

# Mainframe Internet Integration

Prof. Dr. Martin Bogdan  
Prof. Dr.-Ing. Wilhelm G. Spruth

SS2013

Java Connection Architecture Teil 4

JCA

# Enterprise Application Integration (EAI)

„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 EISs sind reinrassige Middleware; der Hersteller eines EIS nimmt an, dass ein Benutzer seine eigenen Anwendungen für das EIS schreibt. Manche Hersteller von EISs liefern umfangreiche betriebswirtschaftliche Anwendungen mit ihrem EIS aus. Diese EISs 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.

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.

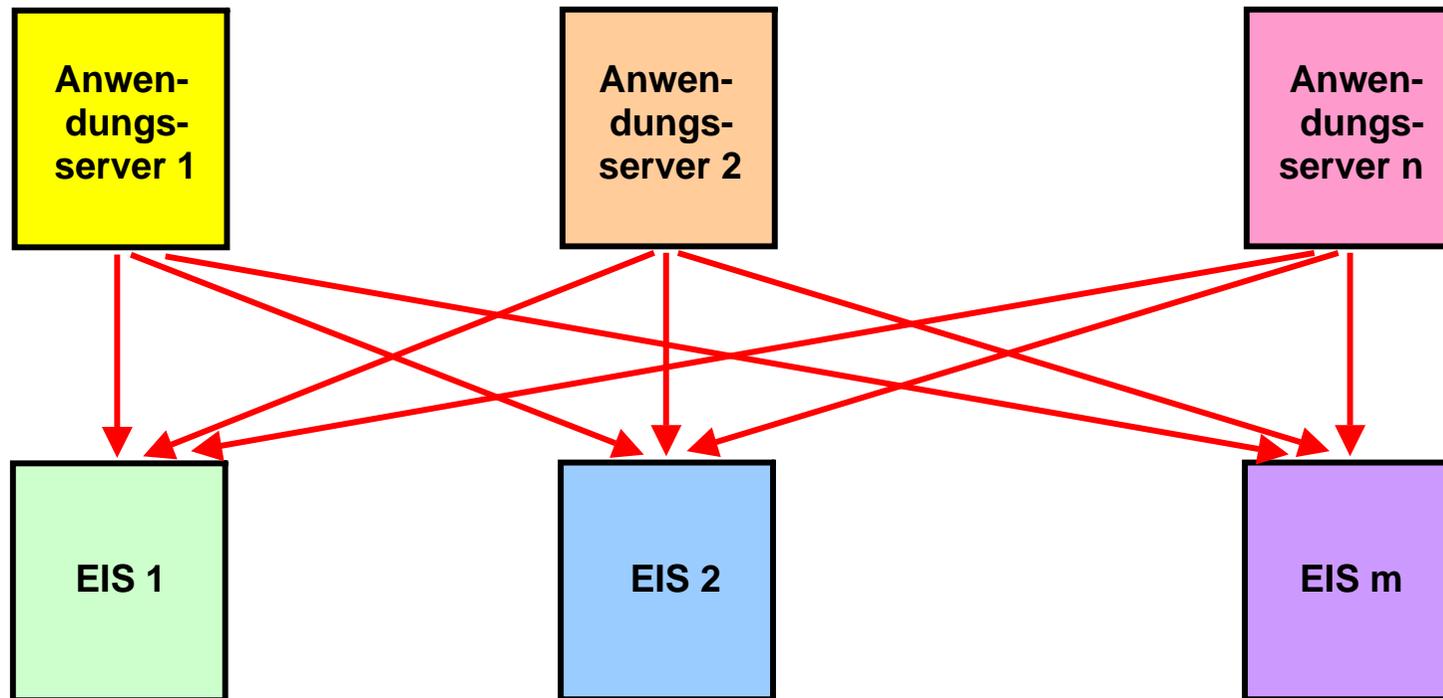
\*) Auch die Firma SAP liebt es, die Namen zu ändern. Bis Dezember 2003 wurde das Produkt unter dem Namen SAP R/3 geführt, bis 2007 unter dem Namen mySAP ERP. Die derzeitige Version heißt R/3 Enterprise oder SAP ERP.

Es existieren viele Anwendungen auf unabhängigen Anwendungsservern und viele unterschiedliche Enterprise Information Systems (z.B. CICS, SAP, IMS, MQSeries, DB2, ...)-

Anwendungen 1.1, 1.2, 1.3

Anwendungen 2.1, 2.2, 2.3

Anwendungen 3.1, 3.2, 3.3



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

# 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

# JEE Connector Architecture JCA

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

JCA ist eine Standard Achitektur 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

# JEE Connector Architecture

## JCA

Wichtigste Bestandteile der **JEE Connector Architecture** (JCA) sind:

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

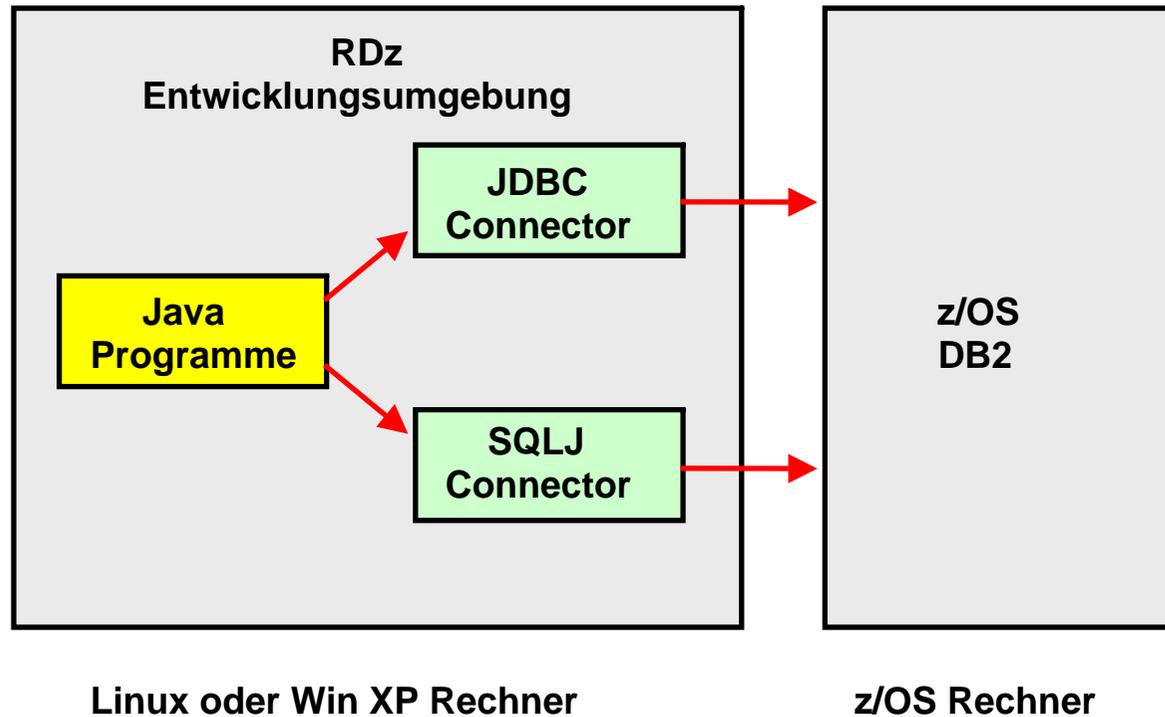
*A Resource Adapter is a system level software library that is used by an application server or client to connect to a Resource Manager. A Resource Adapter is typically specific to a Resource Manager. It is used within the address space of the client using it. An example of a resource adapter is the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

Die Common Client Interface (CCI) ist eine universelle API für die Interaktion mit beliebigen Resource Adaptern. 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.

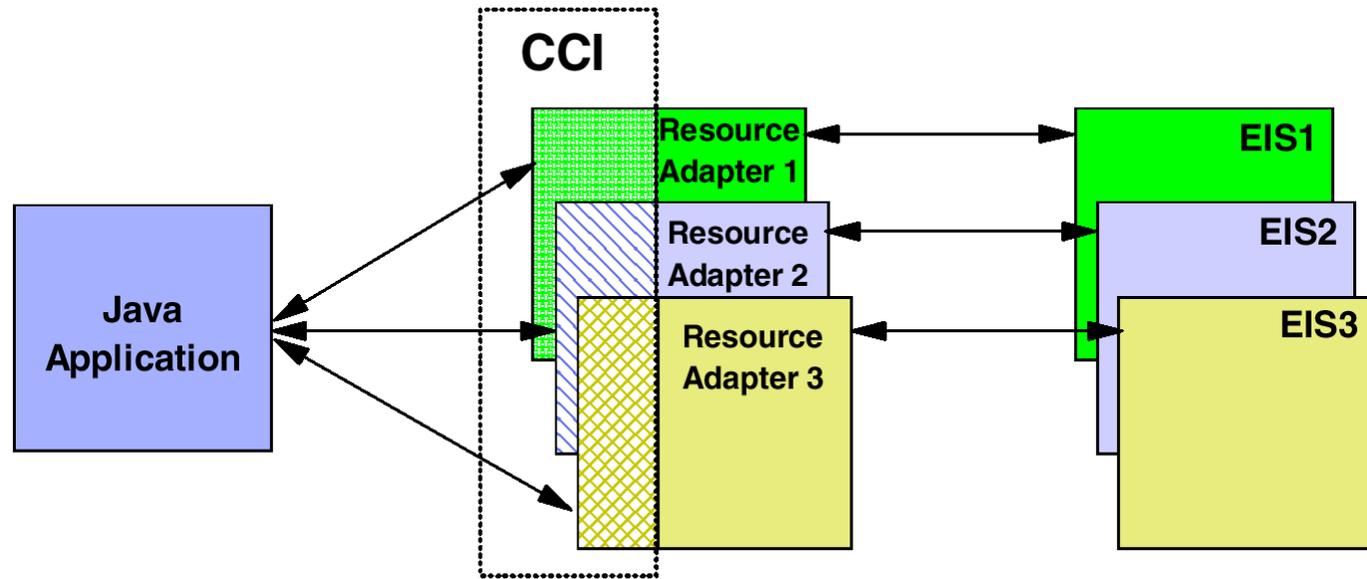


## Unterschiedliche Connector Arten

Es existieren viele unterschiedliche Konnektor Arten. Beispielsweise sind für den DB2 Zugriff JDBC und SQLJ Connectoren 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 Connectoren, spezifisch auch JDBC und SQLJ.

Die Common Client Interface (CCI) definiert eine Standard Client API. Die CCI ist eine EIS unabhängige API. Anwendungen können mit einer identischen API unterschiedliche Resource Adapter verwenden. Die JDBC Interface ist ein Beispiel für die CCI (mit einigen Besonderheiten).



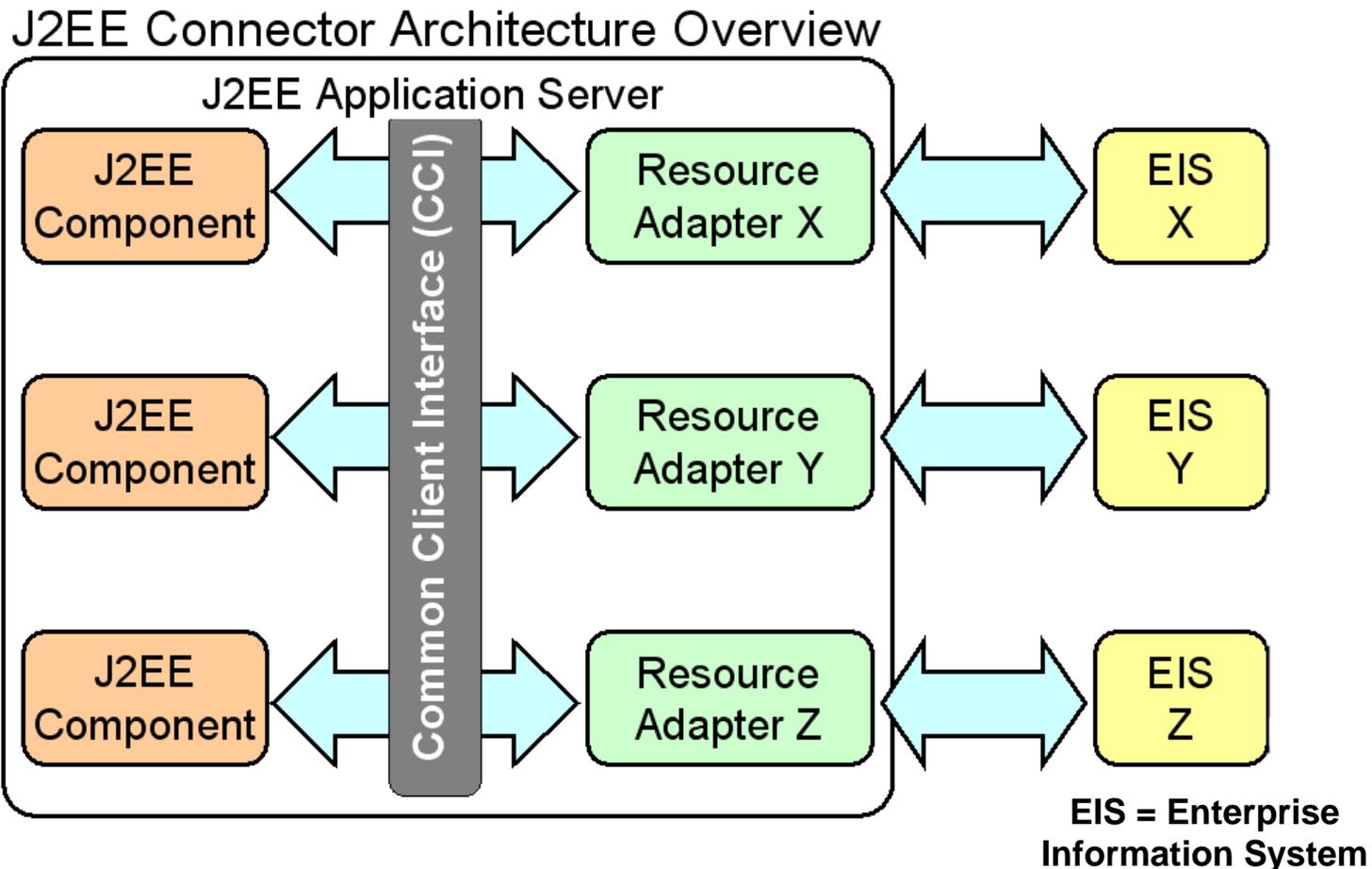
JEE Connector Architecture

CCI Common Client Interface  
EIS Enterprise Information System

## JCA Common Client Interface CCI

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.



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 Logic Middleware als Enterprise Information System (EIS) bezeichnet.

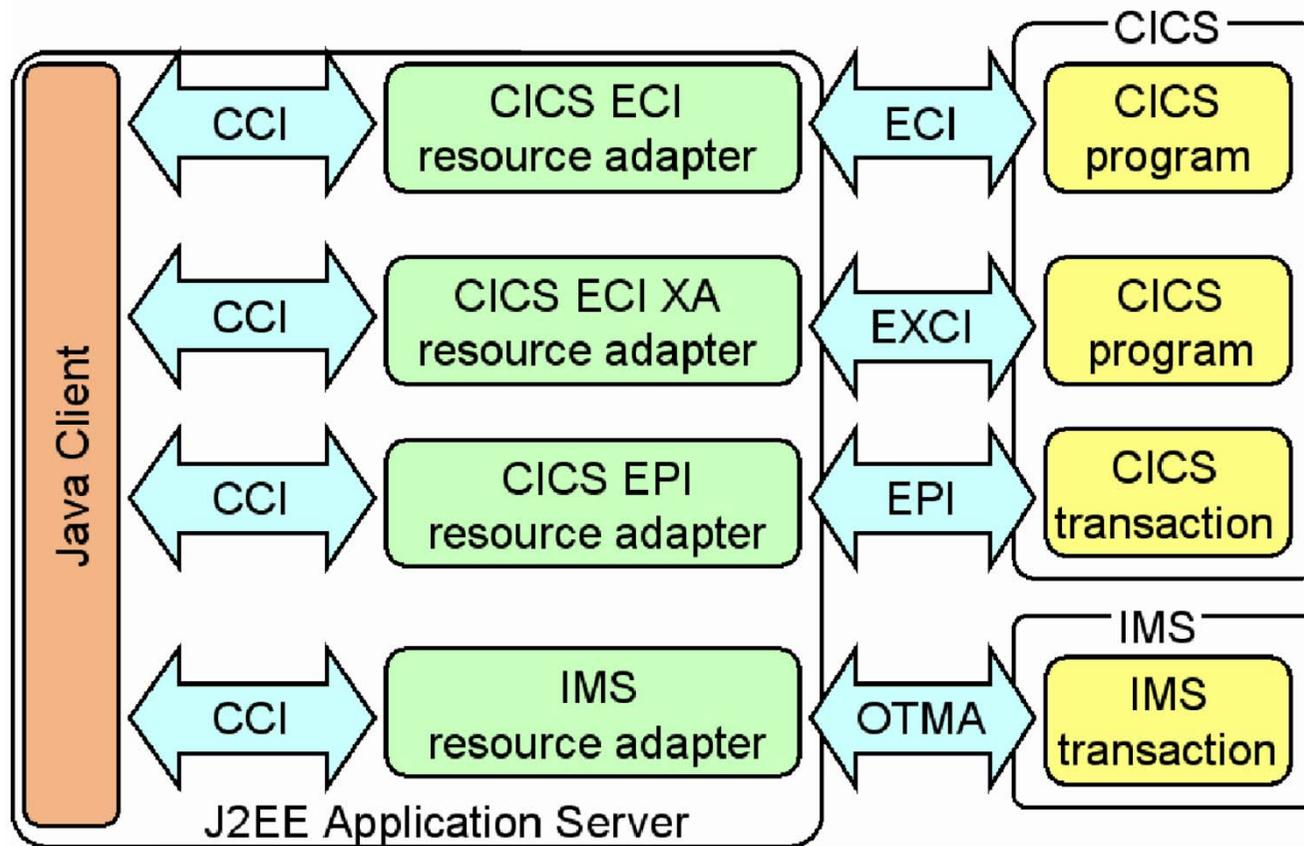
Resource Adapter werden von den Herstellern von Standard Middleware Komponenten zur Verfügung gestellt, z. B. von Oracle, IBM, SAP, und vielen anderen.

# **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 Plattform 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.**



Als Beispiel sind 4 verschiedene Resource Adaptern (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 wird verwendet, wenn der JCA Resource Adapter auf einem entfernten System installiert ist. Der EXCI wird verwendet, wenn der JCA Resource Adapter und CICS auf dem gleichen Mainframe installiert sind. In der Regel läuft der 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.

## Windows Client



### Java CCI Anwendung

```
Create Connection  
Create InteractionSpec  
Create Record  
Flow request  
Examine the Record  
Close the connection
```

## Windows

### CTG

Gateway  
daemon

Client  
daemon

Port  
2006

## z/OS

### CICS Region

SCSCPAA6

C  
O  
M  
M  
A  
R  
E  
A

ECIPROG  
program

## CTG Scenario für eine ECI CCI Anwendung

Dargestellt ist eine Java Client Anwendung und ihre CCI Connection zum CICS Transaction Gateway.

## CCI Sample For 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);

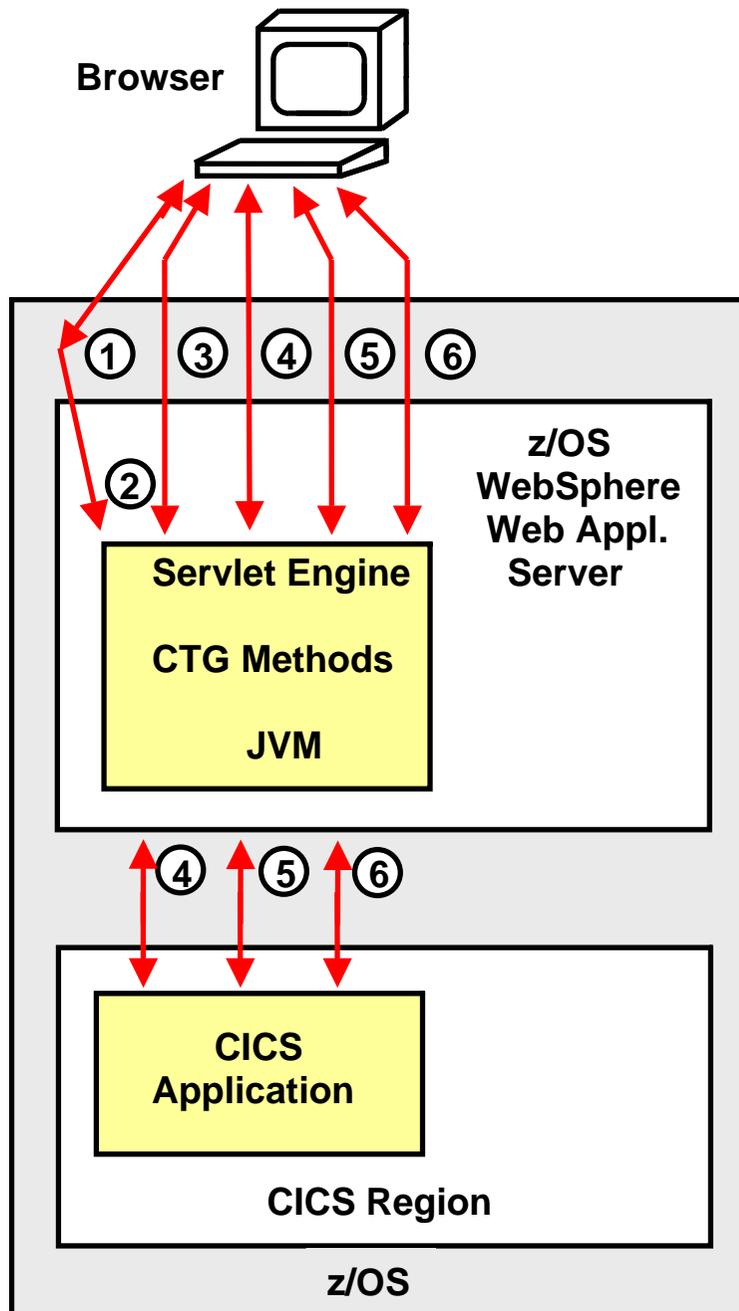
//Create an interaction
Interaction inter = conn.createInteraction();
ECIInteractionSpec iSpec = new ECIInteractionSpec();
iSpec.setFunctionName("CICSPROG");

//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());
```

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



## CICS Transaction Gateway

Dargestellt ist eine 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.

1,2 Der Browser stellt eine Verbindung zu dem Websphere Web Application Server her.

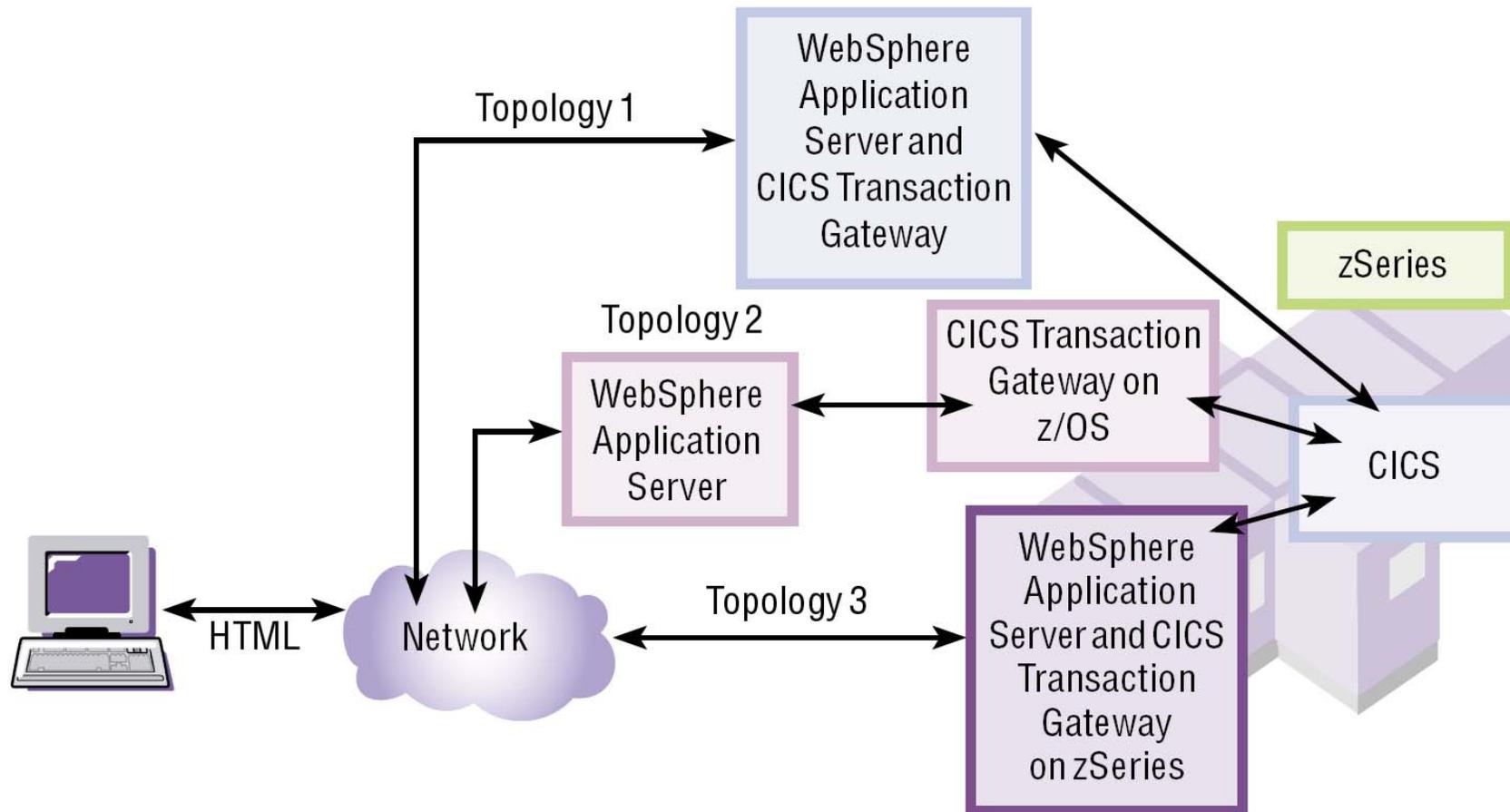
3 bis 6 Jeder weitere Schritt benutzt Servlets für die Präsentations-Logik und das CICS Transaction Gateway für den Aufruf der CICS Business Logik.

## Versionen des CTG

Das CICS Transaction Gateway (CTG) wird häufig als der Java Enterprise Edition (JEE)-Connector 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), Plattformen 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.

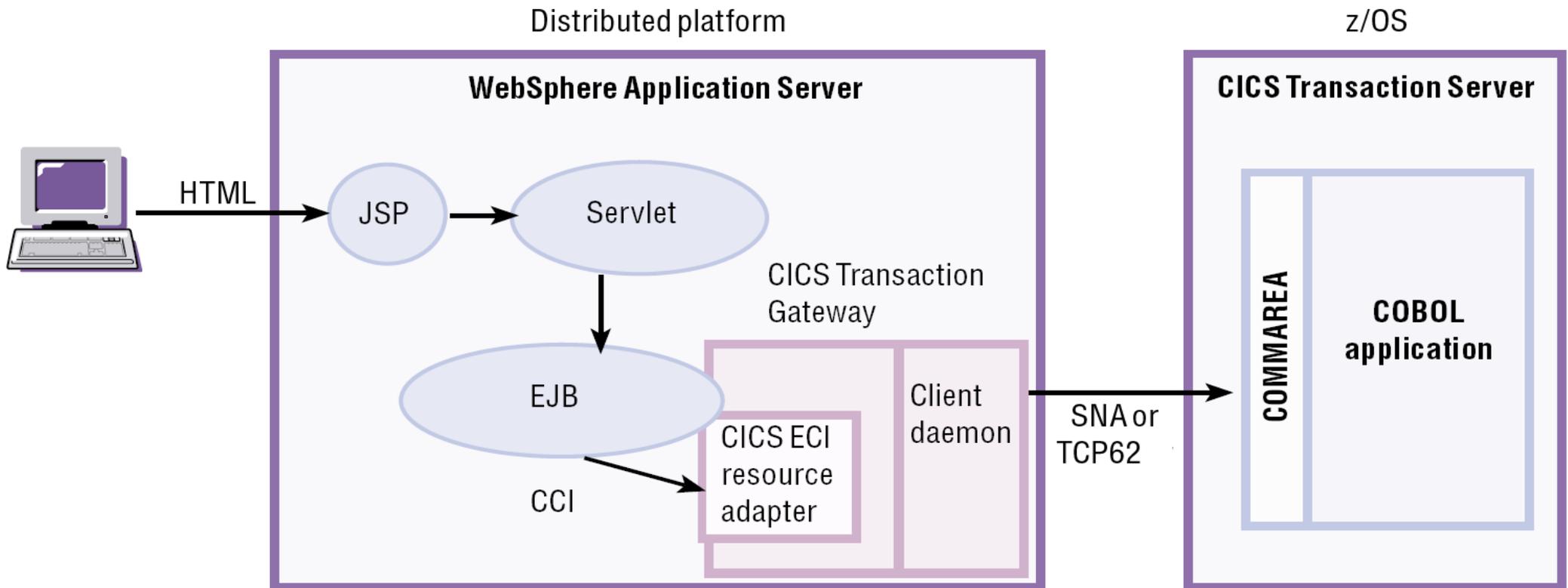
CTG ist in drei Varianten erhältlich, um verschiedenen Anforderungen und Einsatzszenarien unterzubringen.

- Am unteren Ende der Skala existiert die CTG Desktop Edition, eine Single-User-Lizenz für eine Installation auf einer einzelnen Workstations.
- Für kleine und mittelständische Unternehmen bietet die Multiplattform (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.

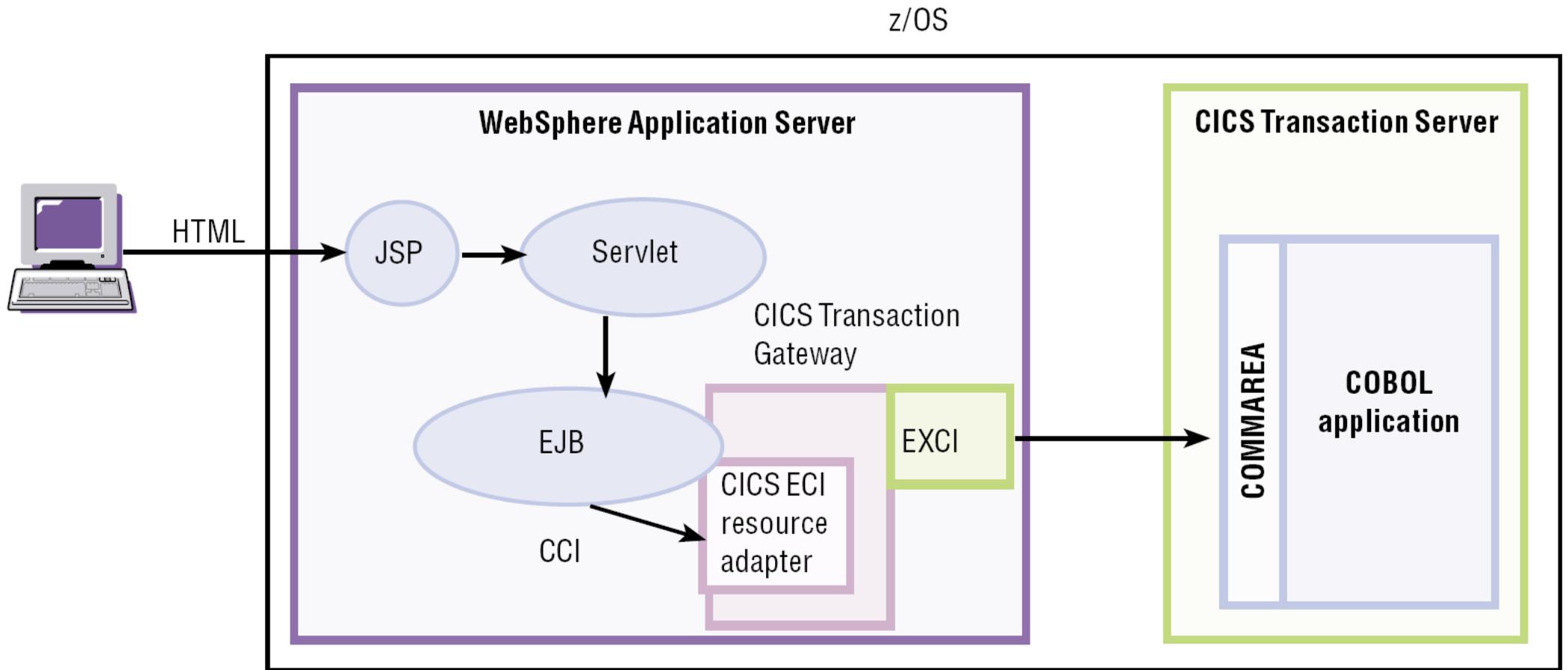
- **Topology 1:** WebSphere Application Server und CICS Transaction Gateway sind beide auf einer distributed (non-zSeries) Plattform installiert.
- **Topology 3:** WebSphere Application Server und CICS Transaction Gateway sind beide auf einem z/OS Server installiert.



## CICS Transaction Gateway Topologie 1

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

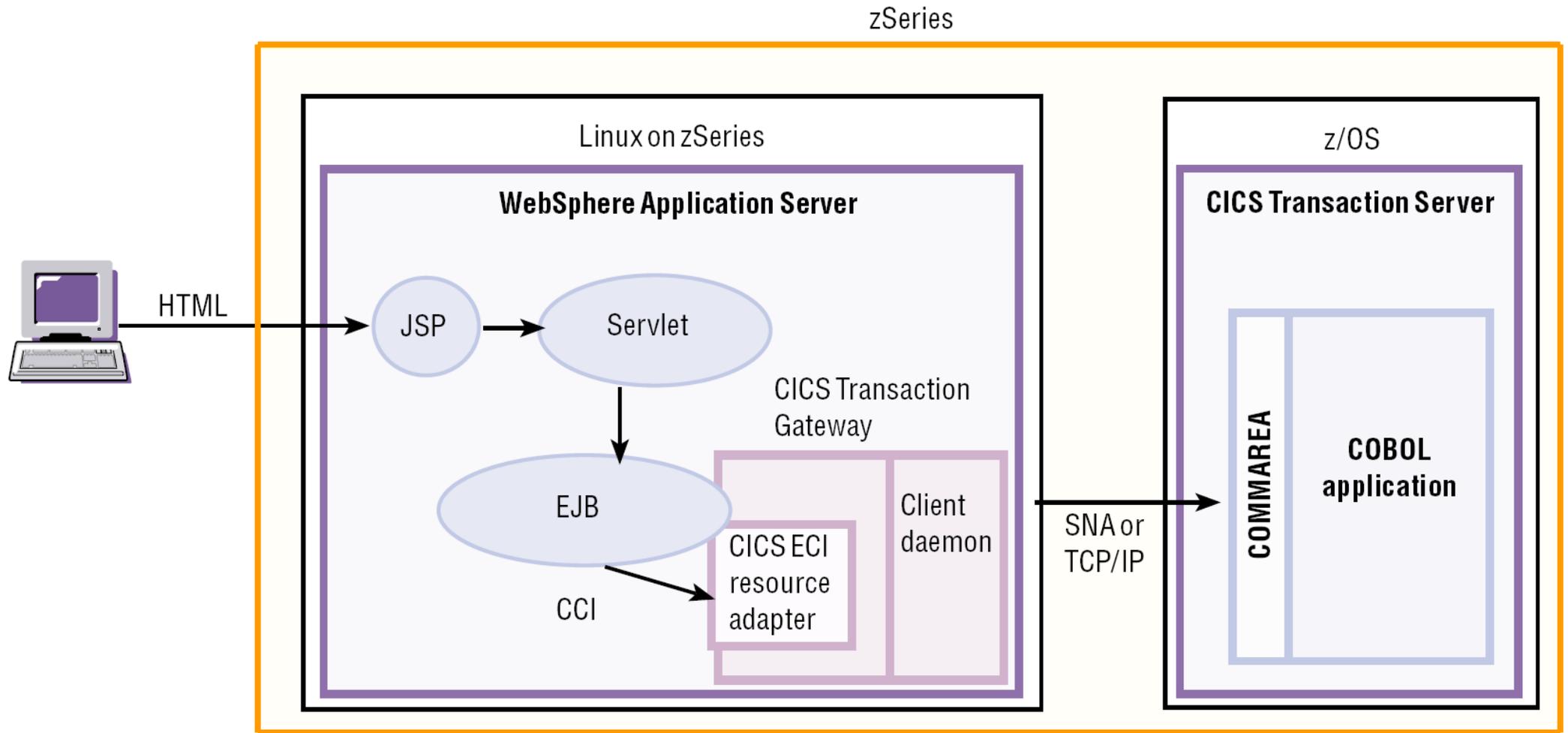
Gezeigt ist eine Enterprise JavaBeans (EJB) Anwendung, die den ECI Resource Adapter benutzt um auf eine COMMAREA-basierte CICS COBOL Anwendung zuzugreifen.



## CICS Transaction Gateway Topologie 2

In dieser 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 Address Space auf Programme innerhalb einer CICS Transaction Server Region zuzugreifen..



### CICS Transaction Gateway 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. .

# Java Cryptography Architecture

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

**Die Java Cryptography Architecture definiert die Abstimmung zwischen der Anwendung, dem Connector und dem Anwendungsserver, auf dem die Anwendung bereitgestellt wird.**

**JCA bedeutet alternativ JEE Connector Architecture und Java Cryptography Architecture.**

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