

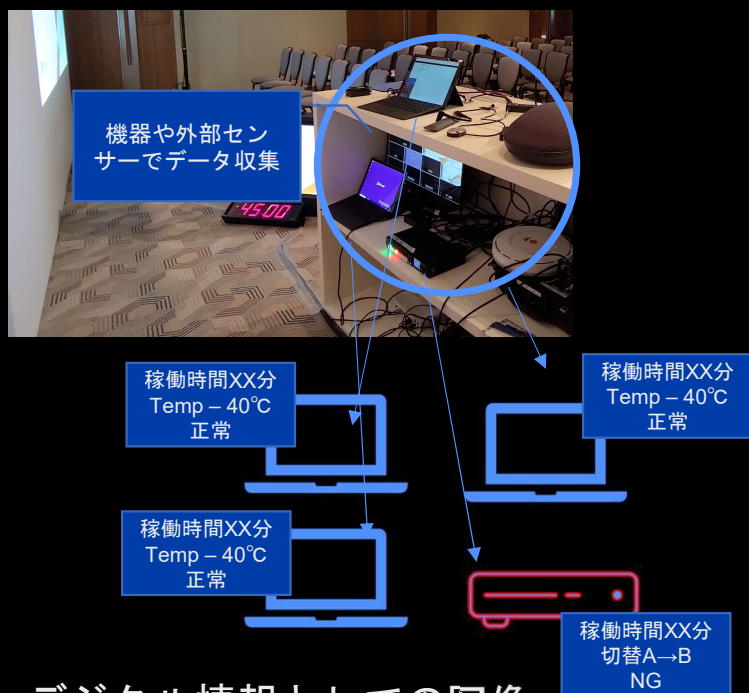
Mixed Realityを活用したIoT Edgeの管理と 情報可視化による「Digital Twins」の実現

「Digital Twins」と関連する技術

Digital Twinsとは

実データを活用してより精度の高いシミュレーションを実現

現実空間内の分析対象



デジタル情報としての写像

例：配信時の機器情報を可視化し改善する

【従来】

- ✓ 対象をモデル化
- ✓ 仮想化したモデルパラメータを変更
- ✓ 状態を分析

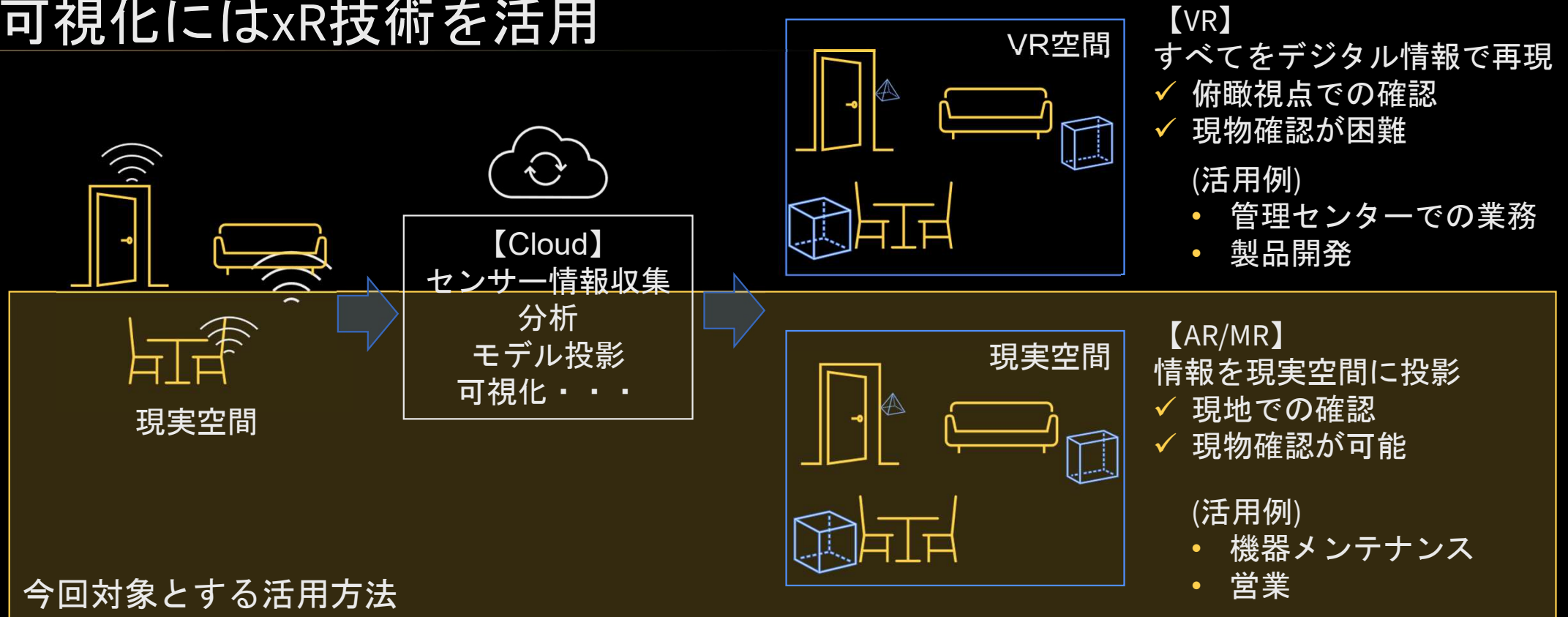


【デジタルツイン】

- ✓ 対象をモデル化
- ✓ 対象にセンサーなどを設置や取り付け
- ✓ 実データを取得しモデルに情報を反映
- ✓ リアルタイムに状態を分析

XR技術とDigital Twins

可視化にはXR技術を活用



本日のテーマ

情報の可視化を現実空間で実施

機器メンテナンスを想定。対象機器特定/センサー情報を補助空間に出力



検証に利用した技術の紹介

デモと活用方法に関する考察

Mixed Realityを活用したIoT Edgeの管理と 情報可視化

概要

簡単なモデルケースを実際に構成

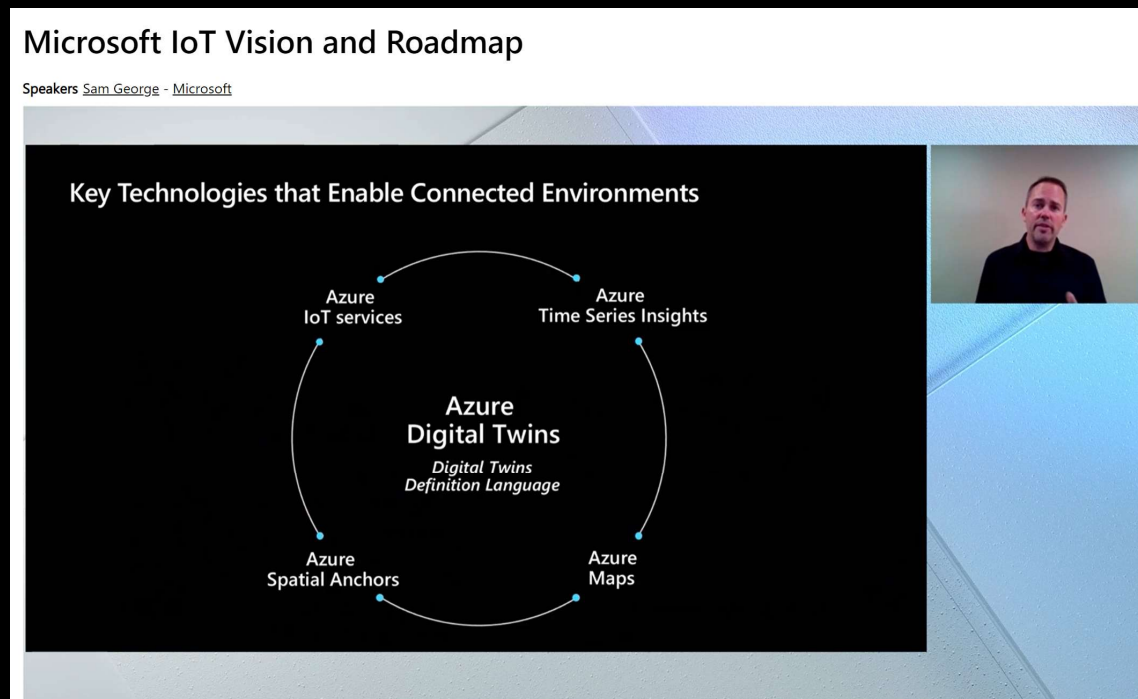
- ✓ 現実空間内にセンサーを設置
- ✓ センサー情報をクラウドを活用してデータを収集
- ✓ 収集データから作業指示などの情報を“センサー近傍で出力”



Azureで実現できるDigital Twins

～ 提供サービスと役割 ～

Microsoft Build 2020で紹介されていたIoT戦略とロードマップ

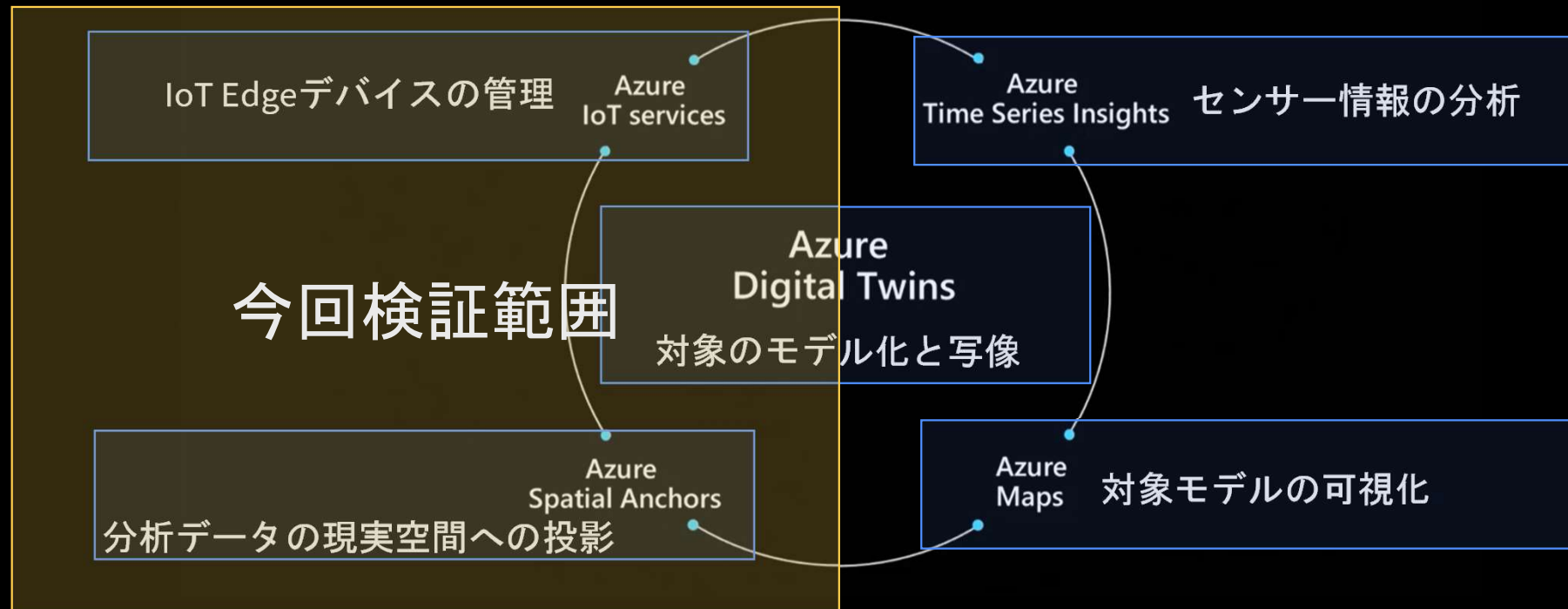


引用元 : 「Microsoft Buile 2020:Microsoft IoT Vision and Roadmap」 <https://mybuild.microsoft.com/sessions/007295e3-c677-488c-b5df-6dad2a0f90cc>, May 21 2020

Azureで実現できるDigital Twins

～ 提供サービスと役割 ～

IoT Edgeデバイスの管理から可視化までをサポートするサービスの提供

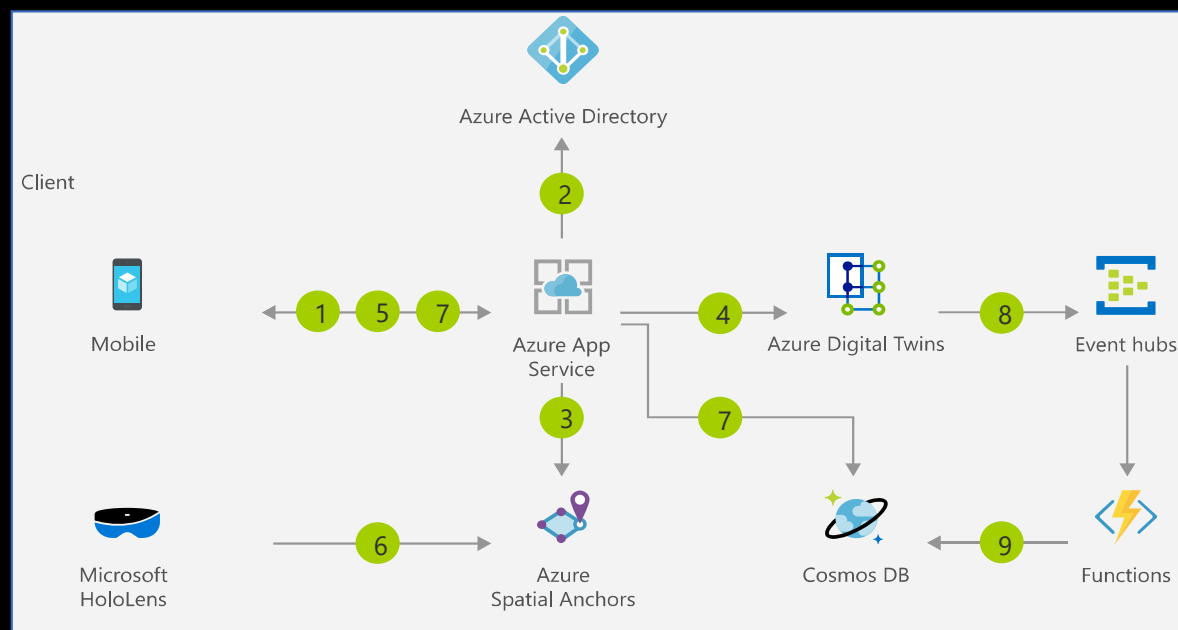


Azureで実現できるDigital Twins

～ 参考アーキテクチャ～

Azure Architecture Center: 「複合現実とIoTによる設備管理の促進」

https://docs.microsoft.com/ja-jp/azure/architecture/solution-ideas/articles/facilities-management-powered-by-mixed-reality-and-iot?WT.mc_id=MR-MVP-5003104

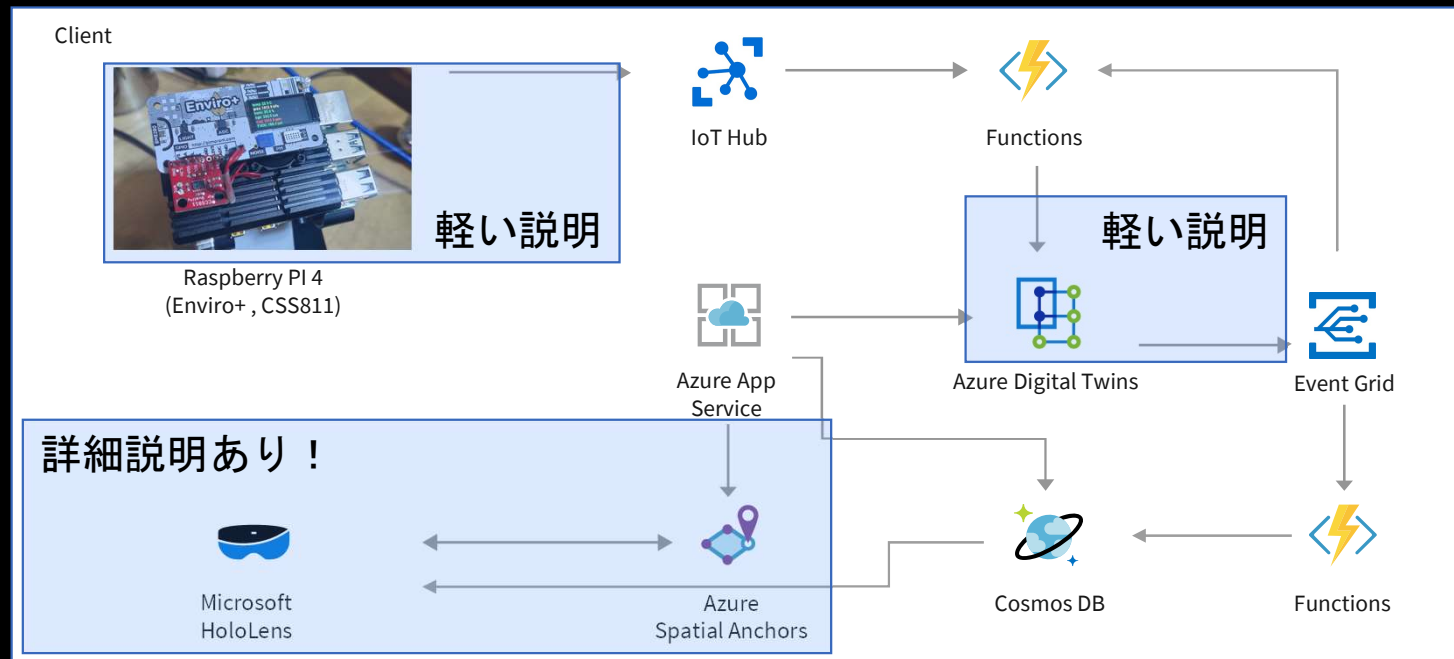


このアーキテクチャをベースにサービスを構成

システム構成

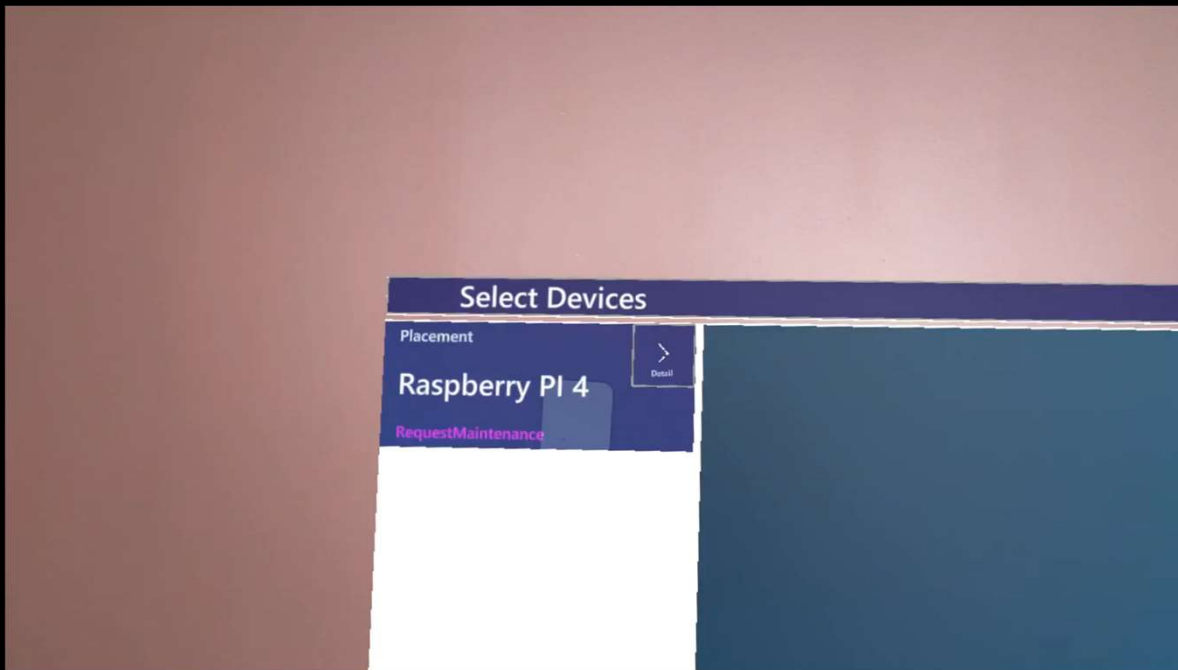
空気環境情報をセンシングするデバイスを空間に設置

- ✓ Azure上で情報を管理 (Azure IoT Hub, Azure Digital Twins, CosmosDB)
- ✓ 現実空間のセンサー位置をAzure Spatial Anchorsで管理
- ✓ 必要な情報 (対象センサー位置、センサー生データ等) の現実空間での可視化

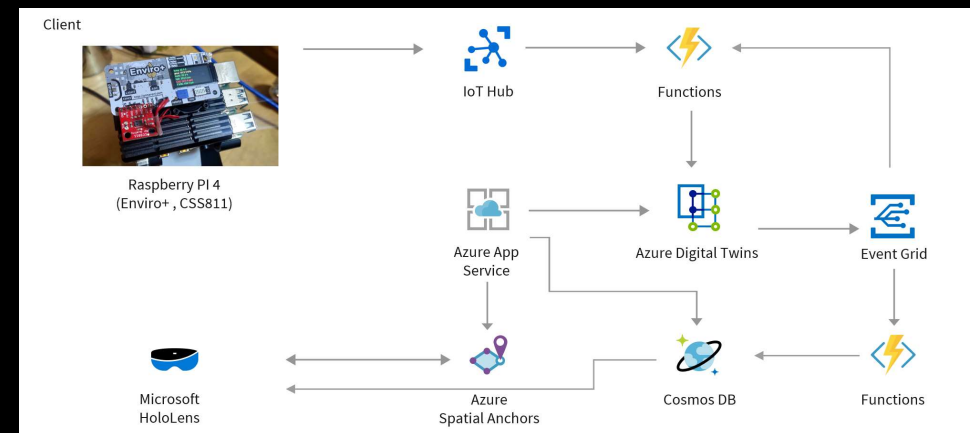


DEMO

確認対象のセンサーの位置特定とメンテナンス

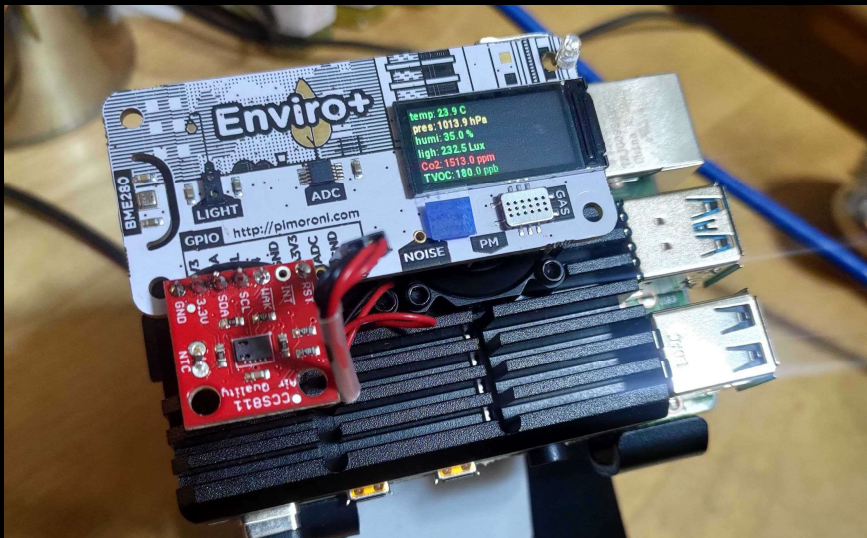


- ✓ Azure Spatial Anchorsによるセンサー位置の特定
- ✓ センサー/作業情報を取得し可視化
- ✓ メンテナンスの報告



IoT Edgeデバイス

IoT EdgeデバイスはRaspberry PI で簡易的に作成



Raspberry PI 4

- ✓ センサー情報
温度、湿度、気圧、照度、Co2濃度等
- ✓ Azure IoT HubとはNode.js用のライブラリを経
IoT Hubのサンプルをほぼ流用
- ✓ iBeaconとしてBluetooth機能を利用
Azure Spatial Anchorsの利用で必要

Azure Digital Twins(Public Preview)

～ 分析対象をモデル化し写像を管理 ～

The screenshot displays the Azure Digital Twins Explorer interface. On the left, the 'MODEL VIEW' shows a list of entities: Building (dtmi:demo:Buildi...), Floor (dtmi:demo:Floor;1), Room (dtmi:demo:Room;1), and Sensor (dtmi:demo:Senso...). The central 'GRAPH VIEW' shows a hierarchical tree structure: Building0 is the root, containing Floor1 and Floor0. Floor1 contains Room2 and Room3. Floor0 contains Room0 and Room1. Room0 is connected to a sensor node labeled 'raspberrypi...'. The right 'PROPERTY EXPLORER' shows the details for the selected sensor node:

- UUID : ff7e5227-af72-426c-aca1-1baade641cdb
- AnchorId : 0b08c611-c94c-4214-cecc-c691c1d41655
- SensorName : Raspberry PI 4
- Temperature : 27.66
- Pressure : 1002.78
- Humidity : 36.97
- Light : 69.71
- Co2 : 2423
- TVOC : 1618
- IsAlert : ☒ true
- StatusCode : [\"000000\"]

情報はJson形式で記述

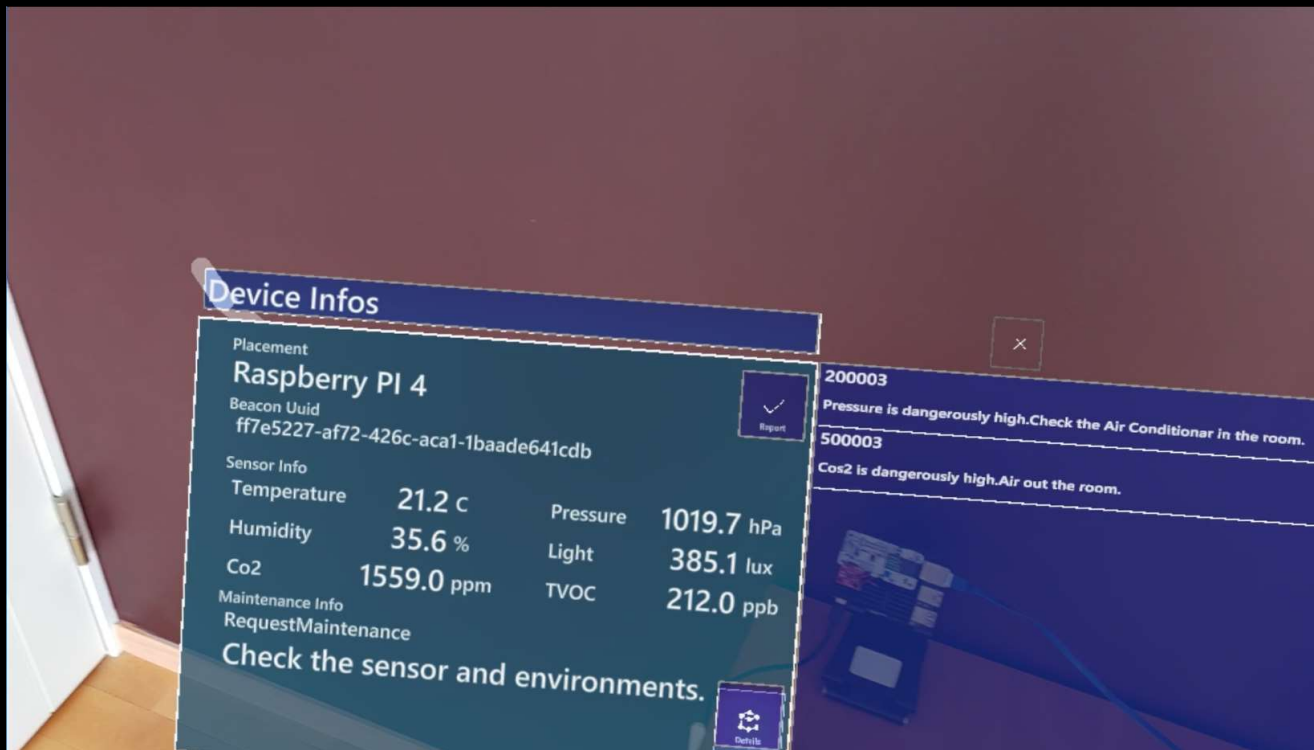
- ✓ 現実世界の情報をモデル化
- ✓ モデル化したデータの写像を生成
- ✓ ノードモデルとして管理
- ✓ 写像データの管理
 - ✓ CRUD
 - ✓ トリガー

今回は以下の機能を付与

- ✓ IoT Hubからの情報を反映
- ✓ 変更をトリガーにCosmosDBを更新

HoloLens 2での可視化

Microsoftの各種MR系サンプルは
MRTK V2.4.0+Unity 2019.4
(MRTKのgithubに記載)



■ HoloLens 2での利用機能

- ✓ ハンドトラッキング
- ✓ 空間マッピング

■ Mixed Reality Toolkit V2.5.1を使用 (Package Manager経由)

■ 主な利用機能

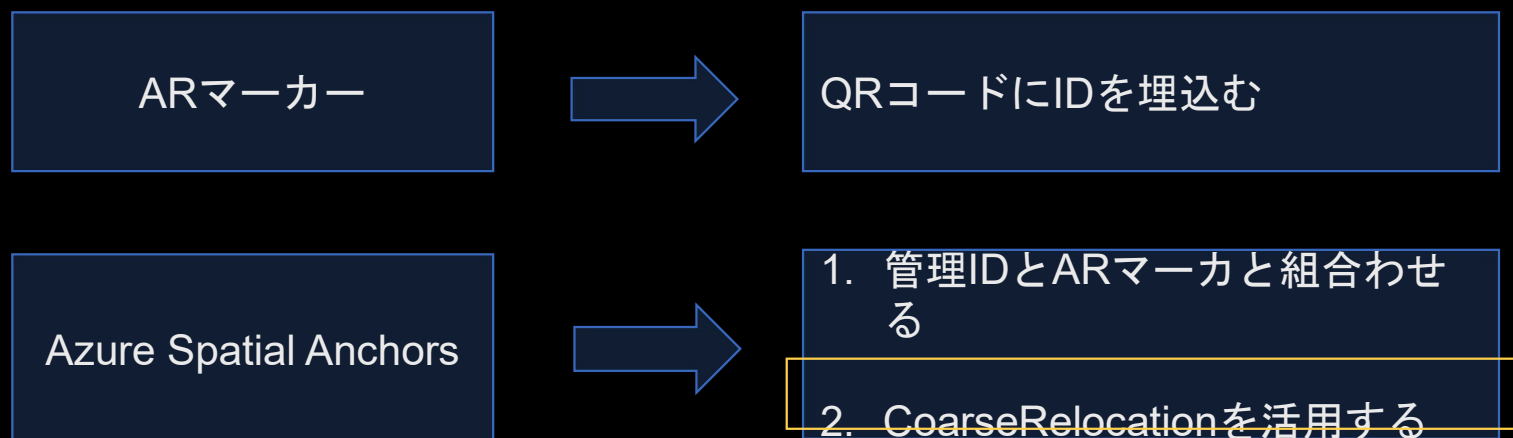
- ✓ Button
- ✓ ProgressIndicator
- ✓ Elastic System
- ✓ Hand Menu

(参考) 少しのUI変更 (今回だとHand Menu) でAndroid,iOSにも対応可能
※MRTK、Azure Spatial Anchorsがクロスプラットフォーム対応のため

可視化にあたってのポイント（要件）

センサーの現実空間での位置の把握が必要

✓ どのセンサーを対象とするか特定が難しくなるため



今回はAzure Spatial AnchorsのCoarseRelocationを活用
【理由】 QRコード等のマーカなしに、現実空間にアンカー設置が可能

Azure Spatial Anchors

～ Coarse Relocation ～

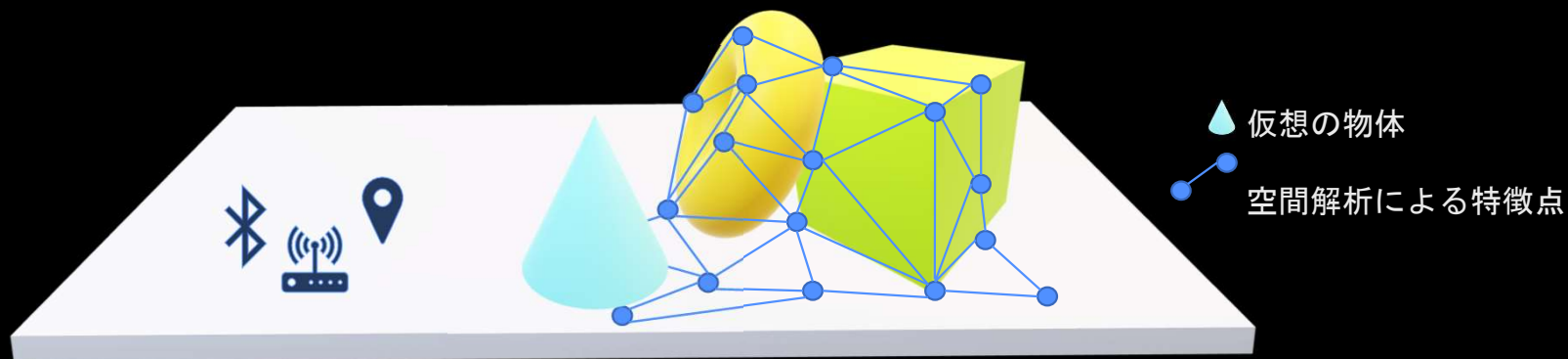
センサー情報を利用してSpatial Anchorの問合せが可能

- ✓ 「空間の特徴」
- ✓ 「位置情報 (=アンカー)」
- ✓ 「デジタル情報」
- ✓ 「センサー情報」
(Wifi/GPS/Beacon)



クラウドで管理 (IDで管理)

- ✓ クロスプラットフォームで活用
- ✓ 共有体験の土台

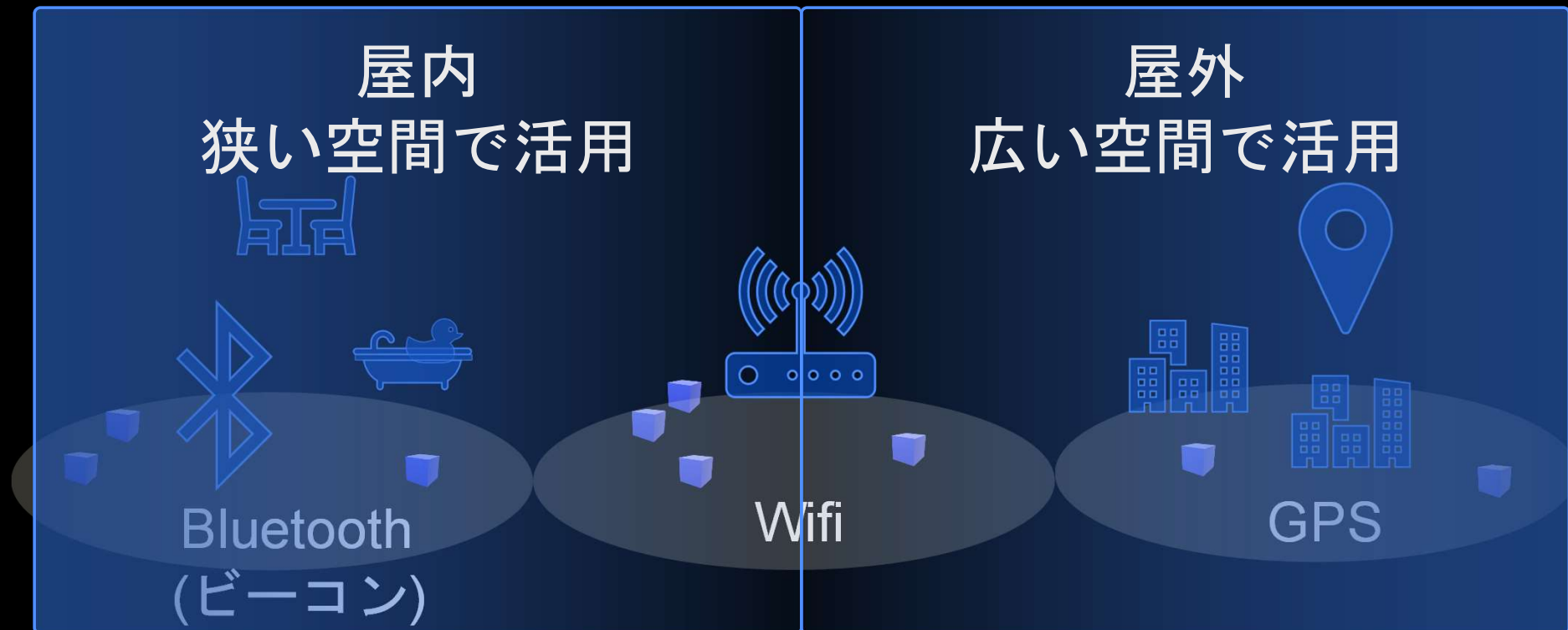


Copyright © 2020 Takahiro Miyaura

Azure Spatial Anchors

～ Coarse Relocation – センサーの使い分け ～

有効範囲によって使い分ける



(参考)Azure Spatial Anchors

～ 各デバイスのサポート状況 ～

有効範囲によって使い分ける

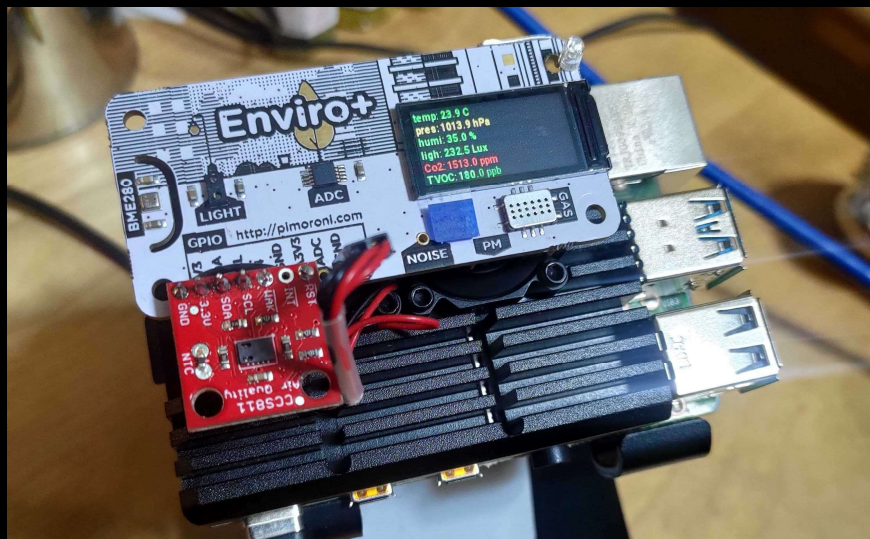
https://docs.microsoft.com/ja-jp/azure/spatial-anchors/concepts/coarse-reloc?WT.mc_id=MR-MVP-5003104

センサー種別	HoloLens	Android	iOS
GPS	× 機能なし	○ APIサポート	○ APIサポート
Wifi	○ 3 秒ごとに約 1 回のスキャン	○ API レベル 28 以降スキャンは 2 分毎に 4 回の呼出に調整。Android 10 からは調整可能	× なし
BLE Beacon	○ Eddystone iBeacon		

Azure Spatial Anchors

～ IoT EdgeデバイスのBeacon化が必要な理由 ～

Beaconとして機能させることで、Coarse Relocationが利用可能



Raspberry PI 4

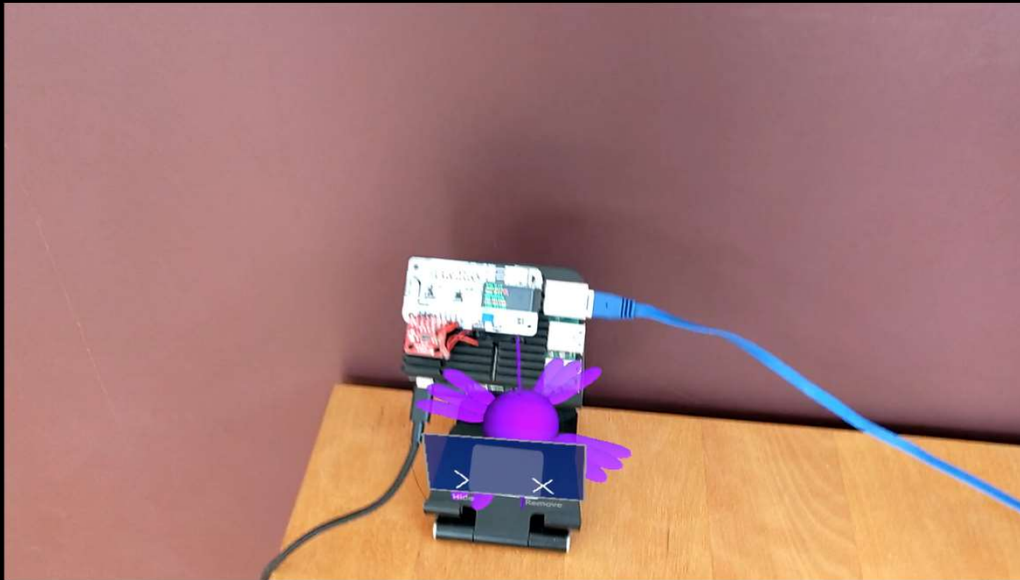
- ✓ センサー情報
温度、湿度、気圧、照度、Co2濃度等
- ✓ Azure IoT HubとはNode.js用のライブラリを経
IoT Hubのサンプルをほぼ流用
- ✓ iBeaconとしてBluetooth機能を利用
Azure Spatial Anchorsの利用で必要

センサーに紐づく Spatial Anchorを生成可能
センサーの位置関係を管理可能に！

Beaconを活用した Azure Spatial Anchorsの実装

Azure Spatial Anchors

～ 実装手順 ～



以降はBluetoothを利用したアンカー制御の実装を解説

- ・ アンカーの登録
- ・ アンカーの検索
- ・ アンカーの削除

Azure Spatial Anchorsはアンカー制御用にSDKが提供

- ・ クロスプラットフォーム
- ・ Unity向け

今回のデモシステムはGithubで公開（詳細を確認したい方は参考にしてください。）

<https://github.com/TakahiroMiyaura/DigitalTwinsDemoSystem>

(参考)開発環境

HoloLens 2 & Azure Functions

- ✓ Windows 10 Pro
- ✓ Unity 2019.4.10f1
- ✓ Mixed Reality Toolkit V2.5.1
- ✓ Azure Spatial Anchors V2.6.0
- ✓ Visual Studio 2019(16.5.3)

IoT Edgeデバイス

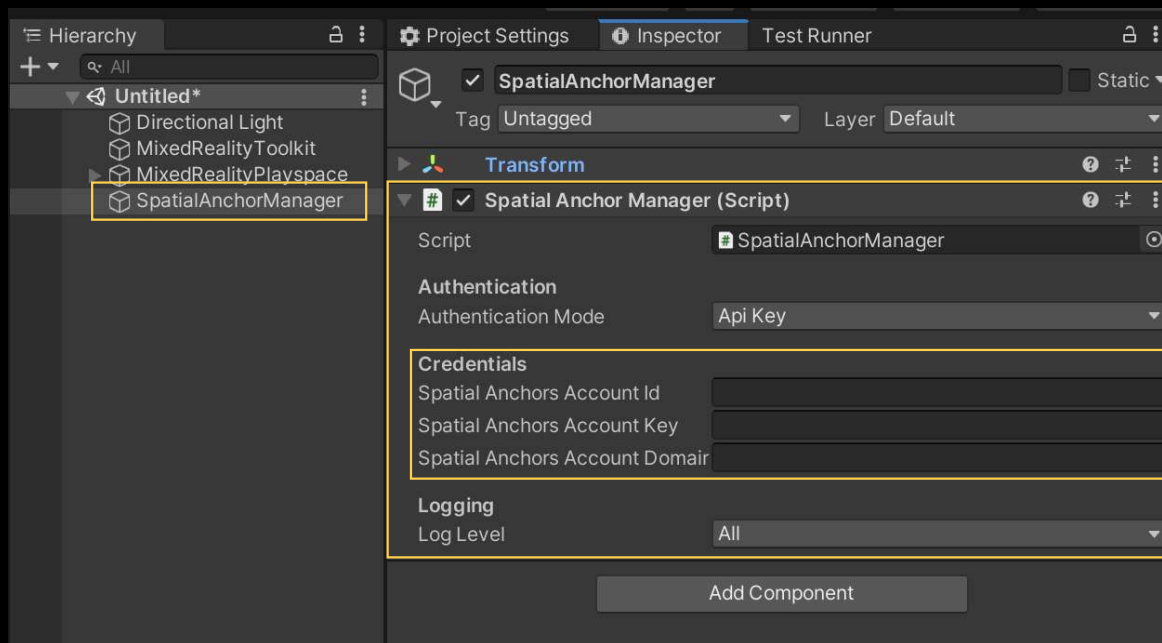
- ✓ Visual Studio Code 1.50.1
- ✓ Plugins: Azure IoT Hub(動作確認用), Remote Development
- ✓ Node.js : v8.11.3(Raspberry PI Beacon化
v10.13.0(Azure IoT Hubとの連携))
- ✓ Python:3.7.3(センサー情報収集、Json化)

Azure Spatial Anchors

～ 基本的な事項～

Azure Spatial Anchorsサービスの導入はPackage Manager経由

詳細な手順はhttps://docs.microsoft.com/ja-jp/azure/spatial-anchors/how-to-setup-unity-project?WT.mc_id=MR-MVP-5003104&tabs=UPMPackage



■ 空GameObjectを追加

■ Spatial Anchor Manager
コンポーネントを追加

Azure Spatial Anchorsの利用に必要な情報を設定

「左記項目に入力」 または 「リソース設定」

※リソースファイルは以下に格納されている
Asset/AzureSpatialAnchors.SDK/Resources/SpatialAnchorConfig

Azure Spatial Anchors

～ 基本的な事項 – 機能系(1/3)～

Azure Spatial Anchorsとの接続と利用定義に関する部品

- `SpatialAnchorManager.CreateSessionAsync`
[メソッド]Azure Spatial Anchorsとの接続に必要なセッションを生成
- `SpatialAnchorManager.StartSessionAsync`
[メソッド]Azure Spatial Anchorsと接続しセッションを開始
- `NearDeviceCriteria`
[クラス]デバイス（センサー系）利用時のルールの定義
- `PlatformLocationProvider`
[クラス]センサー(Beacon,Wifi,GPS)の有効/無効の定義

Azure Spatial Anchors

～ 基本的な事項 – 機能系(2/3) ～

アンカーの設置

- `SpatialAnchorManager.CreateAnchorAsync`
指定のオブジェクトをSpatial Anchorをサービスに登録
 - ✓ 空間に配置したGameObjectをアンカー情報として周辺の空間情報と共に登録
 - ✓ センサー情報も登録する場合はCloudSpatialAnchorSessionに設定必要

アンカーの削除

- `SpatialAnchorManager.DeleteAnchorAsync`
設置済みSpatial Anchorをサービスから登録

Azure Spatial Anchors

～ 基本的な事項 – 機能系(3/3) ～

Azure Spatial Anchorsへのアンカー問合せと設置に関連する部品

- AnchorLocateCriteria
[クラス]Spatial Anchor設置ルールを定義
- CloudSpatialAnchorSession.CreateWatcher
[メソッド]Spatial Anchor設置指示
- SpatialAnchorManager.AnchorLocated
[イベント]サービスから取得したSpatial Anchor設置時に発生
- SpatialAnchorManager.LocateAnchorsCompleted
[イベント]サービスから取得したSpatial Anchorすべてを設置後に発生

Azure Spatial Anchors

～ 基本的な事項 – 設定系(1/2)～

Corse Relocalizationのための設定

NearDeviceCriteria : センサー情報の有効範囲を定義

- DistanceInMeters
 - ✓ センサー位置を中心として有効範囲(半径m)
- MaxResultCount
 - ✓ 一度に検出するSpatial Anchor数

Azure Spatial Anchors

～ 基本的な事項 – 設定系(2/2)～

Corse Relocalizationのための設定

PlatformLocationProvider : センサーの利用条件を設定

- GeoLocationEnabled
 - ✓ GPS情報を利用するかどうか
- WifiEnabled
 - ✓ Wifi情報を利用するかどうか
- BluetoothEnabled
 - ✓ Bluetooth（ビーコン）を利用するかどうか
- KnownBeaconProximityUuids
 - ✓ ビーコンを利用する場合のUUIDのホワイトリスト

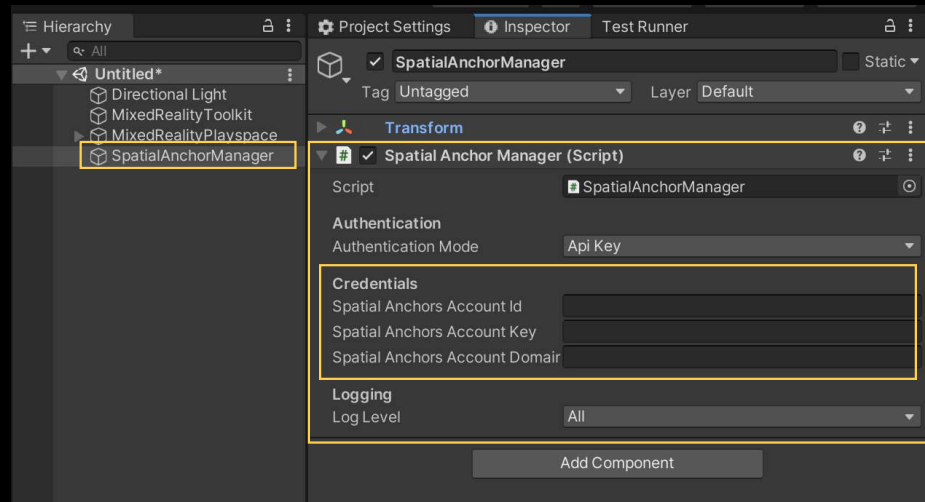
Azure Spatial Anchors

～ 実装：セッションの開始～

セッションの生成と開始を実施

//セッションの開始

```
await CloudManager.CreateSessionAsync();  
await CloudManager.StartSessionAsync();
```



Azure Spatial Anchorsサービスへの接続はCredentials
Or

リソースから取得

Asset/AzureSpatialAnchors.SDK/Resources/SpatialAnchorConfig

Azure Spatial Anchors

～ 実装：検索の実施(1/4)～

イベントの定義

SpatialAnchorManagerのイベントにメソッドを割り当てる

//イベントの設定

```
SpatialAnchorManager.AnchorLocated += AnchorLocated;
```

```
SpatialAnchorManager.LocateAnchorsCompleted += LocateAnchorsCompleted;
```

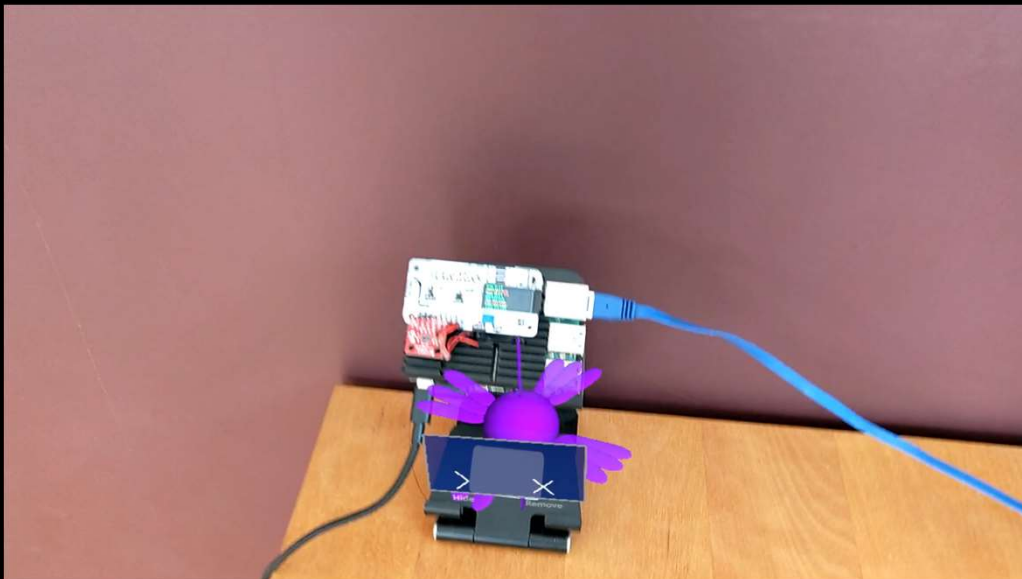
```
SpatialAnchorManager.LogDebug += LogDebug;
```

```
SpatialAnchorManager.Error += Error;
```

Azure Spatial Anchors

～ 実装：検索の実施 (4/4)～

アンカー検索の実施(非同期)



```
// 範囲を設定するための設定
// 検索の開始
NearDeviceCriteria nearDeviceCriteria
CloudManager.SessionProvider
= new NearDeviceCriteria(provider);
provider.Sensors.GeoLocationEnabled = false;
nearDeviceCriteria.DistanceInMeters = 8;
nearDeviceCriteria.MaxResults = 25;
provider.Sensors.BluetoothEnabled = true;
provider.Sensors.KnownBeaconProximityUids
var anchorLocateCriteria;
= new string[] { Criteria };
指定Anchorに合致する登録済みSpatial Anchorを検索
CloudManager.SessionProvider
= nearDeviceCriteria;
配置のタイムアウトが経過した場合
✓ SpatialAnchorManager.AnchorLocated
✓ SpatialAnchorManager.LocateAnchorsCompleted
```


Azure Spatial Anchors

～ 実装：登録 ～

Spatial Anchorの登録

通常のGameObjectで位置決めを実施



```
// アンカー情報の取得
CloudNativeAnchor cna =
spawnedObject.GetComponent<CloudNativeAnchor>();

if (cna.CloudAnchor == null) { cna.NativeToCloud(); }

CloudSpatialAnchor cloudAnchor = cna.CloudAnchor;

//アンカーの寿命を設定（検証では有限に！）
cloudAnchor.Expiration = DateTimeOffset.Now.AddDays(7);

//アンカーの登録
await CloudManager.CreateAnchorAsync(cloudAnchor);
```

まとめと考察

現実空間にセンサー情報を可視化する意義

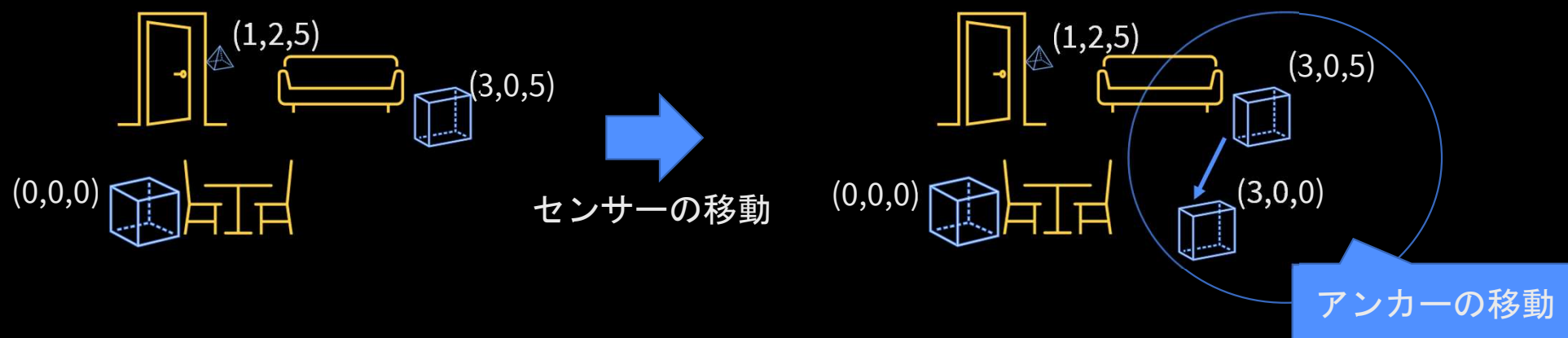
作業支援として一定の効果はあると考えられる

- ✓ センサー情報＋現実空間（空間認識など）を組み合わせるデジタルツインも可能
- ✓ 現物確認が必要なシチュエーションでは可視化による効率化が可能



現実空間にセンサー情報を可視化する意義

デジタル空間でのセンサー位置の管理の実現



Azure Spatial Anchorsにより現実空間のセンサー位置をUnityの座標系で管理
設置センサーの後付けや移動による座標情報までDigital Twinsで管理が可能

本日のテーマ

情報の可視化を現実空間で実施

機器メンテナンスを想定して対象の特定、センサー情報を補助空間に出力



Azure Spatial Anchorsを中心にDigital Twinsを実現する手段の紹介

デモの紹介と考察



ご清聴ありがとうございました