CICS® Transaction Gateway v5.1
and
CICS Universal Clients v5.1

Technical Introduction

Phil Wakelin

IBM Hursley

e business software

Agenda

- CICS TG vs CICS UC
- Infrastructure
- Application Programming
- Deployment scenarios
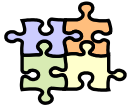- V5 and beyond

What is the CICS TG

Q. What does it do?

A. Provides ability to link to CICS programs or invoke CICS transactions from a Java™ environment

What does it provide:

1. Plumbing - Connectivity into CICS

2. Interfaces - Java and non-Java APIs

3. Integration - WebSphere® and CICS and others.
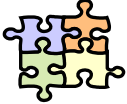
What is the CICS UC?

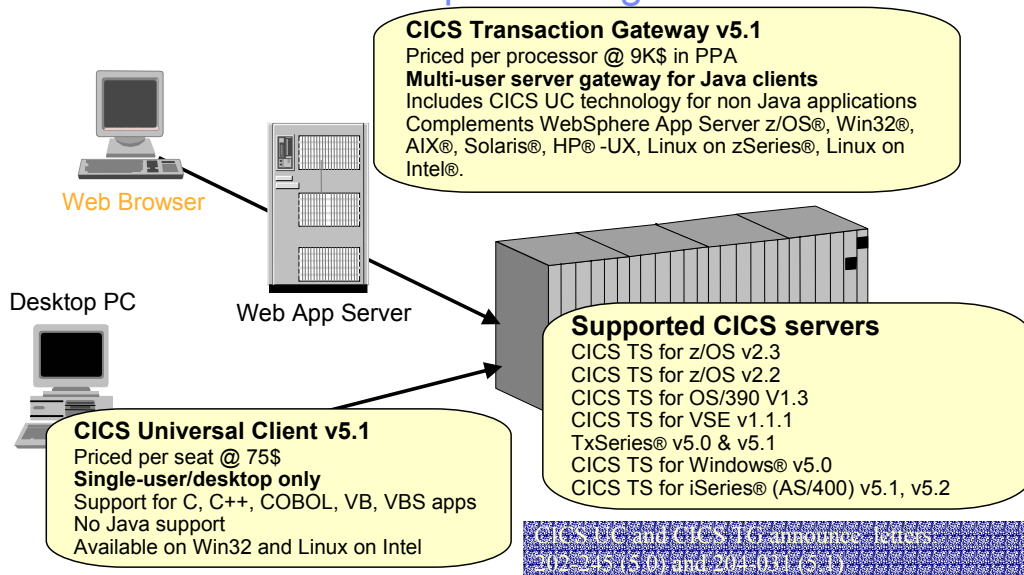- Single user/desktop product

- Based on same communications technology as CICS TG

- Provides C/COBOL/COM APIs
  - Does not provide Java integration
  - Does not provide remote connectivity

CICS TG and CICS UC positioning

Web Browser

Desktop PC

Web App Server

CICS Transaction Gateway v5.1
Priced per processor @ 9K$ in PPA
Multi-user server gateway for Java clients
Includes CICS UC technology for non Java applications
Complements WebSphere App Server z/OS®, Win32®, AIX®, Solaris®, HP® -UX, Linux on zSeries®, Linux on Intel®.

Supported CICS servers
CICS TS for z/OS v2.3
CICS TS for z/OS v2.2
CICS TS for OS/390 V1.3
CICS TS for VSE v1.1.1
TxSeries® v5.0 & v5.1
CICS TS for Windows® v5.0
CICS TS for iSeries® (AS/400) v5.1, v5.2

CICS Universal Client v5.1
Priced per seat @ 75$
Single-user/desktop only
Support for C, C++, COBOL, VB, VBS apps
No Java support
Available on Win32 and Linux on Intel

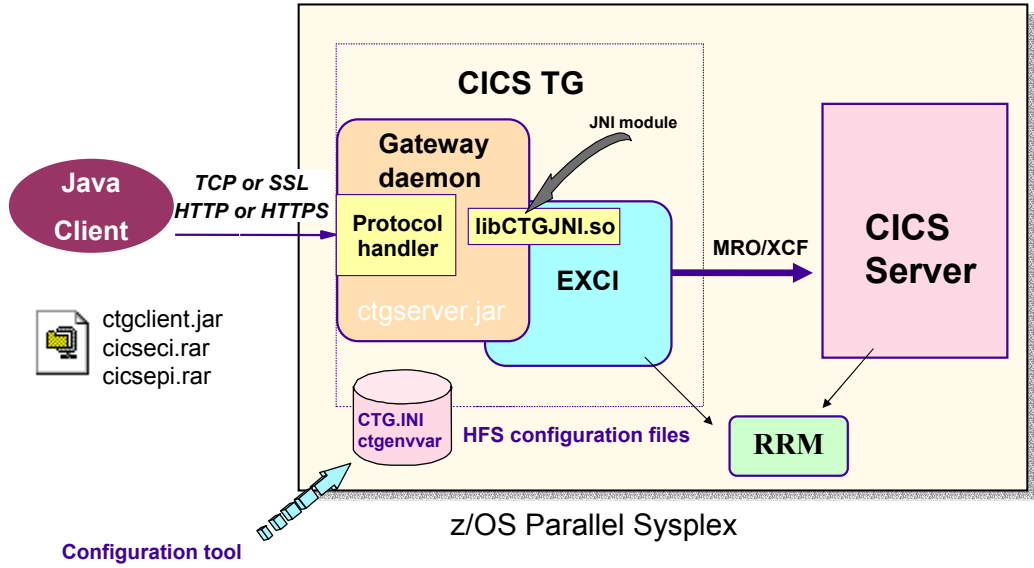CICS UC and CICS TG announce letters
202-245(5.0) and 204-031(5.1)

Which CICS TG platform is for me?

- z/OS
  - Scalability and support for IP workload balancing (i.e. Sysplex Distributor)
  - Transactional EXCI with WAS z/OS
  - Familiar z/OS system environment
  - IP connectivity from Java client
  - Provides ECI only (via EXCI)
  - Timeout support only within EXCI
  - Security integration: SSL support and flexible CICS security
  - Close integration with JCA connection pooling

- Distributed (UNIX®, Linux on zSeries or Intel, Win32)
  - IP connectivity to CICS TS v2.2 for ECI
  - IP connectivity via TCP62 to CICS TS
  - Provides ECI, EPI and ESI interfaces via APPC/TCP62
  - Full timeout support on ECI requests for all protocols
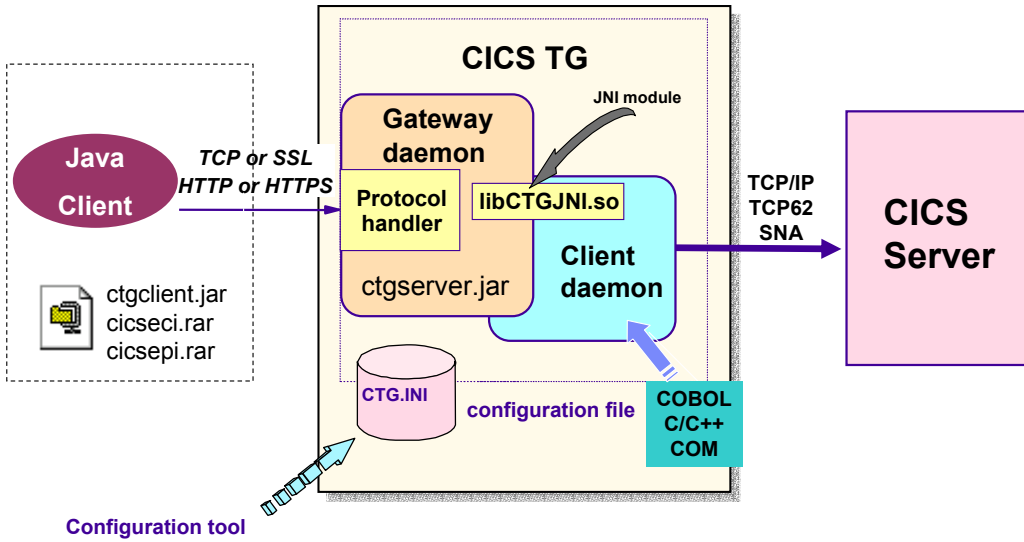 - Security: Attachsec=verify only

For further details on the differences see ibm.com/support FAQ article 1111693 :

http://www-1.ibm.com/support/docview.wss?rs=166&q=cics&uid=swg21111693&loc=en_US&cs=utf-8&lang=en

CICS TG on z/OS - components

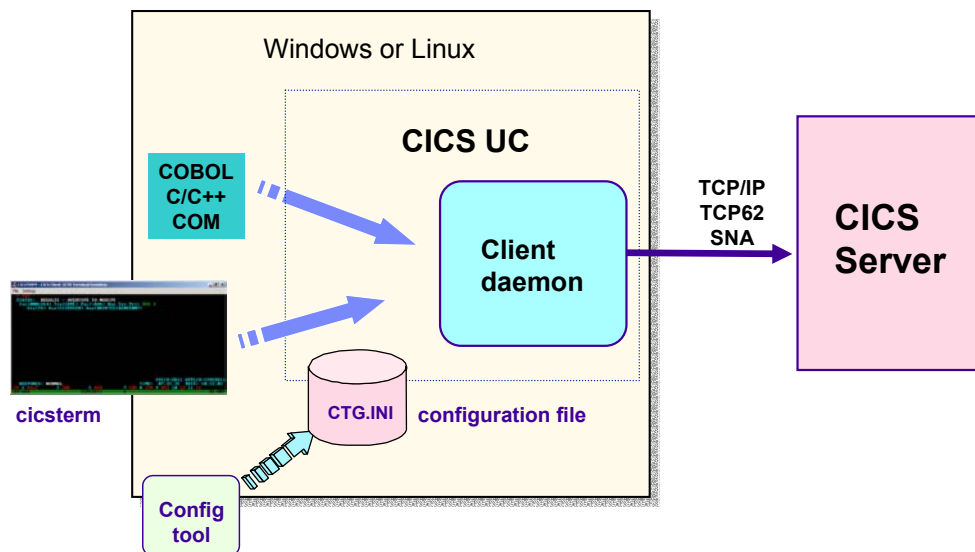# CICS TG on distributed platform - components

Java Client

TCP or SSL HTTP or HTTPS

ctgclient.jar
cicseci.rar
cicsepi.rar

CICS TG

JNI module

Gateway daemon

Protocol handler

libCTGJNI.so

ctgserver.jar

Client daemon

TCP/IP TCP62 SNA

CICS Server

CTG.INI configuration file

COBOL C/C++ COM

Configuration tool

CICS Universal Client - components

Windows or Linux

CICS UC

COBOL
C/C++
COM

Client
daemon

cicsterm

CTG.INI   configuration file

Config
tool

TCP/IP
TCP62
SNA

CICS
Server

Network protocols

Java client to Gateway daemon

Client daemon to CICS

CICS TG

Gateway daemon

Java client

TCP/SSL
HTTP/HTTPS

Client daemon or EXCI

APPC
TCP62
ECI over TCP/IP
EXCI (z/OS only)
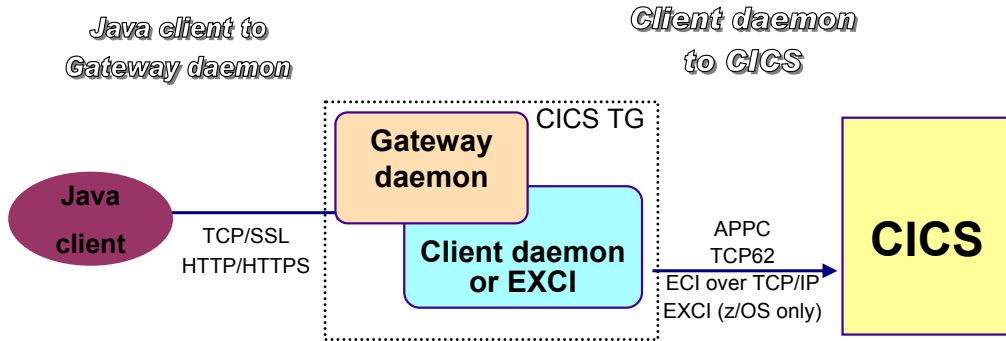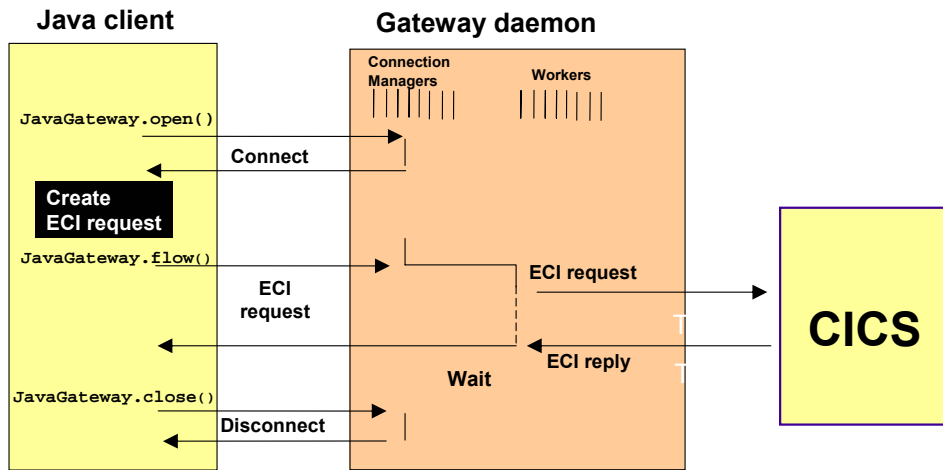
CICS

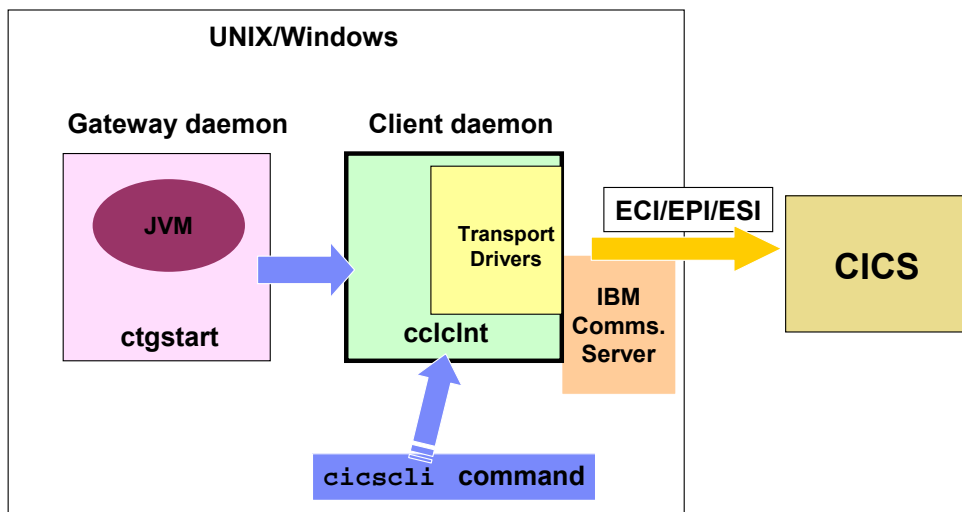TCP and SSL protocols can use persistent connections

TCP62 is APPC encapsulation in IP
EXCI, ECI/IP and APPC give best performance
ECI/IP available with CICS TS v2.2 and v2.3

Gateway daemon – Threading example

Java client

Gateway daemon

Connection Managers

Workers

`JavaGateway.open()`

Connect

Create ECI request

`JavaGateway.flow()`

ECI request

ECI request

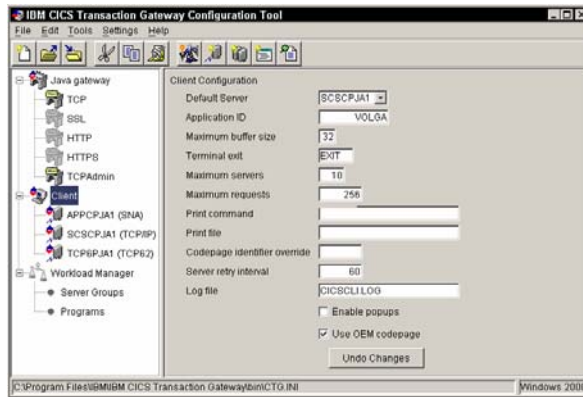**CICS**

ECI reply

Wait

`JavaGateway.close()`

Disconnect

Local protocol allows the Java client to call the JNI library within it's own JVM. In this case the Gateway daemon is not used.

Client daemon

UNIX/Windows

Gateway daemon

Client daemon

JVM

ctgstart

Transport Drivers

cclclnt

IBM Comms. Server
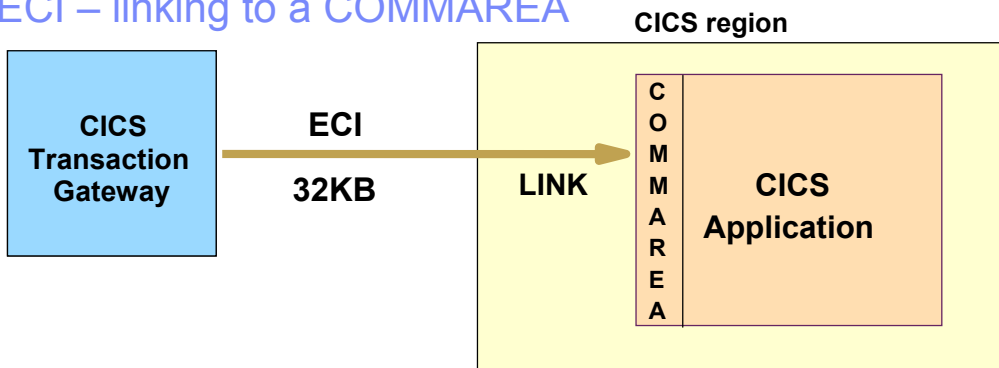
ECI/EPI/ESI

CICS

`cicscli command`

Configuration tool

- Creates CTG.INI file and ctgenvvar for z/OS
- Used by CICS UC and CICS TG
- Can be used with remote platform option to create a config file for z/OS: ctgcfg –PLAT ZOS
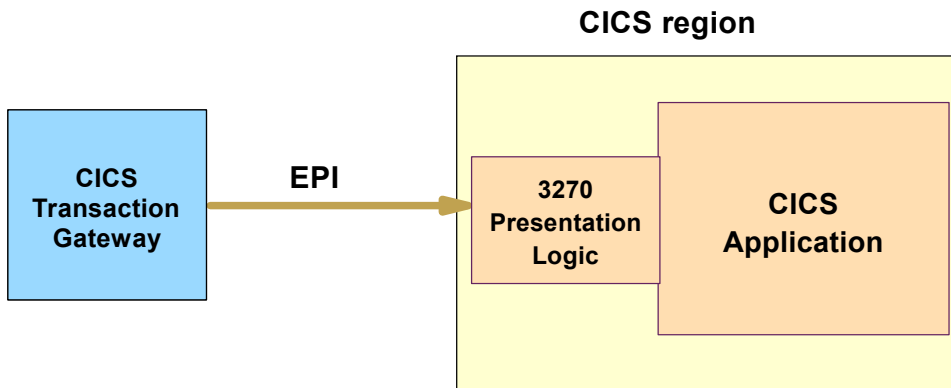
Application Programming

ECI – linking to a COMMAREA

CICS region

- Java API to LINK to a CICS program
  - Available on all platforms
  - With z/OS based on EXCI
  - Can use extended units of work (i.e. without SYNCONRETURN)
  - Interfaces:
    - ➤ Non Java: C/C++, COBOL, COM
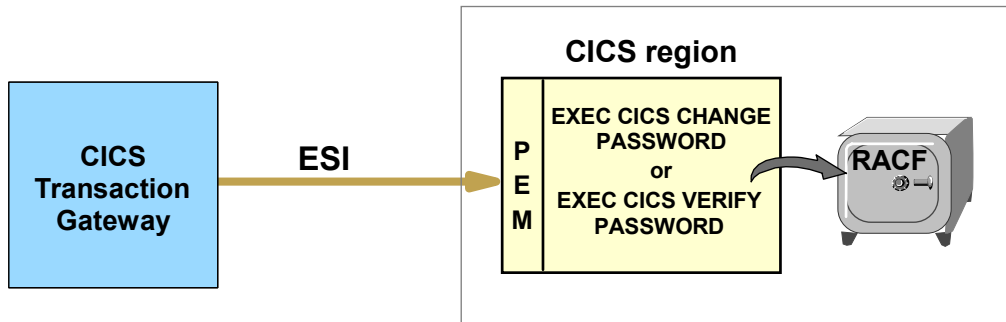   - ➤ Java:ECIRequest base class, JCA interfaces

EPI - accessing 3270 transactions

CICS region

- Java API to access a 3270 transaction
  - Interfaces:
    - ➢Non Java: C/C++, COBOL, COM
    - ➢Java: Base EPIRequest class, EPI support classes, JCA

ESI

CICS region

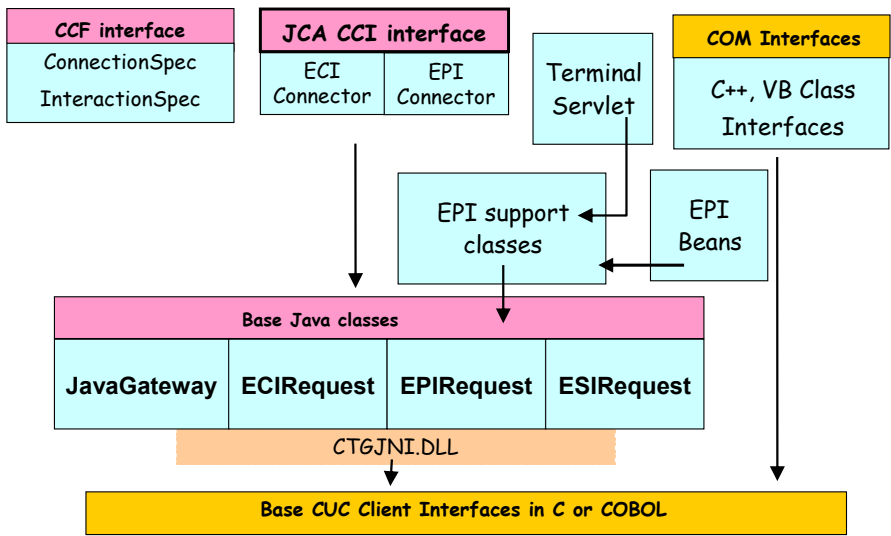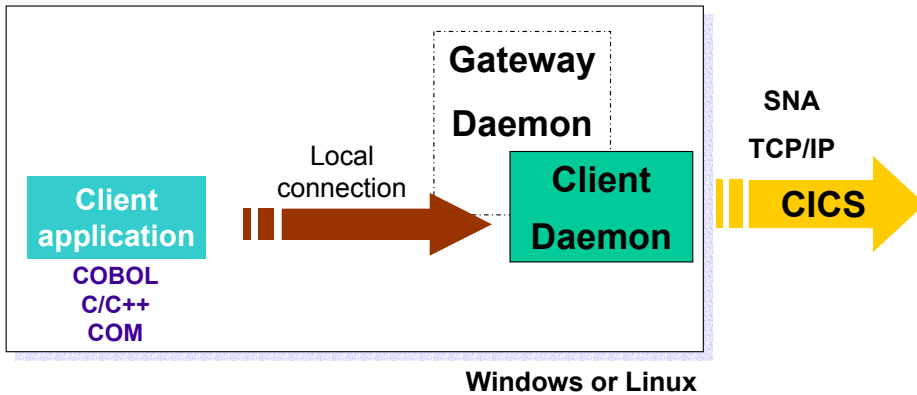| CICS Transaction Gateway | **ESI** → | P E M | EXEC CICS CHANGE PASSWORD or EXEC CICS VERIFY PASSWORD | → RACF |

- Java interface to PEM support in CICS
- Allows userid/password to be verified or changed
- To be used in conjunction with flowed ECI or EPI userids
- Interfaces:
  - ➢ Non Java: C/C++, COBOL, COM
  - ➢ Java: Base ESIRequest class

CICS TG Interfaces

CCF interface
ConnectionSpec
InteractionSpec

JCA CCI interface	
ECI Connector	EPI Connector

Terminal Servlet

COM Interfaces
C++, VB Class Interfaces

EPI support classes

EPI Beans

Base Java classes			
JavaGateway	ECIRequest	EPIRequest	ESIRequest

CTGJNI.DLL

Base CUC Client Interfaces in C or COBOL

Deployment scenarios

CICS TG or CICS UC – non Java deployment

Gateway

Daemon

SNA

TCP/IP

Local
connection

**Client
application**

**COBOL
C/C++
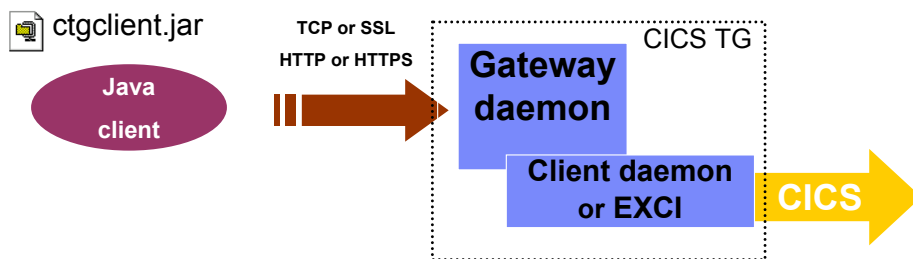COM**

**Client
Daemon**

CICS

Windows or Linux

- **Windows: COBOL, C/C++, COM (Visual Basic)**
- **UNIX or Linux: C/C++:**

- **Note on z/OS: COBOL, C, PL/I, Assembler supported via CICS EXCI**

CICS TG - Java deployment scenarios

- Q. What do I need?

- A. Supported JVM & the Java class libraries
  Also require J2EE resource adapter if using a JCA environment

- Examples are:
  Standalone Java application (fat client)
  Applet or servlet
  JCA: servlet or EJB component

Standalone Java client

ctgclient.jar

Java client

TCP or SSL
HTTP or HTTPS

CICS TG

Gateway daemon

Client daemon or EXCI

CICS

Java client can use
- Base classes (code by hand)
- CCI (J2EE model)
- CGF (from EAB in VAJ)

J2EE connector architecture (JCA)

- **Standards based connectivity**
  - JCA 1.0 is part of J2EE 1.3 specification
  - WebSphere App Server v5.0 and v5.1 are J2EE 1.3 compliant

- **Integration with IBM tooling**
  - Packaged with WSAD-IE
  - CCI provides simplified application development
  - CCI is a common API for all resource adapters

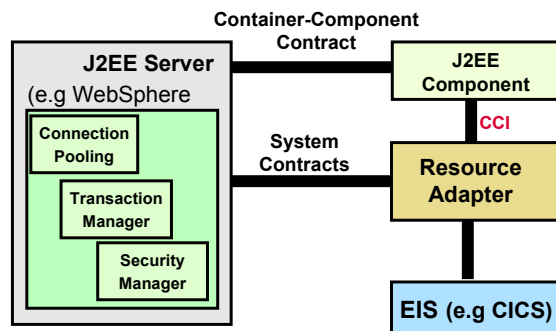- **JCA qualities of service**
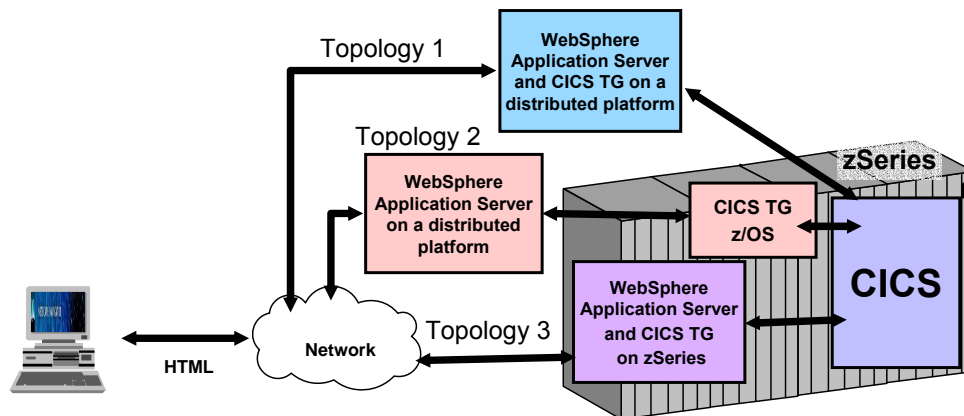 - Transactions
  - Connection pooling
  - Security

- **Resource adapter**
  - CICS TG provides an ECI and an EPI resource adapter
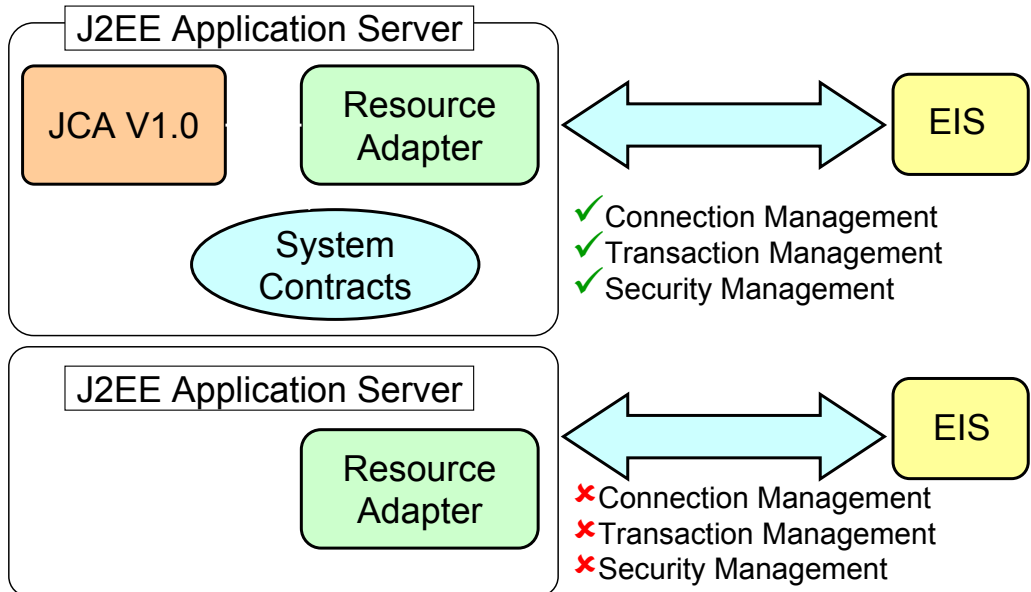 - Provides for simplified deployment to J2EE server

Container-Component Contract

J2EE Server (e.g WebSphere)
- Connection Pooling
- Transaction Manager
- Security Manager

J2EE Component

CCI

Resource Adapter

System Contracts

EIS (e.g CICS)

The WebSphere connector for CICS

Topology 1

WebSphere Application Server and CICS TG on a distributed platform

Topology 2

WebSphere Application Server on a distributed platform

zSeries

CICS TG z/OS

CICS

WebSphere Application Server and CICS TG on zSeries

Topology 3

Network

HTML

J2EE managed environment

J2EE Application Server

| JCA V1.0 | Resource Adapter |

System Contracts

✓ Connection Management
✓ Transaction Management
✓ Security Management

EIS

J2EE Application Server

Resource Adapter

✗ Connection Management
✗ Transaction Management
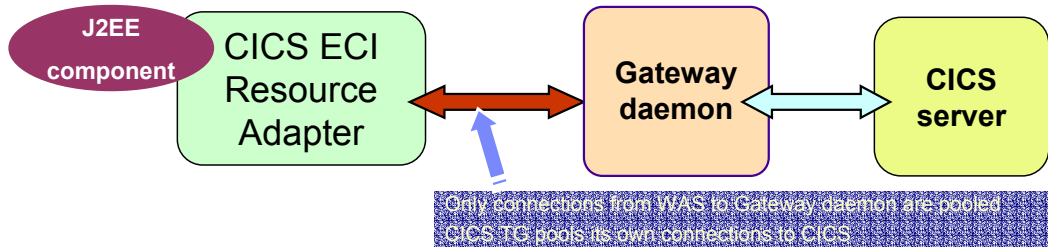✗ Security Management

EIS

Connection pooling

An application server uses the Pool Manager to implement a connection pooling mechanism in its own implementation specific way

- Allows network connections to the CICS TG to be managed by WAS and pooled among J2EE components
  - ✓ Reduces CPU load in Application Servers
  - ✓ Reduces network I/O for App. Server and EIS

J2EE component — CICS ECI Resource Adapter ⟷ **Gateway daemon** ⟷ **CICS server**

Only connections from WAS to Gateway daemon are pooled
CICS TG pools its own connections to CICS

Tooling

- J2EE Connector Architecture recommends using a development tool to generate CCI code

- Tools can be particularly useful for Record generation and data marshalling

- IBM tools supporting JCA:
  - WebSphere Studio Application Developer Integration Edition V5
    - Strategic tool, using Enterprise Services
    - Based on WSIF framework and WSDL

  - WebSphere Studio Enterprise Developer V5
    - Superset of WSAD-IE tooling, including additional COBOL, XML and enhancements

  - VisualAge for Java Enterprise Edition V4
    - Provides CCF and Java Record Framework
    - Uses a JCA beta specification
    - Can develop CCI based Commands/Records
    - EOS in 12/03
    - Limited migration to WSAD-IE/WSIF

V5 and beyond

CICS TG v5 - highlights

- GA July 2002
- Support for the J2EE Connector Architecture
- Support for WebSphere V5
- 128 bit SSL with JSSE
- Dynamic control of tracing in Gateway daemon
- Performance enhancements for data transfers
- Extended EXCI/JNI logging
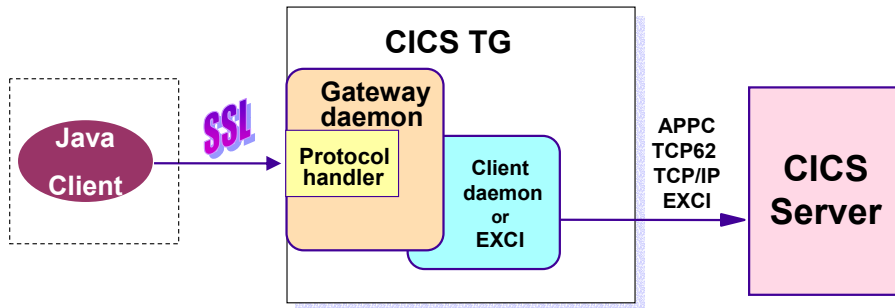- Service concurrency

    V3 is EOS

    V4 is EOS in Nov/03

CICS TG v5.01 - highlights

- GA Aug 2003

- Required for WAS z/OS v5 support (including z/OS.e)

- Also provides
  - **Remote Gateway support on WAS z/OS**
  - **Full accessibility (samples and graphical interfaces)**
  - **AIX v5.2 and z/OS v1.4 support**
  - **Updated Java 1.3.1 Service Refresh**
  - **JCA ECIInteractionSpec methods**: setTPNName() & setTranName()
  - Control of SSL cipher suite (enforcement of 128 bit ssl)
  - High performance memory tracing for Client daemon
  - New EPI exit CICS_EPIStartTranExtendedExit, (with Term Index)
  - EPI/terminal recovery for CICS server outages
  - Important APARs:
    - Improved performance of EPI flows (null stripping)
    - 5x retry of retryable failed EXCI allocates
    - Non swappable CTG address space
    - Improved performance of compression exits

CICS TG V5.1

- **GA March 2004**

- Provides:
  - **Java 1.4 support**
  - **WAS 5.1 support**
  - **Windows 2003 support**
  - **Solaris 9 support**
  - **32 bit Linux on Intel support (also on CICS UC)**
  - **RHEL and United Linux/SLES V8.0**
  - **Updated Java 1.3.1 Service Refresh**
  - **APARs**

128 bit SSL JSSE

CICS TG

Gateway daemon

Protocol handler

Java Client

SSL

Client daemon or EXCI
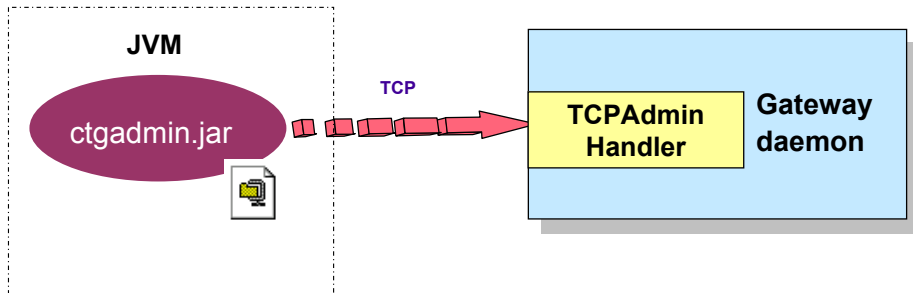
APPC
TCP62
TCP/IP
EXCI

CICS Server

- SSL protocol handler can now use JSSE, SSLight or SystemSSL libraries
- Java Secure Sockets Extension (JSSE) is strategic
- JSSE: provides 128 bit+ encryption in client and server, and zSeries crypto support
- System SSL is z/OS specific and provides 128 bit encryption and zSeries crypto support
- SSLight provides 56bit encryption in client and server
- CICS TG v5.01 provides CTG.INI option to enforce 128bit cipher suite for JSSE
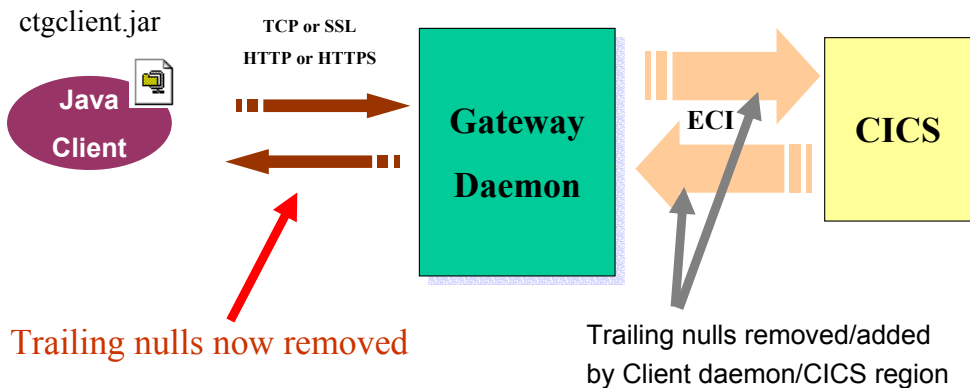 – ciphersuites=128bitonly;

Dynamic tracing

- ## TCPAdmin protocol handler

    – Allows administrator to remotely activate, deactivate and view Gateway daemon Java or JNI trace

    – Requires new TCPAdmin protocol handler in CTG.INI

    – Can specify limited access based on IP addresses

JVM

ctgadmin.jar

TCP

TCPAdmin Handler **Gateway daemon**

Improved data handling

ctgclient.jar

TCP or SSL
HTTP or HTTPS

**Java
Client**

**Gateway
Daemon**

ECI

CICS

Trailing nulls now removed

Trailing nulls removed/added
by Client daemon/CICS region

- Automatic truncation of trailing nulls in ECI flows returned from Gateway daemon to Java client
- Equivalent to automatic invocation of setCommareaInboundLength() method
- Improved internal data transfers from Gateway daemon to Client daemon

EXCI logging

- Gateway daemon on z/OS
- All EXCI errors now logged: `$HOME/ibm/ctg/ctgjnilog.xxxxxxxxxx`
- Unique file used for every new Gateway daemon

  Example message:

```
CICS Transaction Gateway JNI Log file for z/OS Version 5.0 Service Level 00, Build Level c500-20020715
10/07/2002 16:43:32.093 :CCL6806I CcicsInit: Register with RRS. Return code=768.
10/07/2002 16:43:32.129 : CCL6876E CcicsECI: EXCI error.  EXCI Reason = 201, Subreason field-1 = 0, subreason
field-2 = 0
```

Future product considerations

- **JCA connector API is strategic**
  - JCA interfaces provide the Qualities of Service
  - Base Java classes will be stabilized

- **Enhanced systems management and monitoring**
  - Especially on z/OS

- **Interoperation with the latest IBM technologies and tooling**
  - Including WebSphere App Server, Studio and J2EE directions

- **Support for new operating systems**
  - 64 bit support as it develops
  - Linux kernels and NPTL threading

- **Relief from architectural limits**
  - EXCI 100 pipes, 32K COMMAREA, XAResource/2PC, improved TCP/IP communication

- **Improved tracing, logging and serviceability**
  - Performance and usability

- **Consolidation**
  - Deprecation of CCF support
  - JSSE as the SSL technology
  - Removal of out-of-process COM and NPI support
  - Removal of HTTP and HTTPS support, with view to removal of applet support

More information

- Redbooks: www.redbooks.ibm.com
  SG24-6401
    Java Connectors for CICS
  SG24-6133-01
    CICS TG v5, The WebSphere Connector for CICS
  REDP0206
    From code to deployment: Connecting to CICS from WebSphere for z/OS
  SG24-6200
    Exploring WebSphere Studio Application Developer Integration Edition 5.0

- Whitepapers
    ibmlink: http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi
      G224-7218 - Integrating WebSphere Application Server and CICS using the JCA

    Techdocs: http://www-1.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs
      WP100395 - Deployment to WAS z/OS
      WP100395 - Performance Best Practices

    WebSphere Studio zone: http://www-
      106.ibm.com/developerworks/websphere/zones/studio/transition.html#wsadie
      CCF to J2C migration