PROVE & RUN

Security for the Internet of Things

Dominique Bolignano

77, avenue Niel, 75017 Paris, France

contact@provenrun.com

Our mission

- **Help our customers resolve the security challenges linked to the deployment of connected devices**

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

A few recent examples of vulnerabilities affecting the IoT@Home

- 04/21/2014 - DSL router patch merely hides backdoor instead of closing it
  - http://bit.ly/1jC5AAu
- 10/23/2014 - All VeraLite Home Gateways share a single SSH private key stored in ROM
  - http://bit.ly/1uUXmb2
- 04/07/2015 - 6 common home gateways suffer from significant or very significant security issues
  - http://bit.ly/1NRy4V5
- 04/18/2015 - An OTA software update bricks Wink Hubs
  - http://hubfix.wink.com
- 05/20/2015 - At least 700 000 routers given by customers to ISPs are vulnerable to remote hacking
  - http://bit.ly/1Gw0wcO

A few recent examples of vulnerabilities affecting the IoT in Industrial & Smart City

- 05/08/2014 – Vulnerability in traffic-lights management systems leaves them wide open to modifications by hackers
  - http://bit.ly/QyPK0G

- 12/23/2014 – Cyber-attack on German steel mill inflicts serious damage
  - http://bit.ly/1t1nWF1

- 03/12/2015 - US industrial control systems attacked 245 times in 12 months
  - http://1.usa.gov/1DfWPdd

- 05/11/2015 – The Open Smart Grid crypto protocol used by 4 millions smart meters revealed as “extremely weak”
  - http://bit.ly/1bJ62ic

A few recent examples of vulnerabilities affecting the IoT in the Automotive world

- 07/21/2014 - Students hack Tesla Model S, make all its doors pop open IN MOTION
  - http://bit.ly/1rE7OEJ

- 02/16/2015 - 2.2M BMW cars can be unlocked with a simple smartphone
  - http://on.ft.com/1evJuUb

- 20/05/2015 – Thief use jammer to prevent entire car owners to lock their car over an entire car park
  - http://bit.ly/1JH28Eg

A few recent examples of vulnerabilities affecting the IoT in Avionics

- 04/15/2015 – Security researcher Chris Roberts arrested on suspicion of hacking flying planes
  - http://bit.ly/1ILeoCT

- 05/01/2015 - Boeing 787 software bug can shut down planes' generators IN FLIGHT
  - http://bit.ly/1DGP4HM

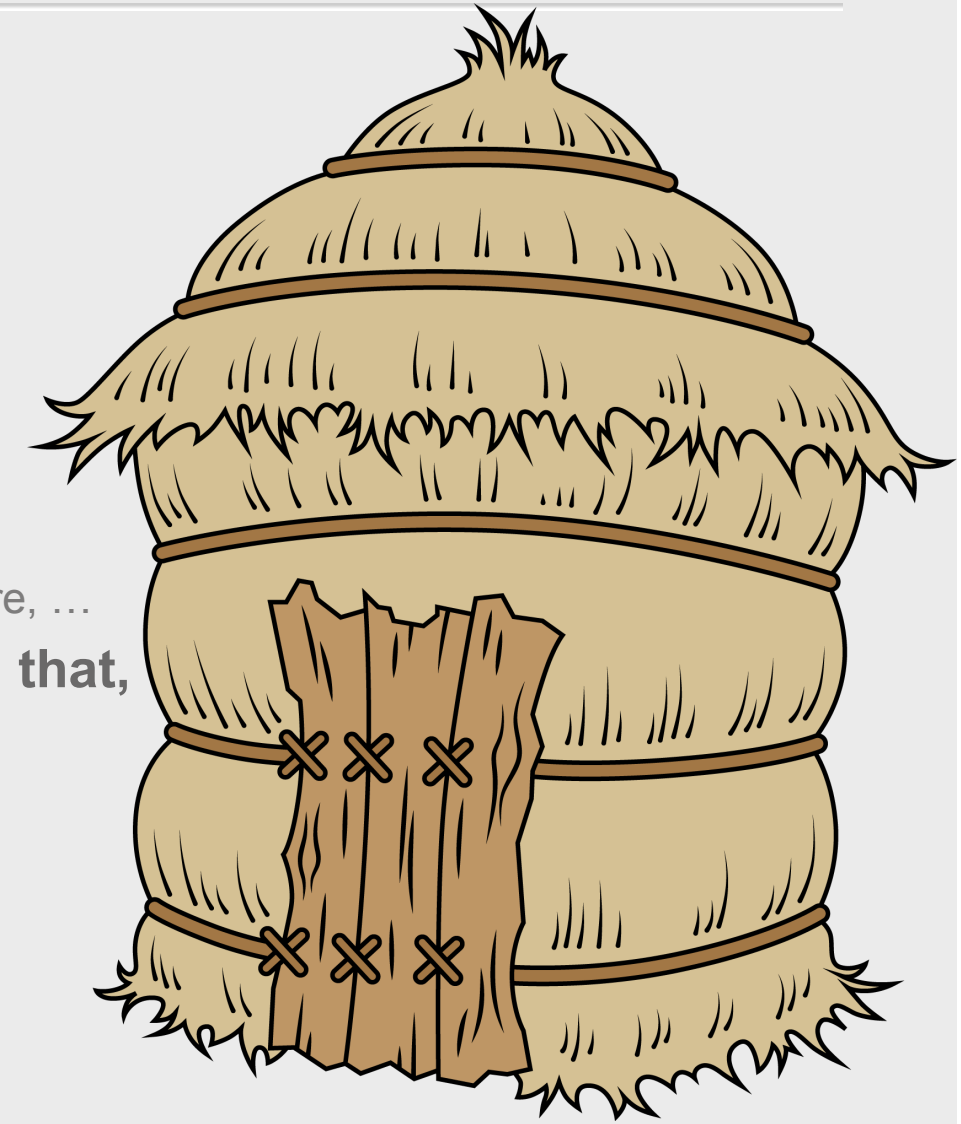
A few recent examples of vulnerabilities affecting the Mobiles

- 05/22/2015 - Factory reset memory wipe FAILS in 500 million Android smartphones
  - http://bit.ly/1JH28Eg

- 04/22/2015 – "Evil" WiFi signal crashes iPhones and iPads in range, even with WiFi turned off
  - http://bit.ly/1G54eZ9

Security is a serious matter

- **Many claim to achieve security**
  - **Just because they :**
    - encrypt,
    - sign,
    - use TLS,
    - a secure element,
    - or even just use a Java architecture, …
- **But security is much more then that,**



On the uses of formal methods for cybersecurity

- Security chain:
  - Cryptographic algorithms
  - Secure elements (e.g. smartcards)
  - Cryptographic protocols
  - Robustness of systems to logical attacks

- Issues with errors and vulnerabilities, particularly in operating systems:
  - An already alarming situation which is still degrading (e.g. the NIST database statistics).

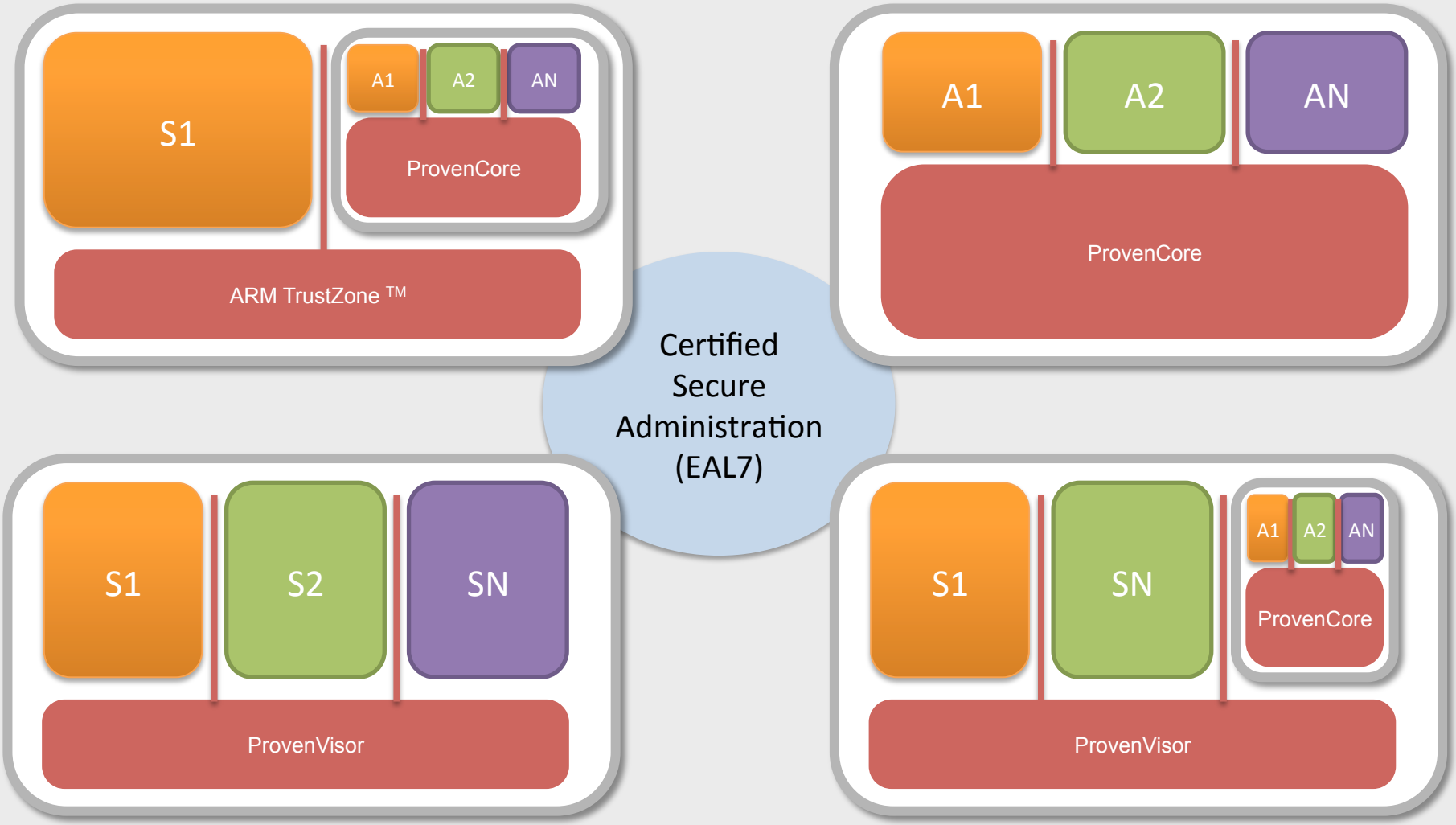
The main challenge is to secure the software

- **Situation on the software side needs to be improved …**
  - For security, every default/bug in either of the architecture, design, configuration or implementation is a potential source of attack
  - It is thus not possible to **directly** protect against attacks Oses such as iOS, Android, Linux, large RTOS ... There are issues with:
    - Size of the software stack to secure
      - "Trusted Computing Base" (TCB) includes kernel whose size and complexity are too big to build trust (and correctness of security properties)
- A basic partial answer:
  - Making weaknesses more difficult to exploit
    - Constraining the software
  - Drawbacks: user experience and security level.
- The global answer:
  - Defining a security architecture with a well defined and reduced-in-scope TCB
  - Applying formal methods to this TCB
    - Software development tools
      - Ability to get as close as possible to "Zero Bug"
    - Ability to demonstrate security (proof and certification)

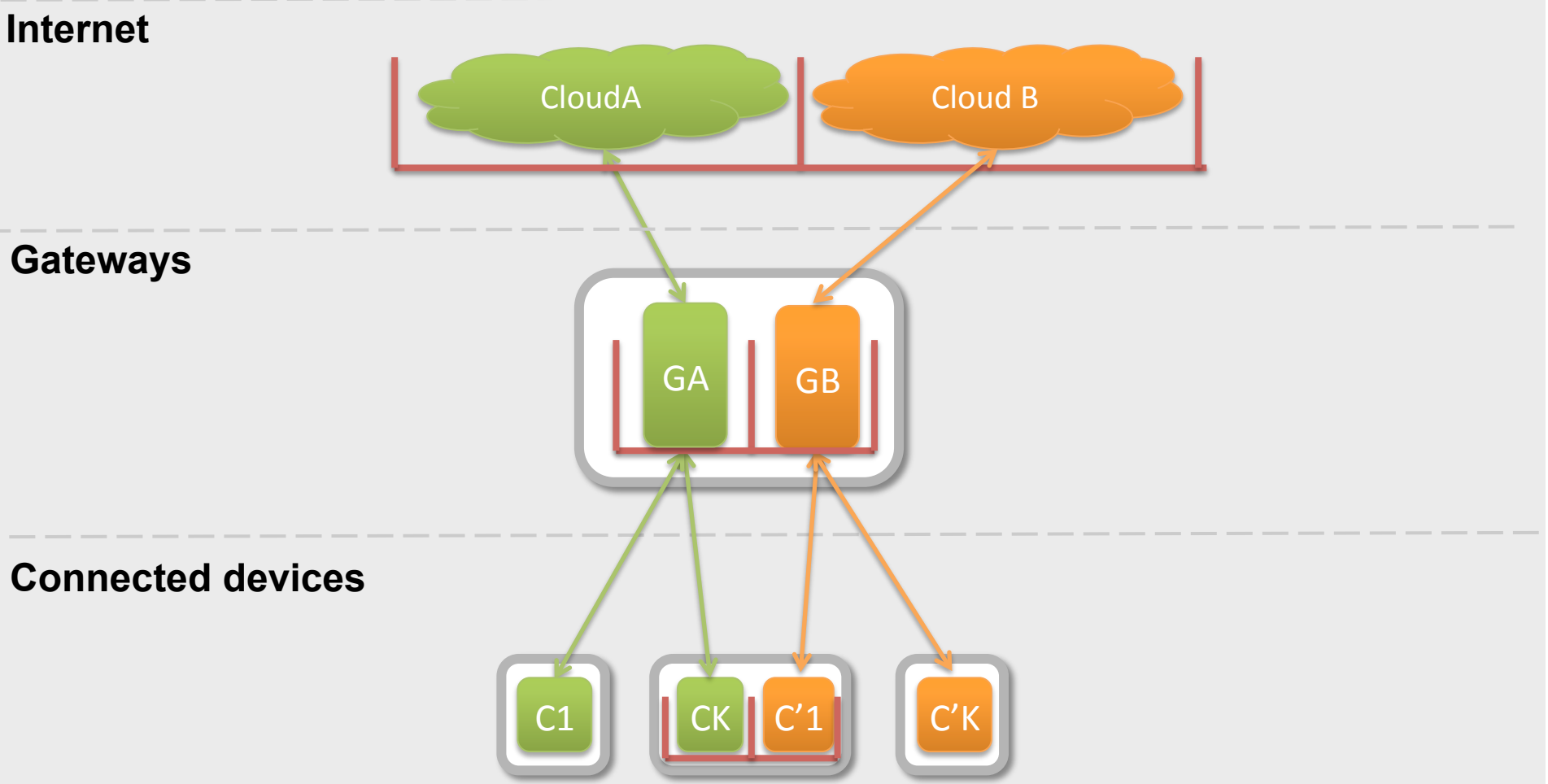
Prove & Run answer's to the challenge

- **ProvenTools:** a patented software development tool that makes it possible to formally prove the correctness of a security component
  - Specifically designed for handling complex security properties.
- **Critical secure COTS ready for integration**
  - *ProvenCore* : formally proven micro-kernel to protect the security of devices at the highest level
  - *Proven Mobile Stack* : bulletproof applicative framework to secure smartphones and tablets.
  - Others: TEE, Hypervisor (ProvenVisor) and IoT solutions
- **Security Professional Services**
  - Help our customers to design/build/develop secure software and/or integrate our COTS

Prove & Run Bricks to secure IOT



Isolation: Key to security architecture



Use Cases


Android (or any rich OS) with its applications
Freescale i.MX 6/i.MX 7

Use Cases

Android (or any rich OS) with its applications

Freescale i.MX 6/i.MX 7

Infotainment
Car gateway

Use Cases

Android (or any rich OS) with its applications

Freescale i.MX 6/i.MX 7

Smart Metering

Use Cases

Android (or any rich OS) with its applications

Freescale i.MX 6/i.MX 7

S Monitoring
Maintenance

Use Cases

Android (or any rich OS) with its applications

Freescale i.MX 6/i.MX 7

Home / Office Gateway

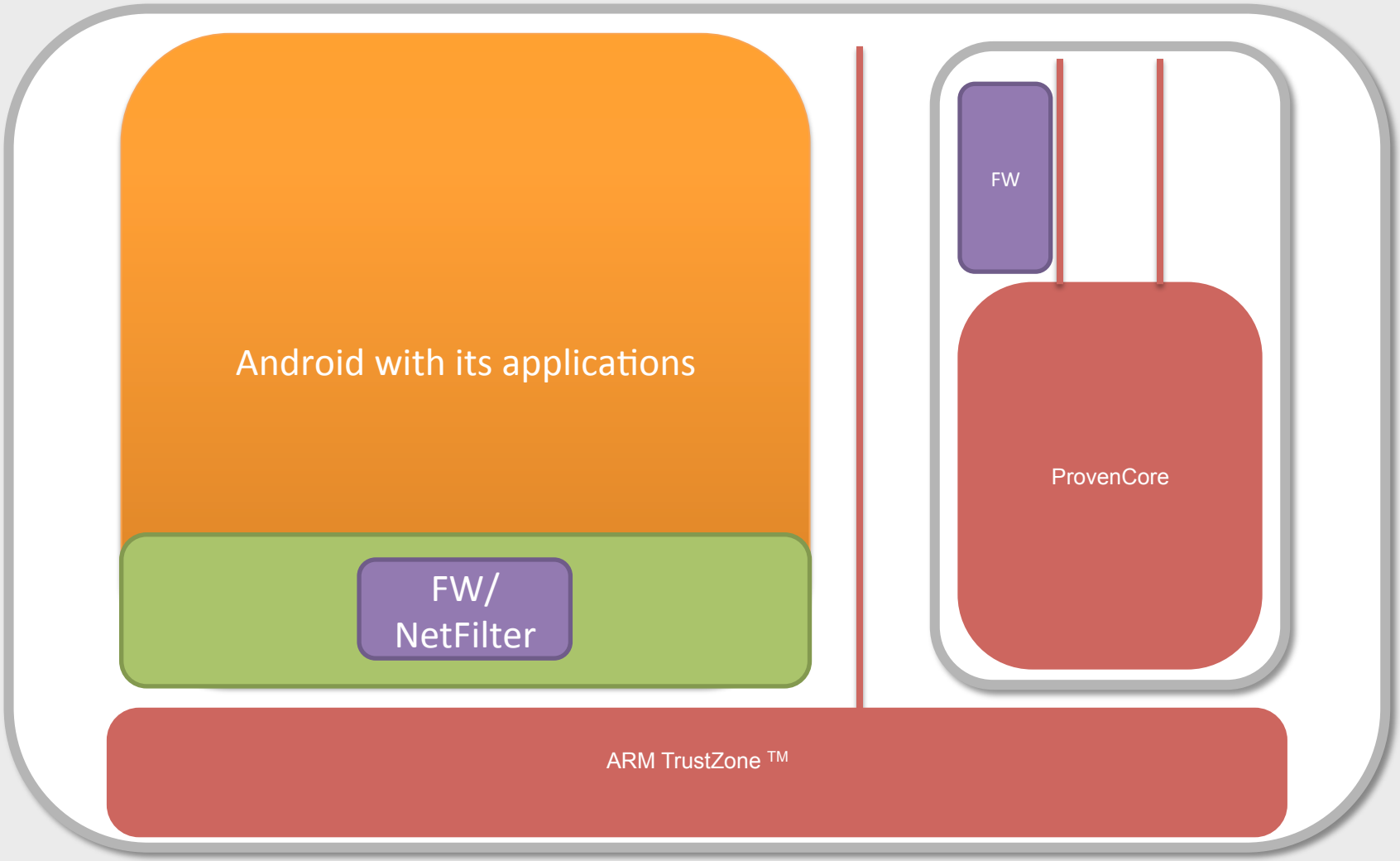
Use Cases

Android (or any rich OS) with its applications

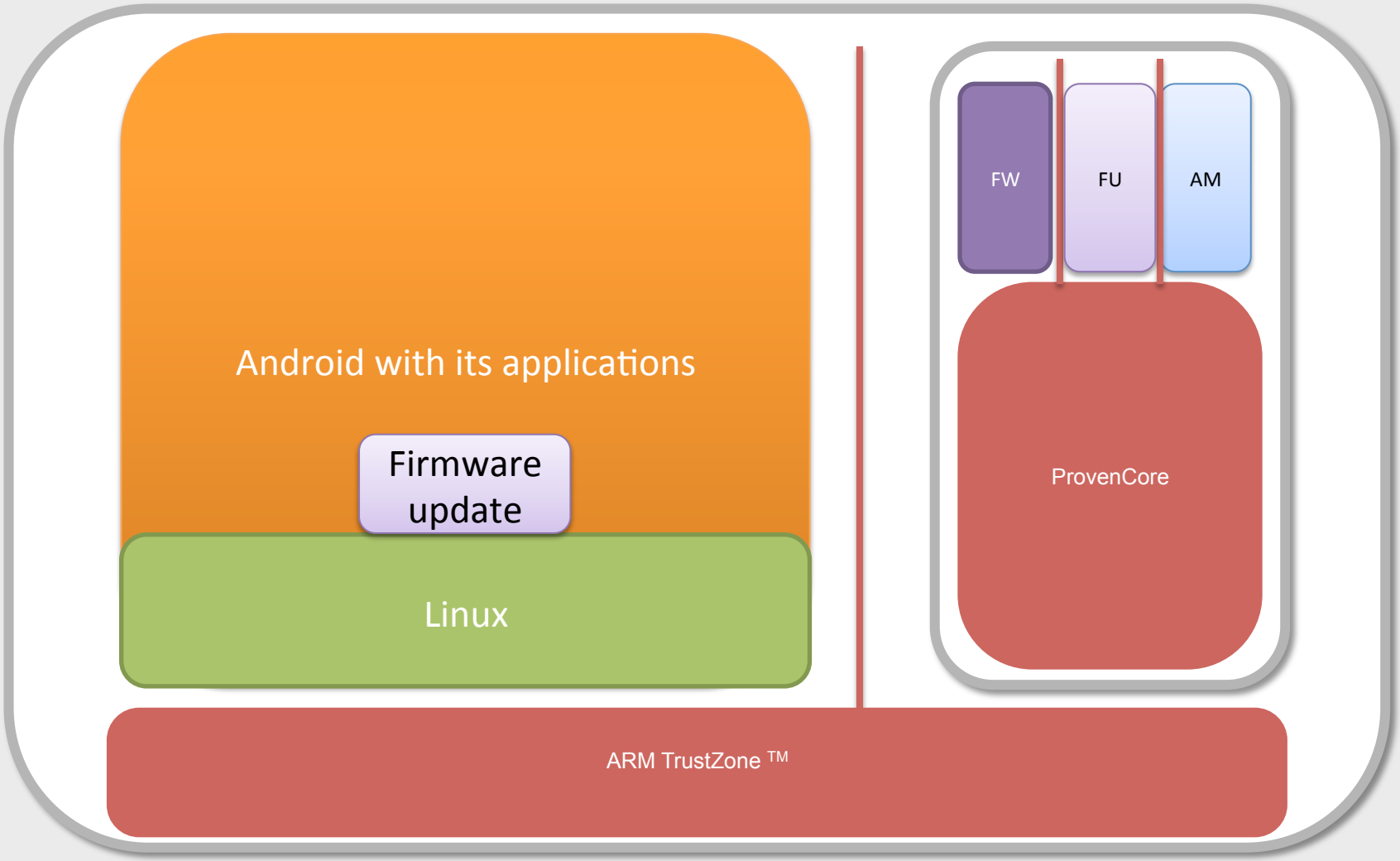
Freescale i.MX 6/i.MX 7

Industry 4.0

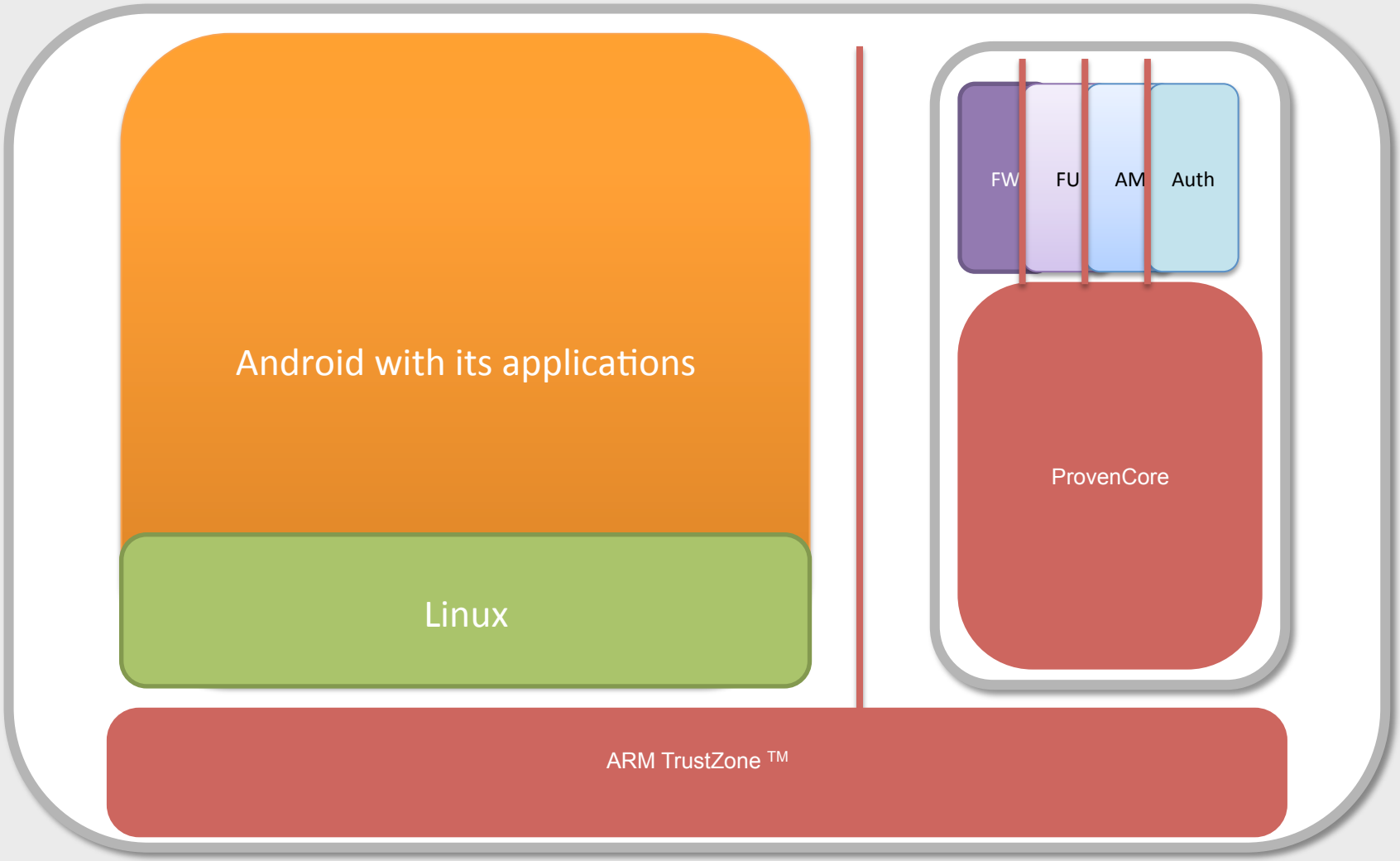
Use Cases


Android with its applications
FW/
NetFilter
FW
ProvenCore
ARM TrustZone ™

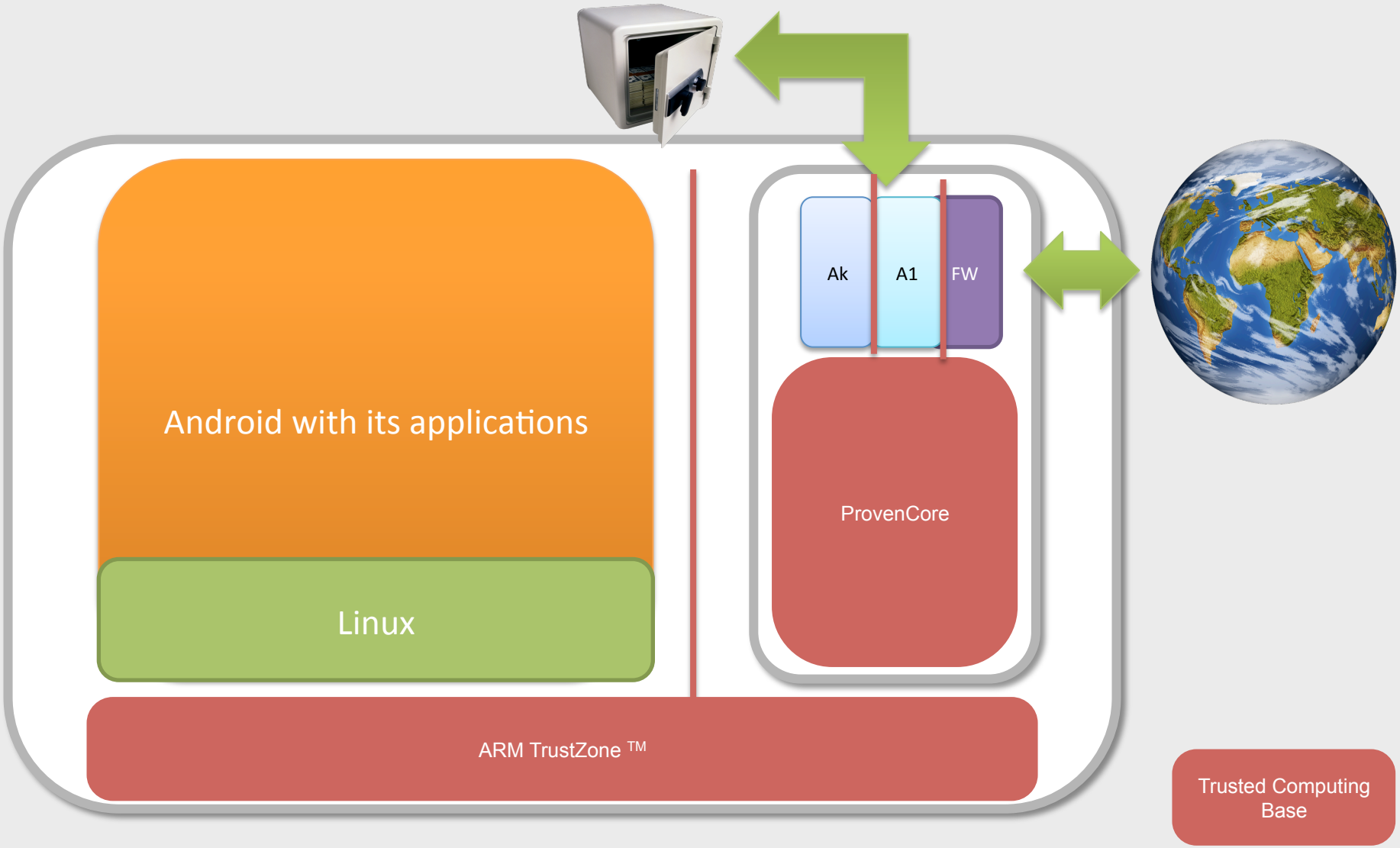
Use Cases


Android with its applications
Firmware update
Linux
FW
FU
AM
ProvenCore
ARM TrustZone ™

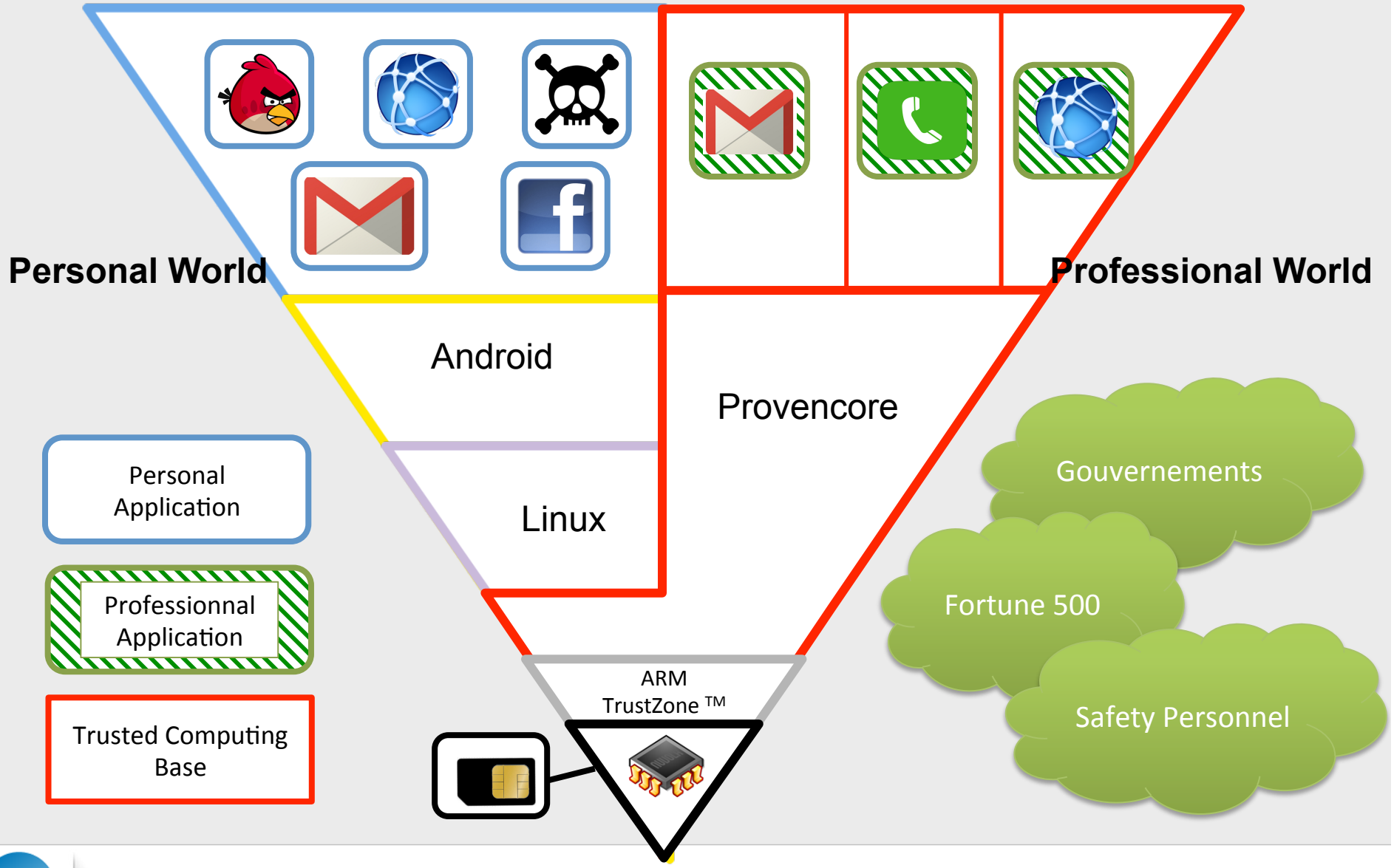
Use Cases


Android with its applications
Linux
FW
FU
AM
Auth
ProvenCore
ARM TrustZone ™

Use Cases


Android with its applications
Linux
Ak
A1
FW
ProvenCore
ARM TrustZone ™
Trusted Computing Base

Proven Mobile Stack – Secure Smartphone (BYOD)



Use Cases

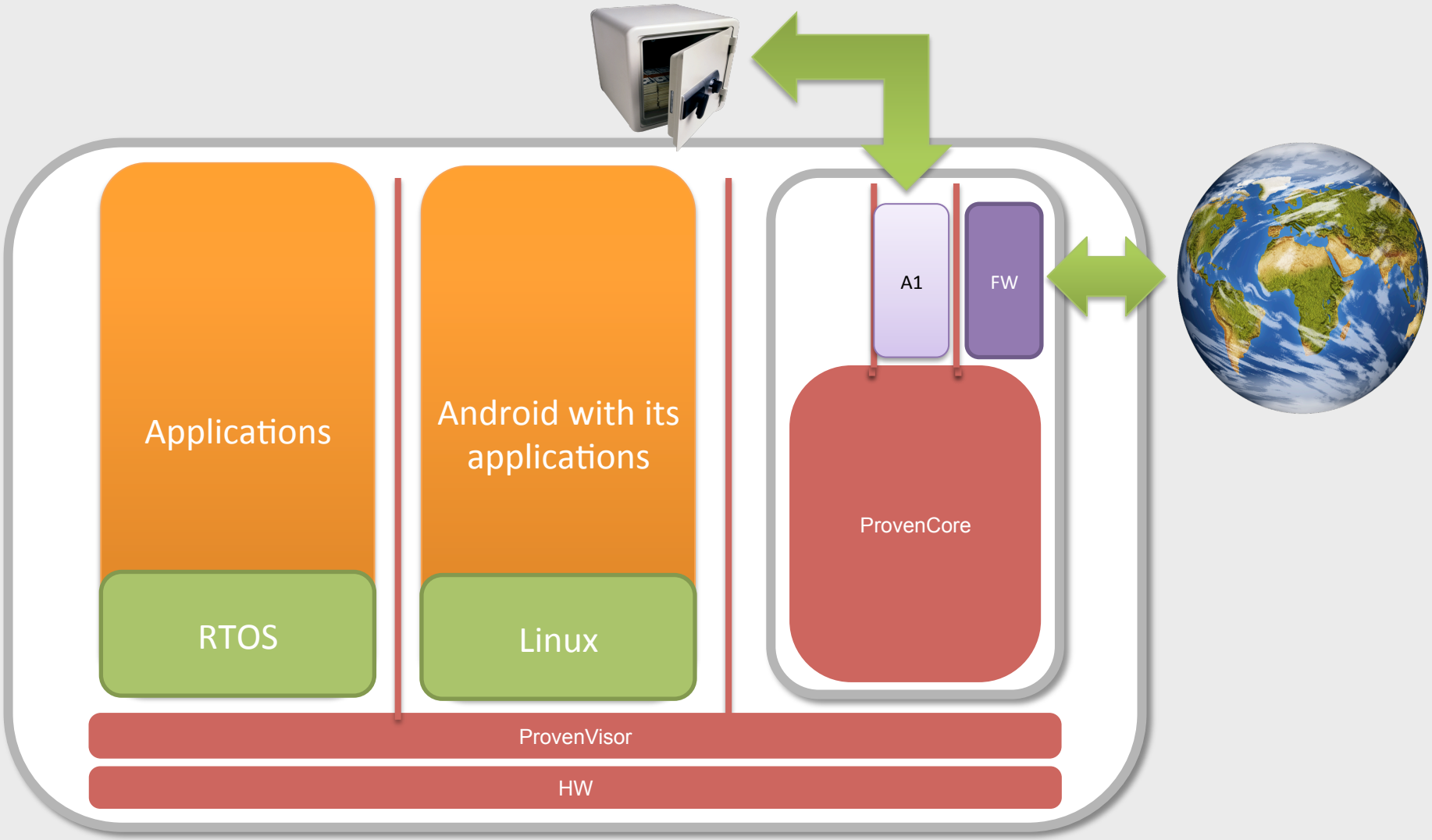
RTOS with its applications

HW

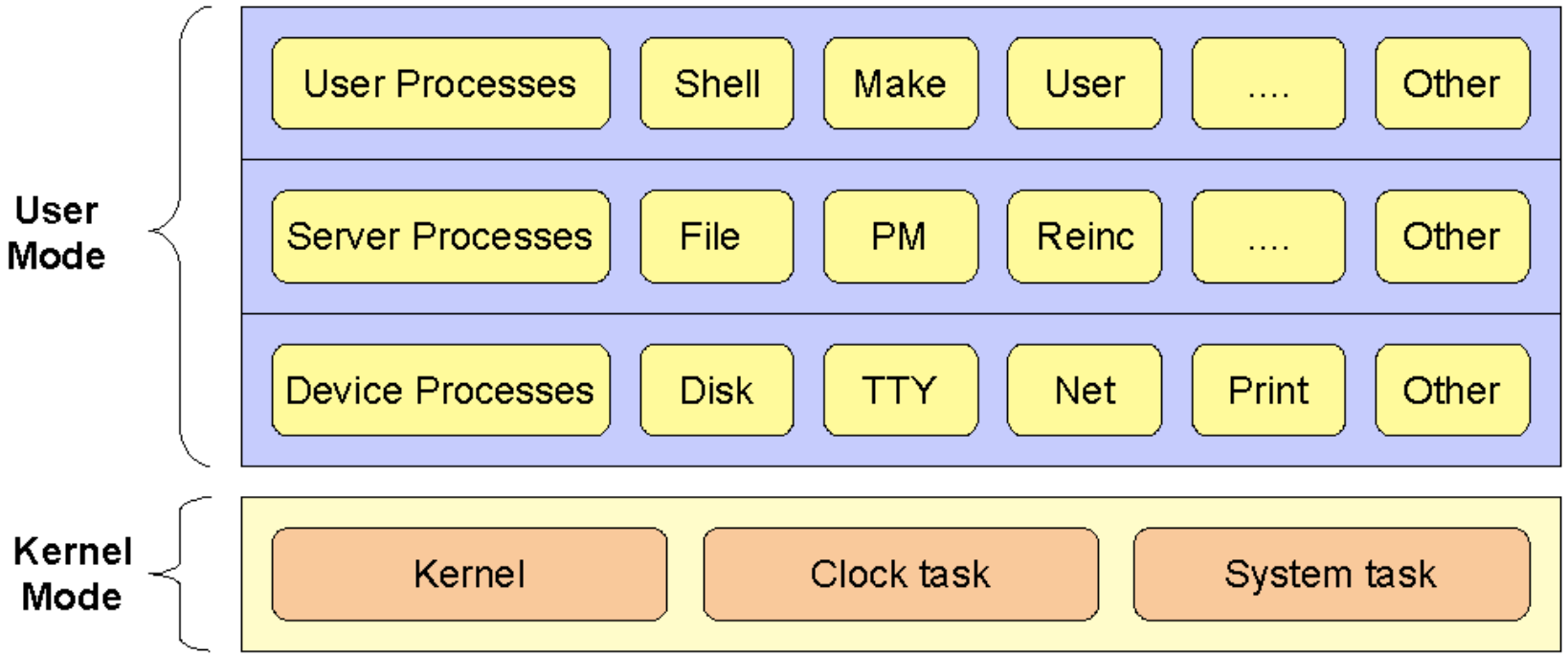
Android (or any rich OS) with its applications

HW

Use Cases


Applications
RTOS
Android with its applications
Linux
A1
FW
ProvenCore
ProvenVisor
HW

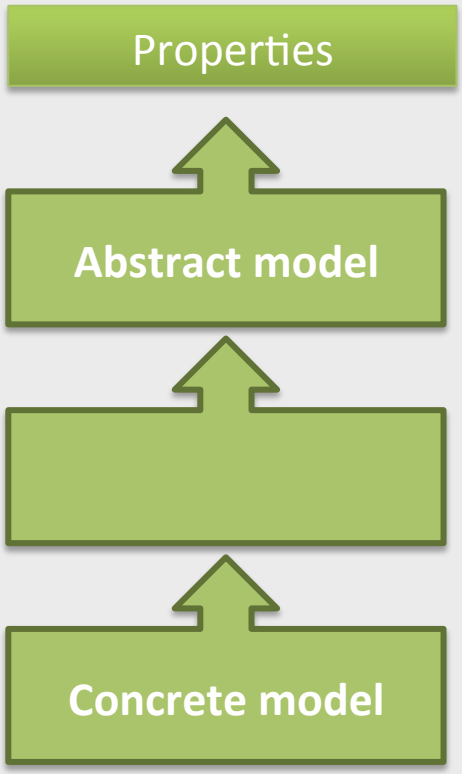
Microkernel modelling



The MINIX 3 Microkernel Architecture

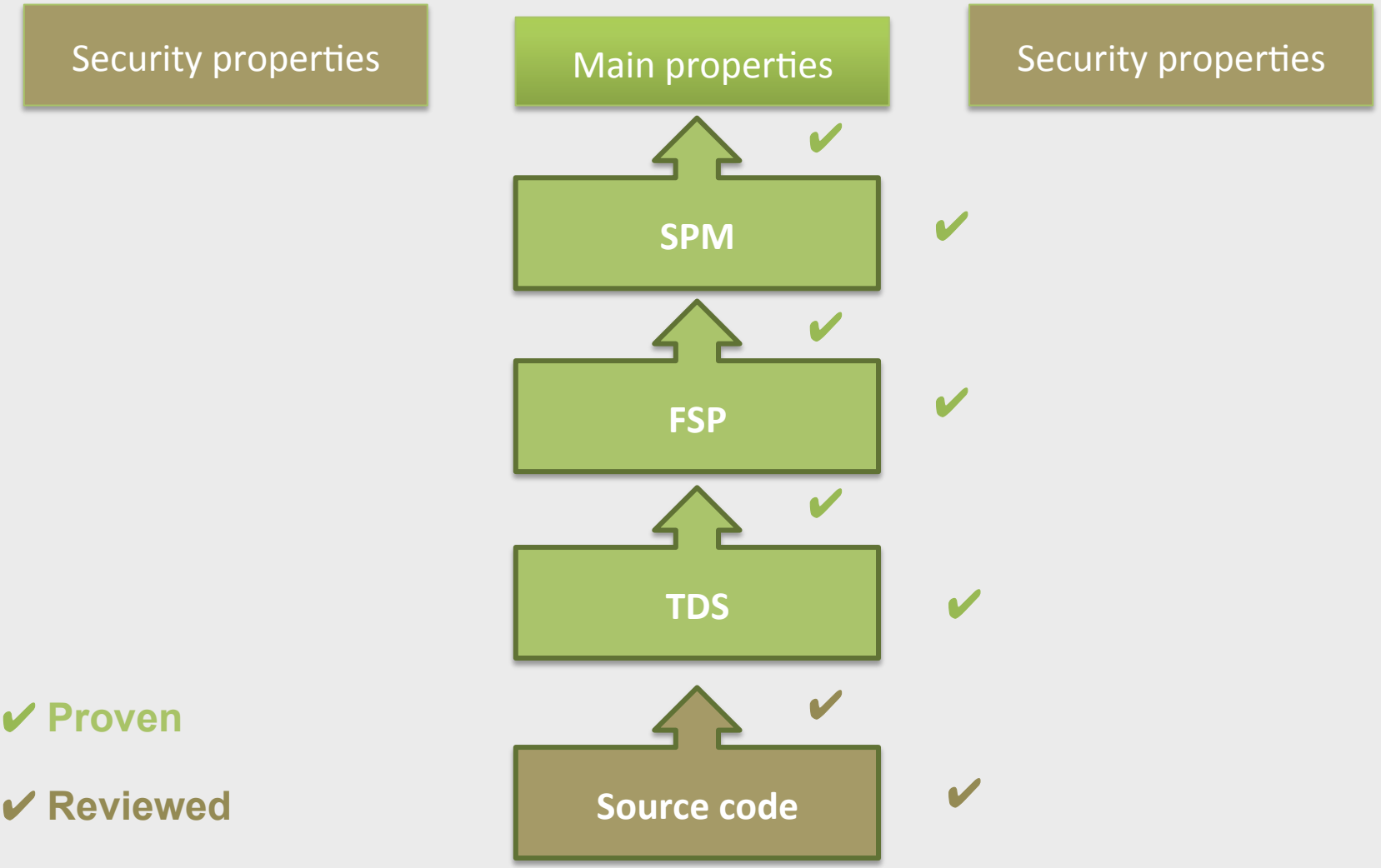
(**Stéphane Lescuyer – Vincent Siles – Benoit Montagu**)
Towards a Verified Isolation microkernel – Stéphane Lescuyer HIPEAC 01/2015

Modelling a microkernel: Global approach

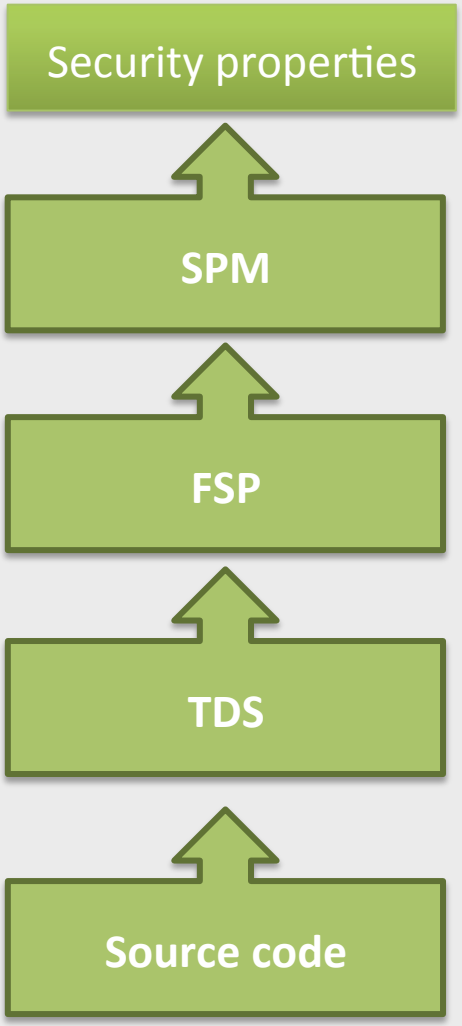


- An abstract layer recreating the behaviors of more concrete layers
- Formal properties expressed at the highest level
- Properties are more natural and simpler to understand

Modelling a microkernel: Links with security schemes



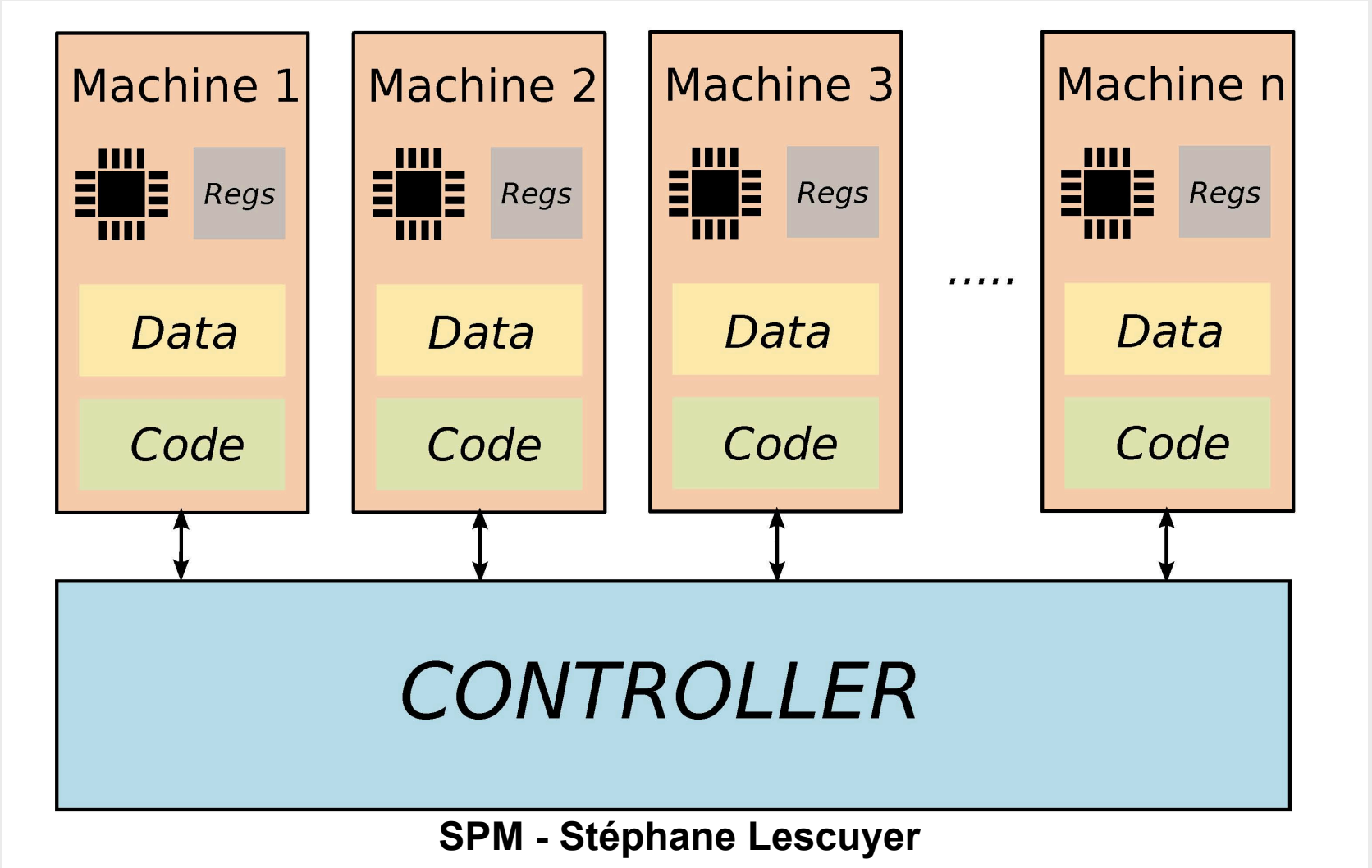
Modelling a microkernel: Proving the source code



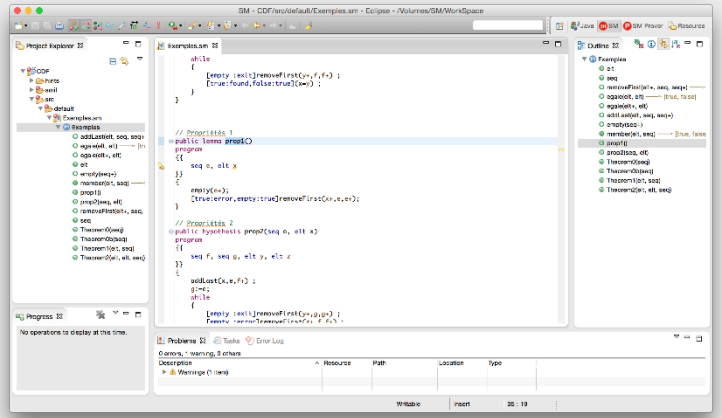
Properties and Abstract Model

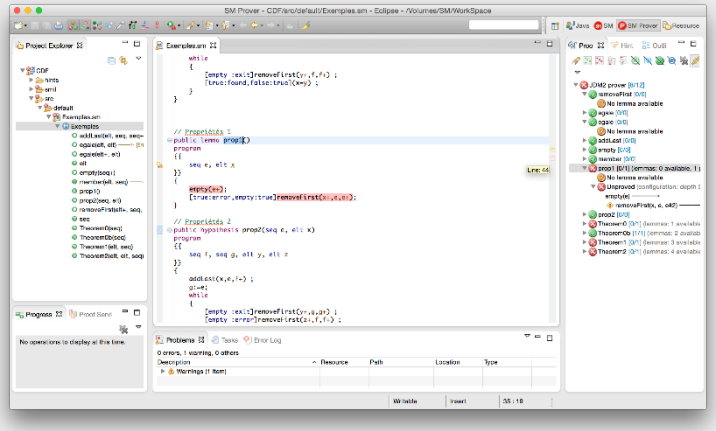
- Model must be as abstract as possible while capturing the desired property
- Paradigm: independent devices, each one using their own resources (code, data, memory, etc), while potentially communicating and/or sharing some resources such as memory pages, file systems, etc.

Separation


Machine 1
Regs
Data
Code
Machine 2
Regs
Data
Code
Machine 3
Regs
Data
Code
.....
Machine n
Regs
Data
Code
CONTROLLER
SPM - Stéphane Lescuyer

SMART development toolchain





Development environment: Eclipse plugin

Prover: Eclipse plugin

P&R Intermediate Language: SMIL

Generator (source code and documentation): Eclipse plugin

Source Code

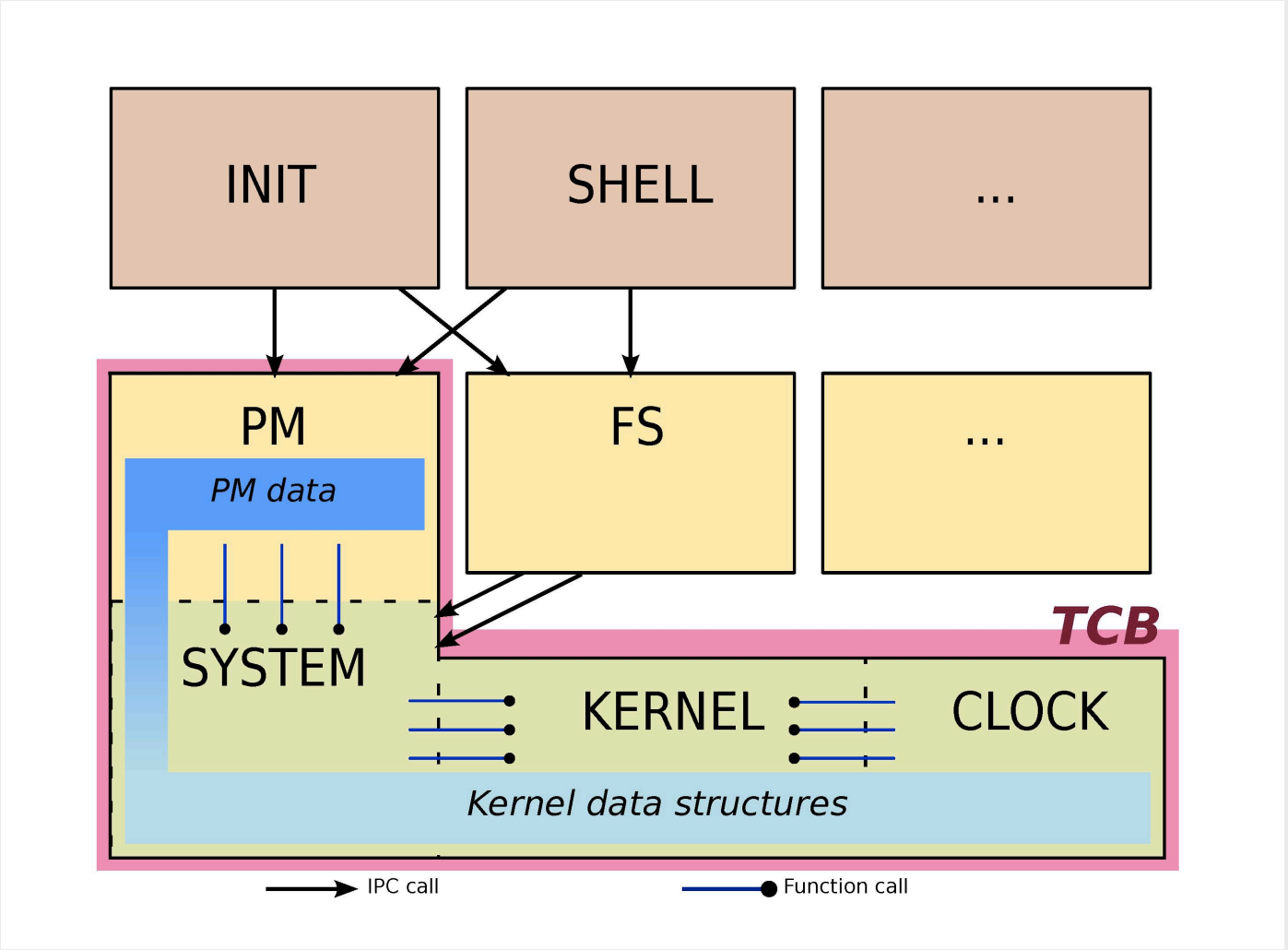
- Compilable
- C, Java, etc.

Certification Documentation

- CC
- DO-178
- etc.

Automated

TCB security and identification



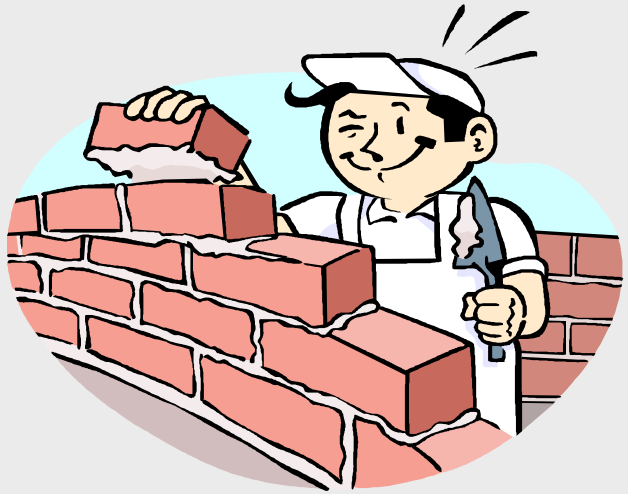
Requirements identification process - Applied strategy (1/2)

- Gathering experience and knowledge, step by step
  - Using many formal approaches on real life use cases over many years
  - Each time in a context where justifying applicability and usefulness of the project was mandatory

- Strategy (1/2):
  - Choose the cases where the benefit/cost ratio is favorable and the market is representative
  - Cost reduction (microkernel, etc.)
    - Identify or improve the TCB's definition
    - Reuse benefits
  - Facilitate maintainability
  - Make Formal Methods easier to use in order to allow software developers to use them by themselves

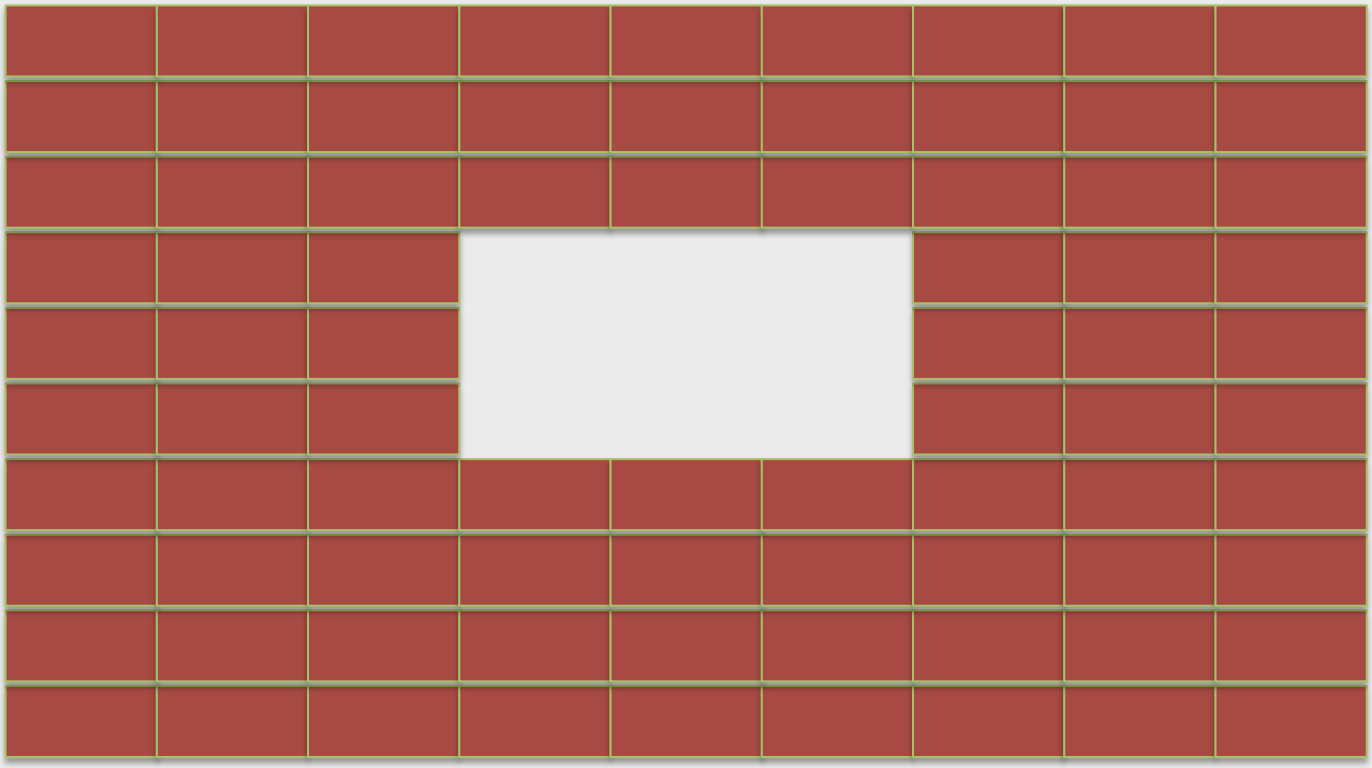
Components

- Building complex systems by composing a small number of types of components is essential for any engineering discipline.
- This confers numerous advantages such as mastering complexity, enhanced productivity and correctness through reuse
- Component composition orchestrates interactions between components. It lies at the heart of the system integration challenge.



Joseph Sifakis – Turing Award – Gérard Berry Seminar – 4th of March 2015

Composition



Requirements identification process - Applied strategy (1/2)

- Gathering experience and knowledge, step by step
  - Using many formal approaches on real life use cases over many years
  - Each time in a context where justifying applicability and usefulness of the project was mandatory

- Strategy (1/2):
  - Choose the cases where the benefit/cost ratio is favorable and the market is representative
  - Cost reduction (microkernel, etc.)
    - Identify or improve the TCB's definition
    - Reuse benefits
  - Facilitate maintainability
  - Make Formal Methods easier to use in order to allow software developers to use them by themselves

Requirements identification process
Applied strategy (2/2)

- **Strategy (2/2):**
  - **Maximize benefits by targeting areas where reliability is key**
    - Mobile security
    - Aeronautics
    - Automobile (increasing role of computers, connected cars, driverless cars)
    - Smart Grids,
    - Industry 4.0
    - Home automation
    - Office management
    - Medical systems
    - etc.
  - **Enable certification**

Conclusions

- A very limited number of proven COTS can make it possible to increase the security level in a very significant way,
- Everything can't be modelled nor proven (hypotheses, resistance to physical attacks, properties appropriateness, unsuitable architectures, human chain, etc.) but it doesn't mean that Formal Methods is not THE right answer to the security and trust challenges of emerging open architectures

THANK YOU FOR YOUR TIME

QUESTIONS?

Prove & Run S.A.S.

dominique.bolignano@provenrun.com

77, avenue Niel, 75017 Paris, FRANCE