



FreeRTOS

Open source and the IoT

Richard Barry
Founder, FreeRTOS Project
Principal Engineer, AWS IoT



Agenda

The MIT Licensed FreeRTOS Kernel

Traditional Use Cases

Internet of Thing (IoT) Use Cases

Challenges of Developing Universal Software

Open Source Everything



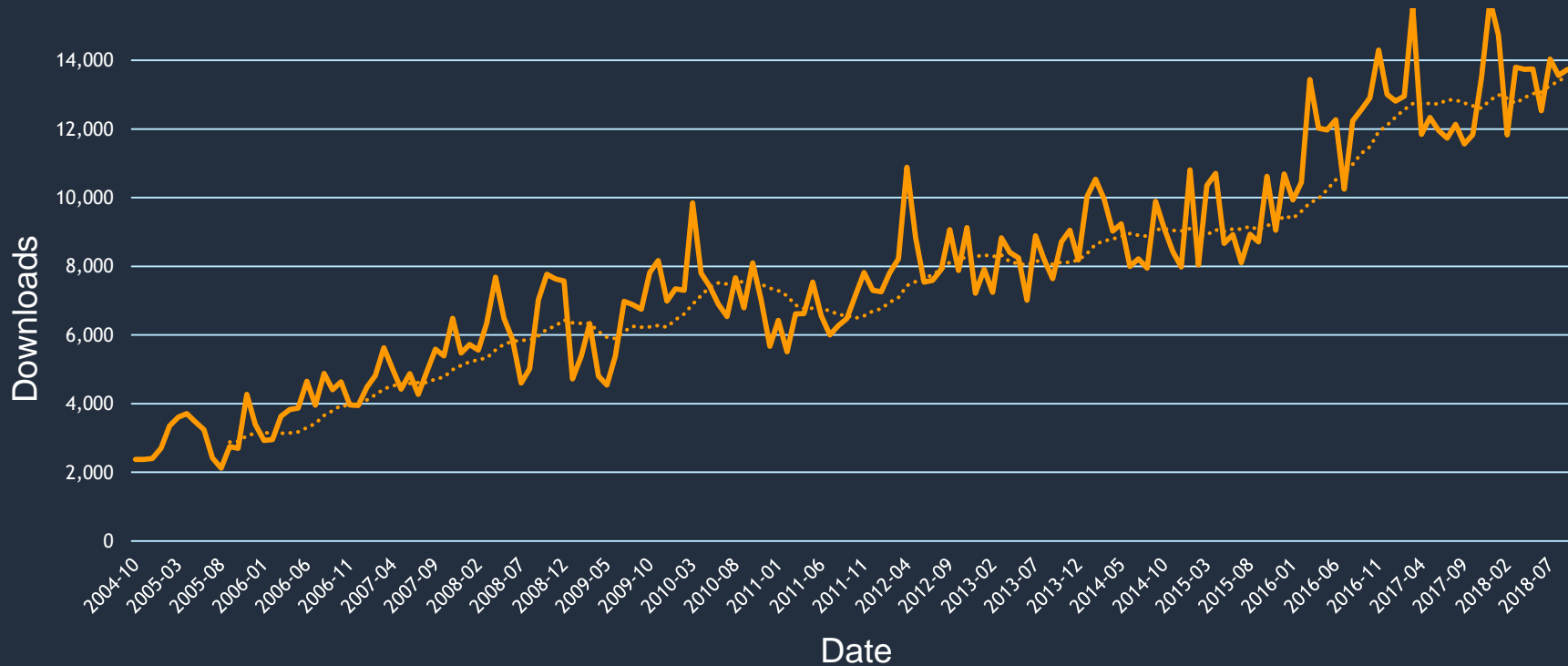
FreeRTOS—Open source real time kernel



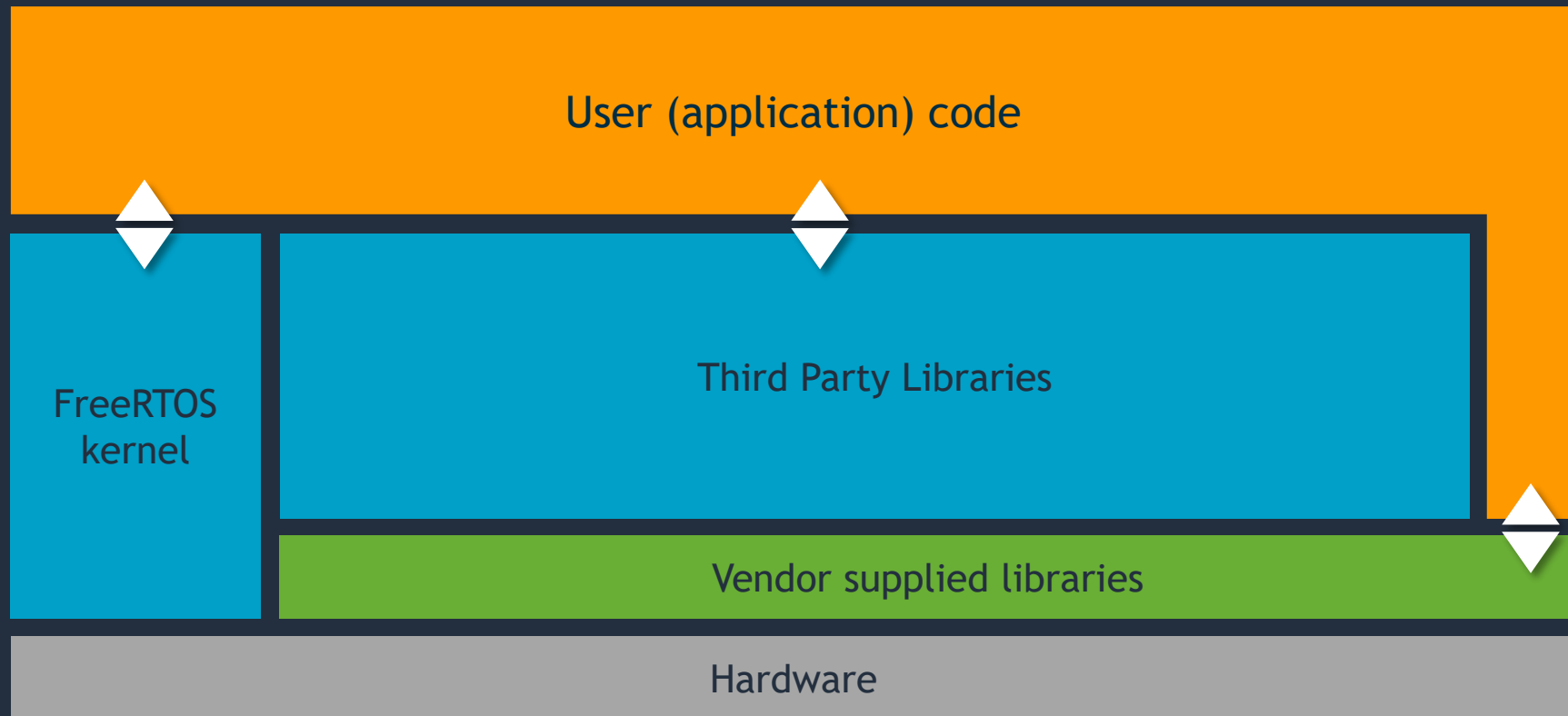
is everywhere...



FreeRTOS downloads per month over 15 years



FreeRTOS Kernel



Use cases



IoT

Cloud Storage & Compute



Fleet onboarding,
management and
SW updates



Fleet
audit and
protection



IoT data
analytics and
intelligence



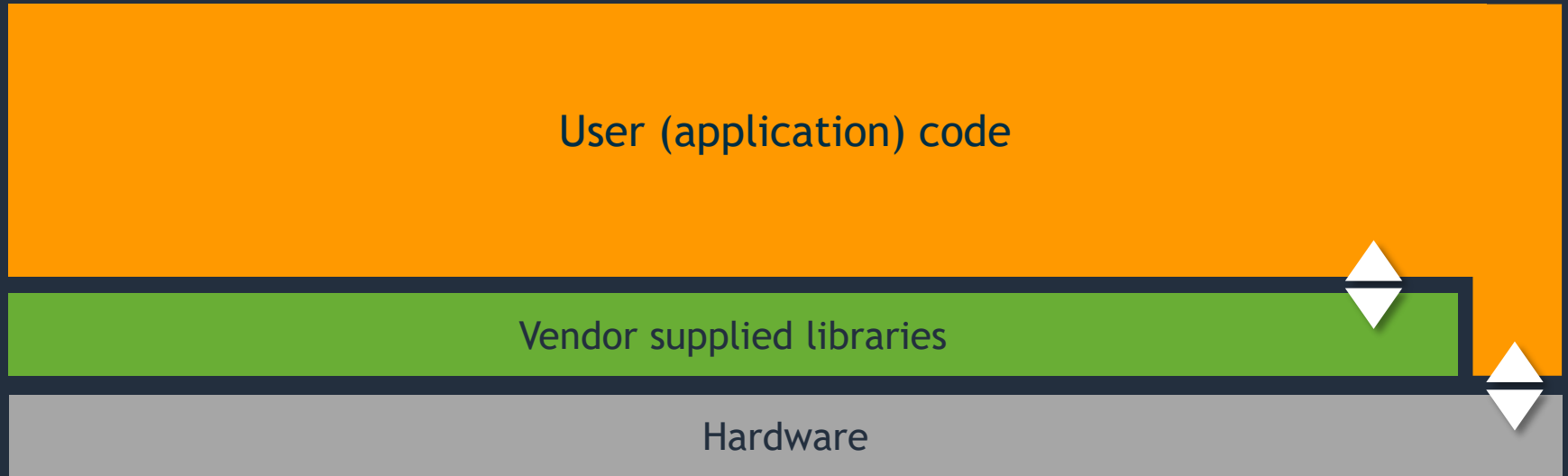
Intelligence
Insights & Logic →
Action

Not internet connected



Functionality

FreeRTOS Kernel



Not internet connected



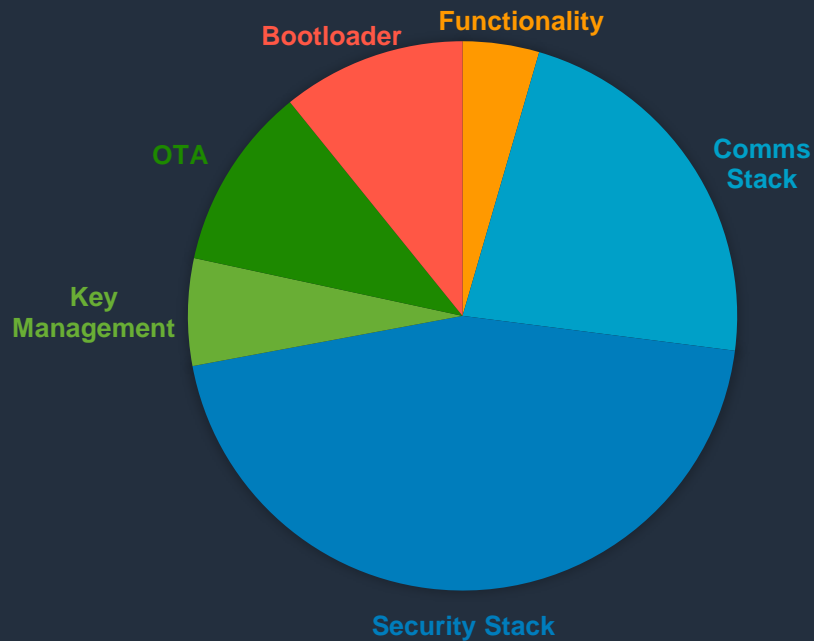
Functionality

Internet connected

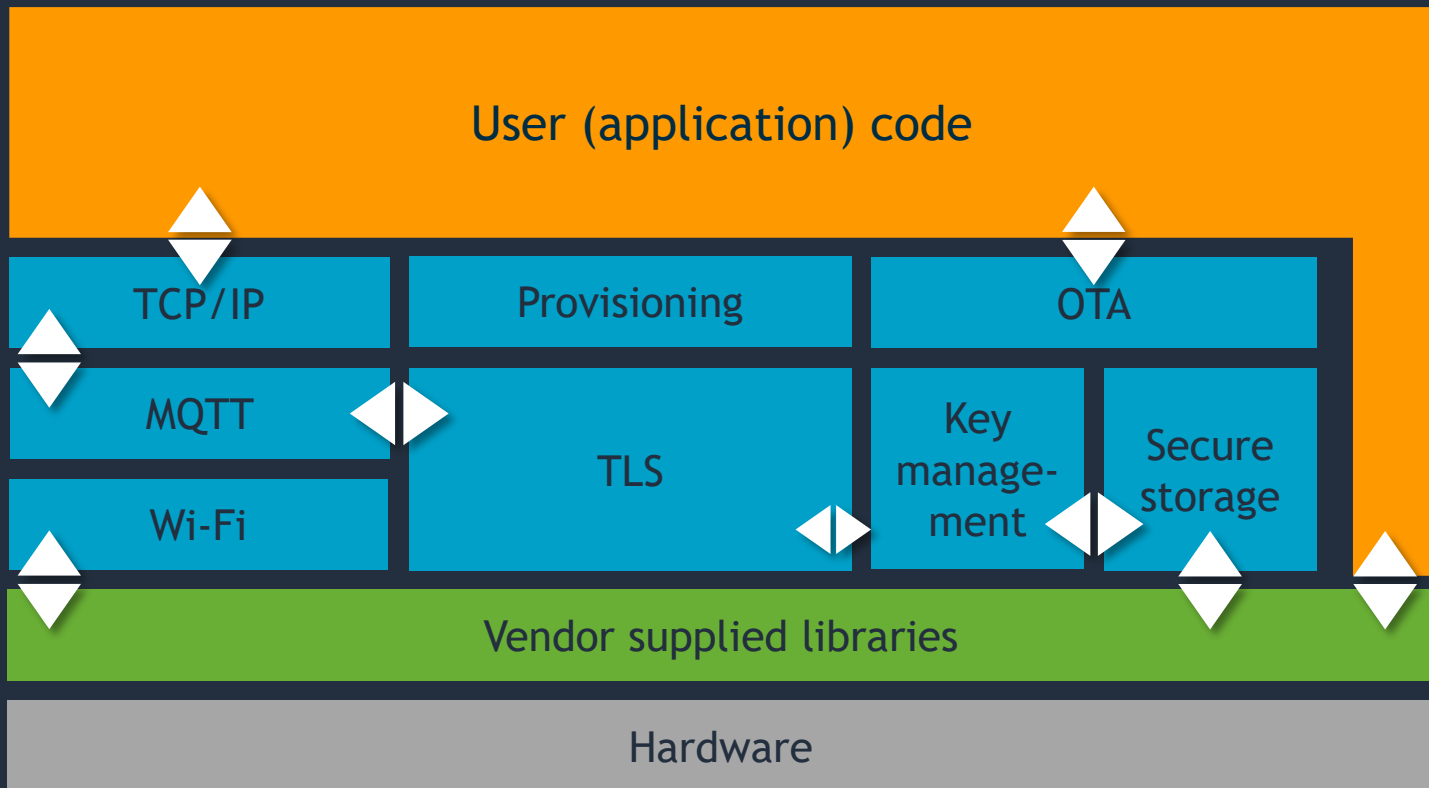


Bootloader
Over the Air Update (OTA)
Secure Storage
Security Stack
Communications Stack
Functionality

CPU cycles



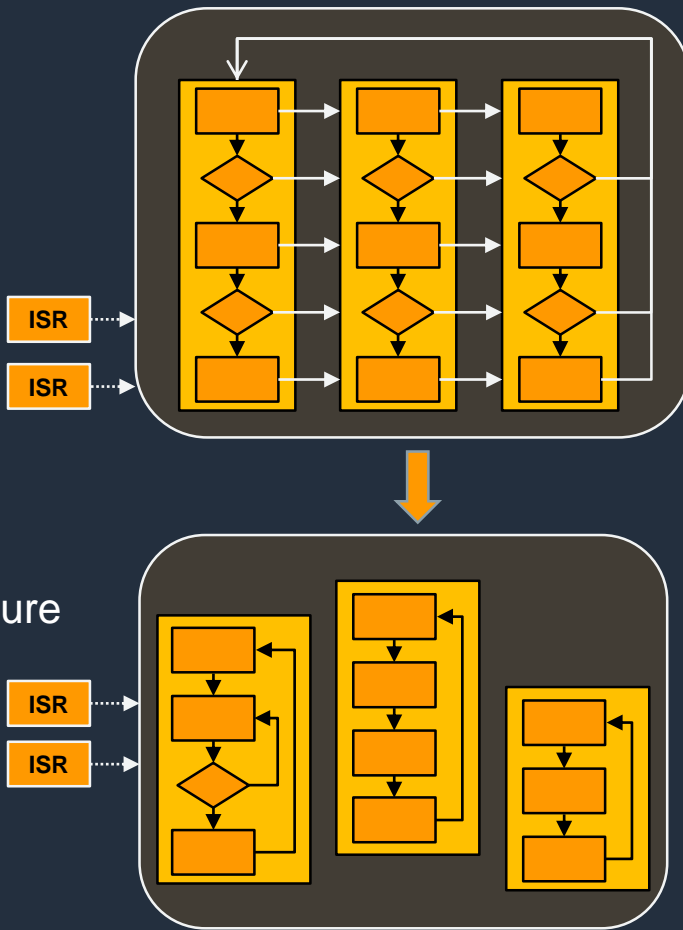
Application view with individual libraries



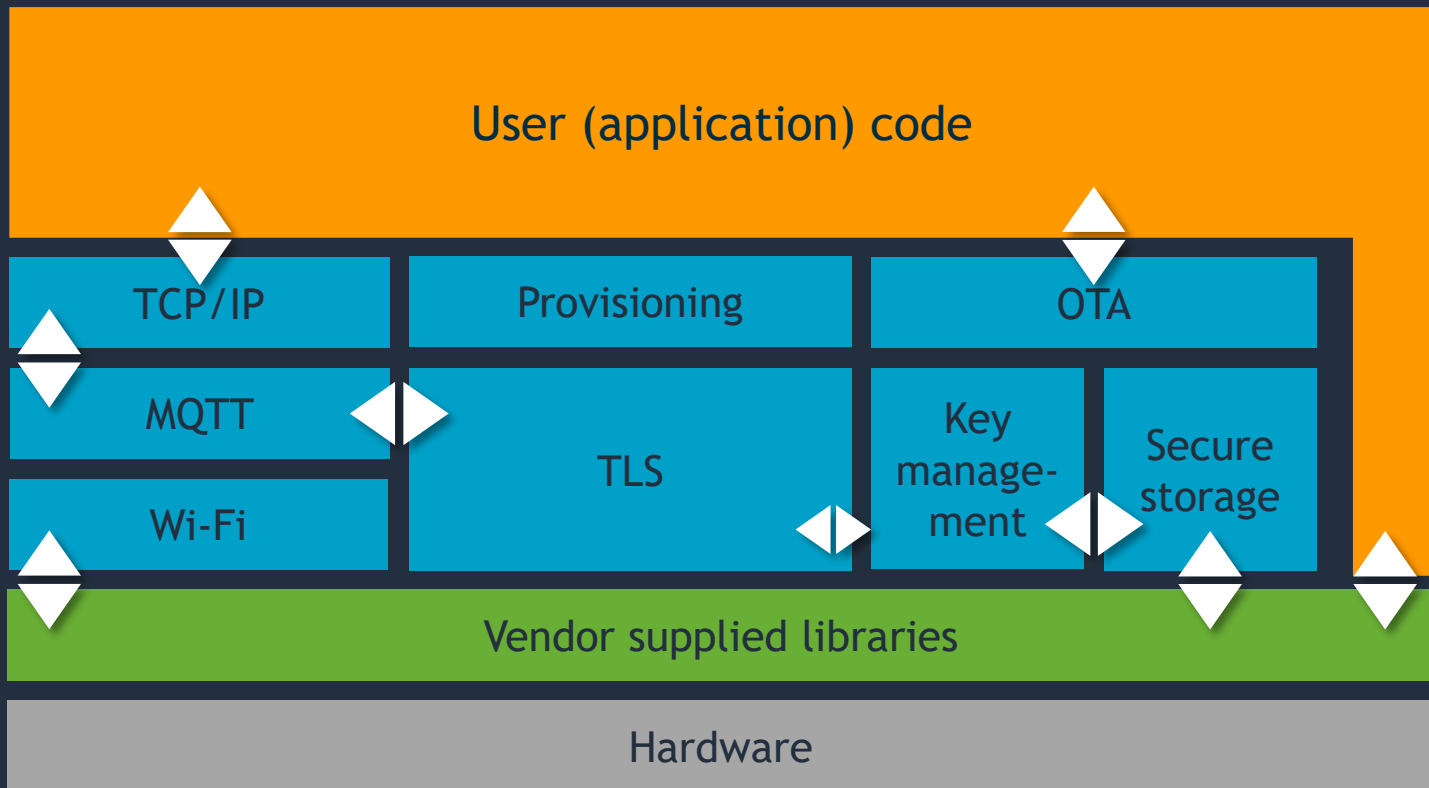
Introducing a library that implements multithreading

Application Design Goals:

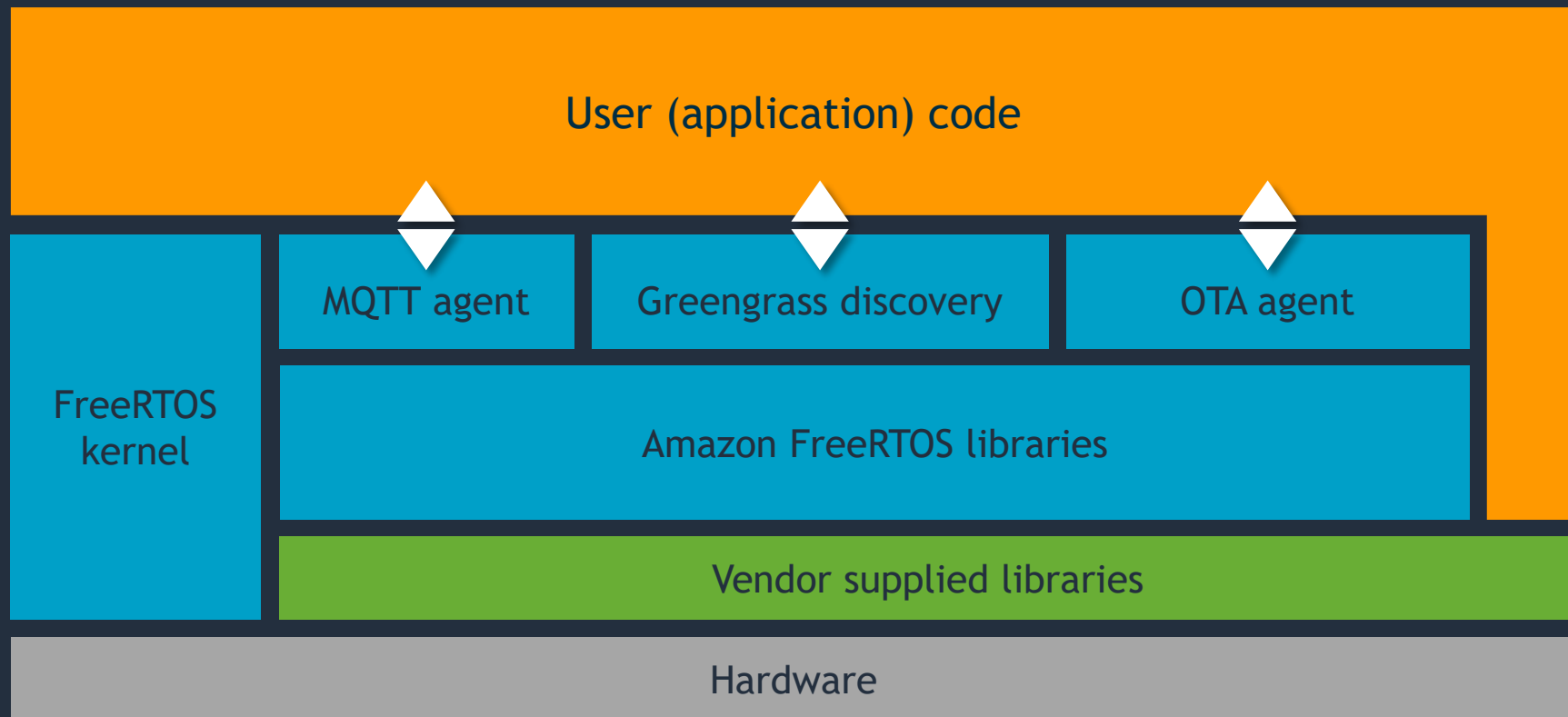
- Meet real time requirements!
- Maximize responsiveness
- Use as little CPU/Power as possible
- Maximize maintainability
- Maximize portability (hardware change)
- Simplicity!
- Fast to market
- Meet requirements with minimum expenditure



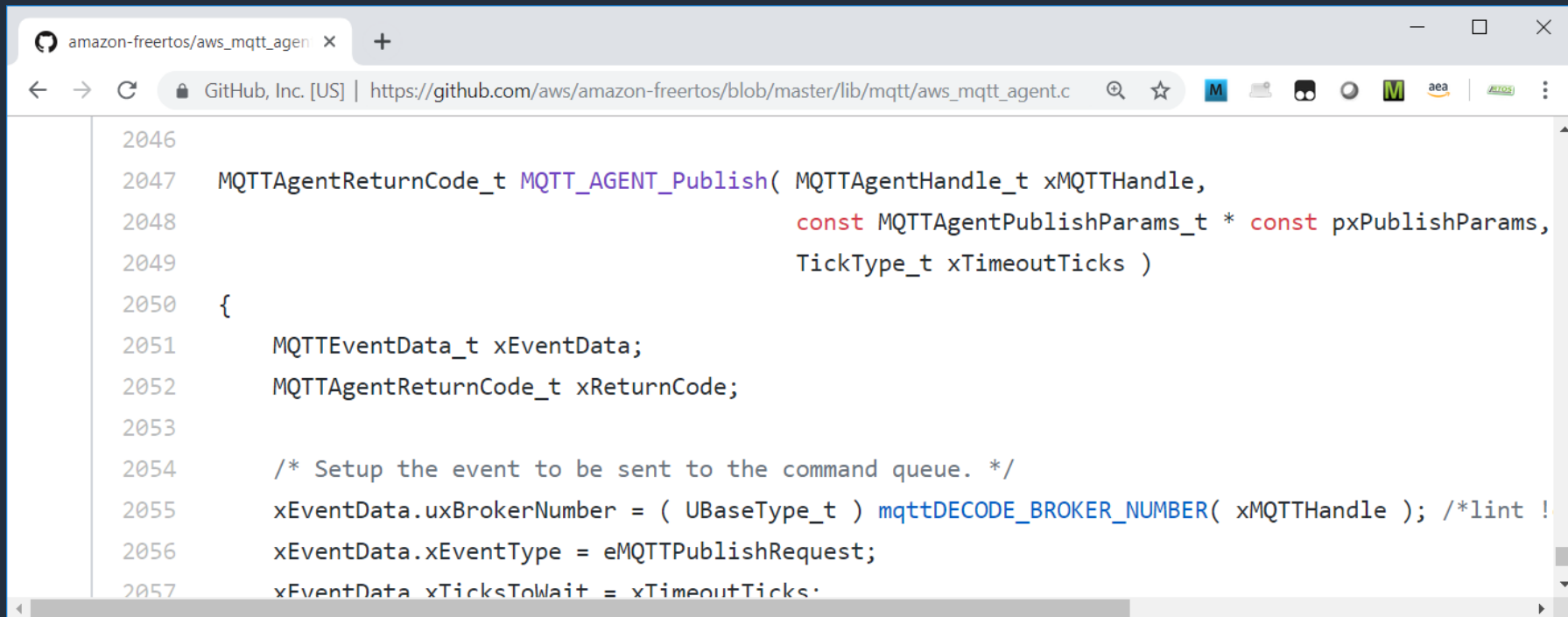
Application view with individual libraries



Application view with Amazon libraries

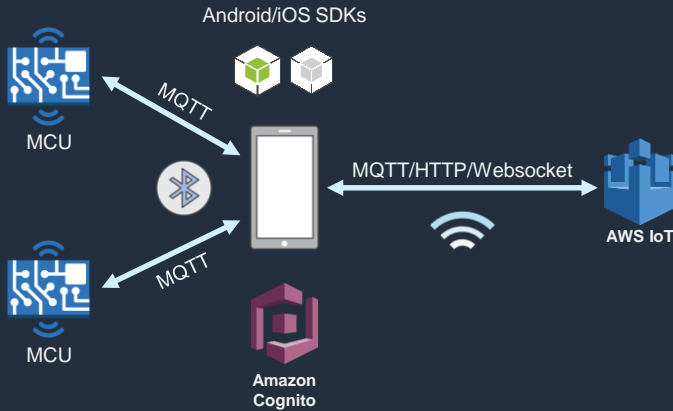


Simplified user experience

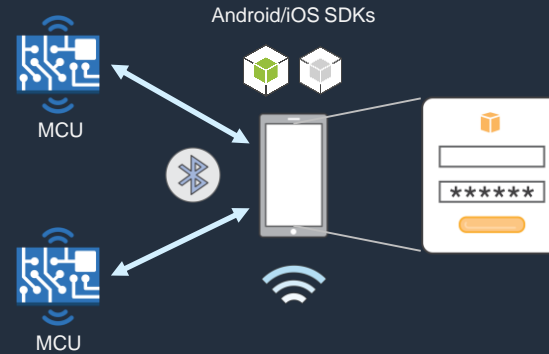


```
2046
2047 MQTTAgentReturnCode_t MQTT_AGENT_Publish( MQTTAgentHandle_t xMQTTHandle,
2048                                           const MQTTAgentPublishParams_t * const pxPublishParams,
2049                                           TickType_t xTimeoutTicks )
2050 {
2051     MQTTEventData_t xEventData;
2052     MQTTAgentReturnCode_t xReturnCode;
2053
2054     /* Setup the event to be sent to the command queue. */
2055     xEventData.uxBrokerNumber = ( UBaseType_t ) mqtttDECODE_BROKER_NUMBER( xMQTTHandle ); /*lint !
2056     xEventData.xEventType = eMQTTPublishRequest;
2057     xEventData.xTicksToWait = xTimeoutTicks;
```

Evolving use cases, wider applicability

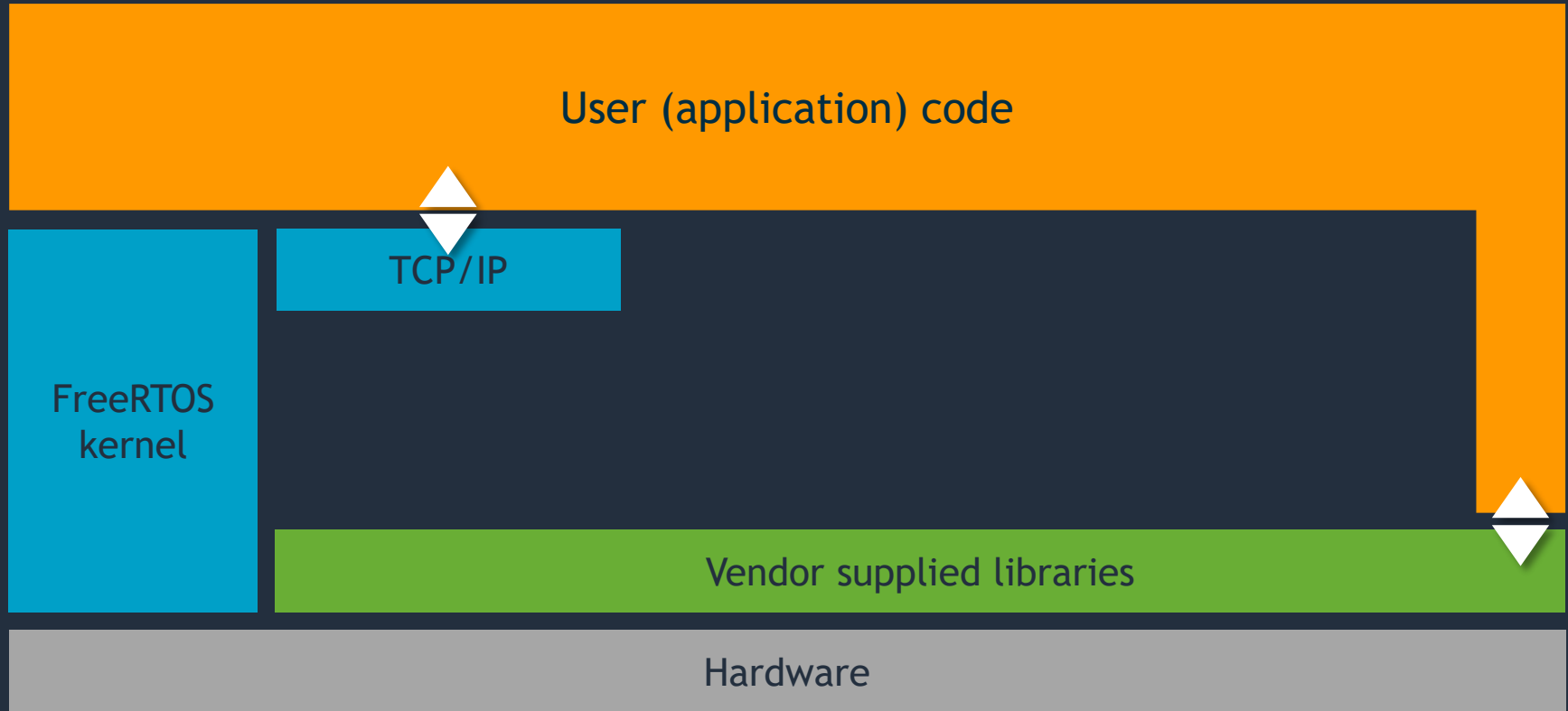


MQTT over Bluetooth
Low Energy

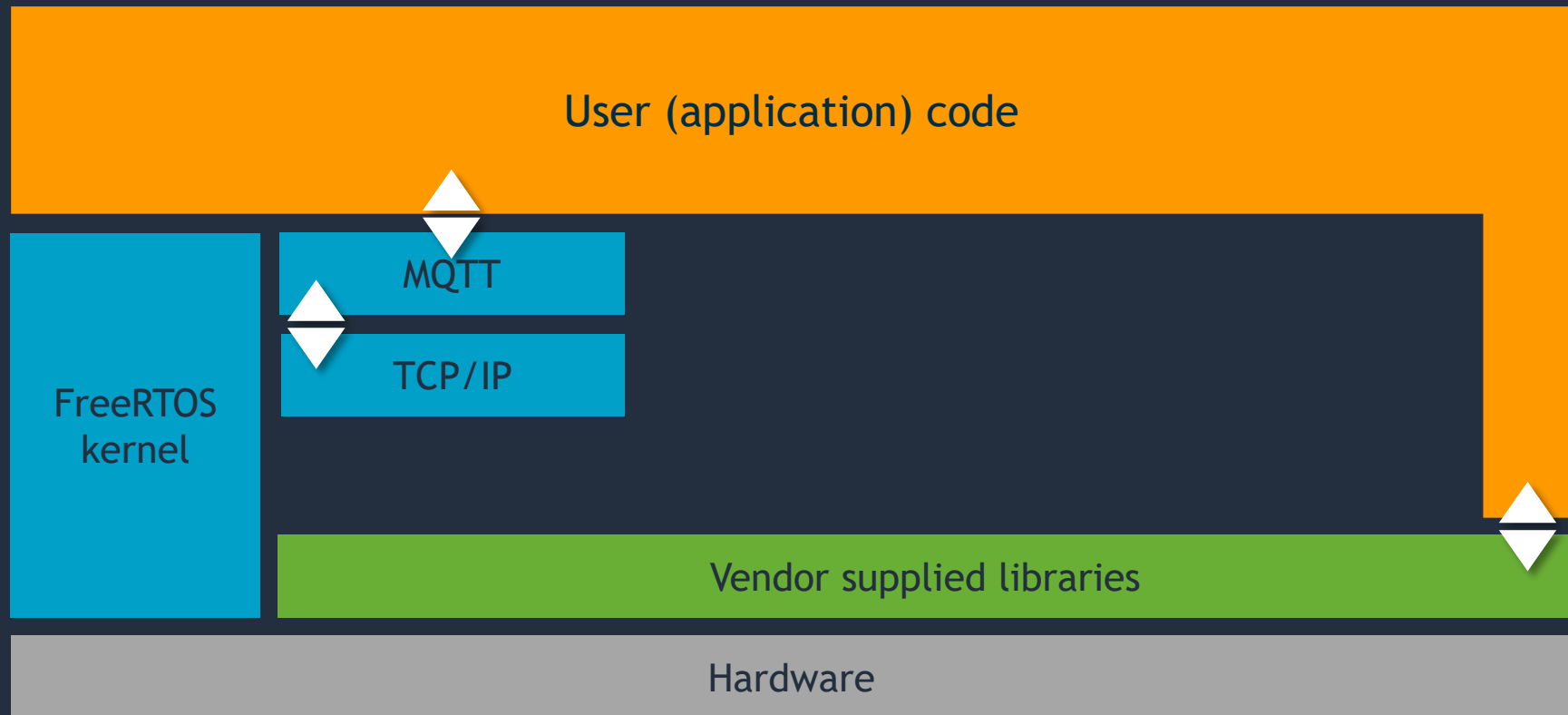


Wi-Fi provisioning over
Bluetooth Low Energy

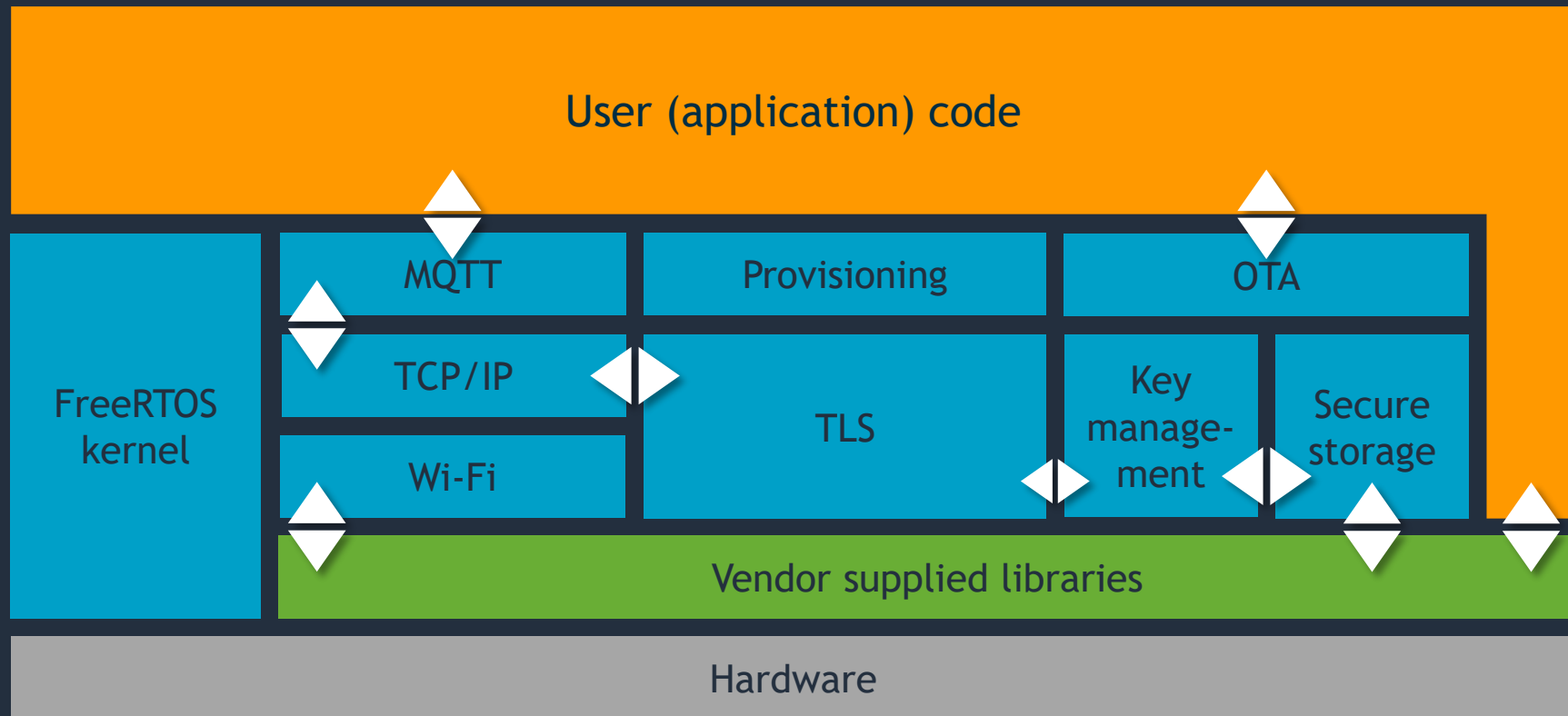
Use cases – common starting point



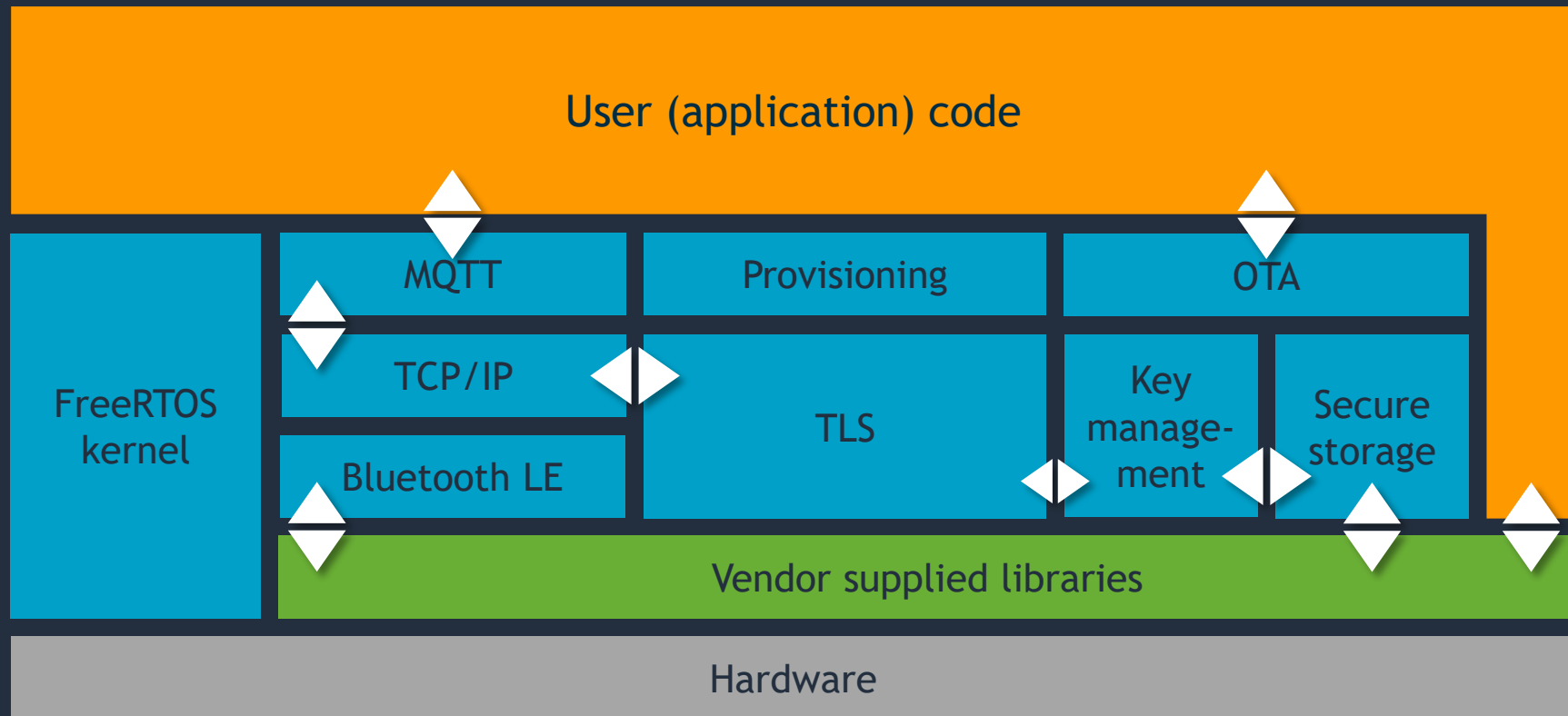
Use cases – basic (often TCP/IP is offloaded)



Use cases – all in Wi-Fi

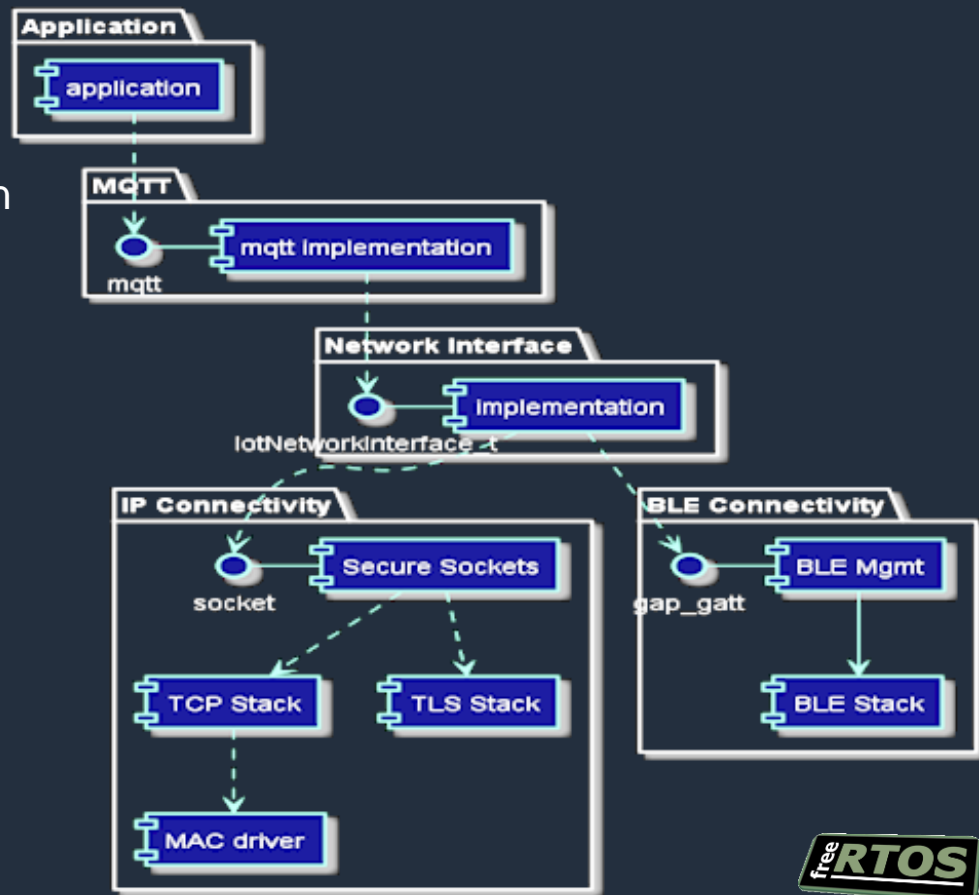


Use cases – all in Bluetooth Low Energy



Evolving software architecture

- Decoupled, stand alone libraries
- Transport agnostic
- Multiple transports in a single application
- Working to making porting layers thinner
- Add to pre-existing projects
- Add to auto-generated projects



Flexible user experience



```
1495 IotMqttError_t IotMqtt_Publish( IotMqttConnection_t mqttConnection,  
1496                               const IotMqttPublishInfo_t * pPublishInfo,  
1497                               uint32_t flags,  
1498                               const IotMqttCallbackInfo_t * pCallbackInfo,  
1499                               IotMqttOperation_t * pPublishOperation )  
1500 {  
1501     IOT_FUNCTION_ENTRY( IotMqttError_t, IOT_MQTT_SUCCESS );  
1502     _mqttOperation_t * pOperation = NULL;
```

Abstracted flexible user experience

```
amazon-freertos/aws_mqtt_agent x +
MQTTAgentReturnCode_t MQTT_AGENT_Publish( MQTTAgentHandle_t xMQTTHandle,
                                           const MQTTAgentPublishParams_t * const pxPublishParams,
                                           TickType_t xTimeoutTicks )
{
    IotMqttError_t xMqttStatus = IOT_MQTT_STATUS_PENDING;
    MQTTConnection_t * pxConnection = ( MQTTConnection_t * ) xMQTTHandle;
    IotMqttPublishInfo_t xPublishInfo = IOT_MQTT_PUBLISH_INFO_INITIALIZER;

    /* Set the members of the publish info. */
    xPublishInfo.pTopicName = ( const char * ) pxPublishParams->pucTopic;
    xPublishInfo.topicNameLength = pxPublishParams->usTopicLength;
    xPublishInfo.qos = ( IotMqttQos_t ) pxPublishParams->xQoS;
    xPublishInfo.pPayload = ( const void * ) pxPublishParams->pvData;
    xPublishInfo.payloadLength = pxPublishParams->ulDataLength;

    /* Call the MQTT v4 blocking PUBLISH function. */
    xMqttStatus = IotMqtt_TimedPublish( pxConnection->xMQTTConnection,
```

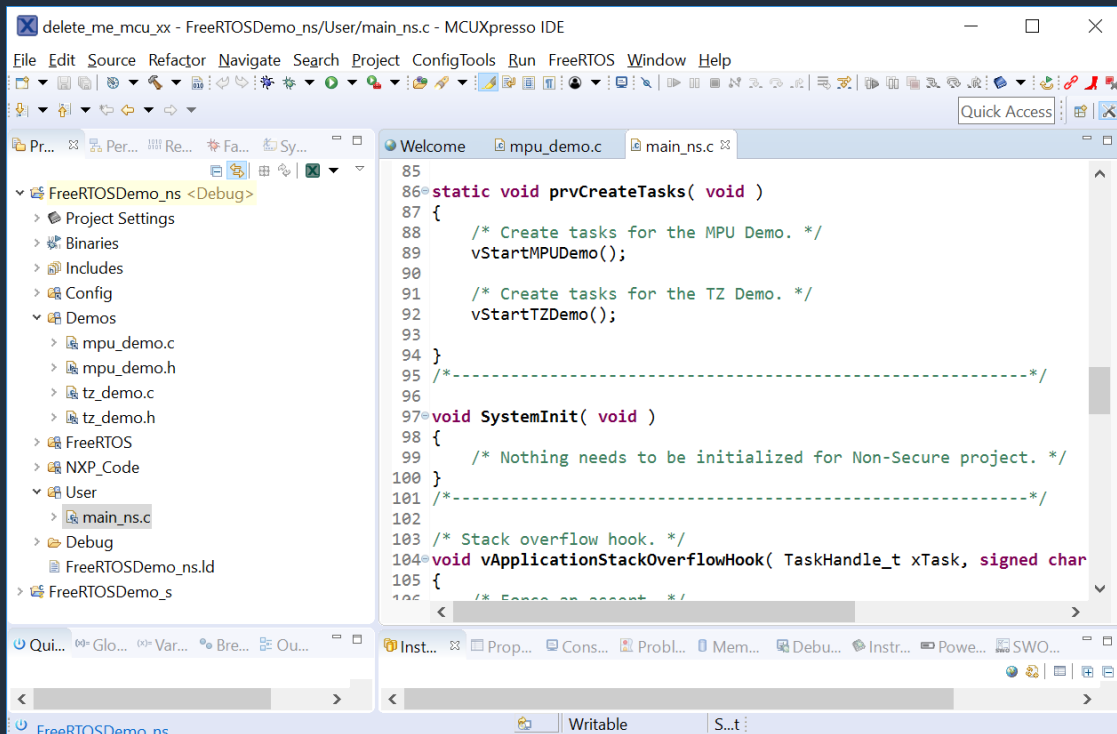
Open source kernel

- FreeRTOS
- Security & connectivity

Pre-configured
examples for
approximately 40
architectures and
20 tool chains -
not all open
source.

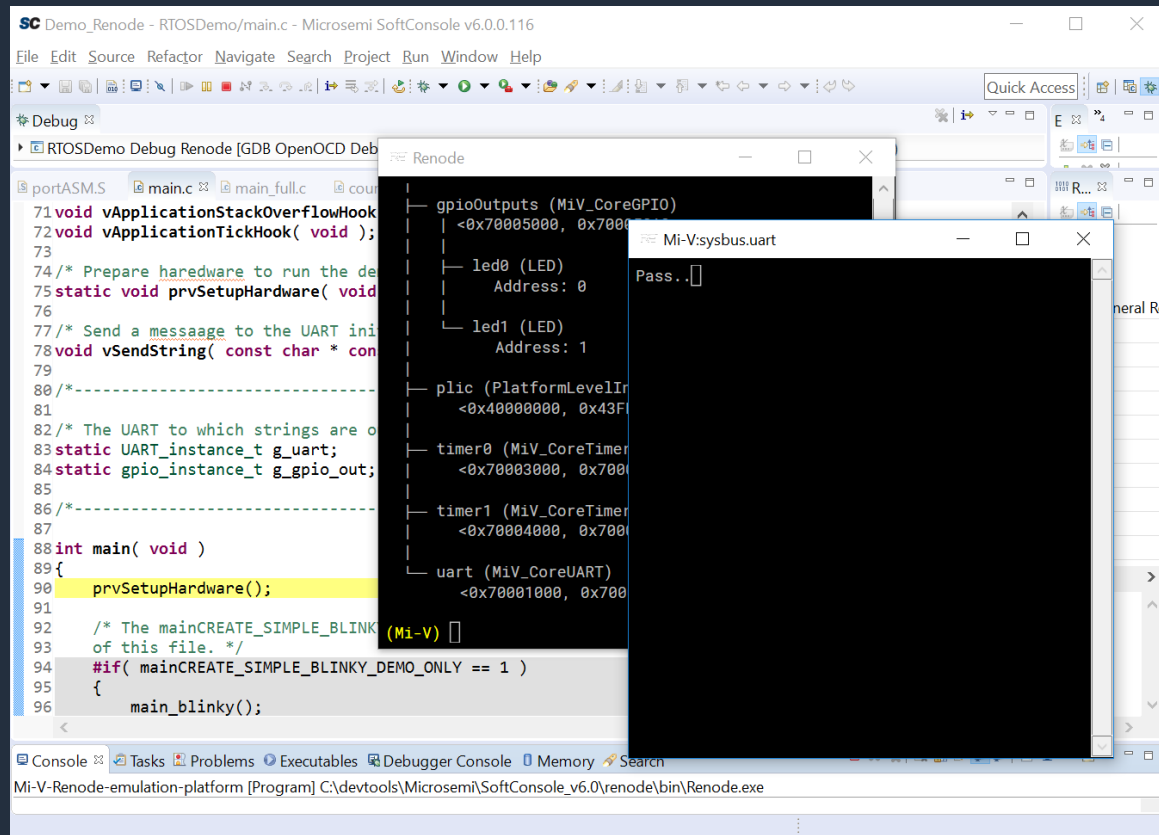
Open source kernel, tools

- FreeRTOS
- Security & connectivity
- Eclipse based
- GCC
- LLVM



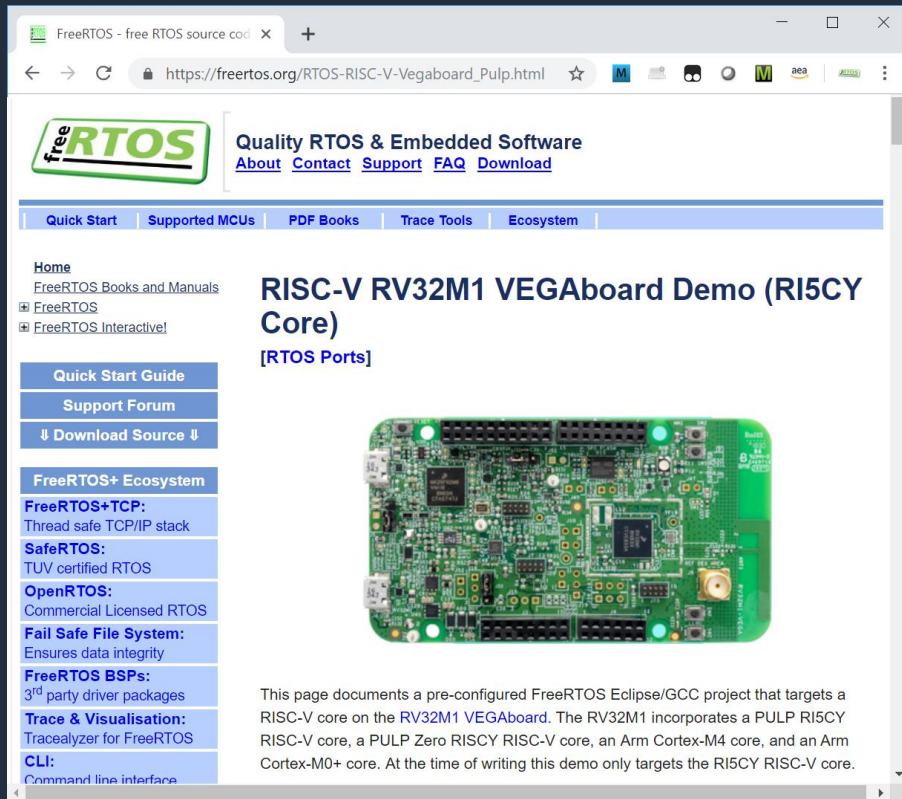
Open source kernel, tools, emulators

- FreeRTOS
- Security & connectivity
- Eclipse based
- GCC
- LLVM
- Renode
- QEMU



Open source kernel, tools, emulators, ISA & hardware

- FreeRTOS
- Security & connectivity
- Eclipse based
- GCC
- LLVM
- Renode
- QEMU
- RISC-V
- PULP RI5KY



The screenshot shows a web browser displaying the FreeRTOS website. The page title is "RISC-V RV32M1 VEGAboard Demo (RI5CY Core)" with a subtitle "[RTOS Ports]". The page features a large image of the VEGAboard hardware. The left sidebar contains navigation links such as "Quick Start Guide", "Support Forum", "Download Source", and "FreeRTOS+ Ecosystem". The main content area includes a description of the demo project, mentioning the RV32M1 VEGAboard, the RI5CY RISC-V core, and the PULP Zero RISC-V core.

FreeRTOS - free RTOS source code

https://freertos.org/RTOS-RISC-V-VegaBoard_Pulp.html

freeRTOS Quality RTOS & Embedded Software
[About](#) [Contact](#) [Support](#) [FAQ](#) [Download](#)

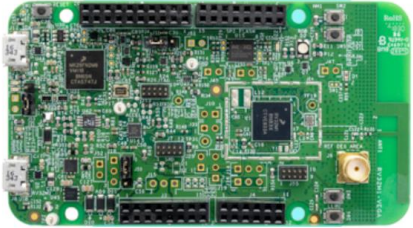
[Quick Start](#) [Supported MCUs](#) [PDF Books](#) [Trace Tools](#) [Ecosystem](#)

Home
[FreeRTOS Books and Manuals](#)
[FreeRTOS](#)
[FreeRTOS Interactive!](#)

[Quick Start Guide](#)
[Support Forum](#)
[Download Source](#)

FreeRTOS+ Ecosystem
FreeRTOS+TCP:
Thread safe TCP/IP stack
SafeRTOS:
TUV certified RTOS
OpenRTOS:
Commercial Licensed RTOS
Fail Safe File System:
Ensures data integrity
FreeRTOS BSPs:
3rd party driver packages
Trace & Visualisation:
Tracealyzer for FreeRTOS
CLI:
Command line interface

RISC-V RV32M1 VEGAboard Demo (RI5CY Core)
[RTOS Ports]



This page documents a pre-configured FreeRTOS Eclipse/GCC project that targets a RISC-V core on the [RV32M1 VEGAboard](#). The RV32M1 incorporates a PULP RI5CY RISC-V core, a PULP Zero RISC-V core, an Arm Cortex-M4 core, and an Arm Cortex-M0+ core. At the time of writing this demo only targets the RI5CY RISC-V core.



Thank You!

Download, share and support

opensource.amazon.com/enterprise-oss-book

@AWSOpen | opensource.amazon.com | aws.github.io

