

自己紹介

- J Z 5 (松江祐輔) @わんくま同盟
- jz5 @Twitter
- 仕事: SystemVerilogを使いたい
- Windows Liveの記事も書いてます @
gihyo.jp
- katamari.wankuma.com
- katamari.jp

はじめに

- CTPを元にした話しです。
- 私が入門しました。

目次

- Live Framework 概要
- Live Mesh Services
- Live Framework Programming

Azure Service Platform & Live Services

- Live Services: Azure Service Platformのビルディングブロックのひとつ

Azure Service Platform

Live
Services

.NET
Services

SQL
Services

...

Windows Azure

Live Services

- ユーザーデータとアプリケーションリソースを扱う

Live Services

Mesh Services

ユーザー

デバイス

アプリ

同期

認証

連絡先

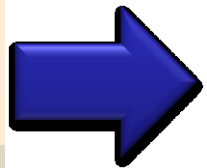
記憶域

通信 &
接続状況

検索 &
地図

Live Framework

- **Live Services**を利用するための**統一的手法**
 - オープン・シンプルな仕様
 - HTTP and (XML or JSON)
 - リソース指向
 - Client・Web共通のプログラミングモデル



Live Servicesを利用したプログラミングができる！

Live Mesh

- Live Mesh Beta <http://mesh.com>
- ファイルの同期・共有
 - エクスプローラと統合
 - Webブラウザアクセス
 - 招待したユーザー同士
- オンラインストレージ
 - 5GB
- リモートデスクトップ



Live Mesh Keywords

- **Devices:** Meshに接続されているデバイス
 - PC, Mac, 携帯電話
 - ゲーム機, カメラ, プリンターなど
 - Amazon, Flickr, YouTubeなど
 - Cloud Devices(Storage & Computing)
- **Live Desktop**
 - Deviceのひとつ, Web上のPC, Webブラウザからアクセス
- **Mesh bar:** ウィンドウに付く情報ウィンドウ

Live Framework CTP

- Live Mesh Betaとは異なる開発者向けLive Mesh Servicesを用意
 - Live Framework Client
 - Developer Sandbox
- Mesh-enabled Web アプリケーション

Mesh-enabled Web App

- **Web App + Desktop App**
 - どこからでもアクセス(Web Sites & Devices & Live Desktop)
 - オフライン実行
 - データの同期・共有
 - 自動アップグレード
 - サンドボックス内で実行
- **Live Meshサービスでホスト**

Live Framework による開発

- Live Framework SDK and Tools

- .NET Kit

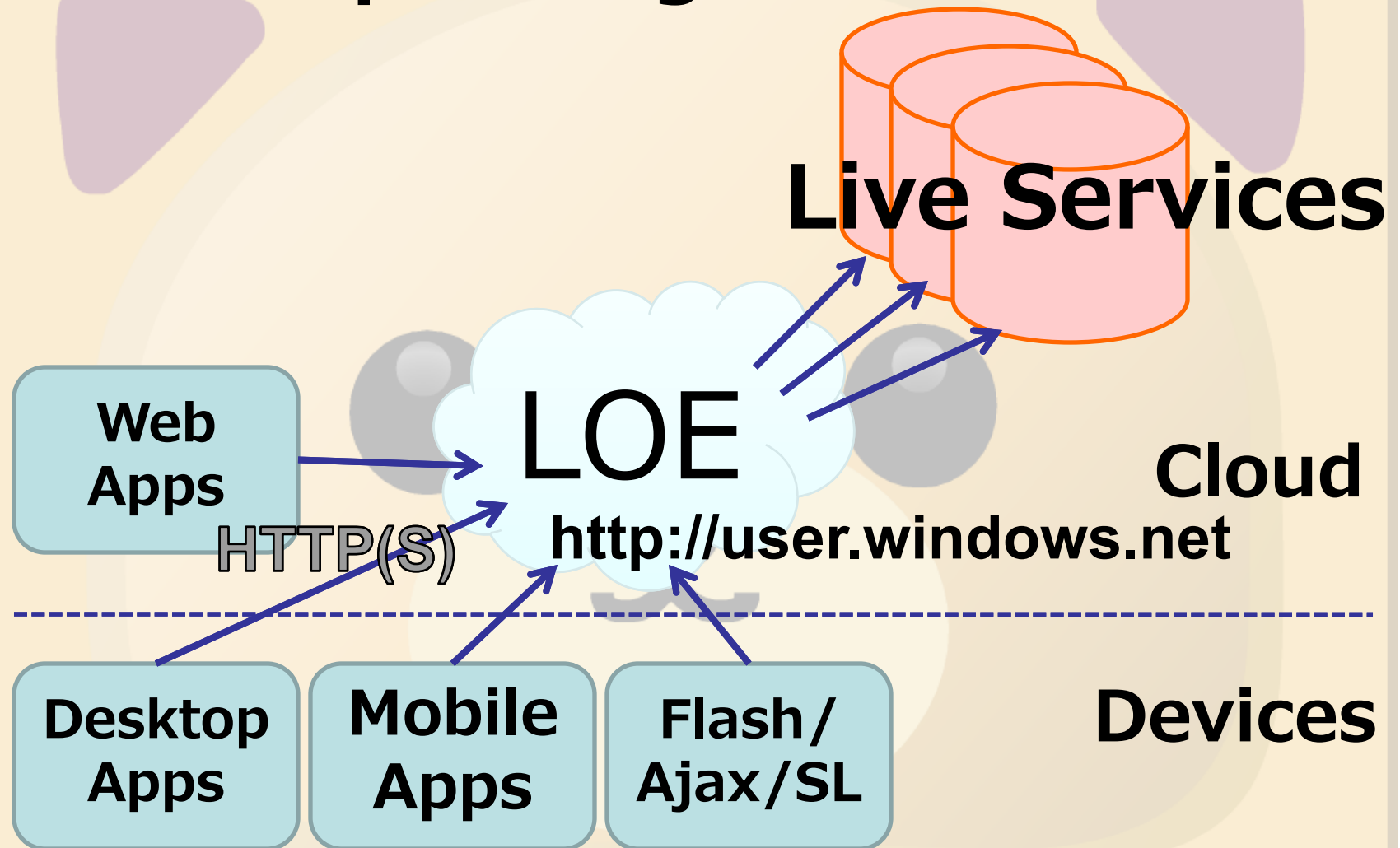
- Silverlight Kit

- JavaScript Kit

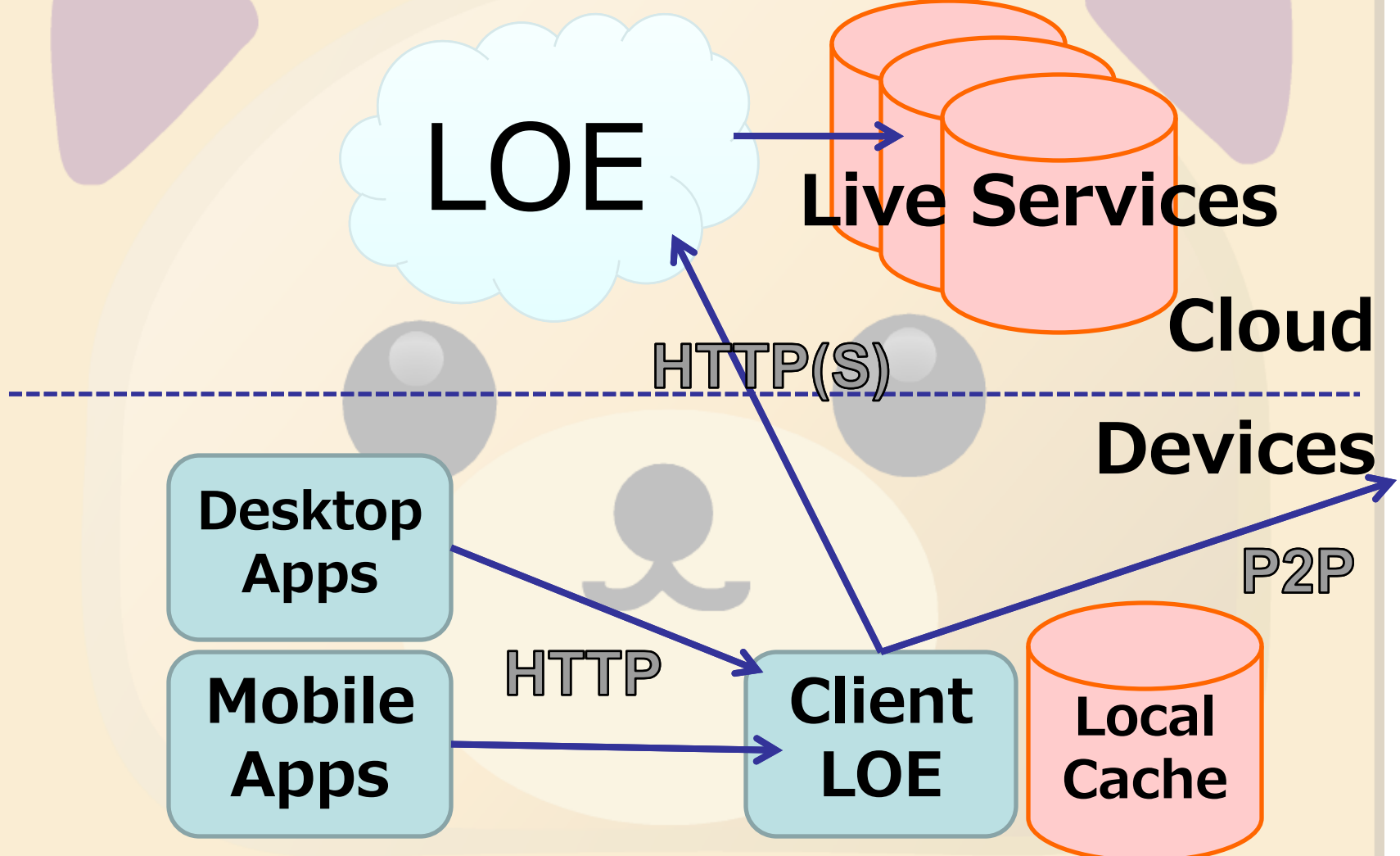
} 低レベルなHTTPやXML/JSONを意識せずにプログラミング可能

- 各種ユーザー登録や環境準備が必要

Live Operating Environment



Live Operating Environment

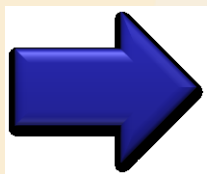


<http://localhost:2048>

LOEへの接続

- Live Framework SDK DLLsを参照

```
var creds = new NetworkCredential("***@hotmail.com", "password");  
var loe = new LiveOperatingEnvironment();  
loe.Connect(creds, new LiveItemAccessOptions(true));
```



<https://user-ctp.windows.net/> or
<http://localhost:2048/>
へアクセス



リソースへのアクセスの前に

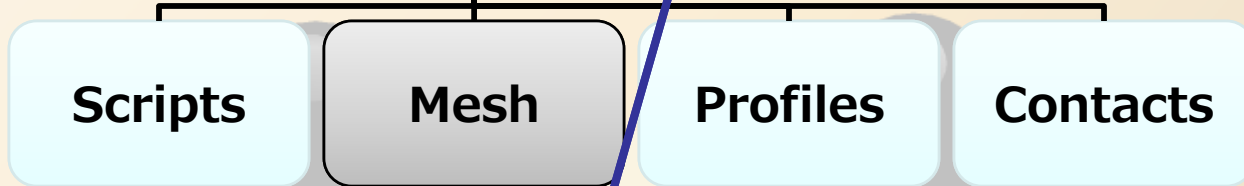
- **リソース指向**（Live Fxの特長のひとつ）
 - リソース: Applications, Folders/Files, Devices, Etc.
 - URIで表現可能
 - HTTPメソッドでアクセス可能

リソースモデル

- Live Frameworkでは何がリソース？

Service Endpoint

<https://user-ctp.windows.net/V0.1/Mesh/MeshObjects>



Mesh Objects

- **Mesh Services**で同期される単位
 - 例: Live Mesh Folder
 - Data Feedのコレクションを持つ
- **Data Entry**
 - 写真や文書などひとつのデータを表す単位
- **Data Feed**
 - Data Entryのコレクションを持つ

リソースへアクセス

- **Devicesの一覧**

```
foreach (var d in loe.Mesh.Devices.Entries)
{
    Console.WriteLine(d.Resource.Title);
}
```

- **Mesh Objectの一覧**

```
foreach (var m in loe.Mesh.MeshObjects.Entries)
{
    Console.WriteLine(m.Resource.Title);
}
```

Mesh Objectの作成

• Live Mesh Folderの作成

```
var folder = new MeshObject();
folder.Resource.Title = "新しいフォルダ";
folder.Resource.Type = "LiveMeshFolder";
loe.Mesh.MeshObjects.Add(ref folder);

var fileSystem = new DataFeed();
fileSystem.Resource.Title = "LiveMeshFiles";
fileSystem.Resource.Type = "LiveMeshFiles";
fileSystem.Resource.HandlerType = "FileSystem";
folder.DataFeeds.Add(ref fileSystem);

var fs = new System.IO.FileStream(@"C:¥misao.png", System.IO.FileMode.Open);
fileSystem.DataEntries.Add(fs, "misao.png", "image/png");
fs.Close();
```



応用？

- **独自データの保存**
 - MeshObject, DataEntryにある
SetUserData, GetUserDataメソッド
- **更新通知**
 - ChangeNotificationReceivedイベント
- **同期と衝突**
 - DataFeed.SyncEntries.Entries プロパティ

Mesh-enabled Web Appの開発

- Mesh-enabled Web AppはMesh Objectのひとつ
- Mesh barが利用可能
- SDK/ToolsでクライアントAppとほぼ同等な記述で開発可能

Note: ユーザーによる許可がない限り、自分のData Feeds, Data Entries以外にはアクセス不可

まとめ

- **Live Framework 概要**
 - Live Servicesへアクセスする統一的手法
- **Live Mesh Services**
 - Betaと開発者向けCTP
- **Live Framework Programming**
 - .NETなら楽ちゃん

Enjoy Live Programming

