



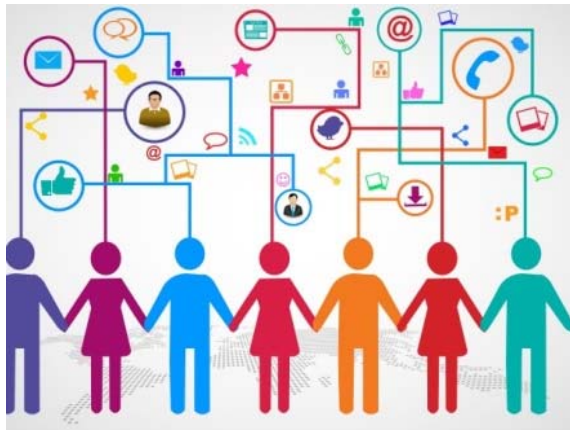
# Outline

- The Evolution of Communication
- What and Why is IoTtalk?
- How to deploy?
- Device Connections with IoTtalk
- IoTtalk System Architecture
- Web-based IoT Applications
- Conclusion

# The Evolution of Communication

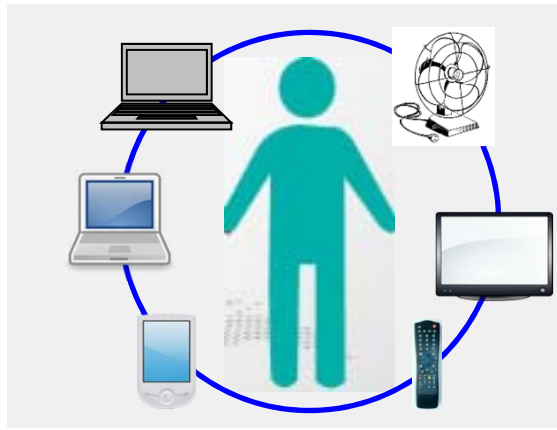
H2H → H2M → M2M

A humans talks to the other humans.



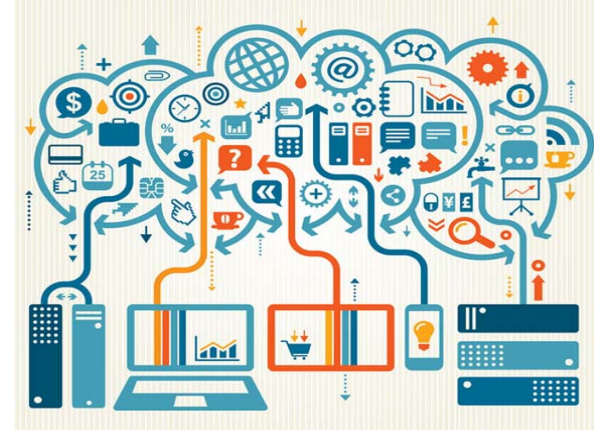
The human talks to the other humans.

A human directly controls the machines.



The human manipulates the machine himself.

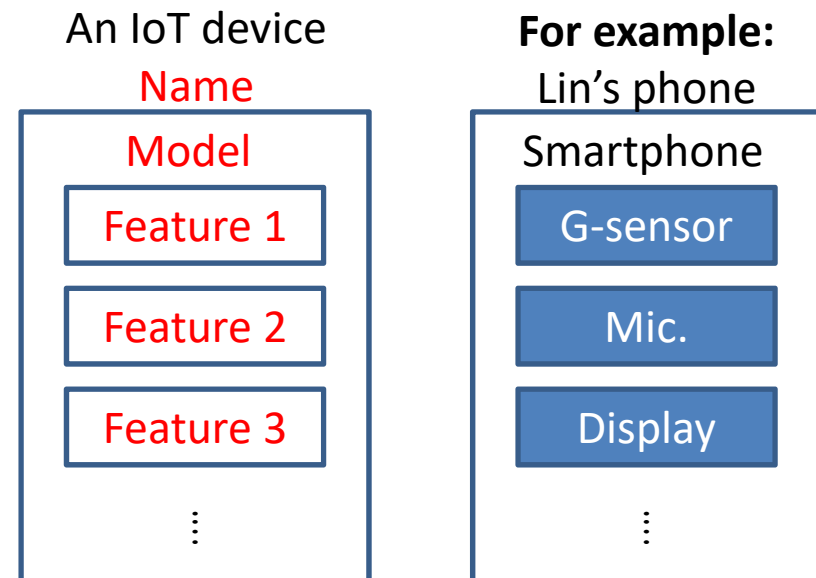
A machine automatically controls the other machines.



Machines have logic or intelligence to manipulate other machines. That is, **IoT devices can talk to each other, IoTtalk!**

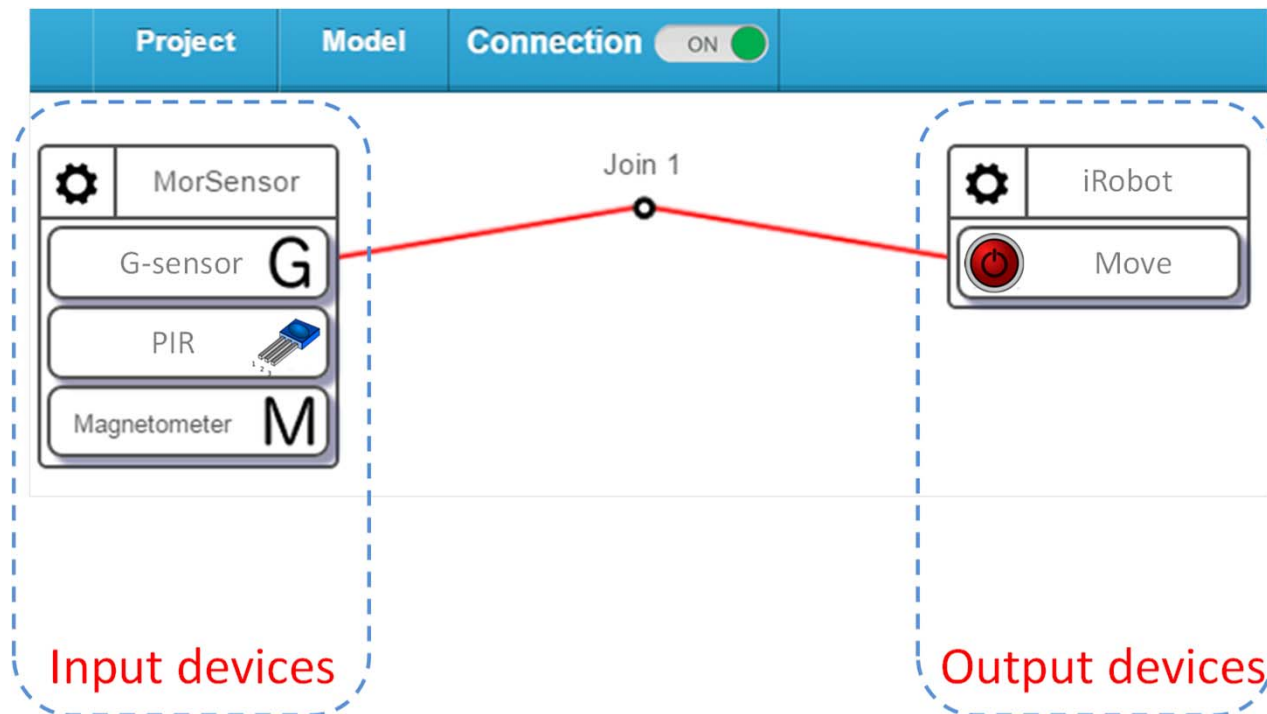
# What is IoTtalk?

- **IoTtalk** is an IoT device management tool
- **IoT management concept**
  - **Device Feature**
    - The function or capability which an IoT can provide
  - **Device Model**
    - A set of device features
    - A device model refers to a specific product
  - **Device name**
    - The name of a specific product



# Why is IoTtalk?

- Applications can simply develop with lower efforts
- Simple and intuitive GUI
- Application development without real devices is feasible



# Simple and Intuitive GUI

- Connections by intuitional drawing links between IoT devices
- Transparently observe the connections between IoT devices
- Monitor the transmitting values between IoT devices
- Application debugging is more easier

The screenshot shows the IoTalk web interface. The browser address bar displays `140.113.199.200:7788/connection/21#`. The interface has a blue header with tabs for **Project21**, **Model**, **Connection** (with an ON toggle), and **Delete**. On the right of the header, there is a **Simulation** section with an ON toggle.

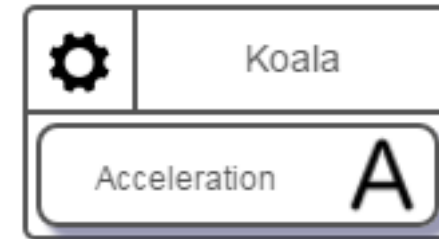
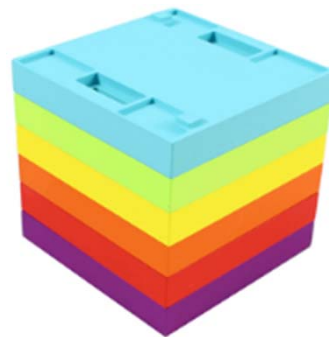
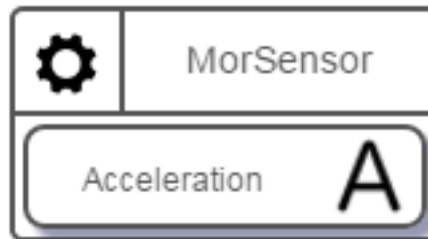
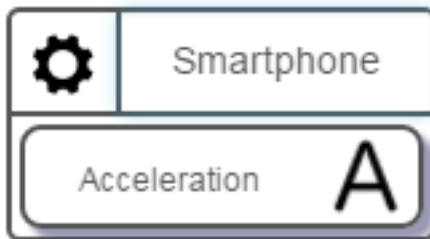
The main workspace shows two IoT device blocks connected by a red line. The left block is labeled **Koala** and contains an **Acceleration** sensor (A). The right block is labeled **Dandelion** and contains a **Color-O** sensor (C). The connection point is labeled **Join 1**.

On the right side, there are two monitoring panels:

- IDF Monitor**: Sub-stage is **Input**. It displays a table of data with columns for **Timestamp**, **X<sub>1</sub>**, **X<sub>2</sub>**, and **X<sub>3</sub>**. The data points are: 04:10:11 (0.56, 0.30, 0.16), 04:10:12 (0.69, 0.78, 0.79), 04:10:13 (0.14, 0.81, 0.61), and 04:10:15 (0.30, 0.79, 1.00). There are **Continue** and **Next** buttons, and a dropdown for **1 Acceleration**.
- ODF Monitor**: Sub-stage is **Function**. It displays a table of data with columns for **Timestamp**, **Y<sub>1,F</sub>**, **Y<sub>2,F</sub>**, and **Y<sub>3,F</sub>**. The data points are: 04:10:11 (0.56, 0.30, 0.16), 04:10:12 (0.69, 0.78, 0.79), 04:10:13 (0.14, 0.81, 0.61), and 04:10:15 (0.30, 0.79, 1.00). There is a dropdown for **1 Color-O**.

# Applications can simply develop with lower efforts

- Reusable DF modules
  - Even they are different IoT devices



# Application development without real devices is feasible

- Do not need the real devices first to develop applications
- The simulator provides the numerical values as inputs



The screenshot shows a software interface for a simulation. At the top, there is a blue header bar with the word "Simulation" and a green toggle switch labeled "ON". Below this, the main area is titled "IDF Monitor". It features a "Sub-stage:" dropdown menu set to "Input", and buttons for "Continue", "Next", and "1 Acceleration". A table displays the following data:

Timestamp	$x_1$	$x_2$	$x_3$
04:10:11	0.56	0.30	0.16
04:10:12	0.69	0.78	0.79
04:10:13	0.14	0.81	0.61
04:10:15	0.30	0.79	1.00

# Easy to Deploy and Operate



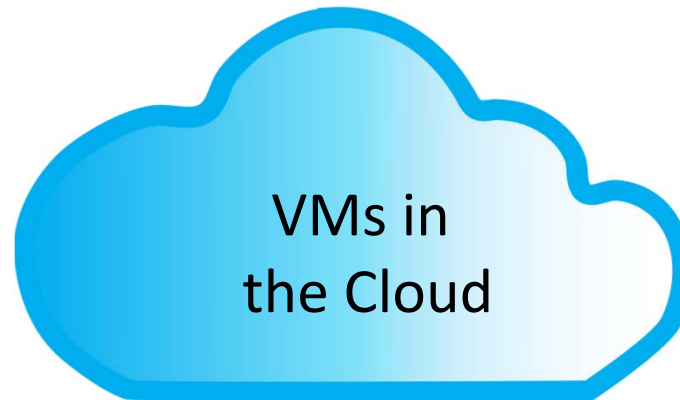
Intel Edison



Raspberry Pi3



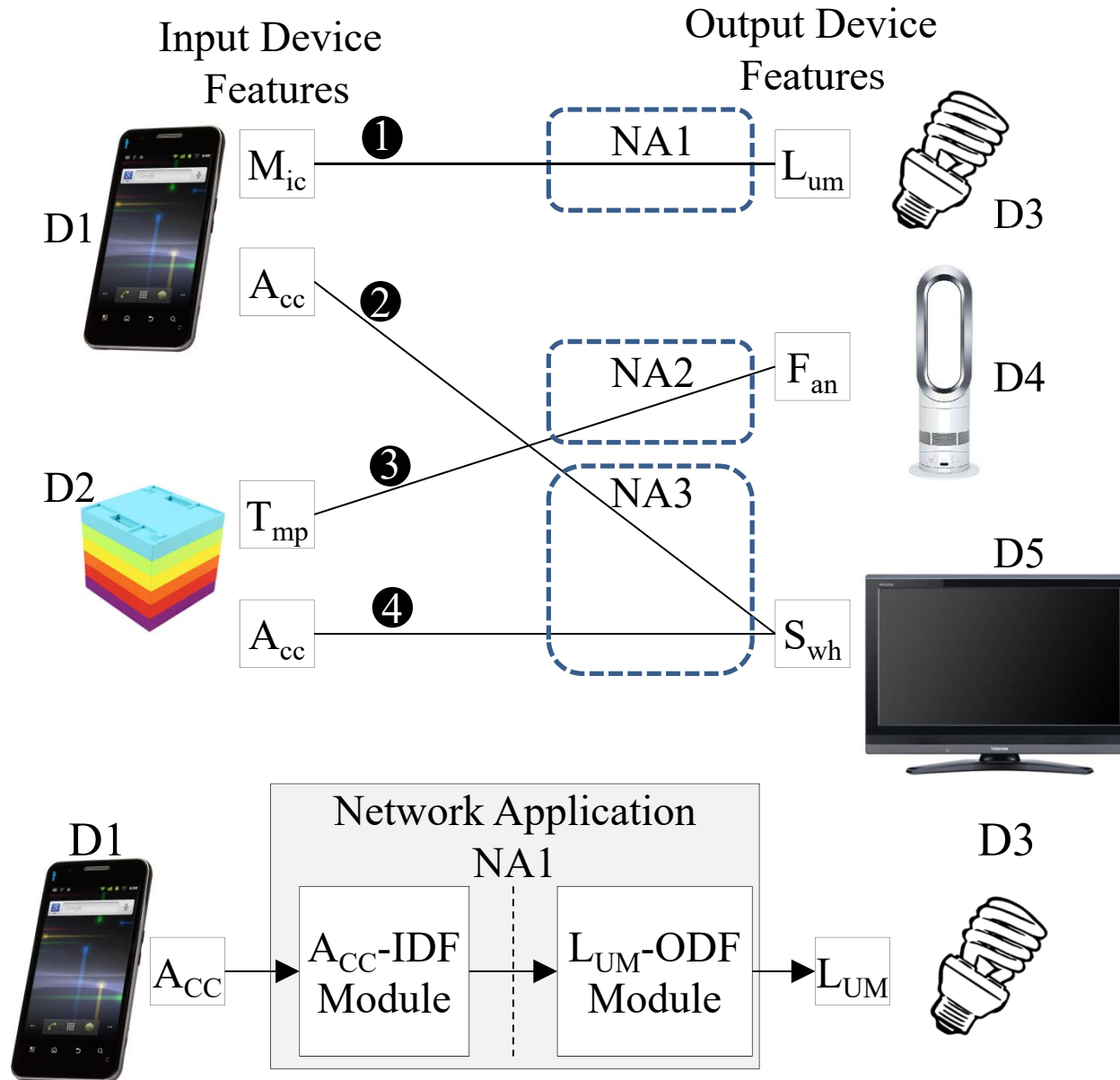
PC server



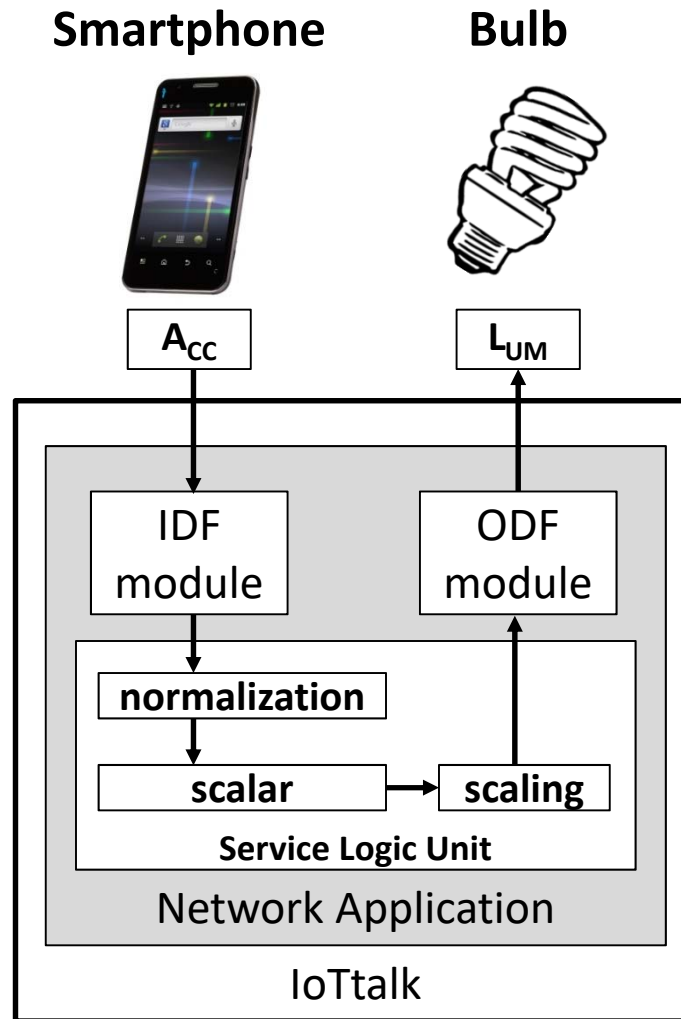
For example, you can try

<http://140.113.199.200:7788/connection>

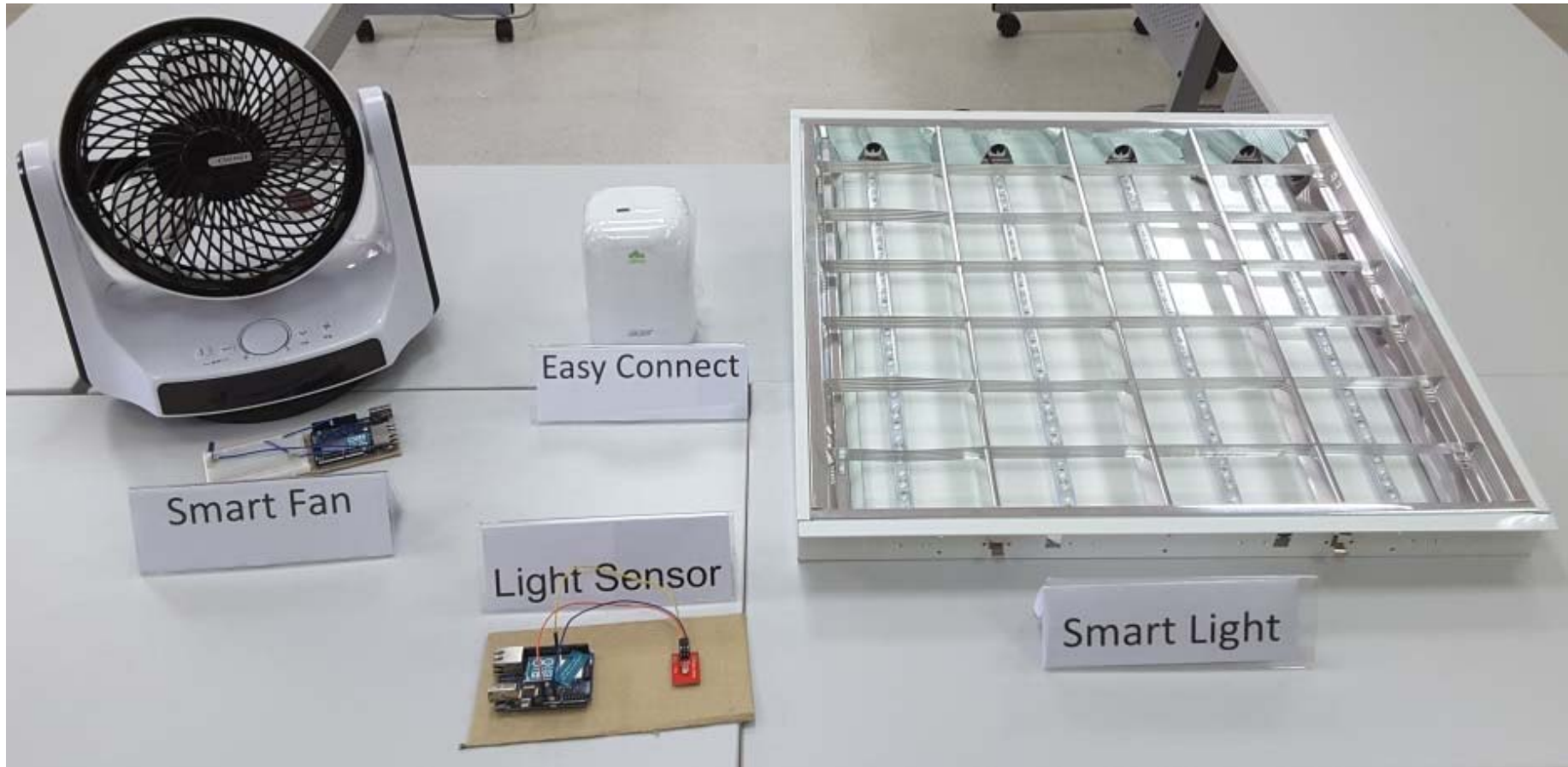
# Appliance Connections with IoTtalk



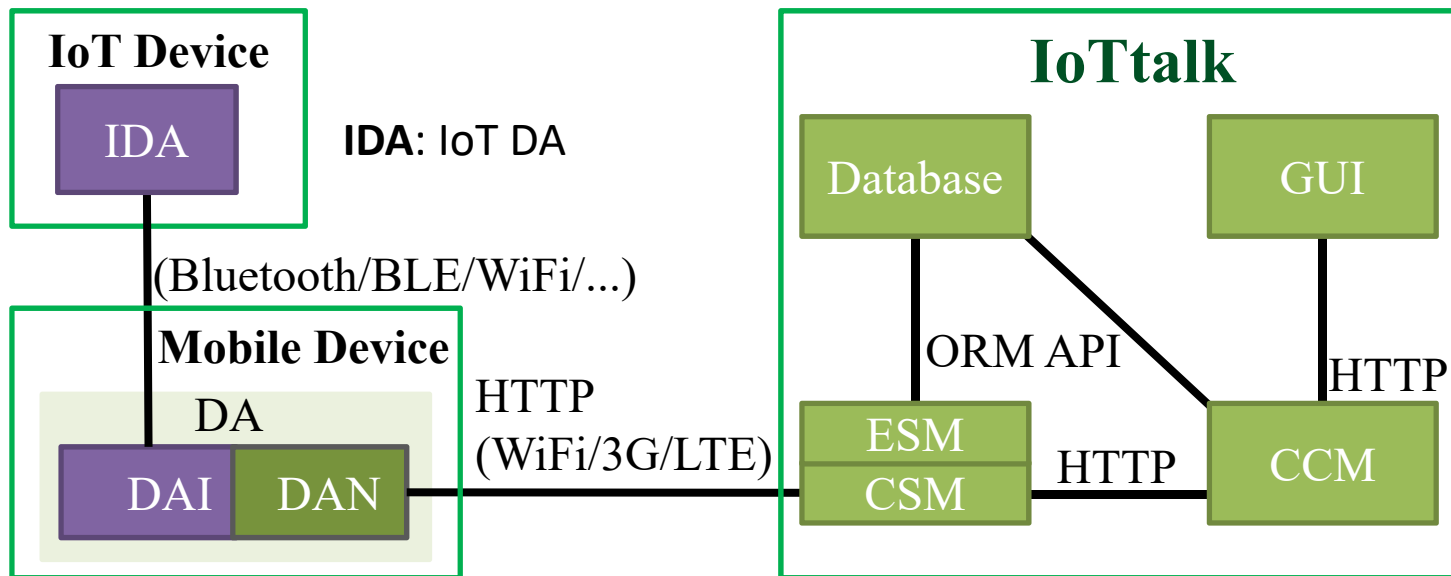
# Connection and Mapping Manner



# Connect Appliances to IoTtalk



# System Architecture



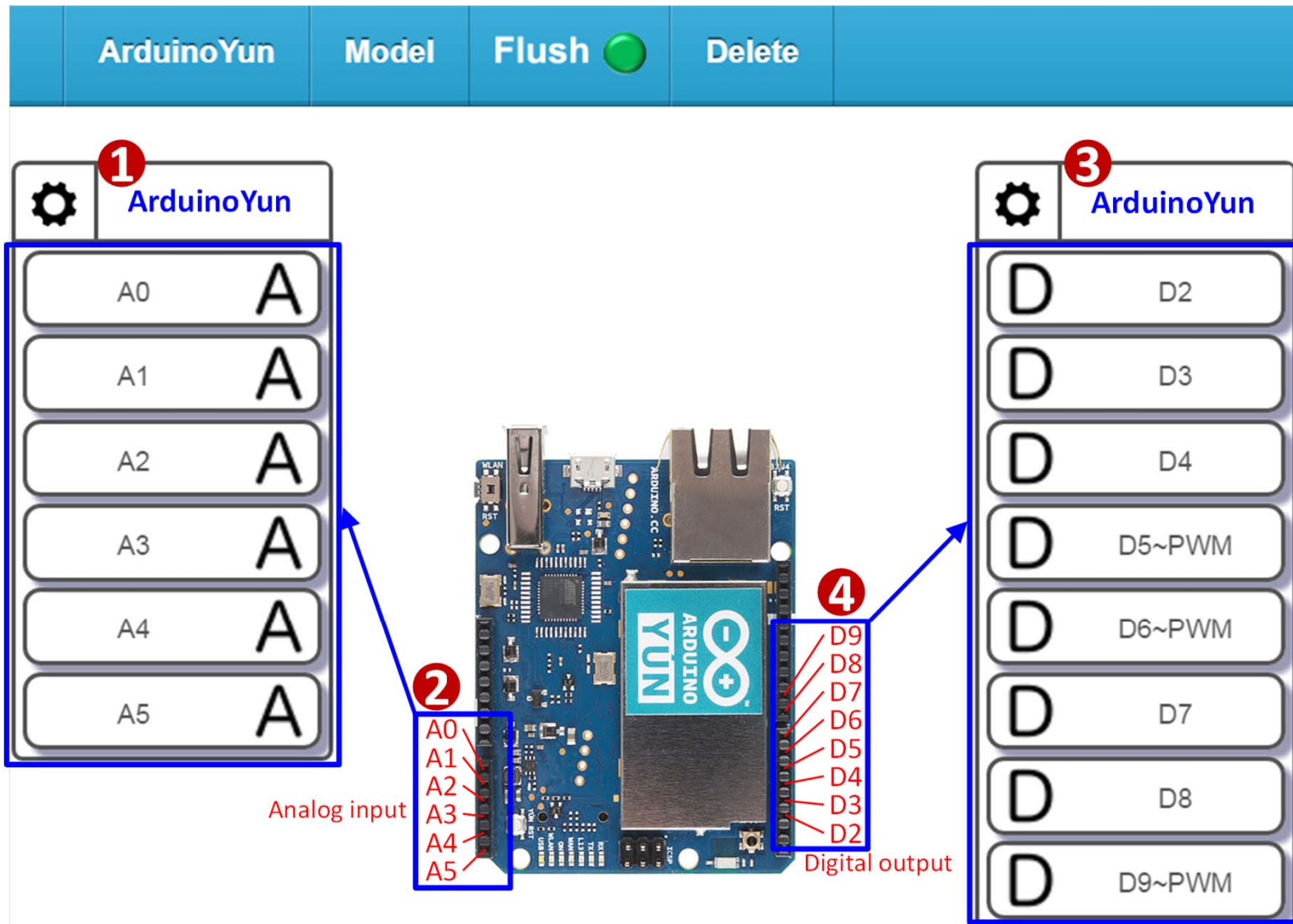
**DA:** device application

**DAN:** DA to Network

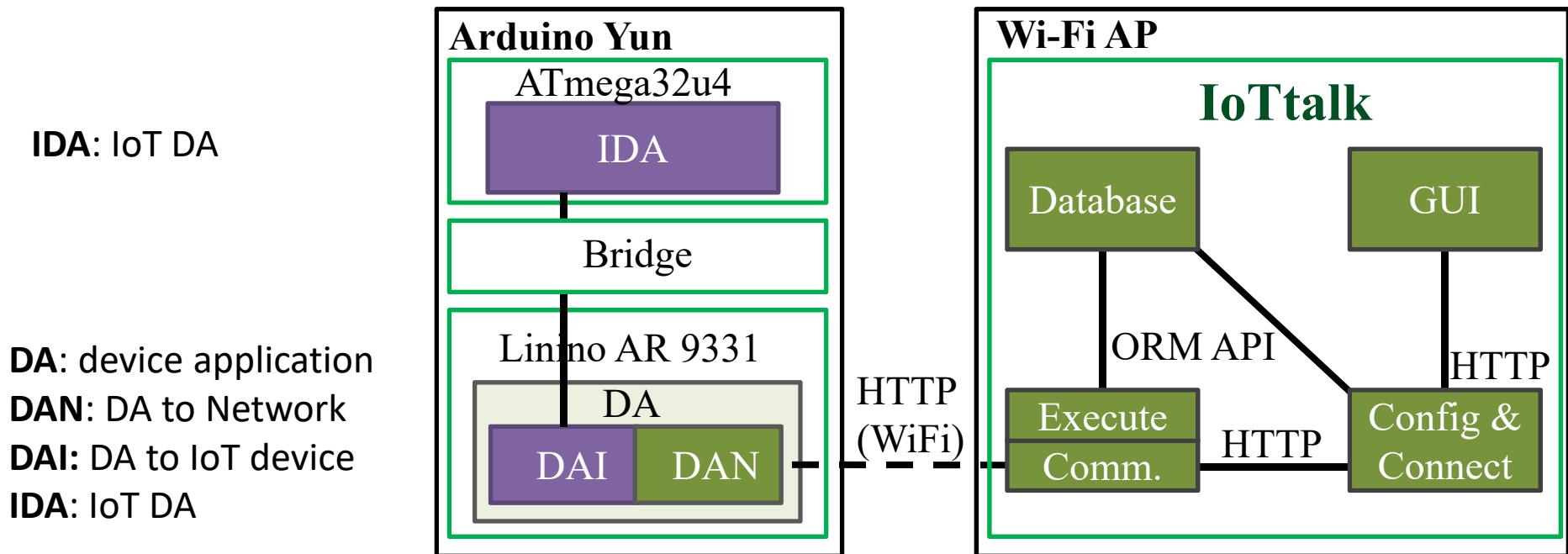
**DAI:** DA to IoT device

**IDA:** IoT DA

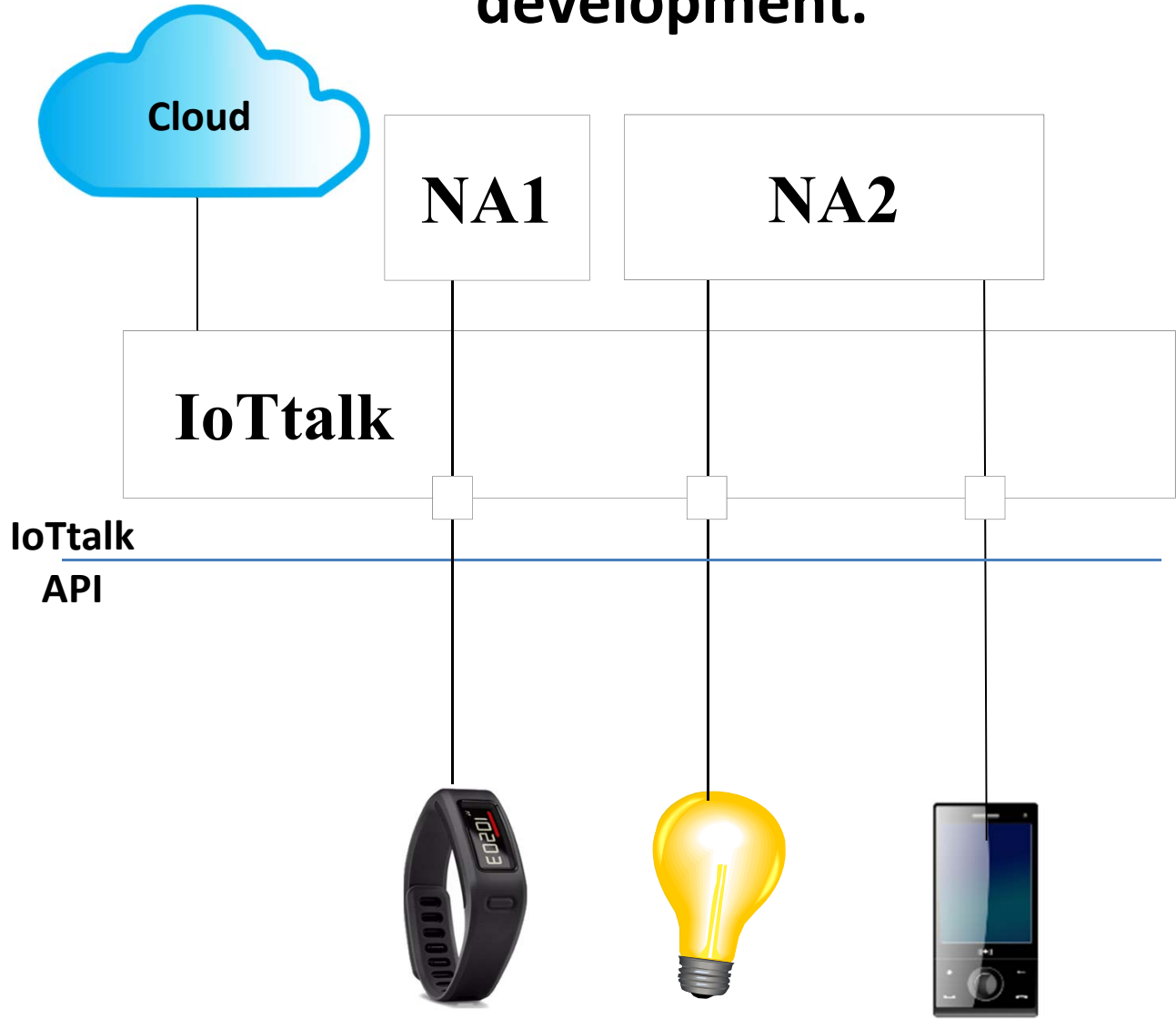
# ArduTalk - IoTtalk and Arduino Yun

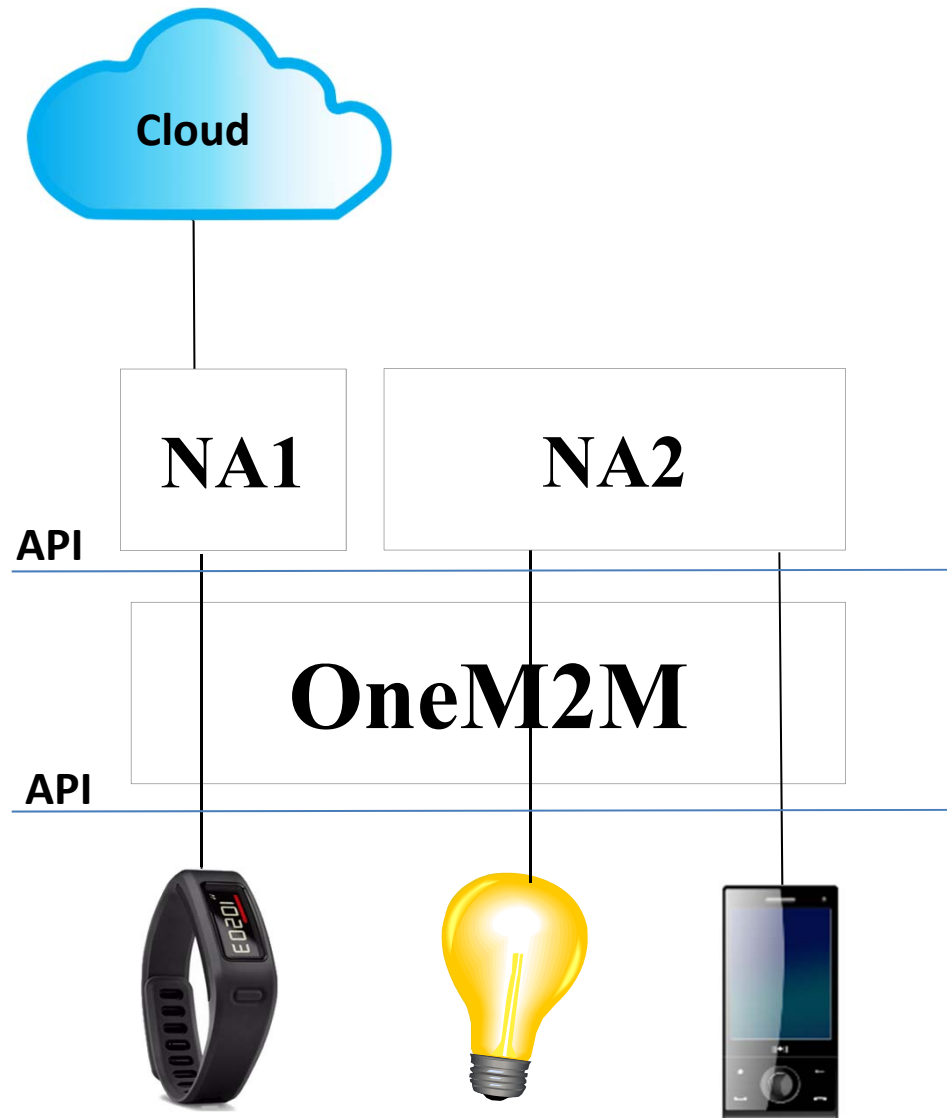


# ArduTalk - IoTtalk and Arduino Yun

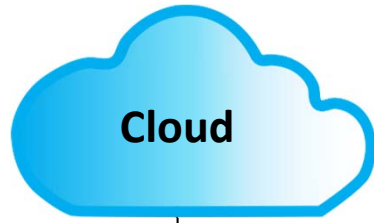


**IoTtalk is a network application platform to simplify the network application development.**

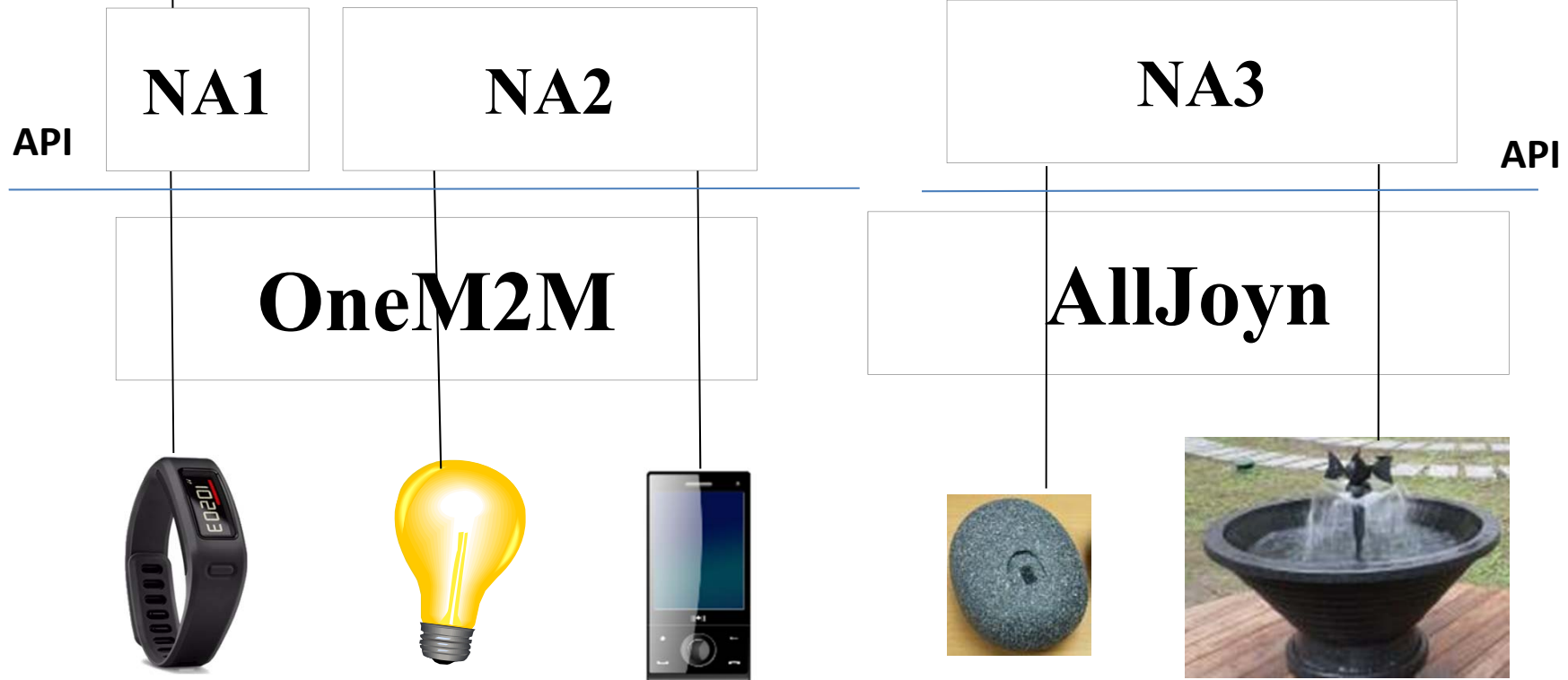


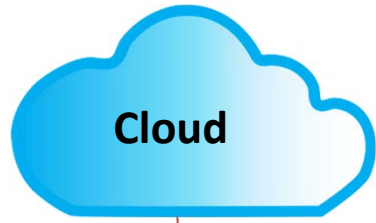


**If the device is developed under oneM2M, openMTC, or AllJoyn, these IoT platforms will provide APIs for the user to develop network application for the device.**

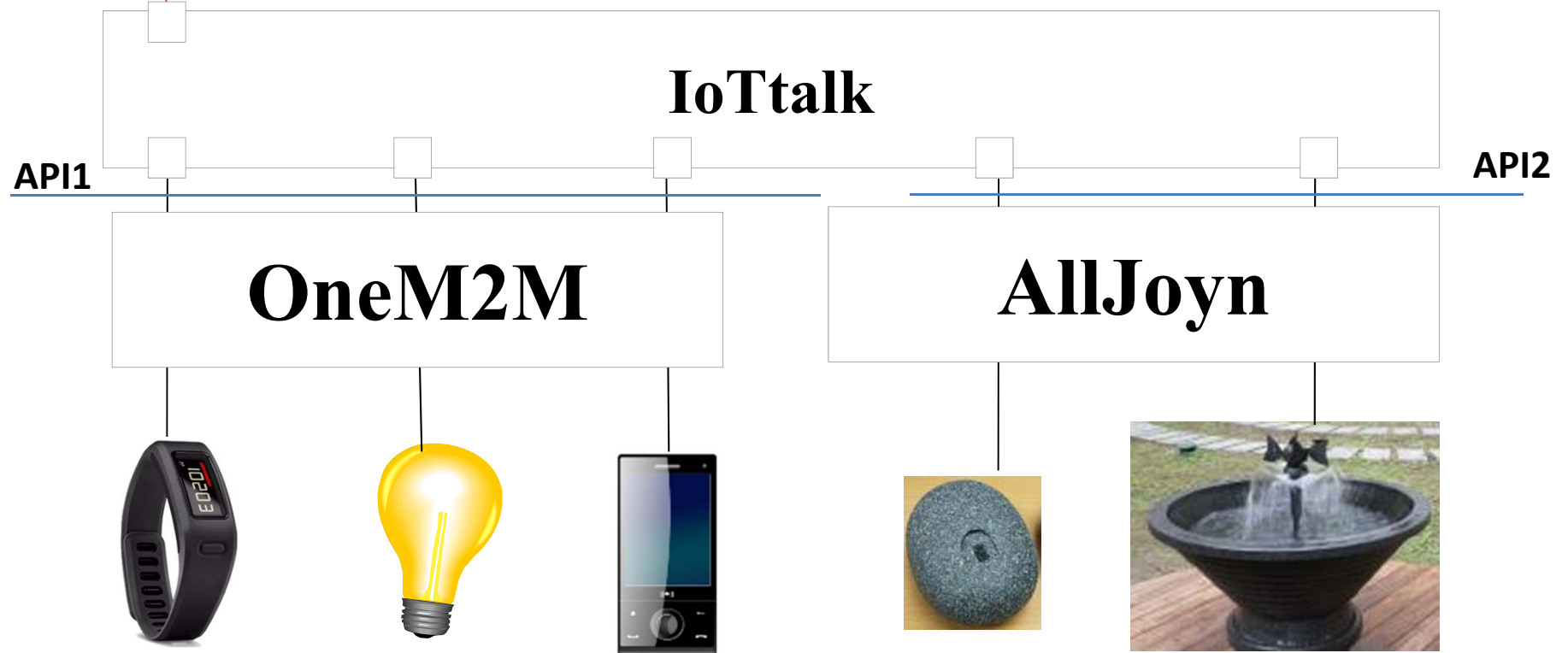


**These standards are difficult to interoperate, and the devices cannot easily communicate**



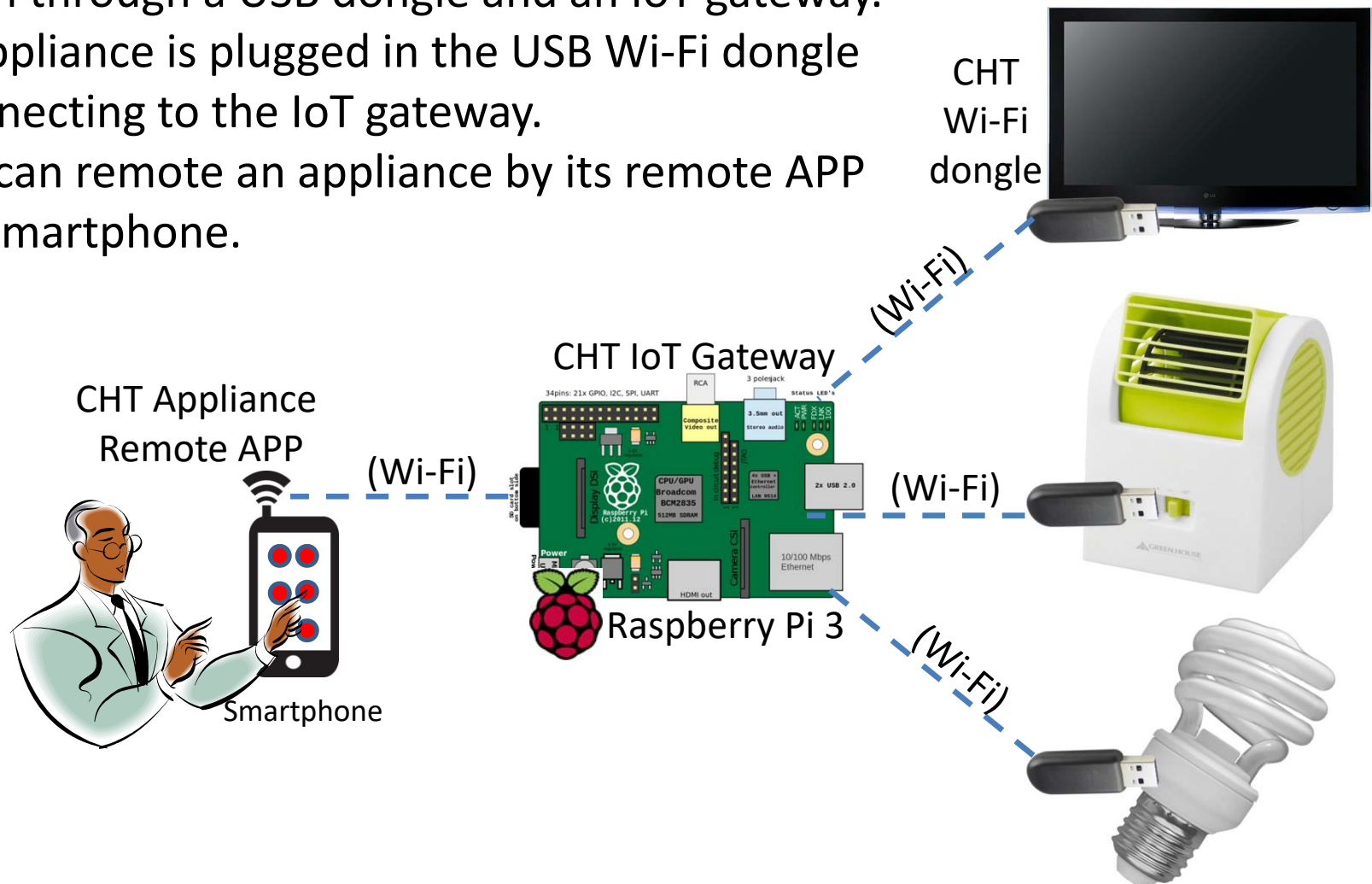


**IoTtalk is a network application platform built on top of the above IoT systems to simplify the network application development.**

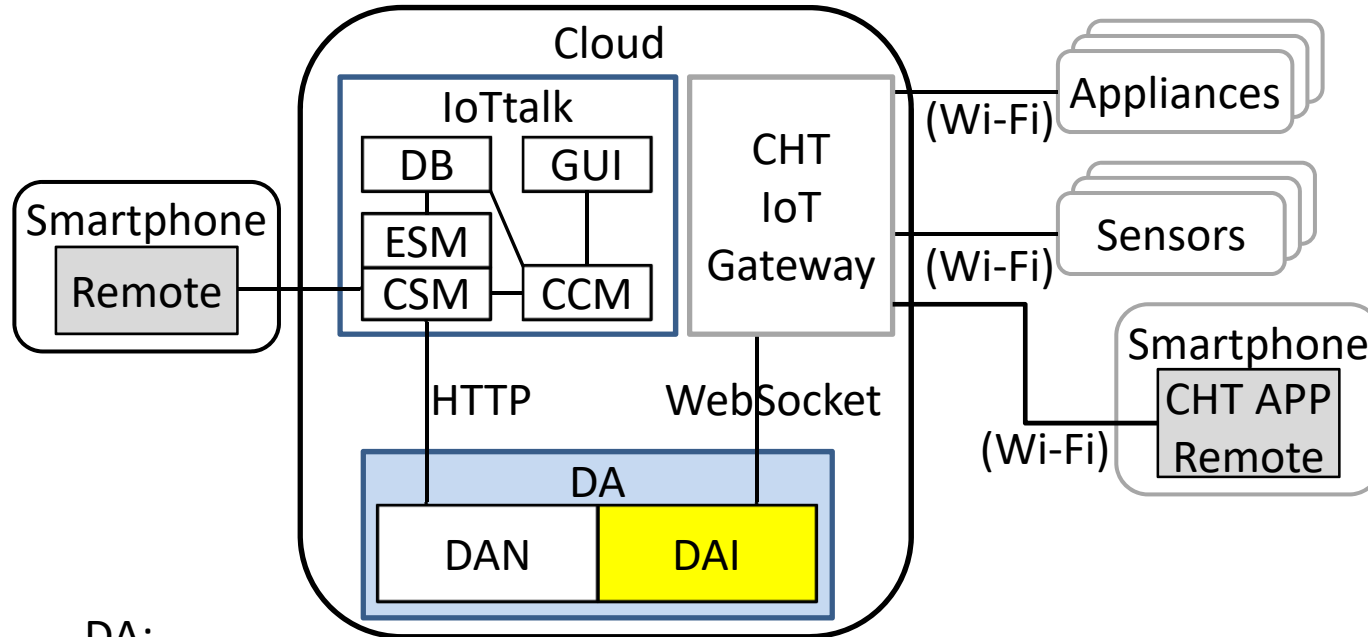


# The Smart Home Solution by CHT

- Chunghwa Telecom (CHT) provides a smart home solution through a USB dongle and an IoT gateway.
- Each appliance is plugged in the USB Wi-Fi dongle for connecting to the IoT gateway.
- A user can remote an appliance by its remote APP in the smartphone.



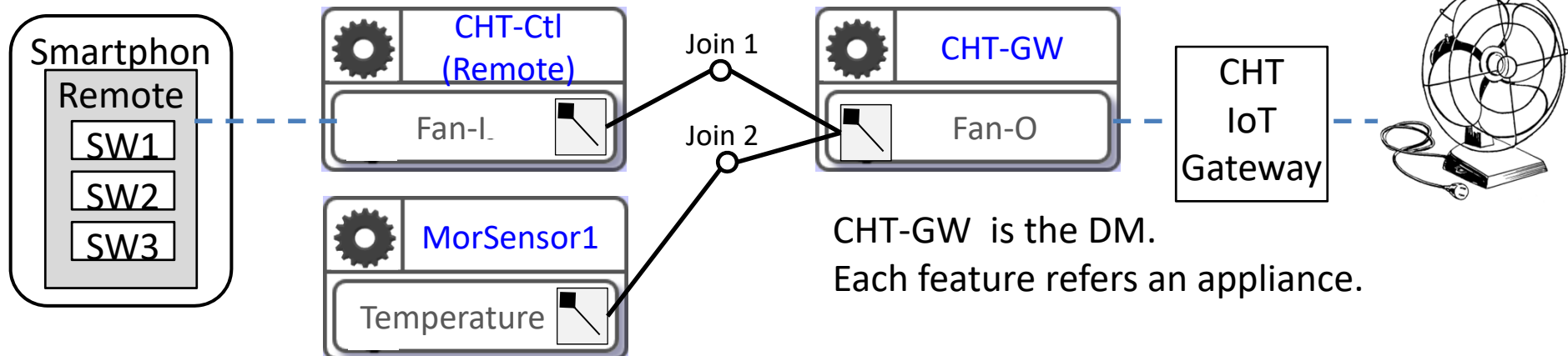
# CHT Smart Home with IoTtalk



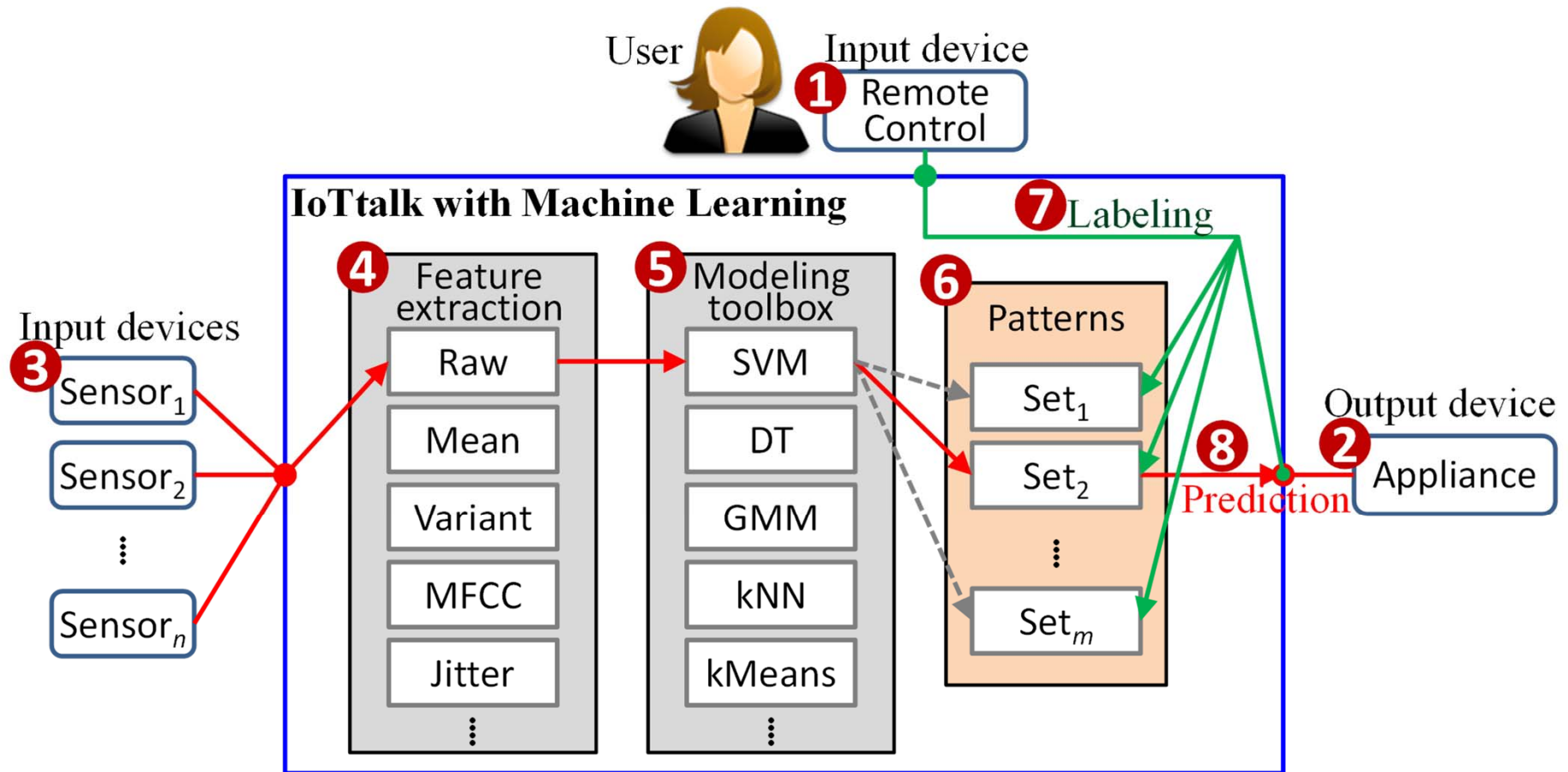
DA:

1. Register/Deregister for appliances/sensors according to **notifications**
2. Pull **commands** from the IoTtalk server to CHT home server
3. Receive **sensor data** then *Push* them to the IoTtalk server

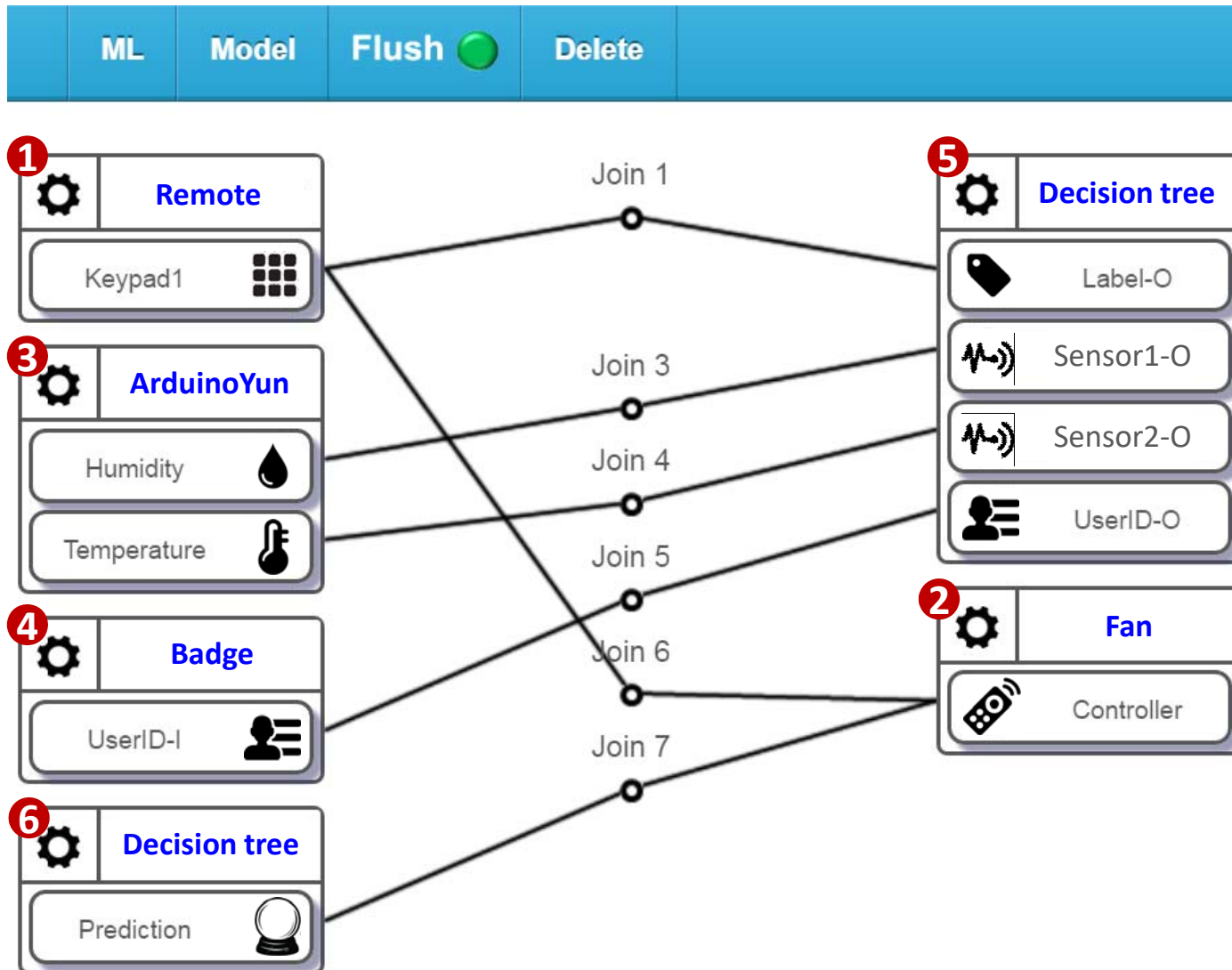
For example:



# IoTtalk with Machine Learning



# Smart Fan by using Decision Tree Algorithm



# Conclusion

- IoTtalk is an IoT device management tool
- Easy to deploy and operate
- Easy to develop applications - simply connect IDFs and ODFs
- Transparently observe the connections between IoT devices
- Monitor the transmitting values between IoT devices
- Application debugging is more easier