Hadronic Physics I

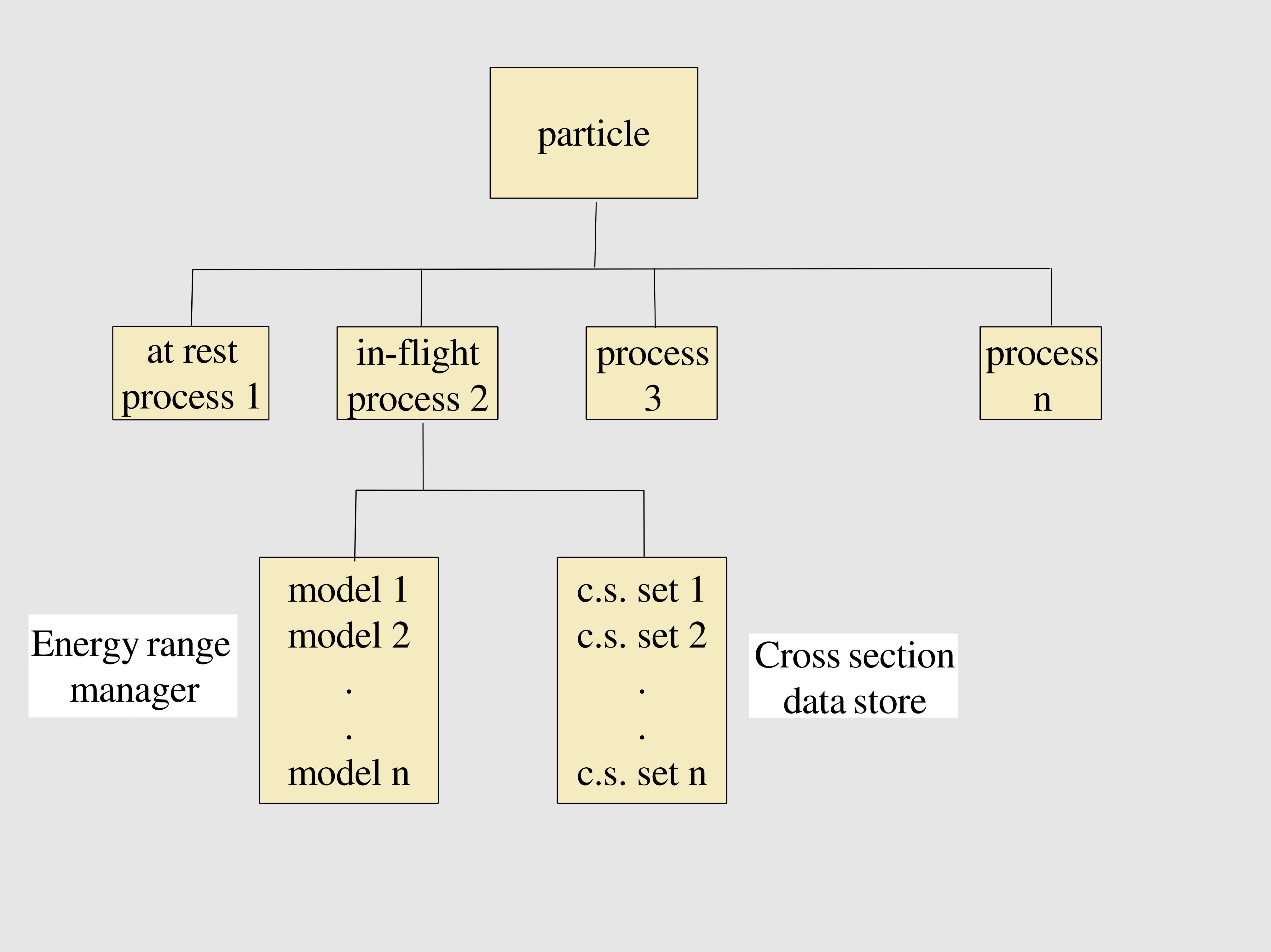
Geant4 Tutorial at McGill University
27 September 2006
Dennis Wright

Outline

- Overview of hadronic physics
  - processes, cross sections, models
  - hadronic framework and organization
- The cascade models
  - Bertini, binary
- Parameterized models
  - high energy, low energy

Hadronic Processes, Models and Cross Sections

- In Geant4 physics is assigned to a particle through processes
- Each process may be implemented
  - directly as part of the process, or
  - in terms of a model class
- In Geant4 hadronic physics there are sometimes many models for a given process
  - user must choose
  - can have more than one per process
- A process must also have cross sections assigned
  - here too, there are options


particle
at rest process 1
in-flight process 2
process 3
process n
model 1
model 2
.
.
model n
c.s. set 1
c.s. set 2
.
.
c.s. set n
Energy range manager
Cross section data store

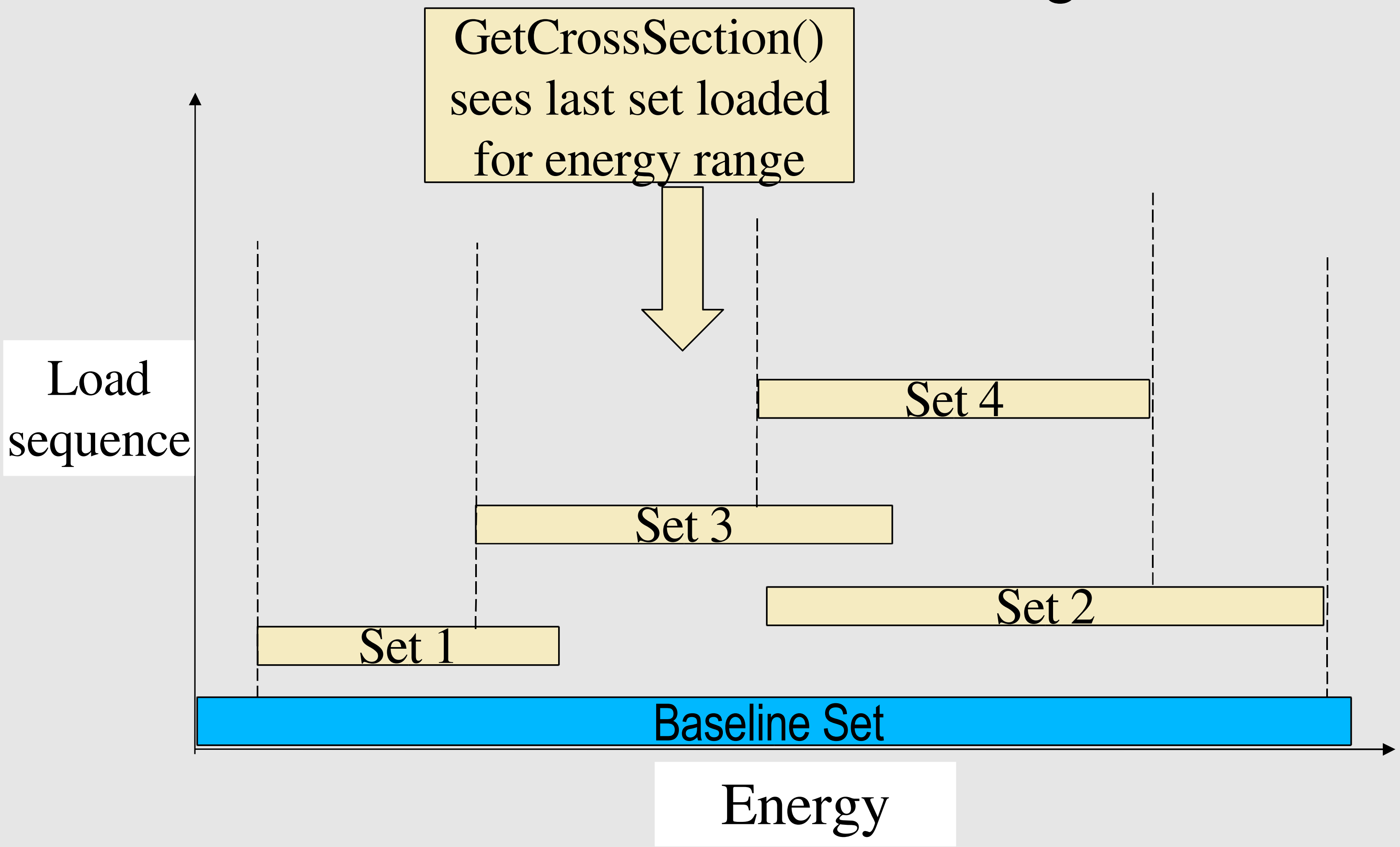
Cross Sections

- Default cross section sets are provided for each type of hadronic process
  - fission, capture, elastic, inelastic
  - can be overridden or completely replaced
- Different types of cross section sets
  - some contain only a few numbers to parameterize c.s.
  - some represent large databases
  - some are purely theoretical

Alternative Cross Sections

- Low energy neutrons
  - G4NDL available as Geant4 distribution data files
  - Available with or without thermal cross sections
- "High energy" neutron and proton reaction σ
  - 14 MeV < E < 20 GeV
- Ion-nucleus reaction cross sections
  - Good for E/A < 10 GeV
- Pion reaction cross sections

Cross Section Management



Hadronic Models – Data Driven

- Characterized by lots of data
  - cross section
  - angular distribution
  - multiplicity
  - etc.
- To get interaction length and final state, models interpolate data
  - cross section, coef of Legendre polynomials
- Examples
  - neutrons (E < 20 MeV)
  - coherent elastic scattering (pp, np, nn)
  - Radioactive decay

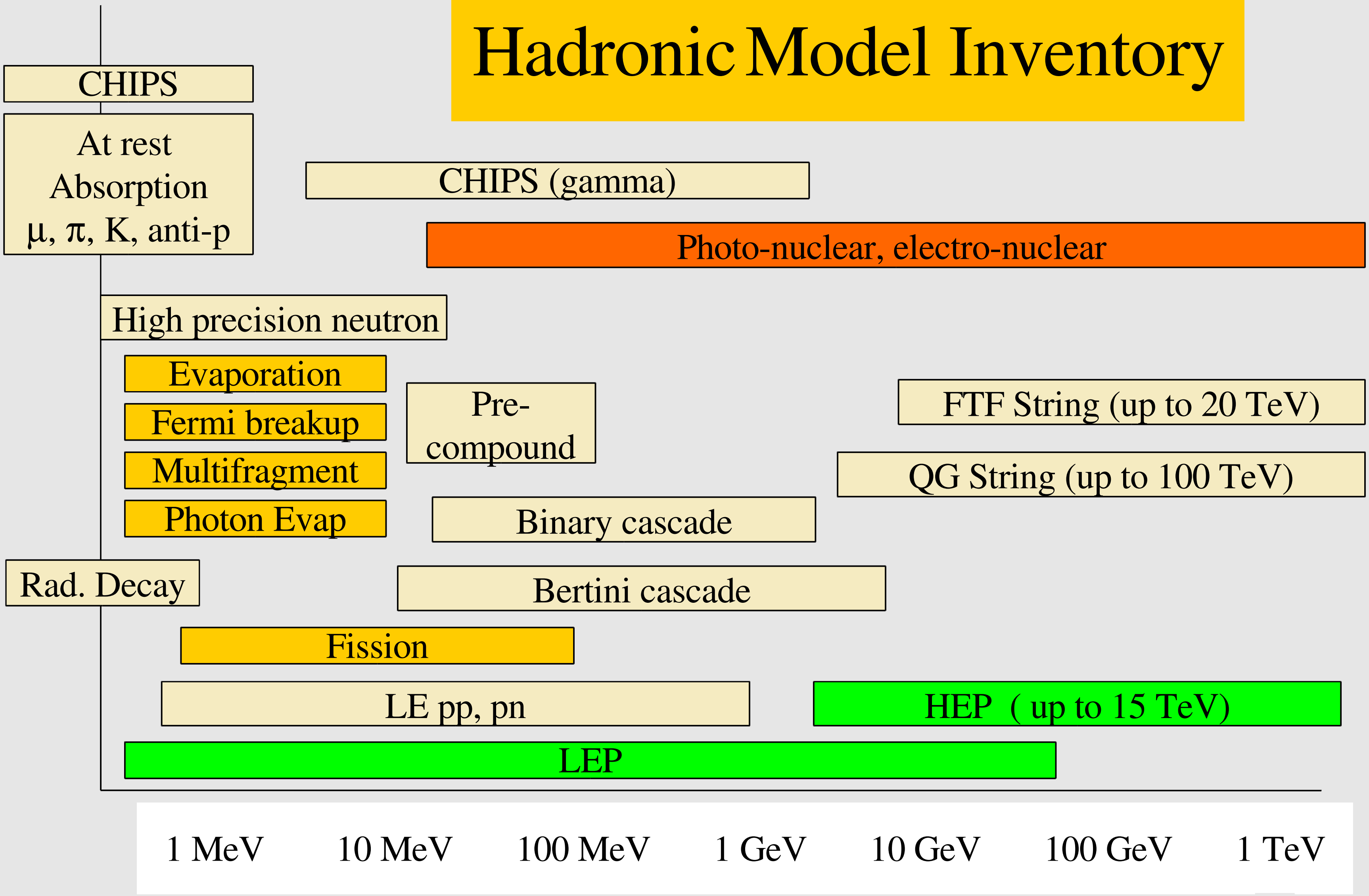
Hadronic Models – Theory Driven

- Dominated by theory (quark-gluon strings, chiral perturbation theory, ...)
  - not as much data to tie things down
- Final states determined by sampling theoretical distributions
- Examples:
  - quark-gluon string (projectiles with E > 20 GeV)
  - intra-nuclear cascade (intermediate energies)
  - nuclear de-excitation and breakup
  - chiral invariant phase space (up to a few GeV)

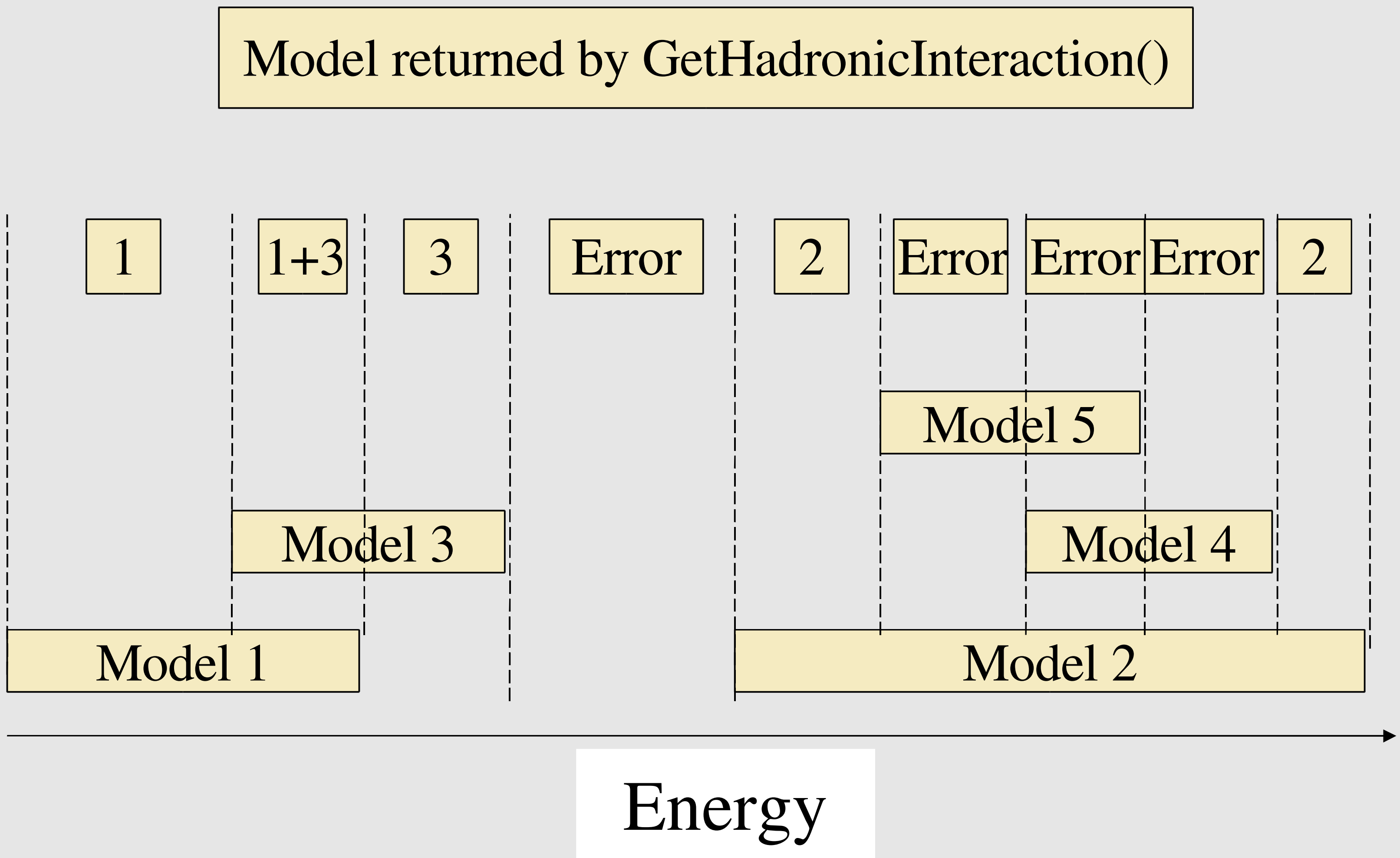
Hadronic Models - Parameterized

- Depend mostly on fits to data and some theoretical distributions
- Two models available:
  - Low Energy Parameterized (LEP) for < 20 GeV
  - High Energy Parameterized (HEP) for > 20 GeV
  - Each type refers to a collection of models
- Both derived from GHEISHA model used in Geant3
- Core code:
  - hadron fragmentation
  - cluster formation and fragmentation
  - nuclear de-excitation

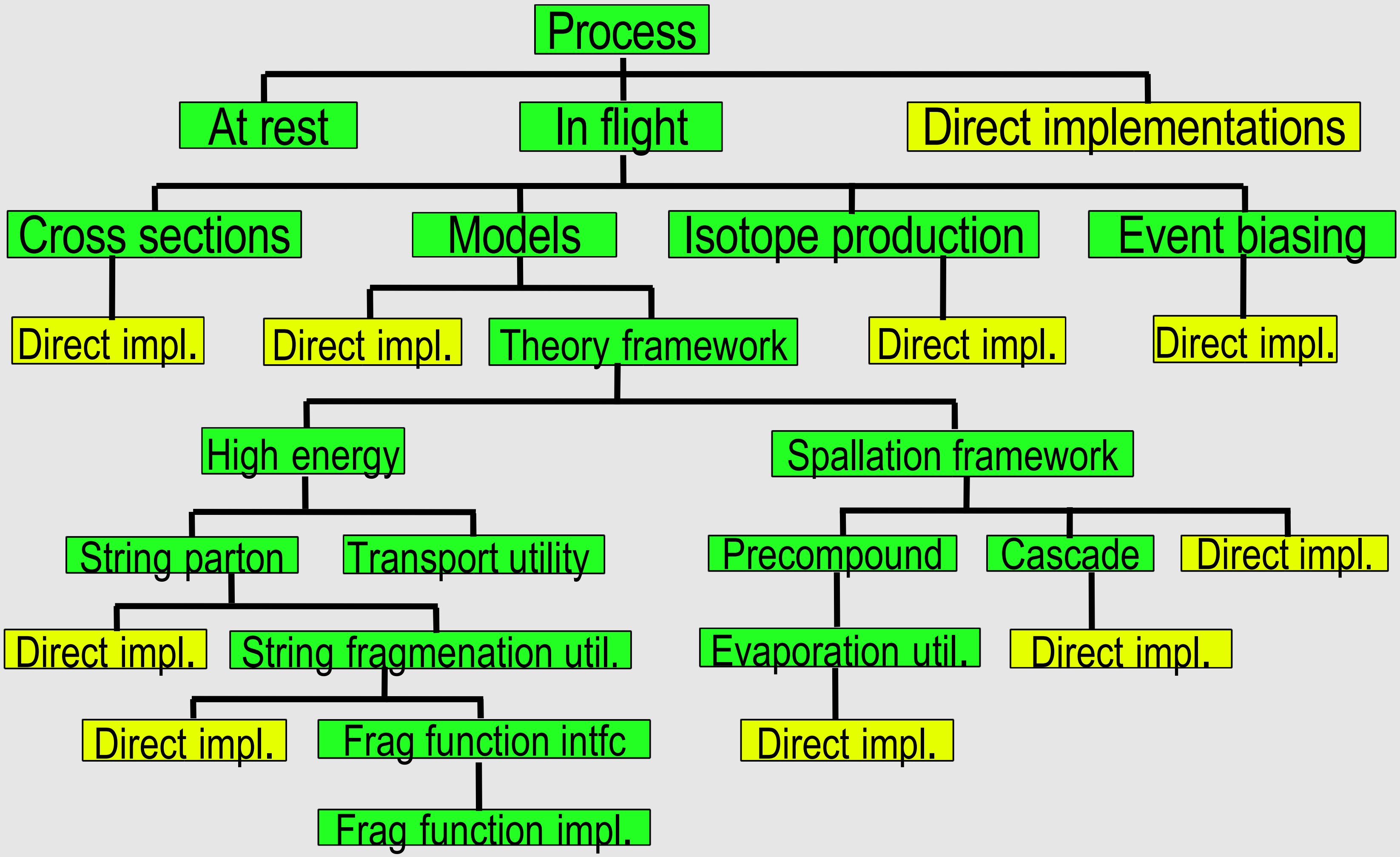
Hadronic Model Inventory



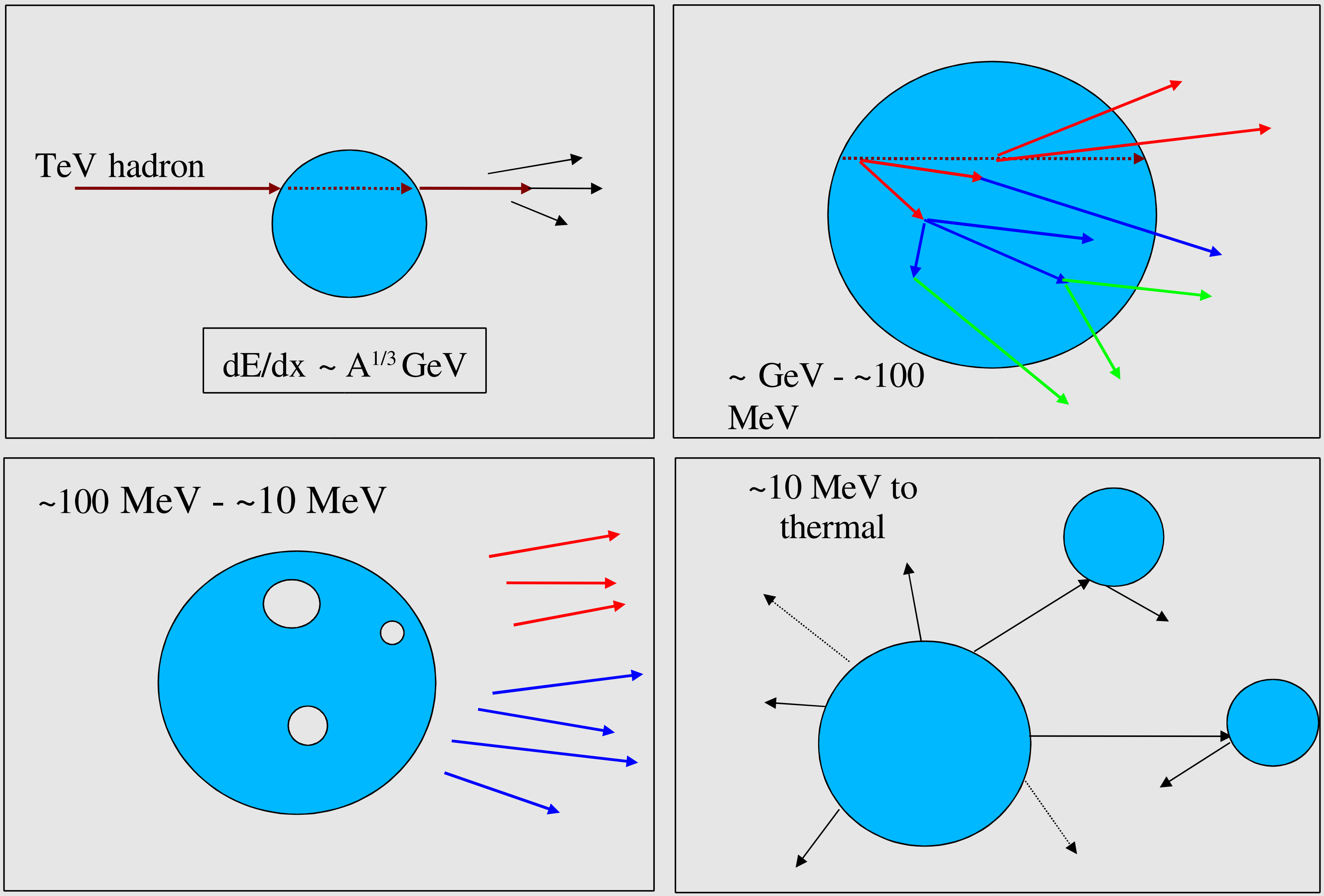
Model Management



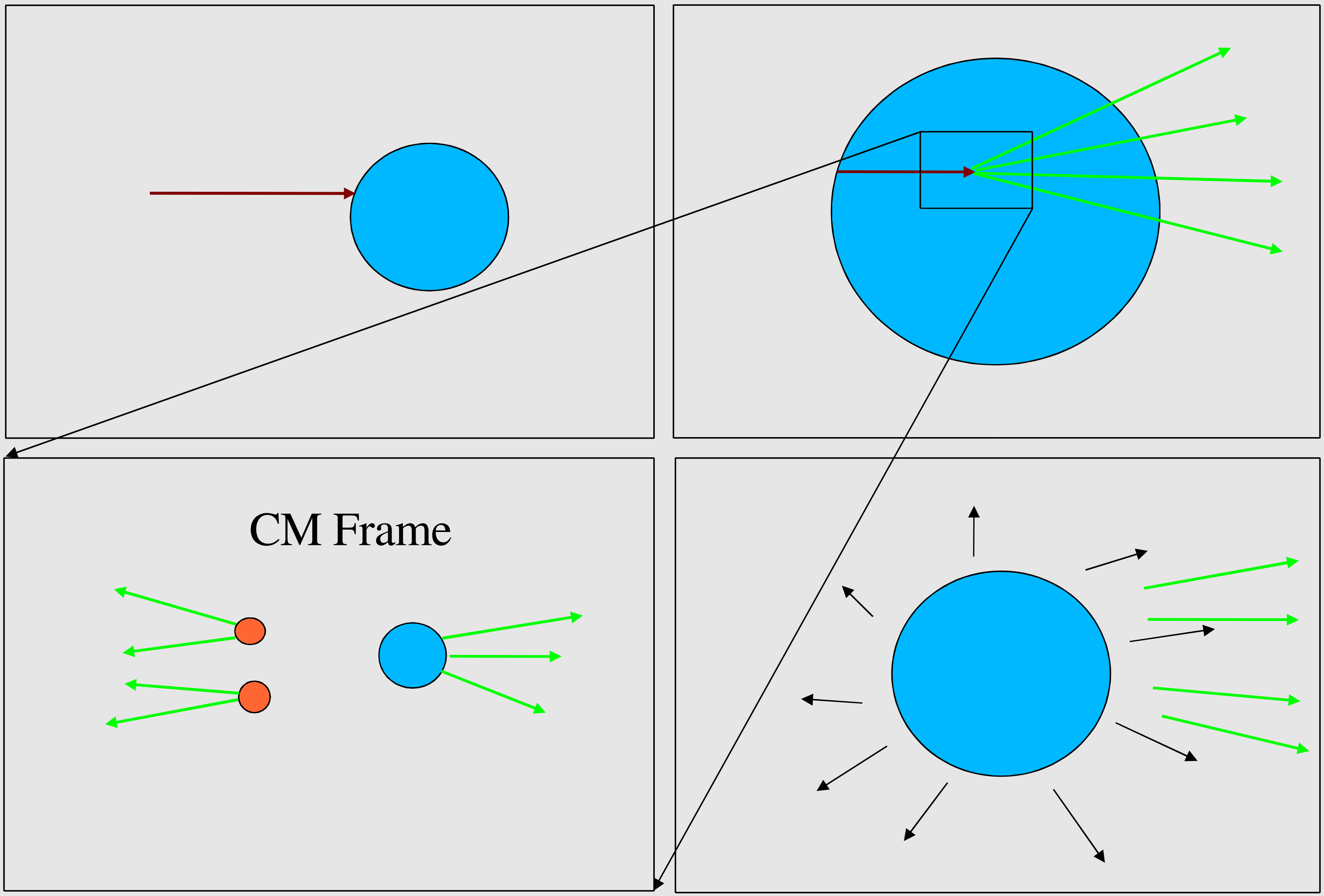
Hadronic Model Organization



Hadronic Interactions from TeV to meV



LEP, HEP (Comic Book Version)



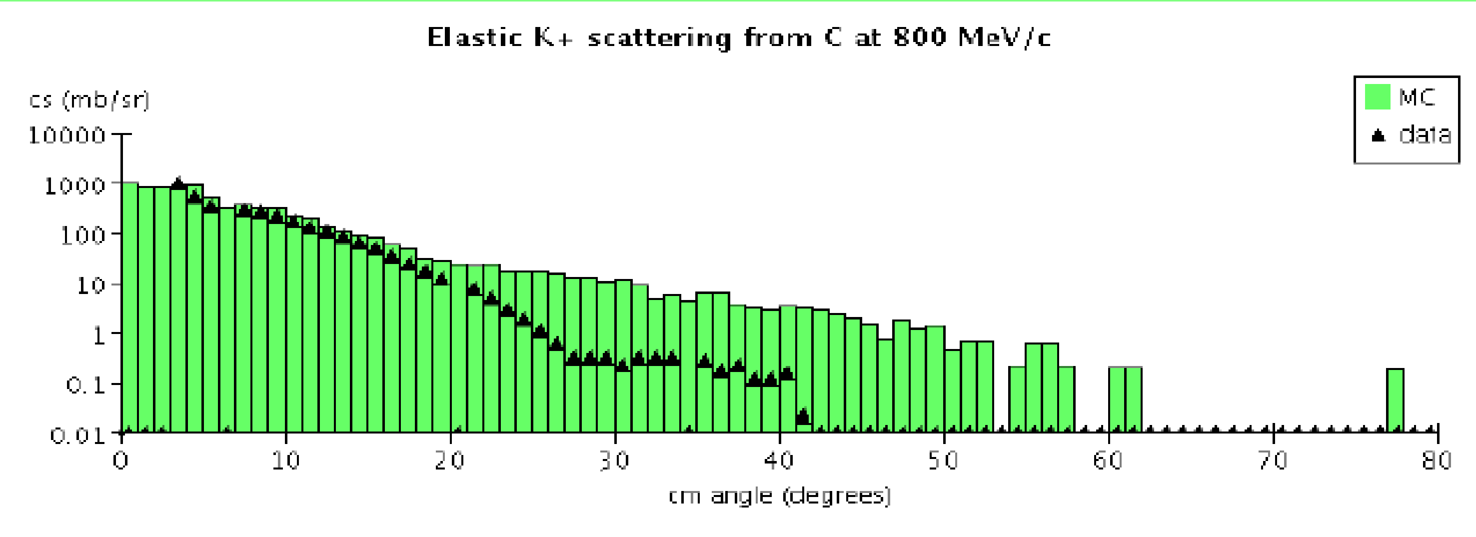
LEP, HEP models (text version)

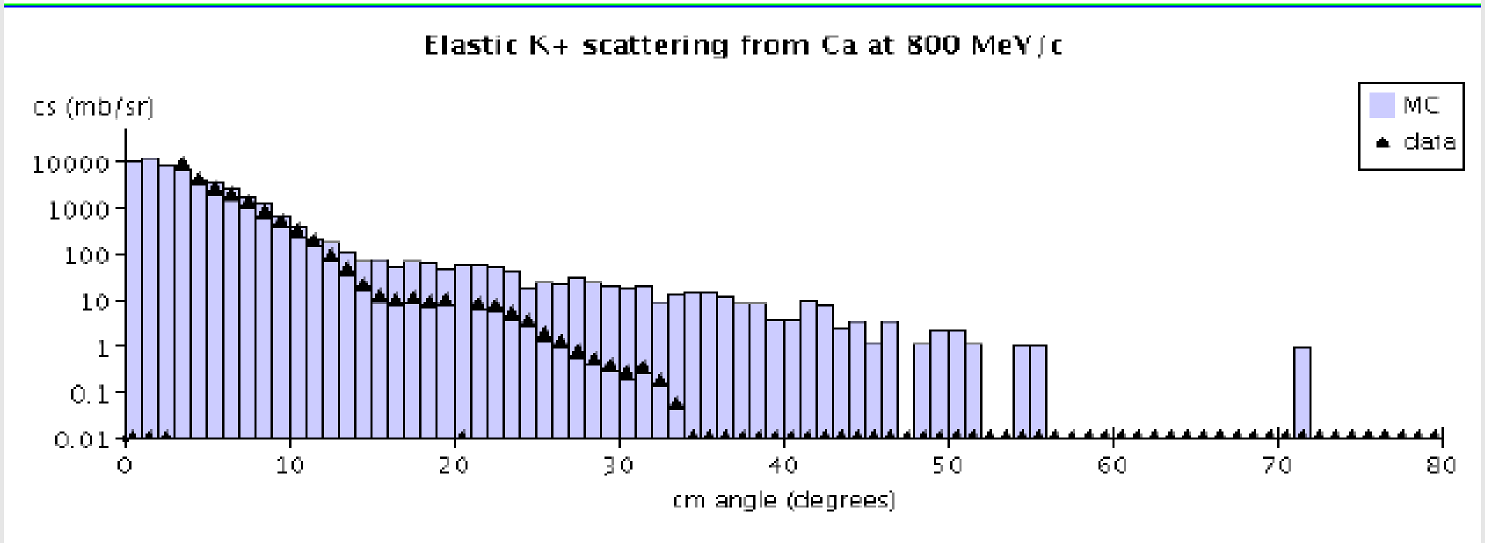
- Modeling sequence:
  - initial interaction of hadron with nucleon in nucleus
  - highly excited hadron is fragmented into more hadrons
  - particles from initial interaction divided into forward and backward clusters in CM
  - another cluster of backward going nucleons added to account for intra-nuclear cascade
  - clusters are decayed into pions and nucleons
  - remnant nucleus is de-excited by emission of p, n, d, t, alpha

Using the LEP and HEP models

- The LEP and HEP models are valid for p, n, π, K, Λ, Σ, Ξ, Ω, α, t, d
  - LEP valid for incident energies of 0 – ~30 GeV
  - HEP valid for incident energies of ~10 GeV – 15 TeV
- Invocation sequence
  - ```
    G4ProtonInelasticProcess* pproc = new G4ProtonInelasticProcess();
    G4LEProtonInelastic* LEproton = new G4LEProtonInelastic();
    pproc -> RegisterMe(LEproton);
    G4HEProtonInelastic* HEproton = new G4HEProtonInelastic();
    HEproton -> SetMinEnergy(20*GeV);
    pproc -> RegisterMe(HEproton);
    proton_manager -> AddDiscreteProcess(pproc);
    ```
    ```

Validation of LEP, HEP Models

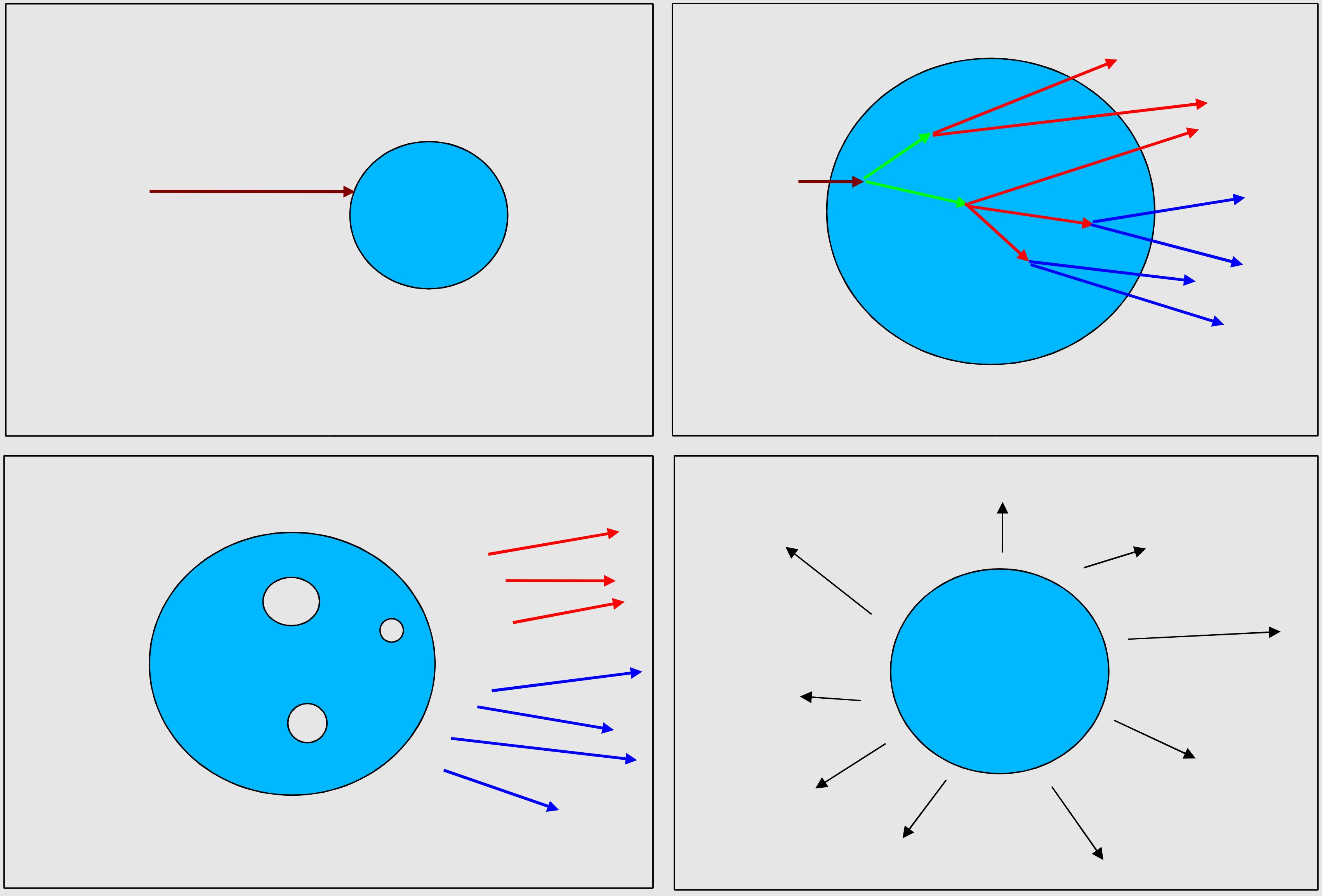




Bertini Cascade Model

- The Bertini model is a classical cascade:
  - it is a solution to the Boltzman equation on average
  - no scattering matrix calculated
  - can be traced back to some of the earliest codes (1960s)
- Core code:
  - elementary particle collider: uses free-space cross sections to generate secondaries
  - cascade in nuclear medium
  - pre-equilibrium and equilibrium decay of residual nucleus
  - detailed 3-D model of nucleus

Bertini Cascade (Comic Book Version)



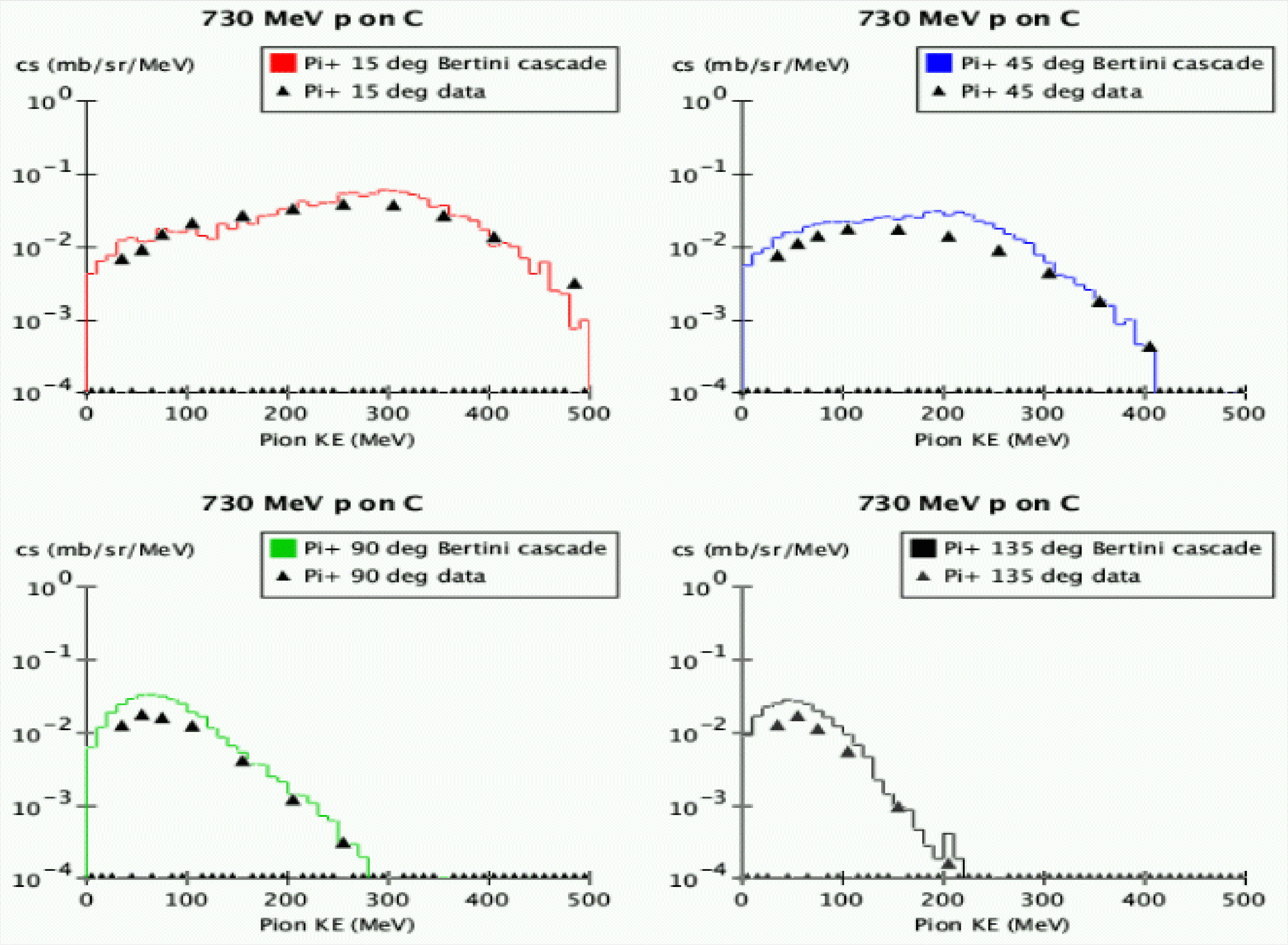
Bertini Cascade (text version)

- Modeling sequence:
  - incident particle penetrates nucleus, is propagated in a density-dependent nuclear potential
  - all hadron-nucleon interactions based on free-space cross sections, angular distributions, but no interaction if Pauli exclusion not obeyed
  - each secondary from initial interaction is propagated in nuclear potential until it interacts or leaves nucleus
  - during the cascade, particle-hole exciton states are collected
  - pre-equilibrium decay occurs using exciton states
  - next, nuclear breakup, evaporation, or fission models

Using the Bertini Cascade

- In Geant4 the Bertini model is currently used for p, n, π, K^+, K^-, K^0_L, K^0_S, Λ, Σ^+, Σ^-, Ξ^-, Ξ^0, Ω^-
  - valid for incident energies of 0 – 10 GeV
  - may be extended to 15 GeV when new validation data are available
  - currently being extended to kaons and hyperons
- Invocation sequence
  - ```
    G4CascadeInterface* bertini = new G4CascadeInterface();
    G4ProtonInelasticProcess* pproc = new G4ProtonInelasticProcess();
    pproc -> RegisterMe(bertini);
    proton_manager -> AddDiscreteProcess(pproc);
    ```
    ```

Validation of the Bertini Cascade



Binary Cascade

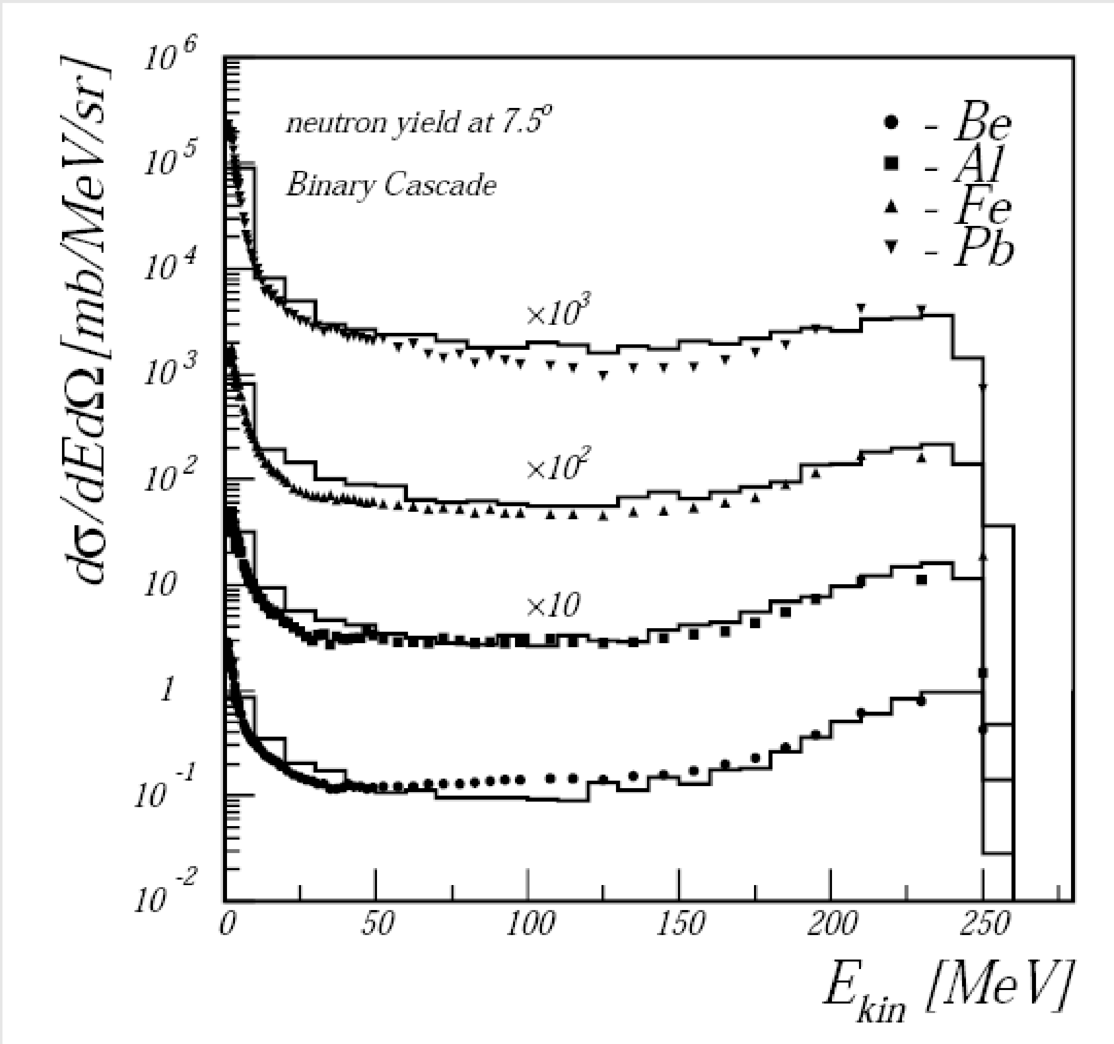
- Modeling sequence similar to Bertini, except that
  - hadron-nucleon collisions handled by forming resonances which then decay according to their quantum numbers
  - particles follow curved trajectories in nuclear potential
- In Geant4 the Binary cascade model is currently used for incident p, n and π
  - valid for incident p, n from 0 to 10 GeV
  - valid for incident π^+, π^- from 0 to 1.3 GeV
- A variant of the model, G4BinaryLightIonReaction, is valid for incident ions up to A = 12 (or higher if target has A < 12)

Using the Binary Cascade

- Invocation sequence Binary cascade
  - ```
    G4BinaryCascade* binary = new G4BinaryCascade();
    G4PionPlusInelasticProcess* pproc = new G4PionPlusInelasticProcess();
    pproc -> RegisterMe(binary);
    piplus_manager -> AddDiscreteProcess(pproc);
    ```

- Invocation sequence BinaryLightIonReaction
  - ```
    G4BinaryLightIonReaction* ionBinary = new G4BinaryLightIonReaction;
    G4IonInelasticProcess* ionProc = new G4IonInelasticProcess;
    ionProc->RegisterMe(ionBinary);
    genericIonManager->AddDiscreteProcess(ionProc);
    ```

Validation of the Binary Cascade 256 MeV protons



Nuclear De-excitation Models (1)

- The parameterized and cascade models all have nuclear de-excitation models embedded in them
- However, there is a de-excitation model that can be used independently: Precompound model
  - valid for incident p, n from 0 to 170 MeV
  - takes a nucleus from a highly-excited set of particle-hole states down to equilibrium energy by emitting p, n, d, t, 3He, alpha
  - once equilibrium state is reached, four other models are called to take care of nuclear evaporation and breakup
    - these models not currently callable by users

Nuclear De-excitation Models (2)

- Invocation of Precompound model:
  - ```
    G4ExcitationHandler*  theHandler = new G4ExcitationHandler;
    G4PrecompoundModel* preModel =
                              new G4PrecompoundModel(theHandler);

    // Create equilibrium decay models and assign to Precompound model


    G4NeutronInelasticProcess* nProc = new G4NeutronInelasticProcess;
    nProc->RegisterMe(preModel);
    neutronManager->AddDiscreteProcess(nProc);

    // Register model to process, process to particle
    ```
    ```

Summary (1)

- Geant4 hadronic physics allows user to choose how a physics process should be implemented:
  - cross sections
  - models
- Many processes, models and cross sections to choose from
  - hadronic framework makes it easier for users to add more

Summary (2)

- Parameterized models (LEP, HEP) handle the most particle types over the largest energy range
  - based on fits to data and some theory
  - not very detailed
  - fast
- Cascade models (Bertini, Binary) are valid for fewer particles over a smaller energy range
  - more theory-based
  - more detailed
  - slower