

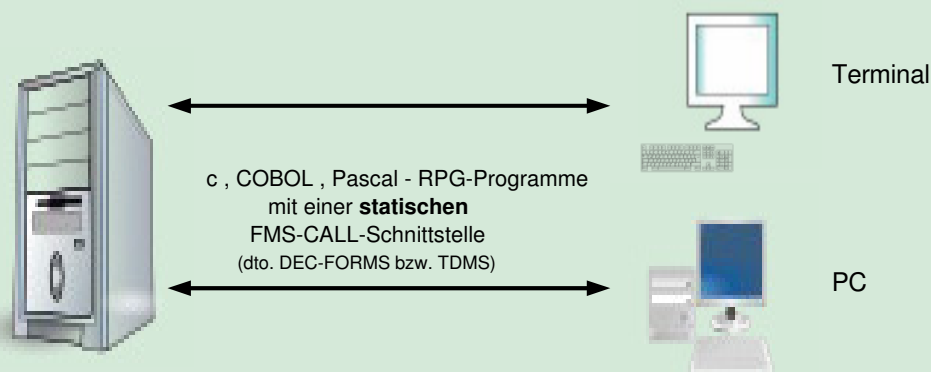
# Plattformunabhängig durch Software-Migration

## vom FMS-Charactermodus zu GUI bzw. Web-Oberflächen

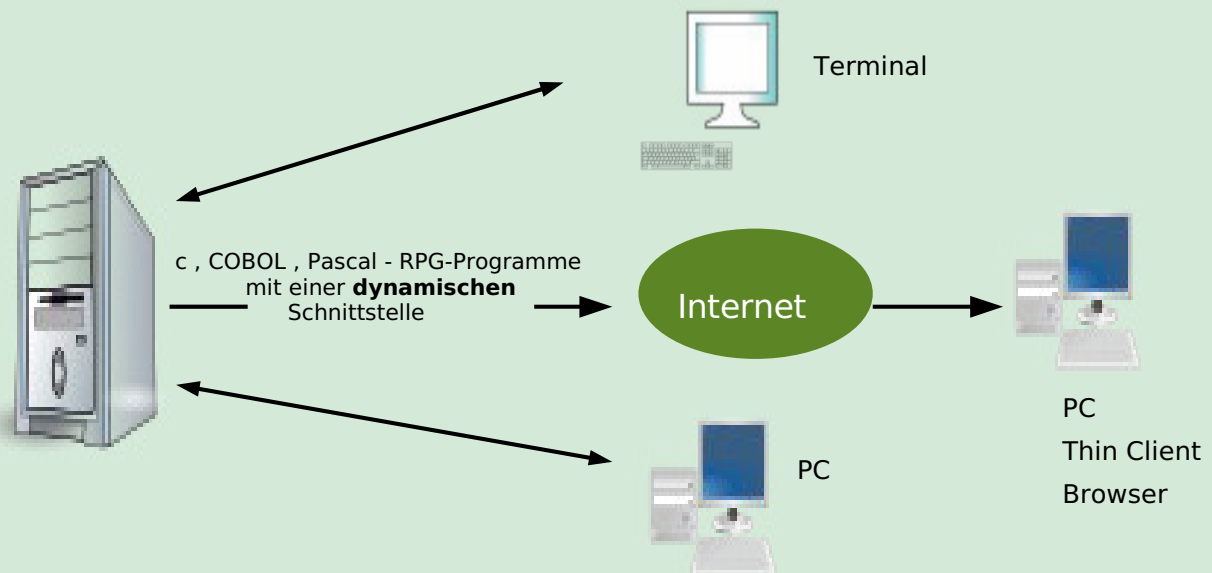
Enno Richter  
Software-Entwicklung

Vortrag vom 18. April 2007 in Nürnberg  
DECUS München e.V.

## Die 'traditionelle' Architektur



## Die 'flexible' Architektur



www.richter-software.de

## Die Alternativen

- GUI mit Hilfe diverser Programmiersprachen
  - Visual Basic (MONO-Projekt)
  - Python (o.a.) incl. Qt4
  - Java
    - Swing
    - RCP (Eclipse)
  - diverse.....
- Grafische Darstellung durch Browser
  - (Web-Oberfläche – Web-Server (Tomcat etc.), AJAX)
  - PHP
  - Java Server Faces / Java Server Pages
  - Java Script
    - mit Unterstützung verschiedener Frameworks
    - (> 50 Alternativen)

## Die Alternativen

### ● GUI mit Hilfe diverser Programmiersprachen

- Visual Basic (MONO-Projekt)
- Python (o.a.) incl. Qt4
- Java
- Swing
- RCP (Eclipse)
- diverse.....

Swing ← .....  
RCP (Eclipse)

### ● Grafische Darstellung durch Browser

(Web-Oberfläche – Web-Server (Tomcat etc.), AJAX)

- PHP
- Java Server Faces / Java Server Pages
- Java Script  
mit Unterstützung verschiedenster Frameworks  
(> 50 Alternativen)

www.richter-software.de

## Die notwendigen Software - Moduln

- 'socket' - Software
- Java - Compiler
- JSON - Software

## Die notwendigen Software - Moduln

- 'socket' - Software z.B. im AcuCOBOL enthalten ("c\$socket"-Routinen)
- Java - Compiler Auf allen Plattformen 'frei' verfügbar. Enthält "socket"-Bibliotheksroutinen.
- JSON - Software 'Java Script Object Notation' ein Protokoll (ähnlich XML).

## 'socket' - Software

- Wird in Form von Bibliotheksroutinen für (fast) alle gängigen Programmiersprachen bereitgestellt.
- Erlaubt die Verbindung von Programmen systemintern aber auch über Systemgrenzen hinweg !
- Ermöglicht das Erstellen von Software-Services. (SOA - Prinzip)

## Java - Compiler

- Ein Java-Compiler ist auf allen Systemplattformen verfügbar.
- Java ist kostenfrei und (inzwischen) auch 'open Source'.
- Die Java-Performance ist im Dialogbereich absolut ausreichend.

## JSON - Protokoll

- JSON (JavaScript Object Notation) ist ein schlankes Datenaustauschformat, das für Menschen einfach zu lesen und zu schreiben und für Maschinen einfach zu parsen (Analysieren von Datenstrukturen) und zu generieren ist. Es basiert auf einer Untermenge der JavaScript Programmiersprache, (Standard ECMA-262 dritte Edition - Dezember 1999).
- Das JSON-Protokoll baut auf zwei Strukturen auf.
  - Name/Wert Paare. In verschiedenen Sprachen wird dies realisiert als ein Objekt (object), Satz (record), Struktur (struct), Wörterbuch bzw. Verzeichnis (dictionary), Hash-Tabelle (hash table), Schlüssel-Liste (keyed list) oder als ein assoziatives Array (associative array).
  - Eine geordnete Liste von Werten. In den meisten Sprachen wird das als Array (array), Vektor (vector), Liste (list) oder Sequenz (sequence) realisiert.
- Das JSON-Protokoll ist einfacher und kompakter als das XML-Protokoll.

## JSON – Protokoll unter COBOL

- Es gibt für alle gängigen Programmiersprachen (außer COBOL) Bibliotheks-Routinen die einfach in die entsprechenden objektorientierten Programme integriert werden können.  
Die Methoden der bereitgestellten Klassen bieten ein einfache Verarbeitung der Daten !

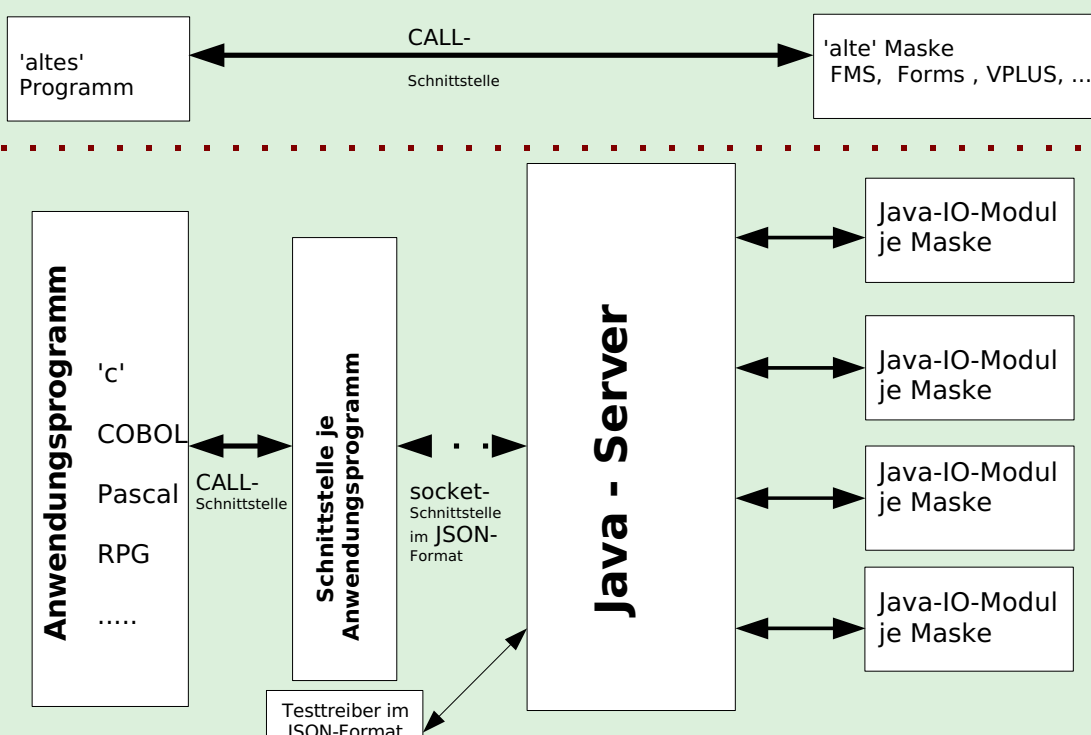
- Ein Beispiel..

```
[{"SysBenutzerName": "EnnoRichter", "SysFunktionsCode": 130},
{"mapName": "n012m", "mapHeaderText": "n012m"},
[{"varName": "FX0001", "varInhalt": "aaaa"}, {"varName": "FX0002",
"varInhalt": "bbbb"}, {"varName": "FX0003", "varInhalt": "CCCC"}....]]
```

- Eine JSON-Schnittstelle für COBOL ist in der Entwicklung.  
Ein Prototyp ist einsatzbereit !

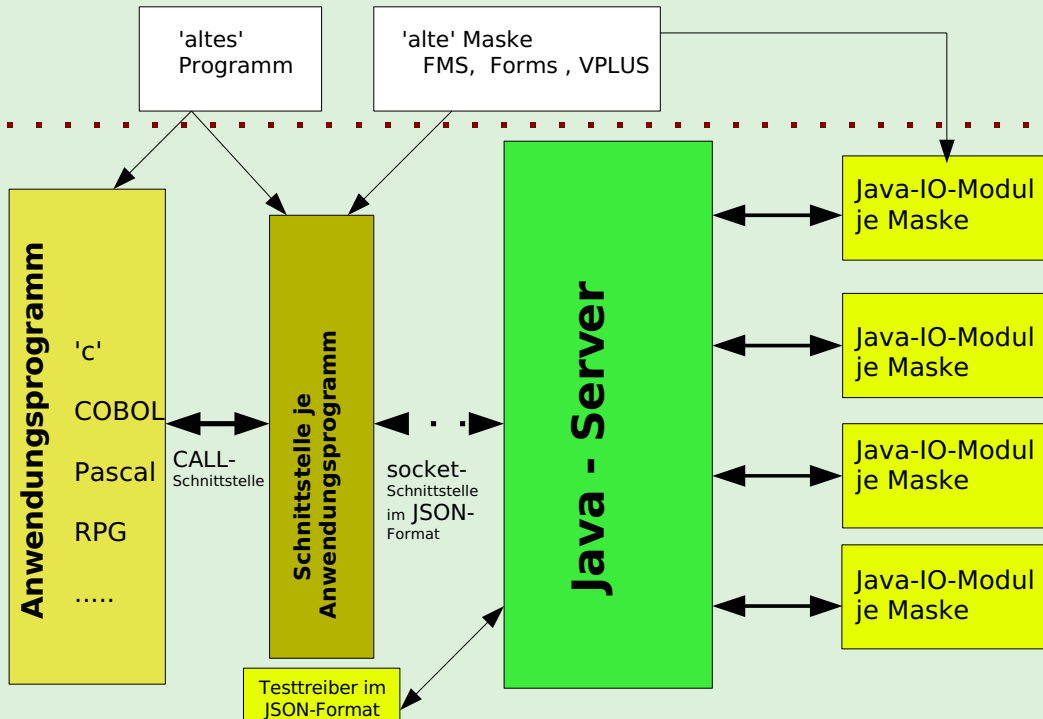
www.richter-software.de

## Das Konzept im Überblick.....



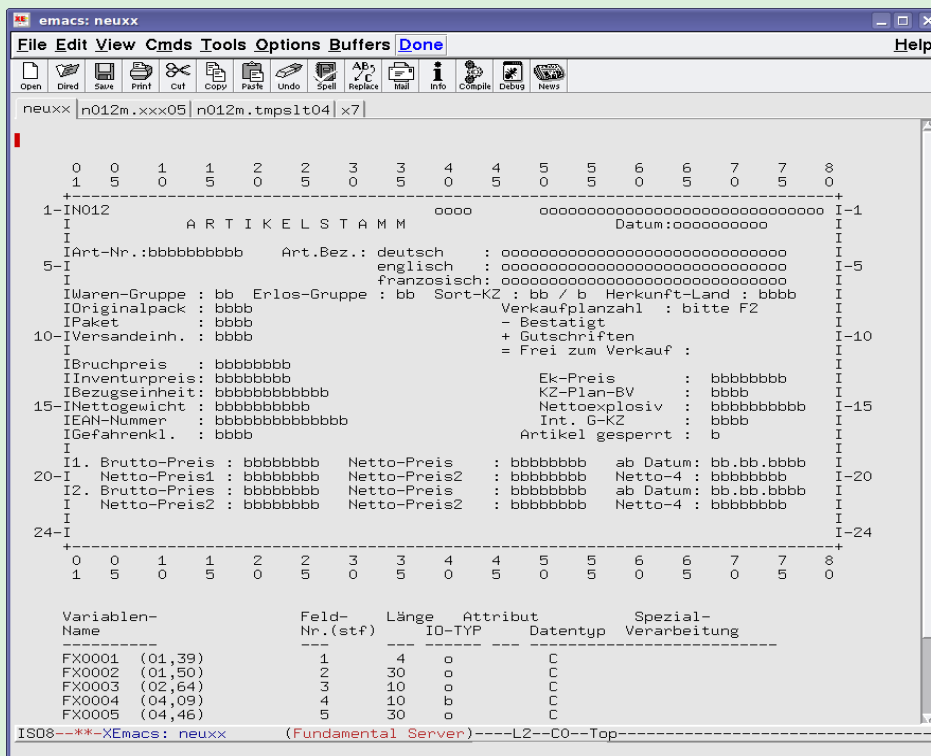
www.richter-software.de

# .....und die Lösung



www.richter-software.de

# Maske im Character-Modus



www.richter-software.de

## Maske im Java-Swing-Modus

The screenshot shows a Java Swing window titled 'n012m' with a form for 'ARTIKELSTAMM'. The form contains the following fields and values:

- Artikelnummer: N012
- ARTIKELSTAMM: aaaa
- Datum: ccccc
- Art-Nr.: ddddd
- Art.Bez.: deutsch: fffff
- englisch: ggggg
- französisch: hhhhh
- Waren-Gruppe: iiii
- Erlös-Gruppe: jjj
- Sort-KZ: kk ? ll
- Herkunft-Land: mrr
- Originalpack: nnnn
- Paket: oooo
- Versandinh.: pppp
- Verkaufplanzahl: bitte F2
- Bestätigt
- + Gutschriften
- = Frei zum Verkauf:
- Bruchpreis: 123
- Inventurpreis: 456,78
- Bezugseinheit: 12345
- Nettogewicht: a
- EAN-Nummer: c
- Gefahrenkl.: f
- Ek-Preis: 98,76
- KZ-Plan-BV: 987?
- Nettoexplosiv: b
- Int. G-KZ: d
- Artikel gesperrt: g
- 1. Brutto-Preis: 123
- Netto-Preis: 456
- ab Datum: 18.04.2007
- Netto-Preis1: 789012
- Netto-Preis2: 987
- Netto-4: 12345
- 2. Brutto-Preis: 9843
- Netto-Preis: 6523
- ab Datum: 19.04.2007
- Netto-Preis2: 98764
- Netto-Preis2: 98765
- Netto-4: 54321

At the bottom of the window, there is a grid of 12 'PF...' buttons and a green 'Enter' button.

www.richter-software.de

## Java-Swing-Modul komplett generiert

- Im 1. Schritt, nach der maschinellen Generierung, ist das Layout des Originals 1:1 übernommen.
- Es werden die kompletten Java-Sourcen geliefert – Eine Erweiterung im grafischen oder funktionalen Bereich ist somit problemlos möglich.
  - Listboxen
  - verschiedenen Fonts
  - 'Tabbed Panes'
  - etc.

www.richter-software.de



## Der Original FMS-Source (1/2)

```

!           FMS Form Description Application Aid
!           Version V2.5

FORM NAME='N012M'
  AREA_TO_CLEAR=1:23
  WIDTH=CURRENT
  BACKGROUND=CURRENT
  HIGHLIGHT=REVERSE
  ;

TEXT (1,1) 'N012'
  REVERSE
  ;
TEXT (2,13) 'A R T I K E L S T A M M           Datum:'
  ;

TEXT (4,1) 'Art-Nr.:'
  ;
TEXT (4,23) 'Art.Bez.: deutsch      :'

:..... usw .....

```

www.richter-software.de

## Der Original FMS-Source (2/2)

```

:..... usw .....
ATTRIBUTE_DEFAULTS FIELD
  CLEAR_CHARACTER=' '
  NOAUTOTAB BLANK_FILL NOBLINKING NOBOLD NOREVERSE
  NOUNDERLINE NODISPLAY_ONLY ECHO NOFIXED_DECIMAL
  LEFT_JUSTIFIED NOSUPERVISOR_ONLY NOSUPPRESS NOUPPERCASE
  ;

FIELD NAME='F$0001' (1,39)      (das '$' wurde generell in 'X' getauscht)
  PICTURE=4'C'
  DISPLAY_ONLY
  ;
FIELD NAME='F$0002' (1,50)
  PICTURE=30'C'
  DISPLAY_ONLY
  ;
FIELD NAME='F$0003' (2,64)
  PICTURE=10'C'
  DISPLAY_ONLY
  ;
:..... usw .....

```

www.richter-software.de

## Die Verbindung zum Java-Server aus dem Python-Anwendungsprogramm (1/3)

```

##-----+
##  Vorbereiten und Senden einer Map... |
##-----+
dSystem={} # alle Server-Steuer-Informationen
dMap={} # alle Map-relevanten Informationen
lVariablen=[] # alle Variablen-relevanten Informationen
# Informationen - als Liste von Hashes

dSystem['SysBenutzerName']='Enno Richter'
dSystem['SysFunktionsCode']=130

#
dMap['mapName']='n012m' # der Name der zu testenden Map
dMap['mapHeaderText']='n012m'

#
dx={} # HASH ist nun 'leer'
dx['varName']='FX0001' # einsetzen des Variablen-Namens...
dx['varInhalt']='aaaaa' # ...und des Inhalts.
lVariablen.append(dx) # ...und nun in der Liste speichern.
dx={} # HASH ist nun 'leer'
dx['varName']='FX0002' # einsetzen des Variablen-Namens...
dx['varInhalt']='bbbbbb' # ...und des Inhalts.
lVariablen.append(dx) # ...und nun in der Liste speichern.
print '\nJSON_Objekt...:' , json_Objekt

```

www.richter-software.de

## Die Verbindung zum Java-Server aus dem Python-Anwendungsprogramm (2/3)

```

#-----+
#  ...bilden des JSON-Objektes... |
#-----+
lJSON=[] # das JSON-Objekt besteht generell
lJSON.append(dSystem) # aus 3 Teilen die als Liste gespeichert
lJSON.append(dMap) # werden.
# 1=SystemInformationen
# 2=MapInformationen
lJSON.append(lVariablen) # 3=VariablenInformationen
##print '\nlJSON.....:' , lJSON
json_Objekt=json.write(lJSON) # das json-Objekt wird gebildet.
print '\nJSON_Objekt...:' , json_Objekt
##-----+
##  ...den Server adressieren... |
##-----+
serverHost='localhost' #
serverPort=50007 #
message =json_Objekt #
sockobj=socket(AF_INET, SOCK_STREAM) #
sockobj.connect((serverHost, serverPort)) #

```

www.richter-software.de

## Die Verbindung zum Java-Server aus dem Python-Anwendungsprogramm (3/3)

```
##-----+
## ...nun das JSON-Objekt per SOCKET an den Java-Server schicken... |
##-----+
sockobj.send(message + '\n')      #-->-->-->-->-->-->-->-->-->-->-->-->-->-->-->-->
                                   #-->-->-->-->-->-->-->-->-->-->-->-->-->-->-->-->
                                   #-->-->-->-->-->-->-->-->-->-->-->-->-->-->-->-->
##-----+
## ...und die Antwort erwarten...
##-----+
                                   #<--<--<--<--<--<--<--<--<--<--<--<--<--<--<--<--<
                                   #<--<--<--<--<--<--<--<--<--<--<--<--<--<--<--<--<
data=sockobj.recv(1024)          #<--<--<--<--<--<--<--<--<--<--<--<--<--<--<--<--<
                                   #<--<--<--<--<--<--<--<--<--<--<--<--<--<--<--<--<
```

*Die eingegebenen Daten des Benutzers stehen nun im gleichen Format zur Verfügung*

www.richter-software.de

## Die Alternativen

### GUI mit Hilfe diverser Programmiersprachen

Visual Basic (MONO-Projekt)

Python (o.a.) incl. QT4

Java

Swing

RCP (Eclipse)

diverse.....

- **Grafische Darstellung durch Browser**  
(Web-Oberfläche – Web-Server (Tomcat etc.), AJAX)
  - PHP
  - Java Server Faces / Java Server Pages
  - Java Script  
mit Unterstützung verschiedenster Frameworks  
(> 50 Alternativen)

## Web-Oberfläche mit COBOL

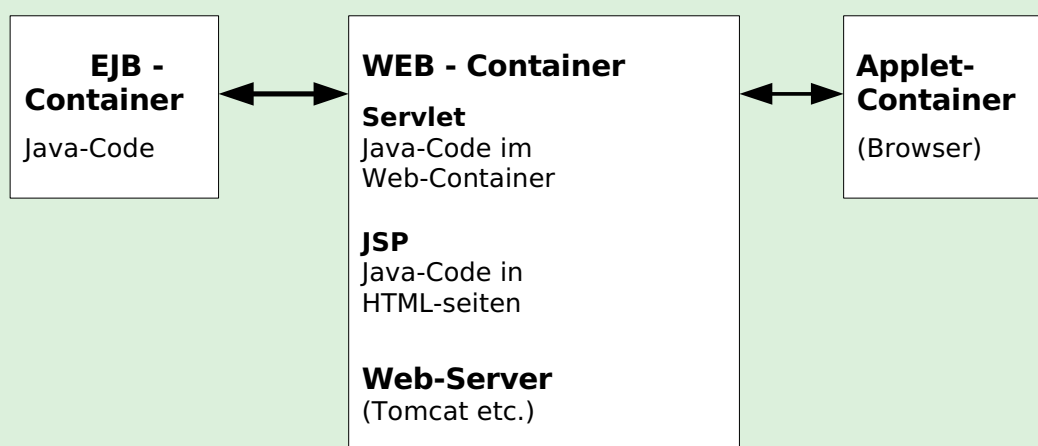
- Unter AcuCOBOL ist es möglich im HTML-Format Browser-Informationen anzuzeigen und einzulesen.
- andere COBOL-Derivate bieten die gleichen Erweiterungen ! ?
- Diese Lösungen sind nicht standardisiert - also Herstellerabhängig !

www.richter-software.de

## Web-Oberfläche mit JEE5

(Java Enterprise Edition 5)

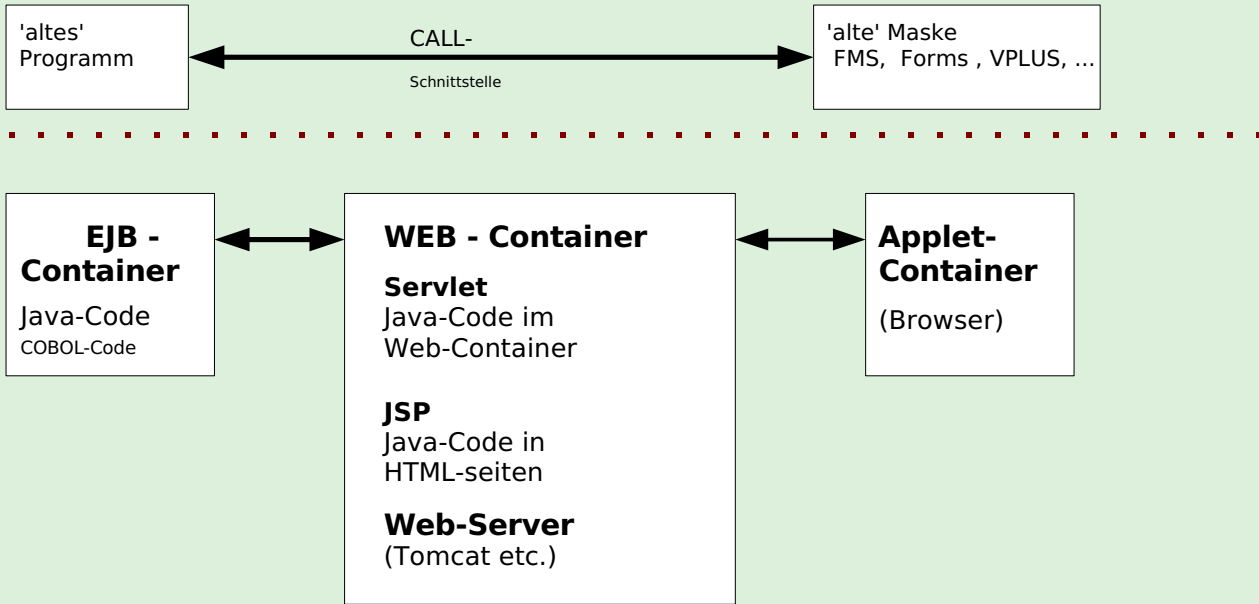
- Herstellerunabhängiger Standard
- Plattformunabhängig



www.richter-software.de

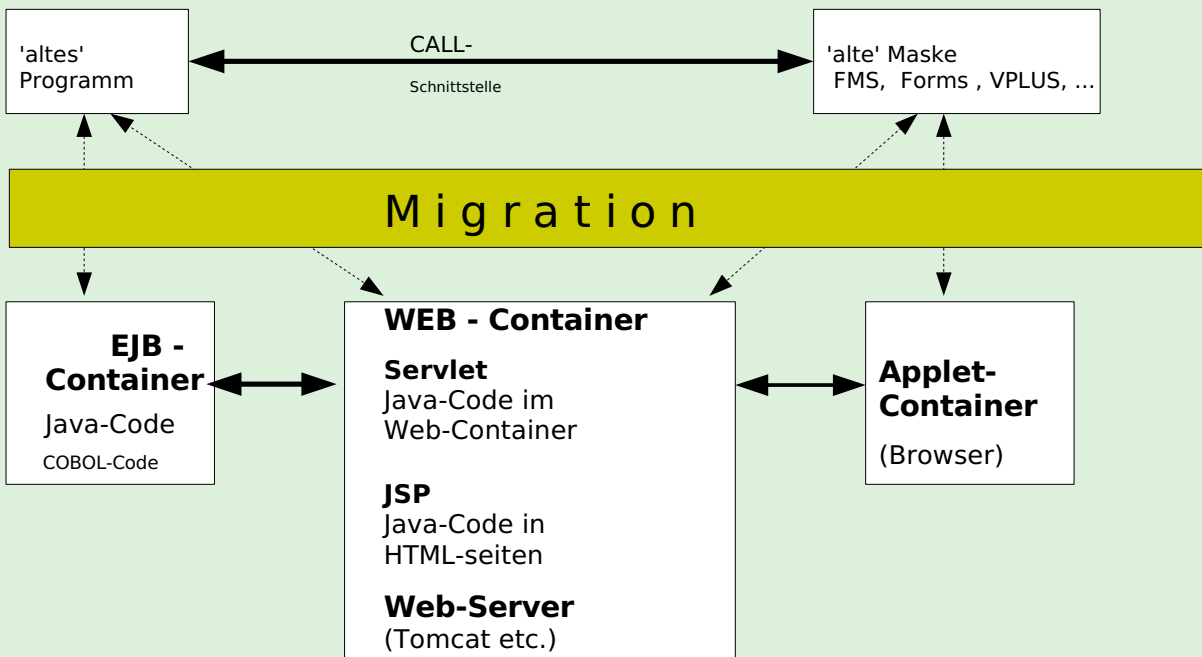
# Web-Oberfläche mit JEE5

● Eine (weitgehend) maschinelle Migration ist möglich !



www.richter-software.de

## ...die technische Realisierung



www.richter-software.de

**Fragen /  
Anregungen**

**???**

[www.richter-software.de](http://www.richter-software.de)

**Vielen Dank  
für Ihre  
Aufmerksamkeit !**

[www.richter-software.de](http://www.richter-software.de)

## Infos im Web - Literaturquellen

Introduction to VAX-11 FMS

[www.very-clever.com/json.php](http://www.very-clever.com/json.php)

[www.acucobol.com](http://www.acucobol.com)

[java.sun.com/javaee/5/index.jsp](http://java.sun.com/javaee/5/index.jsp)

[java.sun.com/javaee/glassfish](http://java.sun.com/javaee/glassfish)

[www.netbeans.info](http://www.netbeans.info)

[www.eclipse.org](http://www.eclipse.org)

DEC-FMS - Dokumentation

JSON-Dokumentation

COBOL-compiler

Infos rund um das Java EE 5 SDK

Ablaufumgebung für JEE 5

Freie Entwicklungsumgebung - Sun

Freie Entwicklungsumgebung

[www.richter-software.de](http://www.richter-software.de)

## ***Enno Richter***

Software - Entwicklung

Talstraße 38  
71139 Ehningen

Tel.           +49 7034 / 993020  
Mobil         +49 172 / 7303136  
email         migration@richter-software.de  
URL           www.richter-software.de

[www.richter-software.de](http://www.richter-software.de)

---

## Ende des Foliensatzes

Erstellt mit *"OpenOffice 2.0.4 für Linux"*

Gespeichert als `/raid/staroffice6.0/RTC-Marketing/DECUS_2H02_Vortrag.odp`

Letzte Änderung `4/17/07`

---