Where and when can we use CAN?

Gianluca Furano
ESA-ESTEC Data System Division

ADCSS 2013
ESTEC - 22 October 2013

How will we spend the next 30 minutes

- **On board busses as ESA's favourite coffee-corner topic**
- **CAN & Space, who's rotten idea is this ?**
  - The case behind distributed intelligence
  - The concept of interface controller
  - Is 1Mbps enough in 2013 ?
  - Will 1Mbps be enough in 2033 ?
  - The unhappy wedding between bus contention and real-time
- **Do we have all the toolbox for CAN in space ?**
  - Physical layer
  - Protocol layer
  - SW & development tools
- **Design for testability as a (sufficient ?) push**

How will we spend the next 30 minutes

- **Use cases for CAN**
  - Payload bus
  - Command & Control bus
  - Essential TM (&TC) bus
  - CAN as enabler of system level functional redundancy
  - ADULTS ONLY: Use of CAN as backplane
- **How to tackle obsolescence with std interfaces**
- **Some things you can't really stop**
  - Trends in space avionic systems
  - Heritage & Reuse: gain or burden?
  - Discussion points, rather than conclusions

Disclaimer (tengo famiglia)

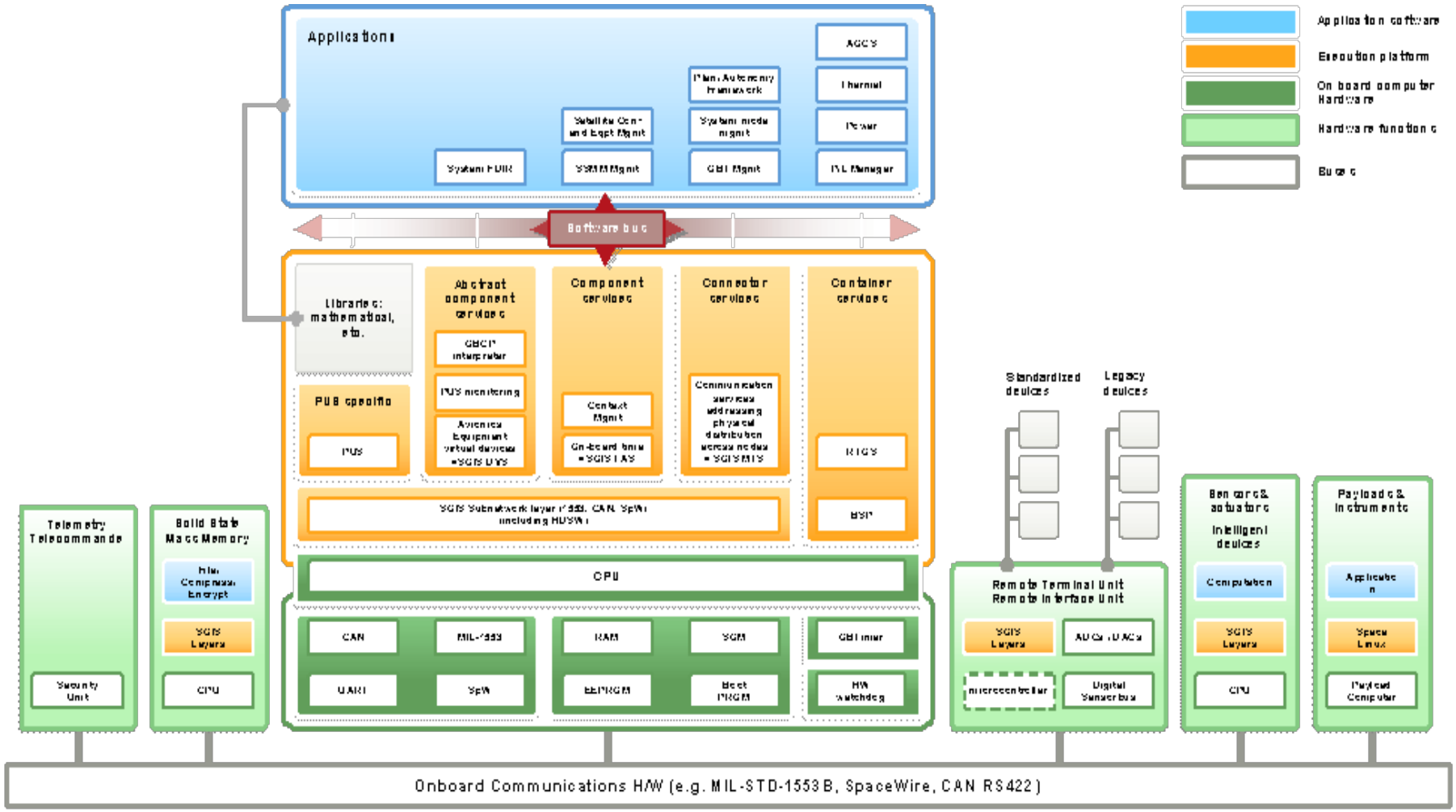
- Most of the things presented here shall be taken as discussion points
- I expect everything to be challenged by the audience
- Ideas presented here are CC+Attribution, use them for free but send me an email saying that you liked them
- Use hashtag #adcss2013 to have your tweets visualized
- Follow @ADCSS2013 to be up to date


START

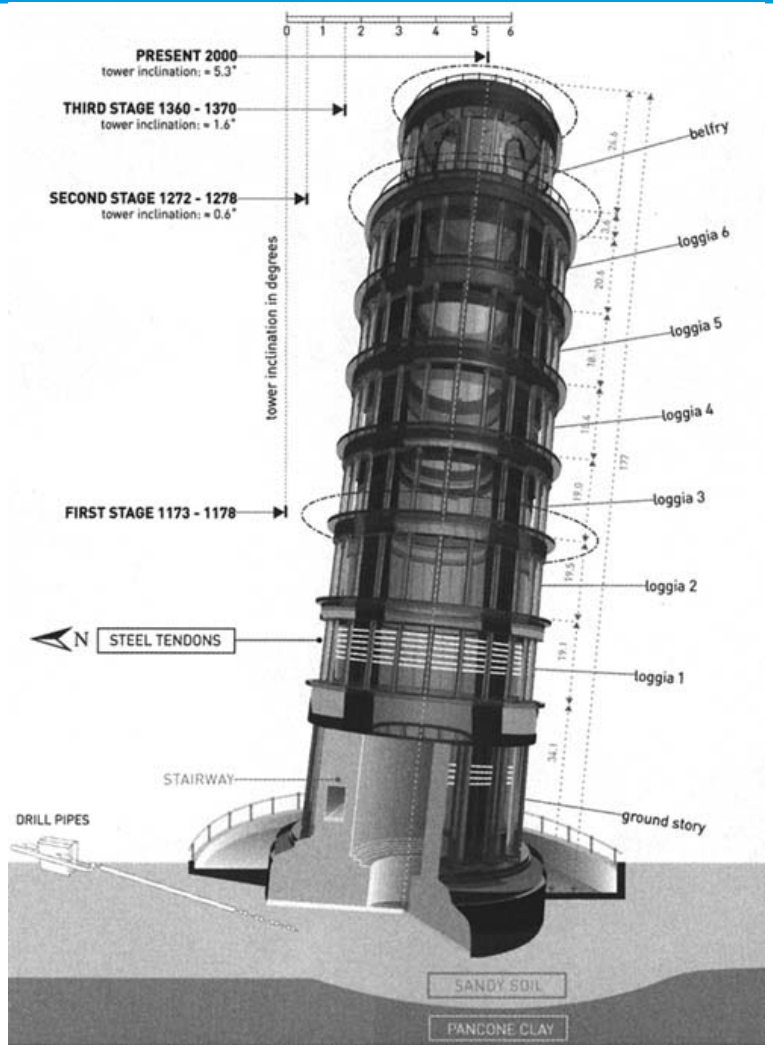
How we think future for system busses

The system busses (C&C, payload) are key elements in any DHS and should be treated as such

- Is a system-wide single point failure (and dual redundancy sometimes is not enough)
- It may be over designed and/or too demanding in terms of resources
- ‘Standard’ is a word that shall be used with caution, and verified by tests
- Deterministic behavior, Fault tolerance, and Redundancy shall be provided
- There’s not too much choice in the market besides MIL-STD-1553B and SpW.
- Levels of background industry experience and know how may vary a lot
- Master/Slave - unidirectional control over one or more other devices
- All AIT&V operations are driven by the C&C bus capabilities


Application software
Execution platform
On board computer Hardware
Hardware functions
Buses
Applications
AOCS
Thermal
Power
PUS
Software bus
Libraries: mathematical, etc.
Abstract component services
Component services
Connector services
Container services
PUS specific
CPU
CAN
MIL-1553
UART
SpW
RAM
SDRAM
EEPROM
Flash PROM
Standardized devices
Legacy devices
Remote Terminal Unit
Remote Interface Unit
Sensors & actuators
Intelligent devices
Payloads & Instruments
Telemetry Telecommands
Solid State Mass Memory
Security Unit
Onboard Communications H/W (e.g. MIL-STD-1553B, SpaceWire, CAN RS422)

Robust Engineering for system Buses



The Open Systems Interconnection (OSI) model (ISO/IEC 7498-1) provides the best conceptual model to understand the internal functions of a communication system.

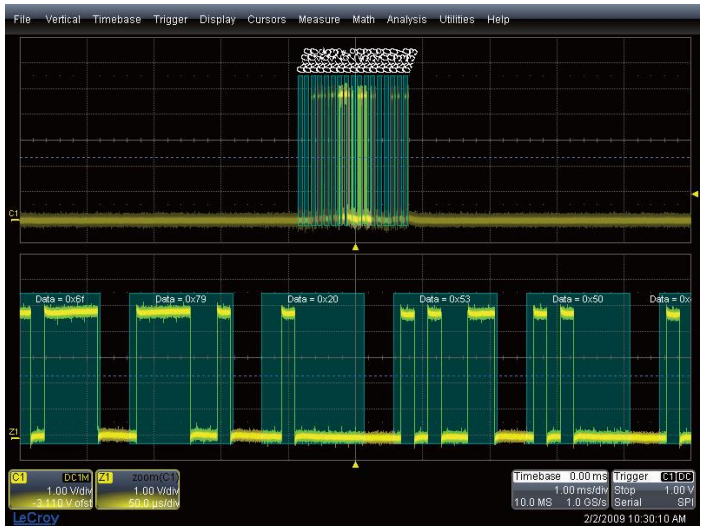
In critical application, robustness shall be tackled at each layer to avoid “ad hoc” corrections.

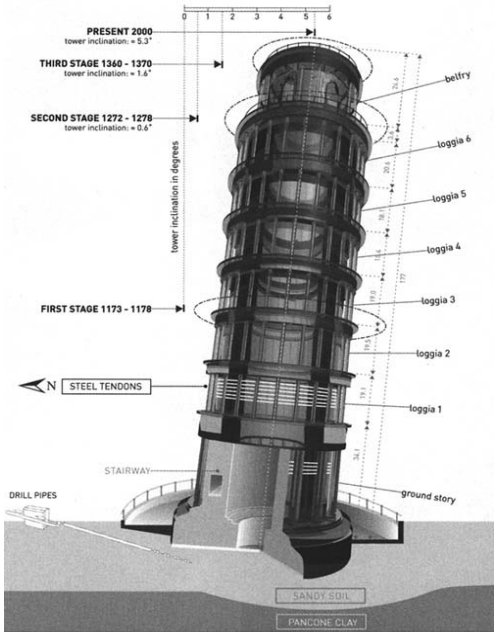
We have a long list of learnt lessons where bad foundations resulted in “leaning” design

A space bus Problem Example

Triggered by problem detected during FLIGHT MODEL PAYLOAD integration, comprehensive investigation on mission CAN bus started in ESA involving 6 people and 2 external companies.

Problem was found to have multiple concurrent causes, in all communication layers and documentation, leading towards an overall communication weakness.

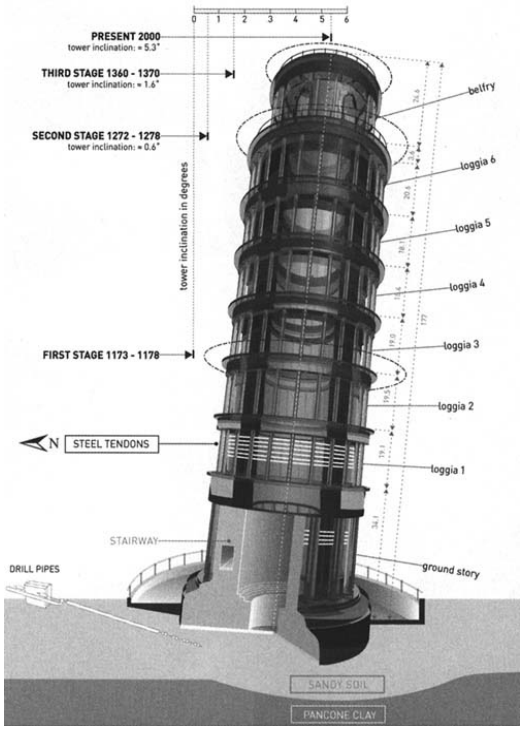




- ❑ inconsistencies were found in physical layer implementation
- ❑ re-validation of the controller IP cores used against standard test bench was performed
  - Update of ESA standard IP User Manual was triggered
- ❑ SW and HW workarounds were proposed
  - Proposed workaround was accepted by project
  - Method for network consistency check was agreed
- ❑ Lessons learnt were transferred in ECSS CAN

Robust Engineering for system Bus – the basic requirements

- There is the tendency to consider the on board bus as a 'proven' building block in space systems
- But we need to keep enough engineering capabilities to remember the technical requirements that were behind the original development of different command & control buses
- New requirements and possibilities on space avionics may lead to reconsider the current capabilities.



CAN & Space, who's rotten idea is this ?

- **September 27, 2003 ESA launched SMART-1 mission, that used CAN as unique C&C bus.**
  - It worked spotlessly until mission planned crash on moon surface on Sept 3, 2006.
- **Use of CAN as payload bus picked up in non-systematic way in latest years**
  - In payloads heritage is less important
  - Keeping payload and platform bus separate is sometimes convenient
- **Development of Exomars and ECSS standard (2009 - …) has followed this process**
- **CAN is now baselined to be used in future Telecom platforms**

CAN: Pros and Cons

Main CAN advantages

- **Very robust error detection and correction mechanisms**
- **Low cost**
- **Low power consumption**
- **Inherently multi-master**
- **European technology available**
- **Widely used and validated in terrestrial applications**

Main CAN drawbacks

- **Latency jitter is bus-load dependent**
- **Bus redundancy not part of the ISO standard**
- **Availability of components for high radiation dose tolerance**
- **Lack of standard galvanic isolation**

CAN: Pros and Cons

Main CAN advantages

- **Very robust error detection and correction mechanisms**
- **Low cost**
- **Low power consumption**
- **Inherently multi-master**
- **European technology available**
- **Widely used and validated in terrestrial applications**

Main CAN drawbacks

Latency jitter is bus-load dependent

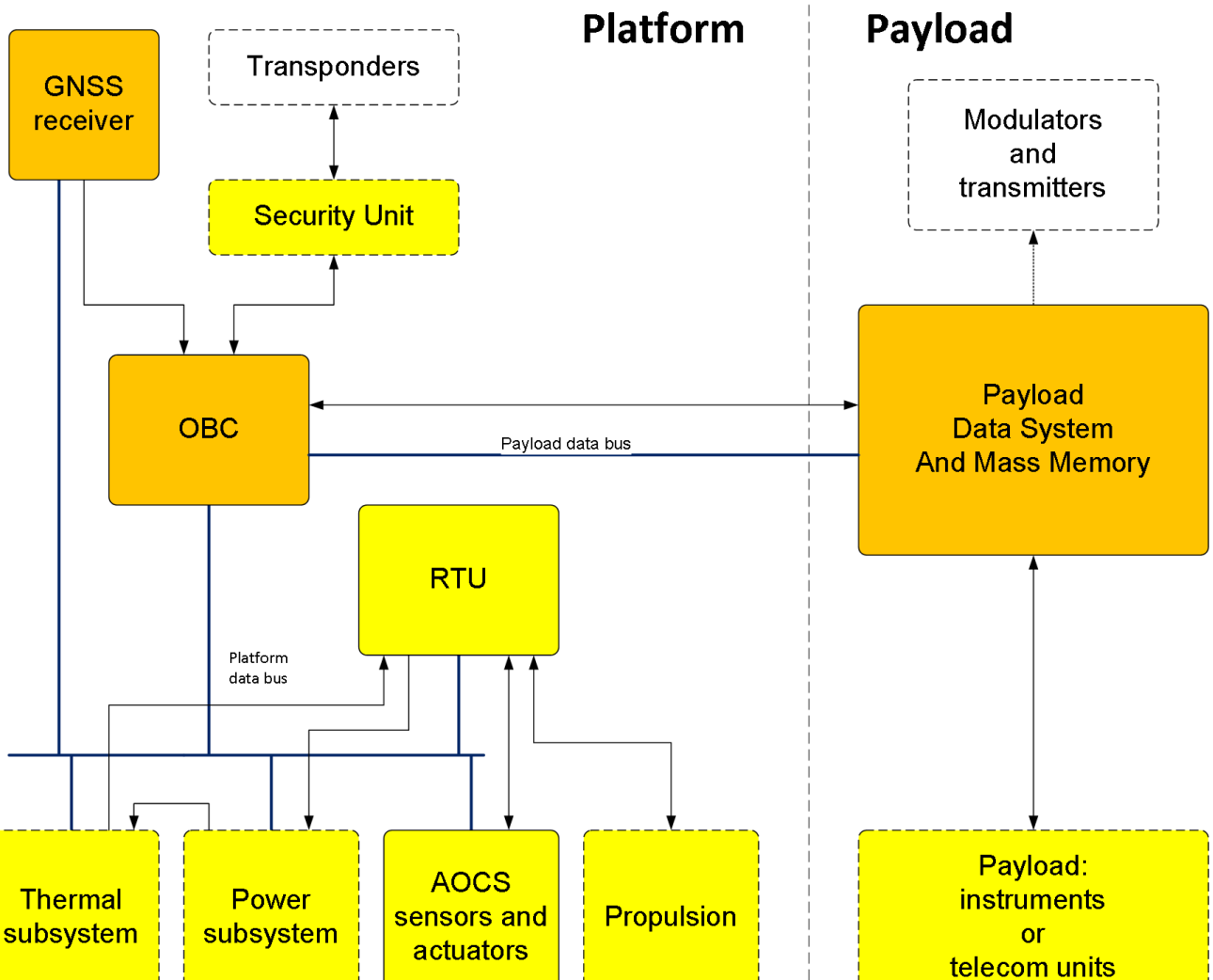
Bus redundancy not part of the ISO standard

Availability of components for high radiation dose tolerance

- **Lack of standard galvanic isolation – *is it really so important ?***

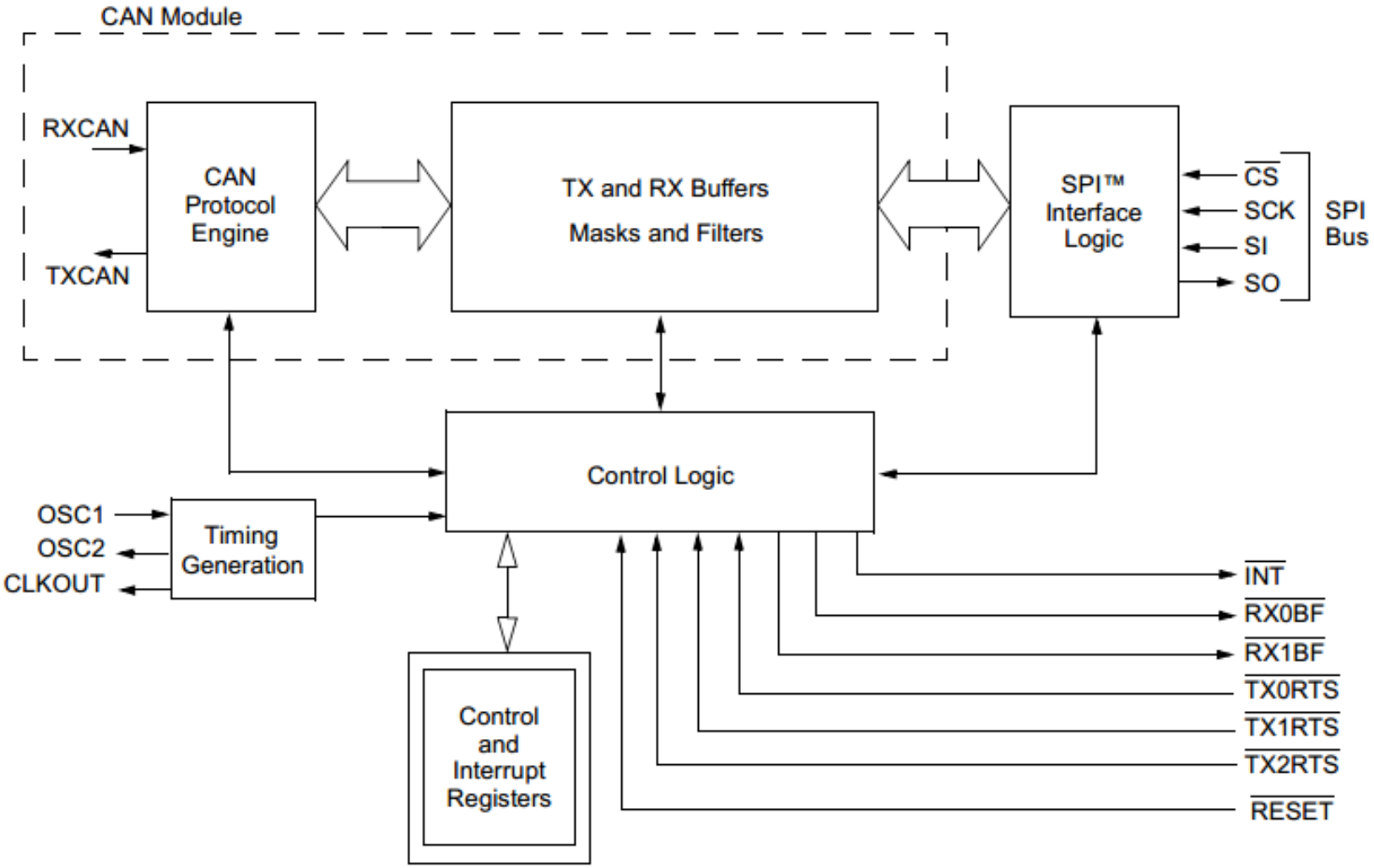
CAN Adoption, some new reference scenari

- **The case behind distributed intelligence**
  - Intelligence is distributed unevenly in our world (referring only to space avionics here…)



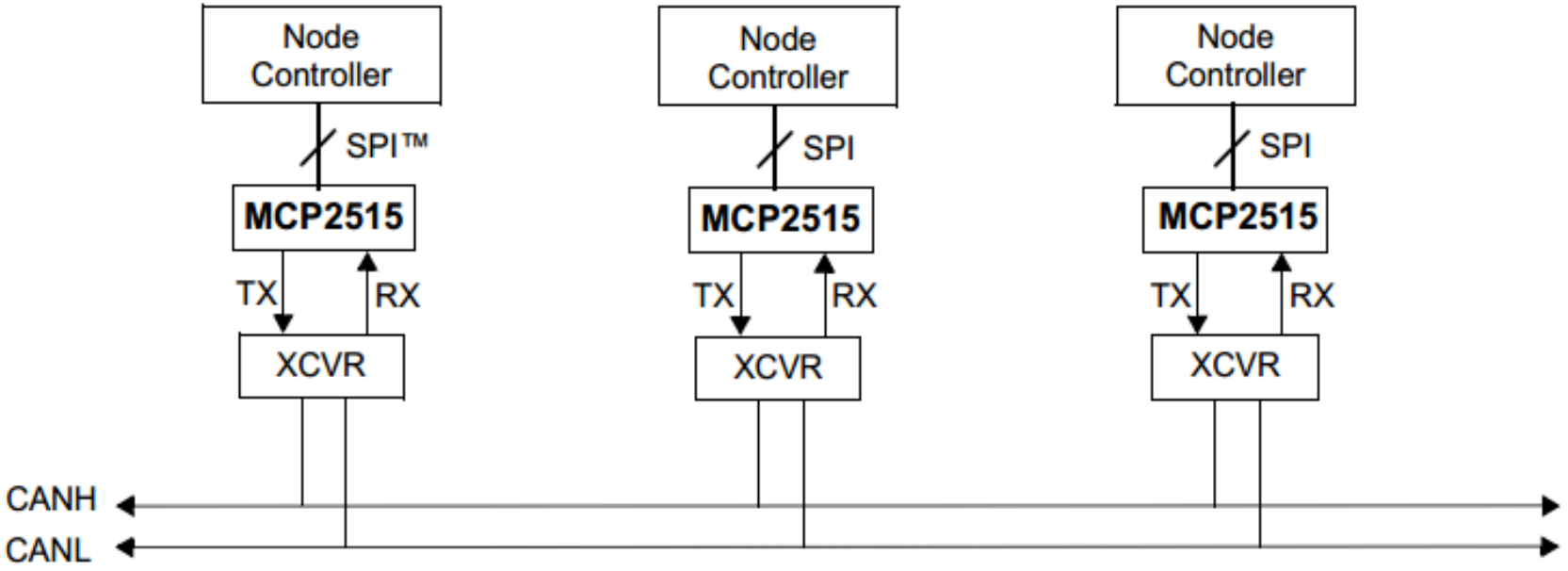
CAN Adoption, some new reference scenari

- **The concept of CAN IO expander**

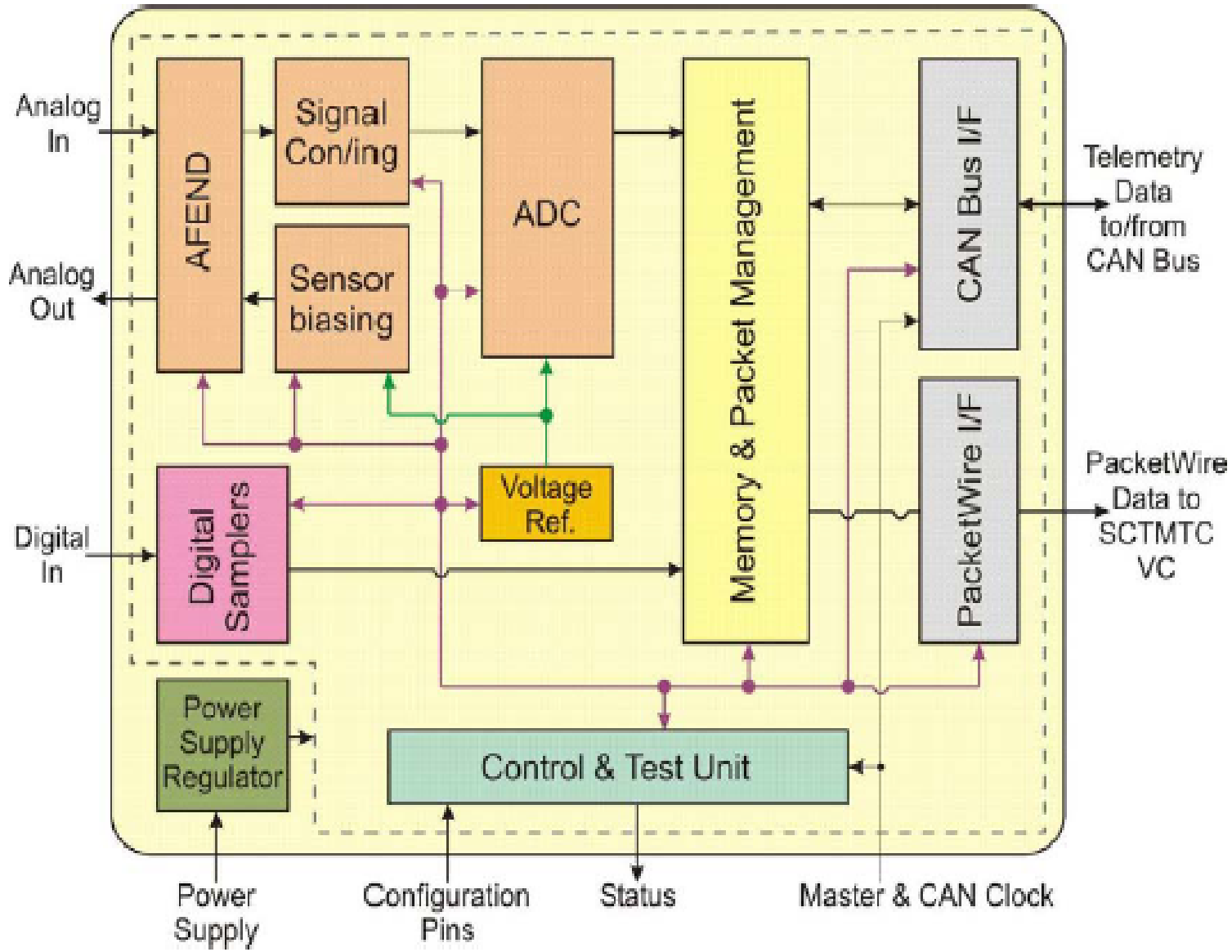


Interface controllers in CAN

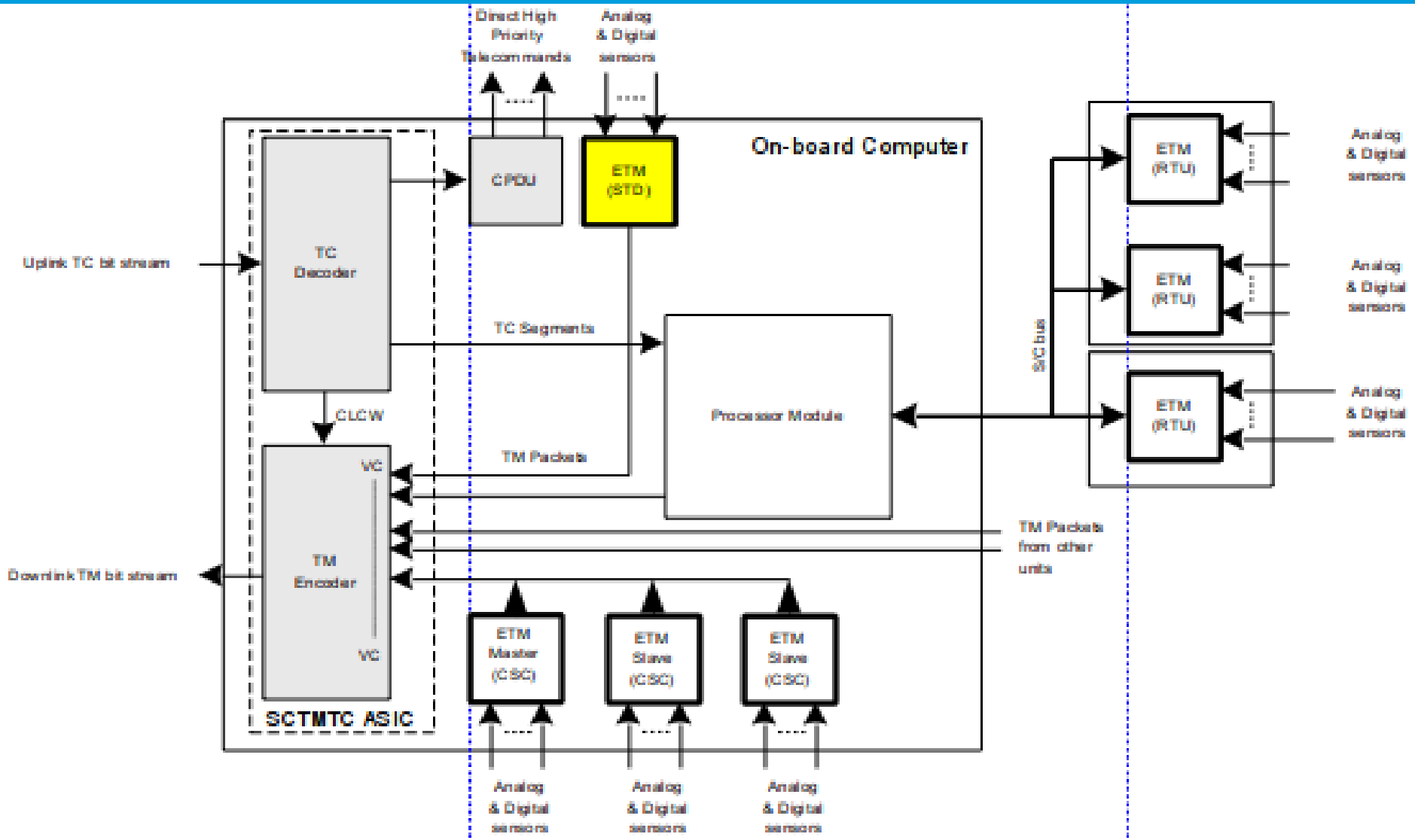
- **Stand-alone CAN controllers are one of the common CAN implementation**
  - Allow to discharge Node controller from CAN management
  - Allow very intelligent traffic management
  - Have standard SPI interface



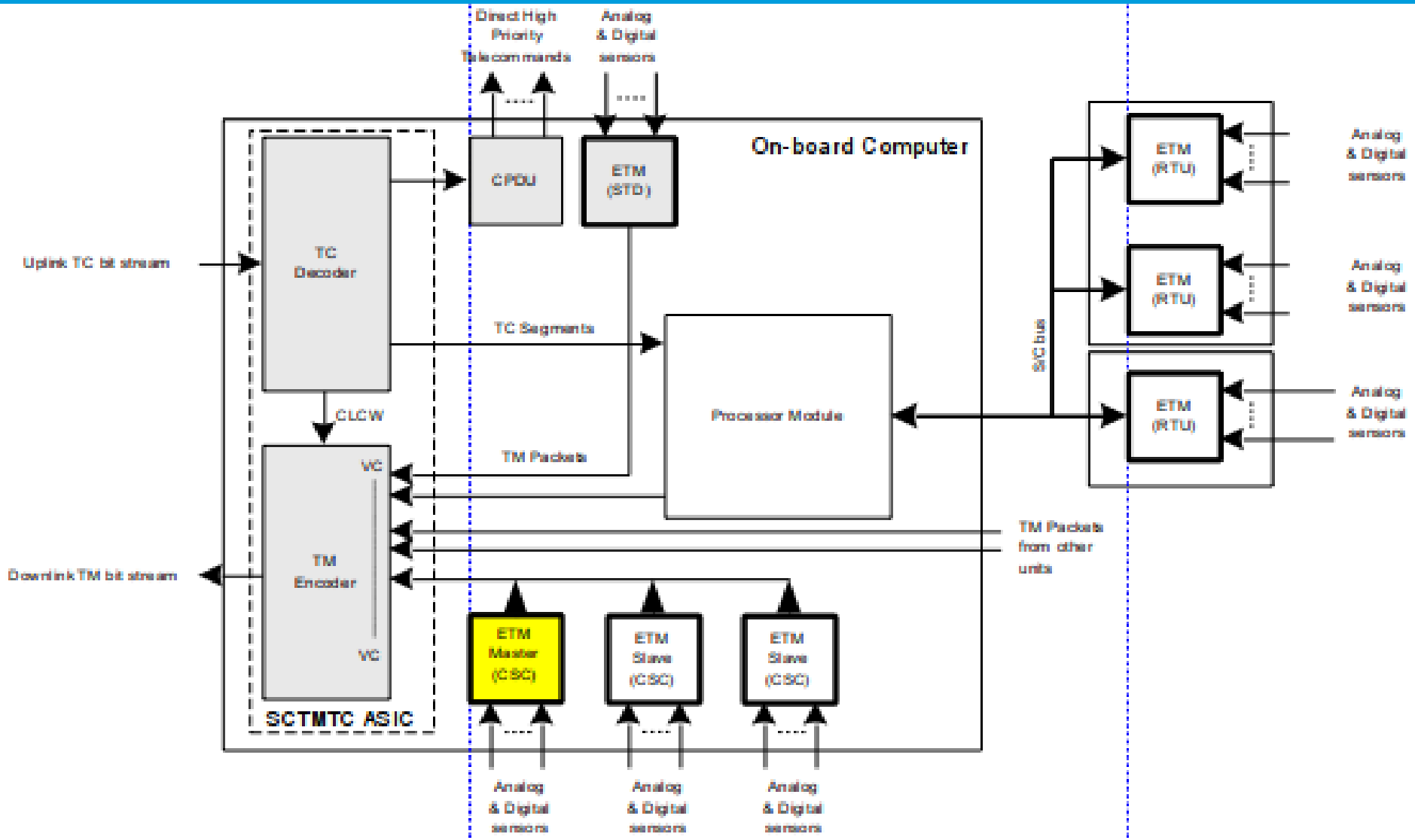
ESA's IO expander: ETM Chip



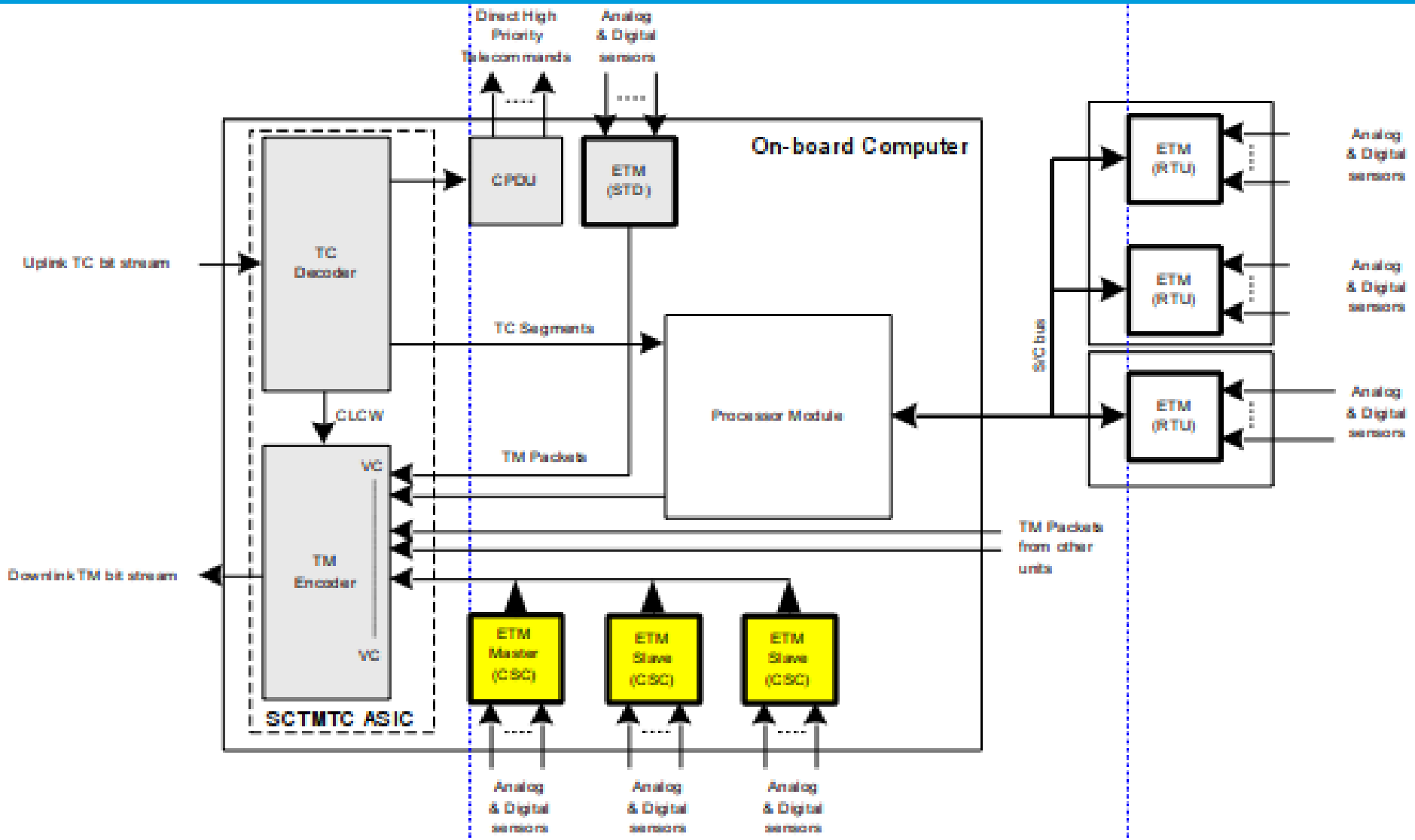
ETM ASIC System Context



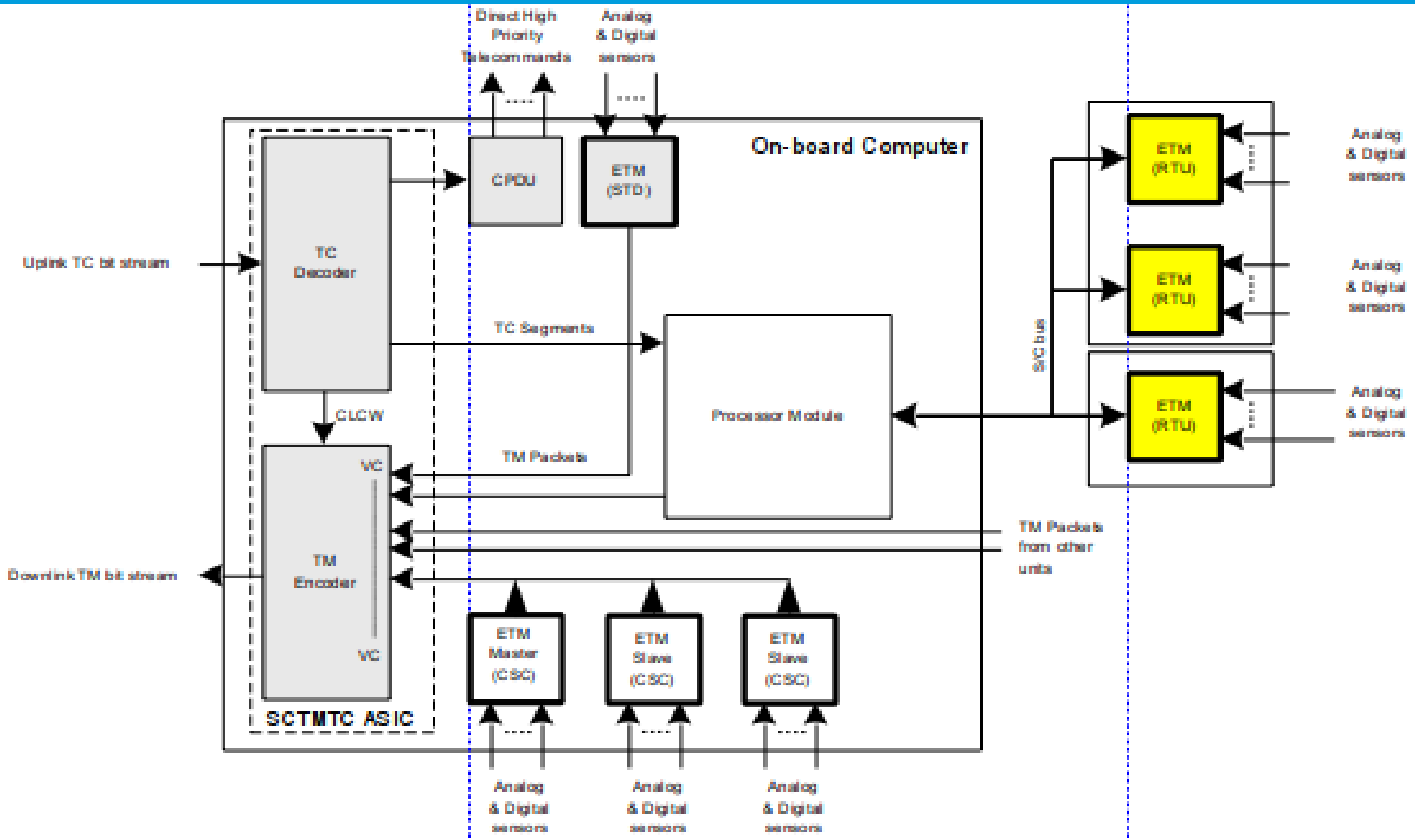
ETM ASIC System Context



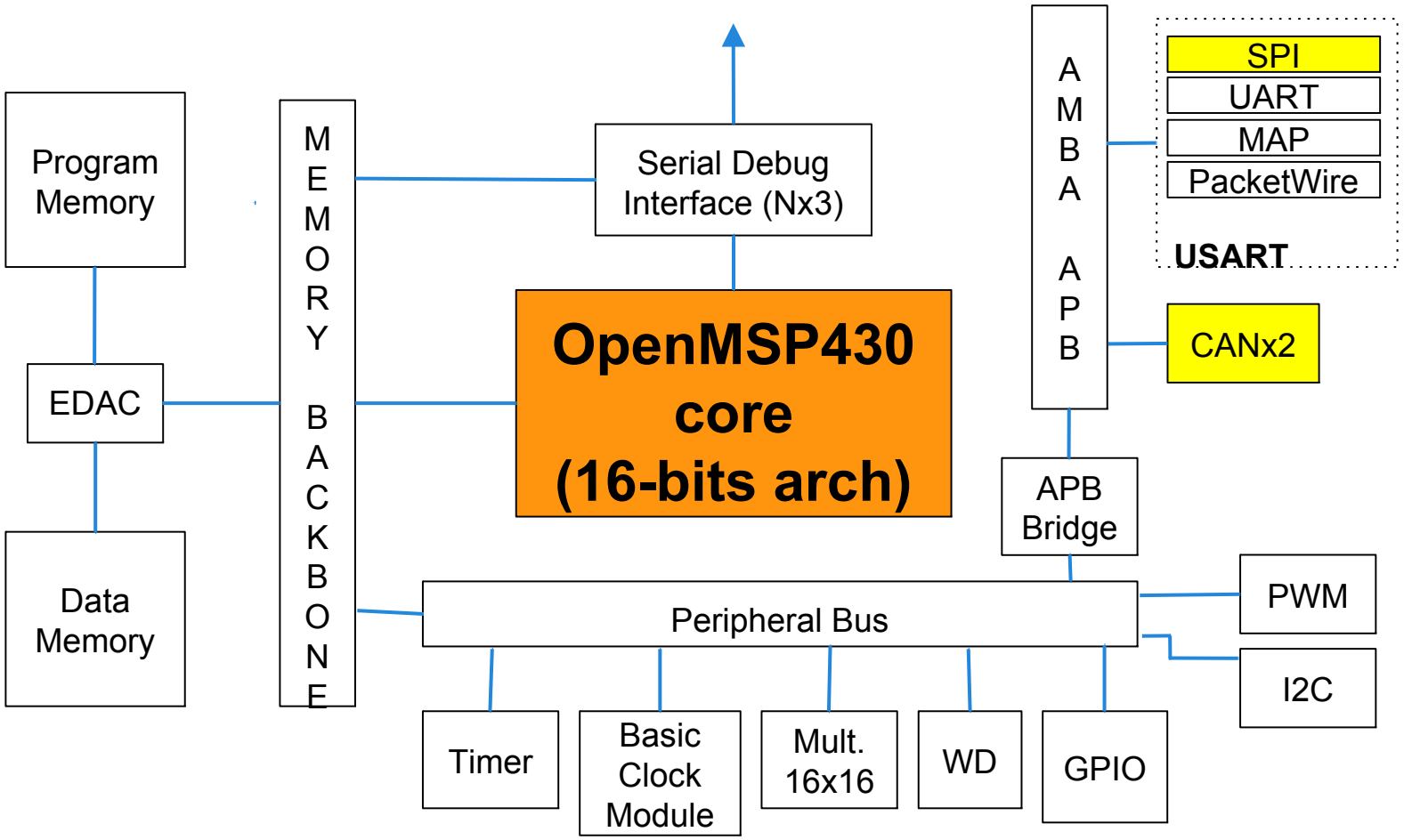
ETM ASIC System Context



ETM ASIC System Context



Example of a digital CAN uC



Different options

- **CAN I/O expanders**
  - No intelligence, relies on standard interface (preferred is SPI)
- **Digital signal controllers (DSCs)**
  - CAN, PWMs, GPIO + (analog peripherals … ADC, DAC)
- **8-bit, 16-bit and 32-bit microcontrollers**
  - With integrated “full CAN”
- **Mixed Signal ASICs solutions are possible !**
- **Depending on user's need all the above can fulfil the aim of :**
  - Reducing number of physical interfaces between units
  - Providing standard interface for different unit classes
  - Integrate or not the transceiver/redundancy management
  - Provide a safe, standard, unique SET/RESET IF line

CAN Adoption, some new reference scenari

- **Is 1Mbps enough in 2013 ?**
  - Secure transmission is possible at a maximum rate of 1000 kbps.
  - The maximum data rate at VOLKSWAGEN and AUDI has been fixed at 500 kbps, other manufacturers have even less.
  - In cars, the CAN bus system is divided into (at least) 3 special systems due to the different requirements regarding signal repetition rate and the large data volume:
    - Drive train CAN bus (high-speed) at 500 kbps with almost real time requirements
    - Convenience CAN bus (low-speed) at 100 kbps with low time requirements
    - Infotainment CAN bus (low-speed) at 100 kbps with low time requirements

CAN Adoption, some new reference scenari

- **Will 1Mbps be enough in 2033 ?**
  - Current telecom payload busses support 32 subscribers at 16kbps.
  - “Horizontal” data distribution and data push schemes help in keeping data throughput very efficient
  - High signal rate is mostly required by time distribution or low latency requirements rather than high data throughput on command and control system
- **CAN has demonstrated in past ~ 30 years to be a very resilient communication standard.**
  - It will be likely thrive rather than fade out with the advent of e-mobility
  - All e-bikes mount CAN hardware !

CAN Adoption, the troubles with bus contention

- **The unhappy wedding between bus contention and real-time**
  - For the highest priority message, a maximum latency time of the longest possible message, (130 bit times) may occur.
  - For other high priority messages deterministic maximal latency times shall be provided by means of higher layer protocol or application specific measures.

CAN Message Details	# of bit	Mps
No data, No bit stuffing needed	44	22727
No data, maximum bit stuffing (~6)	50	20000
Average case: 4 bytes of data, 5 bit stuffs	81	12346
8 bytes of data, max. bit stuffing (~11)	125	8000

CAN Specific capabilities that need to be exploited

- High availability: if a control unit fails, the rest of the system must continue to be functional as far as possible in order to exchange information.
- High data density: all control units have the same information status at all times. This means there is no difference in data between the control units. In case of faults anywhere in the system, all the connected users can be informed with equal certainty.
- Since the transmission of a CAN message may be initiated by any node of a system (multimaster capability) as soon as the bus is idle, any node of the system may exchange information with any other node. This feature is very important because it very efficiently supports event-oriented message transmission.

- **Do we have all the toolbox for CAN in space ?**
  - Physical layer
    - ECSS foresees use of ISO Transceivers
    - Other PHY (including wireless, optical) not qualified yet
    - RS-485 (referenced in ECSS) shall be considered as a legacy solution
    - If Redundancy is implemented, it shall be as described in ECSS
  - Protocol layer
    - No protocol is a protocol
    - But in any case you will need a redundancy management scheme
    - CANOpen provides all the services for redundancy management and much more
  - SW & development tools
    - Electronic Data Sheets are a key here … provided by CANOpen
- **Is Design for testability a (sufficient) push ?**

ISO transceivers, an update

- At ESA we have tested (TID and SEE) some commercial transceivers. Results available upon request.
- We are also developing DARE library additions to have CAN-compatible pads in ASICs.

- TWO major companies are planning very soon release of QML ISO CAN transceivers, rad hard.

- INTERSIL (that is assisting to this conference) has 3.3V CAN Bus Transceiver planned, with Ground-up Rad Hard development.

- You can contact me or them to get more information on all the above.

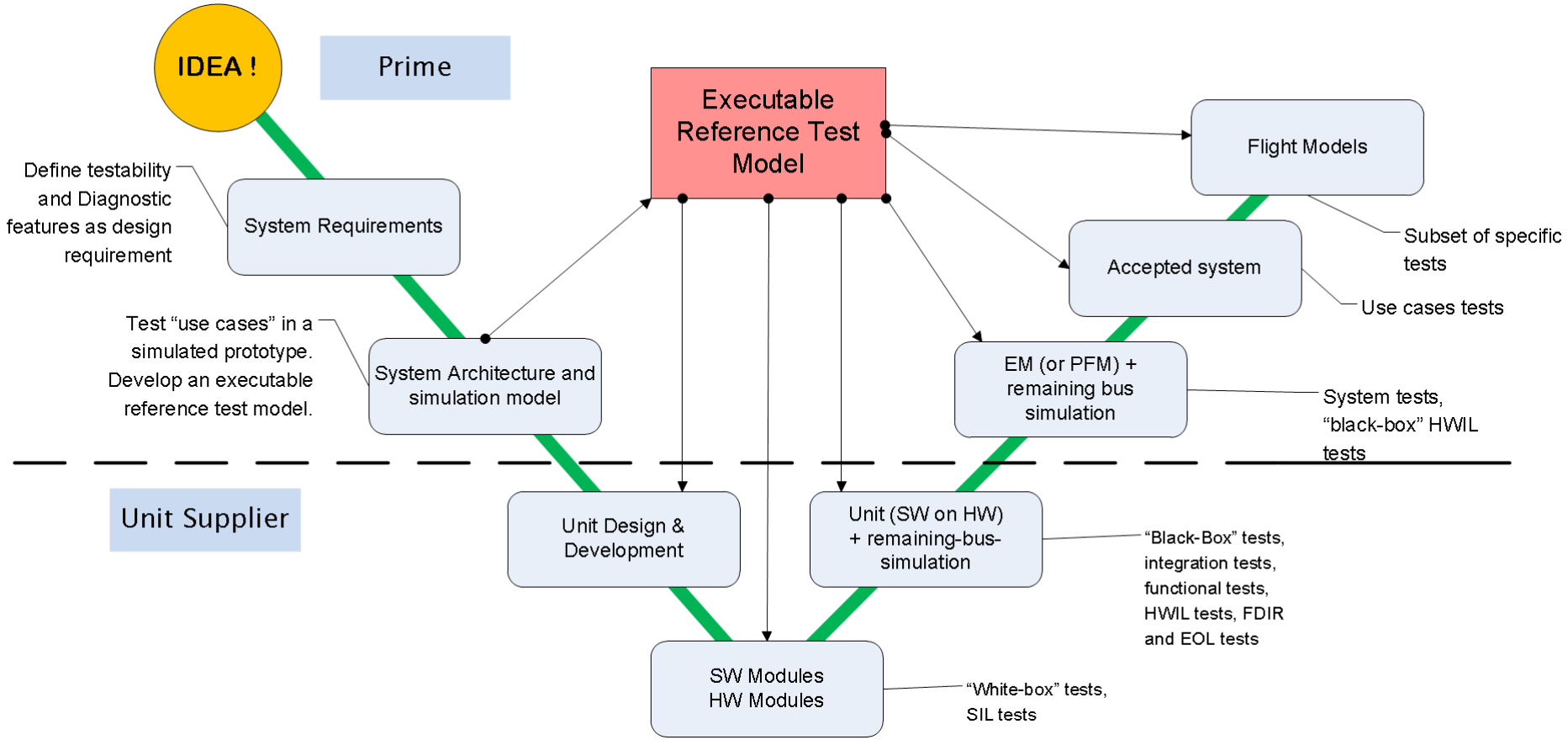
Protocol Layer: what does CANOpen for me ?

- **CANOpen provides a set of useful services for space avionics**
  - Event driven communication
  - Synchronous and Isochronous cyclic communication
  - Error management
  - Determinism (if needed)
  - Master-Slave model for management
  - Server-subscriber model for configuration (SDO)
  - Producer-consumer model for distributed control (PDO)
  - EDS (electronic data sheets), standardized (CiA)
    - With conformance and testing tools !

CAN in space toolbox

- The use of CAN with CANopen based higher layers leads to cost efficient and flexible solutions, but together with a high increase of the electronics' complexity.
- Big complication for space systems is the typical approach of distributed development between primes and several suppliers.
- The consequence has to be a systematic improvement of the development process.
- Project risks are to be reduced by taking testability as a design requirement and by performing the appropriate tests in the very early project phases.

CANOpen-based testability design cycle



Use cases for CAN: the easy one

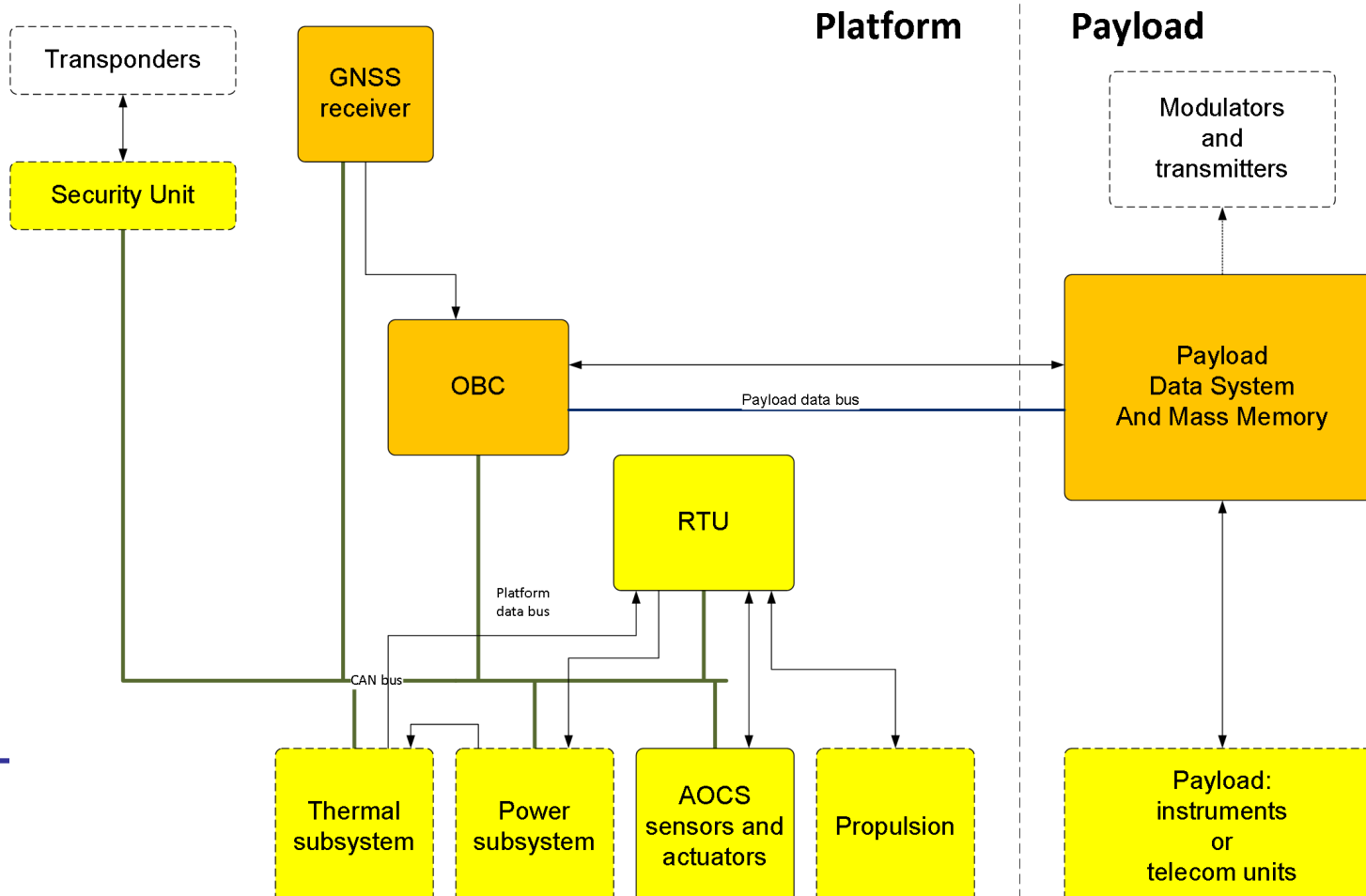
- **Payload Bus**
  - Is in general preferrable to have Payloads C&C separate from platform's
  - CAN data throughput once real time requirements are relaxed is suitable both for command and data transfer (leaving high speed TM transfer to point-to-point IF)
  - CAN can support up to 100 users in a network
  - CAN is a solution for payload-to-payload communication
  - Time distribution via CAN could be adequate for most of the needs (less than 250 ms accuracy with less than 100 ms jitter is achievable)
  - Payloads are often (e.g. scientific missions) designed by groups with limited spacefaring experience.

Use cases for CAN: the obvious one.

- **Command & Control bus**
  - CAN is made for use as C&C.
  - But its widespread adoption is limited by heritage

Use cases for CAN: today's bright idea

- **Essential TM & TC**
  - We have already discussed CAN as ETM bus
  - What if we have also TCs on the same bus ?

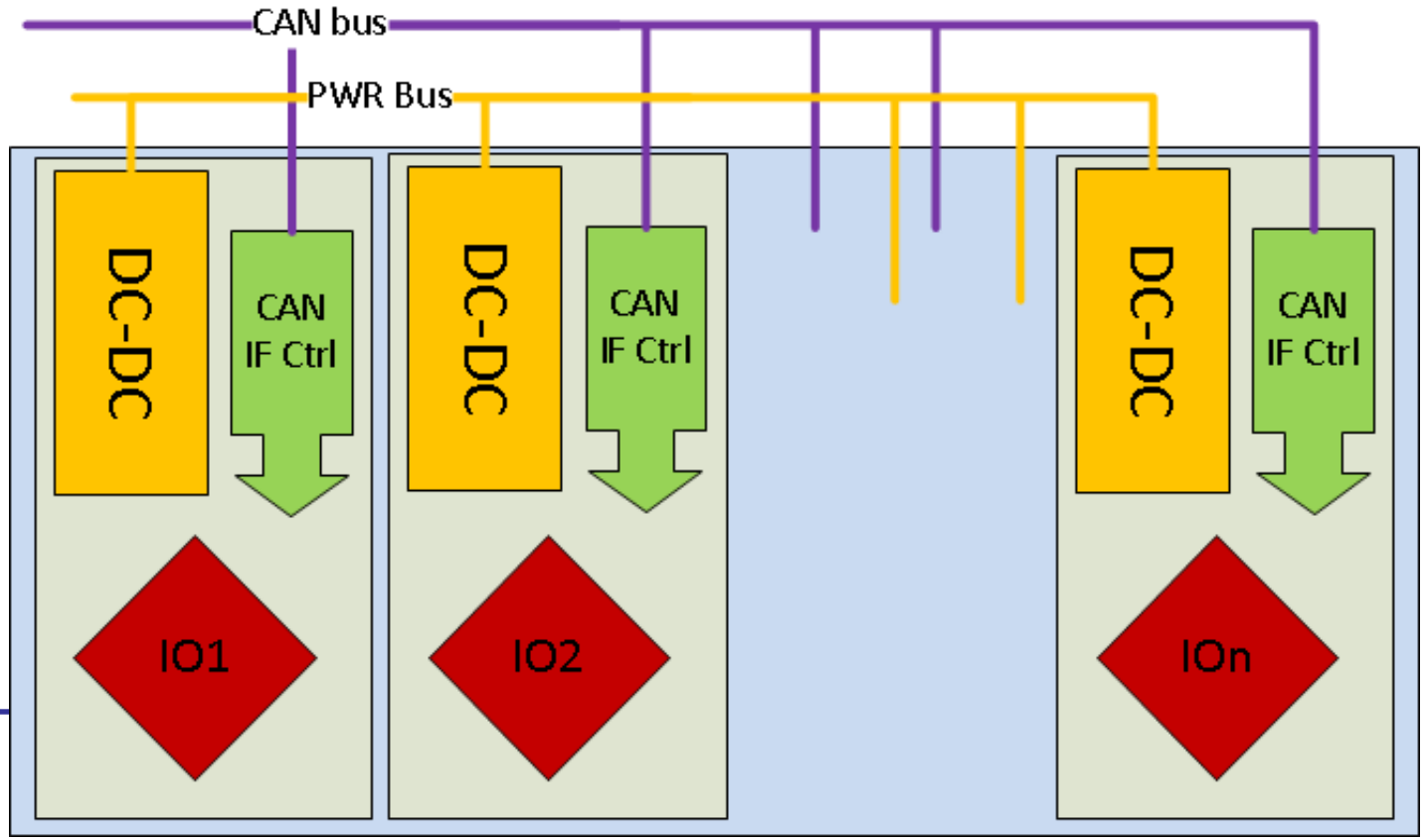


Use cases for CAN

- **CAN as enabler of system level functional redundancy**
  - 'Horizontal', context base data management allow functional redundancy.
  - It is SW rather than HW that limits a unit capability.
- **Stupid example:**
  - If everything is interconnected, nothing stops payload computer to take over functions from OBC in case of unavailability
  - More or less the 'M' as in IMA
- **Less stupid example**
  - Clean space requirement will impose very strict reliability figures on deorbiting functions
  - Instead of adding on cold redundancy (and cost) limited scope functional redundancy can be the answer
  - SSTL is spearheading this concept with its Safeguard Computer

Use cases for CAN: Backplane – Module interconnection

- **A slow differential bus as backplane is insane... but:**
  - Allows fully modular solutions
  - Physical box contention does not mean logical
  - Eases cross-strapping



How to tackle obsolescence with std interfaces

- **How to tackle obsolescence with std interfaces**
  - One of the biggest advantage of standard, digital only CAN interface is that the unit's behaviour can be describes via a single EDS
  - Replacing a unit (from IF point of view) could be as simple as plugging a new one.
  - This could change the Part obsolescence management from simply reactive to fully proactive, concurrent approach.
  - Units and units part could have 'life cycles' rather than having to go towards selective part replacements, LTB and re-qualifications.
- **Unit obsolescence could be separated from platform's**
  - SW reuse will increase
  - The CANOpen-based testability design cycle remains intact
  - Will this drive down unit's costs ?

Some things you can't really stop

- **Trends in space avionic systems**
  - We aim to have 3.3 V only digital electronic (with obvious advantages)
    - Shall RS-485 disappear (at least for low speed interfaces) ?
  - Everybody is going SoC nowadays.
    - We should remember that 1553 interface (and relative conformance tests) was designed for discrete component implementation.
    - With SoC babbling idiot becomes a real error possibility rather than a theoretical one
  - Number of 1553-related NCRs is on the rise. On average all ESA missions have a couple of mayor 1553 NCRs.
    - > 50 % of Validation Tests on 1553 RTs have NC.
- **Heritage & Reuse: gain or burden?**
  - This could be a major discussion point
  - Are we sure at the end that “heritage” is a good argument to keep 1553 at all costs ?

What do we miss

- **ECSS CAN will be accompanied by an implementation handbook.**
- **We still miss a comprehensive validation test plan for ECSS CAN, even if we can inherit from commercial ones**
  - ISO 16845:2004 CAN -- Conformance test plan
  - CANopen conformance test tool from CiA
- **Nevertheless, SAE VTPs (AS411x) failed to educate industry to the 'early testing' habit.**
  - VTPs are often waived for recurrent units, missing the whole point behind them
- **In ESA we soon plan to gain experience on CANOpen-based testability design cycle. Stay Tuned.**

Some thank you words

- **To all participants of the WG (in no particular order)**

  Olivier Notebaert, Maurizio Caramia, Bruno Storni, Chris Taylor, Pierrik Vuilleumier, Peter Roos, Giorgio Magistrati, Luca Bolognino, Max Porciani, Jean Dalenq, Koen Puimege, Massimiliano Calaprice and Stephen Bury (ECSS Secretariat)

Conclusion/Q&A