Interoperabilität und IIOP

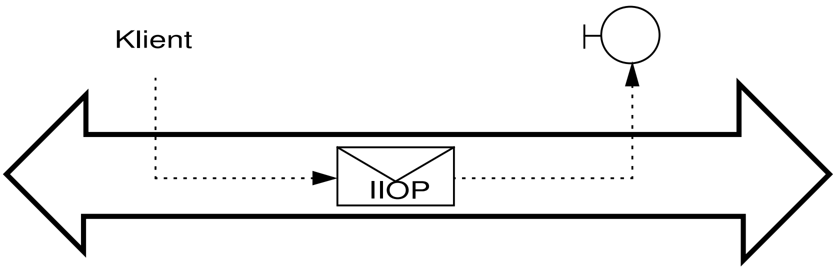
Kay Römer
Institut für Informationssysteme
ETH Zürich

Eidgenössische
Technische Hochschule
Zürich

Überblick

- Einordnung in die CORBA-Architektur
- Motivation
- Funktionsweise von IIOP
- Interoperabilität mit "fremder" Middleware
- Firewalls

Einordnung



IIOP := Internet Inter ORB Protocol

- standardisiertes Protokoll (auf dem “Software-Bus”) zur Kommunikation zwischen ORBs
- basiert auf Internet-Protokollen TCP/IP

Warum IIOP?

- Heterogenität verteilter Systeme:
  - verschiedene ORB-Hersteller
  - verschiedene Hardware, Betriebssysteme, Programmiersprachen
- spezifische Probleme: Byteorder, Zeichensätze, Fließkommaformate, ...
- Zusammenarbeit von ORBs in heterogenen Umgebungen:

Interoperabilität

Elemente von IIOP

- **Datenrepräsentation** (Darstellung von IDL-Datentypen als Bytestrom)
- **Protokoll** (Nachrichten, die zwischen Sender und Empfänger ausgetauscht werden können)
- **Transport** (Übertragung von Nachrichten zwischen Sender und Empfänger)

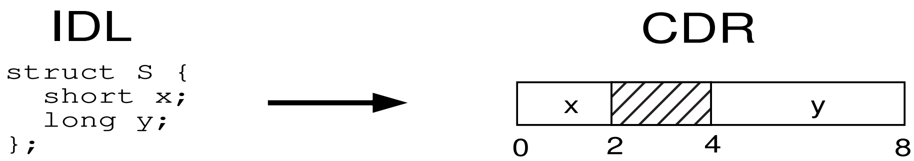
Datenrepräsentation: CDR

CDR := Common Data Representation

- Definiert für jeden IDL-Datentyp eine Abbildung auf einen Bytestrom
- Enhält keine Typinformationen
- Annahme: Kommunikations-Verzögerung hängt nur geringfügig von Packet-Größe ab (gilt das immer??)
- Daher: Optimiert auf Marshalling-Performance, nicht auf Größe des resultierenden Bytestroms:
    - Alignment
    - SMIR (*Server Makes It Right*)

CDR: Alignment

CPU kann besonders schnell lesen und schreiben, wenn Daten an bestimmten Positionen im Speicher plaziert sind: Einfügen von “Lücken”, um diese optimale Plazierung zu erreichen:



- Vorteil: Einpacken/Auspacken schnell
- Nachteil: Packete unnötig groß (im schlimmsten Fall nahezu doppelt so groß)

CDR: SMIR

Sender schickt Daten in seinem “nativen” Format (ohne zu konvertieren), Empfänger wandelt Daten gegebenenfalls in sein eigenes natives Datenformat um:

- Vorteil: Konvertierung nur wenn tatsächlich notwendig
- Nachteil: Informationen über den Sender müssen übertragen werden
  - Byteorder
  - Zeichensätze
  - ...

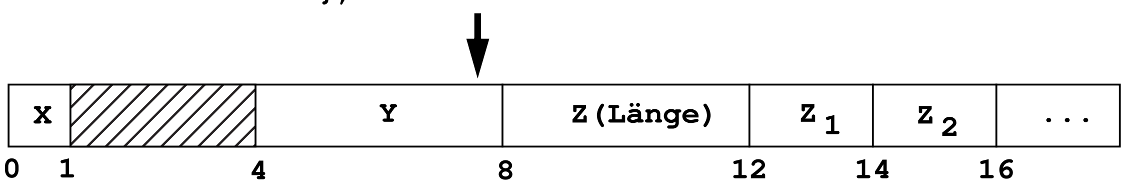
CDR: Abbildung der Datentypen

Marshalling := Abbildung IDL-Datentyp -> Bytestrom
Demarshalling := Abbildung Bytestrom -> IDL-Datentyp

Typ	Alignment	Abbildung
Char	1	1 Byte
Short	2	2 Bytes
Long	4	4 Bytes
Float	4	4 Bytes (IEEE)
String	4	ULong (Länge), Char, Char, ..., 0
Seq<X>	4	ULong (Länge), X, X, ...
Struct	?	Elemente
Union	?	Discriminator, Case

CDR: Beispiel

```
struct Beispiel {
  char X;
  long Y;
  sequence<short> Z;
};
```



Protokoll: GIOP

GIOP := General Inter ORB Protocol

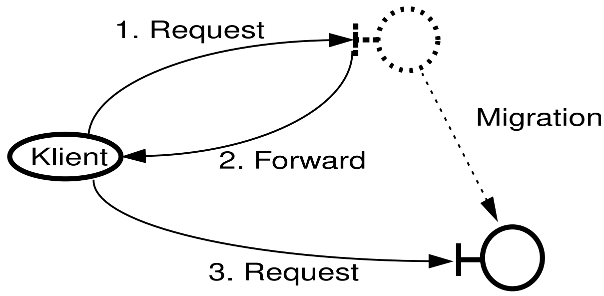
- Definiert Nachrichten für
  - Methodenaufruf (Request, Response, Forward, Cancel)
  - Objekt-Lokation (LocateRequest, LocateResponse)
  - Andere (Error, CloseConnection)
- Unterstützung für Fragmentierung
- Unterstützung für Objekt-Migration

GIOP: Fragmentierung

- Große Nachrichten können in kleinen Häppchen verschickt werden
- Reduziert im ORB benötigten Puffer-speicher

GIOP: Migration

Klienten, die Anfragen an alte Lokation eines Objektes senden, erhalten als Antwort eine “Forward” Nachricht mit der neuen Adresse des Objektes:



Transport: TCP/IP

- Überträgt GIOP-Nachrichten
- Muß bestimmten Anforderungen genügen:
  - verbindungsorientiert
  - Bytestrom-orientiert
  - zuverlässig
- IIOP = GIOP über TCP/IP
- Auch andere Transportprotokolle möglich (beispielsweise über SSL-ified TCP/IP)

Interoperabilität mit anderer Middleware

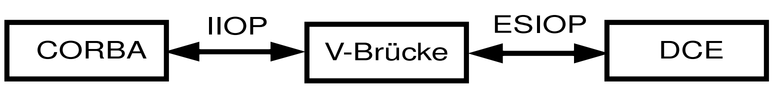
ESIOP := Environment Specific Inter ORB Protocol

- ESIOP definiert Abbildung von CORBA/IDL auf eine andere Middleware
- Momentan verfügbar für DCE, DCOM, Java RMI
- Brücken vermitteln zwischen CORBA und der anderen Middleware

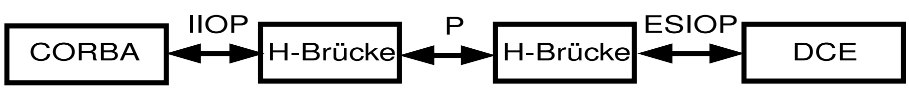
Brücken

- **Inline-Brücke**: ORB versteht direkt ESIOP

- **Voll-Brücke**: Direkte Umsetzung der Protokolle

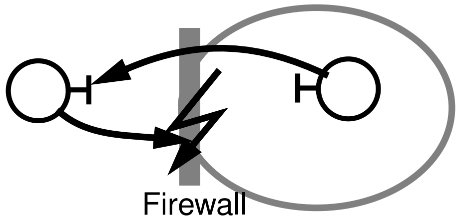


- **Halb-Brücke**: Indirekte Umsetzung der Protokolle über gemeinsames Protokoll P



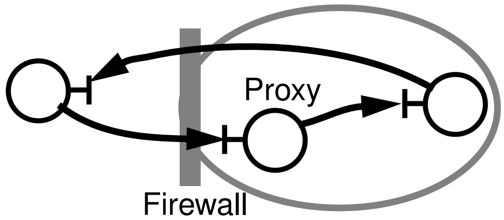
IIOP und Firewalls 1

- Firewalls schützen einen bestimmten Netzabschnitt ("LAN") vor unberechtigtem Zugriff:
  - statisch konfiguriert durch erlaubte/verbotene IP-Adressen und Port-Nummern
  - Problem: Kommunikation zwischen CORBA-Objekten durch Firewalls hindurch (IP-Adressen und Portnummern sehr dynamisch)
  - Beispiel: Distributed Callback



IIOP und Firewalls 2

- Lösungsansätze:
  - spezieller Proxy (Adresse im Firewall eingetragen)



  - Bi-Directional GIOP (Callbacks gehen über gleiche Verbindung wie der ursprüngliche Aufruf)