

**Enterprise Computing  
Einführung in das Betriebssystem z/OS**

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

**WS2012/2013**

**CICS Communication Teil 3**

**Multiregion and Intersystem Communication**

# Typische CICS oder IMS Transaktionen

Als Pfadlänge bezeichnet man die Zahl der während der Abarbeitung einer Transaktion ausgeführten Maschinenbefehle. Eine typische CICS oder IMS Transaktionen hat eine Pfadlänge von mehr als 100 000 Maschinenbefehlen.

Transaktionen mit einer Pfadlänge von 1 Million Maschinenbefehlen sind nicht ungewöhnlich.

Annahme: Eine CPU führt 1 Milliarde Maschinenbefehle / Sekunde aus.

Bei einer Pfadlänge von 250 000 Maschinenbefehlen sind 4 000 Transaktionen / Sekunde theoretisch möglich, wenn man Overhead vernachlässigt.

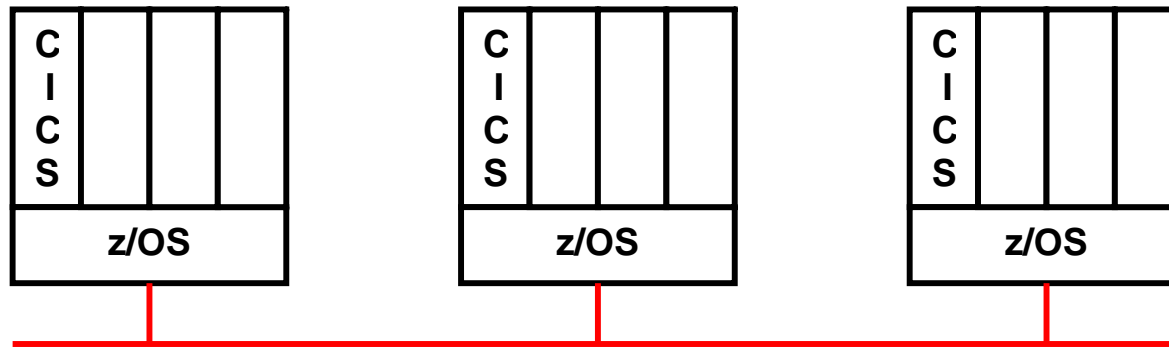
Große Installationen bewältigen 5000 Transaktionen / Sekunde. Spitzenwert von 9 000 Transaktionen / Sekunde sind schon aufgetreten. Das IBM Entwicklungslabor in Hursley, Südengland hat 19 000 Transaktionen / Sekunde demonstriert.

Derartig hohe Transaktionsraten bedingen mehrere CPUs in einer „Symmetrischen Multiprozessor“ (SMP) Konfiguration und häufig auch mehrere Mainframe Rechner..

## Typische Anzahl der Komponenten eines Transaktionsverarbeitungssystems

Hardware	klein/einfach	mittel	groß/komplex	
Endbenutzer	100	10 000	> 100 000	
CPUs	3	20	> 100	
Plattenspeicher TeraByte	1 - 10	10 - 100	100 – 5 000	
Magnetband Terabyte	100	100 - 1000	> 10 000	
<b>Software</b>				
Transaktions- programme X)	400	4 000	40 000	X) einschl. Berichte, Maps
Quell- programme XX)	1 000	10 000	100 000	XX) einschl. alte Versionen

Pfadlänge pro Transaktion: 100 000 - 1 000 000 Maschinenbefehle,  
effektive Plattenspeicherzugriffe; 5 - 20



Mehrere z/OS Rechner können entweder über TCP/IP (Internet) miteinander verbunden sein, oder eine spezielle als „Sysplex“ bezeichnete Integrationssoftware benutzen.

## CICS Interprocess Communication

Häufig ist gewünscht, dass 2 CICS Transaktionen, die auf unterschiedlichen CICS Transaktions-Monitoren laufen, miteinander kommunizieren können. Es existieren 2 Arten von CICS Interprocess Communication:

- multiregion operation (MRO)
- Intersystem communication (ISC)

**Multiregion Operation (MRO)** wird für die Communication von zwei CICS Regions benutzt, die sich auf dem gleichen Mainframe System befinden.

Ein Spezialfall ist ein Verbund aus mehreren Mainframe Rechnern, wobei auf jedem Rechner eine CICS Region installiert ist.

Für einen Cluster bestehend aus mehren Mainframes innerhalb eines Rechenzentrums existiert eine als „Parallel Sysplex“ bezeichnete Integrationssoftware, die ebenfalls MRO zwischen unterschiedlichen z/OS Instanzen ermöglicht. Sind die z/OS Rechner in geographisch voneinander getrennten Rechenzentren untergebracht, spricht man von einem Geographically Dispersed Parallel Sysplex (GDPS), Das Thema Sysplex wird in einem späteren Modul behandelt.

**Intersystem Communication (ISC)** wird für die Communication von zwei CICS Systemen benutzt, die lediglich über TCP/IP und das Internet miteinander verbunden sind. Hierbei können die CICS Instanzen alle auf z/OS, oder aber auch auf Unix oder Windows Rechnern installiert sein.

# CICS Multiregion Operation

CICS spezifische MRO-Standards sind:

## Transaction Routing

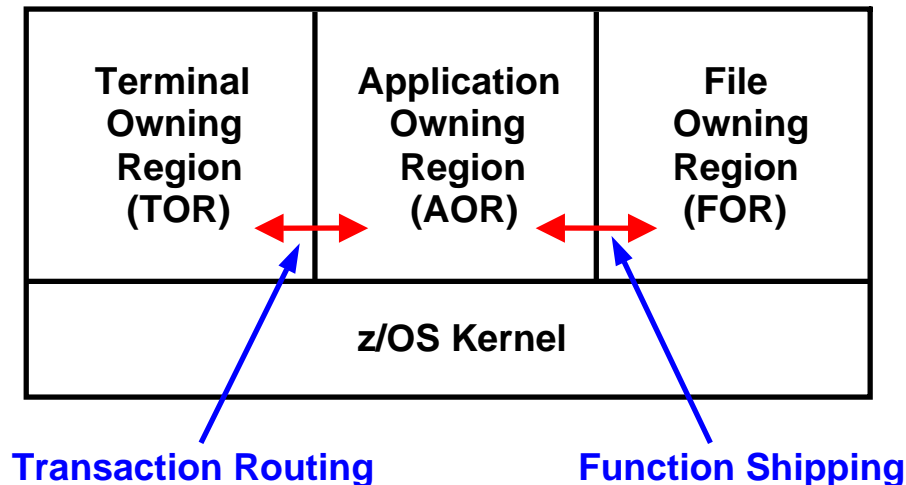
Transaktionen (Eingabe vom Klienten) können zwecks Ausführung von einem CICS-System einem anderen CICS-System unverändert übergeben werden

## Function Shipping

Eine Anwendung in einem CICS-System kann auf Daten eines anderen CICS-Systems zugreifen

# Multiregion Operation

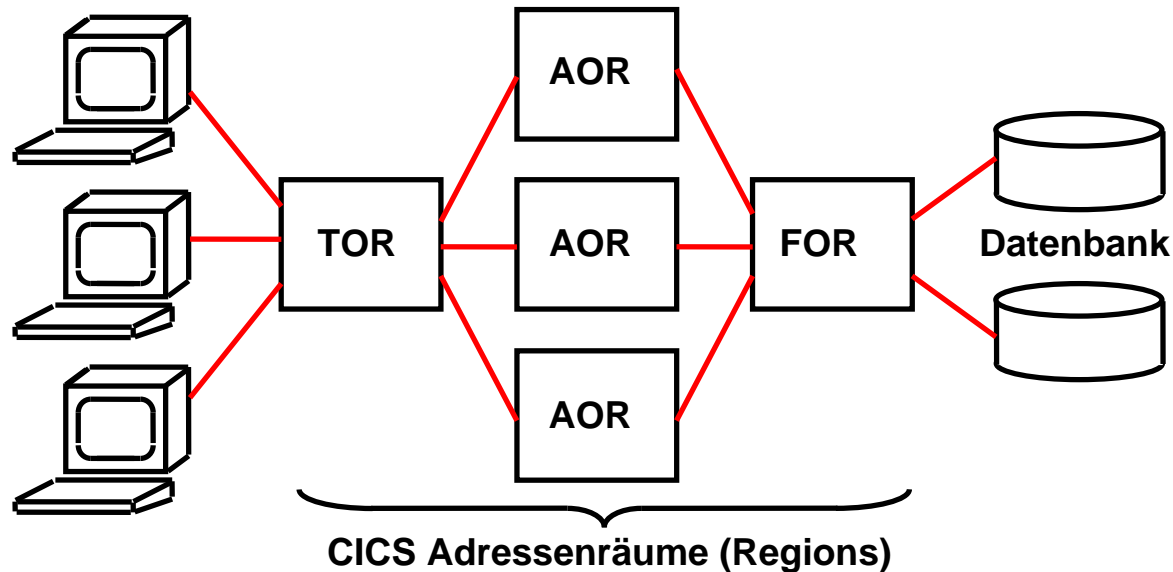
CICS multiregion operation (MRO) ermöglicht es zwei oder mehr CICS Systemen, die auf dem gleichen z/OS System oder innerhalb des gleichen Parallel Sysplex laufen, miteinander zu kommunizieren.



Häufig wird CICS auf 3 Regionen (virtuelle Adressenräume) innerhalb des gleichen Rechners aufgeteilt, die jede über einen eigenen CICS Transaktionsmonitor verfügen, und über MRO miteinander kommunizieren.

Die Terminal Owning Region verwaltet alle angeschlossenen Terminals. Sie gibt über das „Transaction Routing“ Protokoll die Transaktion an die Application Owning Region weiter.

Die Application Owning Region führt die Anwendung aus. Wenn diese Daten (z.B. von einem VSAM Data Set) benötigt werden, werden diese von der File Owning Region über das Function Shipping Protokoll zur Verfügung gestellt.



## Verteilung auf mehrere Adressenräume

Gezeigt ist ein Beispiel mit 5 CICS Instanzen, die in 5 virtuellen Adressenräumen (Regions) laufen. In einem symmetrischen Multiprozessor können die 5 Instanzen auf 5 CPUs verteilt werden.

TOR	Terminal Owning Region
AOR	Application owning Region
FOR	File owning Region (Resource Region)

Die AORs können für bestimmte Anwendungen, spezielle Zugriffsarten, Datenbankverbindungen usw. optimiert sein. AORs können auf unterschiedliche Regions eines Sysplex verteilt sein.

Es existieren Mainframe Installationen mit über 100 CICS Instanzen in ebenso vielen CICS Regions.

## Warum mehrfache AORs ?

Prozesse unter z/OS sind durch einen Task Control Block (TCB) gekennzeichnet. Der TCB ist ein Speicherbereich, der z.B. den Inhalt der Mehrzweck-, Gleitkomma-, Controlregister und das Programm Status Wort (PSW) aufnimmt, wenn der Prozess von dem laufenden in den wartenden Zustand versetzt wird.

CICS einschließlich Nucleus und aller Anwendungsprogramme läuft in einem einzigen z/OS Address Space und wird in den Augen des z/OS Kernels durch einen einzigen TCB (Task Control Block) repräsentiert.

Innerhalb des CICS Address Spaces laufen gleichzeitig hunderte oder tausende von Transaktionen, die durch die Scheduler Komponente des CICS Nucleus gescheduled werden. Jede Transaktion wird durch einen „quasi-reentrant“ TCB (QR TCB) repräsentiert. Im Gegensatz zu normalen TCBs werden die QR TCBs nicht vom Betriebssystem Kernel, sondern vom CICS Nucleus gescheduled und verwaltet. Dies geschieht unsichtbar für den z/OS Kernel.

Da CICS nur über einen einzigen TCB verfügt, kann der z/OS Kernel es auch nur für eine einzige CPU schedulen. Nur ein einziger QR TCB ist in jedem Augenblick active (running). Multiple AORs bedeuten multiple CICS Instanzen und erlauben die Transaktionsverarbeitung auf mehreren CPUs. Der z/OS Work Load Manager (WLM, diskutiert in einem späteren Module) ordnet neue Transaktionen den einzelnen AORs zu.



## **AOR's und VSAM**

**AOR's laufen in getrennten z/OS Regions. Die dort laufenden Programme können unabhängig voneinander auf die gleichen DB2 Daten zugreifen. Das DB2 Datenbanksystem verfügt über entsprechende Lock Management Einrichtungen.**

**Die von CICS verarbeiteten VSAM Datenbestände haben eine ähnliche Größe wie die von CICS verarbeiteten DB2 Datenbestände. Häufig greift eine CICS Anwendung parallel sowohl auf VSAM wie auf DB2 Daten zu. VSAM fehlen jedoch die Lock Management Einrichtungen, die für einen parallelen Zugriff durch mehrere AOR's erforderlich sind.**

**Abhilfe schafft eine als „VSAM Record Level Sharing“ (RLS) bezeichnete Funktion, die von DFSMS (siehe Thema „z/OS Betriebssystem“ Teil 3 dieses Moodle Vorlesungsscriptes) als getrenntes Subsystem (SMSVSAM) zur Verfügung gestellt wird. CICS RLS ermöglicht es, dass zahlreiche CICS Anwendungen in unterschiedlichen AOR's parallel auf die gleichen VSAM Data Sets mit voller Update Fähigkeit zugreifen können.**

# Background – Information

## Quasireentrant Programs

**Multithreading ermöglicht es, dass eine einzelne Kopie eines Anwendungsprogramms von mehreren Transaktionen gleichzeitig ausgeführt wird. Zum Beispiel kann eine Transaktion anfangen, ein Anwendungsprogramm auszuführen. Wenn ein EXEC CICS Befehl erreicht wird, wird die Transaktion in den Wartezustand versetzt. Der CICS Dispatcher scheduled eine andere Transaktion, welche die gleiche Kopie des Anwendungsprogramms ausführt.**

**Multithreading verlangt, dass alle CICS Anwendungsprogramme quasi-reentrant geschrieben sind. Dies bedeutet, sie müssen seriell wiederverwendbar zwischen Einstiegs- und Ausstiegspunkten sein. CICS Anwendungsprogramme mit der EXEC CICS-Schnittstelle gehorchen dieser Regel automatisch. Für COBOL, C/C++- und PL/1 Programmen wird Reentrancy erreicht, indem jede Transaktion einen eigenen Speicherbereich für die Speicherung von Variablen erhält. Eine gewisse Ähnlichkeit mit Java Objekten ist unverkennbar..**

**CICS Anwenderprogramme (Transaktionen) laufen unter einer CICS-Managed Task Control Block (TCB). Wenn die Programme als Quasi-reentrant definiert sind (das CONCURRENCY Attribut der Programm Ressourcen-Definition gesetzt ist), läuft CICS immer unter dem CICS „quasi-reentrant (QR) TCB“. Die Voraussetzungen für ein quasi-reentrant Programm in einem Multithreading Kontext sind weniger streng, als wenn das Programm gleichzeitig auf mehreren „OpenTCBs“ und mehreren CPUs ausgeführt werden soll.**

**„Quasi-Reentrant“ bedeutet, dass das Anwendungsprogramm vor und nach jedem EXEC CICS-Befehl in einem konsistenten Zustand sein muss. Derartige quasi Reentrance garantiert, dass jeder Aufruf eines Anwendungsprogramms unbeeinflusst von früheren Durchläufen, oder dem Multithreading durch durch mehrere CICS QR-TCB Tasks ist.**

**CICS Transaktionen können alternativ unter dem “Open TCB” an Stelle des QR TCBs laufen. Open TCBs werden vom Betriebssystem Kernel gescheduled und können damit mehreren CPUs zugeordnet werden. In anderen Worten, die Multiprocessor Eigenschaften eines Mainframes werden genutzt. Hiermit kann der Gesamtdurchsatz verbessert werden; allerdings erfordert das Scheduling durch den Betriebssystem Kernel mehr Aufwand (CPU Zyklen) als das Scheduling durch den CICS Nucleus. Vor allem Java Programme machen vom Open TCB Gebrauch.**

# CICSplex

**Was ist ein CICSplex?**

**Wenn mehrere CICS Instanzen in getrennten Regions installiert werden, nennt man die resultierende Architektur einen CICSplex. Als CICSplex wird eine Gruppe von logisch verbundene CICS Regions betrachtet. Ein typischer CICSplex ist eine miteinander verbundene Gruppe von TORs, AORs, FORs usw. Auf einem Sysplex-Verbund von Mainframe Rechnern wird normalerweise ein (oder mehrere) CICSplex eingesetzt.**

**Jede CICS Instanz verwaltet eine Region (Address Space). Jede CICS Region kann für eine bestimmte Aufgabe optimiert sein, und dennoch leicht mit anderen CICS Regions kommunizieren.**

**Ein System von miteinander verbundenen CICS Regions kann z.B. enthalten:**

- **Test Regions für das Austesten neuer Anwendungen**
- **Function-spezifische Regions für die Steuerung von Terminals, Benutzer Interfaces oder System Services**
- **Nutzung spezifischer Regions für das Management bestimmter Anwendungen, wie z.B. Datenbanken**

**Eine spezielle CICS Komponente, der CICSplex System Manager (CICSplex SM )ist für die Steuerung einer Vielzahl von CICS Regions zuständig. In zunehmendem Maß finden wir Unternehmen mit Dutzenden oder Hunderten von CICS Regions.**

**Ein Beispiel ist die Installation einer Schweizer Großbank.**

# **Union Bank of Switzerland Global CICS Environment's 3Q 2006**

**Beispiel einer CICS Großinstallation. Die Union Bank of Switzerland unterhält in zwei Standorten in der Nähe von Zürich zwei Rechenzentren, die als Geographically Dispersed Parallel Sysplex (GDPS) zusammengeschaltet sind. Ein weiterer GDPS befindet sich im Staate New Jersey in den USA.**

**Die CICS Konfiguration des Schweizer GDPS besteht aus:**

- **11 CICS Plexes**
- **943 CICS Regions**

**Die CICS Konfiguration des USA GDPS besteht aus:**

- **4 CICS Plexes**
- **194 CICS Regions**

# **Intersystem Communication (ISC)**

**CICS Multi Region Operation findet innerhalb eines einzigen z/OS Systems statt, oder innerhalb eines Sysplex statt, der darauf abzielt, ein „Single System Image“ anzubieten. Eine CICS Kommunikation zwischen zwei (ansonsten unkoordinierten) Rechnern wird als CICS Intersystem Communication (CICS ISC) bezeichnet.**

**Es existieren viele, nicht CICS spezifische Arten, von Intersystem Communication (ISC).**

**Einfache Beispiele für ISC-Standards sind Sockets, Named Pipes und APPC. In verteilten Systemen ist ISC häufig schwierig zu entwickeln, debug, portieren, besonders für unterschiedliche Plattformen, z.B. auf Grund von Abhängigkeiten von der Rechnerhardware (Byte Order, Adressierung).**

**Der RPC (Remote Procedure Call) ist eine klassische Software für die Implementierung von Client/Server Konfigurationen. Ein RPC liefert automatisch Funktionen, die beim Programmieren mit Sockets vom Programmierer erstellt werden müssen, z.B. Fehlerbehandlung, Adressierung und Datendarstellung. RPC erleichtert gegenüber Sockets die Entwicklung und Wartung von verteilten Anwendungen.**

**Es existieren eine ganze Reihe von inkompatiblen RPC Standards, z.B. Sun RPC, DCE, SOAP (Web Services), sowie objekt-orientierte Versionen wie CORBA und RMI. Der RPC ist ein synchrones Protokoll, d.h. der Klient wartet (blockiert) auf die Antwort des Servers. Im Gegensatz dazu ist MQSeries ein weit verbreitetes asynchrones RPC Protokoll.**

**Normalerweise wird ISC durch Bibliotheken eines Betriebssystems implementiert.**

# **CICS Intersystem Communication (ISC)**

**CICS Distributed Program Link (DPL) ist eine CICS-spezifische, besonders einfach zu benutzende, Art des Remote Procedure Call. Voraussetzung ist, dass sowohl der Client als auch der Server unter CICS laufen. DPL benutzt TCP/IP, SNA oder NetBios in der Netzwerk Schicht 4.**

**Ein Programm eines CICS-Systems kann mit Hilfe des DPL Kommandos EXEC CICS LINK eine Verbindung mit einem Programm eines anderen CICS-Systems aufnehmen.**

**Es existieren viele Vereinfachungen gegenüber dem normalen RPC, da einheitliche Datendarstellung, Namenskonventionen, Fehlerbehandlung, u.s.w. Z.B. existieren keine Datenrepräsentations-Probleme, da alle CICS Implementierungen einen einheitlichen EBCDIC Standard einsetzen.**

**CICS TP Monitore existieren für viele unterschiedliche Betriebssystem-Plattformen, neben z/OS z.B. für Windows, i5/OS, oder Unix. Eine DPL Kommunikation zwischen CICS TP Monitoren auf unterschiedlichen Plattformen ist sehr einfach zu implementieren.**

**Neben DPL existieren weitere CICS ISC Protokolle mit spezifischen Eigenschaften, z.B. Distributed Transaction Programming (DTP).**

# Distributed CICS

Der CICS Transaction Server wurde ursprünglich für Mainframe Rechner entwickelt. Heute existiert zusätzlich eine relativ weit verbreitete Version, die als „Distributed CICS“ bezeichnet wird und die auf anderen Plattformen läuft, z.B.:

- Solaris,
- Microsoft Windows,
- IBM AIX,
- Hewlett-Packard UNIX (HP-UX) und
- Linux

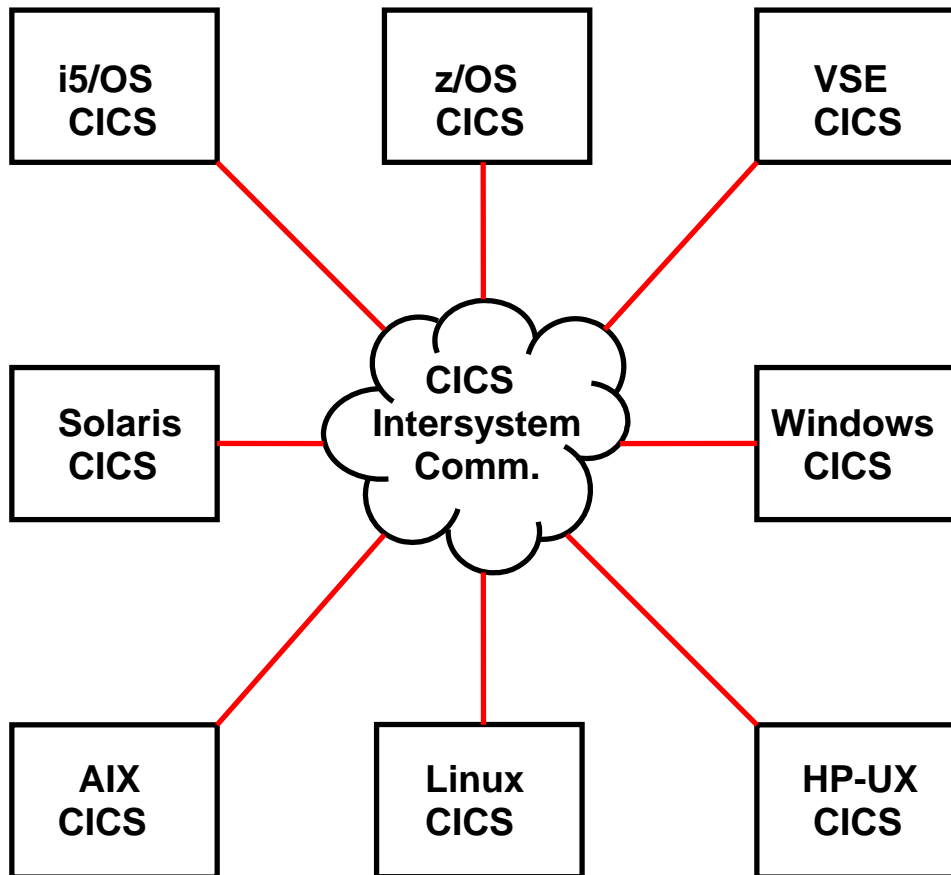
Distributed CICS verfügt über die gleiche Application Programming Interface (API) und die gleiche System Programming Interface (SPI) wie der Mainframe CICS Transaction Server.

Es unterstützt eine Vielzahl von Datenbanken wie DB2, Oracle, IBM Informix, Microsoft SQL Server und Sybase. Datenintegrität wird durch die Unterstützung der XA-Schnittstelle bereitgestellt. XA ist die Industrie-Standard-Schnittstelle für das Commitment und die Recovery von transaktionalen Daten, einschließlich des Two-Phase Commit-Prozesses. Weiterhin verfügt Distributed CICS über einen „Structured File Server“ (SFS), eine Record-orientierte VSAM Emulation, welche indexierte, relative und sequentielle File Zugriffe ermöglicht. Unterstützt wird das Äquivalent eines Open TCB, nicht aber eines QR TCB.

Anwendungen können in C, C++, PL/I, COBOL, und Java geschrieben werden und sind problemlos auf die Mainframe CICS Version portierbar.

Spezifisch ist hiermit eine sehr einfache und problemlose Intersystem Communication möglich.

<http://www.redbooks.ibm.com/redbooks/pdfs/sg247185.pdf>



CICS ISC verwendet das *Distributed Program Link* Protokoll (DPL) und den EXEC CICS LINK Befehl. DPL ist wesentlich einfacher zu programmieren als RPC, da einheitliche Konventionen existieren, (z.B. keine Datenrepräsentations-Probleme). DPL benutzt TCP/IP, SNA oder NetBios in Schicht 4 .

Nicht-CICS Anwendungen benutzen die ECI (External Call Interface) oder EPI (External Program Interface) um über DPL auf ein CICS Subsystem zuzugreifen. Ein Beispiel werden wir später in dem Modul Java Connection Architecture vorstellen.

## Intersystem Communication zwischen CICS TP Monitoren auf unterschiedlichen Plattformen



Als Beispiel können CICS Programme auf unterschiedlichen Plattformen (z.B. z/OS, Linux, Windows) Daten über die CICS COMMUNICATION AREA (COMMAREA) austauschen

In COBOL ist dies ein Teil der DATA DIVISION / LINKAGE SECTION, z.B.

```
O1 DFHCOMMAREA.  
  05 PROCESS-SW          PIC X.  
    88 INITIAL-ENTRY    VALUE '0'.  
    88 VERIFICATION     VALUE '1'.  
  05 ACCOUNT-NUMBER     PIC X(10).  
    ●  
    ●  
    ●  
  
EXEC LINK PROGRAM(ACCTPGM)  
  COMMAREA(DFHCOMMAREA)  
  LENGTH(11)  
END-EXEC.
```