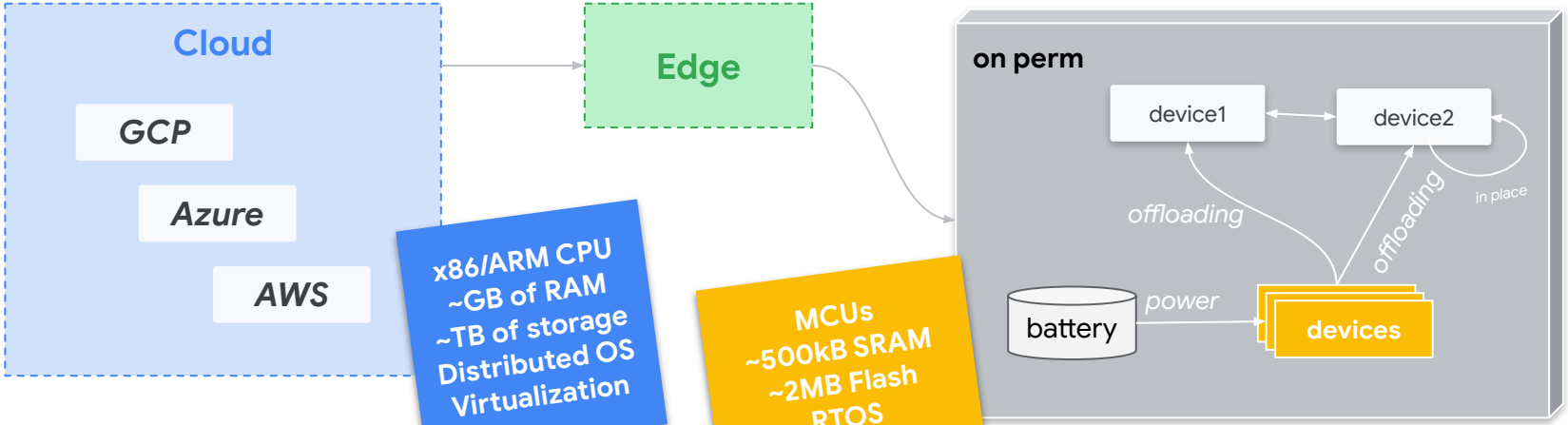
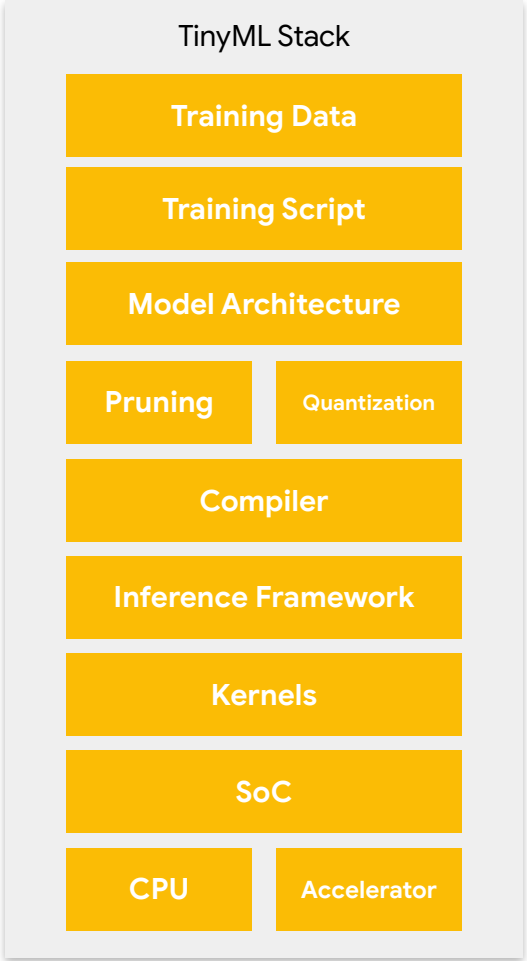
# Challenges for Scaling TinyML Deployment (Part 2)





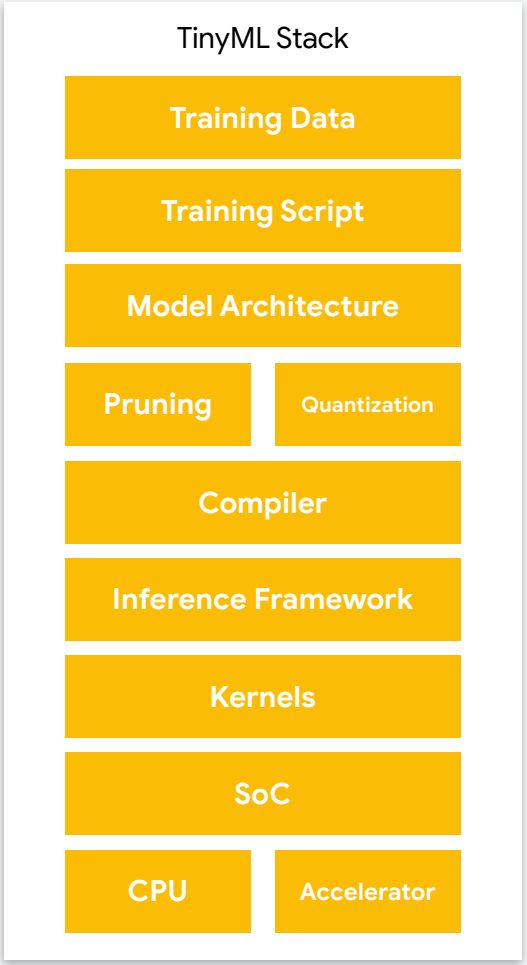
**ML-dedicated hardware:** CPU, GPU, TPU
**ML-dedicated software:** many tools
**ML Tasks** ⟶ Data collection and preprocessing, data transformation, model training, model deployment, inference

**No ML-dedicated Hardware**
**No ML-dedicated software**
**ML Tasks** ⟶ Inference



# TinyML Computing Stack is Complex

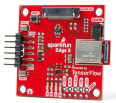
- Different models and network architectures
- Different model optimization methods
- Specialized compilers for embedded devices
- Specialized hardware and ML accelerators

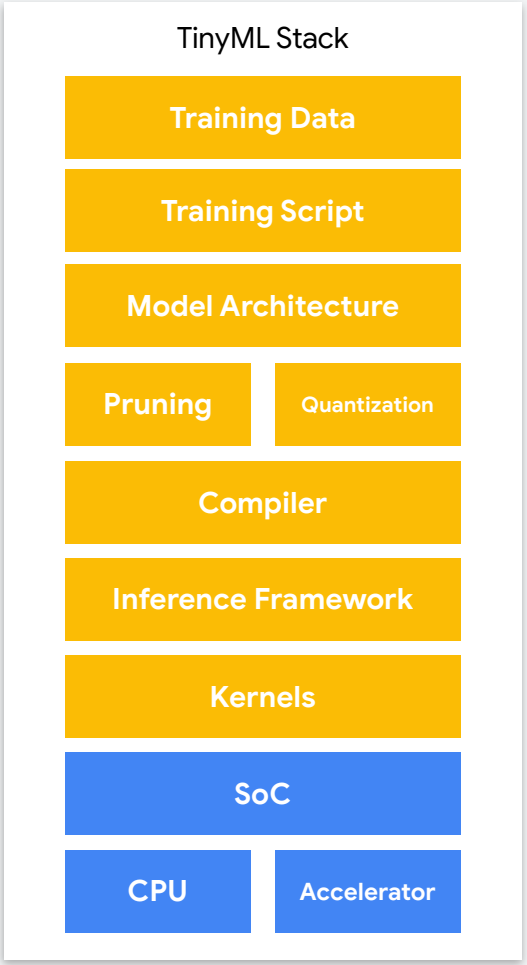


# TinyML Computing Stack is Complex

- **Specialized hardware** and ML accelerators

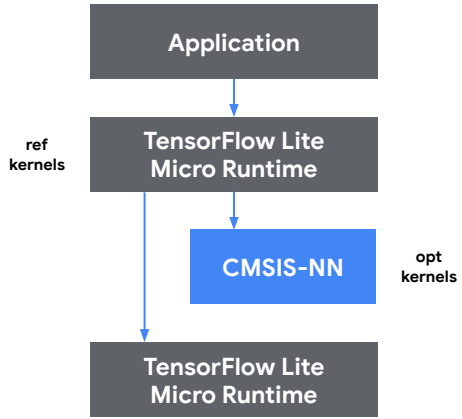


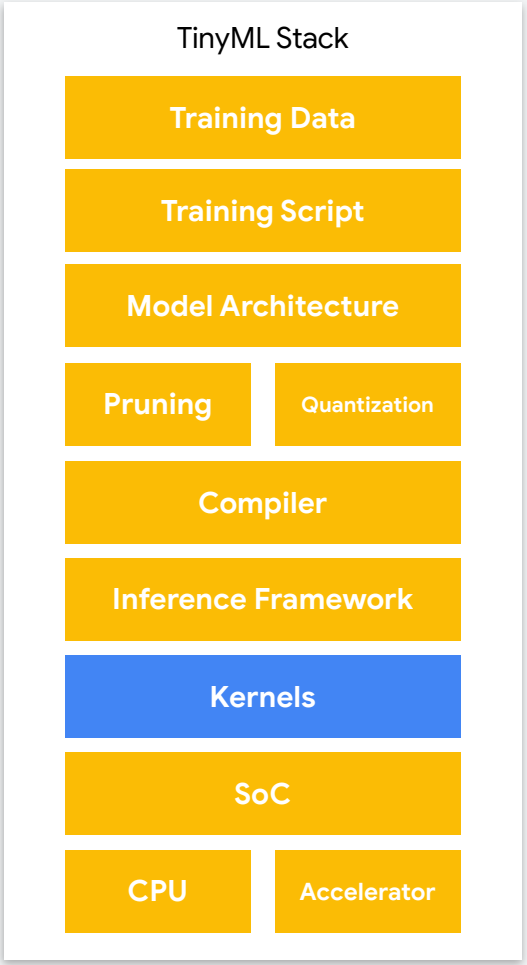


# TinyML Computing Stack is Complex

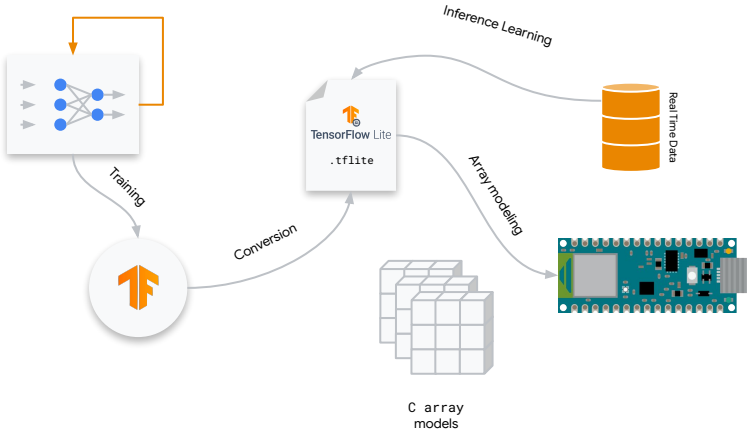
- **Software kernels** for unlocking the hardware's performance

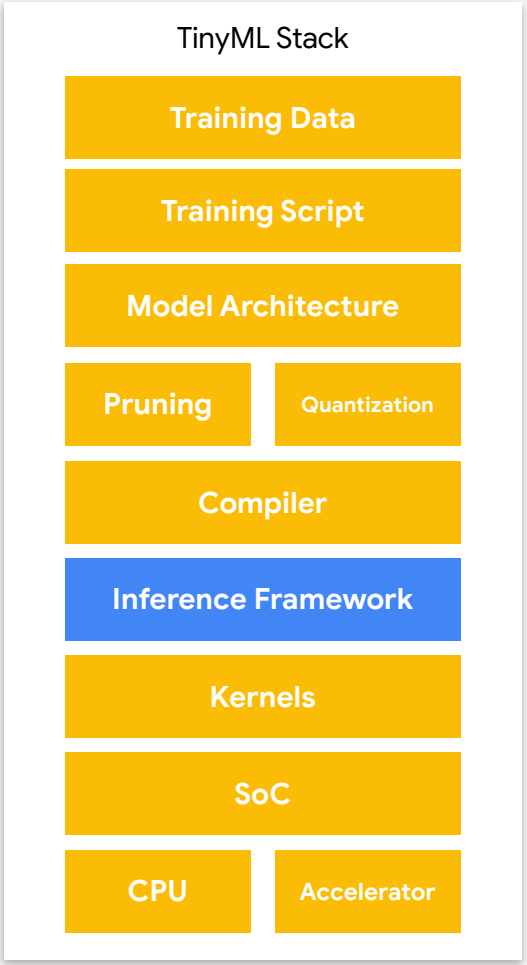


# TinyML Computing Stack is Complex

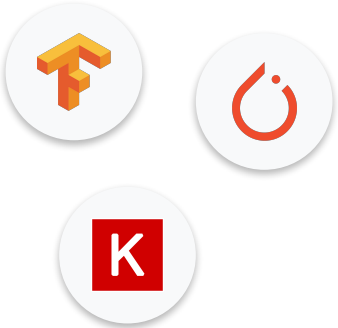
- **Inference frameworks** for embedded deployment

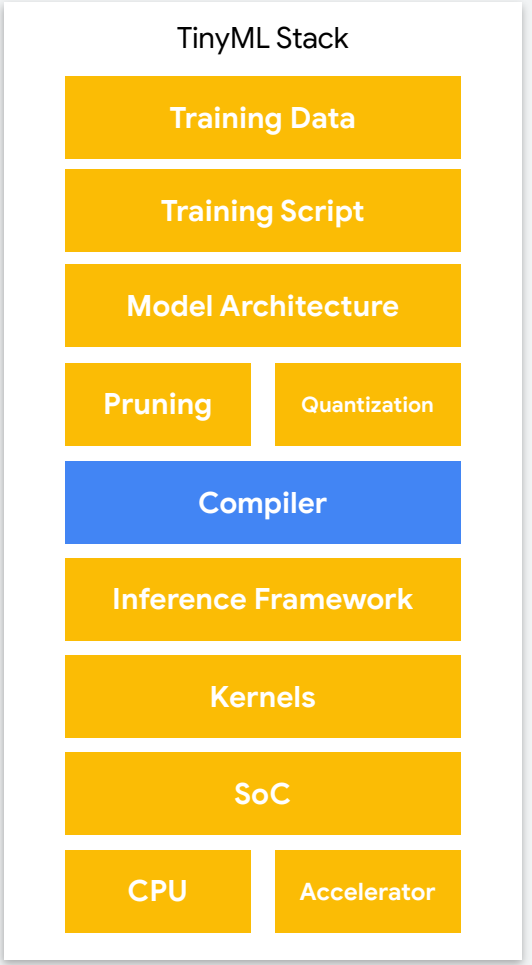


# TinyML Computing Stack is Complex

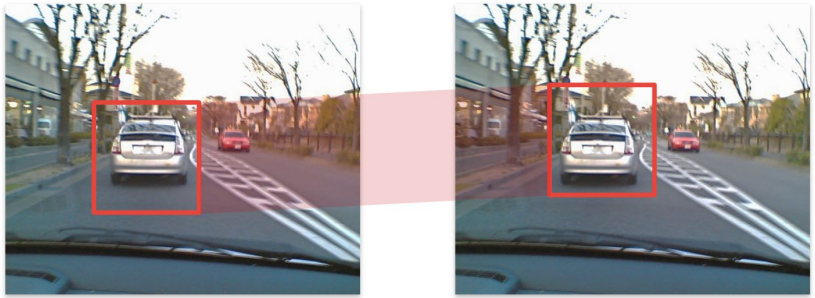
- **Specialized compilers** for embedded devices

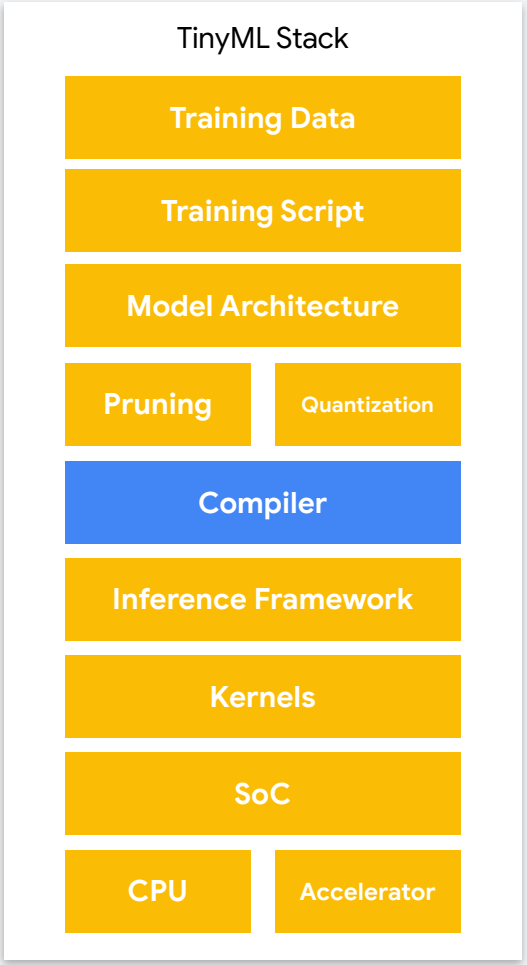




# TinyML Computing Stack is Complex

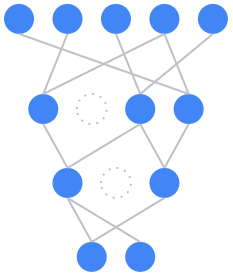
- **Specialized compilers** for embedded devices

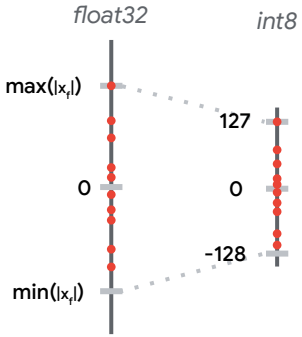


# TinyML Computing Stack is Complex

- **Different model optimization** algorithms



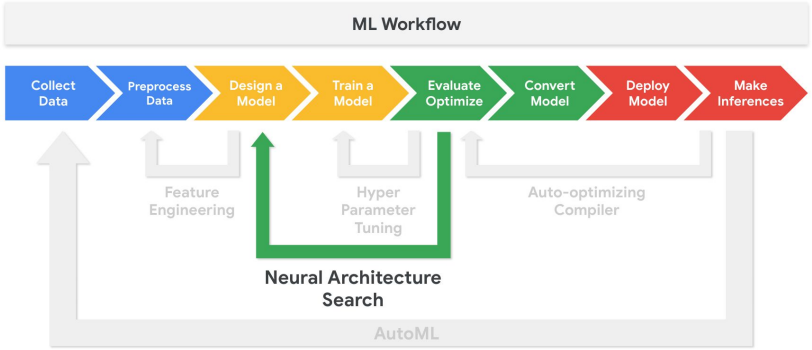


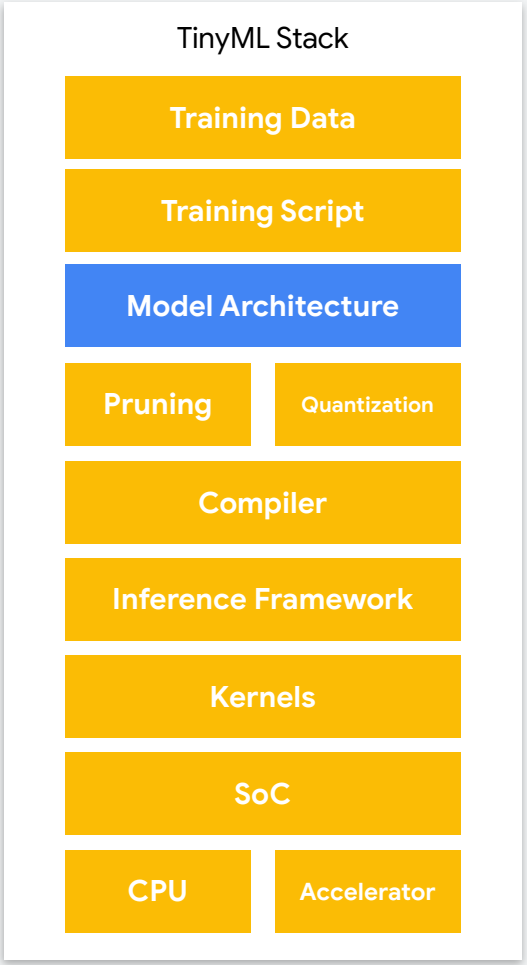

TinyML Stack

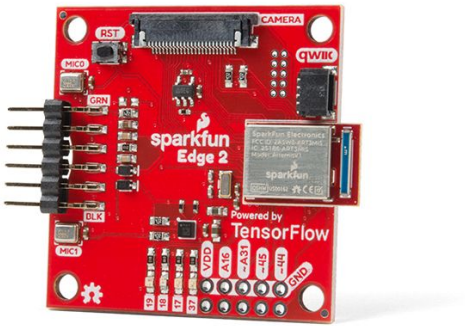
- Training Data
- Training Script
- Model Architecture
- Pruning
- Quantization
- Compiler
- Inference Framework
- Kernels
- SoC
- CPU
- Accelerator

# TinyML Computing Stack is Complex

- Lots of **different model optimization** methods



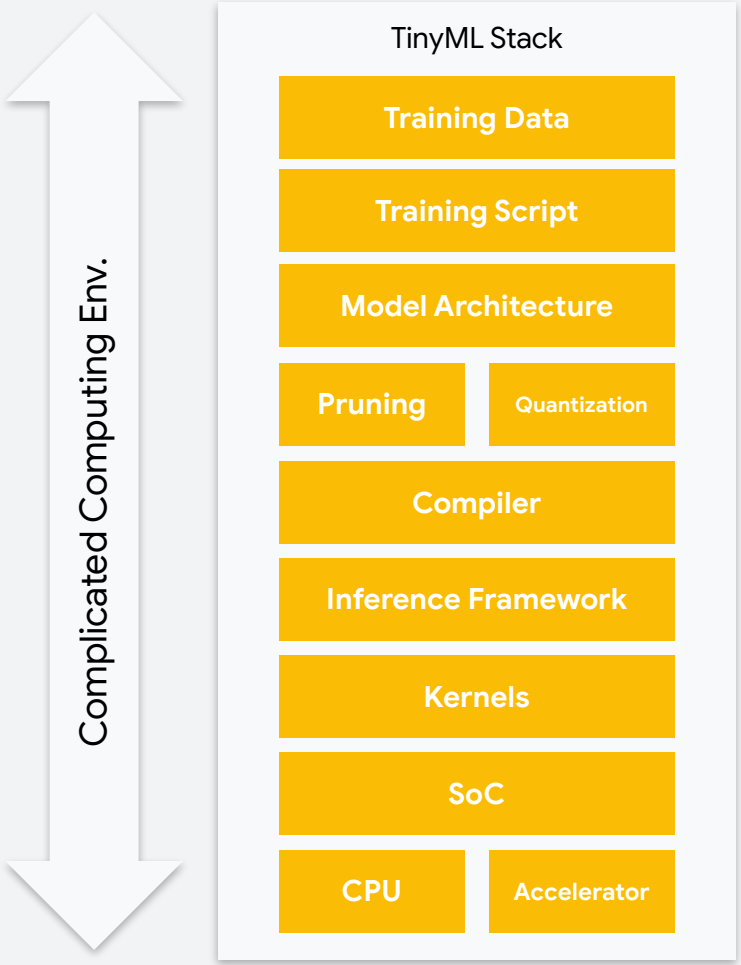



Complicated Computing Env.

TinyML Stack

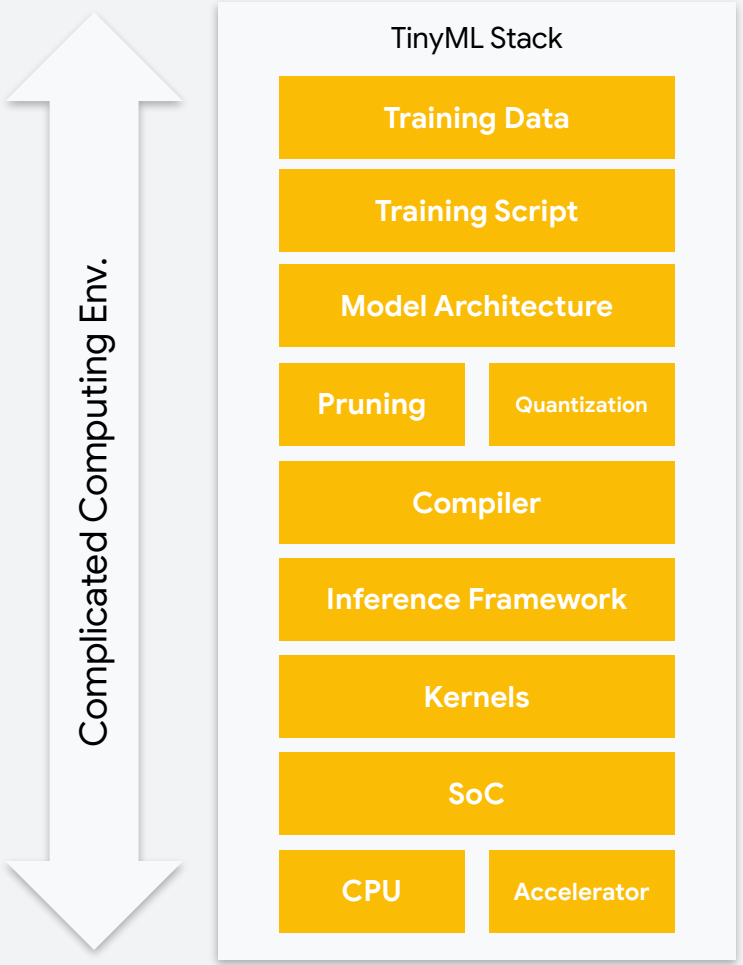
- Training Data
- Training Script
- Model Architecture
- Pruning
- Quantization
- Compiler
- Inference Framework
- Kernels
- SoC
- CPU
- Accelerator

# Cross-Product Diversity Issue



# Cross-Product Diversity Issue

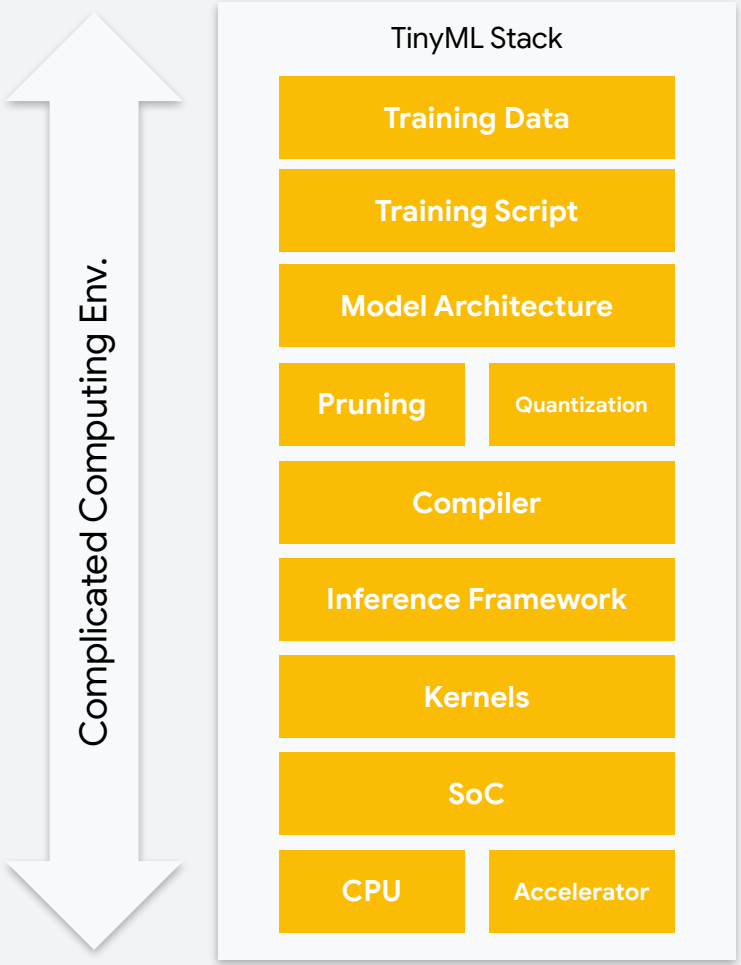
The advantages of using **specialized hardware** for ML must be balanced with the use of dedicated ML compilers that adapt a certain ML model to the targeted hardware platform.



# Cross-Product Diversity Issue

The advantages of using **specialized hardware** for ML must be balanced with the use of dedicated ML compilers that adapt a certain ML model to the targeted hardware platform.

This hardware and **associated compilers'** heterogeneity (i.e. application of various kinds of special hardware) generates additional fragmentation.



# Cross-Product Diversity Issue

The advantages of using **specialized hardware** for ML must be balanced with the use of dedicated ML compilers that adapt a certain ML model to the targeted hardware platform.

This hardware and **associated compilers'** heterogeneity (i.e. application of various kinds of special hardware) generates additional fragmentation.

It also offers **poor flexibility** against the possibility of easily switching hardware context due to the need to re-compile the ML inference model for the targeted device.