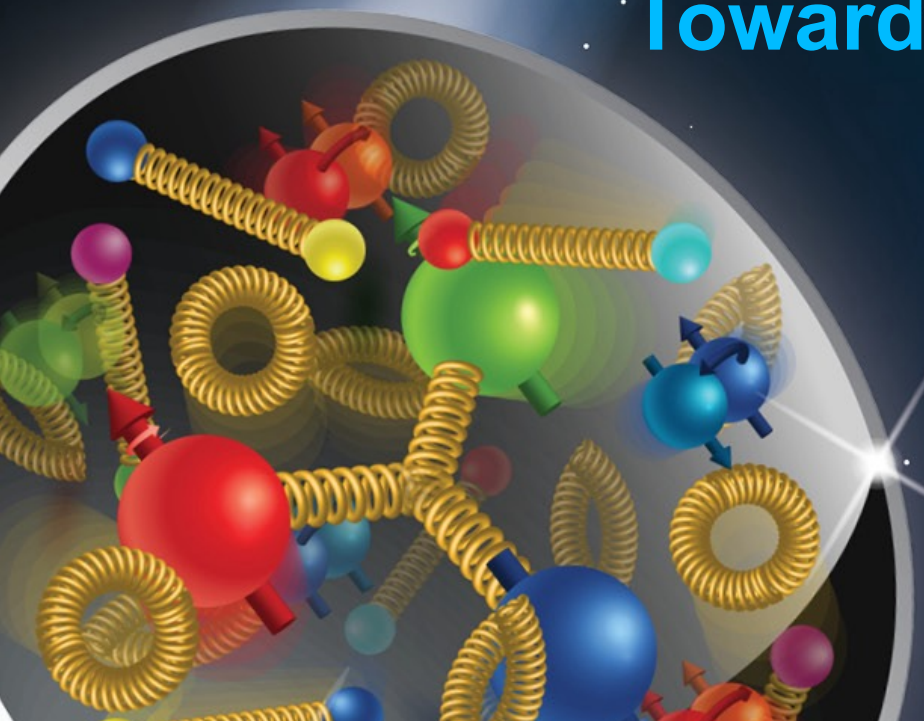
GEANT4
A SIMULATION TOOLKIT



Toward Sub-Event Parallelism

Makoto Asai (JLab/SCT)
asai@jlab.org

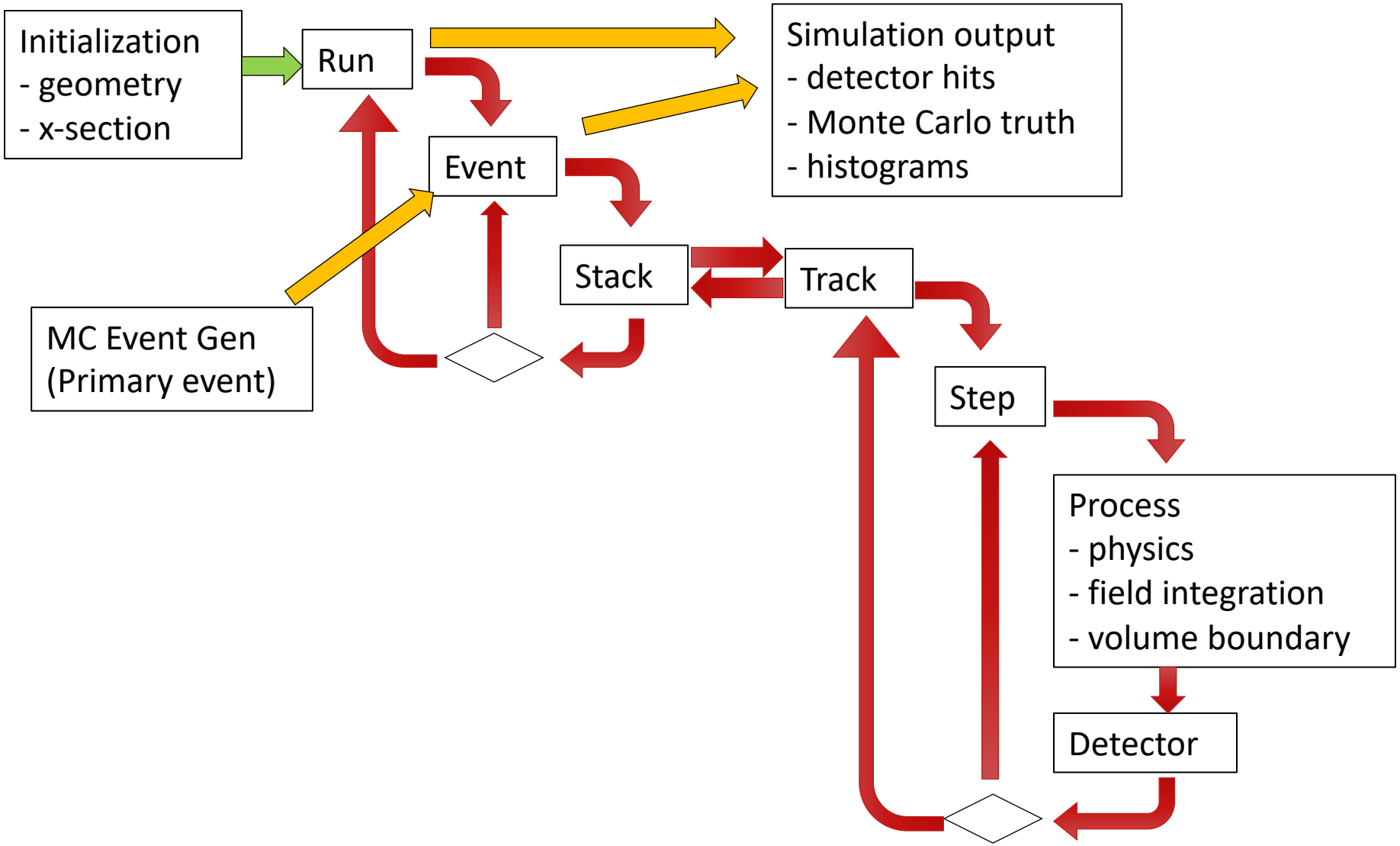
Jefferson Lab
U.S. DEPARTMENT OF ENERGY | Office of Science
JSA

Geant4 evolutions in parallelization

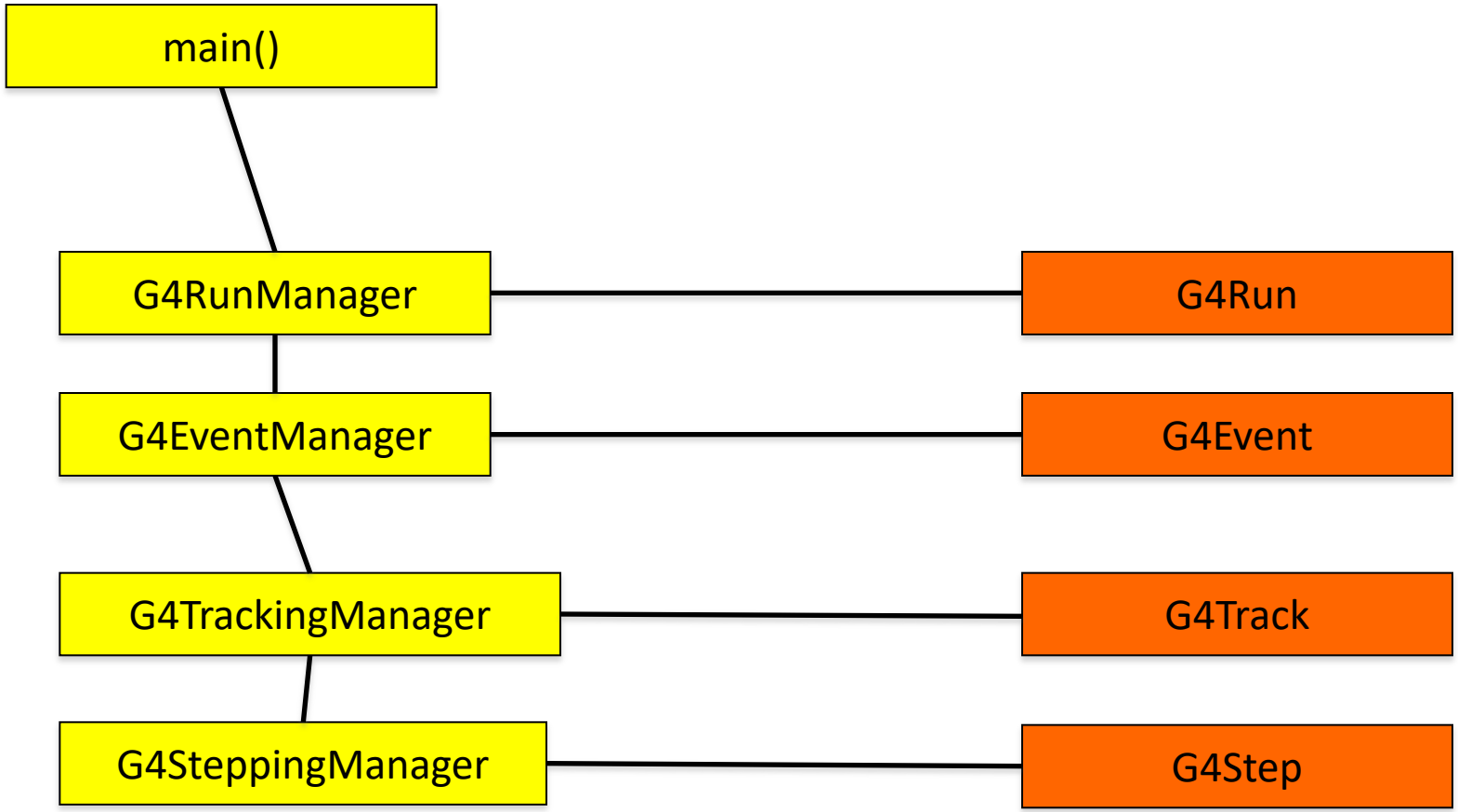
1. Sequential mode : original since Geant4 v1.0
   - Single core (thread) does everything
2. Multithreaded event-level parallel mode : since Geant4 v10.0 (Dec.2013)
   - Taking the advantage of independence of events, many cores (threads) process events in parallel (event-level parallelism)
   - Geometry / x-section tables are shared over threads
3. Task-based event-level parallel mode : since Geant4 v11.0 (Dec.2021)
   - Decoupling task (event loop) from thread
   - More flexible load-balancing
4. Task-based sub-event parallel mode : planned (Dec.2022~)
   - Split an event into sub-events and task them separately
   - Sub-event :
     - Sub-group of primary tracks, or
     - Group of tracks getting into a particular detector component
       - Suitable for heterogeneous hybrid hardware

- N.B. We made these evolutions without forcing the user to migrate
  - Except for using the new functionalities

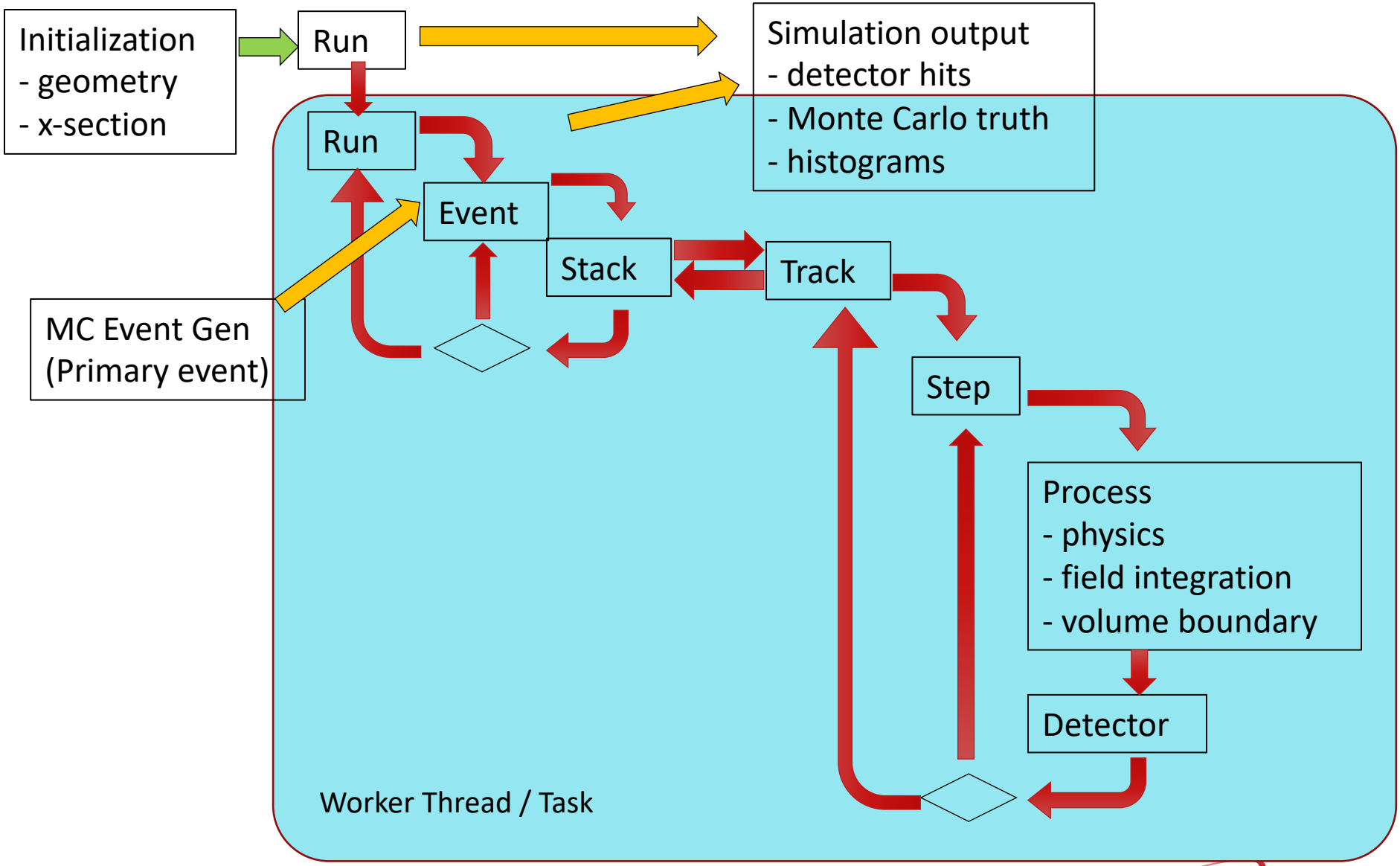
Geant4 as a detector simulation engine



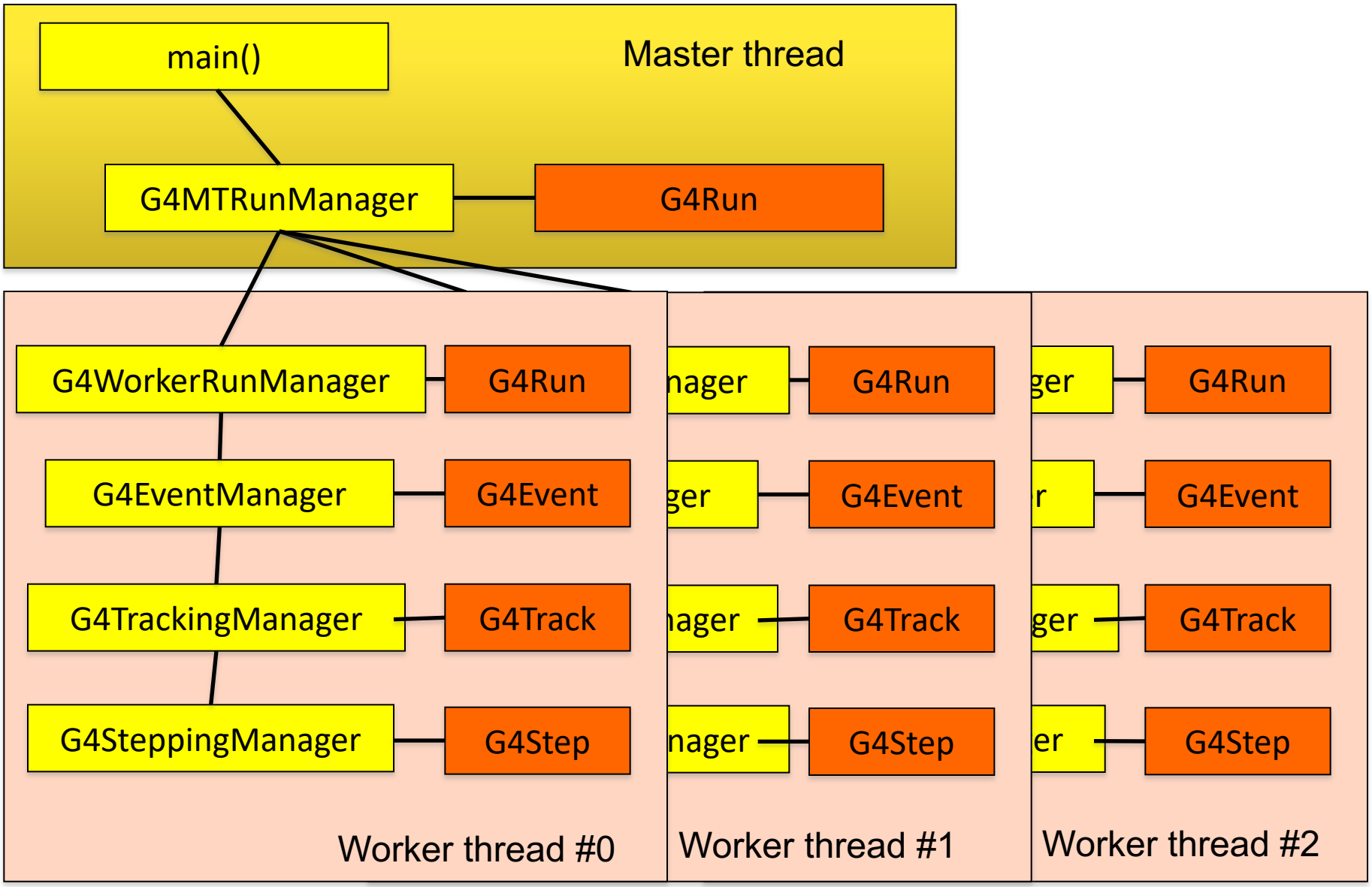
Sequential mode



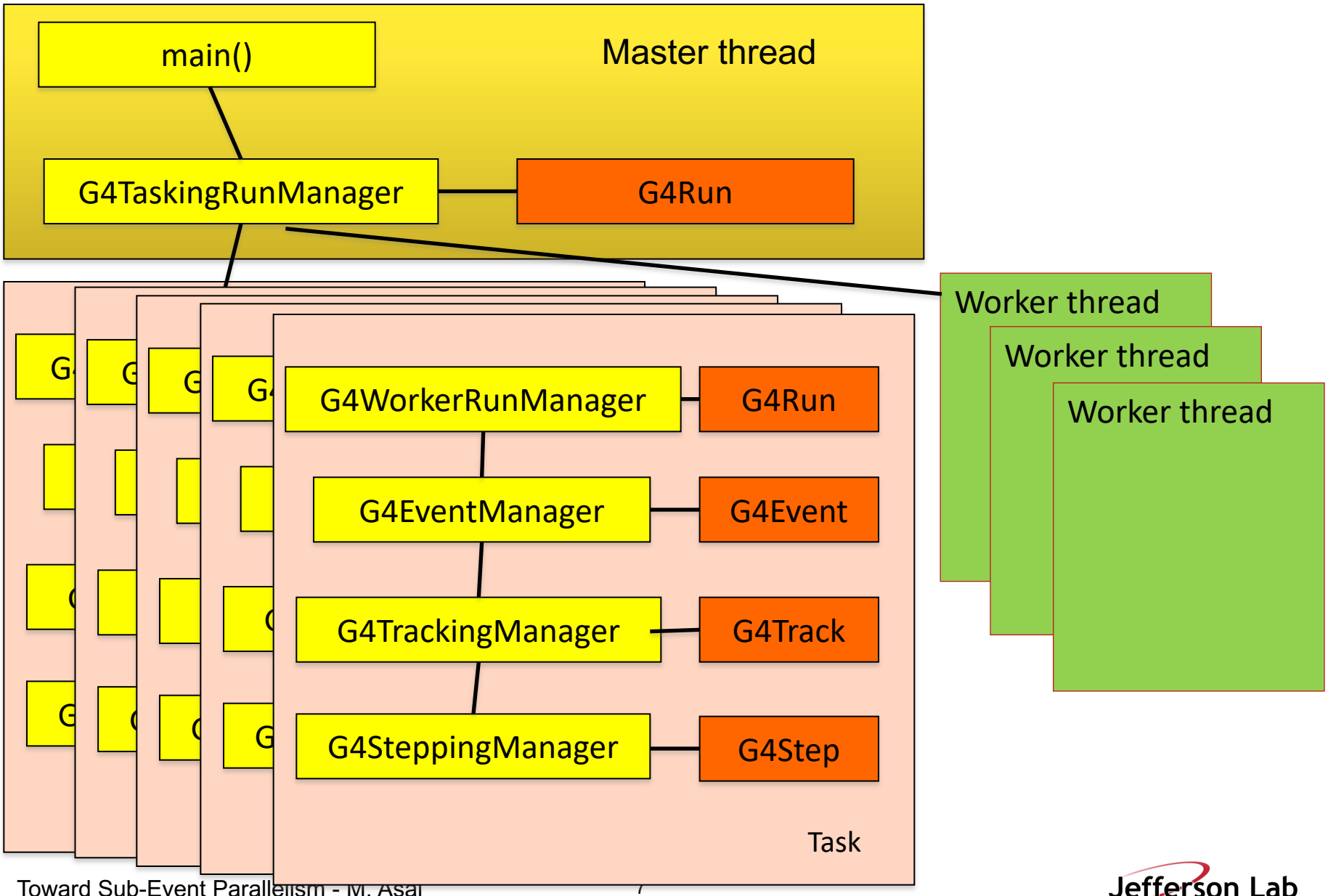
Event-level parallel mode (thread / task)



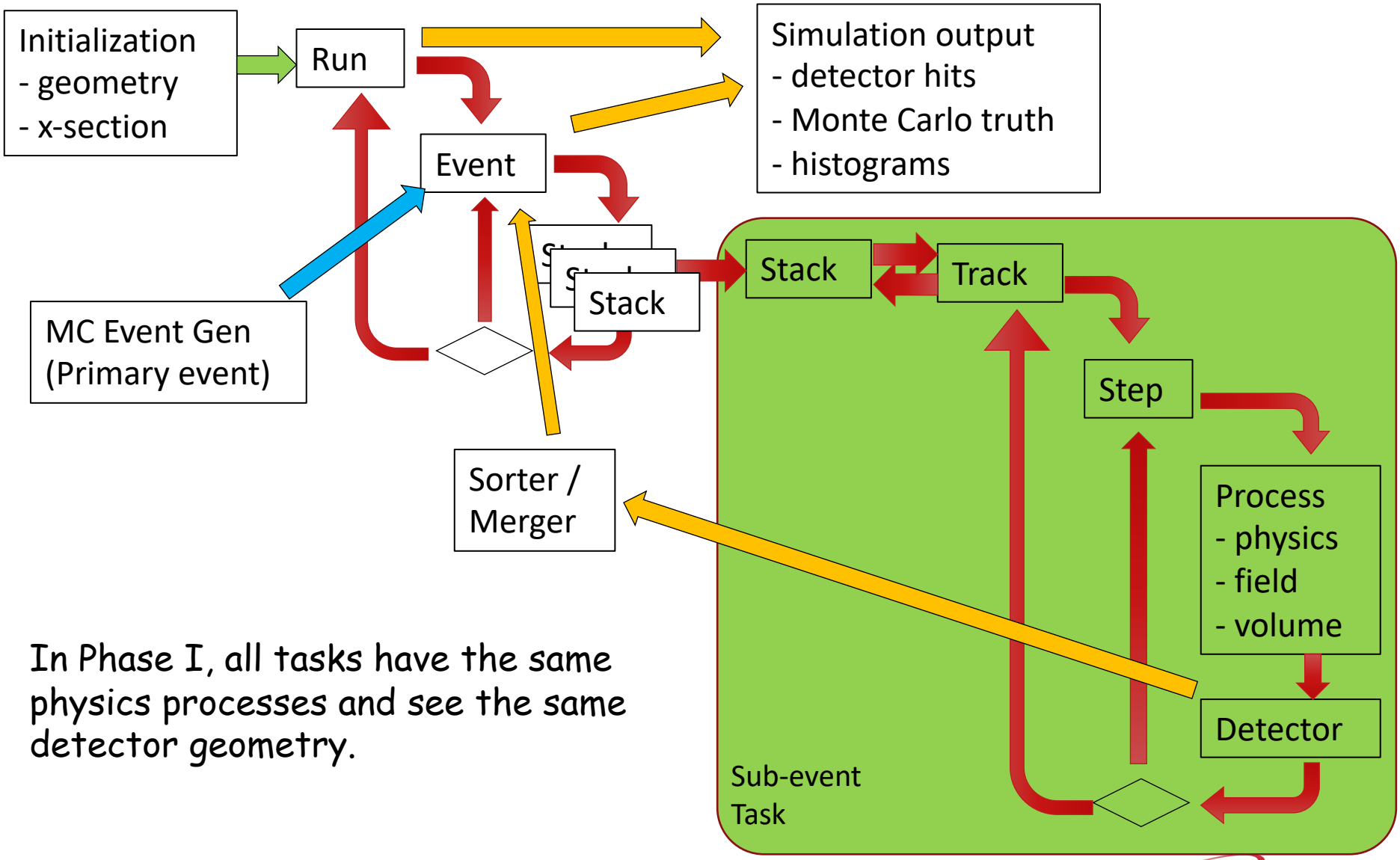
Multithreaded mode



Task-based parallel mode

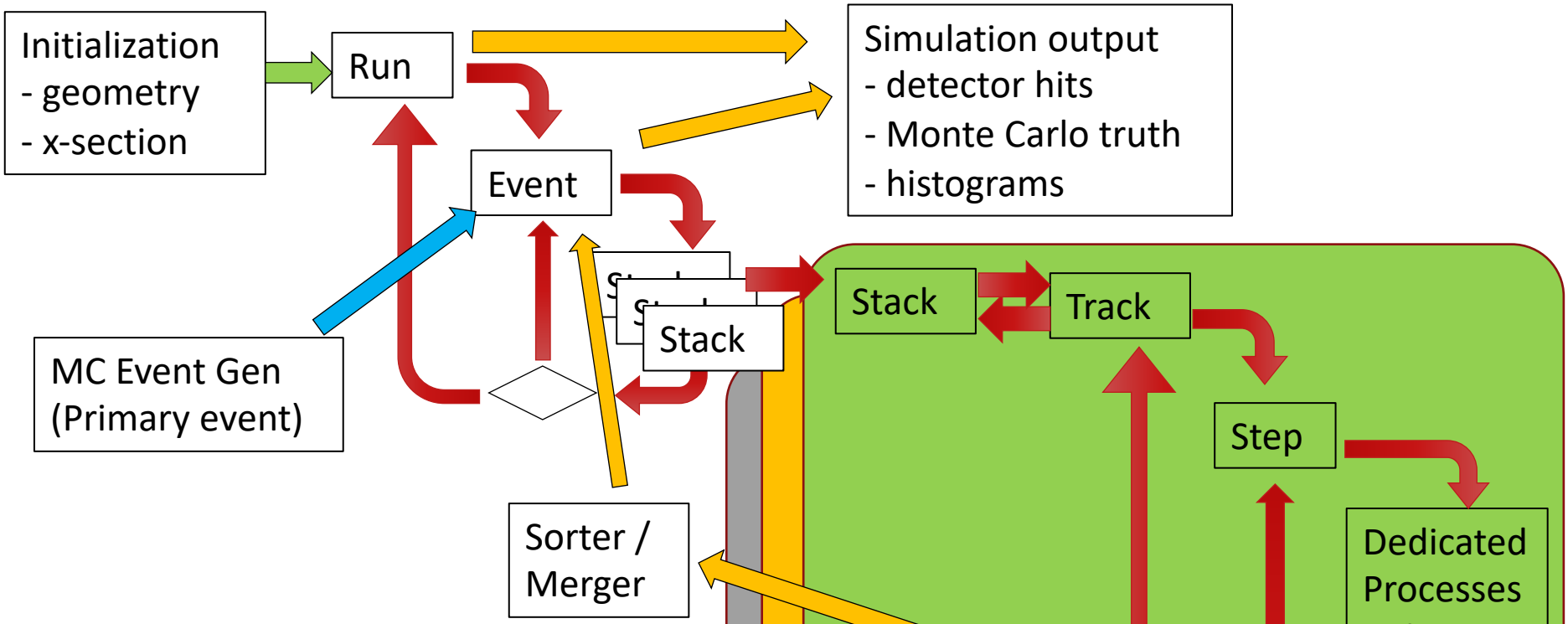


Task-based sub-event parallel mode (Phase I)



In Phase I, all tasks have the same physics processes and see the same detector geometry.

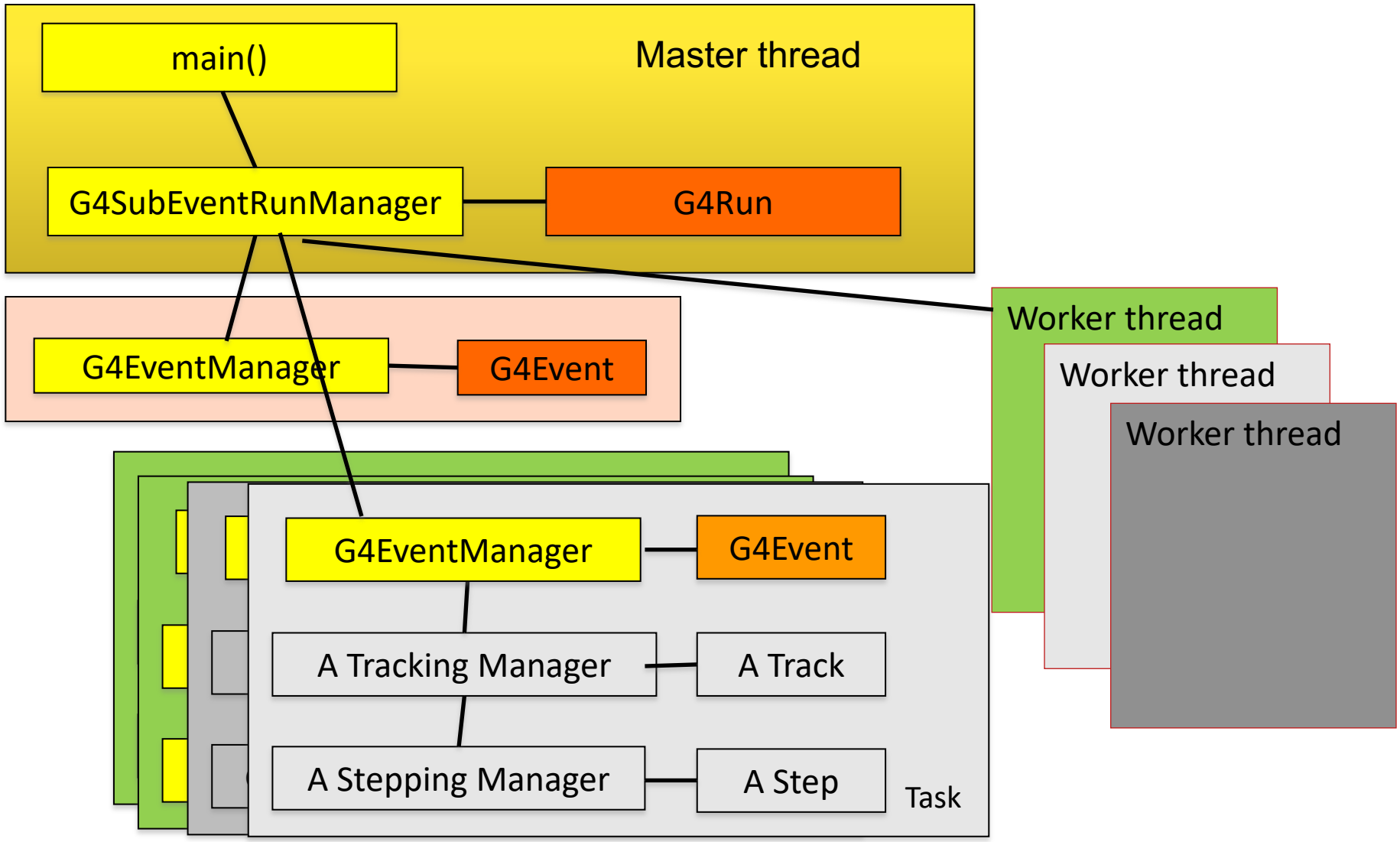
Task-based sub-event parallel mode (Phase II)



In Phase II, each task has only the limited physics processes and sees the limited detector geometry that are necessary for that particular task.

- Essential for heterogeneous simulation

Sub-event parallel mode



Monte Carlo simulation on GPU

- It is hopeless to port the entire Geant4 to a single process on GPU.
  - Each GPU process should have strictly limited scope.
    - Physics coverage, particle type, geometry/material
    - E.g. optical photon transport in Cerenkov detector, EM shower in calorimeter (w/o back splash or punch through)
- A task on GPU should behave like a blackhole.
  - The darker a task is, the better performance it has.
    - "Darker" means "less output".
  - Individual step/track/trajectory should not be taken out from a task.
    - Reshuffling tracks over tasks is no a good thing to do.
  - Minimize output information.
    - E.g. transferring output is a serious bottleneck, even for shared memory.
- Sub-event parallelism is the only solution that allows various tasks running on GPU in parallel while conducting the full event simulation.

Heterogeneous simulation with Geant4

- Simulation throughput needs to increase by O(10-100)
  - Contribution of simulation to maintain the systematic uncertainty of the experiment.
  - New detector hardware comes with higher demands
    - One of the primary targets is Electron Ion Collider.
      - Detector design/construction will start soon.
- Strategy
  - Heterogeneous computing
    - The main (master) process manages sub-tasks.
    - Each sub-task has strictly limited scope with only the limited kind of physics processes and particle types, and see only the limited detector geometry that are necessary for that particular task.
      - Essential for heterogeneous simulation
  - On CPU
    - When sustainable or when no alternative
      - With creativity (e.g. stack management, event biasing, …)
    - Managing / integrating granular sub-tasks
  - On GPU (or co-processors,…)
    - Variety of single-purpose, optimized sub-tasks
    - Output must be minimized

Sub-event parallelism in Geant4

- Sub-event parallelism will be introduced without forcing user's code to migrate.
  - Sequential, threading and tasking modes will work fine as they do now.
- To use sub-event parallelism (for Phase-I)
  - Use a newly introducing RunManager for sub-event parallelism
  - Implement UserStackingAction to sort tracks into sub-events
  - Implement merge() method in G4Event if special merging treatment is required
    - Ordinary HitCollection and HitVector (for scoring) will be automatically merged.
- To use sub-event parallelism (for Phase-II)
  - In addition to above
  - Physics list and/or detector construction dedicated to each task if needed
    - For example with G4HepEM

Questions?