

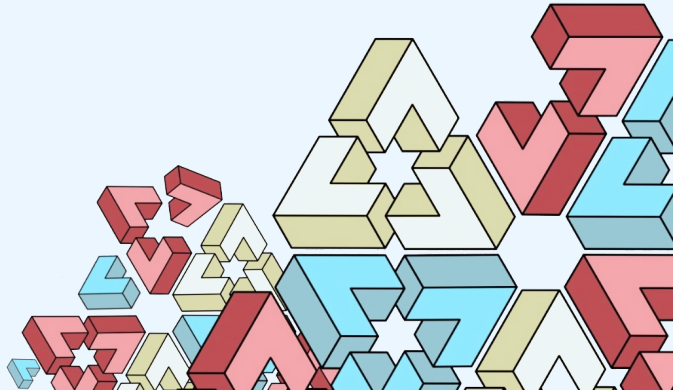
★2本立て★ A-Frameを活用したxR入門 & IoT活用事例紹介

株式会社ブリッジ 継枝 由希子
荻原 宙佳 ジェン

 **BRIDGE** - Since 1974 -

『想像を超えた創造を』

株式会社ブリッジ



自己紹介

株式会社ブリッジ

継枝 由希子

- 普段のお仕事: パッケージシステムの開発
- Joined in April 2017

株式会社ブリッジ

荻原 宙佳 ジェン

- 普段のお仕事: サーバーの保守運用
- Joined in April 2018



株式会社 ブリッジ

所在地：札幌市 中央区 北1条 西7丁目 1番地
CARP札幌ビル 4F

TEL: 011-219-8250

URL: <https://www.bridge.co.jp/>

事業内容：システム受託開発
ネットワーク・サーバ技術サービス
デジタル文字フォント販売



ここです！



セミナーの内容

1. A-Frameを活用したxR入門
2. IoT活用事例紹介

A-Frameを活用したxR入門

株式会社ブリッジ 継枝 由希子

目次

1. xR とは
2. A-Frame とは
3. A-Frame で「Hello WebVR ! 」
4. まとめ

VR(仮想現実)や AR(拡張現実),MR(複合現実)の総称

1. xRとは

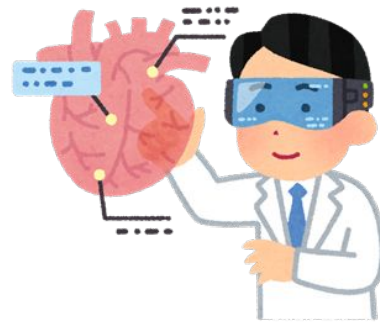
VR(仮想現実)

ヘッドマウントディスプレイなどを利用して仮想空間を表示、
ユーザが仮想空間内に入り込む体験ができる技術



AR(拡張現実)

スマホなどの画面上で現実空間に仮想情報を表示する技術
仮想情報はスマホなどの端末から操作



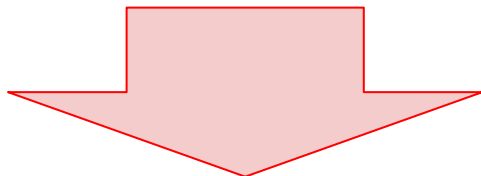
MR(複合現実)

専用のゴーグル等を通して、現実空間に仮想情報を重ねて表示する技術
仮想情報を直接操作(手で触れたり、掴んだり)

※ARと区別されず、ARと呼ばれることも多い

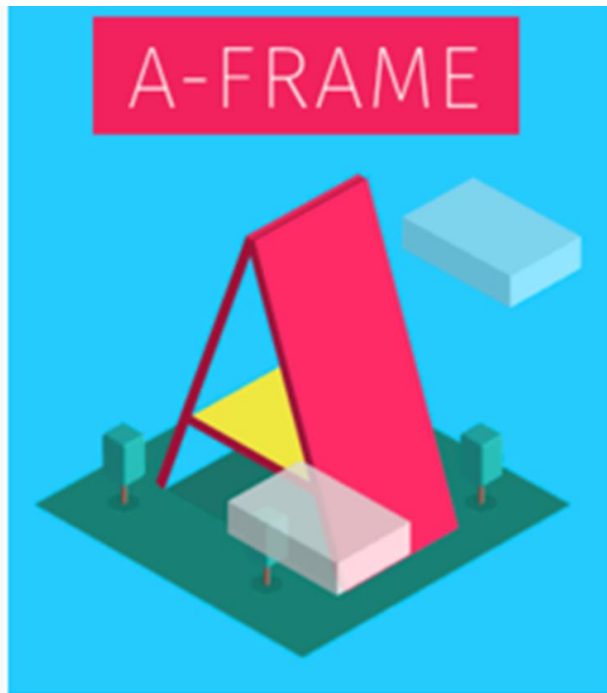
xRに興味あるけれど・・・

- 機材は何を用意すればいいのだろうか？
- 一式そろえるのに費用が掛かりそう・・・
- 開発ツールは色々メニューがあって難しそうだなあ



入門編として、**お手軽にXRを体験**できるものをご紹介します！

2. A-Frameとは



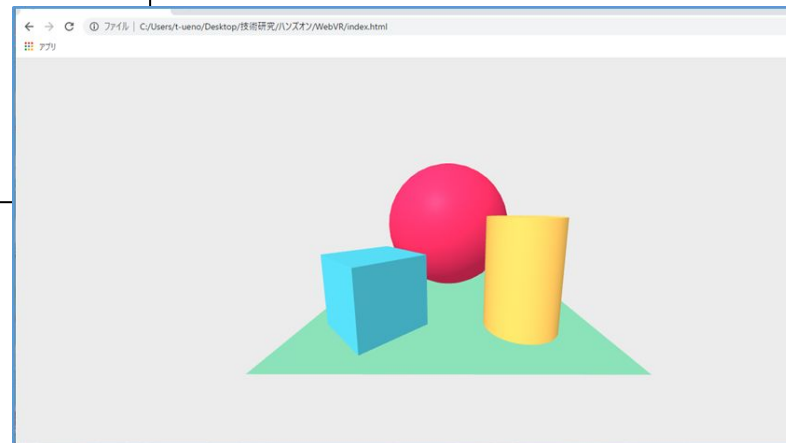
公式サイト: <https://aframe.io/>

- MozillaのVRチームで開発されたオープンソースのWebフレームワーク
- three.js をベースに開発されている
- 簡単なVRシーンであれば、**HTMLタグだけで**プリミティブオブジェクト(球や立方体など)を配置したりアニメーションをつけたりできる

2. A-Frameとは

生成イメージ

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
  </head>
  <body>
    <a-scene>
      <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
      <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>
      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
      <a-sky color="#ECECEC"></a-sky>
    </a-scene>
  </body>
</html>
```



3. A-Frameで Hello WebVR



BRIDGE - Since 1974 -

ステップ1:HTMLファイルを作成する

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
```

```
</head>
```

```
<body>
```

```
<a-scene>
```

```
<a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
```

```
<a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
```

```
<a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>
```

```
<a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
```

```
<a-sky color="#ECECEC"></a-sky>
```

```
</a-scene>
```

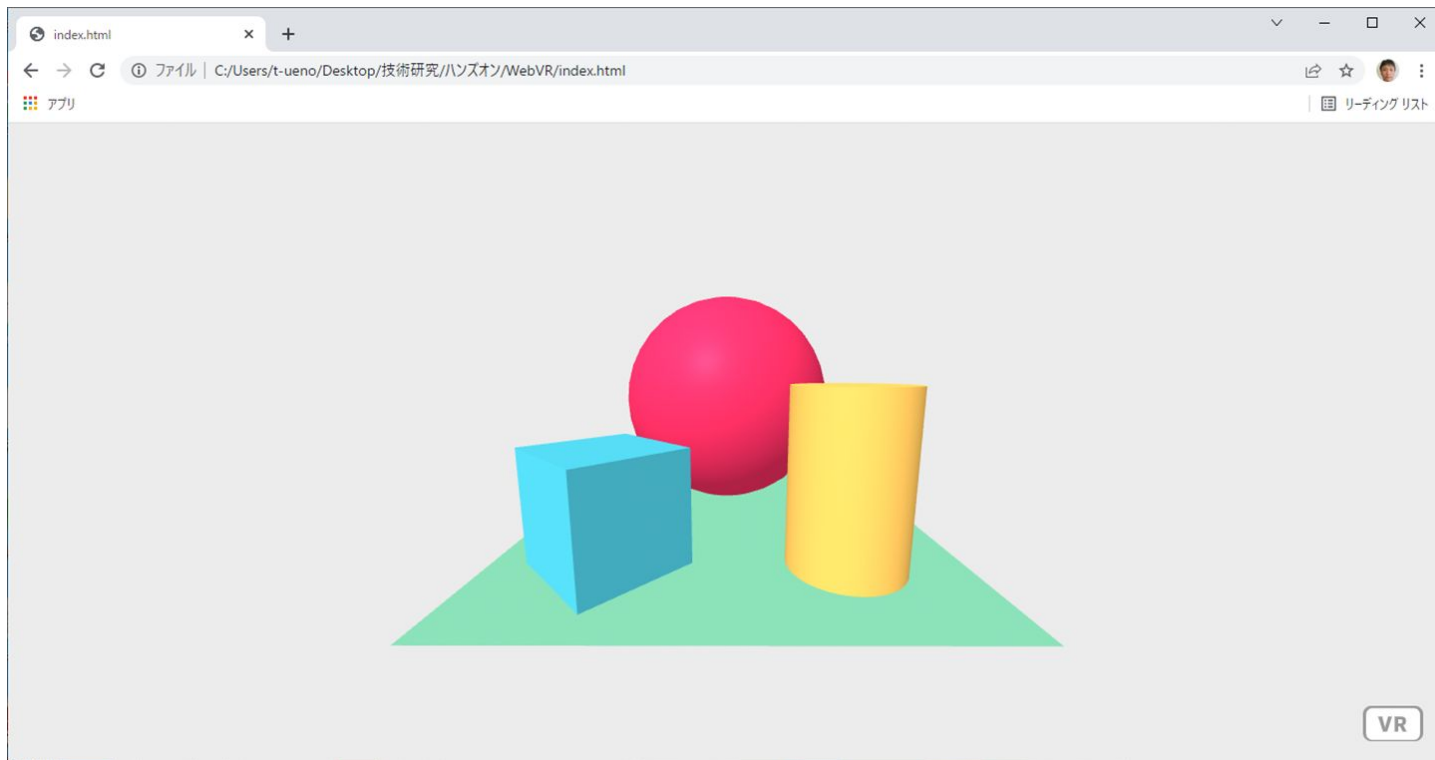
```
</body>
```

```
</html>
```

A-FRAME本体

3. A-Frameで Hello WebVR

ステップ2:ブラウザでファイルを開く



3. A-Frameで Hello WebVR

代表的なタグ

<a-scene>	VRシーンを作成するタグ
<a-box>	立方体
<a-sphere>	球体
<a-cylinder>	円柱
<a-panel>	パネル
<a-sky>	背景

3. A-Frameで Hello WebVR



BRIDGE - Since 1974 -

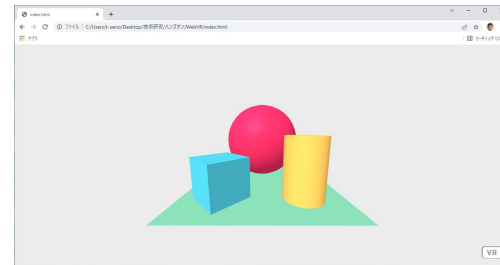
```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
  </head>
  <body>
    <a-scene>
      <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
      <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>
      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
      <a-sky color="#ECECEC"></a-sky>
    </a-scene>
  </body>
</html>
```

3. A-Frameで Hello WebVR

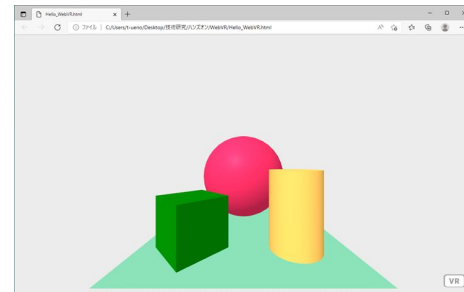
オブジェクトの色を変えてみよう



```
<a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
```



```
<a-box position="-1 0.5 -3" rotation="0 45 0" color="green"></a-box>
```

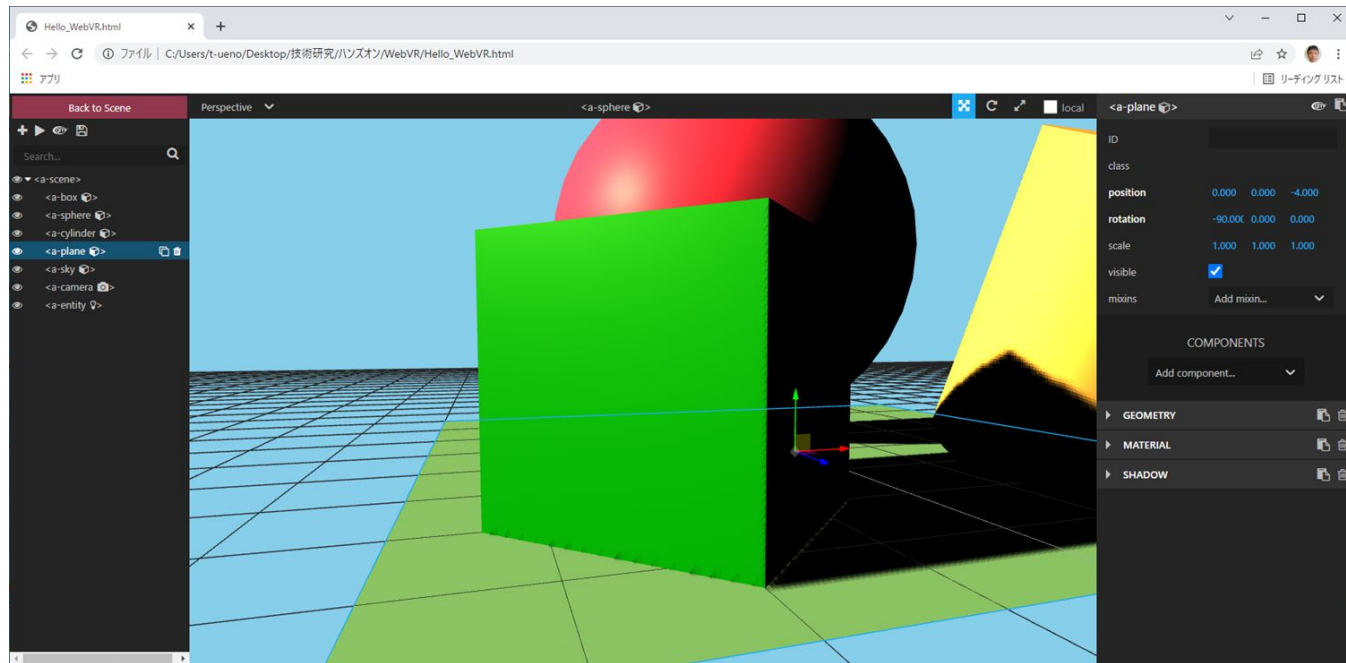


3. A-Frameで Hello WebVR

「インスペクター機能」を利用すると、GUIでVRシーンの編集も可能！



使い方：ブラウザでVRシーンを開き「Ctrl + Alt + i」



3. A-Frameで Hello WebVR



BRIDGE - Since 1974 -

アニメーションでオブジェクトのサイズを変えてみよう



```
<a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
```



```
<a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"
```

```
  animation="property: scale; ← アニメーションの種類:スケール
```

```
    from: 1 1 1; ← 初期角度
```

```
    to: 1.5 1.5 1.5; ← 最終角度
```

```
    loop: true; ← ループの有無:trueで無限、数値で回数
```

```
    dur: 2000; ← 1再生の時間(ミリ秒)
```

```
    dir: alternate" ← アニメーションの方向:
```

- normal ... to到達でfromに戻る

- alternate ... from → to → from → to ...

```
></a-sphere>
```

3. A-Frameで Hello WebVR

マウスクリックでアニメーションを開始してみよう

① カーソルを作成する

```
<a-camera position="0 2 0" >
```

```
  <a-cursor scale="5 5 5"
```

```
    color="blue"
```

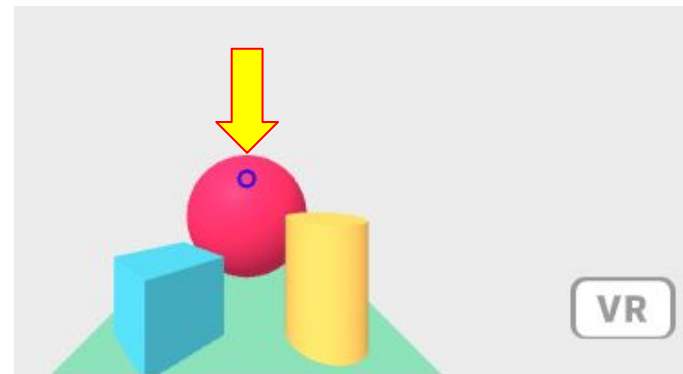
```
    position="0 0 -2"
```

```
    geometry="primitive: ring">
```

```
  </a-cursor>
```

```
</a-camera>
```

※カーソルは、基本カメラの内部に作成する





マウスクリックでアニメーションを開始してみよう



② オブジェクトのアニメーションに開始イベントクリックを追加

```
<a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"  
  animation="property: scale;  
    from: 1 1 1;  
    to: 1.5 1.5 1.5;  
    loop: true;  
    dur: 2000;  
    dir: alternate;  
    startEvents:click;" ← 開始イベント:クリック  
></a-sphere>
```

3. A-Frameで Hello WebVR



BRIDGE - Since 1974 -

マウスクリックで「再生→一時停止→再開」を制御してみよう

```
<a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"
  animation="property: scale;
    from: 1 1 1;
    to: 1.5 1.5 1.5;
    loop: true;
    dur: 2000;dir: alternate;
    startEvents: click;
    pauseEvents: click;
    resumeEvents: click;"
></a-sphere>
```

← 開始イベント: クリック

← 一時停止イベント: クリック

← 再開イベント: クリック

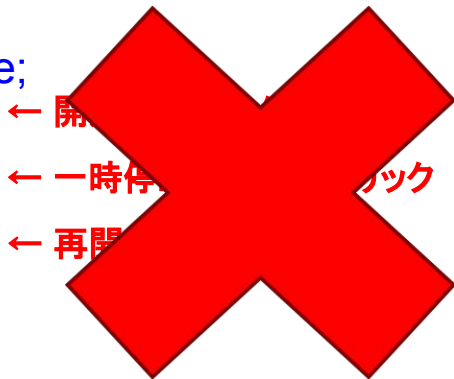
3. A-Frameで Hello WebVR



BRIDGE - Since 1974 -

マウスクリックで「再生→一時停止→再開」を制御してみよう

```
<a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"  
  animation="property: scale;  
    from: 1 1 1;  
    to: 1.5 1.5 1.5;  
    loop: true;  
    dur: 2000;dir: alternate;  
  startEvents: click;  
  pauseEvents: click;  
  resumeEvents: click;"  
></a-sphere>
```



想定通り動かない！！

← 開

← 一時停

← 再開

マウス
クリック

マウスクリックで「再生→一時停止→再開」を制御してみよう

① 独自コンポーネントを定義

```
<script>
```

```
var st = 0;
```

```
AFRAME.registerComponent('anime-control', { ← anime-control という名前でコンポーネントを定義
```

```
  init: function () {
```

```
    this.el.addEventListener('click', function (evt) { ← クリック時のイベントリスナーを登録
```

```
      if (st==0) {
```

```
        this.emit("start")
```

```
        st = 1;
```

```
      } else if(st==1) {
```

```
        this.emit("pause")
```

```
        st = 2;
```

```
      } else {
```

```
        this.emit("resume")
```

```
        st = 1;
```

```
      }
```

```
    });
```

```
  }
```

```
});
```

```
</script>
```

3. A-Frameで Hello WebVR



BRIDGE - Since 1974 -



マウスクリックで「再生→一時停止→再開」を制御してみよう

② オブジェクトにコンポーネントを設定

```
<a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"
```

```
  anime-control  ← 独自コンポーネントを設定
```

```
  animation="property: scale;
```

```
    from: 1 1 1;
```

```
    to: 1.5 1.5 1.5;
```

```
    loop: true;
```

```
    dur: 2000;
```

```
    dir: alternate;
```

```
    startEvents: start;
```

```
    pauseEvents: pause;
```

```
    resumeEvents: resume;"
```

← 開始/一時停止/再開イベントの設定

```
></a-sphere>
```


3. A-Frameで Hello WebVR



BRIDGE - Since 1974 -

視点操作(注視)でアニメーションを開始してみよう

```
<a-camera position="0 2 0" >  
  <a-cursor scale="5 5 5"  
    color="blue"  
    position="0 0 -2"  
    geometry="primitive: ring"  
    fuse="true"      ← 視点操作を利用する  
    fuse-Timeout="1000" ← 何ミリ秒の注視でクリック扱いとするか  
  ></a-cursor>  
</a-camera>
```

※視点操作を有効にするとマウスクリック
による操作が不可になります

視点操作(注視)でアニメーションを開始してみよう



```
<a-camera position="0 2 0" >
  <a-cursor scale="5 5 5"
    color="blue"
    position="0 0 -2"
    geometry="primitive: ring"
    fuse="true"
    fuse-Timeout="1000"
    animation__fusing="property: scale; startEvents: fusing; easing: easeInCubic; dur: 1500; from: 5 5 5; to: 0.1 0.1 0.1"
    animation__click="property: scale; startEvents: click; easing: easeInCubic; dur: 150; from: 0.1 0.1 0.1; to: 5 5 5"
    animation__mouseleave="property: scale; startEvents: mouseleave; easing: easeInCubic; dur: 500; to: 5 5 5"
  ></a-cursor>
</a-camera>
```

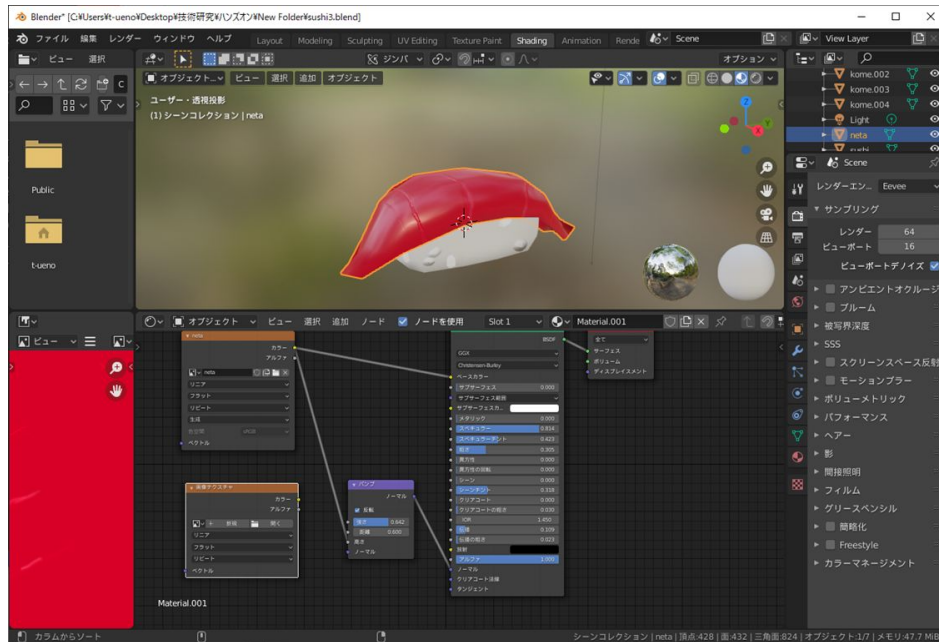
注視状態を分かりやすくするために、カーソルにアニメーションを追加

animation__fusing : 注視しはじめると、リングが小さくなる
animation__click : 注視でクリックイベントが発生すると、リングの大きさが戻る
animation__mouseleave : 注視中に視点を外すと、リングの大きさが戻る

3. A-Frameで Hello WebVR

自作の3Dモデルを表示してみよう

※3Dモデルは Blender などの一般的ツールから作成可能！



3. A-Frameで Hello WebVR

自作の3Dモデルを表示してみよう

※注意※

ローカルファイルではアセットを参照できないため、HTTPサーバが必要！



自作の3Dモデルを表示してみよう

① 自作の3Dモデルを「アセット」として登録

```
<a-assets>
```

```
  <a-asset-item id="sushi1" src="akami.glb"></a-asset-item>
```

```
  <a-asset-item id="sushi2" src="tamago.glb"></a-asset-item>
```

```
  <a-asset-item id="sushi3" src="ebi.glb"></a-asset-item>
```

```
</a-assets>
```

3. A-Frameで Hello WebVR

自作の3Dモデルを表示してみよう



② シーンに配置

```
<a-gltf-model
```

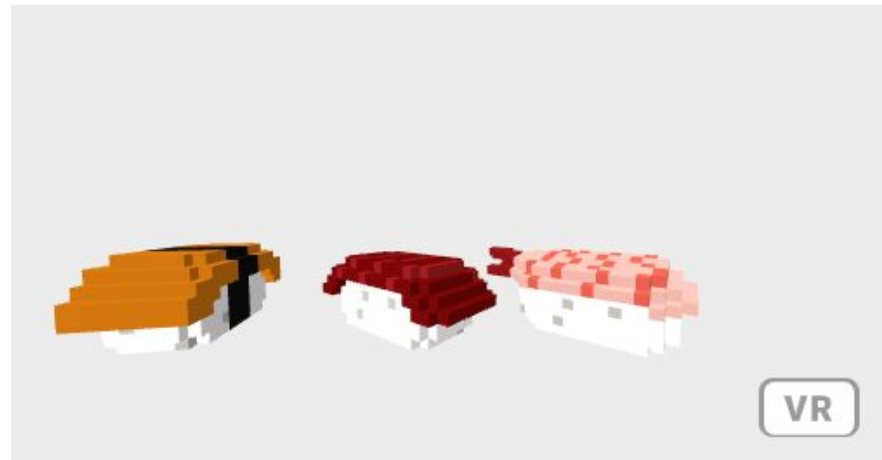
```
  src="#sushi1"
```

```
  position="-0.5 0.5 -3"
```

```
  rotation="0 50 0"
```

```
  scale="1.0 1.0 1.0"
```

```
></a-gltf-model>
```



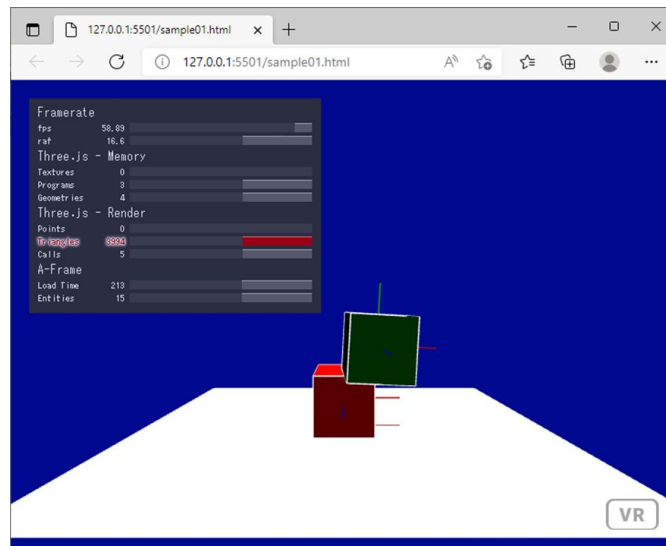
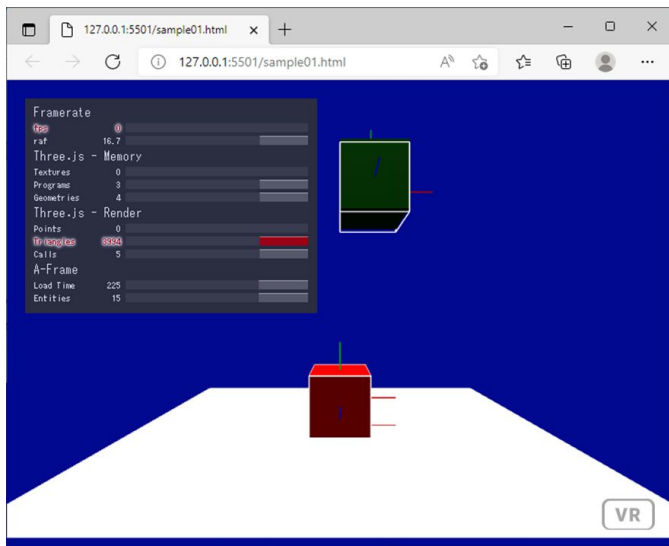
3. A-Frameで Hello WebVR

応用編(こんなこともできる！)



① 物理演算ライブラリの利用

Ammo.js と frame-physics-system.js を利用して物理演算が可能！



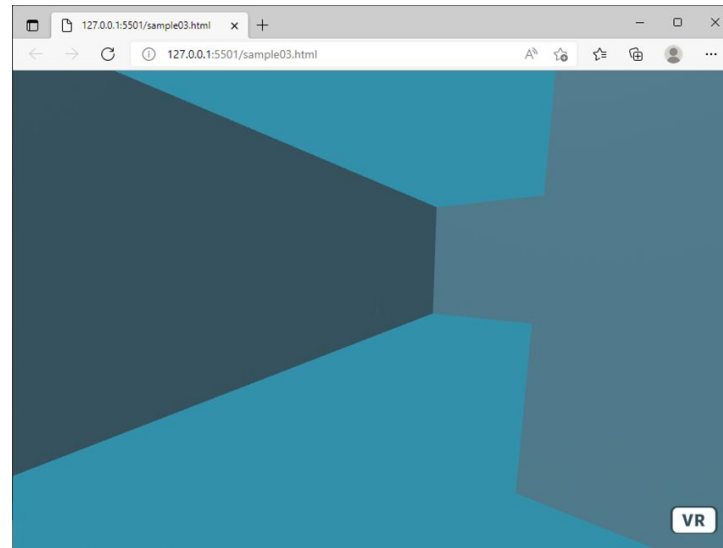
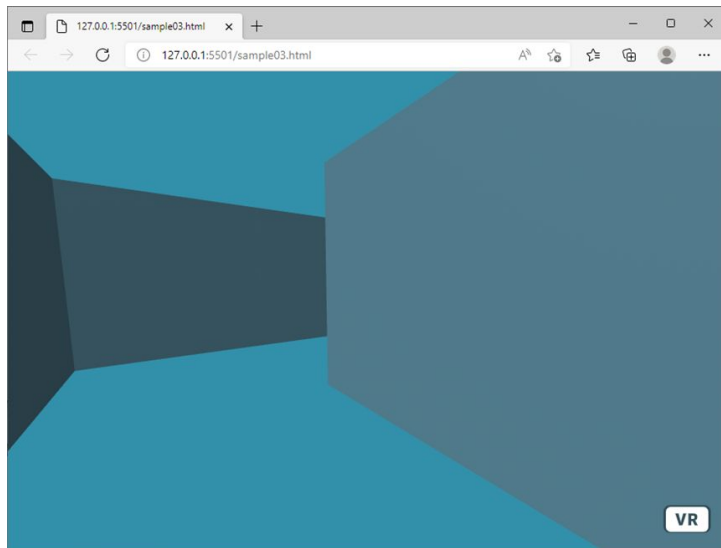
3. A-Frameで Hello WebVR

応用編(こんなこともできる！)



② WebVRで迷路

プレイヤー(カメラ)と壁の衝突判定を実装して迷路を作成



Web-xR/ A-Frame の特徴

HTML+ α なので、テキストエディタがあれば開発可能！
⇒ VRや3D開発で要求されがちな「PCスペック」を気にしなくてよい！

テキストベースで開発可能なので、VRや3Dに慣れていない技術者でも
とっつきやすい！

Webアプリなので、PC・VRゴーグル・スマホなどの端末やOSを選ばない！
⇒ URLを公開すればOK！専用アプリのインストールが不要で体験できる！



xRの入門として ちょうどよい！！

W

A-Frame で xR 始めてみませんか？

IoT活用事例紹介

株式会社ブリッジ 荻原 宙佳 ジェン

目次

1. はじめに
2. 抱えている問題と解決方法
3. ツールの構成
4. デモ
5. 実装上のポイント
6. まとめ

(用語説明)IoT とは？

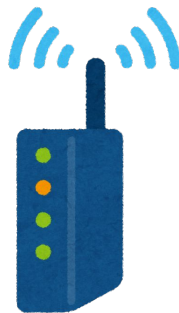
デバイス・センサetc.

“モノ”(Things)



現場の状態をデジタル化

インターネット



現場とクラウドでつなげる

クラウド



デジタルデータを活用

1. はじめに

不便なこと, 改善したいこと、ありませんか？

家のペットの様子を
どこに居ても確認したい！

窓やドアの鍵、
閉めたっけ？！

ポットのコーヒー、
少なくなったら
知らせてほしい...



身の回りにある問題はIoTで解決できるかも！

弊社内にある困りごとの一部をIoTを活用して、問題解決に挑んできました。

本日はその一部を活用事例としてご紹介します！



1. はじめに

今まで作成してきたIoTツール



去年の動画はYouTubeに
アーカイブがあります！
興味のある方は是非！



<https://www.youtube.com/watch?v=4QAUe3J8DgM>

1. はじめに

今まで作成してきたIoTツール

今年は『エアコン監視ツール』を作成！



<https://www.youtube.com/watch?v=4QAUe3J8DgM>

作成したツールの目的

【監視対象】

- ・サーバー室のエアコン×2台

【ツールの目的】

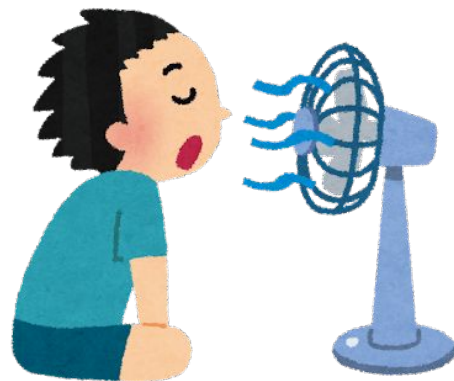
- ・**目視(カメラ)**と**温度計**を使ってエアコンが正常に動いているか、
どこからでも確認できる

【ツールを作った背景】

- ・リモートワークが主流になり、出社する機会が減少
⇒エアコンの稼働状況＋室内を簡単に見れるツールを開発

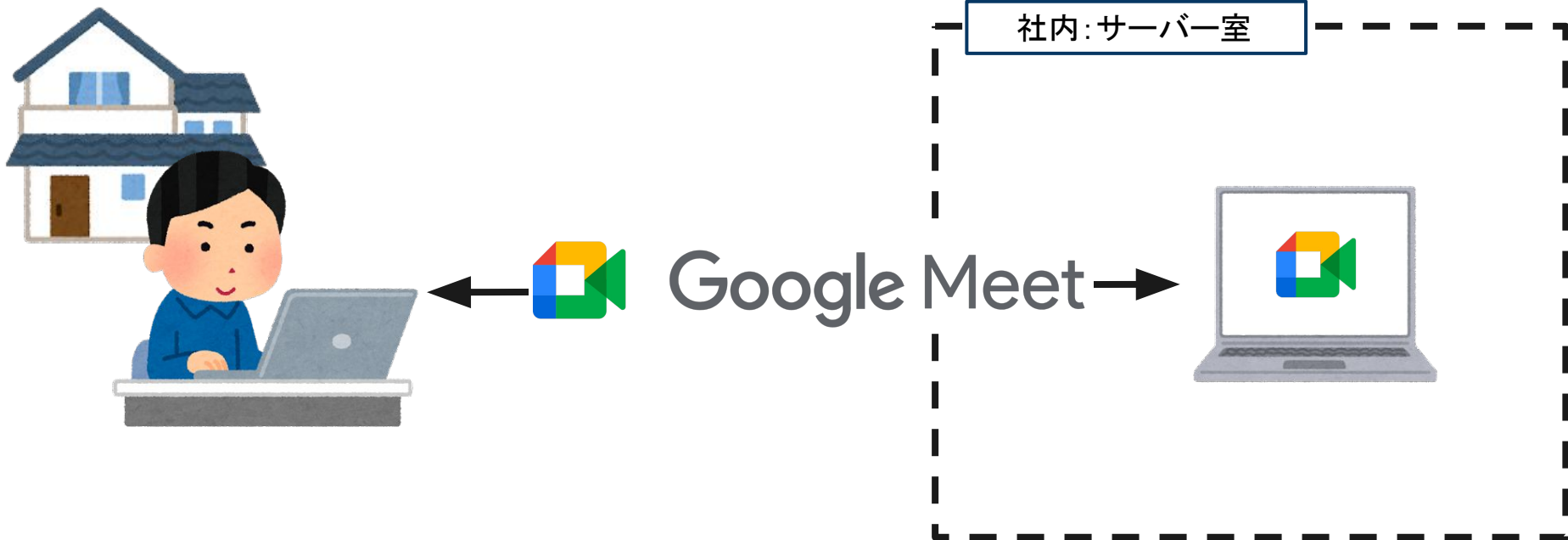
【備考】

- ・既にエアコン監視ツールはある
- ・今回チャレンジしたのは、現行版の改良



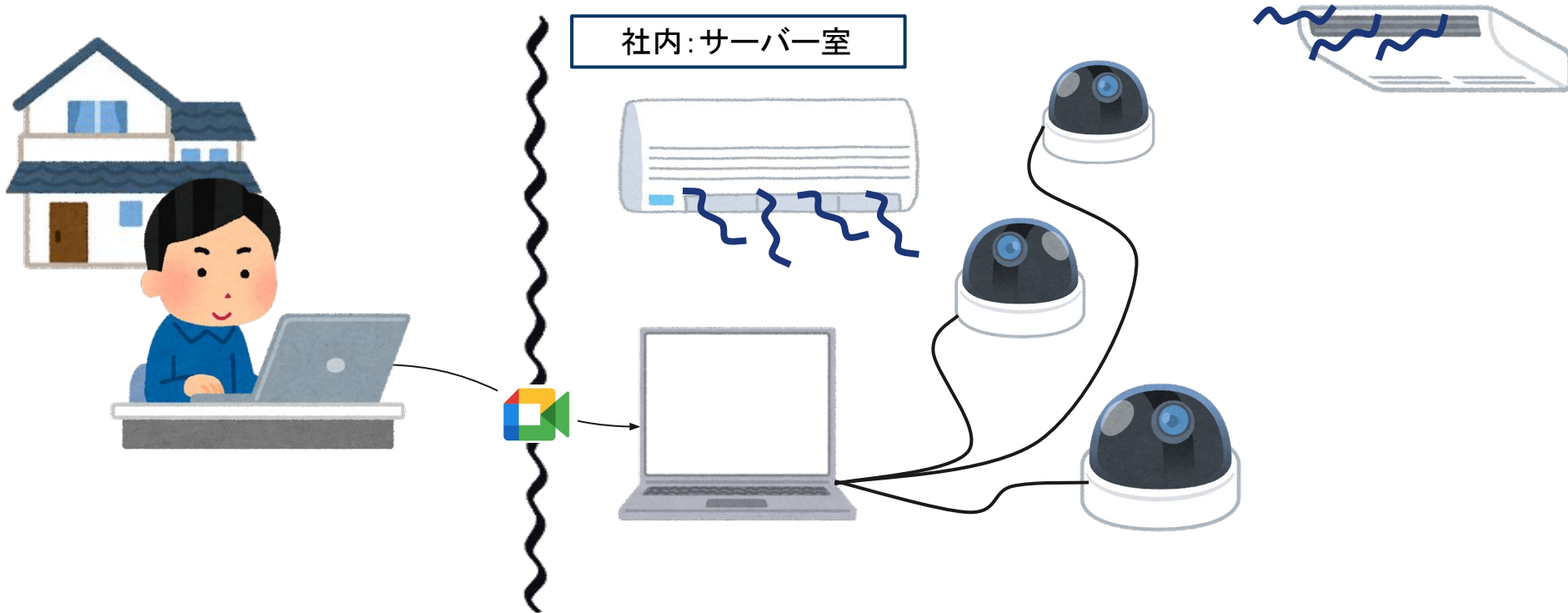
1. はじめに

現状使っているツールの構成



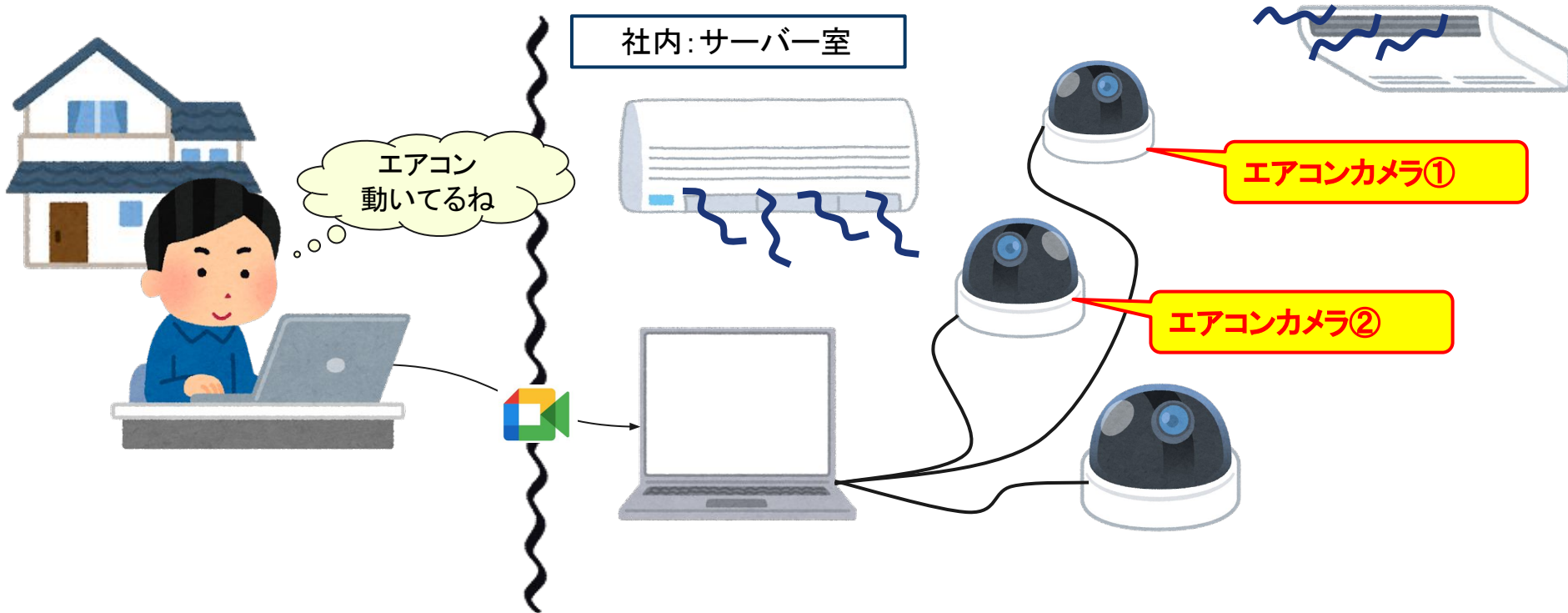
1. はじめに

現状使っているツールの構成



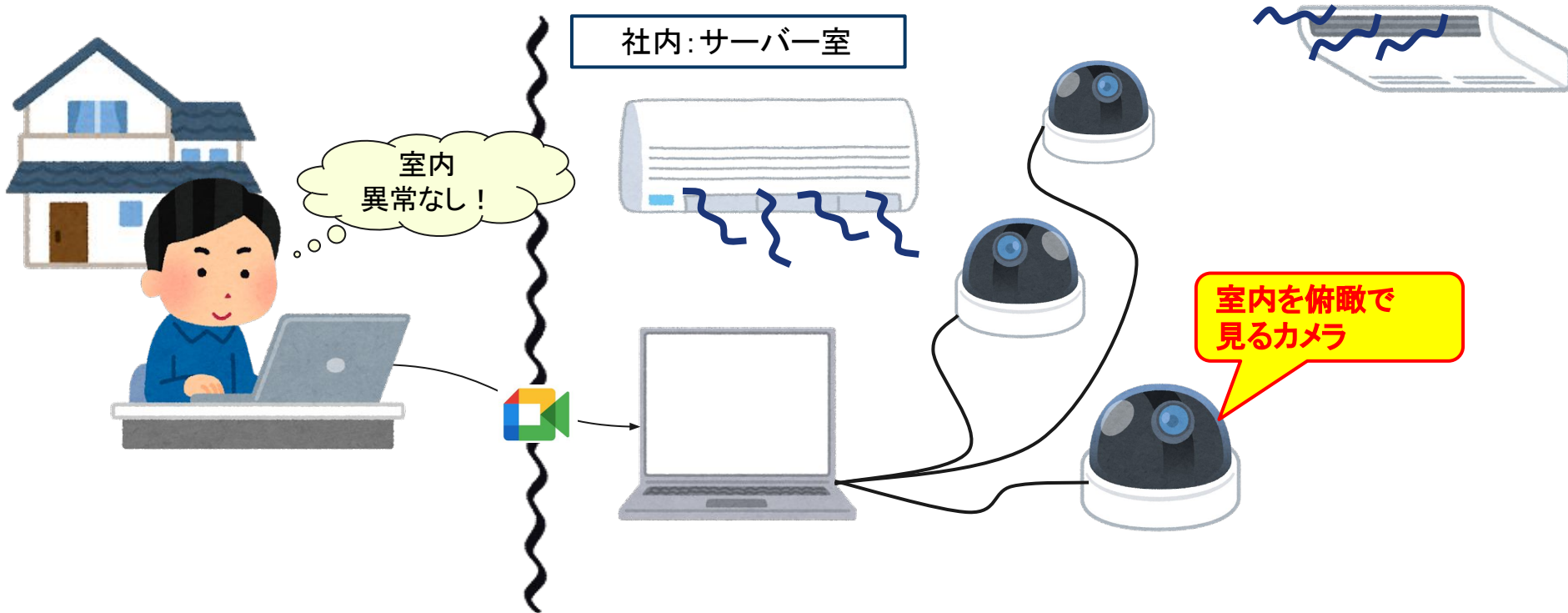
1. はじめに

現状使っているツールの構成



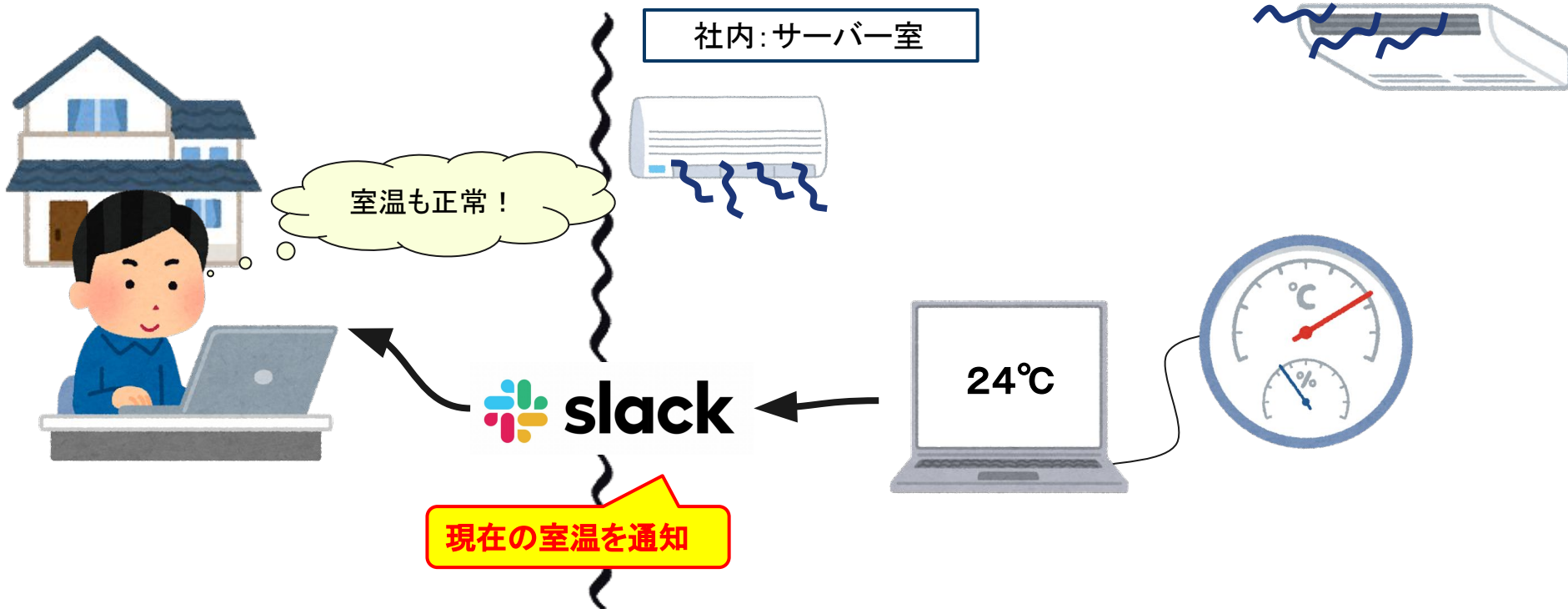
1. はじめに

現状使っているツールの構成



1. はじめに

現状使っているツールの構成



抱えている問題

- ① **意図しないタイミングで再起動される** (例: Windows Update等)
 - 定期的に様子を見に行く必要があり、手間がかかる
 - 再起動してしまうと、設定のために出社しなくてははいけない
- ② PCにUSBカメラを3台接続しているためか、**処理落ちしている**
 - 映像が乱れてしまう
- ③ **設置用スペース**を取られてしまう
 - スペースが限られているので、なるべく小型化したい

解決案

①意図しないタイミングで再起動される（例：Windows Update等）

⇒**ラズパイであれば、勝手に再起動されない想定**

②PCにUSBカメラを3台接続しているためか、**処理落ちしている**

⇒処理落ちしないように、**実装面でカバー**

③**設置用スペース**を取られてしまう

⇒**ラズパイを使うことで、省スペース**

PCよりも使用する電力が少ないので、**省エネにも！**

3. ツールの構成

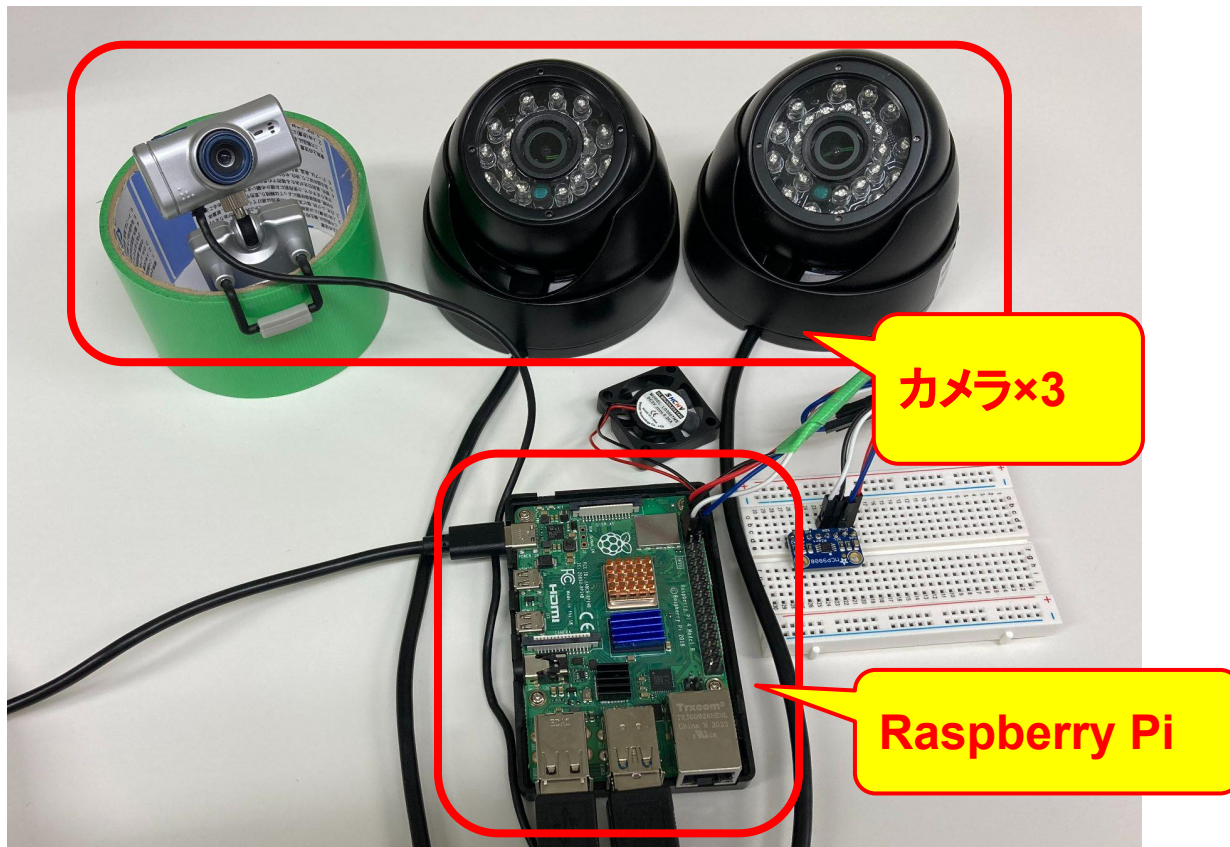
構成(部品)



※本資料に記載されているロゴ、システム名称、 企業名称、 製品名称は各社の登録商標または商標です

3. ツールの構成

構成(部品)



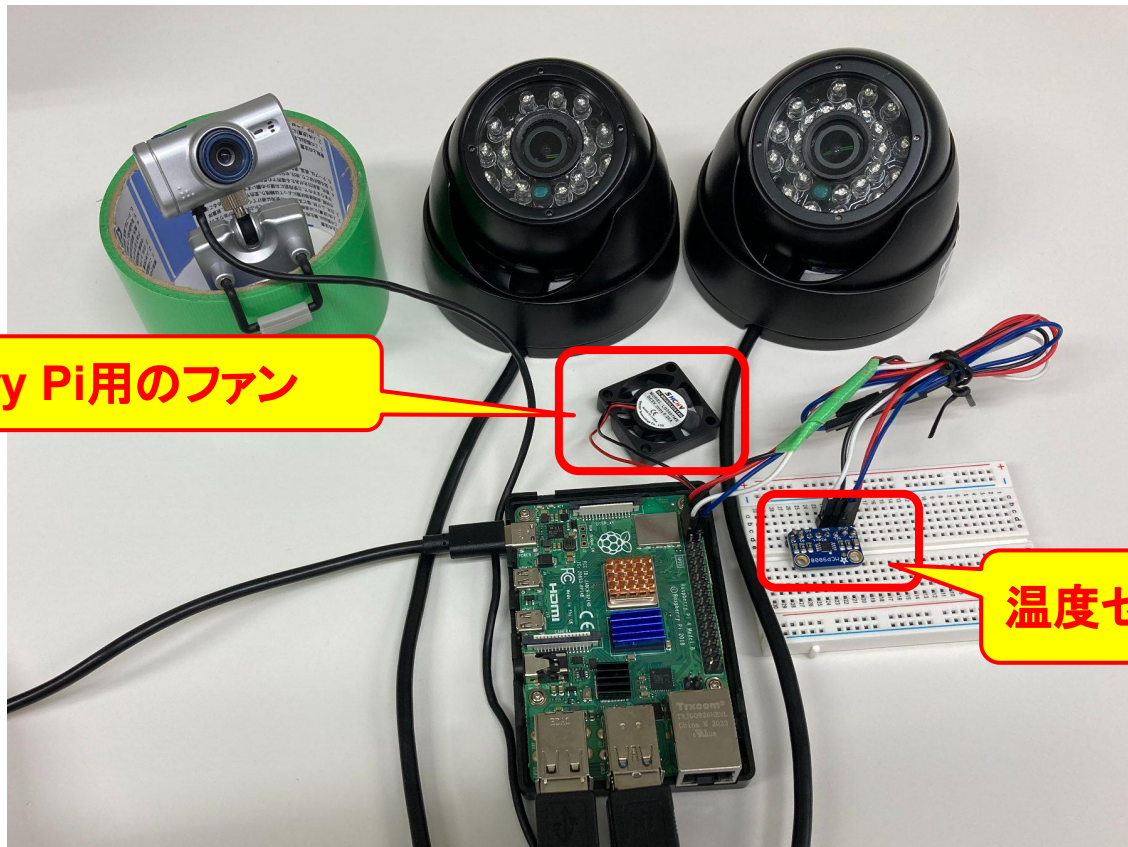
※本資料に記載されているロゴ、システム名称、企業名称、製品名称は各社の登録商標または商標です

3. ツールの構成

構成(部品)

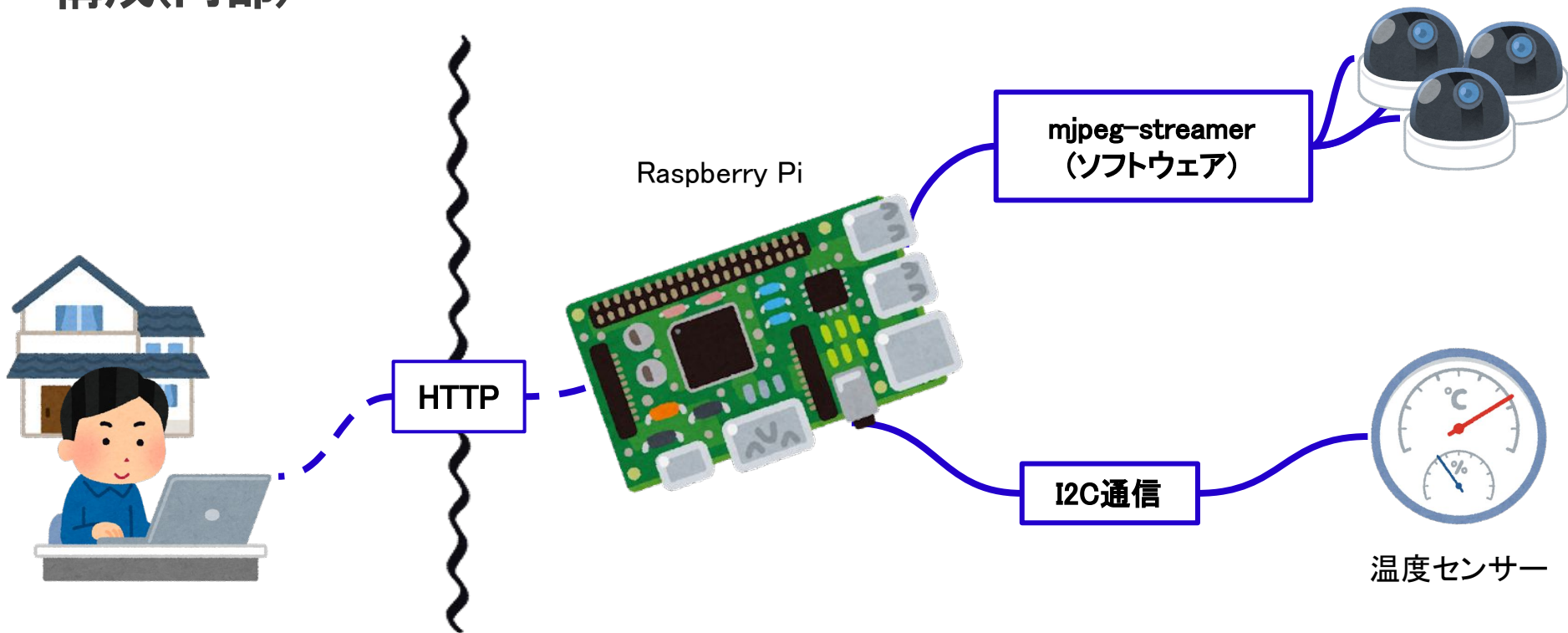
Raspberry Pi用のファン

温度センサー



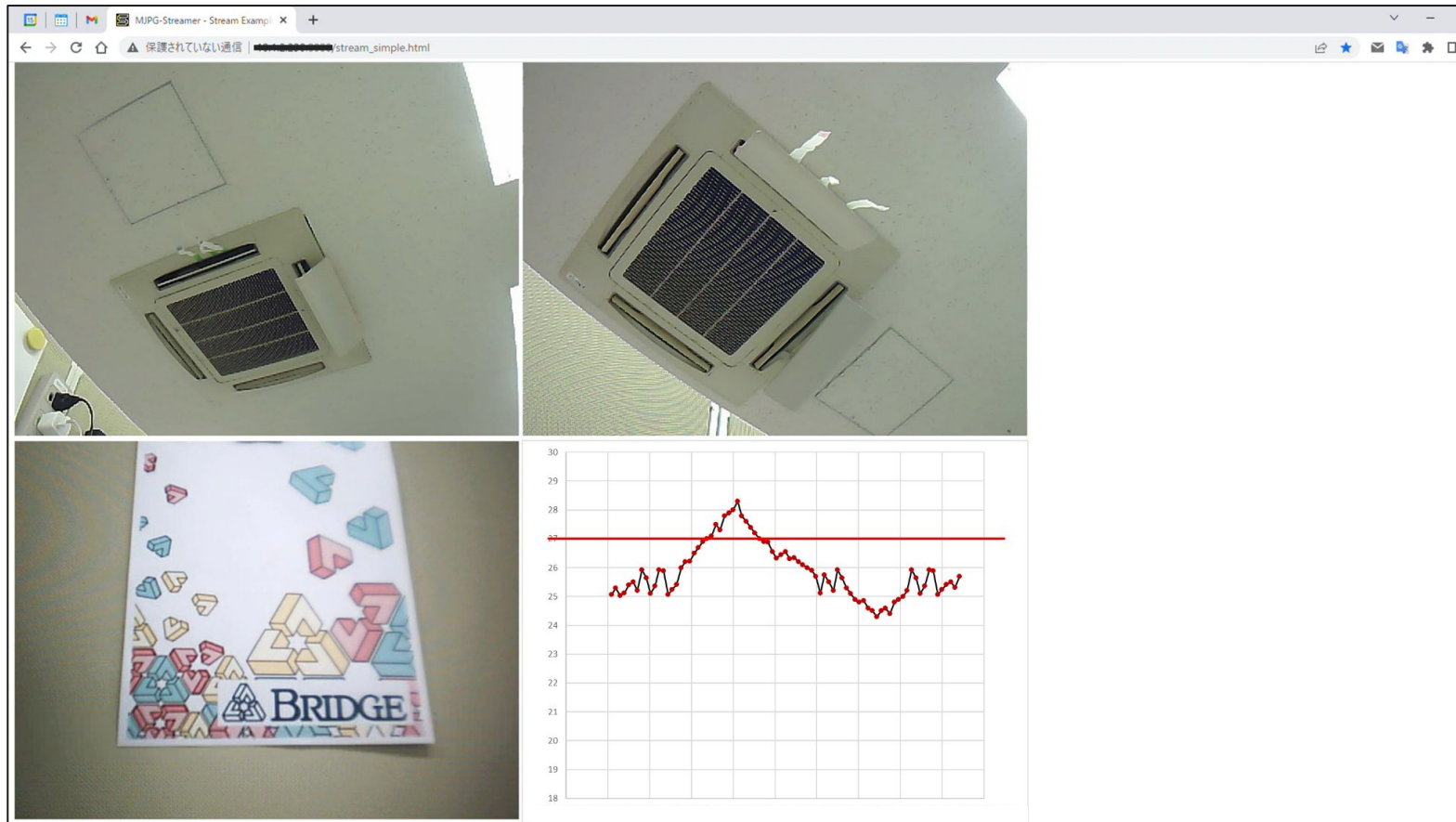
3. ツールの構成

構成(内部)



3. ツールの構成

画面



※本資料に記載されているロゴ、システム名称、 企業名称、 製品名称は各社の登録商標または商標です

3. ツールの構成

RaspberryPi

- ・通称:ラズパイ
- ・使用モデル:Raspberry Pi 4(メモリ:4GB)

【Raspberry Pi でできること】

①カメラ活用

- ・監視カメラ
- ・AIカメラ

②プログラミング

- ・Python
- ・Scratch

③電子工作

- ・LED/モーター
- ・センサー

④サーバー

- ・Webサーバー
- ・ファイルサーバー

⑤ゲーム

- ・MINE CRAFT(ラズパイ版)



3. ツールの構成

MCP9808(温度センサー)

- ・高精度な温度センサー
- ・シンプルな制御でRaspberry Piでも使用可能(I2C通信)
- ・範囲 : $-40^{\circ}\text{C} \sim +125^{\circ}\text{C}$
- ・標準精度: 0.25°C
- ・用途 : 工業用冷凍機, 冷蔵庫
食品加工
PC, 家電, ポータブルデバイス等の温度管理



ガイド: <https://learn.adafruit.com/adafruit-mcp9808-precision-i2c-temperature-sensor-guide>

3. ツールの構成

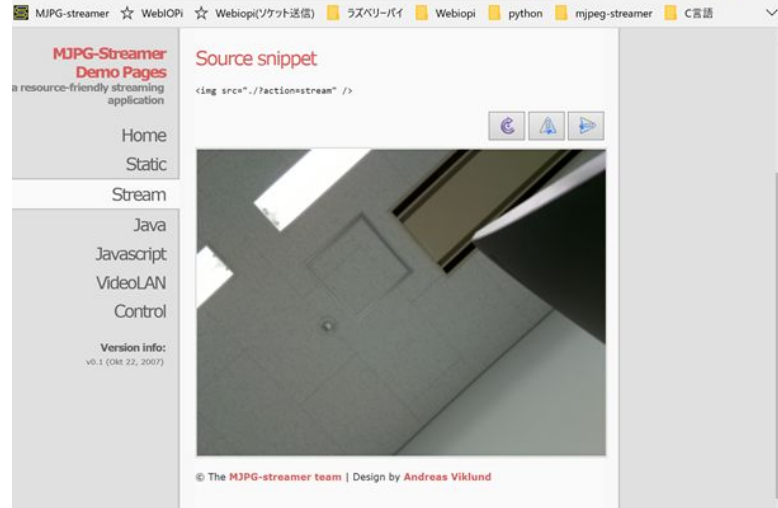
mjpg-streamer

- ・ラズパイでカメラの映像をストリーミング配信するためのソフトウェア
- ・Webカメラ1台であれば、インストール, 起動シェルの実行ですぐ使える
- ・カメラの画質の指定は、起動シェルを一部修正するだけでOK

【仕組み】

- ①Raspberry Pi上でWEBサーバーを立ち上げる
- ②カメラ画像を投影

GitHub:<https://github.com/jacksonliam/mjpg-streamer>



4. デモ

デモ



実装上のポイント(躓きポイント)

- ・複数台のカメラをラズパイに接続、ストリーミング配信したい場合、
mjpg-streamerの**起動シェルに修正が必要**

- ・温度はセンサーから送られてくる**バイトデータ**を加工する必要あり

※参考URL: <https://www.pentacreation.com/blog/2016/12/161214.html>

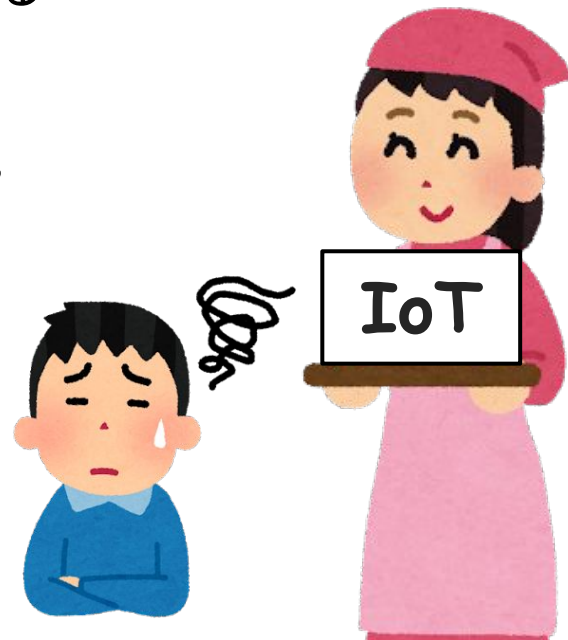
- ・SDカードは寿命があるため、**データが壊れてしまう可能性**がある
⇒短期動かす(開発中)場合は、SDカード
⇒長期間動かす(運用)場合は、USBにデータ移行するのがおすすめ

まとめ

IoTは問題解決するための手段になる！

- 監視カメラであれば、2時間程度で作成できる

⇒身近な問題はIoTでお手軽に解決できるかも？



おわりに

弊社は、積極的に技術研鑽に取り組んでいます。

今後も、このような場で研究成果を発表していきますので、

ブリッジを見かけた際は、是非、覗いてみてください！！



株式会社ブリッジ

BRIDGE to the brilliant tomorrow