White Paper

The Benefits of Using Arm® TrustZone® in Your Design

November 2020

Abstract

Security is a crucial and far-reaching topic for any electronics-based equipment manufacturer, from battery-powered wireless connected temperature sensors, to high-power industrial motor control applications. Ensuring an end-product is secure is not just about protecting adversaries gaining access to secret keys and passwords. It also encompasses protecting the application's software intellectual property from being copied, and the compromised device being used as a platform to take control and exploit other systems.

The nature of digital security involves using cryptographic keys and encryption algorithms to protect access to various system resources and capabilities. Such functions need processing resources to operate, and these critical security components must be isolated from other parts of an application that might be accessible from other systems.

Isolation is a crucial aspect of security but achieving it with today's functionally-rich and highly integrated processors is a challenge.

In this white paper, we introduce the Arm® TrustZone® system-wide approach to achieving security and how TrustZone is implemented on the Renesas Advanced (RA) Family of 32-bit microcontrollers. The Renesas RA Family is based on the 32-bit Arm® Cortex®-M microcontroller architecture and is available in four different series optimized between low power and high-performance operation.

The Need for Isolation

Today, there are very few embedded systems that do not connect to other devices and applications. Connectivity, either wired or wireless, is omnipresent, and internet-connected devices, such as an internet of things (IoT) devices are exceptionally vulnerable to attack. The whole premise of the IoT is based on having access to a local gateway or direct access to cloud resources. For example, an environment-sensing edge node located within a manufacturing plant routinely connects to an associated resource, local or cloud-based, to transfer temperature, humidity, and other environmental parameters for analysis. The interpretation of that data by a process control application might instigate an instruction being sent to the plant's heating and ventilation system edge node to turn on. In both scenarios, the edge node application will access confidential data and keys to authenticate and encrypt or decrypt the communication. For an application to process data in this way, with other application code running and peripheral interfaces available to any prying eyes, leaves the edge node open to attack from adversaries.

The application code accessing and using the security processes and keys needs to be completely isolated from other code sections not involved with secure data. Also, isolation at the lowest physical

layer is crucial, including during boot, since many exploits specifically target a device during the bootup process.

Arm TrustZone technology provides an efficient embedded mechanism to separate a single-core device's hardware domain into two distinct isolated environments. One is used for running any code and the other for running code that requires an entirely isolated, separate, and secure area. See Figure 1.

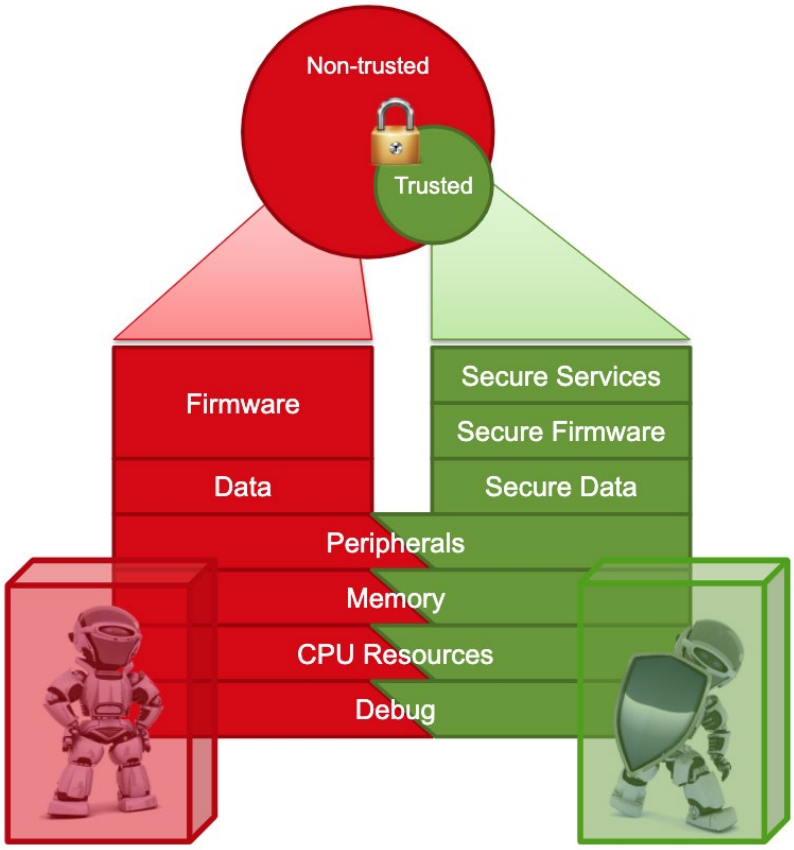


Figure 1: How Arm TrustZone creates hardware-enforced isolation of a microcontroller's features (source Renesas Electronics Corp.)

Introducing Arm TrustZone

Arm TrustZone has been available for over ten years and was first introduced as a security extension to the Cortex®-A class of application processors. On the Cortex-A microprocessors, the two distinct and isolated environments – termed a secure world or trusted environment and a non-secure world or non-trusted environment – would typically run separate operating system instances. Only in the recent past has TrustZone been made available for the Cortex-M33 series of microcontroller cores. Naturally, microcontrollers have some resource limitations compared to a microprocessor, so the TrustZone implementation for the M series MCUs is slightly different because of the lower overheads in performance and power efficiency. Not all microcontroller vendors have yet implemented TrustZone, and the ecosystem of software resources and tools are still being developed.

Arm TrustZone is an example of a trusted execution environment (TEE) that provides isolation of data, services, and specific areas of memory between the secure and non-secure environments. What runs within the secure area is flexible and can include private keys, firmware, and security routines; however, it could also contain a commercially sensitive library or algorithm representing significant IP and needs protection. TrustZone blocks non-secure software from accessing software and resources in the secure area. The secure area can access everything within its area. Managing access between the two worlds is the non-secure callable veneer. See Figure 2. The veneer provides a mechanism of defined access points for non-secure code to call services within the secure area.

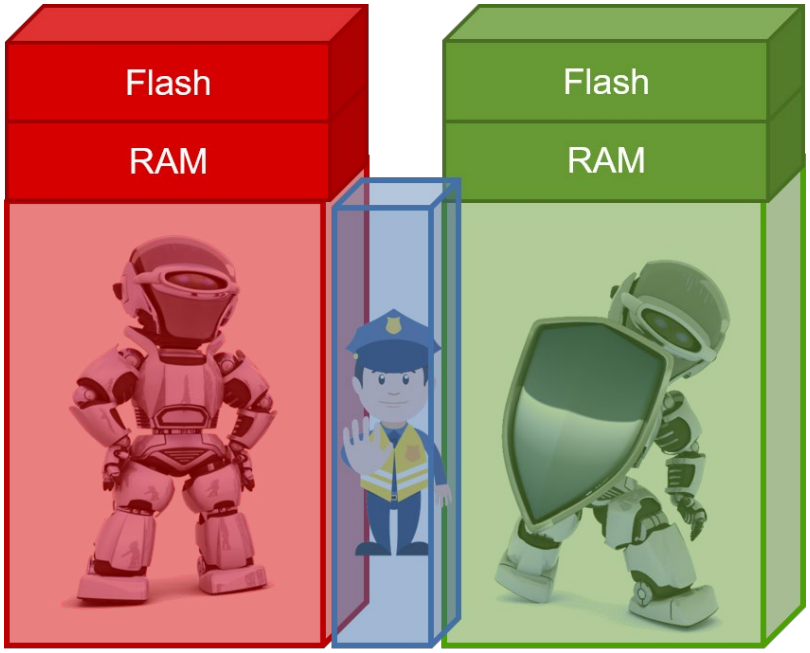


Figure 2: The non-secure callable veneers provide functions through which the non-secure world can access secure world services (source Renesas Electronics Corp.)

The separation that TrustZone provides significantly reduces the attack surfaces of critical components, as seen by an adversary, simplifying an embedded device's security assessment. Arm's approach to trust not only includes the encapsulation of software, but also extends to Flash memory and RAM; however, Arm's original definition of TrustZone does not cater to direct memory access (DMA) or a direct transfer controller (DTC) requests. Also, peripherals and pin access in Arm's TrustZone definition are not separated in the same way that application code and data are. Thus, exposed general-purpose IO (GPIO) and peripheral features leave a microcontroller vulnerable to a wide variety of exploits.

Arm TrustZone on the Renesas RA Family MCUs with Cortex-M33

The separation and isolation architecture of the Renesas RA Family microcontrollers with Cortex-M33 is built on Arm's v8-M TrustZone as described above but with several significant enhancements. See Figure 3. The use of TrustZone with the RA is entirely optional for customer engineering teams; however, in our connected world, the confidence and reassurance of using these features to deploy IoT endpoints and edge nodes securely is highly recommended. In recognizing the growing need to provide resilient and robust security features for all applications, Renesas has taken the Arm TrustZone approach several

steps further to provide a secure separation and isolation regime that meets the future needs of embedded systems today.

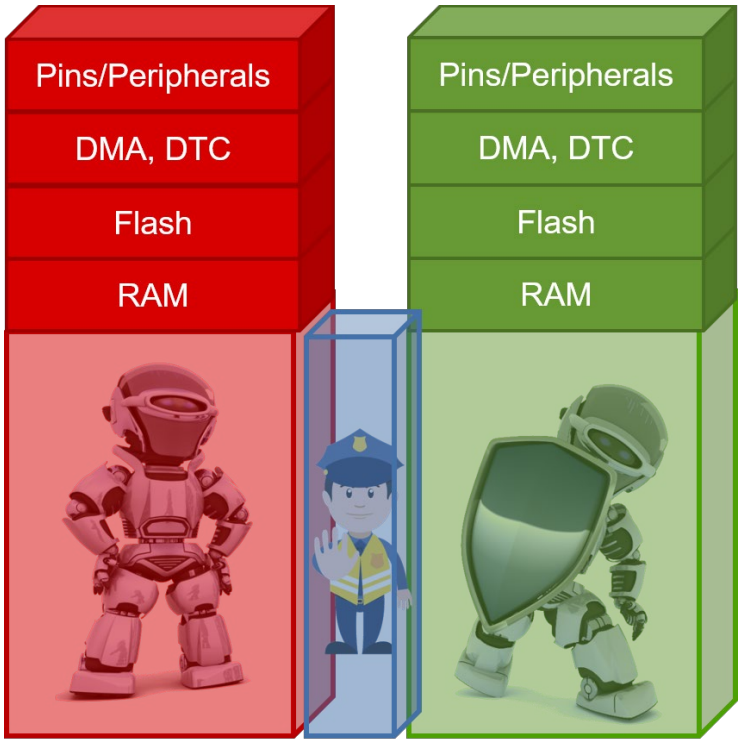


Figure 3: Renesas implementation of Arm TrustZone v8-M on the RA Family of Cortex-M33 microcontroller (source Renesas Electronics Corp.)

The RA Family of Cortex-M33 microcontroller implementation of Arm TrustZone prevents non-secure code from extracting secure code and data using DMA, DTC, and other memory access mechanisms. Also, the RA TrustZone filters are applied to all peripherals and pins to protect all external interfaces, prevent non-secure code from overriding outputs, and prohibit non-secure code eavesdropping on inputs.

Moving Forward from a Memory Protection Unit

Before the availability of Arm TrustZone for microcontrollers, Renesas recognized the importance of security for connected devices and innovated the security memory protection unit (MPU). The security MPU is a way to enhance a processor's security that does not have isolation and separation capabilities built-in; however, despite the capabilities that a security MPU adds to a design's architecture, several scenarios warranted further advances in isolation. Renesas has advanced the isolation and security capabilities of the Renesas RA Family line-up with the implementation of Arm TrustZone in the RA. See Figure 4.

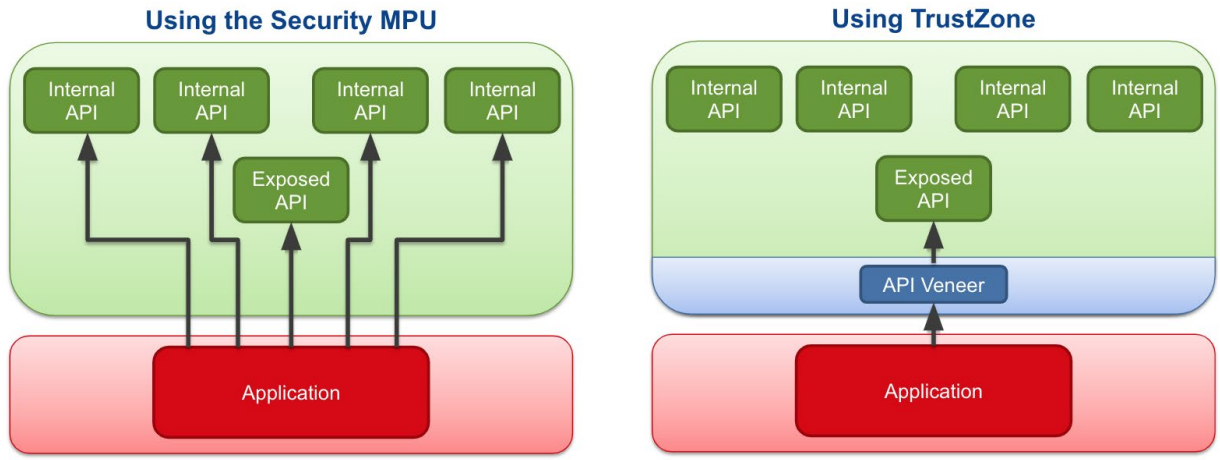


Figure 4: Isolating APIs from being exposed. A comparison of using a security memory protection unit and using a Renesas RA with Arm TrustZone. (source Renesas Electronics Corp.)

With the security MPU, the application logically accesses the APIs, and there is no way to restrict access to them. Consider a situation where the exposed and documented API might involve specific security functions or provide access to a commercially sensitive algorithm. As illustrated in Figure 4, with the security MPU, there is no restriction in how internal APIs are accessed. So long as you know where to access an API routine and what parameters you call it with, you can use it. Should you find out that a particularly useful subroutine within an internal API is accessible and the parameters available to use, you can still reach it using non-secure code despite it not being a recommended access method; however, with TrustZone, access to the internal APIs via the exposed API is enforced through hardware. Non-secure application code can only access the API veneer layer in the non-secure callable space. There is no mechanism for non-secure application code to access the functions of the internal APIs directly.

Another similar example is preventing code misuse. See Figure 5.

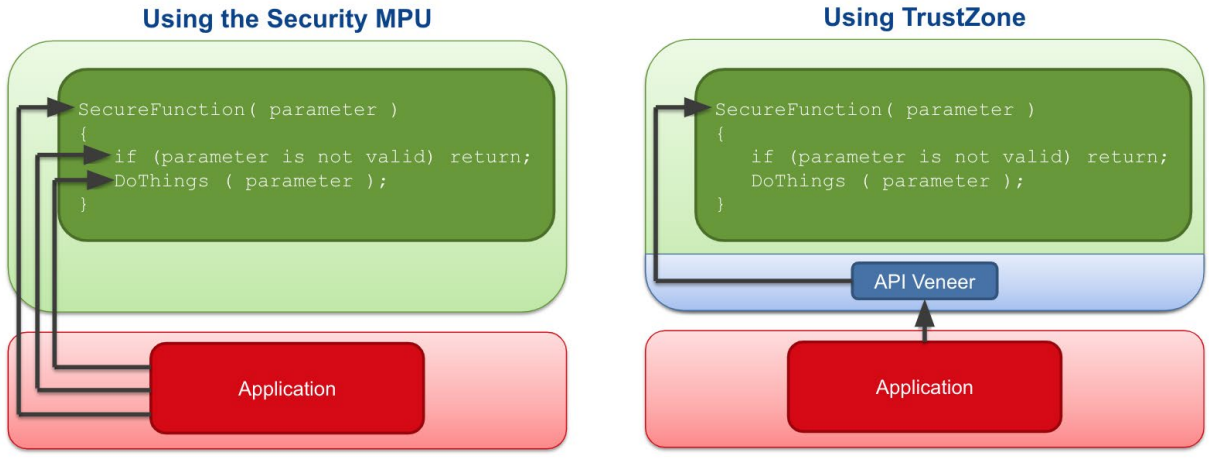


Figure 5: Preventing code misuse with Arm TrustZone (source Renesas Electronics Corp.)

Using the security MPU, non-secure access can bypass formal entry points by jumping to a specific address. Code misuse in this way, either accidentally or on purpose, is not restricted. Should a developer realize there may be some performance or operation advantages to be gained by the non-secure code jumping to a specific point with a function rather than the top, there is nothing the security MPU can do about it. With TrustZone, access to the secure code in the correct manner is hardware enforced. The non-secure code can only access the top of the secure code via a single-entry point, the API veneer. This approach inhibits any attempts to call secure code using other physical addresses.

The Renesas implementation of TrustZone also prohibits unrestricted access to external interfaces and pins. Non-secure code is prevented from directly reading or writing to pins and, in so doing, defends against eavesdropping and impersonation.

By restricting application access to APIs, preventing code misuse, and external interfaces, the potential attack surface available to adversaries is reduced.

Application Use Cases

The isolation capability that Arm TrustZone provides between non-secure and secure environments is suitable for several different use cases. This section of the white paper investigates three use cases; IP protection, code separation, and protecting the root of trust.

IP Protection

With the time and effort involved in creating specialist algorithms, such as a motor control algorithm, it is understandable that algorithm developers wish to protect their IP from being copied. One method of achieving this while still maintaining application flexibility is for the algorithm developer to pre-program the microcontroller's secure area with the algorithm and provide an API for application developers to access the algorithm's functions. See Figure 6.

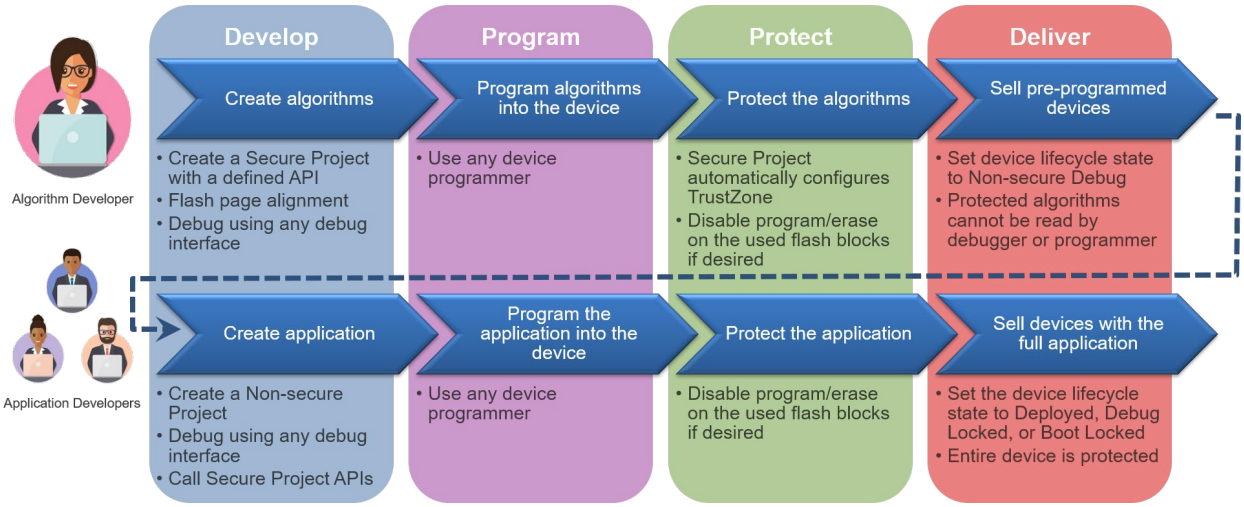


Figure 6: Application use case for pre-programmed algorithm IP protection using TrustZone (source Renesas Electronics Corp.)

Once the algorithm has been programmed into the secure area, the program and erase function of the algorithm's memory blocks are disabled. The algorithm developer can then sell these pre-programmed

devices ready for the application developer to program with a non-secure code that calls the functions via the non-secure callable veneer. The algorithm is wholly protected from unauthorized access and copyright infringement.

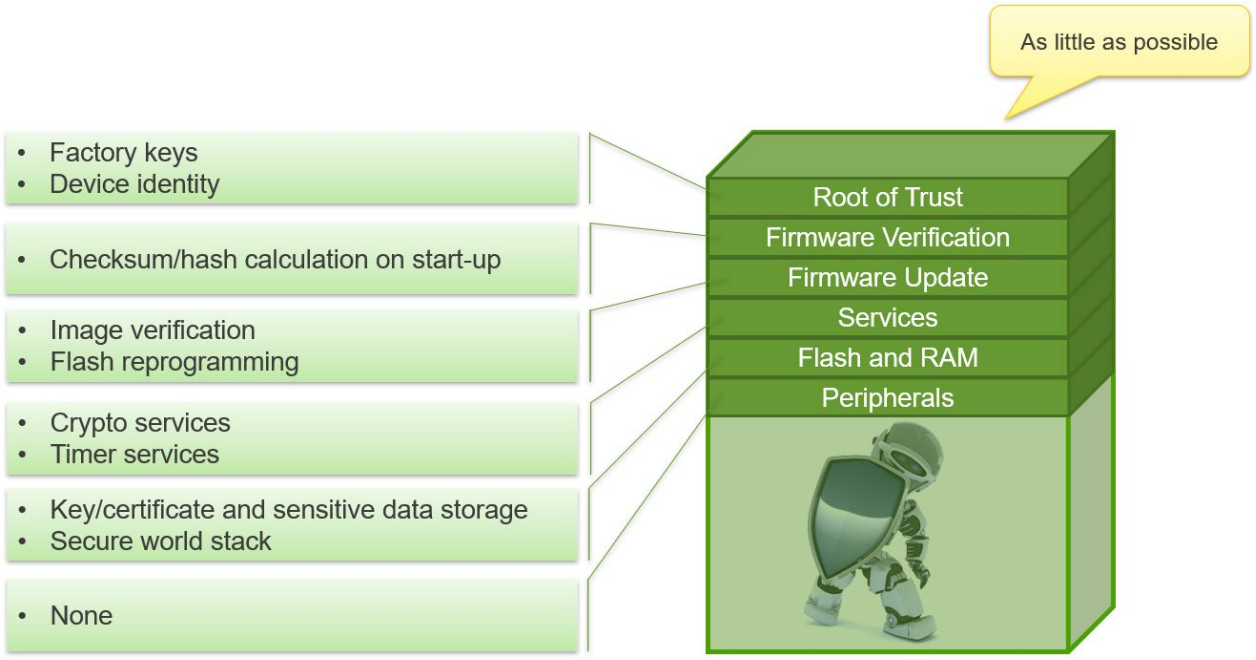
Code Separation

An excellent example of a code separation use case for TrustZone is for the European smart-meter specification. Within the measuring industry directive (MID) specification, aspects of the application code used for measurement and billing calculations must be certified. This code, identified as legally relevant code, must be isolated from the rest of the smart meter application. Currently, the only way of achieving this is through the physical separation of two microcontrollers; however, while it might simplify the certification process, the use of two MCUs involves an additional bill of material cost, a higher power consumption budget, and more PCB space.

An alternative approach is to use the logical separation provided by Arm TrustZone. All the legally relevant certified code can be placed in the secure area and the rest of the code that does not need certification can be placed within the non-secure area.

Protecting the Root of Trust

A device's root of trust (RoT) provides the bedrock foundation of security on which everything else is built. The RoT includes factory-set secret keys, authenticated firmware, device identity, and the credentials necessary for checksum calculation at start-up. If a device becomes compromised due to high-level security failures, it can be recovered from the RoT; however, if the RoT is breached, all the security built from it can no longer be trusted. It is highly recommended that all the root of trust elements be kept within the secure world – see Figure 7.



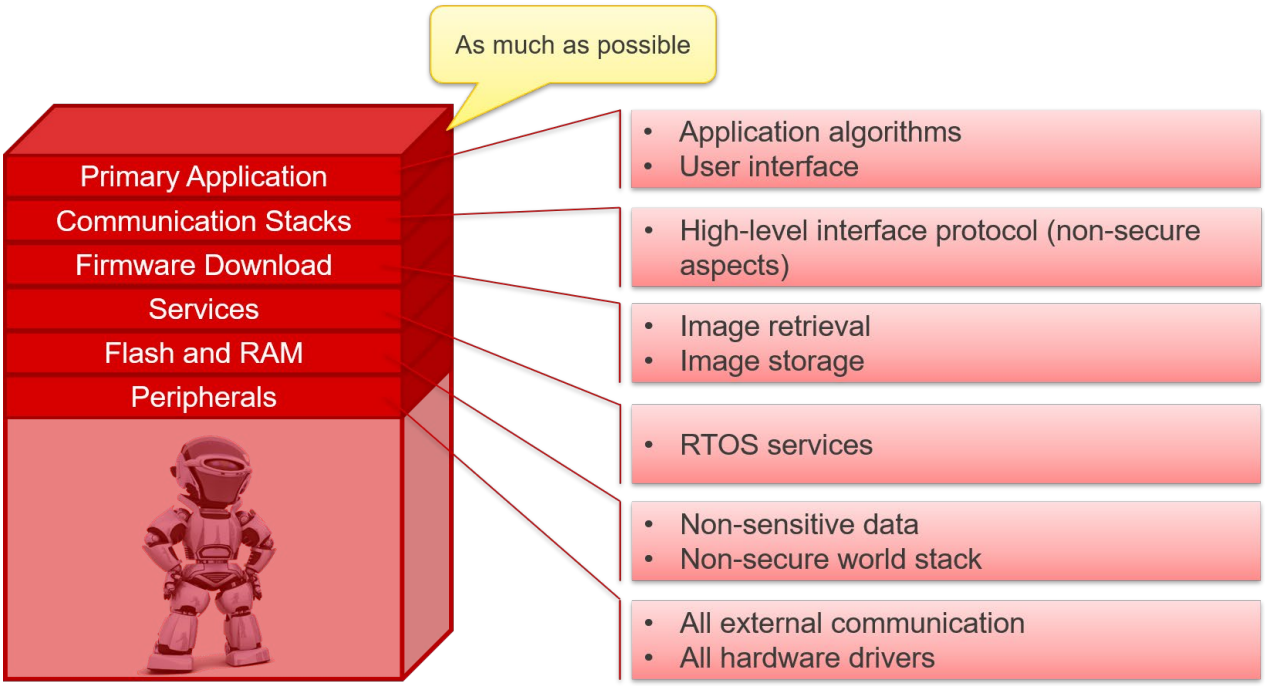


Figure 7: Using Arm TrustZone for a root of trust protection (source Renesas Electronics Corp.)

The Renesas RA Family Microcontroller with Cortex-M33

The Renesas RA Family with Arm Cortex-M33-based are based on the Arm v8-M TrustZone technology. Available in a variety of performance configurations from 100 MHz to 200 MHz, it integrates an enhanced security cryptographic engine and cryptographic accelerators within a secure element function. Security functions also include key management support, tamper detection, and power analysis resistance capabilities. Peripheral functions include a capacitive touch sensing controller, Ethernet, USB 2.0 Full Speed, SDHI, OctaSPI, QuadSPI, and SPI interfaces.

Development support for the RA Family includes the Renesas Flexible Software Package (FSP), a GitHub-based and easy-to-use ecosystem on which to create secure IoT applications. The FSP includes middleware for functions such as connectivity, security, and graphics, and hardware abstraction layer (HAL) drivers. Board support packages and FreeRTOS are also included in the FSP.

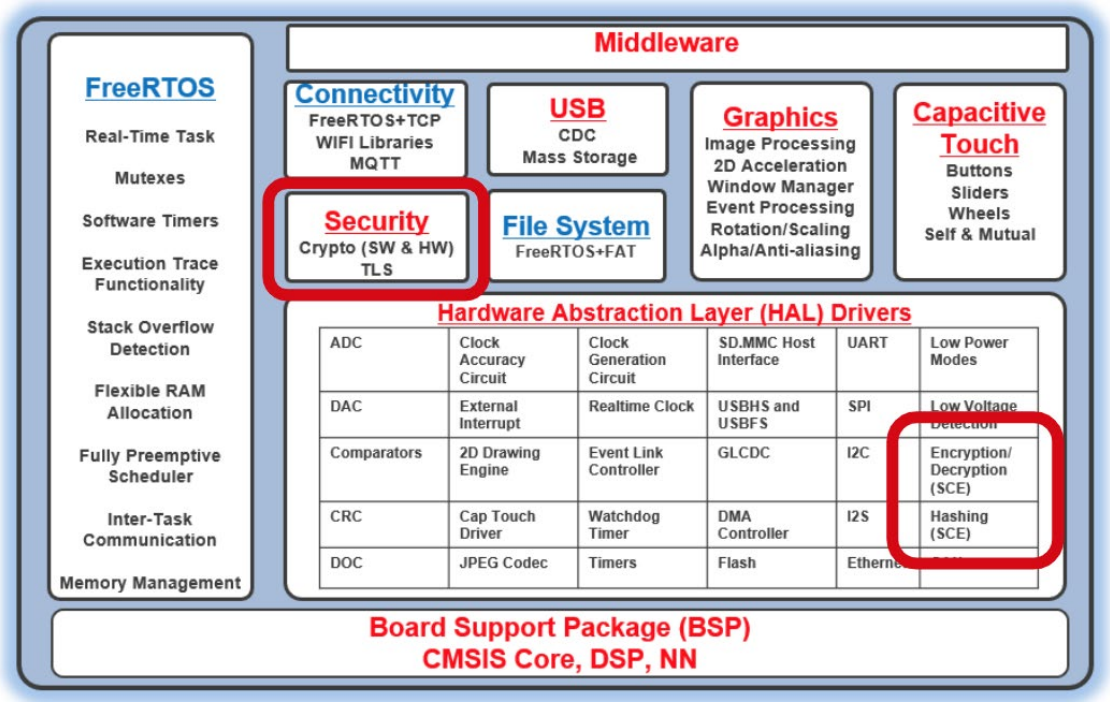


Figure 8: FSP feature overview (source Renesas Electronics Corp.)

The Renesas RA Partner Ecosystem lists a variety of partners that provide software and hardware building blocks and resources that accelerate the development of IoT applications.

Conclusion

The Renesas RA Family of microcontroller with Cortex-M33 series of high-performance low-power microcontrollers is the ideal platform for building a wide variety of industrial and consumer applications requiring the logical separation and isolation of code, security, and IP assets. The Renesas implementation of Arm TrustZone on the RA enhances the capabilities of TrustZone with the additional separation of peripheral interfaces, RAM and flash memory, and direct memory access requests.

Learn More

1. RA6M4 Product Page
2. RA4M3 Product Page
3. RA Partner Ecosystem
4. Flexible Software Package (FSP)
5. RA Family of MCUs
6. Arm® TrustZone®