Connectivity of Data Foundry

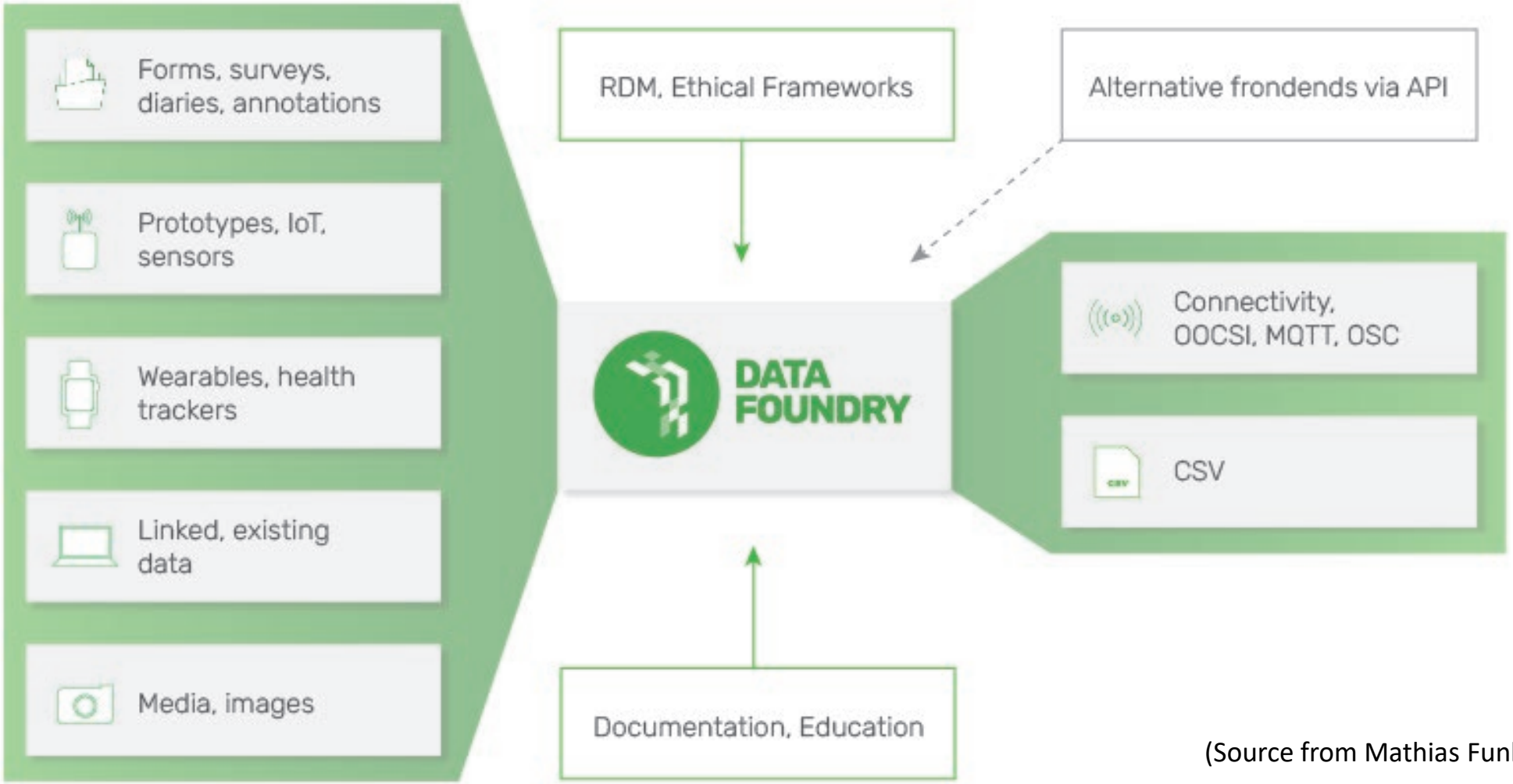
Workshop – IoT dataset

I-Tang (Eden) Chiang
i.chiang@tue.nl
d.Search Lab @ Atlas 2.125

Outline

- About Data Foundry
- Why using Data Foundry
- Practices with IoT dataset and ESPs -
  - Data collection by Data Foundry through OOCSI
  - Data collection by Data Foundry via API (next workshop)

About Data Foundry



(Source from Mathias Funk)

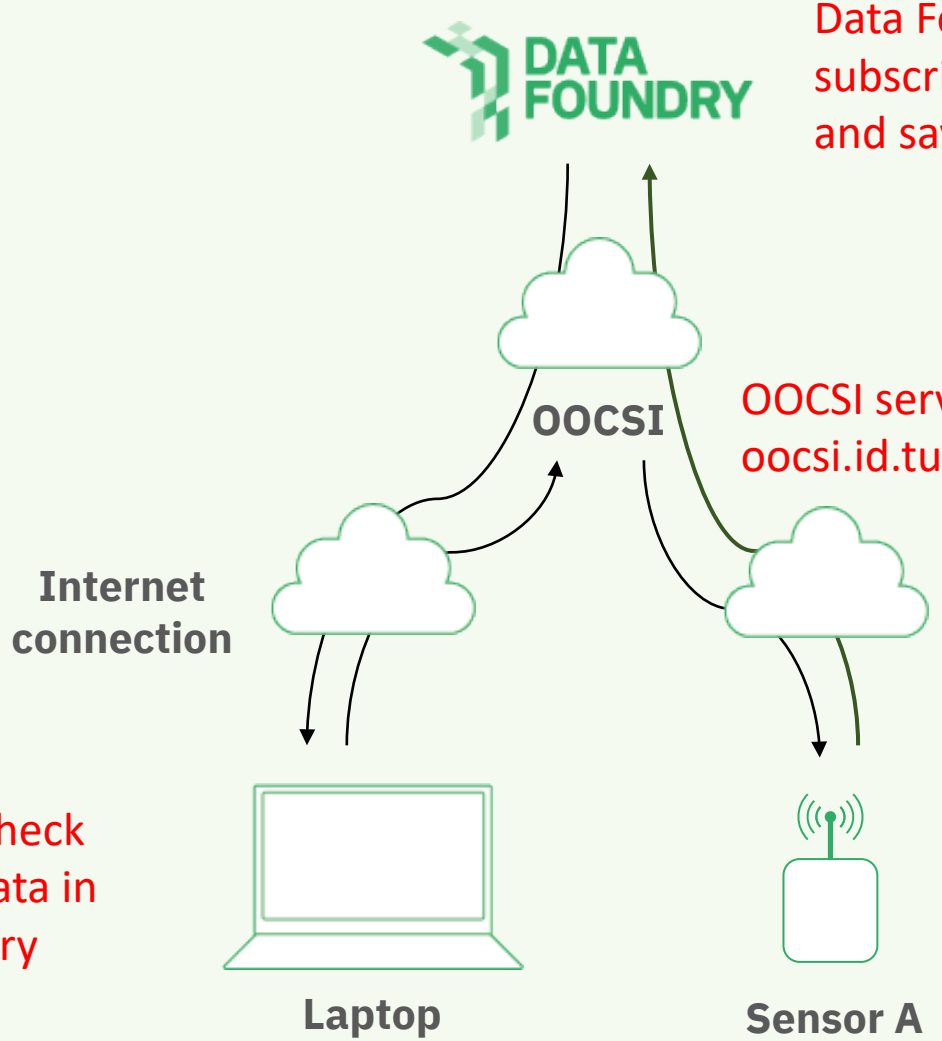
Why using Data Foundry

- Safety
- Simplicity
- Community
- Well organized “Projects” structure
- Offers support for various courses such as:
  - Making Sense of Sensors (Bachelor)
  - Digital Craftsmanship (Bachelor)
  - Data-Enabled Design (Master)
  - Creativity and Aesthetics of data & AI (Master)

Practice: Data collection by Data Foundry through OOCSI

1. Preset of Data Foundry
    1. Create a project → Should be done by Data Foundry automatically
    2. Create an IoT dataset
    3. Create a device
2. Connect IoT device (ESP32) to iotroam network
3. Send data through OOCSI
4. Check data in OOCSI UI Client page
5. Save data to own IoT dataset

How Data Foundry and OOCSI work together



Data Foundry obtains data by subscribing to OOCSI channel and save it

OOCSI server domain name:
oocsi.id.tue.nl

IoT network in campus:
SSID: iotroam
Password: set in iotroam.org

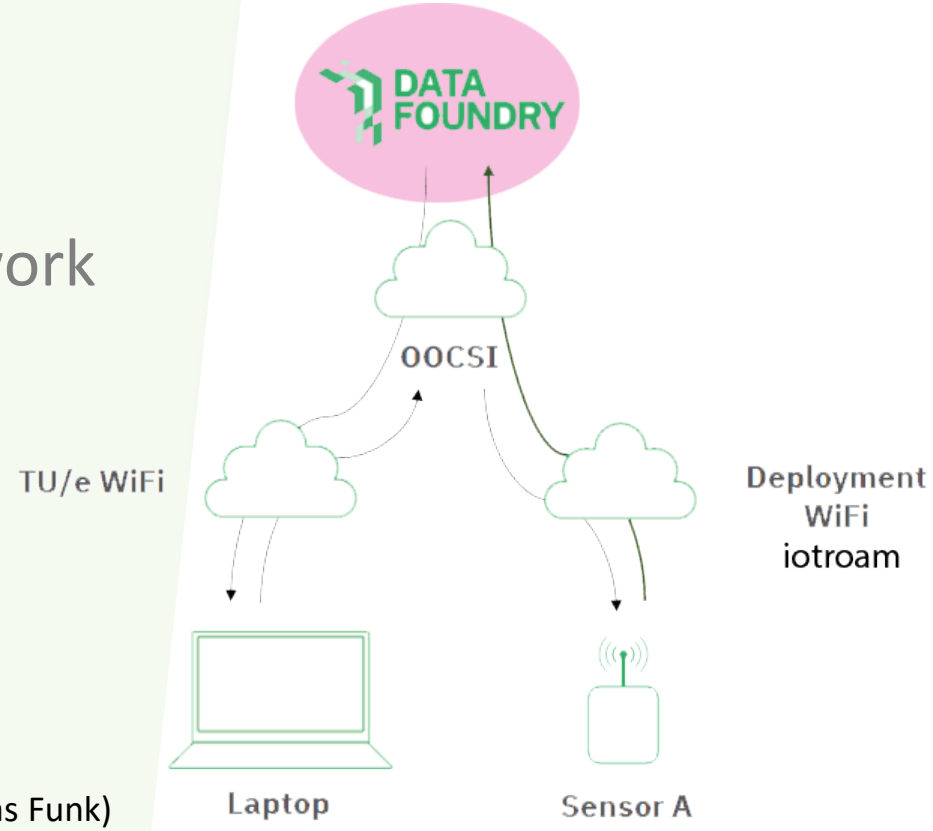
Users can check real-time data in Data Foundry

Micro controller
Reference ID in Data Foundry

(Source from Mathias Funk)

Practice: Data collection by Data Foundry through OOCSI

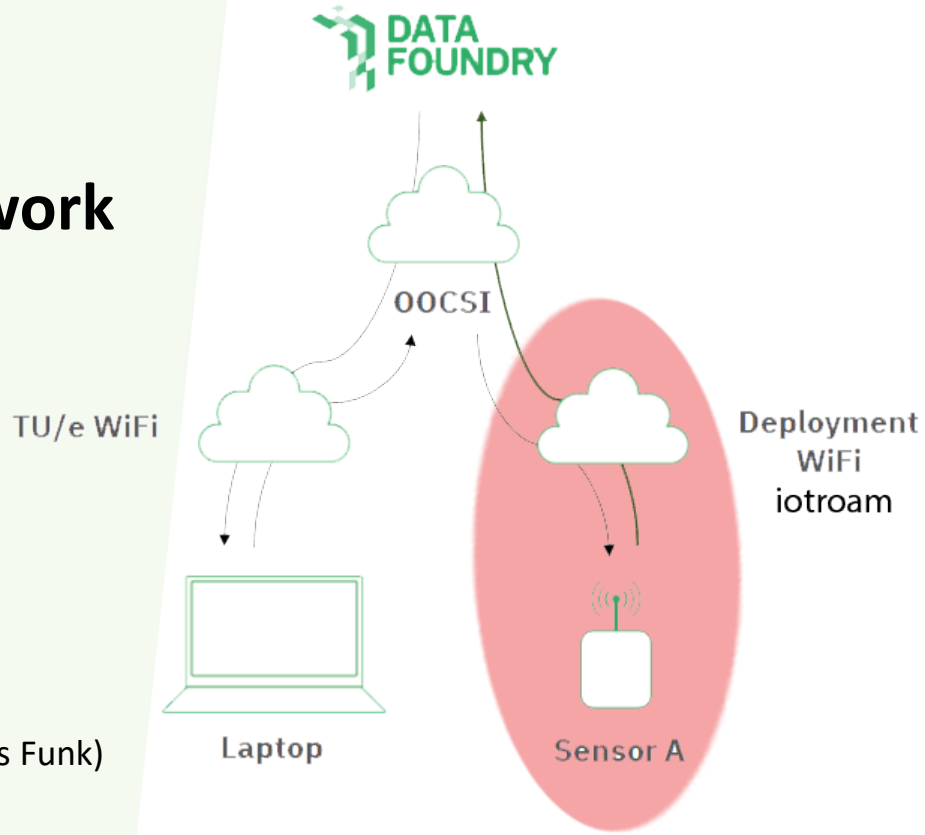
1. **Preset of Data Foundry**
    1. **Create a project → Should be done by Data Foundry automatically**
    2. **Create an IoT dataset**
    3. **Create a device**
2. Connect IoT device (ESP32) to iotroam network
3. Send data through OOCSI
4. Check data in OOCSI UI Client page
5. Save data to own IoT dataset



(Source from Mathias Funk)

Practice: Data collection by Data Foundry through OOCSI

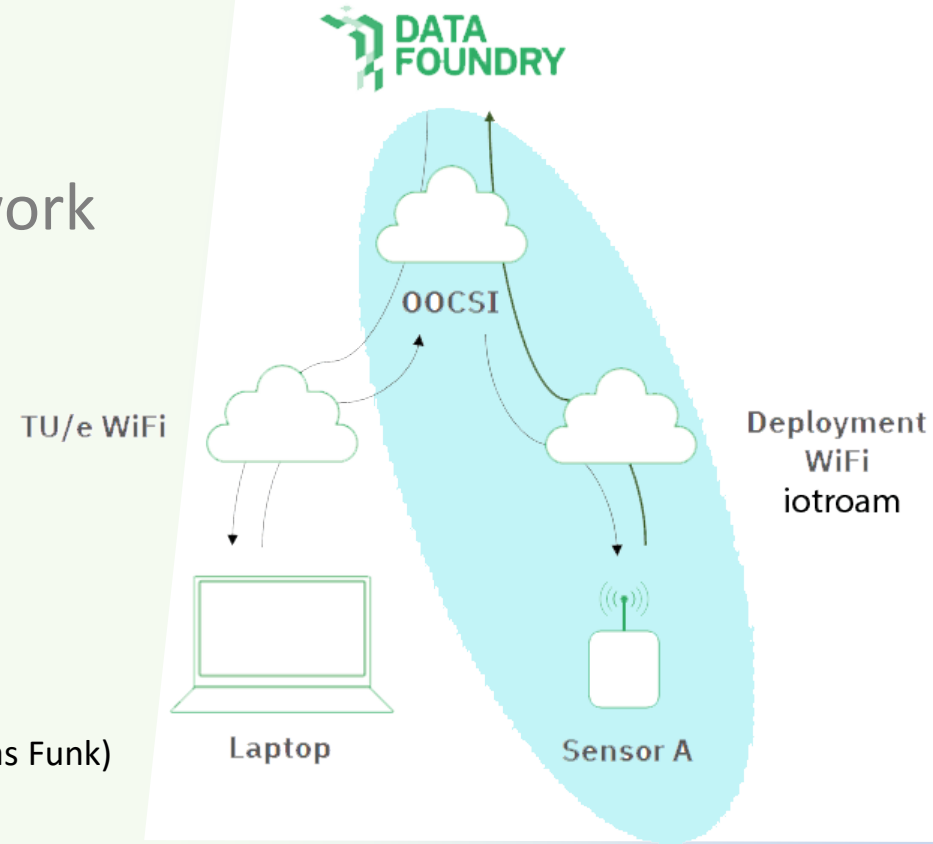
1. Preset of Data Foundry
    1. Create a project → Should be done by Data Foundry automatically
    2. Create an IoT dataset
    3. Create a device
2. **Connect IoT device (ESP32) to iotroam network**
3. Send data through OOCSI
4. Check data in OOCSI UI Client page
5. Save data to own IoT dataset



(Source from Mathias Funk)

Practice: Data collection by Data Foundry through OOCSI

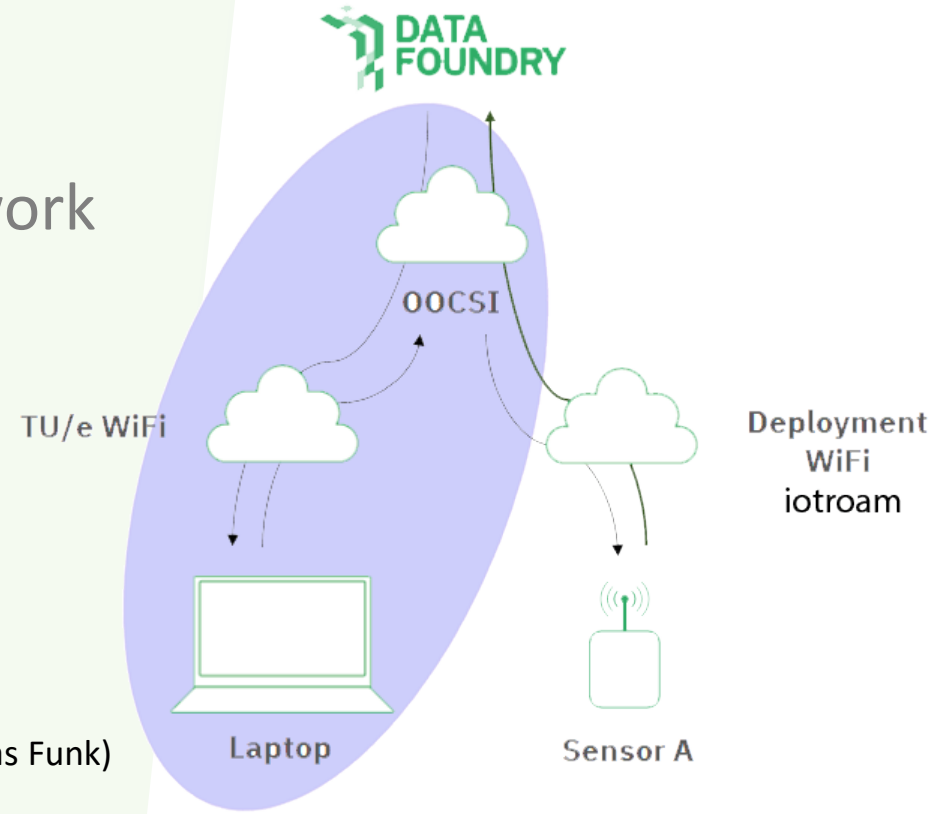
1. Preset of Data Foundry
    1. Create a project → Should be done by Data Foundry automatically
    2. Create an IoT dataset
    3. Create a device
2. Connect IoT device (ESP32) to iotroam network
3. **Send data through OOCSI**
4. Check data in OOCSI UI Client page
5. Save data to own IoT dataset



(Source from Mathias Funk)

Practice: Data collection by Data Foundry through OOCSI

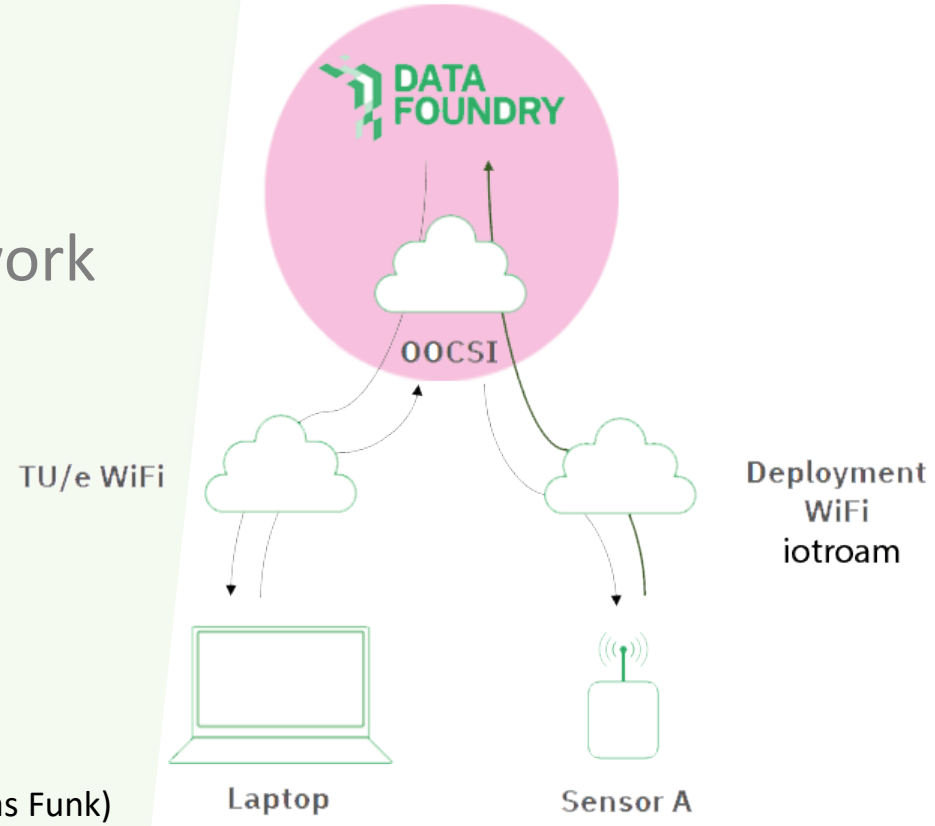
1. Preset of Data Foundry
    1. Create a project → Should be done by Data Foundry automatically
    2. Create an IoT dataset
    3. Create a device
2. Connect IoT device (ESP32) to iotroam network
3. Send data through OOCSI
4. **Check data in OOCSI UI Client page**
5. Save data to own IoT dataset



(Source from Mathias Funk)

Practice: Data collection by Data Foundry through OOCSI

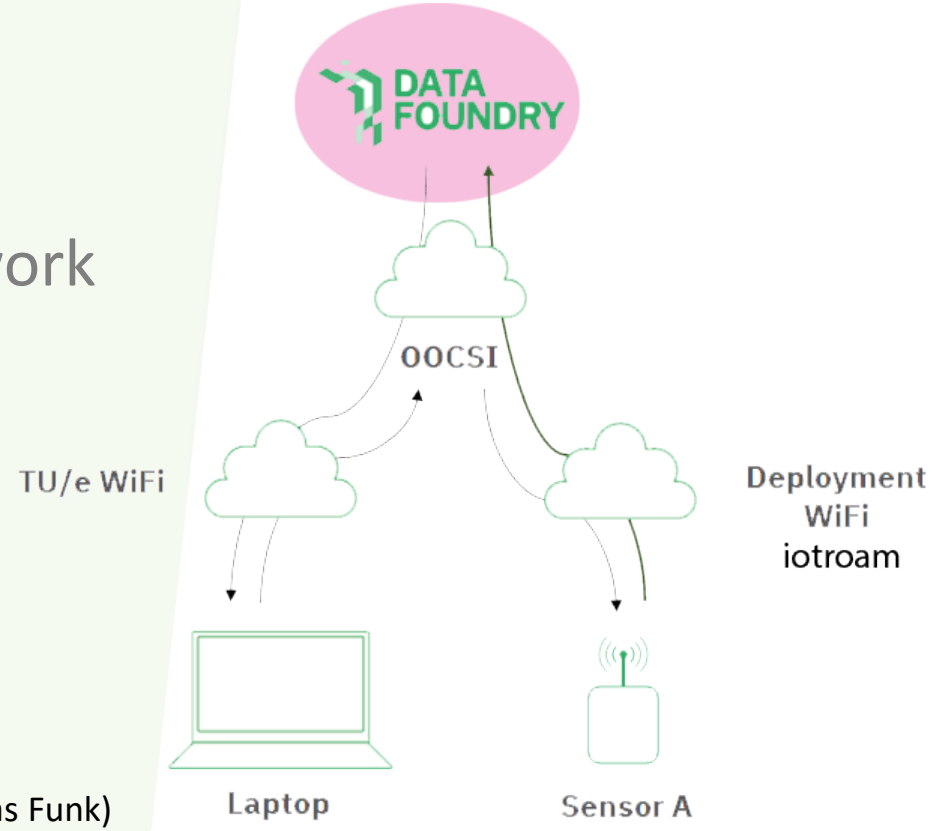
1. Preset of Data Foundry
    1. Create a project → Should be done by Data Foundry automatically
    2. Create an IoT dataset
    3. Create a device
2. Connect IoT device (ESP32) to iotroam network
3. Send data through OOCSI
4. Check data in OOCSI UI Client page
5. **Save data to own IoT dataset via OOCSI**



(Source from Mathias Funk)

Practice: Data collection by Data Foundry through OOCSI

1. **Preset of Data Foundry**
    1. **Create a project → Should be done by Data Foundry automatically**
    2. **Create an IoT dataset**
    3. **Create a device**
2. Connect IoT device (ESP32) to iotroam network
3. Send data through OOCSI
4. Check data in OOCSI UI Client page
5. Save data to own IoT dataset

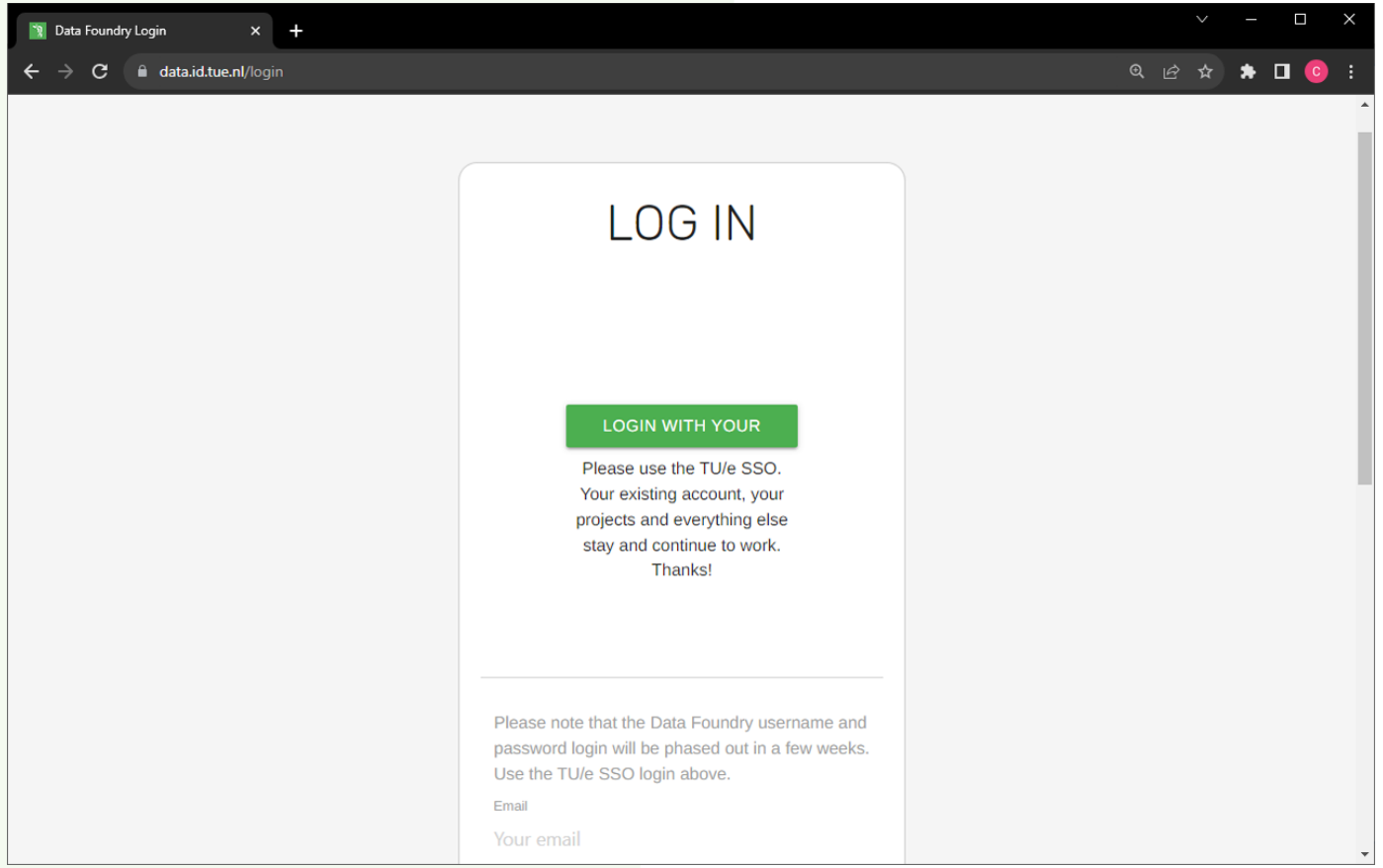


(Source from Mathias Funk)

Step 1.1 - CREATING YOUR FIRST PROJECT (1/3)

Log in with TU/e account

1. Browse to https://data.id.tue.nl
2. Click on “LOGIN” button at the left side
3. Click on “LOGIN WITH YOUR TU/E ACCOUNT” button
4. Finish the authorization process (to login with TU/e account)
5. You are in!

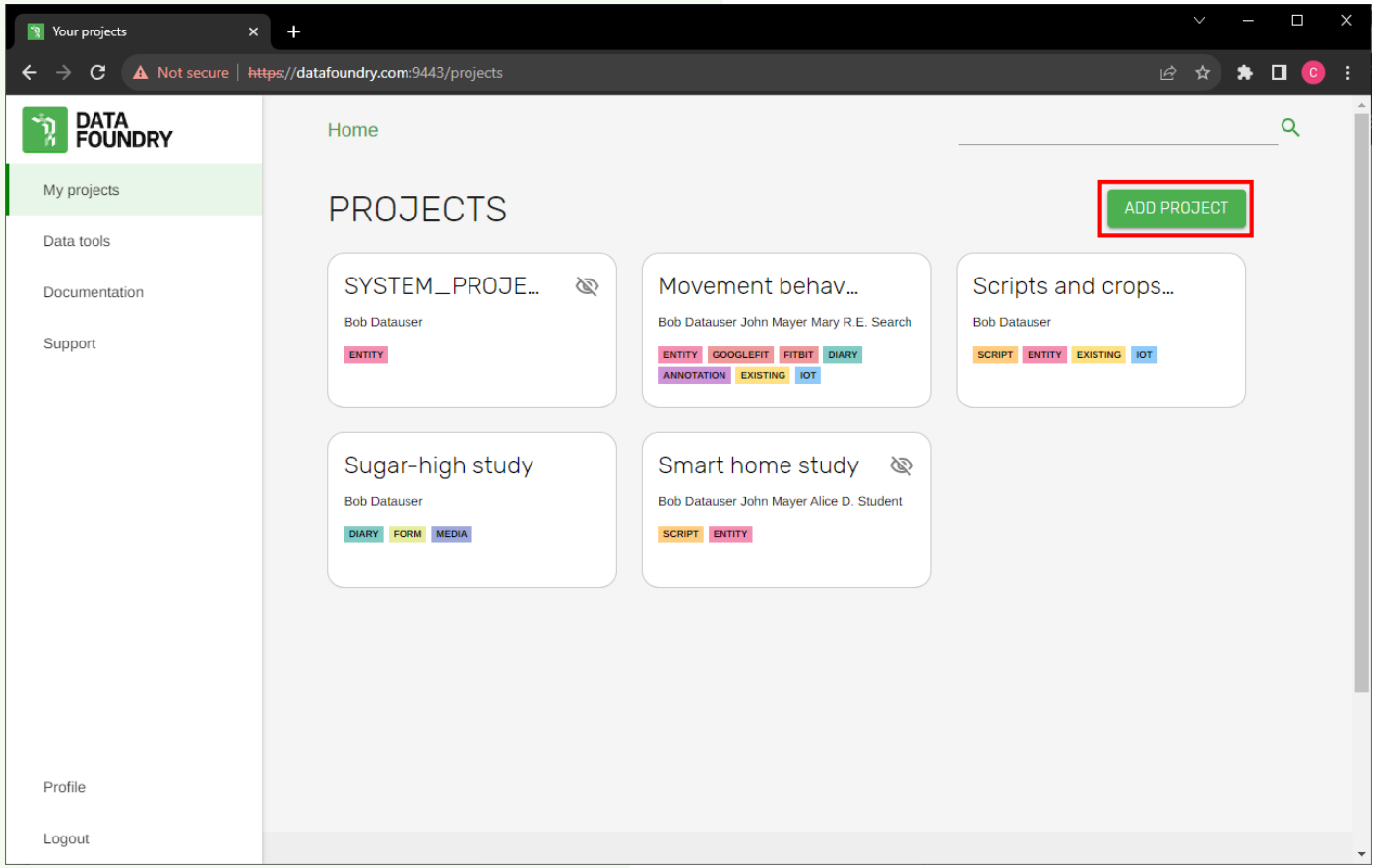


(Source: https://data.id.tue.nl)

Step 1.1 - CREATING YOUR FIRST PROJECT (2/3)

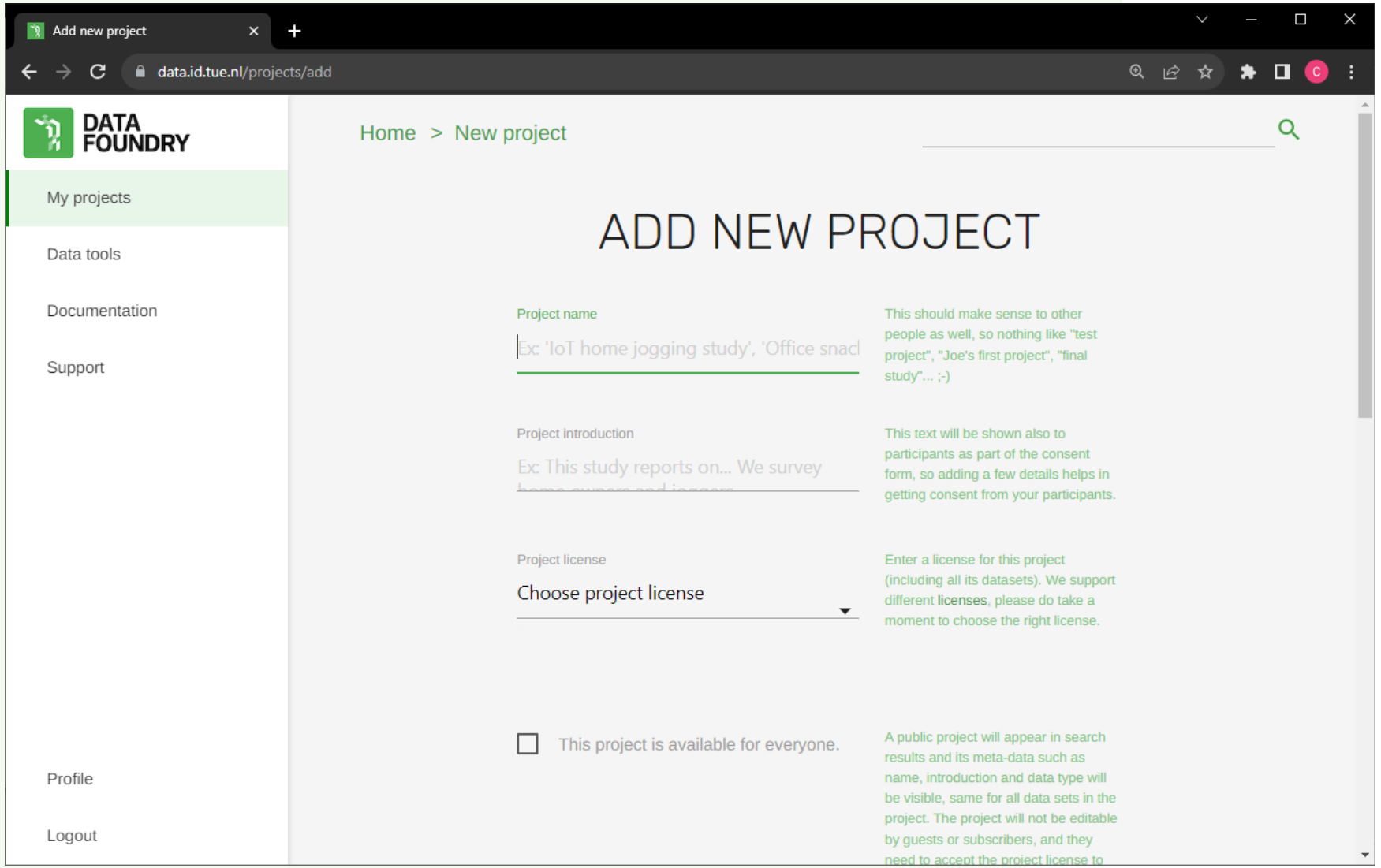
Create a project
(should be done automatically after first logging in)

1. Go to the home page
2. Click on add project on the right.
3. Fill in the details
4. Save it



(Source: https://data.id.tue.nl)

Step 1.1 - CREATING YOUR FIRST PROJECT (3/3)



(Source: https://data.id.tue.nl)

Step 1.2 - CREATING YOUR FIRST IOT DATASET (1/2)

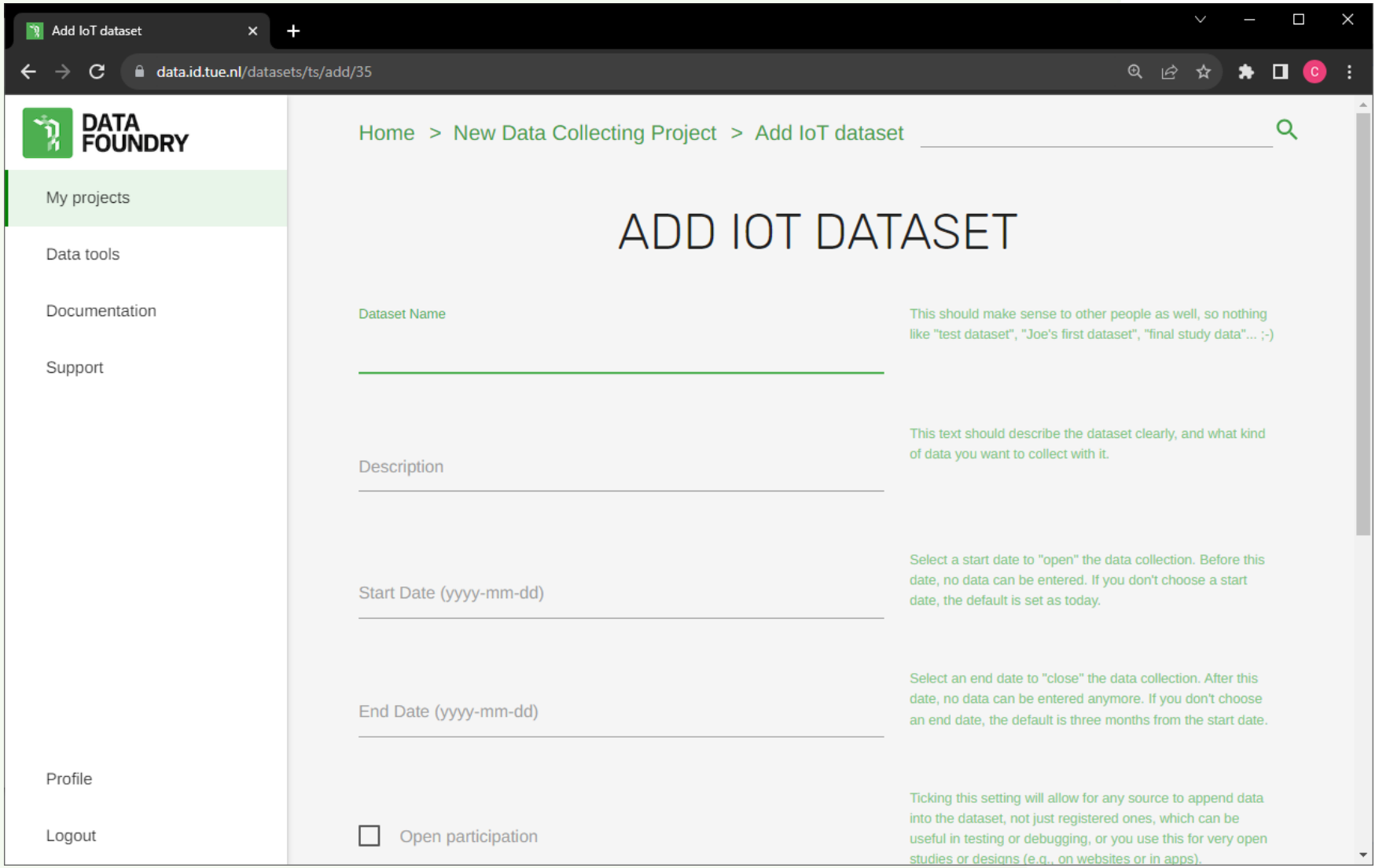
Create an IoT dataset within that project

1. Click on add dataset in the project page.
2. Select IoT dataset
3. Fill in the details
4. Save it

Important!

- The title of the dataset should make sense, and the text should describe the dataset clearly.
- The dataset only works in the period it is active.
- The default value for the start date is today and the end date is 3 months after the start date.

Step 1.2 - CREATING YOUR FIRST IOT DATASET (2/2)

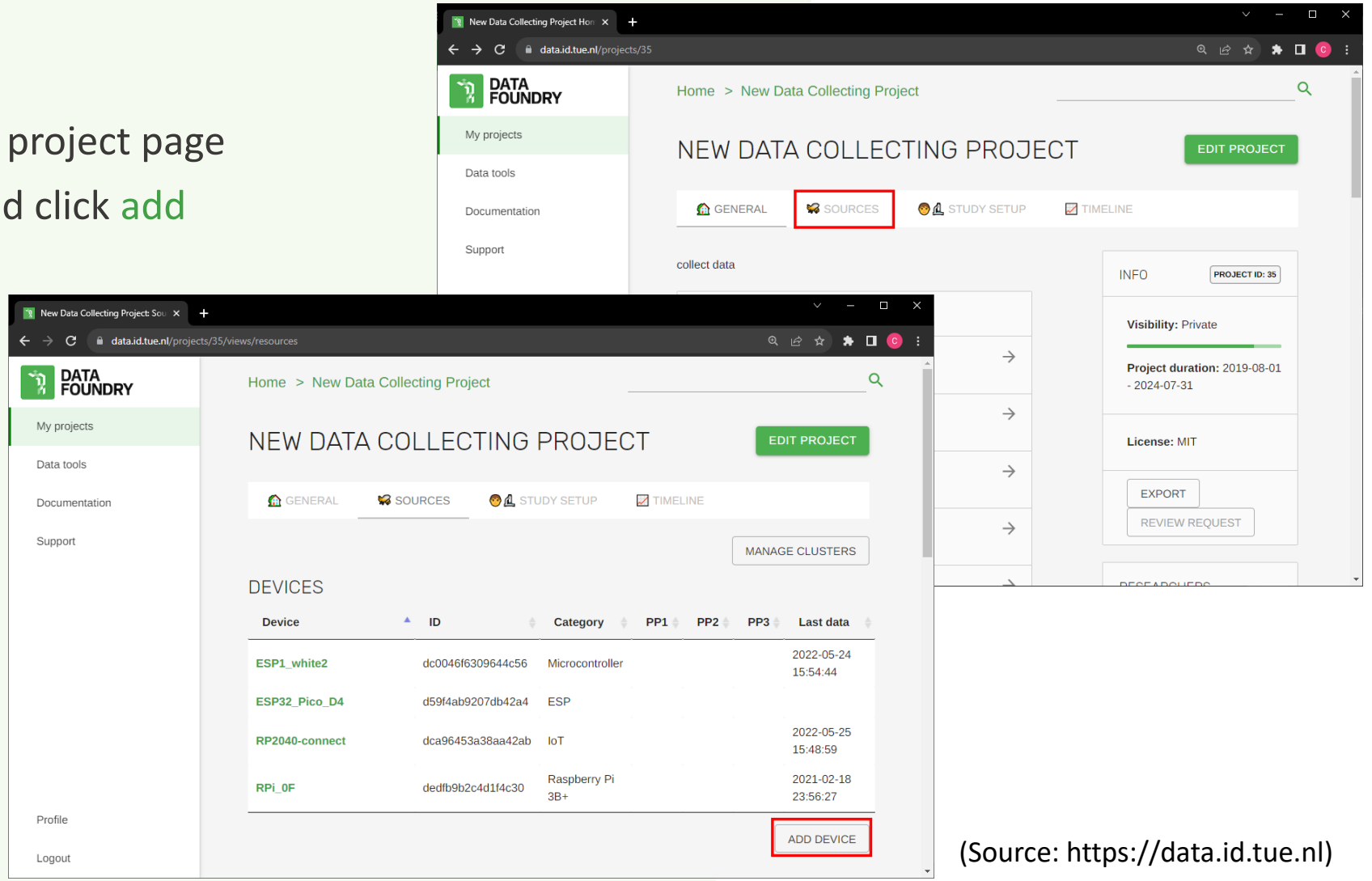


(Source: https://data.id.tue.nl)

Step 1.3 - CREATING YOUR FIRST DEVICE (1/2)

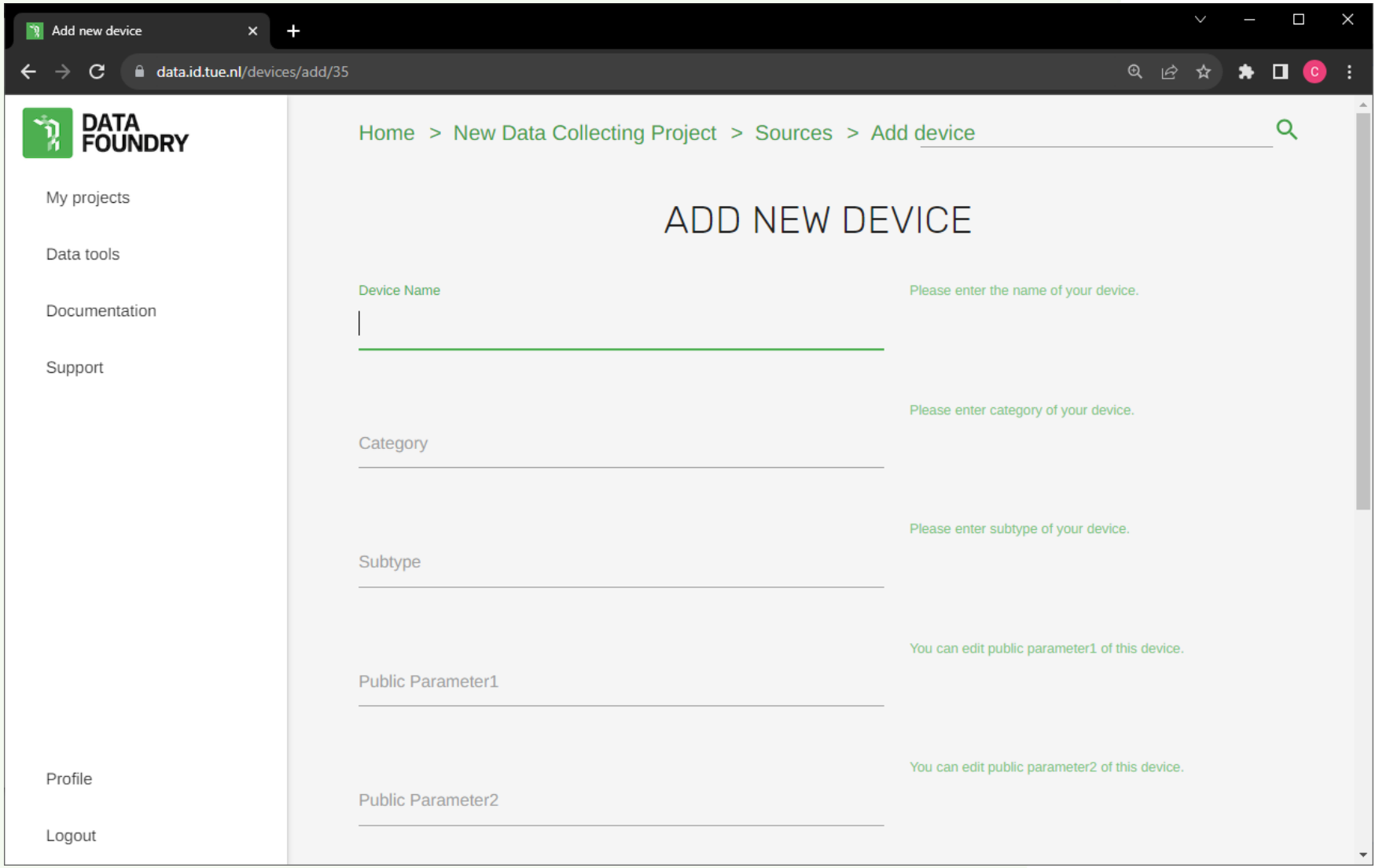
Create a new device

1. Click on SOURCES tab on project page
2. Find the devices table and click add device
3. Fill in the details
4. Save it



(Source: https://data.id.tue.nl)

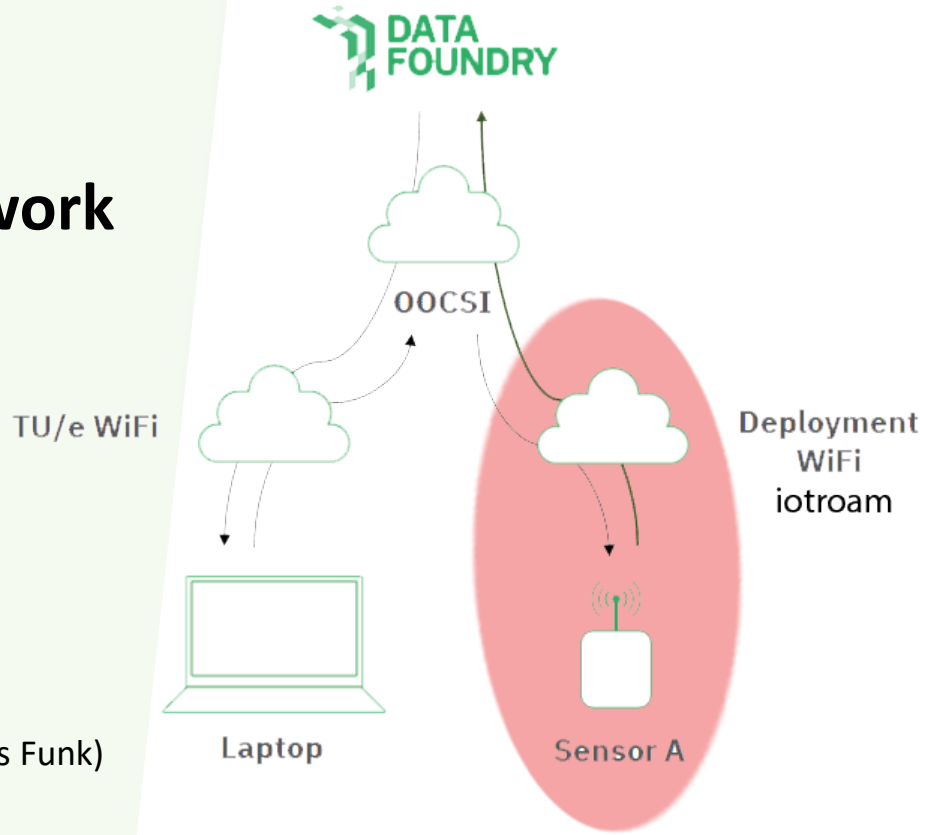
Step 1.3 - CREATING YOUR FIRST DEVICE (2/2)



(Source: https://data.id.tue.nl)

Practice: Data collection by Data Foundry through OOCSI

1. Preset of Data Foundry
    1. Create a project → Should be done by Data Foundry automatically
    2. Create an IoT dataset
    3. Create a device
2. **Connect IoT device (ESP32) to iotroam network**
3. Send data through OOCSI
4. Check data in OOCSI UI Client page
5. Save data to own IoT dataset



(Source from Mathias Funk)

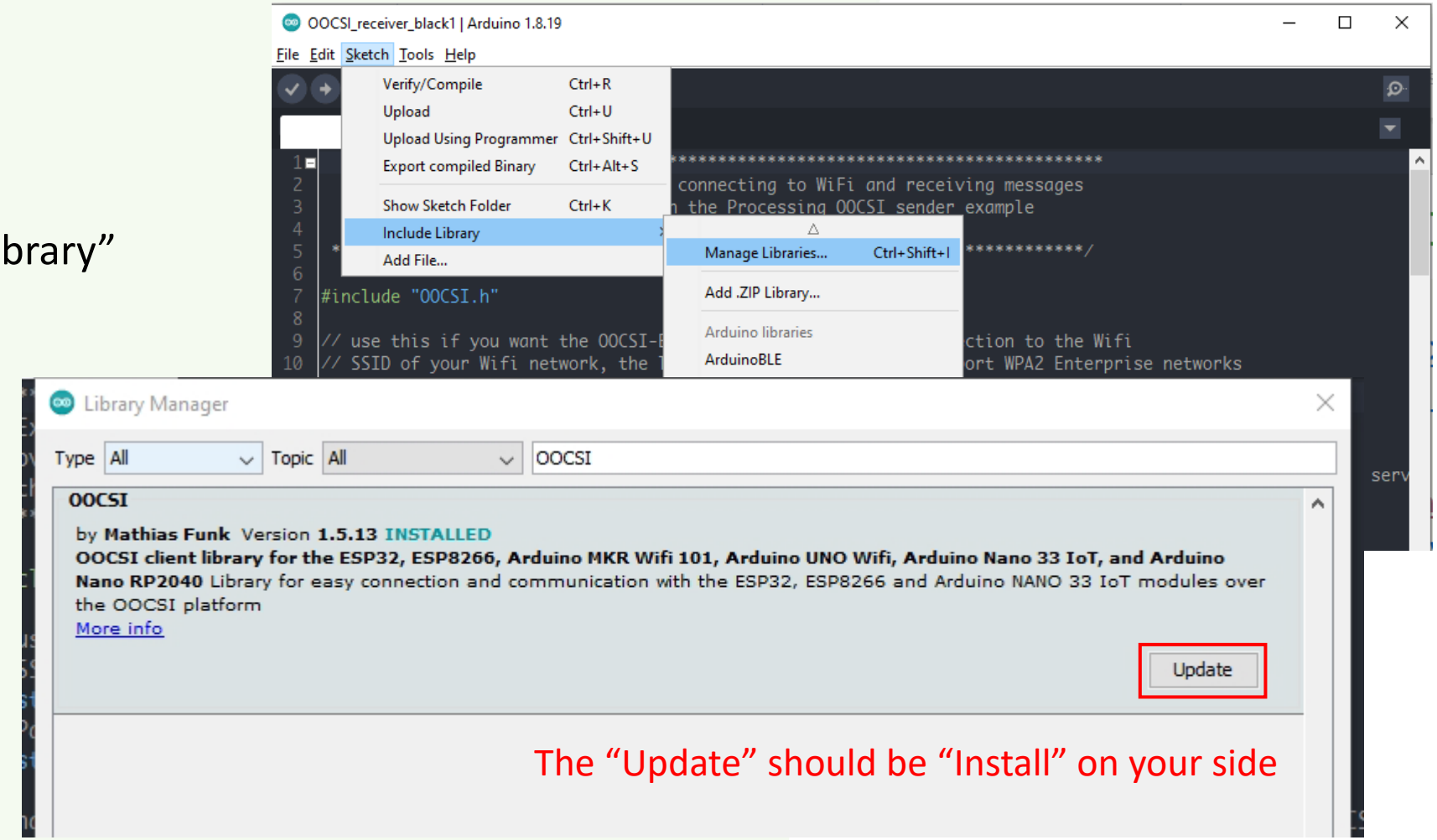
Step 2 – CONNECT IOT DEVICE (ESP32) (1/5)

- Arduino IDE
  - Install Arduino IDE
  - Add ESP32 board package to Arduino IDE
- Install ESP32 USB driver
  - Connect ESP32 board and check first, try the drivers if it’s not working
  - Windows machine
    - Download driver
    - How to install
  - Mac machine
    - Try to connect directly and check

Step 2 – CONNECT IOT DEVICE (ESP32) (2/5)

Install OOCSI library to Arduino IDE

1. Open Arduino IDE
2. “Sketch” -> “Include Library” -> “Manage Libraries”
3. Search “OOCSI”
4. Click “Install”
5. More information

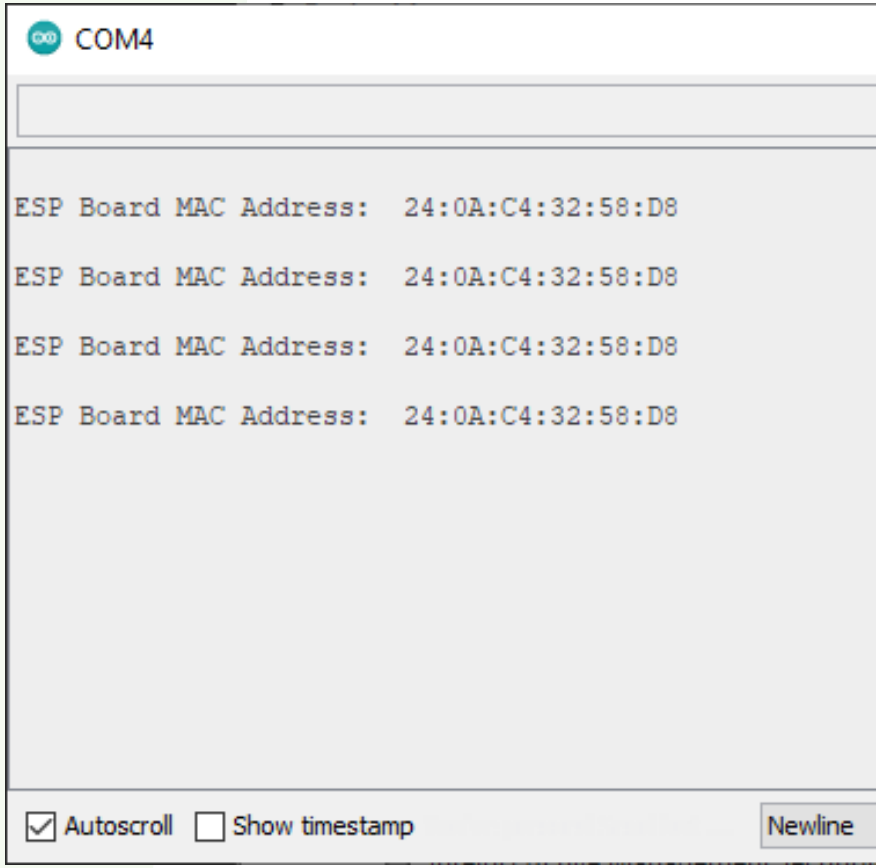


Step 2 – CONNECT IOT DEVICE (ESP32) (3/5)

- Paste the following code to a new sketch of Arduino IDE and save:

```
#include <WiFi.h>
void setup(){
  Serial.begin(115200);
}

void loop(){
  Serial.println();
  Serial.print("ESP Board MAC Address:  ");
  Serial.println(WiFi.macAddress());
  delay(3000);
}
```



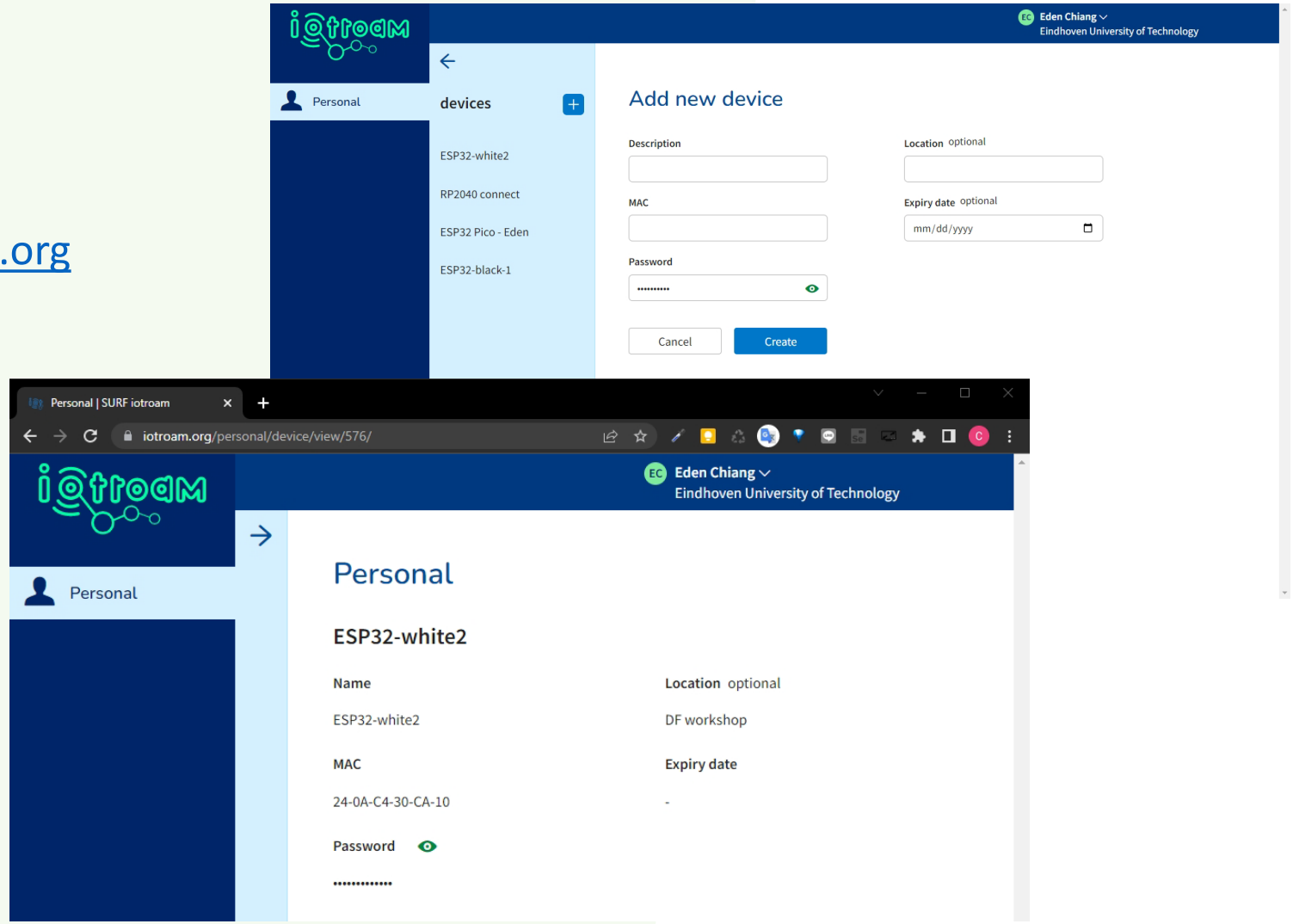
Step 2 – CONNECT IOT DEVICE (ESP32) (4/5)

- Check the MAC address of the IoT device
  - Upload a sketch to IoT device with Arduino IDE
    - Choose the board we're using:
      "Tools" -> "Board:…something_here…" -> "ESP32 Arduino" -> "DOIT ESP32 DEVKIT V1"
    - Select the COM port you checked on the Device Manager:
      "Tools" -> "Port" -> "COM #", # is a number with 1 or 2 digits
  - Serial Monitor: "Tools" -> "Serial Monitor"
  - The MAC address would be a string of 6 x 2 characters combined with ":", which looks like -- ab:83:fd:83:e4:89
  - Copy the MAC address you got

Step 2 – CONNECT IOT DEVICE (ESP32) to iotroam network (5/5)

Register IoT device to “iotroam” network

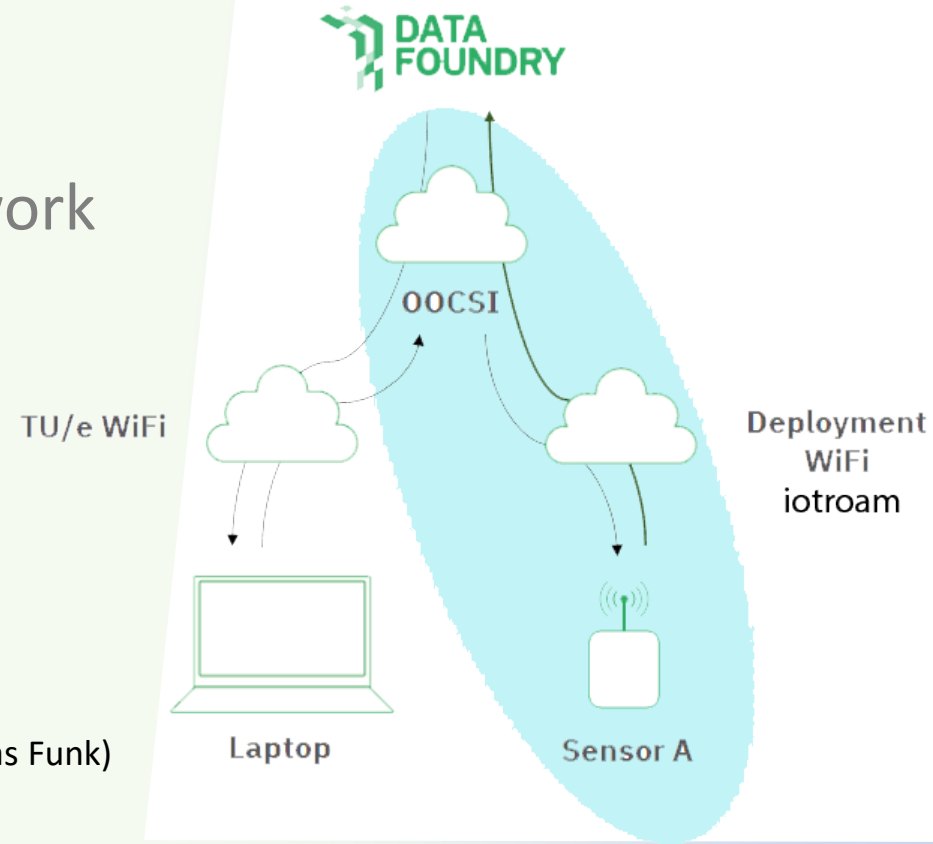
1. Register page: https://iotroam.org
2. Click on “add device” or “+”
3. Fill in the MAC address and other details
4. Set the “Expiry date” as tomorrow
5. Set and copy the password
6. Click Create



(Source: https://iotroam.org)

Practice: Data collection by Data Foundry through OOCSI

1. Preset of Data Foundry
    1. Create a project → Should be done by Data Foundry automatically
    2. Create an IoT dataset
    3. Create a device
2. Connect IoT device (ESP32) to iotroam network
3. **Send data through OOCSI**
4. Check data in OOCSI UI Client page
5. Save data to own IoT dataset



(Source from Mathias Funk)

Step 3 – SEND DATA THROUGH OOCSI (1/5)

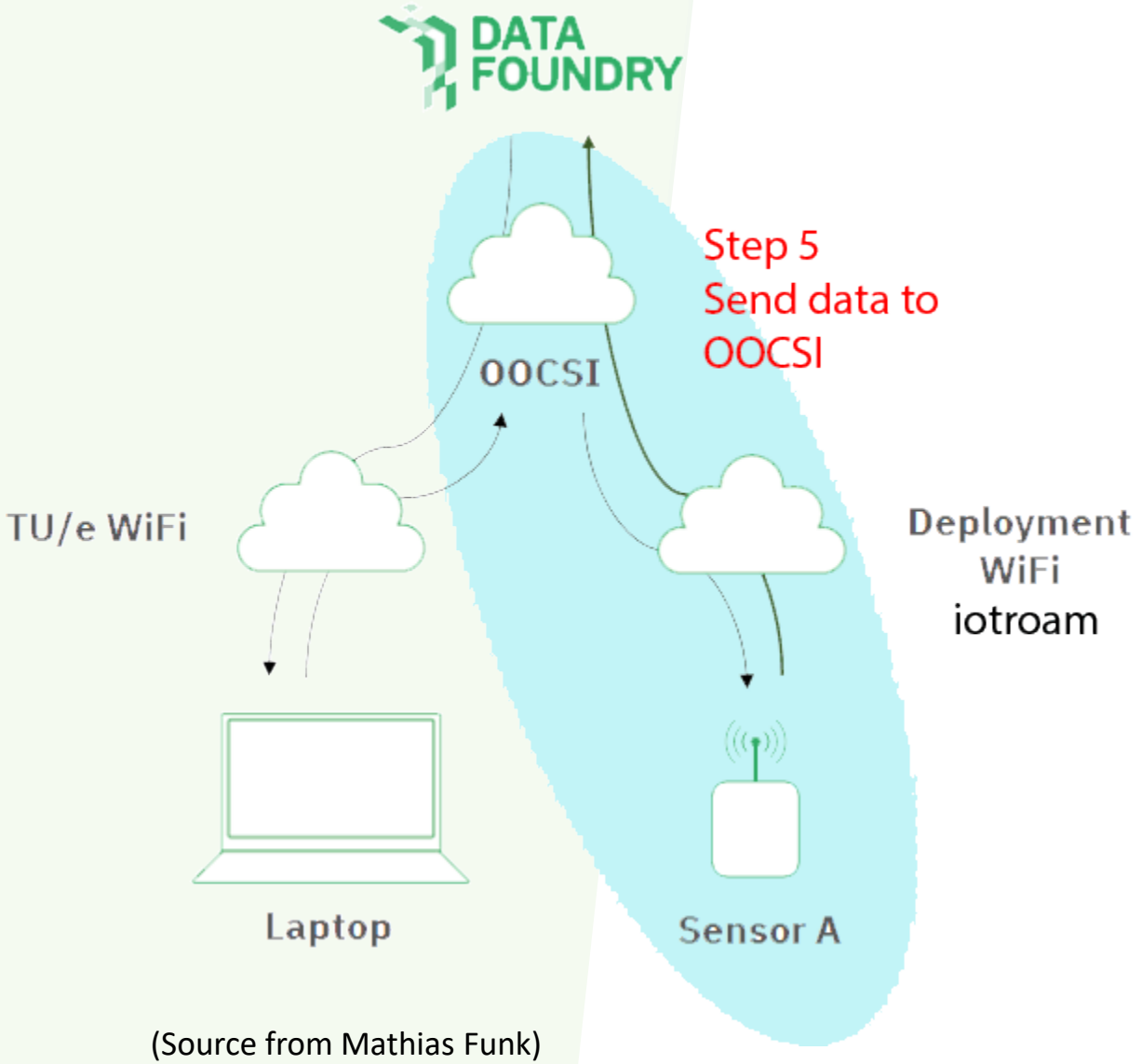
What is OOCSI?

“ OOCSI is a design prototyping *middleware* that allows ‘clients’ across platforms and programming languages to communicate via a ‘server’ (or ‘broker’ if you prefer middleware terminology). ”

-- Mathias Funk, OOCSI creator

Website: https://oocsi.net/

Github: https://github.com/iddi/oocsi

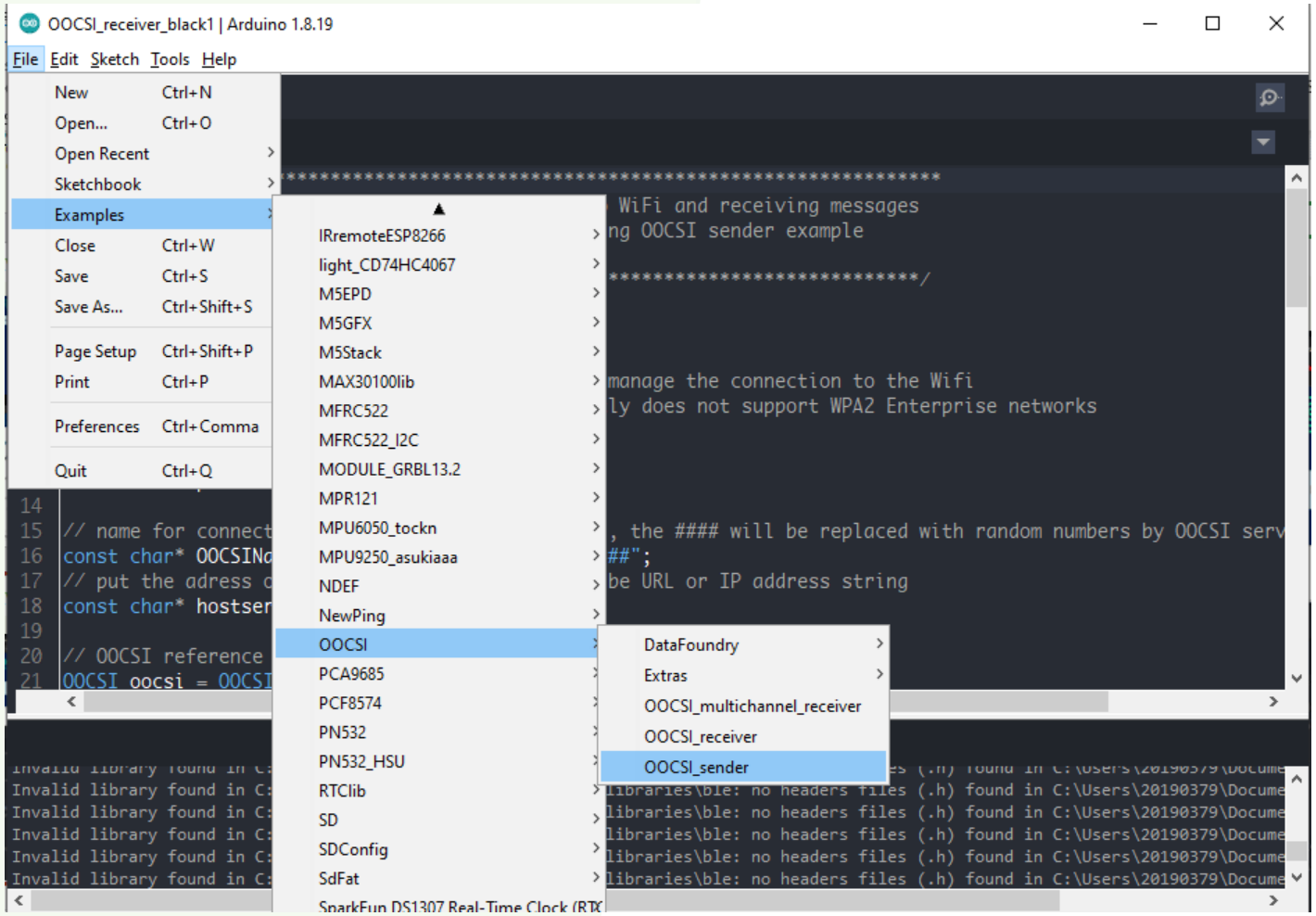


(Source from Mathias Funk)

Step 3 – SEND DATA THROUGH OOCSI (2/5)

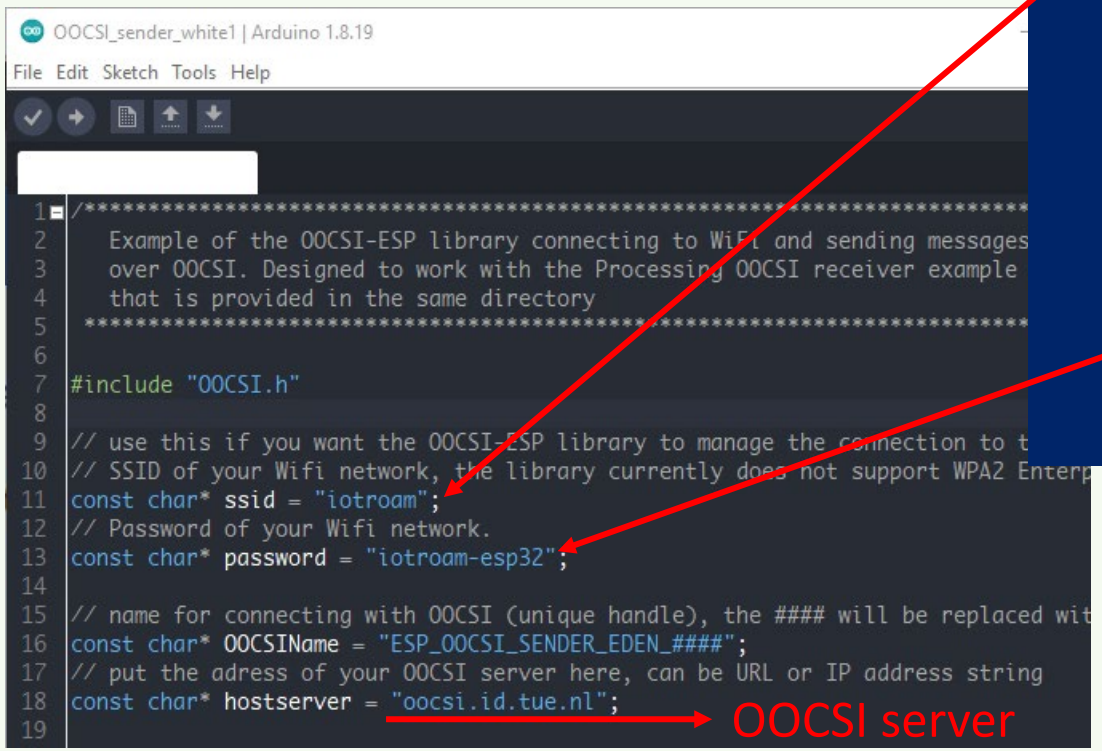
Open example code of sending data to Data Foundry through OOCSI

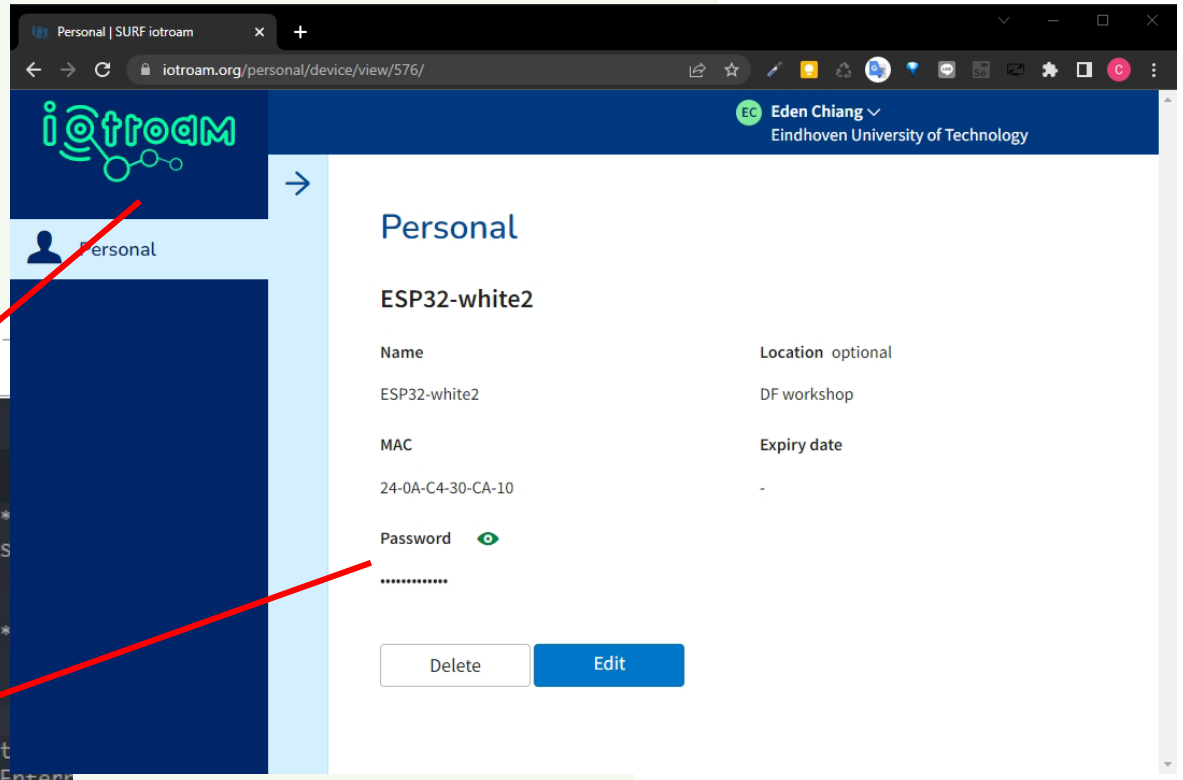
1. Open Arduino IDE
2. “File”
   -> “Examples”
   -> “OOCSI”
   -> “OOCSI_sender”



Step 3 – SEND DATA THROUGH OOCSI (3/5)

iotroam WiFi connection setting





(Source: https://iotroam.org)

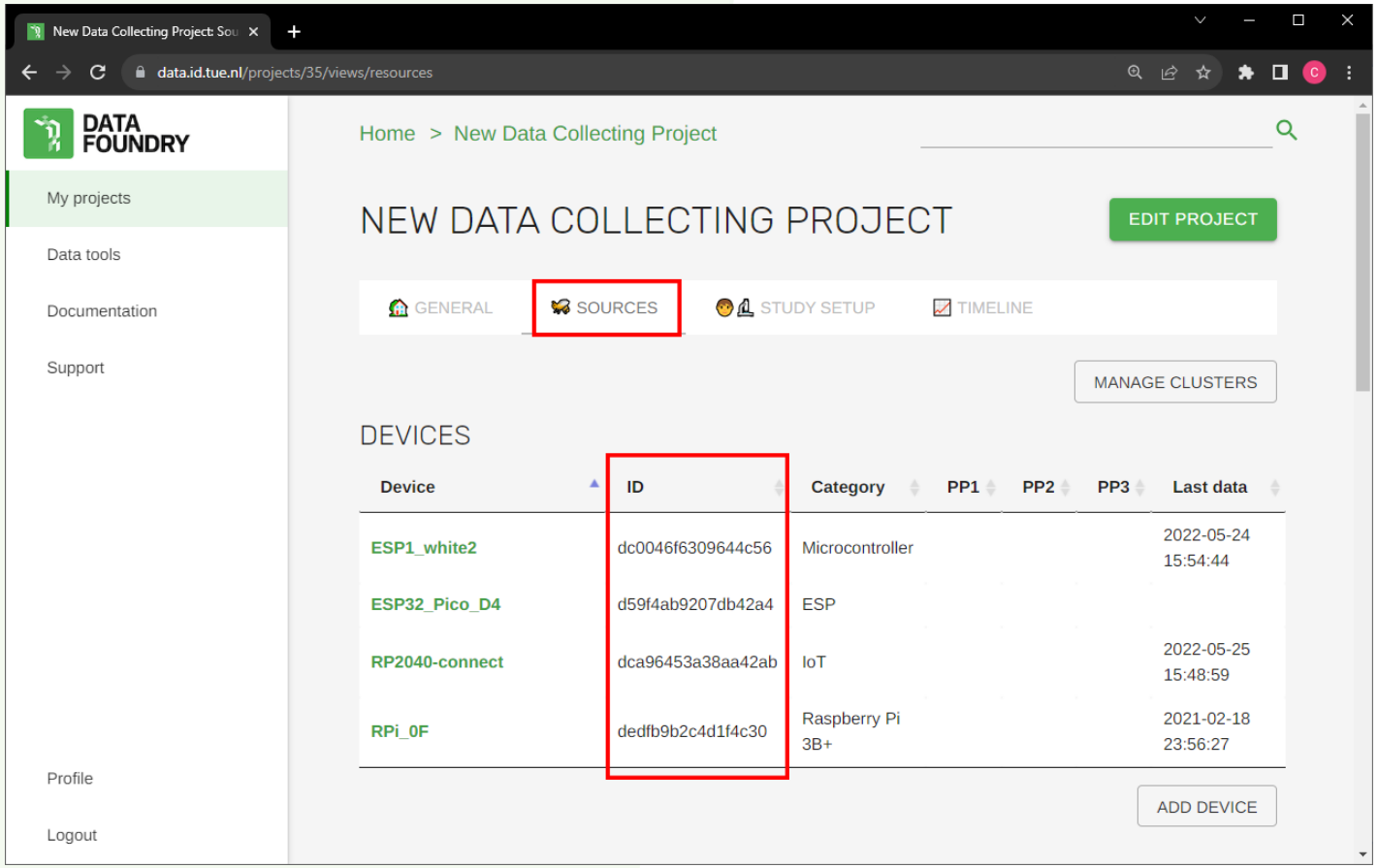
 # Copy code here:

```
const char* ssid = “iotroam”;
const char* password = “what_you_set_in_iotroam”;
const char* hostserver = “oocsi.id.tue.nl”;
const char* OOCSIName = “Eden_workshop_ESP_####”;
```

Step 3 – SEND DATA THROUGH OOCSI (4/5)

Copy the ID of the IoT device

1. Click on SOURCES tab on project page
2. Find the newly created IoT device from the devices table
3. Copy the ID

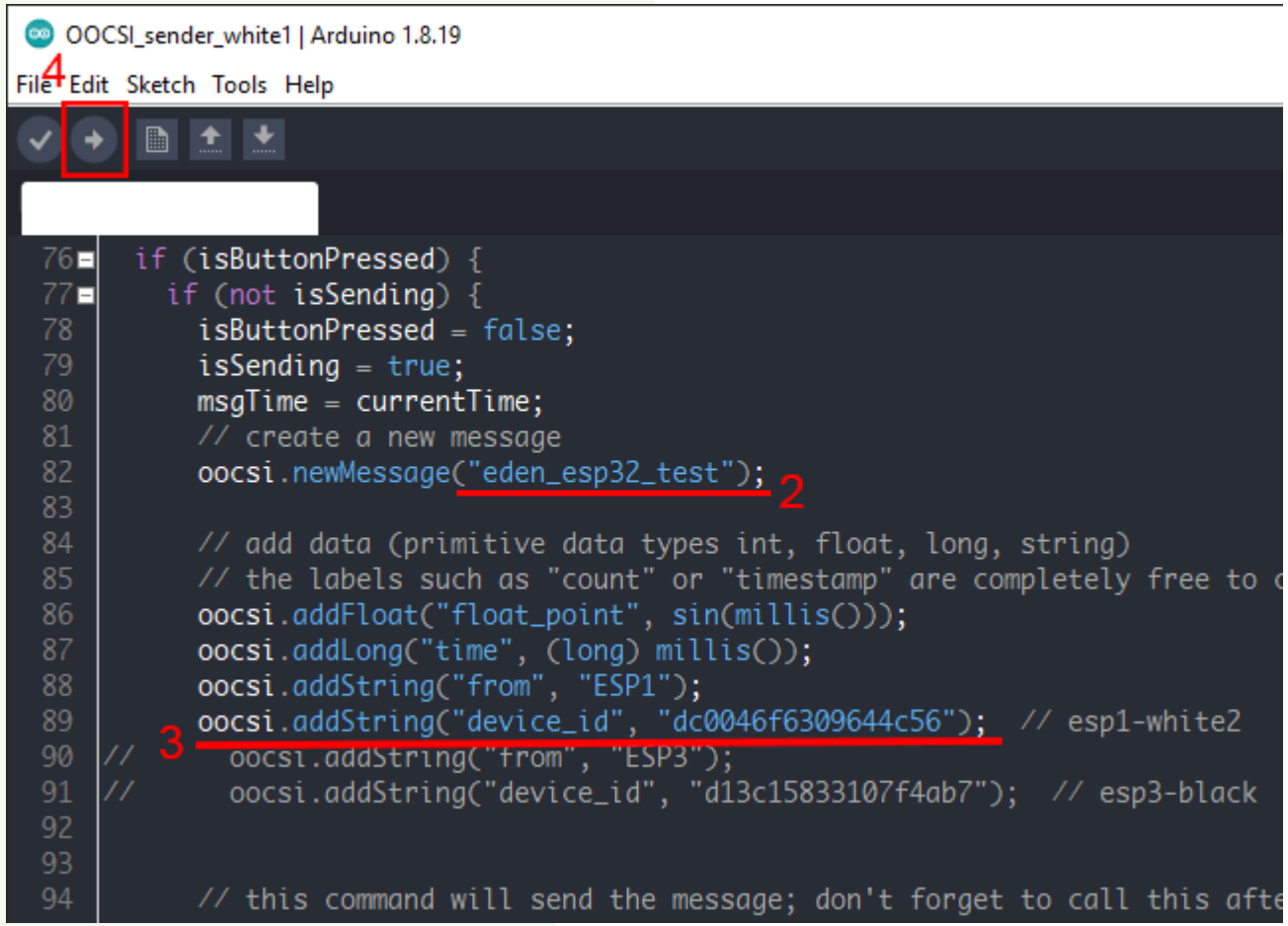


(Source: https://data.id.tue.nl)

Step 3 – SEND DATA THROUGH OOCSI (5/5)

Configuration for OOCSI in code

1. Back to OOCSI_sender in Arduino IDE
2. Replace the channel name as “eden_esp32_test” in the line starts with “oocsi.newMessage”
3. To set device ID: Add one line code between the “oocsi.newMessage(…);” and “oocsi.sendMessage();”:
   oocsi.addString(
   “device_id”,
   “PASTE_DEVICE_ID_HERE”);
4. Upload the code to IoT device

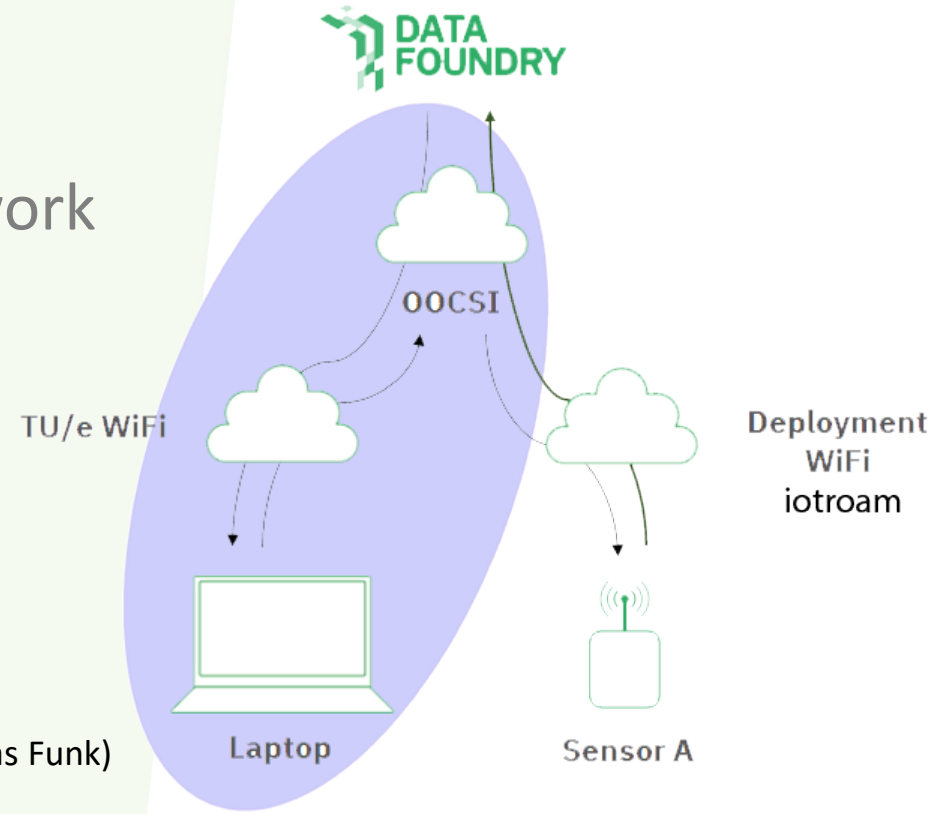


 # Copy code here:

oocsi.newMessage(“eden_esp32_test”);
oocsi.addString(“device_id”, “your_device_reference_ID”);

Practice: Data collection by Data Foundry through OOCSI

1. Preset of Data Foundry
    1. Create a project → Should be done by Data Foundry automatically
    2. Create an IoT dataset
    3. Create a device
2. Connect IoT device (ESP32) to iotroam network
3. Send data through OOCSI
4. **Check data in OOCSI UI Client page**
5. Save data to own IoT dataset

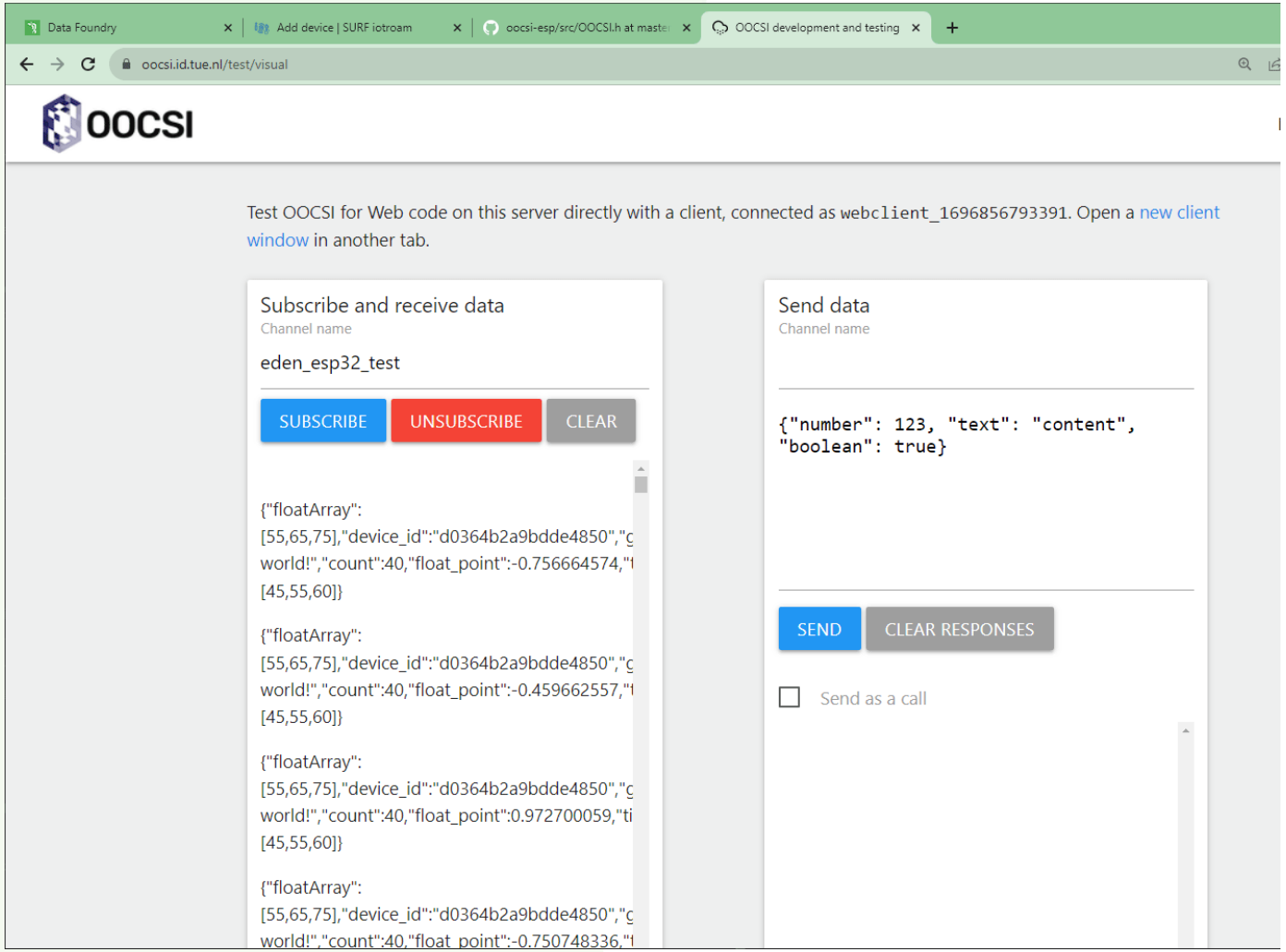


(Source from Mathias Funk)

Step 4 – CHECK DATA ON OOCSI UI CLIENT PAGE

Check data on UI Client page

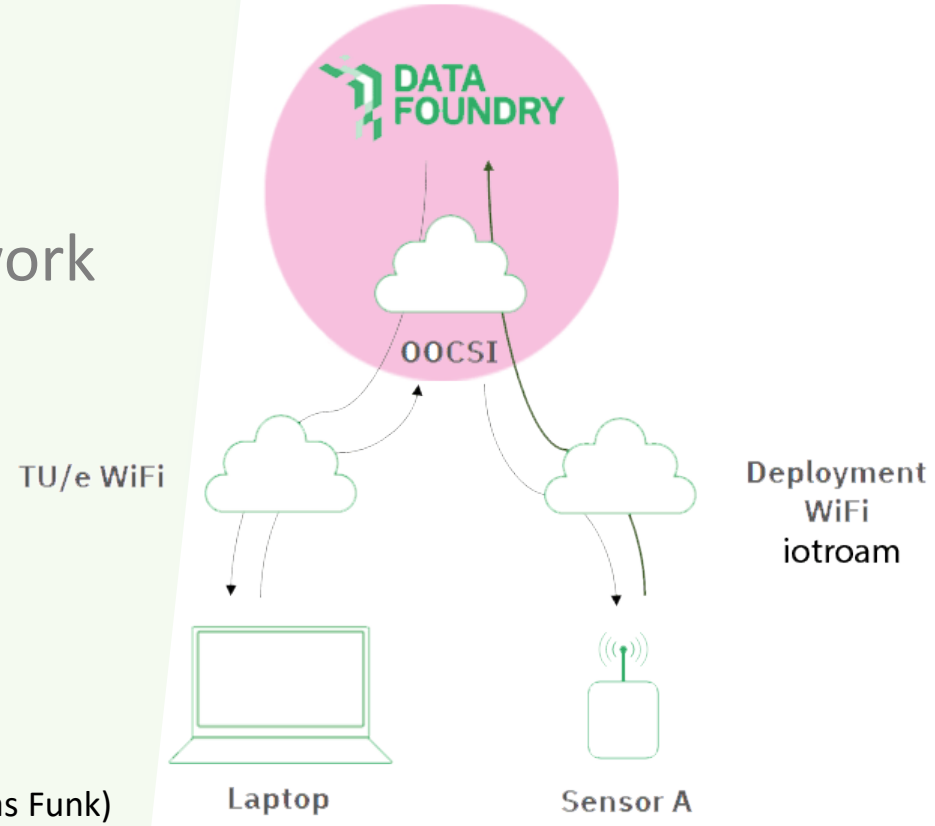
1. Open OOCSI UI Client page: https://oocsi.id.tue.nl/test/visual
   SUBSCRIBE to channel: “**eden_esp32_test**”
2. Check whether the data is coming into the channel
3. Unsubscribe the channel after checking



(Source: https://oocsi.id.tue.nl)

Practice: Data collection by Data Foundry through OOCSI

1. Preset of Data Foundry
    1. Create a project → Should be done by Data Foundry automatically
    2. Create an IoT dataset
    3. Create a device
2. Connect IoT device (ESP32) to iotroam network
3. Send data through OOCSI
4. Check data in OOCSI UI Client page
5. **Save data to own IoT dataset via OOCSI**

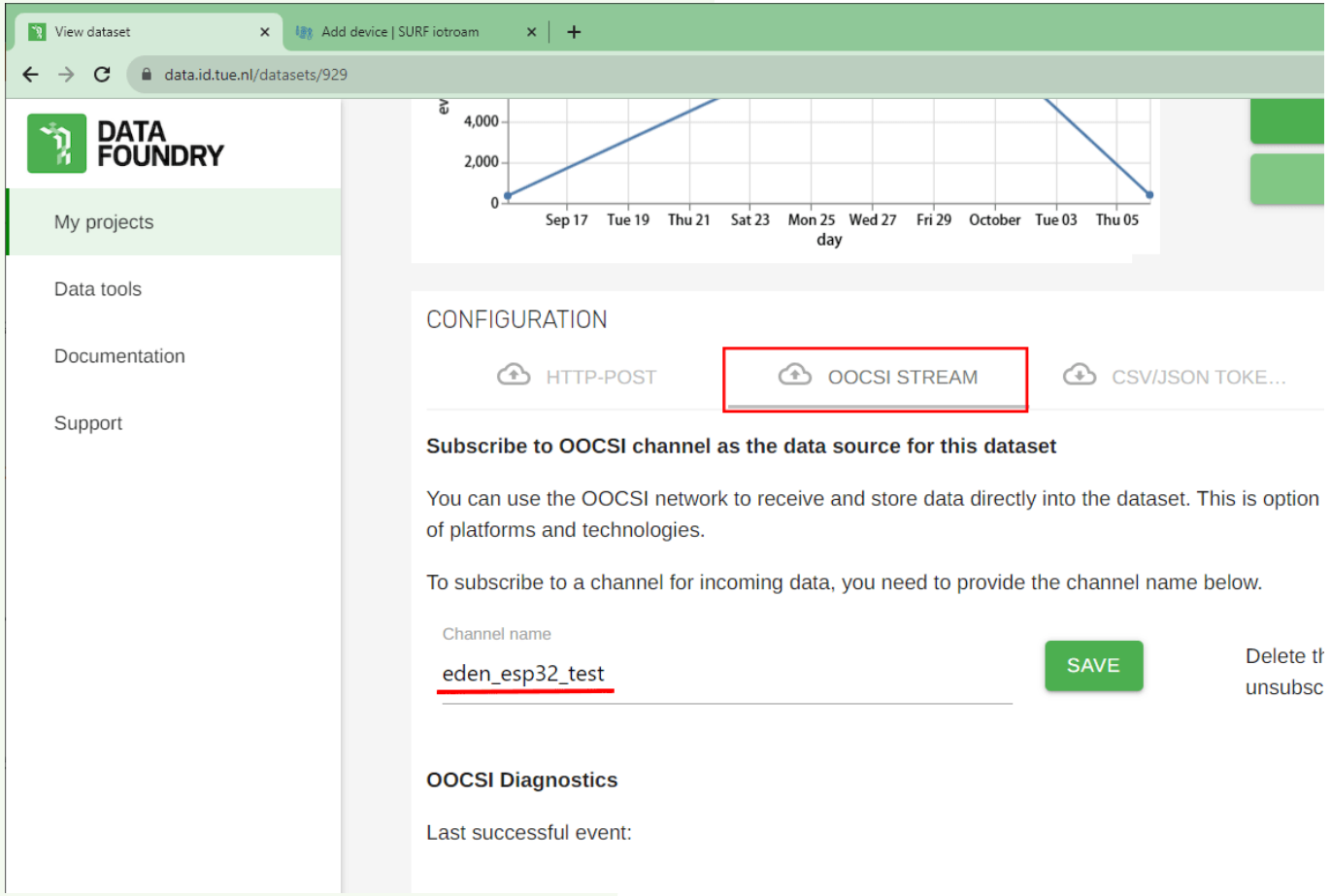


(Source from Mathias Funk)

Step 5 – SAVE DATA INTO OWN IOT DATASET VIA OOCSI (1/4)

Configuration of OOCSI channel

1. On Data Foundry, open the IoT dataset page created at the beginning
2. Define the OOCSI channel name under the OOCSI STREAM tab (second one by the left side) in Configuration block
3. Click “SAVE”

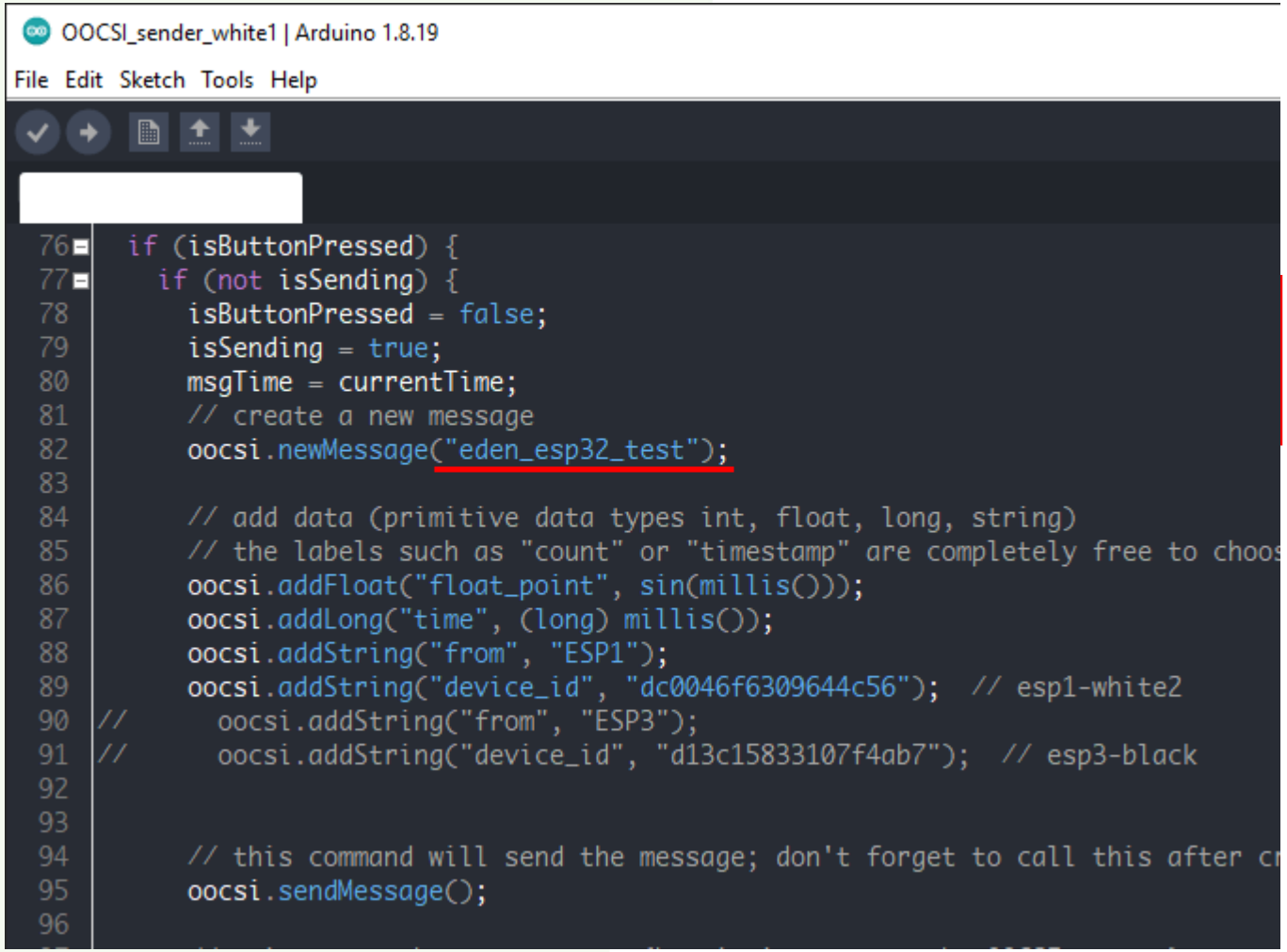


(Source: https://data.id.tue.nl)

Step 5 – SEND DATA TO OWN IOT DATASET VIA OOCSI (2/4)

Configuration with OOCSI channel in OOCSI_sender code

1. Back to OOCSI_sender code
2. Update the OOCSI channel name for sending message as the same one you defined in your IoT dataset
3. Upload code to IoT device



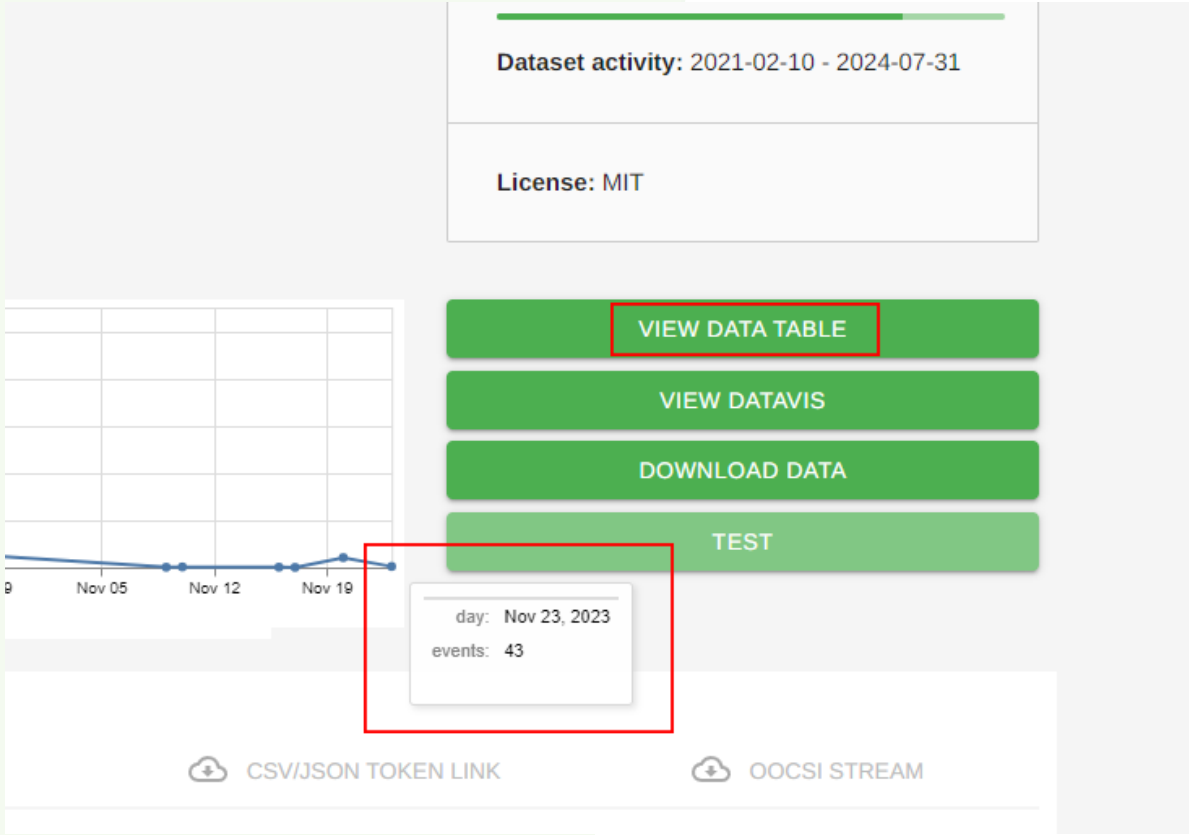
Step 5 – SEND DATA TO OWN IOT DATASET VIA OOCSI (3/4)

Store upcoming data in Data Foundry

Open and refresh the IoT dataset page and check the blue dots on the chart of the page or “VIEW DATA TABLE” and check the latest log.

The blue dots represent the events captured by Data Foundry. You can check the detail by hovering the dots with mouse.

As the number of events is increasing by every refreshing the page, which means the IoT dataset is set properly.



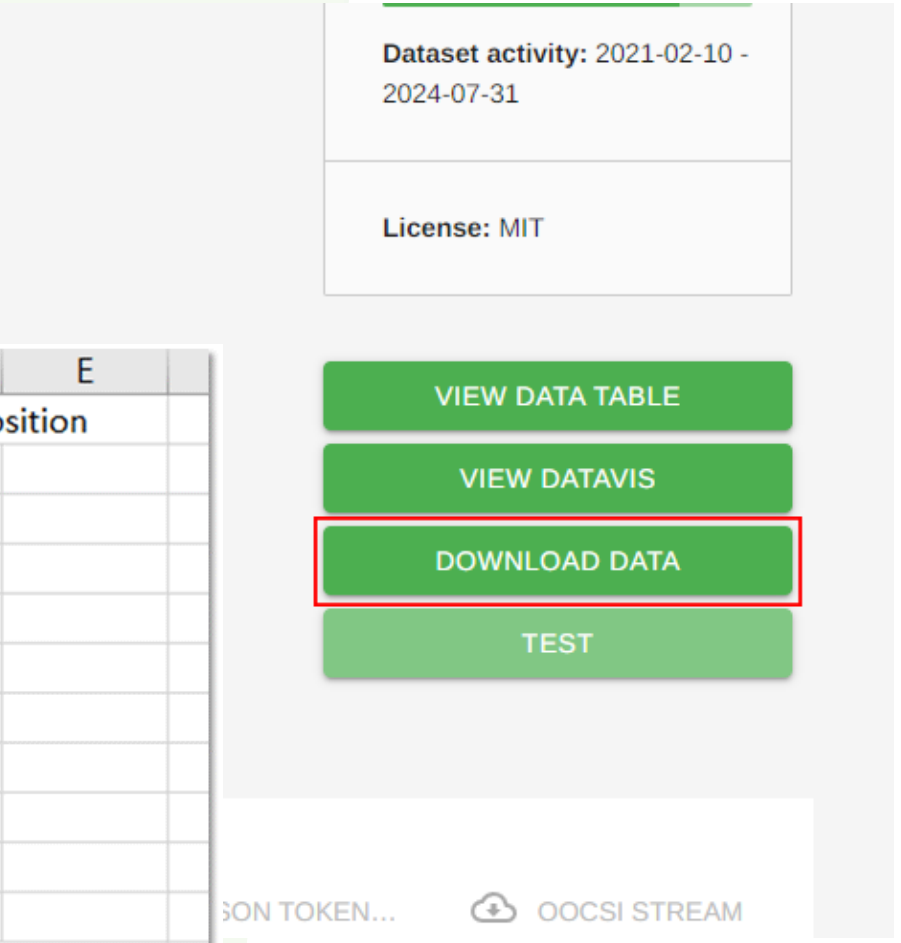
(Source: https://data.id.tue.nl)

Step 5 – SEND DATA TO OWN IOT DATASET VIA OOCSI (4/4)

Download your data as a CSV file

Click on the download button on the right to download the data directly as a CSV file.

	A	B	C	D	E
1	id,device_id,ts,activity,pp1,pp2,pp3,color,position				
2	1,240,2020-09-29T10:42:48.985,,,,,0,0				
3	2,240,2020-09-29T10:42:49.566,,,,,0,0				
4	3,240,2020-09-29T10:42:50.163,,,,,0,0				
5	4,240,2020-09-29T10:42:50.763,,,,,0,0				
6	5,240,2020-09-29T10:42:51.266,,,,,88,49				
7	6,240,2020-09-29T10:42:51.866,,,,,55,78				
8	7,240,2020-09-29T10:42:52.462,,,,,113,74				
9	8,240,2020-09-29T10:42:52.965,,,,,113,74				
10	9,240,2020-09-29T10:42:53.467,,,,,130,99				
11	10,240,2020-09-29T10:42:54.065,,,,,89,112				
12	11,240,2020-09-29T10:42:54.664,,,,,158,198				
13	12,240,2020-09-29T10:42:55.164,,,,,194,142				



(Source: https://data.id.tue.nl)

FAQ

Q. ESP32 is not uploaded automatically after clicking the upload button?

A. You have to press the **BOOT** button on the ESP32 board when the "**Connecting…**" message is showing up in the console window of Arduino IDE as uploading the sketch to ESP32.

Q. Why is some sensors (pins) will not work as the Wi-Fi module is working?

A. This is because the design of this model, when the Wi-Fi module is working, only **ADC2** pins are available, which means **ADC1** will be occupied by Wi-Fi module; otherwise, **ADC1** pins will work. Check the pinout here.

Q. Does DOIT ESP32 DevKit V1 (the one we're using in the workshop) have **5V** output pin?

A. Yes, but 5V output is **only available** by the **Vin** pin as the ESP32 is powered by **USB cable**.

Q. What is the case of the data can be saved into IoT dataset without **device_id**?

A. This happened as the target dataset is set as "**open participation"** (which can be found on the edit page of the dataset), which means data from any resources are available for the dataset. Otherwise, the device_id is required for identifying the sources in the project.

Reference links (1/2)

- Data Foundry: https://data.id.tue.nl/
  - FAQ of Data Foundry
- OOCSI: https://oocsi.id.tue.nl/
- Practical exercises with Data Foundry (Github): PlayWithDataFoundry
- How to get MAC address and register to iotroam?
- IoT network in TU/e: Iotroam

Reference links (2/2)

- Arduino IDE
  - Install Arduino IDE
  - Add ESP32 board package to Arduino IDE
- Install ESP32 USB driver
  - Connect ESP32 board and check first, try the drivers if it’s not working
  - Windows machine
    - Download driver
    - How to install
  - Mac machine
    - Try to connect directly and check
- Check port of Windows and Mac machines

Real things