



## ROOT-free *art*

9 August 2018

*art* stakeholders meeting



# Definitions

By ROOT-free *art*, I mean that **core framework functionality** would not depend on ROOT—i.e. an *art* user could run an *art* job with no dependence on ROOT whatsoever. **This is not currently possible:**

- The *art* executable (and *lar*, *mu2e*, *nova*, *gm2*) until now has linked either directly or transitively against ROOT libraries.

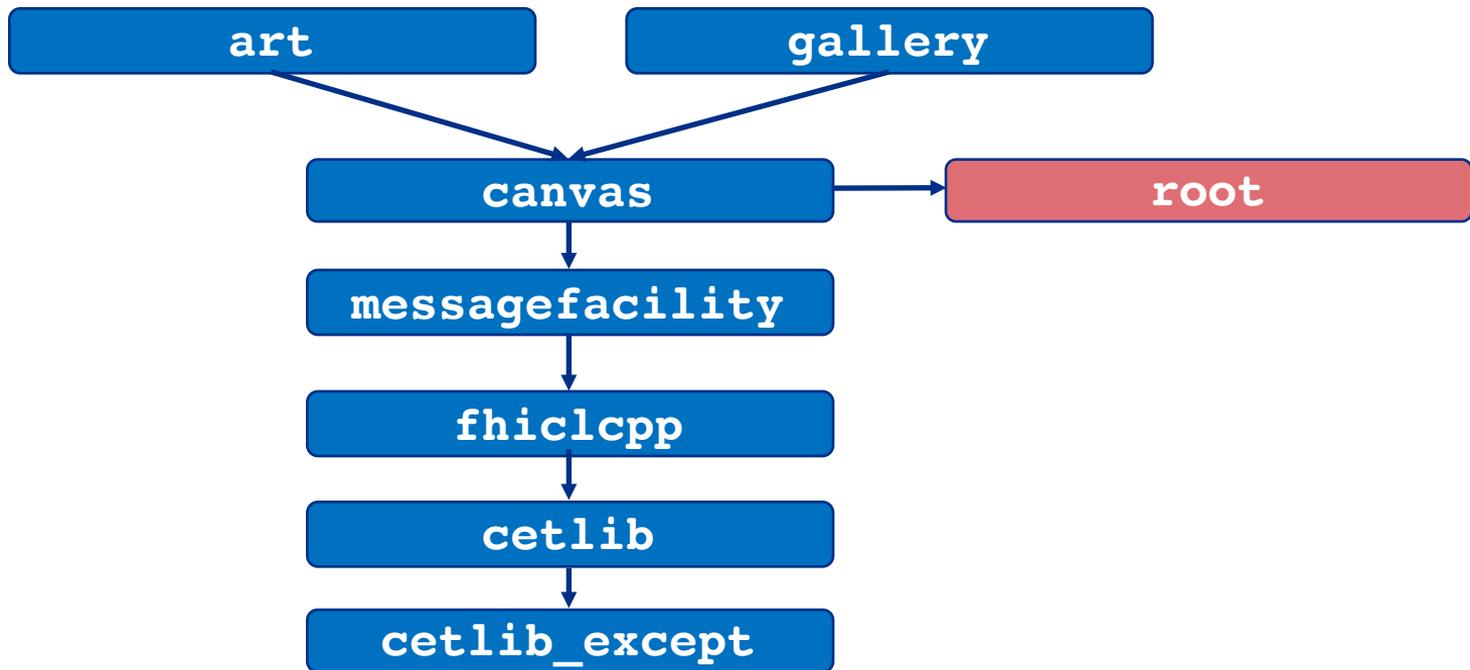
By ROOT-free *art*, I do **not** mean that we (the *art* project) will no longer provide functionality for interacting with ROOT.

# Motivation for a ROOT-free *art*

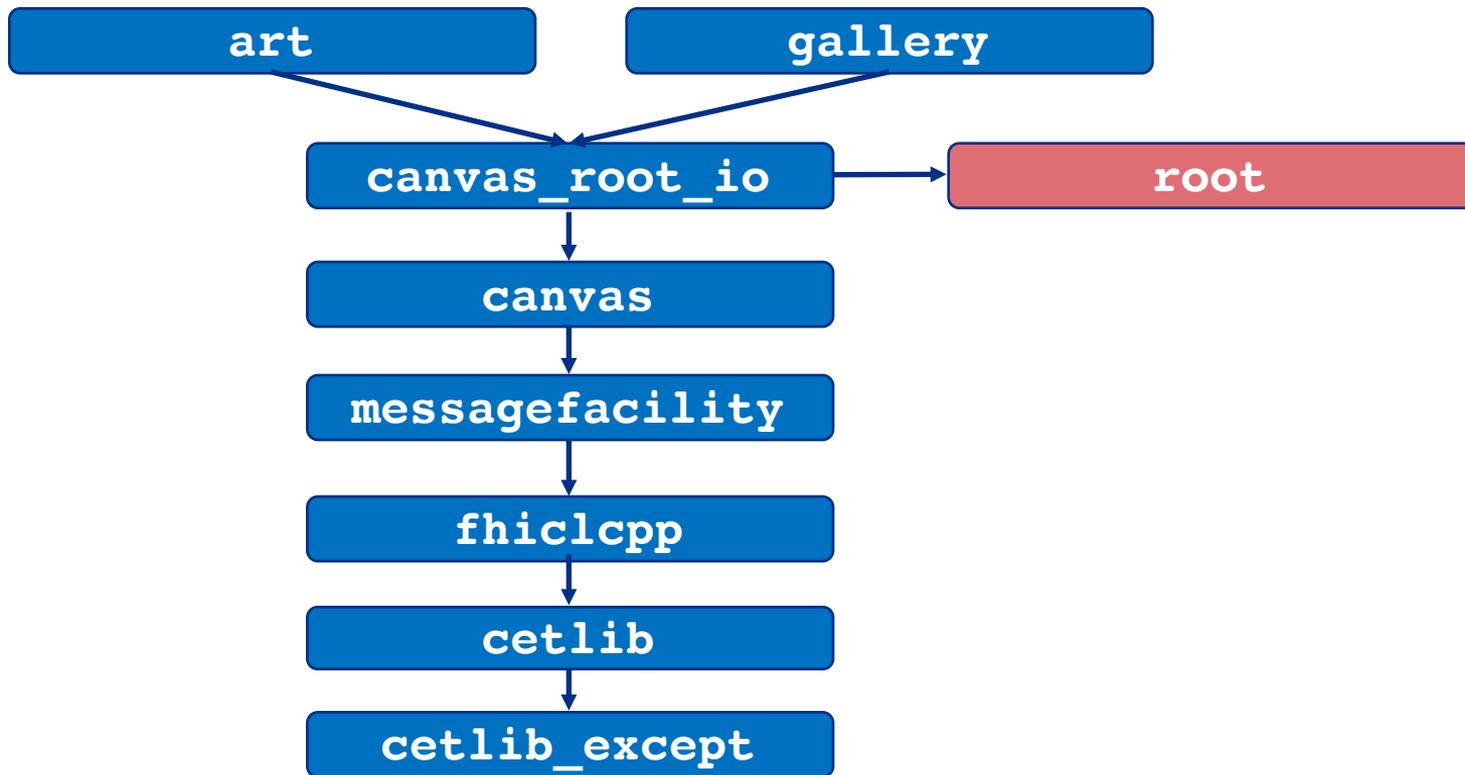
- Changing ROOT versions would not necessitate *art* version changes → simpler release process.
  - Fewer binaries need to be installed (don't need to reinstall many of the *art* libraries that do not change because of a ROOT change).
- Experiments could change ROOT versions without changing *art* versions, and vice versa.
- Enables you to consider other I/O formats.

# What work has been done so far?

## art dependencies (Sept. 2017 before)

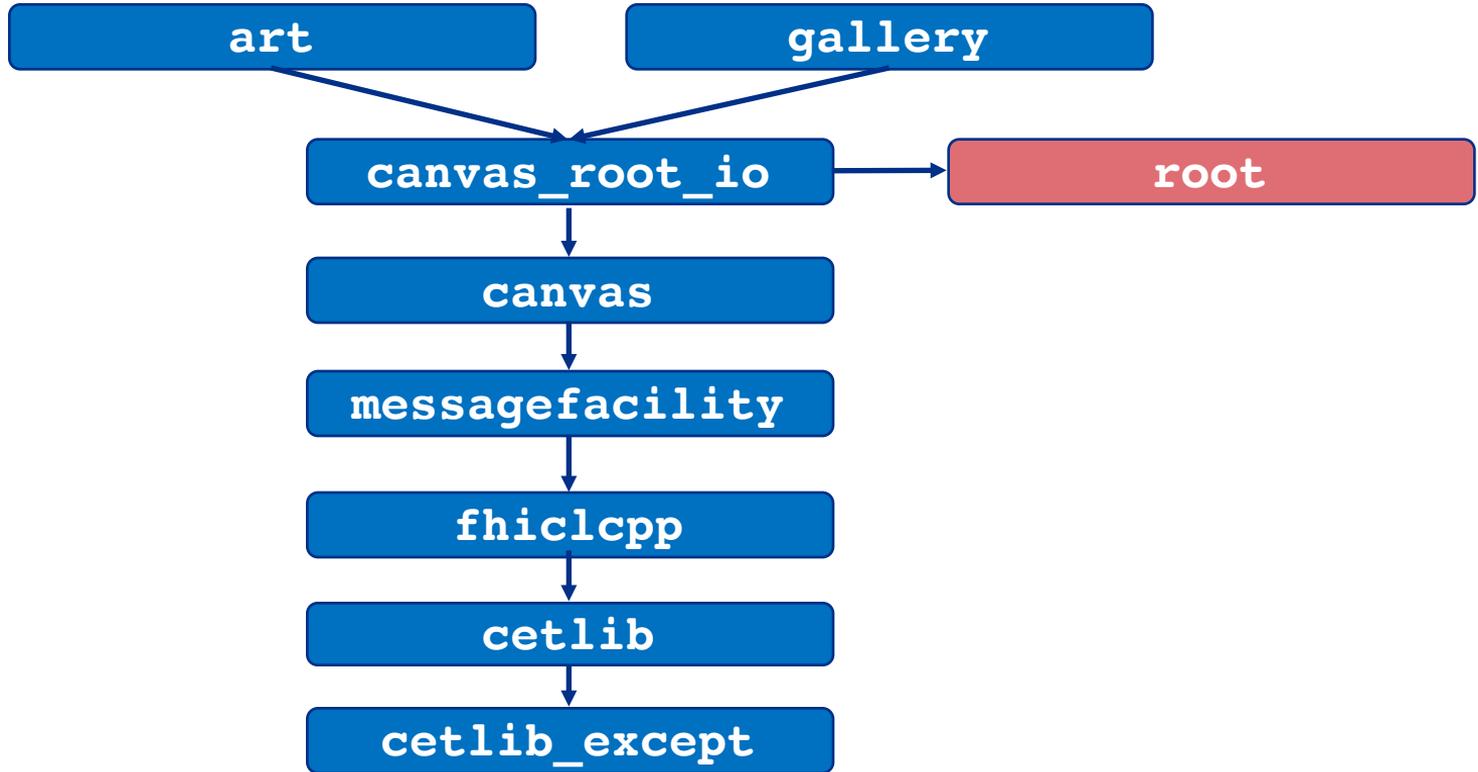


# art dependencies (Sept. 2017 after)

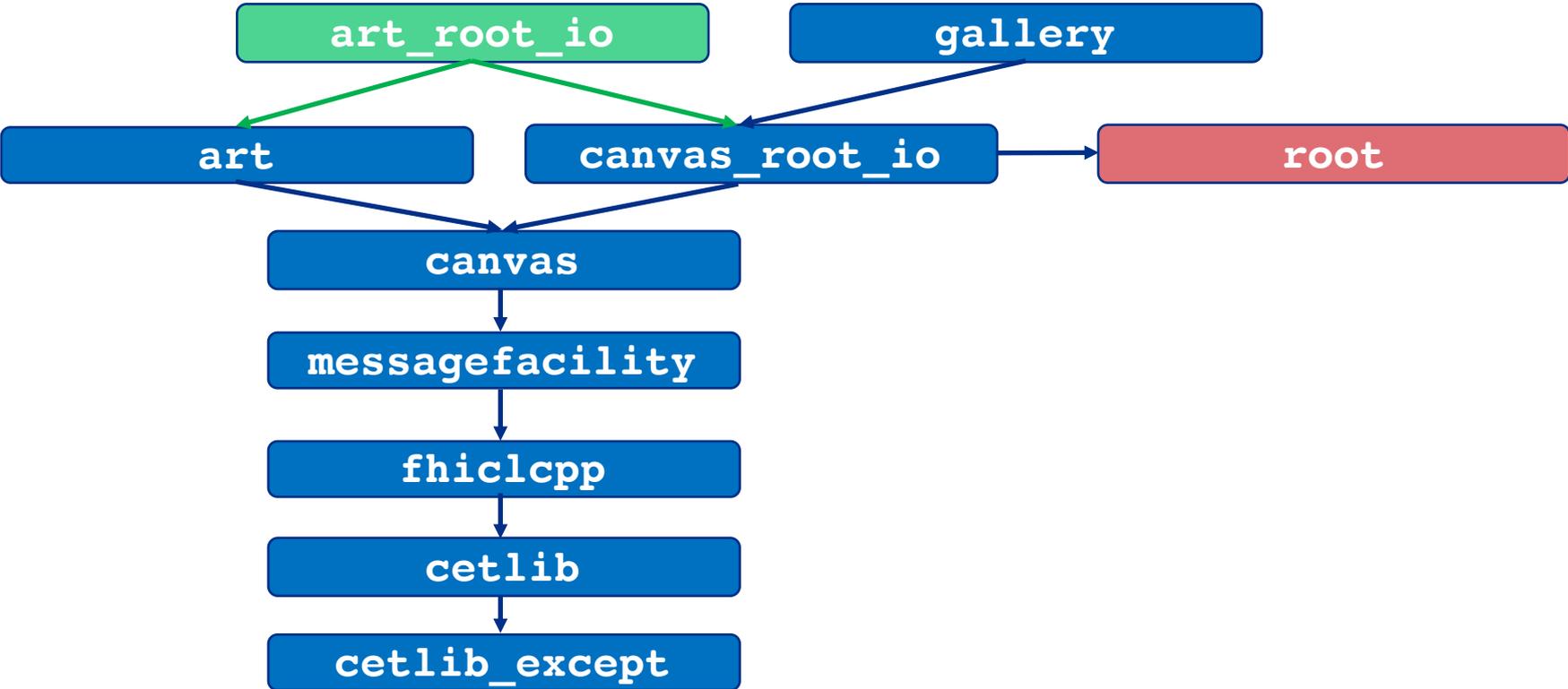


# The proposal

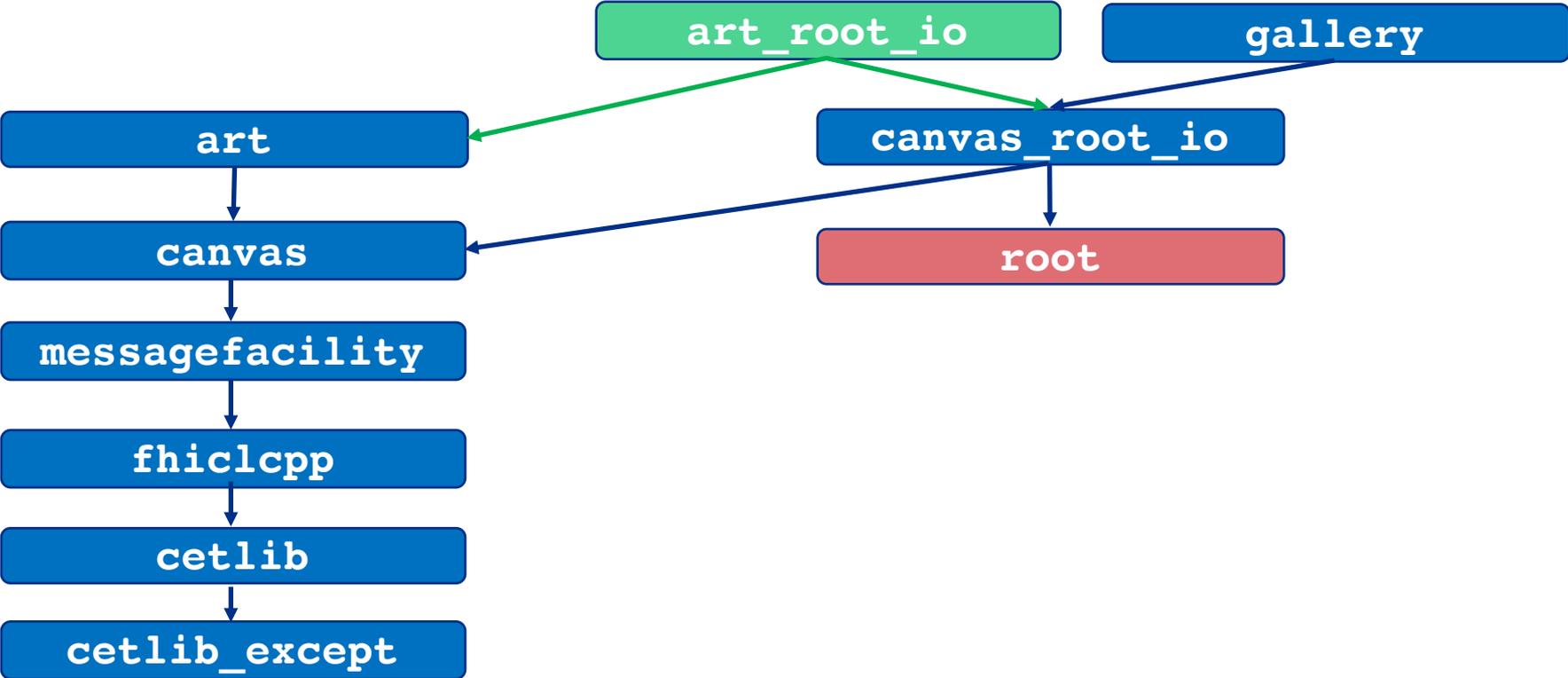
# art dependencies (current)



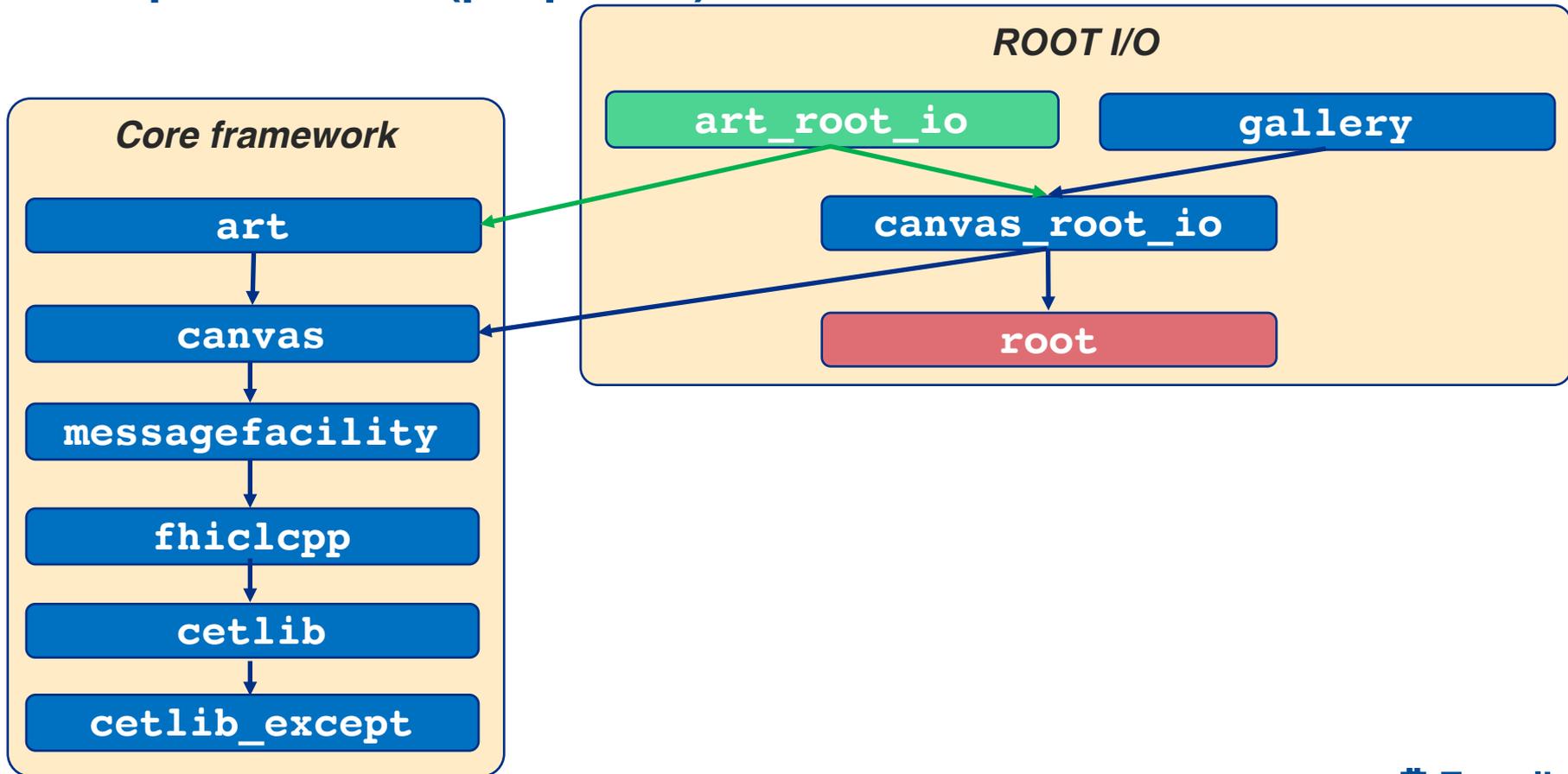
# art dependencies (proposed)



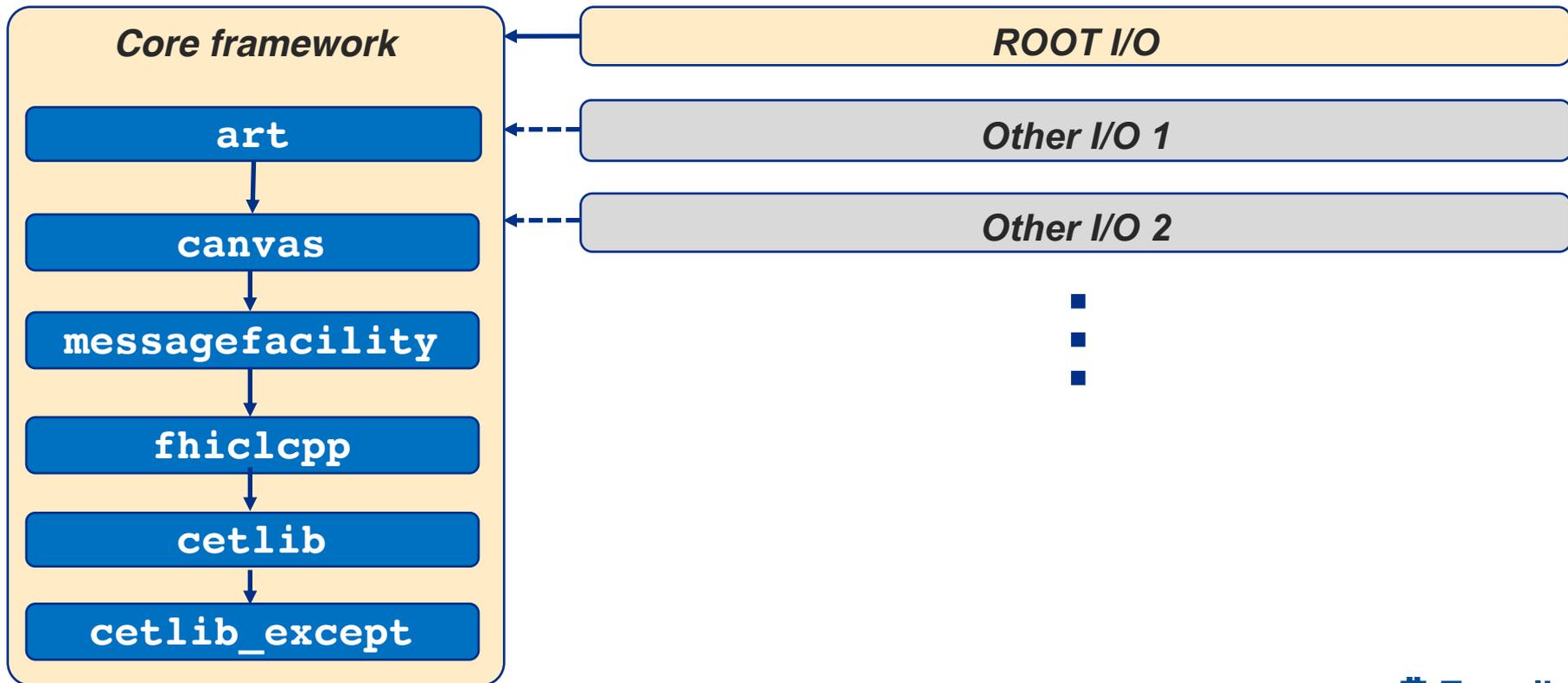
# art dependencies (proposed)



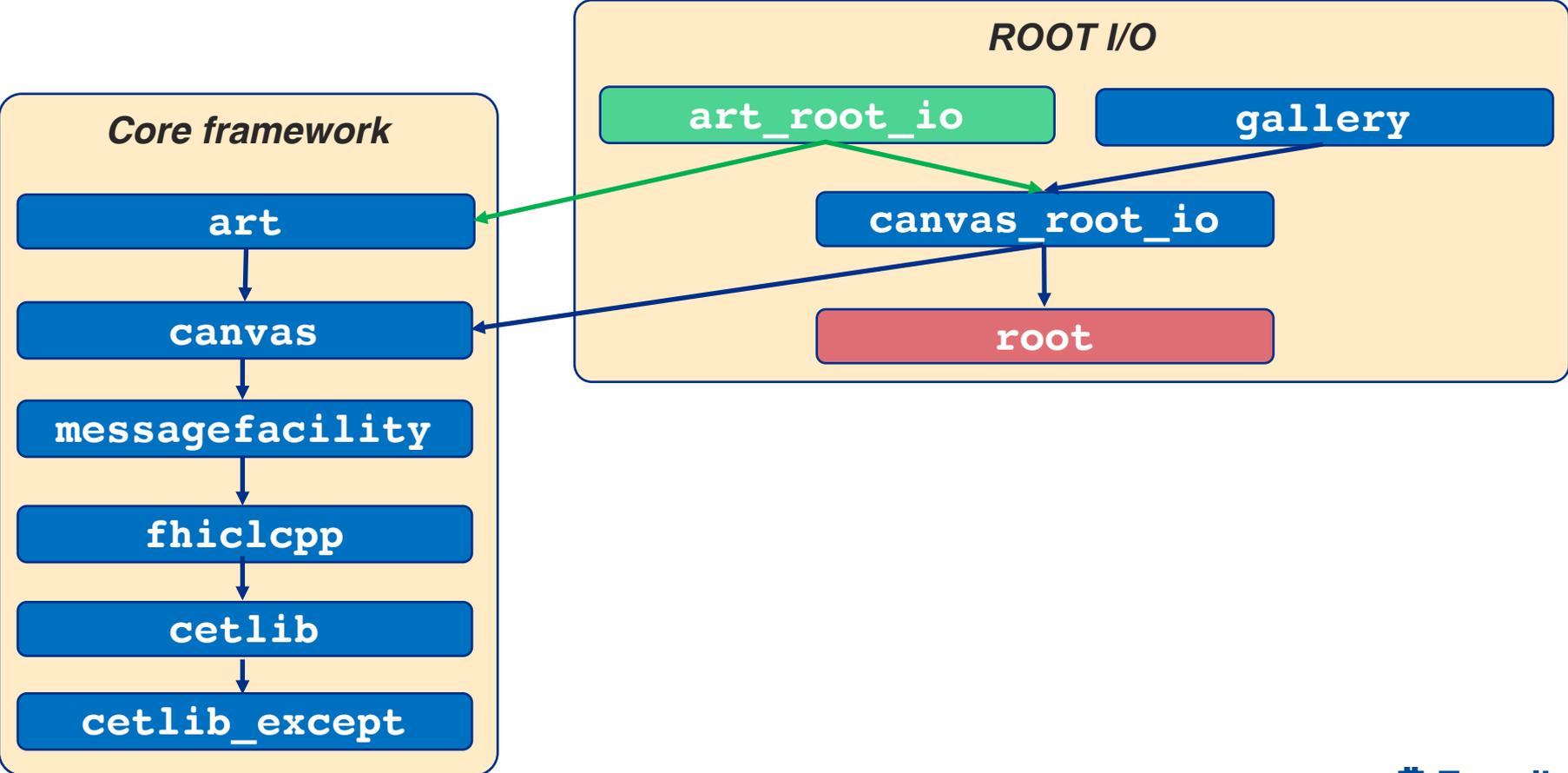
# art dependencies (proposed)



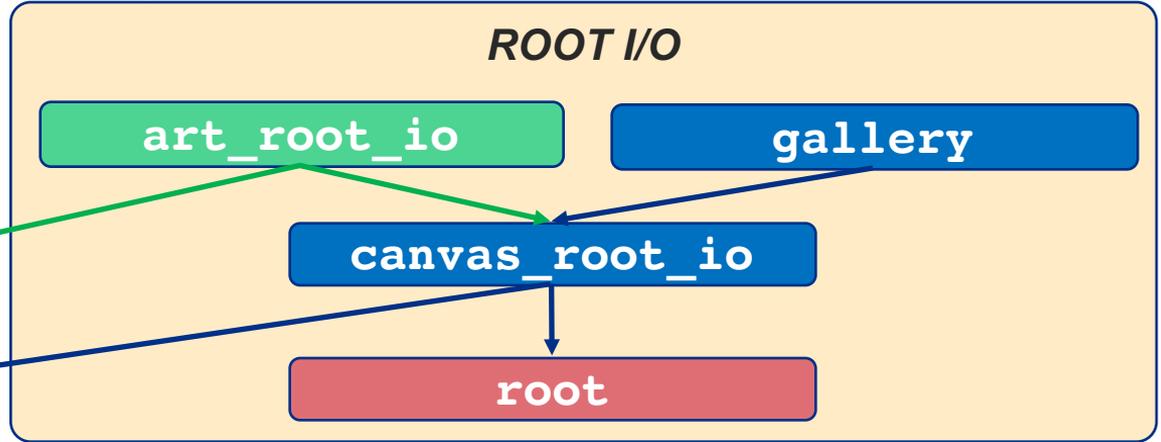
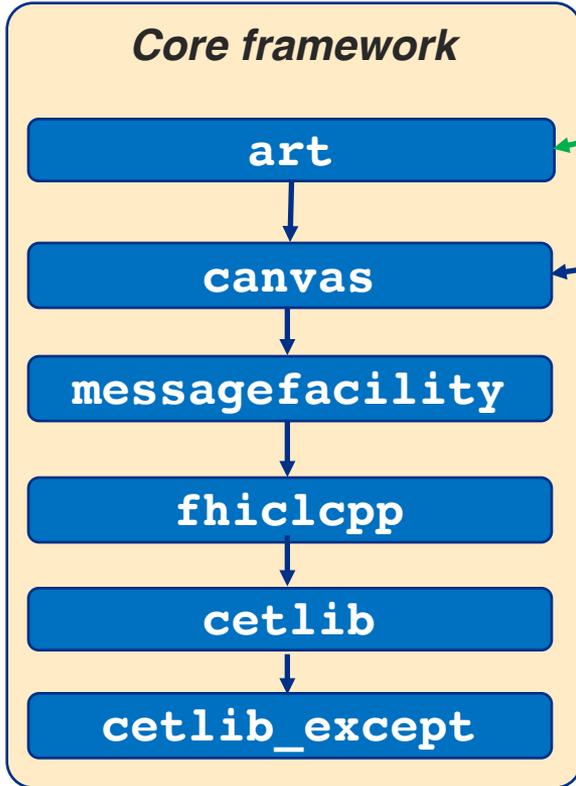
# art dependencies (future)



# art dependencies (proposed)

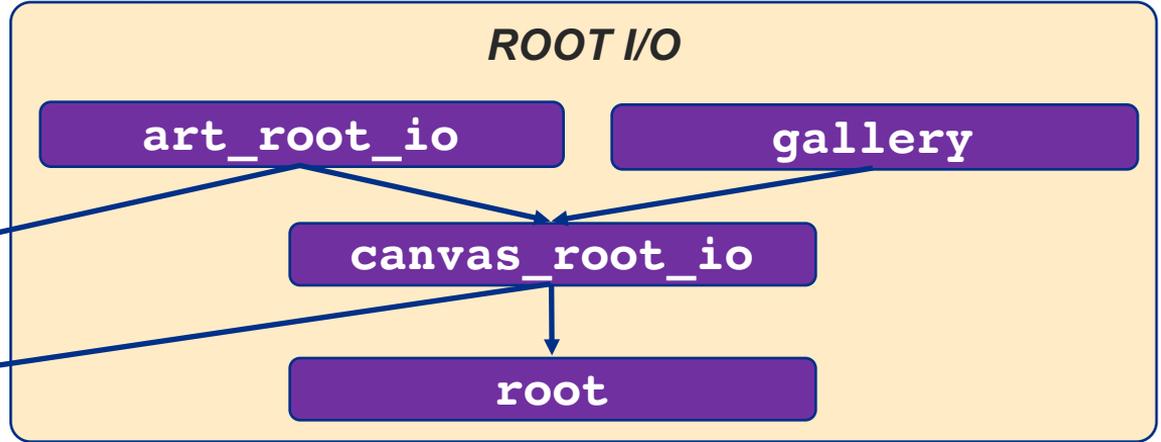
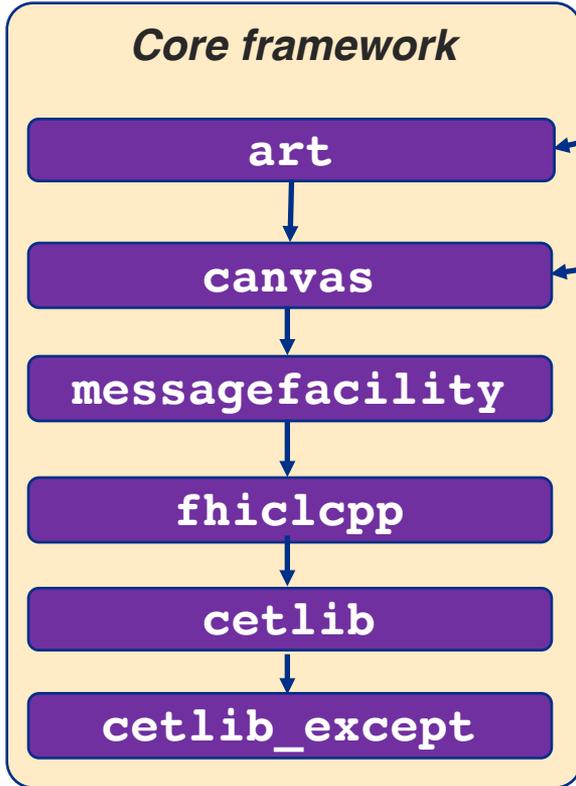


# art dependencies (proposed)



To provide easier use for package maintainers, we will introduce an umbrella UPS Product (placeholder name of *art\_suite*).

# art dependencies (proposed)



To provide easier use for package maintainers, we will introduce an umbrella UPS Product (placeholder name of *art\_suite*).

```
setup -B art_suite v1_03_01 -q +e17:+prof  
➔ setup -B art_root_io v1_02_01 -q +e17:+prof  
➔ setup -B gallery v2_04_03 -q +e17:+prof
```

# What changes would be noticeable to users?

- New location for TFileService/TFileDirectory (e.g.)
  - `#include "art/Framework/Services/Optional/TFileService.h"`
  - + `#include "art_root_io/Services/TFileService.h"`
- For MixHelper to provide I/O agnostic behavior, some interface changes will be required.
- The `unloadRootSigHandler` and `resetRootErrorHandler` scheduler options will be removed (in fact, they were removed in *art 3*).
- Locations of `RootInput` and `RootOutput` will change...but that will be invisible to you, unless you specify fully-qualified module types (e.g.):

```
module_type: "art/Framework/I0/RootInput"
```

# When would this split happen?

- Possibly as early as *art* 3.01.