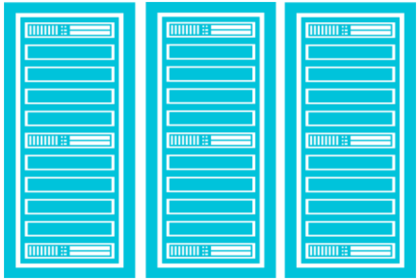
Designing security into low cost IoT systems

ARM

Jim Wallace
Director, SSG Marketing

Linaro Connect, Bangkok 2016
8th March 2016

From Sensors to Servers



LITE
IoT/EMBEDDED

ARM® mbed™
IoT Device Platform

ARM TRUSTZONE
System Security

Productivity

Connectivity

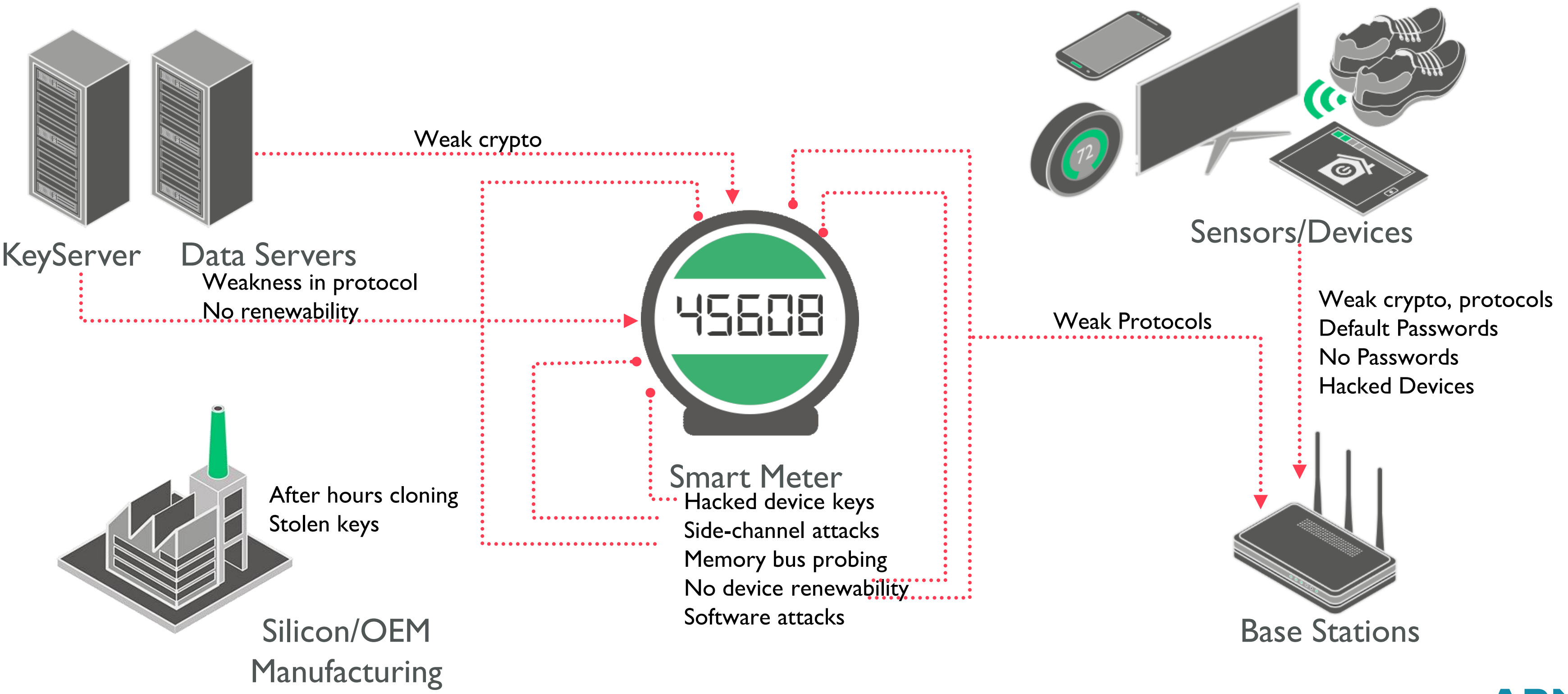
Security

Management

Efficiency

IoT is going everywhere

Risks are hard to predict



Weak crypto

KeyServer

Data Servers

Weakness in protocol
No renewability

Sensors/Devices

Weak crypto, protocols
Default Passwords
No Passwords
Hacked Devices

Weak Protocols

Smart Meter

Hacked device keys
Side-channel attacks
Memory bus probing
No device renewability
Software attacks

After hours cloning
Stolen keys

Silicon/OEM Manufacturing

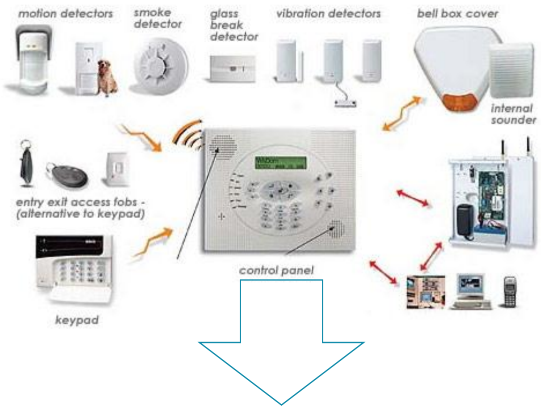
Base Stations

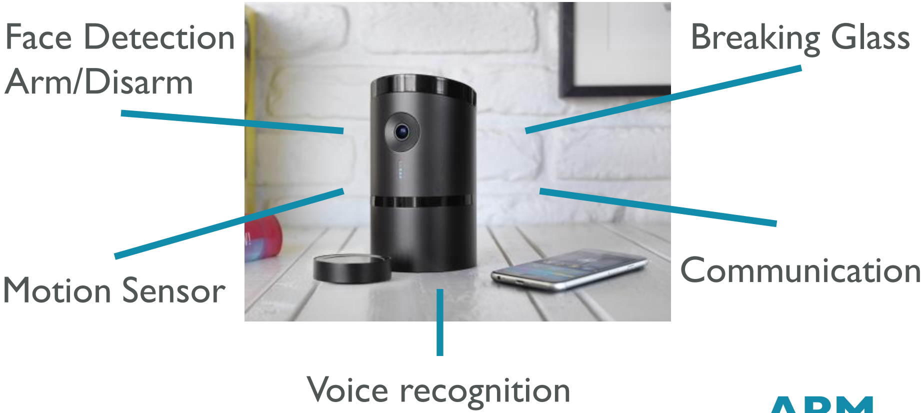
IoT - From Cortex-M to Cortex-A class devices

	Ultra-low cost	Low cost	Generic
	BBC micro:bit BT Smart beacon	Rich BT Smart Thread node	WiFi node Gateway
Intelligent Connected **Device HW Resources**	ARMv6-M ARMv8-M Baseline	ARMv8-M Mainline ARMv7-M with MPU TRNG + Crypto	Cortex-A Class TRNG + Crypto + GPU + VPU
Secure **Device SW Capabilities**	BT Smart	IP + TLS mbed OS uVisor Management Security Firmware OTA	IP + TLS OP-TEE Management Security Firmware over-the-air Rich UI/Multimedia
	mbed OS / RTOS	mbed OS / RTOS	Linux / Rich OS

Evolution of IoT driving need for generic devices

- Local intelligence enables: Camera/microphone/other sensors
  - Raw data does not need to be sent to the cloud, only processed meta-data is being sent
  - Reduced data bandwidth, transfer overhead and processing latency to/from cloud
  - Increased security





Security in IoT end points

Device security

- Device integrity
  - Protect from untrusted S/W
  - Allow recovery from attack
- Asset protection
  - Prevent access to certain resources
- Data security
  - Keep data confidential
  - Prevent data alteration
- Physical Security
  - Anti-tampering

Communications security

- Link encryption
  - Prevent eavesdroppers listening
- Authentication
  - Identity of endpoint / server

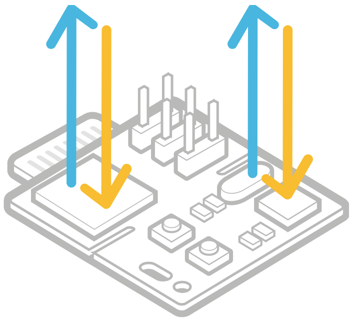
Management security

- Device management
  - Support for bootstrapping / provisioning / Behaviour monitoring…
- Keep firmware up-to-date

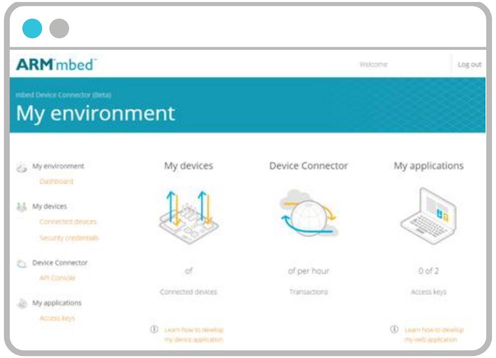
Security must be built into all stages of the system

Management security: mbed Device Connector

- mbed Device Connector eases development, management and scaling of IoT
- Available at https://connector.mbed.com
- Management security implemented via standards such as OMA LWM2M



Build IoT Device



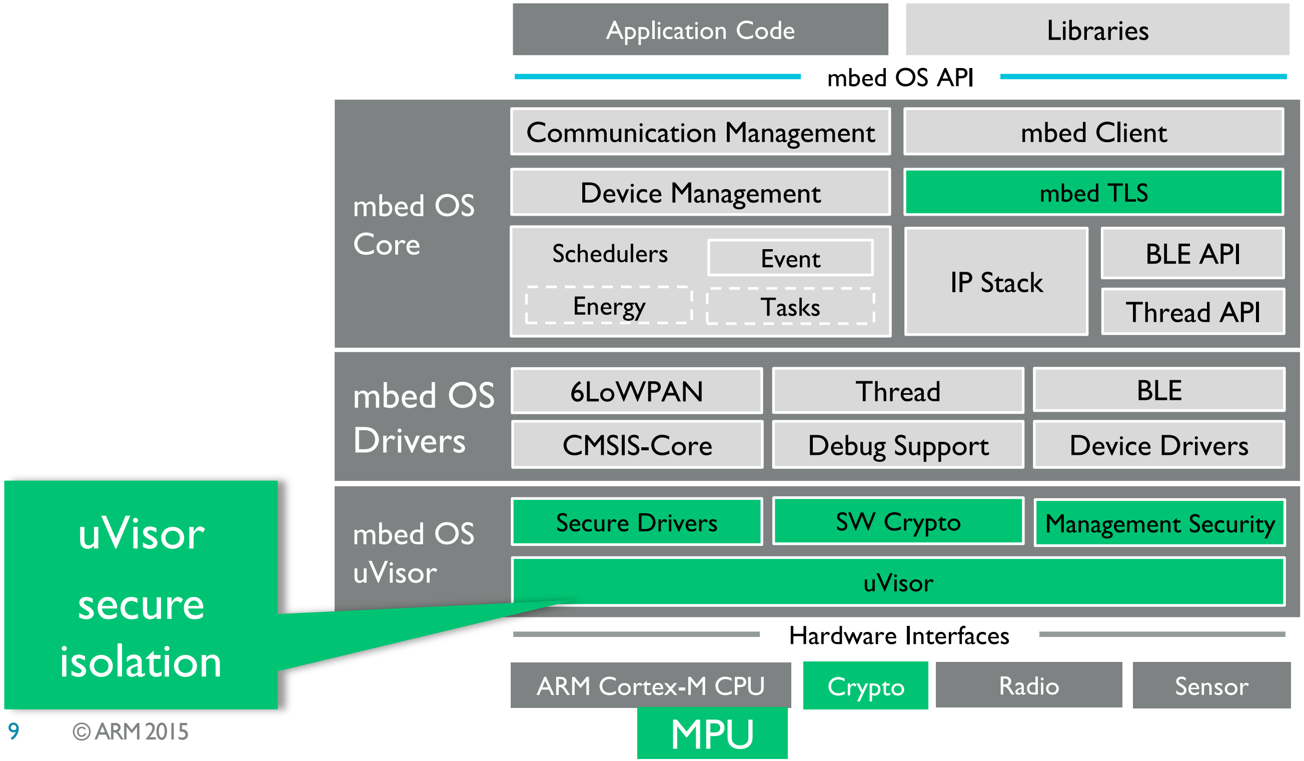
Connect your devices

Build application with example code

Utilize cloud solutions

mbed OS 15.11

- mbed OS is a modular, secure, efficient, open source OS for IoT
- Connects to mbed Device Connector



Application Code | Libraries

mbed OS API

mbed OS Core
- Communication Management
- mbed Client
- Device Management
- mbed TLS
- Schedulers: Event, Energy, Tasks
- IP Stack
- BLE API
- Thread API

mbed OS Drivers
- 6LoWPAN
- Thread
- BLE
- CMSIS-Core
- Debug Support
- Device Drivers

mbed OS uVisor
- Secure Drivers
- SW Crypto
- Management Security
- uVisor

Hardware Interfaces

ARM Cortex-M CPU | Crypto | Radio | Sensor

MPU

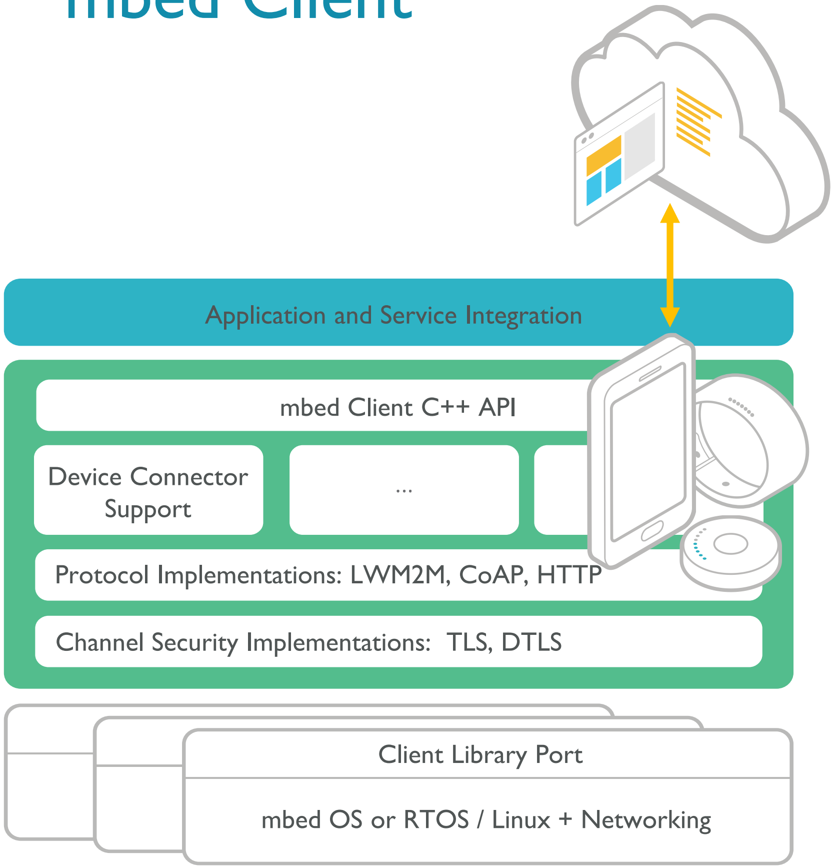
uVisor secure isolation

Communication Security

Management Security

Device Security

mbed Client

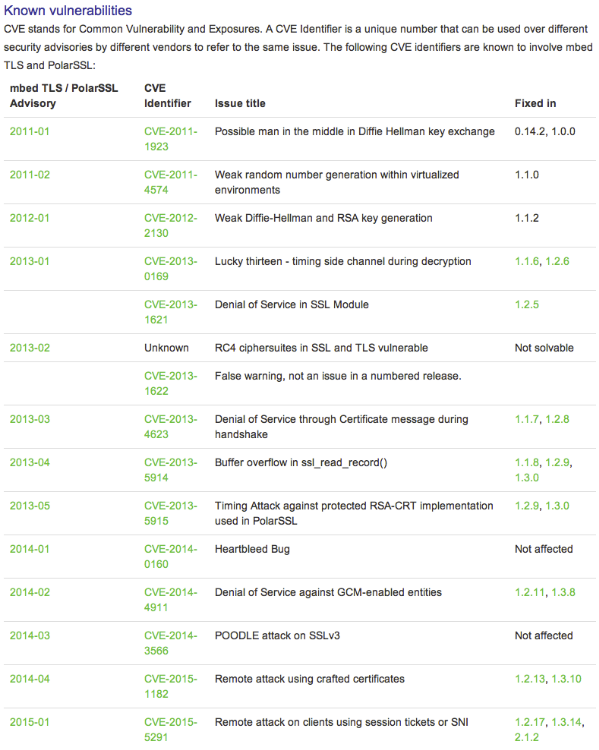

Application and Service Integration
mbed Client C++ API
Device Connector Support
…
Protocol Implementations: LWM2M, CoAP, HTTP
Channel Security Implementations: TLS, DTLS
Client Library Port
mbed OS or RTOS / Linux + Networking

- Connects to mbed Device Connector
- Included as part of mbed OS, also portable to other platforms including Linux and third party RTOS
- Implements protocols and support for securely publishing resources (e.g. sensor data), and managing the device from the cloud

Communication security: mbed TLS

- Fully-fledged SSL / TLS / DTLS Library
- Developer friendly: Clean API and documentation
- Open Source under Apache 2.0 license at https://tls.mbed.org/
- Suitable for use on Cortex-M and Cortex-A processors based targets

Transport Security	Symmetric Encryption	Public Key Algorithms	Hash Algorithms	Random Number Generation	X.509 Certificate Handling
TLS/DTLS, etc	AES, etc	ECDHE, ECDSA, etc	SHA, etc	Entropy pool, CTR_DEBUG, etc	✓



Known vulnerabilities

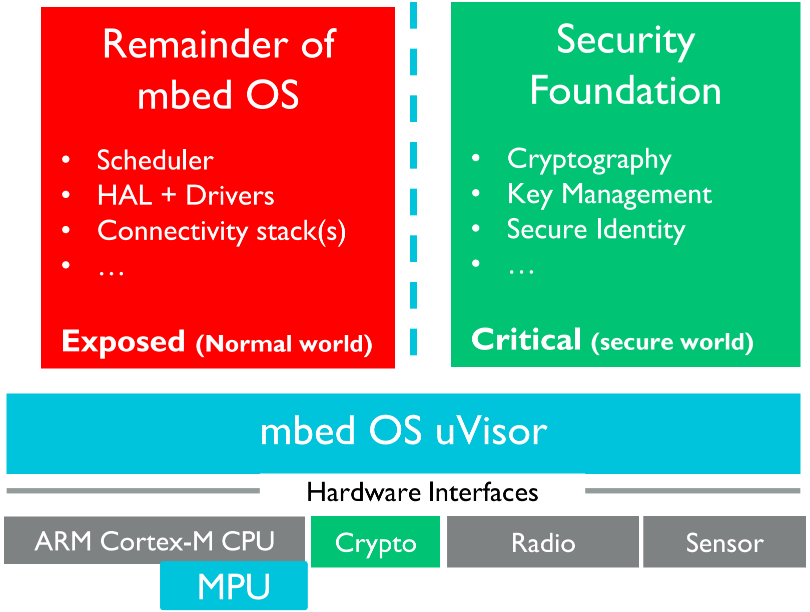
CVE stands for Common Vulnerability and Exposures. A CVE Identifier is a unique number that can be used over different security advisories by different vendors to refer to the same issue. The following CVE identifiers are known to involve mbed TLS and PolarSSL:

mbed TLS / PolarSSL Advisory	CVE Identifier	Issue title	Fixed in
2011-01	CVE-2011-1923	Possible man in the middle in Diffie Hellman key exchange	0.14.2, 1.0.0
2011-02	CVE-2011-4574	Weak random number generation within virtualized environments	1.1.0
2012-01	CVE-2012-2130	Weak Diffie-Hellman and RSA key generation	1.1.2
2013-01	CVE-2013-0169	Lucky thirteen - timing side channel during decryption	1.1.6, 1.2.6
	CVE-2013-1621	Denial of Service in SSL Module	1.2.5
2013-02	Unknown	RC4 ciphersuites in SSL and TLS vulnerable	Not solvable
	CVE-2013-1622	False warning, not an issue in a numbered release.	
2013-03	CVE-2013-4623	Denial of Service through Certificate message during handshake	1.1.7, 1.2.8
2013-04	CVE-2013-5914	Buffer overflow in ssl_read_record()	1.1.8, 1.2.9, 1.3.0
2013-05	CVE-2013-5915	Timing Attack against protected RSA-CRT implementation used in PolarSSL	1.2.9, 1.3.0
2014-01	CVE-2014-0160	Heartbleed Bug	Not affected
2014-02	CVE-2014-4911	Denial of Service against GCM-enabled entities	1.2.11, 1.3.8
2014-03	CVE-2014-3566	POODLE attack on SSLv3	Not affected
2014-04	CVE-2015-1182	Remote attack using crafted certificates	1.2.13, 1.3.10
2015-01	CVE-2015-5291	Remote attack on clients using session tickets or SNI	1.2.17, 1.3.14, 2.1.2

https://tls.mbed.org/security

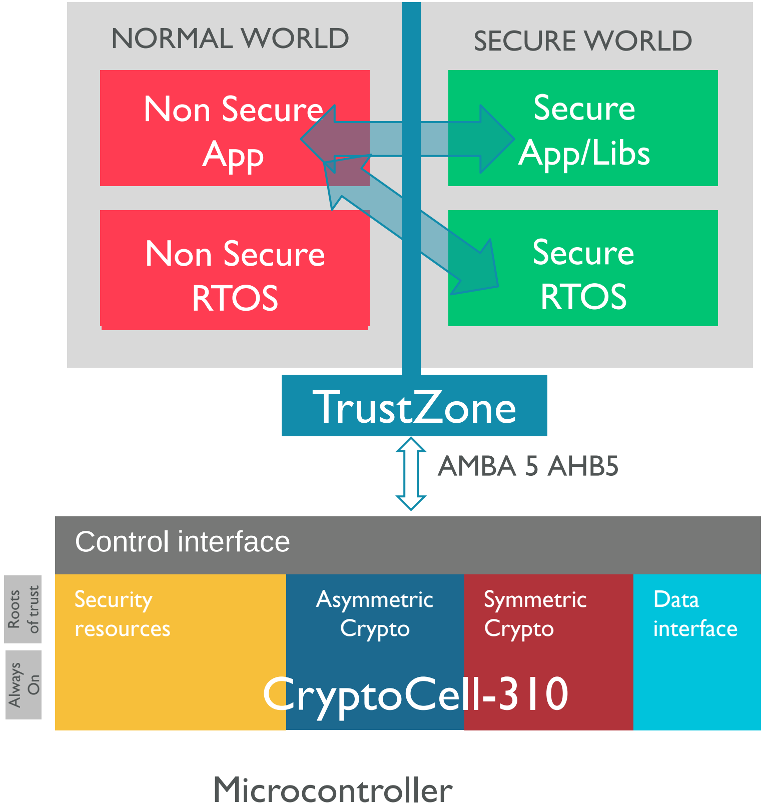
Device security services in low cost devices

- Existing IoT solutions use flat address spaces with little privilege separation
  - Especially on microcontrollers
- Mitigating strategy to split security domains into
  - Exposed code
  - Protected critical code

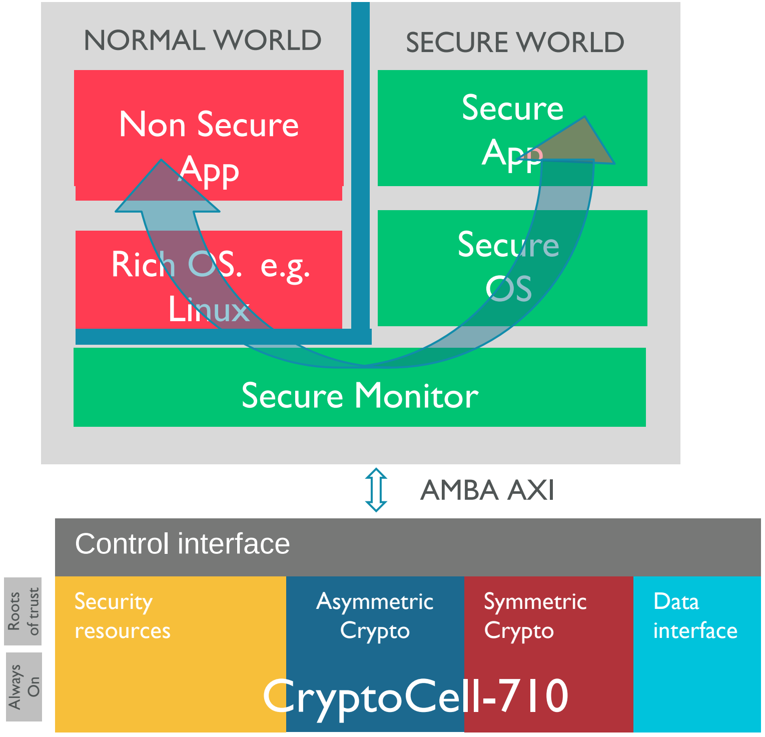


TrustZone for low cost ARMv8-M IoT platforms

- The ARMv8-M architecture introduces secure and non-secure code execution
  - Code running in non-secure memory can only access non-secure devices and memory
  - Code running in secure memory can access whole address space
- So low cost devices can
  - Have trusted code & Apps in secure memory
  - Can have non trusted applications installed in non secure memory safe in the knowledge that they cannot be used to attack the system
- CryptoCell augments TrustZone
  - Providing a range of security subsystems for acceleration and offloading



TrustZone technology for every IoT platform



NORMAL WORLD

SECURE WORLD

Non Secure App

Rich OS. e.g. Linux

Secure App

Secure OS

Secure Monitor

AMBA AXI

Control interface

Roots of trust

Always On

Security resources

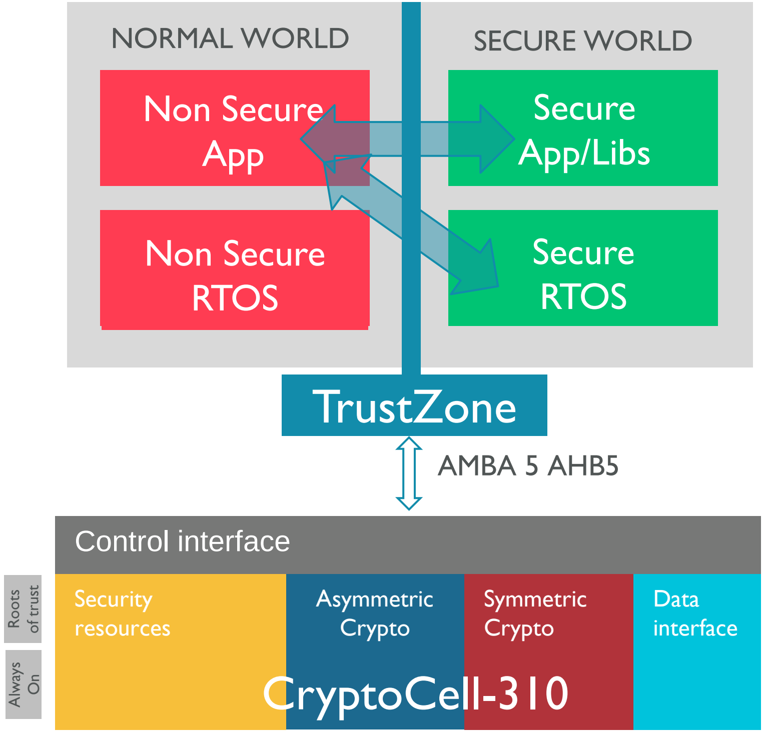
Asymmetric Crypto

Symmetric Crypto

Data interface

CryptoCell-710

Apps Processor



NORMAL WORLD

SECURE WORLD

Non Secure App

Non Secure RTOS

Secure App/Libs

Secure RTOS

TrustZone

AMBA 5 AHB5

Control interface

Roots of trust

Always On

Security resources

Asymmetric Crypto

Symmetric Crypto

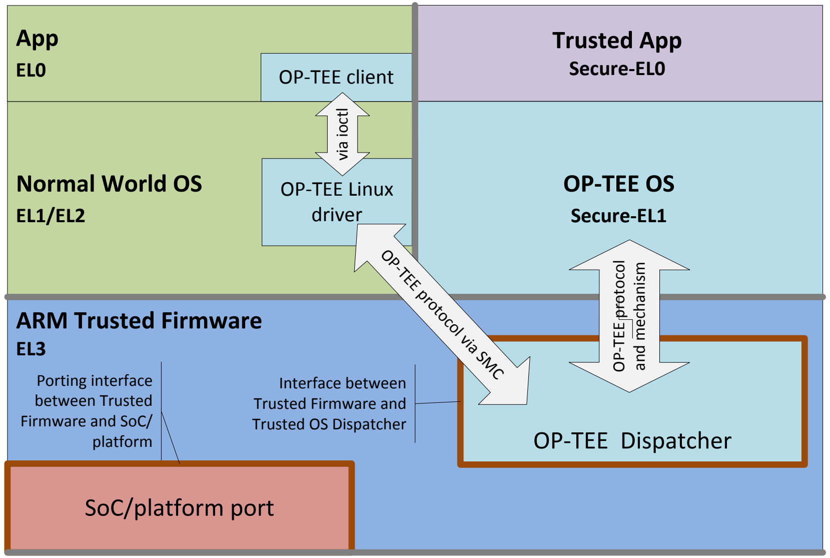
Data interface

CryptoCell-310

Microcontroller

Trusted Firmware, OP-TEE reduce fragmentation

- Secure World foundations for ARMv8-A:
  - Trusted Board Boot
  - Secure World runtime – world switch, interrupt routing, PSCI, SMC handling
  - Open source projects on GitHub https://github.com/ARM-software/arm-trusted-firmware
    https://github.com/OP-TEE
- v1.2 (December)
  - + Trusted Boot baseline features
  - + PSCI v1.0 key optional features
  - + OS vendor alignment
  - GICv3 drivers



App
EL0

OP-TEE client

via ioctl

Normal World OS
EL1/EL2

OP-TEE Linux driver

Trusted App
Secure-EL0

OP-TEE OS
Secure-EL1

OP-TEE protocol and mechanism

OP-TEE protocol via SMC

ARM Trusted Firmware
EL3

Porting interface between Trusted Firmware and SoC/ platform

Interface between Trusted Firmware and Trusted OS Dispatcher

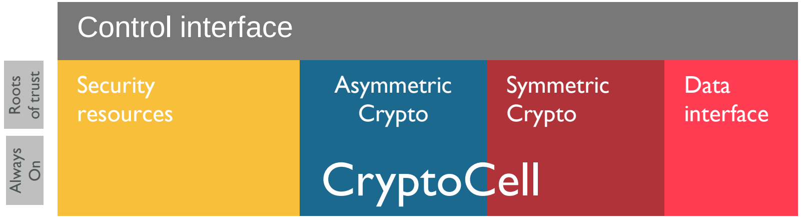
OP-TEE Dispatcher

SoC/platform port

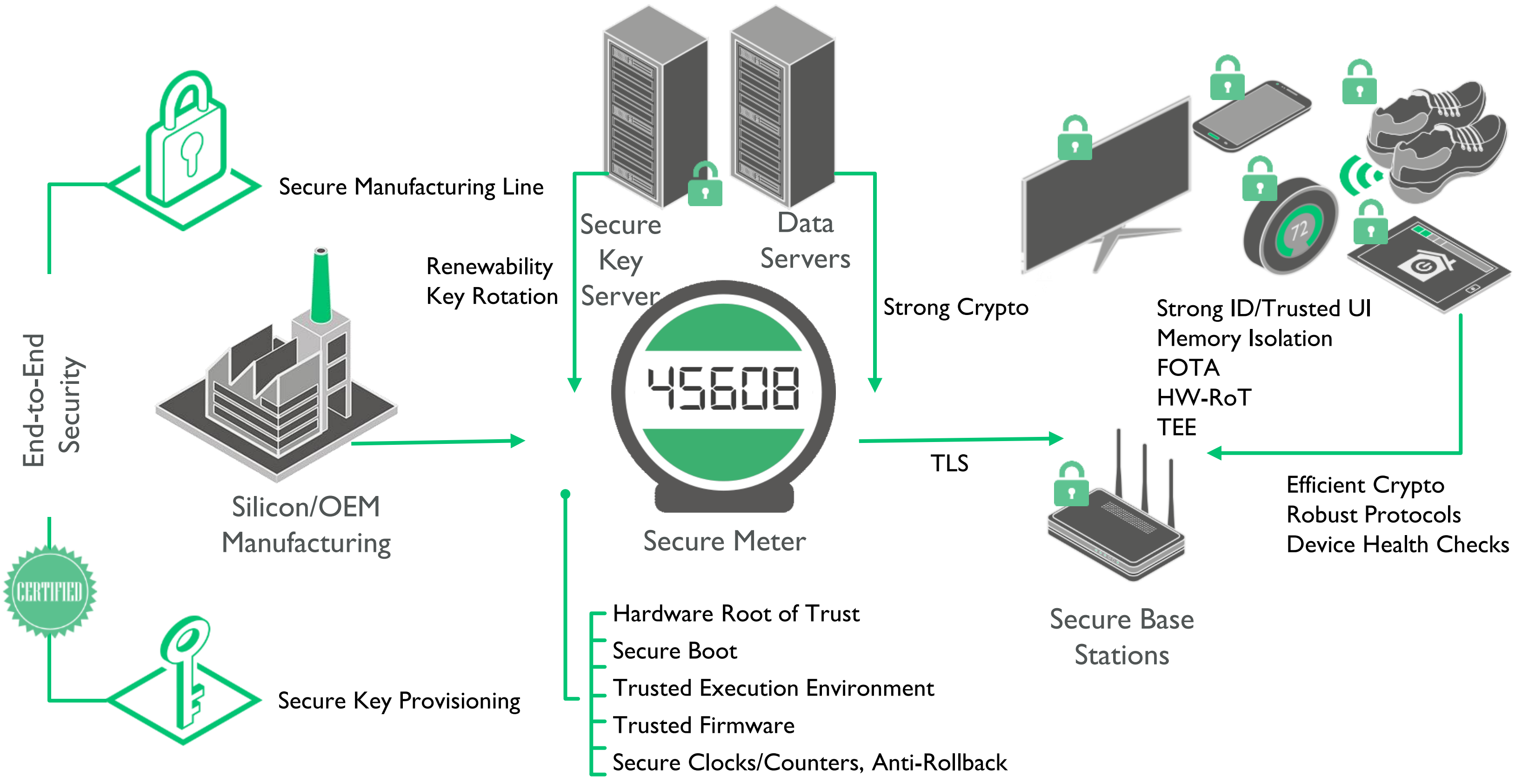
SoC supplier	ARM Trusted Firmware	Trusted App supplier
OS/hypervisor supplier	Trusted OS supplier	Internal TOS interface

ARM TrustZone CryptoCell

- TrustZone, TEE and CryptoCell provide platform level security
  - with a hardware Root of Trust / Trust Anchor for the system
    - Crypto acceleration
    - TRNG
- Configurable to target application – right size
- Enhances usability e.g. time for DTLS handshake & door lock to open
- Simplifies security implementations



LITE using this to enable a security foundation



End-to-End Security

Secure Manufacturing Line

Silicon/OEM Manufacturing

Secure Key Provisioning

Renewability
Key Rotation

Secure Key Server

Data Servers

Secure Meter

- Hardware Root of Trust
- Secure Boot
- Trusted Execution Environment
- Trusted Firmware
- Secure Clocks/Counters, Anti-Rollback
- Secure Key Storage, Robust Crypto

Strong Crypto

TLS

Strong ID/Trusted UI
Memory Isolation
FOTA
HW-RoT
TEE

Efficient Crypto
Robust Protocols
Device Health Checks

Secure Base Stations

Imagine a world where…

- From the wide choice of ARM-based devices, you chose the perfect one for you
  - Price, performance, power, form, security etc.

- And what software you ran on it was up to you…
  - Android / Brillo, BSD, CentOS, ChromeOS, RHEL, SUSE, Tizen, Snappy Ubuntu, Windows, Yocto/OE, etc …or something we haven't even thought of yet

- But once you made that choice, it should all just work!

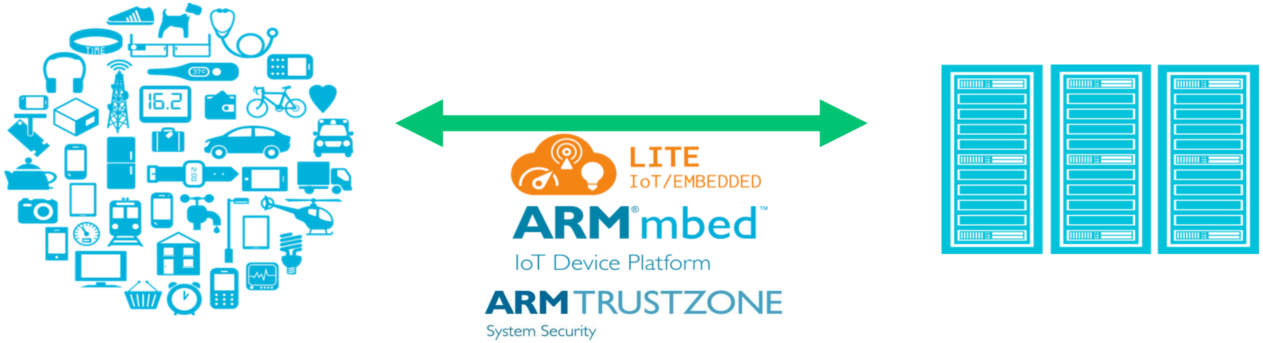
- ARM & Linaro are committed to making this happen

Linaro and ARM providing the foundation for IoT

- ARM working with Linaro to provide an end-to-end open source IoT framework for specific IoT implementations

- ARM part of LITE WG
  - “Place to collaborate on ARM architecture for IoT”, enabling

- Software solutions from Cortex-M to Cortex-A based platforms



THANK YOU!

ARM

?