How to protect your firmware against malicious attacks using the latest Kinetis development board

April 25, 2017

IoT and Security Solutions

SECURE CONNECTIONS FOR A SMARTER WORLD

PUBLIC

Agenda

- IoT Phishing: “*I have a bad feeling about this…”*
- Applying a security model
- NXP Kinetis MCU solution
  - Kinetis K28F MCU
  - mbed TLS
  - KBOOT
- Overview of methods
- Development steps
- Key management options
- Resources and next steps

1

IoT: Phishing with Edge Nodes

Driving Internet of Things (IoT) Innovation

Society

Data centers

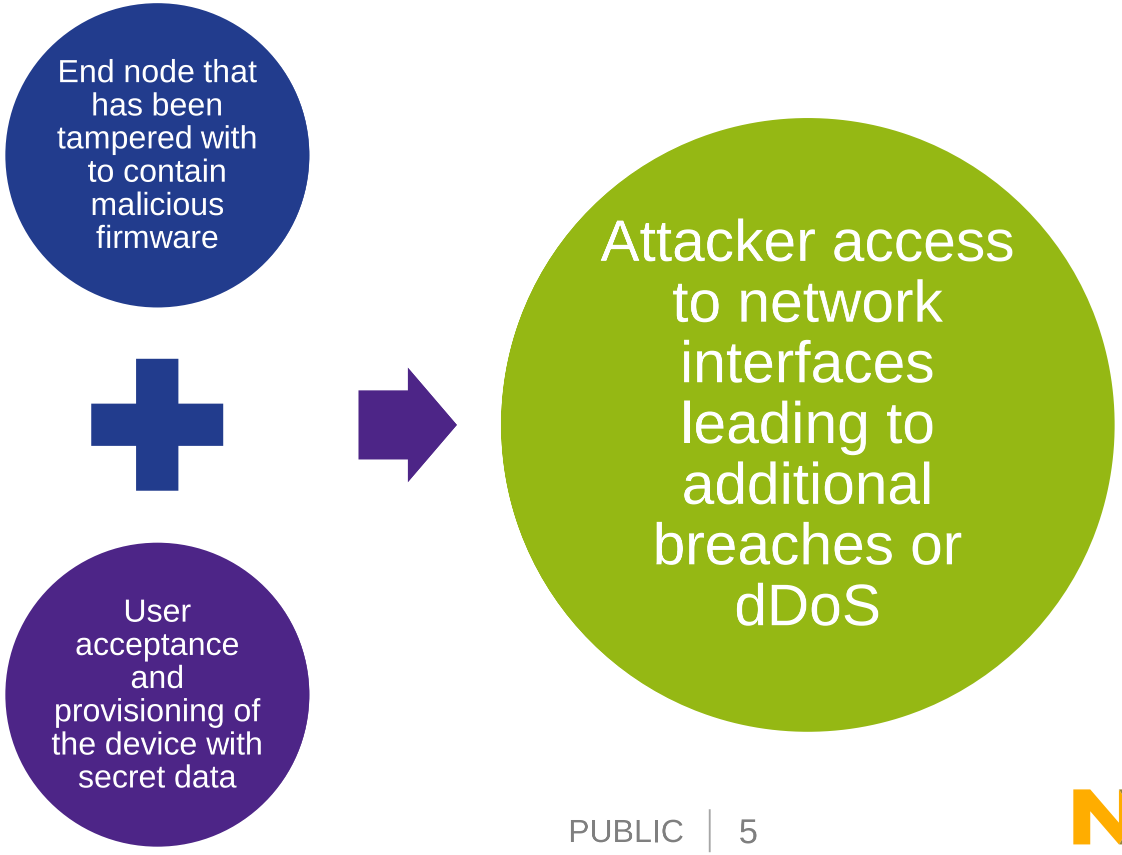
Vehicle

Cities

Transactions

Phishing with IoT Edge Node Lures – A new attack vector to prepare for.

- By now, people from all walks of life are aware of email phishing scams that are used to inject malware onto personal computing devices.
- **But what about a phishing attack that uses an IoT edge node as a lure?**
- **How can this happen?**



End node that has been tampered with to contain malicious firmware

+

User acceptance and provisioning of the device with secret data

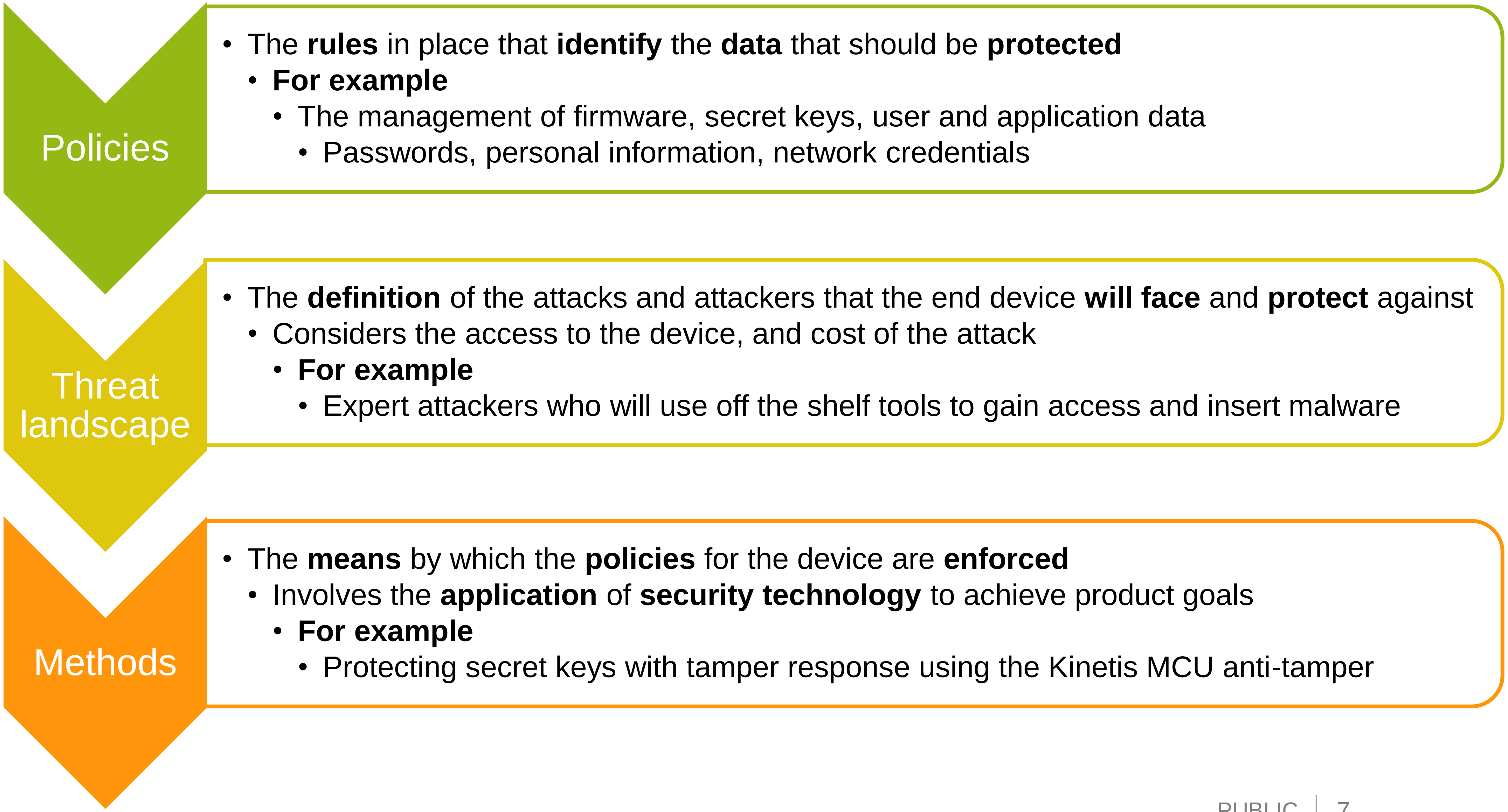
→

Attacker access to network interfaces leading to additional breaches or dDoS

2

Applying a Security Model

Begin with a Security Model



Policies

- The **rules** in place that **identify** the **data** that should be **protected**
  - **For example**
    - The management of firmware, secret keys, user and application data
      - Passwords, personal information, network credentials

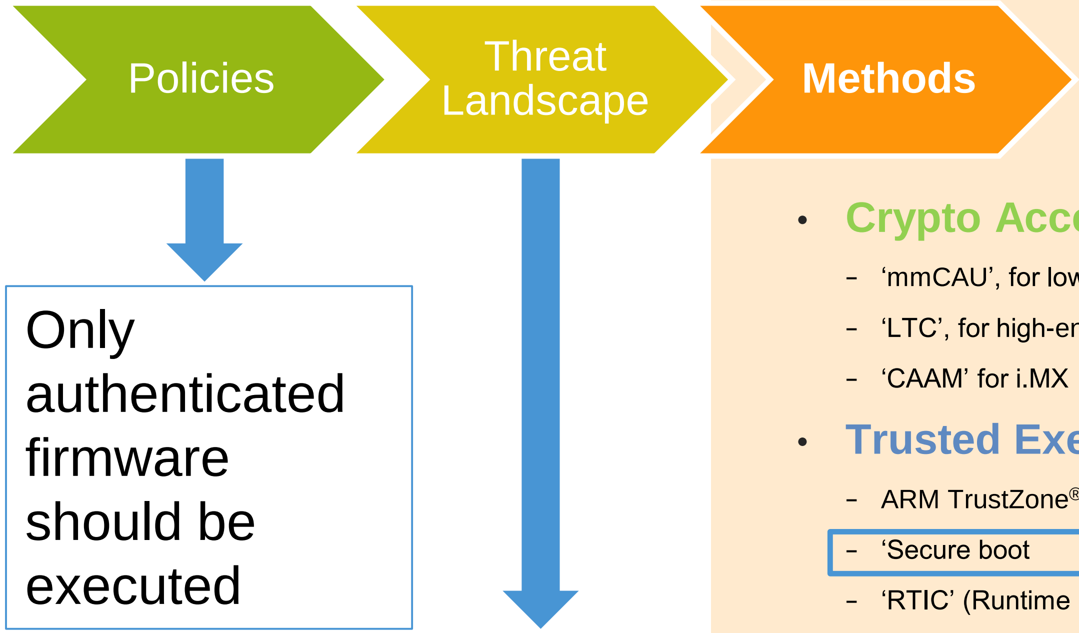
Threat landscape

- The **definition** of the attacks and attackers that the end device **will face** and **protect** against
  - Considers the access to the device, and cost of the attack
    - **For example**
      - Expert attackers who will use off the shelf tools to gain access and insert malware

Methods

- The **means** by which the **policies** for the device are **enforced**
  - Involves the **application** of **security technology** to achieve product goals
    - **For example**
      - Protecting secret keys with tamper response using the Kinetis MCU anti-tamper

A Security Model | Methods



Policies

Threat Landscape

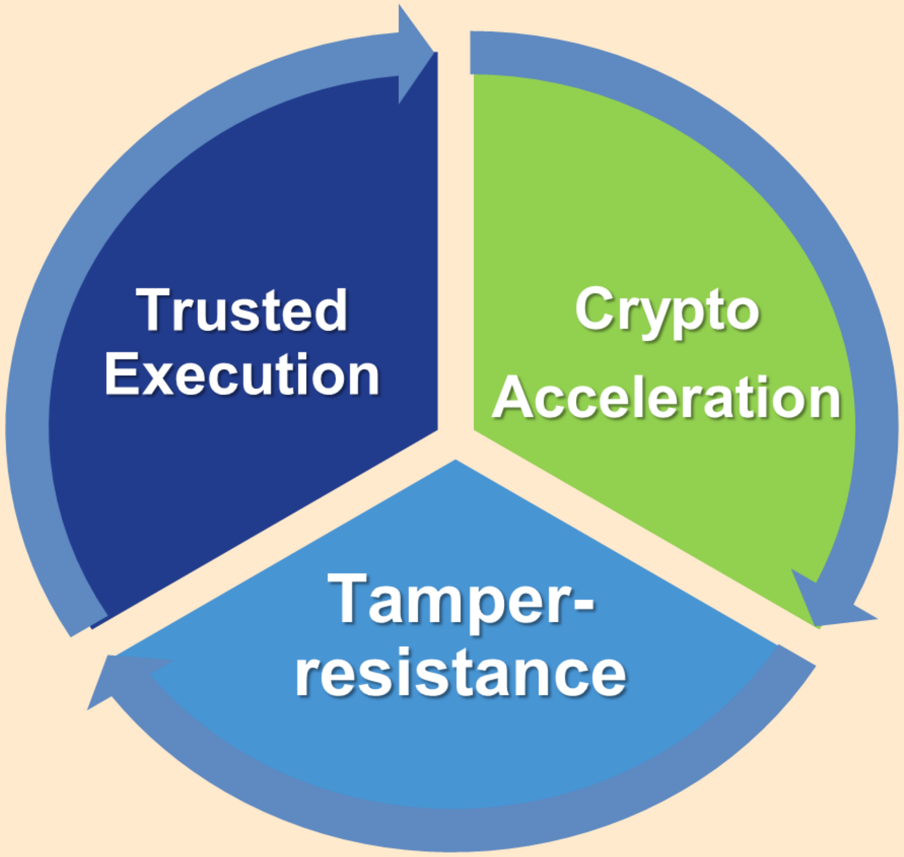
Methods

Only authenticated firmware should be executed

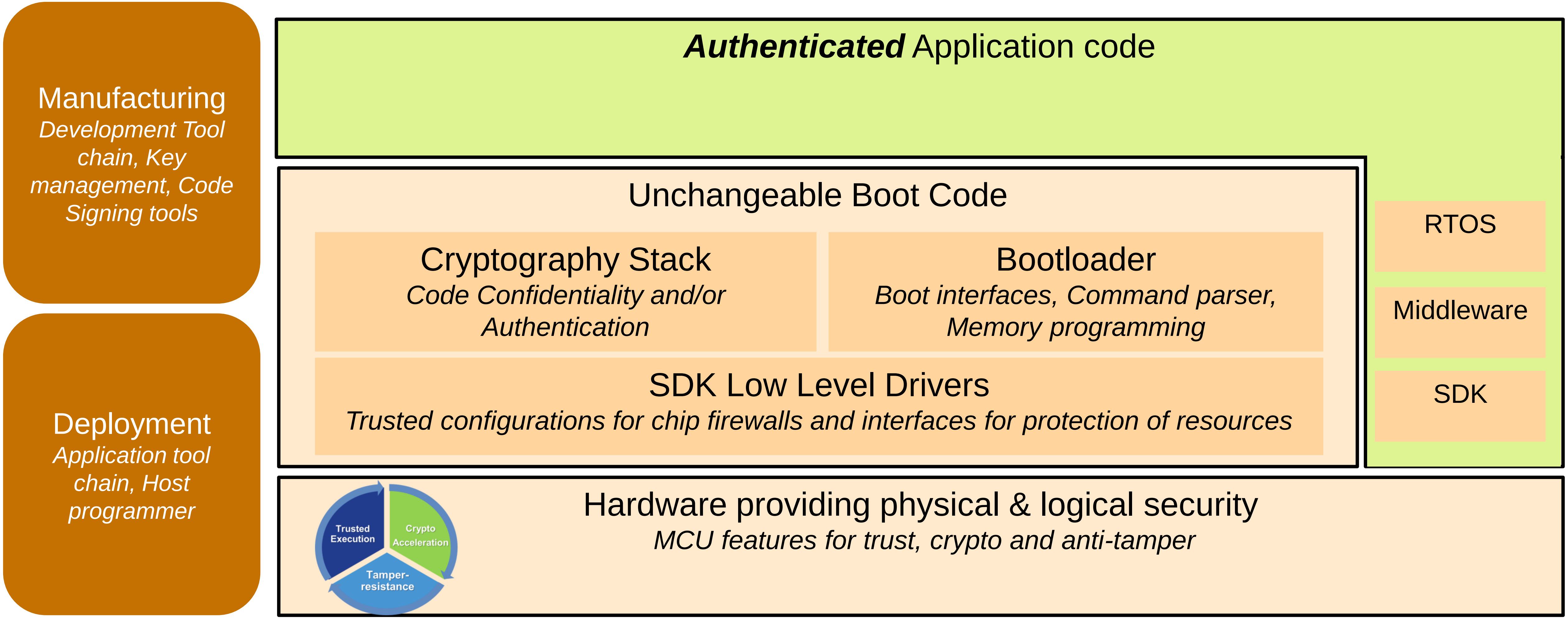
Physical access to the device communication ports

Security Technology: Hardware View

- **Crypto Acceleration**
  - ‘mmCAU’, for low-end Kinetis
  - ‘LTC’, for high-end Kinetis
  - ‘CAAM’ for i.MX
- **Trusted Execution**
  - ARM TrustZone®
  - ‘Secure boot
  - ‘RTIC’ (Runtime Integrity Checker)
  - Secure debug
  - Secure storage
  - Resource domain isolation (MPU, FAC)
- **Tamper Resistance**
  - Erases secrets at tamper detect
  - Active and passive tampers



Security Technology | Secure Boot System View



Manufacturing
Development Tool chain, Key management, Code Signing tools

Deployment
Application tool chain, Host programmer

Authenticated Application code

Unchangeable Boot Code

Cryptography Stack
Code Confidentiality and/or Authentication

Bootloader
Boot interfaces, Command parser, Memory programming

SDK Low Level Drivers
Trusted configurations for chip firewalls and interfaces for protection of resources

RTOS

Middleware

SDK

Hardware providing physical & logical security
MCU features for trust, crypto and anti-tamper

3

NXP Kinetis MCU Solution

Kinetis K27/K28 USB MCUs

Industry's Largest Embedded SRAM Memory on
ARM® Cortex®-M4-based MCU, Optimized for Portable Devices

Largest Embedded SRAM

- 1MB of embedded SRAM plus 2MB of Flash memory to enable longer battery life and richer graphics in portable display applications

Lower System Power

- 150 MHz Kinetis MCU enables advanced integration in battery-operated applications

Advanced Integration

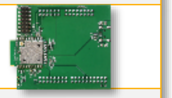
- Reduces system board footprint required by wearables and other low-end graphic display systems

Complete Enablement

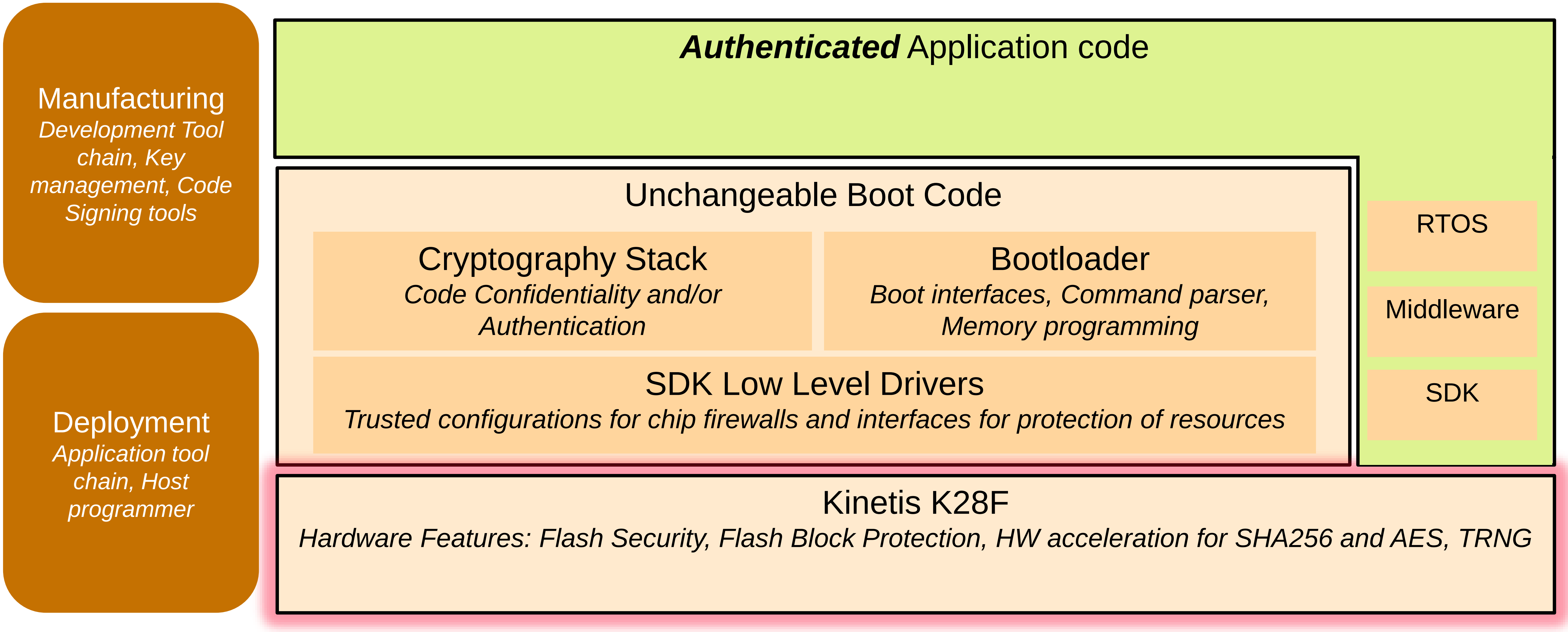
- Low-cost FRDM-K28F development platform, optional 5" LCD display board with capacitive touch from MikroElektronika, MCUXpresso software and tools

Kinetis K27F/K28F HW and SW Enablement Plan

K28F	
HW BOARD	BASELINE SW ENABLEMENT
 FRDM-K28F Low-cost evaluation platform for K27F/K28F family with on-board discrete power management, Accelerometer, SDRAM memory, QuadSPI Serial Flash, USB High-Speed connector and Full-Speed USB OpenSDA Target resale price: $40	IDE: • MCUXPresso IDE • IAR • KEIL
	RTOS: • FreeRTOS Bare metal (no RTOS)
	Kinetis Expert: • Power Estimator tool • Pin Configuration tool • Clock Configuration tool • Peripheral configuration tool
	MCUXPresso SDK 2.x

ADDS-ON				
OPTIONAL FEATURES	HW BOARDS	SW ENABLEMENT	BORAD CONNECTOR	
BLE	FRDM-KW41Z BLE stack running on KW41Z	Kinetis SDK 2.x + IAR + FreeRTOS	Arduino (UART)	
WI-FI	Arrow GT202 Wi-Fi stack running on QCA4002	QCA4002 Wi-Fi drivers to be ported to Kinetis SDK 2.x	Arduino (UART)	
THREAD	FRDM-KW24D512 or FRDM-KW41Z SW Stack running on KWx wireless SoC	Kinetis SDK 2.x (TBD) + FreeRTOS + IAR + NXP Thread SW SDK	Arduino (UART)	
LCD DISPLAY	MikroElectronika 5' LCD display + capacitive touch connected through FlexIO interface (8080 and/or 6800 modes)	MCUXPresso SDK 2.x + MicroEJ (3rd party) SW support + emWIN (3rd party) SW support	FlexIO	
SENSOR	FRDM-STBC-AGM01 (Sensor Fusion) 9-axis inertial measurement solution: 3-axis Gyro, 3D Accelero + Magneto	Kinetis SDK 2.x (TBD) + FreeRTOS/Bare Metal + Sensing SDK 1.0	Arduino (I2C / SPI)	
AUDIO	ARD-AUDIO-DA7212 2-channel audio codec w/ capless headphone driver and 3.5mm stereo AUX input jack socket	*Kinetis SDK 2.x (TBD)*	Arduino (I2S)	
HOMEKIT & MFI	FRDM-TWRPI + TWRPI-I2C* MFi Adaptor boards	NXP HomeKit SDK 1.x + Kinetis SDK 1.3 + FreeRTOS + IDE (IAR or KDS)	Arduino (I2C)	

Security Technology | System view : Hardware



Manufacturing
Development Tool chain, Key management, Code Signing tools

Deployment
Application tool chain, Host programmer

Authenticated Application code

Unchangeable Boot Code

Cryptography Stack
Code Confidentiality and/or Authentication

Bootloader
Boot interfaces, Command parser, Memory programming

SDK Low Level Drivers
Trusted configurations for chip firewalls and interfaces for protection of resources

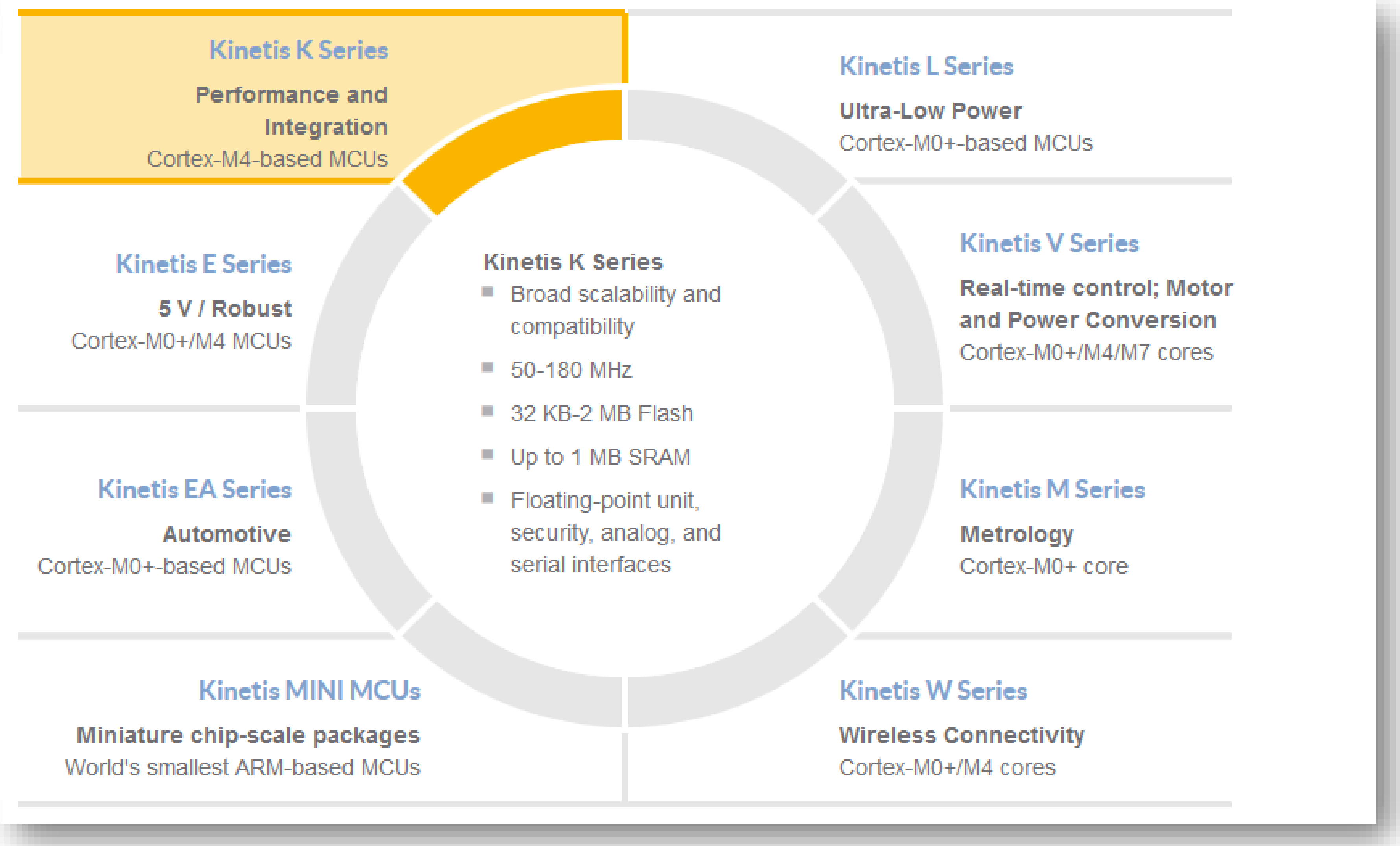
RTOS

Middleware

SDK

Kinetis K28F
Hardware Features: Flash Security, Flash Block Protection, HW acceleration for SHA256 and AES, TRNG

Kinetis Low Power 32-bit MCs Based on ARM® Cortex®-M Cores



Kinetis K Series

Performance and Integration

Cortex-M4-based MCUs

Kinetis L Series

Ultra-Low Power

Cortex-M0+-based MCUs

Kinetis E Series

5 V / Robust

Cortex-M0+/M4 MCUs

Kinetis V Series

Real-time control; Motor and Power Conversion

Cortex-M0+/M4/M7 cores

Kinetis K Series

- Broad scalability and compatibility
- 50-180 MHz
- 32 KB-2 MB Flash
- Up to 1 MB SRAM
- Floating-point unit, security, analog, and serial interfaces

Kinetis EA Series

Automotive

Cortex-M0+-based MCUs

Kinetis M Series

Metrology

Cortex-M0+ core

Kinetis MINI MCUs

Miniature chip-scale packages

World's smallest ARM-based MCUs

Kinetis W Series

Wireless Connectivity

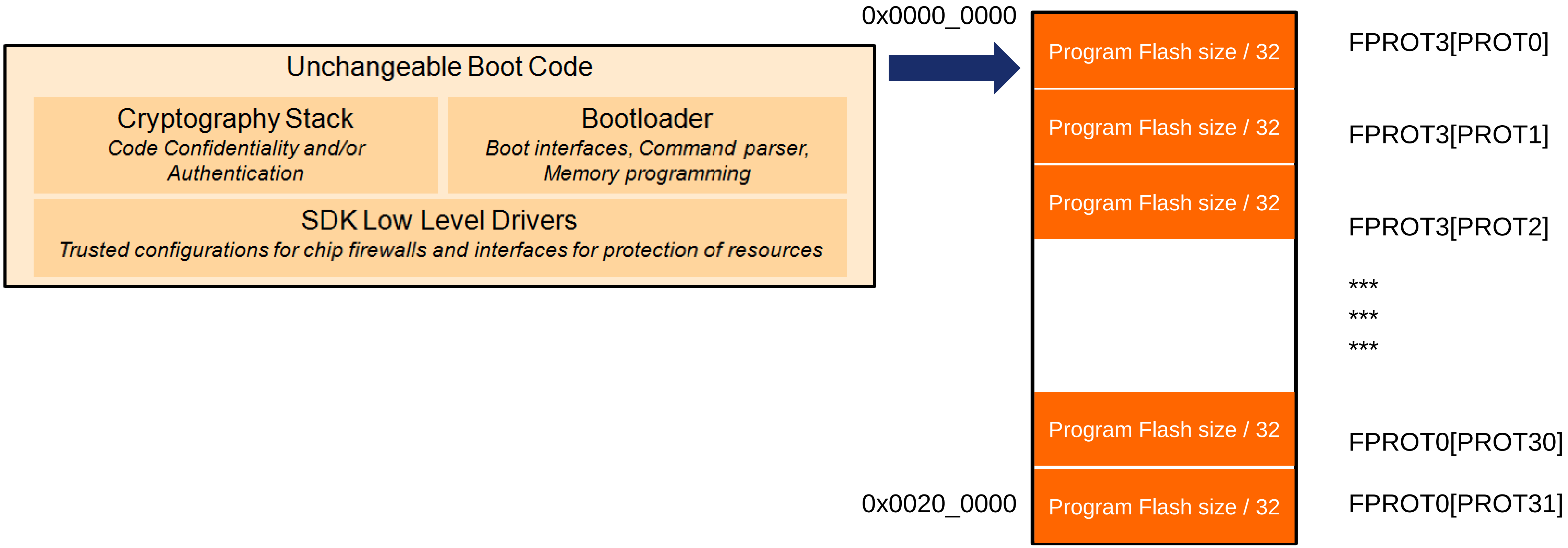
Cortex-M0+/M4 cores

Kinetis Security Technology | Essential Hardware Features

- Flash security and protection features are found on all Kinetis devices
- **Security features**
  - Kinetis offers several levels of flash security
  - Flash security is a system-level feature
    - The flash is fully functional when secured (firmware updates are still possible if resident firmware is setup to program the flash)
    - Security effects are really a system level concern. The security setting determines what the SoC will allow.
  - **Software IP is a large investment. Enabling security helps to protect that IP investment.**

- **Protection features**
  - Flash protection can be used to prevent erase or programming
  - Initial protection values are loaded from the flash configuration field at reset

Flash Block Protections to Protect Boot Code



Unchangeable Boot Code

Cryptography Stack
Code Confidentiality and/or Authentication

Bootloader
Boot interfaces, Command parser, Memory programming

SDK Low Level Drivers
Trusted configurations for chip firewalls and interfaces for protection of resources

0x0000_0000

Program Flash size / 32 FPROT3[PROT0]

Program Flash size / 32 FPROT3[PROT1]

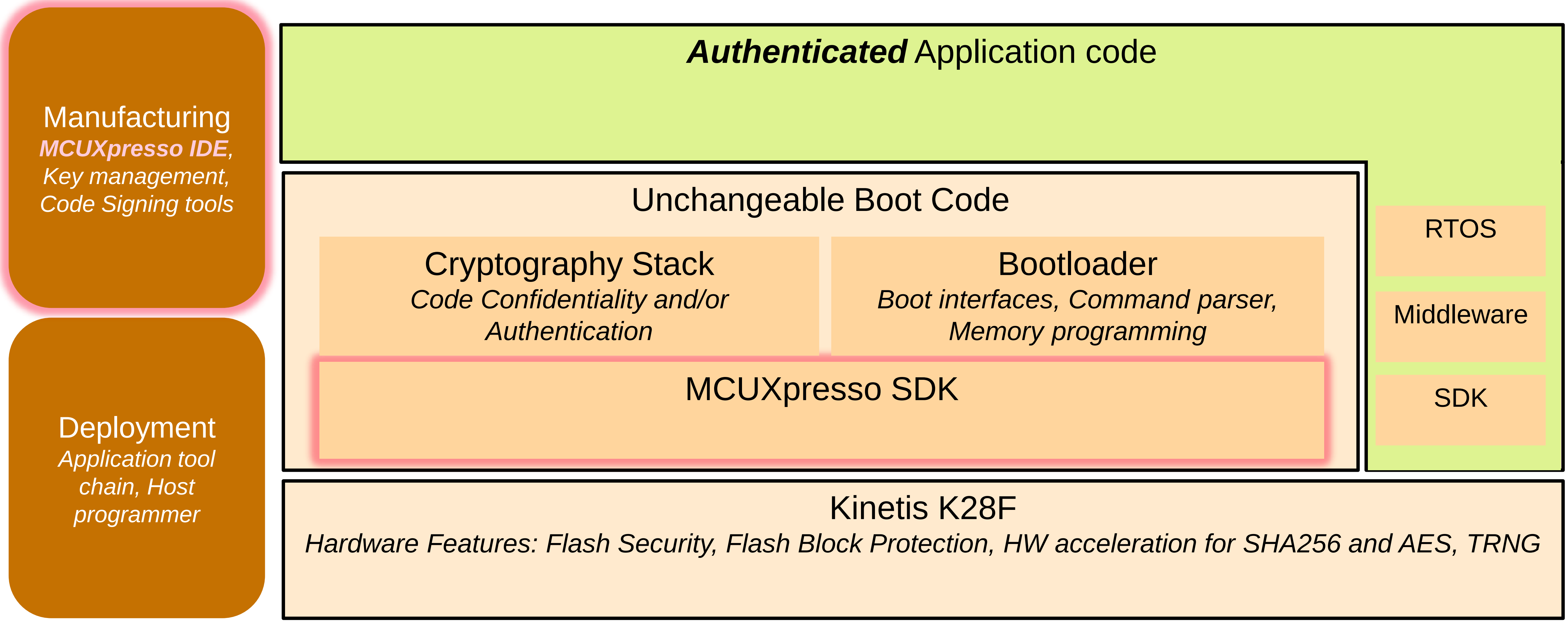
Program Flash size / 32 FPROT3[PROT2]

***
***
***

Program Flash size / 32 FPROT0[PROT30]

0x0020_0000 Program Flash size / 32 FPROT0[PROT31]

Security Technology | System View : SDK & Toolchain



Manufacturing
MCUXpresso IDE, Key management, Code Signing tools

Deployment
Application tool chain, Host programmer

Authenticated Application code

Unchangeable Boot Code

Cryptography Stack
Code Confidentiality and/or Authentication

Bootloader
Boot interfaces, Command parser, Memory programming

MCUXpresso SDK

RTOS

Middleware

SDK

Kinetis K28F
Hardware Features: Flash Security, Flash Block Protection, HW acceleration for SHA256 and AES, TRNG

Tool Chain and Software

MCUXpresso Software and Tools

for Kinetis and LPC microcontrollers

MCUXpresso IDE

Edit, compile, debug and optimize in an intuitive and powerful IDE

Available now!

MCUXpresso SDK

Runtime software including peripheral drivers, middleware, RTOS, demos and more

Available now!

MCUXpresso Config Tools

Online and desktop tool suite for system configuration and optimization

Available now!

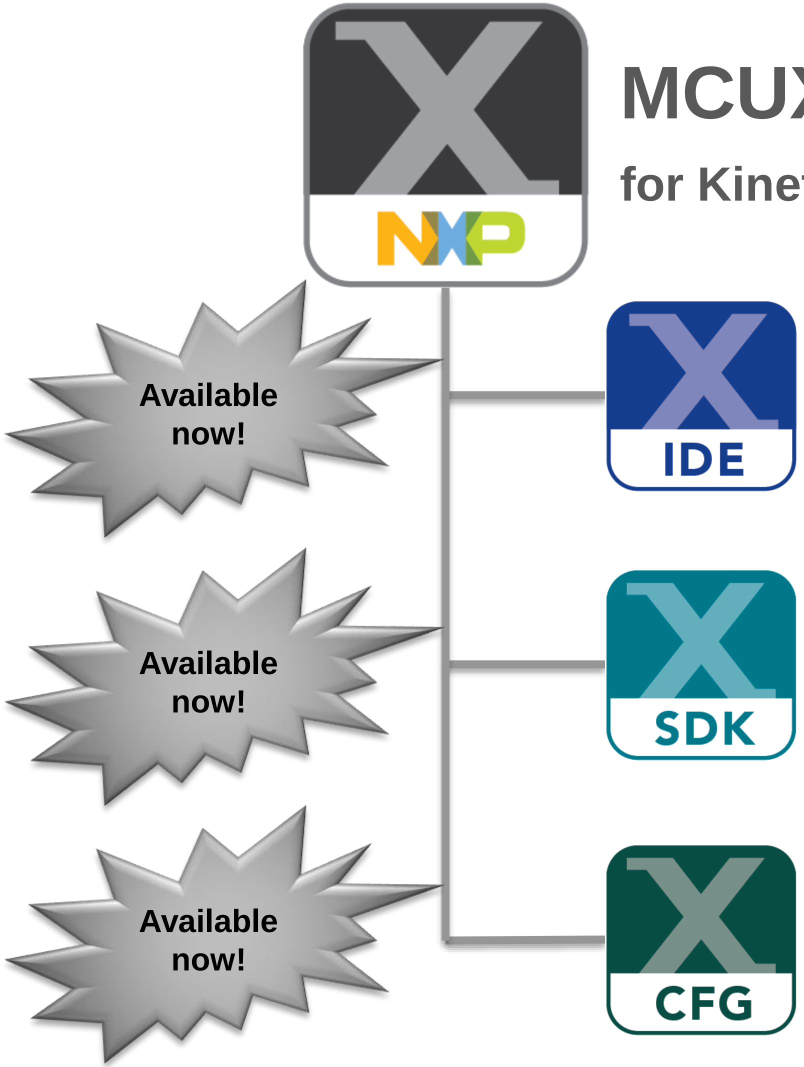
Feature-rich, **unlimited code size**, optimized for **ease-of-use**, based on **industry standard Eclipse** framework for **NXP's Kinetis** and **LPC MCUs**

Application development with Eclipse and GCC-based IDE for advanced **editing**, **compiling** and **debugging**

Supports **custom** development boards, **Freedom**, **Tower** and **LPCXpresso** boards with debug probes from **NXP**, **P&E** and **Segger**

Free Edition: Full Featured, **unlimited Code Size**, no special activation needed, community based support
Pro Edition: Email IDE support, **Advanced Trace Features**

Tool Chain and Software



MCUXpresso Software and Tools

for Kinetis and LPC microcontrollers

MCUXpresso IDE

Edit, compile, debug and optimize in an intuitive and powerful IDE

MCUXpresso SDK

Runtime software including peripheral drivers, middleware, RTOS, demos and more

MCUXpresso Config Tools

Online and desktop tool suite for system configuration and optimization

Architecture:

- CMSIS-CORE compatible
- Single driver for each peripheral
- Transactional APIs w/ optional DMA support for communication peripherals

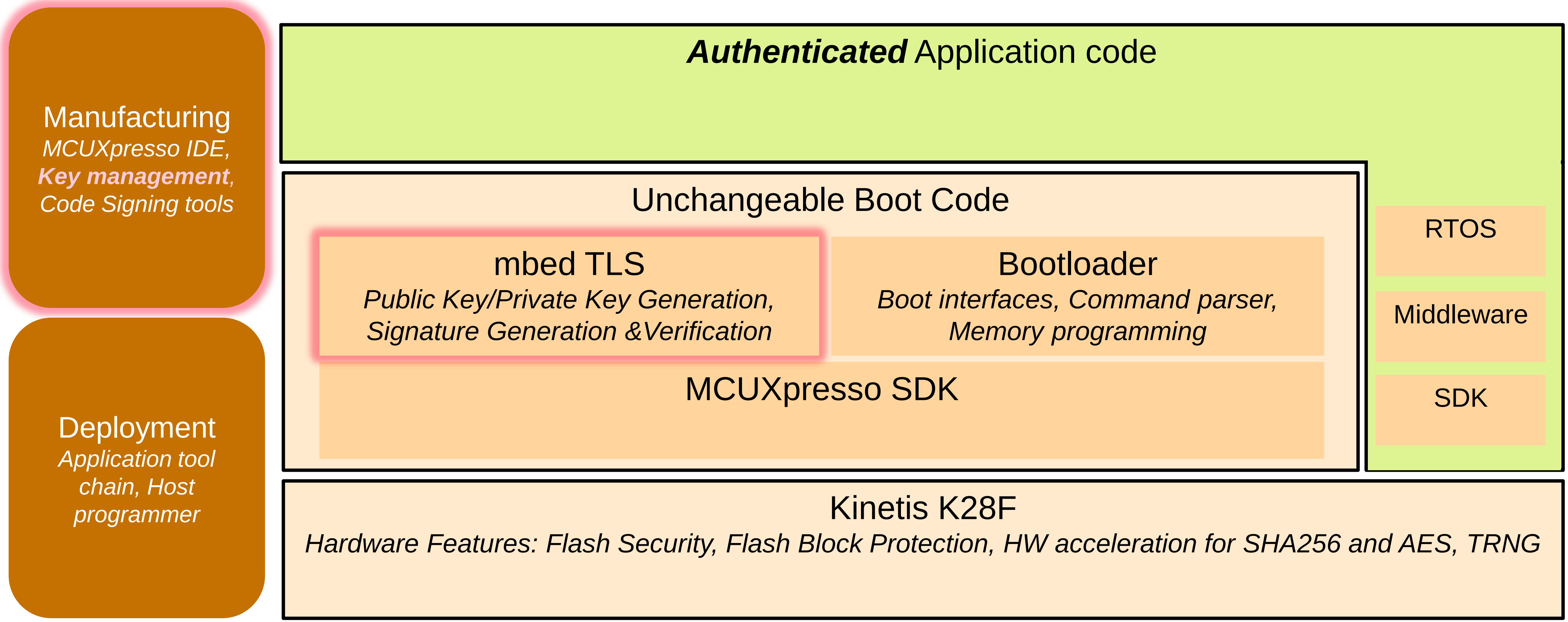
Integrated RTOS:

- FreeRTOS v9
- RTOS-native driver wrappers

Integrated Stacks and Middleware

- USB Host, Device and OTG
- lwIP, FatFS
- Crypto acceleration plus wolfSSL & **mbedTLS**
- SD and eMMC card support

Security Technology | System View : Cryptography Stack



Manufacturing
MCUXpresso IDE,
Key management,
Code Signing tools

Deployment
Application tool chain, Host programmer

Authenticated Application code

Unchangeable Boot Code

mbed TLS
Public Key/Private Key Generation, Signature Generation &Verification

Bootloader
Boot interfaces, Command parser, Memory programming

MCUXpresso SDK

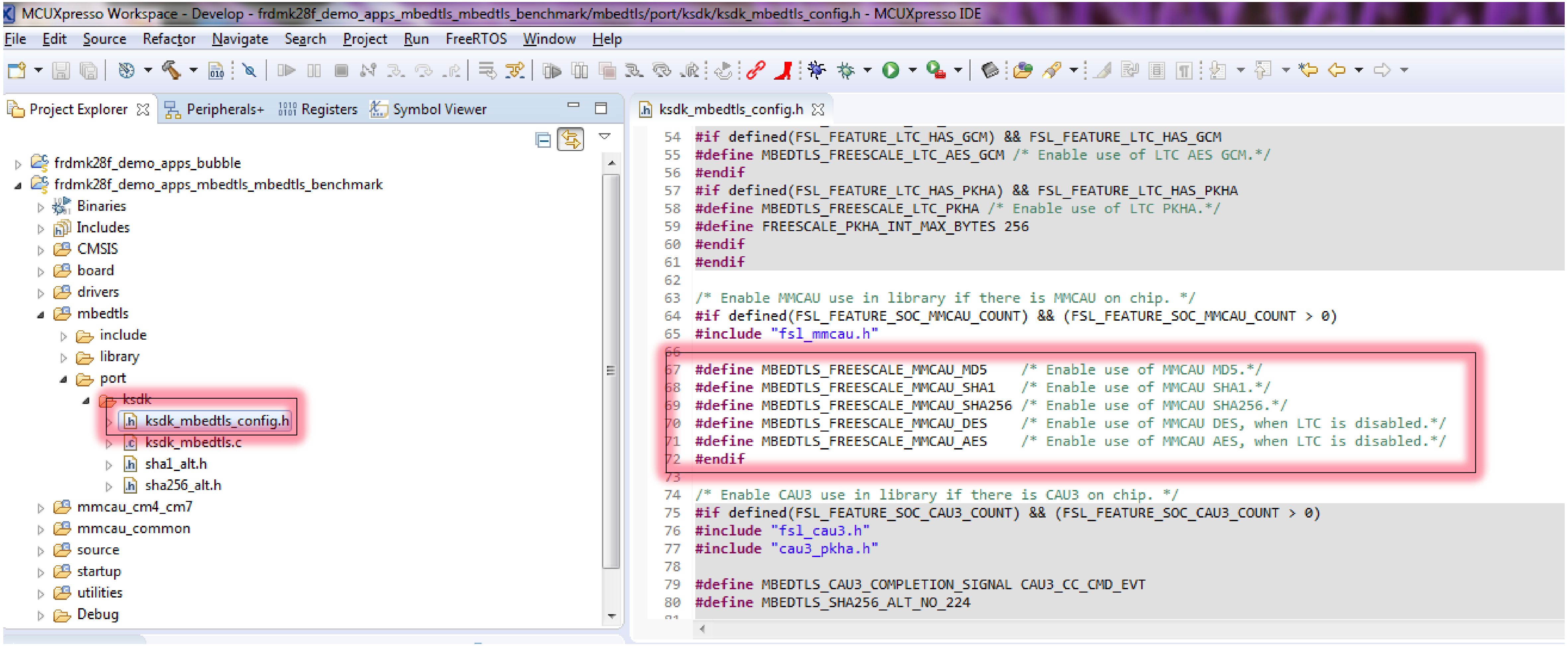
RTOS

Middleware

SDK

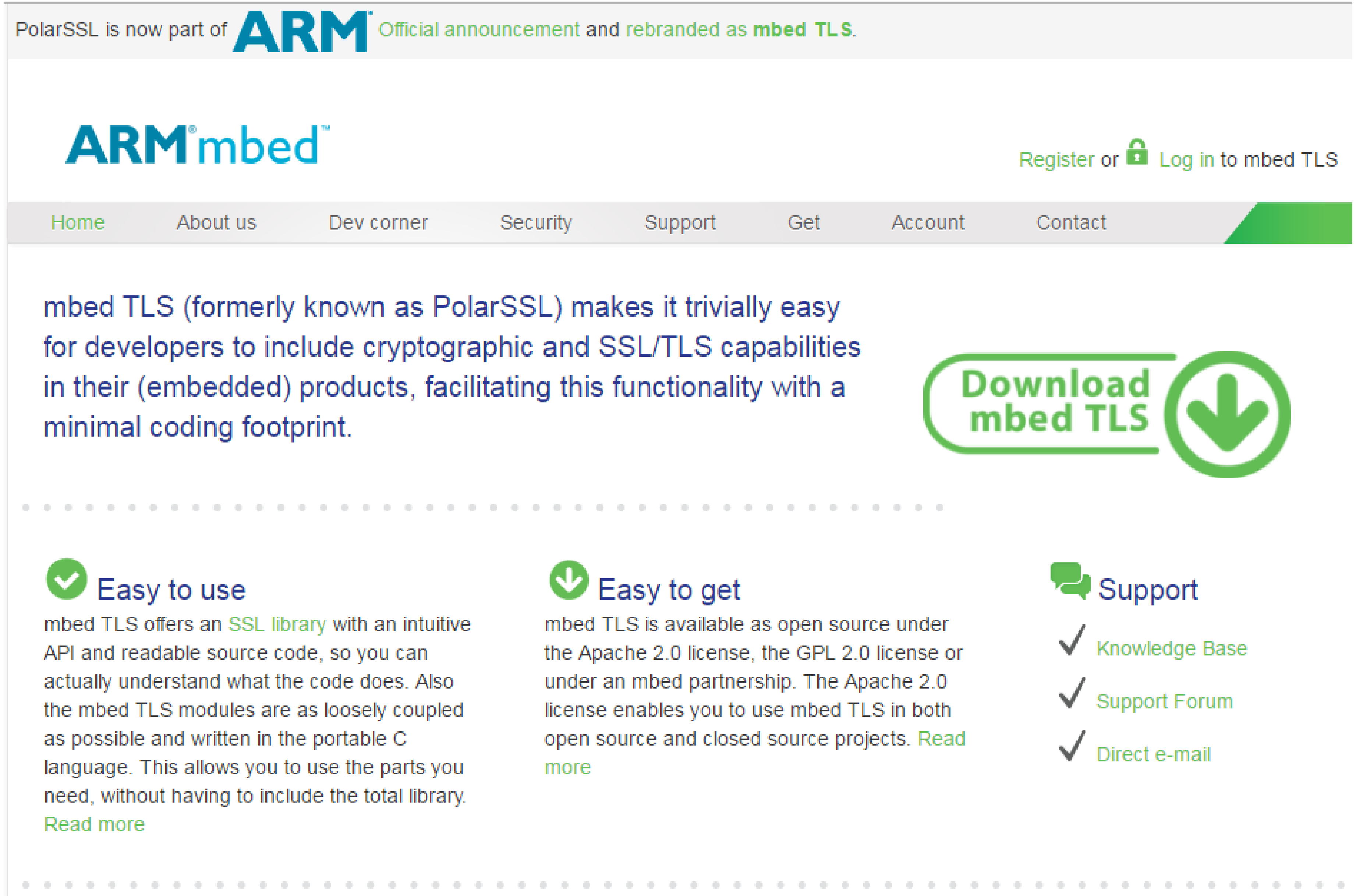
Kinetis K28F
Hardware Features: Flash Security, Flash Block Protection, HW acceleration for SHA256 and AES, TRNG

Support For Use of HW Accelerators with mbed TLS


MCUXpresso Workspace - Develop - frdmk28f_demo_apps_mbedtls_mbedtls_benchmark/mbedtls/port/ksdk/ksdk_mbedtls_config.h - MCUXpresso IDE
File Edit Source Refactor Navigate Search Project Run FreeRTOS Window Help
Project Explorer
Peripherals+
Registers
Symbol Viewer
frdmk28f_demo_apps_bubble
frdmk28f_demo_apps_mbedtls_mbedtls_benchmark
Binaries
Includes
CMSIS
board
drivers
mbedtls
include
library
port
ksdk
ksdk_mbedtls_config.h
ksdk_mbedtls.c
sha1_alt.h
sha256_alt.h
mmcau_cm4_cm7
mmcau_common
source
startup
utilities
Debug
ksdk_mbedtls_config.h

```
#if defined(FSL_FEATURE_LTC_HAS_GCM) && FSL_FEATURE_LTC_HAS_GCM
#define MBEDTLS_FREESCALE_LTC_AES_GCM /* Enable use of LTC AES GCM.*/
#endif
#if defined(FSL_FEATURE_LTC_HAS_PKHA) && FSL_FEATURE_LTC_HAS_PKHA
#define MBEDTLS_FREESCALE_LTC_PKHA /* Enable use of LTC PKHA.*/
#define FREESCALE_PKHA_INT_MAX_BYTES 256
#endif
#endif

/* Enable MMCAU use in library if there is MMCAU on chip. */
#if defined(FSL_FEATURE_SOC_MMCAU_COUNT) && (FSL_FEATURE_SOC_MMCAU_COUNT > 0)
#include "fsl_mmcau.h"

#define MBEDTLS_FREESCALE_MMCAU_MD5    /* Enable use of MMCAU MD5.*/
#define MBEDTLS_FREESCALE_MMCAU_SHA1   /* Enable use of MMCAU SHA1.*/
#define MBEDTLS_FREESCALE_MMCAU_SHA256 /* Enable use of MMCAU SHA256.*/
#define MBEDTLS_FREESCALE_MMCAU_DES    /* Enable use of MMCAU DES, when LTC is disabled.*/
#define MBEDTLS_FREESCALE_MMCAU_AES    /* Enable use of MMCAU AES, when LTC is disabled.*/
#endif

/* Enable CAU3 use in library if there is CAU3 on chip. */
#if defined(FSL_FEATURE_SOC_CAU3_COUNT) && (FSL_FEATURE_SOC_CAU3_COUNT > 0)
#include "fsl_cau3.h"
#include "cau3_pkha.h"

#define MBEDTLS_CAU3_COMPLETION_SIGNAL CAU3_CC_CMD_EVT
#define MBEDTLS_SHA256_ALT_NO_224
```

mbed TLS



PolarSSL is now part of ARM Official announcement and rebranded as **mbed TLS**.

ARM mbed

Register or Log in to mbed TLS

mbed TLS (formerly known as PolarSSL) makes it trivially easy for developers to include cryptographic and SSL/TLS capabilities in their (embedded) products, facilitating this functionality with a minimal coding footprint.

Easy to use

mbed TLS offers an SSL library with an intuitive API and readable source code, so you can actually understand what the code does. Also the mbed TLS modules are as loosely coupled as possible and written in the portable C language. This allows you to use the parts you need, without having to include the total library. Read more

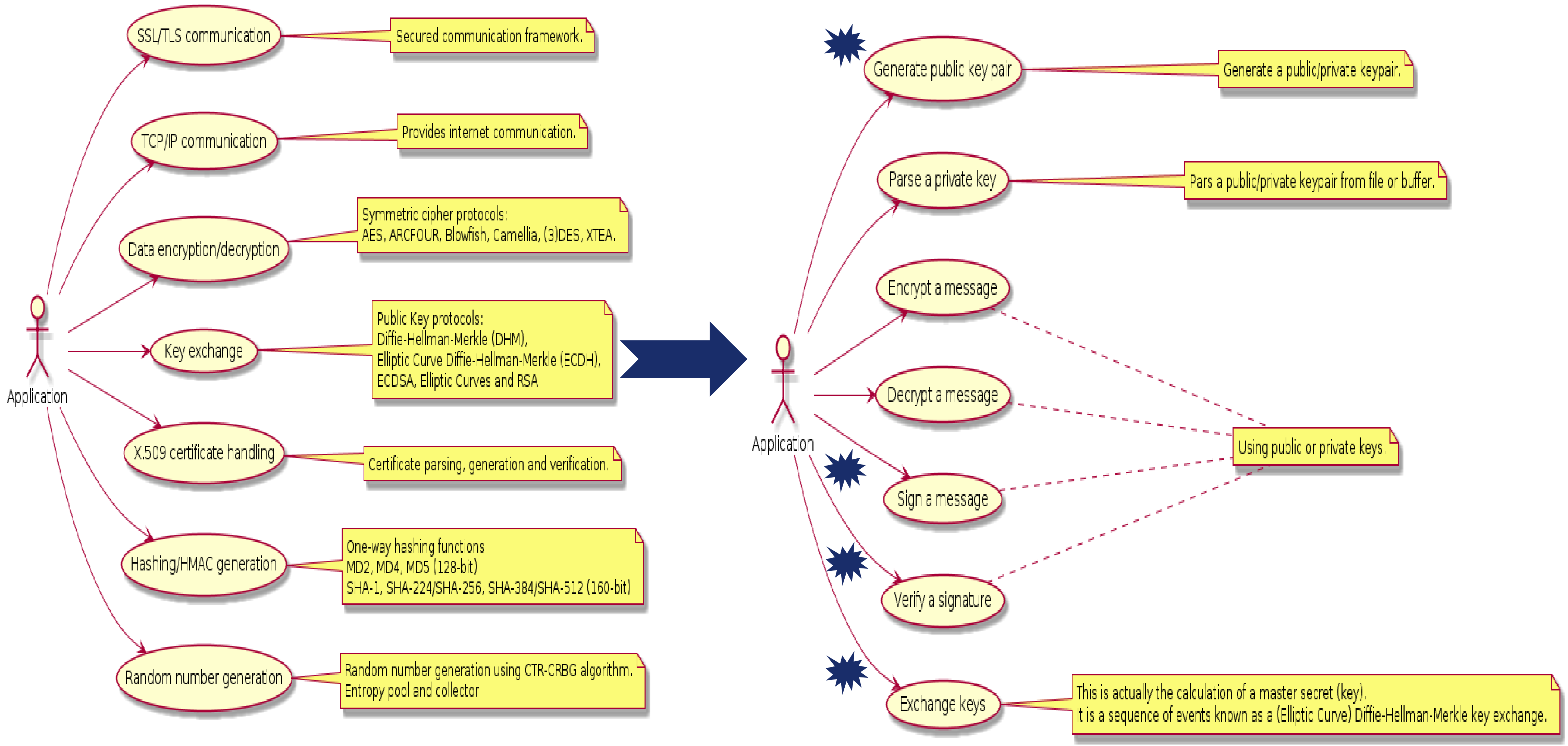
Easy to get

mbed TLS is available as open source under the Apache 2.0 license, the GPL 2.0 license or under an mbed partnership. The Apache 2.0 license enables you to use mbed TLS in both open source and closed source projects. Read more

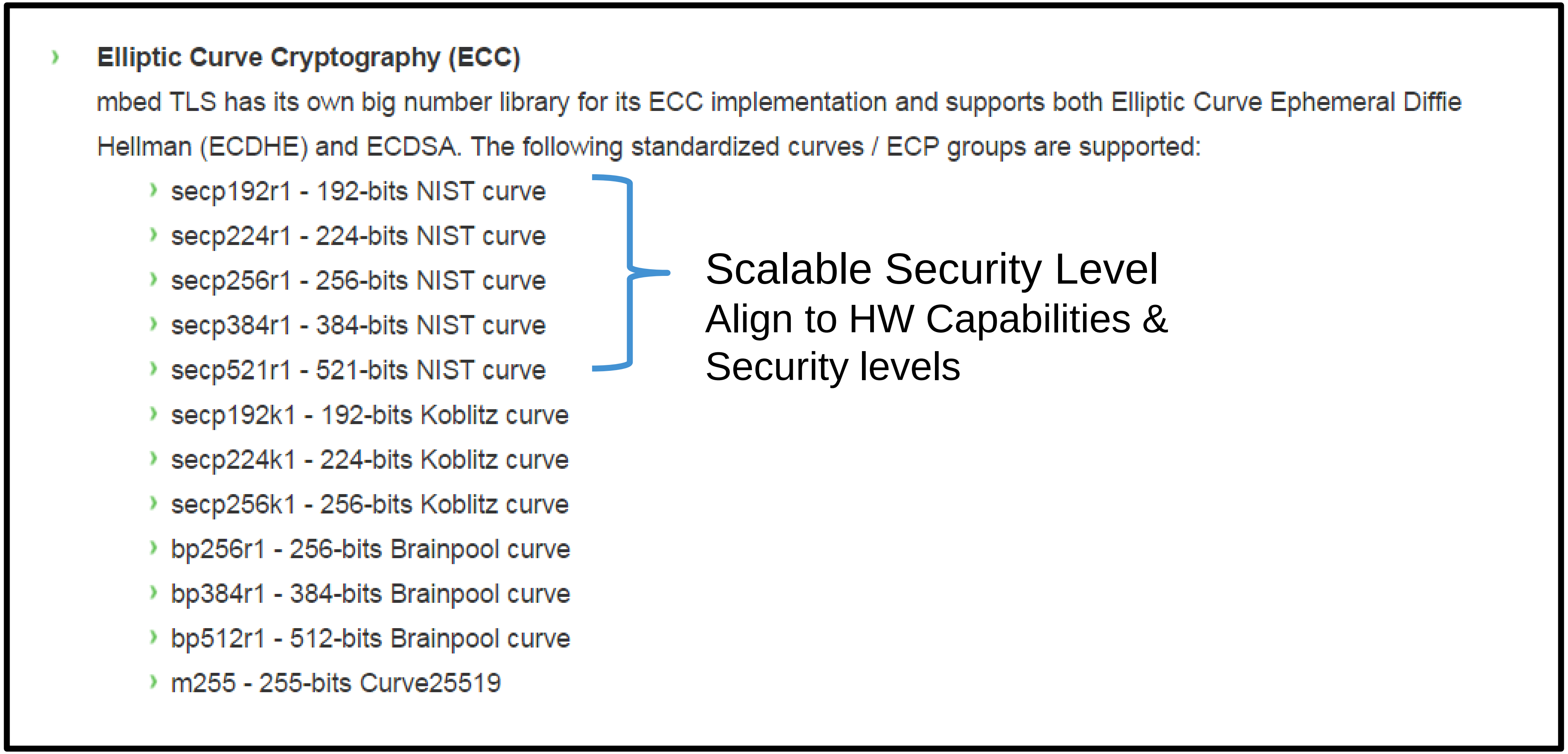
Support

- ✓ Knowledge Base
- ✓ Support Forum
- ✓ Direct e-mail

https://tls.mbed.org/high-level-design
https://tls.mbed.org/module-level-design-public-key


SSL/TLS communication
Secured communication framework.
TCP/IP communication
Provides internet communication.
Data encryption/decryption
Symmetric cipher protocols:
AES, ARCFOUR, Blowfish, Camellia, (3)DES, XTEA.
Key exchange
Public Key protocols:
Diffie-Hellman-Merkle (DHM),
Elliptic Curve Diffie-Hellman-Merkle (ECDH),
ECDSA, Elliptic Curves and RSA
X.509 certificate handling
Certificate parsing, generation and verification.
Hashing/HMAC generation
One-way hashing functions
MD2, MD4, MD5 (128-bit)
SHA-1, SHA-224/SHA-256, SHA-384/SHA-512 (160-bit)
Random number generation
Random number generation using CTR-CRBG algorithm.
Entropy pool and collector
Application
Generate public key pair
Generate a public/private keypair.
Parse a private key
Pars a public/private keypair from file or buffer.
Encrypt a message
Decrypt a message
Sign a message
Verify a signature
Using public or private keys.
Exchange keys
This is actually the calculation of a master secret (key).
It is a sequence of events known as a (Elliptic Curve) Diffie-Hellman-Merkle key exchange.
Application

https://tls.mbed.org/core-features



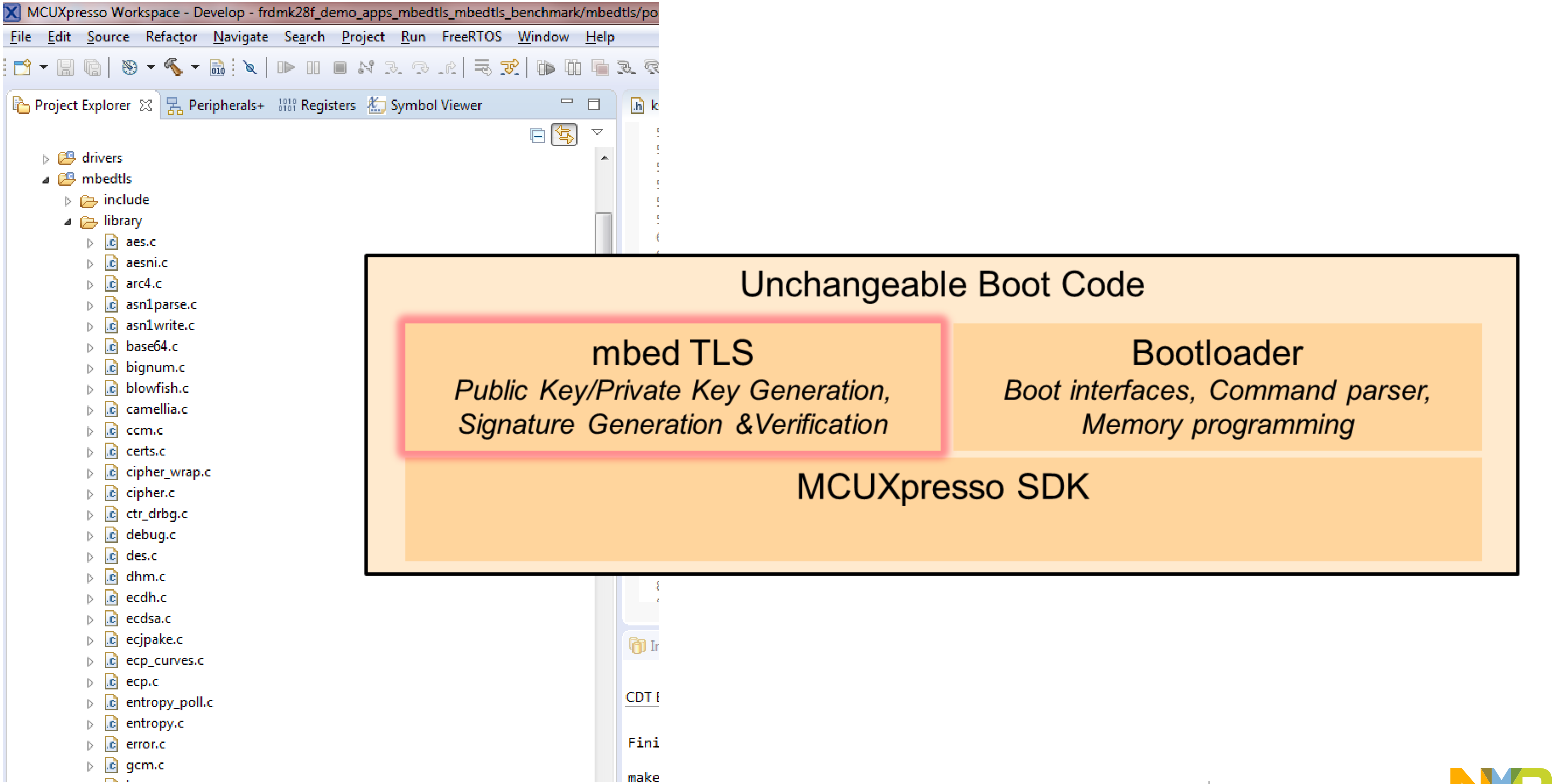
- **Elliptic Curve Cryptography (ECC)**

  mbed TLS has its own big number library for its ECC implementation and supports both Elliptic Curve Ephemeral Diffie Hellman (ECDHE) and ECDSA. The following standardized curves / ECP groups are supported:

  - secp192r1 - 192-bits NIST curve
  - secp224r1 - 224-bits NIST curve
  - secp256r1 - 256-bits NIST curve
  - secp384r1 - 384-bits NIST curve
  - secp521r1 - 521-bits NIST curve
  - secp192k1 - 192-bits Koblitz curve
  - secp224k1 - 224-bits Koblitz curve
  - secp256k1 - 256-bits Koblitz curve
  - bp256r1 - 256-bits Brainpool curve
  - bp384r1 - 384-bits Brainpool curve
  - bp512r1 - 512-bits Brainpool curve
  - m255 - 255-bits Curve25519

Scalable Security Level
Align to HW Capabilities &
Security levels

mbed TLS file Structure Allows Lightweight Implementations



MCUXpresso Workspace - Develop - frdmk28f_demo_apps_mbedtls_mbedtls_benchmark/mbedtls/po
File Edit Source Refactor Navigate Search Project Run FreeRTOS Window Help
Project Explorer
Peripherals+
Registers
Symbol Viewer
CDT
Fini
make

- drivers
- mbedtls
  - include
  - library
    - aes.c
    - aesni.c
    - arc4.c
    - asn1parse.c
    - asn1write.c
    - base64.c
    - bignum.c
    - blowfish.c
    - camellia.c
    - ccm.c
    - certs.c
    - cipher_wrap.c
    - cipher.c
    - ctr_drbg.c
    - debug.c
    - des.c
    - dhm.c
    - ecdh.c
    - ecdsa.c
    - ecjpake.c
    - ecp_curves.c
    - ecp.c
    - entropy_poll.c
    - entropy.c
    - error.c
    - gcm.c

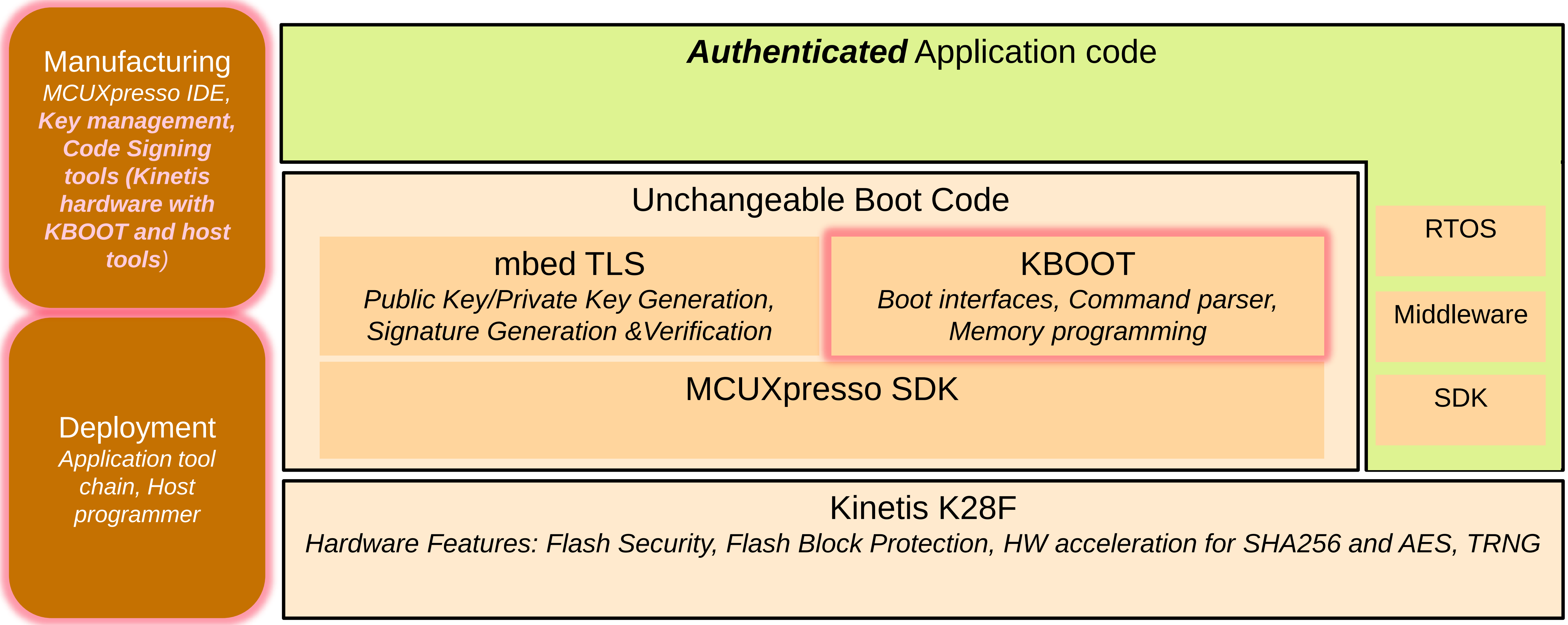
Unchangeable Boot Code

mbed TLS
Public Key/Private Key Generation, Signature Generation &Verification

Bootloader
Boot interfaces, Command parser, Memory programming

MCUXpresso SDK

Security Technology | System View: Bootloader and Tools



Manufacturing
MCUXpresso IDE,
Key management, Code Signing tools (Kinetis hardware with KBOOT and host tools)

Deployment
Application tool chain, Host programmer

Authenticated Application code

Unchangeable Boot Code

mbed TLS
Public Key/Private Key Generation, Signature Generation &Verification

KBOOT
Boot interfaces, Command parser, Memory programming

MCUXpresso SDK

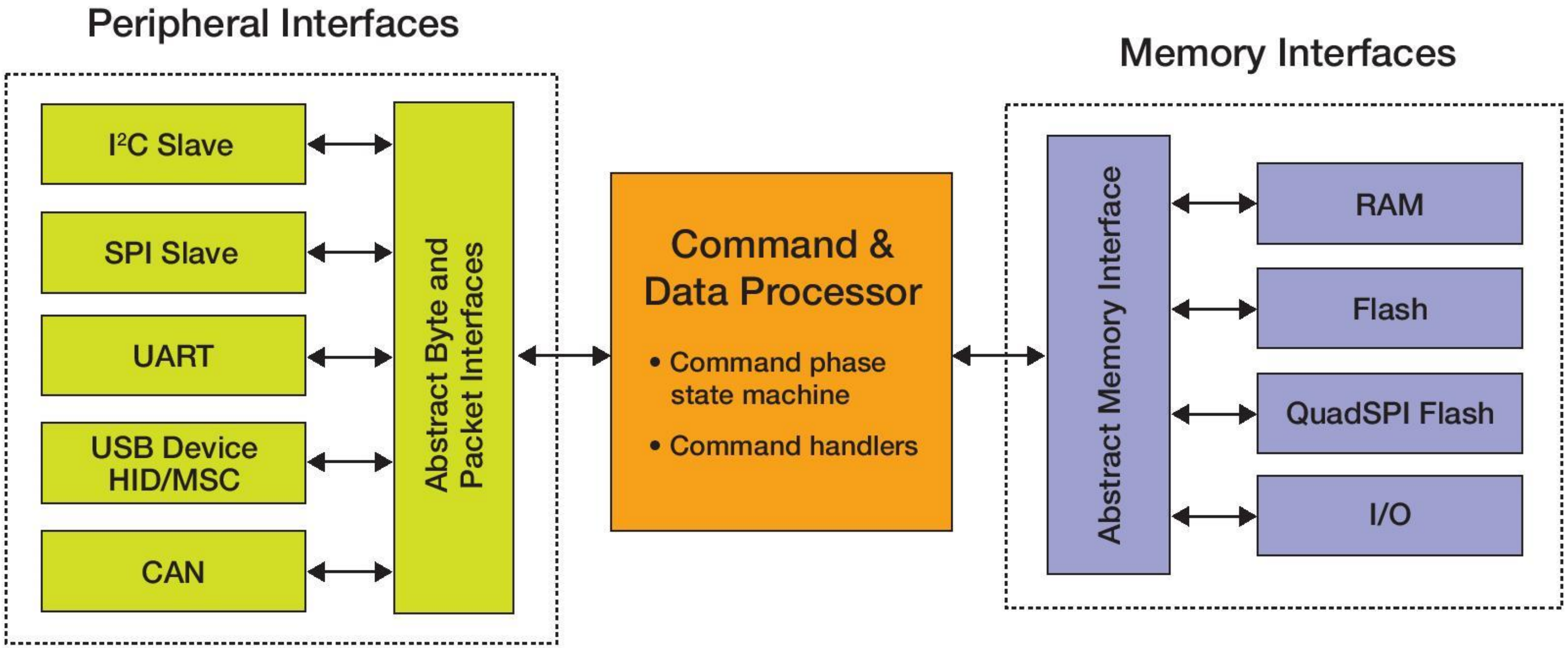
RTOS

Middleware

SDK

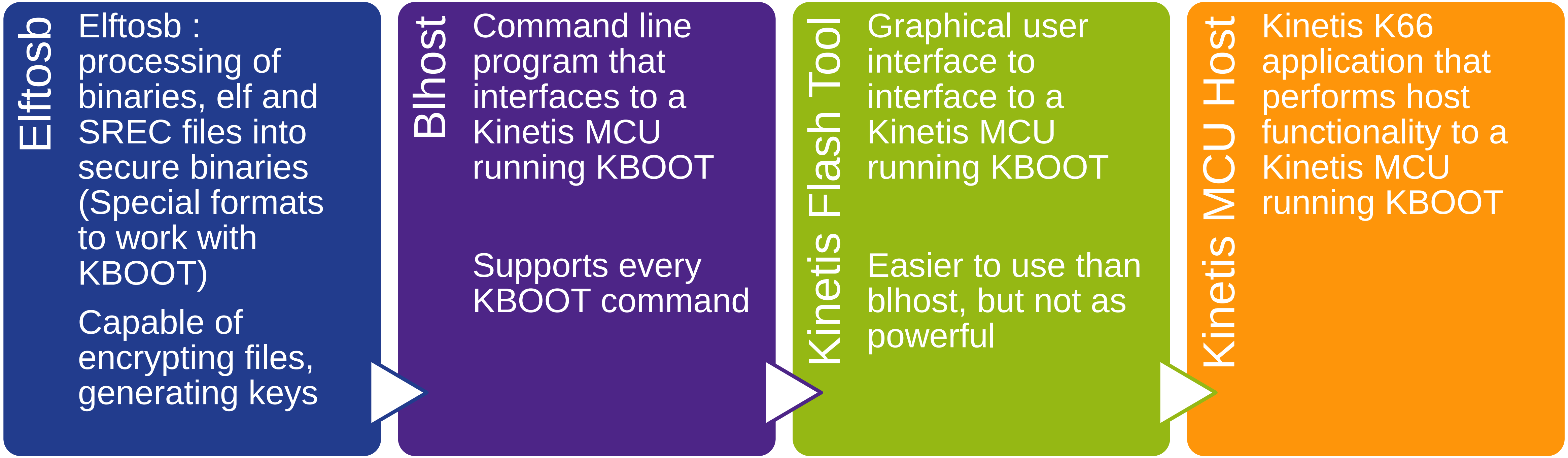
Kinetis K28F
Hardware Features: Flash Security, Flash Block Protection, HW acceleration for SHA256 and AES, TRNG

KBOOT: Kinetis Bootloader



HOST TOOLS: Kinetis Flash Tool, blhost, elftosb, Kinetis MCU Host

HOST TOOLS: Kinetis Flash Tool, blhost, elftosb, Kinetis MCU Host



Elftosb

Elftosb : processing of binaries, elf and SREC files into secure binaries (Special formats to work with KBOOT)

Capable of encrypting files, generating keys

Blhost

Command line program that interfaces to a Kinetis MCU running KBOOT

Supports every KBOOT command

Kinetis Flash Tool

Graphical user interface to interface to a Kinetis MCU running KBOOT

Easier to use than blhost, but not as powerful

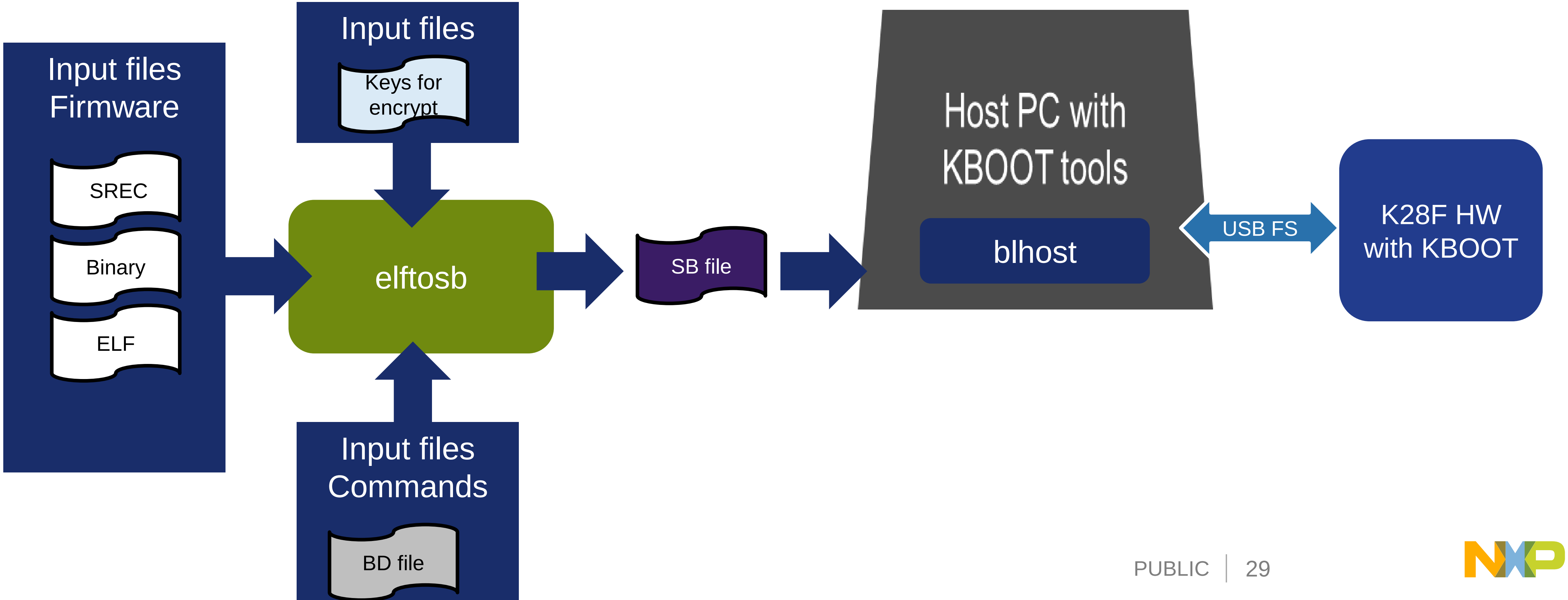
Kinetis MCU Host

Kinetis K66 application that performs host functionality to a Kinetis MCU running KBOOT

The Elftosb and blhost tool is command line driven and can be separately built to run on Windows® OS, Linux® OS, and Apple Mac® OS.

KBOOT Definitions and Use

- **BD file:** Short for boot descriptor file. This is an input command file to be used by elftosb for created SB files
- **SB file**: Short for secure binary file. This is the output of elftosb which is used to pass commands and data to a Kinetis MCU running KBOOT



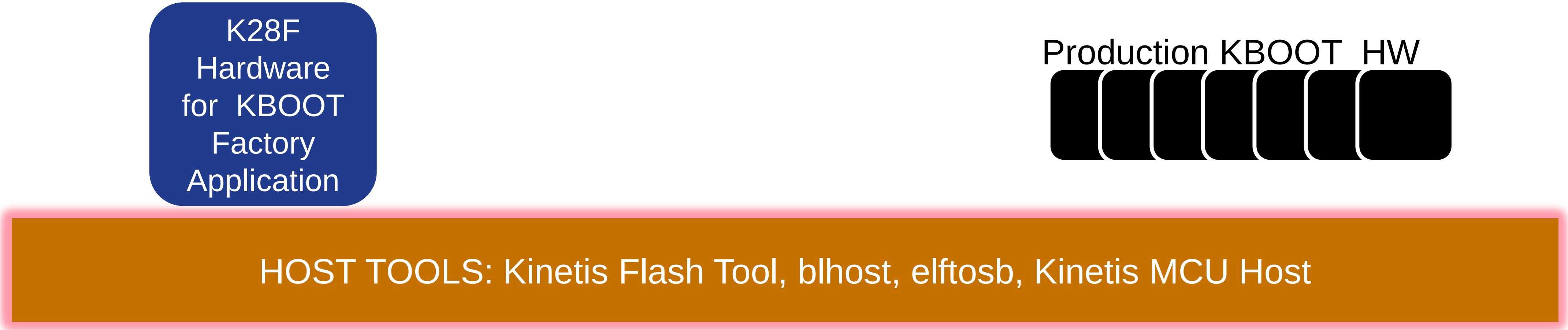
4

Overview of Methods

Using KBOOT for Signature Generation

- Factory KBOOT application
  - This bootloader application is for use in a secure manufacturing environment. The main security functions in addition to bootloader functions are to generate a PUB/PRIV key pair and to generate the signature for application code using the **private key**.

- Production KBOOT application
  - This bootloader application is for use in a deployed device. The main security functions in addition to bootloader functions are to check the signature of application code using the **public key**, and only allow execution of the application code if the signature is authentic.



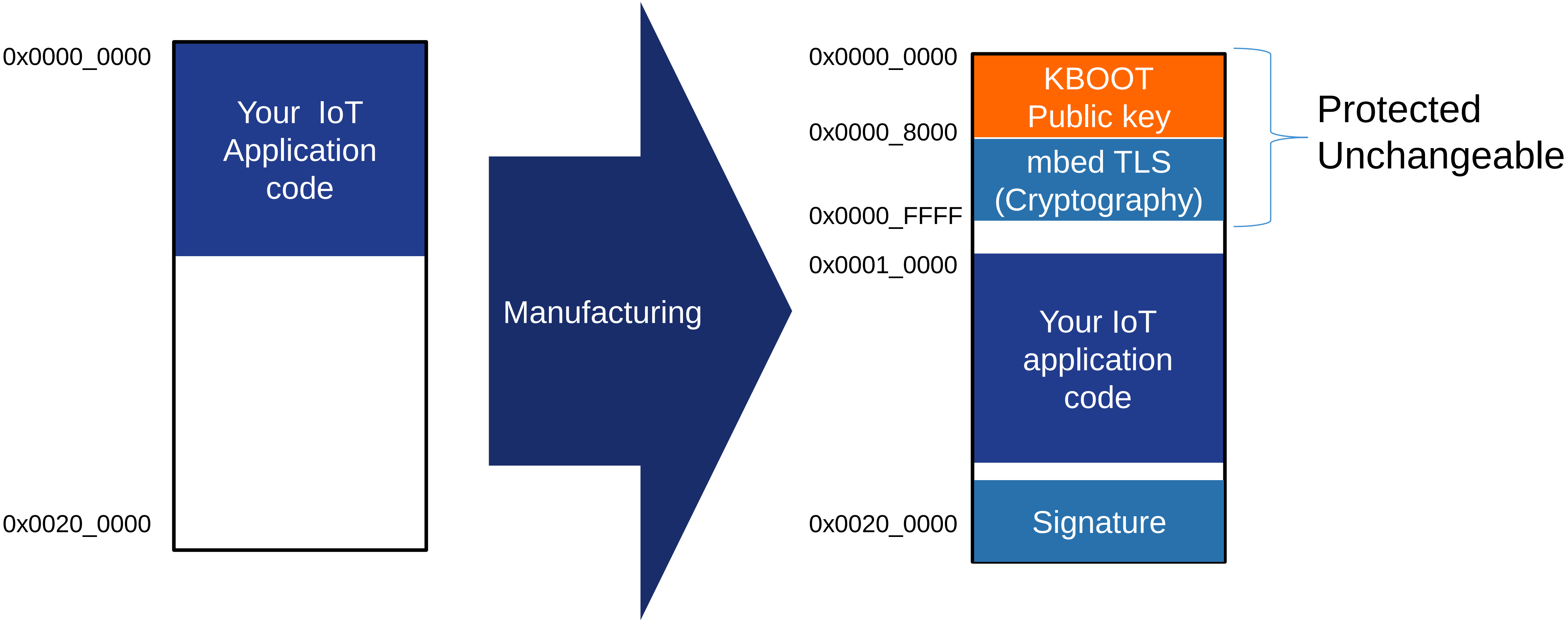
K28F Hardware for KBOOT Factory Application

Production KBOOT HW

HOST TOOLS: Kinetis Flash Tool, blhost, elftosb, Kinetis MCU Host

Overview of the Method

- Typical Application Development
- Final production image



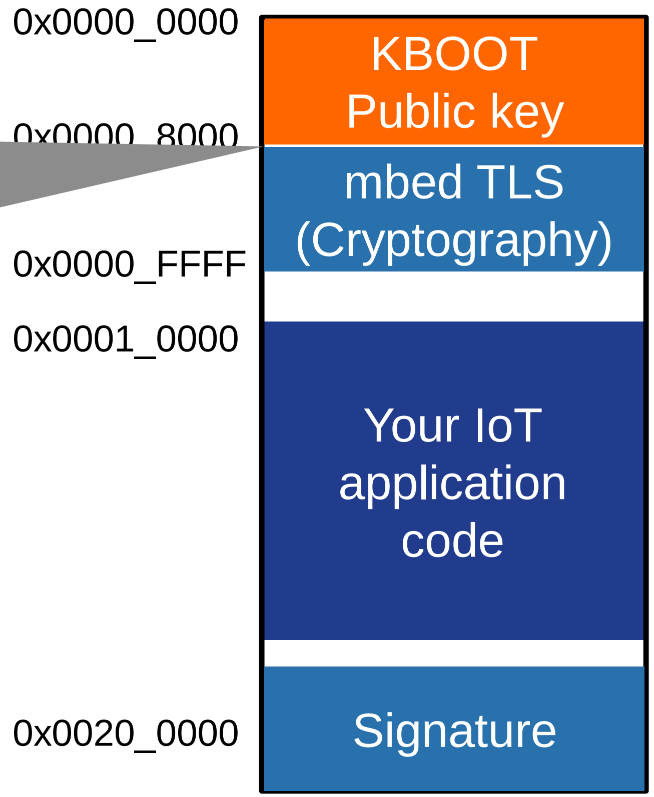
Overview of the Method

- Typical Application Development

KBOOT : Secure Boot

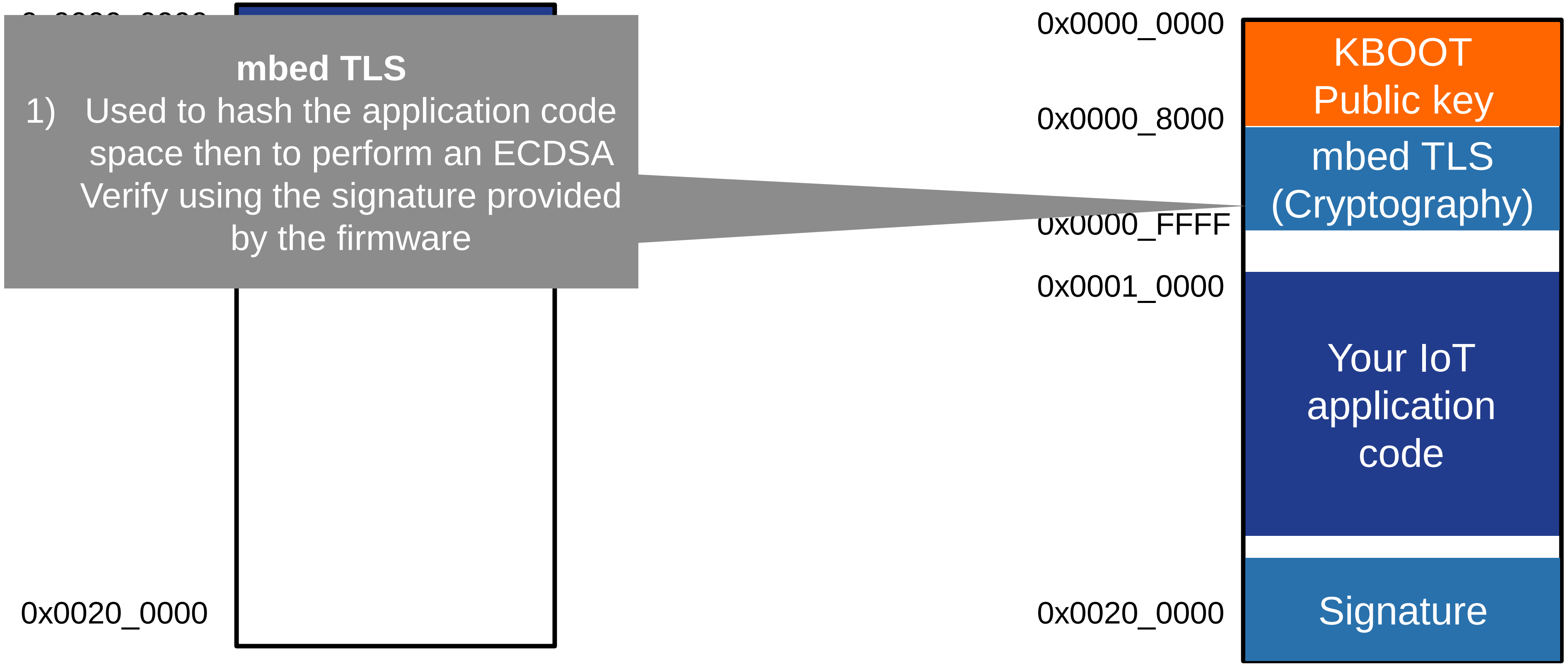
1) Always runs after chip reset and checks defined interfaces (ie. USB) for host connection to get new firmware
    - Application code authentication is applied before allowing application to run
    - Protected by chip HW mechanisms, can be made immutable

- Final production image



Overview of the Method

- Typical Application Development
- Final production image

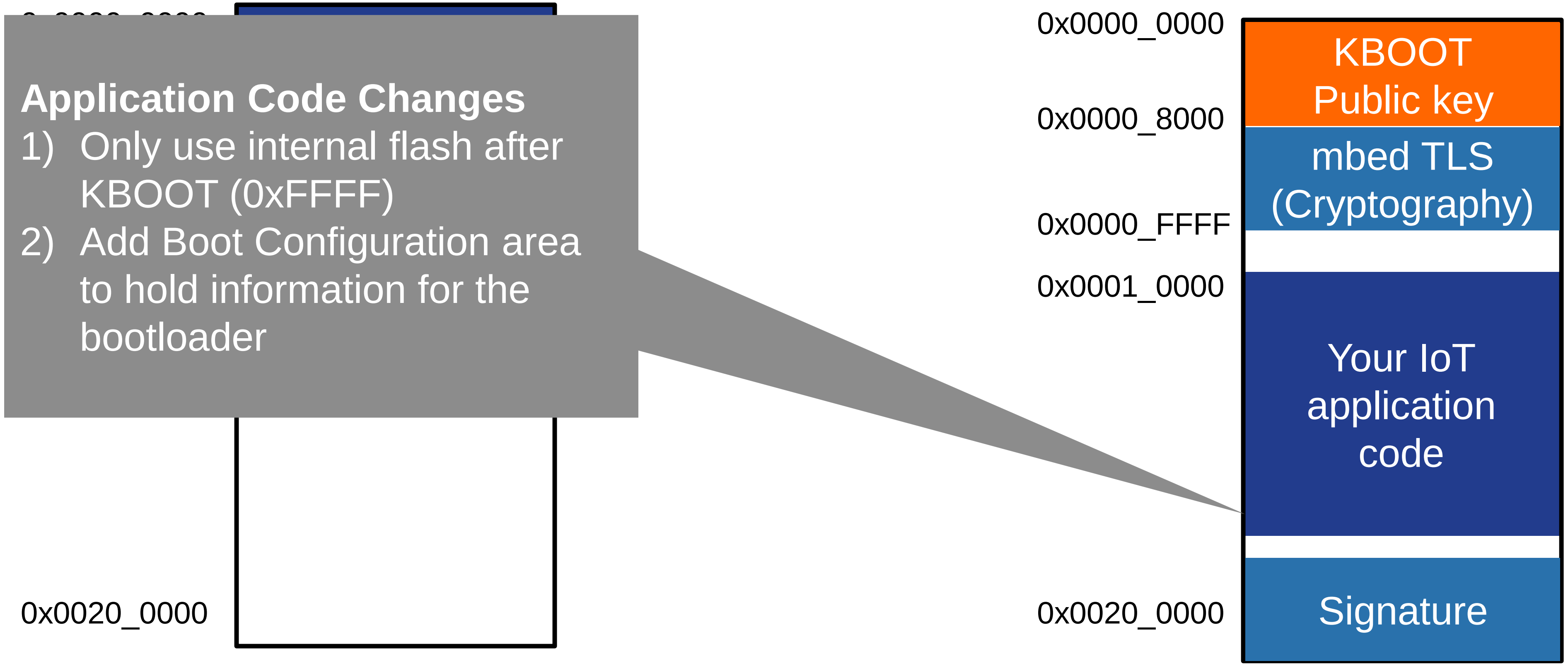


mbed TLS

1) Used to hash the application code space then to perform an ECDSA Verify using the signature provided by the firmware

Overview of the Method

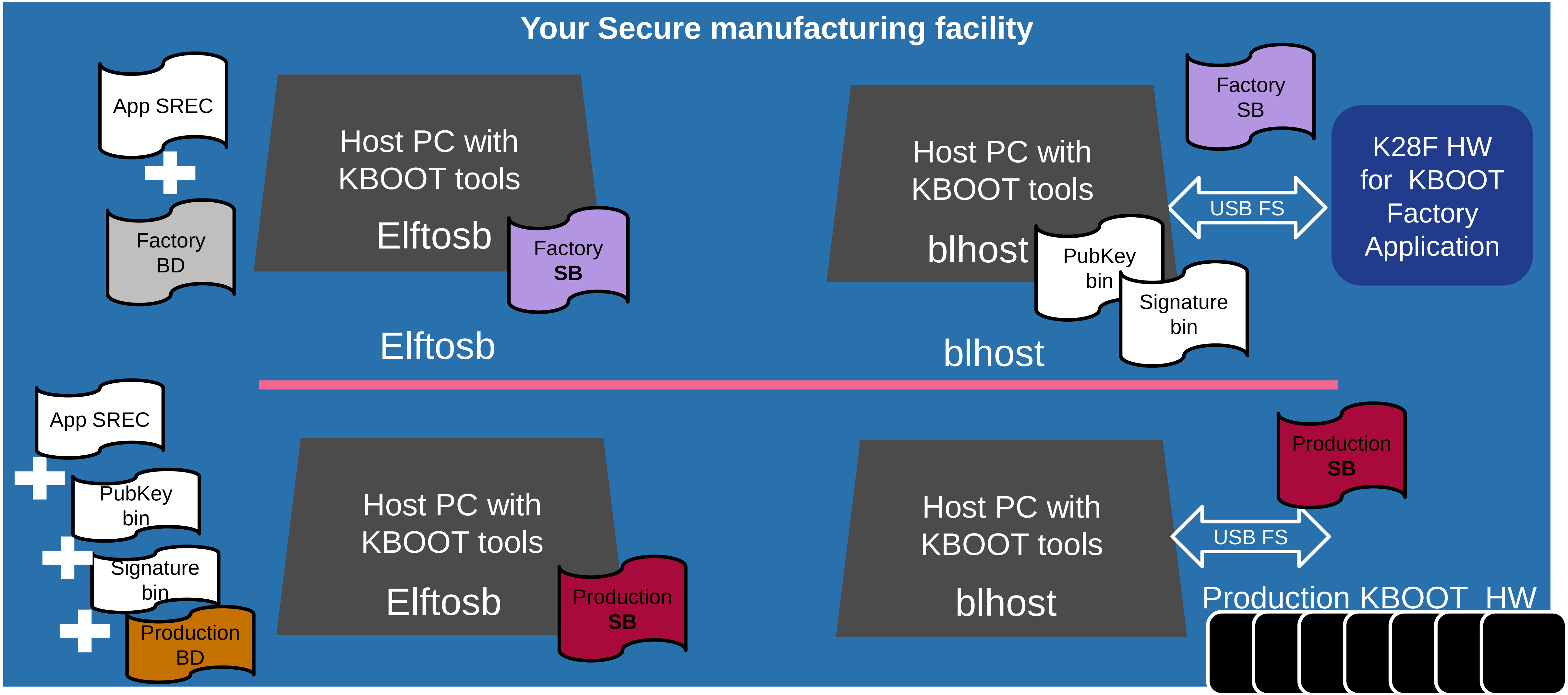
- Typical Application Development
- Final production image



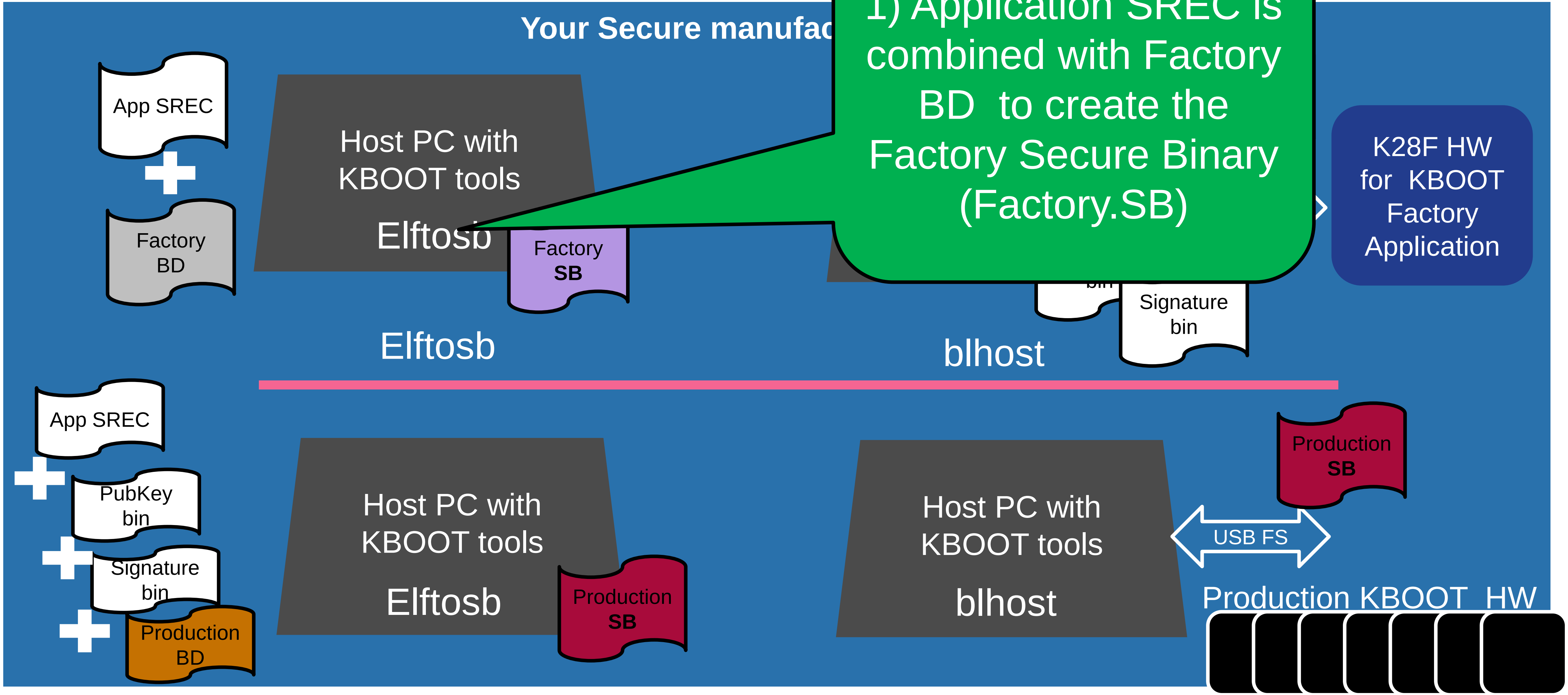
Application Code Changes

1) Only use internal flash after KBOOT (0xFFFF)
2) Add Boot Configuration area to hold information for the bootloader

Using KBOOT for Signature Generation

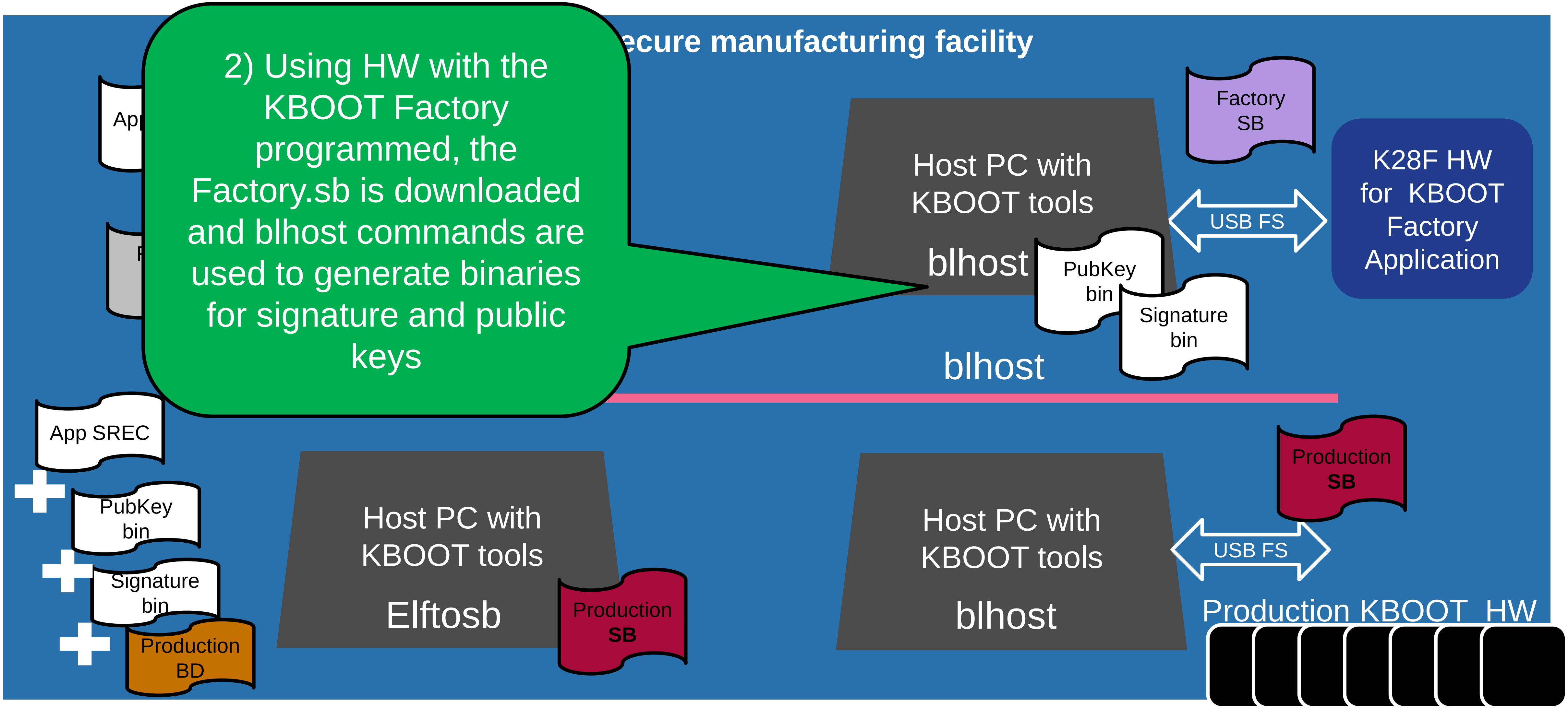


Using KBOOT for Signature Genera



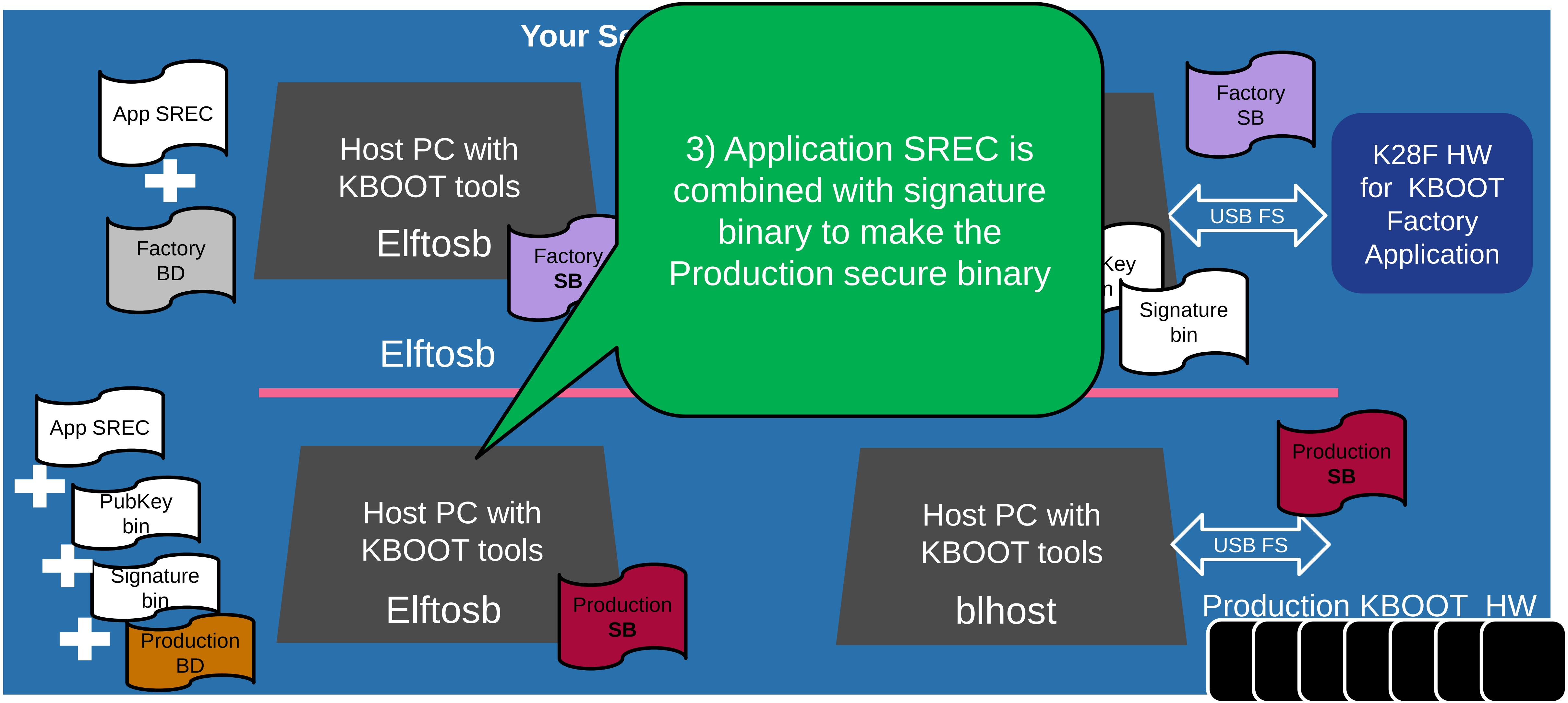
1) Application SREC is combined with Factory BD to create the Factory Secure Binary (Factory.SB)

Using KBOOT for Signature Generation



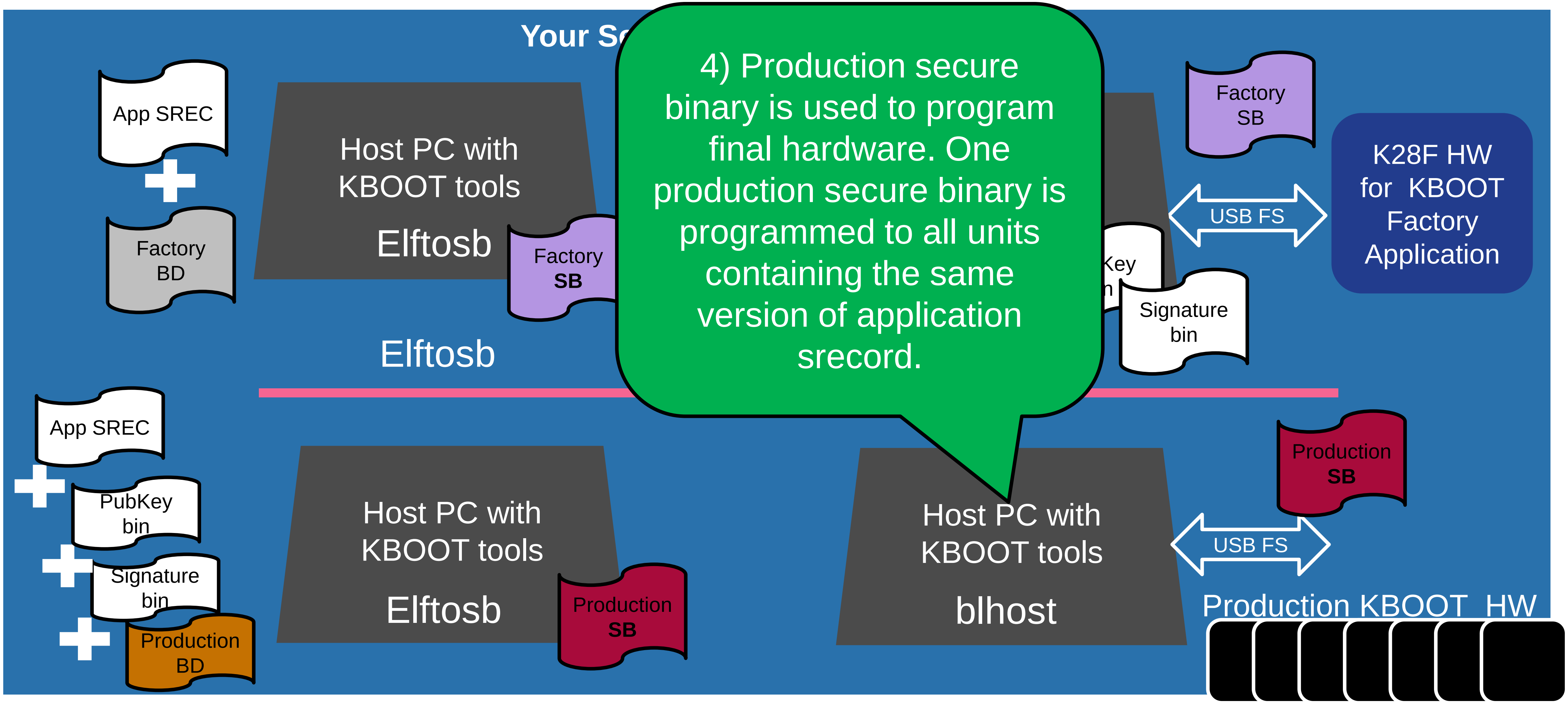
2) Using HW with the KBOOT Factory programmed, the Factory.sb is downloaded and blhost commands are used to generate binaries for signature and public keys

Using KBOOT for Signature Generation



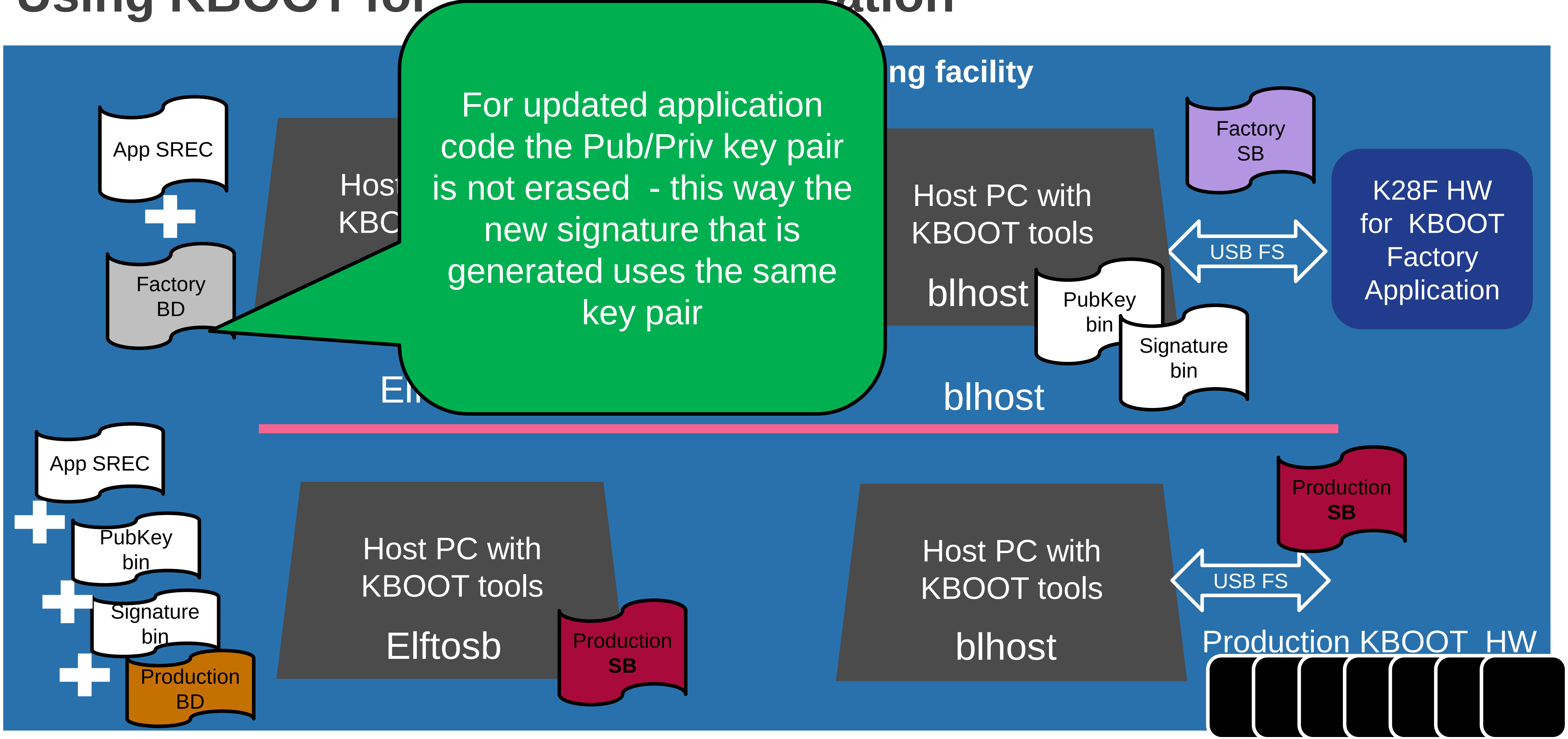
3) Application SREC is combined with signature binary to make the Production secure binary

Using KBOOT for Signature Generation



4) Production secure binary is used to program final hardware. One production secure binary is programmed to all units containing the same version of application srecord.

Using KBOOT for Signature Generation



For updated application code the Pub/Priv key pair is not erased - this way the new signature that is generated uses the same key pair

5

Development Steps

Overview of Development Steps for K28F KBOOT

1. Port KBOOT for K28F
   - Porting guidelines are provided in the KBOOT reference Manual Chapter 10
     - K66F is the starting point
   - File renaming and copying over from SDK of K28F
   - Account for HW differences
     - LPUART versus standard UART
2. Add mbed TLS support to KBOOT for cryptography
   - Add relevant files
     - SHA-256, ECC, ECDSA
3. Defines are used to use one application which can be configured for factory mode or production mode

- **Development Environments**
  - KDS is used for KBOOT development
    - Other tool chains are available and on the roadmap
    - Could be ported to MCUXpesso
  - PC with KBOOT tools is used for factory signing and initial provisioning

Overview of Development Steps for K28F Application

1. Application development changes when starting from K28F SDK
   - Update Linker File
     - Code must be placed after KBOOT
     - New range from 0x3C0 to 0x400 for BCA (boot config area)

- **Development Environments**
  - MCUXpresso with SDK

5

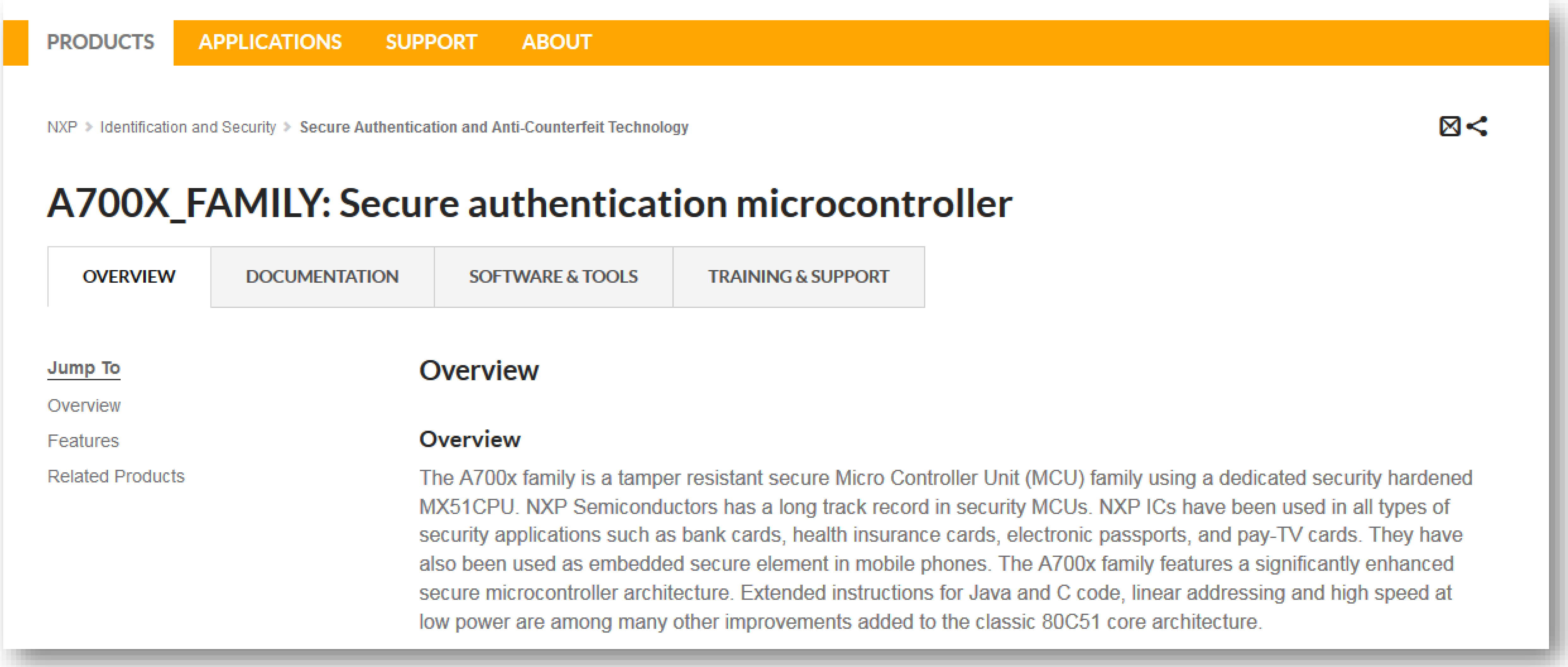
Key Management Options

Cryptography Key Table

Key Name or Description	Key Type	Key Location(s)	Comments
Private Key Enc. Key	Simple Xor with key	Factory.bd text file (calls 2 parameter as simple private key enc key)	Used in SEC Kboot Factory to output the private key after encryption. (To improvement to use AES CBC enc this key.)
Private Key for KBOOT	ECDSA-BP256	On-chip flash 0x0003_f000 only at factory mode	RAM of SEC Boot Factory, *encrypted* and stored externally
Public Key for KBOOT	ECDSA-BP256	Included in the product bootloader image at compiling. No fixed address.	Exported to binary by factory bootloader, stored in production boot code
Signature of application firmware	SHA-256 Based for ECDSA BP256	On-chip flash 0x001F_ff80, after on chip application image.	

Private key must be protected, and a secure manufacturing environment is needed

Alternative Key Management with embedded secure element



PRODUCTS
APPLICATIONS
SUPPORT
ABOUT
NXP > Identification and Security > Secure Authentication and Anti-Counterfeit Technology

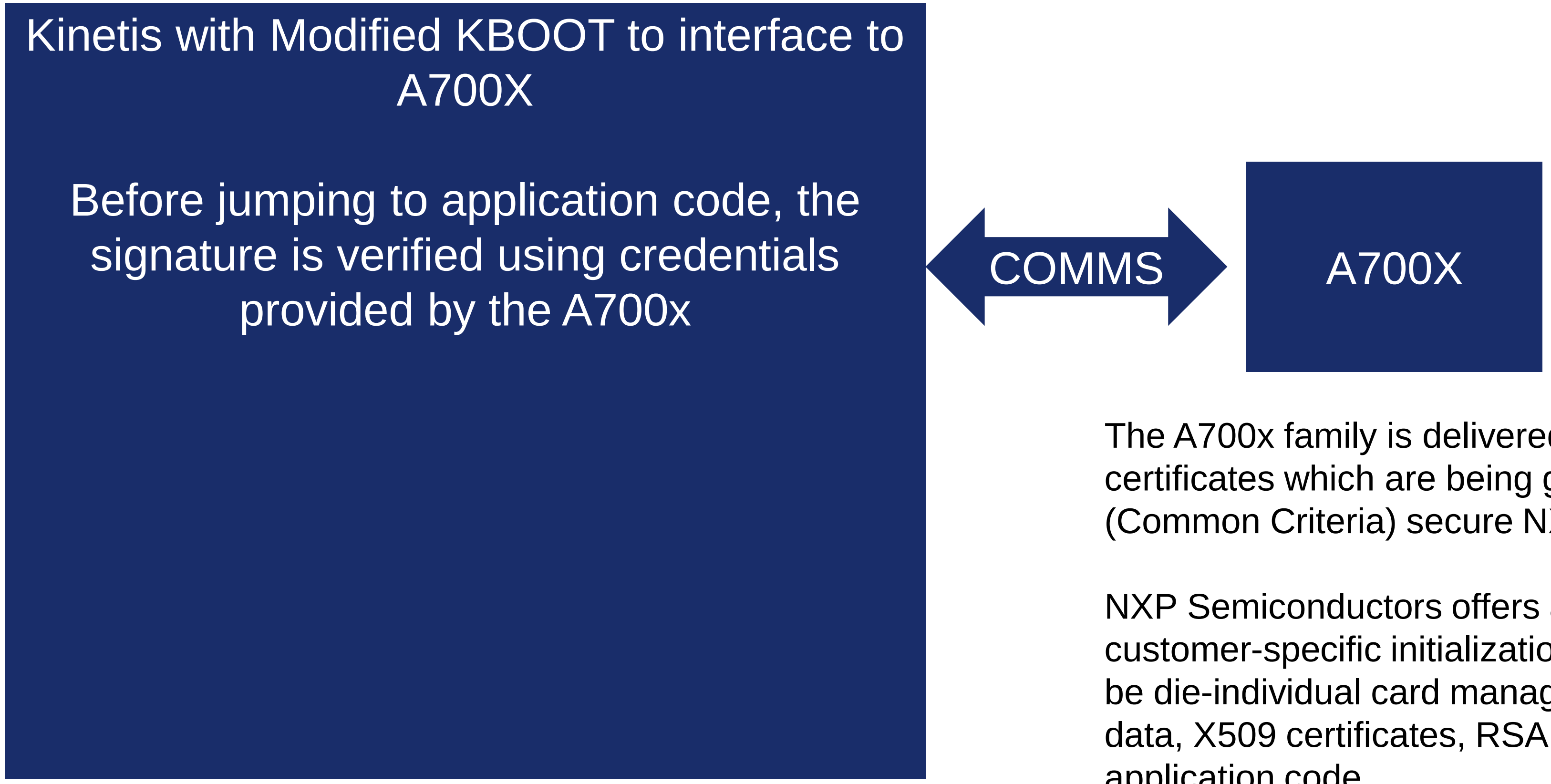
A700X_FAMILY: Secure authentication microcontroller

Overview

Overview

The A700x family is a tamper resistant secure Micro Controller Unit (MCU) family using a dedicated security hardened MX51CPU. NXP Semiconductors has a long track record in security MCUs. NXP ICs have been used in all types of security applications such as bank cards, health insurance cards, electronic passports, and pay-TV cards. They have also been used as embedded secure element in mobile phones. The A700x family features a significantly enhanced secure microcontroller architecture. Extended instructions for Java and C code, linear addressing and high speed at low power are among many other improvements added to the classic 80C51 core architecture.

A700X with KBOOT



Kinetis with Modified KBOOT to interface to A700X

Before jumping to application code, the signature is verified using credentials provided by the A700x

COMMS

A700X

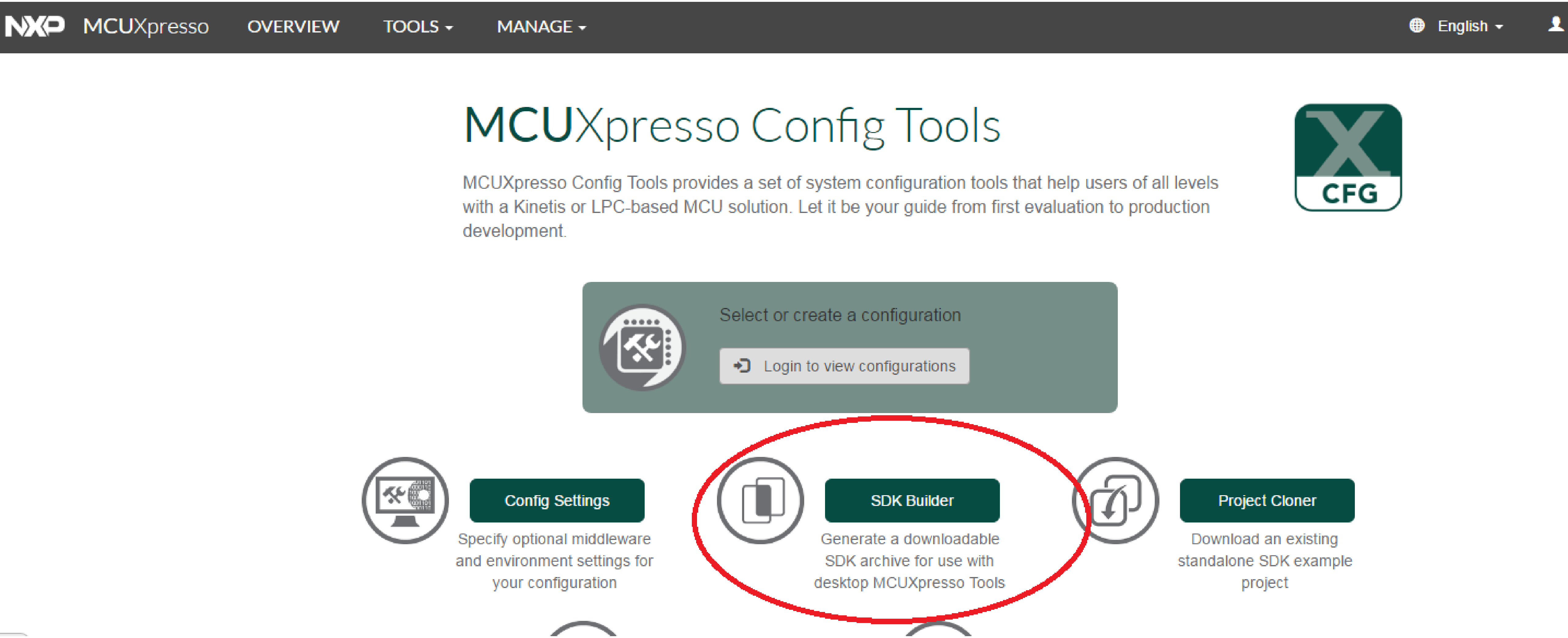
The A700x family is delivered with pre-programmed, die-specific keys and certificates which are being generated and programmed in a certified (Common Criteria) secure NXP internal environment

NXP Semiconductors offers a pre-personalizations service where customer-specific initialization data can be preprogrammed. This data can be die-individual card manager keys, symmetric DES-or AES keys, random data, X509 certificates, RSA signing keys or any other constant data like application code.

6

Resources and Next Steps

https://mcuxpresso.nxp.com/en/welcome



MCUXpresso Config Tools

MCUXpresso Config Tools provides a set of system configuration tools that help users of all levels with a Kinetis or LPC-based MCU solution. Let it be your guide from first evaluation to production development.

Select or create a configuration

Login to view configurations

Config Settings

Specify optional middleware and environment settings for your configuration

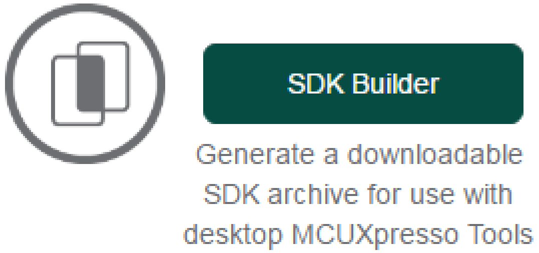
SDK Builder

Generate a downloadable SDK archive for use with desktop MCUXpresso Tools

Project Cloner

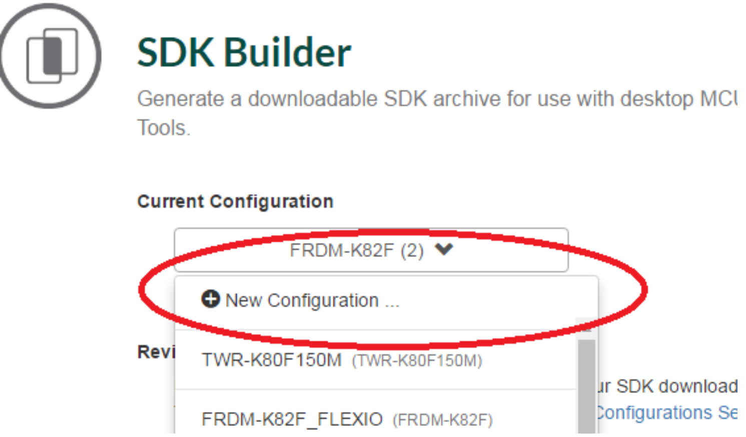
Download an existing standalone SDK example project

Downloading SDK with mbed TLS



SDK Builder

Generate a downloadable SDK archive for use with desktop MCUXpresso Tools



SDK Builder

Generate a downloadable SDK archive for use with desktop MCU
Tools.

Current Configuration

FRDM-K82F (2)

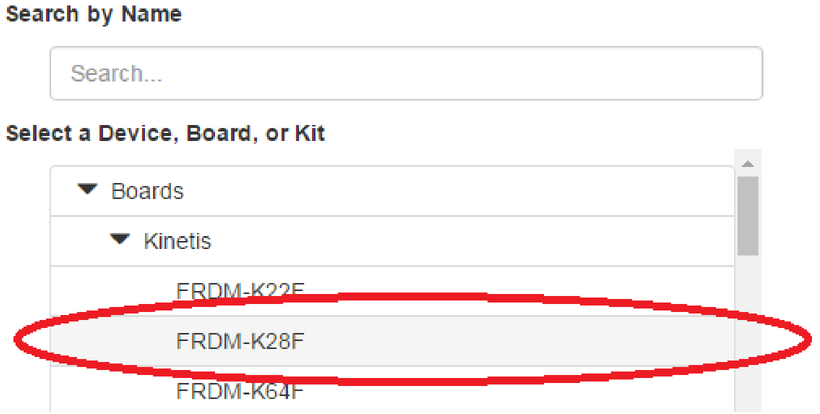
New Configuration ...

TWR-K80F150M (TWR-K80F150M)

FRDM-K82F_FLEXIO (FRDM-K82F)

Revi

ır SDK download
Configurations Se

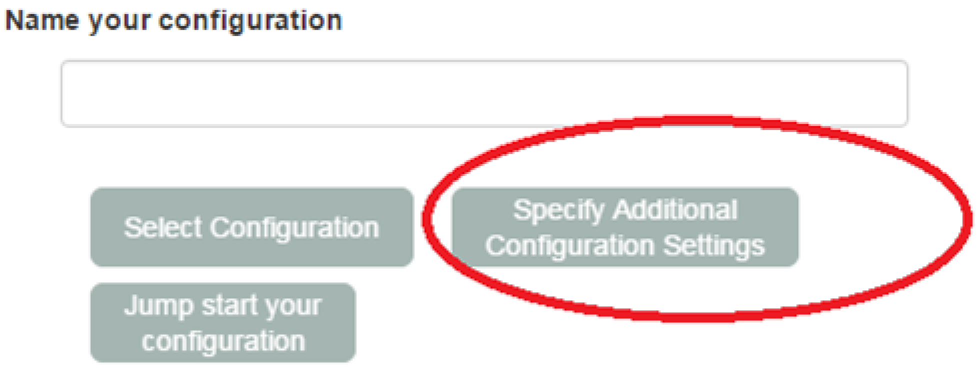


Search by Name

Search...

Select a Device, Board, or Kit

- Boards
  - Kinetis
    - FRDM-K22F
    - FRDM-K28F
    - FRDM-K64F

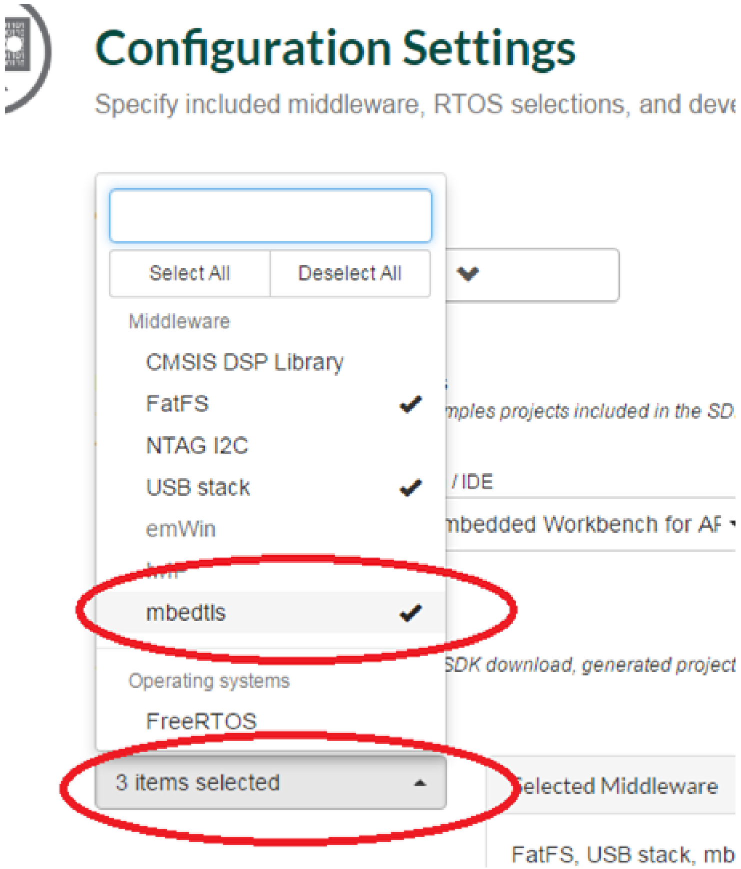


Name your configuration

Select Configuration

Specify Additional Configuration Settings

Jump start your configuration



Configuration Settings

Specify included middleware, RTOS selections, and deve

Select All | Deselect All

Middleware

- CMSIS DSP Library
- FatFS ✓
- NTAG I2C
- USB stack ✓
- emWin
- lwip
- mbedtls ✓

Operating systems

- FreeRTOS

3 items selected

mples projects included in the SD

/ IDE

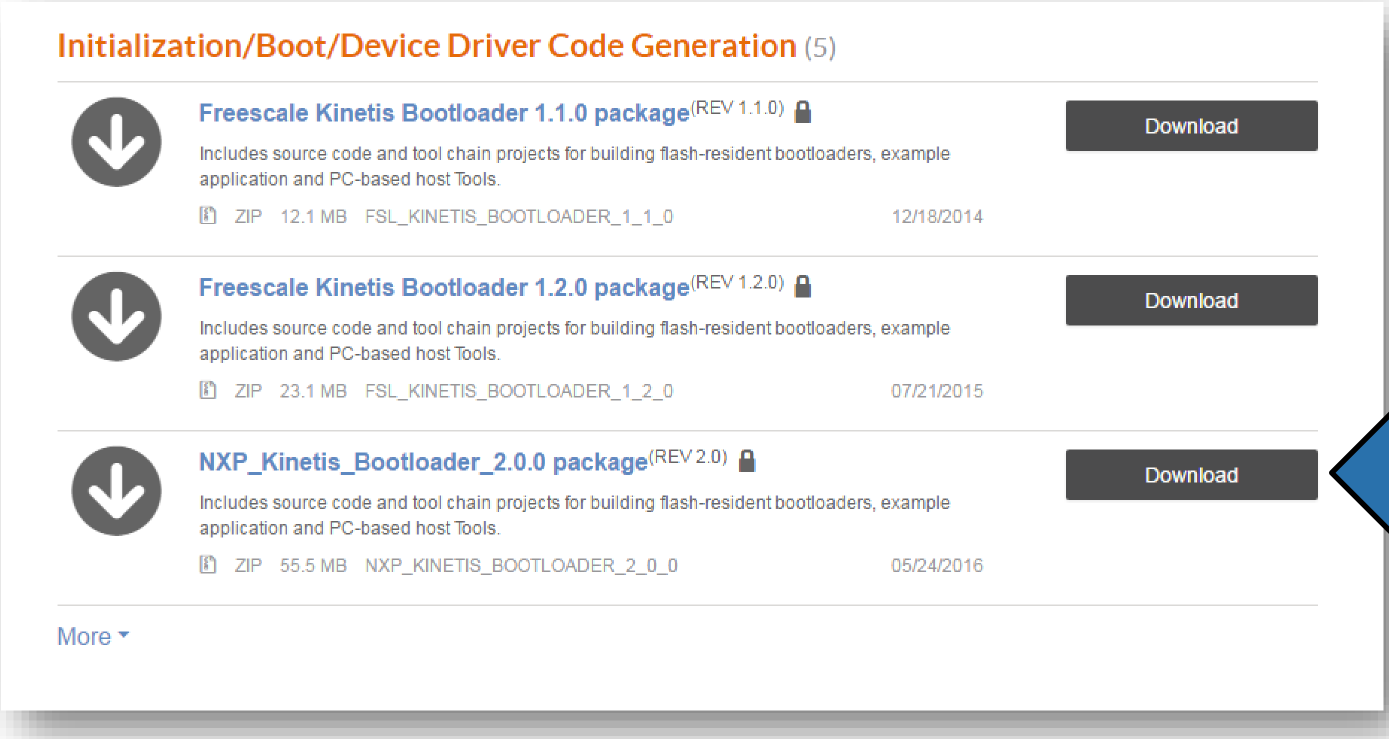
mbedded Workbench for AF

SDK download, generated project

elected Middleware

FatFS, USB stack, mb

LINK: Download KBOOT 2.0 Package



Initialization/Boot/Device Driver Code Generation (5)

Freescale Kinetis Bootloader 1.1.0 package(REV 1.1.0)

Includes source code and tool chain projects for building flash-resident bootloaders, example application and PC-based host Tools.

ZIP 12.1 MB FSL_KINETIS_BOOTLOADER_1_1_0 12/18/2014

Download

Freescale Kinetis Bootloader 1.2.0 package(REV 1.2.0)

Includes source code and tool chain projects for building flash-resident bootloaders, example application and PC-based host Tools.

ZIP 23.1 MB FSL_KINETIS_BOOTLOADER_1_2_0 07/21/2015

Download

NXP_Kinetis_Bootloader_2.0.0 package(REV 2.0)

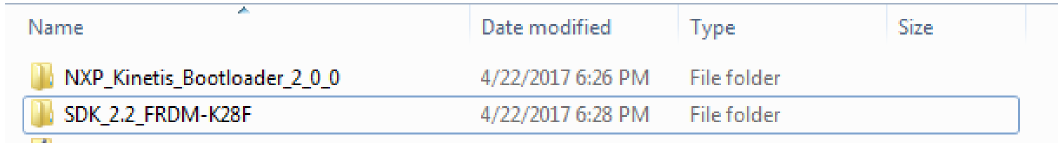
Includes source code and tool chain projects for building flash-resident bootloaders, example application and PC-based host Tools.

ZIP 55.5 MB NXP_KINETIS_BOOTLOADER_2_0_0 05/24/2016

Download

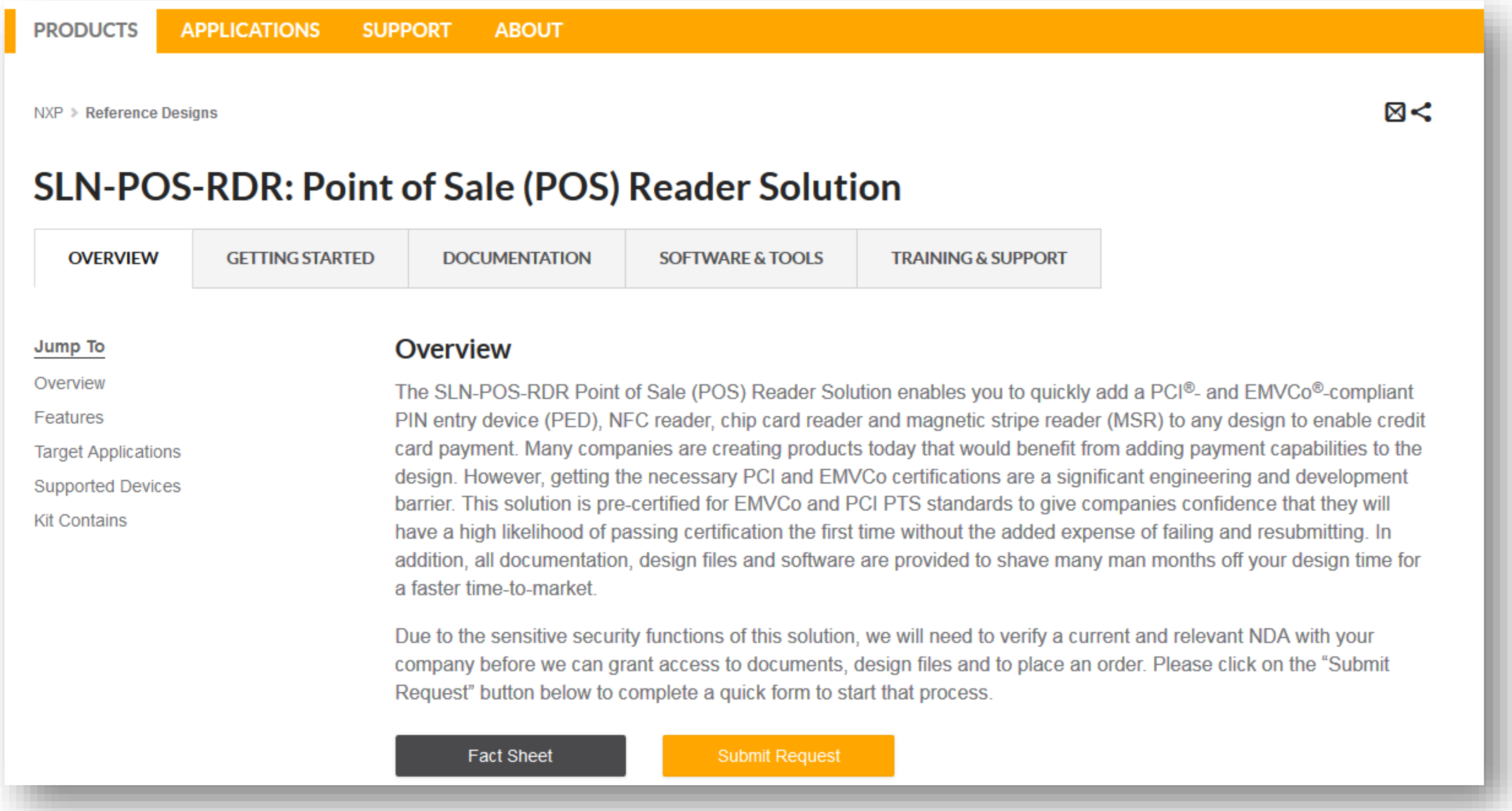
More

Extract the zip file to create \NXP_Kinetis_Bootloader_2_0_0 and SDK for K28F



Name	Date modified	Type	Size
NXP_Kinetis_Bootloader_2_0_0	4/22/2017 6:26 PM	File folder	
SDK_2.2_FRDM-K28F	4/22/2017 6:28 PM	File folder	

SLN-POS-RDR – Secure Card Reader Solution



NXP > Reference Designs

SLN-POS-RDR: Point of Sale (POS) Reader Solution

Jump To

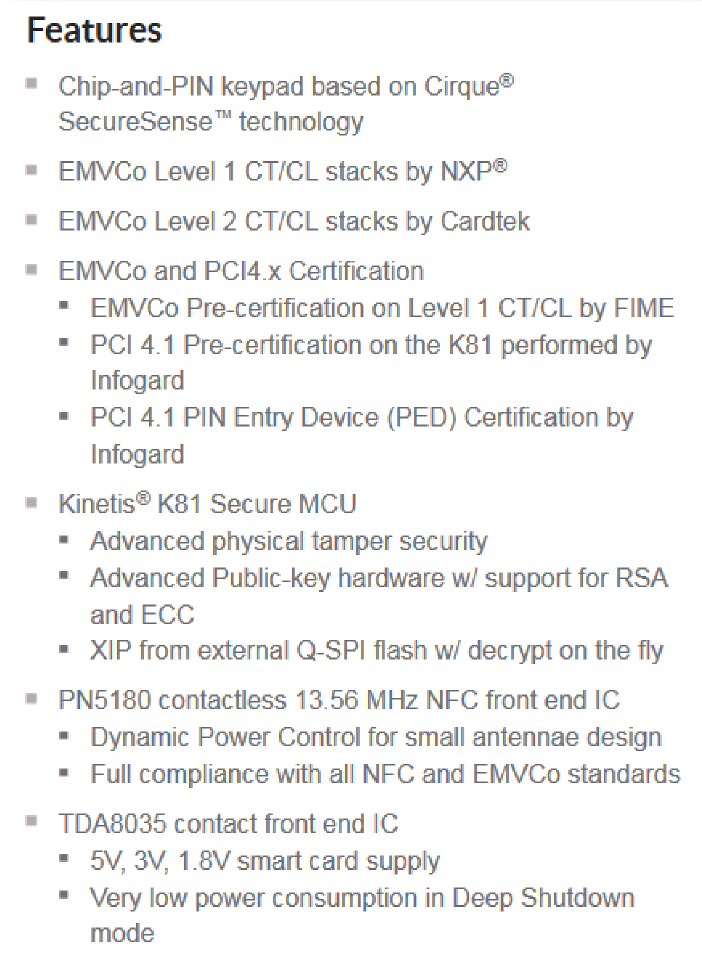
- Overview
- Features
- Target Applications
- Supported Devices
- Kit Contains

Overview

The SLN-POS-RDR Point of Sale (POS) Reader Solution enables you to quickly add a PCI®- and EMVCo®-compliant PIN entry device (PED), NFC reader, chip card reader and magnetic stripe reader (MSR) to any design to enable credit card payment. Many companies are creating products today that would benefit from adding payment capabilities to the design. However, getting the necessary PCI and EMVCo certifications are a significant engineering and development barrier. This solution is pre-certified for EMVCo and PCI PTS standards to give companies confidence that they will have a high likelihood of passing certification the first time without the added expense of failing and resubmitting. In addition, all documentation, design files and software are provided to shave many man months off your design time for a faster time-to-market.

Due to the sensitive security functions of this solution, we will need to verify a current and relevant NDA with your company before we can grant access to documents, design files and to place an order. Please click on the “Submit Request” button below to complete a quick form to start that process.

Fact Sheet
Submit Request



Features

- Chip-and-PIN keypad based on Cirque® SecureSense™ technology
- EMVCo Level 1 CT/CL stacks by NXP®
- EMVCo Level 2 CT/CL stacks by Cardtek
- EMVCo and PCI4.x Certification
  - EMVCo Pre-certification on Level 1 CT/CL by FIME
  - PCI 4.1 Pre-certification on the K81 performed by Infogard
  - PCI 4.1 PIN Entry Device (PED) Certification by Infogard
- Kinetis® K81 Secure MCU
  - Advanced physical tamper security
  - Advanced Public-key hardware w/ support for RSA and ECC
  - XIP from external Q-SPI flash w/ decrypt on the fly
- PN5180 contactless 13.56 MHz NFC front end IC
  - Dynamic Power Control for small antennae design
  - Full compliance with all NFC and EMVCo standards
- TDA8035 contact front end IC
  - 5V, 3V, 1.8V smart card supply
  - Very low power consumption in Deep Shutdown mode

Resources

- **AN4507**: “*Using the Kinetis Security and Flash Protection Features*”

- **AN5112**: “*Using the Kinetis Flash Execute-only Access Control Feature*”

- **AN4307**: “*Using the mmCAU in Kinetis*”
  - **AN4307SW**: Example software for AN4307

- **AN4733**: “*Using the DryIce Tamper Detection Unit on Kinetis Microcontrollers*” (available under NDA only)

Summary

- In today’s connected world, security is important for protecting you and your customers.
- Firmware must be protected to maintain the security of end devices and the data they generate
- NXP’s microcontrollers contain HW features and software enablements that can be integrated to strengthen your end device

Download MCUXpresso SDK for K28F and KBOOT today to secure your firmware!

NXP UNIQUELY POSITIONED TO DELIVER SECURE SMART CONNECTED SOLUTIONS

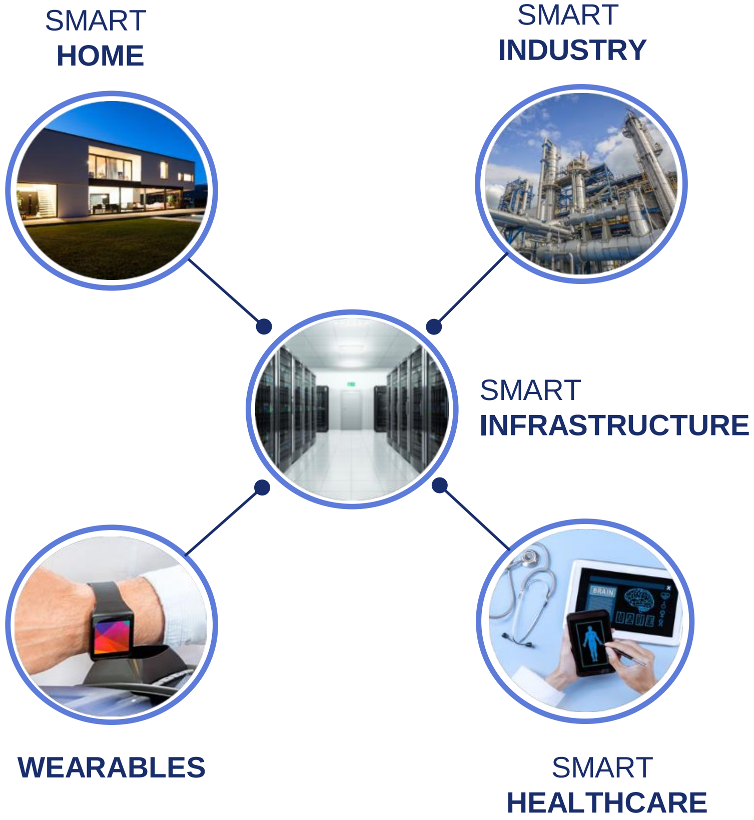
Security Technology

Application Identification	Device Identification
Certification	Compliance
Cryptography Acceleration	Network Security
NFC	RFID
Secure Boot	Secure Keys
Secure Memory	Secure Update
Trusted Execution Environments	Unique Chip Identity

Security Expertise

- E-Passport
- Mobile Transactions
- Banking

Smart Connected


SMART HOME
SMART INDUSTRY
SMART INFRASTRUCTURE
WEARABLES
SMART HEALTHCARE