

# Magic Wand Application Architecture

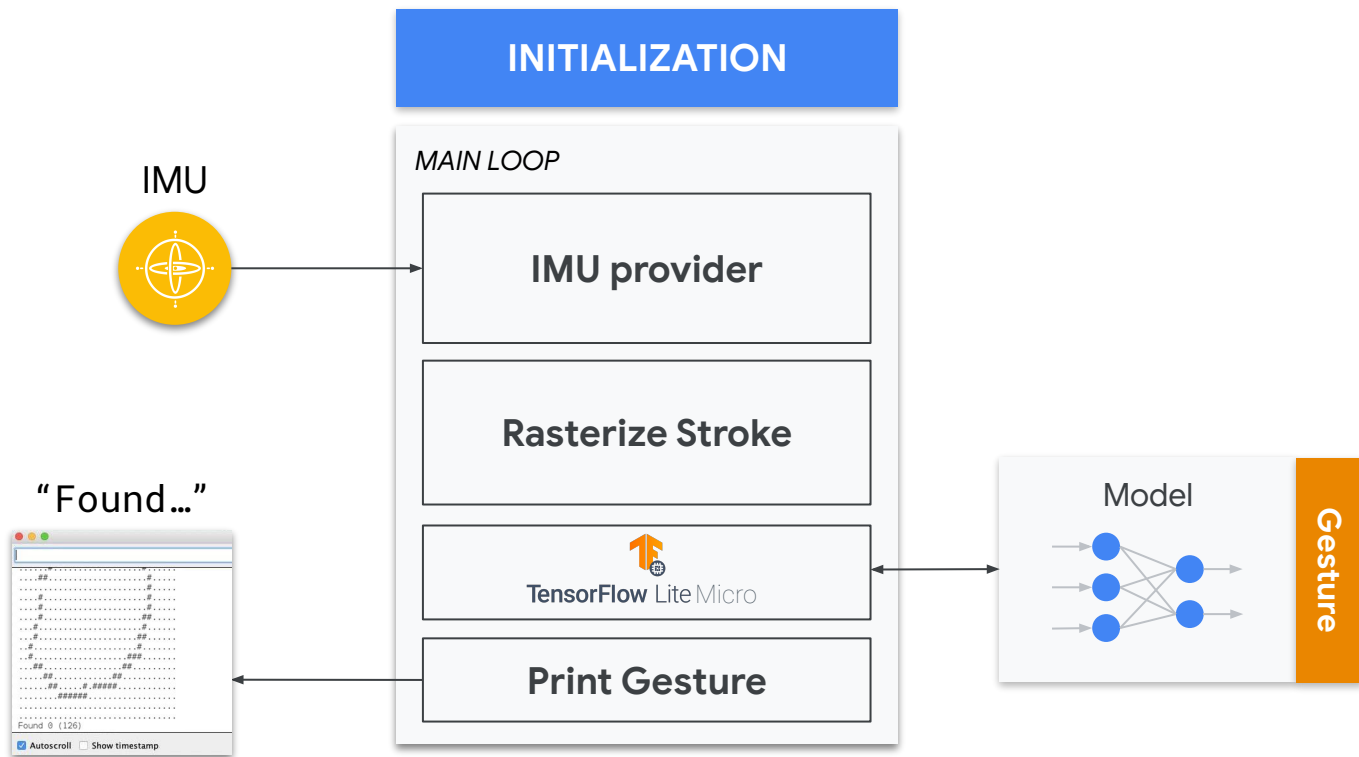


# Magic Wand Components

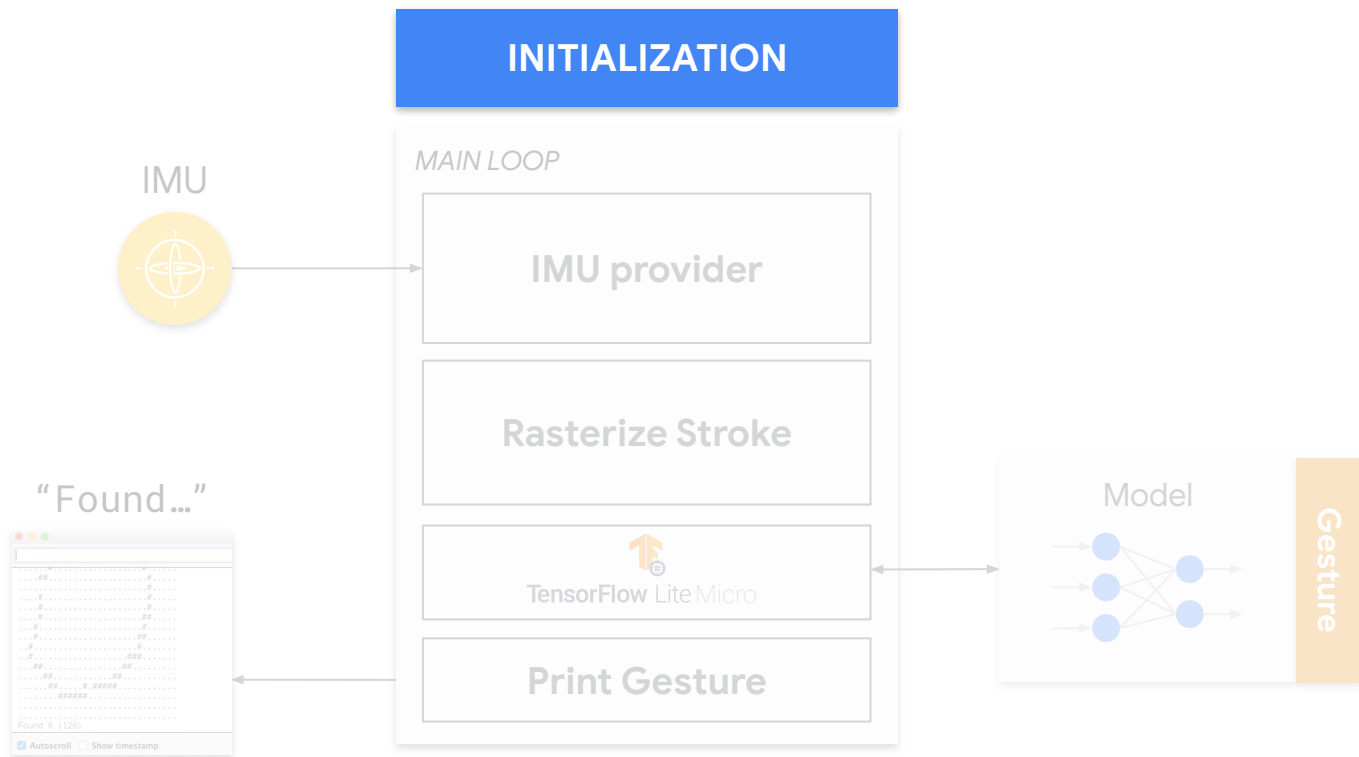
INITIALIZATION

*MAIN LOOP*

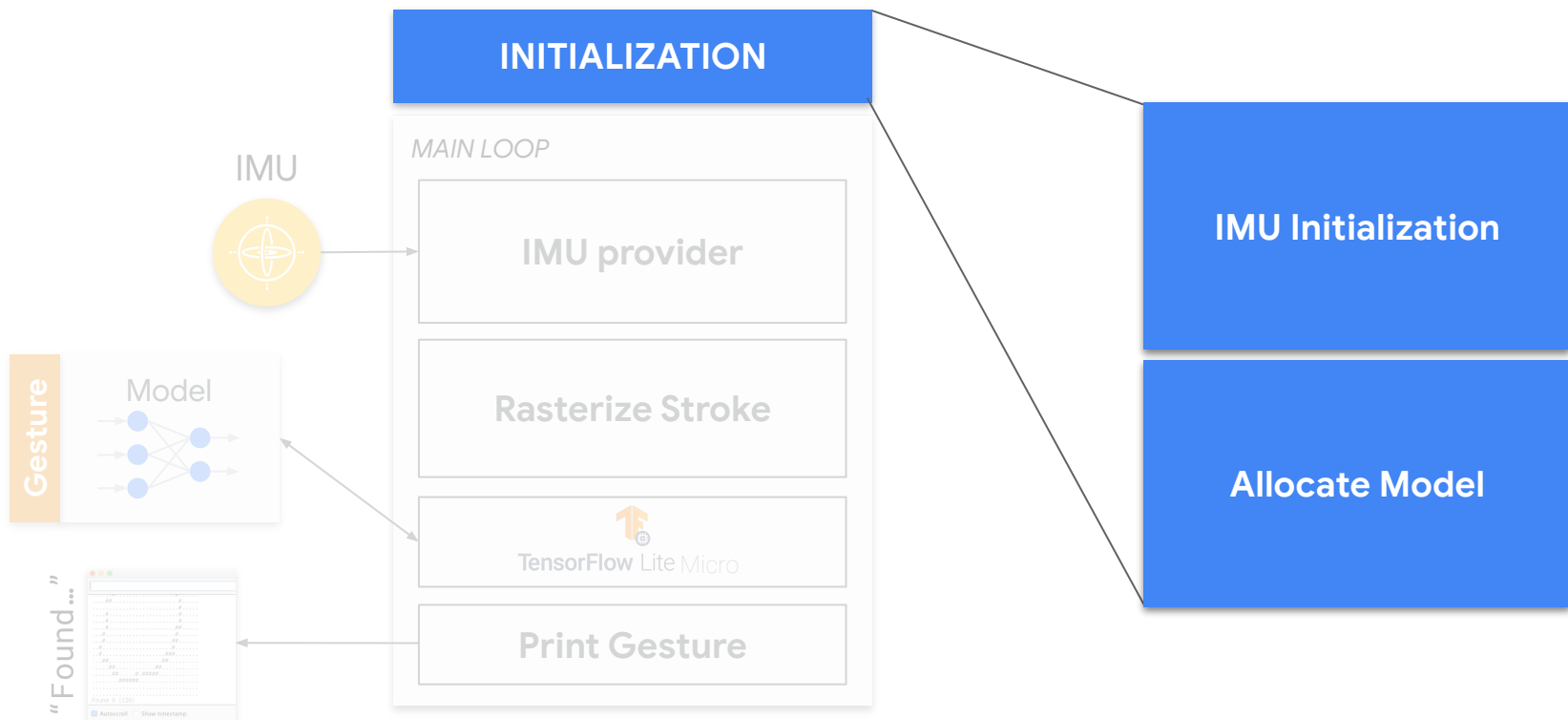
# Magic Wand Components



# Magic Wand Components



# Initialization



# Initialization

**IMU Initialization**

Allocate Model

```
// Start IMU
if (!IMU.begin()) {
    Serial.println("Failed to initialize IMU!");
    while (1);
}
SetupIMU();
```

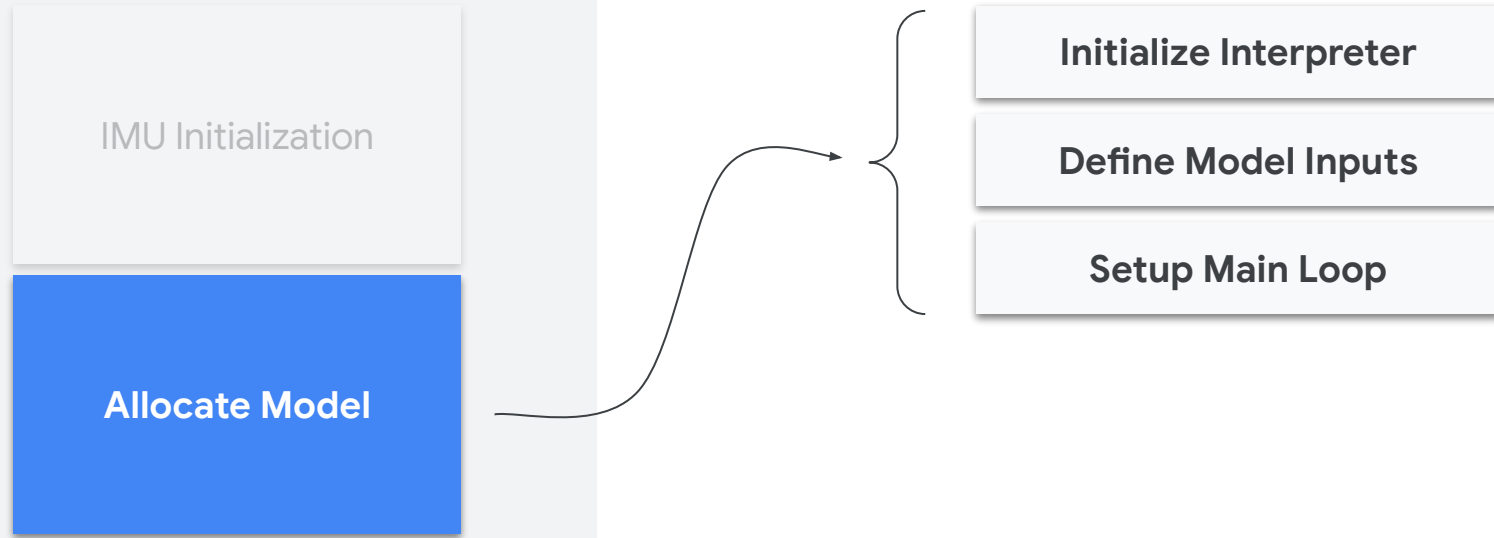
# Initialization

IMU Initialization

Allocate Model

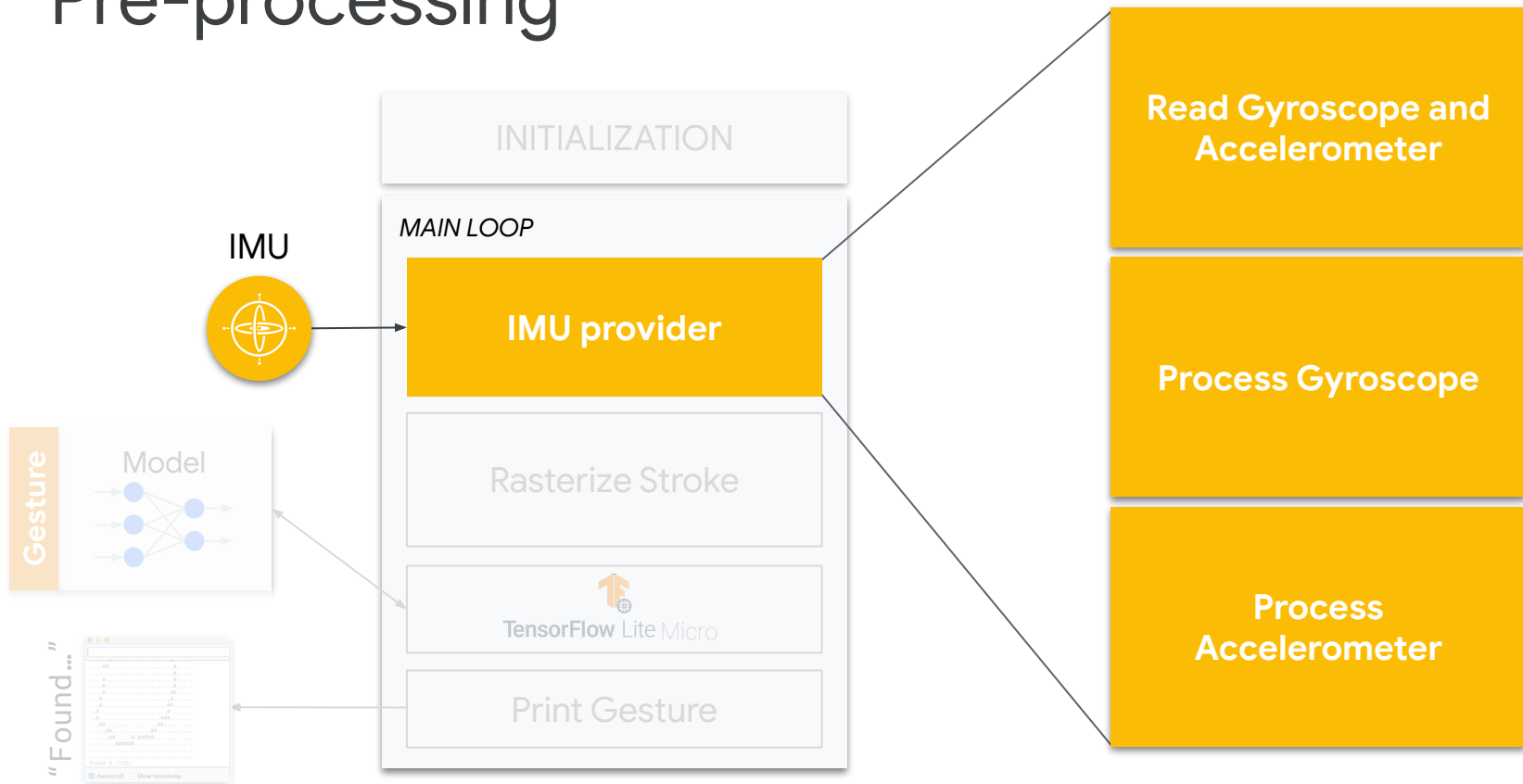
```
void SetupIMU() {  
  
    // Make sure we are pulling measurements into a FIFO.  
    IMU.setContinuousMode();  
  
    acceleration_sample_rate = IMU.accelerationSampleRate();  
    gyroscope_sample_rate = IMU.gyroscopeSampleRate();  
}
```

# Initialization





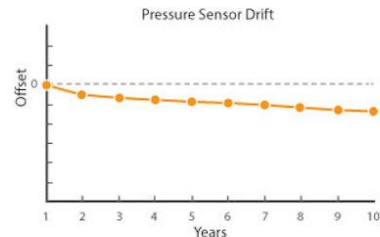
# Pre-processing





# Pre-processing

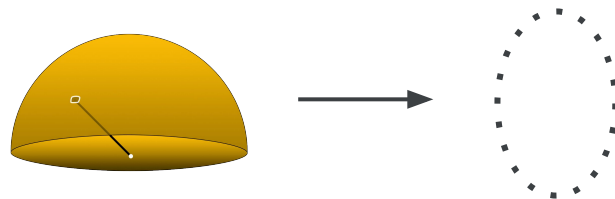
## Process Gyroscope

[illegible]



# Pre-processing

## Process Gyroscope

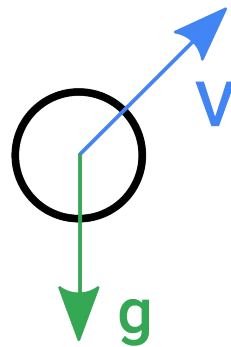
[illegible]

# Pre-processing

Read Gyroscope and  
Accelerometer

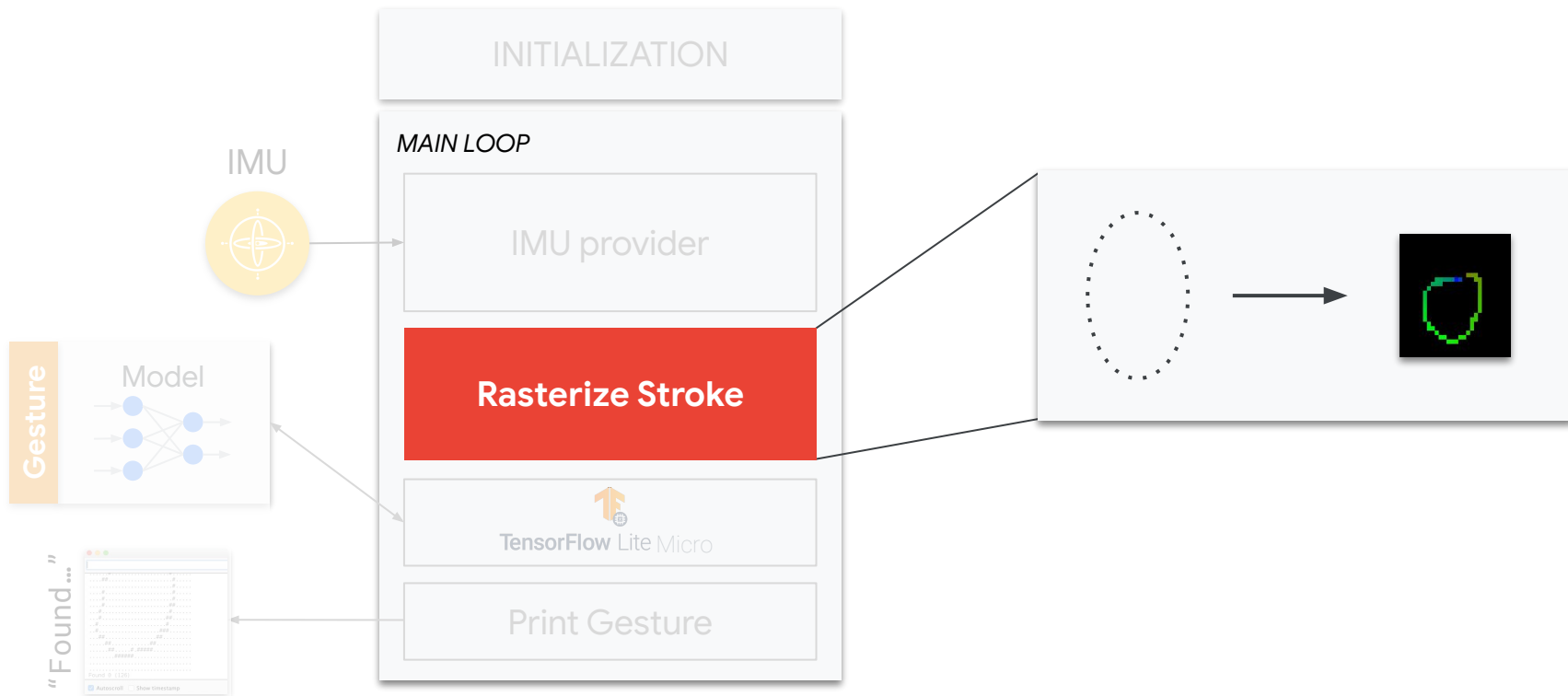
Process Gyroscope

**Process  
Accelerometer**



```
if (accelerometer_samples_read > 0) {  
    EstimateGravityDirection(current_gravity);  
    UpdateVelocity(accelerometer_samples_read, current_gravity);  
}
```

# Pre-processing



# Rasterize

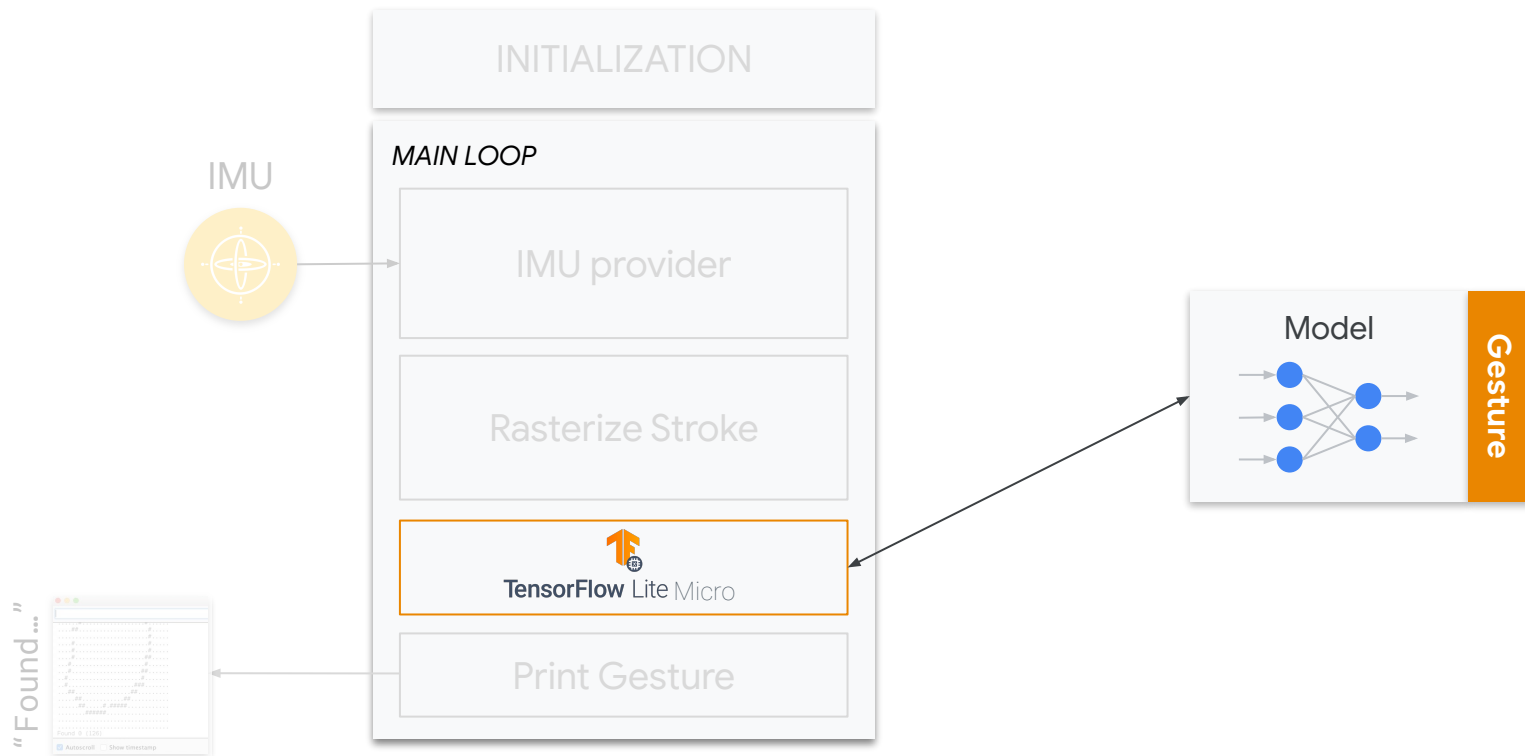
## Rasterize Stroke



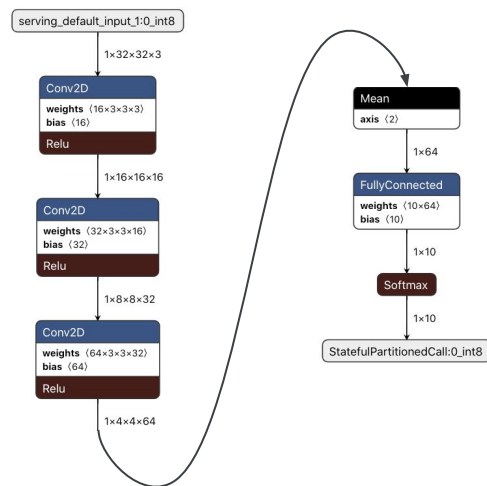
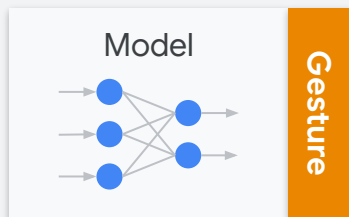
```
RasterizeStroke(stroke_points, *stroke_transmit_length, 0.6f,  
               0.6f, raster_width, raster_height, raster_buffer);
```



# Interpreter + Model

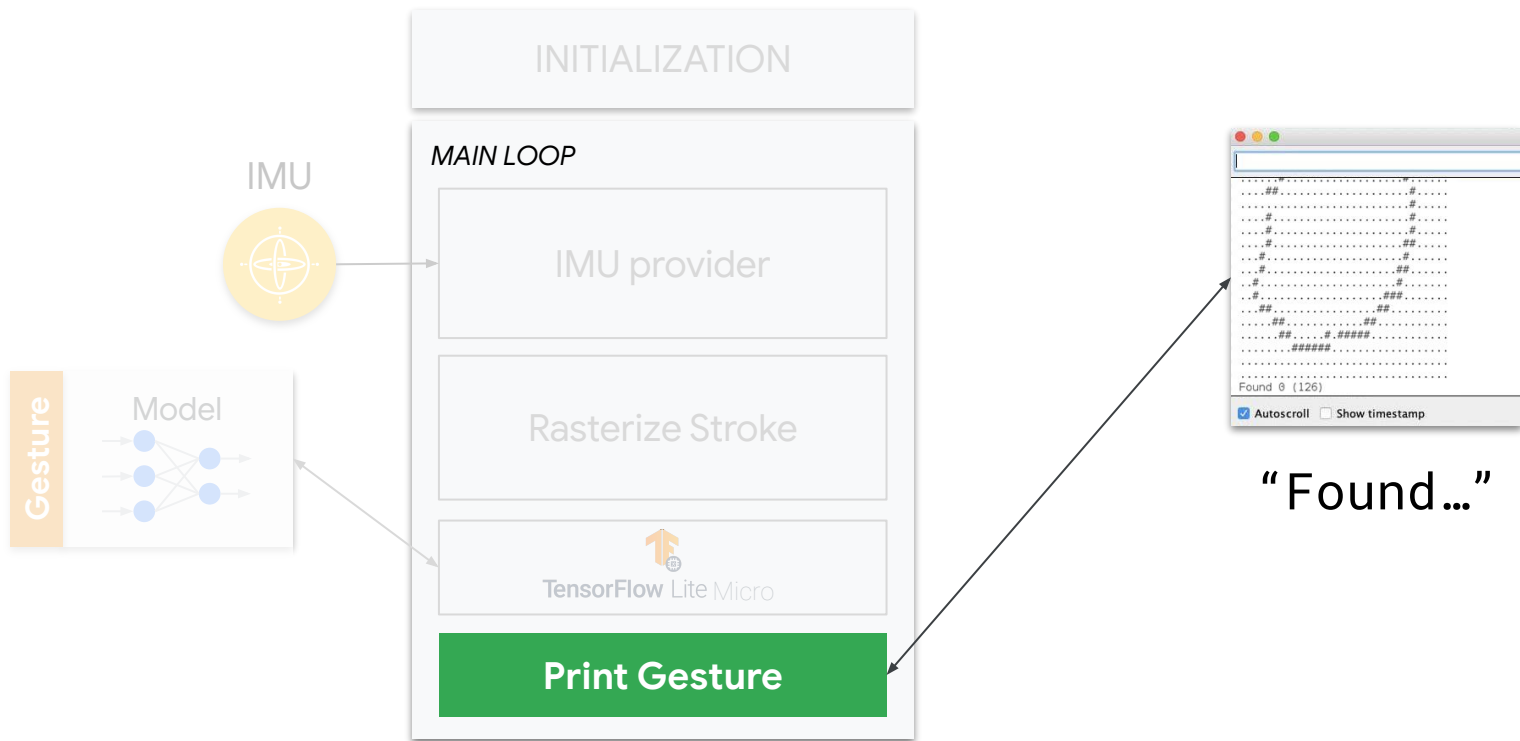


# Interpreter + Model



```
// Pass to the model and run the interpreter
TfLiteTensor* model_input = interpreter->input(0);
for (int i = 0; i < raster_byte_count; ++i) {
    model_input->data.int8[i] = raster_buffer[i];
}
TfLiteStatus invoke_status = interpreter->Invoke();
```

# Output



# Output

Print Gesture

```
// Parse the model output
int8_t max_score;
int max_index;
for (int i = 0; i < label_count; ++i) {
    const int8_t score = output->data.int8[i];
    if ((i == 0) || (score > max_score)) {
        max_score = score;
        max_index = i;
    }
}
TF_LITE_REPORT_ERROR(error_reporter, "Found %s
(%d)", labels[max_index], max_score);
```