CORBA: Interface description language (IDL) in a nutshell

Thomas Braun

byte physics e. K.

11. May 2023

For what does Tango use CORBA?

- Client-Server (intraprocess, interprocess, cross-machine)
- Abstracting away the network and the operating system
- Handling serialization/deserialization in CDR-format[1]

We will only talk about Client-Server in this talk.

[1] http://www.omg.org/cgi-bin/doc?formal/02-06-51

CORBA: Definition of the server interface

Options:

- Static
- Dynamic[2]

Tango only uses static, that means at compile time.

[2] DII (Dynamic Invcocation Interface)

IDL: Interface Description Language

- Domain specific language without flow-control and iterators
- Allows to define complex types using `union`, `struct`, `typedef`, `enum`,... keywords
- Supports inheritance
- Static server definition is done via defining an `interface`

Interface

- An IDL `interface` is a contract for a CORBA object to perform in certain ways[3]
- Interface building blocks: operations, attributes and exceptions
- The CORBA implementation[4] provides the mapping between IDL interface and the target programming language

[3] Very much like a class definition in object oriented programming

[4] omniORB for cppTango/pytango, JacORB for JTango

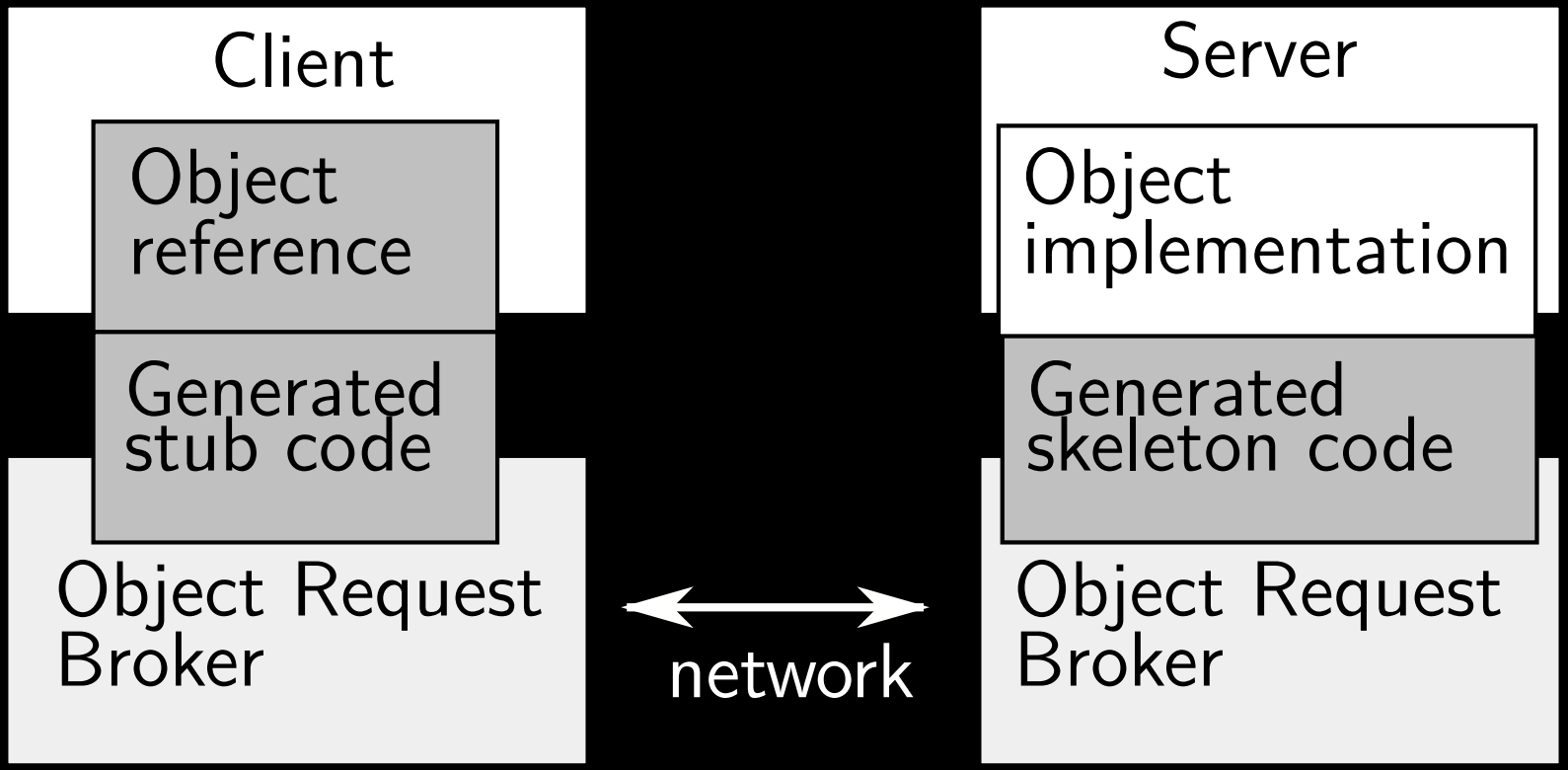
Example interface: echo

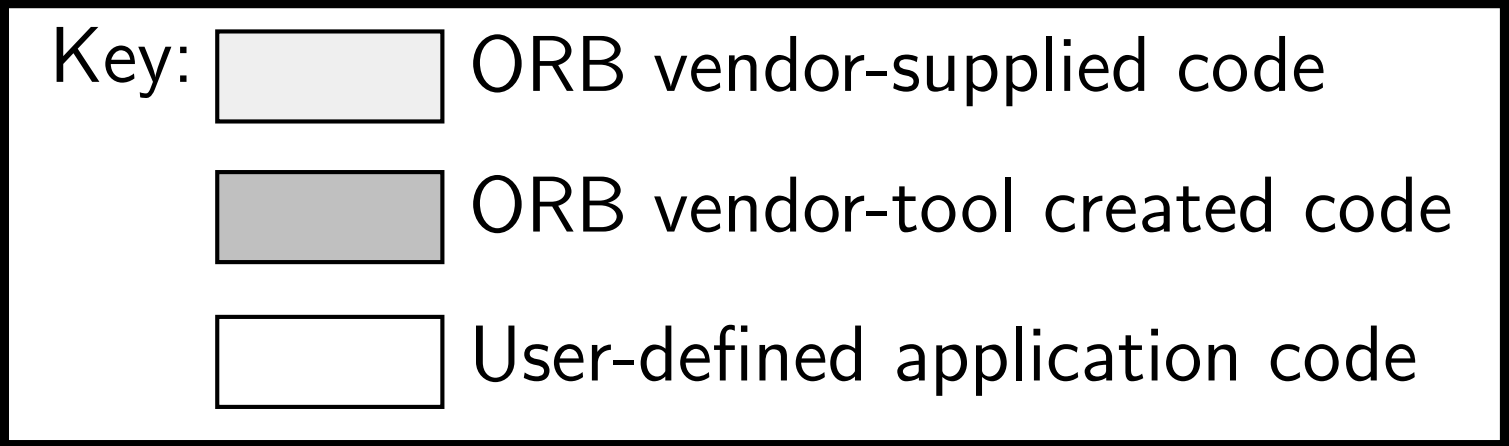
```
interface Echo {
  string echoString(in string mesg);
};
```

The server implementing the `Echo` interface implements one operation, `echoString`, which sends that string back to the client.[5]

[5] Taken from omniORB in `src/examples/echo`

Generating code from IDL





cppTango

- omniidl generates tango.h, tangoDynSK.cpp, tangoSK.cpp
- tango.h declares the **skeleton** classes (`Device_2`, `Device_3`, `Device_4`, `Device_5`) in namespace `POA_Tango`
- tango*SK.cpp contain the **skeleton** class definitions
- cppTango inherits from the skeleton via

```
class Device_5Impl
    : public virtual POA_Tango::Device_5,
      public Device_4Impl
{ // ...
```

Questions ?

Bibliography

[1] Alan Pope. *The CORBA Reference Guide: Understanding the Common Object Request Broker Architecture*. Addison-Wesley, 1998. ISBN: 9780201633863.