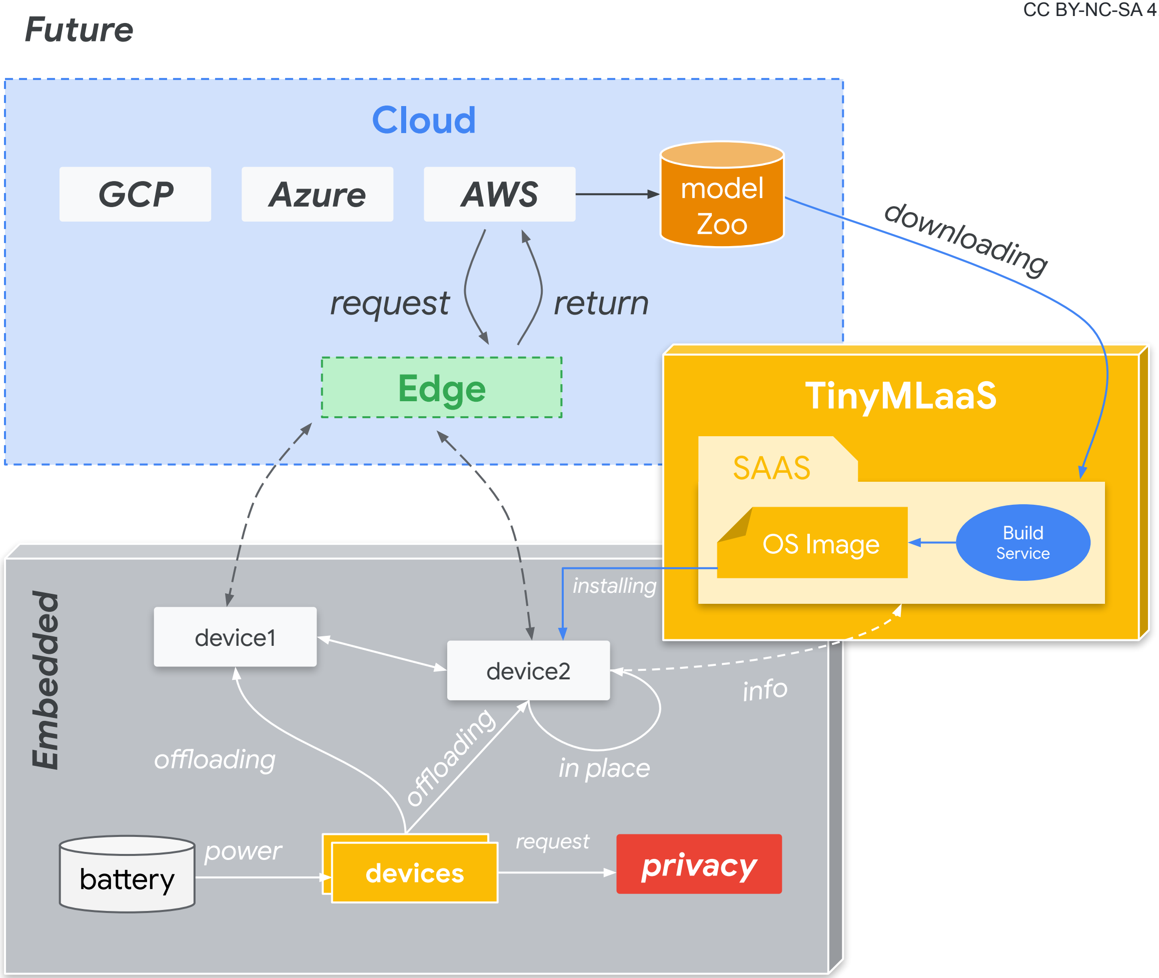
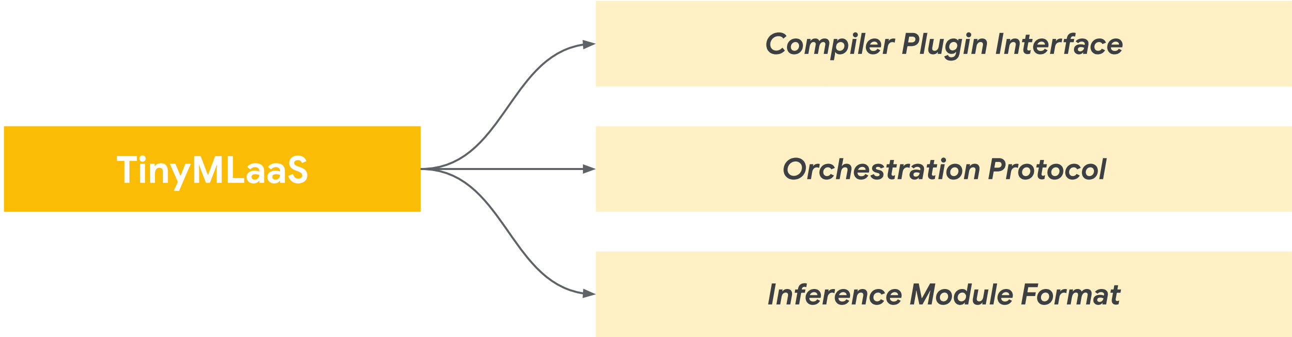
TinyMLaaS (Part 2): Design Overview

Recap: TinyMLaaS

- TinyML as a Service is a cloud or edge-based machine learning as a service

- Simplifies the deployment of ML models → abstraction



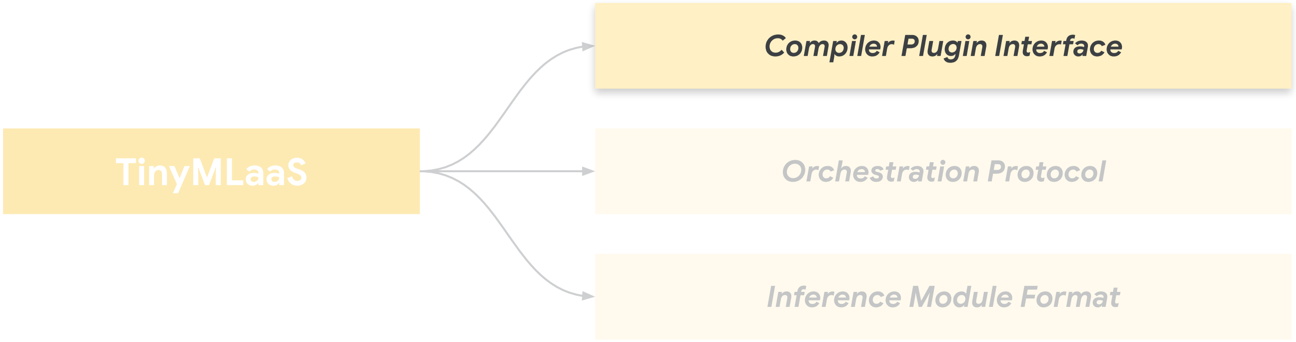


TinyMLaaS

Compiler Plugin Interface

Orchestration Protocol

Inference Module Format



TinyMLaaS

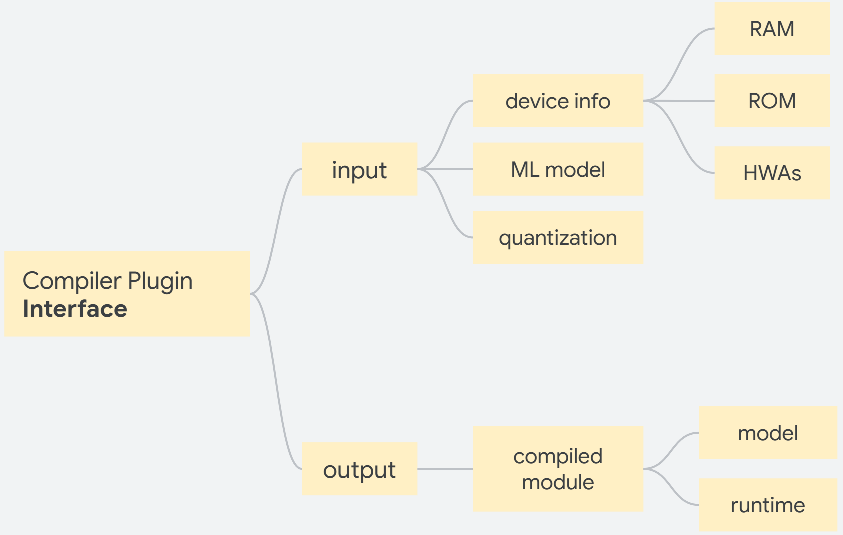
Compiler Plugin Interface

Orchestration Protocol

Inference Module Format

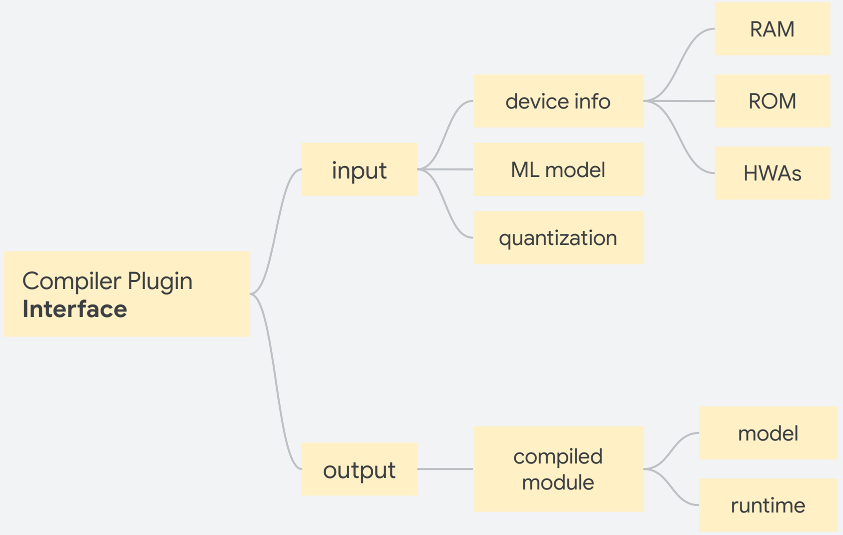
Compiler Plugin Interface

- **Decouple** the “front-end” ML model zoo from the “back-end” ML model code



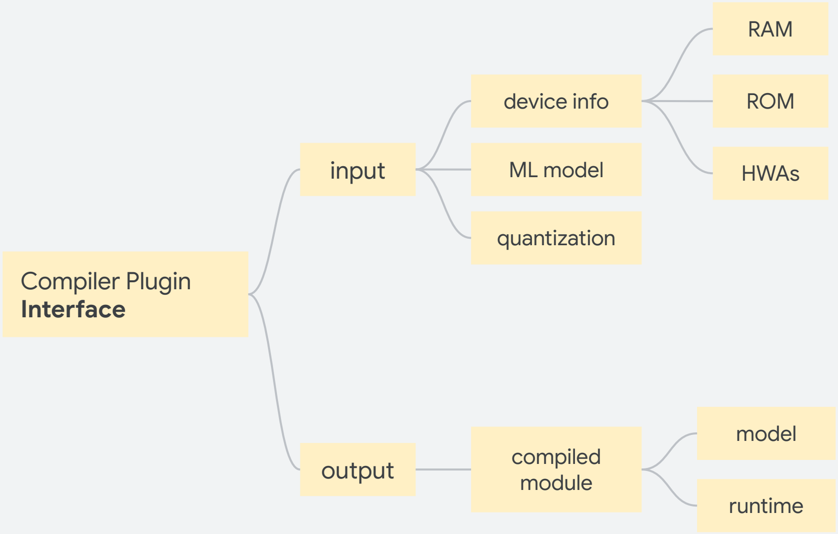
Compiler Plugin Interface

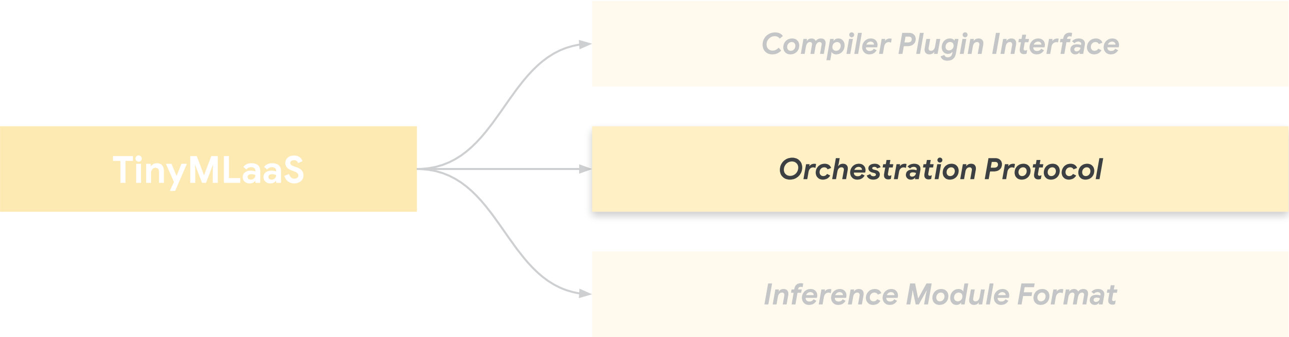
- **Decouple** the “front-end” ML model zoo from the “back-end” ML model code
- Service **pulls** in the models from a model zoo



Compiler Plugin Interface

- **Decouple** the “front-end” ML model zoo from the “back-end” ML model code
- Service **pulls** in the models from a model zoo
- Uses a **custom** compiler to generate the target code





TinyMLaaS

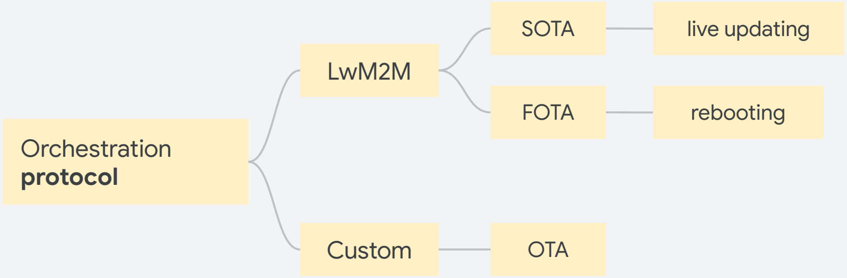
Compiler Plugin Interface

Orchestration Protocol

Inference Module Format

Orchestrator Plugin Interface

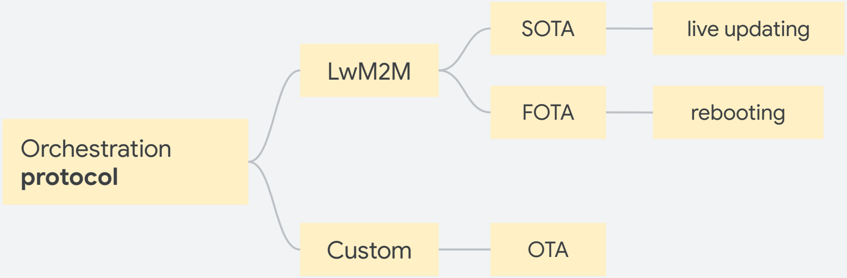
- Provide a **standard** way to interface with the device from the TinyMLaaS server



Orchestrator Plugin Interface

- Provide a **standard** way to interface with the device from the TinyMLaaS server

- Interacts with the end-devices to **gather information** about their baseline and real-time software and hardware capabilities

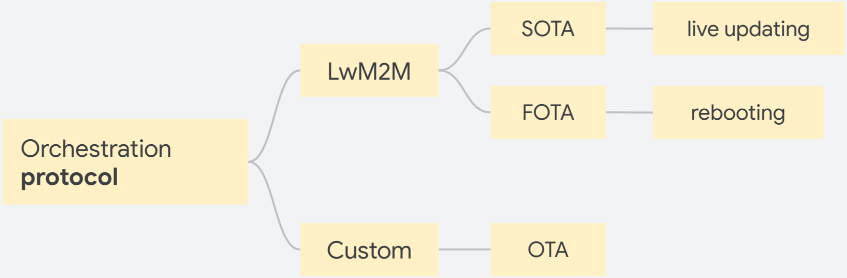


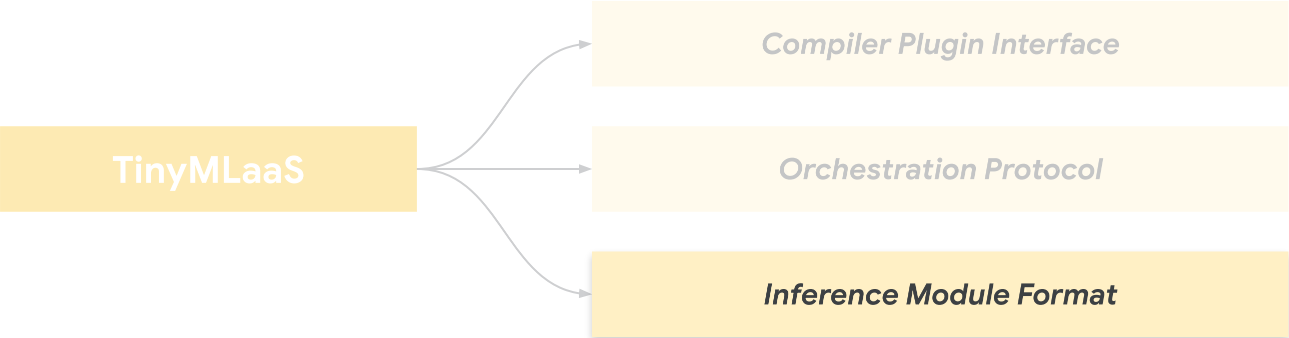
Orchestrator Plugin Interface

- Provide a **standard** way to interface with the device from the TinyMLaaS server

- Interacts with the end-devices to **gather information** about their baseline and real-time software and hardware capabilities

- Offer Firmware **Over the Air** Firmware (FOTA) and Software (SOTA) update capabilities to comply with requests





TinyMLaaS

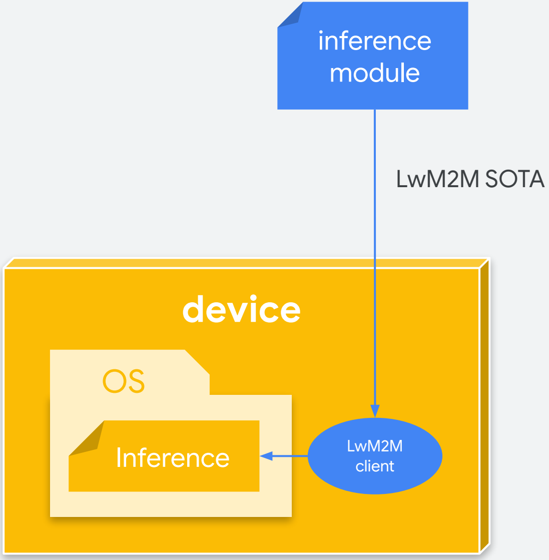
Compiler Plugin Interface

Orchestration Protocol

Inference Module Format

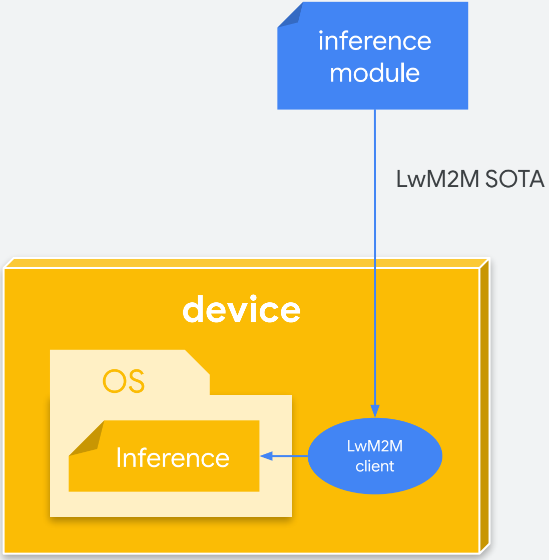
Inference Module

- **Standardization** is key in the inference module stage



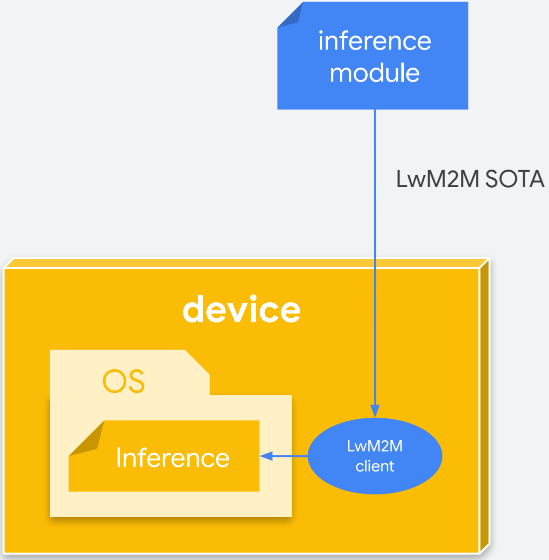
Inference Module

- **Standardization** is key in the inference module stage
- Standardization allows us to represent a wide range of compiler and inference applications, across heterogeneous features of OS and hardware chipsets



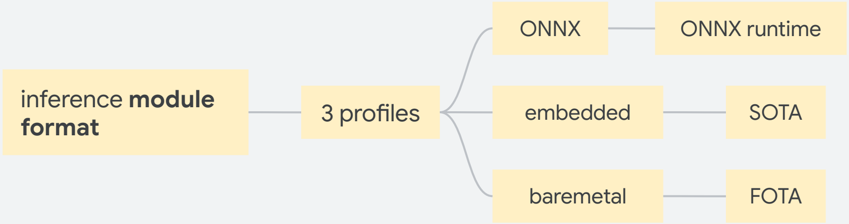
Inference Module

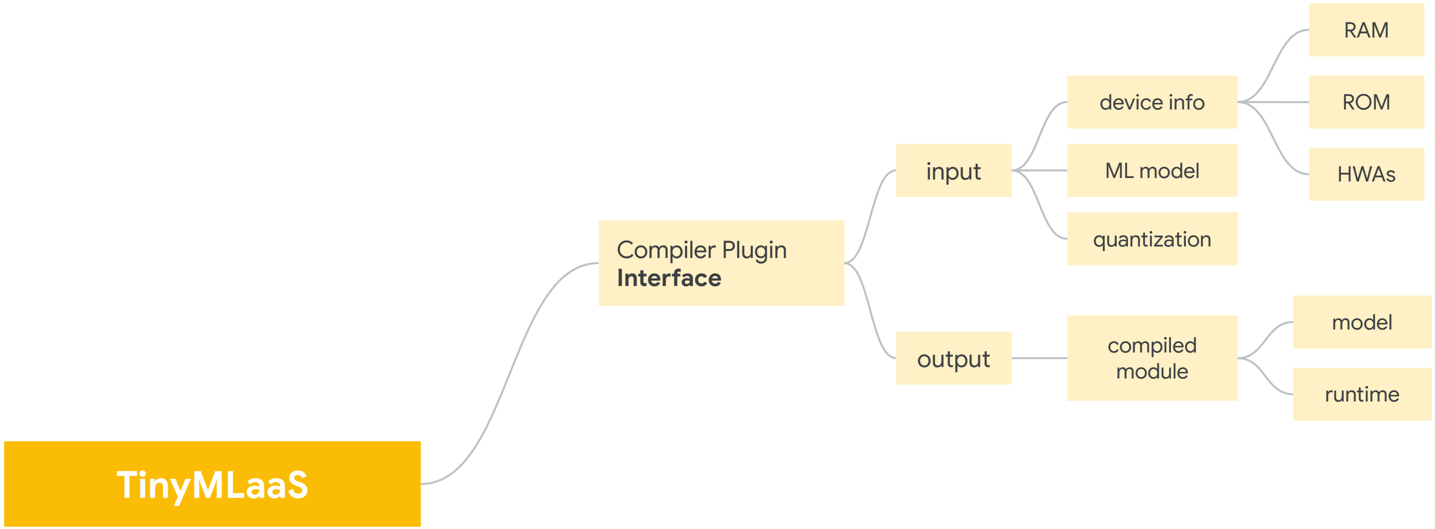
- **Standardization** is key in the inference module stage
- Standardization allows us to represent a wide range of compiler and inference applications, across heterogeneous features of OS and hardware chipsets
- Inference module format provides a predefined representation format for the output of the compiled module

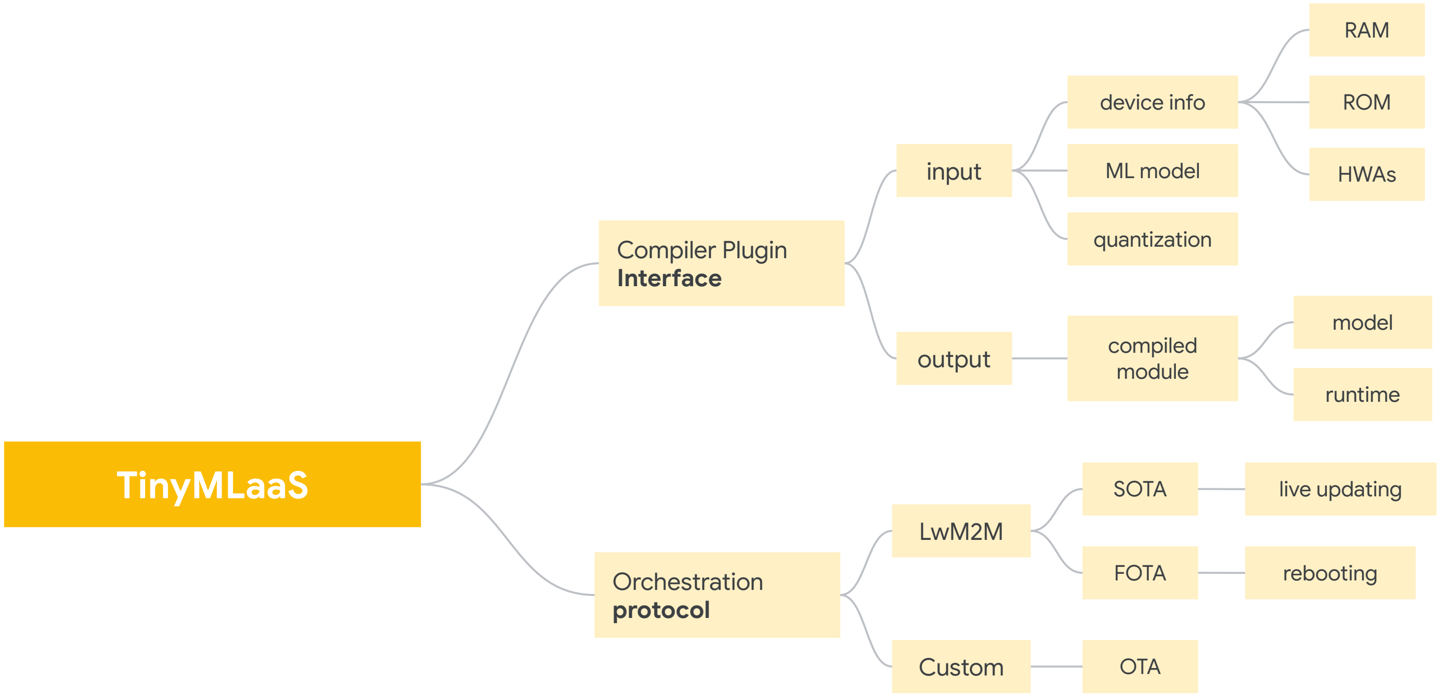


Inference Module

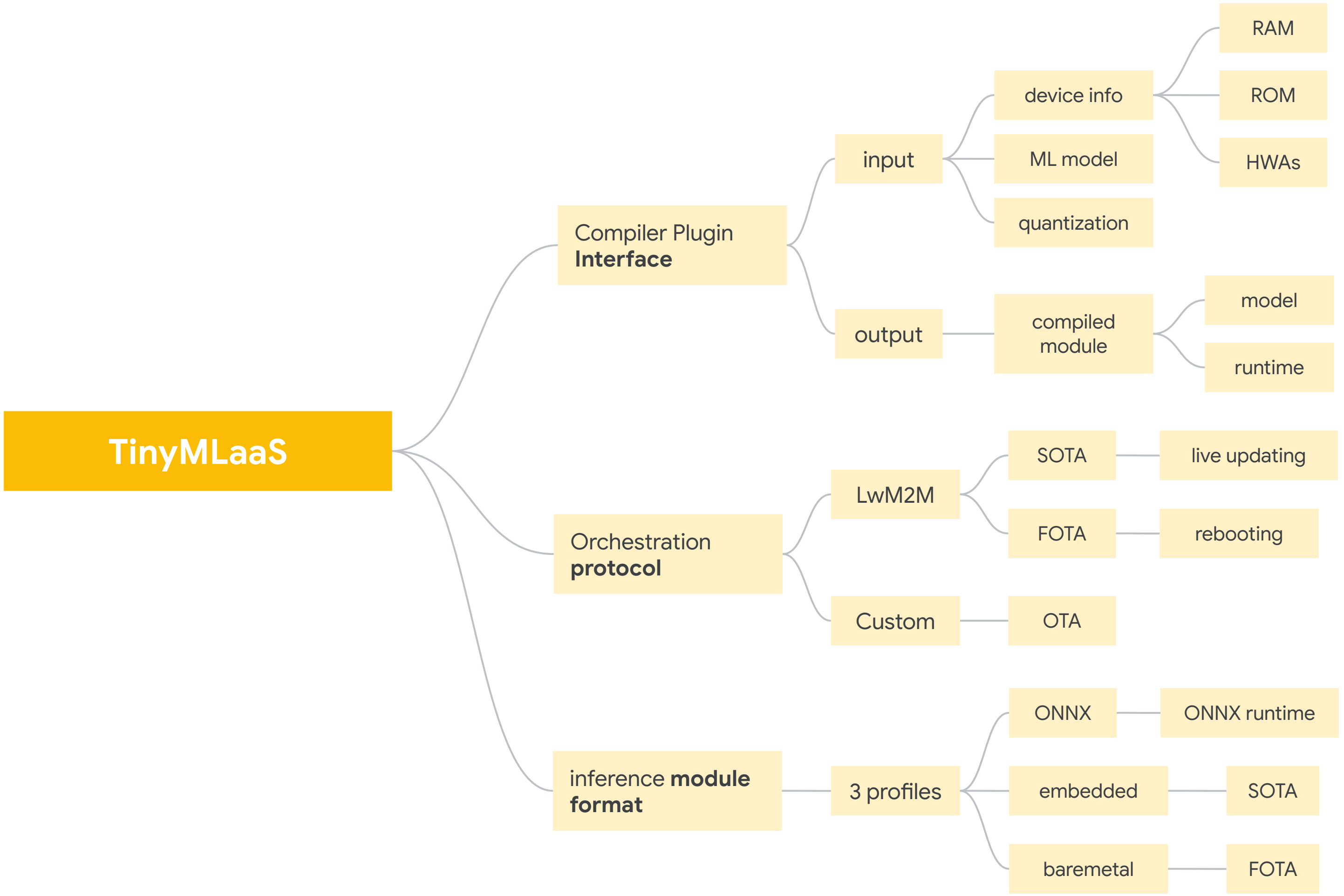
- **Standardization** is key in the inference module stage
- Standardization allows us to represent a wide range of compiler and inference applications, across heterogeneous features of OS and hardware chipsets
- Inference module format provides a predefined representation format for the output of the compiled module



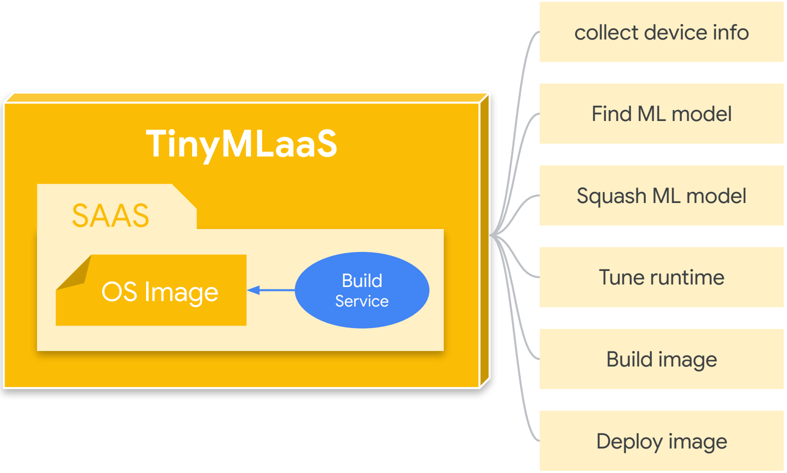

TinyMLaaS
Compiler Plugin **Interface**
input
device info
RAM
ROM
HWAs
ML model
quantization
output
compiled module
model
runtime



- TinyMLaaS
  - Compiler Plugin **Interface**
    - input
      - device info
        - RAM
        - ROM
        - HWAs
      - ML model
      - quantization
    - output
      - compiled module
        - model
        - runtime
  - Orchestration **protocol**
    - LwM2M
      - SOTA
        - live updating
      - FOTA
        - rebooting
    - Custom
      - OTA


TinyMLaaS
Compiler Plugin **Interface**
input
device info
RAM
ROM
HWAs
ML model
quantization
output
compiled module
model
runtime
Orchestration **protocol**
LwM2M
SOTA
live updating
FOTA
rebooting
Custom
OTA
inference **module format**
3 profiles
ONNX
ONNX runtime
embedded
SOTA
baremetal
FOTA

Execution flow


TinyMLaaS
SAAS
OS Image
Build
Service
collect device info
Find ML model
Squash ML model
Tune runtime
Build image
Deploy image