XFS4IoT SP-Dev Workgroup

5th September 2023

XFS4IoT SP-Dev Workgroup agenda

- Recap from previous meeting
- End to End security
- Updates in SP-Dev Framework v1.7.0
- Next meeting

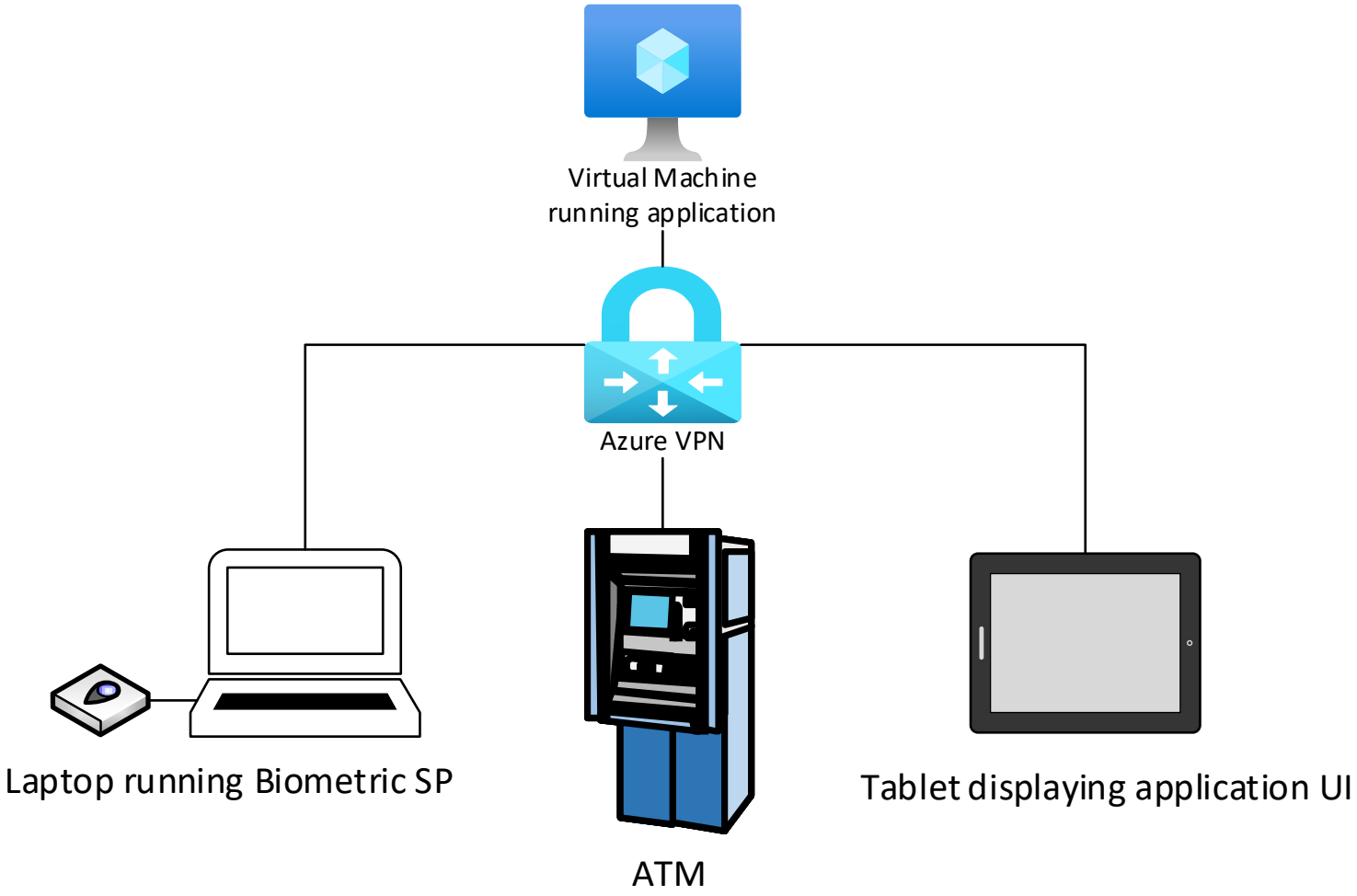
Recap from previous meeting

Recap from previous meeting

- XFS4IoT with KAL Demo
- What's next?

Recap of June Meeting: KAL XFS4IoT Demo

Demonstrating the concept of a deconstructed ATM, highlighting configuration flexibility providing a vision of what XFS4IoT ATMs can be.



- Azure VM running ATM application connected securely to devices using XFS4IoT
- Biometric SP built on the XFS4IoT Biometric sample from GitHub
- KAL provided XFS4 layer running on top of ATM vendor XFS3 SPs
- Microsoft Surface (W10) Tablet connected to ATM application VM via Remote Desktop using Container technology

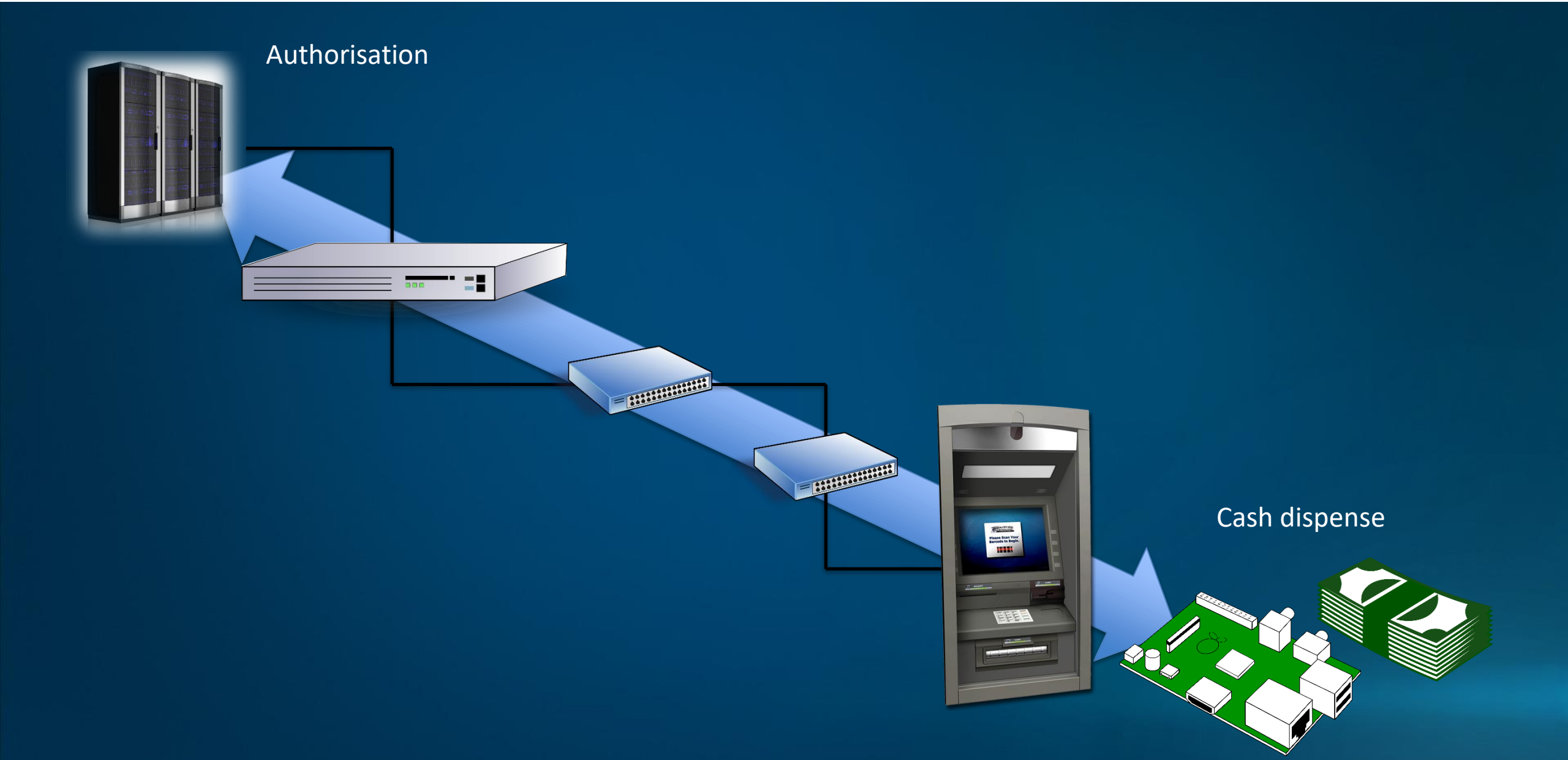
E2E: GetPresentStatus

Security for returned data

Kit Patterson

Confidential

E2E security - recap



E2E security – current attacks

- Current real world attacks
  - **Black box** : Between PC and dispenser
  - **Man in the Middle** : Network between host and PC

These are blocked by E2E security on the Dispense command

E2E security – round trip

Authorisation

2

Dispense Token

Nonce

1

Cash dispense

E2E Security – Incoming and Outgoing

- Dispense token protects the authorisation coming into the dispenser
- Important data also goes *out* of the dispenser
- GetPresentStatus is used to protect against ‘Transaction Reversal Fraud’
- An attacker on the network could change the present status to make it appear that they didn’t receive cash

E2E security – Protect outgoing data

Authorisation

1

Present Token

Nonce

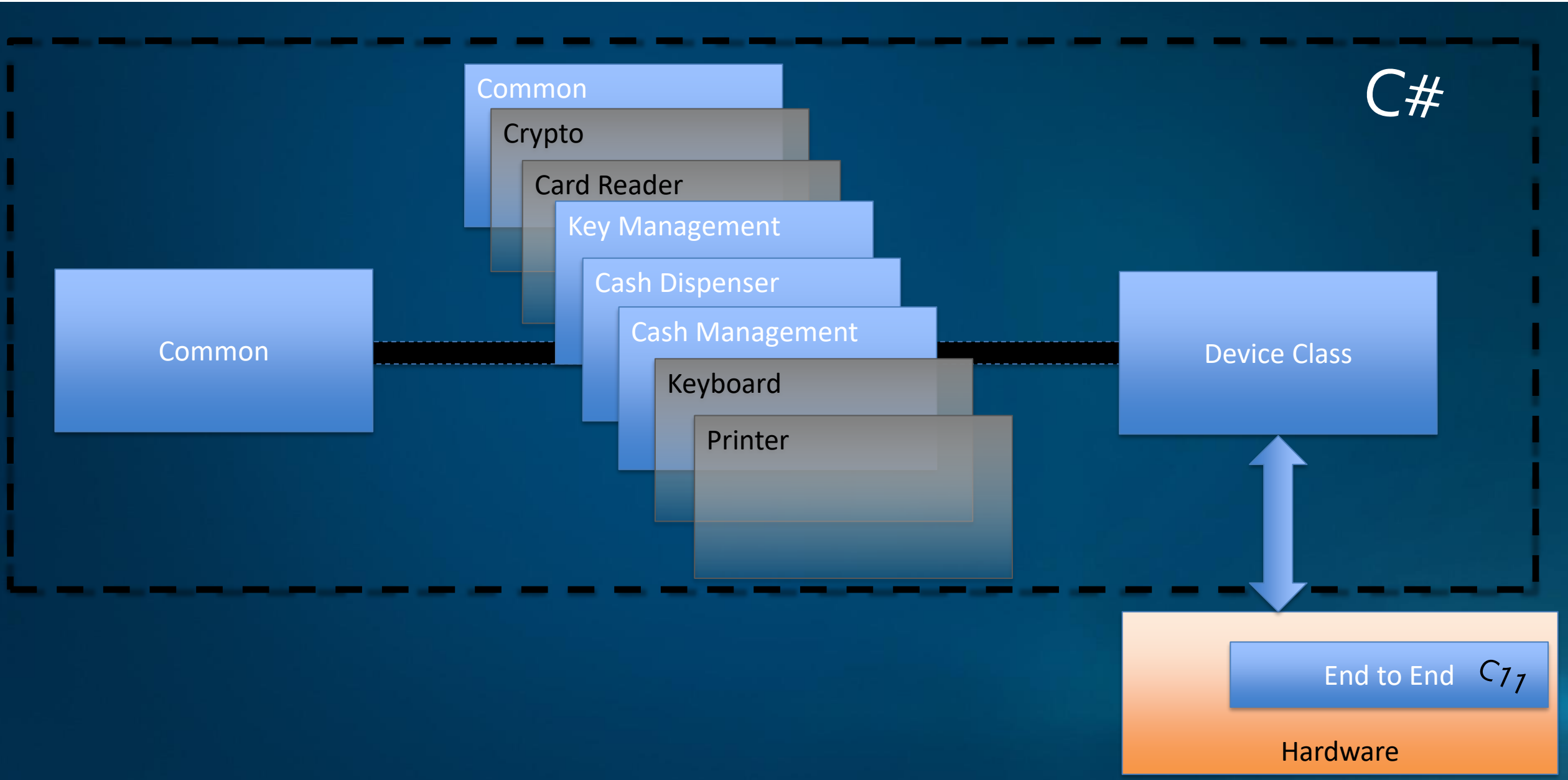
2

Cash dispense

E2E Security – GetPresentStatus token fields

- DISPENSEID – Last dispense ID
- DISPENSED1 – Amount dispensed, USD100.00, or EUR123.32
- PRESENTED1 – YES or NO
- PRESENTEDAMOUNT1 – Amount accessible to customer
- RETRACTED1 – YES or NO
- RETRACTEDAMOUNT1 – Amount, if known, or “?”

SP-Dev Framework E2E security



SP-Dev Framework - GetPresentStatus

```
const int HMAC_SIZE = 32;
static char LastDispenseID[HMAC_SIZE] = { 0 }; // This value should be persistent across power-cycles
bool GetLastDispenseIDSet(){…}
extern "C" bool GetLastDispenseID( bool* KnownValue, char ThisDispenseID[HMAC_SIZE]){…}
extern "C" bool SetLastDispenseID(char ThisDispenseID[HMAC_SIZE]){…}
extern "C" bool GetLastDispenseAmount(unsigned int* UnitValue, unsigned int* SubUnitValue, char Currency[3]){…}
extern "C" bool GetLastDispensePresented(bool* Presented){…}
extern "C" bool GetLastPresentedAmount(unsigned int* UnitValue, unsigned int* SubUnitValue, char Currency[3]){…}
extern "C" bool GetLastDispenseRetracted(bool* Retracted){…}
extern "C" bool GetLastRetractedAmount(bool* ValueKnown, unsigned int* UnitValue, unsigned int* SubUnitValue, char Currency[3])
extern "C" bool NewHMAC(char const* const Token, size_t  TokenLength, unsigned char* const TokenHMAC){…}
```

- To implement E2E security, implement these functions
- Values *must* be stored persistently
- NewHMAC must use strong encryption

SP-Dev Framework - GetPresentStatus

- SP-Dev Framework will handle the rest
  - Receiving nonce
  - Constructing token from firmware values
  - Returning full token
- The sample SP code includes GetPresentStatus E2E support
- Currently only supports one currency at a time

What's new in the latest release v1.7.1

JSON Schema Validation Support

- JSON Schema published by XFS Committee
- SP-Dev Framework injection option to add message validation

- Implemented by the SP-Dev Framework user to fit their needs
- Example implementations provided in SP-Dev Samples repository

ISensitiveDataFormatter interface

- Added ISensitiveDataFormatter interface for logging customisation
- Can be used to format message content output through the ILogger interface
- Can be used to mask full track data, pinblock, or other fields as required

- Implemented by the SP-Dev Framework user to fit their needs
- Default implementation provided does not alter any logging

Updates in v1.7.1

- Added new “ServerPort” configuration
- Allow receiving Null or Zero value retract index for CashManagement Reset and Retract commands
- Allow Device implementation to handle Printer Graphic field when FORMAT is not specified

Updates in v1.7.1

- Added ToString override for all messages to output as JSON
- Cancel any active or queued commands when a client disconnects
- Improve handling large messages from clients. Fix hang if message doesn’t fit in the default buffer
- Fix exception in Dispense handler when mix values are invalid

Recap on available resources

Available resources

- SP-Dev Framework packages
  https://www.nuget.org/profiles/KAL-ATM-Software
- SP-Dev Framework Project Templates
  https://www.nuget.org/packages/KAL.XFS4IoT.SP-Dev.Framework.Templates
- Samples and Test Clients
  https://github.com/KAL-ATM-Software/KAL_XFS4IoT_SP-Dev-Samples

Available resources

- SP-Dev Documentation
  https://kal-atm-software.github.io/KAL_XFS4IoT_SP-Dev-Documentation/

- Workgroup presentations
  https://github.com/KAL-ATM-Software/KAL_XFS4IoT_SP-Dev/tree/main/Workgroup%20meeting%20presentations

- Demos videos
  https://www.youtube.com/playlist?list=PLS7hfupK8VAMhjsk44zSndx5WTUZQENe2

What's next?

Demos, POCs and guest speakers

- Demos
  - ATM application in the cloud
  - Standalone device(s)
  - TLS support
- POCs
  - Different OS and Cloud Platforms
  - Various hardware types
- Guest speakers

Next call

Zoom

- First Tuesday of each month at 1300 UK time for 30 mins

Next call: 3rd October 2023
1300 UK, 0800 US EST, 2100 Tokyo time

Calls are 30 mins long

We will continue to use Zoom
(Interpretation in Japanese, Chinese and Spanish is available using Zoom's interpretation feature)