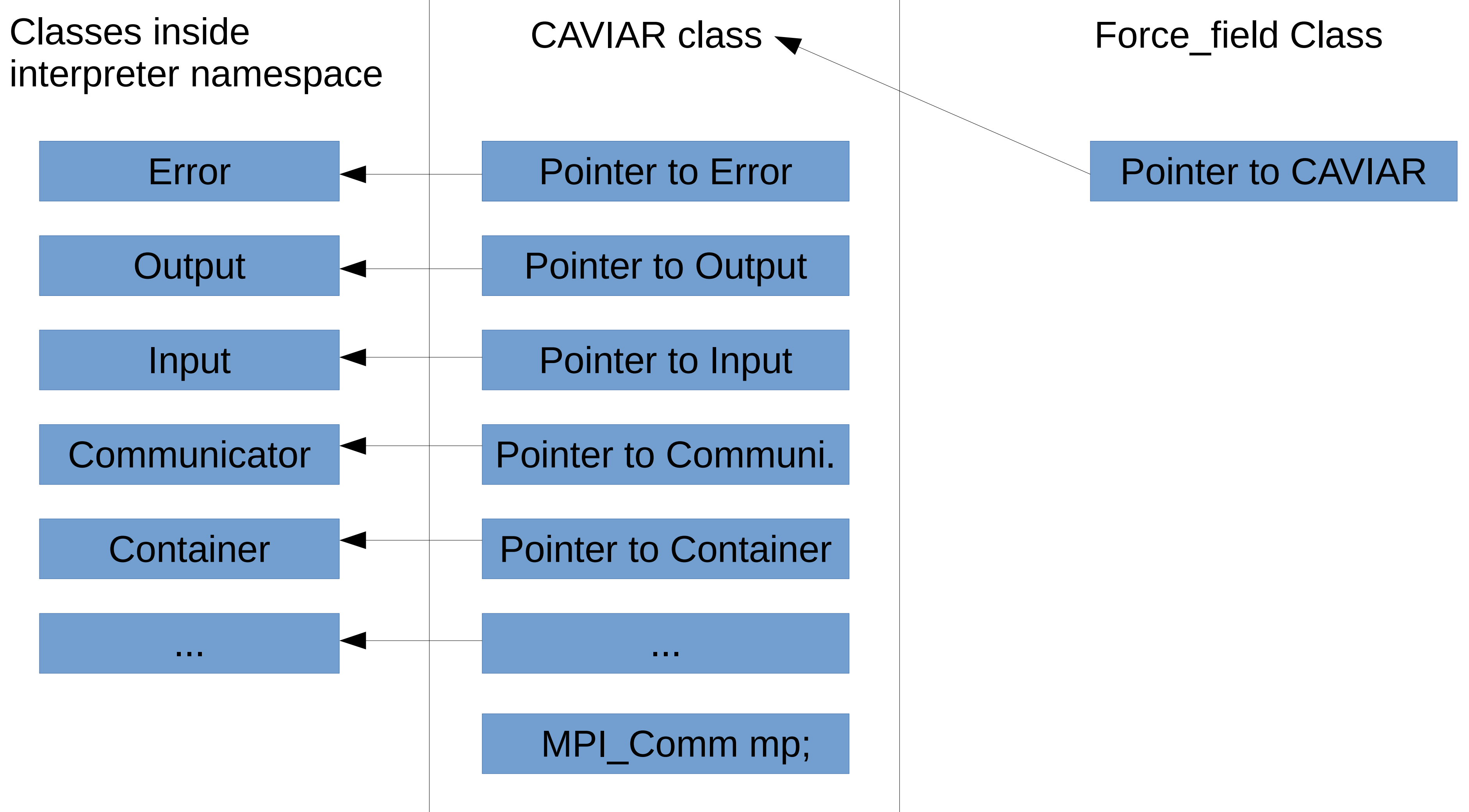
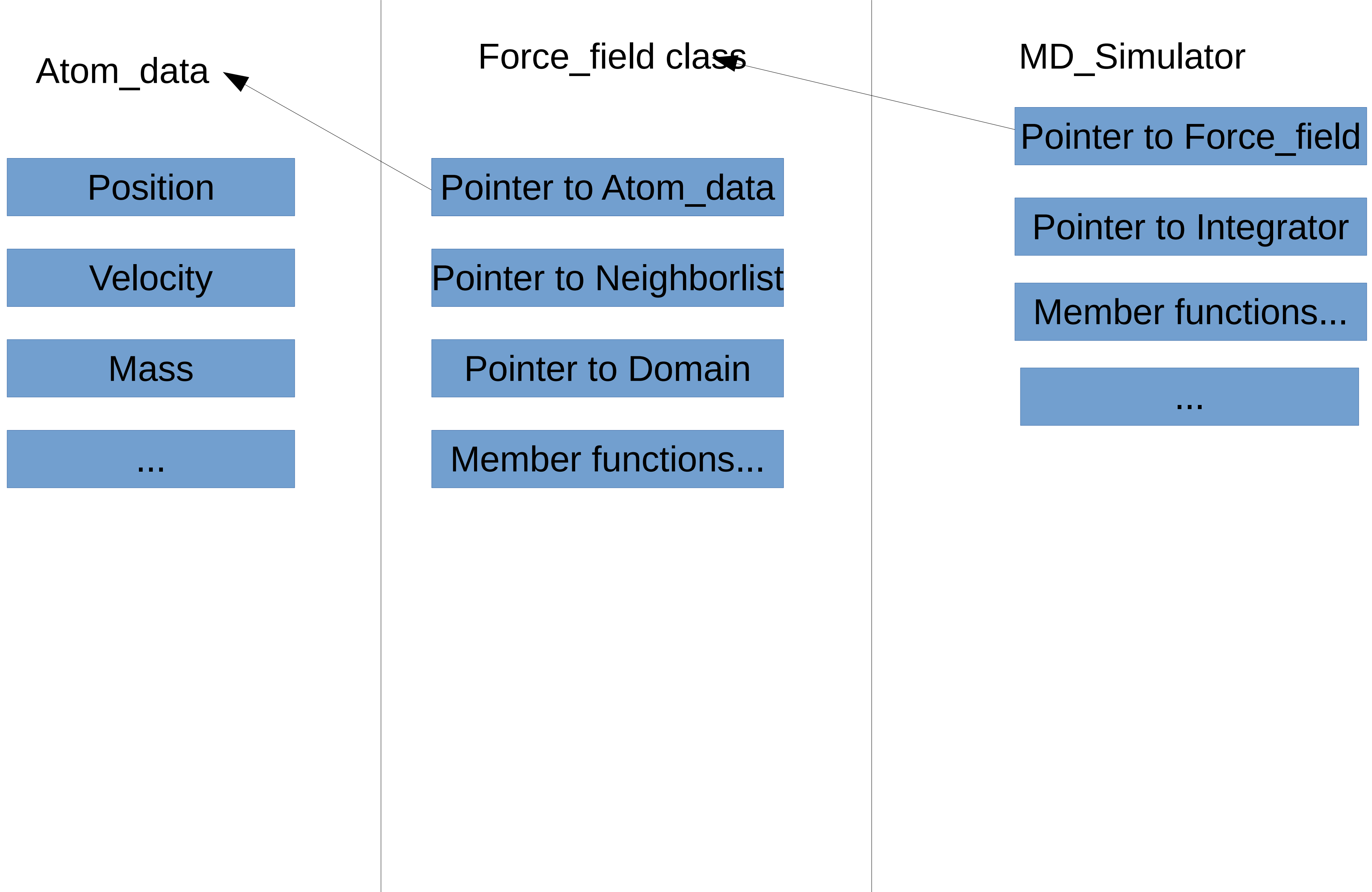
CAVIAR with Boost Python II

```
#===== Atom definition =====

unique atom a1
a1 type 0
a1 position -1 0 0

unique atom a2
a2 type 0
a2 position 1 0 0


#===== Domain

domain box dom
dom xmin -50 xmax 50
dom ymin -50 ymax 50
dom zmin -50 zmax 50
dom boundary_condition 0 0 0
dom generate

#========== Atom_data

atom_data basic adata
adata ghost_cutoff 5
adata cutoff_extra 0.01
adata set_domain dom
adata add_atom a1
adata add_atom a2
adata add_type_mass   0 1.0
adata add_type_charge 0 0.0
```

```
import caviarmd

c = caviarmd.caviar("")

#===== Atom definition =====

a1 = caviarmd.unique.Atom(c)
a1.type = 0
a1.position=[-1,0,0]

a2 = caviarmd.unique.Atom(c)
a2.type = 0
a2.position=[1,0,0]

#===== Domain

dom = caviarmd.domain.Box(c)
dom.lower_global = [-50,-50,-50]
dom.upper_global = [50,50,50]
dom.boundary_condition = [0,0,0]
dom.generate()

#========== Atom_data

adata = caviarmd.atom_data.Basic(c)

adata.ghost_cutoff = 5
adata.cutoff_extra = 0.01
adata.domain = dom
adata.add_atom(a1)
adata.add_atom(a2)
adata.add_type_mass(0,1.0)
adata.add_type_charge(0,0.0)
```


Classes inside interpreter namespace
CAVIAR class
Force_field Class
Error
Output
Input
Communicator
Container
...
Pointer to Error
Pointer to Output
Pointer to Input
Pointer to Communi.
Pointer to Container
...
MPI_Comm mp;
Pointer to CAVIAR


Atom_data
Position
Velocity
Mass
...
Force_field class
Pointer to Atom_data
Pointer to Neighborlist
Pointer to Domain
Member functions...
MD_Simulator
Pointer to Force_field
Pointer to Integrator
Member functions...
...

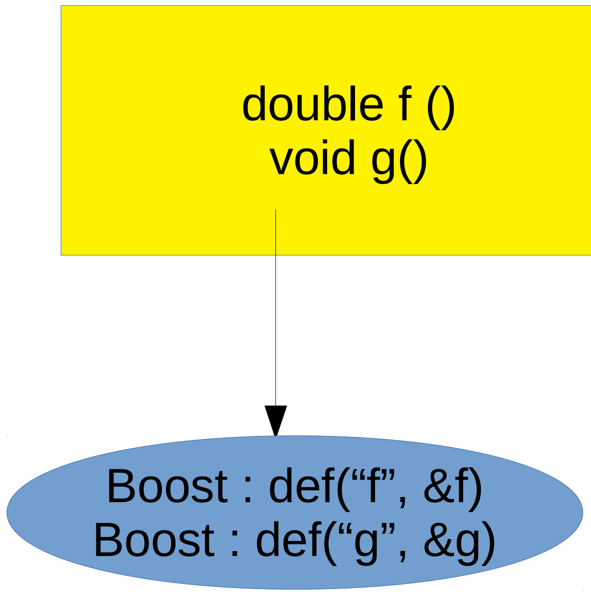
src/caviar/CAVIAR.cpp

include/caviar/objects/force_field/lj.h

src/caviar/objects/force_field/lj.cpp

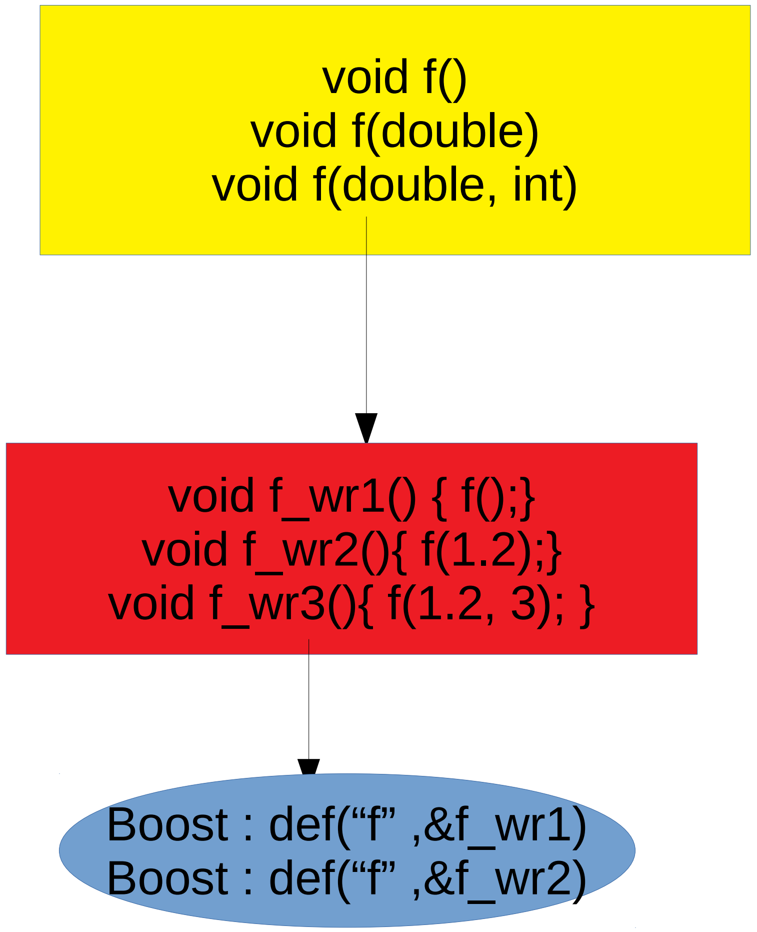
include/caviar/utility/python_utils_dec.h

include/caviar/utility/python_utils_def.h



```
double f ()
void g()
```

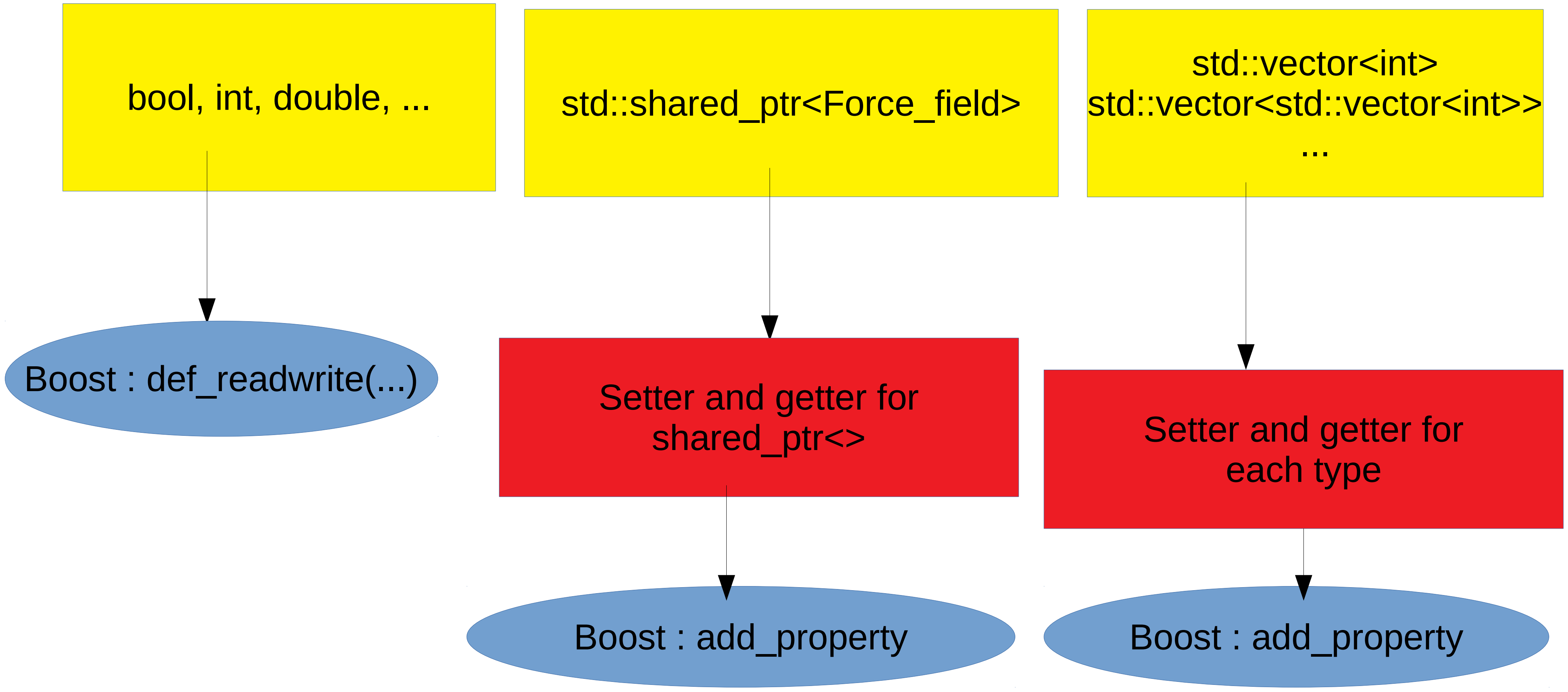
```
Boost : def("f", &f)
Boost : def("g", &g)
```



```
void f()
void f(double)
void f(double, int)
```

```
void f_wr1() { f();}
void f_wr2(){ f(1.2);}
void f_wr3(){ f(1.2, 3); }
```

```
Boost : def("f" ,&f_wr1)
Boost : def("f" ,&f_wr2)
```


bool, int, double, ...
std::shared_ptr<Force_field>
std::vector<int>
std::vector<std::vector<int>>
...
Boost : def_readwrite(...)
Setter and getter for shared_ptr<>
Setter and getter for each type
Boost : add_property
Boost : add_property