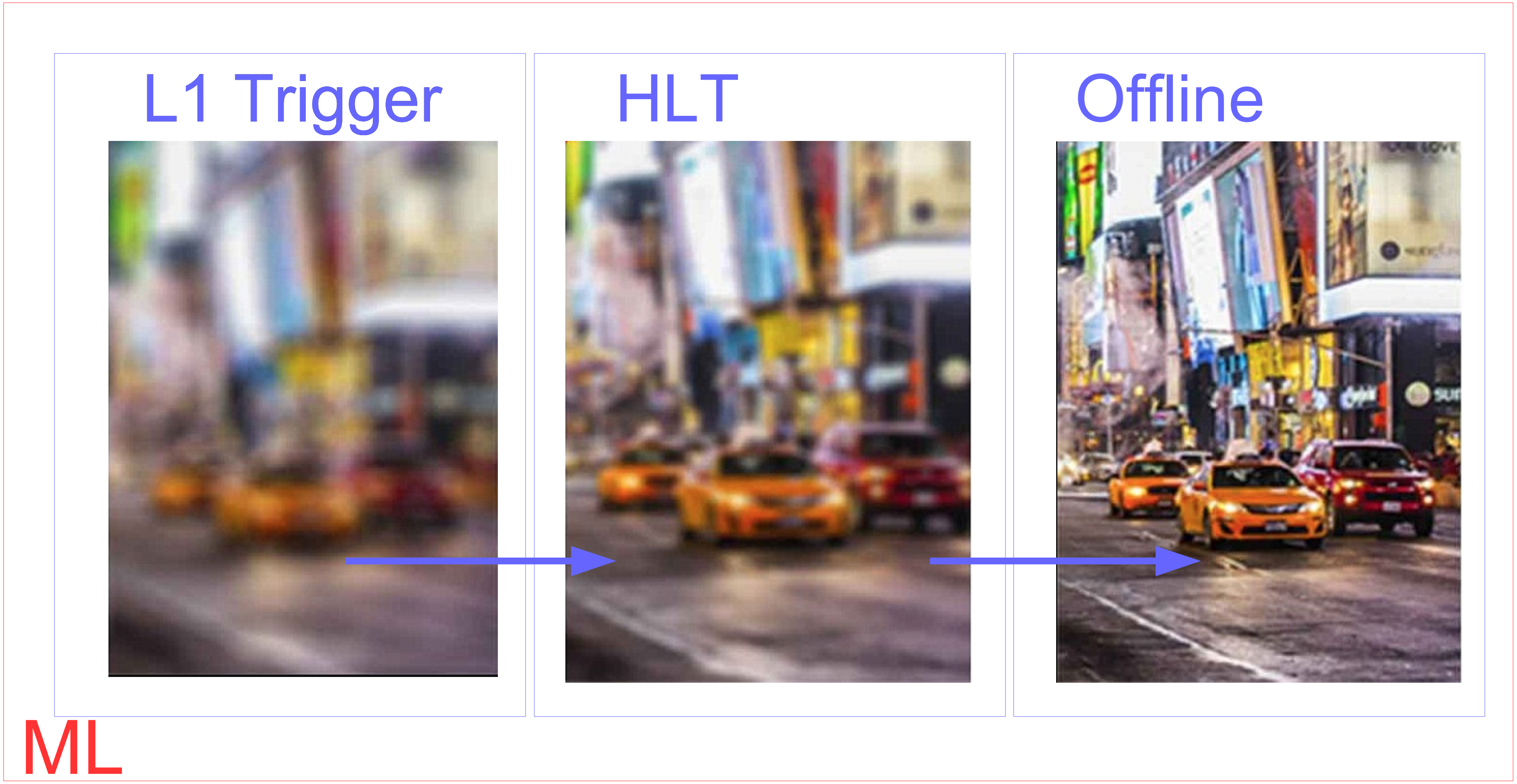
ML in FPGA ....

Phil Harris (MIT), Nhan Tran (FNAL),
Ben Kreis(FNAL), Javier Duarte(FNAL)

Fermilab

CMS
Compact Muon Solenoid

Preface



L1 Trigger

HLT

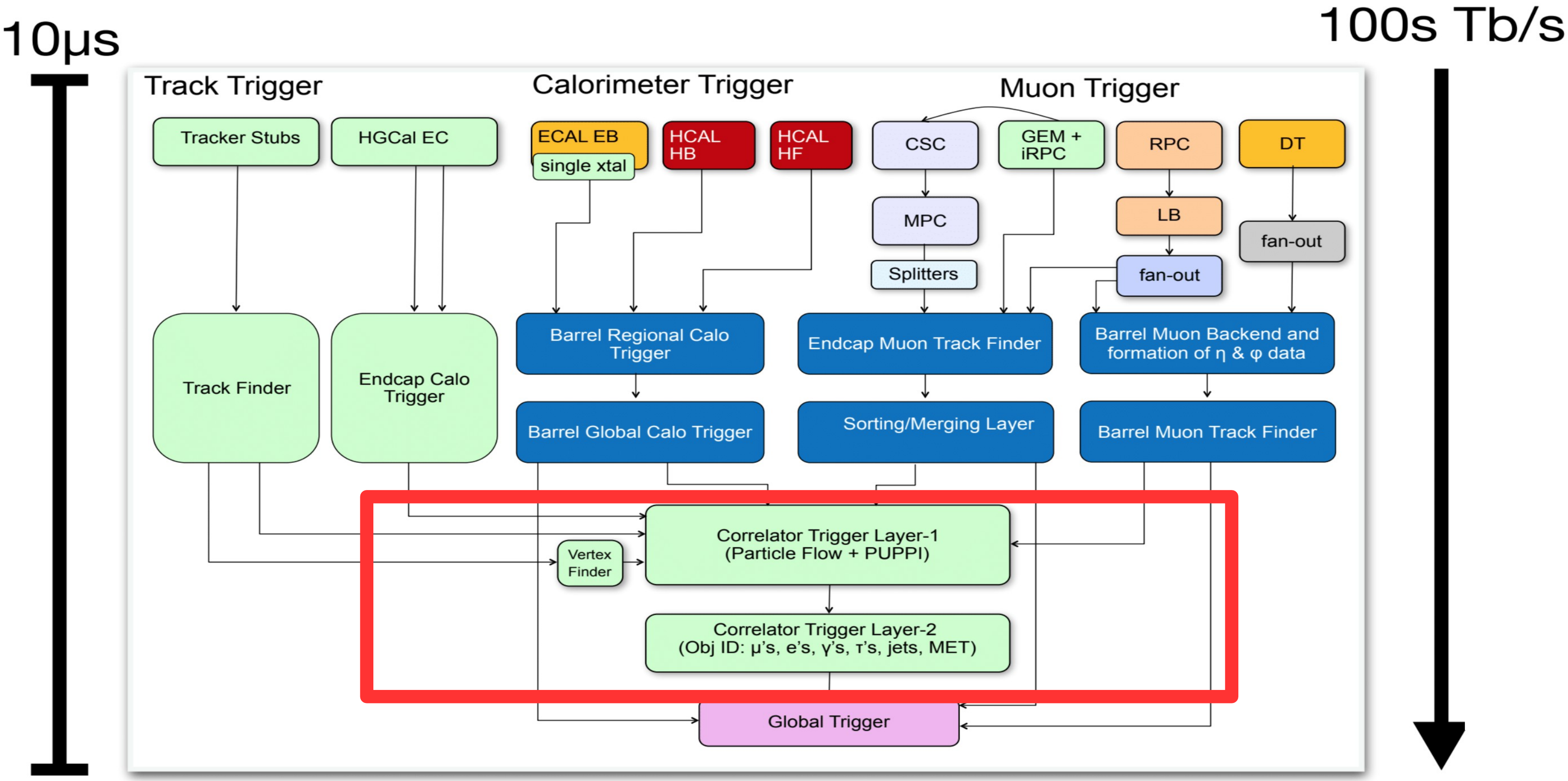
Offline

ML

- ML as a great way to tie levels of reconstruction
  - Algorithms can be used with varying precission
  - Fast inference and quick dev time are compelling for reco
  - We aim to explore ML at all levels with FPGAs

Our main objective

- Actively working to the CMS L1 trigger upgrade
  - We are working on the correlator layer

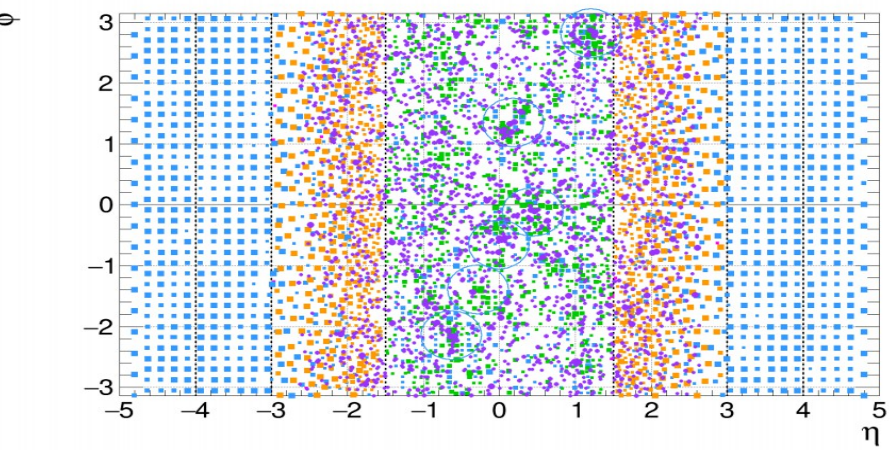


By virtue of being the last stage before the trigger decision we touch all detector components

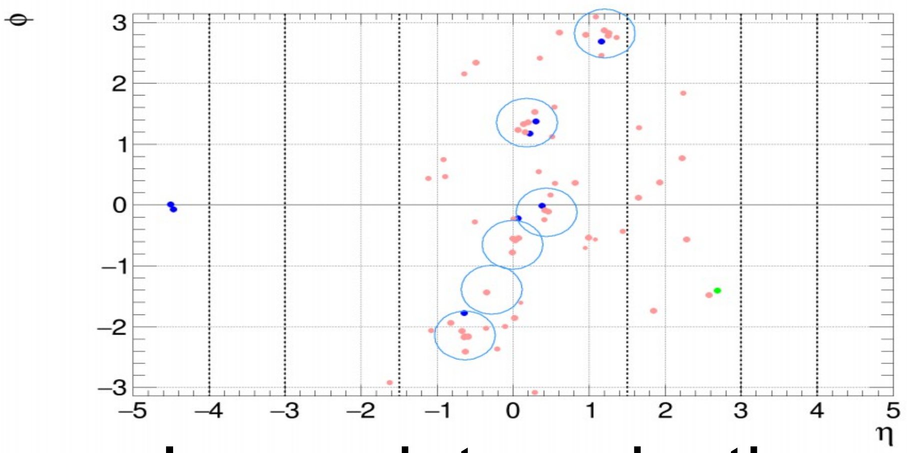
Reconstruction in Correlator layer 1

- With the correlator layer we are working to build
  - Particle flow candidates with L1 inputs
    - Combining tracks and calorimeter cells
  - PUPPI at level one
    - Primary vertex based reconstruction of charged/neutrals

L1 trigger inputs



PF+PUPPI output



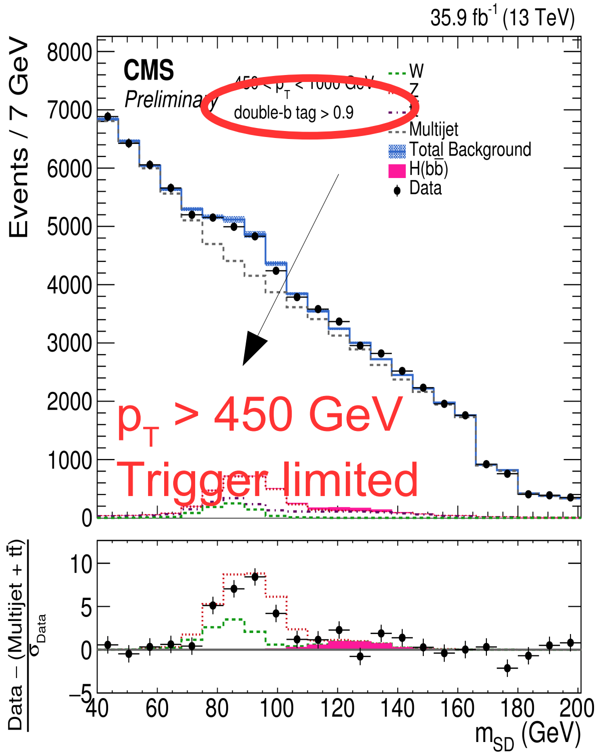
Large data reduction possible

Complicated FPGA algorithms

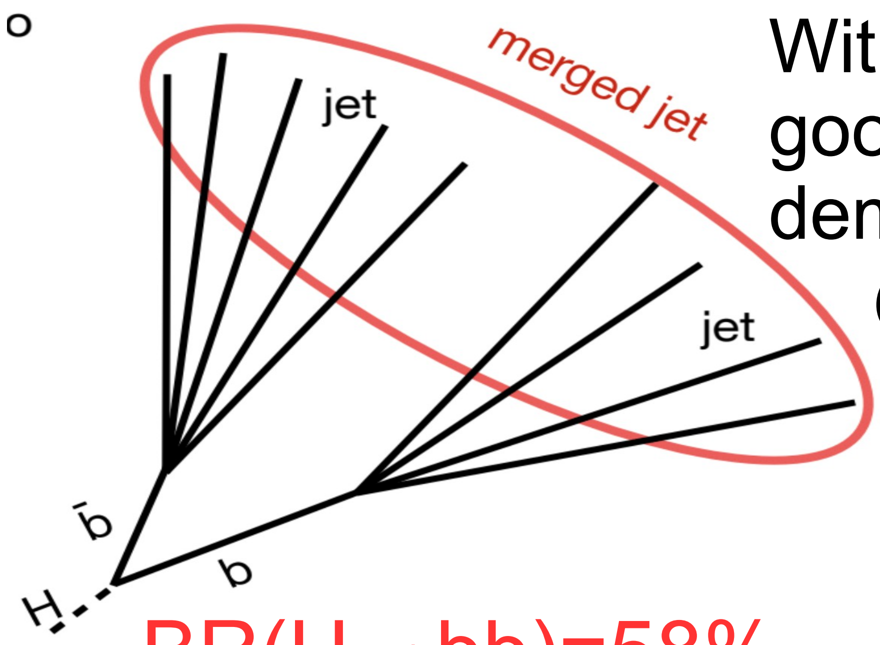
PF+PUPPI!

Reconstruction in Correlator layer 2

- With PF/PUPPI candidates
  - All the inputs to do offline physics equivalent algos
  - Many object ids in CMS/ATLAS use MVAs
    - We aim to have the same capability on L1
- Consider the trigger limited Higgs to bb:



p_T > 450 GeV
Trigger limited



With many objects good example to demonstrate ML @L1 Trigger

BR(H→bb)=58%

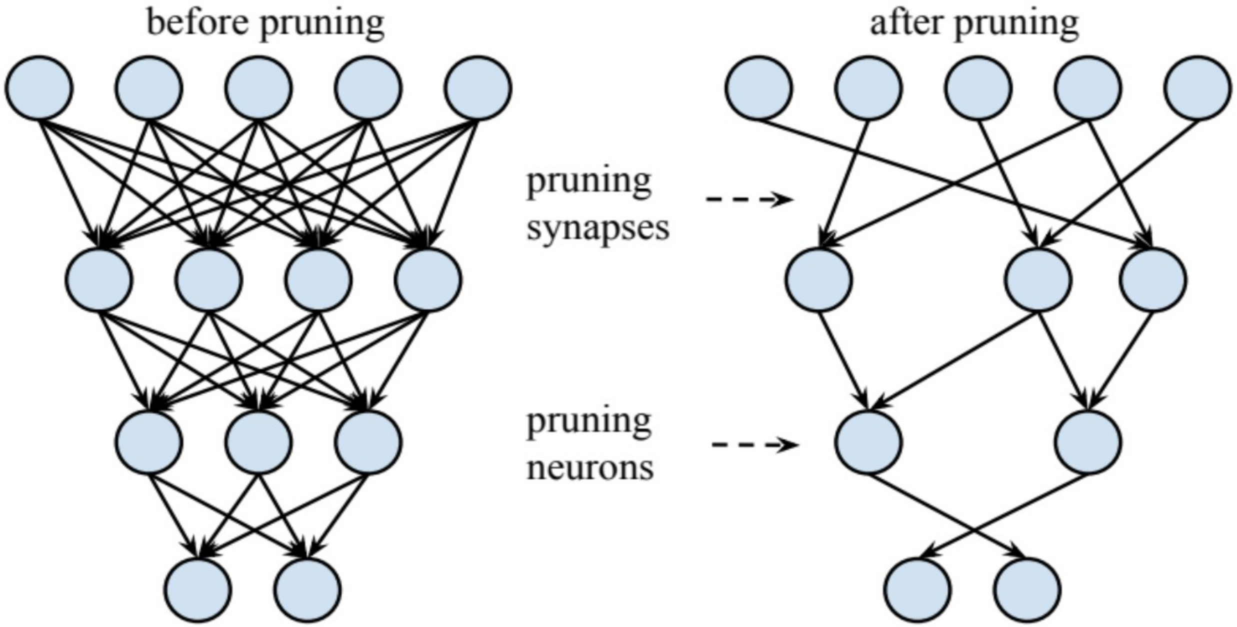
Enter: HLS4ML

- HLS : High Level Synthesis
  - C-like code that generates Verilog/VHDL
    - HLS code of 50 lines is easily VHDL code of 1000s of lines
  - Fast development of FPGA Algorithms
    - Some limitations: these are rapidly going away
  - L1 upgrade algorithms written primarily in HLS
- ML : Deep neural nets
  - Rapidly replacing object and physics ids at LHC
  - Many developments for jet physics & lepton ids
  - Serious consideration for fast inference demands

HLS + ML = HLS4ML (our project)

ML@FPGA

- Machine Learning on FPGA an emergent field
  - Over past two years has been considerable development
  - Demonstrated large speedups in inference @same power
    - Developments reduced precision training/network compression
    - Pipelining of data and resource sharing



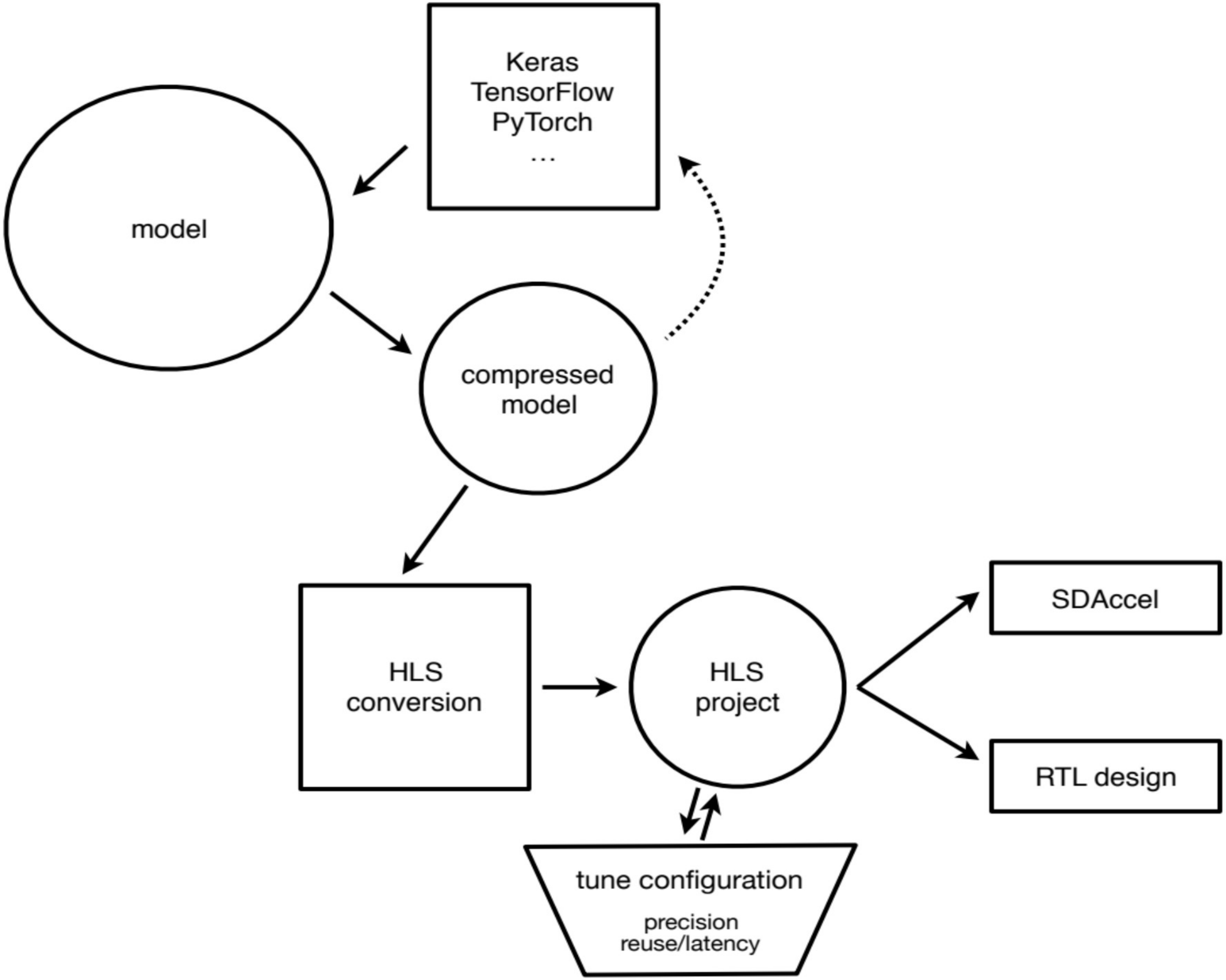
Speed up gains on complicated networks
> 100x CPU
> 10x GPU
Larger for smaller MLs

Power consumption:
~1 CPU
~1/10th GPU

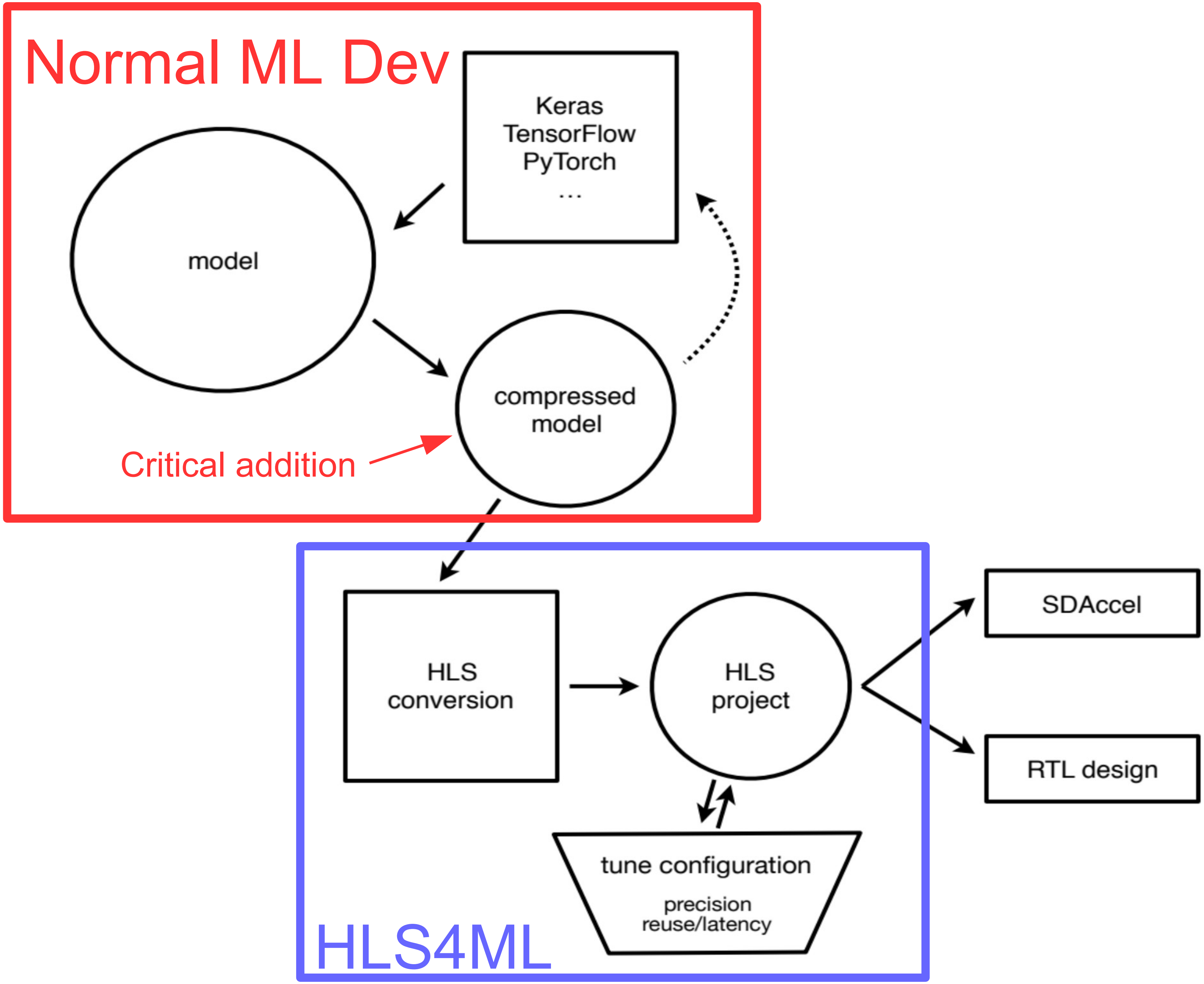
In collaboration with google/MIT (S. Han)

In collaboration with FPGA developer EJ Kreiner

Current design

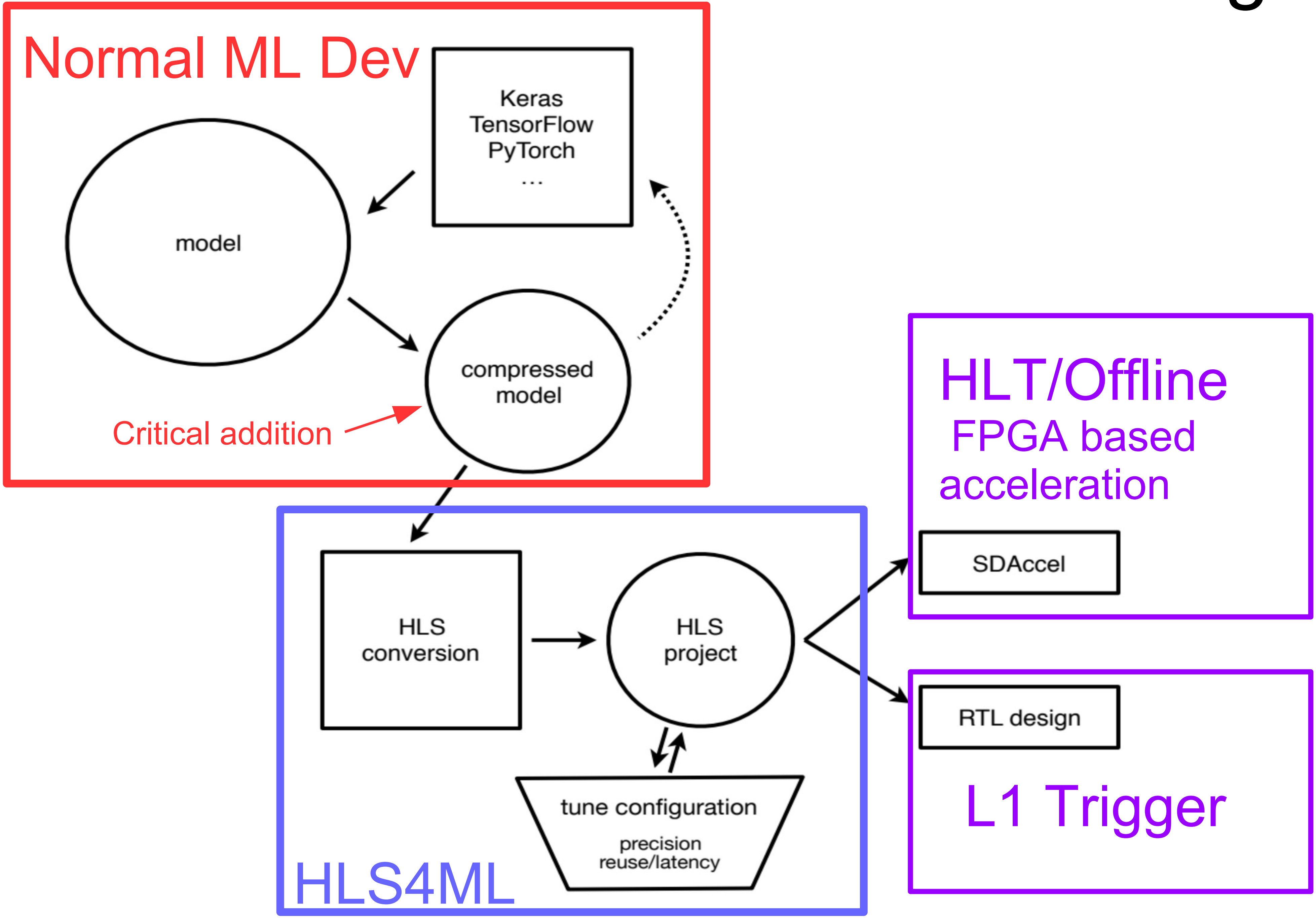


Current design



In collaboration with FPGA developer EJ Kreiner

Current design



Normal ML Dev

Keras
TensorFlow
PyTorch
…

model

compressed model

Critical addition

HLS4ML

HLS conversion

HLS project

tune configuration

precision
reuse/latency

HLT/Offline

FPGA based acceleration

SDAccel

RTL design

L1 Trigger

In collaboration with FPGA developer EJ Kreiner

Understanding Use Case

FPGA acceleration(co-processor)

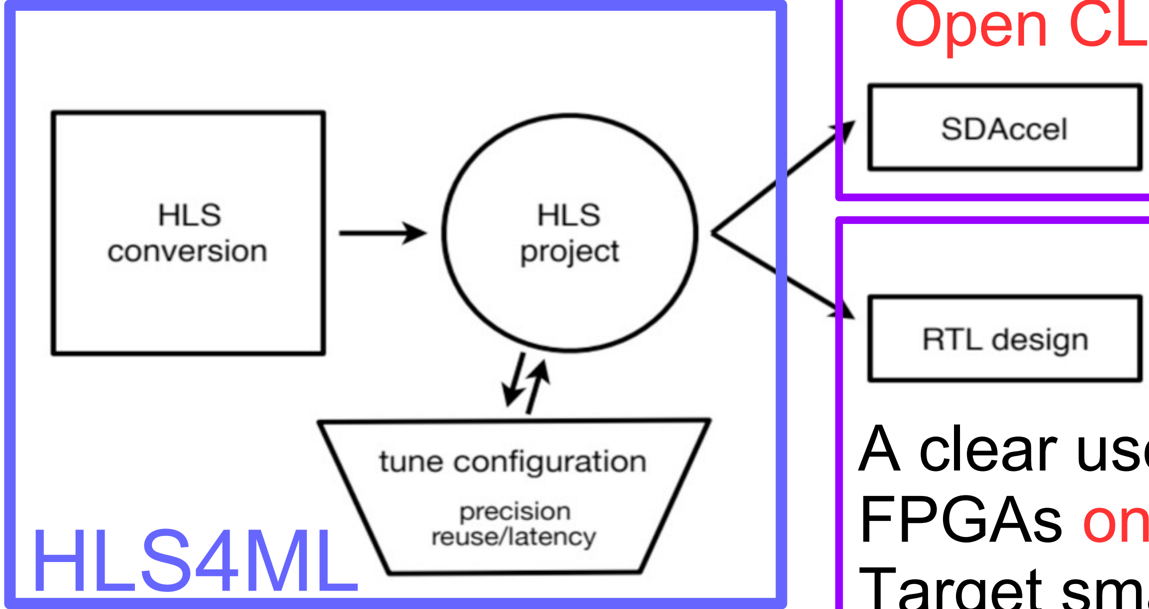
Commercial developments :

- Microsoft Brainwave : very developed
- AWS F1 instance : bare metal

Other tools based on Open CL exist

- Open CL has more limitations than HLS

Looking to expand direction



L1 trigger

A clear use case

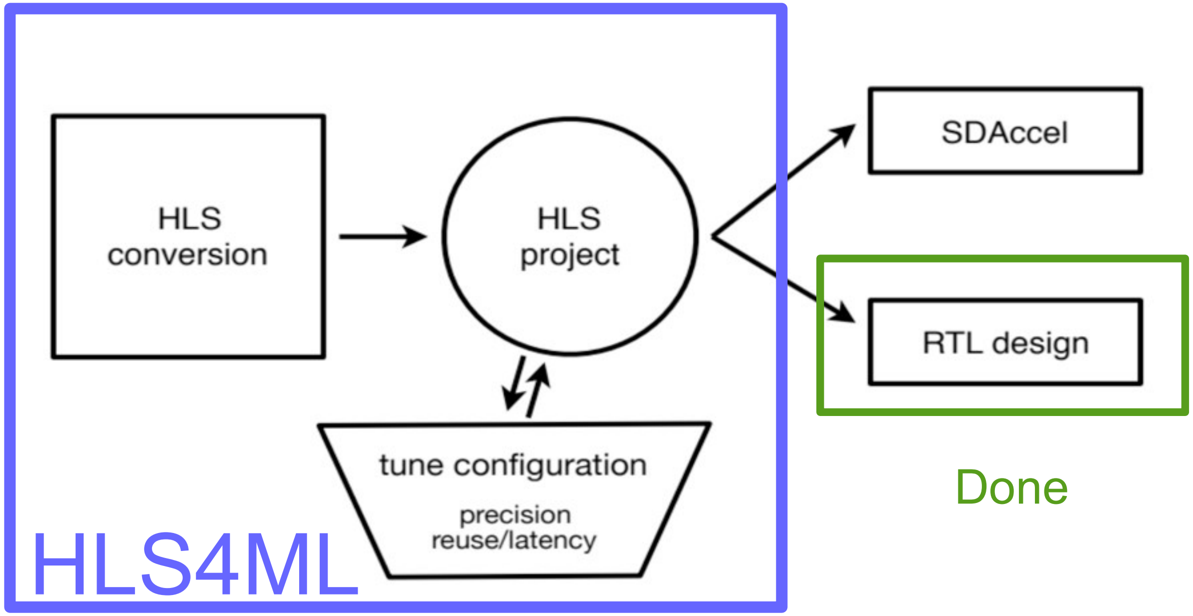
FPGAs only option for low latency

Target small networks/high throughput

We are first to go in this direction

Current status

- We have run the full chain for dense networks
  - From Keras model to testing it on an FPGA
    - Working to extend the network to CNN/LSTM
    - Actively working to improve network compression



Rapidly developing

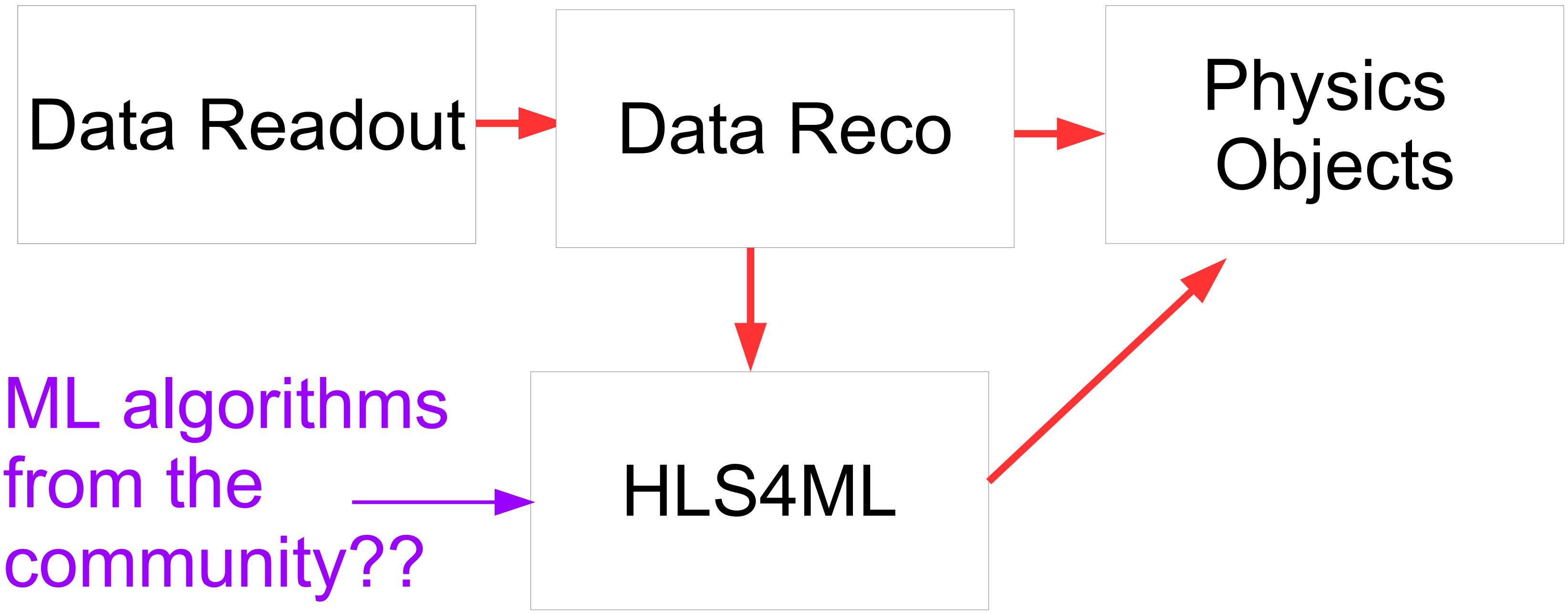
Done

Aiming for a first set performance studies+code by March

Pontification

- Going beyond our basic studies...
  - Hardware co-processor
    - Push from industry to establish hardware accelerators
    - Not clear whether FPGA/GPU are the right topic
      - FPGA based acceleration is less developed (with more potential)
  - Would like to extend tools to other examples
    - Tracking? Vertexing?
  - Large networks and low latency
    - Can we extend to multi-FPGA based Ais
      - Giant Ais would require distributing the inference on FPGA network
      - Microsoft Brainwave has a few examples
  - Usage of FPGA clusters
    - Do we stand to gain from AWS/Brainwave prototyping?

Synergies: FPGA co-processor



Data Readout

Data Reco

Physics Objects

ML algorithms from the community??

HLS4ML

- Looking into accelerator tests on existing systems
  - Comparing performance against GPU is a 1st start
  - This is an area that can benefit the community
- AWS vs. MS Brainwave: would like to explore both
  - Microsoft provides a bit of infrastructure and support
  - AWS is a “bare-metal” resource with limited infrastruct.