XFS4IoT SP-Dev Workgroup

1 March 2022

XFS4IoT SP-Dev Workgroup Agenda

- Recap from previous meeting
- What is the CEN XFS Committee and how to join
- E2E overview and focus on TR31/TR34
- Barcode Scanner release
- Biometrics release
- What’s next

XFS4IoT Specification Update

- CEN XFS4IoT specification is finalized
- 2021-1 release submitted to CEN
- Already publicly available on GitHub
- Next release target mid 2022 by CEN Committee *(faster pace release than XFS3)*

XFS4IoT Workgroup Progress

- Frameworks are now completed to support a complete Cash Out ATM.
- All now available with C# and C++ sample code released and demos on YouTube.

☑ Card Reader (released May 2021) with support for dispensing (Nov-21)

☑ Cash Dispenser (Jul-21), without end-to-end security

☑ Text Terminal Unit (Jul-21)

☑ EPP Key Management and Crypto classes (Sep-21)

☑ Keyboard and PinPad classes (Oct-21)

☑ End-to-end security partially complete (Oct-21); added required functions (Nov-21)

☑ Printer / Guide lights (Nov-21)

☑ Vendor Mode and Vendor Application (Jan-22)

☑ Auxiliaries (Jan-22)

The XFS4IoT spec is ready

- The XFS4IoT specification has been published
- The Workgroup’s framework code has been updated and is available to support the published version of the XFS4IoT spec
- It is time to implement production XFS4 SPs

ATM Software

The CEN XFS Committee
and
the KAL XF4IoT SP-Dev workgroup

CEN XFS Committee

- CEN XFS Committee is composed of various members including KAL, NCR and Diebold Nixdorf
- New members are welcome
- AFNOR is the committee secretary and can help new members through the process of joining
- CEN XFS Committee monthly online meetings and regular online workshop

KAL XFS4IoT SP-Dev Workgroup

- Created and led by KAL since March 2021
- Building an open-source and free framework for XFS4IoT SPs
- Free to join and open to anyone
- This workgroup doesn’t make decisions on the standard specification
- However, new ideas and proposals can be brainstormed to put forward to the CEN XFS Committee

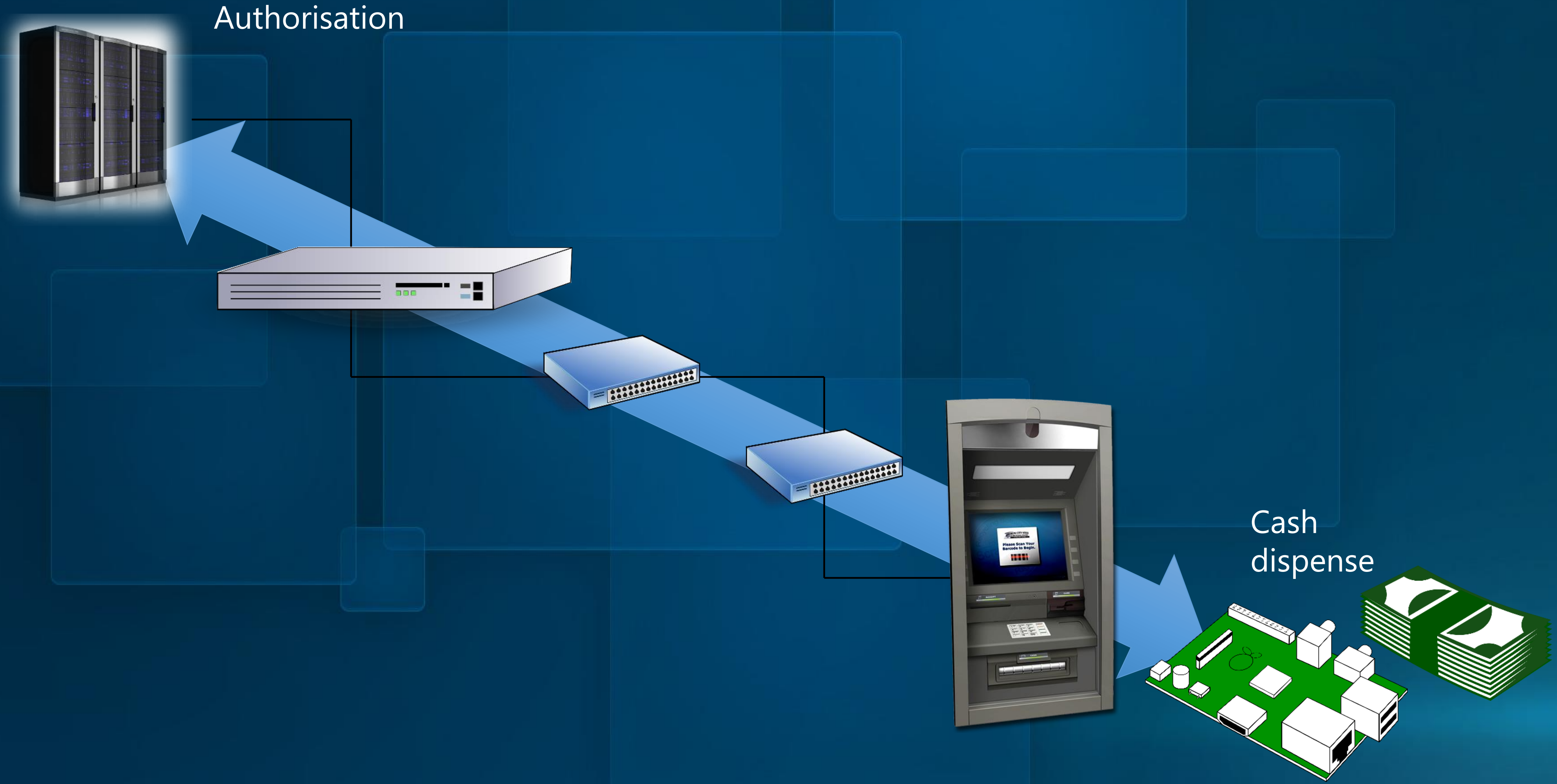
ATM Software

E2E security and key loading

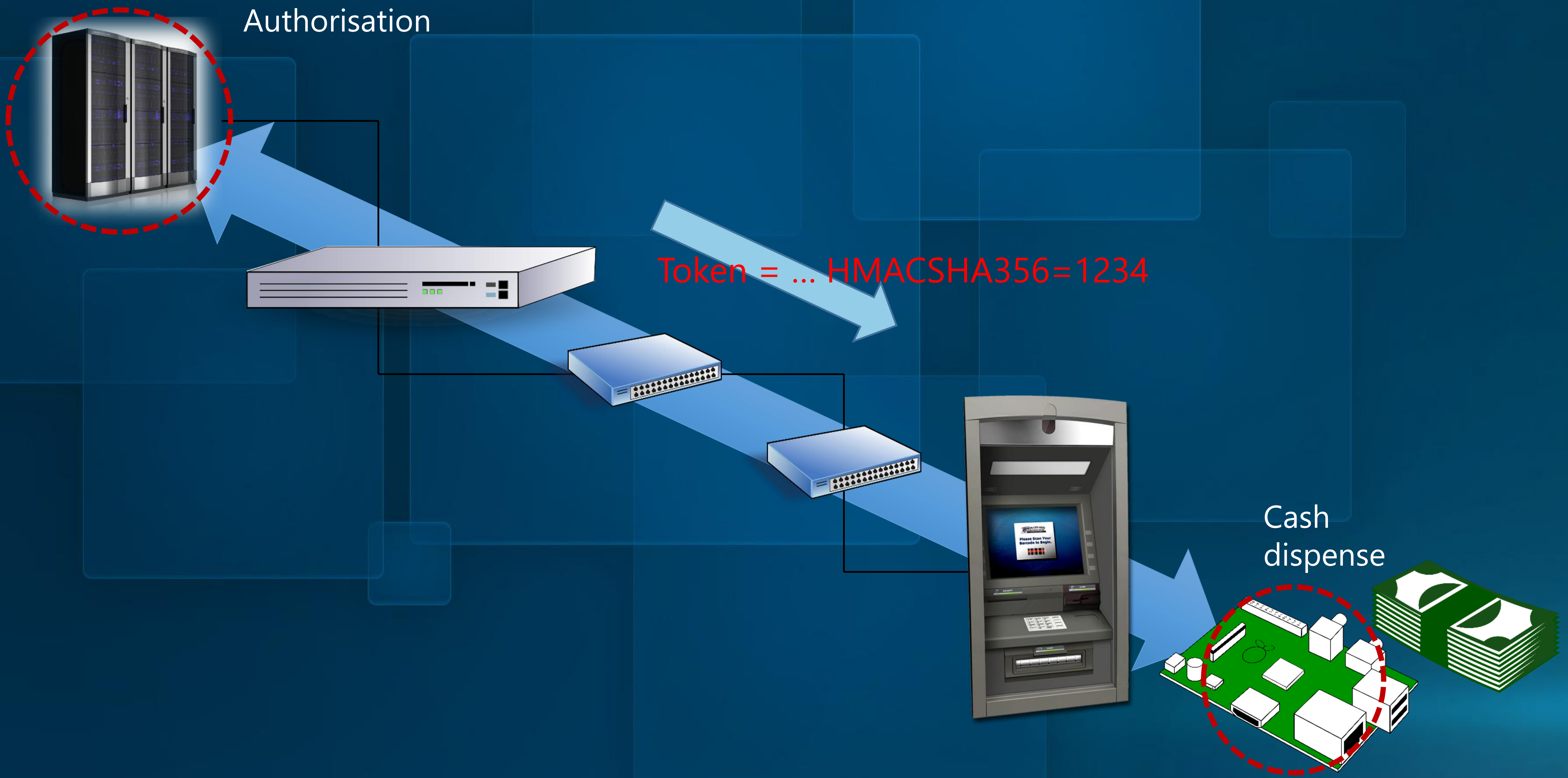
Introduction to TR34 for End-to-End security

Kit Patterson

E2E security



E2E security

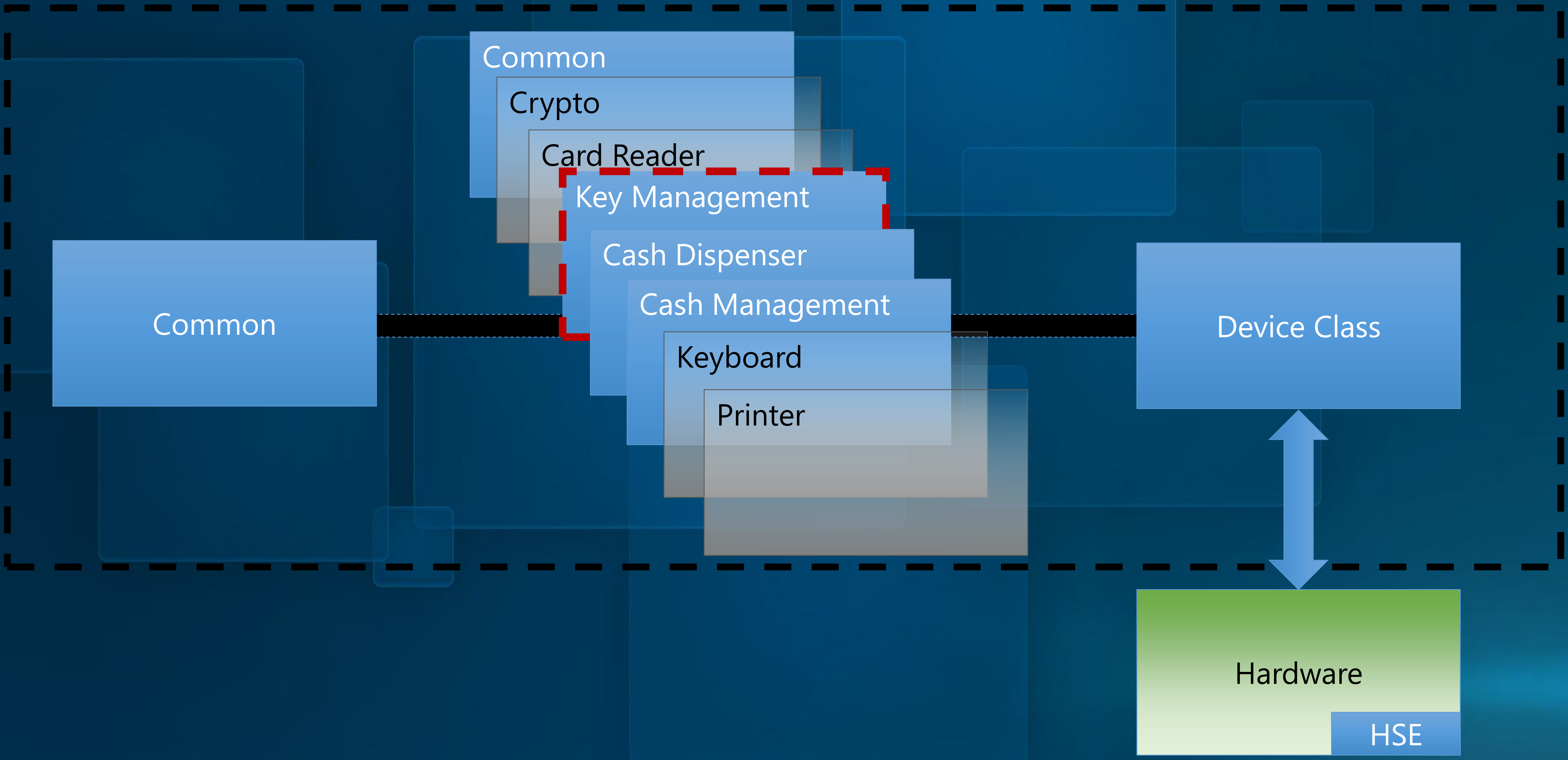


Token

```
NONCE=254611E63B2531576314E86527338D61,TOKENFORMAT=1,TOKENLENGTH=0164,D
ISPENSE1=50.00EUR,
HMACSHA256=CB735612FD6141213C2827FB5A6A4F4846D7A7347B15434916FEA6AC16F3
D2F2
```

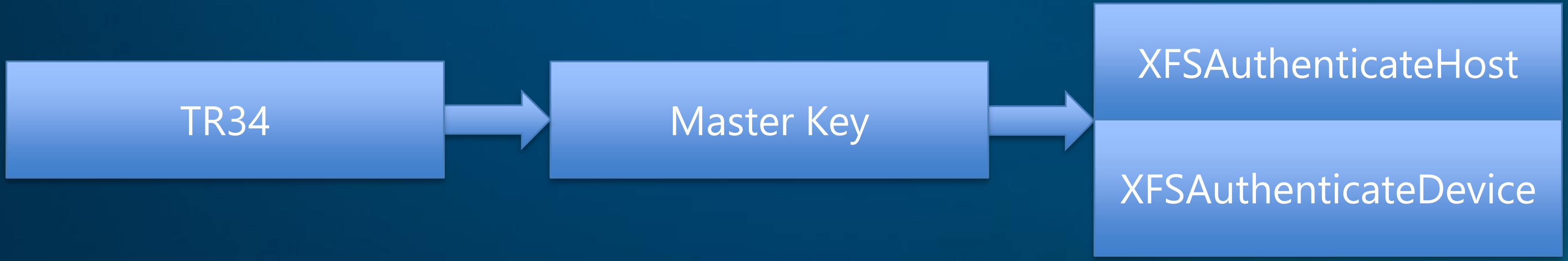
- Token contains data *and* an ‘HMAC’
- HMAC requires a secret key, to create and to check
- ‘Secret’ key needs to be securely shared ahead of time

Framework with End-to-End security



Key loading

- Use cryptographic API to load keys
- Keys must be named “XFSAuthenticateHost” (incoming tokens,) and “XFSAuthenticateDevice” (outgoing tokens)
- Tree structure doesn’t matter, as long as working keys are correct
- Master key should be loaded with “TR34” remote key loading



End-to-end security requires keys

ANSI TR34 / X9.143 for secure remote key loading

TR34 / X9.143

- Previously known as ‘TR34’ now X9.143
- New standard for remote key loading using public key cryptography

- Supports RSA 2048 public/private keys
- Supports loading keys:
  - — 3DES 112-168 bit
  - — AES 128 bit
  - — HMAC
  - — ...

Two types of Cryptography

Symmetric encryption

- Single key used to encrypt/decrypt. Must be kept secret

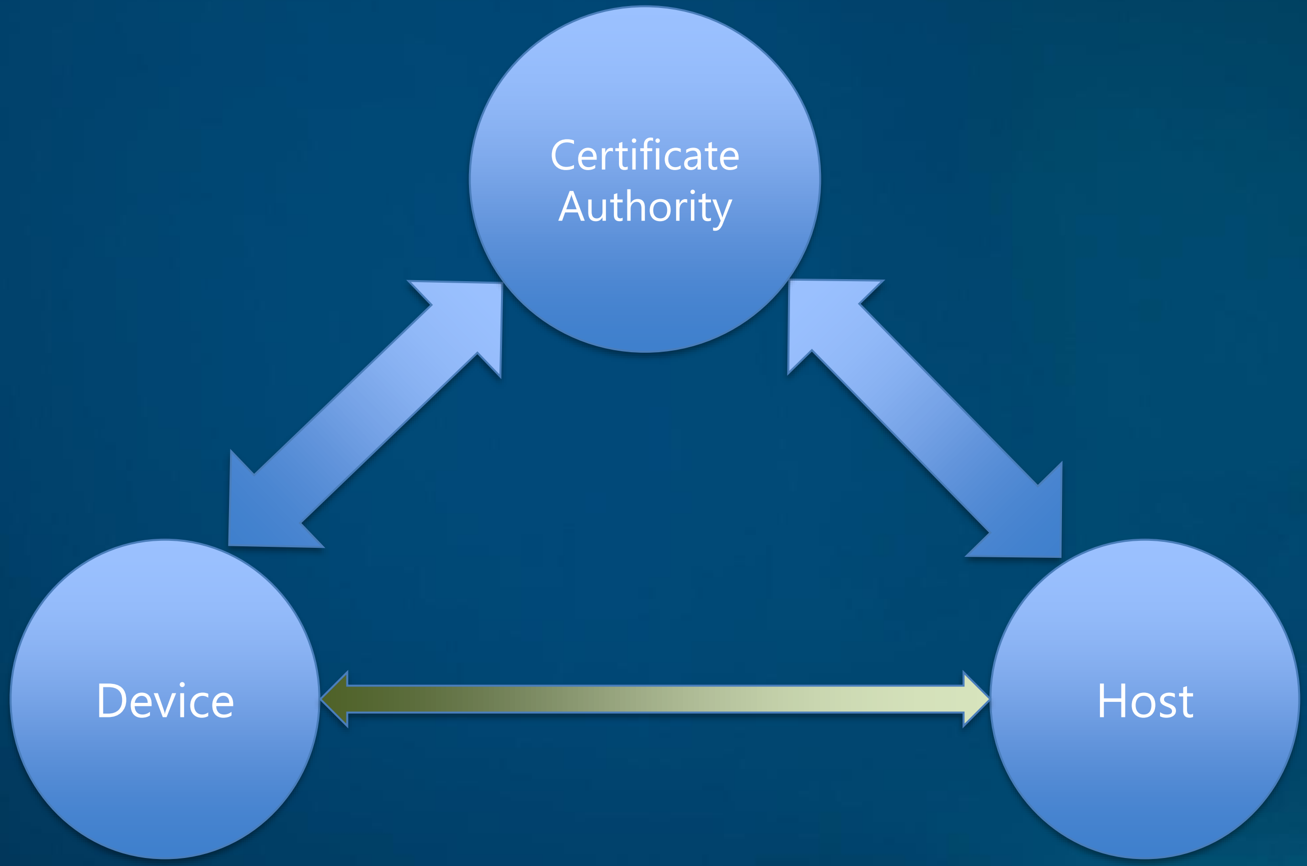
Public/private key pair

- Encryption
  - Public key used to *encrypt* data
  - Private key used to *decrypt* data
- Also ‘signing’
  - Private key used to *create* a ‘digital signature’
  - Public key used to *check* a ‘digital signature’

Certificate authority

- Public key can be easily shared - but we need to confirm that we get the ‘right’ public key
- We need a trusted third party to ‘sign’ each public key so that we can trust it
- A combination of bits of data with a signature is called a ‘certificate’. Usually stored in X.509 format
- So, we create a certificate with the public key, signed by a trusted ‘certificate authority’

Certificate authority



Key exchange

1. Device sends its certificate (public key) to the host. Host checks the CA signature to know that it trusts the public key.

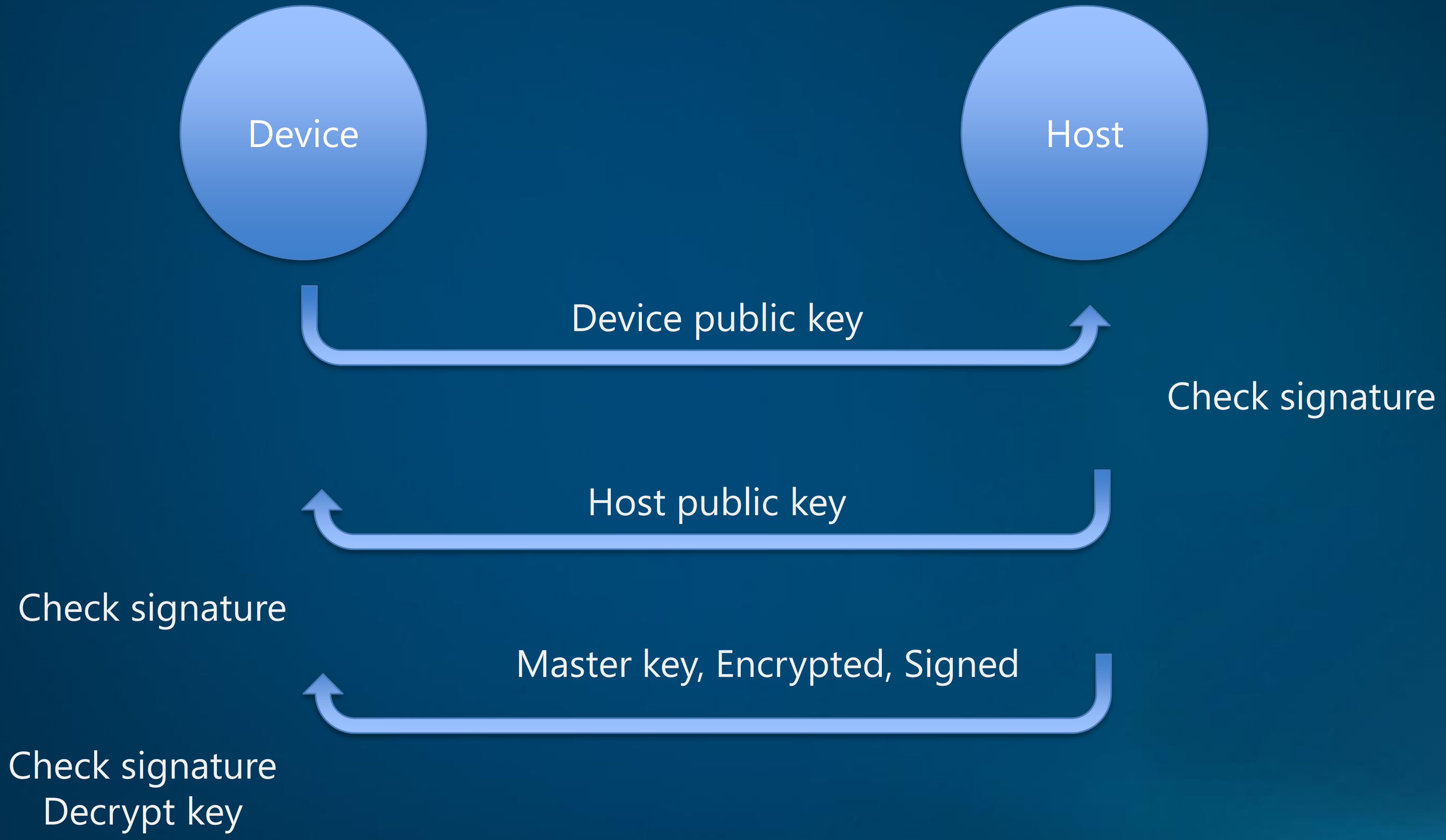
Same in other direction

2. Host sends its certificate (public key) to the device. Device checks CA signature.

Both ends now have both public keys.

3. Host takes the secret master key and
   - Encrypts with the device public key
   - Signs with the host private key
4. Device checks the host signature with the host public key and decrypts the master key with the device private key.

Certificate authority



Device

Host

Device public key

Check signature

Host public key

Check signature

Master key, Encrypted, Signed

Check signature
Decrypt key

Details

- Device should send an ID to the host, so the host can check it's a real device and belongs to the bank
- Device should send a random number. The host includes this in the response so the device knows that the response is 'new'
- The host should check a 'certificate revocation list' – but this might not be possible if offline
- The master key is also encrypted with an 'ephemeral' key, so that key meta-date can also be encrypted

Binding

- TR34 goes further than just using a CA, by enforcing ‘binding’
- Binding means that the first certificate that is used ‘sticks’ – it’s the only certificate that can be used in future
- This means that even a different *valid* certificate is blocked
- Devices can be ‘unbound’ or ‘rebound’ explicitly

SP-Dev framework

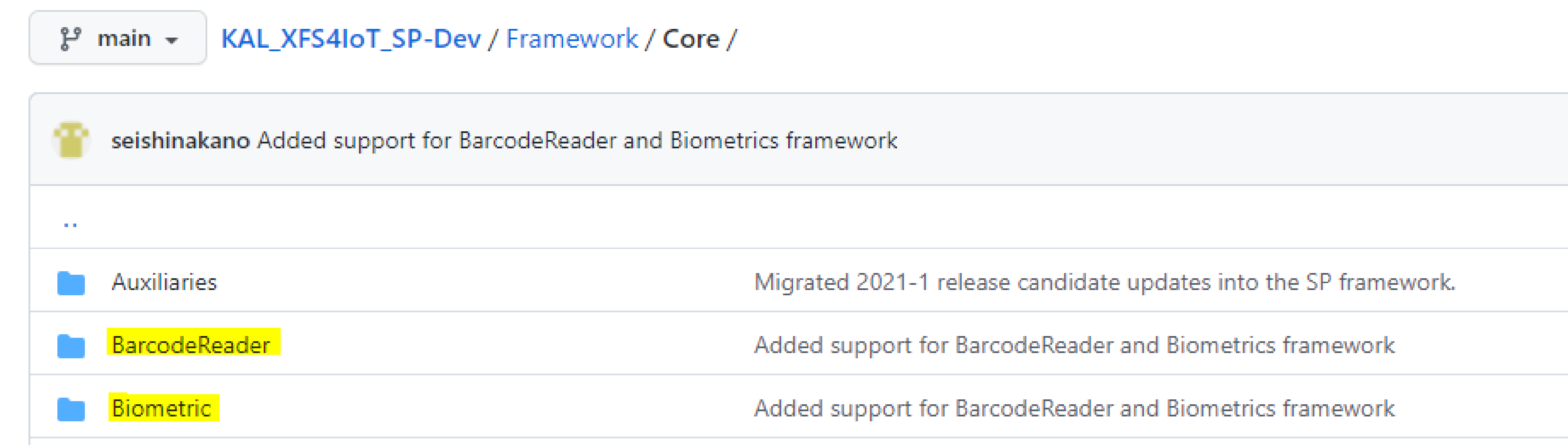
- Framework currently includes KeyManagement class, to support XFS4IoT API
- Firmware or HSE must implement TR34, certificate handling – this is normally already true for EPP hardware
- E2E security is supported by the framework, including firmware code
- KAL are experimenting with creating more code to support TR34 in the firmware, possibly using ‘TPM’ chips

ATM Software

Barcode reader and Biometric classes release

BarcodeReader and Biometric support

- Both classes now available:



main

KAL_XFS4IoT_SP-Dev / Framework / Core /

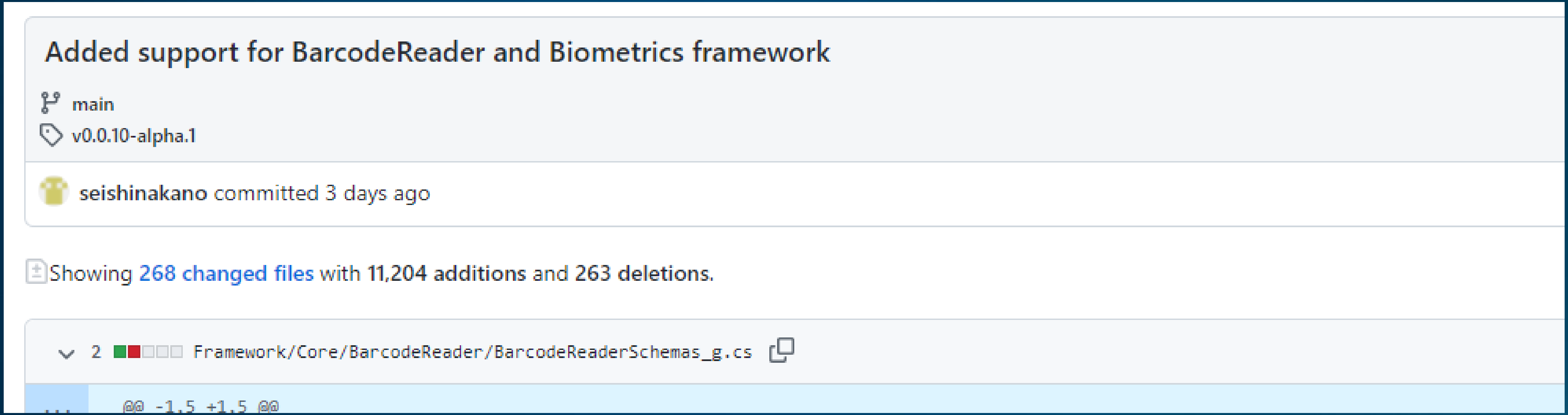
seishinakano Added support for BarcodeReader and Biometrics framework

..

Auxiliaries	Migrated 2021-1 release candidate updates into the SP framework.
BarcodeReader	Added support for BarcodeReader and Biometrics framework
Biometric	Added support for BarcodeReader and Biometrics framework

BarcodeReader and Biometric support

- Details of the changes in the following commit:



Added support for BarcodeReader and Biometrics framework

main

v0.0.10-alpha.1

seishinakano committed 3 days ago

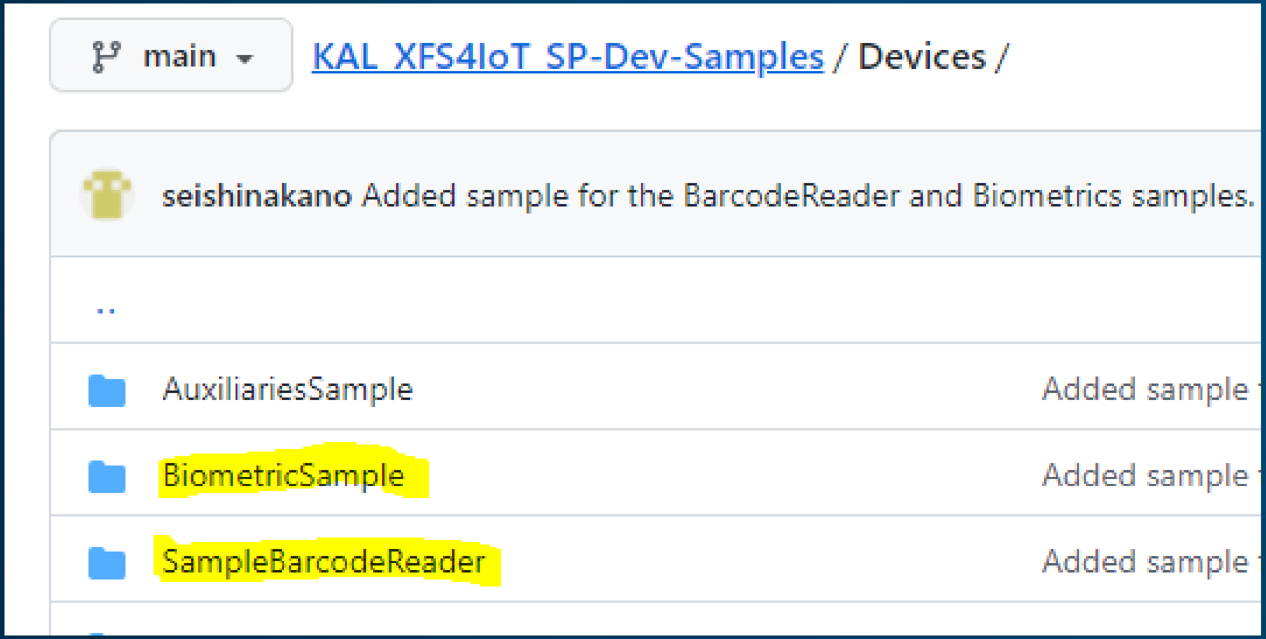
Showing 268 changed files with 11,204 additions and 263 deletions.

```
2 🟩🟥□□□ Framework/Core/BarcodeReader/BarcodeReaderSchemas_g.cs
```

```
@@ -1,5 +1,5 @@
```

BarcodeReader and Biometric support

- Sample available
- Specific commit for all sample changes:



main

KAL_XFS4IoT_SP-Dev-Samples / Devices /

seishinakano Added sample for the BarcodeReader and Biometrics samples.

..

AuxiliariesSample	Added sample
BiometricSample	Added sample
SampleBarcodeReader	Added sample

Functions supported

- BarcodeReader:
  - — Supports one main function – *Read*
  - — Capabilities command will return barcodes supported such as CODE39, CODE128 or even QRCode

ABCabc123
ABCabc123

https://github.com/KAL-ATM-Software/KAL_XFS4IoT_SP-Dev

Functions supported

- Biometric class supports all methods required to:
  - Import
  - Read
  - Match
  - and Clear biometric data
- All biometric data types supported (fingerprint, iris/facial recognition…)

ATM Software

Demos with real devices

Demo with real devices

All demo videos with real devices will be available on YouTube.

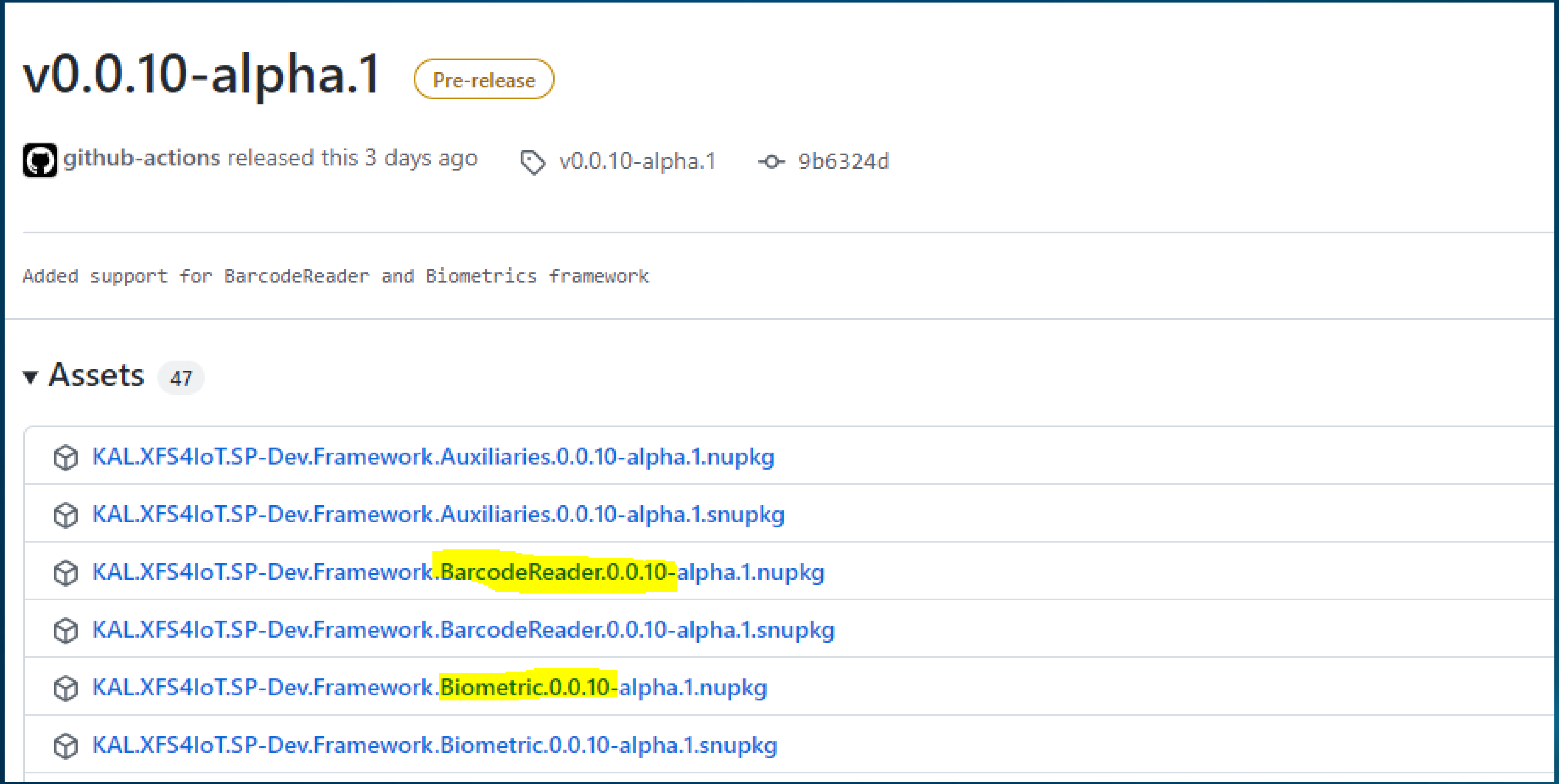
All previous demo videos can be found on the KAL ATM Software YouTube channel:

https://www.youtube.com/user/ATMsoftware/videos

Latest version and next steps

Latest Pre-release support

- Latest Pre-release v.0.0.10-alpha.1 for NuGet packages



v0.0.10-alpha.1 Pre-release

github-actions released this 3 days ago v0.0.10-alpha.1 9b6324d

```
Added support for BarcodeReader and Biometrics framework
```

▾ Assets 47

- KAL.XFS4IoT.SP-Dev.Framework.Auxiliaries.0.0.10-alpha.1.nupkg
- KAL.XFS4IoT.SP-Dev.Framework.Auxiliaries.0.0.10-alpha.1.snupkg
- KAL.XFS4IoT.SP-Dev.Framework.BarcodeReader.0.0.10-alpha.1.nupkg
- KAL.XFS4IoT.SP-Dev.Framework.BarcodeReader.0.0.10-alpha.1.snupkg
- KAL.XFS4IoT.SP-Dev.Framework.Biometric.0.0.10-alpha.1.nupkg
- KAL.XFS4IoT.SP-Dev.Framework.Biometric.0.0.10-alpha.1.snupkg

The next steps

- Support for acceptor and recycler machines
- Migrate framework to .Net6
- XFS4IoT SPs to be developed
- XFS4IoT is already part of RFPs
- XFS4IoT will evolve faster than XFS3 ever did
- The CEN XFS committee is targeting mid of 2022 for the next release of the CEN XFS4IoT Specification

Next call

MS Teams

- First Tuesday of each month at 1300 UK time

Next call: 5th April 2022, 1300 UK, 0800 US EST, 2100 Tokyo time

(Note: **UK** changes clocks on 27th March, no impact for **US** but the call will be 1 hour earlier for **Tokyo** time).

We will change to Zoom calls (instead of Teams) in future

(We will provide interpretation in Japanese, Chinese and Spanish using Zoom's interpretation feature in future)