

# OMG Common Facilities

## Einsatz im Frameworking

Arif Chughtai, Dr. Bernhard Scheffold

# Inhalt

- Kontext der Common Facilities
- Überblick zu Common Facilities
- Common Facilities und FrameWorks
- Business Object Facility (BOCA)
- Weitere Common Facilities
- Literatur & Links
  
- Object Services

# Kontext der Common Facilities

**Object Management Group (OMG)**

## **Was tut die OMG:**

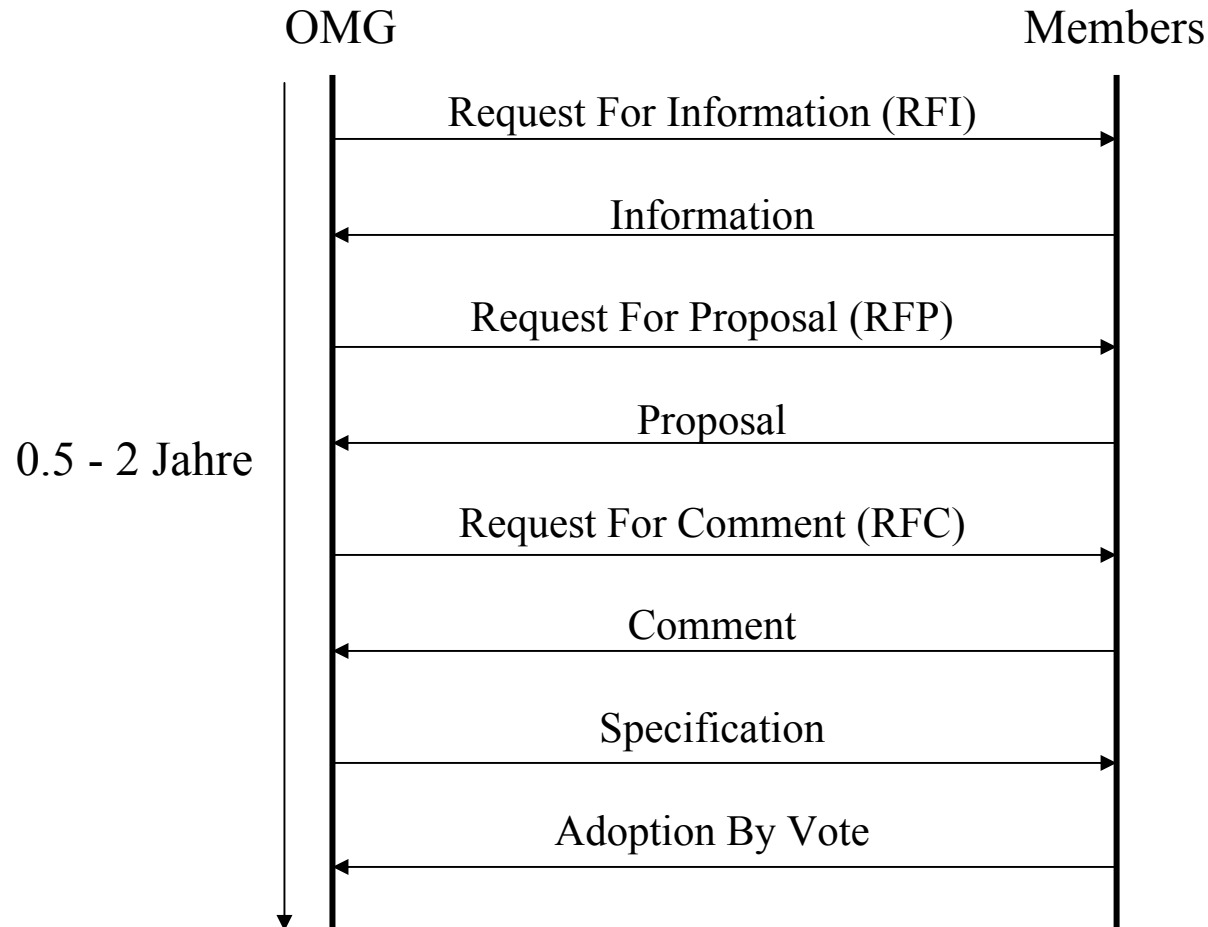
Es wird keine Software sondern es werden Spezifikationen (Standards) für OO-Software produziert

## **Ziele der OMG:**

Wiederverwendbarkeit, Portabilität und Interoperabilität von objekt-basierter Software in verteilten heterogenen Umgebungen.

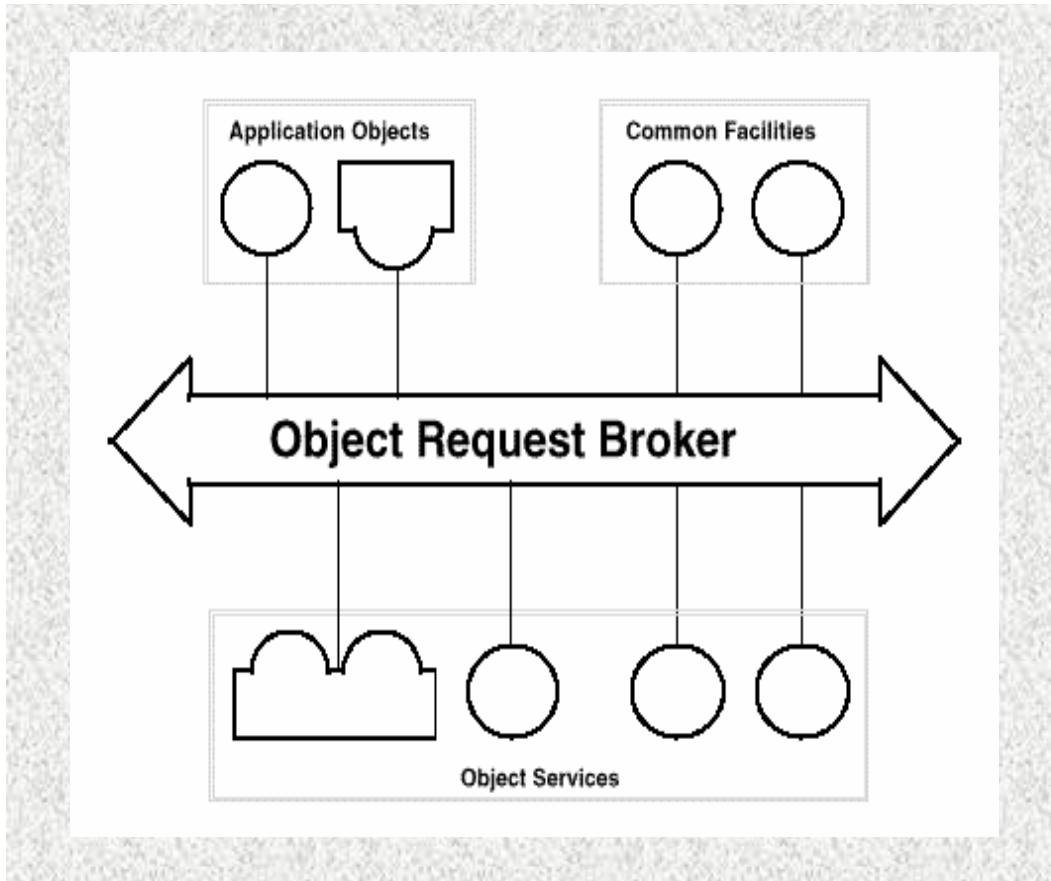
# Kontext der Common Facilities

## Entstehungsprozess einer OMG-Spezifikation



# Kontext der Common Facilities

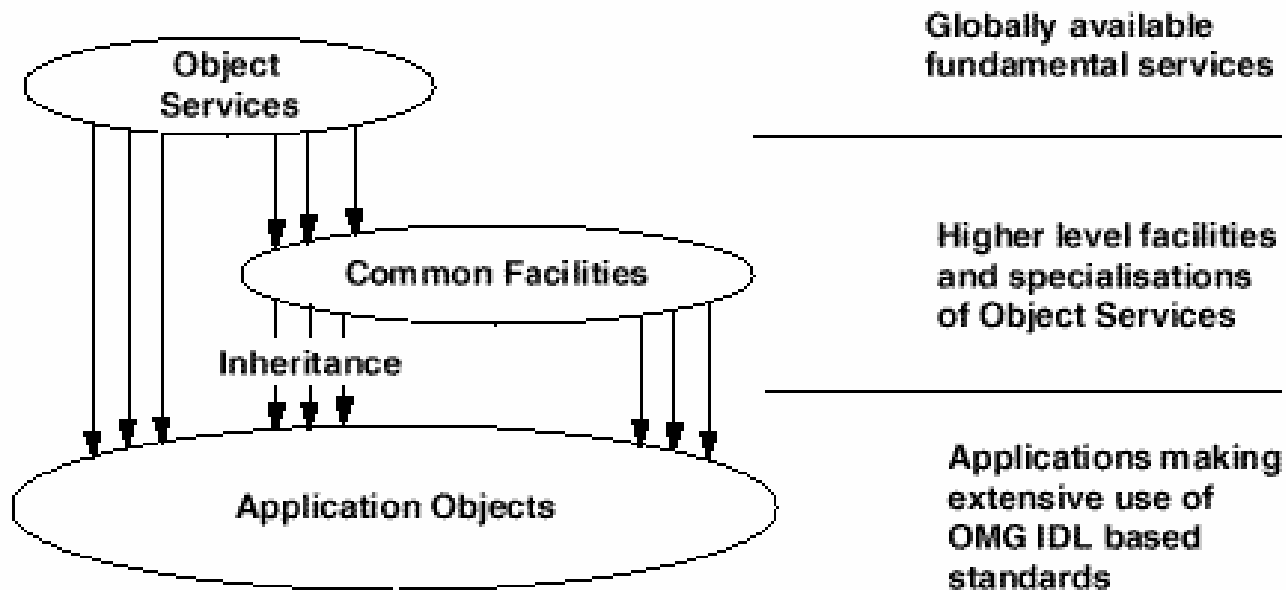
## Object Management Architecture (OMA)



Common Facilities sind **Business-nah**

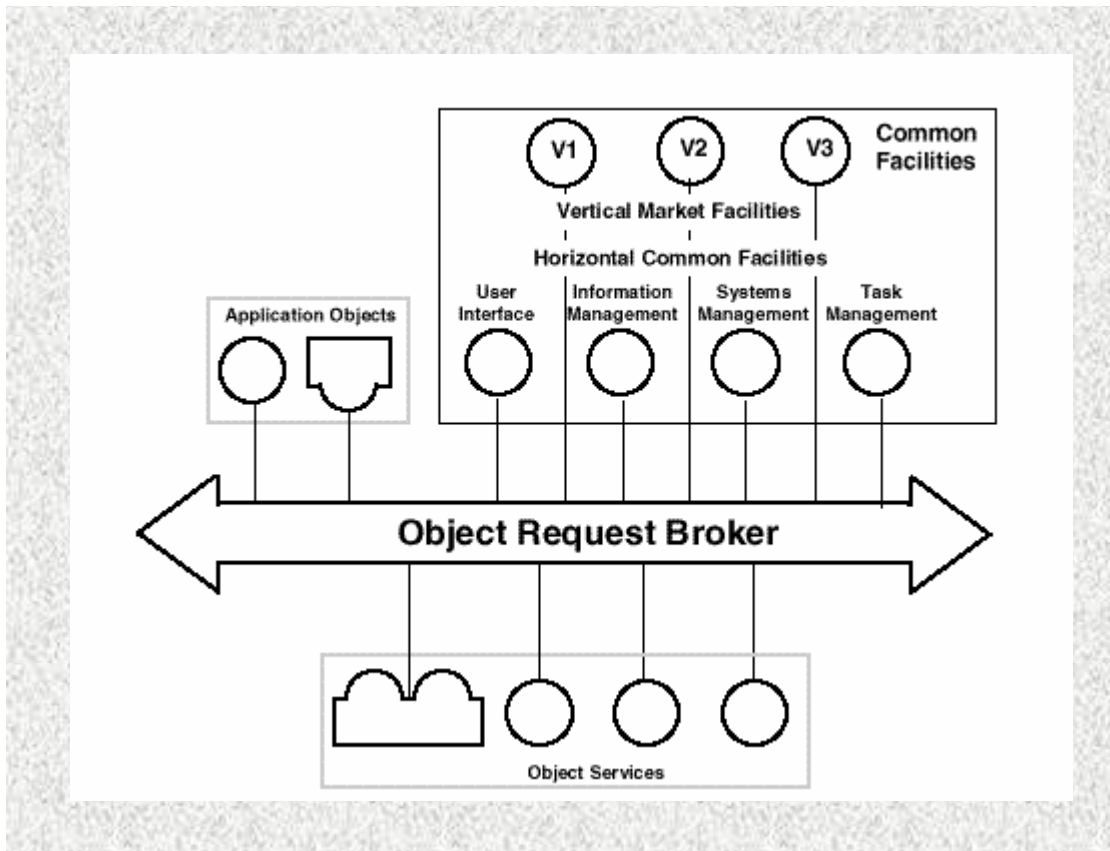
Object Services sind **Systemnah**

# Überblick zu Common Facilities



# Überblick zu Common Facilities

## Kategorien von Common Facilities



Vertical Market Facilities:

- Finance
- Information Superhighways
- Manufacturing
- Oil and Gas Industry Exploitation and Production
- ...

# Überblick zu Object Services

Life Cycle Service	: Anlegen und Zerstören
Persistence Service	: Speichern
Naming Service	: Lokalisieren
Event Service	: Zuteilen von Nachrichten
Concurrency Control	: Sperren von Threads oder Transaktionen
Transaction Service	: 2-Phasen-Commit
Relationship Service	: Erzeugen und Untersuchen von Beziehungen
Externalization Service	: Lesen und Schreiben von Daten



# Überblick zu Object Services

Query Service : Abfragen

Licensing Service : Verwendungskontrolle

Properties Service : Arbeiten mit Eigenschaften

# Object Services und Frameworks

## Einsatz der Object Services für Applikation-Framework-Dienste

Dienst	Object Service
	<b>Präsentation</b>
Visualisierung Modellschicht	Event Service, Naming Service, Property Service, Object Query Service
	<b>Model</b>
Dynamisches Rollenverhalten von Fachobjekten	Property Service
Aggregationsbeziehung zwischen Fachobjekten	Relationship Service, Collection Service, Query Service

# Object Services und Frameworks

## Einsatz der Object Services für Applikation-Framework-Dienste

Dienst	Object Service
	<b>Geschäftsvorfall/Workflow</b>
Ablauf von Geschäftsaktivitäten	Transaktion Service
Geschäftsaktivität	Transaktion Service
	<b>Persistenz</b>
Fachobjekt-ID aus DB	Persistence Service
Geänderte Fachobjekte zurückschreiben	Persistence Service

# Überblick zu Common Facilities

## **User Interface Common Facility:**

- Rendering Management
- Compound Document Presentation
- User Support
- Desktop Management
- Scripting

## **Information Management Common Facility:**

- Information Modeling
- Information Storage And Retrieval
- Compound Interchange
- Data Interchange
- Information Exchange
- Data Encoding And Representation
- Time Operations

## **System Management Common Facility:**

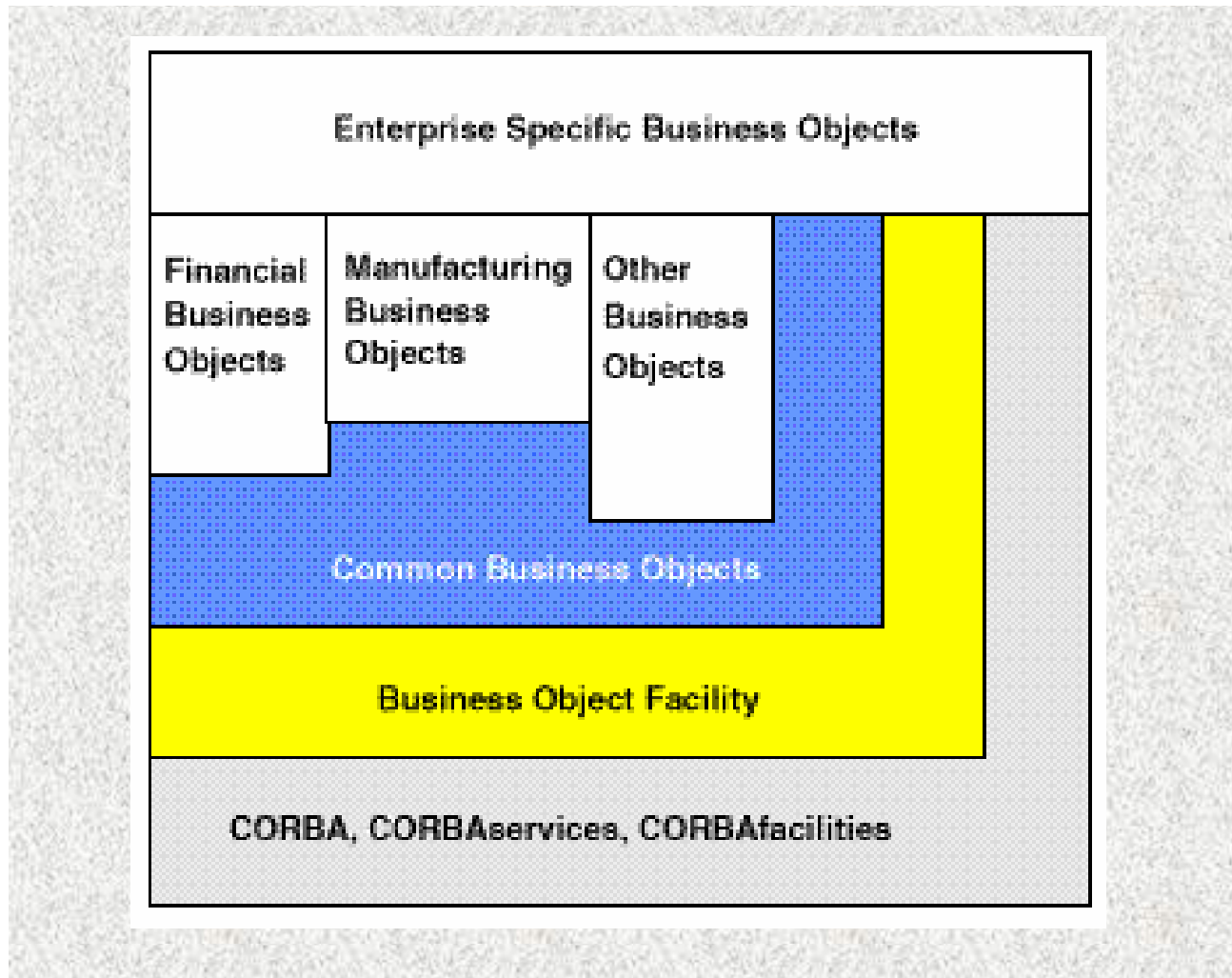
- Management Tools
- Collection Management
- Control

## **Task Management Common Facility:**

- Workflow
- Agent
- Rule Management
- Automation

Business Object Facility  
Common Business Objects

# Überblick zu Common Facilities



# Common Facilities und Frameworks

## Common Facilities in der Framework Spezialisierungshierarchie

Spezialisierungshierarchie	Common Facility Category	Beispiel für Common Facility
Fachliches Framework	Vertical	Manufacturing, Accounting
Application Framework	Horizontal	Workflow, Agent
Support Framework	Horizontal	Rendering Mngmt., Information Modeling

# Common Facilities und Frameworks

## Common Facilities in der Framework-Architektur-Schichten-Hierarchie

Architektur-Schichten-hierarchie	Common Facility Category	Beispiel für Common Facilities
Presentation	Horizontal	Rendering Mngmt., Desktop Mngmt.
Workflow	Horizontal	Workflow
Businessmodel	Horizontal/ Vertical	Data Interchange ,Manufacturing, Common Business Objects, Business Object Facility
DB Access	Horizontal	Information Modelling
Persistence	Horizontal	Information Storage & Retrieval

# Common Facilities und Frameworks

## Einsatz der Common Facilities für Applikation-Framework-Dienste

Dienst	Common Facility
	<b>Präsentation</b>
Visualisierung Modellschicht	Compound Presentation
Darstellung Fachobjekt kontextabhängig	Desktop Management
Darstellung Fachobjekt zustandsabhängig	Desktop Management
Aktivierung Oberflächen-elemente kontextabhängig	Desktop Management
Konsistenzprüfung	User Support
GUI-Mapping	Compound Presentation



# Common Facilities und Frameworks

## Einsatz der Common Facilities für Applikation-Framework-Dienste

Dienst	Common Facility
	<b>Präsentation</b>
Oberfläche erstellen	Rendering Management
Datenkonsistenz zwischen Model und View	Compound Presentation, Compound Interchange
	<b>Geschäftsvorfall/Workflow</b>
Ablauf von Geschäftsaktivitäten	Workflow
Geschäftsaktivität	Agent
	<b>Model</b>
Fachobjekte	Common Business Objects, Business Domain Objects
Dienste eines Fachobjektes	Common Business Objects, BOCA

# Common Facilities und Frameworks

## Einsatz der Common Facilities für Applikation-Framework-Dienste

Dienst	Common Facility
	<b>Model</b>
Konsistenz zwischen Attributen	Business Object Facility (BOCA-Rules)
Rollen eines Fachobjektes	Business Object Facility (BOCA-Roles)
Historisierung	Collection Management
Erstellung	Business Object Facility (BOCA-CDL)
Aggregationsbeziehung zwischen Fachobjekten	Business Object Facility (BOCA-Relationships)
Mehrsprachigkeit	Internationalization
Mehrwährungsfähigkeit	Internationalization, Financial Domain (Currency RFP)

# Abstimmungsstatus BOF

Vote Tally:	BOC
YES Votes:	52
NO Votes:	2
ABSTAIN Votes:	7
(179 voters)	
Assuming no further ABSTAIN's	BOC
YES votes needed to pass:	63
NO votes needed to fail:	56

# Wer hat BOCA eingereicht?

- Data Access Technologies, Inc.
- Electronic Data Systems (EDS)
- National Industrial Information Infrastructure Protocols (NIIP)
- SEMATECH, Inc.
- Genesis Development Corporation
- Prism Technologies
- IONA Technologies

# OMG Dokumente zu BOCA

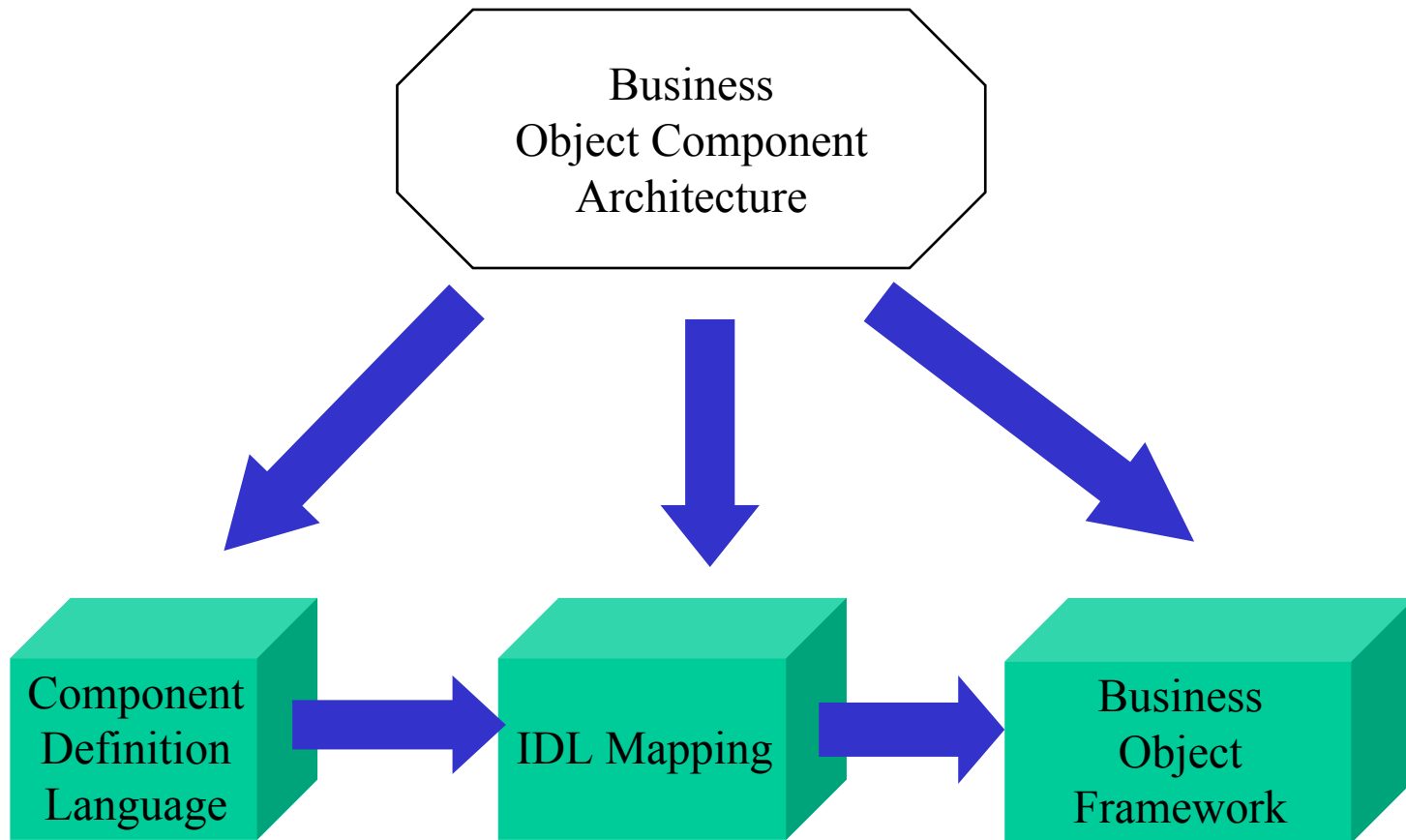
BOCA bom/98-01-07  
BOCA Errata bom/98-03-15  
BOCA Encyclopedia (non-normative) bom/98-01-08  
BOCA Presentation bom/98-02-06  
Interoperability Specification bom/98-03-08  
Interoperability Specification Errata bom/98-03-21  
Interoperability Specification Presentation bom/98-03-14  
Common Business Objects bom/98-03-02  
Common Business Objects Errata bom/98-04-01  
Common Business Objects Presentation bom/98-03-20

BOCA am 2. 4. 1998 verabschiedet,  
Interoperability Spec. und CBO stehen noch aus

# Ziele von BOCA

- Offener Markt für BO-Komponenten.
- Integration mit Internet und Intranets.
- Unternehmensweite Automation und Integration.
- Einfaches “Zusammenstöpseln” von Lösungen.
- Kostenreduktion bei Erzeugen, Ändern, SW-Verteilung.
- Reuse von Komponenten und Frameworks.
- Vereinfachte Entwicklung, SW-Verteilung und Benutzung.
- “Separation of concerns” zwischen BOen und Middleware.
- Interoperabilität zwischen unabhängig entwickelten BOen.
- Direkte Verbindung zwischen Modell und Komponenten.
- Unterstützung für massive Verteilung.

# Übersicht über BOCA-Vorschlag



# Component Definition Language

- CDL ist zentral für BOCA
- Interface und Semantik werden in CDL textuell beschrieben
- Assoziierte Semantik ist im Metamodell beschrieben
- CDL schirmt den BO-Entwickler von technischen IDL-Details ab
- Die CDL Grammatik ist eine Übermenge der OMG-IDL
- CDL ist eine deklarative Sprache



# CDL: Business-Objekt-Typen

- Subsysteme
- Entitäten
- Prozesse
- Business-Ereignisse

```
subsystem trucking
{ entity Truck
  { // Other truck stuff
    process MoveTruck
      { [readonly]
        relationship the_truck References Truck;
        relationship move_to References place;
        state_set moving {ready_to_move, moving, moved, move_aborted}
      };
  };
  void service_trucks ();
  entity location ();
};
```

```
[FROZEN, keys=purchase_id]
business_event Purchase
{ attribute string purchase_id;
  attribute Pounds amount_purchased;
  relationship item References product inverse purchase;
}
```

# Abhängige Typen

Abhängige Typen können nur “innerhalb” eines BOs existieren. Sie haben keine eindeutige ID und sind nur persistent oder transactional wenn es das BO, in dem sie enthalten sind, ist.

- Primitives (integer, string, float)
- Elementary Objects (Datum, Maß, Eigenschaft)
- Composite Objects (mehrere Primitives, Elementaries, oder Identifier)
- Immutable Objects (Elementary oder Composite)
- Collections (Set, Bag, List, Ulist, Array, Extent, Home)
- Command Objects

# BOCA-Features

- Operations (CORBA-Erweiterung der Operations um Pre-/Postconditions, ...)
- State Sets
- During
- Signals
- Attributes
- Relationships
- Apply
- Rules

```
entity airplane
{ state_set flight {parked, taxi, flying};

  during (parked) while_parked
  { state_set external_power {external_power_connected, internal_power};
  };
  during (flying)
  { attribute float altitude;
    state_set seat_belt {fasten_seat_belts, seat_belts_optional};
    void land_immediately ();
  };
};
```

# BOCA-Relationships

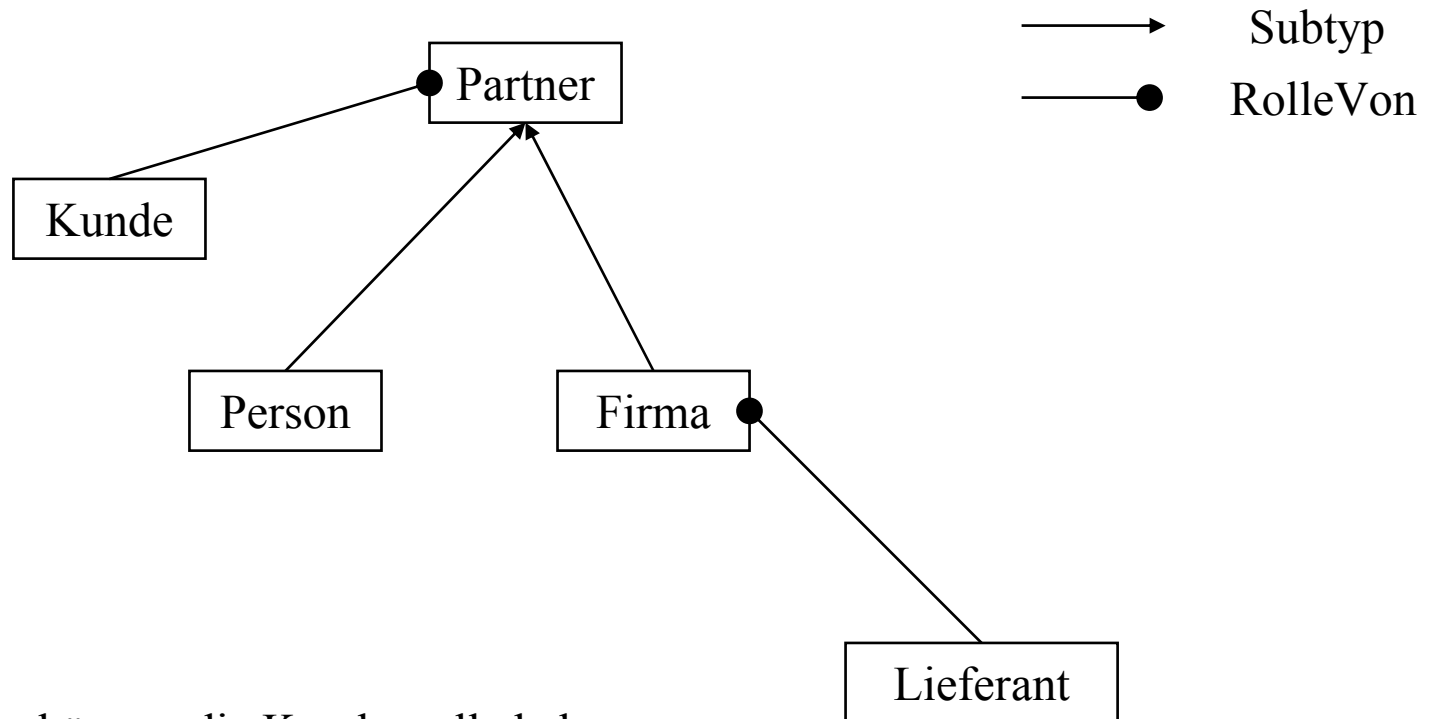
## Standardarten von Relationships

	Implicit Copy	Implicit Delete	Implicit Move	Inherit Spec	Inverse Type	is Navigable	Multiplicity
<b>Adapter</b>	N <sup>1</sup>	Y <sup>1</sup>	N <sup>1</sup>	N	N/A	Y	1..1
<b>RoleOf</b>	N	Y <sup>1</sup>	N	Y	HasRole	Y	1..1
<b>HasRole</b>	N	N	N	N <sup>1</sup>	RoleOf	Y	0..1 <sup>2</sup> to 1..*
<b>References</b>	N	N	N	N	Many	Y	0..1 <sup>2</sup> to 1..1
<b>Many</b>	N	N	N	N <sup>1</sup>	References	Y	0..* <sup>2</sup> to 1..*
<b>Composes</b>	N	N	N	N <sup>1</sup>	IsOwnedBy	Y	0..1 to 1..* <sup>2</sup>
<b>IsOwnedBy</b>	Y	Y <sup>1</sup>	Y	N	Composes	Y	1..1
<b>Aggregates</b>	N	N	N	N <sup>1</sup>	IsPartOf	Y	0..* to 1..* <sup>2</sup>
<b>IsPartOf</b>	Y	N	N	N	Aggregates	Y	0..1 to 1..1 <sup>2</sup>
<b>Uses</b>	N <sup>1</sup>	N <sup>1</sup>	N <sup>1</sup>	N <sup>1</sup>	N/A	N	0..1 , 1..1 <sup>2</sup> to 1..*

<sup>1</sup> -Locked

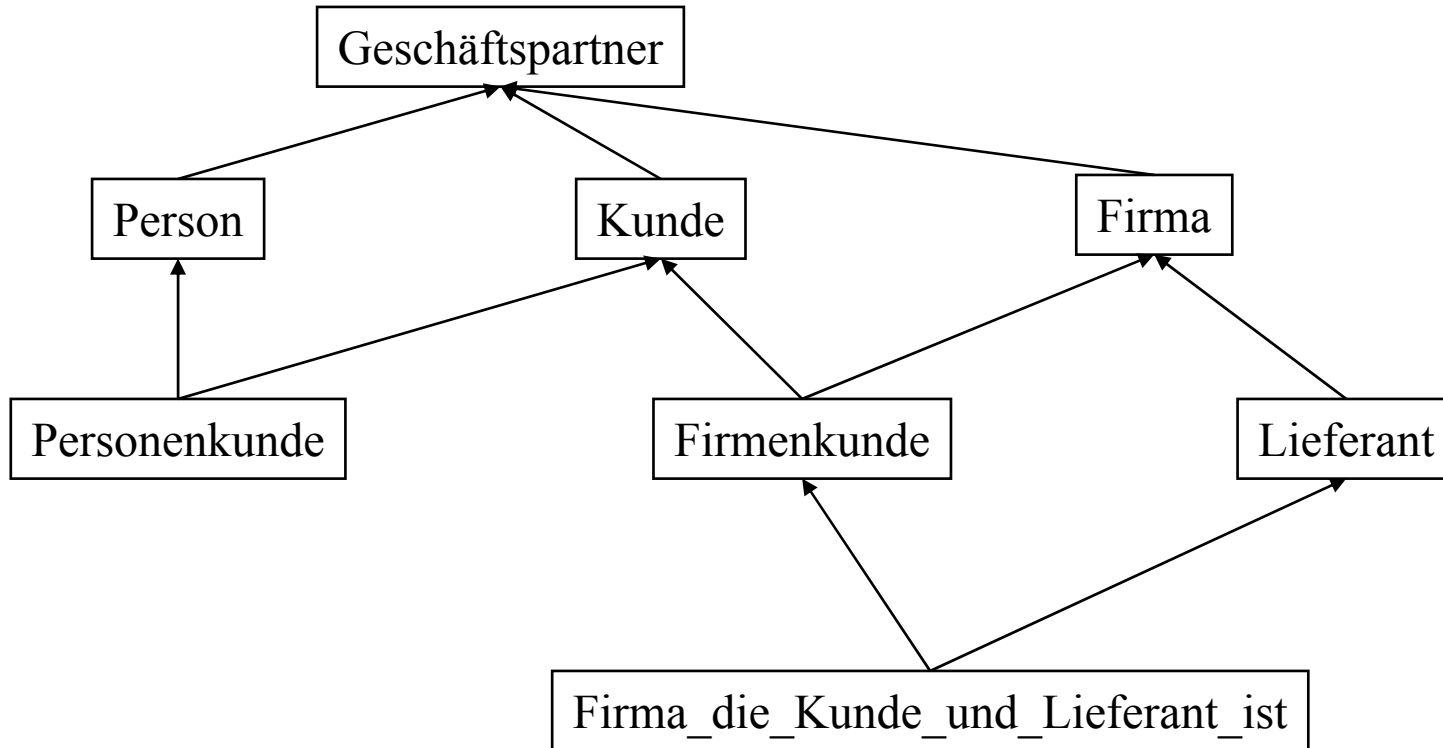
<sup>2</sup> - Default

# Role-Relationship



Alle Partner können die Kundenrolle haben,  
nur Firmen können die Lieferantenrolle haben

# Rollen mittels Vererbung



# Financial Domain

Financial Domain Task Force:

Currency RFP

Party Management RFP

General Ledger RFP

Risk Management RFI

# Literatur & Links

- The Essential Distributed Objects Survival Guide. Robert Orfali, Dan Harkey, Jeri Edwards. Wiley. ISBN 0-471-12993-3.
- CORBA Fundamentals and Programming. Jon Siegel. Wiley. ISBN 0-471-12148-7
- The Essential CORBA Systems Integration using Distributed Objects. Thomas J. Mowbray, RonZahavi. Wiley.
- Common Facilities Architecture. OMG. Revision 4.0, 1995
- <http://www.omg.org>
- <http://www.dataaccess.com/bodtf/LatestInfo.htm>