# Amazon FreeRTOS

## IoT Operating System for Microcontrollers

# Introduction

**Richard Barry**

Founder – FreeRTOS Project

Principal Engineer – Amazon Web Services

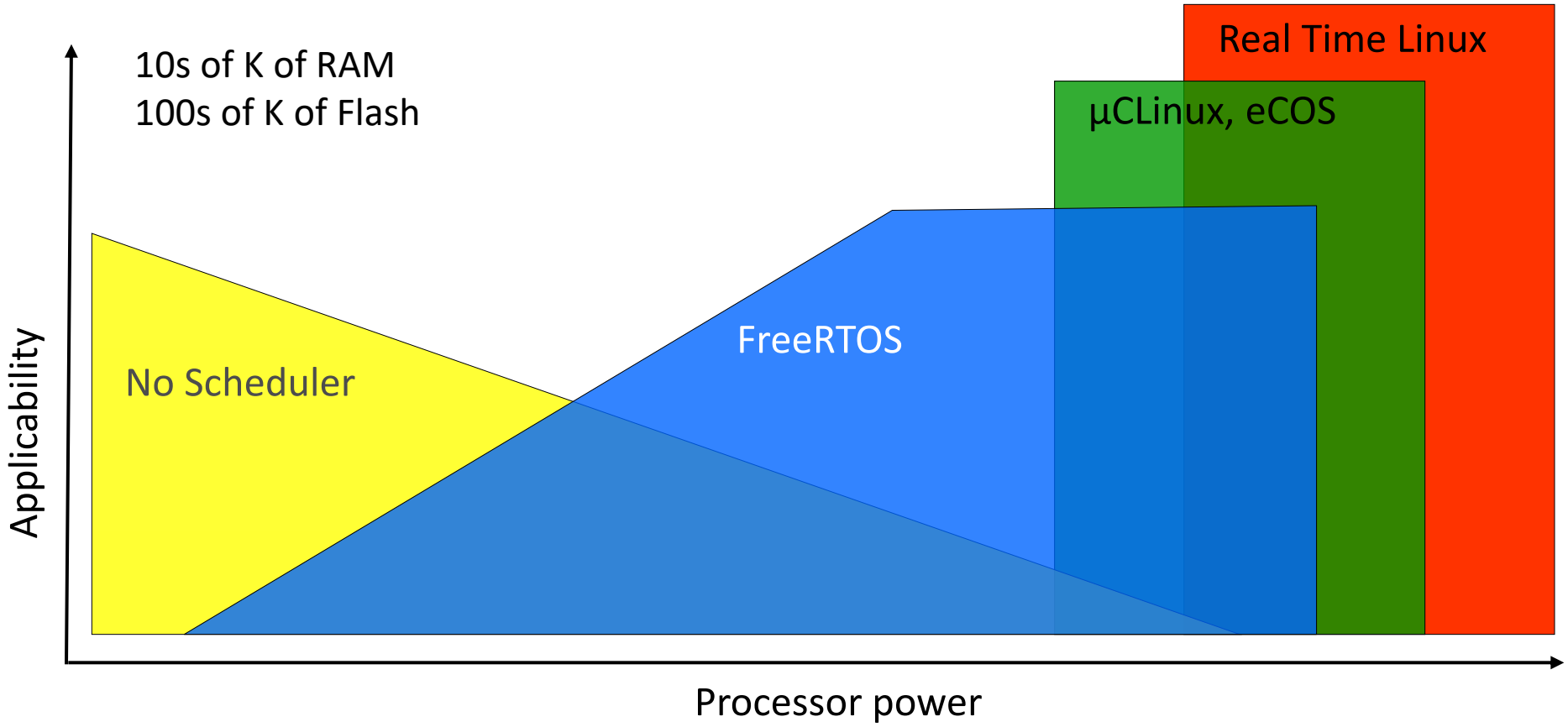
# Agenda

- The FreeRTOS kernel, and Traditional Use Cases
  - Design Goals
  - Differences from General Purpose Operating Systems
- IoT Relevant AWS Cloud Services
- Amazon FreeRTOS Libraries
- Amazon FreeRTOS, Connected Use Cases

# (Very Brief) Introduction to the FreeRTOS Kernel

- It is a kernel, or scheduler, or RTOS, depending who you ask
- 15 Year Heritage
- MIT Licensed C Code and Pre-configured Projects

# For MCUs



# Technical Deep Dive? Removing Adoption Blockers

Rapid Support

Leadership

Documentation

Windows Host

Customer Obsession

No IP Infringement

Paid Options

Demonstrable Code Quality

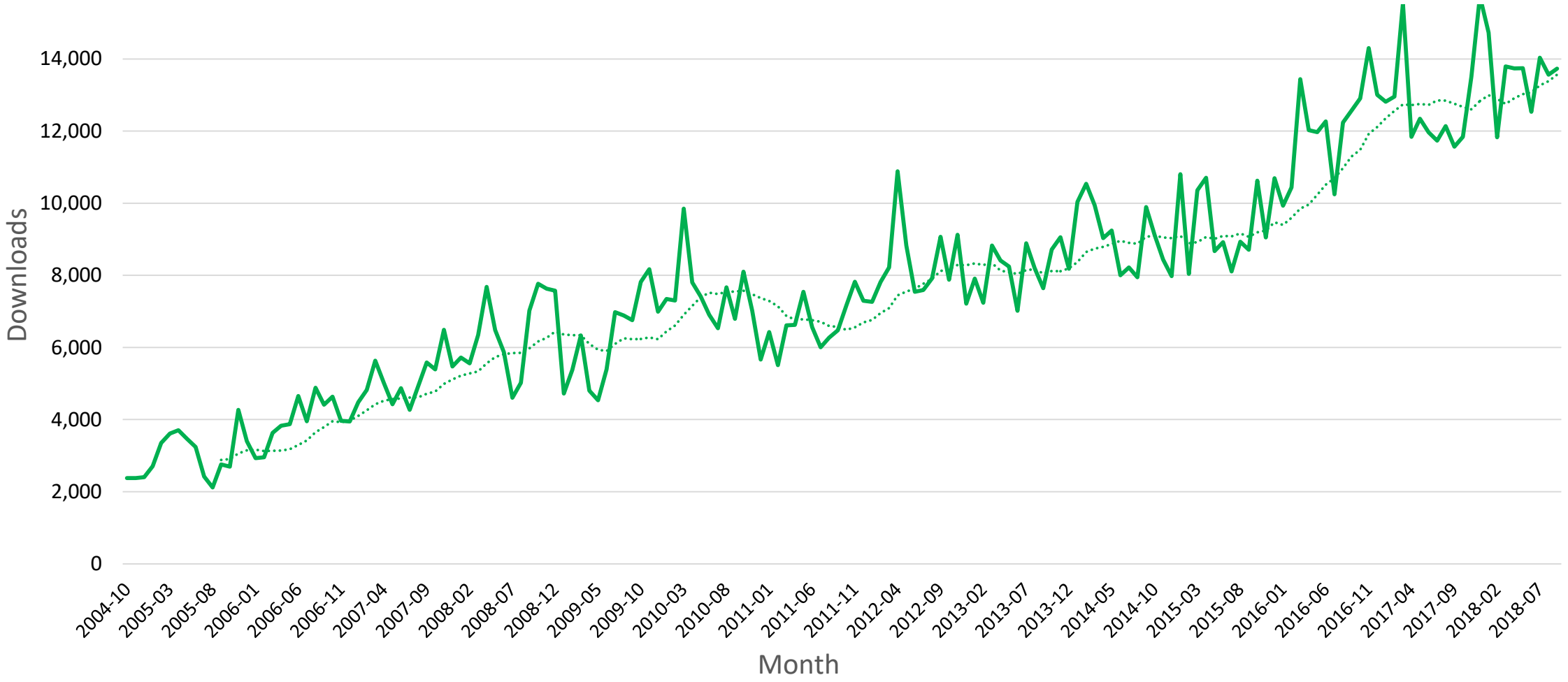
Robustness

Knowledgeable Support

Visible Activity

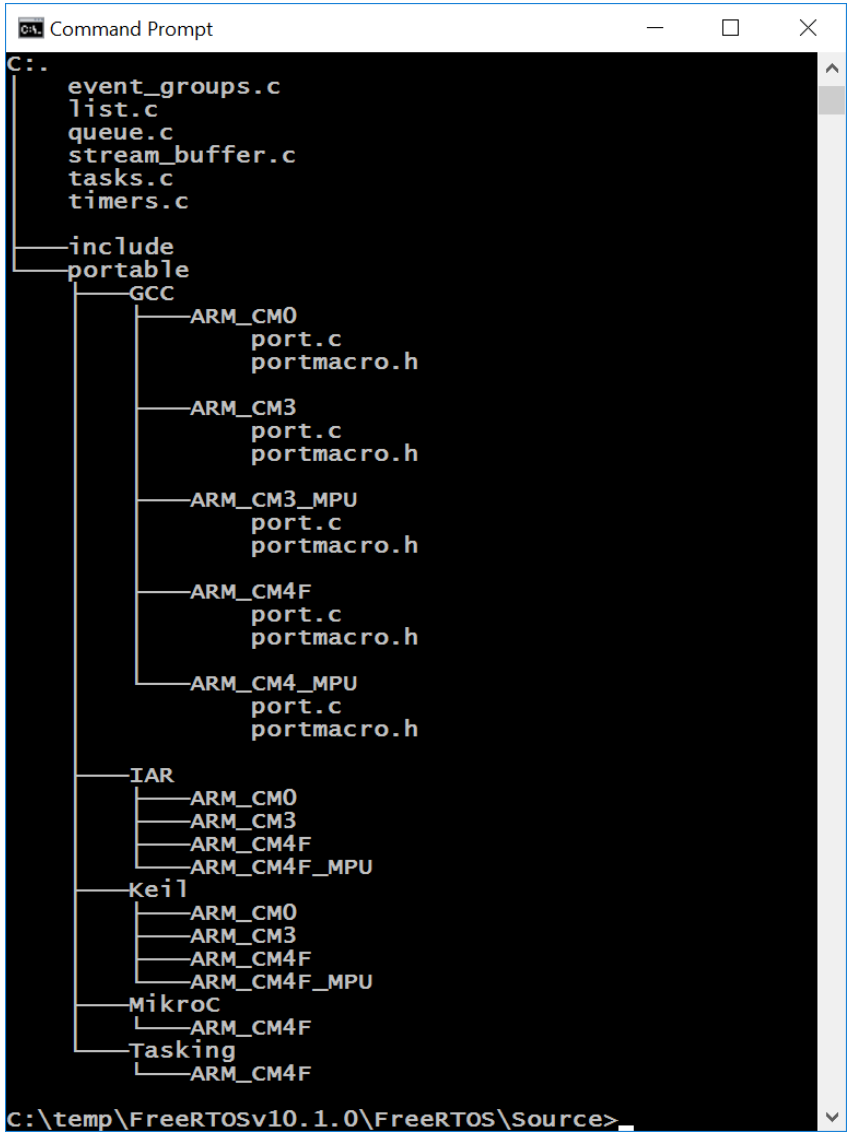
Enterprise Friendly Licensing

# Downloads Per Month

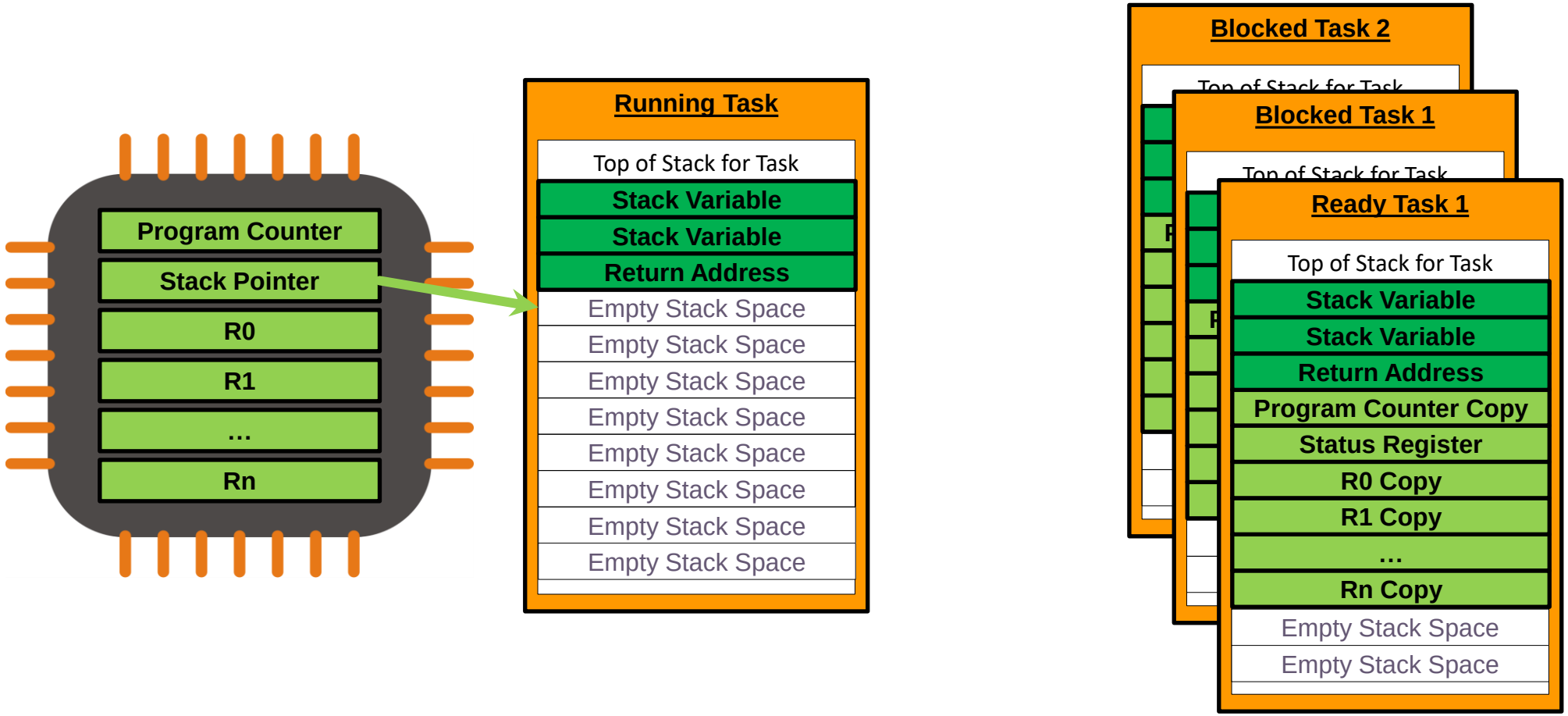


# MIT Licensed C Source Code

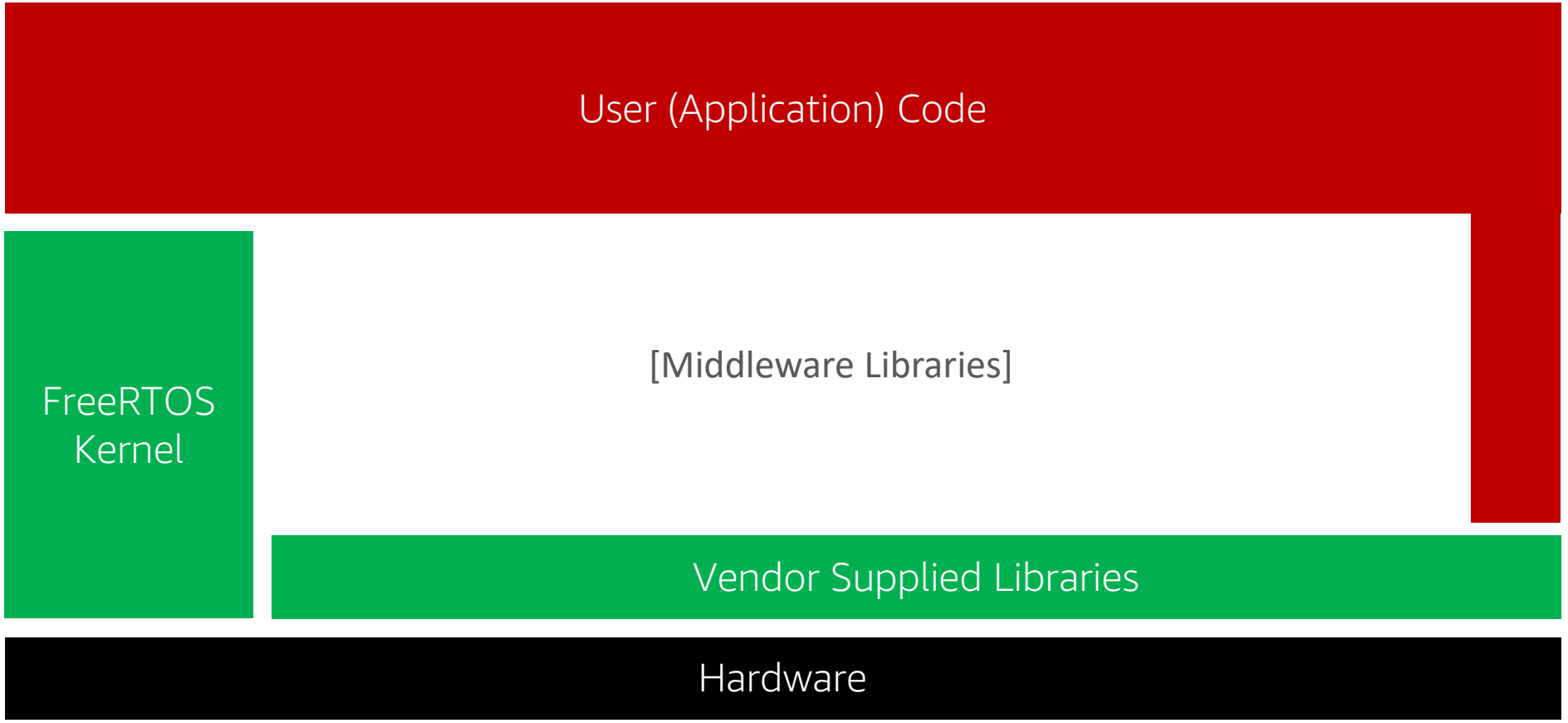
- Source File Downloads
  - FreeRTOS.org
    - SVN
  - aws.amazon.com/FreeRTOS
    - GitHub



# Port Layer

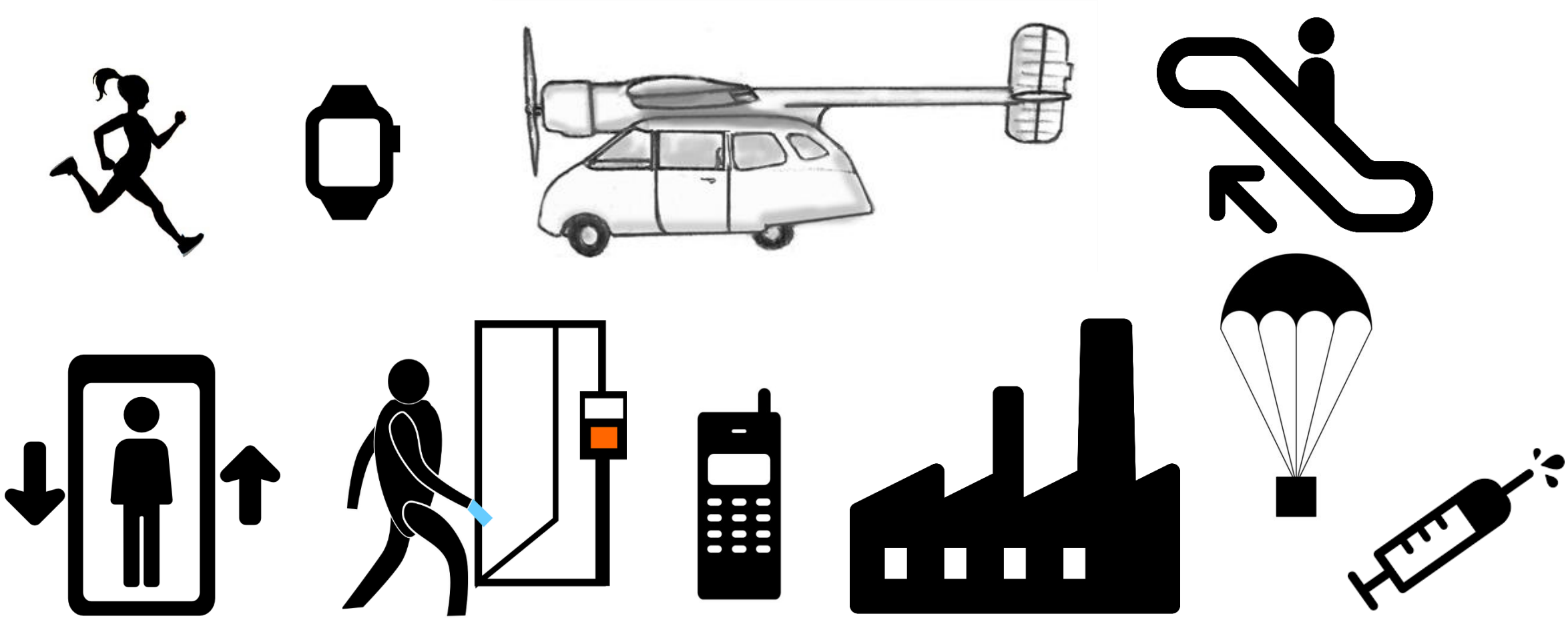


# Usage Model



# Use Cases

# Use Cases – Syringe Pumps

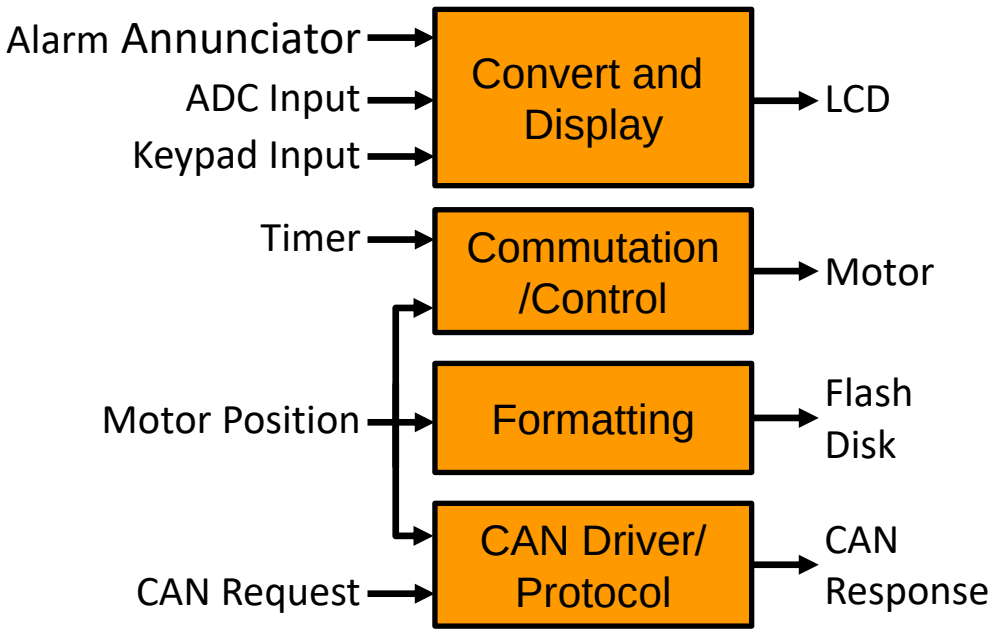


# Why?

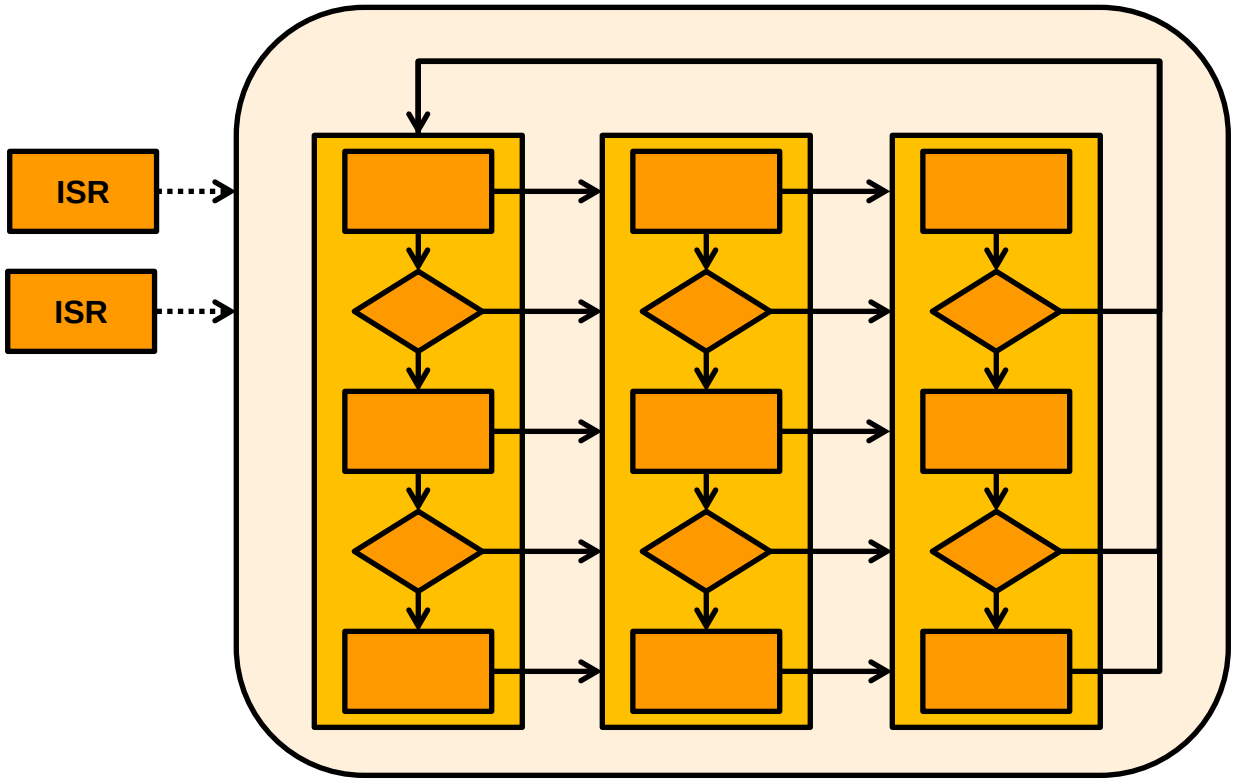
# Application Design Goals

- Depends on the application, but generally:
  - Meet real time requirements!
  - Maximize responsiveness
  - Use as little CPU/Power as possible
  - Maximize maintainability
  - Maximize portability (hardware change)
  - Simplicity!
  - Fast to market
  - Meet requirements with minimum expenditure

# Physical Challenges



# Superloops, Foreground, Background



# RTOS Operation

```
Function1_Run();
```

```
Function3_Run();
```

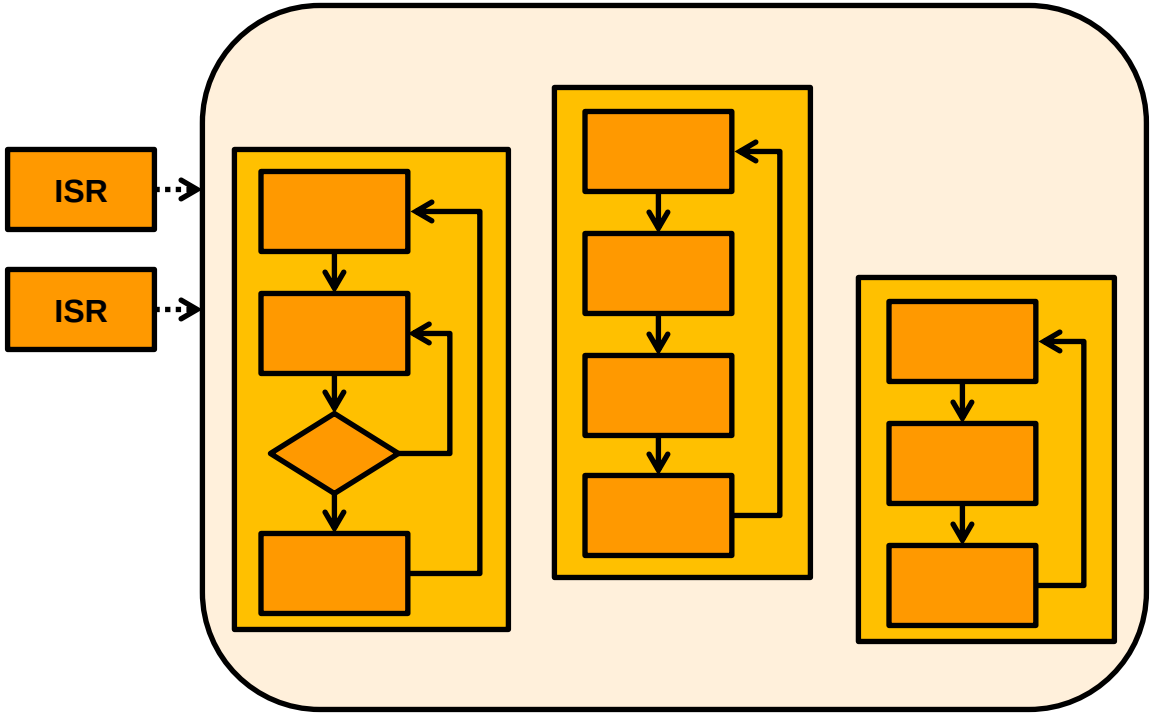
```
Function2_Run();
```

```
Function4_Run();
```

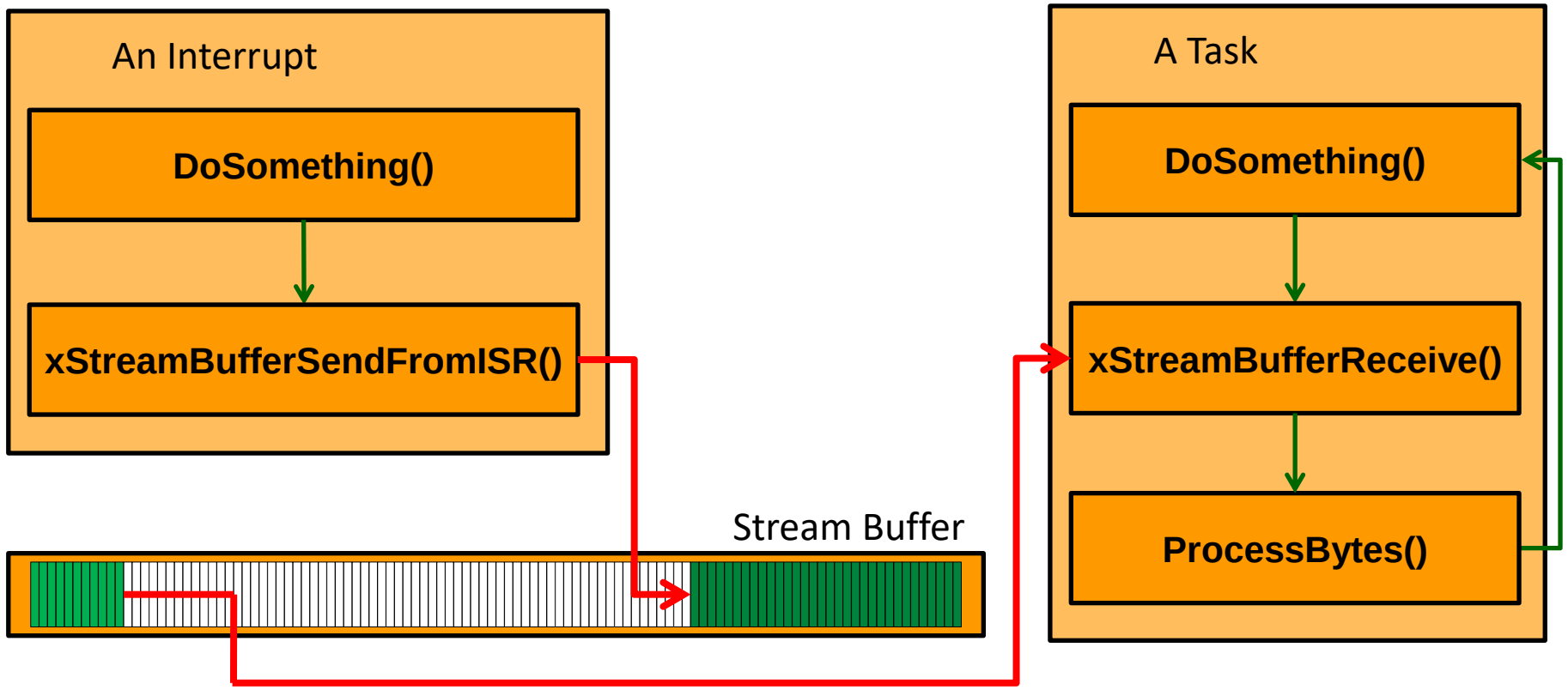
## Application Design Goals

- Depends on the application, but generally:
  - Meet real time requirements!
  - Maximize responsiveness
  - Use as little CPU/Power as possible
  - Maximize maintainability
  - Maximize portability (hardware change)
  - Simplicity!
  - Fast to market
  - Meet requirements with minimum expenditure

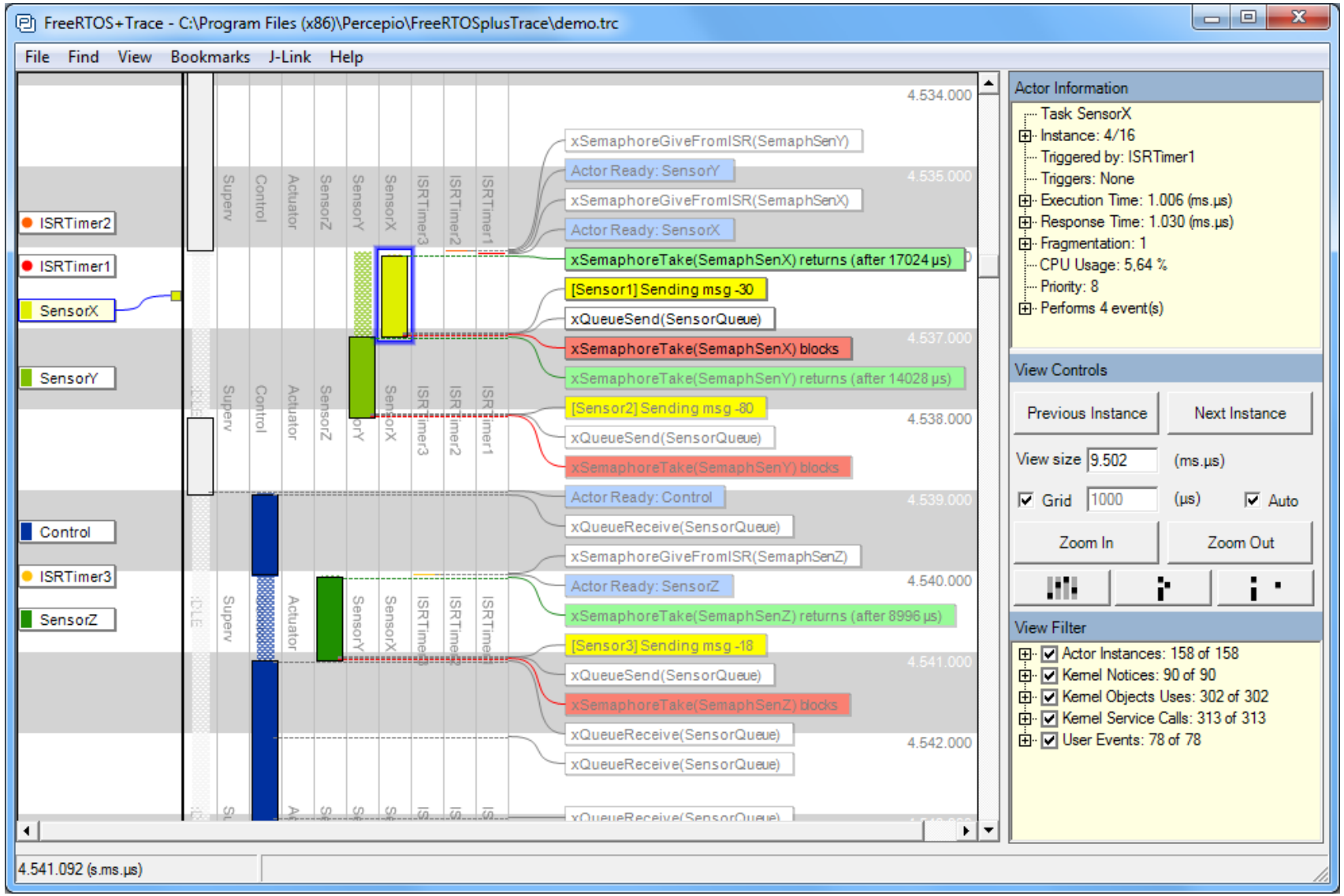
# Multithreaded Design



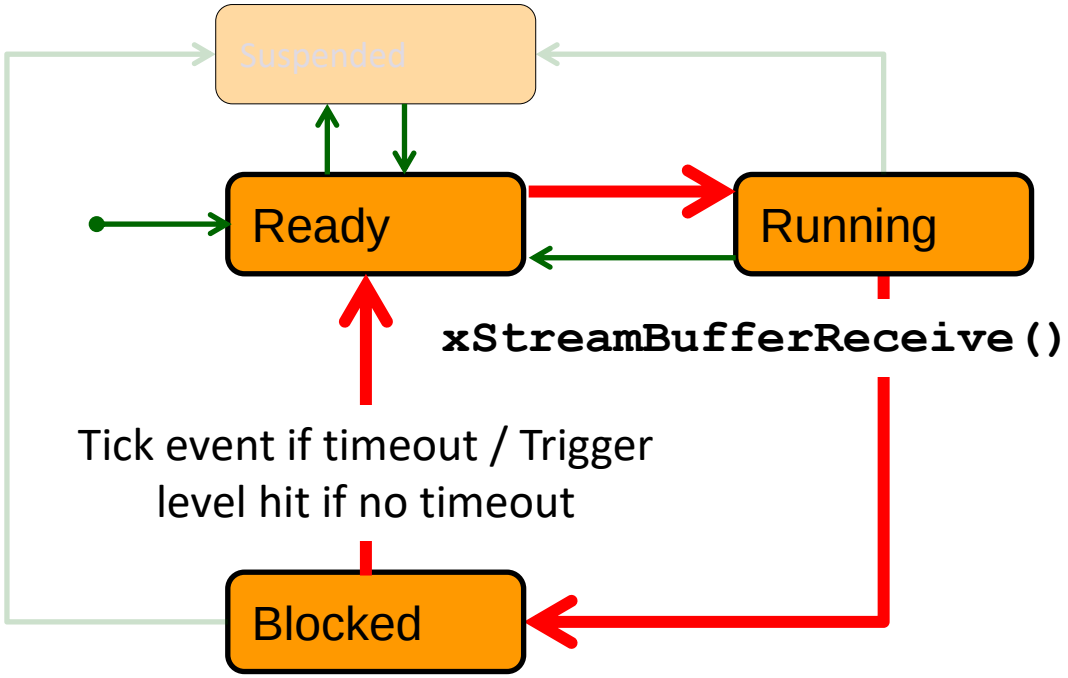
# Maximizing Utilization, Minimizing Power Consumption

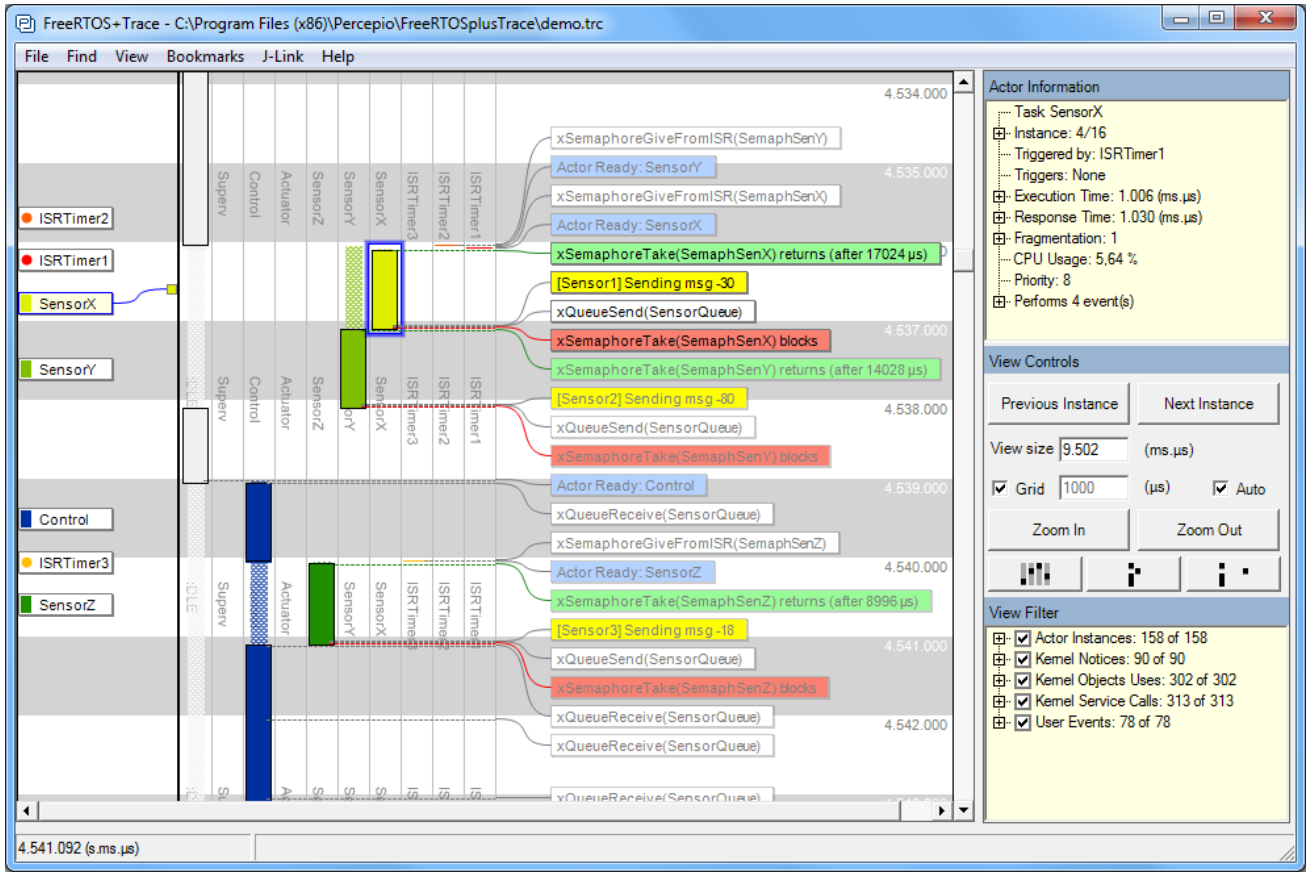


# Improved CPU Utilization



# Improved CPU Utilization

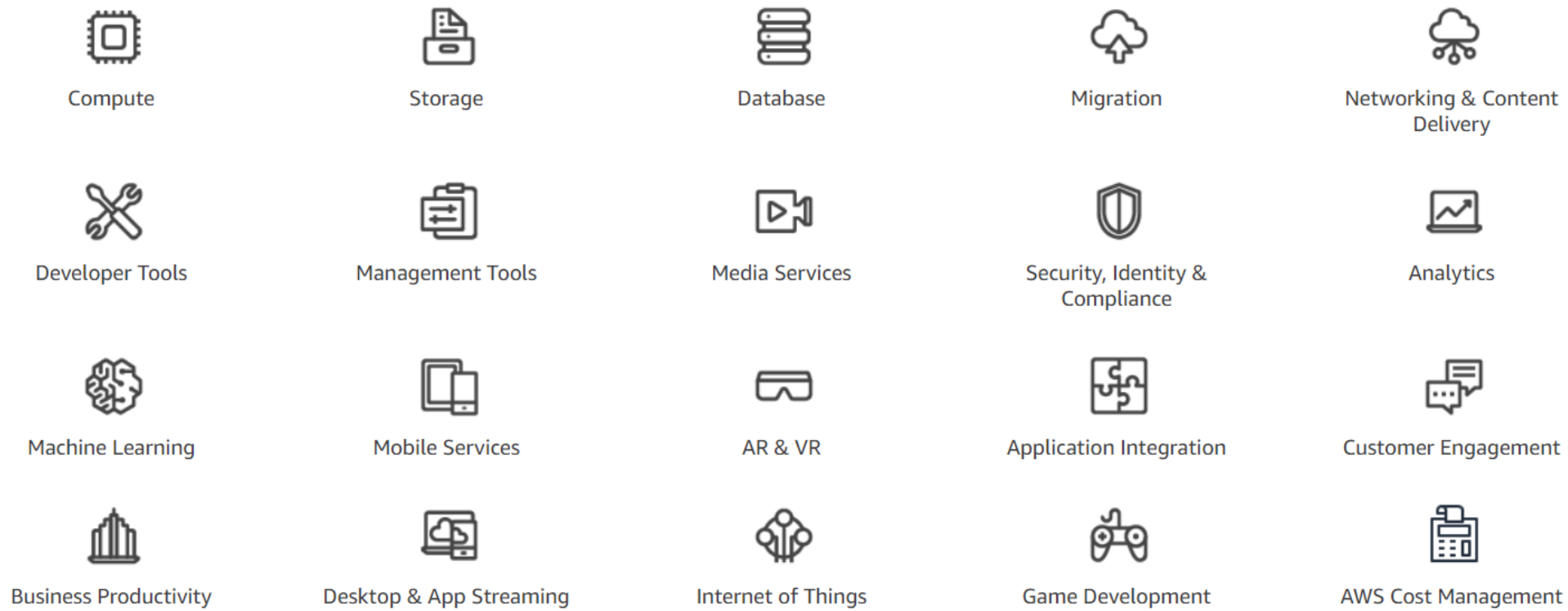


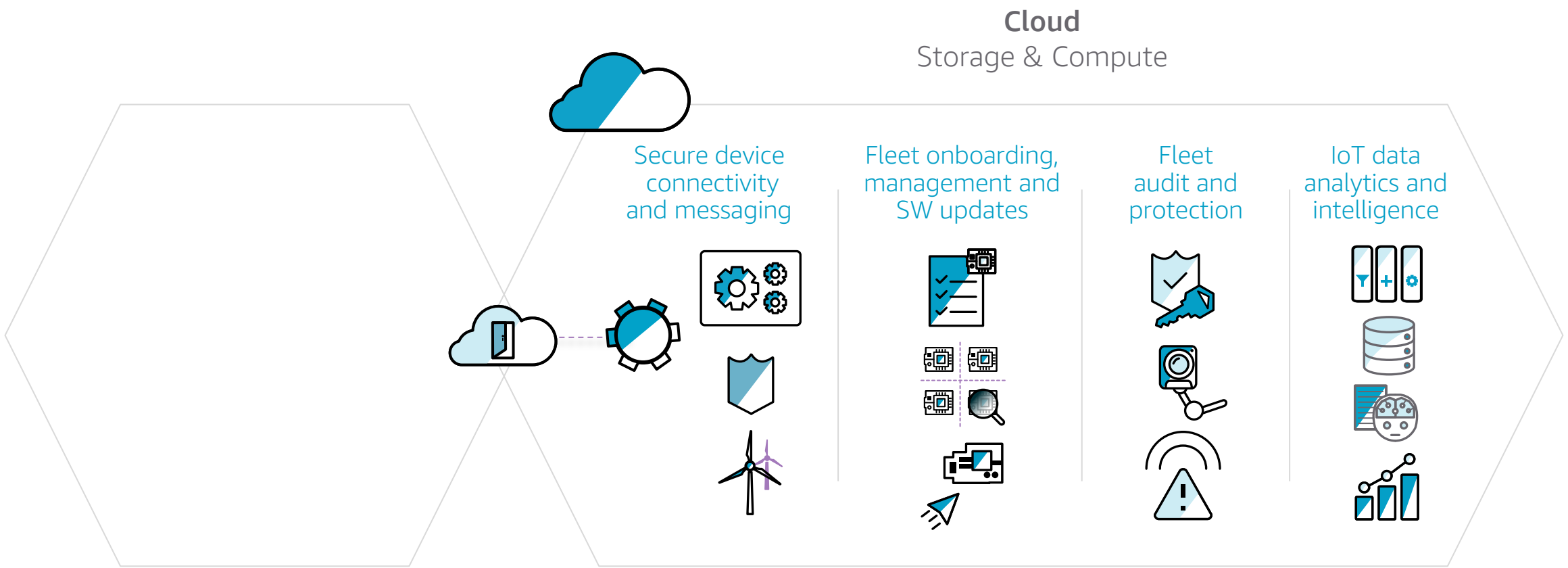
# Introduction to Amazon FreeRTOS

# AWS – Cloud Computing

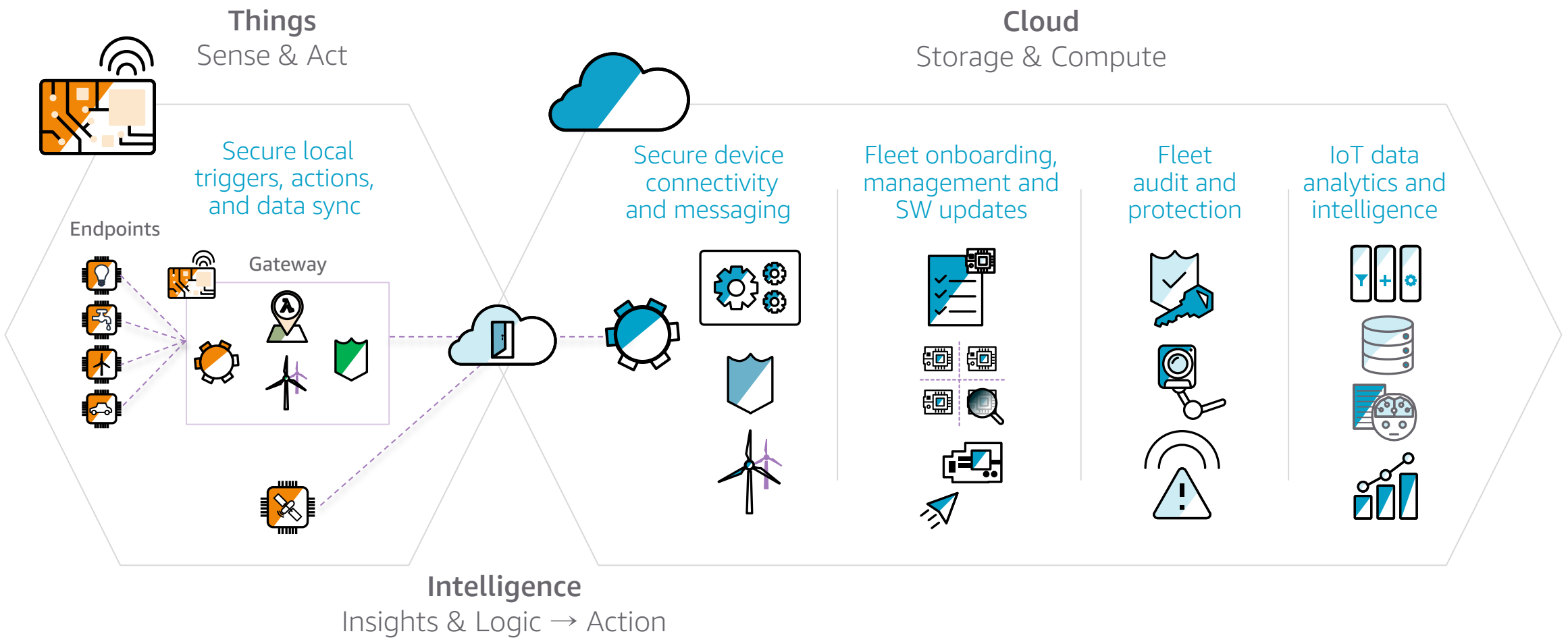
- On-demand delivery of compute power, databases, applications, etc. via the internet, >125 services
- Pay-as-you-go pricing



# AWS IoT - In the Cloud

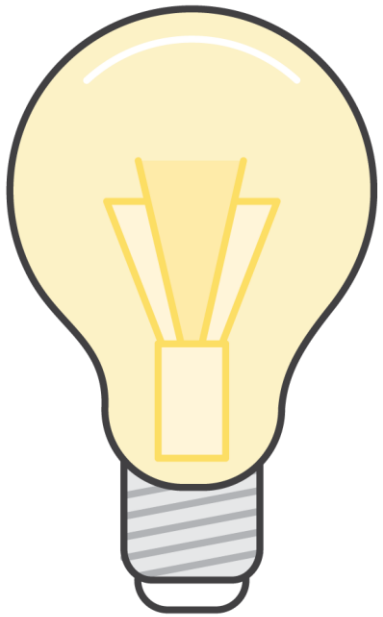


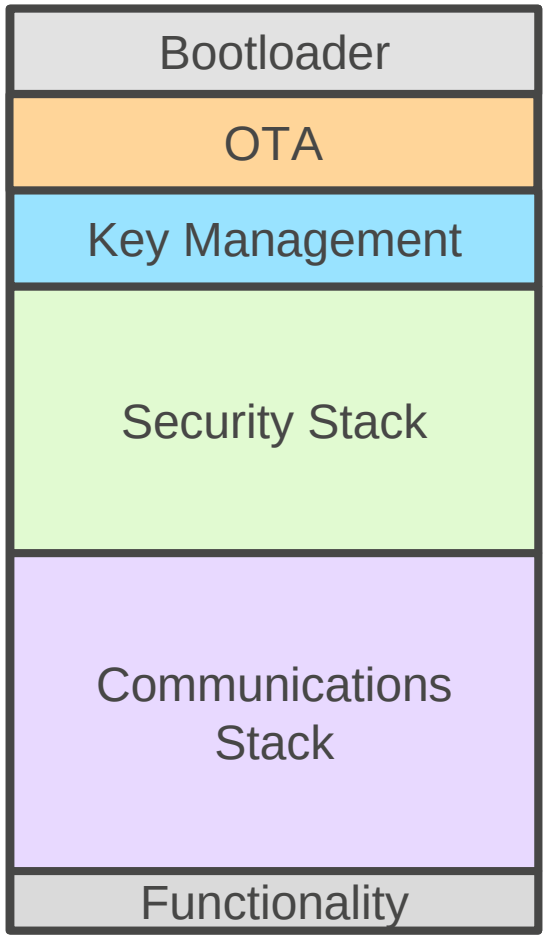
# AWS IoT – Cloud and Edge



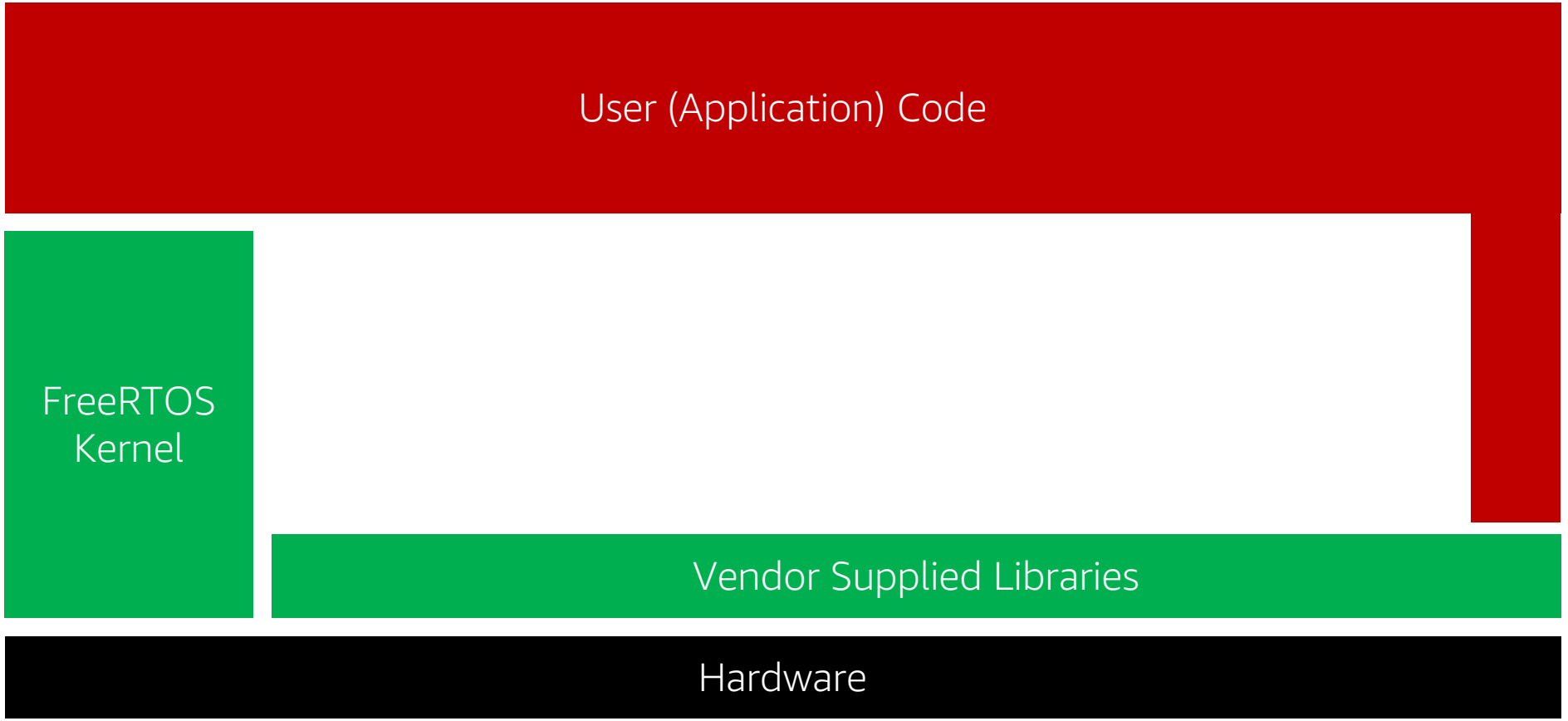
# Why Amazon FreeRTOS?

# Primary Functionality Vs Security, Connectivity

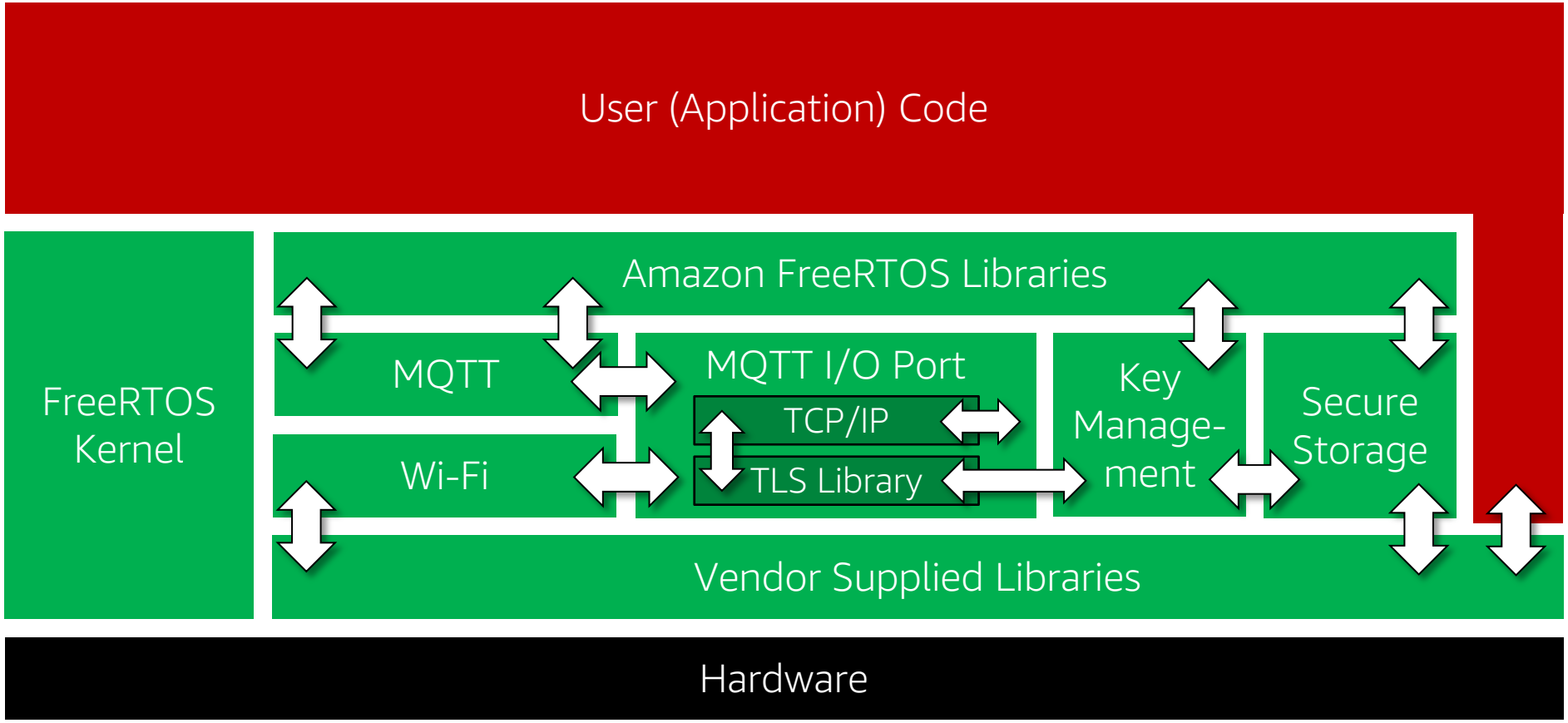




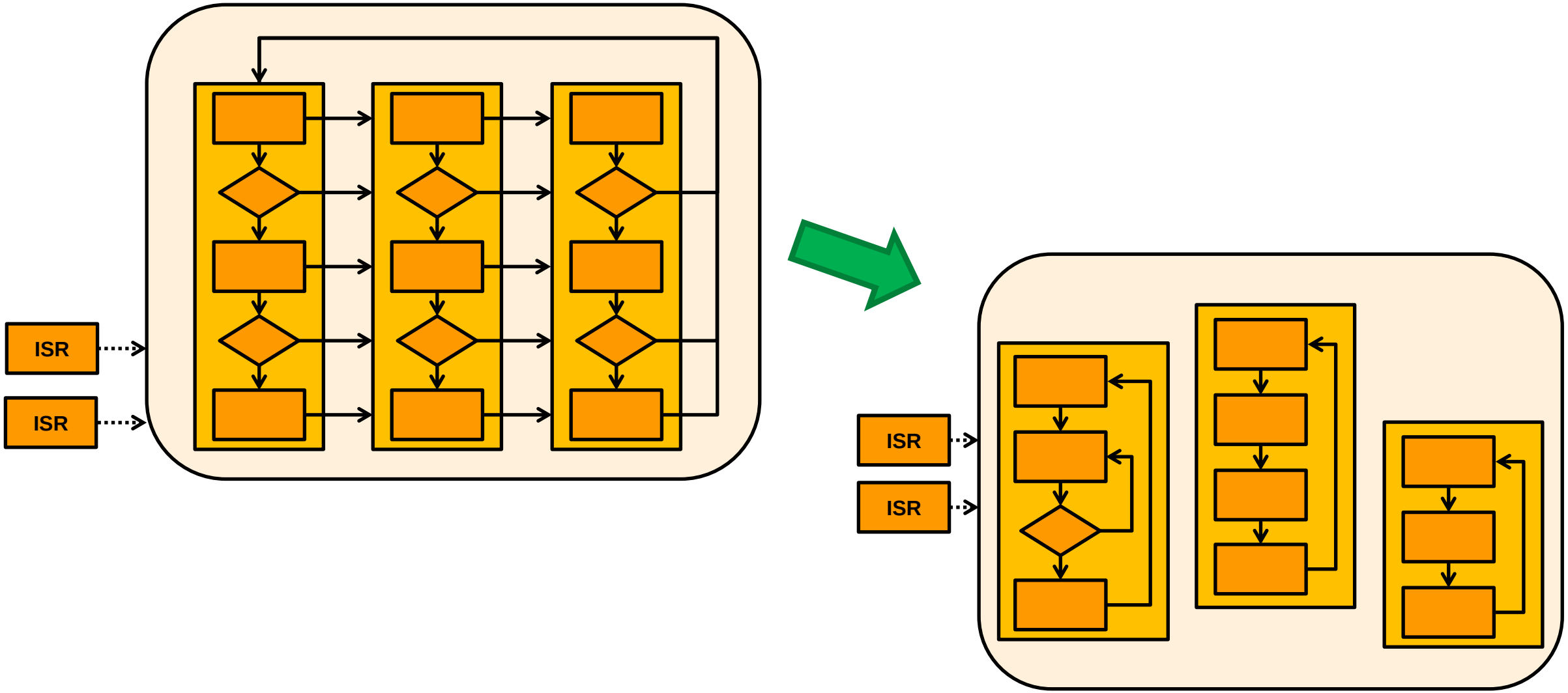
# The FreeRTOS Kernel



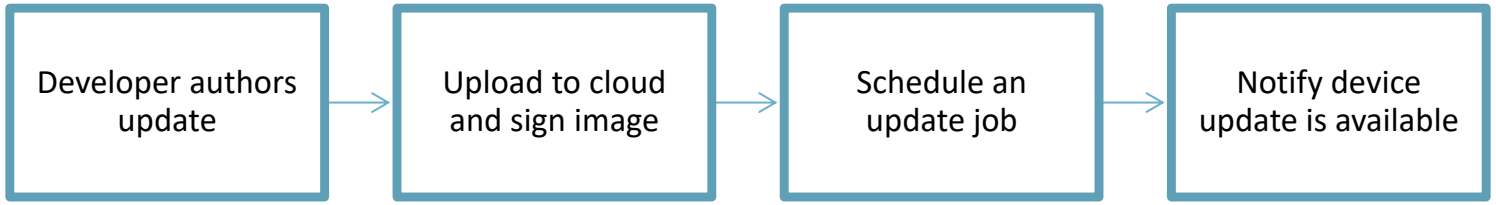
# Amazon FreeRTOS – Kernel and Libraries
# Library Porting View



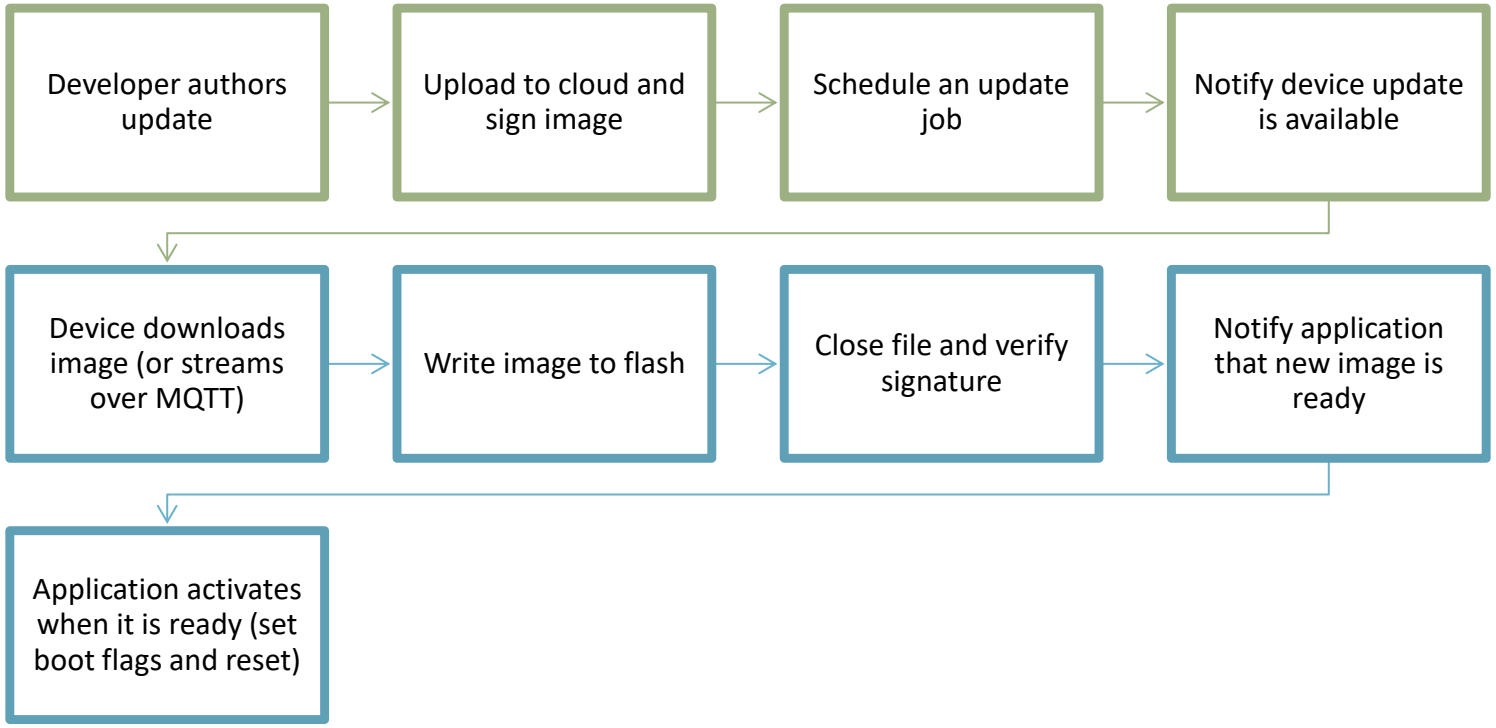
# Design Goals



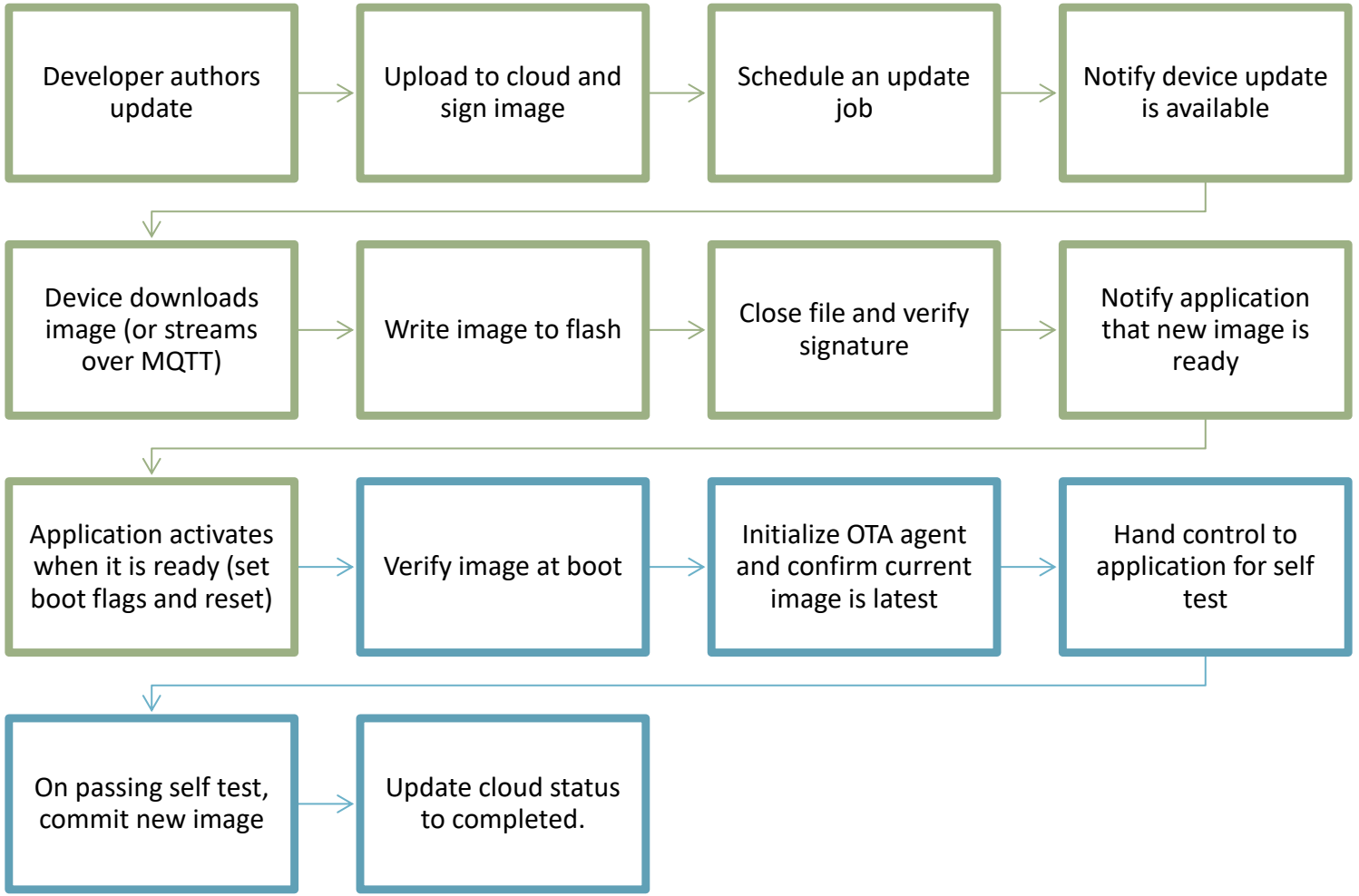
# OTA on Amazon FreeRTOS – Operator Actions



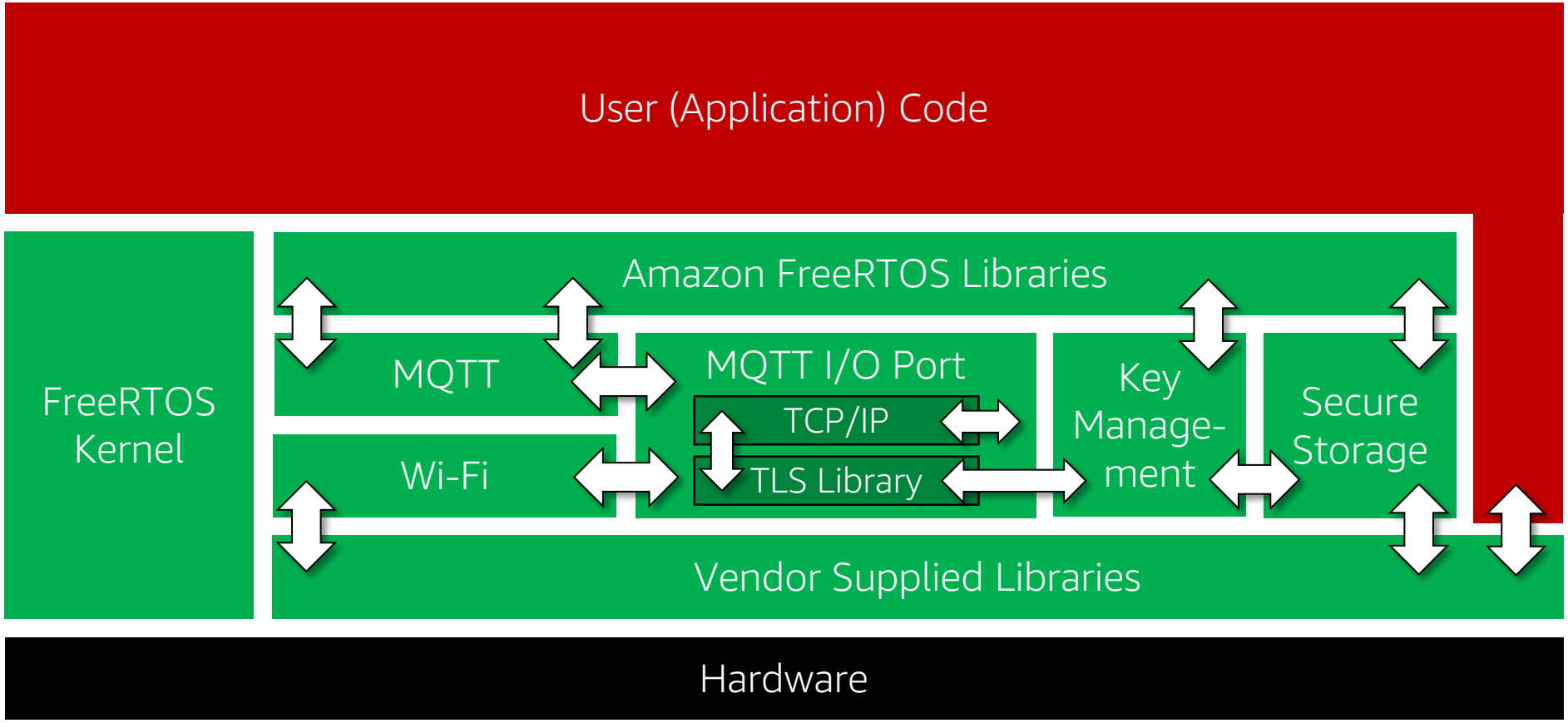
# OTA on Amazon FreeRTOS – OTA Agent Actions



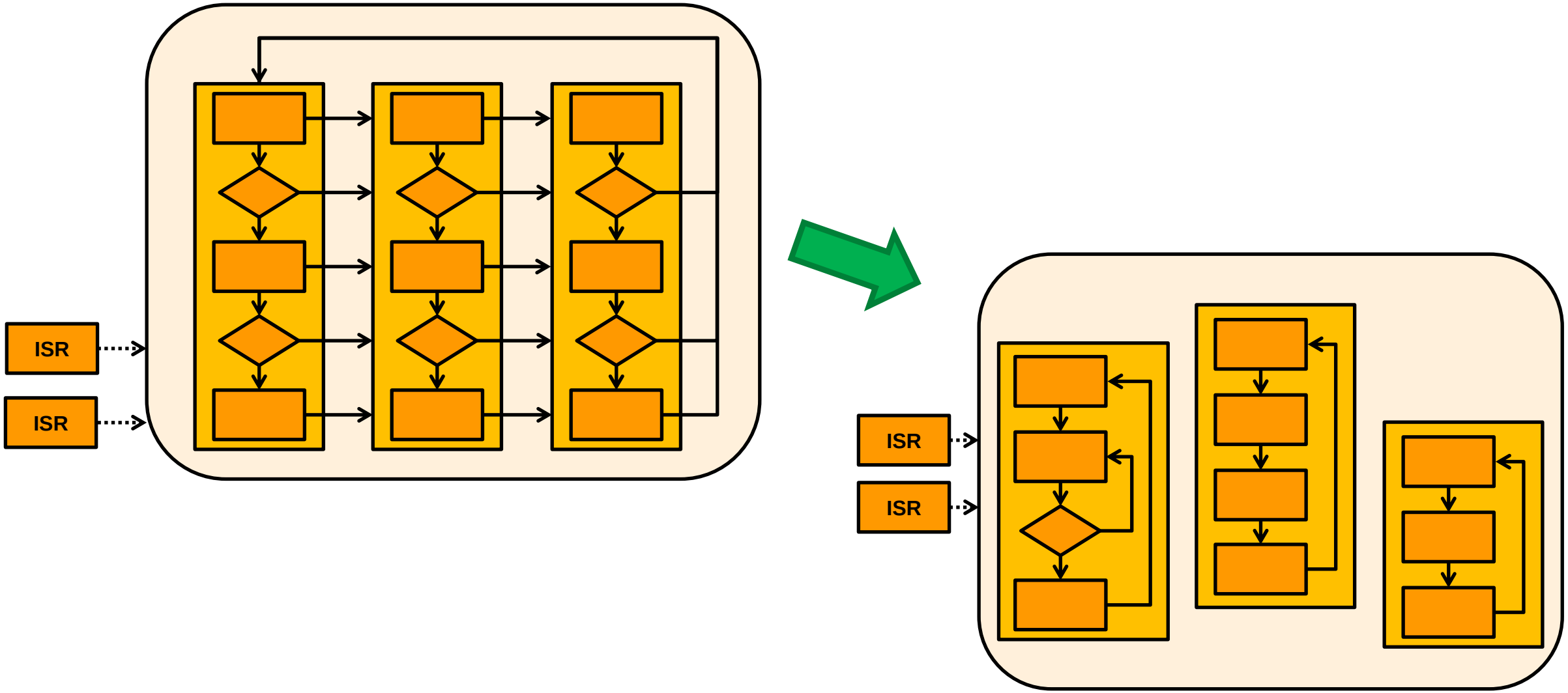
# OTA on Amazon FreeRTOS – Bootloader Actions



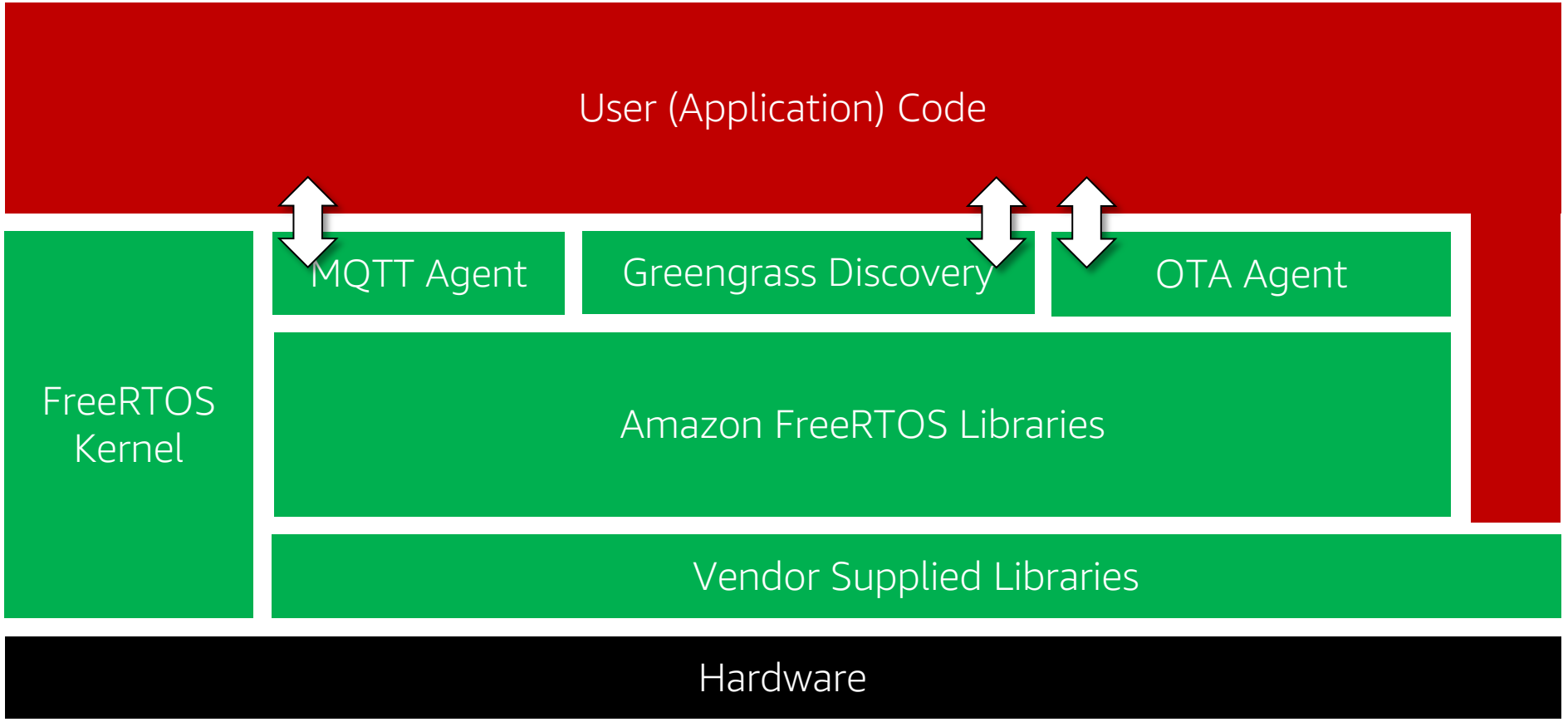
# Amazon FreeRTOS – Kernel and Libraries
# Library Porting View



# Design Goals



# Amazon FreeRTOS – Kernel and Libraries
# Application Writer's View



# Predictive Maintenance Example

## Detect Anomalies to Predict Pump Failure


1. Run Amazon FreeRTOS on vibration sensors to securely collect data and connect to AWS Greengrass enabled device
2. The AWS Greengrass enabled device runs the predictive model locally to identify when vibrations hit dangerous levels. AWS Greengrass triggers alert to maintenance staff when anomalies are detected. When Internet connectivity is available, the AWS Greengrass device sends data to the cloud for analytics filtering out “normal” data
3. AWS IoT Analytics analyzes vibration data and adds time stamp and device information such as serial number pulling from AWS IoT Core. Sends updated model to the AWS Greengrass enabled device

**Predictive maintenance**

Predictive quality

Asset condition monitoring

# Thank You!

- https://www.FreeRTOS.org
- https://freertos.org/FreeRTOS-quick-start-guide.html
- https://www.freertos.org/Documentation/RTOS_book.html
- https://aws.amazon.com/freertos

# Predictive Quality Example


## Predict Crop Quality

1. Soil sensors measure PH, moisture, nutrients, and gases

2. AWS IoT Analytics enriches soil sensor data with geolocation, rainfall, and weather information and predicts crop health and quality. Makes suggestions on watering and fertilization schedule to increase crop yield

Predictive maintenance

**Predictive quality**

Asset condition monitoring

# Asset Condition Monitoring

Global Mining
Company Minimizes Unexpected Downtime

**Problem**
Needs to understand degradation equipment

**Solution**
Collect data from equipment, identify potholes and other problems on mining routes that can contribute to degradation

**Impact**
Monitor equipment status, health, and performance to detect issues in real-time. Detect road issues and identify equipment degradation and minimize downtime

Predictive maintenance

Predictive quality

**Asset condition monitoring**