

Seamless TinyML lifecycle management

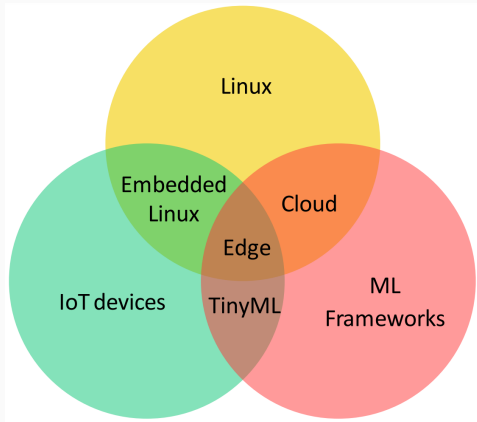
In Software Engineering Project with University of Helsinki CS

16/1/2023

Origami@NEXUS: Hiroshi Doyu, Roberto Morabito, Michihito Mizutani

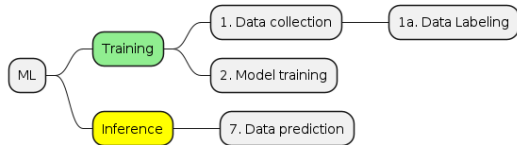
Project goal

*“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.”*

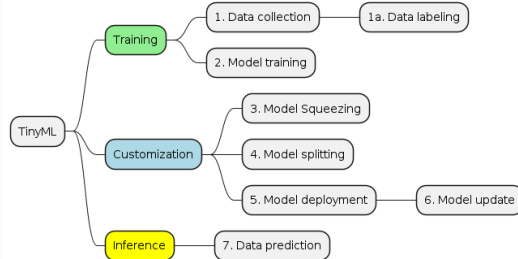


Lifecycle of: ML vs TinyML

(Cloud) ML

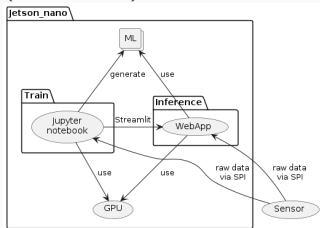


TinyML

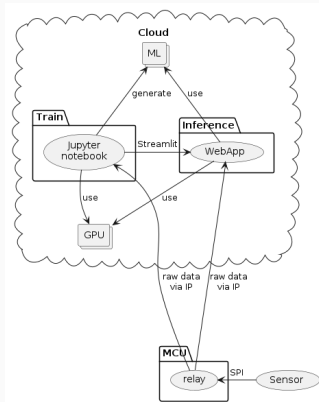


Arch: Edge ML vs Cloud ML vs TinyML

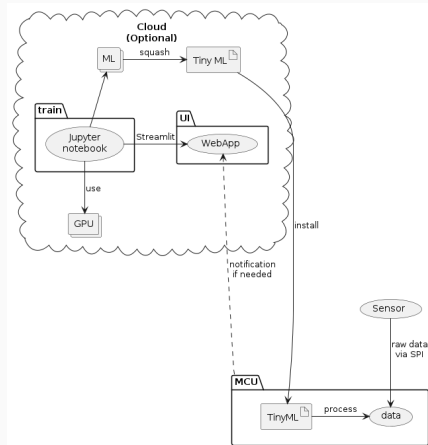
Edge ML (Local ML)



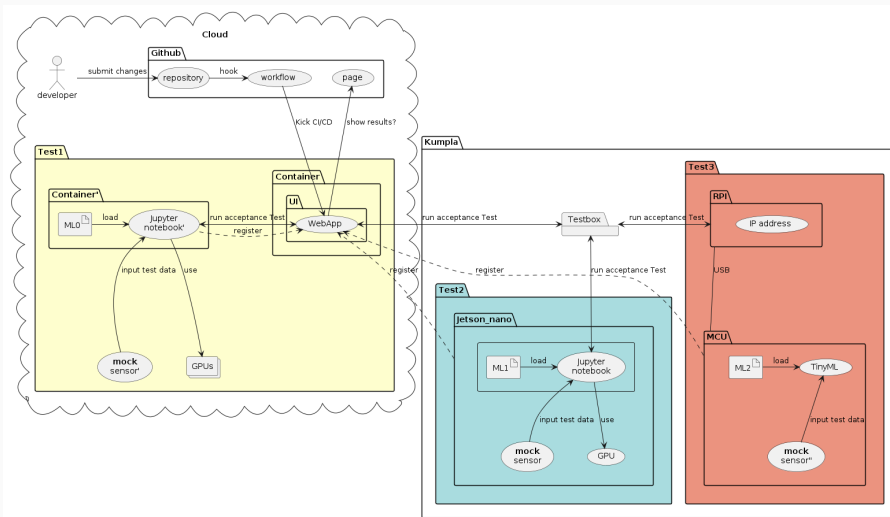
Cloud ML



TinyML



CI / CD / ATDD

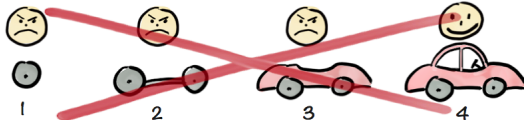


The simplest **Test1** can run the *TFLite micro Hello World* in a container w/o HW.

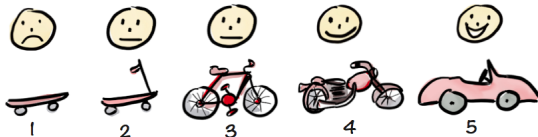
MVP iteration

Always runnable MVP at Day 1

Not like this....

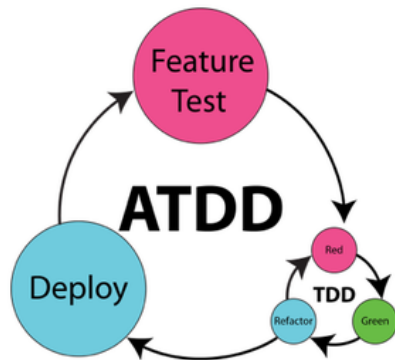


Like this!



Henrik Kniberg

Acceptance Test Driven Development

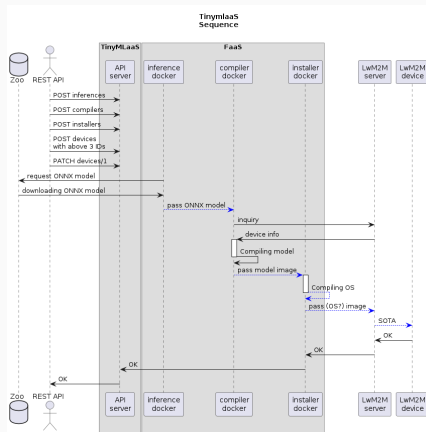


Automate with TinyMLaaS API

OpenAPI spec over simple IoT system

The screenshot displays the SwaggerHub interface for a TinyMLaaS API. The left sidebar shows the API structure with endpoints for DeviceController, DeviceCompileController, and DeviceInferenceController. The main area shows the OpenAPI specification for the DeviceController, including endpoints like GET /devices/count, PUT /devices/{id}, PATCH /devices/{id}, DELETE /devices/{id}, POST /devices, and GET /devices. The specification details parameters, request bodies, and responses for each endpoint. The right sidebar shows the PATCH /devices/{id} endpoint selected, with a 'Try it out' button and a request body field set to 'application/json'.

Function as-a-Service (FaaS)



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

TensorFlow Lite for Microcontrollers*

ML model Examples

- hello_world
- magic_wand
- memory_footprint
- micro_speech
- mnist_lstm
- network_tester
- person_detection

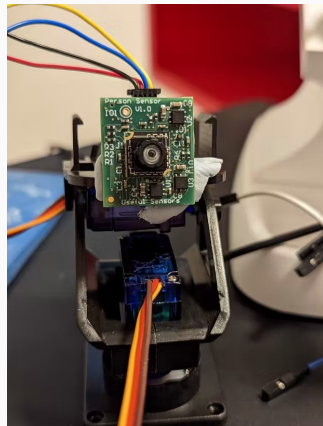
Supported platforms

TensorFlow Lite for Microcontrollers is written in C++ 11 and requires a 32-bit platform. It 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 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](#)

Face-Following Pan/Tilt Stand



Origami

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