

CERN SUMMER STUDENT
PROGRAMME

Refactorization of MTD DAQ Software

By Sana Ruknudin KHATTAK

Supervisor: Özgür Sahin

Date: 07 August, 2024



OUTLINE

- 01 LHC Timeline
- 02 CMS
- 03 MTD
- 04 Data Flow
- 05 DAQ Software
- 06 MTD Timeline
- 07 My work
- 08 Conclusion

LHC TIMELINE: RUNS AND PHASES

RUN 1

2010- 2013

- Initial physics run
- Energy: 7-8 TeV
- Discovery of Higgs boson

LS-1

2013-2014

PHASE-1
UPGRADES

RUN 2

2015-2018

- Energy: 13 TeV
- Improved Higgs measurements

LS-2

2019-2021

*Upgrades
and
Maintenance*

RUN 3

2022-2025

- Present moment
- Record collision of 13.6TeV
- Probing for Physics BTSM and precision measurements

LS-3

2026-2028

PHASE-2
UPGRADES

HL-LHC

2029-2041

5-7.5 times higher
instantaneous
luminosity than
LHC's nominal
value

Our Focus: CMS

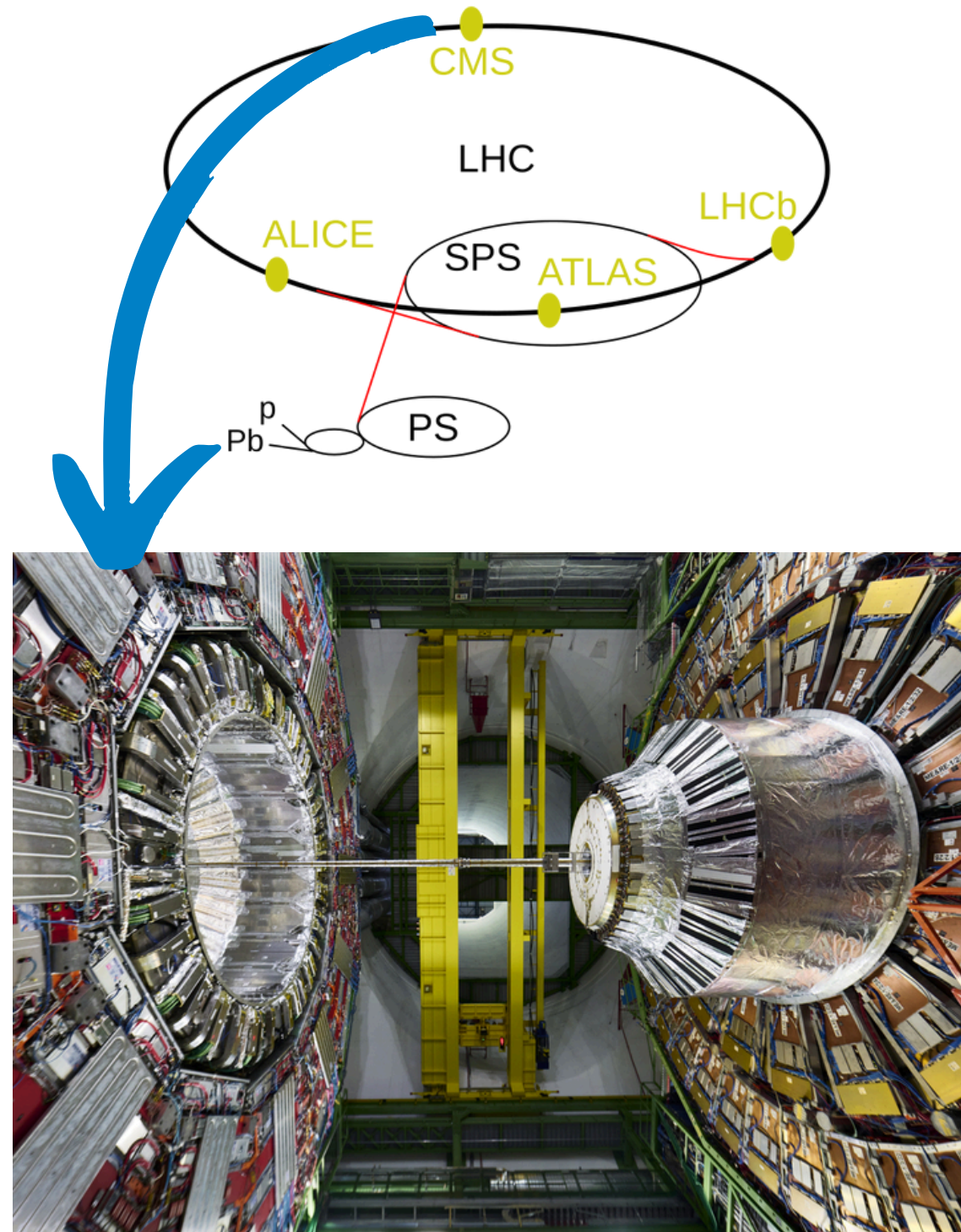


Fig.[1] : Views on the open CMS detector to be closed up after the Long Shutdown 2 (LS2).

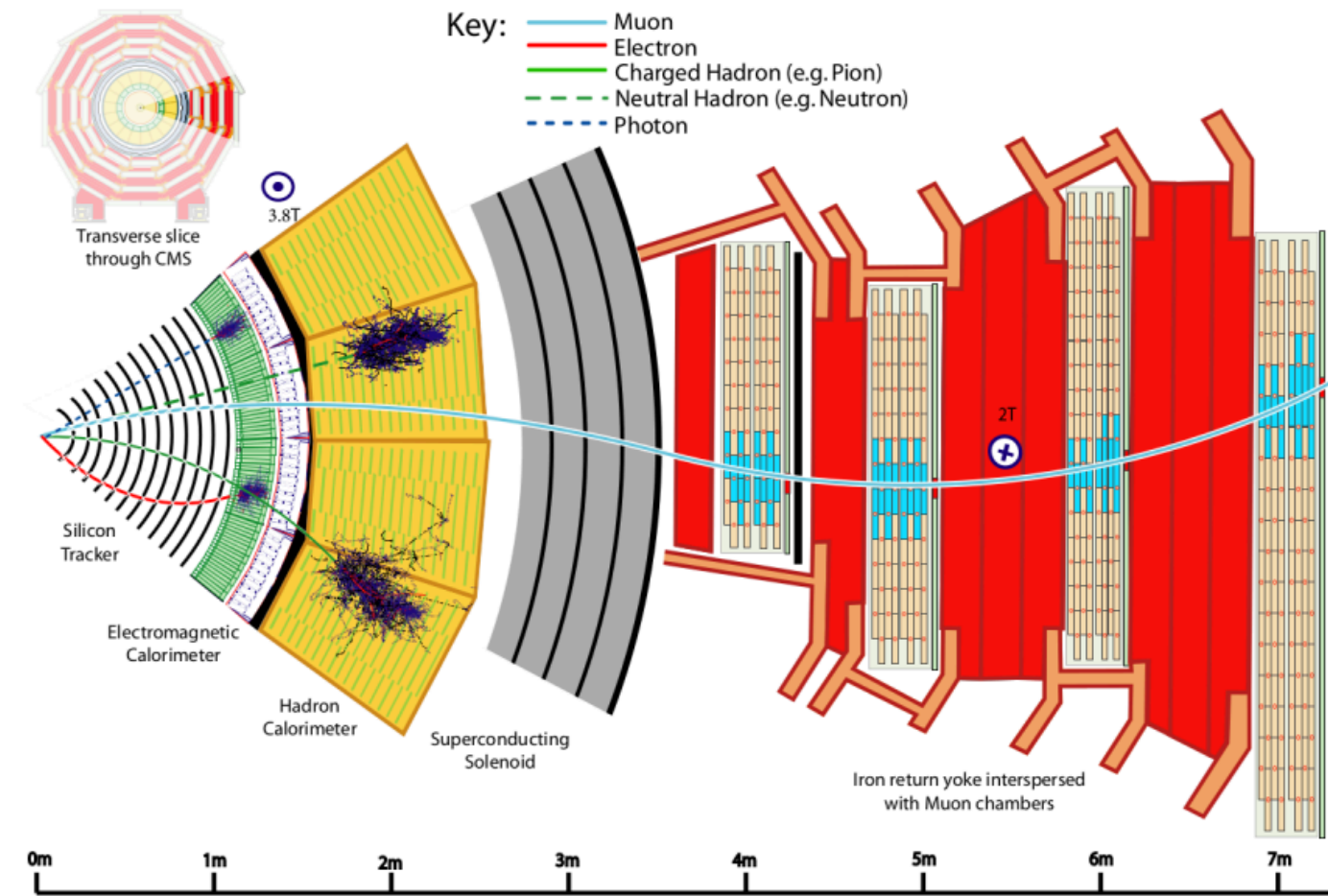
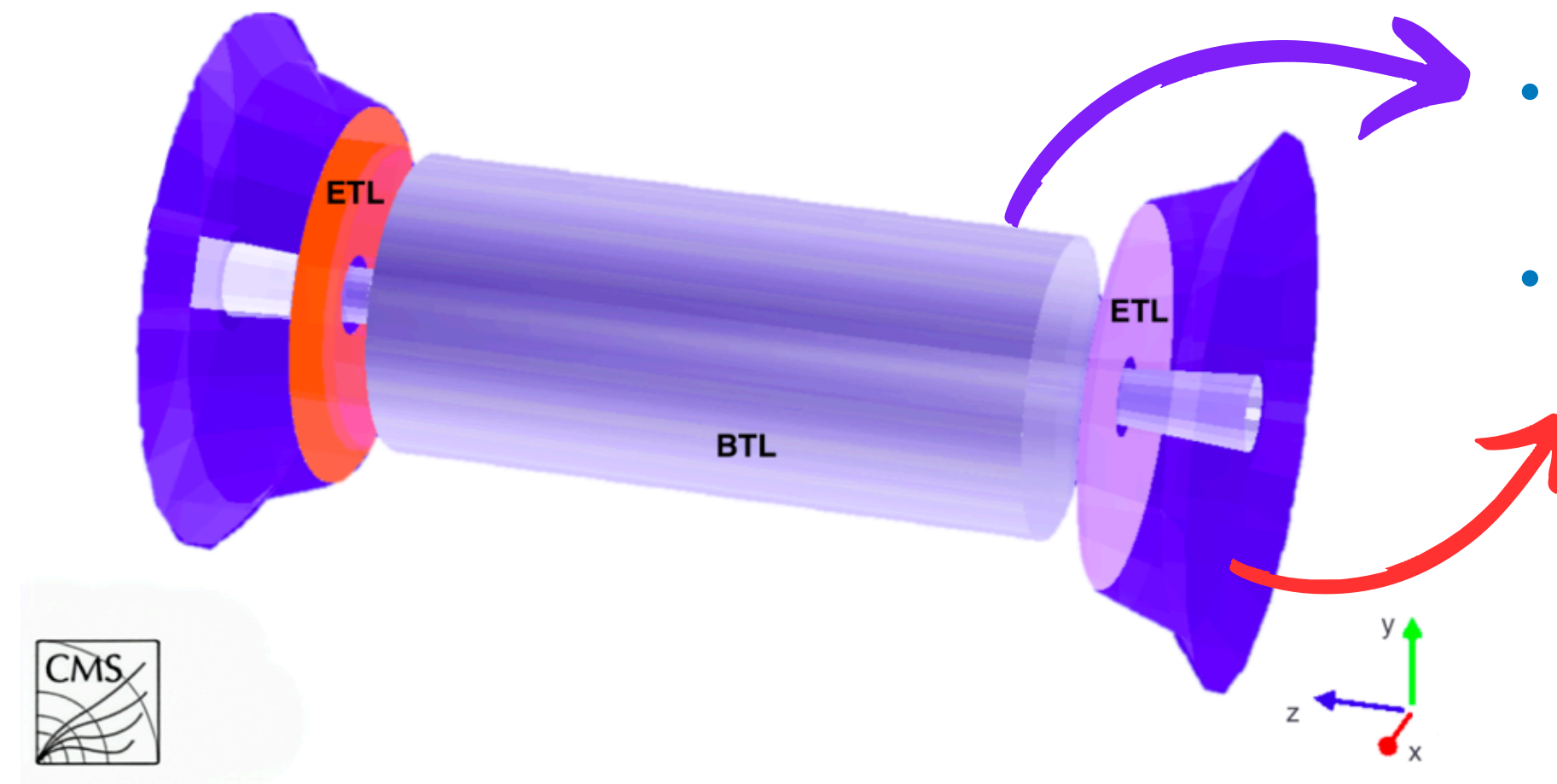


Fig.[2]: Slice of CMS in the transverse view and experimental signature of particles in the sub-detectors.

PILE-UP MITIGATION WITH MTD:

- Higher Luminosity---> Denser Beams -----> Result: Pileup
- Solution: **MIP Timing Detector:**

MTD will measure the time-of-arrival of each charged particle with a time resolution of about 30ps.



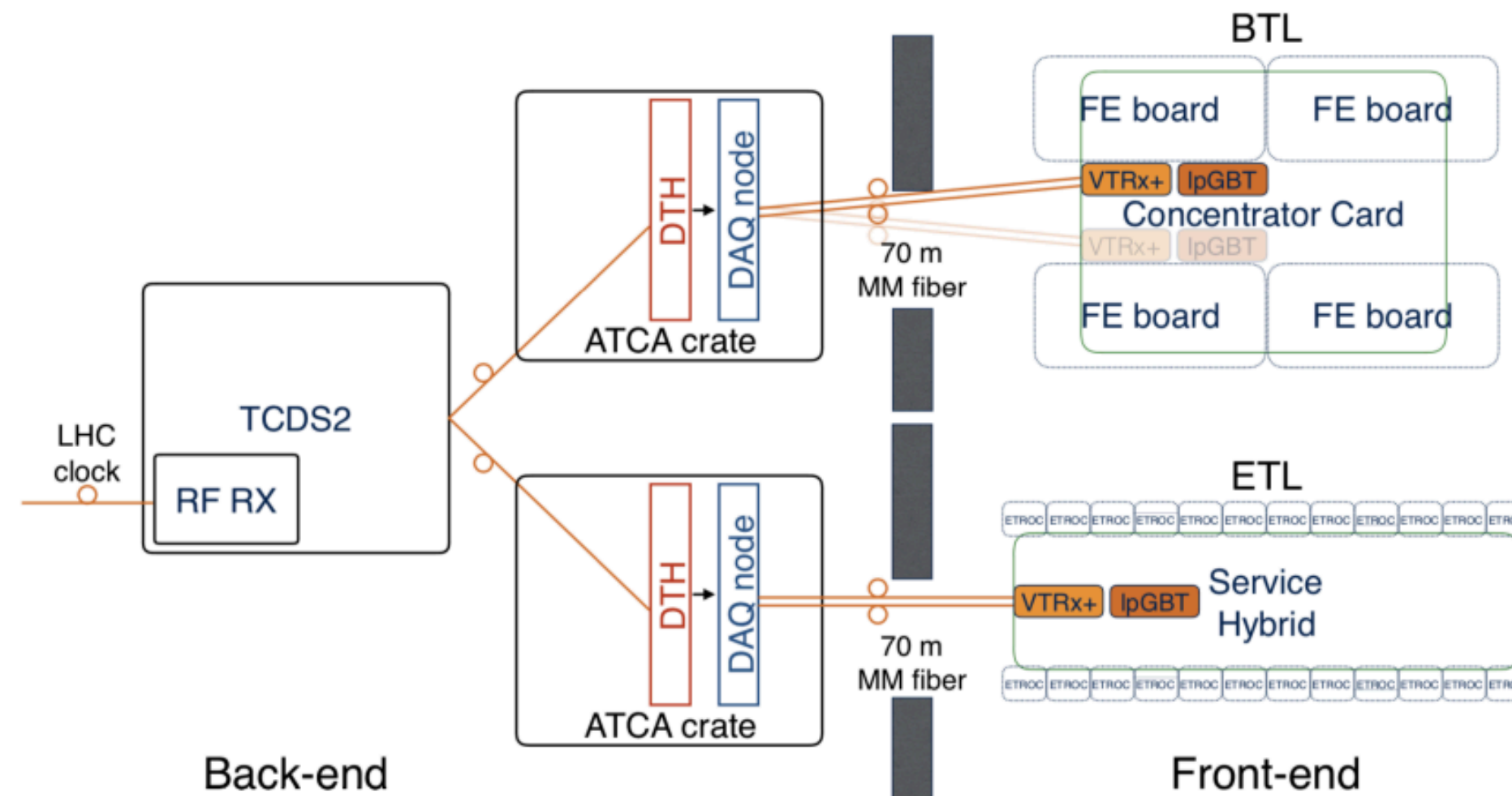
- The BTL has a surface area of 38 m^2 with SiPMs
- For ETL, each disk has a sensitive area of 7.9 m^2 . LGAD are chosen as the sensitive element.



Fig.[3]: A schematic layout of the MIP Timing Detector created in Geant4.

DATA FLOW

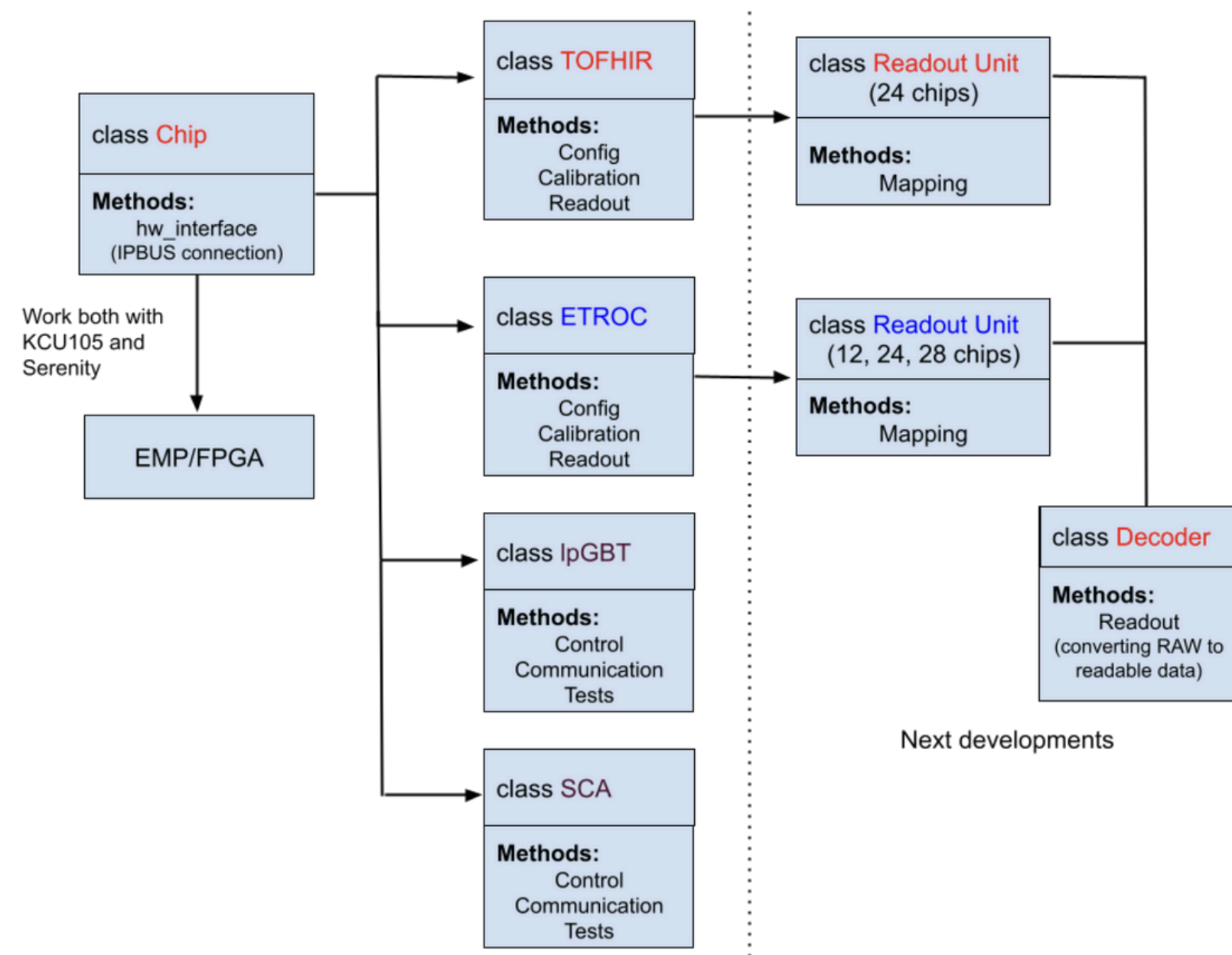
- Front end: ASICs for Readout (TOFHIR & ETROC)
- Back end: ATCA structure (Serenity)
- Links: Versatile Links like LpGBT and VTRx+



Fig[4]: Data flow through the main systems for the MTD detector

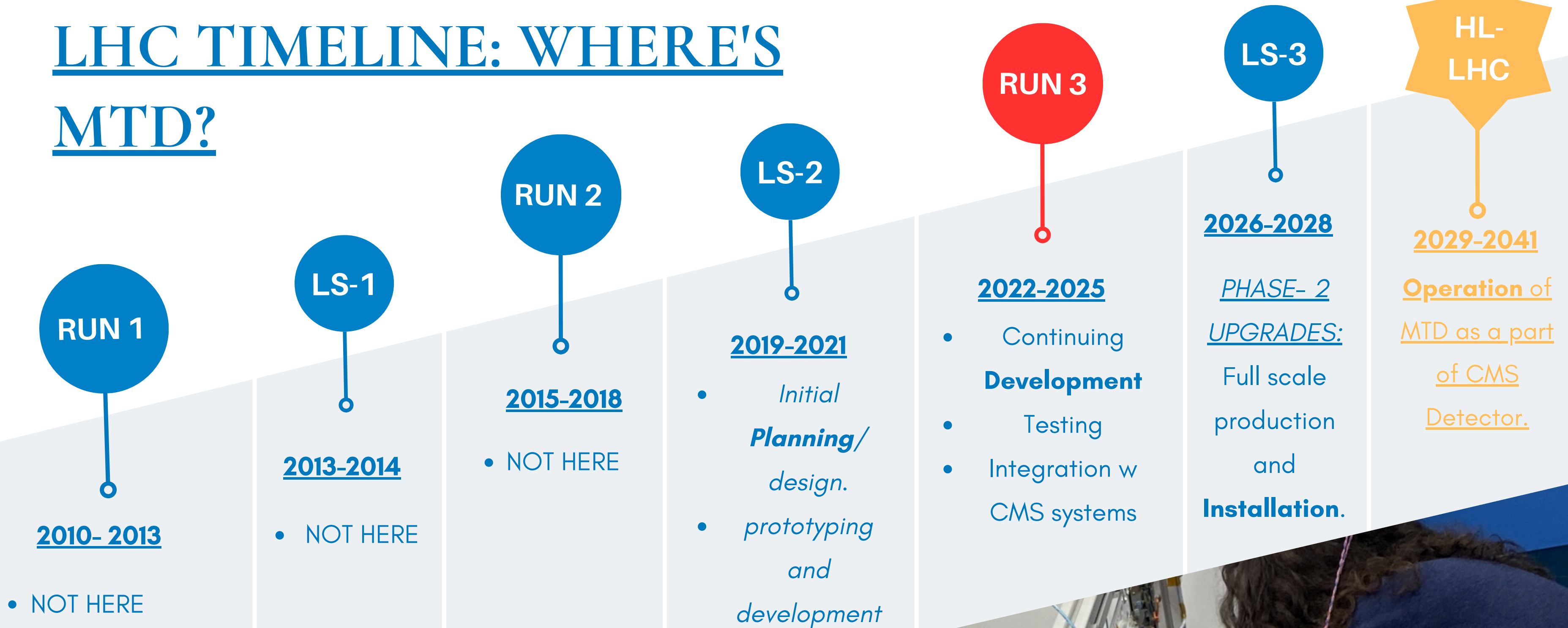
DAQ SOFTWARE

- The MTD DAQ software is a modular framework accommodating different communication protocols with large variety of radiation hard chips positioned on MTD.



Fig[5]: DAQ framework for the MTD detector.

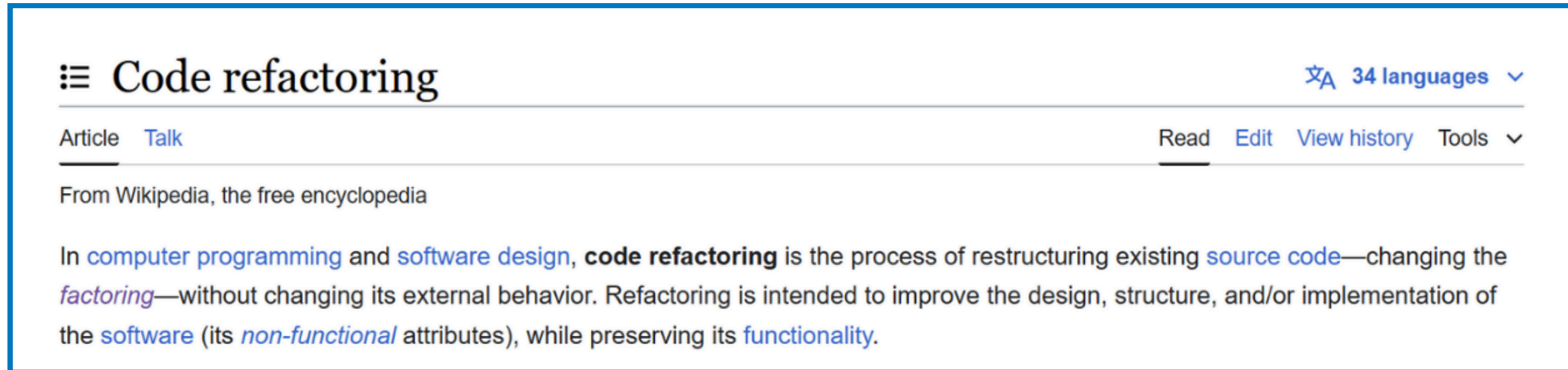
LHC TIMELINE: WHERE'S MTD?



We are currently moving from the development to the production phase



My work: Refactorization of code and documentation



The screenshot shows the Wikipedia article for 'Code refactoring'. At the top, it says 'Code refactoring' with a language dropdown set to '34 languages'. Below the title are links for 'Article' and 'Talk', and a menu with 'Read', 'Edit', 'View history', and 'Tools'. The main text reads: 'From Wikipedia, the free encyclopedia' followed by a paragraph: 'In computer programming and software design, **code refactoring** is the process of restructuring existing **source code**—changing the *factoring*—without changing its external behavior. Refactoring is intended to improve the design, structure, and/or implementation of the **software** (its *non-functional* attributes), while preserving its **functionality**.'

Fig.[6]: Definition of Code Refactoring.

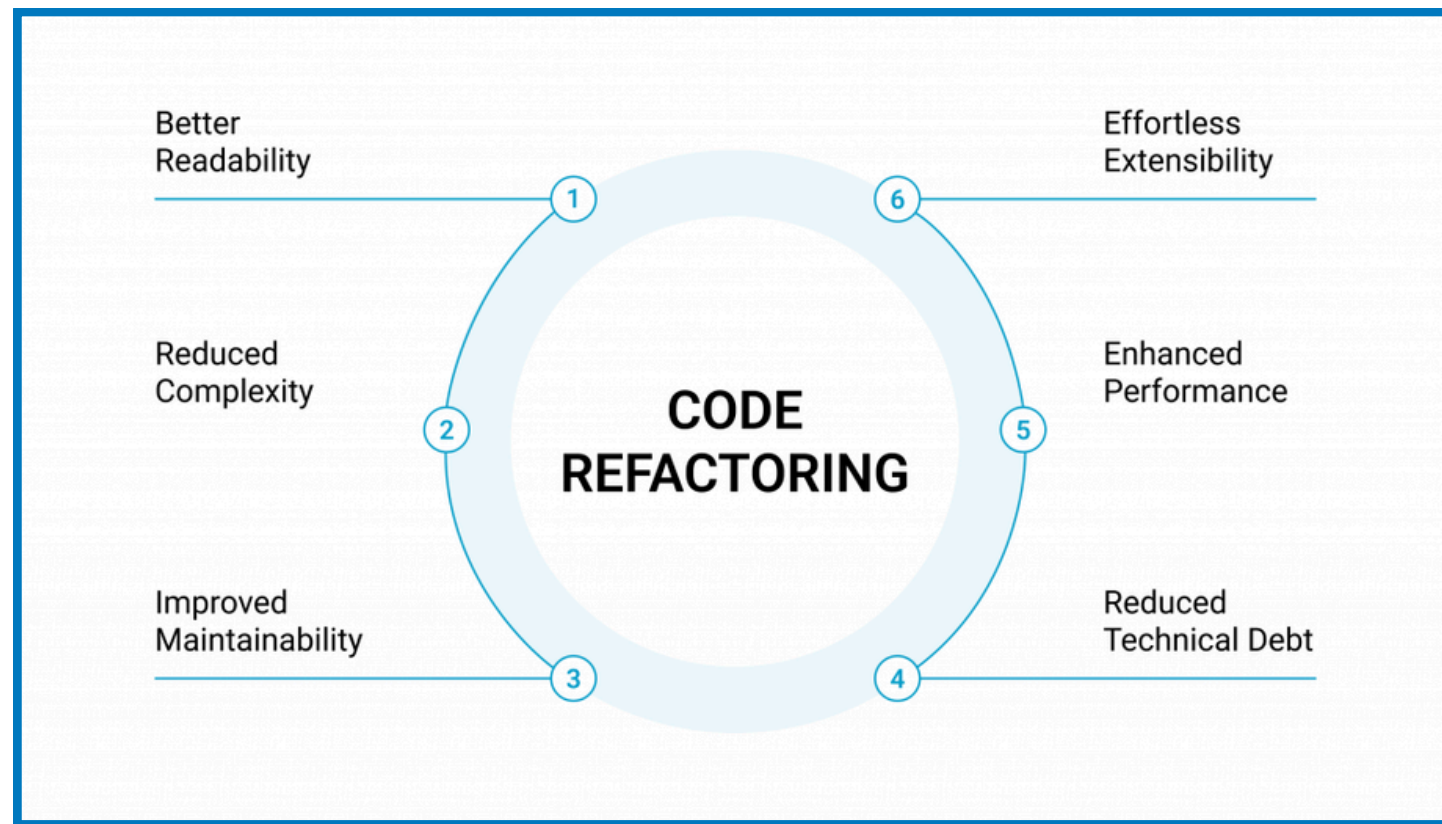
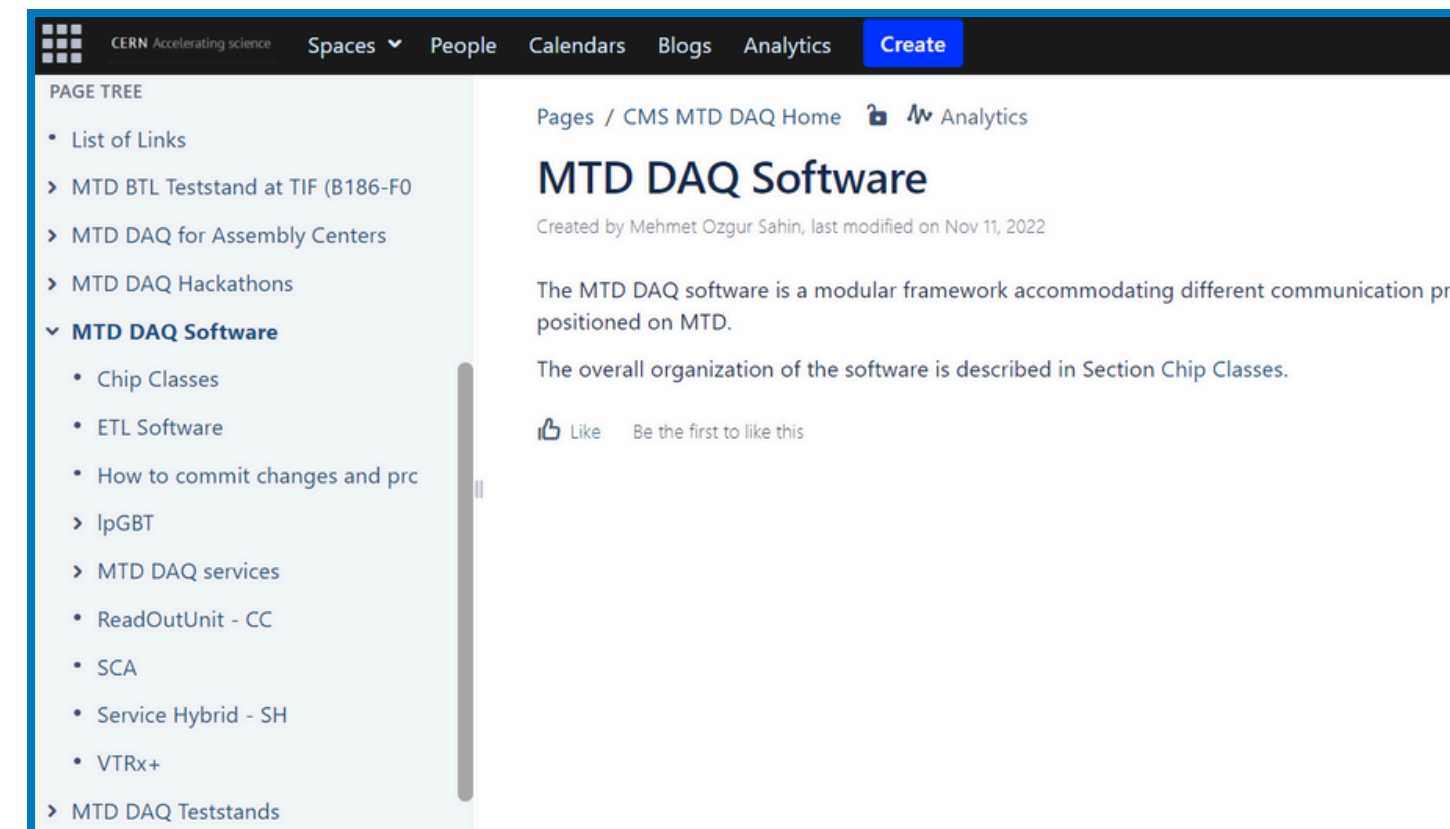


Fig.[7]: Advantages of code refactorization.



The screenshot shows a Confluence page for 'MTD DAQ Software'. The page title is 'MTD DAQ Software' and it was created by Mehmet Ozgur Sahin on Nov 11, 2022. The main content states: 'The MTD DAQ software is a modular framework accommodating different communication protocols positioned on MTD. The overall organization of the software is described in Section Chip Classes.' There is a 'Like' button and the text 'Be the first to like this'. A sidebar on the left shows a 'PAGE TREE' with a list of links, including 'MTD DAQ Software' which is expanded to show sub-items like 'Chip Classes', 'ETL Software', 'How to commit changes and prc', 'lpGBT', 'MTD DAQ services', 'ReadOutUnit - CC', 'SCA', 'Service Hybrid - SH', 'VTRx+', and 'MTD DAQ Teststands'.

Fig.[8]: Snippet of the MTD DAQ documentation in Confluence.



Thank You