



FreeRTOS on RISC-V

Richard Barry
Founder, FreeRTOS Project
Principal Engineer, AWS IoT

Agenda

The FreeRTOS Kernel

Running FreeRTOS on RISC-V



FreeRTOS—Open source real time kernel

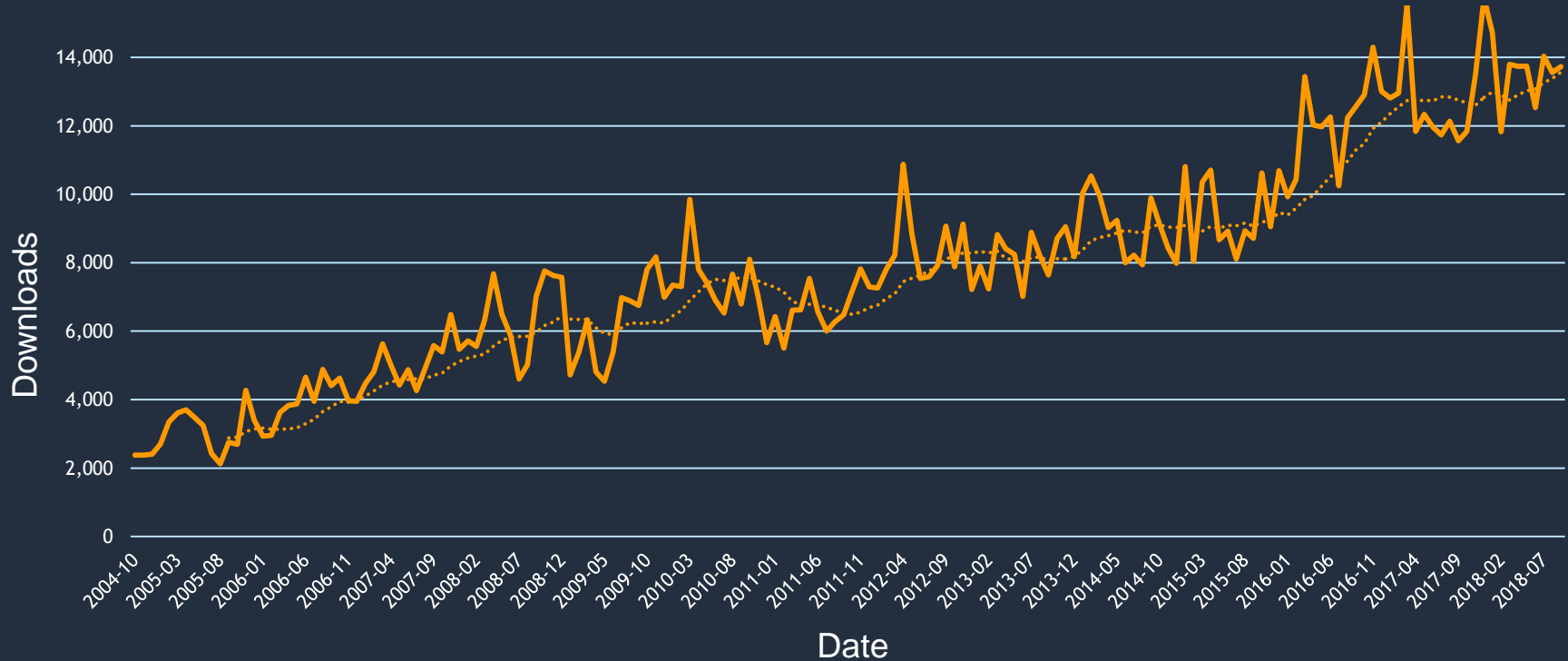


is everywhere...

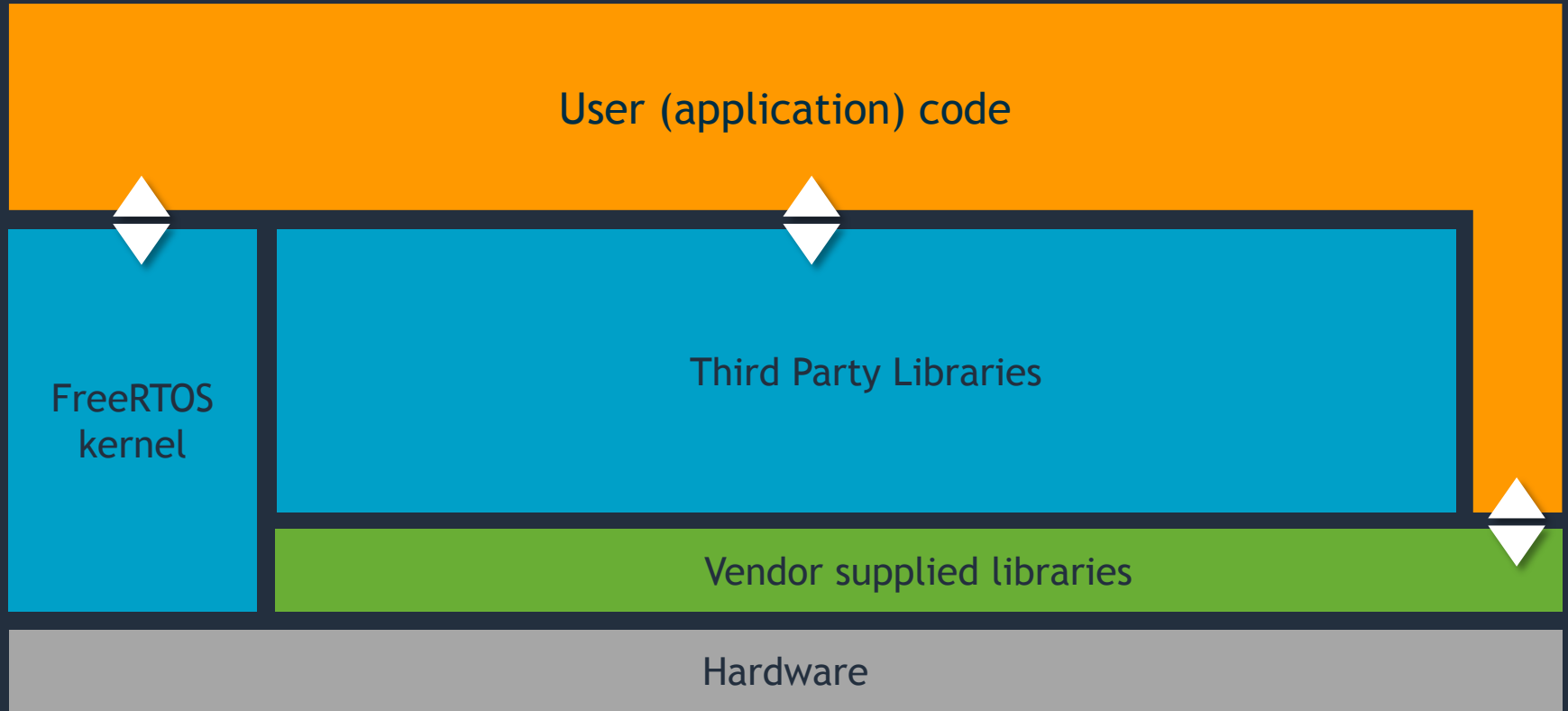


FreeRTOS downloads per month over 15 years

Official RISC-V port released February 2019



FreeRTOS kernel



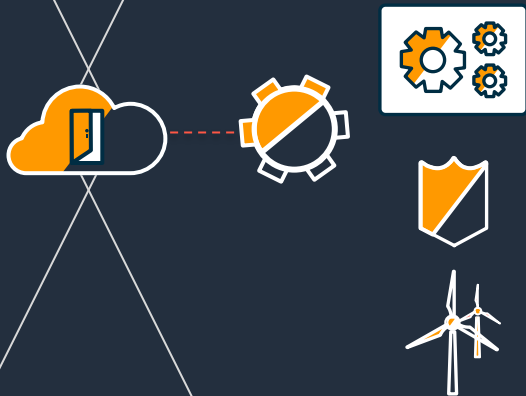
Use cases



Recent trend - IoT



Secure device connectivity and messaging



Fleet onboarding, management and SW updates



Fleet audit and protection

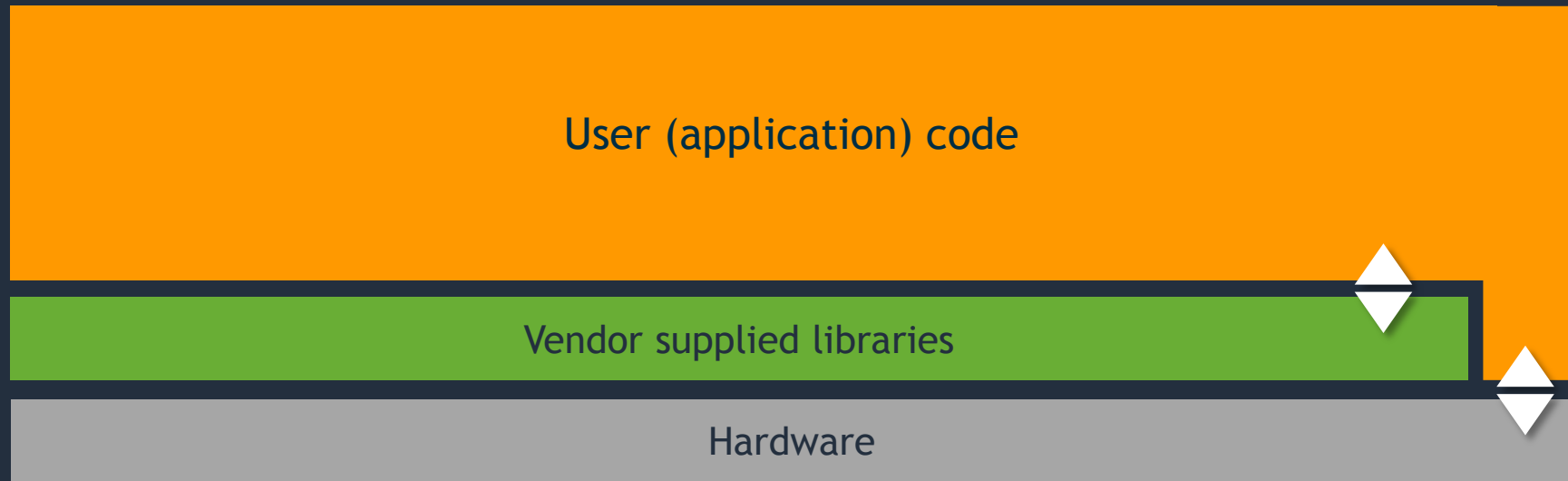


IoT data analytics and intelligence

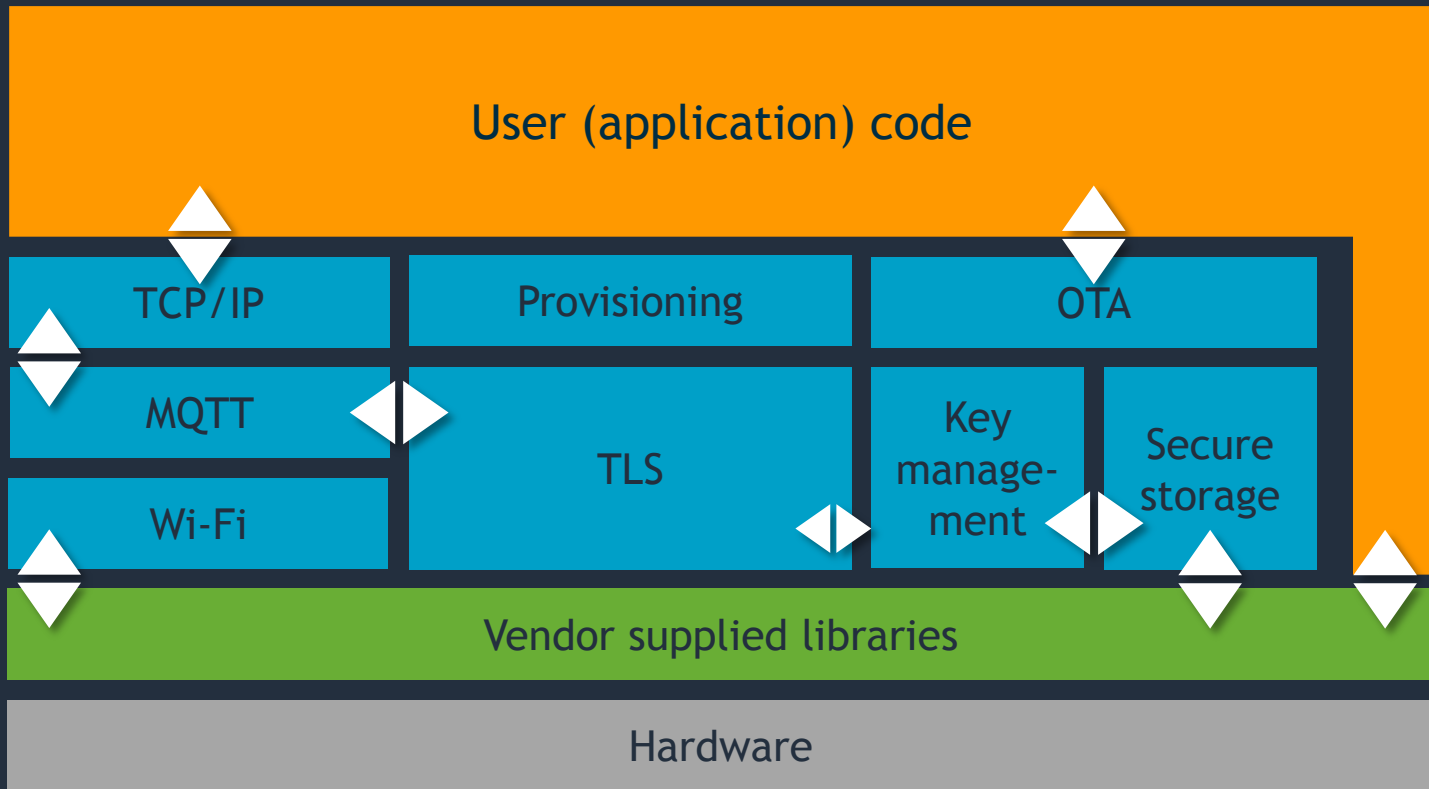


Intelligence
Insights & Logic → Action

Bare metal, assembly or C



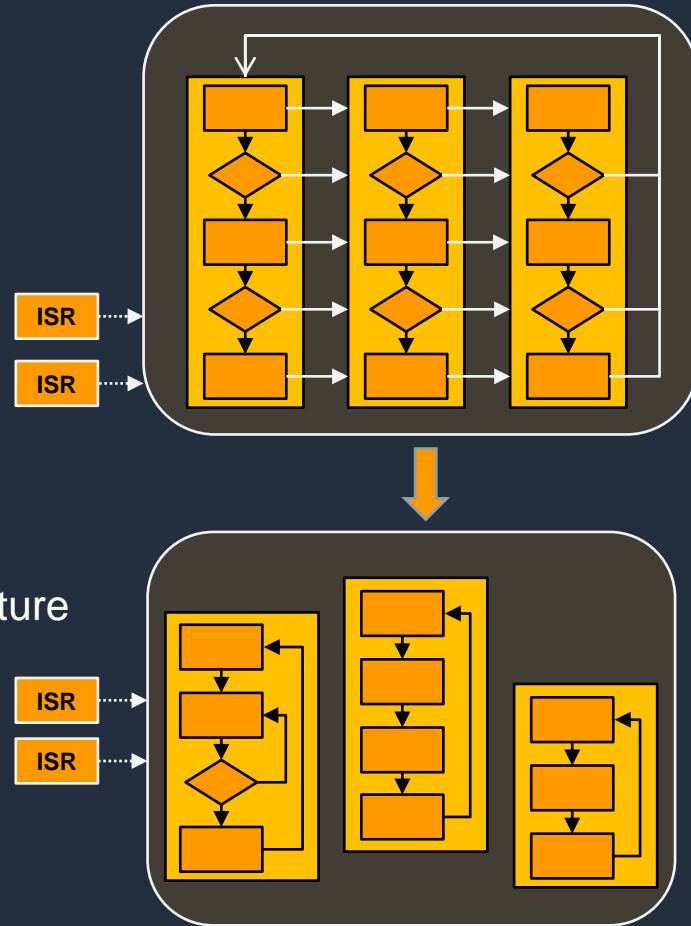
Application view with individual libraries



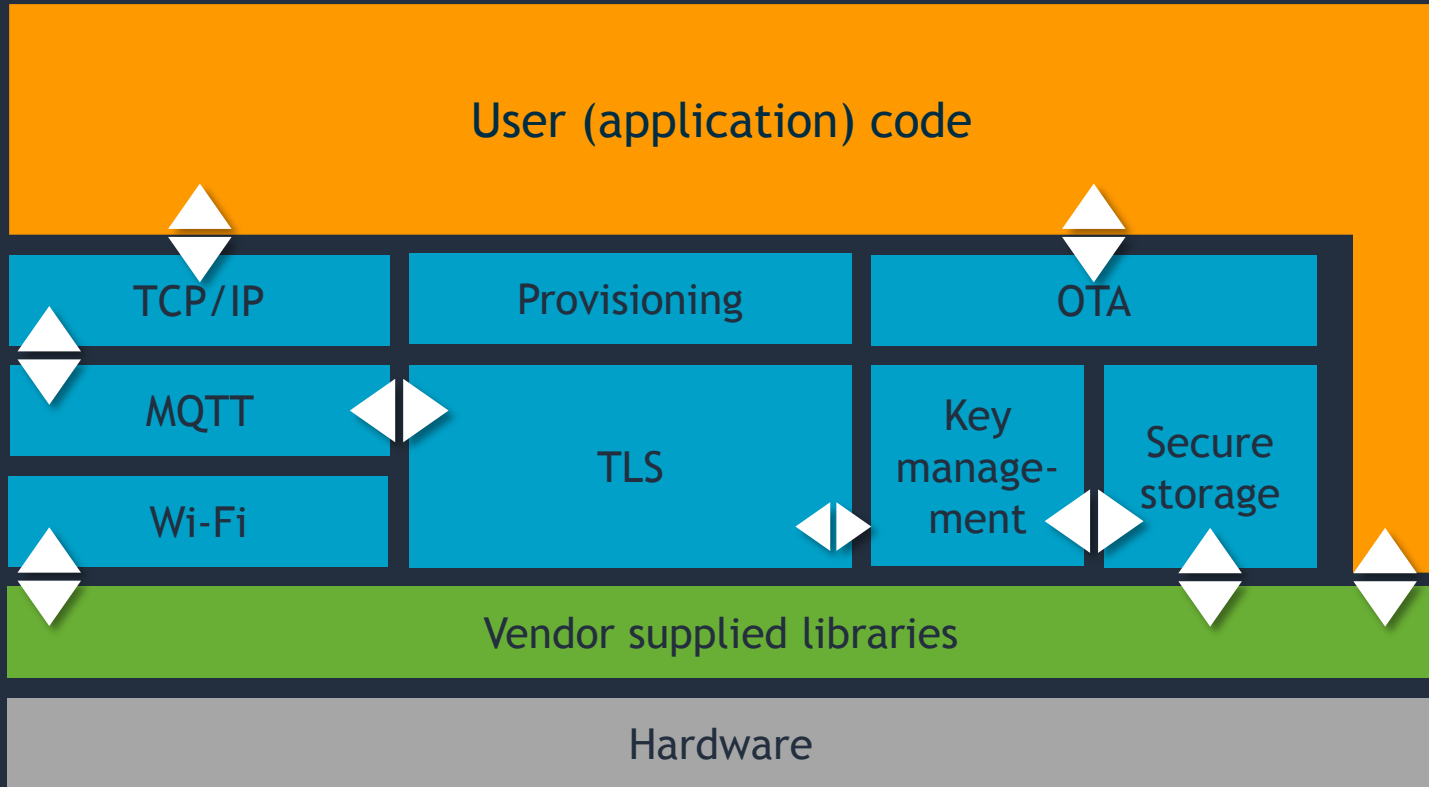
Introducing a library that implements multithreading

Application Design Goals:

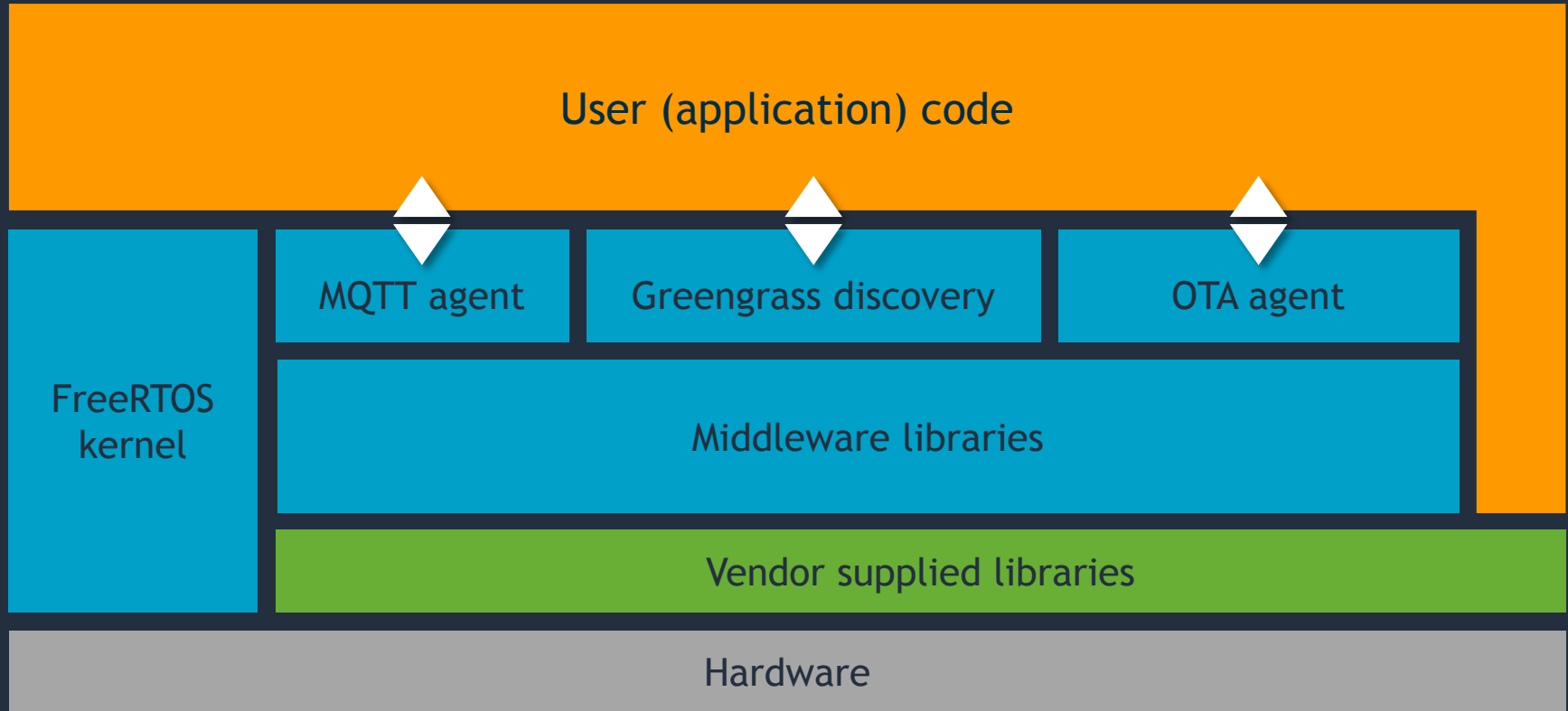
- Meet real time requirements!
- Maximize responsiveness
- Use as little CPU/Power as possible
- Maximize maintainability
- Maximize portability (hardware change)
- Simplicity!
- Fast to market
- Meet requirements with minimum expenditure



Application view with individual libraries



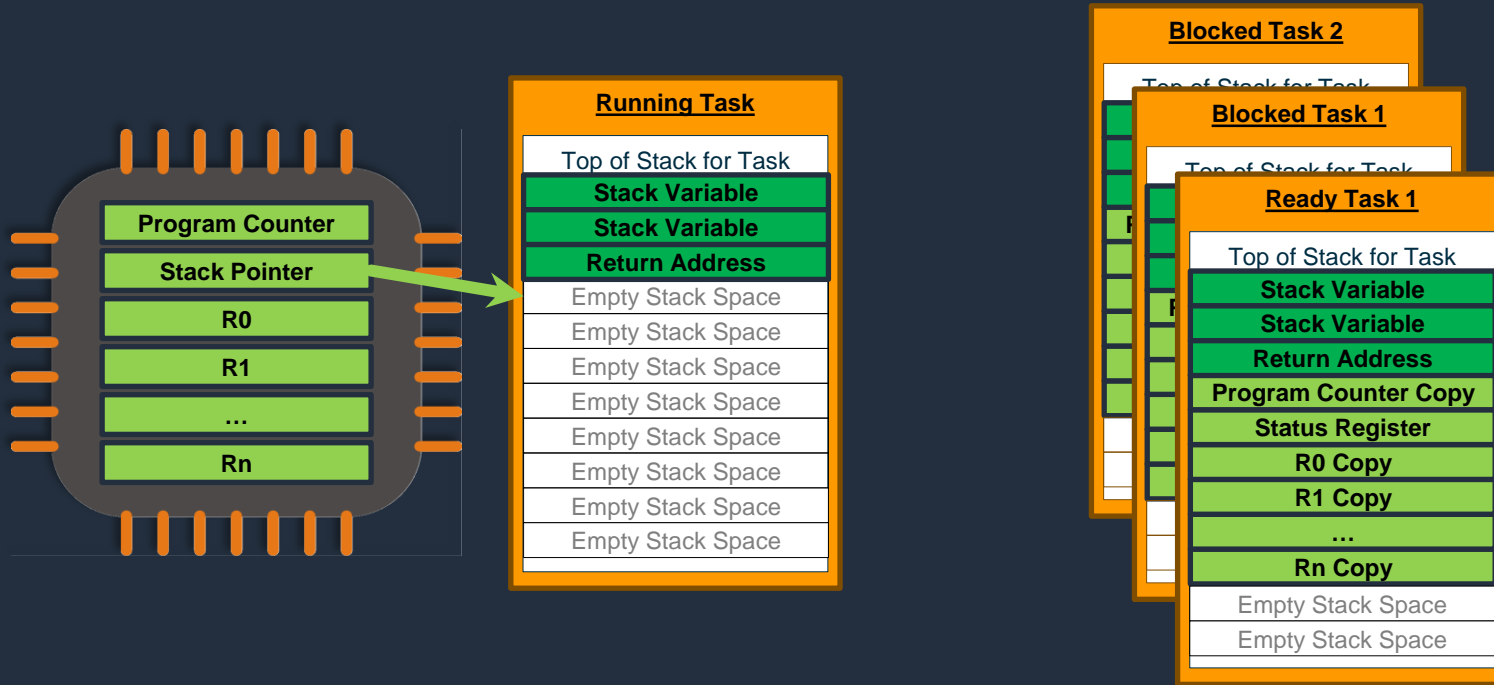
Application view with security and connectivity libraries



FreeRTOS on RISC-V: Source Files



Common source files and port specific source files

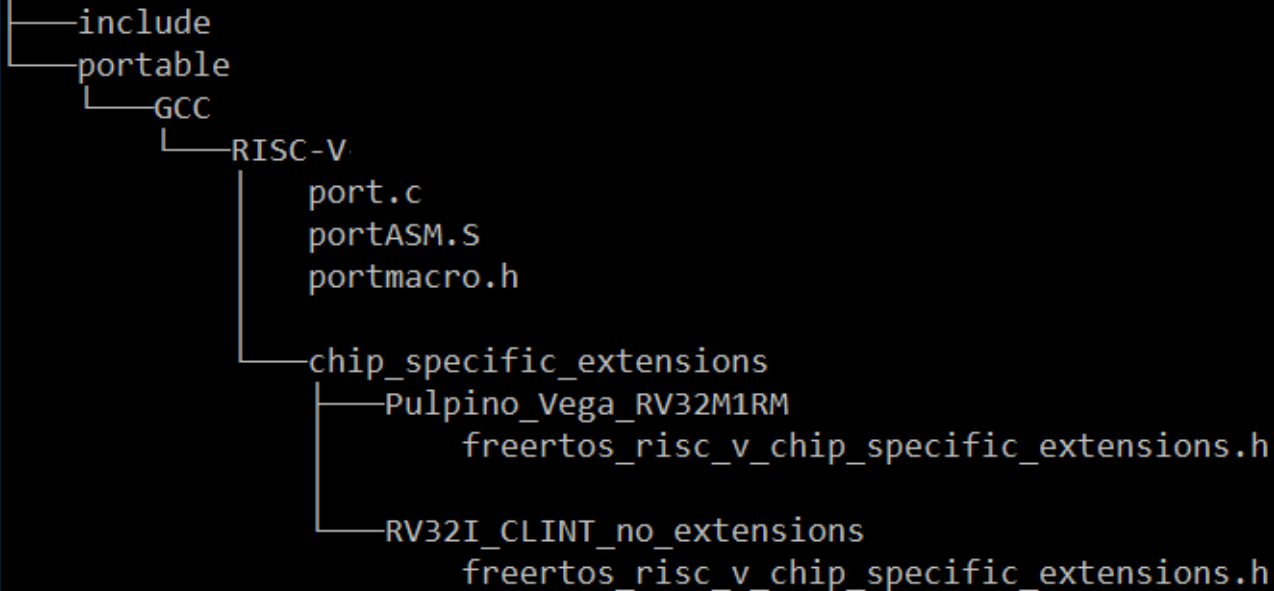


Example for Arm

```
Command Prompt
C:.\
  event_groups.c
  list.c
  queue.c
  stream_buffer.c
  tasks.c
  timers.c
  include
  portable
    GCC
      ARM_CM0
        port.c
        portmacro.h
      ARM_CM3
        port.c
        portmacro.h
      ARM_CM3_MPU
        port.c
        portmacro.h
```


Additional header file for RISC-V

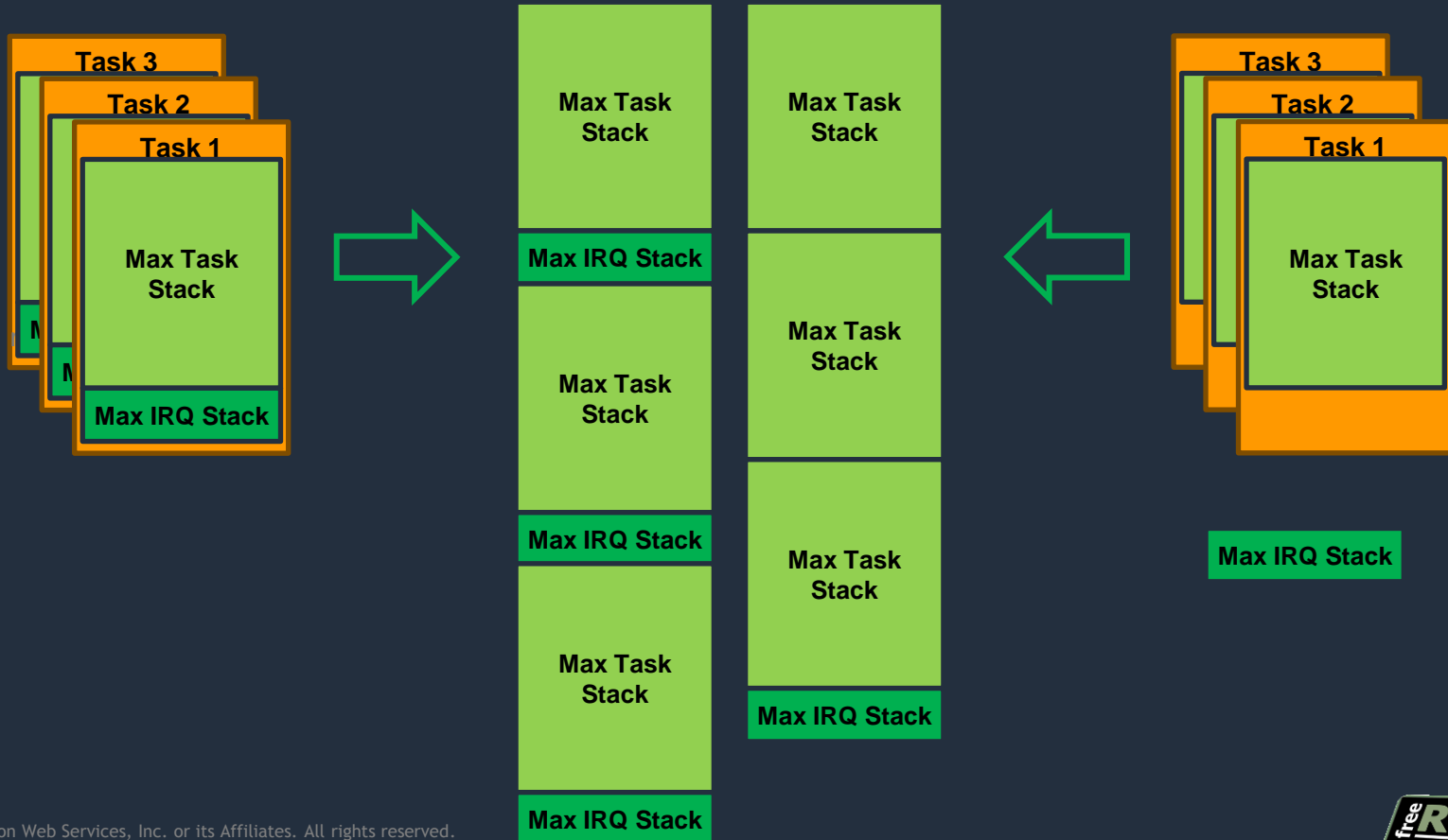
```
event_groups.c  
list.c  
queue.c  
stream_buffer.c  
tasks.c  
timers.c
```



FreeRTOS on RISC-V: Interrupt stack



Defining the interrupt stack



FreeRTOS on RISC-V: Setting CLINT base address



FreeRTOS on RISC-V: Installing the FreeRTOS trap handler



FreeRTOS on RISC-V: Calling an external interrupt handler



type filter text

- > Resource Builders
- ▼ C/C++ Build
 - Build Variables
 - Environment
 - Logging
 - Settings
 - Tool Chain Edit
- > C/C++ General
- CodeMR
- Linux Tools Path
- > MCU
- Project Natures

Settings

- Tool Settings
- Toolchains
- Devices
- Container Settings
- Build Steps

- Target Processor
- Optimization
- Warnings
- Debugging
- ▼ GNU RISC-V Cross Assembler
 - Preprocessor
 - Includes
 - Warnings
 - Miscellaneous
- ▼ GNU RISC-V Cross C Compiler
 - Preprocessor

Use preprocessor
 Do not search system directories (-nostdinc)
 Preprocess only (-E)

Defined symbols (-D)

DEBUG

portasmHANDLE_INTERRUPT=SystemIrqHandler



Apply and Close

Cancel



Factory Settings

Category:

General Options

Static Analysis

C/C++ Compiler

Assembler

Output Converter

Custom Build

Build Actions

Linker

Debugger

I-jet

Simulator

Language

Output

List

Preprocessor

Diagnostics

Extra Options

Ignore standard include directories

Additional include directories: (one per line)

\$PROJ_DIR\$\\..\\Source\\portable\\IAR\\RISC-V\\chip_specific_extensio

Preinclude file:

Defined symbols: (one per line)

RRUPT=vApplicationHandleTrap

Preprocessor output to file

Preserve comments

Thank You!

Download, share and support

<https://www.freertos.org>

<https://github.com/aws/amazon-freertos>

@real_FreeRTOS | @AWSOpen

