Secure Update

SYSGO
ESCRYPT

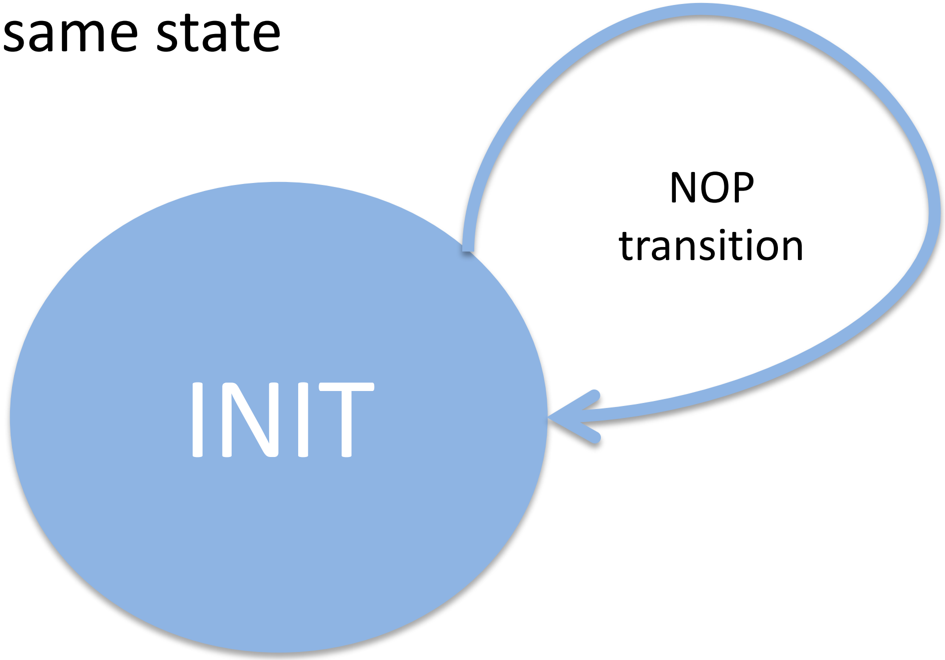
"This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 644080.”

SAFURE

SAFety and secURity by dEsign for interconnected mixed-critical cyber-physical systems

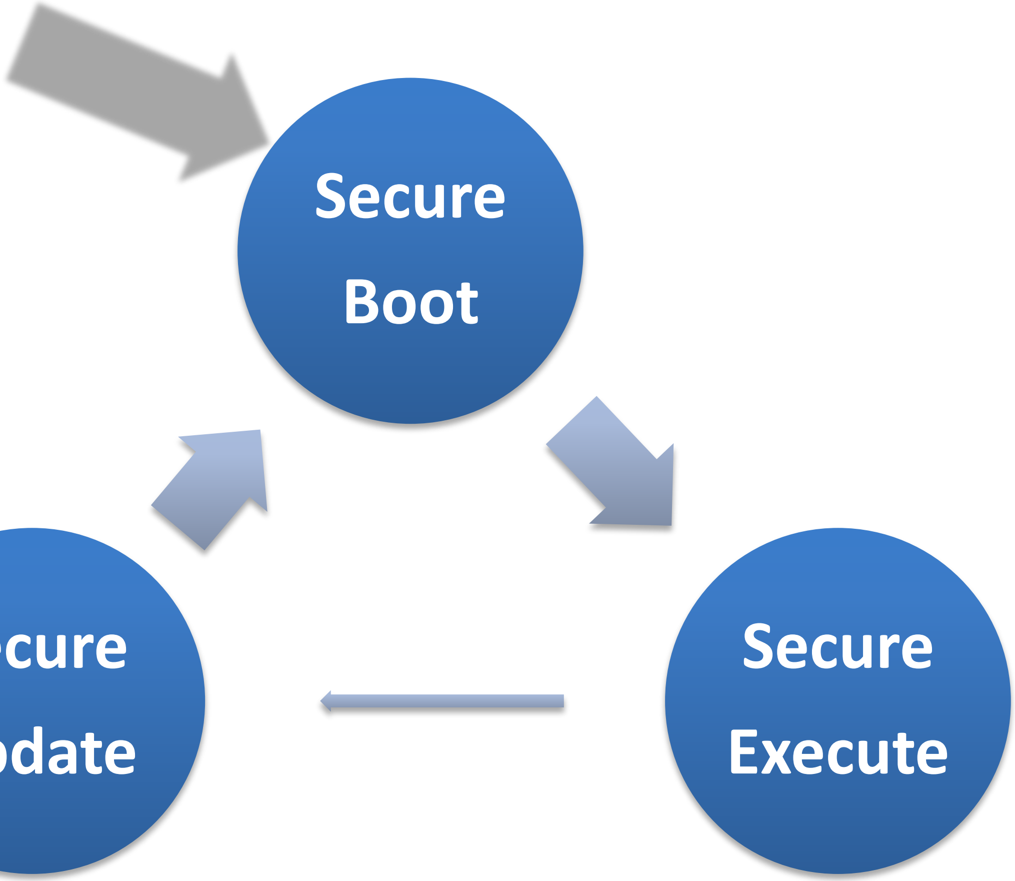
Perfectly Secure Product

It is secure because it's doing nothing,
i.e. it stays in the same state



ONLY IF INITIAL STATE IS SECURE

Secure Lifecycle for System in the Field



The Goal

- Guarantee authenticity and integrity of the application image
- Provide protection against attacks about smuggling malicious applications
- Provide flexibility for system functionality life-cycle and in-the-field update

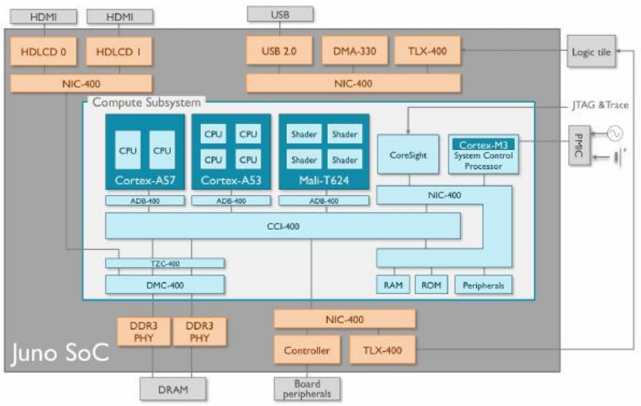
Highlights

- Operating over network
- Keeping already operating devices updated
- Higher safety and security levels
  - Provide critical patches as soon as they are ready
- Provide best customer experience
  - Deliver new or update existing features

Demo target hardware

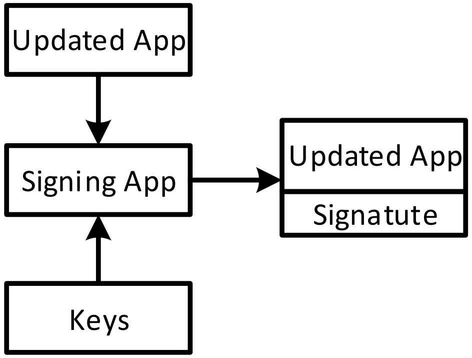
- Design is platform agnostic

- This demo on Juno Board:
  - Cortex®-A57 & Cortex-A53 MPCore
  - ARMv8, big.LITTLE™, 64bit
  - PikeOS 4.2 fully supported
    - HW Virtualization
    - Trust Zone
    - Serial and Ethernet drives
    - ElinOS BSP



Signing

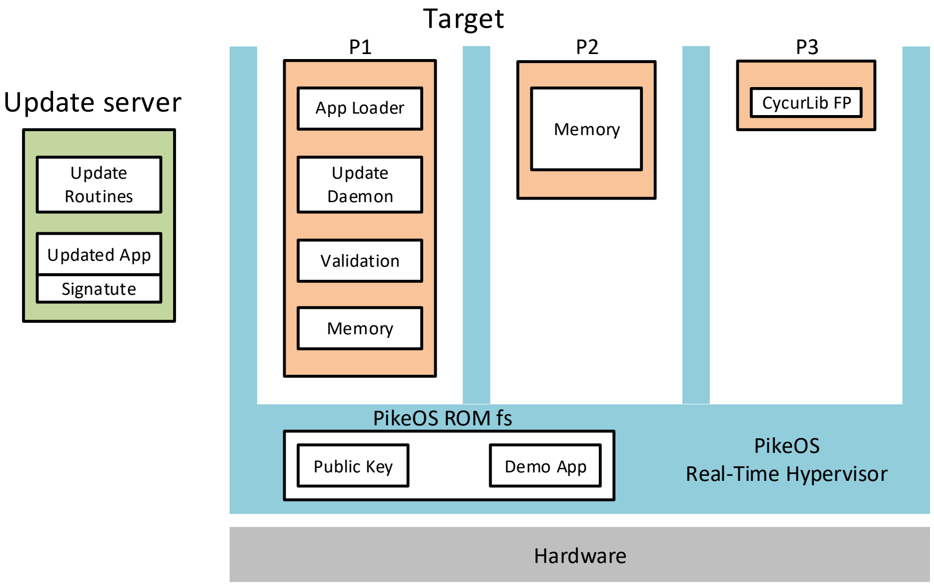
- Server:
  - Setup
    - Generate private key
    - Calculate public key
  - Sign
    - Sign updated app (with private key)
  - Distribution
    - Signed app = Updated app + Signature
- Client:
  - Setup
    - Install server's public key
    - Protect it against manipulation
  - Verify signature
    - Download signed app from server
    - Verify signature (with public key)
    - Install updated app only if signature is correct



Demo setup

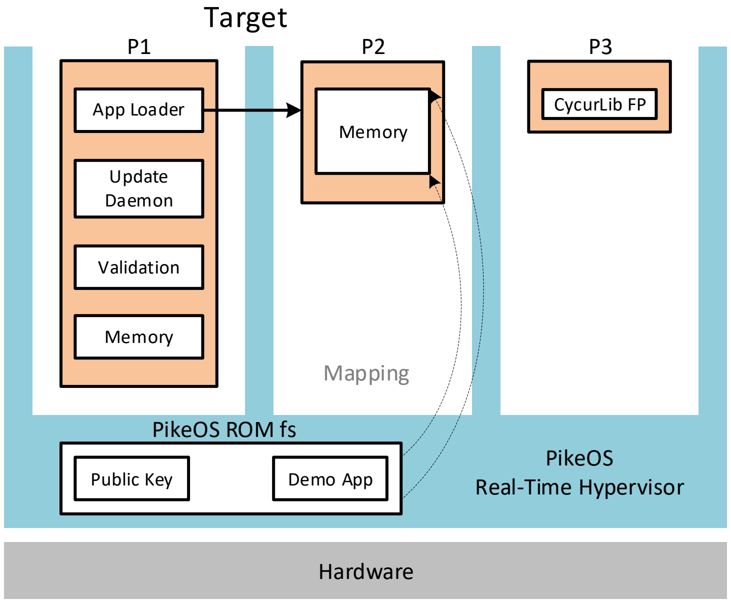
- There are three partitions
  - P1: Application Manager
  - P2: Application for update
  - P3: CycurLIB file provider
    - (Crypto Server)

- The demo is started after secure boot



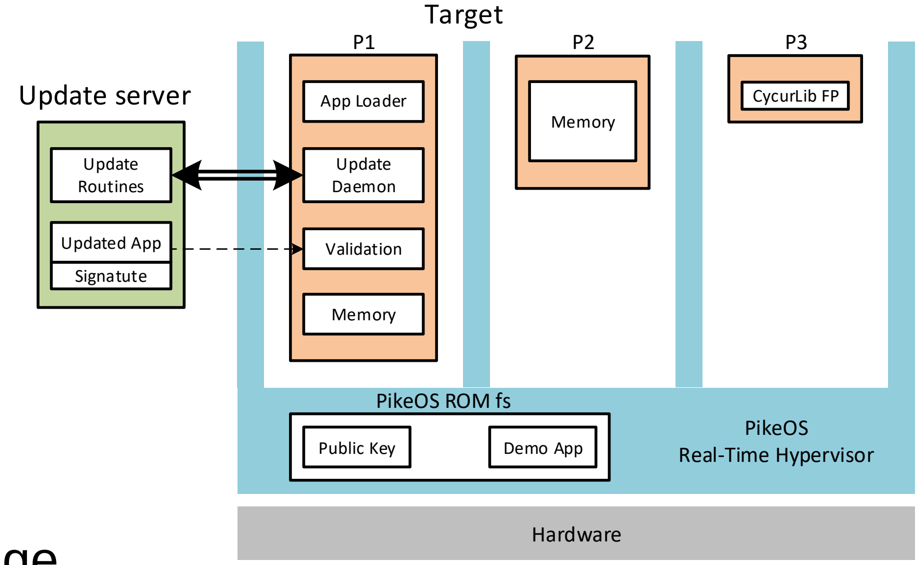
Initial state

- Application Manager (P1)
  - Loads application
    - App is in P2
  - Launches P2 partition
  - Starts update daemon
    - Listening for incoming update orders



Update process

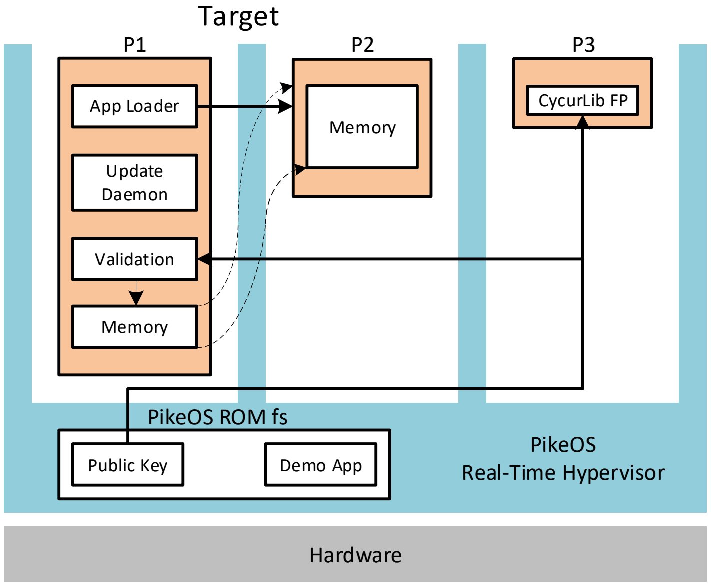
- Server
  - Connects to Application Manager (P1)
  - Sends new App image
  - Image contains:
    - New Application
    - Application Signature



- Update Daemon (P1)
  - Accepts connection
  - Download new App image
  - Locally stores the Updated App

Validation

- Application Manager (P1)
  - Validates the Updated App against its signature
  - CycurLIB file provider (P3) performs security operations
  - If validation successful, Stops P2
  - Updates the P2 memory region with the Updated App
  - Restarts P2

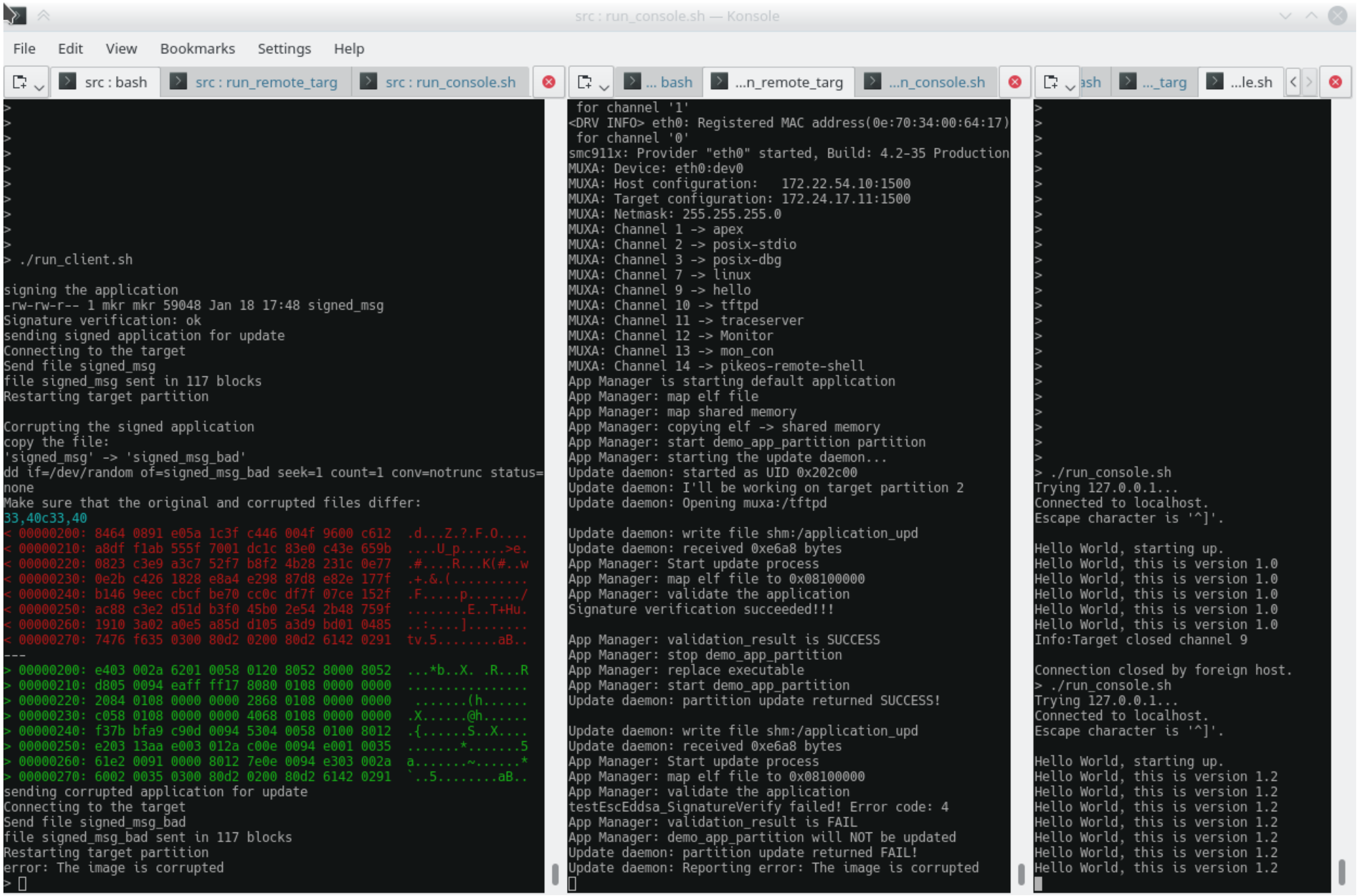


- P3 is running ESCRIPT CycurLIB for security operations
- The public key is stored in a non volatile memory region
  - PKI is part of secure boot
- The public key can be updated during the project integration.

CycurLIB

- Library with cryptographic algorithms
- Tailored towards embedded systems
  - Optimization for execution speed or small code size
- EdDSA: state-of-the-art digital signature algorithm
  - Based on elliptic curves
  - Public/private key pair: 256 bits each
    - Security equivalent to RSA with ~3072 bits
  - Signature: 512 bits
  - No branches or array indexes that depend on secret data → robust against many side-channel attacks

Video Demonstrator



```
>
>
>
>
>
>
>
>
>
>
> ./run_client.sh

signing the application
-rw-rw-r-- 1 mkr mkr 59048 Jan 18 17:48 signed_msg
Signature verification: ok
sending signed application for update
Connecting to the target
Send file signed_msg
file signed_msg sent in 117 blocks
Restarting target partition

Corrupting the signed application
copy the file:
'signed_msg' -> 'signed_msg_bad'
dd if=/dev/random of=signed_msg_bad seek=1 count=1 conv=notrunc status=
none
Make sure that the original and corrupted files differ:
33,40c33,40
< 00000200: 8464 0891 e05a 1c3f c446 004f 9600 c612  .d...Z.?.F.O....
< 00000210: a8df f1ab 555f 7001 dc1c 83e0 c43e 659b  ....U_p......>e.
< 00000220: 0823 c3e9 a3c7 52f7 b8f2 4b28 231c 0e77  .#....R...K(#..w
< 00000230: 0e2b c426 1828 e8a4 e298 87d8 e82e 177f  .+.&.(..........
< 00000240: b146 9eec cbcf be70 cc0c df7f 07ce 152f  .F.....p......./
< 00000250: ac88 c3e2 d51d b3f0 45b0 2e54 2b48 759f  ........E..T+Hu.
< 00000260: 1910 3a02 a0e5 a85d d105 a3d9 bd01 0485  ..:....]........
< 00000270: 7476 f635 0300 80d2 0200 80d2 6142 0291  tv.5........aB..
---
> 00000200: e403 002a 6201 0058 0120 8052 8000 8052  ...*b..X. .R...R
> 00000210: d805 0094 eaff ff17 8080 0108 0000 0000  ................
> 00000220: 2084 0108 0000 0000 2868 0108 0000 0000   .......(h......
> 00000230: c058 0108 0000 0000 4068 0108 0000 0000  .X......@h......
> 00000240: f37b bfa9 c90d 0094 5304 0058 0100 8012  .{......S..X....
> 00000250: e203 13aa e003 012a c00e 0094 e001 0035  .......*.......5
> 00000260: 61e2 0091 0000 8012 7e0e 0094 e303 002a  a.......~......*
> 00000270: 6002 0035 0300 80d2 0200 80d2 6142 0291  `..5........aB..
sending corrupted application for update
Connecting to the target
Send file signed_msg_bad
file signed_msg_bad sent in 117 blocks
Restarting target partition
error: The image is corrupted
> ▯
```

```
 for channel '1'
<DRV INFO> eth0: Registered MAC address(0e:70:34:00:64:17)
 for channel '0'
smc911x: Provider "eth0" started, Build: 4.2-35 Production
MUXA: Device: eth0:dev0
MUXA: Host configuration:   172.22.54.10:1500
MUXA: Target configuration: 172.24.17.11:1500
MUXA: Netmask: 255.255.255.0
MUXA: Channel 1 -> apex
MUXA: Channel 2 -> posix-stdio
MUXA: Channel 3 -> posix-dbg
MUXA: Channel 7 -> linux
MUXA: Channel 9 -> hello
MUXA: Channel 10 -> tftpd
MUXA: Channel 11 -> traceserver
MUXA: Channel 12 -> Monitor
MUXA: Channel 13 -> mon_con
MUXA: Channel 14 -> pikeos-remote-shell
App Manager is starting default application
App Manager: map elf file
App Manager: map shared memory
App Manager: copying elf -> shared memory
App Manager: start demo_app_partition partition
App Manager: starting the update daemon...
Update daemon: started as UID 0x202c00
Update daemon: I'll be working on target partition 2
Update daemon: Opening muxa:/tftpd

Update daemon: write file shm:/application_upd
Update daemon: received 0xe6a8 bytes
App Manager: Start update process
App Manager: map elf file to 0x08100000
App Manager: validate the application
Signature verification succeeded!!!

App Manager: validation_result is SUCCESS
App Manager: stop demo_app_partition
App Manager: replace executable
App Manager: start demo_app_partition
Update daemon: partition update returned SUCCESS!

Update daemon: write file shm:/application_upd
Update daemon: received 0xe6a8 bytes
App Manager: Start update process
App Manager: map elf file to 0x08100000
App Manager: validate the application
testEscEddsa_SignatureVerify failed! Error code: 4
App Manager: validation_result is FAIL
App Manager: demo_app_partition will NOT be updated
Update daemon: partition update returned FAIL!
Update daemon: Reporting error: The image is corrupted
▯
```

```
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
> ./run_console.sh
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello World, starting up.
Hello World, this is version 1.0
Hello World, this is version 1.0
Hello World, this is version 1.0
Hello World, this is version 1.0
Hello World, this is version 1.0
Info:Target closed channel 9

Connection closed by foreign host.
> ./run_console.sh
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello World, starting up.
Hello World, this is version 1.2
Hello World, this is version 1.2
Hello World, this is version 1.2
Hello World, this is version 1.2
Hello World, this is version 1.2
Hello World, this is version 1.2
Hello World, this is version 1.2
▮
```

Takeaways

- Secure Update is mechanism to update securely and safely deployed applications
  - Proposed solution provide independence of applications and platform updates
- Secure Update on top of Real-Time Hypervisors enables split safety- and security-critical applications
  - Reduce attack surface
  - Reduce amount of safety-critical code to certify

Thanks!

SAFURE Grant Agreement No. 644080

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644080."

"This work was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 15.0025. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the Swiss Government."

If you need further information, please contact the coordinator:

TECHNIKON Forschungs- und Planungsgesellschaft mbH
Burgplatz 3a, 9500 Villach, AUSTRIA
Tel: +43 4242 233 55   Fax: +43 4242 233 55 77
E-Mail: coordination@safure.eu

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.