School of Engineering
InES Institute of Embedded Systems

Blueprint for AI Execution in Space: Beyond the CPU

Bridging Reliability and High Performance in AI Execution with Radiation-Hardened Co-Processing.

Pablo Ghiglino, Mandar Harshe, Rafael Tordoya and **Hans Dermot Doran**

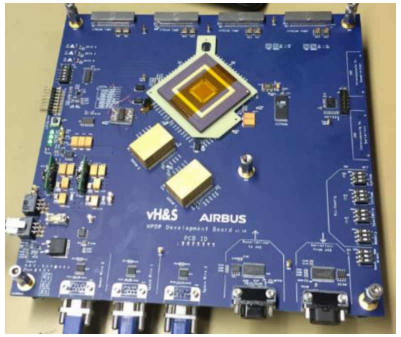
Agenda

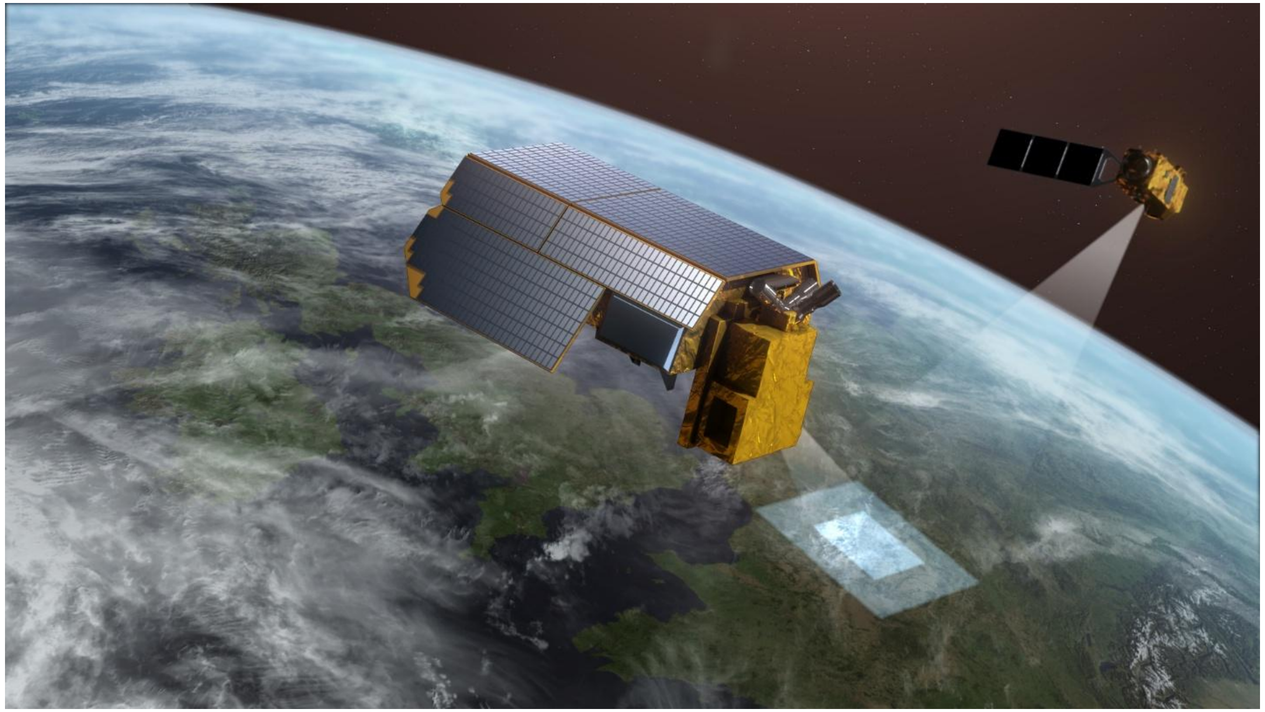
- Motivation
- Optimized AI-Execution Framework – Klepsydra AI
- HPDP Architecture
- Verification and Validation
- Results
- Conclusions
- Perspectives

Motivation

- **Agenda:**

- HPDP development started mid 2010's
- The HPDP is already part of ESA's mission TRUTHS (Launch in 2030)
- The development of HPDP has been initiated by the European Space Agency (ESA) and DLR to address the need for a flexible and re-programmable high-performance data processor.
- It is being implemented in the 65nm radiation hardened technology of ST Microelectronics (C65SPACE).

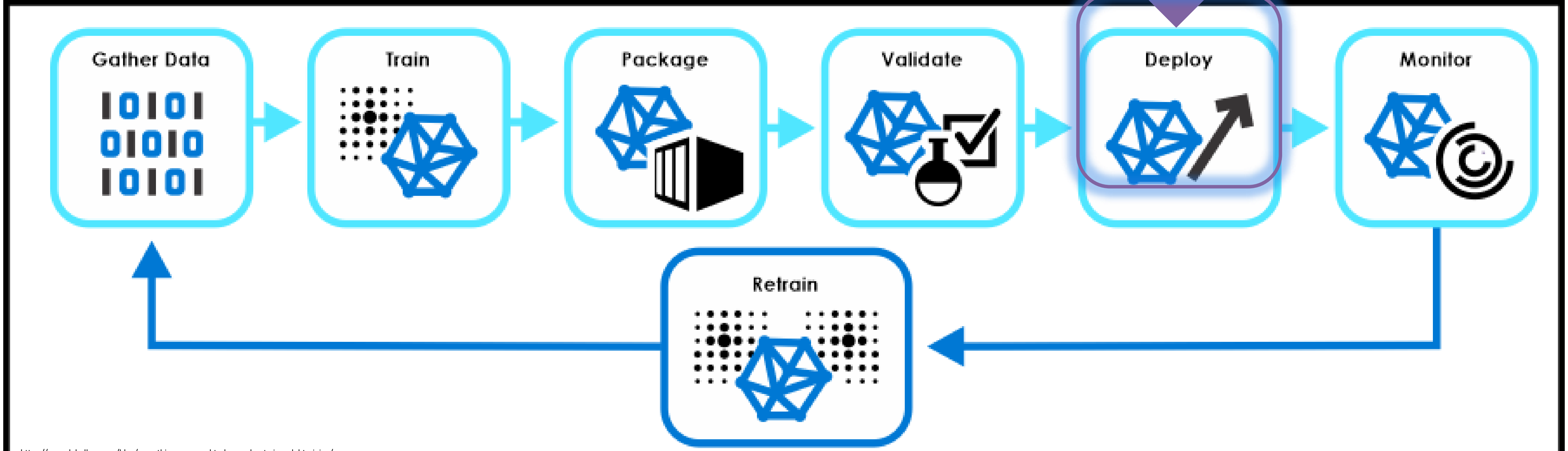




ESA TRUTHS Mission.
Image source: https://www.esa.int/Applications/Observing_the_Earth/TRUTHS

Motivation

MANDALA is aimed to AI deployment to the onboard processor



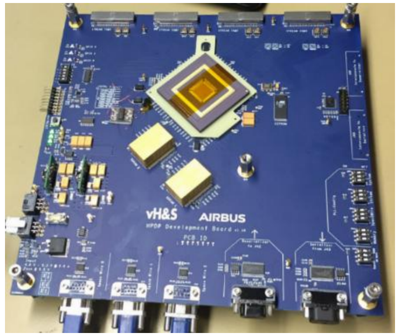
https://www.labellerr.com/blog/everything-you-need-to-know-about-ai-model-training/

Motivation

- Agenda:
  - **Motivation**
  - Optimized AI-Execution Framework
  - HPDP Architecture
  - Verification and Validation
  - Results
  - Conclusions
  - Perspectives

- To address the challenges posed by AI in space we use radiation-hardened hardware.
- Well-known solutions in this domain include the Gaisler series of processors.
  - **Dependable Processors:**
    - E.g. GR740 (LEON4), GR765 (RISC-V).
    - Local AI inference.
    - Proven reliability, RTEMS 6 SMP support.
    - Application-specific requirement (e.g., rad-hard).

- Computation expense of AI is such that available computing power can be consumed by AI calculations – The concept of an offloadable "co-processor" becomes attractive
  - **Streaming Co-Processor**:
    - E.g. High-Performance Data Processor (HPDP).
    - Executes compute-intensive tasks.
    - Rad-hard, parallel, dataflow-oriented architecture.
    - Part of the forthcoming mission TRUTH

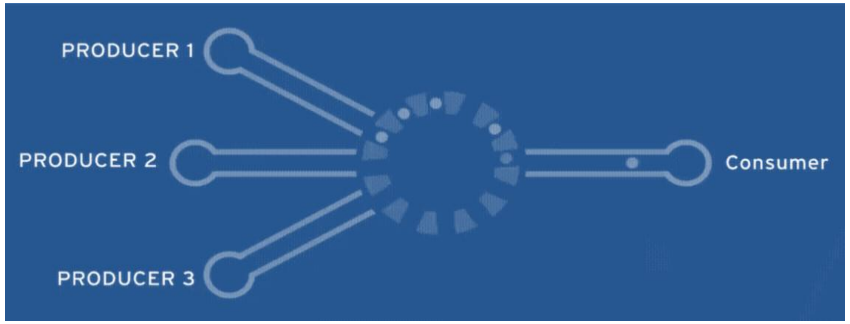


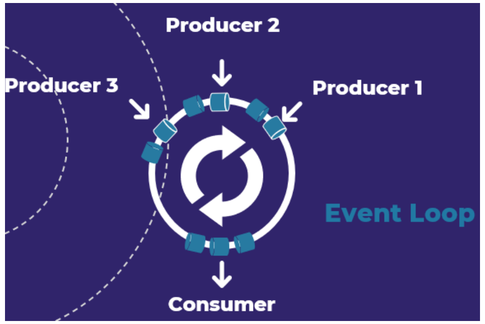
Optimized Execution Framework

- **Agenda**:
  - Motivation
  - **Optimized AI-Execution Framework**
  - HPDP Architecture
  - Verification and Validation
  - Results
  - Conclusions
  - Perspectives

- **Traditional frameworks** (PyTorch/TensorFlow):
  - Optimized for prototyping, not embedded or dependable systems.
  - Introduce **latency** -> centralized execution control and blocking operations.
  - **Unsuitable** for **deterministic**, low-power inference.

- **Klepsydra AI framework** (Lock-free, non-blocking design):
  - Optimized for embedded systems.
  - Parallel, dataflow-oriented architecture. Low CPU overhead.
  - Supports x86/x86_64, ARM, RISC-V, SPARCV8.
  - Supports RTEMS 6 SMP / bare-metal / Linux.
  - Moving towards IEC 61508 compliance.

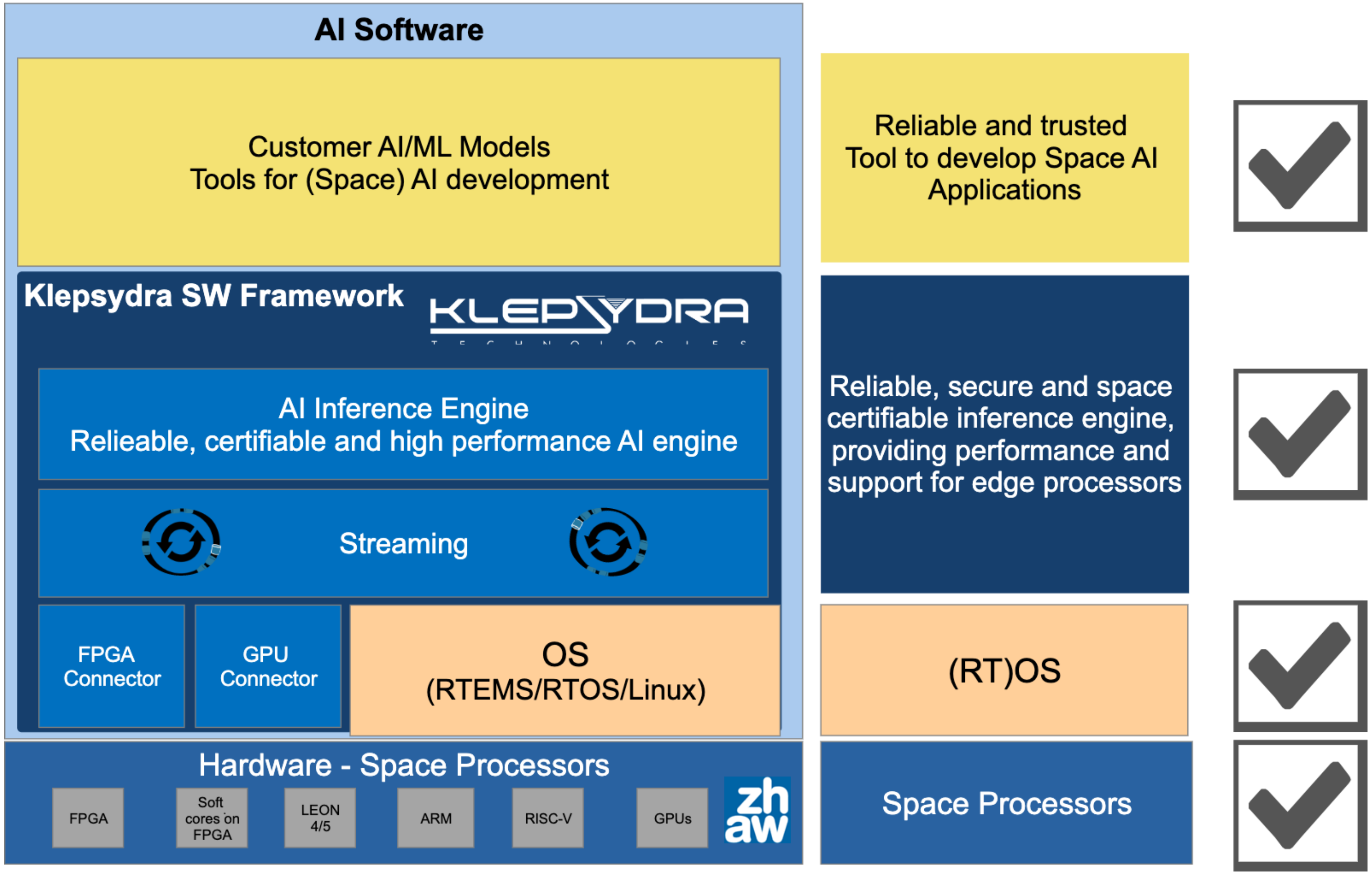




- Klepsydra AI inference execution:
  - Executes **inference locally** on the main processor.
  - **Offloads** to a streaming **co-processor** when available.

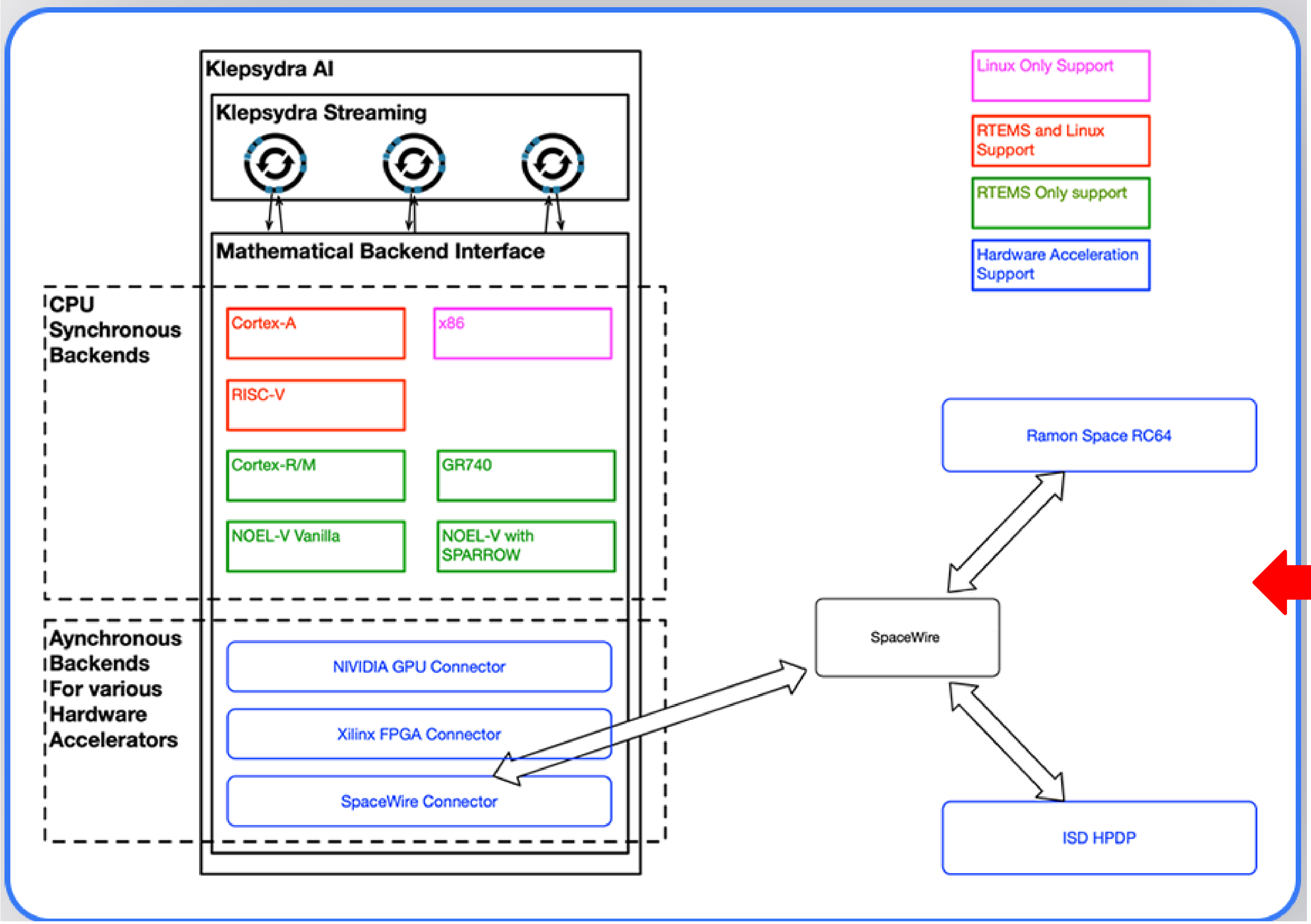
Optimized Execution Framework

- **Agenda:**
  - Motivation
  - **Optimized AI-Execution Framework**
  - HPDP Architecture
  - Verification and Validation
  - Results
  - Conclusions
  - Perspectives



Optimized Execution Framework

- **Agenda:**
  - Motivation
  - **Optimized AI-Execution Framework**
  - HPDP Architecture
  - Verification and Validation
  - Results
  - Conclusions
  - Perspectives



- Support to a large number of processors, operating systems and hardware accelerators:
  - Unified API
  - Unified use of SDO
  - Unified use of KITL
  - Supported by ESA and other European
  - organisations

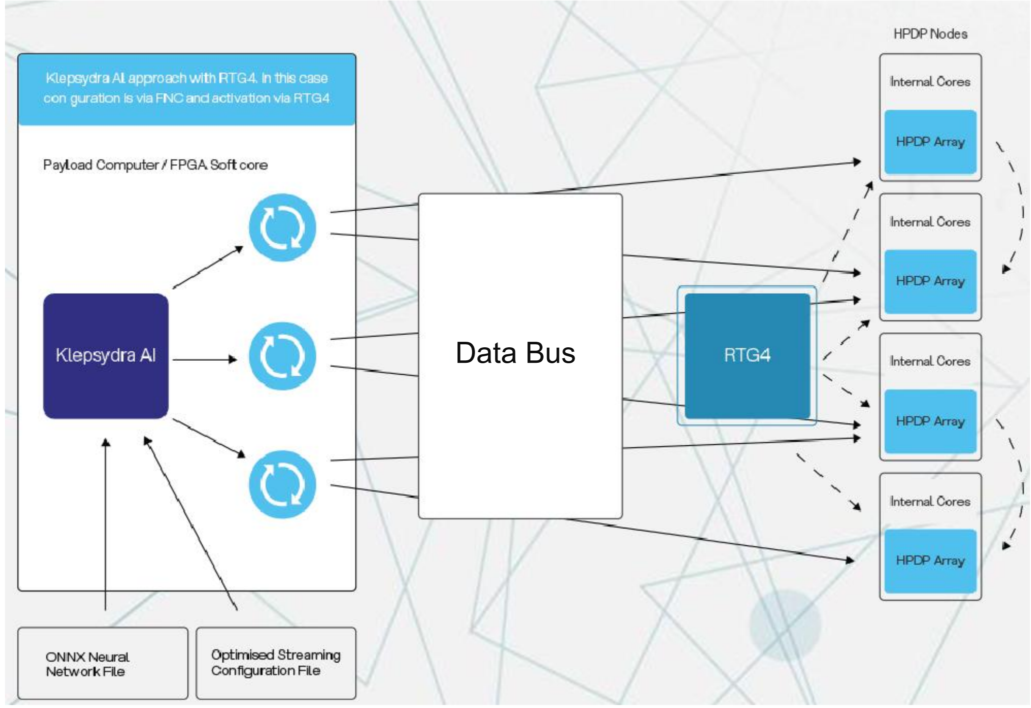
AI execution system

- **Agenda**:
  - Motivation
  - **Optimized AI-Execution Framework**
  - HPDP Architecture
  - Verification and Validation
  - Results
  - Conclusions
  - Perspectives

- System design for **dependable AI execution** integrates:
  - Klepsydra AI framework: **Coordinates data flow** and **model execution.**
    - Orchestration.
    - Math-backend platform.

- Execution **only in Local Processor**:
  - Inference runs entirely on the radiation-hardened CPU.
  - RTEMS6 SMP:
    - Deterministic scheduling.

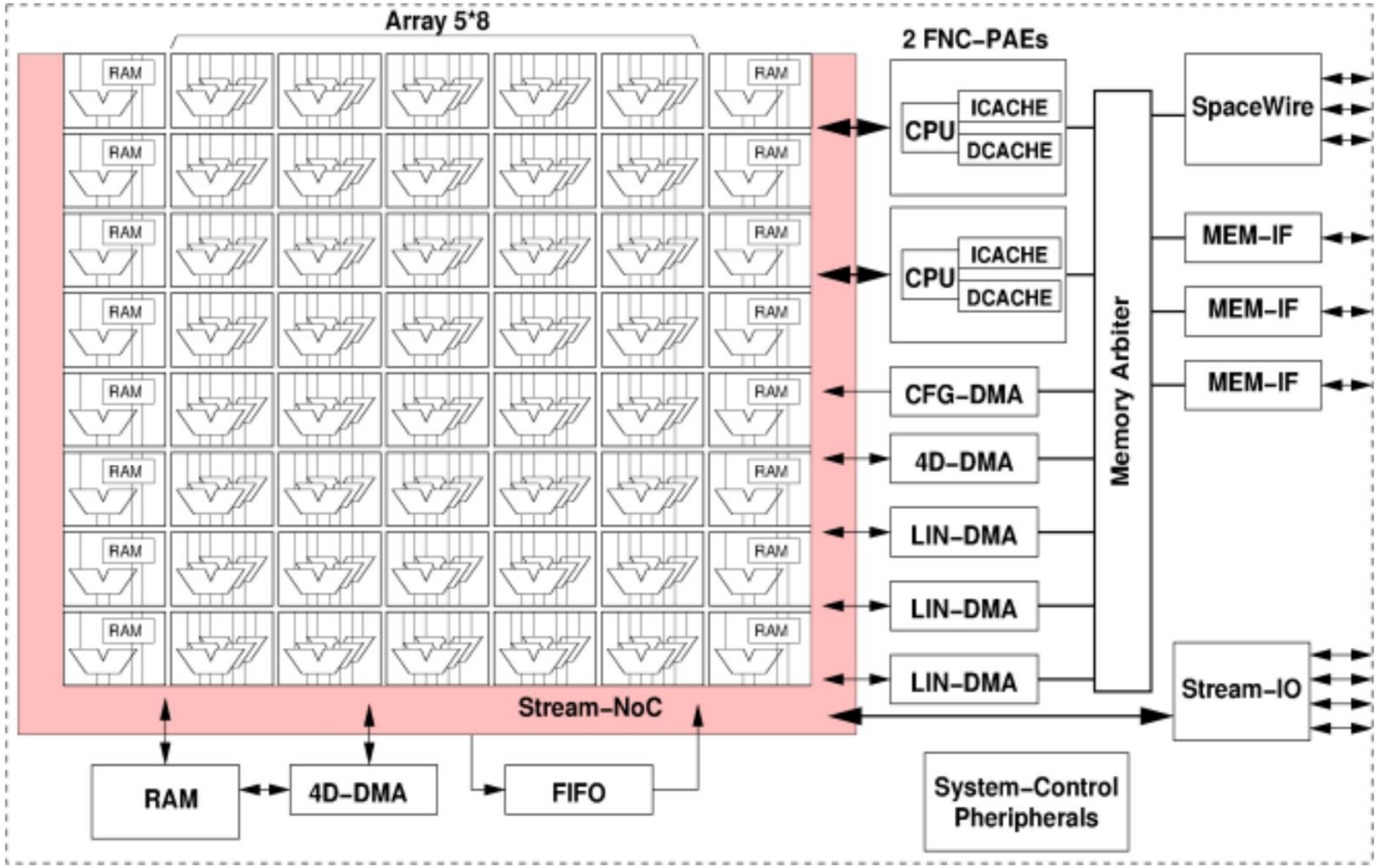
- **Execution with Streaming Co-Processor**:
  - Offload **computationally intensive** operations.
  - Payload CPU manages **orchestration** and light operations.
  - Offloaded operations executed with **predictable timing** and low power.



HPDP Architecture

- **Agenda**:
  - Motivation
  - Optimized AI-Execution Framework
  - **HPDP Architecture**
  - Verification and Validation
  - Results
  - Conclusions
  - Perspectives

- The HPDP is composed by the following elements



HPDP architecture

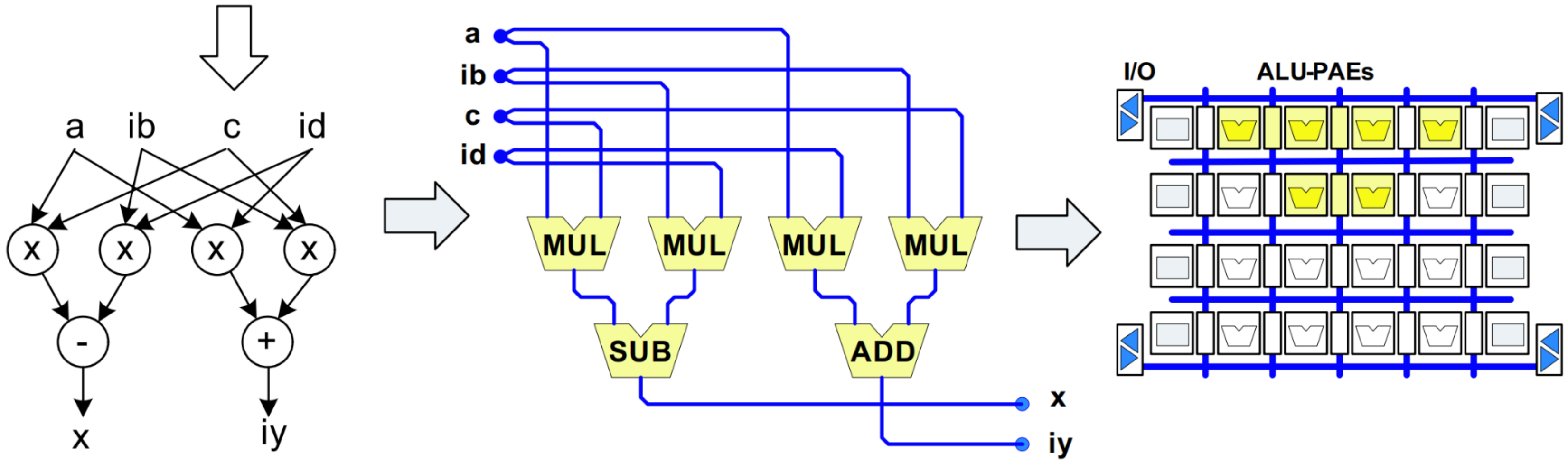
[1] https://indico.esa.int/event/225/contributions/4251/attachments/3379/5380/OBDP2019-paper-Airbus_Helfers_HPDP-40_High_Performance_Data_Processor_A_New_Generation_Space_Processor_in_Demonstration.pdf

XPP Array Dataflow

- **Agenda**:
  - Motivation
  - Optimized AI-Execution Framework
  - **HPDP Architecture**
  - Verification and Validation
  - Results
  - Conclusions
  - Perspectives

- Each PAE operates in parallel and process the data as soon as it arrives
- Data can be processed as a continuous data stream that flows through XPP array
  - Low latency and optimized throughput
  - Suitable for applications that benefit from parallel execution

$$x + iy = (a+ib) * (c+id) = (ac - bd) + i\,(ad + bc)$$



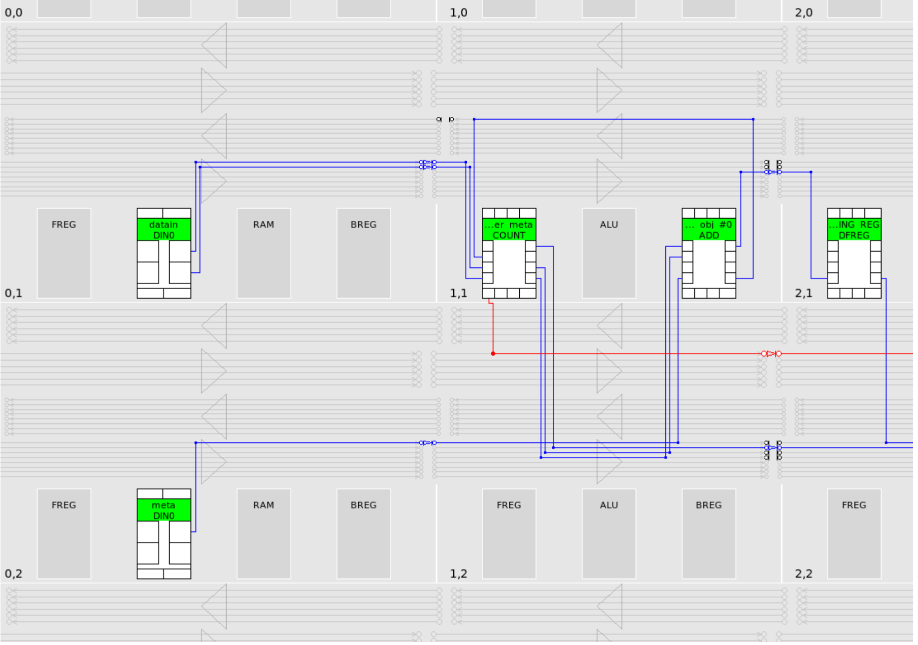
Mapping a complex multiplication to XPP array

[1] https://courses.cs.washington.edu/courses/cse591n/06au/papers/XPP-III_overview_WP.pdf

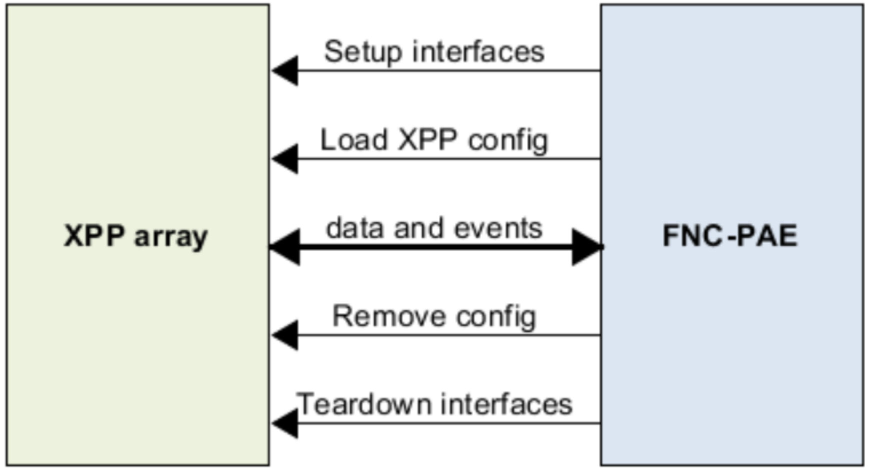
HPDP Configuration

- **Agenda**:
  - Motivation
  - Optimized AI-Execution Framework
  - **HPDP Architecture**
  - Verification and Validation
  - Results
  - Conclusions
  - Perspectives

- Two main components should be programmed:
  - FNC-PAE: programmed in C
  - XPP array: programmed using NML



XPP array configuration example

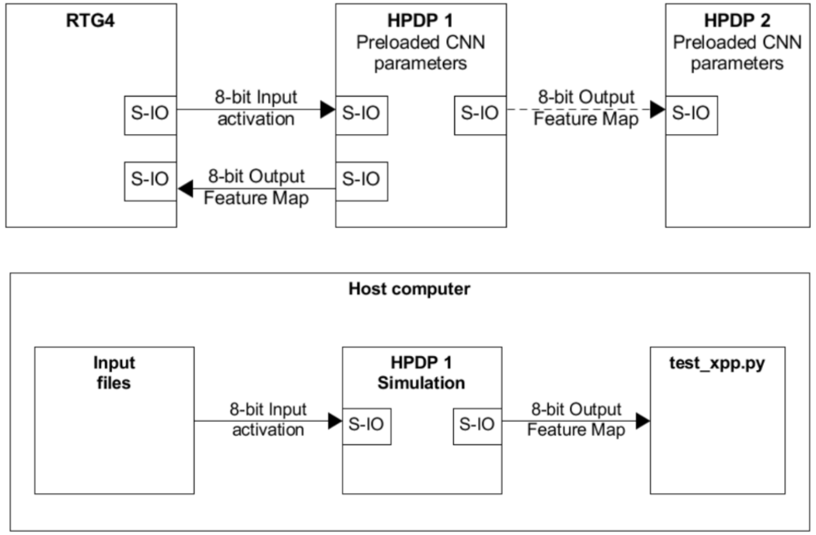


FNC-PAE and XPP array communication

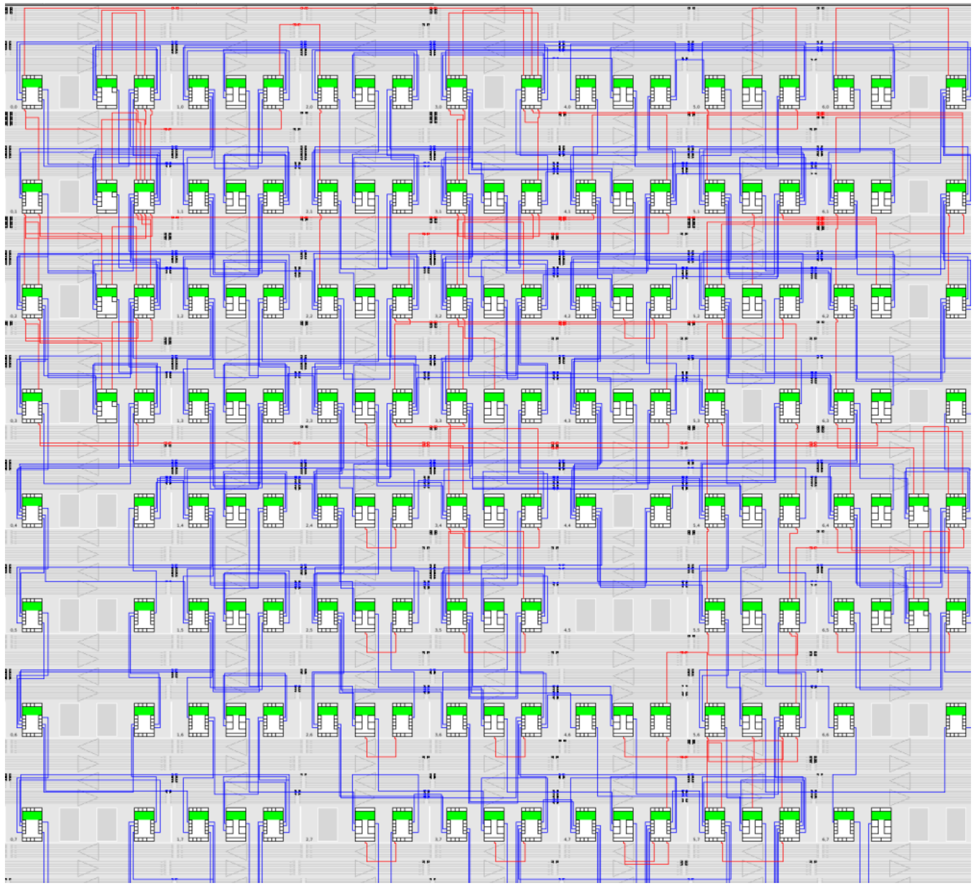
HPDP as a Co-processor for CNN

- **Agenda:**
  - Motivation
  - Optimized AI-Execution Framework
  - **HPDP Architecture**
  - Verification and Validation
  - Results
  - Conclusions
  - Perspectives

- Iterative process of validation and refinement:
  - Implementation progressively improved
  - Stable version:
    - Convolution + Re-quantization
    - Rely solely on input parameters
  - The implemented solution supports multiple kernel sizes



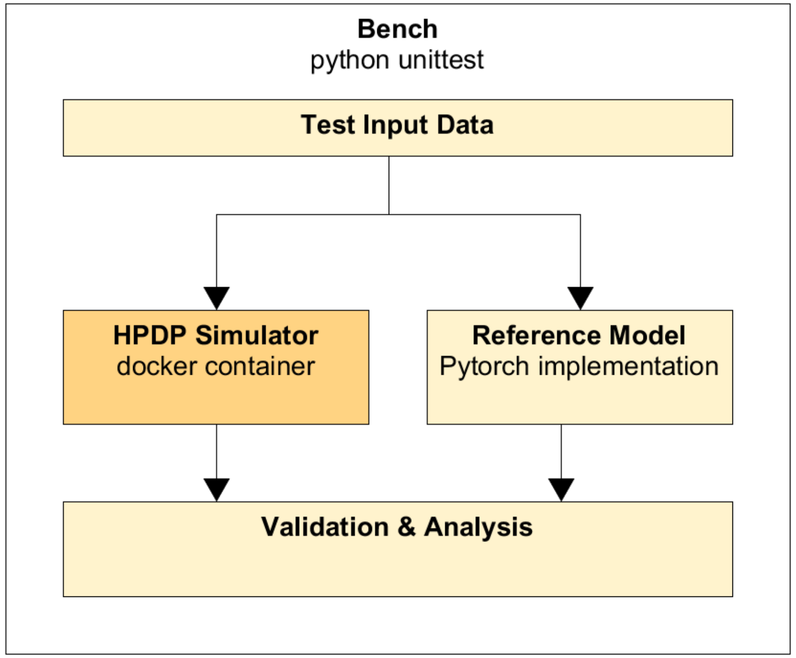
Convolution and Re-quantization operations in HPDP



PAE usage for Convolution + Re-quantization

Verification and Validation

- Debug
  - XPP debugger XDBG
- Verification
  - Unit-test mathematic implementation
  - HPDP simulator -> cycle-accurate representation.
    - Results from simulation closely reflects HW results.
- Validation
  - Test against a golden model
  - Test against a battery of kernel sizes
  - Systematic on-target validation

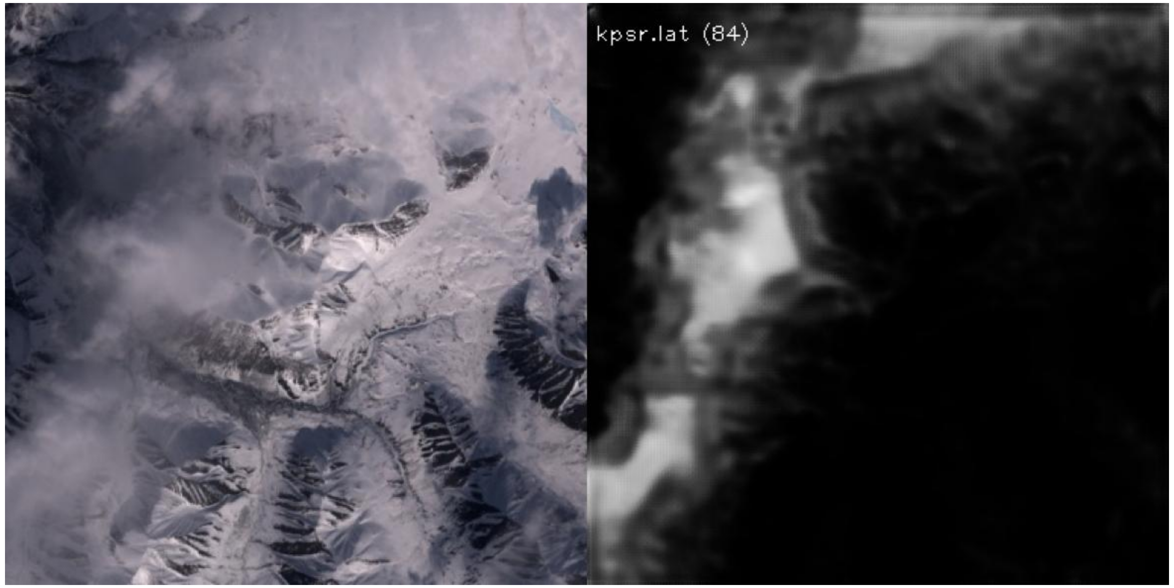


Validation methodology diagram

Results

- **Agenda:**
  - Motivation
  - Optimized AI-Execution Framework
  - HPDP Architecture
  - Verification and Validation
  - **Results**
  - Conclusions
  - Perspectives

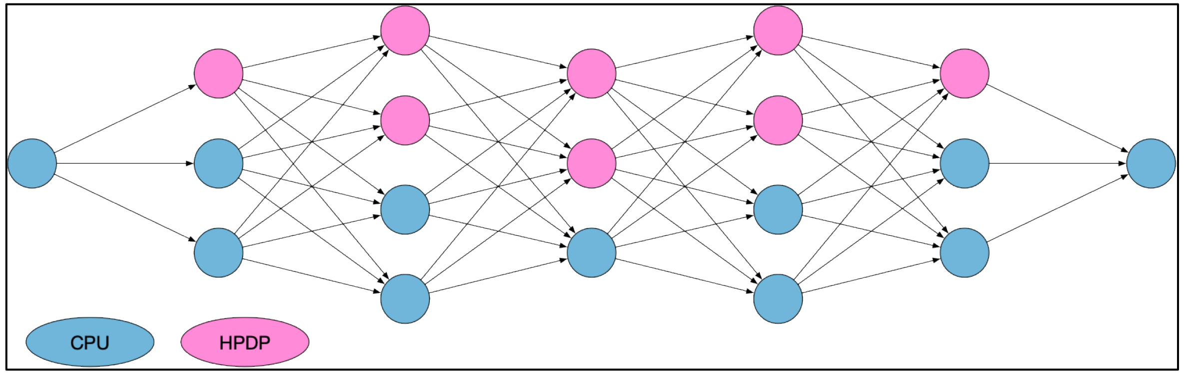
- **OBPMark-ML** was commissioned by ESA as set of DNNs to benchmarks for different Space hardware. The two most important DNNs included are Cloud Detection (Unet based) and Ship Detection (YoloX)

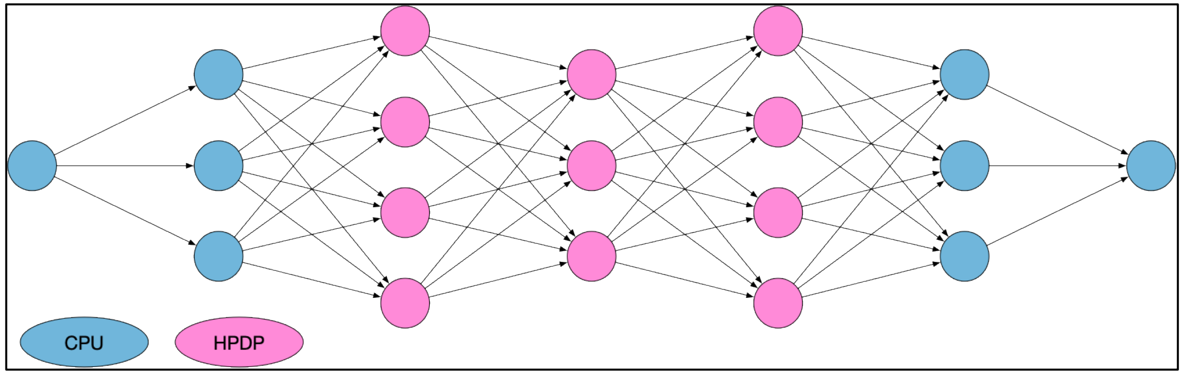


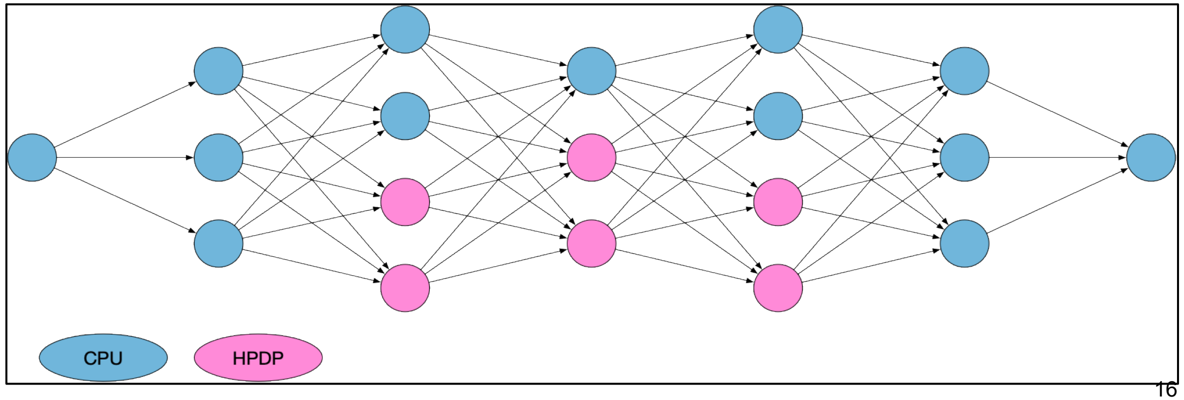
Future work

- **Agenda:**
  - Motivation
  - Optimized AI-Execution Framework
  - **HPDP Architecture**
  - Verification and Validation
  - **Results**
  - Conclusions
  - Perspectives

- SDO tool can gather statistics of the execution of CPU and HPDP including:
  - Latency of each layer
  - Overhead time on the activation and output transmission via Stream-IO or SpaceWire
  - Chaining layers in the HPDP or CPU
- This can be done dynamically for all the kernel size supported by HPDP. I.e., only the layers of the supported sized might be send to the HPDP.







Conclusions

School of Engineering
InES Institute of Embedded Systems

- **Agenda**:
  - Motivation
  - Optimized AI-Execution Framework
  - HPDP Architecture
  - Verification and Validation
  - Results
  - **Conclusions**
  - Perspectives

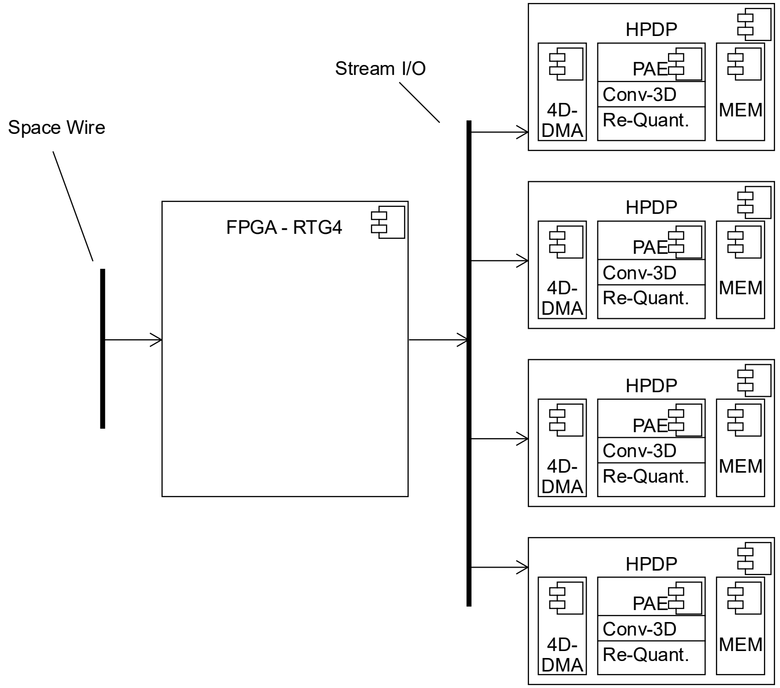
- Operative Conclusions
  - HPDP achieved highly competitive performance.
  - HPDP tightly integrated into a performant (Klepsydra AI) execution queue
  - One-shot configuration (for most input shapes)
  - Excellent relationship between computational efficiency and dependability.
  - IEC 61508 compliant implementation includes rigorous V&V
- Tactical/Strategic conclusions
  - Klepsydra AI decouples orchestration and HW-specific execution
  - Orchestration now offers scheduling for computationally intensive tasks – secure AI execution
  - The execution deployment develops from tightly coupled through loosely coupled to distributed architectures
  - **Model an accelerator as a mathematical operation -> SW definable Compute Unit**

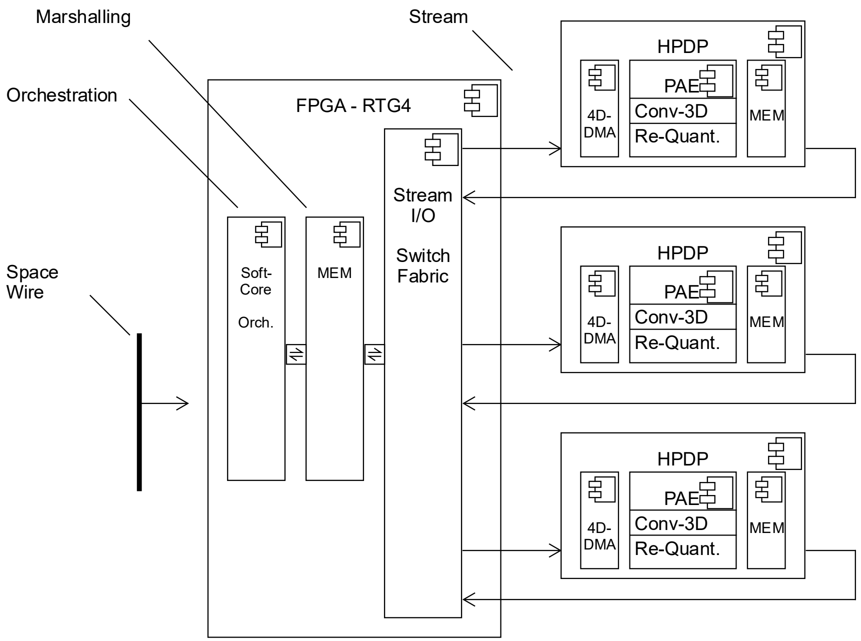
KLEPSYDRA TECHNOLOGIES

zh aw School of Engineering
InES Institute of Embedded Systems

- **Agenda**:
  - Motivation
  - Optimized AI-Execution Framework
  - HPDP Architecture
  - Verification and Validation
  - Results
  - Conclusions
  - **Perspectives**

- Replication of developed Compute Unit in external accelerator facilitates both performance and reliability predictability
- HW can be scaled to required performance KPIs with known ramifications for other KPIs (power, cost …)

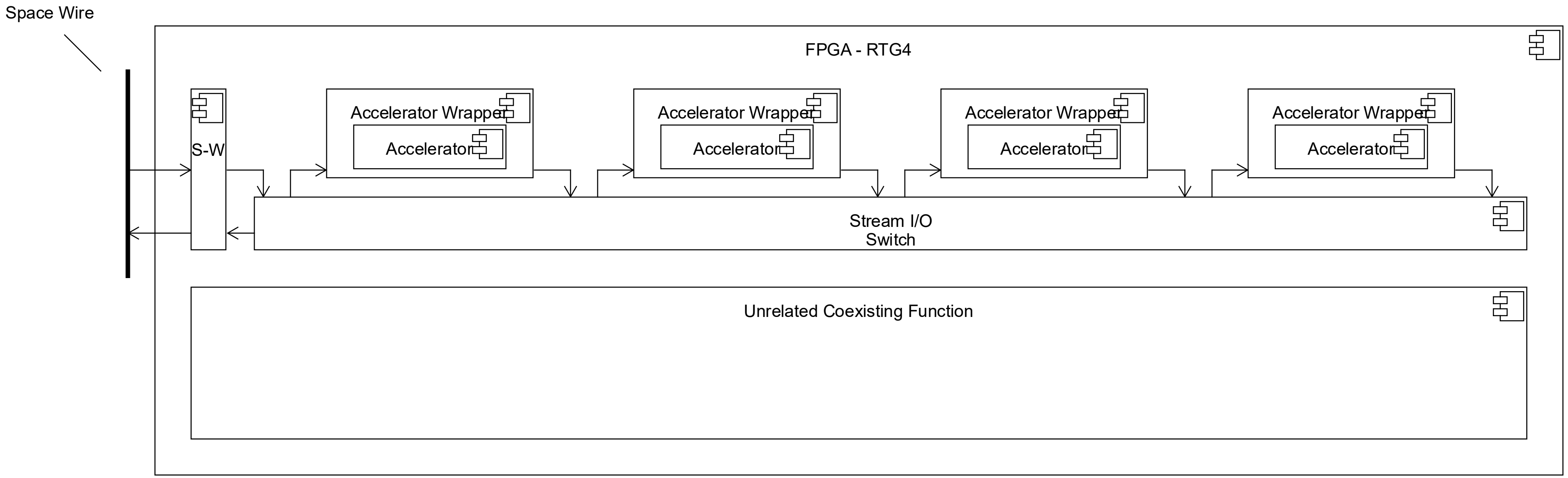




Perspective 2 -> Configurable co-processors

- **Agenda:**
  - Motivation
  - Optimized AI-Execution Framework
  - HPDP Architecture
  - Verification and Validation
  - Results
  - Conclusions
  - **Perspectives**

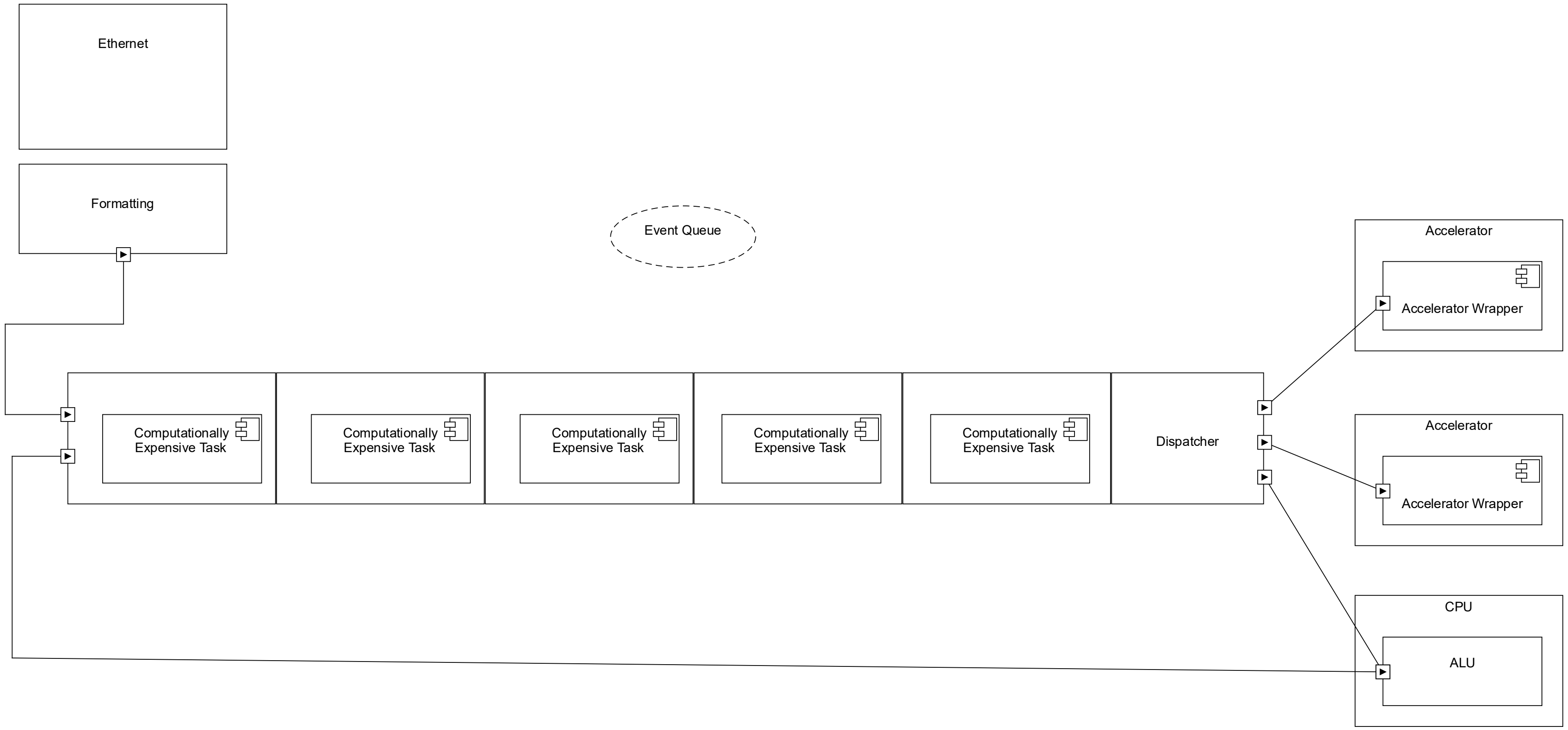
- Replication of developed Compute Unit in configurable/programmable co-processors (FPGA GPU, NPU …) facilitates both performance and reliability predictability together with co-existence of other functions.
- Vendor dependent and independent performance predictions and results.



Perspective 2 -> Configurable co-processors

- **Agenda:**
  - Motivation
  - Optimized AI-Execution Framework
  - HPDP Architecture
  - Verification and Validation
  - Results
  - Conclusions
  - **Perspectives**

- Frontload event-queue with direct data from inputs
  - Example – some Ethernet variant



Conclusions

Support industry standard file format (e.g. ONNX)

Space qualified solution (HW + SW)

Different AI models, same API (e.g. no coding needed)

Thank you
for your attention!

Klepsydra Technologies AG
pablo.ghiglino@klepsydra.com

Zurich University of Applied Sciences
hans.doran@zhaw.ch