9. CICS Communication

9.1 Das 3270 Protokoll

9.1.1 Business- und Präsentationslogik

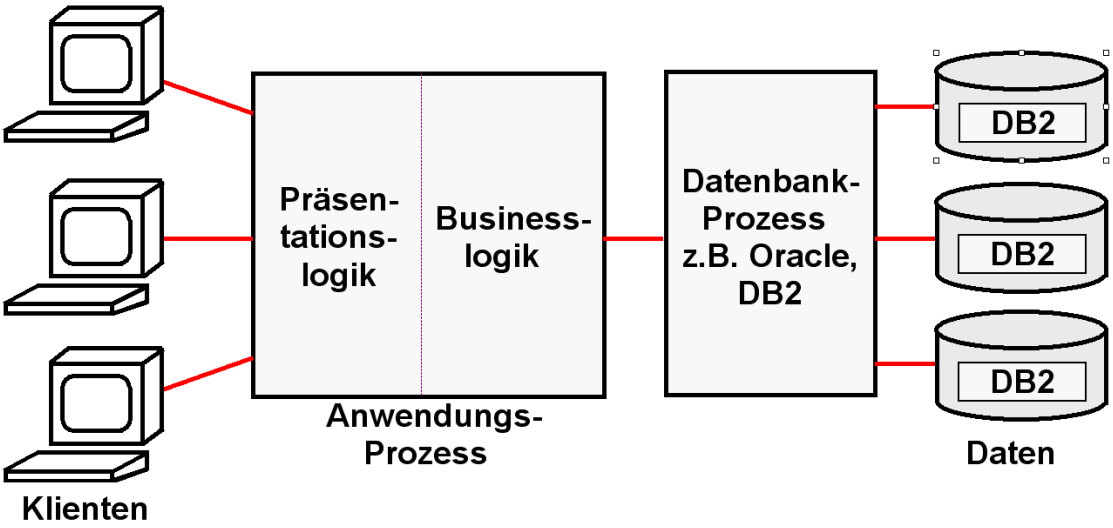

Abb. 9.1.1
Aufteilung des Anwendungsprogramms

Ein sauber strukturiertes CICS Programm besteht aus zwei Teilen: Business Logik und Präsentations-Logik.

Business Logik ist der Teil, in dem Berechnungen erfolgen und Daten in einer Datenbank gelesen/geschrieben werden.

Präsentations- Logik ist der Teil, in dem die Ergebnisse der Berechnungen so aufgearbeitet werden, dass sie dem Benutzer in einer ansprechenden Art auf dem Bildschirm dargestellt werden können.

Business Logik wird in Sprachen wie C, C++, COBOL, PL/1, Java usw. geschrieben.

Für die Präsentations-Logik gibt es viele Möglichkeiten. Die modernste Alternative benutzt Java Servlets und Java Server Pages und einen Web Application Server um den Bildschirminhalt innerhalb eines Web Browsers darzustellen.

Die älteste (und einfachste) Alternative verwendet das CICS BMS (Basic Mapping Support) Subsystem. BMS Programme werden in der BMS Sprache geschrieben.

9.1.2 Endgeräte für die Transaktionsverarbeitung

Es existieren viele unterschiedliche Arten von Endgeräten (Klienten) für die Transaktionsverarbeitung:

- **Arbeitsplatzrechner**

  Browser GUI (Java Swing Classes)
  Windows GUI
  Motiv, KDE, Gnome
  SAPGUI
  3270 CUI

GUI	**Graphical User Interface**
CUI	**Character User Interface**

- **Hand Held Geräte, z.B. Tablet, Mobiltelephon**
- **Geldausgabeautomaten, Kontoauszugsdrucker**
- **Supermarkt Registrierkasse, Tankstellen Zapfsäule**
- **Produktionssteuerungselektronik**

Endgeräte (Klienten) für das Arbeiten mit CICS sind keinesfalls nur Arbeitsplatzrechner mit Tastatur und Bildschirm. Neue Geräte erscheinen ständig. Ein Beispiel ist der Fahrkartenautomat der deutschen Bundesbahn

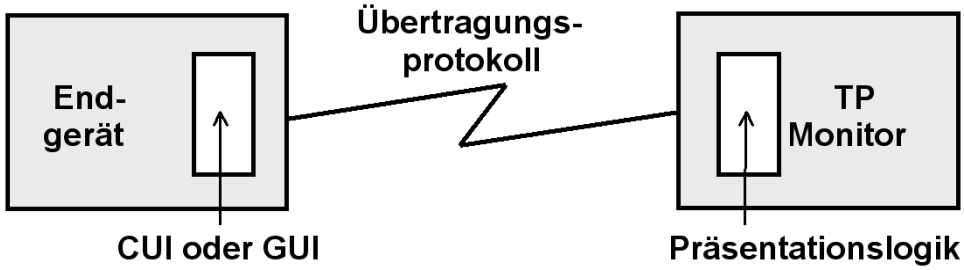

Abb. 9.1.2
Für die Verbindung mit CICS benötigt der Klient eine spezielle Komponente

Die GUI oder CUI ist ein Prozess in den Endgeräten, welcher für die visuelle Ein/Ausgabe zuständig ist. Es ist die Aufgabe der Präsentationslogik, Information von/zu den Endgeräte GUIs oder CUIs in geeigneter Form aufzubereiten.

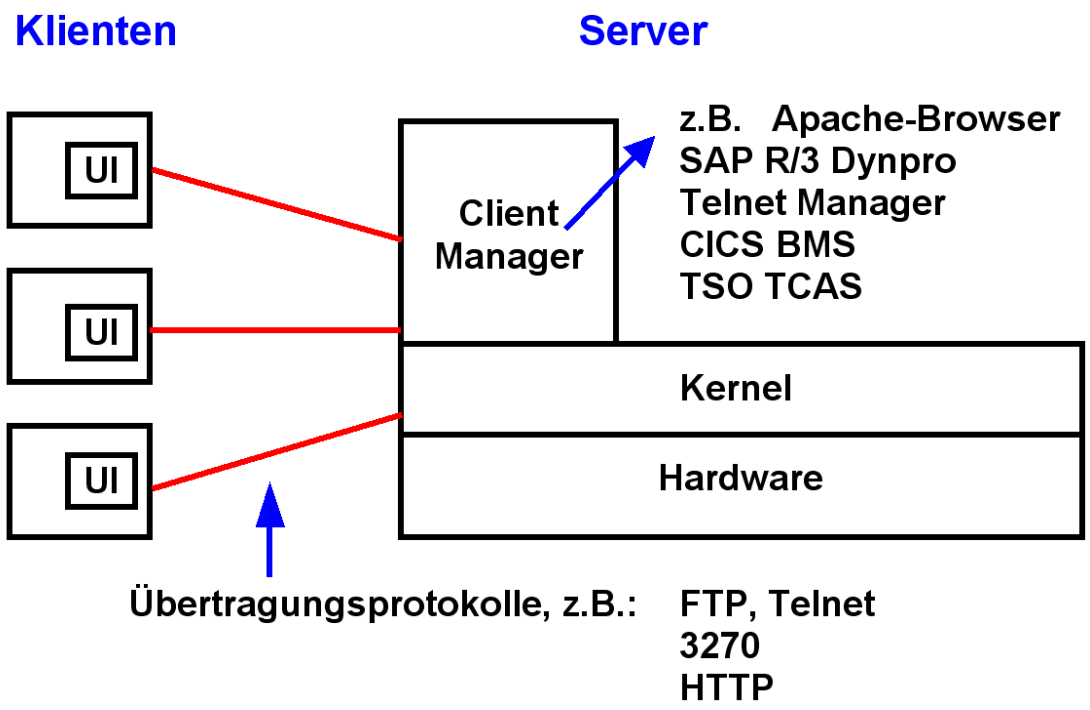

Abb. 9.1.3
Client-Server User Interfaces

Auf der Klientenseite wird eine User Interface (UI) benötigt, z.B. ein 3270 Emulator oder ein Browser.

Auf der Serverseite wird ein Client Manager benötigt, z.B. CICS Terminal Manager oder Apache Web Server.

Für jedes Paar (UI) – Client Manager wird ein entsprechendes Übertragungsprotokoll in Schicht 5 des OSI Modells benötigt. Schicht 3 – 4 benutzen heute meistens TCP/IP.

9.1.3 Alternativen der CICS Bildschirmausgabe

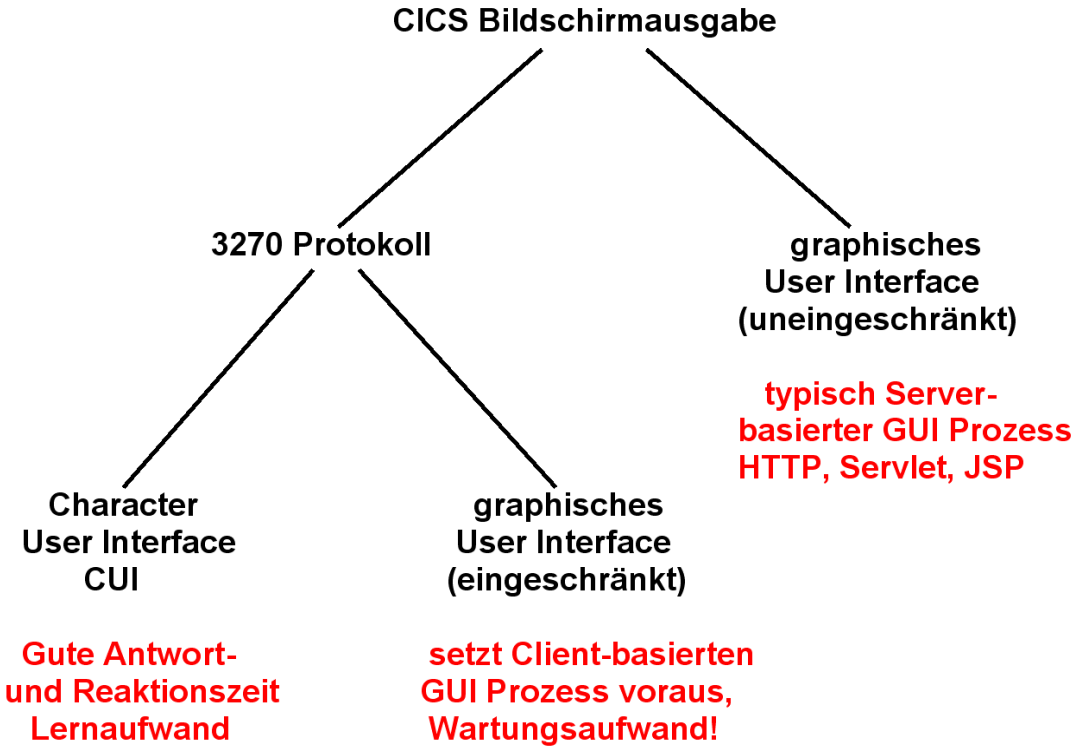

Abb. 9.1.4
existieren zwei unterschiedliche Möglichkeiten für eine grafische Benutzerschnittstelle

Die traditionelle CICS Bildschirmausgabe verwendet den Basic Mapping Support (BMS), sowie das weit verbreitete 3270 Übertragungsprotokoll, welches auch von vielen nicht-IBM Software Produkten verwendet wird, z.B. vom System SAP R/3.

BMS und 3270 können in einer grafischen und der ursprünglichen Zeichen-orientierten Version eingesetzt werden. Die grafische Version ist aber im Funktionsumfang eingeschränkt.

Deshalb existiert eine vom 3270 Protokoll unabhängige grafische Version ohne Einschränkungen.

Für CICS existieren viele Alternativen, mit grafischen Oberflächen unabhängig vom 3270 Protokoll zu arbeiten. Zwei weit verbreitete Alternativen sind das CICS Transaction Gateway (CTG) und die MQSeries CICS Bridge. Auch Web Service Schnittstellen gewinnen an Bedeutung. Einzelheiten hierzu im Abschnitt 18.3, Java Connection Architecture.

Interessanterweise ist die Verwendung der 3270/CUI nach wie vor weit verbreitet. Neu entwickelte Anwendungen stellen häufig neben einer GUI auch eine 3270/CUI zur Verfügung. Viele Benutzer schätzen sie, weil hiermit eine bessere Produktivität möglich ist.

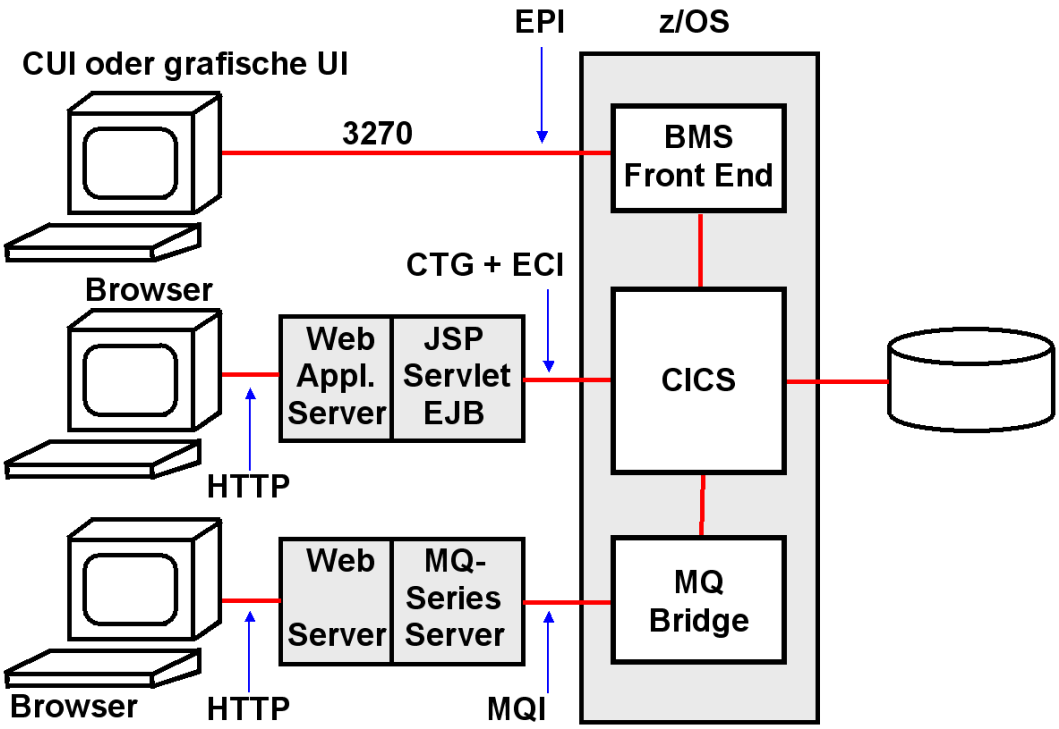

Abb. 9.1.5
Alternativen der CICS Klienten Anbindung

Die hier gezeigten Alternativen benutzen unterschiedliche Schnittstellen:

EPI **Die BMS Maps werden weiter verwendet. Keine Änderung der Information, die auf dem Bildschirm wiedergegeben wird. Die Darstellung der Information kann geändert werden.**

ECI **Die Presentation Service Komponente von CICS (BMS)wird nicht genutzt. Direkter Zugriff auf COMMAREA. Häufig im Zusammenhang mit dem CICS Transaktion Gateway (CTG) benutzt (siehe Kapitel 18).**

MQSeries **Asynchrone Übertragung durch Message oriented Middleware (siehe Kapitel 10).**

Es existieren viele weitere Möglichkeiten, z.B. CICS-Corba Bridge (EJBs) oder SOAP.

9.1.4 3270 Bildschirm

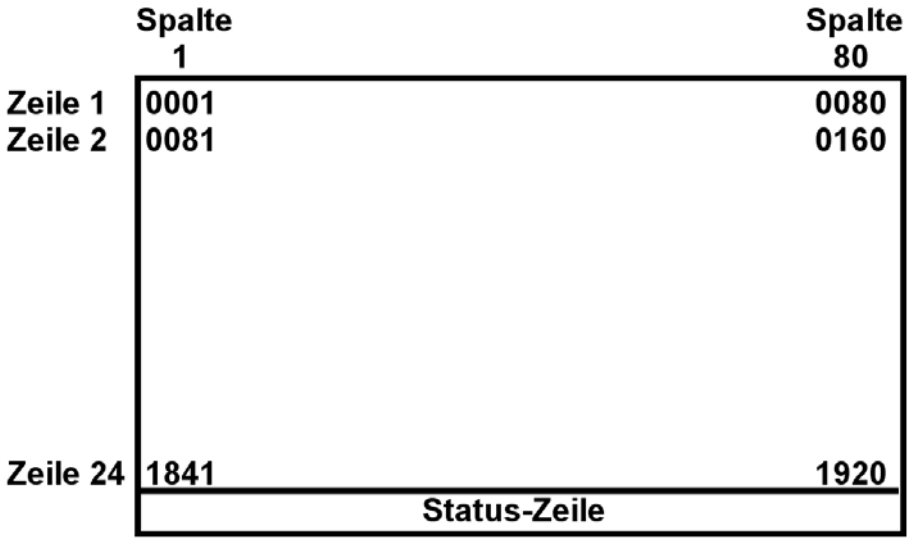

Abb. 9.1.6
Positionen im Bildschirm Puffer (Presentation Space)

Das 3270 Protokoll unterstellt im Terminal einen 1920 Byte großen Bildschirm Puffer, der aus 24 Zeilen mit je 80 Byte besteht. Jedes Byte kann einzeln adressiert werden. Der Bildschirmpuffer wird auch als „Presentation Space“ bezeichnet.

Der 3270 Datenstrom lädt die Information in den Presentation Space. Die interne Logik platziert den Inhalt des Puffers an die entsprechende Stelle des Bildschirms, der hierfür in 80 Spalten und 24 + 1 Zeilen aufgeteilt ist.

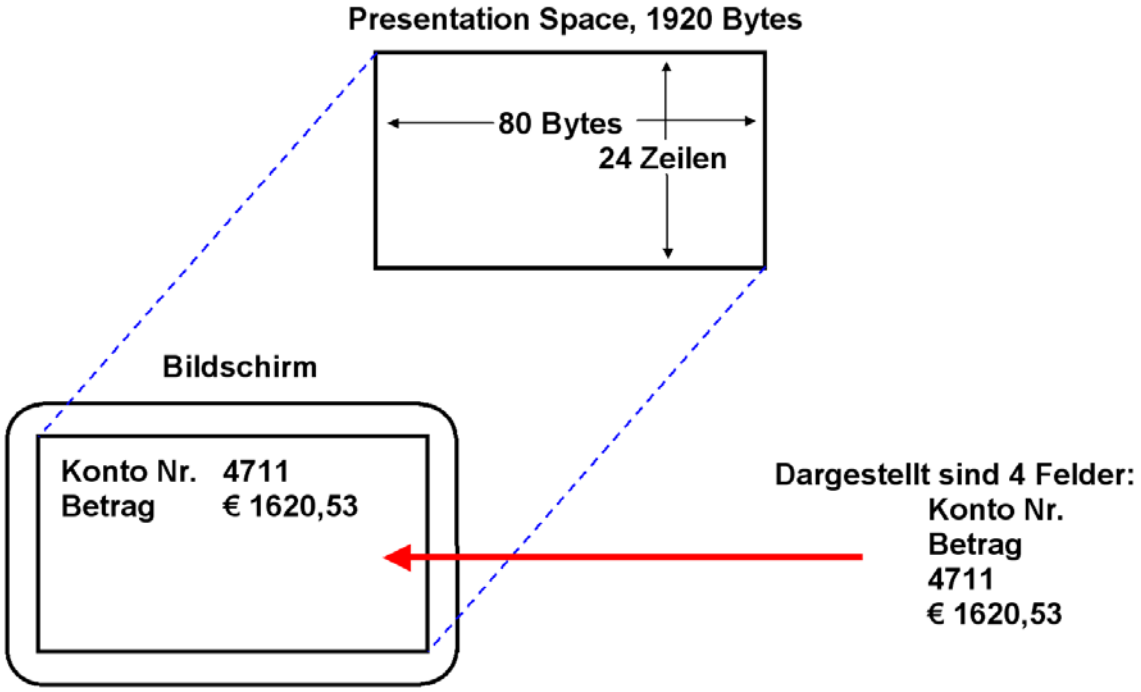

Abb. 9.1.7
3270 Bildschirmdarstellung

Es wird eine Nachricht übertragen, die einen 24 x 80 = 1920 Byte großen Presentation Space mit Character Daten füllt und vom CICS Terminal interpretiert wird.

Der Pufferinhalt wird auf dem Bildschirm als 24 Zeilen mit 80 Zeichen/Zeile wiedergegeben. Jede der 1920 Byte Positionen kann einzeln adressiert werden. In der Regel werden Gruppen von Bytes (Felder) adressiert. Die Felder erscheinen auf dem Bildschirm an der Stelle, an der sie in dem Presentation Space gespeichert sind.

```
ACCOUNTS MENU

   TO SEARCH BY NAME, ENTER SURNAME AND IF REQUIRED, FIRST NAME

      SURNAME     :                          (1 TO 18 ALPHABETIC CHRS)
      FIRST NAME  :                          (1 TO 12 ALPHABETIC CHRS OPTIONAL)

   TO PROCESS AN ACCOUNT, ENTER REQUEST TYPE AND ACCOUNT NUMBER

      REQUEST TYPE:            (D-DISPLAY, A-ADD, M-MODIFY, X-DELETE, P-PRINT)
      ACCOUNT     :            (10000 TO 79999)
      PRINTER ID  :            (1 TO 4 CHARACTERS (REQUIRED FOR PRINT REQUEST))

ACCT    SURNAME       FIRST    MI  TTL   ADDRESS                   ST      LIMIT
26001   Meier         Rolf      A        Ritterstr. 13             N     1000.00
26002   Meier         Stefan    A        Wilhelmstr. 24            N     1000.00
26003   Meier         Tobias    A        Nikolaistr. 23            N     1000.00




ENTER DATA AND PRESS ENTER FOR SEARCH OR ACCOUNT REQUEST OR PRESS CLEAR TO EXIT
```

Abb. 9.1.8
Beispiel eines CICS Basic Mapping Support α/n Bildschirms

Der Basic Mapping Support ist eine CICS Präsentationslogik Komponente, die eine zeichenorientierte (alpha/numerisch, α/n) Bildschirm-Ein/Ausgabe ermöglicht.

9.1.5 3270 Protokoll

Das 3270 Protokoll wurde ursprünglich für nicht-intelligente Terminals eingesetzt.

Es arbeitet mit einem zeichenorientierten Bildschirm, bestehend aus 24 Zeilen mit je 80 α/n Zeichen sowie einem Bildschirmpuffer mit identischer Organisation. Der ursprüngliche 3270 Terminal verfügte hierfür über einen 1920 Byte großen Speicher. Eine Hardware Logik aus diskreten Bausteinen stellte jedes Byte in dem Bildschirmpuffer (Presentation Space) automatisch in die entsprechende Position auf den Bildschirm.

Jede der 24 x 80 = 1920 Positionen ist vom Anwendungsprogramm im CICS Server individuell adressierbar.

Normalerweise werden „Felder“ angesprochen. Ein Feld ist eine Folge von Zeichenpositionen. Felder können gelesen und geschrieben werden

Eine CICS-Utility, Basic Mapping Support (BMS) erleichtert dem Anwendungsprogrammierer die Entwicklung von „Screens“ (Bildschirminhalten).

Auf heutigen Arbeitsplatzrechnern wird die 3270 Bildschirmdarstellung mit Hilfe eines Programms implementiert, welches als „3270 Emulator“ bezeichnet wird. Gelegentlich wird auch die Bezeichnung „3270 Klient“ verwendet. Der 3270 Emulator hat eine Funktion vergleichbar mit dem Telnet Klienten für das Telnet Protokoll auf Unix/Linux Rechnern, z.B. „putty“.

Das Anwendungsprogramm erzeugt die Datenausgabe an den Bildschirm in der Form einer seriell übertragenen Datenstroms mit dem folgenden Format:

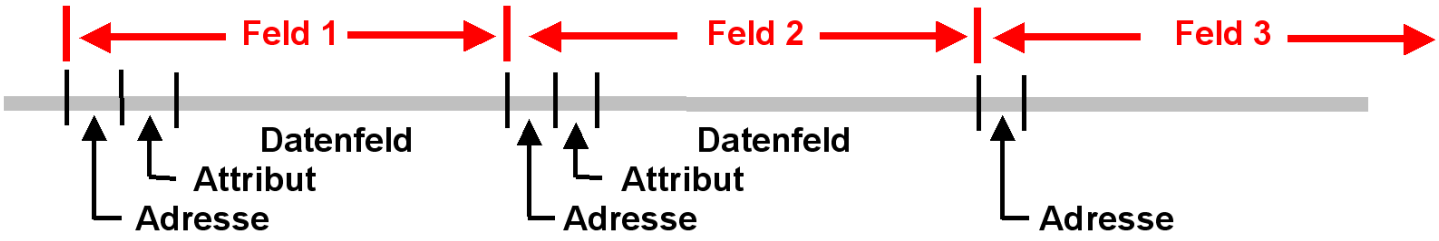

Abb. 9.1.9
3270 Datenstrom

Der 3270 Bildschirm besteht aus 24 α/n Zeilen mit je 80 fixed Font-Width Zeichenpositionen pro Zeile. Das Adressenfeld kennzeichnet eine der 24 x 80 = 1920 Zeichenpositionen auf dem Bildschirm (Zeile und Spalte).

Das Datenfeld enthält eine variable Anzahl von α/n Zeichen, welche auf dem Bildschirm in einem Feld wiedergegeben werden.

Das Attributfeld (3 Bytes bei BMS/CICS) enthält Steuerzeichen, welche Informationen über die Art der Wiedergabe des folgenden Datenfeldes enthalten, z.B. Darstellung in roter Farbe, blinkender Cursor, Font, andere...

Das 3270 Protokoll verwendet eine Untermenge der 256 Zeichen des ASCII oder EBCDIC Zeichensatzes zur Datenwiedergabe auf dem Bildschirm. Die restlichen Zeichen werden als Steuerzeichen für Steuerungszwecke eingesetzt.

Es existieren Sonderversionen für kyrillische, griechische, hebräische usw. Zeichensätze und für die Schreibweise von rechts nach links (z.B. hebräisch).

9.1.6 Entwicklung der Präsentationslogik

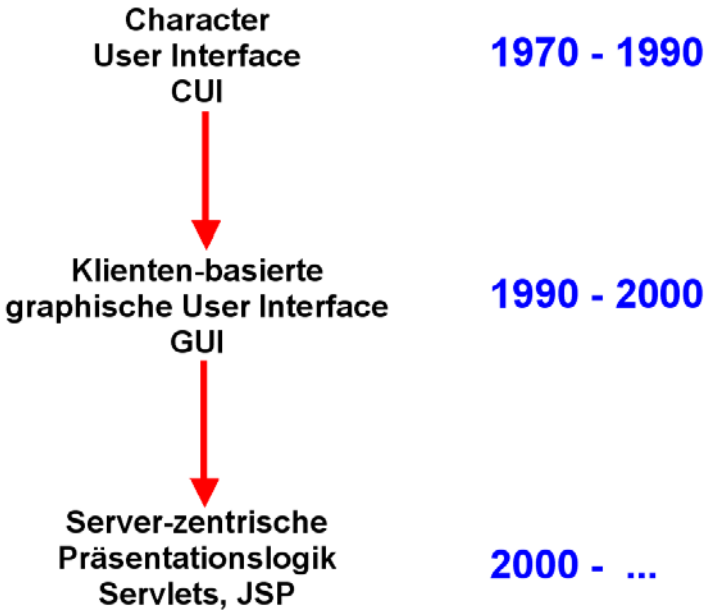

Abb. 9.1.10
Unterschiedliche Entwicklungsstufen

CICS Terminals verwendeten ursprünglich die 3270/CUI Oberfläche und Darstellung. Später kam eine 3270 Protokoll basierte GUI dazu, die ein entsprechendes Umsetzungsprogramm in jedem Terminal erforderte.

Vom 3270 Protokoll unabhängige grafische Versionen entstanden gleichzeitig mit der Entwicklung von Java. Obwohl nicht grundsätzlich erforderlich, wird hierbei die Präsentationslogik in der überwiegenden Mehrzahl der Fälle in Java Servlets und Java Server Pages geschrieben.

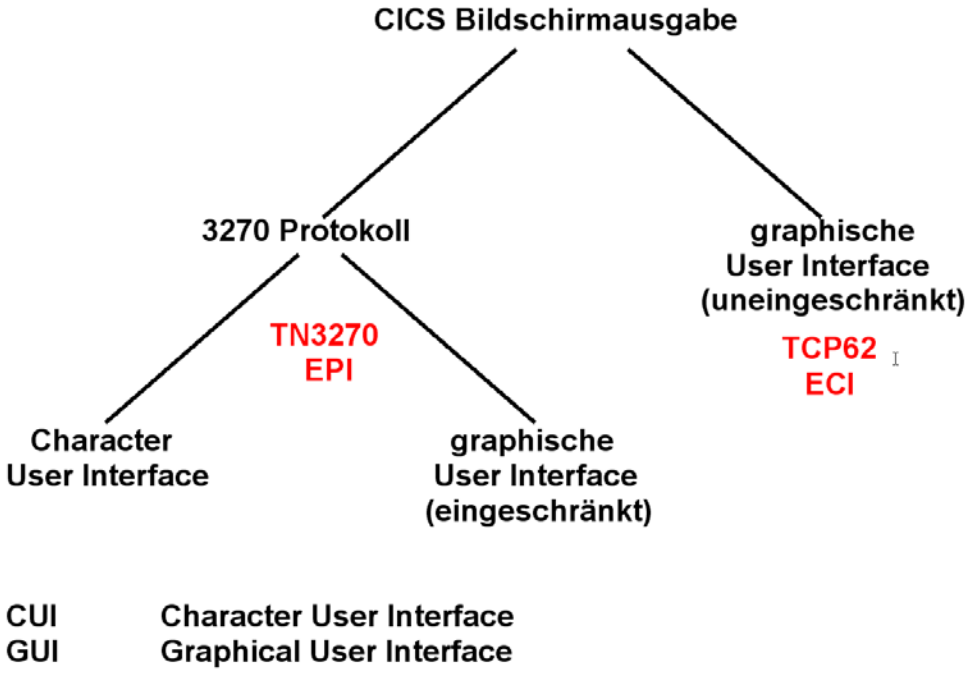

Abb. 9.1.11
Alternativen der Bildschirmausgabe

CICS arbeitete ursprünglich ausschließlich mit dem SNA Protokoll: das 3270 Protokoll ist Bestandteil von SNA. Später wurde eine TCP/IP Version geschaffen, die als TN3270 bezeichnet wird.

Anwendungen auf dem Klienten können mit Hilfe der „External Programming Interface“ (EPI) über TN3270 mit CICS kommunizieren. Dies gilt auch für die eingeschränkte graphische User Interface über 3270.

Die uneingeschränkte graphische User Interface verwendet statt TN3270 das TCP62 Protokoll und statt EPI die ECI (External Communication Interface) Schnittstelle. Die Hintergründe werden in Kapitel 18 diskutiert.

9.1.7 Basic Mapping Support (BMS)

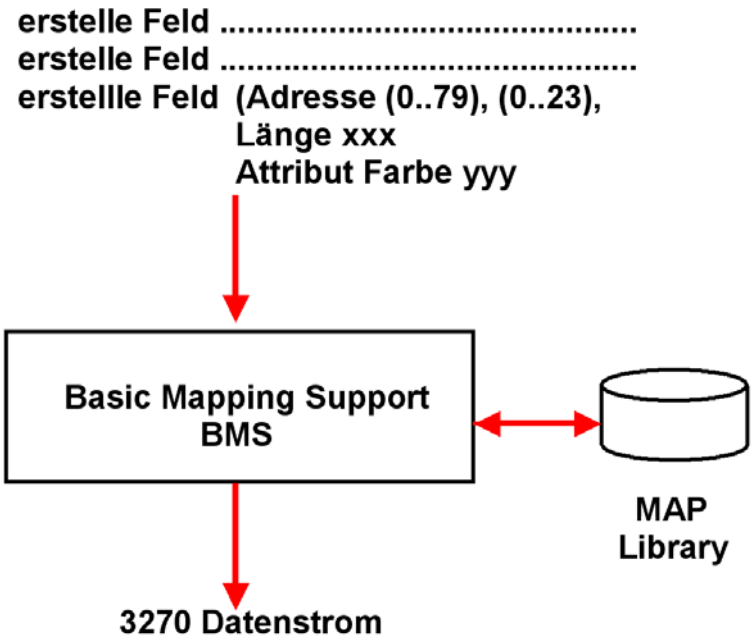

Abb. 9.1.12
BMS Programmierung

Eine Map (auch als Screen bezeichnet) definiert die Wiedergabe von Daten auf dem Bildschirm

Bei einer CUI (Character User Interface) definiert die Map:
Feld Nr. 1 am Ort mit der Adresse aaa,
Feld 2 am Ort mit der Adresse bbb, usw.

wobei aaa und bbb Adressen des 24 x 80 Presentation Spaces sind.

Diese Beschreibung erfolgt mit Hilfe einer eigenen Sprache, der BMS-Sprache.

Bei einer eingeschränkten GUI (Graphical User Interface) redefiniert der GUI Prozess innerhalb eines PC zusätzlich Aussehen und Anordnung der 3270 Datenelemente im Presentation Space Buffer.

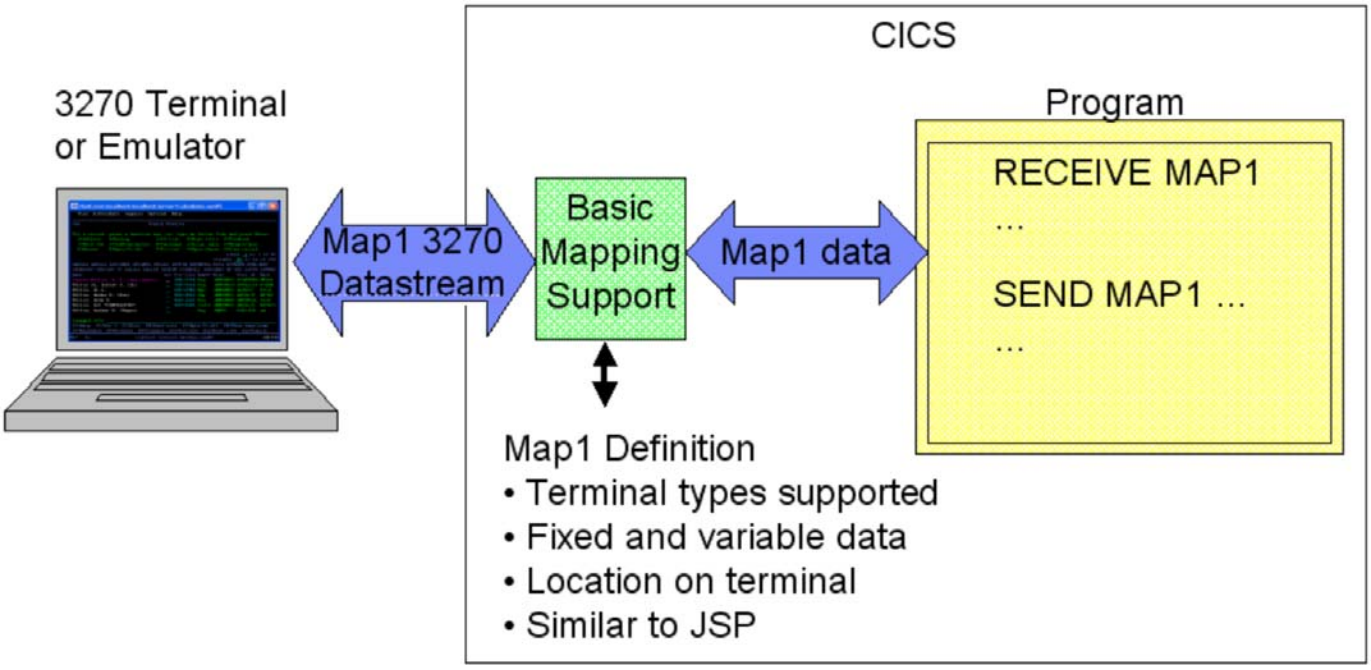

Abb. 9.1.13
BMS ist ein Teil des CICS Terminal Managers

Die zu sendende Map wird mit Hilfe einer eigenen BMS-Sprache beschrieben. Dieser Vorgang wird als Map Definition bezeichnet.

Die so definierte Map wird mit Hilfe der BMS Komponente des CICS Terminal Managers in einen seriellen Datenstrom übersetzt, der mittels des 3270 Protokolls in den Presentation Space Buffer geschrieben wird.

Die CICS Kommandos EXEC CICS SEND MAP und EXEC CICS RECEIVE MAP werden hierfür benutzt.

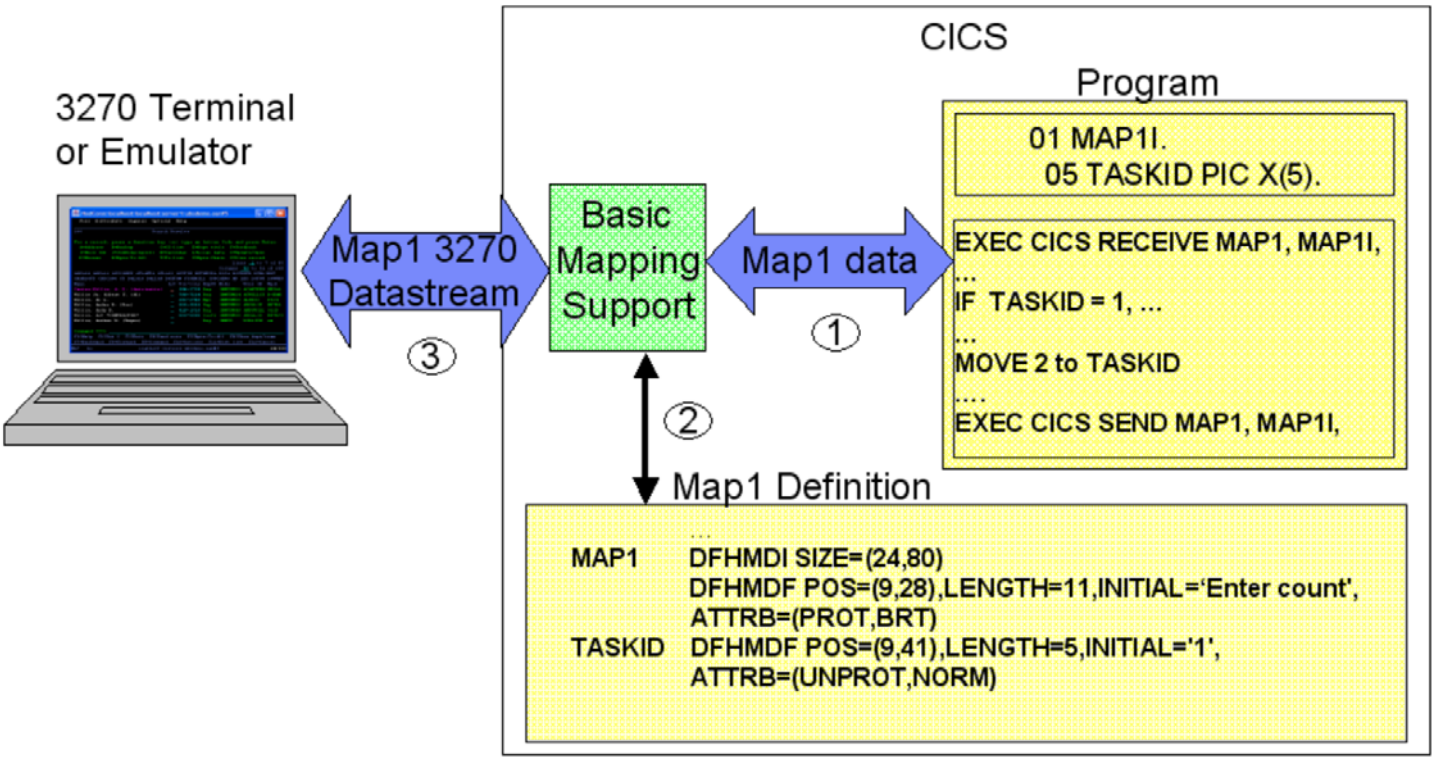

Abb. 9.1.14
BMS Beispiel

Gezeigt wird die Map Definition in der BMS Sprache. Die Ausführung eines EXEC CICS SEND MAP Kommandos bewirkt die Übersetzung der MAP Definition in einen 3270 Datenstrom und dessen Transmission an den Presentation Space Buffer des angesprochenen CICS Terminals.

Die Bildschirmwiedergabe besteht aus einer Menge von Feldern. Diese werden durch zwei Komponenten dargestellt: Map und Inhalt.

BMS (Basic Mapping Support) als Bestandteil von CICS erzeugt und verwaltet „“Maps“. Jede Map beschreibt das Aussehen eines Bildschirm-Fensters des Klienten, spezifisch die Anordnung und die Art (Attribut) von Feldern. Die Felder werden mit statischen oder dynamischen Ausgabedaten (Inhalt) gefüllt. Statische Ausgabedaten sind Teil der Map; dynamische Ausgabedaten werden durch das Anwendungsprogramm erstellt. Je nach Transaktionsart wird die dazugehörige Map in den Klienten geladen.

Jeder Klient enthält ein CUI Programm, welches die Daten auf dem Bildschirm ausgibt. Die Map bestimmt, wie und wo die Ausgabedaten in dem Fenster dargestellt werden.

Das Anwendungsprogramm kann einige der Felder mit Daten füllen. Nach Betätigung der Enter Taste werden diese in der Form einer 3270 Nachricht an den CICS Terminal übertragen.

9.1.8 Benutzung der Maps

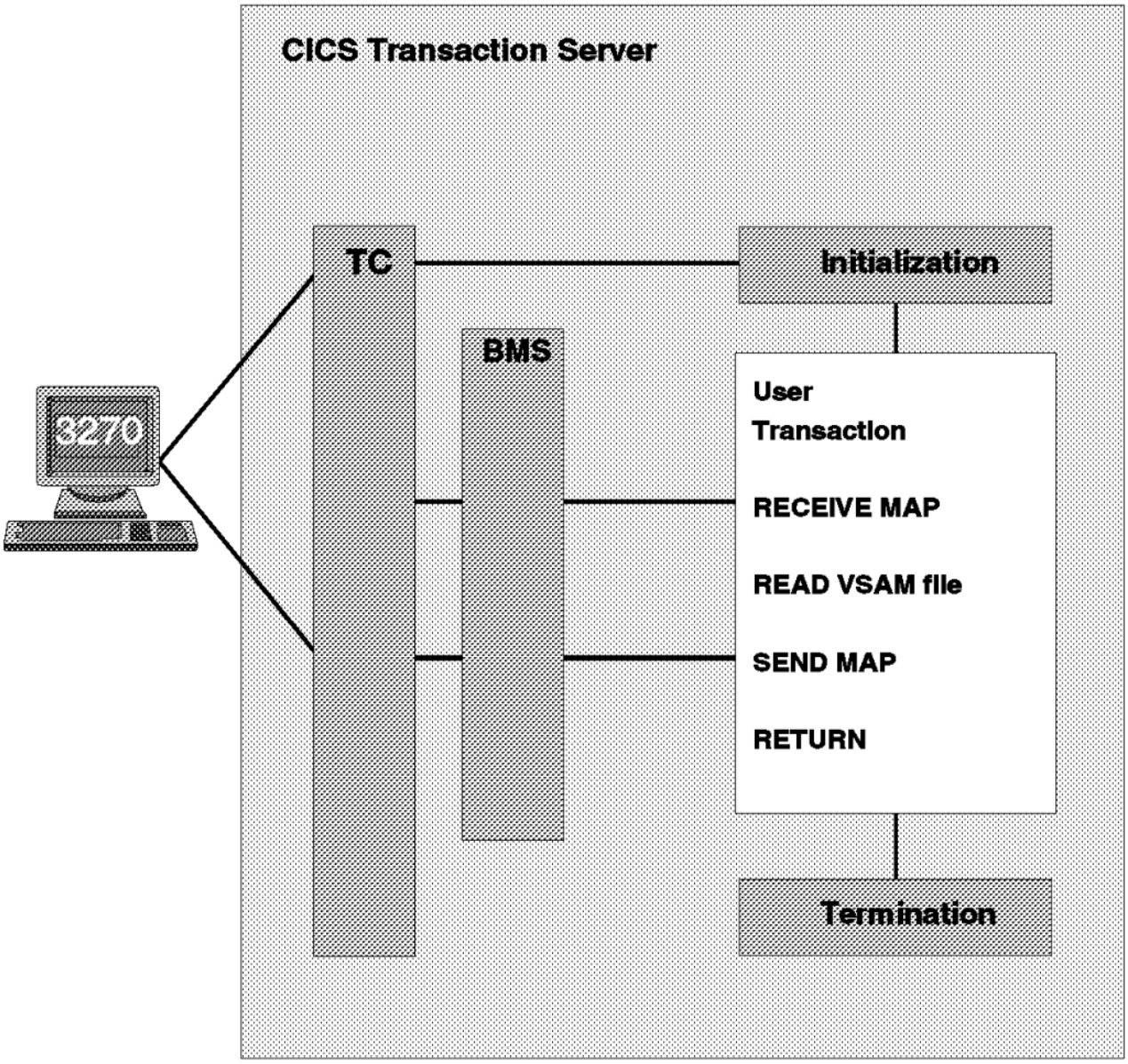

Abb. 9.1.15
Ablauf einer Transaktion

Gezeigt ist der typische Ablauf einer CICS Transaktion.

Die CICS BMS Komponente ist ein Bestandteil des CICS Terminal Managers (Terminal Control, TC).

Nach Initialisierung einer neuen Transaktion liest diese mit Hilfe eines EXEC CICS RECEIVE MAP Kommandos den Inhalt des Presentation Space Buffers.

Bei der anschließenden Verarbeitung wird z.B. ein VSAM Dataset gelesen.

Das Ergebnis der Verarbeitung wird mittels eines EXEC CICS SEND MAP Kommandos an den CICS Terminal zurückgeschickt.

9.2 3270 Bildschirmausgabe Alternativen

9.2.1 Darstellung auf dem Bildschirm

Die Datenübertragung vom /zum CICS Terminal erfolgt normalerweise über einen Input/Output Puffer mit dem Namen COMMAREA.

COMMAREA ist Teil des Scratchpad Bereiches, der vom CICS Storage Manager unterhalten wird.

```
struct adresse { char vorname  [20];
                 char nachname [20];
                 char plz      [20];
                 char ort      [20];
                 char strasse  [20];
                 char tel      [20];
               };

main()
{
...
EXEC CICS RECEIVE MAP(„adresse“) MAPSET(„wgsset“);
...
EXEC CICS SEND MAP(„adresse“) MAPSET(„wgsset“);
...
}
```

Abb. 9.2.1
Beispiel Struktur in C/C++

Die Daten, welche ein Anwendungsprogramm in den COMMAREA I/O Puffer zwecks Ausgabe an den Terminal stellt, werden normalerweise in der Form einer Struktur dargestellt, die oft auch als „Unit Record“ bezeichnet wird.

Abb. 9.2.1 zeigt ein Beispiel in der Programmiersprache C/C++ .

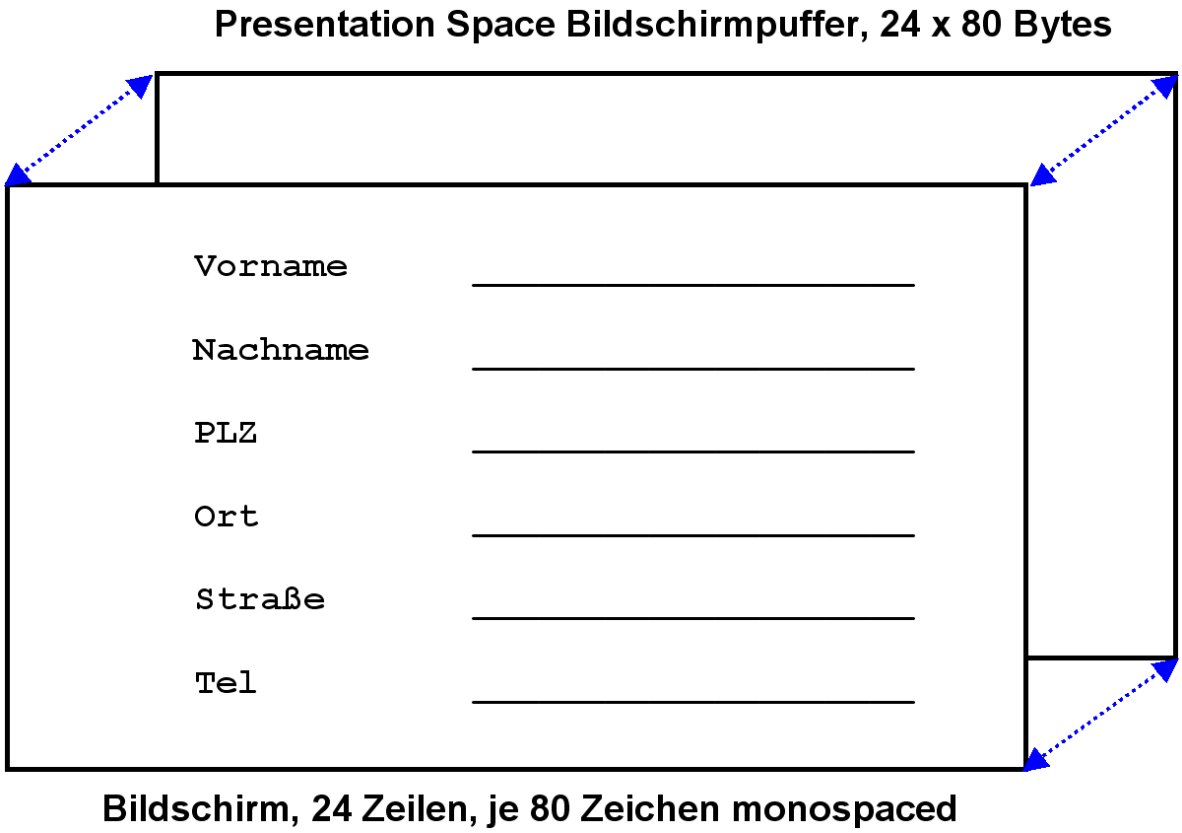

Abb. 9.2.2
Abbildung des Presentation Spaces auf den Bildschirm

Der Presentation Space Bildschirmpuffer hat eine Struktur von 24 Worten zu je 80 Byte. Jede Wortadresse und Byte Adresse innerhalb eines Wortes entspricht genau der Zeilen- und Spaltenadresse. auf der das Byte auf dem Bildschirm wiedergegeben wird.

9.2.2 Kommunikation mit dem Bildschirm

```
#include <//'PRAKT20.LIB(MSET020)'>

main()
{
EXEC CICS SEND MAP("map020") MAPSET("s04set") ERASE;
}
```

Abb. 9.2.3
Hello World in C++

Im Folgenden schauen wir uns ein einfaches CICS Hello World Programm in C/C++ an, einschließlich seiner BMS Präsentationslogik.

Dargestellt ist der CICS Hello World Programm Code. Es wird eine Map mit dem Namen `map020` gesendet, die ein Teil des Mapsets `s04set` ist.

```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
 ------------------------------------------------------------------------------
 EDIT       SPRUTH.CICS.TEST04(MAP04) - 01.02               Columns 00001 00072
 ****** ***************************** Top of Data ******************************
 ==MSG> -CAUTION- Profile changed to CAPS ON (from CAPS OFF) because the
 ==MSG>           data does not contain any lower case characters.
 ==MSG> -Warning- The UNDO command is not available until you change
 ==MSG>           your edit profile using the command RECOVERY ON.
 000001 //PREPARE  JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID
 000002 //ASSEM    EXEC DFHMAPS,MAPNAME='S04SET',RMODE=24
 000003 //SYSUT1   DD *
 000004 S04SET  DFHMSD TYPE=MAP,MODE=INOUT,LANG=C,STORAGE=AUTO,TIOAPFX=YES
 000005 *       MENU MAP.
 000006 map020  DFHMDI SIZE=(24,80),CTRL=(PRINT,FREEKB)
 000007         DFHMDF POS=(9,23),ATTRB=(ASKIP,NORM),LENGTH=34,                X
 000008                INITIAL='WELCOME TO THE MAGIC WORLD OF CICS'
 000009         DFHMDF POS=(12,27),ATTRB=(ASKIP,NORM),LENGTH=26,               X
 000010                INITIAL='MAY THE FORCE BE WITH YOU!'
 000011         DFHMSD TYPE=FINAL
 000012         END
 000013 /*
 000014 //
 Command ===>                                                 Scroll ===> PAGE
  F1=Help      F3=Exit      F5=Rfind     F6=Rchange  F12=Cancel
```

Abb. 9.2.4
MAP für das Hello World Programm

…und hier ist der ISPF Editor Screen mit der Beschreibung der MAP in der BMS Sprache. Der Mapset unseres Hello World Programms besteht aus einer einzigen Map, die mit Hilfe von EXEC CICS SEND MAP gesendet wird.

9.2.3 BMS Programmierung

```
S04SET  DFHMSD TYPE=MAP,MODE=INOUT,LANG=C,STORAGE=AUTO,TIOAPFX=YES
*       MENU MAP.
map020  DFHMDI SIZE=(24,80),CTRL=(PRINT,FREEKB)
        DFHMDF POS=(9,23),ATTRB=(ASKIP,NORM),LENGTH=34,
               INITIAL='WELCOME TO THE MAGIC WORLD OF CICS'
        DFHMDF POS=(12,27),ATTRB=(ASKIP,NORM),LENGTH=26,
               INITIAL='MAY THE FORCE BE WITH YOU!'
        DFHMSD TYPE=FINAL
(1)                     END
```

Abb. 9.2.5
BMS Programm

Ein BMS Programm verwendet ausschließlich drei Arten von Befehlen, nämlich

DFHMSD	**Data Facility Hierarchical Map Set Definition.** **Mit Hilfe dieses Befehls wird ein Mapset definiert.**
DFHMDI	**Data Facility Hierarchical Map Definition Information.** **Mit Hilfe dieses Befehls wird eine Map definiert.**
DFHMDF	**Data Facility Hierarchical Map Data Field.** **Mit Hilfe dieses Befehls wird ein Feld innerhalb einer Map definiert.**

Der Mapset hat den Namen S04SET, die einzige Map dieses Mapsets hat den Namen map020. Jede Map innerhalb des Mapsets muss durch einen Namen gekennzeichnet sein.

Der Parameter TIOAPFX=YES in dem DFHMSD Befehl fügt ein 12-Byte Feld an den Anfang jeder Map ein.

… und hier verraten wir Ihnen ein Geheimnis: Es gibt überhaupt keine eigene BMS Sprache. Was Sie hier sehen ist in Wirklichkeit ein Assembler Programm, welches ausschließlich aus Assembler Makros besteht, nämlich den drei Makros DFHMSD. DFHMDI und DFHMDF. IBM hat sich die Entwicklung einer eigenen BMS Sprache erspart, und stattdessen Assembler Macros verwendet. EXEC DFHMAPS in der JCL File bewirkt ein Assembler Compile, Link and Go.

Die beiden DFHMDF Befehle in der JCL File definieren 2 Felder in der Map, die mit Hilfe des EXEC CICS SEND MAP Befehls an den Terminal gesendet werden. Diese beiden Felder werden mit statischer Information initiiert. Für das erste der beiden Felder ist dies:

WELCOME TO THE MAGIC WORLD OF CICS

Angegeben wird, in welcher Zeile (Zeile 9) und Spalte (Spalte 23) dieses Feld in den Presentation Space Puffer geladen werden soll.

Es ist grundsätzlich möglich, CICS Maps mit Hilfe von Sprachen wie Cobol, C/C++ oder PL/1 zu erstellen. Dies geschah häufig in der Anfangszeit von CICS, wird heute aber praktisch so gut wie nie mehr gemacht.

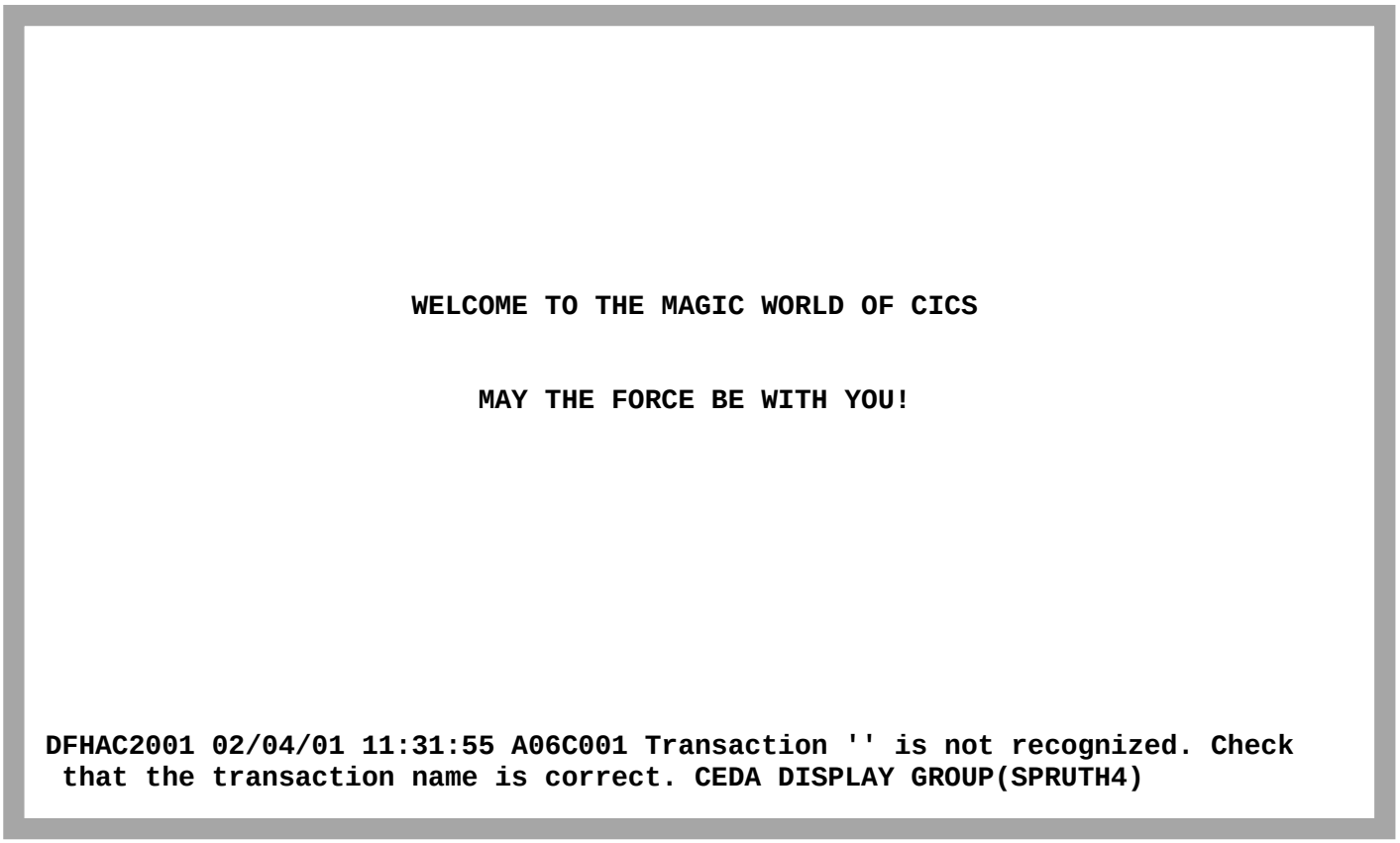

Abb. 9.2.6
Bildschirm Darstellung

… und dies ist das tolle Ergebnis unseres CICS Hello World Programms.

Im Anhang Abschnitt 9.2.8 befindet sich ein erweitertes Mapset-Beispiel mit 2 Maps.

9.2.4 Screen Scraping

CICS Terminals verwendeten ursprünglich die 3270/CUI Oberfläche und Darstellung. Später kam eine 3270 Protokoll basierte GUI dazu, die ein entsprechendes Umsetzungsprogramm in jedem Terminal erforderte.

Mit der Verfügbarkeit von PCs begann man, diese an Stelle der bisherigen Terminals einzusetzen. Dies ermöglichte auf dem Klientenrechner Programme, die den Inhalt des Presentation Puffer für die Erstellung einer (eingeschränkten) graphischen Ausgabe benutzte. Hierfür wurde die EPI (External Programming Interface) Schnittstelle zur Verfügung gestellt.

Diese Art der Verarbeitung wird auch heute noch vor allem für ältere CICS Anwendungen eingesetzt und als Screen Scraping bezeichnet.

Das Klienten Anwendungsprogramm (z.B. in Java geschrieben) empfängt den 3270 Datenstrom über die EPI (External Programming Interface) Schnittstelle und erzeugt eine graphische Darstellung der Datenausgabe. Der Vorteil ist, dass diese Art der Bildschirmdarstellung transparent für die Server-seitigen CICS Anwendungsprogramme ist.

Ein weiterer Vorteil ist eine einfache, schnelle und kostengünstige Umstellung auf eine graphische Darstellung. Es existieren zahlreiche Software Produkte von unterschiedlichen Herstellern, welche die Umstellung erleichtern. Ein Beispiel ist „Host-on-Demand“ von IBM; es existieren viele ISV (Independent Software Vendor) Alternativen (z.B. von der Firma Attachmate).

Erreicht wird eine gefälligere Darstellung, aber es fehlen manche Funktionen. Nicht möglich ist zum Beispiel eine Scroll Funktion mittels eines Scroll Balken, weil im 3270 Protokoll hierfür die Vorraussetzungen fehlen.

Probleme:

- **2000 Benutzer erfordern die Wartung für 2000 Emulatoren auf den Arbeitsplatzrechnern**
- **Schwierigkeiten mit der Wartung der Maps. Maps können nicht mehr bequem geändert werden, z.B. "move field 1 Byte“.**

Eine Server-seitige Lösung umgeht das Administrationsproblem. Z.B. „Host Access Transformation Services“ (HATS) von IBM läuft auf einem Server (z.B. einer zLinux Maschine) und konvertiert eine 3270 Nachricht in Browser-fähiges HTML Format.

9.2.5 Beispiel Computer-Schachprogramm

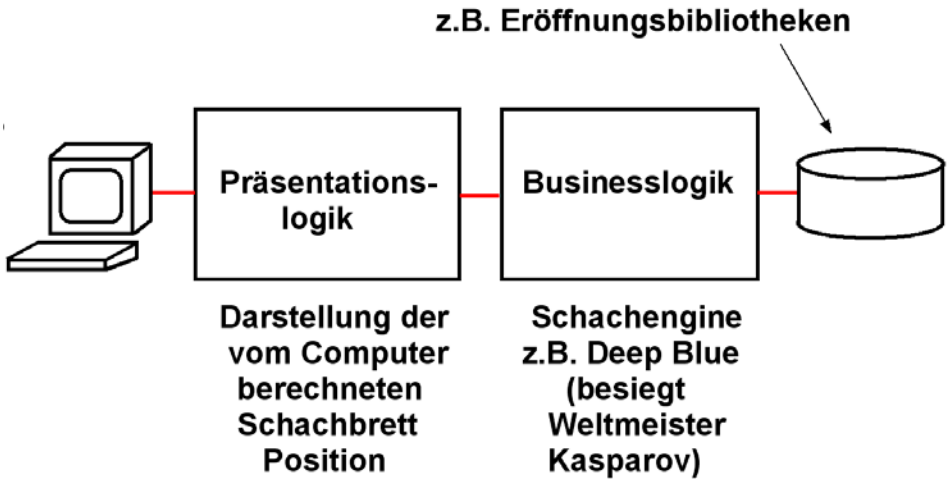

Abb. 9.2.7
Gliederung in Präsentationslogik und Businesslogik

Als Beispiel schreiben wir ein Computer Schachprogramm. (Wir übersehen, dass man ein Mainframe vermutlich nicht zum Schachspielen benutzen würde, und wenn doch, das Schachprogramm wahrscheinlich nicht unter CICS laufen würde).

Die Business Logik läuft als CICS Anwendung. Die Präsentationslogik wird einmal mit Hilfe von BMS erstellt und als 3270/CUI dargestellt, und ein zweites Mal mit Hilfe von Screen Scraping graphisch dargestellt.

```
Mein Zug: c5 x d4

Position:

Ke1, DF3, Ta1, h1, Lc1, e2, Ba2, c3, e5, g2, h2

Ke8, Dd8, Ta8, h8, Lc8, f8, Ba7, b7, d4, f7, g7,
h7

Ihr Zug _______
```

Abb. 9.2.8
Minimale Präsentationslogik

Abb. 9.2.8 zeigt die Darstellung mit Hilfe der BMS 3270/CUI Präsentationslogik. Der Inhalt des Schachbretts ist mit Hilfe der „International Chess Notation“ dargestellt. Wiedergegeben sind die Positionen der einzelnen Figuren auf dem Schachbrett. Der Computer spielt mit den schwarzen Steinen, und eine seiner Figuren auf dem Feld c5 hat gerade eine weiße Figur auf dem Feld d4 geschlagen.

Der dargestellte Bildschirminhalt ist in dieser Form im Presentation Space Puffer des Terminals abgelegt, und wird unverändert auf dem Bildschirm wiedergegeben. Mitglieder eines Schachclubs sind mit dieser Notation vertraut.

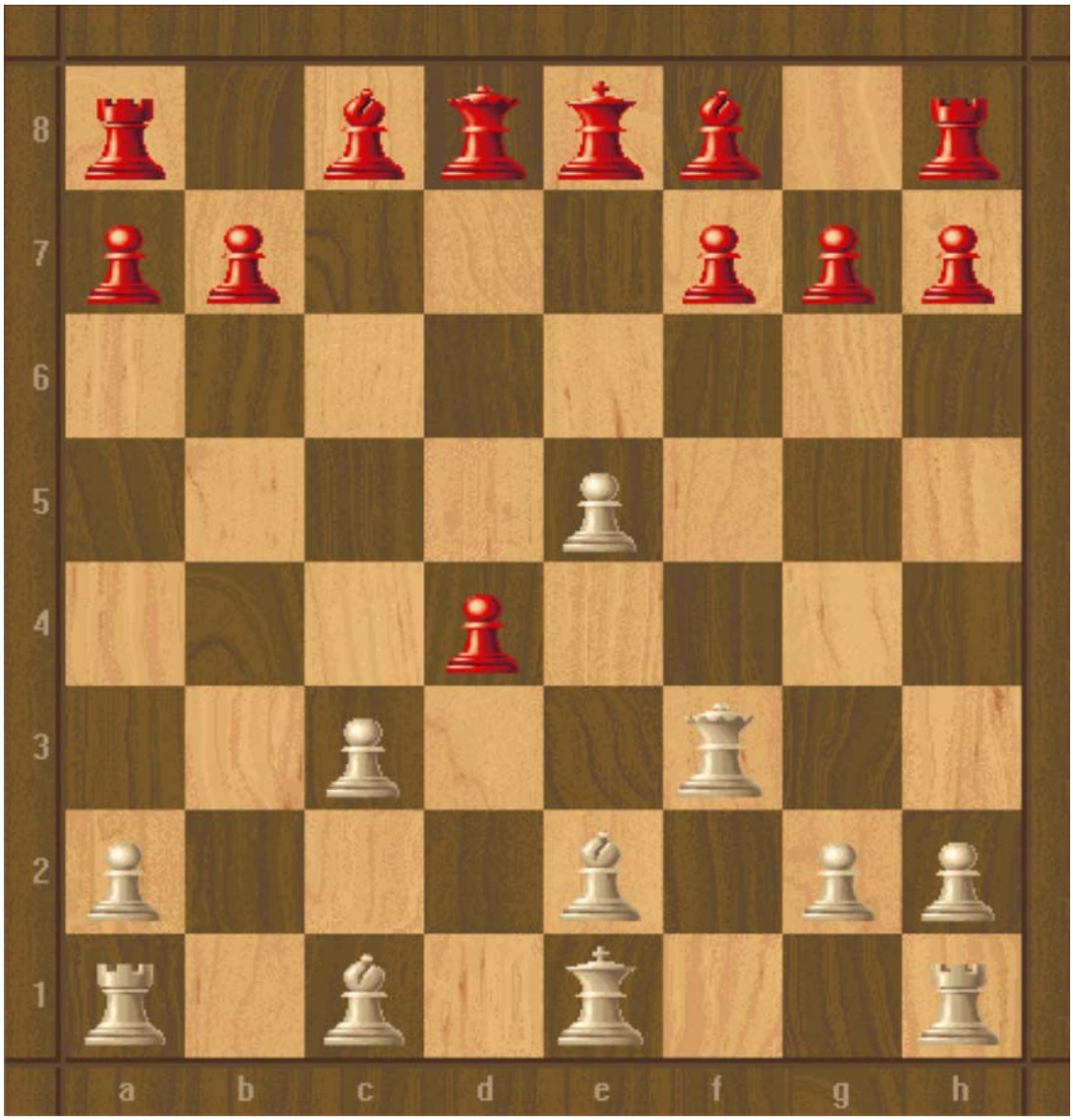

Abb. 9.2.9
Graphische Präsentations- Logik

Alternativ greift ein Java Programm mit Hilfe der EPI Schnittstelle auf den Inhalt des Presentation Space Buffers zu und bereitet die Darstellung, wie in Abb. 9.2.9 gezeigt, graphisch auf.

Wichtig ist, dass die gezeigten beiden Bildschirm-Darstellungen inhaltlich identisch sind.

Es sind zusätzliche Funktionen möglich. Z. B. kann die zuletzt gezogene Figur blinkend dargestellt werden, Figuren können mit der Maus bewegt werden, usw.

9.2.6 Screen Scraping in der Praxis

Unser Schachbeispiel ist sicher sehr einprägsam, aber sicher auch sehr realitätsfern. Ein mehr realistisches Beispiel würde z.B. den in Abb. 9.2.10 gezeigten Screen produzieren.

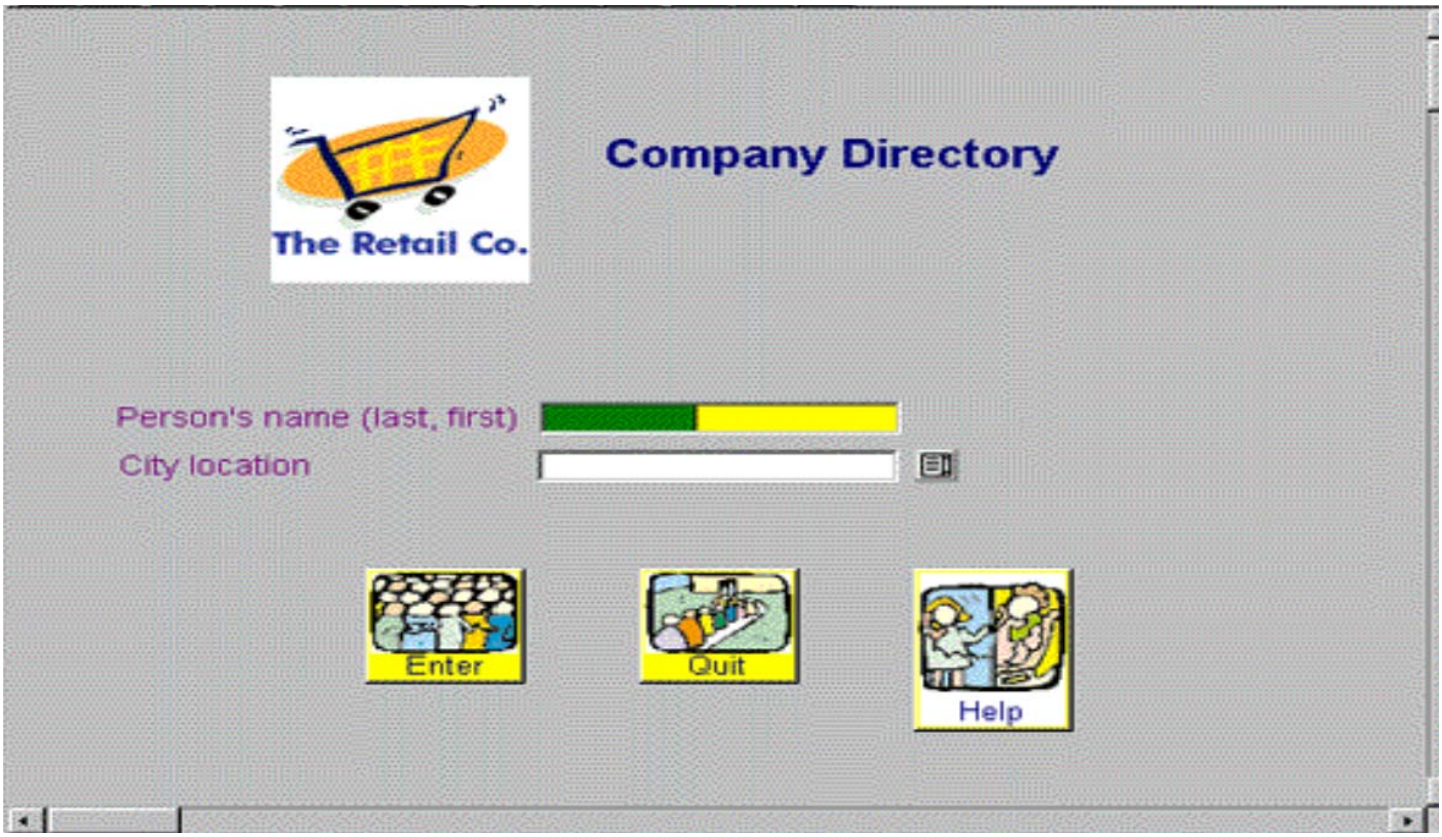

Abb. 9.2.10
Screen Scraping Ergebnis

Zu beachten ist, dass auch Aktionen mit der Maus möglich sind, die über die EPI Schnittstelle in 3270 Aktionen umgesetzt werde können, oder aber lokal (z.B. eine Help Funktion) abgearbeitet werden.

Nachteilig ist beim Screen Scraping Ansatz, dass ein Teil der Präsentationslogik auf dem Klientenrechner ausgeführt wird. Dies verursacht zusätzliche Administrationskosten.

Unintelligente 3270 Terminals haben den Vorteil, dass nicht sehr viel schief gehen kann. Das Terminal funktioniert entweder, oder er funktioniert nicht. Im Problemfall kommt der Techniker mit einem Ersatzterminal, prüft mit einem einfachen Testgerät ob der Kabelanschluss ok ist (ja/nein), wenn ja tauscht den Terminal aus und nimmt den alten Terminal mit in die Werkstatt. Für diese Tätigkeit ist keine Spezialausbildung erforderlich. PCs mit zusätzlicher Software als Terminal Ersatz erfordern schnell im Problemfall einen Spezialisten.

Zur Abhilfe setzt man heute als Terminals gerne PCs mit nur rudimentären Funktionen ein, z. B. ohne Plattenspeicher und ohne USB Anschluss, auf denen nur ein Browser läuft. Derartige PCs werden als „Thin Clients“ bezeichnet. Die 3270 Emulations- und Screen Scraping Software wird auf einen Server verlagert, der eine ganze Reihe von derartig abgerüsteten Arbeitsplatzrechnern bedient. Auf dem Server läuft Software, welche 3270 Daten in eine Browser Darstellung umsetzt.

Es gibt zahlreiche Software Produkte, die eine derartige Umsetzung durchführen. Ein Beispiel ist „Host Access Transformation Services“ (HATS) von der IBM. Wir werden später in Kapitel 18, „Java Connection Architecture“, ein Beispiel zeigen.

Die Firma IGEL Technology GmbH in 28199 Bremen ist ein führender europäischer Hersteller von Thin Client PC Hardware, siehe https://www.igel.com/de/. Besonders in der öffentlichen Verwaltung werden Thin Client PCs gerne eingesetzt.

9.2.7 Probleme des Screen Scraping Ansatzes

Viele Hersteller haben das 3270 Protokoll für ihre Client/Server Produkte eingesetzt, aber es existieren Probleme des Screen Scraping Ansatzes

Das 3270 Protokoll lässt einige Funktionen nicht zu, z.B.:

- **Scroll Bar**
- **Mehrere Fenster**

Als Lösung bieten sich zwei Ansätze an:

- **Erweiterung des 3270 Protokolls um die fehlenden Funktionen. Dies wurde z.B. von der Firma SAP für ihre System R/3 SAPGUI implementiert.**

- **Java und http für die Präsentationslogik. Wir werden dies später in dem Abschnitt 18.3 diskutieren.**

9.2.8 Anhang

Zur Vertiefung zeigen wir ein einfaches CICS Programm, welches zwei Maps in seinem Mapset verwendet.

Die erste Map (Eingabe Map) stellt einen Eingabe Screen dar, in den der Benutzer zwei Zahlen in zwei dafür vorgesehene Felder eingeben kann.

Die zweite Map stellt einen Ergebnis Screen dar. Hier wird das Ergebnis der Addition der beiden Zahlen wiedergegeben.

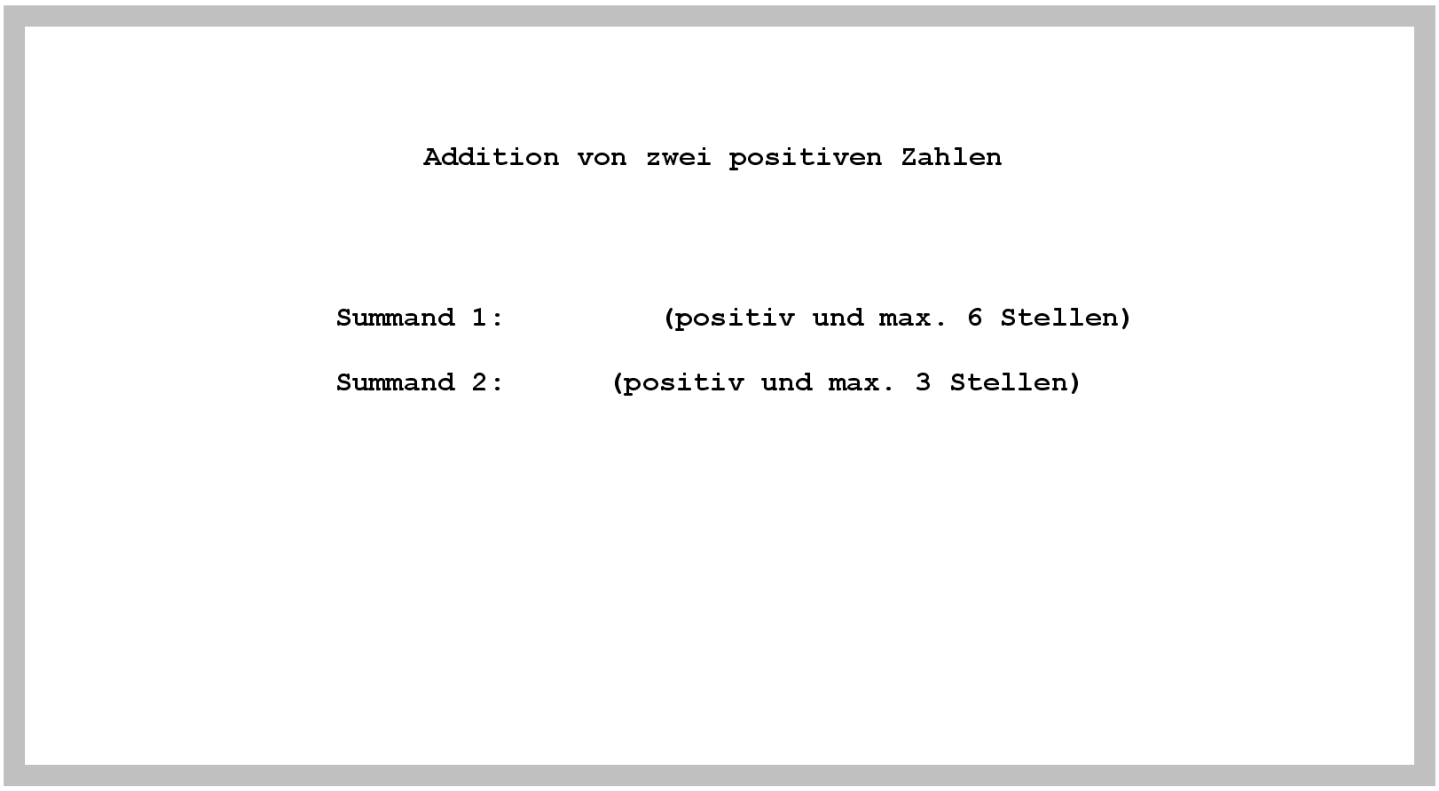

Abb. 9.2.11
Dies ist die Eingabe Map

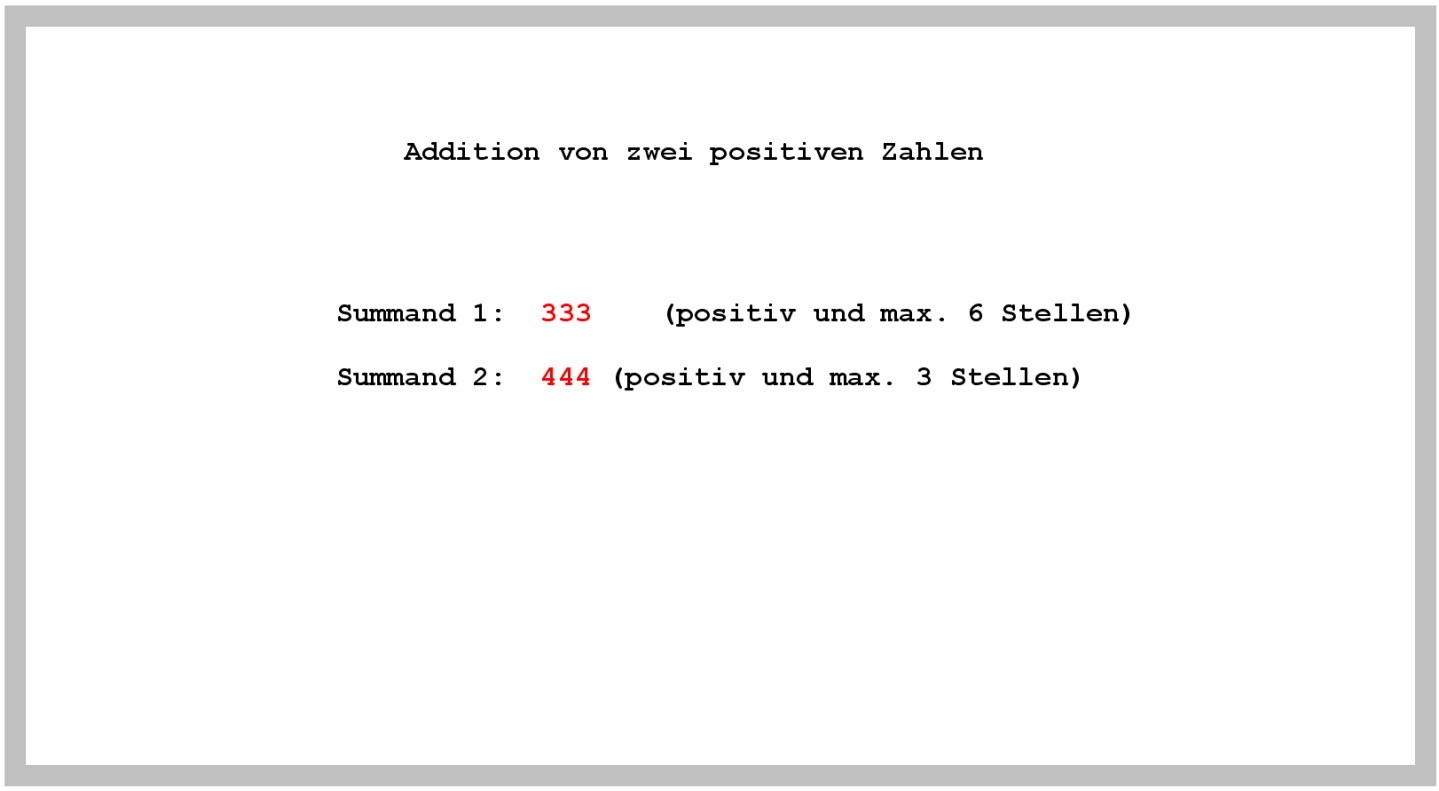

Abb. 9.2.12
Input in die Eingabe Map

Dies ist die Eingabe Map, nachdem der Benutzer 2 Zahlen in die dafür vorgesehen (nicht markierten Felder) eingegeben hat, ...

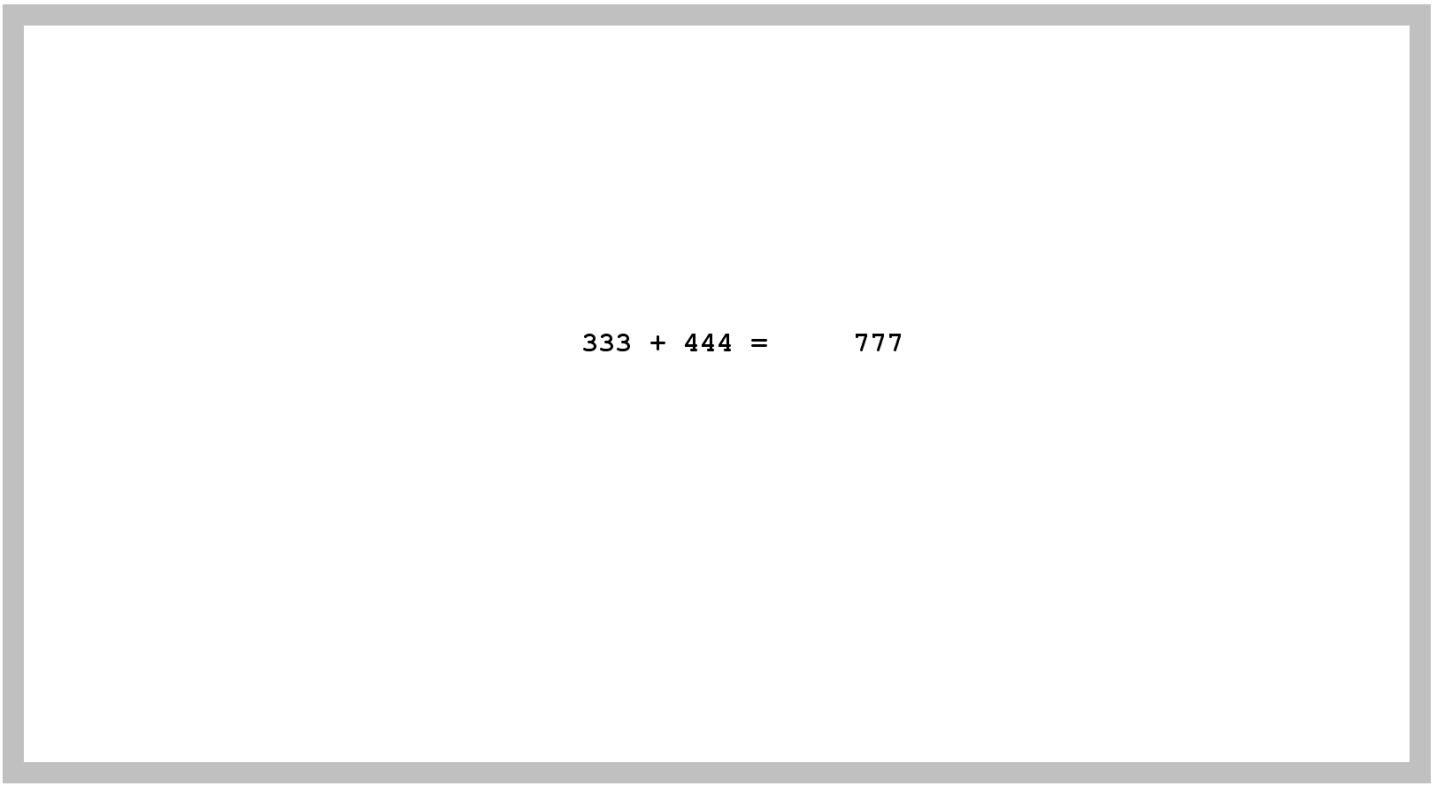

Abb. 9.2.13
Ausgabe Map

… und dies ist die CICS Ausgabe. Die nächsten beiden Abbildungen zeigen das dafür benutzte BMS Programm, welches aus zwei ISPF Panels besteht.

```
  File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
--------------------------------------------------------------------------------
EDIT       PRAKT20.CICS.BMS(MAPSET) - 01.25               Columns 00001 00072
****** ***************************** Top of Data ******************************
==MSG> -CAUTION- Profile changed to CAPS OFF (from CAPS ON) because data
==MSG>           contains lower case characters.
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000100 //PRAKT20M JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID
000101 //ASSEM    EXEC DFHMAPS,MAPNAME='M3BM020',RMODE=24
000102 //SYSUT1   DD *
000110 M3BM020 DFHMSD TYPE=MAP,MODE=INOUT,LANG=COBOL2,STORAGE=AUTO,TIOAPFX=YES
000120 *       Die Eingabemap des Mapsets.
000200 EINGMAP DFHMDI SIZE=(24,80),LINE=1,COLUMN=1,CTRL=FREEKB
000400         DFHMDF POS=(5,22),LENGTH=34,ATTRB=(ASKIP,NORM),                X
000500                INITIAL='Addition von zwei positiven Zahlen'
000700         DFHMDF POS=(10,18),LENGTH=10,ATTRB=(ASKIP,NORM),               X
000800                INITIAL='Summand 1:'
000900 A       DFHMDF POS=(10,30),LENGTH=6,ATTRB=(UNPROT,NUM,IC)
000910         DFHMDF POS=(10,37),LENGTH=28,ATTRB=(ASKIP,NORM),               Z
000920                INITIAL='(positiv und max. 6 Stellen)'
Command ===>                                                 Scroll ===> PAGE
  F1=Help       F3=Exit       F5=Rfind      F6=Rchange   F12=Cancel
```

Mapset

Map # 1

Abb. 9.2.14
Die erste Hälfte des BMS Programms

```
  File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
--------------------------------------------------------------------------------
EDIT       PRAKT20.CICS.BMS(MAPSET) - 01.25               Columns 00001 00072
001100         DFHMDF POS=(12,18),LENGTH=10,ATTRB=(ASKIP,NORM),               E
001110                INITIAL='Summand 2:'
001200 B       DFHMDF POS=(12,30),LENGTH=3,ATTRB=(UNPROT,NUM)
001210         DFHMDF POS=(12,34),LENGTH=28,ATTRB=(ASKIP,NORM),               W
001211                INITIAL='(positiv und max. 3 Stellen)'
001212 *       Die Ausgabemap des Mapsets.
001320 AUSGMAP DFHMDI SIZE=(24,80),CTRL=(PRINT,FREEKB)
001330 SUMMND1 DFHMDF POS=(11,29),ATTRB=(ASKIP,NORM),LENGTH=6
001340         DFHMDF POS=(11,36),ATTRB=(ASKIP,NORM),LENGTH=1,INITIAL='+'
001341 SUMMND2 DFHMDF POS=(11,38),ATTRB=(ASKIP,NORM),LENGTH=3
001342         DFHMDF POS=(11,42),ATTRB=(ASKIP,NORM),LENGTH=1,INITIAL='='
001350 SUMME   DFHMDF POS=(11,44),ATTRB=(ASKIP,BRT),LENGTH=7
001410         DFHMSD TYPE=FINAL
001500         END
001600 /*
001700 //
****** **************************** Bottom of Data ****************************

Command ===> SUB                                             Scroll ===> PAGE
 F1=Help       F3=Exit       F5=Rfind      F6=Rchange   F12=Cancel
```

Map # 2

Abb. 9.2.15
Die zweite Hälfte des BMS Programms, in der die Map für die Ergebnisausgabe definiert wird.

9.3 Multiregion and Intersystem Communication

9.3.1 CICS Interprocess Communication

Als Pfadlänge bezeichnet man die Zahl der während der Abarbeitung einer Transaktion ausgeführten Maschinenbefehle. Eine typische CICS oder IMS Transaktionen hat eine Pfadlänge von mehr als 100 000 Maschinenbefehlen.

Transaktionen mit einer Pfadlänge von 1 Million Maschinenbefehlen sind nicht ungewöhnlich.

Annahme: Eine CPU führt 1 Milliarde Maschinenbefehle / Sekunde aus.

Bei einer Pfadlänge von 250 000 Maschinenbefehlen sind 4 000 Transaktionen / Sekunde theoretisch möglich, wenn man Overhead vernachlässigt.

Große Installationen bewältigen 5000 Transaktionen / Sekunde. Spitzenwert von 9 000 Transaktionen / Sekunde sind schon aufgetreten. Das IBM Entwicklungslabor in Hursley, Südengland hat 19 000 Transaktionen / Sekunde demonstriert.

Derartig hohe Transaktionsraten bedingen mehrere CPUs in einer „Symmetrischen Multiprozessor“ (SMP) Konfiguration und häufig auch mehrere Mainframe Rechner..

Hardware	klein/einfach	mittel	groß/komplex
Endbenutzer	100	10 000	> 100 000
CPUs	1 - 3	5 -20	> 100
Plattenspeicher TeraByte	1 - 10	10 – 100	100 – 5 000
Magnetband Terabyte	100	100 – 1000	> 10 000
Software			
Transaktions-programme X)	400	4 000	40 000
Quell-programme XX)	1 000	10 000	100 000

Abb. 9.3.1
Anzahl der Komponenten eines Transaktionsverarbeitungssystems

X) einschl. Berichte, Maps XX) einschl. alte Versionen

Pfadlänge pro Transaktion: 100 000 - 1 000 000 Maschinenbefehle

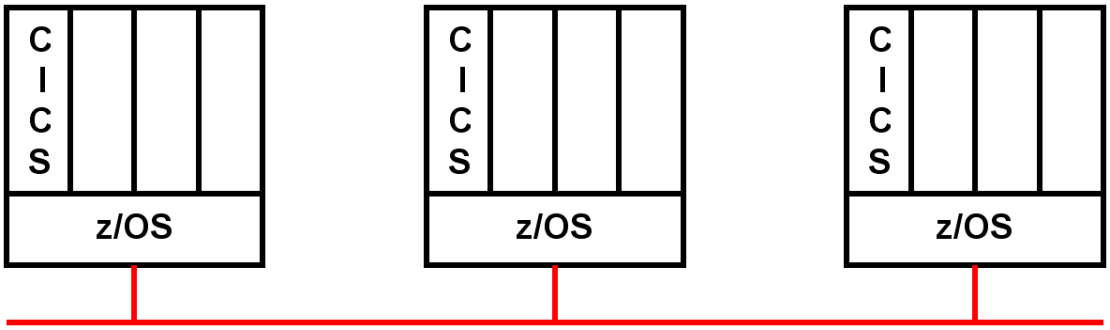

Abb. 9.3.2
Verbund mehrerer CICS Instanzen

Häufig ist gewünscht, dass 2 CICS Transaktionen, die auf unterschiedlichen CICS Transaktions-Monitoren laufen, miteinander kommunizieren können. Es existieren 2 Arten von CICS Interprocess Communication:

- **Multiregion operation (MRO)**
- **Intersystem communication (ISC)**

Multiregion Operation (MRO) wird für die Communication von zwei CICS Regions benutzt, die sich auf dem gleichen Mainframe System befinden.

Ein Spezialfall ist ein Verbund aus mehreren Mainframe Rechnern, wobei auf jedem Rechner eine CICS Region installiert ist.

Für einen Cluster bestehend aus mehren Mainframes innerhalb eines Rechenzentrums existiert eine als „Parallel Sysplex“ bezeichnete Integrationssoftware, die ebenfalls MRO zwischen unterschiedlichen z/OS Instanzen ermöglicht. Sind die z/OS Rechner in geographisch voneinander getrennten Rechenzentren untergebracht, spricht man von einem Geographically Dispersed Parallel Sysplex (GDPS, siehe Abschnitt 14.1.1), Das Thema Sysplex wird in Kapitel 12 behandelt.

Intersystem Communication (ISC) wird für die Communication von zwei CICS Systemen benutzt, die lediglich über TCP/IP und das Internet miteinander verbunden sind. Hierbei können die CICS Instanzen alle auf z/OS, oder aber auch auf Unix oder Windows Rechnern installiert sein.

9.3.2 CICS Multiregion Operation

CICS spezifische MRO-Standards sind:

Transaction Routing

Transaktionen (Eingabe vom Klienten) können zwecks Ausführung von einem CICS System einem anderen CICS System unverändert übergeben werden

Function Shipping

Eine Anwendung in einem CICS System kann auf Daten eines anderen CICS Systems zugreifen

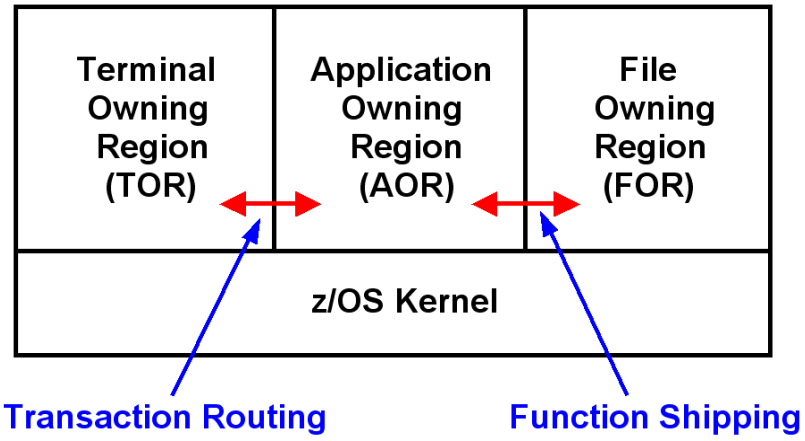

Abb. 9.3.3
Benutzung von Tranaction Routing und Function Shipping

CICS Multiregion Operation (MRO) ermöglicht es zwei oder mehr CICS Systemen, die auf dem gleichen z/OS System oder innerhalb des gleichen Parallel Sysplex laufen, miteinander zu kommunizieren.

Relativ häufig wird CICS auf 3 Regionen (virtuelle Adressenräume) innerhalb des gleichen Rechners aufgeteilt, die jede über einen eigenen CICS Transaktionsmonitor verfügen, und über MRO miteinander kommunizieren.

Die Terminal Owning Region verwaltet alle angeschlossenen Terminals. Sie gibt über das „Transaction Routing“ Protokoll die Transaktion an die Application Owning Region weiter.

Die Application Owning Region führt die Anwendung aus. Wenn diese Daten (z.B. von einem VSAM Data Set) benötigt werden, werden diese von der File Owning Region über das Function Shipping Protokoll zur Verfügung gestellt.

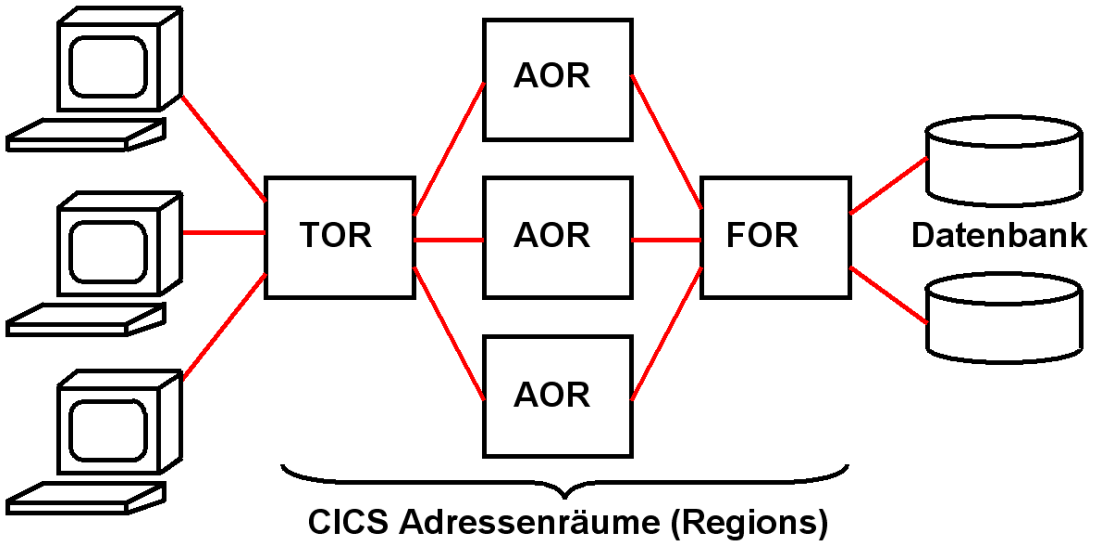

Abb. 9.3.4
Verteilung auf mehrere Adressenräume

Gezeigt ist ein Beispiel mit 5 CICS Instanzen (Ausprägungen), die in 5 virtuellen Adressenräumen (Regions) laufen. In einem symmetrischen Multiprozessor können die 5 Instanzen auf 5 CPUs verteilt werden.

TOR	**Terminal Owning Region**
AOR	**Application owning Region**
FOR	**File owning Region (Resource Region)**

Die AORs können für bestimmte Anwendungen, spezielle Zugriffsarten, Datenbankverbindungen usw. optimiert sein. AORs können auf unterschiedliche Regions eines Sysplex verteilt sein.

Es existieren Mainframe Installationen mit über 100 CICS Instanzen in ebenso vielen CICS Regions.

Warum mehrfache AORs ?

Prozesse unter z/OS sind durch einen Task Control Block (TCB) gekennzeichnet. Der TCB ist ein Speicherbereich, der z.B. den Inhalt der Mehrzweck-, Gleitkomma-, Controlregister und das Programm Status Wort (PSW) aufnimmt, wenn der Prozess von dem laufenden in den wartenden Zustand versetzt wird.

CICS einschließlich Nucleus und aller Anwendungsprogramme läuft in einem einzigen z/OS Address Space und wird in den Augen des z/OS Kernels durch einen einzigen TCB (Task Control Block) repräsentiert.

Innerhalb des CICS Address Spaces laufen gleichzeitig Hunderte oder tausende von Transaktionen, die durch die Scheduler Komponente des CICS Nucleus gescheduled werden. Jede Transaktion wird durch einen „quasi-reentrant“ TCB (QR TCB) repräsentiert. Im Gegensatz zu normalen TCBs werden die QR TCBs nicht vom Betriebssystem Kernel, sondern vom CICS Nucleus gescheduled und verwaltet. Dies geschieht unsichtbar für den z/OS Kernel.

Da CICS nur über einen einzigen TCB verfügt, kann der z/OS Kernel es auch nur für eine einzige CPU schedulen. Nur ein einziger QR TCB ist in jedem Augenblick active (running). Multiple AORs bedeuten multiple CICS Instanzen und erlauben, die Transaktionsverarbeitung auf mehreren CPUs zu verteilen. Der z/OS Work Load Manager (WLM, diskutiert in Kapitel 14) ordnet neue Transaktionen den einzelnen AORs zu.

9.3.3 Quasireentrant Programme

Multithreading ermöglicht es, dass eine einzelne Kopie eines Anwendungsprogramms von mehreren Transaktionen gleichzeitig ausgeführt wird. Zum Beispiel kann eine Transaktion anfangen, ein Anwendungsprogramm auszuführen. Wenn ein EXEC CICS Befehl erreicht wird, wird die Transaktion in den Wartezustand versetzt. Der CICS Dispatcher scheduled eine andere Transaktion, welche die gleiche Kopie des Anwendungsprogramms ausführt.

Multithreading verlangt, dass alle CICS Anwendungsprogramme quasi-reentrant geschrieben sind. „Quasi-Reentrant" bedeutet, dass das Anwendungsprogramm vor und nach jedem EXEC CICS Befehl in einem konsistenten Zustand sein muss. Das Anwendungsprogramm muss seriell wiederverwendbar zwischen Einstiegs- und Ausstiegspunkten sein. Quasi Reentrance garantiert, dass jeder Aufruf eines Anwendungsprogramms unbeeinflusst von früheren Durchläufen, oder dem Multithreading durch mehrere CICS Tasks ist. CICS Anwendungsprogramme mit der EXEC CICS-Schnittstelle gehorchen dieser Regel automatisch. Für COBOL, C/C++ und PL/1 Programmen wird Reentrance erreicht, indem jede Transaktion eine eigenen Speicherbereich für die Speicherung von Variablen erhält.

Eine gewisse Ähnlichkeit mit Java Objekten ist unverkennbar.

CICS Anwenderprogramme (Transaktionen) laufen unter einer CICS managed Task Control Block (TCB). Wenn die Programme als Quasi-Reentrant definiert sind (das CONCURRENCY Attribut der Programm Ressourcen-Definition gesetzt ist), läuft CICS immer unter dem CICS „Quasi-Reentrant (QR) TCB“. Die Voraussetzungen für ein Quasi-Reentrant Programm in einem Multithreading Kontext sind weniger streng, als wenn das Programm gleichzeitig auf mehreren CPUs ausgeführt werden soll.

CICS Transaktionen können alternativ unter dem “Open TCB” an Stelle des QR TCBs laufen. Open TCBs werden vom Betriebssystem Kernel (und nicht vom CICS Nucleus) gescheduled und können damit mehreren CPUs zugeordnet werden. In anderen Worten, die Multiprocessor Eigenschaften eines Mainframes werden genutzt. Hiermit kann der Gesamtdurchsatz verbessert werden; allerdings erfordert das Scheduling durch den Betriebssystem Kernel mehr Aufwand (CPU Zyklen) als das Scheduling durch den CICS Nucleus. Vor allem Java Programme machen vom Open TCB Gebrauch.

Damit Open TCB Programme die Integrität der gemeinsam genutzten Ressourcen wahren , müssen Serialisierungs-Techniken verwendet werden, welche den gleichzeitigen Zugriff auf gemeinsam genutzte Ressourcen verhindern. Programme, welche die Serialisierung beim Zugriff auf gemeinsam genutzte Ressourcen sicherstellen, werden als „threadsafe“ oder "reentrant“ bezeichnet.

9.3.4 AORs und VSAM

AORs laufen in getrennten z/OS Regions. Die dort laufenden Programme können unabhängig voneinander auf die gleichen DB2 Daten zugreifen. Das DB2 Datenbanksystem verfügt über entsprechende Lock Management Einrichtungen.

Die weltweiten, von CICS verarbeiteten VSAM Datenbestände, haben eine ähnliche Größe wie die von CICS verarbeiteten DB2 Datenbestände. Häufig greift eine CICS Anwendung parallel sowohl auf VSAM wie auf DB2 Daten zu. VSAM fehlen jedoch die Lock Management Einrichtungen, die für einen parallelen Zugriff durch mehrere AORs erforderlich sind.

Abhilfe schafft eine als „VSAM Record Level Sharing“ (RLS) bezeichnete Funktion, die von DFSMS (siehe Abschnitt 3.3 „z/OS Betriebssystem“) als getrenntes Subsystem (SMSVSAM) zur Verfügung gestellt wird. CICS RLS ermöglicht es, dass zahlreiche CICS Anwendungen in unterschiedlichen AORs parallel auf die gleichen VSAM Data Sets mit voller Update Fähigkeit zugreifen können.

9.3.5 CICSPlex

Wenn mehrere CICS Instanzen in getrennten Regions installiert werden, nennt man die resultierende Architektur einen CICSPlex. Als CICSPlex wird eine Gruppe von logisch verbundene CICS Regions betrachtet. Ein typischer CICSPlex ist eine miteinander verbundene Gruppe von TORs, AORs, FORs usw. Auf einem Sysplex-Verbund von Mainframe Rechnern wird normalerweise ein (oder mehrere) CICSPlex eingesetzt.

Jede CICS Instanz verwaltet eine Region (Address Space). Jede CICS Region kann für eine bestimmte Aufgabe optimiert sein, und dennoch leicht mit anderen CICS Regions kommunizieren.

Ein System von miteinander verbundenen CICS Regions kann z.B. enthalten:

- **Test Regions für das Austesten neuer Anwendungen**
- **Function-spezifische Regions für die Steuerung von Terminals, Benutzer Interfaces oder System Services**
- **Nutzung spezifischer Regions für das Management bestimmter Anwendungen, wie z.B. Datenbanken**

Eine spezielle CICS Komponente, der CICSPlex System Manager (CICSPlex SM)ist für die Steuerung einer Vielzahl von CICS Regions zuständig. In zunehmendem Maß finden wir Unternehmen mit Dutzenden oder Hunderten von CICS Regions.

Ein Beispiel ist die Installation einer Schweizer Großbank. Die Union Bank of Switzerland unterhält in zwei Standorten in der Nähe von Zürich zwei Rechenzentren, die als Geographically Dispersed Parallel Sysplex (GDPS) zusammengeschaltet sind. Ein weiterer GDPS befindet sich im Staate New Jersey in den USA.

Die CICS Konfiguration des Schweizer GDPS besteht aus (3Q 2006):

- **11 CICSPlexes**
- **943 CICS Regions**

Die CICS Konfiguration des USA GDPS besteht aus:

- **4 CICSPlexes**
- **194 CICS Regions**

9.3.6 Intersystem Communication (ISC)

CICS Multi Region Operation findet innerhalb eines einzigen z/OS Systems oder innerhalb eines Sysplex statt, der darauf abzielt, ein „Single System Image“ anzubieten. Eine CICS Kommunication zwischen zwei (ansonsten unkoordinierten) Rechnern wird als CICS Intersystem Communication (CICS ISC) bezeichnet.

Es existieren viele, nicht CICS spezifische Arten, von Intersystem Communication (ISC).

Einfache Beispiele für ISC-Standards sind Sockets, Named Pipes und APPC. In verteilten Systemen ist ISC häufig schwierig zu entwickeln, debug, portieren, besonders für unterschiedliche Plattformen, z.B. auf Grund von Abhängigkeiten von der Rechnerhardware (Byte Order, Adressierung).

Der RPC (Remote Procedure Call) ist eine klassische Software für die Implementierung von Client/Server Konfigurationen. Ein RPC liefert automatisch Funktionen, die beim Programmieren mit Sockets vom Programmierer erstellt werden müssen, z.B. Fehlerbehandlung, Adressierung und Datendarstellung. RPC erleichtert gegenüber Sockets die Entwicklung und Wartung von verteilten Anwendungen.

Es existieren eine ganze Reihe von inkompatiblen RPC Standards, z.B. Sun RPC, DCE RPC, SOAP (Web Services), sowie objekt-orientierte Versionen wie CORBA und RMI. Der RPC ist ein synchrones Protokoll, d.h. der Klient wartet (blockiert) auf die Antwort des Servers. Im Gegensatz dazu ist MQSeries ein weit verbreitetes asynchrones RPC Protokoll.

Normalerweise wird ISC durch Bibliotheken eines Betriebssystems implementiert.

CICS Distributed Program Link (DPL) ist eine CICS-spezifische, besonders einfach zu benutzende, Art des Remote Procedure Call. Voraussetzung ist, dass sowohl der Client als auch der Server unter CICS laufen. DPL benutzt TCP/IP, SNA oder NetBios in der Netzwerk Schicht 4.

Ein Programm eines CICS-Systems kann mit Hilfe des DPL Kommandos EXEC CICS LINK eine Verbindung mit einem Programm eines anderen CICS-Systems aufnehmen.

Es existieren viele Vereinfachungen gegenüber dem normalen RPC, da einheitliche Datendarstellung, Namenskonventionen, Fehlerbehandlung, u.s.w. Z.B. existieren keine Datenrepräsentations-Probleme, da alle CICS Implementierungen einen einheitlichen EBCDIC Standard einsetzen.

CICS TP Monitore existieren für viele unterschiedliche Betriebssystem-Plattformen, neben z/OS z.B. für Windows, i5/OS, oder Unix. Eine DPL Kommunikation zwischen CICS TP Monitoren auf unterschiedlichen Platformen ist sehr einfach zu implementieren.

Neben DPL existieren weitere CICS ISC Protokolle mit spezifischen Eigenschaften, z.B. Distributed Transaction Programming (DTP).

9.3.7 Distributed CICS

Der CICS Transaction Server wurde ursprünglich für Mainframe Rechner entwickelt. Heute existiert zusätzlich eine relativ weit verbreitete Version, die als „Distributed CICS“ bezeichnet wird und die auf anderen Plattformen läuft, z.B.:

- **Solaris,**
- **Microsoft Windows,**
- **IBM AIX,**
- **Hewlett-Packard UNIX (HP-UX) und**
- **Linux**

Distributed CICS Systeme sind über das Imternet miteinander verbunden

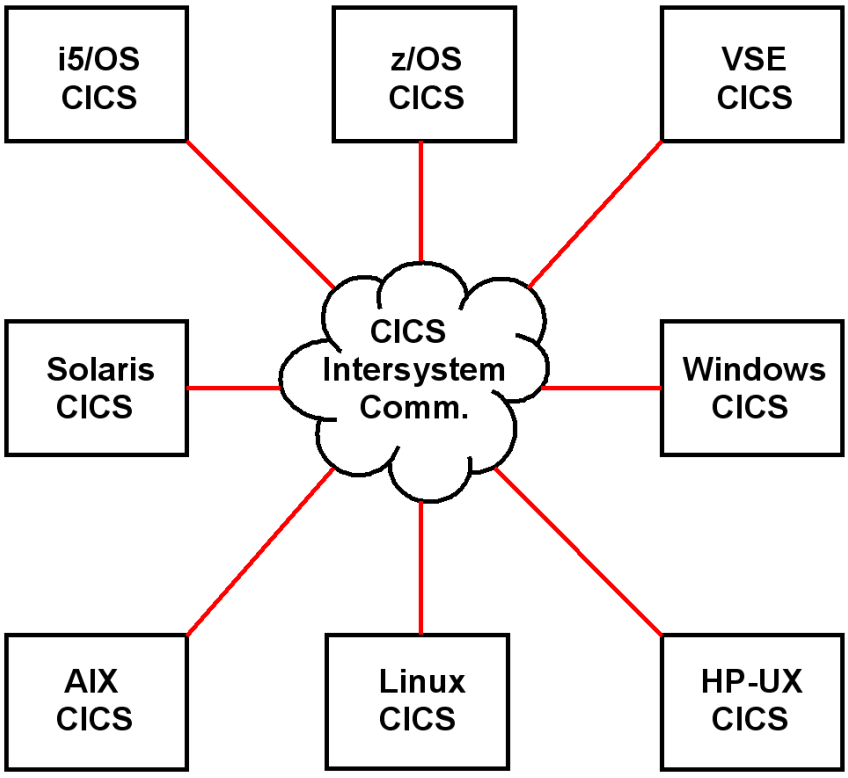

Abb. 9.3.5
Verbund von Mainframe und Distributed CICS Instanzen

Distributed CICS verfügt über die gleiche Application Programming Interface (API) und die gleiche System Programming Interface (SPI) wie der Mainframe CICS Transaction Server.

Es unterstützt eine Vielzahl von Datenbanken wie DB2, Oracle, IBM Informix, Microsoft SQL Server und Sybase. Datenintegrität wird durch die Unterstützung der XA-Schnittstelle bereitgestellt. XA ist die Industrie-Standard-Schnittstelle für das Commitment und die Recovery von transaktionalen Daten, einschließlich des Two-Phase Commit-Prozesses. Weiterhin verfügt Distributed CICS über einen „Structured File Server“ (SFS), eine Record-orientierte VSAM Emulation, welche indexierte, relative und sequentielle File Zugriffe ermöglicht. Unterstützt wird das Äquivalent eines Open TCB, nicht aber eines QR TCB.

Anwendungen können in C, C++, PL/I, COBOL, und Java geschrieben werden und sind problemlos auf die Mainframe CICS Version portierbar.

Spezifisch ist hiermit eine sehr einfache und problemlose Intersystem Communication möglich.

http://www.redbooks.ibm.com/redbooks/pdfs/sg247185.pdf

Als Beispiel können CICS Programme auf unterschiedlichen Plattformen (z.B. z/OS, Linux, Windows) Daten über die CICS COMMUNICATION AREA (COMMAREA) austauschen

In COBOL ist dies ein Teil der DATA DIVISION/LINKAGE SECTION, z.B.

```
01 DFHCOMMAREA.
    05 PROCESS-SW           PIC X.
       88 INITIAL-ENTRY     VALUE ‘0’.
       88 VERIFICATION      VALUE ‘1’.
    05 ACCOUNT-NUMBER       PIC X(10).
            •
            •
            •
EXEC LINK PROGRAM(ACCTPGM)
      COMMAREA(DFHCOMMAREA)
      LENGTH(11)
END-EXEC.
```

Abb. 9.3.6
COMMAREA als Cobol Copybook

9.4 Weiterführende Information

Die Benutzung von mobilen Endgeräten für einen Zugriff auf CICS Anwendungen hat in den letzten Jahren zunehmend an Bedeutung gewonnen. Zahlreiche Hersteller von Software Produkten sind mit eigenen Entwicklungen in diesen Markt eingestiegen.

Ein Beispiel ist die Firma Hostbridge, welche Software Produkte für die CICS Communication herstellt. Ein Beispiel für einen Zugriff auf den z/OS CICS Transaction Server mittels eines iPads oder anderer Android-Endgeräte ist zu finden unter

www.cedix.de/VorlesMirror/Band1/HostBridgeWIRE.pdf

Das folgende Video demonstriert, wie man mit einem normalen iPhone auf eine CICS VSAM Anwendung zugreift.

http://testiphone.com/?address=on&url=http://zserveros.demos.ibm.com:9080/iPhone/egl.html

und dieses Video demonstriert, was man mit dem CICS iPhone alles machen kann.

http://www.youtube.com/watch?v=5JyJ0XXR_3c

Das nächste Video enthält eine detaillierte Vorführung, wie mit Hilfe des Rational Developers for System z (RDz) eine iPhone Anwendung entwickelt wird, welche auf eine existierende z/OS COBOL Transaktion zugreift.

http://www.youtube.com/watch?v=5Fu66xMQdcY

Ein Tutorial über CICS for Intersystem Communication finden sie hier

http://docstore.mik.ua/univercd/cc/td/doc/product/ibd/ctrc/ctrchost.htm