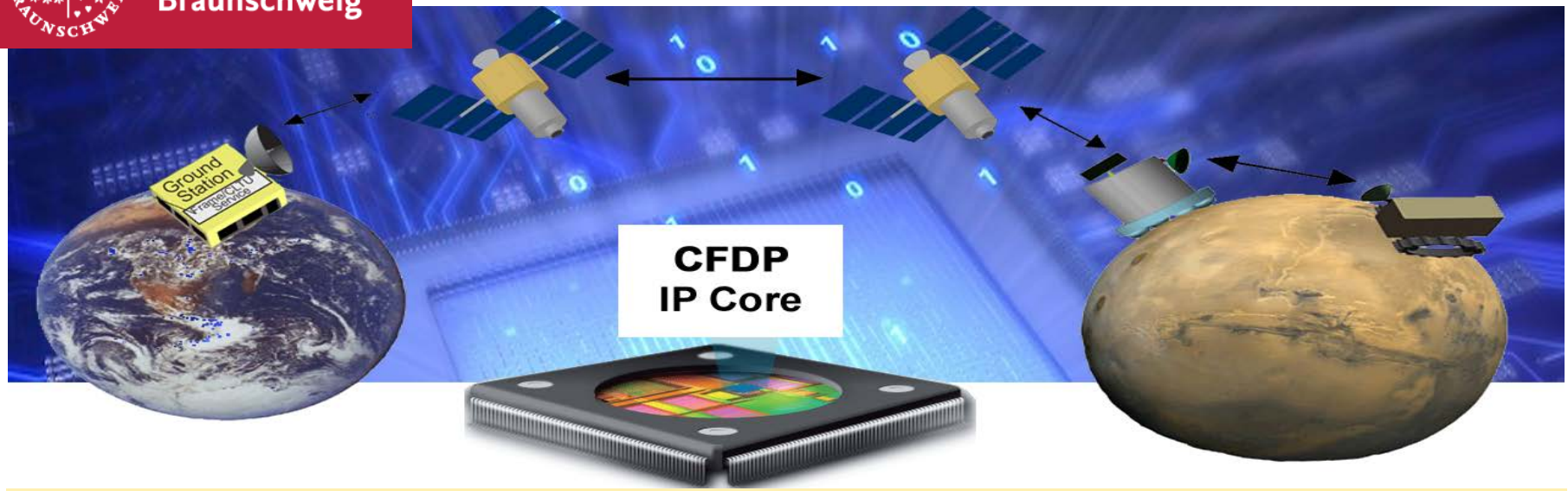
Technische Universität Braunschweig

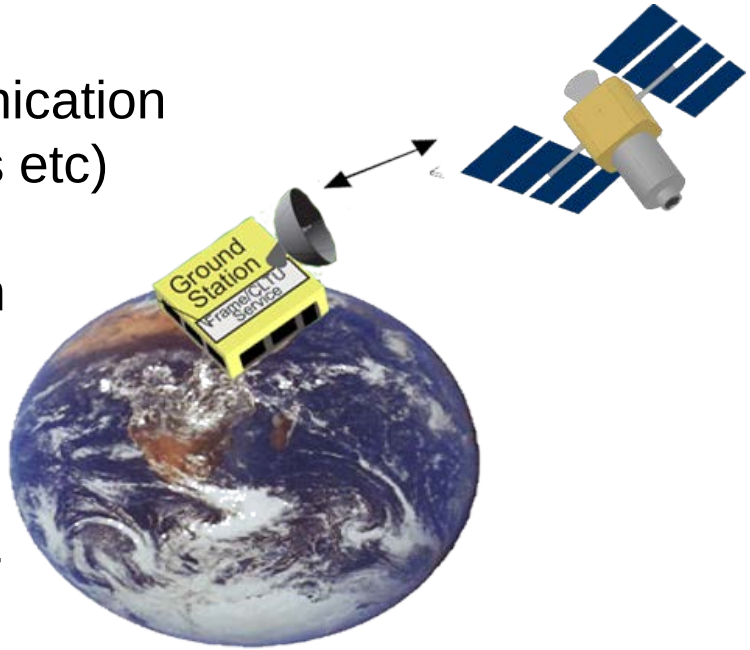


Advanced CCSDS File Delivery Protocol Hardware IP Core

Mladen Berekovic, Sören Michalik, Sönke Michalik, 10th June 2016

The CCSDS File Delivery Protocol (CFDP)

- Delay-tolerant File Delivery Protocol for Space
- Configuration parameters to adapt communication (distance in light-time, timer limits, entity IDs etc)
- Reliable file transfer and remote file system management over interplanetary distances



Features

- Unreliable and reliable sender and receiver
- Delivery of files and user messages
- Reliability: CRC / File Checksum
- maximum 64kB packet size, 4GB file size

External requirements driving the Architecture

Performance

ESA Euclid Mission
- 850Gbit/Day
- 75Mbit/s downlink

Next Generation Mass Memory
- Average 1.5Gbit/s
- Maximum 5Gbit/s

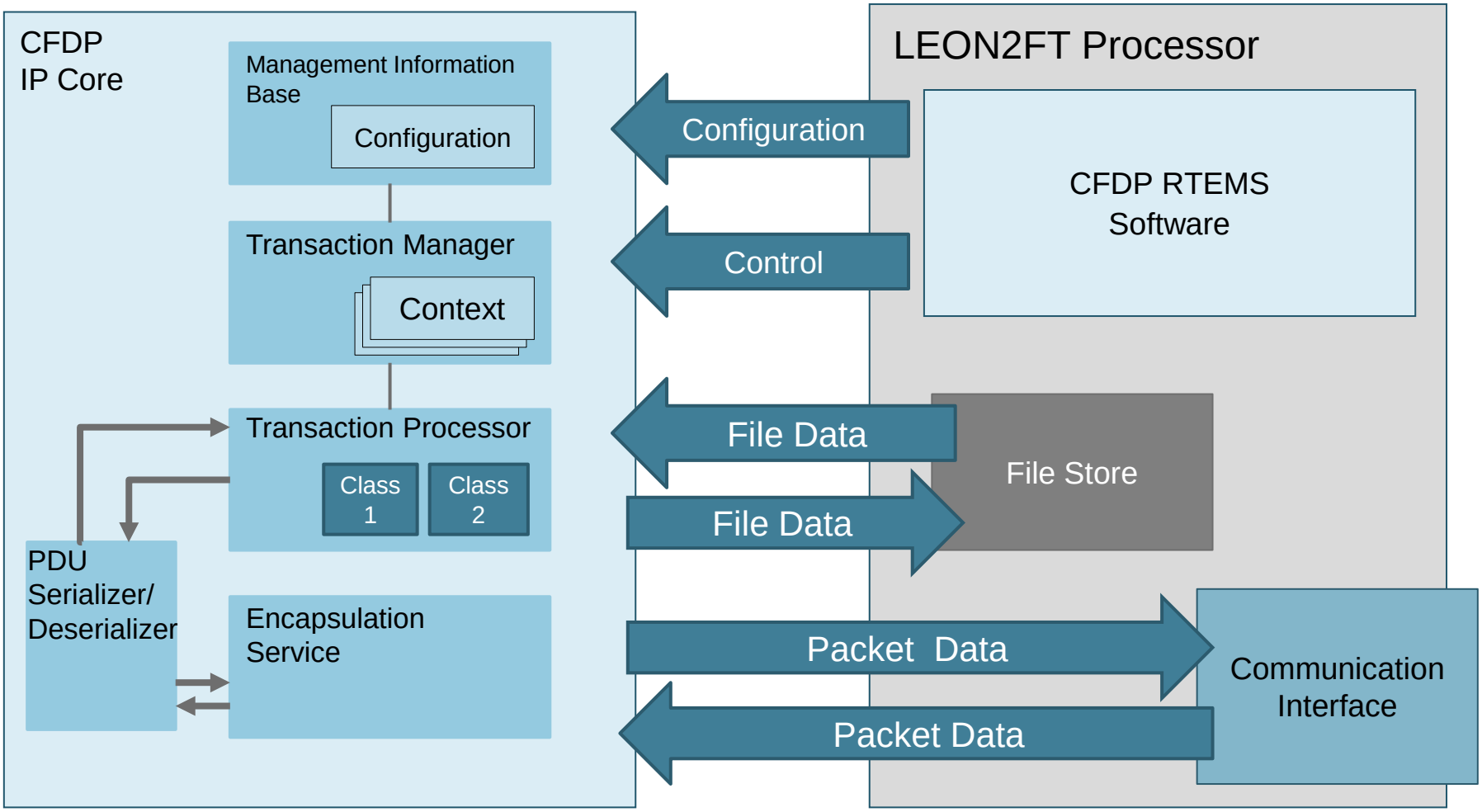
Variability

- High Configurability
- Different CCSDS encapsulation formats
- Generic Filestore Interface
- CAD Tool & commercial library independence

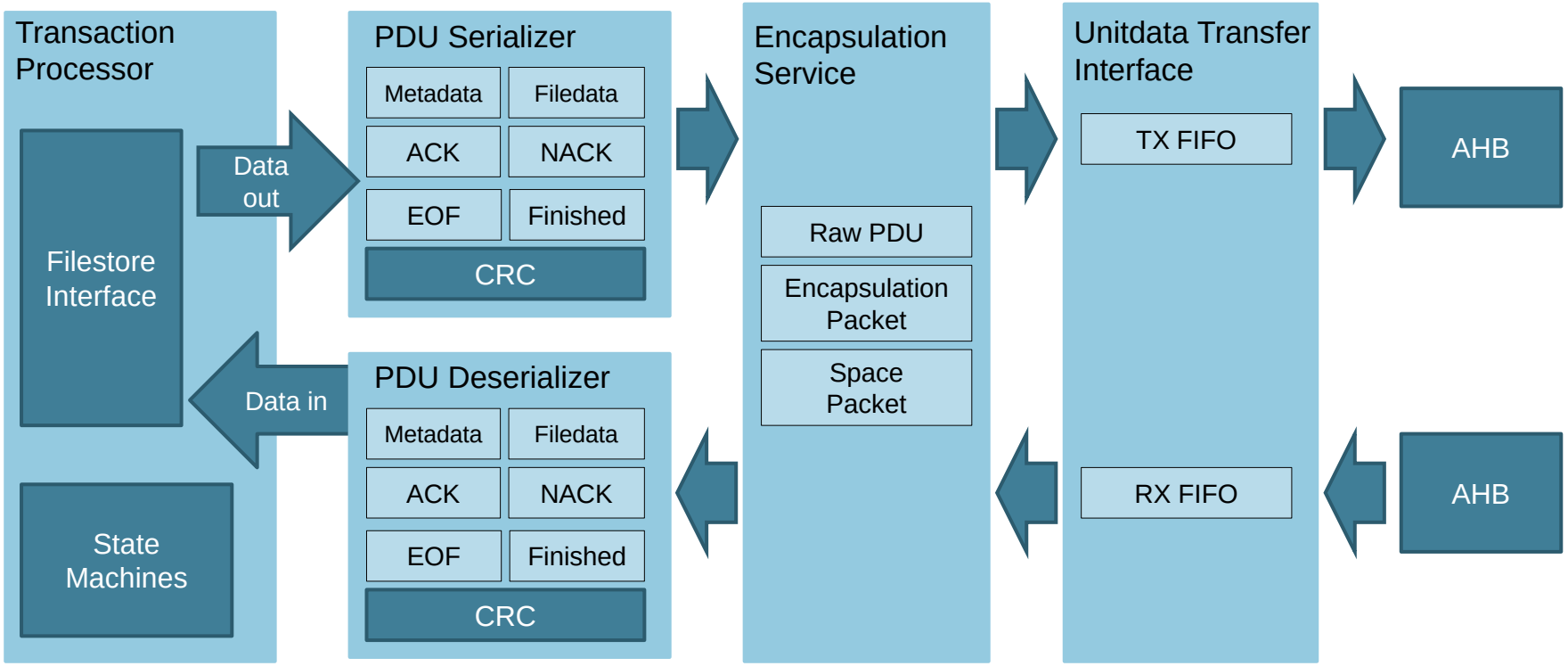
Workflow of the Study

IP Core Definition	• Features Selection • Hardware/Software Partitioning
SystemC IP Core Implementation	• IP-Hardware modeled in SystemC • Tests on SoCRocket Virtual Platform
VHDL IP Core Implementation	• VHDL - Hardware design • SEE Mitigation and LEON3 / LEON2FT System integration • RTEMS software library and driver development
IP Core Validation and Deployment	• Map to a FPGA prototyping board • Verification and Validation tests
Technology Mapping	• Map to FPGA technologies • Map to ASIC technologies

CFDP-IP System Architecture

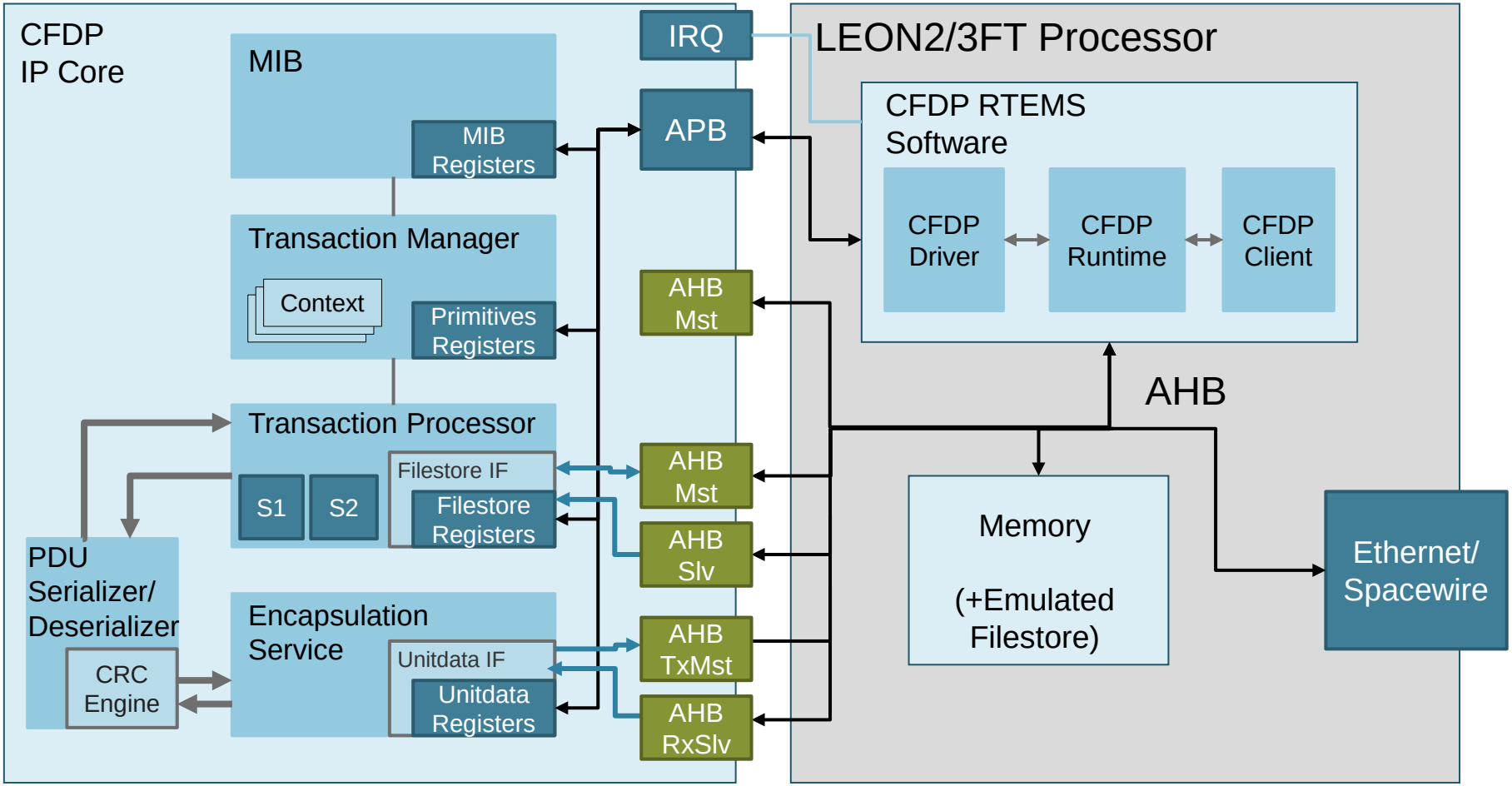


PDU Serialization/Deserialization Pipeline

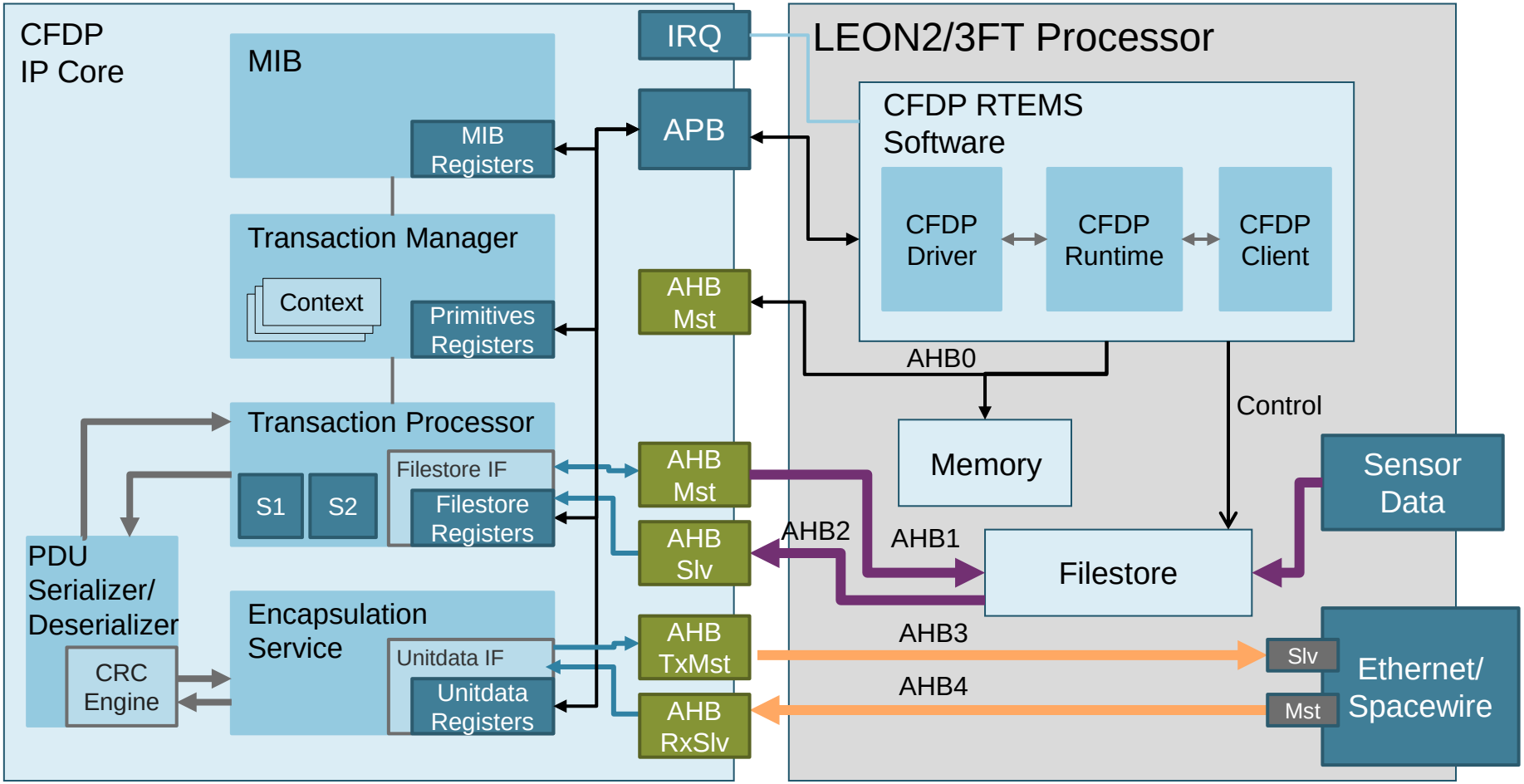


High throughput of processing pipeline:
up to 1Word/cycle → 400 MB/s @ 100MHz

CFDP-IP System Architecture



CFDP-IP System Architecture (streaming config)



Details On The Hardware Architecture

Transaction Manager

- Handles creation, management and scheduling of the active transactions
- Stores transaction contexts
- Timer implementation

Transaction Processor

- Executes transaction using state machine
- Filestore Interface
- PDU serializer / deserializer interface

Details On The Hardware Architecture

Management Information Base

- Global configuration storage accessible through APB registers
  - Local- and remote Entity-IDs
  - Remote Entity UT addresses
  - Light times (distances to remote entities)
  - Timer and counter limits
  - Flags
  - Fault handler information
  - System information and configuration

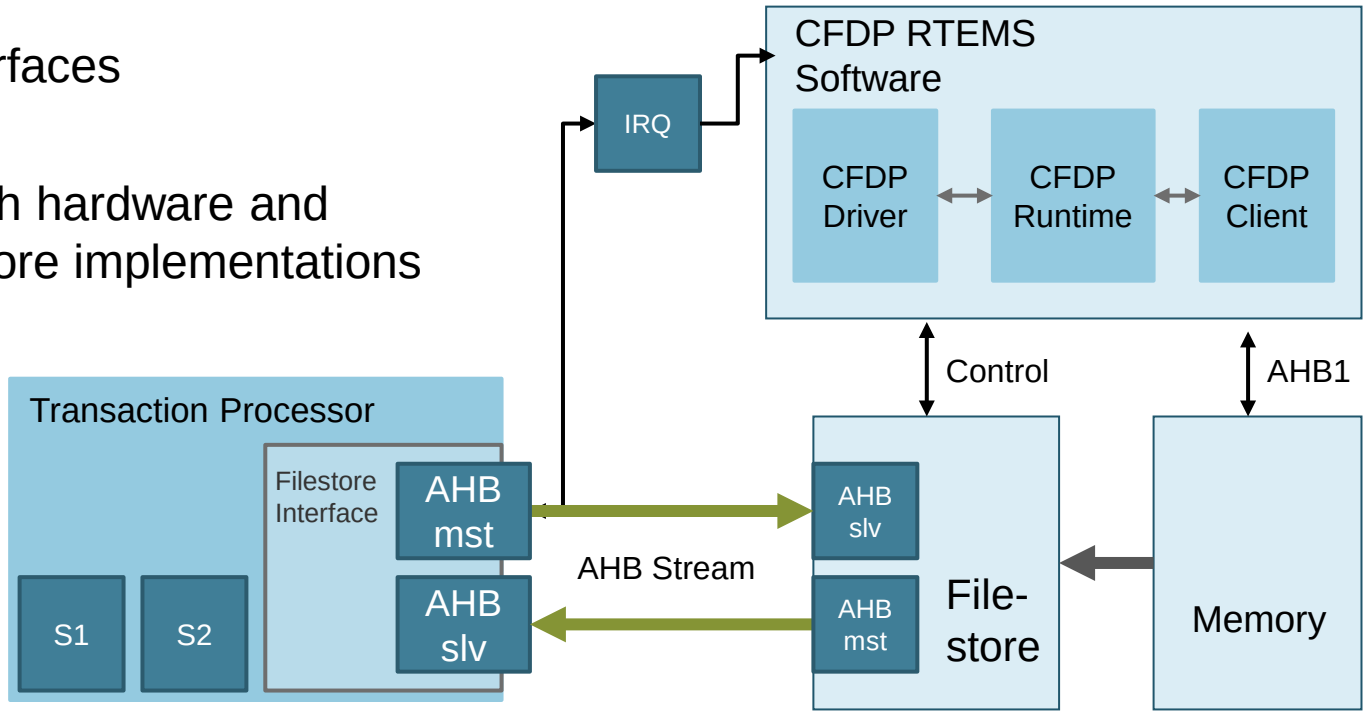
Details On The Hardware Architecture

CRC Engine

- 16-bit standardized CCSDS CRC code at the end of PDUs
- Calculates and attaches checksum for outgoing PDUs
- Validates checksum for incoming PDUs
- 16-bit checksum with the generator polynomial: $G = g(x) = x^{16} + x^{12} + x^5 + 1$

Filestore Interface

- Generic interface protocol
- Streaming interfaces
- Compatible with hardware and software file-store implementations



Filestore Interface

FILESTORE REQUEST FORMAT

The Filestore interfaces uses the following format to request file segment data via AHB:

Create file - request (Request Length = 4):

1st Word	2nd Word
Request Header	File Size

Filedata open/close file - request (Request Length = 4):

1st Word	2nd Word
Request Header	File Handle

Filedata read next segment - request (Request Length = 12):

1st Word	2nd Word	3rd word	4th word
Request Header	File Handle	Seek Location	Segment length

(Seek location = file position offset in bytes)

Filedata write - request (Request length = Segment length + 12):

1st Word	2nd Word	3rd word	4th word	5th word
Request Header	File Handle	File position	Segment length	Data

Filestore Interface

FILESTORE RESPONSE FORMAT

The Filestore interfaces uses the following format to response to requests via AHB:

Open File / Close File / write next segment - response (Response Length = 0):

1st Word
Response Header

Create File - response (Response Length = 4):

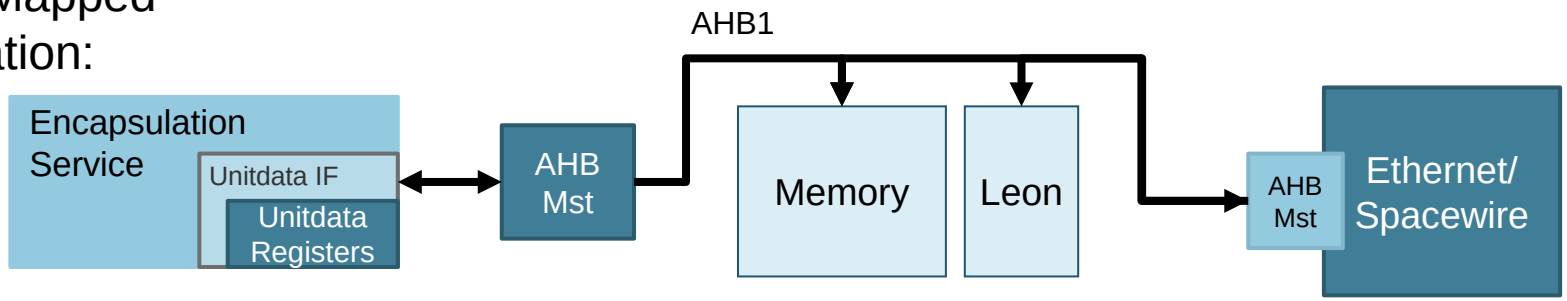
1st Word	2nd Word
Response Header	File Handle

Read next segment - response (Response Length = segment length):

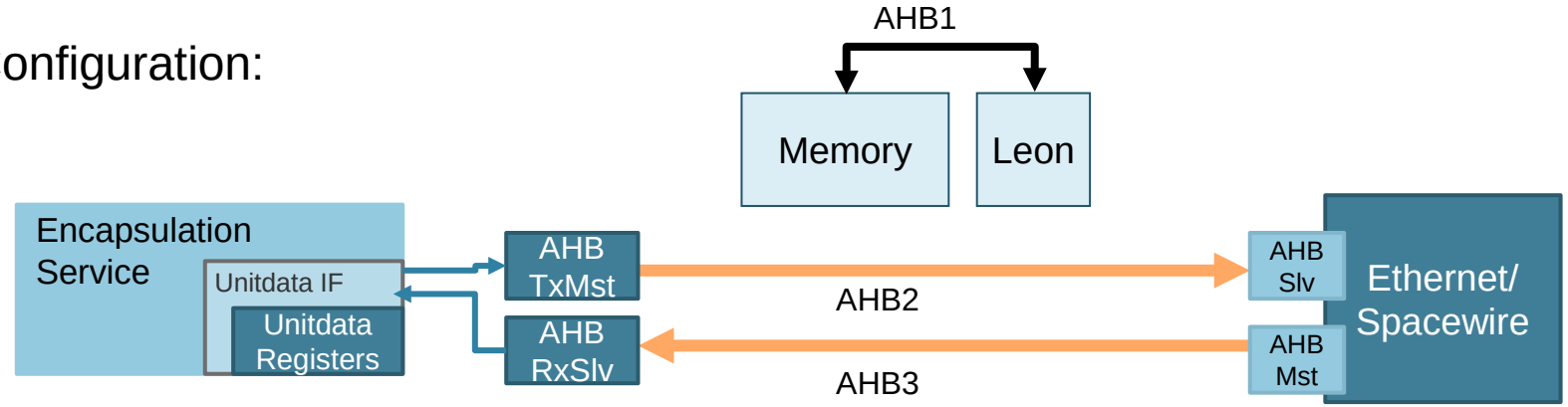
1st Word	2nd Word	3rd word	4th word	5th word
Response Header	Data	Data	...	Data

Unitdata Transfer Interface

Memory Mapped Configuration:

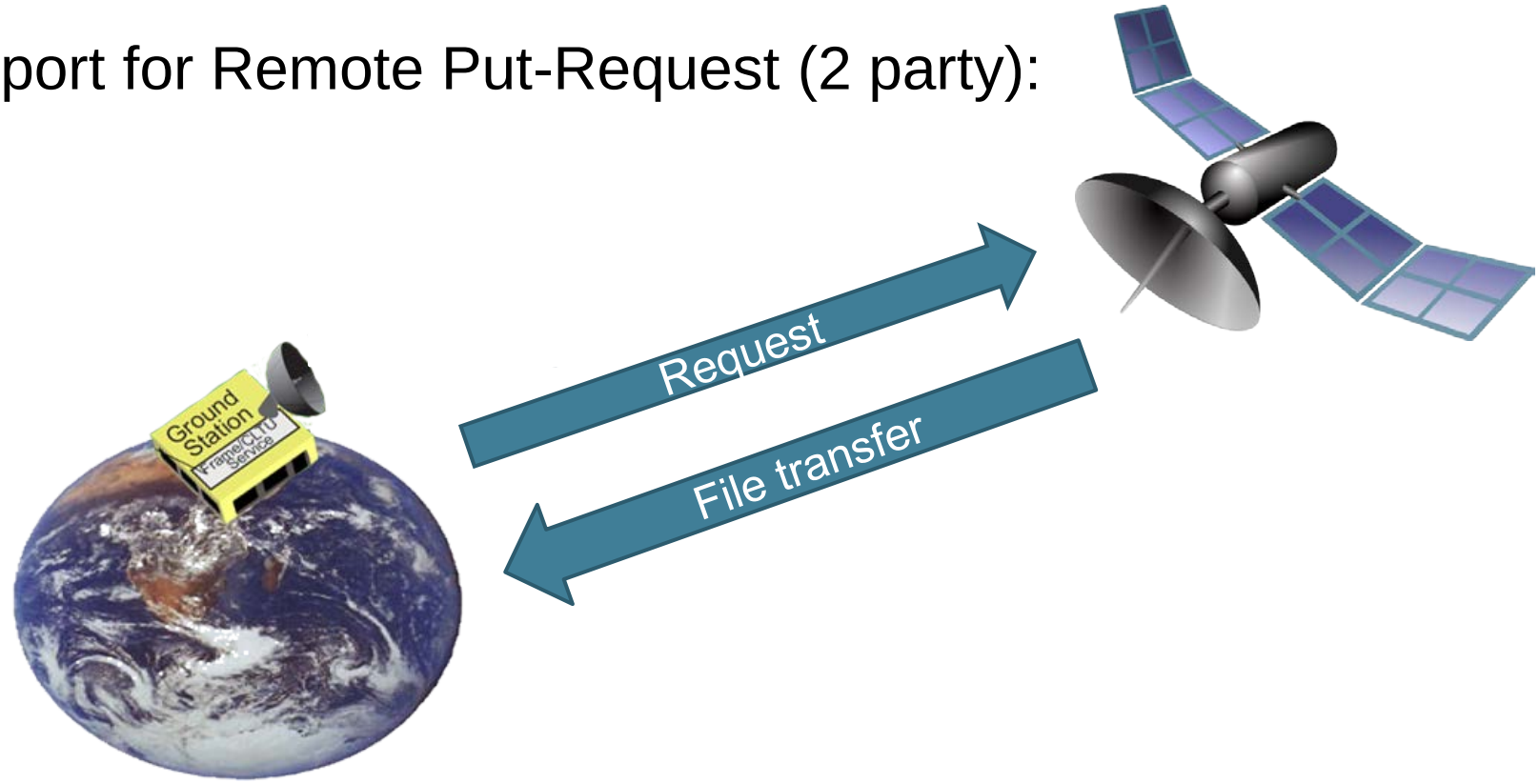


Stream Configuration:



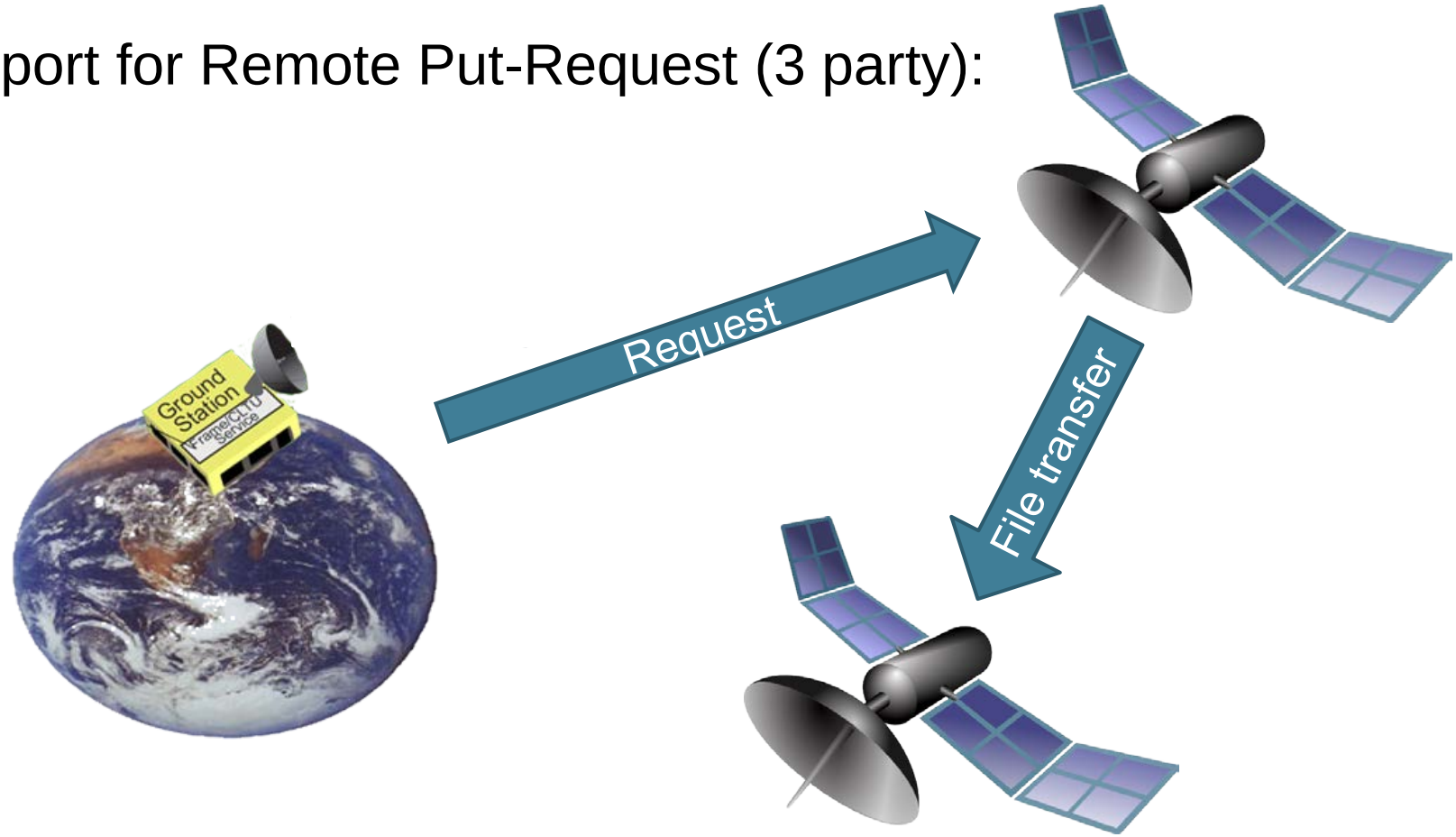
Advanced protocol features: remote operations

- Support for Remote Put-Request (2 party):



Advanced protocol features: remote operations

- Support for Remote Put-Request (3 party):



SystemC IP Development

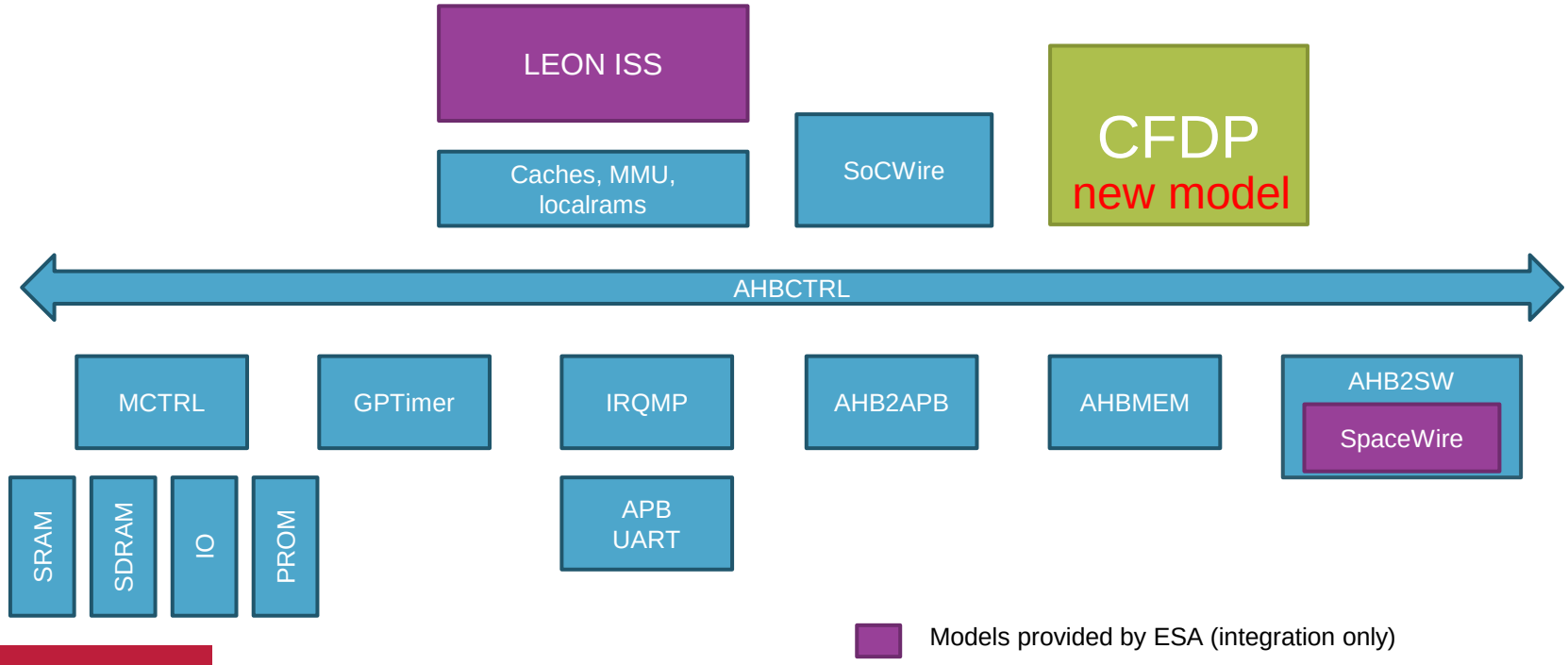
- Early software and driver development
- Evaluation of SystemC-IP architecture and interface configuration during the development
- Measuring performance
- Modeled in SoC-Rocket virtual processor platform

Design Space Exploration

Find an optimal configuration of the IP-Core architecture and interfaces to improve the maximum performance

The SoCRocket Virtual Platform
SystemC/TLM2 Simulation IP + Design Infrastructure

- Simulation model for GRLIB core components
- Models featuring different operating point:
  - Extensive Design Space Exploration (high accuracy / medium speed)
  - Low-Level SW Development & Early Exploration (medium accuracy / high speed)



SoCRocket – Example use cases and features

HW IP Development:

- Infrastructure for quickly building new components
  e.g. Bus-Interfaces, Register Container inherited from library base-classes
- Integration of RTL components with custom transactors

Low-Level SW Development:

- Register/bit true system environment
- SW Flow equivalent to GRLIB: including boot code
- Support for real-time OS: RTEMS, µC OS, …

System Exploration:

- Runtime reconf. – all parameters can be change without compilation
- Lots of debugging and analysis features

Verification and Validation

- The IP was verified and validated with automated tests using the ESE CFDP reference software and additional VHDL module testbenches

- The verification and validation process followed the test series of the yellow-book test specification:

"CCSDS CFDP – Notebook of common interagency test for core procedures"

  - F1 - Unacknowledged and Acknowledged modes, canceling an ongoing transaction, user messages
  - F2 - Acknowledged mode, including automatic recovery from dropping of the PDUs, timer tests
  - F3 - Check two party Remote Put and proxy operation
  - F4 – NAK modes, suspend and resume, CRC tests
  - F5 – two party remote operations

IP Performance

- The maximum throughput was measured for different file sizes (32bytes, 5KByte) with a segment size of 512 Bytes on Virtex 5 FPGA
- Throughput highly depends on communication interface and File-store implementation
- Measurements for unreliable (Class 1) and reliable (Class2) file transfers

Configuration	Downlink Throughput	Uplink Throughput	Loopback Throughput
Class 1, big file size	2079,8 Mbit/s	1860,5 Mbit/s	1964,0 Mbit/s
Class 2, big file size	1641,0 Mbit/s	1715,4 Mbit/s	1766,3 Mbit/s

TABLE 5 – IP CORE PERFORMANCE (HW FILESTORE, 5KB File Size)

IP Performance II

- The CFDP-IP was mapped to various technologies (ASIC and FPGA):
  - Virtex XC5VLX110T - 60MHz (incl. Leon3)
  - DARE 0.18µm - 150MHz
  - TSMC 65nm - 450MHz

- IP logic area is highly configurable using parameters:
  - Downlink only, Uplink only, Uplink + Downlink
  - File segment length
  - FIFO sizes
  - Number of parallel transactions

Demonstration Video

Summary

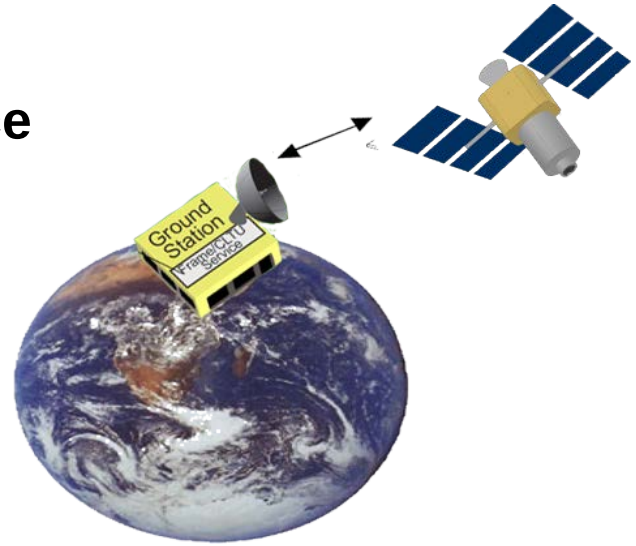
CFDP VHDL IP implementation

Delay-tolerant file delivery protocol for space

Highly configurable

High throughput streaming interfaces

→ reduced memory accesses
→ direct interface option
→ maximum AHB throughput:
3,2Gbit/s = 400MB/s @ 100MHz



Compatible with hardware and software based Filestore implementations