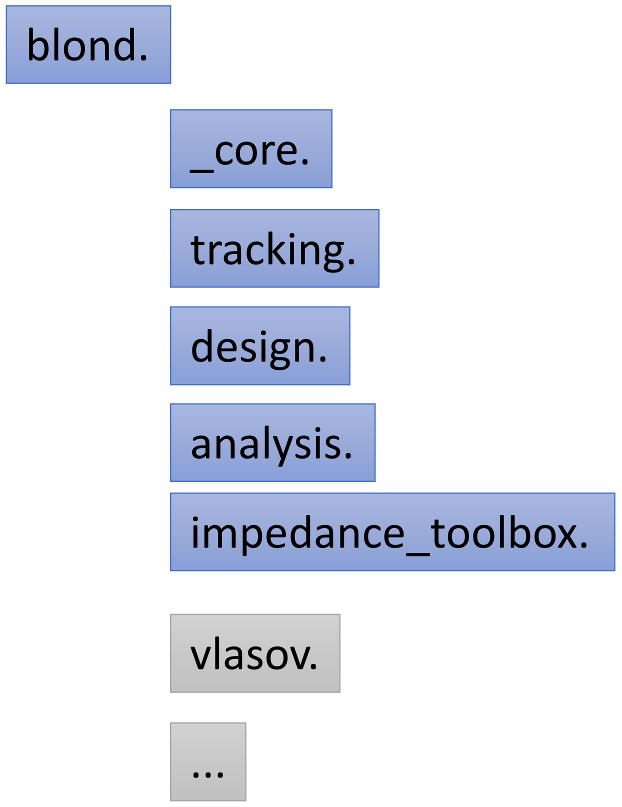
SPARE
“BLonD suite”
Structure details

Structure proposal ("BLonD suite")



- One _core package, where common functions and dependencies are located (acts as the "toolbox")
- One package per functionality, that picks functions and inherits from common objects in the _core package.
- The tracking package is the present version of BLonD, that can be encapsulated here as a whole.
- Common functions and classes can be migrated gradually to the _core package.
- The user only uses the package(s) he is interested in.
- Only the "public" parts of the codes are included, the "CERN internal" parts can be kept as plugins available on Gitlab.

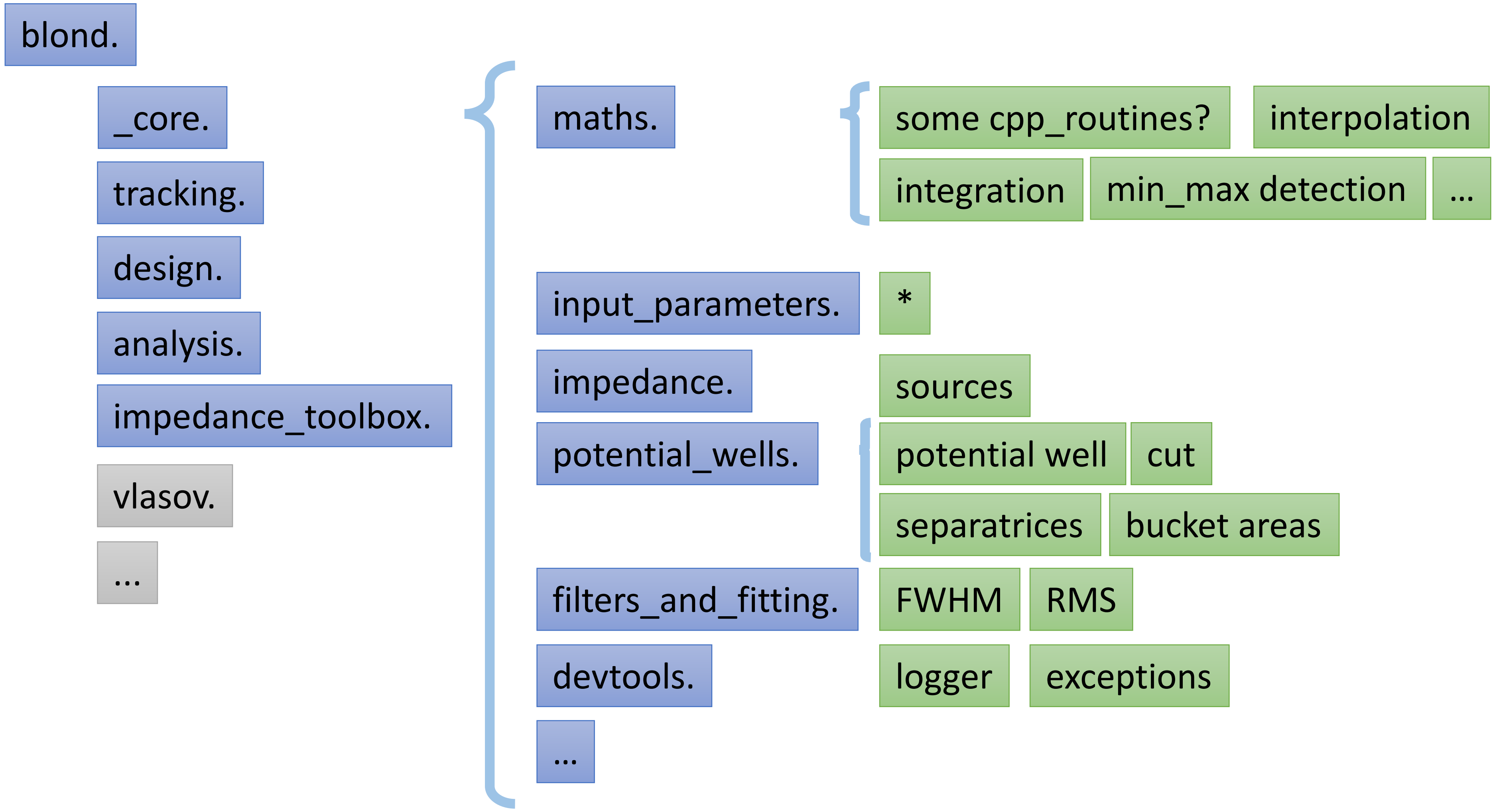
Pros/cons

- Summary: the core package acts as the "toolbox", other packages are built in the same project around the core module. The core package is intended to be used by developers/experts, all the other packages are only dependent to the core and not with each other.

- Pros:
  - No need to use (and struggle with) submodules
  - Each module only rely on the core module, and are independent from each other, a bug in the design module is not propagated to the tracker and vice-versa
  - All modules can be proposed to LCG software altogether, accessible through swan and the control room since CO's python distribution will rely on LCG

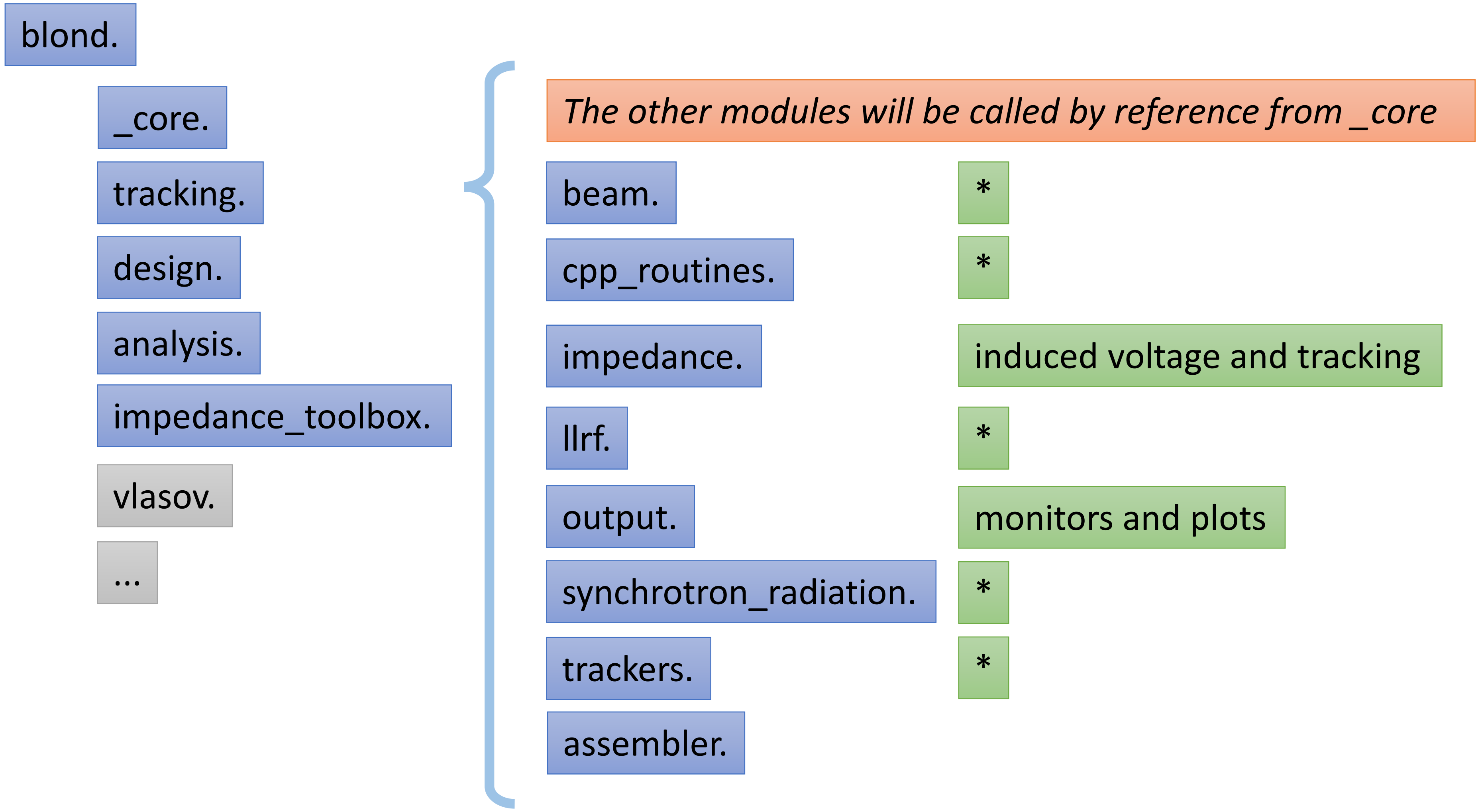
- Cons:
  - Need to adapt C++ libraries handling, some modules will only rely on python and should not be limited if a compiler is not available
  - Some reorganization needed, but not unreasonable (-> put all existing modules into a tracking package, and progressively migrate selected parts in core package)
  - Some effort on documentation to explicitly define the purpose of each package

Structure proposal: _core



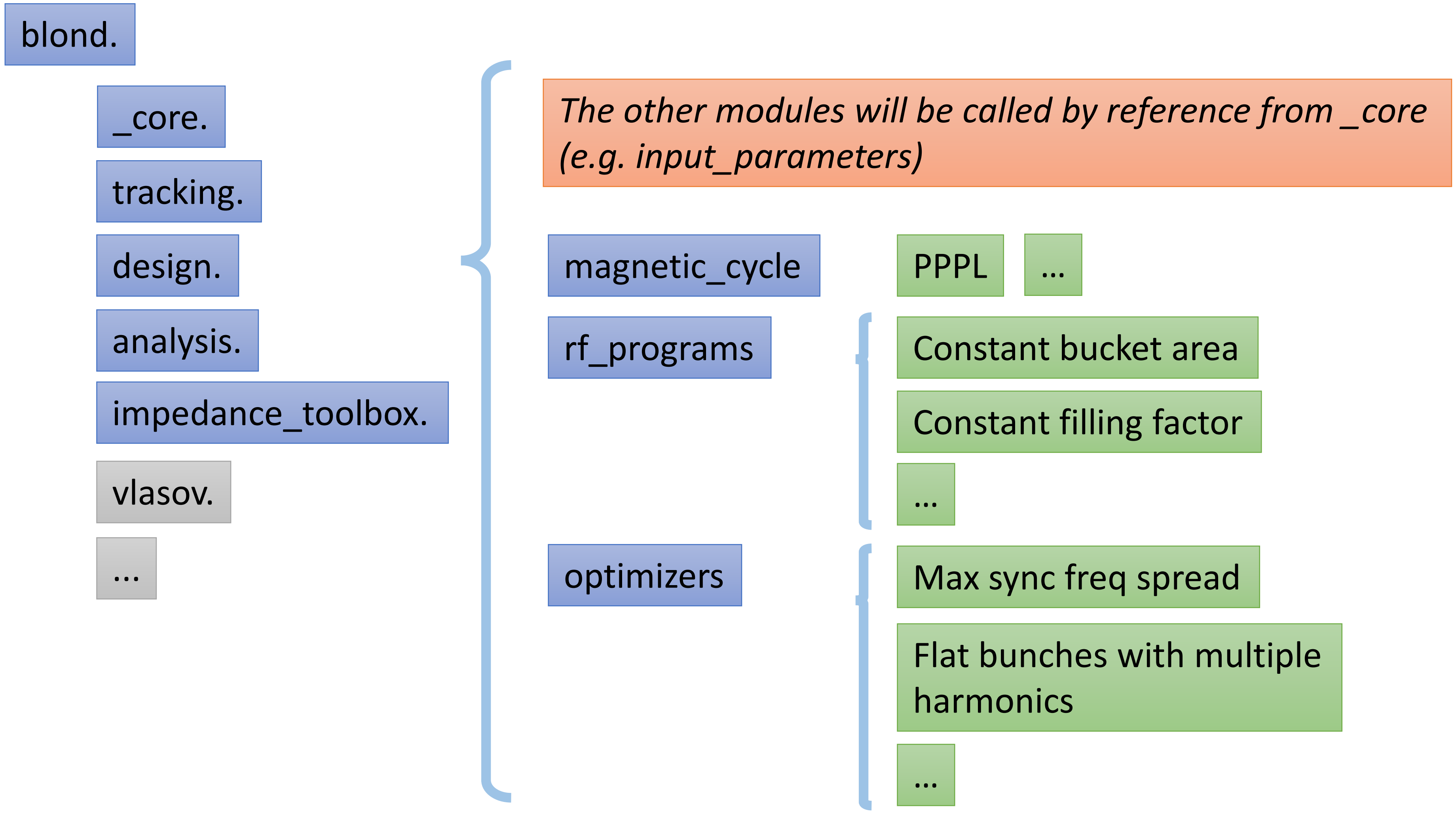
- _core package intended for developers/expert users
- Acts as a toolbox for the other packages

Structure proposal: tracking



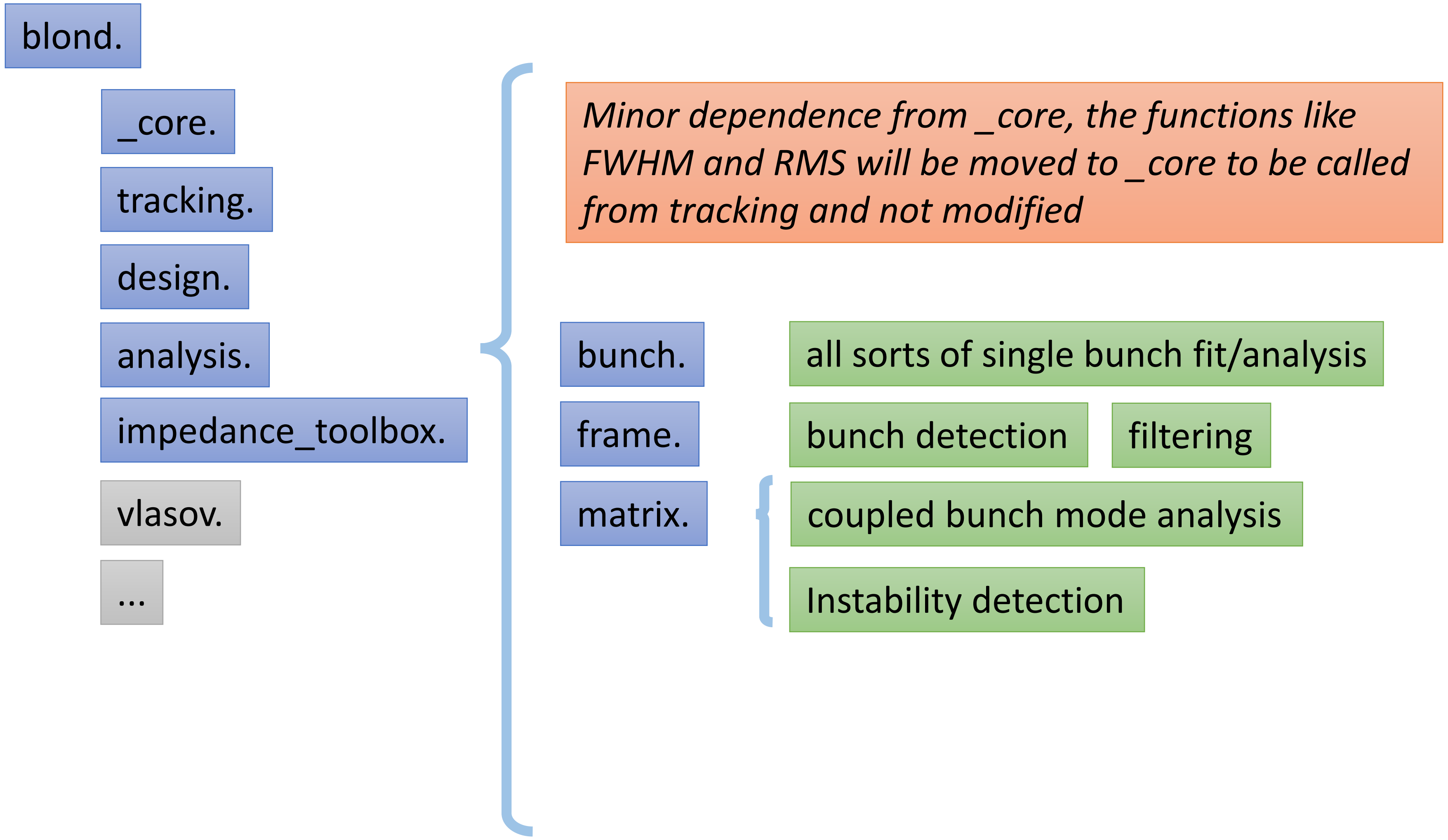
- tracking package is present BLonD minus modules moved to _core
- Further developments (e.g. assembler) can be done in this package with no impact on the others

Structure proposal: design



- Inherit from the core class, or even overwrite it (e.g. RingOptions)
- Scripts from Elena in FORTRAN to be recovered

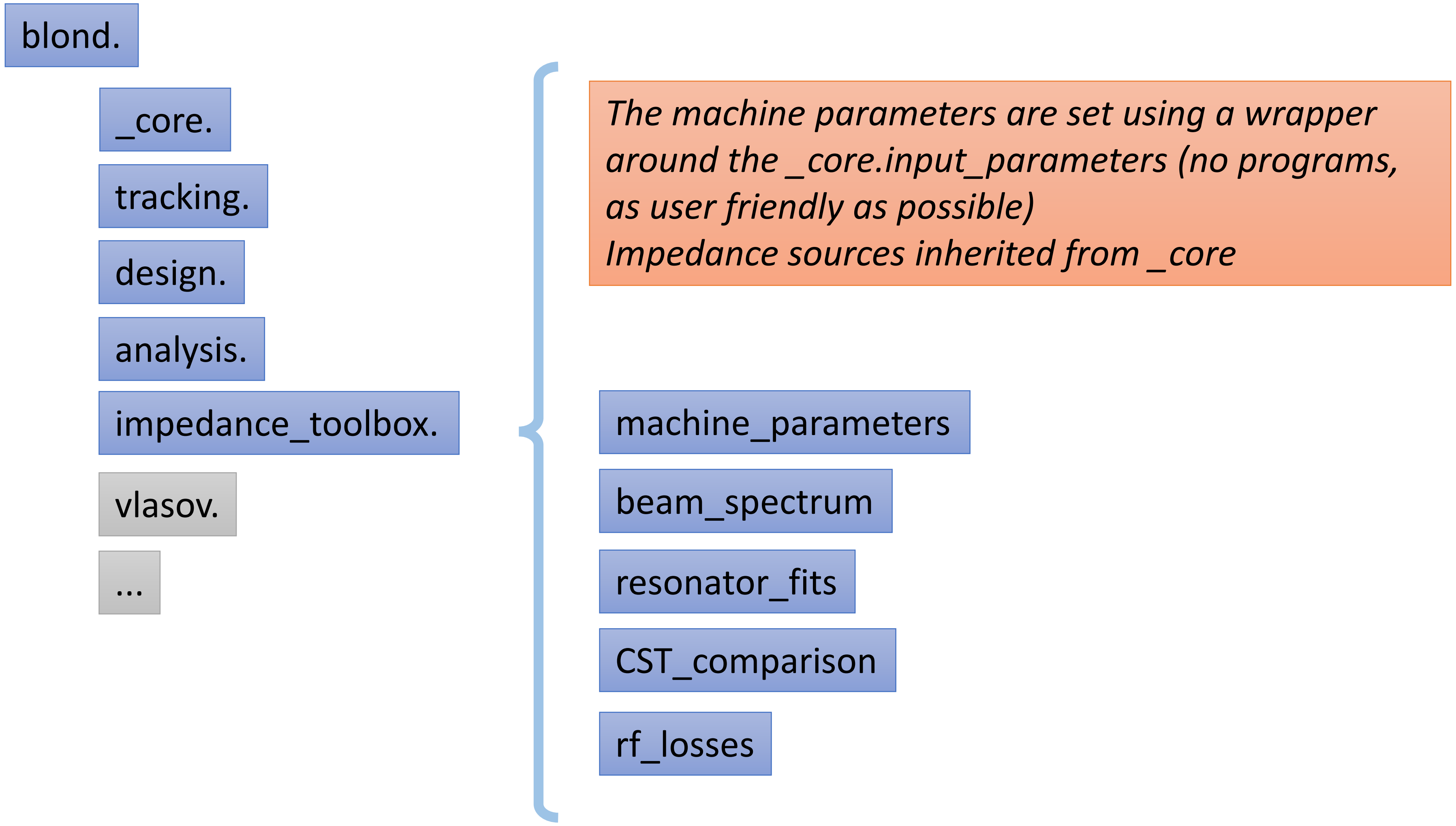
Structure proposal: analysis



Minor dependence from _core, the functions like FWHM and RMS will be moved to _core to be called from tracking and not modified

- Standalone functions with simple usage

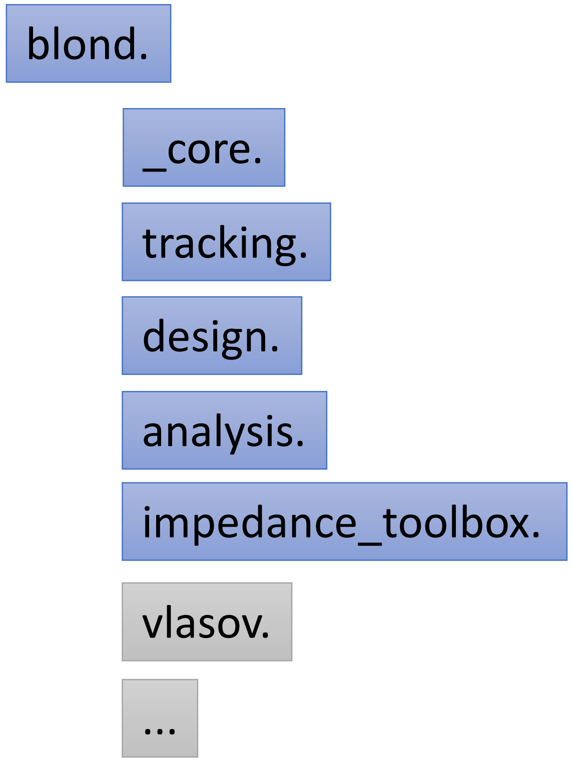
Structure proposal: impedance_toolbox



The machine parameters are set using a wrapper around the _core.input_parameters (no programs, as user friendly as possible)
Impedance sources inherited from _core

- Toolbox to ease the information sharing between impedance/beam dynamics

Structure proposal: others…



- Future packages can be included using the same principle