

6. Customer Information Control System CICS

6.1 Überblick

6.1.1 CICS

In der Wirtschaft werden eine ganze Reihe von TP Monitor Produkten eingesetzt. Für Unix Systeme ist Tuxedo von der Firma BEA Systems populär. Der MS-Transaktionsserver ist in Microsoft Umgebungen häufig zu finden. SAP- und IMS-TP-Monitore sind in SAP- und IMS-Umgebungen zu finden.

Der mit Abstand am weitesten verbreitete Transaktionsmonitor ist CICS (ausgesprochen Kicks oder auch Zicks). 1999 setzen weltweit etwa 15 000 Unternehmen CICS ein. Es wird angenommen, daß bei 490 der weltweit 500 größten Unternehmen CICS verwendet wird. Von den 2000 größten Unternehmen setzen ca. 90% CICS ein. Sie generieren etwa 20 Milliarden Transaktionen pro Tag. Außenstehende, z.B. Jim Gray (Microsoft) vermuten, daß die Anzahl noch größer ist, und daß CICS- Transaktionen im Durchschnitt komplexer sind und mehr Arbeit leisten als dies bei anderen Transaktionsmonitoren der Fall ist.

Momentan existieren etwa 30 Mill. CICS-Terminals. Im Vergleich dazu waren im Jahre 1999 ca. 500 Mill. PC's und 200 Mill. Internet-Anschlüsse installiert, die meisten davon in Privathaushalten. Laut IBM war 1999 die Anzahl der weltweit ausgeführten CICS-Transaktionen etwa so groß wie die Anzahl der Hits auf Seiten des World Wide Web. Wenn man bedenkt, daß die durchschnittliche CICS- Terminal-Benutzungsdauer ca. 4 - 6 Stunden / Tag beträgt, erscheint dies plausibel.

CICS ist unter den S/390-Betriebssystemen OS/390 und VSE sowie als getrennte Implementierung (als Encina-Erweiterung) unter OS/400, OS/2, NT, AIX, HPUX, Sinix, Solaris sowie Digital Unix verfügbar. Die Kompatibilität zwischen den einzelnen Versionen ist excellent. CICS verdankt seine Popularität einer Spitzenposition bezüglich Durchsatz, Zuverlässigkeit und Verfügbarkeit.

Unter OS/390 (und VSE) laufen alle CICS-Anwendungen und -Dienste im Problemstatus, ungeschützt voneinander innerhalb eines einzigen virtuellen Adressraums. Anwendungen und Ressource- Manager laufen als Threads innerhalb dieses Adressraums.

Bei der Implementierung eines Transaktionsmonitors wie CICS ist zu bedenken, daß Operationen über Adressraumgrenzen hinweg Millisekunden benötigen, während Operationen im gleichen Adressraum nur Mikrosekunden-Bruchteile erfordern. Ein Faktor 1000 für den Geschwindigkeitsunterschied ist möglich. Deshalb läuft der vollständige CICS-Transaktionsmonitor unter OS/390 im Problemstatus. Dies erfordert stabile Anwendungen, da alle Steuerfunktionen nicht geschützt sind. Der Novell-Netware-Server benutzt einen ähnlichen Ansatz.

6.1.2 TPF

Ein mit CICS verwandtes Produkt ist die IBM Transaktion Processing Facility (TPF). TPF ist ein eigenständiges Betriebssystem, das nur eine einzige Anwendung erlaubt, nämlich die Transaktionsverarbeitung. Bei TPF laufen Anwendungen, Transaktionsmonitor und Überwacher im Überwacherstatus. Eigenschaften sind:

- Keine Einrichtungen für Softwareentwicklung (erfolgt auf einem anderen Rechner)
- Run-to-Completion (non-preemptive) Scheduler/Dispatcher
- Betriebssystem-Aufruf erfordert ca. 500 Maschinenbefehle
- E/A-Operation erfordert etwa 500 - 800 Maschinenbefehle
- > 10000 Transaktionen/s

TPF wird eingesetzt wenn es um besonders hohe Raten von relativ einfachen Transaktionen geht. Beispiele hierfür sind Flugplatzreservierung, Geldausgabeautomaten und Kreditkartenverifizierung.

TPF ist aus dem SABER (SABRE) Flugplatzreservierungssystem hervorgegangen, welches 1959 gemeinsam von American Airlines und IBM entwickelt wurde.. Später wurde es in ACP und dann in TPF umbenannt.

6.2 CICS-Programmierung

CICS-Anwendungen werden in Sprachen wie Cobol, C++, Java, PL/1 u.a. geschrieben. Zum Vergleich: SAP R/3-Anwendungen benutzen ausschließlich die proprietäre Programmiersprache ABAP/4.

Zur Ausführung von CICS-spezifischen Funktionen enthält das Anwendungsprogramm CICS- Statements. Ein CICS-Statement beginnt mit dem Ausdruck „EXEC CICS“ und endet mit einem Delimiter, der je nach Programmiersprache unterschiedlich ist.

Beispiel für ein CICS-Statement innerhalb eines Cobol-Programms:

```
EXEC CICS
WRITEQ TS QUEUE('ACCTLOG') FROM(ACCTDTLO)
LENGTH(DTL-LNG)
END EXEC
```

Ein existierender Datensatz "ACCTDTLO" wird in eine temporäre Warteschlange ACCTLOG geschrieben, die als Log zur Datensicherung dient

Beispiel für ein CICS-Statement innerhalb eines C++-Programms:

```
EXEC CICS SEND MAP("label04") MAPSET("s04set") ERASE;
```

Neue CICS-Anwendungen werden häufig in C oder Java geschrieben. Daneben hat Cobol nach wie vor eine überragende Bedeutung.

Cobol wird in akademischen Kreisen häufig als eine veraltete Programmiersprache angesehen, die schleunigst durch eine modernere ersetzt werden sollte. Die industrielle Praxis sieht anders aus. Die folgenden Tatsachen sind einem Sonderheft zum Thema "Cobol" der Zeitschrift "IEEE-Software" entnommen (Vol. 17, No.2, March/April 2000).

70-75 % aller ausgeführten OS/390 Transaktionen sind in Cobol geschrieben.

Die existierende Menge an Cobol-Programmen besteht aus etwa 180 Milliarden (10^9) Zeilen Cobol- Code, mit einer jährlichen Zuwachsrate von 5 Milliarden Lines of Code (LOC). Etwa 35 % von allen neuen Business-Anwendungen LOC werden derzeit in Cobol geschrieben. (Für weitere 35% wird Visual Basic benutzt, der Rest verteilt sich auf C++, Java, PL/1, RPG und andere).

Von 142 untersuchten Firmen und Organisationen setzen 87 % Cobol ein. Die Unternehmen investieren 50 % ihres Budgets in die Wartung vorhandener Cobol-Anwendungen, 20 % in die Entwicklung neuer Cobol-Anwendungen und 30 % in einer Mischung von beiden. Lediglich 25 % der Unternehmen haben Pläne für den Einsatz von Objekt-Cobol.

45 % der Unternehmen planen die Benutzung von Cobol auf dem derzeitigen Niveau für die nächsten 10 Jahre, 30 % gehen von einer schrumpfenden Bedeutung aus, 15 % wollen Cobol ganz ersetzen und 5 % gehen von einer wachsenden Bedeutung aus.

Die Jahr 2000 Umstellungskosten waren für Cobol-Programme deutlich günstiger als für Programme in anderen Sprachen (28\$ gegenüber 45\$ pro Funktion Point). Als Gründe dafür werden angegeben:

- Cobol ist eine sehr verbosede Programmiersprache und hat deshalb überlegene Dokumentationseigenschaften. Die meisten Computersprachen optimieren für ein Minimum an Keystrokes auf Kosten der Verständlichkeit.
- Cobol hat bessere Daten-Behandlungsmöglichkeiten als -C++ oder Java. Die Cobol-hierarchischen und -rekursiven Datendeskriptoren erlauben eine einfache Abbildung auf eine hierarchische Datenbank wie IMS, aber auch auf XML.

Eine weitere eindrucksvolle Erkenntnis ist:

- Cobol ist die einzige verbreitete Computersprache, die vorgelesen werden kann. Das menschliche Gehirn liest Texte, indem Worte in ihre Phoneme aufgebrochen werden. Dies erhöht die Verständlichkeit von Cobol.

Die OMG hat bei der Entwicklung des Corba-Standards frühzeitig eine IDL für Cobol definiert (zusätzlich zu C++ und Java).

6.3 CICS Struktur

6.3.1 Übersicht

Das OS/390-Betriebssystem besteht aus einem Kernel und aus einer Reihe von Prozessen, die innerhalb ihrer eigenen virtuellen Adressräume laufen. OS/390 bezeichnet virtuelle Adressräume als Regions. Diese Struktur ist in Abb. 6.3.1 wiedergegeben.

Manche Regions beherbergen Subsysteme (Systemprozesse), die Teil des Betriebssystems sind, aber im Benutzerstatus laufen. Einige der (zahlreichen) Subsysteme sind:

- CICS Transaktionsverarbeitung
- TSO Native Shell, Entwicklungsumgebung
- DB2 relationale Datenbank
- USS Unix System Services, Unix Shell
- WAS WebSphere Application Server

Es ist möglich, daß ein Prozess in Region A mit einem anderen Prozeß in Region B kommuniziert; diese Kommunikation wird aber vom OS/390-Betriebssystem streng kontrolliert. Eine Kommunikation zwischen Regions wird als "Interprocess Communication" (IPC) bezeichnet.

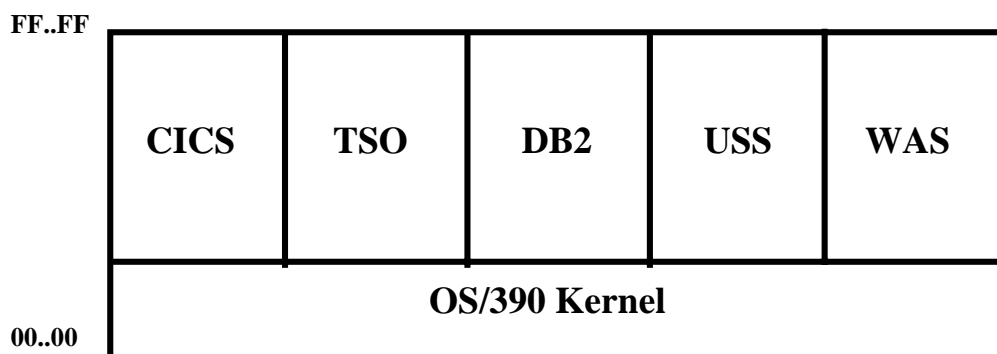


Abb. 6.3.1 : OS/390 Grundstruktur

CICS ist einer dieser Systemprozesse und läuft in einer eigenen CICS-Region. In der Mehrzahl der Fälle greifen die Transaktionen innerhalb einer CICS-Region auf Daten zu, die von einem Datenbankprozess in einer anderen Region verwaltet werden. In großen Installationen findet man häufig mehrere CICS-Regions. Die Transaktionen innerhalb dieser Regions kommunizieren über IPC- Verfahren miteinander.

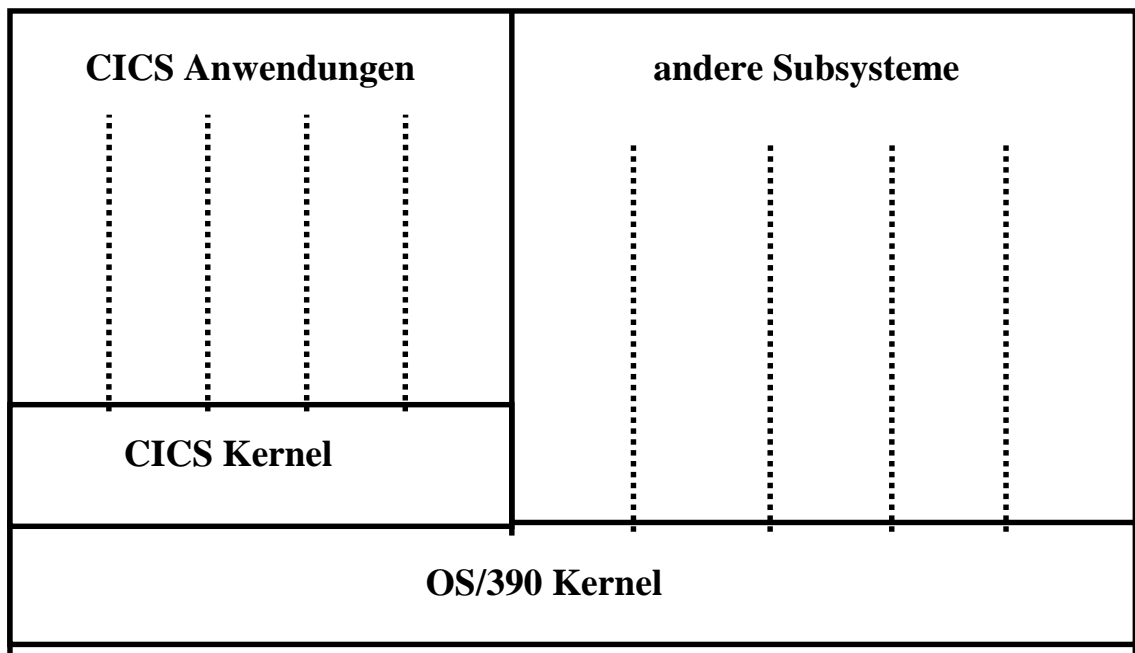


Abb. 6.3.2 : CICS-Grundstruktur

Abb. 6.3.2 zeigt die Grundstruktur einer CICS-Region. Entsprechend der Darstellung in Abb. 6.4.2 laufen der TP-Monitor (CICS-Kernel) und die Server-Prozesse (hier CICS-Anwendungen) innerhalb des gleichen virtuellen Adressraums (Region), während der Datenbankprozess typischerweise sich in einer separaten Region befindet. Alle CICS-Anwendungen (Transaktionen) und -Dienste (Resource Manager aus Abb. 6.4.4) laufen im Problemstatus, ungeschützt voneinander innerhalb eines einzigen virtuellen Adressraums: Die Ressourcen-Manager laufen als Threads innerhalb dieses Adressraums.

6.3.2 Struktur einer CICS-Transaktion

Wir unterscheiden zwischen einer CICS-Transaktion und einem CICS-Anwendungsprogramm. Eine Transaktion ist die Ausführung des Programms. CICS-Anwendungsprogramme sind reentrant geschrieben. Für mehrere Transaktionen, die gleichzeitig das gleiche Anwendungsprogramm benutzen, genügt eine einzige Kopie des Programms im Hauptspeicher.

Die Transaktion erzeugt eine Bildschirmausgabe am Terminal des Benutzers. In dem Beispiel von Abb. 6.3.3 sind drei Bildschirme einer Personaldatenabfrage mit identischer TRID aber unterschiedlichen Eingabedaten dargestellt.

Nachname	Müller	Nachname	Meier	Nachname	Schmitz
Vorname	Fritz	Vorname	Boris	Vorname	Stefan
Per. Nr.	12345	Per. Nr.	23456	Per. Nr.	34567
Straße	Ahornstr. 29	Straße	Marienstr. 72	Straße	Herdweg. 92
PLZ	70178	PLZ	72076	PLZ	71032
Wohnort	Stuttgart	Wohnort	Tübingen	Wohnort	Böblingen

Abb. 6.3.3 : Drei Bildschirmwiedergaben, welche die gleiche Map benutzen.

Die Bildschirmausgabe besteht aus einem festen und einem variablen Anteil. Der feste Anteil besteht aus der kursiv dargestellten Information (Adressinformation), wo welches Feld auf dem Bildschirm dargestellt wird und mehr. Dieser feste Anteil wird als "Map" bezeichnet und ist Bestandteil des CICS- Anwendungsprogramms. Während des Transaktionsablaufs wird die Map-Information durch Anfrage-spezifische Daten ergänzt, die typischerweise mit Hilfe eines Datenbankzugriffs gewonnen werden.

Für den Benutzer am Bildschirm besteht eine Transaktion in der Regel aus mehreren Schritten.

Der Benutzer ruft beispielsweise CICS auf, identifiziert sich und gibt eine TRID (einen Transaktionstyp) ein, trifft eine Auswahl zwischen mehreren Alternativen aus einem Auswahlmenu, und erhält schließlich eine Antwort. Während dieses Ablaufs erzeugt die Transaktion mehrere Bildschirmausgaben, siehe Abb. 6.3.4 .

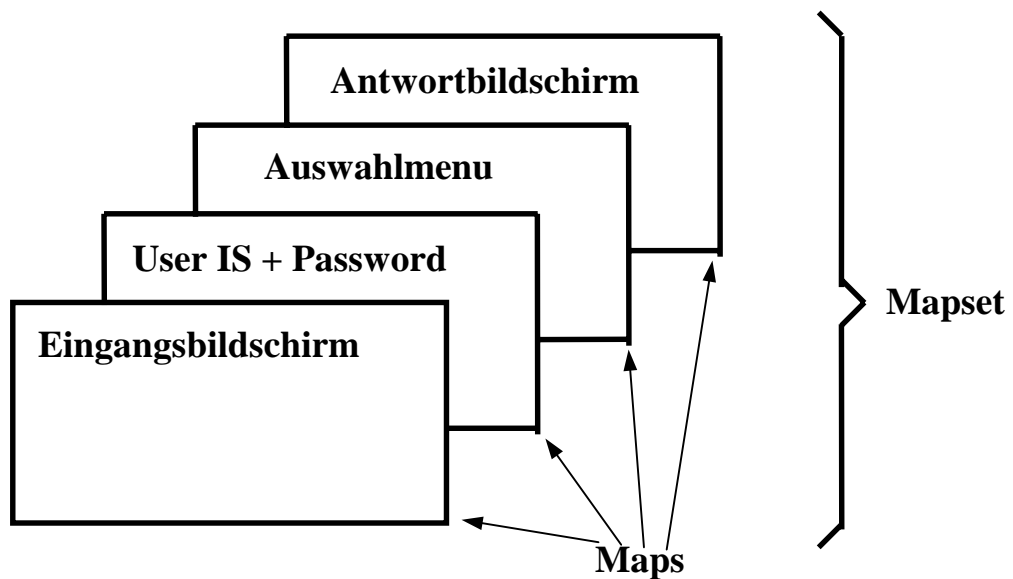


Abb. 6.3.4 : Mapset

Jede Bildschirmausgabe besteht aus einer Map plus Anfrage-spezifischen Daten.

Eine Map enthält ein Gerüst generischer Information. Dieses wird während der Transaktionsausführung mit spezifischer Information angereichert. Alle zu einer Transaktion gehörigen Maps werden als Mapset bezeichnet. Der Mapset ist Teil des CICS- Anwendungsprogramms.

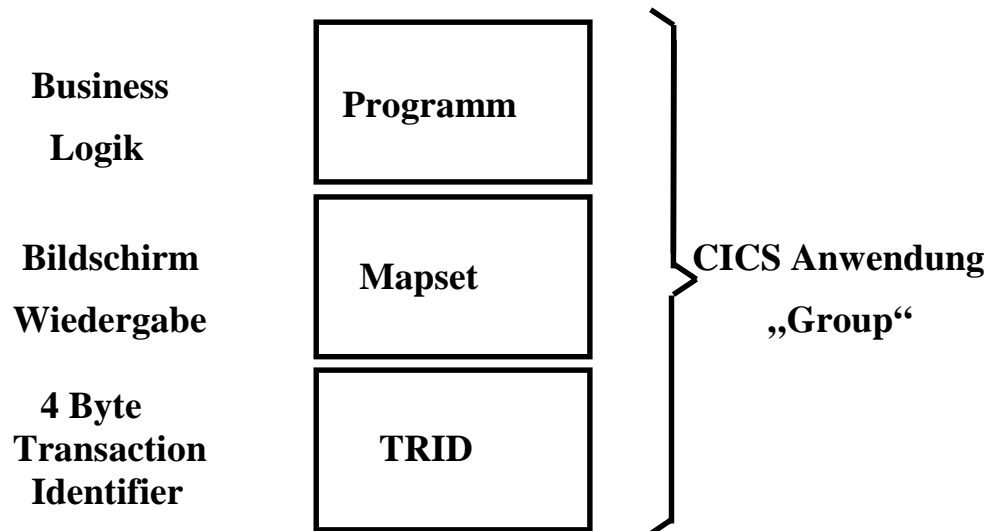


Abb. 6.3.5 : Struktur eines CICS-Anwendungsprogramms

Ein einfaches CICS Anwendungsprogramm besteht in der Regel aus drei Elementen: dem eigentlichen in Cobol, C++ oder Java geschriebenen Programm, dem Mapset, und einer Angabe, die beides mit einer spezifischen Transaktions-ID (TRID) verknüpft, siehe Abb. 6.3.5 . Diese drei Elemente werden zu einer "Group" zusammengefaßt und definieren eine lauffähige Transaktion.

In einer gegebenen CICS-Installation existiert in der Regel eine große, aber endliche Anzahl von CICS-Anwendungsprogrammen, die im Produktionsbetrieb genutzt werden. Jedes Anwendungsprogramm repräsentiert einen Transaktionstyp, der durch eine eindeutige TRID gekennzeichnet wird. Eine Transaktion wird durch Eingabe ihrer TRID aufgerufen. TRIDs werden vom Systemadministrator vergeben. Im Falle von CICS bestehen sie aus 4 alphanumerischen Zeichen. Manche TRIDs plus die entsprechenden Programme sind Bestandteil von CICS. Beispielsweise ist CEDA eine TRID, mit der eine interaktive CICS-Shell aufgerufen werden kann.

Eine CICS-Anwendung wird außerhalb von CICS entwickelt und anschließend in dem CICS-virtuellen Adressraum (CICS-Region) installiert. Der Installationsprozess erfolgt in 2 Schritten:

- Im Schritt 1 wird eine neue Gruppe erstellt. Mit Hilfe des CICS-CEDA-Kommandos "DEFINE" werden die Komponenten definiert, die zu der Gruppe gehören, also (mindestens) ein ausführungsfähiges Program, der Mapset sowie die TRID mit der die Transaktion aufgerufen werden soll.

In einem zweiten Schritt wird mit Hilfe des CEDA-"INSTALL"-Kommandos die neue Gruppe einschließlich aller Komponenten als Teil des CICS-Subsystems aktiviert. Die neue TRID wird in die TRID-Tabelle eingefügt und die erforderlichen einzelnen Referenzen werden aufgebaut.

6.3.3 Interne Struktur des CICS-Subsystems

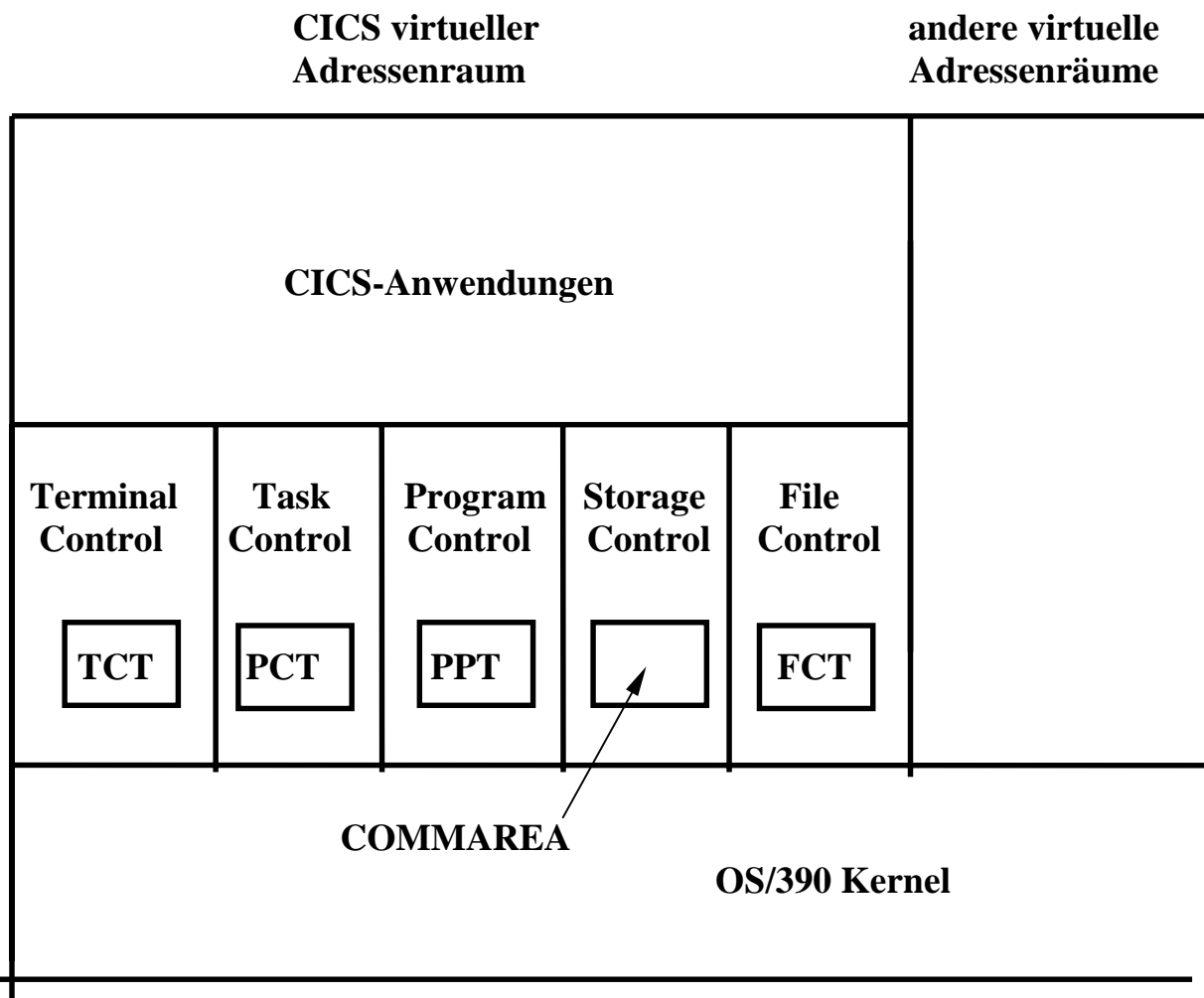


Abb. 6.3.6 : CICS-Komponenten

CICS läuft als Stapelverarbeitungs-Job in einem einzigen virtuellen Adressraum (Region in OS/390-Terminologie). CICS-Anwendungsprogramme laufen "run to completion"; Interaktivität wird programmtechnisch gewährleistet, indem ihre maximale Ausführungszeit eine vorgegebene Grenze nicht überschreitet.

Die CICS-Nucleus-Komponenten (Terminal Control, Task Control, Program Control, Storage Control and File Control) nutzen denselben virtuellen Adressraum wie alle Anwendungen. Jede Nucleus-Komponente hat eine zugeordnete Tabelle: TCT, PCT, PPT, FCT. Über COMMAREA werden Sessions eingerichtet: Der Zustand einer Transaktion ist für die Folgetransaktion verfügbar.

Die COMMAREA-Tabelle spielt eine besondere Rolle für CICS-Internet-Anwendungen und für Web-Transaktionen.

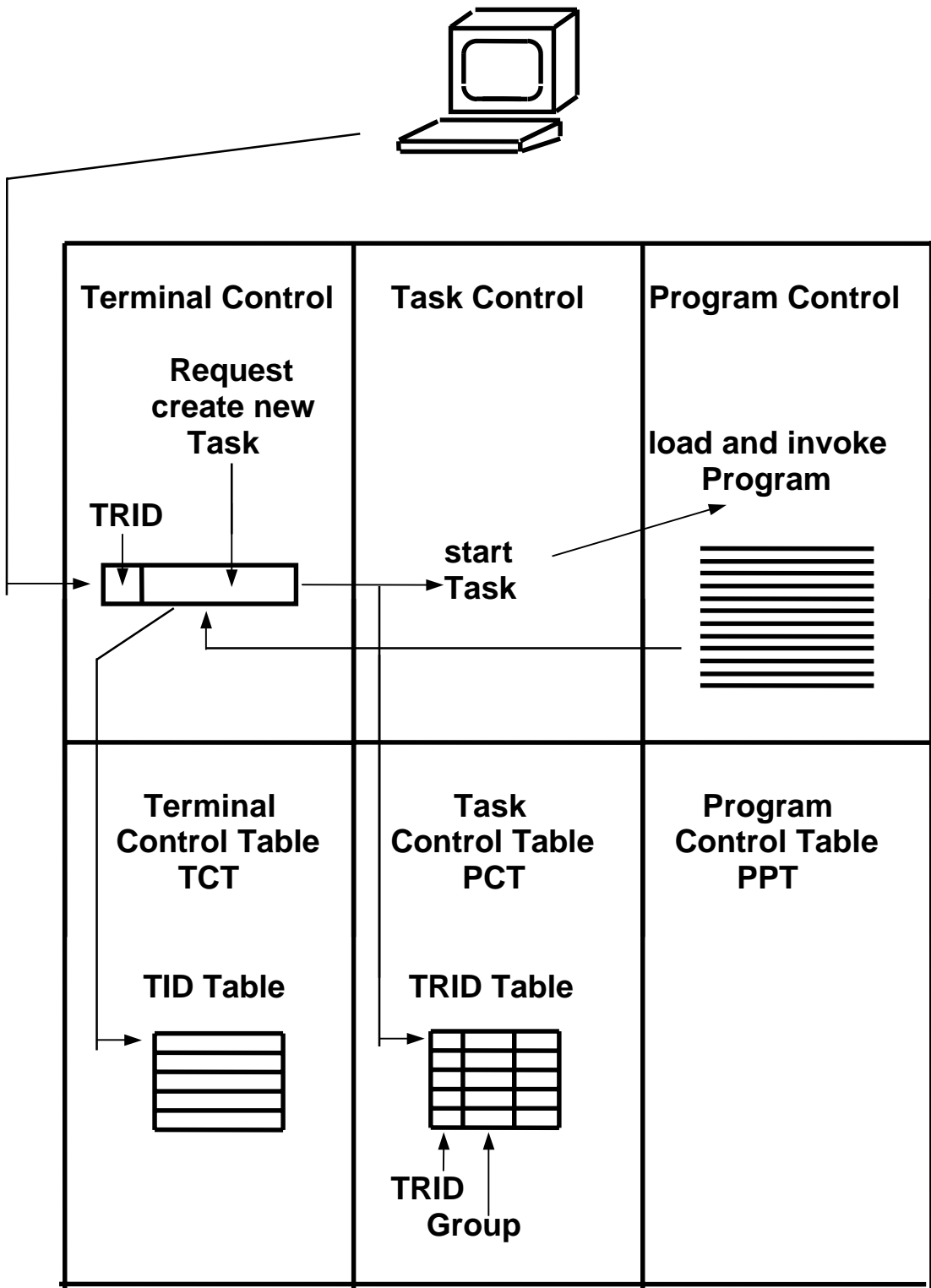


Abb. 6.3.7 Start einer CICS-Sitzung

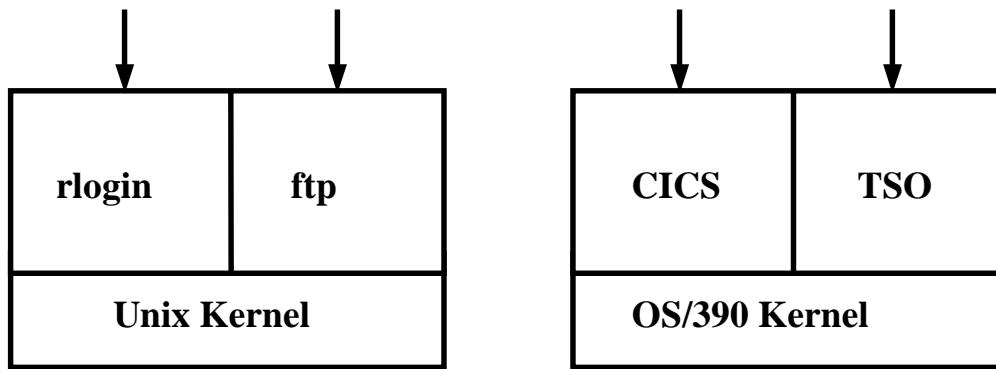


Abb. 6.3.8 : Aufbau einer Sitzung

Die folgenden Schritte beschreiben die Ausführung einer CICS-Transaktion. Siehe hierzu Abb. 6.3.7, die eine Erweiterung von Abb. 6.3.6 darstellt:

- Der Klient (Terminal) logged sich in das CICS-Subsystem ein. (Ähnlich wie bei Unix stehen unter OS/390 mehrere Subsysteme für ein remote-Login zur Verfügung, z.B. TSO, siehe Abb.3.7.8).
- Der Klient sendet eine Nachricht an CICS. Die Nachricht enthält die Adresse des Klienten.
- CICS Terminal Control prüft mit Hilfe des Terminal Identification Table (TID), ob eine Session mit dem Klienten besteht. Wenn nein, werden die ersten 4 Bytes der Nachricht als TRID interpretiert.
- Die Nachricht mit der TRID wird an CICS Task Control weitergegeben. Task Control liest aus seinem TRID Table die zu der TRID gehörige Group aus, darunter Referenz auf Mapset und Anwendungsprogramm.
- CICS Program Control lädt das Anwendungsprogramm in den Hauptspeicher (wenn nicht schon dort vorhanden), und ruft es auf. Das Anwendungsprogramm liest die Nachricht des Klienten.
- Wenn vom Klienten die nächste Nachricht eintrifft, erinnert sich CICS Terminal Control, daß eine Sitzung bereits besteht. Die Nachricht wird unmittelbar an CICS Task Control weitergereicht.

6.3.4 CICS Interprocess Communication (IPC)

Auf einem OS/390 Rechner können mehrere CICS-Regionen mit unabhängigen CICS-Monitoren installiert sein. Ein Rechnerverbund kann aus mehreren, unterschiedlichen Rechnern bestehen, z.B. Solaris, Linux, OS/390, Windows, usw. Auf allen ist CICS installiert.

Die unterschiedlichen CICS-Monitore können mit Hilfe der CICS Interprocess Communication (IPC)-Protokolle miteinander kommunizieren. Diese Protokolle funktionieren ähnlich wie die Socket- und RPC-Mechanismen in traditionellen TCP/IP Netzen. Die wichtigsten CICS IPC-Protokolle sind:

Transaction Routing

Transaktionen 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

Distributed Program Link

Ein Programm eines CICS-Systems kann mit Hilfe des Kommandos EXEC CICS LINK eine Verbindung mit einem Programm eines anderen CICS-Systems aufnehmen. Hiervon wird vor allem bei CICS-Internet-Anwendungen Gebrauch gemacht.

Mit Hilfe dieser Protokolle ist es möglich (und üblich), komplexe verteilte CICS-Netze aufzubauen, siehe als Beispiel Abb. 6.3.9.

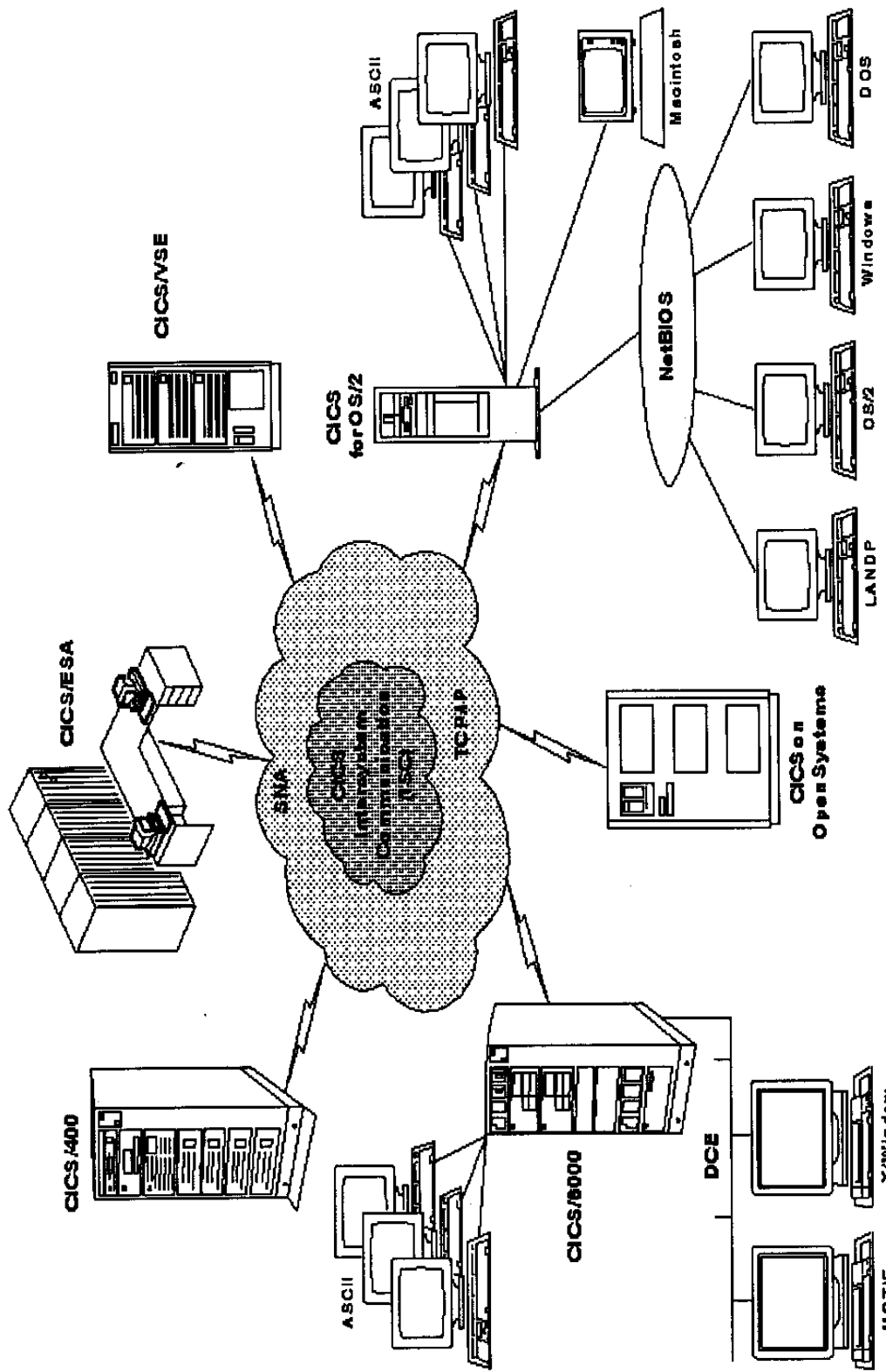


Abb.6.3.9 : Verteiltes CICS-Netz

6.4 BMS und das 3270 Übertragungsprotokoll

Ein CICS-System besteht aus einem oder mehreren CICS-Servern und einer großen Anzahl von Klienten. Viele Klienten fungieren als Arbeitsplatzrechner. Die Bildschirmausgabe kann aus einer Reihe von Zeilen mit alphanumerischen Text bestehen, ähnlich einer telnet- oder rlogin-Darstellung. In diesem Fall reden wir von einem "Character User Interface" (CUI) Prozess. Alternativ kann sie ein graphisches Aussehen in Anlehnung an eine Windows-, Motiv- oder KDE-Oberfläche haben. In diesem Fall reden wir von einem "Graphischen User Interface" (GUI)-Prozess. Spezifische TP- Monitore können ihre eigene GUI haben. Ein Beispiel ist die SAPGUI des SAP-System R/3, oder die Key GUI des UBS-KEY-ERP-Systems.

Neben den Arbeitsplatzrechnern existieren eine große Anzahl weiterer CICS-Klienten:

- Hand-Held-Geräte, z.B. Palmtops, Mobiltelefon
- Geldausgabeautomaten
- Kontoauszugsdrucker
- Supermarkt-Registrierkasse
- Tankstellen-Zapfsäule
- Produktionssteuerungselektronik

Die GUI oder CUI ist ein Prozeß in den Endgeräten, der für die visuelle Ein/Ausgabe zuständig ist. Je nach Art des Endgerätes können Umfang und Funktionalität sehr unterschiedlich sein. Es ist die Aufgabe der Präsentationsdienste, Information von/zu den Endgeräte-GUI's/CUI's in geeigneter Form aufzubereiten.

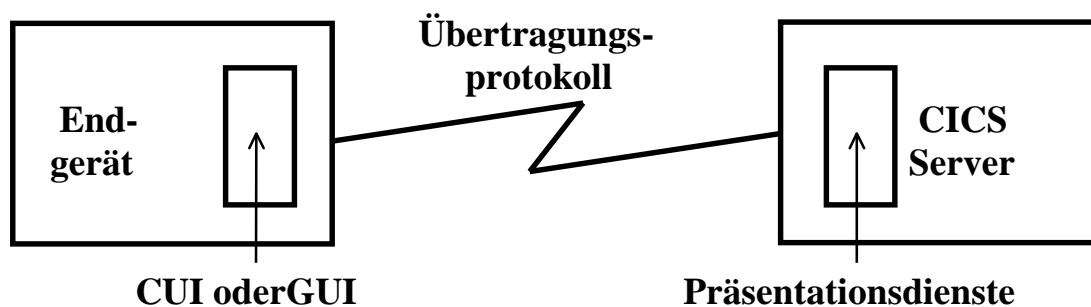


Abb. 6.4.1 : Endgeräte Anschluß

Abb. 6.4.2 zeigt Alternativen der CICS-BildschirmAusgabe.

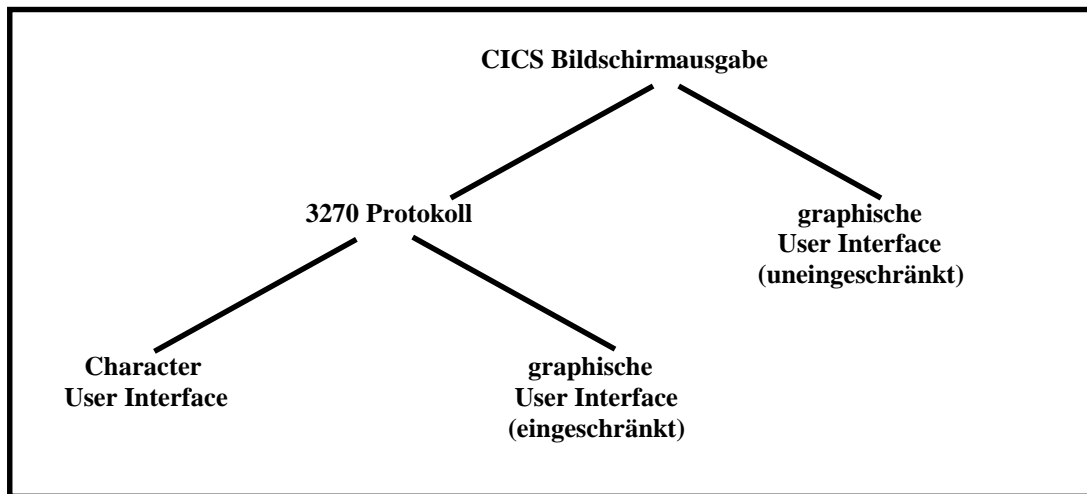


Abb. 6.4.2 : Alternativen der BildschirmAusgabe

Traditionell verwendet CICS das 3270-Protokoll und ein Character-User-Interface. Alternativ kann mit Hilfe des 3270-Protokolls ein graphisches User-Interface mit eingeschränkter Funktionalität eingesetzt werden. Bei Verzicht auf das 3270-Protokoll ist eine uneingeschränkte GUI möglich.

Die drei Alternativen werden der Reihe nach erläutert. Abschnitt 6.4 beschreibt das 3270-Protokoll, Abschnitt 6.5 die CUI. Die mit dem 3270-Protokoll ebenfalls mögliche GUI wird in Abschnitt 6.6 dargestellt.

Die letztere hat einige Einschränkungen. Beispielsweise ist es schwierig, auf dem Bildschirm eine Sliding-Window-Funktion zu implementieren. Um eine in der Funktionalität uneingeschränkte GUI zu realisieren, ist es möglich, die Terminal Control-Komponente des CICS-Kernels (Abb. 6.3.6) zu umgehen. Dies ist der modernste und zukunftsorientierste Ansatz. Es geschieht meistens unter Einsatz von Java mit Hilfe eines Web Application Servers und ist in Kapitel 9 dieses Buches beschrieben.

Fachleute mit einem kombinierte Wissen über Java, Web Application Server und CICS sind jedoch noch sehr rar. Deshalb wird heute in den meisten Fällen mit dem 3270-Protokoll gearbeitet. 3270- Anwendungen sind auch in vielen Fällen einfacher zu implementieren als dies bei einem Java/Web Application Server-Ansatz der Fall ist.

6.4.1 Das 3270-Protokoll

Das 3270-Protokoll ist die älteste, aber auch die am weitesten verbreitete Möglichkeit, CICS-Klienten an einen CICS-Server anzuschließen. Der Name rührt von der IBM 3270-Produktfamilie her, die etwa 1980 auf den Markt kam. Dies war das erste Mal, daß Bildschirmgeräte in größerem Umfang in den Unternehmen eingesetzt wurden.

Die 3278-Bildschirmgeräte erschienen zu einem Zeitpunkt, da es noch keine PCs gab. Sie bestehen aus einem Mikroprozessor, der einen CRT-Bildschirm ansteuert. Es können 24 Zeilen Text mit maximal 80 Zeichen pro Zeile dargestellt werden (siehe Abb. 6.4.3).

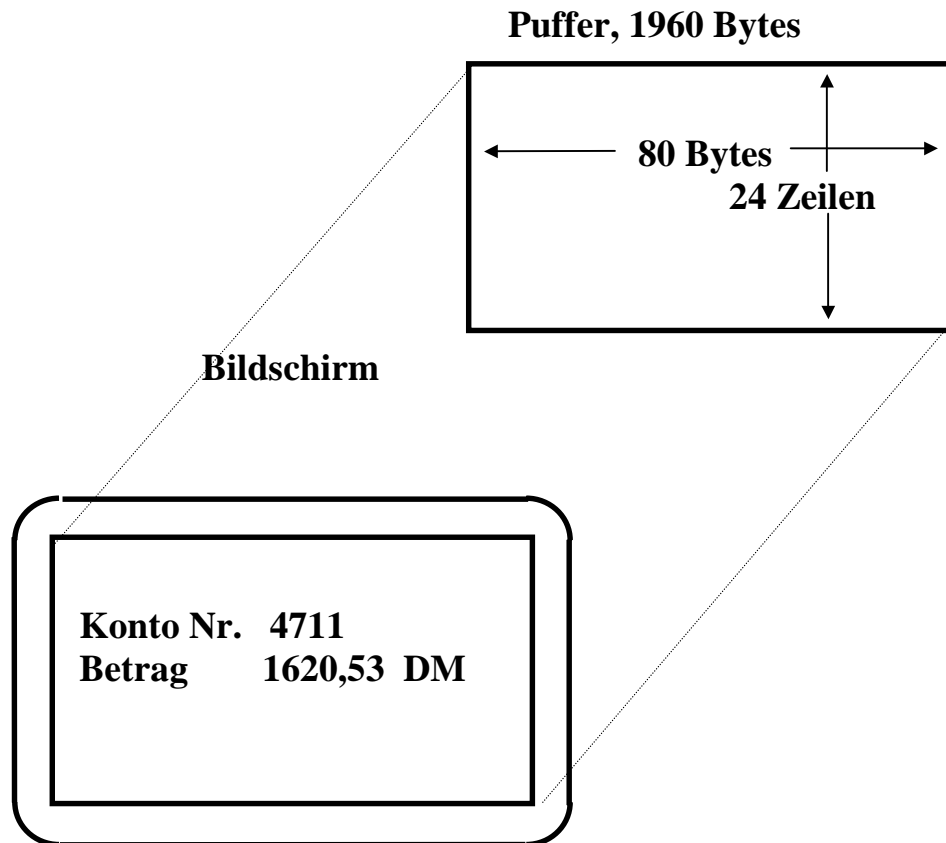


Abb. 6.4.3 : 3270-Bildschirmdarstellung

Die Informationsdarstellung erfolgt mit Hilfe eines 24 x 80 = 1960 Byte großen Bildschirm-Puffers. Es wird vom CICS-Server mit Hilfe des 3270-Protokolls eine Nachricht an das 3278-Terminal übertragen, die diesen Puffer mit Character-Daten füllt.

Diese werden von dem Mikroprozessor des 3278-Terminals interpretiert. Der Pufferinhalt wird als 24 Zeilen mit 80 Zeichen/Zeile auf dem Bildschirm wiedergegeben. Jede der 1960 Byte-Positionen kann einzeln adressiert werden. In der Regel werden Gruppen von Bytes zu Feldern zusammengefaßt und als solche adressiert.

In Abb. 6.4.3 sind 4 Felder dargestellt:

Konto Nr.
Betrag
4711
1620,53 DM

Die Adressen der Felder sind Teil der Map. Der Inhalt der Felder hat entweder einen Initialwert, der ebenfalls Teil der Map ist, oder er wird von dem CICS-Anwendungsprogramm bestimmt.

Abb. 6.4.4 zeigt ein Beispiel eines CICS Basic Mapping Support (BMS) α/n Bildschirms.

```

Col: 1234567890123456789012345678901234567890123456789012345678901234567890123
Row:
1  +ACCOUNT FILE:+RECORD DISPLAY
2
3  +ACCOUNT NO:+_____ +SURNAME: + _____|
4  +FIRST: + _____+ MI:+_+ TITLE:+_____|
5  +TELEPHONE:+_____ +ADDRESS: + _____|
6  + _____|
7  + _____|
8  +OTHERS WHO MAY CHARGE:
9  + _____| + _____|
10 + _____| + _____|
11
12 +NO. CARDS ISSUED:+_+ DATE ISSUED:+_+__+__+ REASON:+_+|
13 +CARD CODE:+_+| +APPROVED BY:+_+| +SPECIAL CODES:+_+__+__+|
14
15 +ACCOUNT STATUS:+_+ CHARGE LIMIT:+_____|
16
17 +HISTORY: BALANCE BILLED AMOUNT PAID AMOUNT
18 + _____ / / / / _____
19 + _____ / / / / _____
20 + _____ / / / / _____
21
22 + _____ (message area) _____

```

Abb. 6.4.4 : BMS-Beispiel

Das CICS-Anwendungsprogramm erzeugt die Datenausgabe an den Bildschirm in der Form eines seriellen Datenstroms. Dieser hat das in Abb. 6.4.5 gezeigte Format.

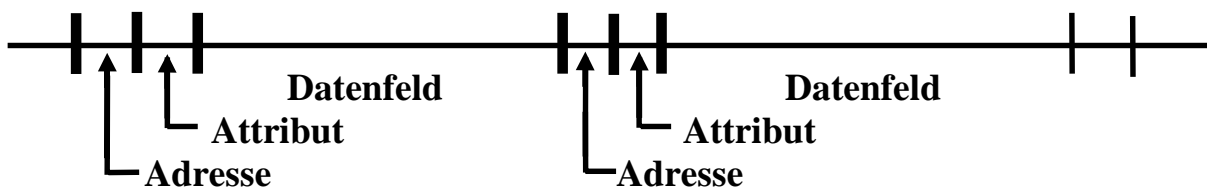


Abb. 6.4.5 : 3270-Bildschirm-Datenausgabe

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

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.

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

Der 3270-Bildschirm besteht aus 24 α/n -Zeilen mit je 80 fixed Font-Width-Zeichenpositionen pro Zeile. Das Adressenfeld kennzeichnet eine der $24 \times 80 = 1920$ Zeichenpositionen auf dem Bildschirm.

Der ursprüngliche 3278-Terminal verfügt über einen monochromen Bildschirm, der aufgrund der damaligen Technologie mit grünem Phosphor beschichtet ist. Es entsteht damit ein Screen mit grünen Buchstaben auf schwarzem Hintergrund. Ein "green screen" ist eine Bildschirmwiedergabe, die so wie auf dem ursprünglichen 3278-Terminal aussieht. Später entstanden weitere Modelle mit einem farbigen Bildschirm. Diese Modelle werden allgemein unter dem Begriff der 327x-Terminal-Familie zusammengefasst.

Das Attributfeld des 3270-Datenstroms (Abb. 6.4.5) gestattet es, Felder farbig darzustellen.

Das Programm des eingebauten 327x Mikroprozessors befindet sich in einem Read-only Speicher. Das 3270-Terminal ist ein "unintelligentes Terminal", es kann nicht programmiert werden. Heute werden bevorzugt PCs als CICS Klienten eingesetzt. Auf dem PC läuft eine als "3270 Emulator" bezeichnete Komponente, welche die ursprüngliche 3278-CUI-Funktionalität in einem Windows- oder Browser-Fenster emuliert.

Weit mehr als ein Dutzend Firmen vertreiben 3270-Emulatoren. Die Mehrzahl (aber nicht alle) verwenden das Telnet-Protokoll nur für die erste Verbindungsaufnahme. Danach wird ein Java-Applet geladen; die gesamte weitere Kommunikation findet innerhalb eines Web-Browsers statt. Das 3270-Protokoll setzt dann auf dem HTTP-Protokoll auf. Das Applet wird entweder lokal gespeichert, oder bei Bedarf heruntergeladen.

Von der Firma IBM werden u.a. die Produkte "Personal Communication" und "Host on Demand" angeboten.

Von den etwa 30 Millionen CICS-Endgeräten sind heute (2001) noch knapp 10 Millionen 327x-Terminals. Der Rest sind PCs mit einem 3270-Emulator. Trotz der geringeren Funktionalität sind 3270-Terminals nach wie vor populär, weil auf Grund der nicht erforderlichen Softwarewartung die laufenden Betriebskosten deutlich geringer als bei einem 3270-PC-Emulator sind.

Das 3270-Protokoll gilt als ein sehr effektives Protokoll und wird deshalb in modifizierter Form auch außerhalb der OS/390-Umgebung häufig eingesetzt. Beispielsweise benutzt die SAP-R/3-GUI ein modifiziertes 3270-Protokoll.

6.4.2 Basic Mapping Support

Die CICS-Bildschirmdarstellung besteht aus 2 Komponenten: Map und Inhalt.

Eine Map definiert die Wiedergabe von Daten auf dem Bildschirm. Jede Map beschreibt das Aussehen eines Bildschirm-Fensters des Klienten, spezifisch die Anordnung und Art von Feldern. Je nach Transaktionsart wird die dazugehörige Map in den Klienten geladen. Die Map wird mit Ausgabedaten (Inhalt) gefüllt, die das CICS Anwendungsprogramm erstellt. Ein Screen ist eine Map, die vom CICS Anwendungsprogramm mit dynamischen Daten gefüllt worden ist. Jeder Klient (CICS-Terminal) enthält ein CUI Programm, das die Daten auf dem Bildschirm ausgibt.

Bei einer CUI (Character User Interface) definiert die Map:

Feld Nr. 1 an Ort aaa, Feld 2 an Ort bbb
wobei aaa, bbb Adressen des 24 x 80 Bildschirm-Grids sind

Bei einer GUI (Graphical User Interface) redefiniert der GUI-Prozess innerhalb eines PC zusätzlich Aussehen und Anordnung der 3270-Datenelemente

BMS (Basic Mapping Support) ist eine Komponente von CICS. Sie erzeugt und verwaltet Maps (siehe Abb. 6.4.6).

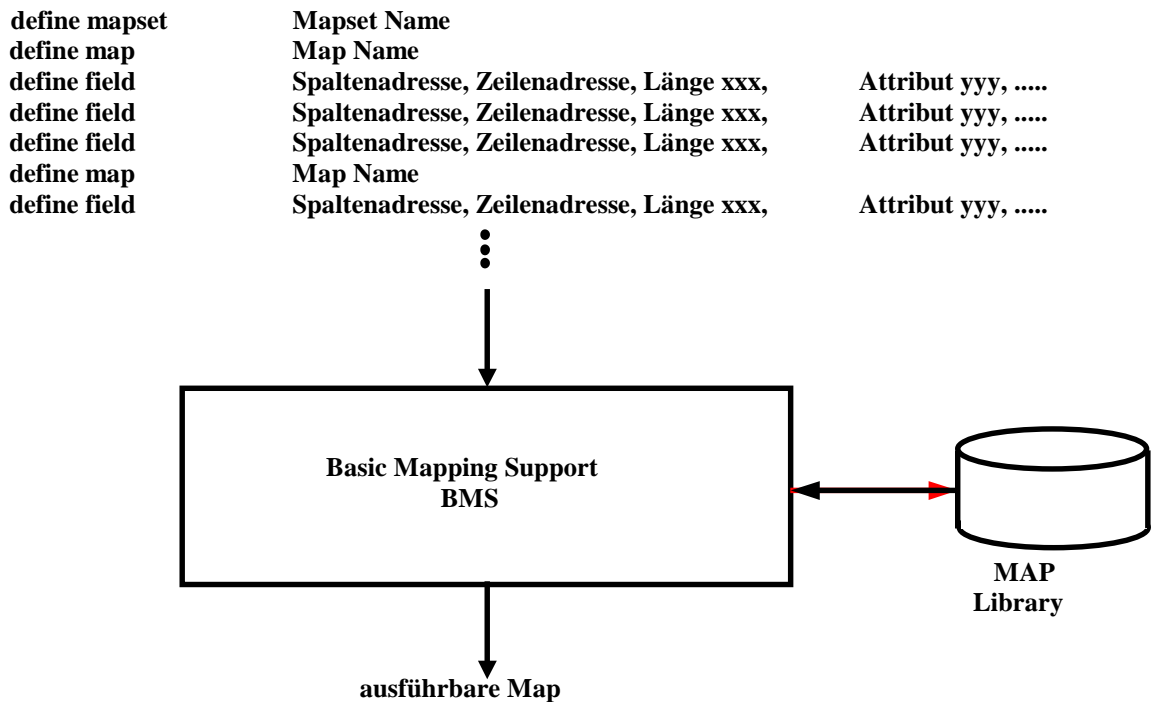


Abb. 6.4.6 : Basic Mapping Support

Während das CICS-Anwendungsprogramm in einer normalen Programmiersprache wie C++, Cobol, Java usw. programmiert wird, erstellt man den Mapset mit Hilfe einer eigenen Map-Definitionssprache. Letztere besteht im Wesentlichen aus drei Befehlstypen: define mapset, define map und define field. Die Bezeichnungen der drei Befehlstypen sind etwas ungewöhnlich, nämlich:

define mapset	DFHMSD
define map	DFHMDI
define field	DFHMDF

(Der Grund besteht darin, daß als Übersetzer für die Map-Definitionssprache in Wirklichkeit der OS/390-Assembler eingesetzt wird; DFHMSD, DFHMDI und DFHMDF sind Assembler Makros.)

```

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 LABEL04 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 //

```

Abb. 6.4.7 : Beispiel einer sehr einfachen Map-Definition

Das in der Abb. 6.4.7 dargestellte Beispiel stellt eine sehr einfache Map-Definition dar. Die eigentliche Definition ist in den Zeilen 000004 bis 000012 enthalten. Zeile 000004 definiert einen Mapset mit dem Namen S04SET. Der Mapset enthält eine einzige Map. Zeile 000005 definiert die Map mit dem Namen LABEL04. Die Map enthält zwei Felder, die jeweils in den Zeilen 000007 und 000008 bzw. 000009 und 000010 untergebracht

sind. Die beiden Felder werden mit den Werten "Welcome to the Magic World of CICS" bzw. „May the Force be with you“ initialisiert.

Zeilen 000001 bis 000003, sowie Zeilen 000013 und 000014 stellen ein Script dar, das die Mapset- Definition übersetzt und in eine Map-Library stellt. Die Übersetzung erfolgt spezifisch durch die Routine DFHMAPS in Zeile 000002.

```
struct adresse {.....  
        .....  
        };  
  
main()  
{  
EXEC CICS SEND MAP(„s04set“) MAPSET(„label04“);  
}
```

Abb. 6.4.8 Ein einfaches CICS-Anwendungsprogramm

Ein einfaches CICS-Anwendungsprogramm (in C), das die in Abb. 6.4.7 dargestellte Map benutzt, ist in Abb. 6.4.8 wiedergegeben.

Der normale Vorgang besteht darin, die Map und das Anwendungsprogramm zu übersetzen. Die Entwicklung muß in einer anderen Region als der CICS-Region erfolgen. Dies könnte z.B. eine TSO Region auf demgleichen Rechner sein; in vielen Fällen wird für die Entwicklung neuer CICS- Anwendungen ein separater Rechner benutzt.

Anschließend müssen Anwendungsprogramm und Map in der CICS-Region installiert werden. Gleichzeitig muß die neue Transaktion mit einer 4-stelligen TRID verknüpft werden.

Wird die Transaktion über ihre TRID aufgerufen, erscheint der in Abb. 6.4.9 dargestellte Screen.

```
WELCOME TO THE MAGIC WORLD OF CICS  
  
MAY THE FORCE BE WITH YOU!
```

Abb. 6.4.9 : Bildschirmwidergabe aus Abb. 6.4.7 und Abb. 6.4.8

6.5 Terminal Anschlußmöglichkeiten

6.5.1 Traditionelle Anschlüsse

Das 3270-Protokoll entstand vor der Entwicklung der Ethernet- und Tokenring- Technologie. Die ursprüngliche Konfiguration ist in Abb. 6.5.1 wiedergegeben.

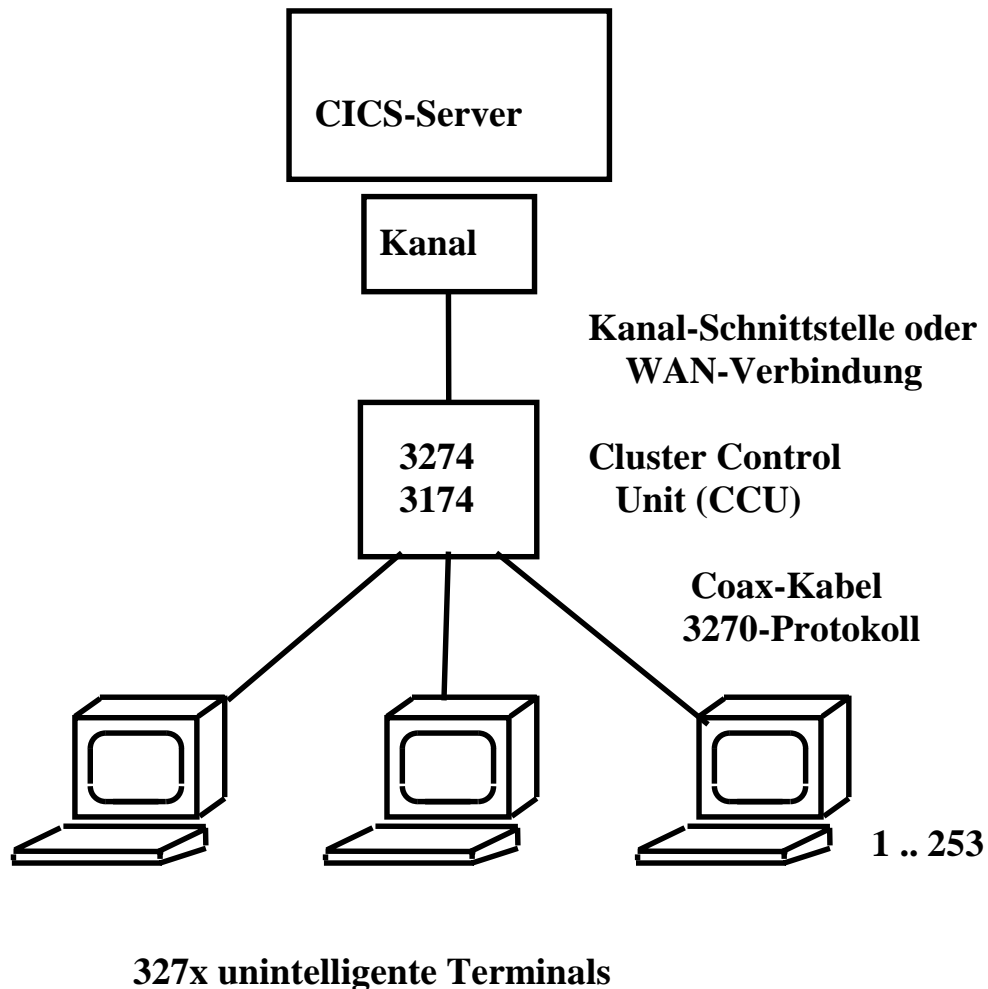


Abb. 6.5.1 : Ursprüngliche 327x-Konfiguration

Der Anschluss der 327x-Terminals erfolgte mit einer sternförmigen Koax-Kabel-Verdrahtung; Eine Steuereinheit, die 3274 Cluster Control Unit (CCU), schloss bis zu 253 Terminals an. Die CCU selbst wurde lokal an den S/370-Rechner angeschlossen.

Später entstanden Anschlußmöglichkeiten, welche die Koax-Kabel-Verdrahtung durch Tokenring- und Ethernet-LANs ersetzen. Außerdem wurde ein Modem-Anschluß für eine Verbindung über das Telephonnetz geschaffen. Dabei bewährte sich das 3270-Protokoll wegen seiner sehr effektiven Nutzung der begrenzten Bandbreite.

Ebenfalls existierte zum Zeitpunkt der 3270-Einführung das TCP/IP-Protokoll noch nicht. CICS und das 3278-Terminal verwenden das SNA-Protokoll. Die Abbildung 6.5.2 zeigt das Mapping des 3270- auf die Schichten des SNA-Protokolls.

Schicht

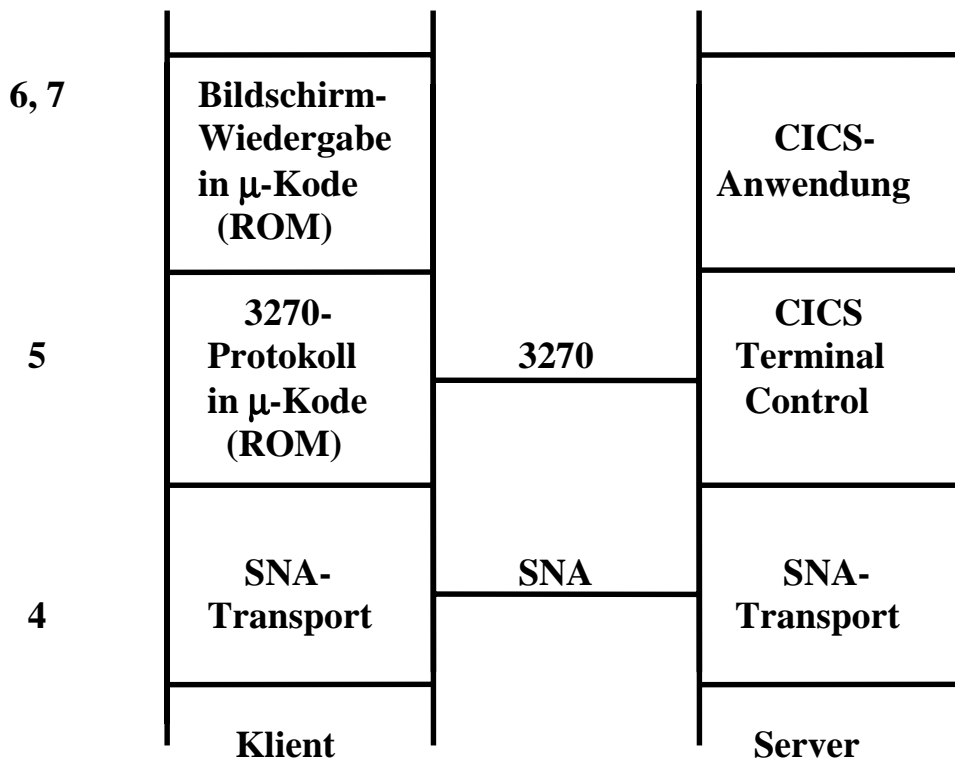


Abb. 6.5.2 : Abbildung des 3270-Protokolls auf SNA

SNA ist ein Netzwerkmodell, das ähnlich wie TCP/IP arbeitet und wie OSI auf einer 7-Schichten-Architektur basiert. SNA hat einen vergleichbaren Funktionsumfang wie TCP/IP und OSI. TCP/IP ist heute deutlich weiter verbreitet als SNA. Die vorhandenen SNA-Installationen werden nach und nach auf TCP/IP umgestellt. Die Umstellungen erfolgen jedoch relativ langsam. SNA ist in größeren Unternehmen jedoch nach wie vor häufig zu finden; einige der SNA-Funktionen sind auch heute unter TCP/IP noch nicht verfügbar.

Die Terminal Control-Komponente des CICS-Kernels (Abb. 6.3.6) ist eng mit SNA verzahnt. Es ist möglich, die Terminal Control-Komponente zu umgehen; sie wird jedoch von BMS, und auch fast allen CICS-Anwendungen, die nicht BMS benutzen, verwendet. Sind die Terminals über ein SNA-Netz verbunden, ist das kein Problem. Erfolgt der Anschluß über ein TCP/IP-Netz, sind spezielle Maßnahmen erforderlich.

6.5.2 SNA

Auf einer konzeptuellen Ebene sind TCP/IP und SNA einander ähnlich. Beide verwenden ein sauber strukturiertes Schichtenmodell. Verwirrend sind vor allem die unterschiedlichen Bezeichnungen.

Router in einem SNA-Netzwerk werden LU's (Logical Units) genannt. Der derzeit gängige Typ wird als LU 6.2 bezeichnet. Der ältere Typ LU 2 wurde ursprünglich für nicht-intelligente Terminals entwickelt und eignet sich nur für die Kommunikation zwischen einem Arbeitsplatz-Rechner und einem zentralen Host. CICS-3270-Sessions verwenden die LU2.

Die Hardware und Software eines Knotenrechners wird als PU (Physical Unit) bezeichnet. Eine PU steuert die Verbindungsleitungen zu einer anderen PU. Der derzeit gängige Typ wird als PU 2.1 bezeichnet. Eine PU kann eine oder mehrere LU's unterstützen; die letzteren verwalten vor allem die "Sessions".

(Eine PU 4 ist direkt an einen /390-Rechner (über einen „Kanal“) angeschlossen und hat erweiterte Funktionen. Die auf einem /390-Rechner selbst ablaufende Netzsoftware, z.B. "VTAM“, wird als PU 5 bezeichnet).

Eine SNA-"Session" ist eine zuverlässige virtuelle Verbindung zwischen zwei LU's. Mehrere parallele Sessions zwischen zwei LU's sind möglich.

Eine "APPC Conversation" ist eine (von vielen) Transaktionen, die - unter Verwendung des APPC-API - über eine Session mit einer anderen Transaktion kommuniziert. CICS-IPC's können über APPC durchgeführt werden. In der Regel laufen viele Transaktionen der Reihe nach über eine Session.

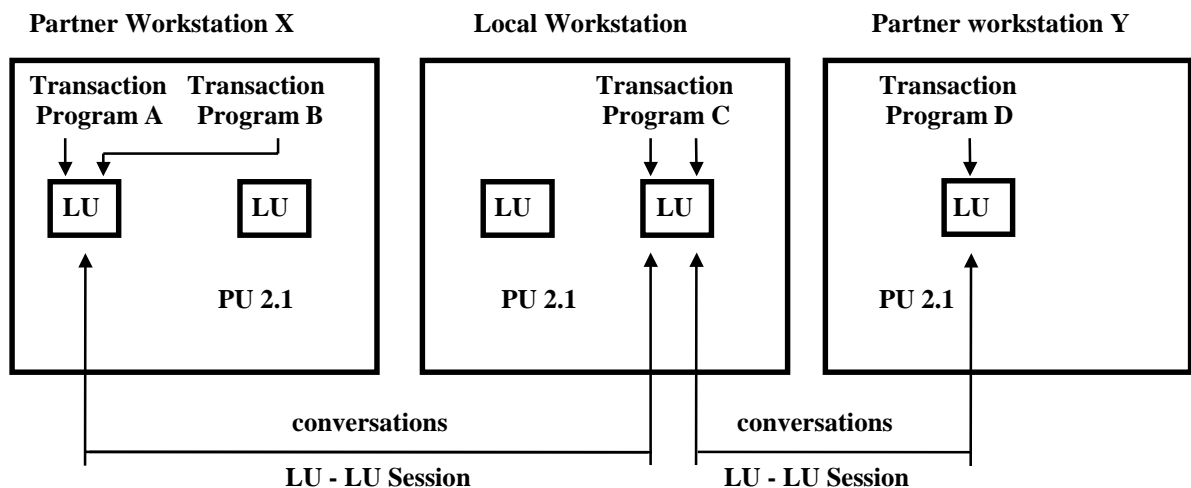


Abb. 6.5.3 : Transaktionsverarbeitung über ein SNA-Netz

Abb. 6.5.3 zeigt ein Beispiel für eine Transaktionsverarbeitung über ein SNA-Netz. Dargestellt ist eine lokale LU, die mit 2 Partner-LU's kommuniziert. Der Zugriff zu den LU's erfolgt über die APPC-API

TCP und UDP stellen TSAP's in der Form von Sockets zur Verfügung. Mit Hilfe der Socketschnittstelle können 2 Anwendungen auf 2 Rechnern miteinander kommunizieren.

LU 6.2 stellt ähnliche TSAP's zur Verfügung. Mit Hilfe der APPC-Schnittstelle können zwei Anwendungen auf zwei Rechnern miteinander kommunizieren

"SNA-Sessions"-äquivalente Einrichtungen sind unter RPC und Named Pipes verfügbar

6.5.3 Nutzung von PCs als CICS-Terminals

Der moderne Ansatz ist, unintelligente, nicht programmierbare 327x-artige Terminals durch PCs zu ersetzen.

Um mit Hilfe eines Arbeitsplatzrechners auf einen Server zugreifen zu können, braucht man eine Zugriffskomponente, die allgemein als Client bezeichnet wird. Einige Beispiele sind:

Arbeitsplatzrechner Client	Übertragungsprotokoll	Komponente auf dem zentraler Server
Browser	HTTP	Web Server, z.B. Apache
FTP Client	FTP	FTP Server
Telnet Client	Telnet	Telnet Server
3278 Terminal	3270 über SNA	CICS Terminal Control

OS/390 unterstützt all diese Möglichkeiten. Am weitesten gebräuchlich ist jedoch die Benutzung eines 3270-Klienten, der über das 3270-Übertragungsprotokoll mit der OS/390 "Communication Server"- Komponente des OS/390-Kernels kommuniziert.

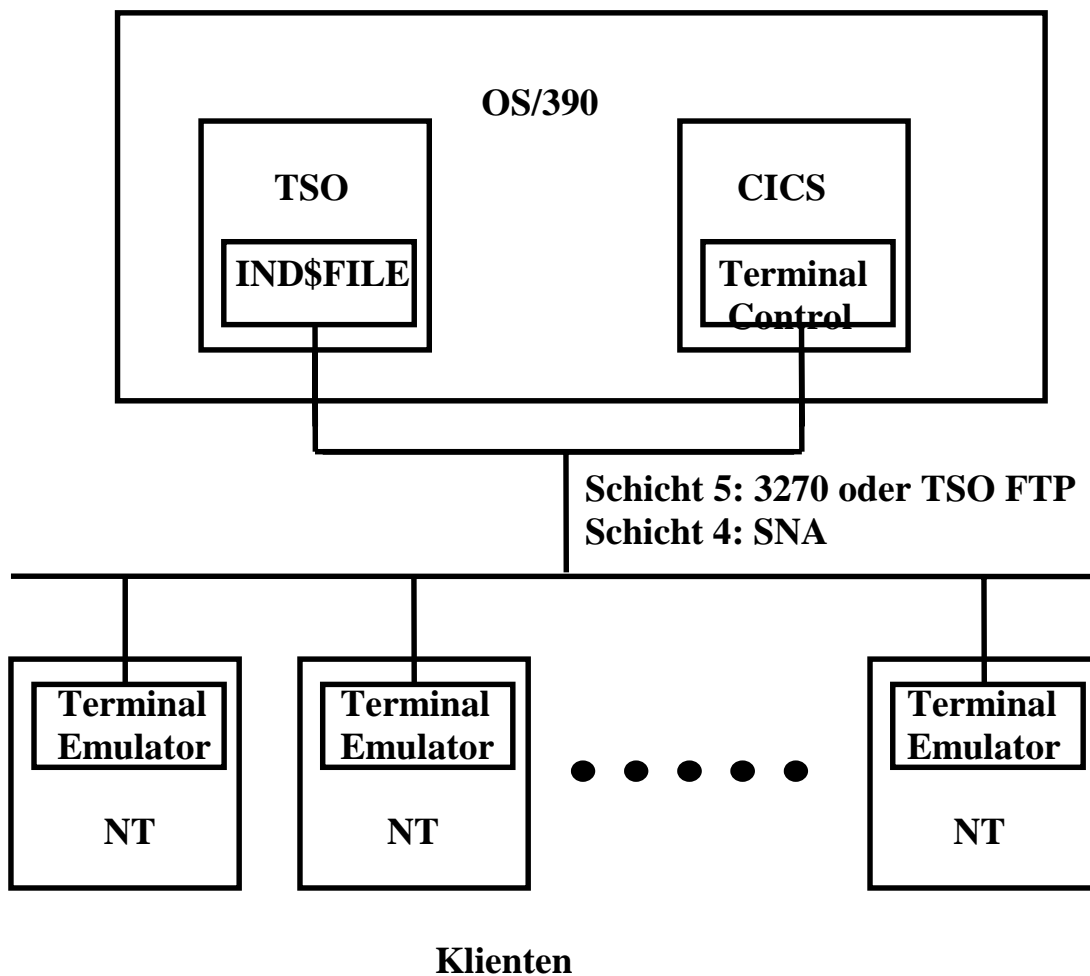


Abb. 6.5.4 : CICS-Client/Server-Kommunikation über ein SNA-Netz

Während ein 327x-Terminal grundsätzlich über SNA angeschlossen wird, gibt es für einen PC als CICS-Klient 2 Alternativen: Anschluß über ein SNA-Netz oder Anschluß über TCP/IP. Abb. 6.5.4 zeigt eine CICS-Client/Server-Kommunikation über ein SNA-Netz.

Der Terminal-Emulator ist eine Anwendung, die unter dem PC-Betriebssystem läuft (beispielsweise NT). Sie besteht aus 2 Komponenten:

- 3270-Emulator, der den 3270-Datenstrom interpretiert und die CUI-Funktion eines 3278-Terminals übernimmt
- SNA -Stack, über den ein Windows-Betriebssystem normalerweise nicht verfügt.

Beispielsweise läuft der Terminal-Emulator in einem Fenster des Windows-Betriebssystems. Innerhalb dieses Fensters wird eine 327x-Funktionalität erzeugt. Für einige Sonderfälle sind spezielle Maßnahmen erforderlich; z.B. fehlen dem PC einige Funktionen der 3270-Tastatur.

Vielfach wird der Terminal-Emulator mit weiterer Funktionalität angereichert. Das OS/390-TSO- Subsystem verfügt über eine FTP-vergleichbare Funktionalität, die auf den Namen "IND\$FILE" hört. Vielfach wird der Terminal-Emulator um einen IND\$FILE-Klienten erweitert.

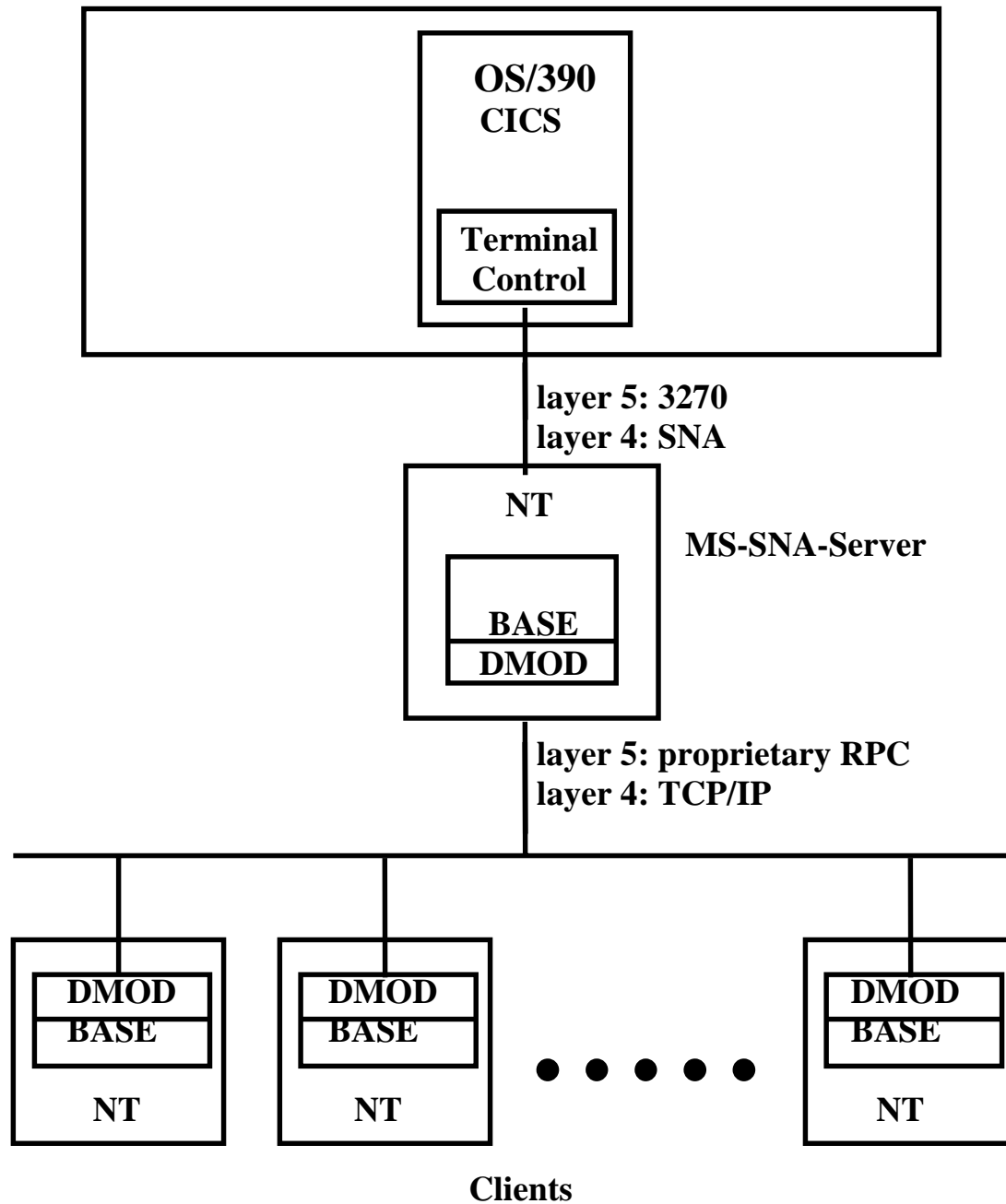


Abb. 6.5.5 : CICS-Client/Server-Kommunikation über einen SNA-Server

Wie bereits erwähnt, arbeitet die Terminal Control-Komponente des CICS-Kernels (Abb. 6.3.6) mit SNA. In den letzten Jahren stellen jedoch mehr und mehr Unternehmen ihre hauseigenen SNA-Netze auf TCP/IP um.

Abb. 6.5.5 zeigt den Anschluß von PC-basierten CICS-Klienten über ein TCP/IP Netz. In diesem Beispiel wird hierfür der SNA-Server der Fa. Microsoft verwendet. Die CICS-Klienten kommunizieren mit dem SNA-Server über ein TCP/IP-Netz unter Einsatz eines Microsoft-proprietären Protokolls. Der SNA-Server ist direkt an den OS/390-Rechner angeschlossen. Er übersetzt zentral das Microsoft- Protokoll in das SNA-Protokoll. Zahlreiche Startup-Unternehmen haben innovative Produkte für den 30 Millionen großen CICS- Terminal-Markt entwickelt. Mehr als ein Dutzend Firmen vertreiben 3270-Emulatoren.

Die Mehrzahl der TCP/IP-basierten 3270-Emulatoren verwenden das TN3270-Protokoll. Dieses transportiert einen 3270-Datenstrom über das Telnet-Protokoll. Auf der OS/390-Seite wird ein TN3270- Server benötigt, der

Nachrichten über den TCP/IP-Port 23 (Telnet Port) entgegennimmt und in ein SNA-Format umsetzt. OS/390 verfügt über das "Communications Server" Subsystem, das diese Funktionalität enthält.

Einige der 3270-Emulatoren verwenden das Telnet-Protokoll nur für die erste Verbindungsaufnahme. Danach wird ein Java-Applet geladen; die weitere Kommunikation findet innerhalb eines Applet-fähigen Web Browsers (z.B. Netscape Version 4.7) statt.

6.6 Präsentationslogik

6.6.1 Business- und Präsentationslogik

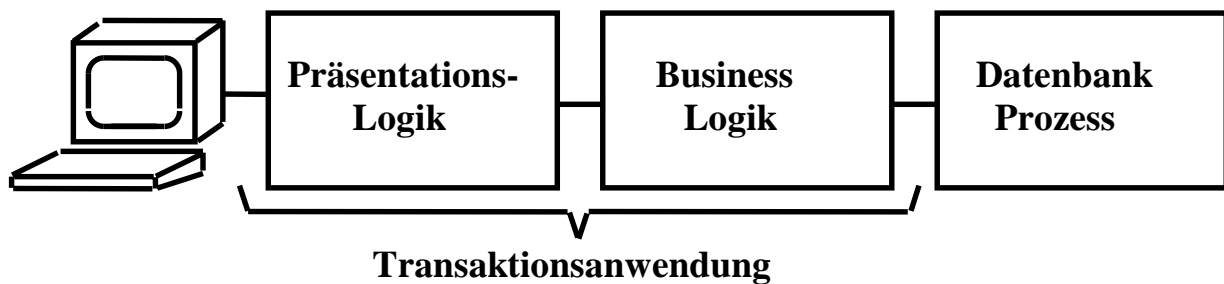


Abb. 6.6.1 : Business- und Präsentationslogik

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 usw. geschrieben. Für die Präsentations- Logik gibt es viele Möglichkeiten.

Die modernste Alternative benutzt 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-Maps werden in der BMS-Sprache erstellt.

Standardmäßig erzeugt der 3270-Datenstrom auf dem Bildschirm den berühmten "green Screen". Es existiert noch ein weiteres Verfahren, die gleiche Information auf eine graphisch ansprechende Art darzustellen. Dieses Verfahren wird als "Screen Scraping" bezeichnet.

Dies sei an Hand des in Abbildung 6.6.2 wiedergegebenen Beispiels eines Computer-Schachprogramms erläutert (übersehen wir, daß Computerschach nicht gerade eine typische Transaktionssystem- Anwendung darstellt).

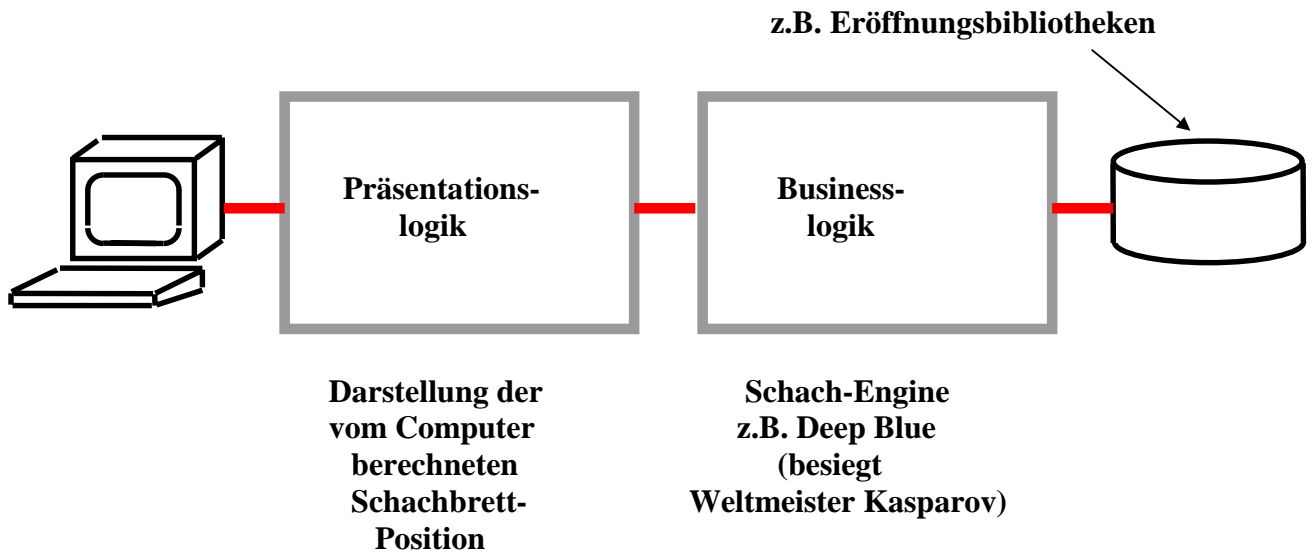


Abb. 6.6.2 : Beispiel eines Computer-Schachprogramm

Dargestellt ist die Gliederung des Programms in Präsentationslogik und Businesslogik. Wenn der Computer mit der Berechnung eines Zuges fertig ist, kann eine CUI-Ausgabe in der in Abb. 6.6.3 dargestellten Form erfolgen.

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. 6.6.3 : Minimale Präsentations-Logik

Für einen erfahrenen Schachspieler ist die hier wiedergegebene Beschreibung eines Schachbretts eindeutig.

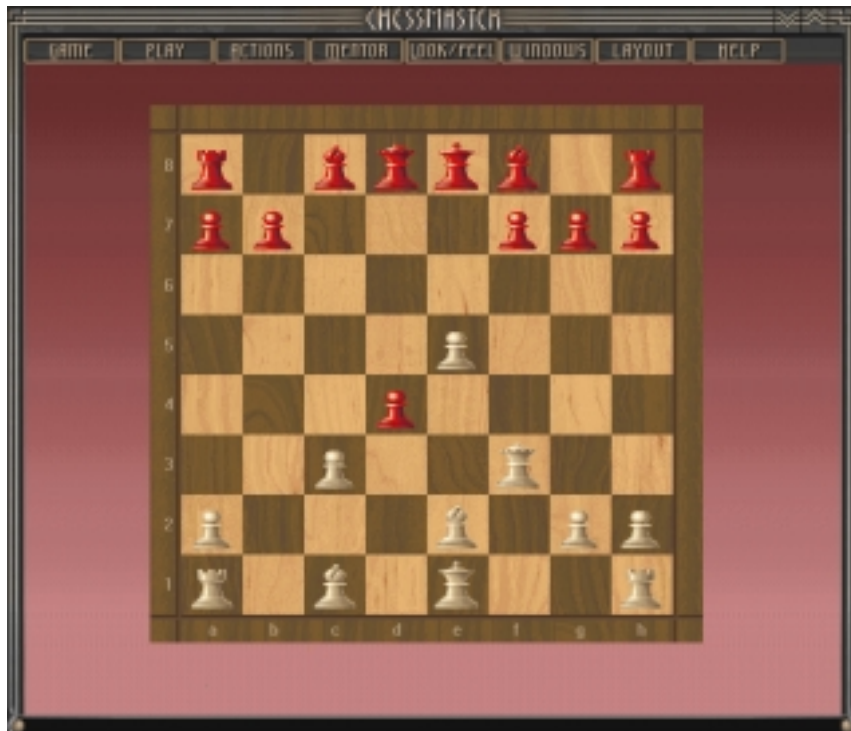


Abb. 6.6.4 : Graphische Präsentations-Logik

Abb. 6.6.4 ist eine GUI-Darstellung der in Abb. 6.6.3 wiedergegebenen CUI. Diese Darstellung wird als "Screen" bezeichnet. Wichtig ist, daß Abb.6.6.4 keine Variablen-Information enthält, die nicht bereits in Abb. 6.6.3 enthalten ist. Der GUI-Prozess des Schachspiel-Klienten ist so strukturiert, daß er aus der in Abb. 6.6.3 dargestellten Information die Graphik von Abb. 6.6.4 erzeugt.

Dies gilt auch für Elemente wie das Feld "Mein Zug: c5 x d4 ", das in Abb. 6.6.4 durch ein Blinken des Bauern auf dem Feld d4 dargestellt wird, oder die Eingabe des nächsten Zuges, die durch ein Ziehen mit dem Mauszeiger bewerkstelligt wird.

Zur Erinnerung: Der 3270-Datenstrom stellt das Ergebnis einer Transaktionsverarbeitung in einen 24 x 80 Byte großen Puffer innerhalb des CICS-Terminals dar. Der GUI-Prozess liest diese Information aus und erzeugt aus ihr die in Abb. 6.6.4 wiedergegebene Darstellung. Wird eine Schachbrettfigur mit der Maus auf ein anderes Feld gezogen, stellt der GUI-Prozess diese Information in den 24 x 80 Byte- Puffer und startet die Datenübertragung zum CICS-Server.

6.6.2 EPI und ECI

Um die GUI-Programmierung zu vereinfachen, stellt CICS eine Schnittstelle zur Verfügung, die u.a. den Zugriff auf die Felder des 24 x 80 Byte-Puffers vereinfacht. Dieses API wird als das "External Programming Interface (EPI)" bezeichnet.

Mit Hilfe des EPI ist es möglich, eine gefälligere Bildschirmdarstellung zu erzeugen. Dies ist transparent für die Server-seitigen CICS-Anwendungsprogramme. Es ist jedoch keine zusätzliche Funktionalität möglich (z.B. bei der Scroll-Funktion).

Die Umstellung ist einfach und kann in kurzer Zeit geschehen. Diese Prozedur für die existierenden 30 Millionen CICS-Endgeräte ist jedoch eine große Aufgabe. Zahlreiche Startup-Firmen haben eigene Software-Produkte entwickelt, um den Umstellungsprozess zu erleichtern (siehe z.B. die Firmen. Attachmate, Platypuspartners, oder "Host-on-Demand" von IBM).

Es entstehen Schwierigkeiten mit der Wartung der Screens. Angenommen für 2000 Benutzer müssen 2000 Emulatoren gewartet werden. Screens können nicht mehr bequem geändert werden, z.B. ist die Operation "move field 1 byte" leichter möglich, wenn nur eine zentrale Map geändert werden muß als wenn 2000 Endgeräte-Emulatoren geändert werden müssen..

Es existiert eine moderne Alternative zu dem EPI, die "External Communication Interface (ECI)". Sie basiert auf Java und einem Web Application Server und eliminiert alle 3270- und/oder EPI-Probleme.

Bisher gibt es aber nur eine sehr geringe Anzahl von Fachleuten, die auf den Gebieten Java, Web Application Server, CICS und OS/390 gleichermaßen zu Hause sind. Der Einsatz der ECI- und ähnlicher Client/Server-Werkzeuge (z.B. Corba) läuft deshalb nur sehr langsam an.

Abb. 6.6.5 gibt die drei prinzipiellen Anschlußmöglichkeiten für CICS-Klienten wieder:

- 3270 Protokoll mit und ohne EPI/Screen Scraping
- Java, Web Application Server und ECI
- Corba

Die Abschnitte 6.4 bis 6.6 erläuterten die erste der drei Alternativen. Die beiden anderen Anschlußmöglichkeiten werden in Kapitel 8 dieses Buches behandelt.

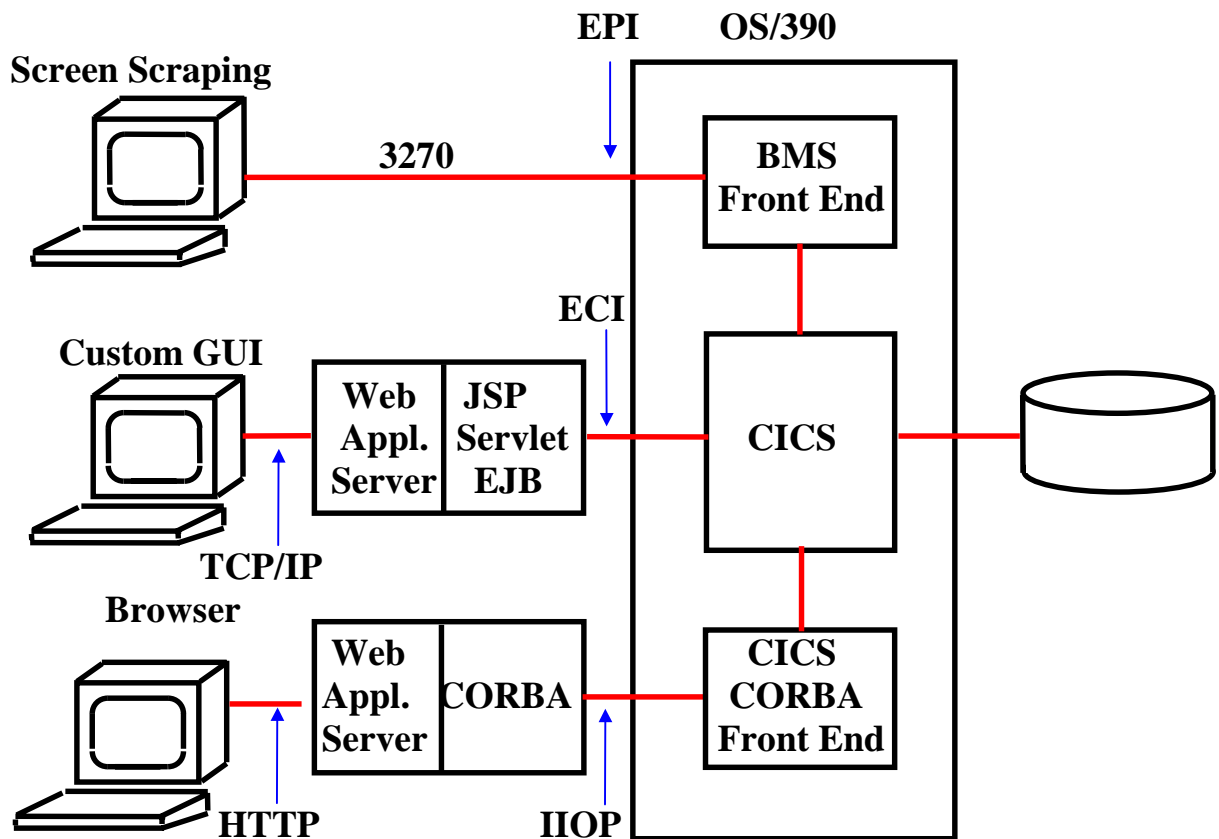


Abb. 6.6.5 : CICS-Klienten-Anbindung