



WoT セッション2: WoT の仕様と実装 - WoT の作り方/使い方

東村 邦彦 / 浅井 智也

2022.3.11

Web of Things (WoT)の標準化内容

主な標準化文書は以下の5つ

Architecture: 全体アーキテクチャ

IoTデバイス/プラットフォームの概念モデル

Thing Description: WoTデバイスの記述方式

WoTデバイスの記述文書に記載すべき情報

Protocol Binding: プロトコル記述語彙の定義

HTTP、CoAP、OPC UA、Modbus/TCPなどの産業向けプロトコルをTDで記述するための語彙

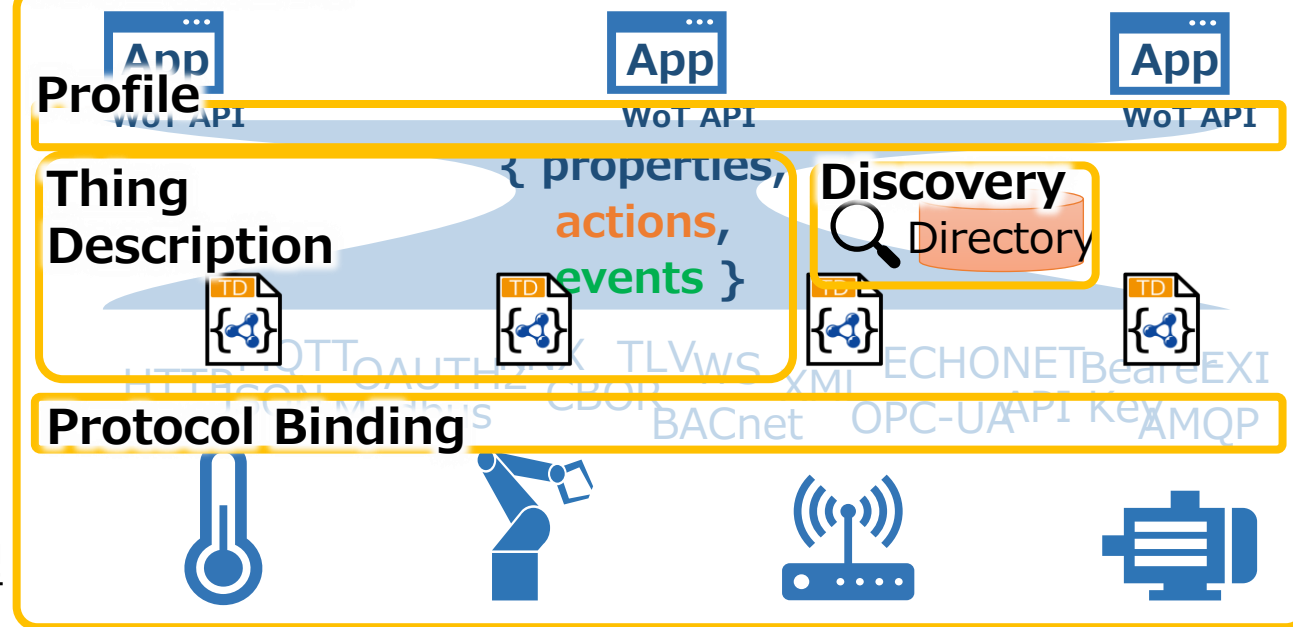
Profile: WoT APIの典型例

新たにWoT対応デバイス/アプリを開発する際に参考となる典型的なAPIをプロフィールとして定義

Discovery: WoTデバイスの探索・発見

既存機器発見メカニズムを使ったThing発見のための方法およびThing DescriptionディレクトリのAPIを定義

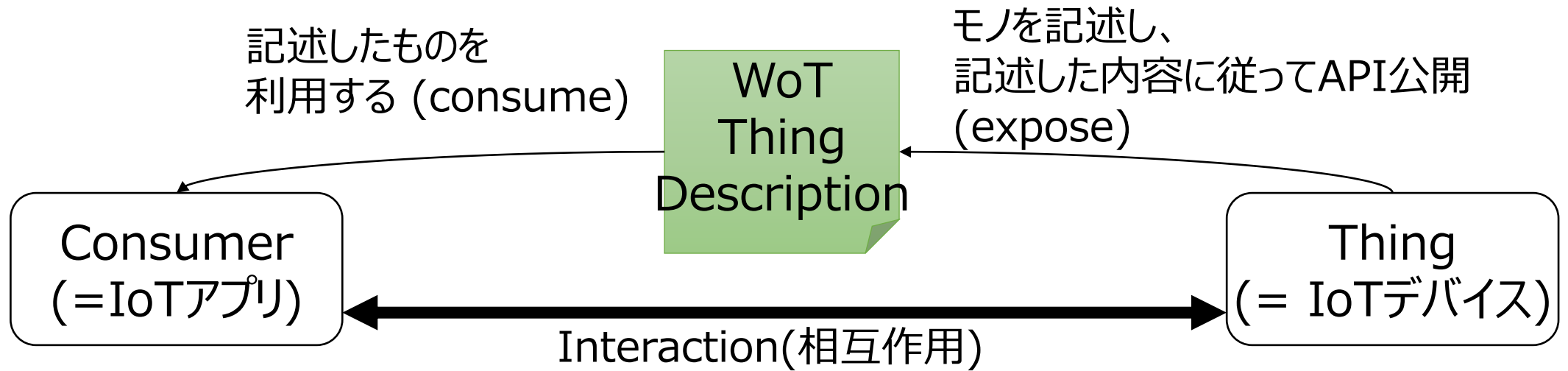
Architecture



Web of Things全体アーキと各標準化項目の対応

Architecture (1/2): 基本モデル

Web of Thingsの基礎として最小限のモデルを定義



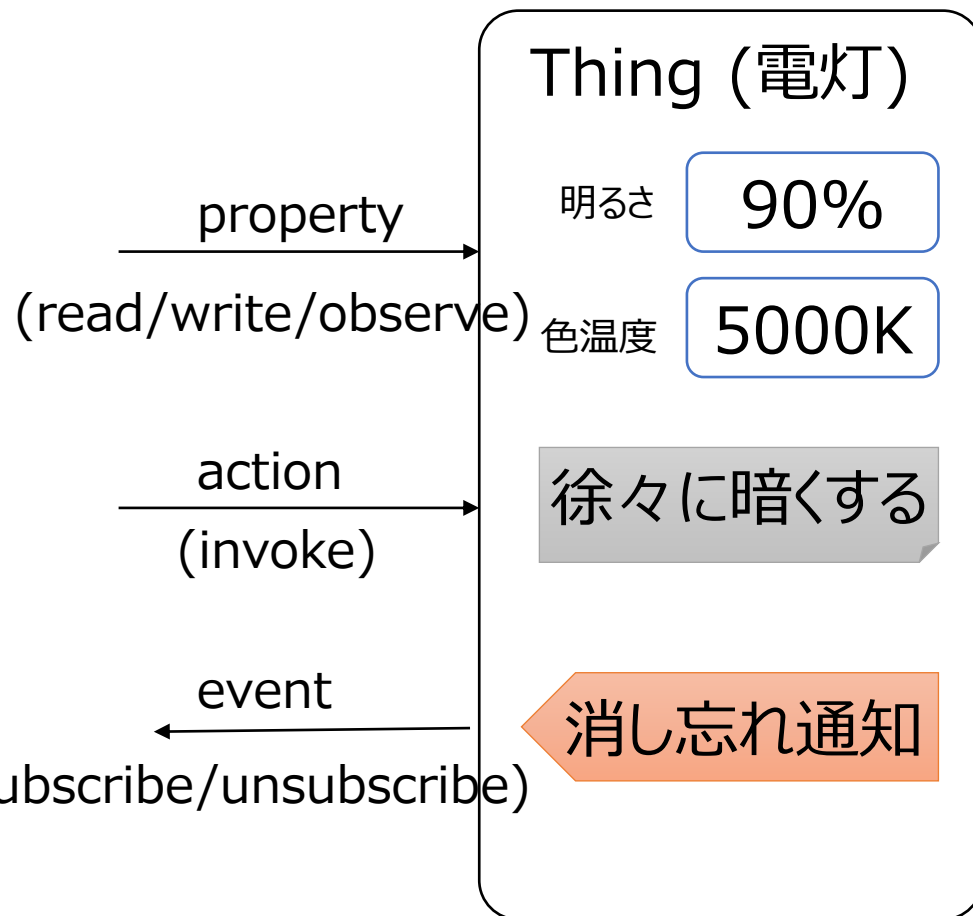
Architecture (2/2): 相互作用のモデル化

Thingとの相互作用を Property・Action・Eventの3つのパターンでモデル化

- Property: Thingの状態などへの読み書き
 - 例: センサの計測値の読み込み、設定値の書き込み

- Action: Thingが持つ機能の実行
 - 例: 複数のプロパティの変更、時間のかかる動作の実行

- Event: ThingからConsumerへの通知
 - 例: 値の変更通知、警告、定期的な状態通知



Thing Description: Thingの記述方式

そのモノがなんであるか(What)とモノと相互作用するための方法(How)を機械可読なフォーマットで記述

- IoTサービスのメタデータ
 - ID, バージョン, 種類, 作成日, ...
 - 名称, 説明, ...
- 相互作用の記述
 - プロパティ、アクション、イベント
 - どうやって使うのか(プロトコルバインディング)
 - データをどう解釈するのか (スキーマ)
- JSON-LD 1.1に準拠
 - 語彙の拡張
 - プロトコルごとの語彙

```
{
  "@context": [
    "https://www.w3.org/2019/wot/td/v1",
    { "iot": "http://iotschema.org/" }
  ],
  "id": "urn:dev:org:32473:1234567890",
  "title": "MyLEDThing",
  "description": "RGB LED torchiere",
  "@type": ["Thing", "iot:Light"],
  "securityDefinitions": [{"default": {
    "scheme": "bearer"
  }
}],
  "security": ["default"],
  "properties": {
    "brightness": {
      "@type": ["iot:Brightness"],
      "type": "integer",
      "minimum": 0,
      "maximum": 100,
      "forms": [ ... ]
    }
  },
  "actions": {
    "fadeIn": {
      ...
    }
  }
}
```



実装 (1/6)

Web of Things Working Groupで主に利用されているものとして Node-wot と Node-RED Node Generatorがある

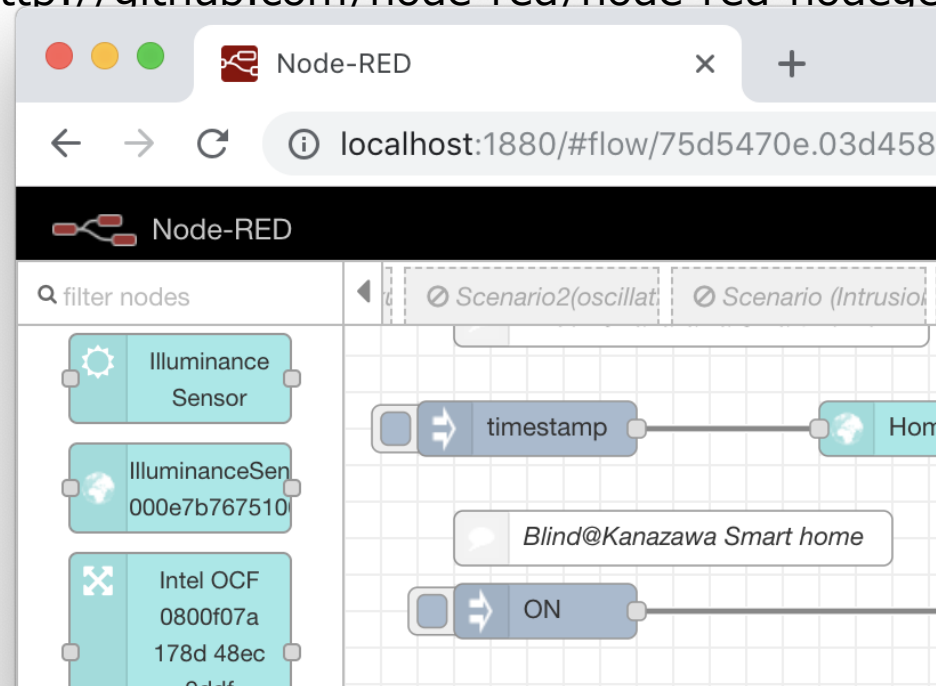
Node-wot

- WoT Scripting APIのリファレンス実装
- <https://thingweb.io>

```
1 let thing = WoT.produce({
2   title: "counter",
3   description: "counter example Thing",
4   "@context": ["https://www.w3.org/2019/wot/td/v1", {"iot": "http://example.org/td/v1"}],
5   properties: {
6     count: {
7       type: "integer",
8       description: "current counter value",
9       "iot:Custom": "example annotation",
10      observable: true,
11      readOnly: true
12    }
13  },
14  actions: {
15    increment: {
16      description: "Incrementing counter value (with optional step parameter)",
17      uriVariables: {
18        step: { "type": "integer", "minimum": 1, "maximum": 250 }
```

Node-RED Node Generator

- OpenAPI, WoT TDなどの定義ファイルからNode-REDのノードを自動生成
- <http://github.com/node-red/node-red-nodegen>

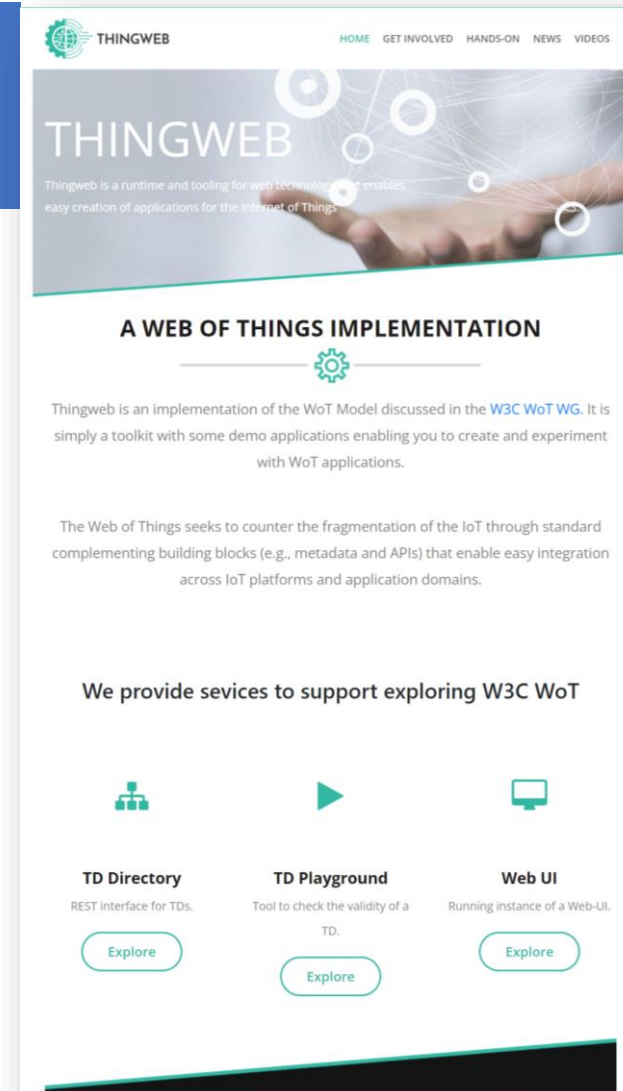


実装 (2/6): Eclipse Thingweb node-wot



WoT Scripting API仕様に基づいた オープンソース実装

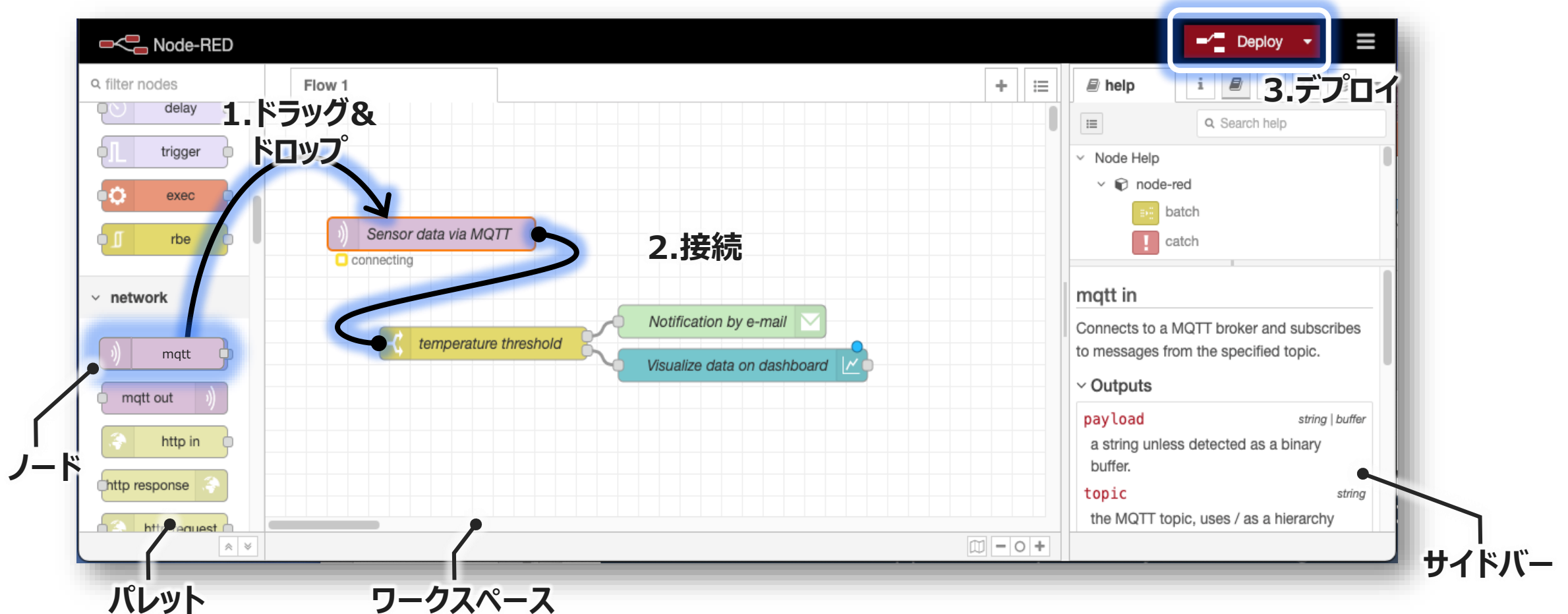
- Eclipse Thingwebプロジェクトによる実装
 - 主要な実装者: Siemens, ミュンヘン工科大
- 未標準化の機能も含め、多様な機能を実装
 - プロトコルサポート: HTTP(S), CoAP(S), MQTT, Firestore, Websocket, OPC UA, NETCONF, Modbus, M-Bus
 - メディアタイプ: JSON, Text(HTML, CSS, XML, SVG), Base64(PNG, JPEG, GIF), Octet stream, CBOR(予定), EXI(予定)
- Node.jsやWebブラウザ内で利用可能
- 多くのプロジェクトで実装のベースとして利用されている



<https://www.thingweb.io/>

実装 (3/6): Node-RED Node Generator - Node-RED概略

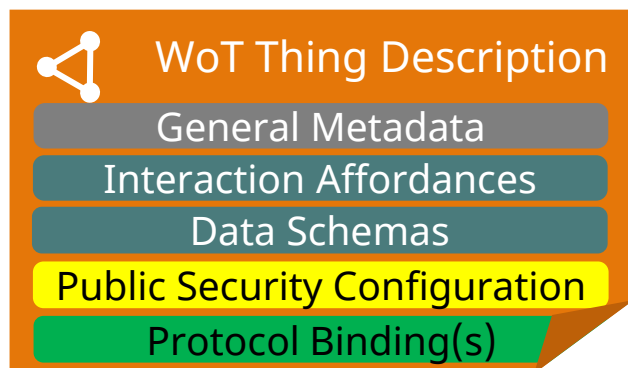
IoTおよびシステム連携アプリケーションのためのビジュアルプログラミングツール
ブラウザベースで高機能部品(ノード)を接続するだけでローコード開発を実現



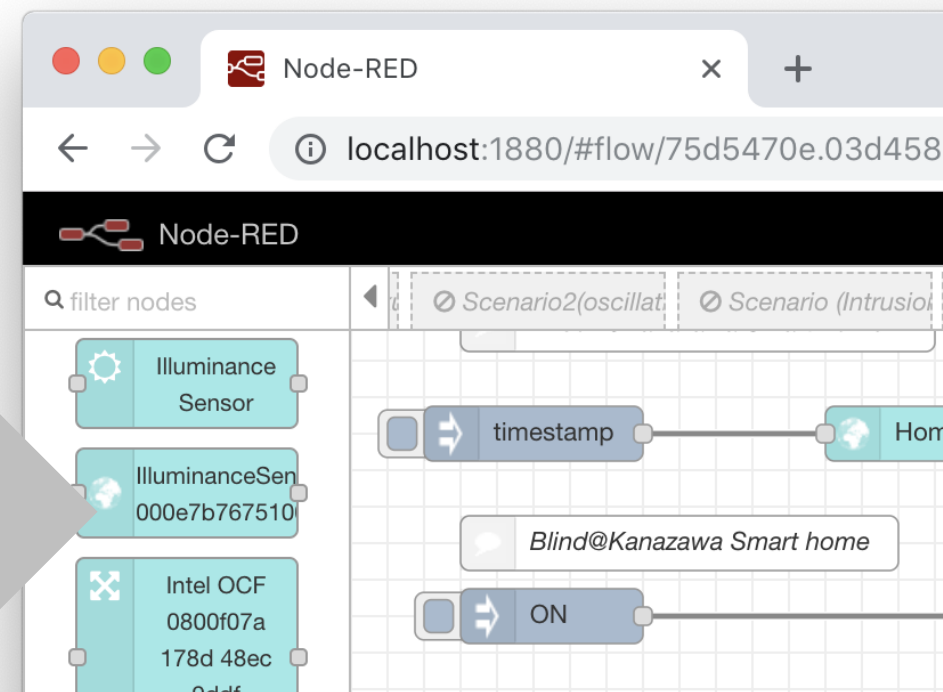
実装 (4/6): Node-RED Node Generator – 特長

Thing Descriptionで表現されたモノが Node-REDのノードとして直感的に扱える

- 「モノ」が、Node-REDエディタ上の1つの「ノード」として見える
 - モノと相互作用したければ、対応するノードにメッセージを送ればよい
 - モノから情報を得るためには、対応するノードからのメッセージを受ければよい



**Node
Generator**



実装 (5/6): Node generatorの利用例 – ランプの制御

The screenshot displays a web browser window with the Node-RED interface. The left sidebar shows a list of nodes categorized under 'input' and 'output'. The main workspace is a grid for building flows. The right sidebar contains an 'info' panel with details about the current flow.

Node-RED Interface Details:

- Flow 1:** The active flow is named 'Flow 1'.
- Input Nodes:** inject, catch, status, link, mqtt, http, websocket, tcp, udp.
- Output Nodes:** debug, link.
- Information Panel:**

Information	
Flow	"8e608fd6.8e624"
Name	Flow 1
Status	Enabled
- Description Panel:** Currently empty.

The background image shows a room with a desk, a monitor, a lamp, and a small robot on the floor. A blue user icon labeled 'You' is visible in the bottom right corner of the room scene.

実装 (6/6): 生成したノードを利用したGUIからのThing操作

