

Client/Server-Systeme

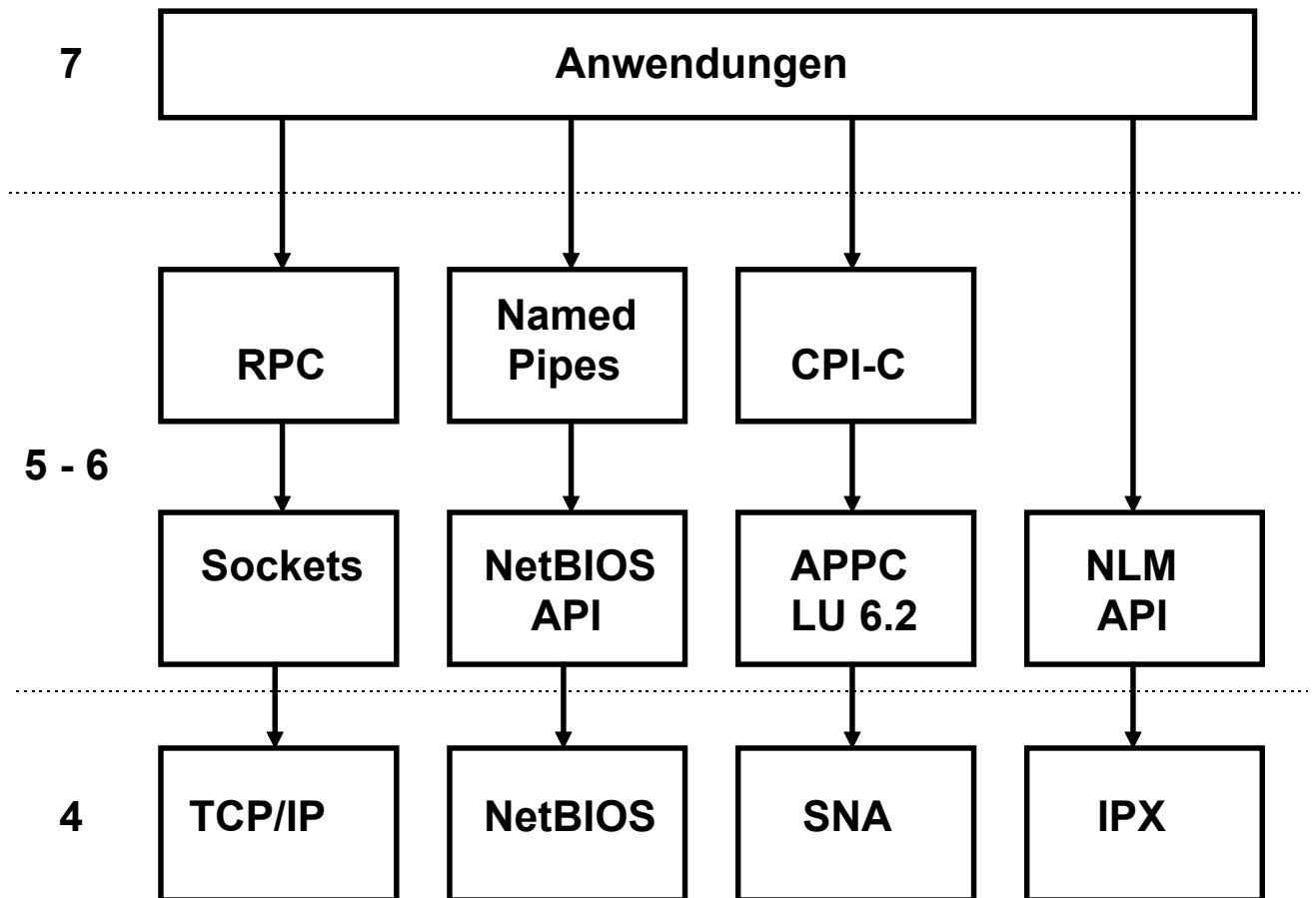
Prof. Dr.-Ing. Wilhelm G. Spruth

SS 2003

Teil 15

OS/390 Internet Integration

Schicht



Gebräuchliche Programmierschnittstellen (zahlreiche Querverbindungen)

weitere Protokolle: HTTP, 3270 (verwendet SNA)

Corba / IIOP, RMI, DCOM, verwenden Sockets und/oder RPC. Microsoft SMB verwendet Named Pipes.

CICS Distributed Program Link

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 die LU2 um mit einer OS/390 TSO oder CICS Session zu kommunizieren.

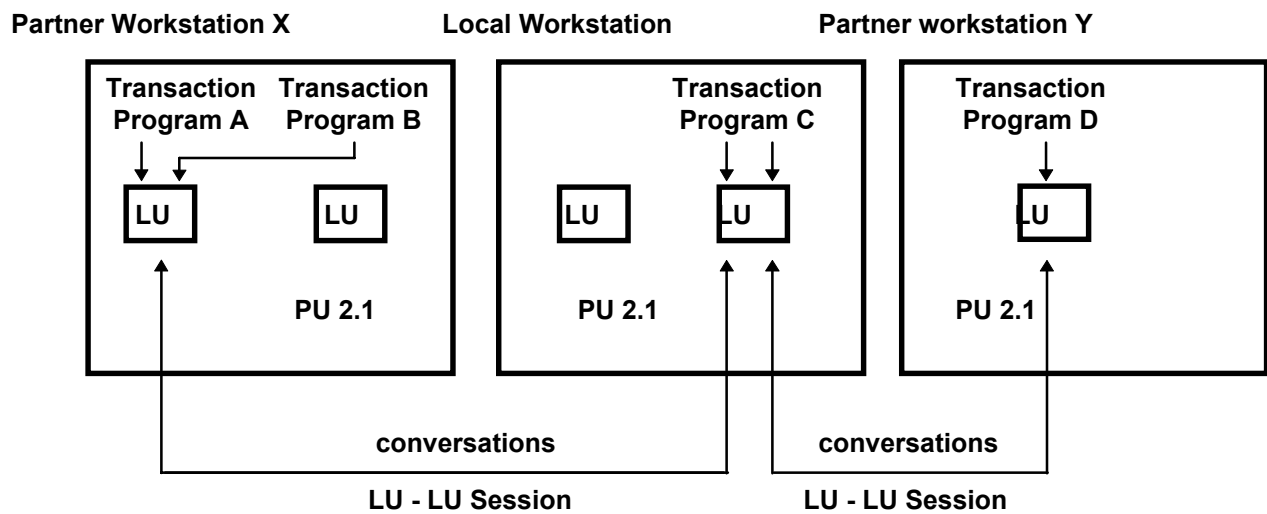
Der derzeitig gängigste Typ wird als LU 6.2 bezeichnet. Er hat volle Peer-to-Peer Funktionalität in beiden Richtungen.

Die Hardware und Software 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 /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



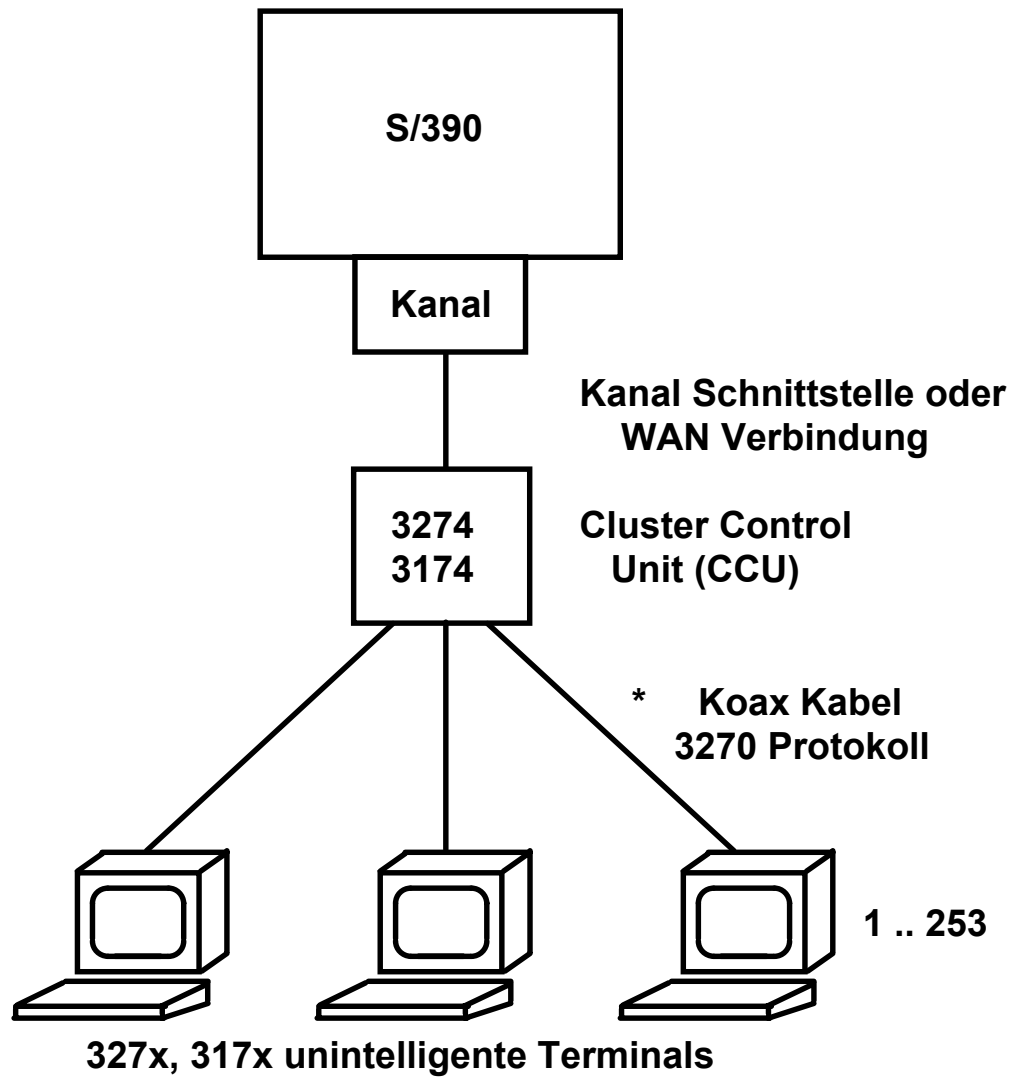
Transaktionsverarbeitung über ein SNA Netz

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

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

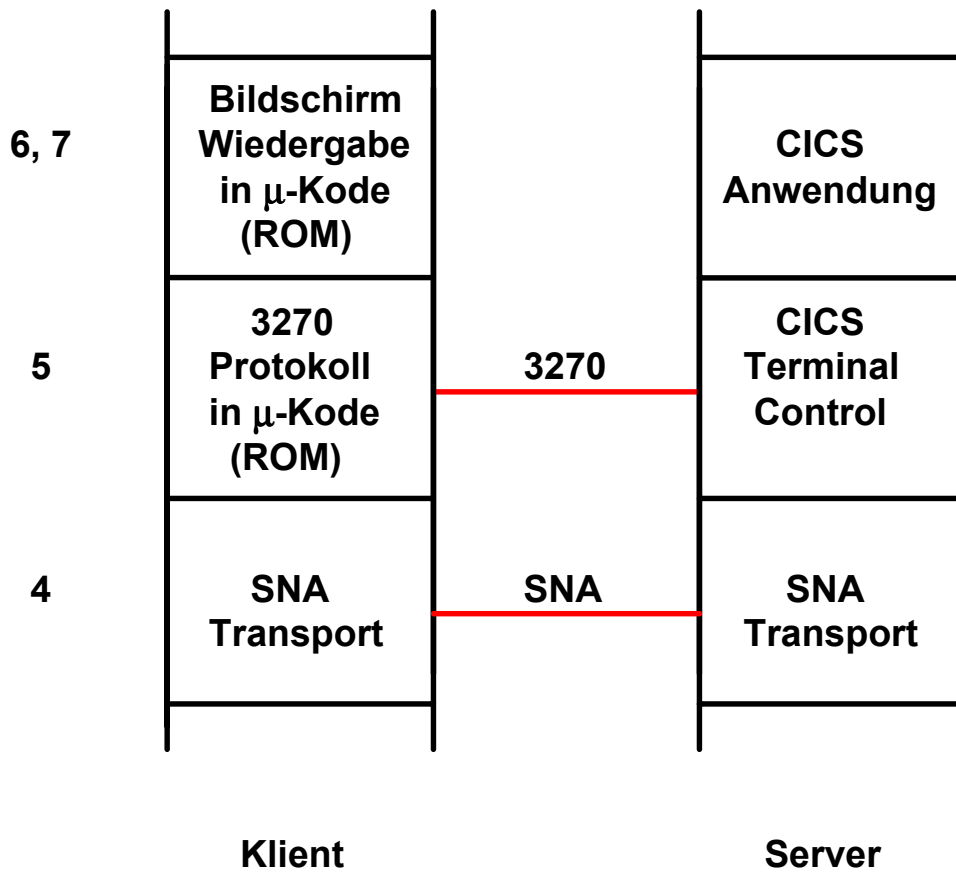
LU 6.2 stellt ähnliche TSAP's zur Verfügung. Mit Hilfe der APPC-Schnittstelle können 2 Anwendungen auf 2 Rechnern miteinander kommunizieren

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

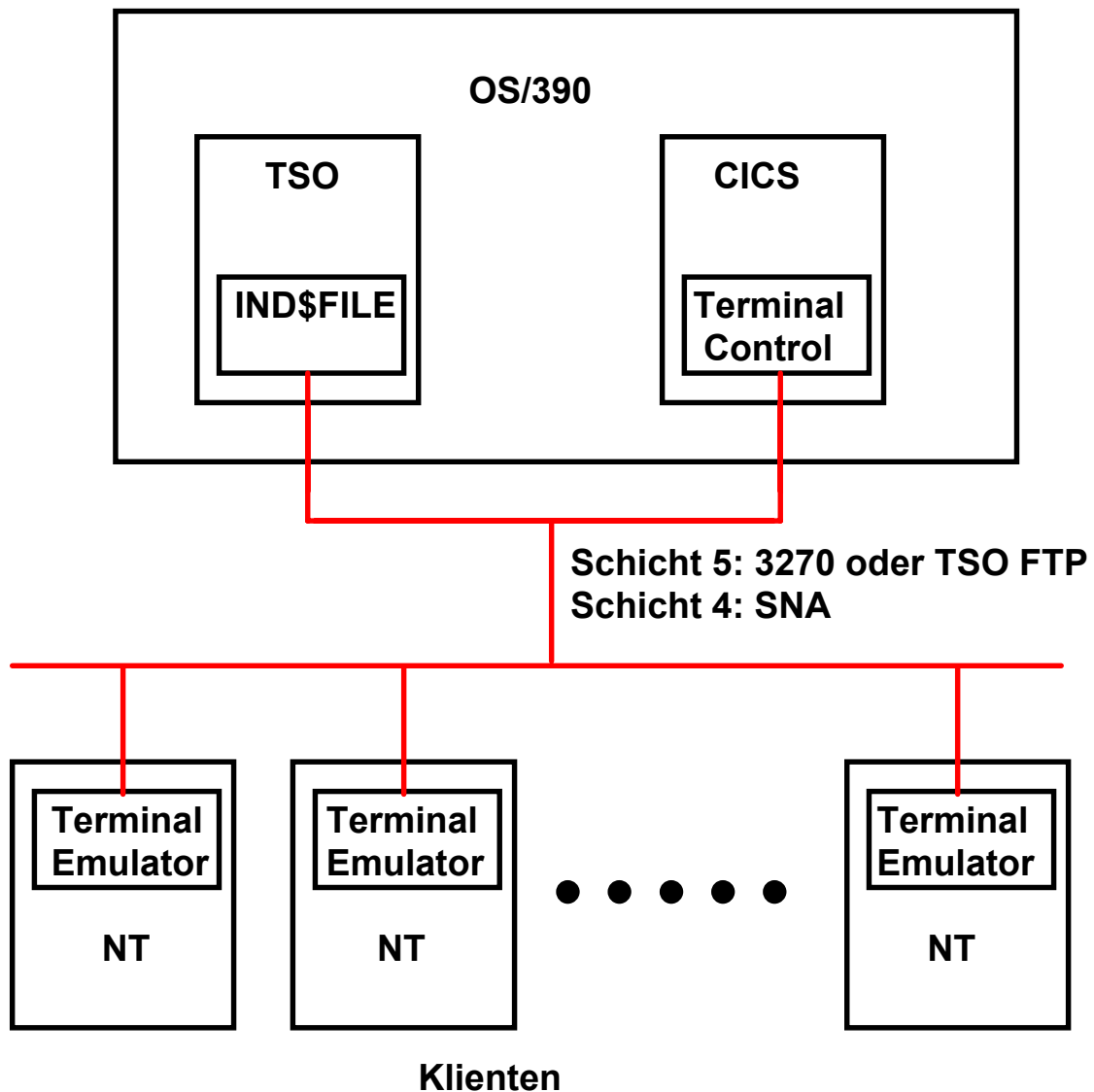


Ursprüngliche 327x Konfiguration

Schicht



CICS Client/Server Kommunikation ursprüngliche 327x Terminals



CICS Client/Server Kommunikation über ein SNA Netz

**Windows Klienten mit Terminal Emulation
und zusätzlicher File Transfer Einrichtung**

Parallelbetrieb von SNA und TCP/IP TN3270 und TCP62

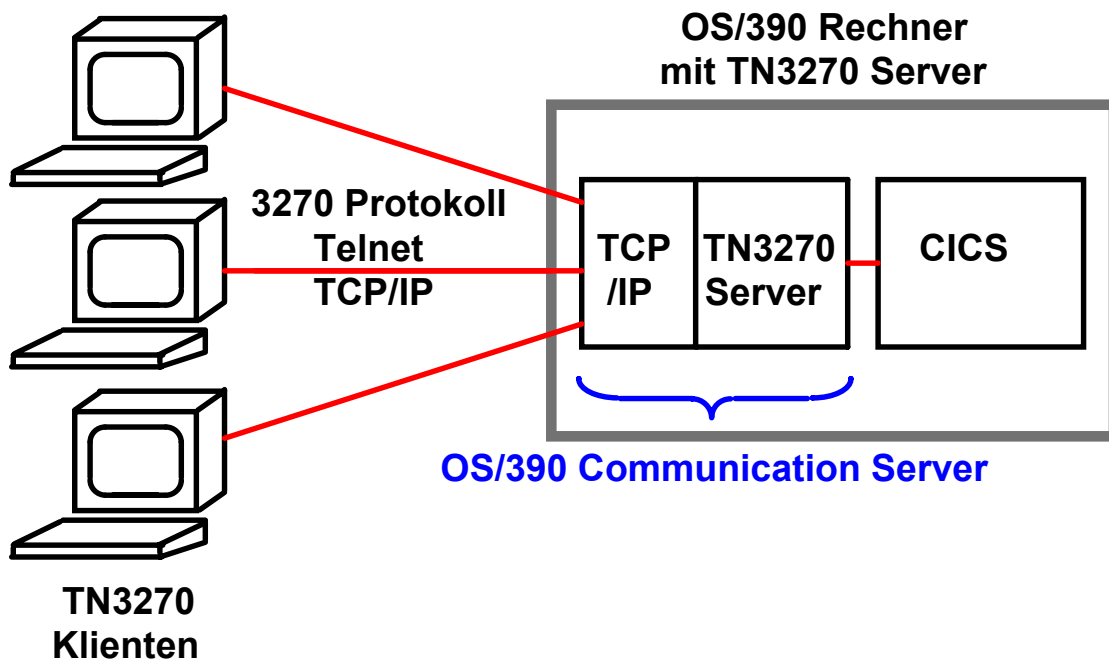
CICS Klienten und Server kommunizieren miteinander über das SNA Netzwerkprotokoll.

Viele Wirtschaftsunternehmen benutzen intern nach wie vor SNA, stellen (sehr) langsam von SNA auf TCP/IP um. Für Internet Verbindungen ist TCP/IP erforderlich.

Lösung: SNA Protokolle über TCP/IP transportieren. Zwei unterschiedliche Implementierungen.

3270 Terminals (und 3270 PC-Emulatoren) verwenden SNA LU2. Die TCP/IP Implementierung erfolgt über TN3270 (LU 2 über TCP/IP)..

Verteilte CICS Systeme kommunizieren miteinander über „Distributed Program Link“ (häufig über die EPI Schnittstelle implementiert). DPL verwendet SNA LU6.2, erhöhter Funktionsumfang gegenüber LU2. Die TCP/IP Implementierung erfolgt über TCP62 (LU 6.2 über TCP/IP).



TN3270 Protokoll

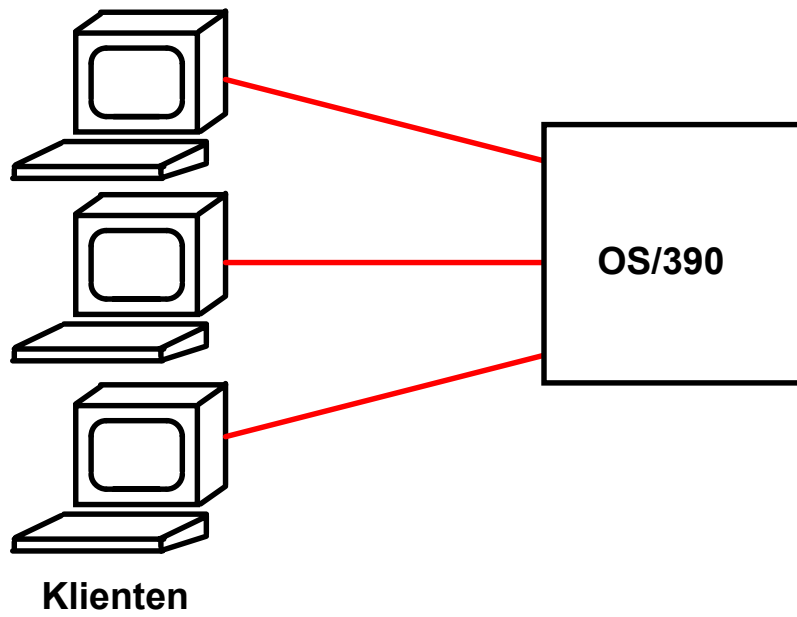
TN3270 Klienten sind über ein TCP/IP Netzwerk mit dem TN3270 Server verbunden.

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).

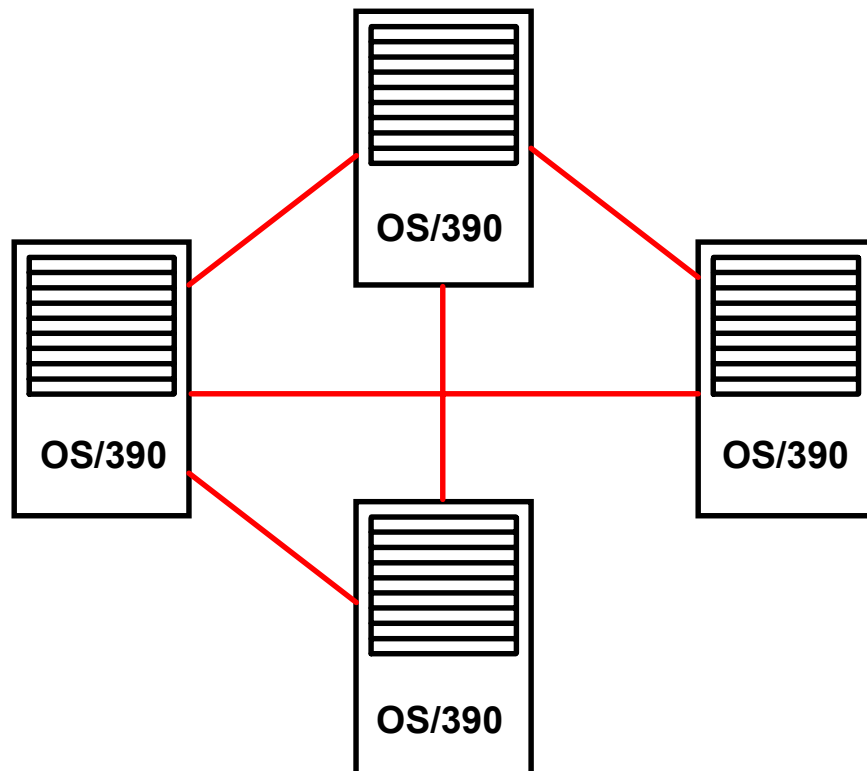
Zwischen dem TN3270 Server und der OS/390 Anwendung wird das SNA Protokoll benutzt.

Zwischen dem TN3270 Server und dem TN3270 Klienten wird das IP Protokoll benutzt.

TN3270e erweitert das ursprüngliches TN3270 Protokoll und ist in RFC 1647 spezifiziert.



Client - Server Konfiguration
Verbindung über TN3270



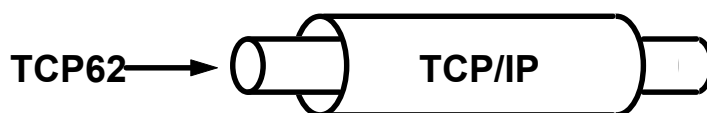
Server - Server Konfiguration
Verbindung über TCP62

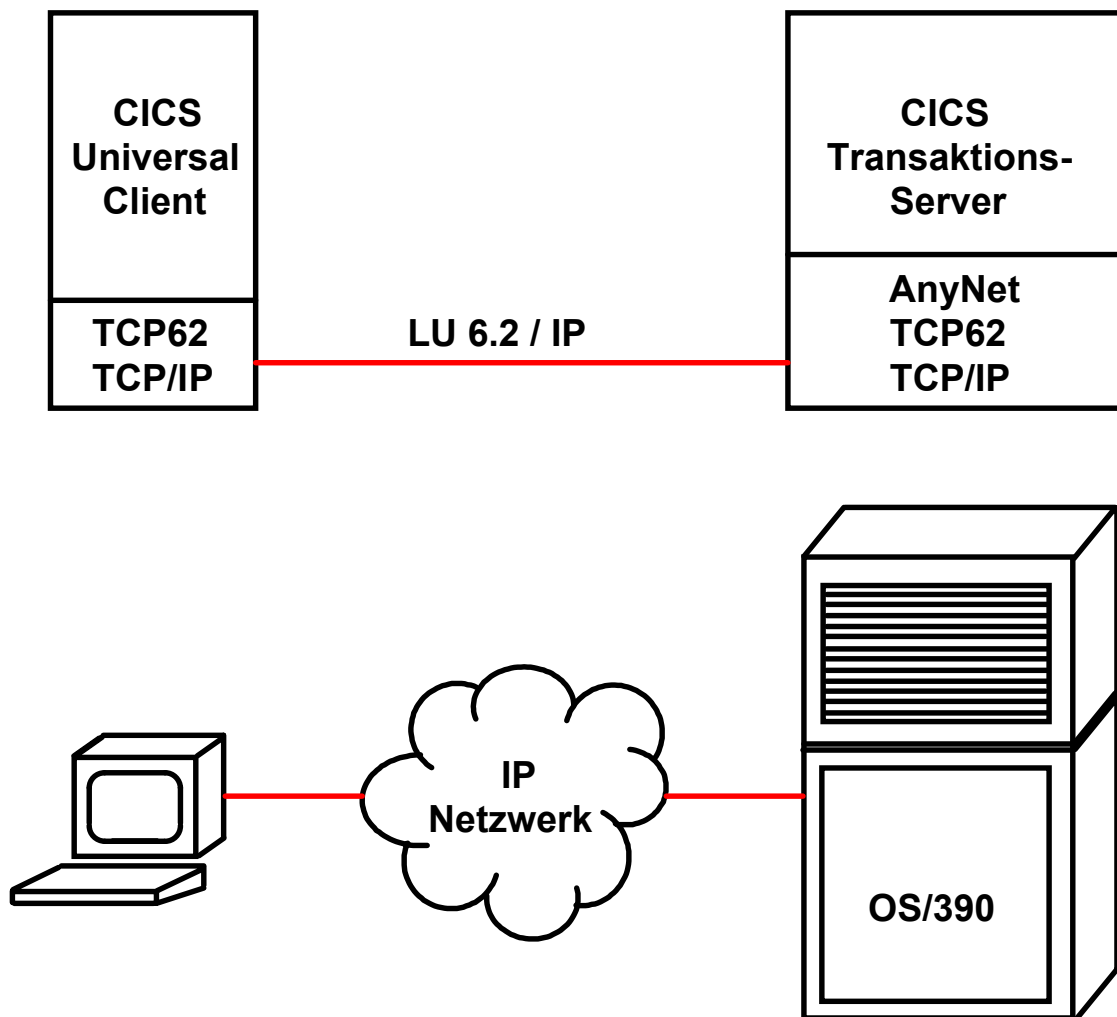
AnyNet

AnyNet ist eine OS/390 Komponente, die es SNA Anwendungen ermöglicht, über nicht-SNA Netze miteinander zu kommunizieren.

TCP62 ist eine AnyNet Komponente, die es zwei LU6.2 Einheiten ermöglicht, über TCPIP miteinander zu kommunizieren.

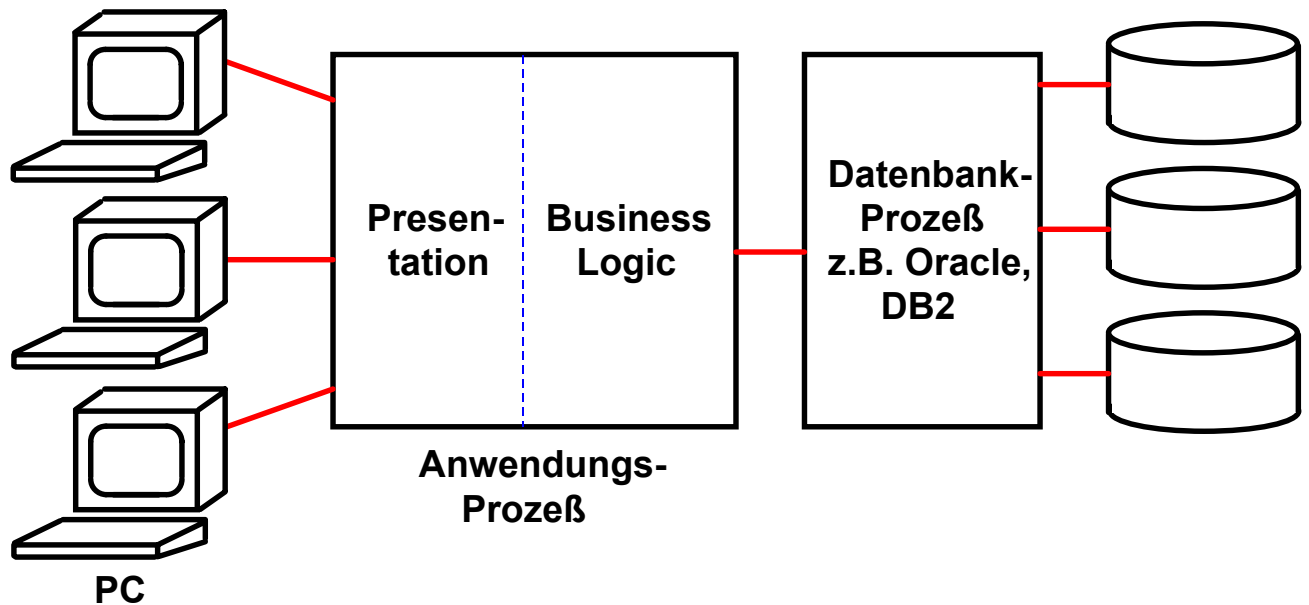
TCP/IP Pakete transportieren SNA Pakete als Nutzlast. Der Sender verpackt ein SNA Paket als TCP/IP Paket; der Empfänger entpackt es wieder.





CICS Universal Client

1. The CICS Universal Client, using your CICSCLI.INI definitions, passes data to the AnyNet component on the workstation.
2. The AnyNet component on the client workstation uses the domain name suffix with the partner LU name to generate an Internet Protocol (IP) name. The IP address for this name is then determined either from the local IP hosts file or from a domain name server (DNS). Using standard TCP/IP flows, the data is shipped from the TCP/IP component on the workstation to TCP/IP on OS/390.
3. TCP/IP on OS/390 routes the data received from the workstation to the AnyNet component on OS/390.
4. The SNA over TCP/IP feature of AnyNet translates the inbound IP routing information to SNA routing information. The data is passed to VTAM and, in turn, to the CICS Transaction Server for OS/390.



Registrierkasse
Geldausgabeautomat

Business- und Präsentationslogik

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

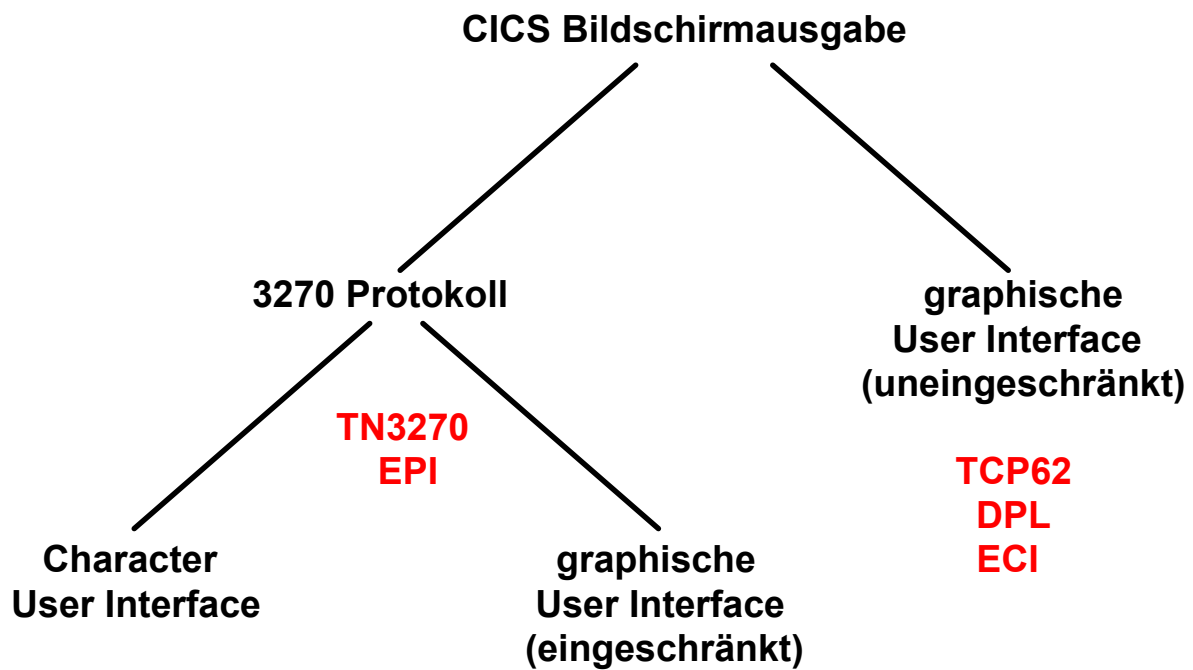
Business Logik ist der Teil, in dem Berechnungen erfolgen und Daten in einer Datenbank gelesen/geschrieben werden.

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, Java usw. geschrieben.

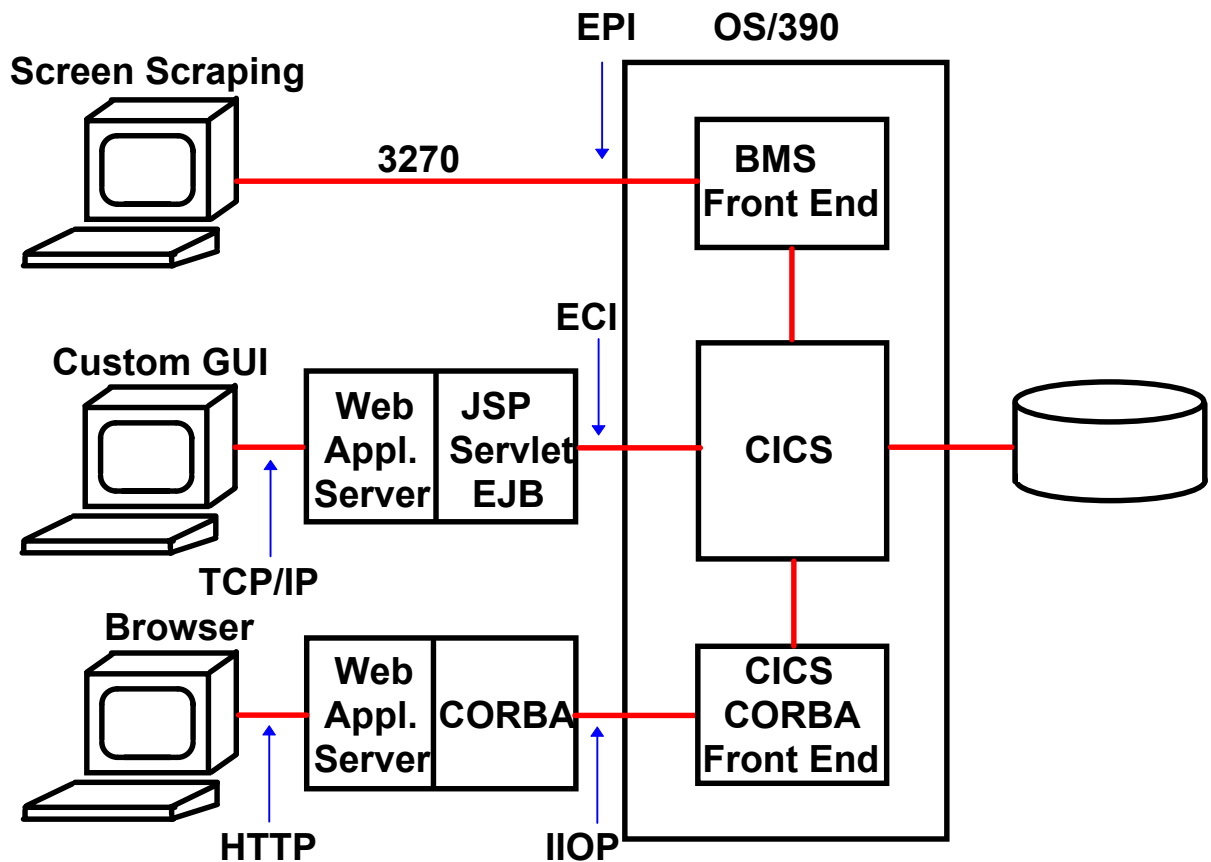
Für die Präsentations - Logik gibt es viele Möglichkeiten. Die modernste Alternative benutzt Java Server Pages und einen Web Application Server um den Bildschirminhalt innerhalb eines Web Browsers darzustellen.

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



Alternativen der Bildschirmausgabe

CUI	Character User Interface
GUI	Graphical User Interface

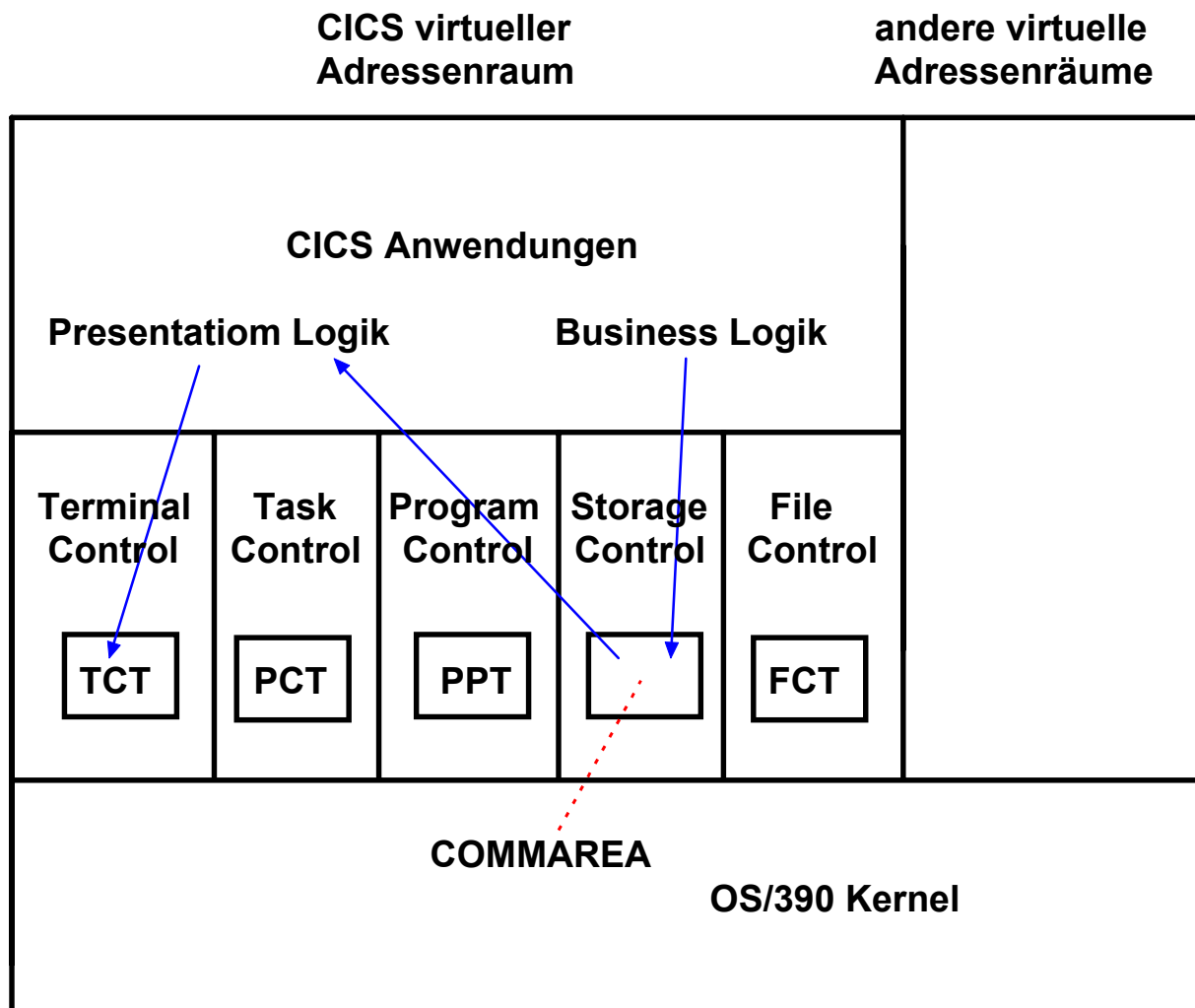


CICS Klienten Anbindung

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

ECI Die Presentation Service Komponente von CICS (BMS) wird nicht genutzt. Direkter Zugriff auf COMMAREA

IIOIP Ähnlich ECI, vollwertiges CORBA Protokoll

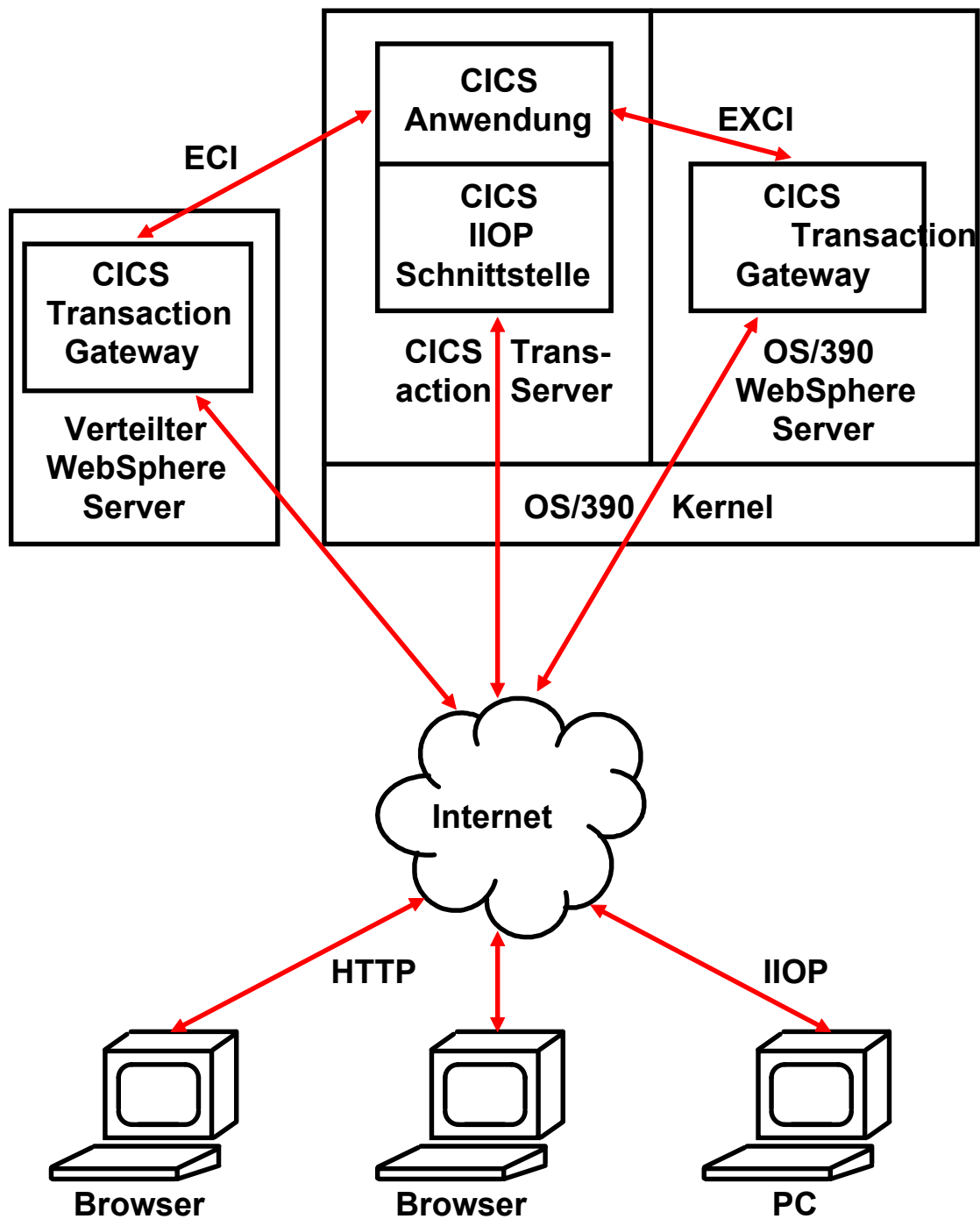


COMMAREA

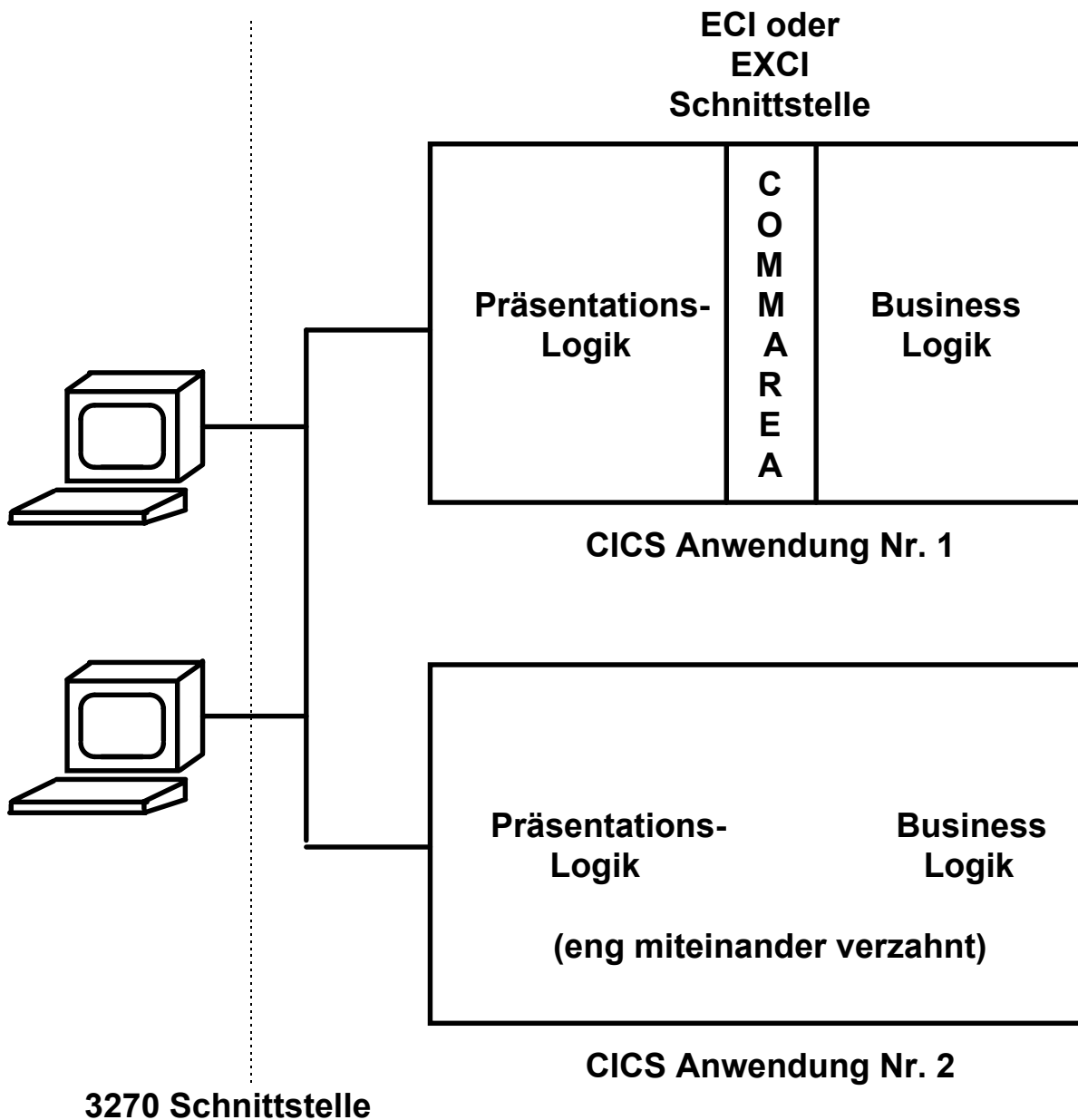
Communication Area

COMMAREA kann benutzt werden um Ein/Ausgabedaten zu übergeben:

- An ein CICS Anwendungsprogramm im gleichen Adressenraum
- An ein CICS Anwendungsprogramm in einem anderen Adressenraum auf dem gleichen physikalischen Rechner,
- An ein CICS Anwendungsprogramm auf einem getrennten physikalischen Rechner unter Benutzung von EXEC CICS LINK (...)
- An ein nicht unter CICS laufendes Programm, z.B. ein GUI Prozess unter Verwendung der EPI Schnittstelle

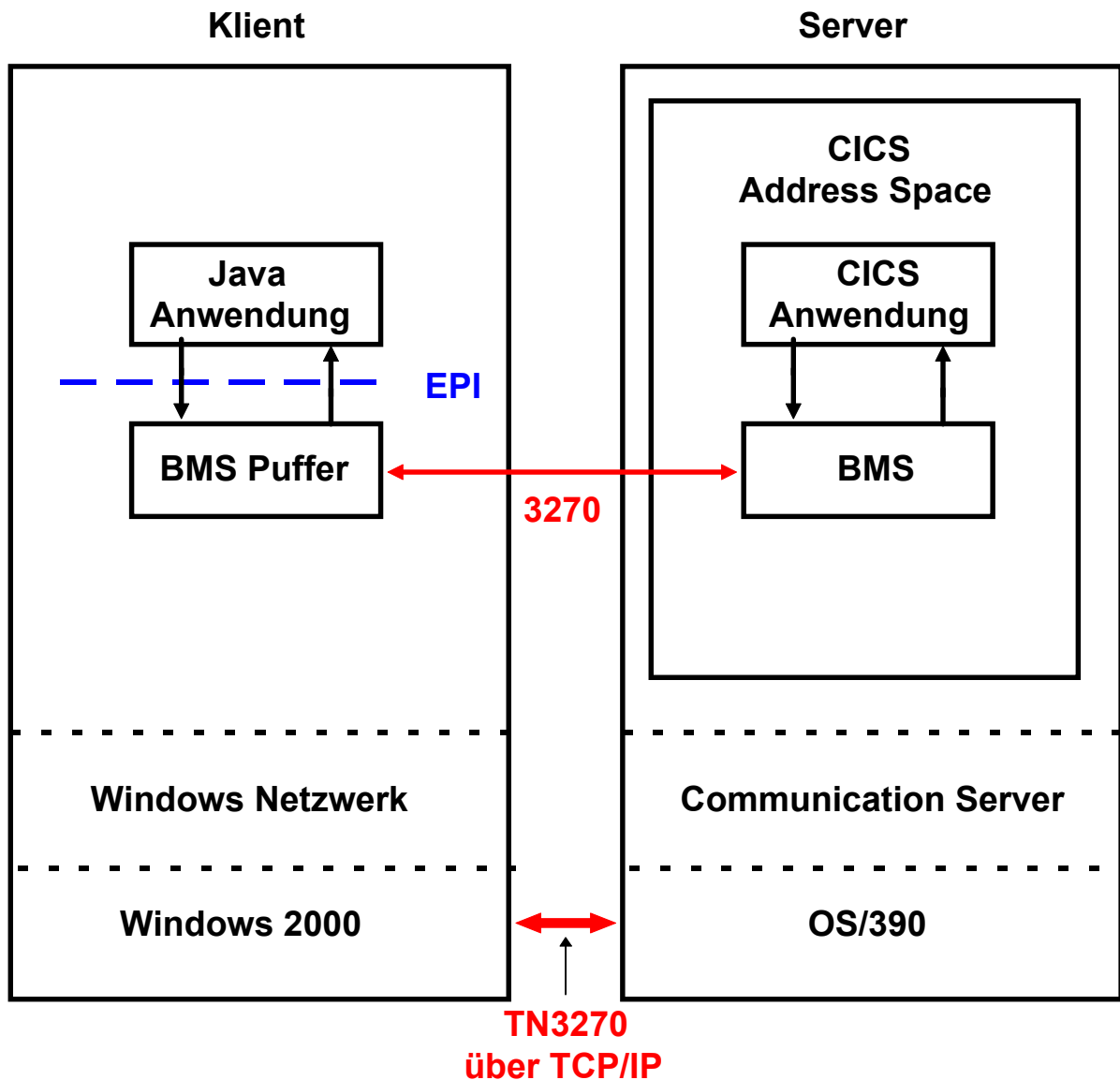


Alternative CICS Zugriffe über das Internet



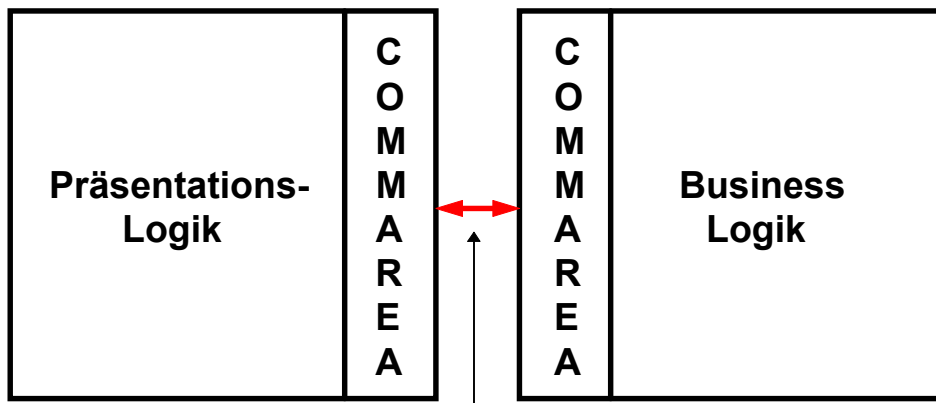
Bei vielen älteren CICS Anwendungen (und bei manchen modernen Anwendungen) sind Presentation Logik und Business Logik eng miteinander verwoben. Es ist vielfach nicht möglich oder sinnvoll, diese Anwendungen umzuschreiben, um eine saubere Trennung zwischen Business Logik und Presentation Logik zu erreichen.

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



Ein Nicht-CICS Programm kommuniziert mit einem CICS Programm über die EPI oder die ECI Schnittstelle.

Die EPI Schnittstelle benutzt den normalen 3270 Datenstrom. Ein Beispiel ist ein 3270 Klient (Terminal Emulator). Eine Java Anwendung kann über die EPI Schnittstelle eine gefälligere GUI erzeugen.



ECI oder EXCI Schnittstelle

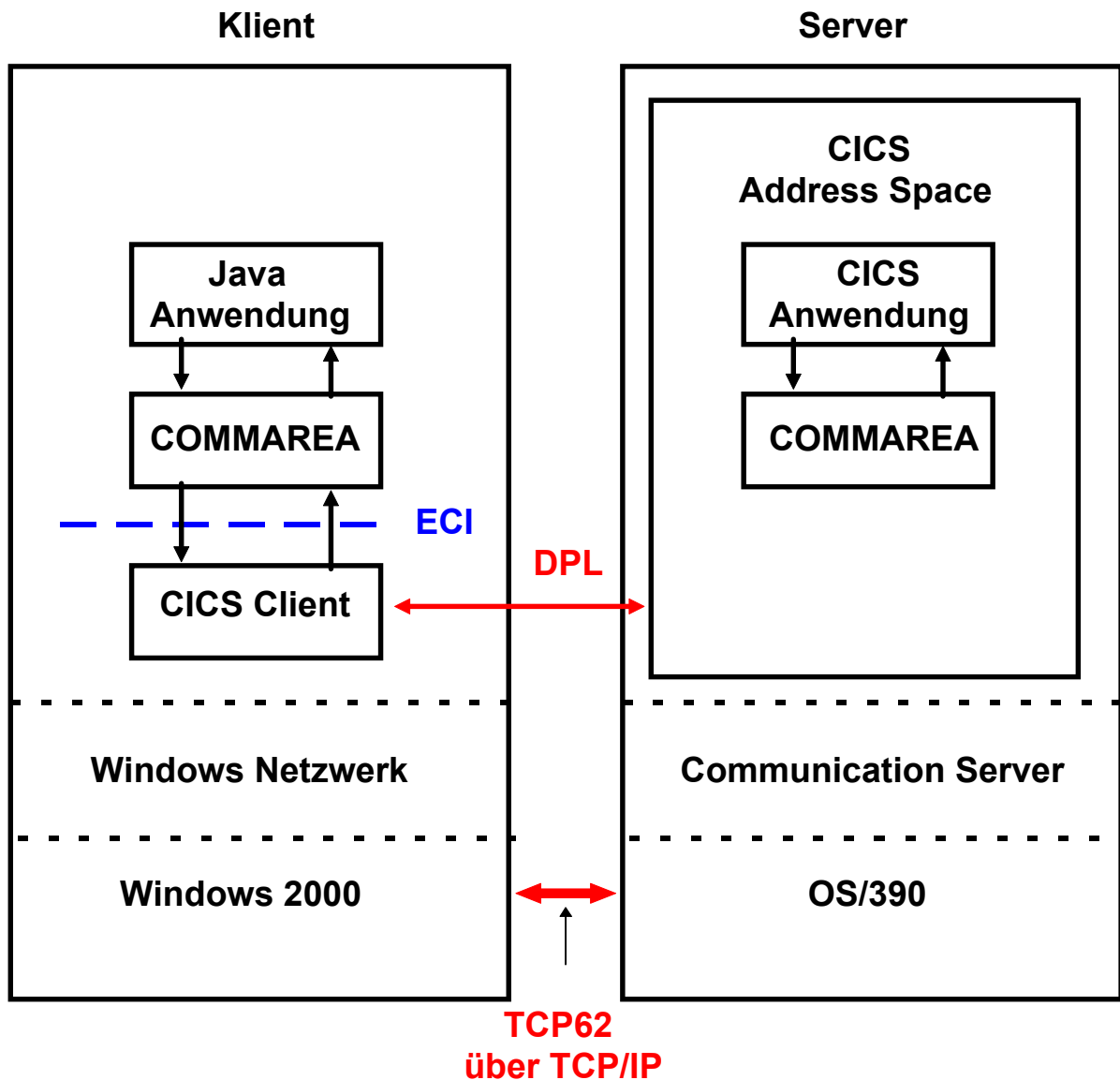
Eine CICS Anwendung besteht aus der Business Logik und der Presentation Logik.

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, der von der Storage Control Komponente des CICS Subsystems bereitgestellt wird. COMMAREA wird u.a. für die Verwaltung von Sessions verwendet, wobei der State einer Transaktion für die Nachfolgetransaktion verfügbar ist.

Hierfür existiert eine Schnittstelle, die „External Call Interface“ (ECI). Es wird die CICS „Distributed Program Link“ (DPL) Interprocess Communication Einrichtung verwendet. 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 OS/390 Rechner (oder Sysplex), kann eine als EXCI bezeichnete Version der ECI Schnittstelle verwendet werden. Diese verwendet einen Speicherbereich des OS/390 Kernels und vermeidet den Kommunikations-Overhead.

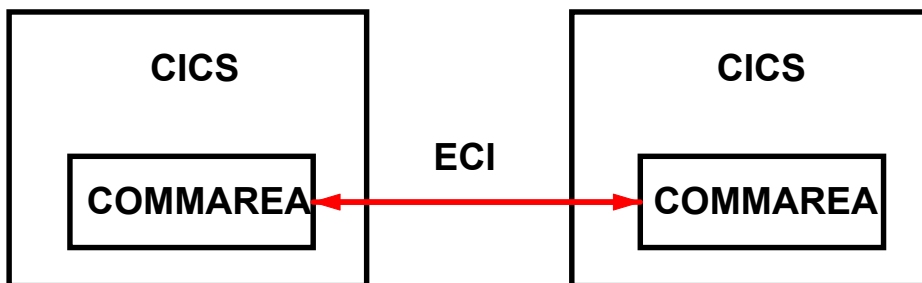


Ein CICS Client ist eine echte CICS Anwendung, die mit anderen CICS Anwendungen über Distributed Program Link (DPL, einem RPC ähnlichen Mechanismus) verkehrt.

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.

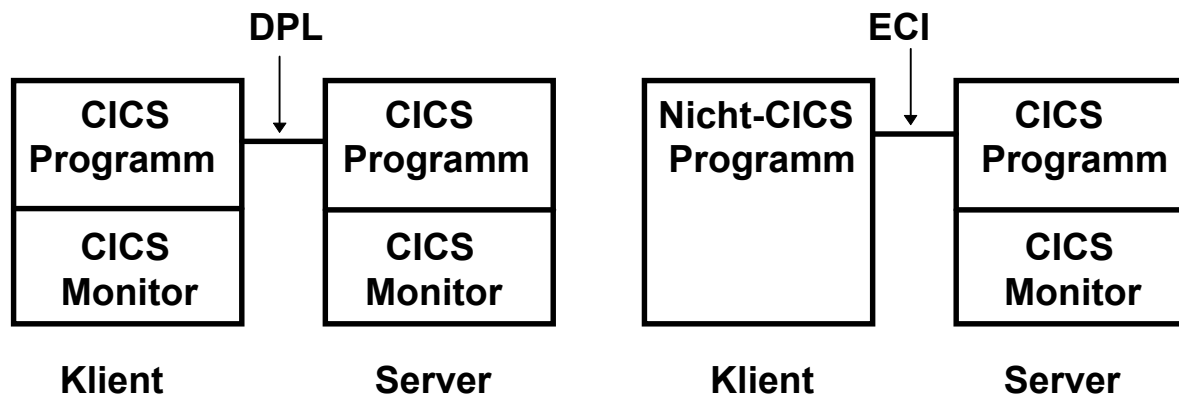
External Call Interface

Die ECI (External Call Interface) ist eine Programmierschnittstelle, mit der eine Nicht-CICS Anwendung eine CICS Anwendung auf einem getrennten Rechner aufrufen kann.



```
ECI_PARMS      EciBlock;  
cics_ushort_t Response;  
.....  
.....  
Response = CICS_ExternalCall (&EciBlock);
```

ECI Call in C/C++



CICS Remote Procedure Call Mechanismen

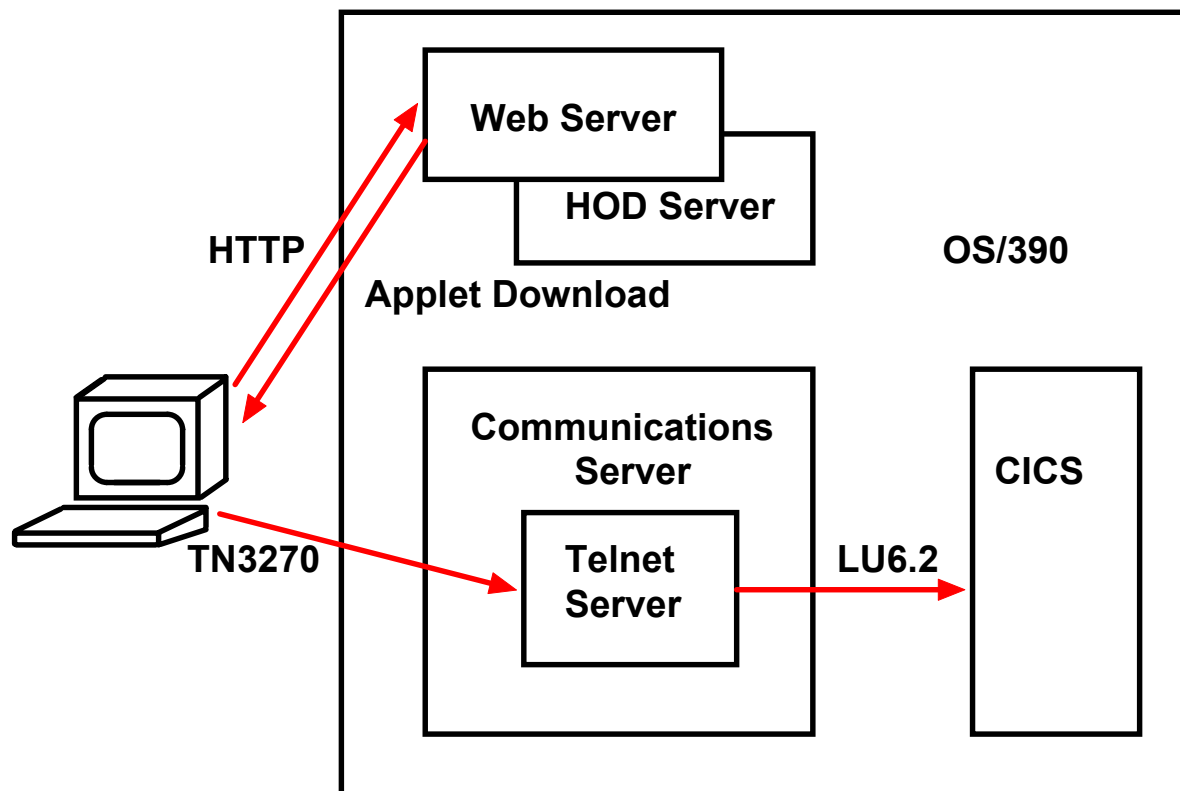
Connectors

Connectoren sind Java Beans, welche eine Schnittstelle zu existierenden Legacy Systemen bilden. Beispielsweise sind folgende Connectoren für die IBM WebSphere verfügbar:

- o JDBC
- o DB2
- o Oracle
- o Adabas
- o CICS
- o IMS
- o MQSeries
- o SAP R/3
- o Lotus Domino

Bei den Implementierungen von Web Commerce Lösungen spricht man vom Frontend, welches typischerweise mit einem Web Application Server realisiert wird, und einem Backend (Beispiele Auftragseingang, Finanzbuchhaltung), wofür vorhandene Legacy Systeme eingesetzt werden.

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



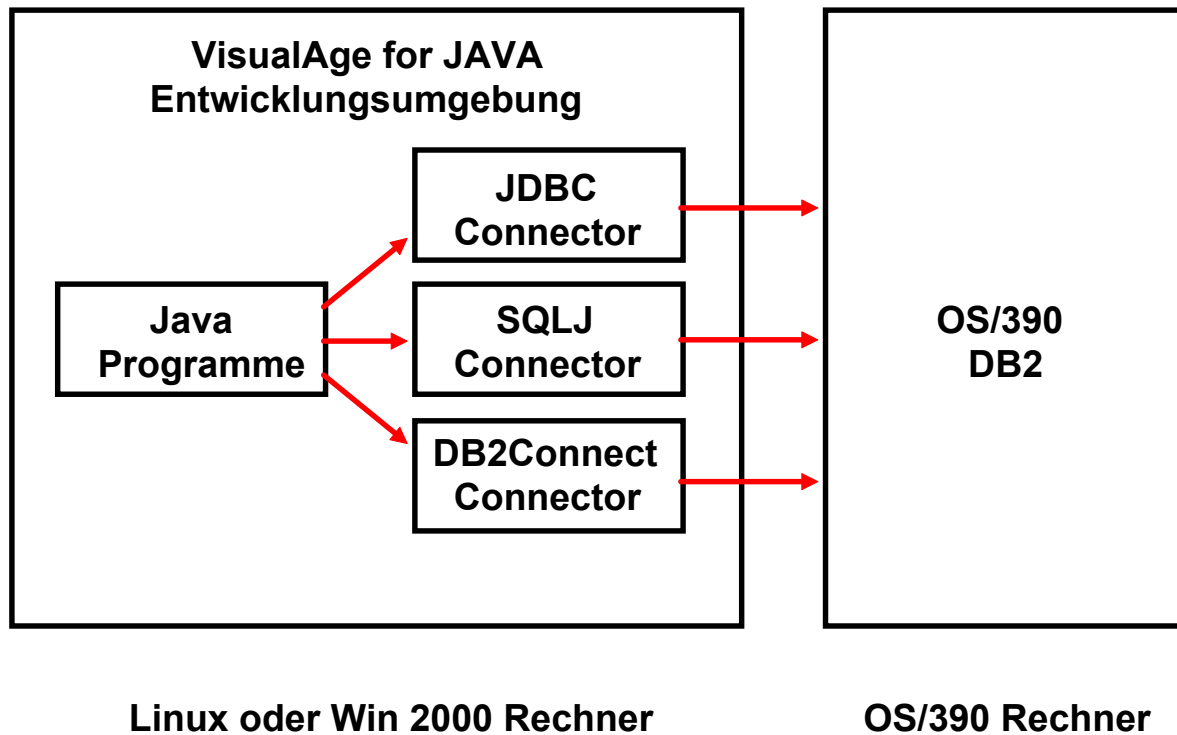
Host On-Demand

Host On-Demand (HOD) ist ein ganz in Java geschriebener 3270 Emulator.

Es ermöglicht einem Web Browser ein Applet herunterzuladen, welches die 3270 Emulator Funktion enthält.

Der 3270 Emulator enthält die Telnet 3270 Komponente, die auf die Telnet Server Komponente des OS/390 Communications Server zugreift.

Diese wiederum kommuniziert mit CICS über LU 2 oder LU 6.2 .

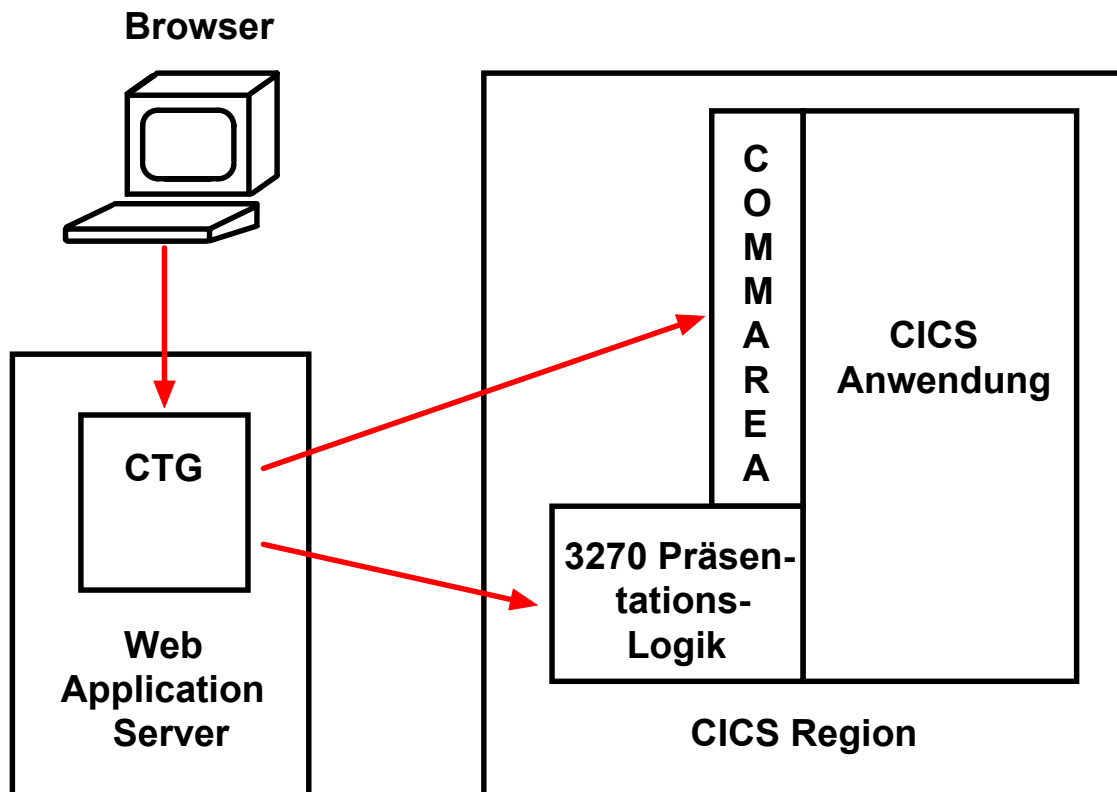


Unterschiedliche Connector Arten

JDBC ist von ODBC abgeleitet, implementiert dynamische Datenbankzugriffe

SQLJ implementiert statische Datenbankzugriffe

DB2Connect ist ein für die DB2 API optimierter Connector



CICS Transaction Gateway

Das CICS Transaction Gateway (CTG) ist eine von mehreren verfügbaren CICS Client Implementierungen. Es ermöglicht die Reduzierung der Klienten Funktionalität auf einen regulären Web Browser.

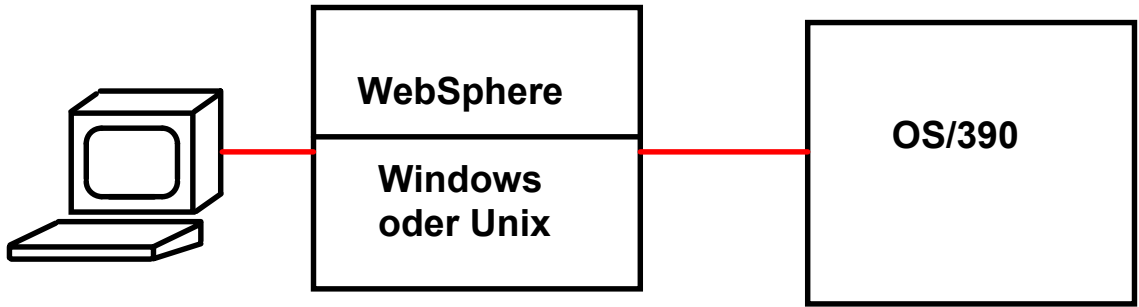
Als 2-Tier oder 3-Tier Konfiguration verfügbar

Vorteile der 2-Tier Konfiguration:

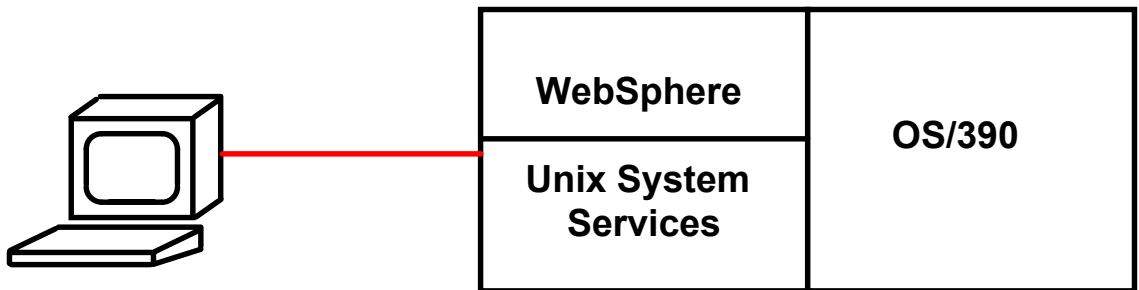
- Kein SNA (TCP/62 oder TN3270))
- Keine Sicherheitsprobleme zwischen Tier 2 und Tier 3
- Administration/System Management

Nachteile der 2-Tier Konfiguration:

- 3-Tier ist quick and dirty
- Performance/Kosten
- Produktionsumgebung anders als Entwicklungsumgebung

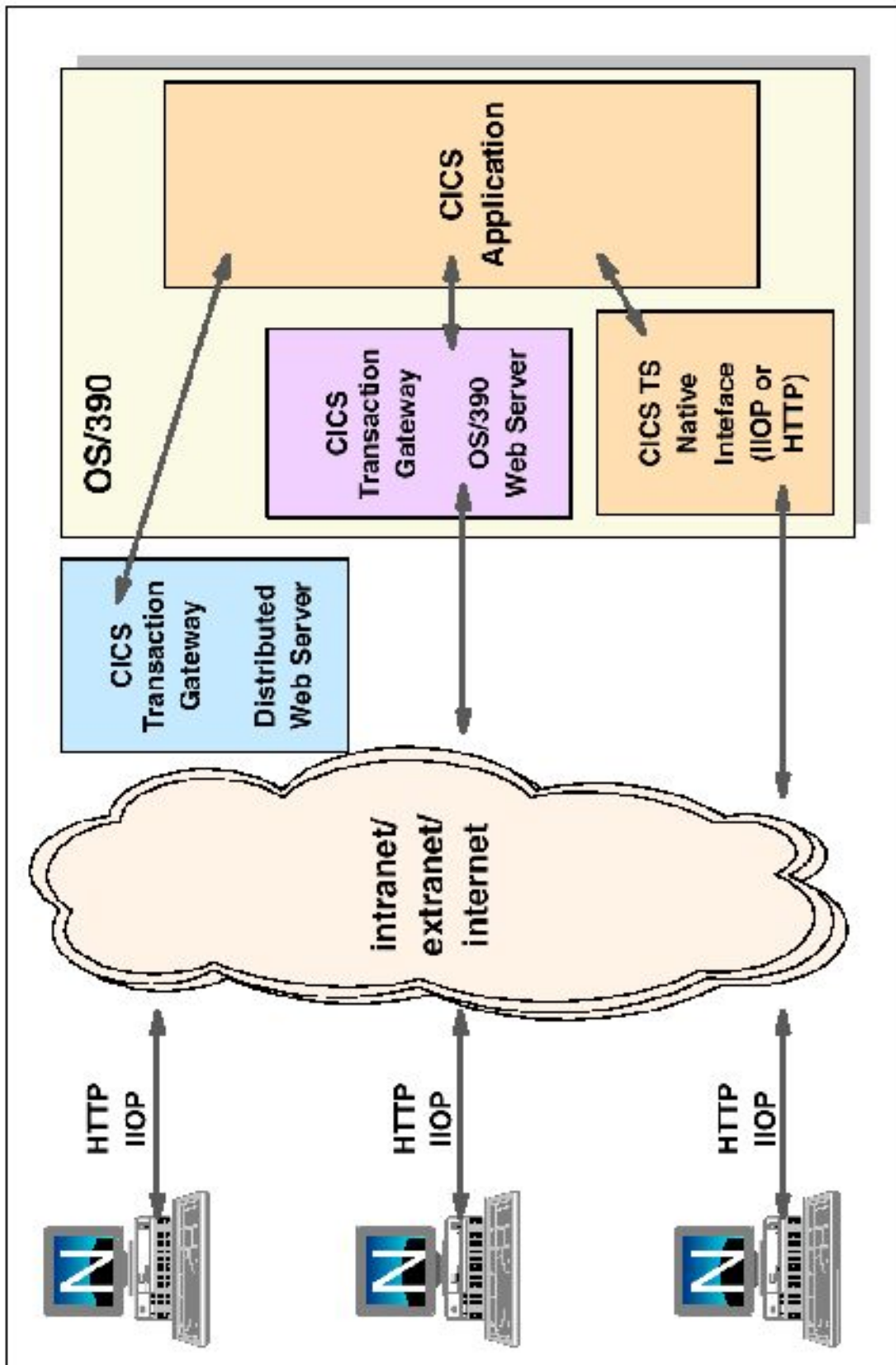


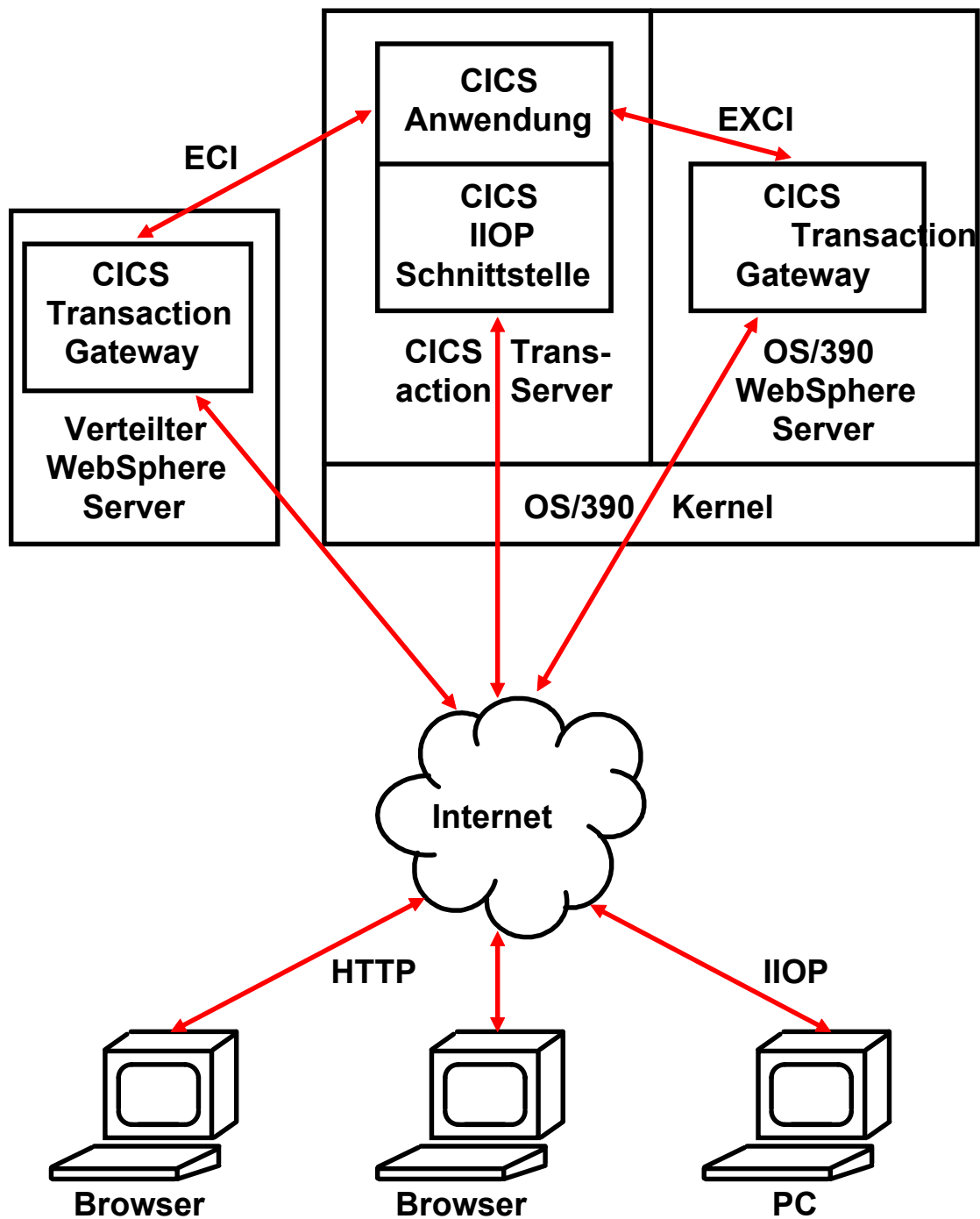
3-Tier Konfiguration



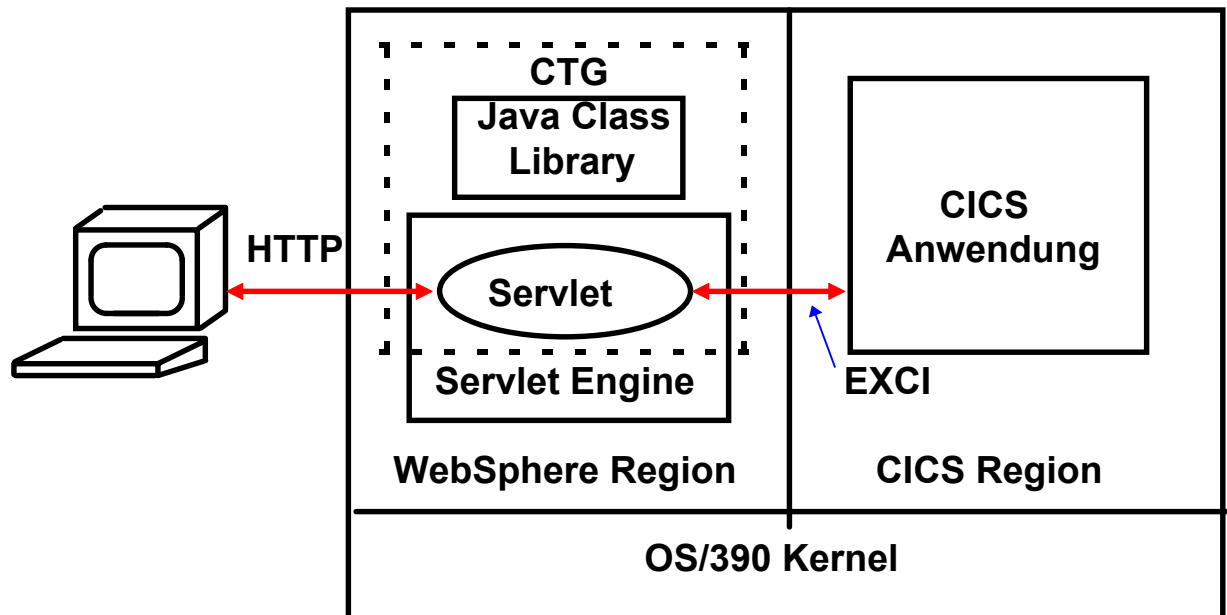
2-Tier Konfiguration

Alternative WebSphere - OS/390 Konfigurationen



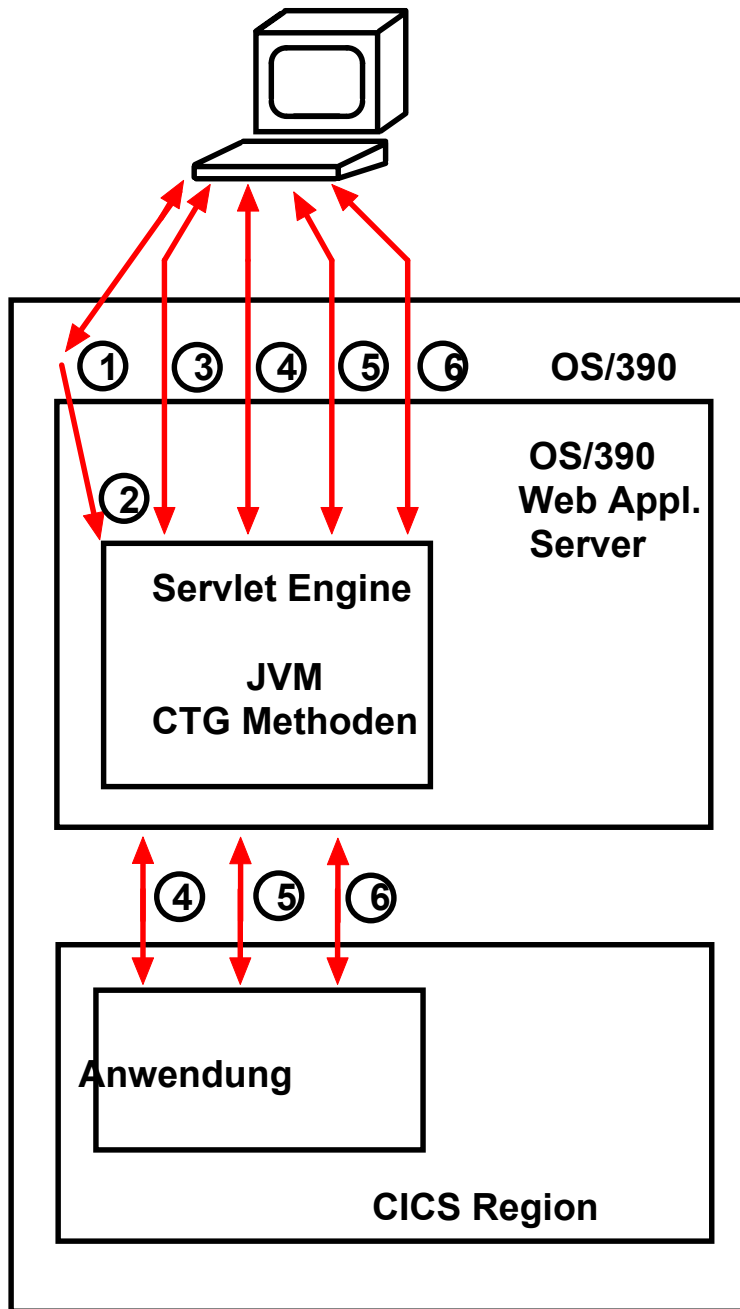


Alternative CICS Zugriffe über das Internet



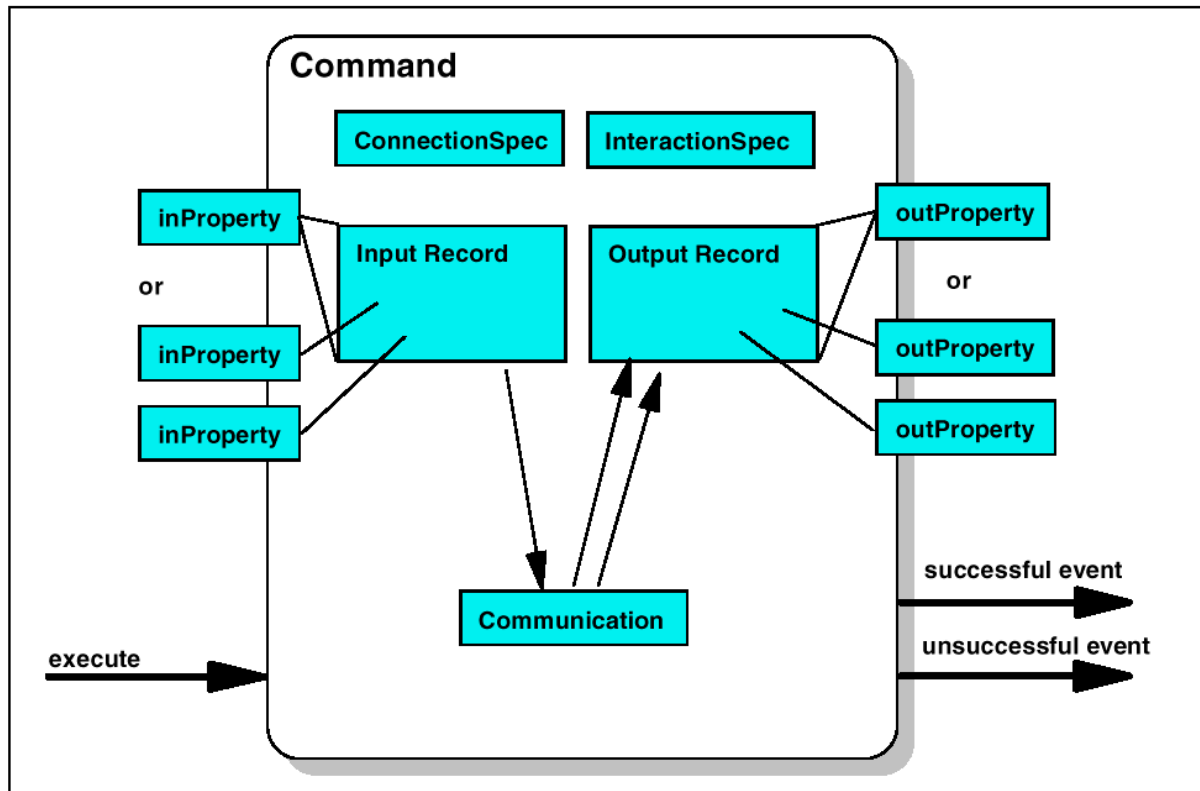
CICS Transaction Gateway Servlet Architecture

Diese Implementierung ist unter
<http://jedi.informatik.uni-leipzig.de> verfügbar



CICS Transaction Gateway

Common Connector Framework



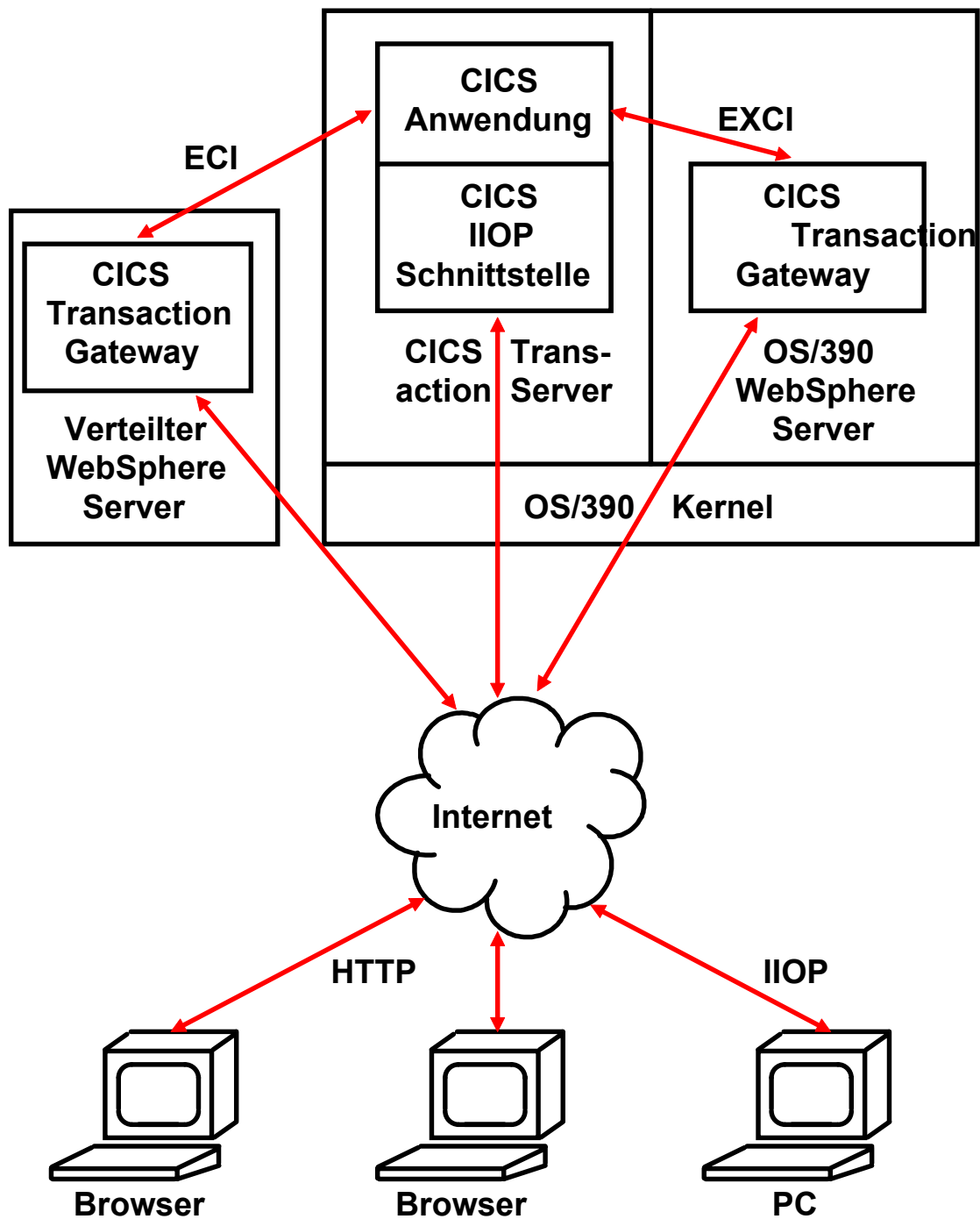
Aufruf des Common Connector Framework mit Hilfe des CCF Commands . Drei Schnittstellen: ConnectionSpec, InteractionSpec, und Communication, sowie Input und Output Records für die Datenübertragung.

ConnectionSpec spezifiziert eine Verbindung Attribute, wie Host Name, Port Nr., und Timeout Werte. Weitere Attribute können Connector spezifisch sein; beispielsweise ein Channel Identifier für einen MQSeries Connector.

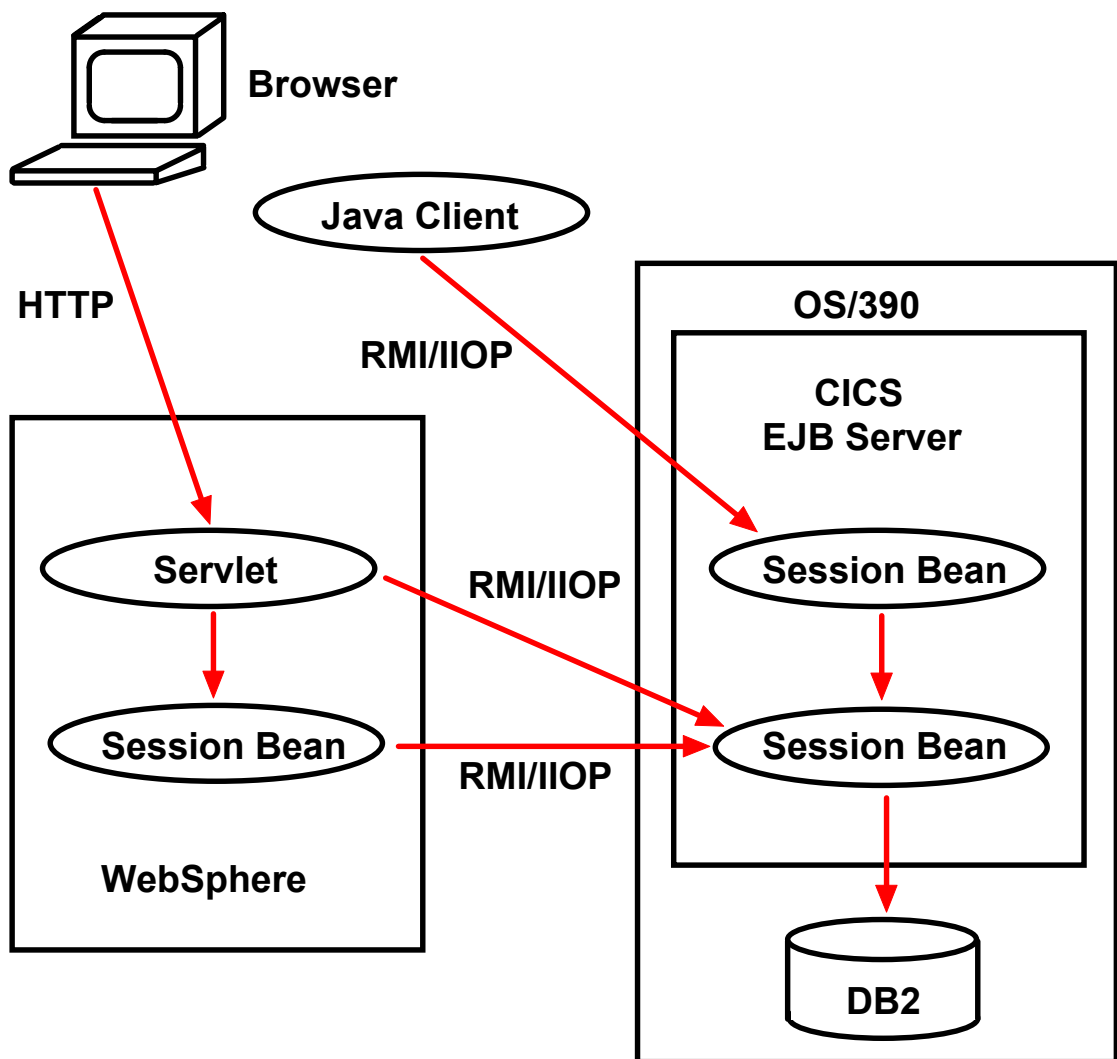
InteractionSpec enthält Angaben darüber was geschehen soll. Beispiele sind der Programmname oder Interaktionsmodus.

Eine Implementierung der Communication ruft deren Execute Methode auf. Die Execute Methode verwendet drei Argumente: InteractionSpec, Input Record und Output Record.

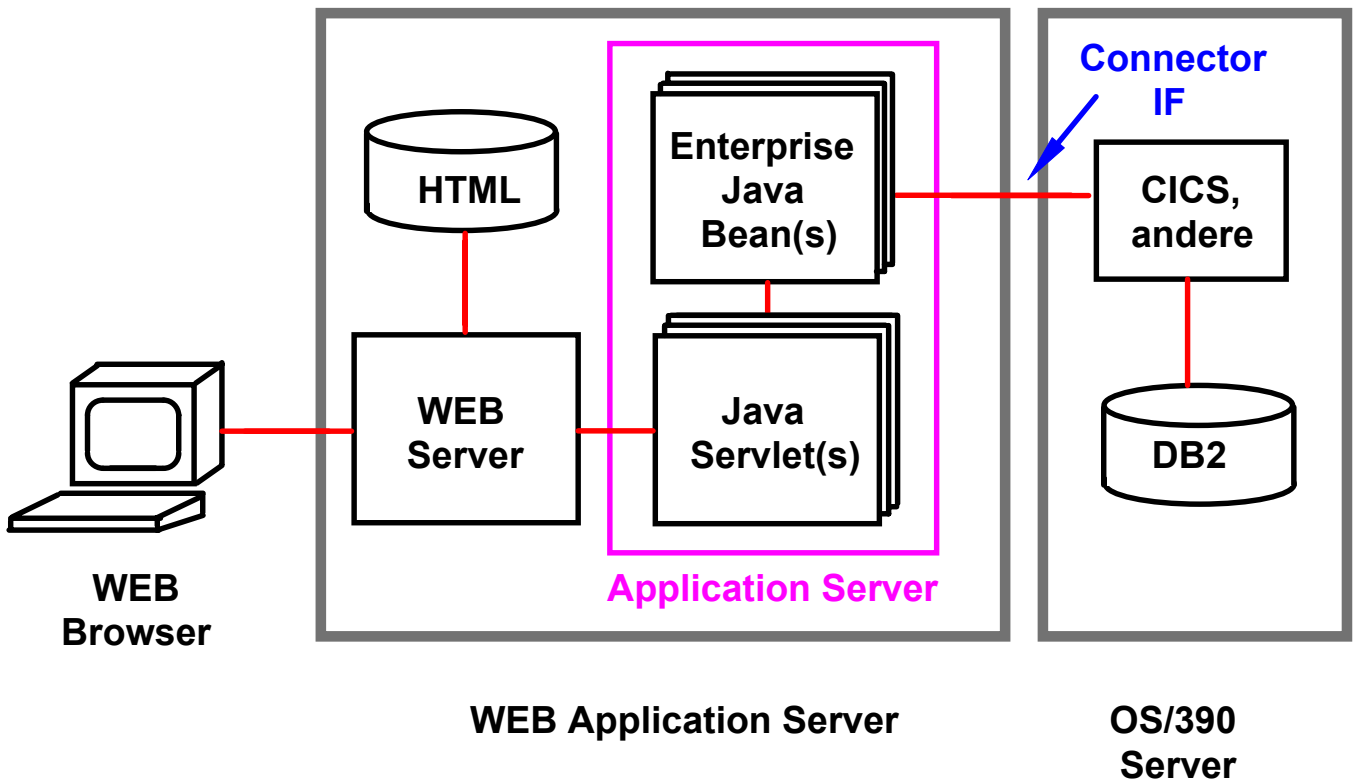
Input/Output Records repräsentieren die übertragenen Daten.



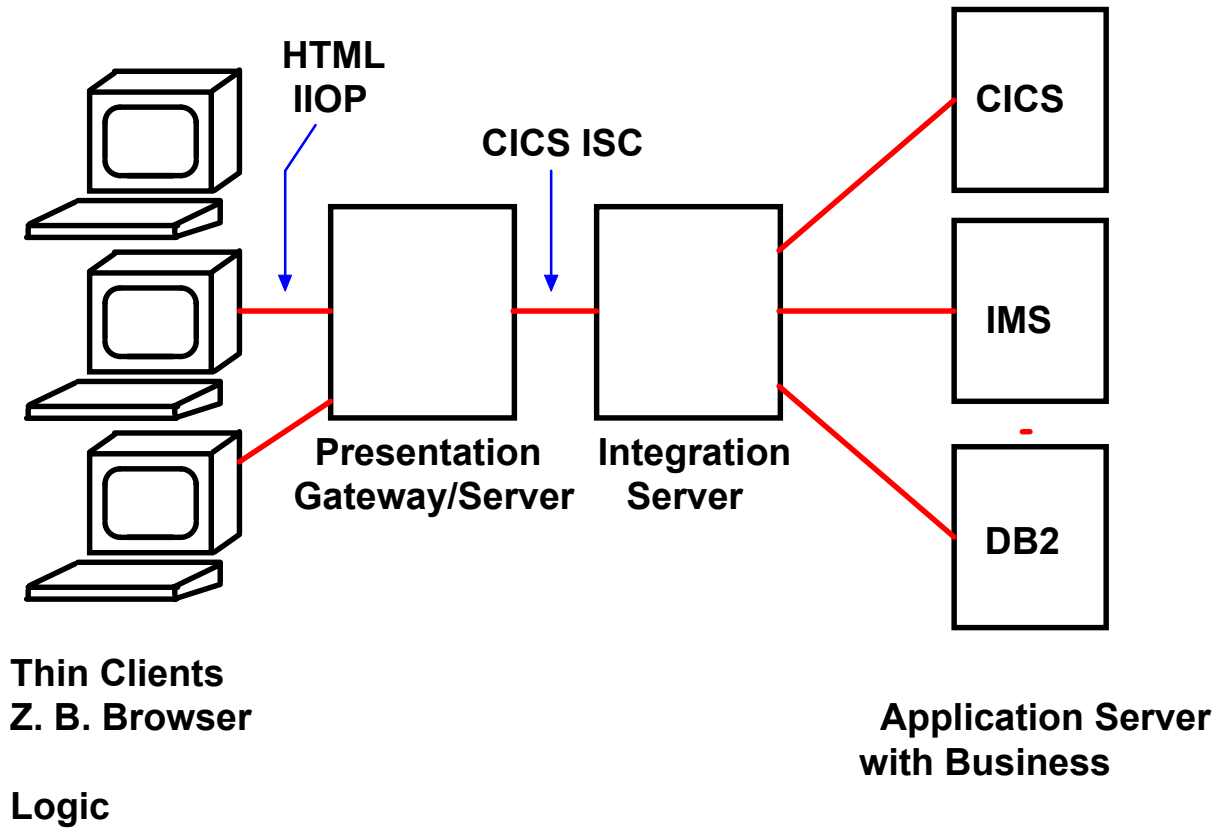
Alternative CICS Zugriffe über das Internet



Benutzung von Enterprise Java Beans in einer CICS Anwendung



Web Application Server als Presentation und Integration Server



Network Evolution

WebSphere Funktionen unter OS/390

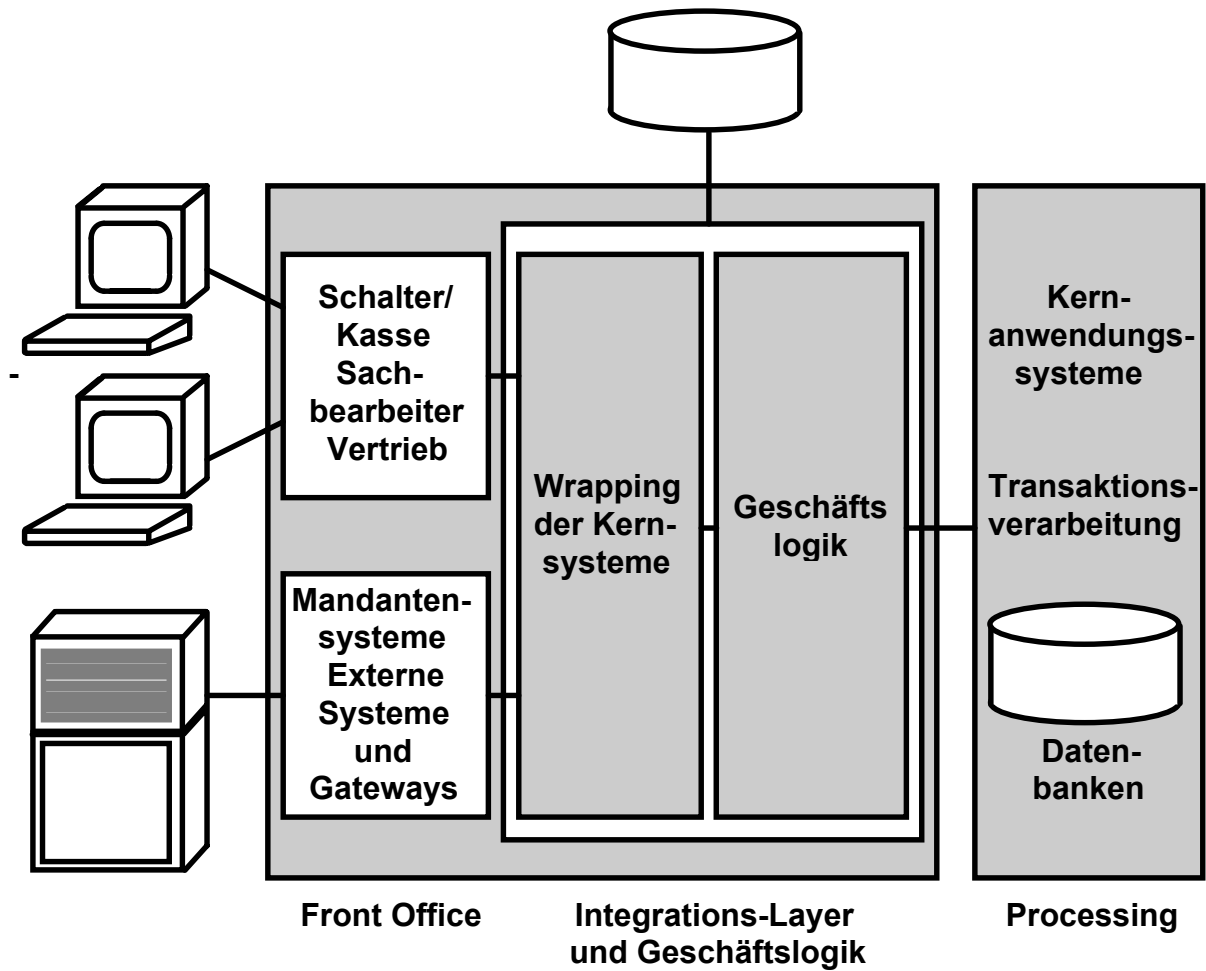
Parallel Sysplex

Workload Manager

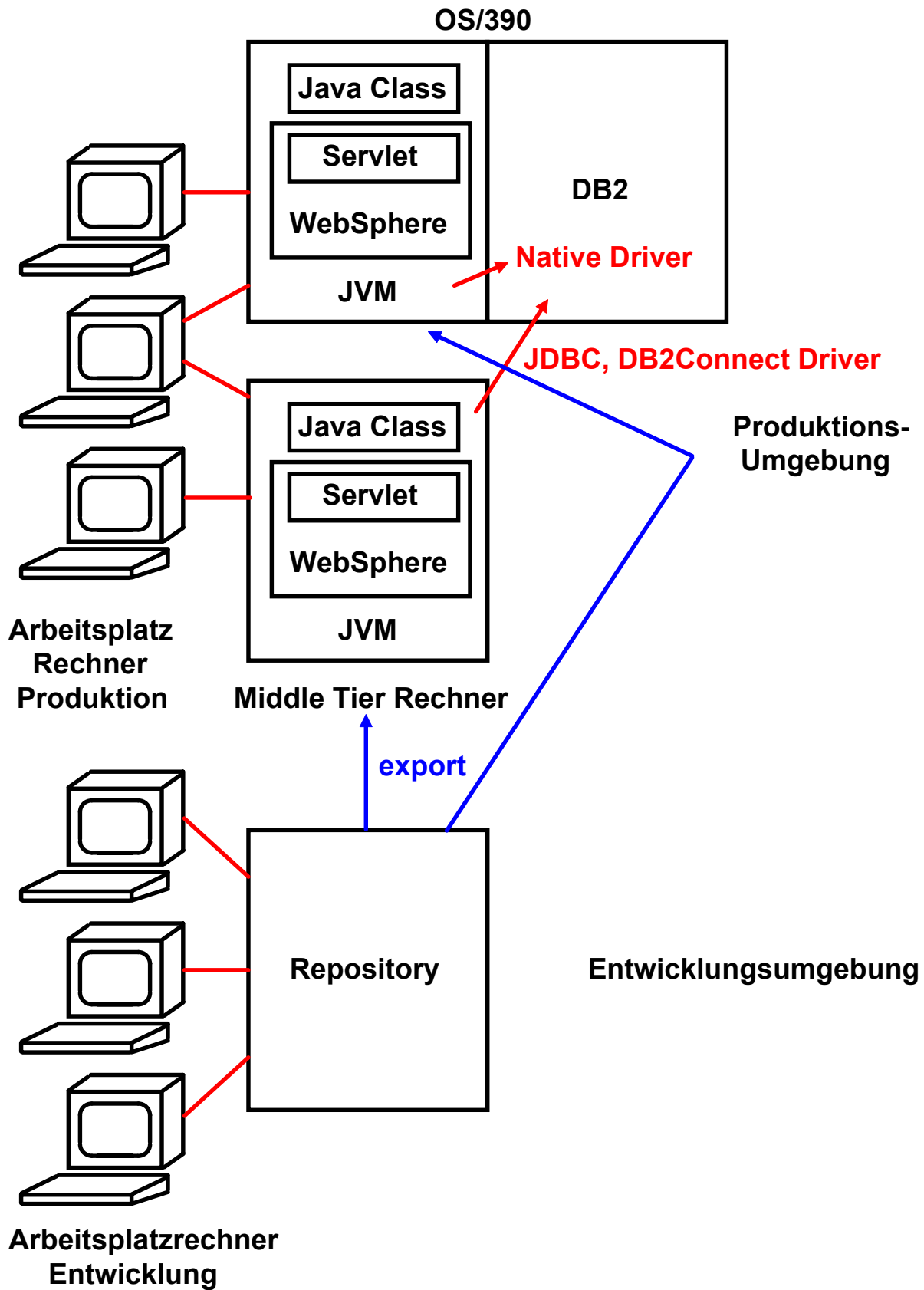
RACF / Crypto

Common Connector Framework / JCA

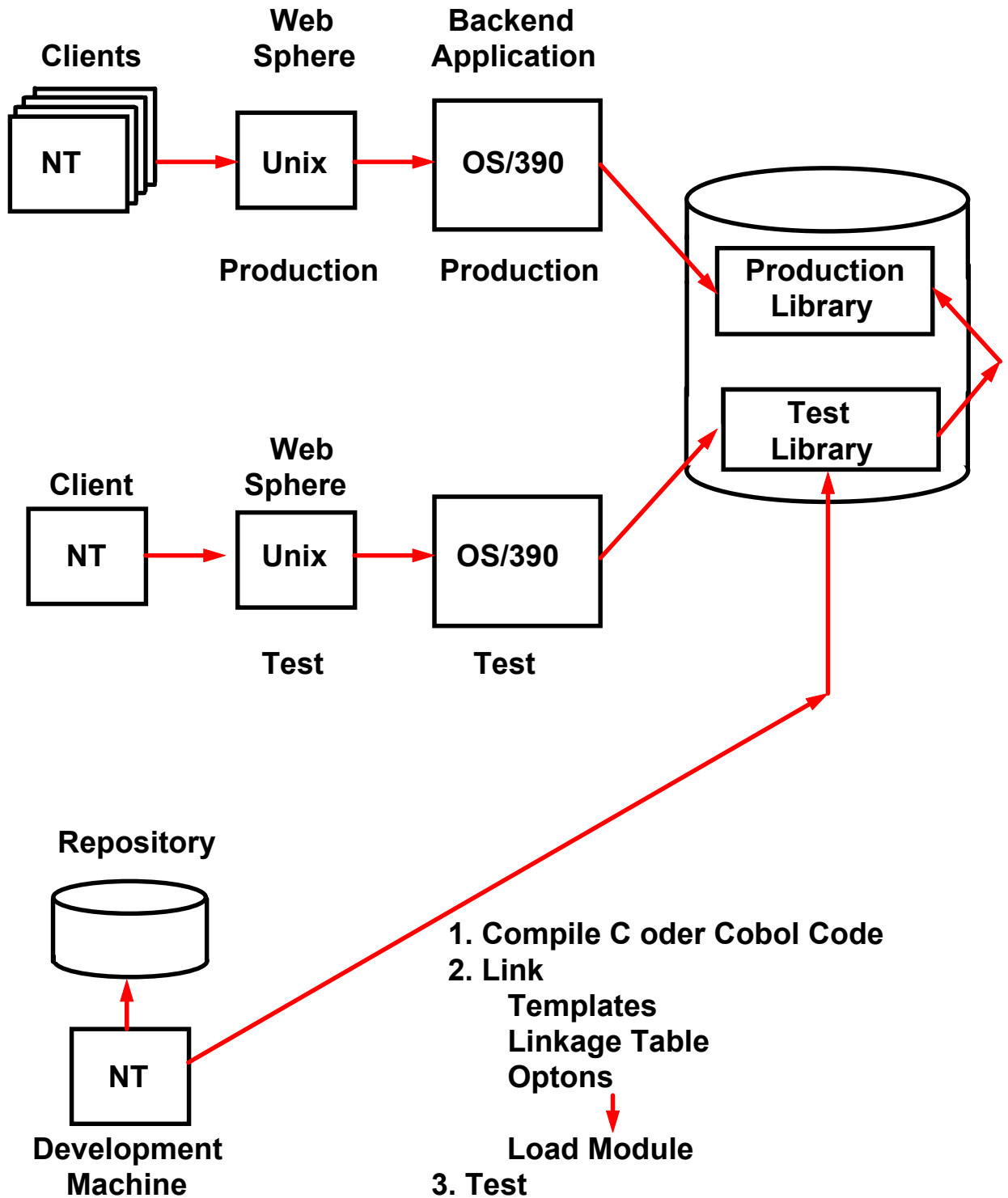
Virtuelle Server



Anwendungsarchitektur in Front-Office (mit Präsentationslogik) und Integrationsebene mit der Geschäftslogik sowie Processing mit der Transaktionsverarbeitung, wird verwendet um bestehende Kernanwendungssysteme durch den Integration-Layer zu wrappen



Trennung zwischen Entwicklung und Produktion



Translating Development Code into Production Code