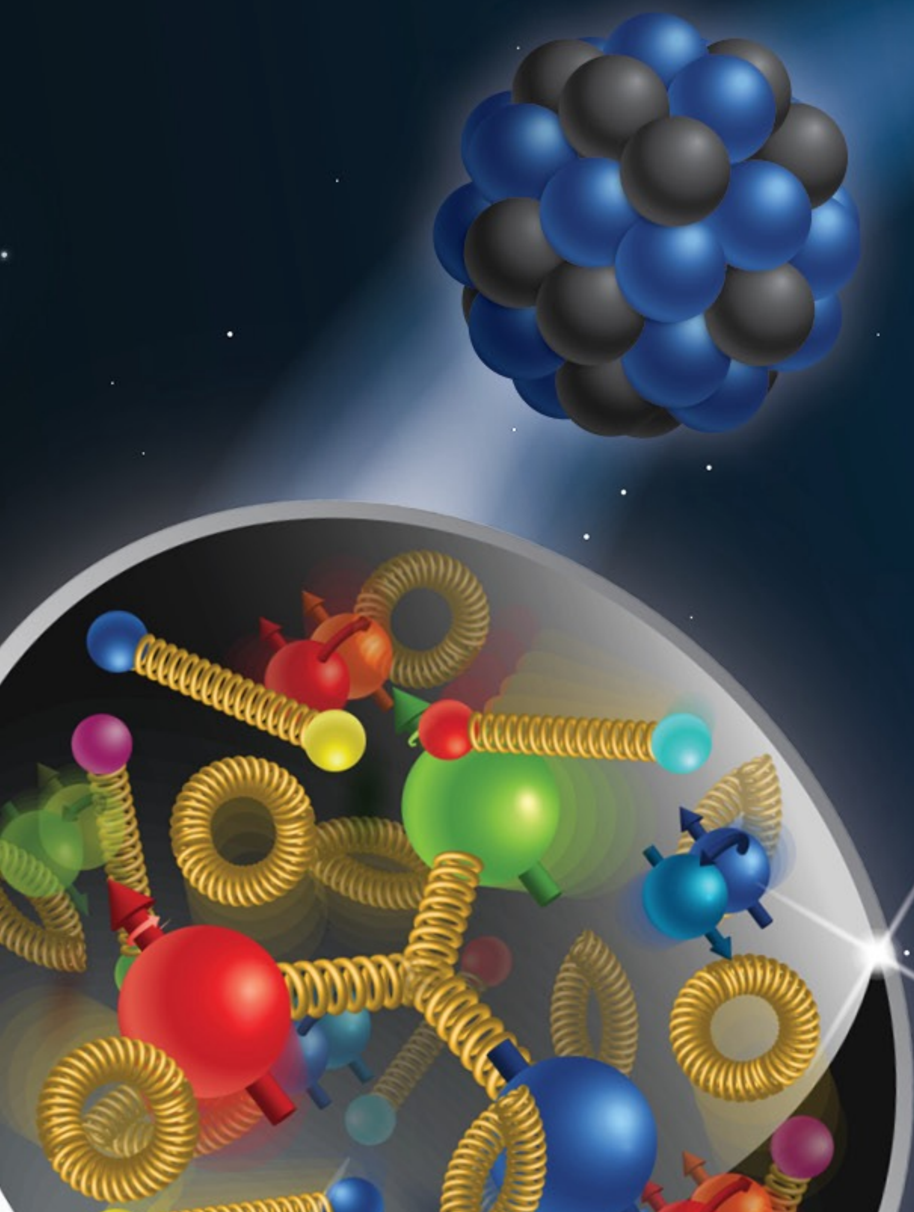


Kernel updates

Makoto Asai (Jefferson Lab)

15th Geant4 Space Users Workshop

Pasadena, CA, USA

December 6th, 2023

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

Releases

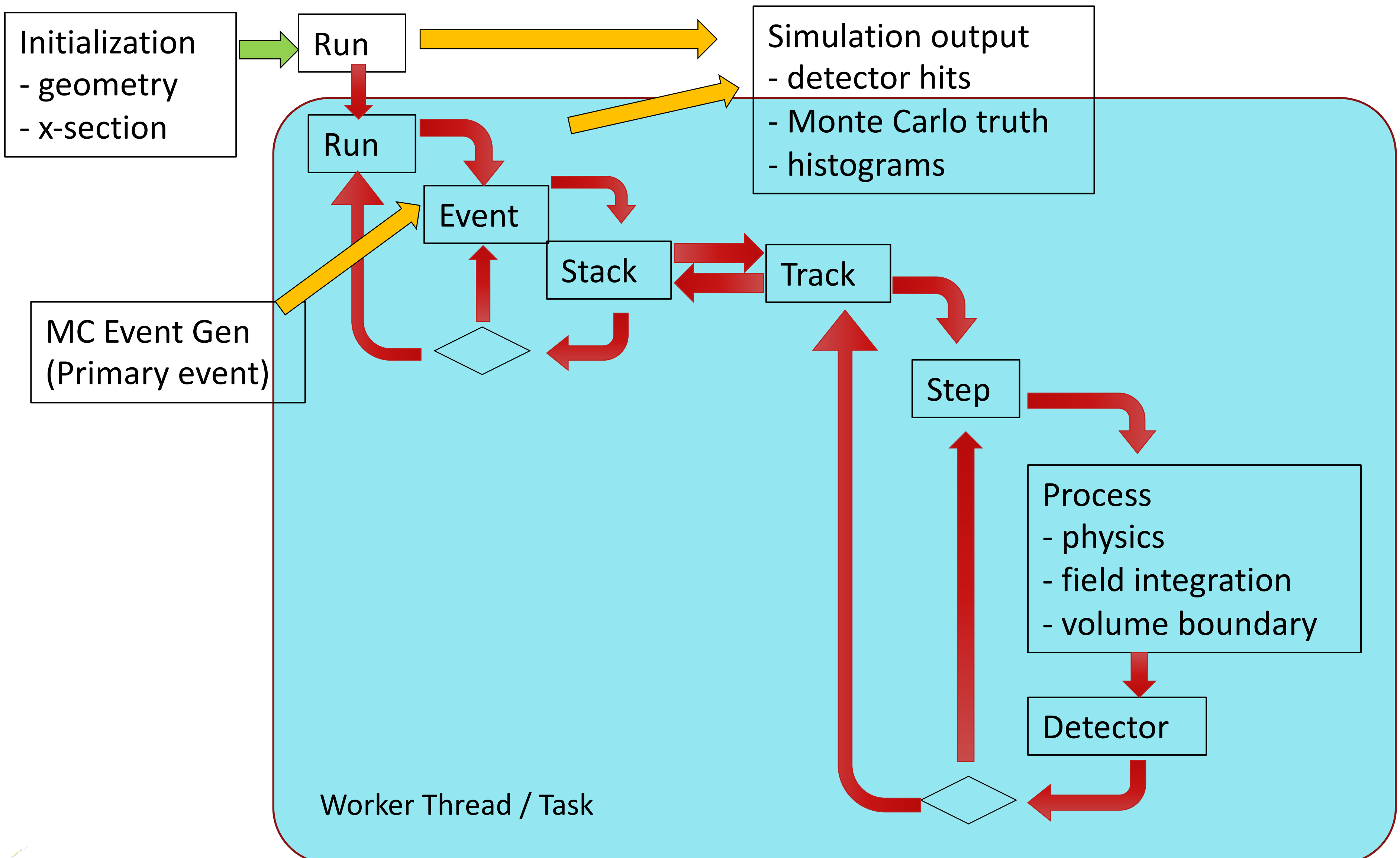
- 10.6 – Dec 06, 2019
  - 10.6-patch03 – Nov 06, 2020
- 10.7 – Dec 04, 2020
  - 10.7-patch04 – Sep 09, 2022
- 11.0 – Dec 12, 2021
  - 11.0-patch04 – Mar 14, 2023
- 11.1 – Dec 09, 2022
  - 11.1-patch03 – Nov 10, 2023
- **11.2 – Dec 08, 2023**

  - **Past releases and their release notes are available at https://geant4.web.cern.ch/download/all**

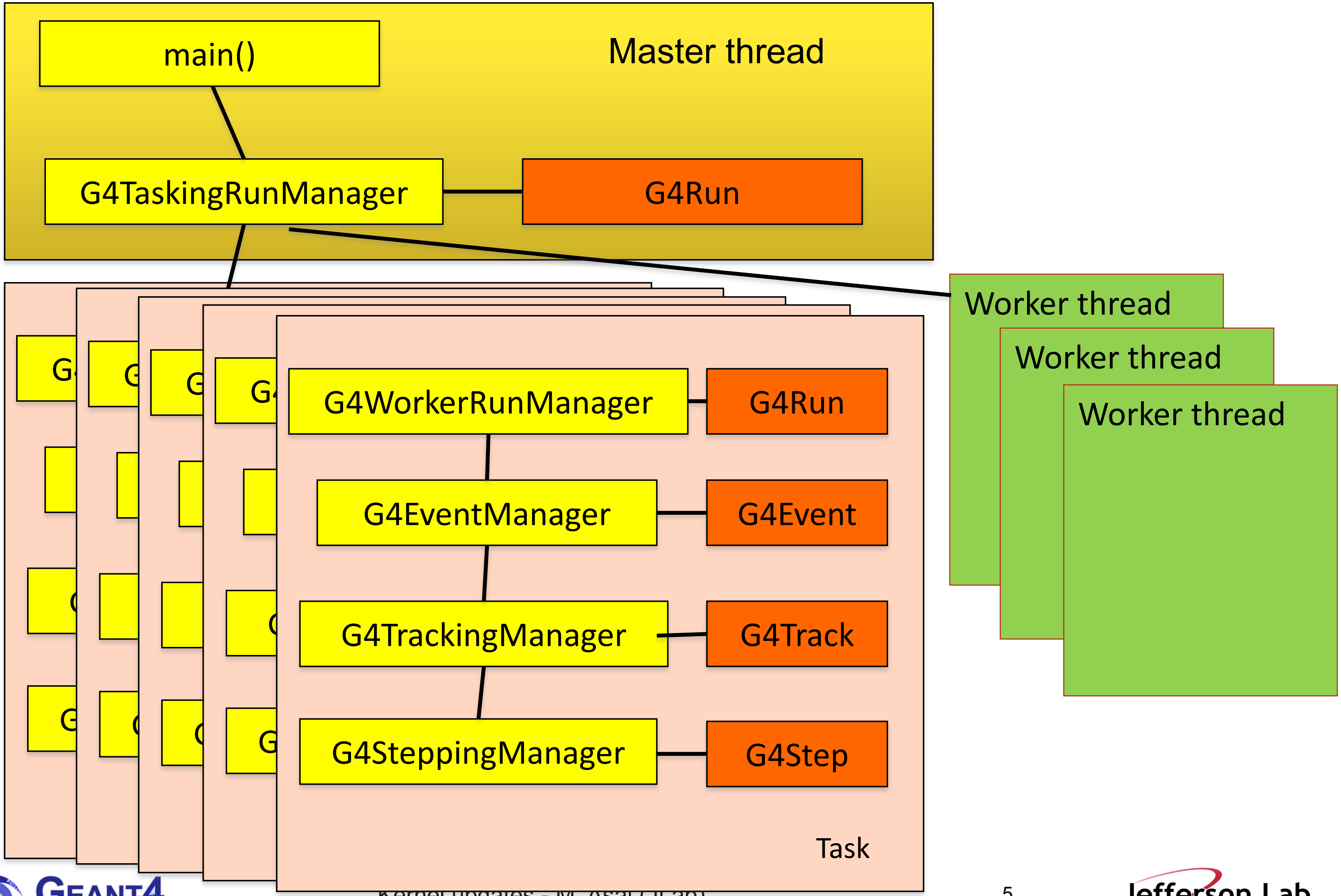
Kernel (run, event, tracking, particle, etc.)

- Tasking with PTL (Parallel Tasking Library) is now the default multithreading mode.
  - Sequential mode as well as multithreading mode with p-thread or TBB (Tasking Building Block) are also available.
- Introducing G4VTrackingManager, an interface for custom tracking managers specialized for a small number of selected particle types.
  - G4EventManager hands over tracks to custom tracking manager, if registered for the particle.
- New G4SteppingVerboseWithUnits class, an alternative to G4SteppingVerbose with printout of proper units.
  - Simplified way of defining a user-specific stepping verbose, that is now common to all sequential, MT and tasking modes.
- Partial introduction of G4SubEvent and related functionalities.

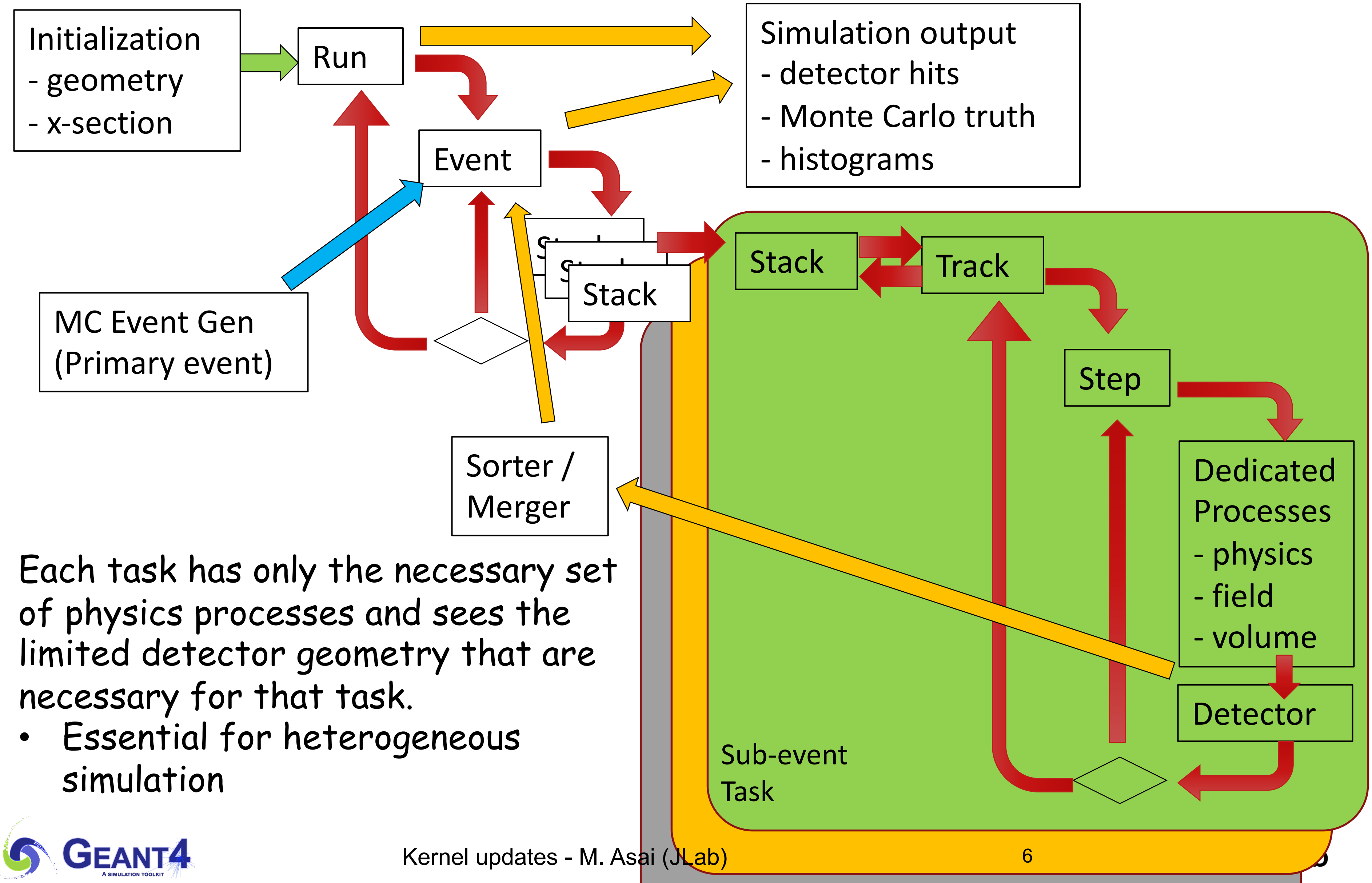
Event-level parallel mode (thread / task)



Task-based parallel mode



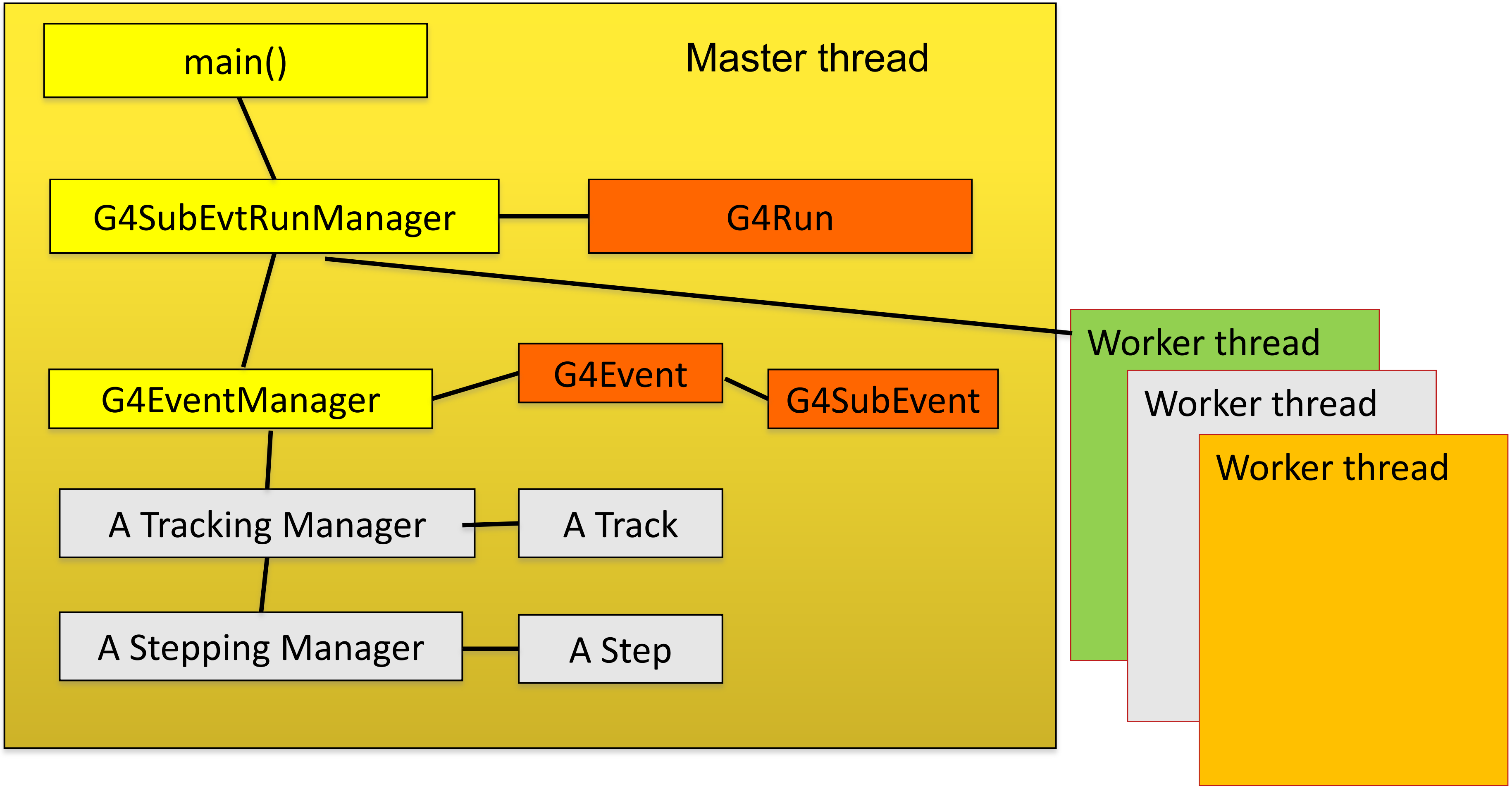
Task-based sub-event parallel mode (in 2024~)



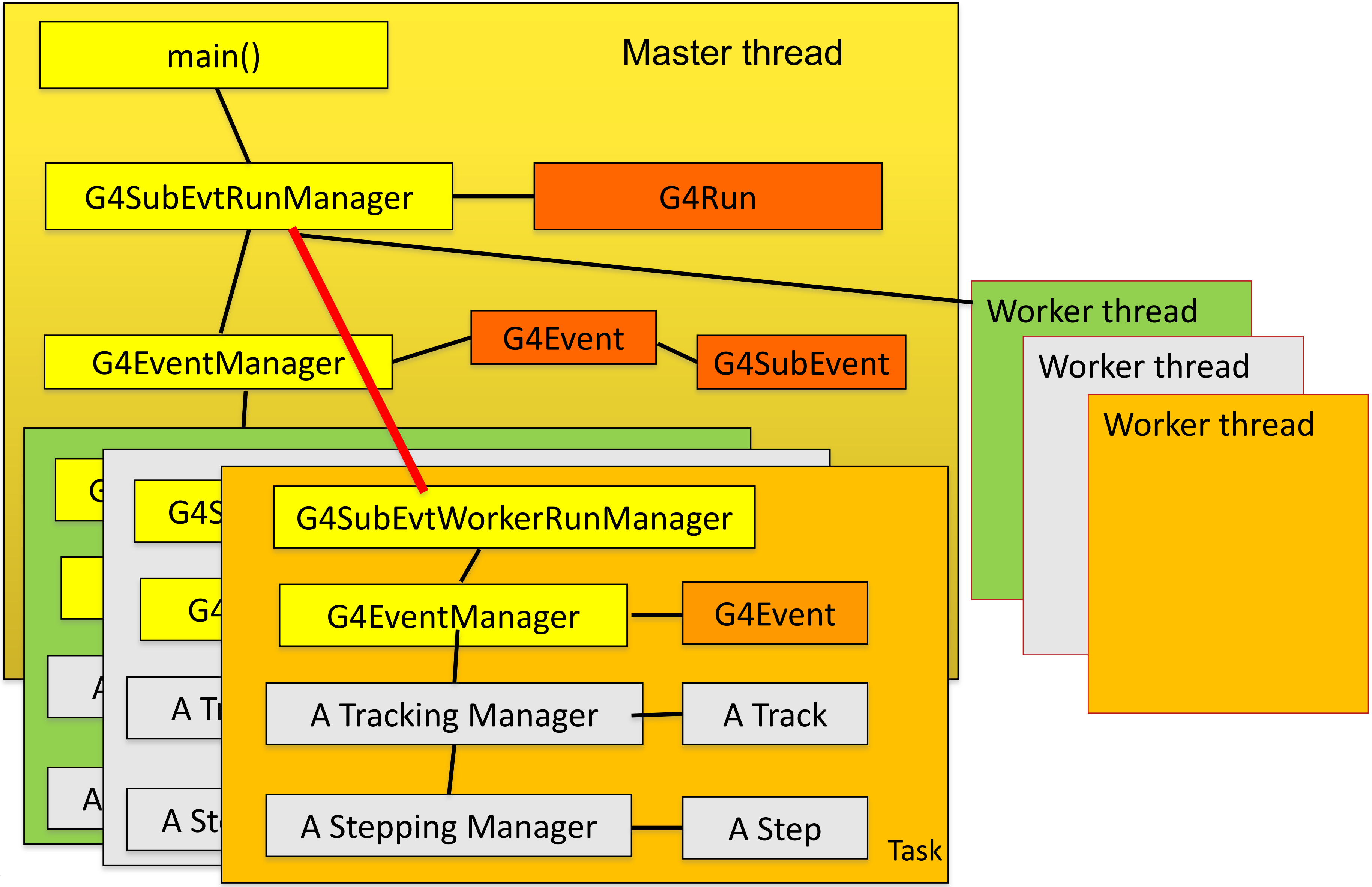
Each task has only the necessary set of physics processes and sees the limited detector geometry that are necessary for that task.

- Essential for heterogeneous simulation

Sub-event parallel mode (to be released in 2024~)



Sub-event parallel mode (to be released in 2024~)



Kernel (run, event, tracking, particle, etc.)

- Enabled VecGeom wrapper for G4GenericPolycone, G4EllipticalTube, G4EllipticalCone and G4Ellipsoid. Requiring VecGeom version 1.1.5.
- Added support for writing out assembly envelopes in GDML.
- Added ability to optionally check for overlaps in parallel geometries through the /geometry/run/test UI command.
- A new class G4TransportationParameters enables fine grain control of parameters for killing charged particles looping in a field.
  - It is optional, but if created it applies to all stable charged particles. It must be instantiated in the master thread and all its parameters need to be assigned values.
- The inheritance level for G4TouchableHistory has been removed, simplifying touchables handling.
  - G4VTouchable is now a simple typedef of G4TouchableHistory, therefore forward declarations of G4VTouchable in the code will have to be promoted to concrete inclusion of the related header file.
  - Unused concrete touchable types G4GRSSolid and G4GRSVolume have been retired and are no longer available.

Kernel (run, event, tracking, particle, etc.)

- Particle properties have been updated according to PDG-2023.
- Added G4DensityEffectCalculator: new class, providing on fly computation of the density effect correction using "exact" formulas.
- New class G4OpticalMaterialProperties, a store for optical material properties and allow use of predefined optical material properties.
- In G4NistMaterialBuilder, all liquid materials are marked as *kLiquid*.

Kernel (run, event, tracking, particle, etc.)

- Simplified G4String interfaces and implementation with C++11/17.
  - Removed obsolete/std::string synonym member functions and deprecated non-std::string member functions, replacing with new free functions in G4StrUtil namespace.
  - Might require user code adaptation where G4String functions are explicitly used; for details see https://geant4.web.cern.ch/download/release-notes/notes-v11.0.0.html#string-migration-notes
    - E.g.

```
auto i = aG4String.index("arg");  →  auto i = aG4String.find("arg");
bool c = aG4String.isNull();      →  bool c = aG4String.empty();
```

- New Geant4-FLUKA interface and two hadronic examples, providing access to FLUKA-CERN hadron-nucleus inelastic physics.

Three types of command-based scorers

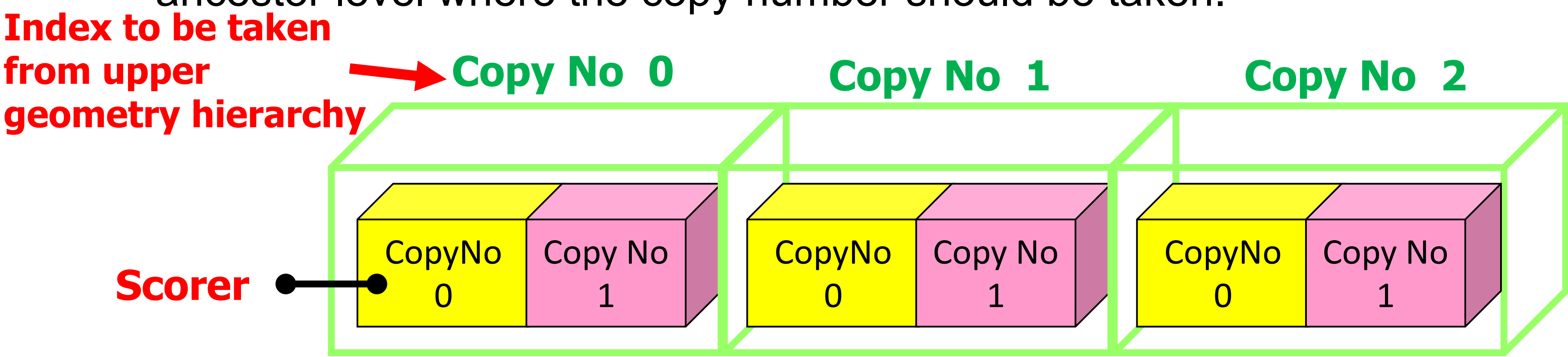
1) Scoring mesh
   - Define 3-D mesh (box or cylinder)
   - The mesh may overlap with real-world volumes
   - Assign arbitrary number of primitive scorers to mesh cell
2) Assigning scorers to a real-world logical volume
   - Declare a real-world logical volume as a detector
   - Assign arbitrary number of primitive scorers to the detector
   - If the volume is placed more than once, assigned scorers individually score for each physical volume
3) Scoring probe
   - A probe is a small cube that is located at arbitrary position. It may overlap with real-world volumes.
   - Assign arbitrary number of primitive scorers to the probe
   - If probe is placed more than once, assigned scorers individually score for each probe.

Define scorer to a tracking volume

- Define a scorer to a logical volume.

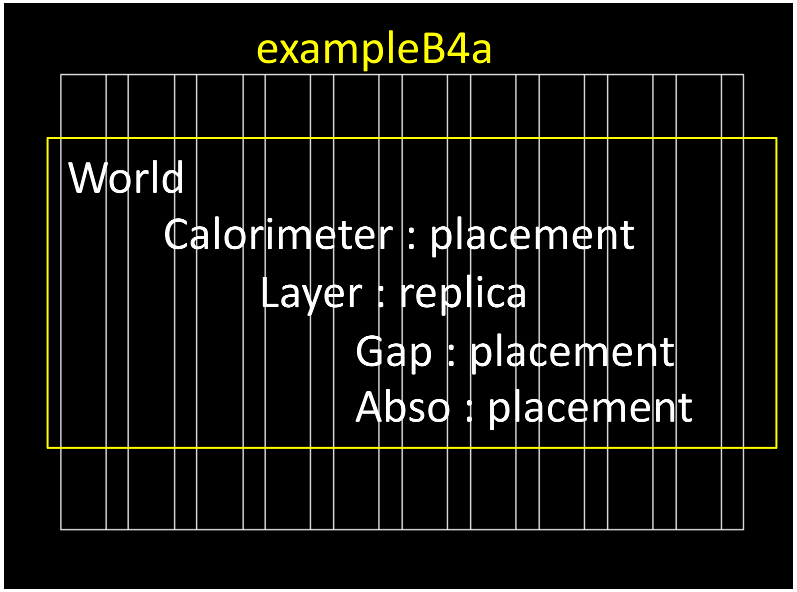
```
/score/create/realWorldLogVol <LV_name> <anc_lvl>
```

- One can define arbitrary scoring quantities and filters.
  - Same recipe as scoring mesh.
  - Scores are automatically merged other worker threads and written to a file.
  - Drawing is not yet supported.
- All physical volumes that share the same *<LV_name>* have the same primitive scorers but score separately.
  - Copy number of the physical volume is the index.
  - If the physical volume is placed only once, but its (grand-)mother volume is replicated, use the *<anc_lvl>* parameter to indicate the ancestor level where the copy number should be taken.



Command-based real-world scorer

- Do not use this /score/create/realWorldLogVol command to a mother logical volume.
  - For example of this exampleB4, "Layer" is fully filled with "Gap" and "Abso" daughter volumes. You won't see any energy deposition in "Layer" volume.

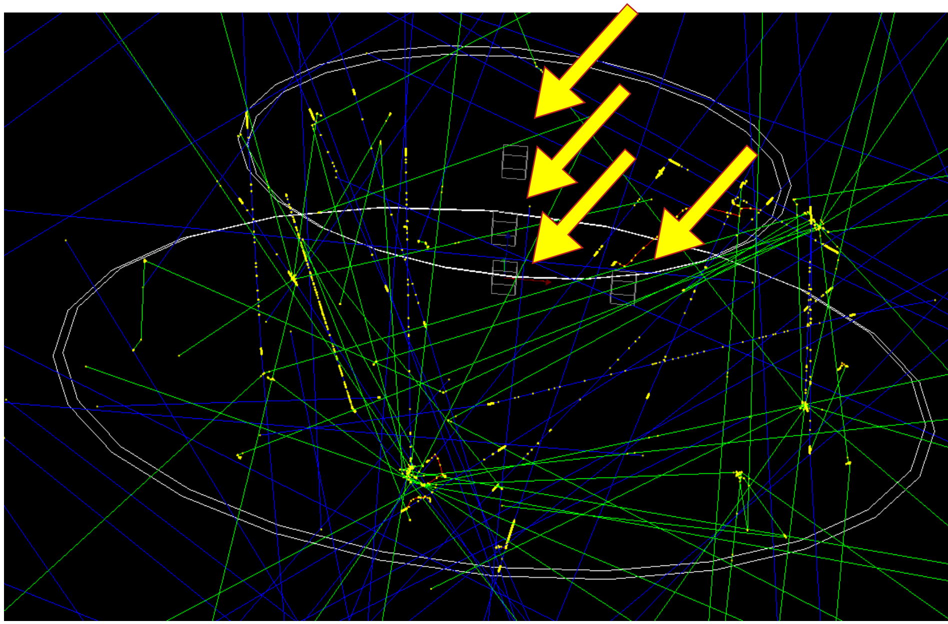


/score/create/realWorldLogVol *Gap* ***1***
/score/quantity/energyDeposit *eDep* MeV
/score/quantity/trackLength *sLen* mm
/score/filter/charged *cFilter*
/score/create/realWorldLogVol *Abso* ***1***
/score/quantity/energyDeposit *eDep* MeV
/score/quantity/trackLength *sLen* mm
/score/filter/charged *cFilter*
/score/close

If this is not set, given "Gap" and "Abso" are placed with copy number 0, energy deposition and track length are accumulated for all layers.

Command-based probe scorer

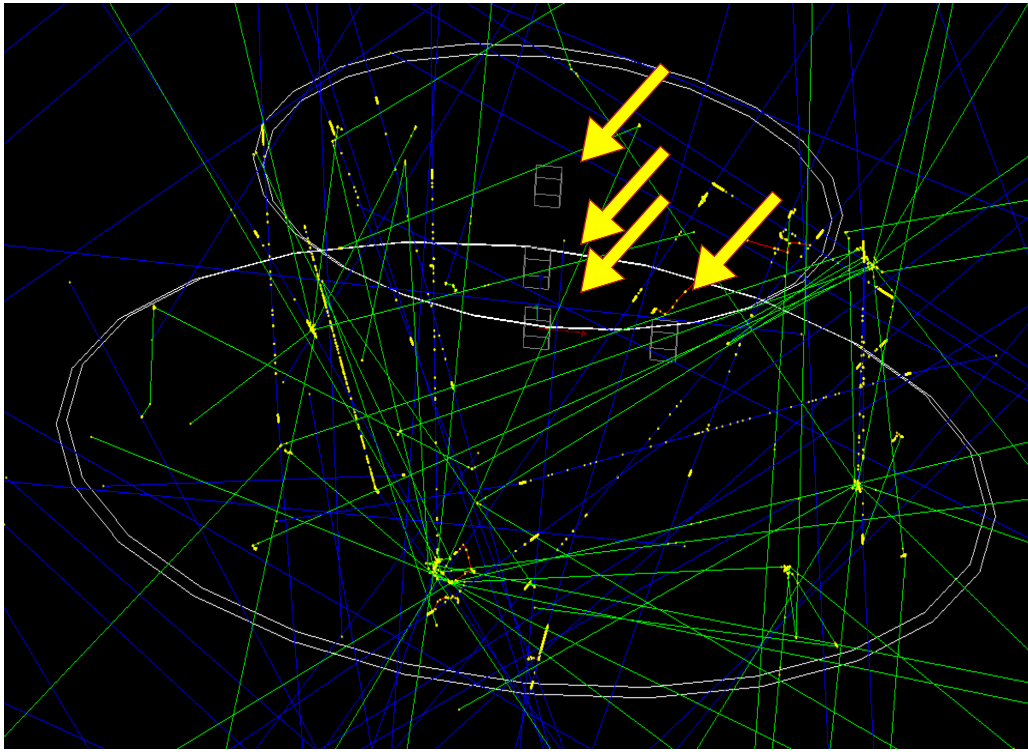
- User may locate scoring "probes" at arbitrary locations. A "probe" is a virtual cube, to which any Geant4 primitive scorers could be assigned.
- Given these probes are located in an artificial "parallel world", probes may overlap to the volumes defined in the mass geometry.
- If probes are located more than once, all probes have the same scorers but score individually.



- In addition, the user may optionally set a material to the probe. Once a material is set to the probe, it overwrites the material(s) defined in the mass geometry when a track enters the probe cube.
  - Because of this overwriting, physics quantities that depend on material or density, e.g. energy deposition or dose, would be measured accordingly to the specified material
- Once a probe is defined, user can associate arbitrary number of primitive scorers and filters like the conventional scoring mesh.
- All probes have the same scorers but score individually.

Scoring probe

```
/score/create/probe Probes 5. cm
/score/probe/material G4_WATER
/score/probe/locate 0. 0. 0. cm
/score/probe/locate 25. 0. 0. cm
/score/probe/locate 0. 25. 0. cm
/score/probe/locate 0. 0. 25. cm
/score/quantity/energyDeposit eDep MeV
/score/quantity/doseDeposit dose mGy
/score/quantity/volumeFlux volFlx
/score/quantity/volumeFlux protonFlux
/score/filter/particle protonFilter proton
/score/close
```



Note: To visualize the probes defined in a parallel world, the following command is required.

/vis/drawVolume worlds

1-D histogram directly filled by a primitive scorer

- Through a newly introduced interface class (G4TScoreHistFiller) a primitive scorer can directly fill a 1-D histogram defined by G4Analysis.
  - Track-by-track or step-by-step filling allows command-based histogram such as energy spectrum.
- G4TScoreHistFiller template class must be instantiated in the user's code with his/her choice of analysis data format.

```
#include "G4AnalysisManager.hh"
#include "G4TScoreHistFiller.hh"
auto analysisManager = G4AnalysisManager::Instance();
analysisManager->SetDefaultFileType("root");
auto histFiller = new G4TScoreHistFiller<G4AnalysisManager>;
```

- Primitive scorer must be defined in advance to setting a histogram.
- Histogram must be defined through /analysis/h1/create command in advance to setting it to a primitive scorer.
- This functionality is available only for primitive scorers defined in real-world scorer or probe scorer.
  - Not available for box or cylindrical mesh scorer due to memory consumption concern.

1-D histogram directly filled by a primitive scorer

/score/create/probe Probes 5. cm

/score/probe/locate 0. 0. 0. cm

/score/quantity/volumeFlux volFlux

/score/quantity/volumeFlux protonFlux

/score/filter/particle protonFilter proton

/score/close

/analysis/h1/create volFlux Probes_volFlux

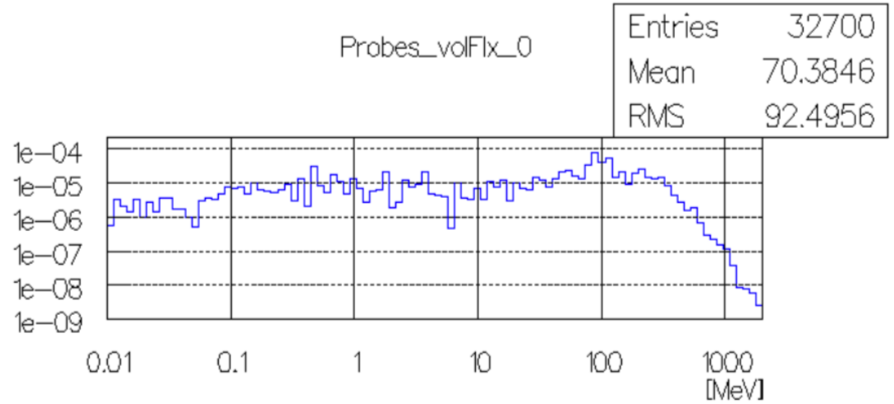
100 0.01 2000. MeV ! log

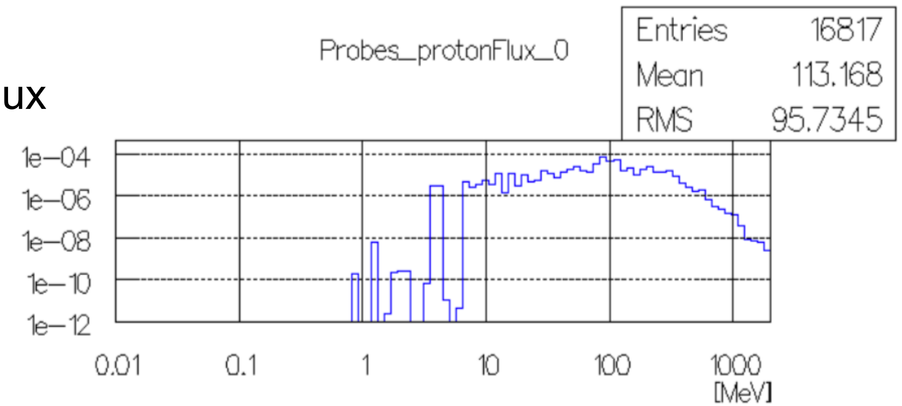
/score/fill1D 1 Probes volFlux

/analysis/h1/create protonFlux Probes_protonFlux

100 0.01 2000. MeV ! log

/score/fill1D 2 Probes protonFlux





N.B. If probe is placed more than once, *fill1D* command should be called to each *copyNo.*

/score/fill1D 1 Probes volFlux 0

Gorad

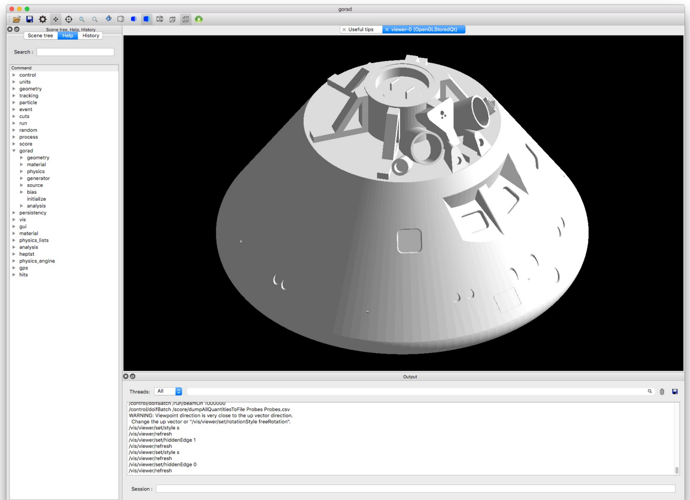
- ***Gorad*** (Geant4 Open-source Radiation Analysis and Design) is a new Advanced Example released with Geant4 version 11.0.
- ***Gorad*** is developed as a turn-key application for radiation analysis and spacecraft design built on top of Geant4.
  - ***Gorad*** is developed under the NASA JSC contract *NNJ15HK11B* to be used primarily for radiation shielding studies of Orion spacecraft.
- As a turn-key application, user of ***Gorad*** does not need to write any C++ source code.
  - Simulation geometry should be provided in the form of GDML.
  - Radiation environment can be specified as an ascii input file.
  - ***Gorad*** is controlled by UI commands, and it works both in interactive mode (Qt recommended) and in batch mode with an input macro file.
  - It runs in multithreaded mode with automatic data reductions.
- ***Gorad*** utilizes some of new developments in Geant4 version 10.7/11.0 so that it does not work with earlier version of Geant4.
- ***Gorad*** is under further extension with a DOE contract for activation/cooling studies.

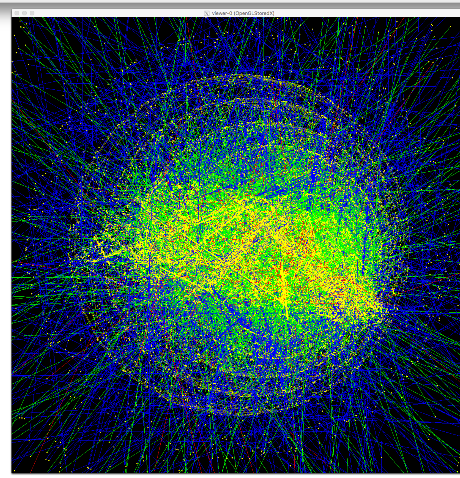
How *Gorad* works

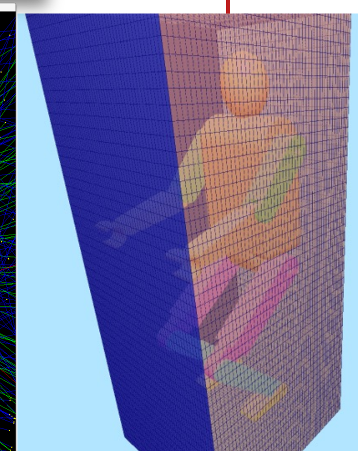
GDML

Radiation spectrum

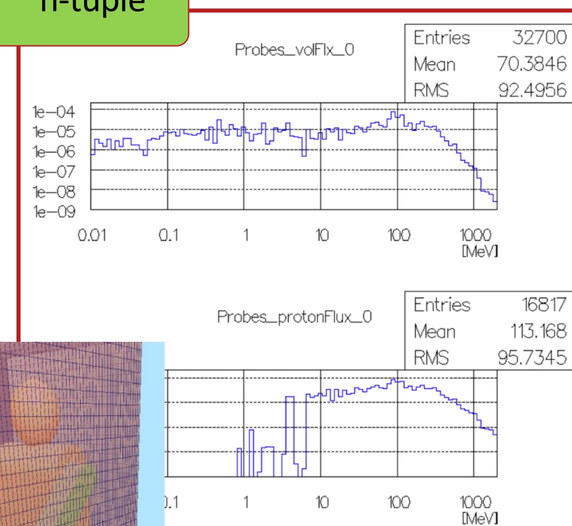
Macro file
or
interactive command





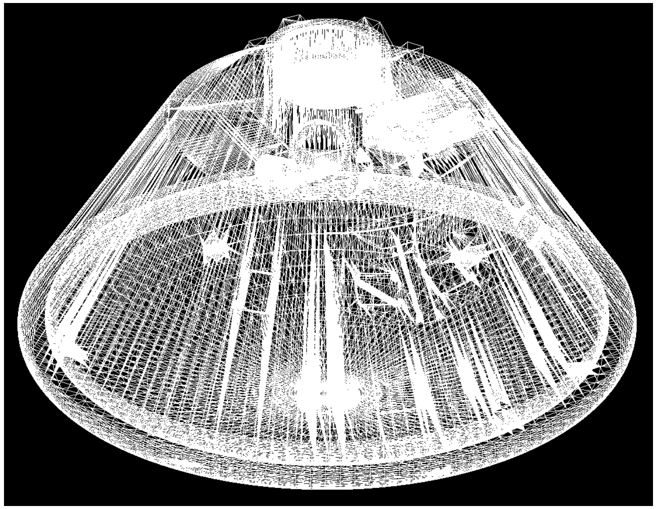


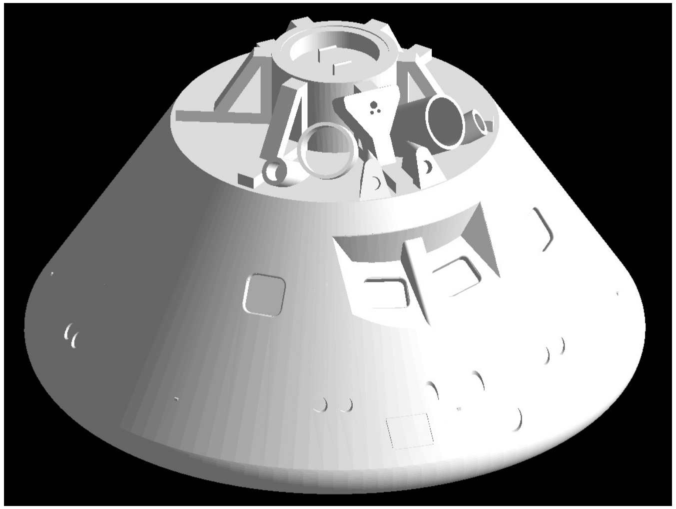
Scores
histos
n-tuple



Geometry

- Simulation geometry should be provided in the form of GDML.
- ***Gorad*** offers some utility commands.
  - Setting/changing material of a volume
  - List volumes with their properties such as mass
  - Define a region and set cuts for it
- As a sample geometry, partial and simplified shielding structure of Orion Spacecraft is available (~22 MB).

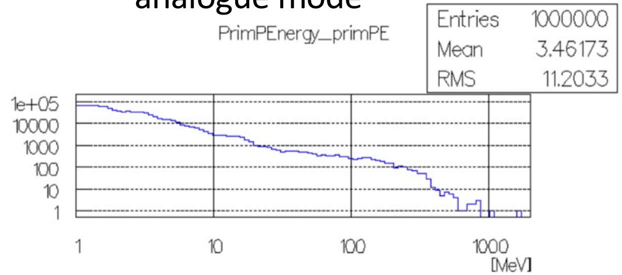




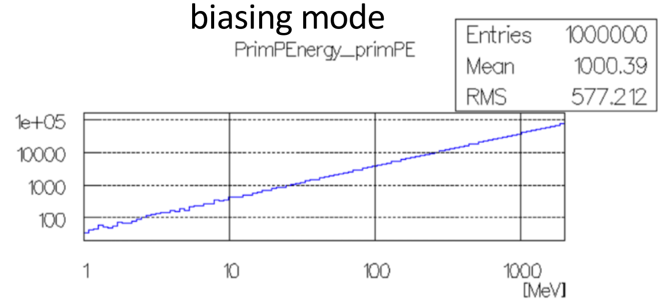
Primary radiation spectrum

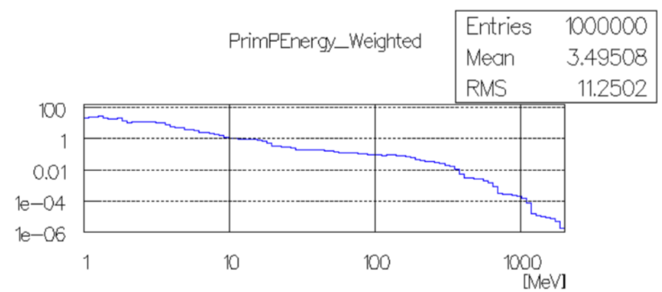
- ***Gorad*** provides two differential radiation spectra (King and trapped proton) as ascii input files.
  - User may use other spectra as well as Geant4 general particle source commands.
- Primary particle biasing option is available.

analogue mode



biasing mode



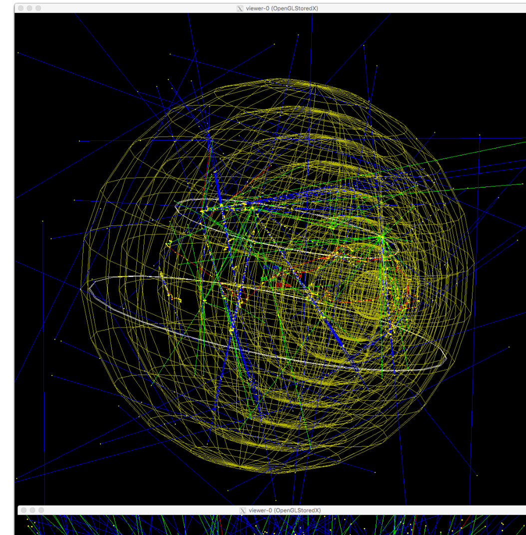


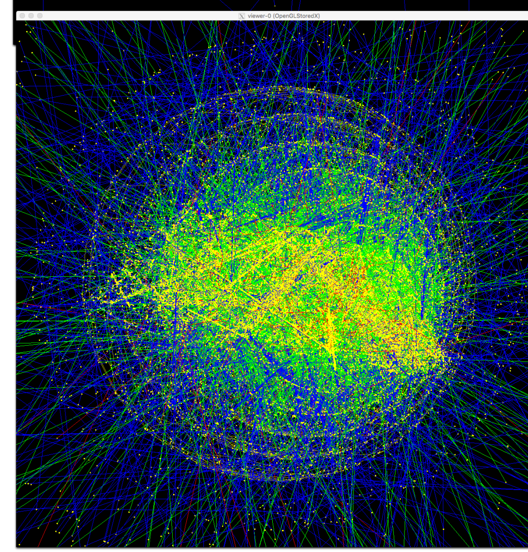
Please note that despite the trapped proton spectrum file having data up to 2 GeV, analog mode plot does not have many entries higher than 1 GeV due to insufficient statistics, i.e. 1 million events are not enough to have tracks higher than 1 GeV. And most of lower-energy tracks do not penetrate the spacecraft shield.

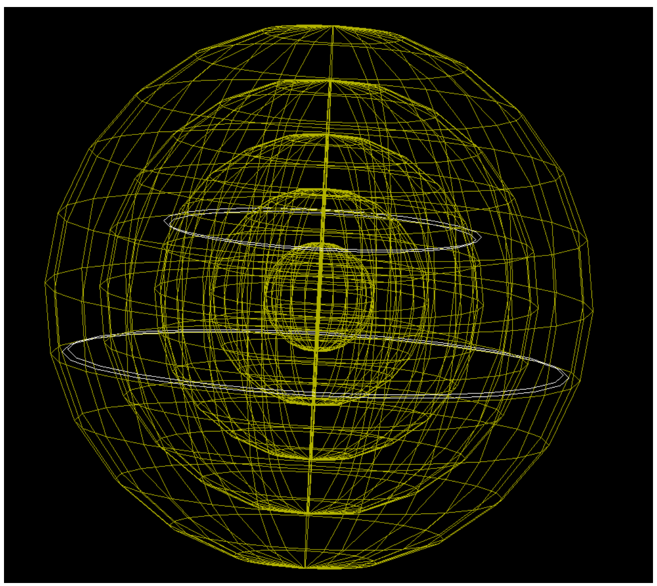
Here, the upper plot shows the energy spectrum of the generated primary tracks without their weights, while lower plot shows the same energy spectrum of the generated primary tracks with weights. This option can statistically enhance higher-energy tracks that are rare but contribute largely to the scores.

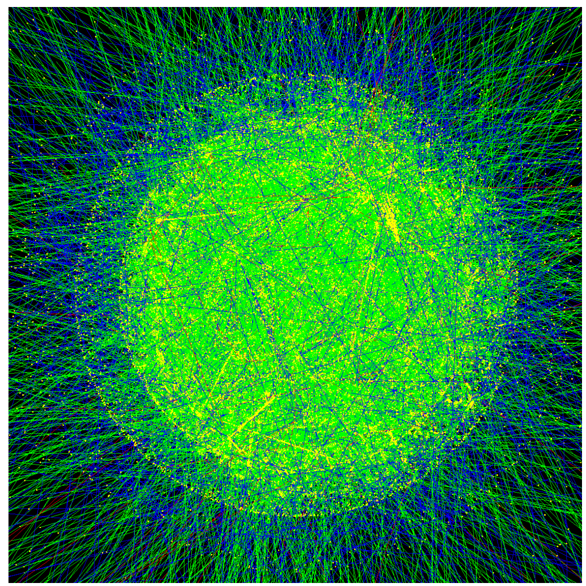
Geometry importance biasing

- ***Gorad*** offers UI commands to define geometry importance biasing.
  - Virtual spheres are created in a dedicated parallel world.
  - Each sphere has its importance.
  - Tracks are enhanced toward the center of the geometry.









A few comments with my Jefferson Lab hat

- We are aware of the boiling demand of Geant4 tutorial course in US.
  - The “most recent” tutorials in US were at MD Anderson in June 2018 (medical highlighted) and at MIT in November 2017.
  - We are planning to organize one at Jefferson Lab in Virginia early next year. Announcement will be sent out through the Geant4 announcement mailing list.
- Jefferson Lab is now transitioning to “multi-science laboratory”.
  - We would like to engage in space-related projects.
  - Please let us know if there is any collaborative project we can involve.