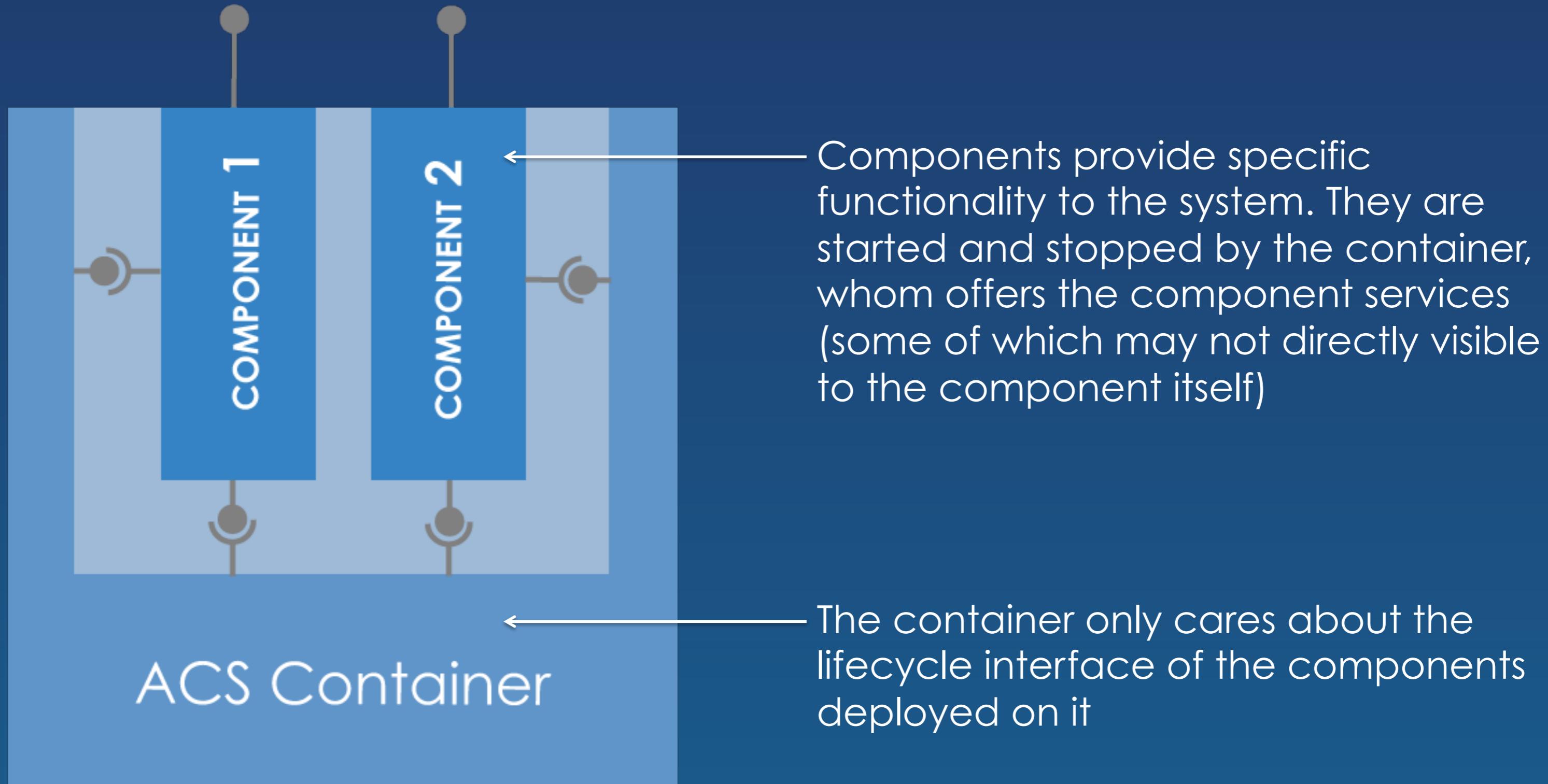


# ALMA Common Software Basic Track

Component/Container Model and Lifecycle Management

# Container/Component





# Component

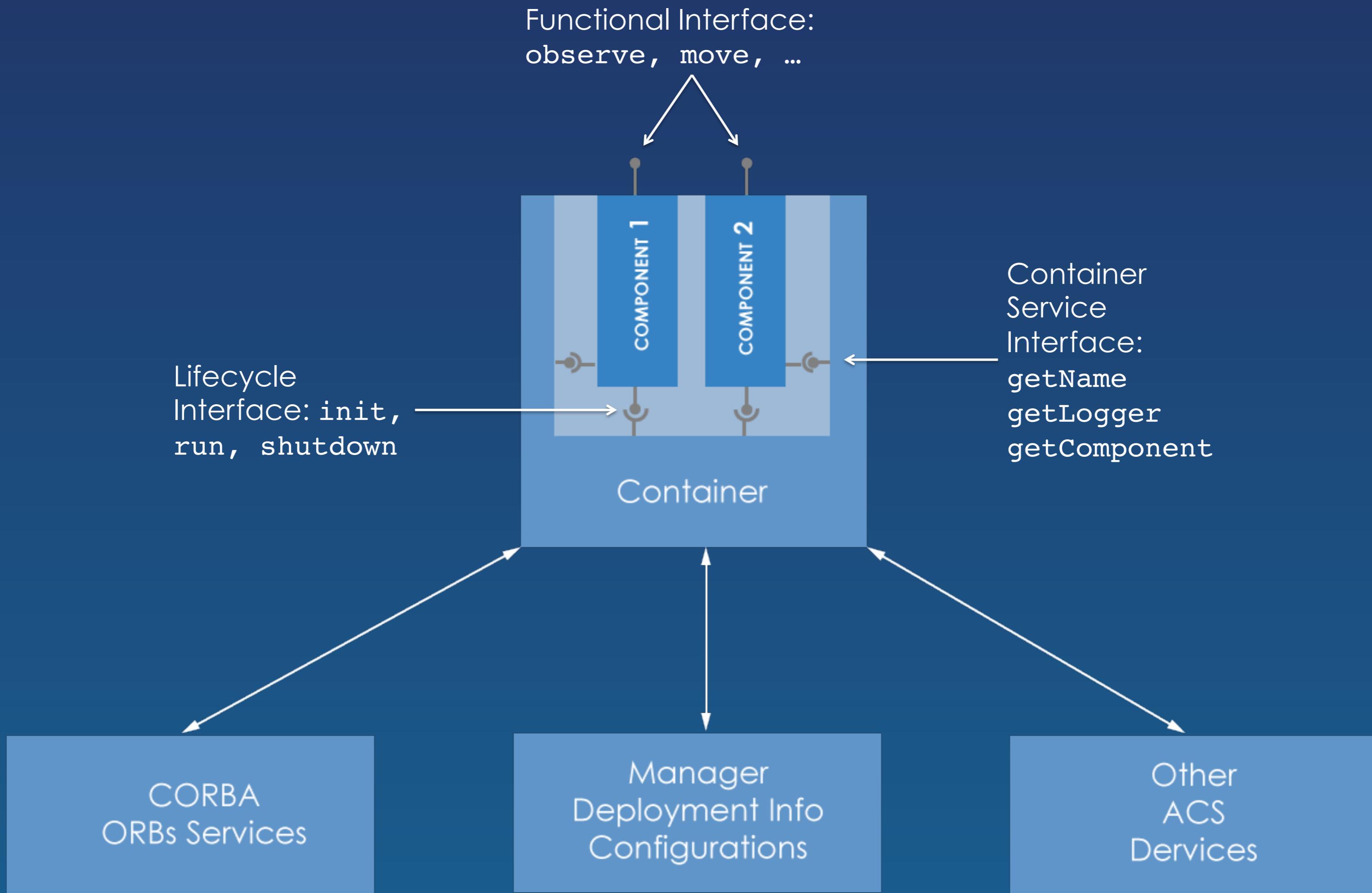
- ✧ Deployable unit of ALMA software
  - ✧ Same concept from device level to data flow application
  - ✧ 1...many classes per component
  - ✧ 1...many components per subsystem
- ✧ Functional interface defined in CORBA IDL
- ✧ Deployed within a Container
- ✧ Well-defined lifecycle (initialization, finalization)
- ✧ Accessible as a plain CORBA object if required
- ✧ Focus on functionality with little overhead for remote communication and deployment
- ✧ Concept promotes modular and decoupled application code



## Container

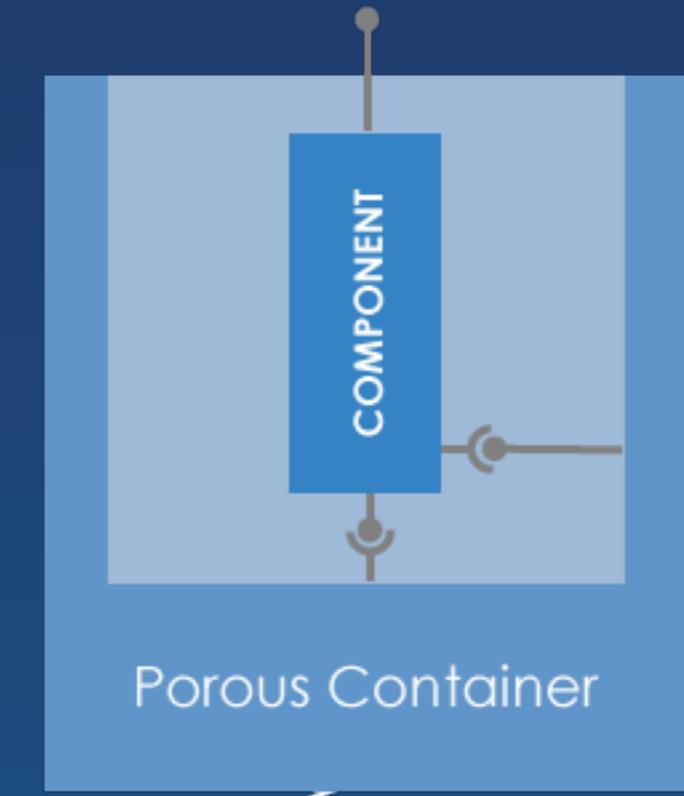
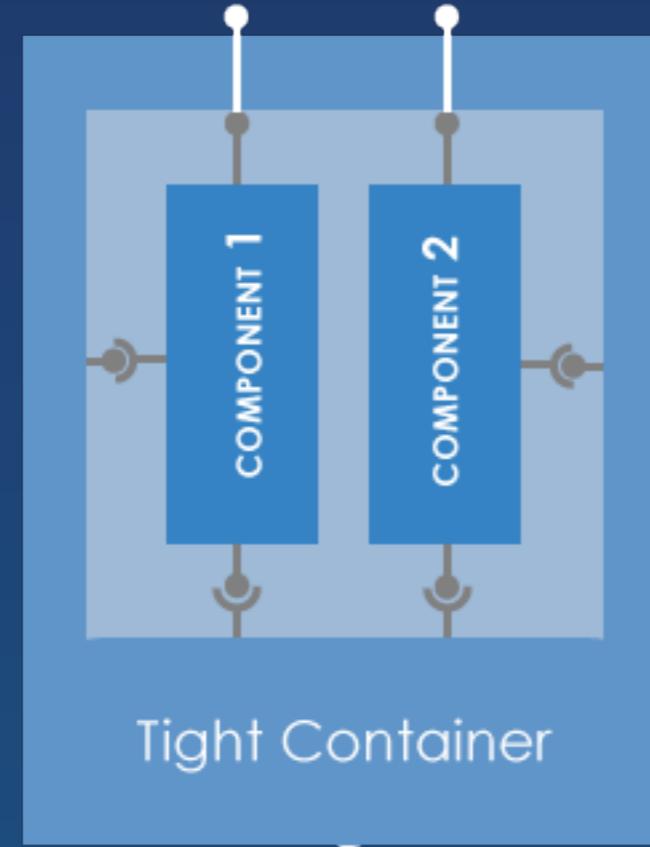
- ✧ Centrally handles technical concerns and hides them from application developers
  - ✧ Deployment, Start-up
  - ✧ Selection and configuration of various ORBs; here CORBA alone is much too complicated.
  - ✧ Selection of CORBA Services, integration with ACS Services (error, logging, configuration, ...)
  - ✧ Convenient access to other components and resources
- ✧ New technical aspects can be integrated in the future, without modifying the application software

# Container/Component Interfaces



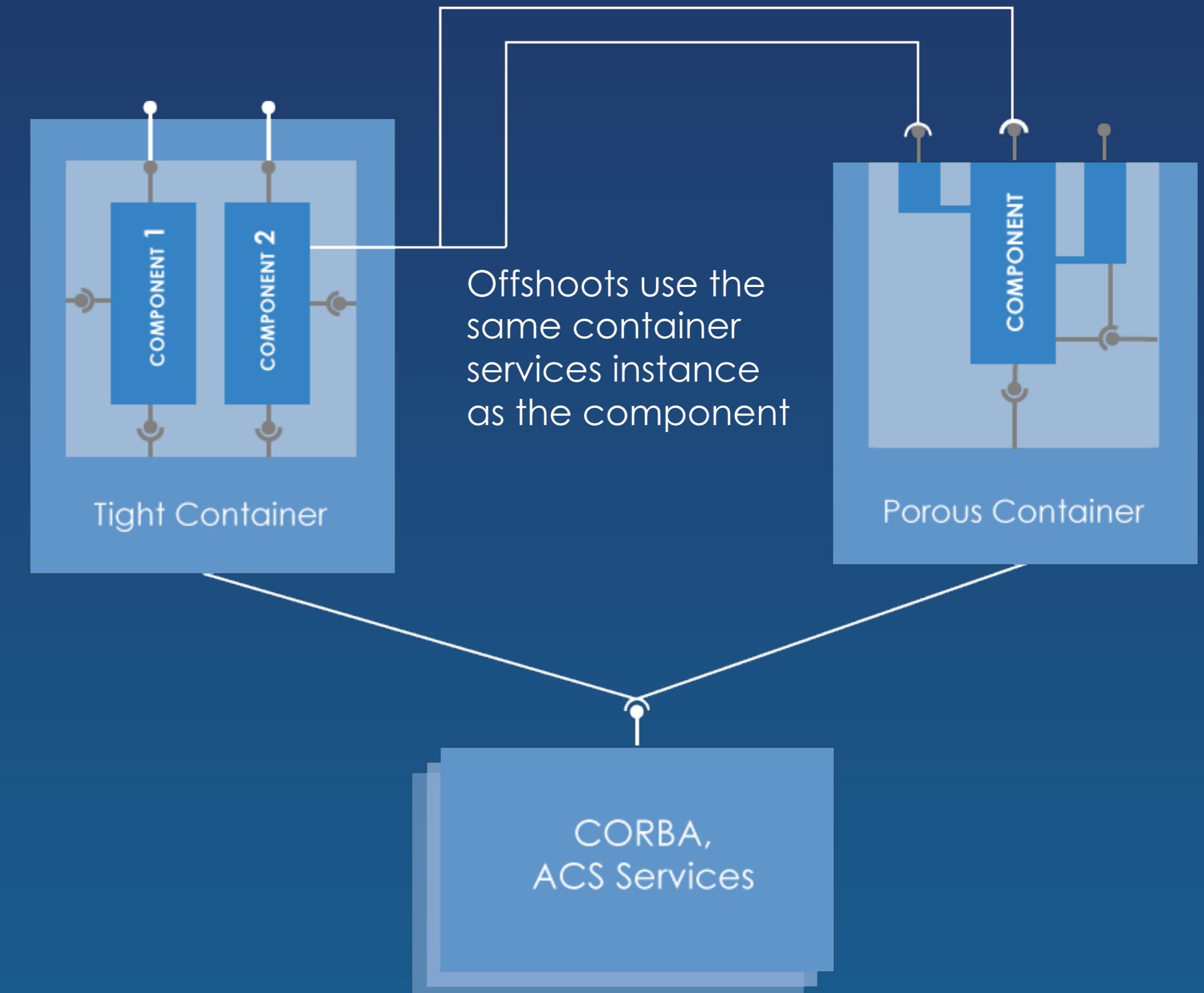
# Tight versus Porous Containers

Functional interface  
is intercepted by  
the container for  
logging and/or  
exception handling,  
security, ...

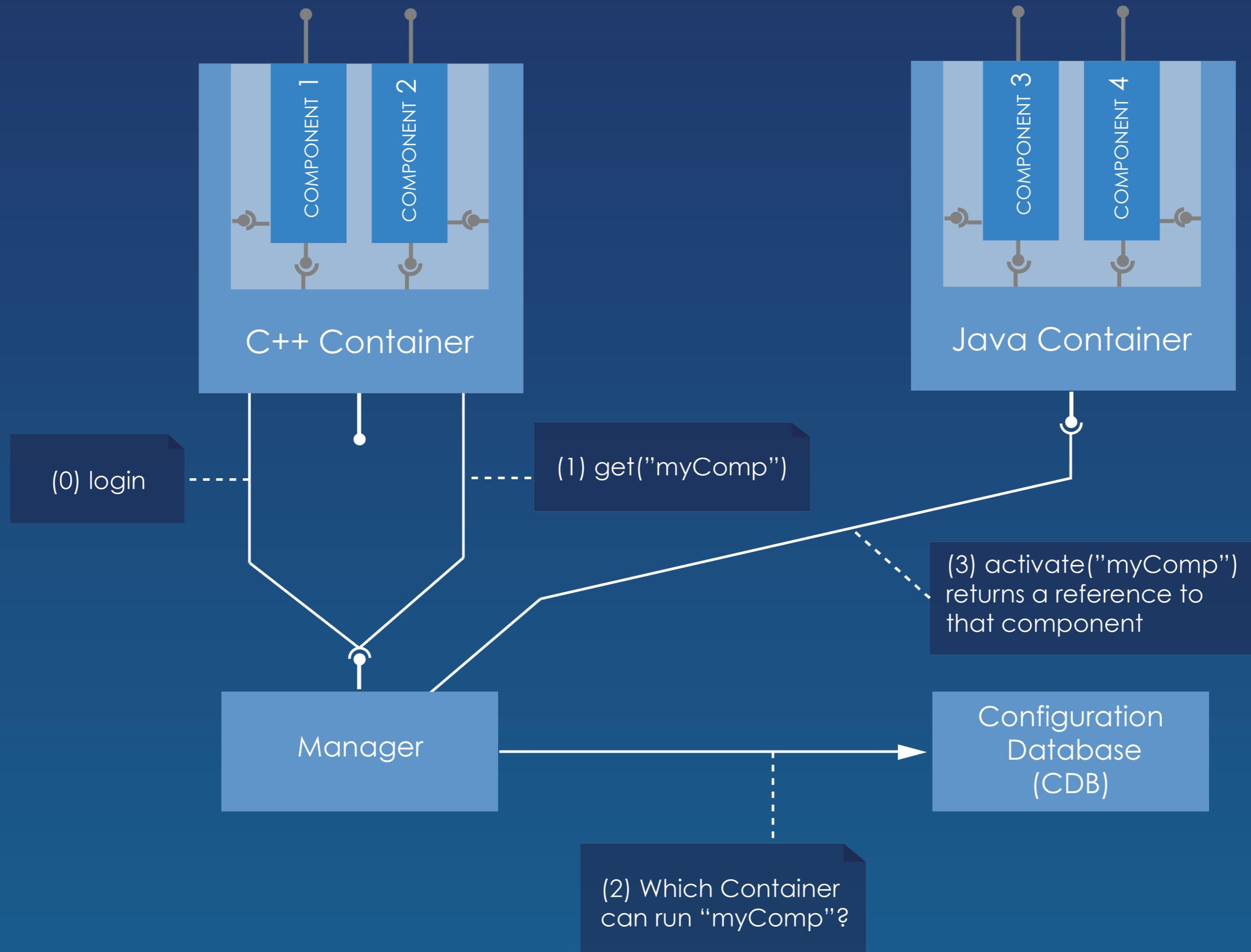


# Component Offshoots

- ✧ Remotely visible object created by a component
- ✧ Life is limited to that of the component
- ✧ Offshoots are conceptually “in between” components and programming-language-specific objects



# Interactions: component activation and retrieval



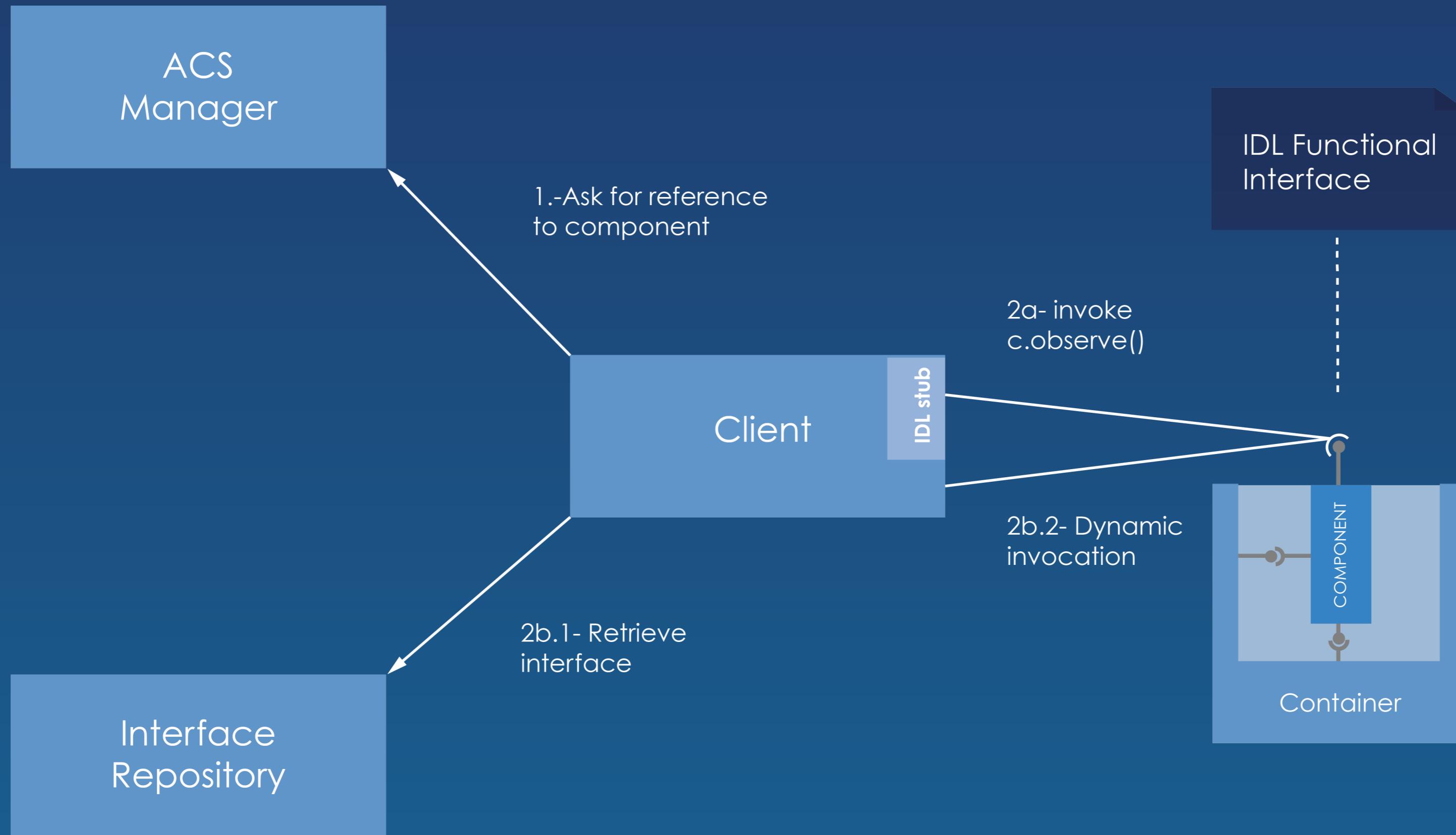
# Interactions: component activation and retrieval

- ✧ Manager and CDB (with deployment info) are running
- ✧ Containers are logged in to the Manager with their names, f.i. “Telescope”
- ✧ We assume that one component is running already...
- ✧ The Component requests a reference to another Component from its Container
- ✧ Container asks Manager for that other Component
- ✧ Manager asks CDB which Container hosts the Component (can be the same container as before or a different one)
- ✧ Manager tells Container to load the new Component

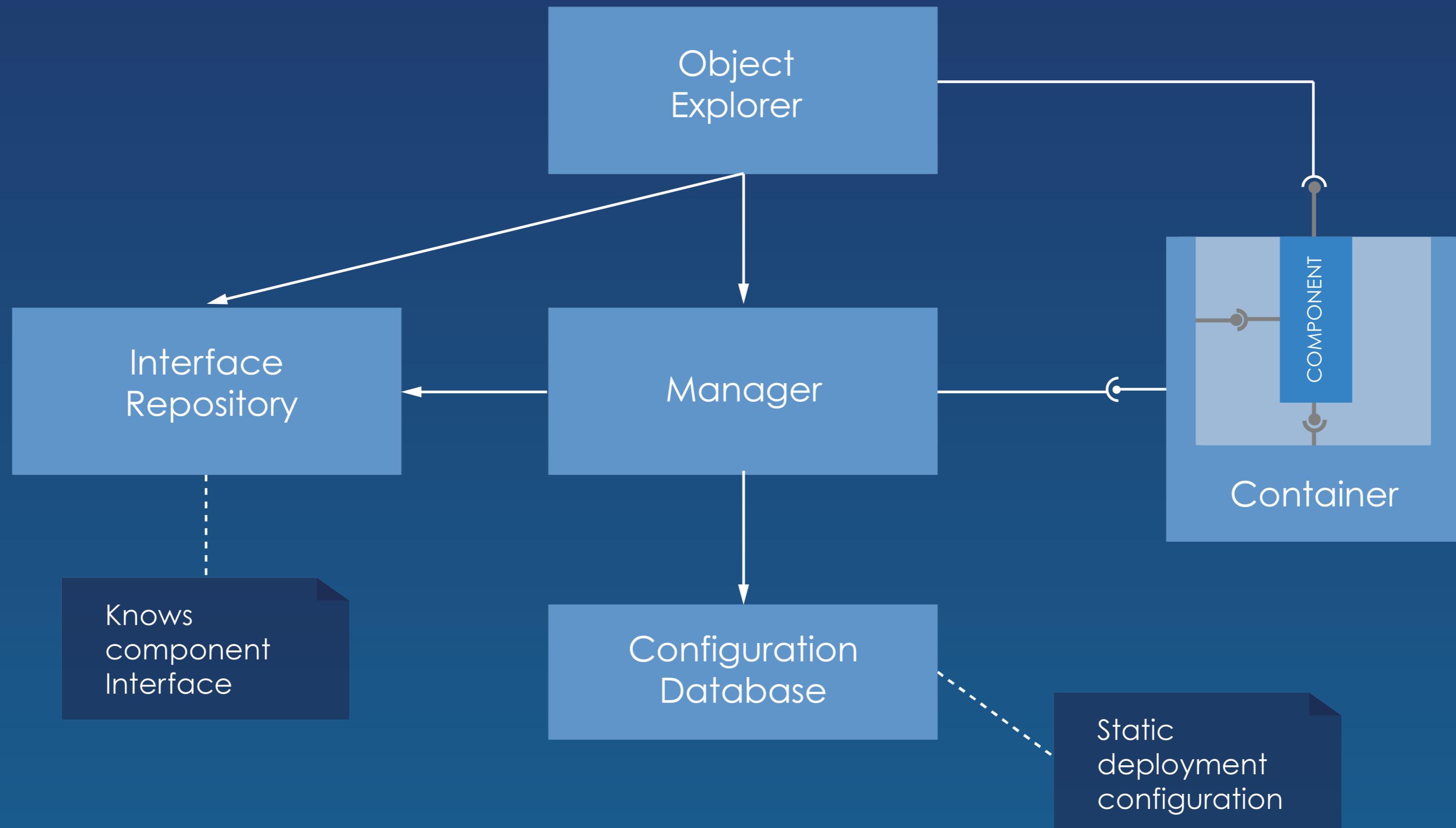
- ✧ Container returns reference to new Component to the Manager
- ✧ Manager returns reference to new Component to first Container
- ✧ Now the Container gives its Component the reference to the other Component

**NOTE:** It is not allowed to pass directly Component references between Components as parameters of interfaces. Components must be passed around **always** by name and a request to the Manager must be issued using the Container Services `getComponent()` interface.

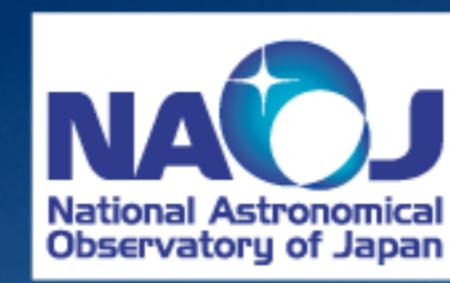
# Interactions: client's view



# Interactions: client's view



# Questions?



## Acknowledgements

ACS presentations were originally developed by the ALMA Common Software development team and has been used in many instances of training courses since 2004. Main contributors are (listed in alphabetical order): Jorge Avarias, Alessandro Caproni, Gianluca Chiozzi, Jorge Ibsen, Thomas Jürgens, Matias Mora, Joseph Schwarz, Heiko Sommer.

The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of Europe, North America and East Asia in cooperation with the Republic of Chile. ALMA is funded in Europe by the European Organization for Astronomical Research in the Southern Hemisphere (ESO), in North America by the U.S. National Science Foundation (NSF) in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and in East Asia by the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Academia Sinica (AS) in Taiwan. ALMA construction and operations are led on behalf of Europe by ESO, on behalf of North America by the National Radio Astronomy Observatory (NRAO), which is managed by Associated Universities, Inc. (AUI) and on behalf of East Asia by the National Astronomical Observatory of Japan (NAOJ). The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.