



Metrology for Integrated marine maNagement and Knowledge-transfer nEtwork

INFRAIA-02-2020: Integrating Activities for Starting
Communities



Project funded by the European Commission within the Horizon 2020
Programme (2014-2020)
Grant Agreement No. 101008724

Requirements

A computer

Access to the internet

Some data to analyse

Data Basics

Data Basics

HOW DO YOU WANT TO USE YOUR KIT?

Recommended



ONLINE (USING WI-FI)



OFFLINE (SD CARD)

Data Basics

Two modes of operation

Network Mode

Data goes from the SCK to the Platform

You can get the data in many ways, in near realtime:

- The REST API (for everything) via the web, the dashboard, python scripts... or anything!
- Websockets (for real time updates)

Offline Mode

Data doesn't leave the SCK

You need to extract the SD card from the SCK and load it somewhere (a spreadsheet, a script)

More on the data

Offline Data

Offline mode

We will see how to concatenate the data using python in the next session

We will also see how to load the data in python or other software!

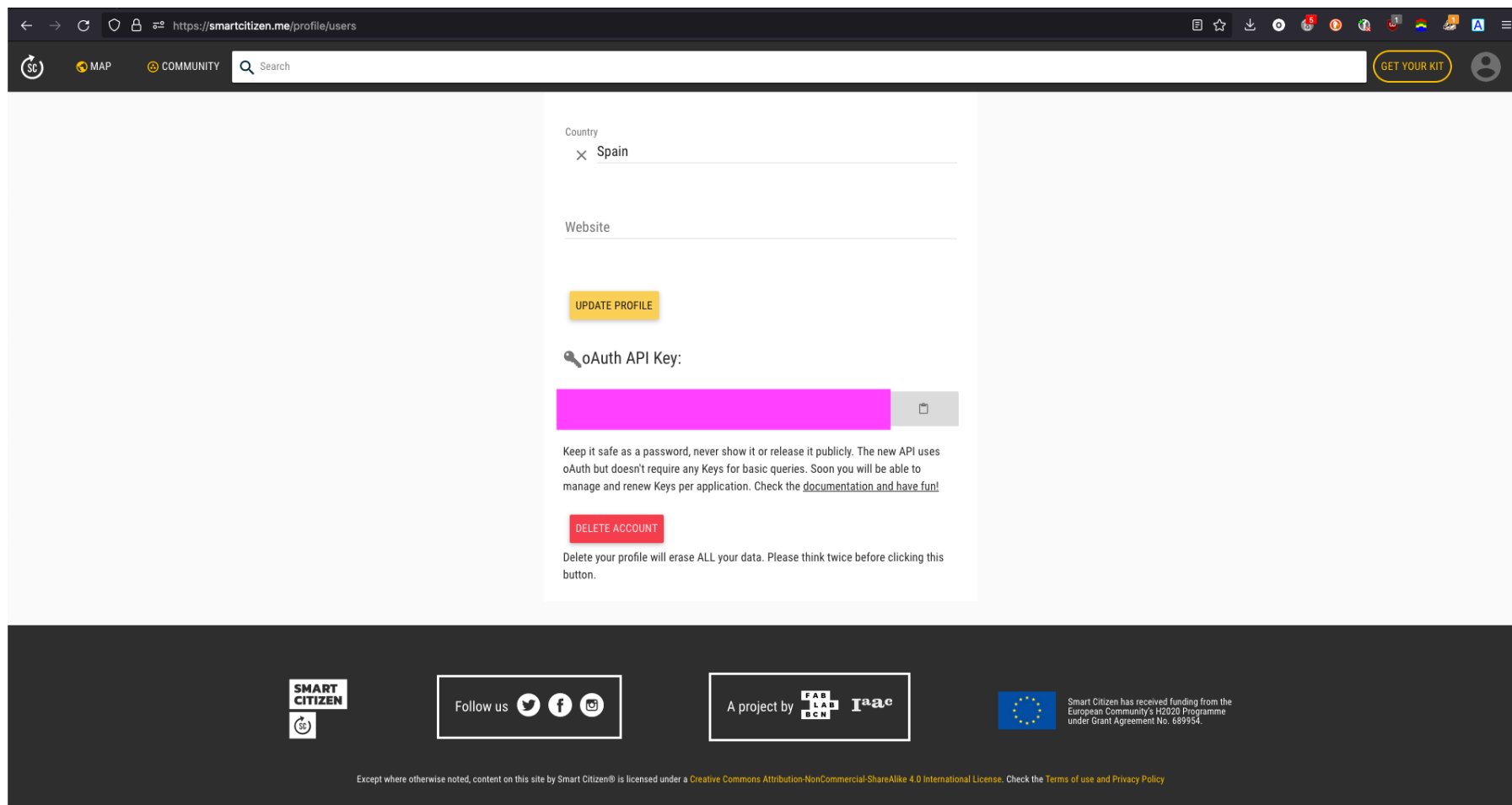
Data doesn't leave the SCK. It's only stored on the SD card

You need to extract the SD card from the SCK and load it somewhere (a spreadsheet, a script). One CSV file per day in the SD card:

One CSV file
per day in
the SD card!

TIME	TEMP	HUM	BATT	LIGHT	NOISE_A	PRESS	CCS811_VOCS	CCS811_ECO2	PM_1	PM_25	PM_10	PM_10	PM_10
ISO 8601	C	%	%	Lux	dBA	kPa	ppb	ppm	ug/m3	ug/m3	ug/m3	ug/m3	PM_10
Time	Temperature	Humidity	Battery	Light	Noise dBA	Barometric pressure	VOC Gas CCS811	eCO2 Gas CCS811	PM 1.0	PM 2.5	PM 10.0	ug/m3	PM_10
	55	56	10	14	53	58	113	112	89	87	88	PM 10.0	ug/m3
2021-06-27T23:58:47Z	27.18	61.34	59	0	47.22	100.96	148.00	1372.00	null	null	null	88	PM 10.0
2021-06-28T00:03:47Z	27.18	61.40	59	0	46.93	100.96	148.00	1372.00	null	null	null	null	88
2021-06-28T00:08:47Z	27.18	61.43	58	0	47.34	100.95	146.00	1360.00	null	null	null	null	null
2021-06-28T00:08:47Z	27.18	61.43	58	0	47.34	100.95	146.00	1360.00	null	null	null	null	null
2021-06-28T00:08:47Z	27.18	61.43	58	0	47.34	100.95	146.00	1360.00	null	null	null	null	null

Getting a token for posting data



The screenshot shows a web browser window with the URL `https://smartcitizen.me/profile/users`. The page has a dark header with navigation links: SC, MAP, and COMMUNITY, followed by a search bar. On the right of the header is a 'GET YOUR KIT' button and a user profile icon. The main content area is a user profile form. It includes a 'Country' field with 'Spain' selected, a 'Website' field, and an 'UPDATE PROFILE' button. Below this is the 'OAuth API Key' section, which features a large red text input field and a copy icon. A warning message states: 'Keep it safe as a password, never show it or release it publicly. The new API uses OAuth but doesn't require any Keys for basic queries. Soon you will be able to manage and renew Keys per application. Check the [documentation and have fun!](#)'. At the bottom of the form is a 'DELETE ACCOUNT' button with a warning: 'Delete your profile will erase ALL your data. Please think twice before clicking this button.' The footer contains the Smart Citizen logo, social media links, project partners (FAB LAB CERN and IaaC), the European Union flag, and funding information: 'Smart Citizen has received funding from the European Community's H2020 Programme under Grant Agreement No. 689954.' A small disclaimer at the very bottom reads: 'Except where otherwise noted, content on this site by Smart Citizen® is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#). Check the [Terms of use and Privacy Policy](#)'.

Processing Data

Processing data

Basic operations

[15]:

	ADC_48_0	ADC_48_1	ADC_48_2	ADC_48_3	ADC_49_0	ADC_49_1	ADC_49_2	ADC_49_3
TIME								
2021-10-15 09:33:00-04:00	0.244188	0.244563	0.291594	0.413922	0.311656	0.293016	0.326922	12.287812
2021-10-15 09:34:00-04:00	0.244563	0.244937	0.289375	0.408406	0.310906	0.295875	0.326906	12.287812
2021-10-15 09:35:00-04:00	0.244937	0.244438	0.295453	0.417281	0.310391	0.294047	0.326906	12.287812
2021-10-15 09:36:00-04:00	0.244563	0.244438	0.301688	0.415688	0.311109	0.294953	0.326906	12.287812
2021-10-15 09:37:00-04:00	0.244438	0.244813	0.293953	0.410937	0.310891	0.295578	0.326875	12.287812
...
2022-02-14 17:38:00-05:00	0.247719	0.248656	0.295203	0.328500	0.309828	0.290594	0.323703	12.287812
2022-02-14 17:39:00-05:00	0.248688	0.245750	0.292594	0.331234	0.310844	0.289750	0.323687	12.287812
2022-02-14 17:40:00-05:00	0.248719	0.246750	0.296094	0.333781	0.309984	0.290922	0.323687	12.287812
2022-02-14 17:41:00-05:00	0.248063	0.248594	0.292313	0.337328	0.310234	0.289734	0.323672	12.287812
2022-02-14 17:42:00-05:00	0.245594	0.245375	0.295969	0.344344	0.310000	0.291781	0.323672	12.287812

176230 rows x 22 columns

Processing data

Basic operations

[15]:

	ADC_48_0	ADC_48_1	ADC_48_2	ADC_48_3	ADC_49_0	ADC_49_1	ADC_49_2	ADC_49_3
TIME								
2021-10-15 09:33:00-04:00	0.244188	0.244563	0.291594	0.413922	0.311656	0.293016	0.326922	12.287812
2021-10-15 09:34:00-04:00	0.244563	0.244937	0.289375	0.408406	0.310906	0.295875	0.326906	12.287812
2021-10-15 09:35:00-04:00	0.244937	0.244438	0.295453	0.417281	0.310391	0.294047	0.326906	12.287812
2021-10-15 09:36:00-04:00	0.244563	0.244438	0.301688	0.415688	0.311109	0.294953	0.326906	12.287812
2021-10-15 09:37:00-04:00	0.244438	0.244813	0.293953	0.410937	0.310891	0.295578	0.326875	12.287812
...
2022-02-14 17:38:00-05:00	0.247719	0.248656	0.295203	0.328500	0.309828	0.290594	0.323703	12.287812
2022-02-14 17:39:00-05:00	0.248688	0.245750	0.292594	0.331234	0.310844	0.289750	0.323687	12.287812
2022-02-14 17:40:00-05:00	0.248719	0.246750	0.296094	0.333781	0.309984	0.290922	0.323687	12.287812
2022-02-14 17:41:00-05:00	0.248063	0.248594	0.292313	0.337328	0.310234	0.289734	0.323672	12.287812
2022-02-14 17:42:00-05:00	0.245594	0.245375	0.295969	0.344344	0.310000	0.291781	0.323672	12.287812

176230 rows x 22 columns

$$\text{Result} = f(\text{ADC_48_1}) + g(\text{ADC_48_3})$$

This can be done in any programming language without much hassle...

Processing data

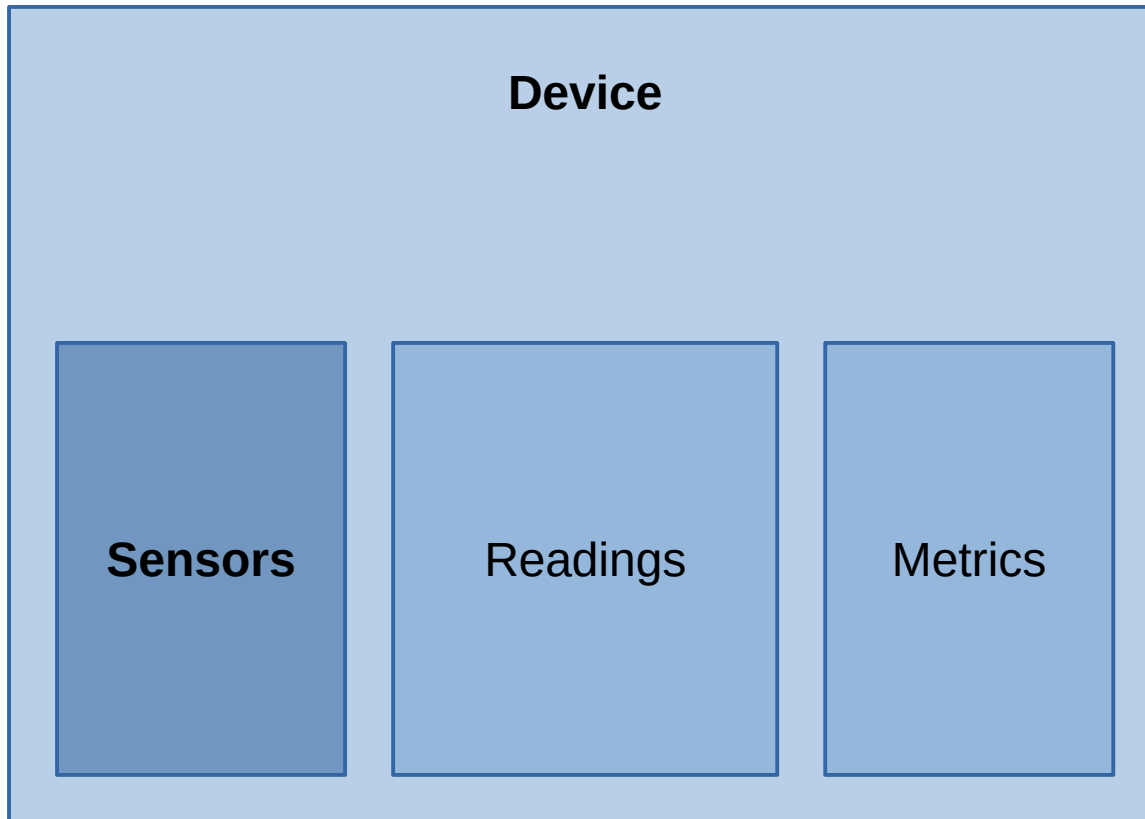
Making it repeatable

Objectives

- We want to be able to process the data ourselves without relying on scripts in someone's computer
- We want to be able to share recipes with the processing algorithms
- We want to be able to automatize the process for the data processing

Processing data

Making it repeatable

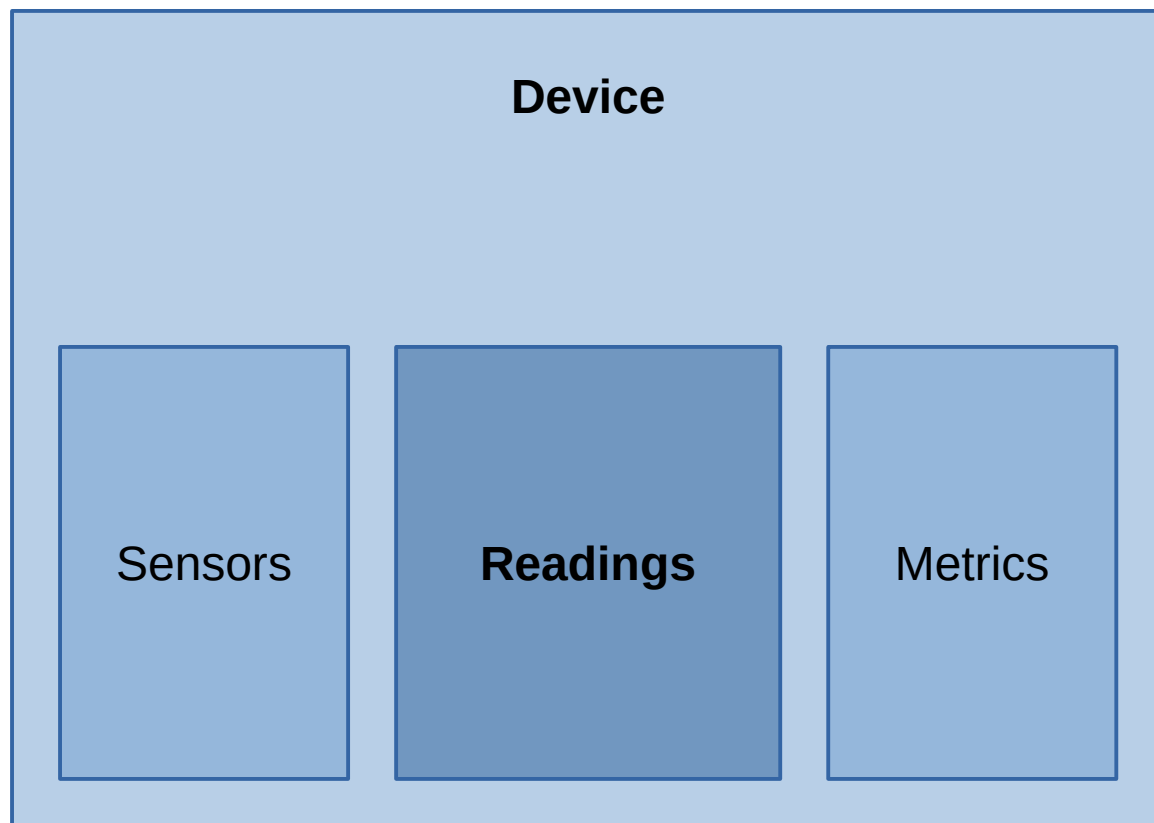


Where the sensors of that particular device live:

```
'CCS811_ECO2': {'desc': 'Equivalent CO2', 'id': '112', 'units': 'ppm'},  
  
'CCS811_VOCS': {'desc': 'total Volatile Organics Compounds', 'id': '113', 'units': 'ppm'},  
  
'PM_1': {'desc': 'PM1 measurement from PMS5003', 'id': '89', 'units': 'ug/m3'},  
  
'PM_10': {'desc': 'PM10 measurement from PMS5003', 'id': '88', 'units': 'ug/m3'},  
  
'PM_25': {'desc': 'PM2.5 measurement from PMS5003', 'id': '87'...
```

Processing data

Making it repeatable

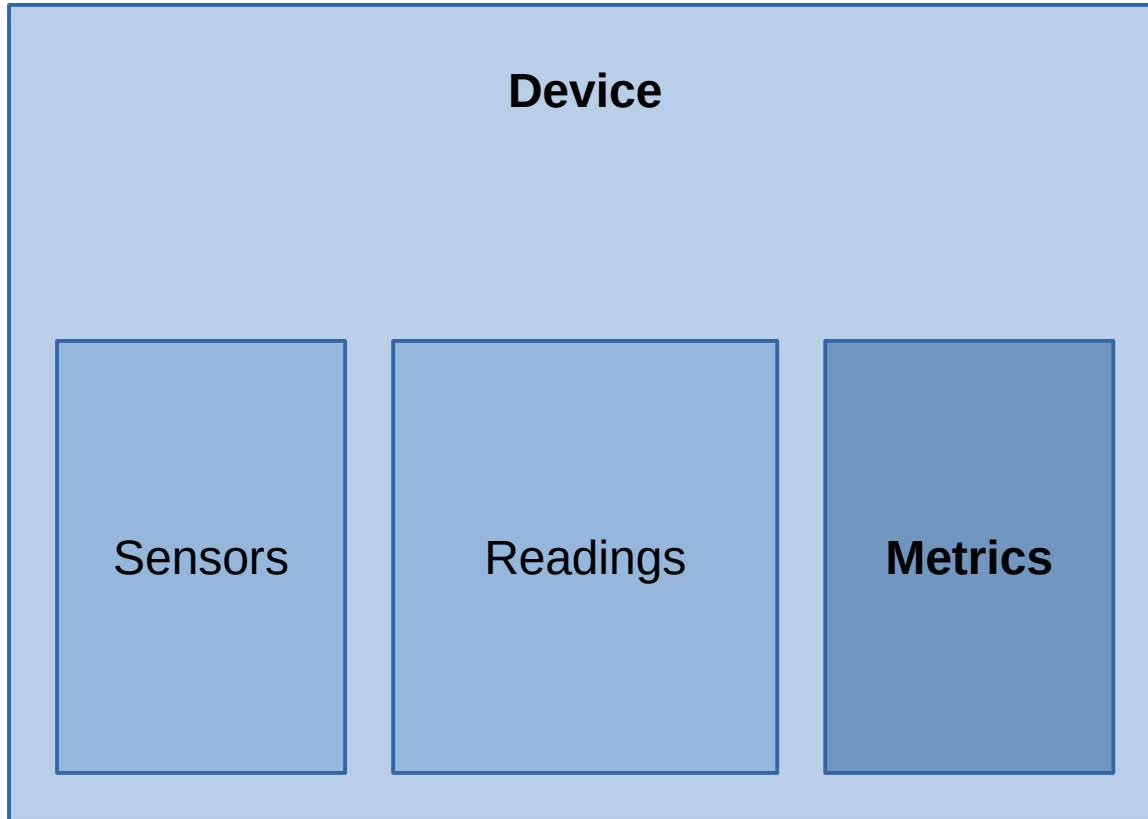


Where the actual data is

	ADC_48_0	ADC_48_1	ADC_48_2	ADC_48_3	ADC_49_0	ADC_49_1
TIME						
2021-10-15 09:33:00-04:00	0.244188	0.244563	0.291594	0.413922	0.311656	0.293016
2021-10-15 09:34:00-04:00	0.244563	0.244937	0.289375	0.408406	0.310906	0.295875
2021-10-15 09:35:00-04:00	0.244937	0.244438	0.295453	0.417281	0.310391	0.294047
2021-10-15 09:36:00-04:00	0.244563	0.244438	0.301688	0.415688	0.311109	0.294953
2021-10-15 09:37:00-04:00	0.244438	0.244813	0.293953	0.410937	0.310891	0.295578
...
2022-02-14 17:38:00-05:00	0.247719	0.248656	0.295203	0.328500	0.309828	0.290594
2022-02-14 17:39:00-05:00	0.248688	0.245750	0.292594	0.331234	0.310844	0.289750
2022-02-14 17:40:00-05:00	0.248719	0.246750	0.296094	0.333781	0.309984	0.290922
2022-02-14 17:41:00-05:00	0.248063	0.248594	0.292313	0.337328	0.310234	0.289734
2022-02-14 17:42:00-05:00	0.245594	0.245375	0.295969	0.344344	0.310000	0.291781

Processing data

Making it repeatable



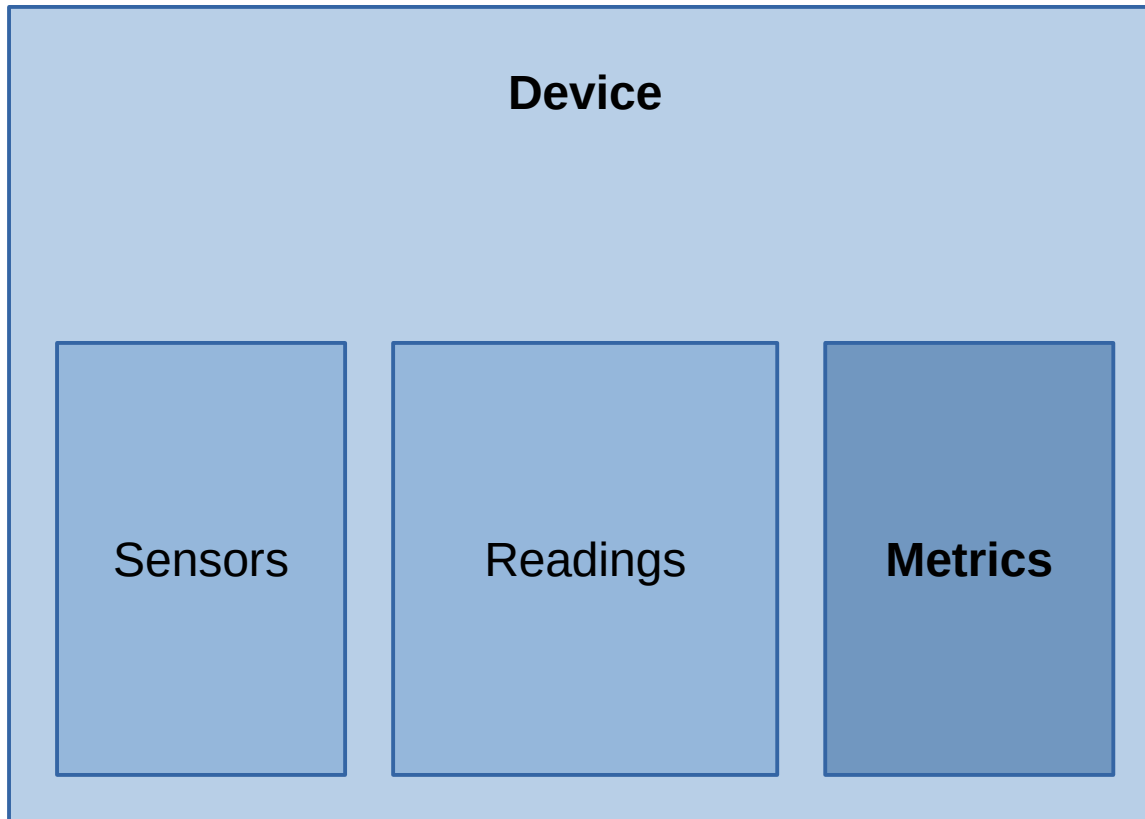
The “recipes” for data processing

All you need to do is:

1. Create a python function that processes the data as simple or as complex as you need
2. Define the arguments for that function
3. Put together a recipe in JSON format

Processing data

Making it repeatable



The “recipes” for data processing

```
'CO': {'desc': 'Calculation of CO based on AAN 803-04',  
      'id': 152,  
      'kwargs': {'ae': 'ADC_48_2',  
                  'alphasense_id': '132070362',  
                  'from_date': '2021-01-21',  
                  'timezone': 'America/New_York',  
                  't': 'EC_SENSOR_TEMP',  
                  'to_date': None,  
                  'we': 'ADC_48_3'},  
      'post': True,  
      'process': 'alphasense_803_04',  
      'units': 'ppb'},
```

What this recipe does:

Gets analog sensor data and calculates CO based on a function called `alphasense_803_04`

The function also uses calibration data and returns data in ppb

We post the CO data to the platform in ID=152

Processing data

Automatizing the process

We can now create **blueprints** that will hold these recipes and that automatically will have a description for the **algorithms in ALL our devices** with the same blueprint

We can then share the blueprint and have it available for everyone:

<https://github.com/fablabbcn/smartcitizen-data/tree/master/blueprints>

More info is available at:

<https://docs.smartcitizen.me/Guides/data/Custom%20data%20processing/>

Processing data

What happens with individual calibrations?

We need to identify the individual hardware with a particular **HARDWARE_ID**

- We can associate a **blueprint** with it
- We can associate **calibration** data with it too

Both of them, can be stored in a single file and then used for processing the data

<https://github.com/fablabbcn/smartcitizen-data/tree/master/hardware>

Note: this is not necessary for the water sensors in the default setup. Only needed if you want to go beyond it.



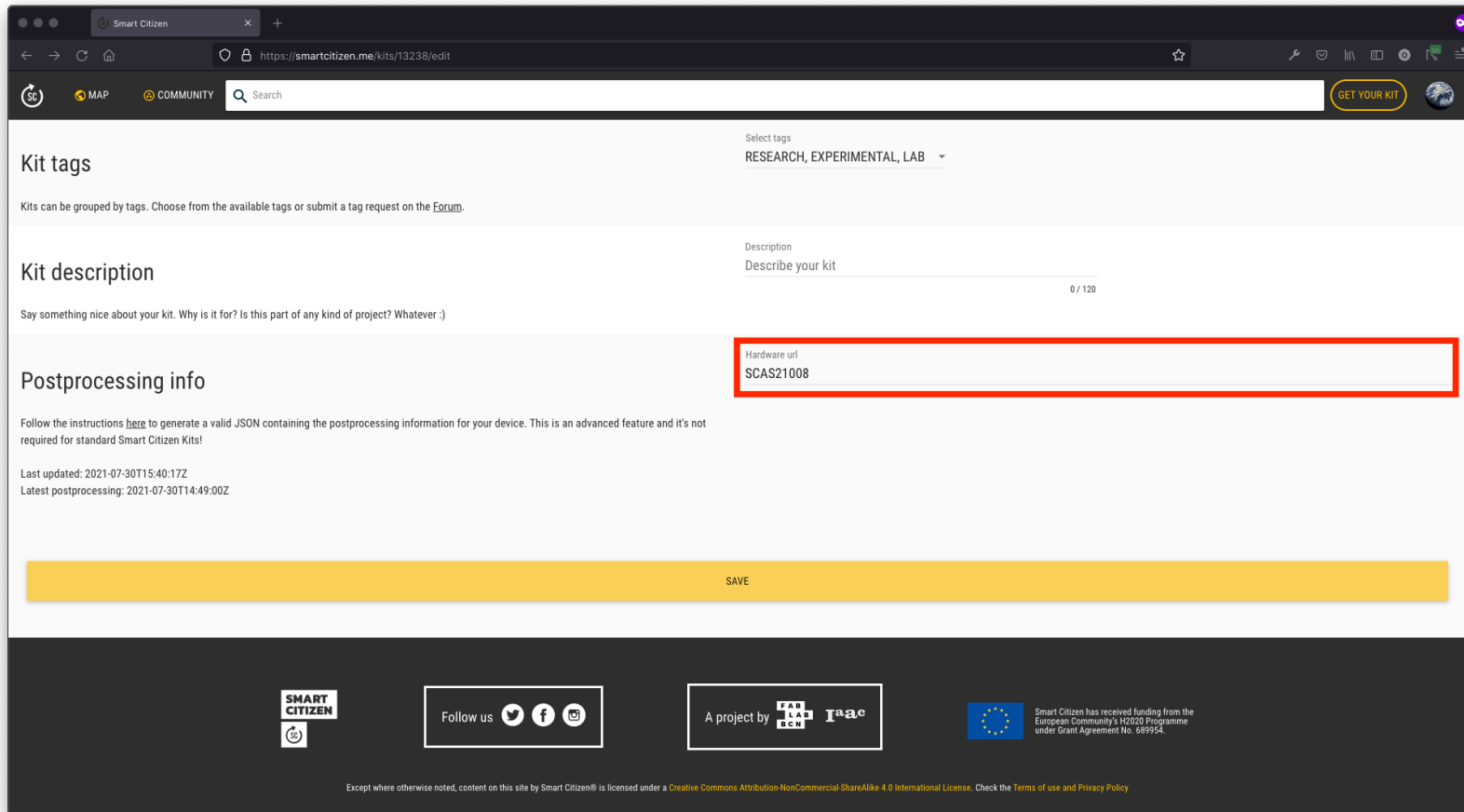
Processing data

How does this look in practical terms?



Processing data

How does this look in practical terms?



The screenshot shows the 'Smart Citizen' web interface for editing a kit. The browser address bar shows 'https://smartcitizen.me/kits/13238/edit'. The page has a dark header with navigation links: MAP, COMMUNITY, and a search bar. A 'GET YOUR KIT' button is in the top right. The main content area is divided into sections: 'Kit tags' with a dropdown menu set to 'RESEARCH, EXPERIMENTAL, LAB'; 'Kit description' with a text area and a character count '0 / 120'; and 'Postprocessing info' with a text area. A red rectangular box highlights the 'Hardware url' field, which contains the text 'SCAS21008'. Below the text areas is a large yellow 'SAVE' button. The footer contains the 'SMART CITIZEN' logo, social media links, a project logo 'A project by I²A²c', the European Commission logo, and a funding notice: 'Smart Citizen has received funding from the European Community's H2020 Programme under Grant Agreement No. 689954'. A small legal notice at the bottom states: 'Except where otherwise noted, content on this site by Smart Citizen® is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. Check the Terms of use and Privacy Policy'.

Smart Citizen

https://smartcitizen.me/kits/13238/edit

MAP COMMUNITY Search GET YOUR KIT

Select tags
RESEARCH, EXPERIMENTAL, LAB

Kit tags

Kits can be grouped by tags. Choose from the available tags or submit a tag request on the [Forum](#).

Kit description

Description
Describe your kit
0 / 120

Say something nice about your kit. Why is it for? Is this part of any kind of project? Whatever :)

Postprocessing info

Hardware url
SCAS21008

Follow the instructions [here](#) to generate a valid JSON containing the postprocessing information for your device. This is an advanced feature and it's not required for standard Smart Citizen Kits!

Last updated: 2021-07-30T15:40:17Z
Latest postprocessing: 2021-07-30T14:49:00Z

SAVE

SMART CITIZEN

Follow us

A project by I²A²c

EUROPEAN COMMISSION

Smart Citizen has received funding from the European Community's H2020 Programme under Grant Agreement No. 689954.

Except where otherwise noted, content on this site by Smart Citizen® is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#). Check the [Terms of use](#) and [Privacy Policy](#).

Processing data

When is this needed?

ONLY when you have something special going on:

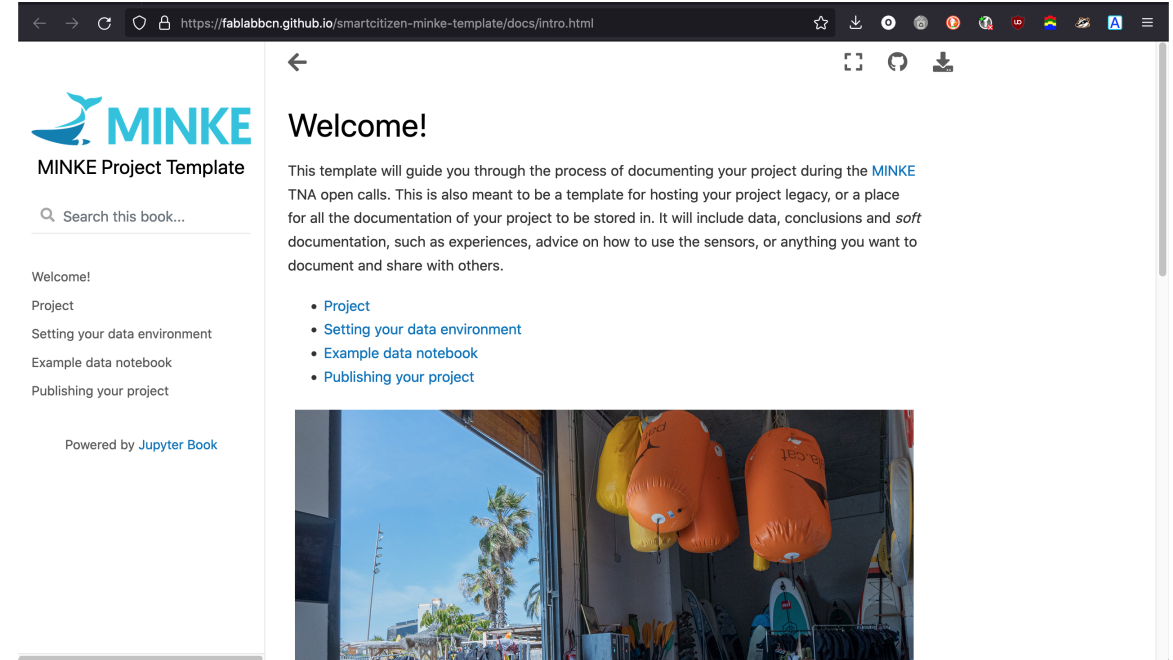
- either you have a **very specific algorithm** you want to apply
- or, you have sensors that **need calibration and the calibration can't be stored on the sensor itself** (water sensors don't need this!)

Legacy

Legacy

Why is this important?

- We want to have a **digital presence** of your projects
- We want to be able to show that **fully reproducible science** is possible
- We want to test if this is all **useful and practical**



Legacy

Structure

- You have **content** pages that can have documentation about the project you are doing
- Some **pages about the sensor deployment, setup and issues found in the process**
- **Data** is also part of the process website, with analysis, and conclusions
- All **public**!



MINKE Project Template

 Search this book...

Welcome!

Project

Setting your data environment

Example data notebook

Publishing your project

Powered by [Jupyter Book](#)

Legacy

First steps

- You need to have a github account (other platforms are OK if you rather not use github)
- Now visit the template and let's **follow the steps in the setup section**
<https://fablabbcn.github.io/smartcitizen-minke-template/docs/intro.html>

And on the next session...

And on the next session...

Visual programming

Other projects