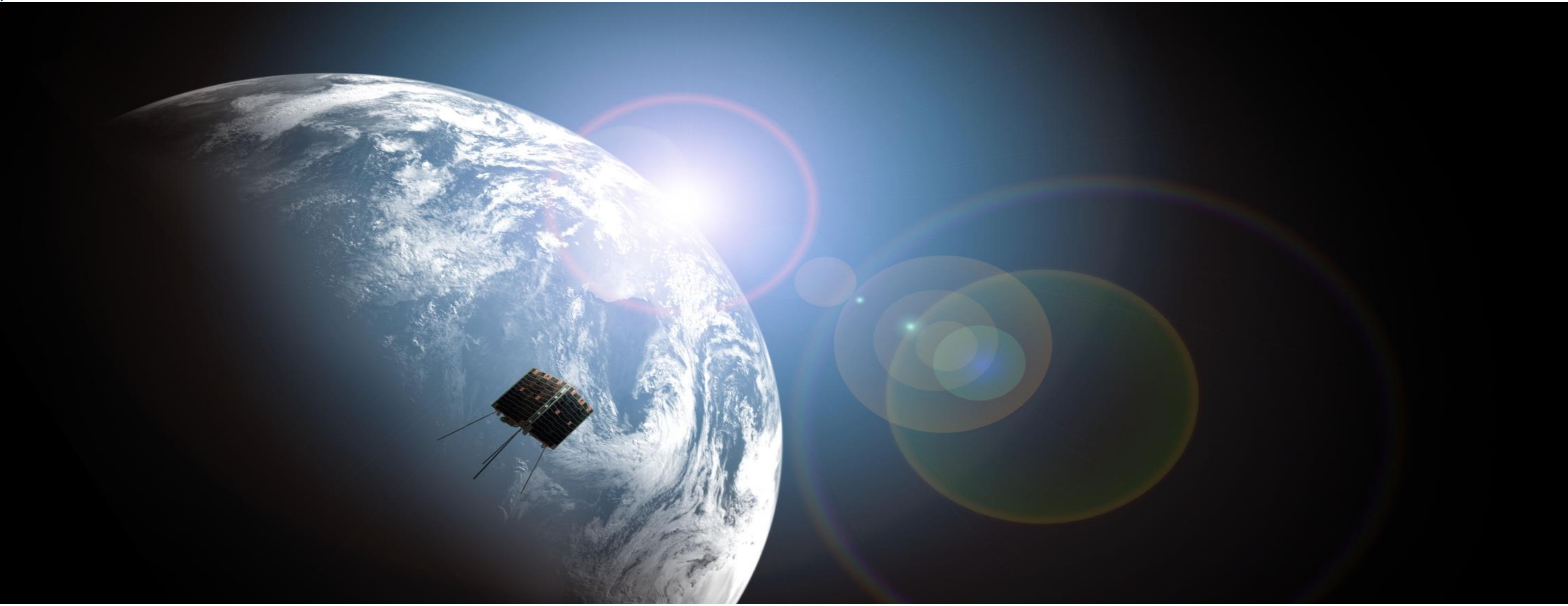


The ESAIL mission: a CANopen high-performance platform

Miroslav Brada , Yago Isasi , Christophe Molon-Noblot

CAN in Space Conference Goteborg 2019

Agenda

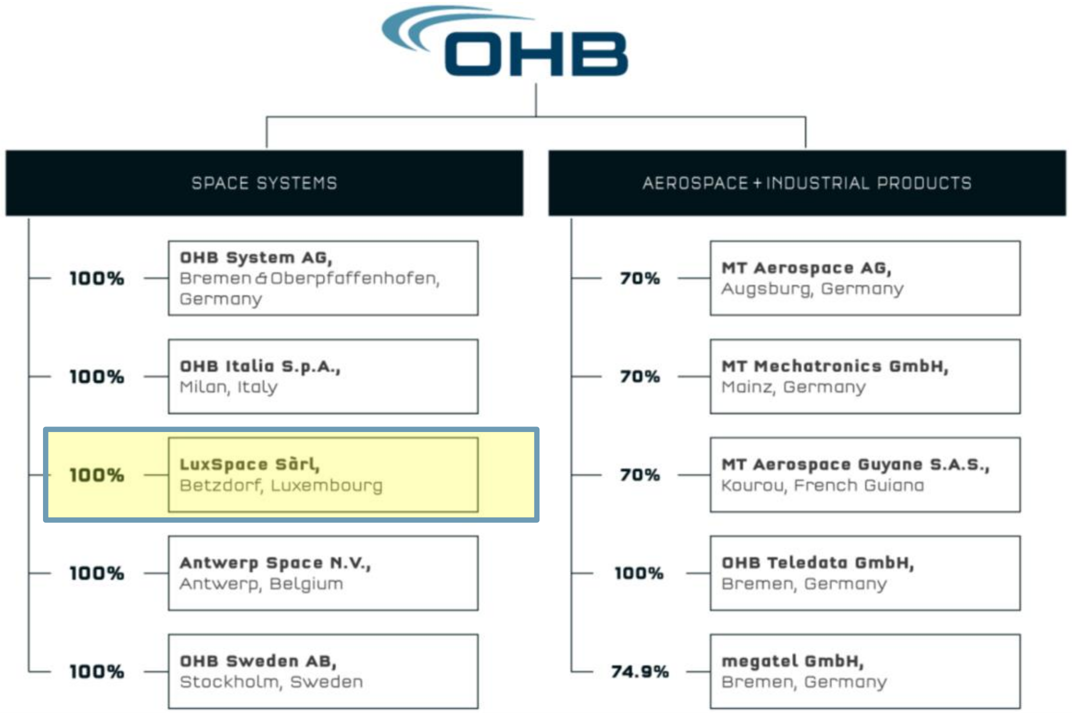
- LuxSpace company introduction
- Microsatellite history at LXS
- Introduction to ESAIL
- Why we use CAN bus
- CAN bus use in ESAIL (HW / SW Elements / ECSS CAN)
- Challenges on the road.
- Details on design / implementation of the specific ECSS CAN features
- Validation & Verification
- Conclusions

ESAIL Mission : A CANOpen high performance platform

History
- Founded in November 2004
- Business activities started in January 2006
- ISO 9001 certification since 2008

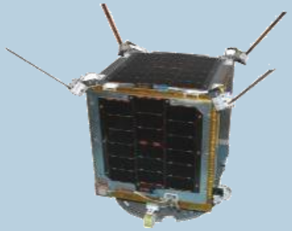
Shareholders
- Independent legal entity
- Owned 100% by OHB SE/Bremen

Site and size
- Offices close to SES Campus in Betzdorf (LU)
- 55 staff from 12 different countries



LuxSpace microsatellites

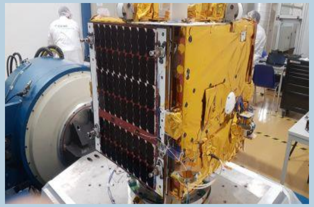
launched



VesselSat-1 and -2:
- Based on Trition-1 platform
- Platform 30 cm cube, 30 kg, 4W, payload 3 kg
- Embarking AIS receiver

No CAN

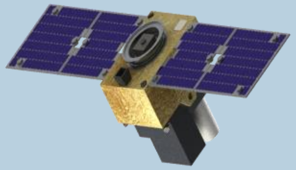
9th of September



E-Sail (Triton-2 platform)
- 70 cm cube, 100 kg, 10W, payload 15 kg, 3-axis stabil.
- ESA SAT-AIS program (AIS with high data rate link)

CANopen used for OBC/GPS/Payload Units (PDHU/PDD/AIS Rx)

Under development

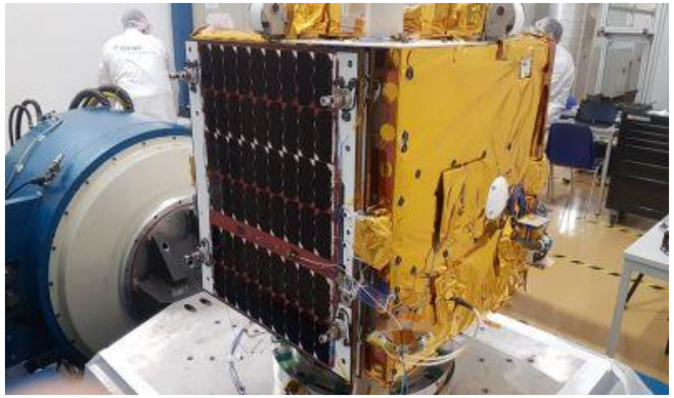


Triton-X Platform (S-M-L sizes)
- Platform Mass: 25 to 100 kg,
- Payload:
  - Mass 6 to 75 kg
  - Power 10 to 160 W

Full CANopen in all units and subsystems

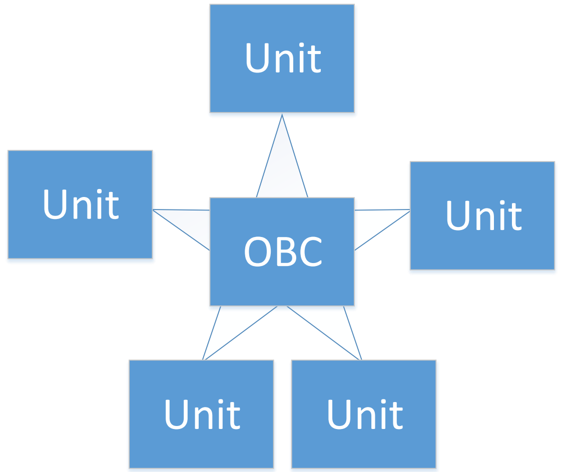
Introduction to ESAIL

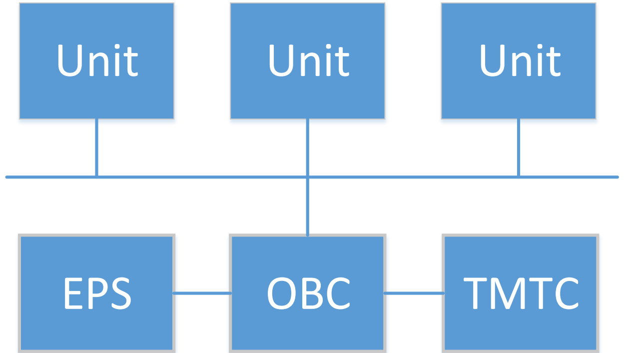
- European SAT AIS Constellation
- ARTES 21
- LuxSpace is prime contractor
- Cost-oriented ESA AIS mission
- 100 kg
- 3 axis stabilized platform
- High performance AIS payload
- High speed C-Band downlink
- Launch planned in 9th September 2019



Why we used CAN bus in ESAIL ?

- **VesselSat mission used star architecture**
- **Problematic for larger satellites such as ESAIL**
  - Large number of interfaces required on OBC that take space and power
  - Low flexibility
  - Complicated Harness
- **Obvious solution to use Command and Control bus**
- **Highly critical interface (TMTC and EPS) still directly connected to the OBC**





Why we used CAN bus in ESAIL?

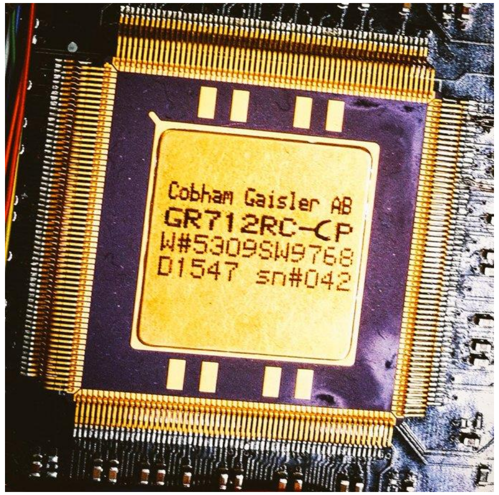
CAN

- Well known industry standard
- Robust, designed for harsh automotive environment
- Availabe on COTS parts
- Low power
- Standard test equipment and tools available (with affordable cost)

CAN ECSS

- ECSS-E-ST-50-15C available from 2015.
- Based on widely known CANopen.
- Later reusability for other projects.
- Test equipment and tools were available on affordable costs.

ESAIL OBC CAN components



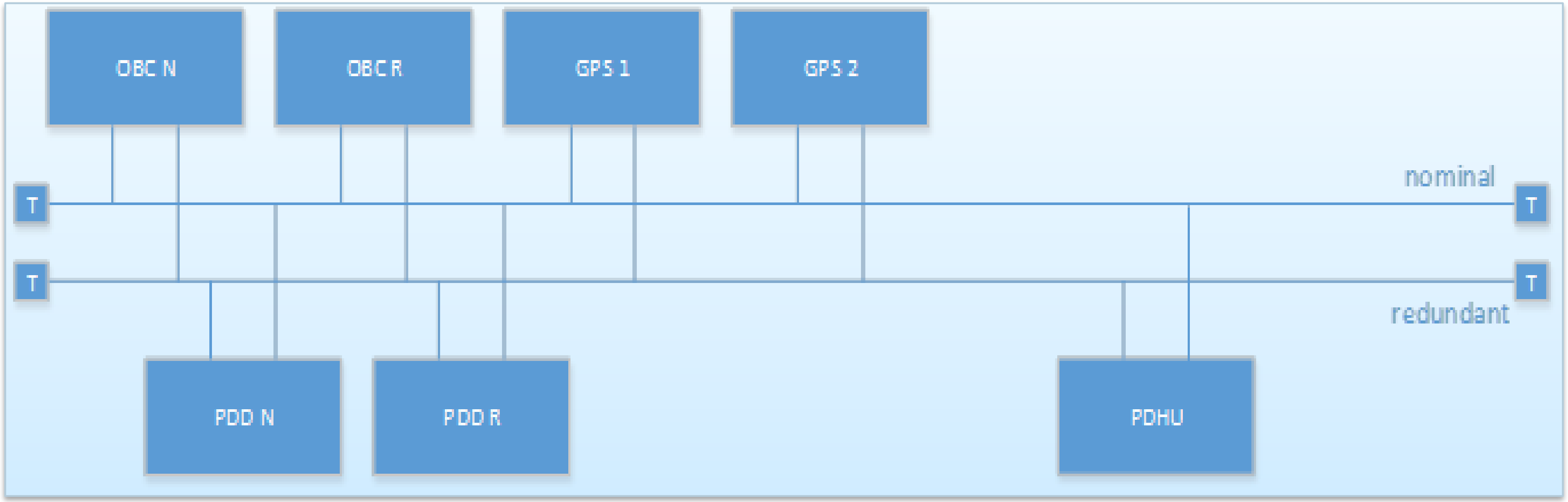
CANopen
LXS implementation

CAN DRIVERS
RTEMS 4.8 **OC_CAN** drivers modified

RTEMS 4.8
QUALIFIED

- **OS: RTEMS 4.8 qualified (NO CAN DRIVERS)**
- **DRIVERS: OC_CAN Driver ported from the mainline RTEMS 4.8 distribution**
- **CANopen** solution: **Developed by LXS in the ESAIL Context**
- **HW: LEON 3/ GR712RC**

ESAIL Block diagram of units using CAN



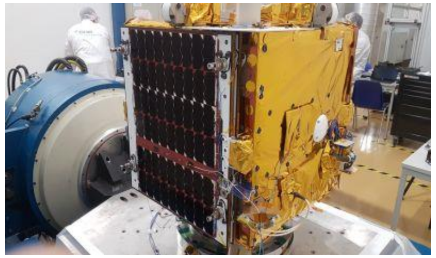
- **2 OBC** Nominal and Redundant in Cold Redundancy.
- **2 GPS** Units Nominal and Redundant in Cold Redundancy
- Payload: **1 PDHU** and **2 PDD (Payload Data Downlink)** units

ESAIL ECSS CAN solution

ECSS CAN (close to Minimal Set Protocol):

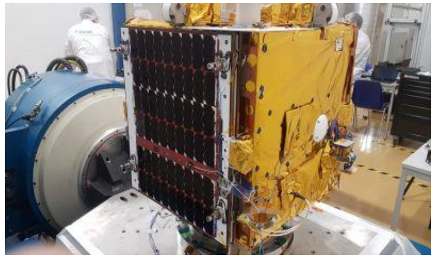
- LXS has designed / implemented a CANOpen like protocol based on a tailored version of the ECSS standard to suit the mission requirements removing non-mission related features. (Closer to the ECSS “Minimal Set Protocol”).
- This implementation is used by the ESAIL **On-board Computer** as part of its **Application SW** (ASW)

Network Management / Bus Selection



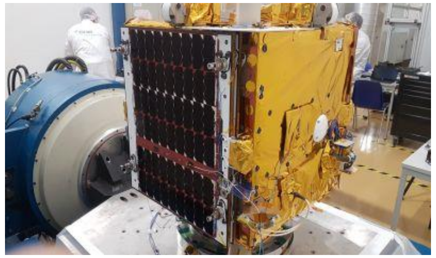
- **Heart beat message**
  - Implemented as per ECSS.
- **Boot message**
  - Implemented as per ECSS.
- **Emergency messages**
  - Optional, currently not used.
- **Node State Machine**
  - Implemented as per ECSS.
- **Bus selection**
  - Cold redundant bus architecture.
  - Bus selection implemented as per ECSS.

COB-ID and PDOs



- **Different priorities allow efficient exchange of data**
  - No need of predefined timeslots to ensure timely delivery
  - Priority not primarily based on unit, but on type of message (COB-ID).
- **Predefined connection set has been applied**
- **Remote Frame Request**
  - Not implemented (as recommended per ECSS)
  - Functionality implemented by a pair of RX / TX PDO as described in "**minimum implementation"** in the ECSS section 9.4
- **Limitation of PDO size (8 bytes)**

Object Dictionary / PDO definitions



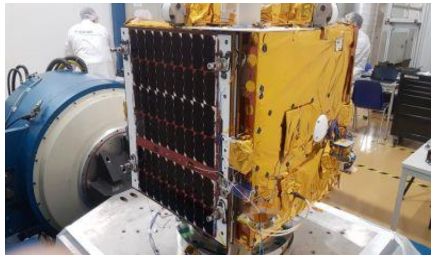
Object Dictionary

- Standard way of access data in a unit.
- Simple units don‘t implement OD.
- OBC doesn’t implement the OD; OBC acts as a master and initiates all the communications.
- Only PDHU (Payload Data Handling Unit) implements OD because of complexity of payload configuration and operations.

PDO definitions

- All PDO definitions (mapping/parameters) are hardcoded in ESAIL.
- Flexible PDO definition as per CANopen would highly increase complexity on unit and OBC side while not adding any value.

SDO

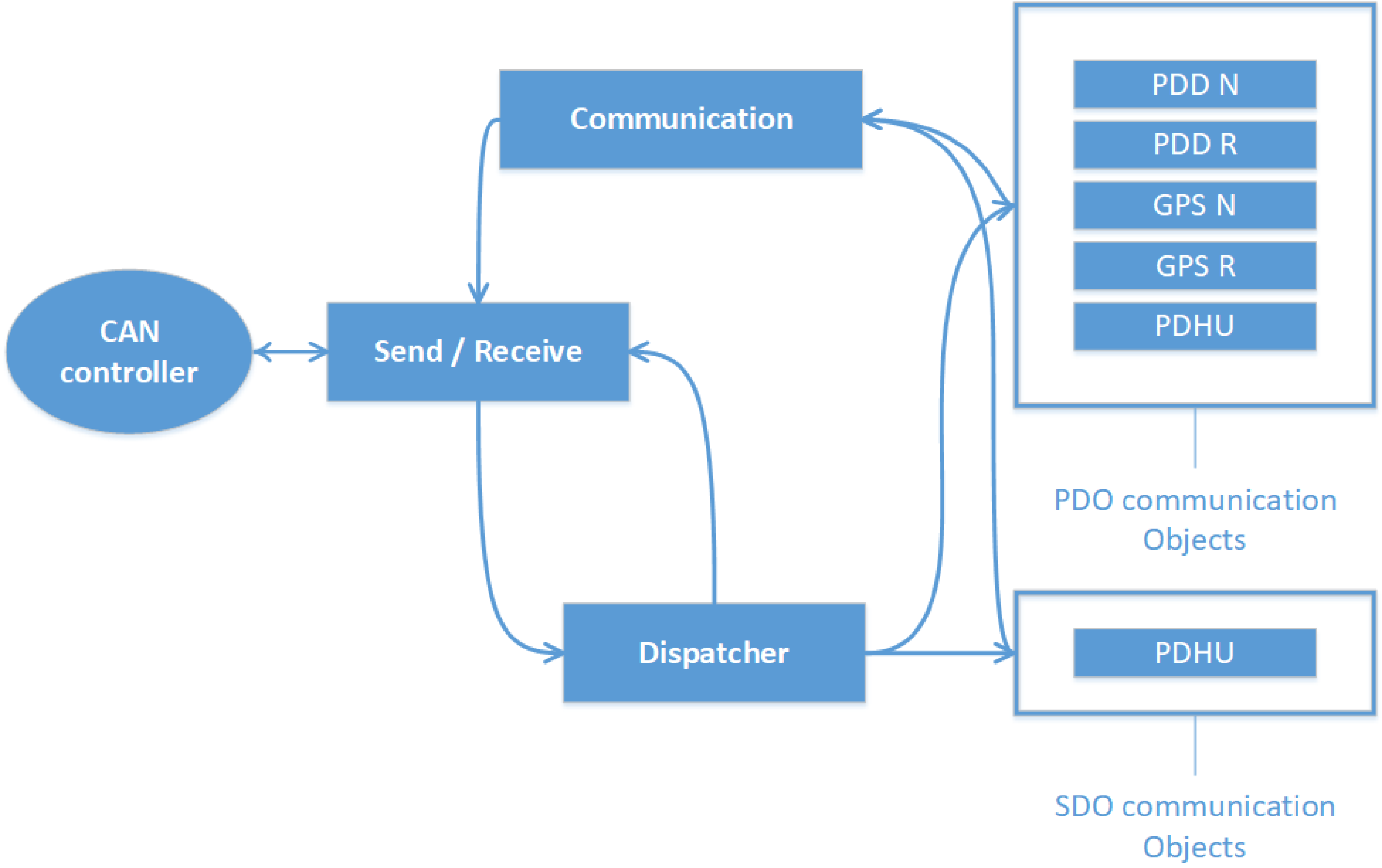


- SDO is only standard way to transmit bigger blocks of data
- ESAIL uses SDO´s to configure units and when sending/receiving data not fitting in a PDO.
- For ECSS CAN SDO are optional
- Simple units (PDD, GPS)
  - No SDO implemented
  - Parameters can be set with PDO (command/response protocol)
  - Configuration on ground or hardcoded
- Complex units (PDHU):
  - Change configuration with SDO (but no dynamic PDO reconfiguration)
  - SW / Firmware uploads
- No different priorities for different data types

Design / Implementation details

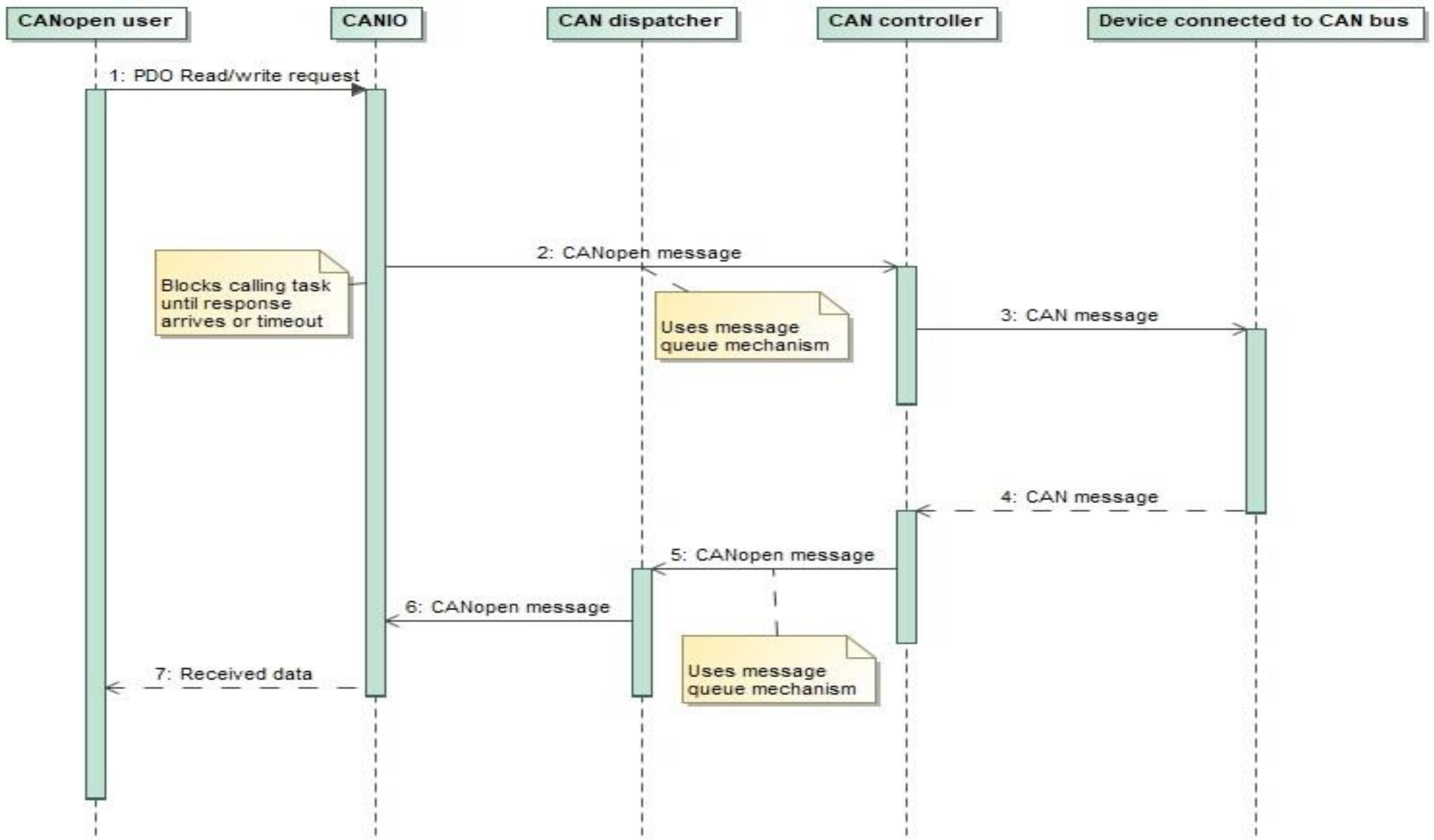
- **Four different logical modules** have been modelled:
  - **Initialization module** :Handles the initialization and configuration of CAN core using standard calls of modified RTEMS OC_CAN driver and initialization of CANopen related data structures, i.e. input and output message queues and synchronization semaphores.
  - **Send/Receive module:** Handles the sending and receiving of CAN messages using read/write operations provided by RTEMS OC_CAN driver and retrieving and inserting of CAN messages from/to internal message queues of ASW.
  - **Dispatcher module:** Handles incoming CAN messages, decodes their recipients. It also handles actions needed by stateful communication (SDO block transfers or PDO serial tunnel).
  - **Communication module**: Provides the communication interface between ASW tasks and devices connected to CAN bus. In this module, data is encapsulated according to corresponding CANopen protocol needs (SDO, PDO, PDO serial tunnel or NMT message) and inserted into the internal CAN output message queue, from where the message is then sent to CAN controller.

Design / Implementation details



Flow of data between the logical modules

Design / Implementation details



Example sequence diagram of PDO communication

Design / Implementation details

- Metrics by Modules:

7	files
22	functions
249	statements
3.86	avg. complexity/function
11.32	avg. statements / function
1281	SLOC without comments including header files

CAN Communication module

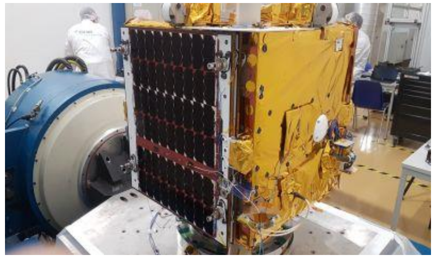
9	files
43	functions
405	statements
3.81	avg. complexity/function
9.42	avg. statements / function
1023	SLOC without comments including header files

CAN dispatcher module

3	files
3	functions
117	statements
10.33	avg. complexity/function
39.00	avg. statements / function
339	SLOC without comments including header files

CAN Send/receive module + initialization

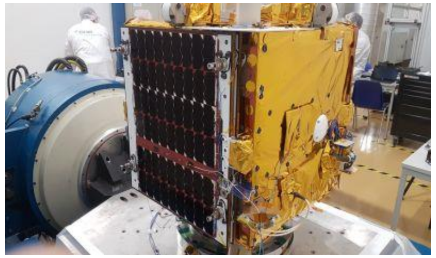
Validation and Verification details :



- **Unit Testing**
  - Using check framework, test have been later integrated in VectorCast for test coverage computation.

- **SVF (Software Validation Facility)**
  - Validation of the ASW has been performed using also SVF with emulated equipment providing simulation of the CAN bus controller and lines.
  - It allows to produce simulation of **failure injections** to demonstrate the robustness of the ASW and in general of the different units communicating via CAN.
  - Platform used: OHB System **Ru**ntime for **S**imulation (**Rufos**).

Validation and Verification details :



- **Vector CANoe has been used:**
  - Tool for **CAN unit verification** coming from suppliers.
  - Tool to monitor the proper CAN communication between different units in a **Flatsat** environment (using Engineering Model equipment).
  - **Enables analysis of communication problems**, replaying communications but also can be used as debugging tool (e.g. analysis of what was being communicated just before the crash of SW).

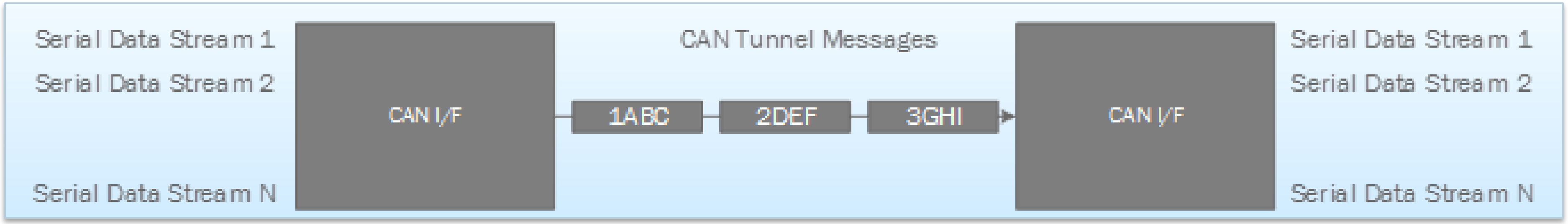
Challenges achieved

- **ECSS Standard use:**

  Final good synchronization of all CAN units from different suppliers despite preliminary issues due to the standard had some door open to interpretation: e.g. Supplier understood that that the "NMT Master message" could be considered as being the periodic Heart beat message.

- **Communication with legacy devices outputting a serial stream:**

  For the sake of tunneling serial data streams, e.g. from GPS through the CAN bus, tunnel PDO was used in ESAIL. In this case multiple continuous serial data streams has been multiplexed in frame-based CAN Tunnel messages and are de-multiplexed at the receiver side again.



Challenges achieved

- **Integration of CAN legacy drivers :**

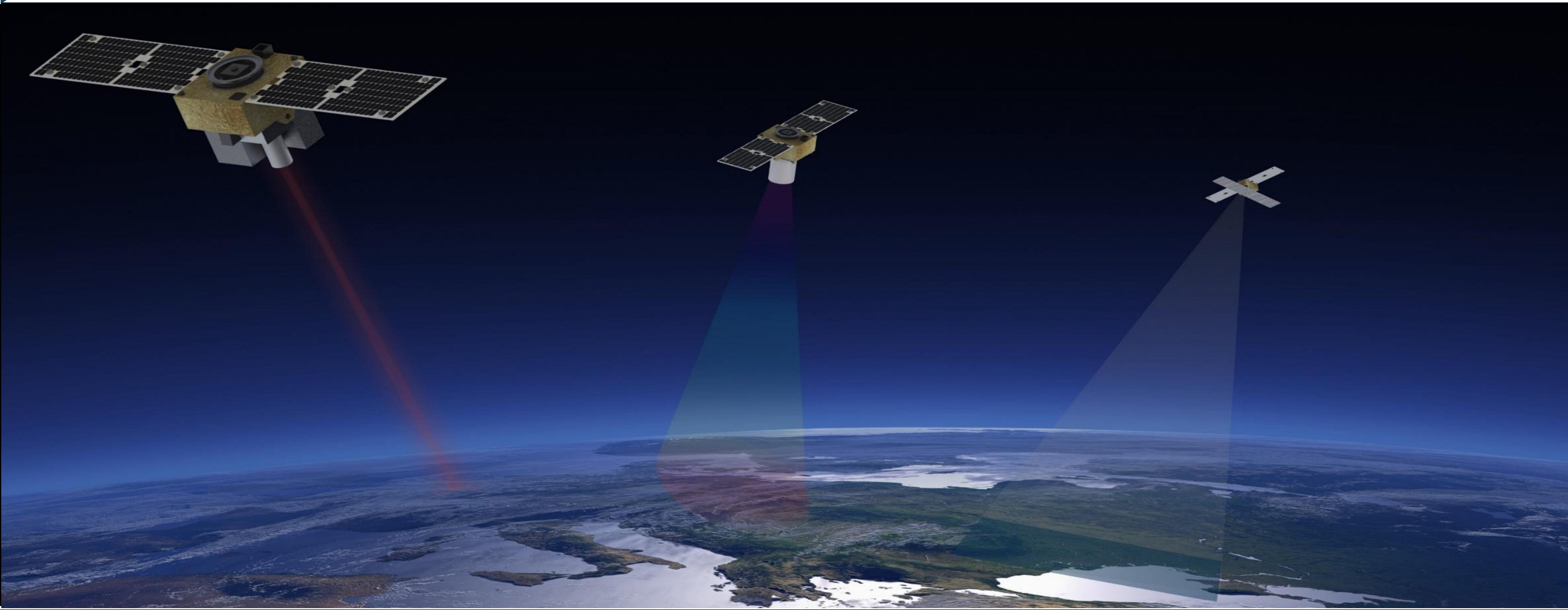
  Drivers for the CAN coming from the RTEMS open-source distribution had to be adapted for the desired performance of the mission.

- **Encapsulation of SpaceWire packets (ADCP) for indirect communication with a unit :**

  For communication with AIS receiver, the PDHU had to be used as a bridge (no direct link between OBC and AIS receiver). Therefore, the communication with AIS receiver had to be encapsulated in CAN communication.

Conclusions:

- LuxSpace has adopted ECSS CAN within the ESAIL project
- LuxSpace **has design / implemented and validated** a tailored version of the **ECSS CAN standard** based on the "**minimum implementation**" (section 9.4) used in the ESAIL OBC.
- **No major deviations** from the ECSS CAN
- **No major problems** have been identified.
- **Good synchronization of all CAN units** from different suppliers**.**
- LuxSpace ESAIL launch is scheduled for the **9th of September of 2019.**
- LuxSpace will continue using CAN as main communication bus in our new satellite platform line Triton-X, adapting / enhancing the ESAIL solution to the new platform requirements/needs



Thank you !

CAN Conference Goteborg 2019

Space.Solutions
Professional+Smart+Affordable