HiPeCTA – High Performance analysis for CTA

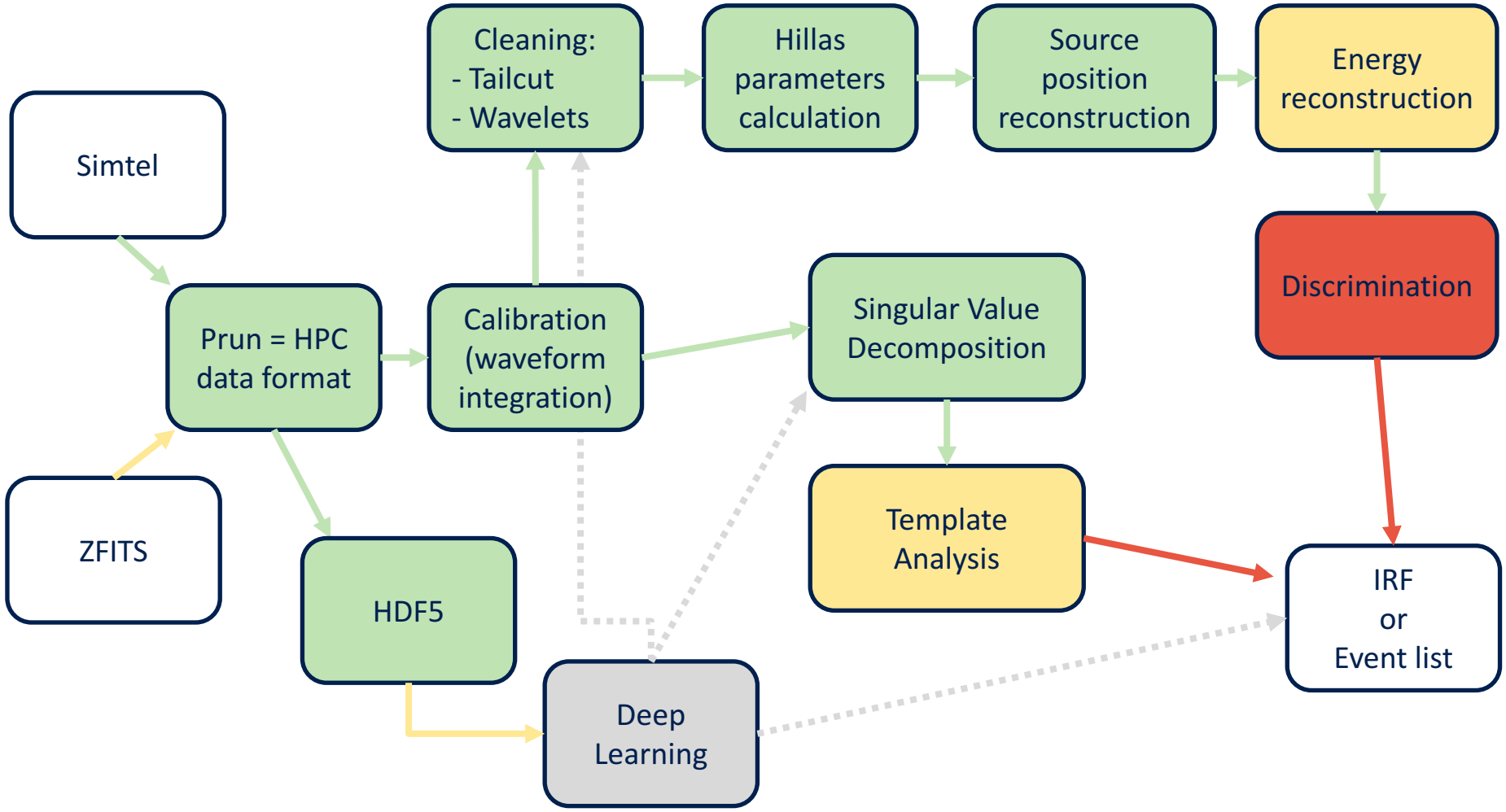
CEA-LAPP meeting, January 2018
Thomas VUILLAUME on behalf of the LAPP, CNRS
Thomas Vuillaume, Pierre Aubert, Jean Jacquemier, Gilles Maurin, Florian Gaté, Giovanni Lamanna

ASTERICS is a Horzon 2020 project supported under grant agreement n.653477

High Performance Computing (HPC)

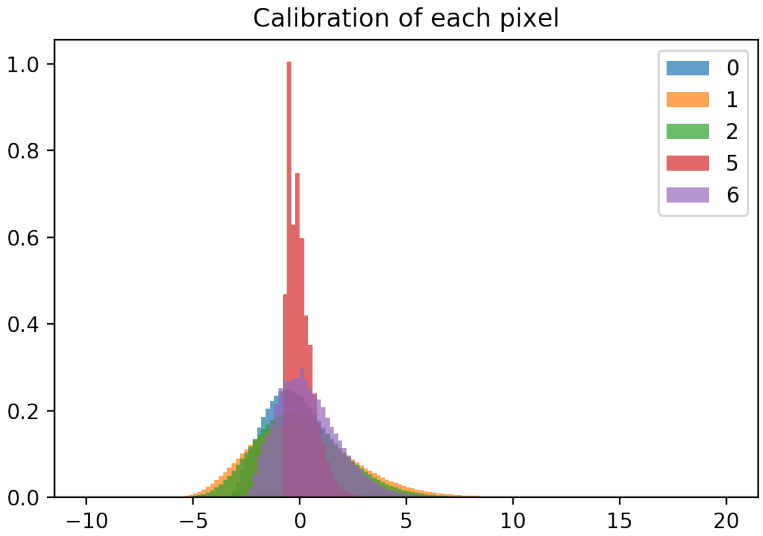
- Goal:
  - Optimize computing resources
- Use :
  - CPU data-prefetching
  - CPU vectorization
- Needs :
  - Optimized data format → See Pierre's slides
  - No special computer

HPC pipeline blocks



HPC Calibration and Cleaning

- Calibration from waveform signal and cameras' refshape
  - consistent results with ctapipe



- Tailcut cleaning
- Wavelet cleaning

HPC Hillas - computing performances

- Based on HPC generic and open-source library* developed at LAPP
- Example on signal amplitude calculation:

	Speed (cy/el)	Speed up
Non-vectorized	2.70	1
Vectorized (GCC, SSE4)	0.70	3.8
PLIBS_8 vectorization on SSE4	0.23	11.9
PLIBS_8 vectorization on AVX	0.11	23.7

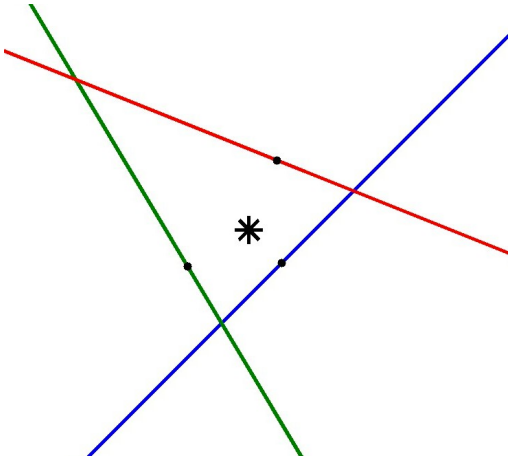
- Geometric reconstruction: 327MB/s when data is in RAM = 4475 events/s
  (2,7 GHz Intel Core i5)

*PLIBS_8 : https://gitlab.in2p3.fr/CTA-LAPP/PLIBS_8.git

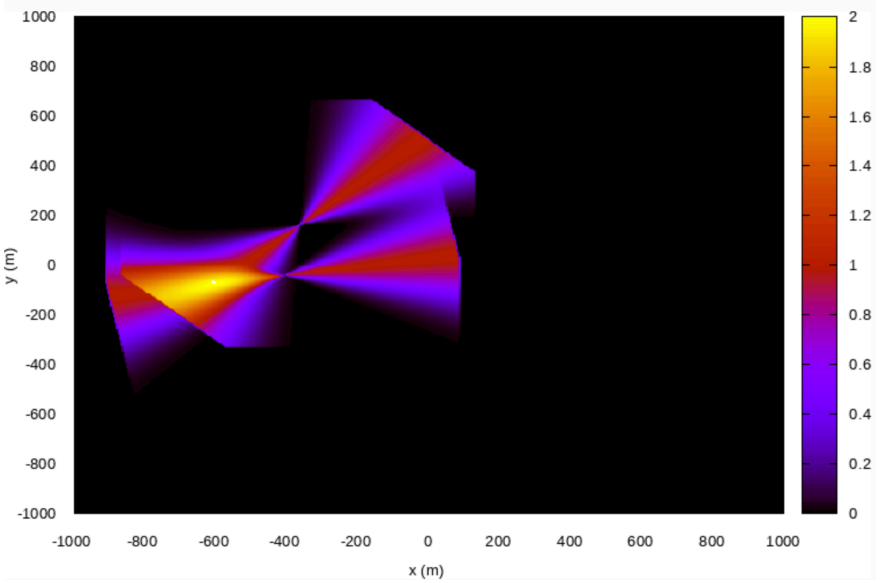
HPC – Hillas based methods

Several methods to reconstruct the angular direction and impact parameter:

- Standard reconstruction: lines intersection two by two and weighted average
- Multiline:
  - distance minimization to each line
  - one matrix computation

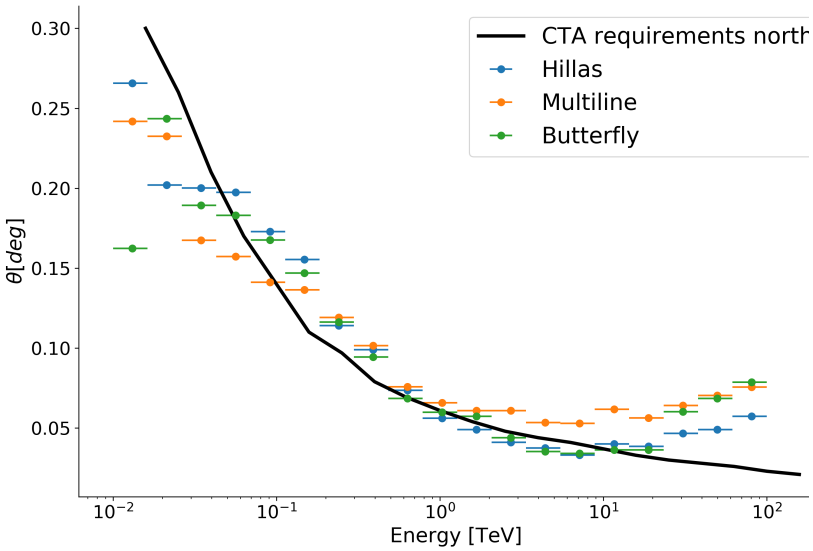


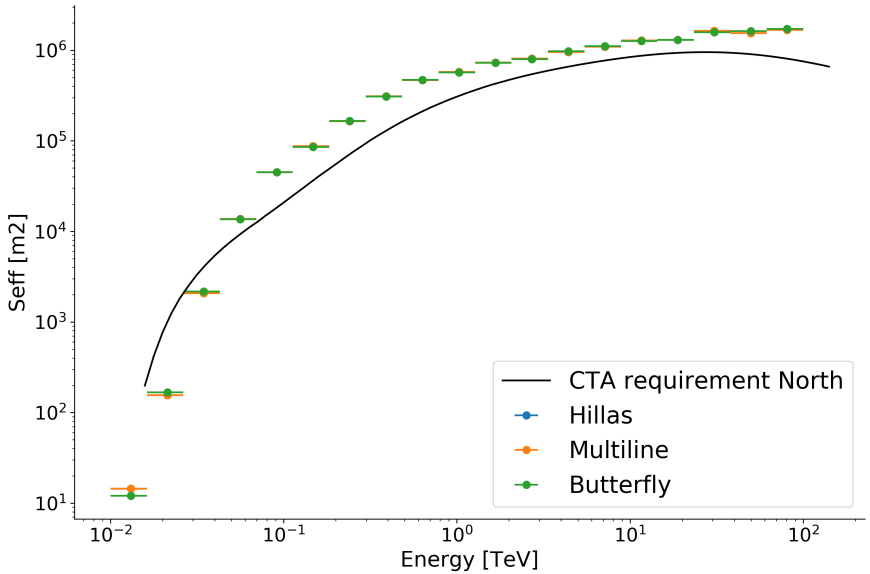
- Butterfly: create a grid on the ground and compute the ”field of view” of each telescope on the ground based on Hillas parameters. Find most probable intersection point.



HPC Hillas reco – physics performances

La Palma, layout: CTA.prod3Nb.3AL4-BN15





HPC energy reconstruction and discri

HPC neural network library

- From scratch
- Open-source
- Includes:
  - Neural network generator
    - HPC friendly (data pre-fetching, vectorization…)
    - User friendly: from simple configuration file
  - Simple data format to train your neural network in Python

- Training can be done with other tools (e.g. scikit-learn) and transferred to HPC network

HPC Hillas – on-site analysis

Conclusion

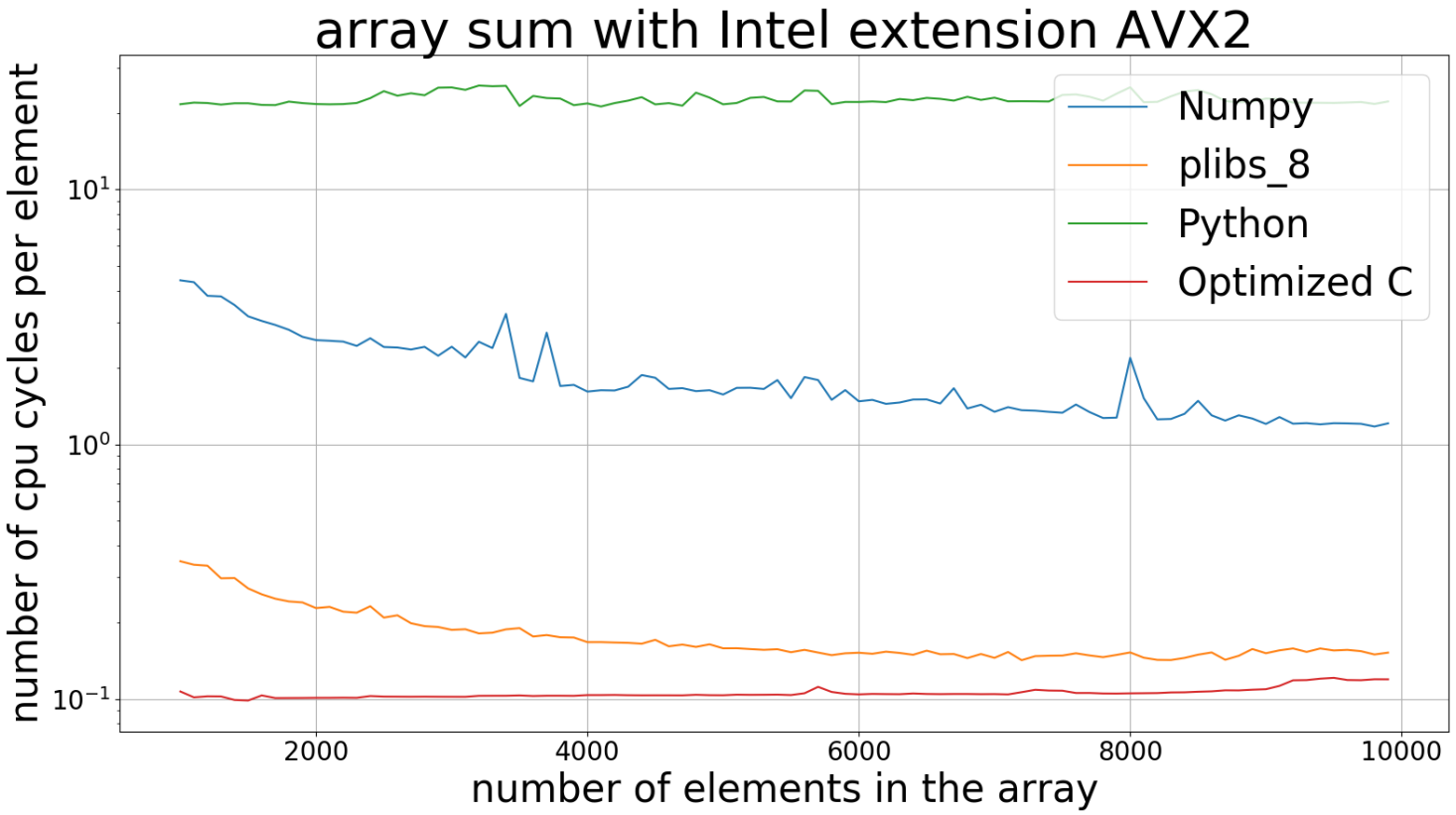
- Complete Hillas-based reconstruction doable on-site
- Probably achievable online

Bringing HiPe to ctapipe

- HPC performances through Python wrapping
  - Huge development effort…
  - … but very simple and transparent calls for developpers (same as using a numpy array – all numpy features included)
    - import plibs_8
    - tab = plibs_8.zeros(42)

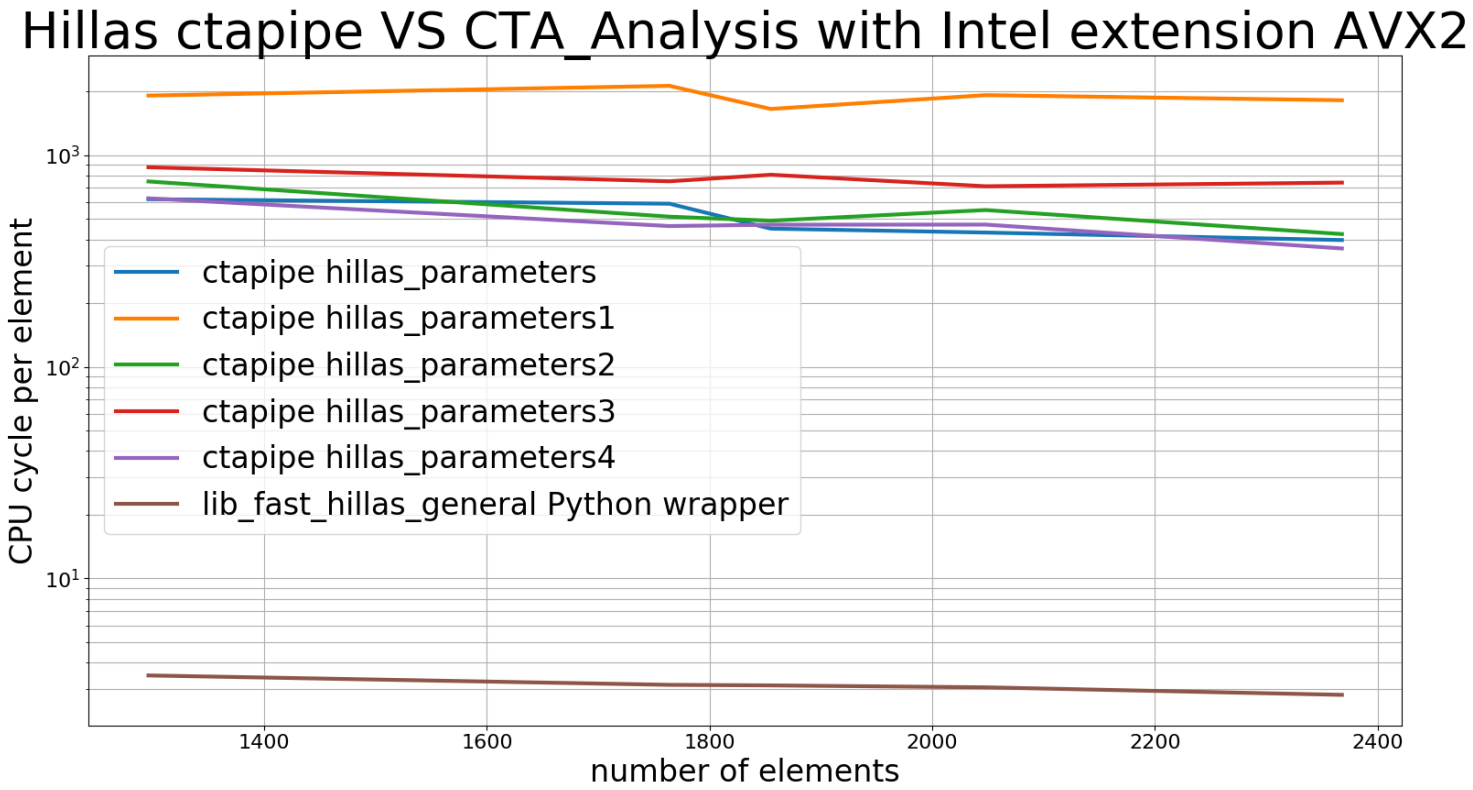
HPC in ctapipe

- Comparison of an array sum called in Python :



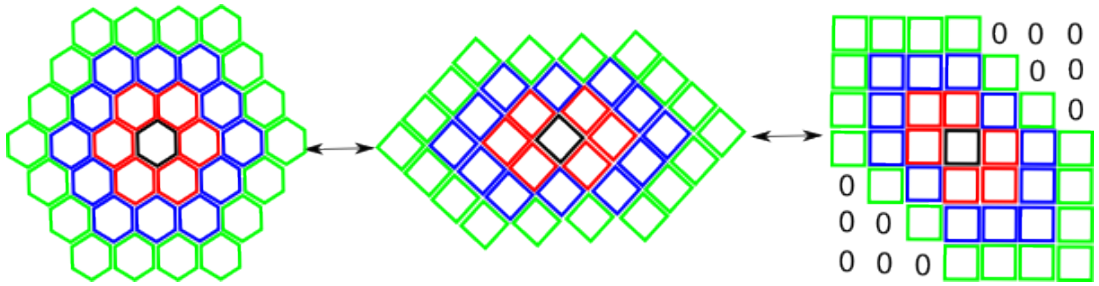
HPC in ctapipe

- Benchmark on Hillas parameters calculation.



Bringing HiPe to ctapipe

- Already wrapped:
  - Images transformation to square matrix
    - Automated on any camera
    - Contiguous pixels in real life are contiguous in memory -> HPC calculation possible

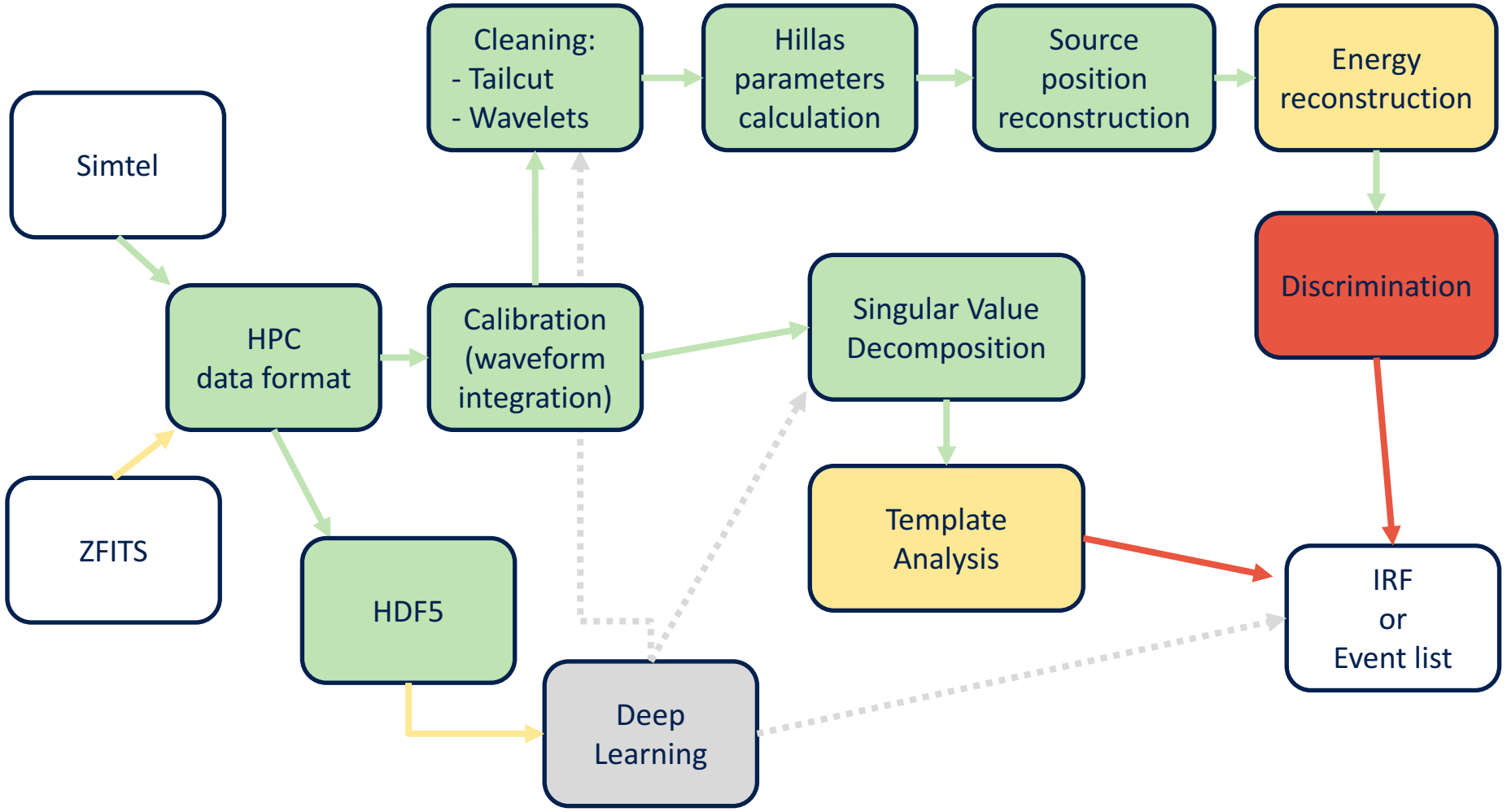


  - Calibration
  - Hillas parameters calculation
  - Cleaning on going

Bringing HiPe to ctapipe

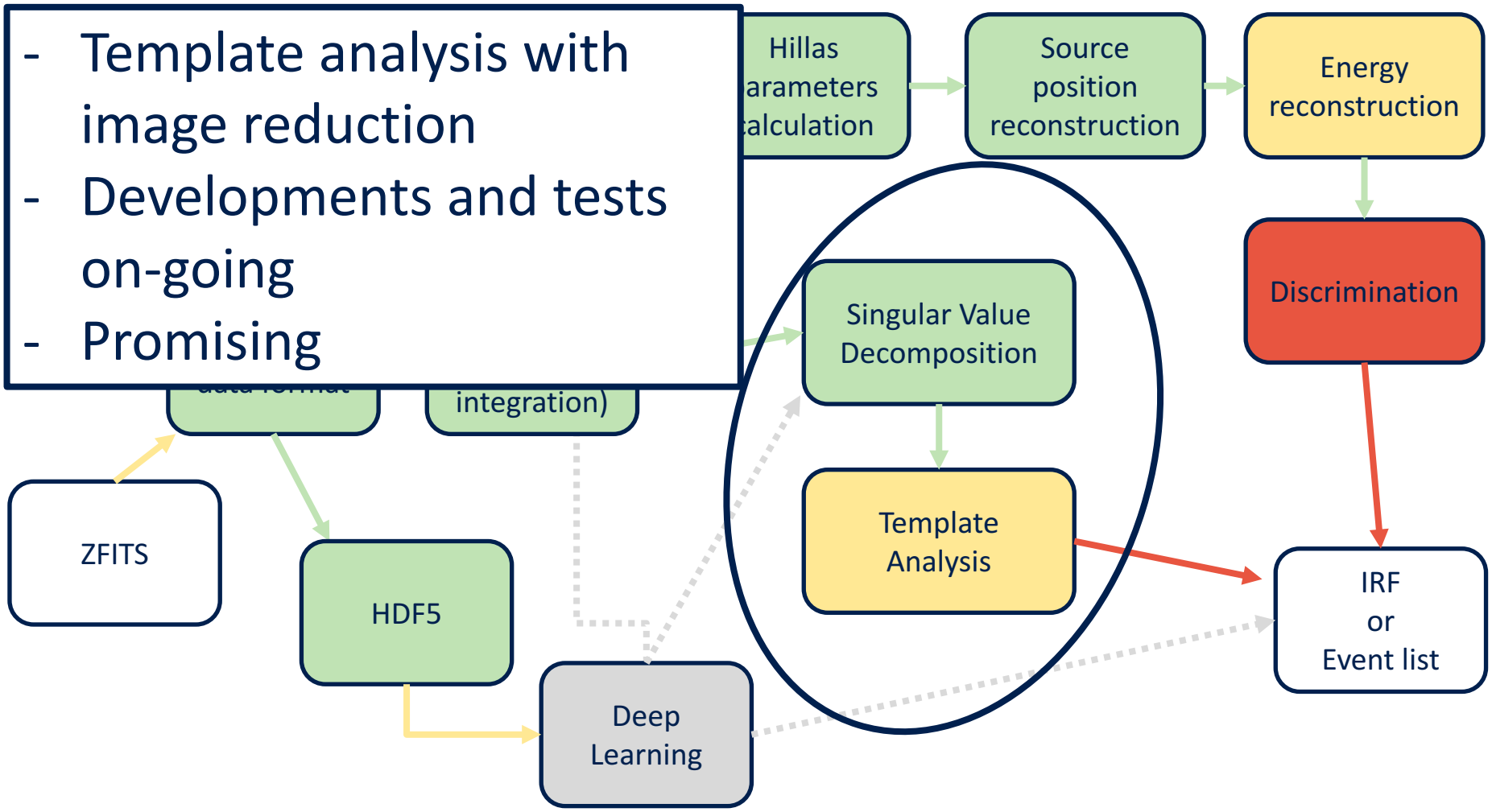
- All callable in ctapipe
- BUT needs a HPC friendly data format – or at least aligned Numpy arrays
  - Not the case today in ctapipe
- On-going effort:
  - converter HPC data format → ctapipe containers
    - to be able to replace any part of the analysis and test them separately
  - Studying what would it take to make containers HPC friendly

HPC pipeline blocks

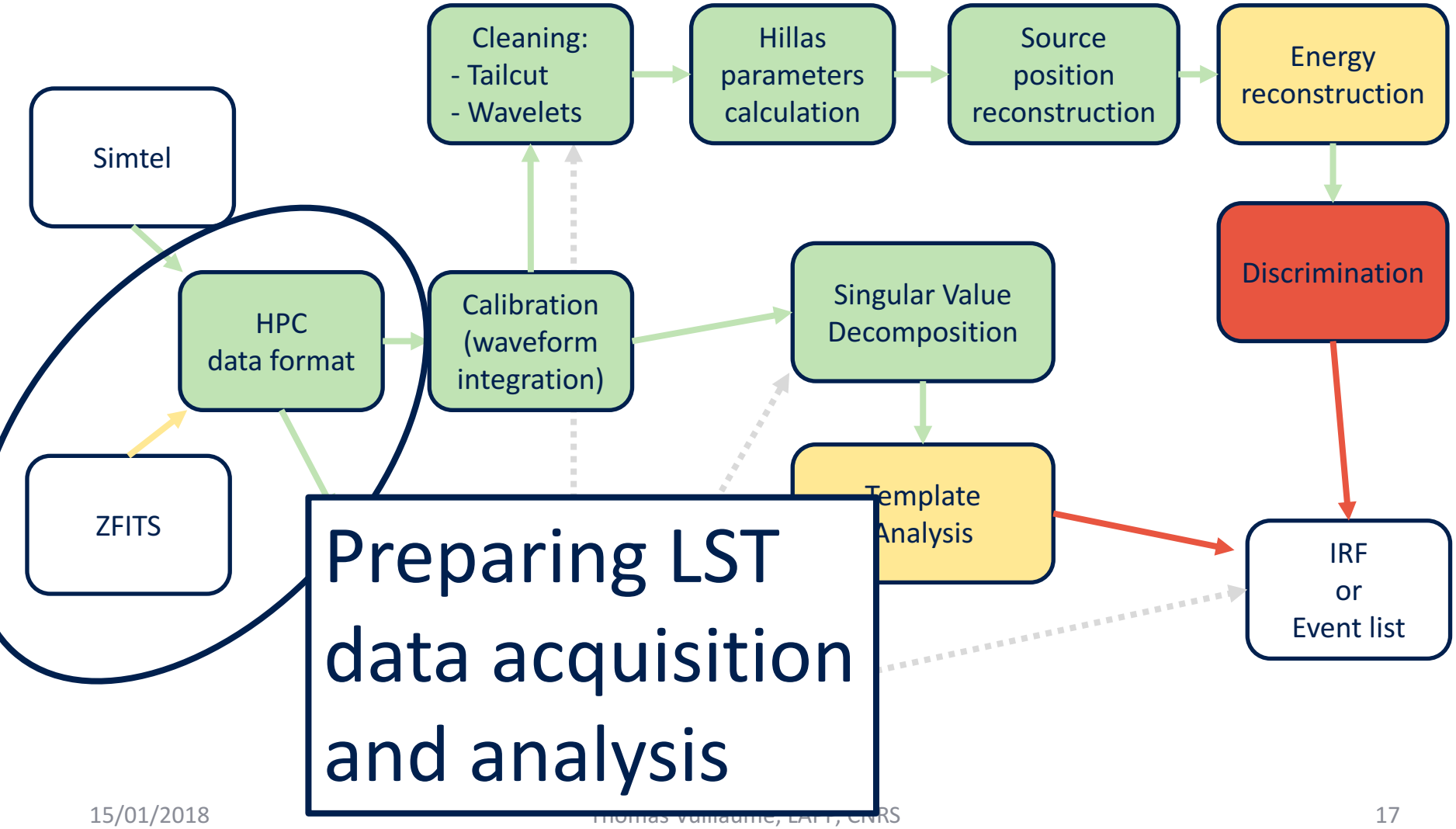


HPC pipeline blocks

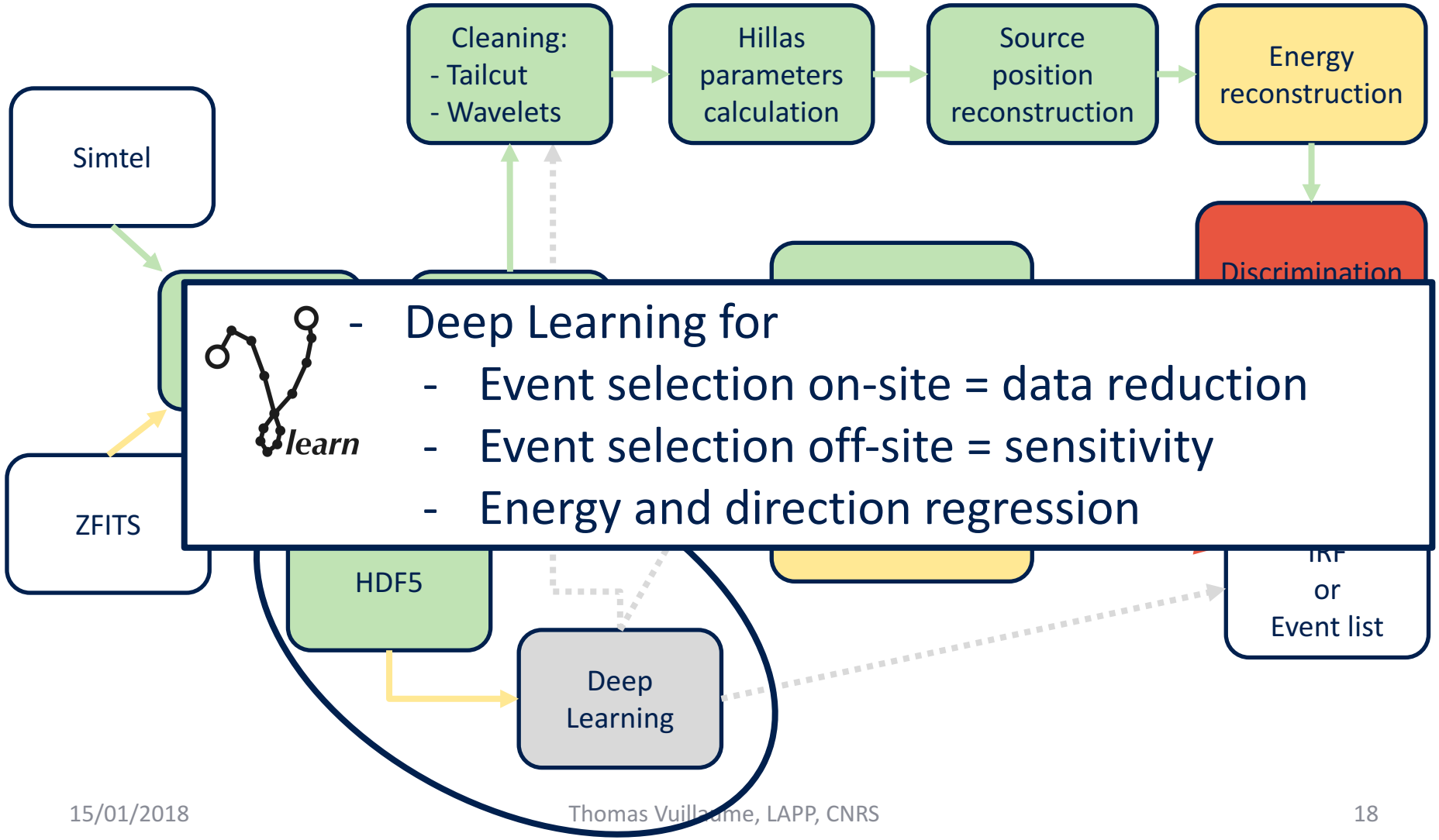
- Template analysis with image reduction
- Developments and tests on-going
- Promising



HPC pipeline blocks



HPC pipeline blocks



HPC pipeline blocks