PROVE & RUN

Proven Security for the Internet of Things (IoT)

77, avenue Niel, 75017 Paris, France

contact@provenrun.com

Prove & Run – Proven Security for IoT

Essential Software Components to secure the Internet of Things:

- **ProvenCore**: OS microkernel proven for security to secure smartphone, tablets, gateways and connected devices (industrial things)
- **ProvenVisor**: proven secure hypervisor for connected devices and IoT virtualization solutions

Main Competencies:

- Security & Architecture
- Operating systems
- Formal methods
- Security certification

Contact

- contact@provenrun.com
- www.provenrun.com

Security by Design Solutions:

- Transparent security perimeter around your preferred OS
- Secure isolation for security-sensitive applications (firmware update, authentication, firewalls, etc.),
- Cost effective and easy path to security certifications (up to the highest level)
- Security domain partitioning
- Security-in-depth

Management with > 45 years of combined experienced in digital security:

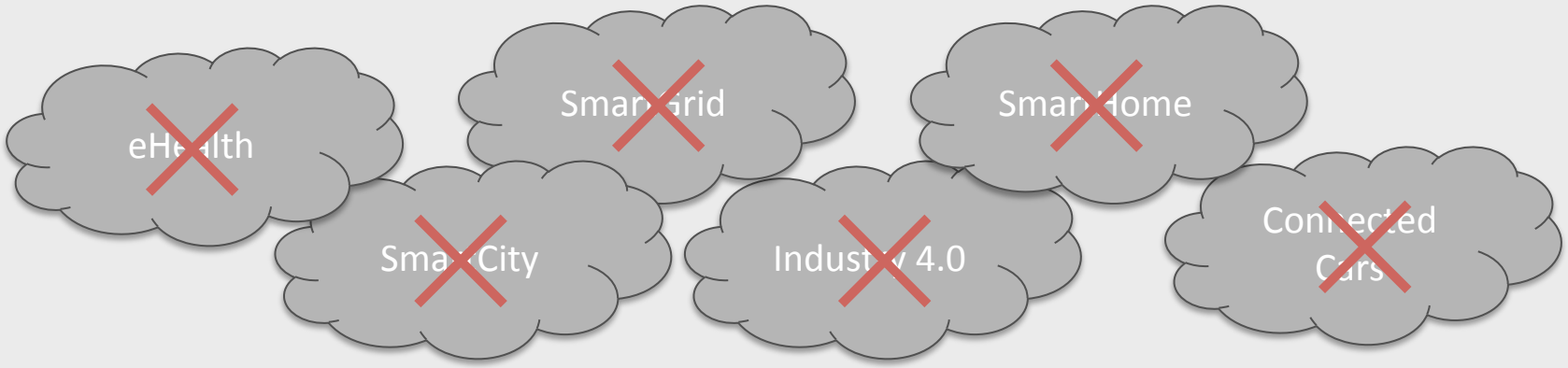
- Dominique Bolignano, previously Founder & CEO of Trusted Logic
- Christophe Pagezy, previously EVP of Gemalto

Prove & Run's mission

Enable the Internet of Tomorrow = Internet of Things + Security

Without security:

- Impossible to deploy a network of connected devices
- Impossible to scale the Internet of Things
- Impossible to trust a system to keep data private & confidential



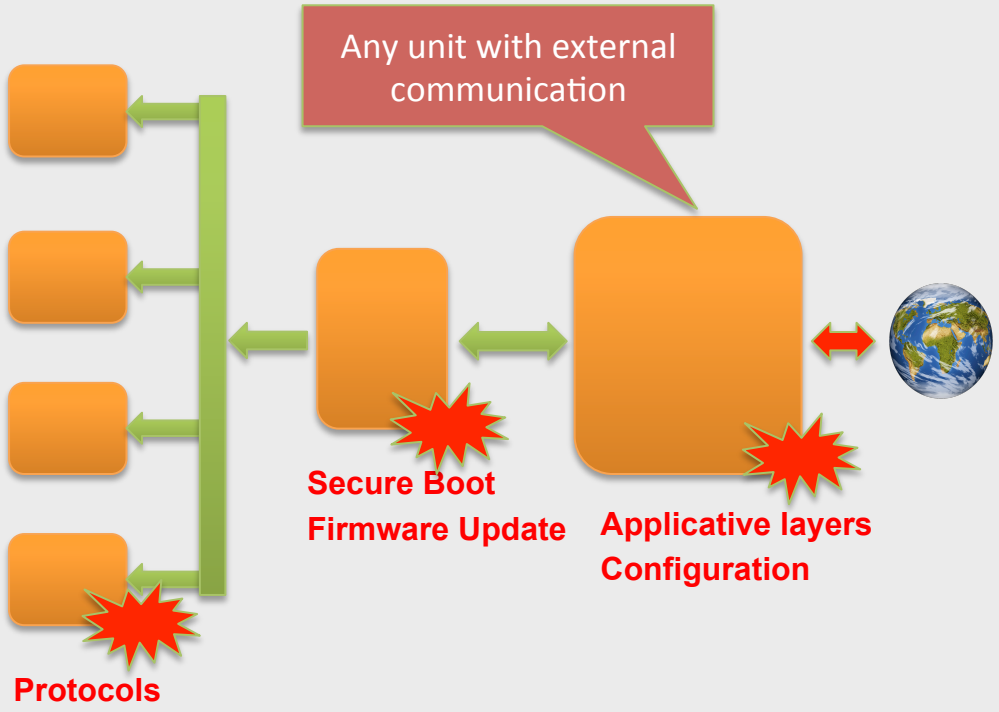
Hacking – Jeep example

Hackers Checking List

Increasing Complexity

Increasing Leverage

- Applicative layers,
- Protocols,
- Configuration,
- Personalization,
- Firmware update,
- Secure Boot,
- OS/Kernel,



Hackers use communications with the external world to exploit logical errors

Security is as strong as its weakest link

- **Security chain:**
  - Cryptographic algorithms
  - Cryptographic protocols
  - Technology and know-how to resist physical attacks
    - Ex: Smartcards
  - Technology and know-how to resist logical attacks
    - Hackers will exploit bugs, weaknesses and errors that exist in thousands in the software of embedded systems, in particular Operating Systems.
    - Existing OSs such as Android, Linux and large RTOSs cannot be technically secured and used as such:
      - 1000's of bugs officially reported / year

Security is changing …

- **Traditional: small TCB with few peripherals and small attack surface**
  - Secure element is usually the right solution
  - Resistance to physical attack is the biggest challenge
- **More peripherals and thus larger TCB and larger attack surface (typically mobile security)**
  - Use a small secure OS kernel (TEE),
  - Resistance to physical attack can be addressed with secure elements or similar embedded IP,
  - Resistance to logical attack becomes the biggest challenge

Security: the IoT disruption

- **IoT case: Still more peripherals, better business model for hackers, larger damages at stake, with large TCB and large attack surface, in many cases remote device is unattended, etc.**
  - Logical and Physical TCB are to be distinguished
  - Resistance to physical attack can still be addressed with secure elements or similar embedded IP
  - The secure OS kernel (such as the TEE), and all other complex parts of the TCB need to be formally verified
  - Resistance to logical attack is achieved using a trusted and reliable security rationale (attacks exploit error in the security rationale)

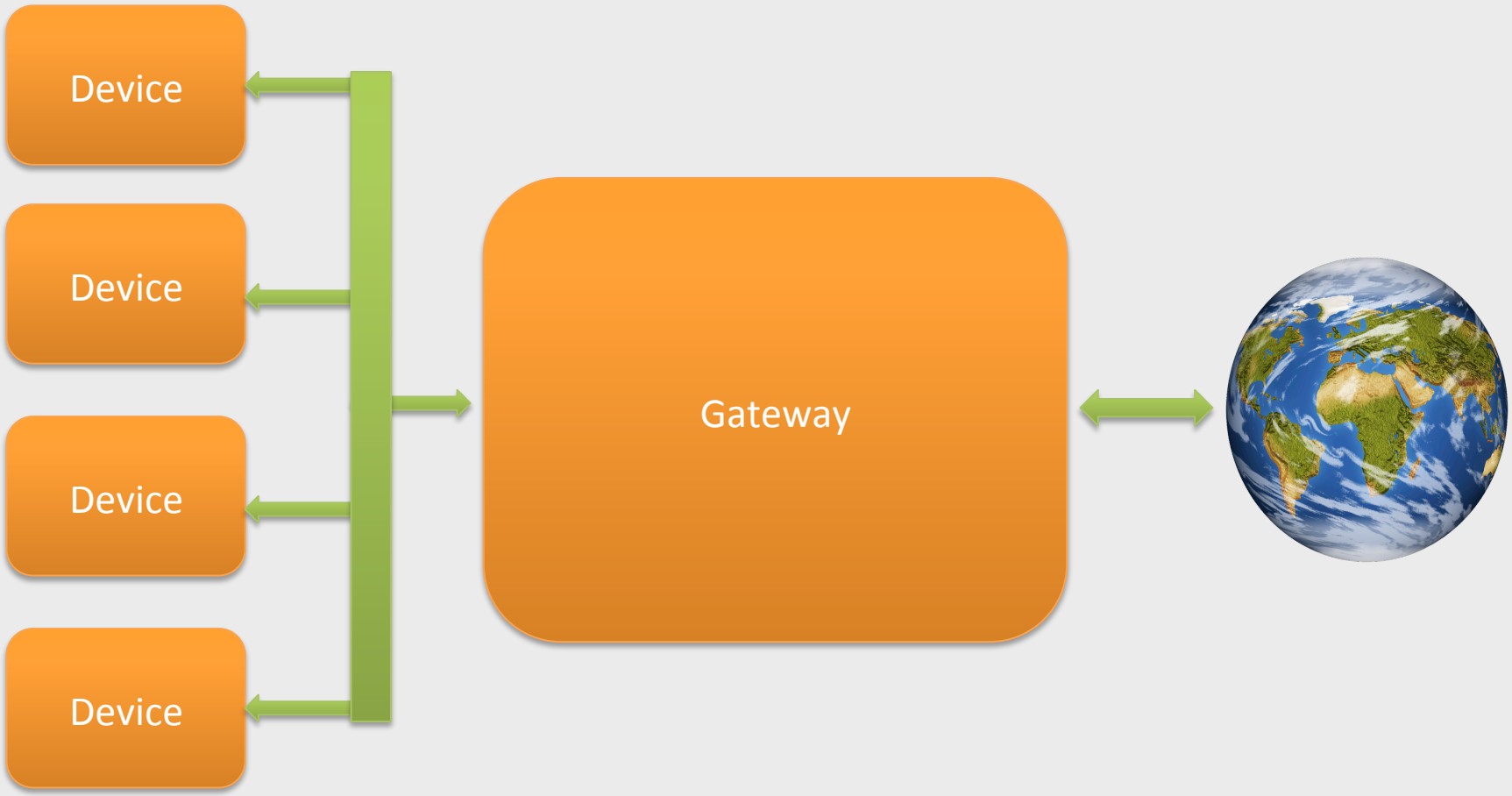
Addressing the New Challenge

- **Use of a state-of-the-art security methodology to clearly identify the security issues of the targeted system**
  - For example the Common Criteria methodology
  - The rationale of why security is achieved needs to be provided in an auditable format:
    - Perform a Risk Analysis
    - Confidence in rationale is key
    - Identify the “Trusted Computing Base” (TCB)
    - TCB should be small enough to be trustable
    - Large OSs such as Linux or Android when used should not be part of the TCB
- **For the OS and kernels that are included in the TCB;**
  - Apply formal methods to the complex part of the TCB (this includes kernels)
  - Ability to get as close as possible to “Zero-Bug”
  - Ability to demonstrate security (proof and certification)
- **Reach the highest levels of security at cost/skills requirements compatible with value chain constraints**
  - Reuse COTS to control the cost of developing a secure product

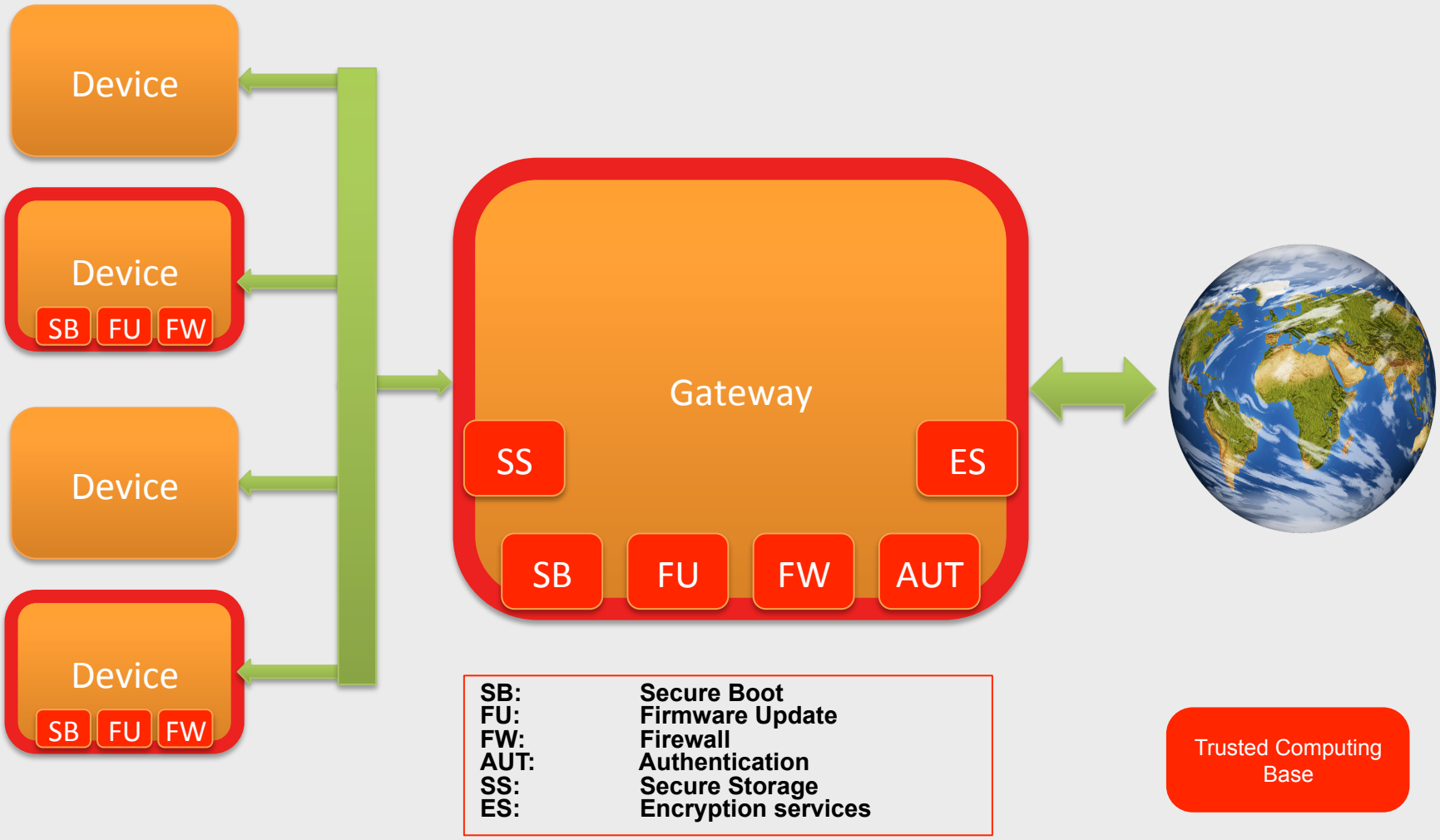
Prove & Run answer's to the challenge

- **Two critical secure COTS (ready for integration) that are needed to host "security sensitive" applications and to build layered security perimeters:**
  - ***ProvenCore***: Microkernel proven for security to secure gateways and connected devices (Industrial Things), smartphones, tablets, etc.
    - Execution of security-critical applications
    - Secure protection of the "Smart and Safe world" (Existing OS)
    - Provided together with its *Secure Boot*
  - ***ProvenVisor:*** Proven secure hypervisor for mobile devices and IoT virtualization solutions
    - Secure isolation of existing OSs and legacy SW stack
  - ***Built with ProvenTools***: A patented software development tool that makes it possible to formally prove the correctness of the software
    - Be as close as possible to "zero-bug"

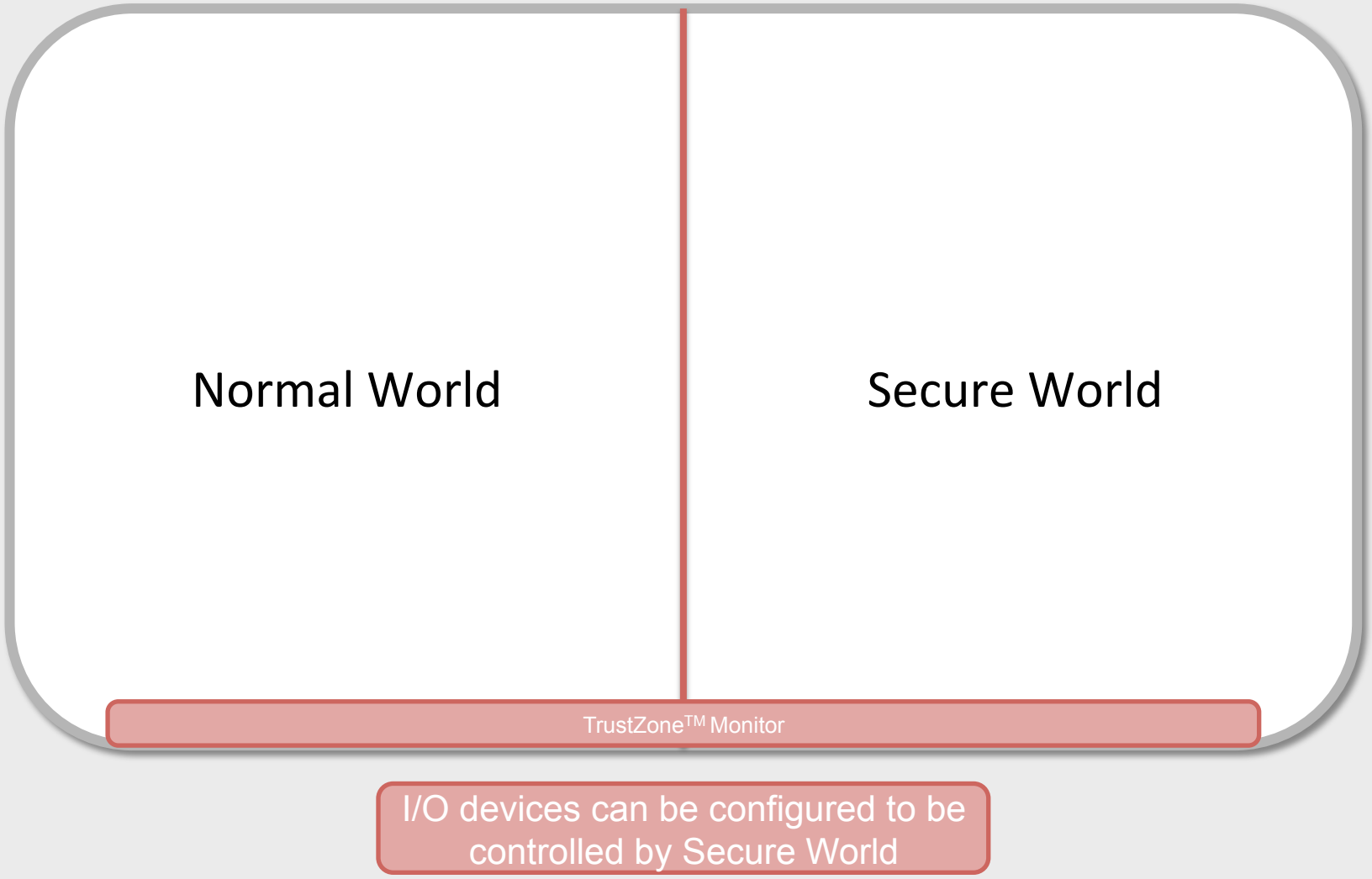
Securing a Simplified IoT Architecture



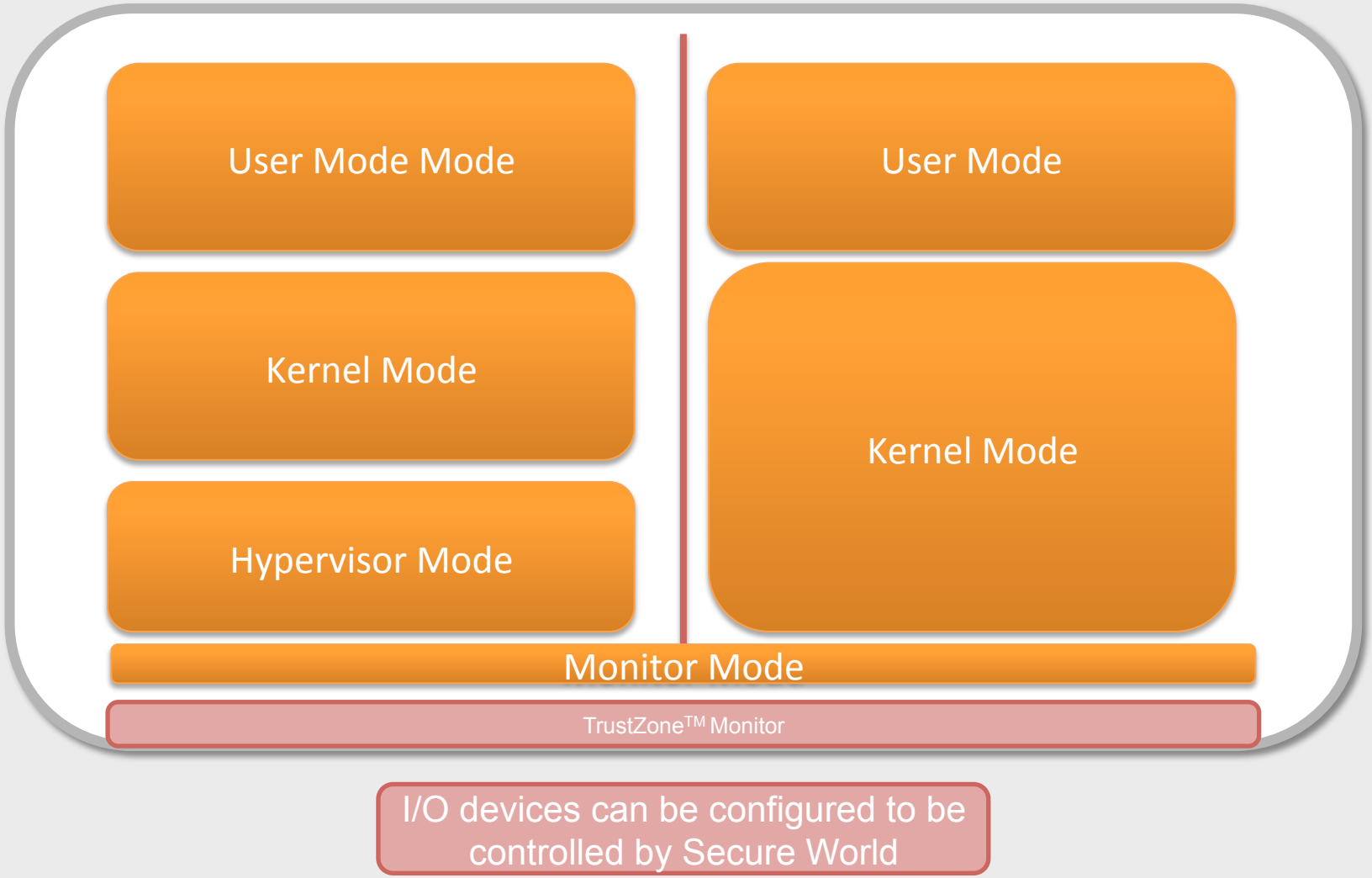
Securing a Simplified IoT Architecture



TrustZone ARM Cortex A – High Level Principles

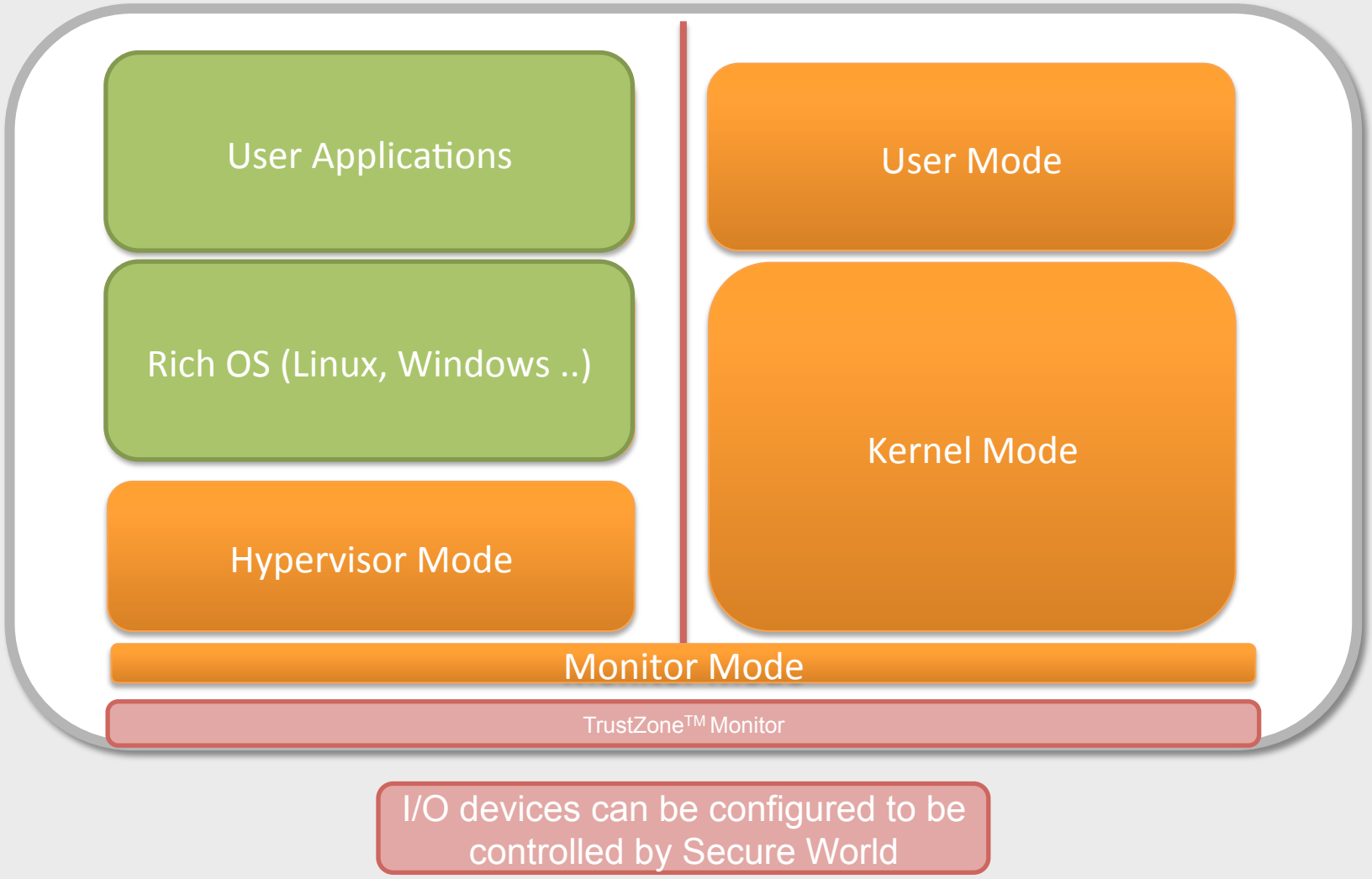


TrustZone ARM Cortex A – High Level Principles

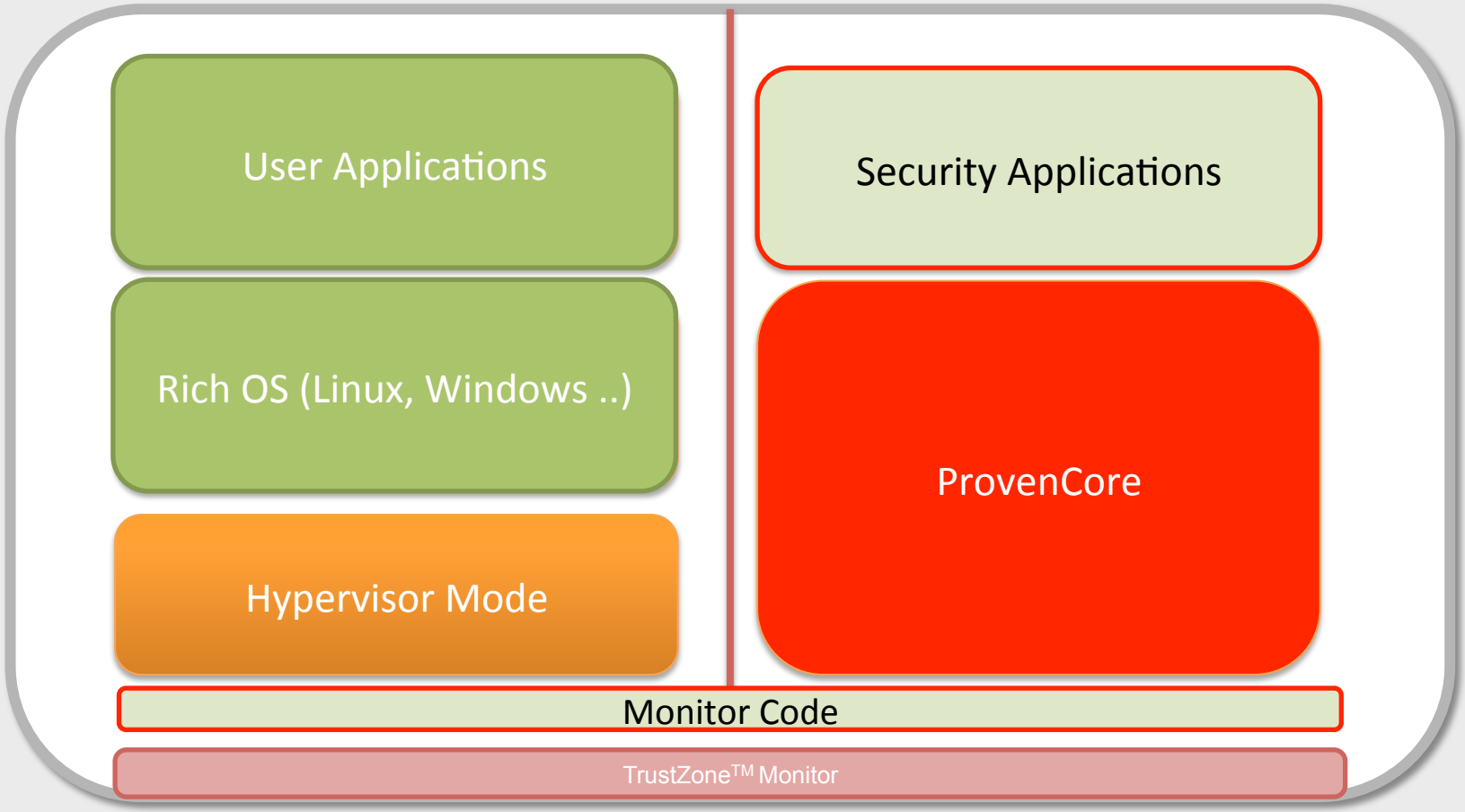


I/O devices can be configured to be controlled by Secure World

TrustZone ARM Cortex A – High Level Principles



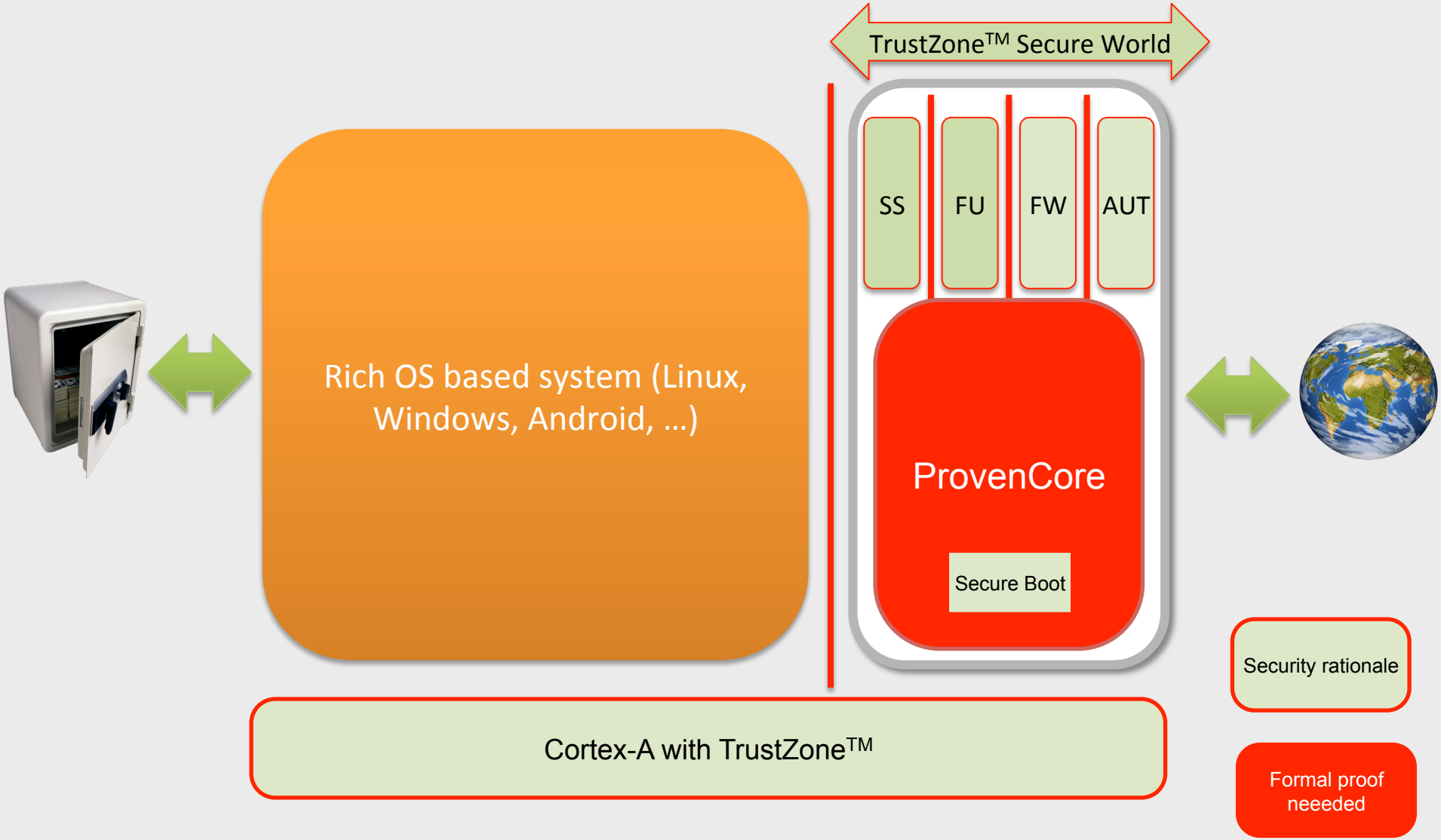
TrustZone ARM Cortex A – High Level Principles



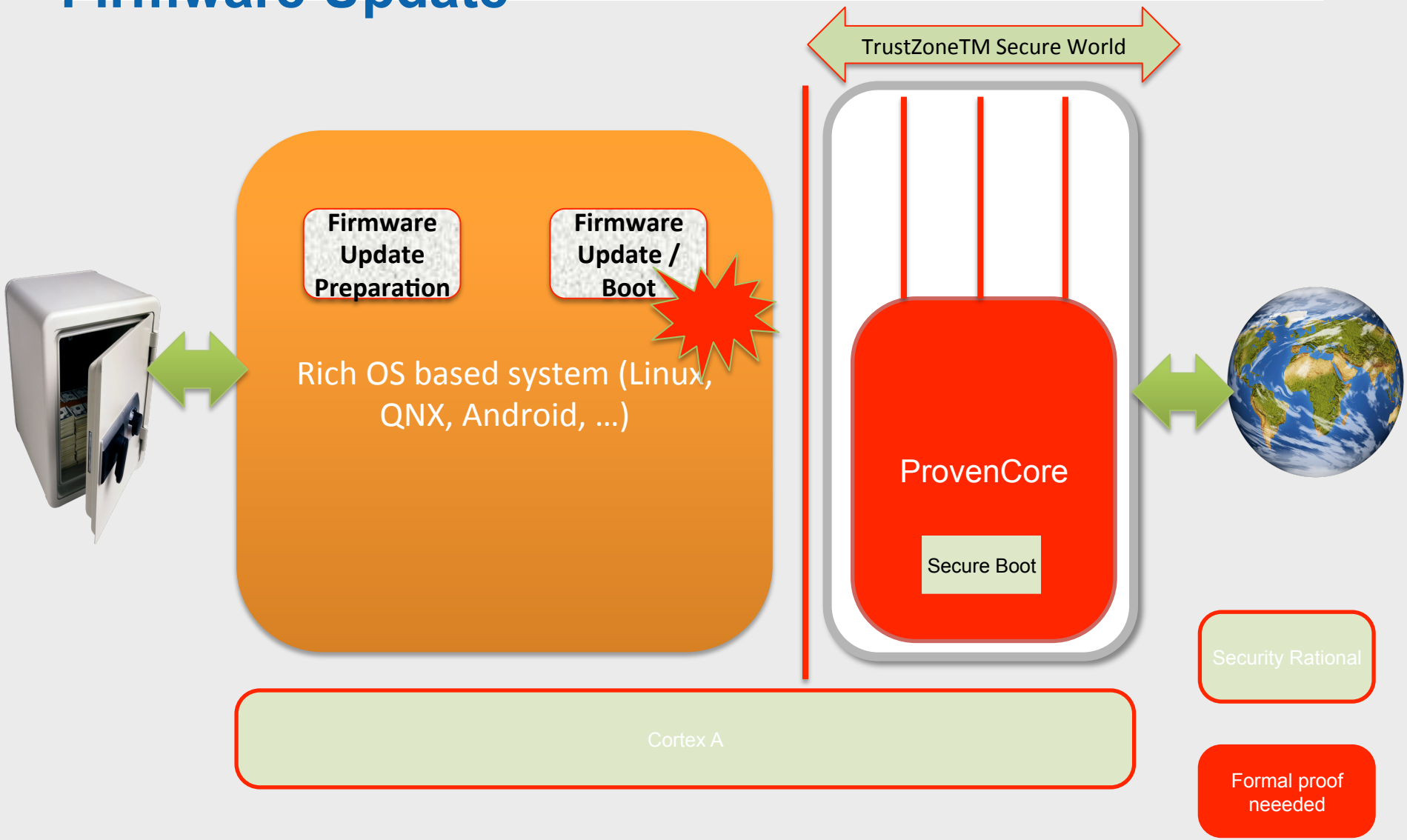
I/O devices can be configured to be controlled by Secure World

Formal proof neeeded

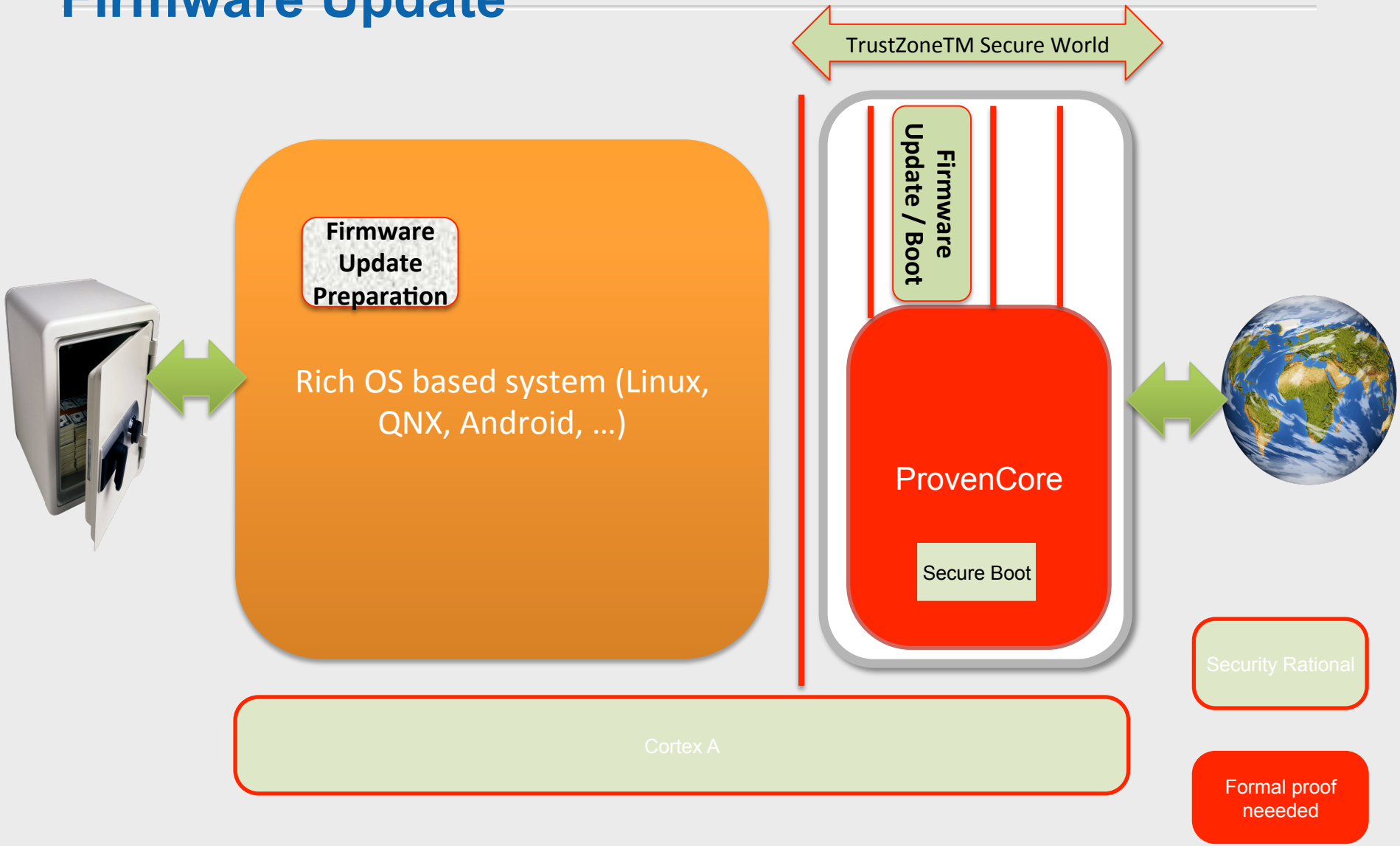
Securing an Entry Point on ARM Cortex-A



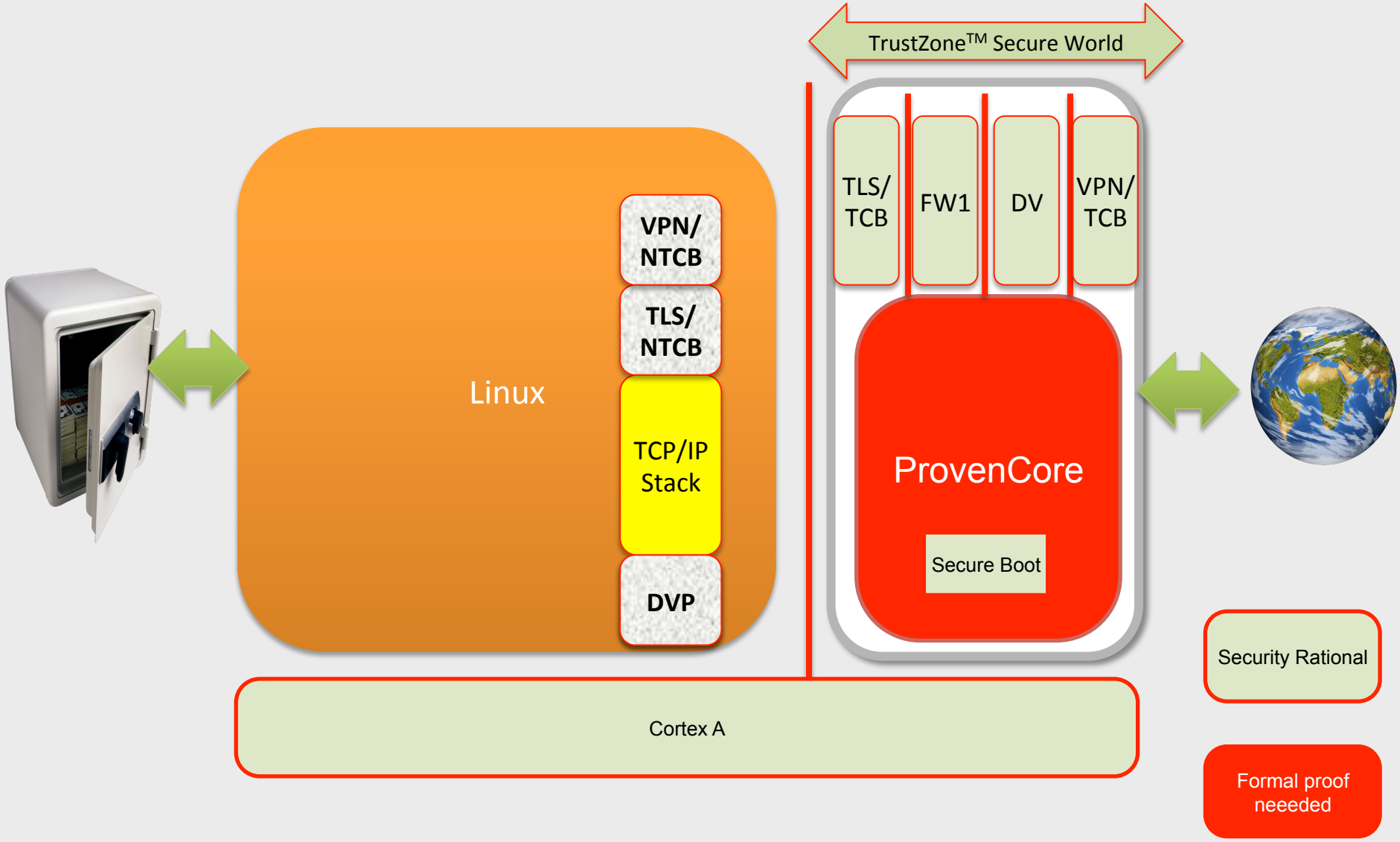
Looking more closely to the Secure Remote Firmware Update



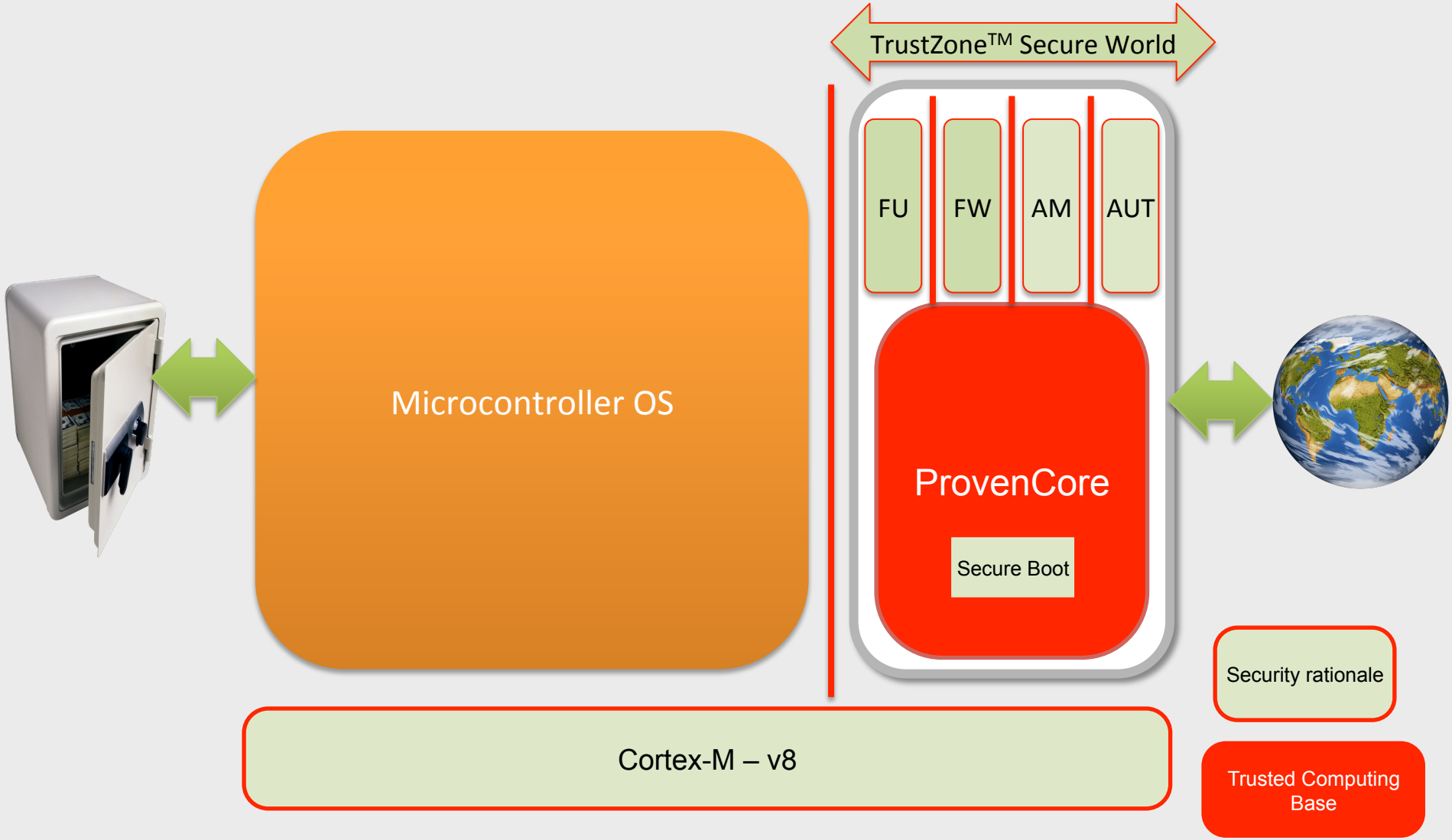
Looking more closely to the Secure Remote Firmware Update



Looking more closely to the TCP/IP Firewall



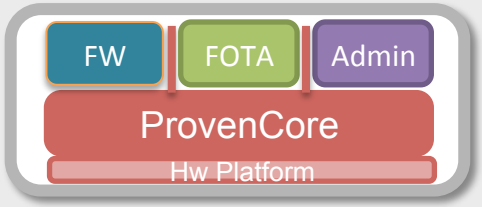
ARM next-generation microcontrollers (Cortex-M v8)



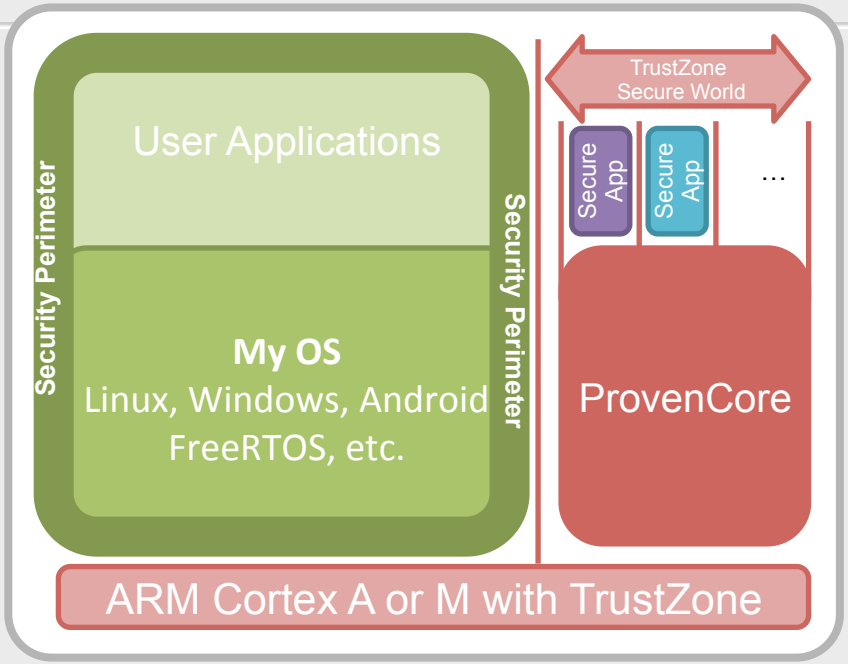
A secure OS kernel is always required

- **You need to have security applications to do various tasks:**
  - Filtering various communications channels, Firmware Update (FOTA), Using and managing keys, Administrating configurations and security, Logging events, possibly Performing various analysis and attack responses, etc.
- **You need to place such secure applications on a trusted and robust ground:**
  - Not on a large untrusted OS such as Linux (even sitting on a hypervisor, as it will have to communicate and interact with the peripherals and is thus vulnerable)
  - Not on hardware,
  - Not on a hypervisor (which would provide by definition a similar hardware abstraction)

Requires a secure OS/kernel



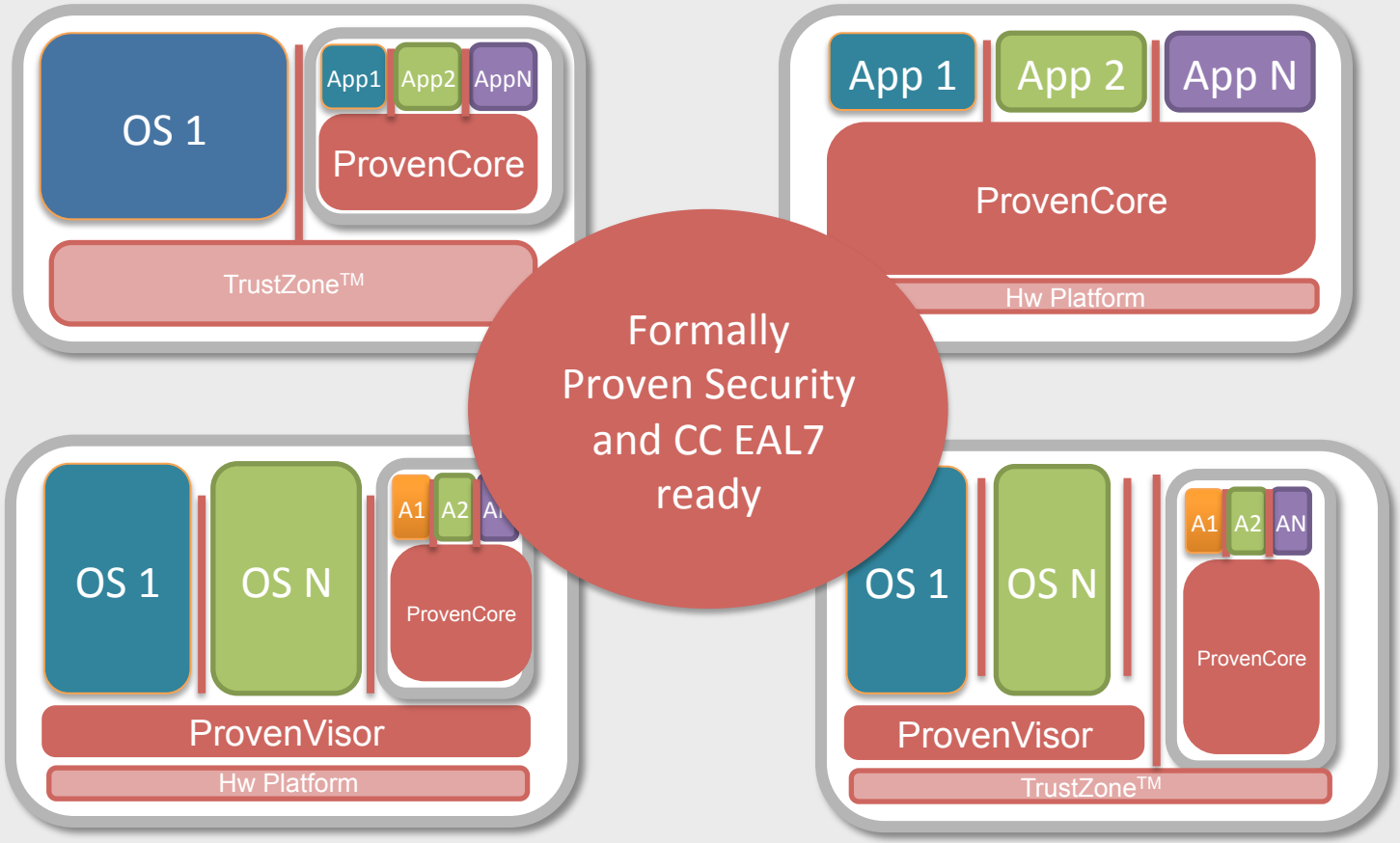
Example: Security Perimeters for Connected Device



- Your Device need to resist to hackers?
- But you want to use your preferred OS and applications, despite the fact it is susceptible to attacks?

- Transparent security perimeter around your preferred OS
- Secure isolation for security-sensitive applications (firmware update, authentication, firewalls, etc.),
- Cost effective and easy path to security certifications (up to the highest level).

With ProvenCore and ProvenVisor, Secure a Smart and Safe Embedded World



The 2 missing bricks needed to create the Internet of Tomorrow