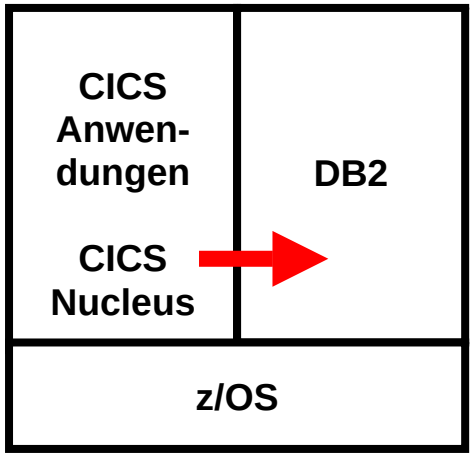
Mainframe Internet Integration

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

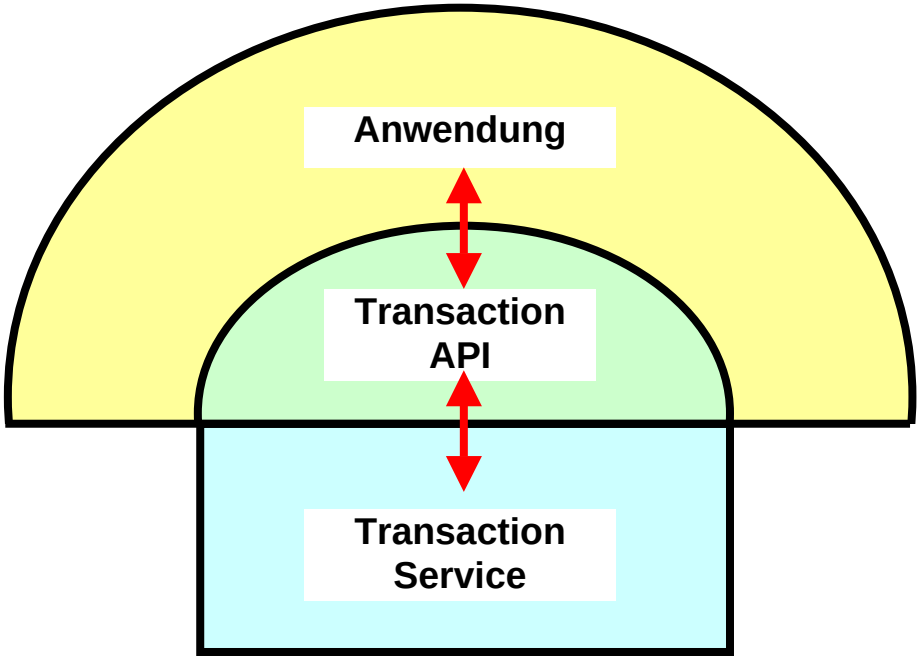
SS2013

Java Transaction Processing Teil 1

EJB Transaktionseigenschaften



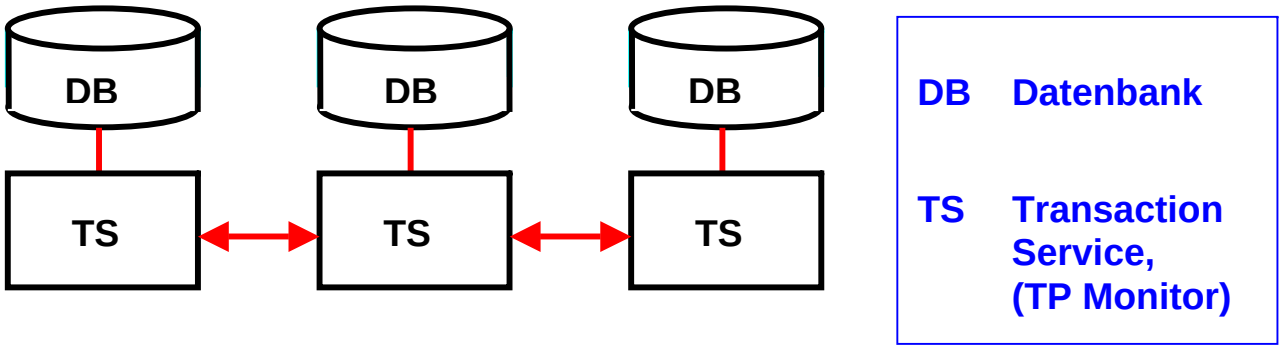
In einer einfachen Konfiguration greift eine transaktionale Anwendung auf eine einzige Data Bank (zum Beispiel DB2) zu, die auf dem gleichen Rechner untergebracht ist. Mehrere unabhängige Anwendungen können auf diese Datenbank zur gleichen Zeit zugreifen. DB2 enthält hierfür eine eigene Transaktionsverarbeitungsfunktion, die z.B. auch bei Stored Procedures eingesetzt wird. Diese ordnet den einzelnen Transaktionen je eine Signatur zu, überwacht die transaktionalen Zugriffe und ist für den Transaktionsabschluss und die Fehlerbehandlung zuständig.



Transaction Manager und Transaction Service

Formal unterscheidet man bei dem JEE Standard zwischen der Java Transaction API und einer Transaction Service Implementation. Die Java Transaction API ist Teil der JEE Spezifikation und ist die Schnittstelle zum Transaction Service.

Der Transaction Service implementiert einen Transaktionsmonitor vergleichbar zu CICS oder Tuxedo und ist die mit Abstand umfangreichste Komponente eines Web Application Servers. Der Transaction Service ist nicht Teil der JEE Spezifikation und ist eine proprietäre Implementierung des Herstellers des Web Application Servers. So verwendete früher der WebLogic Application Server den hauseigenen Tuxedo Transaktionsmonitor. Distributed WebSphere verwendete hier die distributed CICS Version.



Distributed Transaction System

In einem distributed Transaction System wird eine globale Transaktion auf mehreren Rechnern ausgeführt. Dabei können Datenbanken (DB) auf mehreren Rechnern mit ihrem eigenen lokalen Transaktionsmonitoren (hier als Transaction Service, TS, bezeichnet) an der Ausführung der Transaktion beteiligt sein. Eine globale Transaktion wird meistens mittels des Two-Phase-Commit Protokolls implementiert.

Enterprise JavaBeans und JEE Server unterstützen neben lokalen Transaktionen, bei denen nur auf eine lokale Datenbank zugegriffen wird, auch globale Transaktionen in verteilten Umgebungen mit mehreren Datenbanken.

Beispiele einer globalen Transaktion

Wenn Information in einer Datenbank geändert wird, muss sichergestellt sein, dass die Änderung korrekt erfolgt. Nehmen wir an, Sie unterhalten bei Ihrer Bank zwei Konten, ein laufendes Konto (Checking Account) und ein Anlagekonto (Savings Account), und Sie wollen Geld von Ihrem Anlagekonto auf Ihr laufendes Konto überweisen. Wird die Transaktion nur teilweise ausgeführt, indem das Geld von Ihrem Anlagekonto abgebucht, aber nicht Ihrem laufenden Konto gutgeschrieben wird, wird Sie das nicht zufrieden stellen.

Weiteres Beispiel: Sie wollen einen Flug von San Francisco nach Paris buchen, aber ein direkter Flug ist nicht verfügbar. Wenn Sie keine Buchungsbestätigung sowohl für die Verbindung San Francisco – Chicago, als auch eine weitere Buchungsbestätigung Chicago – Paris erhalten, werden Sie diesen Flug nach Paris nicht buchen (commit). Sie werden ein "roll back" für diesen Flug nach Paris durchführen, weil eine Buchungsbestätigung nur für einen Teil des Fluges für Sie nicht annehmbar ist.

In beiden Beispielen sind mehrere kleinere Teiltransaktionen erforderlich um eine gesamte (globale) Transaktion durchzuführen. Wenn ein Problem mit einer Teiltransaktion besteht, wollen Sie ein Commit der gesamten Transaktion (Geldtransfer, oder diesen spezifischen Flug nach Paris buchen) nicht durchführen. Statt dessen wollen Sie ein Rollback für jede Teiltransaktion durchführen. Für eine erfolgreiche Geldüberweisung oder Parisreise-Buchung wollen Sie sicherstellen, dass alle Teiltransaktionen koordiniert und gesteuert werden um die globale Transaktion erfolgreich durchzuführen.

Für eine derartige koordinierte globale Transaktionsverarbeitung enthält die JEE Platform ein “distributed Transaction Processing Environment”. Dieses besteht aus einem JEE Application Server, JEE Application Komponenten sowie einem JEE Connector Architecture (JCA) Resource Adapter. Hiermit können Resourcen über physische Rechnergrenzen hinweg transparent und koordiniert abegändert werden.

Distributed Transaction Processing (DTP)

In einer einfachen Konfiguration greift eine EJB Anwendung auf eine einzelne Data Bank (zum Beispiel DB2) zu, die auf dem gleichen Rechner untergebracht ist. Mehrere unabhängige EJBs können auf diese Datenbank zur gleichen Zeit zugreifen. DB2 enthält hierfür eine eigene Transaktionsverarbeitungsfunktion, die z.B. auch bei Stored Procedures eingesetzt wird. Diese ordnet den einzelnen EJB Transaktionen je eine Signatur zu, überwacht die transaktionalen Zugriffe und ist für den Transaktionsabschluss und die Fehlerbehandlung zuständig.

In einer komplexeren Situation greift eine EJB Transaktion auf mehrere “Resources” (Datenbanken, Files, Queues) zu, die von mehreren Transaction Processing Monitoren verwaltet werden, die evtl. wiederum auf getrennten Rechnern laufen und von unabhängigen Herstellern stammen. Eine derartige globale Transaktion erfordert ein “Distributed Transaction Processing” (DTP) Environment. Ein Beispiel ist die verteilte Flat Transaction und das 2-Phase Commit Protocol.

Two Phase Commit

Angenommen, ein CICS Programm will parallel eine Änderung einer CICS eigenen VSAM Datei und einer externen DB2 Datenbank durchführen. Das CICS Programm wird hierzu ein EXEC CICS Syncpoint Kommando ausführen. Dies löst eine 2-Phase Commit Operation aus. Siehe hierzu die Vorlesung aus dem Wintersemester, Thema Transaktions-verarbeitung, Teil 4, oder http://jedi.informatik.uni-leipzig.de/de/VorlesMirror/ii/Vorles/TwoPhaseCommit.pdf.

Existierende Transaction Monitore wie CICS, IMS und DB2 verfügen hierzu über eigene Commit Coordinator Komponentenen. Das Update der VSAM Datei und der DB2 Datenbank wird erst entgültig, wenn der CICS Commit Coordinator Phase 2 des 2-Phase Commit Protokolls erfolgreich abgeschlossen hat.

Neu entwickelte Resource Manager Produkte unter z/OS (z.B. WebSphere Enterprise Java Beans) verwenden z/OS Resource Recovery Services (RRS) an Stelle einer nicht vorhandenen eigenen Two-phase Commit Protocol Komponente. Vermutlich wird sich RRS in der Zukunft zu einer zentralen z/OS Komponente für die Transaktionssteuerung entwickeln.

Resource Recovery Services

In der Vergangenheit haben z/OS Resource Manager wie IMS, CICS und DB2 ihre eigenen Commit Coordinator Funktion für die Two-Phase Commit Verarbeitung entwickelt. Hierbei übernimmt einer der beteiligten Resource Manager die Rolle des Commit Coordinators. Dies erforderte die Entwicklung von getrenntem Commit Coordinator Code jeweils für IMS, CICS und DB2.

Mit der wachsenden Anzahl verfügbarer z/OS Resource Manager entstand das Bedürfnis für einen generischen Commit Coordinator (von IBM als Sync Point Manager bezeichnet). Diese Rolle übernimmt eine neue z/OS Komponente, die Resource Recovery Services (RRS). Heute können Resource Manager wie IMS oder CICS die RRS Komponente an Stelle ihrer eigen Commit Coordinator Funktionen verwenden. Neue Entwicklungen wie WebSphere unter z/OS müssen RRS verwenden.

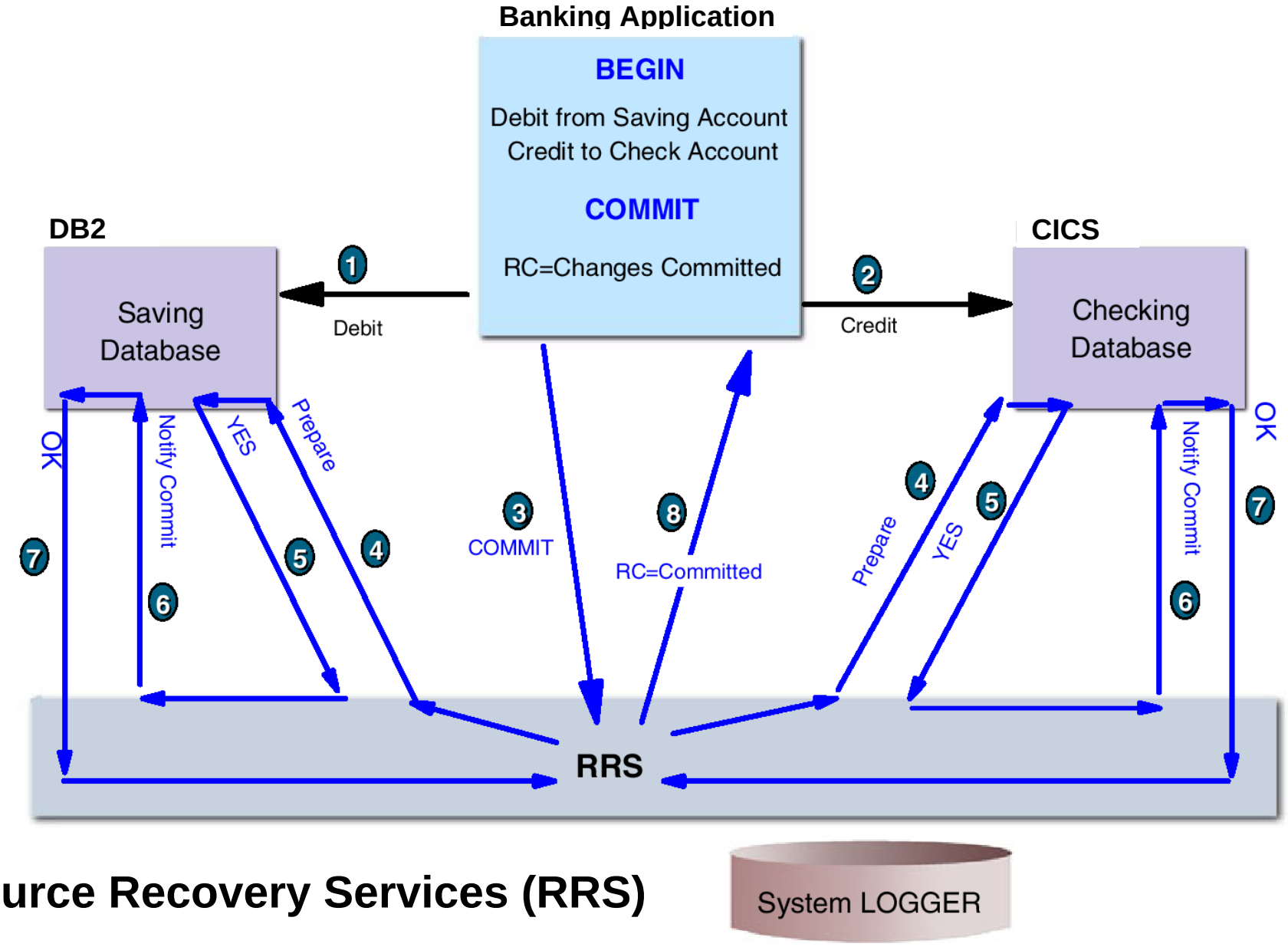
Das CICS Transaction Gateway kooperiert mit dem

- **Java Transaction Manager einer beliebigen JEE Platform (u.a. WebSphere), den**
- **Resource Recovery Services (RRS) unter z/OS, und**
- **CICS**

um konsistente Änderungen für CICS und andere geschützte Resourcen durchzuführen.

Vom Standpunkt von CICS aus gesehen, agiert RRS als ein "external coordinator" für die Änderung oder Recovery von Resourcen. Vermutlich wird sich RRS in der Zukunft zu einer zentralen z/OS Komponente für die Transaktionssteuerung entwickeln.

Literature: http://www.redbooks.ibm.com/redbooks/pdfs/sg246980.pdf



Resource Recovery Services (RRS)

RRS ist ein 2-Phase Commit Coordinator Service, der beliebigen z/OS Subsystemen zur Verfügung steht, so auch für WebSphere unter z/OS. CICS hat eine eigene 2-Phase Commit Komponente, kann aber auch RRS benutzen.

X/Open

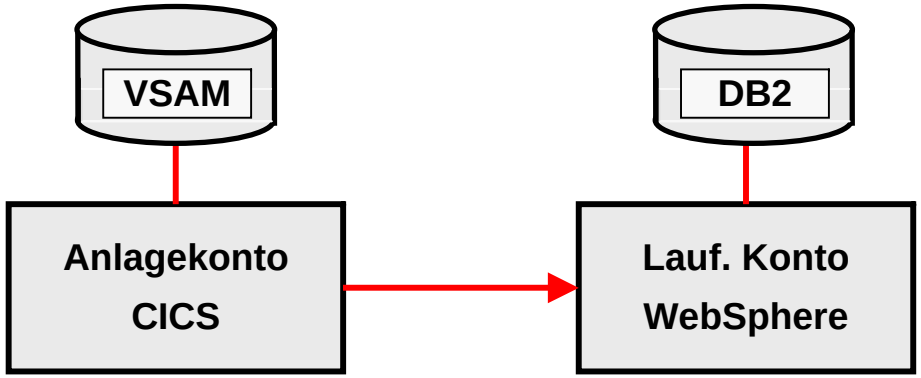
Das X/Open Consortium wurde 1984 von mehreren UNIX Systems Herstellung mit der Zielsetzung gegründet, offene Standards auf dem Gebiet der Informationstechnologie zu etablieren. Spezifisch wurde eine einheitlich Unix Spezifikation für die Interoperability von Anwendungen und die Portierung von Software angestrebt.

Es existieren eine ganze Reihe von unterschiedlichen X/Open Standards. Sie gelten für alle Sprachen, einschließlich Java.

X/Open XA (kurz XA) ist ein von X/Open spezifizierter Informatik-Standard für Distributed Transaction Processing (die Abarbeitung von Transaktionen, die über mehrere Systeme verteilt sind). Mithilfe des XA-Standards können verschiedene „Ressource Manager“ (wie Datenbanken, Transaktionsmonitore, Applikationsserver, Messaging Systeme) innerhalb einer Transaktion angesprochen werden, unter Bewahrung der ACID-Eigenschaften von Transaktionen.

The XA Specification beschreibt, was ein Ressource Manager leisten muss um gemäß XA verteilte Transaktionen zu ermöglichen. Damit wird die Vorgangsweise und Schnittstelle zwischen einem globalen „Transaktion Manager“ und den lokalen Resource Managern festgelegt. Ressource Manager, die den Standard erfüllen nennt man „XA compliant“. Der XA-Standard basiert auf dem Zwei-Phasen-Commit-Protokoll.

Die Java Implementierung des X/Open XA definiert die Schnittstelle (Interfaces und Exception-Klassen), über die Java-Programme mit Transaktionsmanagern kommunizieren können. Transaktionsmanager ihrerseits implementieren üblicherweise die Java Transaction API (JTA) Programmierschnittstelle, Teil des EJB Standards. JTA stellt die Standardschnittstelle für XA fähige Transaktionsserver dar.



Globale Transaction und der two-phase commit Process

Ein JEE-konformer Anwendunggsserver (wie z.B. der WebSphere Application Server) benutzt den Java Transaction Manager für die Kommunikation zwischen den Anwendungskomponenten (z.B. Java Servlets oder Enterprise Java Beans) und externen Resource Managers (z.B. CICS oder DB2). Die Koordination einer globalen Transaktion erfolgt typischerweise über Java Connection Architecture (JCA) konforme Resource Adapter, wie z.B. dem CICS Transaction Gateway.

Wenn ein Transaction Manager (TM) eine globale Transaction mit mehr als einem Resource Manager koordiniert, verwendet die Commit Coordinator Funktion des Transaction Managers das Two-Phase Commit Protokoll.

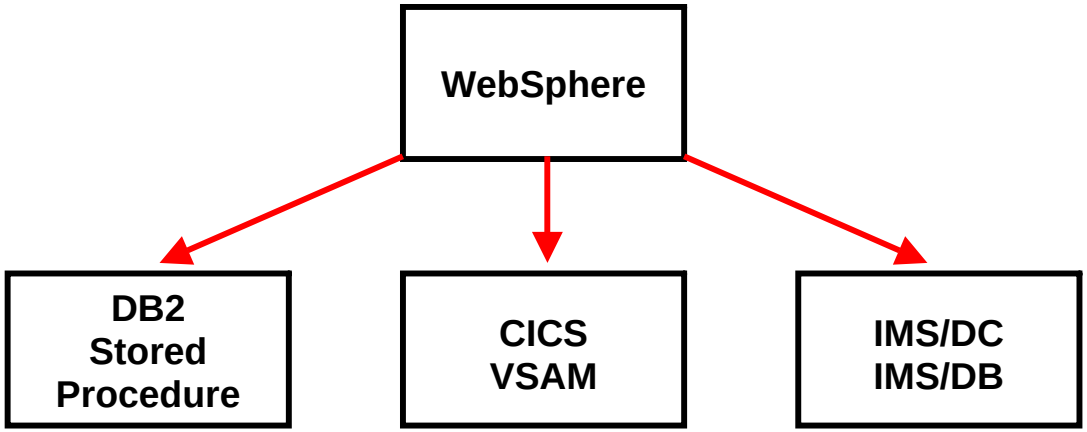
In dem vorher erwähnten Bank-Beispiel unterhielten Sie bei Ihrer Bank zwei Konten, ein laufendes Konto (Checking Account) und ein Anlagekonto (Savings Account). Sie wollten Geld von Ihrem Anlagekonto (Savings Account) auf Ihr laufendes Konto (Checking Account) überweisen. Angenommen, Ihr Anlagekonto wird von einer CICS Anwendung bearbeitet, deren Daten in einem VSAM Data Set gespeichert sind, und Ihr laufendes Konto wir von einer z/OS WebSphere EJB Anwendung bearbeitet, deren Daten in einer DB2 Datenbank gespeichert sind. In diesem Fall würde die Commit Coordinator Funktion des WebSphere Transaction Managers die Änderungen zwischen VSAM und DB2 koordinieren.

Transaktionale Konnektoren

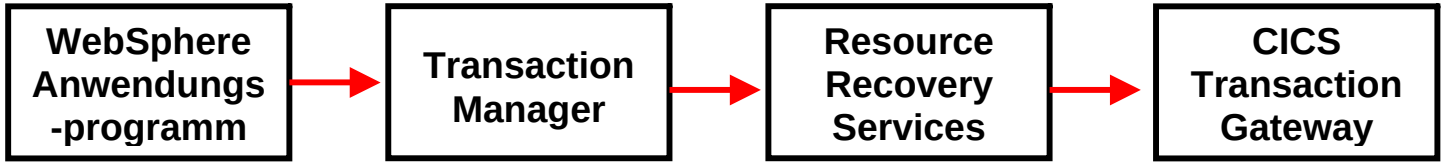
Konnektoren sind spezielle EJBs, welche eine Verbindung zu einem Backend System wie CICS oder DB2 herstellen. WebSphere für z/OS unterstützt zwei Arten von Konnektoren: non-transactional und transaktional. Die Connector Verarbeitung variiert je nach Konfiguration für jede installierte Instanz eines Connectores.

Die Transaktion Policy der EJB bestimmt, ob die Verarbeitung im Rahmen einer globalen Transaktion ausgeführt wird, oder nicht. Wenn ja wird der Connector ebenfalls eine globalen Transaktionsverarbeitung durchführen. Für transaktionale Connectoren wird die Art der Transaktionsverarbeitung zu dem Zeitpunkt festgelegt, wenn eine Anforderung an das Ziel Enterprise Information System (EIS) gesendet wird.

z/OS WAS ist in der Lage, in Cooperation mit z.B. CICS eine distributed Transaction durchzuführen, die das 2-Phase Commit Protokoll erfordert. Diese benutzt den transaktionalen Typ von Connector. Letzterer ist konfiguriert, mit der Ressource Recovery-Service (RRS) Komponente von z/OS eine Two-Phase Commit Verarbeitung durchzuführen. RRS beinhaltet die Commit Coordinator Funktion für die 2-Phase Commit Verarbeitung; z/OS WAS beinhaltet keine eigene Commit Coordinator Funktion,



Ein WebSphere Anwendungsprogramm führt eine globale Transaktion durch, an der DB2, CICS und IMS/DC als Resource Manager beteiligt sind.



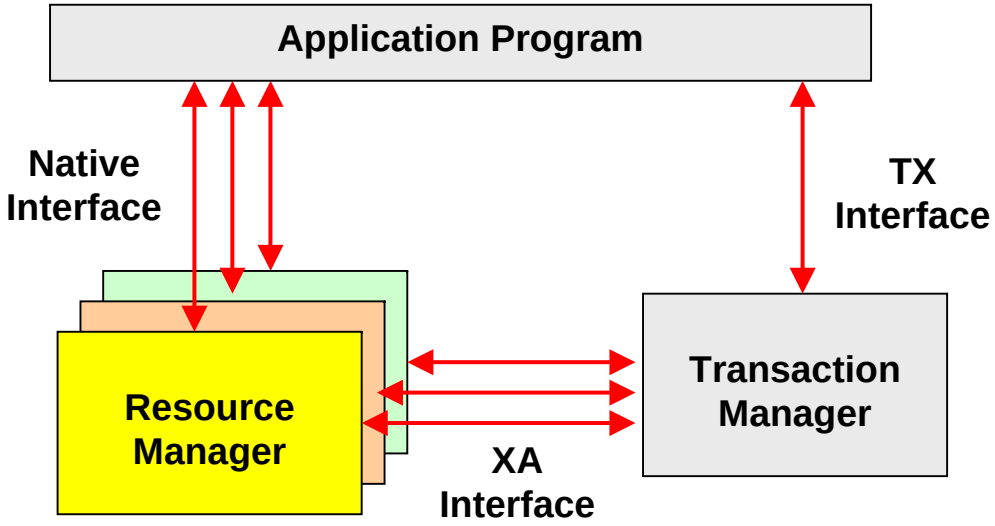
WebSphere übernimmt die Rolle des Transaktionsmanagers

Die 2-Pase Commit Funktion wird durch Resource Recovery Services implementiert

Das CICS Transaction Gateway ist ein JCA konformer KonneKtor, dessen Konfiguration für eine globale Transaktion enabled wurde.

Die JCA Konnektoren für DB2 und IMS/DC wurden ebenso enabled.

X/Open Distributed Transaction Processing Modell



Das X/Open Distributed Transaction Processing (DTP) Modell ist eine Specifikation für das Management von Transaktionen, deren Operationen auf unterschiedlichen Rechnern oder auf Datenbanken unterschiedlicher Hersteller (z.B. Oracle, DB2) erfolgen. DTP besteht aus 3 Kernkomponenten:

- Einem Application Program. Dieses definiert Transaktionsgrenzen und spezifiziert die Aktionen, die eine Transaktion darstelllen.
- Resource Managers (RM) sind Subsysteme oder Komponenten wie CICS, IMS, oder DB2. Diese verwalten Resourcen, die von den Transaktionen in Anspruch genommen werden, wie z.B. Datenbanken, Files oder Queues. Alternativ kann ein RM auch ein Data Access Komponente wie ODBC oder JDBC sein.
- Ein Transaction Manager (TM). Dieser markiert Transaktionen mit Idendifiern, überwacht den Fortschritt und ist zuständig für den erfolgreichen Abschluss (commit), ode alternativ für Fault Recovery (Rollback). Ein TM beinhaltet vor allem einen 2-Phase Commit Coordinator.

Vorsicht: Als Transaktion Manager wird manchmal ein Transaktionsserver wie CICS bezeichnet. X/Open DTP bezeichnet statt dessen als Transaktion Manager eine Funktion, die einen 2-Phase Commit Coordinator enthält.

X/Open Interfaces

Der X/Open Standard ist für beliebige Sprachen anwendbar, einschließlich Java.

Das Application Program, der Resource Manager (RM) und der Transaction Manager (TM) kommunizieren über drei spezifische Interfaces: Native, TX, and XA.

Über die Native interface sendet das Application Program Requests directly an die Resource Managers. Diese Schnittstelle ist Resource Manager specifisch; embedded SQL oder EXEC CICS sind Beispiele.

Die TX Interface ist das Medium zwischen dem Application Program und dem Transaction Manager, zum Beispiel der Commit Coordinator Komponente von CICS. Die API verwendet TX Calls (über dieTX Interface), um den Transaction Manager zum Start, Commit oder Roll Back einer globalen Transaktion aufzufordern. Diese Schnittstelle ist Transaction Manager specifisch.

Die XA Interface ist das Medium zwischen den Resource Managern und dem Transaction Manager. Mit Hilfe von XA Calls fordert der Transaction Manager den Resource Manager zum Transactions Start, Commit, oder Roll Back auf. Der Transaction Manager ist auch für eine Recovery zuständig.

X/Open XA Interface

Die X/Open XA Interface ist ein offener Standard um Änderungen für multiple Resources zu koordinieren, wobei die Integrität dieser Änderungen gewährleistet wird. Beispiele sind Resourcen, die eine persistente Speicherung beinhalten wie Datenbanken, Queues und File Systeme. Mehrere unterschiedliche Transaction Processing Monitore benutzen typischerweise die XA Interface für eine gegenseitige Kommunikation. Ein gleichzeitiger Zugriff auf mehrere Datenbanken ist damit möglich.

Der X/Open Standard ist für beliebige Sprachen anwendbar, einschließlich Java..