



クラウド開発 & Exchange Apps 公開セミナー

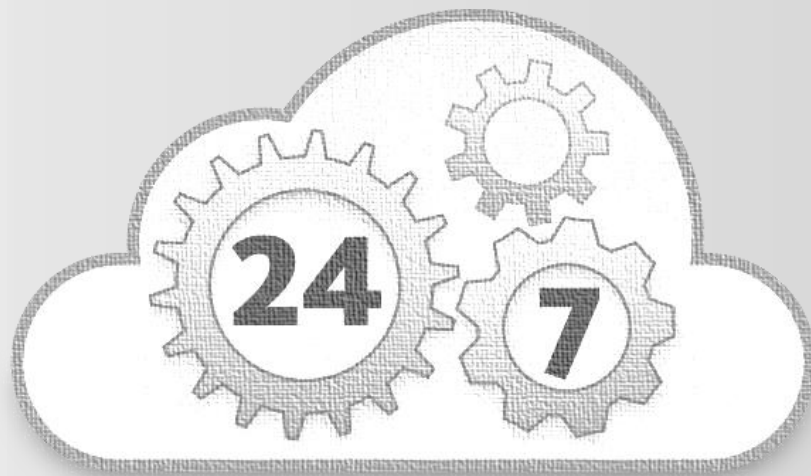
クラウドを使ったアプリケーション例

伊勢崎 俊明

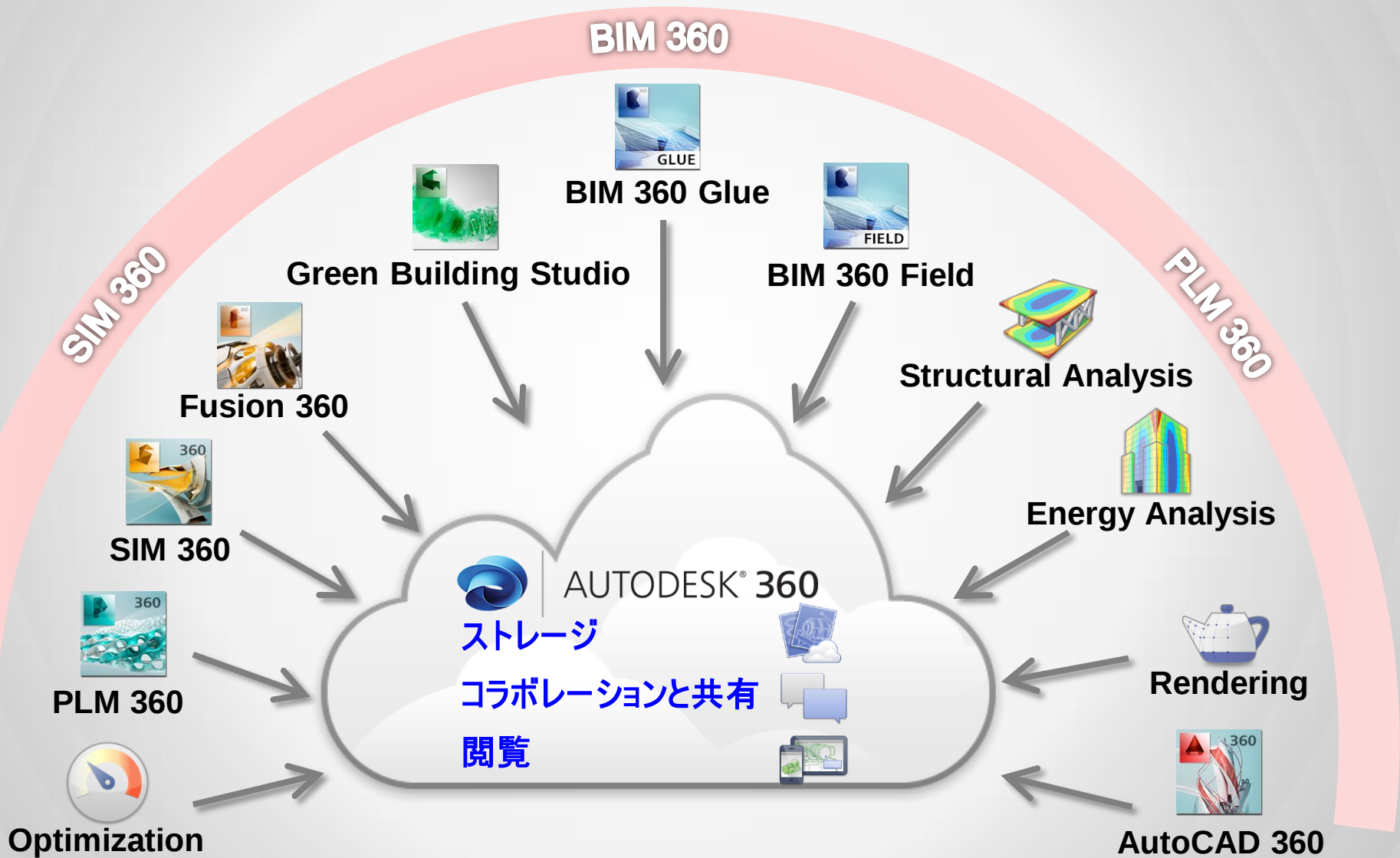
Developer Technical Services



クラウド&モバイルの活用



Autodesk 360 クラウド サービス



クラウドを利用した設計環境へのメリット

1. いつでも、どこからでも図面・設計図書にアクセス

- クラウドに保存することでデータ中心の設計環境を実現

2. スマートフォン、タブレットなどのモバイル デバイスの活用が可能

- 印刷出力した紙図面の持ち出しが不要、あるいは大幅低減

3. 複数の設計者間で図面の共有やオンライン コラボレーションを実現

- 明示的に指定したメンバでクラウド上の図面を同時に閲覧、協調編集

4. クラウドリソースを集中利用してデスクトップPCを演算負荷から解放

- 高負荷なレンダリングや解析演算をクラウドに代替させて手元の作業を継続

クラウドインフラを提供する代表的な企業

- サービスベンダーに物理的なハードウェアを提供
 - 独自の API を公開してサービスの構築をサポート
 - 多くが世界各地に大規模データセンターを持つ

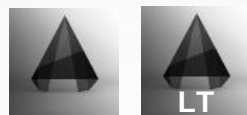


クラウドにまつわる用語

データセンター 密結合 シンククライアント IaaS
サービス指向アーキテクチャ 仮想化 データベース サービス
HTTP ホスティング サービス プライベートクラウド
ストレージ サービス Web サービス グリッドコンピューティング
疎結合 SOAP REST オンプレミス システム
ファイル同期型サービス 非同期
分散システム インターネット
アプリケーション サービス ファイル共有サービス
PaaS SaaS
ハイブリッドクラウド パブリッククラウド

AutoCAD 360 へのアクセス ポイント

- 従来通り 3 つの方法でアクセスできます



デスクトップ製品から



AutoCAD 360 Web

Web ブラウザから



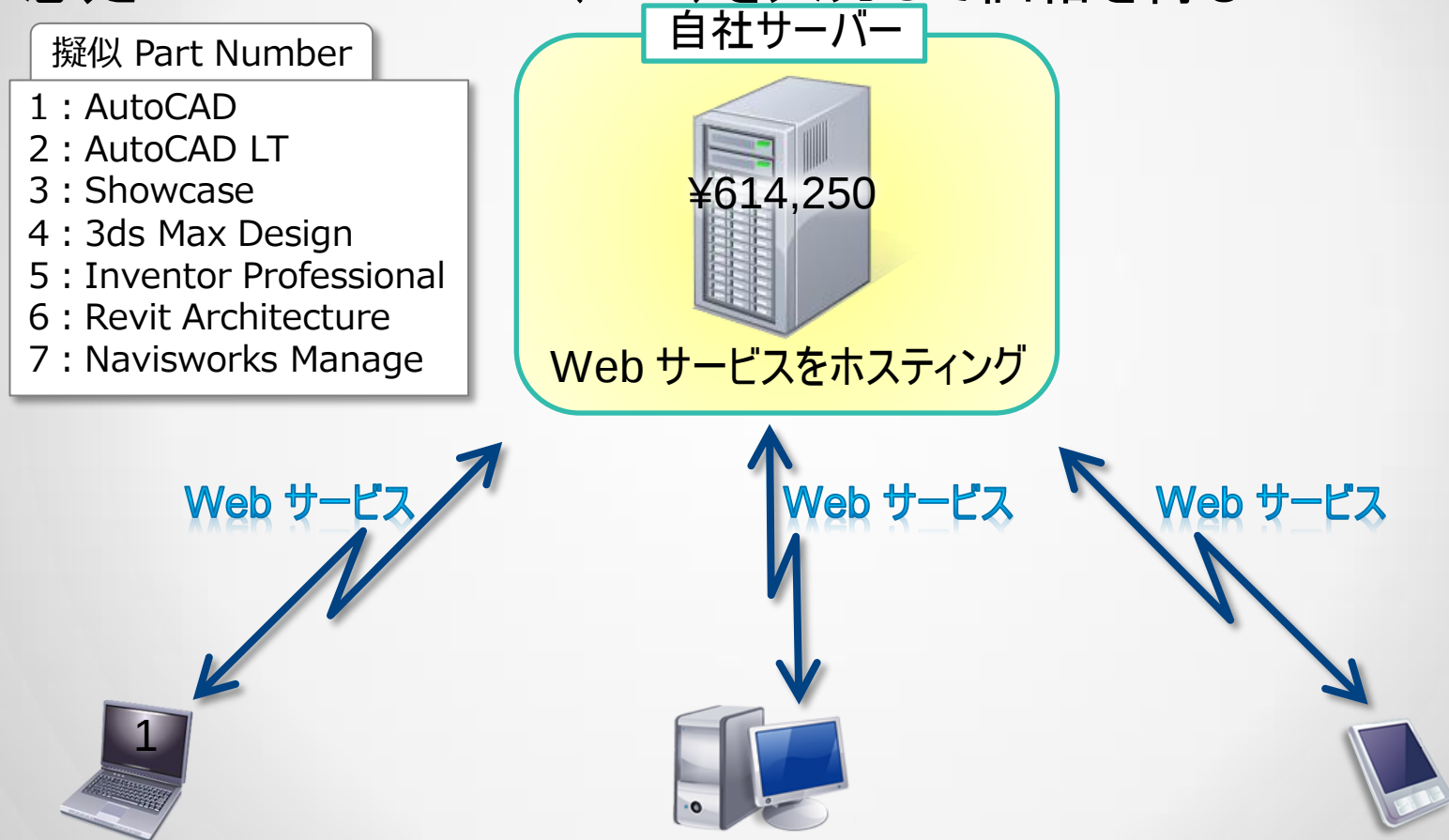
AutoCAD 360 Mobile

Mobile アプリから



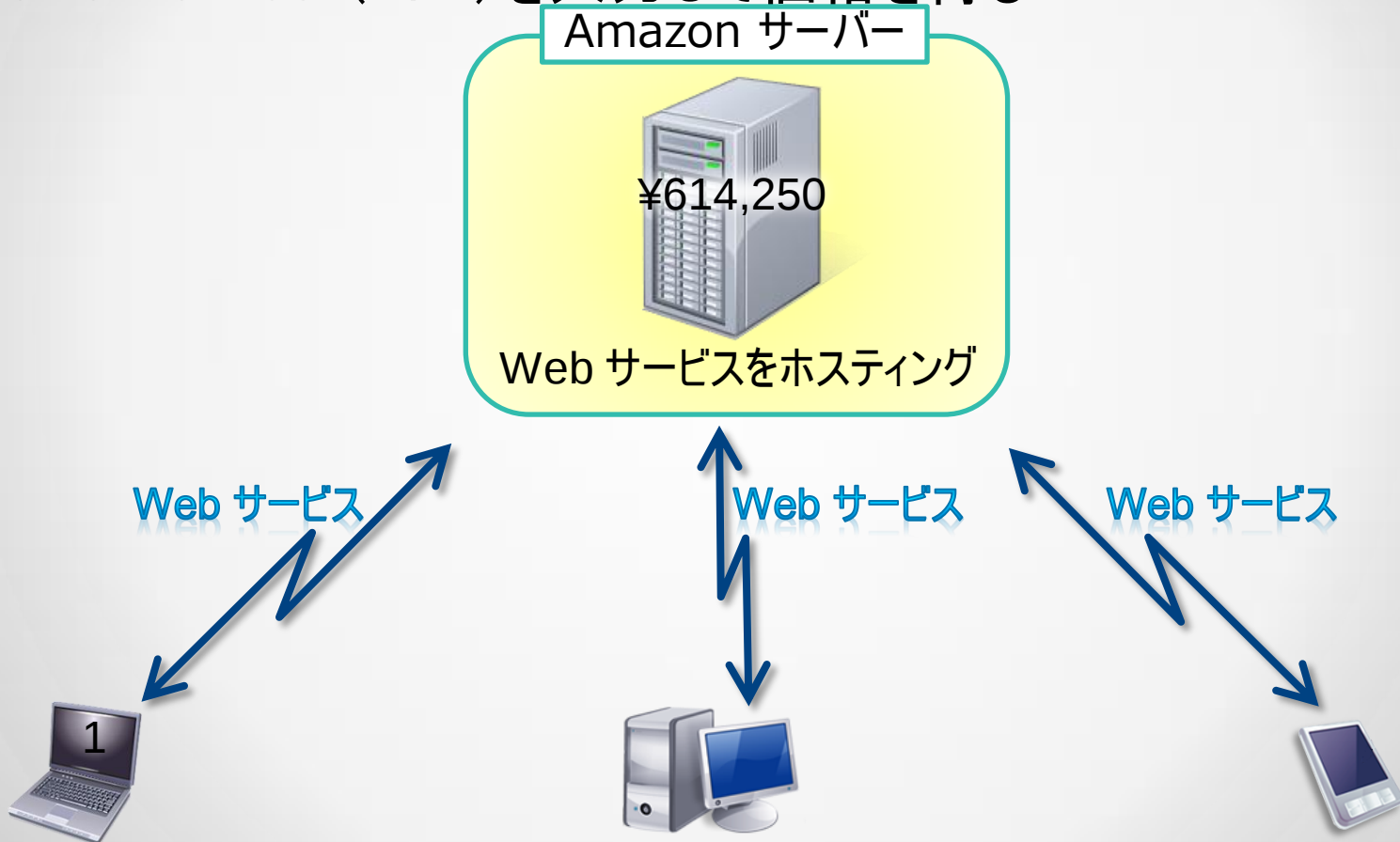
① ローカル Web サイトにホストした Web サービス

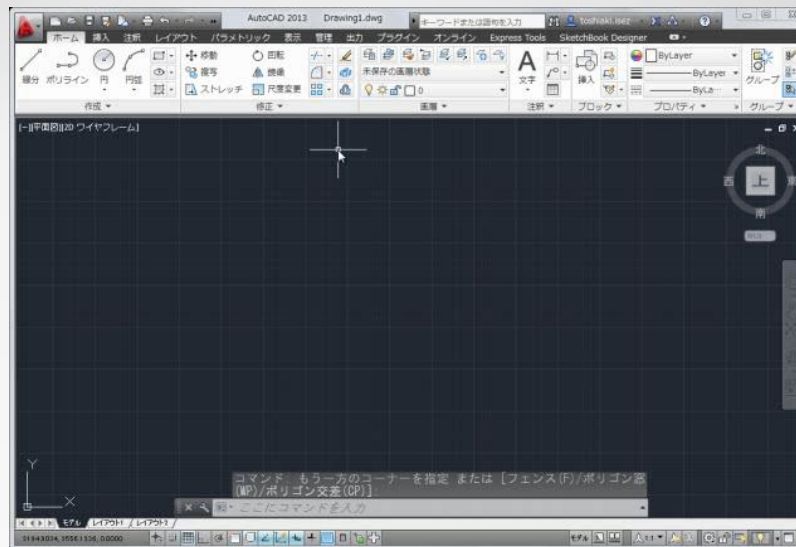
- 想定：Parts Number (P/N) を入力して価格を得る



② AWS仮想サーバー上に構築した Web サイトにホストした Web サービス

- Parts Number (P/N) を入力して価格を得る



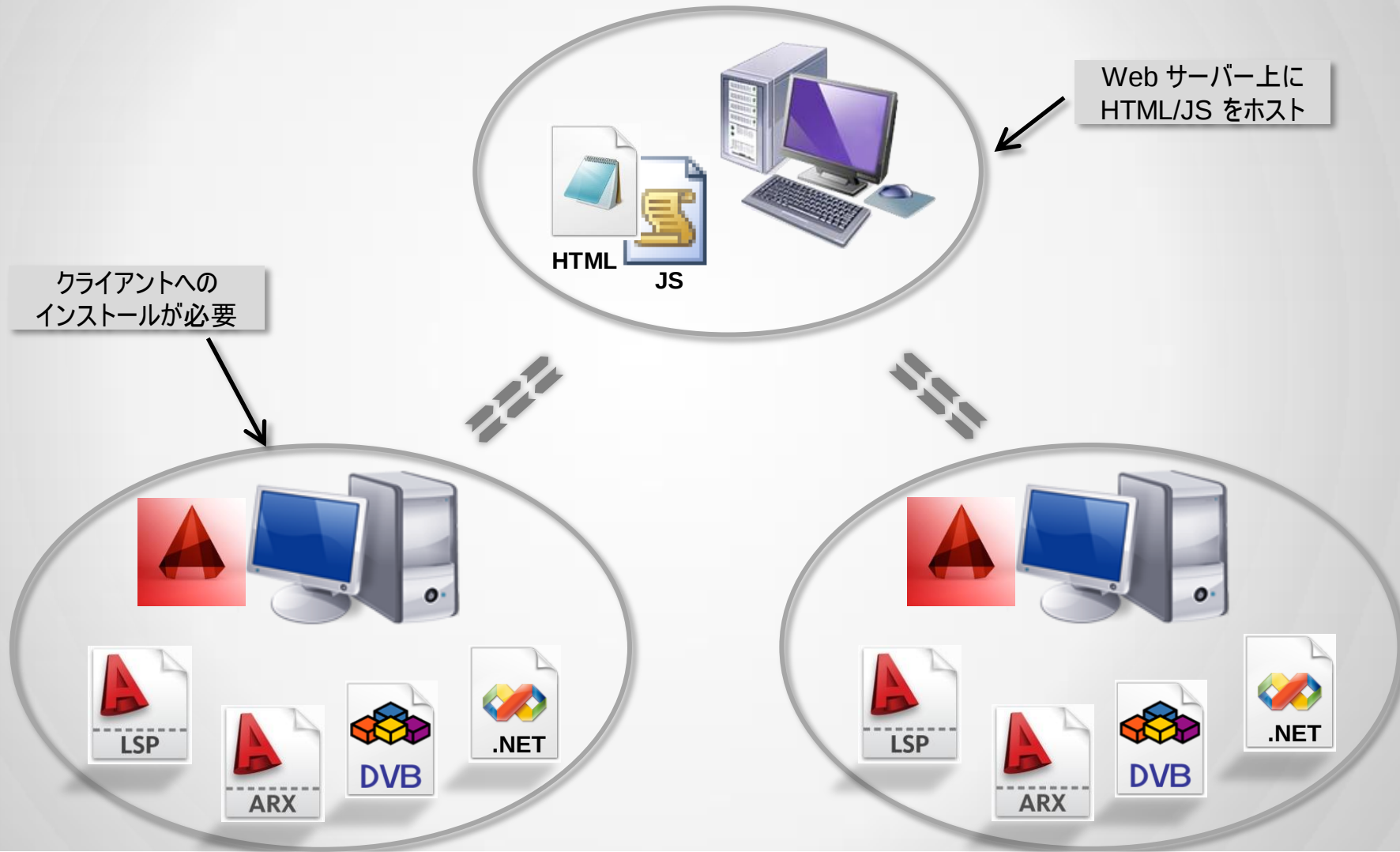


参考: 実際の Web サービス例

- Web サービス API の例
 - じゃらん Web サービス:
<http://www.jalan.net/jw/jwp0000/jww0001.do>
 - カーセンサーnet Webサービス:
<http://web-service.recruit.co.jp/carsensor/>
 - Google Maps API: <https://developers.google.com/maps/>
 - GeOAP: <http://www.geoap.jp/>
- Web サービス API を持つオートデスク製品
 - AutoCAD BIM 360 Gule
 - ...

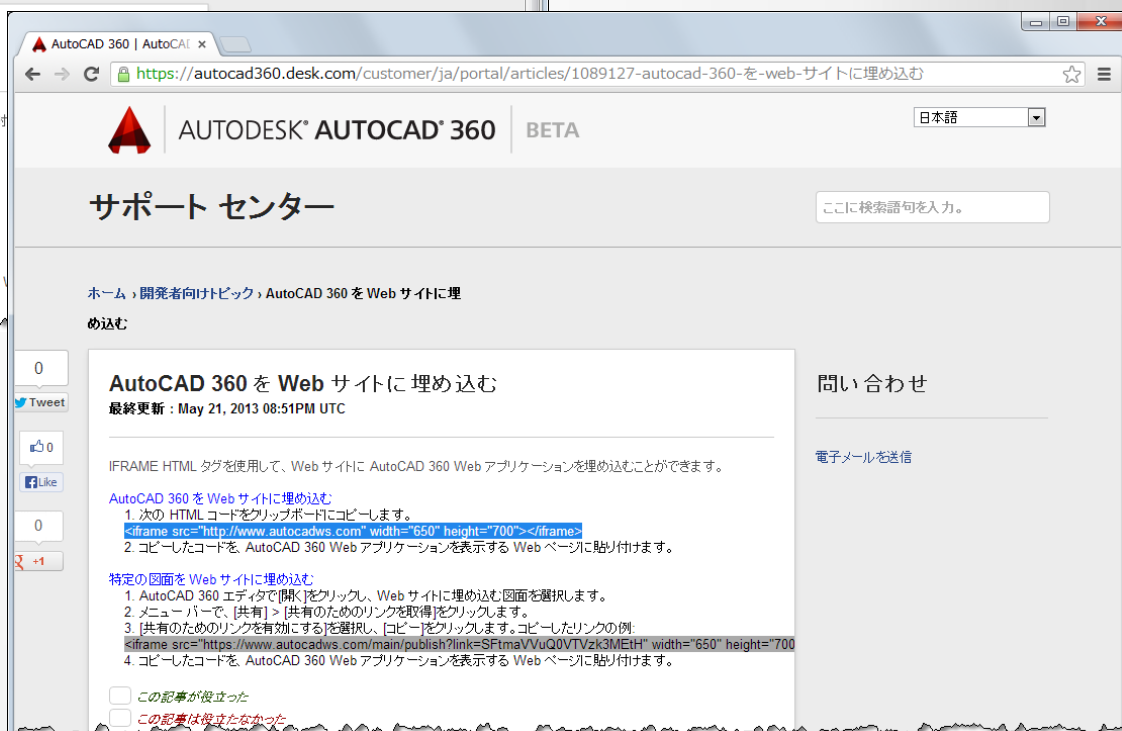
クラウドとモバイル – AutoCAD JavaScript API

サンプル

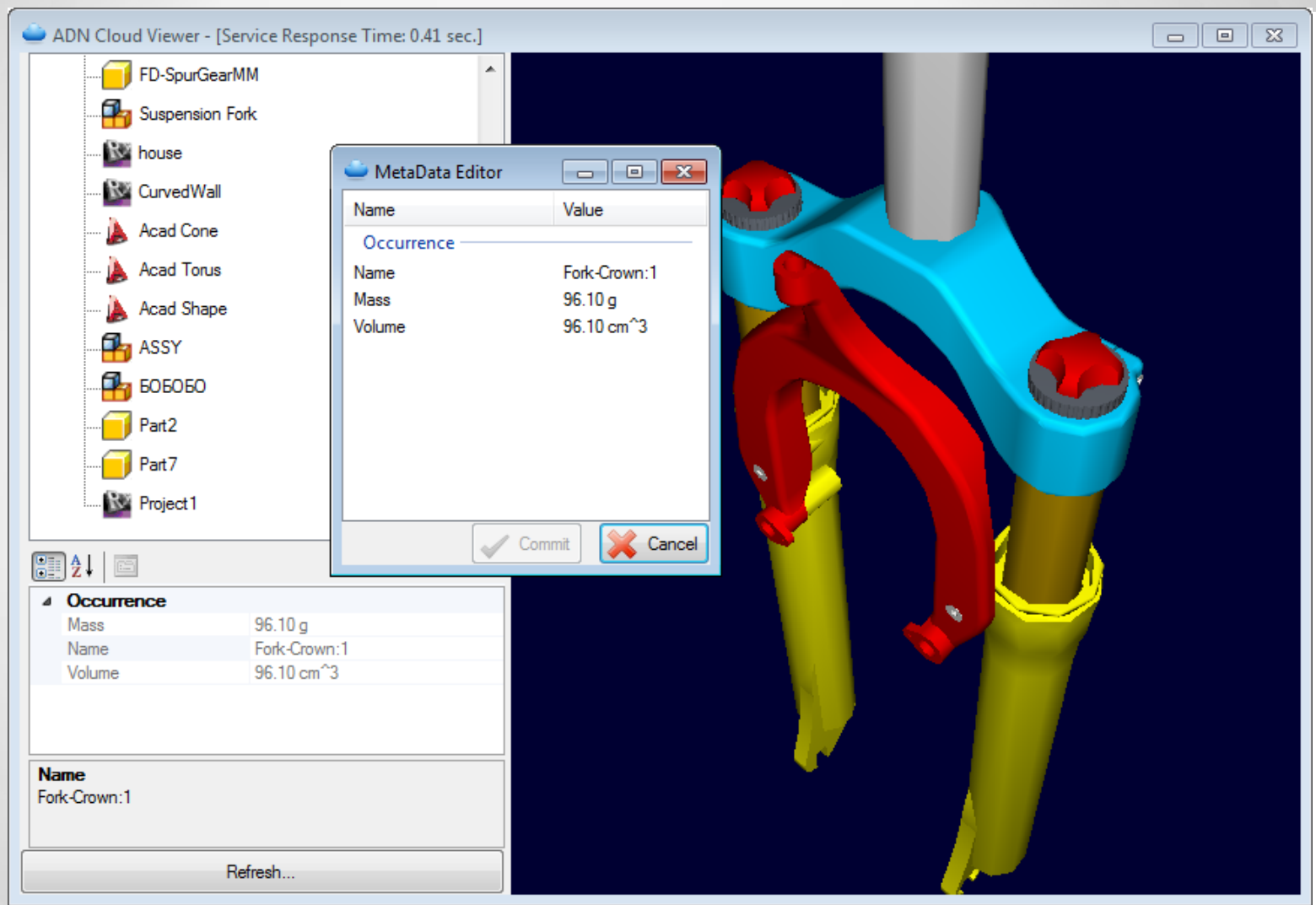


クラウドとモバイル – AutoCAD 360

サンプル

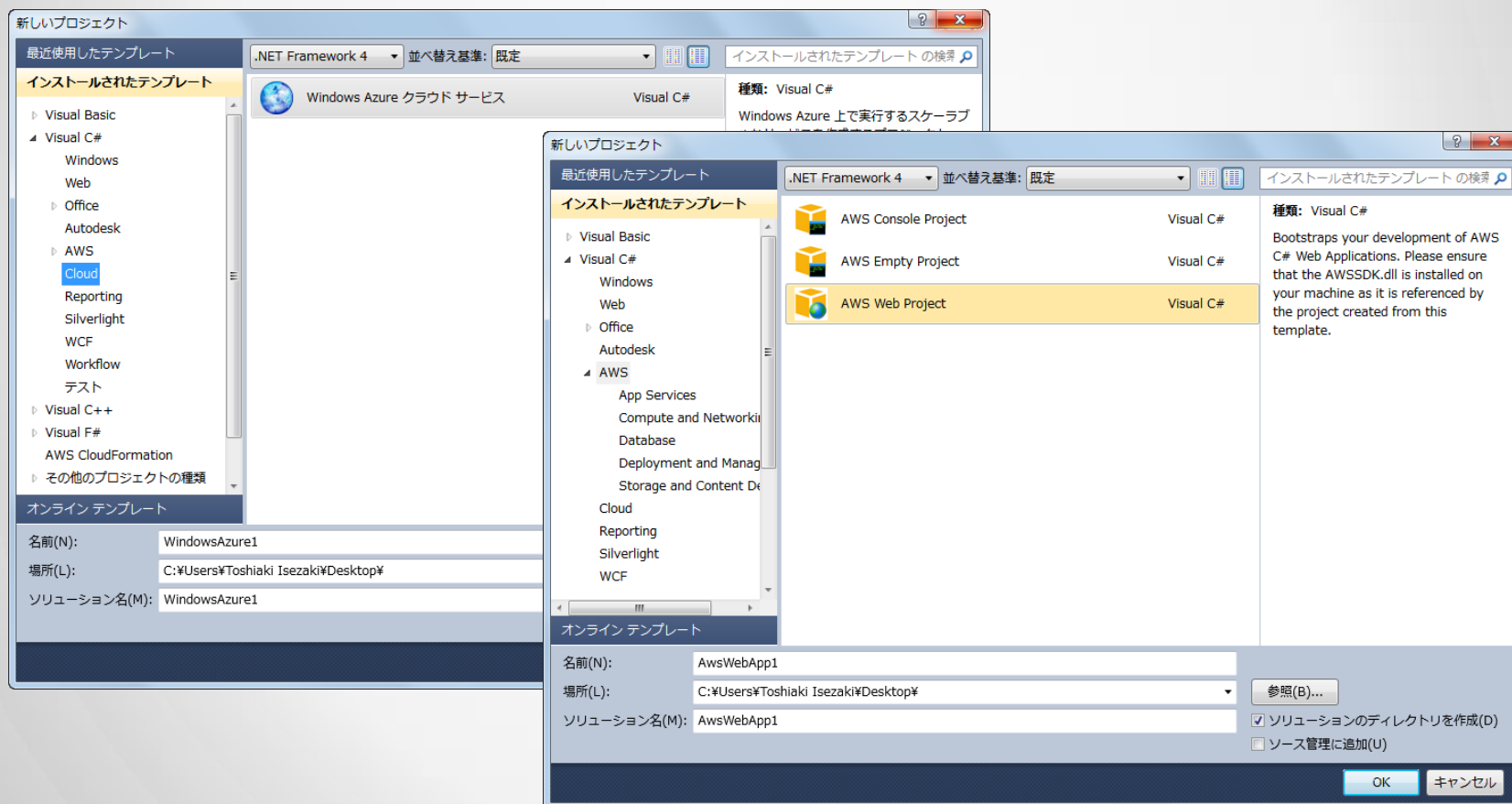


クラウドとモバイル – 独自開発

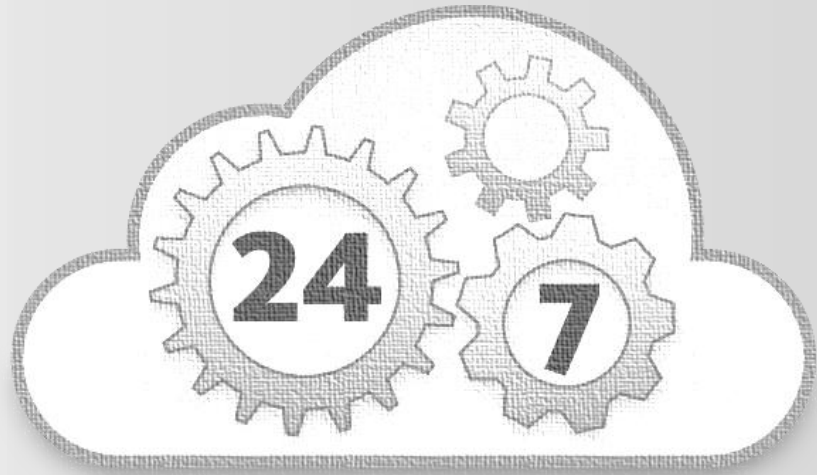


どのように開発すればよいのか？

- クラウド ベンダーによる支援ツールを利用可能

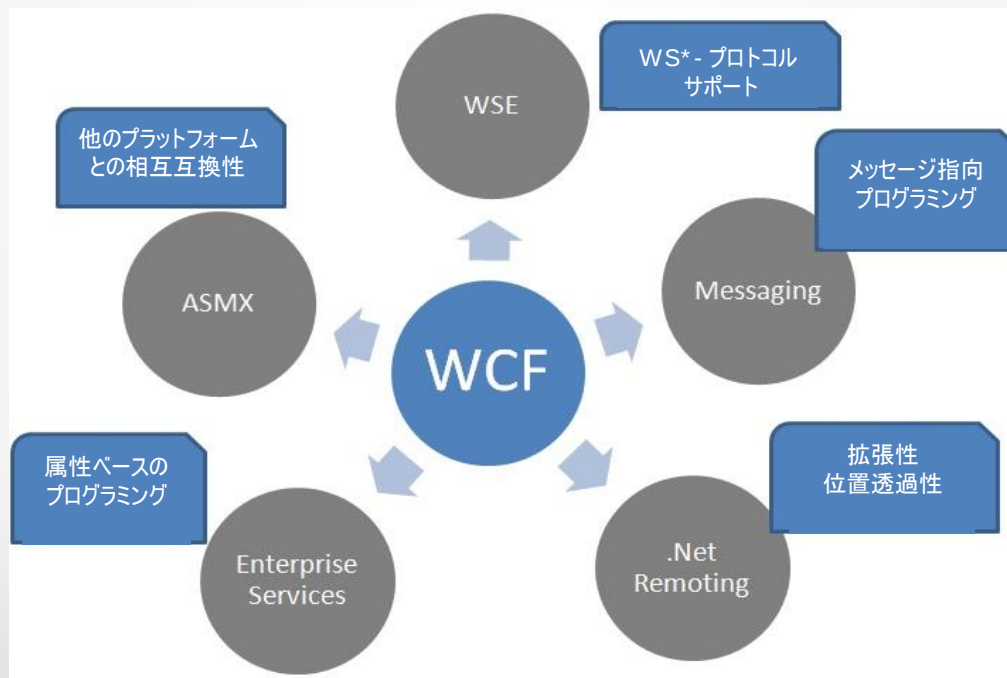


WCF Web サービスの実装



WCF とは ?

- **WCF (Windows Communication Foundation) とは**
 - Microsoft の次世代プログラミング プラットフォームでバインディング、設定、ビルド、ネットワーク配布されるサービスの展開のためのランタイム システムです(出典 MSDN)



WCF の基礎 - ABC

- “ABC” はどう WCF サービスが構成されるか理解する鍵:
 - ✓ “A” は **Address** を表現:
サービスはどこにあるのか？
 - ✓ “B” は **Binding** を表現:
クライアントはどうサービスと会話するのか？
 - ✓ “C” は **Contract** を表現:
サービスはクライアントに何ができるのか？

データコントラクト

```
[DataContract]
public class AdnMaterial
{
    [DataMember]
    public string Name
    {
        get;
        set;
    }

    [DataMember]
    public double Price
    {
        get;
        set;
    }

    [DataMember]
    public string Manufacturer
    {
        get;
        set;
    }
}
```

サービスコントラクト

```
[ServiceContract]
public interface IMaterialSrv
{
    [OperationContract]
    AdnMaterial[] GetMaterials();

    [OperationContract]
    AdnMaterial GetMaterial(
        string materialName);

    [OperationContract]
    bool PostMaterial(
        AdnMaterial material);
}
```

サービスの実装

```
public class MaterialSrv : IMaterialSrv
{
    public MaterialSrv()
    {
        //Constructor...
    }

    public AdnMaterial[] GetMaterials()
    {
        //Implementation...
    }

    //Implementation other methods...
}
```

サービスの動作

```
[ServiceBehavior(  
    InstanceContextMode=InstanceContextMode.PerSession,  
    TransactionTimeout="00:00:30")]  
public class MaterialSrv : IMaterialSrv  
{  
    [OperationBehavior(  
        TransactionAutoComplete=true,  
        TransactionScopeRequired=false)]  
    public AdnMaterial[] GetMaterials()  
    {  
  
    }  
}
```

サービスのバインディング

- バインディングはアドレスにサービス コントラクトの実装をバインド
- 次の項目を含む:
 - 適切な転送プロトコルを選択
 - どうメッセージがエンコードされるかを選択
 - どうセキュリティが動作するかを定義

サービスのホスティング

- WCF サービスをホスティングするために利用可能なオプション:
 - マネージ .NET アプリケーションのセルフ ホスティング
 - Winform、WPF、コンソール アプリケーション、 ...
 - Windows サービスでのホスティング
 - IIS でのホスティング
 - ASP.Net
 - ダイレクト ホスティング

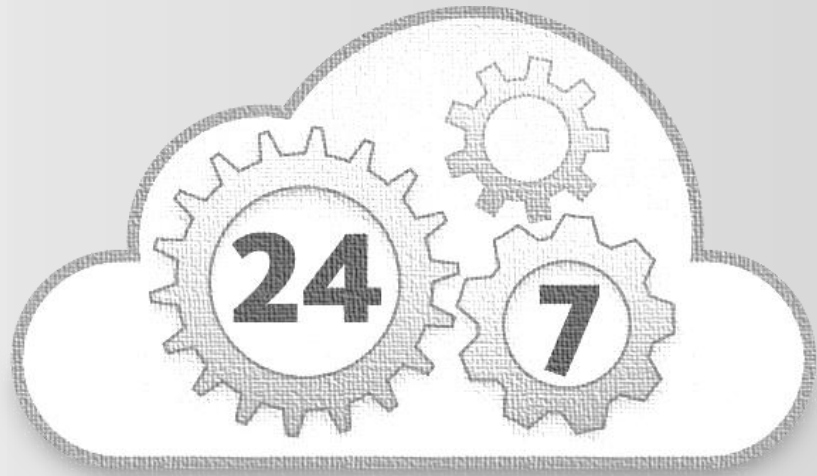
REST サポートの追加

```
[ServiceContract]
public interface IMaterialSrv
{
    [OperationContract]
    [WebInvoke(
        Method = "GET",
        UriTemplate = "/Material/{materialName}",
        ResponseFormat = WebMessageFormat.Json)]
    AdnMaterial GetMaterial(string materialName);

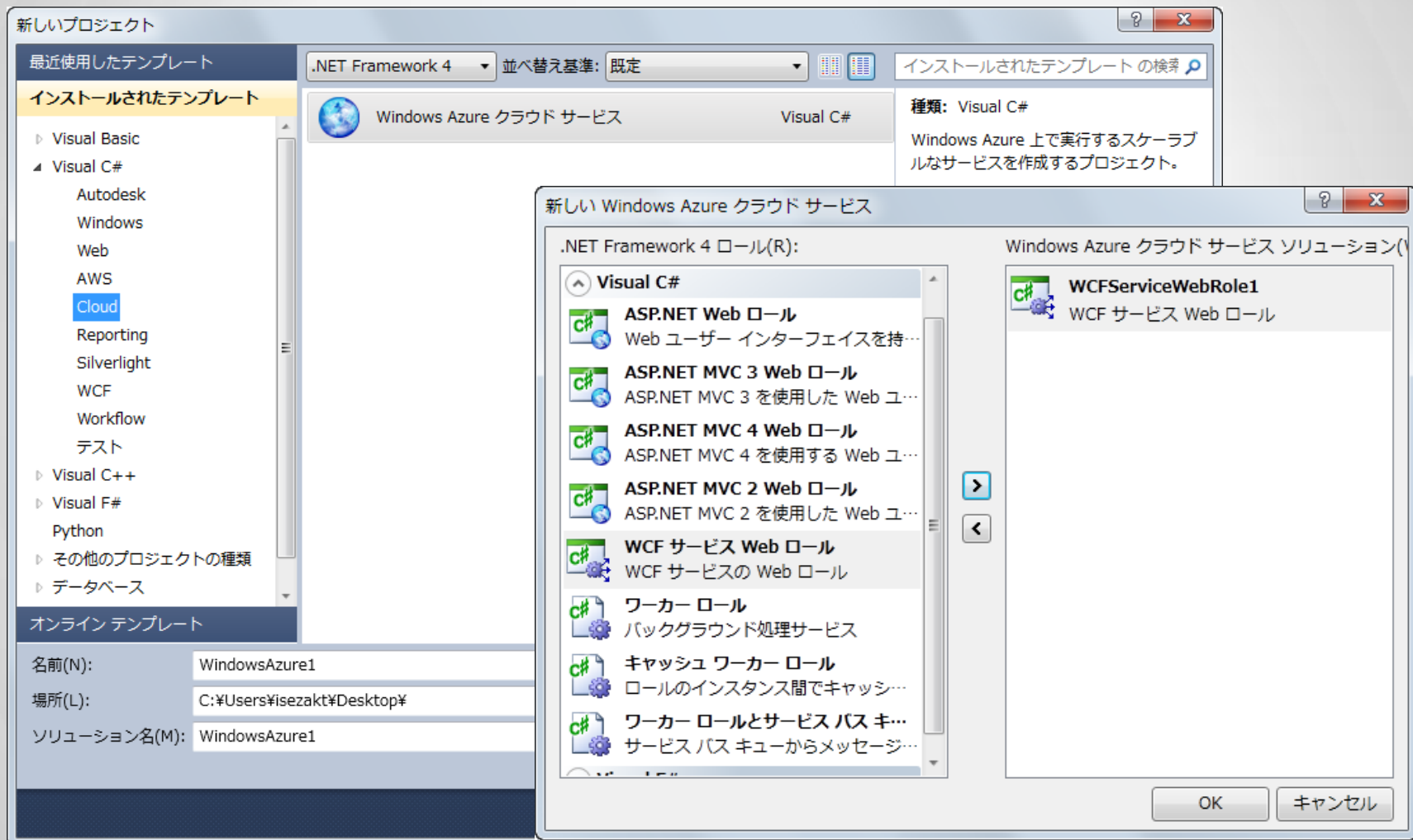
    [OperationContract]
    [WebInvoke(
        Method = "POST",
        UriTemplate = "/Material",
        RequestFormat = WebMessageFormat.Json,
        ResponseFormat = WebMessageFormat.Json)]
    bool PostMaterial(AdnMaterial material);
}
```



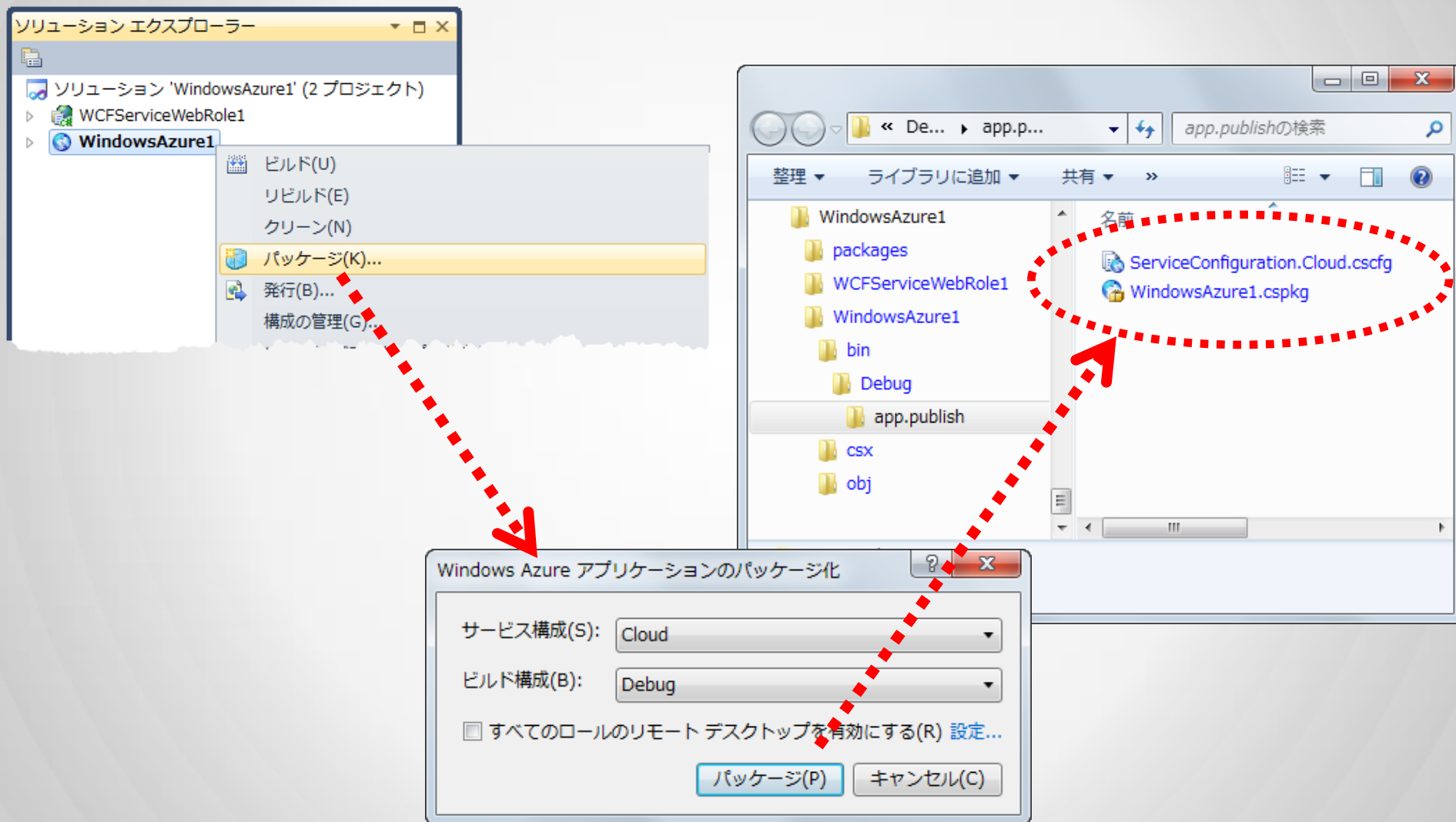
クラウドへのデプロイメント Windows Azure の例...



Step 1 - Azure プロジェクト



Step 2 – パッケージ作成



Step 3 - Azure のデプロイ

Windows Azure

すべてのアイテム

- クラウド サービス 0
- SQL データベース 0
- ストレージ 0
- サービス バス 0
- アドオン 0
- 設定

+ 新規

新しいクラウド サービス - カスタム作成

クラウド サービスを作成します

新しいクラウド サービス - カスタム作成

URL
adnmaterial

クラウド サービスの発行

クラウド サービス

名前	サービスの状態	運用	ステージング	サブスクリプション	場所	URL
adnmaterial →	✓ 作成日時	✓ 実行中	-	Windows Azure...	東アジア	http://adnmaterial...

AdnMaterialSrv.cspkg

ローカルから ストレージから

構成

ServiceConfiguration.Cloud.cscfg

ローカルから ストレージから

環境

運用 ステージング

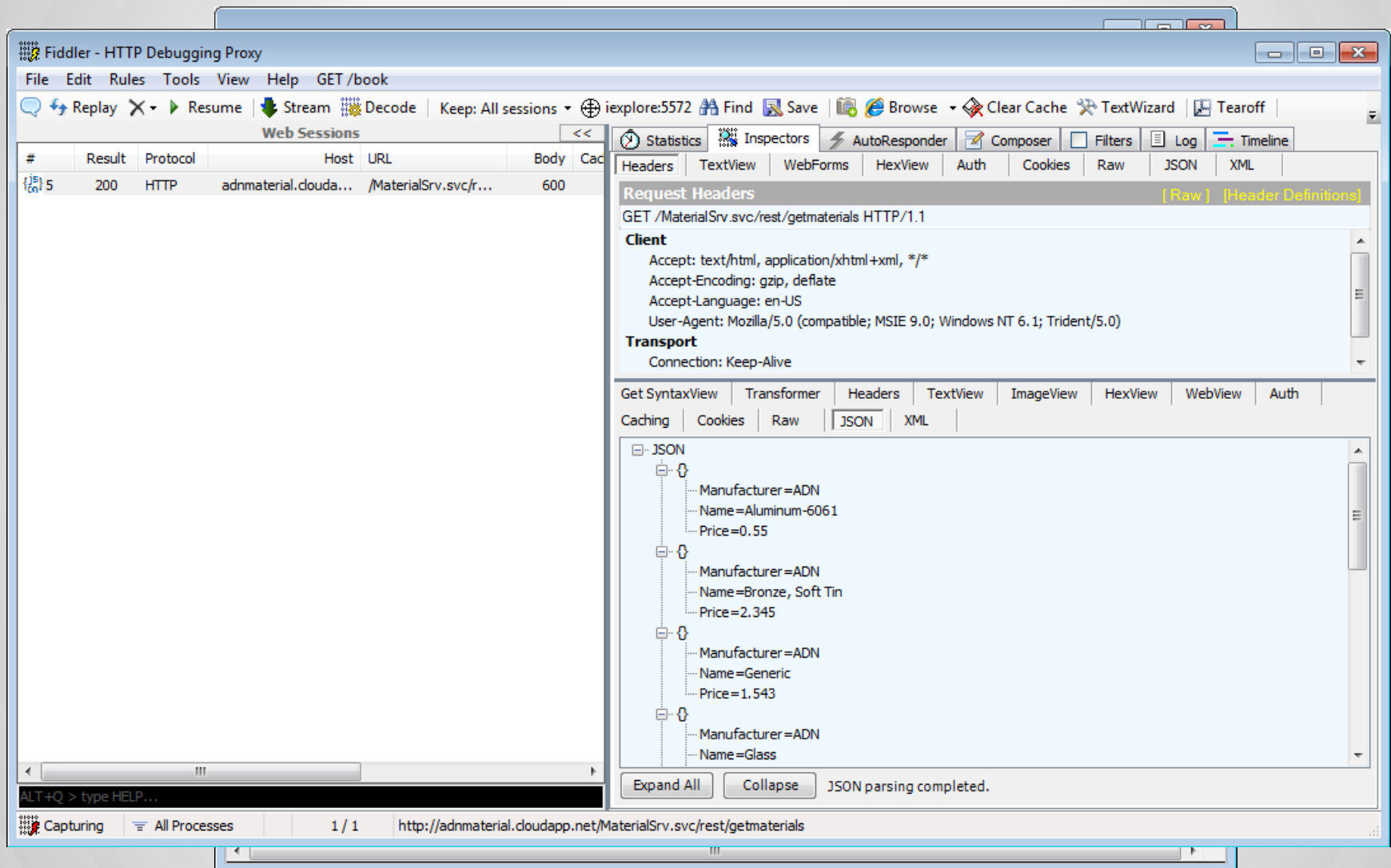
☐ 1 つ以上のロールに単一のインスタンスが含まれている場合でも展開します。?

☒ 展開の開始

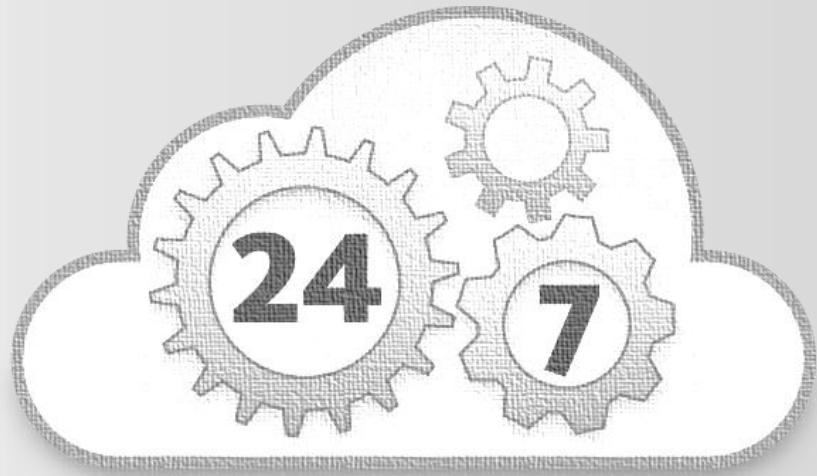
☐ 今すぐ証明書を追加

1

Step 4 – テスト



学習リソース



アドオン開発者のためのクラウド&モバイルの開発情報

- ブログ記事で公開:
 - http://adndevblog.typepad.com/cloud_and_mobile/

09/11/2012

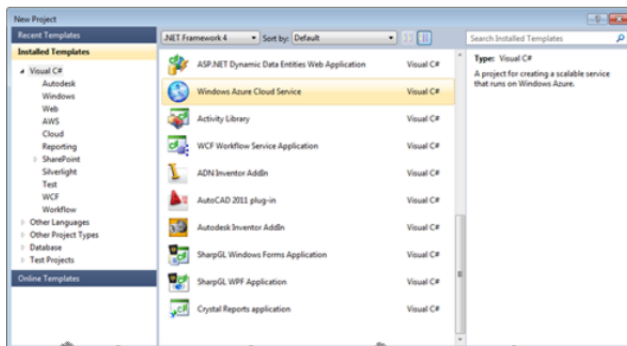
Creating a cloud based viewer – Part VIII

By [Philippe Leefsma](#)

Hi Guys, this week I will show you how I migrated my previous WCF service to be hosted on the Azure cloud instead of Amazon. I decided to follow a two step process: first I will migrate the service as it was previously, that is it will still rely on the AWS API to store and retrieve the data but it will be deployed on Azure, then in the next post I will migrate the code in order to replace Amazon SimpleDb and S3 by Azure Tables and Blob services.

I – Creating a new Azure project

If you haven't done it yet, you will first need to install the Azure SDK, which is rather straightforward, so I can just skip that step. We then need to create a new "Windows Azure Cloud Service" project:



10/22/2012

Real Cross-Platform Interactive 2D Graphics Using SVG

By [Jeremy Tammik](#)

Hi everybody.

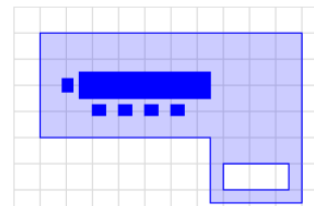
This is my first post to the Cloud and Mobile DevBlog.

All I will do here is point out that I spent my education day last Friday exploring how to display and interact with 2D graphics in a browser using SVG and JavaScript by implementing a very simple [room polygon and furniture picker in SVG](#).

The main goal was to enable picking and identifying objects and ensuring that the same code and interaction is supported on all mobile devices, in particular both Android and Safari on iOS.

The latter added an additional little challenge at the end.

Here is the final result:



Welcome. Please click around.

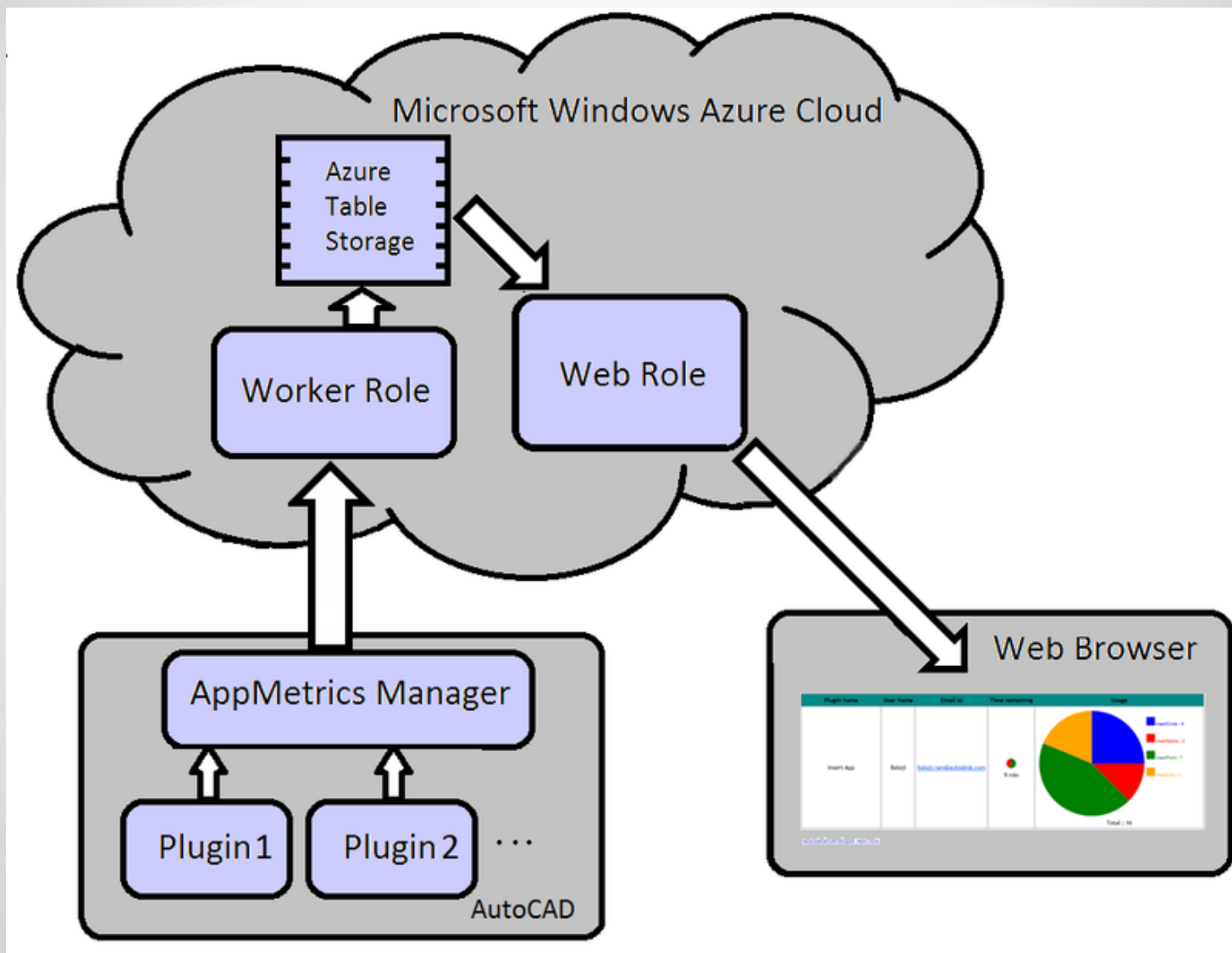
AutoCAD JavaScript API

- ブログ記事で公開:
 - http://adndevblog.typepad.com/technology_perspective/2013/05/autocad-javascript-api-part1.html
 - http://adndevblog.typepad.com/technology_perspective/2013/05/autocad-javascript-api-part2.html
 - http://adndevblog.typepad.com/technology_perspective/2013/06/autocad-javascript-api-part3.html
 - http://adndevblog.typepad.com/technology_perspective/2013/07/autocad-2014-javascript-api_development_preps.html

公開したアドオン アプリのライセンス管理のアイデア

- ブログ記事でアイデアを公開：
 - <http://adndevblog.typepad.com/autocad/2012/08/licensing-applications-infralution.html>
 - <http://adndevblog.typepad.com/autocad/2012/09/licensing-applications-crypto.html>
 - <http://adndevblog.typepad.com/autocad/2012/10/trial-licensing-and-usage-metrics-gathering-for-autocad-plugin-using-azure-cloud.html>

クラウドを利用した期限付きライセンスの実現





Autodesk is a registered trademark 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.