

XFS4IoT SP-Dev Workgroup

7 December 2021

- Recap from previous meeting
- Guest speaker
- Printer release, sample and demo
- Lights release, sample and demo
- E2E security follow up
- Status, roadmap and what's next

- Frameworks for most major ATM devices now available with C# and C++ sample code released, demos on YouTube:
 - Card Reader (released May 2021) with support for dispensing (released November 2021)
 - Cash Dispenser (July 2021), without end-to-end security
 - Text Terminal Unit (July 2021)
 - EPP Key Management and Crypto classes (Sept 2021)
 - Keyboard and PinPad classes (October 2021)
 - End-to-end security partially complete (October 2021); added required functions (November 2021)
 - Printer / Lights (Today's meeting)
- Frameworks yet to be done in order to support a complete Cash Out ATM
 - Vendor Mode / Vendor Application / Auxiliaries

- Framework code updated regularly. Framework code is available to:
 - Implement XFS4 SPs
 - Review, test with our samples
 - Write test tools

Guest speaker - Serquo

Serquo-KAL

XFS4IoT Presentation

Serquo, a new player in the XFS world?

- Serquo (formerly Cashware), is a company created in 1991
- SW company independent of any hardware manufacturer
- Member of the European Committee for XFS Standardization (CEN/XFS)
- Former member of CEN/JXFS
- Providing XFS software since 1995, using WOSA/XFS 1.x
- Developing financial device interfaces using XML/JSON over TCP/IP since 2007
- Mainly focused on QA tools, development and consultancy for financial devices connectivity

Why is Serquo investing in XFS4IoT?

- Mainly because we believe in this new architecture of XFS
- It is clear to us that XFS4IoT is the future for new self-service applications and that is why:
 - We are adapting our QA tools (Atmirage, Quarterback, and Atmosphere Testing Center) to the new XFS4IoT standard
 - We need a new service provider development framework based on XFS4IoT to serve third party companies and create new device drivers

Our technical vision



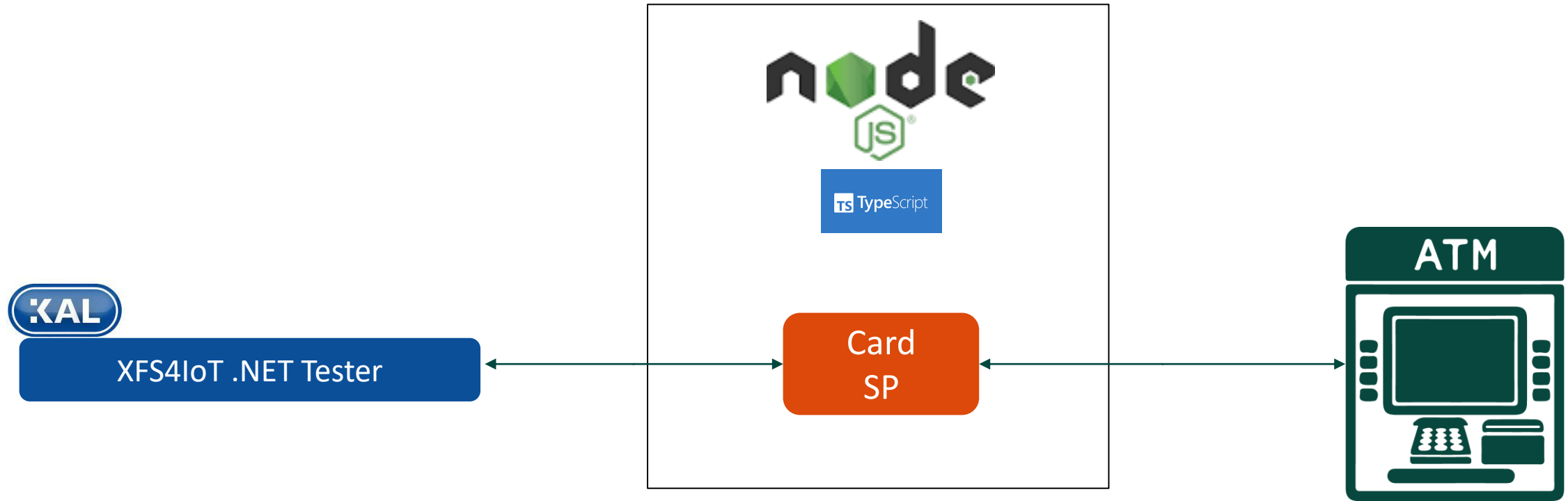
Free
Open source
Cross platform
Runs on Google's V8 JS Engine
Fast to market development
Active user community
Easy to learn

Free
Open source
Object oriented
Active user community
Similar to JavaScript
Ideal for handling JSON structures
Easy to learn

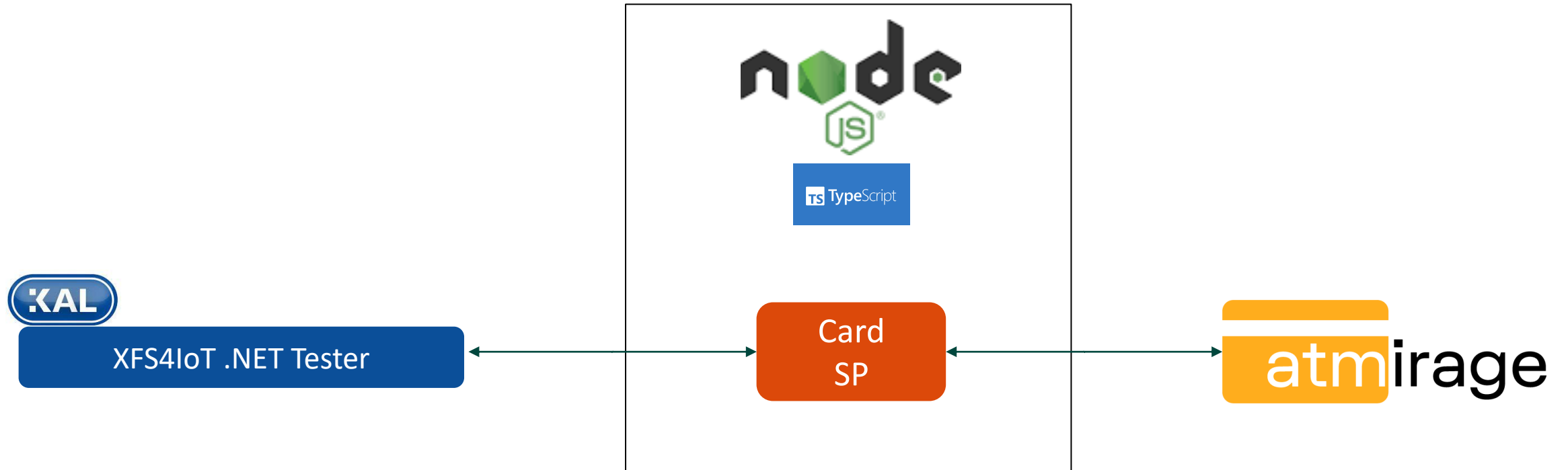
The company that is betting the
most on XFS4iot
Proven XFS experience over
many years
**We use KAL's tester to verify
our developments**

Although we do not use their
framework because we do not
have .NET and C# skills

Today's demo



Today's demo



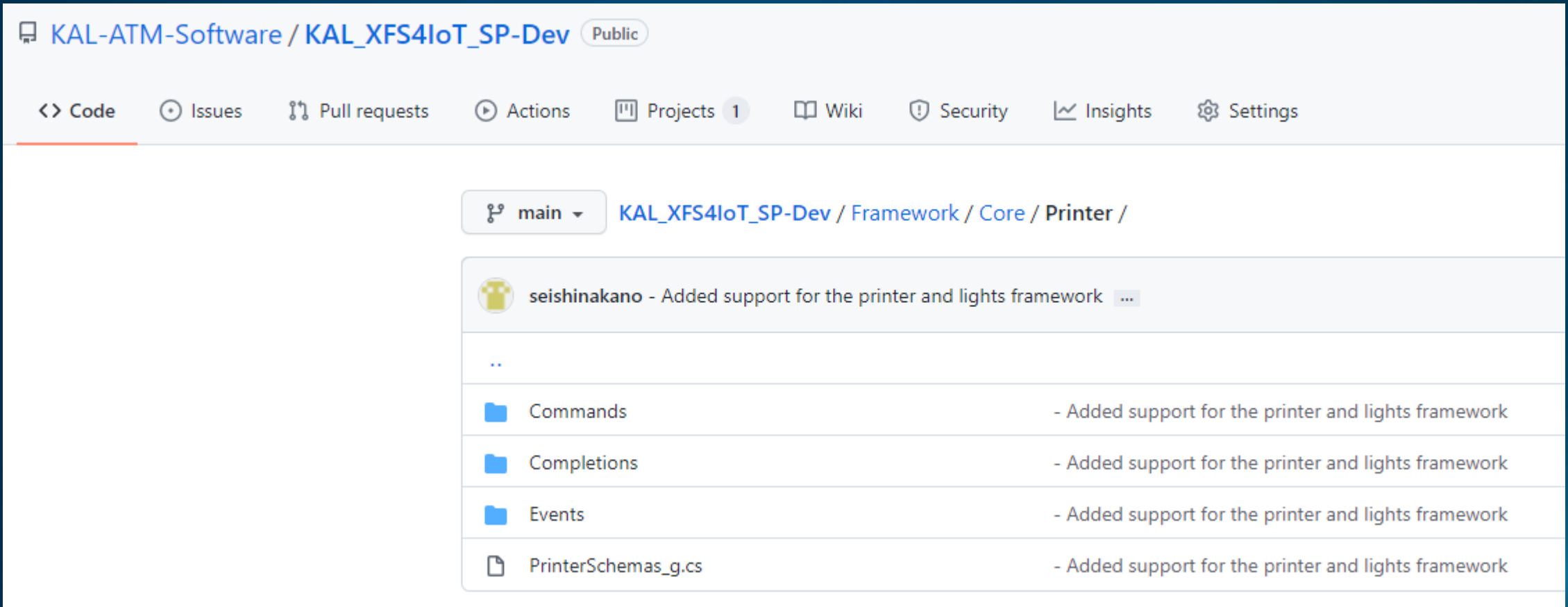
Thank you!



Printer and Lights classes release

© 2021 KAL ATM Software GmbH (KAL)

- Printer class now available:



KAL-ATM-Software / KAL_XFS4IoT_SP-Dev Public

<> Code Issues Pull requests Actions Projects 1 Wiki Security Insights Settings

main KAL_XFS4IoT_SP-Dev / Framework / Core / Printer /

seishinakano - Added support for the printer and lights framework ...

..

Commands	- Added support for the printer and lights framework
Completions	- Added support for the printer and lights framework
Events	- Added support for the printer and lights framework
PrinterSchemas_g.cs	- Added support for the printer and lights framework

Printer and Lights support



- Details of the changes in the following commit:

main

Commits on Dec 2, 2021

Changed key name of the build-in mixes available to be consistent wit...
seishinakano committed 4 days ago

- Added support for the printer and lights framework
seishinakano committed 4 days ago

KAL-ATM-Software / KAL_XFS4IoT_SP-Dev Public

Code Issues Pull requests Actions Projects 1 Wiki Security Insights Settings

- Added support for the printer and lights framework

- Updated most recent XFS4IoT specification in November.
- XFS4IoT payload is handled by the framework for events.
- Updated E2E failure cases of dispensing cash, partial dispensing operation.

main
v0.0.7-alpha.2 v0.0.7-alpha.1

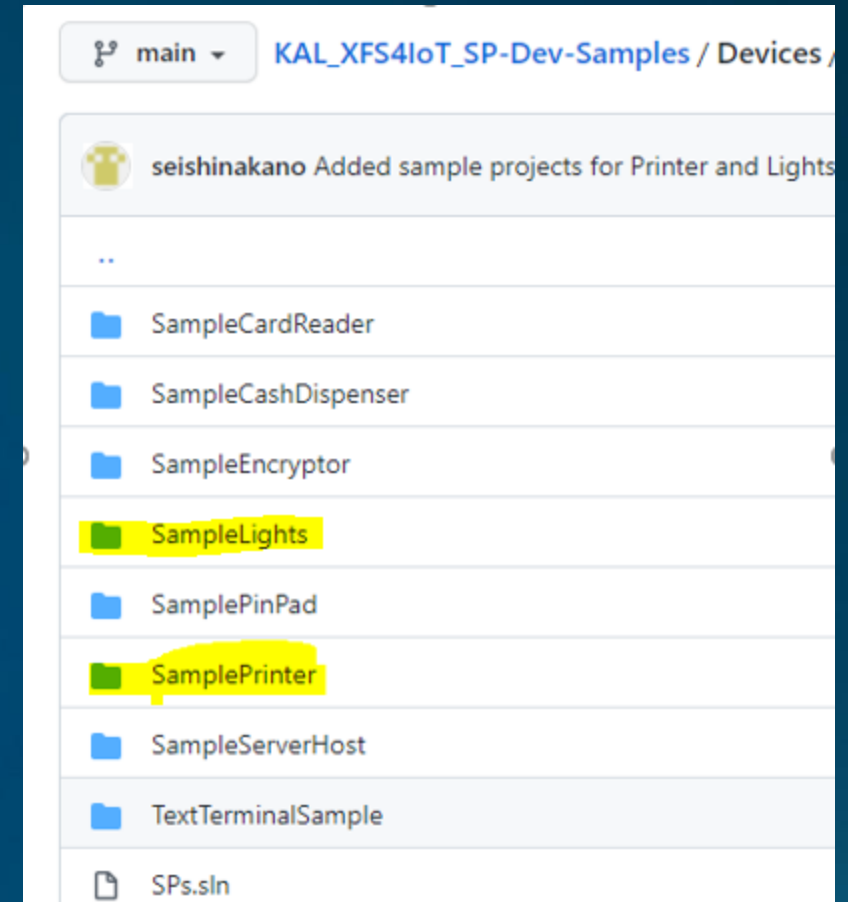
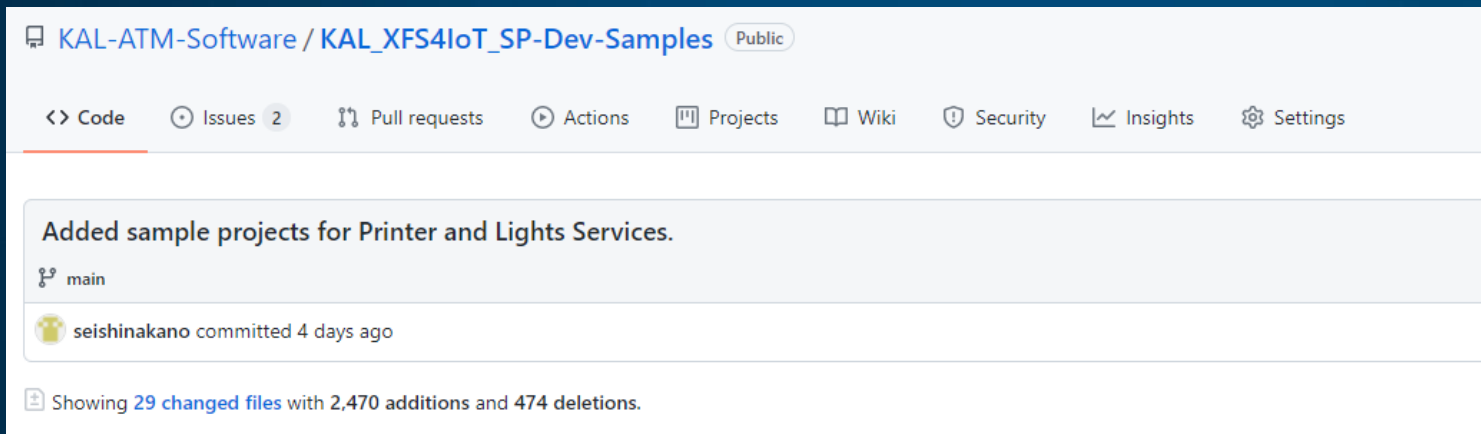
seishinakano committed 4 days ago

Showing 465 changed files with 19,333 additions and 4,169 deletions.

Printer and Lights support



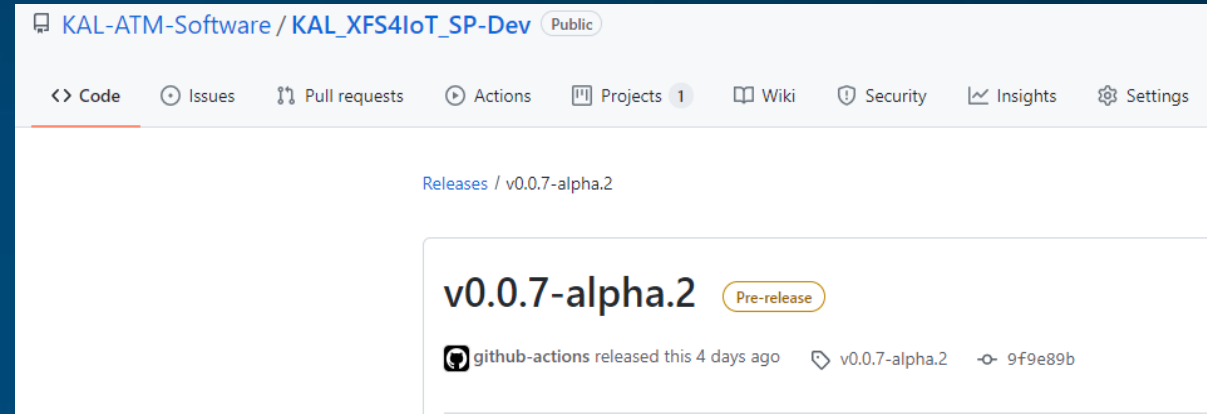
- Sample available
- Specific commit for sample changes:



Printer and Lights support



- Latest pre-release
v.0.0.7-alpha2
for NuGet packages



- Printer full support including:
 - Form and Media support
 - PrintForm() and PrintRaw() methods
 - ControlMedia() to eject, retract, flash data...
 - Reset

This class interfaces for all printer devices such as Receipt printer, Journal printer, Statement printer...

- Lights interface contains only one handler: SetLightHandler.cs
- All pre-defined guidelights are supported across the different devices like Card reader, Cash dispenser...
- All LEDs and vendor specific lights are supported too through this interface

Lights sample demo



Demo with real Printer

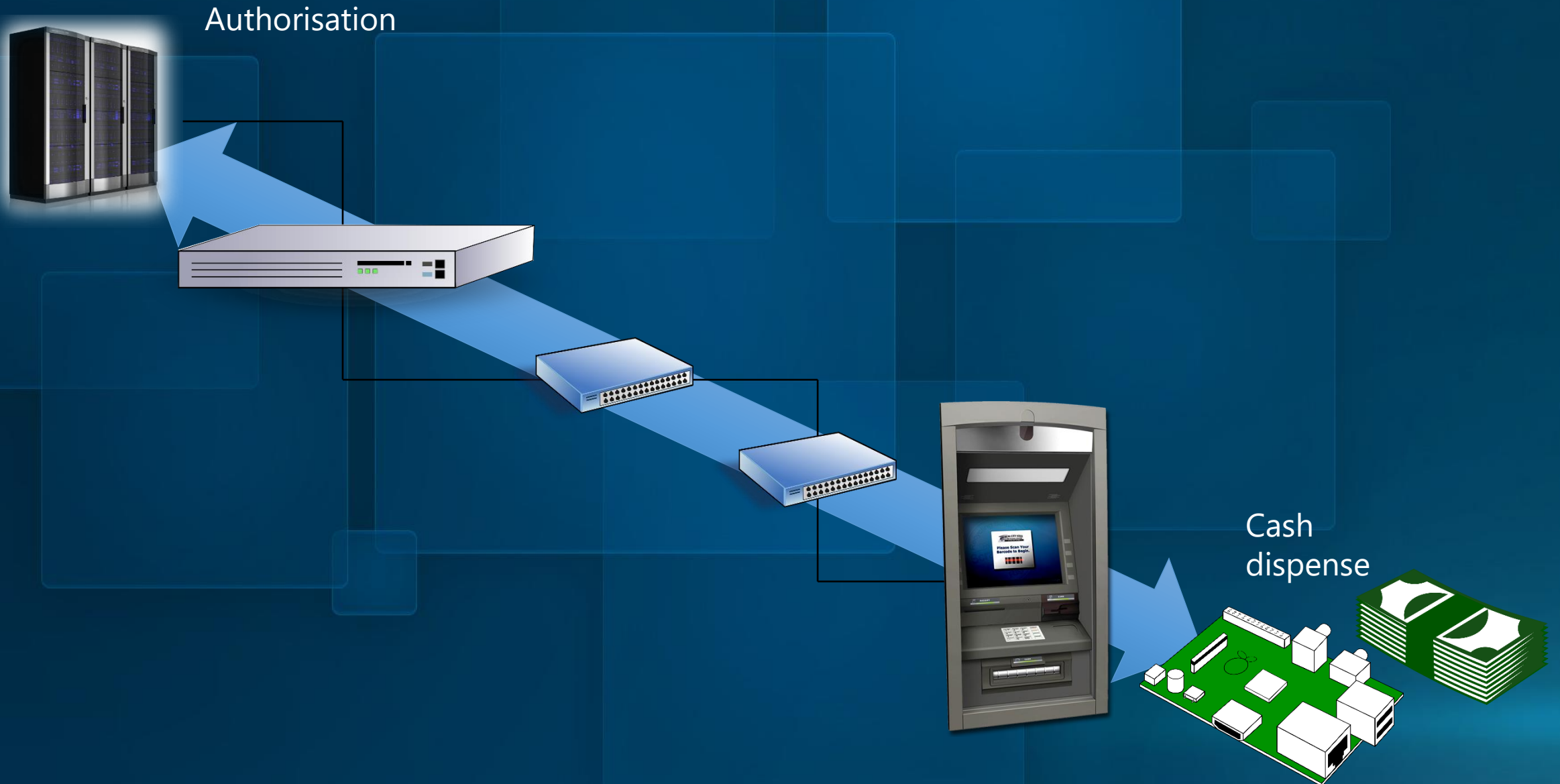
All demo videos with real devices will be available on YouTube.

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

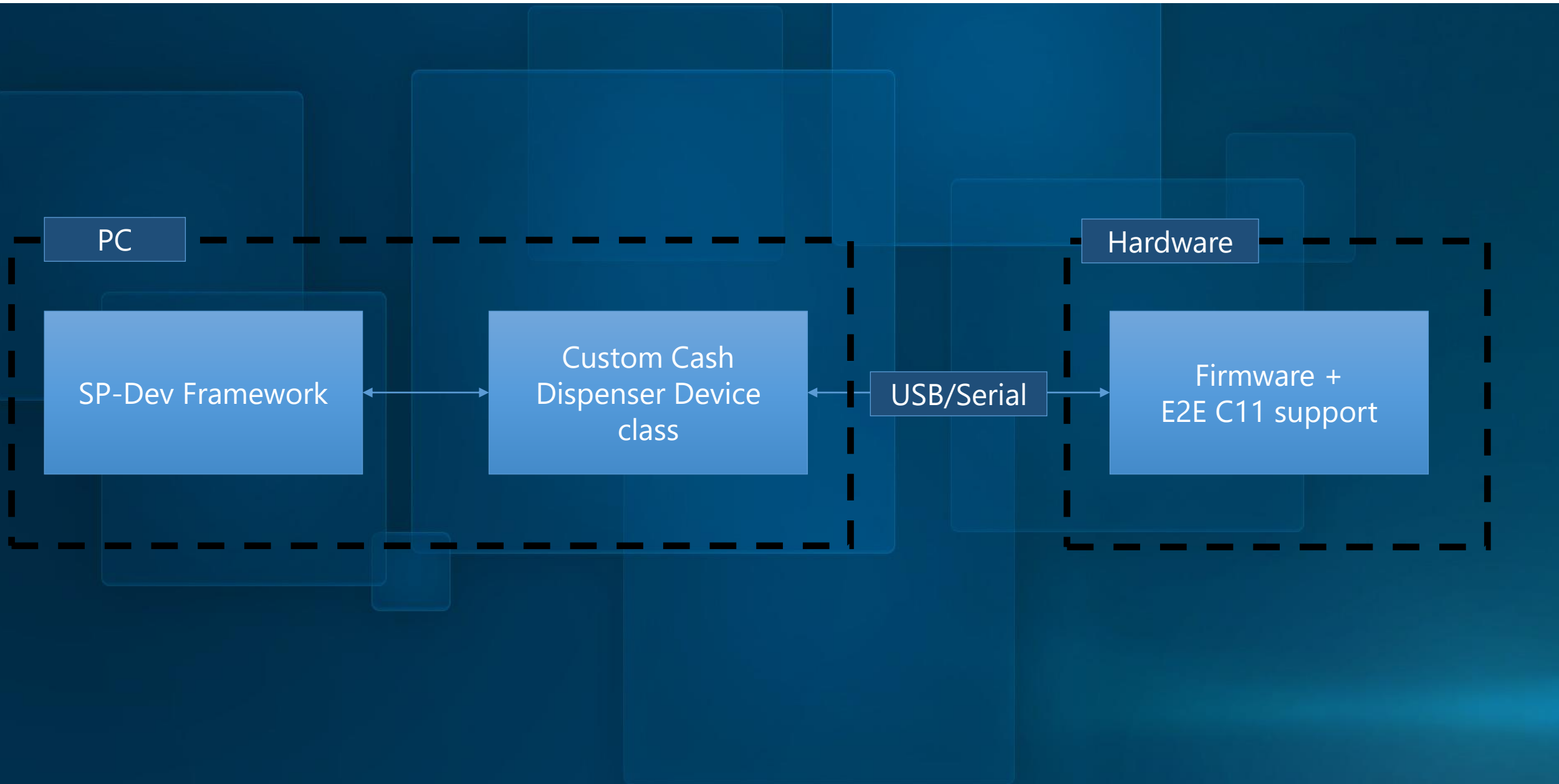
<https://www.youtube.com/user/ATMsoftware/videos>

XFS4IoT SP-Dev E2E support

E2E support, December update



Components



- Multi-part dispense – dispense large amounts in multiple bundles
- Retry failed dispense
 - Stacking dispenser fails to pick notes. Need to retry with a different amount.
 - Spray dispenser dispenses part of the total value. Need to dispense the remainder.

Sample 'firmware' - November



```
// Check that the token is valid and authorises the requested dispense.
// Include null in token (buffer) size.
auto Authorised = AuthoriseDispenseAgainstToken(utf8Token.c_str(), utf8Token.size() + 1, Value, 0, utf8Currency.c_str());
if (!Authorised)
    return false;

auto TokenValues = GetDispenseKeyValues();
cout << "Got dispense token values: Currency:" << string(TokenValues->Currency, 3) << " value:" << TokenValues->Value << " cents:"

// Simulate the actual dispense
cout << "dispensing: " << TokenValues->Value << "." << TokenValues->Fraction << string(TokenValues->Currency, 3) << "\n";
Sleep(1000);
cout << "dispensed\n";
```

Sample 'firmware' - December



```
// Check that the token is valid and authorises the requested dispense.
// Include null in token (buffer) size.
auto Authorised = AuthoriseDispenseAgainstToken(utf8Token.c_str(), utf8Token.size() + 1, Value, 0, utf8Currency.c_str());
if (!Authorised)
    return false;

auto TokenValues = GetDispenseKeyValues();
cout << "Got dispense token values: Currency:" << string(TokenValues->Currency, 3) << " value:" << TokenValues->Value << " cents:

// Simulate the actual dispense
auto dispenseResult = Dispense(TokenValues);

// At this point the "dispense" has completed successfully, so we must confirm that to E2E security.
// If the dispense failed we just skip this, or even confirm a partial amount if only part of the
// cash was dispensed.
if( dispenseResult )
    ConfirmDispenseAgainstToken(utf8Token.c_str(), utf8Token.size() + 1, Value, 0, utf8Currency.c_str());
```

```
extern C_LINKAGE bool AuthoriseDispenseAgainstToken(char const *const Token, size_t TokenLength,  
                                                    unsigned int UnitValue, unsigned int SubUnitValue, char const Currency[3]);  
extern C_LINKAGE bool ConfirmDispenseAgainstToken(char const *const Token, size_t TokenLength,  
                                                  unsigned int UnitValue, unsigned int SubUnitValue, char const Currency[3]);  
extern C_LINKAGE bool InvalidateToken();
```

- Complete
 - Added firmware C library support (including unit tests.)
 - Framework support for Common.GetCommandNonce and CashDispenser.Dispense token commands.
 - Sample SP implementation using 'firmware' code and dispenser implementation
 - (Command line) test client showing dispense and present sequence
 - Read and track token keys – such as dispense amount
 - Enforce dispense values
 - Automatically delete the nonce/invalidate tokens
 - Support multiple partial and failed dispense and retry

- Support for generating response tokens
- GetPresentStatus token support



Status, roadmap and what's next

- Support for all main devices on Cash out ATM
- Support of the new E2E security and Service Discovery features
- Samples available for all the classes supported
- Demos using the real devices available publicly

- Integration and use in different ways
 - Integration with **RedHat Linux** with containers (August meeting)
 - Testing tool to help verify development for **Serquo**
- Workgroup members all around the world using and providing us feedback on the framework

- Support for full Cash Out ATM ready
- Next classes:
 - Vendor mode/Vendor application
 - Auxiliaries
- CEN XFS4IoT Specification official release is coming
- Framework will be updated regularly

- XFS4IoT SPs to be developed
- XFS4IoT will be part of future RFPs (already the case)
- XFS4IoT will evolve faster than XFS3 ever did
- The first ATMs running XFS4IoT will be coming soon...

MS Teams

- First Tuesday of each month at 1300 UK time

No call in January

Next call: 1st February 2022, 1300 UK, 0800 US EST, 2200 Tokyo time