Optimal Real-Time Event Selection in the Global Trigger system

Group: D. Miller (Chicago), N. Konstantinidis (UCL) and **I. Xiotidis (CERN)**
Meeting: Next Generation Triggers 1st Technical Workshop
Date: 26-Nov-2024

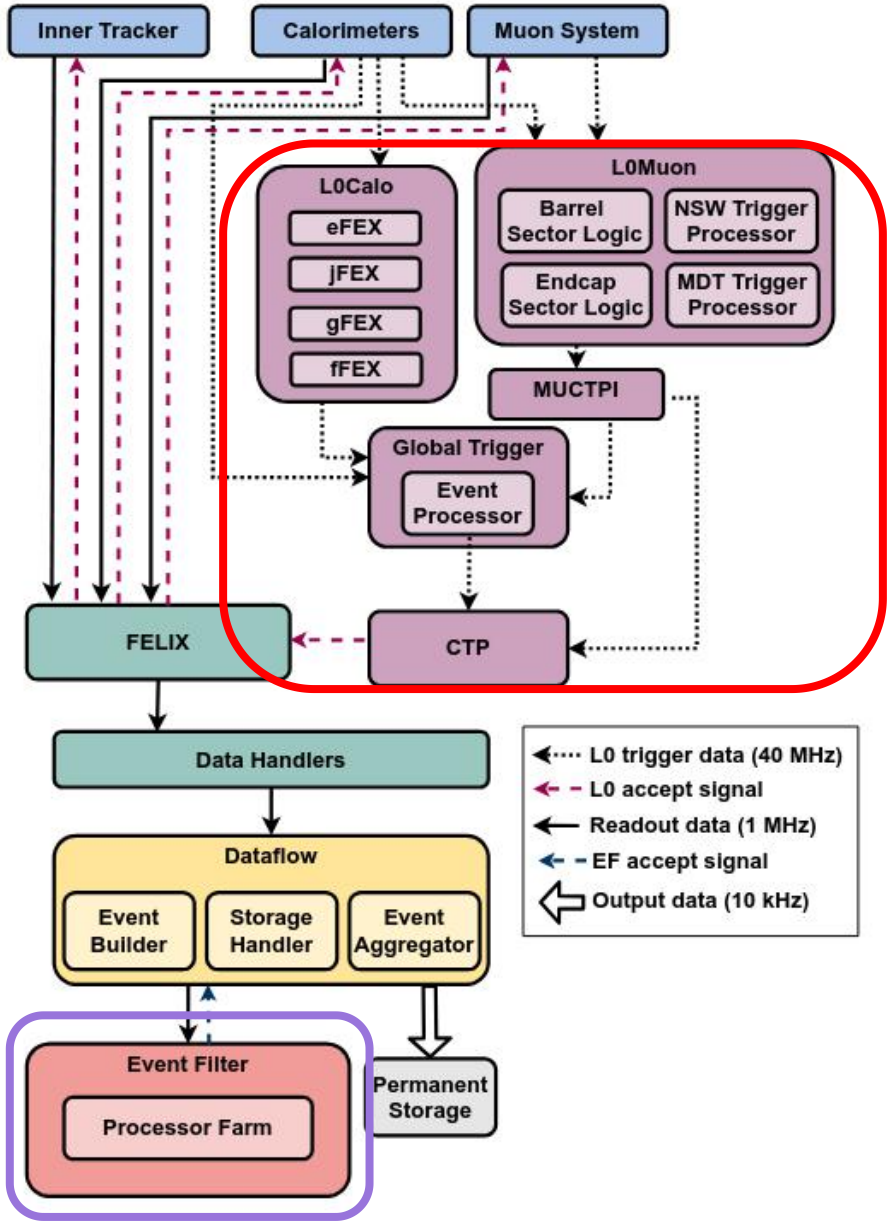
CERN | NexTGen Next Generation Triggers

Overview

- Introduction
  - ATLAS TDAQ and NGT project
  - Interactions with other WPs
  - Global Trigger
- WP2.1
  - Project overview
  - Framework work
  - Algorithm interfaces
  - Resource extraction
- Evaluation of new technologies
  - AMD AI Engines
  - Example implementations for comparisons (e/gamma BDT)
  - AIE in L0-Global
- Summary

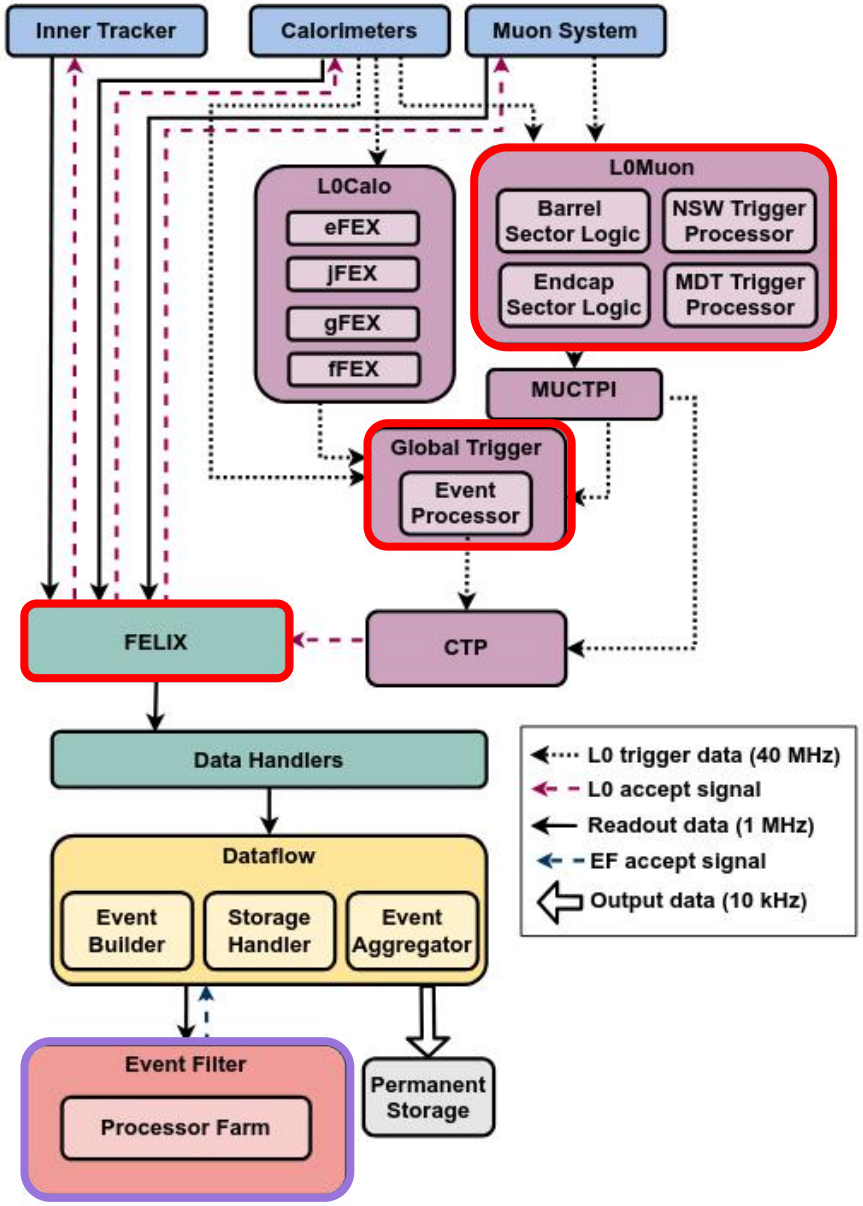
ATLAS NGT Introduction

- The ATLAS Trigger and Data Acquisition system for HL-LHC follows a dual approach on event selection
  - **Hardware Trigger** (Level-0): Quickly analyses the calorimeter and muon system data providing a 40:1 rate reduction
  - **Software Trigger** (EventFilter): Analyses L0-Accepted data and adds also tracker information for a further 100x rate reduction



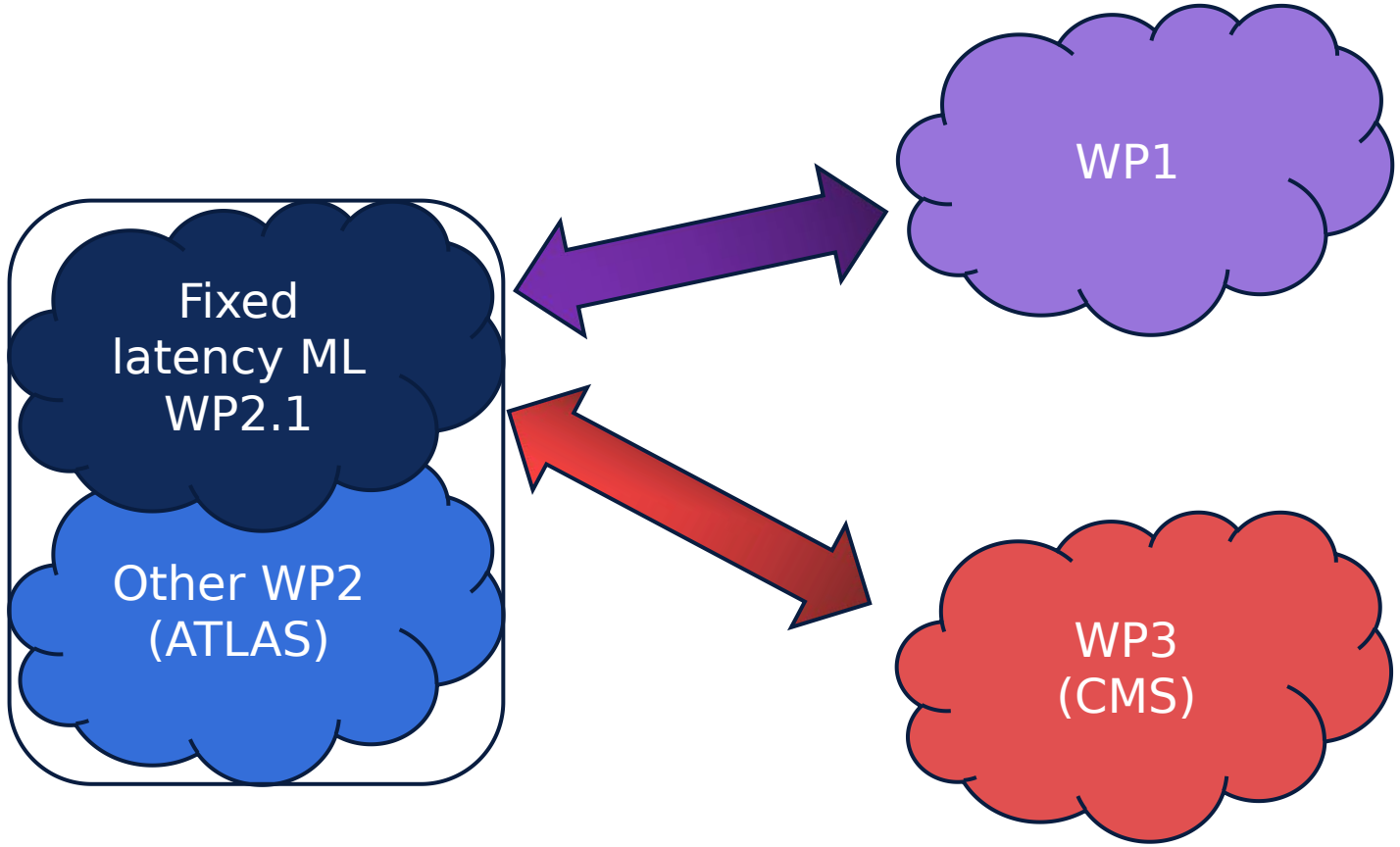
ATLAS NGT Introduction

- The ATLAS Trigger and Data Acquisition system for HL-LHC follows a dual approach on event selection
  - **Hardware Trigger** (Level-0): Quickly analyses the calorimeter and muon system data providing a 40:1 rate reduction
  - **Software Trigger** (EventFilter): Analyses L0-Accepted data and adds also tracker information for a further 100x rate reduction
- The **Next Generation Trigger** project has multiple work packages exploring innovative trigger solutions, with a natural split following the ATLAS TDAQ:
  - **Custom Electronics: WP2.1**, WP2.2 and WP2.3
  - **Processor Farm:** WP2.4, WP2.5, WP2.6 and WP2.7
- All packages in communication for efficient result extraction



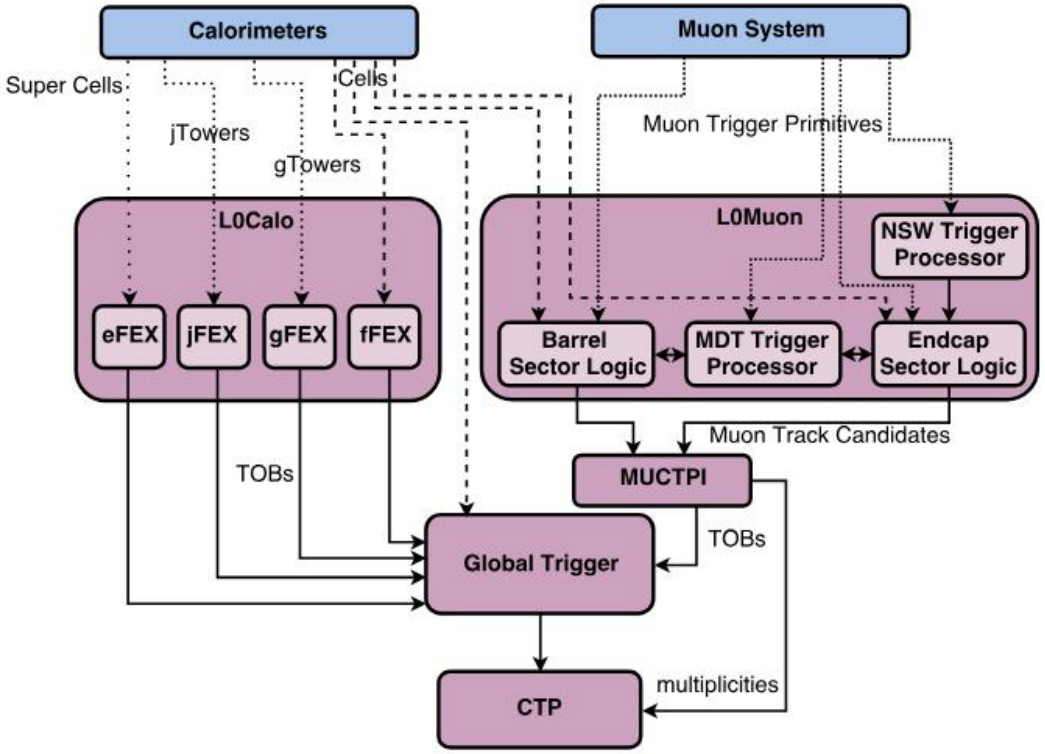
NextGen Trigger interactions

- WP2.1 naturally in contact with the ATLAS WPs
- Many developemnts in WP1 and WP3 are similar to our aims
  - WP1.1-1.3
  - WP3.5-3.7
- What would be the optimal way to talk to each other?
  - Mainly avoid re-inventing the wheel or work in parallel on same topics



Introduction to Global Trigger

- WP2.1 focuses on implementations **within** the context of the ATLAS Global Trigger (L0-Global)
- L0-Global receives data from the **calorimeter** and **muon** systems
  - 50Tbps of data streamed through
  - Tight latency budget of 10μs
- Offline-like algorithms implemented in firmware for event selection
- Due to its central position, L0-Global is **critical** system with two main constraints
  - Number of incoming links (from all the pre-processors)
  - FPGA resources due to algorithm complexity
- Constraints drive decisions on FPGA selection: VP1802 (largest AMD FPGA on the market)



Introduction to WP2.1

- **Main goal**: Exploration of novel machine learning reconstruction within the ATLAS Global Trigger
- WP2.1 team:
  - Project Leads: D. Miller and N. Konstantinidis
  - Fellows: I. Xiotidis

WP2.1 Fellow

I.Xiotidis

WP2.1 PIs

David Miller

Nikolaos Konstantinidis

Introduction to WP2.1

- **Main goal**: Exploration of novel machine learning reconstruction within the ATLAS Global Trigger
- WP2.1 team:
  - Project Leads: D. Miller and N. Konstantinidis
  - Fellows: I. Xiotidis
  - 2025: 3x PhD students and 1x Fellow
  - Total: 3x Fellows, 3x PhD (a lot of exciting opportunities!)

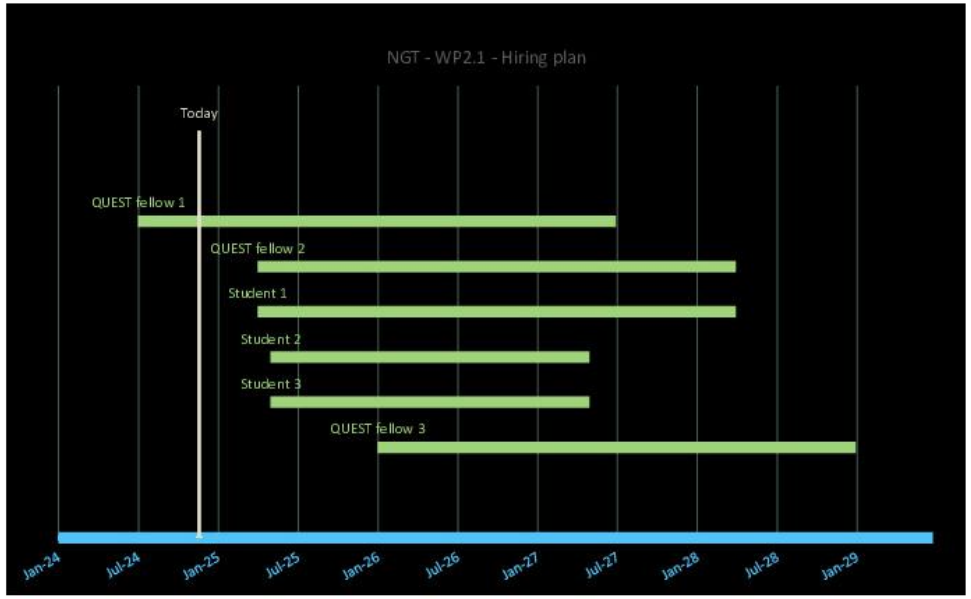
WP2.1 Fellow

I.Xiotidis

WP2.1 PIs

David Miller

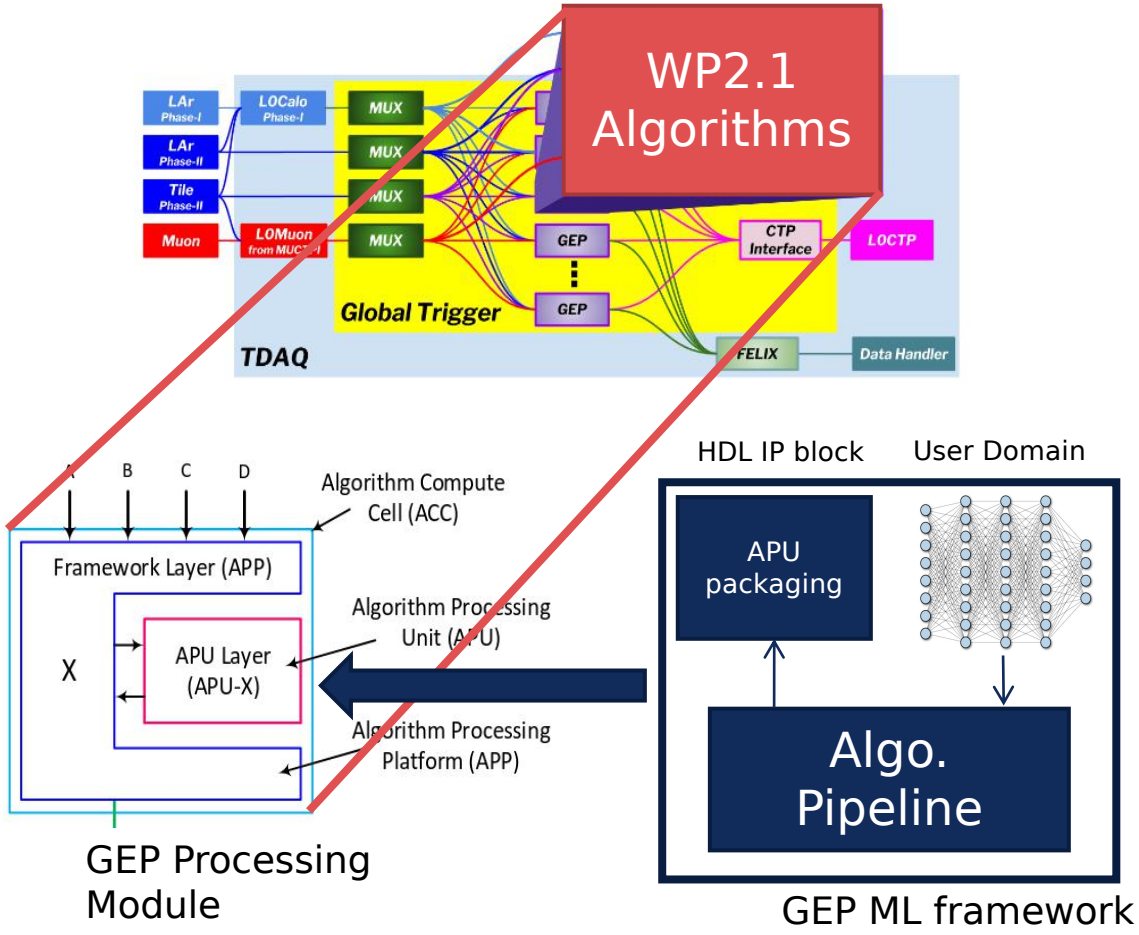
Nikolaos Konstantinidis



Hiring Schedule

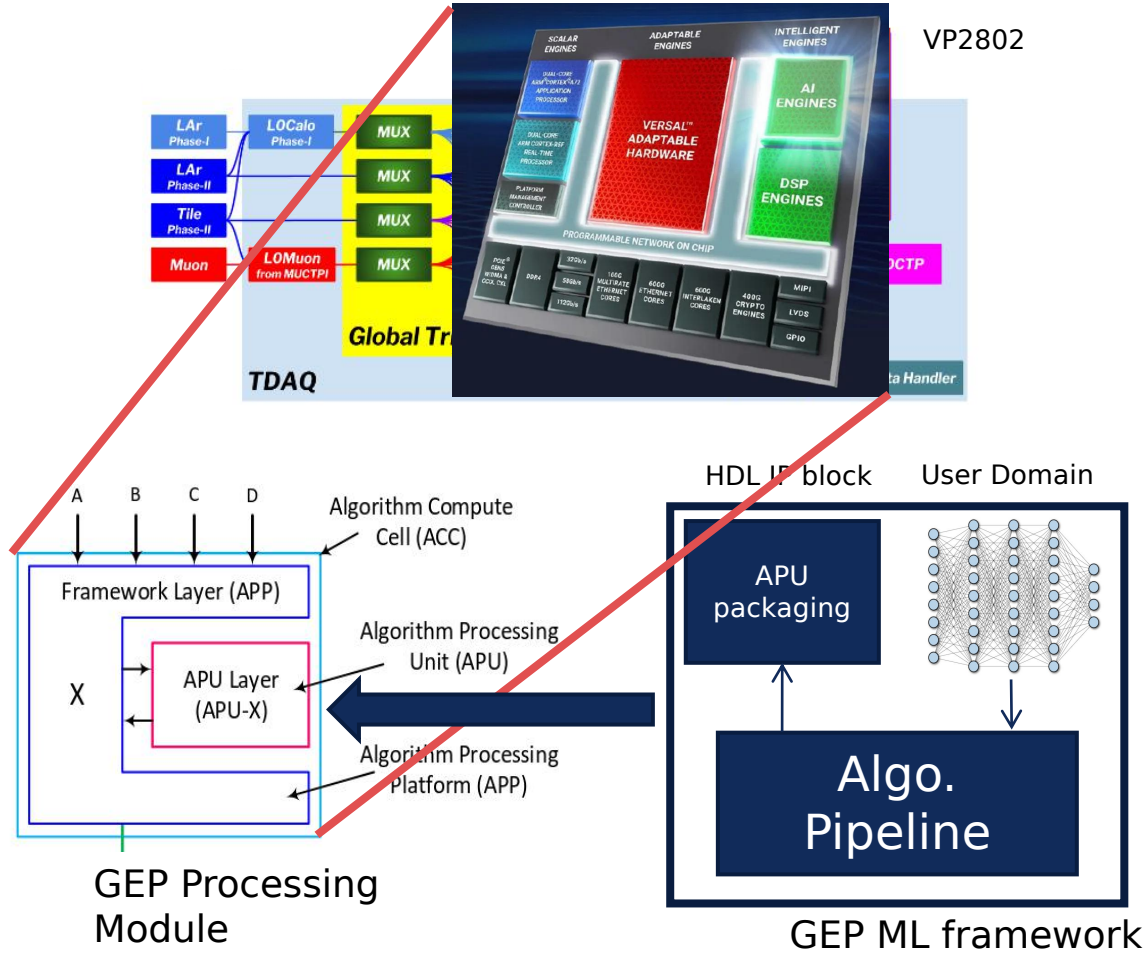
Introduction to WP2.1

- **Main goal**: Exploration of novel machine learning reconstruction within the ATLAS Global Trigger
- WP2.1 team:
  - Project Leads: D. Miller and N. Konstantinidis
  - Fellows: I. Xiotidis
  - 2025: 3x PhD students and 1x Fellow
  - Total: 3x Fellows, 3x PhD (a lot of exciting opportunities!)
- Three main components within WP2.1:
  - Develop a common framework for ML optimisation for FPGA development
  - Explore novel ML algorithms for L0-Global



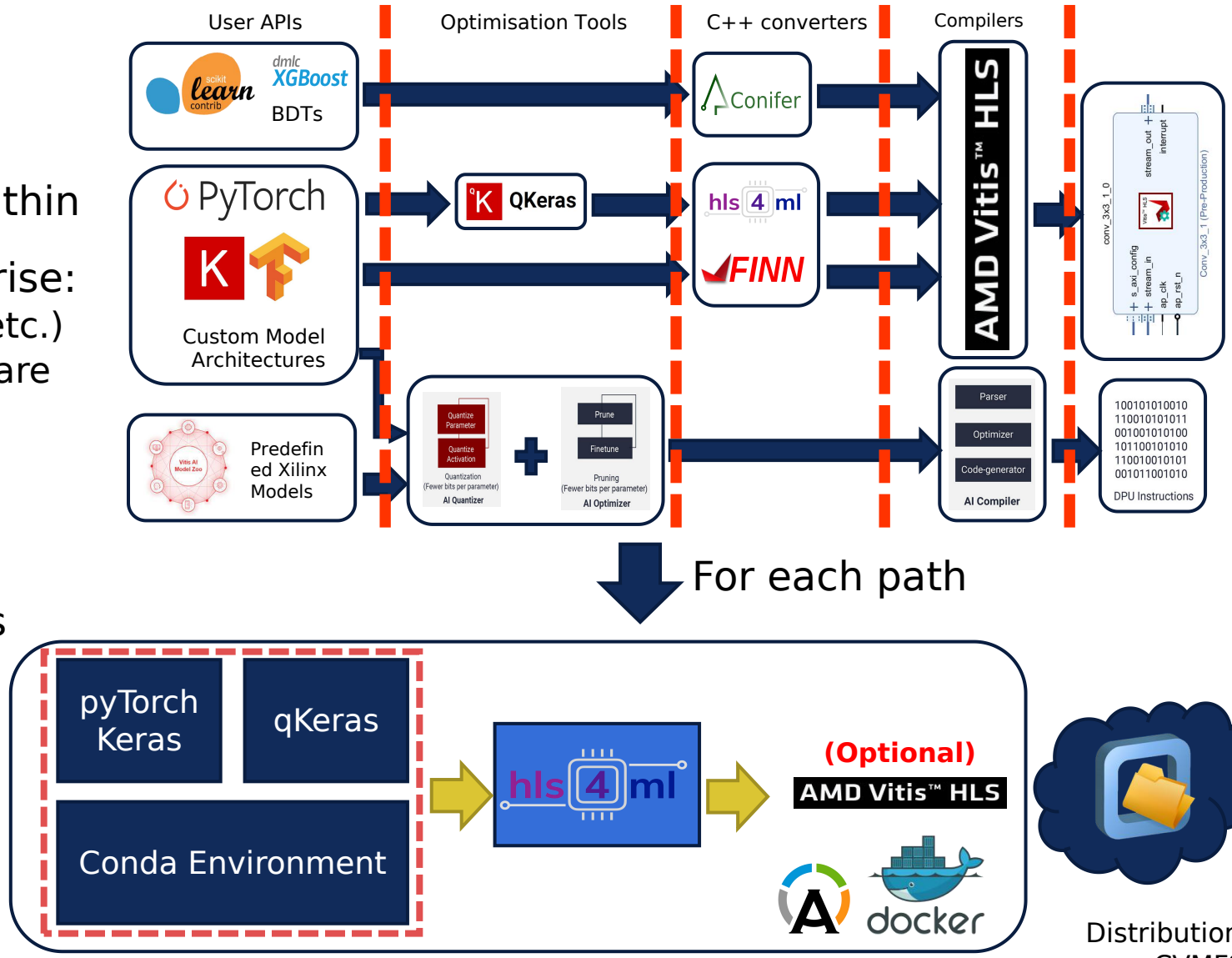
Introduction to WP2.1

- **Main goal**: Exploration of novel machine learning reconstruction within the ATLAS Global Trigger
- WP2.1 team:
  - Project Leads: D. Miller and N. Konstantinidis
  - Fellows: I. Xiotidis
  - 2025: 3x PhD students and 1x Fellow
  - Total: 3x Fellows, 3x PhD (a lot of exciting opportunities!)
- Three main components within WP2.1:
  - Develop a common framework for ML optimisation for FPGA development
  - Explore novel ML algorithms for L0-Global
  - Evaluate of new industry technologies



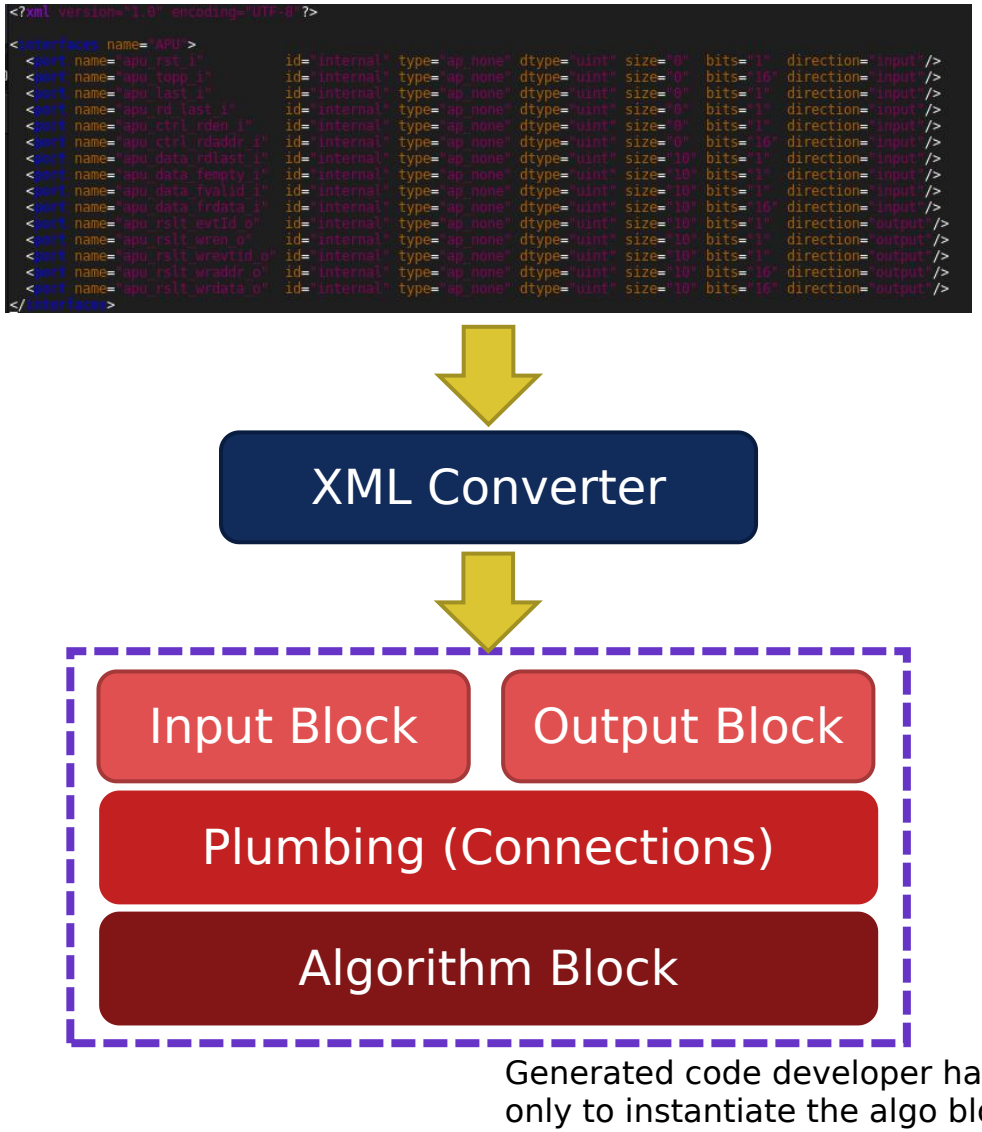
WP2.1 framework

- WP2.1 work aims to **integrate** within the Global Event Processor (GEP) where the following constraints arise:
  - FPGA type (e.g. Versal Premium, etc.)
  - Interface between different firmware blocks (APU)
  - Allowed resources per algorithm (optimisation)
  - Latency budget (environmental constraints)
- Integrating ML in those conditions requires **unified** framework
- First step was to survey existing software tools for ML in FPGAs
  - All pipelines wrapped in Apptainer/Docker for developers
  - Shareable via CVMFS



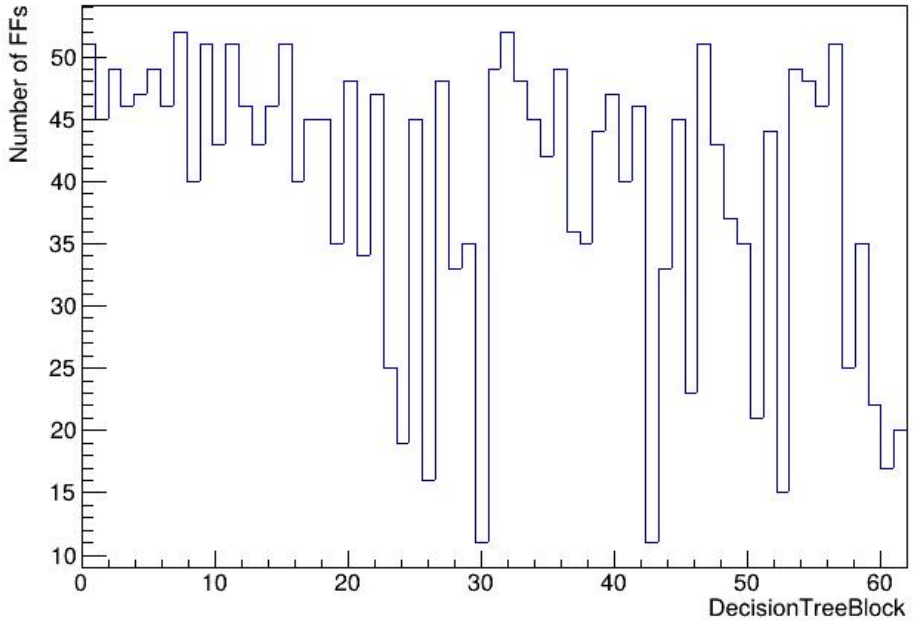
Algorithm interfaces

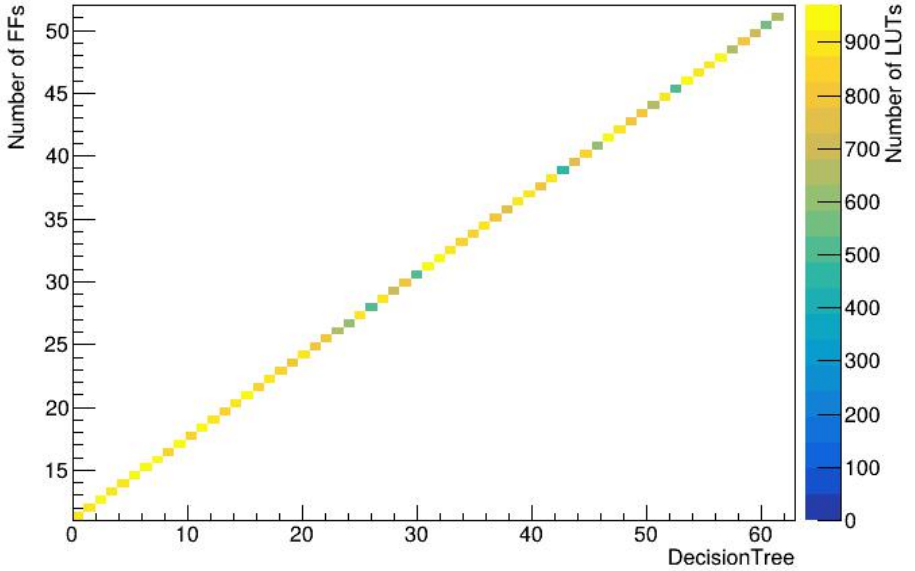
- Developed algorithms face challenges when integrated into frameworks
  - Interfaces might not be the same
  - Resource utilisation/timing closure might be affected
- Getting the algorithm into the framework as quickly as possible crucial for estimating overall performance
- GEP has a **standardised** interface towards the algorithms (**APP**), each algorithm (**APU**) should follow that
  - Getting the correct interface allows for quicker validation with realistic conditions!
- Dedicated library within WP2.1 providing interfaces to framework and necessary plumbing
  - XML-based package
  - Developer/Framework engineer provide configuration files for their respective counterpart
  - Status: **In-Development**



Resource extraction - physics perfromance

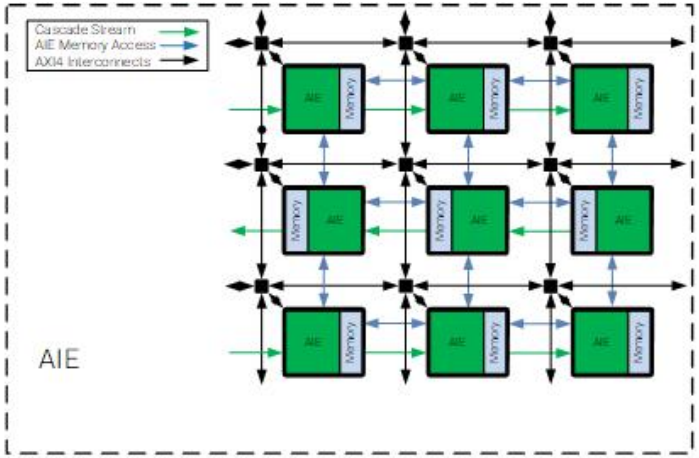
- Critical to the task would be to be able to extract performance metrics
  - Parsing all types of the Vitis Unified Platform reports
  - Allows for quick cross resource plotting (e.g. LUTs vs latency, latency vs max clock frequency per RTL block, etc.)
- Included in the framework development
  - Currently supporting Vitis HLS, Vitis AIE, Vitis AI
  - Support to be added on extracting physics performance from ML software tools (e.g. Keras, pyTorch)
  - Part of the containers for ML algorithm development
  - Growing number of plotting scripts and implementing unified Command-Line Interface (CLI) for cross-platform comparisons (e.g. HLS vs AIE)
  - Status: **In-Development**

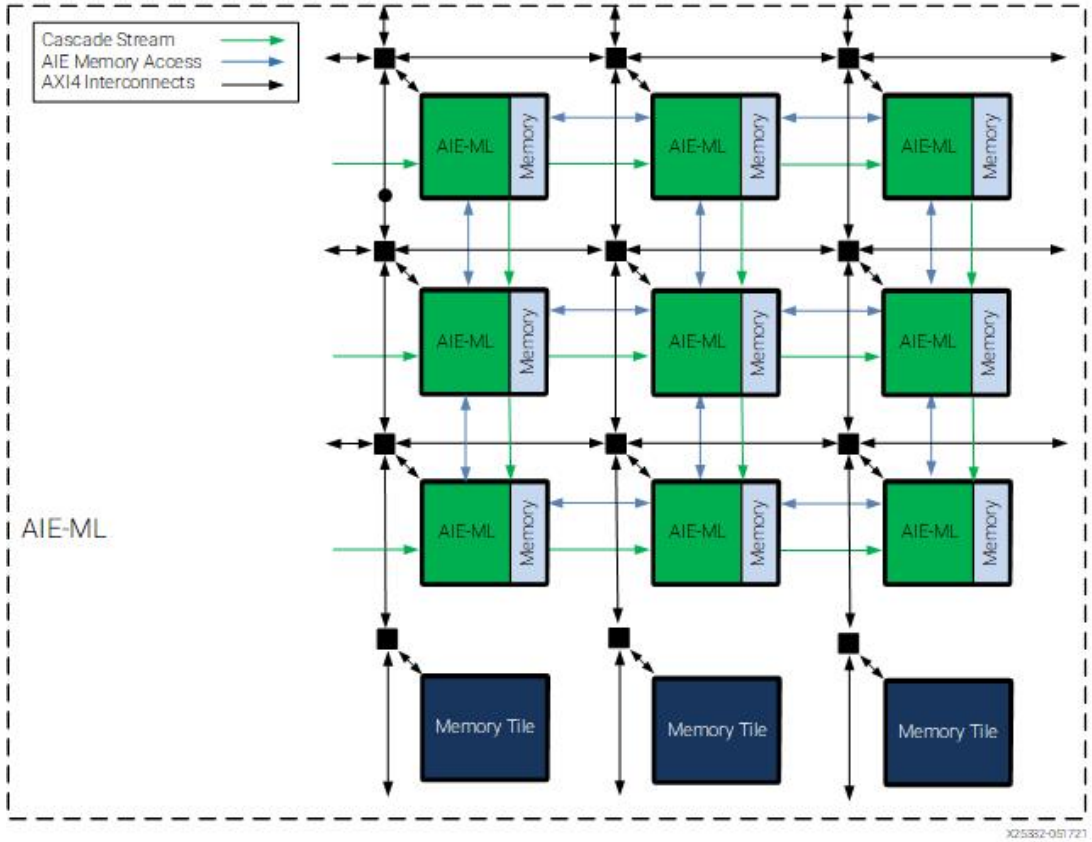




Evaluation of new technologies

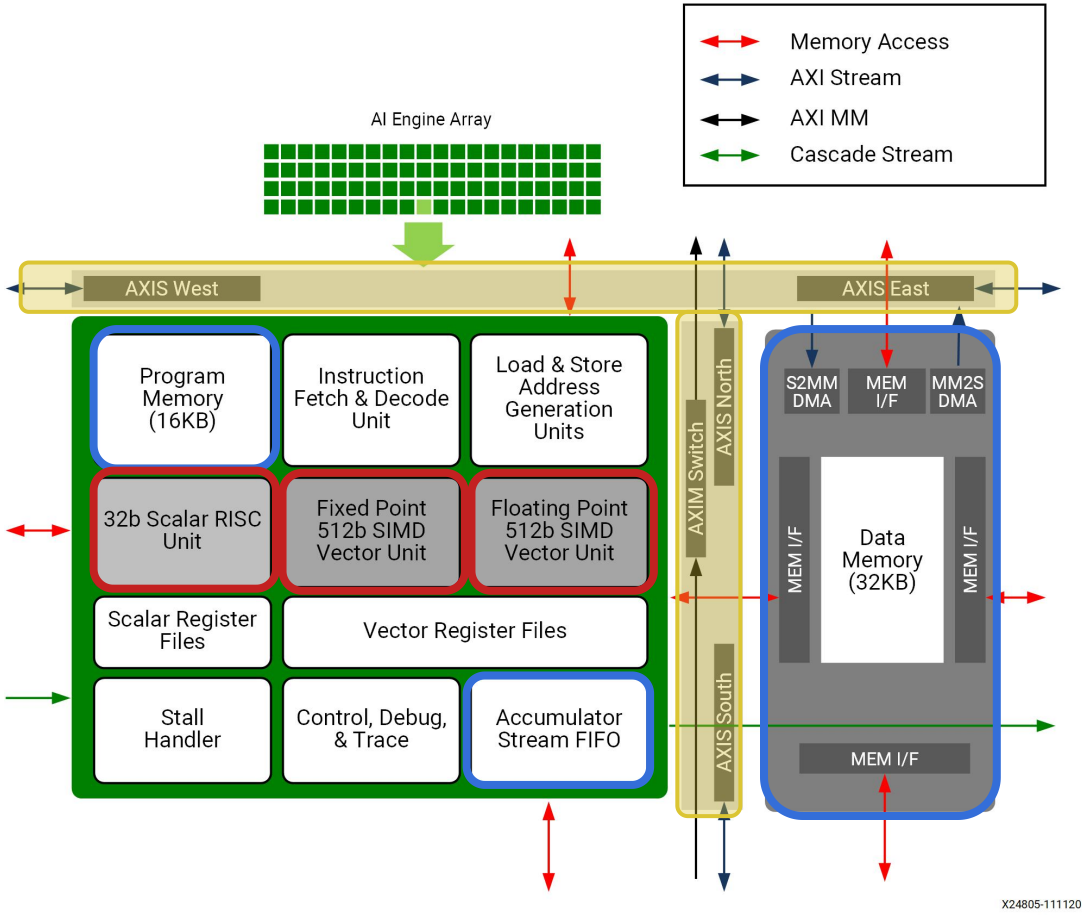
- New technology in the AMD ecosystem focuses on dedicated vector processors (AI-Engines)
  - Optimised for Digital Signal Processing and Machine Learning
  - Two flavours on the market, AIE and AIE-ML





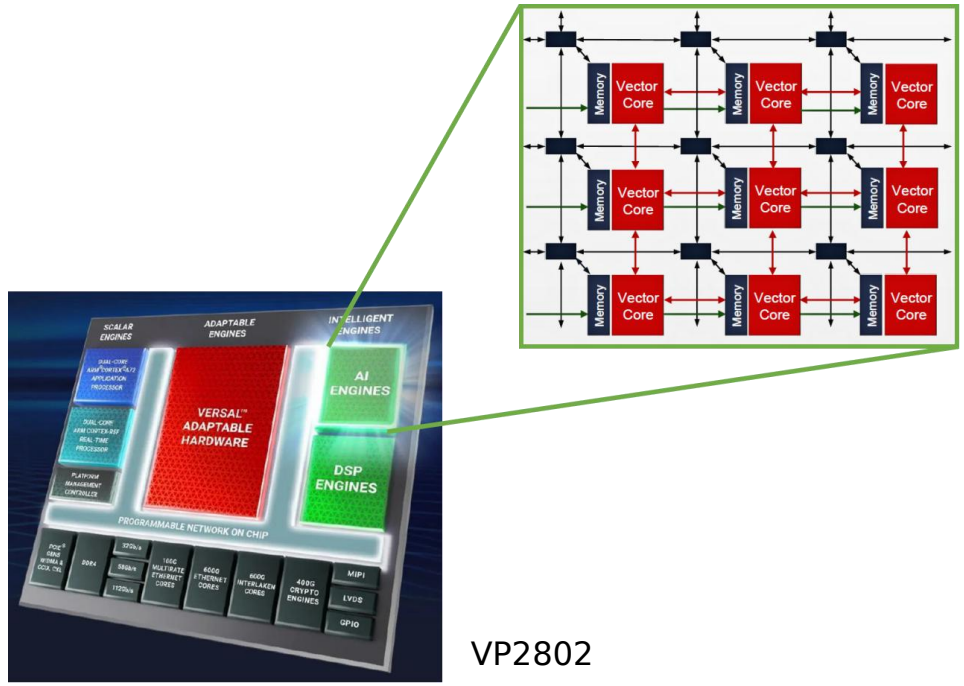
Evaluation of new technologies

- New technology in the AMD ecosystem focuses on dedicated vector processors (AI-Engines)
  - Optimised for Digital Signal Processing and Machine Learning
  - Two flavours on the market, AIE and AIE-ML
- Both AIE versions contain the following parts
  - Scalar processor (reduced instructions)
  - Vector processor
  - Memory
  - Interface (to neighbouring AIE, FPGA, Network on Chip)
- Allow for data-level parallelisation and instruction level!
  - Multiple optimisation paths for developers many of them still to be done by hand

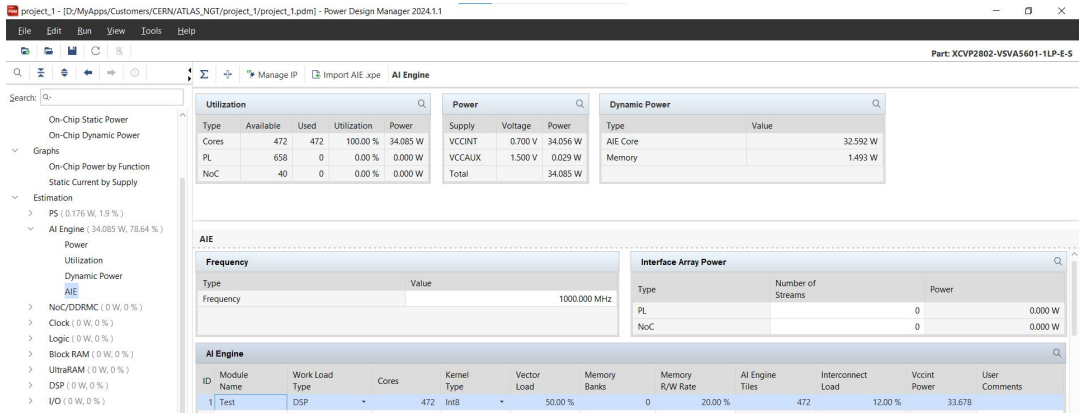


AI-Engines in Global

- L0-Global deals with the ATLAS hardware trigger constraints by deploying a single hardware board called Global Common Module (GCM)
  - The final system will contain O(50) GCMs processing around 50Tbps
  - Includes large VP1802 FPGAs to deal with the I/O demands and the feature extraction algorithm implementations
- Recently AMD released VP2802, pin-compatible to VP1802 but also **contains the AI-Engines**
- Demonstrating that the AIEs can provide a significant advantage within the constraints of the L0-Trigger is a unique opportunity
  - Discussing with AMD to purchase a VP2802 for a full slice test evaluation of the WP2.1 algorithms
- One of the first hardware boards with AIE at CERN!



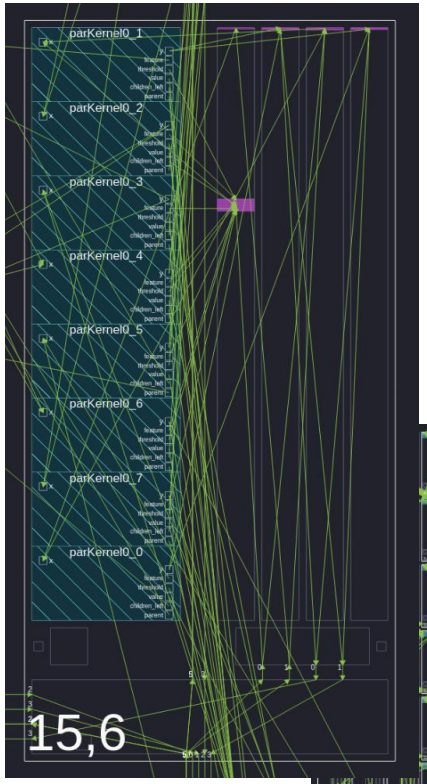
VP2802



Power estimates from AIEngines

AIE programming

- Two methods for configuring the AIE
  - **Vitis AI**: Constrains the FPGA resources in using a dedicated IP block
  - **Vitis AIE Compiler**: Compiles C++ (**with enhancements**) for the AIE processors (Scalar, Vector)

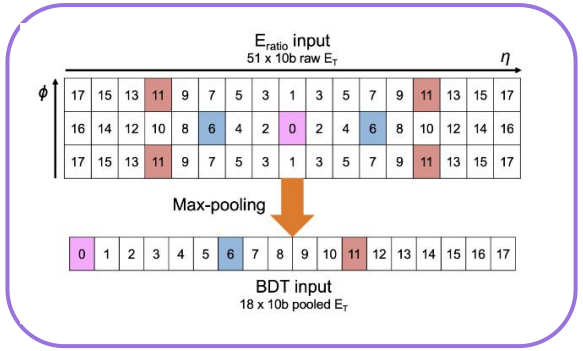


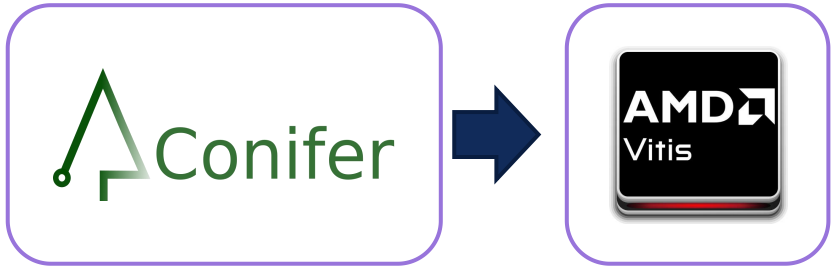
Single AI Engine Utilisation (different cores)

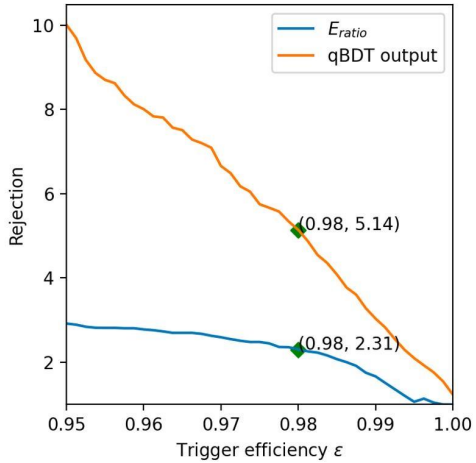
AI Engine array

AIE programming

- Two methods for configuring the AIE
  - **Vitis AI**: Constrains the FPGA resources in using a dedicated IP block
  - **Vitis AIE Compiler**: Compiles C++ (**with enhancements**) for the AIE processors (Scalar, Vector)
- For the WP2.1 example we used a BDT designed and optimised for e/gamma identification
  - As Vitis AI **doesn't support** BDTs and we didn't want to use the IP, we focused on the AIE Compiler option
  - Choise of the BDT mainly to minimise uncertainties and allowing us to familirise with the compiler
- BDT implemented already in HLS which allowed for comparisons
  - Inputs: 11b integers, 11b fixed point features
  - Output: 11b fixed point
  - Trees: 63x, with depth 5



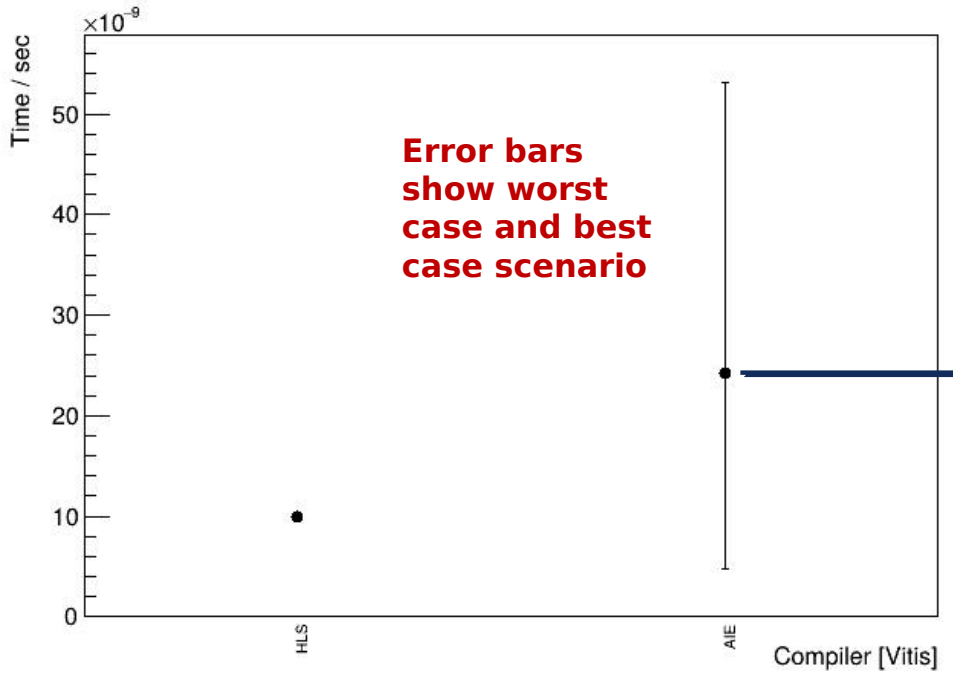




Preliminary AIE results

- Implementation process
  1. Extract HLS code from Conifer
  2. Modify for using the AIE compiler
  3. Use the scalar processor only (not expecting to give an improvement but allows for functional tests)
  4. Utilise instruction and data level parallelisation
  5. Optimise data transfers
- The scalar processor pass yieldied a **10µs** latency
- Walking through steps 3-5 allowed for an average **25ns** decision per tree
  - Using 32b integers, floats everywhere: 10% physics performance increase
- The BDT might not be the best use case as its computationally lightweight
  - Next attempt focusing on getting a CNN implemented (using HLS4ML)
  - The study will teach us also how to automate (if possible)

Important: Function only latency (evaluated over 10x e/gamma obects) work is required for “proper” interfacing



HLS: Implemented in Vitis HLS
AIE: Implemented with Vitis AIE Compiler

Summary

- The NextGeneration Trigger project is a unique R&D opportunity aiming to amplify the physics reach of ATLAS (and other future HEP experiments)
- Within ATLAS, the L0-Global is a critical system responsible for reconstructing in real time all the ATLAS events
- Implementing ML-based algorithms within the context of Global can further enhance the physics reach
  - WP2.1 can assist significantly
- WP2.1 completed hiring and necessary hardware procurement for 2024
  - Hiring for 2025 on-going (Doctoral student selection under way, 2nd Fellow post ad out soon!)
  - Hardware procurement for standalone ML training and firmware integration completed
  - GCM with VP2802 in discussion with AMD for final quote to move into production
- Framework for algorithm optimisation in progress
  - First libraries/tools have already prototype versions and will move into production level when more person power arrives
- Evaluation of new technologies advances
  - BDT results show promising latency and pointed out key areas to be taken care of
  - CNN example progressing with results expected within the next month


CERN
NexTGen
Next Generation Triggers