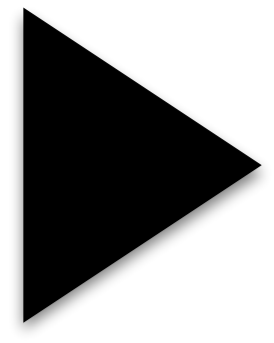


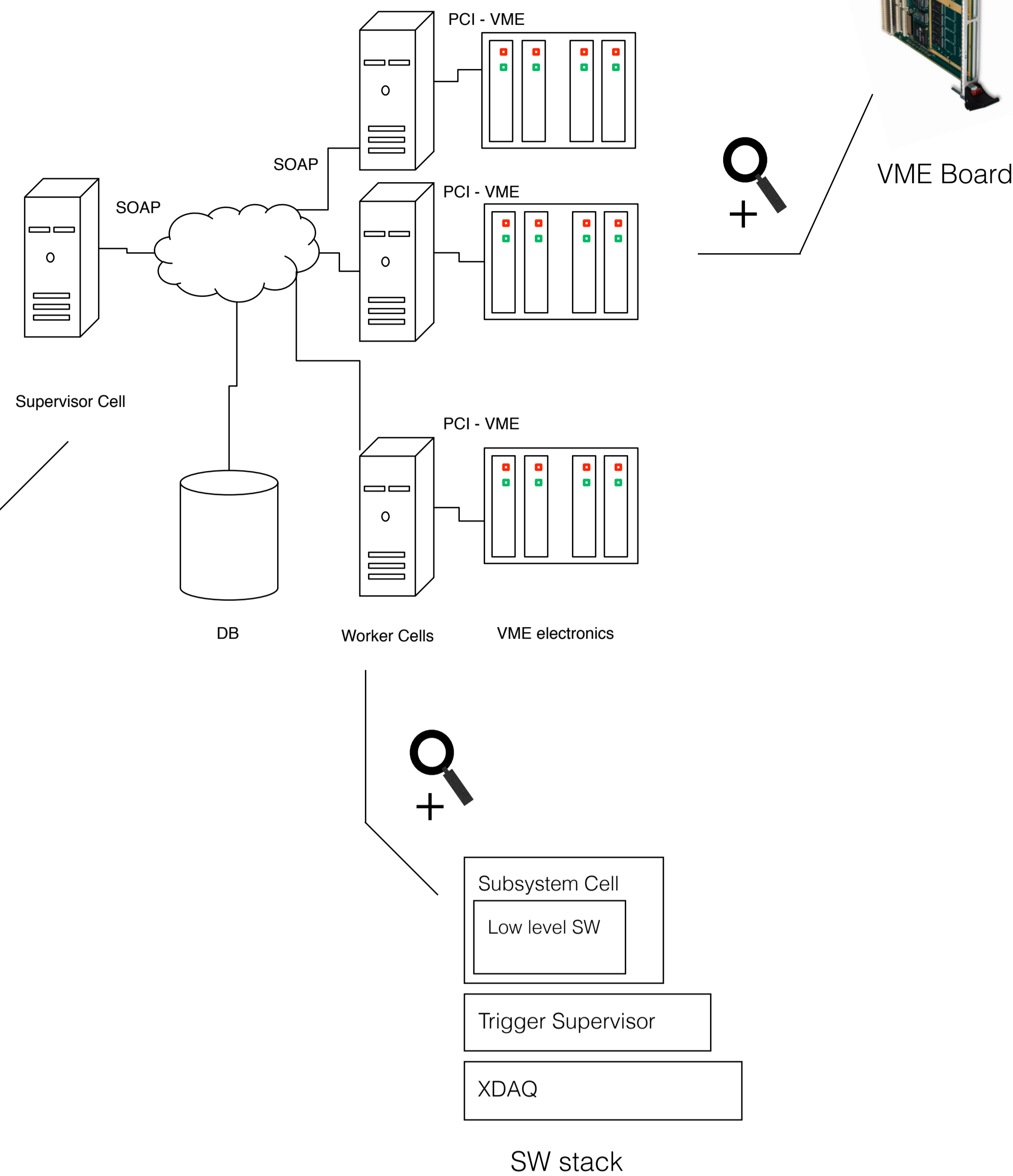
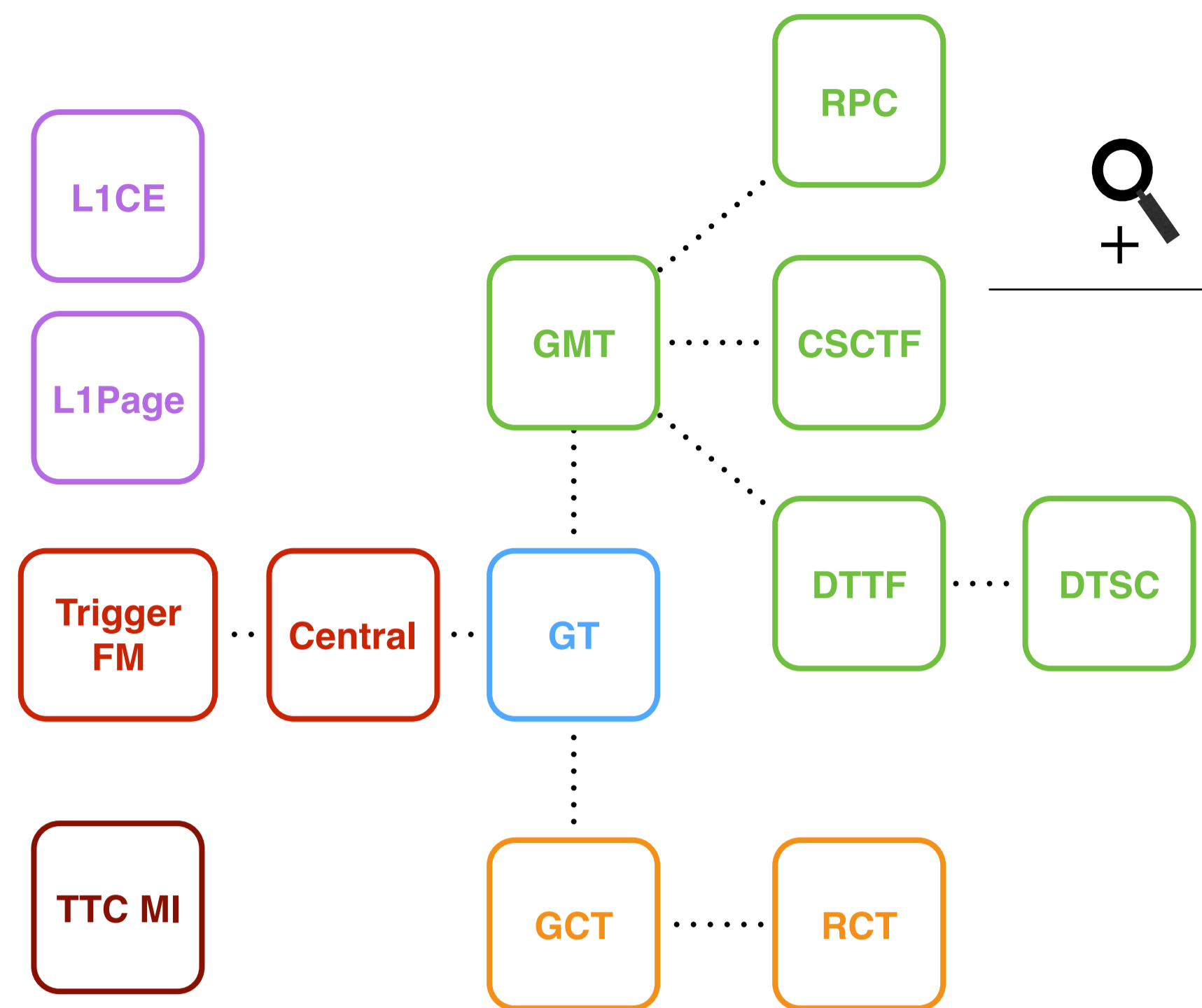
2010



R
U
N

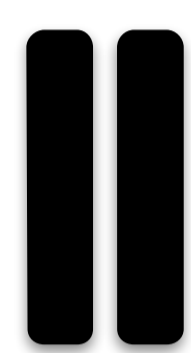
1

- **O(40) PCs**
- **O(4000) VME boards**
- **O(200) processes**
- **O(10³ KLOC) over 6 years**
- **O(1000) DB tables** (configuration and monitoring)



- **The HW model** influences the design of FW and SW stacks
- **Non uniformity on the HW model** translates into a non uniform model in FW, SW and DB models
- **FW, SW and DB models are specific** to each type of board and subsystem
- **Necessary to keep custom knowledge** for each subsystem, driven by the tailored solutions adopted
- **Huge manpower turnover** within the trigger groups makes difficult to keep knowledge and to maintain the legacy system
- **Some pieces of the current SW framework, Trigger Supervisor, are difficult to maintain** and extend for the parallel run and upgraded trigger
- **Lack of manpower** encourages the development of common solutions
- **SW lacks general purpose tools**, such as HW monitoring panels and testing infrastructure

2012



L
S

1

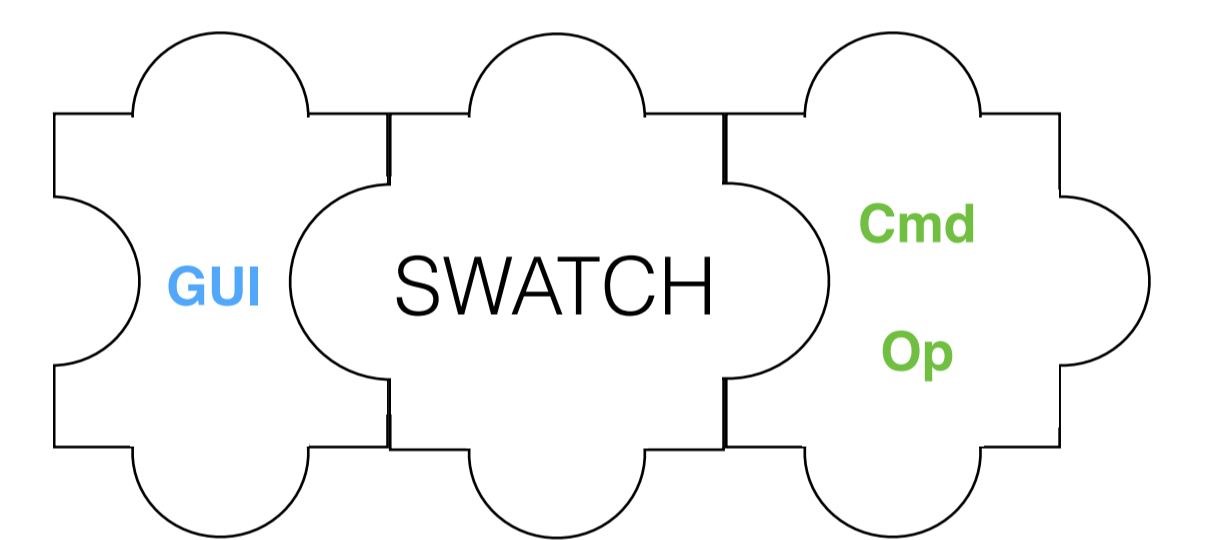
- The CMS VME electronics are being replaced to adapt the L1 trigger to the higher collision rates the LHC will deliver
- The upgraded trigger is set of general purpose processors of similar technology that follow the uTCA specification, increasing the system homogeneity
- The SW that controls, monitors and tests the HW needs to be re-written
- A great effort has been made to identify the common FW blocks and components shared across different cards
- A similar line of work has been followed to identify common functionalities in the control SW, as well as in the configuration DB
- As positive effects, the homogeneity on the SW and DB sides will increase, and the manpower needed to accommodate the online SW to the changes on HW will be reduced

SWATCH

Common control SW
for the
uTCA-based upgraded
CMS L1 Trigger

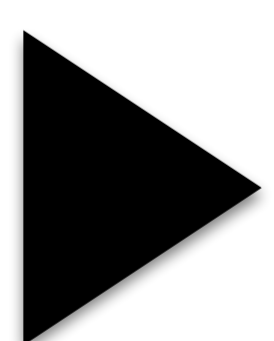
Carlos Ghabrous Larrea
Alessandro Thea
Jim Brooke
Ivan Amos Cali
Karol Bunkowski
Christos Lazaridis

- **SoftWare for Automating conTrol of Common Hardware**
- Written in C++, using STL and BOOST
- Run Control interface: SWATCH will be integrated within the current SW framework
- Common DB schema for all systems. Tables are more decoupled from the HW, allowing more independence between FW address tables and DB entities



- Provides with a highly normalised SW stack but allows for customisation for subsystems using Commands, Operations and GUIs plugins

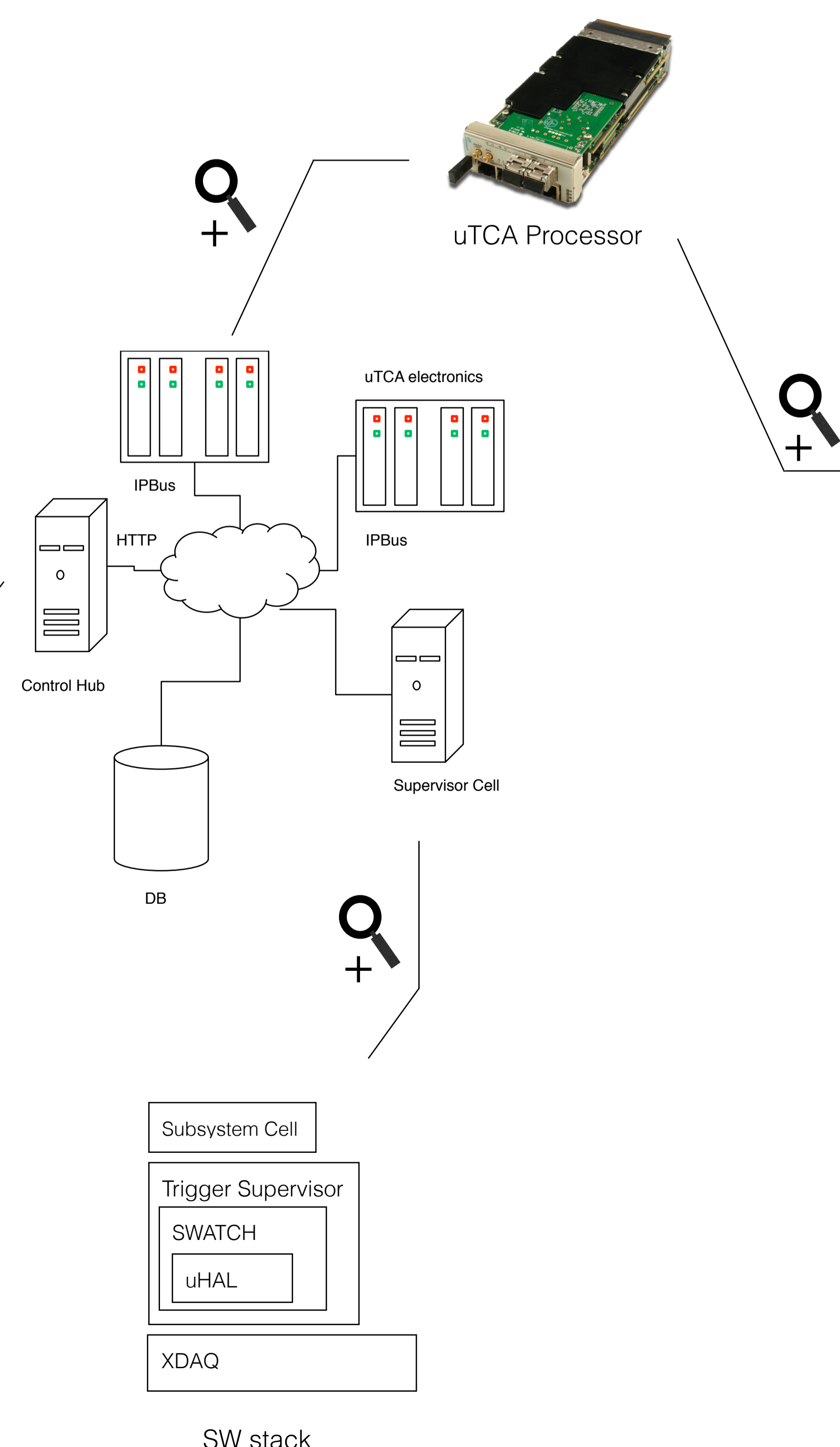
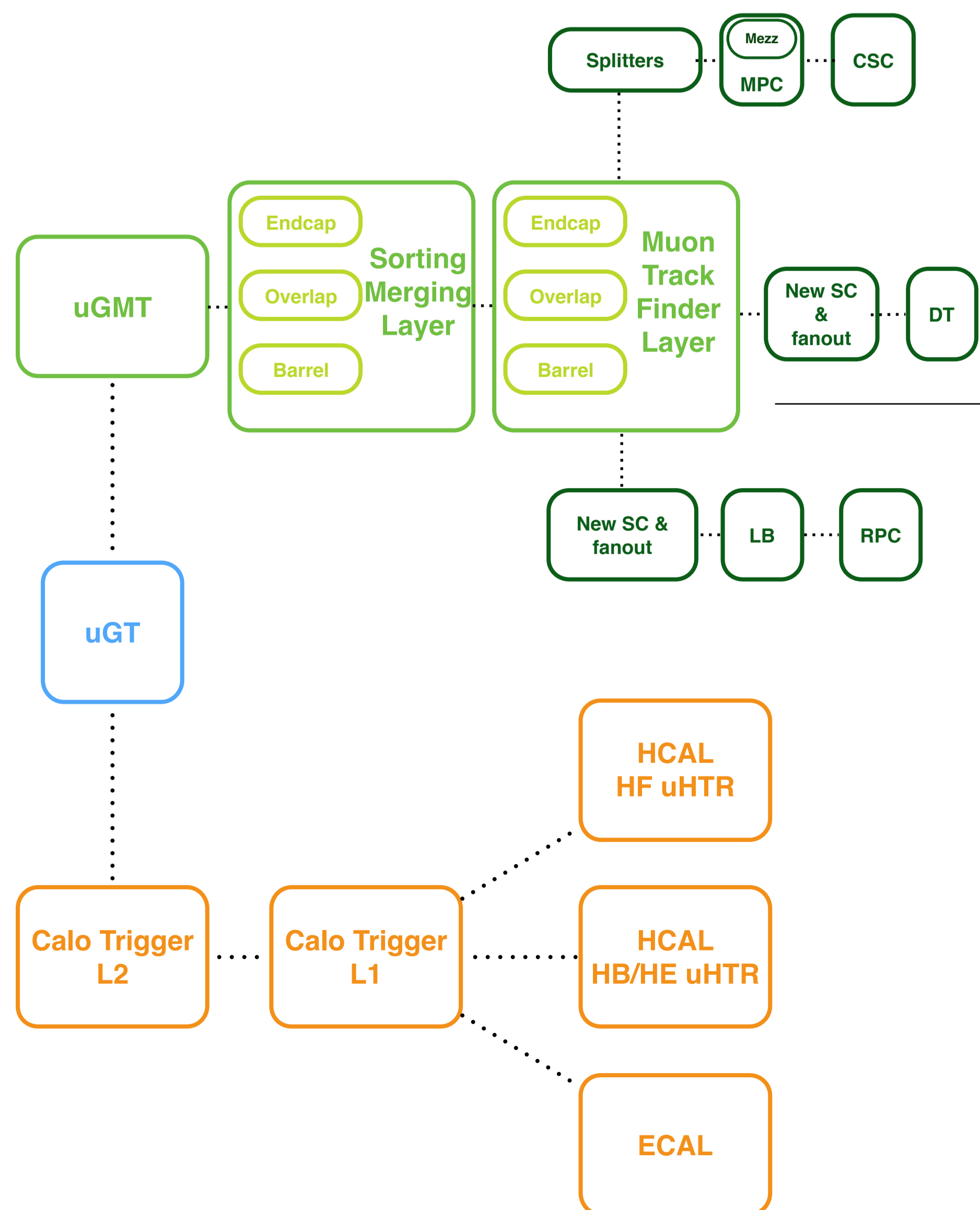
2015



R
U
N

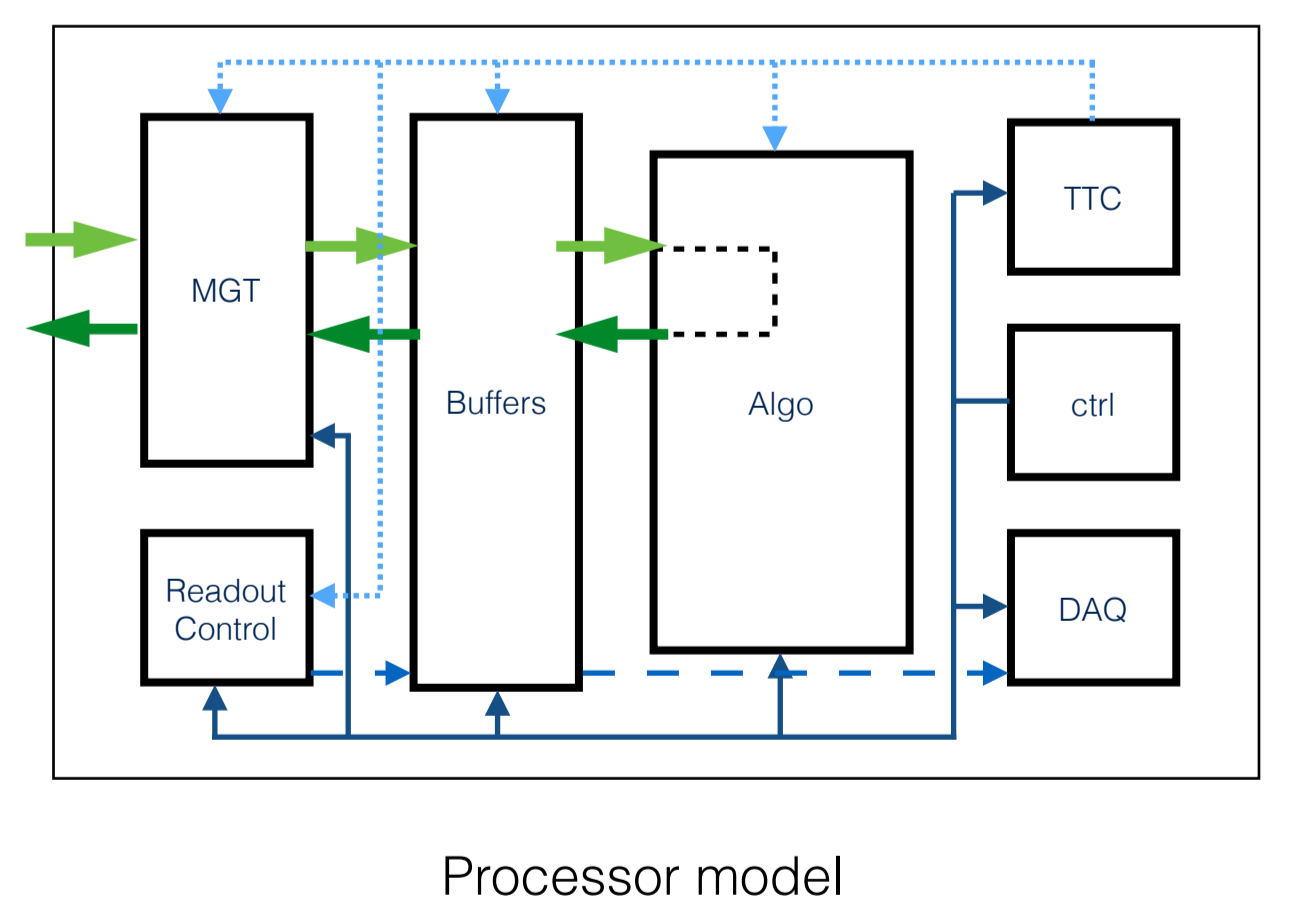
2

- **7 brand new subsystems**
- **O(100) electronic boards**
- **3 uTCA processors**
- **O(3000) optical links**
- **O(500) KLOC** to be rewritten

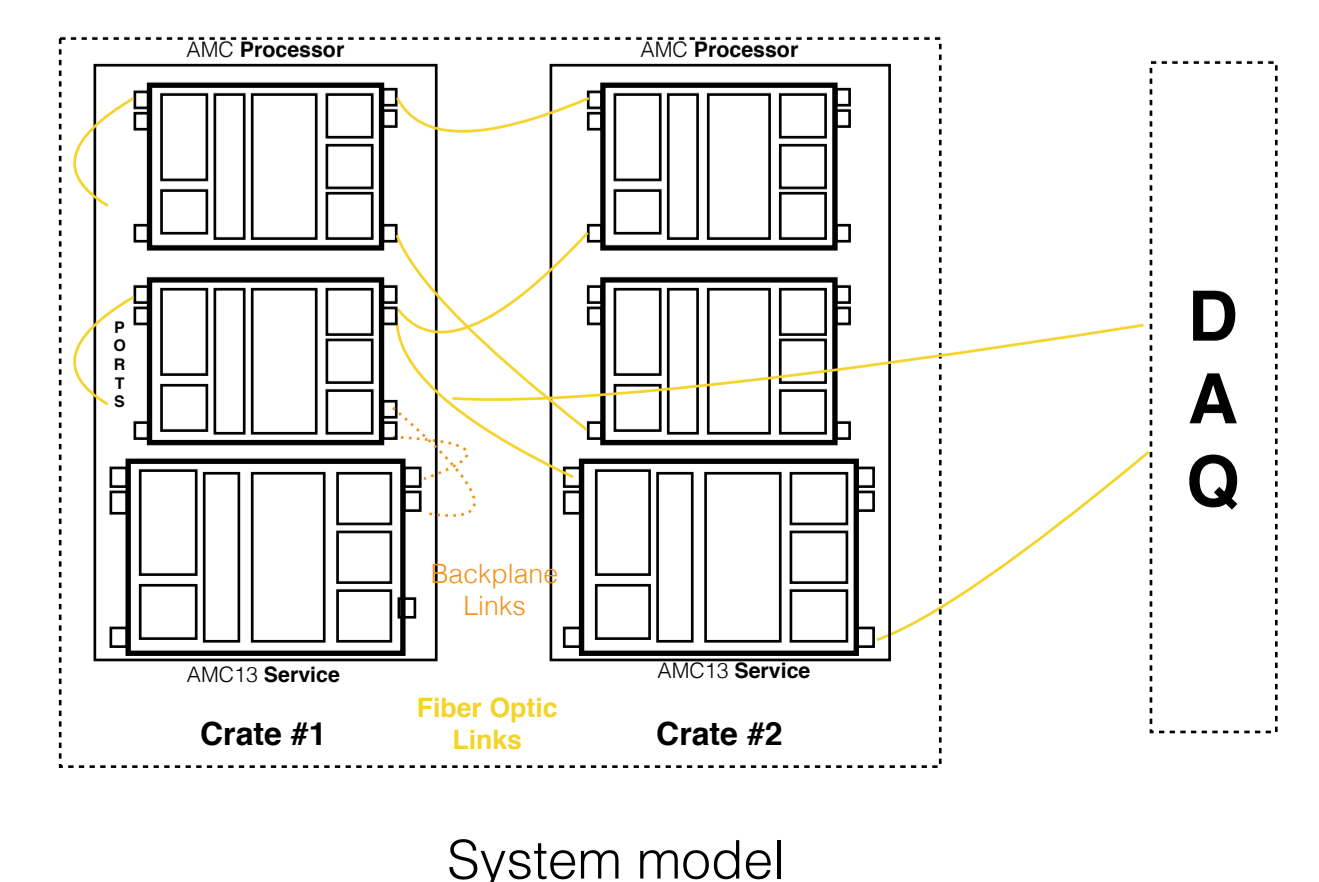


- Processor model: represented in the `switch::Processor` class, interface between `switch::system` and the HW specific APIs

- Common FW blocks have been standardised



- `switch::system` class represents an upgraded trigger system
- Physical objects such as processors, links and ports have their SW counterparts, making it easier tasks such as monitoring



2017