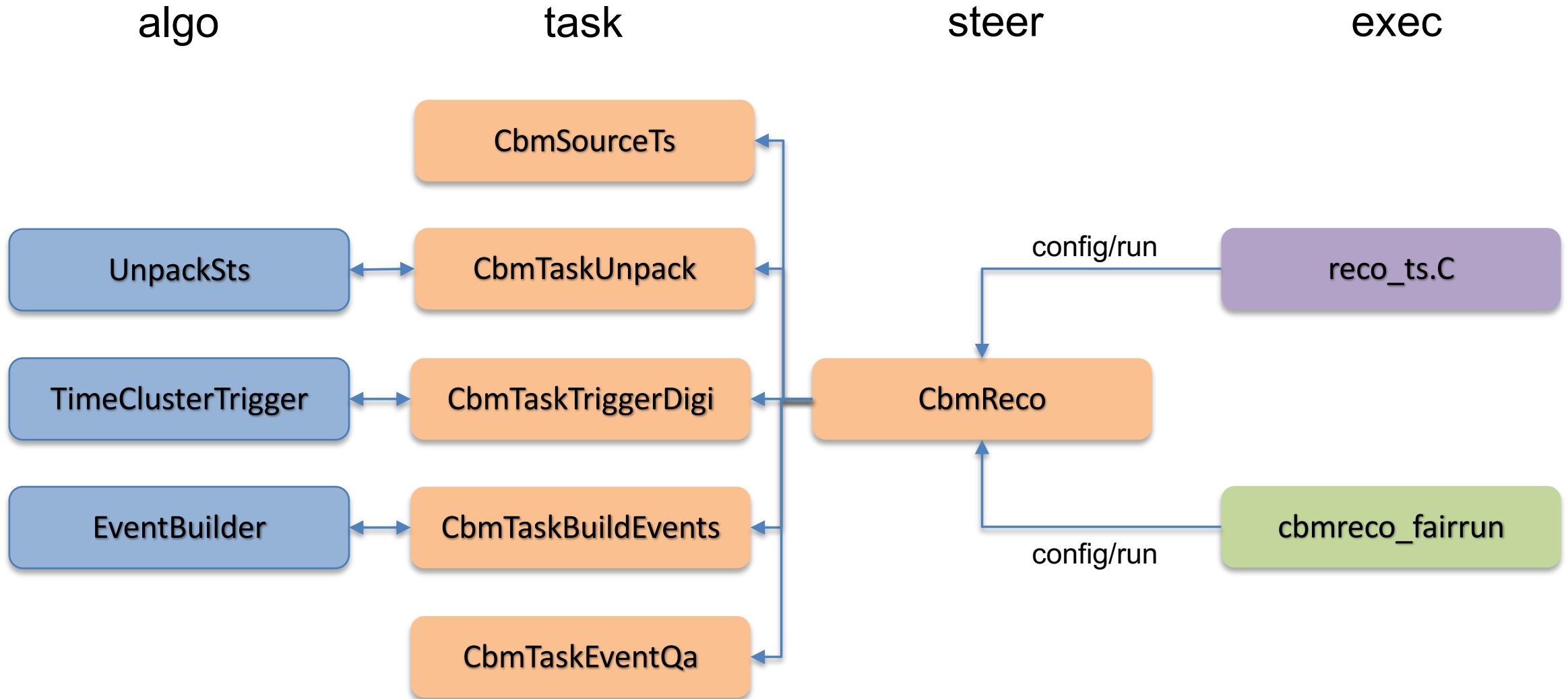


# Status and outlook MS1

V. Friese

CBM Online Meeting, 1 April 2022

# Class Scheme: FairRun/FairTask integration



# Algo Layer

- Classes in top-level directory algo
- namespace cbm::algo
- ROOT-free
- Formulated as functors on the smallest independent data unit
  - Unpacker: FLES microslice -> vector<CbmStsDigi>
  - Digi Trigger: vector<double> (digi times of trigger detector) -> vector<double> (trigger times)
  - Event Builder: CbmDigiTimeslice, one trigger -> one CbmDigiEvent

# Task Layer

- Classes in directory reco/tasks
- no namespace
- Classes inherit from FairTask (-> TTask)
- Read data from ROOT tree
- Instantiate and configure algorithms
- Perform loops over independent data units, call algorithms within loops
- Aggregate data received from algorithms; receive and process monitoring data
- Write data to ROOT tree

# Steering Layer

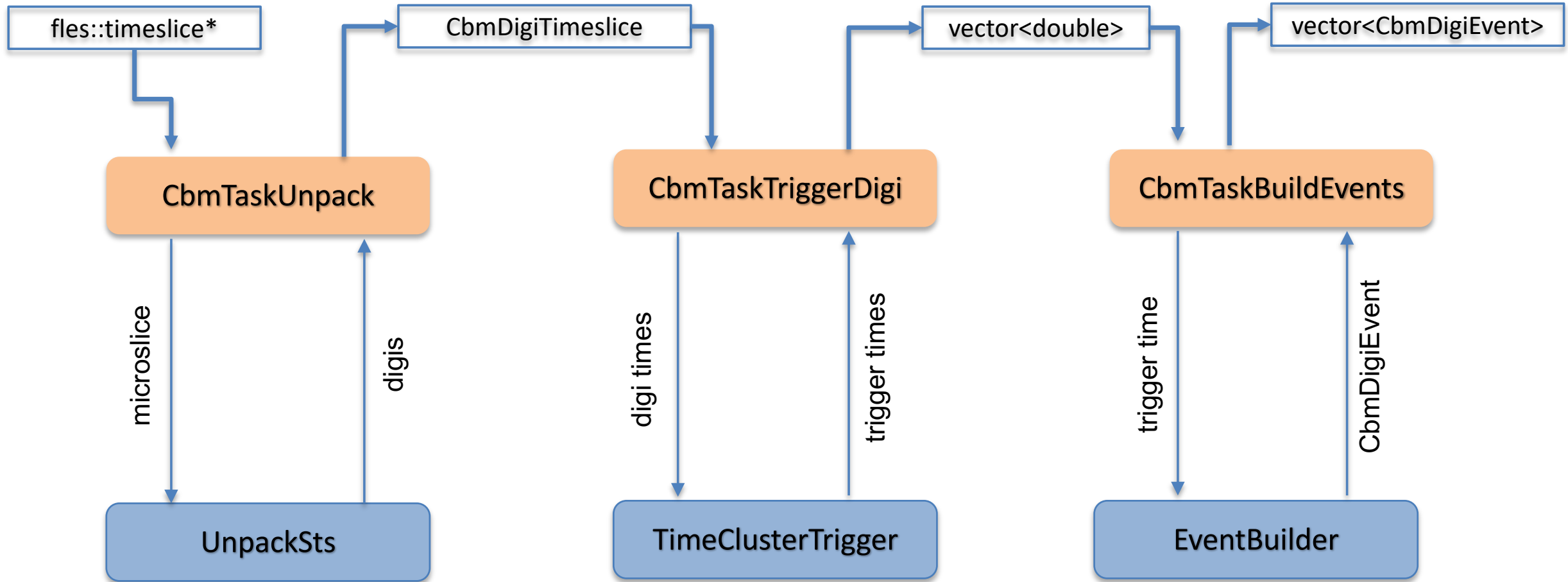
- reco/tasks/CbmReco
- Configured through CbmRecoConfig (parameters of algorithms)
- Wraps FairRunOnline
- Instantiate tasks and registers them to FairRunOnline
- Sequential execution of tasks through FairRunOnline

# Execution layer

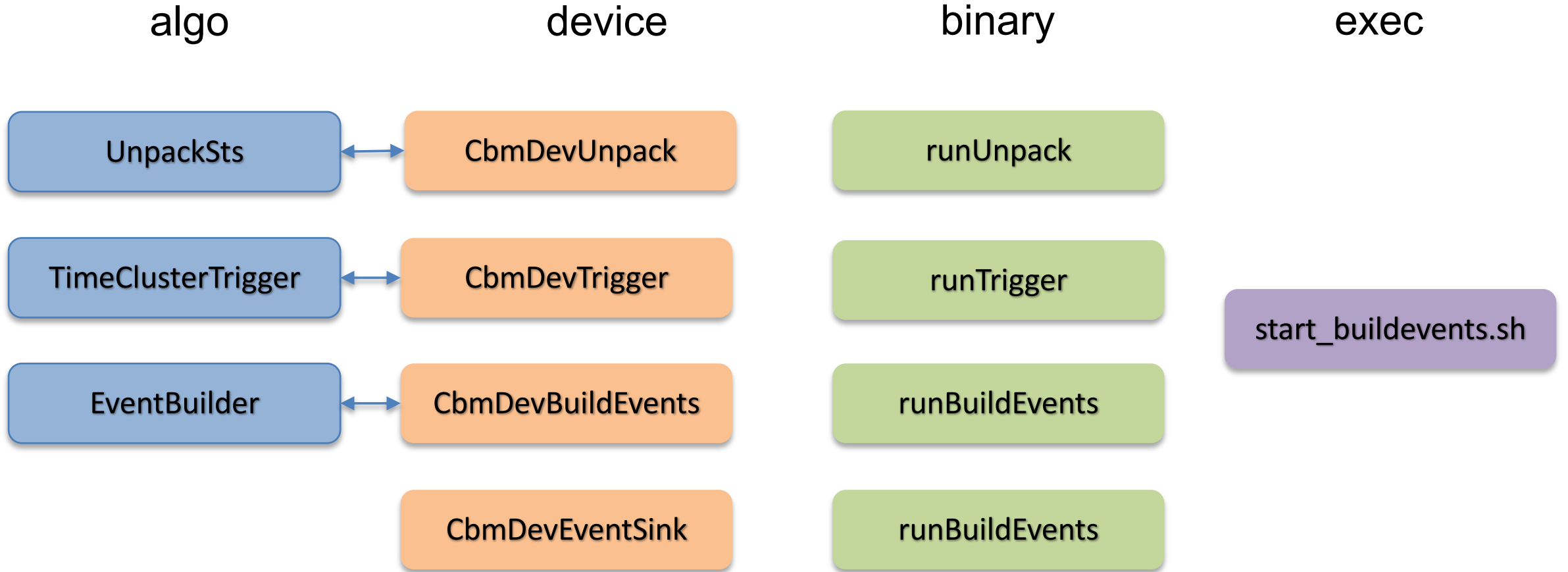
- bin/cbmreco\_fairrun (executable)
- Reads configuration from YAML or JSON file
- Source URI, output and configuration file names through command line arguments
- Instantiates and executes CbmReco
- Alternative: ROOT executable with macro/reco/reco\_ts.C



# Data flow: algo level



# Class Scheme: MQ integration





# Open issues

- Verify STS unpacker against current implementation (same output data)
- How to handle QA histogram(s) (display online)?
- How to execute the standalone binary in an online run?

# Possible further developments within MS1

- Exchange of transient data between tasks not through ROOT tree (performance?)
- Isolate code shared by tasks and devices (loop over algorithm calls): layer between task/device and algo
- Concurrency in MQ scheme: multiple devices for one task?

# Beyond MS1 / next steps

- Include unpackers for other detectors
  - Who can do that? Expertise required.
- Add more processing steps: STS local reconstruction (cluster and hit finding)
  - After event building: event-by-event reco
    - Introduce event filter based on StsHits
  - Before event building: 4d reco;
    - Trigger on STS hits instead of STS digis; event building afterwards