



Forge Online

Viewer アップデート Overlay Manager と Scene Builder

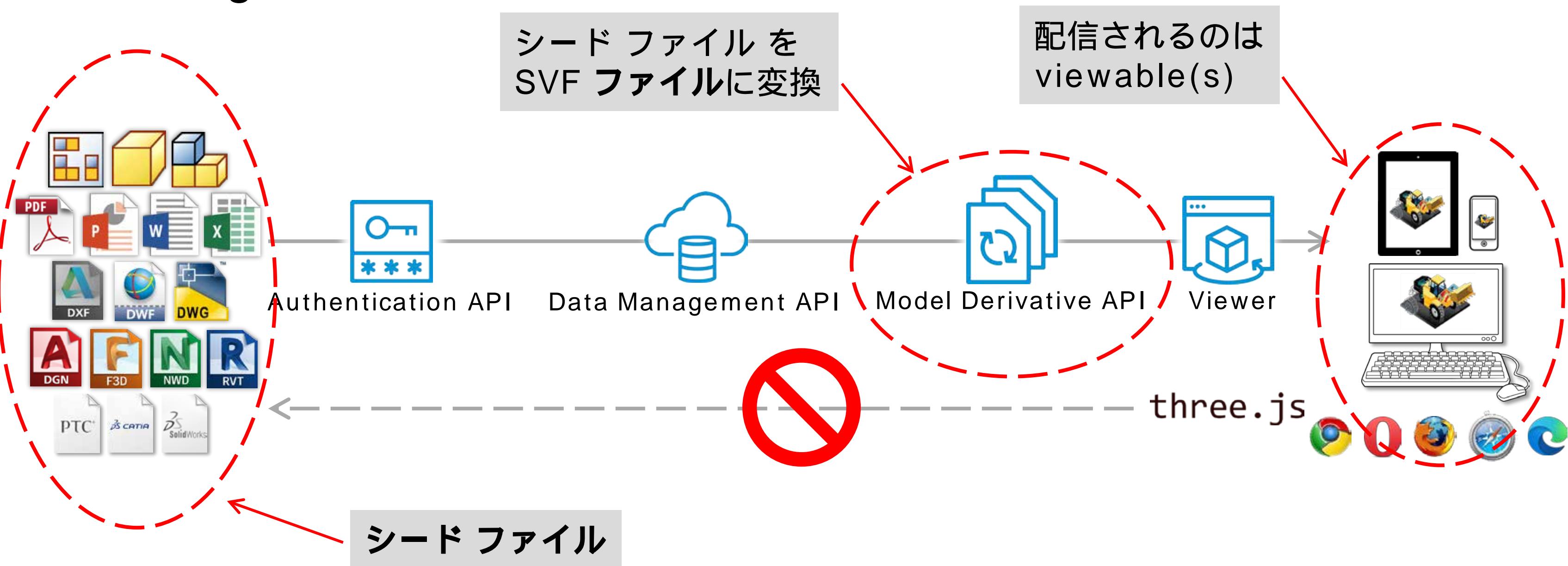
伊勢崎俊明

オートデスク 株式会社

Forge Viewer ソリューションのながれ

・・・

§ Forge Viewer からのオリジナルデータへの反映は不可



Forge Viewer 'JavaScript' ライブラリ



Forge Viewer

CAD/CG ソフトウェアで
作成したデザインを変換して
表示/操作するユーティリティ

three.js



プログラムでシーンや 照明、
メッシュ モデルを定義して
作成するのが一般的な利用法

シードファイルがないモデルの表示

§ Viewer カンバスへのカスタム メッシュ表示

1. シーンへのオーバーレイ（重ね合わせ表示）
2. シーンへの描画

The screenshot shows the Autodesk Forge Developer's Guide for 'Add Custom Geometry'. The left sidebar has a 'Developer's Guide' section expanded, showing 'Overview', 'Viewer Essentials', 'Advanced Options' (with 'Add Custom Geometry' highlighted), 'Query the Property Database', 'Profiles', 'Scene Builder' (with a blue arrow labeled '2.' pointing to it), and 'Set Up Edit2D'. The main content area title is 'Add Custom Geometry'. It explains that the Viewer allows adding small custom geometry using the `viewer.overlay` API. It describes how to overlay additional data onto a loaded model, noting that every custom geometry added to the overlay scene will be rendered on every frame even with progressive rendering. It also mentions that framerate will decline with many custom geometries. Finally, it states that custom geometry uses the main scene depth buffer for depth testing.

AUTODES FORGE Success Stories Solutions v Getting Started Documentation Community v Support v Pricing SIGN IN

Viewer v7 v

Developer's Guide

Overview

> Viewer Essentials

Advanced Options

Add Custom Geometry **1.**

Query the Property Database

Profiles

Scene Builder **2.**

Set Up Edit2D

Add Custom Geometry

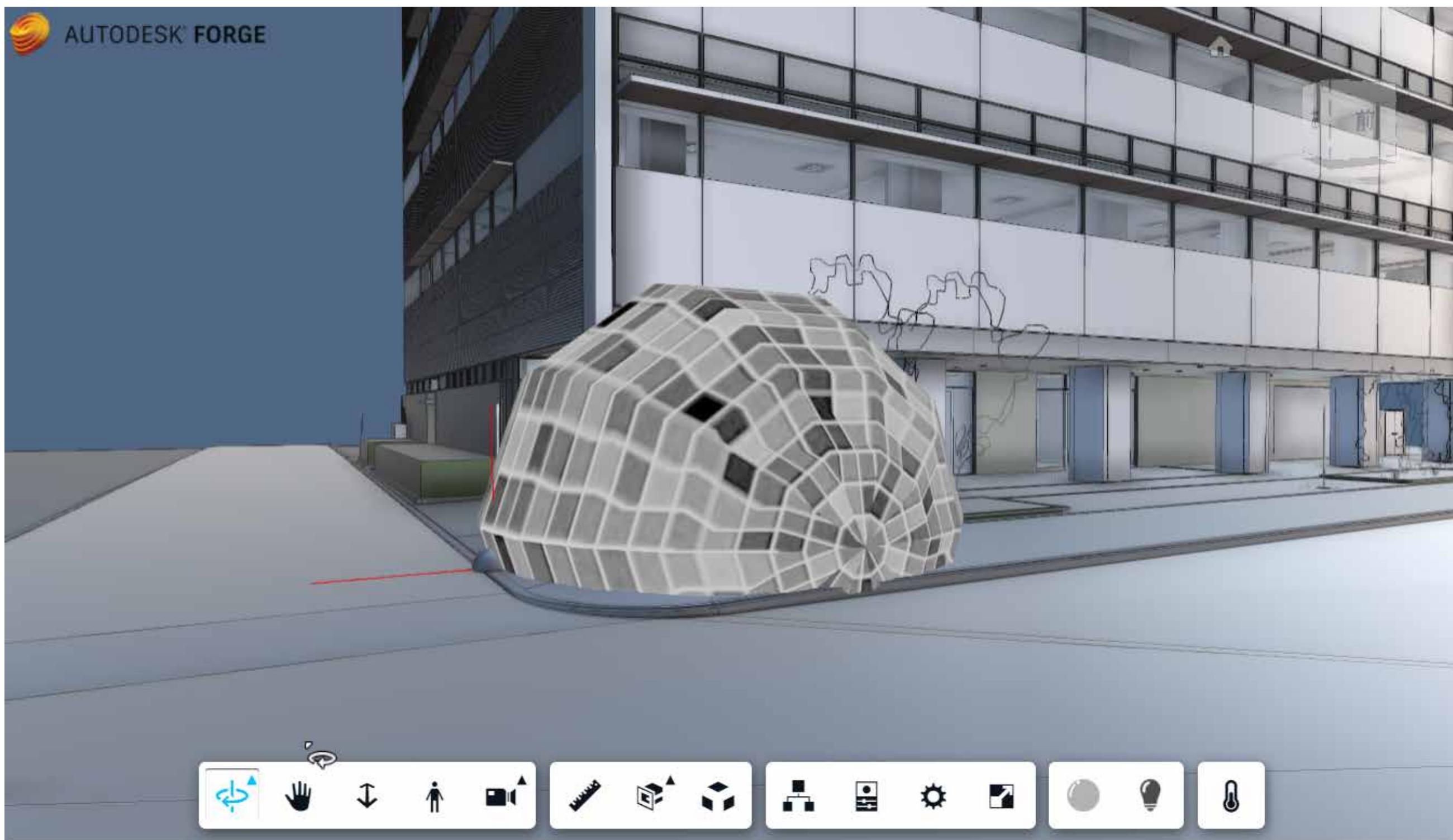
The Viewer allows you to add small custom geometry into a scene by using the `viewer.overlay` api.

Use this feature to overlay additional data to the loaded model. Every custom geometry added into the overlay scene will get rendered on every frame, even when progressive rendering is on.

Framerate will decline when too many custom geometries are added into overlay scenes.

Custom geometry uses the main scene depth buffer for depth testing, allowing the custom geometry to appear within the loaded model.

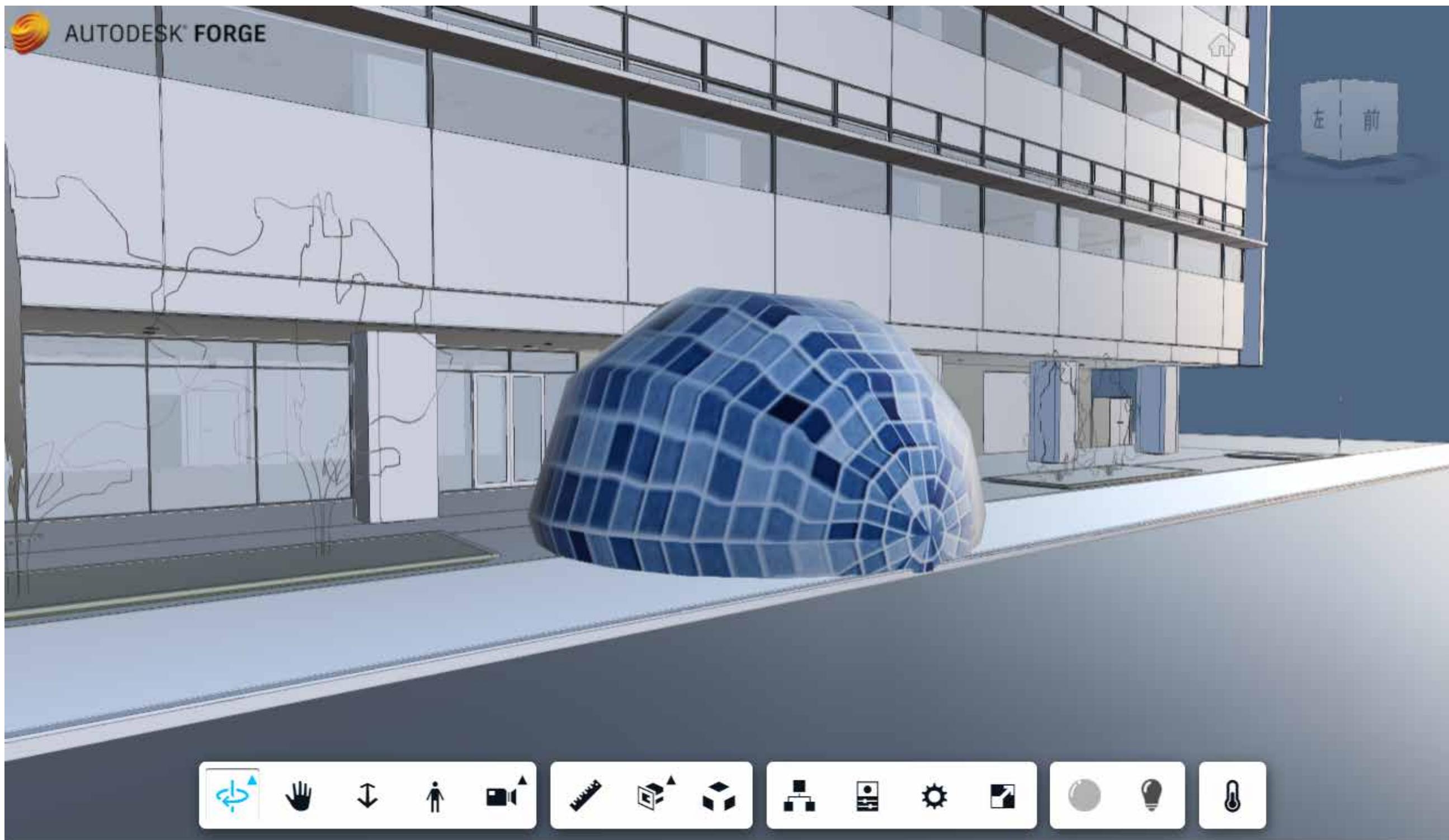
カスタム メッシュのオーバーレイ



カスタム メッシュのオーバーレイ

```
function addScene() {  
    _scene = new sceneMember(  
        new THREE.SphereGeometry(11, 10, 10),  
        new THREE.MeshBasicMaterial({ color: 0xa6a6a6 })  
    );  
  
    _scene.mesh = new THREE.Mesh(_scene.geom, _scene.material);  
    _scene.mesh.position.set(-37, -33, -33);  
    _scene.material.map =  
        THREE.ImageUtils.loadTexture('images/Finishes.Tiling.Ceramic.Mosaic.Grey.png');  
    if (!_viewer.overlays.hasScene('custom-scene')) {  
        _viewer.overlays.addScene('custom-scene');  
    }  
    _viewer.overlays.addMesh(_scene.mesh, 'custom-scene');  
}
```

シーンビルダーでのカスタム メッシュ描画

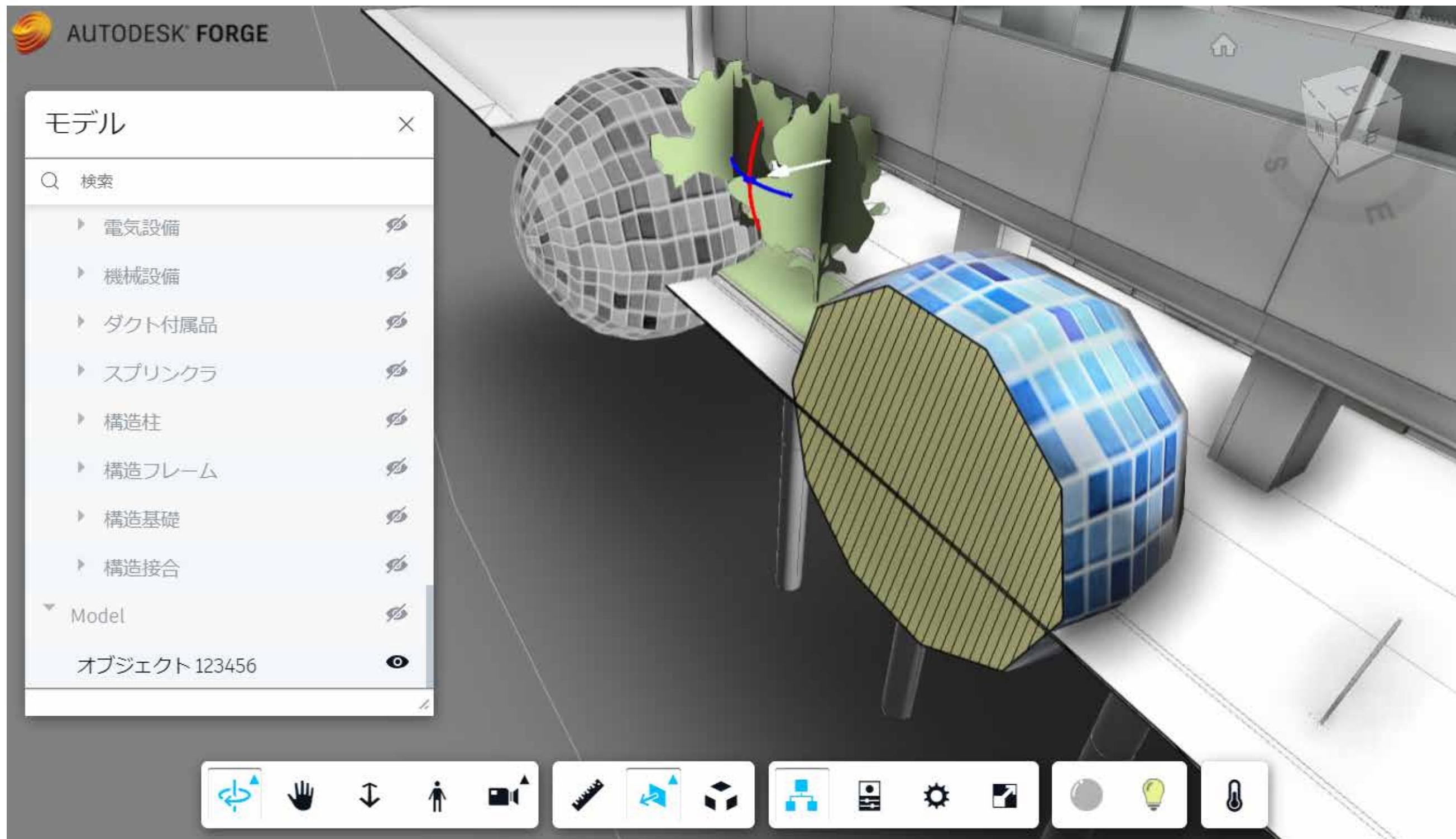


シーンビルダーでのカスタムメッシュ描画

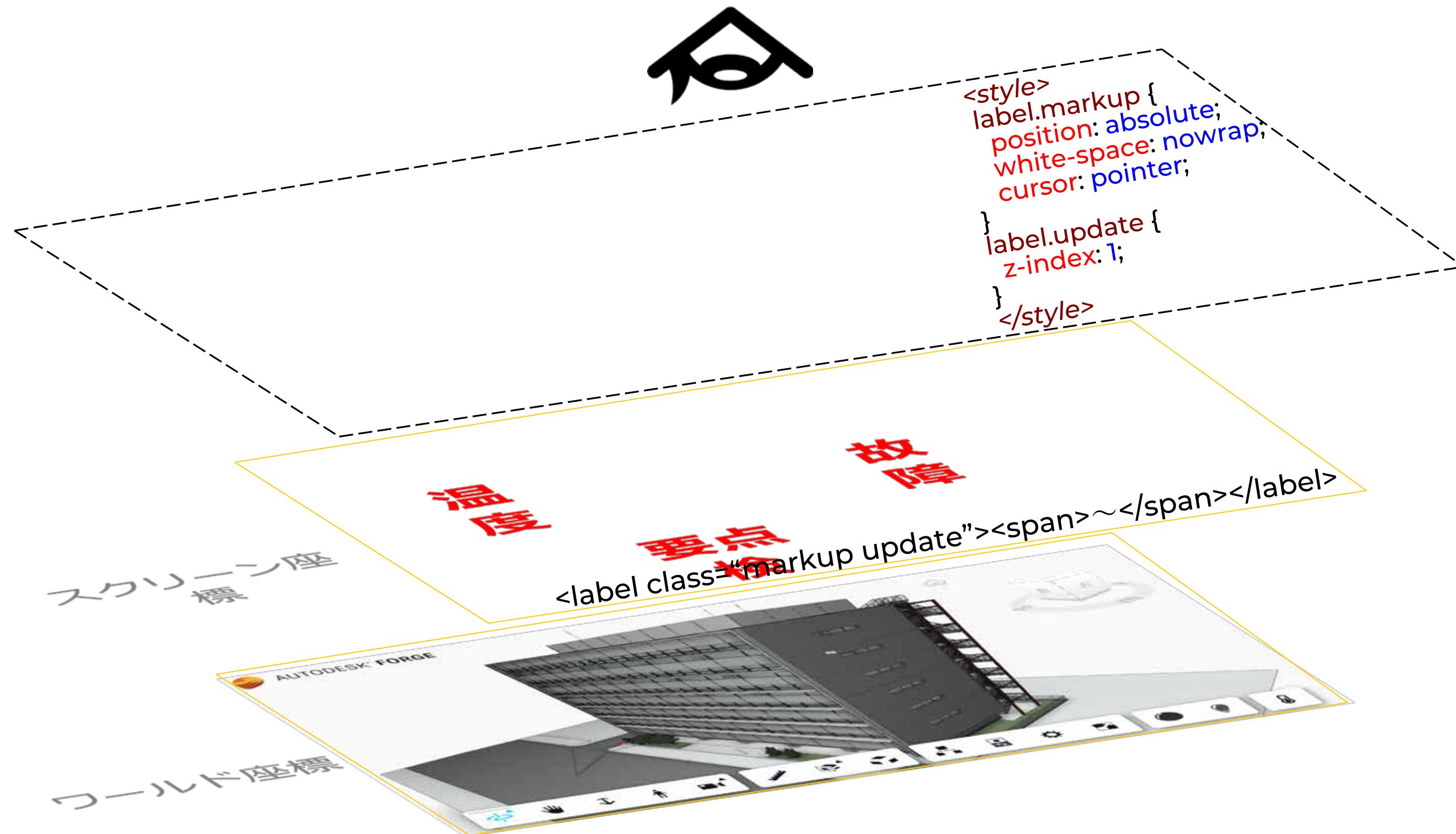
```
async function addModel() {
  const sceneBuilder = await _viewer.loadExtension('Autodesk.Viewing.SceneBuilder');
  _modelBuilder =
    await sceneBuilder.addNewModel({ conserveMemory: false, modelNameOverride: 'custom-model' });

  const sphereGeometry =
    new THREE.BufferGeometry().fromGeometry(new THREE.SphereGeometry(11, 10, 10));
  const sphereMaterial = new THREE.MeshPhongMaterial();
  sphereMaterial.map = THREE.ImageUtils.loadTexture('images/Finishes.Tiling.Ceramic.Mosaic.Blue.png');
  const sphereMesh = new THREE.Mesh(sphereGeometry, sphereMaterial);
  sphereMesh.matrix = new THREE.Matrix4().compose(
    new THREE.Vector3(0, -33, -33),
    new THREE.Quaternion(0, 0, 0, 0),
    new THREE.Vector3(1, 1, 1)
  );
  sphereMesh.dbId = 123456;
  _modelBuilder.addMesh(sphereMesh);
}
```

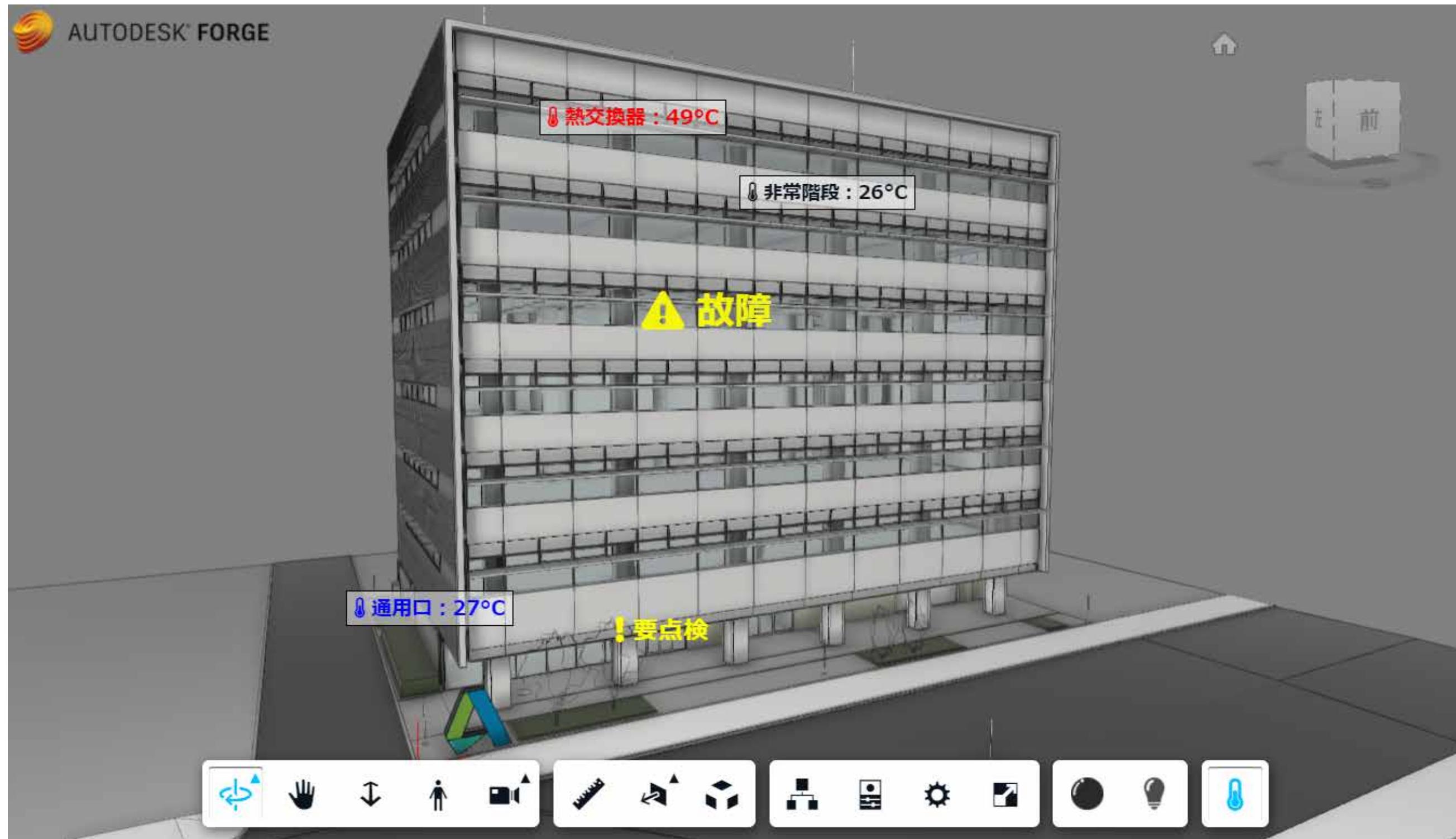
表現メッシュの差異



HTML要素の z-index を使ったオーバーレイ



Label 要素を使ったマークアップ





Forge Online

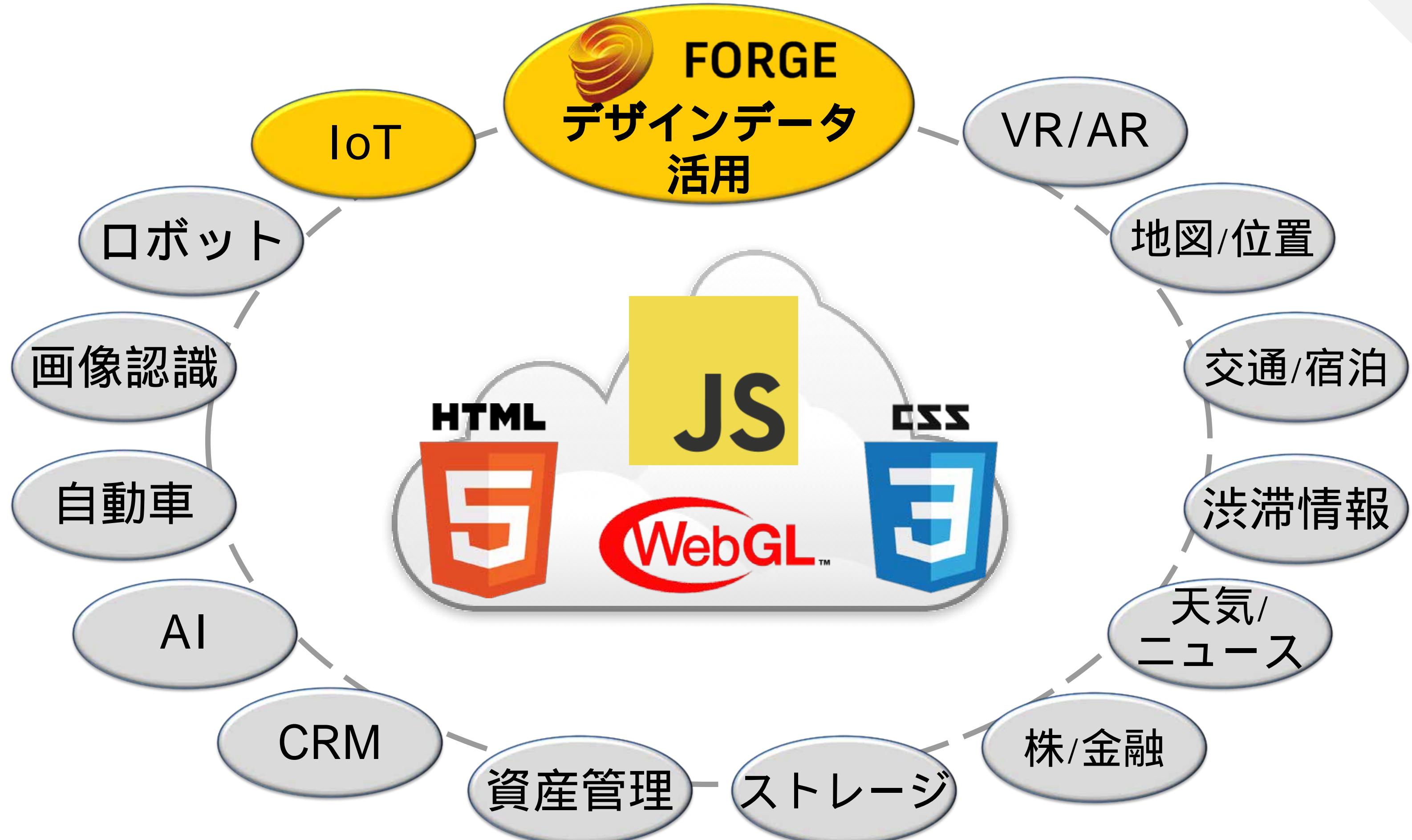
Viewer アップデート

デジタルツインの取り組みと

Data Visualization エクステンション

伊勢崎俊明

オートデスク 株式会社



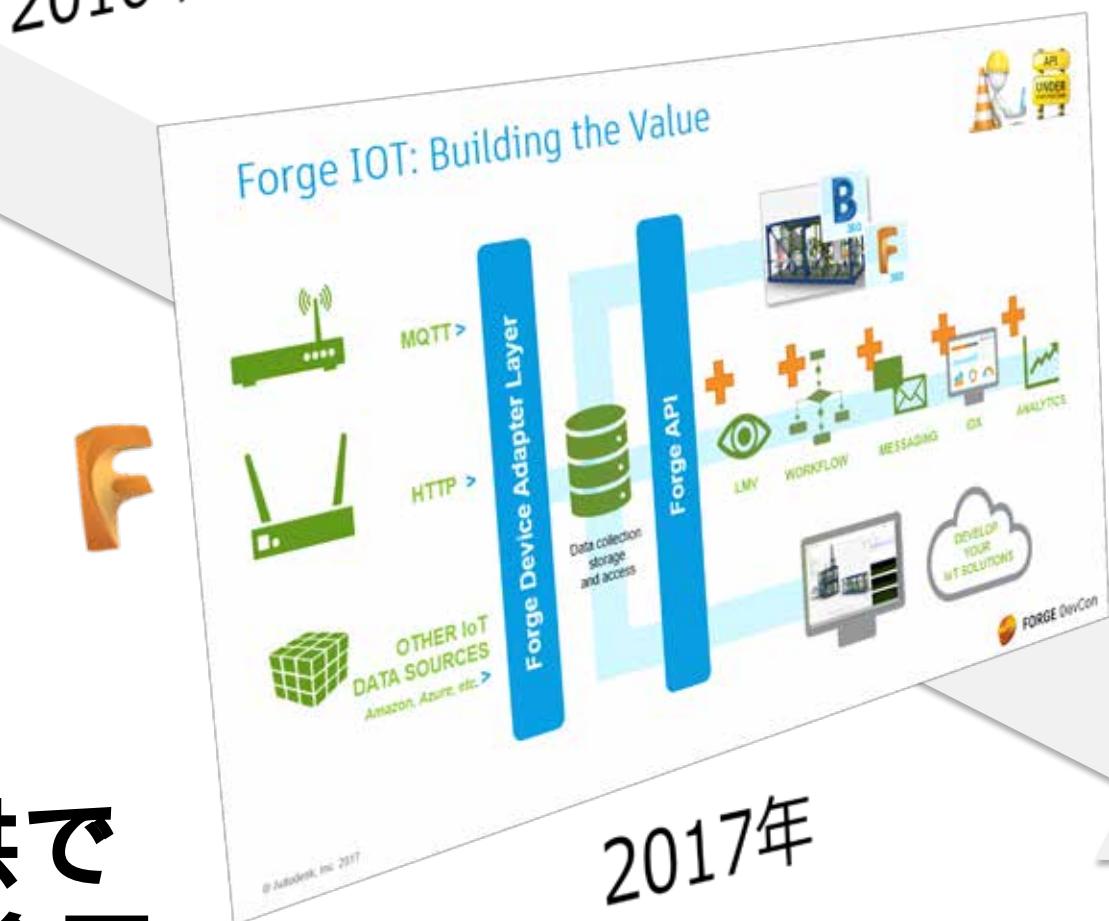


2015年



FORGE

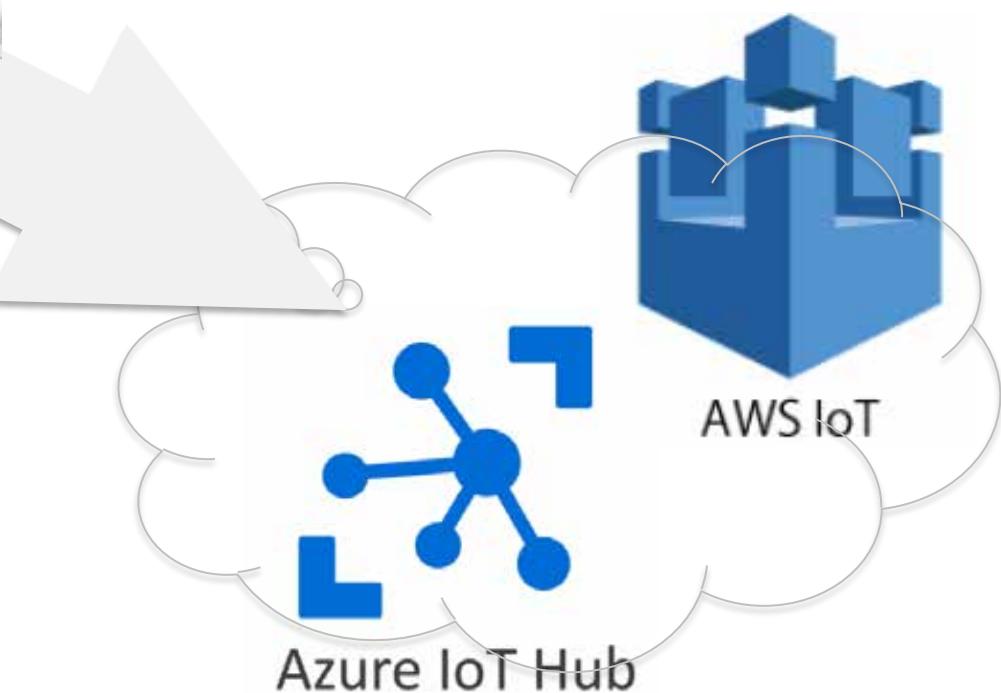
2016年



2017年

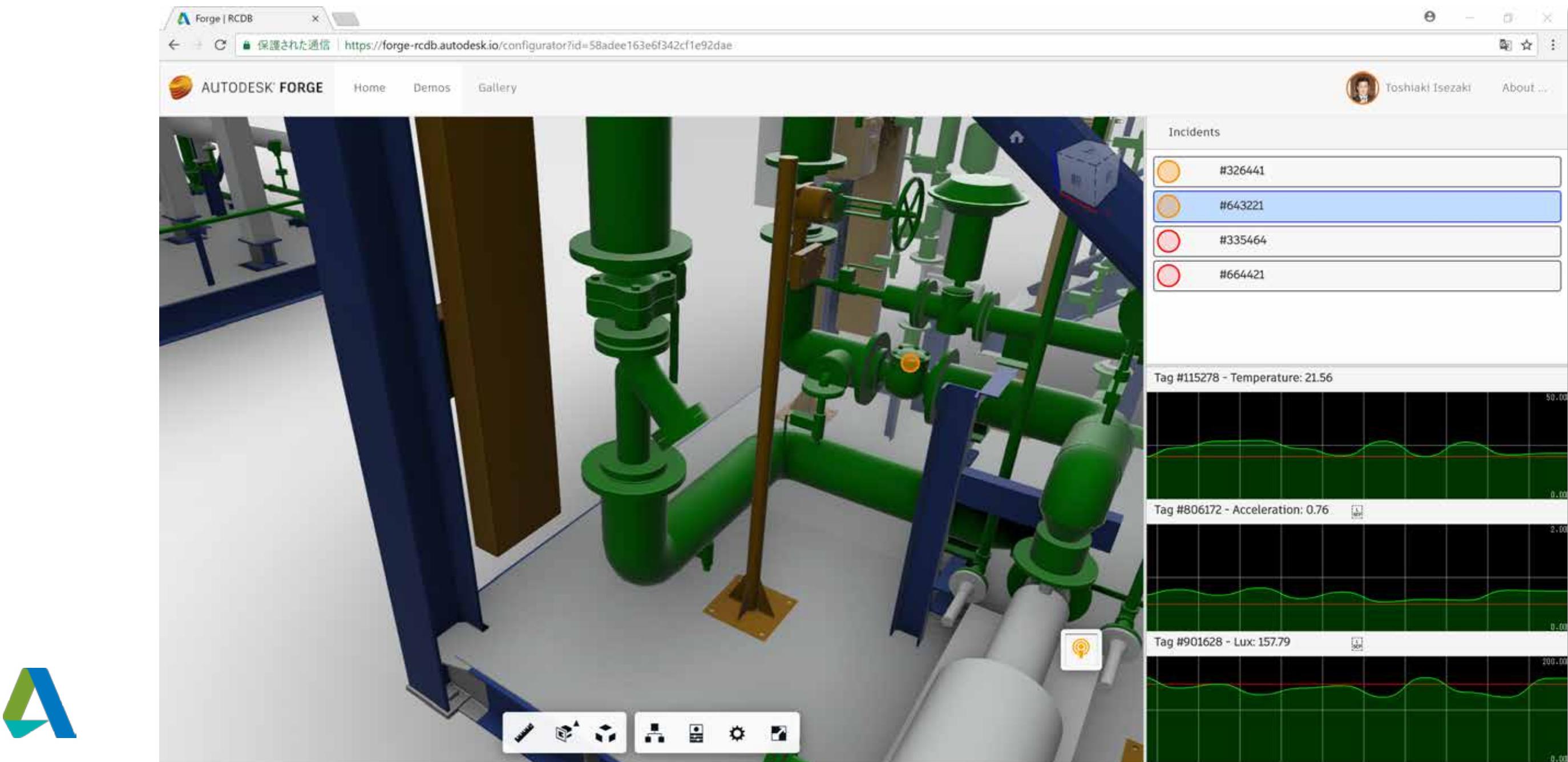
クラウドバックエンド提供で
IoT デバイス管理/運用を企図

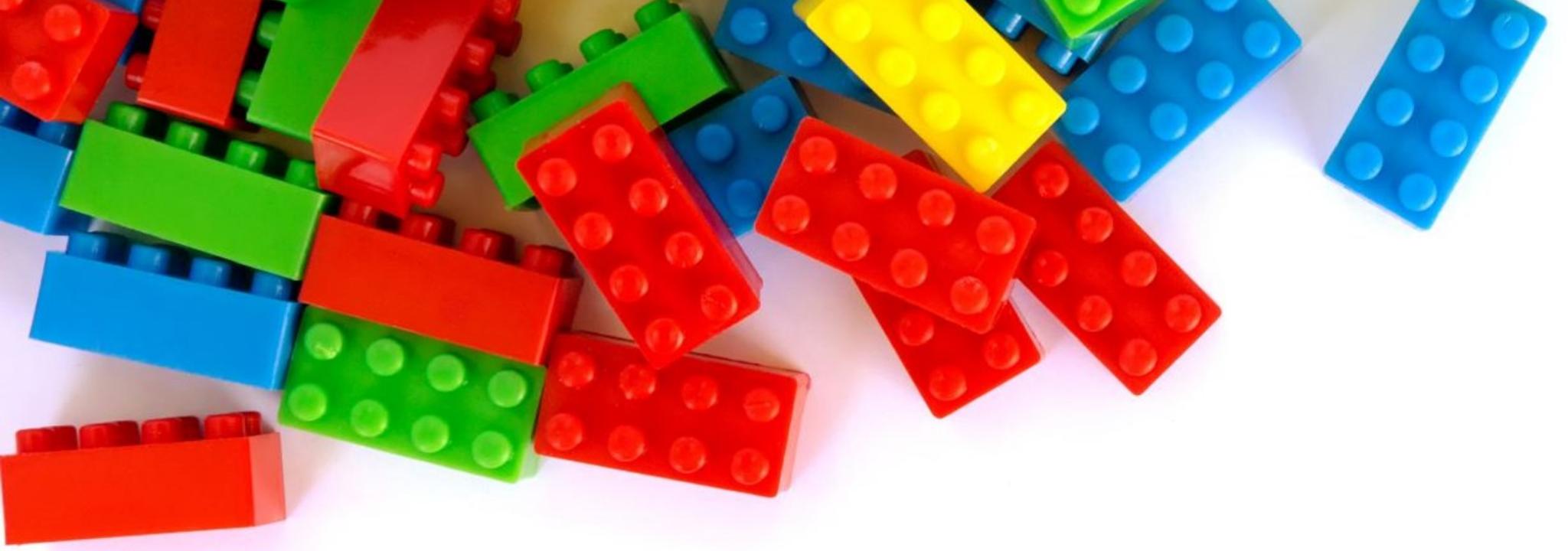
IoT 戦略の推移



センサー情報のリアルタイム取得

§ WebSocket（リアルタイム双方向通信テクノロジ）を利用





Forge の使命はイノベーションを可能にする
ビルディング ブロックを皆様に提供することです。

Data Visualization エクステンション

- § React ベースのリファレンスアプリ
 - § オープンソースのウィジェットライブラリ (Apache echart)
- § IoT ハブへのデータアダプタ (Azure、AWS)
- § IoT の視覚化
 - § ドット (センサー)
 - § ヒートマップ
 - § タイムライン





Nov 2020

8th 9th 10th 11th 12th 13th 14th 15th 16th 17th 18th 19th 20th 21st 22nd 23rd 24th 25th 26th 27th 28th 29th 30th Dec 1st 2nd

Nov 13, 2020



If you're interested in learning more, we'd love to hear from you!

Please note that this is a Private Tech Preview and is subject to additional terms.

[CONTACT US](#)[LEARN MORE](#)

Sens

Search

▼ 01 - ENTRY LEVEL

Lobby 102 North

21.60 celsius

490.80 ppm

Lobby 102 South

21.64 celsius

492.38 ppm

Cafeteria 121 East

21.58 celsius

490.54 ppm



Web 開発者の方への利点

- § 開発時間の短縮（helloWorldまでの時間）
- § より低レベルのグラフィックスの専門知識が必要
 - § 複雑さを軽減する
- § 全体的なエクスペリエンスの向上



The screenshot shows the Autodesk Forge API Reference page for the `DataVisualization` extension. The left sidebar lists various API categories and extensions, with `DataVisualization` currently selected. The main content area displays the `DataVisualization` class documentation, including its purpose, constructor, parameters, and a code example.

Autodesk.Viewing.Extensions

DataVisualization

Data Visualization extension provides an API for building scenes with custom sprites (dots) and heatmaps (surface shading) within model boundaries. The extension id is: `Autodesk.DataVisualization`

`new DataVisualization(viewer, options)`

Parameters

viewer *	Viewer instance
<code>Viewer3D</code>	

options *	Configurations for the extension
<code>object</code>	

Device
Room
LevelRoomsMap
ModelStructureInfo
ViewableStyle
CustomViewable
SpriteViewable
ViewableData



AUTODESK®

Make anything.