



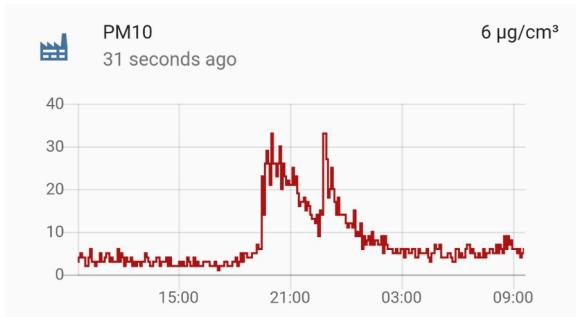
*An introduction to citizen data science
with Home-Assistant*

Dr Robin Cole & Mr Oliver Crask

PyLondinium 9/6/2018

Talk outline

- What are smart-home hubs?
- Introduction to Home-Assistant
- A platform for citizen science
- Case studies:
 1. Bird identification
 2. Personalized air-quality monitoring (Oliver)



'Smart' home hubs

- All kinds of connected devices entering the home
- A hub centralises control of these devices
- BUT commercial hubs such as Alexa are pretty locked down, not easy to access their data, & privacy concerns



Learning thermostat



The potential of a true smart hub

- Automated routines to save you time/energy/money
- Non invasive monitoring e.g. of elderly
- A home that knows how you like your toast (learning!)
- Your ideas here...
- We require a suitable platform to experiment



Introducing Home-Assistant

- An open source smart-home hub, python 3, local – typically raspberry pi
- Over 14,600 stars on [Github](#)
- Vibrant community ~ 60k active users

The screenshot shows the Home Assistant website at <https://home-assistant.io>. At the top, there's a browser header with a lock icon, the URL, and various icons. Below it, the main navigation bar includes links for Getting started, Components, Docs, Examples, Developers, Blog, and Need help? with a search icon.

The main content area features a large smartphone displaying the Home Assistant mobile application. The app's interface includes a top navigation bar with 'HOME' and 'OTHER' tabs, a sidebar menu, and a central dashboard showing various status cards. One card displays weather information for Sun, Anne Therese, and Paulus. Another card shows humidity and temperature for 20.95°C and 15.6°C. Below this, a section for the 'Living Room' shows two light switches labeled 'Bowl' and 'Ceiling'.

On the right side of the website, a large white text box contains the slogan 'Awaken your home'. Below it, a descriptive paragraph explains that Home Assistant is an open-source home automation platform running on Python 3, designed to track and control all devices at home and automate control, perfect for a Raspberry Pi.

At the bottom, there are three calls-to-action: 'GET STARTED', 'VIEW DEMO', and 'BROWSE CODE ON GITHUB'.

What can Home-Assistant do?

- Connect to over 1000 services & devices
- Control and automate

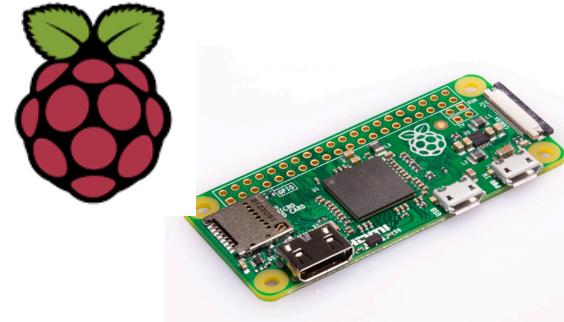
 Alexa / Amazon Echo voice	 Arduino diy	 Belkin WeMo hub	 Dark Sky weather
 ecobee hub	 Google Assistant voice	 Google Cast media-player	 IFTTT automation
 IKEA Trådfri (Tradfri) hub	 Kodi media-player	 MQTT hub	 MySensors hub
 Nest hub	 OwnTracks (via HTTP) presence-detection	 Philips Hue hub	 Philips Hue Light light
 Plex media-player	 Pushbullet notifications	 Sonos media-player	 Wink hub

What do I need to get started?

Your Cart

What a bounty! Make sure that everything in your cart is ship-shape, then click the checkout button to proceed.

	1	Raspberry Pi Zero W RPI-019	£9.16
	1	Official Raspberry Pi Zero Case RPI-020	£6



Item Total

£15.16
£12.63 ex. VAT

Shipping is calculated during checkout based on weight, destination, and service selected.

Add your own power supply and SD card.

Setup:

1. Head to <https://www.home-assistant.io/hassio/installation/>
2. Flash disk image to the SD card
3. Edit text file on the card to add wifi credentials
4. Insert card into pi, wait 20 mins, head to <http://hassio.local:8123/>
5. Let auto-discovery find your devices
6. Start adding extra services, customizing, automating

What is the UI?

- Browser GUI

The screenshot displays the Home Assistant web interface with the title "GooseCreek Drive". The left sidebar includes links for States, Map, History, Logbook, Log Out, Streaming updates (which is turned on), Developer Tools, and several icons. The main content area is organized into several cards:

- _GooseCreek Home**: Contains a switch for "Living Room" (off), a switch for "Kitchen Fan" (on), a switch for "Bedroom" (off), and two "Activate" buttons for "Rainy Day Lights" and "TV Time".
- Living Room**: Contains three light switches: "Couch_front" (off), "Couch_back" (off), and "Couch_Jamp" (off).
- Bedroom**: Contains a light switch for "Bedroom_lamp" (off).
- Family**: Shows two people: Carlo (Home) and Stacey (Home).
- Kitchen Fan**: Contains a switch for "Kitchen Full Bright" (on) and three light switches for "Fan_bulb_1", "Fan_bulb_2", and "Fan_bulb_3" (all on).
- Devices**: Lists "Hue Hub" (Online), "Tablo TV" (Online), and "Wii" (Offline).
- Internet**: Shows "Download" and "Upload" metrics with "Unknown Mbit/S" values.
- Bedroom Roku**: Shows "Idle" status and a "Default Screensaver".
- HD Poster**: Shows "Living Room Roku" playing "Roku Digital Clock".

At the bottom right, there is a blue circular button with a play icon and a small blue arrow pointing up.

Or create custom UI..!

<https://github.com/CCOSTAN/Home-AssistantConfig>

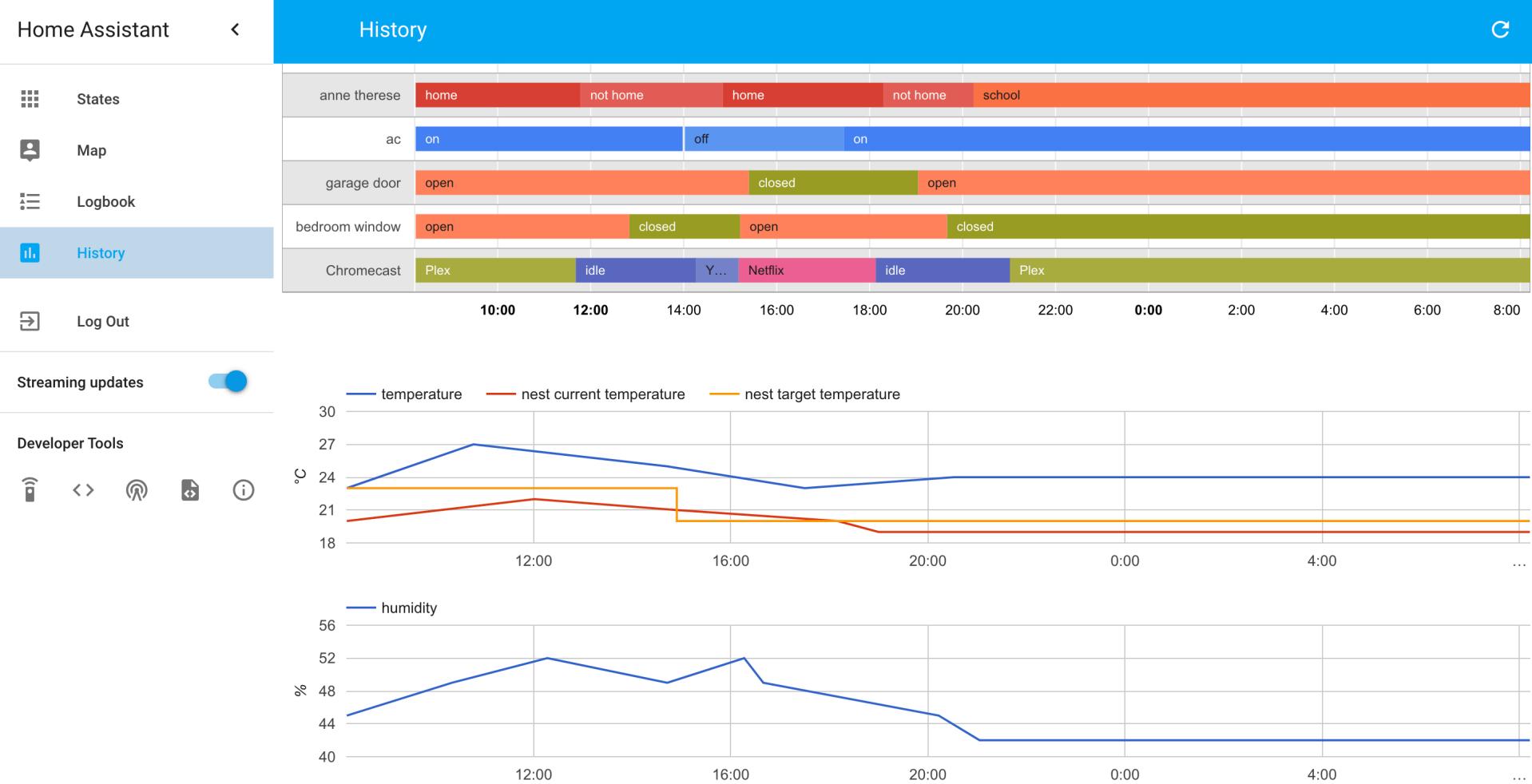


REFRESH

powered by 

How do I view data?

Graphs and charts built in



Extend Home-Assistant with 1-click install Addons

Many community add-ons, e.g. influxDB/Grafana <https://github.com/hassio-addons>



Talking of addons, JupyterLab addon is in development

<https://github.com/hassio-addons/addon-jupyter>

The screenshot shows the Home Assistant web interface with the Jupyter Lab addon installed. The left sidebar includes links for Overview, Map, Logbook, History, Hass.io, Jupyter Lab (which is selected), Pi-hole, Terminal, Configuration, and Log out. Below this is a section for Developer tools with icons for code editor, diff, network, file manager, and info. The main content area has a blue header "Jupyter Lab". On the left, there's a file browser showing an "Untitled Folder" and two files: "Lorenz.ipynb" (modified 2 minutes ago) and "lorenz.py" (modified 4 minutes ago). The main pane displays a Jupyter notebook cell titled "The Lorenz Differential Equations". It contains the following text and code:

Before we start, we import some preliminary libraries. We will also import (below) the accompanying `lorenz.py` file, which contains the actual solver and plotting routine.

```
In [2]: %matplotlib inline
from ipywidgets import interactive, fixed
```

We explore the Lorenz system of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

```
In [4]: from lorenz import solve_lorenz
w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
w
```

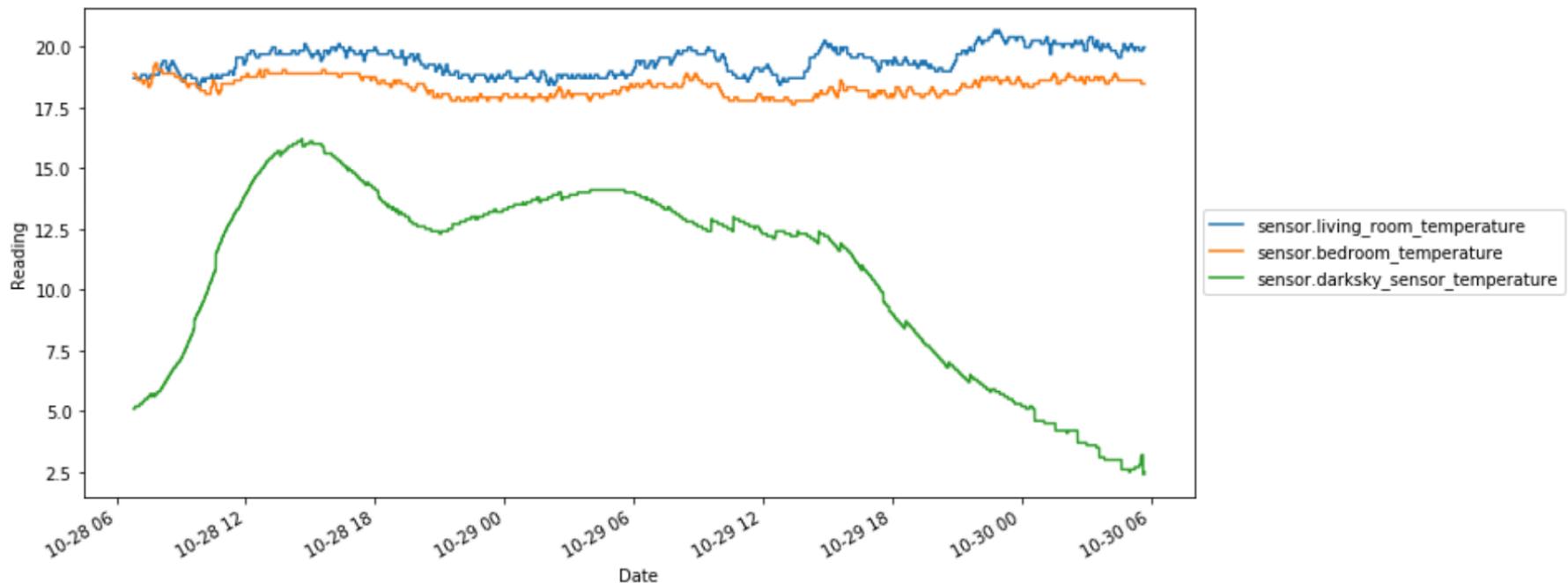
Below the code cell is a 3D plot of the Lorenz attractor, showing two points of rotation.

For the default set of parameters, we see the trajectories swirling around two points, called attractors.

The object returned by `interactive` is a Widget object and it has attributes that contain the current result and arguments:

What is the backend database?

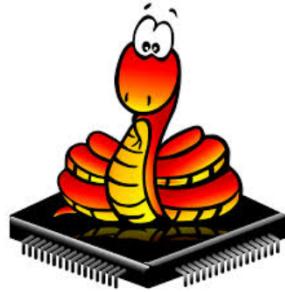
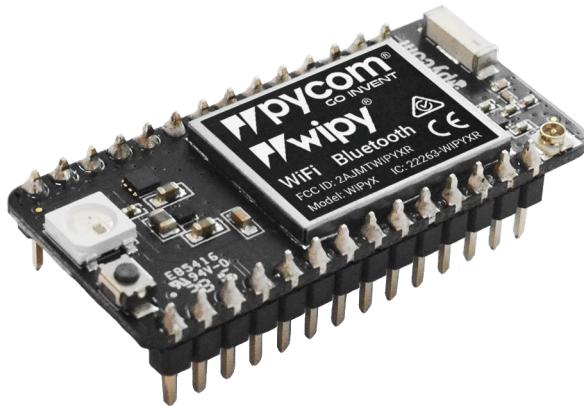
- All state and event data in a single, time series SQL database
- Use standard tools to query your data
- Library for working with data built with Pandas -> Hass-data-detective
- Even upload the sqlite.db file to Google CoLaboratory (online jupyter) for analysis!



<https://github.com/robmarkcole/HASS-data-detective>

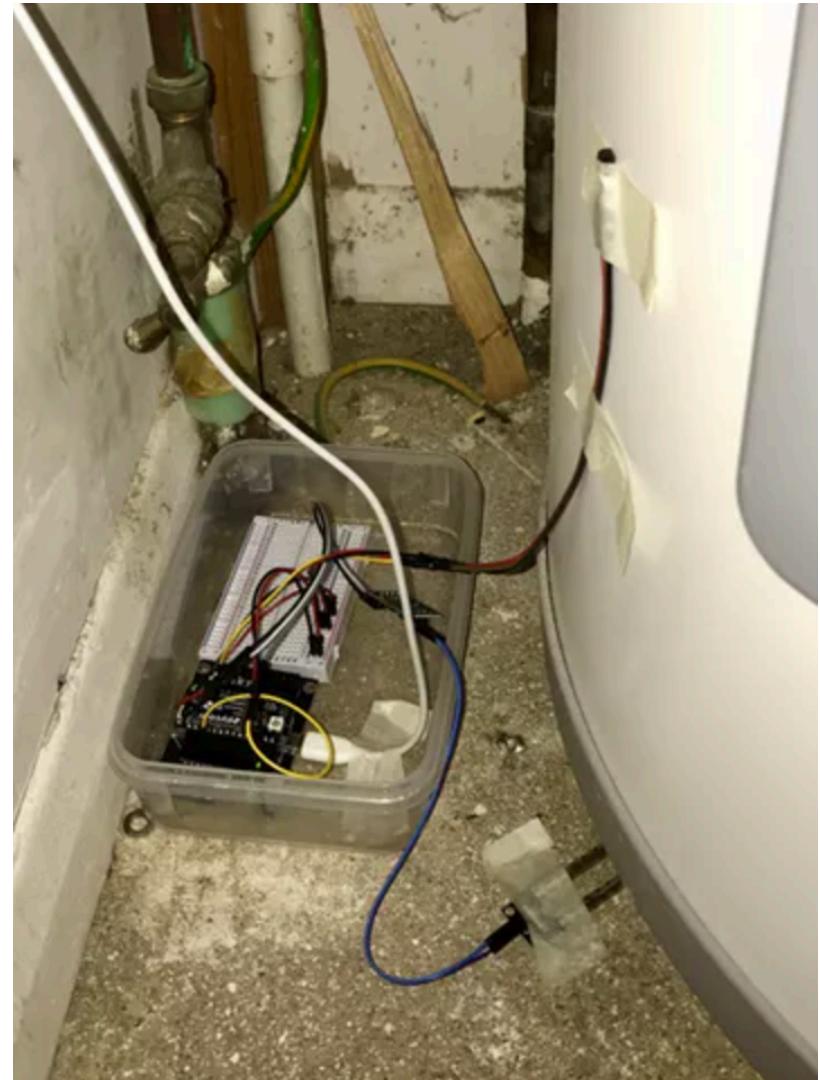
Getting data into Home-Assistant from Micropython

- Physical connection
- Rest API
- MQTT



Wipy

	Date & Time	2017-05-13, 12:52
	Temperature	22.375 Celsius
	Moisture	3548.18



<https://www.hackster.io/robin-cole/micropython-leak-detector>

Citizen science case study 1: Bird identification

Project goal: automatically photograph and identify birds visiting my bird feeder, gather data that I can share with birdlife study <https://www.rspb.org.uk/birdwatch>



Motion triggered £10 camera & Pi3 Home-Assistant

Task 1: identifying bird/not-bird images

MACHINE BOX

For a simple to use classifier I used <https://machinebox.io/docs/classificationbox>

```
Robins-MacBook-Air:birds_vs_not_birds robincole$ imgclass
```

```
Classes
```

```
-----
```

```
birds: 501 image(s)  
not_birds: 514 image(s)
```

```
Create new model with 2 classes? (y/n): y
```

```
new model created: 5b0bfba7cb439f4b
```

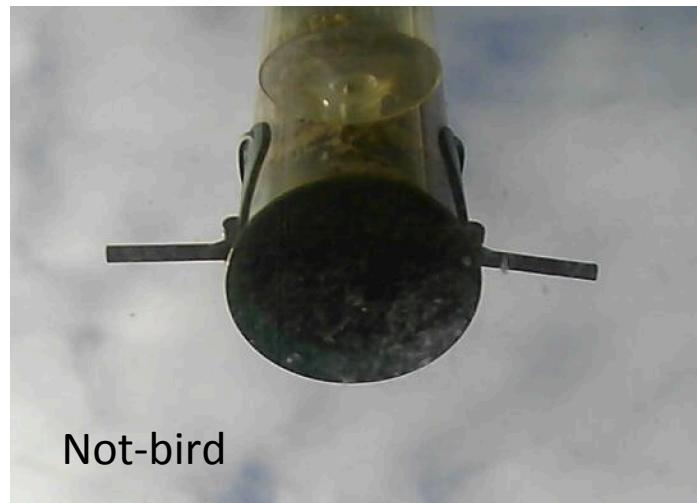
```
Teach and validate Classificationbox with 812 (80%) random images? (y/n): y  
pass 1 of 1...
```

```
447 / 812 [=====
```

```
Correct: 188  
Incorrect: 15  
Errors: 0  
Accuracy: 92.61083743842364%
```



Bird

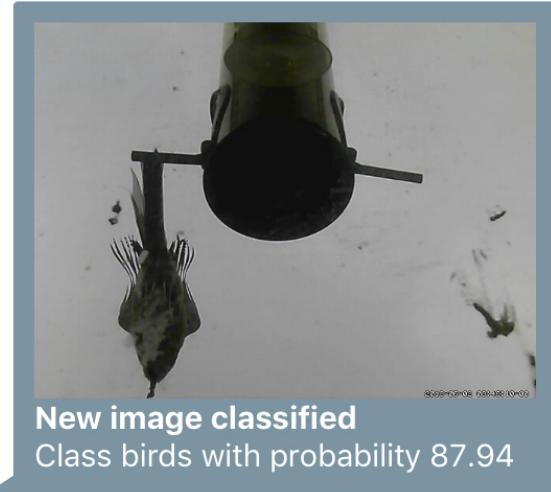


Not-bird

- Trained classifier with 1000 images
- bird/not-bird discrimination with ~ 90% accuracy,
beats my expectations!

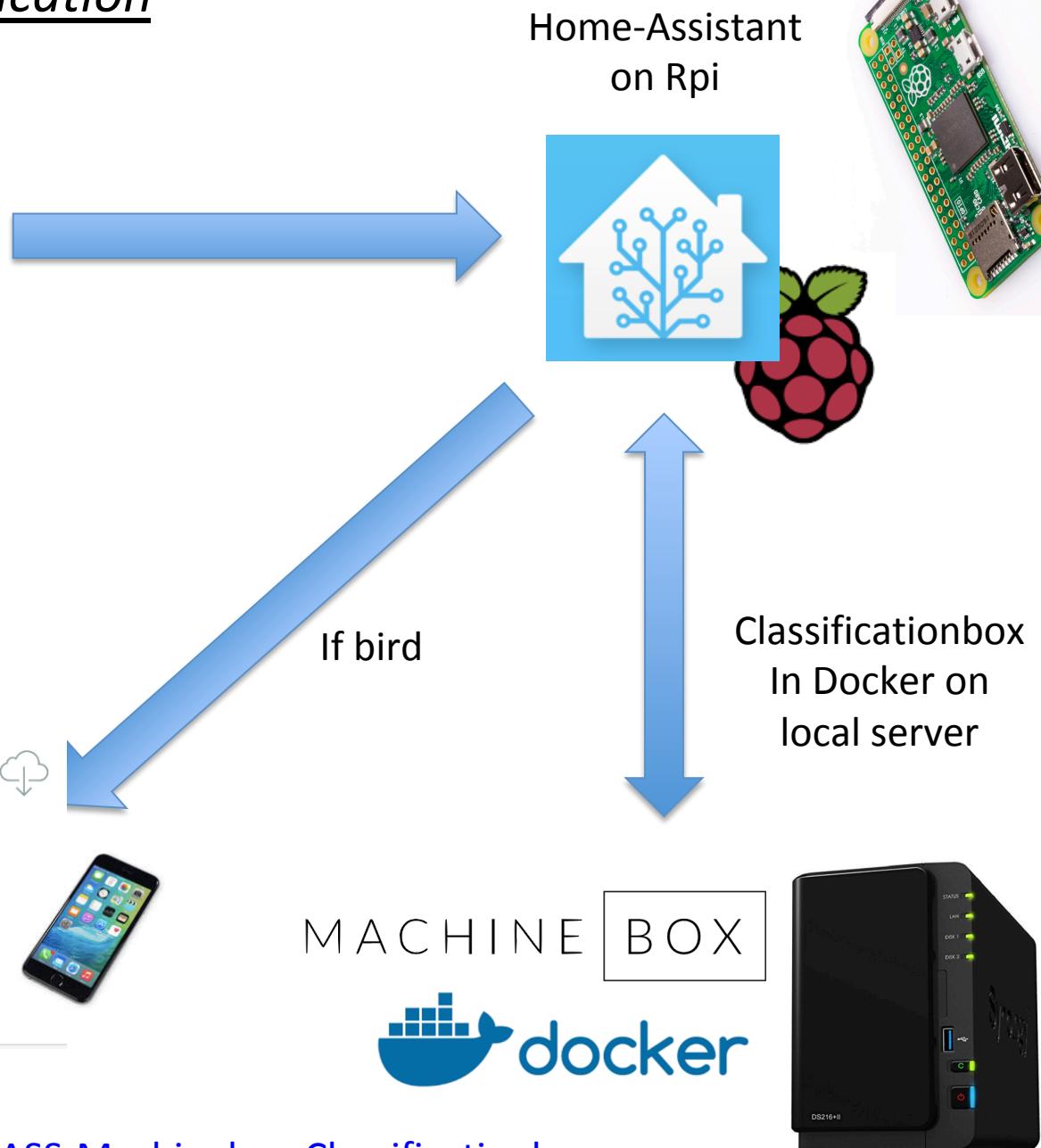
Task 2: Automating identification

Motion triggered
USB Image capture



8:47 am

Notification with image



Next steps:

1. Experiment with curating the images used in training to improve accuracy
2. Classify birds by species
3. Contribute data to Birdwatch survey
4. Add Classificationbox to Home-Assistant
5. Share classifiers and know-how with less technical bird-fanciers
6. Make bird-feeder magpie proof!



Thank you!

Now, over to Oliver