Manufacture stage

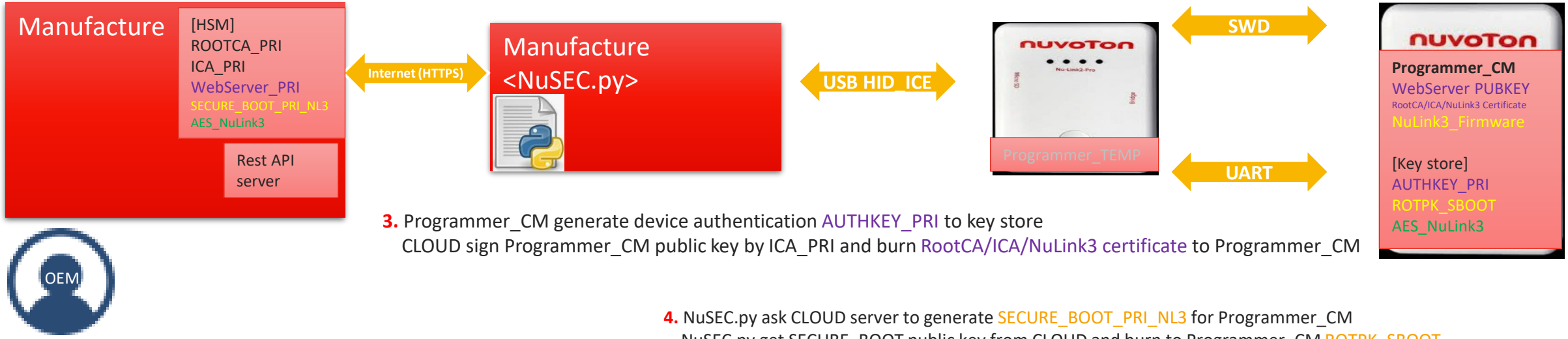
Cloud Server | Desktop PC | Adapter | target board

Prepare NuLink3 programmer for CM (by OEM)

0. OEM server setup
Rest API server

1. Generate private key for ROOTCA, ICA and WebServer in HSM
Generate certificate of ROOTCA (self signed), ICA (chain of trust),
Generate WebServer PUBKEY

2. OEM get a blank NuLink3 as Programmer_CM
Program WebServer PUBKEY to Programmer_CM (SWD)
Write DevAuth.bin to SRAM of Programmer_CM (SWD)
Run DevAuth.bin and Programmer_CM talk with host



3. Programmer_CM generate device authentication AUTHKEY_PRI to key store
CLOUD sign Programmer_CM public key by ICA_PRI and burn RootCA/ICA/NuLink3 certificate to Programmer_CM

The RootCA is owned by OEM
ICAs belong to different projects of OEM
WebServer key pair is for device to authenticate web server

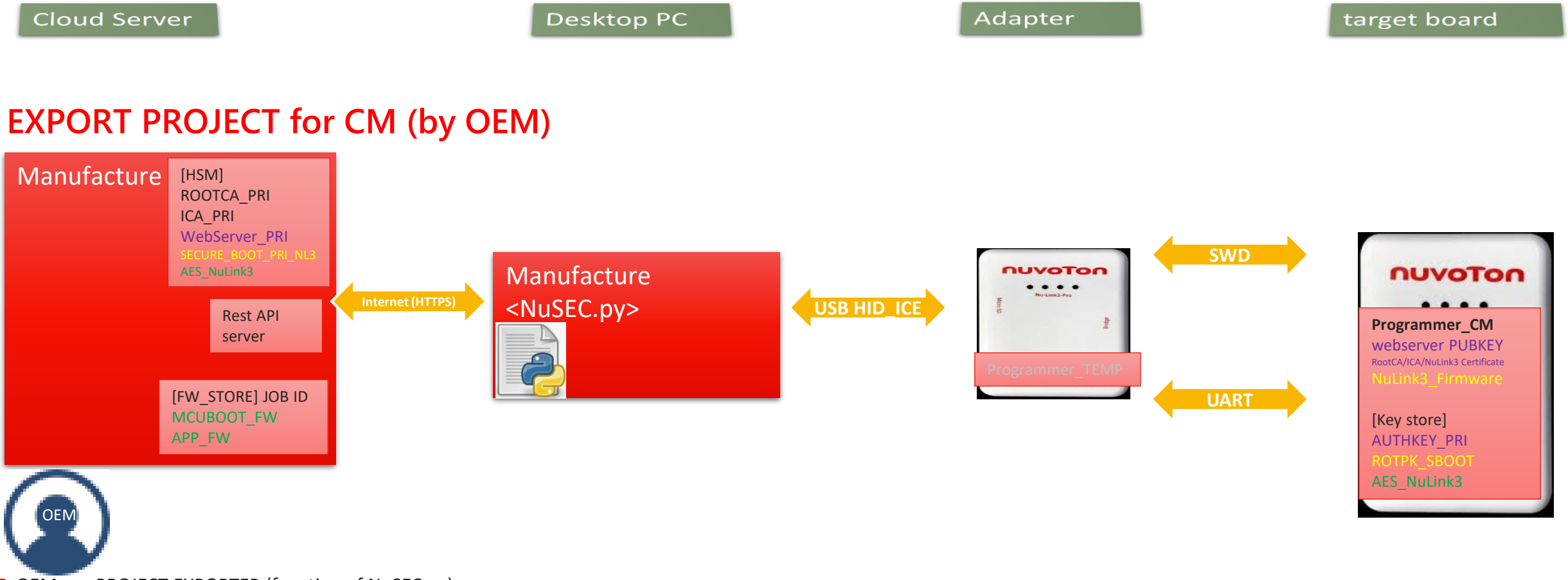
4. NuSEC.py ask CLOUD server to generate SECURE_BOOT_PRI_NL3 for Programmer_CM
NuSEC.py get SECURE_BOOT public key from CLOUD and burn to Programmer_CM ROTPK_SBOOT
NuSEC.py send NuLink3 firmware to cloud server and HSM signed it by SECURE_BOOT_PRI_NL3
NuSEC.py install NuLink3 firmware (signed) to Programmer_CM
Enable secure lock of Programmer_CM

5. Run NuLink3 firmware, Generate AES_NuLink3 by ECDH between HSM and Programmer_CM ***(note1)***

6. OEM send Programmer_CM adapter to CM as secure programmer

(note1) AES_NuLink3 may derive from TRNG, encrypted by "WebServer PUBKEY" and decrypted by "WebServer_PRI" in cloud

EXPORT PROJECT for CM (by OEM)



7. OEM use PROJECT EXPORTER (function of NuSEC.py) make package for CM, the package include:

- Production count
- JOB ID
- IDs to check (e.g. DID of target device)

8. Encrypt package by AES_NuLink3, OEM send package to CM
OEM upload target firmware (MCUBOOT: BL2, APP_FW: BL3) to FW_STORE server
Bind uploaded firmware with corresponding Job ID

9. OEM send package to CM by Email

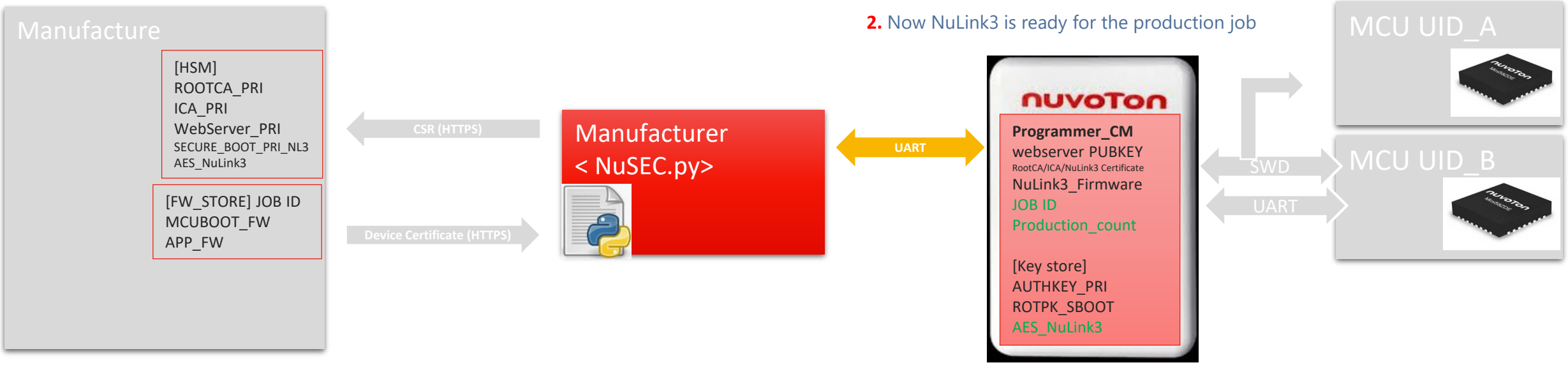
Import OEM package (CM)

Cloud Server | Desktop PC | Adapter (NuLink3) | target board

0. CM use package importer function of NuSEC.py to import OEM package to NuLink3

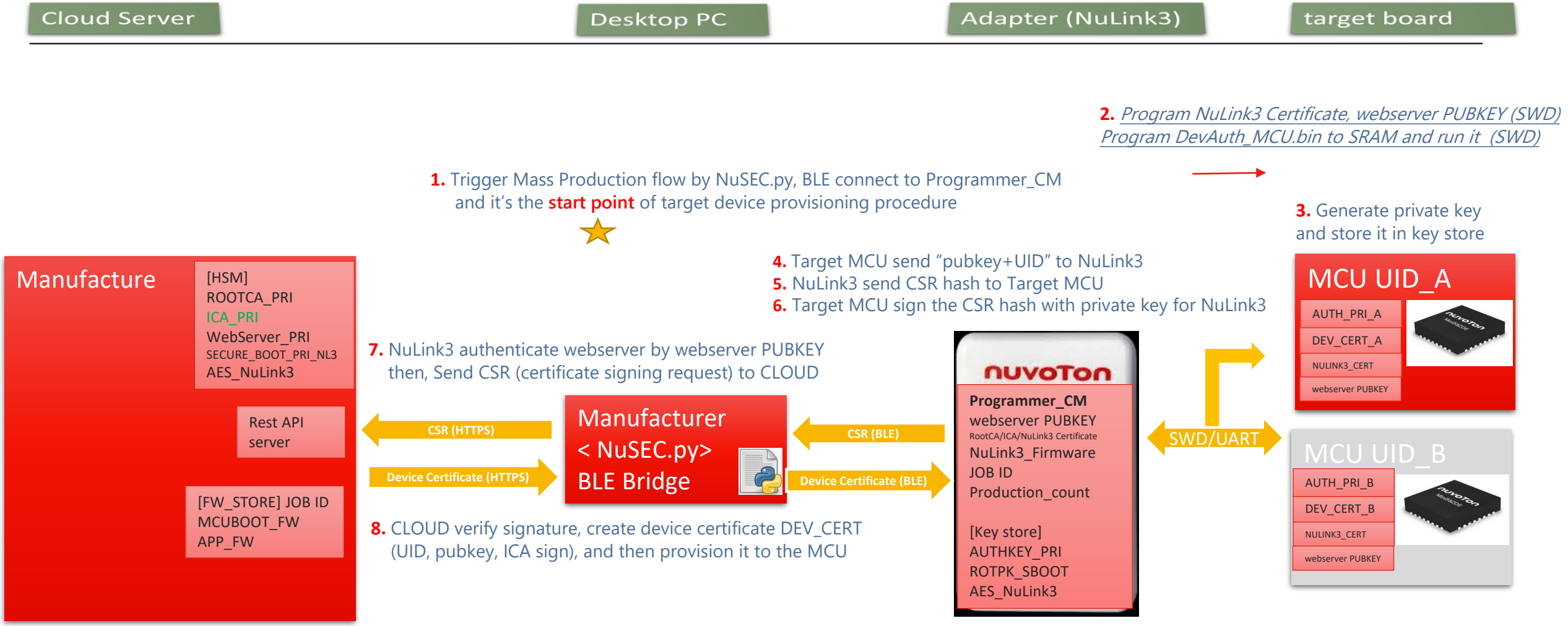
1. The package is processed in NuLink3 SECURE ENVIRONMENT to ensure authenticity and integrity after transporting from OEM to NuLink3.
The package is then decrypted (by AES_NuLink3) in NuLink3 SECURE ENVIRONMENT and production_count/Job_ID/.. are programmed in the NuLink3

2. Now NuLink3 is ready for the production job



3. For each provision.
NuLink3 decrease the production count number
Read device UID and communicate with cloud server
Device authentication provision (the following slide)
Firmware installation will be rejected if device authentication fail
Secure boot key provision and FW installation (the following slide)

Device authentication key provision



Public key: 512 bits (64B)
Unique ID: 128 bits (16B)
Certificate: 16000 bits (2000bytes)

Firmware attestation - secure boot key and FW install

Cloud Server | Desktop PC | Adapter (NuLink3) | target board

8. Use ECDH to derive symmetric AES_SESSION key *(note1)*

10. MCUBoot get APP_FW with signature, decrypt it by AES_SESSION program to BL3 address.

9. NuLink3 attach signature_BL3 and PK_BL3_A to APP_FW
NuLink3 encrypts APP_FW (BL3) by AES_SESSION then send it to MCUBoot firmware running on target (via UART/I2C..interface)

Manufacturer

[FW_STORE] JOB ID
MCUBOOT_FW
APP_FW

Rest API server

[HSM]
ROOTCA_PRI
ICA_PRI
WebServer_PRI
SECURE_BOOT_PRI_NL3
AES_NuLink3

ID	FwVer	PUB KEY	secure boot PRI KEY (HSM)
UID_A	V1	ROTPK_A PK_BL3_A	SECURE_BOOT_PRI_A SECURE_BOOT_PRI_A_BL3
UID_B		ROTPK_B	SECURE_BOOT_PRI_B
UID_C		ROTPK_C	SECURE_BOOT_PRI_C

1. *Continue running NuLink3 firmware*
NULINK3 and TARGET MCU mutual authentication
get UID_A (SWD)

2. NuLink3 tell cloud about UID_A, HSM generate SECURE_BOOT_PRI_A and ROTPK_A

Internet (HTTPS) | NuSEC.py BLE Bridge | BLE | nuvoTon: MCUBoot_FW, APP_FW, Programmer_CM | SWD/UART/I2C.. | SWD/UART/I2C..

MCU UID_A: ROTPK_A, AUTH_PRI_A, DEV_CERT_A, NULINK3_CERT

MCU UID_B: ROTPK_B, AUTH_PRI_B, DEV_CERT_B, NULINK3_CERT

3. Cloud sign MCUBoot_FW with SECURE_BOOT_PRI_A
4. Cloud send APP_FW, MCUBoot_FW, signature_BL2 to NuLink3 and store them. <**ONCE FOR ONE JOB ID**>
5. *NuLink3 program MCUBoot_FW (with signature_BL2) and ROTPK_A to MCU_UID_A (SWD)*
6. Cloud gen key and sign APP_FW (BL3) with SECURE_BOOT_PRI_A_BL3, send signature_BL3 and PK_BL3_A to NuLink3

7. *NuLink3 issue Secure_Lock & chip reset to MCU_UID_A (SWD)*
MCU_UID_A run from SecureBoot -> MCUBoot (BL2 with firmware upgrade function)

11. NuLink3 notify the could MCU UID_A has been successful provisioned with FwVer v1. **End point**

12. Loop until production counter decrease to zero, each target MCU has being provisioned from **start point** to the **end point** by the NuLink3 system.
13. MCUBoot_FW and APP_FW on NuLink3 will be erased. Traceability log table can be found on the cloud

(note1) AES_SESSION may derive from AUTH_PRI_A and DEV_CERT_A

Status report

Cloud Server | Desktop PC | Adapter (NuLink3) | target board

Manufacture

[HSM]
ROOTCA_PRI
ICA_PRI
WebServer_PRI
SECURE_BOOT_PRI_NL3
AES_NuLink3

[FW_STORE] JOB ID
MCUBOOT_FW
APP_FW

ID	FwVer	PUB KEY	secure boot PRI KEY (HSM)
UID_A	V1	ROTPK_A PK_BL3_A	SECURE_BOOT_PRI_A SECURE_BOOT_PRI_A_BL3
UID_B	v1	ROTPK_B	SECURE_BOOT_PRI_B
UID_C	v1	ROTPK_C	SECURE_BOOT_PRI_C

Internet (HTTPS)

Manufacturer
< NuSEC.py>

1. OEM/CM use NuSEC.py to dump the table of provision status, and firmware version of each device

SWD

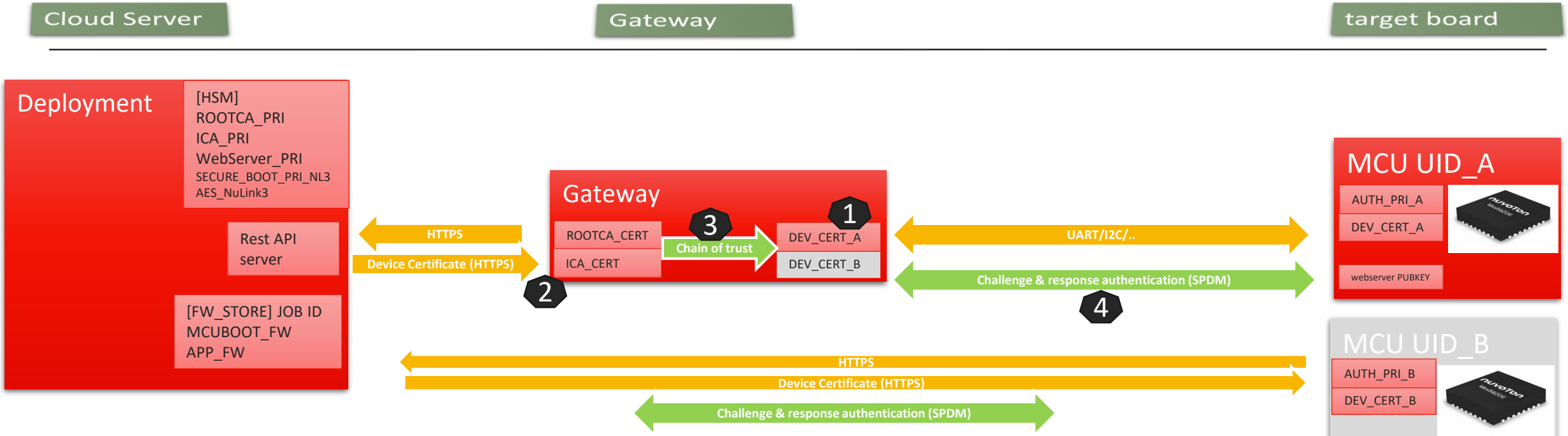
UART

MCU UID_A

MCU UID_B

Deployment stage

Device authentication



If devices are online:
Cloud can manage & track the issued, expired, and revoked certificates

Maintenance stage

Firmware upgrade (NuSEC.py + NuLink3 bridge)

Cloud Server | Desktop PC | Adapter (NuLink3) | target board

4. Use ECDH to derive symmetric AES_SESSION key

5.
NuLink3 and Target MCU mutual authenticate (by DEV_CERT/ROOTCA/ICA cert, NULINK3_CERT)
NuLink3 check firmware rollback
NuLink3 attach signature_BL3 and PK_BL3_A to APP_FW2
NuLink3 encrypts APP_FW2 by AES_SESSION then send it to MCUBoot firmware running on target (via UART/I2C..interface)

6. MCUBoot get APP_FW with signature decrypt it by AES_SESSION program to BL3 address.

Manufacturer

[FW_STORE] JOB ID
MCUBOOT_FW
APP_FW2

Rest API server

[HSM]
ROOTCA_PRI
ICA_PRI
WebServer_PRI
SECURE_BOOT_PRI_NL3
AES_NuLink3

ID	FwVer	PUB KEY	secure boot PRI KEY (HSM)
UID_A	V2 +1 after complete	ROTPK_A PK_BL3_A	SECURE_BOOT_PRI_A SECURE_BOOT_PRI_A_BL3
UID_B	V1	ROTPK_B	SECURE_BOOT_PRI_B
UID_C	V1	ROTPK_C	SECURE_BOOT_PRI_C

0. Put upgrade **APP_FW2** to FW_STORE server

1. Trigger FW upgrade by NuSEC.py,
NuLink3 run FW upgrade application,
NuLink3 and cloud mutually authenticate each other
get UID_A and firmware version from device

2. NuLink3 tell cloud about UID_A, NuSEC know UID_A FwVer is V1 and need update

Internet (HTTPS) → NuSEC.py BLE Bridge → BLE → nuvoTon

APP_FW2
Programmer_CM
webserver PUBKEY
[Key store]
AUTHKEY_PRI

SWD/UART/I2C.. → MCU UID_A

SWD/UART/I2C.. → MCU UID_B

3. Cloud send APP_FW2 <**ONCE, if batch update**>
Cloud sign **APP_FW2** with SECURE_BOOT_PRI_A_BL3, send signature_BL3 and PK_BL3_A to NuLink3.

Internet (HTTPS) → Target MCU C

MCU UID_A

ROTPK_A
AUTH_PRI_A
DEV_CERT_A
NULINK3_CERT

MCU UID_B

ROTPK_B
AUTH_PRI_B
DEV_CERT_B
NULINK3_CERT

Target MCU C

(note1) AES_SESSION may derive from AUTH_PRI_A and DEV_CERT_A

Firmware OTA upgrade (direct)

Cloud Server | Desktop PC | Adapter (NuLink3) | target board

Manufacturer

[FW_STORE] JOB ID
MCUBOOT_FW
APP_FW2

Rest API server

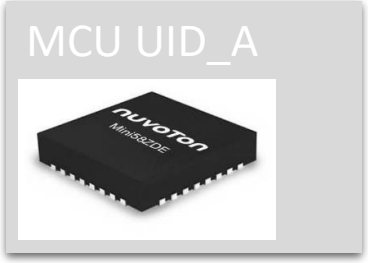
[HSM]
ROOTCA_PRI
ICA_PRI
WebServer_PRI
SECURE_BOOT_PRI_NL3
AES_NuLink3

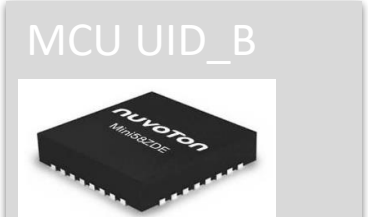
ID	PUB KEY	secure boot PRI KEY (HSM)
UID_A	ROTPK_A PK_BL3_A	SECURE_BOOT_PRI_A SECURE_BOOT_PRI_A_BL3
UID_B	ROTPK_B	SECURE_BOOT_PRI_B
UID_C	ROTPK_C	SECURE_BOOT_PRI_C

Put upgrade **APP_FW2** to FW_STORE server

SWD/UART/I2C..

SWD/UART/I2C..

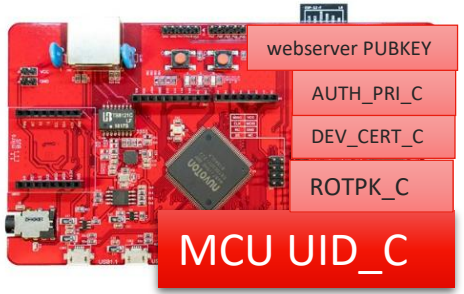




MCUBoot get APP_FW with signature, decrypt it by AES_SESSION program to BL3 address.

HTTPS/MQTTS

Device and cloud mutual
Cloud prevent firmware rollback
Cloud attach signature_BL3 and PK_BL3_A to APP_FW2
Cloud use ECDH to derive symmetric AES_SESSION key
Cloud encrypts APP_FW2 by AES_SESSION then send it to MCUBoot



(note1) AES_SESSION may derive by other method

Decommission

Decommission

Cloud Server | Desktop PC | Adapter (NuLink3) | target board

Manufacturer

[FW_STORE] JOB ID
MCUBOOT_FW
APP_FW2

Rest API server

[HSM]
ROOTCA_PRI
ICA_PRI
WebServer_PRI
SECURE_BOOT_PRI_NL3
AES_NuLink3

ID	Revoke cert
UID_A	
UID_B	revoked
UID_C	revoked

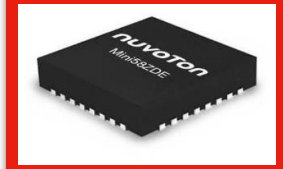
Decommission device

Internet (HTTPS) → NuSEC.py BLE Bridge → BLE → NuLink3 → UART

SWD/UART/I2C...

MCU UID_A

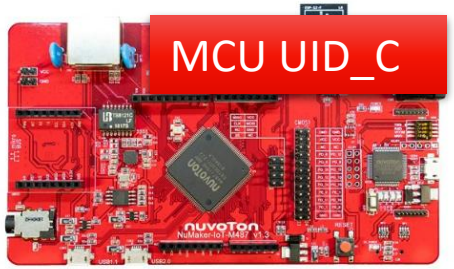
MCU UID_B



HTTPS/MQTTS

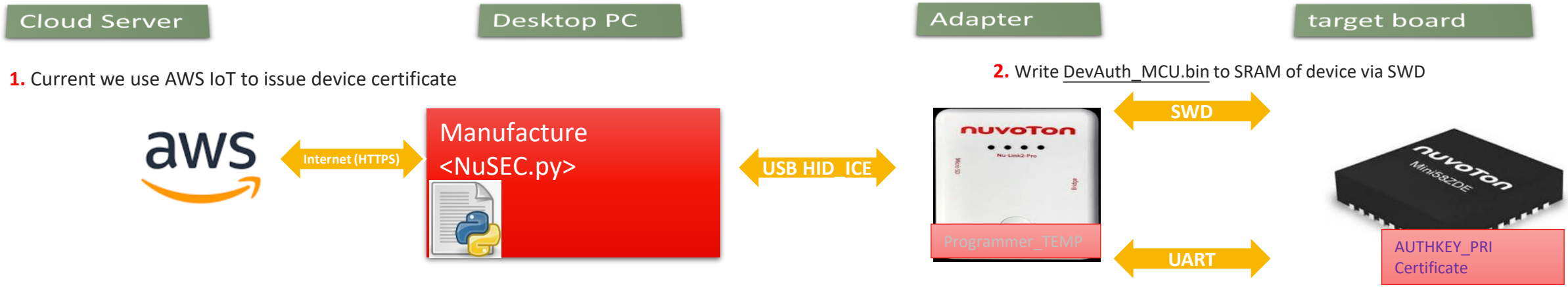
Decommission device

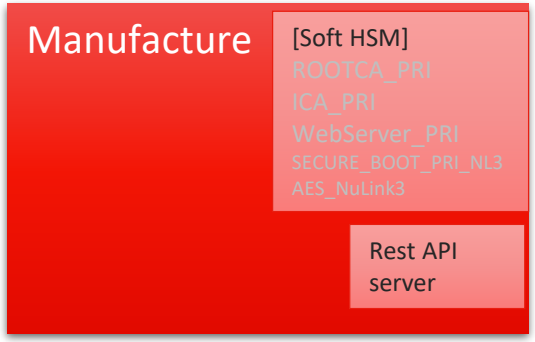
MCU UID_C



Current status

- Current status (Coworking with A008 JY33)
  - Device authentication: AWS <-> NuSEC.py <-> NuLink2 <-> (M2354,KM1M7C)





We setup server for next step
(use Soft HSM as HSM simulator)

3. Target MCU generate device authentication AUTHKEY_PRI to key store (UART command)
AWS sign Target MCU public key and burn certificate back to target MCU

UART protocol of DevAuth_MCU.bin is standardized.
It can also communicates with other devices. (e.g. DATA IO)

Joy of innovation

nuvoTon

Thank You

Danke

Merci

ありがとう

Gracias

Kiitos

감사합니다

धन्यबाद

ك ل ا ر ك ش

הדות