Power Matters.

SmartFusion2 – Embedded system "Root-of-Trust"

Secure Boot Demonstration

Peter Trott
Snr FAE - Microsemi

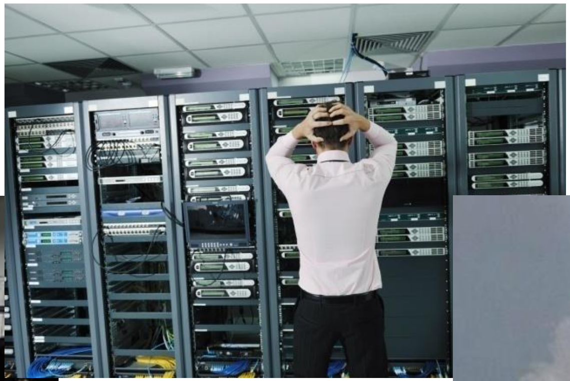
Agenda

- Embedded System Threats
- Secure Boot / Root-of-Trust (RoT)
- Multi-Stage CPU Boot Process
- Non-Secure Processor Problem
- Microsemi Secure Boot Solution
- Secure Boot Demo

Embedded Systems Threats

- We are under attack!
- Hacking of embedded microprocessor based systems is on the rise
  - Easy to obtain embedded products
  - Readily available tools / techniques on hacker websites
- Improved security measures required to counter these increasing sophisticated attacks.

```
A problem has been detected and windows has been shut down to prevent damage
to your computer.

An initialization failure occurred while attempting to boot from the network.

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x000000BB (0x00000003,0xC0000001,0x00000000,0x
```



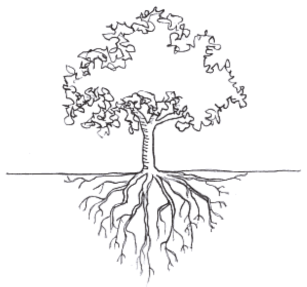
Secure Boot

- Cornerstone of embedded system security is to ensure that only <u>authorized</u> microprocessor code is loaded and executed.
- The process of guaranteeing the execution of authentic code is called ‘Secure Boot’
- Booting processors from known trusted code
  - Prerequisite for the secure operation of embedded systems
- If the initial hardware and boot code can be trusted, then this trust can be extended to code loaded and executed later.
- The foundation for Secure Boot is the Root-of-Trust (RoT)

Root-of-Trust (RoT)

- A root-of-trust is essential to implement secure boot

- Root-of-trust is an entity that can be trusted to behave in an expected manner
  - Supports verification of system, software and data integrity
  - Keeps keys and critical data confidential
  - Its processes are immutable
  - Works in conjunction with other system elements to ensure system security



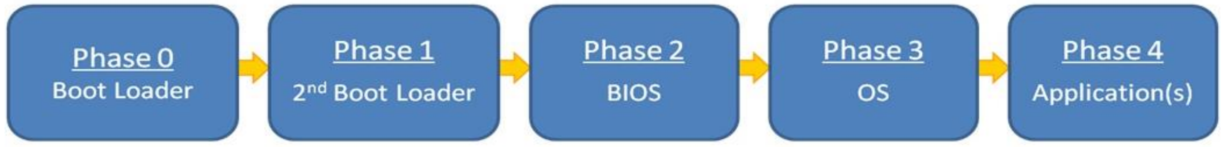
A Root-of-Trust is the foundation upon which all other security layers are created and trusted.

Root-of-Trust Examples

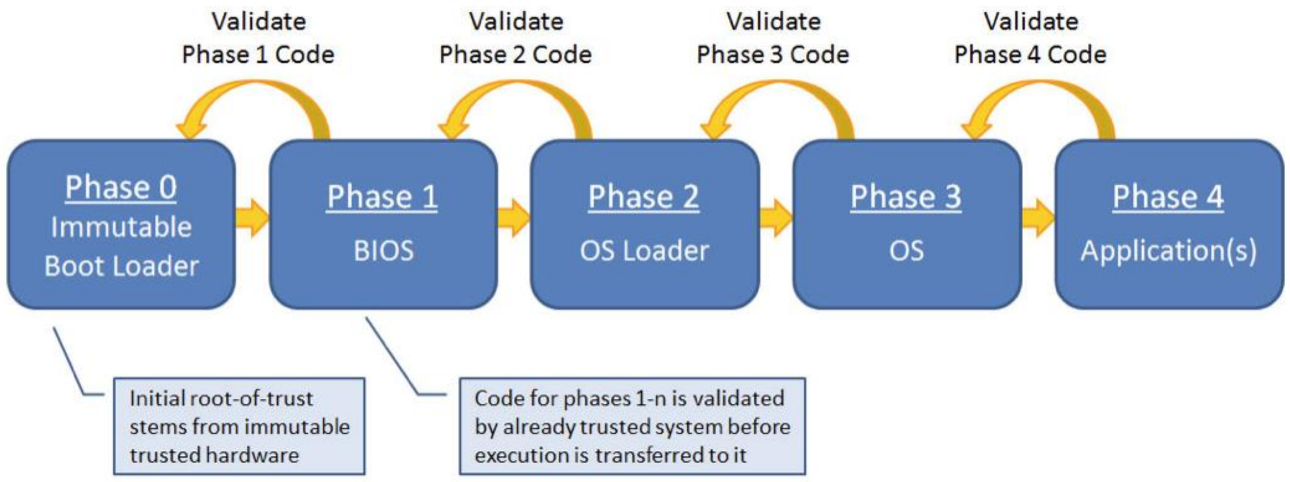
- Trusted Platform Module (TPM)
  - Industry standard external Root-of-Trust device
  - Provides cryptographic services with static RSA key
- Processors with built in security features
  - One time programmable key storage
  - Immutable on-chip microcode
  - May not support the full suite of security functions needed
    - Anti-tamper, data encryption at rest, DPA resistance, ROM scrambling, etc
- Select SoC FPGAs (SmartFusion2 / Igloo2)
  - Cryptographic services, True random number generator
  - Stronger design security, On-chip oscillators
  - Anti-tamper measures (DPA resistance, Zeroization)
  - On-chip Flash
  - Vastly better computational power, more IOs and a variety of built-in interfaces

CPU Multi-Stage Boot Process

- Typical Multistage Boot (unsecure)
  - From boot to app its non stop execution!

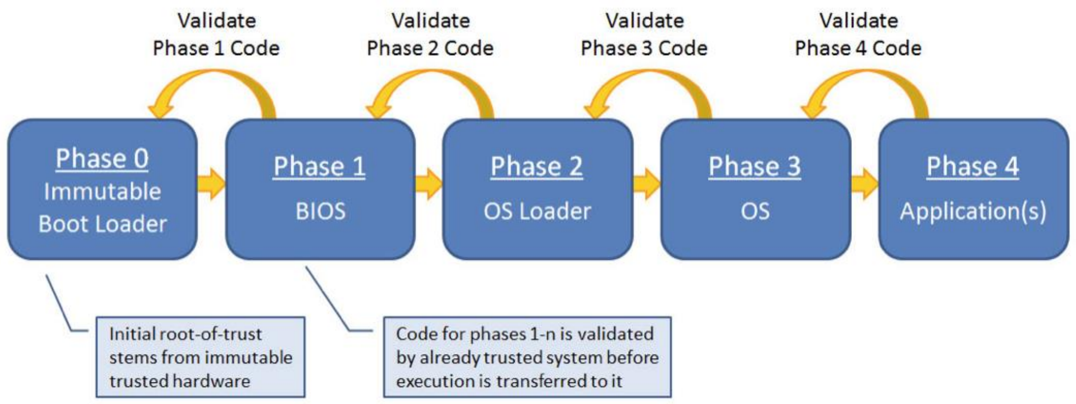


- Secure Multistage Boot
  - Initialize an embedded system from rest with trusted code
  - Validation of each stage performed by the prior stage
    - Establishes a 'chain of trust' all they way to the top application layer



Secure Multi-Stage Boot Code Validation

- Code validation can be done by symmetric or asymmetric key cryptography techniques
  - Build an inherently trusted key (public) into the phase-0 boot loader
  - Phase-1 code is digitally signed using key (private)
  - During Phase-0 the system validates the digital signature of the Phase-1 code, prior to execution
    - Boot if signature validates
    - Abort boot apply system penalties if invalid
- Immutability of the Phase-0 Key is critically important
  - Anti-Tamper Monitors

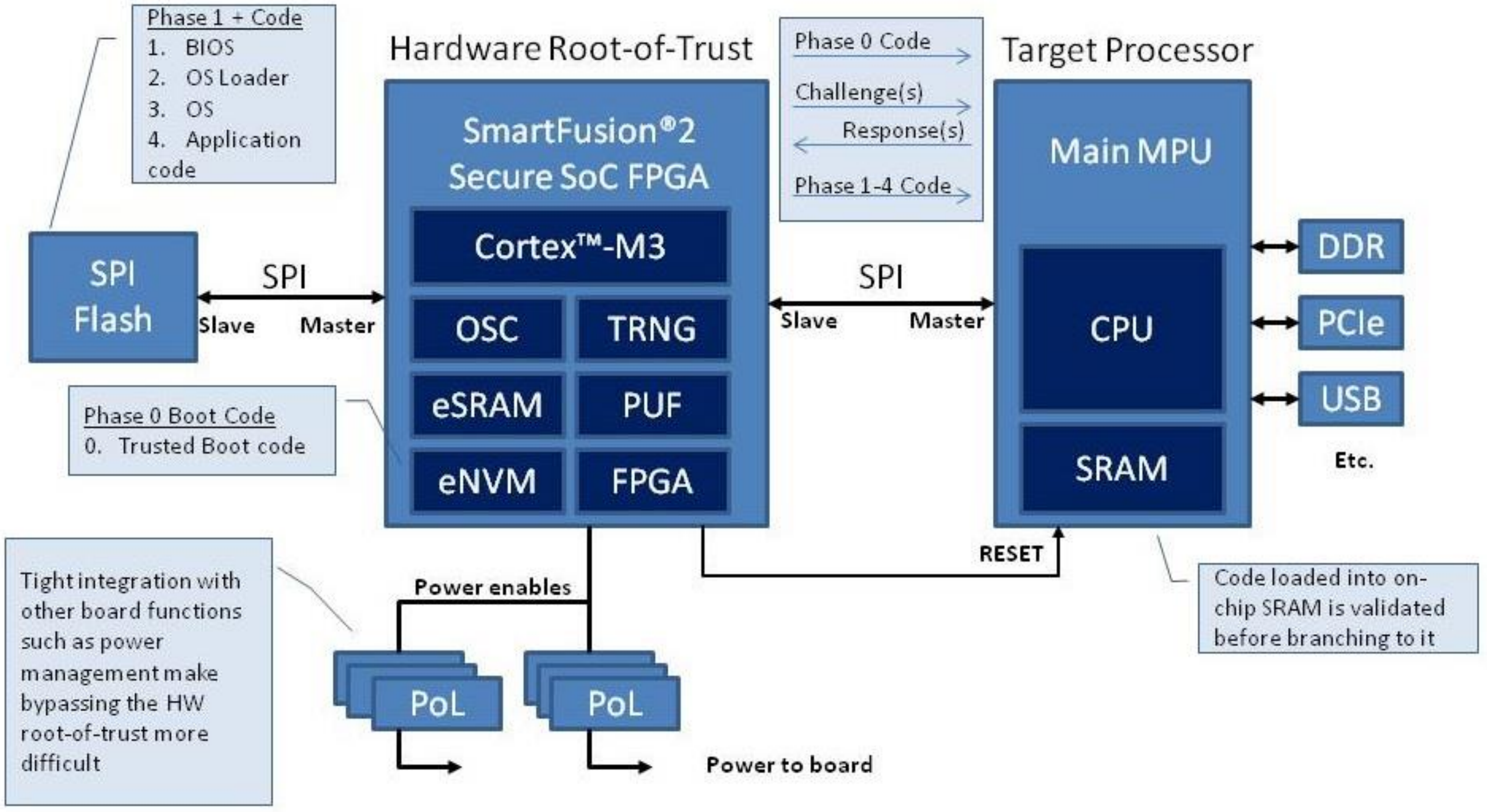


Non-Secure Processors

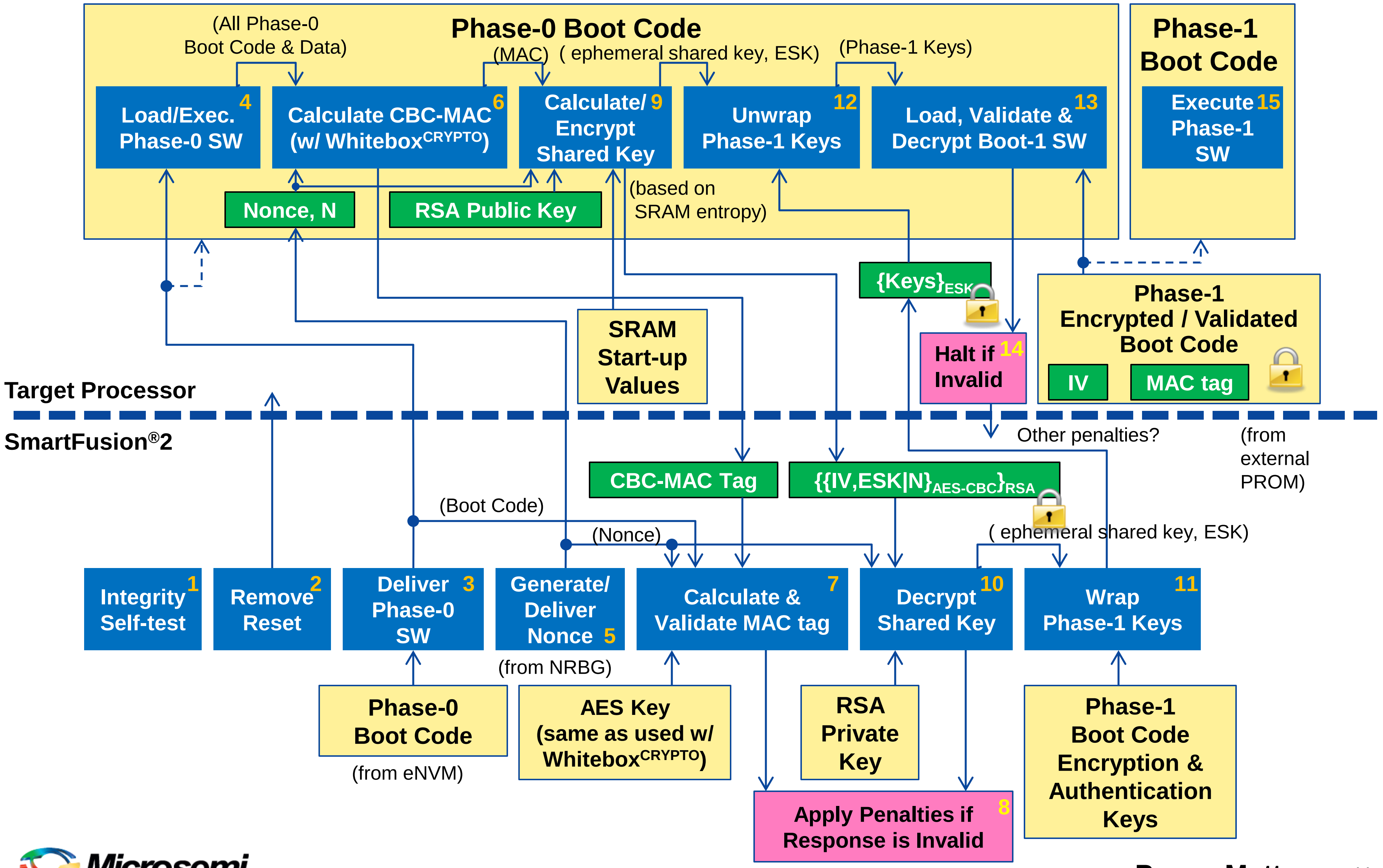
- Secure boot is challenging when the target processor has no inherent security capabilities
  - Does not have root-of-trust capability
  - Low end Microprocessors, Microcontrollers, DSP’s
- Customers in many market segments face this issue
  - Telecom, Military, Industrial, Medical, Energy
- Processors located in peripheral or remote subsystems are especially vulnerable to attack

Secure Boot Hardware Configuration

- SmartFusion2/Igloo2 can be used as a root-of-trust
  - Assist in securing the multi-phase boot processes



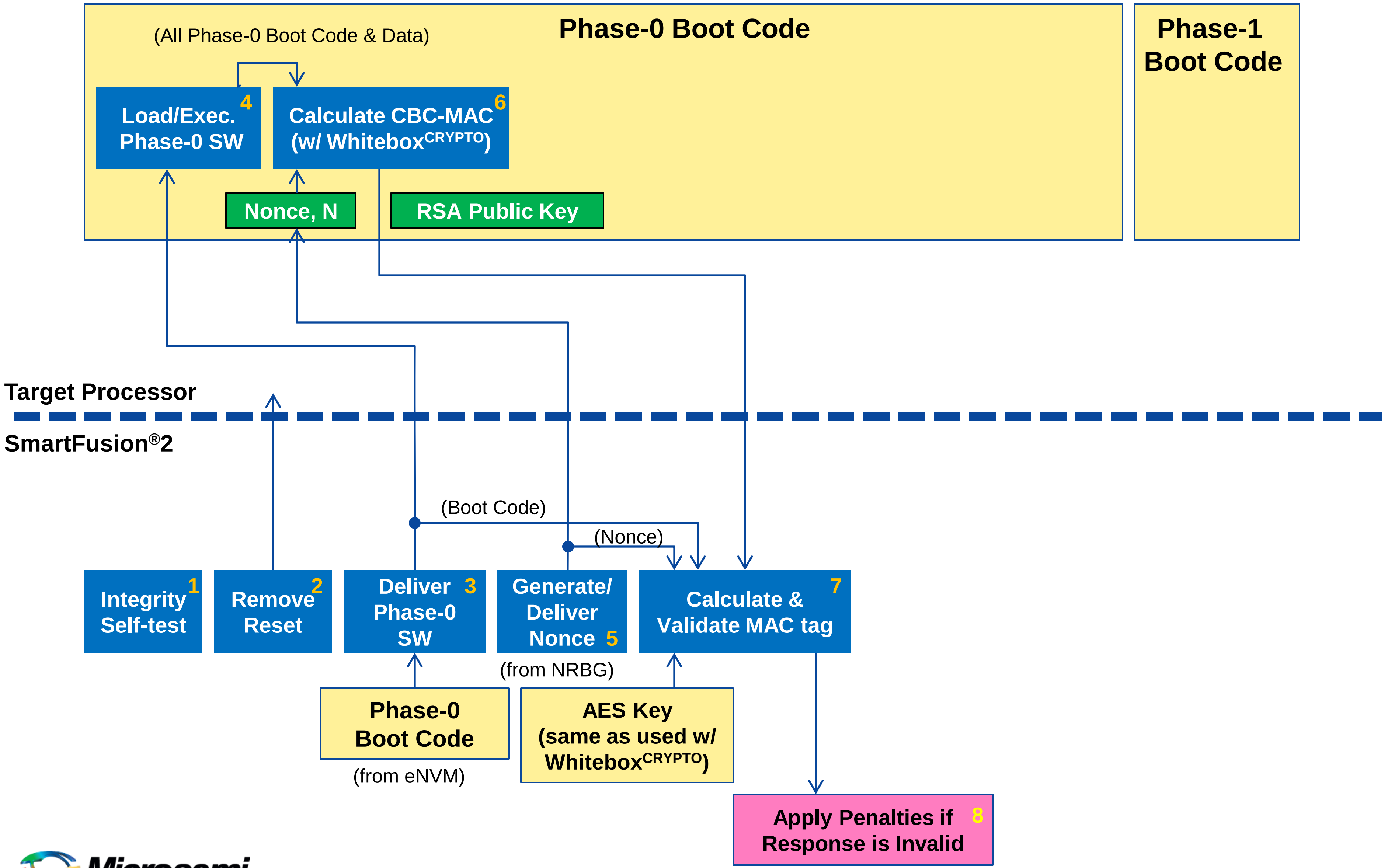
Secure Boot Process (Phase-0 and 1)



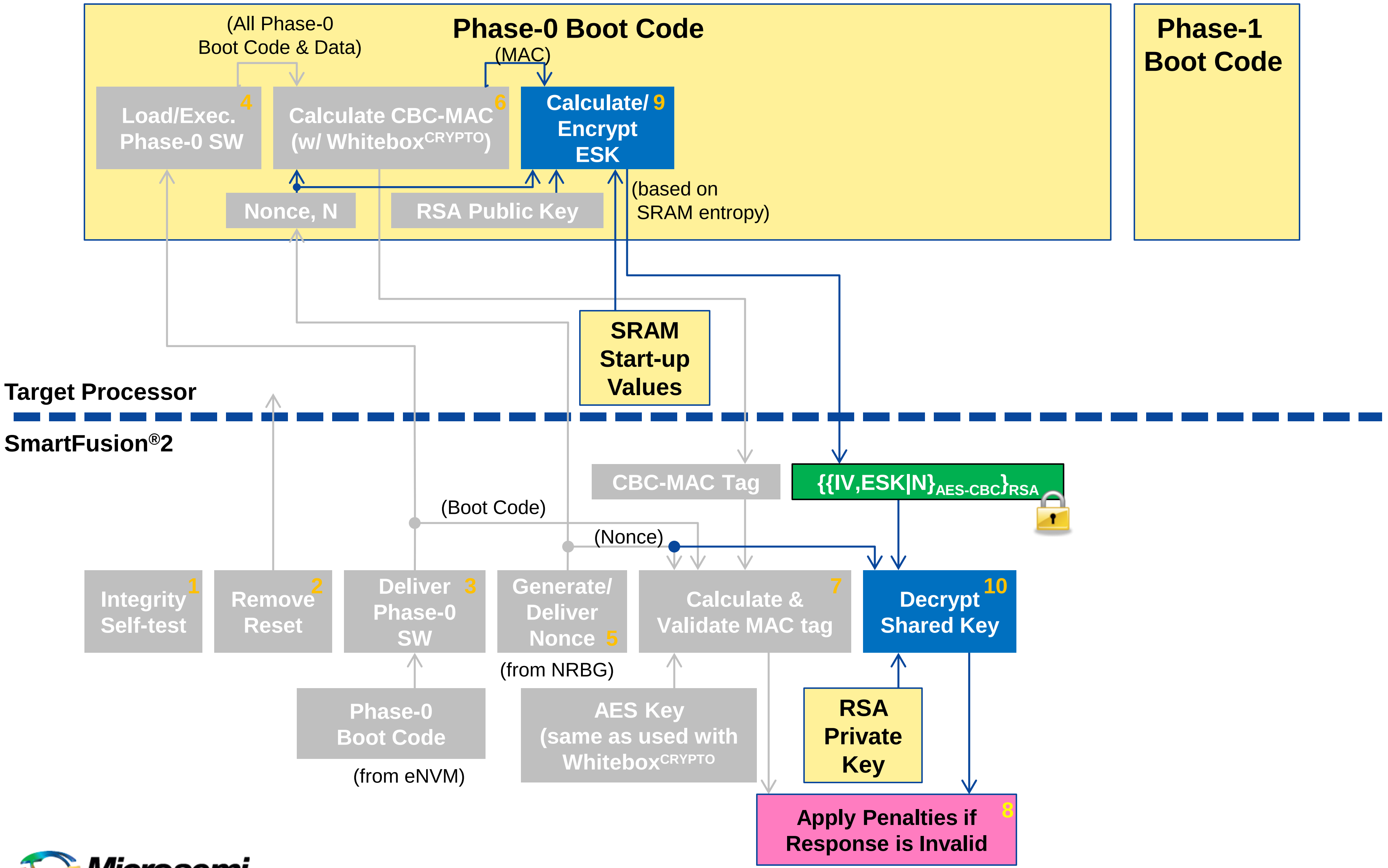
Secure Boot in a ‘nut shell’

- Assumes processor has no inherent security capabilities
- Securely boot processor with Phase-0 code
- Establish a AES session key between processor and SmartFusion2 RoT to secure the communication channel
- Transfer subsequent Phase-x code and code validation keys over secure channel to target processor.
- Target processor will validate Phase-x code
  - Execute code if validated
  - Apply system penalties if code does not validate

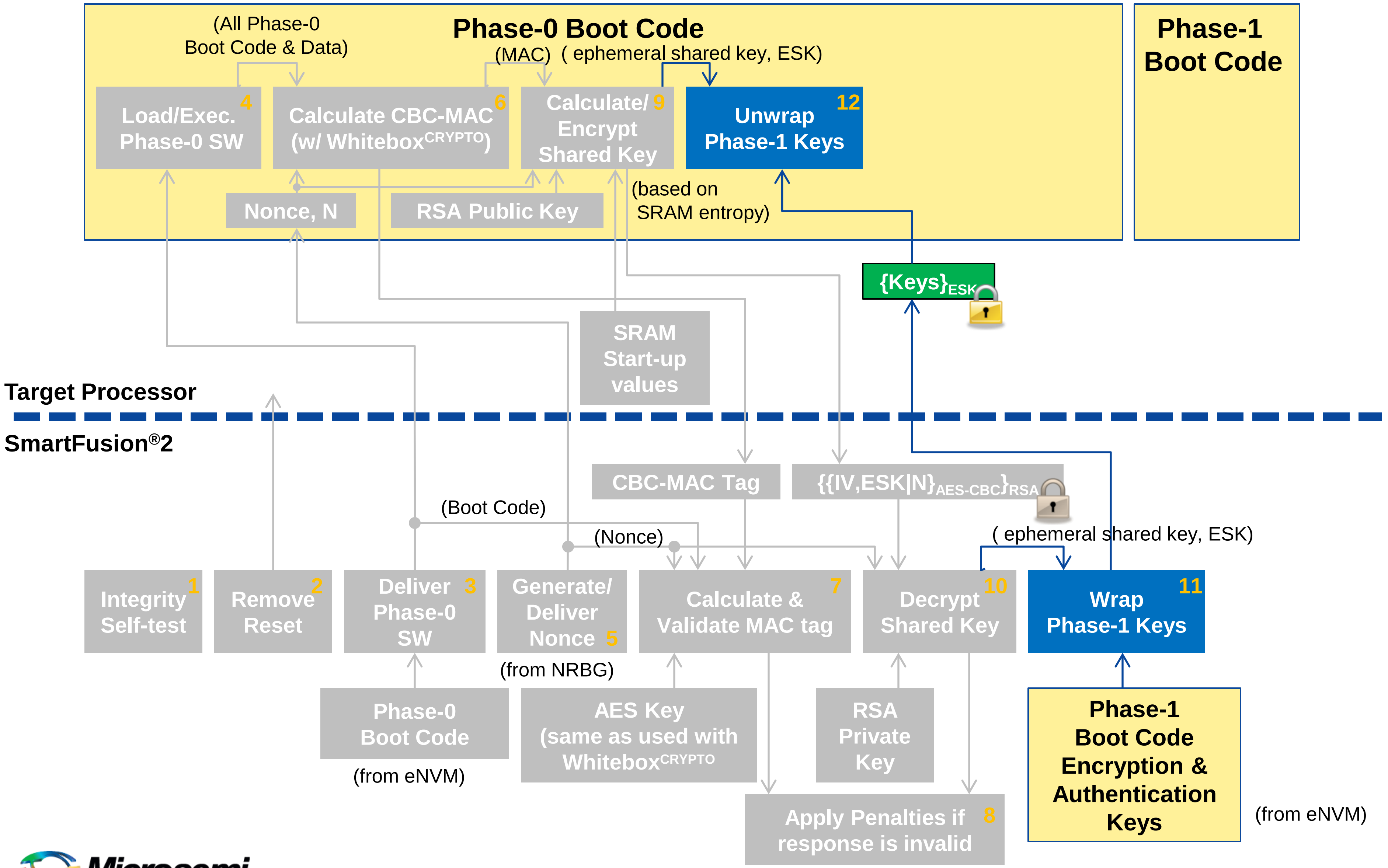
Load/Validate Phase-0 Code



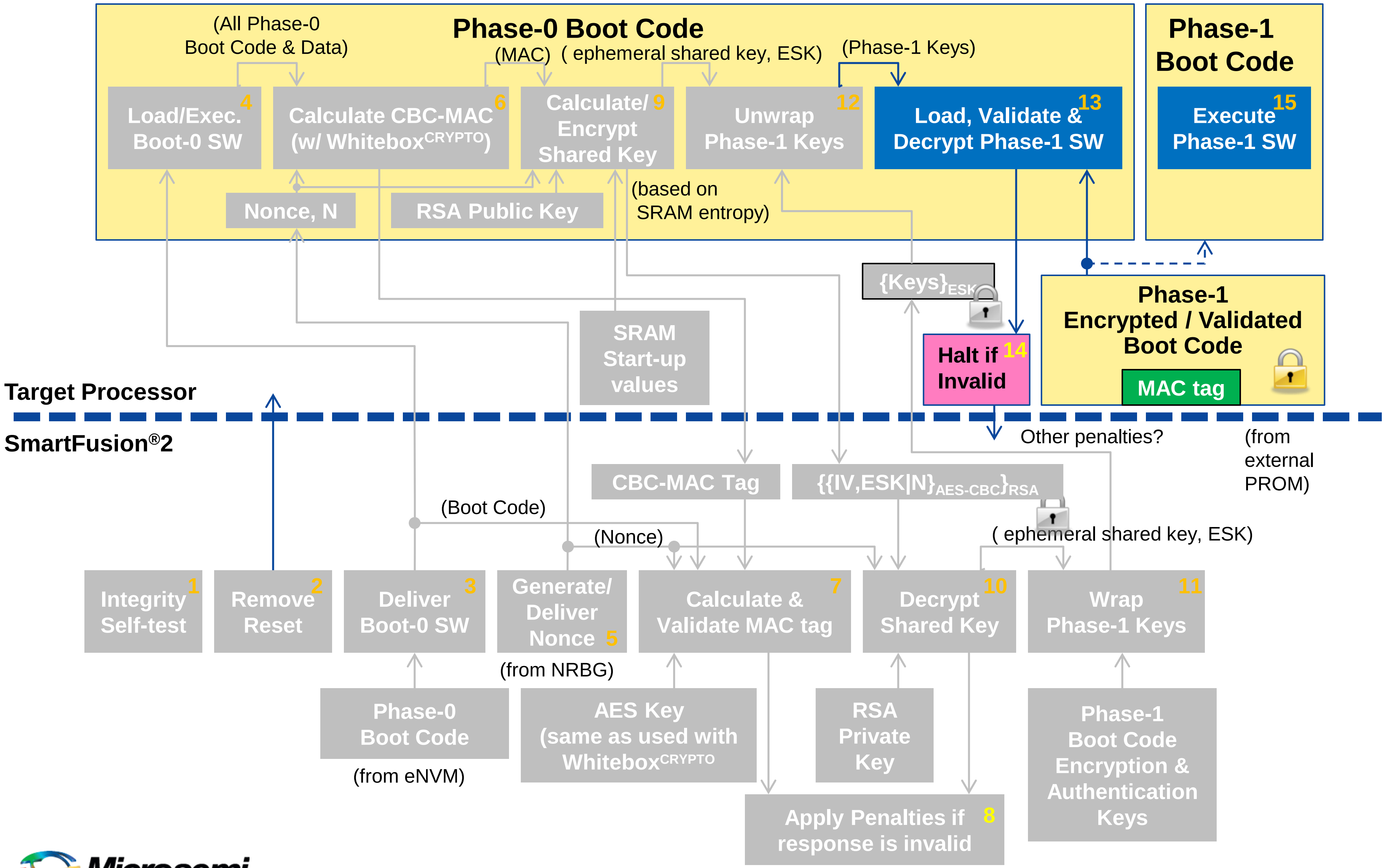
Establish Ephemeral Session Key(ESK)



Share Phase-x Encryption/Validation Keys


Phase-0 Boot Code
(All Phase-0 Boot Code & Data)
(MAC)
(ephemeral shared key, ESK)
Load/Exec. Phase-0 SW 4
Calculate CBC-MAC (w/ Whitebox^{CRYPTO}) 6
Calculate/ Encrypt Shared Key 9
Unwrap Phase-1 Keys 12
Nonce, N
RSA Public Key
(based on SRAM entropy)
Phase-1 Boot Code
$\{Keys\}_{ESK}$
SRAM Start-up values
Target Processor
SmartFusion®2
CBC-MAC Tag
$\{\{IV,ESK|N\}_{AES-CBC}\}_{RSA}$
(Boot Code)
(Nonce)
(ephemeral shared key, ESK)
Integrity Self-test 1
Remove Reset 2
Deliver Phase-0 SW 3
Generate/ Deliver Nonce 5
Calculate & Validate MAC tag 7
Decrypt Shared Key 10
Wrap Phase-1 Keys 11
(from NRBG)
Phase-0 Boot Code
(from eNVM)
AES Key (same as used with Whitebox^{CRYPTO})
RSA Private Key
Apply Penalties if response is invalid 8
Phase-1 Boot Code Encryption & Authentication Keys
(from eNVM)

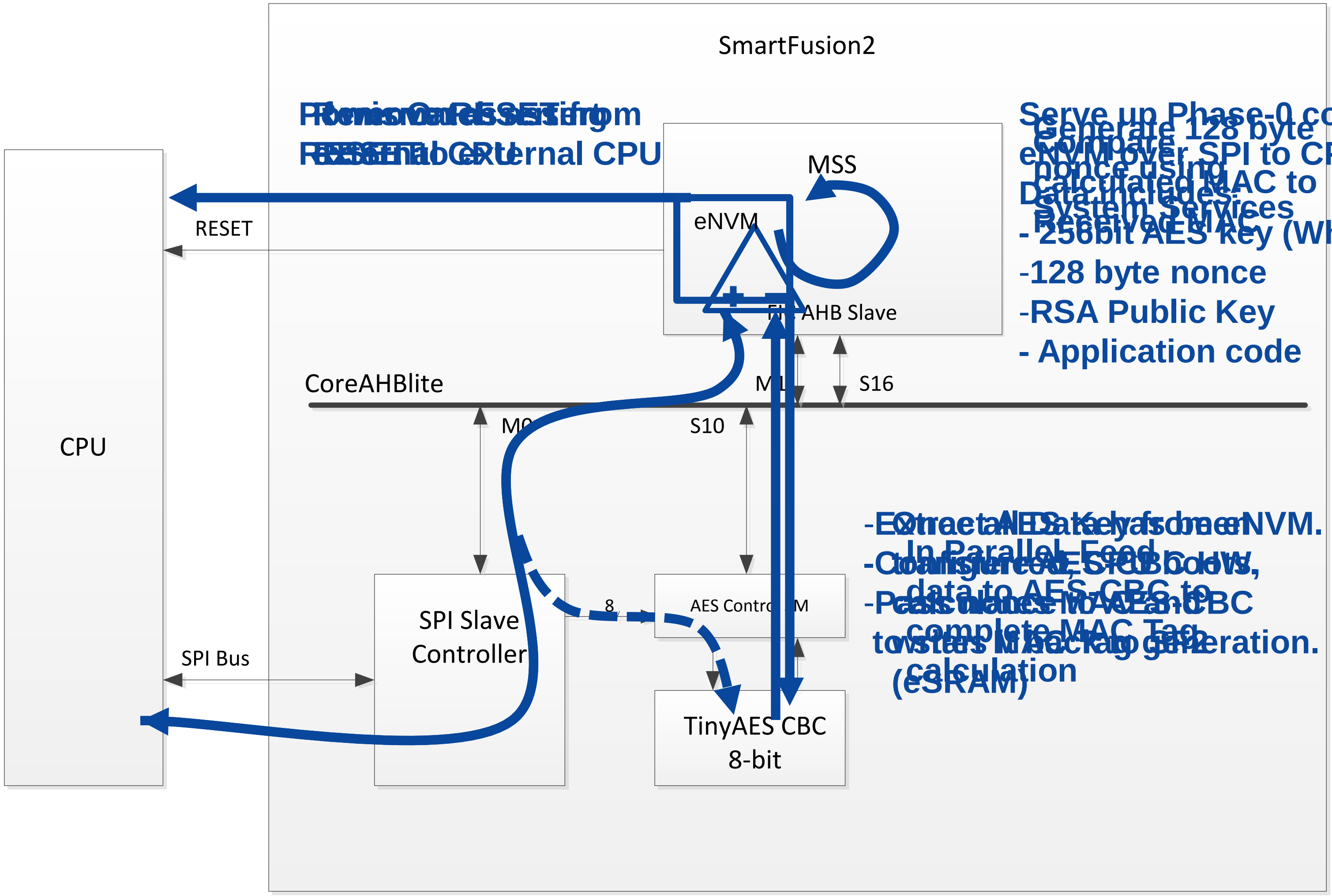
Load/Validate/Decrypt/Execute Phase-x Code



Summary

- SmartFusion2/Igloo2 are ideal devices to implement RoT
  - Anti-tamper and Integrity check hardware
  - Data storage with HW firewalls
  - True HW RNG and AES engine
  - Significant FPGA space for system integration, tamper monitoring and strong penalties.
- Secure Boot architecture is scalable
  - Modify based on target processors capabilities
- Single Secure Boot solution
  - Same solution regardless of processor capability
  - Ideal for customers deploying different processors.

Phase-0 Secure Boot Process Flow



Power On RESET from
Remove RESET to external CPU

Serve up Phase-0 code from eNVM over SPI to CPU. Data includes:
- 256bit AES key (Whitebox)
-128 byte nonce
-RSA Public Key
- Application code

Generate 128 byte nonce using System Services

Compare calculated MAC to Received MAC

-Extract AES Key from eNVM.
-Configure AES-CBC HW,
-Pass nonce to AES-CBC to start MAC Tag generation.

In Parallel–Feed data to AES-CBC to complete MAC Tag calculation

(eSRAM)

Secure Boot Demo

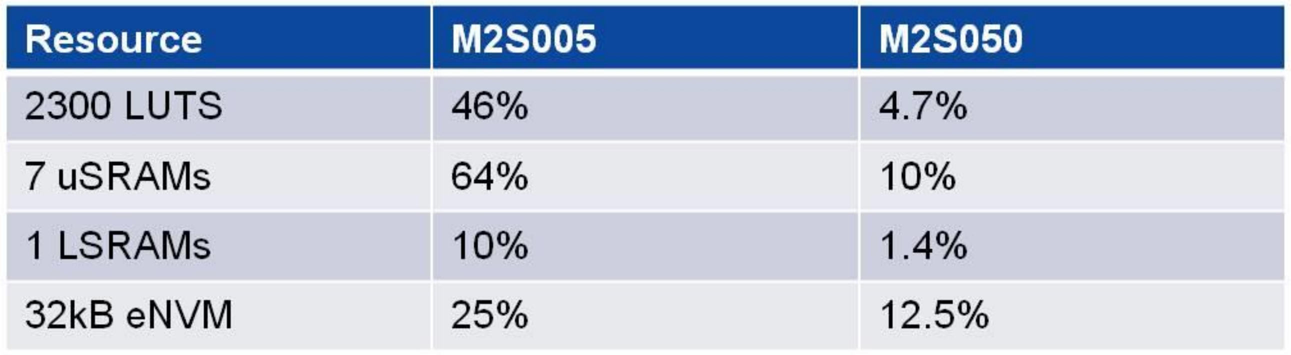
- Demonstrate secure boot of a non-secure processor
  - Microprocessor that has no inherent security capabilities
  - SmartFusion Eval board used for convenience
- Demonstrates Phase-0 and Phase-1 boot stages
  - Boot an unsecure microprocessor with trusted code
  - Validate that microprocessor is executing this trusted code
- Demonstrates use of SmartFusion2 as a root-of-trust

Secure Boot Device Resource Utilization

- Cortex-M3 Based SmartFusion2 Utilization (Phase-0 and 1)

Resource	M2S005	M2S050
1170 LUTS	24%	2.4%
7 uSRAMs	64%	10%
0 LSRAMs	0%	0%
64kB eNVM	50%	25%

- CoreABC based SmartFusion2/Igloo2 utilization (Phase-0 only)



Resource	M2S005	M2S050
2300 LUTS	46%	4.7%
7 uSRAMs	64%	10%
1 LSRAMs	10%	1.4%
32kB eNVM	25%	12.5%

- Performance
  - Max SPI Clock = 30MHz (eNVM & AES limited)

Thank You!

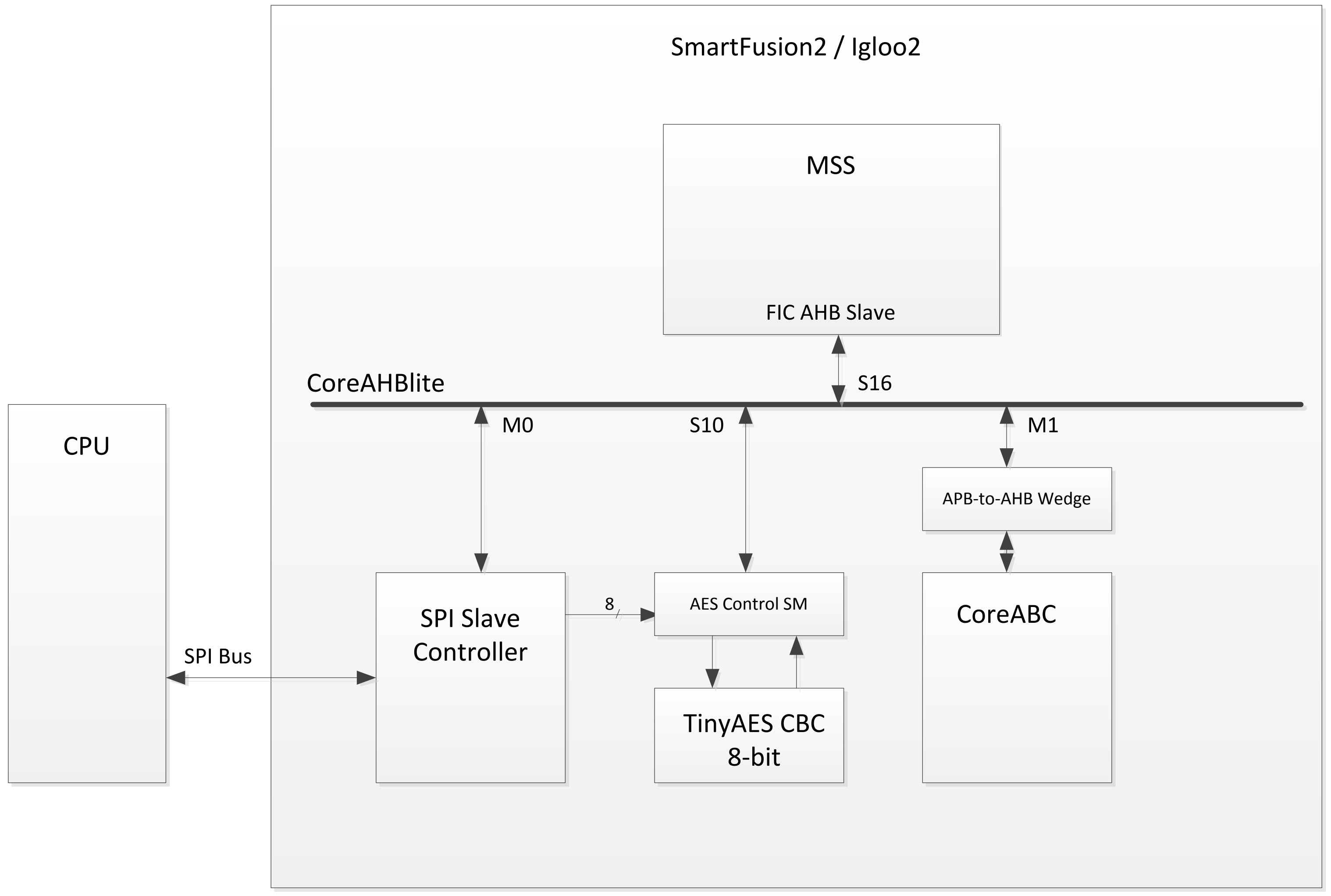
SmartFusion® 2

Microsemi Corporation
One Enterprise, Aliso Viejo, CA 92656
Ph: 949-380-6100
Fax: 949-215-4996
www.microsemi.com
For more information: Sales.Support@microsemi.com

Backup Slides

Backup Slides

Secure Boot Using CoreABC


SmartFusion2 / Igloo2
MSS
FIC AHB Slave
CoreAHBlite
S16
M0
S10
M1
APB-to-AHB Wedge
CPU
SPI Bus
SPI Slave Controller
8
AES Control SM
CoreABC
TinyAES CBC 8-bit