Framework Getting Started To-Do

- Geometry, geometry, geometry . . .
- Interface with Geant4
- Data model, class definitions (digits, hits, clusters, . . .)
- Simple algorithms, *e.g.*
  - GEM & calorimeter digitization
  - GEM cluster finder
  - Basic calorimeter cluster finder
  - Similar for Cherenkovs
- Conditions database (try CCDB)
- Packaging, local documentation
- Research tracking algorithms (NB: our tracking problem has already been solved somewhere!)

Guidelines

- Develop in *art* to evaluate the framework
- Mind portability — we might want to use the code elsewhere after all
  - Generalize interfaces
  - Embed algorithms in adapter classes (like LArSoft does)
  - Avoid direct use of *art*-internals where possible
- Mind concurrency. Write thread-safe code:
  - No globals, statics (except constants)
  - Use thread-local storage where needed (supported in `C++11`)
  - Seek advice if it looks like something might need to be locked
  - Assume the framework will manage concurrency. Don't parallelize anything yourself (*e.g.* no OpenMP)