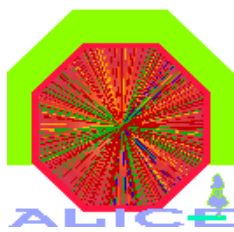


A Control Software for the ALICE High Level Trigger



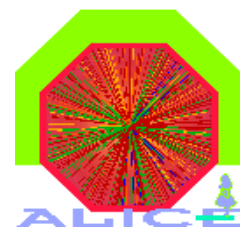
Outline



- **Overview ALICE / High Level Trigger (HLT)**
- **Overview Control System / Requirements**
- **Architecture**
 - **Configuration**
 - **Program State Handling**
 - **Program Interface**
 - **Master/Slave Operation**
- **Sample Configuration**
- **Summary & Conclusion**

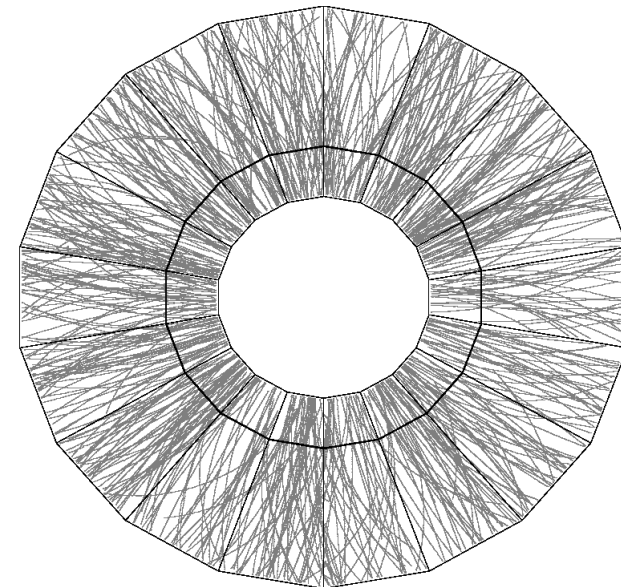
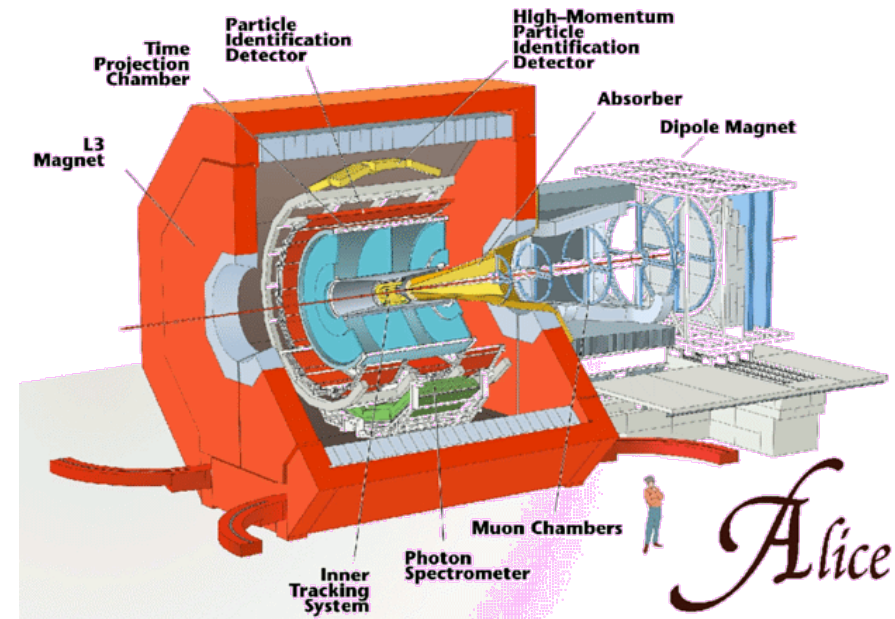


Overview - ALICE



ALICE High Level Trigger

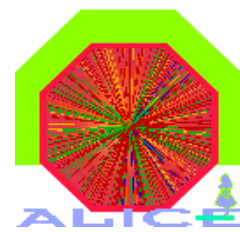
- ALICE: A Large Ion Collider Experiment
- Heavy-Ion Mode:
 - Up to 15000 particles in event
 - Max. event size >70 MB
 - Max. input rate from TPC: 200 Hz
 - Input data stream: ≤ 25 GB/s
 - Output data stream: ≤ 1.2 GB/s
- Proton-Proton Mode:
 - Max. input rate from TPC: 1kHz
 - Event size ≈ 3 MB



See also Talk by M. Richter and first talk by T. M. Steinbeck

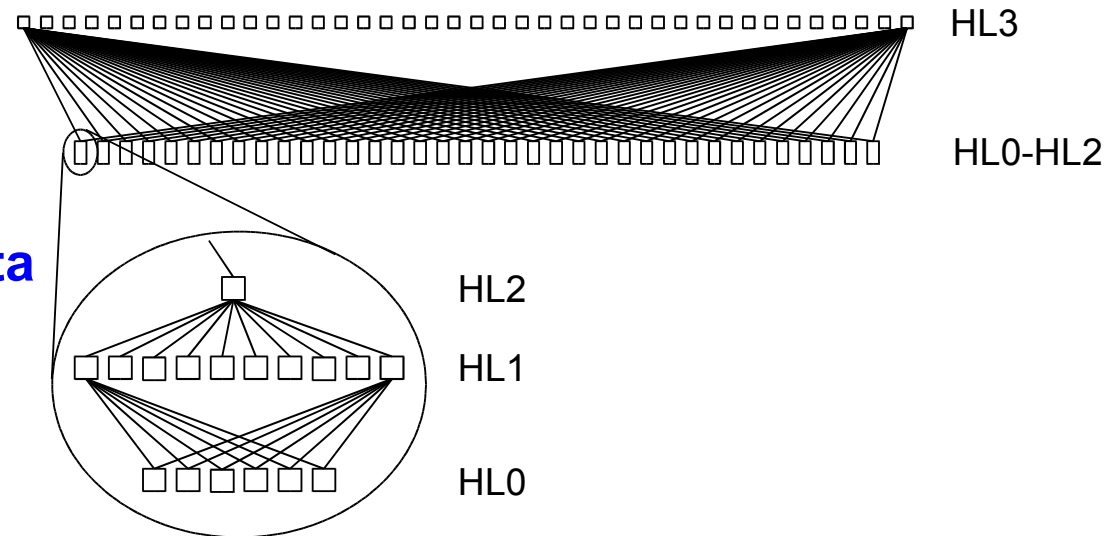
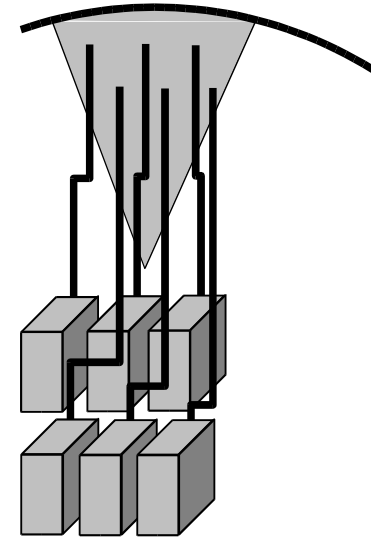


Overview - HLT



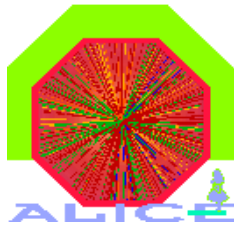
ALICE High Level Trigger

- Processing in several steps from raw detector data to full event reconstruction
- Large PC cluster
 - Initially ≈ 400 -500 nodes
 - Arranged in hierarchy levels (HL) that match detector layout and analysis steps
- Exact processing sequence and hierarchy not known
- Framework consisting of multiple independent, communicating components used to transport data
- Software needed to manage components and their interaction





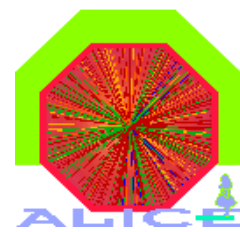
Outline



- **Overview ALICE / High Level Trigger (HLT)**
- **Overview Control System / Requirements**
- **Architecture**
 - **Configuration**
 - **Program State Handling**
 - **Program Interface**
 - **Master/Slave Operation**
- **Sample Configuration**
- **Summary & Conclusion**

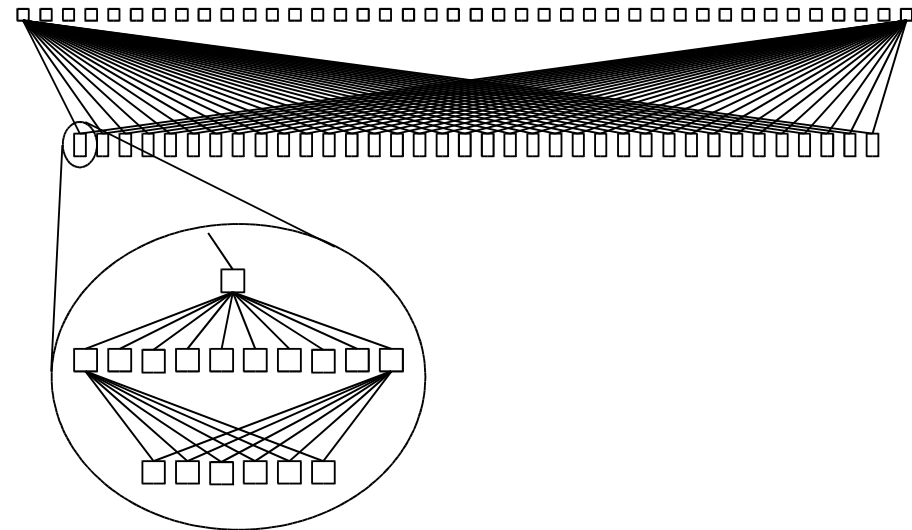


HLT Control System



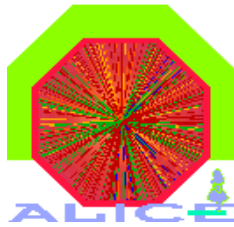
Purpose of the HLT Control System: Management of HLT Processes

- **Supervise processes**
- **React to state changes**
 - **E.g. errors, unexpected termination**
- **Manage system startup**
 - **Connect components only when they are ready**
 - **Connect bridges between nodes when nodes are ready**
- **Orchestration of HLT system**





Requirements



- **Flexible**
 - HLT Configurations not yet defined
 - Usable for other programs as well

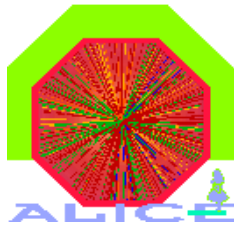
- **Hierarchical**
 - >2000 processes not manageable by single supervisor instance
 - Eases configurations

- **No single point of failure**
 - PC cluster nodes are unreliable
 - No need for highly reliable, expensive, equipment

- **Should be able to run on the cluster nodes themselves, along the analysis processes.**



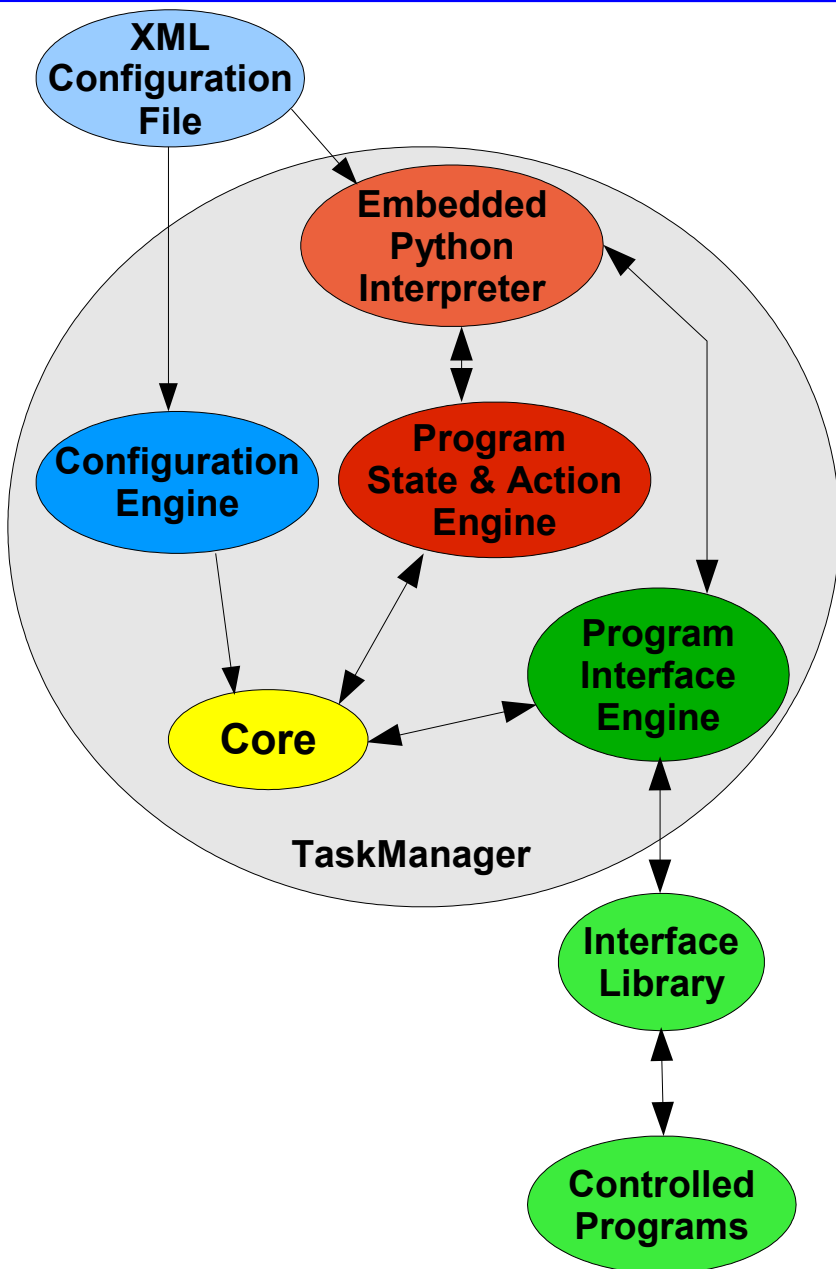
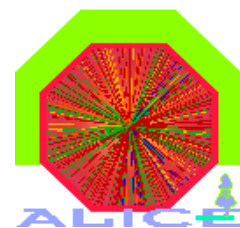
Outline



- **Overview ALICE / High Level Trigger (HLT)**
- **Overview Control System / Requirements**

- **Architecture**
 - **Configuration**
 - **Program State Handling**
 - **Program Interface**
 - **Master/Slave Operation**
- **Sample Configuration**
- **Summary & Conclusion**

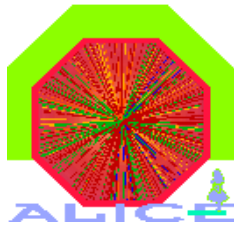
TaskManager Architecture



- TaskManager configuration is stored in XML files
- "State machine" logic is Python code, read from configuration file and executed by embedded Python interpreter
- Communication with external programs is done via external interface library, supplied with programs, specified in configuration file



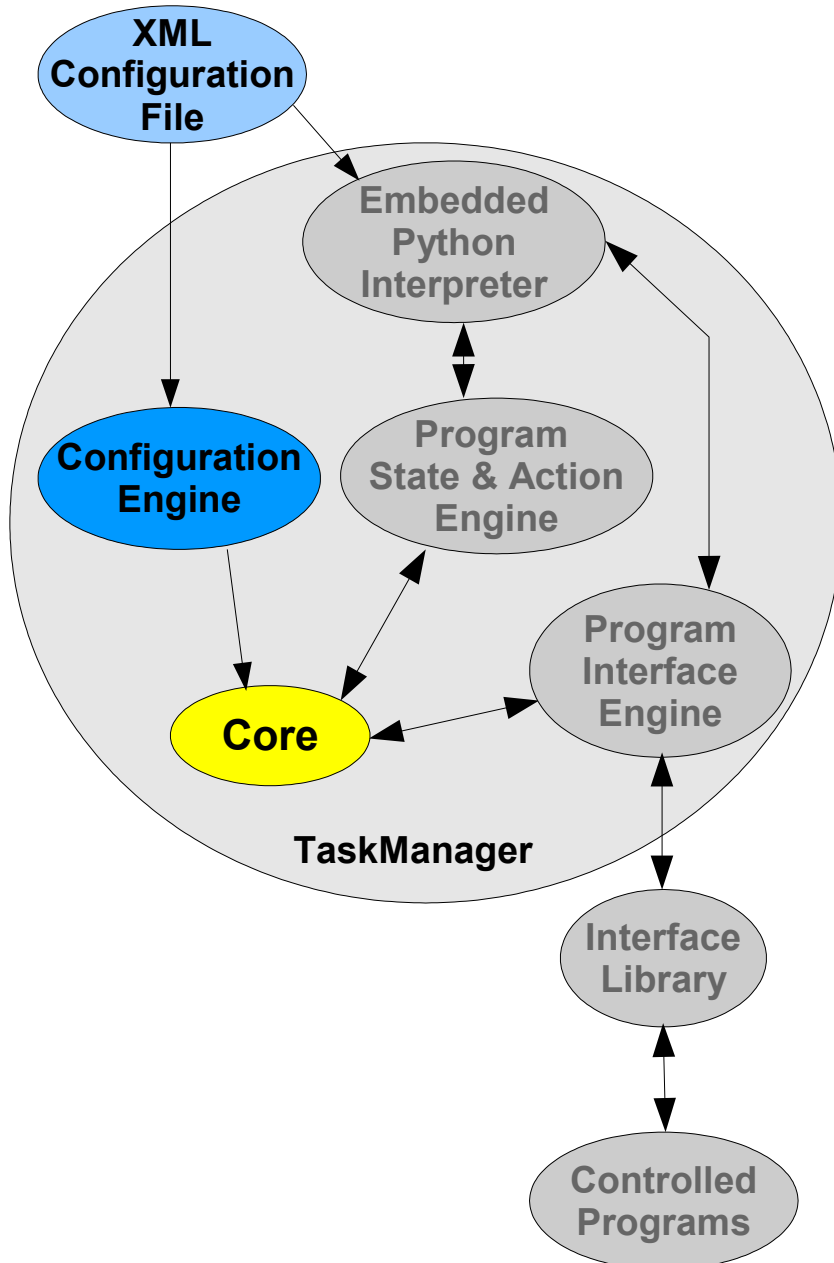
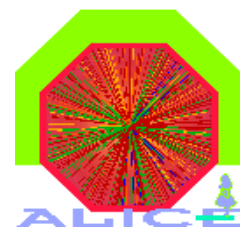
Outline



- **Overview ALICE / High Level Trigger (HLT)**
- **Overview Control System / Requirements**
- **Architecture**
 - **Configuration**
 - **Program State Handling**
 - **Program Interface**
 - **Master/Slave Operation**
- **Sample Configuration**
- **Summary & Conclusion**



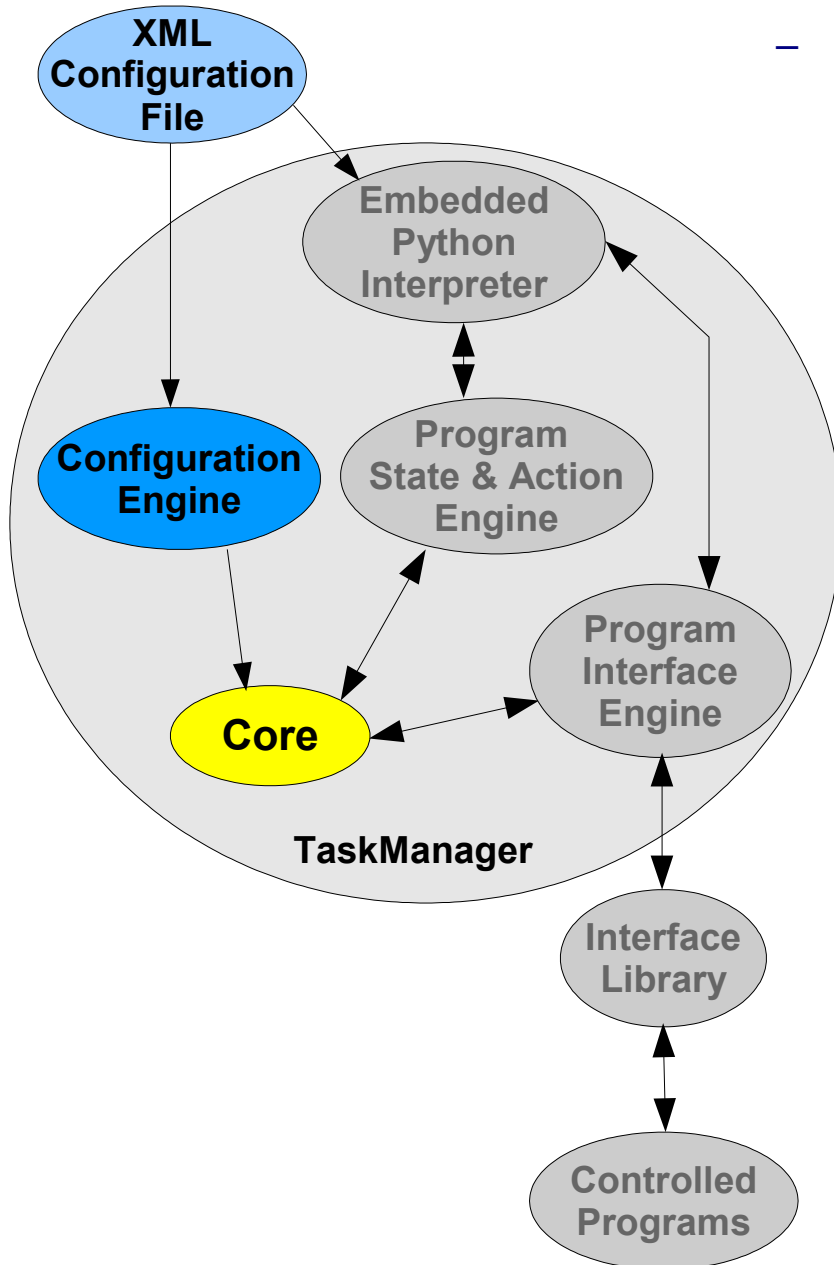
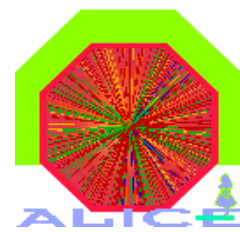
Configuration Files



- Configuration is stored in XML files
- Configuration Engine reads in configuration
- TaskManager Core queries configuration items from Configuration Engine
- Configuration files store
 - Programs to be started and controlled
 - Python code for "state machines" of controlled programs and global ones
 - Interface libraries to be used for communication with programs
 - Master/Slave TaskManager configuration items



Configuration Files

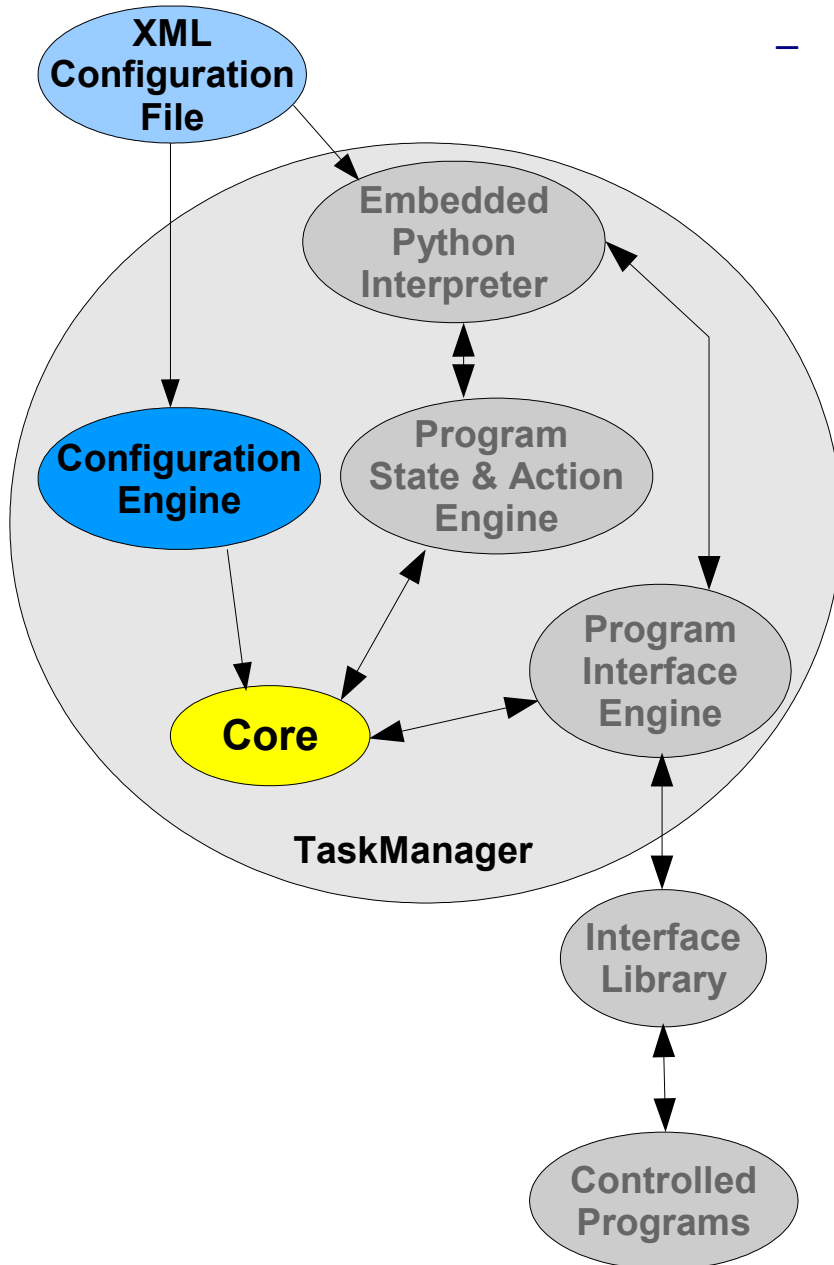
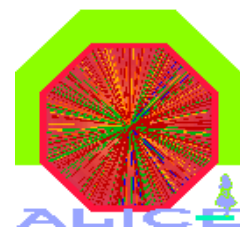


– For each controlled program the configuration can describe:

- The command line to be executed
- The address for communication with the program (passed to the interface library)
- Resources used by the program (for cleanup upon termination)
- Different situations in which to execute Python code include
 - State changes
 - Termination
 - Configuration change



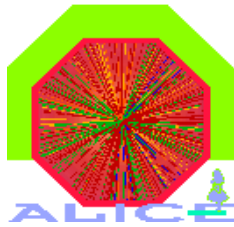
Configuration Files



- **Global items in a configuration file include:**
 - **The interface library to be used for communication with controlled programs**
 - **Global state change action Python code**
 - **Slave control specification (only slave TM)**
 - **Master TM configuration is handled via normal controlled program specification**



Outline



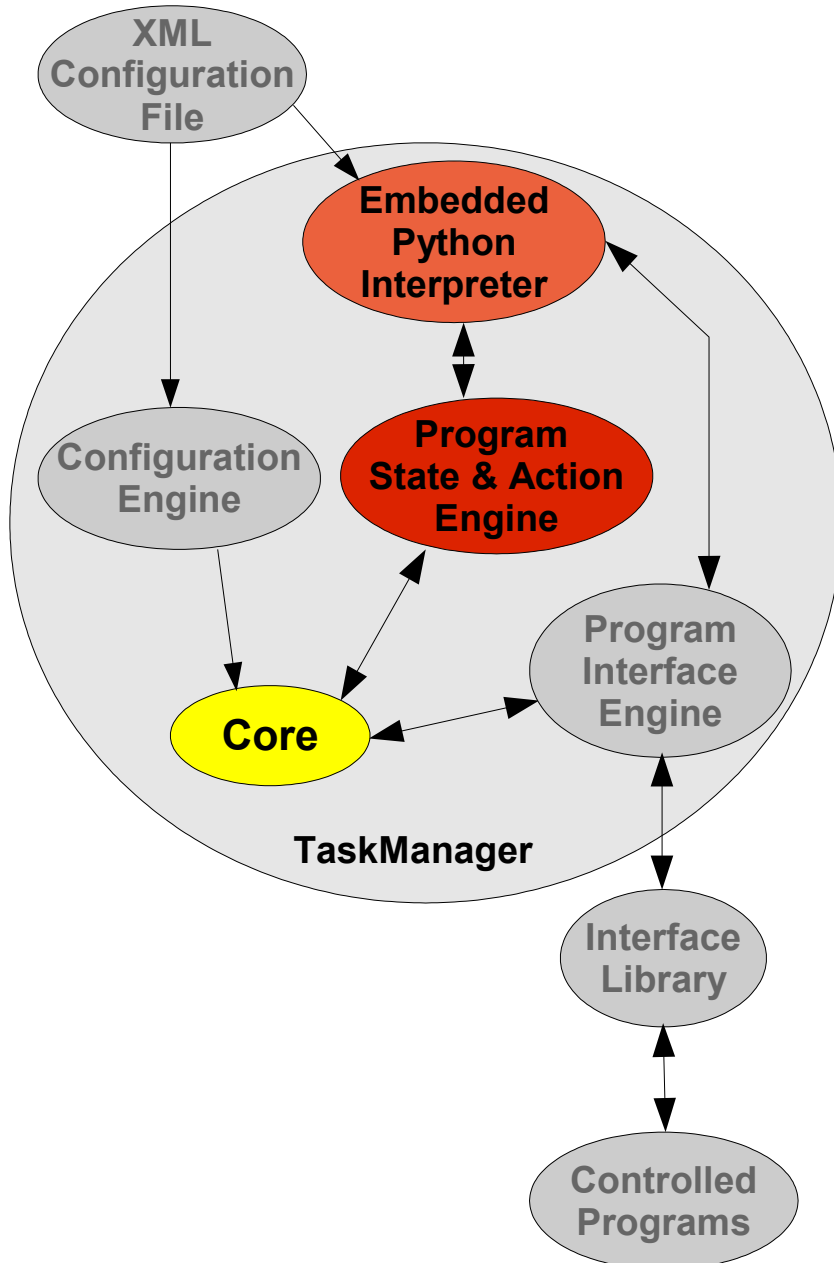
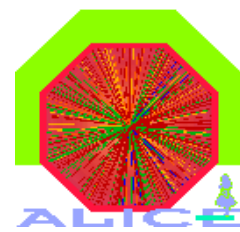
- **Overview ALICE / High Level Trigger (HLT)**
- **Overview Control System / Requirements**
- **Architecture**
 - **Configuration**

 - **Program State Handling**

 - **Program Interface**
 - **Master/Slave Operation**
- **Sample Configuration**
- **Summary & Conclusion**



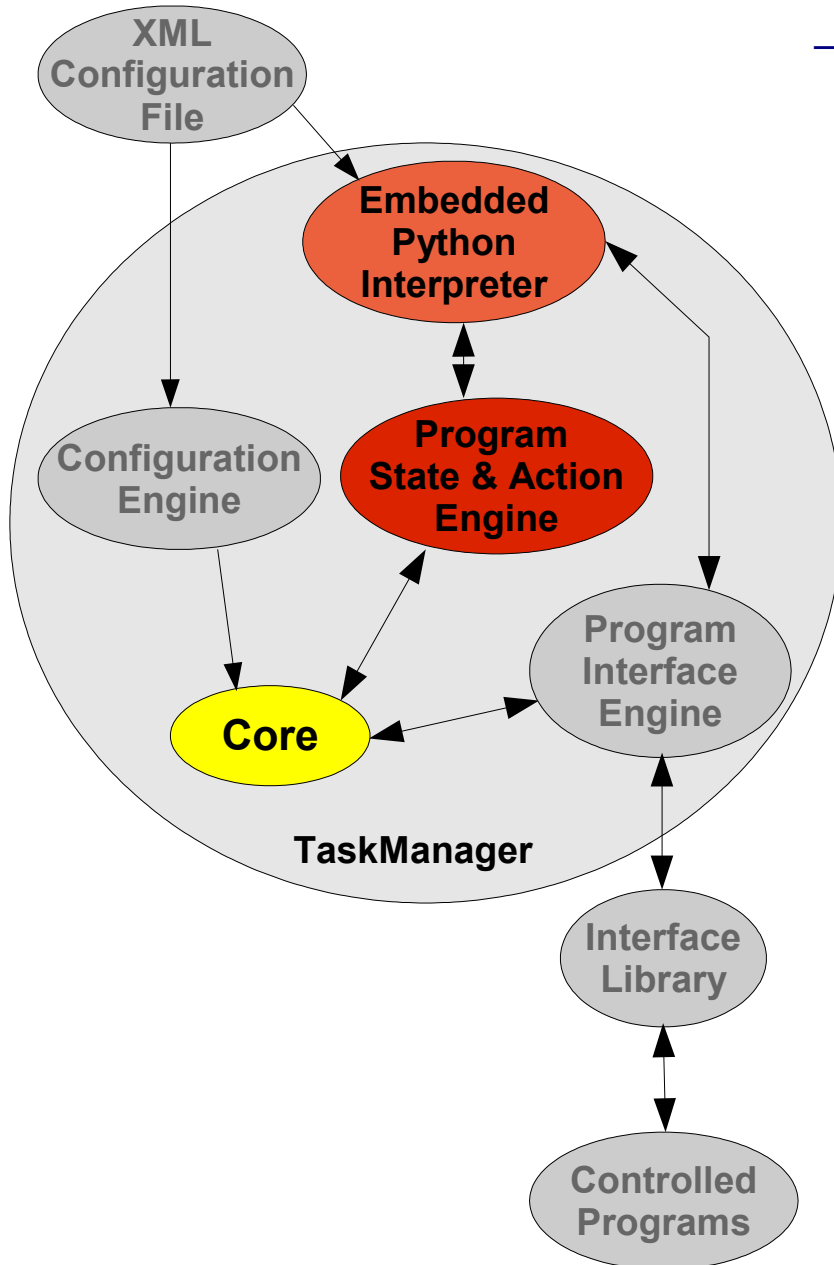
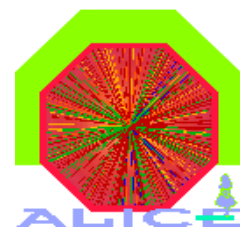
Program State Handling



- „State Machine“ logic contained in *Actions*, Python code in configuration file
- TaskManager has embedded Python interpreter
- When conditions are met interpreter executes specified code
- More powerful and flexible than pure state machine
- Python code is specified for different occasions, e.g.:
 - Program state change
 - Program termination
 - Configuration changes
- Set of special functions provide interface from Python code to TaskManager and controlled programs



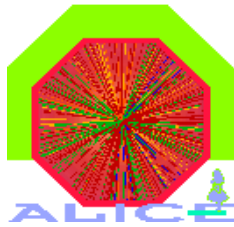
Program State Handling



- Python TaskManager Interface contains functions for multiple purposes
 - Query state and status data of controlled programs
 - Send commands to controlled programs
 - Start programs
 - Terminate (forcibly) programs
 - Exchange information between master & slave TaskManagers



Outline



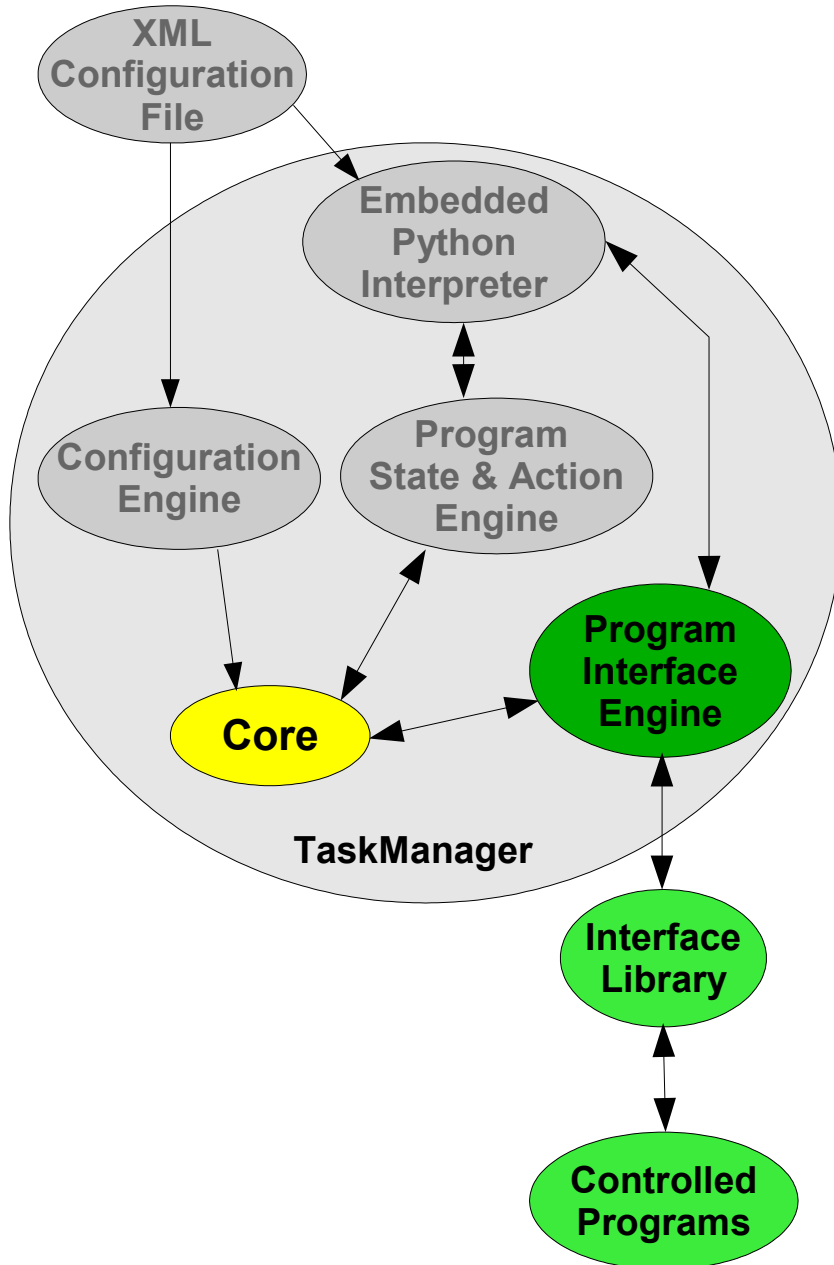
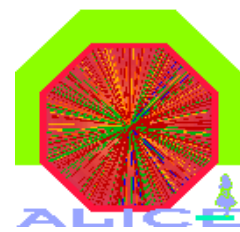
- **Overview ALICE / High Level Trigger (HLT)**
- **Overview Control System / Requirements**
- **Architecture**
 - **Configuration**
 - **Program State Handling**

 - **Program Interface**

 - **Master/Slave Operation**
- **Sample Configuration**
- **Summary & Conclusion**



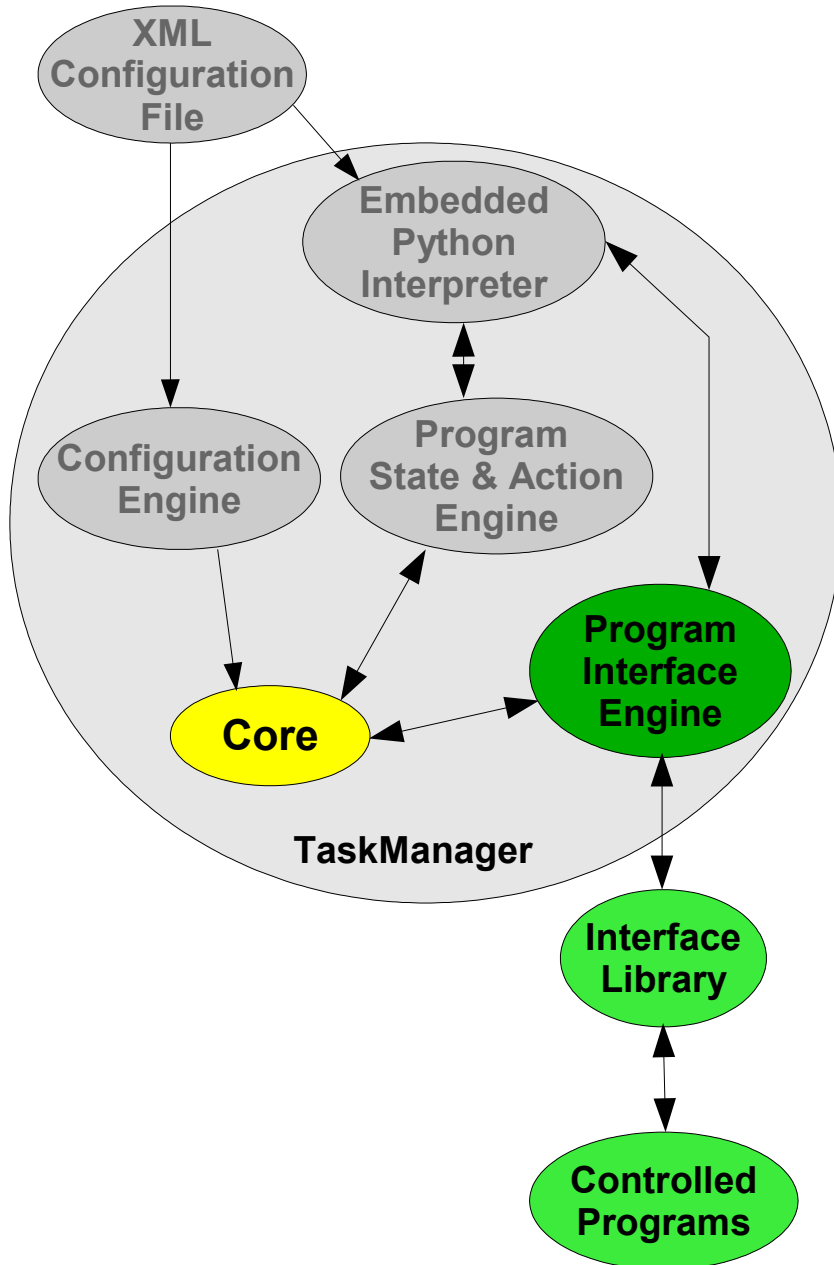
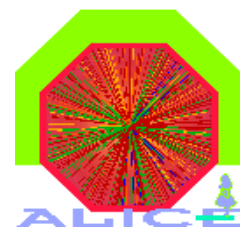
Program Interface



- Programs use different methods to present status to the outside (including the TaskManager)
- ⇒ Communication between TaskManager and programs is handled by interface libraries
- Interface libraries
 - are external to the TaskManager
 - belong to a set of programs
 - are loaded at runtime
 - know how to communicate with their programs
 - implement specific, defined set of functions
- TaskManager can call specified functions
 - Consistent interface API for all programs



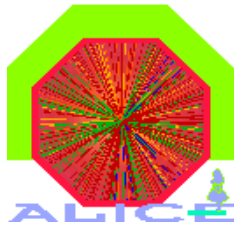
Program Interface



- Program interface includes functions to
 - initialize the interface library
 - deinitialize the interface library
 - query a program's state
 - query additional status data for a program
 - send a command to a program
 - wait for interrupts (or look-at-me-s) from programs



Outline

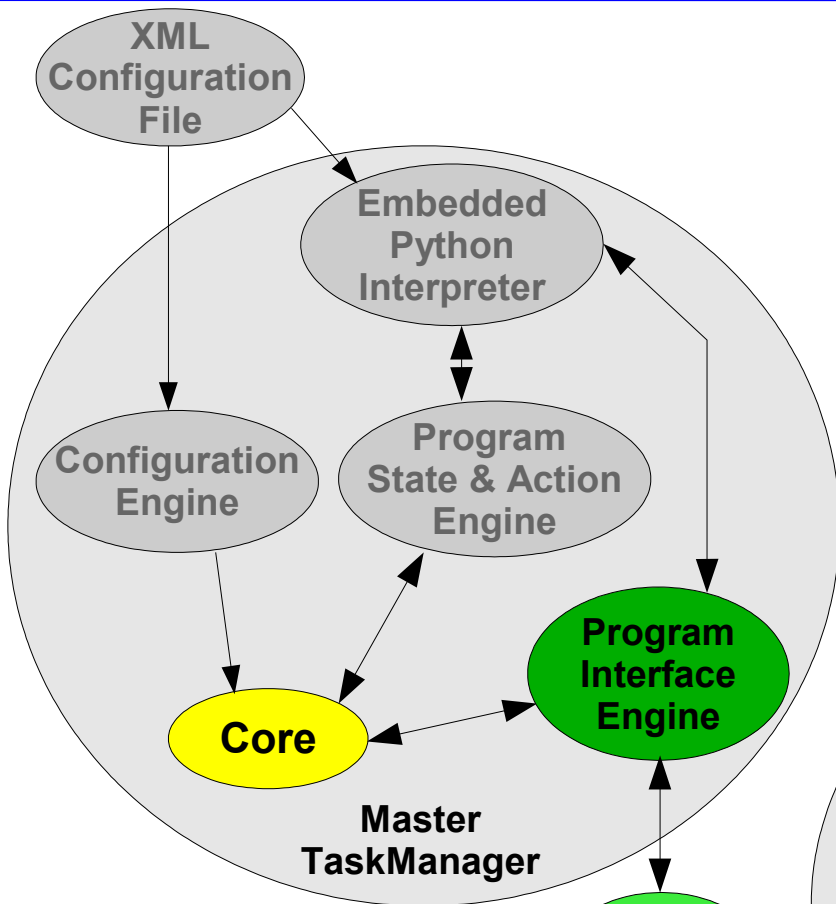
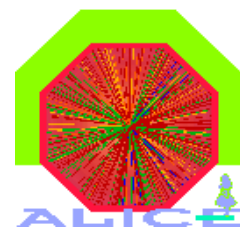


- **Overview ALICE / High Level Trigger (HLT)**
- **Overview Control System / Requirements**
- **Architecture**
 - **Configuration**
 - **Program State Handling**
 - **Program Interface**

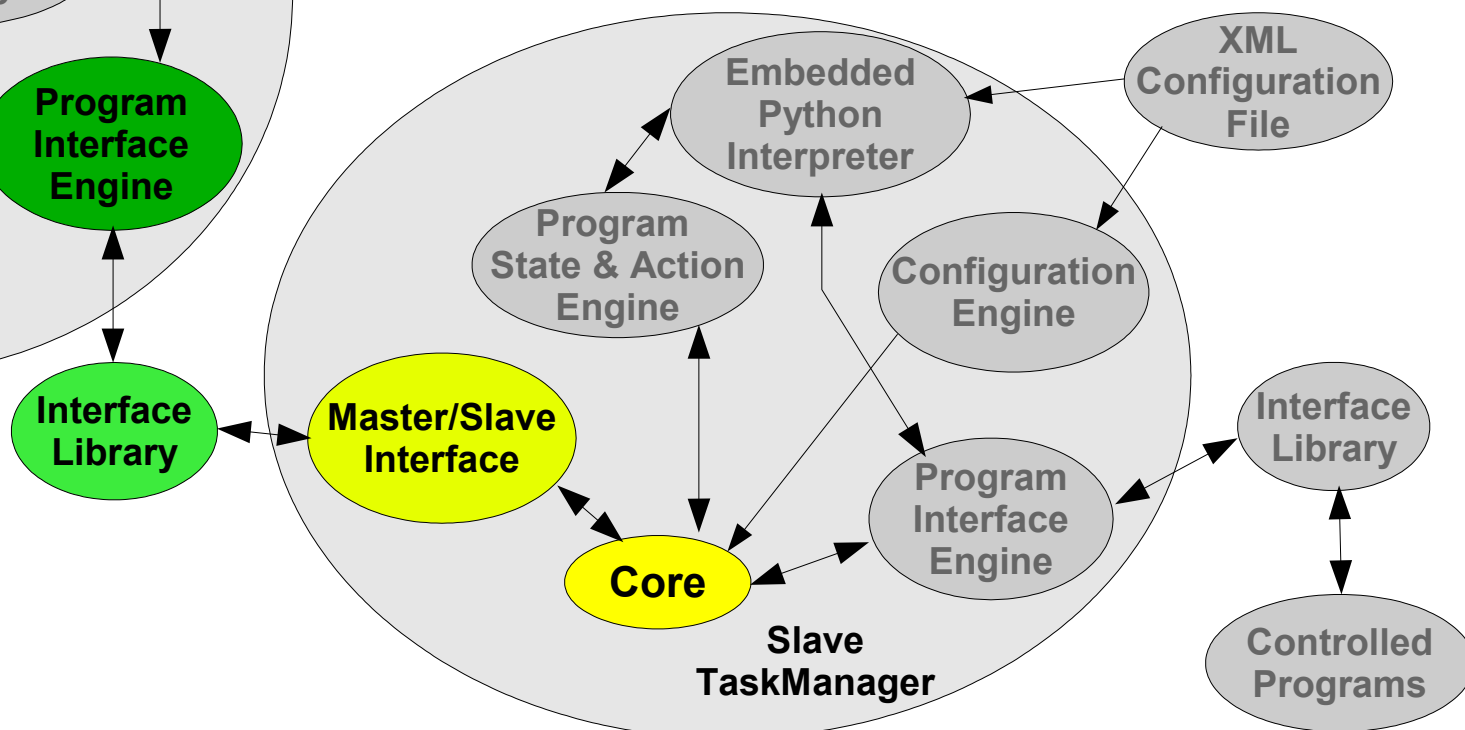
 - **Master/Slave Operation**

- **Sample Configuration**
- **Summary & Conclusion**

Master Slave Configuration

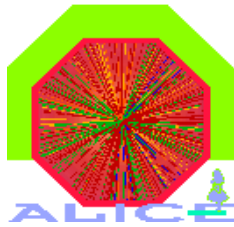


- Slave TaskManager uses additional internal object to receive commands from master and provide master with status data
- Master uses interface library to communicate with slaves
- Master/Slave communication is simple extension to standard communication w. controlled programs





Outline



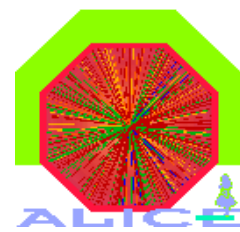
- **Overview ALICE / High Level Trigger (HLT)**
- **Overview Control System / Requirements**
- **Architecture**
 - **Configuration**
 - **Program State Handling**
 - **Program Interface**
 - **Master/Slave Operation**

- **Sample Configuration**

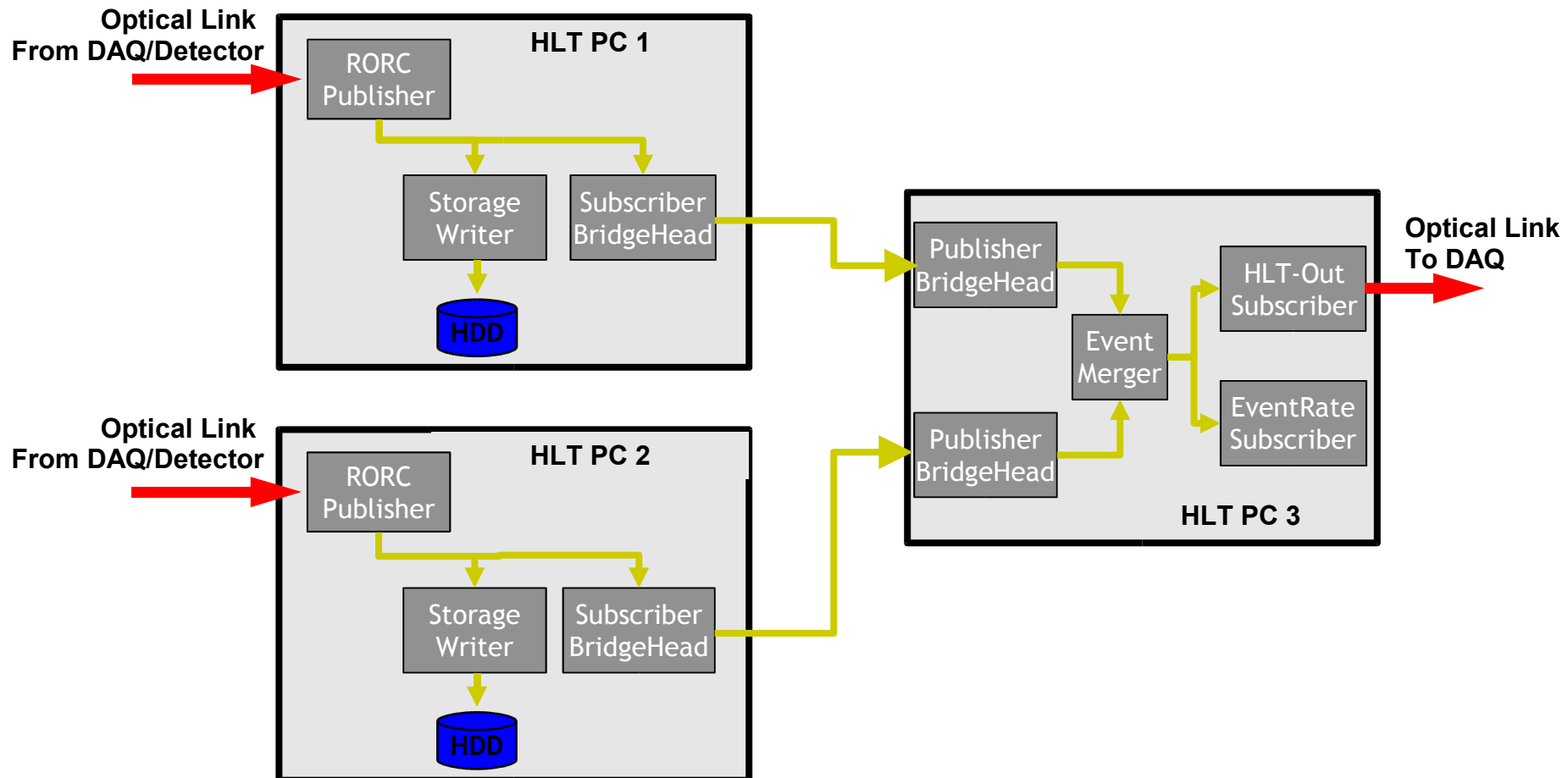
- **Summary & Conclusion**



Sample Configuration

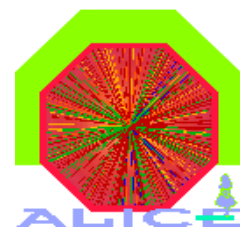


- Configuration planned for HLT use in ALICE TPC Testbeam
- 2 optical readout links from detector via DAQ splitter module
- 1 optical output link going to DAQ
- 3 PCs, one per optical link, several software components on each PC



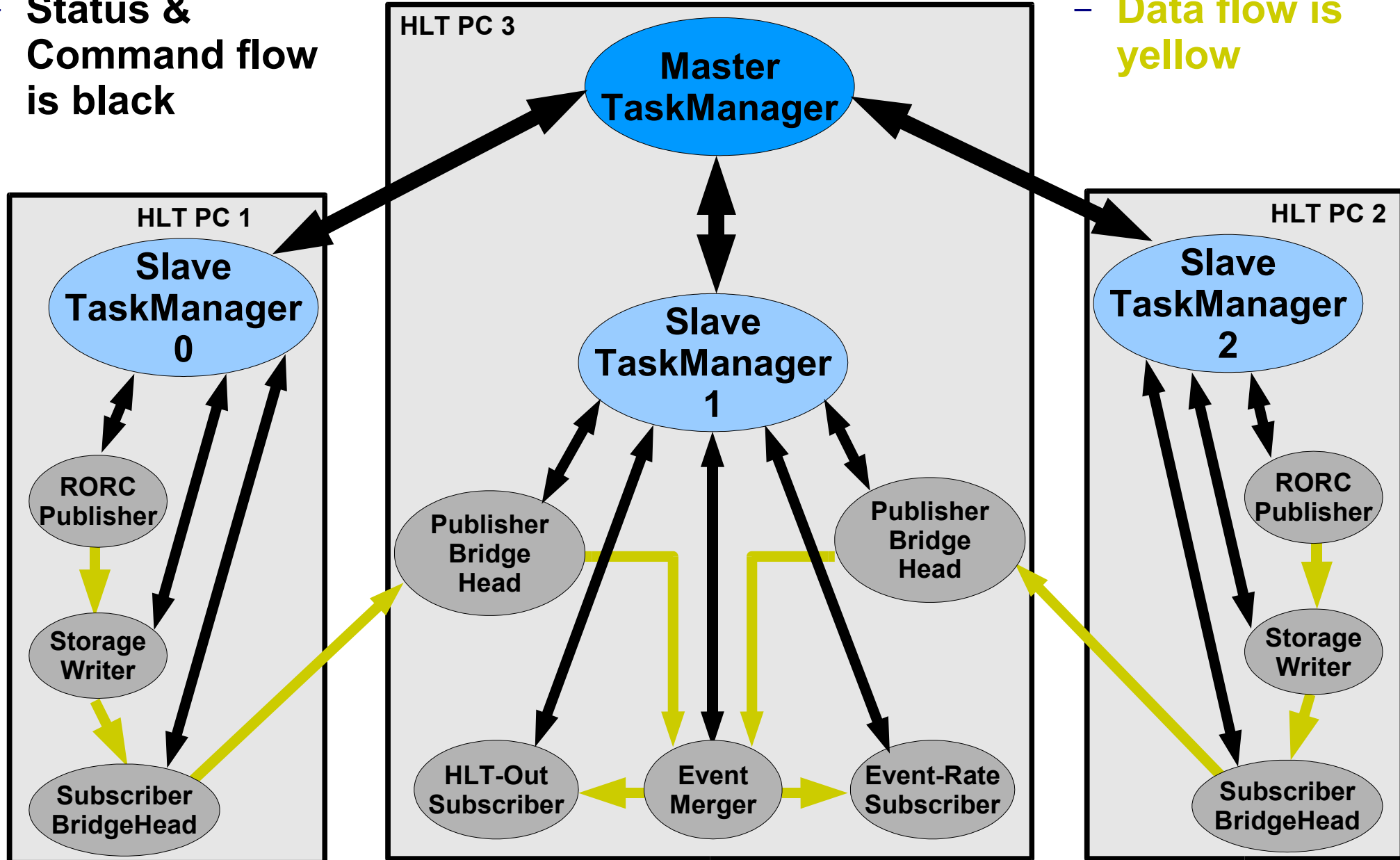


Sample Configuration



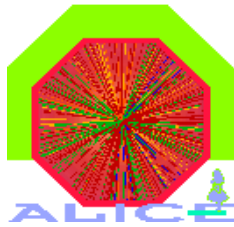
– Status & Command flow is black

– Data flow is yellow





Outline

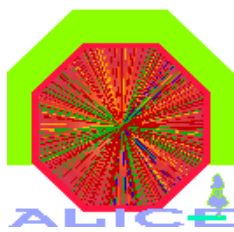


- **Overview ALICE / High Level Trigger (HLT)**
- **Overview Control System / Requirements**
- **Architecture**
 - **Configuration**
 - **Program State Handling**
 - **Program Interface**
 - **Master/Slave Operation**
- **Sample Configuration**

- **Summary & Conclusion**



Summary & Conclusion



- **TaskManager provides flexible and hierarchical program control**
- **Powerful program state handling through embedded Python**
- **Adaptable to many programs through interface libraries**
- **Machine- and human-readable/writeable XML configuration files**
- **Avoids single-points-of-failure**

- **Used for ALICE High Level Trigger Data Challenges**
- **Used for HLT control in ALICE TPC testbeam**

- **Designed to be adaptable to other problems, in- and outside of Alice HLT**