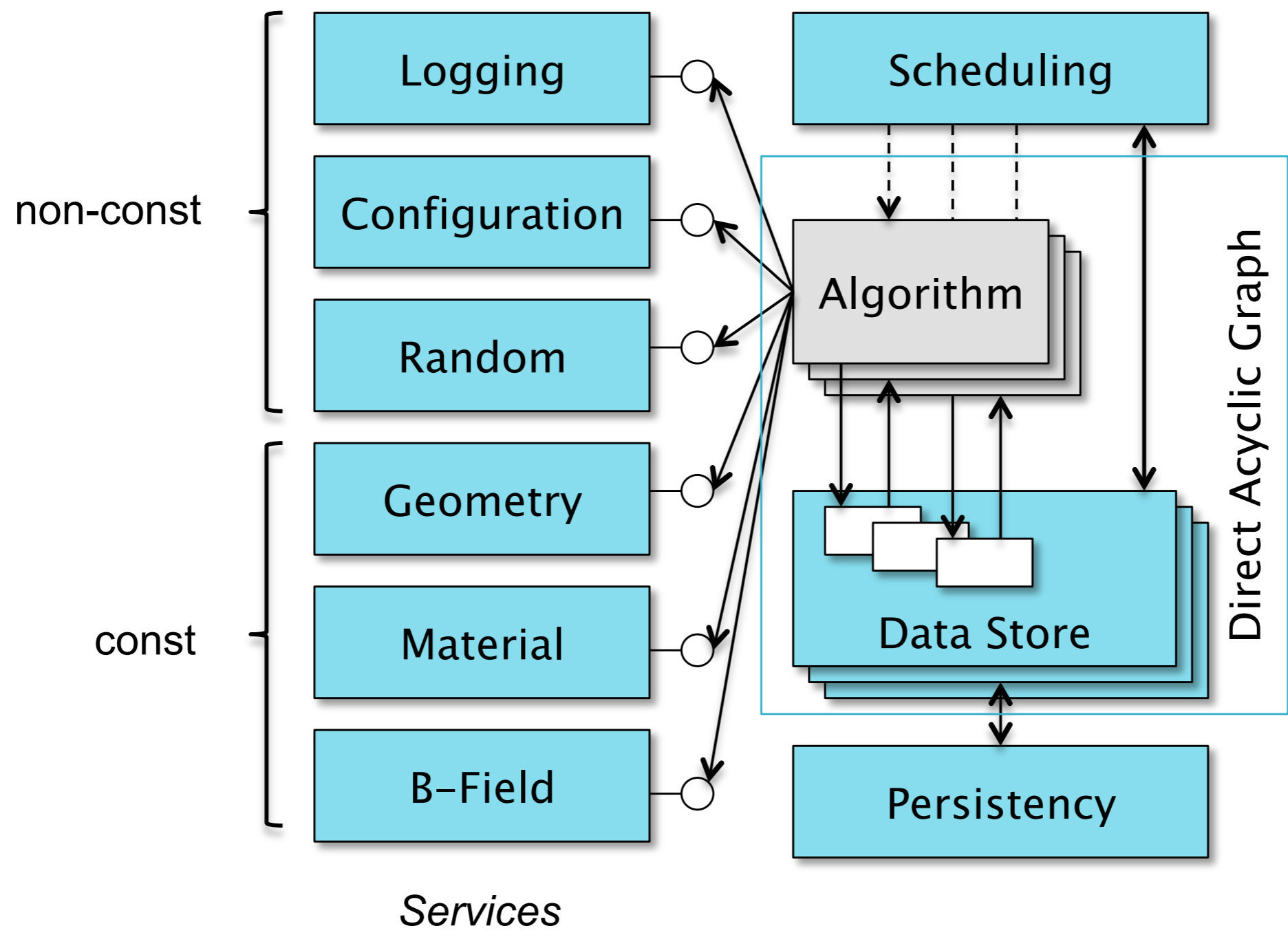




Concurrent Whiteboard Demonstrator Plans

Pere Mato, Benedikt Hegner
CERN

$\mu = 500 \text{ GeV}/c$
 $H, A \rightarrow \tau\tau \rightarrow \text{two } \tau \text{ jets} + X, 60 \text{ fb}^{-1}$



- **Defining a single memory model essential for a parallel application**
 - Knowing the dependencies of algorithms (i.e. data transformations) determines when things can be scheduled without conflicts
 - Coding for state integrity
 - move all states out of algorithms
 - provide context (*) at each call
 - make data transformations atomic
- This can be realized by a **concurrent whiteboard**

(*) Context can e.g. be the event to be processed including all conditions etc

Design considerations

- **Requirements to concurrent whiteboard**
 - Multi-event data store
 - Data items are identified by type, an id and the *context*
 - Allowed transactions could be get, put and update
- **Requirements to algorithms**
 - Re-entrant w.r.t. to different contexts
 - To be carried out transactions and their dependencies declared upfront
- **Requirements to scheduler**
 - The concurrent whiteboard contents + declared data dependencies + knowledge about reentrancy should be sufficient to know when an algorithm can be run
 - Details of actual execution could be left to e.g. libdispatch
- **Demonstrator plan until April**
 - Lay out a full API covering the proposed design concepts
 - Implement this API and combine it with prototypes of Markus Frank
 - Use common tests and metrics from “Evaluation of Frameworks” demonstrator