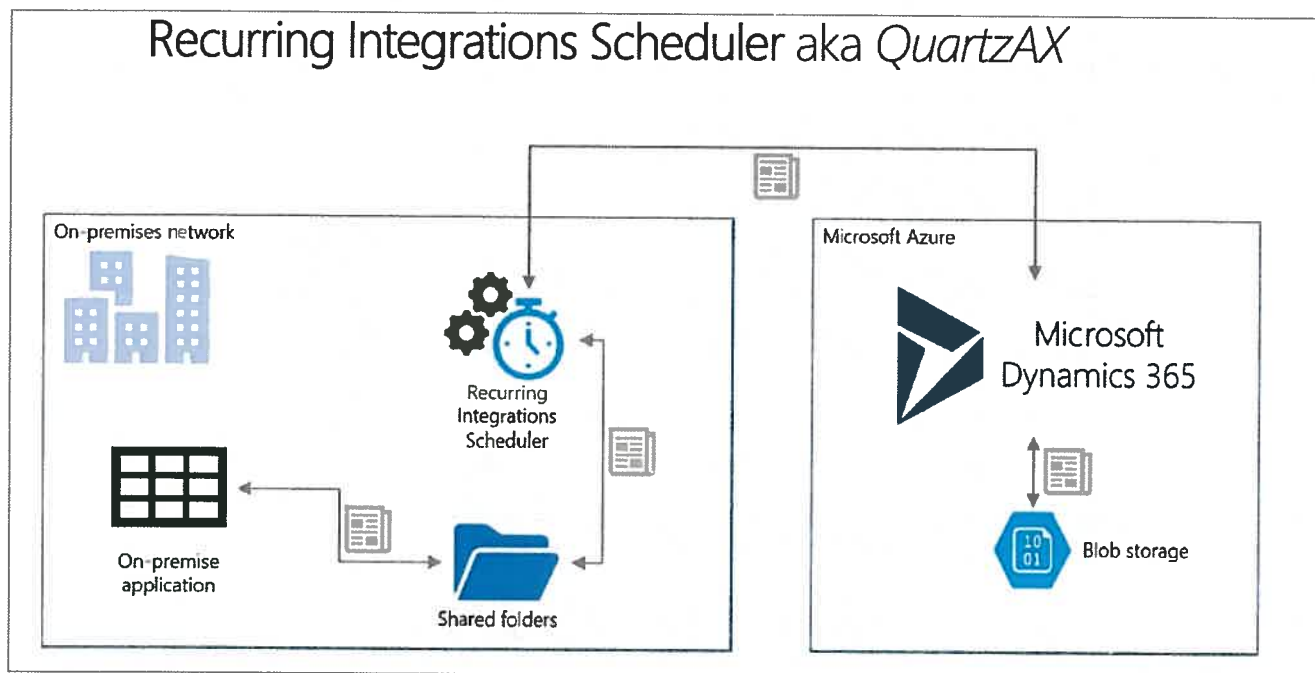


2-3. BYOD

Azure SQLデータベースに直接連携する機能です。
Azure SQLはD365FOとは別に契約が必要です。 非同期の大量データのコピーに適しています。



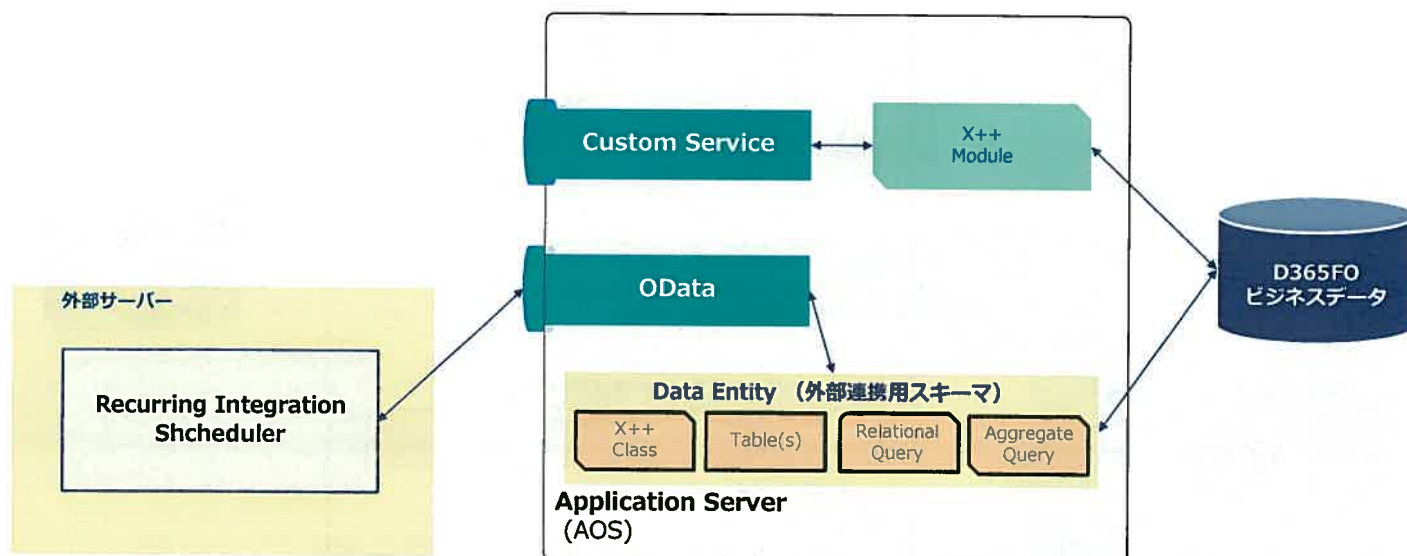
2-4. Recurring Integration



2-4. Recurring Integration

Entityに対してCSVファイルのINOUTを行うツールです。

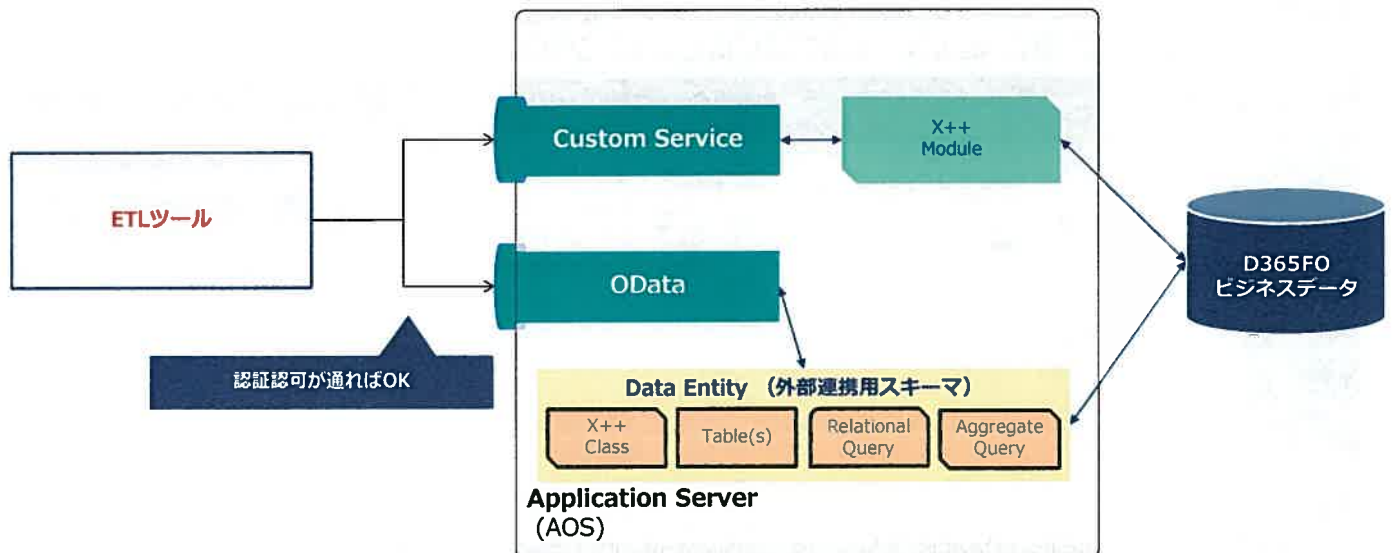
<https://docs.microsoft.com/en-us/dynamics365/unified-operations/dev-itpro/data-entities/recurring-integrations?toc=/fin-and-ops/toc.json>



2-5. その他ETLツール

ODataやCustom Servicesを利用すれば、ETLツールを利用してのデータの受け渡しも可能です。

※oAuth 2.0の対応が必要になります。

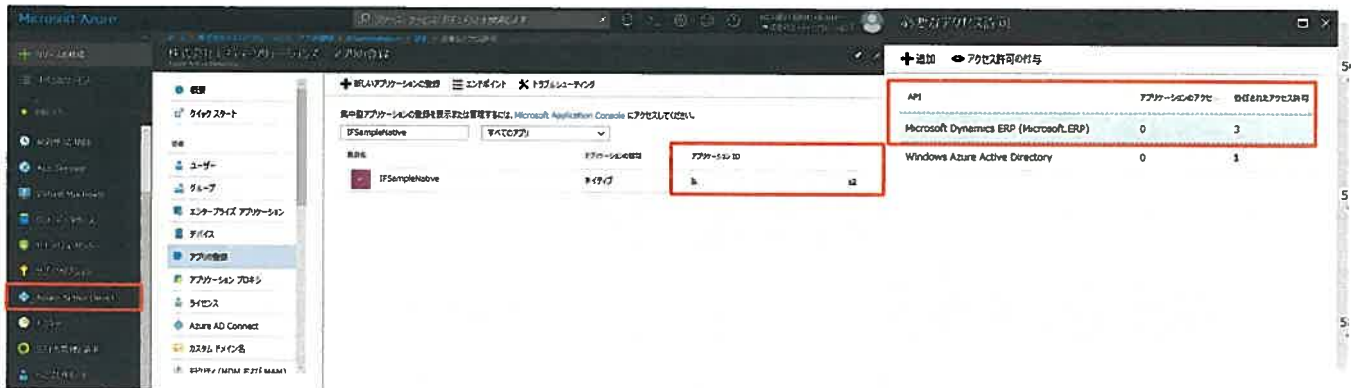


3. Inbound Interface C#からのアクセス (Custom Service)

3. Inbound Interface C#からのアクセス (Custom Service)



Azure PortalからAzure Active Directoryの設定よりアプリ登録を行い、アプリケーションIDの発行、および、そのアプリに対してアクセス許可を設定します。



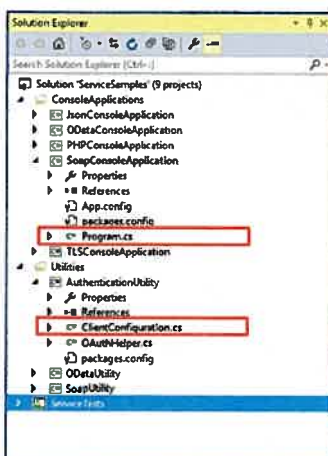
参考URL：

<https://docs.microsoft.com/en-us/dynamics365/unified-operations/dev-itpro/data-entities/services-home-page>

3. Inbound Interface C#からのアクセス (Custom Service)



Visual Studioよりプロジェクト「ServiceSamples」を開きます。
今回メインで利用するプログラムは以下の赤枠のプログラムが該当します。



SoapConsoleApplication > Program.cs (メイン処理が実行されるプログラム)

AuthenticationUtility > ClientConfiguration.cs (接続先情報を保持)

3. Inbound Interface C#からのアクセス (Custom Service)

事前準備

C#
プログラム

ClientConfiguration.cs より、接続先情報を書き換えます。

```
namespace AuthenticationUtility
{
    24 references
    public partial class ClientConfiguration
    {
        16 references | 0/2 passing
        public static ClientConfiguration Default { get { return ClientConfiguration.OneBox; } }

        public static ClientConfiguration OneBox = new ClientConfiguration()
        {
            UriString = "https://surveys92ec383263177127a0s.cloudax.dynamics.com/",
            UserName = "nts0101@nts01.0nmicrosoft.com",
            Password = " ",
            //SOAP
            ActiveDirectoryClientAppId = "b3abe660-b999-418b-81d2-18d92d0a99e2",
            ActiveDirectoryClientAppSecret = "",
            ActiveDirectoryResource = "https://surveys92ec383263177127a0s.cloudax.dynamics.com",
            ActiveDirectoryTenant = "https://login.microsoftonline.com/nts01.0nmicrosoft.com",

            //ODat
            //ActiveDirectoryClientAppId = "f6880549-8203-4fe6-8107-b2eb4e6fc618",
            //ActiveDirectoryClientAppSecret = "IqQKwFE2rW/Hjagutp/5uBIAkZc/ktCFLhXHz3hKso=",

            // Change TLS version of HTTP request from the client here
            // Ex: TLSVersion = "1.2"
            // Leave it empty if want to use the default version
            TLSVersion = "",
        };
    }
}
```

- 接続先情報
- ユーザーID
- パスワード
- AzureActiveDirectoryのアドレス
- アプリケーションキー

3. Inbound Interface C#からのアクセス (Custom Service)

事前準備

C#
プログラム

Program.cs より、どのサービス (X++のプログラム) を呼び出すか接続先情報を書き換えます。

```
public const string UserSessionServiceName = "UserSessionService";

[STAThread]
0 references
static void Main(string[] args)
{
    var aosUriString = ClientConfiguration.Default.UriString;

    var oAuthHeader = OAuthHelper.GetAuthenticationHeader();
    var serviceUriString = SoapUtility.SoapHelper.GetSoapServiceUriString(UserSessionServiceName, aosUriString);

    var endpointAddress = new System.ServiceModel.EndpointAddress(serviceUriString);
    var binding = SoapUtility.SoapHelper.GetBinding();

    var client = new UserSessionServiceClient(binding, endpointAddress);
    var channel = client.InnerChannel;

    UserSessionInfo sessionInfo = null;

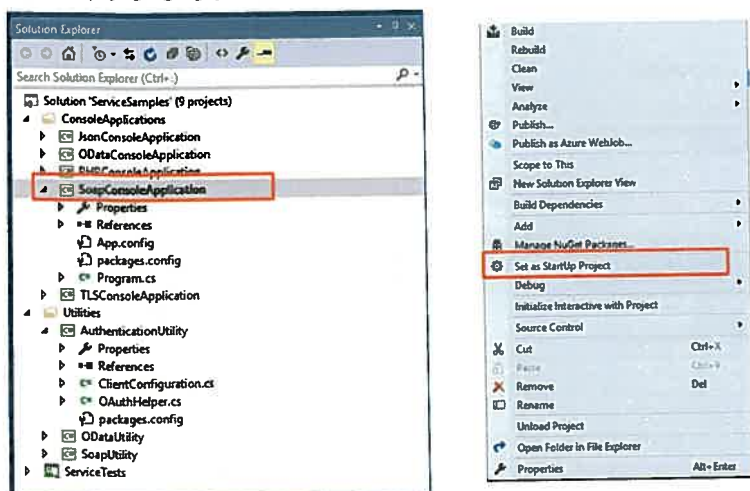
    using (OperationContextScope operationContextScope = new OperationContextScope(channel))
    {
        HttpRequestMessageProperty requestMessage = new HttpRequestMessageProperty();
        requestMessage.Headers[OAuthHelper.OAuthHeader] = oAuthHeader;
        OperationContext.Current.OutgoingMessageProperties[HttpRequestMessageProperty.Name] = requestMessage;
        sessionInfo = ((UserSessionService)channel).GetUserSessionInfo(new GetUserSessionInfo()).result;
    }
}
```

- サービスURL

3. Inbound Interface C#からのアクセス (Custom Service)



SoapConsoleApplicationを右クリック > Set as StartUp Objectをクリックした後、Startボタンよりプログラムを実行します。



3. Inbound Interface C#からのアクセス (Custom Service)



サービスが実行され、コンソール画面に結果が表示されることを確認します。



4. Outbound Interface

4. Outbound Interface

D365FOから外部のWebサービスをCallする事が可能です。

【作成手順概要】

1. C#でClass Libraryプロジェクトを作成。
Webサービスを指定したプログラムを作成後、DLLを発行
2. D365FO側でDLLを参照に追加
3. X++でCall

■ 参考記事

<https://docs.microsoft.com/en-us/dynamics365/unified-operations/dev-itpro/data-entities/consume-external-web-service?toc=/fin-and-ops/toc.json>

