



CORBA

Kris Kostro SL/CO

**some slides are from
Xavier Defago/EPFL**



CORBA Basics

- ◆ What is OMG
- ◆ Object Management Architecture (OMA)
- ◆ Interface Description Language
- ◆ Simple client
- ◆ Method calls
- ◆ Static and dynamic invocations
- ◆ Some services
- ◆ Interoperability

Nice



What is CORBA ?

**CORBA is OMG standard
(Object Management Group)**

Common

**Object Request Broker (ORB)
Architecture**

Nice



OMG

- ◆ Consortium composed of 800+ members (SunSoft, IBM, IONA, Oracle, ...)
- ◆ Generates Specifications (e.g. CORBA, UML)
- ◆ OMA (Object Management Architecture)
- ◆ <http://www.omg.org>

Nice



OMG Working Principle

- ◆ Request for Proposal (RFP)
- ◆ RFP defines goals and submission date
- ◆ Members can submit a proposition - working prototype has to exist
- ◆ Subsequent revisions until one proposition is left
- ◆ Once accepted, authors must release a final implementation within a year
- ◆ Usually takes almost 2 years

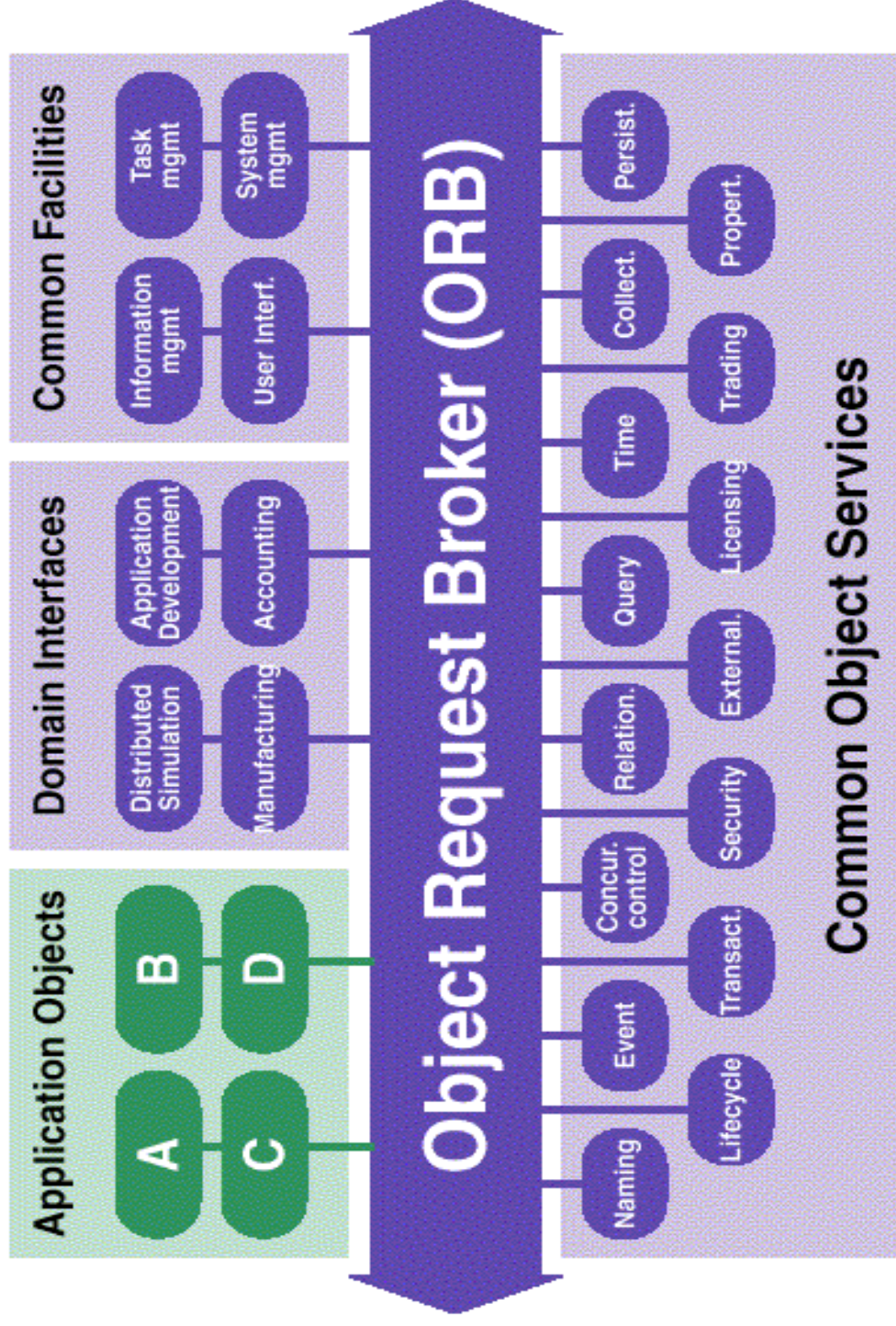
Nice



Evolution of CORBA

- ◆ **CORBA 1.1 (1992)**
 - ◆ Basic ORB, Naming, Interface Repository, C bindings, Events, Basic Object Adapter, ...
- ◆ **CORBA 2.0 (1995)**
 - ◆ Internet Inter-ORB Protocol (IIOP), C++ bindings, Transactions, Security, ...
- ◆ **CORBA 3.0 (and 2.2) (1998)**
 - ◆ Messaging (MOM), Portable Object Adapter (POA), CORBA/DCOM interoperability, Java bindings, Objects by Value, Component Model, ...

Object Management Architecture (OMA)





Interface description

- ◆ **Interface Description Language (IDL)**
 - ◆ **Used by the IDL compiler to generate language specific stubs and skeletons in source code.**
- ◆ **All necessary functionality**
 - ◆ **Precompilation**
 - ◆ **Modular design**
 - ◆ **Inheritance (Multiple)**
 - ◆ **Exceptions**
- ◆ **But no operation overloading**



IDL example

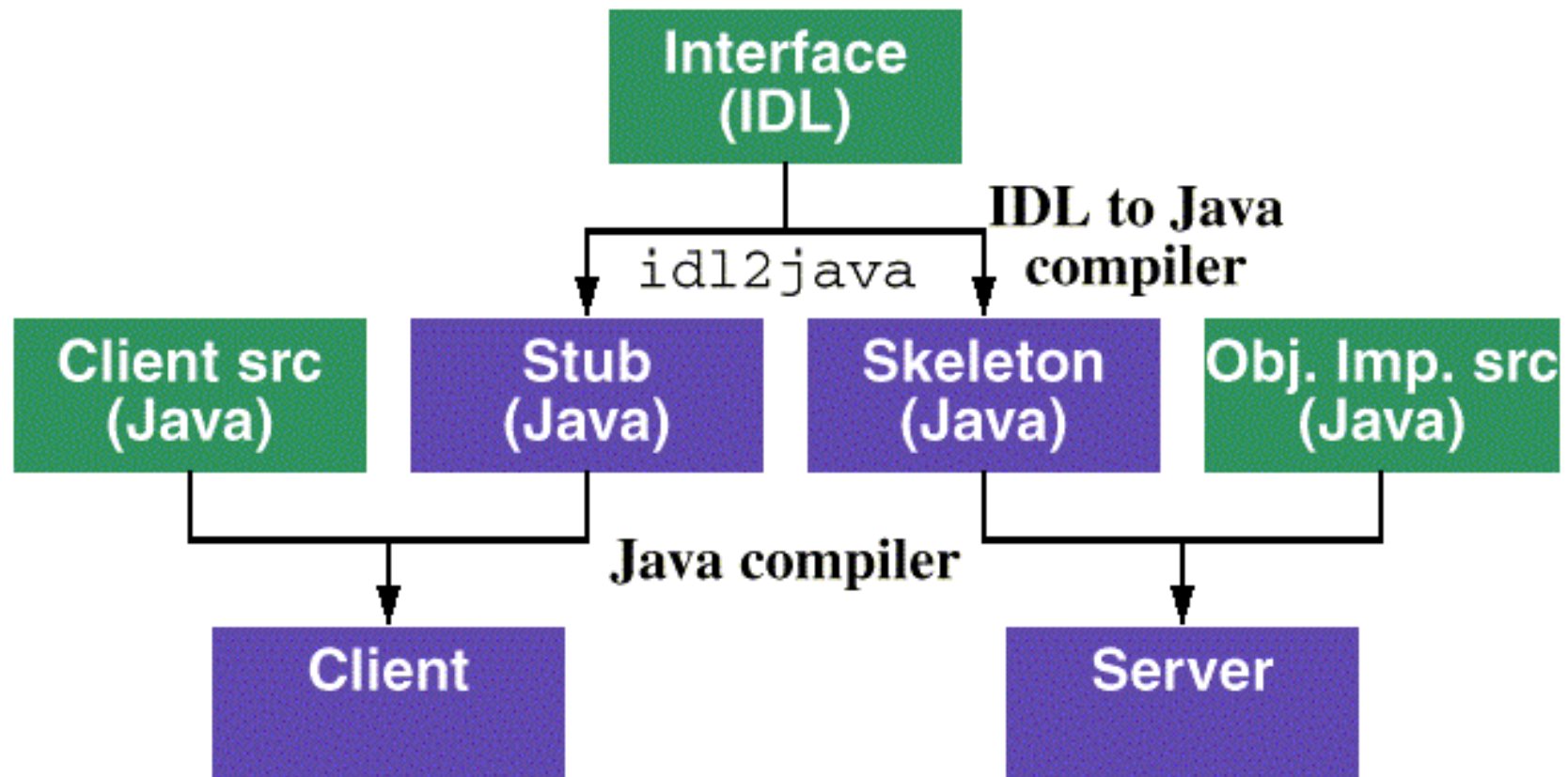
```
module SimpleAgendum
{
    typedef unsigned long Time;
    typedef string Person;

    interface Agenda
    {
        readonly attribute Person owner;

        void add_appointment(in Time at, in Person with);
        void remove_appointment(in Time at);
        Person get_appointment(in Time at);
    };
};
```



From IDL to Java stubs





CORBA client

```
class Client
{
    public static void main(String[] args)
    {
        org.omg.CORBA.ORB orb;
        Agenda agenda;

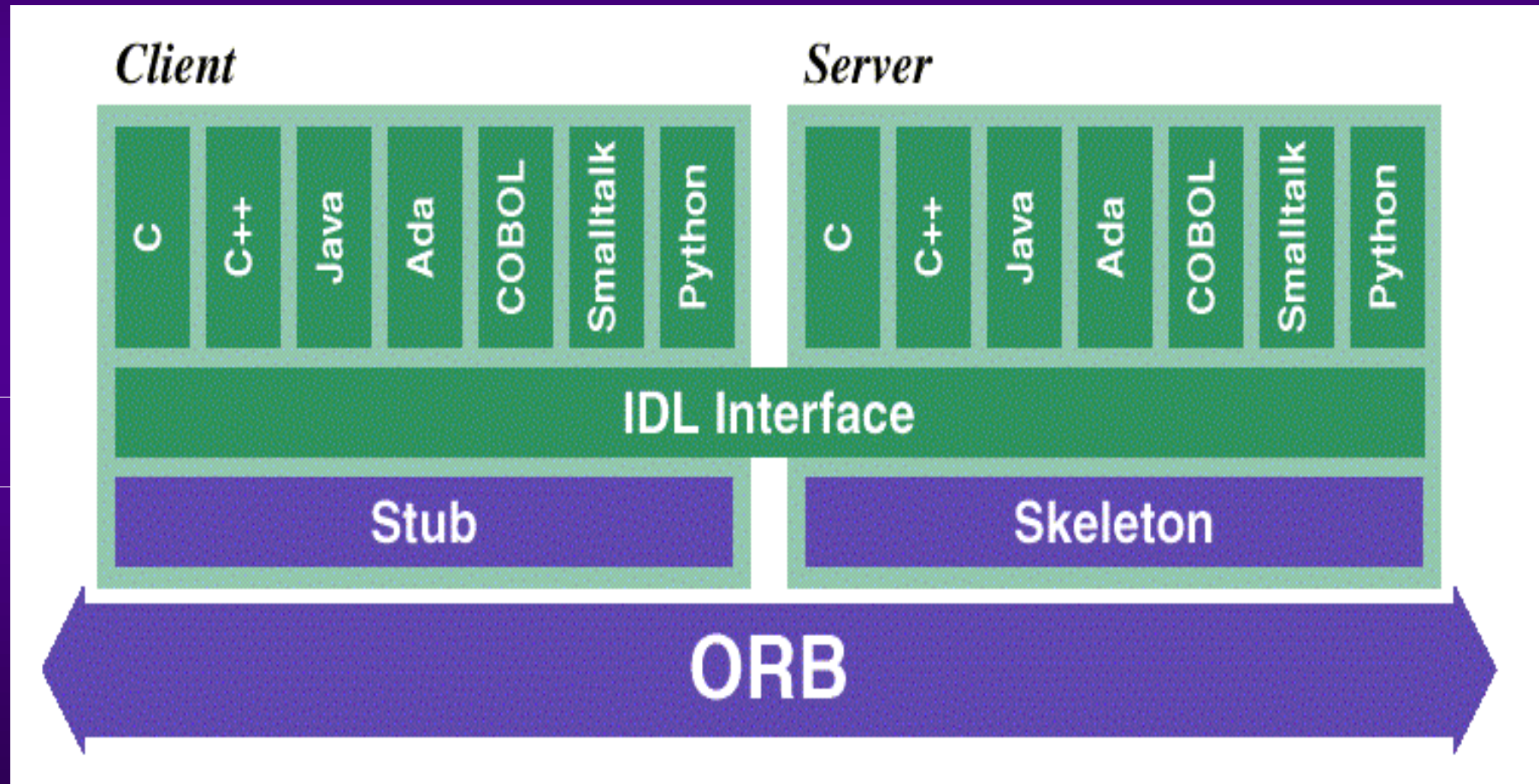
        // Initialisation
        orb = org.omg.CORBA.ORB.init(args, null);

        // object lookup & binding
        agenda = AgendaHelper.bind(orb, "Dilbert's Agenda");

        // object invocations
        agenda.remove_appointment(TODAY);
        agenda.add_appointment(TOMORROW, "Wally");
    }
}
```

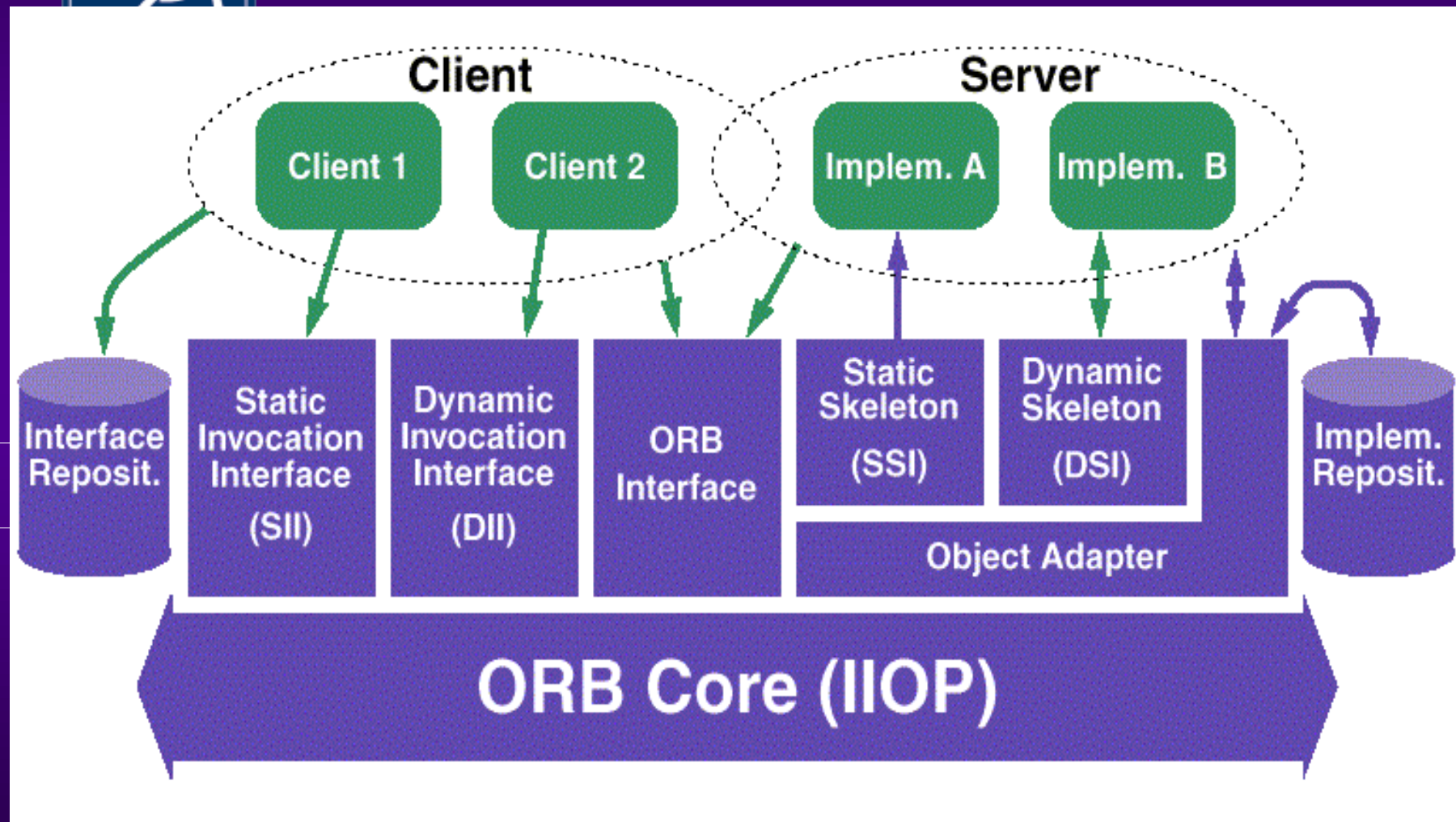


Java language bindings



Nice

Static and dynamic invocations





Object adapter

- ◆ Interface for an object implementation
- ◆ Basic Object Adapter (POA) mandatory
- ◆ Portable Object Adapter in CORBA 2.2
 - ◆ Allows for portable object implementations
 - ◆ Thread managing
 - ◆ Persistent object references



Method calls in CORBA

- ◆ **Synchronous method call**
 - ◆ The client invokes an operation, then blocks waiting for the reply
- ◆ **Asynchronous method call**
 - ◆ The client invokes an operation, then continues processing without waiting for reply. It can later poll or block waiting for the reply.
 - ◆ Only possible with DII
- ◆ **Oneway method call**
 - ◆ Operation without return value.
 - ◆ No guarantee for delivery (best-effort)



Implementing subscription

◆ Callbacks

- ◆ Server implements subscription interface
- ◆ Client implements distribution interface (normally in a separate thread)
- ◆ Client has to register the subscription receiver object with the ORB
- ◆ CORBA callbacks in 3.0

◆ Event service

- ◆ Push model
- ◆ Event channel allows multiple suppliers and multiple consumers

◆ MOM in CORBA 3.0 (CORBA messaging)



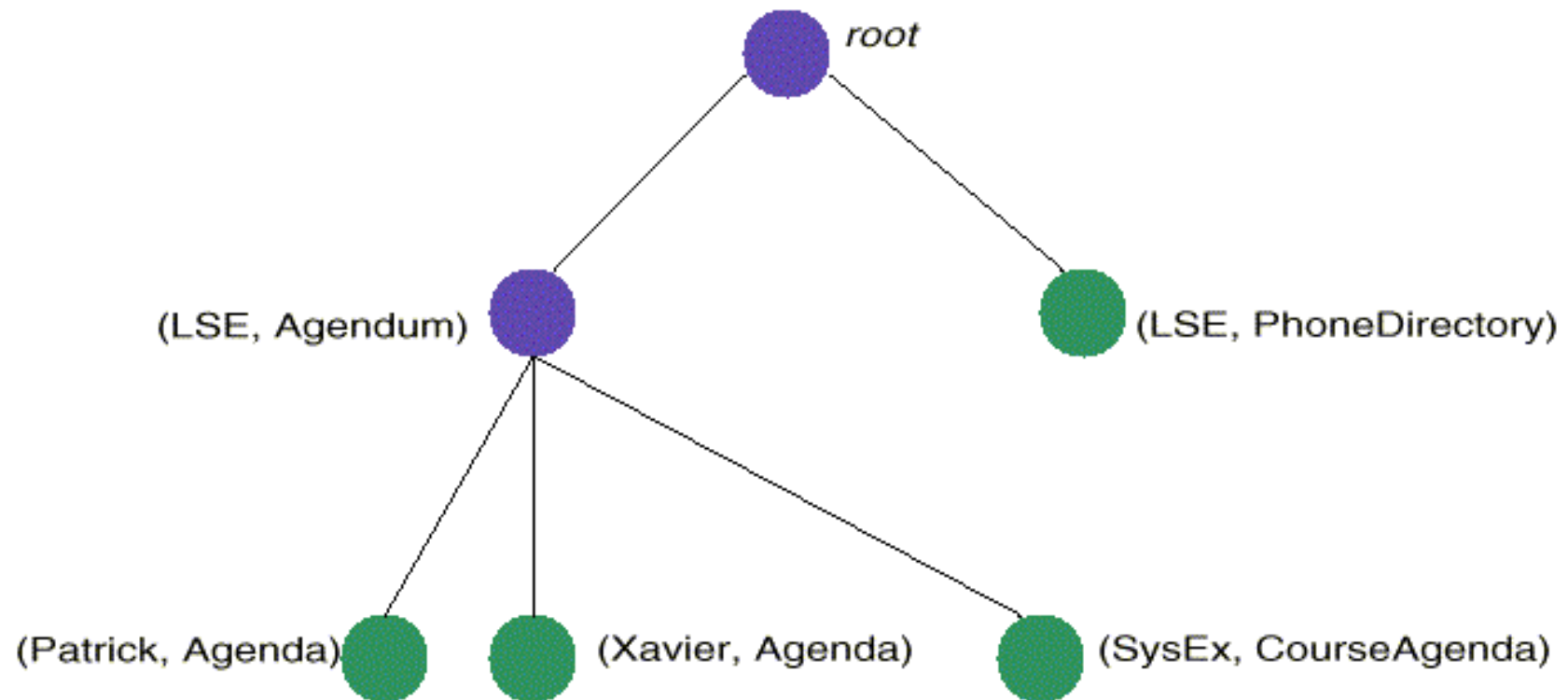
Naming service

- ◆ **Centralized Object Reference Repository**
 - ◆ Registration of object references
 - ◆ Retrieving of object references by name
- ◆ **File system like structure**
 - ◆ Object reference = file
 - ◆ Context = directory
- ◆ The root can be obtained with
 resolve_initial references("NameService")
- ◆ Resolved objects have to be "narrowed" (casted)

Nice



Example naming graph





Object references

- ◆ Unique object designation across address space boundaries
 - ◆ Ex: <ORB type, host, IDL type, instance identifier>
 - ◆ Vendor-specific or Interoperable Object Reference (IOR)
 - ◆ Can be converted to string and vice-versa
- ◆ Objects always passed by reference



Event Service: Interaction Models

❑ Push model

⇒ Initiated by the supplier.



❑ Pull model

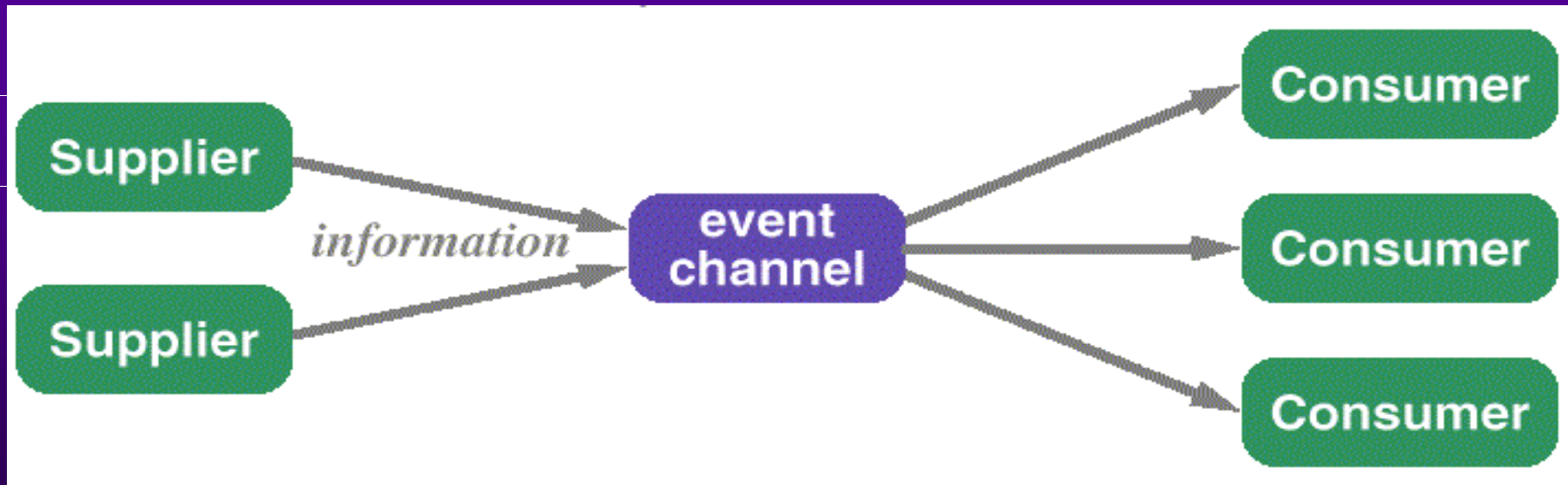
⇒ Initiated by the consumer.





Event Channel

- ◆ Multiple suppliers, multiple consumers
- ◆ Event channel is both consumer and supplier
- ◆ Event channel is CORBA object

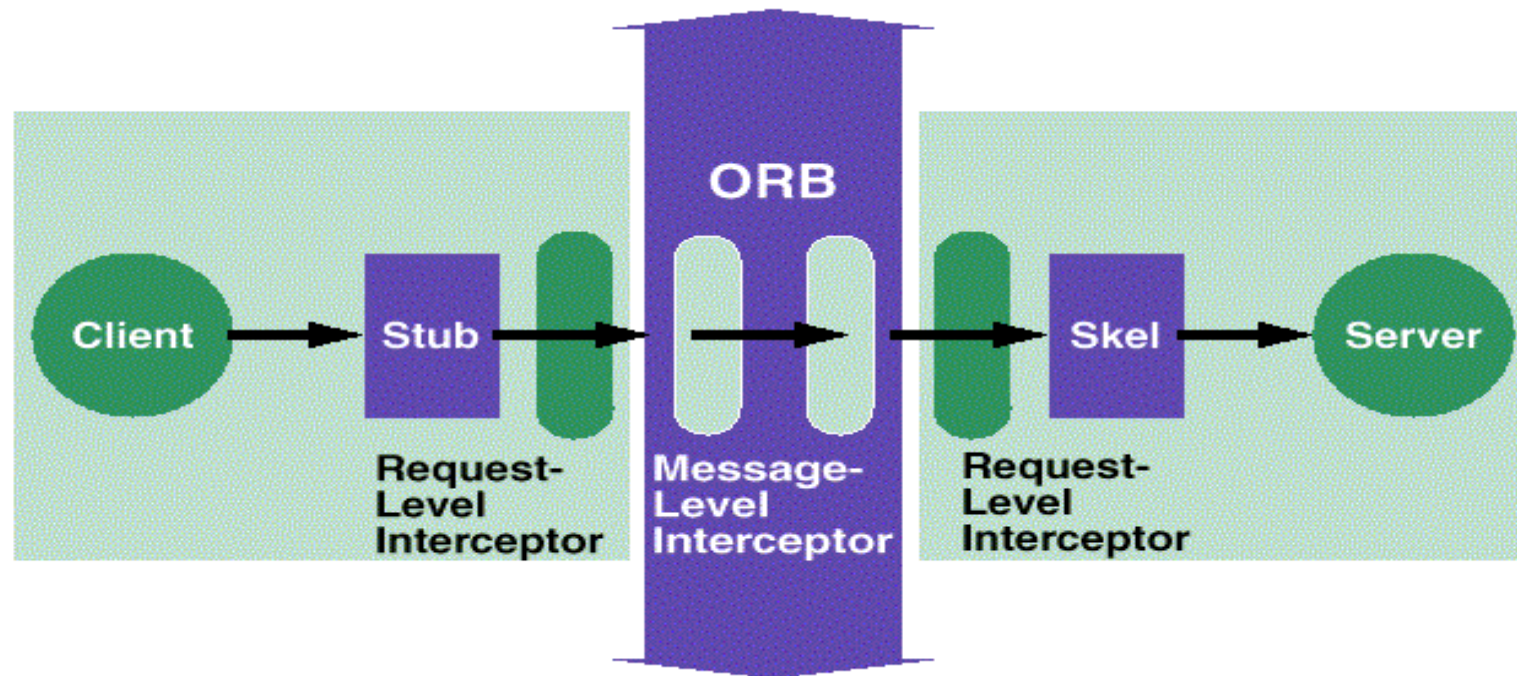


Nice



Interceptors (CORBA 2.2)

- ◆ Statistics, caching, security
- ◆ Modifications of request target





Support for OO model in CORBA

- ◆ Support for encapsulation
- ◆ Support for inheritance (multiple)
- ◆ Objects cannot be passed by value
- ◆ Attributes can be accessed
- ◆ Operations must have unique signature and cannot be overloaded



CORBA Component Model

- ◆ Based on Java Beans
- ◆ White paper May 97 (Sun, Netscape, Oracle, IBM)
- ◆ Joint submission Nov 97
- ◆ Finalized in 1998



ORB Interoperability

- ◆ Assured by IIOP and GIOP
- ◆ Interoperability with COM defined by OMG
- ◆ Bridges available on the market
 - ◆ Enterprise Client from Visual Edge
 - ◆ Object Bridge from Inprise (Visibroker)
 - ◆ COMET from IONA (Orbix)
 - ◆ Power Bridge from ExperSoft
- ◆ In Java 1.2 RMI is using IIOP



Platform support

- ◆ Big choice for common platforms (Windows, Sun, HP, IBM)
- ◆ Public domain implementations

Visibroker

Leading CORBA ORB, Client integrated in Netscape

Orbix

Leading CORBA ORB, was ported to LynxOS

Arbacus

Commercial ORB, free for non-commercial use.

OmniORB

Free from AT&T, has been ported to LynxOS2.5.1 on PC

ILU

Free from Xerox. Ported to LynxOS 2.5.1 PowerPC by Atlas

Nice



CORBA/Java Coalition

- ◆ Netscape is bundling VisiBroker with their browser
- ◆ Oracle has adopted CORBA/Java and it has IIOP ORB for the server
- ◆ JavaSoft is integrating CORBA into Java
 - ◆ IIOP for RMI
 - ◆ No need for IDL
- ◆ Pure Java CORBA ORBs

Nice



Conclusions

- ◆ CORBA strength is in interoperability: languages, platforms, public domain software
- ◆ Has a very rich and complete set of facilities and services
- ◆ There are many services defined but their implementations are slow to appear
- ◆ By it's working method OMG is creating compromises