

PM473593

FusionとForgeを用いたCNCツール  
パスの最適化。IoTによるリアルタ  
イムでの製造データ収集のデモ

**Nathan C. Skalsky**

Senior Product Manager, Autodesk Forge | @speedbirdpdx

**Varun Patil**

Developer Technical Consultant, Autodesk Forge



## スピーカーについて

### Nathan Skalsky

Nathan Skalskyは、オートデスクのForgeのシニアプロダクトマネージャであり、モノづくりを愛しています。自動化と簡素化により、人間の創造性をより効率的に表現できるワークフローと機能を実現しています。

IBMでは大きなシステムの一部分の設計を担当し、その後、Fusion 360のソフトウェア開発マネージャーや、スタートアップも経験しました。現在はオートデスクでFusionのプロダクトマネージャをしています。



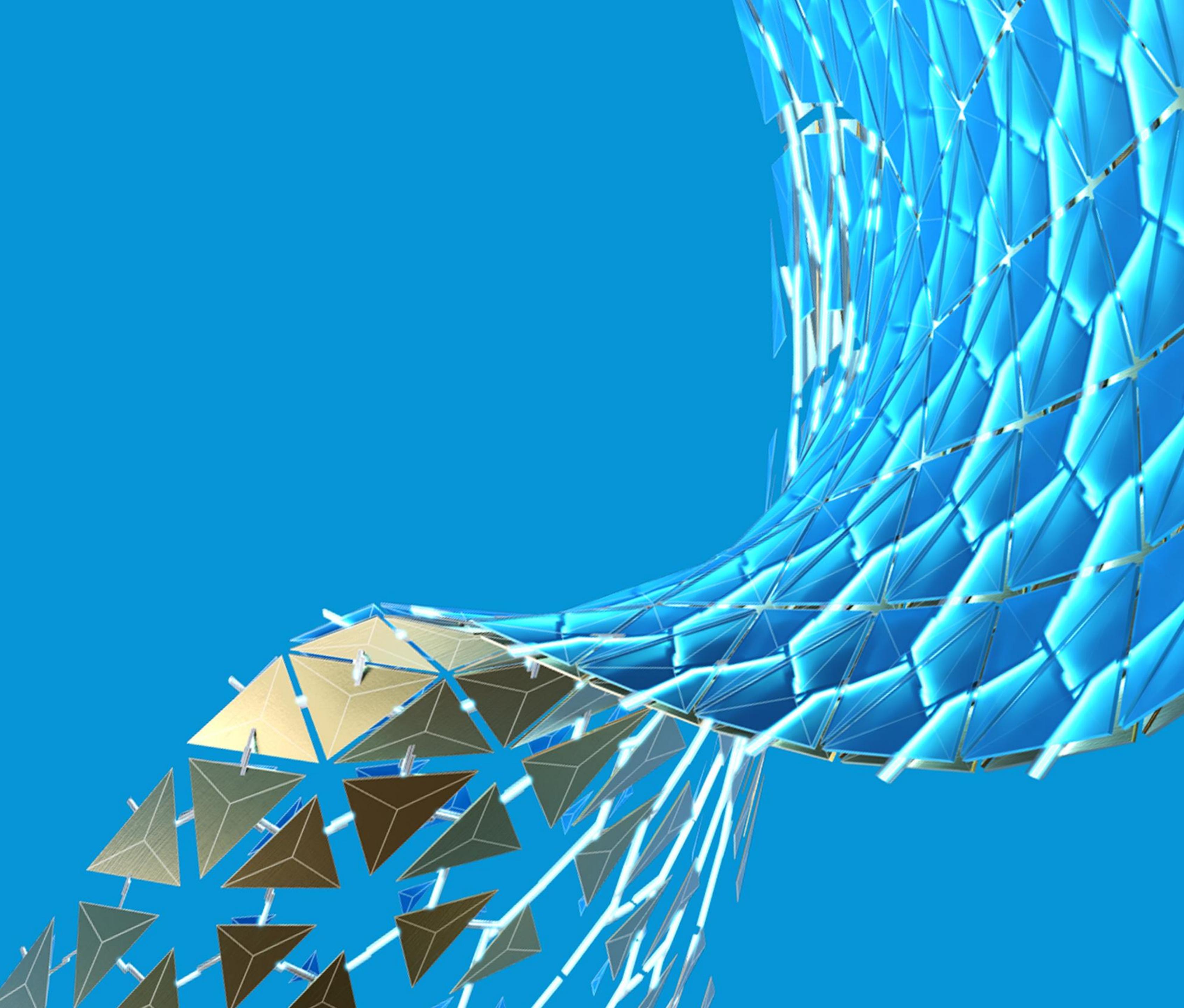
## スピーカーについて

### Varun Patil

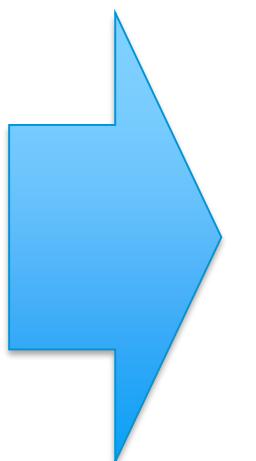
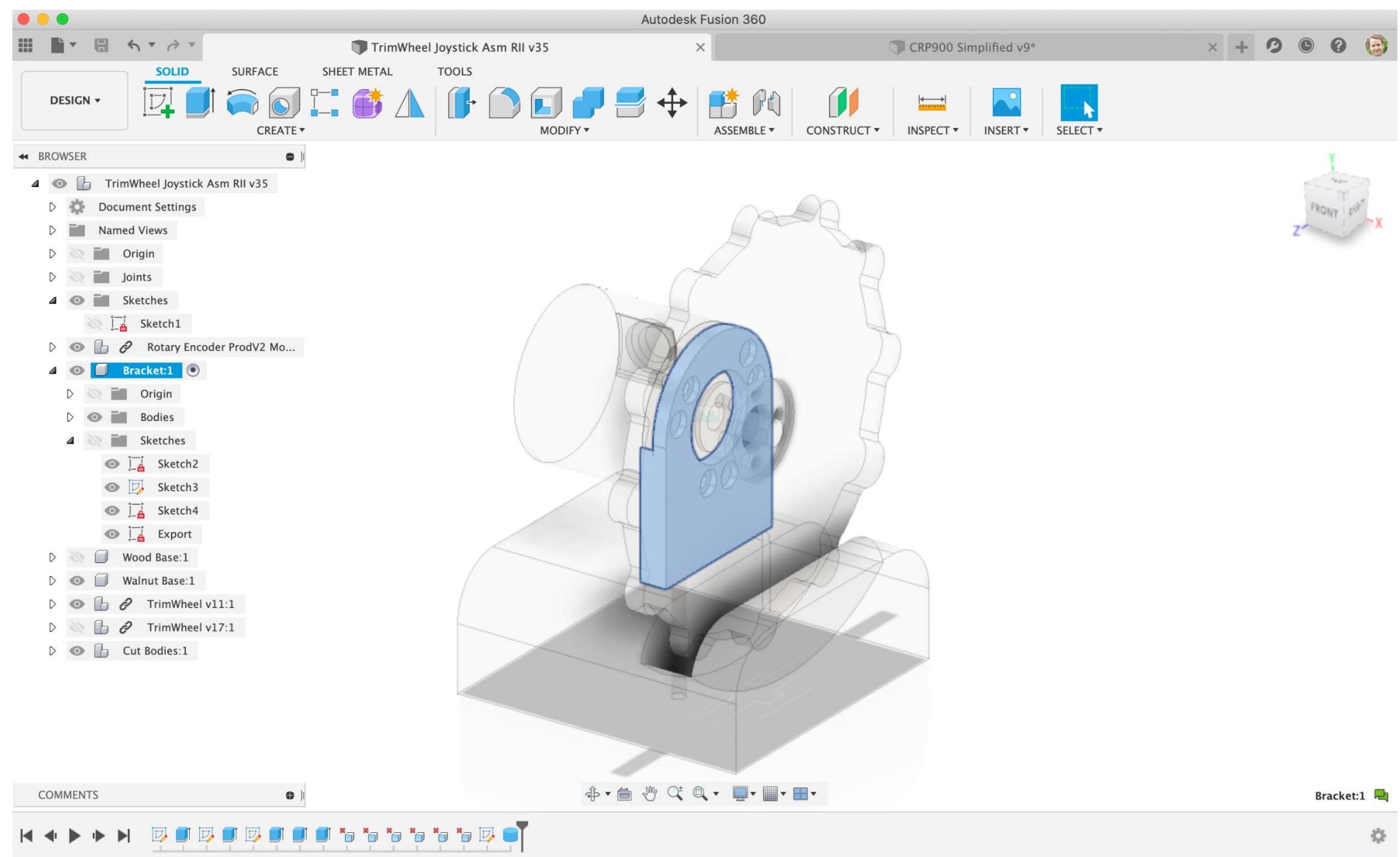
Varun Patilは、フルスタック開発者であり、Forge開発チームの一員としてオートデスクで2年半以上勤務。

Forge APIを用いたサンプルプロジェクトの開発を通して、お客様の支援をしています。

# 問題領域



# Fusion 360、ForgeおよびCNC工作機械を使ったより良いモノづくり



Fusion 360, 3D CAD, CAM、他色々

CNC ルータ (Benchtop PRO 2'x2')

# メーカーのジレンマ

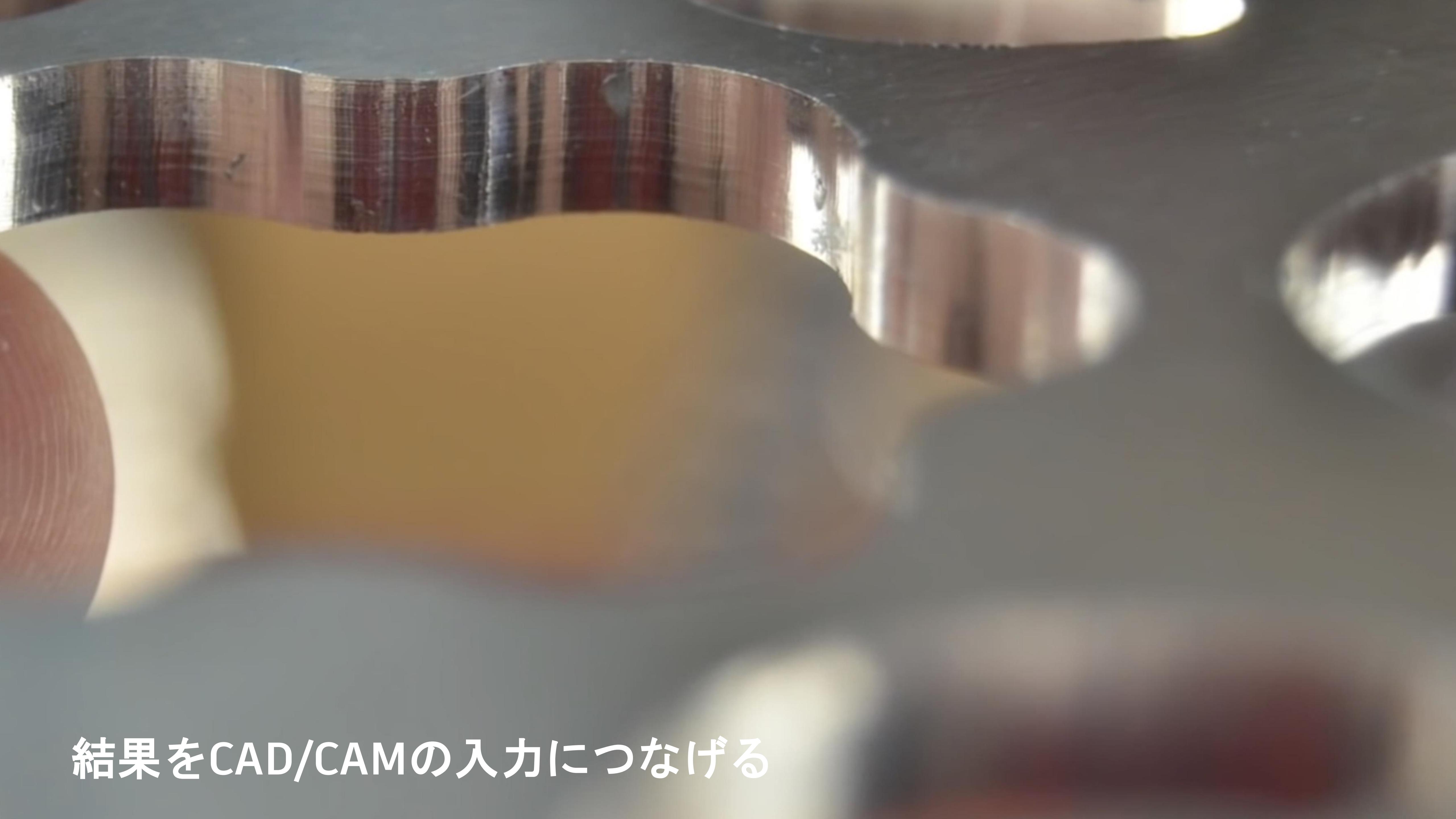
設計から製造までのワークフロー (CNC、3DP、レーザー) は、主に一方向のプロセス。

製造結果から、確実にフィードバックを得る方法は？

- ループを閉じる、別名「Forge APIを用いた軽量なプロセス制御」
- 結果から学習する、意味のある結果をキャプチャ
- より良いものを作る！ワークフローが改善することにより、よりよい結果が生まれる

The screenshot shows the Autodesk Fusion 360 interface with the following details:

- Project:** TrimWheel
- Part:** TrimWheel Joystick Asm RII
- History:** The part was last updated at 9:58:51 PM yesterday by Nathan Skalsky. It shows 34 steps in total, starting from 6/16/20, 10:38:53 PM and ending at 9/16/20, 9:58:51 PM.
- Tool:** #1 - Ø1/4" flat (L...)
- Coolant:** Flood
- Feed & Speed:**
  - Preset: Custom
  - Spindle Speed: 23000 rpm
  - Surface Speed: 1505.35 ft/min
  - Ramp Spindle Speed: 23000 rpm
  - Cutting Feedrate: 100 in/min
  - Feed per Tooth: 0.00217391 in
  - Lead-In Feedrate: 80 in/min
  - Lead-Out Feedrate: 80 in/min
  - Ramp Feedrate: 80 in/min
  - Plunge Feedrate: 80 in/min
  - Feed per Revolution: 0.00347826 in
- Operations Preview:** Shows the sequence of operations: [T1] 2D Pocket2, [T1] 2D Pocket2 (2), [T1] 2D Contour3, Bracket Bottom, and [T2] Drill3 [Deep dri...].



結果をCAD/CAMの入力につなげる

# Feedback brainstorm for simplified Process Control

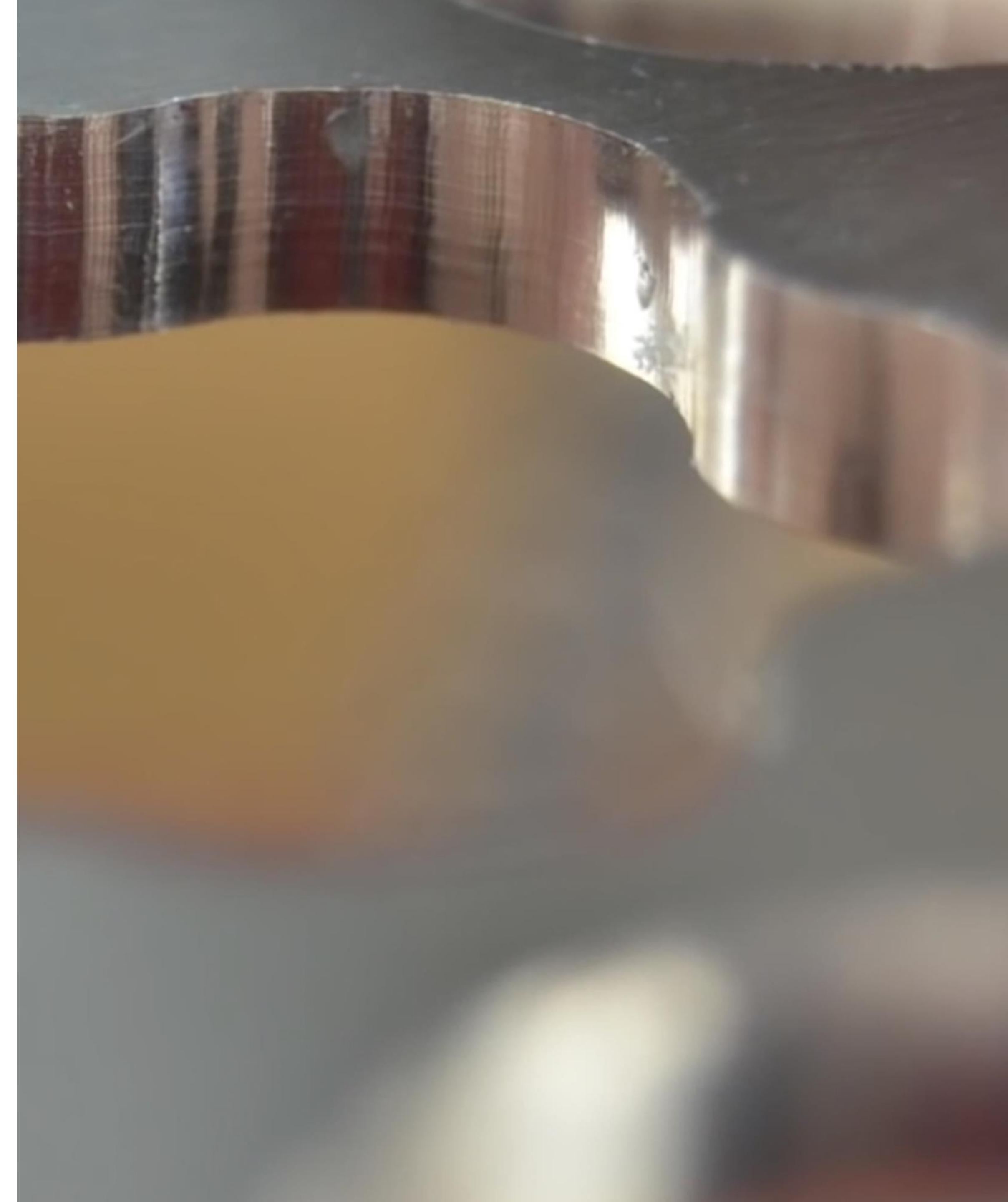
自動化されたフィードバック:

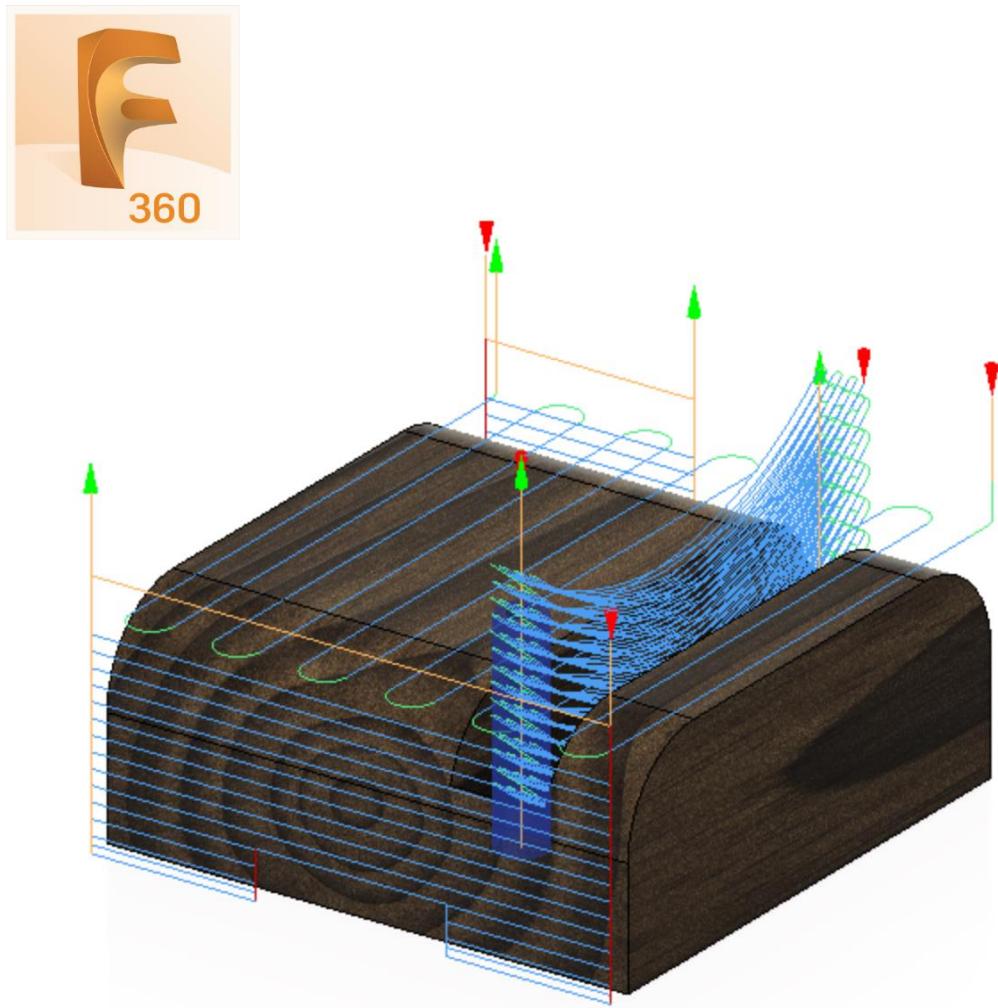
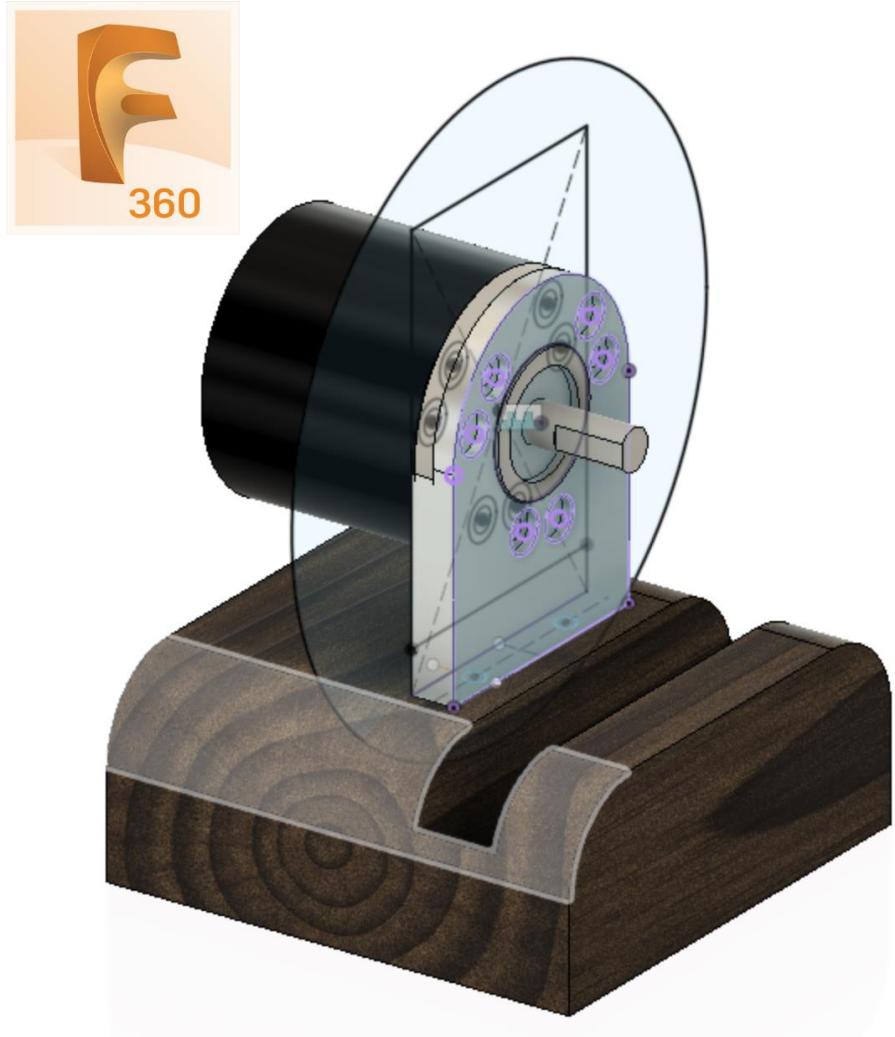
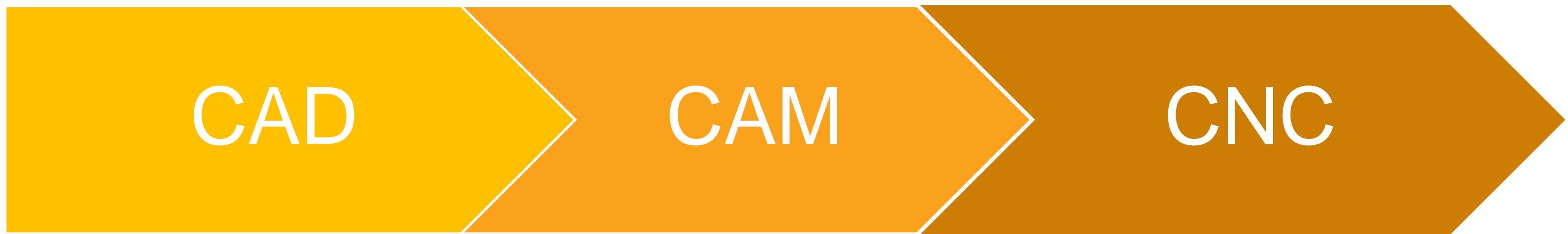
- マシンビジョン
- 音
- > • **温度**
- サーボ/PLC(Programmable Logic Controller)/CMM フィードバック

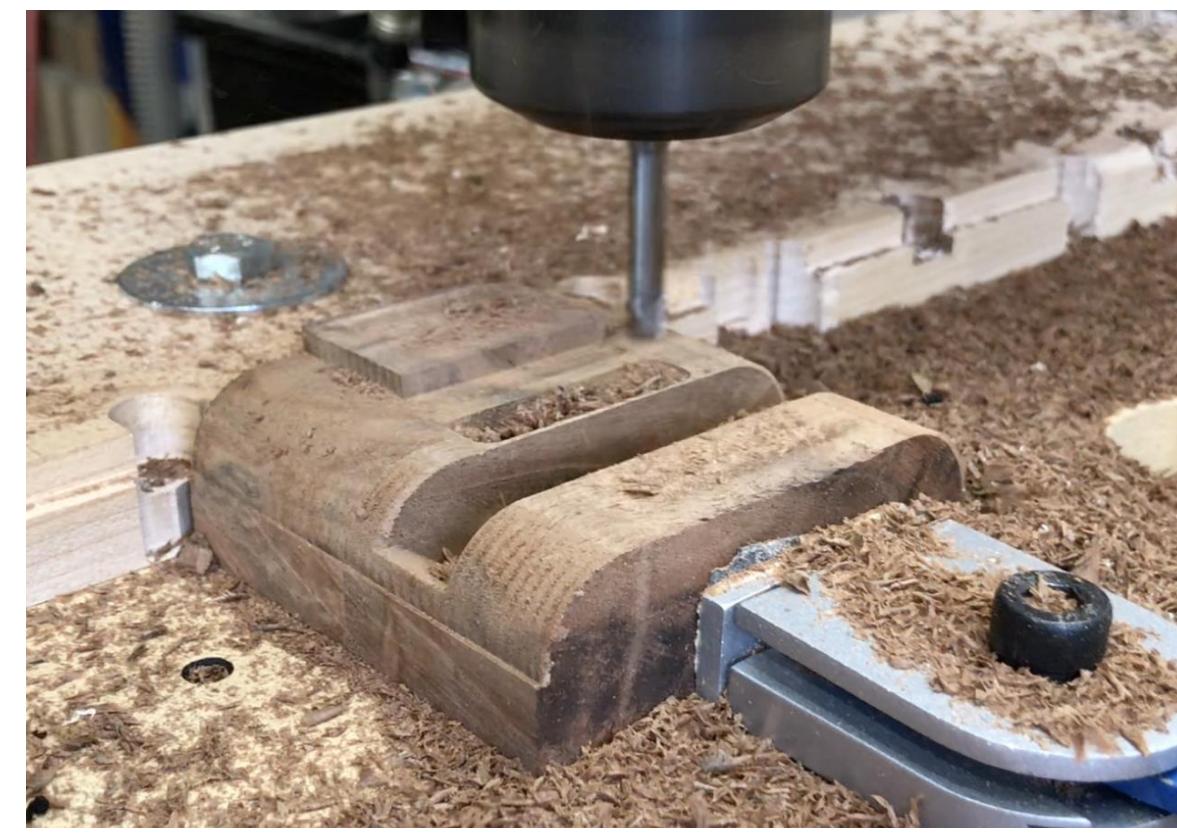
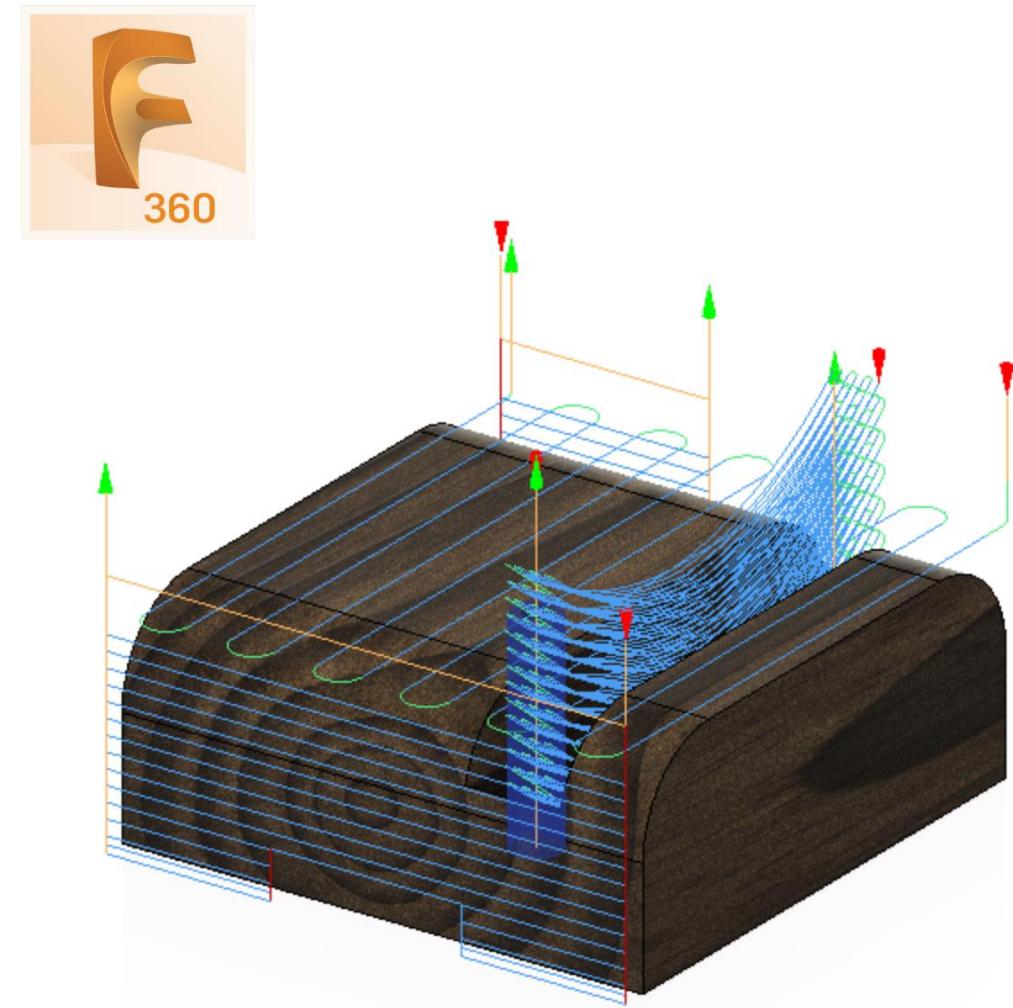
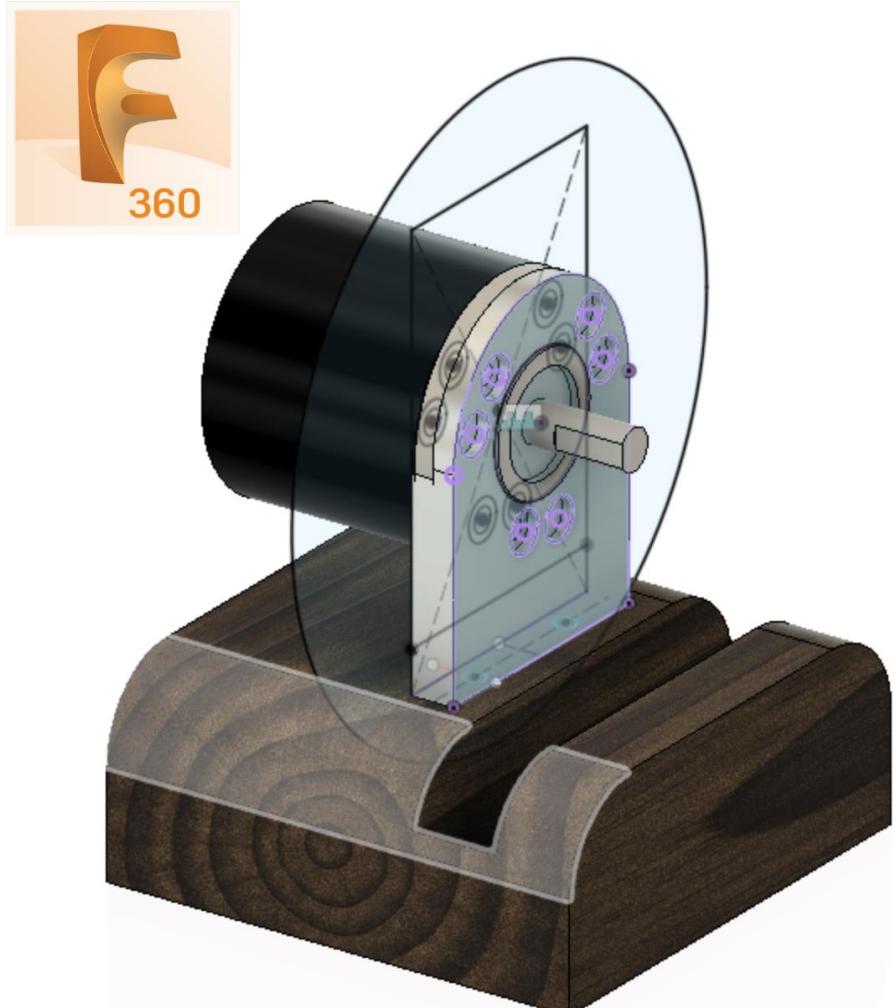
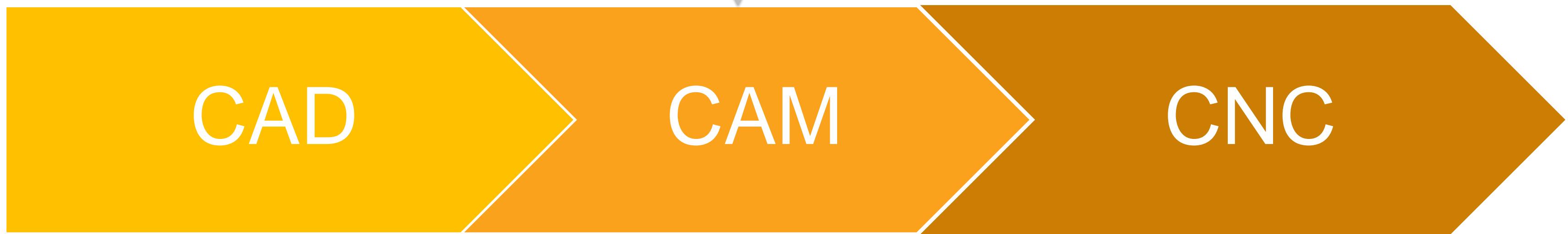
手作業による分析:

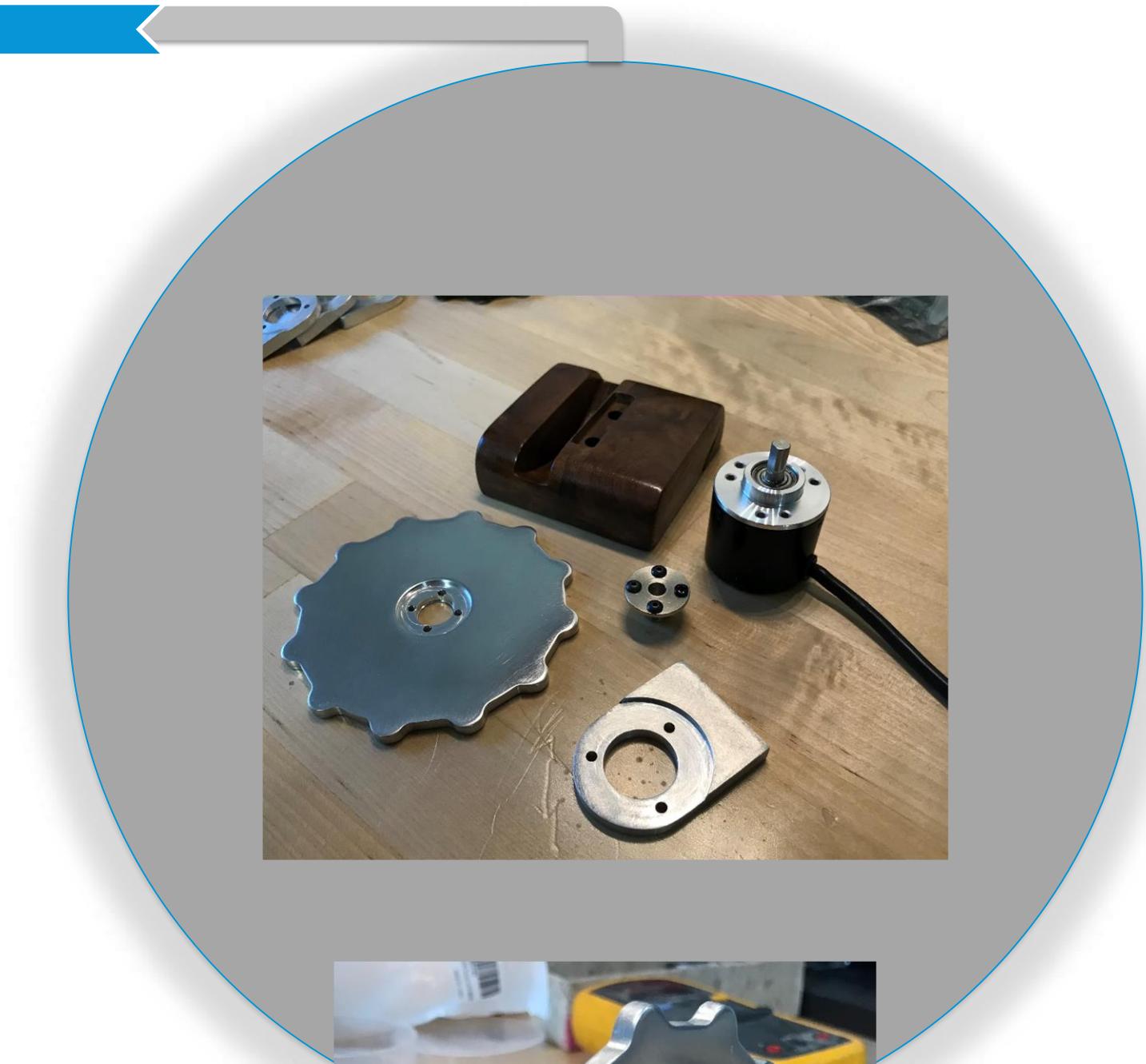
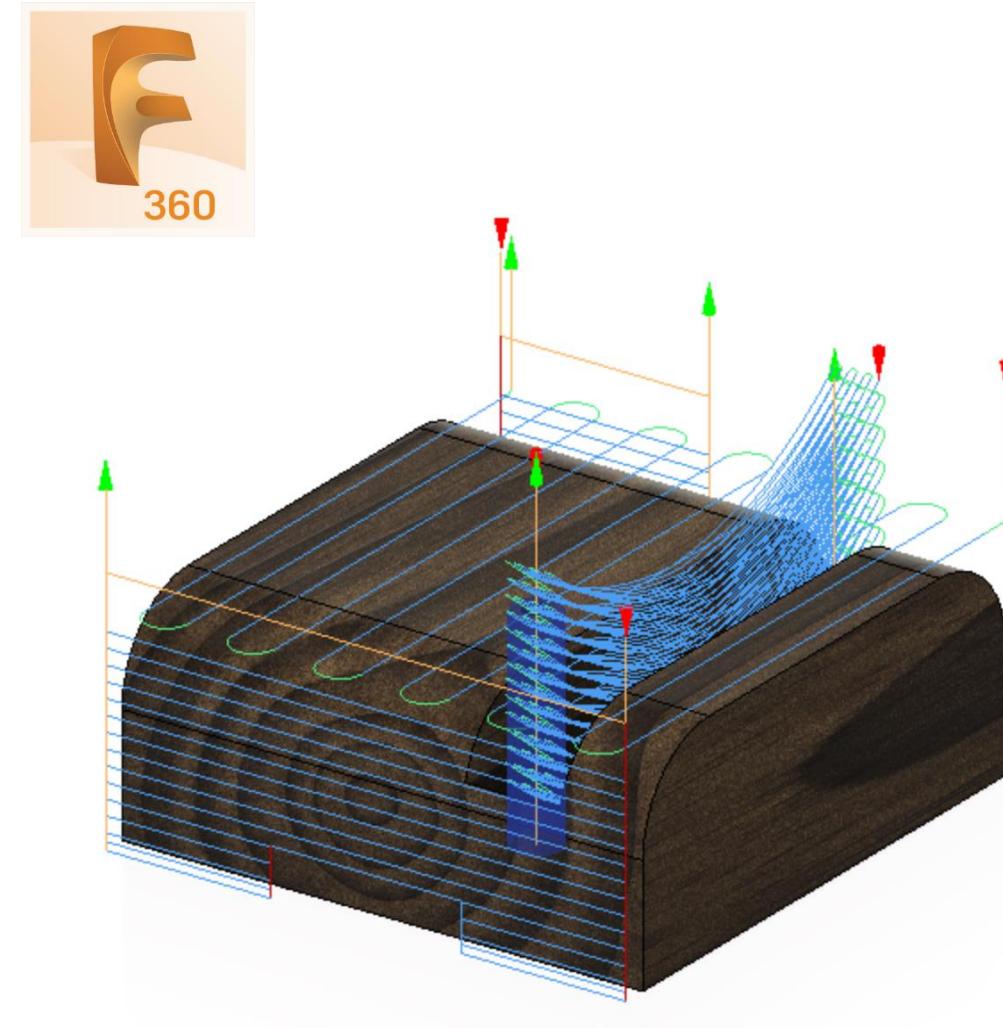
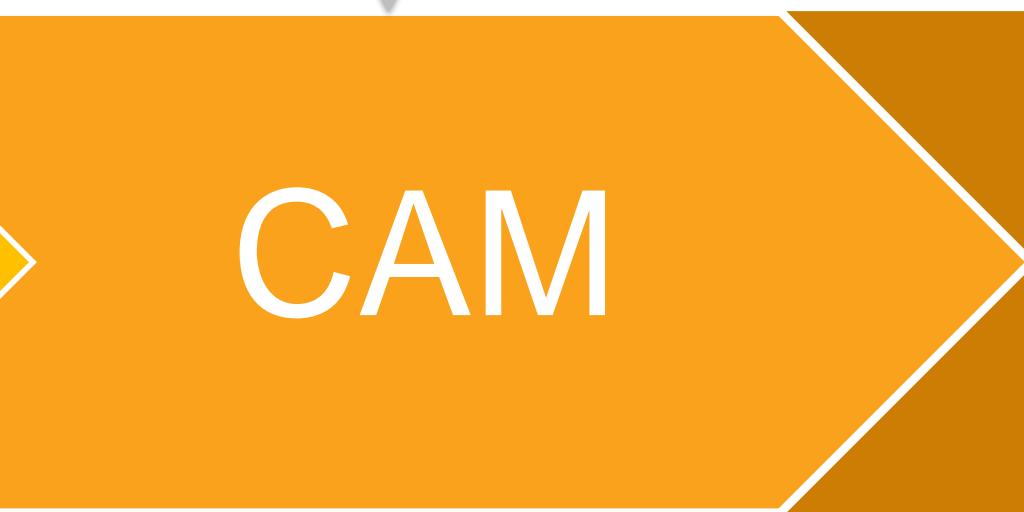
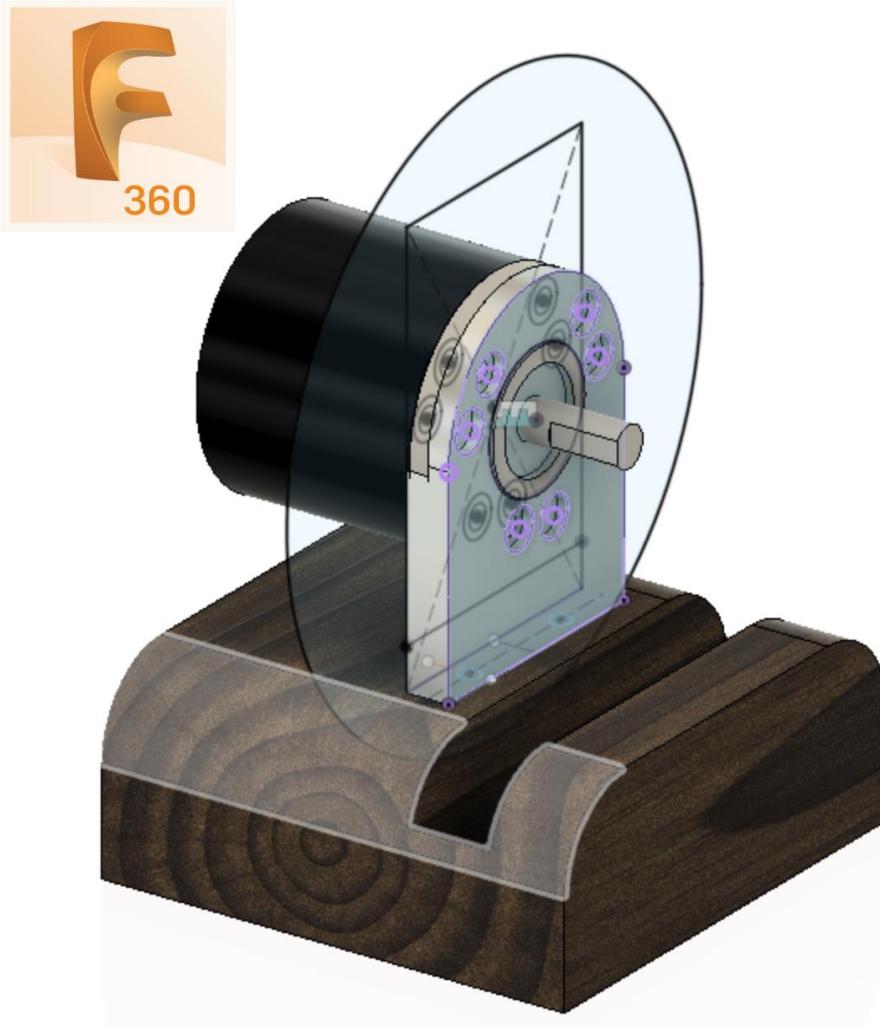
- 表面仕上げの目視検査
- マシーンオペレータの記録
- エンジニアリング機能フィードバック
- 顧客からのフィードバック

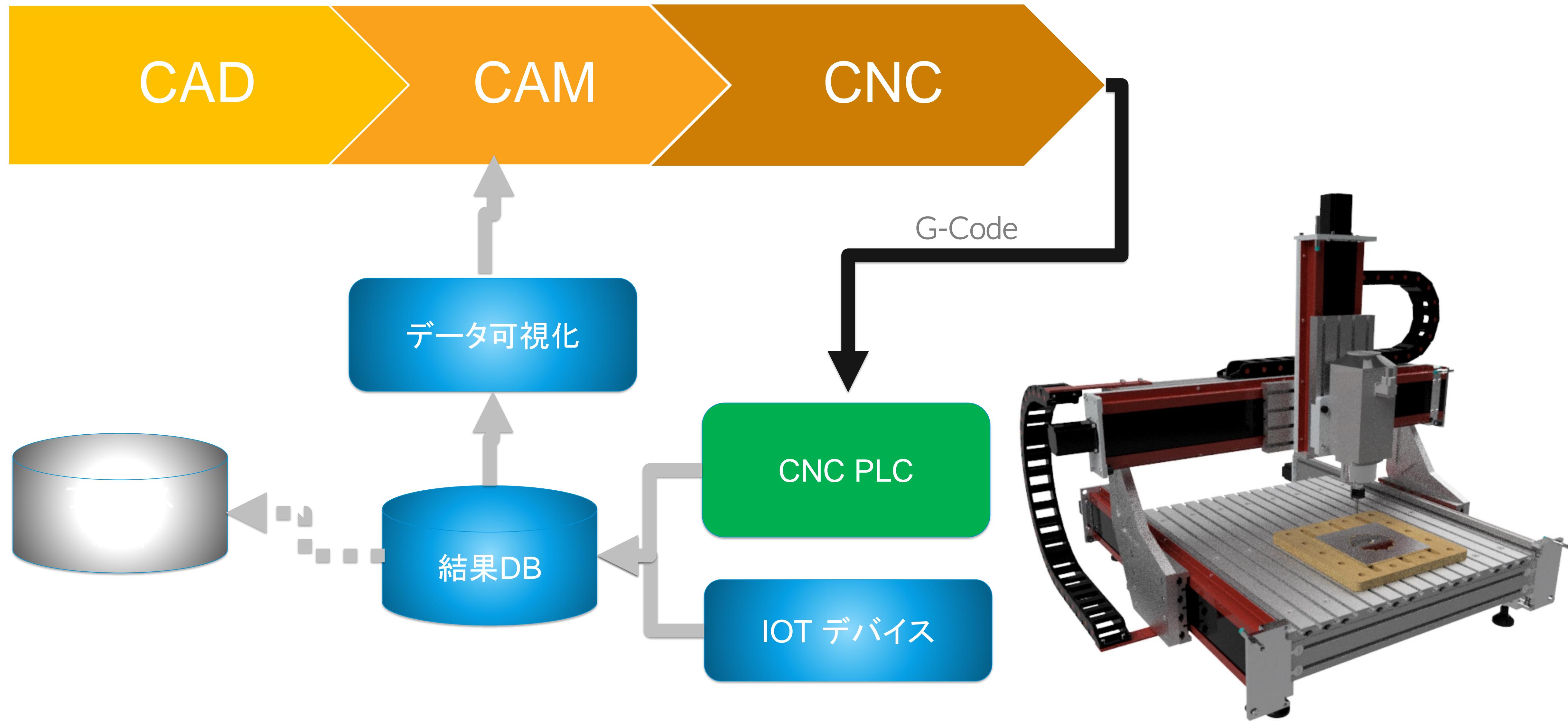
他のアイデアは?





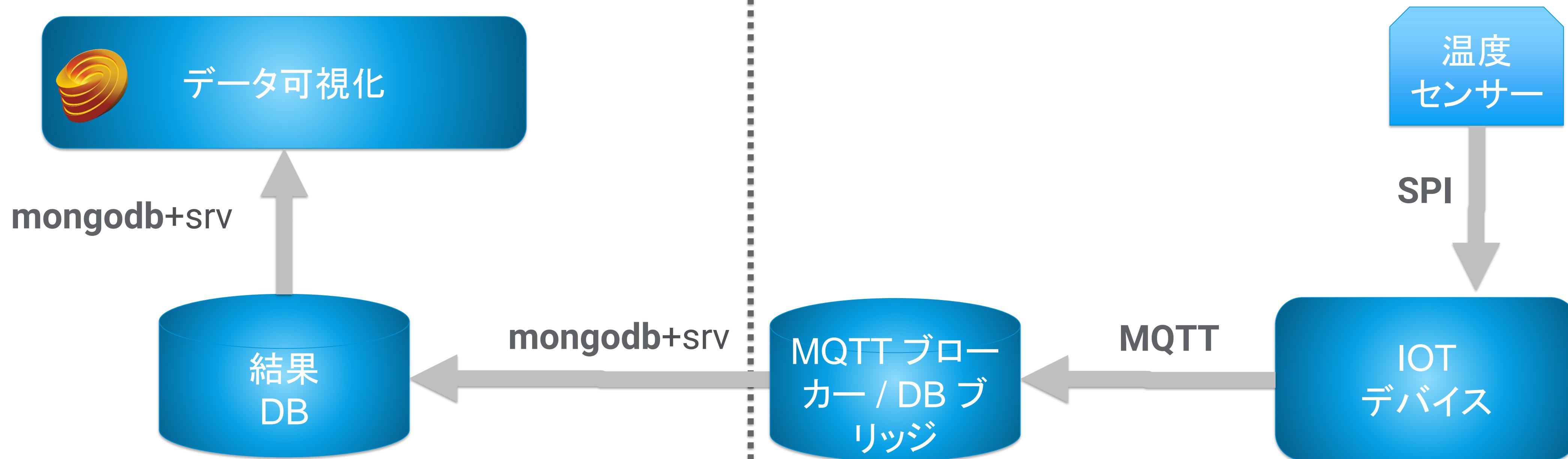






クラウド

エッジ (ish)



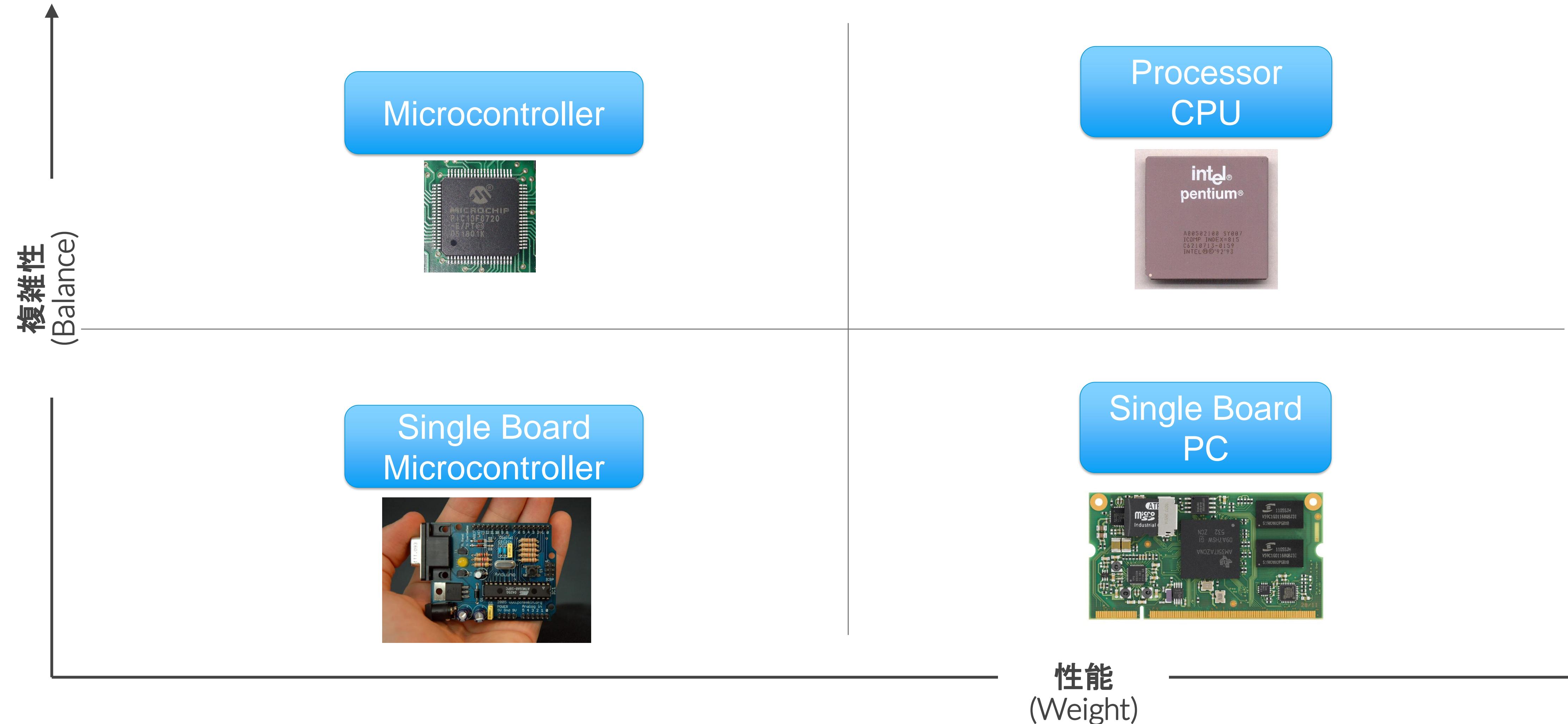
simplify, then add

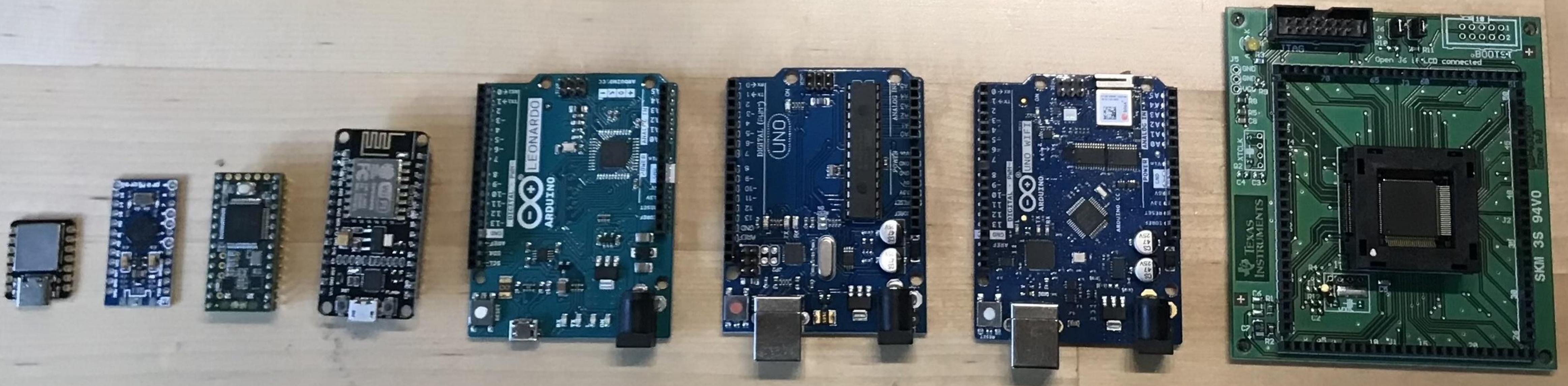
lightness

(加える物は軽さだけ)

Colin Chapman

# コンピューターソリューションマトリクス





NodeMCU ESP12E (WiFi with Limited-IO)

# ソリューションのレシピ

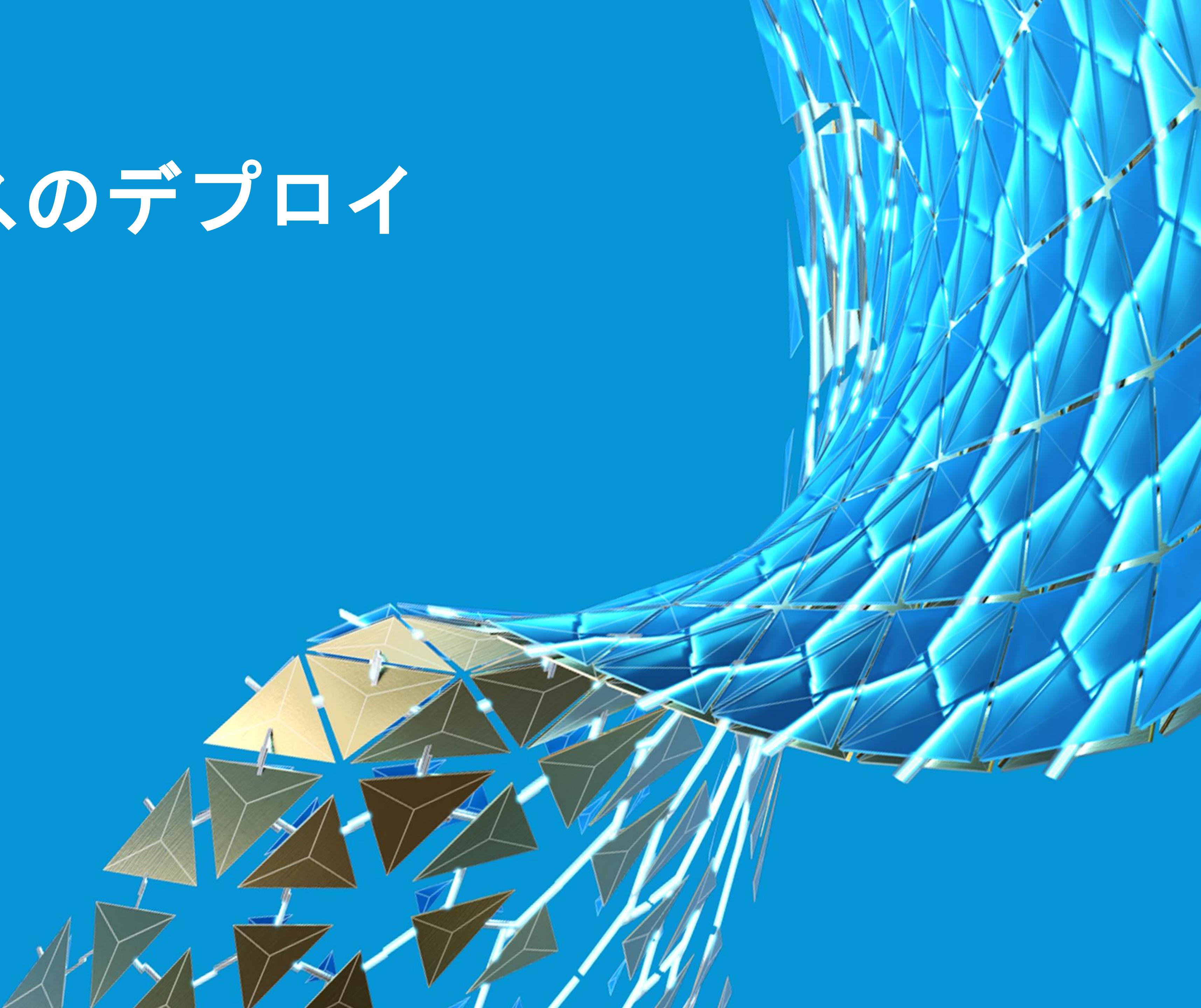
視覚化エンジン	Forge Viewer
データベース	MongoDB
DB プロトコル	MongoDB+SRV
IoT プロトコル	MQTT
IoT デバイス	ESP8266 NodeMCU ESP-12E
温度センサー	DS18B20

# 手順

- データベースのデプロイ
- MQTT ブローカーのビルド
- IoTセンサーのビルド
- IoT ファームウェアのビルド
- MQTT → Databaseのテスト
- Fusion 360モデルのトランスレート
- Forge visualization サービスのビルド
- 結合
- まとめ



# データベースのデプロイ





## Get started free

No credit card required.



Sign up with Google

or

Your Company (optional)

Autodesk

How are you using MongoDB?

Other

Your Work Email

nathan.skalsky@autodesk.com

First Name

Nathan

Last Name

Skalsky

Password

\*\*\*\*

8 characters minimum

I agree to the [terms of service](#) and [privacy policy](#).

Get started free

### Included with your free cloud database:

- ✓ 512 MB of Storage
- ✓ Shared RAM
- ✓ Highly available replica sets, end-to-end encryption, automated patches, REST API

### Additionally, get access to the following when you launch a dedicated cluster:

- ✓ 10 GB or more of storage
- ✓ Dedicated RAM
- ✓ Performance optimization tools
- ✓ Backups & point-in-time recovery
- ✓ Enterprise security features including encryption key management, LDAP integration, and granular database auditing
- ✓ Global Clusters

Autodesk Forge

Access Manager

Support

Billing

AU2020IoTDemo



Atlas



Charts



## DATA STORAGE

Clusters

Triggers

Data Lake

## SECURITY

Database Access

Network Access

Advanced

We are deploying your changes (current action: configuring MongoDB)

AUTODESK FORGE &gt; AU2020IOTDEMO

## Clusters

Create a New Cluster



## SANDBOX

## Cluster0

Version 4.2.10

CONNECT

METRICS

COLLECTIONS

...

## CLUSTER TIER

M0 Sandbox (General)

## REGION

AWS / Oregon (us-west-2)

## TYPE

Replica Set - 3 nodes

## LINKED REALM APP

None Linked

## ATLAS SEARCH

Create Index

Operations R: W:

100.0/s

Last 6 Hours

Connections 0

500 max

Last 6 Hours

Logical Size 0.0 B

512.0 MB max

Last 6 Hours

## Enhance Your Experience

For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

Upgrade

Get Started 2

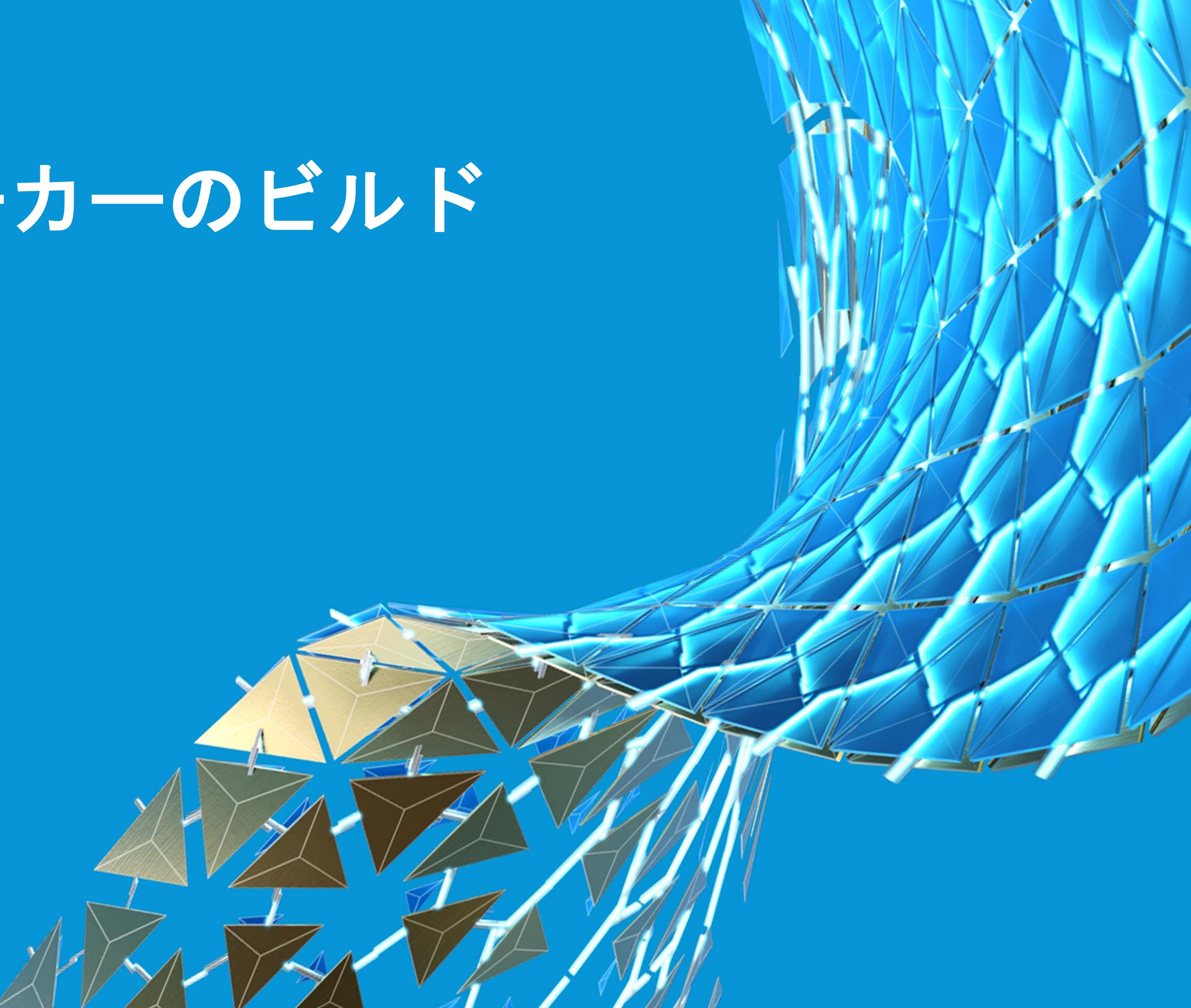
Feature Requests

System Status: All Good

©2020 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales



# MQTT ブローカーのビルド



# シンプルなMQTT ブローカのビルド (+MongoDB ブリッジ)

- 利用ライブラリ: mongodb, mosca (MQTT)
  - 説明: ポート1888で MQTT メッセージを受信し、シンプルなセンサースキーマを用いて結果データベースに挿入

MQTTBroker.js

```

1 //Mongo DB Connection
2 const MongoClient = require('mongodb').MongoClient
3 const dbConnectionString = MongoDB = "mongodb+srv://MQTTBroker:forme2no@cluster0.cnyhr.mongodb.net/
• <dbname>?retryWrites=true&w=majority"
4 const client = new MongoClient(dbConnectionString, {useNewUrlParser: true, useUnifiedTopology: true})
5 client.connect() // create persistent connection
6
7 //MQTT Broker
8 var mosca = require('mosca')
9 var config = {port: 1888}
10 var MQTTBroker = new mosca.Server(config)
11
12 MQTTBroker.on('ready', ()=>{
13   console.log('MQTT Broker Service Started!')
14 })
15
16 MQTTBroker.on('published', (packet)=>{
17   message = packet.payload.toString()
18
19   if(!isNaN(message)){
20     console.log('Publishing Topic: '+packet.topic.toString()+' Value: '+message)
21     var activeCollection = client.db('AU2020IoTDemo').collection('test_ESP12') //TODO: map collection
• name to CNC job name dynamically
22     activeCollection.insertOne({
23       when: new Date(),
24       temperature: message, // Sensor value, in this case (Temp), it will be degrees celsius (float)
25       sensor_type: packet.topic.toString().slice(0,4), // sensor type is only 'Temp' presently
26       sensor_name: packet.topic.toString().slice(5,25) // sensor name can be 'StockMaterial', 'Tool',
• 'Spindle', 'StepperX', etc.
27     })
28   }
29 })

```

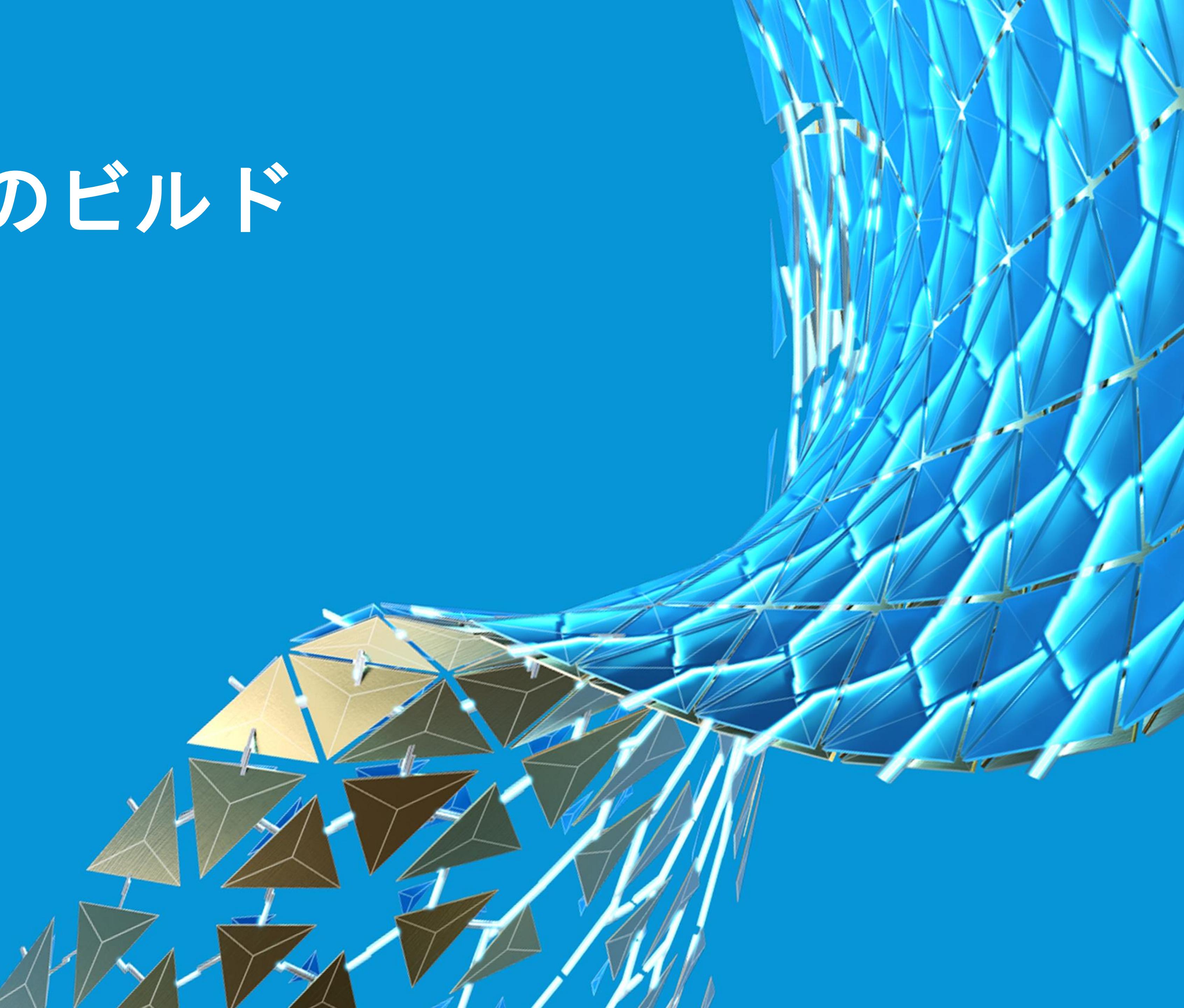
# シンプルな MQTT ブローカーのテスト

- 利用ライブラリ: mongodb, mosca (MQTT)
- 説明:
  - MQTTPublish.js – topic/subtopic 文字列およびセンサーを元に、MQTTメッセージを送信
  - MQTTSsubscribe.js – Will subscribe to receive 特定のtopic/subtopicを用いて、MQTTメッセージを受信するようにサブスクライブ

MQTTBroker.js	MQTTPublish.js	
<pre>1 //MQTT client 2 var mqtt = require('mqtt') 3 var client = mqtt.connect('mqtt://localhost:1888') 4 5 //what topic? 6 var topic = 'Temp/Default' 7 var sensorValue = 0 8 9 client.on('connect', ()=&gt;{ 10   process.argv.forEach(function (val, index, array){ 11     if (index == 2){ 12       topic = val 13     } 14     if (index == 3){ 15       sensorValue = val 16     } 17   }) 18   client.publish(topic, sensorValue.toString()) 19   console.log('Sending Topic: '+topic.toString()+' Value: '+sensorValue.toString()) 20   return process.exit(22) 21 }) 22 }</pre>		

MQTTBroker.js	MQTTPublish.js	MQTTSsubscribe.js
<pre>1 //MQTT client 2 var mqtt = require('mqtt') 3 var client = mqtt.connect('mqtt://localhost:1888') 4 5 //what topic? 6 var topic = process.argv.slice(2); 7 8 client.on('message', (topic, message)=&gt;{ 9   message = message.toString() 10  console.log(message) 11 }) 12 13 client.on('connect', ()=&gt;{ 14   client.subscribe(topic) 15 }) 16</pre>		

# IoTデバイスのビルト

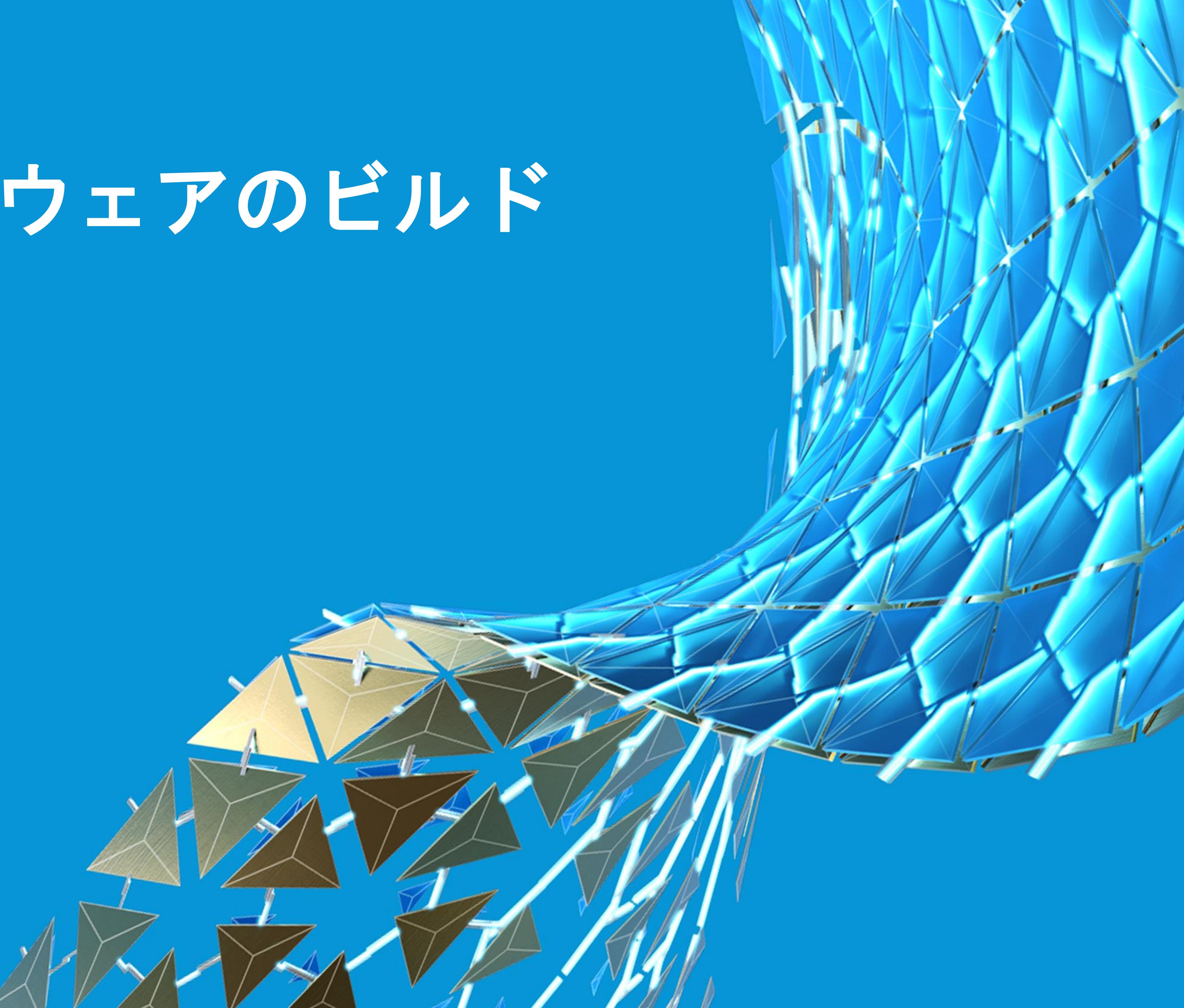


# IoTデバイスのビルト

- 詳細はクラス動画とハンドアウトを参照
- **BOM:**
  - ESP8266 NodeMCU ESP-12E
  - DS18B20 デジタル温度計
  - 4.7k 抵抗
- 接続:
  - Temp Sensor.VCC = Vin on ESP board
  - Temp Sensor.GND = GND on ESP Board
  - Temp Sensor.Data = GPIO 5
  - 4.7k 抵抗は GPIO 5 および Vinに接続



# IoT ファームウェアのビルド



# IoT ファームウェアのビルド

- 詳細はクラス動画とハンドアウトを参照
- Nathanの4つの重要なIoTデバイスの機能
  1. アクセスポイントに接続するWiFiクライアント
  2. ユーザからのWiFi認証を受け付けるWiFiアクセスポイントとしての能力
  3. ユーザにコントローラのリセットを行うことを許可する機能(2重リセット)
  4. MQTT クライアントおよびMQTTイベントへのサブスクライブ

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** CNC-IoT-TempSensor-ESP12 | Arduino
- Sketch Name:** CNC-IoT-TempSensor-ESP12 §
- Code Preview:** The code block contains functions for checking temperature sensors, updating MQTT data, handling MQTT callbacks, and connecting to an MQTT client.

```
void checkTempSensors(){
    // get the temps
    // update MQTT data with temp info + sensor type and name (MQTT topic/subtopic)
}

void updateMQTTData(float temp){
    char mqtt_payload[100] = "";
    sprintf(mqtt_payload, 100, "%2f", temp);
    Serial.print("Publish Message: ");
    Serial.println(mqtt_payload);
    client.publish(MQTT_Topic, mqtt_payload, true);
    Serial.println("> MQTT data pub");
}

void mqttCallback(char* topic, byte* payload, unsigned int length){
    Serial.print("MQTT Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i=0; i < length; i++){
        Serial.print((char)payload[i]);
    }
    Serial.println();
}

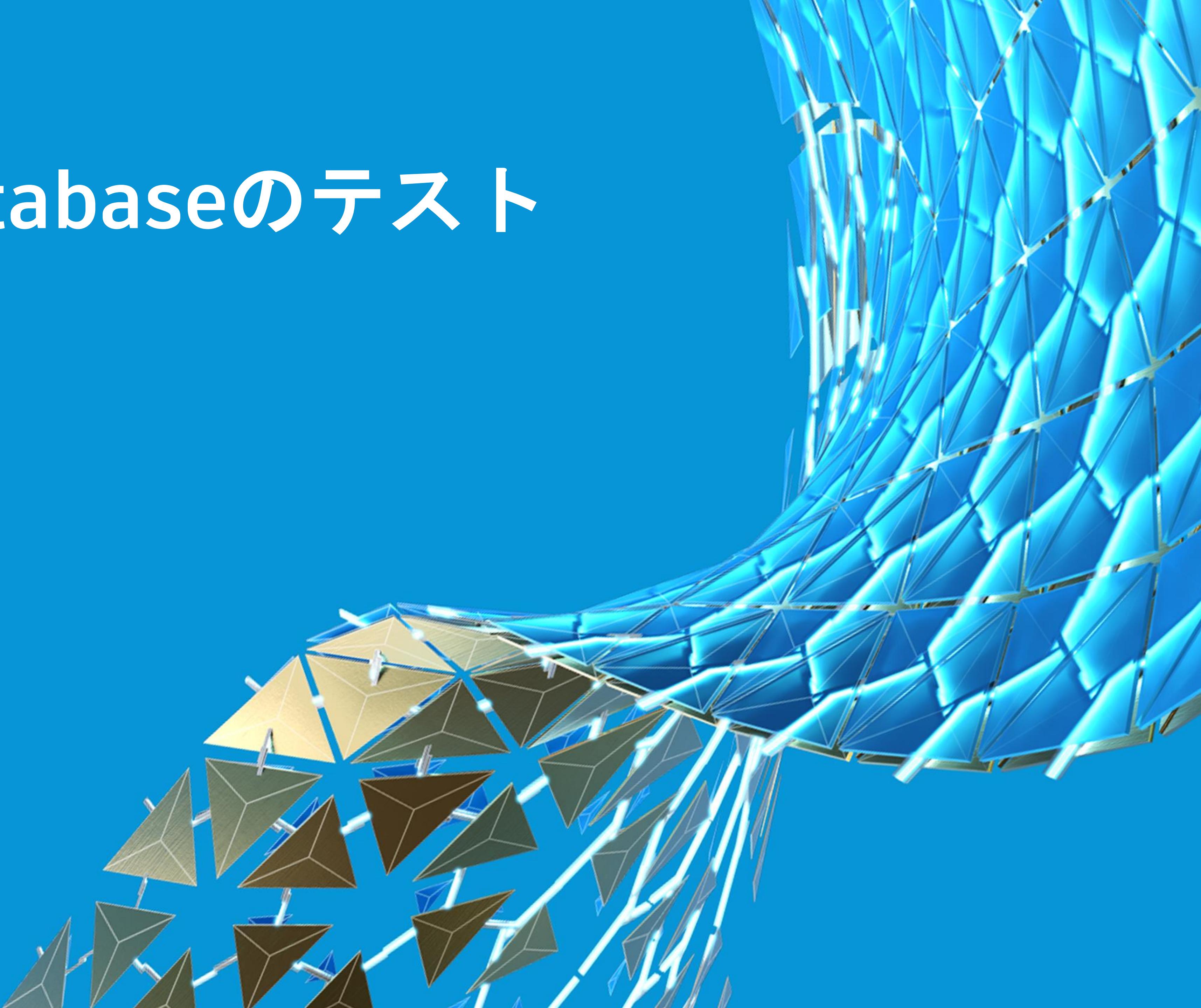
void connectMQTTClient(){
    while (!client.connected()){ //loop until connected
        Serial.print("***** Re-Attempting MQTT connection...");
        if (client.connect(UUIDClientName, mqtt_username, "") ){
            Serial.println(" -> MQTT client connected");
        } else {
            Serial.print("failed, rc=");
        }
    }
}
```

- Build Status:** Done compiling.
- Memory Usage:** Executable segment sizes and sketch memory usage.

```
Executable segment sizes:
IROM   : 281288  - code in flash      (default or ICACHE_FLASH_ATTR)
IRAM   : 27312  / 32768 - code in IRAM  (ICACHE_RAM_ATTR, ISRs...)
DATA   : 1276 )   - initialized variables (global, static) in RAM/HEAP
RODATA : 1544 ) / 81920 - constants     (global, static) in RAM/HEAP
BSS    : 29584 )  - zeroed variables   (global, static) in RAM/HEAP
Sketch uses 311420 bytes (29%) of program storage space. Maximum is 1044464 bytes.
```

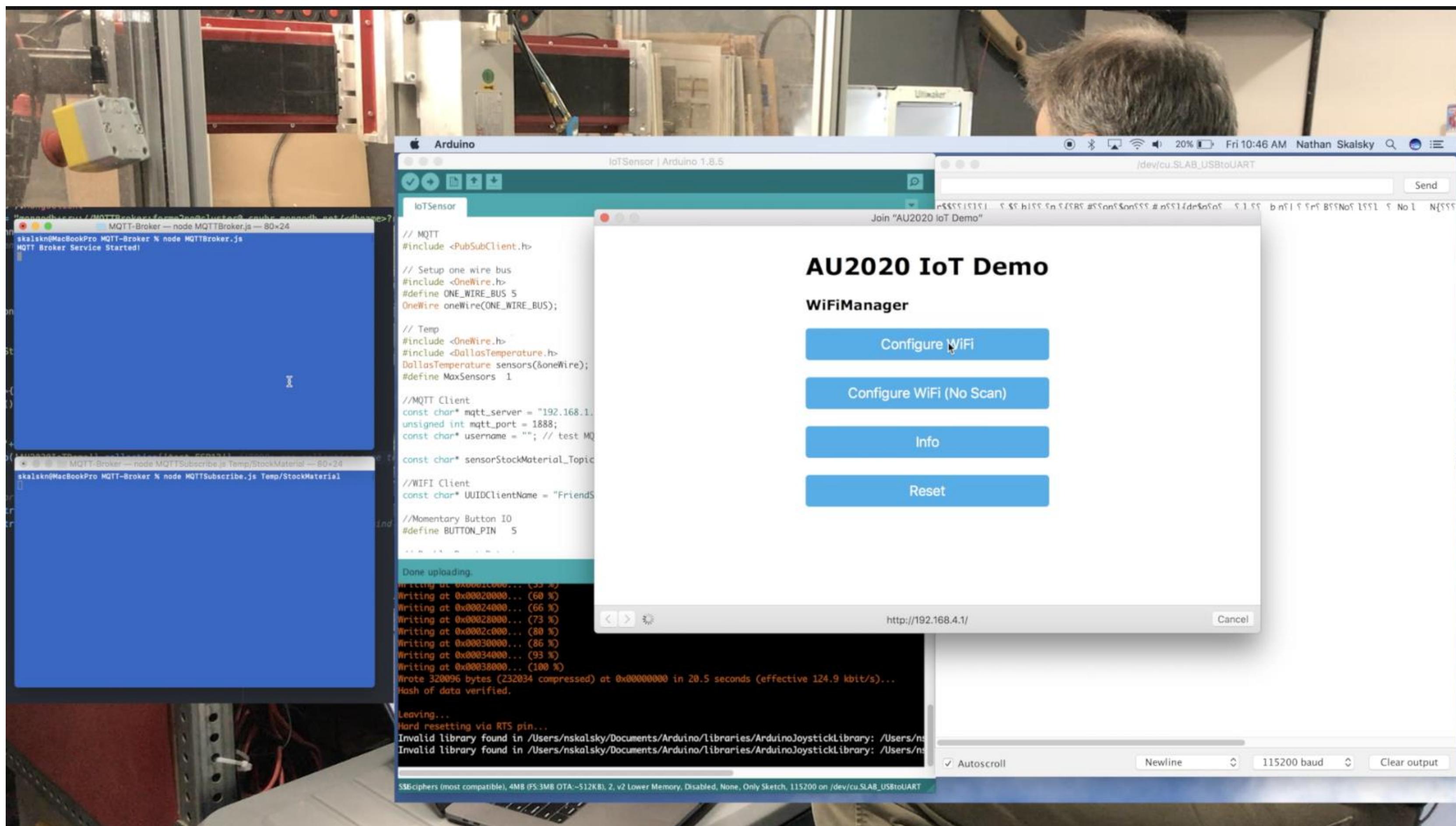
- Page Footer:** 92 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return nullptr), All SSL ciphers (most compatible), 4MB (FS:3)

# MQTT → Databaseのテスト

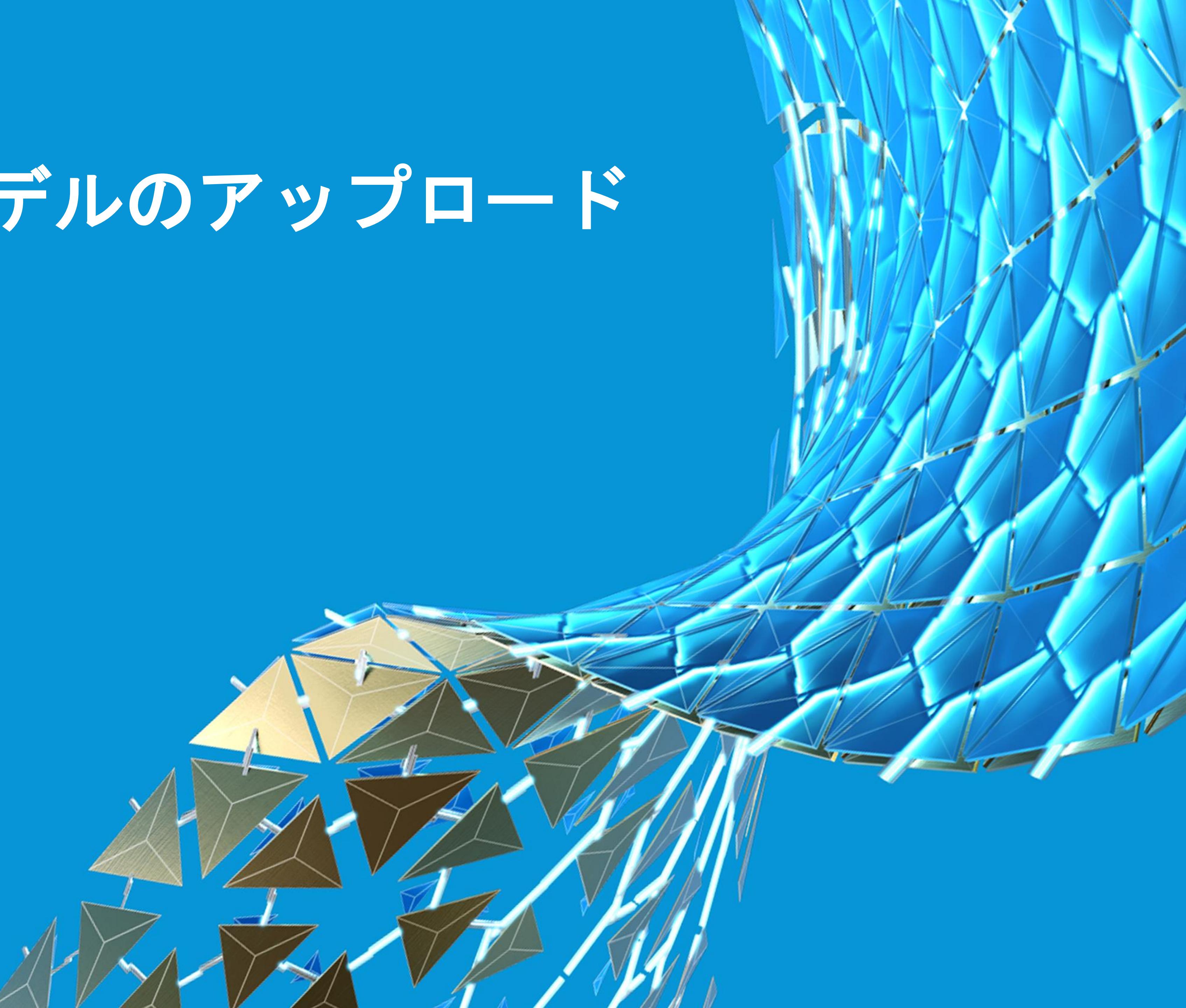


# MQTTからdatabaseのユニットテスト

- 詳細はクラス動画とハンドアウトを参照

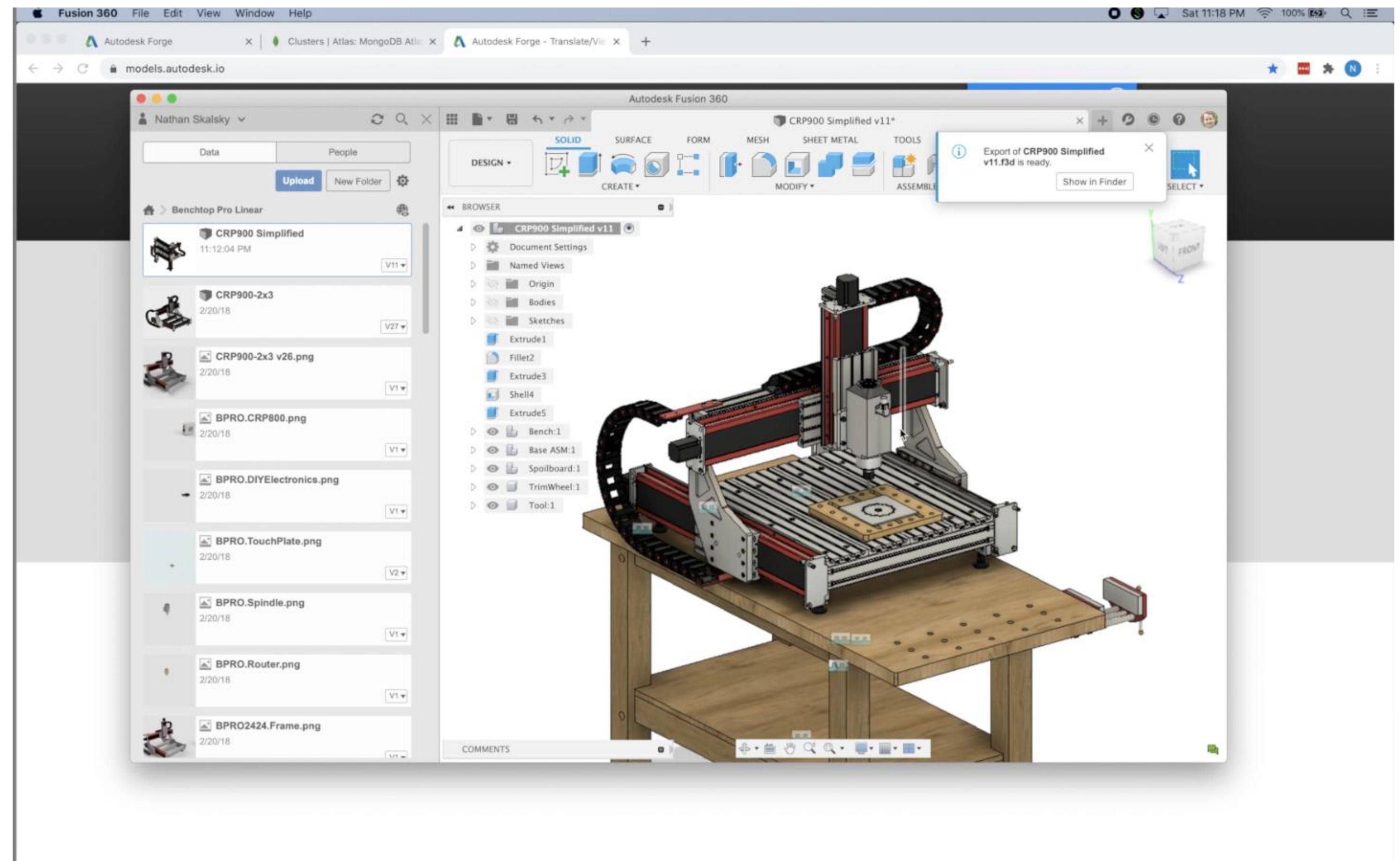


# Fusion 360モデルのアップロード

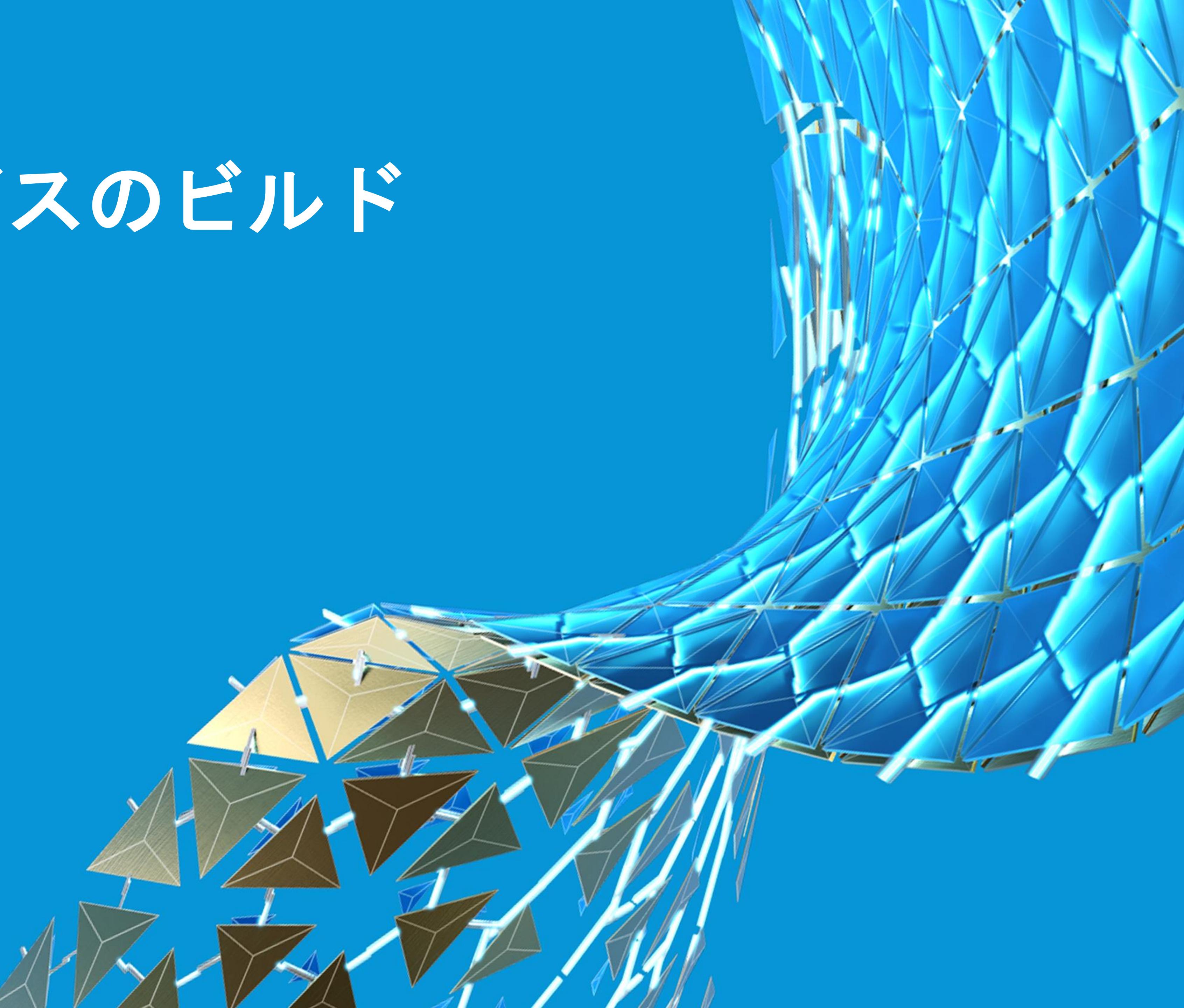


# Upload Fusion 360 model to Forge

- 詳細はクラス動画とハンドアウトを参照
- Forge API認証を用いてモデルをmodels.autodesk.ioにアップロード

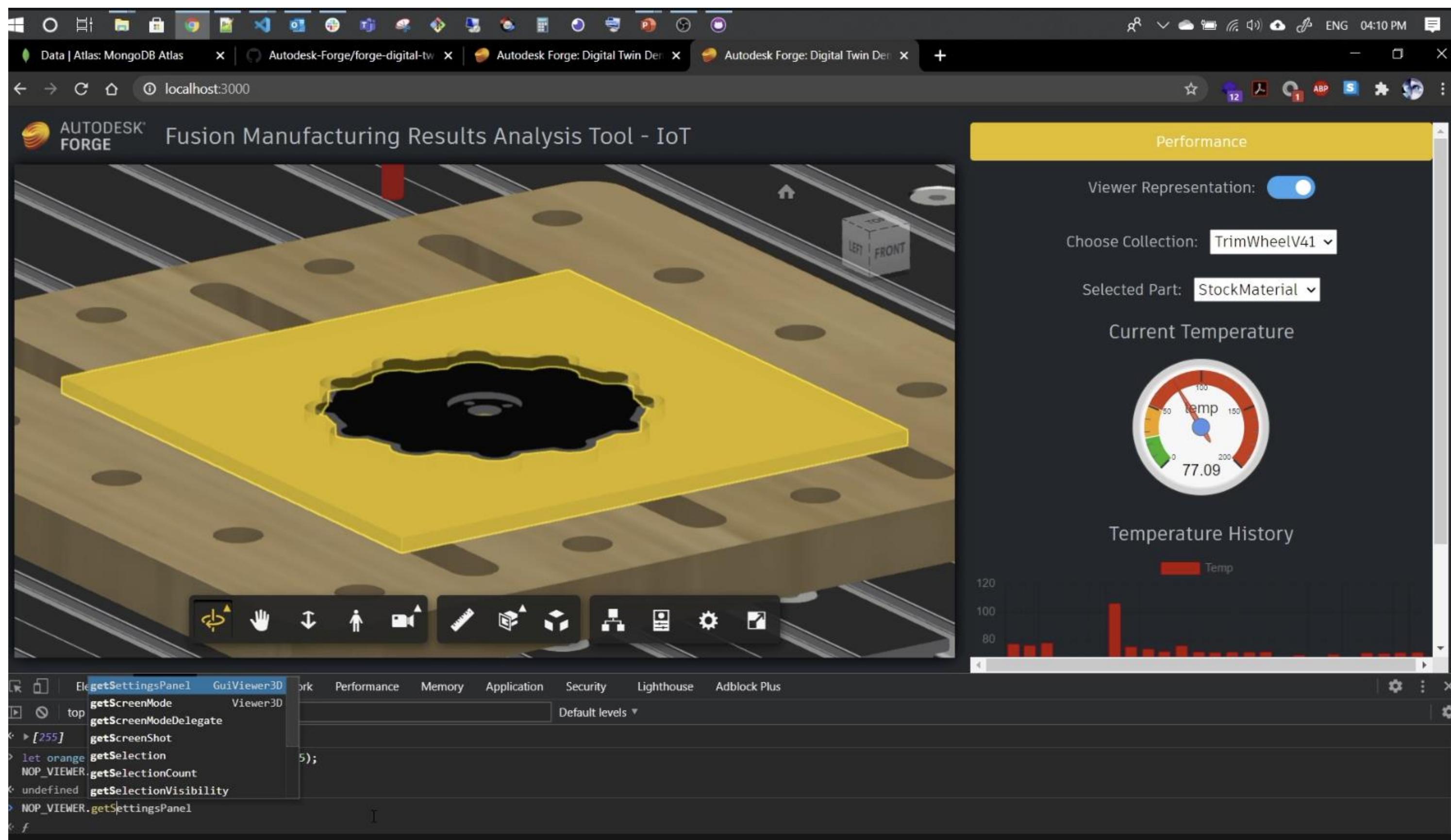


# 可視化サービスのビルド

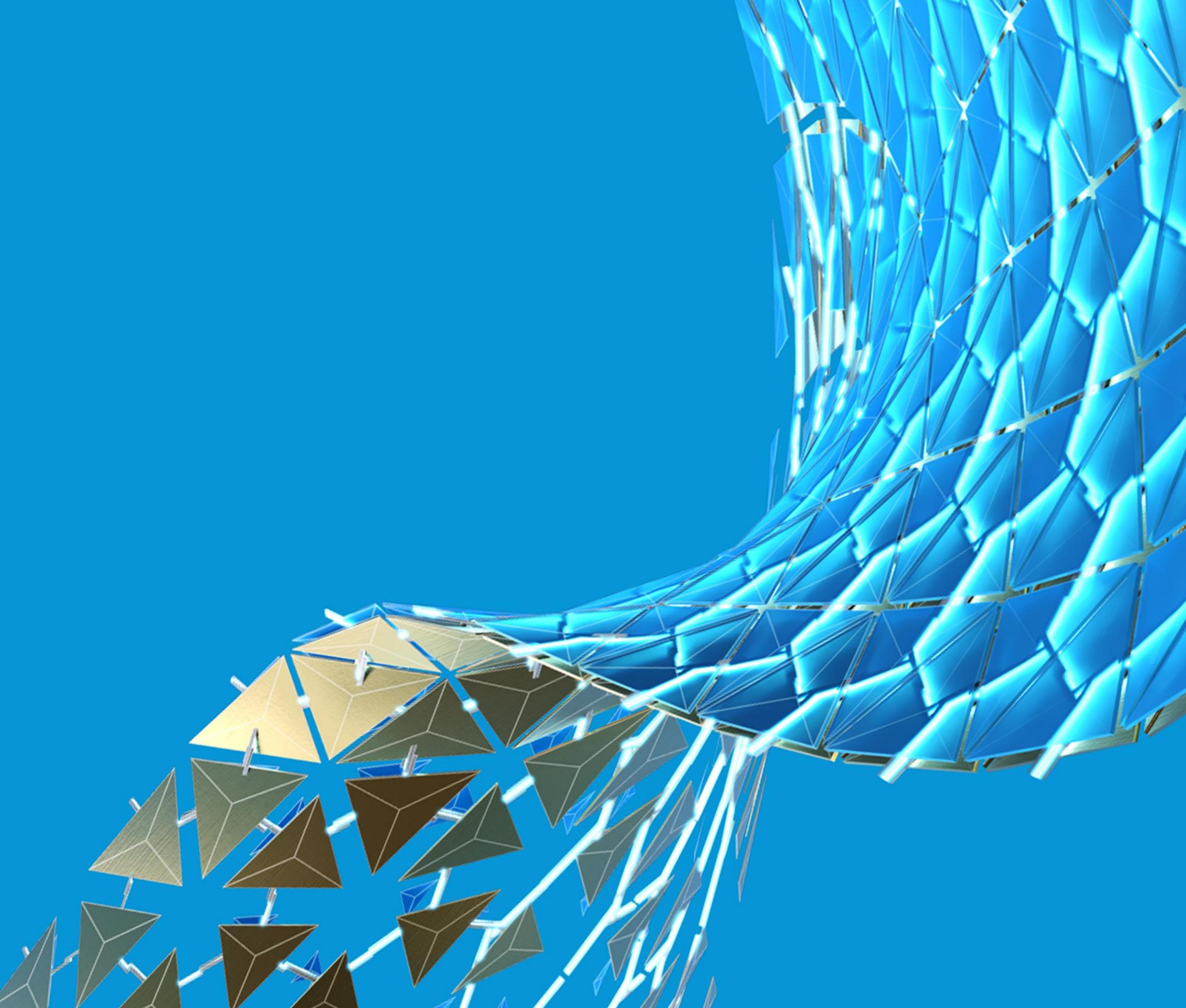


# 可視化Webエクスペリエンスのビルト

- 詳細はクラス動画とハンドアウトを参照

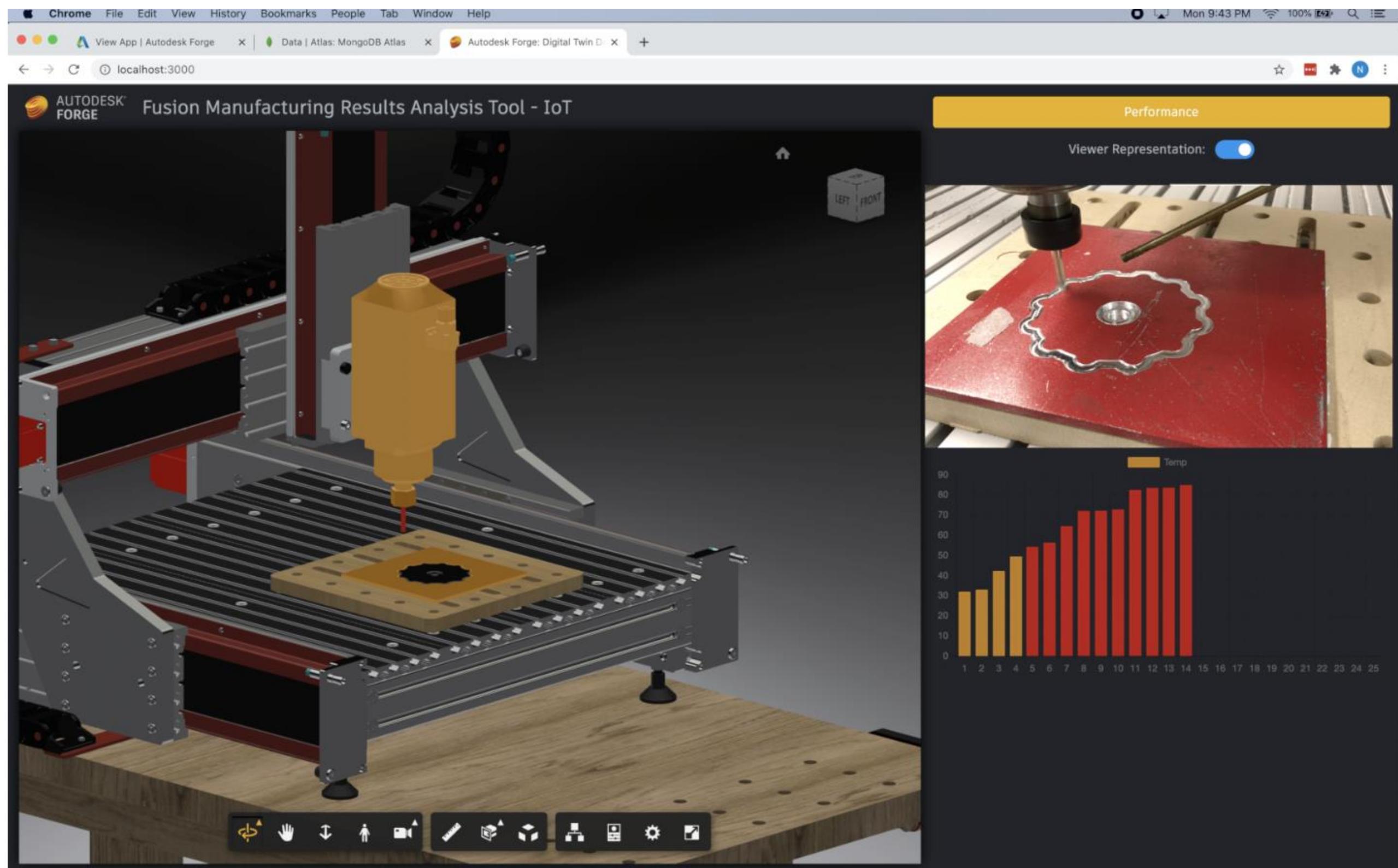


結合

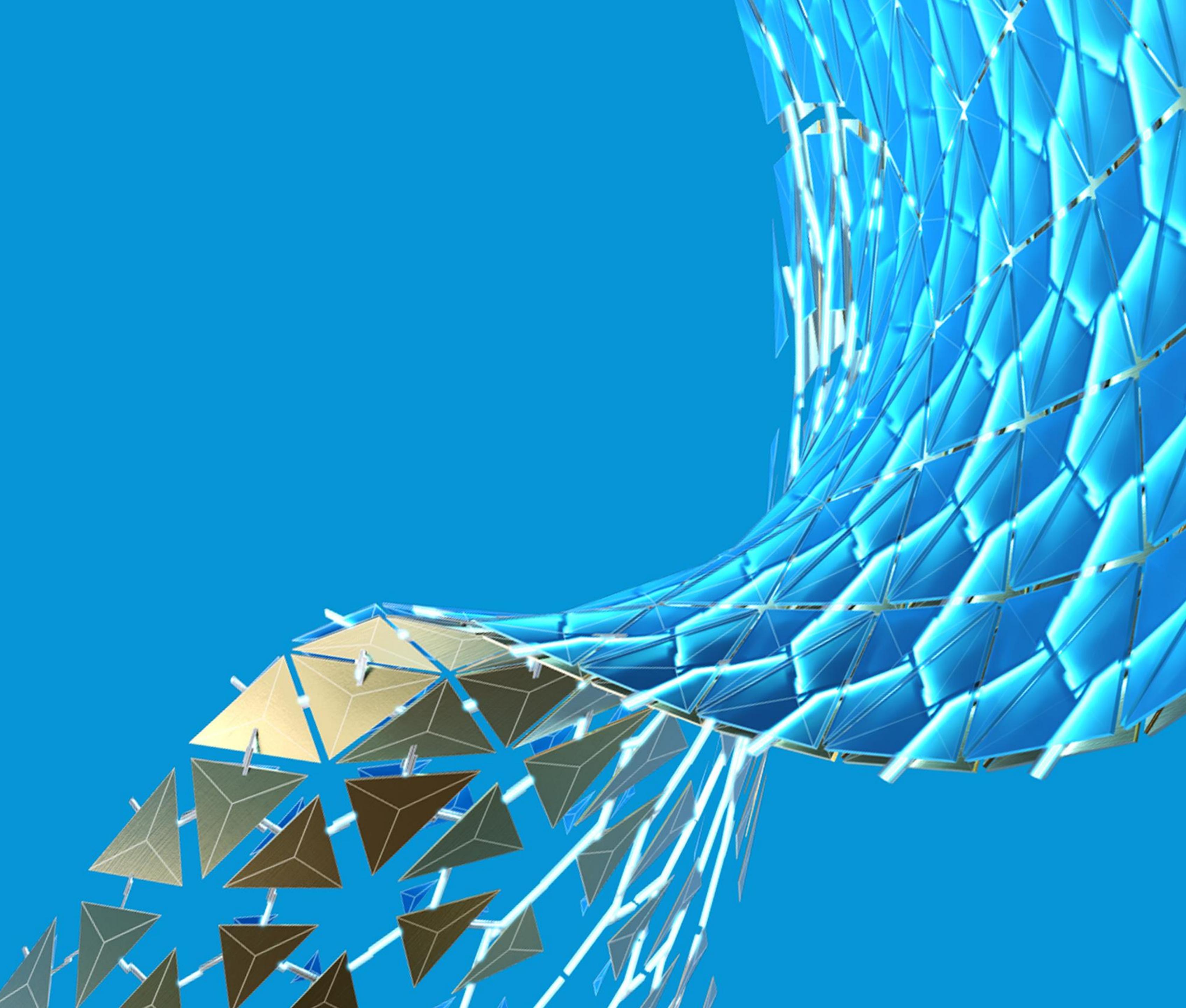


# 結合

- 詳細はクラス動画とハンドアウトを参照



# まとめ



# まとめ

- データベースのデプロイ
- MQTT ブローカーのビルド
- IoTセンサーのビルド
- IoT ファームウェアのビルド
- MQTT → Databaseのテスト
- Fusion 360モデルのトランスレート
- Forge visualization サービスのビルド
- 結合
- まとめ



# ソリューションのレシピ

視覚化エンジン	Forge Viewer
データベース	MongoDB
DB プロトコル	MongoDB+SRV
IoT プロトコル	MQTT
IoT デバイス	ESP8266 NodeMCU ESP-12E
温度センサー	DS18B20

# 外部リソース・リファレンス

- サンプルプロジェクトのダウンロード > <https://github.com/Autodesk-Forge/forge-iot-cnc>
- Forge API ドキュメント > <https://forge.autodesk.com/developer/documentation>
- Forge Viewer > <https://forge.autodesk.com/en/docs/viewer/v7/overview/>
- MongoDB Atlas > <https://docs.atlas.mongodb.com/>
- NodeMCU ESP8266 > <https://www.instructables.com/NodeMCU-ESP8266-Details-and-Pinout/>
- MQTT > <https://mqtt.org/mqtt-specification/>
- デジタル温度計 (DS18B20) > <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>



Autodesk and the Autodesk logo are registered trademarks or trademarks 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 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.

© 2020 Autodesk. All rights reserved.

