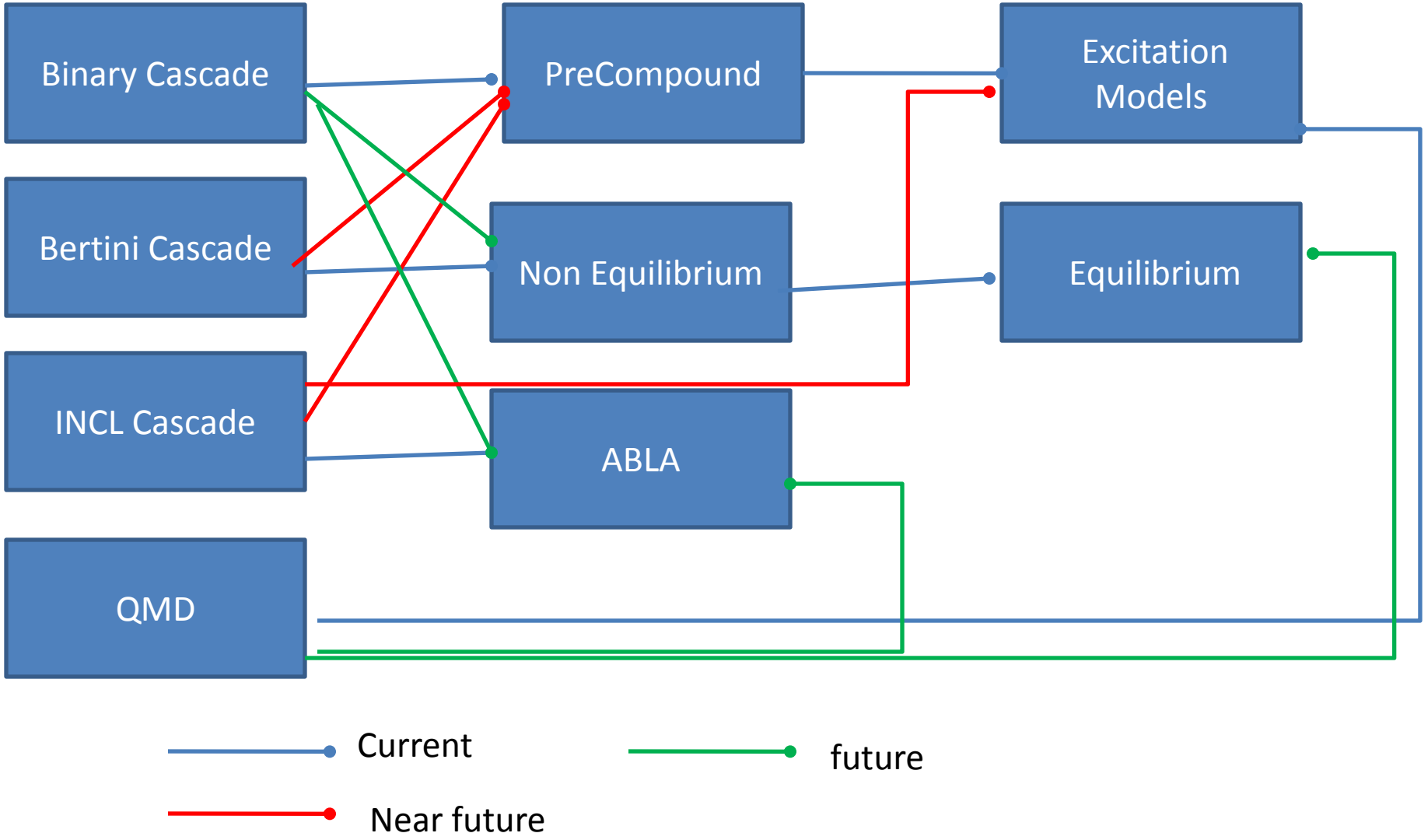
Integration of Precompound/De-excitation with Other Models

IVANTCHENKO, Vladimir

KOI, Tatsumi

QUESADA MOLINA, Jose Manuel

Exchange (sub) models



Internal electron & Auge Electron

- Electron Occupancies in
  - Nucleon – Nucleus interaction
  - Nucleus – Nucleus interaction
    - Usually projectile ions are naked
- Photon Evaporation
- Radio Active Decay

PreCompound/Deexcitation Interface

G4VPreCompound interface

- Constructor:

G4VPreCompoundModel(G4ExcitationHandler*, const String& name)

- Methods:

G4HadFinalState* ApplyYourself(…) – is not used

G4RecactionProductVector* DeExcite(G4Fragment&) – initial fragment may be modified

SetExcitationHandler(G4ExcitationHandler*)

G4Fragment – key class

Do we have enough Constructors
excitation energy, excitons??

- Constructors:

G4Fragment(G4int A, G4int Z, const G4LorentzVector&) - nucleus
G4Fragment(const G4LorentzVector&, G4ParticleDefinition*) – gamma, e-

- Fragment can be modified:

Void SetZandA_asInt(G4int Znew, Anew)
Vois SetMomentum(const G4LorentzVector&)

- Keep number of excitons : p, n, and holes for pre-compound model
- Keep time of the decay and number of e- (new) at shell for evaporation model
- Should allocator be used for G4Fragment?

Who use G4int numberOfParticles;
G4int numberOfHoles;
G4int numberOfCharged;

G4PreCompoundModel
This is only ONE concrete class

G4ExcitationHandler

- Main method:

G4ReactionProductVector* BreakItUp(const G4Fragment&) const – initial G4Fragment unchanged (?)

- Methods for steering:

Void SetEvaporation(G4VEvaporation*)

Void SetMultiFragmentation(G4VMultiFragmentation*)

Void SetFermiBreakUp(G4VFermiBreakUp*)

Void SetPhotonEvaporation(G4VEvaporationChannel*)

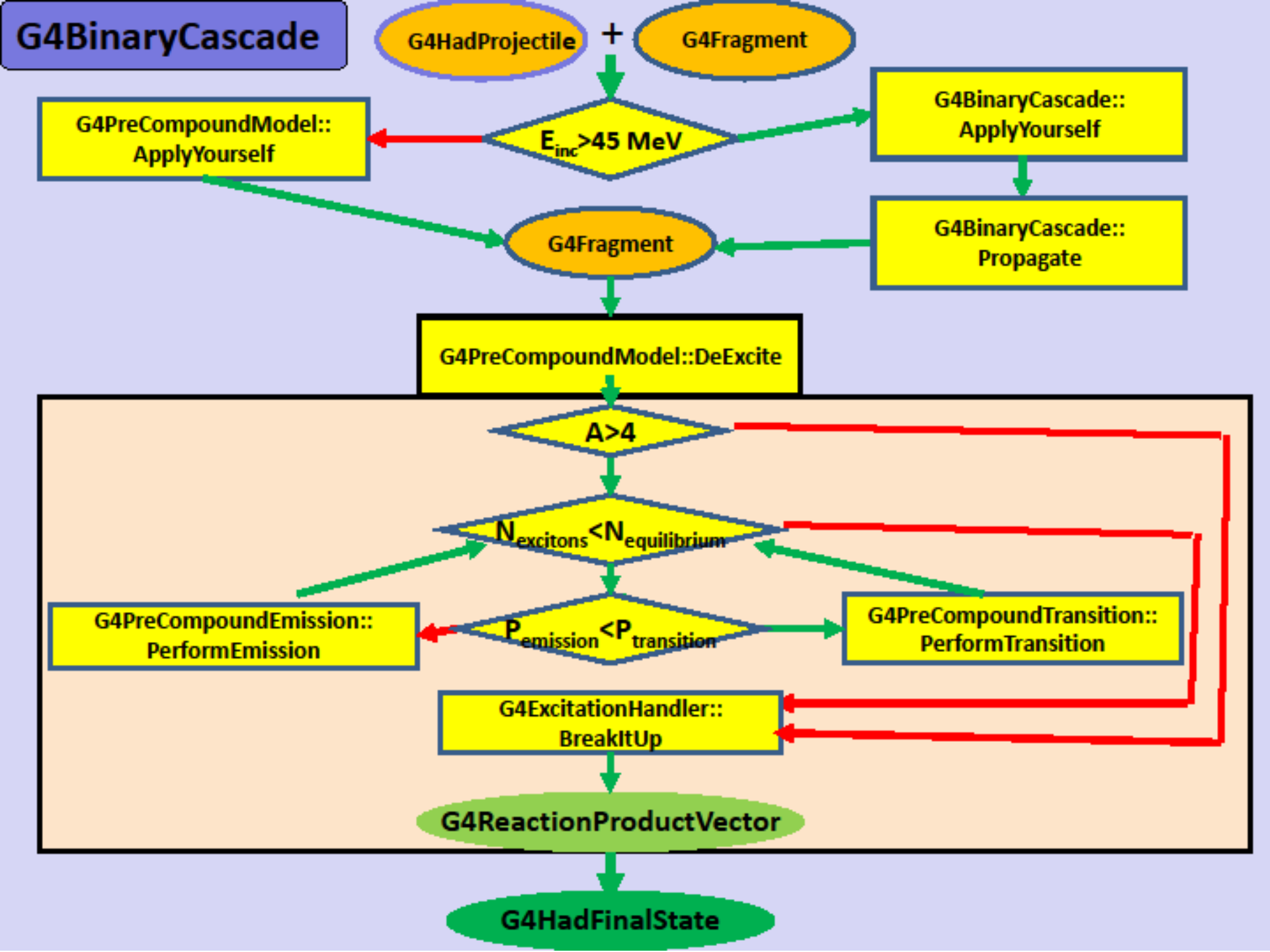
Do we need this level abstraction?

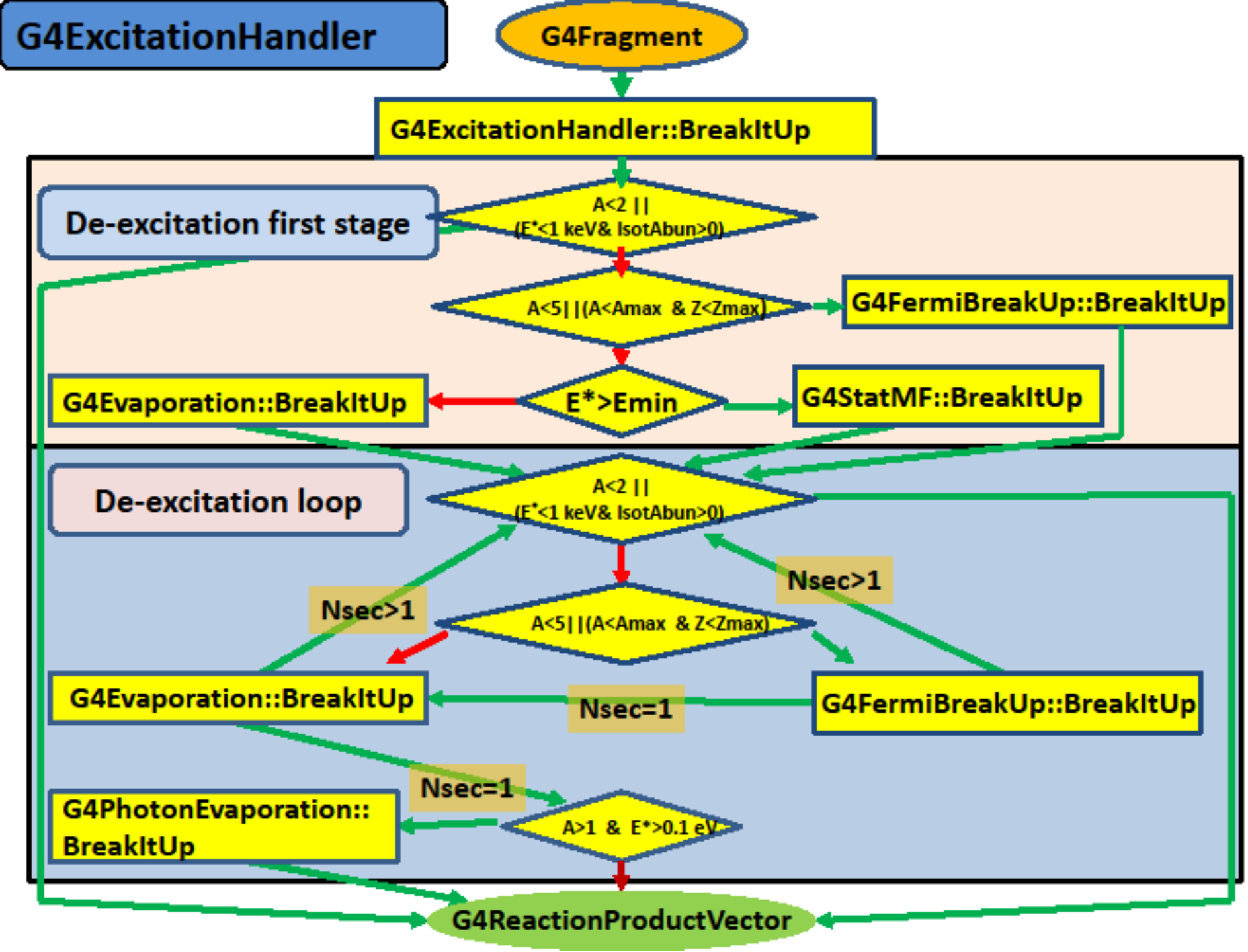
Void SetMaxAandZForFermiBreakUp(G4int Anew, G4int Znew)

Void SetMinEForMultiFrag(G4double)

Void SetOPTxs(G4int)

Void SetSICB()


G4BinaryCascade
G4HadProjectile
+
G4Fragment
G4PreCompoundModel::
ApplyYourself
E_inc>45 MeV
G4BinaryCascade::
ApplyYourself
G4Fragment
G4BinaryCascade::
Propagate
G4PreCompoundModel::DeExcite
A>4
N_excitons<N_equilibrium
G4PreCompoundEmission::
PerformEmission
P_emission<P_transition
G4PreCompoundTransition::
PerformTransition
G4ExcitationHandler::
BreakItUp
G4ReactionProductVector
G4HadFinalState


G4ExcitationHandler
G4Fragment
G4ExcitationHandler::BreakItUp
De-excitation first stage
A<2 ||
(E*<1 keV& IsotAbun>0)
A<5||(A<Amax & Z<Zmax)
G4FermiBreakUp::BreakItUp
G4Evaporation::BreakItUp
E*>Emin
G4StatMF::BreakItUp
De-excitation loop
A<2 ||
(E*<1 keV& IsotAbun>0)
Nsec>1
A<5||(A<Amax & Z<Zmax)
Nsec>1
G4Evaporation::BreakItUp
Nsec=1
G4FermiBreakUp::BreakItUp
Nsec=1
G4PhotonEvaporation::
BreakItUp
A>1 & E*>0.1 eV
G4ReactionProductVector

Summary

- Create G4Fragment
- And Set required member (holes, excitons and so on)

```
2321
2322    G4Fragment * fragment = new G4Fragment(a,z,GetFinalNucleusMomentum());
2323    fragment->SetNumberOfHoles(holes);
2324
2325  //GF  fragment->SetNumberOfParticles(excitons-holes);
2326    fragment->SetNumberOfParticles(excitons);
2327    fragment->SetNumberOfCharged(zCaptured);
2328    G4ParticleDefinition * aIonDefinition =
2329         G4ParticleTable::GetParticleTable()->FindIon(a,z,0,z);
2330    fragment->SetParticleDefinition(aIonDefinition);
```

- G4Fragment
  - Constructor
  - Electron ?
- High Energy <-> Cascade integration.