

# Vectorization of EM physics

---

Marilena Bandieramonte - GeantV by-weekly meeting

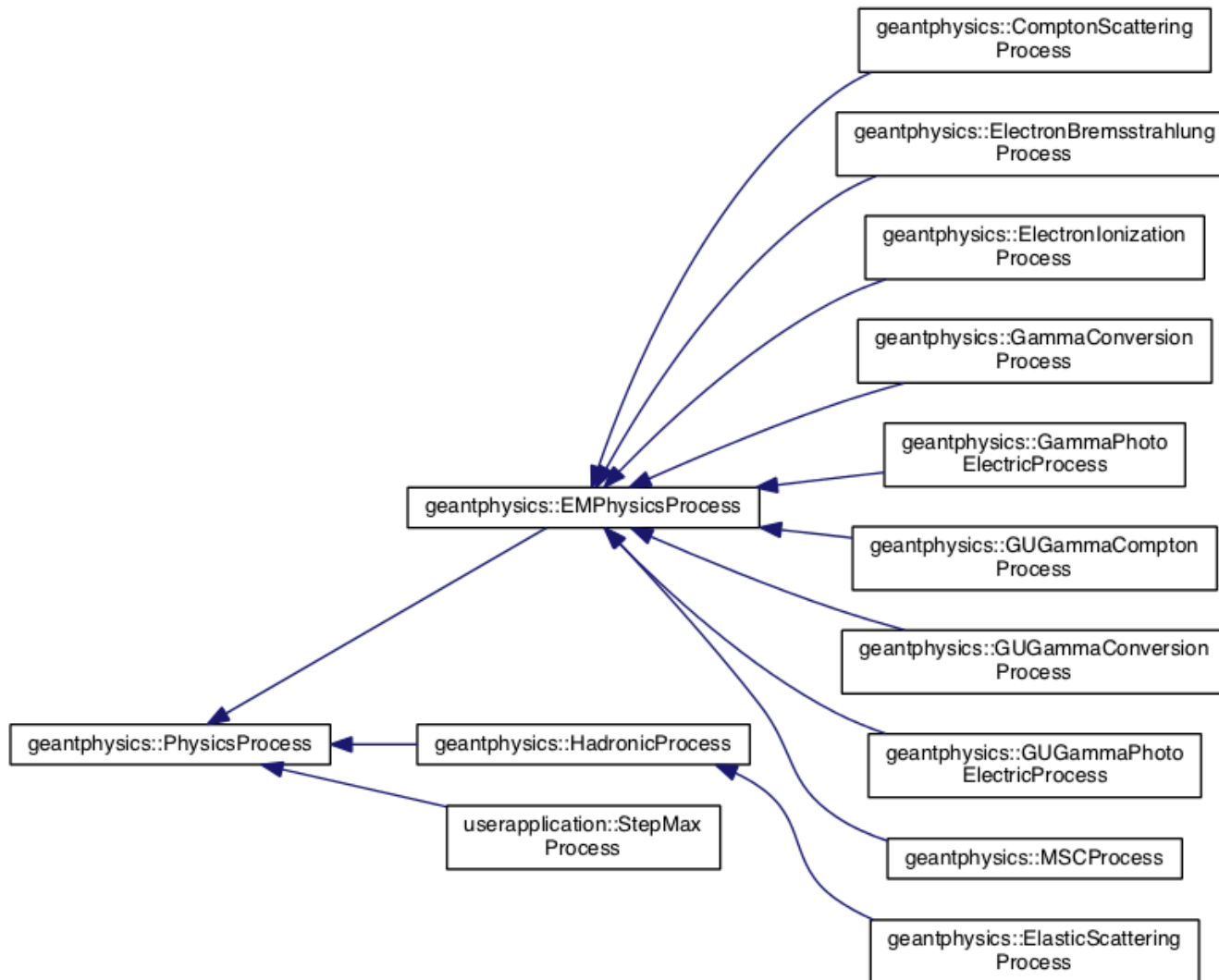
# EM real physics models - Status

---

Primary	Process	Model
e- e+	Bremsstrahlung	Seltzer-Berger Relativistic Brems
	Ionization	Moller-Bahba
	Multiple Scattering	Goudsmit-Saunderson
	Photoelectric Eff.	Sauter-Gavrila
gamma	Compton	KleinNishina
	Pair Production	Bethe-Heitler Relativistic PP

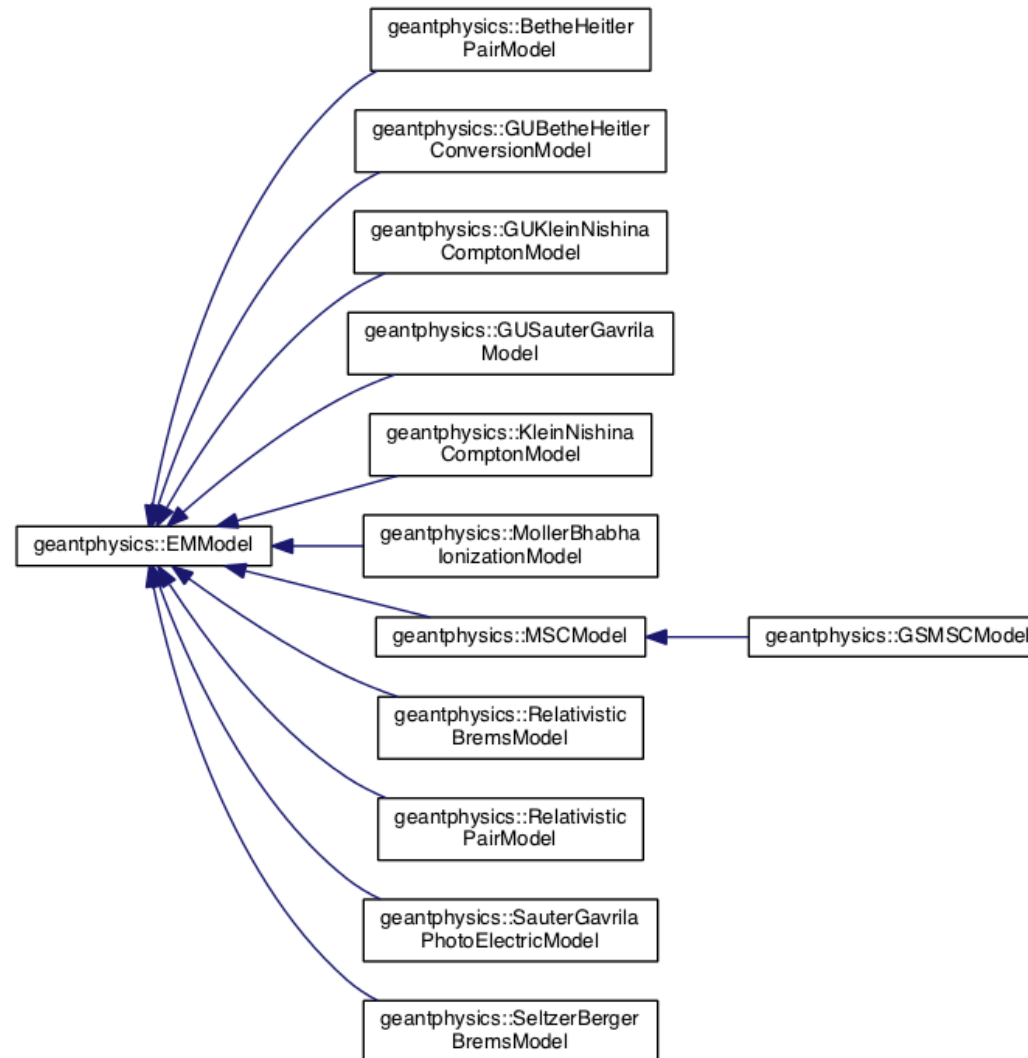
# RealPhysics Class Hierarchy: EMPhysicsProcess

---



# RealPhysics Class Hierarchy: EMModels

---



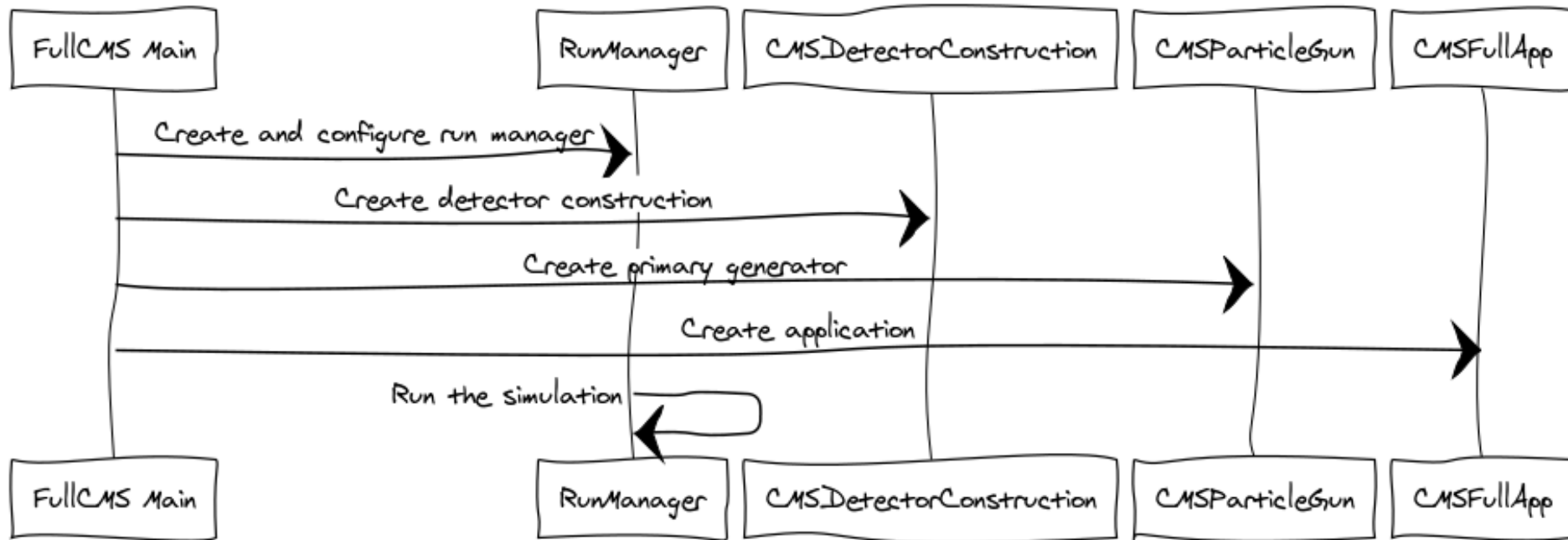
# Towards vectorisation

---

- All the models are developed in a “**vectorization-friendly**” way:
  - Implemented with both alias and rejection sampling algorithm options
- But there are still plenty of **conditional branches** (the use has to be minimised - use of Mask operation when not possible)
- **Memory contiguity** issues:
  - **Structure of arrays** (SOA) have to be used for track data (Geant::GeantTrack vs LightTracks?)
  - How to handle the **access** to Alias Tables data? (Gather and Scatter operations required?)
- **Inlined functions** should be used when possible

# Full CMS app Sequence Diagram

---



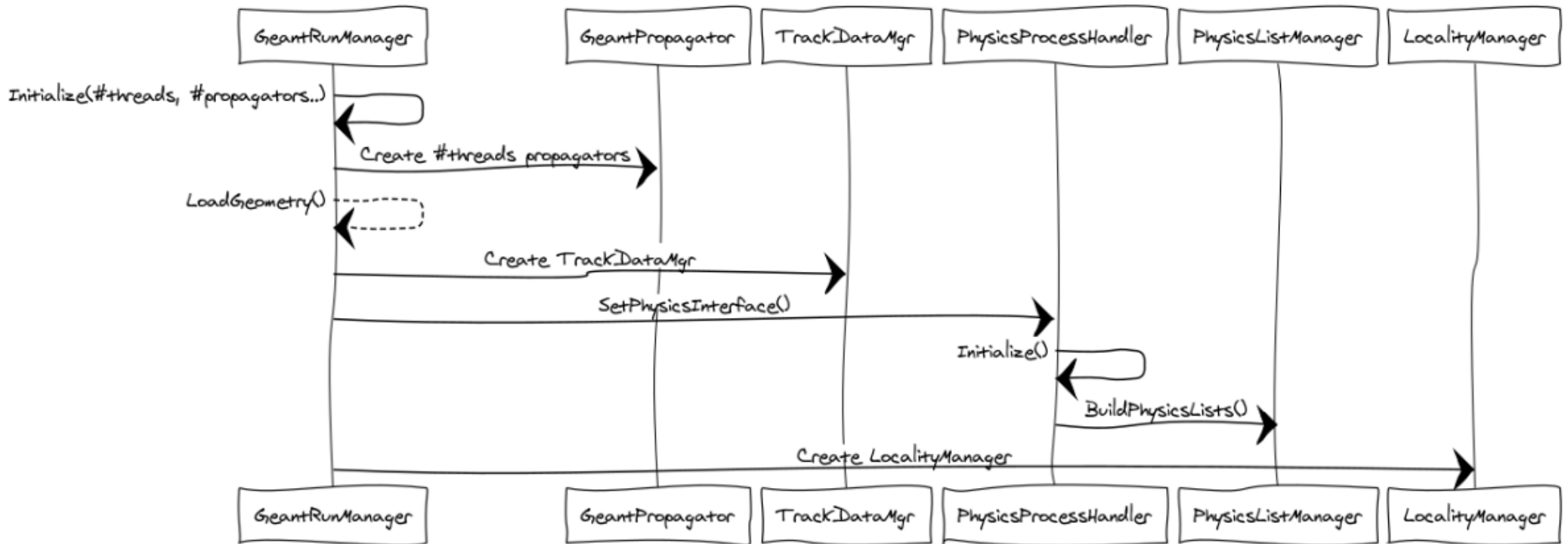
# GeantRunManager

---

```
bool GeantRunManager::Initialize() {
    [...]

#ifdef USE_REAL_PHYSICS
    if (!fPhysicsInterface) {
        Geant::Fatal("GeantRunManager::Initialize", "The physics process interface has to be initialized before
this");
        return false;
    }
    // Initialize the physics
    fPhysicsInterface->Initialize();
#else
    if (!fProcess) {
        Geant::Fatal("GeantRunManager::Initialize", "The physics process has to be initialized before this");
        return false;
    }
#endif
    [...]
}
```

# GeantRunManager::Initialize() Sequence Diagram





# WorkloadManager

---

```
void *WorkloadManager::TransportTracksV3(GeantPropagator *prop) {
[...]
```

```
    #ifdef USE_REAL_PHYSICS
        propagator->GetPhysicsInterface()->AlongStepAction(mat, output.GetNtracks(), output, nextra_at_rest,
        td);
    #else
        propagator->Process()->Eloss(mat, output.GetNtracks(), output, nextra_at_rest, td);
    #endif

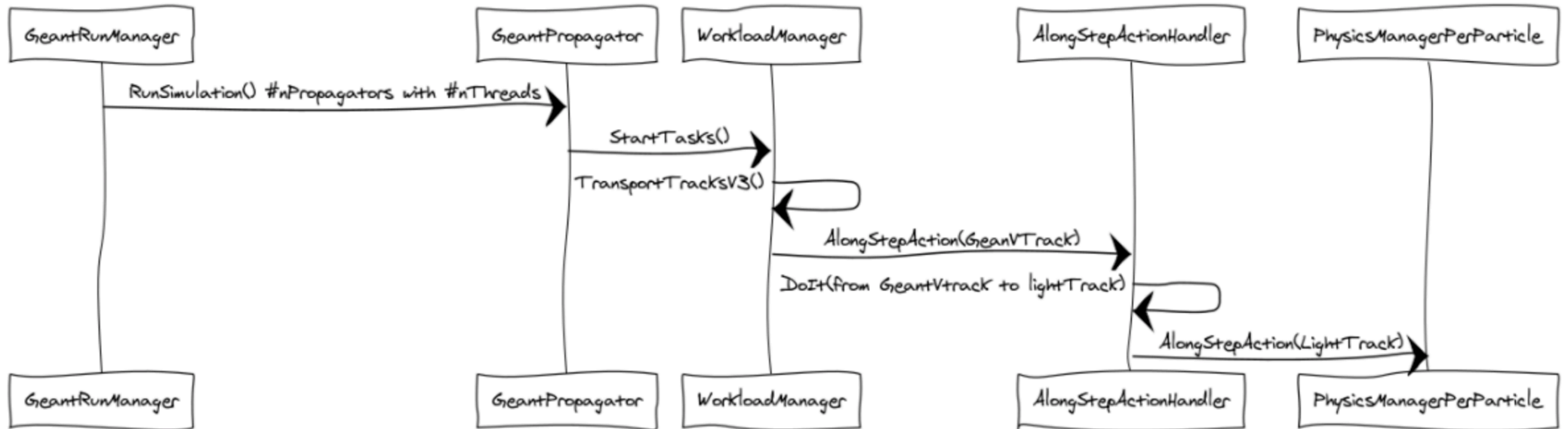
[...]
```

```
    #ifdef USE_REAL_PHYSICS
        propagator->GetPhysicsInterface()->PostStepAction(mat, nphys, output, ntotnext, td);
    #else
        // first: sample target and type of interaction for each primary tracks
        propagator->Process()->PostStepTypeOfIntrActSampling(mat, nphys, output, td);
    #endif

[...]
```

```
}
```

# GeantRunManager::RunSimulation()



# GeantPropagator

---

```
void GeantPropagator::ProposeStep(int ntracks, GeantTrack_v &tracks, GeantTaskData *td) {  
    [...]  
#ifdef USE_REAL_PHYSICS  
    fPhysicsInterface->ComputeIntLen(mat, ntracks, tracks, 0, td);  
#else  
    fProcess->ComputeIntLen(mat, ntracks, tracks, td);  
#endif  
}  
  
#ifdef USE_REAL_PHYSICS  
int GeantPropagator::CreateSimulationStages()  
{  
    [...]  
}  
#else  
VECCORE_ATT_HOST_DEVICE  
int GeantPropagator::CreateSimulationStages()  
{  
    [...]  
}  
#endif
```

# FieldPropagationHandler

---

```
VECCORE_ATT_HOST_DEVICE
void FieldPropagationHandler::DoIt(GeantTrack *track, Basket& output, GeantTaskData *td)
{
    [...]
#ifdef USE_REAL_PHYSICS
    track->SetStage(kPostStepActionStage);
#else
    track->SetStage(kContinuousProcStage);
#endif
    [...]
}
```

```
VECCORE_ATT_HOST_DEVICE
void FieldPropagationHandler::DoIt(Basket &input, Basket& output, GeantTaskData *td)
{
    [...]
#ifdef USE_REAL_PHYSICS
    //      track->SetStage(kLongStepActionStage);
    track->SetStage(kPostStepActionStage);
#else
    track->SetStage(kContinuousProcStage);
#endif
    [...]
}
```

# Roadmap and People

---

- **Expected time:** May-June 2018?
- **People** interested in working on EM physics Vectorization:
  - Marilena
  - Vitali (starting from Feb 2018)
  - ?
- Present EM vectorisation **results @CHEP2018** - Abstract due by 31st Dec