GREAT FUN WITH TINY ML

PÄR-ANTON WESTBOM

CADEC 2023.01.19 & 2023.01.25 | CALLISTAENTERPRISE.SE

CALLISTA

AGENDA

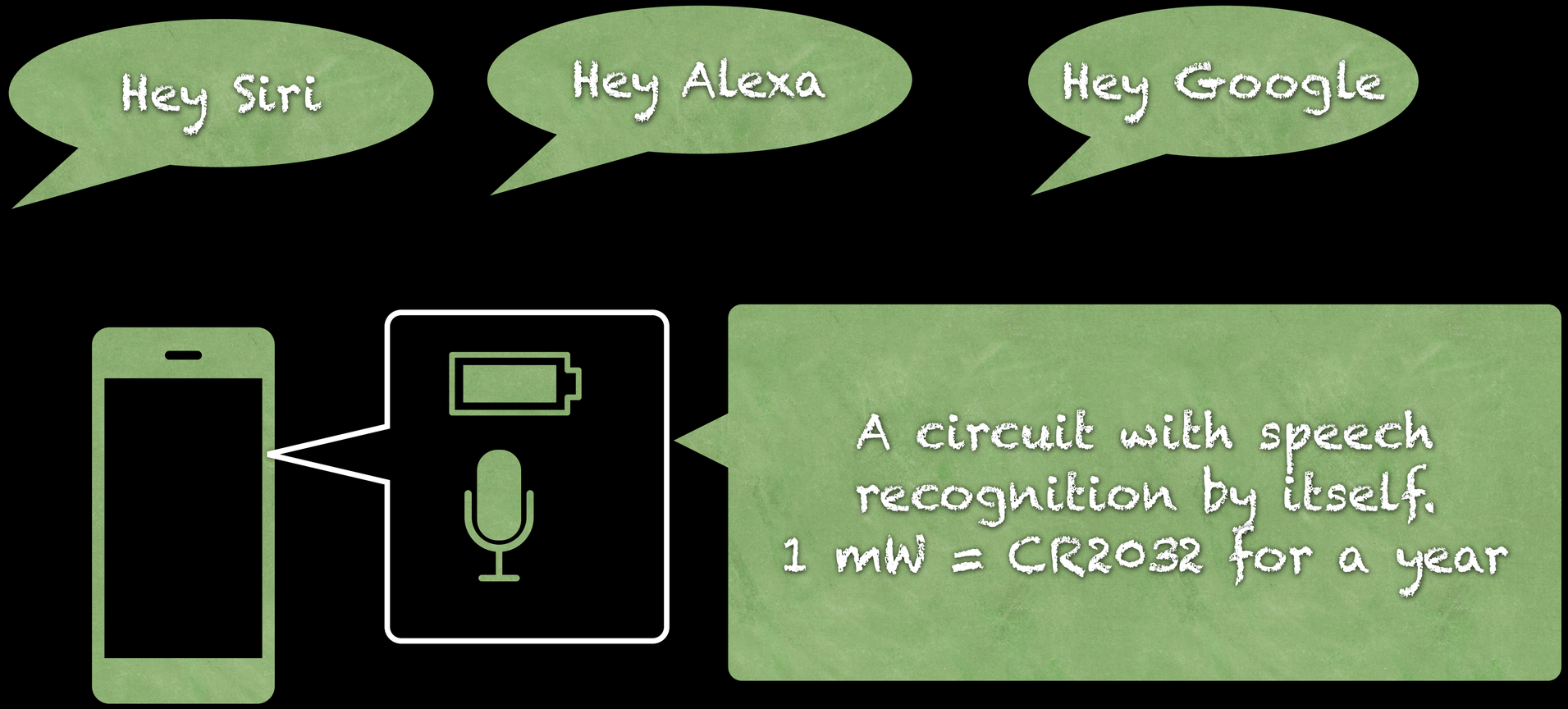
- What is TinyML (Tiny Machine Learning)
- Demo
- Conclusion

WHAT IS TINYML (TinyMachineLearning)

- Tiny refer to the device size that does the computation. Typically different sensors (camera, microphone, gyroscope, thermometer, bloodpressure, …) aka IoT devices
- ML stands for Machine Learning
- What makes it interesting
  - It is possible
  - There are open source initiatives to be used
  - It solves or helps with some challenges with todays approaches.

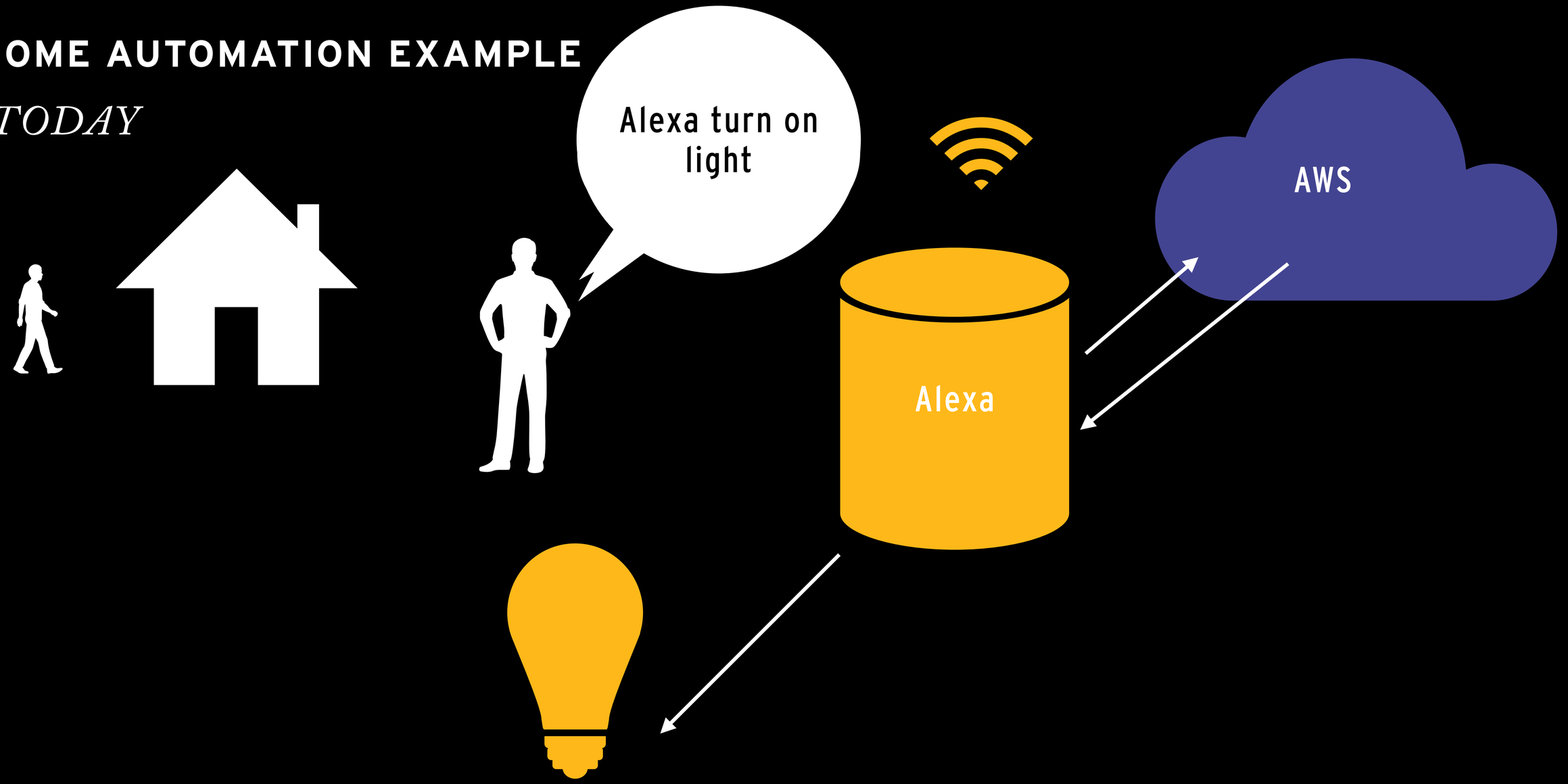
AN EXAMPLE OF TINYML TODAY

- Mobile wake up call



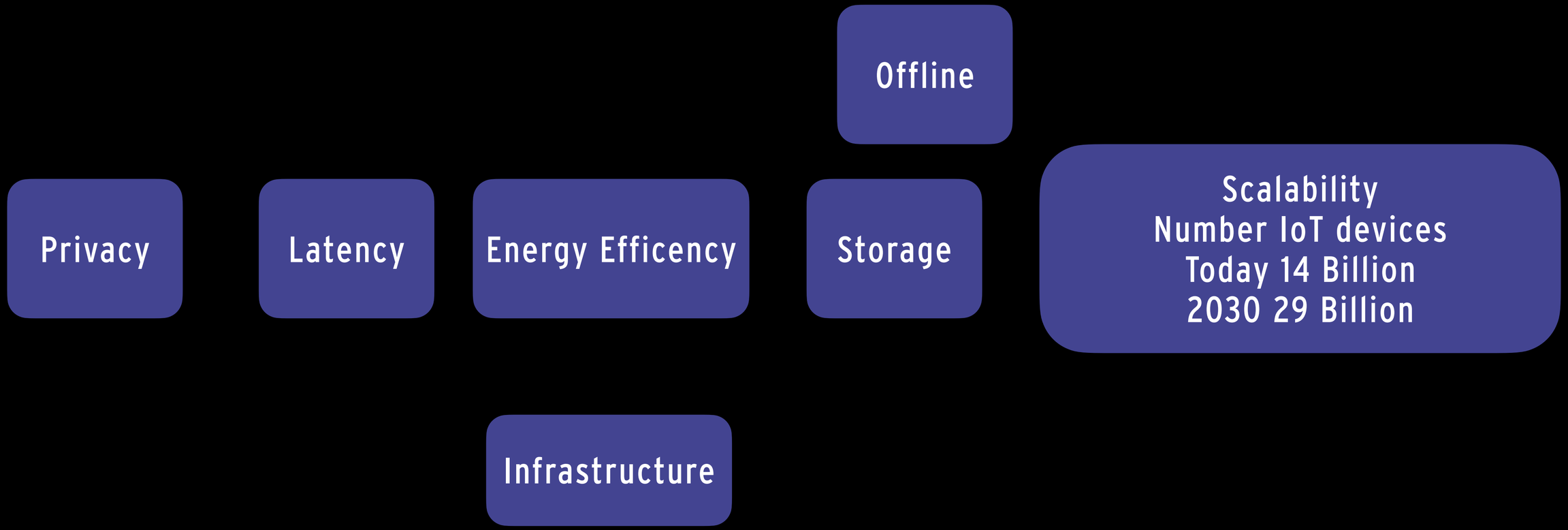
HOME AUTOMATION EXAMPLE

TODAY



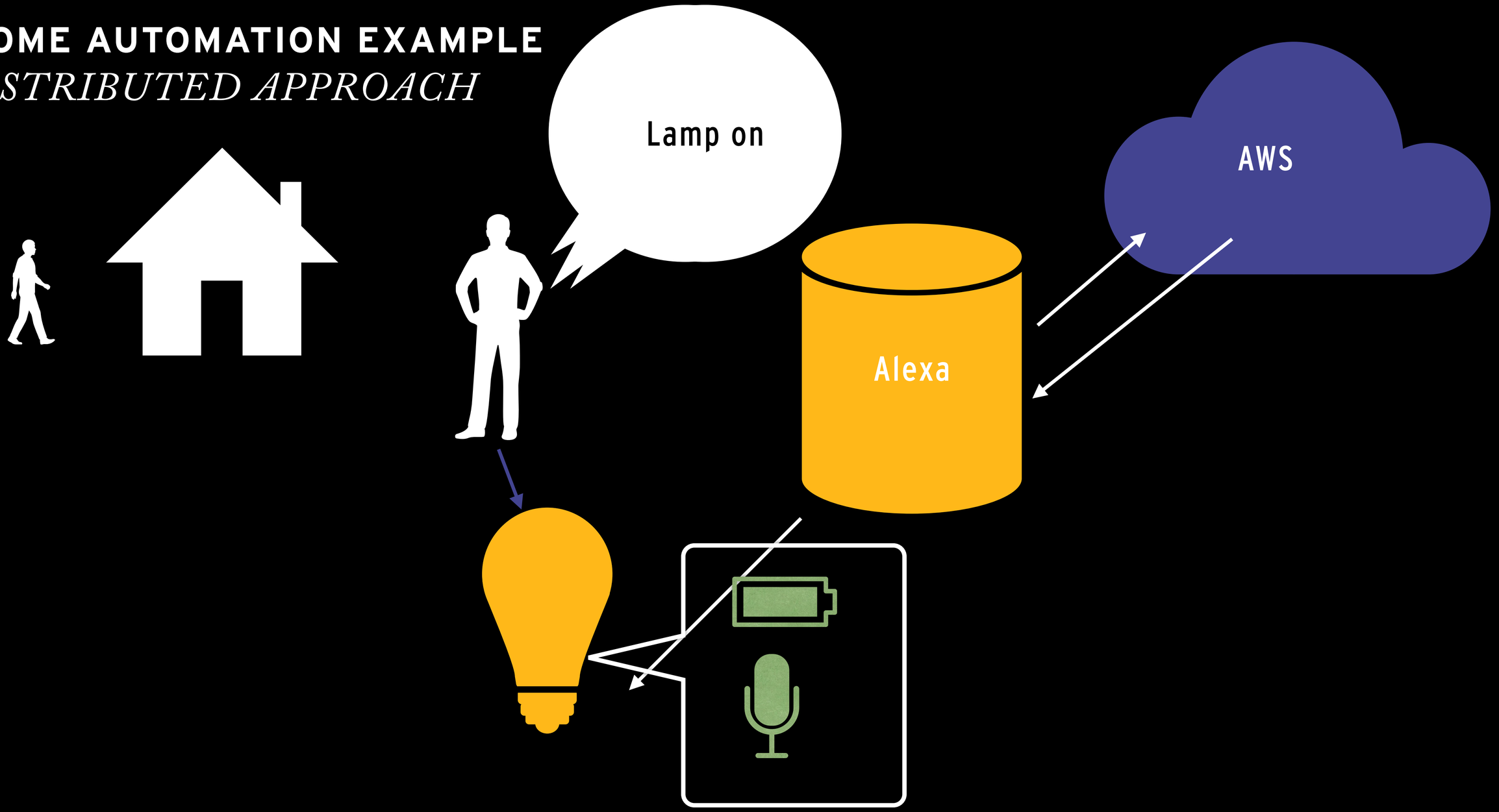
TODAY MOST DEVICES TODAY HAVE A CENTRAL APPROACH

- The traditional idea of IoT was to send data from a local device to the cloud for processing.
- There are some problems doing it this way



HOME AUTOMATION EXAMPLE

DISTRIBUTED APPROACH



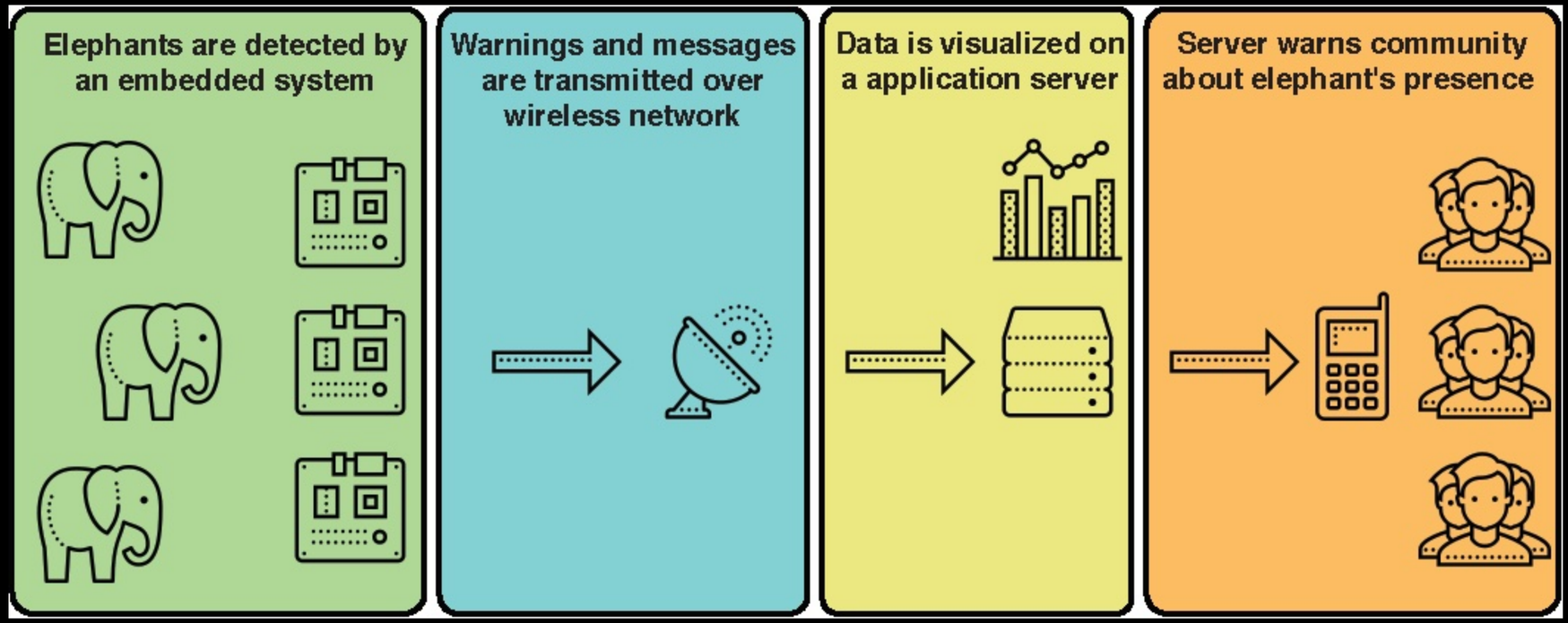
THERMAL CAMERAS

- Animal detection
- Human detection
- Traffic
- Privacy
- ….

https://www.weforum.org/agenda/2021/03/human-elephant-conflict-camera-solution-zsl/

HUMAN ELEPHANT CONFLICT (HEC)

https://www.weforum.org/agenda/2021/03/



https://www.irnas.eu/energy-efficient-system-for-detection-of-elephants-with-machine-learning/

THERMAL CAMERAS

- Animal detection
- Human detection
- Traffic
- Privacy
- ….

https://www.weforum.org/agenda/2021/03/human-elephant-conflict-camera-solution-zsl/

PROS DOING MORE THING ON THE DEVICE (DISTRIBUTED APPROACH)

- Energy Efficient it cost less energy to have an IoT device do its own computation. Transmitting raw data is energy intensive.
- Privacy & Security Keeping data on the device.
- Storage only keep data that are of interest.
- Latency
- Connectivity
- Infrastructure
- …

CHALLENGES DOING COMPUTATION AS PROCESSING ML MODELS ON SMALL DEVICES

- **Memory** (Low) 1MB Flash, 256 KB Ram
- **Power** (Limited) 3.3 V
- **Processing Power** (Limited) 64 Mhz
- **Troubleshooting** and updating of the software when distributed

TENSORFLOW LITE (2017) (OVER 2 BILLION DEVICES IN 2019)

PYTORCH MOBILE

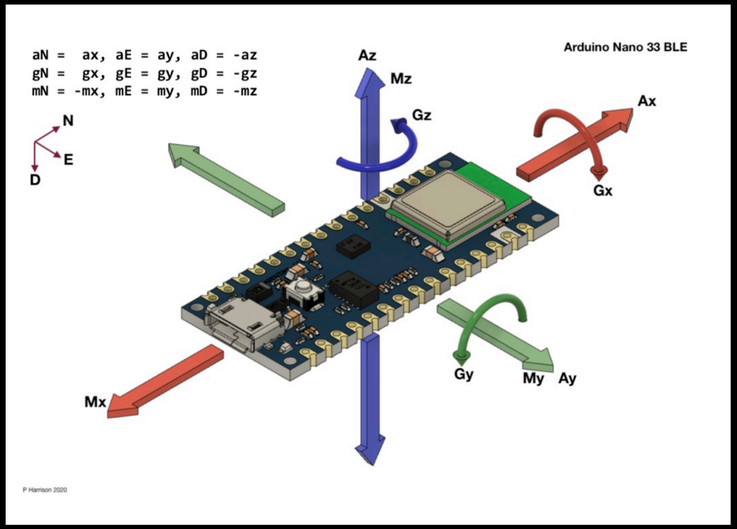
1. Tensorflow trains the model
2. Tensorflow Lite
    1. Convert the model
    2. Optimize the model
    3. A model that can be deployed to (Microcontrollers, Linux Embedded devices, Android, iOS)

THE STAGE IS SET FOR THE DEMO(S)

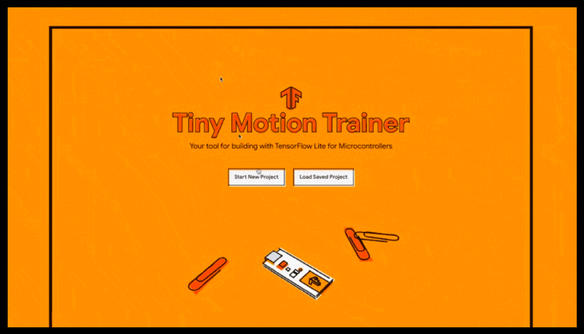
- We have set the context for TinyML and what it is all about. Let's explore the field practical.

In this experiment

- Arduino 33 BLE Sense
- Released Autumn 2019
- 1 MB Flash, 256 KB Ram, 3.3 V
- 64 Mhz Nina-b3
- 18 x 45 mm (Like an EarPod)
- Temp, pressure, microphone …
  Gesture Sensor, BLE (BlueTooth)
- All about gestures this time



TINY MOTION TRAINER (TMT) BY GOOGLE CREATIVE LABS

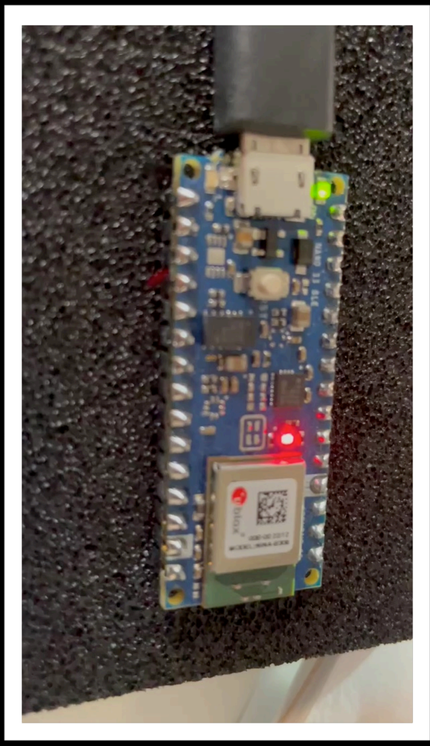


- Tiny Motion Trainer is a part of the TensorFlow Lite for Microcontrollers Experiments, a collection of open source interactive projects designed to demonstrate some fun ways to combine Arduino (the microcontroller) and TensorFlow Lite for microcontrollers.
- Projects using Tiny Motion trainer are build with Arduino Sense 33 BLE, TensorFlow Lite for microcontrollers and standard web technologies.
- https://github.com/googlecreativelab/tiny-motion-trainer

This is what we are going to try now !

FIRST STEP

- Before the Arduino has been setup to work with Tiny Motion Trainer (TMT) (Gyro, BlueTooth, TMT-lib)
- Connect our Arduino to (TMT)
- Collect data from the gyroscope to TMT
- Let TMT train a model on the data
- Download the trained model to our laptop



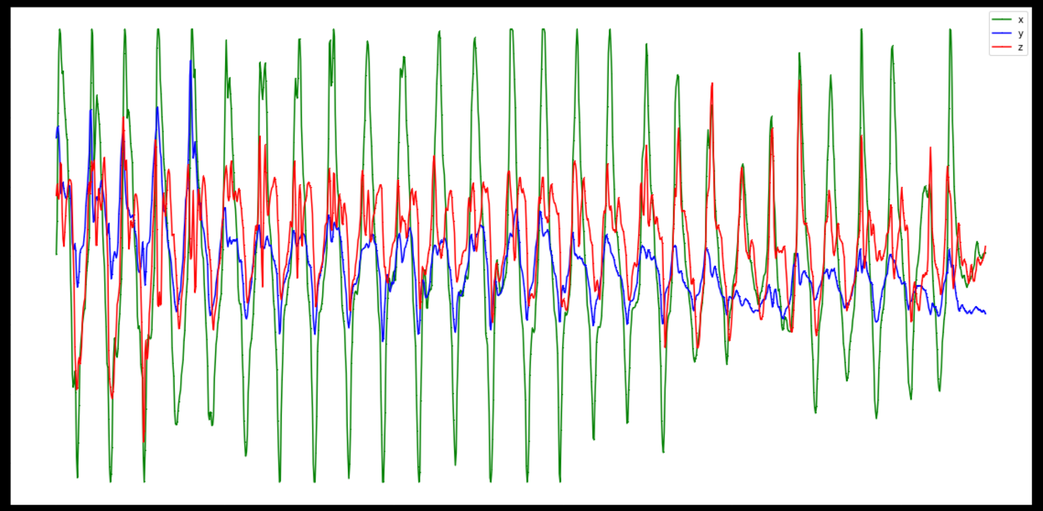
Experiment

COLLECT DATA

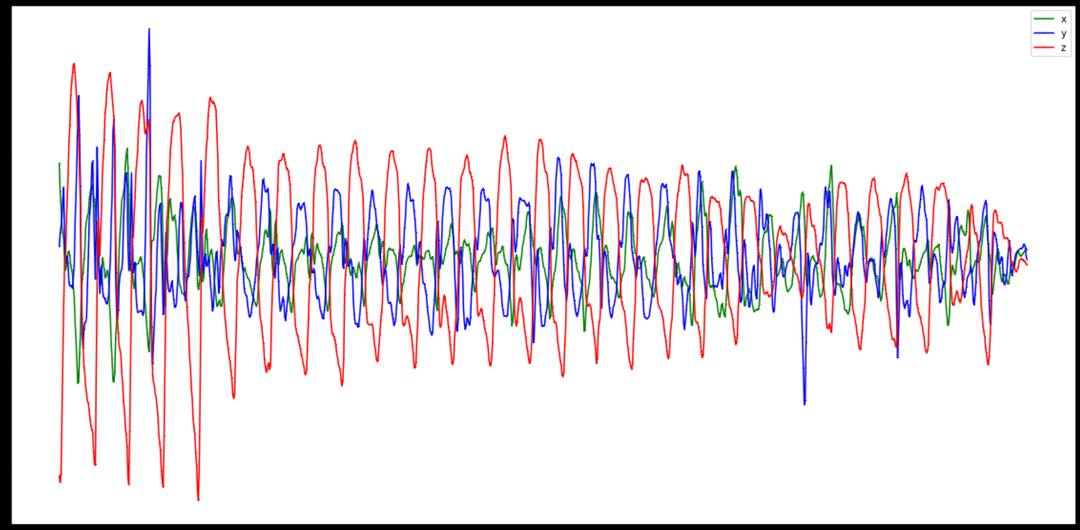
HTTPS://EXPERIMENTS.WITHGOOGLE.COM/TINY-MOTION-TRAINER/VIEW/

COLLECTED DATA (ACCELEROMETER AND GYROSCOPE)

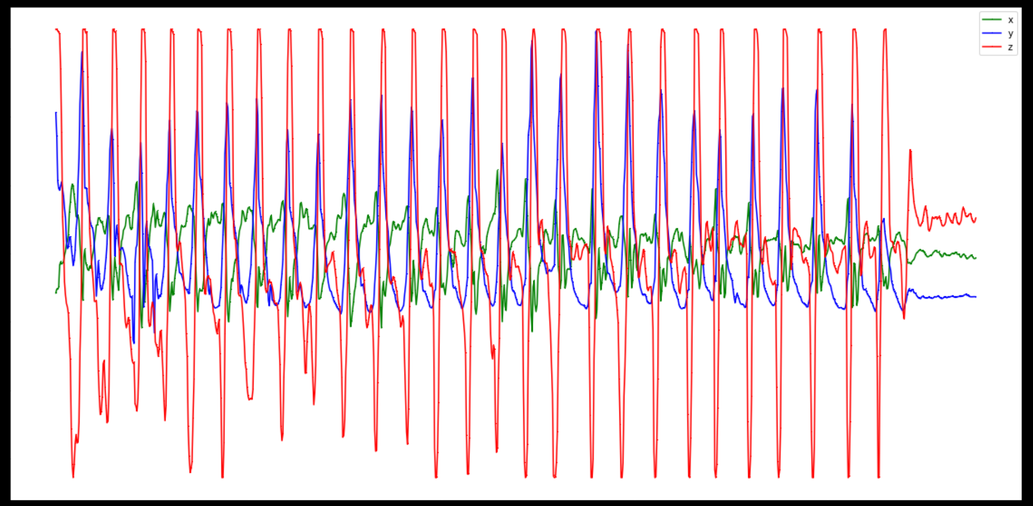
WAVE



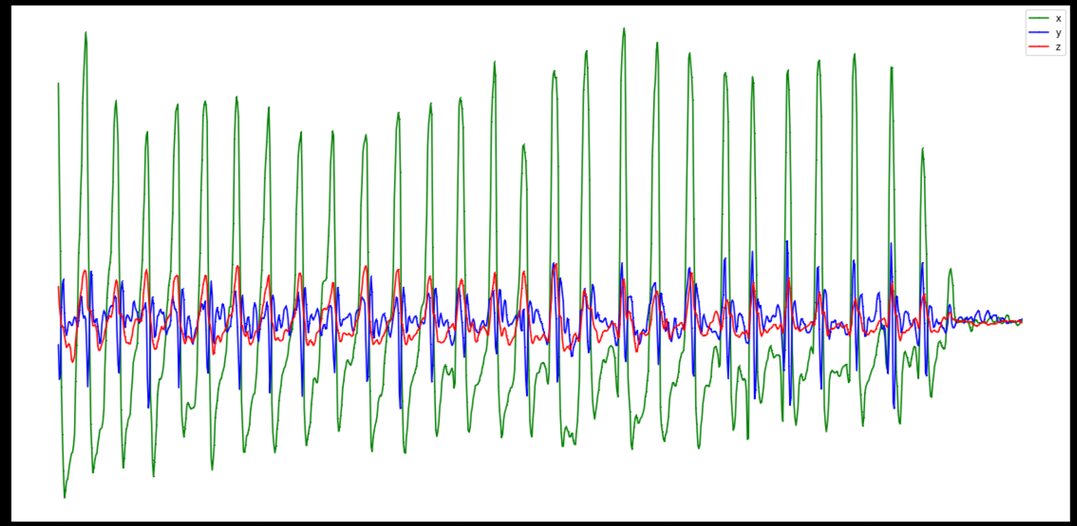
WAVE



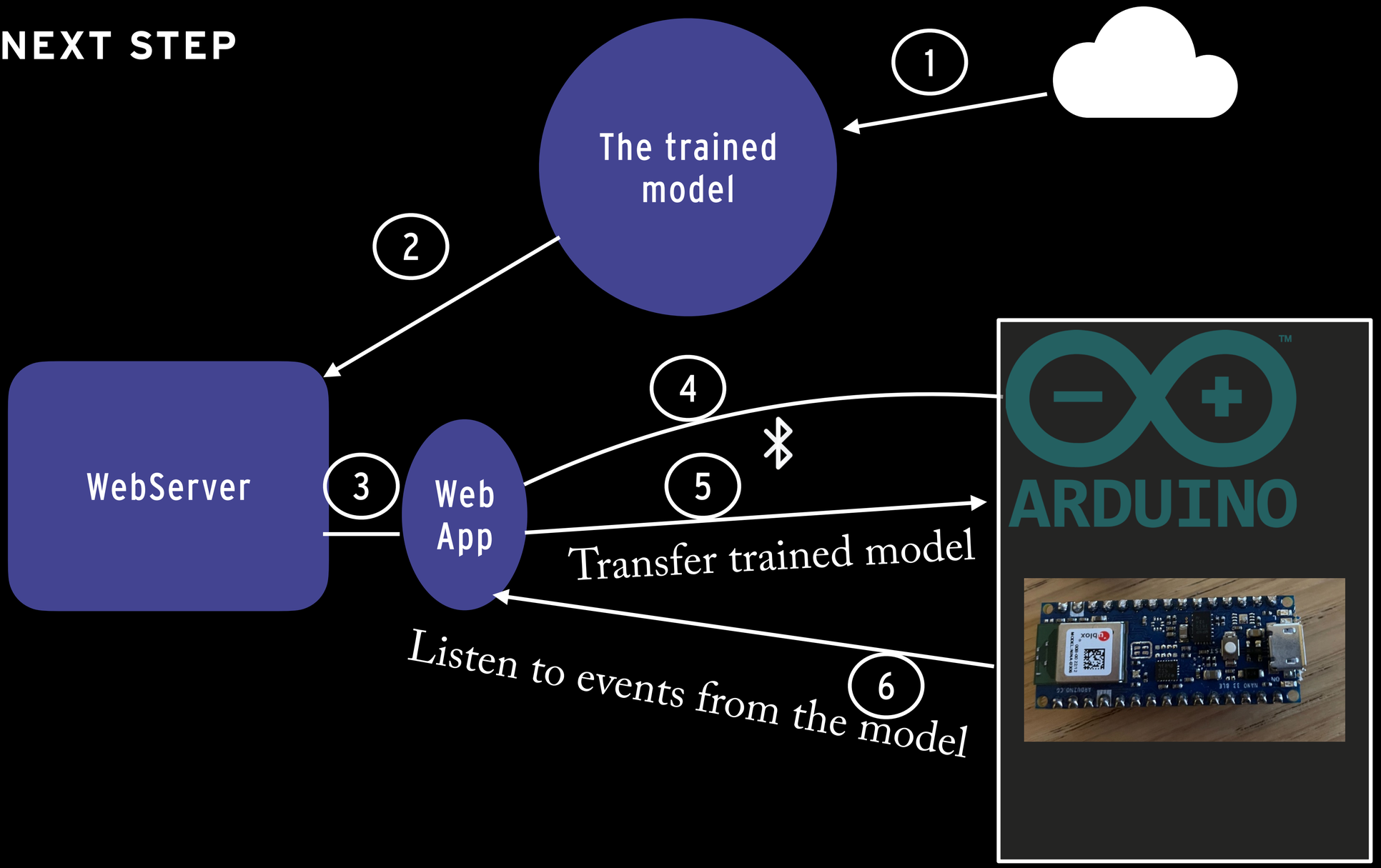
PUNCH



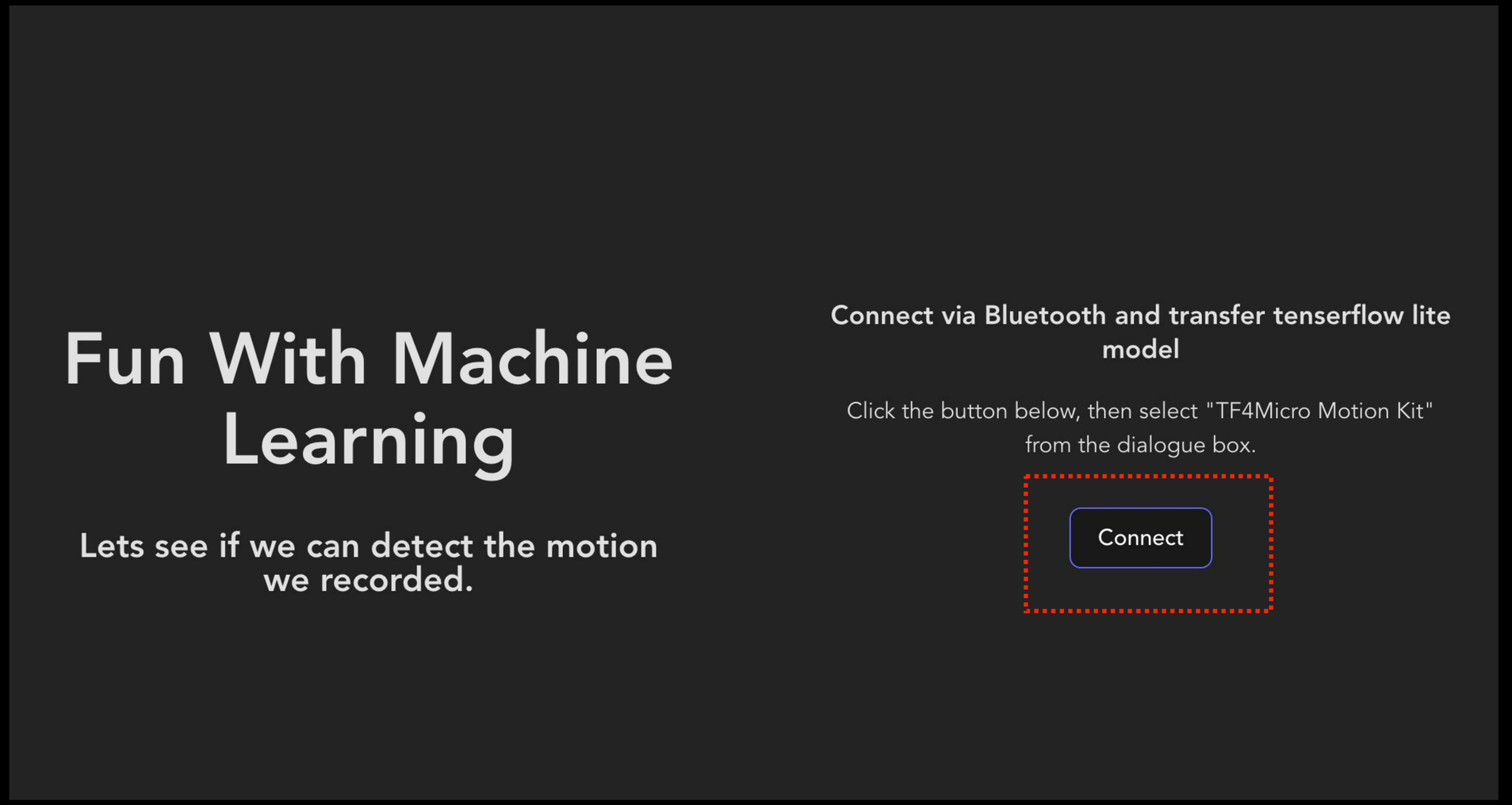
PUNCH



NEXT STEP


1
The trained model
2
WebServer
3
Web App
4
5
Transfer trained model
6
Listen to events from the model
ARDUINO

CONNECT BUTTON



CONNECT BUTTON.JS

```
import connectButton from "tf4micro-motion-kit/web/button";
```

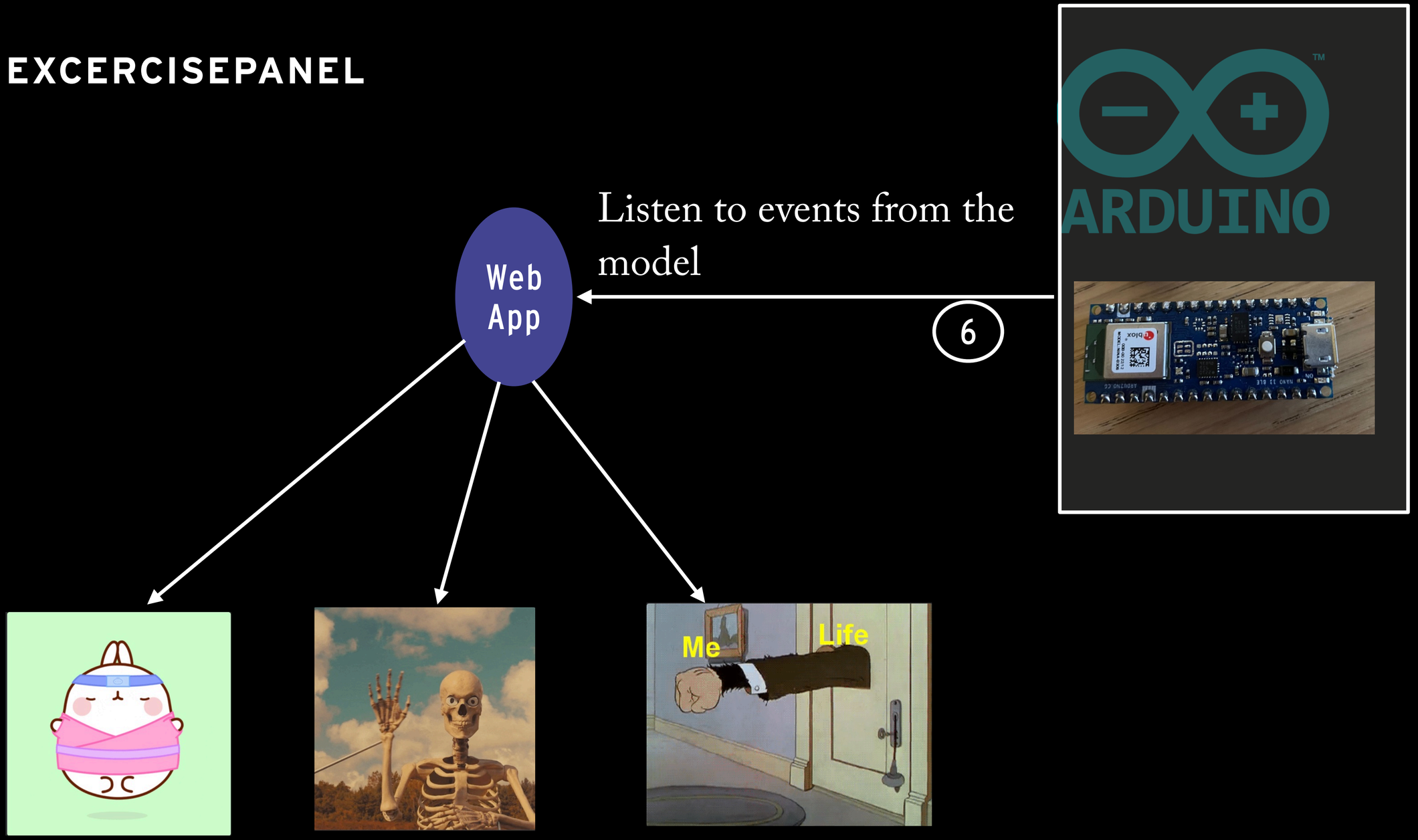
```
onMount(() => {

  button = connectButton(containerEl, {
    model: `./model.tflite?cache-bust=${Math.floor(
              Math.random() * 1000000000
            )}`,
    onInference(data){
      $lastInference = data;
    },




  });
```

EXCERCISEPANEL



EXCERCISE PANEL.JS

```
$: if ($lastInference) {
  handleInference($lastInference);
}
```

```
function handleInference(data) {
  noUpdate = false;
  const index = data.index;
  if (data.index === 0) {
    // Wave to the crowd
    numberOfWaves++;
    exerciseImageSource = "./images/wavegiphy.gif";
    activeExercise = "wave";
  }
  if (data.index === 1) {
    //This is the punch
    numberOfPunches++;
    exerciseImageSource = "./images/punchgiphy.gif";
    activeExercise = "punch";
  }
  lastExcerciseScore = data.score;
  excerciseScore = excerciseScore + (data.score ? data.score : 0);
}
```

Let's do it

Experiment done

In our experiment one step missing, and that is that still all the training is done on a more powerful device

- TensorFlow Lite states (https://www.tensorflow.org/lite/guide)

  - Unsupported on-device training, however it is on our Roadmap

REFERENCES

- https://www.tensorflow.org/lite/guide
- https://experiments.withgoogle.com/tiny-motion-trainer
- https://store.arduino.cc/products/arduino-nano-33-ble-sense
- https://www.techtarget.com/searchenterpriseai/feature/Why-TinyML-use-cases-are-taking-off
- https://www.irnas.eu/energy-efficient-system-for-detection-of-elephants-with-machine-learning/
- All gifs shown was from https://giphy.com/

Bonus Demo