Reaching into space TOGETHER

FILE MANAGEMENT SERVICES INTERFACE STANDARDISATION

Final Presentation Days 09.06.2016

Presenter: Arnaud Bourdoux ESA TO: Christophe Honvault





FMSIS FP - Agenda

- Introduction What, who and why ?
- Study logic
- Study results:
 - Survey of past, present and future mass memories
 - From User, to Technical, to Protocol specification
 - The AAML framework, and its FMSIS dedicated extensions
 - A use case and a demonstrator
 - Lessons learned, and Impacts on Reference Architectures
- Study conclusions
- Steps forward

Study context / Mass Memories

Evolution of Mass Memories

- From simple storage areas to a central part of the avionics architecture
- Can hold mission-critical items such as
 - Telecommand timelines
 - Software images
 - OBCPs

Study context / Files

Introduction of files into operational concepts

- Significant quantity of data to be transferred between space & ground;
- Data naturally structured as files, or aggregated into files
- Availability of reliable file transfer protocols (e.g. CFDP)
- Likely that usage of files becomes common in the near-future missions

Study context / SAVOIR MASAIS

SAVOIR-MASAIS Working group constituted

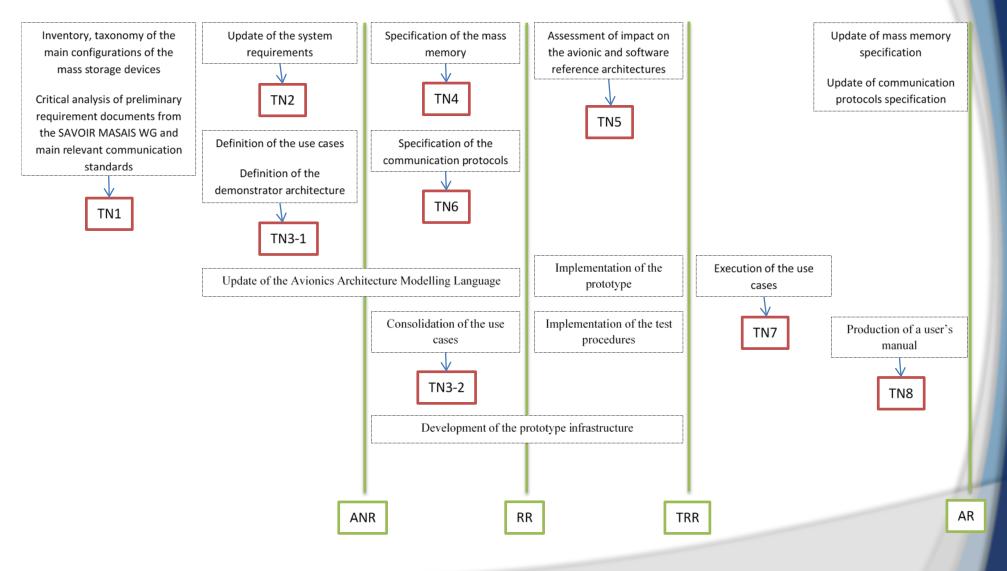
- Stands for SAVOIR Mass Storage Access Interfaces and Services
- Dedicated to the establishment of unified functional and interface requirements for the mass memory function
- FMSIS study in the frame of SAVOIR-MASAIS activities

Study context / Partners

FMSIS consortium

- SPACEBEL:
 - Prime contractor
 - Requirements reviewer
 - Demonstrator
- TAS
 - Mass Memory survey
 - User, Technical and Protocol Specification
 - OSRA impact analysis
- GMV
 - AAML tool update

Study logic



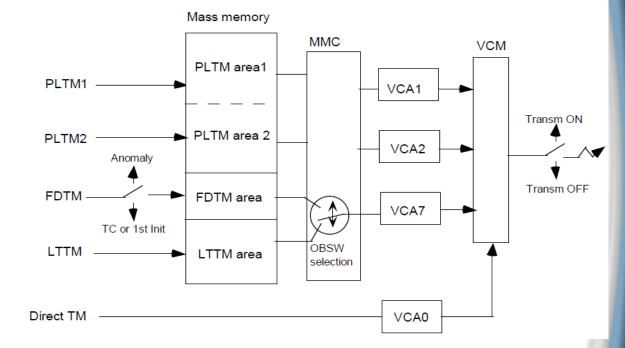
Study results / Survey / Avionics Survey of mass memory architectures on past, present and future missions

- Proteus MK I & II
- Sentinel-3
- Herschel/Planck
- Lisa Pathfinder
- Galileo

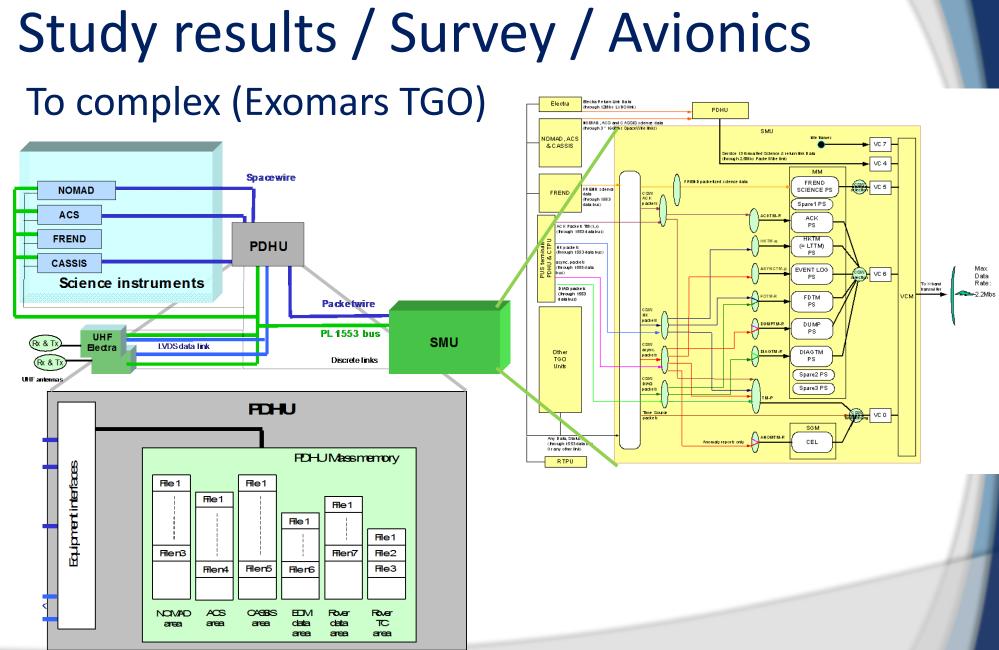
- Exomars
- Gaïa
- EUCLID
- JUICE

Study results / Survey / Avionics From simple (Proteus Mk I MM)

- Only 4 areas
- Delimited data
- Basic pointer management



Final Presentation Days - ESTEC



IV-

© 2012 by SPACEBEL

Study results / Survey / Protocols

Many access protocols used or to be used:

- Packet Stores PUS service 15 widely adopted
- Files:
 - PUS 13 (Large Data Transfer) adapted to files
 - ISIS standardization
 - PUS 23 (File Management) introduced in PUS C
 - SOIS FPSS
 - CFDP
 - Ad-hoc

Study results / Survey / Protocols

Common denominator to all file accessing protocols:

Mandatory functions:

- File creation/deletion function: PUS(144), PUS(23), CFDP, SOIS FPSS
- File attributes management function: PUS(144), PUS(23), CFDP
- File listing function: PUS(144), PUS(23)
- File integrity management function: PUS(144)
- File read/write function: PUS(13), CFDP, SOIS FPSS
- File copy/move function: PUS(23), SOIS FPSS

Optional functions:

- Directory creation/deletion function: PUS(23), CFDP, SOIS FPSS
- Directory attributes management function: PUS(23)
- Directory listing function: PUS(23), CFDP, SOIS FPSS
- File protection management function: PUS(23), SOIS FPSS

Specification / User

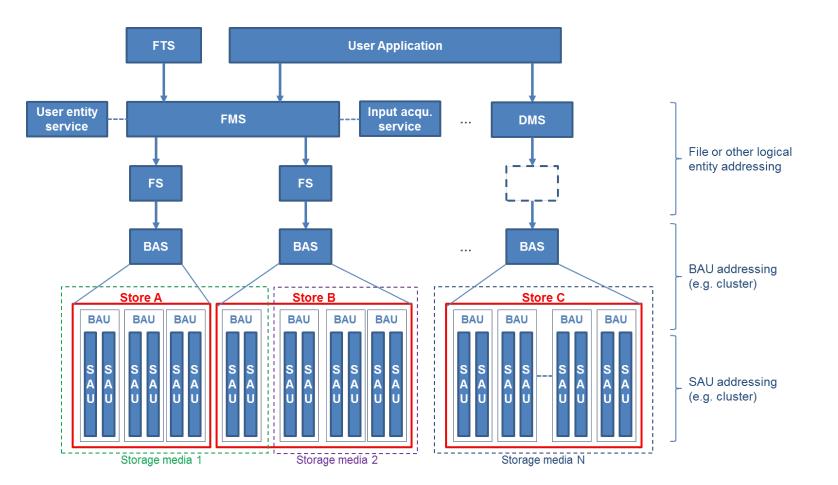
- First output of the study, based on a first iteration provided by the SAVOIR community
- Consolidates needs identified during the MM architecture and protocol survey
- Integrates file and directory awareness at mass memory level
- File abstraction layer has multiple advantages, e.g.
 - On-board autonomy through smart selection of files for communication sessions (according to creation date, data type...)
 - Efficiency of search operations
 - Easier visualisation of available data for operators

Specification / User

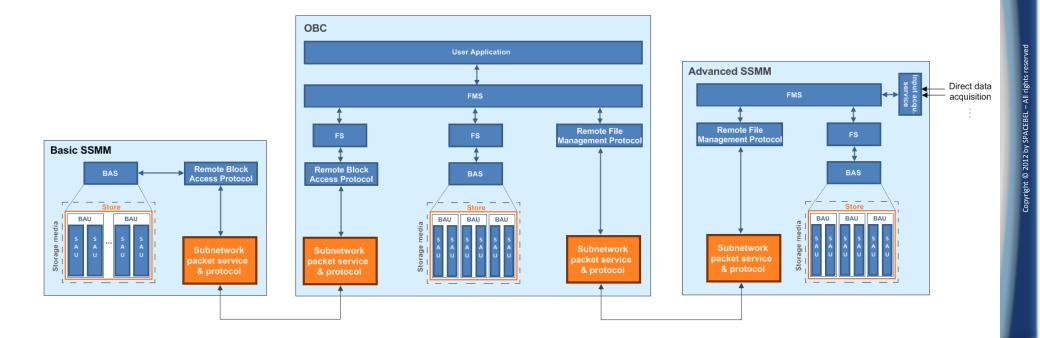
User requirement document addresses several aspects of mass memories:

- Generic system requirements (overall functionality)
- Interface requirements (Enforcing standards for different communication link types)
- File Transfer requirement (CFDP based)
- Data organisation requirements:
 - Concept of Store (one data area managed by a block access system)
 - File Store and Packet Store as specialisation of store to specific data
- Performance
- FDIR

Specification / Data Storage Architecture



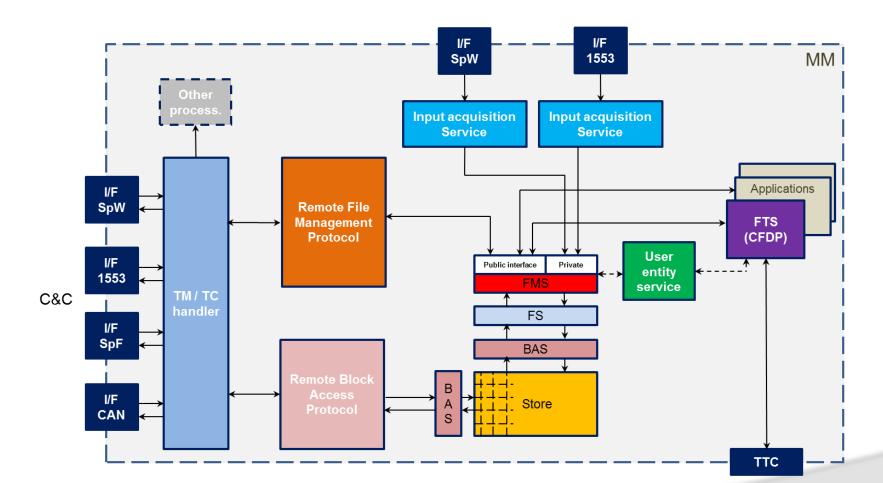
Specification / Deployment example Deployment example used as use case for requirement verification: 3 Mass memories deployed



Specification / FMSIS specification

- Document derived from user specification (after review by SAVOIR-MASAIS)
- Introduces the generic mass memory functional design
- Specifies all file and directory related directives with their parameters.
 - Usual primitive expected in a file system
 - Specific services related to the mass memory operation heritage

Specification / Functional design



Data Acquisition

Specification / Primitives

In addition to your usual file primitives...

- Set the maximum file size at creation
- Define behaviour upon full file
- Configure autonomous data acquisition to files (map)
- Copy & Move between file stores
- Control (suspend, resume, abord, get status) copies
- Manage custom attributes (statuses and metadata)
- Request file synchronisation (buffering issues)
- Reporting of file store events

Specification / Opening Schemes

Access Type	Full File Action	Packet Storage	Behavior
Read- only	N/A	N/A	Pointer at beginning, Read, Seek
Read- write	N/A	N/A	Pointer at beginning, Read, Write, Seek. Excess data at end is discarded
Append	Close	No	Pointer at end. Write. Excess data discarded and file closed
Append	Close & Create Next	No	Pointer at end. Write. File closed on excess data, new file opened to store this data (split).
Append	Close File	Yes	Pointer at end. Write. All data discarded on excess data and file closed
Append	Close & Create Next	Yes	Pointer at end. Write. File closed on excess data, new file opened to store this data (atomic)

Specification / Map

Autonomous acquisition into files:

- Builds upon the append opening options;
- Allows mapping data flowing into data acquisition service according to selection criteria:
 - Exclusive interface mapping
 - Protocol specific address
 - Custom packet fields
 - PUS specifics (APID, type, subtype)

Specification / File events

Notification of events happening in the file system

- Allows maintaining a ground image of the file system
- Completes the autonomous closing and opening of subsequent files
- Three monitored events:
 - File creation
 - File autoclosing
 - File deletion

Specification / Protocol

Last specification output from the FMSIS project

- Defines the Remote File Access and Remote Block Access Protocols
- Directly derives the FMSIS technical specification

FMSIS specification and protocol definition prototyped into a demonstrator to verify implementability

AAML / Overview

The AAML (Avionics Architecture Modelling Language) framework provides a toolchain to advance the avionics engineering practices towards a model-based approach.

Main elements:

- AAML Metamodel.
- Graphical Editor at three different levels: functional, logical and physical.
- A set of analyses to refine/verify the avionics architecture.

Technologies:

- Developed as a set of Eclipse plug-ins that configure a design and analysis environment integrated into the Eclipse platform (Eclipse Luna).
 - \geq The AAML Metamodel is based on *Ecore* and developed using *EMF*. It is inspired in the Space Component Model (SCM) of OSRA.

FMSIS Use Case EMSIS Use Case

> > BusSchedul ComAndObsA Convright 7

OnBoardFunc Z Sentinel3.aa

Sentinel3.air

Avionics Fu & Avionics I

> Model Project Depend > Use Case 7

The Graphical Editor is based on Sirius.

24

Quick Access

Bus type

UART

MIL-STD-1553

MIL-STD-1553

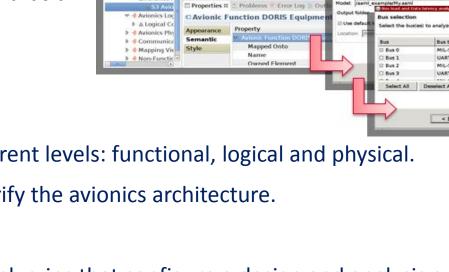
< Back

aaml

All Fine

Cancel

aaml



nics Functional Diagram II

N 16 16.

AAML / FMSIS and AAML toolchain

 In the frame of FMSIS project, new capabilities have been added to the AAML Framework in order to:

- Address <u>FMSIS requirements</u> concerning Mass Memories and Communication Protocols.
- Support the demonstration of <u>FMSIS Use Case</u>.

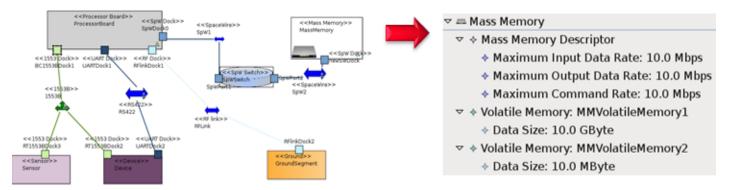
□ These extensions are gathered in three **main groups**:

- 1. Mass Memories.
- 2. SpaceWire.
- 3. Standards/Protocols.

Next slides list the main updates. Their implementations impact the AAML Component Model, the Graphical Editor and the Analysis Tools.

AAML / Extensions / Mass Memories

Definition of **SAVOIR Mass Memories (MM)** and their configuration.



MM organization characteristics: **Data Storage Systems** (DSS) supporting both file and packet systems.

	Kind	Physical Memory	Endiann	Initial Ac	Final Ada	SAU Valu	SAU Uni	SALU Va	SALU	Un			
Z Log. Comp.: Control_Management	t .					-							
	PayloadData	Volatile Memory MMVolatile					Name	Ident	ifier	Directory Max Entrie	e File Type	File/Packet Size V	File/Packet Siz
Memory Area: Area 1				🗖 Log. Cor		ol_Manag	1						
Memory Area: Area2			BIG_EN		: Store1 Organizati	ion							_
	PlatformData	Volatile Memory MMVolatile			irectory	ion	RootDirector	y 8319811	6-2041-42	2			
Memory Area: Area1			BIG_EN	*	File		conf.h	79cb5f6	ib-14d0-4		Standard	10.0	bit
Memory Area: Area2			BIG_EN	÷ ⇒ ♦ Store:	File Store2		readme.txt	317b44a	1-12fa-4		Standard	20.0	bit
				⇒ + PS (Organizati	ion							
				+ Pa	acket		Packet1					30.0	byte
Mass Memory J	Analy	/SIS.		+ Pa	acket		Packet2					50.0	byte

SpaceWire:

- Support of SpaceWire communication links and their configuration:
 - > SpaceWire Link with its maximum throughput and maximum load margin.
 - > SpaceWire Nodes with its core frequency, data rate and logical address.
 - > SpaceWire Routing Switches with its type, maximum speed and physical ports.
- The analyses include the "SpaceWire load" through SpaceWire links, nodes and routing switches.

Standards and Protocols:

- Standards: CCSDS Space Packets, PUS Packets and Native Packets.
- Protocols: CCSDS File Delivery Protocol (CFDP) Class 1 and Class 2.

Connection	HW Communication Media	Standarc	Protocol
Interface Connection conn_Control_Data_Storage_IF_Control_Data_Storage	Space Wire SpW1, Space Wire SpW2	PUS	NONE
Interface Connection conn_Control_Data_Storage_IF_Control_Data_Storage	Space Wire SpW1, Space Wire SpW2	PUS	NONE
Interface Connection conn_Device_IF_Device_IF	MIL STD 1553B 1553B , RS422 RS422	PUS	NONE
Interface Connection conn_Sensor_Data_Acquisition_IF_Sensor_Data_Ac	MIL STD 1553B 1553B	PUS	NONE
Interface Connection conn_Ground_IF_Ground_IF	RFlink RFLink	PUS	CFDP_CLASS_1

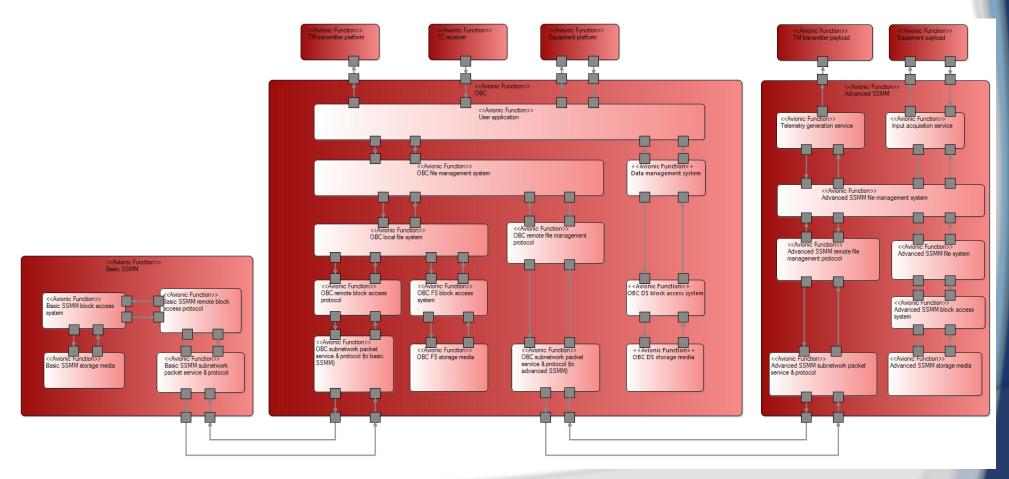
SpaceWire 'SpW1'

Requirements - Mode 'Operational': Maximum Throughput: 300.0 <u>Mbps</u> Maximum Load: 50.0 %

Results - Mode 'Operational': Worst case bus load: 0.5248 % Worst case data throughput: 262400 bps Best case bus load: 0.2624 % Best case data throughput: 131200 bps

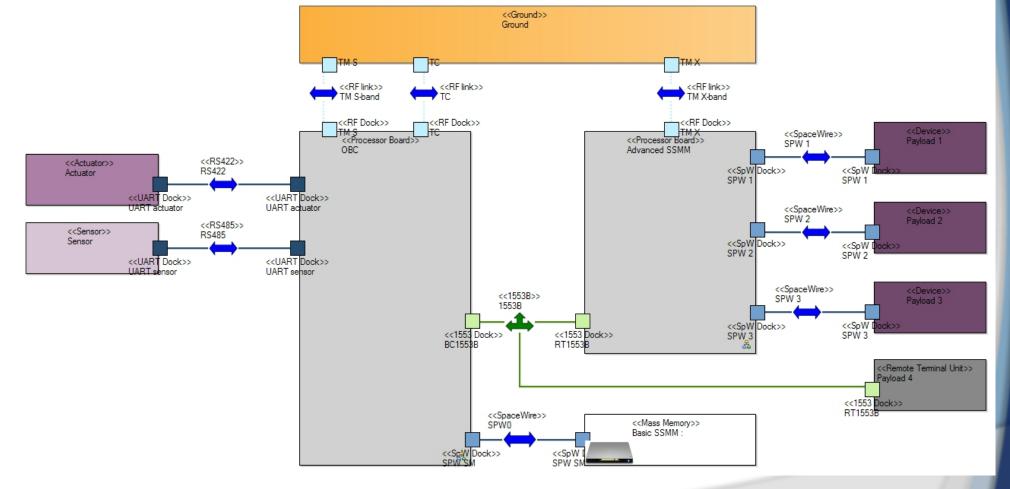
Demonstrator / Use case

Use case storage configuration



Demonstrator / Use case

Communications configuration



Copyright © 2012 by SPACEBEL – All rights reserv

Demonstrator / Scenarios

Setup:

- Basic SSMM
 - Equipment platform data files
 - File management from the OBC
- OBC
 - HKTM packet store, CEL packet store, SW configuration
- Advanced SSMM
 - Payload data files
 - File management on the SSMM

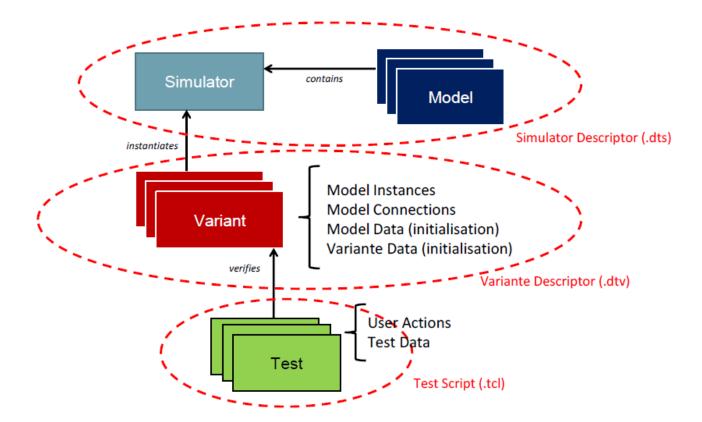
Demonstrator / Scenarios

Scenarios:

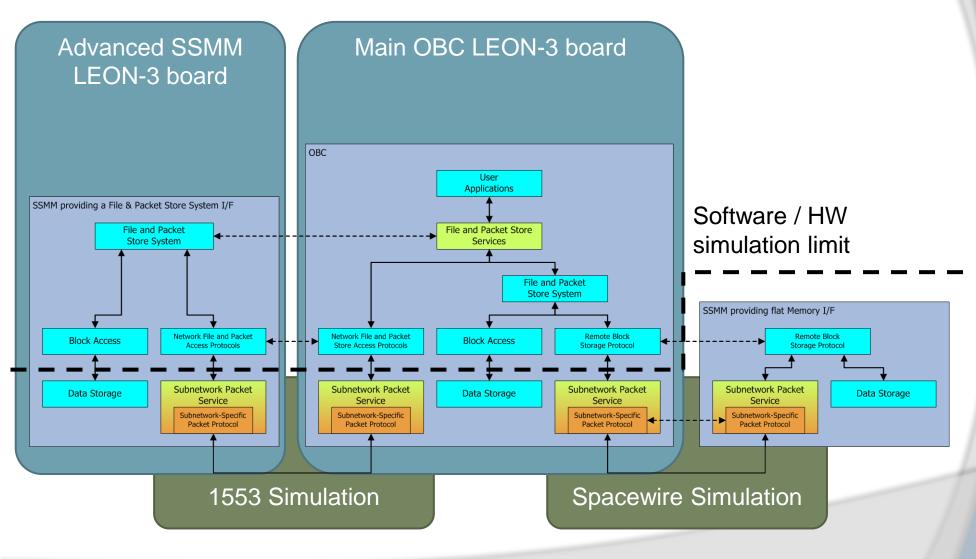
- SC1: Record without visibility
- SC2: Download in visibility
- SC3: Reconfigure recording capabilities
- SC4: Uplink files
- SC5: Manage directories and files
- SC6: Perform maintenance operations
- SC7: Perform FDIR operations

Demonstrator / Architecture

Demonstrator based on the BASILES architecture



Demonstrator / Architecture



Demonstrator / Results OBC telemetry window with CFDP PDU decoding

	ITTED PACKETS NUM: 1				» 🚐
- TOTAL RECEIVE	ED (AND PROCESSED) PA	ACKETS NUM: 62 (ACK PDUs INCLUDED)			👋 🌒 💼 🛛 Mon May 30, 11:29 AM
		~~~~~			
		N [02_01] TERMINATED ! ********			
(+-+ Successfi	ully (No Error) ++ )				
Transaction Duration:	0 Hours : 0 Mins : 6 secs :	661 millisecs **	on tests/scenarios _ 🗆 🗙	🗉 fmsis@vm-fmsis:/projects/fmsis/obsw/tests/validation_tests/logfiles _ 🗆 🗴	
	ITTED PACKETS NUM: 1		-	Eile Edit View Search Terminal Help	
- TOTAL RECEIVE	ED (AND PROCESSED) PA	ACKETS NUM: 62 (ACK PDUs INCLUDED)	-	TM 1 7. packet header. packet sequence control. sequence flags -> 3	
				TM_1_7.packet_header.packet_sequence_control.sequence_count -> 63 TM_1_7.packet_header.packet_length -> 15	
** <receiving> Tran</receiving>	sactions Scheduler Thread	SUSPENDED (OPTIMIZATION)		TM_1_7.data_field_header.spare_1 -> 0 TM_1_7.data_field_header.packet_PUS_version_number -> 1	
		NSACTIONS RUNNING) ###		TM 1 7.data field header.spare 2 -> 0 TM 1 7.data field header.service type -> 1	
			-	TM 1 7.data field header.service subtype -> 7 TM 1 7.data field header.cuc time.sec -> 132	
		+	tcl	TM 1 7.data_field_header.cuc_time.frac -> 12079360 TM 1 7.application data.packet id.version number -> 0	
RECEIVING TRA	ANSACTIONS	SENDING TRANSACTIONS		<pre>TM 1 7.application data.packet id.type -&gt; 1 TM 1 7.application data.packet id.data field header flag -&gt; 1</pre>	
	ction[02_01]- Receix +		tcl	TH 17.application data.packet id.application process id -> 10 TH 17.application data.packet sequence control.sequence flags -> 3	
:29:56 139 -> -Transac :29:56 139 -> - Transac	ction[02_01]- Recei		tcl	TM 17.application data.packet_sequence_control.sequence_count -> 20 TM 17.application data.packet_sequence_control.sequence_count -> 20	
			.tcl .tcl	SVS Tool: AUTO-OK: TM reception (type 1,7) within 100 seconds: '\$tm received' is '"true"' as expected	
-Transaction[02_01]- (Checksum: 46F	F0FAE9)		.tcl .tcl	OK: TM reception (type 1,7) within 100 seconds: Expression '\$tm_received' is true SVS Tool: AUTO-OK: Reception of completion report success for TC (APID 10 SSC 20) within 100 seconds: '\$tm_rece	
:29:58 173 -> -Transac	tion[02_01]- Sent P		t.tcl .tcl	<pre>ived' is ""true"' as expected OK: Reception of completion report success for TC (APID 10 SSC 20) within 100 seconds: Expression '\$tm_received</pre>	
				' is true MSG  GEI :	
:29:58 773 -> -Transac	ction[02_01]- Receiv		alyse test logs."	MSG [GEI : ACK PDU for transaction spacecraft 01 MSG [GEI : 12[bytes], ack, CRC, from ground to spacecraft	
29:58 773 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>				MSG GEI : ACK for the EOF PDU (no error) MSG GEI : Transaction status : active	
( +-+ Succes	sfully (No Error) +-+			MSG [GEI :	
				MSG [GEI : Finished PDU for transaction spacecraft_01 MSG [GEI : 9[bytes], ack, no CRC, from ground to spacecraft	
	-		-	MSG [GEI : no error   generated by end system   data are complete MSG [GEI : Delivered file retained in filestore successfully	
	► <	Þ	_	MSG [GEI :	
s Generated 4	Buff Out 0	Released 4 Received 124		PM A: Completion report for operation 0x403cc908 forwarded to app	
	FMSIS-5.tcl:load /proje	ects/fmsis/ground_cfdp/GEI/GEIF.so ects/fmsis/ground_cfdp/GEI/GEIF.so		PM A: Error code: 0 PM A: Data size: 0 PM A: 1111	
	REMOTE-A.tcl:#load /use	oad /users/ocn/fmsis/ground_cfdp/GEI/GEIF.so ers/ocn/fmsis/ground_cfdp/GEI/GEIF.so		PM A: *****	
	[fmsis@vm-fmsis scenar:	ers/ocn/fmsis/ground_cfdp/GEI/GEIF.so ios]\$ export TEST_CASE=2;rm \$S_MISSION/RES/*1	ock;SIMU -tfB execute_sbtb_test.tcl	PM A: PI - FINISHED PM A: Source entity: 2	
	S_MISSION : /data/proje	ans les variables d'environnement. ects/fmsis/svf/FMSIS-0.9/simulators/S_MISSION		PM A: Sequence number: 1 PM A: Filestore responses ptr: 0x00000000	
	-> Execution de simu_c Exécution sans CDT : ru	lient.tclrcFin un.sh (pid=4925)		PM A: Condition code: 0 PM A: Delivered file retained successfully	
	Binaire FMSIS_o trouvé			PM A: Data is complete PM A: *****	
	Connexion en cours PID	=4930 OK		PM A: Sending TM(25,18) of 43 bytes Comm: Received TM via TTM Stub: 0805c00a002410191200000084b999000200000001000000000000000000000000	
	[fmsis@vm-fmsis scenar: S MISSION est trouv@ da	ios]\$ export TEST_CASE=2;rm \$S_MISSION/RES/*1 ans les variables d'environnement.	ock;SIMU -tfB execute_sbtb_test.tcl	0003e50 No format for service type 25 service subtype 18	
	S_MISSION : /data/proje -> Execution de simu c	ects/fmsis/svf/FMSIS-0.9/simulators/S MISSION		MSG  GEI :	
	Exécution sans CDT : re			MSG [GEI : 12[bytes], ack, GRC, from spacecraft to ground MSG [GEI : aCK for the Finished PDU (no error)	
	Binaire FMSIS_o trouvé	avec le PID=5129 !		MSG [GEI : Transaction status : active	
	Connexion en cours PID	=5129 UK		MSG  GEI :	



### **Demonstrator / Results** ESTEC's CFDP ground entity integrated in simulation and verification environment 23 /cygdrive/i/WIN_server cn@l-pcocn /cygdrive/i 20151206_153534_DSC_3050.JPG file1copy logging tests Leon3NL.zip small_file.txt WIN_server CFDP_Temp ndications To User Traffic Directory Listing Running Transaction Info Test cn@l-pcocn /cvadrive/i UPLOAD cd WIN_server/ Dest I... Next Ho... Size Status Progre... Bytes S., Duration Rate cn@l-pcocn /cygdrive/i/WIN_server ./WIN_server.exe WIN_server: --WIN_server: Killing the ground CFDP entity ERROR: The process "CFDP_SVF.exe" not found. |WIN_server: Wait for 5 seconds to kill the ground CFDP entity. Please wait 1 dir(s) moved. 15G WIN_server: Waiting for a connection from SVF... 15G SRV10091TC: Server is now waiting for a connection... 1SG WIN_server: Waiting for messages from SVF... WIN_server: starting new entity WIN_server: result of command <cmd /C "I:\CFDP_Temp\CFDP_SVF.exe" & DOWNLOAD Show Image: |WIN_server: Waiting for messages from SVF... Size Dest I... Next Ho... Status Progre.. Bytes R. Duration Bate 30000 02 02 00 3:/pltf_eqpt_data I:\tests\pltf_eqpt_d... 01 Termin 100 25088 0H:0M:3s 66 27 K 02 01 3:/pltf egpt data.00 I:\tests\pltf egpt d... 30000 01 02 Termin... 100 24576 0H:0M:2s 98.60 K. 11:30:00 RCV Transaction[02 01] Terminated ! [ACKNOWLEDGED TRANSMISSION] SND Transactions: 0 Receiving Transactions: 0 6/9/2016 Final Presentation Days - ESTEC 35

### Demonstrator / Results

Requirement prototyping activity results:

- Successful execution of scenarios
- Feedback to specification
  - Necessity of an offset parameter in read & write requests (performance over time slotted buses like 1553)
  - Missing protocol messages related to the continuous storage of the « continuous storage » feature, and removal of pointer management implementation constraint (effect on other file descriptors)

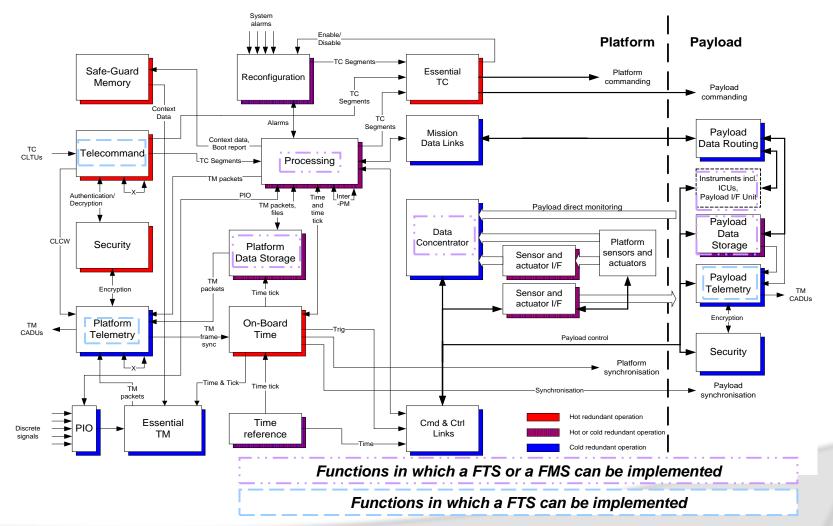
### Demonstrator / Results

Requirement prototyping activity results:

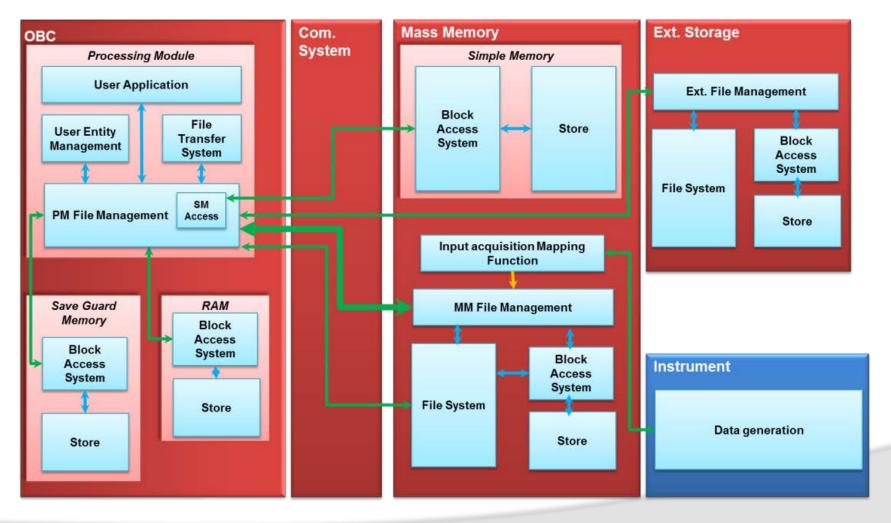
- Feedback to operation / TMTC standards
  - Expose file mapping capabilities to PUS
  - Extend the concept of S23 file attributes
    - File Statuses (Inherent to FS: size, lock status, ...)
    - File Metadata (User-Modifiable: Downlink priority, datatype...)
- First attempt at Service 15 (packet storage) implementation using files

# Lessons learned and ASRA impacts

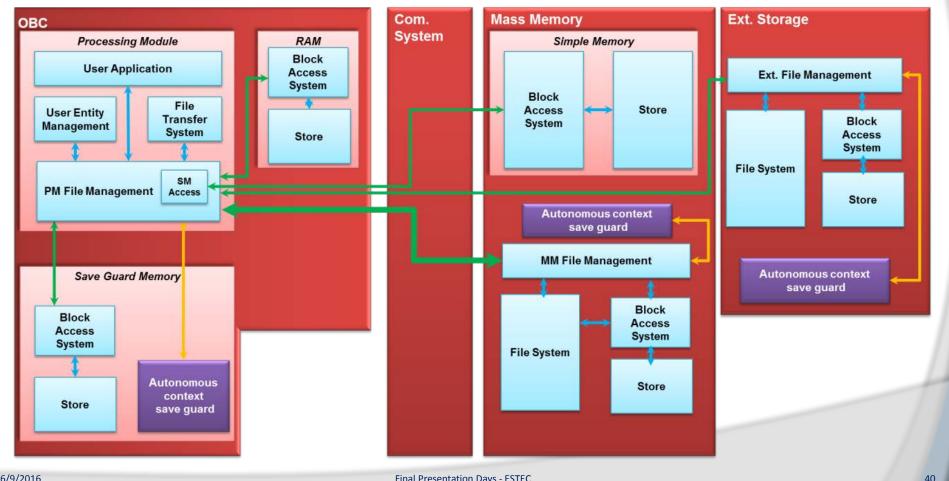
Possible insertions of FMS & FTS functions in ASRA:



### Lessons learned and ASRA impacts Interaction between OBC FMS and storage devices:

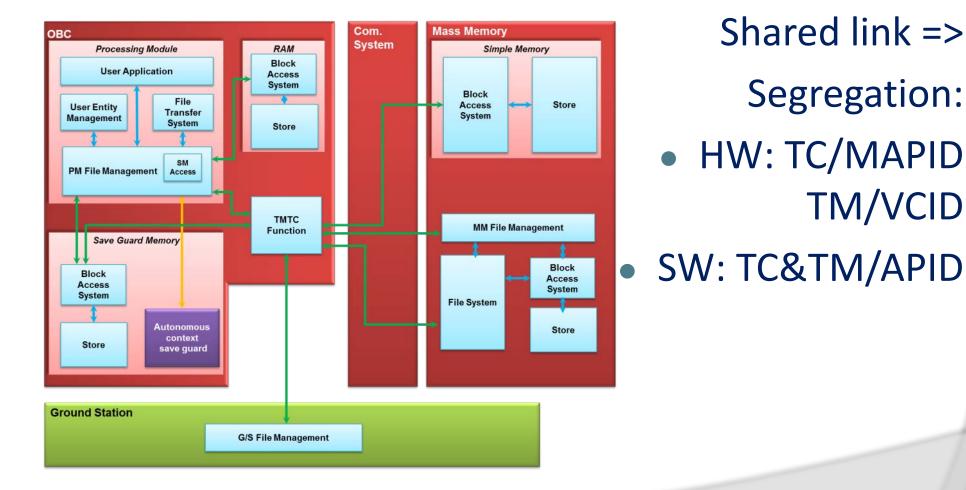


### Lessons learned and ASRA impacts Autonomous storage for memory organisation (store creation, BAU / SAU allocation...)

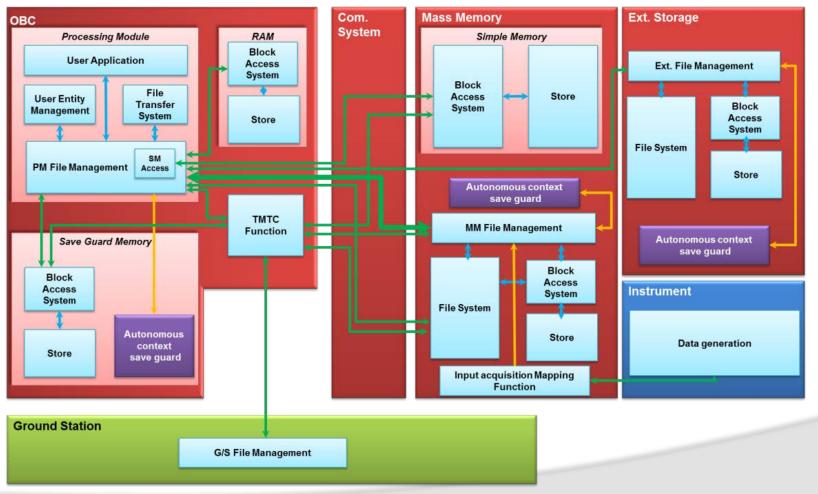


**Final Presentation Days - ESTEC** 

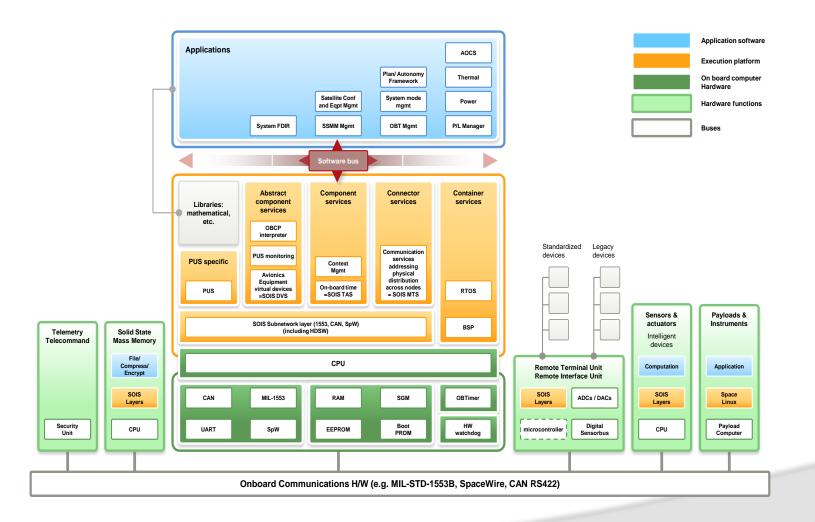
### Lessons learned and ASRA impacts Ground – board communication







### **OSRA** impacts for SW FMS/FTS



### Conclusions

- Generic user, technical, and protocol requirements produced for mass memories, integrating file awareness;
- Implementability of these requirements demonstrated thanks to a flexible demonstrator based on a software simulator;
- Impacts of introducing the file transfer and file management services into the ASRA & OSRA examined

### With the FMSIS study completed, several tasks are

**Steps forward** 

still at hand:

- Review by SAVOIR-MASAIS
- Implementation of comments and transfer to a SAVOIR document
- Review by SAVOIR Advisory Group
- Public review

### Reaching into space TOGETHER

### **QUESTIONS ?**

Thank you for your attention



