

5. Internet Communication Engine

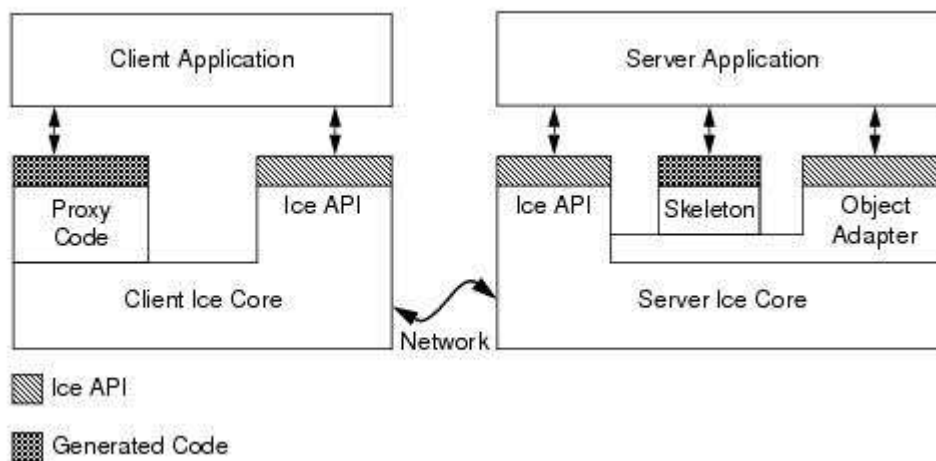
... arkitektur og runtime-miljø

- Arkitektur
- Proxies
- Runtime
- Server-client
- Diskussion og sammenligning

Arkitektur

Objekt-orienteret middleware platform, som kan køre på forskellige OS, programmeringssprog og kommunikere vha. forskellige protokoller. Nyere end CORBA, men ligner meget.

- **Slice:** ICEs pendant til CORBAs IDL. Der er nogle fordele i forhold til IDL, bl.a. C++-preprocessor.
- **Ice Object:** Består af ObjectId og interface + servant. Lidt ligesom CORBA-objekt.
- **At-most-once semantic:** Ice sørger for, at et metodekald kun sker én gang. Hvis ikke det er muligt, får klienten en besked herom. Kan være et problem i forhold til netværksfejl, hvorfor man kan gøre en metode idempotent, hvor metoden altså er ligeglad med, at den udføres flere gange.
- **Ice Protocol:** RPC protocol, som benytter TCP/IP eller UDP. Nogle beskedtyper, tilstande, encoding, header som beskriver beskedtype, størrelse osv.



- **Ice Core:** Sørger for at kommunikere over netværket.
- **Proxy code:** Genereres ud fra Slice. Kald på proxy resulterer i RPC. Marshalling og unmarshalling.
- **Skeleton code:** Genereres ud fra Slice. Står for det samme som Proxy, bare ind i serveren.
- **Object Adapter:** Ligesom POA: kreerer proxier, mapper requests til objecter, osv.

Proxies

En proxy er en lokal repræsentation (client-side) af et Ice objekt. Indeholder information om serverplacering og objectId på serveren.

- **Direct proxy:** Indeholder al information om (objectId + server), hvordan et Ice object lokaliseres.
- **Indirect proxy:** Kan enten angive objectId alene (well-known) eller objectId sammen med object adapter – så er det OA, som finder objektet på serveren. Man kan bruge en location service, da man ikke angiver, hvor serveren befinder sig (bruger objectId eller adapter som key).

Fordelen ved indirect er, at servere kan flyttes (*location transparency*).

Runtime

Når en klient kalder en metode på en proxy så gør Ice:

1. Finder Ice objektet.
2. Aktiverer Ice objektets server, hvis den ikke kører.
3. Aktiverer Ice objektet på serveren.
4. Overfører in-parametre til Ice objektet.
5. Venter på, at metoden udføres.
6. Returnerer out-parametre og returværdien til klienten (eller en exception, hvis der er fejl).

Server-client

- Slice-fil oversættes med en passende slice2*-compiler. Den genererer proxy, holder, helper, _Disp og operations.
- Server implementeres (se kode), sammen med en servant, som skal extendere _Disp-filen.
- Registrer Ice object i IceGrid.
- Klient implementeres. Helper.checkedCast() bruges til at caste Ice objektet til den korrekte type specificeret af Slice.

Diskussion og sammenligning

CORBA understøtter større heterogenitet, da der er tale om en åben standard, hvor man "bare" kan lave sin egen ORB, hvis man ønsker et nyt sprog understøttet. CORBA standarden er fuldstændig åben, og OMG styrer processen omkring nye features.

ICE herimod laver selv disse understøttelser, og dækker derfor kun indtil videre 7 programmeringssprog.

Fordelen ved ICE modellen er, at man ved, at tingene fungerer sammen. ICE understøtter både Linux, OSX, og Windows.

CORBA er gratis at bruge, idet CORBA ikke udvikler noget, men kun fremlægger standarden – det er således en ORB man bruger. ICE derimod er ikke gratis at bruge, hvis man skal bruge det i et kommercielt system.

De services de siger, er til rådighed i Ice, ved man er der i ICE.

- **Java RMI:** Ikke heterogent i forhold til sprog, men i forhold til.
- **Web Services:** Openness, ultra heterogent. Fungerer ikke i større systemer.
- **.Net Remoting:** Kun .Net sprog. Windows pendant til Java RMI.
- **WCF:** For Windows, men kan også snakke med andre systemer. Server kan kun implementeres i .Net, dog kan objekter publiceres over SOAP.