MILS Compliant Software Architecture for Satellites

ESA Contract No. 4000108471/13/NL/LvH
MILS Workshop, Prague, 19.01.2016

Is there a security issue in space applications?

- Yes, demonstrated in several James Bond movies (Golden Eye, …)
Solutions today

Satellite

- Satellite Attitude & Orbit Control
- Payload A
- Payload B

Uplink (with encryption)  Downlink (with encryption)

Encrypted data streams

Ground System

Uplink (with encryption)  Downlink (with encryption)

Separation of DataStreams

No encryption

Spacecraft Operator

User A  User B

No encryption
Security oriented architecture

Satellite

- Satellite Attitude & Orbit Control
- Payload A
- Payload B

Ground System

- VC
- VC
- VC
- VC

End-to-End Encrypted data streams

Bypass for safety reasons

Spacecraft Operator

User A

User B
The Starting Point

- **Security**
  - Encryption on TM/TC link (hardware)
  - Access control implemented on ground

- **IMA-SP Study**
  - Supports the principle of „separation of concerns“ through Time & Space Partitioning (TSP)
  - Focus on development flow and scheduling („safety aspects“)

- **Combining IMA/TSP approach with security features**
  - TSP guarantees non-interference, resilience against malicious actions (safety aspect)
  - TSP ensures integrity, availability and confidentiality of data within each partition (security aspect)
  - Additional components are needed to ensure secure communication between partitions

**Software Elements for Security – Partition Communication Controller**
Use Case „Earth Observation“

Administrator
Key Generation
& Distribution
Facility

Keys

SPCC based
Architecture

End-To_end Encrypt.

Payload A
(hardware)

MMU

Payload Security
(hardware)

Payload A

STC

STM

TM/TC

Security

S-Band Transponder

X-Band Transponder

STM

XTM

Payload Data Ground
Cryptic Security

X-Band Modem

Payload Data Ground
Cryptic Security

X-Band Transponder

X-Band Transponder

User Facility I

Control Facility

User Facility N

TT&C Control
Workstation

Satellite Operator Ground
Cryptic Security

S-Band Modem

SAR Data Processing
and Control Workstation

SAR Data Processing
and Control Workstation

Payload Data Ground
Cryptic Security

X-Band Modem and
record and playback

X-Band Modem and
record and playback

Payload Data Ground
Cryptic Security

Keys

SPCC – Software Elements for Security – Partition Communication Controller

7. December 2015
Use Case „Telecom“

State-of-the-Art Architecture

SPCC based Architecture

SPCC – Software Elements for Security – Partition Communication Controller
7. December 2015
# Threat Assessment - Methodology

1. **Identify relevant threats** based on generic list provided by EBIOS v2 Section 4, Tools for Assessing ISS Risks FIPS PUB 197, Nov 26 2001.

<table>
<thead>
<tr>
<th>EBIOS Ref</th>
<th>Generic Threats</th>
<th>Selected Threat</th>
<th>Ref</th>
<th>Specific Threat</th>
<th>Assumptions</th>
<th>Additional Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>Interception of compromising interfering signals</td>
<td>No</td>
<td>17.1</td>
<td>Electrical signal interception during integration and test</td>
<td>Appropriate facilities are provided during test and integration, to limit the potential for effective signal interception</td>
<td>This is only relevant to flight equipment before launch, for conventional reasons, but the threat is countered by the environment and not specific technical means, so is not considered here.</td>
</tr>
<tr>
<td>19</td>
<td>Remote eavesdropping</td>
<td>Yes</td>
<td>19.1</td>
<td>Eavesdropping between a Platform Operator and Platform Subsystem</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>19.2</td>
<td>Eavesdropping between a Platform Operator and Platform Subsystem</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>19.3</td>
<td>Eavesdropping between a Payload and a Mission Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Theft of media or documents</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Theft of equipment</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Retrieval of records or damaged media</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Disclosure</td>
<td>Yes</td>
<td>23.1</td>
<td>Disclosure</td>
<td>Organisational policies provide for appropriate protection and destruction of media and documents</td>
<td>Information disclosed to an external party i.e. through the RF link</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>23.2</td>
<td>Disclosure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>23.3</td>
<td>Disclosure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Data from untrustworthy sources</td>
<td>Yes</td>
<td>24.4</td>
<td>Data from untrustworthy sources</td>
<td>Information is used without guarantee of origin by the Platform</td>
<td>This considers the potential for modification or insertion of TC on the RF link, including replay of authorised commands.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24.5</td>
<td>Data from untrustworthy sources</td>
<td>Information is used without guarantee of origin by the Platform</td>
<td></td>
</tr>
</tbody>
</table>

**EBIOS=Etude des Besoins et Identification des Objectifs de Sécurité**

**SPCC – Software Elements for Security – Partition Communication Controller**

7. December 2015
Relevant Threats

- Non-authorized User (33.6)
- Authorized User (40.1)

Encryption and Decryption:
- User A
- User Crypto Unit
- Payload A
- Payload B

Routing and Modification:
- Direct Access & Redirecting data (36.4)
- Modify routing (32.1)
- RF Jamming (14)

Authorized User (~40.1):
- Direct access and redirecting data
- Modify routing

Non-authorized User (~33.6):
- Direct access and redirecting data
- Modify routing

Encryption and Decryption:
- Data
- Hdr A
- Hdr
- Payload

Routing and Modification:
- Decryption
- Routing
2. Identify **Target of Evaluation (TOE)**: indicate the boundary and the contents of the security equipment analysed and evaluated.

**TOE identification:** SPCC + Cryptographic Component + Separation Kernel

**Assurance Level:** EAL4.

**TOE Boundary:** The TOE lies within the Spacecraft computer, and specifically comprises:

- The SPCC (software) which performs the security functions
- The separation kernel (software) which prevents the SPCC being bypassed
- Any hardware support in the onboard computer processor (memory management unit), required to ensure that only the SPCC can access particular I/O.
3. Map identified **specific threats** to a set of **Security Objectives** which will prevent the threat from occurring

**O.ADMIN** The TOE must provide functionality which enables an authorised user to effectively manage the TOE and its security functions, and must ensure that only authorised users are able to access such functionality, while also maintaining confidentiality of sensitive management data.

**O.AUDIT** The TOE must provide a means of recording any security relevant events, so as to assist an authorised user in the detection of potential attacks or misconfiguration of the TOE security features that would leave the TOE susceptible to attack, and also to hold users accountable for any actions that they perform that are relevant to security.

**O.ENCRYPT** The TOE must provide the means of protecting the confidentiality of information transmitted across the communications link.

**O.ROLES** The TOE must prevent users from gaining access to, and performing operations on its resources for which their roles is not explicitly authorised.

**O.INITIALKEYS** The TOE must provide a means to manually load a full set of Red keys before launch, ensuring both the integrity and confidentiality of those keys.

**O.OTAR** The TOE must provide means for receiving new keys throughout the operation of the TOE, whilst maintaining the confidentiality and integrity of those keys.

**O.INTEGRITY** The TOE must provide a means of detecting loss of integrity affecting information received by the TOE.

**O.REPLAY** The TOE must provide a means to prevent undetected replay of previous information sent to the TOE.

**O.PROTECT** The TOE must protect itself against external interference or tampering by untrusted subjects, or attempts to bypass the TOE security functions.

**O.FAILEDSAFE** In the event of an error occurring, the TOE must preserve a secure state.

**O.SIDECHANNEL** Authorised user(s) of the TOE, e.g. Operator and Platform Software developer, are not cleared to view Key Material within the TOE (defined in O.ROLES). Therefore, the TOE prevent any key information from leaking from the TOE via a side-channel to another software partition. Examples of side-channels are – cache-timing, cache-contents, power-analysis, differential fault analysis. This is addressed in document [TN02] Internal Security Threat Assessment.
2. Map a set of **Security Requirements** for implementation on the **Target of Evaluation** which will meet those **Security Objectives**

<table>
<thead>
<tr>
<th>Objective</th>
<th>Objective Description</th>
<th>Security Functional Requirements</th>
<th>Requirement Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>ENCRYPT</td>
<td>Provide the means of protecting the confidentiality of information transmitted across the communications link from TOE to Hosted Payload Operator</td>
<td>FCS_CRM 3.1.1 FCS_CRM 3.1.2 PPT_ITC.1.1</td>
</tr>
<tr>
<td>O</td>
<td>PARTITION</td>
<td>Ensure that software partitions co-located on the same processor shall have no visibility of the data of another, except via inter-partition communication</td>
<td>FDP_ACF 1.1.1</td>
</tr>
<tr>
<td>O</td>
<td>HAPARTITION</td>
<td>The Hardware interfaces to the O/Q must be consistent with O.PARTITION, and ensure that the physical partitioning of the data between interfaces / domains / end-users, is maintained, and that no data is passed between interfaces / domains / end-users (that function remains the sole responsibility of the SPCC)</td>
<td>FPT_FLS.1 FPT_HHP.1 FPT_RVM.1 FPT_SEP.1</td>
</tr>
<tr>
<td>O</td>
<td>SIDECHANNEL</td>
<td>Prevent any information from leaking from the security domain of one software partition, to another security domain that may be a software partition or to a physical interface. Specifically, information concerning application timing, from the source application, and impact of confidential</td>
<td>FPT_ITC.1.1.1</td>
</tr>
</tbody>
</table>
TASK 2: Requirements and Architecture

Initial Set of Functional Requirements

- SPCC 6.2 The SPCC shall support the telecommand to upload a new Key
- SPCC 6.3 The SPCC shall unwrap a new uploaded Key, using the appropriate Key Encryption Key stored locally and
  AES cryptographic algorithm – CFB mode, 128-bit key (FIPS 197,FIPS 140-2)
- SPCC 6.4 The SPCC shall authenticate the uploaded Keys before use, using the wrapping integrity-checks. If the
  unwrapped Key fails wrapping integrity-check, the unwrapped Key shall be discarded
- SPCC 6.5 At boot, the SPCC shall use the root key key loaded to EEPROM as its Key Encryption Key
- SPCC 6.6 The SPCC shall report key labels of all Keys currently loaded, in telemetry, but not any key material
- SPCC 6.7 The SPCC shall report any status of the SPCC in telemetry, including:
  - Key unwrapping and validity status
  - Number of authentic / inauthentic telecommand packets received
  - Number of replayed / non-replayed telecommand packets received
  - Current value of Local Authentication Count reply counter
- SPCC 6.8 All SPCC telemetry shall be encrypted and transmitted to S-band transponder interface handler in an
  identical manner to spacecraft platform telemetry
- …

- SEP.1 The Separation Kernel shall control the access of each software partition to defined areas of memory. A
  software partition shall not be allowed to access an area of memory (read or write) unless permitted by the Separation
  Kernel
- SEP.2 The Separation Kernel shall control the execution timing of each software partition, such that it executes at a
  time completely independent of the activities of any other software partition
- SEP.3 The Separation Kernel shall control the access of each software partition to I/O’s, both read and write.
  Specifically, from a security perspective:
  - SEP 3.1 The I/O corresponding to S-band transponder Rx and Tx shall be accessible only to the S-band Rx and Tx
    interface handlers. These two interface handlers may or may not share a partition
  - SEP 3.2 The I/O corresponding to X-band transponder Tx shall be accessible only to the X-band Tx interface handler
  - SEP 3.3 The I/O corresponding to Mass Memory shall be accessible only to the Mass Memory interface handler
  - SEP 3.4 The I/O corresponding to Data Handling Bus shall be accessible only to the Data Handling Bus interface
    handler
- …
Multiple Independent Levels of Safety and Security (MILS) compliant architecture: is a high-assurance security architecture based on the concepts of separation and controlled information flow; implemented by separation mechanisms that support both untrusted and trustworthy components; ensuring that the total security solution is non-bypassable, evaluatable, always invoked and tamperproof.
Basic Execution Platform – Design Decisions

- **Processor Module / IO Module:**
  - OBC: Leon2/3 (MDPA, SCOC3) vs. **Leon4**
  - IOM: Leon2 (MDPA) vs. **same as for OBC**

- **Separation Kernel:**
  - OBC: xTratum vs. **PikeOS**
  - IOM: RTEMS vs. **same as for OBC**

- **Basic Software**
  - OBC: CDHS vs. **KARS**
  - IOM: CDHS stripped down vs. **same as for OBC**
KARS – Controller for Autonomous Spacecraft (developed under DLR contract)

- System Partition (Part. #0)
  - Basic System Support Services
  - Mission Specific Extensions
  - Component Support Services
  - Operating System Abstraction Layer

- Partition #1
  - Application Component (e.g. AOCS)
    - Component Support Services
    - OSAL

Implements PUS services (1..19, 140, …)

Provides an API (mainly ARINC 653 compliant):
  - Tasking, communication, …
KARS System Support Services

**Basic System Support Services**

- **PUS Service 3, 11** Mission Time Line Handler (MTH)
- **PUS Service 151** Orbit Position Scheduler (OPSH)
- **PUS Service 148** Telecommand Sequencer (MACRO)
- **PUS Service 18** On-board Control Procedure Handler (OBCPH)

**Mission Specific Extensions**

- **PUS Service 4, 12, 140** Data Management (DM)
- **PUS Service 5, 19** Event Handler (EVH)
- **PUS Service 14, 15** Logging Handler (LOH)
- **PUS Service 6, 8, 9** Supervisor (SUV)

**Selected for SPCC**

**KARS System Support Services**

- **I/O Handler (IOH)**
  - Ethernet (UDP)
  - UART
  - Spacewire
  - MIL1553
TASK 3: Design and Implementation
Software Components

- **Toolset**
  - Toolset
  - **Python (SYSGO)**
    - Routing table compiler (spcc-rtc)
    - Routing table validation (spcc-validate)

- **SPCC Components**
  - **C using the KARS API**
    - Secure Partitioning Communication Controller (SPCC-R)
    - En-/Decryption Component (SPCC-E)
    - Content Checking Component (SPCC-CC)
    - Equipment Handler (SpW, MIL1553)
    - Sample Applications

- **IOM Components**
  - **Instance of the SPCC**
    - Input/Output Router (IOR)
Overall System Architecture

Partition 0
- System Support Systems (Ctrl. & Data Hdl., FDIR, System,..)
- APP A
- APP B
- SpW Equipm. Handler
- SPCC-CC
- SPCC-E

Partition
- SPCC-R

Operating System Abstraction Layer (OSAL)
- Separation Kernel (PikeOS)
- Processor Module (GR-CPCI-LEON4-N2X)
- SpaceWire

CODEO
(PikeOS Integrated Development Environment)

Partition 0
- System Support Systems (subset)

Partition
- IOR
- SpW Equipm. Handler

Partition
- MIL1553 Equipm. Handler

Operating System Abstraction Layer (OSAL)
- Separation Kernel (PikeOS)
- IO Module (GR-CPCI-LEON4-N2X)
- SpaceWire
- MIL1553

Assigning applications to partitions
**Overall Communication Channels**

![Diagram of communication channels between OBC, IOM, and TOOLSET](image)

- **OBC**
  - Files: `cfg/SPCC.static.rt`, `cfg/SPCC.initial.rt`, `cfg/SPCCE.key1`, `cfg/SPCCE.key2`
  - Services: System Support, IOH, Sample KARS Application

- **IOM**
  - Files: `cfg/IOM.static.rt`, `cfg/IOM.initial.rt`
  - Services: System Support, IOH

- **SPCC**
  - Files: `cfg/SPCC.static.rt`, `cfg/SPCC.initial.rt`, `cfg/SPCCE.key1`, `cfg/SPCCE.key2`
  - Services: System Support, IOH, Sample KARS Application

- **CODEO**
  - PikeOS Integrated Development Environment

---

**Defining communication channels**

- **TOOLSET**
  - System Support, IOH, Sample PUS Application

---

**CODEO**

(PikeOS Integrated Development Environment)
### Packet Routing - Definition

<table>
<thead>
<tr>
<th>Source PRID</th>
<th>Target PRID</th>
<th>PUS Service</th>
<th>Source Port</th>
<th>Target Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS</td>
<td>MP2</td>
<td>(140,*)</td>
<td>GS</td>
<td>SPCC-E</td>
</tr>
<tr>
<td>GS</td>
<td>MP2</td>
<td>(140,1)</td>
<td>SPCC-E</td>
<td>SPW_IOM</td>
</tr>
<tr>
<td>SP1</td>
<td>GS</td>
<td>(*)</td>
<td>SPCC-E</td>
<td>GS</td>
</tr>
<tr>
<td>SP1</td>
<td>GS</td>
<td>(140,1)</td>
<td>SPW_IOM</td>
<td>SPCC-E</td>
</tr>
<tr>
<td>GS</td>
<td>MP2</td>
<td>(140,*)</td>
<td>SPW_SPCC</td>
<td>MIL1553_BC</td>
</tr>
<tr>
<td>SP1</td>
<td>GS</td>
<td>(*)</td>
<td>SPW_PL</td>
<td>SPW_SPCC</td>
</tr>
</tbody>
</table>

Routing Table

![](https://example.com/routing_table_diagram.png)

Integration Project

Target machine
Implementation results

- SLOC: ~ 6000
- Unit test cases: 148 (VectorCAST)
- Statement coverage:
  - SPCC-R: 93%
  - SPCC-CC: 100%
  - SPCC-E: 100%
- Decision coverage:
  - SPCC-R: 94%
  - SPCC-E: 97%
  - SPCC-CC: 100%
**TASK 4/5 Validation and Demonstration**

Which TRL and EAL has been achieved?

<table>
<thead>
<tr>
<th>Family</th>
<th>Identifier</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Management</td>
<td>ACM_AUT.1</td>
<td>Partial CM automation</td>
</tr>
<tr>
<td></td>
<td>ACM_CAP.4</td>
<td>Generation support and acceptance procedure</td>
</tr>
<tr>
<td></td>
<td>ACM_SCP.2</td>
<td>Problem tracking CM coverage</td>
</tr>
<tr>
<td>Delivery and Operation</td>
<td>ADO_DEL2</td>
<td>Detection of modification</td>
</tr>
<tr>
<td></td>
<td>ADO_JOGS.1</td>
<td>Installation, generation, and start-up procedures</td>
</tr>
<tr>
<td>Development</td>
<td>ADV_FSF.2</td>
<td>Fully defined external interfaces</td>
</tr>
<tr>
<td></td>
<td>ADV_HLD.2</td>
<td>Security enforcing high level design</td>
</tr>
<tr>
<td></td>
<td>ADV_IMP.1</td>
<td>Subset of the implementation of the TSF</td>
</tr>
<tr>
<td></td>
<td>ADV_LLD.1</td>
<td>Descriptive low-level design</td>
</tr>
<tr>
<td></td>
<td>ADV_RCR.1</td>
<td>Informal correspondence demonstration</td>
</tr>
<tr>
<td></td>
<td>ADV_SPM.1</td>
<td>Informal TOE security policy model</td>
</tr>
<tr>
<td>Guidance Documents</td>
<td>AGD_ADMIN.1</td>
<td>Administrator guidance</td>
</tr>
<tr>
<td></td>
<td>AGD_USR.1</td>
<td>User guidance</td>
</tr>
<tr>
<td>Life Cycle Support</td>
<td>ALC_DVS.1</td>
<td>Identification of security measures</td>
</tr>
<tr>
<td></td>
<td>ALC_LCD.1</td>
<td>Standardised life-cycle model</td>
</tr>
<tr>
<td></td>
<td>ALC_TAT.1</td>
<td>Compliance with implementation standards</td>
</tr>
<tr>
<td>TESTS</td>
<td>ATE_COV.2</td>
<td>Analysis of coverage</td>
</tr>
<tr>
<td></td>
<td>ATE_DPT.1</td>
<td>Testing low level design</td>
</tr>
<tr>
<td></td>
<td>ATE_FUN.1</td>
<td>Functional testing</td>
</tr>
<tr>
<td></td>
<td>ATE_IND.2</td>
<td>Independent testing — sample</td>
</tr>
<tr>
<td>Vulnerability Assessment</td>
<td>AVA_MBIO.2</td>
<td>Validation of analysis</td>
</tr>
<tr>
<td></td>
<td>AVM_SOF.1</td>
<td>Strength of TOE security function evaluation</td>
</tr>
<tr>
<td></td>
<td>AVM_VLA.2</td>
<td>Independent vulnerability analysis</td>
</tr>
</tbody>
</table>

EAL1: Functionally Tested  
EAL2: Structurally Tested  
EAL3: Methodically Tested and Checked  
EAL4: Methodically Designed, Tested, and Reviewed  
**EAL5: Semi-formally Designed and Tested**  
EAL6: Semi-formally Verified Design and Tested  
EAL7: Formally Verified Design and Tested
Validation Results

- Validation platform:
  - Leon4 with simulated equipments connected via
  - SpaceWire and
  - MIL1553.
Message Routing Example

SPCC – Software Elements for Security – Partition Communication Controller

7. December 2015
Time measurements and data rate

- **Required:**
  1 OBC, 4 Payloads, 10 TC/s and 20TM/s per element, max. packet size: \(1024 = 30 \times 5 \times 1024\) Byte/s = **1229kiB/s**

- **Achieved:**
  - IO overhead: **0.12ms** (time of execution of void IO call),
  - Message transfer time between applic. via SPCC: **1.5ms**
  - Data transfer between two SpW ports via the PikeOS SpW driver: **0.4ms** … **0.43ms** (depending on packet size)
  - maximum data transfer rate: **2340kiB/s**.
Validation Results

- Aim at
  - EAL1: Functionally Tested
  - EAL2: Structurally Tested
  - EAL3: Methodically Tested and Checked
  - EAL4: Methodically Designed, Tested, and Reviewed
  - EAL5: Semi-formally Designed and Tested
  - EAL6: Semi-formally Verified Design and Tested
  - EAL7: Formally Verified Design and Tested

- System test cases: 46
- Acceptance test cases: 16
- approx. 80% of the requirements are verified by testing, 20% by review and inspection.
- Requirements coverage:
  - 100% for 114 software requirements and 98 user requirements.
  - Independent testing has not been performed ➔ EAL 4
### Technology Readiness Level (TRL)

<table>
<thead>
<tr>
<th>Models</th>
<th>TRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual system „flight proven“</td>
<td>TRL 9</td>
</tr>
<tr>
<td>Actual System „flight qualified“</td>
<td>TRL 8</td>
</tr>
<tr>
<td>FM (Flight Model)</td>
<td>TRL 7</td>
</tr>
<tr>
<td>PFM (Protoflight Model)</td>
<td>TRL 7</td>
</tr>
<tr>
<td>QM (Qualification Model)</td>
<td>TRL 6</td>
</tr>
<tr>
<td>EQM (Engineering Qual. Model)</td>
<td>TRL 6</td>
</tr>
<tr>
<td>EM (Engineering Model)</td>
<td>TRL 5</td>
</tr>
<tr>
<td>EBB (Elegant Breadboard)</td>
<td>TRL 4</td>
</tr>
<tr>
<td>BB (Breadboard Model)</td>
<td>TRL 3</td>
</tr>
<tr>
<td>Technology concept</td>
<td>TRL 2</td>
</tr>
<tr>
<td>Basic principles</td>
<td>TRL 1</td>
</tr>
</tbody>
</table>

#### Definition TRL 5:

1. System/subsystem/component validation in relevant environment:
2. Thorough testing of prototyping in representative environment.
3. Basic technology elements integrated with reasonably realistic supporting elements.
4. Prototyping implementations conform to target environment and interfaces.

#### SPCC Implementation:

1. **YES:** see demonstrator setup
2. **YES:** 148 UT, 46 ST, 16 AT
3. **YES:** Software components embedded in TSP environment on next generation platform computer.
4. **YES:** LEON 4 with MIL1553 and SpW Interfaces.
Conclusion (1)

- The implemented architecture allows to combine safety and security features on one platform.
- The additional costs when introducing security on-board a satellite in terms of processing resources, mass, power consumption and development effort is limited.
Conclusion (2)

- **SPCC implements a MILS compliant architecture**
  - a CPU board that provides privilege modes, MMU and a memory bus,
  - a separation kernel guaranteeing separation by allocation of CPU time and memory to partitions
  - controlled information flow by configuring communication rights of partitions,
  - software components providing crypto functionality implementing the information flow policy.

- The TRL achieved is five
- The EAL achieved is four
Utilizing Multicore - Example

Processor Module (GR-CPCI-LEON4-N2X)

- Partition 0: System Support Systems (Ctrl. & Data Hdl., FDIR, System...)
  - SPCC-R
- Partition SPCC-E
- Partition SPCC-CC
- Partition APP1
- Partition APP2
- Partition SpW Equipm. Handler
- Partition MIL1553 Equipm. Handler

Operating System Abstraction Layer (OSAL)

Separation Kernel (PikeOS)

CPU0, CPU1, CPU2, CPU3

SpW Port A, SpW Port B, MIL1553

to "ground"

to equipments

SPCC – Software Elements for Security – Partition Communication Controller

7. December 2015
Questions ?

OR

Darts ?