

ARTIK Partner Training

ARTIK 05x Modules

Agenda

- ARTIK 05x Module Overview
- ARTIK 05x Module Security
- ARTIK 05x Module Development
- ARTIK 05x Hands-on Lab

ARTIK 05x Module Overview

Samsung ARTIK™ 053/053s, 055s Wi-Fi® edge nodes

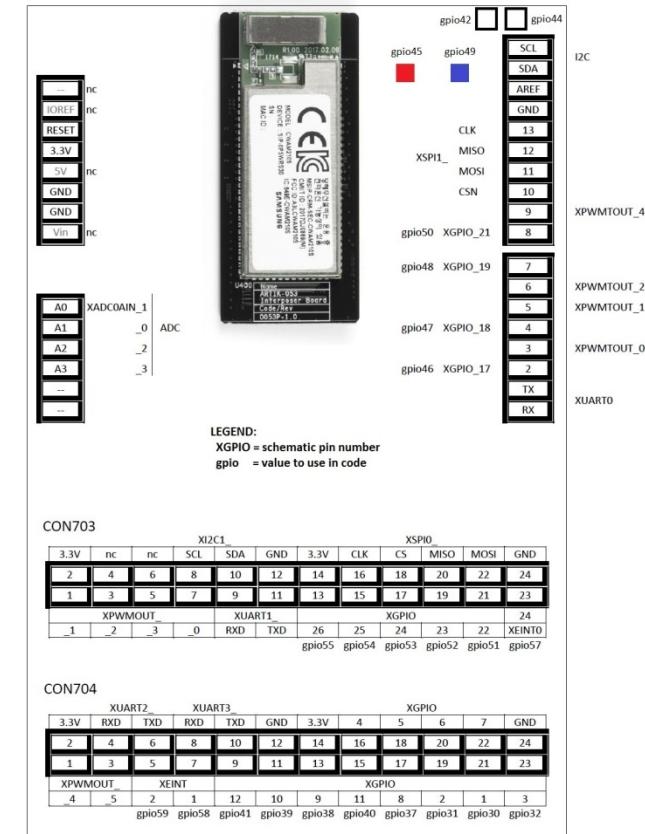
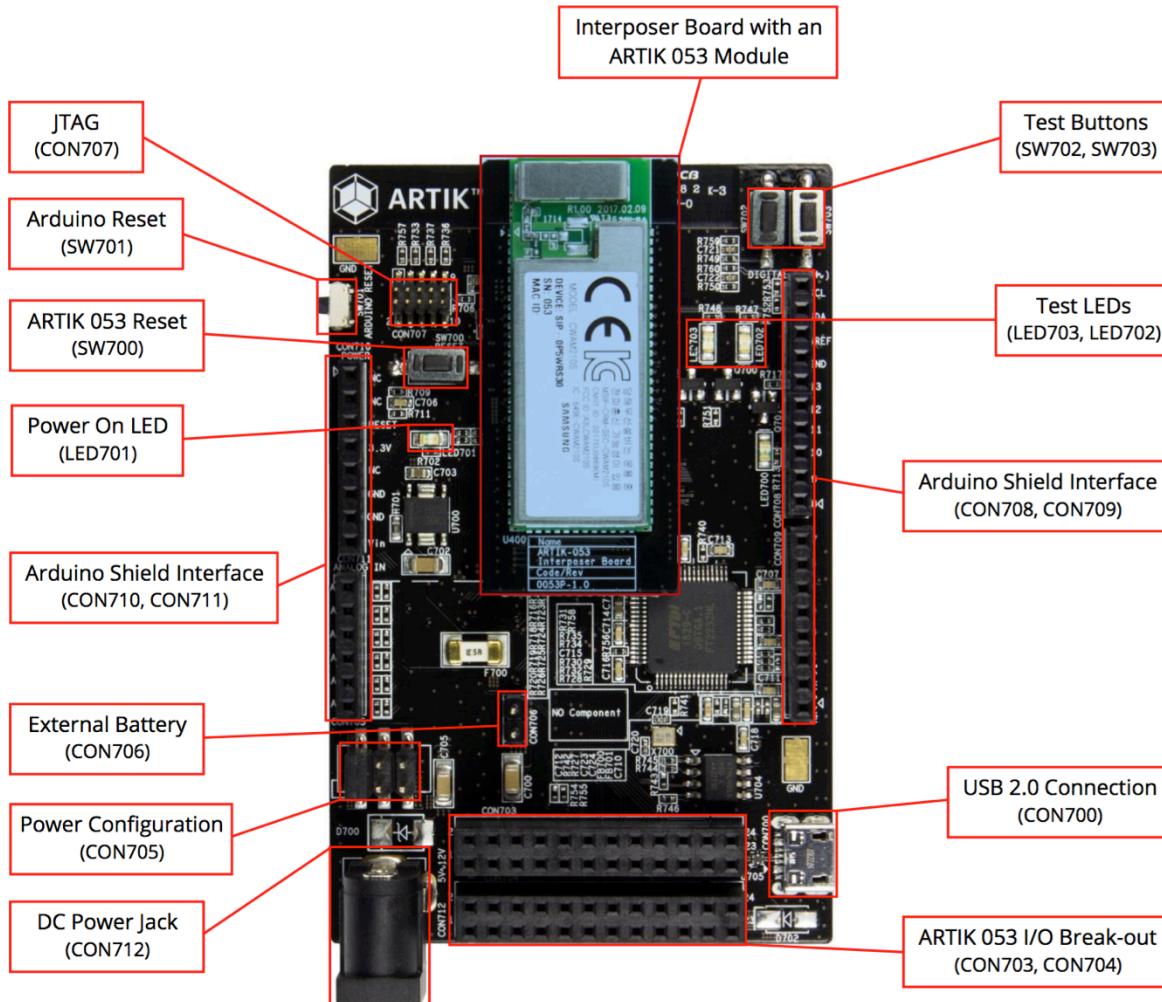
Create secure, next-gen edge products



- Home health monitors, AEDs, fitness equipment, CPAP
- Smoke detectors, thermostats, energy monitors, appliances
- Sensors, lighting controllers, motors, valves
- Access control, fire monitors, smart switches

Processor	Main: ARM Cortex® R4 @ 320 MHz WLAN: ARM Cortex® R4 @ 480 MHz Security: ARM Cortex M0
Memory	RAM: 1.4 MB Flash: 8 MB SPI Flash on module
Connectivity	WLAN (Wi-Fi): IEEE 802.11 b/g/n
Security	Secure Subsystem, Hardware-protected key storage with secure point-to-point authentication and data transfer, secure boot*, KMS* <small>*S-versions only</small>
I/O	2xSPI, 5xUART (2-pin), 4xI2C, 7xPWM, 28xGPIO, 1xJTAG, 4xADC
Operating voltage	055s: 3.3 VDC 053, 053s: 5-12 VDC
Temperature range	-20° to 85° (°C)
Size	055s: 15 mm W x 26 mm H x 3.9 mm D 053, 053s: 15 mm W x 40 mm H x 3.9 mm D

ARTIK 05x Starter Kit



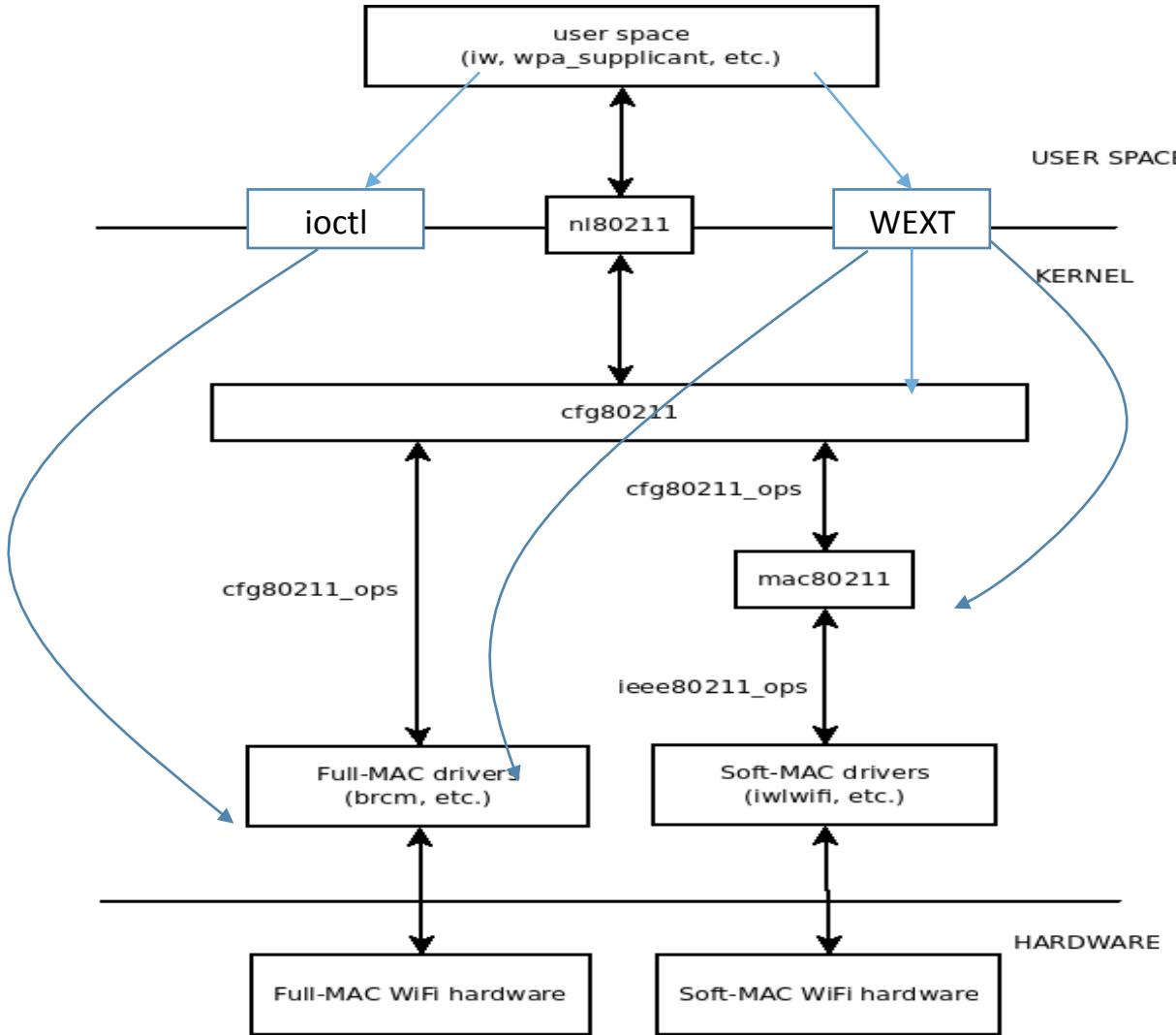
Wi-Fi Subsystem

- ARTIK05x supports 802.11b/g/n Wi-Fi at 2.4GHz
- Dedicated Wi-Fi Processor subsystem with 480MHz 32-bit ARM Cortex R4 supported by 32KB I-Cache and 16KB D-Cache
- WiFi throughput: ~25 Mbps single stream
- WPA/WPA2

Samsung ARTIK™ 05x WiFi – wpa_supplicant

- Supplicant is used in the client stations for key negotiation with a WPA Authenticator.
- wpa_supplicant is designed for Linux, BSD and Windows with support for WPA and WPA2.
- wpa_supplicant was designed to use hardware, driver and OS independent, portable C code for all WPA functionality.
- A daemon program running in the background and acting as the backend component controls the wireless connection.

WiFi Flow



- `nl80211`: Interface between user space and kernel. This works like a socket.
- `cfg80211`: configuration API for 802.11 devices.
- `mac80211`: Implements the MAC layer functions, also the `cfg80211` callbacks.
- `cfg80211_ops`: a set of operations that **Full-MAC** drivers and **mac80211** module register to **cfg80211** module
- `ieee80211_ops`: a set of operations that Soft-MAC drivers register to **mac80211** module

Samsung ARTIK™ 05x Power Management

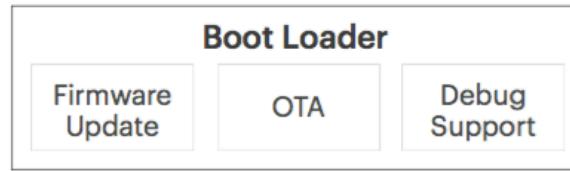
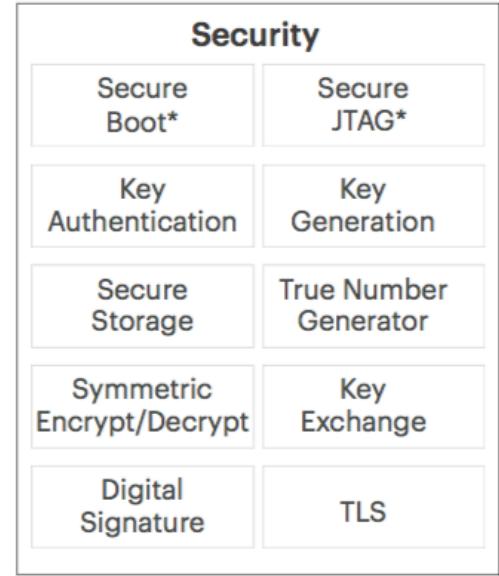
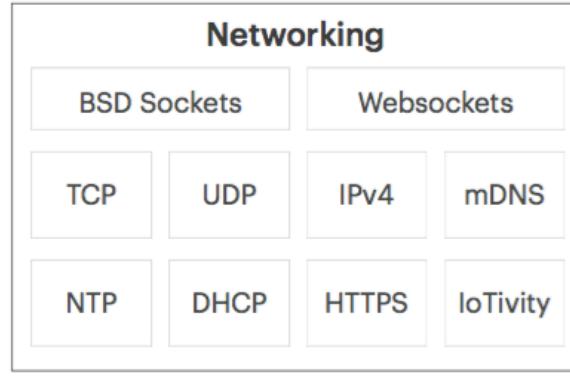
	Description	Power Consumption	Current @ 5V Typical	Current@12V Typical
Normal	Normal states. All components are running.	55mA normal cases	362mA +/- 30	151mA +/- 15
WiFi OFF	In this mode, almost all components are powered down. It consumes low power while downs clock.	Up to 0.05W	46.5mA+/-12	23.3mA+/-1.7
Deep Stop (DSTOP)	Almost all components are powered down except PMU (Power Management Unit).This state is similar to Power Off. However, DSTOP can be waken up by events such as external interrupt(Push button), UART or I2C device. And it also maintains their context before going to DSTOP.	Up to 0.05W	0.1mA	0.15mA

TizenRT OS Basics

- ARTIK 05x are powered by TizenRT
- TizenRT is a lightweight RTOS-based platform to support low-end IoT devices, based on Nuttx
- Primary governing standards are POSIX and ANSI standards
- IP Network Stack

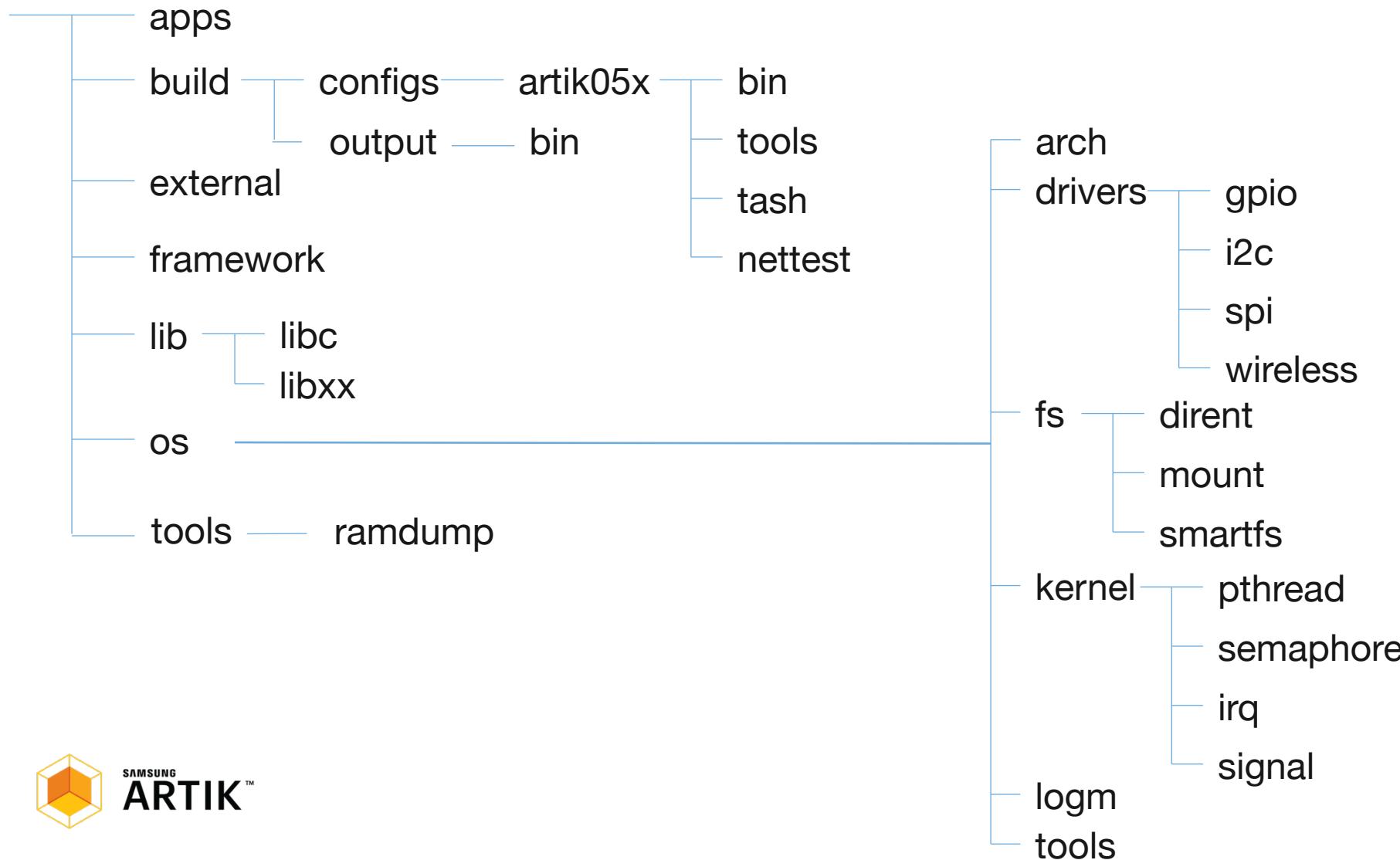
TizenRT OS Hierarchy

Kernel Services	
Realtime	Tasks, threads, queues, mutex, semaphore, signal
Time	Real-time clock, date/time, timer, sleep
Network Services	
Internet	DHCP, NTP Client, DNS Client, mDNS, BSD Sockets, Websockets
Services	Web client/server, MQTT client, IoTivity, cJSON
libc Services	
Libc Compatibility	Flash based Stdio, Stdlib, String, Unistd, Time
Security Services	
Encryption	AES 128/256, RSA 1024/2048, ECC BP/NIST 192/224/256/384/512
Authentication	HMAC 128/256, certificate
Certificate Storage	Secure Flash storage
Firmware Integrity	Secure boot and JTAG protection



*(s version)

Directory Structure



TizenRT Build Types

- In TizenRT, only Flat Build mode is supported.
- Flat Build:
 - The code is built as a blob that runs in a flat address space out of on-chip FLASH(or SRAM) memory.
 - Designed for resource-limited MCUs that have no MMU
 - Build mode is enabled by configuring CONFIG_BUILD_FLAT = y

Tasks/Threads

- TizenRT provides two different interfaces:
 - from sched.h
 - `int task_create(const char *name, int priority, int stack_size, main_t entry, char *const argv[])`
 - from pthread.h
 - `int pthread_create(pthread_t *thread, const pthread_attr_t *attr, pthread_startroutine_t startroutine, pthread_addr_t arg)`
- `task_create()` is the same as calling `fork()` and `execv()` in Linux environment. It creates a real task (or process).
- `pthread_create()` has same interface as `pthread` in Linux. It creates a thread, not an independent task

Semaphore, Mutex, Message Queues

- TizenRT supports the POSIX-like interfaces for Semaphore: `sem_init()`, `sem_wait()`, `sem_post()`...
- For mutex support, developers need to use pthread interfaces: `pthread_mutex_init()`, `pthread_mutex_lock()`...
- TizenRT supports POSIX-like named message queues for inter-task communication:

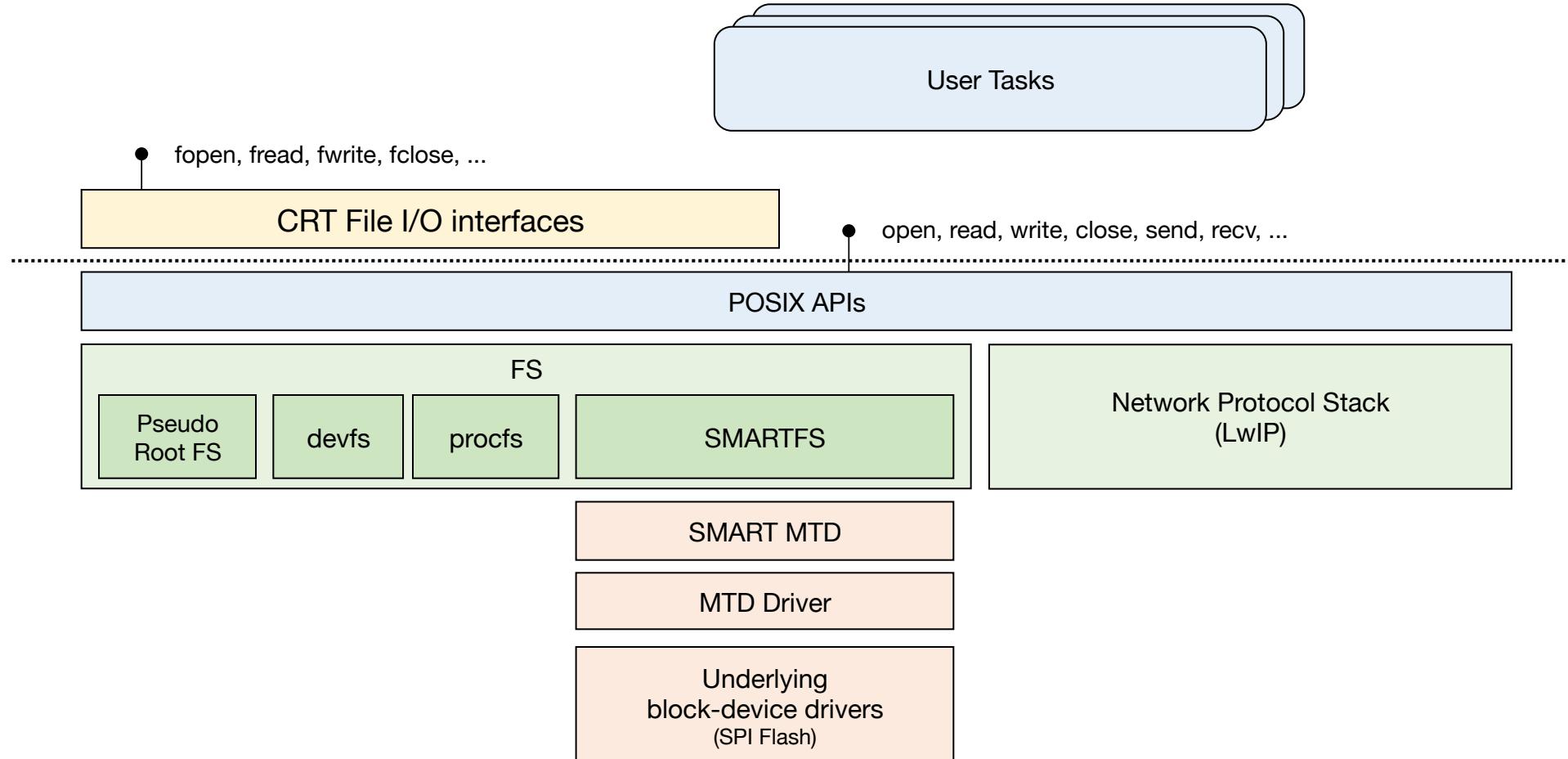
```
mqd_t mq_open(const char *mq_name, int oflags, ...)  
int mq_close(mqd_t mqdes)  
int mq_send(mqd_t mqdes, const char *msg, size_t msglen, int prio)  
int mq_receive(mqd_t mqdes, char *msg, size_t msglen, int *prio)
```

Software Timer

TizenRT supports POSIX-like per-thread timer interfaces:

```
int timer_create(clockid_t clockid, struct sigevent *evp, timer_t *timerid)  
  
int timer_delete(timer_t timerid)  
  
int timer_settime(timer_t timerid, int flags, const struct itimerspec *value, struct itimerspec *ovalue)  
  
int timer_gettime(timer_t timerid, struct itimerspec *value)  
  
int timer_getoverrun(timer_t timerid)
```

File System Hierarchy



File System

- Pseudo is an in-memory file system which does not require any storage medium or block driver support.
- File system contents in this pseudo file system can be accessible via file system APIs (open, close, read, write, etc.)
- Devfs, procfs or true block-based file systems can be mounted at any empty directory node under the root file system.
- As in Linux, name resources such like named semaphore, message queue, and shared memory special files can be created under the root pseudo-file system.

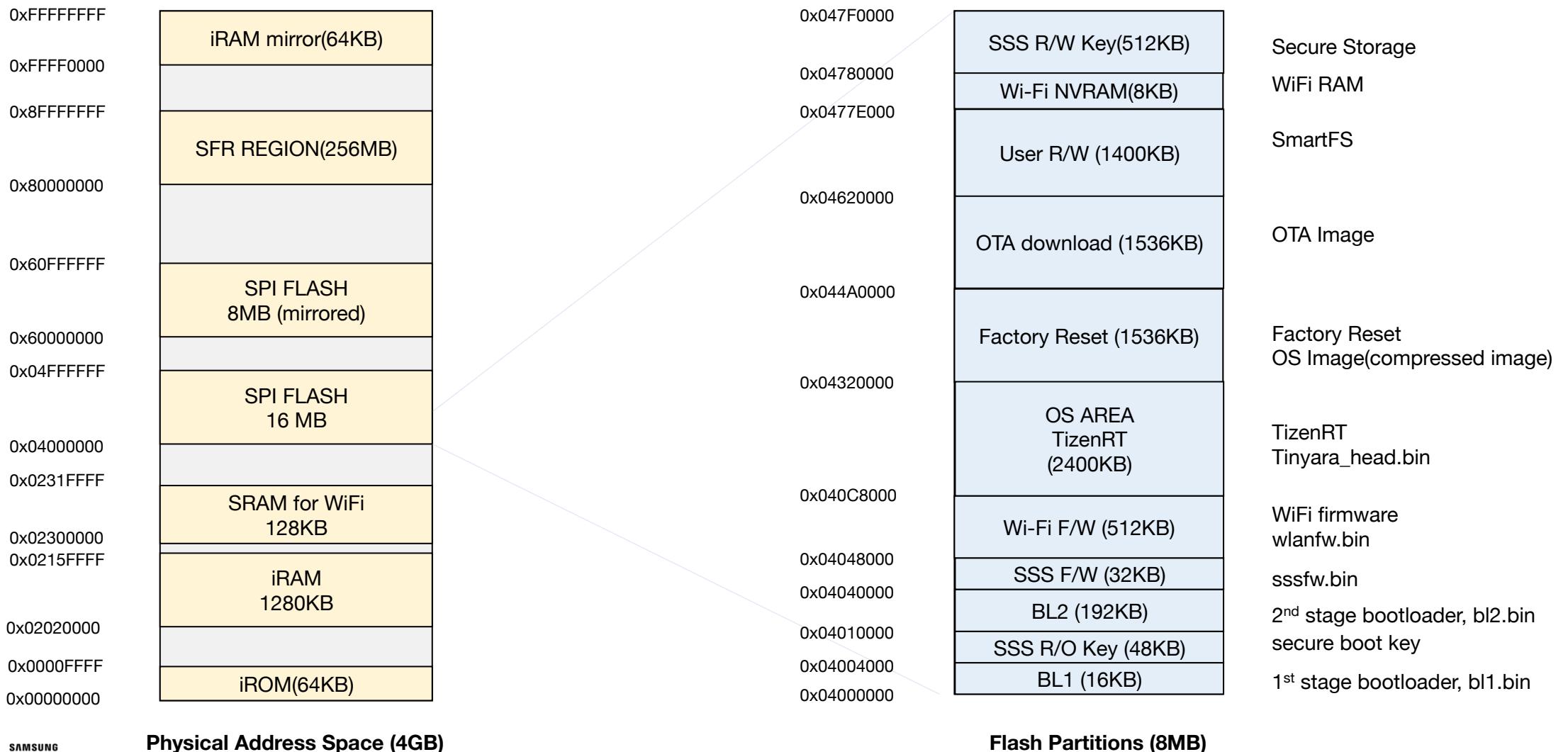
SmartFS

- TizenRT uses SmartFS as a main file system to manage its files. After boot, a SmartFS volume is mounted at /mnt.
- SmartFS consists of two layers
 - The MTD block layer - Smart MTD handles low-level media operations such as logical sector allocation, freeing and management, erase block management etc.
 - SmartFS layer - SmartFS implements the abstraction of file and directory, such as creating new files, chaining logical sectors together to create files, creating directories
- Filesystem interfaces: fcntl.h, unistd.h, stdio.h, aio.h

Database (AraStorage)

- A lightweight database named AraStorage is implemented with SQL-compatible interfaces.
- Advanced features, such as:
 - b+ tree-based indexing algorithm
 - Cursor structure to improve usability for the application layer

Memory map



IoTBus Framework

- GPIO (General Purpose Input/Output)
- I2C (Inter Integrated Circuit)
- SPI (Serial Peripheral Interface)
- PWM (Pulse Width Modulation)
- UART (Universal Asynchronous Receiver/Transmitter)

Device Management and LWM2M stack

- Lightweight M2M (LWM2M) is a device lifecycle management specification
- Provides a specification for functions like: firmware upgrade, provisioning of certificates, access control policies, connectivity monitoring etc.
- Based on CoAP protocol
- LWM2M allows the use of UDP for communication between client and server
- DTLS security to ensure authentication, data confidentiality and integrity between an LWM2M client and ARTIK Cloud server(an LWM2M server).

ARTIK 05x earns OCF 1.3 Certification

- The Open Connectivity Foundation is dedicated to ensuring secure interoperability
- ARTIK 05x family of systems-on-module for IoT became the first product family that is certified on the latest OCF standards for trust and connectivity for IoT.

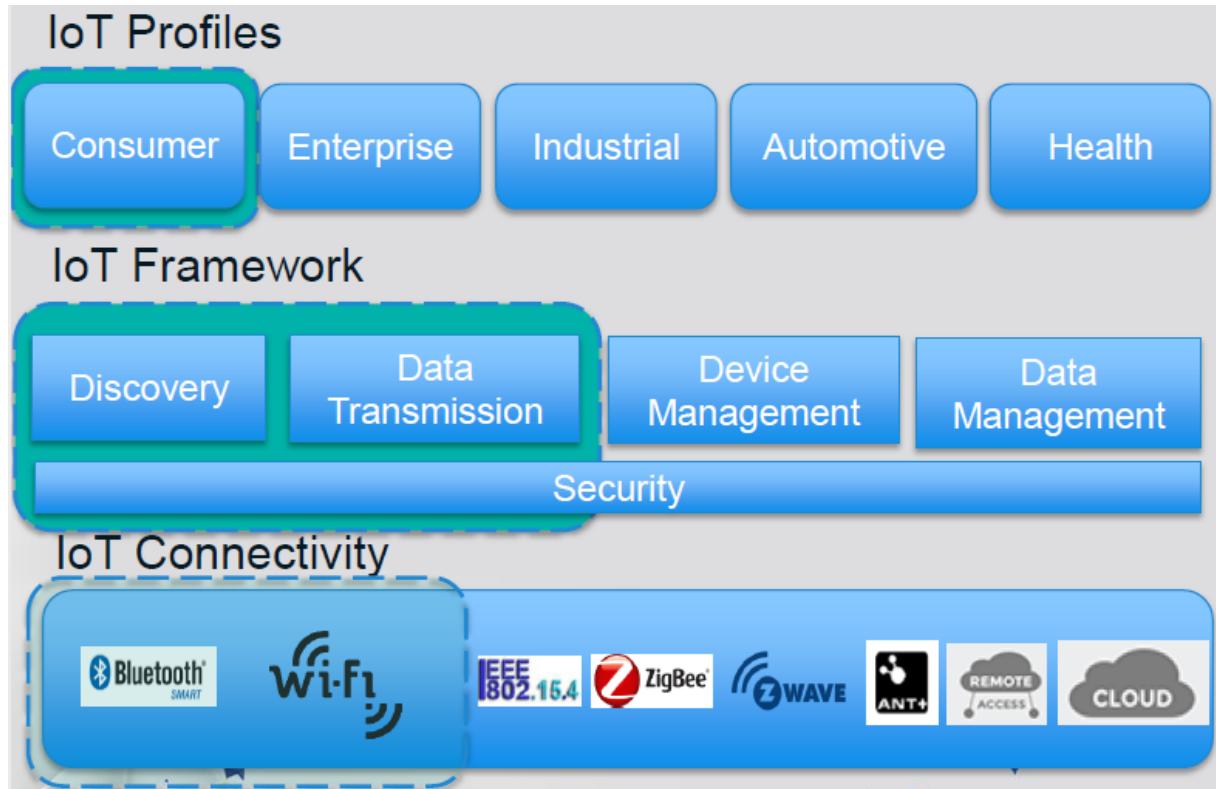
- It ensures products created by using ARTIK 05x can work seamlessly with other OCF certified IoT devices regardless of their form factor, OS or service providers etc.

Product Name	Certification Type	Company Name	Device Type(s)	Date Certified
Haier Washer	OCF	Haier Group	Washer (Laundry)	01/24/2018
InstaView ThinQ	OCF	LG Electronics, Inc.	Air Purifier, Fan, Humidifier, Light, Smart Plug, Switch	12/22/2017
Lynx MiND	OCF	Lynx Technology	Door, Fan, Freezer, Generic Sensor, Light, Oven, Refrigerator, Smart Plug, Switch, Thermostat	12/19/2017
SURE Universal Set Top Box	OCF	SURE Universal	Set Top Box	12/19/2017
SURE Universal Remote	OCF	SURE Universal	Air Quality Monitor, Camera, Cooktop, Dehumidifier, Door, Garage Door, Generic Sensor, Oven, Printer Multi-Function, Receiver, Smart Lock	12/19/2017
Sure Universal Remote	OCF	SURE Universal	Air Quality Monitor, Camera, Cooktop, Dehumidifier, Door, Garage Door, Generic Sensor, Oven, Printer Multi-Function, Receiver, Smart Lock	12/19/2017
ARTIKTM 053		Samsung		
Smart IoT Module	OCF	Electronics Co., Ltd.	Light	12/12/2017
Alegro 100	OCF	VIA Technologies	Smart Plug	02/20/2017
SURE Universal Remote	OCF	SURE Universal	Switch	12/23/2016
Family Hub 1.0	OCF	Samsung Electronics Co., Ltd.	Refrigerator	09/22/2016

IoTivity



IoTivity is an open source software framework enabling seamless device-to-device connectivity to address the emerging needs of the Internet of Things



ARTIK 05x Module Security

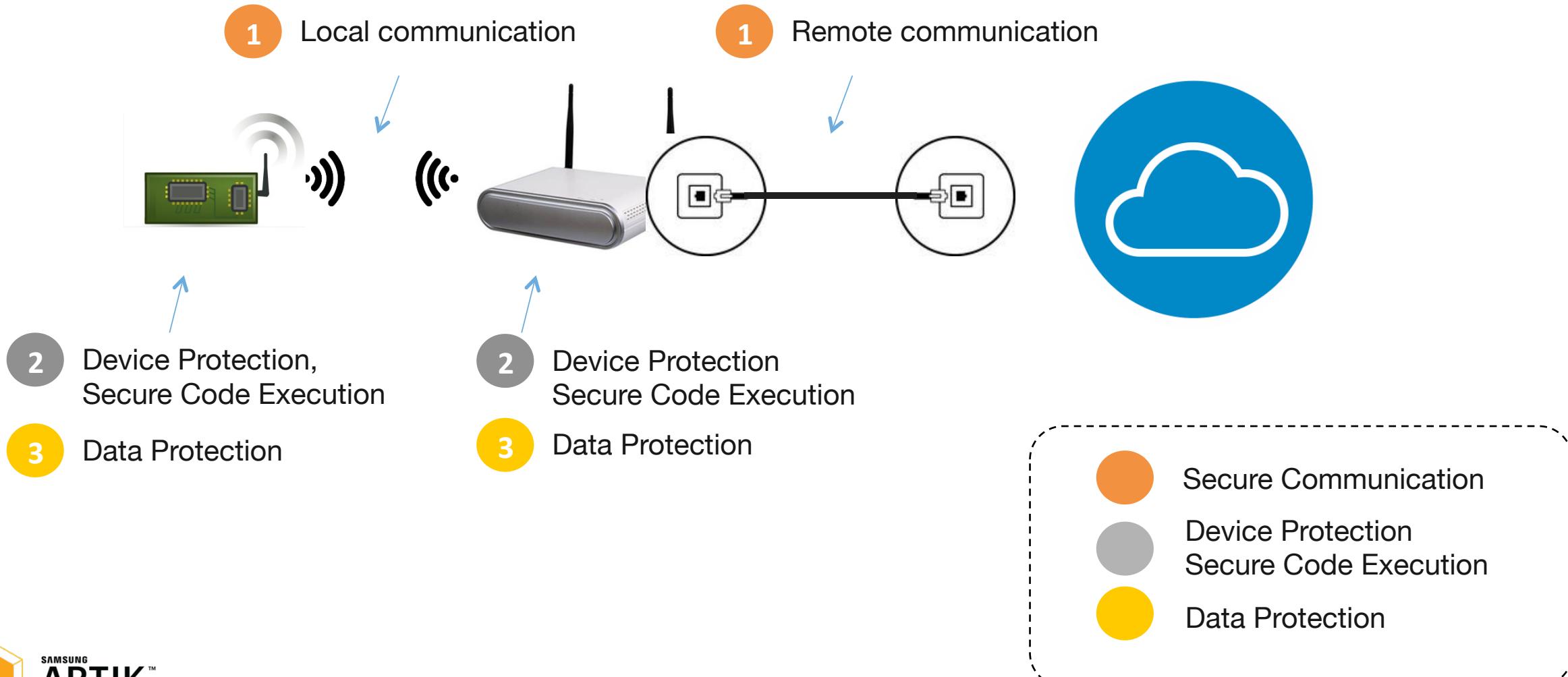
Module Security

Non-S vs. S Modules

- Same HW specifications other than security features
- "s" type modules can be identified by **blue** labeling



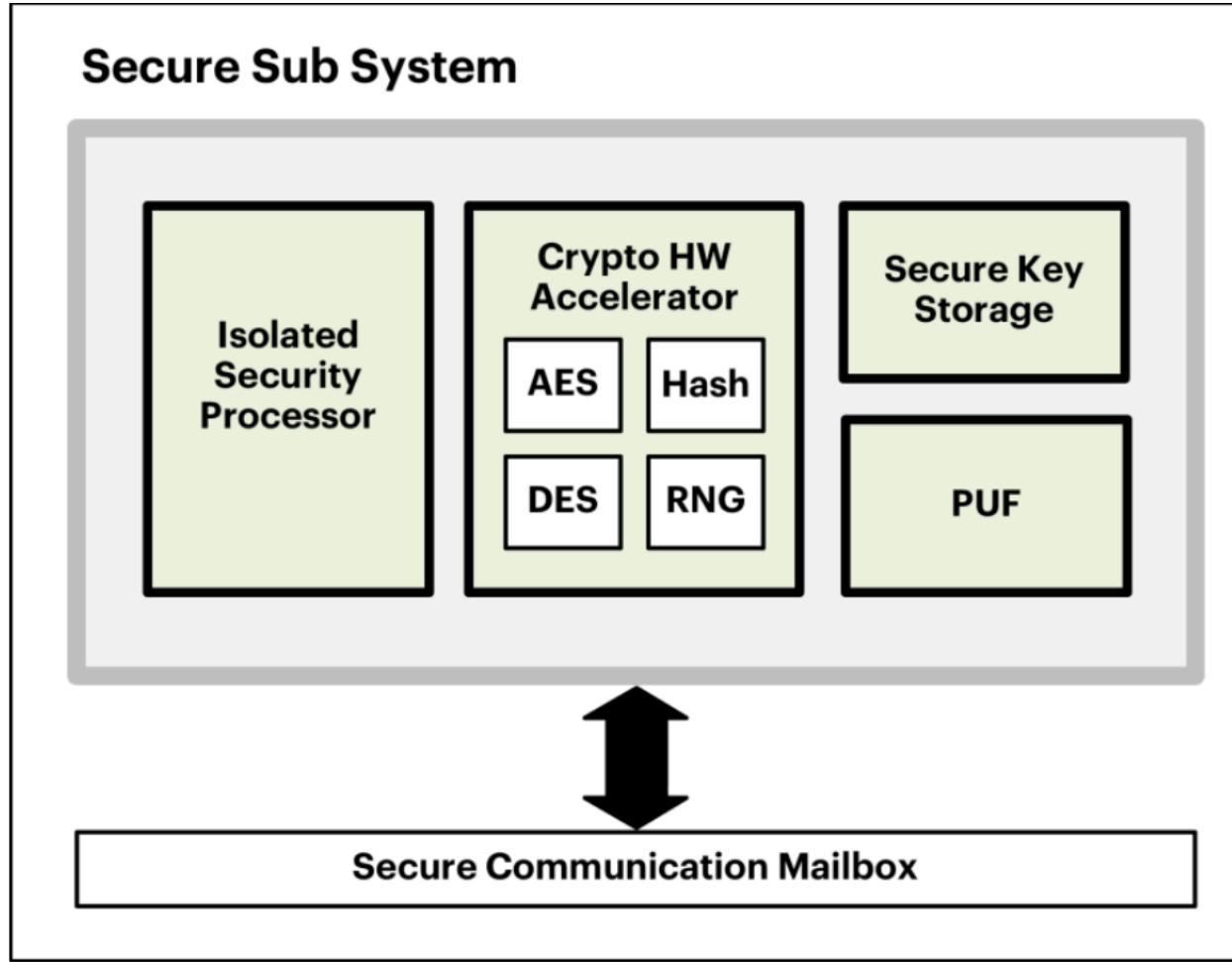
Why S-Module?



Samsung ARTIK™ S-Module Features

		ARTIK module (05x, 5, 7)	ARTIK S-module	Comments
Secure communication	Per device unique key & certificate	✓	✓	Uniquely identifies device
	Key stored in HW secure element	✓	✓	Secure key storage
	PKI infrastructure: Mutual authentication of device and cloud	✓	✓	Device talks to authorized cloud and vice versa
	Post Provisioning		✓	Provision with your own keys and certificates
Device protection/ secure code execution	KMS infrastructure for code signing		✓	Key Management Service
	Code verification key in HW		✓	Secure key storage
	Secure boot (check for authorized code)		✓	Boot image verification
	JTAG access locked		✓	Lock out debug access
Data protection/ Secure storage	Secure OS (separate normal & secure operations)		✓	Hardware enforced secure applications via TEE
	Security Lib API (27 API calls)	Limited(random number generator, get cert and signature)	✓	Key Manager, Authentication, Secure Storage, Post Provisioning, Encrypt/Decrypt
	Secure storage		✓	Encrypt data stored on Flash

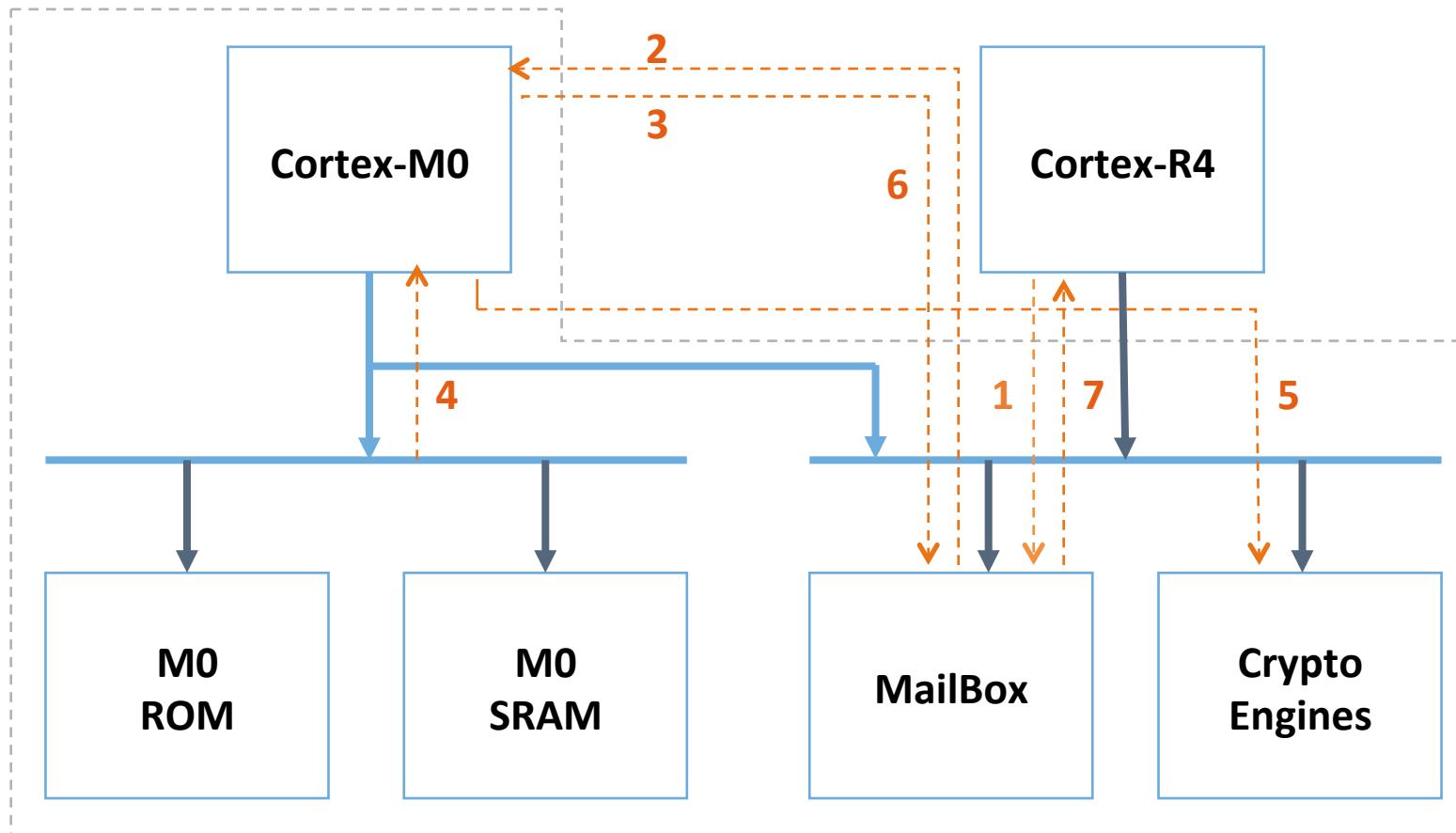
ARTIK 05x Security Subsystem



- Isolated Security Processor
- Cryptographic Hardware Acceleration
- A Physical Uncloneable Function(PUF)
- Secure Key Storage

Isolated Security Processor

Security Subsystem



Cryptographic Hardware Acceleration

Support for high performance cryptographic acceleration

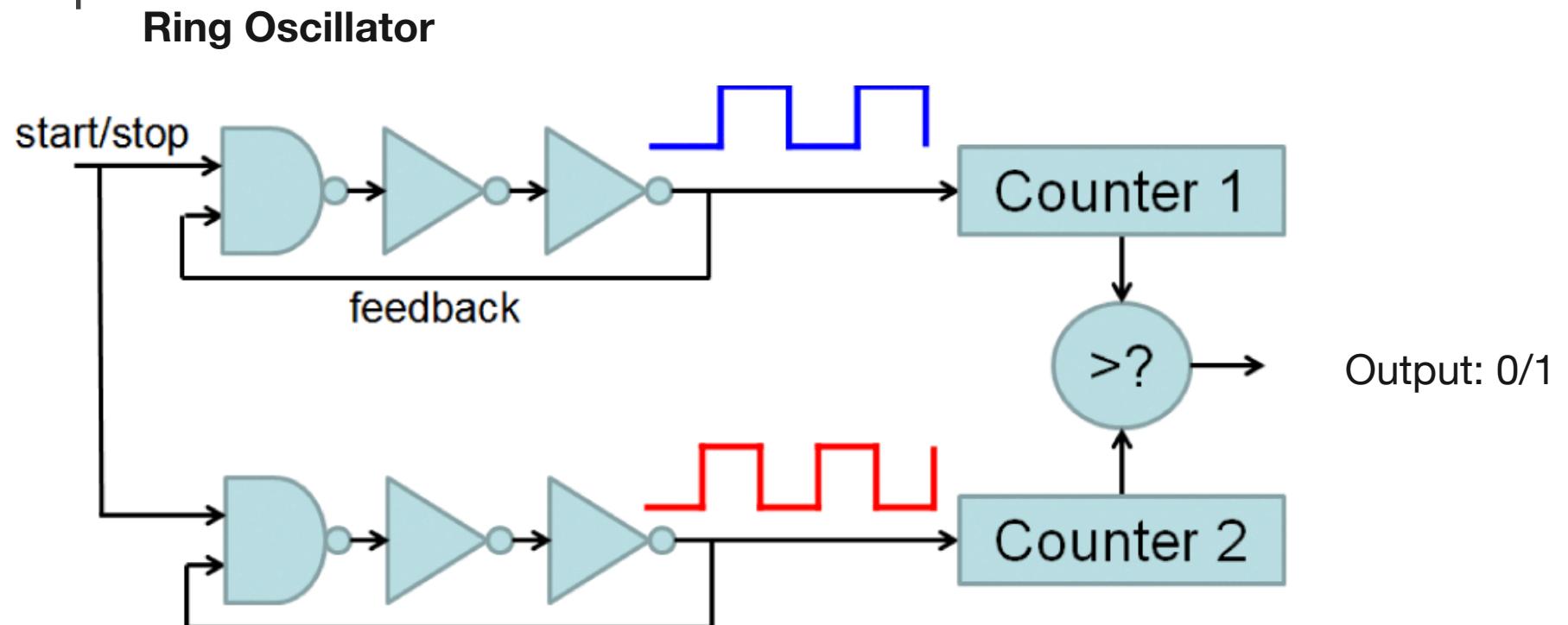
- Random Number Generation: DTRNG, PRNG
- Block Cipher: Secure AES, DES
- Hash Function: SHA1/SHA2/SHA3 with HMAC
- Public Key Cryptosystem: RSA, ECDSA, DH, ECDH
- FIPS Compliant: CAVP, CMVP, MDFPP

PUF (Physically Unclonable Function)

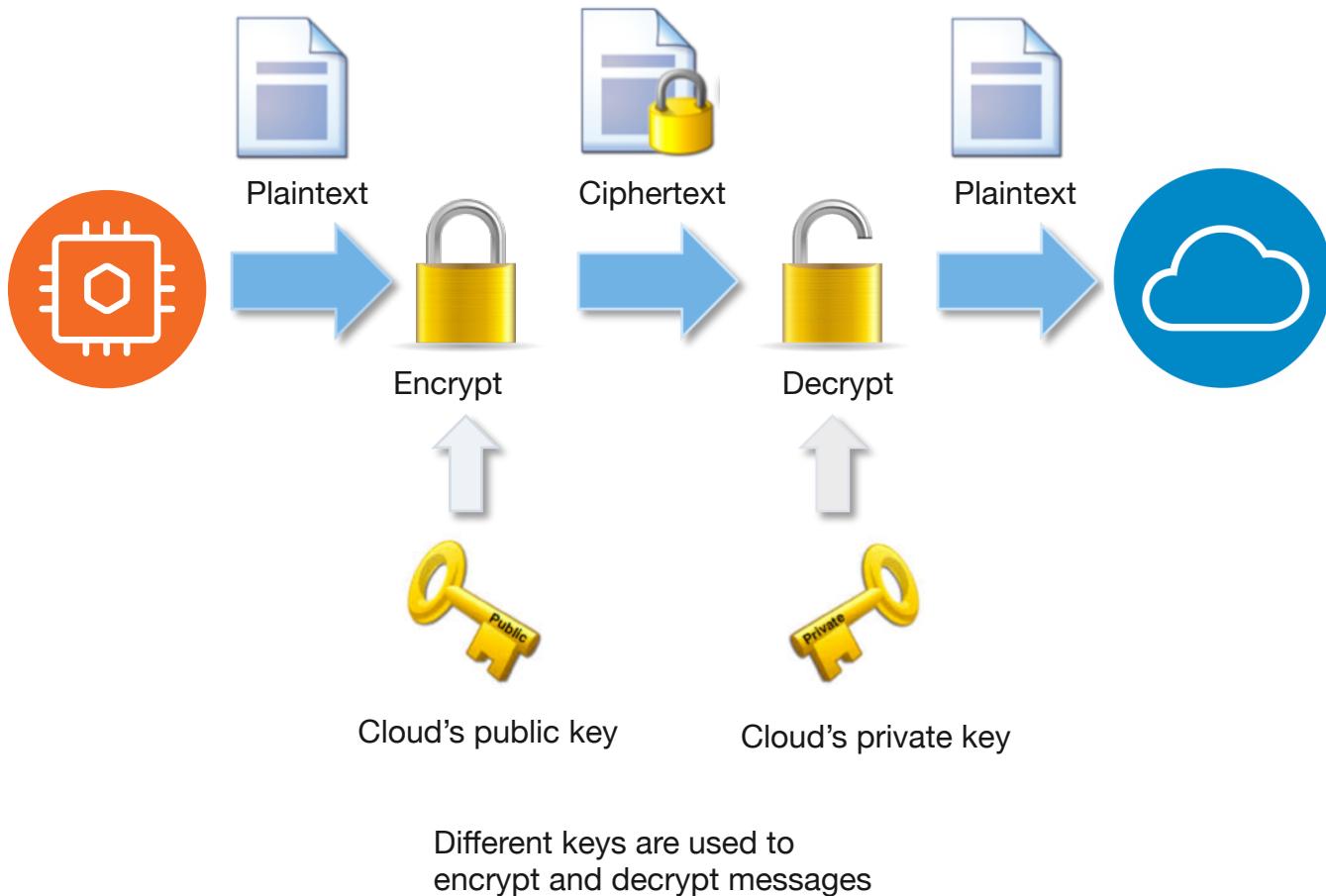
- Create a cryptographic key(PUF KEY) that can not be cloned by anybody else
 - PUF Key is auto generated using process variation during Manufacturing
 - Unchanging value over product lifetime
 - Unclonable
- Applications of PUF:
 - Device identification
 - Key generation and storage (seed that never generates the same key)
 - IP Protection
 - Protocols with challenge-response pairs

RO Frequency PUF

- RO (Ring-Oscillator) frequency is used as the PUF input to generate a unique key for each chip

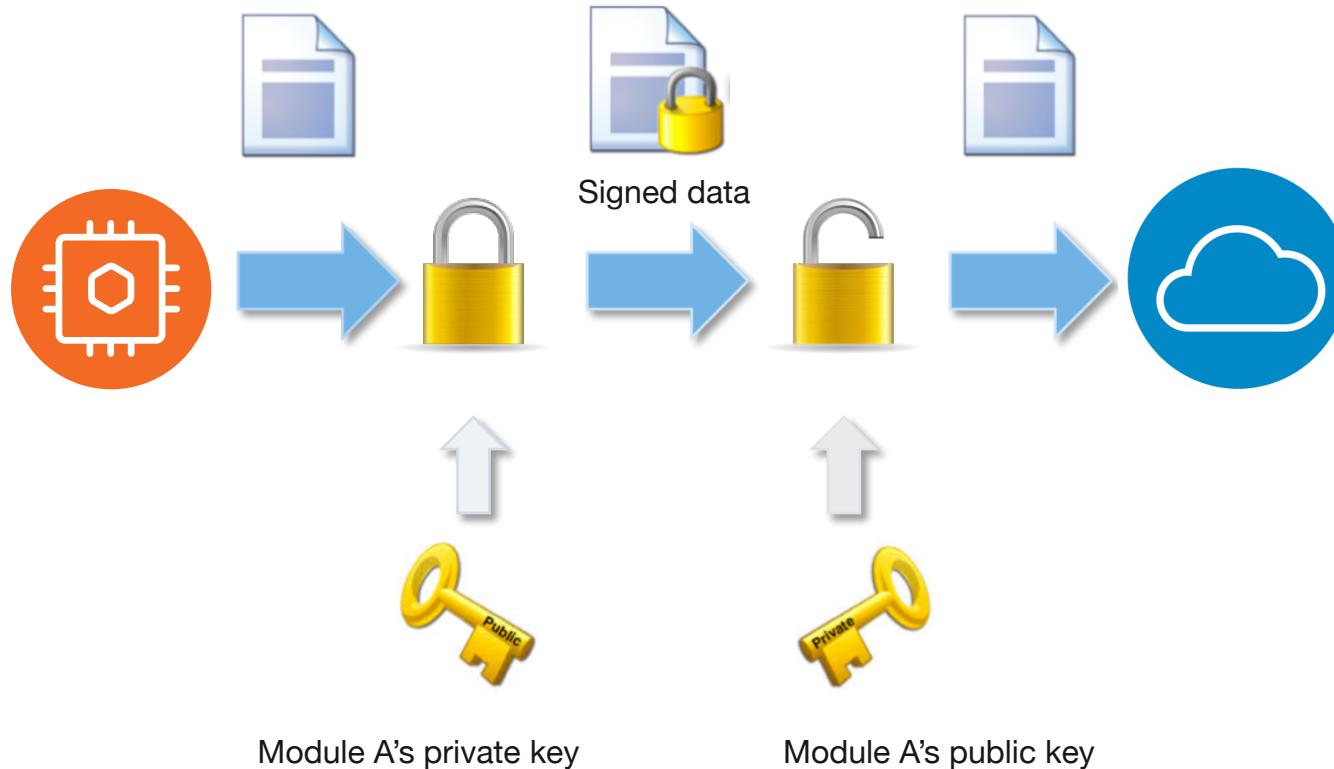


Public Key Infrastructure (PKI)



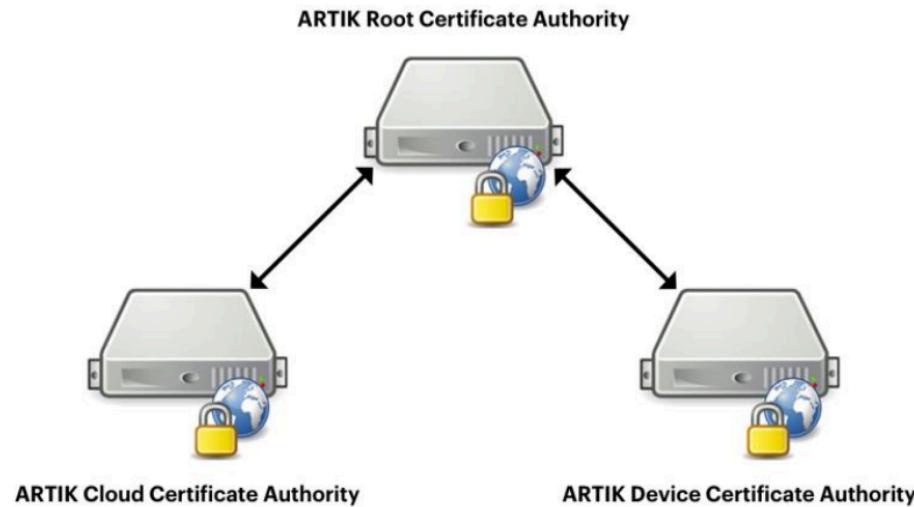
- A Public Key Infrastructure (PKI) supports the distribution and identification of public encryption keys, enabling users to securely exchange data over networks
- ARTIK provides PKI, which is used to generate and apply unique certificates and key pairs to each ARTIK Module during manufacturing.

Signature



Digital Certificate

- PKI's core concept is Digital Certificate
- Issued by a **Certificate Authority**, e.g, GlobalSign, Symantec
- ARTIK Root CA



X.509

- A Digital Certificate contains a public key and an identity (a hostname, or an organization, or an individual)
- X.509 is a standard that defines the format of public key certificates. X.509 certificates are used in many Internet protocols, including TLS/SSL, which is the basis for HTTPS

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

01:00:17:03:07:00:00:00:04

Signature Algorithm: ecdsa-with-SHA256

Issuer: C=KR, O=Samsung Semiconductor ARTIK, OU=ARTIK High Security Device CA, CN=ARTIK High Security Device CA

Validity

Not Before: Mar 7 02:27:05 2017 GMT

Not After : Mar 7 02:27:05 2028 GMT

Subject: C=KR, O=Samsung Semiconductor ARTIK, OU=ARTIK High Security Device, CN=SIP-OP5WRS30 (01001703-0700-0000-041e-0e363c7eb564)

Subject Public Key Info:

Public Key Algorithm: id-ecPublicKey

Public-Key: (256 bit)

pub:

04:75:a5:0e:65:b8:31:40:66:e6:20:63:88:7c:dc:
78:d7:17:23:67:0e:79:4d:de:61:65:93:b0:50:a1:
19:1a:ce:1c:22:d3:ae:11:24:80:ee:96:d5:14:0f:
e0:bc:bc:a7:fa:8f:50:8e:35:2f:bc:db:ed:4b:1c:
fd:35:71:88:7e

ASN1 OID: prime256v1

NIST CURVE: P-256

X509v3 extensions:

X509v3 Key Usage: critical
Digital Signature, Non Repudiation

X509v3 Extended Key Usage:

TLS Web Client Authentication, TLS Web Server Authentication

Signature Algorithm: ecdsa-with-SHA256

30:45:02:21:00:ba:87:ec:ce:7e:83:d1:ec:6b:6b:54:
92:6f:f7:4a:d4:6d:19:4a:5d:e0:df:3d:0e:73:ef:63:
16:02:20:60:ee:16:f9:e5:e0:24:61:04:d6:25:09:5d:c7:87:
68:06:7c:e5:b3:ef:3e:4b:06:d1:5d:90:58:c0:b0:5f:ed

Issuer

Subject Information

Issuer Policies

Issuer Signature

Mutual Authentication

- Each ARTIK module is provisioned with:
 - An unique private key
 - Its associated certificate containing a public version of the key.
 - An ARTIK Root CA certificate
- ARTIK Cloud's server certificate is also rooted to the ARTIK Root CA certificate
- At connect time, server and client exchange certificates for mutual authentication

Post Provisioning

- If you want to connect your ARTIK Module to a 3rd party Cloud service or implement a link between two ARTIK modules, you need to generate your own certificate/key-pair
- We can use Post Provisioning APIs to post provision customer credentials(key, certificate) to Secure Element

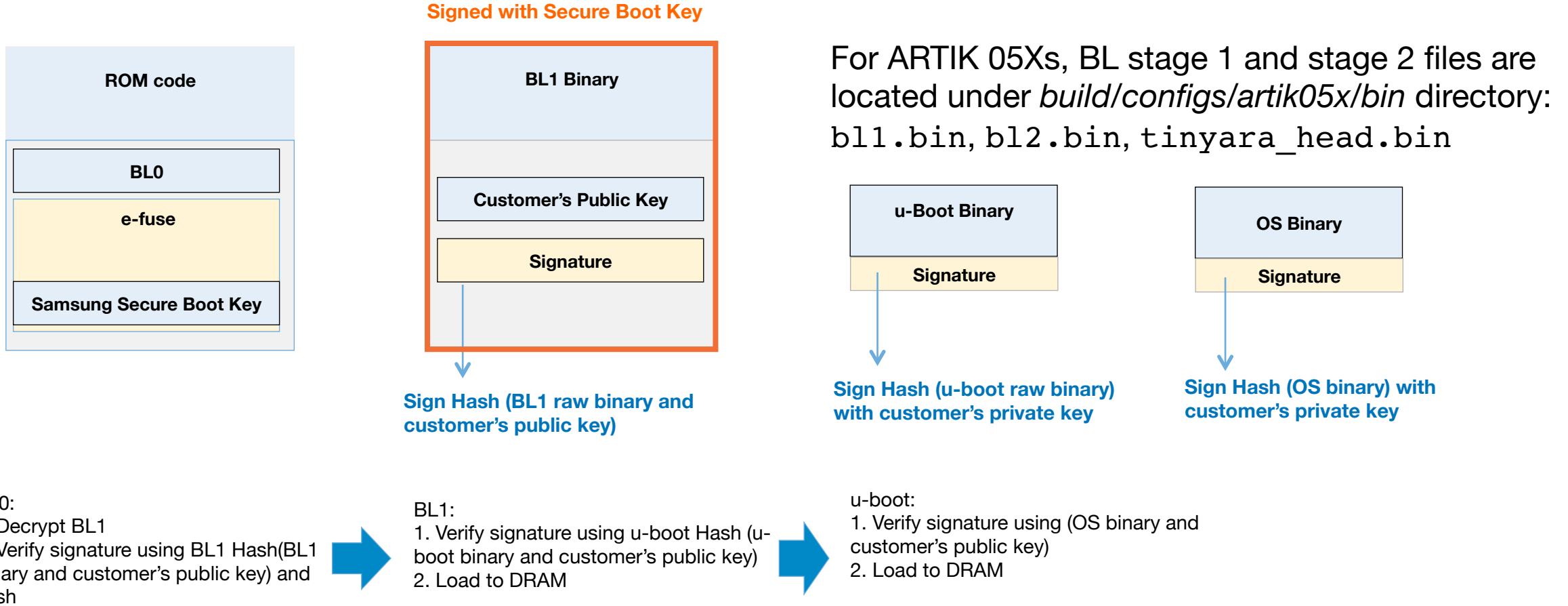
Secure Communication

		ARTIK module (05x, 5, 7)	ARTIK S-module (053s, 055s, 530s, 710s)	Comments
Secure communication	Per device unique key & certificate	✓	✓	Uniquely identifies device
	Key stored in HW secure element	✓	✓	Secure key storage
	PKI infrastructure: Mutual authentication of device and cloud	✓	✓	Device talks to authorized cloud and vice versa
	Post Provisioning		✓	Provision with your own keys and certificates
Device protection/ secure code execution	KMS infrastructure for code signing		✓	Key Management Service
	Code verification key in HW		✓	Secure key storage
	Secure boot (check for authorized code)		✓	Boot image verification
	JTAG access locked		✓	Lock out debug access
Data protection/ Secure storage	Secure OS (separate normal & secure operations)		✓	Hardware enforced secure applications via TEE
	Security Lib API (27 API calls)	Limited(random number generator, get cert and signature)	✓	Key Manager, Authentication, Secure Storage, Post Provisioning, Encrypt/Decrypt
	Secure storage		✓	Encrypt data stored on Flash

Secure Boot

- Secure Boot adds cryptographic checks to each stage of the boot process.
- The first element in the boot process authenticates the second, the second verifies the third.
- Authentication is based on digital signature verification.
- **Chain of Trust:** Every component can be authenticated before being executed.

Secure Boot for ARTIK 05x S-Module



Code Signer (Development Stage)

```
Invoking: ARTIK GCC Create Head Bin
C:/ARTIK/SDK/A055s/v1.7.1/common/tools/s5jchksum.py      "tinyara.bin"
"tinyara_head.bin"

Finished building: tinyara_head.bin

Invoking: ARTIK GCC Create Head Sign
C:/ARTIK/SDK/A055s/v1.7.1/common/codesigner/artik05x_AppCodesigner
C:/ARTIK/SDK/A055s/v1.7.1/common/codesigner/rsa_private.key
"tinyara_head.bin"

. Seeding the random number generator...
. Reading private key from
'C:/ARTIK/SDK/A055s/v1.7.1/common/codesigner/rsa_private.key'
. Generating the RSA/SHA-256 signature
. Done (created "tinyara_head.bin-signed")

+ Press Enter to exit this program.

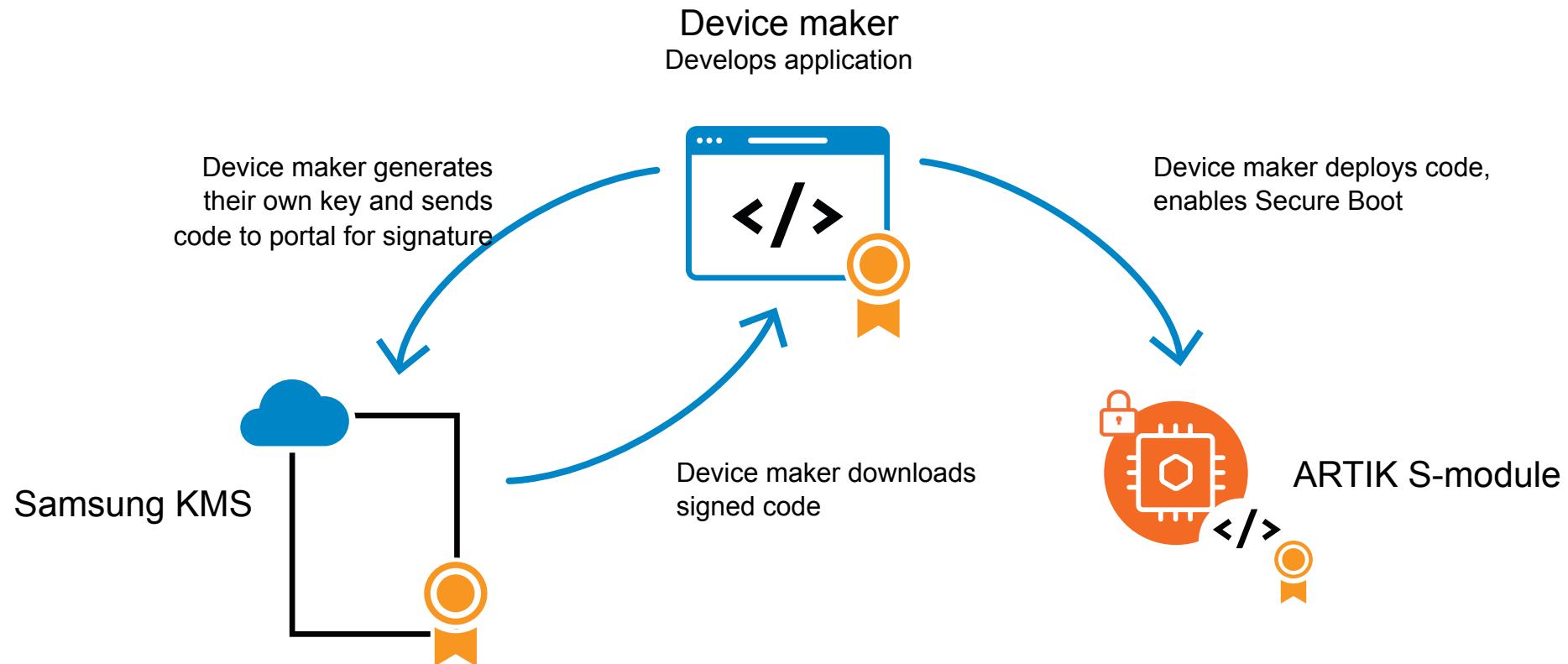
Finished building: tinyara_head.bin-signed
```



NEW

Samsung ARTIK™ Key Management System(KMS)

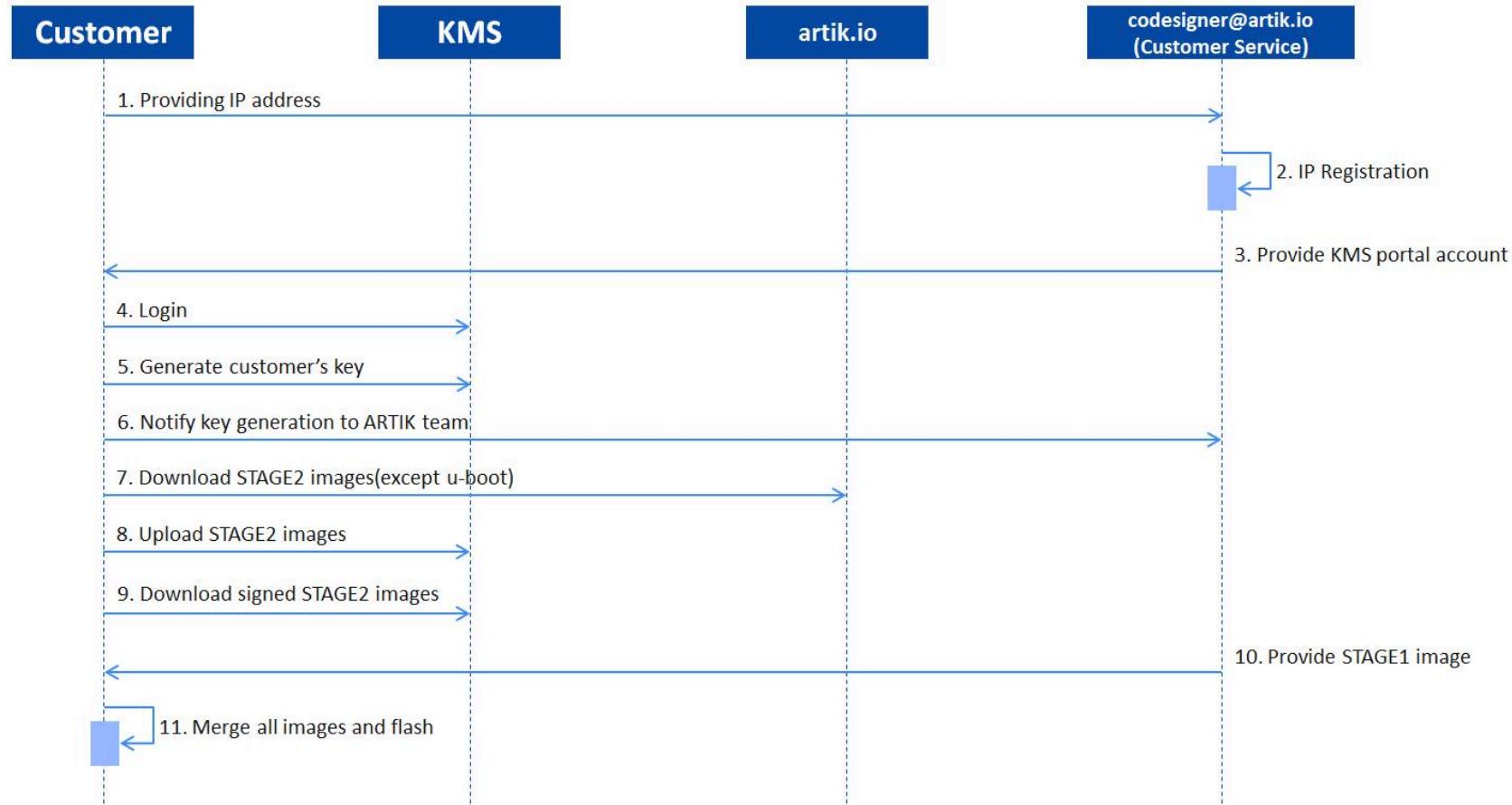
Code signing portal manages key signing



Key Management System

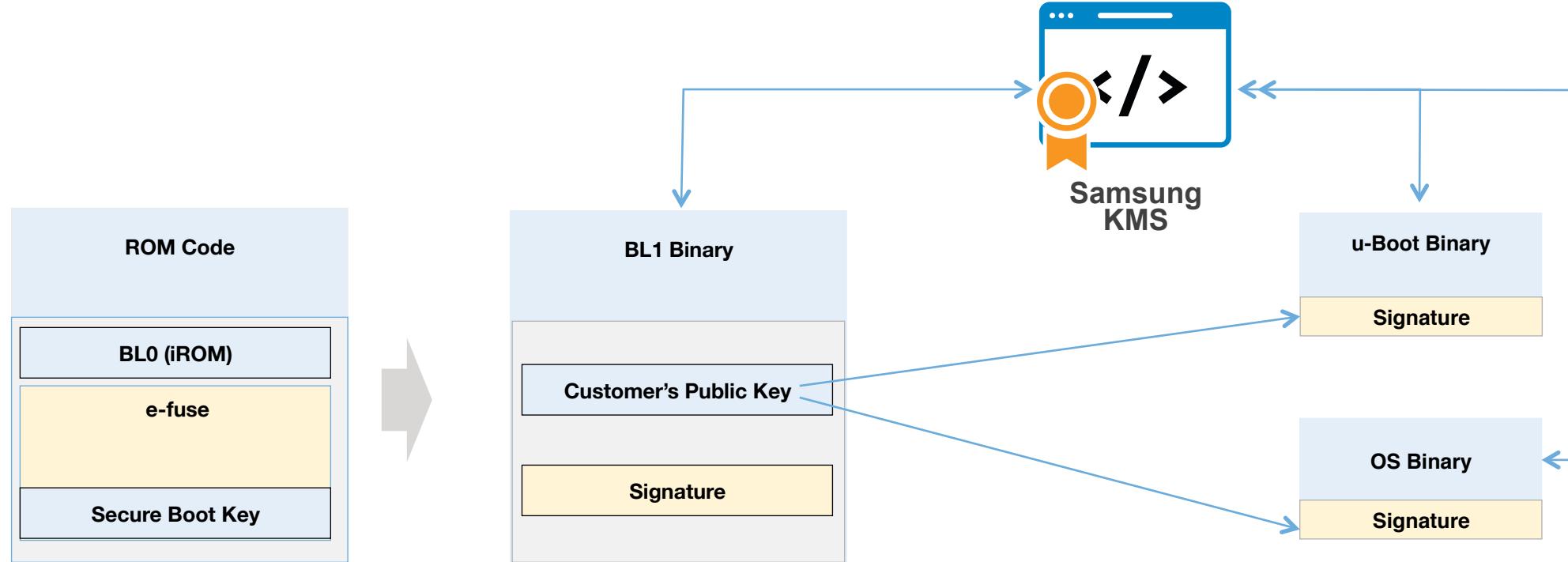
- Images are signed through a highly secure cryptography standard (SHA-256 w/ RSA2048 encryption)
- Signing keys are stored and operated within FIPS 140-2 certified Hardware security modules (HSM)
- Strict access control policies
- Only accessible through whitelisted IPs

Key Management System(Production Stage)



- Key Management System(KMS) is used for the production signing process
- Accessed through a web portal from only whitelisted IP only
- Customer generates signing keys, share the public key with Samsung. Public key gets added to the first bootloader by Samsung signing process. Customer uses private key to sign the second bootloader and OS images.

Key Management System Role



- BL1 image is provided by chip vendor

KMS Key Management

Create a new key

Model: *

New Key Name: * ARTIK_520s
ARTIK_530s_530s-1G
ARTIK_710s
ARTIK_053s_055s

Soft Card Password: *

Description:
0 / 255 characters written

CREATE **CANCEL**

Create a new key

Model: * ARTIK_053s_055s

New Key Name: * 055s-key01

Soft Card Password: *

Description: Key for Partner Workshop
24 / 255 characters written

CREATE **CANCEL**

Key Management

Success! Key "055s-key01" was created successfully.

	Model	Key Name	Public Key	Creation Time	Description
<input type="checkbox"/>	ARTIK_520s	artikaura01-5...	artikaura01-520test.spk	2017/07/24 14:04:42	
<input type="checkbox"/>	ARTIK_710s	artikaura01-7...	artikaura01-710test.spk	2017/07/24 14:09:05	
<input type="checkbox"/>	ARTIK_053...	artikaura01-0...	artikaura01-055s-key01.spk	2018/02/02 09:26:15	Key for P...

Stage 1:

- Key is immediately available on KMS portal
- Send ARTIK team the resulting public key.
- ARTIK team signs the bootloader stage 1 (BL1) image and deliver it to you by e-mail.



KMS File Management (Stage 2 Images)

Upload bootloader stage 2(BL2) and OS files for self-signing

Upload

Model: * ARTIK_053s_055s

File name: tinyara_head.bin

Description: Tinyara head bin for ARTIK 055s
.....
31 / 255 characters written

UPLOAD CANCEL

File Management

Success! File "tinyara_head.bin" uploaded.

UPLOAD EDIT DELETE

#	Model	Source ...	Signed File	Sign Key Name	Upload Time	Sign Time	Description
1	ARTIK_053...	tinyara_head.b	tinyara_head.bin-sig	artikaura01-055s-...	2018/02/02 10:02:48	2018/02/02 10:03:05	Tinyara fo
2	ARTIK_530...	logo.png	No Key Available	-	2017/07/24 14:04:16		
3	ARTIK_053...	tinyara_head.b	SIGN	-	2018/02/05 03:46:30		Tinyara h



KMS File Management (Stage 2 Images)

Sign BL2/OS image with generated key, and download the signed BL2/OS images.

File Management

Sign

Model: ARTIK_053s_055s

Source File: tinyara_head.bin

Sign Key Name: * artikaura01-055s-key01

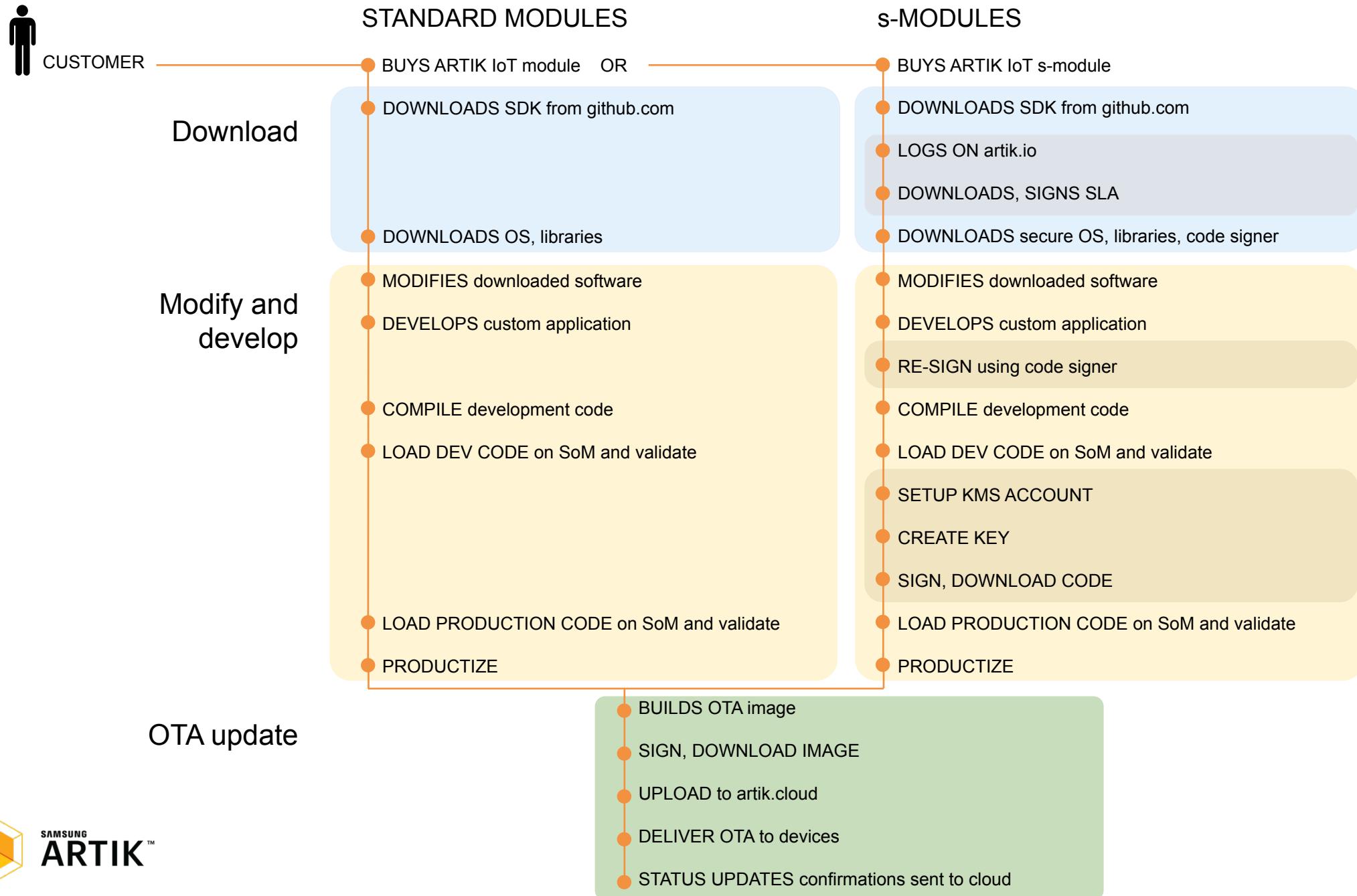
Soft Card Password: *

SIGN CANCEL

File Management

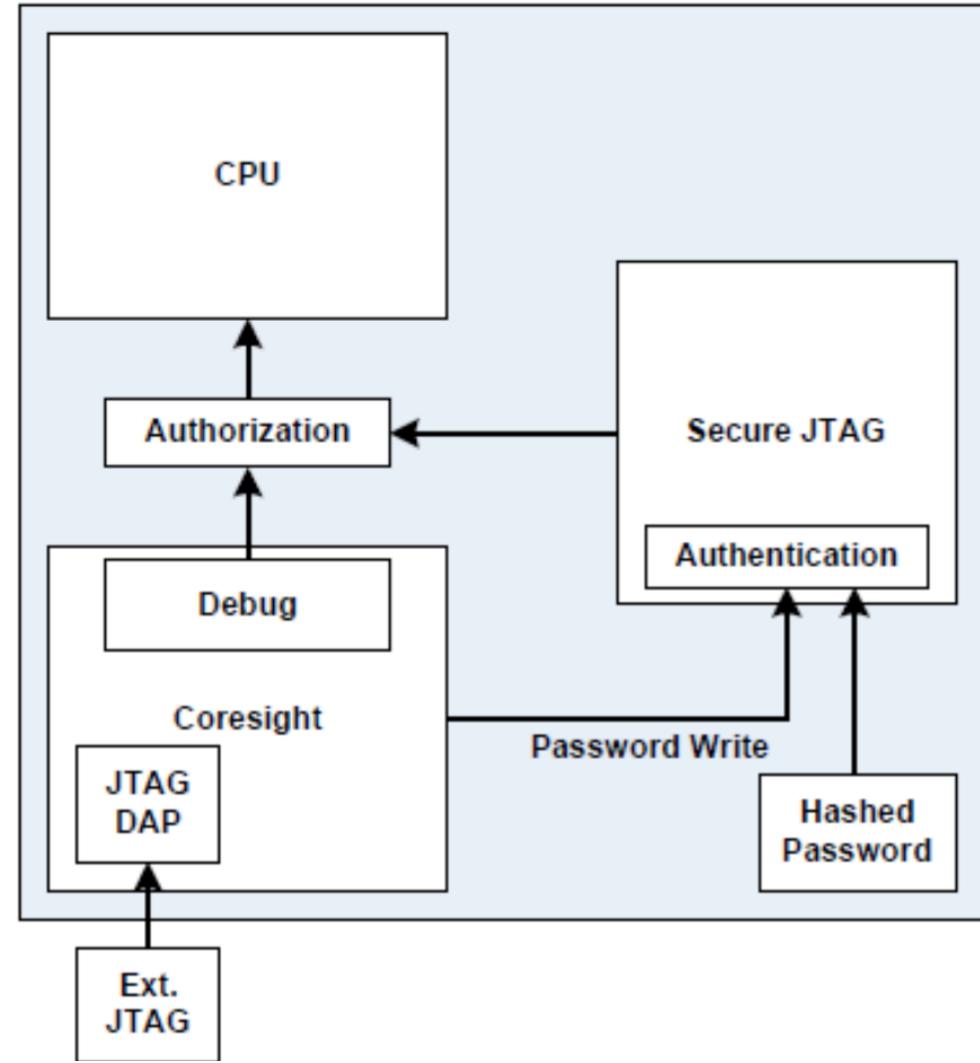
Success! File "tinyara_head.bin" signed.

	UPLOAD	EDIT	DELETE			
	Model	Source ...	Signed File	Sign Key Name	Upload Time	Sign Time
1	ARTIK_053s_055s	tinyara_head.b	tinyara_head.bin-signed	artikaura01-055s-...	2018/02/02 10:02:48	2018/02/02 10:03:05
2	ARTIK_530s_530...	logo.png	No Key Available	-	2017/07/24 14:04:16	
3	ARTIK_053s_055s	tinyara_head.b	tinyara_head.bin-signed	artikaura01-055s-...	2018/02/05 03:46:30	2018/02/05 03:49:31



Secure JTAG

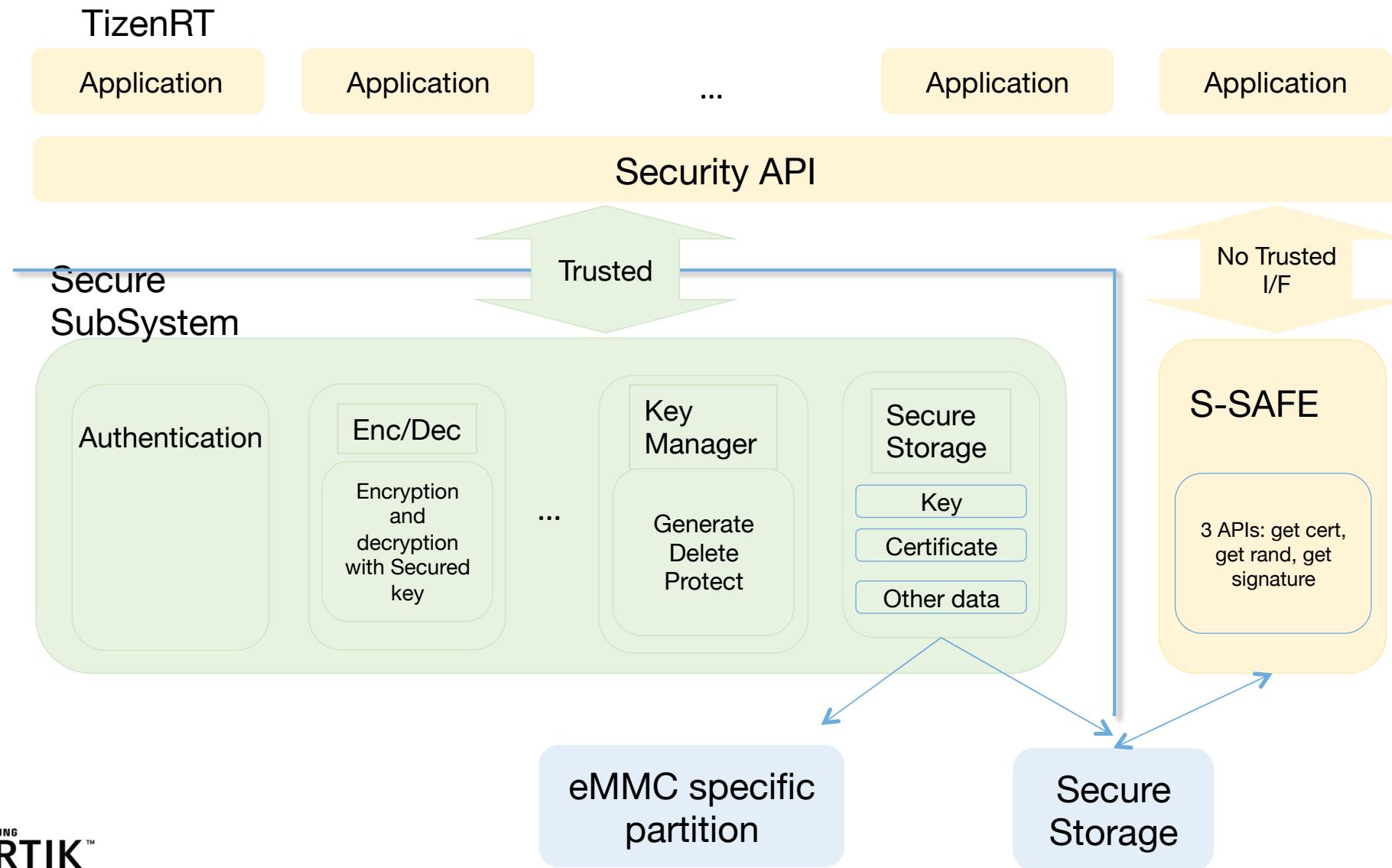
- Secure JTAG authenticates and authorizes JTAG access
- It requires a password to gain access to the JTAG chain. The password is based on the serial number of module
- The information is only made available through an authorized request to Samsung.



Device Protection

		ARTIK module (05x, 5, 7)	ARTIK S-module (053s, 055s, 530s, 710s)	Comments
Secure communication	Per device unique key & certificate	✓	✓	Uniquely identifies device
	Key stored in HW secure element	✓	✓	Secure key storage
	PKI infrastructure: Mutual authentication of device and cloud	✓	✓	Device talks to authorized cloud and vice versa
	Post Provisioning		✓	Provision with your own keys and certificates
Device protection/ secure code execution	KMS infrastructure for code signing		✓	Key Management Service
	Code verification key in HW		✓	Secure key storage
	Secure boot (check for authorized code)		✓	Boot image verification
	JTAG access locked		✓	Lock out debug access
Data protection/ Secure storage	Secure OS (separate normal & secure operations)		✓	Hardware enforced secure applications via TEE
	Security Lib API (27 API calls)	Limited(random number generator, get cert and signature)	✓	Key Manager, Authentication, Secure Storage, Post Provisioning, Encrypt/Decrypt
	Secure storage		✓	Encrypt data stored on Flash

ARTIK SEE Architecture



ARTIK SEE APIs

Category	ARTIK API	Description
Initialize	see_init	
	see_deinit	
Key Management	see_generate_key	generate symmetric and asymmetric keys(AES, ECC Curve, HMAC type)
	see_set_key	set external symmetric and asymmetric key to secure storage
	see_get_pubkey	get public key of asymmetric key from secure storage
	see_remove_key	remove a key from secure storage
Authentication	see_generate_random	Generate a random number
	see_generate_certificate	Generate, set and get a certificate
	see_set_certificate	
	see_get_certificate	
	see_get_rsa_signature	Get , verify signature using RSA, ECDSA algorithm
	see_verify_rsa_signature	
	see_get_ecdsa_signature	
	see_verify_ecdsa_signature	
	see_get_hash,see_get_hmac	Hash Messages
	see_generate_dhparams(ecdhkey)	

ARTIK SEE APIs

Category	ARTIK API	Description
Secure Storage	see_read_secure_storage	Read data from secure storage
	see_write_secure_storage	Write data to secure storage
	see_delete_secure_storage	Remove data from secure storage
	see_get_size_secure_storage	Get data size from secure storage
	see_get_list_secure_storage	List data in secure storage
Post Provision	see_post_provision	Injecting an HMAC key or asymmetric key pair(ECC/RSA) into the secure element
	see_post_provision_lock	
Encryption/ Decryption	see_aes_encryption	AES Encryption/Decryption
	see_aes_decryption	
	see_rsa_encryption	RSA Encryption/Decryption
	see_rsa_decryption	

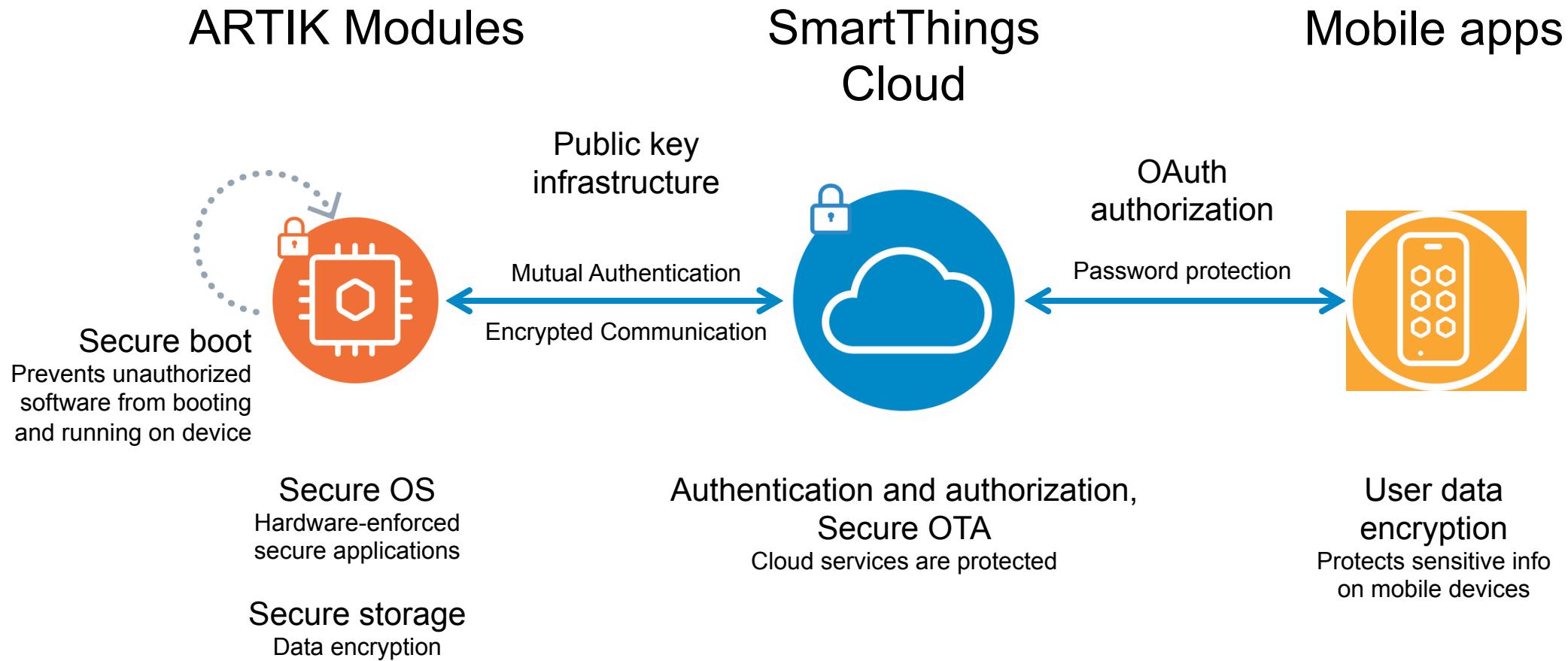
Device Protection

		ARTIK module (05x, 5, 7)	ARTIK S-module (053s, 055s, 530s, 710s)	Comments
Secure communication	Per device unique key & certificate	✓	✓	Uniquely identifies device
	Key stored in HW secure element	✓	✓	Secure key storage
	PKI infrastructure: Mutual authentication of device and cloud	✓	✓	Device talks to authorized cloud and vice versa
	Post Provisioning		✓	Provision with your own keys and certificates
Device protection/ secure code execution	KMS infrastructure for code signing		✓	Key Management Service
	Code verification key in HW		✓	Secure key storage
	Secure boot (check for authorized code)		✓	Boot image verification
	JTAG access locked		✓	Lock out debug access
Data protection/ Secure storage	Secure OS (separate normal & secure operations)		✓	Hardware enforced secure applications via TEE
	Security Lib API (27 API calls)	Limited(random number generator, get cert and signature)	✓	Key Manager, Authentication, Secure Storage, Post Provisioning, Encrypt/Decrypt
	Secure storage		✓	Encrypt data stored on Flash

Platform Security

Samsung ARTIK™ End-to-end Platform Security

End-to-end protection for you and your customers



Samsung ARTIK™ End-to-end Platform Security*

	Feature	ARTIK
Modules	Secure element key storage, secure boot	✓ Included
	Security infrastructure: PKI and KMS	✓ Included
	Unique device ID and certificate	✓ Included
	Secure data storage with data encryption	✓ Included
Platform software	Secure device registration	✓ Included
	Secure OTA updates	✓ Included
Cloud Infrastructure	Supports HIPAA compliant solutions	✓ Included
	OWASP top 10	✓ Included
	Internal and external security audits	✓ Included
Cloud services	AAA (Authentication, Authorization, Accounting)	✓ Included
	API Security	✓ Included
	3 rd party device discovery and mutual authentication	✓ Included
	Data privacy management, identity, permissions,	✓ Included
Communications	TLS, VPN	✓ Included
	DTLS Application level security; BLE session security	✓ Included
Applications	Key and secure app data encryption and storage	✓ Included
	2-factor authentication; OAuth; client side certificates	✓ Included

* Feature list is not exhaustive

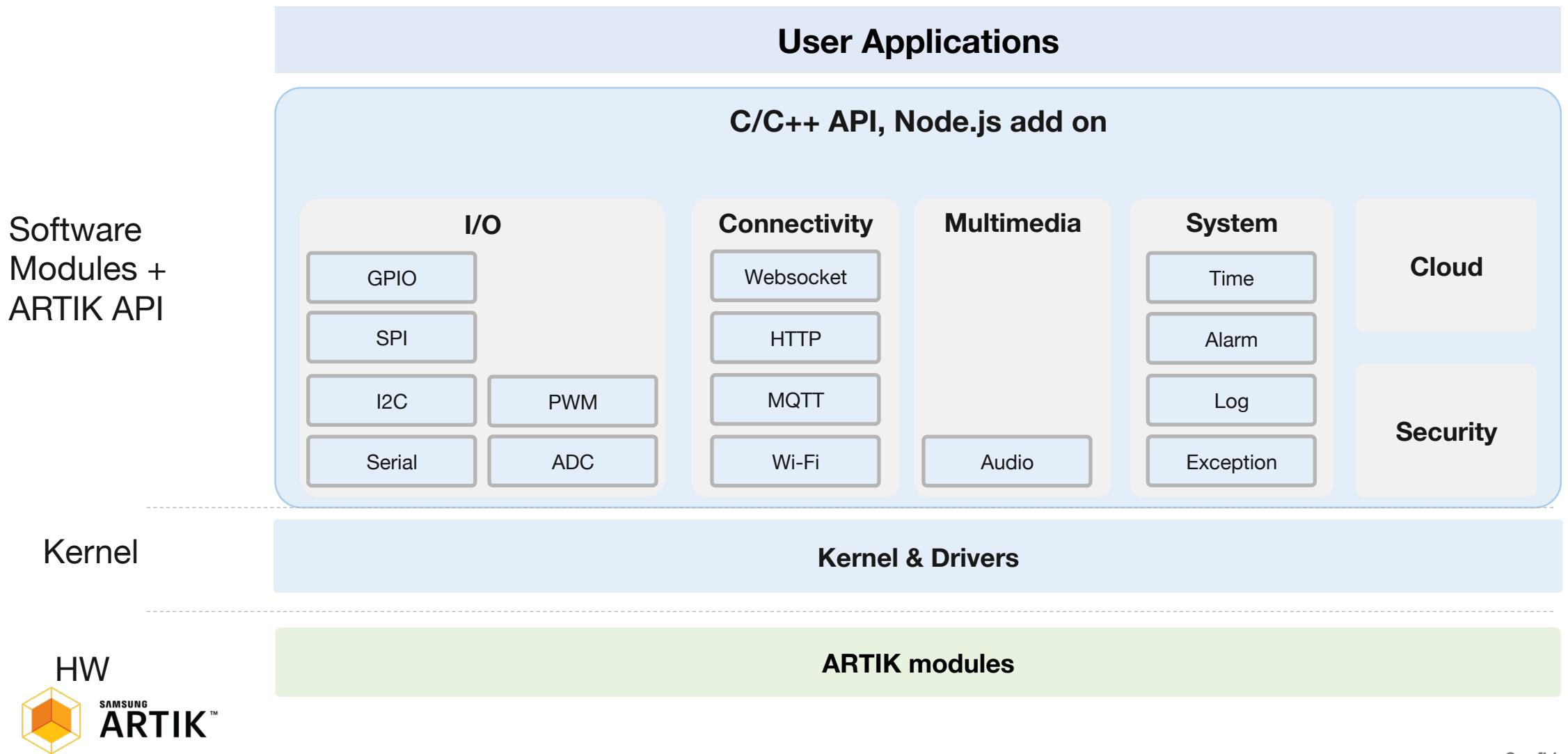
ARTIK 05x

Development

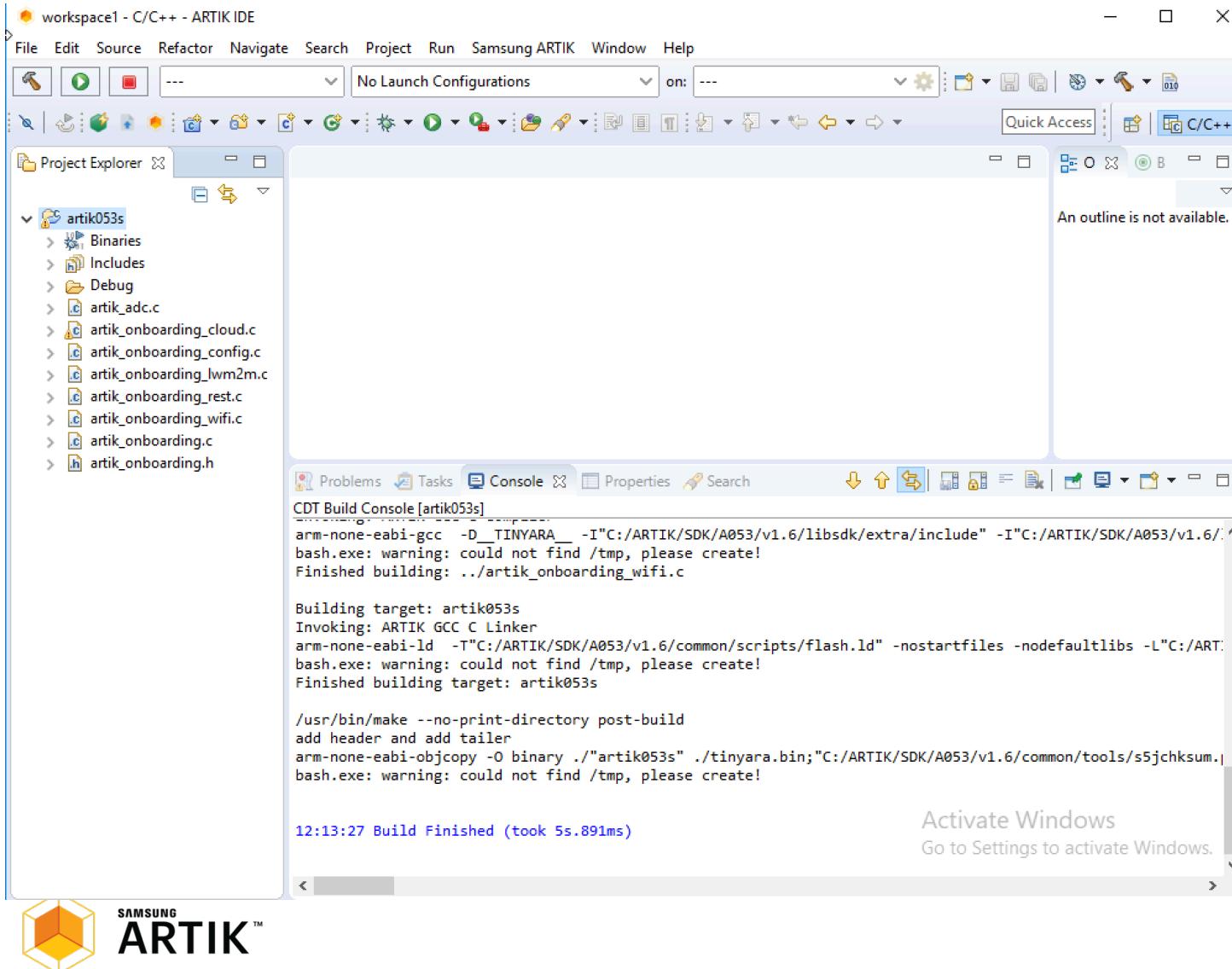
Developing on ARTIK 05x

- ARTIK SDK/IDE (C language)
- Cross-compilation for ARM architecture from command line (C language)
- JerryScript(under development, JS)

ARTIK SDK (05x, 5, 7 series)



ARTIK IDE



gcc-arm-none-eabi

Configuration Files

Typical

OS Kernel
Customized C library
Security(TLS)
BSP(boot loader)
Power
Networking support(LWIP, DHCP C
Watchdog
ARTIK-SDK
JSON
TASH
I2S
DMA
System IO(I2C, UART, GPIO, PWM
Network Utilities(FTP Client, Websc

Extra

OS Kernel
Customized C library
Security(TLS)
BSP(boot loader)
Power
Networking support(LWIP, DHCP Client, DHCP
Watchdog
ARTIK-SDK
JSON
TASH
I2S
DMA
System IO(I2C, UART, GPIO, PWM, SPI)
Network Utilities(FTP Client, Websocket, Webc
CoAP, MQTT)

Minimal

OS Kernel
Customized C library
Security(TLS)
BSP(boot loader)
Power
Networking support(LWIP, DHCP Client, DHCP Server, Wi-Fi)

Cross Compilation from command line

- Github page: <https://github.com/SamsungARTIK/TizenRT>
- How to build:

```
$ git clone https://github.com/SamsungARTIK/TizenRT.git
```

```
$ cd TizenRT/os
```

```
... .
```

```
$ make menuconfig
```

```
$ make
```

JerryScript

- Light weight JavaScript Engine. Base footprint is only 10KB of RAM
- Optimized for microcontrollers
- Portable, can run on bare-metal
- OS Support: Nuttx, RIOT, mbed OS, Zephyr, Linux, macOS

TASH shell

- Github page: <https://github.com/SamsungARTIK/TizenRT/tree/artik/apps/shell>

```
TASH>>help

      TASH command list
-----
cat          cd          date        df
dhcpd        exit        free        heapinfo
help         ifconfig    ifdown     ifup
kill         killall    ls          mkdir
mksmartfs   mount      onboard   ping
ps           pwd         reboot    rm
rmdir       security   sh          sleep
stkmmon     umount     uptime
```

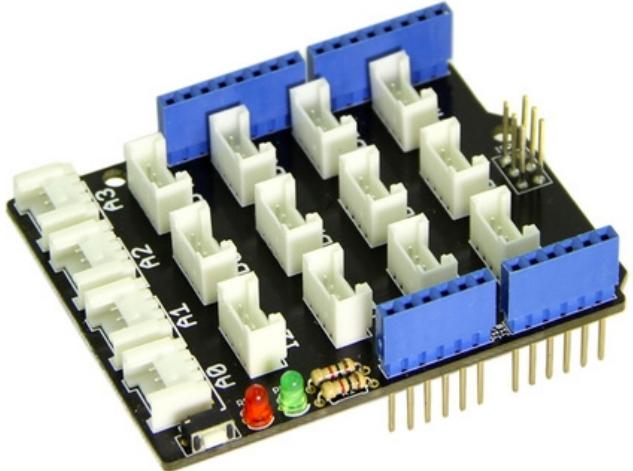
Debugging on ARTIK 05x

The screenshot shows a debugger interface with several windows:

- Top Bar:** Includes standard icons for file operations, search, and navigation.
- Left Sidebar:** Shows a tree view of the project structure under "TizenRT - Debug [GDB OpenOCD Debugging]". The "tinyara" node is expanded, showing "Thread #1 (Suspended : Breakpoint)" which lists several stack frames, including "slsi_wifi_main()", "task_execute_cmd()", and "task_parse_line()". Other nodes include "openocd.exe" and "arm-none-eabi-gdb.exe".
- Registers Window:** A table titled "Registers" showing general purpose and FPU register values. The "General Registers" section is highlighted with a red border. The table includes columns for Name, Value, and Description. Values shown: r0=2, r1=33941096, r2=0, r3=67513348, r4=2, r5=33941096.
- Code Editor:** Displays the source code for "slsi_wifi_main.c". Line 632 contains the instruction "sw_printHeader()", which is highlighted with a red box. The code handles command-line arguments and initializes WiFi callbacks.
- Outline Window:** Shows a list of functions and variables defined in the current module.
- Console Window:** Displays logs from the ARTIK 051 board, indicating it is connected and booting. It shows messages about mailbox registration and LED control initialization.
- Bottom Status Bar:** Shows the TizenRT logo and the command "TASH>>artikwifi startsta".

ARTIK 05x Ecosystem

Grove Base Shield and Modules



Use Case: iCast 2

- Wi-Fi 2.4GHz, BLE 5.0 including 802.15.4 Radio and Thread SW stack
- Analog I/O, GPIO, I2C etc.
- Isolated RS485 serial port
- Supports ARTIK Cloud, AWS IoT, PTC ThingWorx



Use Case: Location Service

Indoor and Outdoor Location Capability from Comtech
Leveraging Wi-Fi Connectivity with Built-In
Flexibility and Security from Samsung ARTIK

Built specifically for IoT applications:

- Fully integrated with ARTIK IoT Platform and ARTIK Cloud
- Power, memory, and bandwidth efficient
- Flexible deployment models



COMTECH
TELECOMMUNICATIONS CORP.

SAMSUNG ARTIK™



3rd Party Cloud Support

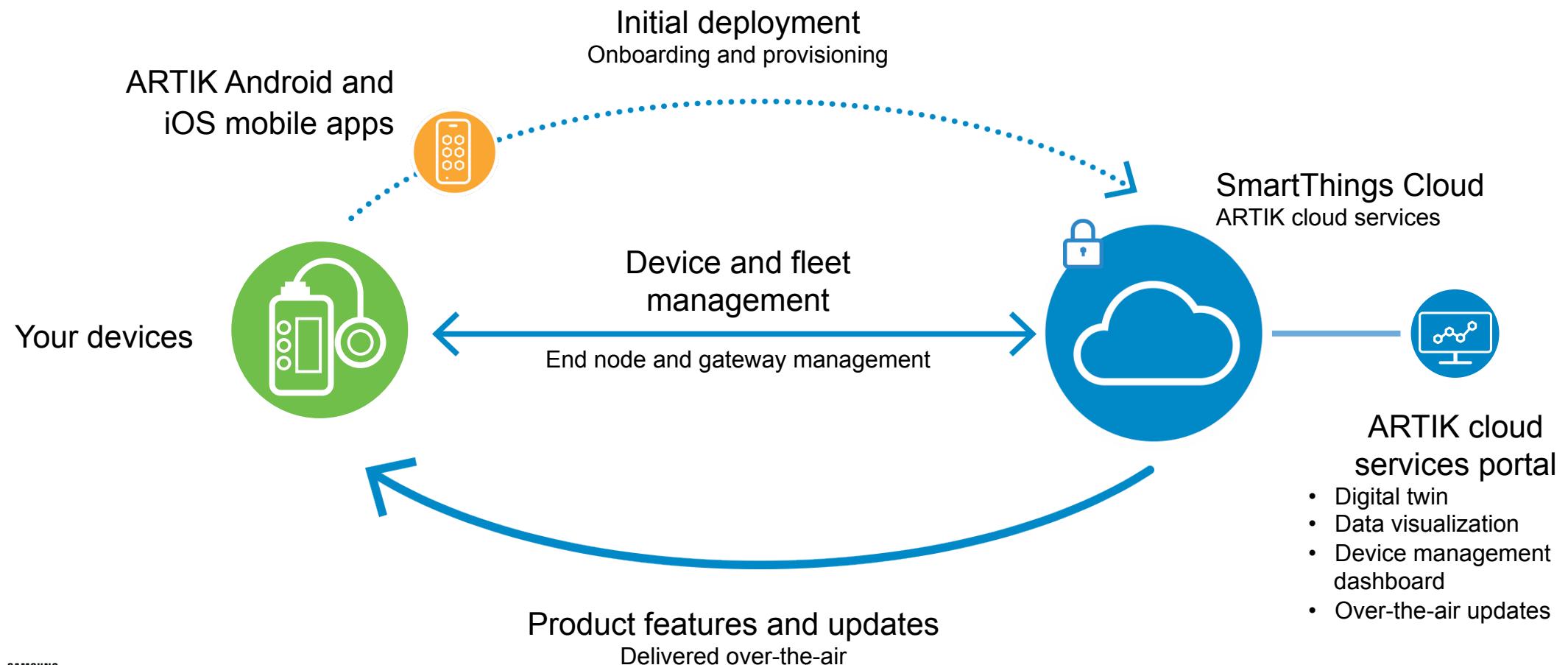
1. AWS IoT:
2. Microsoft Azure:
3. PTC ThingWorx:
4. Other Cloud services that support REST API, MQTT etc.

Lab Sessions

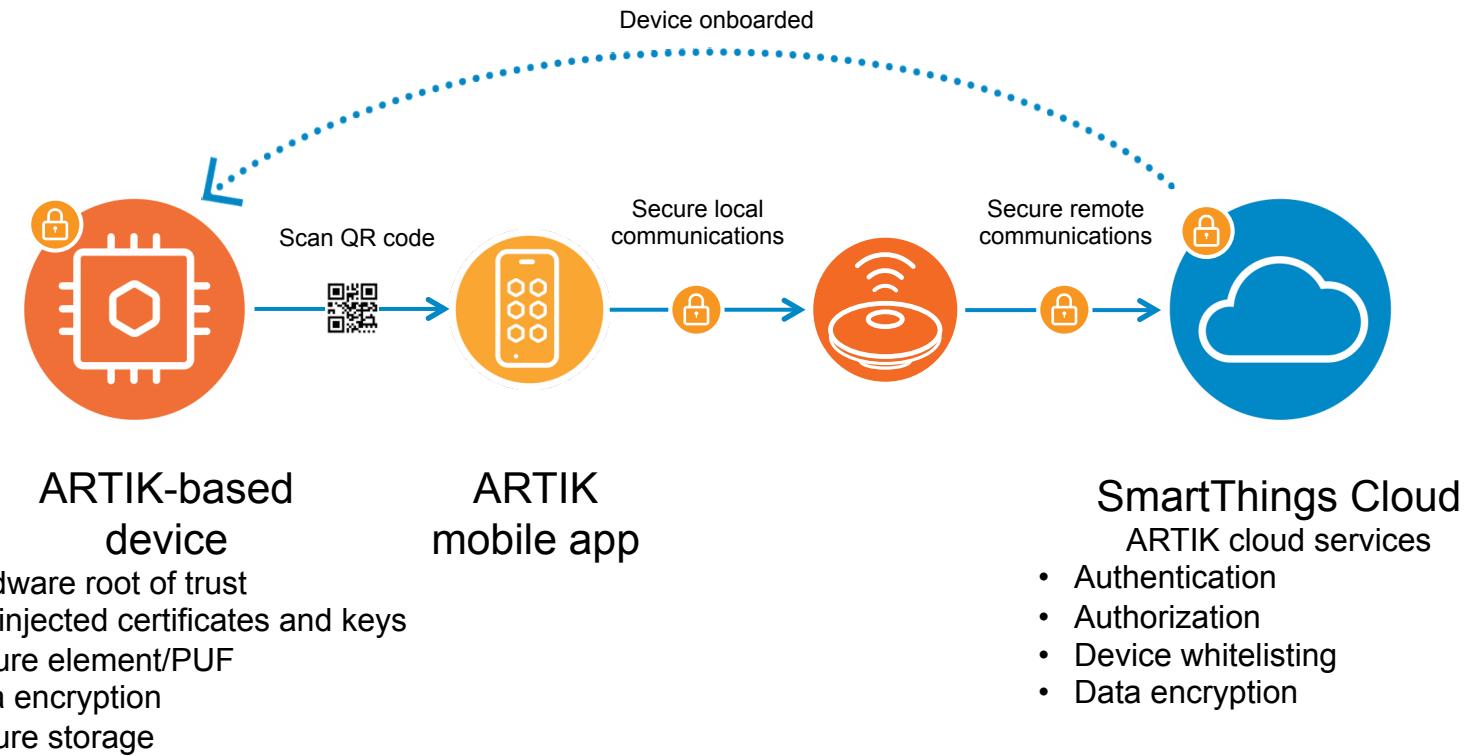
Lab 2: ARTIK 05x Onboarding

Samsung ARTIK™ Device management and OTA

Onboard, manage, and service devices in the field

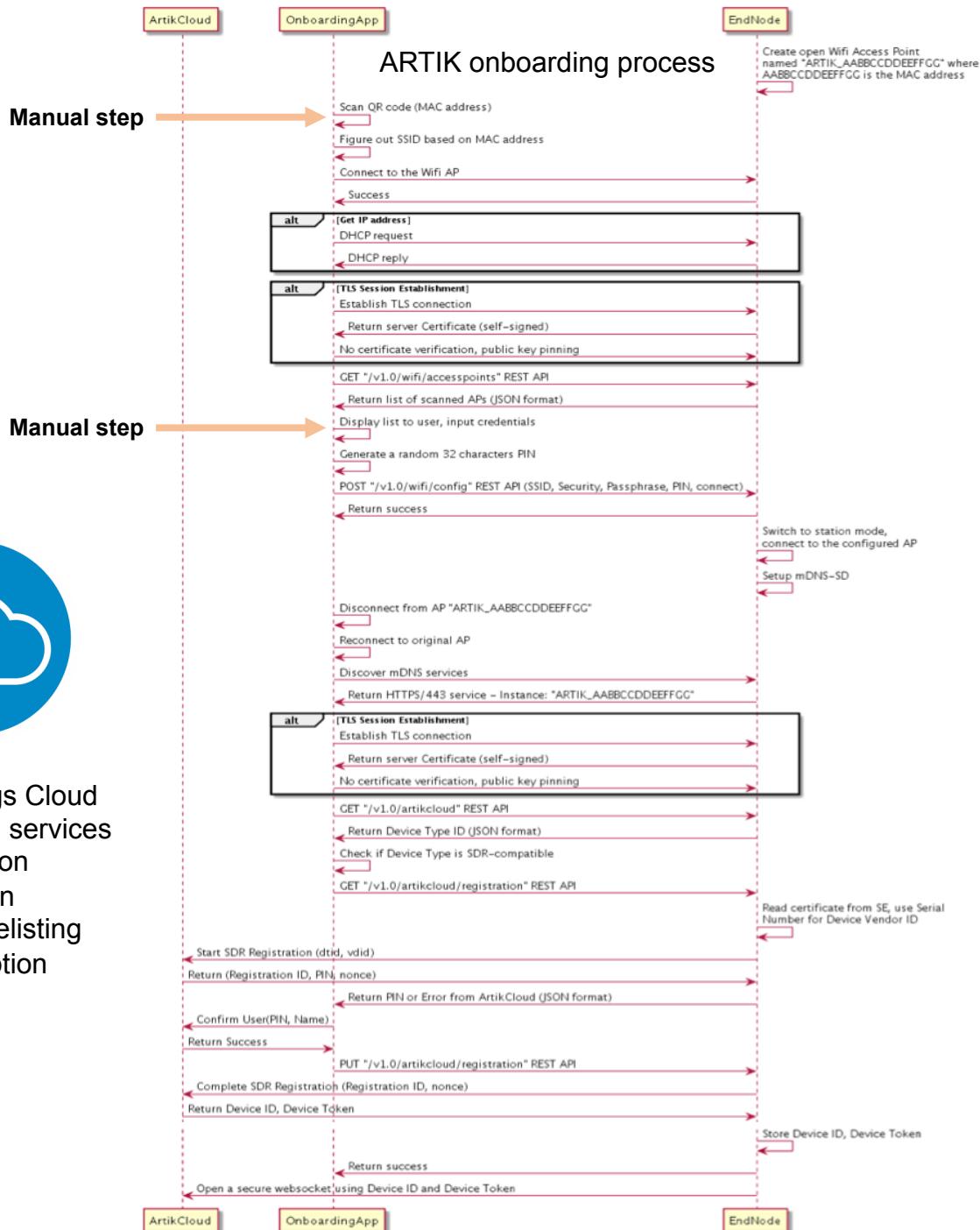
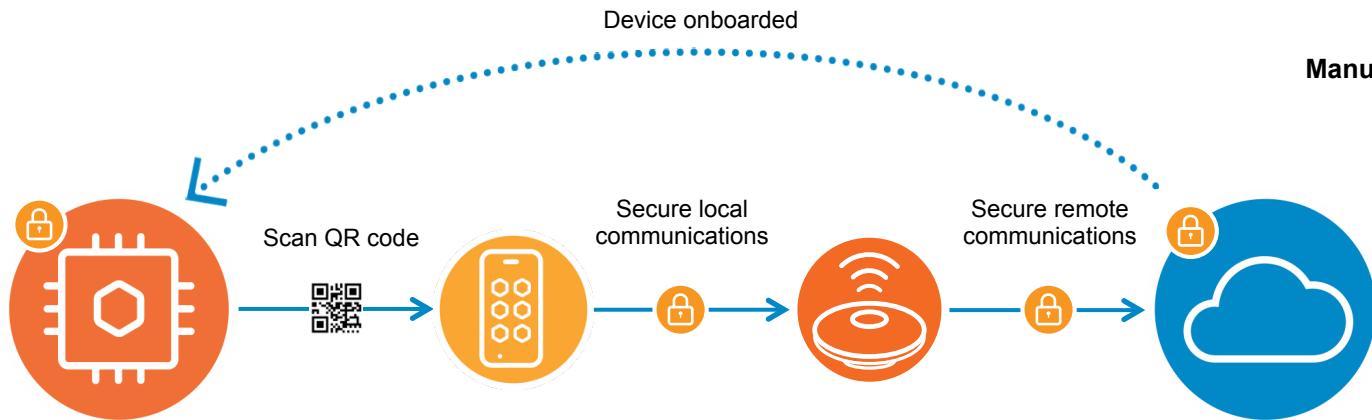


Samsung ARTIK™: Easy, secure onboarding



Samsung ARTIK™

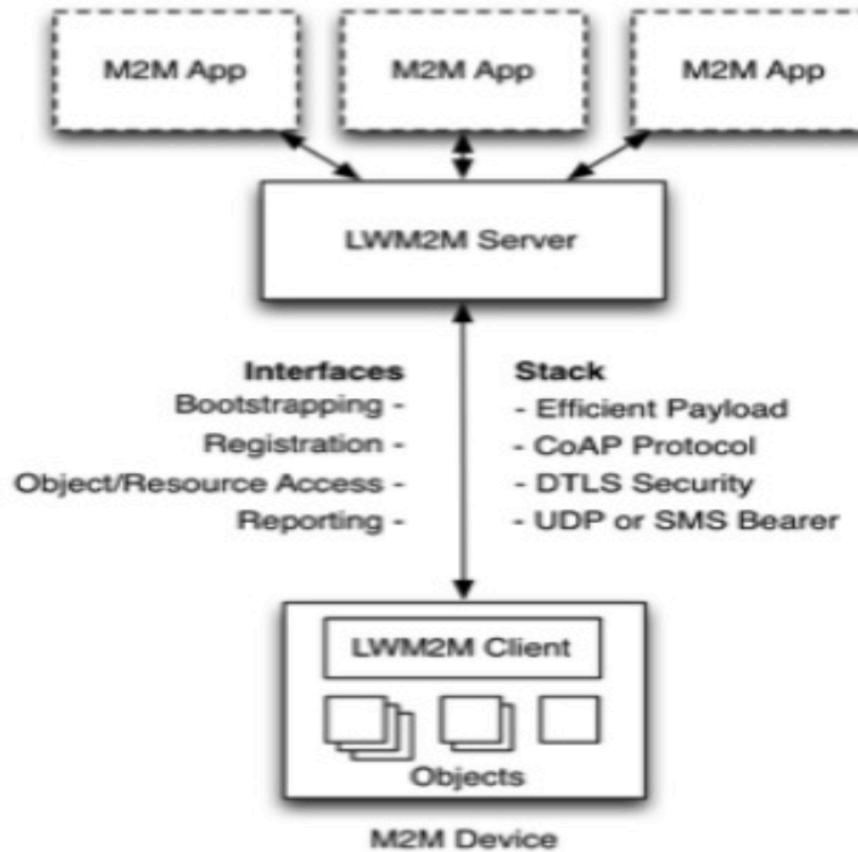
Easy, secure onboarding



Lab 4: ARTIK 05x

Device Management and OTA

LWM2M Architecture Overview



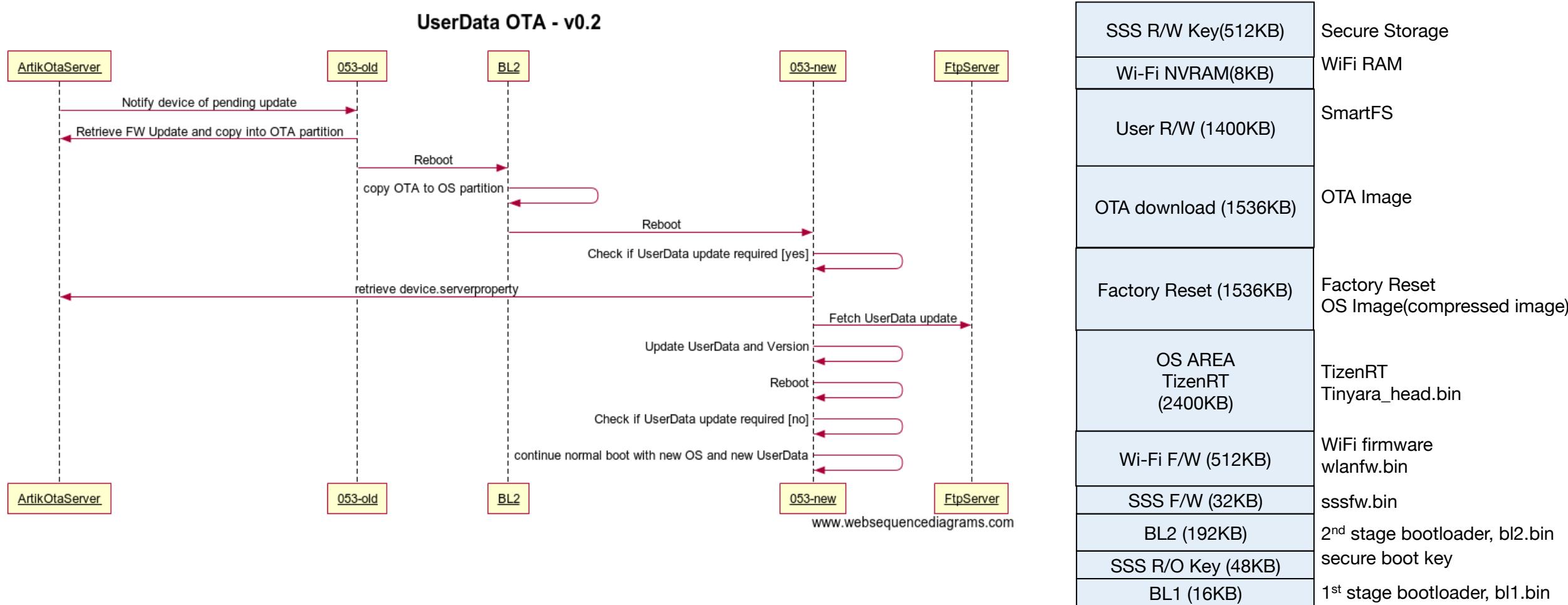
LWM2M server

- Persistent endpoint through which devices and apps interact
- Deployable on gateways and/or in the cloud

LWM2M client

- Hosts resources(objects) that represent a physical device

OTA Sequence



Lab 5: ARTIK 05x Security API

Register Commands to TASH shell

- Register commands to TASH

```
int task_cmd_install(const char *str, TASH_CMD_CALLBACK cb, int thread_exec);  
void task_cmdlist_install(const task_cmdlist_t list[]);
```

- Two modes of execution:

TASH_EXECMD_SYNC – invoked callback in TASH task

TASH_EXECMD_ASYNC – execute callback as a separate task

Q & A