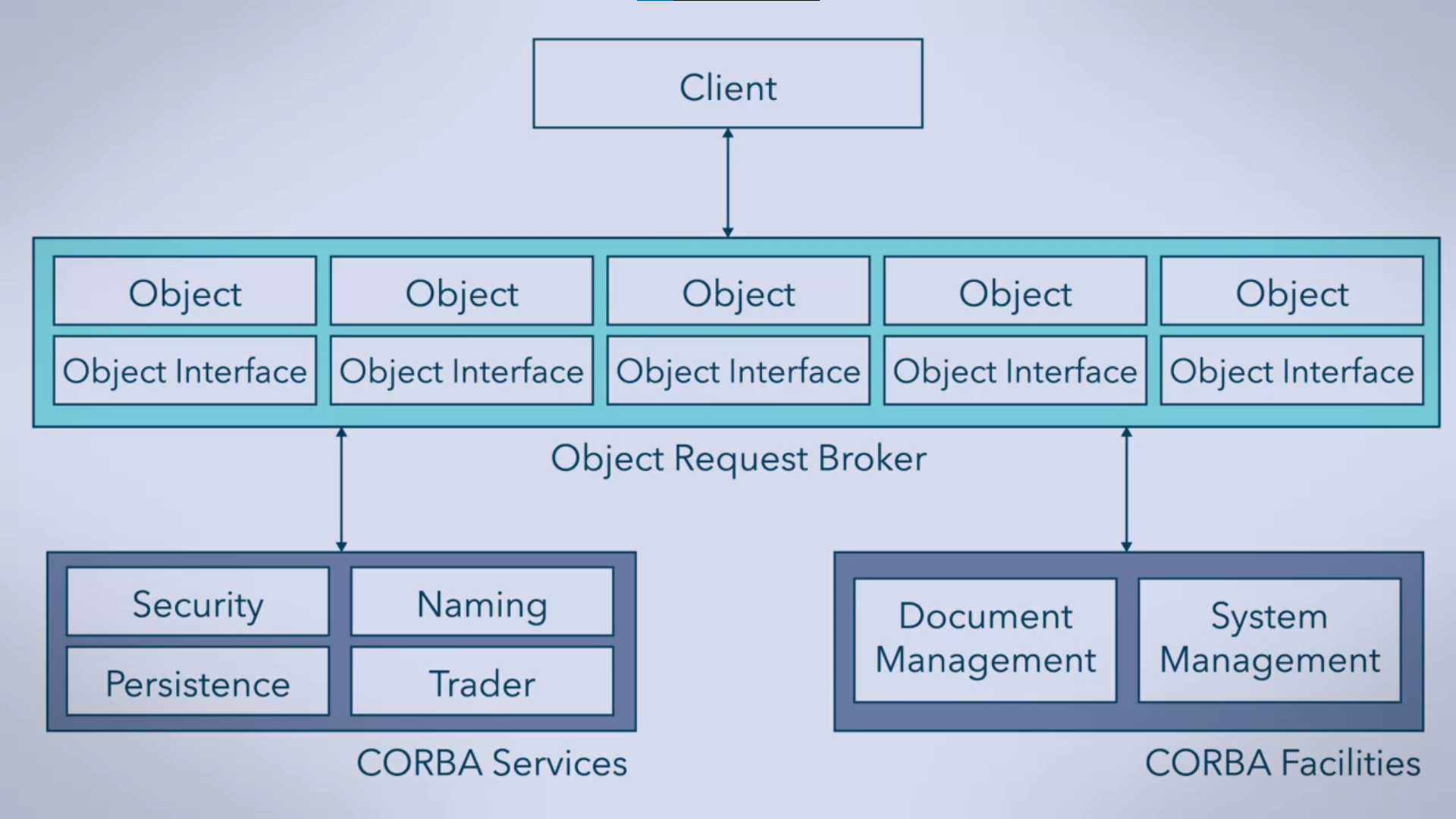


History of Object Brokers

* Late 1970s
* In previous programming paradigm, the signature of a procedure in a program was unique
* In OOP, different classes to have different implementation of the same method signature
  + Causes a problem in middleware
* This leads to Object Brokers in the 1990s, how middleware should handle OOP (inheritance and polymorphism to be exact)
* Most common Object Broker Architecture: Common Object Request Broker Architecture (CORBA)

Common Object Request Broker Architecture (CORBA)

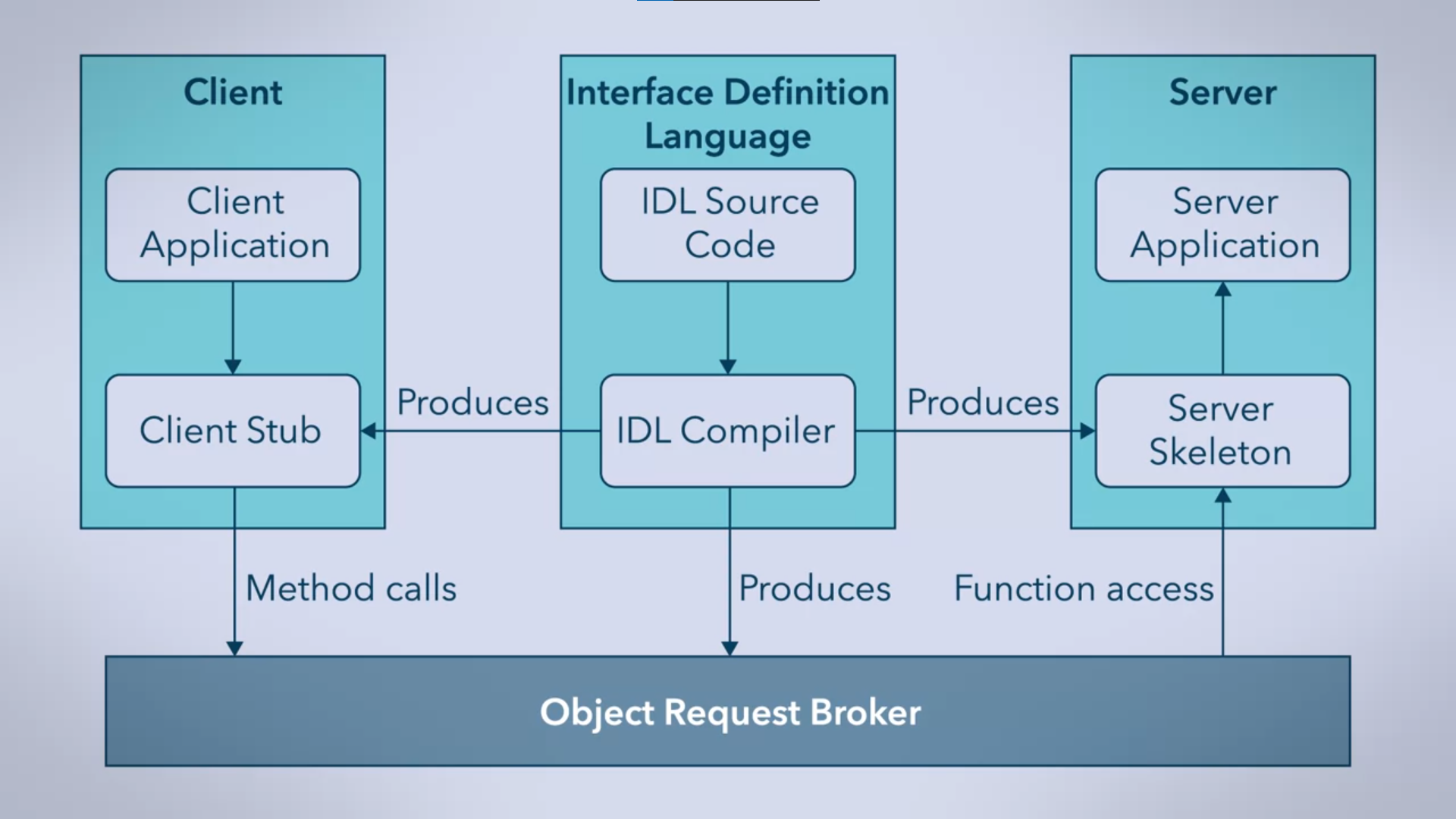
* Set of standards/outline
* Devised by Object Management Group (OMG)
* Create a specification to be
  + independent of the programming languages used to implement clients and servers
  + independent on the operating system
* 3 main components
  + Object Request Broker
    - Marshalling and unmarshalling
  + CORBA Services
    - Object level
  + CORBA Facilities
    - Application level

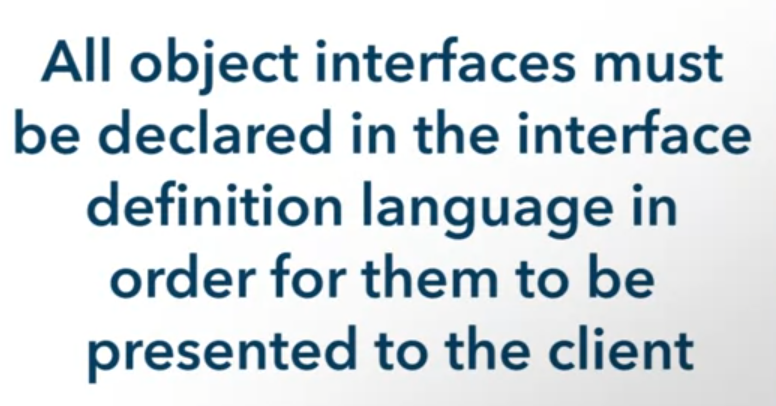
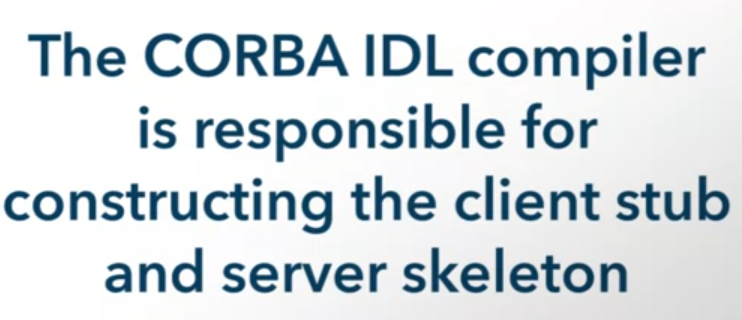
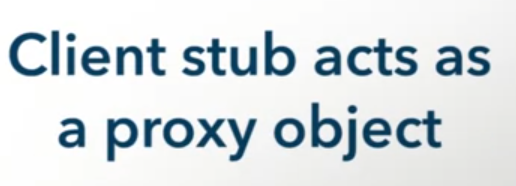


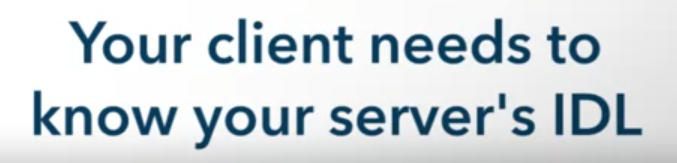
* Follows IDL (Interface Definition Language) for generating stubs

Enhancement in CORBA in IDL

* Allowed inheritance and polymorphism



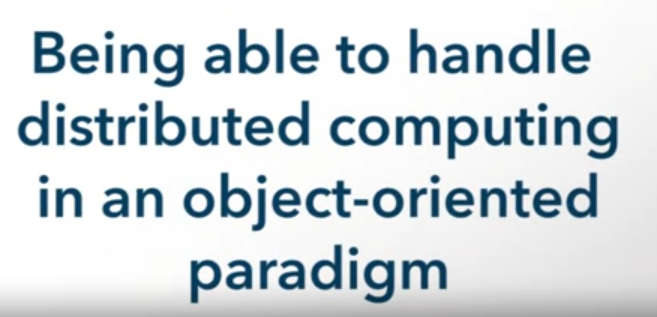
Interoperability

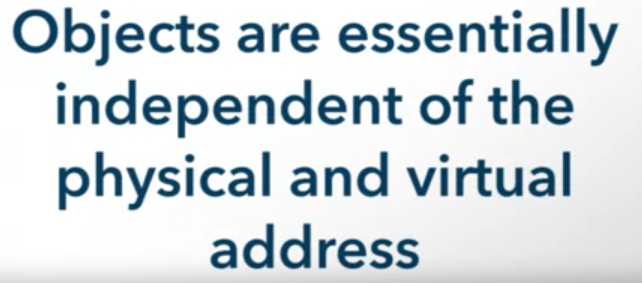
* Can only be achieved when a client binds into one broker
* Without binding, the client can’t access the behaviors of the object
* Easiest: Static Binding
* Can also have Dynamic Binding
  + IDL does not need stub but requires 2 additional components
    - Interface repository – store all IDL definitions
    - Dynamic Invocation Interface – all methods for browsing an interface repository and dynamically constructing methods
      * Naming Service – allows the client to retrieve objects using the “name” of the interface that the client needs
      * Trading Service – will let the client search for objects based on object attributes

Dynamic Invocation Interface Disadvantages

* building is semantically difficult
  + a naming convention needs to be implemented so both client and dynamic interface can understand each other
  + client has to know how to interact with the services that are discovered
* the cost for implementation generally outweighs the benefits

Other advantages of CORBA



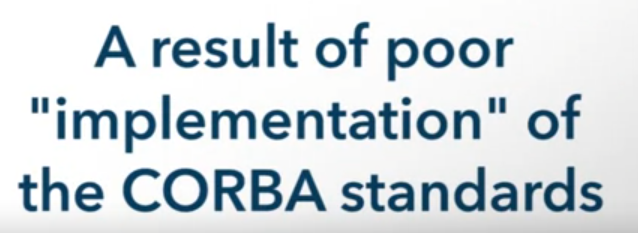


* but CORBA made it possible to reference it





Disadvantages of CORBA





* inexperience
* reduced cost



