18. Java Connection Architecture

18.1 SNA Communication over TCP/IP

18.1.1 System Network Architecture

z/OS Networking besteht heute aus zwei getrennten Netzwerk Architekturen: System Network Architecture (SNA) und TCP/IP. SNA entstand in 1974 und hat einige Funktionen, die TCP/IP erst mit Version 6 ebenfalls haben wird. Bis in die 90er Jahre war SNA der de facto Standard in allen größeren Unternehmen. Wegen der Bedeutung des Internets stellten in den letzten 10 Jahren die allermeisten Unternehmen ihre physischen Netze von SNA auf TCPIP um; dieser Prozess ist nahezu abgeschlossen.

IBM unterstützt weiterhin SNA und wird dies auch in Zukunft tun. Viele z/OS Komponenten nutzen nach wie vor SNA. Hierzu werden SNA Nachrichten in TCP/IP Nachrichten gekapselt und vom Empfänger wieder ausgepackt.

Die Kommunikation eines 3270 Emulators mit TSO oder CICS erfolgt über SNA auf diese Art.

Eine weitere Netzwerkarchitektur neben SNA und TCP/IP ist OSI. OSI wurde 1987 auf der CeBIT in Hannover in einem Verbund vorgestellt, in dem 10 ausstellende Unternehmen eine x.400 Intersystem Kommunikation ihrer eigenen OSI Implementierungen vorstellten. x.400 ist das OSI Äquivalent zu dem TCP/IP Simple Mail Transfer Protocol SMTP.

OSI war geplant, SNA und TCP/IP zu ersetzen, was aber nicht passiert ist. Die meisten OSI Produkte sind wieder verschwunden; einige Teilkomponenten existieren noch und haben sich durchgesetzt. Beispiele sind X.400, X.500, x.509 und x.25.

Die OSI Darstellung eines Schichtenmodells hat sich jedoch durchgesetzt und wird auch von TCP/IP benutzt.

18.1.2 OSI - Protokollhierarchie

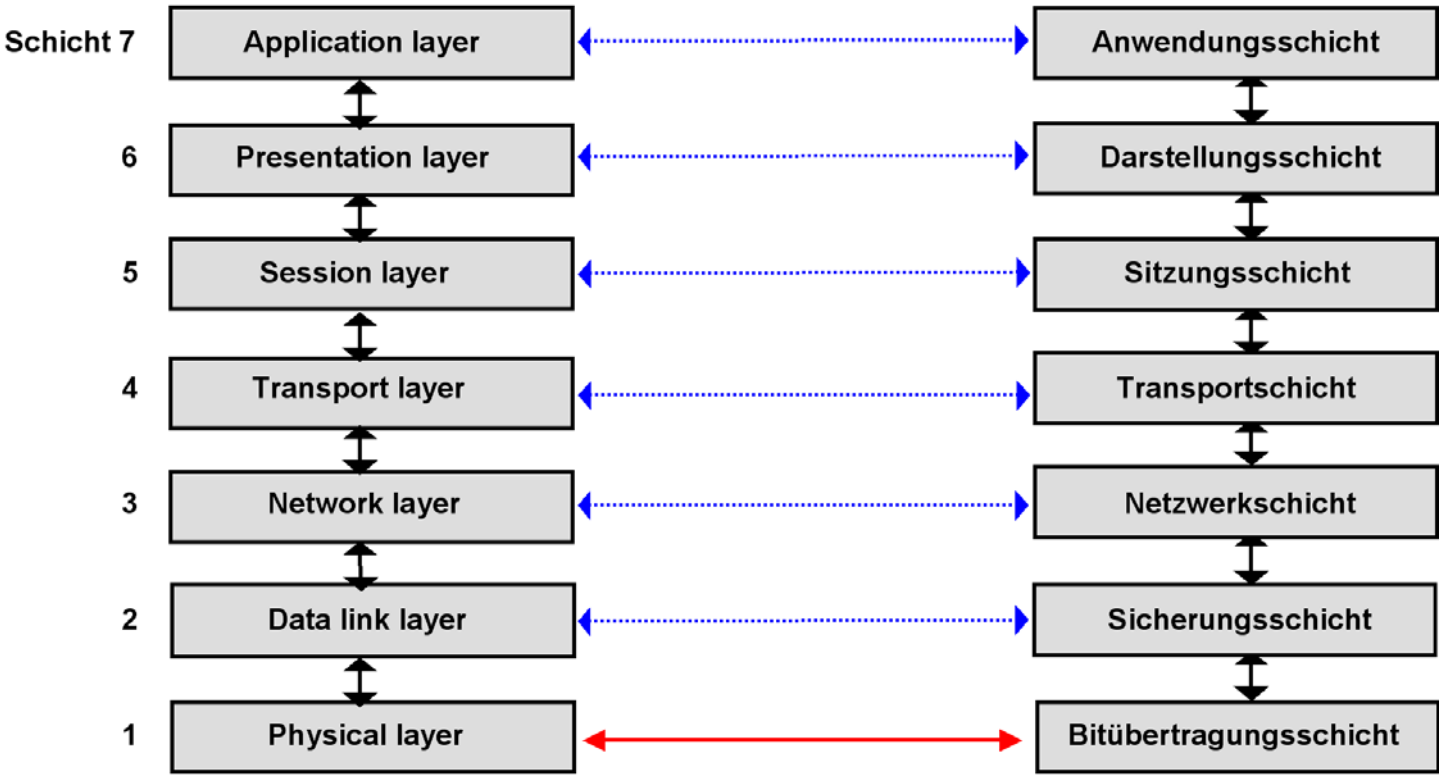


Abb. 18.1.1
Bezeichnung der OSI Schichten

Um die Komplexität einer Netzwerk Implementierung besser verwalten zu können, teilt man die Komponenten in 7 Schichten auf. Jede Schicht glaubt, Nachrichten direkt an die entsprechende Schicht des Partners zu senden. In Wirklichkeit werden Nachrichten an die darunterliegende Schicht weiter gegeben. Nur die beiden Komponenten der untersten Schicht kommunizieren in Wirklichkeit miteinander.

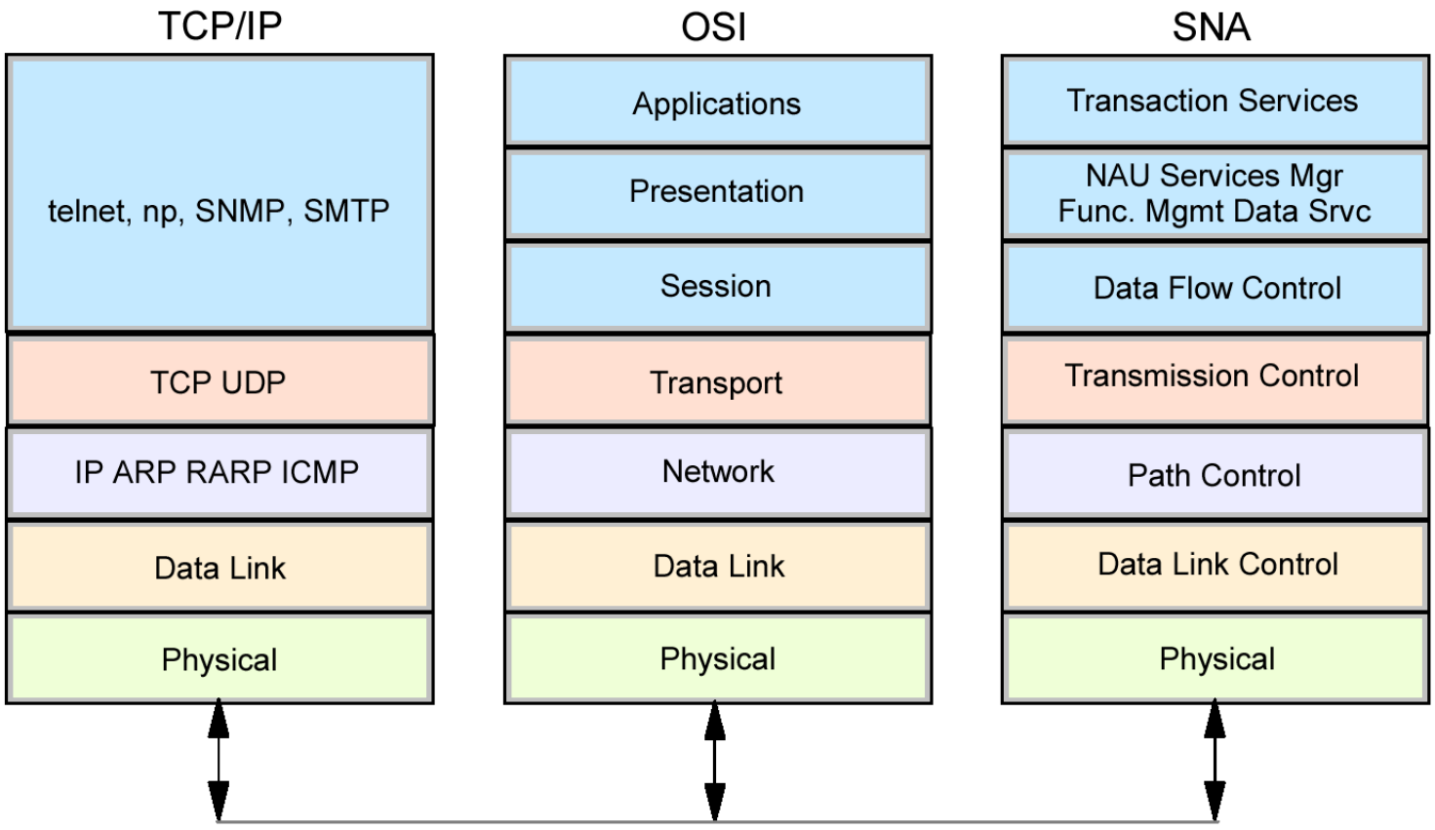


Abb. 18.1.2
SNA und TCP/IP Netzwerk Modell mit 7 Schichten

OSI verlor die Schlacht um Marktanteile an TCP/IP. Mit ganz wenigen Ausnahmen wird kein Geld mehr in weitere OSI Entwicklungen investiert. Das OSI Layer Referenz Modell wurde jedoch fortgeschrieben um auch TCP/IP und SNA zu beschreiben. Abb. 18.1.2 zeigt die Abbildung des OSI Modells auf TCP/IP und SNA.

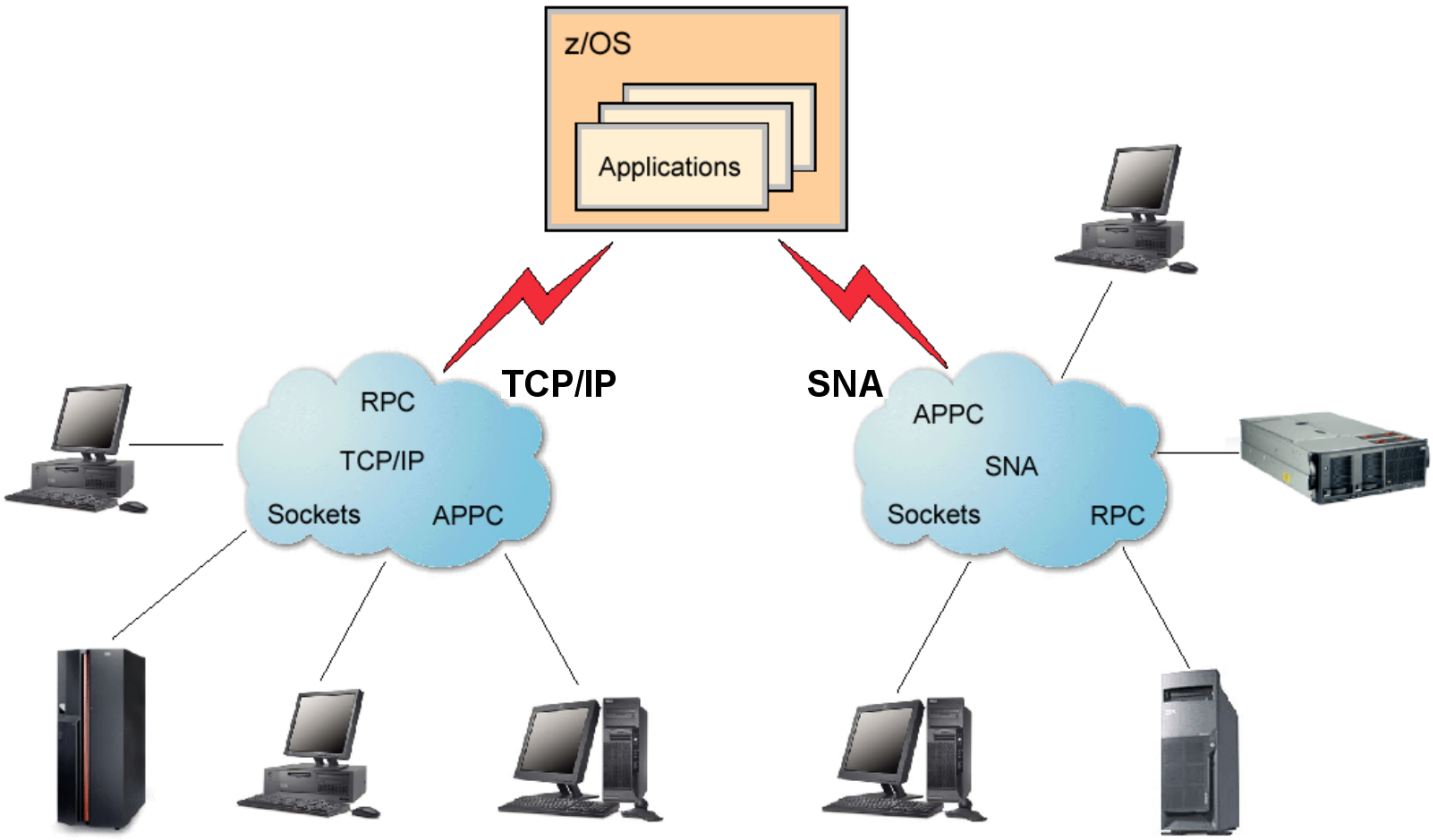


Abb. 18.1.3
Der z/OS Communication Server unterstützt sowohl TCP/IP als auch SNA

Das z/OS „Communication Server Subsystem“ läuft in einem eigenen Address Space und unterstützt neben dem TCP/IP Stack einen vollständigen SNA Stack. Dieser wird heute benutzt, um SNA Nachrichten in TCP/IP Nachrichten zu verpacken und entpacken.

18.1.3 SNA-Vokabular

SNA ist ein Session-orientiertes Protokoll. Eine Session wird zwischen zwei Knotenrechnern aufgebaut (open), Daten werden ausgetauscht, und die Session wird geschlossen (close).

Knotenrechner in einem SNA-Netzwerk werden LU's (Logical Units) genannt. Der ältere Typ LU 2 wurde ursprünglich für nicht-intelligente Terminals entwickelt, und eignet sich nur für die Kommunikation zwischen einem Arbeitsplatzrechner und einem zentralen Server. Es existiert eine RPC ähnliche Funktionalität. Unintelligente 3270 Terminals oder PC´s mit einem 3270 Emulator verwenden LU 2, um mit einer z/OS TSO oder CICS Session zu kommunizieren.

Der derzeitig gängigste Typ wird als LU 6.2 bezeichnet. Er hat volle Peer-to-Peer Funktionalität in beiden Richtungen.

Die Hardware und Firmware (Abschnitt 12.3.1) eines Knotenrechners wird als PU (Physical Unit) bezeichnet. Eine PU steuert die Verbindungsleitungen zu einer anderen PU. Der derzeitig 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 z/OS Rechner (über einen „Kanal") angeschlossen und hat erweiterte Funktionen. Die auf einem z/OS-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 mehreren) Möglichkeiten einer Transaktionen, unter Verwendung des APPC API über eine Session mit einer anderen Transaktion zu kommunizieren. Siehe http://www.cedix.de/VorlesMirror/Band2/APPC.pdf.

CICS „Interprocess Communication (IPC) kann über APPC durchgeführt werden. In der Regel laufen viele Transaktionen der Reihe nach über eine Session.

18.1.4 Transaktionsverarbeitung über ein SNA Netz

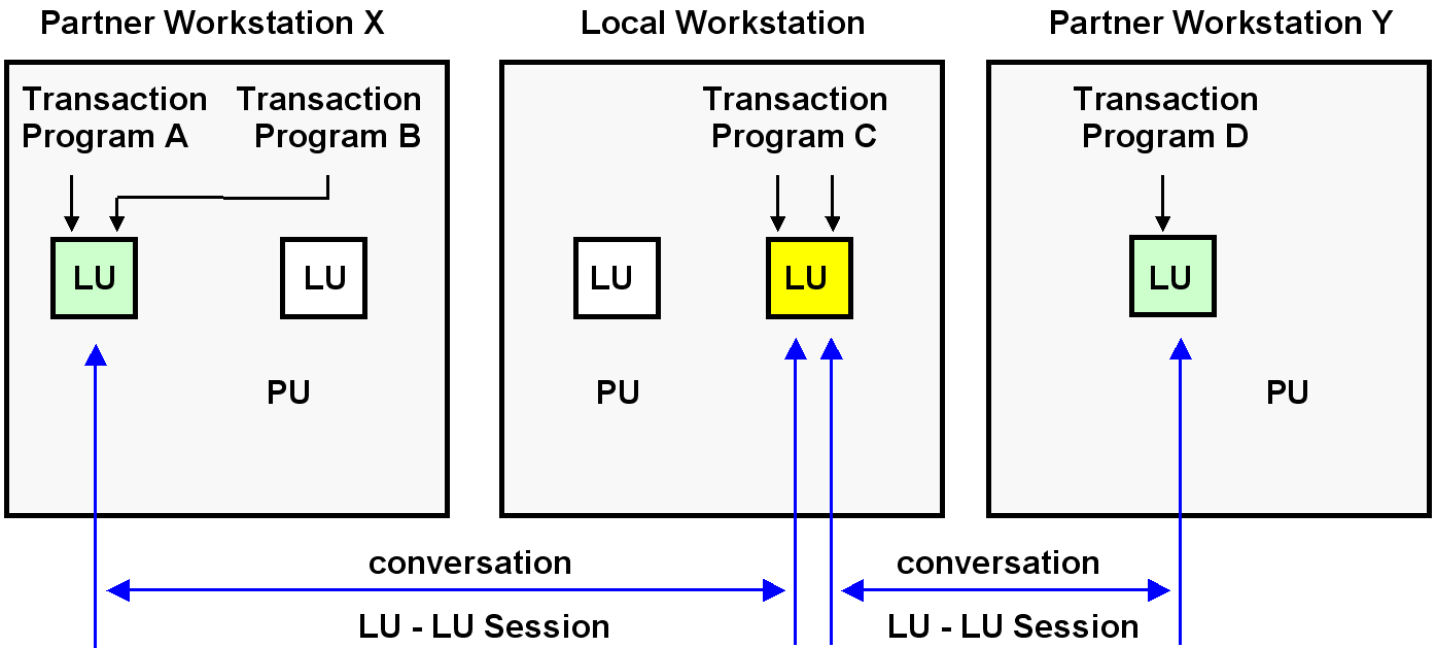


Abb. 18.1.4
LU 6.2 Sessions

Beispiel für eine Transaktionsverarbeitung über ein SNA Netz. Gezeigt ist eine lokale LU, die mit 2 Partner LU s kommuniziert. Der Zugriff zu den LU´s erfolgt über die APPC API.

Das TCP/UDP Protokoll stellt in der Schicht 5 des OSI Modells Sockets zur Verfügung. Mit Hilfe der Socket Schnittstelle können 2 Anwendungen auf 2 Rechnern miteinander kommunizieren.

Das SNA LU 6.2 Protokoll stellt in der Schicht 5 des OSI Modells LUs (Logical Units) vom Typ 6.2 zur Verfügung. Mit Hilfe der APPC-Schnittstelle (Application Program to Program Communication) können 2 Anwendungen auf 2 Rechnern miteinander kommunizieren.

Sockets und LU 6.2 sind in der Schicht 5 des OSI Modells angesiedelt, sehr mächtig, und schwierig zu programmieren. Deswegen werden RPCs benutzt, die unter der Decke Sockets oder LU 6.2 verwenden.

18.1.5 Die Zukunft von SNA

CICS, IMS, and DB2 Anwendungen benutzen auch heute noch unter der Decke SNA, auch wenn relativ erfolgreich versucht wird, dies vor dem Anwender zu verbergen. Dies geschieht mit Hilfe der Enterprise Extender (EE) Komponente.

Enterprise Extender (EE) ermöglicht es Ihnen, ein IP-Netzwerk für den Transport von SNA-Verkehr zu nutzen, einschließlich SNA-Verkehr zwischen verschiedenen Unternehmen.

Siehe http://www.cedix.de/VorlesMirror/Band2/SNA02.pdf

Die existierenden z/OS Installationen haben erhebliche Investitionen in 3270 und SNA Anwendungen getätigt. IBM wird VTAM als Teil des z/OS Communication Servers auch weiterhin zu unterstützen und verbessern, und mit neuen Technologien integrieren. Es existieren keine Pläne, die SNA Unterstützung in dem z/OS Communication Server einzustellen.

Und hier ein Tip für Sie:

Wenn Sie sich um einen Job bewerben, und beim Vorstellungsgespräch mit Vokabeln wie SNA oder LU 6.2 um sich werfen, ist Ihr Interviewer möglicherweise sehr beeindruckt.

Die Wahrscheinlichkeit ist sehr groß, dass der Sie interviewende Abteilungs- oder Hauptabteilungsleiter über SNA auch nicht mehr weiß als das, was auf diesen Folien draufsteht.

Auch in einer großen z/OS Installation gibt es heute in der Regel nur einen (oder wenige) hochbezahlten SNA Spezialisten.

18.1.6 Ursprüngliche 3270 Konfiguration

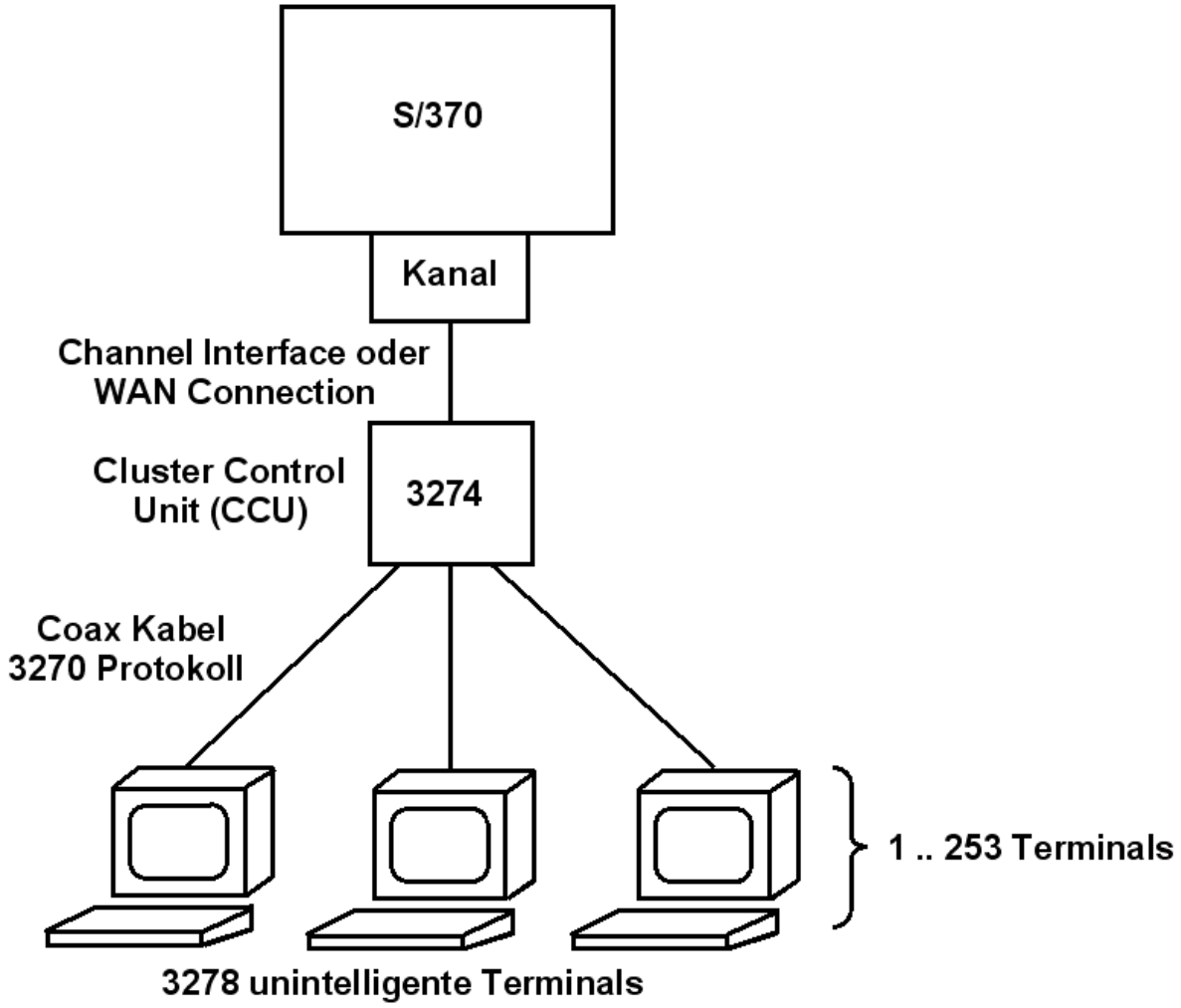


Abb. 18.1.5
Anschluss der 3278 Terminals

Dargestellt ist die ursprüngliche 3270 Konfiguration, wie sie 1972 verfügbar wurde. Ein Terminal bestand aus einem Bildschirm und einer Tastatur, und war mit einem Coax-Kabel mit einer Modell 3274 Steuereinheit verbunden.

Jeder 3278 Terminal war über eine point-to-point Coaxial Kabel Verbindung mit einer IBM 3274 Control Unit verbunden. Die Übertragungsgeschwindigkeit war 2.3 Mbit/s. Ähnlich wie Plattenspeicher werden auch Kommunikationsleitungen über eine Control Unit mit dem Mainframe verbunden

Ein Terminal hatte etwas Logik, aber keinen Processor. Jeder Terminal hatte einen Screen Buffer (memory), der die wiederzugebenden Zeichen speicherte.

Die 3274 Control Unit implementierte eine PU 2 und eine LU 2.1 . Die PU 2 war mit einer VTAM PU 5 verbunden, und die LU 2.1 mit einer TSO oder CICS LU verbunden.

Als IBM 3270 Terminal (siehe Abschnitt 9.1.4) wird eine Klasse von Bildschirmgeräten bezeichnet, die für eine Kommunikation mit einem Mainframe unter Benutzung des 3270 Protokolls entwickelt wurden. Text wurde auf dem ursprünglichen 3278 Terminal augenschonend in grüner Farbe auf einem schwarzen Hintergrund dargestellt; hierher rührt die Bezeichnung „Green Screen“. Anders als serielle ASCII Terminals minimiert der 3270 Terminal die Anzahl der Unterbrechungen auf dem Mainframe.

Später erschienen 3279 Terminals, welche farbige Buchstaben auf einem schwarzen Hintergrund darstellen konnten.

Physische 3270 Terminals werden heute nicht mehr hergestellt; sie wurden fast überall durch 3270 Emulatoren ersetzt, die nach wie vor von zahlreichen Herstellern angeboten werden.

Ein nicht unbeträchtlicher Teil von Benutzern bevorzugt heute nach wie vor einen 3270 Terminal an Stelle eines Bildschirmgerätes mit einer modernen grafischen Oberfläche. Der Grund ist eine höhere Produktivität, vorausgesetzt, man hat sich an die 3270 Darstellung gewöhnt.

Die ursprüngliche Coachs Kabel Verbindung war die erstmalige Implementierung eines Local Area Netzwerkes (LAN). Sie wurde später durch Tosen Ring LAN, und dann durch Ethernet LAN Verbindungen ersetzt.

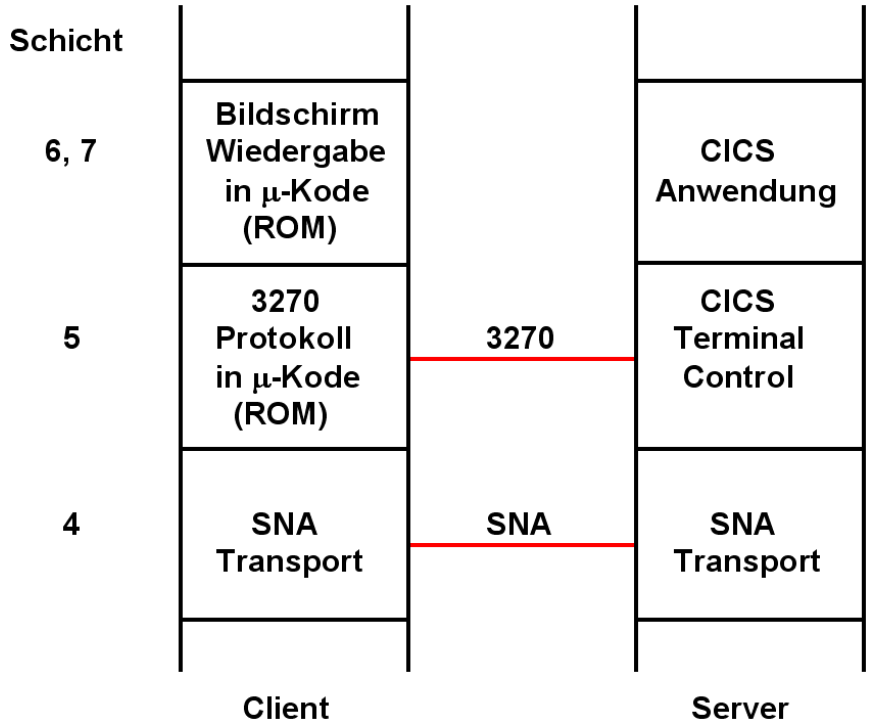


Abb. 18.1.6
Schichtenmodell der 3270 Architektur

Der ursprüngliche 3278 Terminal wurde über SNA mit dem Mainframe-Rechner verbunden.

Das 3270 Protokoll verwendete in der darunterliegenden Schicht eine SNA Session. Die Kombination von 3278 Terminal und 3274 Cluster Control Unit implementierte eine SNA LU 2 und eine PU 2 , wobei auf Grund der geringen logischen Fähigkeiten des 3278 Terminals der Großteil der Arbeit auf die 3274 entfiel.

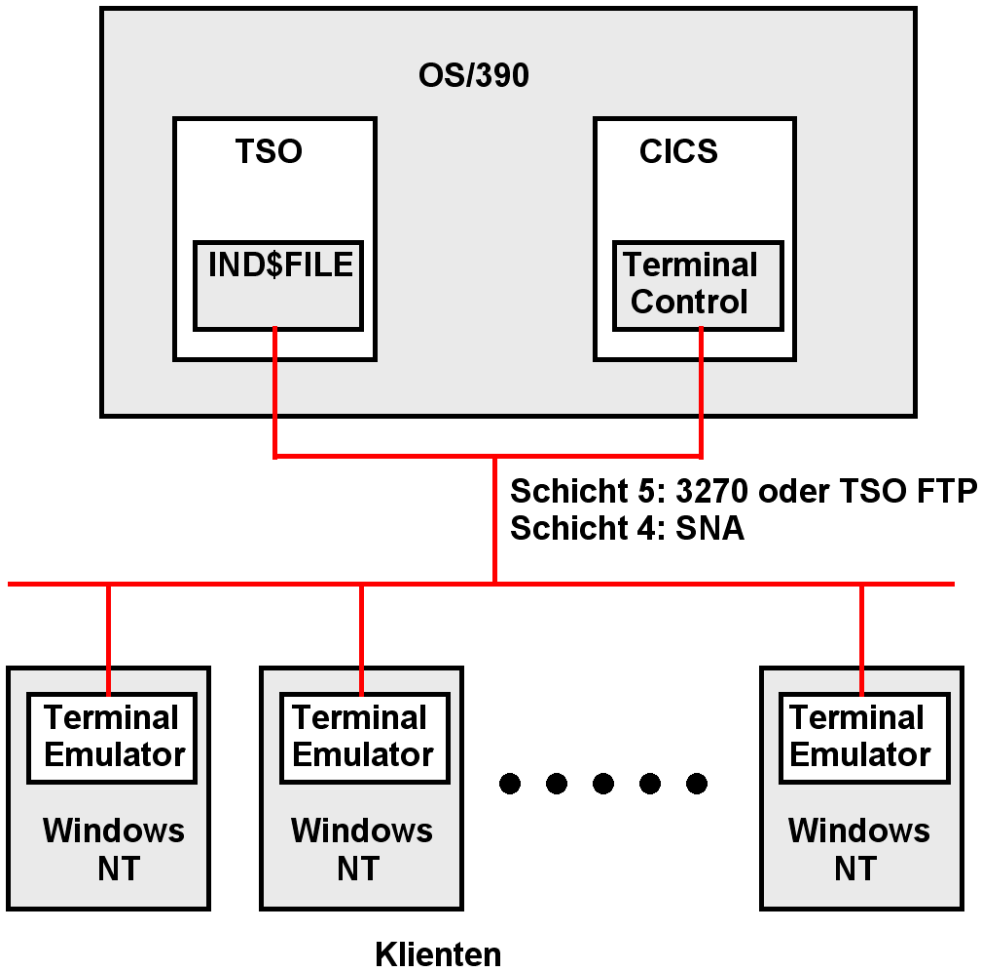


Abb. 18.1.7
Client/Server SNA Kommunikation mit 3270 Terminal Emulatoren

Mit der Verfügbarkeit kostengünstiger Windows (und OS/2) PCs wurden diese an Stelle der ursprünglichen 3278 Terminals und deren Nachfolge-Modellen eingesetzt.

Der Anschluss erfolgt typischerweise über ein LAN, zunächst Token Ring und später Ethernet. Das LAN-Netzwerk selbst benutzte SNA (die Umstellung auf TCP/IP erfolge erst viel später). Jeder PC verfügt über einen Terminal Emulator, welcher eine LU2 und PU2 implementierte. Auf der Mainframe Seite wurde eine PU 5 von VTAM, und eine LU durch die Anwendung implementiert, z.B. durch die CICS Terminal Control Komponente.

Zusätzlich hierzu verfügte diese Konfiguration über eine File Transfer (FTP Äquivalent) Einrichtung mit Hilfe eines TSO Commands namens „IND$FILE“.

18.1.7 Heutiger Zustand

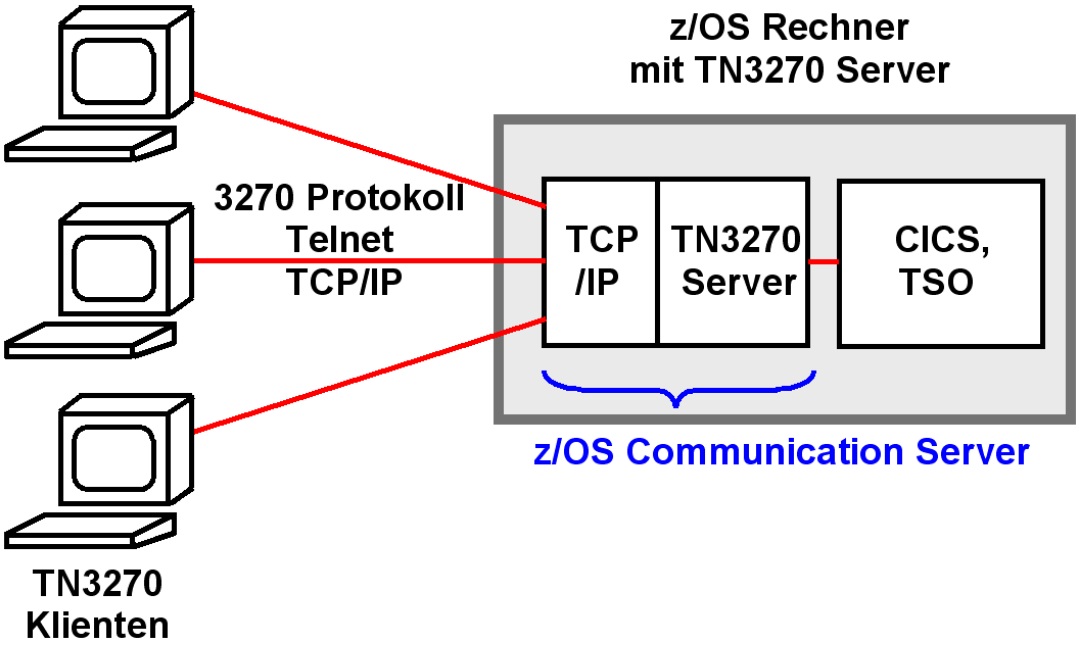


Abb. 18.1.8
TN3270 Protokoll

Heute kommunizieren TSO, CICS und IMS/DC Klienten und Server miteinander nach wie vor über das SNA Netzwerkprotokoll. Es existieren keine Pläne, dies zu ändern.

z/OS Installationen stellen aber ihr Netz von SNA auf TCP/IP um. Für Internet Verbindungen ist TCP/IP erforderlich. Lösung: SNA Protokolle über TCP/IP transportieren.

Die Verbindung eines 3270 Klienten mit dem z/OS Server erfolgt durch das TN3270 Protokoll. TN3270 ist ein erweitertes Telnet Protokoll; aus diesem Grund greifen 3270 Klienten meistens über Port 23 auf einen z/OS Server zu. Dies ist ein hierarchisches Netzwerk.

TN3270 Klienten sind über ein TCP/IP Netzwerk mit dem TN3270 Server verbunden, einer Komponente des z/OS Communication Server Subsystems. Der 3270 Client Klient (z.B. 3270 Emulator) emuliert eine LU2 Session und verpackt jede SNA Nachricht in eine TCP/IP Nachricht.

Der TN3270 Server simuliert jeden angeschlossenen TN3270 Klienten als ein logisches SNA Terminal (LU Typ 2 für Bildschirme, LU Typ 1 und 3 für Drucker).

Zwischen dem TN3270 Server und der z/OS Anwendung wird das SNA Protokoll benutzt. Zwischen dem TN3270 Server und dem TN3270 Klienten wird das 3270 Protokoll benutzt.

TN3270E erweitert das ursprüngliches TN3270 Protokoll und ist für das Internet in RFC 1647 spezifiziert.

18.1.8 TCP62 Protokoll

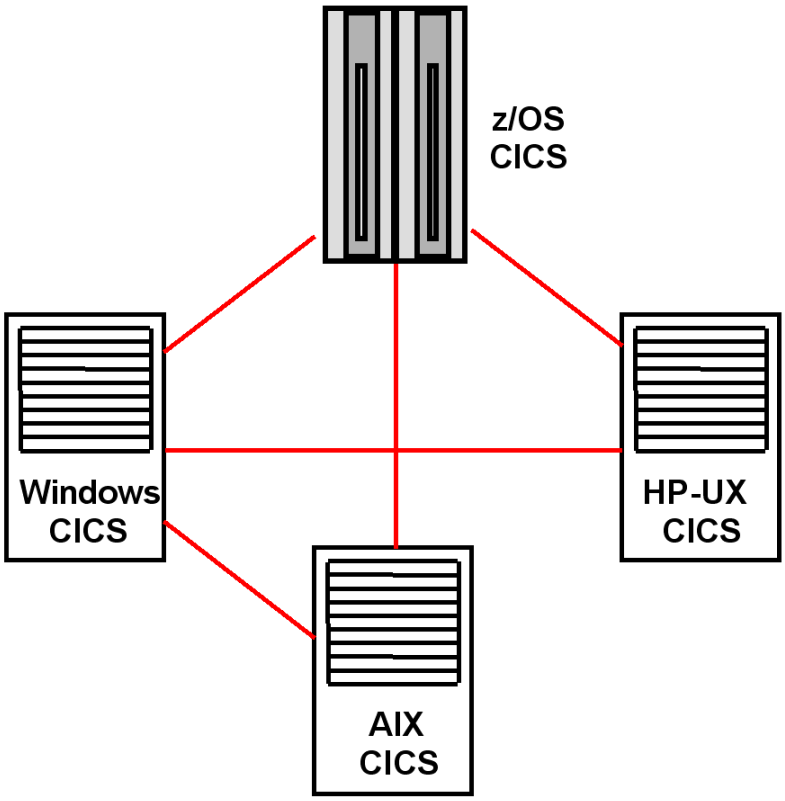


Abb. 18.1.9
Server - Server Konfiguration

Das TN3270 Protokoll ist ein Client/Server Protokoll. Im Gegensatz dazu existiert das Bedürfnis für eine Peer-to-Peer Kommunikation. Ein Beispiel hierfür ist die Kommunikation von zwei CICS Instanzen auf zwei geografisch getrennten z/OS Rechnern.

Die Verbindung erfolgt über das TCP62 Protokoll und eine SNA LU 6.2 –LU 6.2 Session. CICS verwendet hierfür den Distributed Link (DPL) Aufruf.

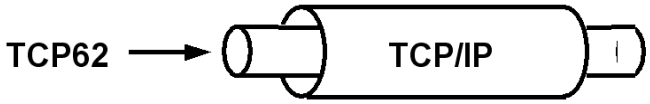


Abb. 18.1.10
Verpackung von TCP62 Nachrichten in TCP/IP Pakete

TCP62 oder TN3270 werden hierzu in TCP/IP Pakete verpackt und über das Internet versendet.

z/OS CICS kommuniziert auf die gleiche Art mit distributed CICS Instanzen auf Platformen wie Windows oder Solaris.

18.1.9 SNA und TCP/IP Protokolle

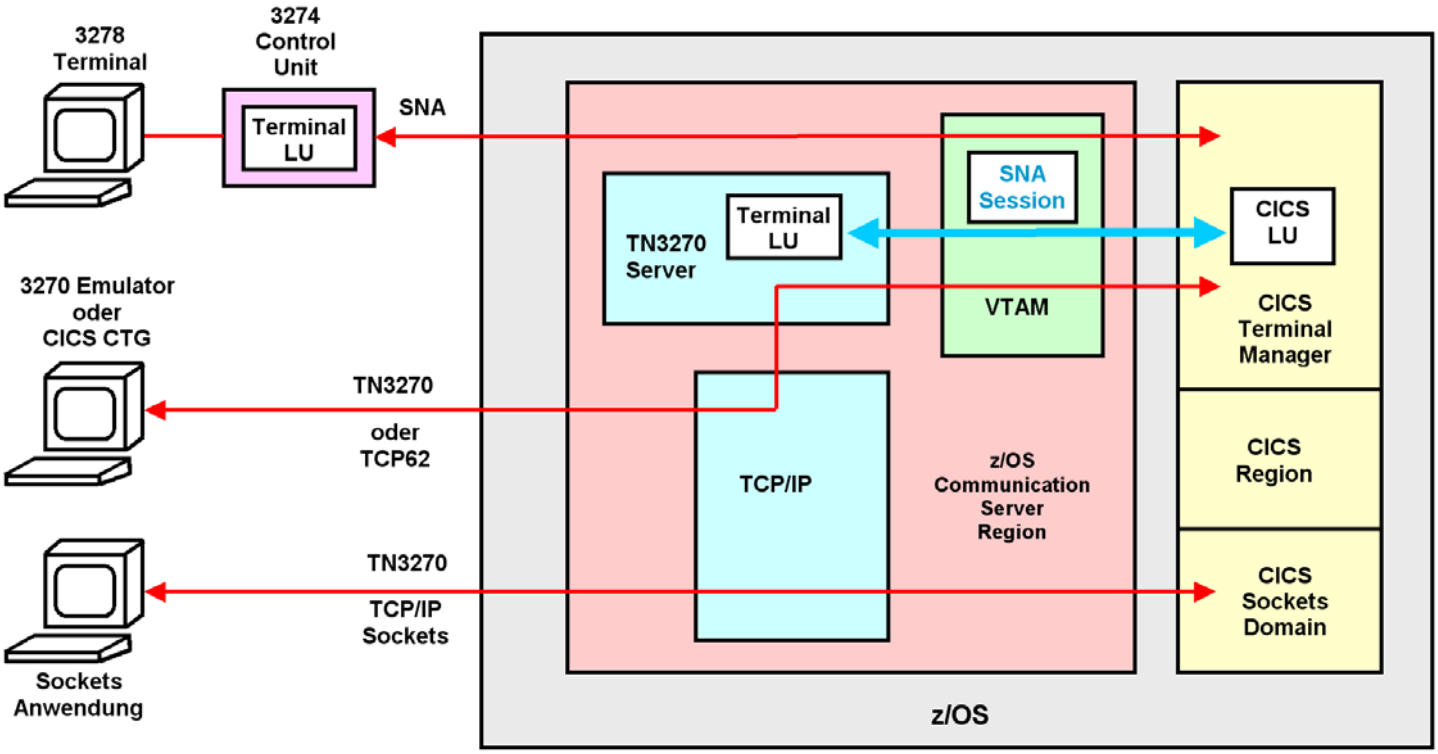


Abb. 18.1.11
CICS verfügt neben SNA Unterstützung auch über eine direkte TCP/IP Unterstützung

Abbildung 18.1.11 zeigt drei Arten der Anbindung eines Terminals an ein z/OS System.

TSO CICS oder IMS/DC Anwendungen glauben, mit einem SNA LU 2 Terminal zu kommunizieren. Der normale 3270 Emulator benutzt TN3270 für eine Verbindung mit z/OS. Der TN3270 Server des z/OS Communication Server Subsystems konvertiert die TN3270 Message in eine SNA Message. Der TN3270 Server kommuniziert mit TSO CICS oder IMS/DC Anwendungen über eine SNA Session über zwei LUs, einer LU2 (Terminal LU) und einer LU5 (z.B. CICS LU).

Heutige PCs haben genügend Leistung, um statt TN3270 das leistungsfähigere TCP62 Protokoll zu benutzen. In diesem Fall, der besonders für CICS interessant ist, wird der 3270 Emulator durch eine TCP62 Komponente, das CICS Transaction Gateway (CTG) ersetzt. Hiermit sind erweiterte Funktionen möglich, die besonders im Zusammenhang mit Java interessant sind.

Sockets sind eine weitere Alternative für Terminal Verbindungen. CICS verfügt über viele Terminal Attachment Alternativen. Eine davon ist die CICS Sockets Schnittstelle.

18.1.10 Zusammenfassung

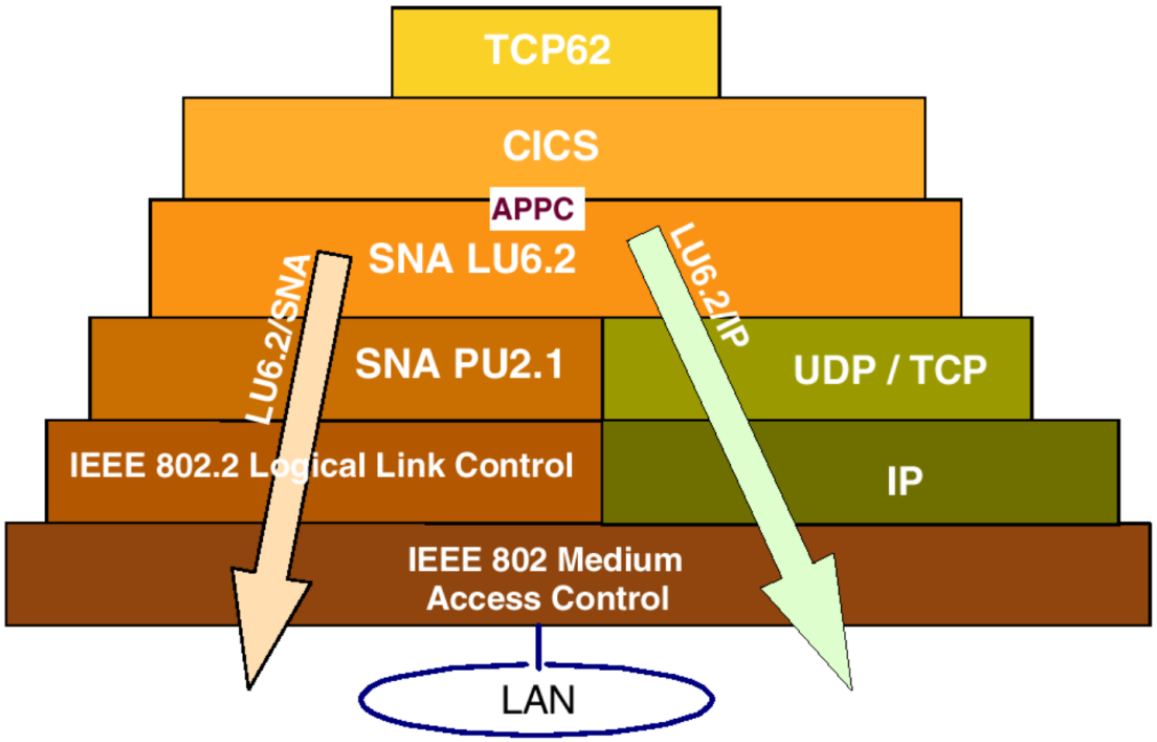


Abb. 18.1.
Koexistenz von SNA und TCP/IP

Ein CICS Subsystem kann mit einem anderen CICS System über SNA und den SNA Protokoll Stack kommunizieren. Die moderne Alternative besteht darin, dass ein CICS System mit einem anderen CICS System mittels des TCP/IP Protokoll Stacks und dem TCP62 Protokoll kommuniziert.

18.2 CICS Interface

18.2.1 Business- und Präsentationslogik

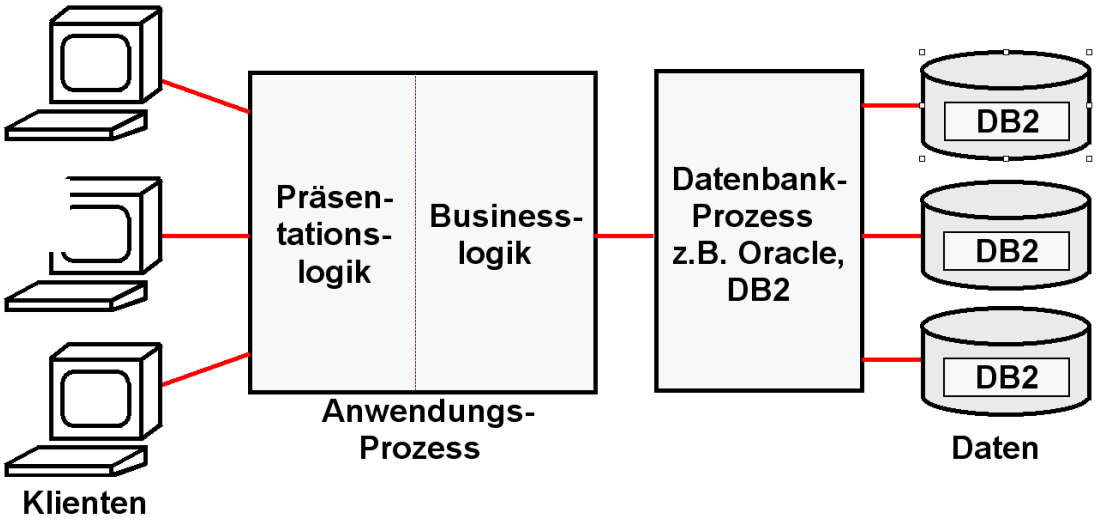


Abb. 18.2.1
Aufteilung in Business Logik und Präsentation Logik

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. Eine moderne 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 Programme werden in der BMS Sprache geschrieben.

18.2.2 z/OS Internet Integration

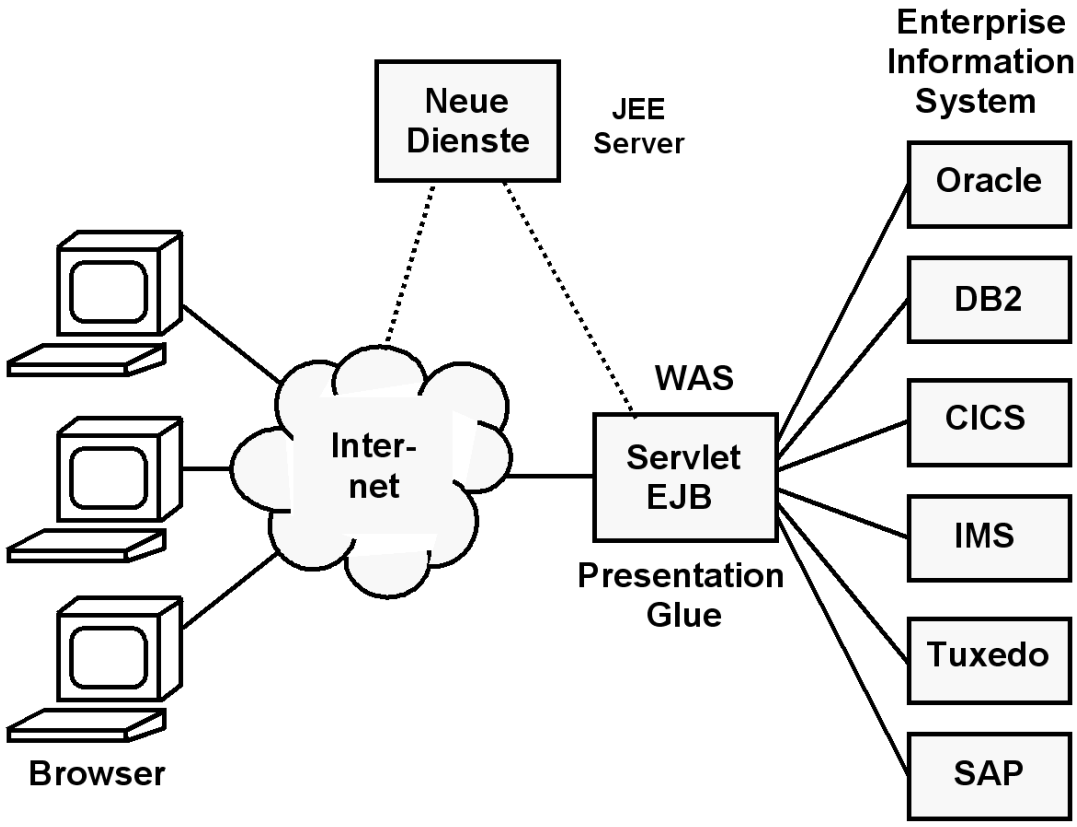


Abb. 18.2.2
z/OS Anbindung an das Internet

Unternehmenskritische Anwendungen und Datenbank-Prozesse laufen in der Regel auf einem zentralen Server. In mittleren und großen Unternehmen und Organisationen ist dies in der Regel ein z/OS Rechner. Aufgabenstellung:

- **Die existierende IT-Infrastruktur mit den Möglichkeiten des Internets integrieren.**
- **Die existierende IT-Infrastruktur so umstrukturieren, so dass sie mit weniger Personal an die sich in immer kürzeren Zeiträumen ändernden Geschäftsbedingungen angepasst werden kann.**

  Hierfür hat sich die Namen Enterprise Application Integration (EAI), sowie Service Oriented Architecture (SOA) eingebürgert.

Die Enterprise Application Integration (EAI) ist ein Ansatz für die Integration von Applikationen und Datenquellen. Dieser soll den Austausch von Daten und die Verbindung von Geschäftsabläufen vereinfachen. Es soll versucht werden, die Zugriffe der unterschiedlichsten Art von (Java) Klienten auf die unterschiedlichsten Arten von Enterprise Information Systems (EIS) zu vereinheitlichen und zu automatisieren. Beispiele für Eins sind CICS, IMS, Oracle, DB2, Tuxedo.

EAI ist im Wesentlichen ein rein technischer Ansatz zur Integration von Anwendungssystemen. Service Oriented Architecture (SOA) betont die Verbindung zu den nicht-technischen Geschäftsprozessen.

18.2.3 Alternativen der CICS Bildschirmausgabe

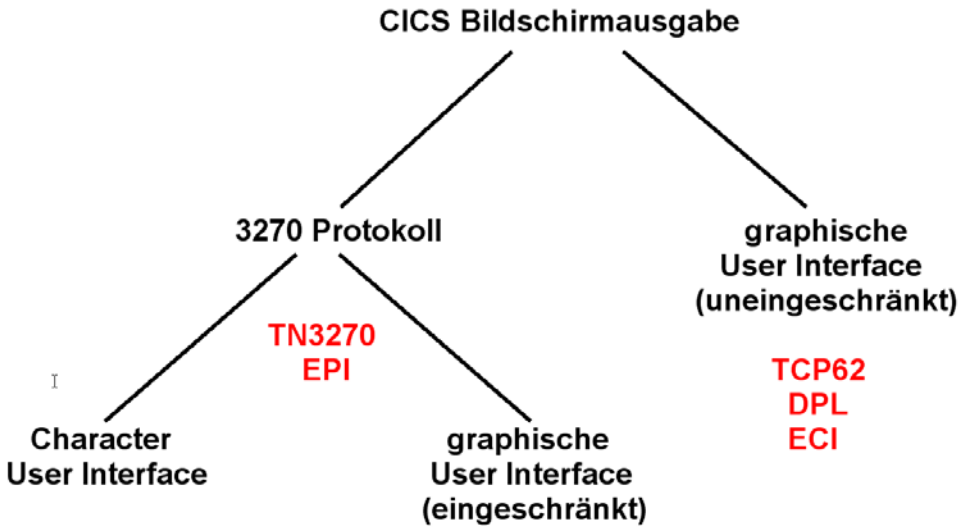


Abb. 18.2.3
Zwei Arten der CICS grafischen Bildschirmausgabe

Die CICS Bildschirmausgabe erfolgt entweder über das 3270 Protokoll unter Nutzung der CICS Terminal Control Komponente und des Basic Mapping Supports (BMS), oder über einen direkten COMMAREA Zugriff.

Im ersten Fall erfolgt die Datenübertragung mit Hilfe des TN3270 Protokolls und der EPI Schnittstelle. Es ist zunächst eine Character User Darstellung möglich (Green Screen). Alternativ kann der 3270 Datenstrom mit Hilfe eines als „Screen Scraping“ bezeichneten Ansatzes grafisch dargestellt werden.

Im zweiten Fall erfolgt die Datenübertragung mit Hilfe des TCP62 Protokolls und der ECI Schnittstelle. Es wird die CICS Distributed Program Link (DPL) Kommunikation eingesetzt. Einschränkungen in den Möglichkeiten der grafischen Darstellung, die im ersten Fall durch das 3270 Protokoll bedingt sind, werden hierbei ausgeschlossen.

18.2.4 Ablauf einer COMMAREA Operation

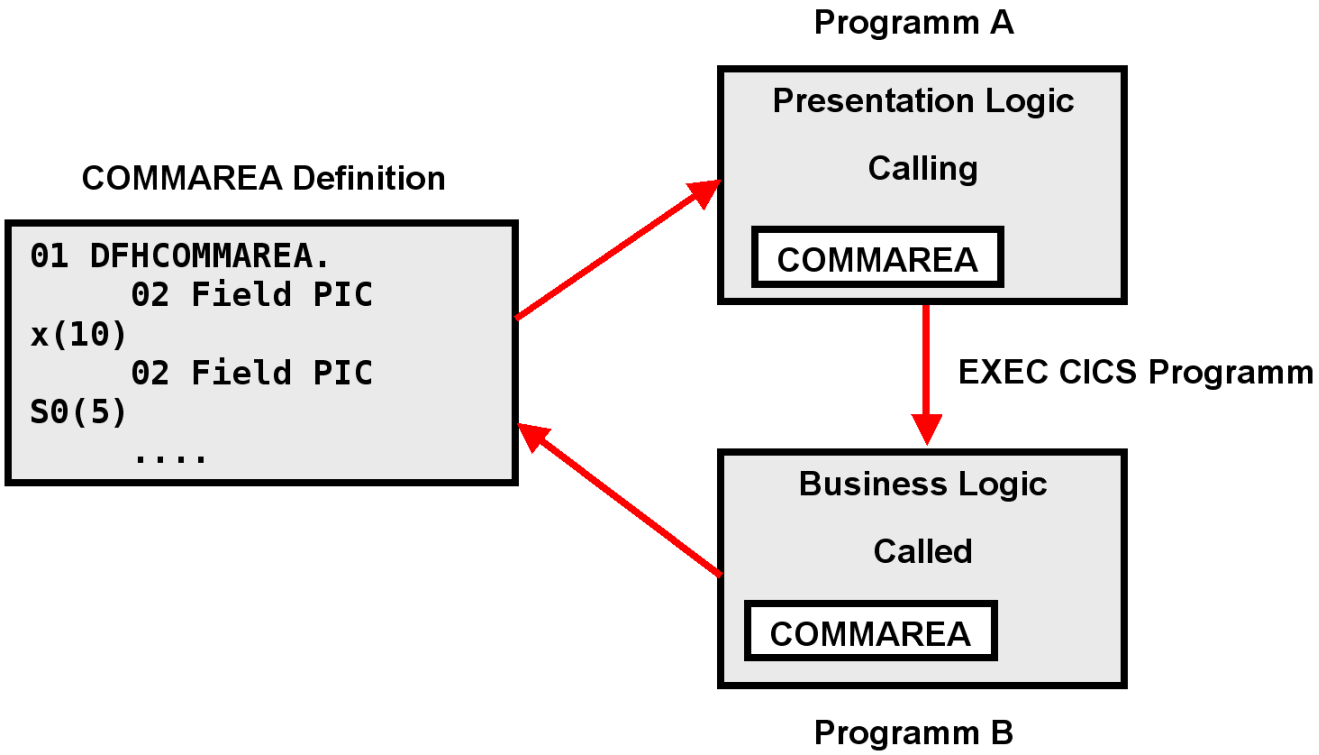


Abb. 18.2.4
Kommunikation mittels COMMAREA

Für die Kommunikation von Eingabe und Ausgabe Informationen extrahiert das aufrufende (calling) Presentation Logik Programm Daten aus der Eingabe Nachricht, und setzt Eingabefelder in der COMMAREA. Es ruft dann ein weiteres (called) CICS-Programm auf. Dieses Programm legt die Ergebnisse der Business Logik Operation wieder in Felder der COMMAREA ab und gibt die Kontrolle an das aufrufende Programm zurück. Letzteres macht das Ergebnis über die COMMAREA verfügbar.

```
01 DFHCOMMAREA.
     02  CA-RETCODE                  PIC 9(8) COMP.
     02  CA-SWSECI1-COMMAREA.
         05  CA-NUMBER-OF-ROWS       PIC 9(4) COMP.
         05  CA-ERROR-MESSAGE        PIC X(10).
         05  CA-CURRENT-DATE         PIC X(8).
         05  CA-CURRENT-TIME         PIC X(8).
         05  CA-CICS-ABSTIME         PIC S9(15) COMP-3.
         05  CA-ROW-DATA  OCCURS 1  TO 1818 TIMES
                          DEPENDING ON CA-NUMBER-OF-ROWS.
             10  CA-ROW-NUMBER       PIC S9(4) COMP.
             10  CA-ROW-NUM-AS-CHAR  PIC X(6).
             10  CA-DATA             PIC X(10).
```

Abb. 18.2.5
Beispiel für den Inhalt einer COMMAREA

https://awebproxyprd.ins.state.ny.us/docs/aiciref/aicirefp10.htm

18.2.5 Die 3270 Schnittstelle

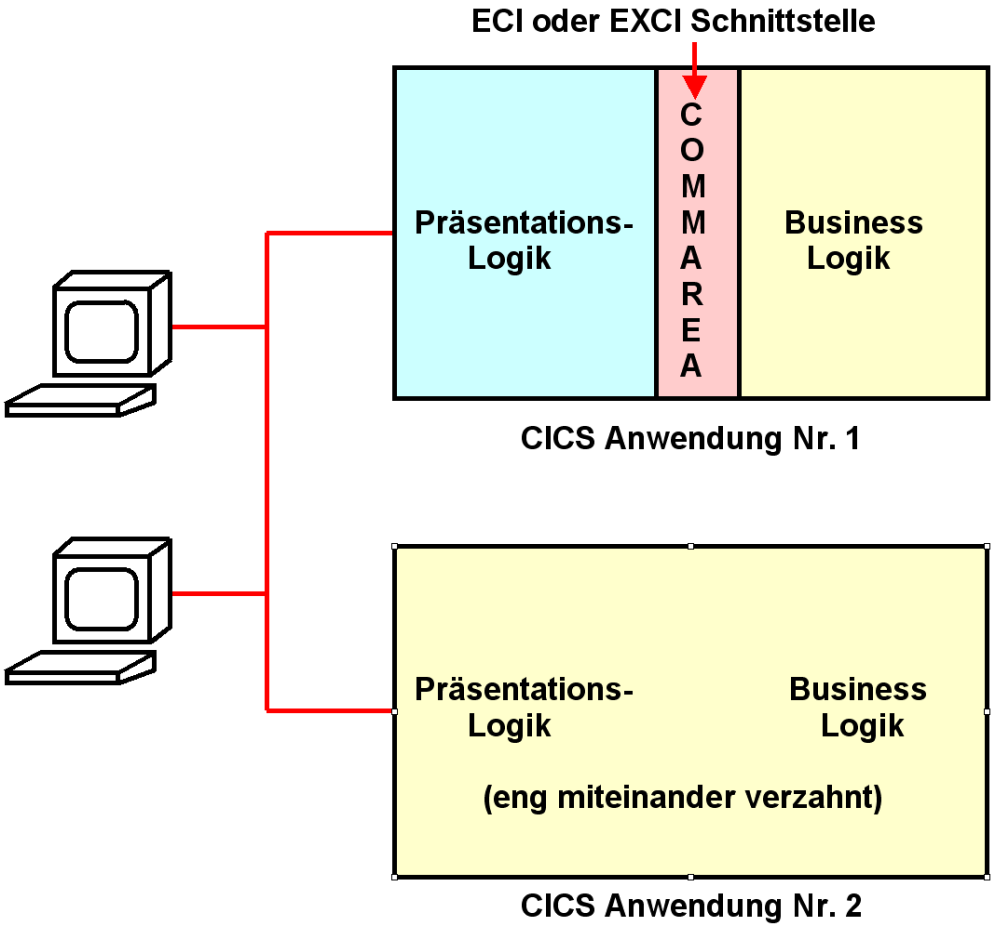


Abb. 18.2.6
COMMAREA als Schnittstelle zwischen Business Logik und Präsentation Logik

Aufgabe: Ersatz der 3270 Schnittstelle durch eine Web Browser Schnittstelle oder eine andere GUI.

Bei modernen CICS Anwendungen (Nr. 1) kommunizieren Business Logik und Präsentationslogik über COMMAREA. Hier ist es relativ einfach, eine Präsentationslogik durch eine andere zu ersetzen, oder für eine gegebene Businesslogik mehrere alternative Präsentationslogiken anzubieten.

Bei vielen älteren CICS Anwendungen (Nr. 2) sind Presentation Logik und Business Logik beide in z.B. Cobol oder PLI geschrieben und eng miteinander verwoben. Es ist vielfach nicht möglich oder sinnvoll, diese Anwendungen umzuschreiben, um eine saubere Trennung zwischen Business Logik und Presentation Logik zu erreichen. Hier ist eine Modernisierung der Oberfläche nur mittels Screen Scraping möglich.

18.2.6 Screen Scraping Implementierung mittels Java

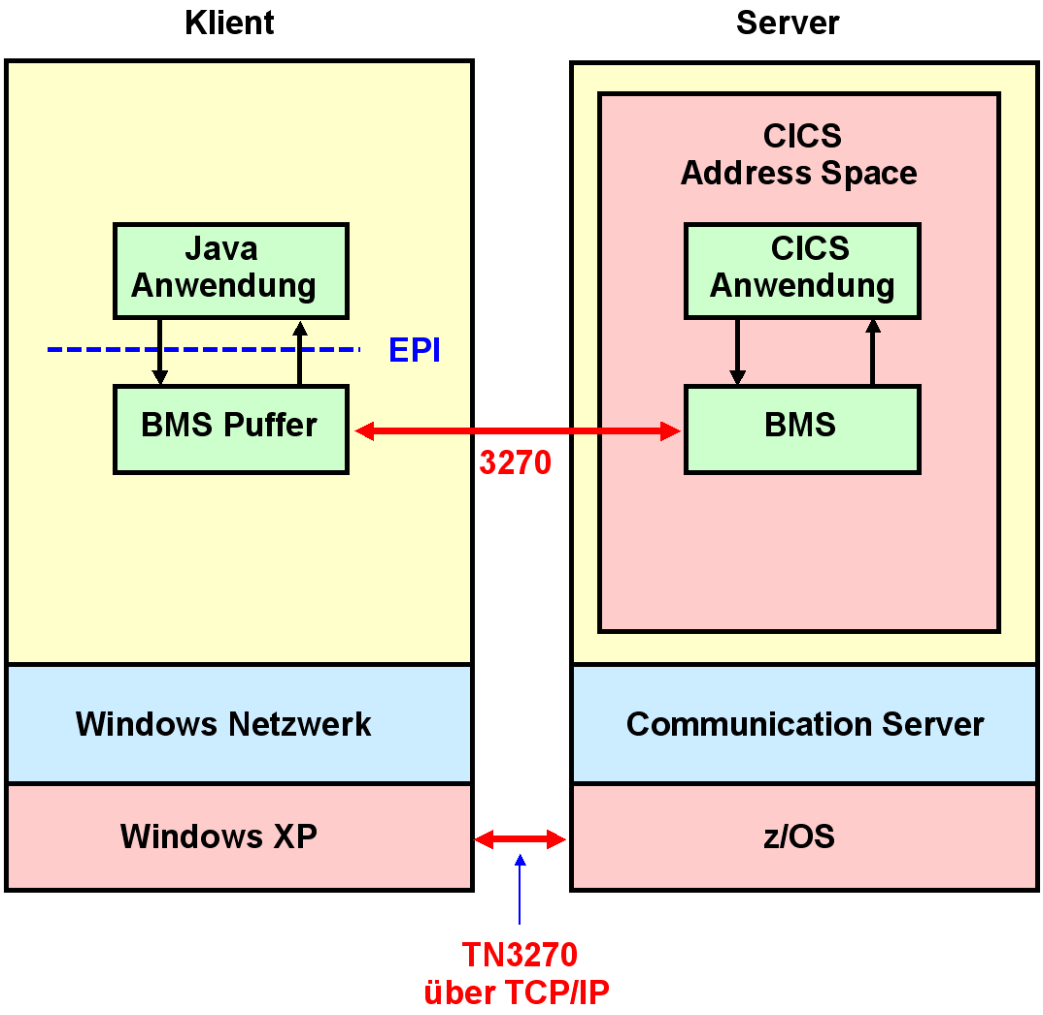


Abb. 18.2.7
Java Screen Scraping Programm auf dem Klienten

CICS Screen Scraping kann mittels eines Java Programms auf dem Klienten Rechner implementiert werden. Hierbei überträgt ein 3270 Datenstrom eine Ausgabenachricht, wie gehabt, in den 24 x 80 Byte großen BMS Puffer des 3270 Emulators. Das Java Programm greift über die EPI Schnittstelle auf den BMS Puffer Inhalt zu.

Eine Java Anwendung auf dem Klienten kann über die EPI Schnittstelle auf den Inhalt des BMS Puffers zugreifen, und mit dessen Daten eine gefälligere graphische Oberfläche (GUI) erzeugen. Die Anwendung kann 3270-basierte CICS Transaktionen starten und Daten senden und empfangen, die mit dieser Transaktion assoziiert sind. Die Anwendung kann alternativ auch in Java, C++, PL/1 oder einer beliebigen anderen Programmiersprache geschrieben sein.

Als Alternative zu dem hier dargestellten Verfahren kann die Screen Scraping Logik auf einem Server laufen.

Der Basic Mapping Support (BMS) ist Bestandteil des CICS Terminal Managers. Er wurde in Band 1, Abschnitt 9.1 erläutert.

18.2.7 Host Access Transformation Services

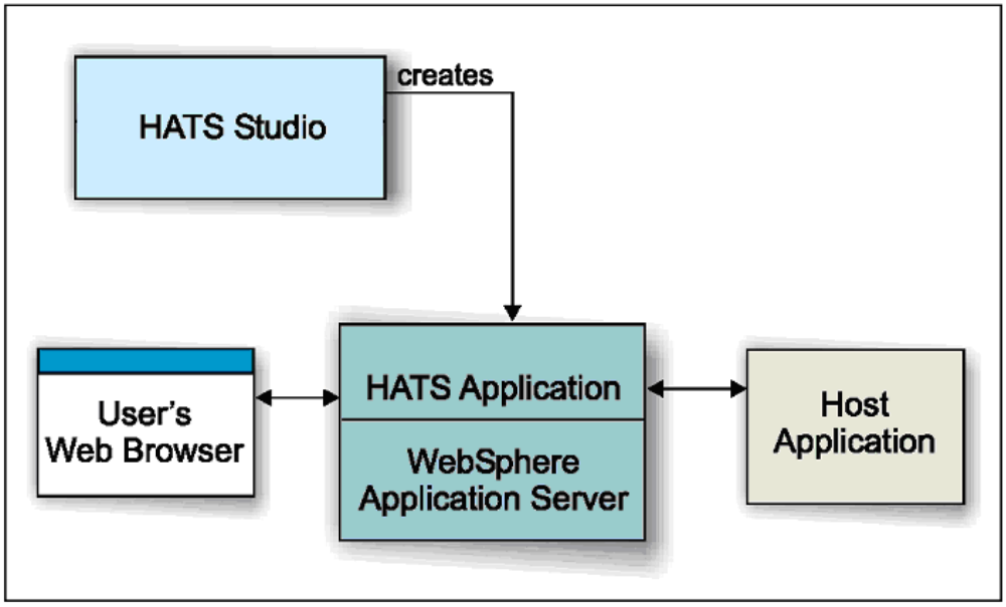


Abb. 18.2.8
HATS Studio erlaubt Erweiterungen der voll automatischen Konvertierung

Host Access Transformation Services (HATS) ermöglicht Screen Scraping auf einem Server. Der HATS Server erstellt für eine Reihe angeschlossener Klienten 3270 Bildschirmdarstellungen als HTML Seiten. HATS erkennt automatisch die Komponenten des 3270 Bildschirm-Inhaltes mit Hilfe eines Satzes vordefinierter Regeln. Es übersetzt die Komponenten des 3270 Bildschirm-Inhaltes in Echtzeit in HTML. Auf dem Klienten ist nur ein Browser erforderlich.

Im einfachsten Fall übersetzt der HATS Server den 3270 Datenstrom automatisch und unmittelbar in eine HTML Seite, die an den Browser des Terminals gesendet wird. HATS Studio ist eine optionale Entwicklungsumgebung, mit deren Hilfe das Erscheinungsbild auf dem Browser verbessert und angepasst werden kann.

Als Beispiel wird die in Abb. 18.2.9 gezeigte Bildschirmdarstellung ...

```
MAIN                        OS/400 Main Menu

Select one of the following:

    1.  User tasks
    2.  Office tasks
    3.  General system tasks
    4   Files, libraries, and folders
    5.  Programming
    6.  Communications
    7.  Define or change the system
    8.  Problem handling
    9.  Display a menu
    10. Information Assistant options
    11. Client Access/400 tasks

    90. Sign off

Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
F23=Set initial menu
```

Abb. 18.2.9
Ursprüngliche Darstellung

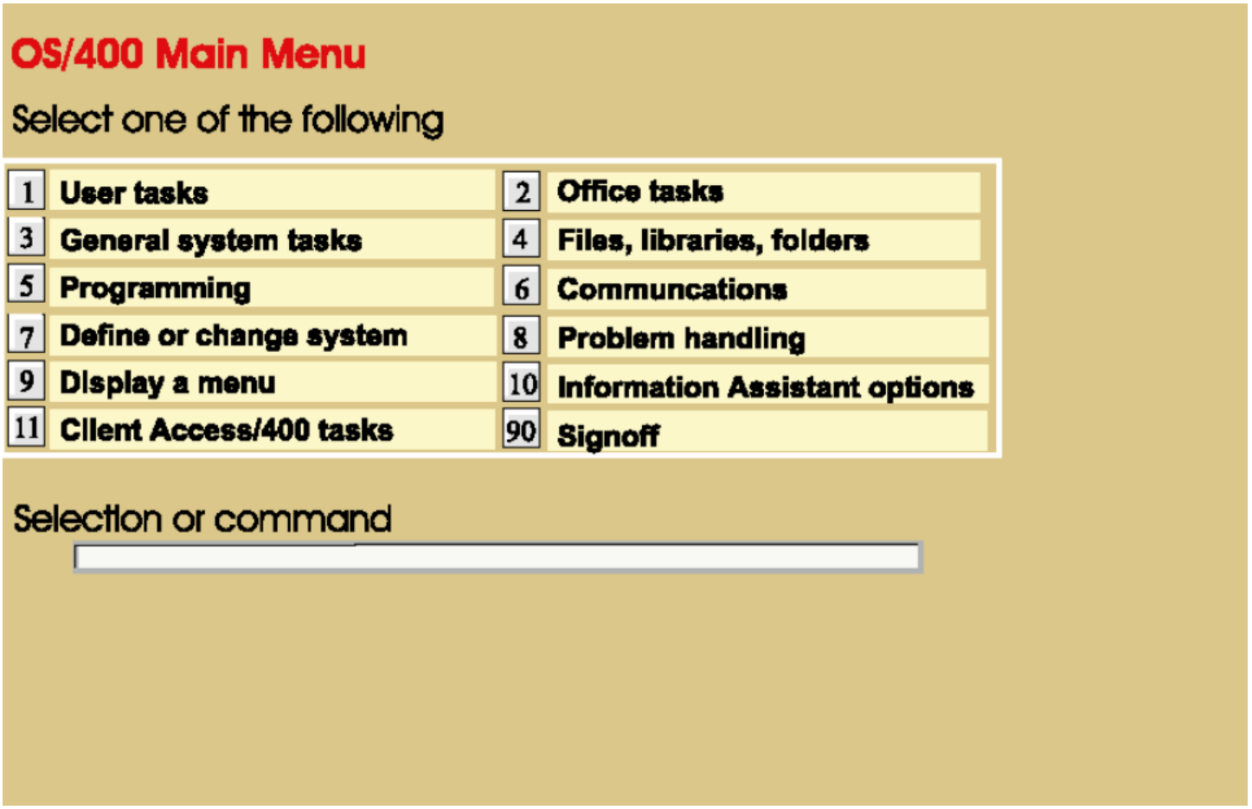


Abb. 18.2.10
Mit HATS automatisch erstellte Darstellung

... in die Darstellung von Abb. 18.2.10 automatisch übersetzt. Das kann für eine große Anzahl von Maps mittels eines automatisch ablaufenden Verarbeitungsvorgang geschehen, ohne dass jede einzelne Map angefasst werden muss.

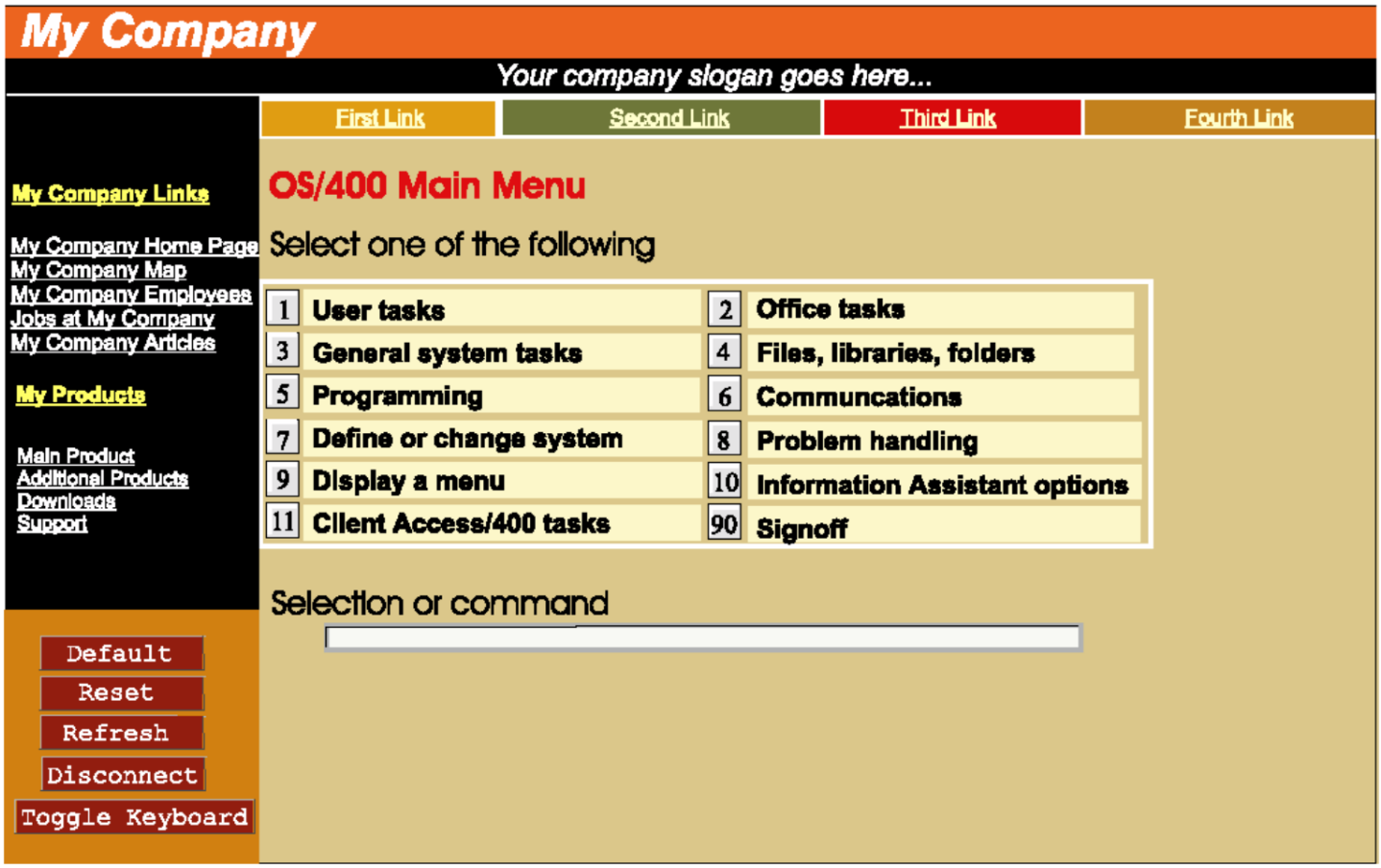


Abb. 18.2.11
Erweiterung mittels HATS Studio

Bei dieser Gelegenheit bietet sich an, die Bildschirmdarstellung um zusätzliche Elemente zu erweitern, die nicht Bestandteil des 3270 Datenstroms sind. Beispiele sind die Einfügung eines Firmenlogos, eine zusätzliche Liste mit Links, usw. Dies erfordert allerdings zusätzlichen Anpassungsaufwand, der mit Hilfe des HATS Studio erbracht werden kann.

18.2.11 HATS Beispiel Universität Tübingen

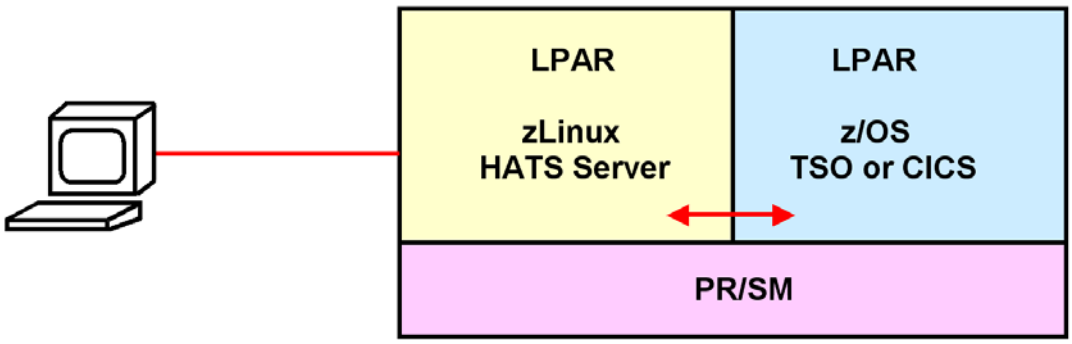
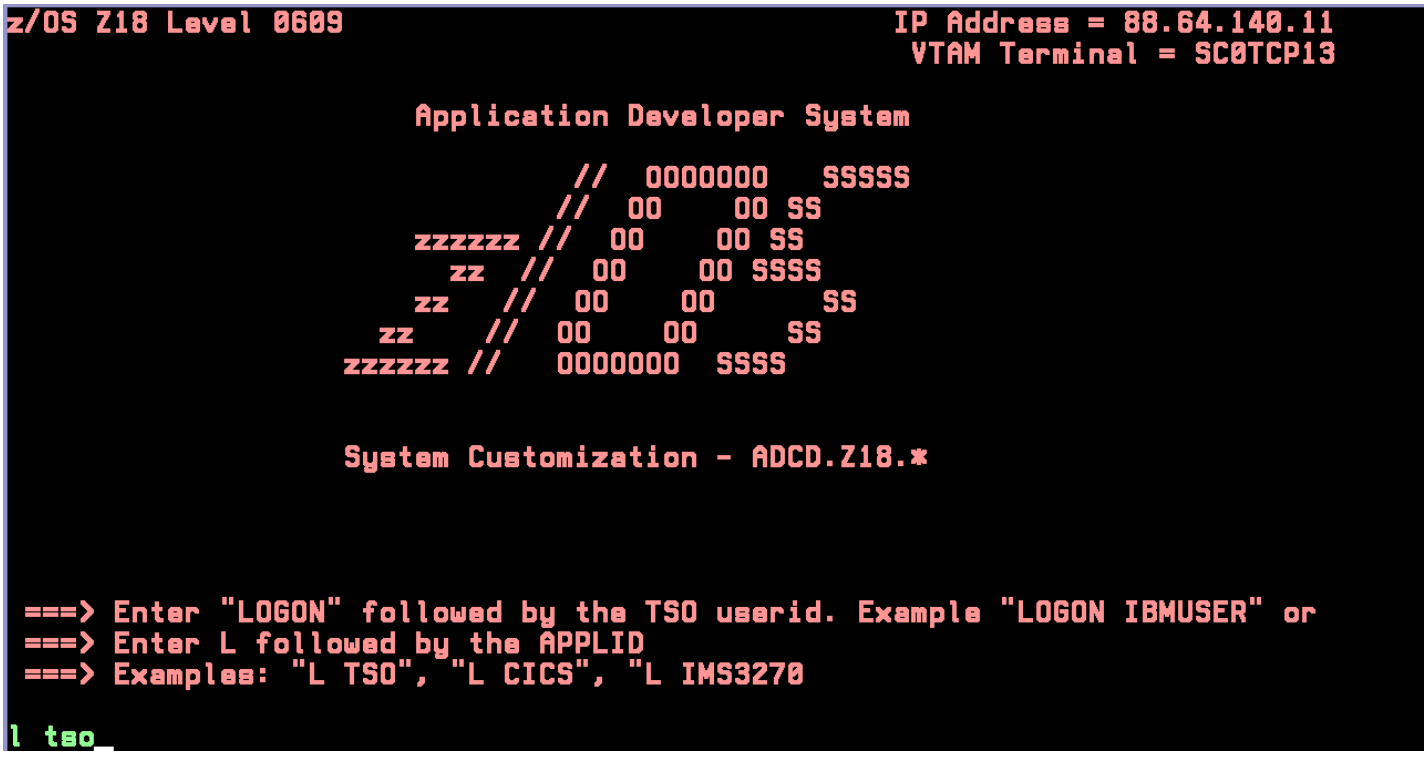


Abb. 18.2.12
HATS installiert auf einer zLinux LPAR

Sie können das vorliegende Beispiel selbst durchführen, indem Sie sich unter

http://galadriel.cs.uni-tuebingen.de:9080/csprak/

einloggen. Die folgenden Abbildungen zeigen einige Beispiele



Aus dem z/OS Welcome Screen wird …

Abb. 18.2.13
Eine mit HATS erstellte Darstellung

```
------------------------------ TSO/E LOGON ------------------------------

   Enter LOGON parameters below:                RACF LOGON parameters:

   Userid    ===> PRAK031

   Password  ===>                               New Password ===>

   Procedure ===> DBSPROC                       Group Ident  ===>

   Acct Nmbr ===> ACCT#

   Size      ===> 5000

   Perform   ===>

   Command   ===> ISPF

   Enter an 'S' before each option desired below:
           -Nomail         -Nonotice         -Reconnect         -OIDcard

PF1/PF13 ==> Help    PF3/PF15 ==> Logoff    PA1 ==> Attention    PA2 ==> Reshow
You may request specific help information by entering a '?' in any entry field
```

Aus dem Logon Screen wird ....

Abb. 18.2.14
eine mit HATS erstellte Darstellung

```
 Menu  Utilities  Compilers  Options  Status  Help
──────────────────────────────────────────────────────────────────────────────
                          ISPF Primary Option Menu                  End of data

0  Settings      Terminal and user parameters          User ID . : SPRUTH
1  View          Display source data or listings       Time. . . : 21:42
2  Edit          Create or change source data          Terminal. : 3278
3  Utilities     Perform utility functions             Screen. . : 1
4  Foreground    Interactive language processing       Language. : ENGLISH
5  Batch         Submit job for language processing    Appl ID . : ISR
6  Command       Enter TSO or Workstation commands     TSO logon : ISPFPROC
7  Dialog Test   Perform dialog testing                TSO prefix: SPRUTH
9  IBM Products  IBM program development products      System ID : ADCD
10 SCLM          SW Configuration Library Manager      MVS acct. : ACCT#
11 Workplace     ISPF Object/Action Workplace          Release . : ISPF 6.1
M  More          Additional IBM Products

     Enter X to Terminate using log/list defaults


Option ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=Actions  F12=Cancel
```

und aus dem ISPF Primary Option Menu wird ....

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

WILHELM-SCHICKARD-INSTITUT

ISPF Primary Option Menu

0 - Settings - Terminal and user parameters	User ID PRAK031
1 - View - Display source data or listings	Time 17:01
2 - Edit - Create or change source data	Terminal 3278
3 - Utilities - Perform utility functions	Screen 1
4 - Foreground - Interactive language processing	Language ENGLISH
5 - Batch - Submit job for language processing	Appl ID ISR
6 - Command - Enter TSO or Workstation commands	TSO logon DBSPROC
7 - Dialog Test - Perform dialog testing	TSO prefix PRAK031
9 - IBM Products - IBM program development products	System ID ADCD
10 - SCLM - SW Configuration Library Manager	MVS acct ACCT#
11 - Workplace - ISPF Object/Action Workplace	Release ISPF 5.8
M - More - Additional IBM Products	

Option:

F1=Help F3=Exit

Abb. 18.2.15
diese mit HATS erstellte Darstellung

18.3 CICS Transaction Gateway

18.3.1 Moderne Oberflächen

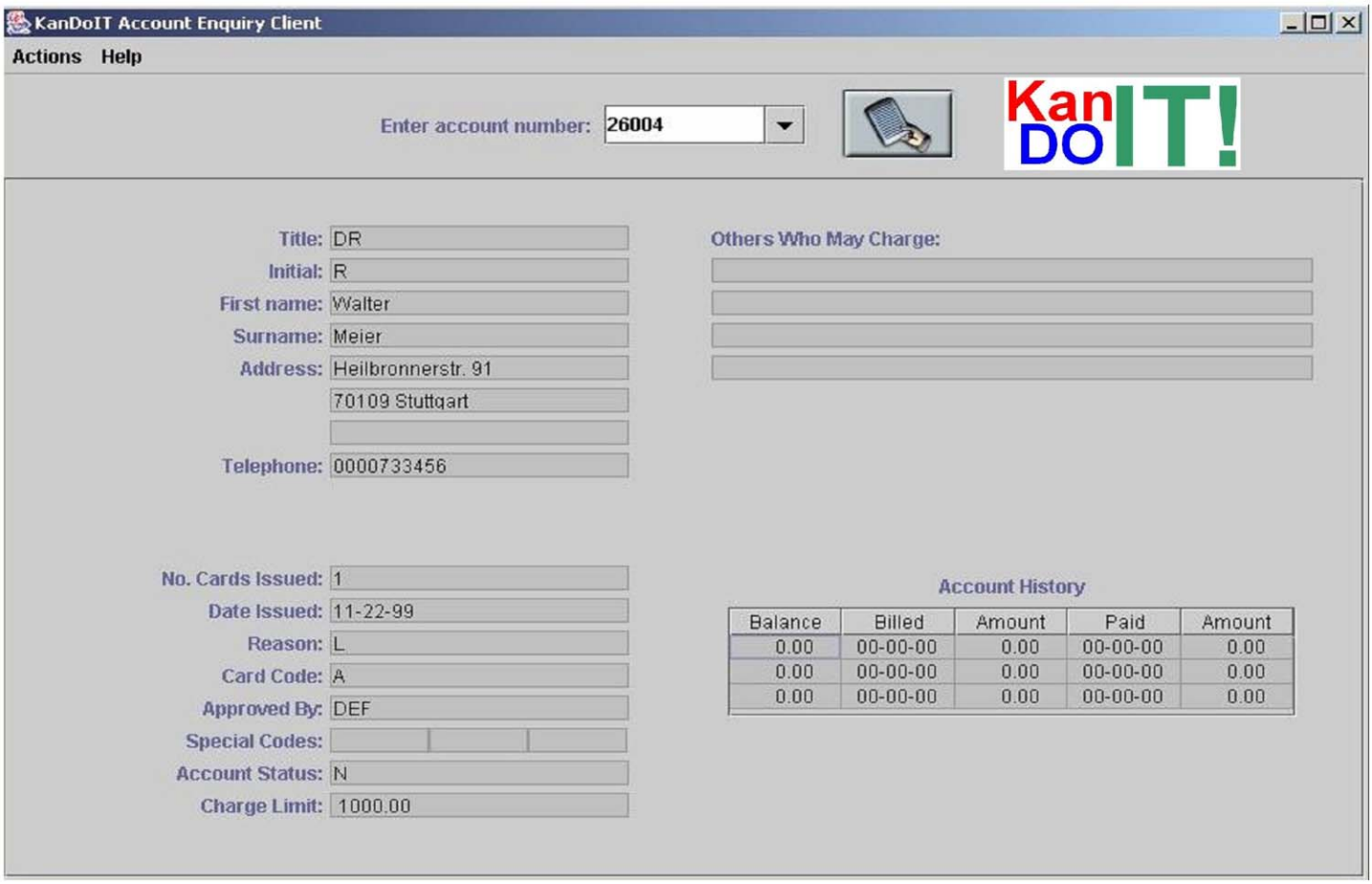


Abb. 18.3.1
Alternative für einen BMS Screen

In traditionellen Fällen benutzen alle Screens das 1971 entstandene „3270 Protokoll“ und die BMS (Basic Mapping Support) Präsentationslogik, welche Bestandteil von jedem CICS TP Monitor ist.

Eine Alternative sind moderne Benutzeroberflächen, die in der großen Mehrzahl der Fälle mit Hilfe von Java programmiert werden. Ein einfaches Beispiel ist in Abb. 18.3.1 wiedergegeben.

In der Wirtschaft und Verwaltung wurden in den letzten Jahren die allermeisten CICS Anwendungen hiermit ausgestattet worden. Dies geschah fast immer als Alternative (und nicht als Ersatz) zu der existierenden BMS Präsentationslogik, die heute immer noch sehr gebräuchlich ist.

18.3.2 COMMAREA als Kommunikationsschnittstelle

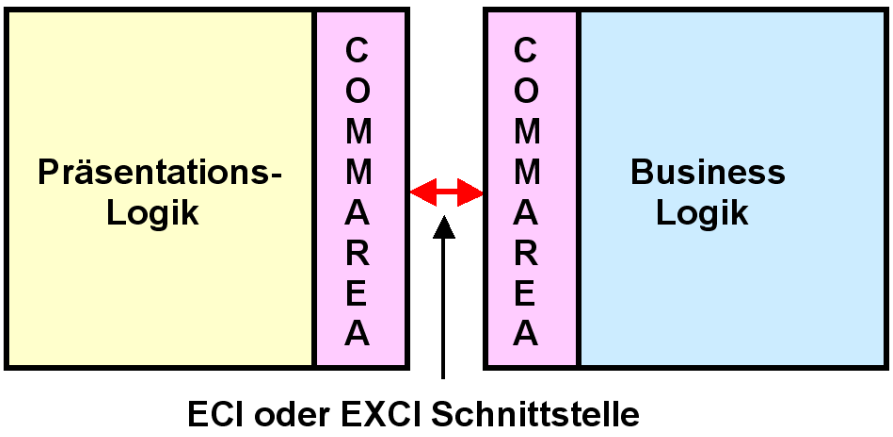


Abb. 18.3.2
Kommunikation über die COMMAREAs von zwei CICS Instanzen

Eine CICS Anwendung besteht aus der Business Logik und der Präsentations- Logik.

Es ist guter Programmierstiel, diese beiden Funktionen voneinander zu trennen und in getrennten Programm Modulen unterzubringen.

Für die Kommunikation zwischen den beiden Modulen wird ein Pufferbereich benötigt. Hierfür bietet sich der COMMAREA Puffer an, dar von der Storage Manager Komponente des CICS Subsystems bereitgestellt wird. COMMAREA ist Bestandteil des Scratchpad - Speicherbereiches der CICS Storage Manager Komponente. Der Scratchpad wird u.a. auch für die Verwaltung von Sessions verwendet, wobei der State einer Transaktion für die Nachfolgetransaktion verfügbar ist.

Für die Kommunikation einer beliebigen Implementierung der Präsentations- Logik mit COMMAREA existiert eine Schnittstelle, die „External Call Interface“ (ECI). Sie verwendet die CICS „Distributed Program Link“ (DPL) Interprocess Communication Einrichtung. DPL ist ein Verfahren ähnlich einem RPC. Ein CICS Programm kann ein anderes CICS Programm mit dem „EXEC CICS LINK (Parameter)“ Befehl aufrufen. Beide Programme können sich auf dem gleichen Rechner befinden, oder über das Netzwerk miteinander kommunizieren.

Befinden sich beide Programme auf dem gleichen z/OS Rechner (oder Sysplex), kann eine als EXCI bezeichnete Variante der ECI Schnittstelle verwendet werden. Diese verwendet Pipes in einem Speicherbereich des z/OS Kernels und vermeidet den Kommunikation Overhead.

18.3.3 CICS Universal Client

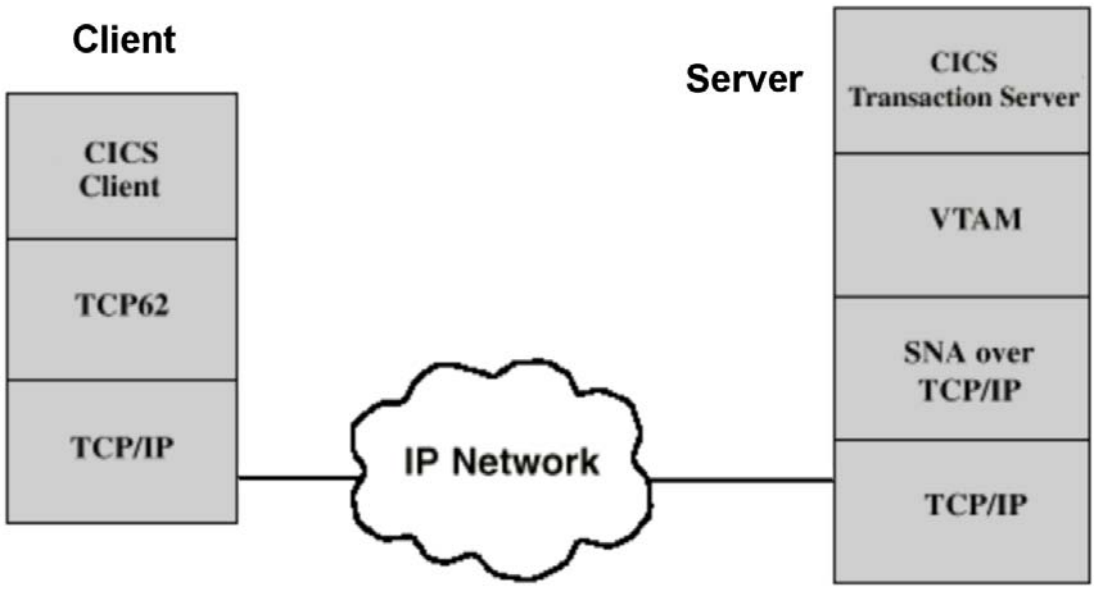


Abb. 18.3.3
Schichtenmodell für die CICS Universal Client Kommunikation

Ein CICS Server kann mit einem anderen CICS Server über eine Peer-to-Peer Verbindung kommunizieren. Dies benutzt die SNA Protokolle APPC, LU 6.2 und TCP62. Die CICS Peer-to-Peer Verbindung erfordert ein CICS System an beiden Communication Endpunkten. Daten werden zwischen den COMMAREAs der beiden CICS Server ausgetauscht.

Nehmen wir an, Sie möchten diese Funktionalität auf Ihrer Workstation haben um mit einem CICS Server zu kommunizieren. Ein Vorteil ist, Sie müssen nicht mit den Einschränkungen des 3270 Protokolls leben. Die Benutzung von DPL und ECI erfordert, dass auf dem Klienten ein funktionsfähiger CICS Transaktionsmonitor installiert wird.

Eine Möglichkeit der Implementierung ist es, einen vollständig en CICS Transaktionsmonitor auf Ihrem Windows oder Linux Arbeitsplatzrechner zu installieren. Dies ist Overkill, wenn Sie lediglich über die TCP62 Peer-to-Peer Funktionalität verfügen wollen. In diesem Fall können Sie eine spezielle Software auf Ihrer Workstation installieren, den "CICS Universal Client", Der CICS Universal Client simuliert ein vollständiges CICS System, verfügt aber lediglich über die Funktionen, die für eine TCP62 Verbindung erforderlich sind.

Der CICS Universal Klient verhält sich bezüglich des DPL Calls wie ein reguläres CICS System, hat sonst aber nur rudimentäre Funktionen.

Spezifisch benutzt der CICS Klient das SNA Protokoll für die Kommunikation mit einem CICS Server (weil auch der heutige CICS Server nur SNA versteht). Die Kommunikation über das Internet und TCP/IP erfolgt über das TCP62 Protokoll.

Heute bezeichnet man den CICS Universal Client oft als „CICS Client".

18.3.4 CICS Universal Client Operation

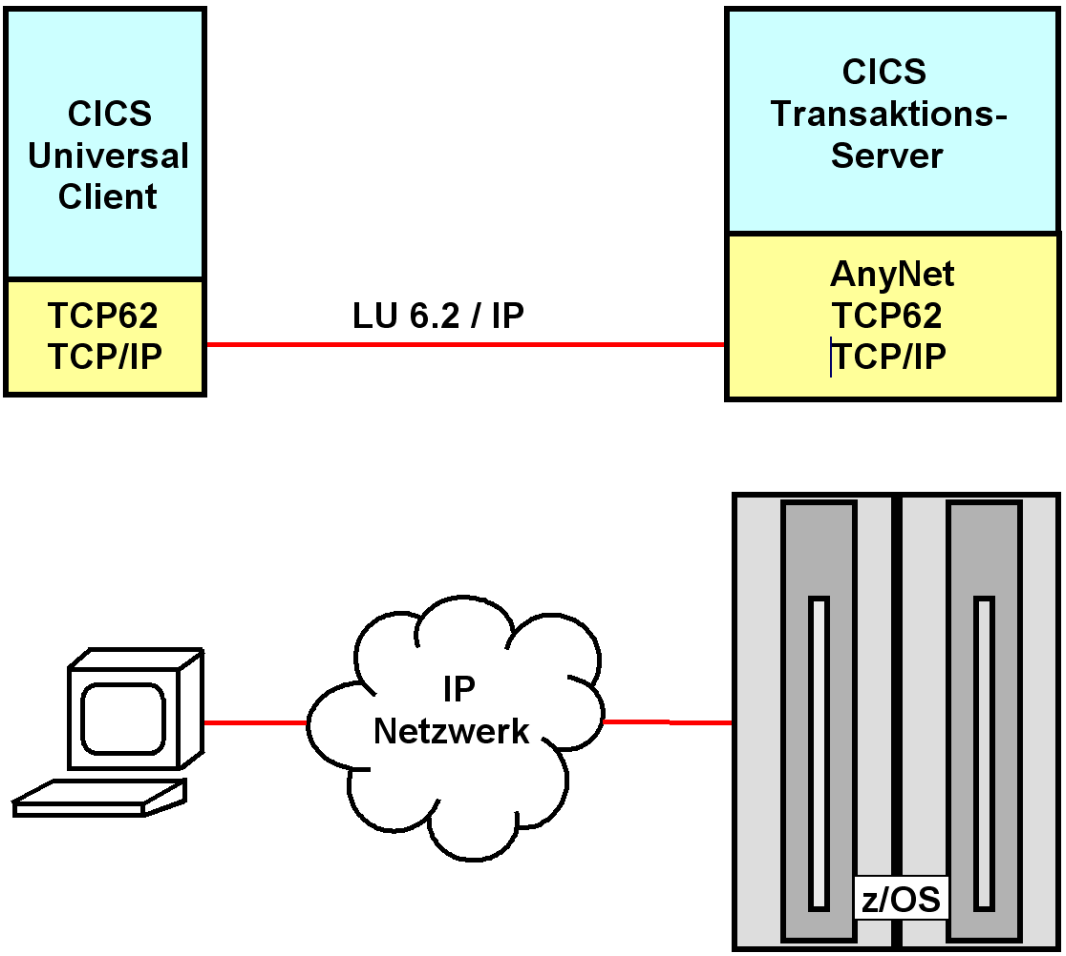


Abb. 18.3.4
Logical Unit Kommunikation

1. Der CICS Universal Client, unter Benutzung der CICSCLI.INI Definitionen, übergibt Daten an die TCP62 Komponente der Workstation.

2. Die TCP62 Komponente auf der Client Workstation verwendet den Domain-Namen Suffix des Partner LU Namens, um einen Internet Protocol (IP) Namen zu generieren. Die IG-Adresse für diesen Namen wird dann entweder von der lokale IP-Hosts-Datei ermittelt, oder von einem Domain Name Server (DNS). Die Daten werden von der TCP/IP-Komponente auf der Workstation an die TCP/IP Komponente des z/OS weiter gereicht

3. TCP/IP auf dem z/OS Host leitet die empfangenen Daten von der Workstation an die SNA über TCP/IP Komponente des z/OS Host weiter.

4. Die SNA über TCP/IP Komponente übersetzt die eingehenden IP-Routing Informationen in SNA Routing-Informationen. Die Daten werden an VTAM übertragen und von dort an den CICS Transaction Server für z/OS weiter geleitet.

18.3.5 CICS Distributed Program Link

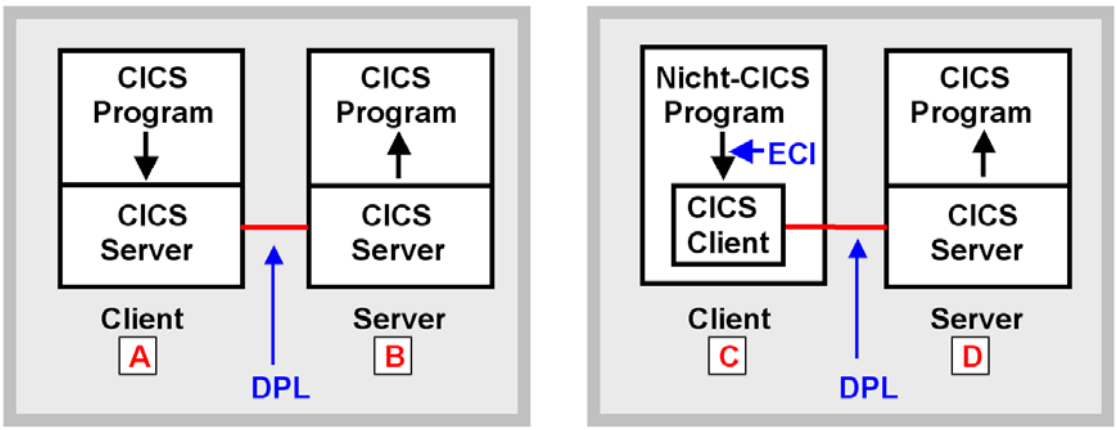


Abb. 18.3.5
CICS Distributed Program Link

Die linke Anordnung zeigt zwei CICS Systeme. Ein Anwendungsprogramm in dem ersten CICS System A ruft ein Anwendungsprogramm in dem zweiten CICS System B mit Hilfe des EXEC CICS LINK Kommandos auf:

```
EXEC CICS LINK PROGRAM(name) COMMAREA(data-area)
```

Das Distributed Program Link (DPL) Protokoll erlaubt es einem Programmierer, das LU 6.2 Link zu benutzen ohne Kenntnis des LU 6.2 Protokolls.

Die rechte Anordnung zeigt einem Client Rechner C, z.B. eine Linux oder Windows Workstation, auf dem der CICS Universal Client installiert ist. Ein nicht-CICS Anwendungsprogramm auf dem Klienten kann sich mit einem CICS Anwendungsprogramm auf dem CICS Server D verbinden, indem es den CICS Client über die ECI Schnittstelle aufruft. Dieser verbindet sich ebenfalls über DPL mit dem CICS Server

18.3. ECI und DPL

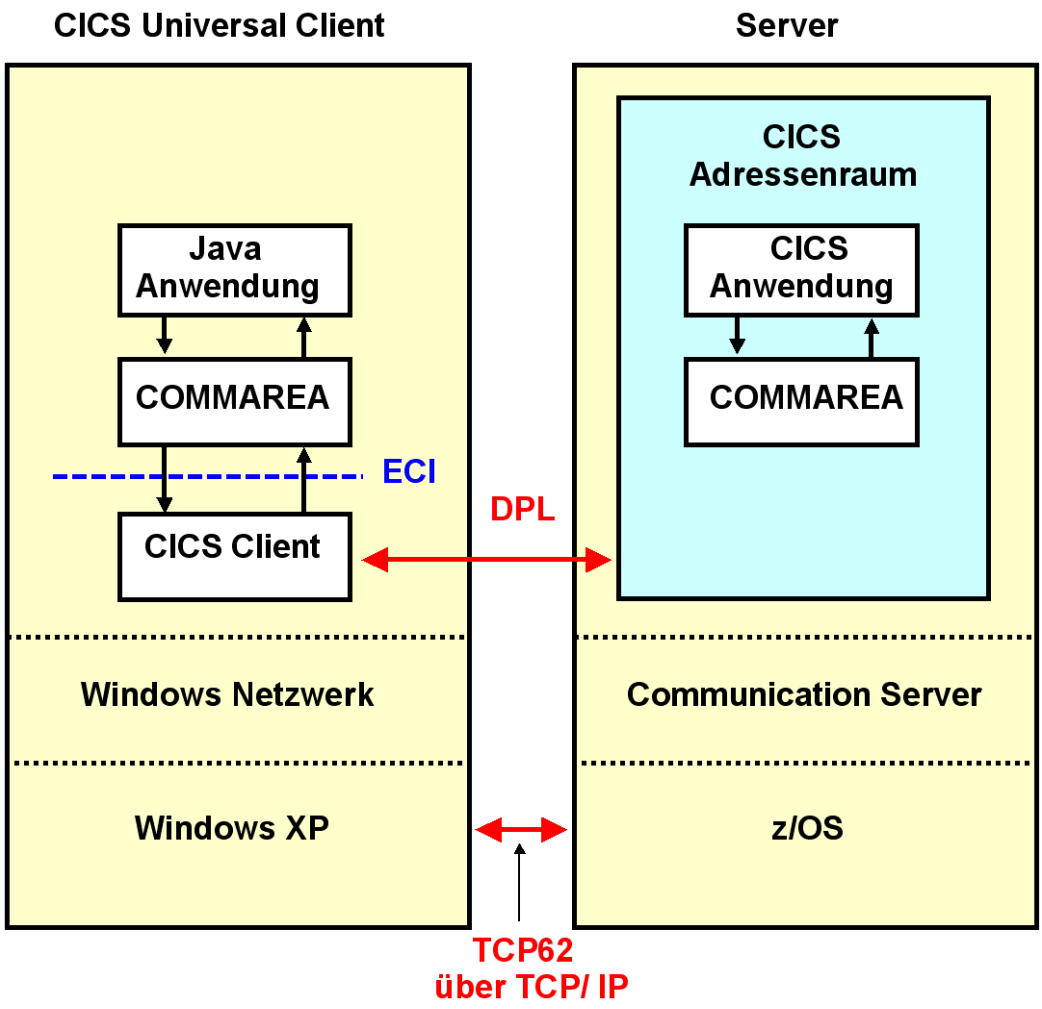


Abb. 18.3.6
CICS Universal Client

Ein CICS Universal Client ist eine echte CICS Anwendung, die mit anderen CICS Anwendungen über Distributed Program Link (DPL, einem RPC ähnlichen Mechanismus des CICS Transaktionsservers) verkehrt.

Eine Java Client Anwendung kann über die ECI Schnittstelle auf den CICS Client zugreifen. Dies ermöglicht einen direkten COMMAREA Datenaustausch zwischen Klienten und Server. Die Beschränkungen des BMS/3270 Datenprotokolls (z.B. keine Scroll Bar) werden damit umgangen.

18.3.6 CICS Transaction Gateway

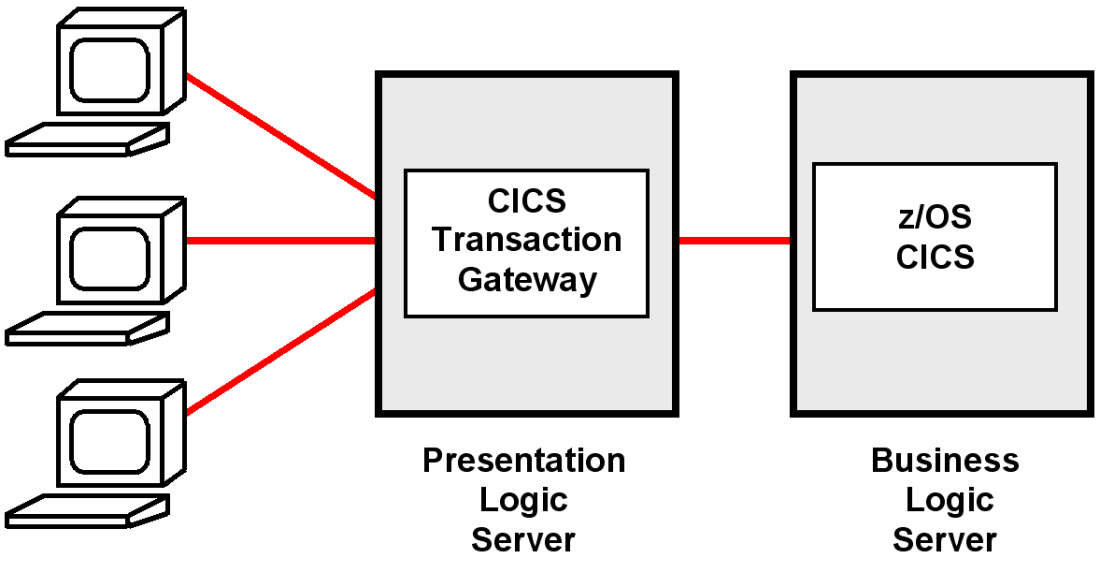


Abb. 18.3.7
Server seitige Präsentationslogik

Der CICS Client hat den Nachteil, dass bei einer Installation mit 10 000 Klienten der CICS Universal Client und die entsprechende ECI Anwendung 10 000 mal administriert und gewartet werden muss.

Deswegen wird man in vielen Fällen statt dessen einen zentralen Presentation Logic Server einsetzen. Hierfür existiert ein entsprechendes Software Product: das CICS Transaction Gateway (CTG).

Das CICS Transaction Gateway ist als EJB implementiert, und läuft normalerweise auf einem WebSphere Application Server. Dieser wiederum läuft entweder auf dem z/OS Rechner unter Unix System Services, oder alternativ auf einem getrennten (distributed) Windows oder Linux Server.

18.3.7 CICS Klienten Anbindung

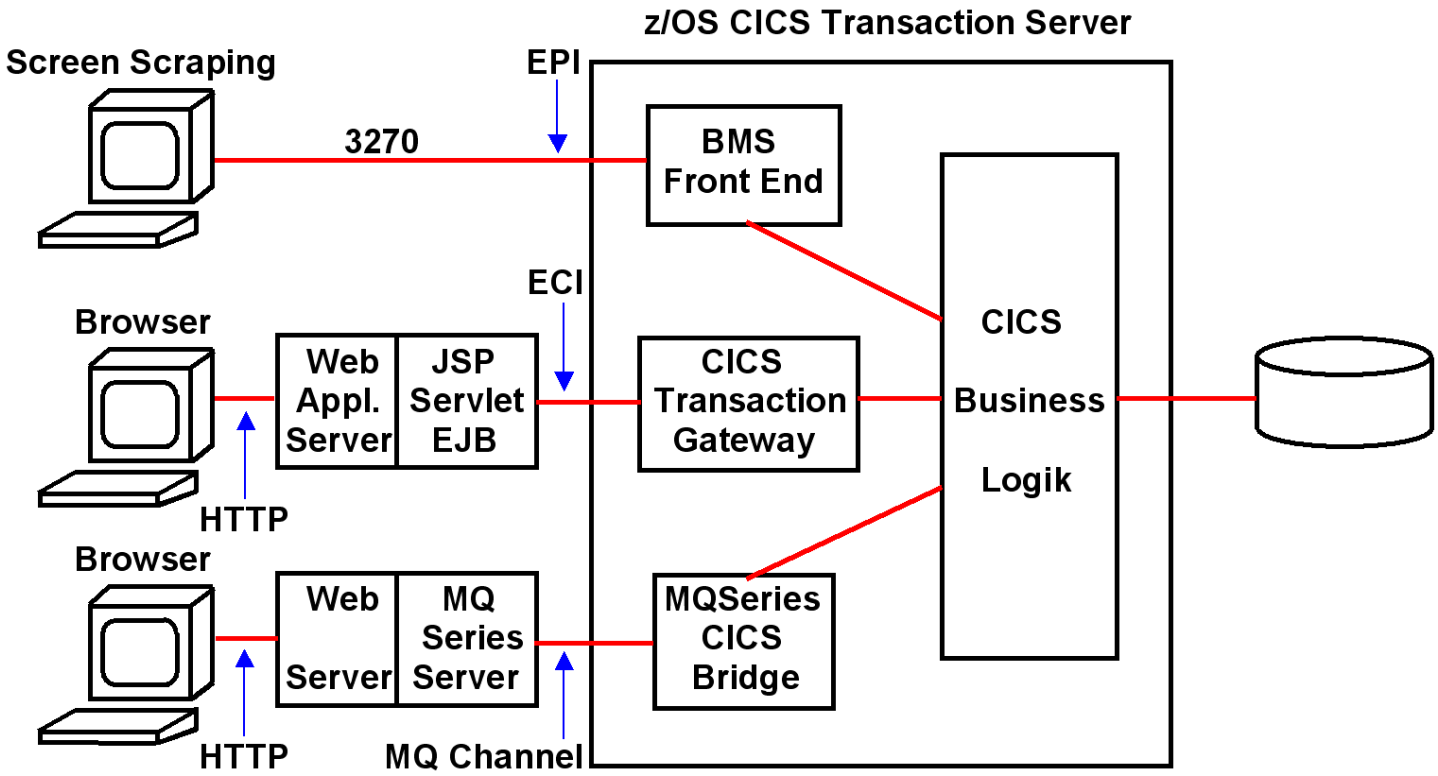


Abb. 18.3.8
Drei populäre Anbindung von CICS Klienten

Das 3270 Protokoll mit der EPI Schnittstelle gestattet nur eine Zeilen-orientierte Ausgabe oder Screen Scraping. Die BMS Maps werden weiter verwendet. Keine Änderung der Information, die auf dem Bildschirm wiedergegeben wird. Die Darstellung der Information kann geändert werden.

Bei einer uneingeschränkten grafischen Ausgabe wird die Presentation Service Komponente von CICS (BMS) wird nicht genutzt. In dem hier gezeigten Beispiel erfolgt ein direkter Zugriff auf COMMAREA. Die Presentation Service Komponente von CICS (BMS) wird nicht genutzt.

18.3.8 Was wird heute eingesetzt ?

Es existieren Dutzende von Alternativen, um CICS Anwendungen in das Internet zu integrieren, für die CICS über die entsprechende Unterstützung verfügt. Eine ganze Reihe von Lösungen werden von IBM geliefert, viele andere von unabhängigen Herstellern (third party vendors). Ein Beispiel eines unabhängigen Herstellers ist die Firma Attachmate.

Drei Alternativen sind besonders bedeutend; sie sind in Abb. 18.3.8 gezeigt.

Besucht man ein heutiges Unternehmen, so stellt man fest, dass häufig für die gleiche CICS Anwendung mehrere unterschiedliche Präsentations-Logiken verfügbar sind. Eine populäre Möglichkeit ist die asynchrone pseudoconversationale Verarbeitung unter Benutzung von MQSeries und der CICS MQSeries Bridge.

Grafische Oberflächen – teilweise Browser basiert – sind sehr populär. Viele – nicht alle – werden in Java mit Servlets und JSPs implementiert. Hier spielt das CICS Transaction Gateway (CTG) eine bedeutende Rolle.

Ein überraschend großer Anteil der CICS Anwender benutzen nach wie vor die 3270 Oberfläche. Der Grund ist, die zeilenorientierte 3270 Oberfläche ist produktiver als eine grafische Oberfläche. Sie braucht allerdings mehr Einarbeitungszeit.

18.4 JCA

18.4.1 Enterprise Information System

„Enterprise Information System“ (EIS) ist ein Begriff, der vor allem von JEE Standards verwendet wird. Ein EIS ist existierende Middleware, auf die mit Hilfe von Java Komponenten, besonders EJBs, zugegriffen wird. EIS Beispiele sind Transaktionsmonitore wie SAP R/3, CICS, IMS/DC, Tuxedo, Datenbanksysteme wie DB2, IMS/DB, Oracle, Adabas oder Message based Queuing Systeme wie MQSeries.

Die meisten EIS sind reinrassige Middleware; der Hersteller eines EIS nimmt an, dass ein Benutzer seine eigenen Anwendungen für das EIS schreibt. Manche Hersteller von EIS liefern umfangreiche betriebswirtschaftliche Anwendungen mit ihrem EIS aus. Diese EIS werden als ERP (Enterprise Resource Planning) Systeme bezeichnet. Ein Beispiel ist SAP R/3. Oracle vertreibt neben seiner relationalen Datenbank ebenfalls mehrere ERP Systeme, darunter Oracle Enterprise One, oder als direkten SAP Konkurrenten die Oracle E-Business Suite.

Kommentar zum Namen SAP R/3: Auch die Firma SAP liebt es, die Namen zu ändern. Bis Dezember 2003 wurde das Produkt unter dem Namen SAP R/3 geführt, bis 2007 unter dem Namen mySAP ERP. Die derzeitige Version heißt R/3 Enterprise oder SAP ERP.

Es existieren viele Anwendungen auf unabhängigen Anwendungsservern und viele unterschiedliche Enterprise Information Systems (z.B. CICS, SAP, IMS, MQSeries, DB2, ...). Aufgrund der großen Anzahl von unterschiedlichen Enterprise Information Systemen (EIS) ist die Lösung der Integrationsfrage sehr komplex. Um aus einer Java Anwendung heraus auf Informationen eines EIS zuzugreifen, war bisher notwendig, eine anwendungsspezifische Verbindung zu programmieren. Die Folge war ein hoher Entwicklungsaufwand, welcher mit der Anzahl unterschiedlicher EIS-Systeme und Anwendungsserver (z.B. WebLogic oder WebSphere) wuchs, da für jeden Anwendungsserver eine Verbindung zu jedem EIS hergestellt werden muss. Bei m Anwendungsservern und n EIS-Systemen bedeutet dies einen Aufwand von "m * n".

Durch die Java Connector Architektur (JCA) soll dieser Aufwand reduziert werden.

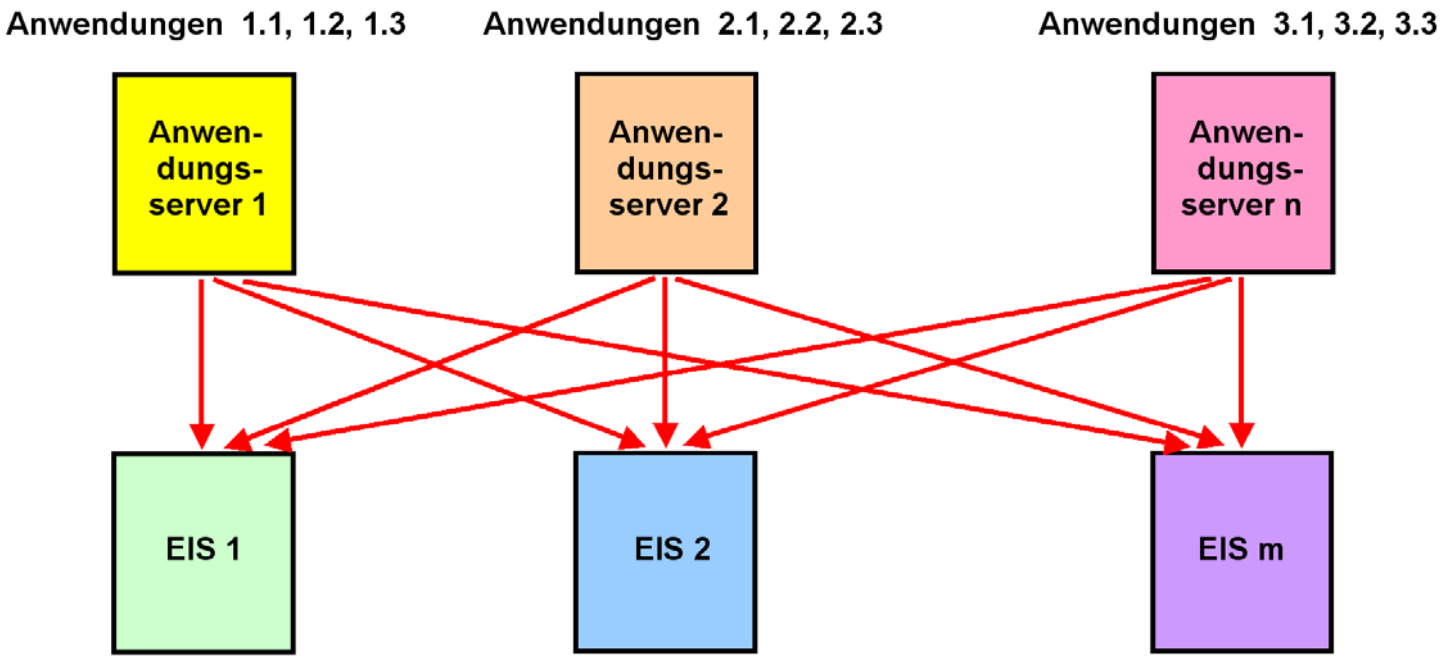


Abb. 18.4.1
Verbindungsvielfalt zwischen Anwendungsservern und EIS Komponenten

Jeder Pfeil stellt eine personalisierte Verbindung zwischen einem Anwendungsserver und einem Enterprise Information System (EIS) dar.

18.4.2 Konnektoren

Bei den Implementierungen von JEE Anwendungen spricht man von der Präsentationslogik (Frontend), welches typischerweise mit einem Web Application Server realisiert wird, und der Businesslogik (Backend, Beispiele: Auftragseingang, Finanzbuchhaltung), wofür häufig vorhandene Systeme wie CICS oder DB2 eingesetzt werden. Zusätzlich werden Erweiterungen als EJBs implementiert.

In vielen Fällen werden 20 % des Projektaufwandes für die Neuentwicklung des Frontends und 80% für dessen Integration in das vorhandene Backend aufgewendet.

Konnektoren sind in beliebigen Sprachen geschriebene Software Komponenten, welche eine Schnittstelle zu existierenden Enterprise Information (Legacy) Systemen bilden. Die meisten Konnektoren werden heute als EJBs erstellt und eingesetzt. Beispielsweise sind folgende EJB Konnektoren für den IBM WebSphere Application Server für die folgende Middleware Software verfügbar:

- **JDBC, SQLJ**
- **DB2**
- **Oracle**
- **Adabas**
- **CICS**
- **IMS**
- **MQSeries**
- **SAP R/3**
- **Lotus Domino**

} **Enterprise Information Systeme**

18.4.3 JEE Connector Architecture

Die Java EE Connector Architecture (JCA) ist eine Software-Architektur und Programmier-Schnittstelle (API) zur Integration von heterogenen Anwendungen und EIS in die JEE Platform.

JCA ist eine Standard Architektur für die Integration von existierenden Business Logik Komponenten .

- **ERP Systeme, z.B. SAP R/3**
- **Mainframe Transaktions- Monitore, z.B. CICS, IMS**
- **Non- Java Legacy Anwendungen**
- **Datenbank Systeme**

} **vom JCA Standard als Enterprise Information Systeme (EIS) bezeichnet**

Wichtigste Bestandteile der JEE Connector Architecture (JCA) sind:

- **JCA Konnektoren, als Resource Adapter (RA) bezeichnet**
- **Common Client Interface (CCI)**

A Resource Adapter is a system level software library that is used by an application server or client to connect to a Resource Manager. A Resource Adapter is typically specific to a Resource Manager. It is used within the address space of the client using it. An example of a resource adapter is the JDBC driver to connect to relational databases.

Die Common Client Interface (CCI) ist eine universelle API für die Interaktion mit beliebigen Resource Adaptoren. Sie ist weitgehend unabhängig von den spezifischen Eigenschaften des über einen JCA Connector verbundenen EIS.

Weiterhin spezifiziert der JCA Standard eine System Level Programming Interface (Service Provider Interfaces, SPI)). Hiermit werden Dienste wie connection pooling, Resource Adapter Deployment und Packaging definiert. Darüber hinaus enthält die JCA noch eine API für lokale Transaktions- Demarkationen. Für CICS existieren:

- **JCA ECI Resource Adapter (für einen COMMAREA Zugriff),**
- **JCA EPI resource adapter.**

Das CICS Transaction Gateway (CTG) ist ein JCA Standard konformer JCA Adapter. Er enthält ECI Resource Adapter, EPI Resource Adapter und CCI, sowie weitere Zugriffsmechanismen außerhalb der JCA.

18.4.4 Unterschiedliche Connector Arten

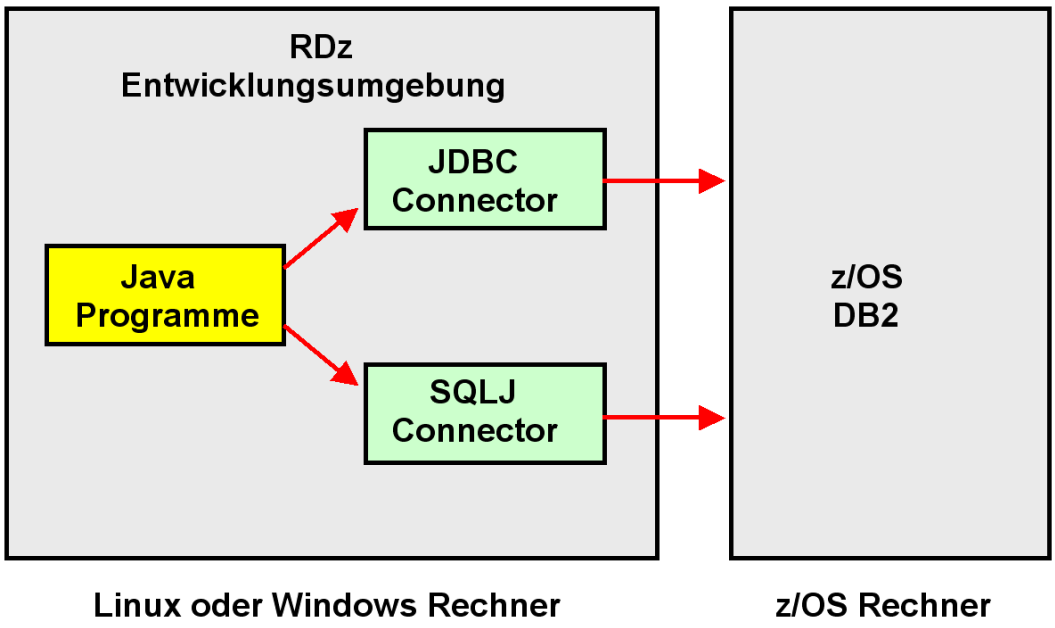


Abb. 18.4.2
JDBC und SQLJ Konnektoren

Es existieren viele unterschiedliche Konnektor Arten. Beispielsweise sind für den DB2 Zugriff JDBC und SQLJ Konnektoren vorhanden. JDBC ist von ODBC abgeleitet, und implementiert einen dynamischen Datenbankzugriff. SQLJ implementiert statische Datenbankzugriffe.

DB2Connect ist ein für die DB2 API optimierter Connector für in beliebige Programmiersprachen implementierte Klienten. Die Java Version implementiert den JCA Standard. Das derzeitige DB2Connect Software Paket beinhaltet mehrere DB2 Konnektoren, spezifisch auch JDBC und SQLJ.

18.4.5 JCA Struktur

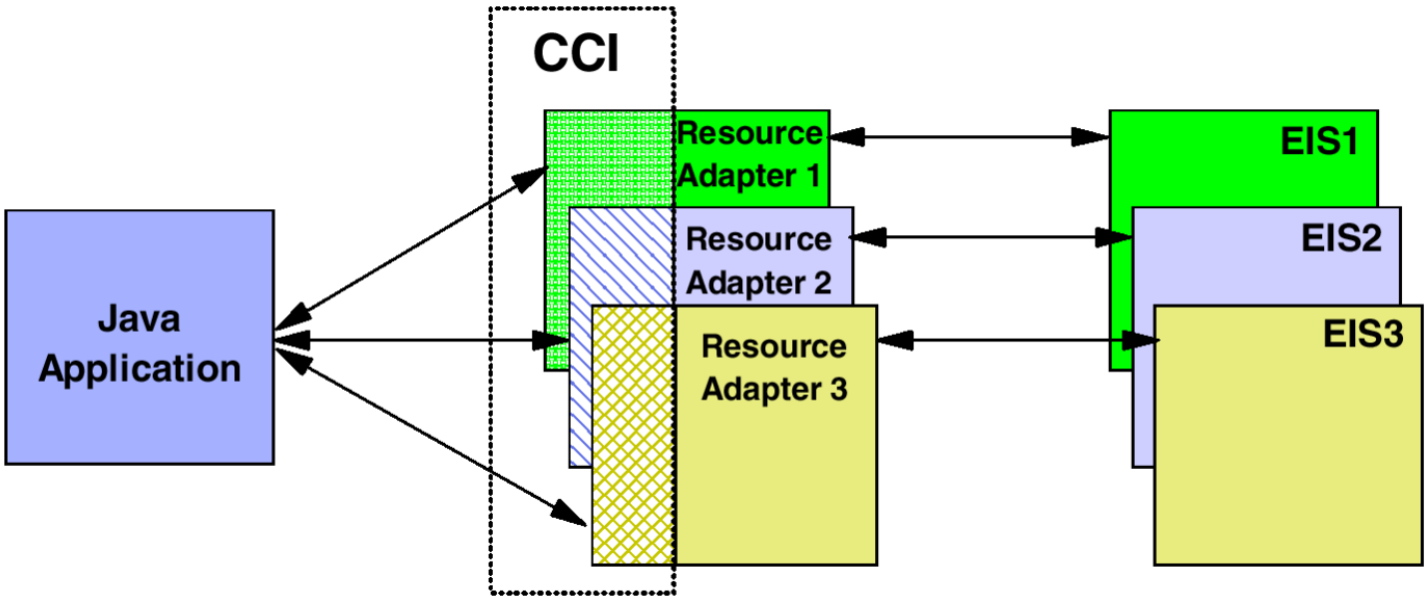


Abb. 18.4.3
Komponenten der JEE Connector Architecture

Der JCA Standard benutzt die folgenden Begriffe:

- **CCI Common Client Interface**
- **EIS Enterprise Information System**

CCI und Resource Adapter sind als Java Klassen implementiert. Sie können von einer Java Anwendung alleinstehend (ohne Benutzung eines JEE Servers) benutzt werden (non-managed environment).

Üblich ist es, CCI und Resource Adapter als Elemente eines JEE Web Application Servers einzusetzen (managed environment). Hierbei kann der Web Application Server das Management von Verbindungen, Transaktionen und Sicherheit direkt übernehmen. Die CCI Entwicklung kann in ein Entwicklungswerkzeug wie Eclipse integriert werden.

18.4.6 Common Client Interface

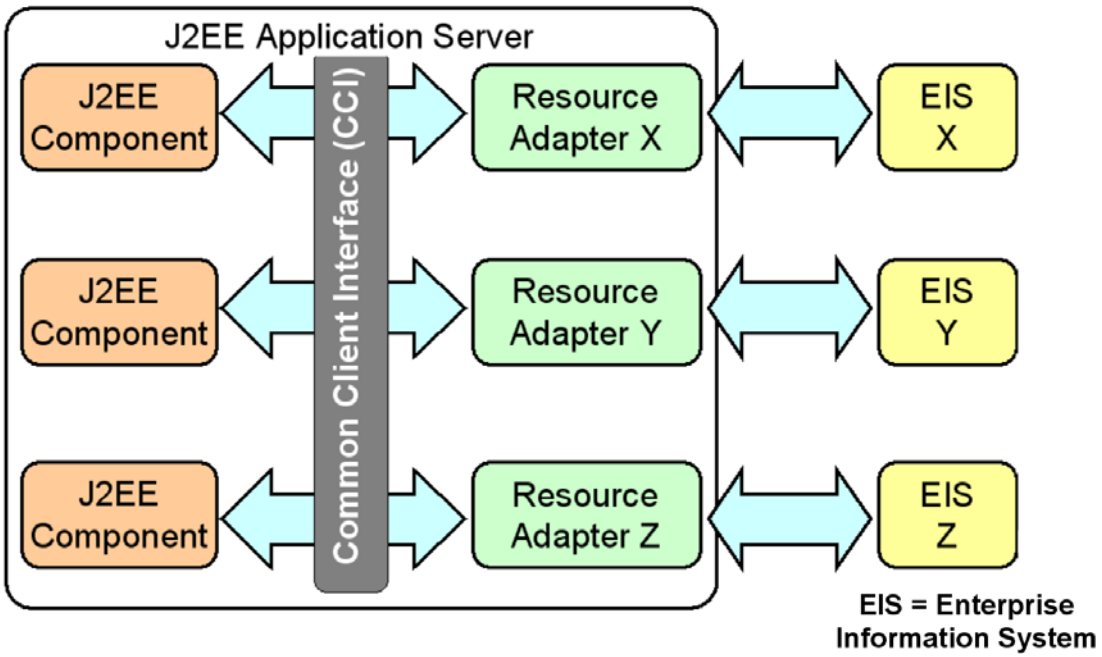


Abb. 18.4.4
Identische Common Client Interface für unterschiedliche Java Anwendungen

Die identische JCA Common Client Interface (CCI) wird von vielen verschiedenen Java Anwendungen verwendet. Die CCI bietet Schnittstellen zu vielen verschiedenen Resource Adapter (RA) an. Ein RA ist eine Schnittstelle zu einer bestimmten Business Logic Middleware. In der JCA Standard Dokumentation wird die Business Logik Middleware als Enterprise Information System (EIS) bezeichnet.

Resource Adapter werden von den Herstellern von Standard Middleware Komponenten zur Verfügung gestellt, z. B. von Oracle, IBM, SAP, und vielen anderen.

18.4.7 CICS als Enterprise Information System

CICS hat eine wachsende Bedeutung. Es werden jedes Jahr mehr CICS Transaktionen als im Vorjahr ausgeführt, und dieser Trend existiert seit 35 Jahren. Die Benutzer tätigen jedes Jahr bedeutende Investitionen in CICS, und auch in MQSeries.

Der WebSphere Application Server hat einen offensichtlichen Wert als Platform für neue JEE-Anwendungen. Die Kombination von CICS, WebSphere MQ und WebSphere Application Server bildet ein sehr leistungsfähiges Ökosystem für die Ausführung von unternehmenskritischen Anwendungen.

Heute ist es sehr üblich, WAS (WebSphere Application Server) Anwendungen zu schreiben, die CICS-Transaktionen auslösen. Das ist ein Umfeld, in dem CICS als Partner mit WAS agiert, wobei die Transaktion von WAS ausgelöst wird.

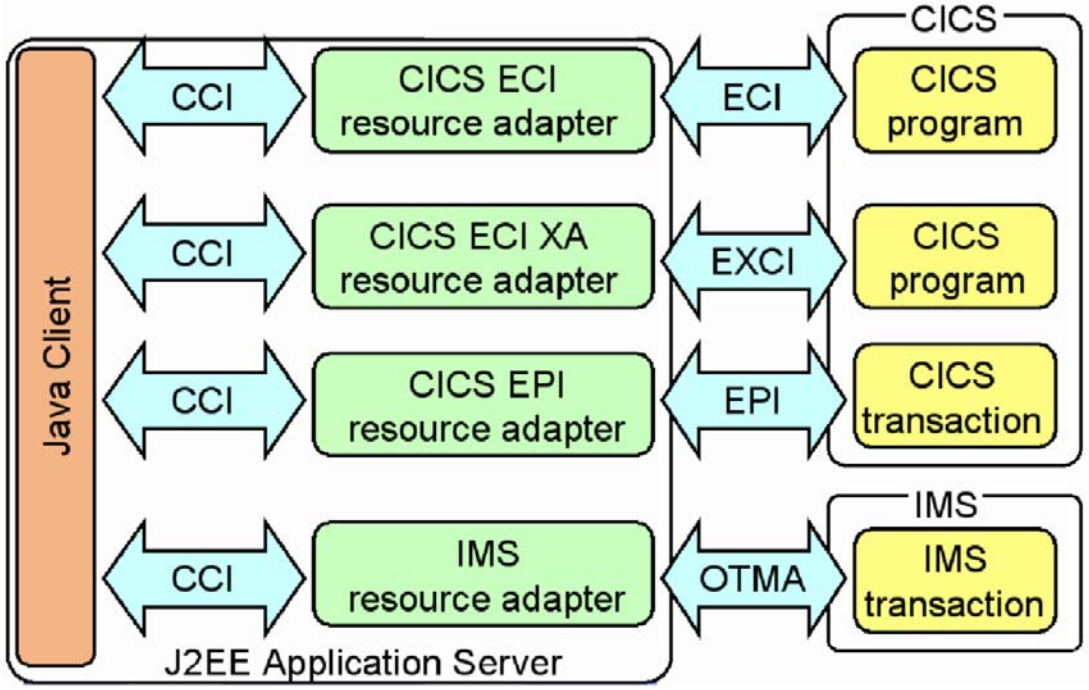


Abb. 18.4.5
Identische CCI für unterschiedliche Resource Adapter

Als Beispiel sind 4 verschiedene Resource Adaptoren (RA) dargestellt, die für IBM Mainframes verfügbar sind. Der CICS ECI und EPI Resource Adapter wurde bereits diskutiert.

Der EXCI Resource Adapter ist eine spezielle Version des ECI. Der ECI Adapter wird verwendet, wenn der JCA Resource Adapter auf einem entfernten System installiert ist. Der EXCI Adapter wird verwendet, wenn der JCA Resource Adapter und CICS auf dem gleichen Mainframe installiert sind. In der Regel läuft der EXCI Resource Adapter unter USS.

Die OTMA Adapter hat die gleiche Funktion für IMS/DC, welche die ECI und EXCI Adapter für CICS haben.

Beim Wechsel zu einer anderen Ressource-Adapter sind aufgrund der CCI keine Änderungen im Java-Client-Code erforderlich.

18.4.8 JCA Benutzung des CICS Transaction Gateways

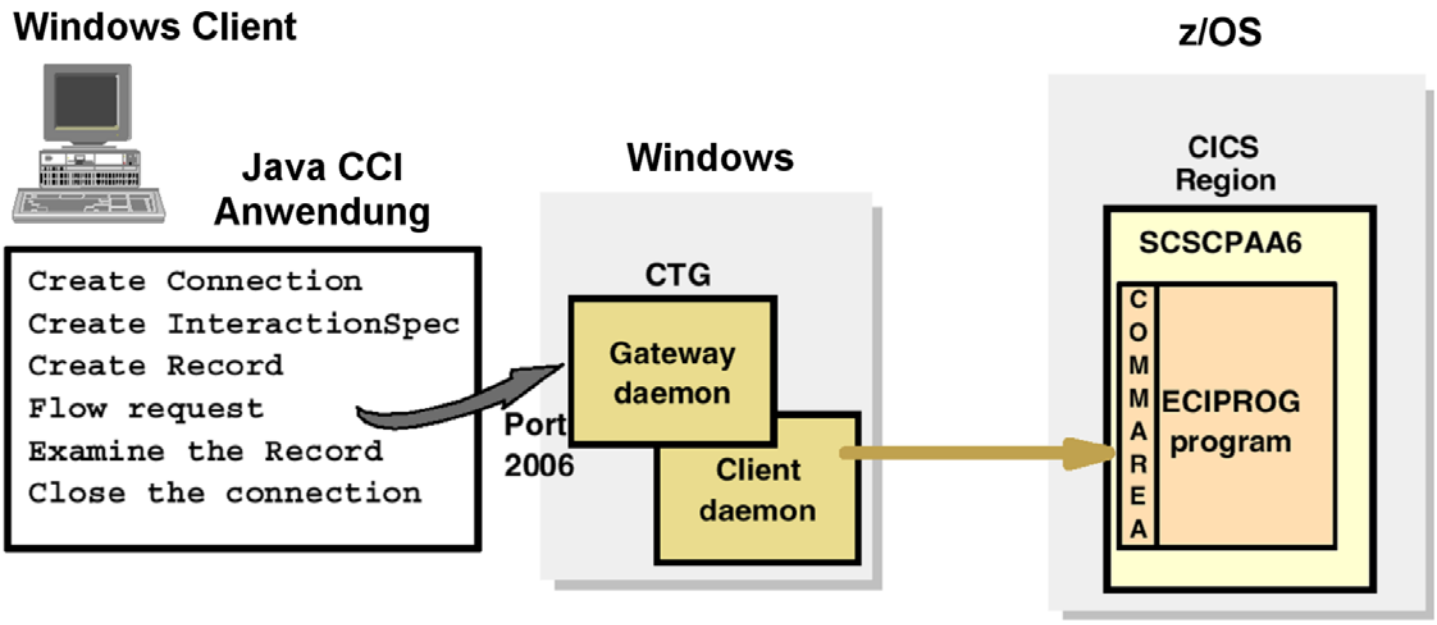


Abb. 18.4.6
CCI Skelett für das CICS Transaction Gateway

Abb. 18.4.6 zeigt CCI Aufrufe einer Java Client Anwendung und ihre CCI Connection zum CICS Transaction Gateway.

18.4.9 CCI Beispiel für die CICS ECI

```
import javax.resource.cci.*;
import com.ibm.connector2.cics.*;

//create a connection
ECIConnectionSpec connSpec = new ECIConnectionSpec();
connSpec.setUserName("CICSUSER");
Connection conn = connfac.getConnection(connSpec);

//Create an interaction
Interaction inter = conn.createInteraction();
ECIInteractionSpec iSpec = new ECIInteractionSpec();
iSpec.setFunctionName("CICSPROG");

//Create a record
MyRecord record = new MyRecord("Data to send to CICS");

//Perform the interaction with CICS
inter.execute(iSpec, record, record);

//Display contents of output record
System.out.println(record.getContents());
```

Abb. 18.4.7
CCI Code für eine Verbindung zu CICS

Gezeigt ist das Skelett einer Java Client Anwendung, welche die CCI verwendet.

18.4.10 CICS Transaction Gateway

Abb. 18.4.8 zeigt einen z/OS-Server, auf dem ein WebSphere Web Application Server mit einer Servlet-Engine und einer EJB-Engine für ein CTG, sowie ein CICS Server für die Geschäftslogik installiert ist.

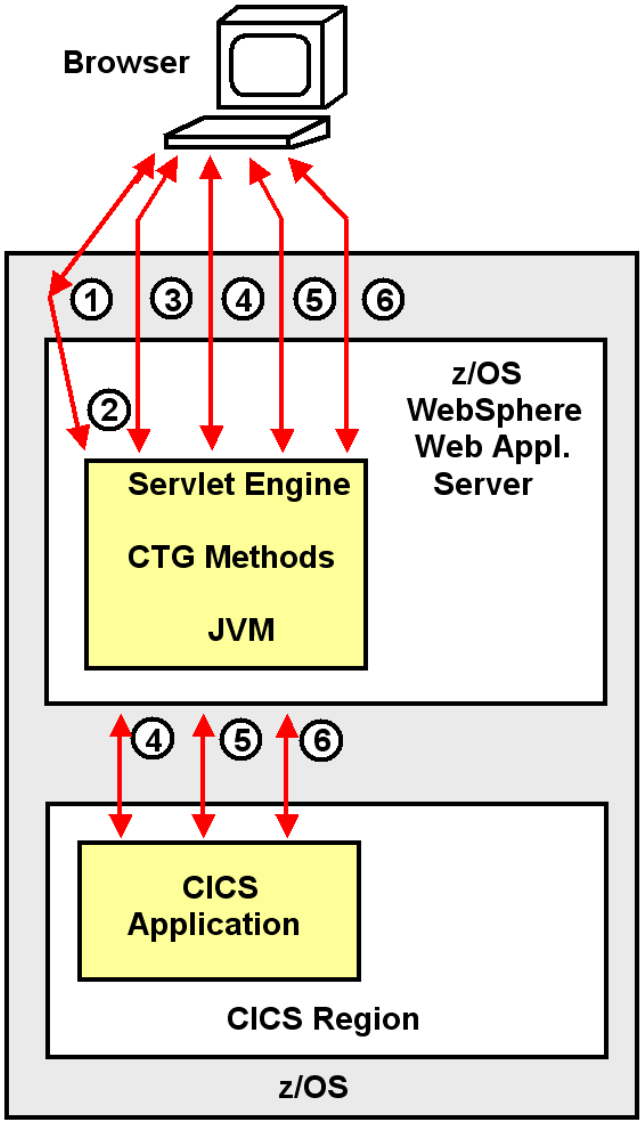


Abb. 18.4.8
Schritte für die Benutzung des CICS Transaction Gateway

Schritte 1 und2: **Der Browser stellt eine Verbindung zu dem Websphere Web Application Server her.**

Schritte 3 bis 6 **Jeder weitere Schritt benutzt Servlets für die Präsentations-Logik und das CICS Transaction Gateway für den Aufruf der CICS Business Logik.**

18.4.11 Versionen des CTG

Das CICS Transaction Gateway (CTG) wird häufig als der Java Enterprise Edition (JEE)-Konnektor für CICS bezeichnet, da es den Zugang zu CICS innerhalb einer JEE Umgebung wie dem IBM WebSphere Application Server (WAS) unterstützt. Allerdings unterstützt das CTG Application Programming Interfaces (API), Platformen und Topologien, die über JEE hinausgehen. Das CTG offeriert CICS-Konnektivität zu C, C++, COBOL und Microsoft DotNet-Anwendungen sowie zu Java Standard Edition (JSE)-Anwendungen.

CTG ist in zwei Varianten erhältlich, um verschiedenen Anforderungen und Einsatzszenarien unterzubringen.

- **Als eine Variante bietet die Multiplatform (distributed) CTG Version einen Multi-User-Zugriff auf CICS aus einer UNIX-, Linux- oder Windows-Umgebung.**
- **Die CTG for z/OS Version offeriert CICS-Konnektivität mit der Skalierbarkeit und den Performance-Vorteilen beim Betrieb in enger Proximity mit dem CICS Transaction Server.**

Für die distributed und die z/OS Version des CTG existieren drei unterschiedliche Topologien, mit denen ein Zugriff auf einen z/OS CICS Transaction Server erfolgen kann.

- **Topologie 1: WebSphere Application Server und CICS Transaction Gateway sind beide auf einer distributed (non-zSeries) Plattform installiert.**
- **Topologie 2: WebSphere Application Server und CICS Transaction Gateway sind beide auf einem z/OS Server installiert.**
- **Topologie 3: WebSphere Application Server und CICS Transaction Gateway sind beide unter zLinux auf dem gleichen physischen Rechner wie z/OS und CICS installiert.**

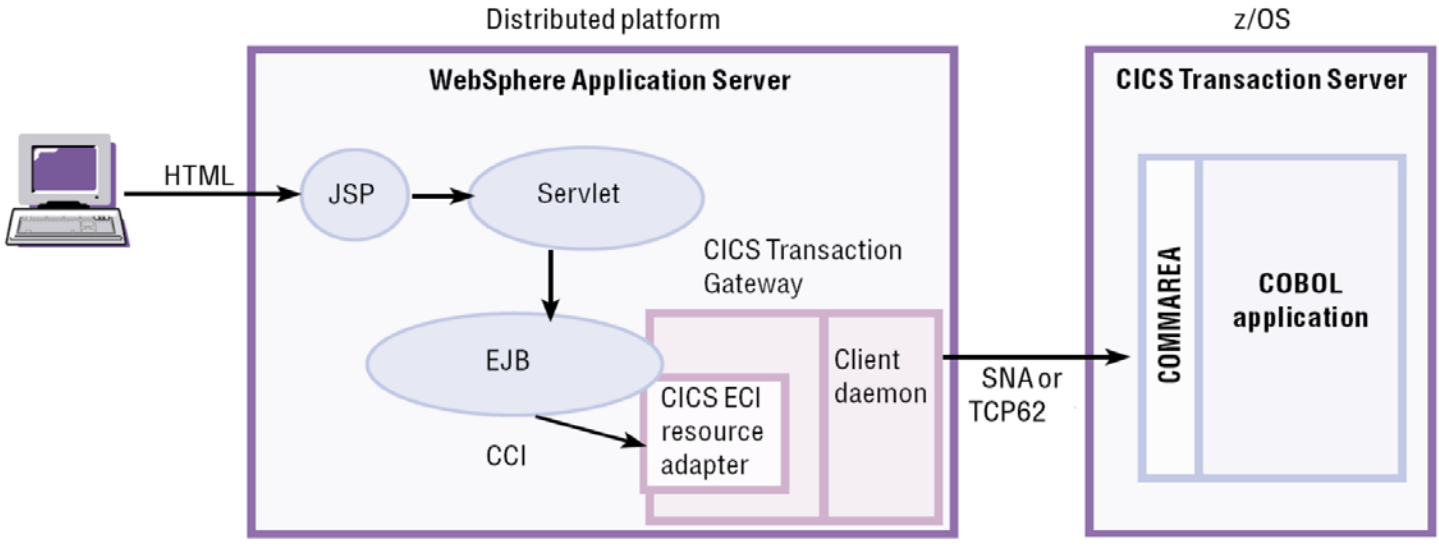


Abb. 18.4.9
CICS Transaction Gateway Topologie 1

In dieser Topologie (3-Tier Konfiguation) sind der WebSphere Application Server und das CICS Transaction Gateway auf einer distributed Plattform installiert, z.B. Windows oderr UNIX. Sowohl ECI als auch EPI (nicht gezeigt) Resource Adapter können benutzt werden. Der CICS Universal Client ist Bestandteil des CTG.

Abb. 18.4.9 zeigt eine Enterprise Java Beans (EJB) Anwendung, die den ECI Resource Adapter benutzt um auf eine COMMAREA-basierte CICS COBOL Anwendung zuzugreifen.

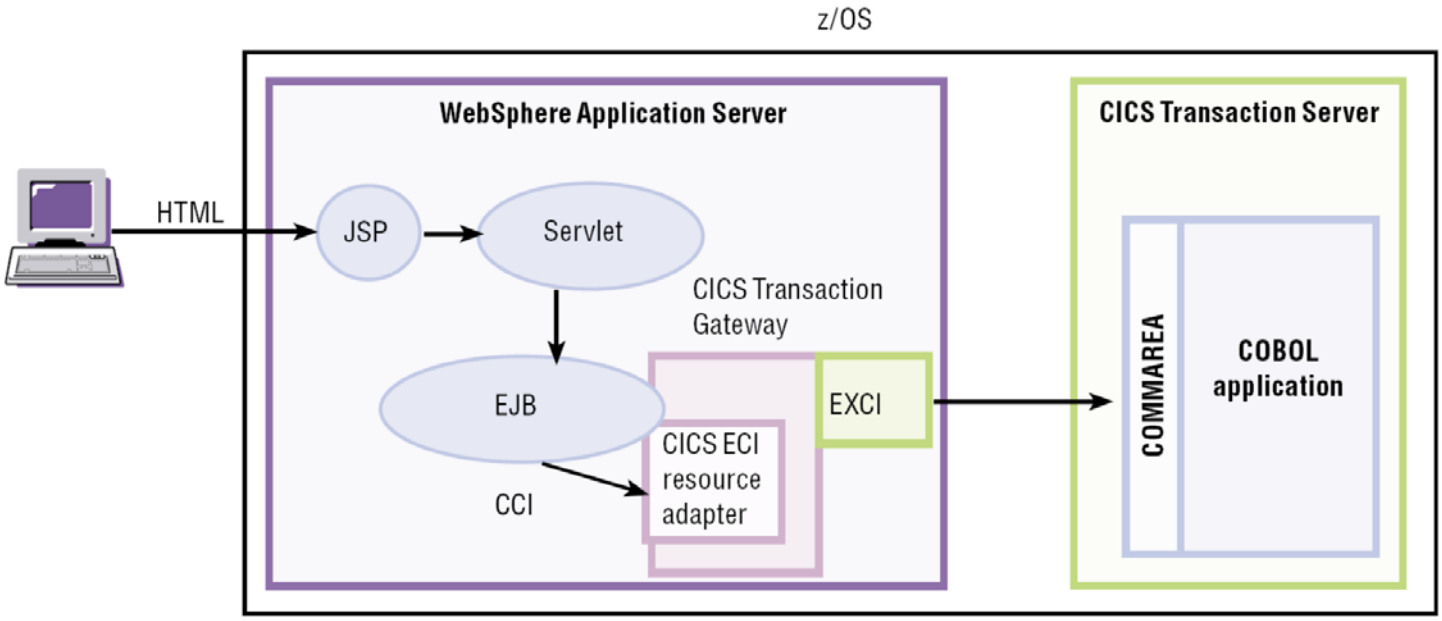


Abb. 18.4.10
CICS Transaction Gateway Topologie 2

In der in Abb. 18.4.10 dargestellten 2-Tier Topologie ist der WebSphere Application Server unter dem gleichen z/OS wie CICS installiert, allerdings in getrennten Address Spaces. Das CTG ist Teil des WAS. Es werden nur CICS ECI Resource Adapters unterstützt. Unter z/OS ermöglicht dies eine direkte Cross-Memory EXCI Verbindung zwischen dem WebSphere Application Server und CICS.

EXCI ist eine Komponente des CICS Transaction Server für z/OS. EXCI ermöglicht es einem z/OS non-CICS Region (Address Space) auf Programme innerhalb einer CICS Transaction Server Region zuzugreifen.

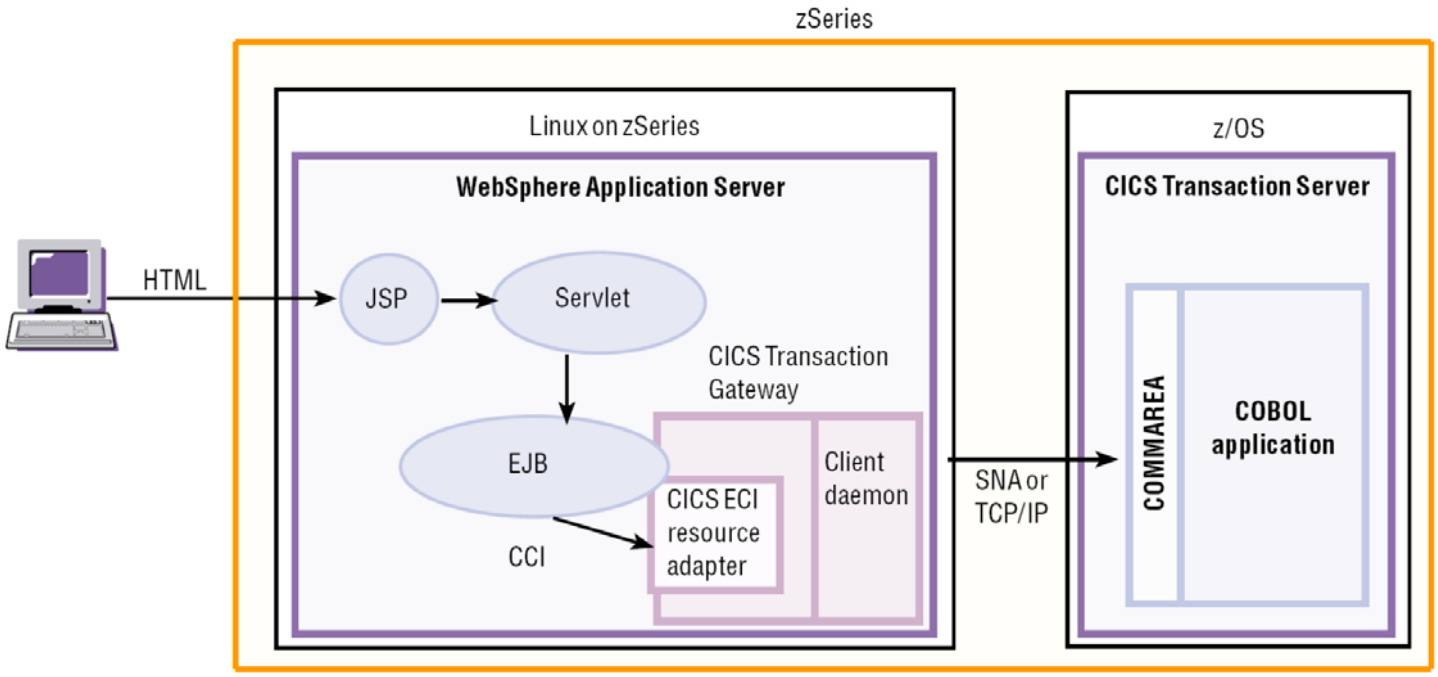


Abb. 18.4.11
CICS Transaction Gateway Topologie 3

Eine weitere Alternative ist es, den WebSphere Application Server auf zLinux zu installieren, und zwar auf dem gleichen System z Rechner auf dem auch z/OS CICS installiert ist, siehe Abb. 18.4.11. Im Vergleich zu Topologie 1 (Abb. 18.4.9) erlauben Hipersockets die Nutzung einer Reihe weiterer Funktionen.

18.4.12 Java Kryptografie Architektur

Beim Öffnen einer z/OS Installation für das Internet hat Sicherheit eine stark wachsende Bedeutung. Kryptografie ist ein leistungsfähiges Werkzeug um die Internet-Sicherheit zu verbessern.

Die Java Kryptografie Architektur definiert die Abstimmung zwischen der Anwendung, dem Connector und dem Anwendungsserver, auf dem die Anwendung bereitgestellt wird.

Die System z Hardware verfügt über Kryptographie Prozessoren und Beschleuniger, welche die Kryptographie Verarbeitungszeit deutlich verbessern.

Security ist ein Problem in allen großen Installationen. System z und z/OS haben deutlich weitergehende Sicherheitseigenschaften als alle anderen Platformen. Die Topologien 2 und 3 in Abb. 18.4.10 und 18.4.11 sind deshalb unter Sicherheitsgesichtspunkten vorteilhaft gegenüber Topologie 1 in Abb. 18.4.9 .

Anmerkung: JCA bedeutet alternativ JEE Connector Architecture und Java Cryptography Architecture.

18.5 Weiterführende Information

Vielleicht interessiert Sie hier ein Video über das CICS Transaction Gateway. Es wurde im IBM Entwicklungslabor in Hursley, Südengland aufgenommen. Das Hursley Labor ist spektakulär in einem früheren englischen Adelssitz untergebracht:

http://www.youtube.com/watch?v=PRJMxWhYNqA

Eine Art der Anwendungsentwicklung:

http://www.youtube.com/watch?v=Wc39pSHHKIg

(1) Eine kurze Einführung in SNA und APPC finden Sie unter

http://www.cedix.de/VorlesMirror/Band2/SNA.pdf

http://www.cedix.de/VorlesMirror/Band2/APPC.pdf

Eine Beschreibungen der SNA Enterprise Extender finden Sie unter

http://www.cedix.de/VorlesMirror/Band2/SNA02.pdf

Ein guter Überblick ist enthalten in:
IBM Redbook: Introduction to the New Mainframe: Networking, August 2006, SG24-6772-00, http://www.redbooks.ibm.com/abstracts/sg246772.html

und
IBM Redbook: System z Connectivity Handbook, Februar 2013, SG24-544-13, http://www.redbooks.ibm.com/redbooks/pdfs/sg245444.pdf

Eine ausführliche Beschreibung von SNA ist enthalten in dem Lehrbuch:
R.C. Cypser: Communications for Cooperating Systems, OSI, SNA, and TCP/IP. Addison Wesley, 1991. ISBN 0-201-50775-7

Eine ausführliche HATS Demo finden Sie unter

http://jedi.informatik.uni-leipzig.de/de/VorlesMirror/ii/Vorles/galadriel.pdf