



COMMENTS ON GEANT4 INITIALISATION

*V. IVANTCHENKO, CERN & TOMSK STATE UNIVERSITY**

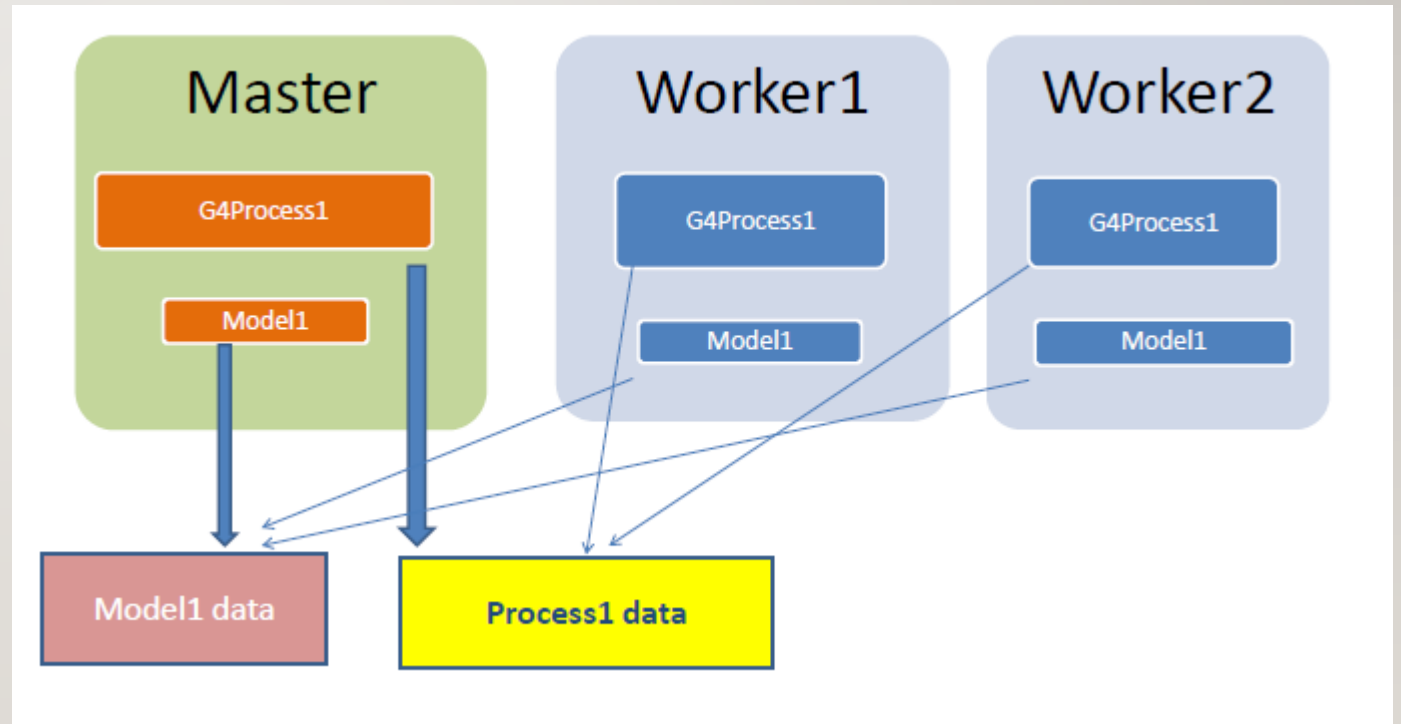
29TH GEANT4 COLLABORATION MEETING

CATANIA, ITALY, 7–11 OCT 2024

* grant of Government (Agreement No. 075-15-2024-667)

EM DATA SHARING FOR GEANT4 MT

- **Shared EM physics data:**
 - tables for cross sections, stopping powers and ranges are kept by processes
 - Data on EM cross section shape
 - Element Selectors
 - Differential cross section data are kept by models
- **Currently tables are filled in master thread**
 - Read only in run time



EM DATA HANDLING IN 11.3

❖ Existing **G4ElementDataRegistry**

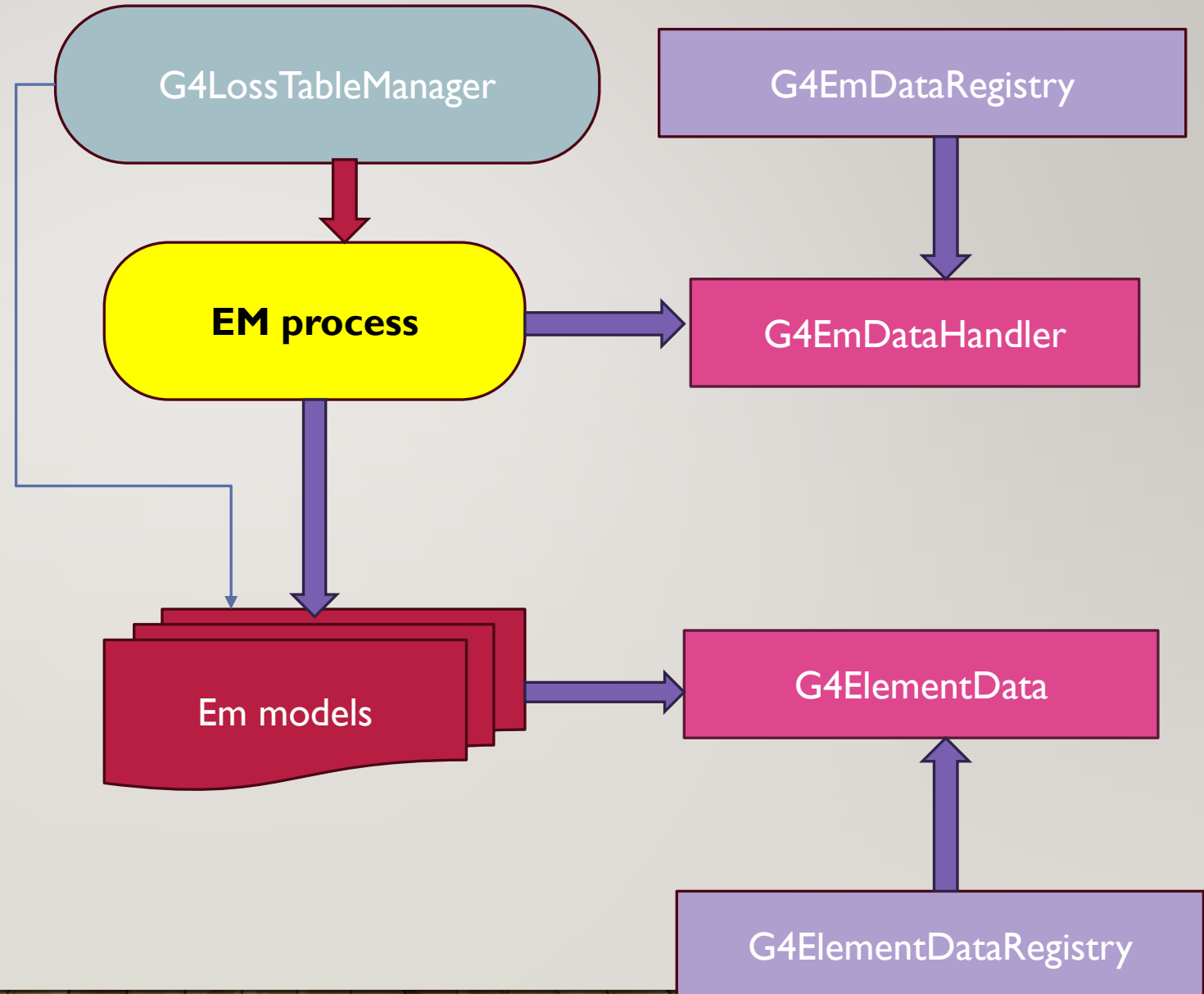
- ❖ Keep G4ElementData for EM models

❖ **G4LossTableManager** define master thread in constructor

- ❖ It may be the first working thread

❖ New class **G4EmDataRegistry**

- ❖ singleton to keep shared data from EM processes
 - ❖ Physics tables
 - ❖ EM cross section shape data
- ❖ this class is responsible for deletion of G4EmDataHandlers



PARALLEL INITIALIZATION PER PARTICLE

- EM and hadronic cannot be considered independent
 - Initialization is performed per particle, not per domain of physics
- Independent initialization of may be done for pair of particles
 - gamma
 - e^+ and e^-
 - μ^+ and μ^-
 - π^+ and π^-
 - K^+ and K^-
 - p and \bar{p}
 - Other particles
- The reason: sharing of tables and processes between particles

PARALLEL INITIALIZATION OF PHYSICS

- Initialization of physics is performed by two calls
 - void `PreparePhysicsTable`(const G4ParticleDefinition&) // initialization flags and pointers
 - void `BuildPhysicsTable`(const G4ParticleDefinition&) // computations and filling
- For parallel initialization it is possible to extend:
 - void `PreparePhysicsTable`(const G4ParticleDefinition&, G4bool isMaster)
 - void `BuildPhysicsTable`(const G4ParticleDefinition&, G4bool isMaster)
 - This extension will require minimal change in process classes
 - Flag name may be different “isMaster”, “isNiltizer”, “fillTables”
 - Why is it needed to set flag by the kernel?
 - Because it is a clear direct message, alternative would be error prong complicate logic within the process class
- Initialization of hadronics uses the same approach as EM
 - The main difference: more reading of data from datasets in the hadronic case

EXTRA COMMENTS

- CMS resource request should be done well in advance (2 year before)
 - So far it is a request for 4-thread queue
 - In future it is foreseen 8-thread queue
- When we speak of parallel initialization, we may consider very limited number of threads
 - Enabling or disabling parallel initiation should change nothing in results