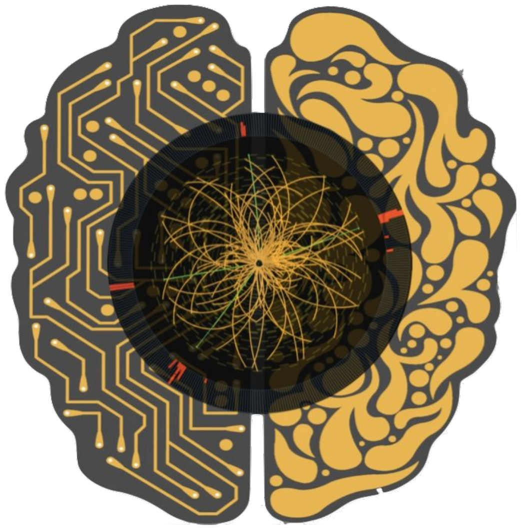
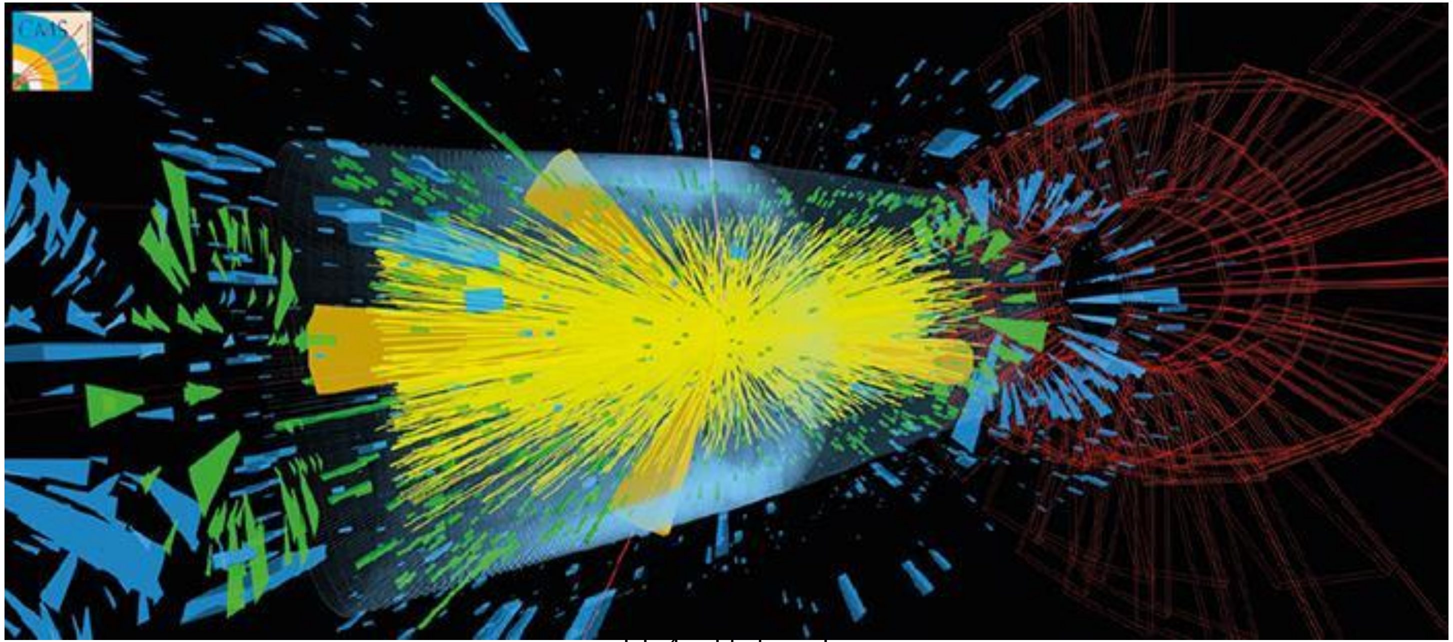


hls4ml

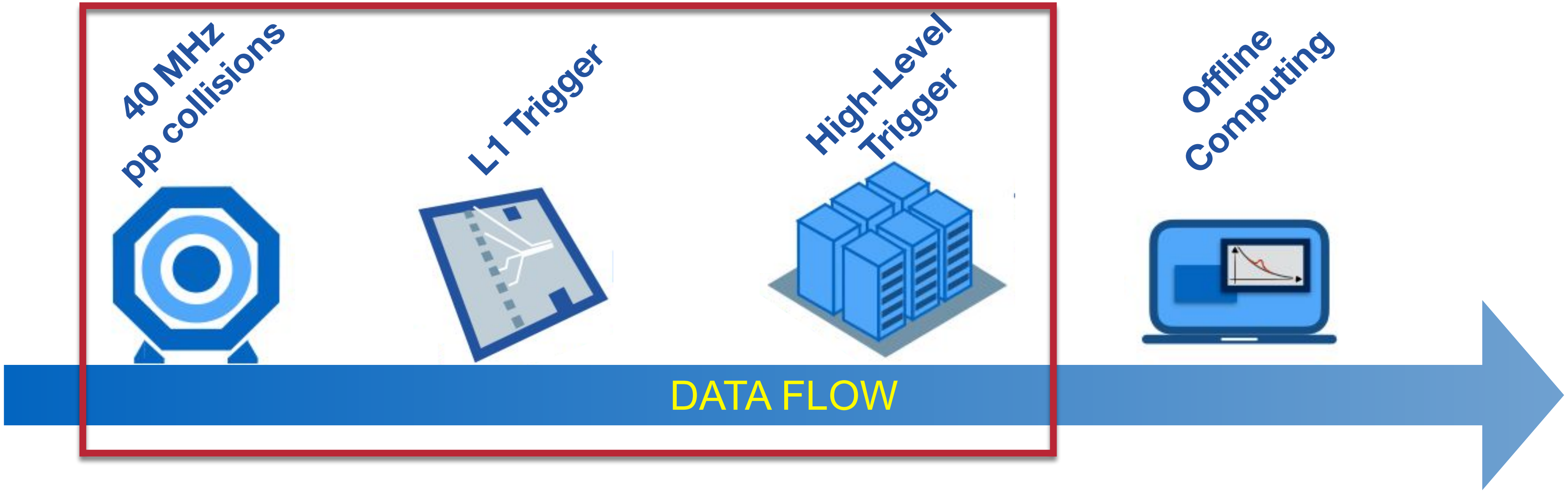
Hands-on Lab Introduction
Artificial Intelligence-on chip: Physics driven hardware co-design

hls4ml origins: triggering at (HL-)LHC

Extreme collision frequency of 40 MHz → extreme data rates O(100 TB/s)
Most collision "events" don't produce interesting physics
"Triggering" = filter events to reduce data rates to manageable levels



LHC Experiment Data Flow



Deploy ML algorithms very early in the game
Challenge: strict latency constraints!

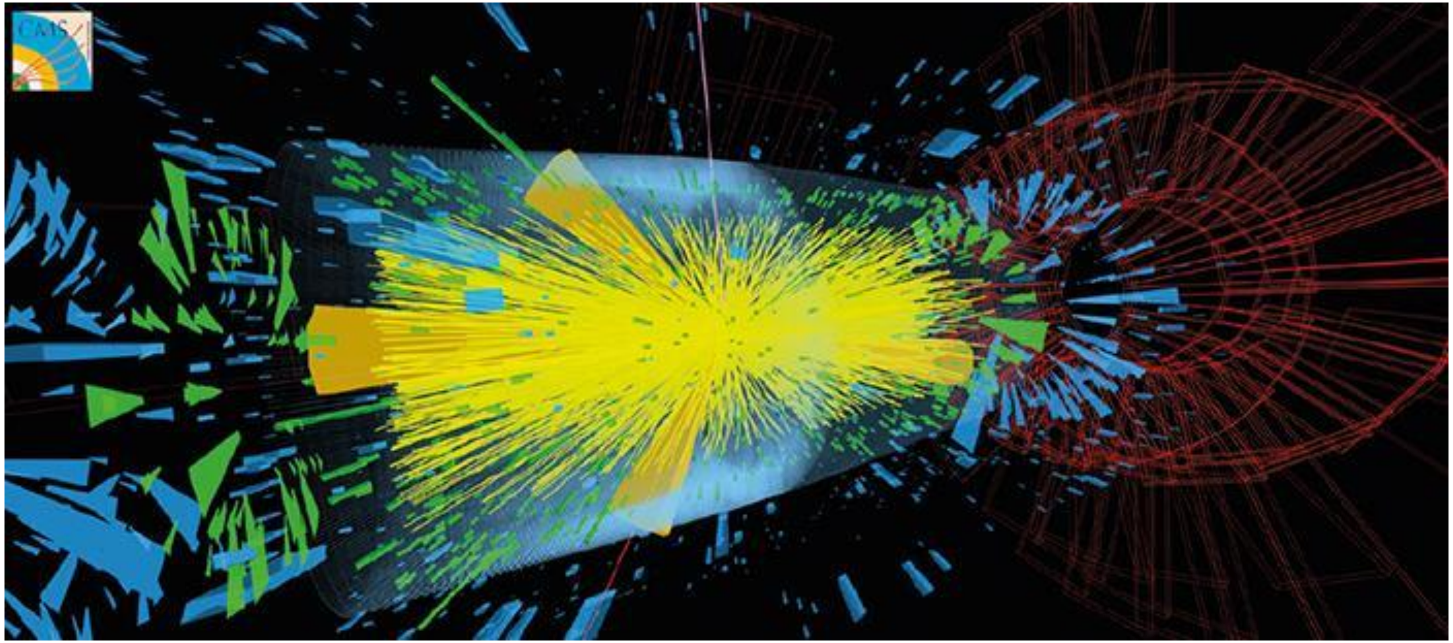
The challenge: triggering at (HL-)LHC

The trigger discards events ***forever***, so selection must be very precise

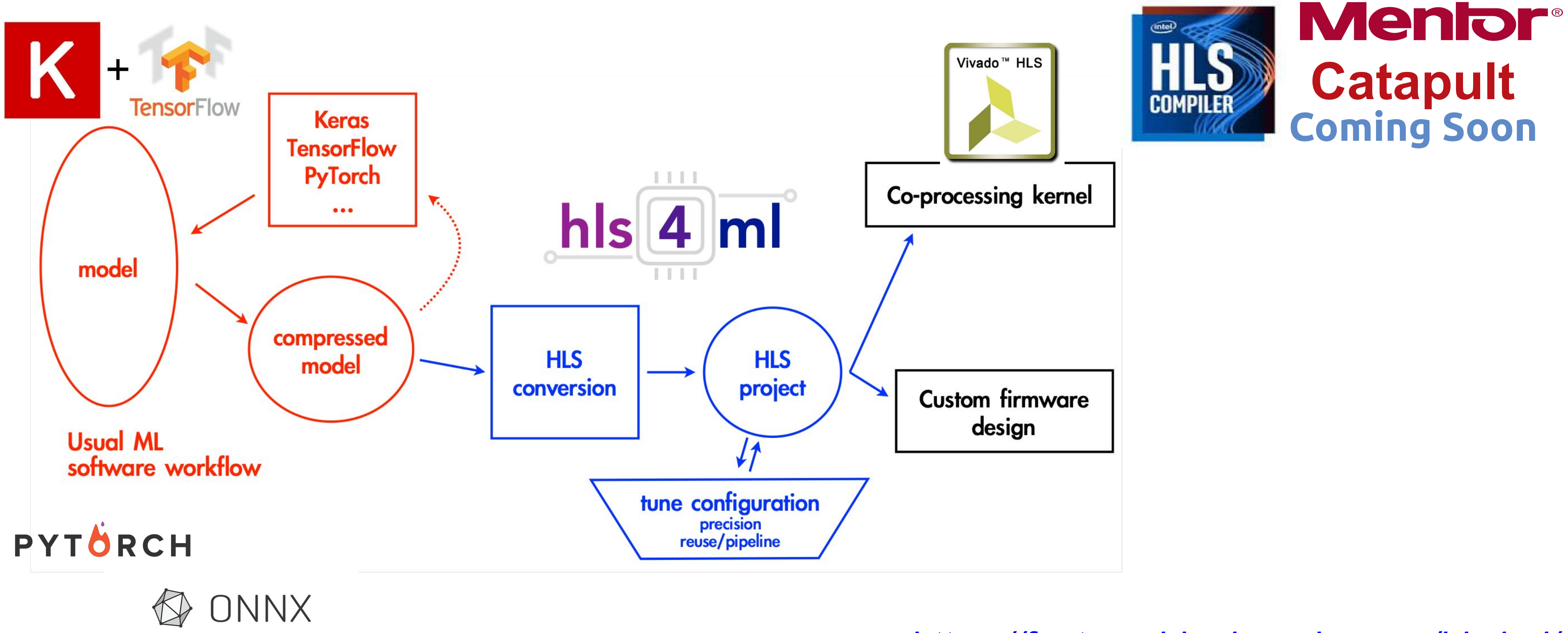
ML can improve sensitivity to rare physics

Needs to be ***fast!***

Enter: **hls4ml** (high level synthesis for machine learning)



high level synthesis for machine learning



https://fastmachinelearning.org/hls4ml/

Efficient NN design for FPGAs

FPGAs provide huge flexibility

Performance depends on how well you take advantage of this

Constraints:

Input bandwidth

FPGA resources

Latency

Today you will learn how to optimize your project through:

- **compression:** reduce number of synapses or neurons
- **quantization:** reduces the precision of the calculations (inputs, weights, biases)
- **parallelization**: tune how much to parallelize to make the inference faster/slower versus FPGA resources

NN training

FPGA project designing

Summary

- After this lab you will have gained some hands on experience with **hls4ml**
  - Translate neural networks to FPGA firmware, run simulation and synthesis
- Tune network inference performance with precision and ReuseFactor
  - Used profiling and trace tools to guide tuning
- Learn how to simply prune a neural network and the impact on resources
- Train a model with small number of bits using QKeras, and use the same spec in inference easily with **hls4ml**