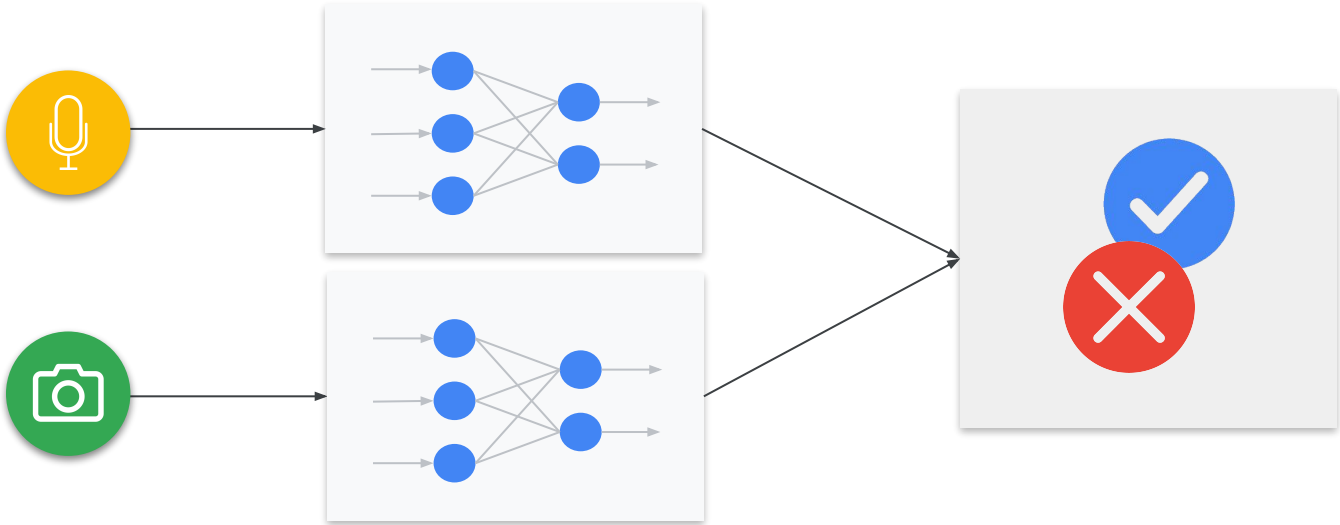
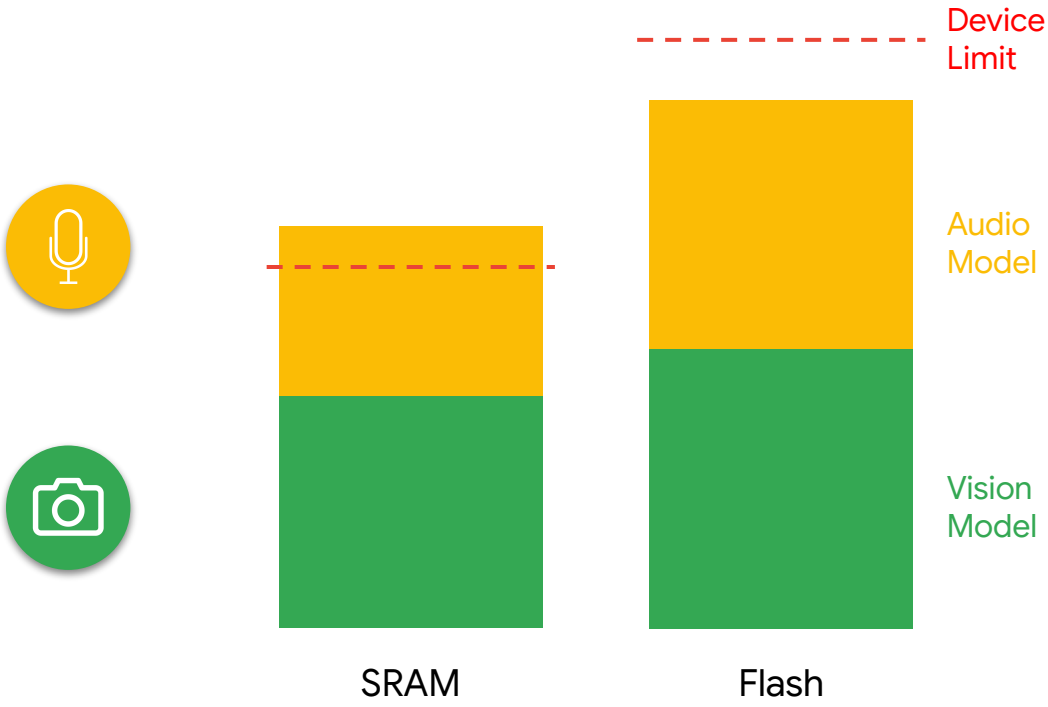
Multi Tenancy in TensorFlow Lite Micro

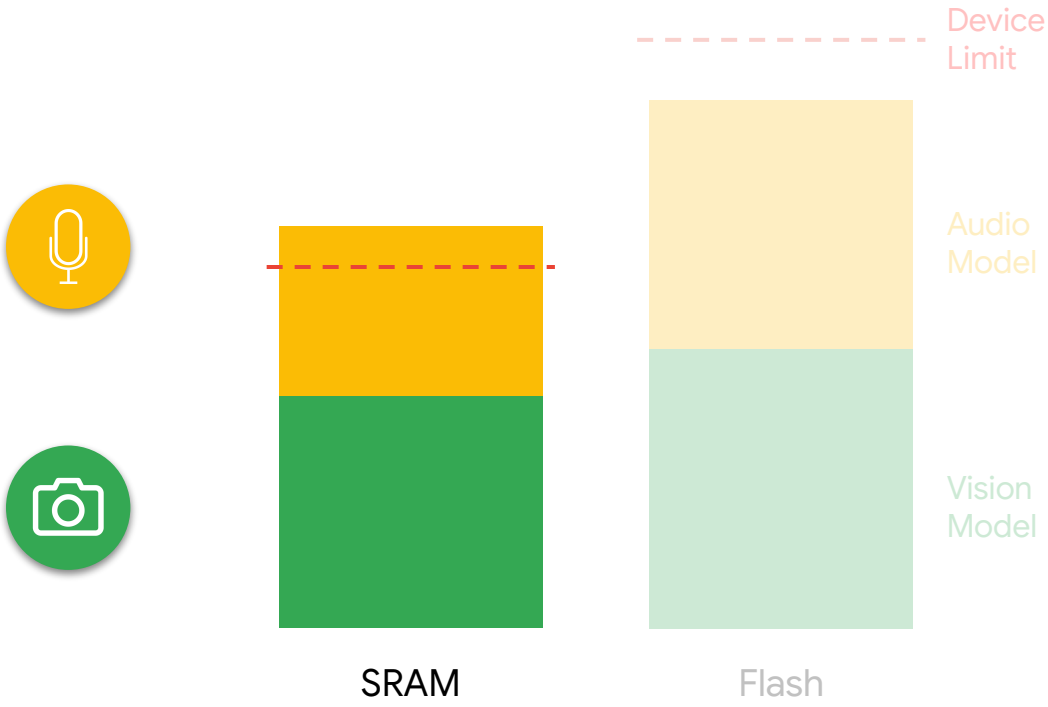
MultiTenant



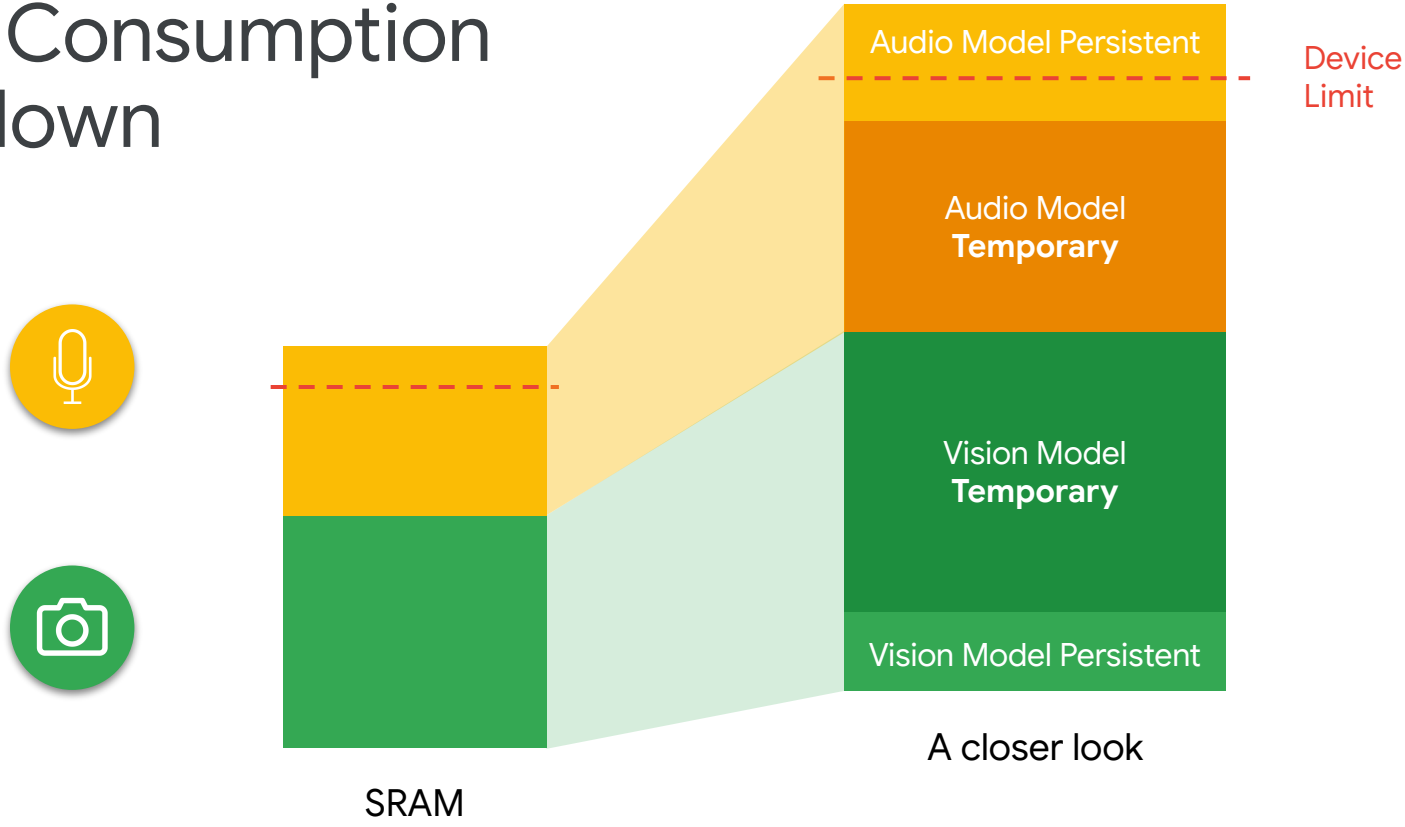
Fitting Multiple Models



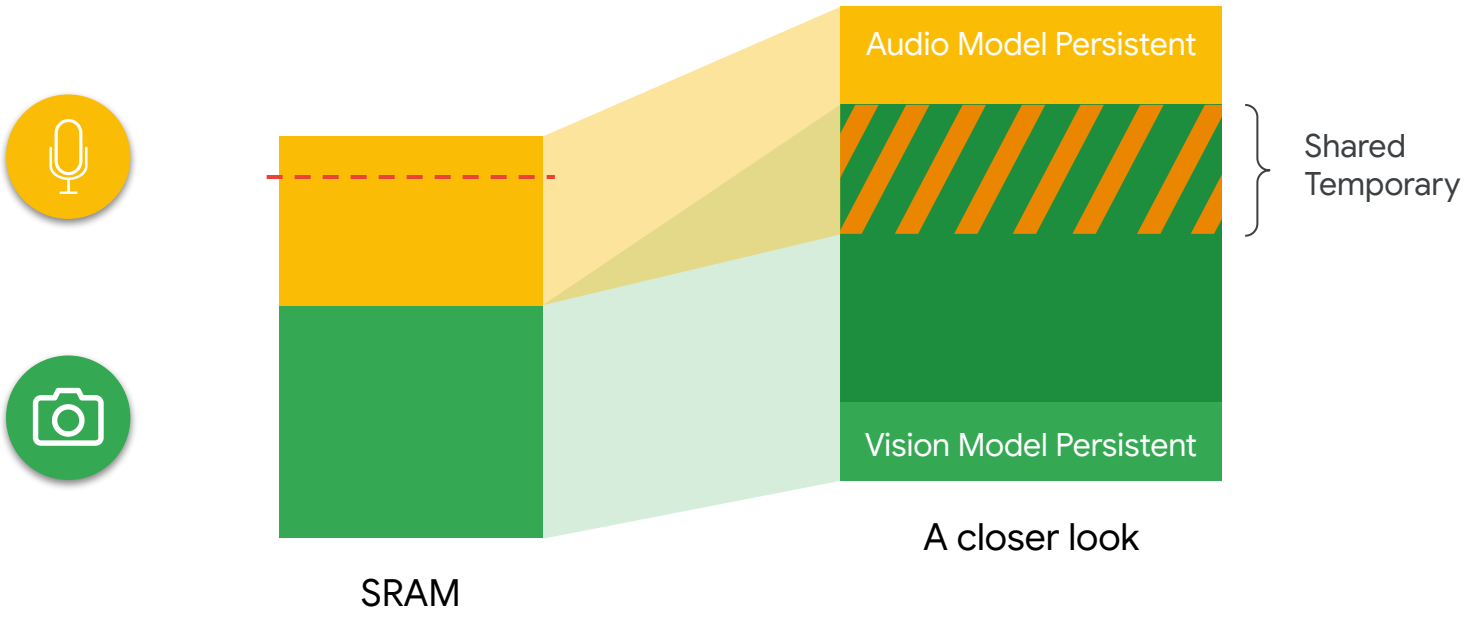
Fitting Multiple Models



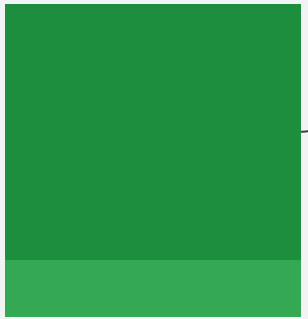
SRAM Consumption Breakdown



SRAM Temporary Reuse



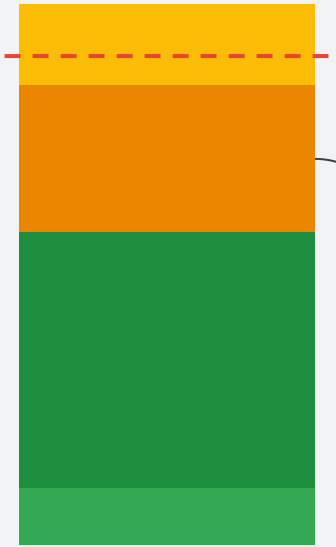
Standard Allocator



SRAM

```
static uint8_t vww_tensor_arena[VWWTensorArenaSize];

...

static tflite::MicroInterpreter vww_static_interpreter(
    vww_model, micro_op_resolver, vww_tensor_arena,
    VWWTensorArenaSize, error_reporter);
vww_interpreter = &vww_static_interpreter;

// allocate the VWW model from tensor_arena
TfLiteStatus allocate_status =
vww_interpreter->AllocateTensors();

static uint8_t kws_tensor_arena[KWSTensorArenaSize];

...

static tflite::MicroInterpreter kws_static_interpreter(
    kws_model, micro_op_resolver, kws_tensor_arena,
    KWSTensorArenaSize, error_reporter);
kws_interpreter = &kws_static_interpreter;

// allocate the KWS model from tensor_arena.
// Some head space is saved
allocate_status = kws_interpreter->AllocateTensors();
```

Standard Allocator



SRAM

```
static uint8_t vww_tensor_arena[VWWTensorArenaSize];

...

static tflite::MicroInterpreter vww_static_interpreter(
    vww_model, micro_op_resolver, vww_tensor_arena,
    VWWTensorArenaSize, error_reporter);
vww_interpreter = &vww_static_interpreter;

// allocate the VWW model from tensor_arena
TfLiteStatus allocate_status =
vww_interpreter->AllocateTensors();

static uint8_t kws_tensor_arena[KWSTensorArenaSize];

...

static tflite::MicroInterpreter kws_static_interpreter(
    kws_model, micro_op_resolver, kws_tensor_arena,
    KWSTensorArenaSize, error_reporter);
kws_interpreter = &kws_static_interpreter;

// allocate the KWS model from tensor_arena.
// Some head space is saved
allocate_status = kws_interpreter->AllocateTensors();
```

Shared Allocator

SRAM

```
static uint8_t combined_tensor_arena[CombinedTensorArenaSize];

tflite::MicroAllocator* allocator =
      tflite::MicroAllocator::Create(combined_tensor_arena,
                          CombinedTensorArenaSize,error_reporter);

...

static tflite::MicroInterpreter vww_static_interpreter(
    vww_model, micro_op_resolver, allocator, error_reporter);
vww_interpreter = &vww_static_interpreter;

// allocate the VWW model from tensor_arena
TfLiteStatus allocate_status = vww_interpreter->AllocateTensors();

...

static tflite::MicroInterpreter kws_static_interpreter(
    kws_model, micro_op_resolver, allocator, error_reporter);
kws_interpreter = &kws_static_interpreter;

// allocate the KWS model from tensor_arena.
// Some head space is saved
allocate_status = kws_interpreter->AllocateTensors();
```

Optimized MultiTenancy