Design and application of

# SAVOIR File Management System Library

for ICE Cube OBCU

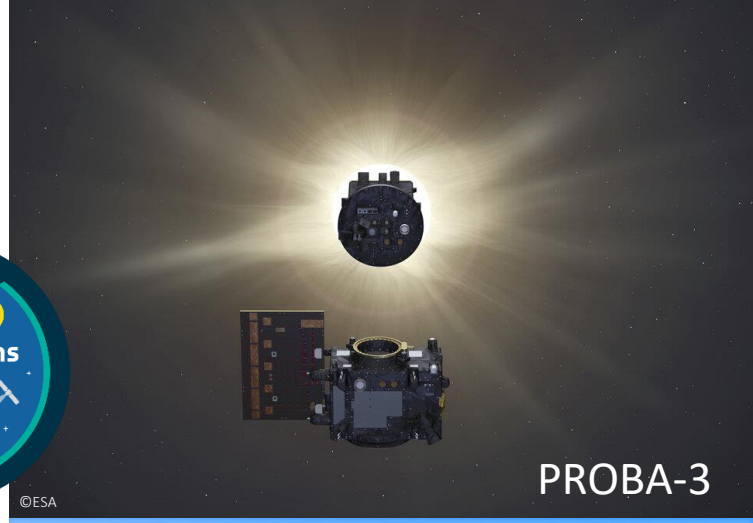
Michał Kostrzewa
michal.kostrzewa@n7space.com

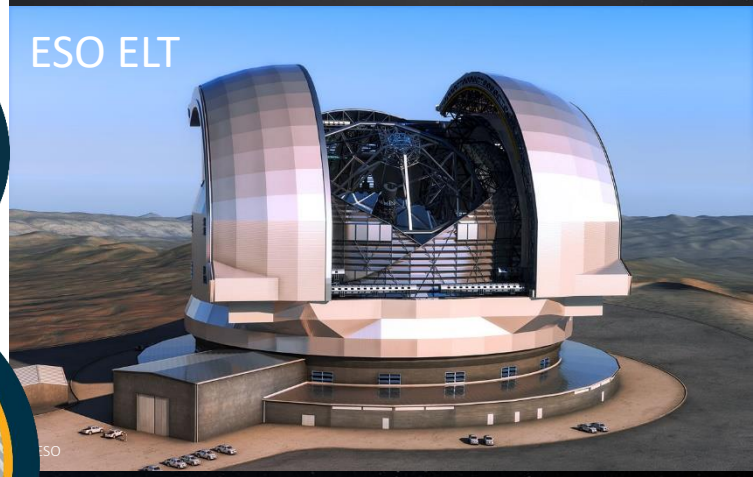
www.n7space.com

2025

# About N7 Space

- 10+ years of experience in software development for space systems
- Key activities:
  - Critical software development for real-time embedded systems
  - Applications of Model Based Software Engineering
  - Software development for telescope control systems
- Team of 25 experts in critical software domain
- Over 30 projects executed in company history
- Headquarter: Warsaw, Poland




# SAVOIR


Space AVionics Open Interface Architecture

SAVOIR-MASAIS working group (Mass Storage Access Interfaces and Services) produced SAVOIR DSSRD (2017) and DSSIRD (2020)

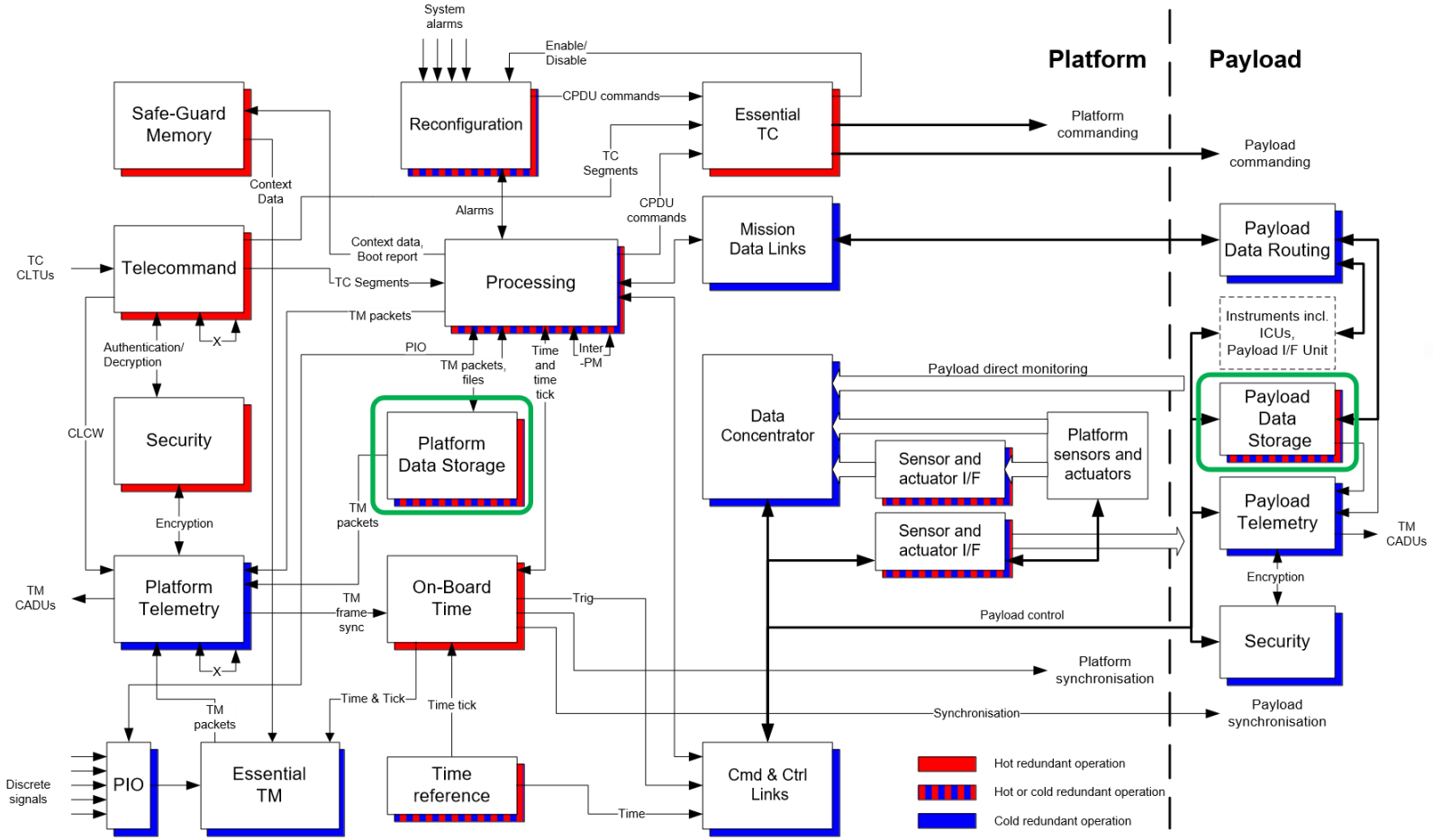
ESA activity to implement a reference implemenation of SAVOIR File Management System

# SAVOIR Data Storage System

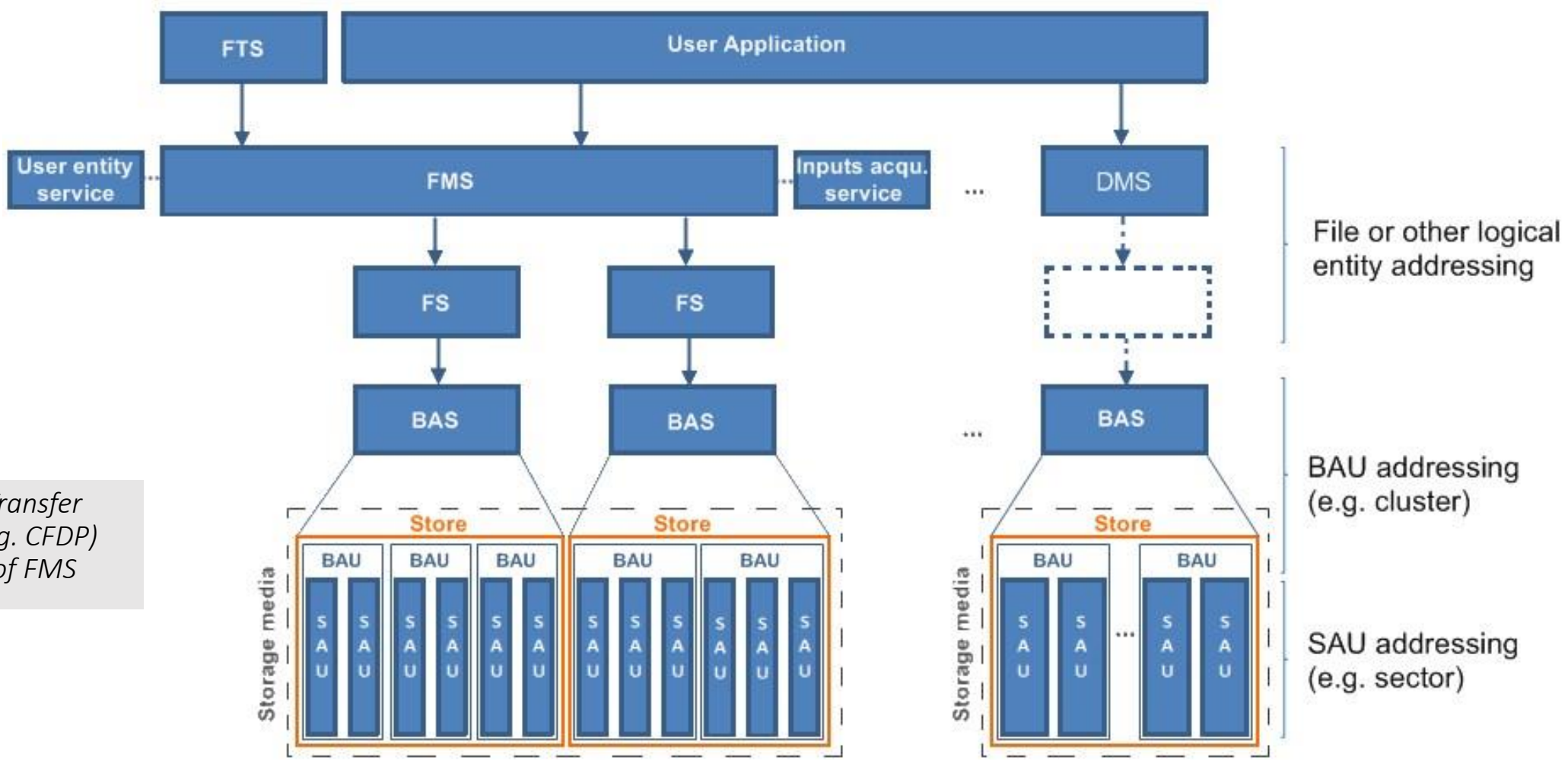
Within SAVOIR OSRA (On-Board Reference Architecture)
SAVOIR DSS realizes:

- Platform data storage
- Payload data storage

Notably NOT Safe-Guard Memory



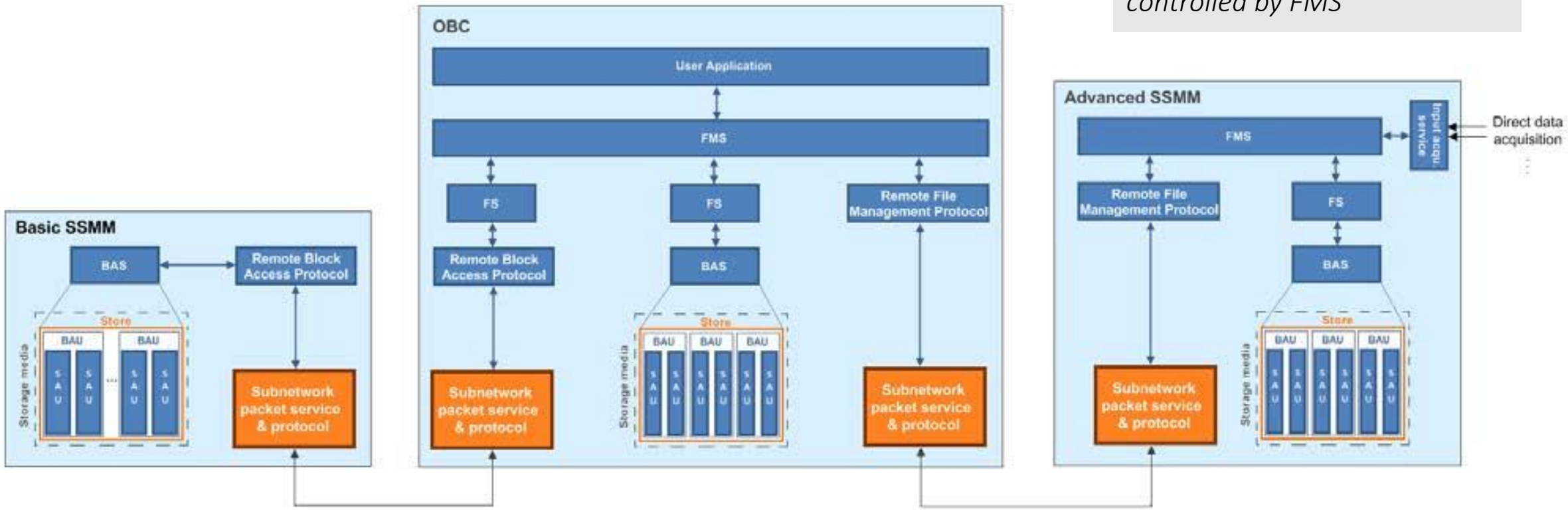
# SAVOIR DSS – layered architecture




*Note: File Transfer Services (e.g. CFDP) are clients of FMS*

# SAVOIR DSS – remote MM

*Note: Direct autonomous data acquisition realized using Input Acquisition Services (IAS) controlled by FMS*

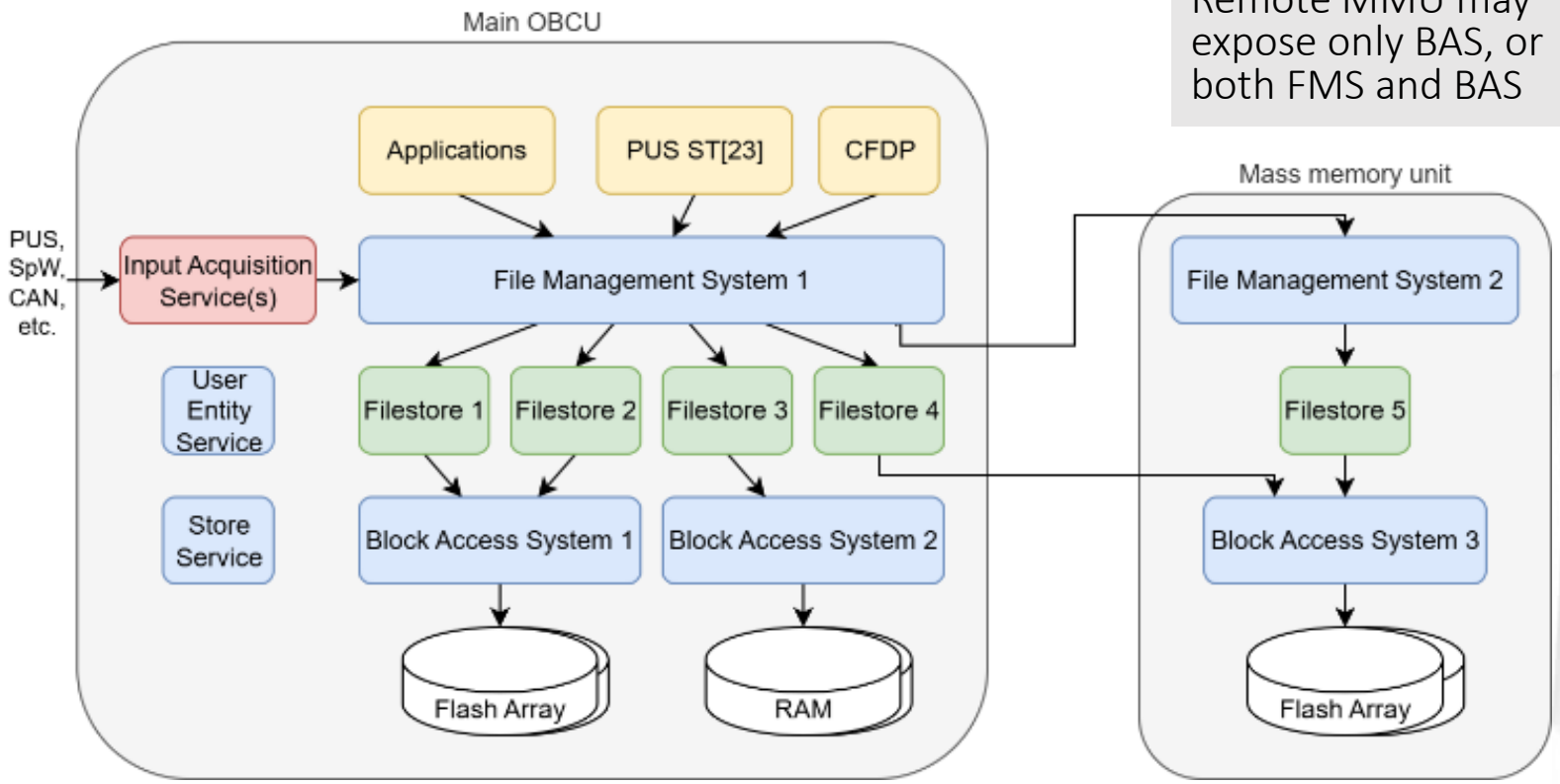


# FMS Library

- Reference, generic implementation of SAVOIR FMS
- C++17
- Targets criticality cat. C aiming at future upgrade to cat. B
- „Roughly” MISRA C++:2023-compliant (for now)
- Platform-agnostic; exposes interfaces and abstraction layers that adapt it to specific OS and hardware
- Complementary Data Storage Simulator

# FMS Library features

- composability (FMS can be sharded across spacecraft's network, transparently to the users)
- platform independence, protocol agnosticism
- automatic input acquisition to files and directories based on filtering criteria

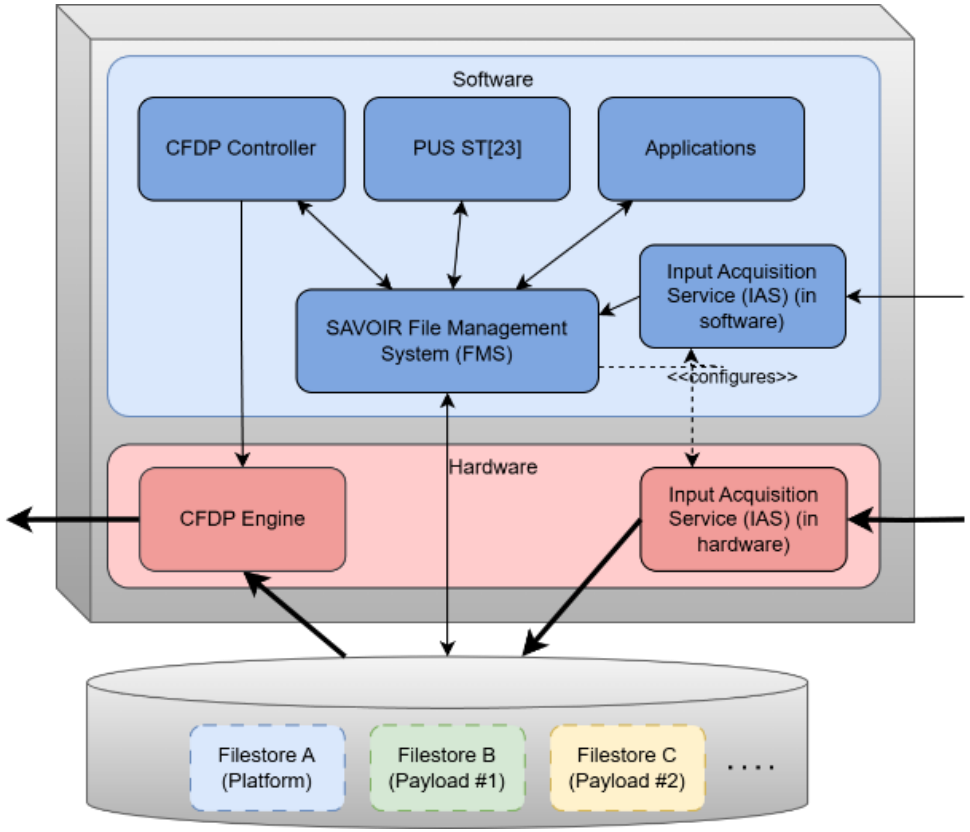



Remote MMU may expose only BAS, or both FMS and BAS

IAS may reside on remote MMU and map to any FMS on-board („proxy mapping")

Remote MMU can be a 3rd party device as long as it's roughly SAVOIR-compliant (FMS is protocol agnostic)

# FMS Library features




IAS configuration protocol is not defined in SAVOIR; FMS library exposes a „simple” interface that could be adapted per platform

- possible offloading of input acquisition and CFDP engine to hardware (platform dependent)
- „Bring-Your-Own-Filesystem” concept; various configurations:
  - LittleFS filesystem
  - SPIFFS filesystem*
  - FreeRTOS+FAT filesystem*
  - Dhara Flash Translation Layer
  - an in-house raw Flash-based filesystem
- in-flight reconfigurability (store resizing and partitioning, handling of multiple different filesystem and storage media types at the same time)
- event generation
- persistent file locking mechanisms

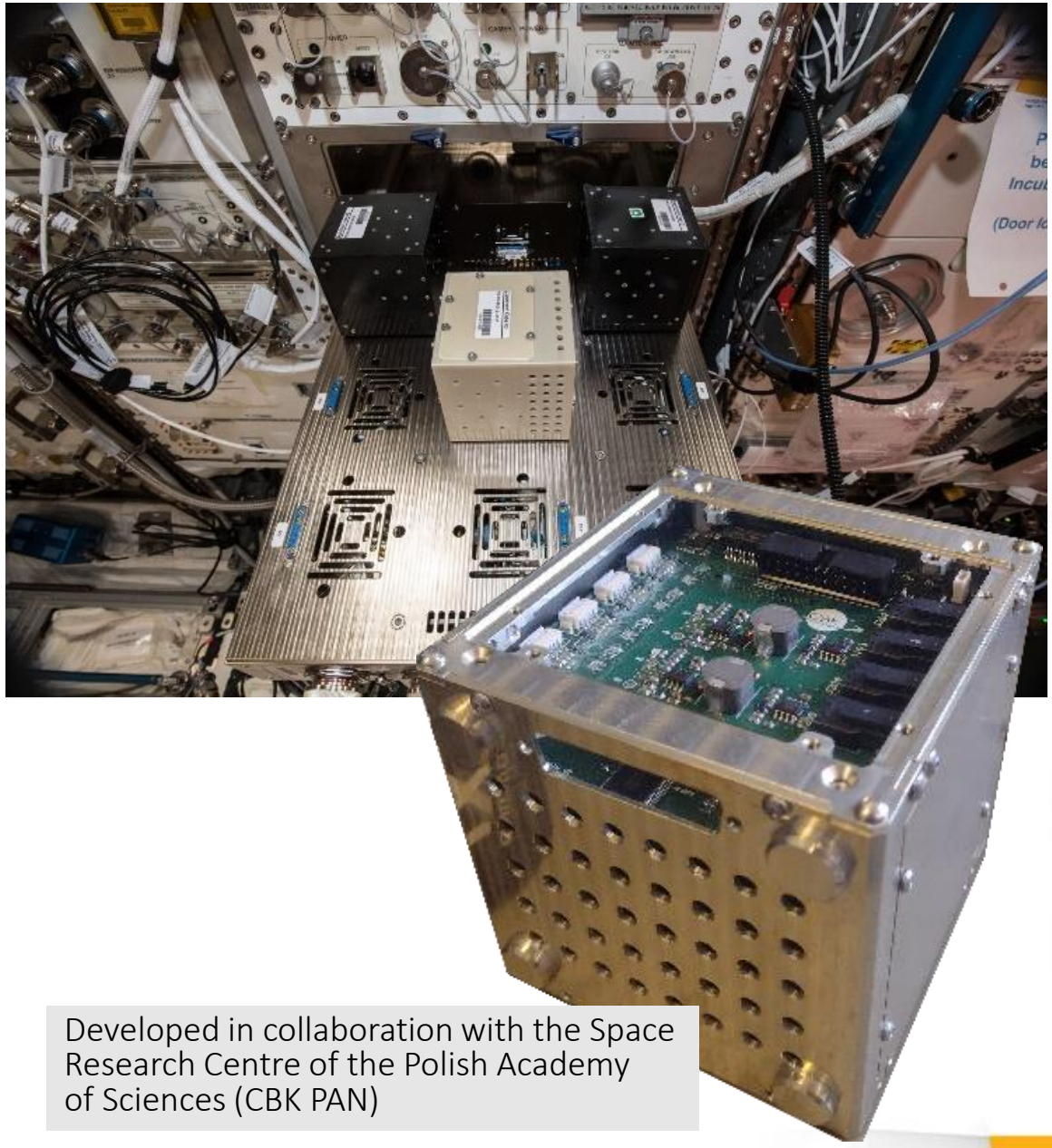
* Not included in baseline release

# ICECube OBCU

*On-board Control Unit for ICE Cube Experiments*

Originally intended for ICE Cubes Facility (ICF) on-board ISS

- PC-104, 1U CubeSat form factor
- SAMV71Q21RT processor
- 16+ Gbit NAND
- Software: FreeRTOS, PUS-C, CFDP, SAVOIR-compliant BSW (crit. B), lwIP
- 100 Mbps Ethernet, UART, GPIO, I2C, CAN





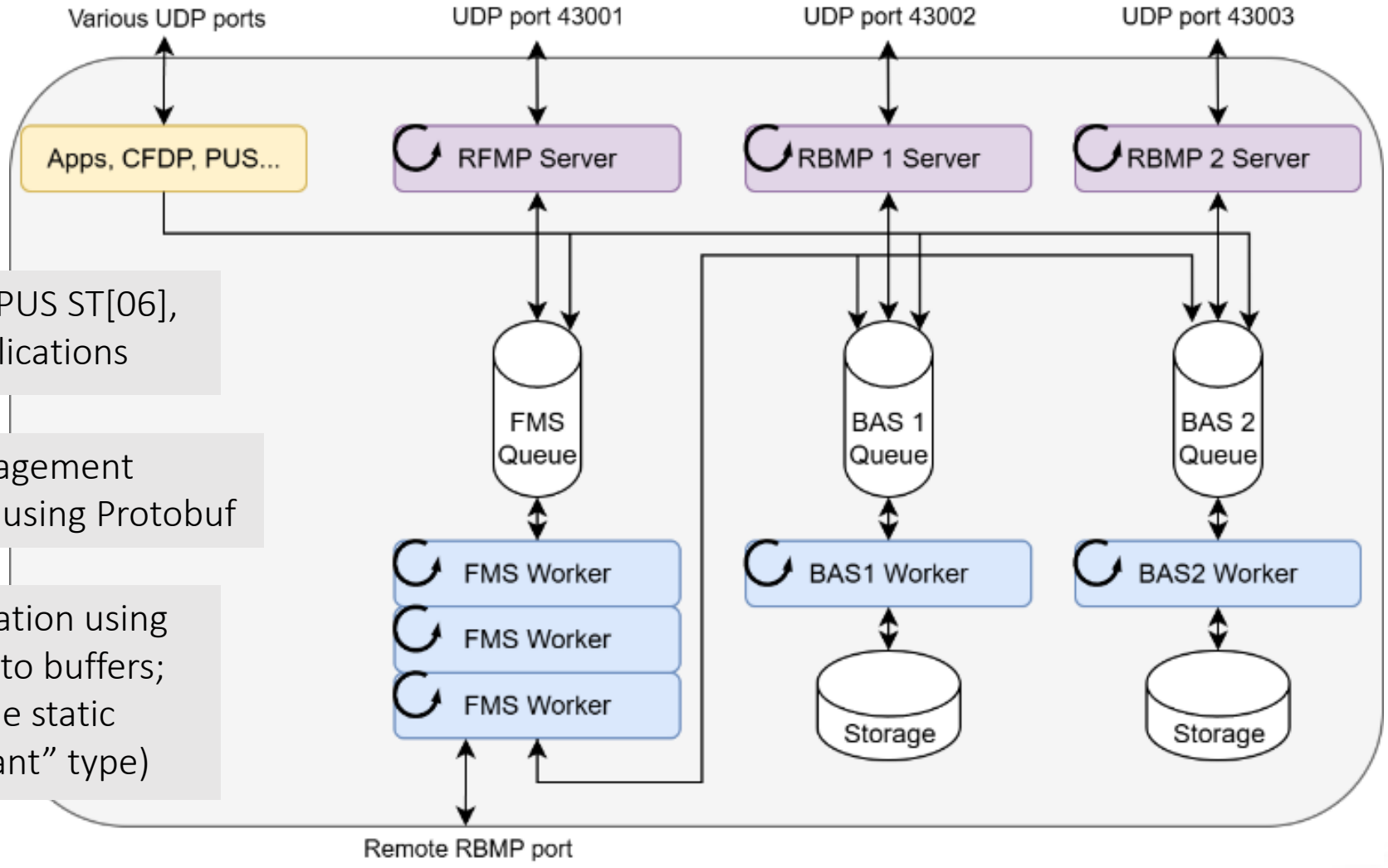
Developed in collaboration with the Space Research Centre of the Polish Academy of Sciences (CBK PAN)

# Implementation for FreeRTOS

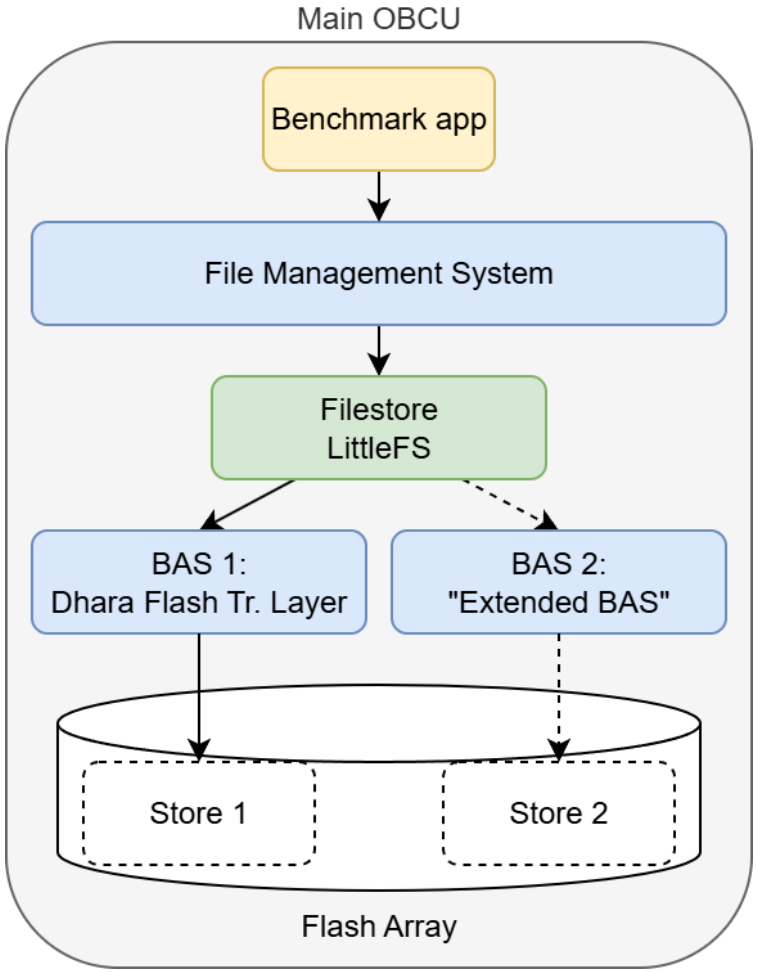
FMS interfaced with PUS ST[06], ST[23] and CFDP applications

Remote File/Block Management Protocols implemented using Protobuf

Inter-task communication using queues and pointers to buffers; relies on compile-time static polymorphism („variant” type)



# Benchmark setup – platform storage use case




Payload Storage is usually hardware dependent, hence benchmarks made *in software* for Platform Storage use case

Filestore choice:
LittleFS (power-loss resilience, checksums);
separate read/write and erase size (designed for NAND)

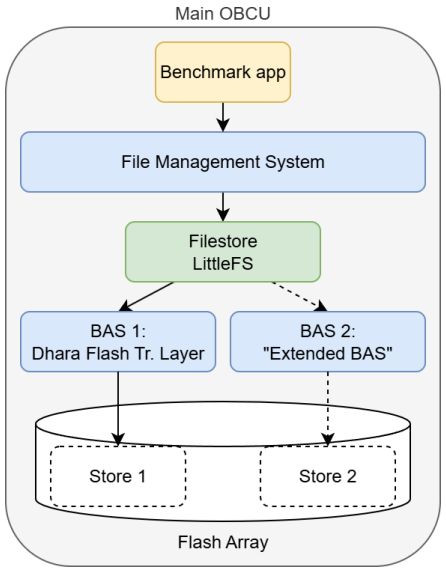
Block Access System choice:
Two variants tested: Dhara FTL-based BAS and „Ext. BAS”

SAVOIR enforces single read/write/erase size (= Block Access Unit size). To avoid large read/write quantum (~MB) that could lead to data loss, we must:

- use FTL translation, or
- modify BAS interface: have Erasable Unit size

# Platform storage – results

Best performance when FMS and BAS worker interval = 5 ms




| Operation | Dhara FTL | Ext. BAS |
|---|---|---|
| BAS.ERASE_BAU | 9.4 ops/s | 1.3 ops/s |
| BAS.WRITE_BAU | 3240 kiB/s | 3644 kiB/s |
| BAS.READ_BAU | 1104 kiB/s | 5947 kiB/s |
| FMS.WRITE_TO_FILE+LittleFS | 140 kiB/s | 83 kiB/s |
| FMS.READ_FROM_FILE+LittleFS | 220 kiB/s | 107 kiB/s |

- Positive effects of Dhara FTL: caching, deferred NAND block erases; Negative: bookkeeping during read
- Non-trivial I/O access patterns of FMS, LittleFS and FTL
- Some FMS I/O is configurable, e.g. can turn off storing of „Last Modified Time” and „Last Access Time”

# Conclusions

Over 80 points of feedback to SAVOIR specifications delivered to ESA

Two main challenges identified:

1. unwieldiness of BAS specification when it comes to Flash based filesystems and storage
2. lacking definition of autonomous data acquisition (interaction of FMS and IAS) makes it difficult to design generic „building block” software and hardware

Library licensed under ESA Software Community License – Type 3 Permissive (v2.4)

Pending publication via European Space Software Repository (ESSR), and possibly N7 Space’s github profile (github.com/n7space)

# Thank you

Michał Kostrzewa
michal.kostrzewa@n7space.com

www.n7space.com