

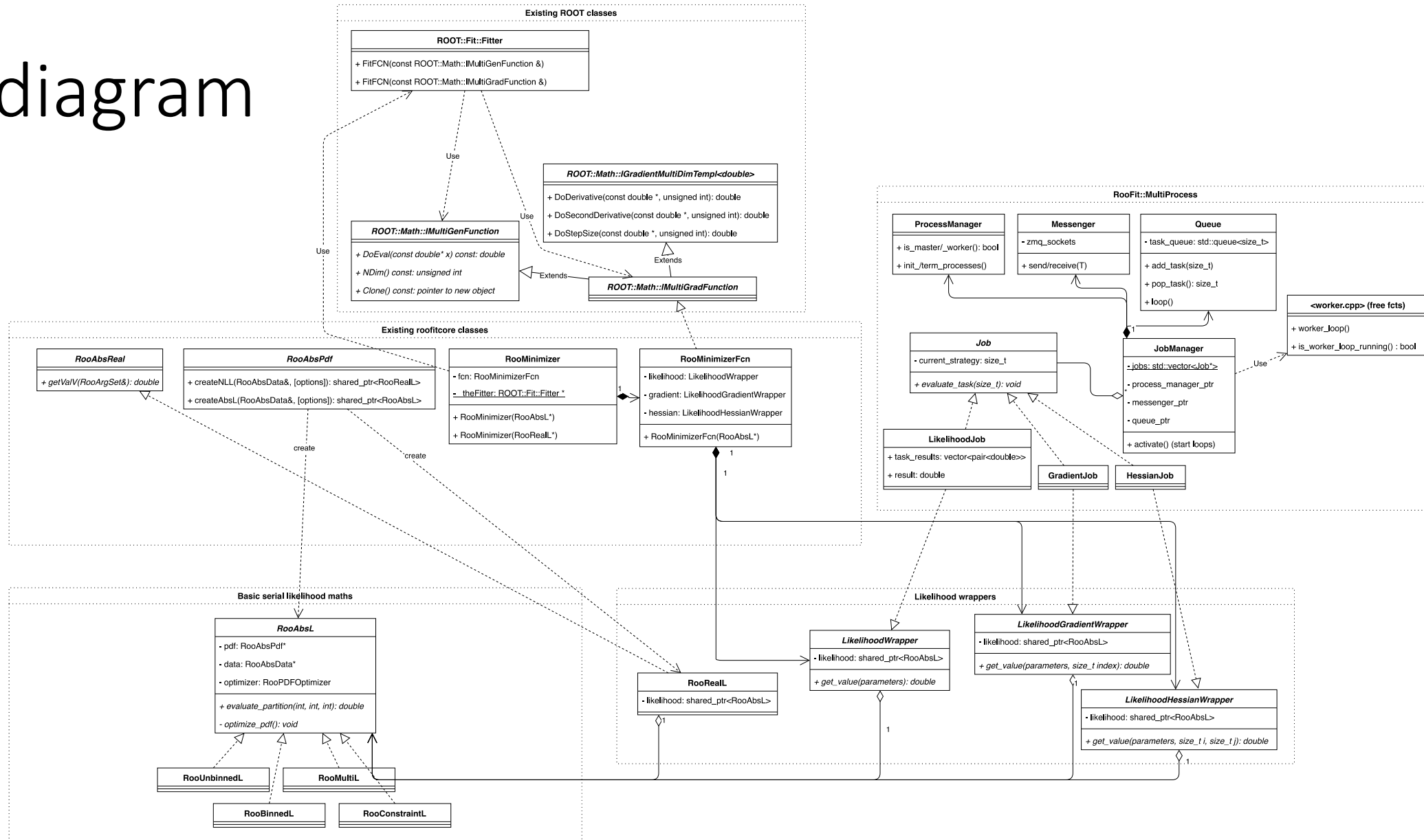
# Roofit:: ::TestStatistics & ::MultiProcess

Redesign of “NumCPU” option for (a.o.) parallelized gradient calculation

# Intro

- Redesign of 2019 first implementation of RooFit::MultiProcess
- Flatten RooAbs(Opt)TestStatistic + RooNLLVar tree
- Split out into more “conceptual” RooAbsL -> Roo(Un)BinnedL, RooSumL & RooSubsidiaryL (constraint terms)
- Back-end for calculation
  - Abstract classes
  - Parallel processes in our case
    - Communication via ZeroMQ -> added dependency
    - Licensing issues?
  - Can be easily implemented for other back-end types (e.g. GPU)

# Class diagram



# User interface

- `likelihood_creator`: (free) functions with functionality from `RooAbsPdf::createNLL`, but for new classes
  - Could in turn be called again from (new version of) `RooAbsPdf::createNLL`
  - Takes in optional parameters
    - Strong types in current implementation, instead of `RooCmdArg` list
    - implemented: `Constrain`, `GlobalObservables`, `ExternalConstraints`, `offsetting`, `Extended`
    - still to be ported: `Range{,*}`, `ProjectedObservables`, `CloneData`, `Optimize`

# Final steps (this year)

- Fix final bugs (hopefully done by this talk)
- Merge!
  - currently at [https://github.com/roofit-dev/root/tree/MP\\_ZeroMQ\\_new\\_multiprocess](https://github.com/roofit-dev/root/tree/MP_ZeroMQ_new_multiprocess)
- Benchmark again
  - First version already benchmarked, see ACAT19/CHEP19 talks
    - ACAT19 talk: <https://indico.cern.ch/event/708041/contributions/3276177/>
    - CHEP19 proceedings: <https://inspirehep.net/literature/1832224>
  - *Should* be similar