University of St Andrews | FOUNDED 1413

Engineering a TinyML system for use off-grid

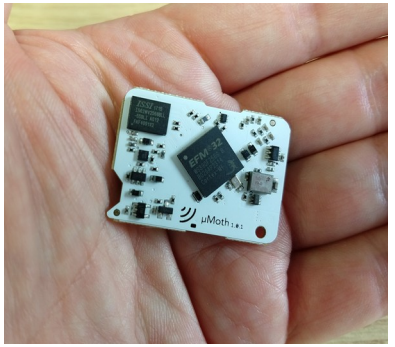
Simon Dobson
School of Computer Science and Institute of Engineering
University of St Andrews UK

simon.dobson@st-andrews.ac.uk
https://simondobson.org

The challenge

- Monitoring birds is important for identifying healthy ecosystems
  - Expensive and error-prone to do manually, especially if dense in space and time

- State of the art: data loggers
  - Control times, *i.e.*, don't record during the hours of darkness
  - Still need to collect the data and process it offline
  - Songs often identified by trained human analysts



AI on edge devices

- Machine learning on small devices (“TinyML”)
  - *E.g.*, wake-word detection for voice-activated assistants
  - Typically models with around 150K parameters
- We want to collect bird observations, not songs
  - Place sensors off-grid, classify on-device, report back
  - Use the standard BirdNET model: recognises the calls of 6,000+ species but uses 40M parameters, so not exactly “tiny” ML…

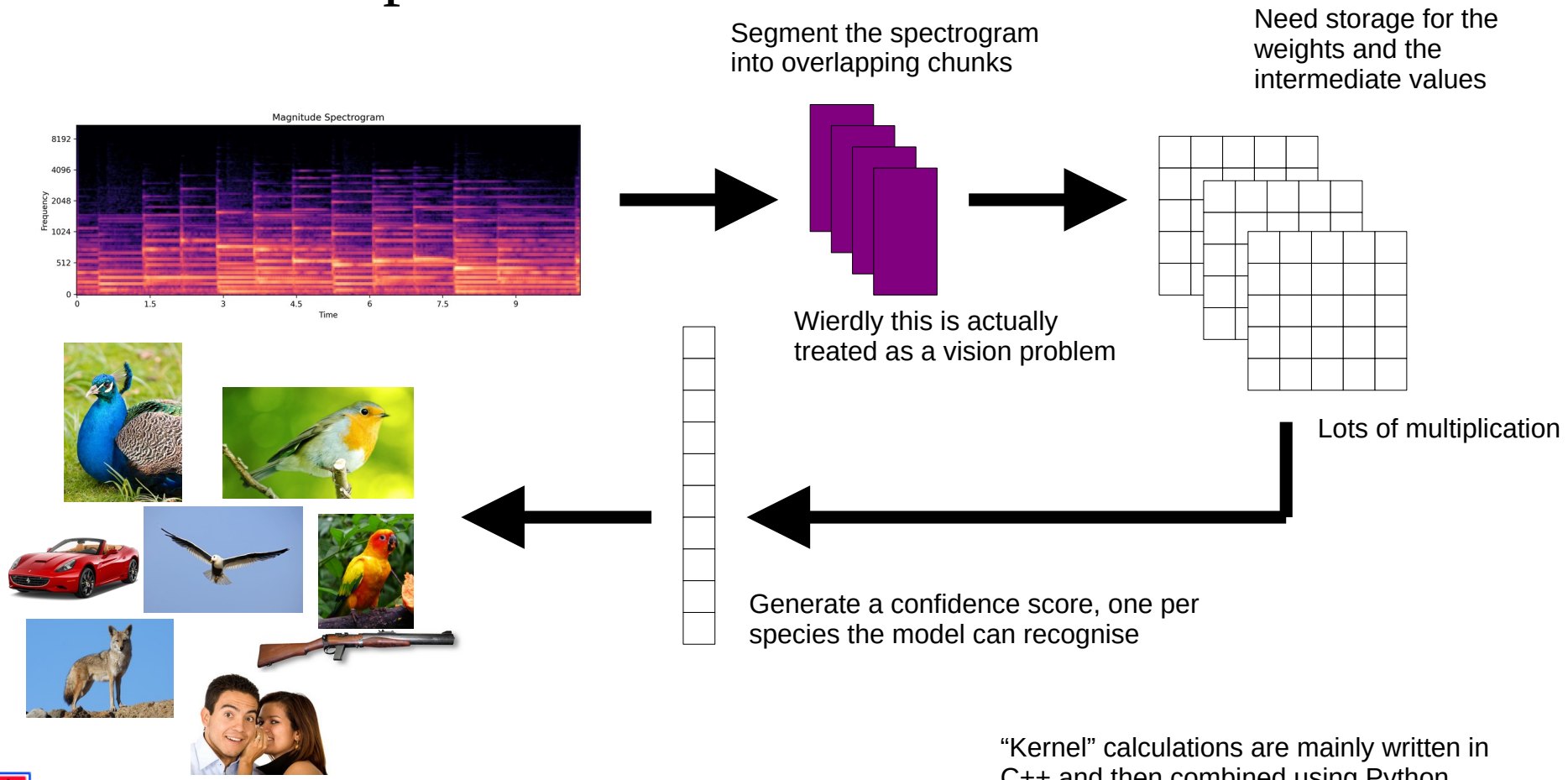
- Can we engineer an appropriate sensor?
  - Off-grid, no infrastructure, off-the-shelf hardware

The engineering challenges

1. Compute
   - Store the weights for a not-exactly-tiny model
   - Run the calculations
2. Communications
   - Long range, low power
   - Interact with wider infrastructure
3. Power
   - Rechargeable
   - Survive unattended for a senible period (*i.e.*, a season)

Brief digresssion: the software

- A machine learning model is just a system of array and tensor operations



Compute

- How large/small a machine do we need to represent and run the model?
  - Microcontrollers
    - Arduino Nano BLE Sense
    - Arduino Nano ESP32
  - Hybrids
    - Raspberry Pi Zero W
    - Raspberry Pi Zero 2 W
  - Single-board computers
    - Raspberry Pi 5

Nothing with a GPU, as they're brutally power-hungry

Compute

- How large/small a machine do we need to represent and run the model?
  - Microcontrollers
    - ~~Arduino Nano BLE Sense~~
    - ~~Arduino Nano ESP32~~
  - Hybrids
    - Raspberry Pi Zero W
    - Raspberry Pi Zero 2 W
  - Single-board computers
    - Raspberry Pi 5

Not enough RAM for the weights or the working tensors, and not a sufficiently powerful core to do the calculations

(The problem isn't using Python, as the core Tensorflow code is all in C++)

Compute

- How large/small a machine do we need to represent and run the model?
  - Microcontrollers
    - ~~Arduino Nano BLE Sense~~
    - ~~Arduino Nano ESP32~~
  - Hybrids
    - ~~Raspberry Pi Zero W~~
    - Raspberry Pi Zero 2 W
  - Single-board computers
    - Raspberry Pi 5

Not a sufficiently powerful core to do the calculations (takes 30s to do an inference step over a single 5s sound sample)

Compute

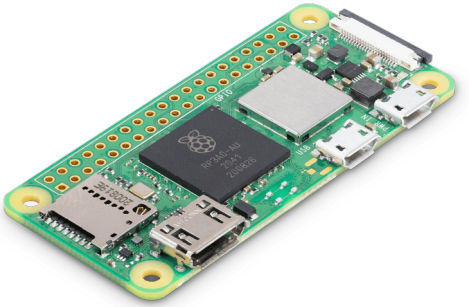
- How large/small a machine do we need to represent and run the model?
  - Microcontrollers
    - ~~Arduino Nano BLE Sense~~
    - ~~Arduino Nano ESP32~~
  - Hybrids
    - ~~Raspberry Pi Zero W~~
    - Raspberry Pi Zero 2 W
  - Single-board computers
    - ~~Raspberry Pi 5~~

Fast, lots of RAM, but draws about 7W of power when doing inference

Compute

- How large/small a machine do we need to represent and run the model?
  - Microcontrollers
    - ~~Arduino Nano BLE Sense~~
    - ~~Arduino Nano ESP32~~
  - Hybrids
    - ~~Raspberry Pi Zero W~~
    - Raspberry Pi Zero 2 W ♫♪♬ ♩
  - Single-board computers
    - ~~Raspberry Pi 5~~

1.4s for inference, draws 1.3W when doing it, and about 750mW when idle

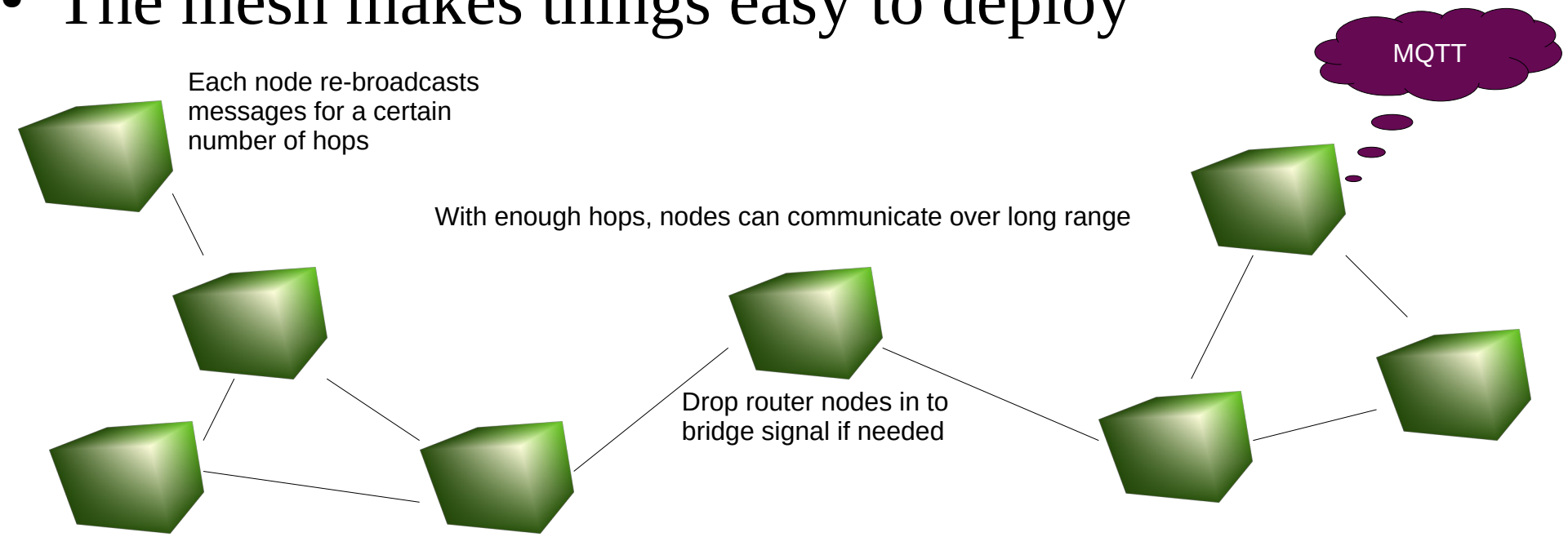


Communications

- MQTT is standard in the Internet of Things
  - Pub-sub to topics hosted by a broker
  - Sits on top of TCP, so needs networking infrastructure
- We decided to use it anyway
  - Simplicity of integration with other tools
  - ...and then deal with the underlying network separately
- Several choices
  - LoRa, LoRaWAN, Zigbee, … – all with pros and cons
  - ...and all with different, dedicated hardware

Communications

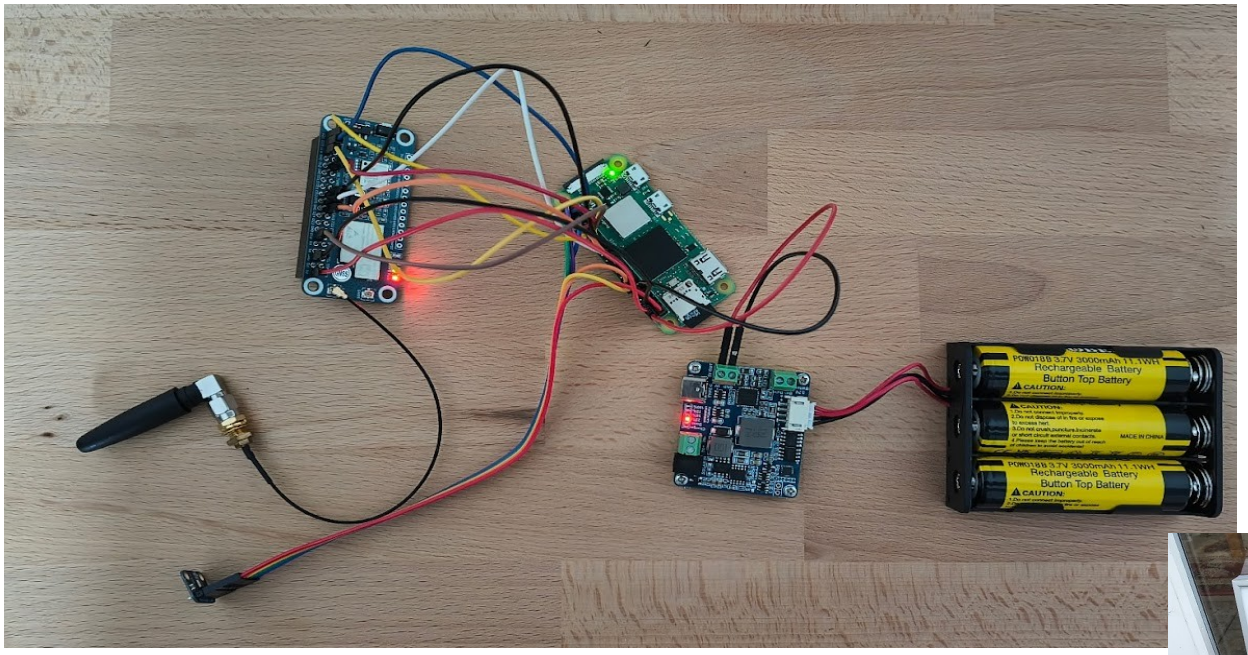
- Then we found Meshtastic
  - Sits on top of LoRa
  - Automatically forwards messages to MQTT
- The mesh makes things easy to deploy



Power

- Wind
  - Surprisingly unreliable, tends to ice-up and burn-out
- Impellors in water/hydro power
  - Gets clogged with weeds
- Solar
  - Have you *been* to Scotland??
- Realistically has to be solar
  - Power and recharge during daylight
  - ~10W panel feeding a 9,000mAh battery

Solar charge managers often don't provide an interface to get battery health, which we'd like for telemetry



Components

Protection from the elements

“Lo-fidelity prototype”

What we've learned

- No new science; lots of engineering
  - Find the trade-offs and work around them
  - Creative use of what's available
  - Easy to share with other groups
- Stripped-down the software
  - 500 lines of Python
  - Each task is a script, piped together, with network drivers
- Aiming to deploy in the Highlands in June