

CORBA

Überblick, IDL

Radu Vatav

Übersicht

- Geschichte
- Architektur
 - Kommunikationsmodell, Objektmodell
 - Adapters
 - ORB
 - Marshalling
 - Clients und Server
- IDL

Geschichte

- OMG (Object Management Group), gegründet 1989
 - „ein gemeinsames, architekturbezogenes Gerüst für objektorientierte Anwendungen, das auf weit verbreiteten Schnittstellenspezifikationen basiert“
- OMA (Object Management Architecture)
- Anfang 1991 – erste CORBA Version
- 1996 – CORBA 2.0
 - Interoperabilität zwischen ORBs verschiedener Hersteller
- 2001 – CORBA 3.0

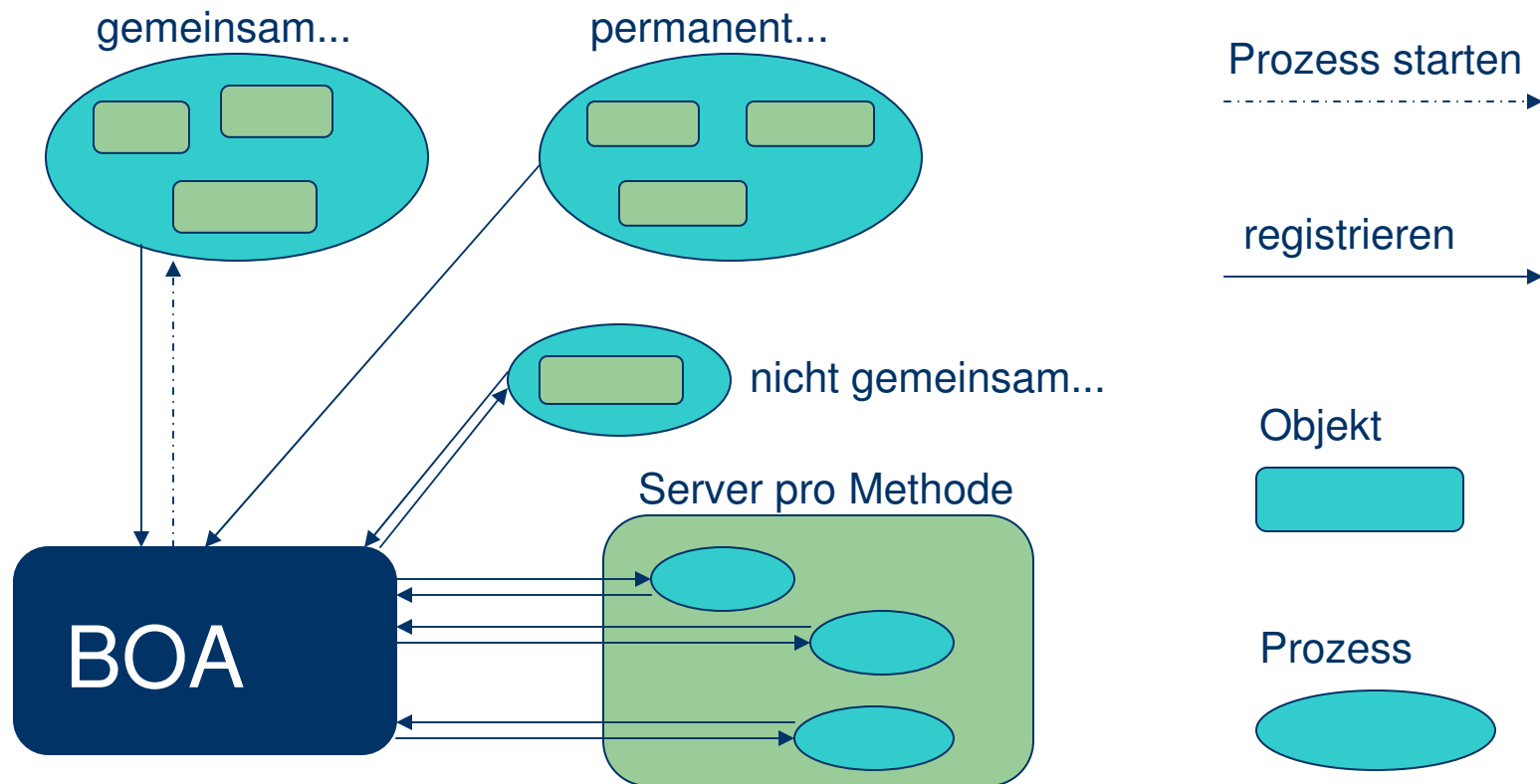
Architektur

- Kommunikationsmodell
 - netzwerkneutral
 - GIOP (General Inter-ORB Protocol)
- Objektmodell
 - Transparenz
 - höhere Fehlerwahrscheinlichkeit
 - System-Exceptions

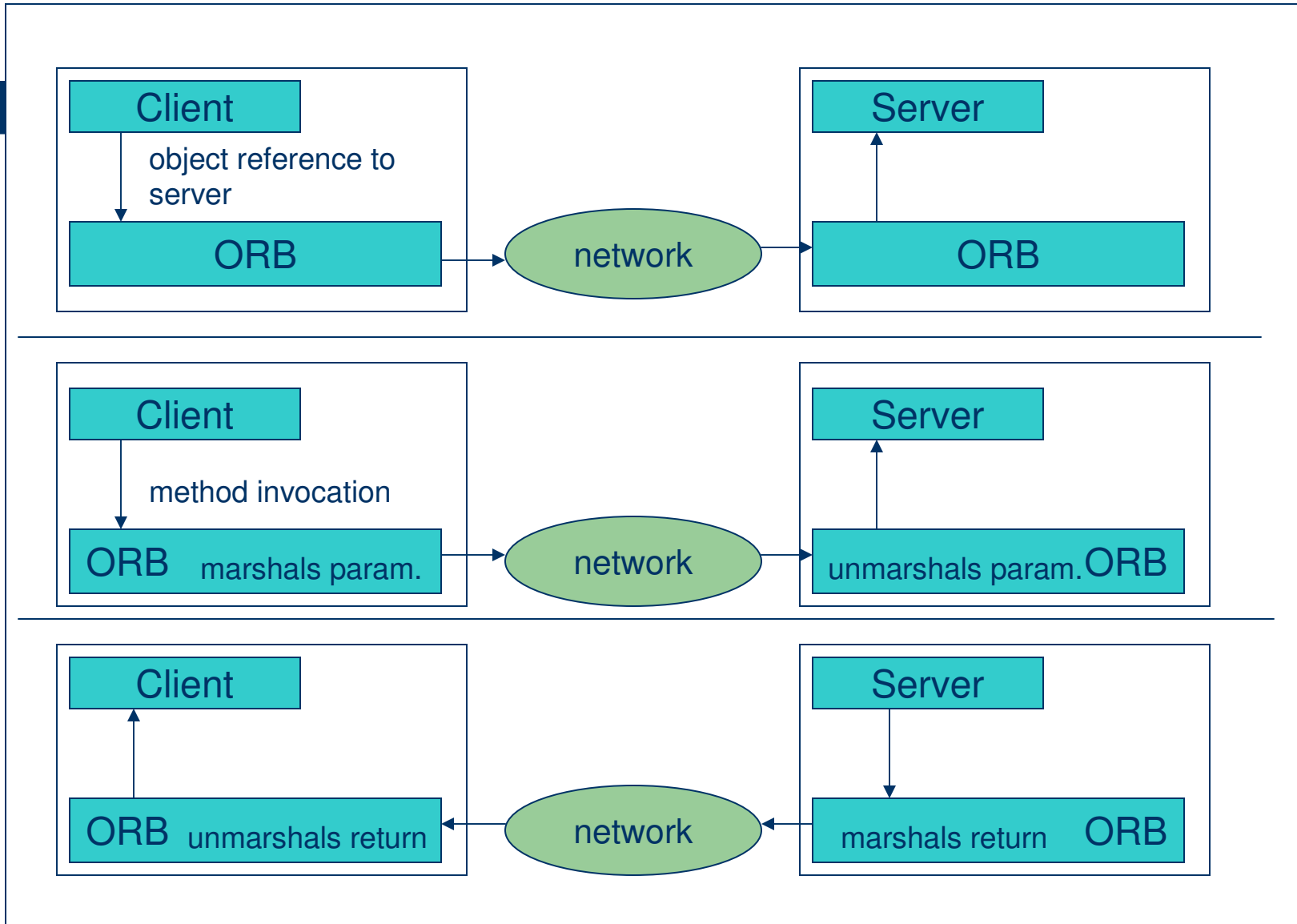
Architektur

- Object-Oriented Database Adapter
- Library Object Adapter
- BOA (Basic Object Adapter)
 - Aktivierungsstrategien:
 - gemeinsam verwendete Server
 - nicht gemeinsam verwendete Server
 - Server-pro-Methode
 - permanente Server

Architektur - Aktivierungsstrategien



Architektur - ORB



Architektur - IOR

- Interoperable Object Reference:

IDL-Schnittstellentyp	Details zu Protokoll und Adresse			Objektschlüssel	
Schnittstellen – Repository-ID	IIOP	Host – Domain - Name	Port - Nummer	Adapter - Name	Objekt - Name

IOR:010400002800000049444c3a6f6d672e6f72672f436f734e616d696e672f4e616d696e67436f6e746578743a312e300001000000000000002c0000000101000014000000746e696e6b7061642e74656c656e742e6e6574004f060000080000000000000001000000

Architektur

- Stubs und Skeletons
- Clients und Server
- Callback



Interface Definition Language

- Datentypen

- void
- boolean
- char – 8 Bit
- wchar – 16 Bit
- float – IEEE einfache Genauigkeit
- double – IEEE doppelte Genauigkeit
- long double – IEEE erweiterte doppelte Genauigkeit
- (unsigned) long – 32 Bit
- (unsigned) long long – 64 Bit
- (unsigned) short – 16 Bit
- octet
- string

Interface Definition Language

- Methoden und Parameter

```
[oneway] return_type MethodenName(  
    param1_dir, param1_typ, param1_name,  
    param2_dir, param2_typ, param2_name...);
```

- param_dir: *in*, *out* oder *inout*
- oneway: die aufrufende Methode blockiert nicht

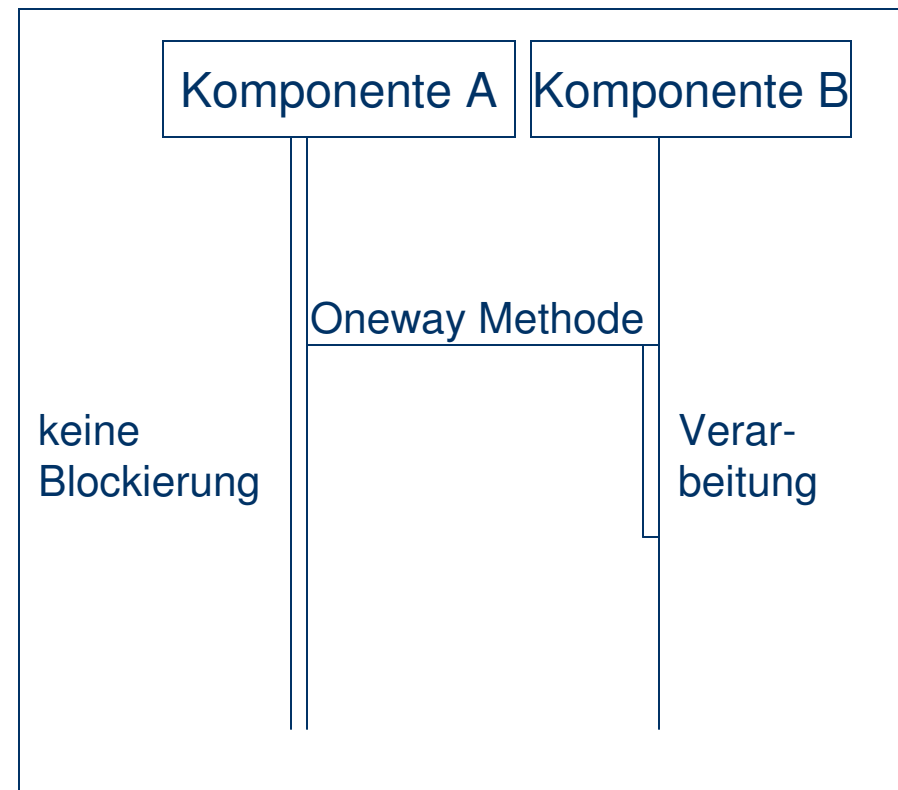
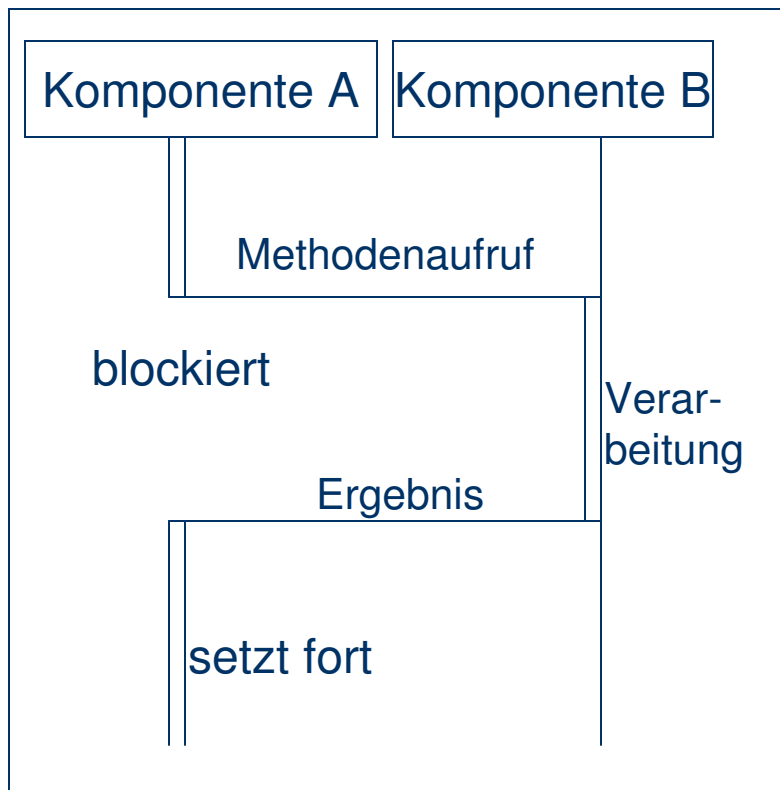
- Attribute

```
[readonly] attribute attribut_typ attributName;
```

- Readonly: Wert des Attributes kann nicht verändert werden

Interface Definition Language

- oneway Methoden



Interface Definition Language

- Syntax

```
enum Tage {  
    Montag,  
    Dienstag,  
    ...  
};  
  
struct Datum {  
    short Jahr,  
    short Monat,  
    long Millisekunde  
};
```

```
Union MeineUnion switch(long) {  
    case 1:  
        short myShort;  
    case 2:  
        double myDouble;  
    default:  
        string myString;  
};
```

Interface Definition Language

- Module, Schnittstellen und Vererbung

```
Module MyModule {  
    Interface A {...};  
    Interface B {...};  
  
    Interface C : A {  
        // eigene Attribute und Methoden  
    };  
    Interface C : A, B {  
        // eigene Attribute und Methoden  
    };  
};
```

Bewertung

- objektorientiert
- plattformunabhängig
- sprachunabhängig
- bei der Programmierung transparent

Danke für die Aufmerksamkeit!

- Fragen?
- Anmerkungen?