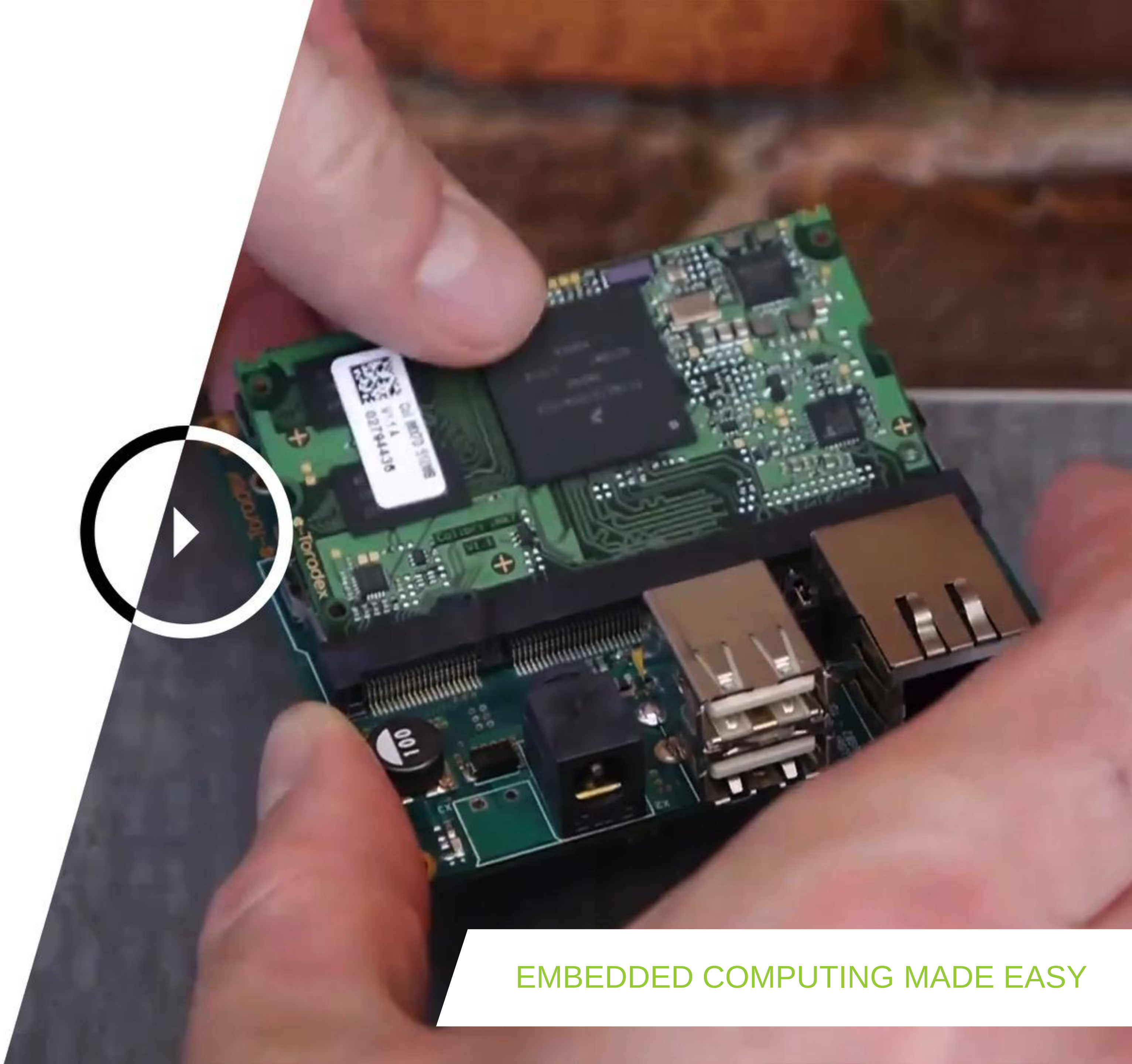
Toradex

Swiss. Embedded. Computing.

Colibri iMX6ULL and A71CH Hands-on



EMBEDDED COMPUTING MADE EASY

WHAT WE’LL COVER TODAY

Setup Overview

Elliptic Curve Cryptography

OpenSSL

Use case of this Hands-on

Hands-on

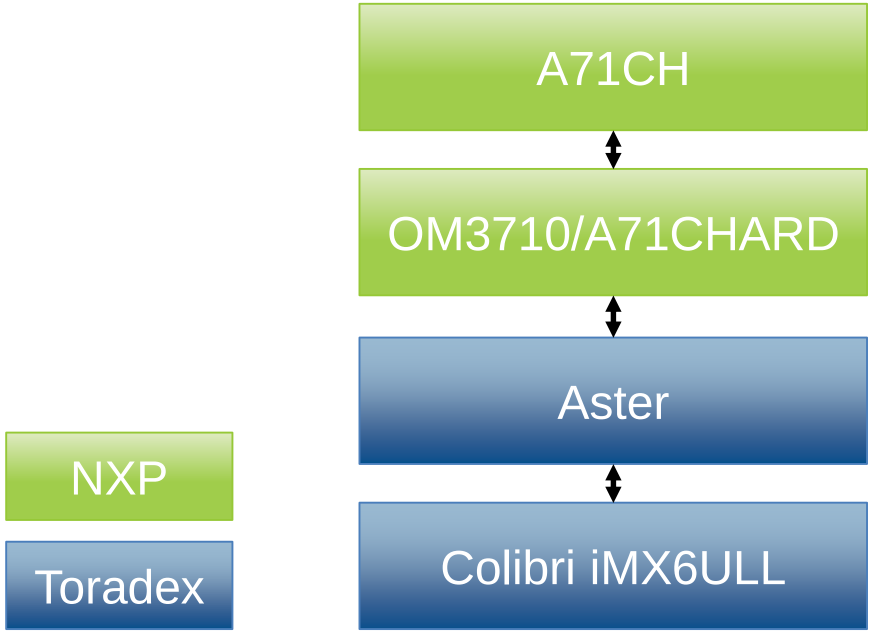
Further Use cases

Toradex
Swiss. Embedded. Computing.

SETUP OVERVIEW

Swiss. Embedded. Computing.

SETUP OVERVIEW
HARDWARE



COLIBRI iMX6ULL
NXP i.MX 6ULL

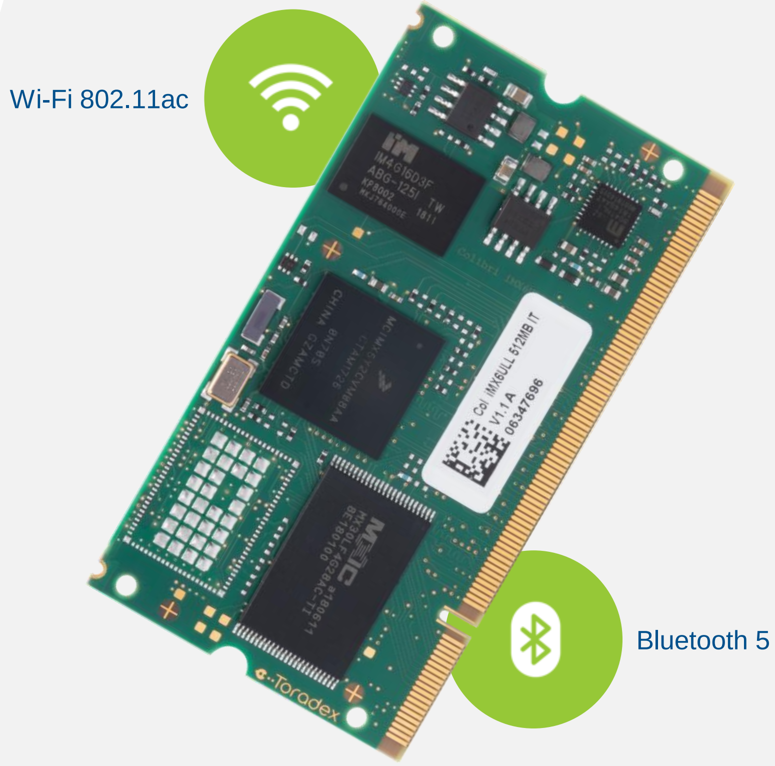
Wi-Fi and **Bluetooth 5**

Ideal for **Industrial IoT** and **Embedded Applications**

Integrated Security Features

Long-term Availability until 2028

Large **Partner Ecosystem**



A7 @900 MHz

Torizon

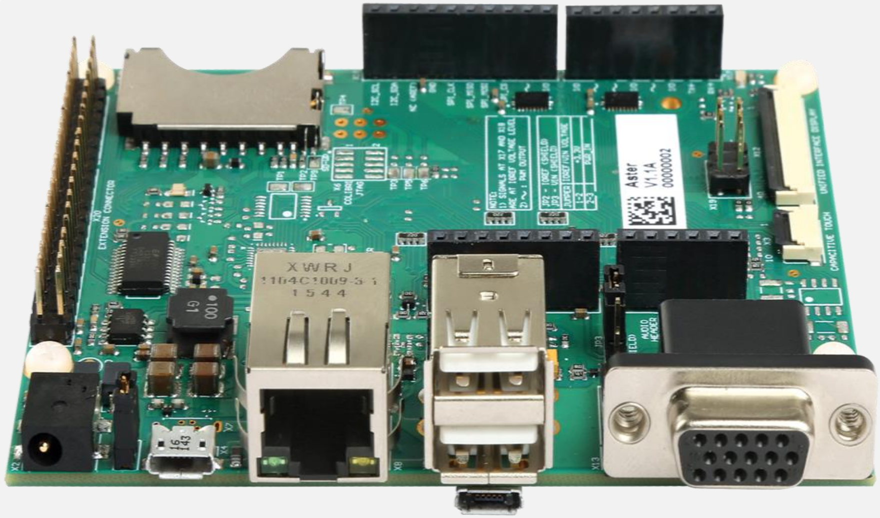
Toradex
Swiss. Embedded. Computing.

PRE-INSTALLED TORADEX EASY INSTALLER

www.toradex.com/computer-on-modules/colibri-arm-family/nxp-imx6ull

ASTER
CARRIER BOARD

- USB 2.0: 2x Host, 1x Client (Shared)
- 10/100 Mbit Ethernet
- Size: 100x80 mm

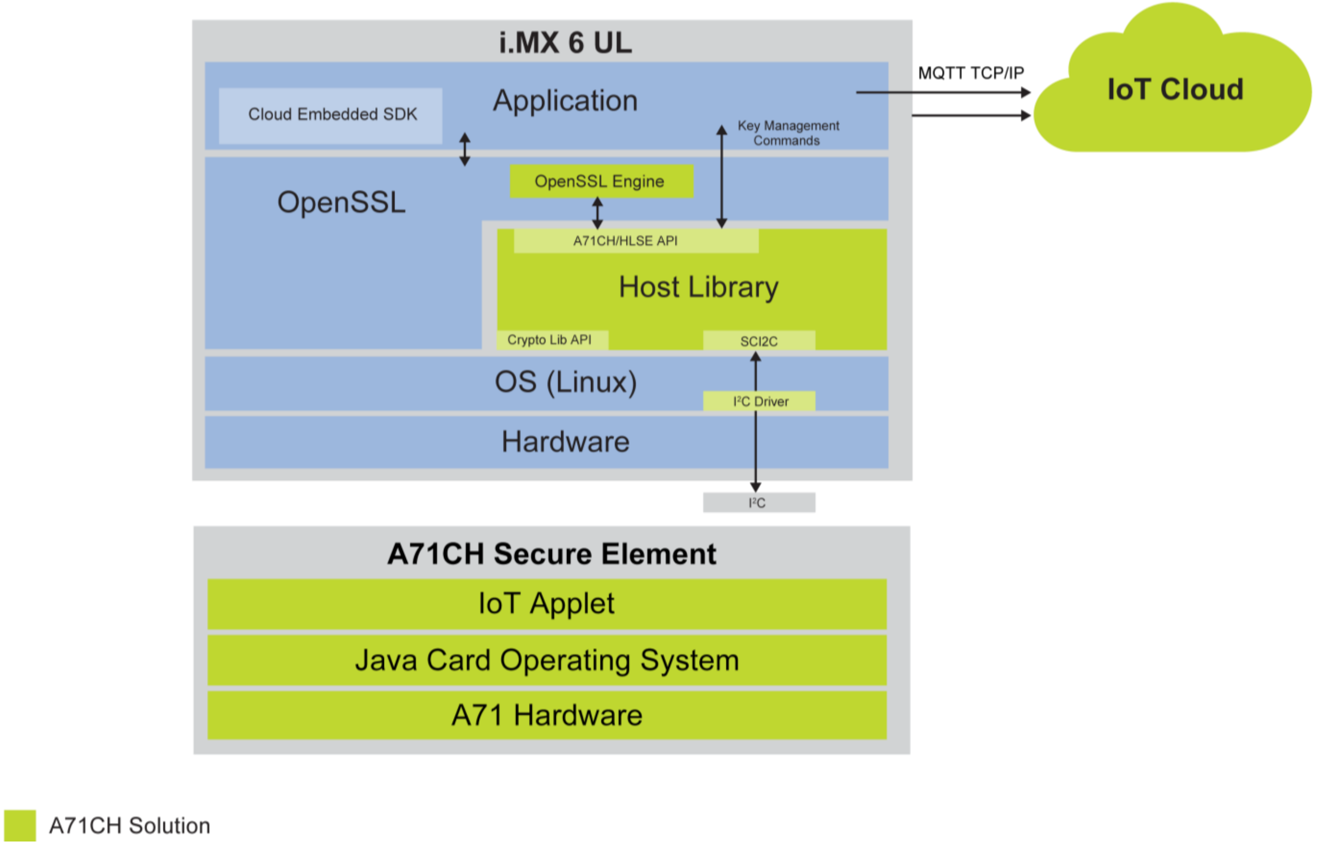


Toradex
Swiss. Embedded. Computing.

SETUP OVERVIEW
SOFTWARE

Application : Custom
OS : Torizon
Hardware : Colibri iMX6ULL

Direct access to OpenSSL
Direct access to Host Library
Access via OpenSSL Engine



https://www.nxp.com/assets/images/en/block-diagrams/A71CH-Block-Diagram2.png

ELLIPTIC CURVE CRYPTOGRAPHY

Toradex
Swiss. Embedded. Computing.

ELLIPTIC CURVE CRYPTOGRAPHY

1976 DHKE	1977 RSA	1985 Elgamal	1985 ECC	1991 DSA	1994 DSS	1998 OpenSSL	~1999 ECDSA
Diffie-Hellman Key Exchange	First asymmetric crypto system	Another asymmetric crypto system	Elliptic-curve cryptography	Digital Signature Algorithm	Digital Signature Standard		

Elliptic Curve Digital Signature Algorithm

1998 ISO standard
1999 ANSI standard
2000 IEEE & NIST standard

Toradex
Swiss. Embedded. Computing.

ELLIPTIC CURVE CRYPTOGRAPHY
SOME TERMS

Mathematatic Group

In mathematics, a group is a set equipped with a binary operation which combines any two elements to form a third element in such a way that four conditions called group axioms are satisfied, namely closure, associativity, identity and invertibility. (Wikipedia)

Finite Field (Galois field)

As with any field, a finite field is a set on which the operations of multiplication, addition, subtraction and division are defined and satisfy certain basic rules. The most common examples of finite fields are given by the integers mod p when p is a prime number. (Wikipedia)

Discrete Logarithm Problem

$$(m^e \; mod \; n)^d \; mod \; n = m \mid \forall \; 1 \leq m \leq M$$

Toradex
Swiss. Embedded. Computing.

ELLIPTIC CURVE CRYPTOGRAPHY
DIGITAL SIGNATURE ALGORITHM

Key Generation

1. Decide on a key length L and N (possible predefined values in the standard)
2. Choose an N-bit prime **q**
3. Choose an L-bit prime **p** such that p − 1 is a multiple of q.
4. Choose **g**, a number whose multiplicative order modulo p is q. This means that q is the smallest positive integer such that $g^q = 1 \; mod \; p$
5. Select a random integer **x** such that $1 \le x \le q - 1$
6. Compute $\boldsymbol{y} = g^x \; mod \; p$
7. Public key: (p, q, g, y), secrect key: x

ELLIPTIC CURVE CRYPTOGRAPHY
DIGITAL SIGNATURE ALGORITHM

Signing

1. Generate a **random per-message** value k where $1 < k < q$
2. Compute $r = \left(g^k \; mod \; p\right) \; mod \; q$
3. Compute $s = k^{-1} \; (h(m) + xr) \; mod \; q$
4. If s or r are 0, restart with 1
5. Your signature is (r,s)

h(m): public, collision-free hash function

Toradex
Swiss. Embedded. Computing.

ELLIPTIC CURVE CRYPTOGRAPHY
DIGITAL SIGNATURE ALGORITHM

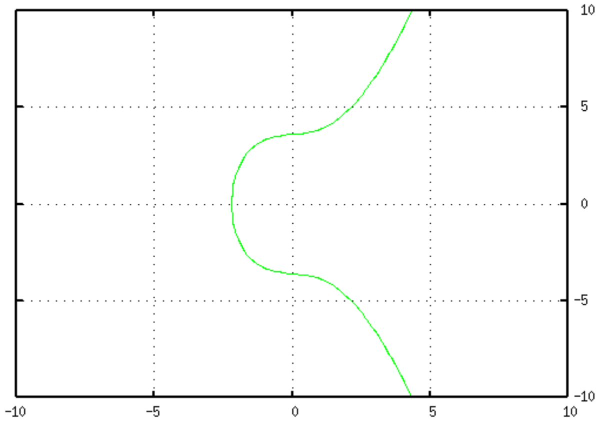
Verifying

1. Reject the signature if $0 < r < q$ or $0 < s < q$ is not satisfied
2. Compute $w = s^{-1} \, mod \, q$
3. Compute $u1 = h(m) * w \, mod \, q$
4. Compute $u2 = r \, * w \, mod \, q$
5. Compute $v = (g^{u1} \, y^{u2} \, mod \, p) \, mod \, q$
6. The signature is valid if $v = r$

Toradex
Swiss. Embedded. Computing.

ELLIPTIC CURVE CRYPTOGRAPHY
ECDSA

- The ellipctic curve is a plane curve over a finite field
- The operations of the group are well defined
- Calculation with points on the elliptic curve
- Special variant of DSA
- Use of named curves (predefined, standard curves)
- Elliptic curve discrete logarithm problem would need $\mathcal{O}(\sqrt{n})$
- With 128-bit Security:
  - ECDSA: 256-bit public keys and 512-bit signature
  - RSA: 3072-bit public keys and 3072-bit signature



Elliptic curve over $\mathbb{R}$
$E: y^2 = x^3 + x + 13$

https://de.wikipedia.org/wiki/Elliptic_Curve_Cryptography#/media/File:Elliptic_curve_crypto.png

ELLIPTIC CURVE CRYPTOGRAPHY
ECDSA

Format of the signature

The ECDSA standards (ANSI X9.62, FIPS 186-4) don't define an ECDSA signature as a sequence of bytes, but as a **pair of values (r,s)***. In practice, two main encodings for ECDSA signatures

- ASN.1 DER
- Raw format of concatenated r and s

0x30|*b1*|0x02|*b2*|**r**|0x02|*b3*|**s**
b1 = Length of remaining data
b2 = Length of r
b3 = Length of s

**Encoding of signatures is considered to be out of scope; the protocol that uses ECDSA signatures is responsible for defining which encoding will be used*

OPENSSL

OPENSSL
GENERAL

- Implementation of TLS
- Current version 1.1.1b (26. February 2019)*
- BSD License
- Implementation of basic cryptographic functions and various utitlity functions
- Written in C, but available for different languages with available wrappers

*retrieved 07.05.2019

OPENSSL
X509

- Standard defining the format of public key certificates
- Contains a public key and an identity
- Either signed by a certificate authority or self-signed
- Expressed in ASN.1

CN: CommonName
OU: OrganizationalUnit
O: Organization
L: Locality
S: StateOrProvinceName
C: CountryName

/C=CH/S=Lucerne/L=Horw/O=Toradex/OU=Demo-Unit/CN=demoCA/emailAddress=demoCA@toradex

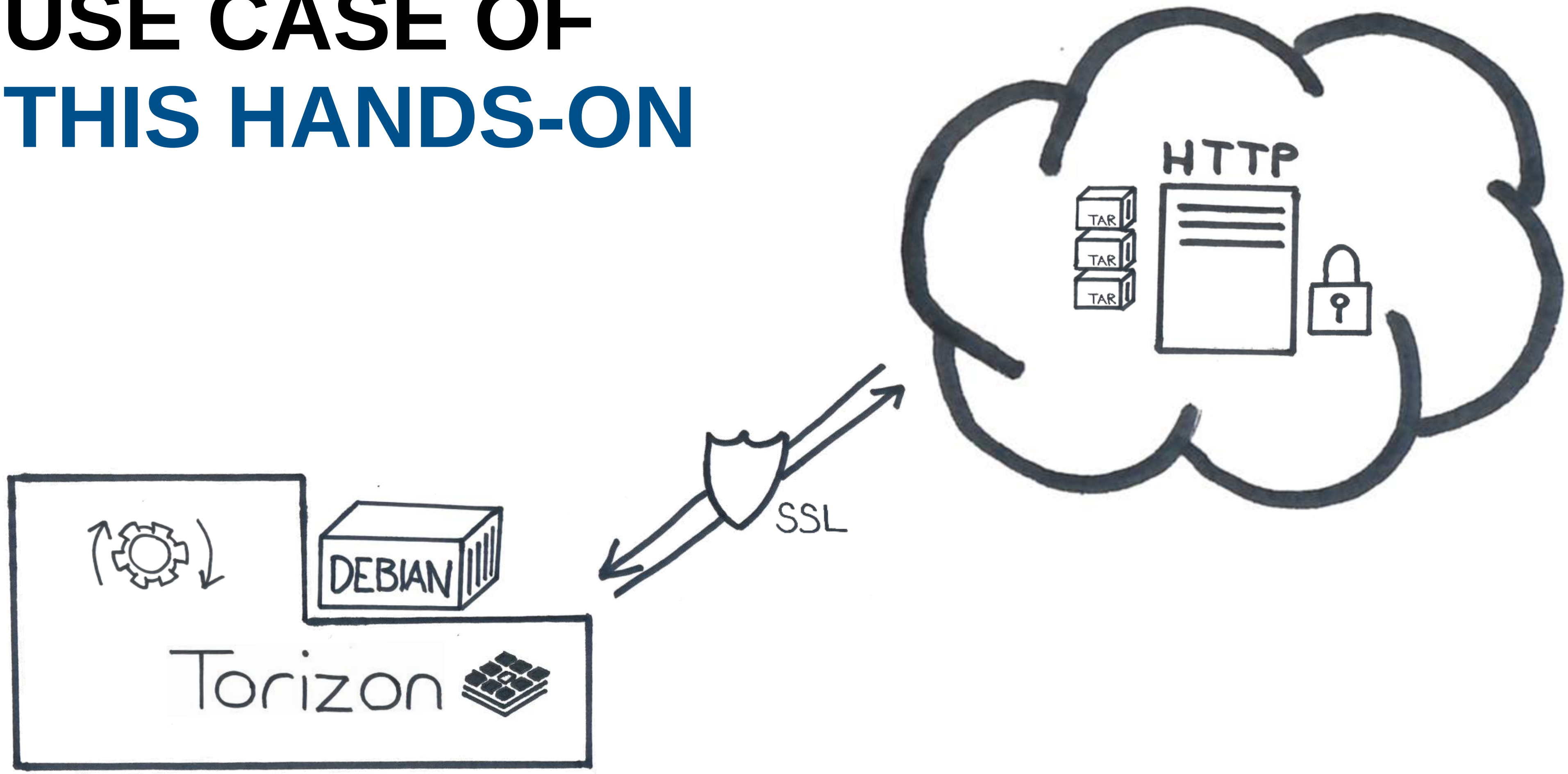
OPENSSL
ENGINE

- Possibility to create a custom OpenSSL engine
  - Connect HW accelerator
- Custom implementation of cryptographic algorithms
- Dynamic loading of the required OpenSSL engine

USE CASE OF THIS HANDS-ON

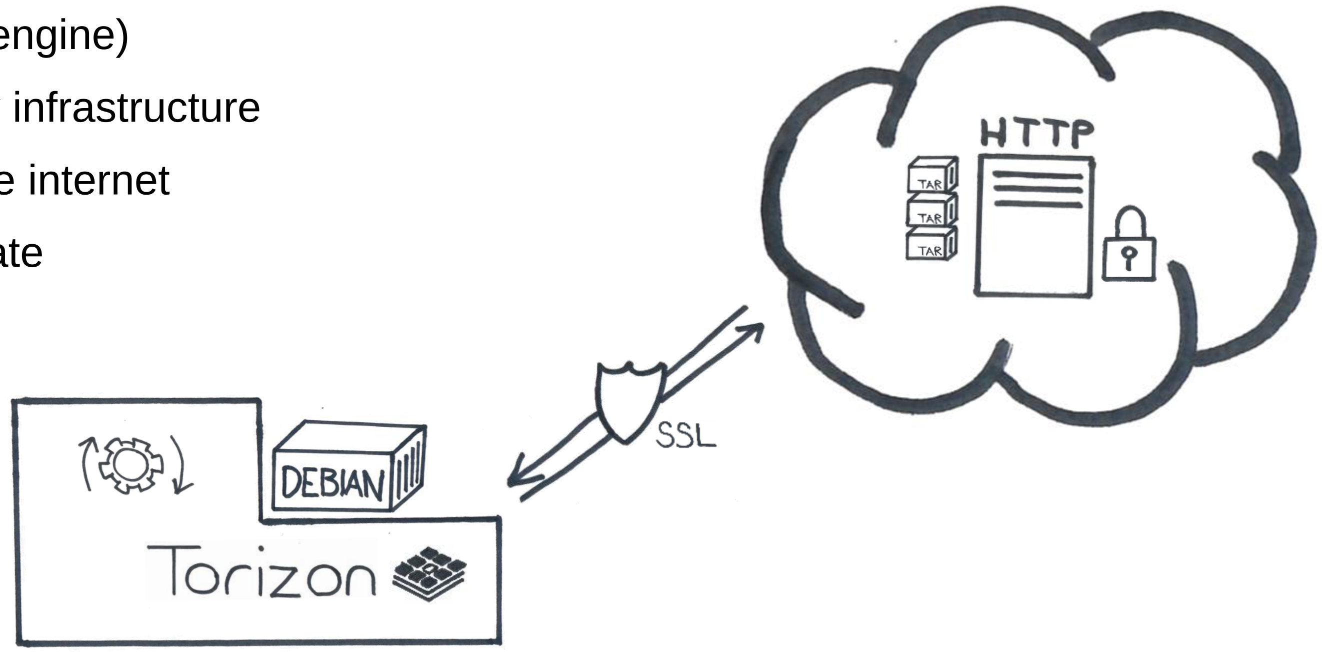
Toradex
Swiss. Embedded. Computing.

USE CASE OF THIS HANDS-ON



USE CASE OF
THIS HANDS-ON

- Secure update of containers
- Use of OpenSSL (A71CH engine)
- Running without public key infrastructure
- Update in the field, over the internet
- Automated or manual update

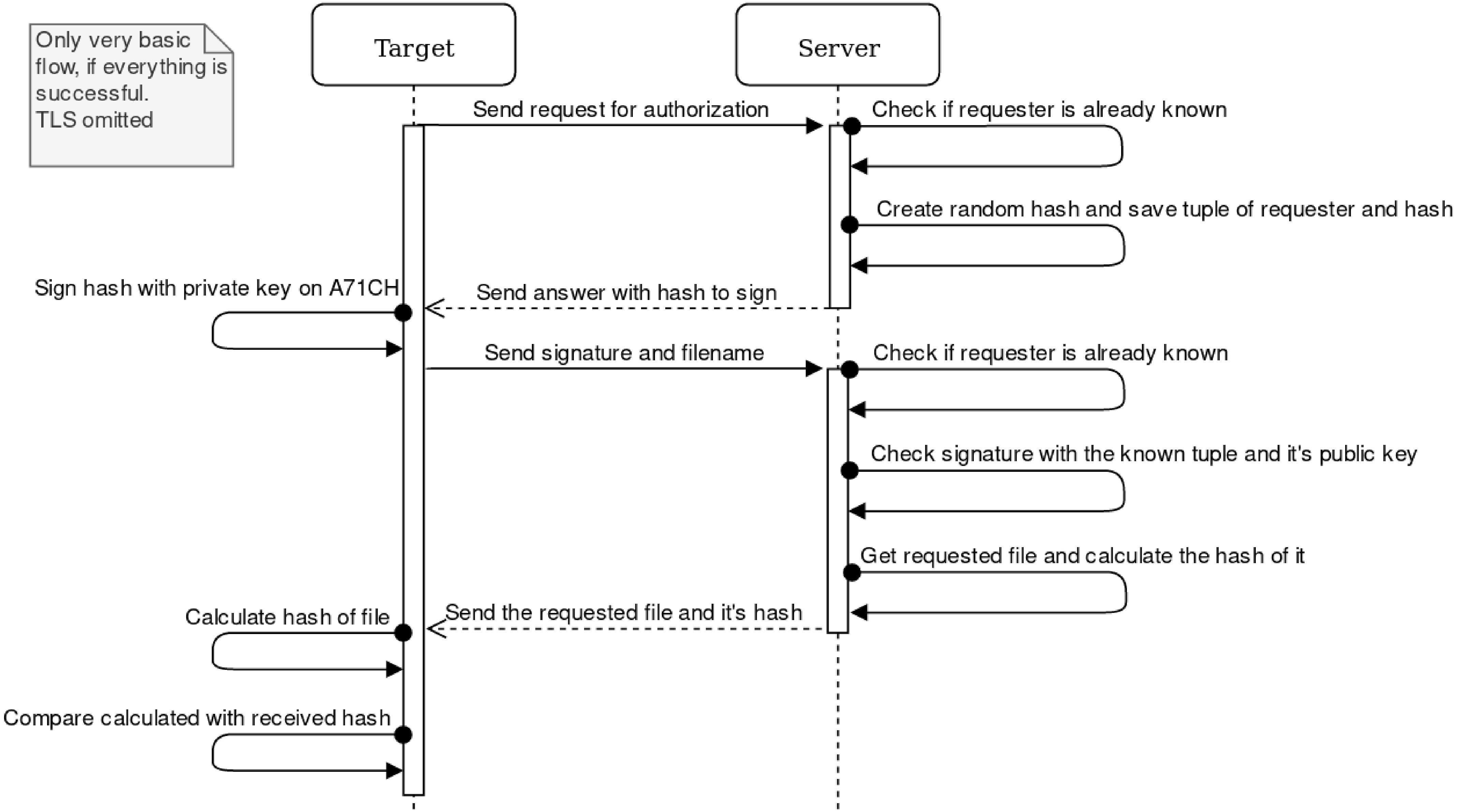


HANDS-ON

HANDS-ON!

- Demonstration with successful update
- Demonstration of security features
- Code walk-through on the server part
- Code walk-through on the client part

The code of this hands-on is specifically written for training purposes and not directly applicable for end-use.


Only very basic flow, if everything is successful.
TLS omitted
Target
Server
Send request for authorization
Check if requester is already known
Create random hash and save tuple of requester and hash
Send answer with hash to sign
Sign hash with private key on A71CH
Send signature and filename
Check if requester is already known
Check signature with the known tuple and it's public key
Get requested file and calculate the hash of it
Send the requested file and it's hash
Calculate hash of file
Compare calculated with received hash

FURTHER USE CASES

FURTHER USE CASES

- Secure communication between IoT devices and the cloud
- Identity proof of devices
- Use of credentials for different purposes
- Direct methods of the A71CH
  - Calcucation of hash values
  - Sign and verification of hashs
  - Secure memory

Toradex
Swiss. Embedded. Computing.

THANK YOU FOR YOUR INTEREST.

www.**toradex**.com | **developer**.toradex.com | **community**.toradex.com | **labs**.toradex.com