

Seamless TinyML lifecycle management

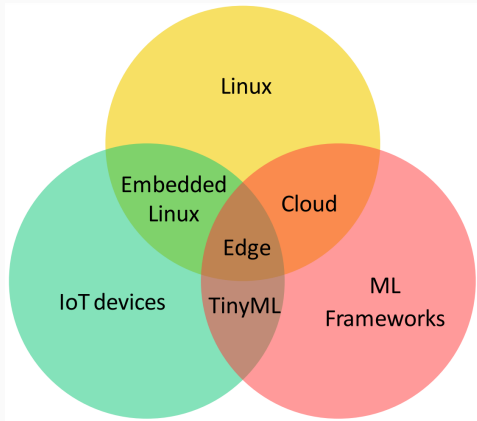
In Software Engineering Project with University of Helsinki CS

16/1/2023

Origami@NEXUS

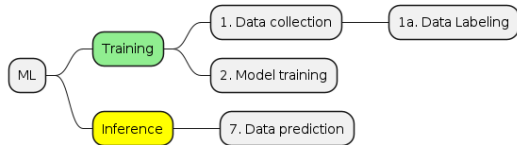
Project goal

*“The main goal of this software engineering project is to develop a solution that enables a seamless **TinyML lifecycle management**. In particular, the idea is to build a framework that **in an automated fashion** performs the different steps of the TinyML lifecycle management.”*

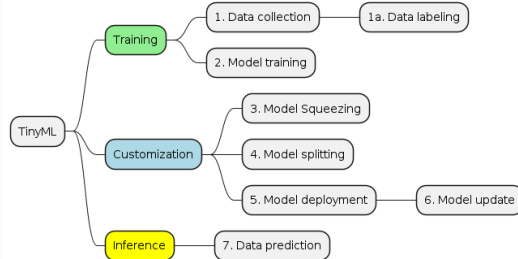


Lifecycle of: ML vs TinyML

(Cloud) ML

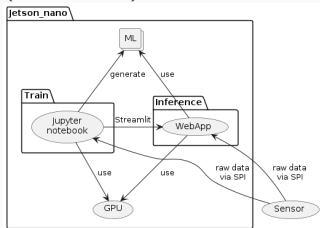


TinyML

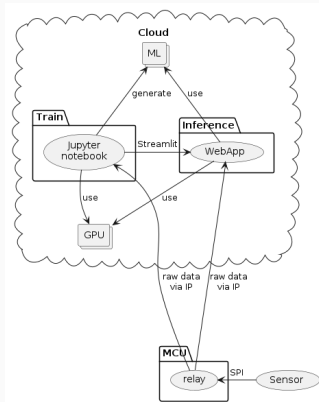


Arch: Edge ML vs Cloud ML vs TinyML

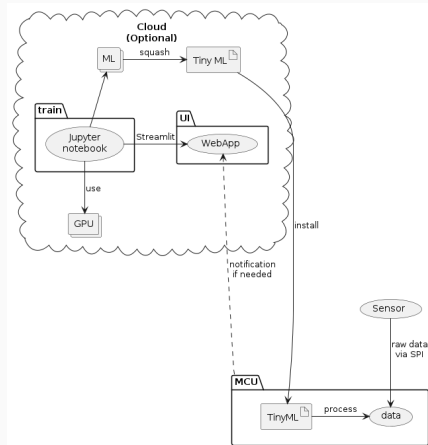
Edge ML (Local ML)



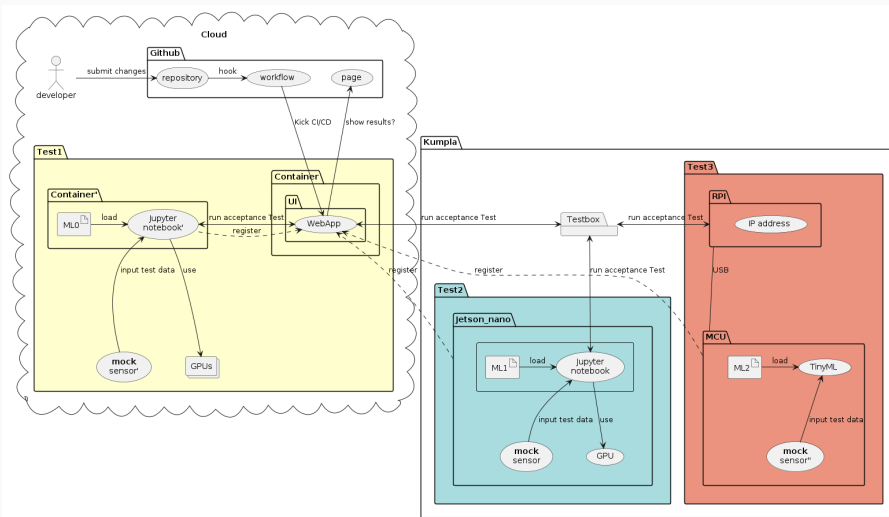
Cloud ML



TinyML



CI / CD / ATDD

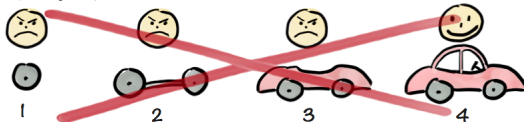


The simplest **Test1** can run the **TFLite micro Hello World** in a container w/o HW.

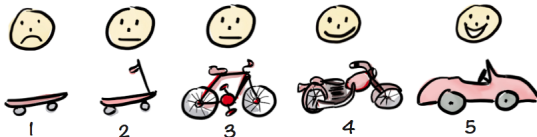
MVP iteration

Always runnable MVP at Day 1

Not like this....

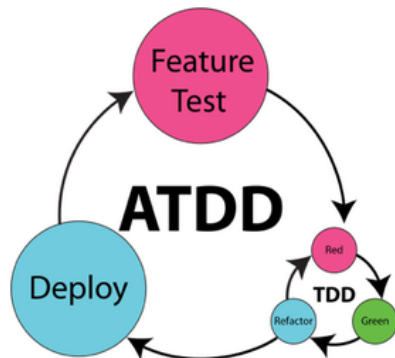


Like this!










Henrik Kniberg

Acceptance Test Driven Development



TensorFlow Lite for Microcontrollers*

ML model Examples

-  [hello_world](#)
-  [magic_wand](#)
-  [memory_footprint](#)
-  [micro_speech](#)
-  [mnist_lstm](#)
-  [network_tester](#)
-  [person_detection](#)

Supported platforms

TensorFlow Lite for Microcontrollers is written in C++ 11 and requires a 32-bit platform. It works with many processors based on the [Arm Cortex-M Series](#) architecture, and has been ported including [ESP32](#). The framework is available as an [Arduino library](#). It can also generate pre-compiled environments such as Mbed. It is open source and [can be included in any C++ 11 project](#).

The following development boards are supported:

- [Arduino Nano 33 BLE Sense](#)
- [SparkFun Edge](#)
- [STM32F746 Discovery kit](#)
- [Adafruit EdgeBadge](#)
- [Adafruit TensorFlow Lite for Microcontrollers Kit](#)
- [Adafruit Circuit Playground Bluefruit](#)
- [Espressif ESP32-DevKitC](#)
- [Espressif ESP-EYE](#)
- [Wio Terminal: ATSAM51](#)
- [Himax WE-I Plus EVB Endpoint AI Development Board](#)
- [Synopsys DesignWare ARC EM Software Development Platform](#)
- [Sony Spresense](#)

Automate with TinyMLaaS API

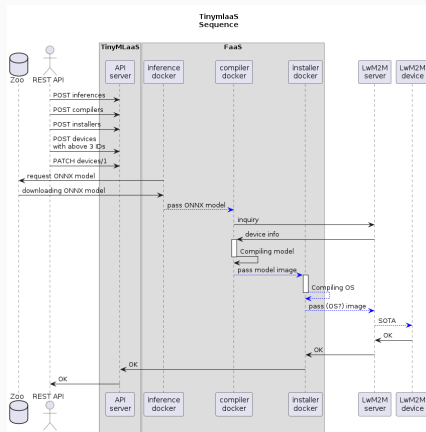
Open API spec over simple IoT system

The screenshot displays the SwaggerHub interface for the 'loop-back_appl...' API version 1.0.0. The left sidebar shows the API structure with sections for Info, Tags, Servers, and a search bar. The main area shows the OpenAPI specification for the 'DeviceController' resource. The specification includes endpoints for GET, PUT, PATCH, and DELETE for '/devices/{id}', and a POST endpoint for '/devices'. The 'POST /devices' endpoint is highlighted, showing its parameters (application/json) and an example value. The example value is a JSON object representing a device:

```
{  "name": "string",  "url": "string",  "compiled": 0,  "inferenceId": 0,  "installId": 0}
```

. The interface also shows a 'Try it out' button and a 'VALID' status at the bottom.

Function as-a-Service (FaaS)



Streamlit vs Pyscript+API server depends on how to demonstrate user story?

Origami

<https://Origami-TinyML.github.io/blog/about.html>