Future of ROOT runtime C++ modules

Yuka Takahashi - Princeton University
Vasil Geogiev Vasilev - Princeton University

Available in 6.16!

-Druntime_cxxmodules=ON

Agenda

1. C++ Modules in a nutshell
2. From C++ Modules to Runtime C++ Modules
3. Implementation
4. Benefits - Make ROOT Modular!
5. Benefits - Correctness
6. Status and roadmap

C++ Modules in a Nutshell

C++ Modules in a Nutshell

#include <stdio.h>

- Compile-time scalability
  - Reparse the same header
- Fragility
  - Conflict with local variables

Rcpp.h

```
#define PI 3.1415..
```

Users’ code

```
#include <Rcpp.h>
…
double PI = 3.1415.. // Screwed
```

C++ Modules in a Nutshell

Modules

- Header information stored in precompiled PCM files.
- Lazily loading AST information

➡ No more header parsing during ROOT's runtime.

✓ Compile-time scalability

✓ Fragility

From C++ Modules to Runtime C++ Modules

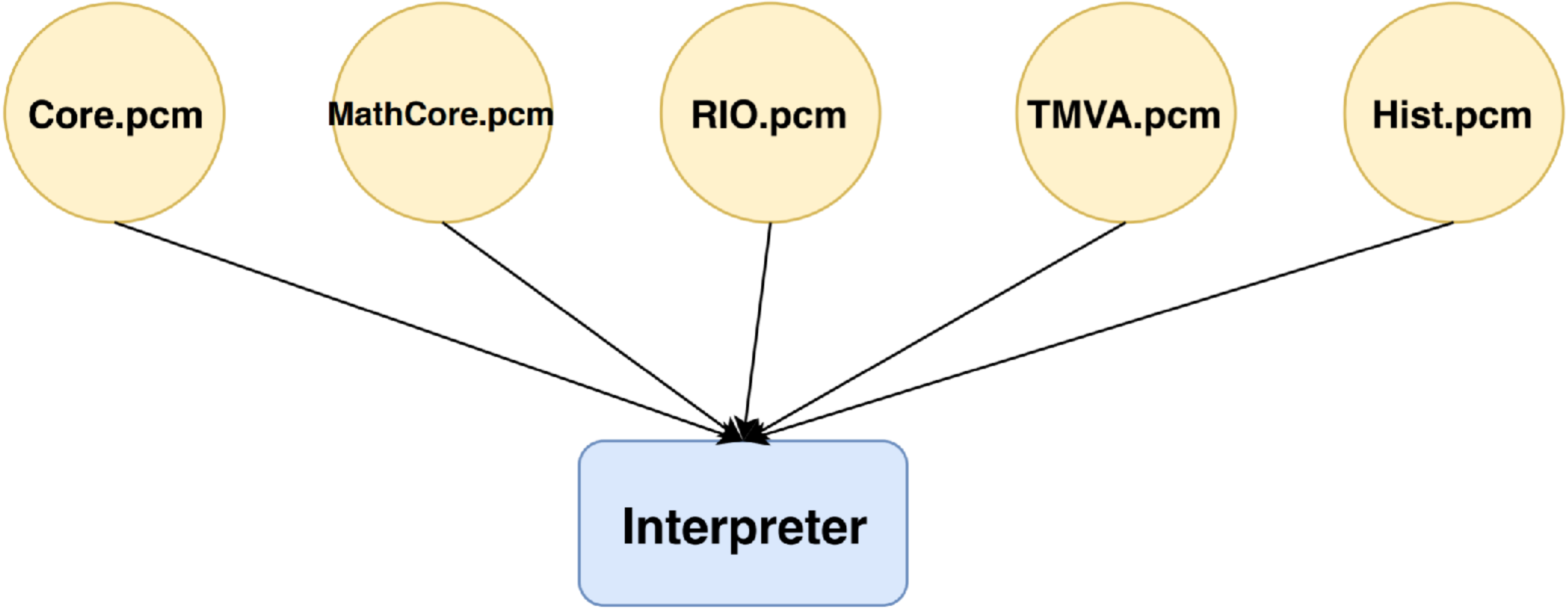
From C++ Modules to Runtime C++ Modules

C++ Modules is a mechanism to boost compilation time

➡ Turns into **Runtime** for ROOT as we have C++ interpreter

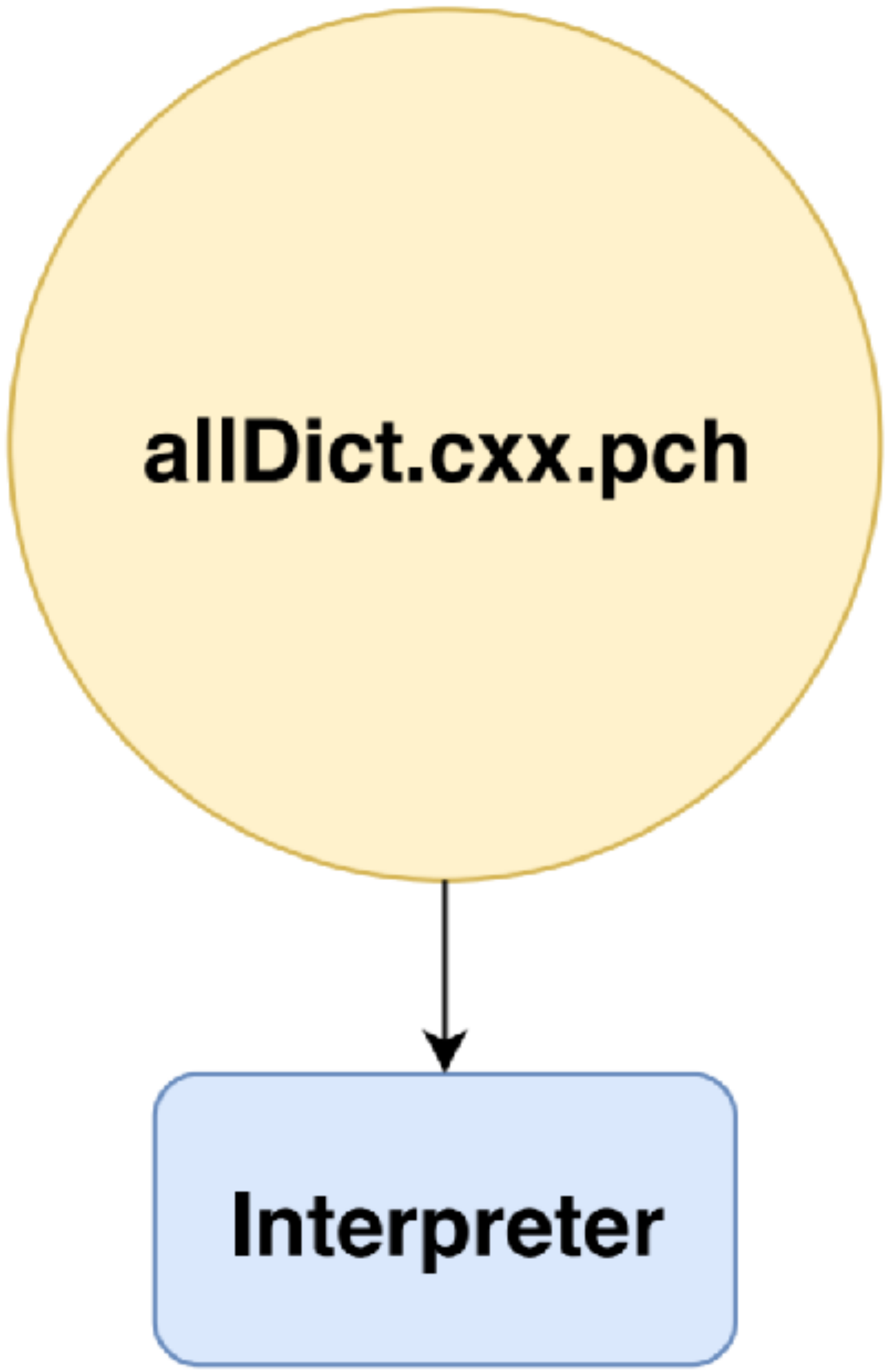
From C++ Modules to Runtime C++ Modules

Each PCM file (e.g. Core.pcm) corresponds to a library (e.g. libCore.so)



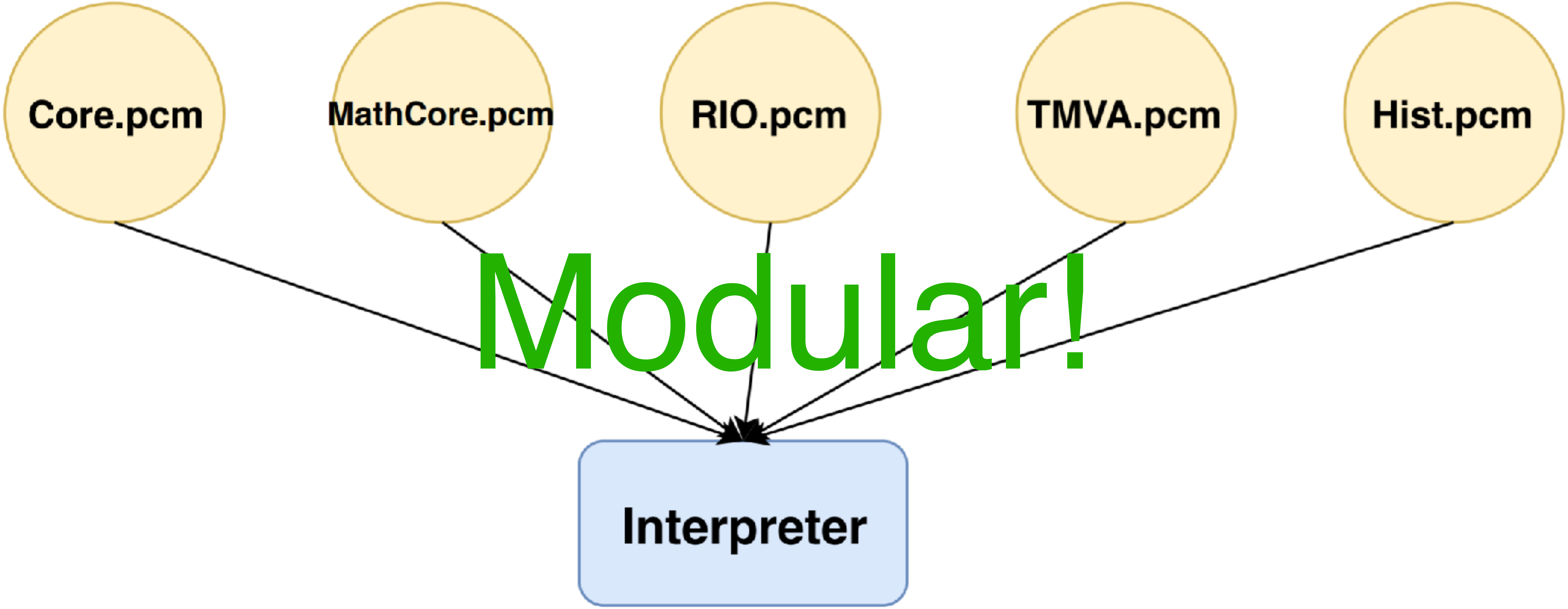
From C++ Modules to Runtime C++ Modules

PCH files are precompiled header files and work similar to C++ modules.



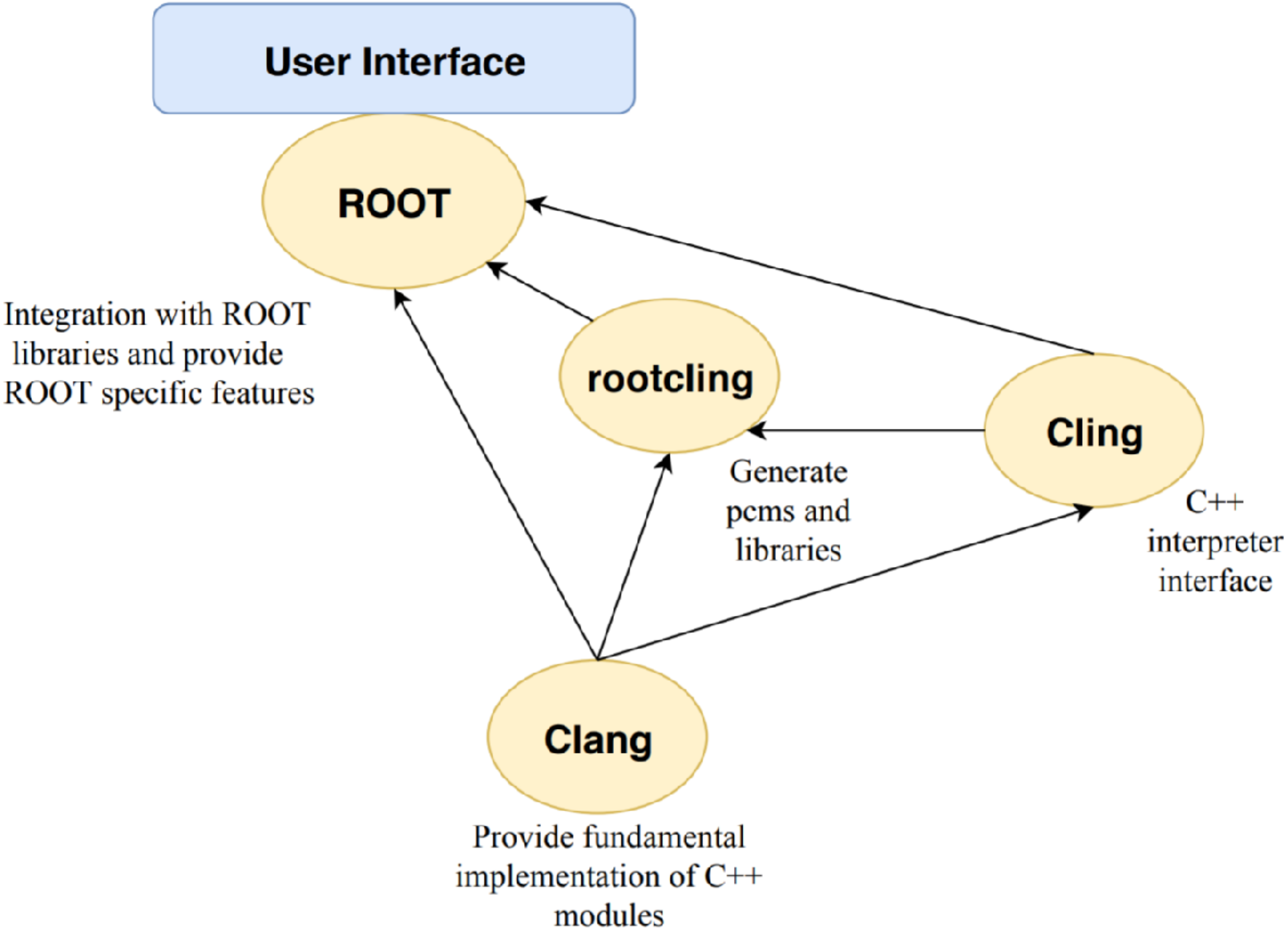
From C++ Modules to Runtime C++ Modules

Each PCM file (e.g. Core.pcm) corresponds to a library (e.g. libCore.so)



Implementation

Implementation



User Interface

ROOT

Integration with ROOT libraries and provide ROOT specific features

rootcling

Generate pcms and libraries

Cling

C++ interpreter interface

Clang

Provide fundamental implementation of C++ modules

Benefits
Make ROOT Modular!

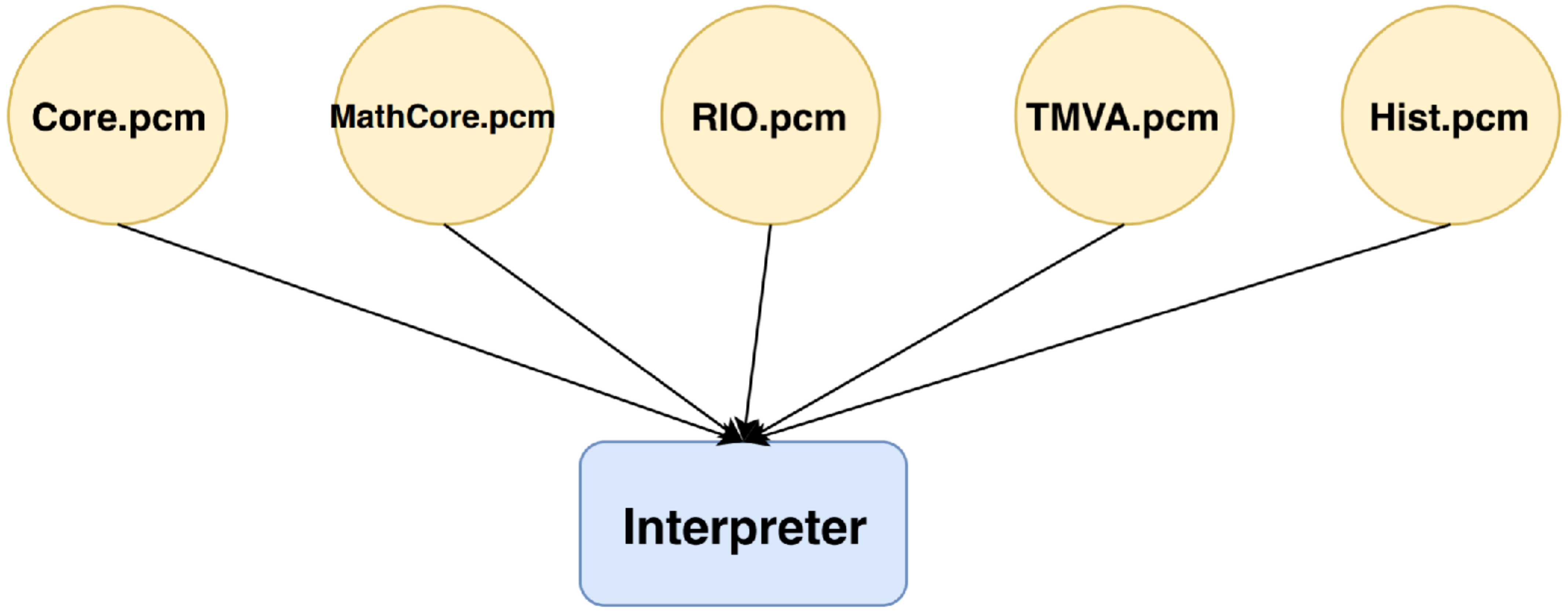
Benefits - Make ROOT Modular!

Modularise experiments

Still using texual includes (Not even PCH!)
PCH cannot be used for technical constraints

Working with CMS to introduce C++ modules into CMSSW

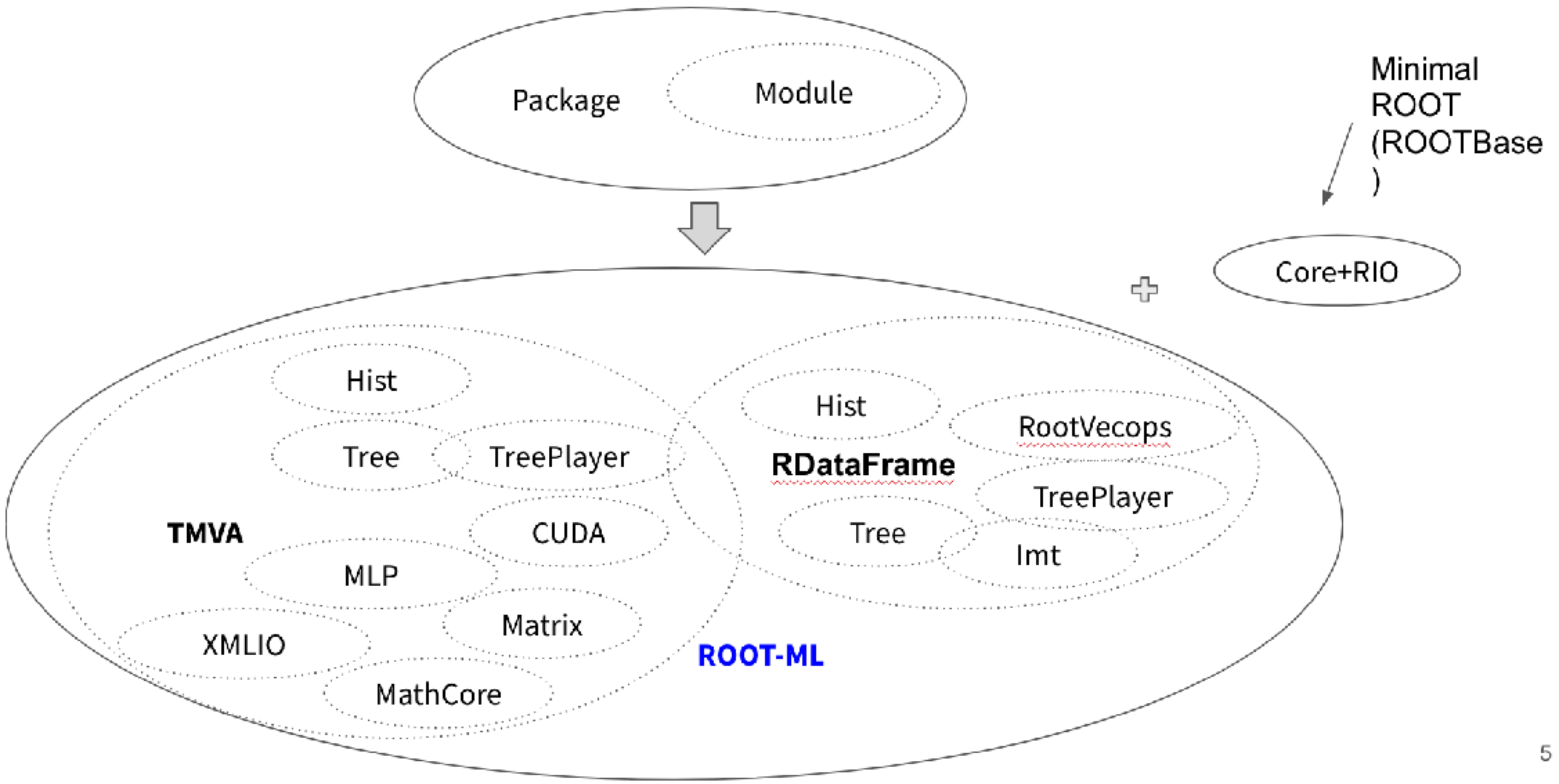
#include <TH1.h>



Benefits - Make ROOT Modular!

We can make ROOT modular for lazy installing packages

- ROOT package manager
- See Oksana's talk for more information!



Benefits - Correctness

Benefits - Correctness

Without Modules

```
$ bin/root.exe -l
root [0] gMinuit // Cannot load variable
IncrementalExecutor::executeFunction:
symbol 'gMinuit' unresolved while
linking [cling interface function]!
```

Benefits - Correctness

With Modules

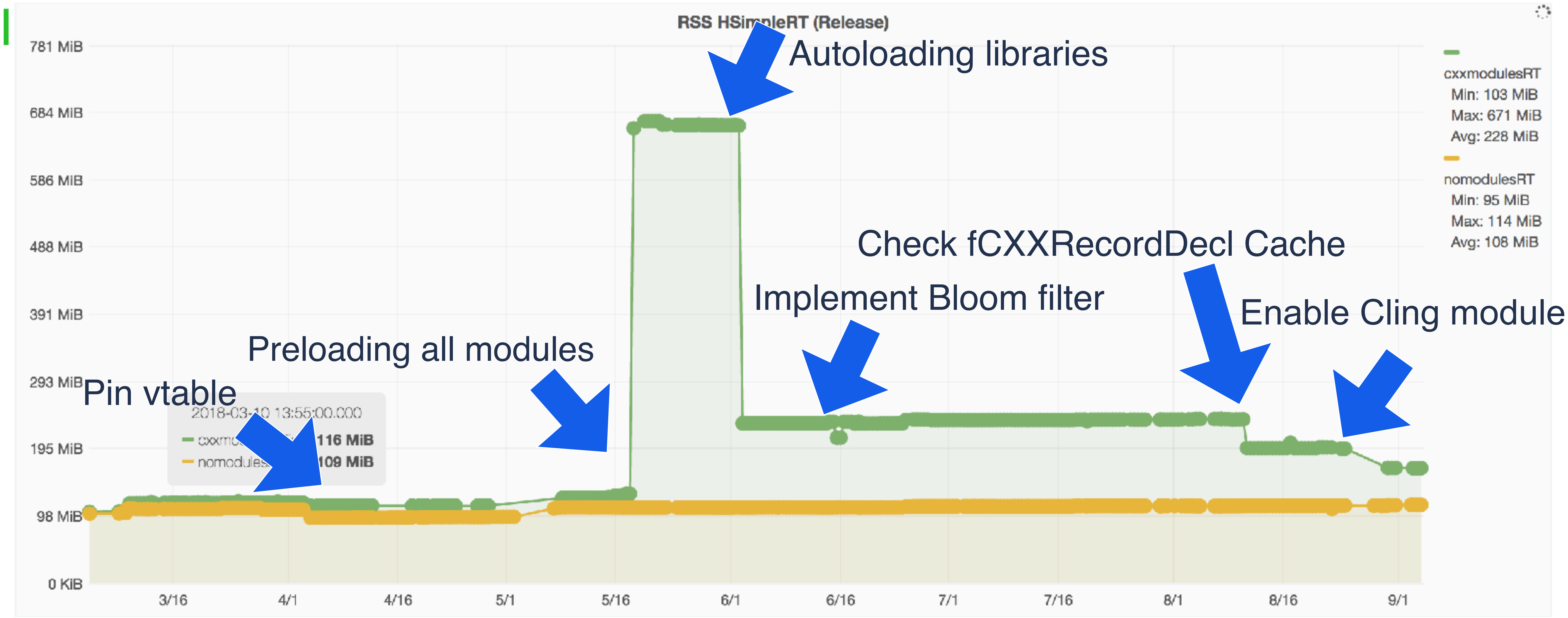
```
$ bin/root.exe -l
root [0] gMinuit // Could load libMinuit
(TMinuit *) nullptr
```

Status and roadmap

Status and roadmap

Memory - hSimple

https://rootbnch-grafana-test.cern.ch/



Autoloading libraries

Check fCXXRecordDecl Cache

Implement Bloom filter

Enable Cling modulemap

Preloading all modules

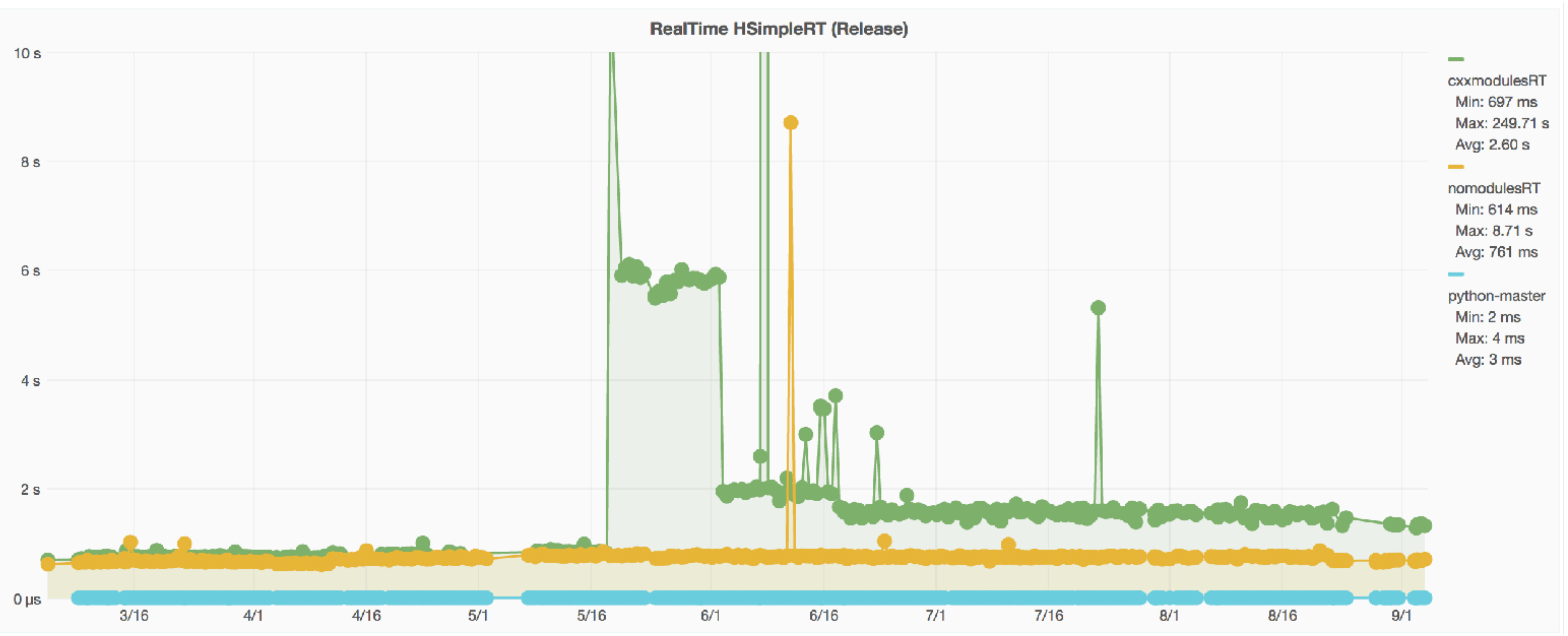
Pin vtable

…. And More!!

Status and roadmap

Real time - hSimple

https://rootbnch-grafana-test.cern.ch/



Status and roadmap

Status

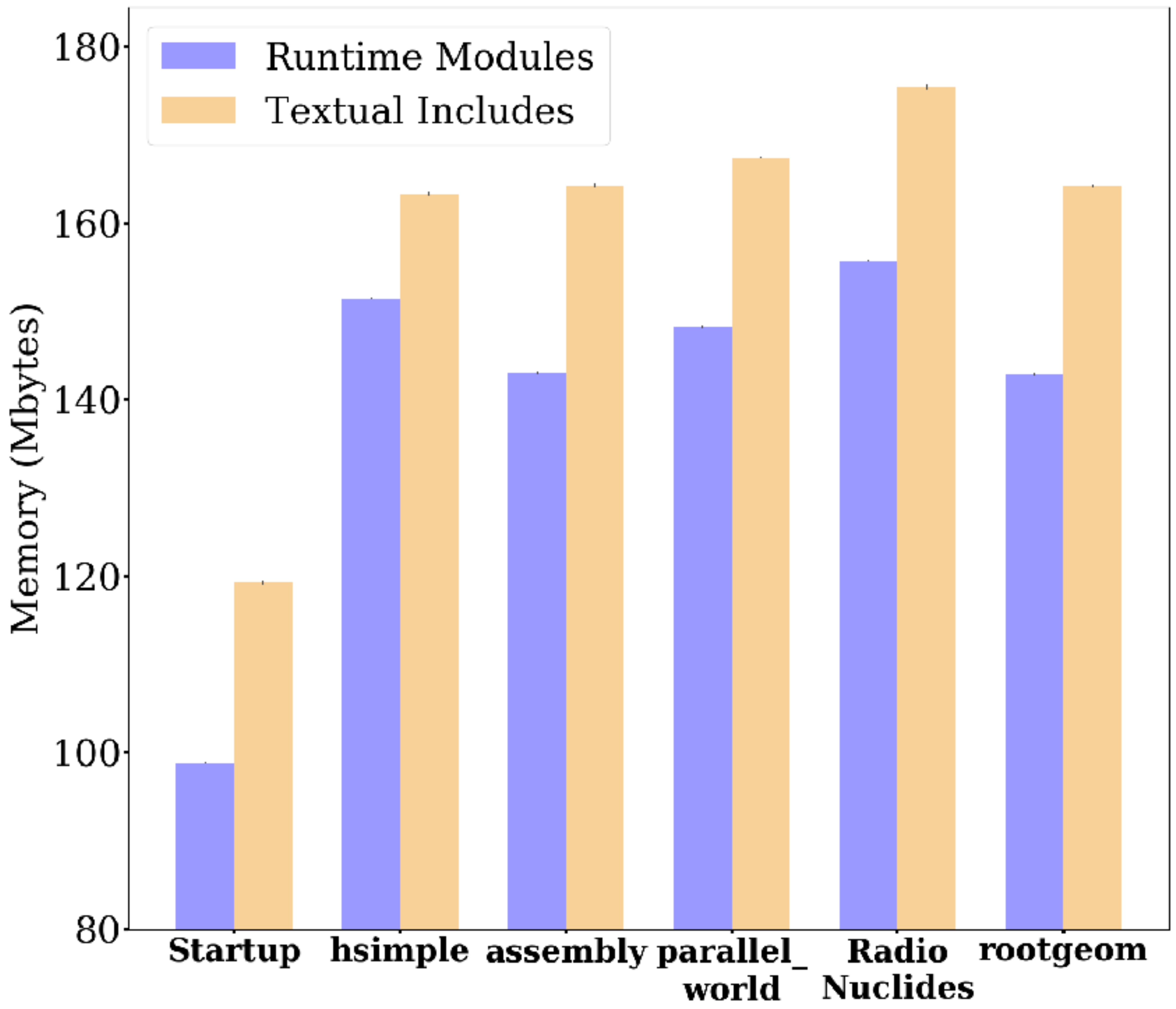
- Fundamental Construction in ROOT Core, which effects every code passed to ROOT
- Working with industry and CMSSW
- Good progress in performance optimization

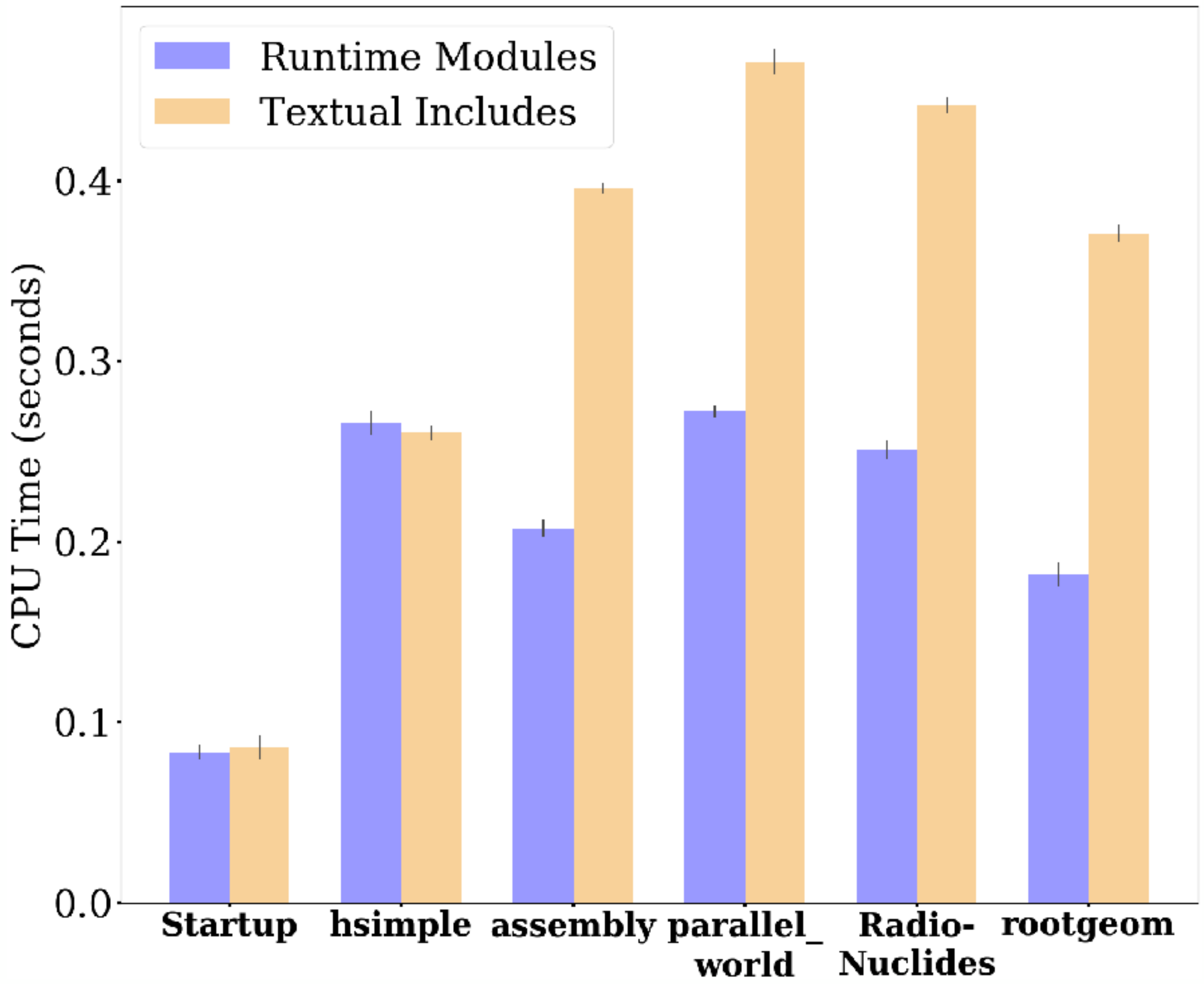
Roadmap

- Reach complete production level before 6.16
- Continue working on optimization
- Modularise CMSSW!

Thank you for your attention!

Backup Slides


Runtime Modules
Textual Includes
Memory (Mbytes)
180
160
140
120
100
80
Startup
hsimple
assembly
parallel_ world
Radio Nuclides
rootgeom


Runtime Modules
Textual Includes
CPU Time (seconds)
0.4
0.3
0.2
0.1
0.0
Startup
hsimple
assembly
parallel_ world
Radio- Nuclides
rootgeom