Integration of PyROOT and Current Cppyy

Enric Tejedor

ROOT

Data Analysis Framework

https://root.cern

Introduction

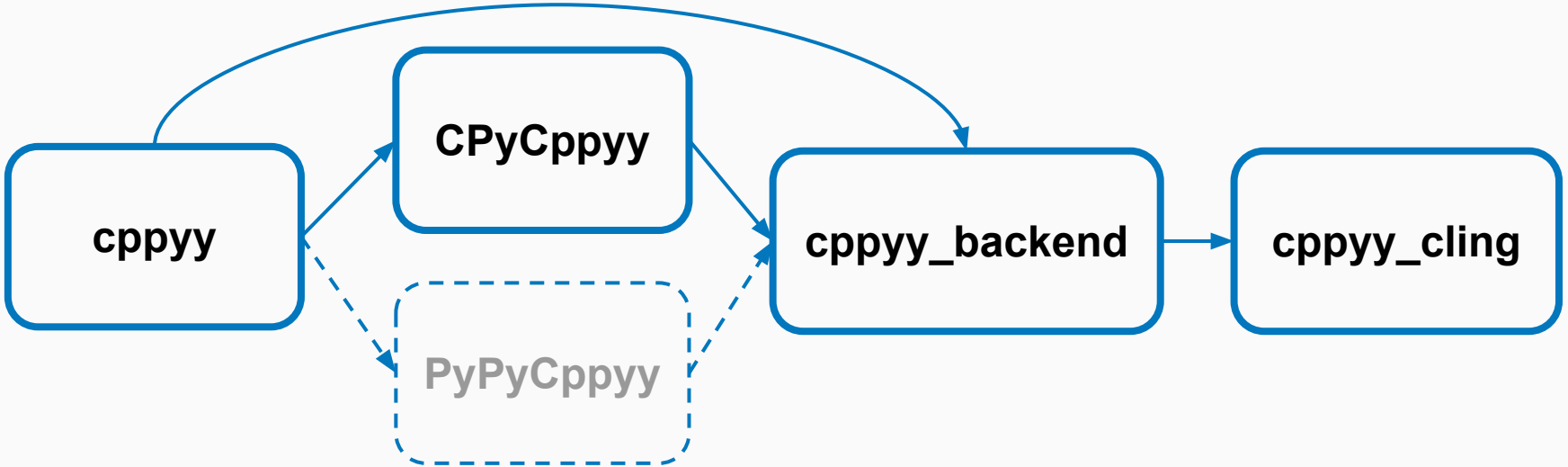
- What I did
  - Took current Cppyy and integrated it into ROOT
  - Basically replaced PyROOT with Cppyy
- Why I did it
  - To understand better the structure of Cppyy and how it compares to PyROOT's
  - To check if we can incorporate into PyROOT the new features of Cppyy

Cppyy

- Automatic Python-C++ bindings generator
- Author: Wim Lavrijsen (former PyROOT developer)
- Aim: standalone layer on top of Cling
  - Although not there yet, still ROOT-dependent

Cppyy: Structure

- A Cppyy installation is divided in 4 packages:



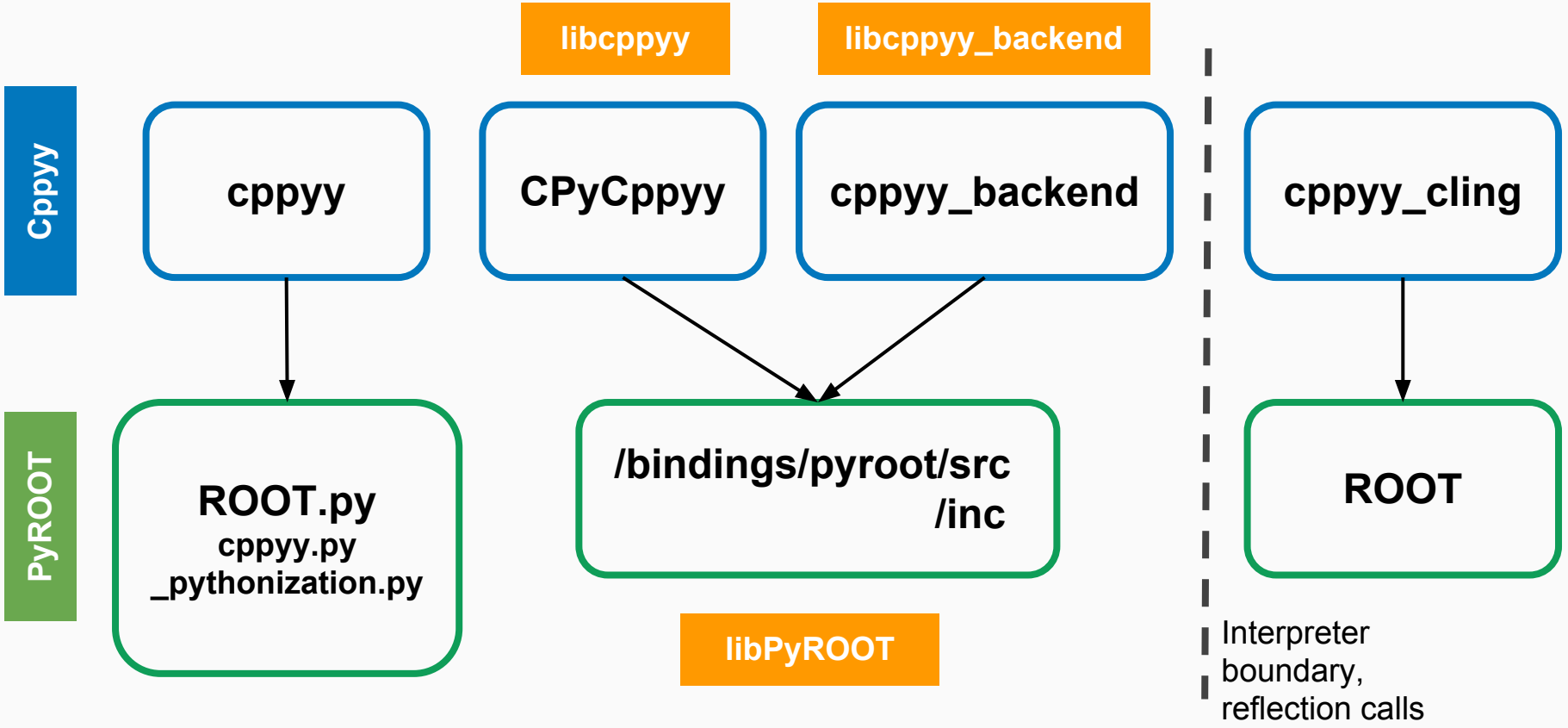
User API

Type Converters, Pythonizations, … (Python/C API)

Small Cling wrapper

Cling + LLVM + “MiniROOT”

Mapping Cppyy to PyROOT



Results of the Integration

- ✗ ROOT-specific pythonizations are no longer there
  - TTree, TFile, TFX, interactive graphics, …
- ✗ Test failures
  - Python tests: 95 %
  - Pyroot tutorials: 57 %
- ✓ New features
  - Lambda variables
  - Move semantics (r-value reference parameters)

Lambdas

- This works now:

```
>>> import ROOT
>>> ROOT.gInterpreter.ProcessLine(
"auto mylambda = [](int i) { std::cout << i << std::endl; };")
140518947094560L
>>> ROOT.mylambda
<cppyy.gbl.function<void(int)>* object at 0x35f9570>
>>> ROOT.mylambda(2)
2
```

Move Semantics

- This is also possible now (requires a fix in cling):

```
>>> import ROOT
>>> ROOT.gInterpreter.ProcessLine(
'void myfunction(std::vector<int>&& v) {
   for (auto i : v) std::cout << i << " ";
 }')
0L
>>> v = ROOT.std.vector(int)(range(10))
>>> ROOT.myfunction(ROOT.std.move(v))
0 1 2 3 4 5 6 7 8 9
>>> ROOT.myfunction(ROOT.std.vector(int)(range(10)))
0 1 2 3 4 5 6 7 8 9
```

Next Steps

- Investigate test failures
  - Check if there is any fundamental issue
- Understand how ROOT pythonizations could be brought back
  - And new ones added
- Check whether we lost anything else in the migration
- Try with Python 3
- If no blockers are found, decide on how we could integrate Cppyy and PyROOT package-wise

The Code

https://github.com/etejedor/root/tree/cppyy-integration

Backup Slides

Name Changes (PyROOT-Cppyy)

PyROOT	Cppyy
MethodProxy	CPPOverload
ObjectProxy	CPPInstance
PropertyProxy	CPPDataMember
PyROOT	CPyCppyy
PyROOTType	CPPScope
RootModule	CPyCppyyModule
RootWrapper	ProxyWrappers
TCallContext	CallContext

Name Changes (PyROOT-Cppyy) (II)

PyROOT	Cppyy
TClassMethodHolder	CPPClassMethod
TConstructorHolder	CPPConstructor
TCustomPyTypes	CustomPyTypes
TFunctionHolder	CPPFunction
TMemoryRegulator	MemoryRegulator
TMethodHolder	CPPMethod
TSetItemHolder	CPPSetItem
TTupleOfInstances	TupleOfInstances

Other Changes (PyROOT-Cppyy)

- New
  - DeclareConverters, DeclareExecutors, LowLevelViews, PyObjectDir27, TypeManip
- Deleted
  - TPyBufferFactory, TPyDispatcher, TPyFitFunction, TPyROOTApplication, TPySelector
- Modified
  - Converters, Cppyy, Executors, PyCallable, PyStrings, Pythonize, TemplateProxy, TPyArg, TPyClassGenerator, TPyException, TPyReturn, TPython, Utility