

2. CORBA arkitektur og runtime-miljø

- Arkitektur
- Runtime
- Objekt-referencer
- Kommunikationsprotokoller
- ORB'er og Services

Arkitektur

Består af en eller flere ORB'er med dertilhørende IDL-compile og services.

Runtime

Når man skal have et CORBA-system op

at køre, skal der være en server skrevet i et programmeringssprog med nogle implementationer af IDL-beskrivne interfaces.

1. **Serveren** laver et objekt og en objekt-reference.
2. **Serveren** skriver referencen til en fil (`object_to_string()`, ikke skalerbart) eller binder den til en Naming Service.
3. **Klienter** bruger referencen til at få en udgave af objektet.
4. IDL-compileren har oprettet en **stub** på klienten, som bruges til at kalde metoder på.
5. **Klienten** kalder metoder på stubben.
6. **ORB'en** overfører parametre (og sørger for at marshalle og unmarshalle undervejs).
7. **Serveren** udfører arbejdet på servanten og **ORB'en** overfører parametre og evt. returværdi.

Vi har arbejdet med JacORB og VisiBroker. Hhv. Java og .Net-ORB'er.

Objekt Referencer

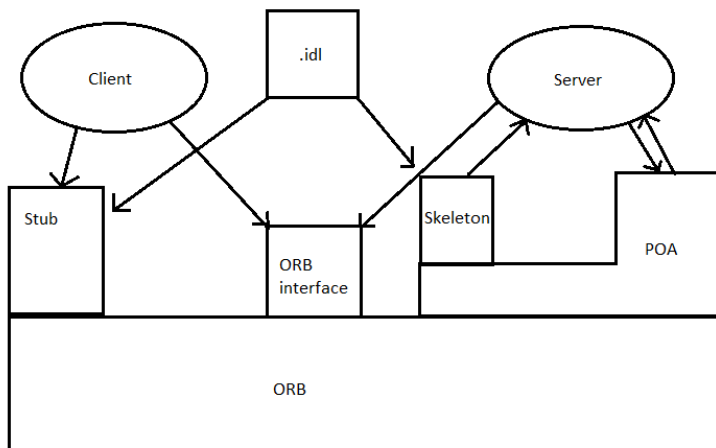
En objekt reference bruges til at **lokalisere** CORBA-objekter. **Kender typen** på CORBA objektet der refereres til. Skjuler at der forbindes til en server. *Location transparency*.

Forskellige typer: **Instance** (lokal), **IOR** (til at sende over netværket) og **stringified** (IOR konverteret til streng).

Kommunikationsprotokoller

Inden CORBA 2.0 var der ingen standardisering af, hvordan kommunikation imellem ORB'er skulle foregå → en masse proprietære protokoller lavet af ORB'er.

- **GIOP (General Inter-ORB Protocol):** Abstrakt protokol, som beskriver kommunikation imellem ORB'er. Standardiserer kommunikation imellem ORB'er. Tre hovedelementer:
 - *CDR*: Beskriver hvordan data skal beskrives når det skal overføres.
 - *Message formats*: GIOP 1.2: 8 beskedformatter til at håndtere requests. Fx til at requeste et objekt.



- *Transport assumptions*: Antagelser om transport: Semi-reliable connection-oriented transport som er implementeret som en byte (octet) stream. Klient laver altid første forbindelse.
- **IIOP (Internet Inter-ORB Protocol)**: Implementation af GIOP på TCP/IP. En enkelt tilføjelse:
 - *IIOP-profil*: Som lægges på en IOR. Består af *IIOP-version*, *host* og *port* på server og *Object Key* (som typisk består af en POA + ObjectId).

ORB'er og Services

- En **ORB (Object Request Broker)** er implementationen af CORBA. Dvs. en ORB sørger for kommunikationen imellem server og klient. Herudover skal ORB'en lave stub- og skeleton-kode ud fra IDL-filer, stille runtime library til rådighed. Skal kunne snakke sammen. OMG laver kun specifikationen af CORBA, 3. partsproducenter laver ORB'erne.
- **Services** er obligatorisk for en ORB, der implementerer CORBA. Eksempler er EventService, som vi har brugt til at lave en chat-klienter vha. push-meddelelser imellem klienter. Problemet er, at det langtfra er alle ORB'er der rent faktisk implementerer de forskellige services.

SE KODE FOR SERVER-KLIENT EKSEMPEL