

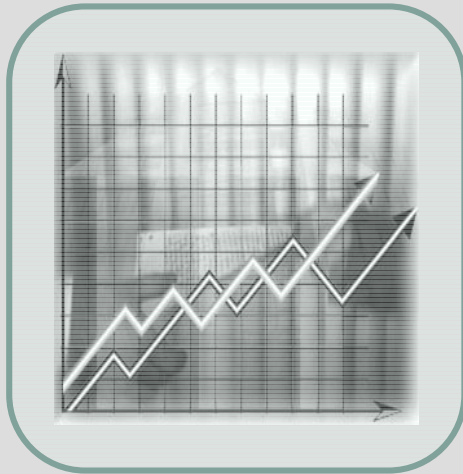
Applikationsentwicklung mit TWISTER 4.0



BROKAT Infosystems AG

Andreas Buttler

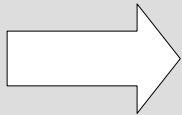
Program Manager



- ◆ **Gegründet am 17. September 1994 in Deutschland**
- ◆ **IPO am 17. September 1998 am Neuen Markt**
- ◆ **26 Tochtergesellschaften in 15 Ländern**
- ◆ **Mehr als 750 Mitarbeiter weltweit**
- ◆ **Marktkapitalisierung von über € 2,6 Milliarden (Juni 2000)**
- ◆ **Marktführer weltweit im Bereich e-Banking ***
- ◆ **Geplante Übernahme der beiden Firmen GemStone und Blaze (Anzahl der Mitarbeiter danach über 1100).**

* Quelle: Meridian Research

Die Produktbereiche von Brokat



◆ **Twister Division**

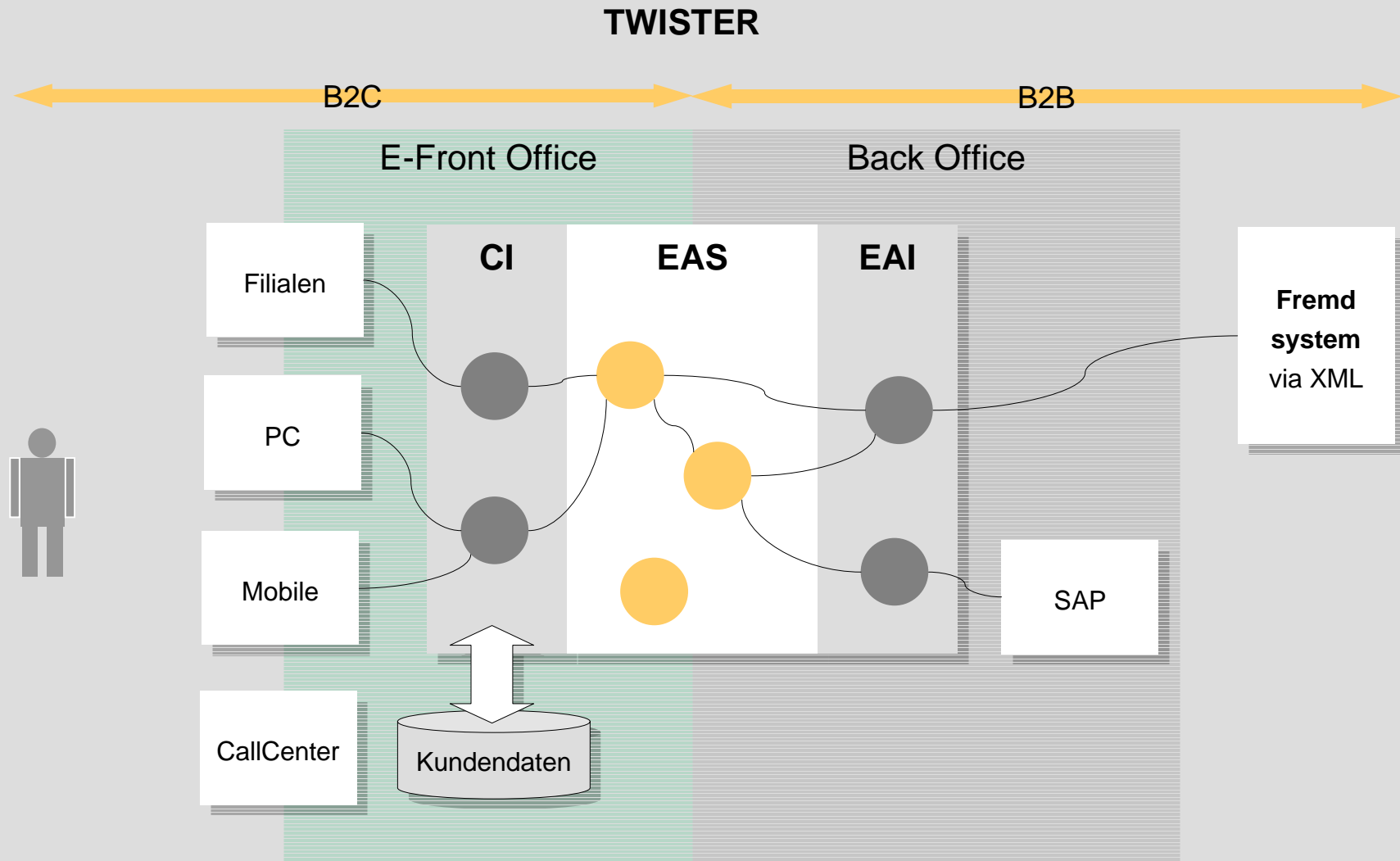
- ◆ Infrastruktur Software für das e-Business.

◆ **Financial Services Division**

- ◆ e-Banking, e-Brokerage Anwendungen (out-of-the-box)

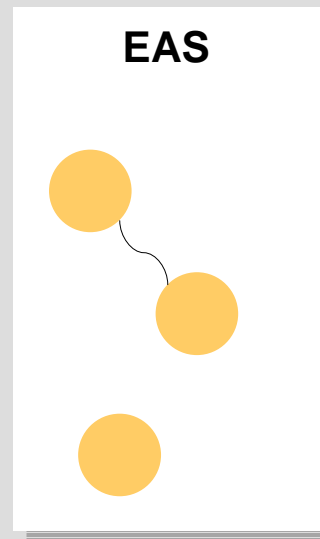
◆ **e- / m - Commerce Division**

- ◆ Payment Systeme
- ◆ Mobile digitale Signatur



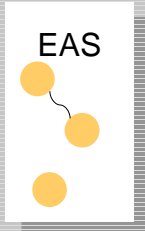
Implementieren der Business Logik

- ◆ Skalierbare, verteilte Architektur, lauffähig auf unterschiedlichen Betriebssystemen
- ◆ Zahlreiche Infrastruktur - Services
- ◆ Unterstützung offener und anerkannter Standards



- ◆ Sicherheit:
 - Verschlüsselte Kommunikation
 - Zugriffskontrolle auf Applikationsobjekte
 - Authentifizierungsmechanismen

- ◆ Unterschiedliche Komponentenmodelle und Programmiersprachen
- ◆ Graphische Entwicklungsumgebung (**IDE**)
- ◆ Graphische Administrations - und Überwachungswerkzeuge (**Management Center**)



Infrastruktur Services:

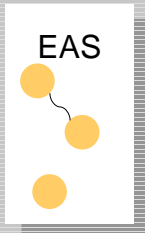
- ◆ **Naming Service**
- ◆ **Logging Service**
- ◆ **Repository Service**
- ◆ **Loadbalancing Service**
- ◆ **Licensing Service**

Base Services

Transparente Lastverteilung.
Ablaufumgebung für verteilte Objekte (Stateful und Stateless).

- ◆ **SNMP Service**
- ◆ **Event Service**
- ◆ **Transaction Service**

Extended Services



Folgende Komponentenmodelle stehen zur Verfügung:

◆ **Repository Defined Objects (RDOs)**

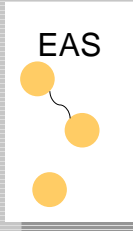
- ◆ Einfaches, Twister-eigenes Komponentenmodell
- ◆ Implementierungssprache: Tcl, **Java**, C++
- ◆ Strong Tagging

◆ **CORBA Objects**

- ◆ Twister ORB entspricht dem CORBA 2.2 Standard
- ◆ Twister IDL Compiler
- ◆ Implementierungssprache: Tcl, **Java**, C++
- ◆ Strong Typing

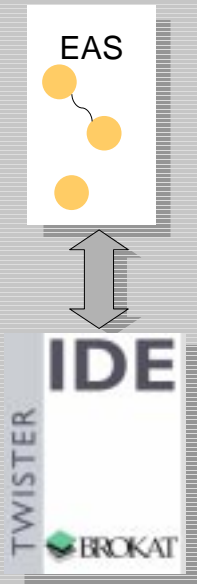
◆ **Enterprise Java Beans**

- ◆ Implementierungssprache: **Java**
- ◆ Strong Typing



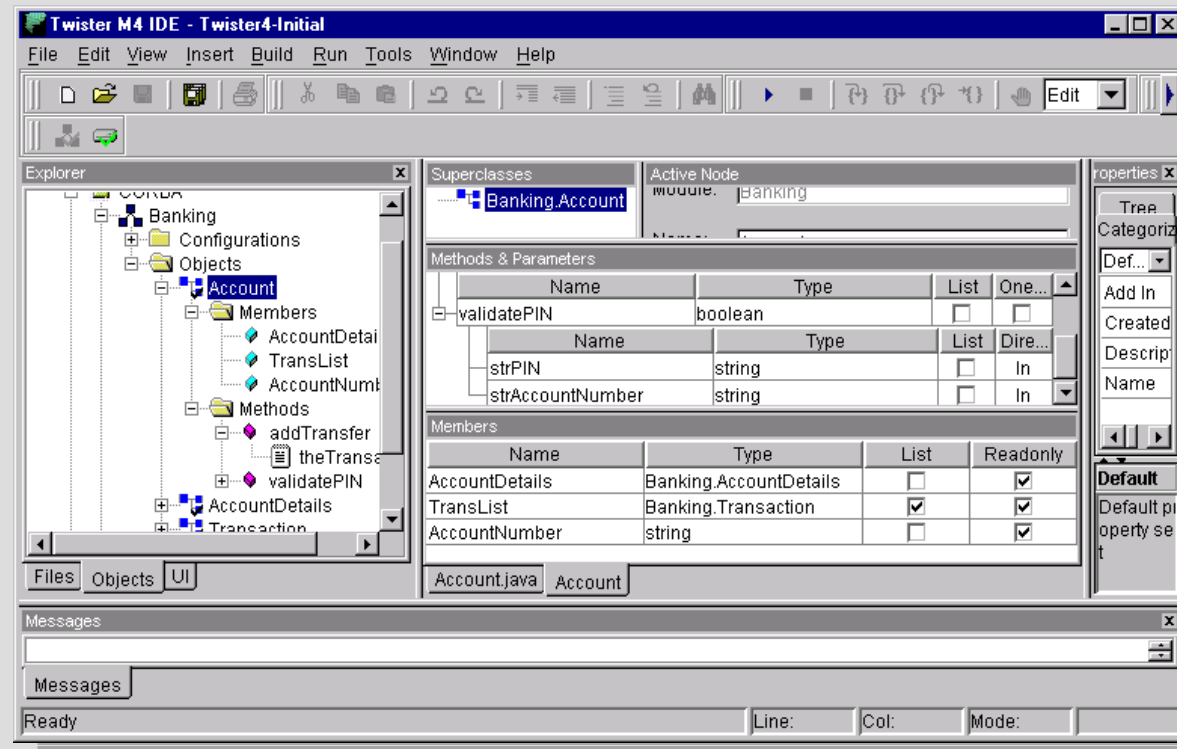
Komponentenmodelle im Vergleich:

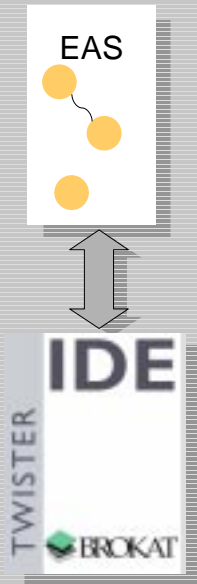
	RDO	EJB	CORBA
Verteilung objektspezifischer Stubs	-	✓	✓
Kompatibilitätsprüfung zur Übersetzungszeit	-	✓	✓
Interface-Evolution	✓	(✓)	(✓)
Standardisierte Schnittstellenbeschreibung	-	(✓)	✓



Phase 1 - Abstraktes Modellieren der Objekte (IDL like)

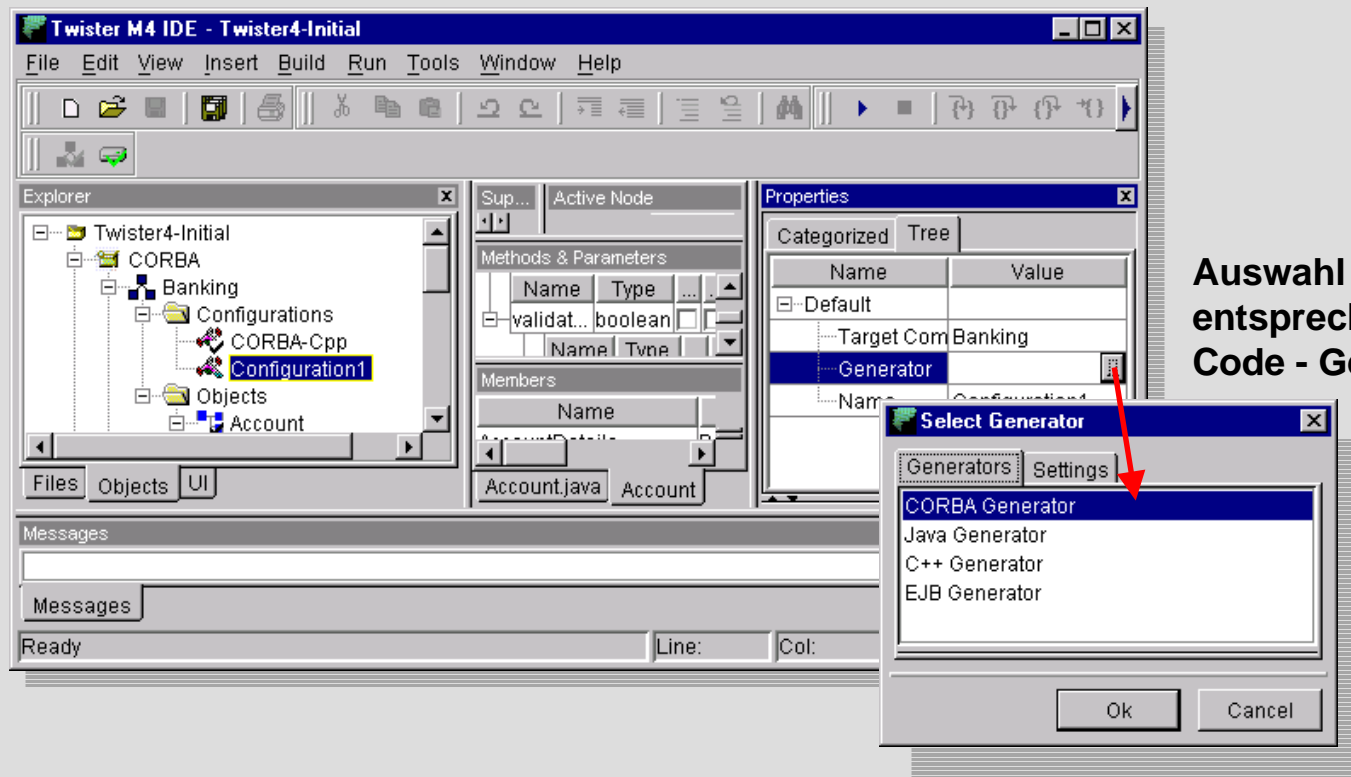
- ◆ Definition von Objekten, Methoden, Parametern und Mitgliedern
- ◆ Modellierung komponentenmodell - unabhängig -> abstrakt.





Phase 2 - Code Generierung

- ◆ Überführen eines Modells in das entsprechende Komponentenmodell.
 - RDO (Java, C++)
 - CORBA (Java, C++, TCL)
 - EJB

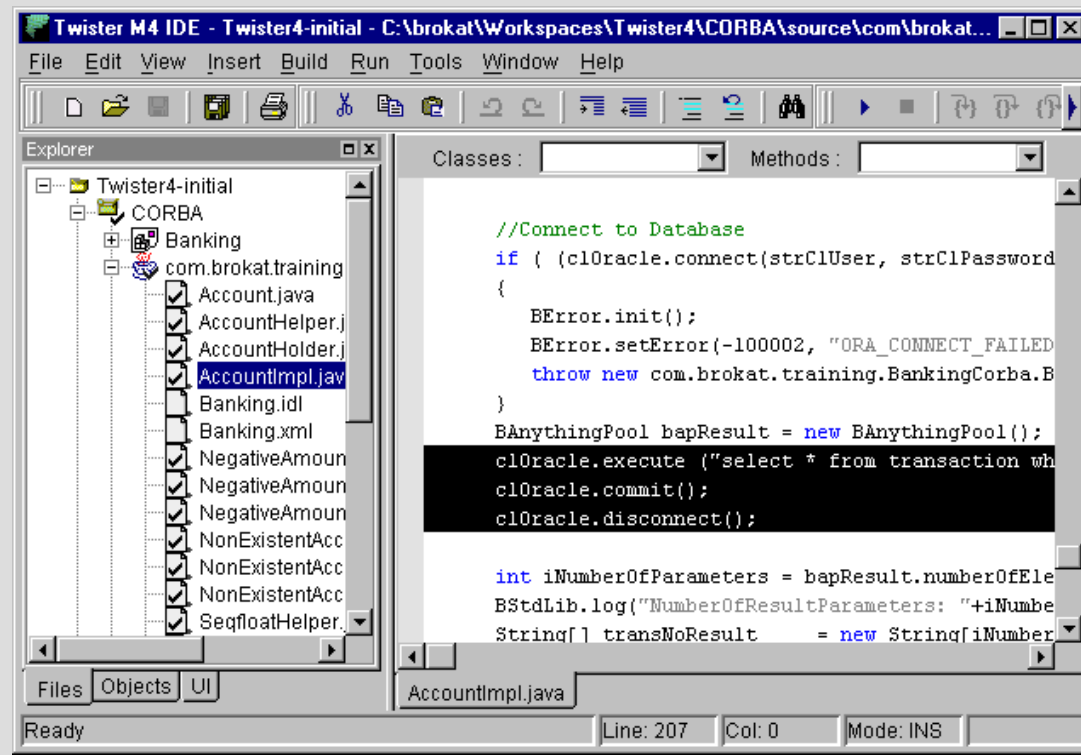
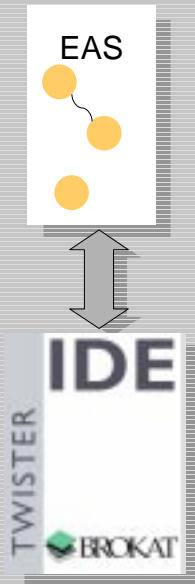


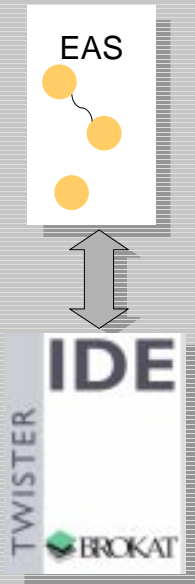
The screenshot shows the Twister M4 IDE interface. The Explorer window displays a project structure with 'Twister4-Initial' containing 'CORBA', 'Banking', 'Configurations', 'CORBA-Cpp', 'Objects', and 'Account'. The 'Configuration1' file is selected. The Properties window shows a table with 'Name' and 'Value' columns, where 'Generator' is selected. A 'Select Generator' dialog box is open, showing a list of generators: 'CORBA Generator', 'Java Generator', 'C++ Generator', and 'EJB Generator'. A red arrow points to the 'CORBA Generator' entry in the list.

Auswahl eines entsprechenden Code - Generators

Phase 3 - Einfügen der eigentlichen Funktionalität

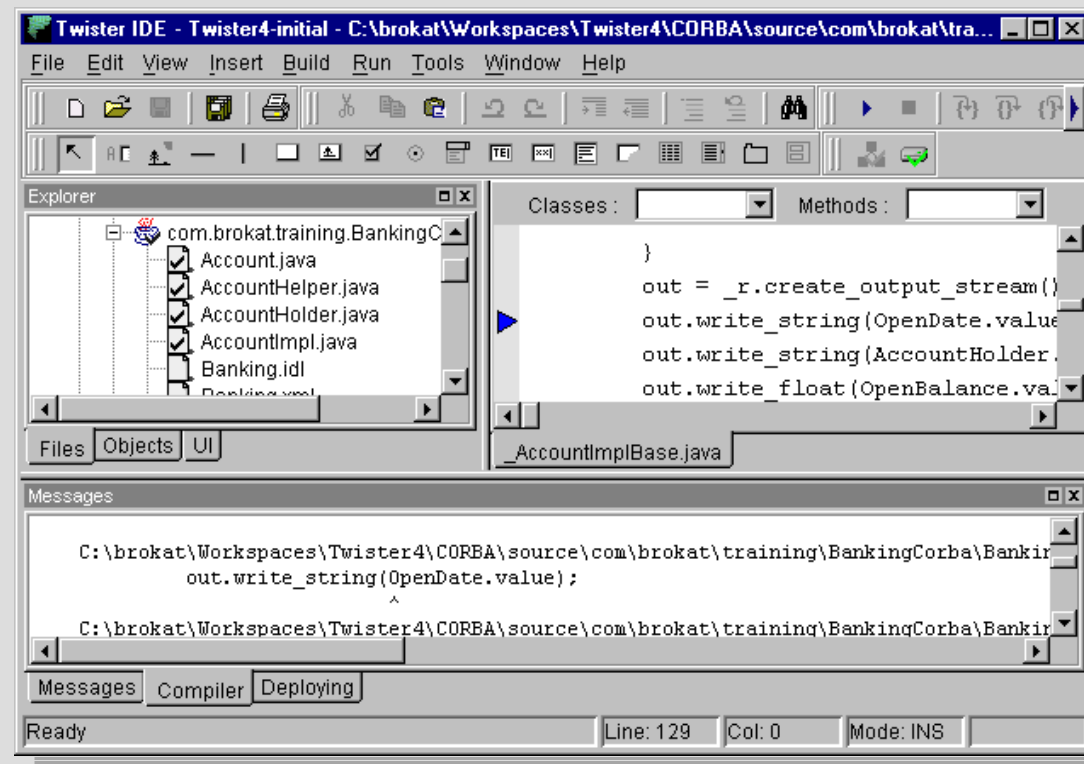
- ◆ Editor mit Syntax - Highlighting.
- ◆ Look & Feel an gängige Entwicklungsumgebungen anpassbar.
- ◆ Einbindung der gesamten Twister API.

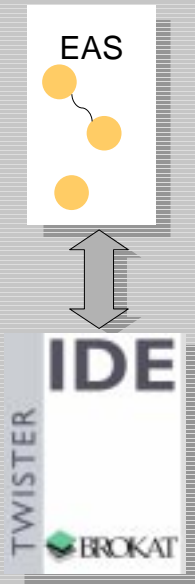




Phase 4 - Bauen der Komponenten

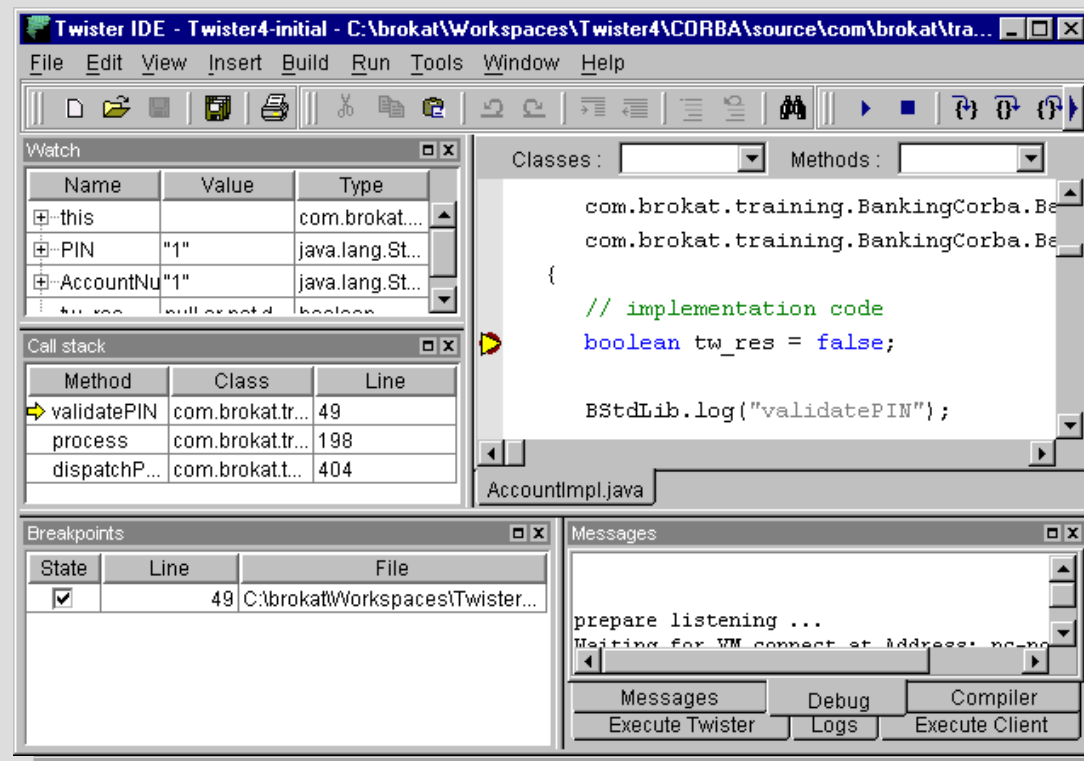
- ◆ Komplette Build - Umgebung für Java und C++.
 - Verwaltung von Projektdateien.
- ◆ Basierend auf 3rd Party Compilern.

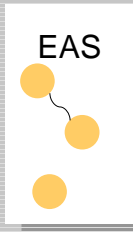




Phase 5 - Testen und Fehlersuche

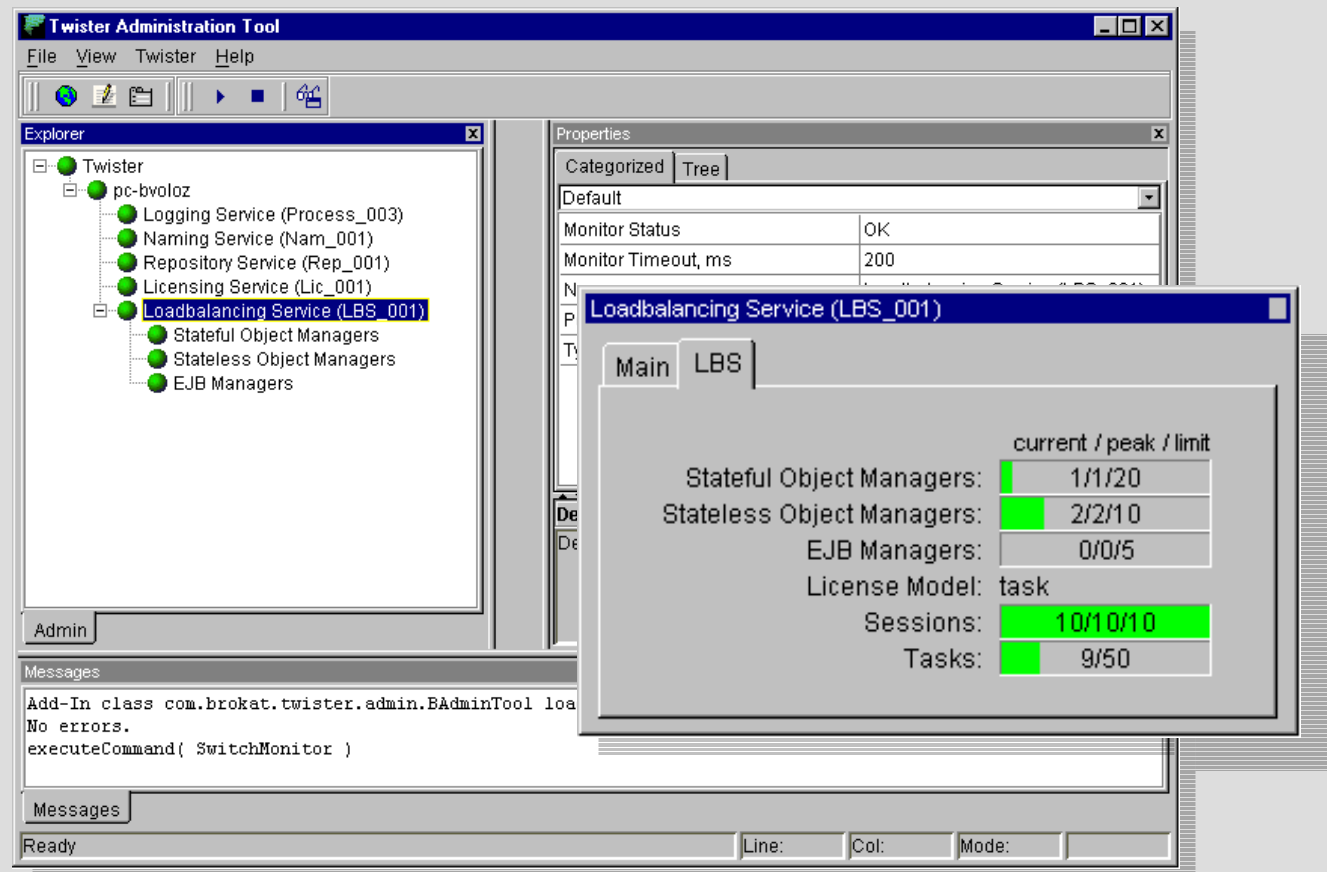
- ◆ Vollständige Integration der Twister Laufzeitumgebung.
 - Twister wird automatisch gestartet, wenn eine Komponente ablaufen soll.
 - Anzeige der zentralen Log - Einträge.
- ◆ Remote Debugging ermöglicht Fehlersuche 'vor Ort'.





Twister Management Center

- ◆ Konfigurieren, Administrieren und Überwachen.
- ◆ Remote - fähig



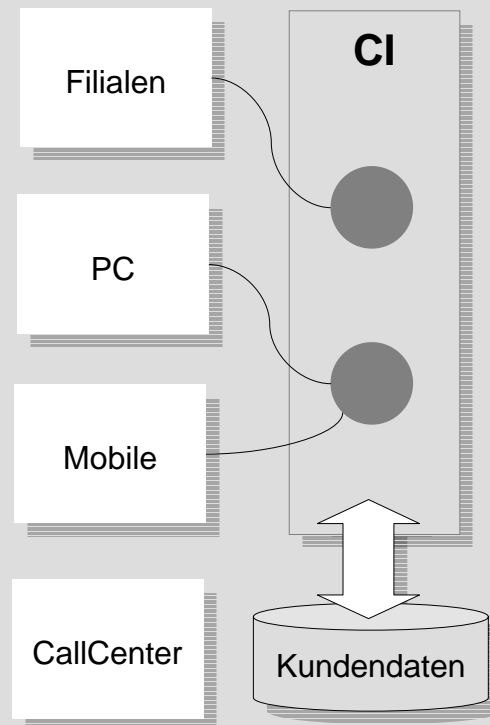
The screenshot displays the Twister Administration Tool interface. The Explorer pane shows a tree structure under 'Twister' with a sub-tree 'pc-bvoloz' containing several services. The 'Loadbalancing Service (LBS_001)' is selected and highlighted. The Properties pane shows details for this service, including 'Monitor Status: OK' and 'Monitor Timeout, ms: 200'. A detailed view of the 'Loadbalancing Service (LBS_001)' is overlaid, showing the following metrics:

	current / peak / limit
Stateful Object Managers:	1/1/20
Stateless Object Managers:	2/2/10
EJB Managers:	0/0/5
License Model:	task
Sessions:	10/10/10
Tasks:	9/50

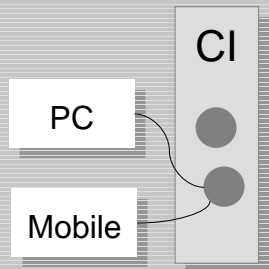
The Messages pane at the bottom shows the following log entry:

```
Add-In class com.brokat.twister.admin.BAdminTool load
No errors.
executeCommand( SwitchMonitor )
```

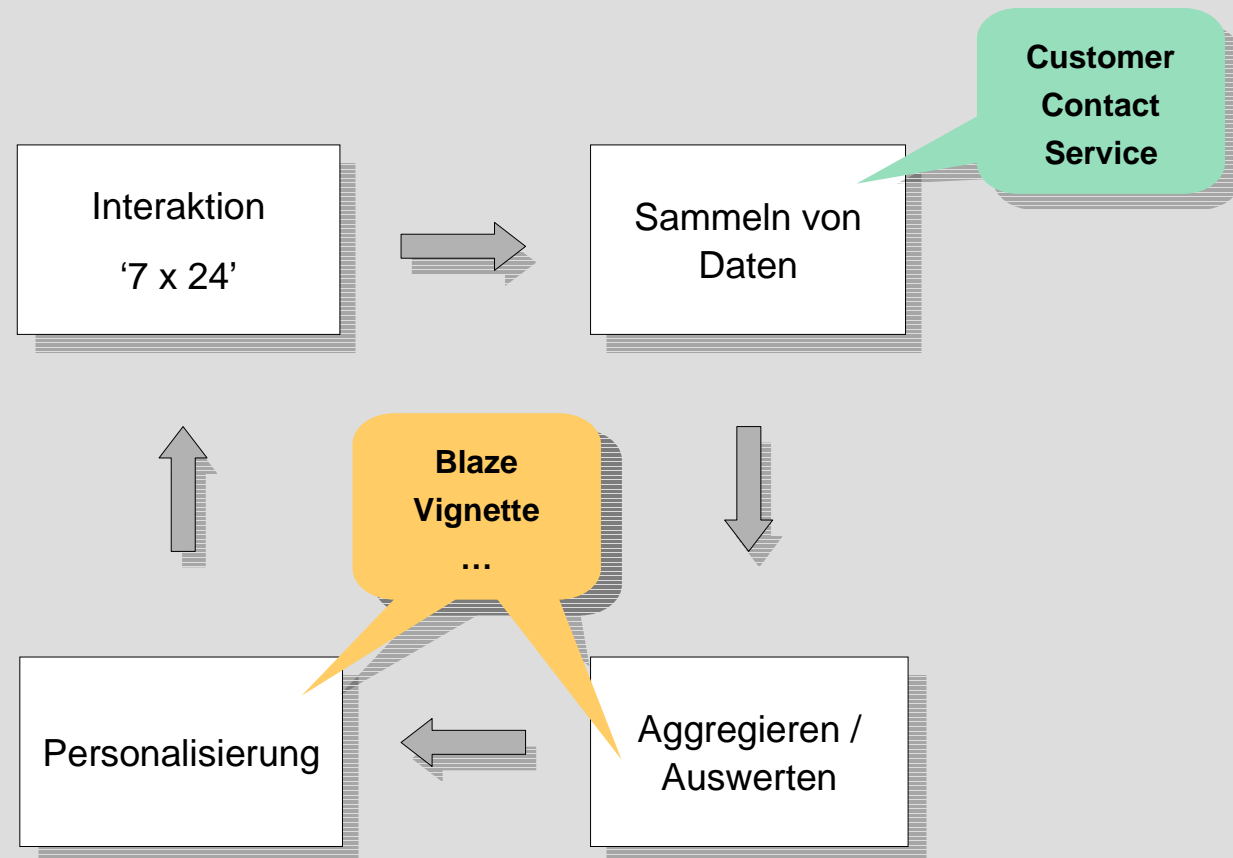
Anbindung von Vertriebskanälen

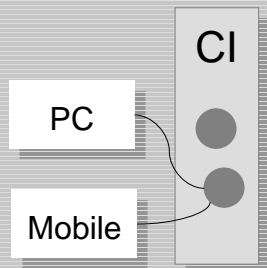


- ◆ Gateways für verschiedenste Endgeräte und Protokolle. (HTML, WAP, HBCI, IVR, ...)
- ◆ Separater Layer für kanal-spezifische Logik. (**Channel Alignment**)
- ◆ Entwicklung der Präsentationslogik wird durch die IDE unterstützt.
- ◆ Cross Channel Messaging.
- ◆ Automatisiertes Sammeln von Kundendaten für CRM - Zwecke (**Customer Contact Service**)



Customer Relation Management - Der CRM Loop



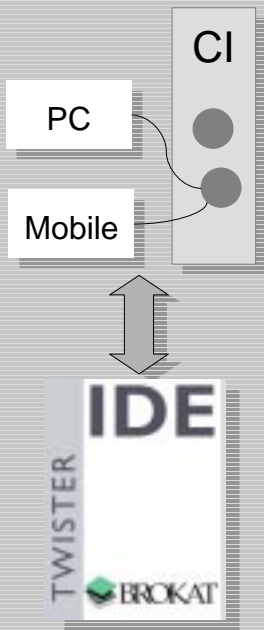


Call - Center Unterstützung mit X♦AGENT

- ◆ TeleWeb System für die 'real-time' Beratung.
- ◆ Gemeinschaftliches Browsen von Web - Seiten.
- ◆ Kontakt über Telefon, 'Voice over IP' oder Chat.

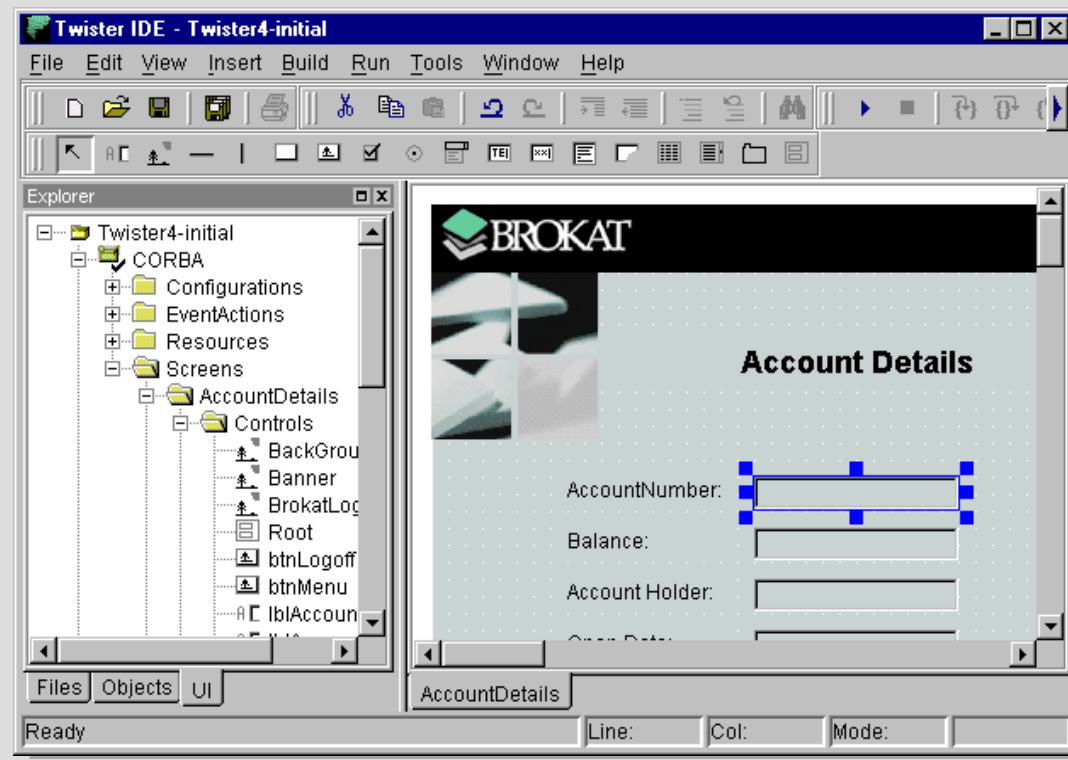


Customer's Desktop



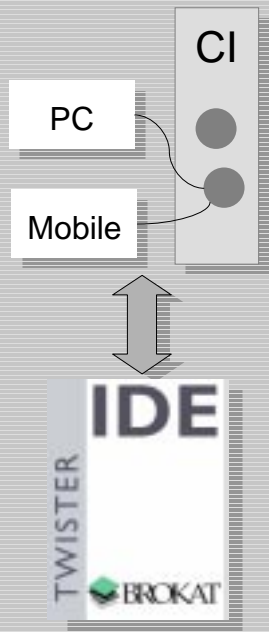
Phase 1 - Screen Design

- ◆ Anlegen einzelner Screens und Festlegung ihrer Reihenfolge.
- ◆ UI Komponenten plazieren mittels Drag&Drop.
- ◆ Template - Mechanismus ermöglicht schnelles Screen Design.



Phase 2 - Code Generierung

- ◆ Auswahl eines entsprechenden Generators.



UI Generators

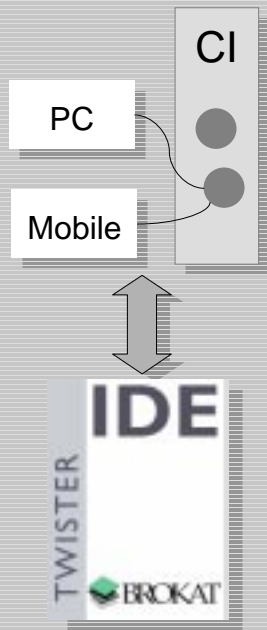
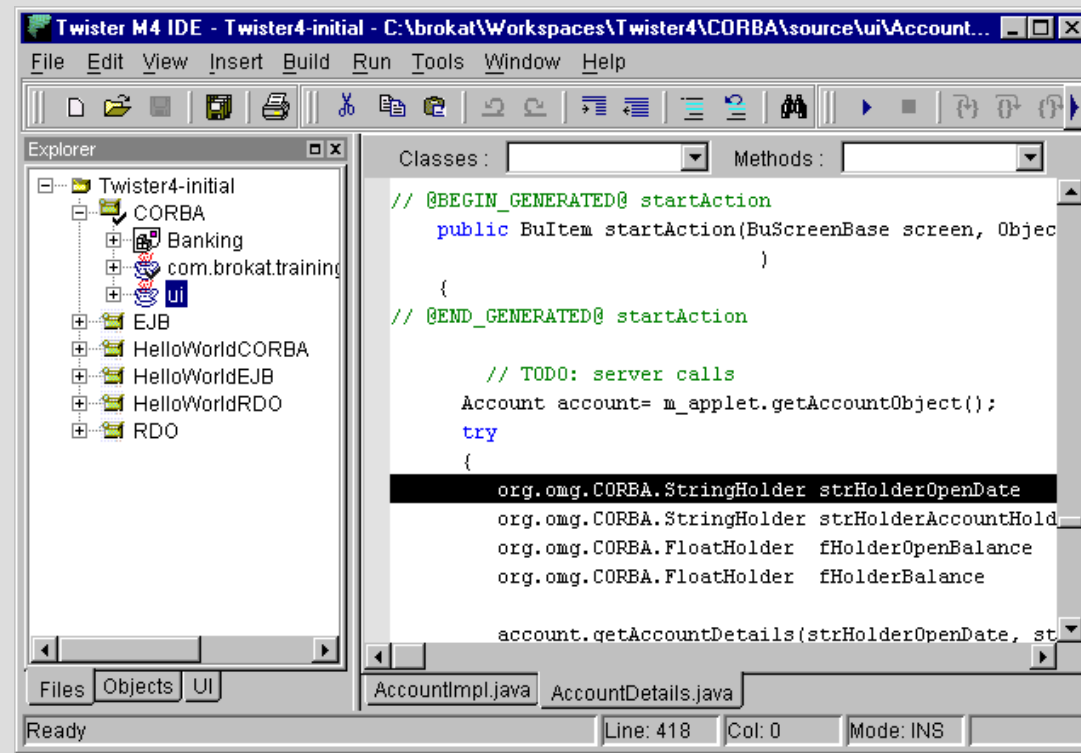
Java Applet

JSP pages (HTML)

JSP pages (WML)

Phase 3 - Einpflegen der Funtionalität

- ◆ Aufruf von Server Objekten und Vorbereitung der Daten für die Anzeige.
- ◆ Bei JSP - Lösungen befindet sich diese Logik in einer Bean.

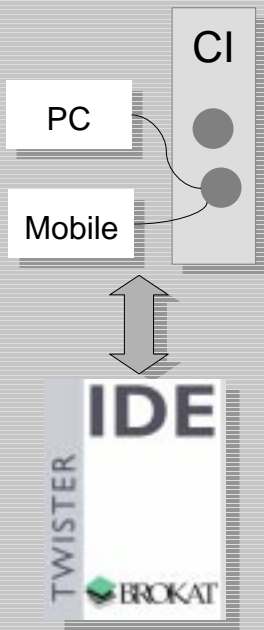



```

// @BEGIN_GENERATED@ startAction
public BuItem startAction(BuScreenBase screen, Objec
    )
{
// @END_GENERATED@ startAction

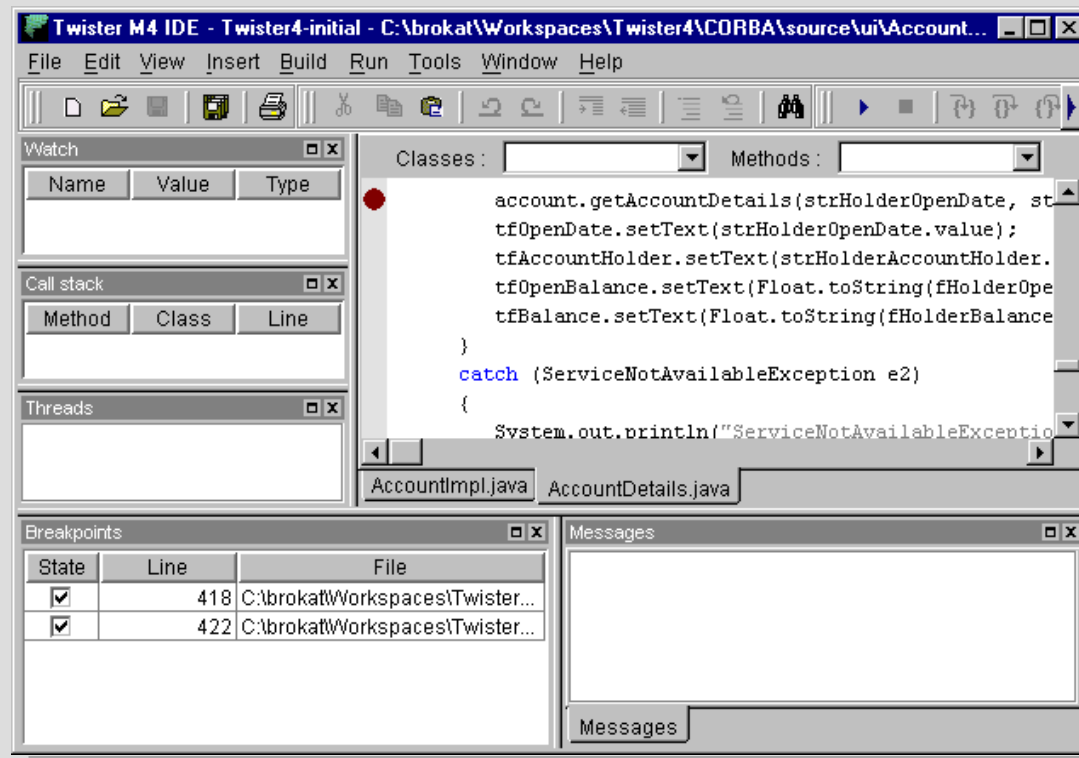
// TODO: server calls
Account account= m_applet.getAccountObject();
try
{
    org.omg.CORBA.StringHolder strHolderOpenDate
    org.omg.CORBA.StringHolder strHolderAccountHold
    org.omg.CORBA.FloatHolder fHolderOpenBalance
    org.omg.CORBA.FloatHolder fHolderBalance

    account.getAccountDetails(strHolderOpenDate, st
    
```



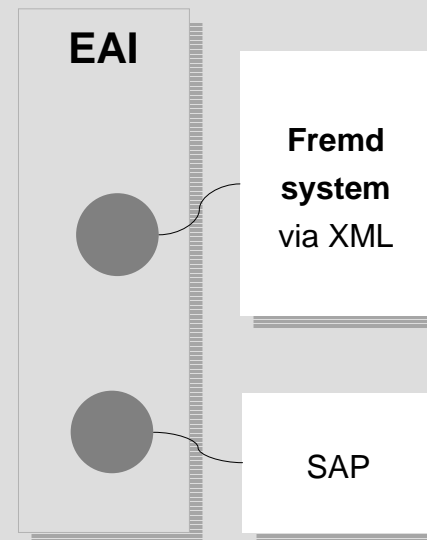
Phase 4 - Testen und Fehlersuche

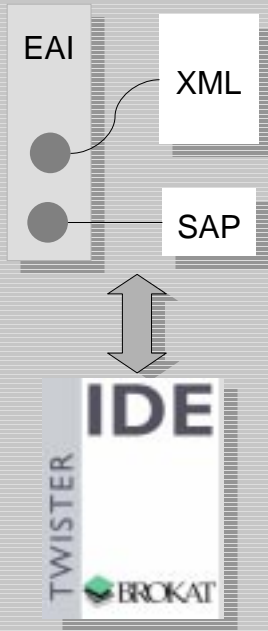
- ◆ Starten der gesamten Anwendung.
 - Sowohl der Client (Applet, HTML, WML) als auch Twister werden aus der IDE gestartet.
- ◆ Applet Debugging.



Zugriff auf Back-end - / Fremdsysteme

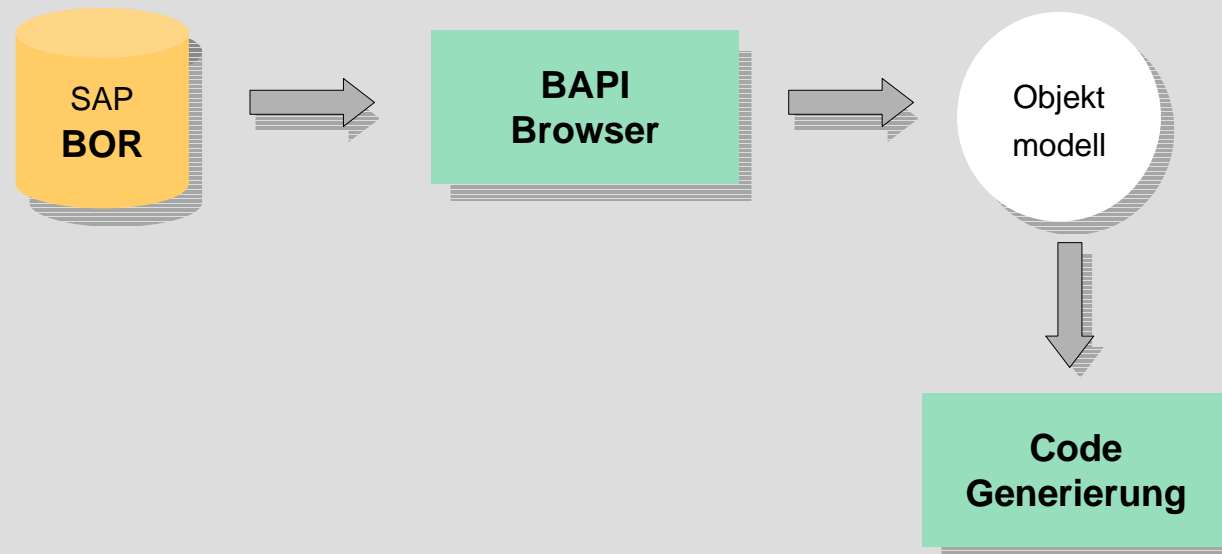
- ◆ Integration von Funktionalität, die bereits im Unternehmen vorhanden ist.
- ◆ Mehr als 40 sogenannte Accessoren, die den Zugriff auf Back-end Systeme erleichtern.
- ◆ Verteilte Transaktionen über mehrere Back-ends.
 - ◆ IDE - Unterstützung durch Wizards.
 - ◆ Austausch von Daten mit verbundenen Unternehmen (**B2B**).

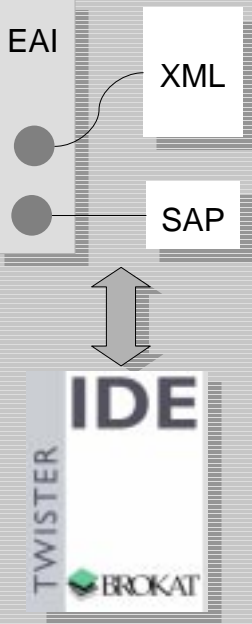




SAP Component Builder

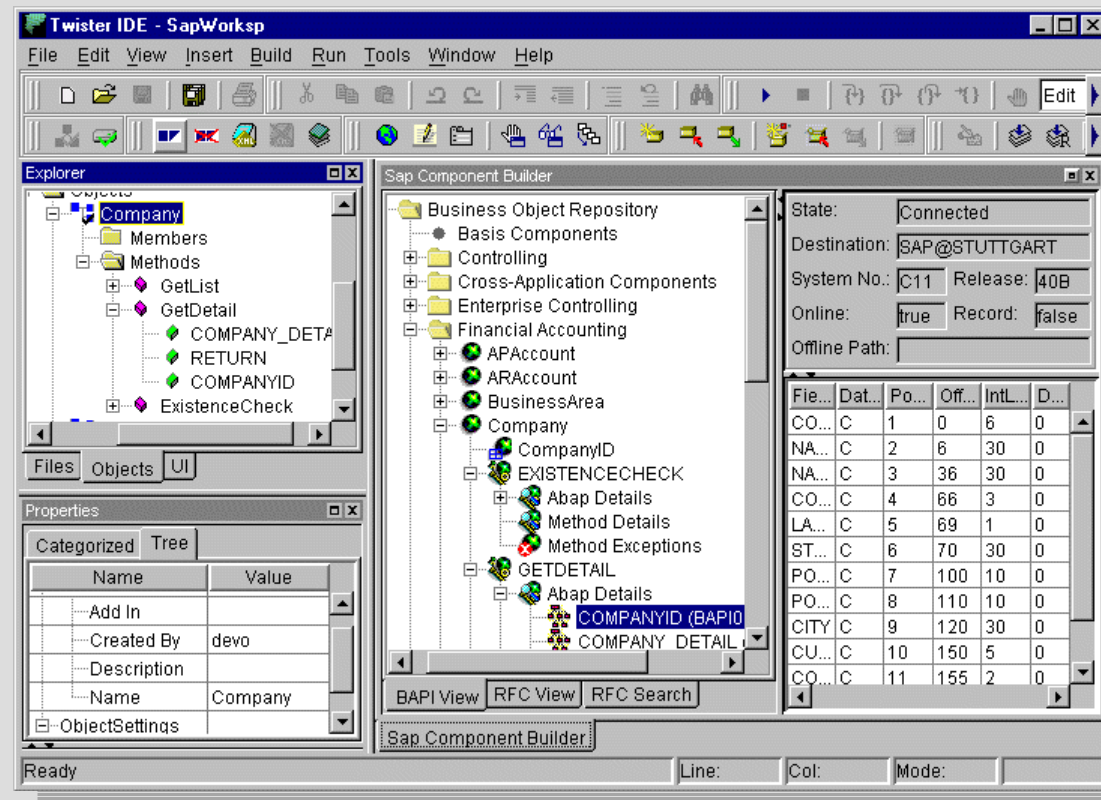
- ◆ Identifikation und Selektion von SAP Business Prozessen aus dem SAP BOR.
- ◆ Überführung in ein Objektmodell.
- ◆ Generierung von e-Business Anwendungscode für TWISTER.
- ◆ Zugriff aus allen Komponentenmodellen (CORBA, EJB und RDO)

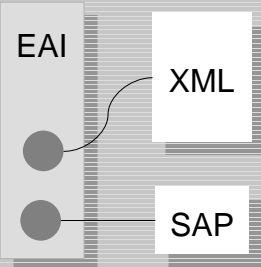




SAP Component Builder

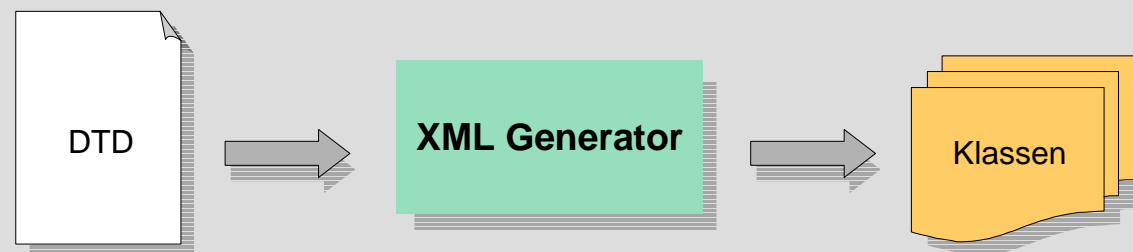
- ◆ Mapping einer funktionalen Sicht in eine objektorientierte Sicht.
- ◆ Offline-Entwicklung von e-Business Anwendungen für R/3.
- ◆ Senkt Entwicklungsaufwand durch automatische Codegenerierung.



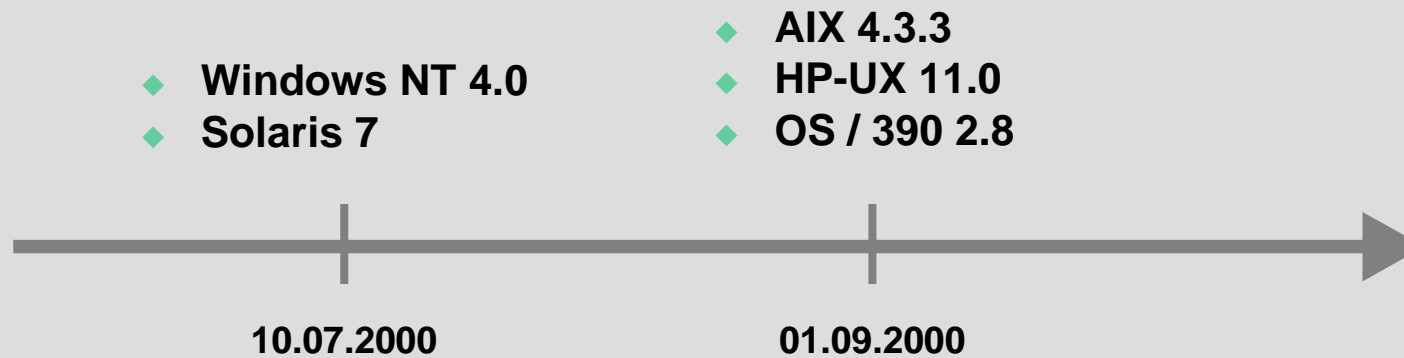


TWISTER B2B Support

- ◆ Automatisches Generieren von Java Klassen anhand einer DTD.
- ◆ Die generierten Klassen stellen das Abbild einer XML Nachricht dar.
- ◆ Entwickler nutzen die Klassen um XML Messages zu empfangen, zu verarbeiten und zu senden.
- ◆ Keine Notwendigkeit für SAX oder DOM Programmierung.



Twister 4.0 - Verfügbarkeit



BROKAT Infosystems AG
www.brokat.com

Danke !

Fragen ???

