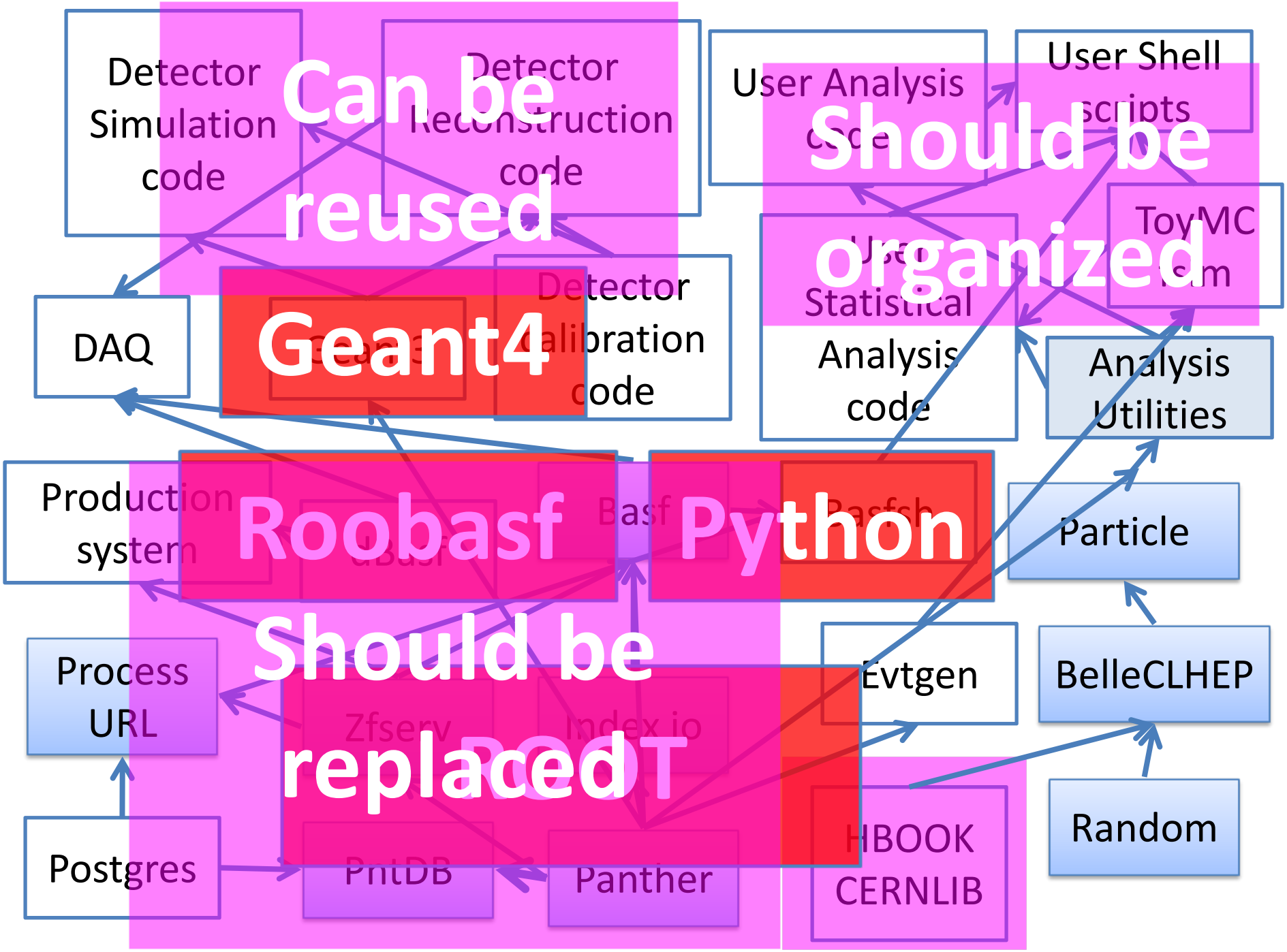
# Transition

Nobu Katayama
B2GM
July, 7, 2009

# Belle II software

- Start from scratch?
  - Need a lot of thinking, extensive work at the beginning (NOW!)
- Reuse Belle software?
  - May not be sustainable for the next 10+ years
- We want to both Belle and Belle II for the next several years
  - Also important for benchmarking/debugging
- I think a smooth transition is possible if we do it carefully
  - we must be determined to do so


Detector Simulation code
Detector Reconstruction code
User Analysis code
User Shell scripts
Can be reused
Should be organized
Geant4
DAQ
Detector Calibration code
Statistical Analysis code
ToyMC
Analysis Utilities
Production system
Roobasf
Basf
Python
Basfsh
Particle
Process URL
Should be replaced
PANTHER
Evtgen
BelleCLHEP
Zfserv
Index io
Postgres
PntDB
Panther
HBOOK CERNLIB
Random

# Examples

- Replace
  - BelleRootHistogram replaced BelleHistogram (HBOOK) without chaining interface (No change in user code)
- Reuse
  - Basf_io (interface to legacy panther IO, zfserv, index etc) is reused in roobasf
- Coexistence $\Rightarrow$ change
  - Old CLHEP is now in Belle:: namespace (in belleCLHEP directory) and can co-exist with the standard CLHEP
- New functionality added
  - Particle classes can now be persistent

# How to proceed?

- Small and full scale prototyping for all components is important
  - It is exactly the same as detector R&D!
- Framework is the glue (not guru)
  - Build our own so that we can use the (standard) components as we wish
- Use the language standard (+de facto standard) components if possible
  - Don't switch from Belle specific to Belle II specific
- Make the most use of C++ and Python
  - Namespace, templates, STL, allocator
  - Make use of free/good libraries/modules