Overview

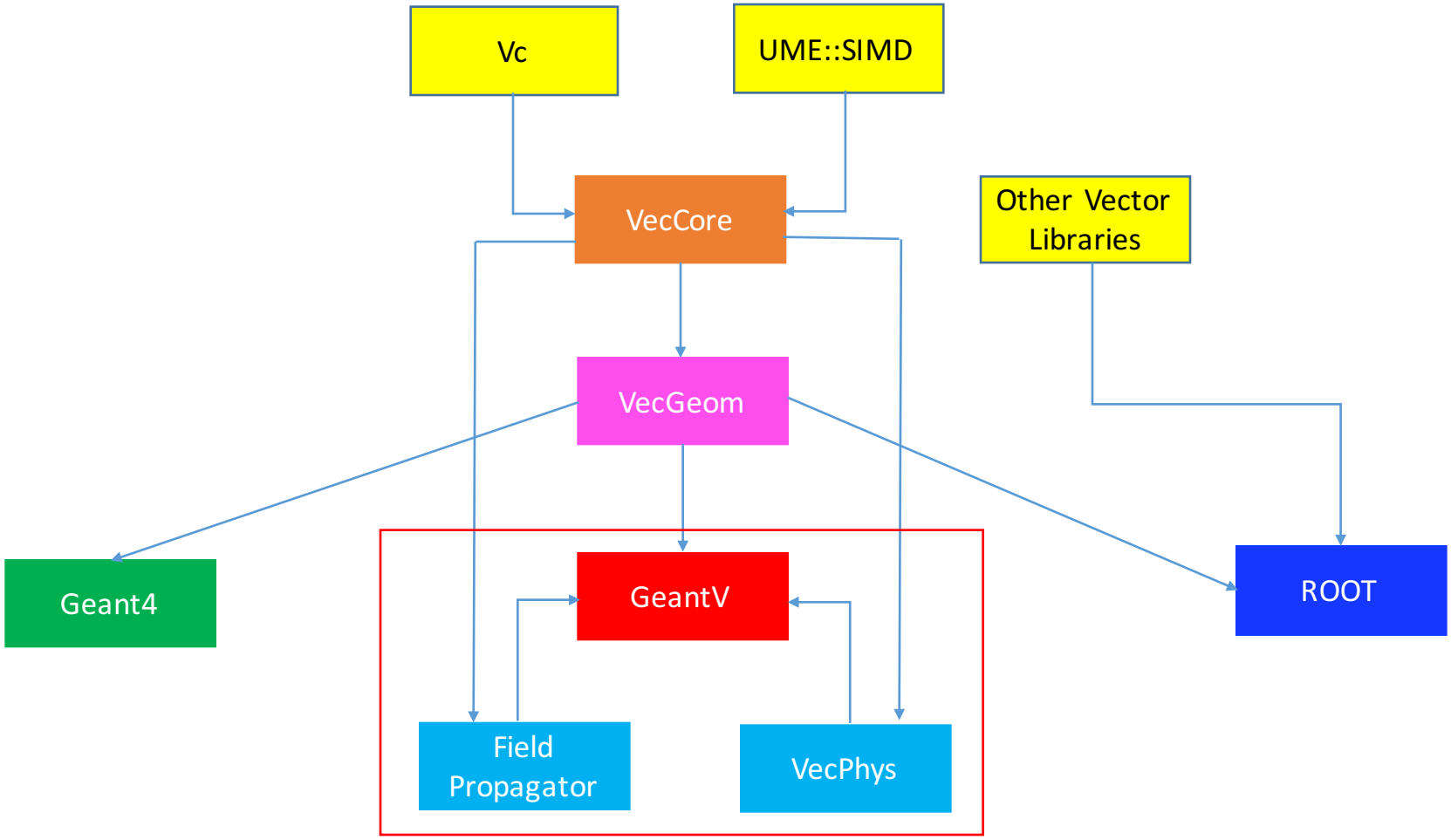
VecMath Meeting

June 1, 2017

JIRA: GEANT-373

- Create VecMath
- Description
  - VecMath is (will be) an independent package containing generic codes (i.e., supporting SIMD/SIMT) of
    1) math (backend/kernel) functions
    2) random number generators (services) and statistical functions
    3) physics utility functions (such as Lorentz 4-vector)
  - Create(d) this JIRA as it is (was) omitted in the current POW. Related work plan and milestones should be discussed among developers who are interested in both (either) consolidating existing codes or developing new implementation.
- Provide a common, consistent and sustainable HEP vector libraries for SIMD and SIMT (subject to the primary use of GeantV sub-packages)

Current Package Hierarchy and Dependencies



Core/redundant components

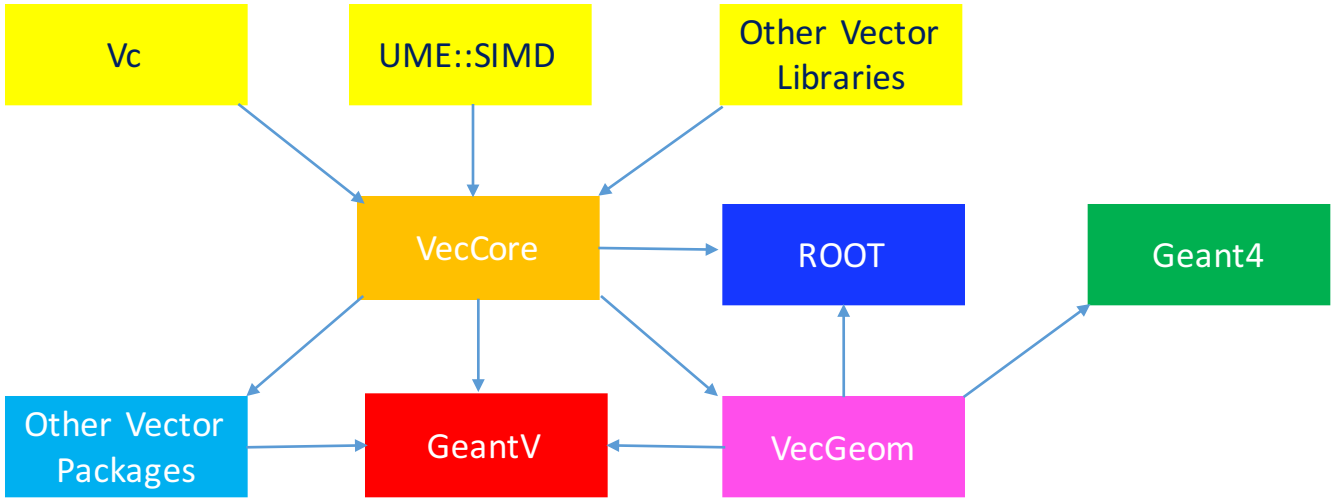
- VecCore
  - Backend
  - VecMath.h
  - Utilities.h, Types.h, …
- VecGeom/backend (obsoleted)
- VecGeom/base
  - Math.h
  - Array/Vector classes, SOA3D, …
  - RNG.h
- Geant/base
  - Math.h, Typedef.h
  - SystemOfUnits, PhysicalConstants
- Geant/vecprot_v2
  - GeantTaskData->fRndm (vecgeom::RNG or Trandom)
- Geant/vecphys
  - ThreeVector
  - rng (VecRNG)
- Geant/magneticfield
  - ThreeVector (vecgeom::Vector3D)
  - Units, Constants

Consolidation and Future extension

- Common Math.h
  - Backend functions/kernels
- Array/Vector classes
  - Lorentz vector and other physics related methods for SIMD/SIMT
  - Cuda extension (data types and hardware specific functions)
- Common random number services and probability distributions
- Other headers?
  - System of units, Physical constants

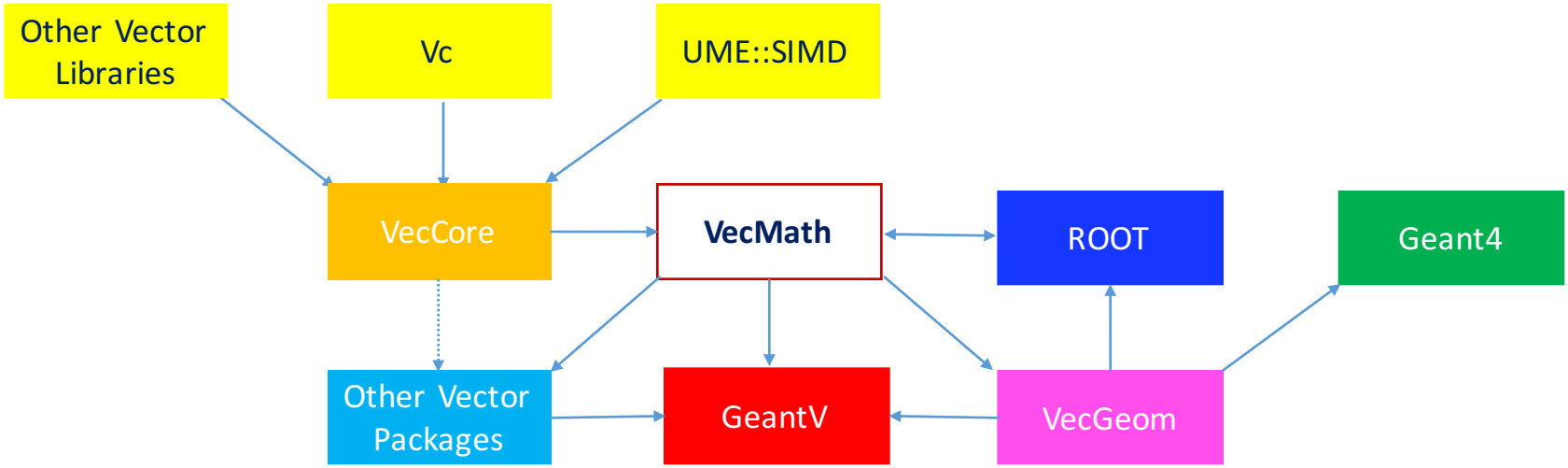
Option-1: Extend VecCore without VecMath

- VecCore
  - Interfaces to all external vector libraries
  - Math (backend/kernel) functions
  - Move array/vector/container classes/methods from VecGeom and extend
  - Add random number services and vector statistical methods
  - Other vector utility classes



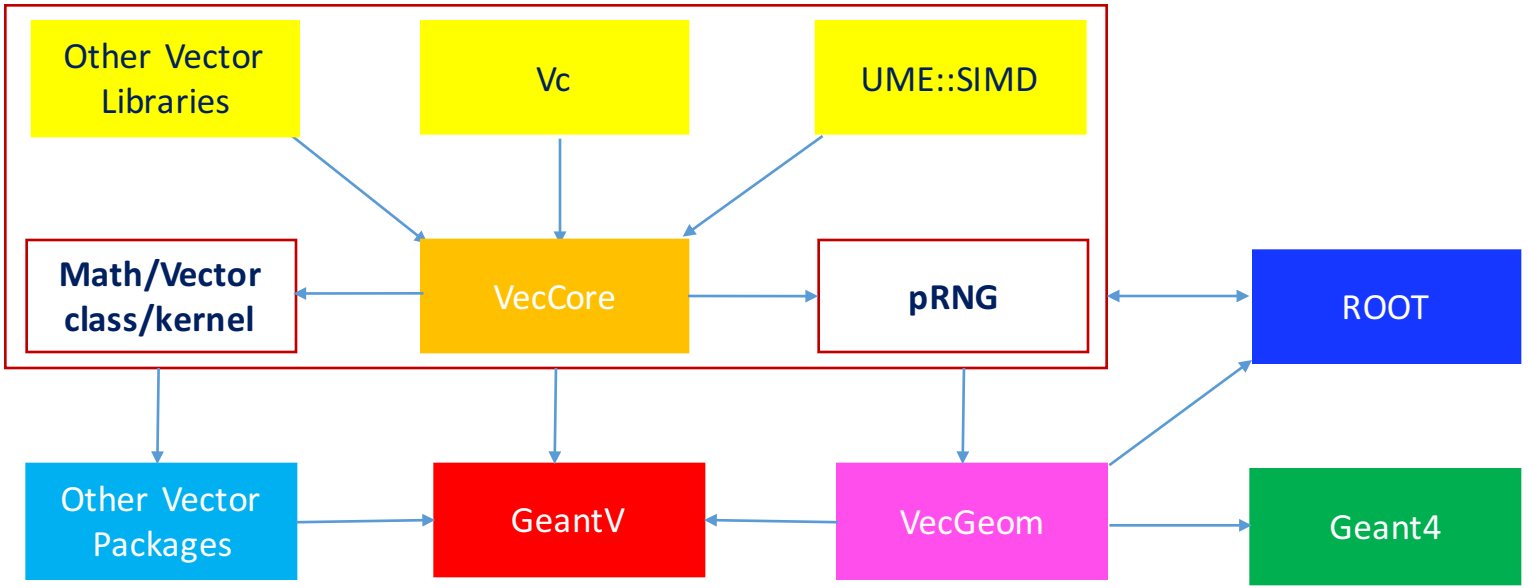
Option-2: Add VecMath

- VecMath : an independent package
  - Depends on VecCore (backend and math backend)
  - math headers and generic functions
  - Collect array/vector/container classes/methods for SIMD/SIMT
  - Random number services and vector statistical methods
  - Other vector utility classes



Option-3: VecMath ∈ VecCore ⊕ External VectorLibs

- VecMath : an integrated package
  - Option-2, but the tighter packaging with VecCore and external vector libraries
  - Something similar to Option-1, but VecCore is an sub-component of VecMath (yet may serve as an standalone package)
  - Simplification of support hierarchy



Open Discussions

- VecCore (Option1) vs. VecMath (Option2/3)
- Other options or nothing?
- Required tasks
  - Consolidating existing components
  - Extension for missing pieces
  - Testing and code management
  - Longer-term support
- Where do we go from here?