



IBM Software Group

WAVV Conference, Chattanooga, TN
April/May 2004

CICS Transaction Gateway – Introduction and V5.1 overview

Robin K Gemmell,
IBM Hursley
gemmell@uk.ibm.com

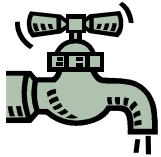
Agenda

1. Introduction
2. Application Development
3. Components
4. CICS Universal Client
5. Network connections
6. Deployment scenarios
7. CICS TG v5 and beyond

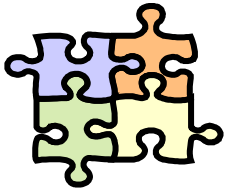
Part One

Introduction

What is the CICS TG?



1. Plumbing - Connectivity into CICS



2. Interfaces - Java and non-Java APIs



3. Integration - WebSphere and CICS and others.

What does it do?

- **What does it do?**

- Provides ability to link to CICS COMMAREA programs or invoke CICS 3270 transactions from Java or *native* (non Java) applications

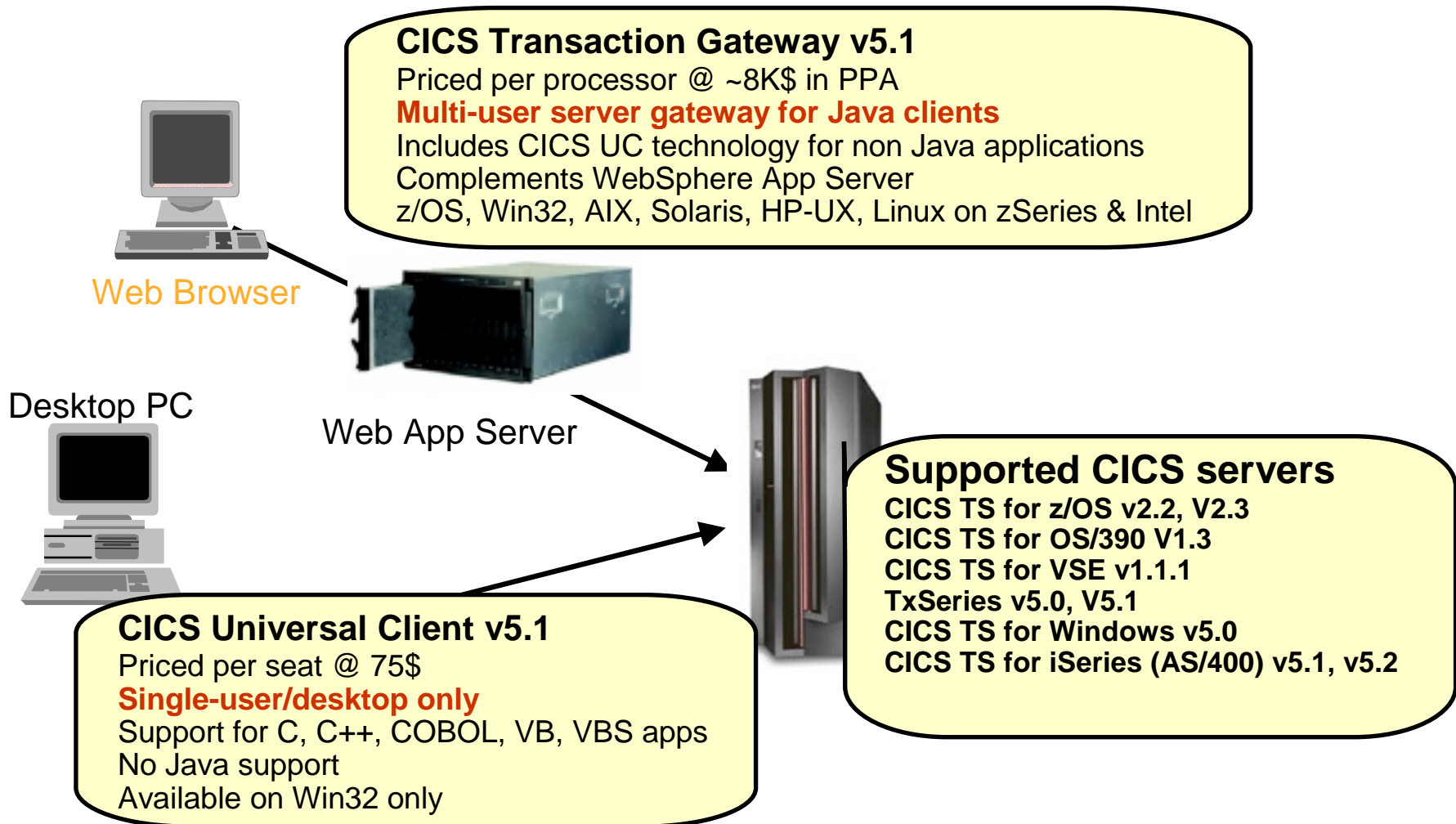
- **What does it offer?**

- API: Java class libraries
- Synchronous transactional access to CICS
- WebSphere J2EE integration: J2EE resource adapter
- Remote Java access: Gateway daemon
- GUI Configuration tool: ctgcfg
- COBOL, C and COM interfaces of CICS Universal Client

- **What it does *not* offer**

- Remote access for COM/VB clients
- Outbound access from CICS
- Data conversion facilities (handled by CICS itself or by Java)

Positioning



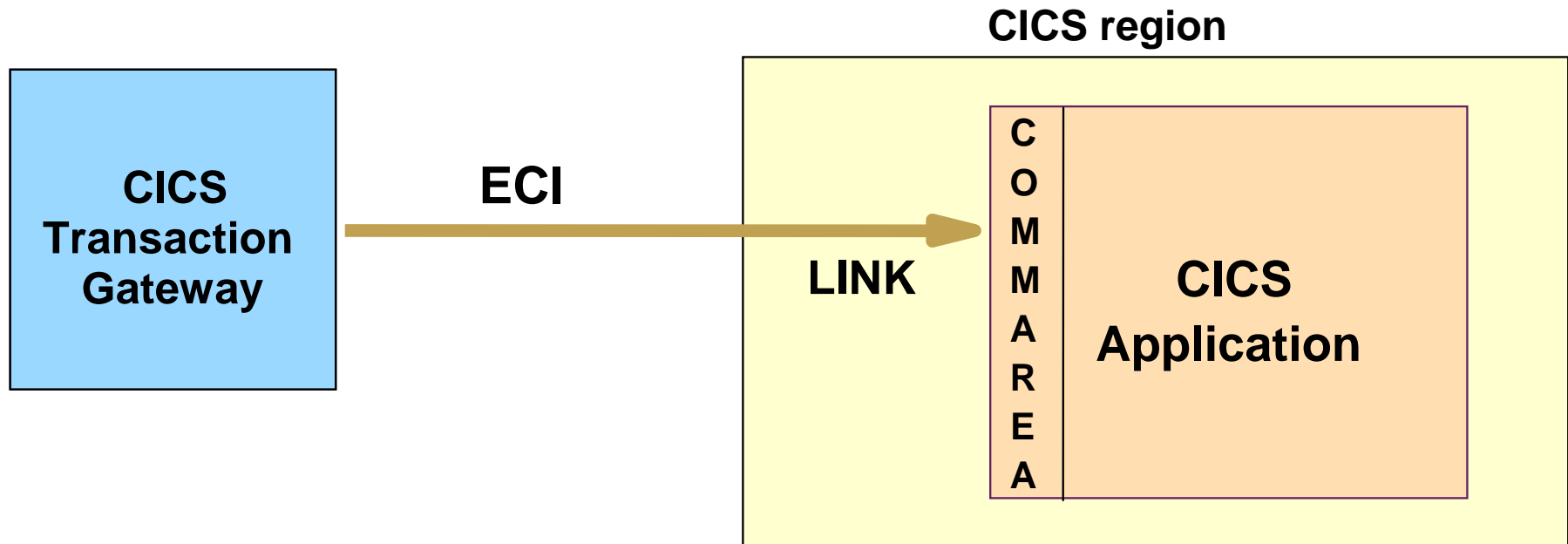
GA of CICS UC and CICS TG v5.1 Mar 12, 2004 - More details in announce 204-031



Part II

Application Development

ECI – linking to a COMMAREA



- API to LINK to a CICS program passing a COMMAREA
- Available on all platforms
- With z/OS based on EXCI
- Can use extended units of work (i.e. without SYNCONRETURN)
- Java Interfaces:
 - ECIRquest base class, CCF, J2EE/CCI interfaces
- Native interfaces
 - COM, C, C++, COBOL

ECI – Java code example

```
try {
// instantiate JavaGateway object
JavaGateway jg=new JavaGateway("tcp://gunner",2006);

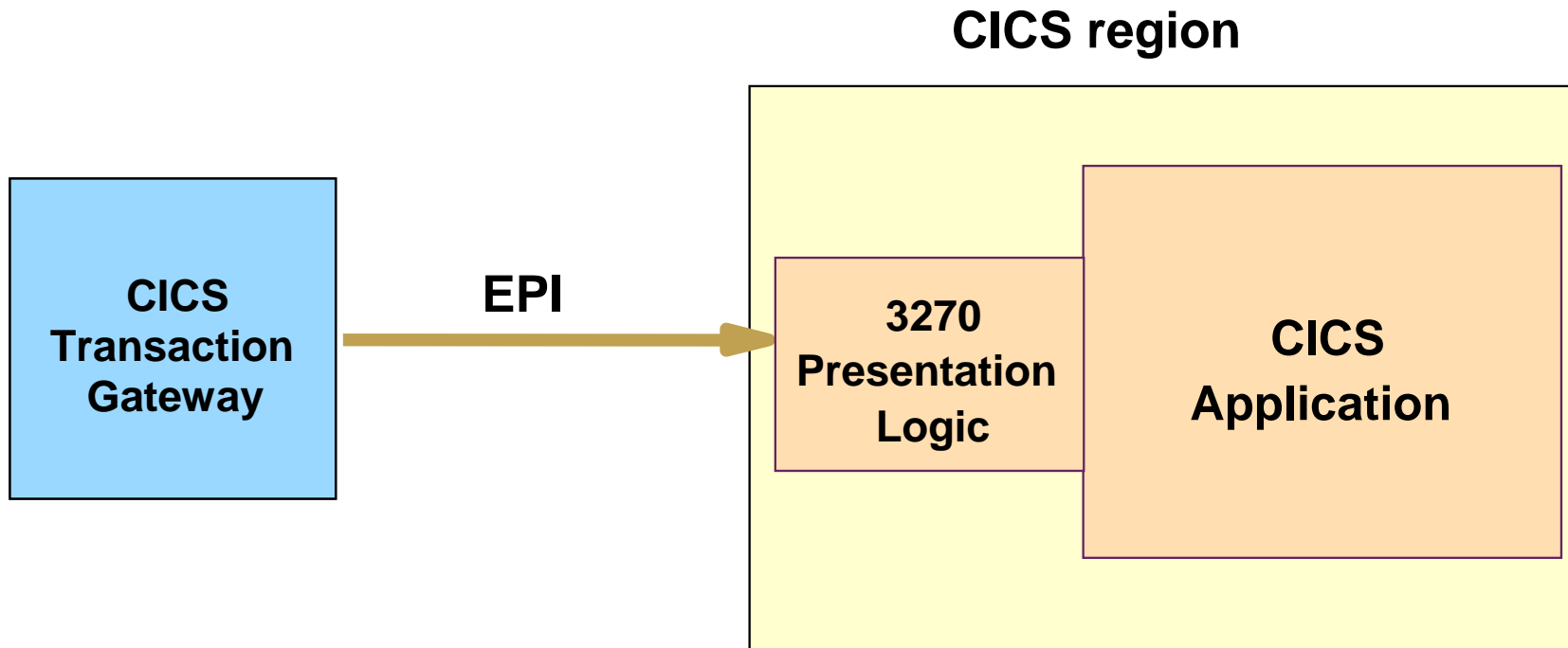
// instantiate COMMAREA byte array
byte commarea []=("-----").getBytes("IBM037");

// instantiate ECIRequest object
ECIRequest req=new ECIRequest(ECIRequest.ECI_SYNC,
    "SCSCPAA6", //CICS server name
    null,null, //userid & password
    "ECIPROG", //program name
    "CPMI", //transaction ID
    commarea, //COMMAREA data
    commarea.length, //COMMAREA length
    ECIRequest.ECI_NO_EXTEND, //Non-extended mode
    ECIRequest.ECI_LUW_NEW); //LUW token

//flow ECI request to the CTG
jg.flow(req);

// close the connection to the CTG
jg.close();
} catch (IOException ioe){
    System.out.println("Handled exception:"+ioe.toString());
}
```

EPI - accessing 3270 transactions



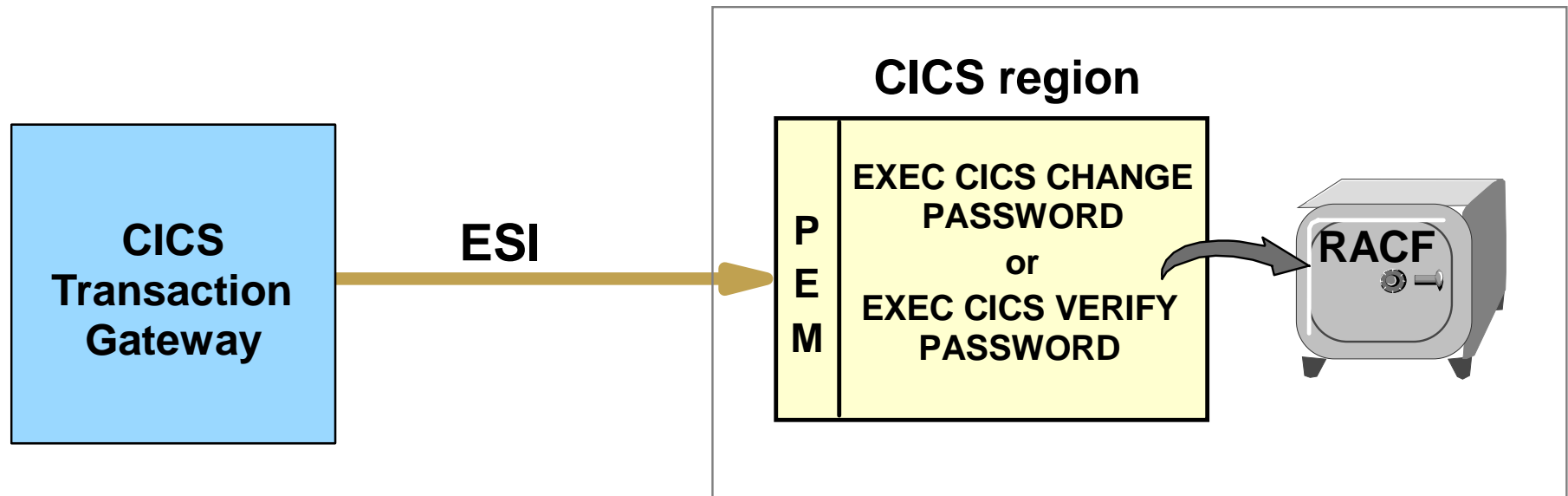
- API to access to a 3270 transaction
- Java Interfaces:
 - Base EPIRequest class, EPI support classes, EPI Beans, CCF, CCI
- Native interfaces
 - COM, C, C++, COBOL

EPI – Java Support classes

- Easier to use than the lower level EPIRequest class
- Elements include:
 - Classes to interact with the content of a 3270 screen including Terminal, Screen, and Field
 - Exception classes for error handling
 - BMSMapConvert utility - converts BMS maps to a Java class

EPI – Java code example

```
try {  
    EPIGateway epiGate = new EPIGateway("tcp://gunner", 2006);  
    Terminal term = new Terminal(epiGate, "SCSCPAA6", null, null);  
    Screen scr = term.getScreen();  
    Field fld = scr.field(1);  
    fld.setText("EPIP");           // Set Tranid to EPIP  
    scr.setAID(AID.enter);        // Sent AID key to enter  
    term.send();                  // Install EPI Terminal and get response  
    // Loop through the scr object and print all non zero length fields  
    for (int i = 1; i <= scr.fieldCount(); i++) {  
        fld = scr.field(i);  
        if (fld.textLength() > 0)  
            System.out.println("Field " + i + ":" + fld.getText());  
    }  
    term.disconnect();           // Disconnect terminal  
    epiGate.close();             // Close JavaGateway  
} catch (EPIException epiEx) {  
    epiEx.printStackTrace();  
}
```

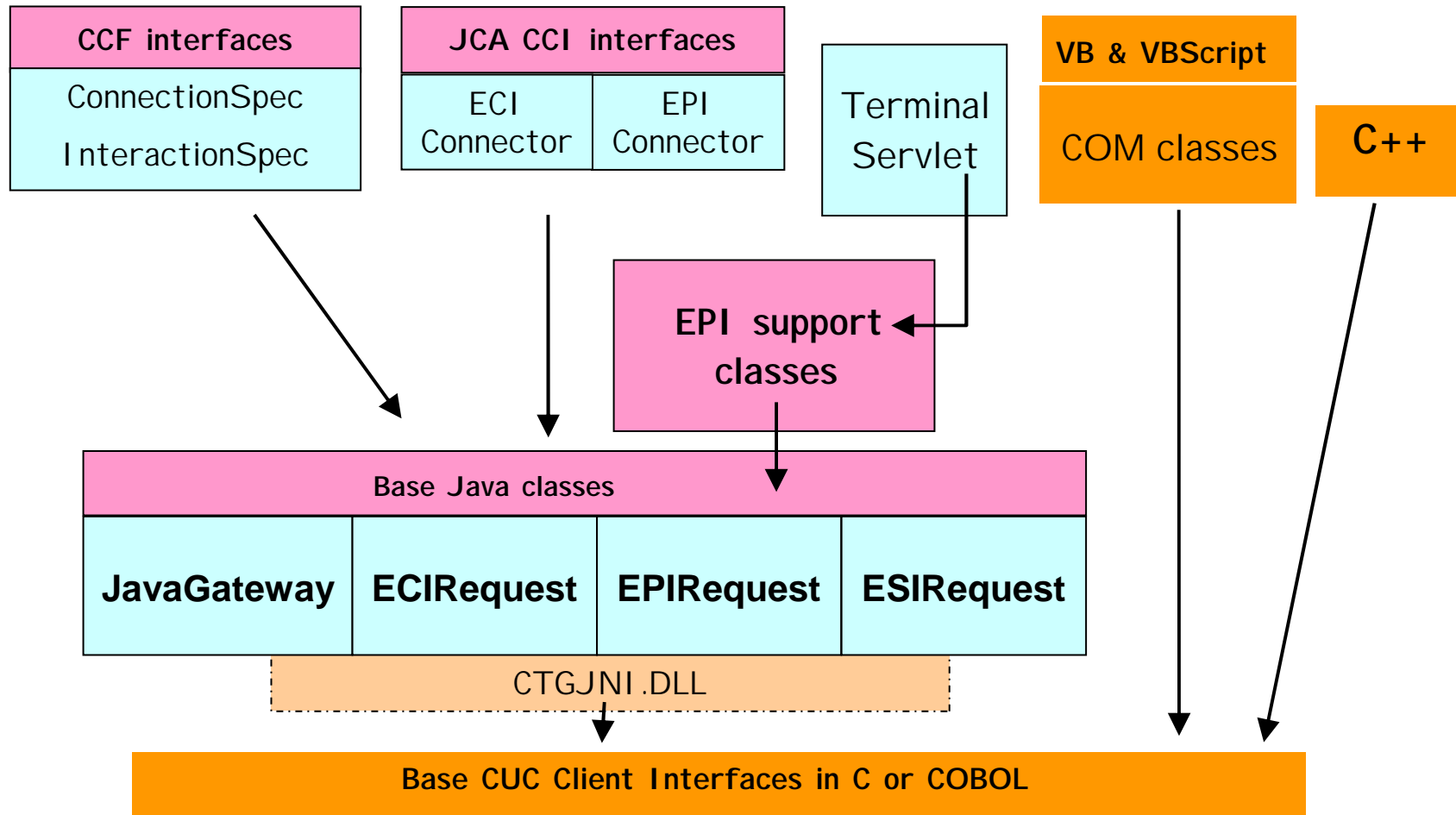


- Java interface to password management support in CICS/SAF
- Allows userid/password to be verified or changed
- To be used in conjunction with flowed ECI or EPI userids
- Java interface
 - Base ESIRequest class only
- Native interfaces
 - COM, C, C++, COBOL

ESI – Java code example

```
try {  
  
    JavaGateway jg= new JavaGateway("tcp://gunner", 2006);  
    ESIRequest req= ESIRequest.verifyPassword  
        ("CICSR3",          // Userid  
         "NOV2001",        // Password  
         "SCSCPAA7");      // CICS region  
    jg.flow(req);          // Flow request to CICS  
  
    System.out.println("Rc: "+req.getRc()); // Check return code  
  
    System.out.println("Expiry date: " + req.getExpiry());  
    System.out.println("Last access: " + req.getLastAccess());  
  
    jg.close();           // Close JavaGateway  
} catch (IOException ioe) {  
    System.out.println("Handled exception: " + ioe.getMessage());  
}  
}
```

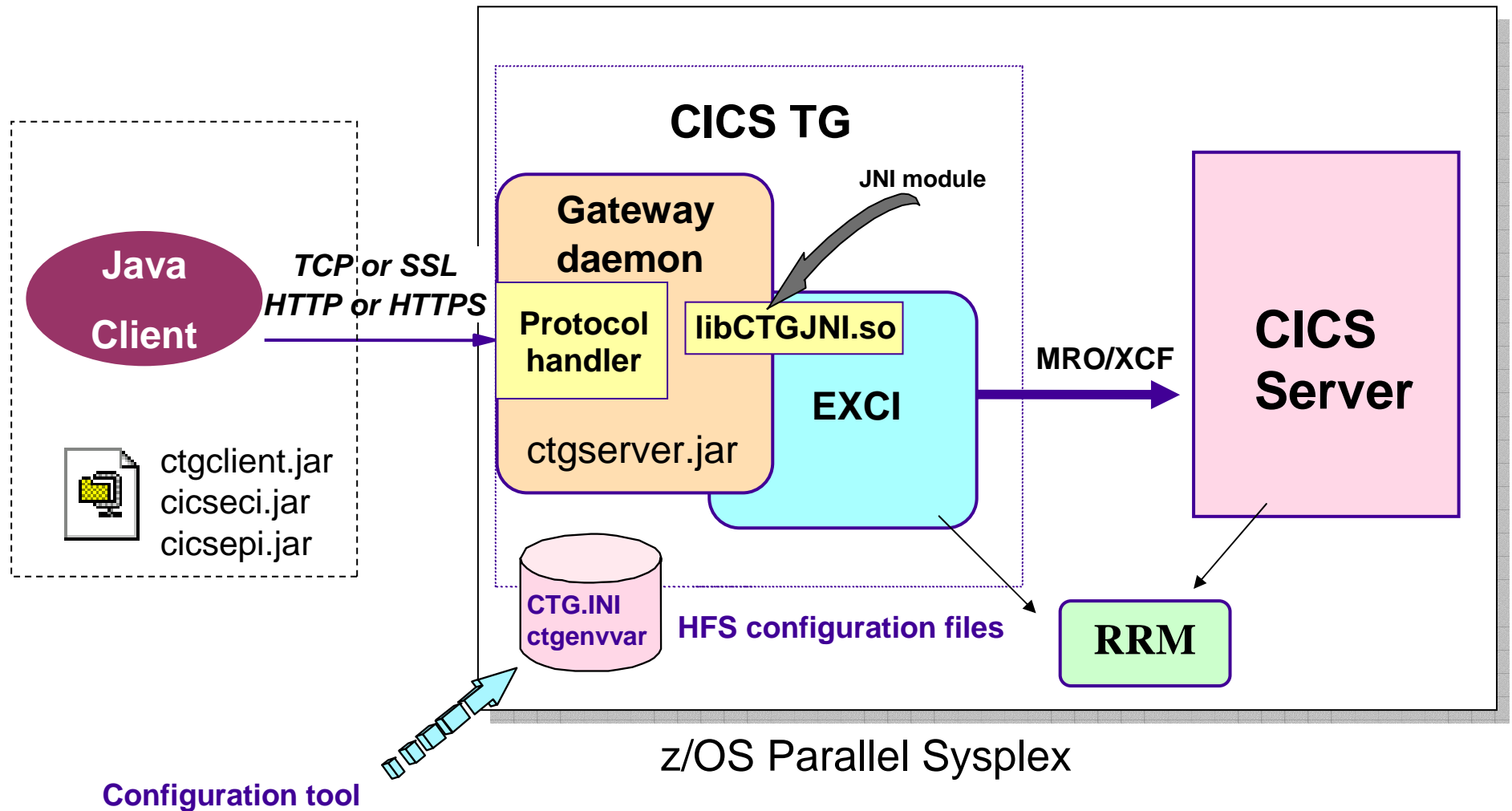
CICS TG - Interfaces



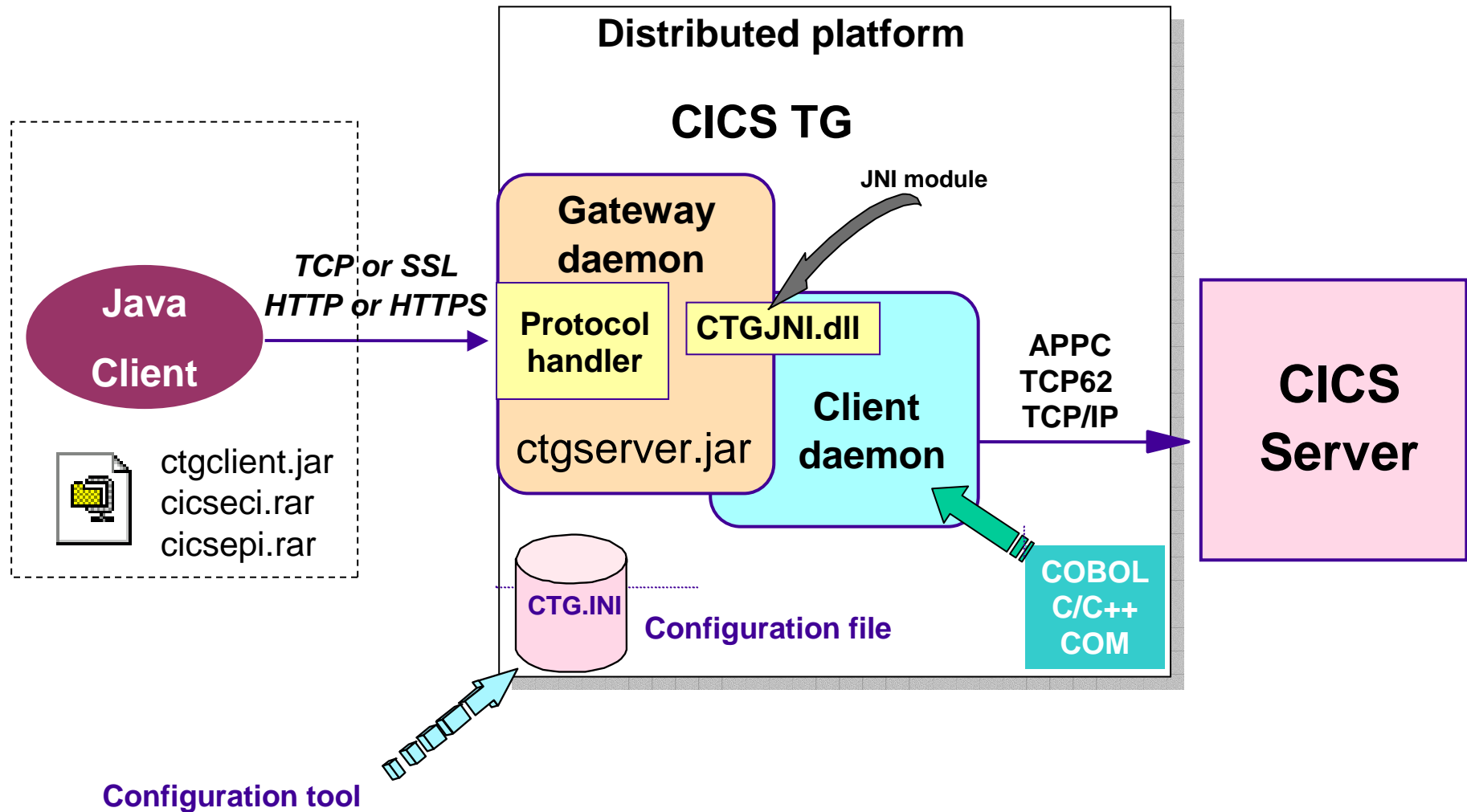
Part III

Components

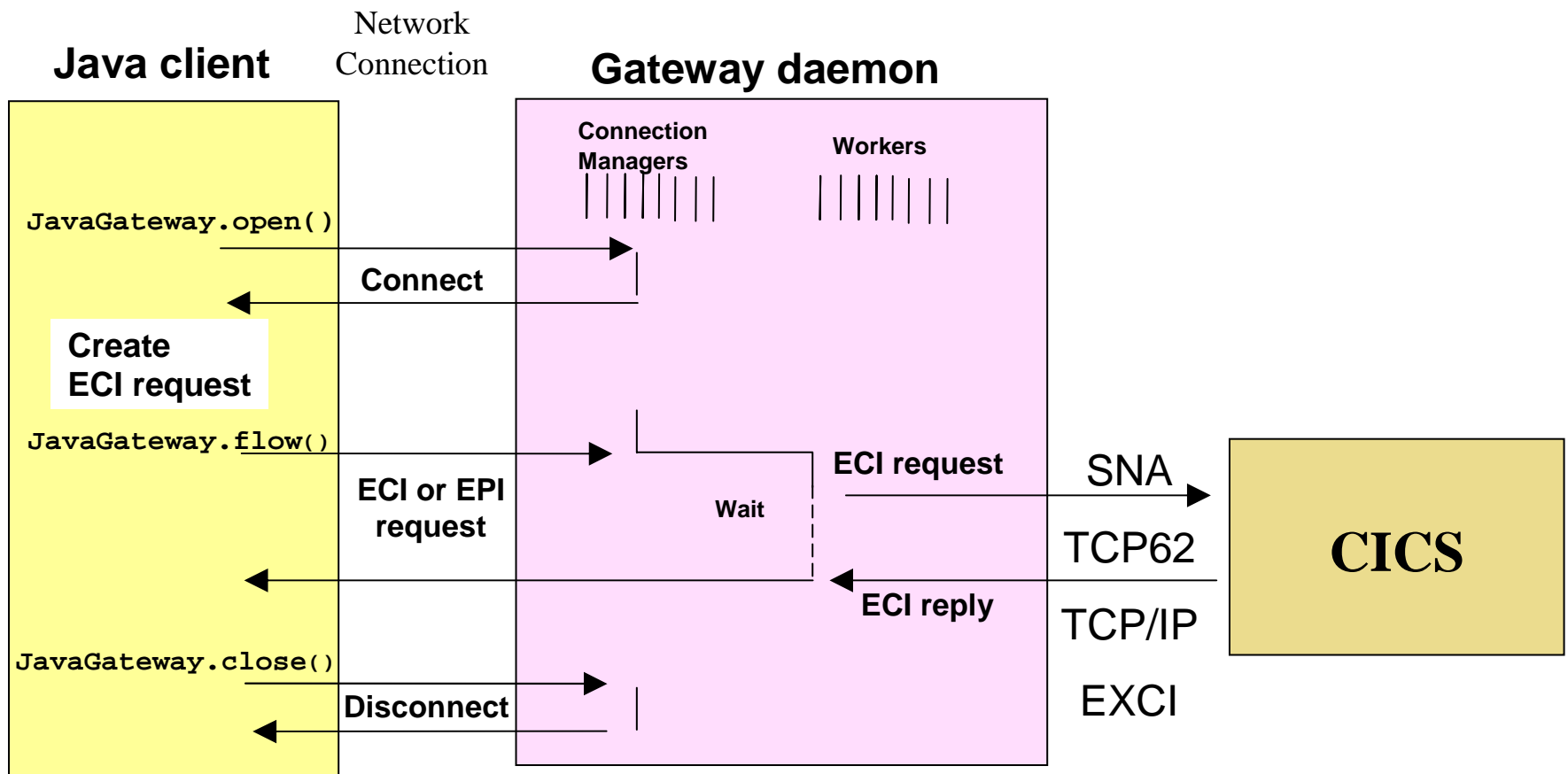
CICS Transaction Gateway – z/OS



CICS Transaction Gateway - distributed

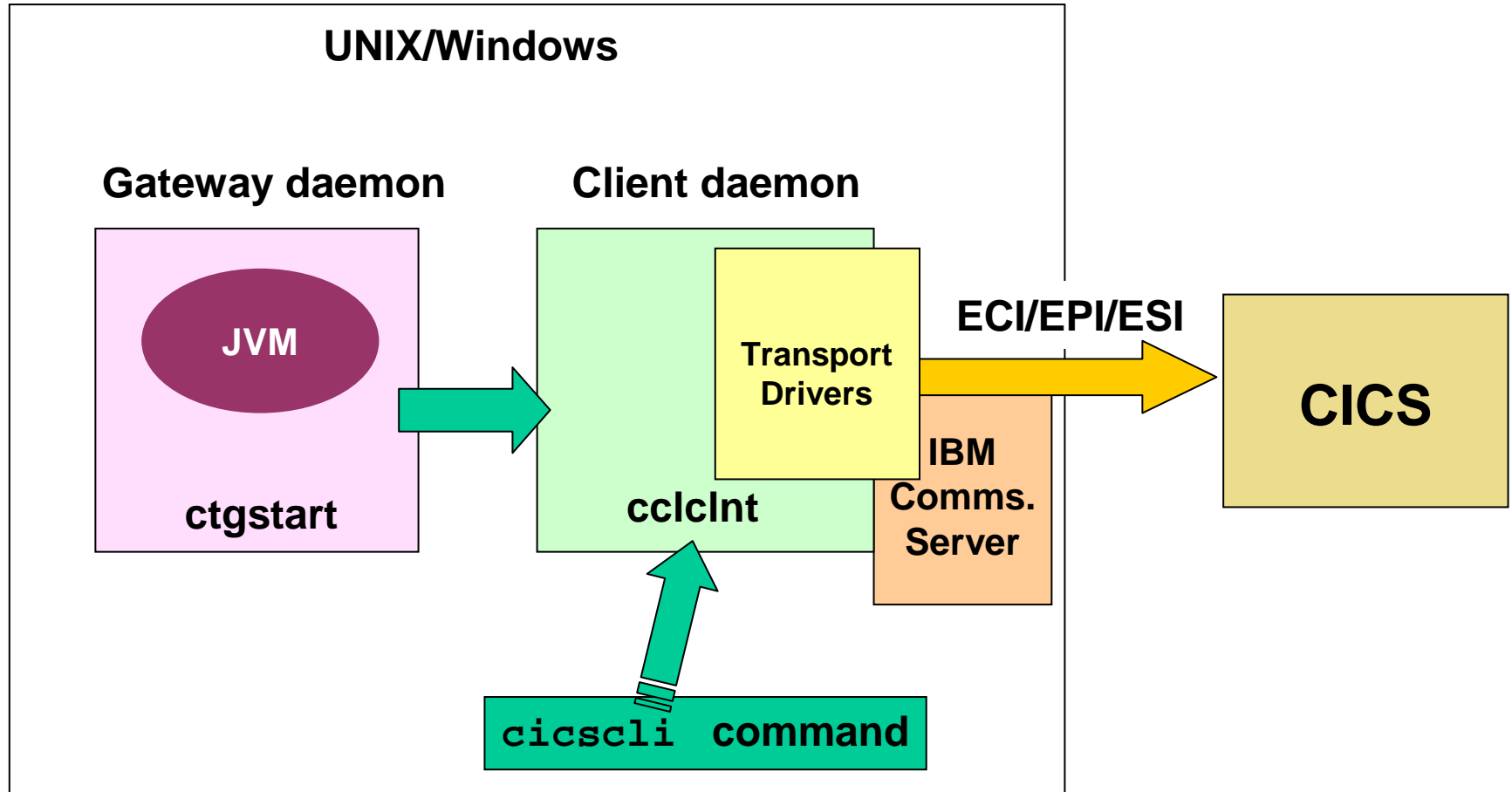


Gateway daemon

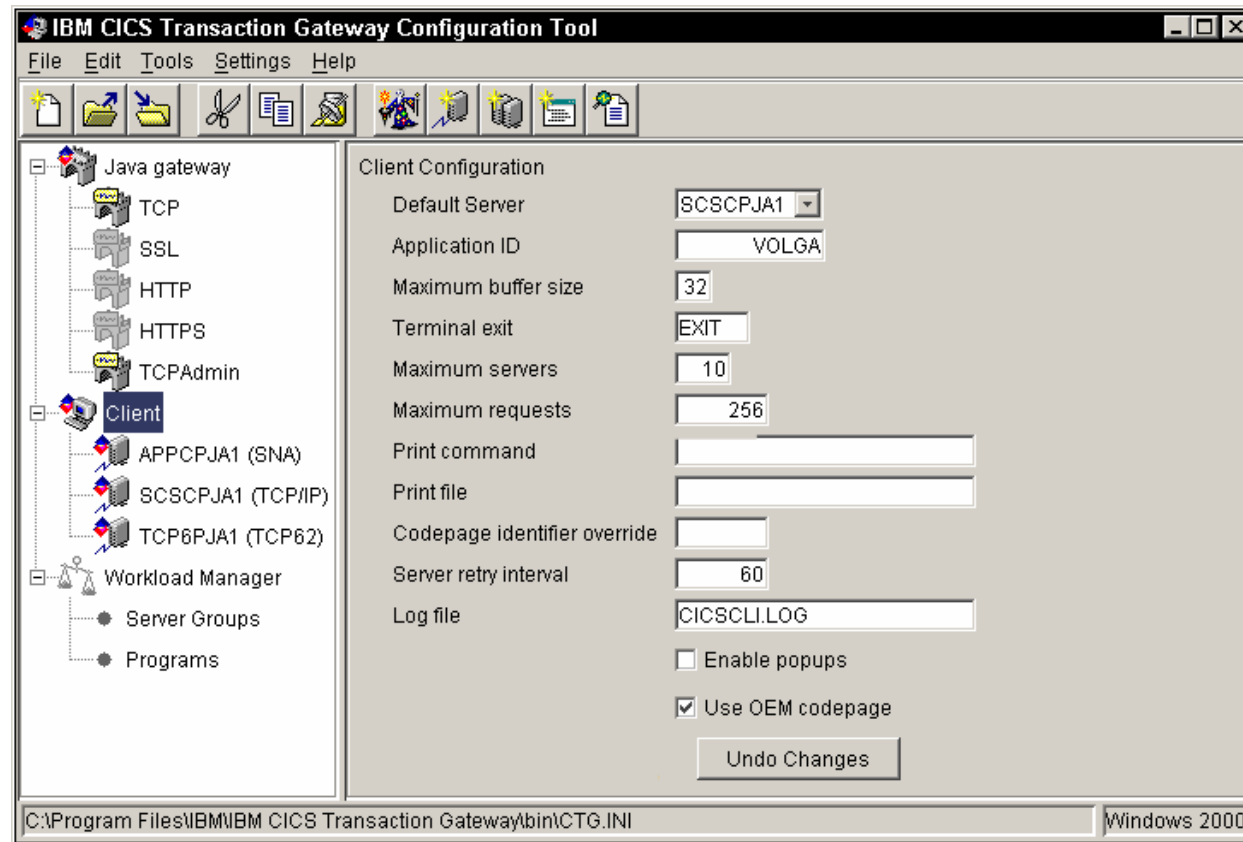


- Network connection can be via TCP, SSL, HTTP or HTTPS

Client daemon

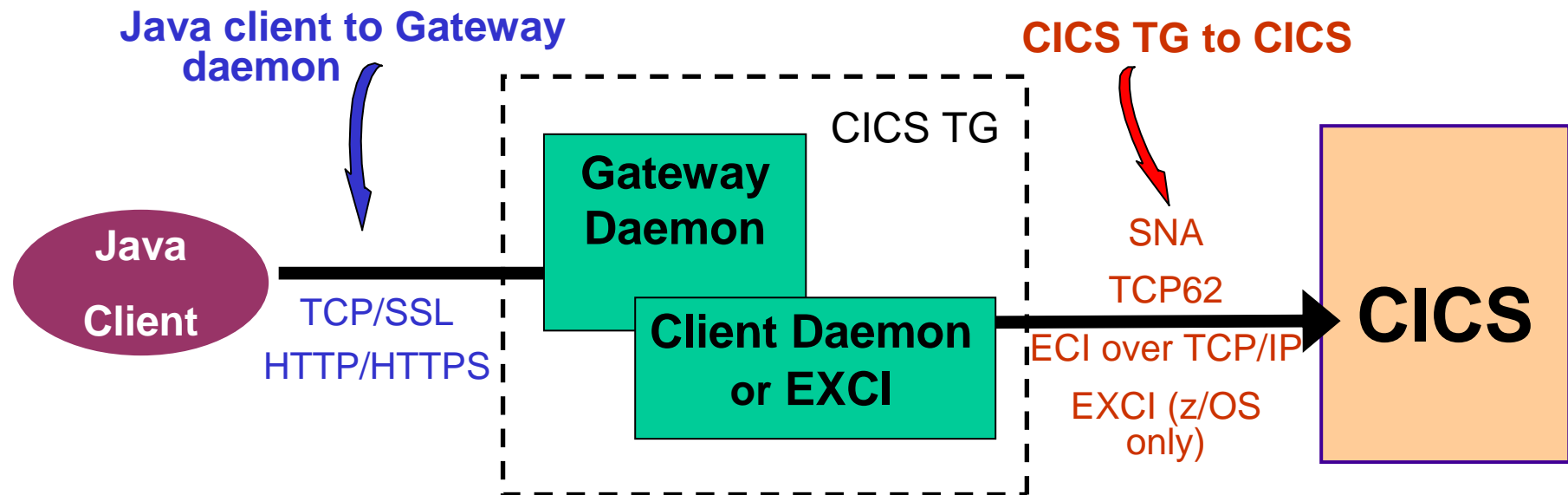


Configuration tool



- 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

Network protocols – Remote Gateway

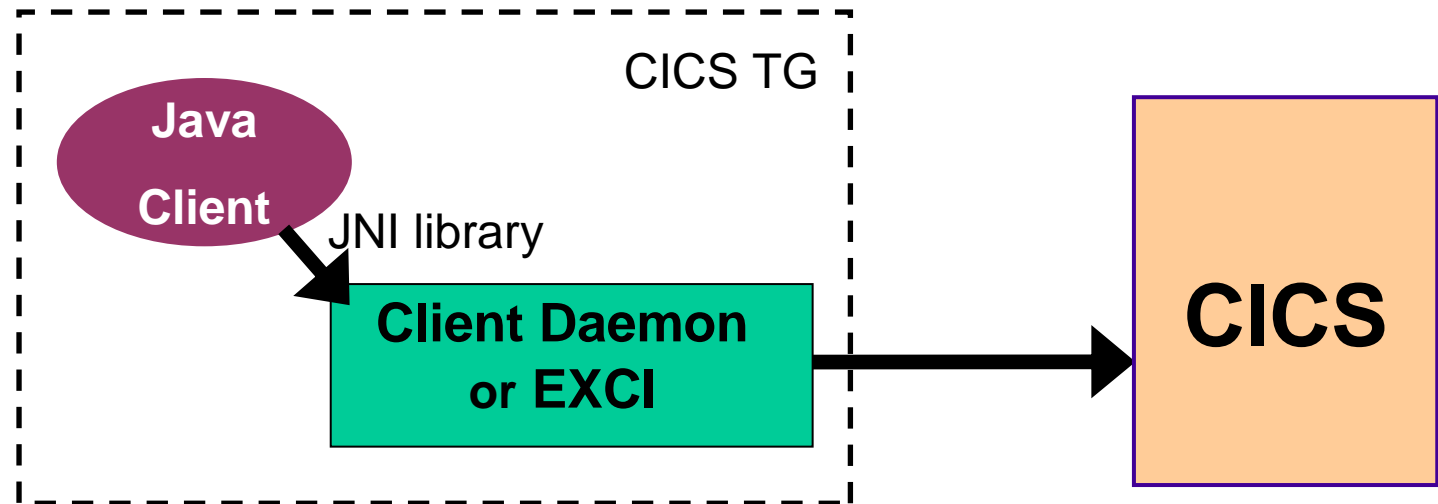


TCP and SSL protocols can use persistent connections - Local protocol gives best performance



Local protocol

- Java client on same machine as Gateway daemon
- Performance optimization for a local CTG



Local protocol gives significant performance advantage if CTG on same m/c as the Java client.
Typically utilized with WebSphere



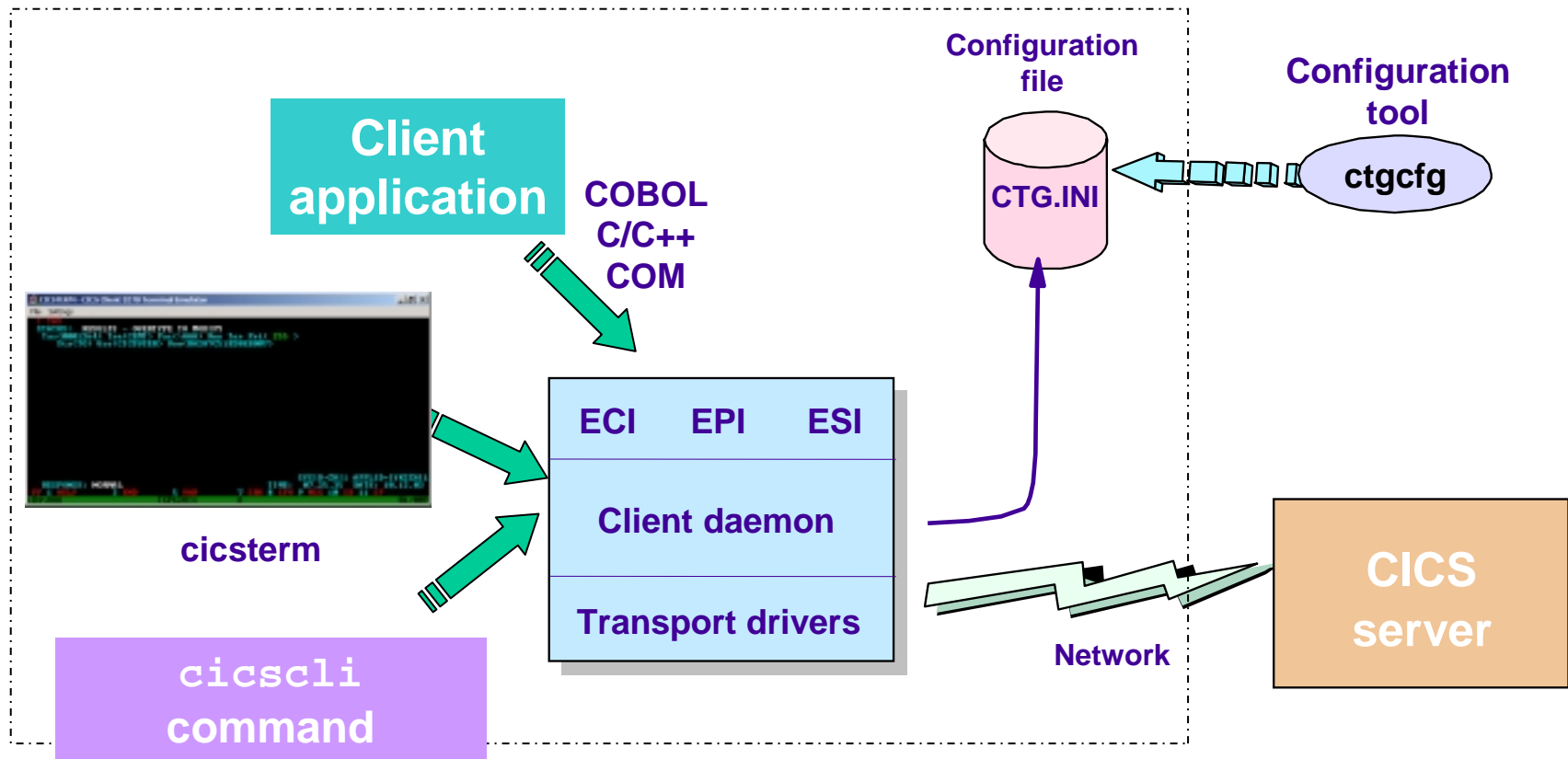
Part IV

CICS Universal Client

CICS Universal Client

- CICS UC is positioned for desktop use ONLY
 - CICS UC not licensed for use on servers
 - Priced offering, per seat
 - Replaces CICS UC V3.1.3 (EoS 2002)
 - Available on Win32: NT, 2000 and XP only
 - Supports C/C++, COBOL, VB, VBScript (COM)
 - ECI, EPI and ESI interfaces
 - Shared code base with CICS TG V5

CICS Universal Client - components



Components

- Client daemon: CCLCLNT.EXE
 - Provides runtime and transport drivers
 - Transport drivers (TCP62, TCP/IP, SNA)
 - APPC requires IBM Comms. Server or Microsoft Host Int. Server
- 3270 Terminal: CICSTERM
 - Provides 3270 terminal
- CICSCLI command interface
 - Provides start, stop, tracing control
- Logging and tracing
- Workload manager (Windows)
- APIs and user exits

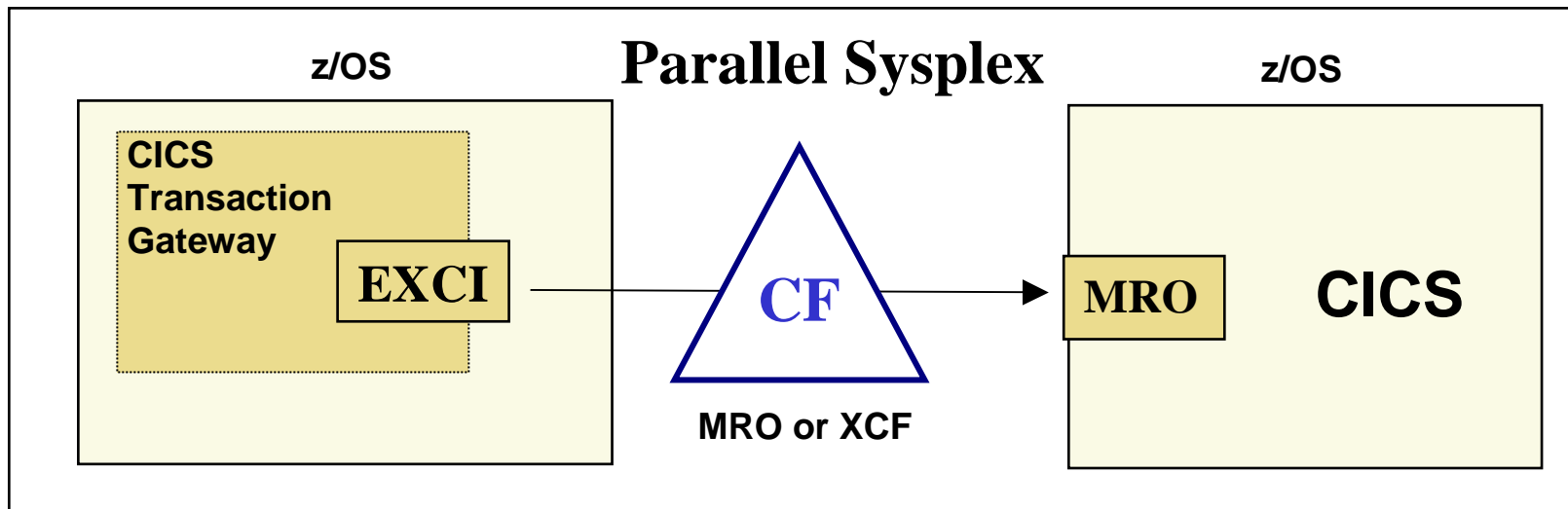
Part V

Network Protocols

Network protocols

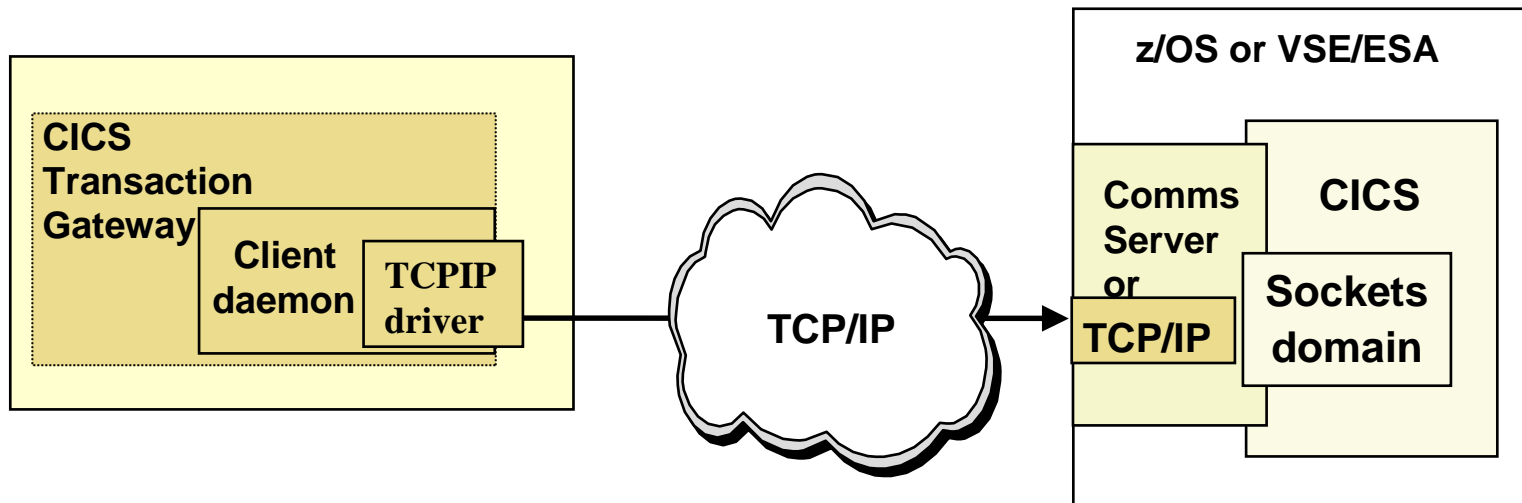
- EXCI
- TCP/IP
- APPC
- TCP62
- Named Pipes

EXCI



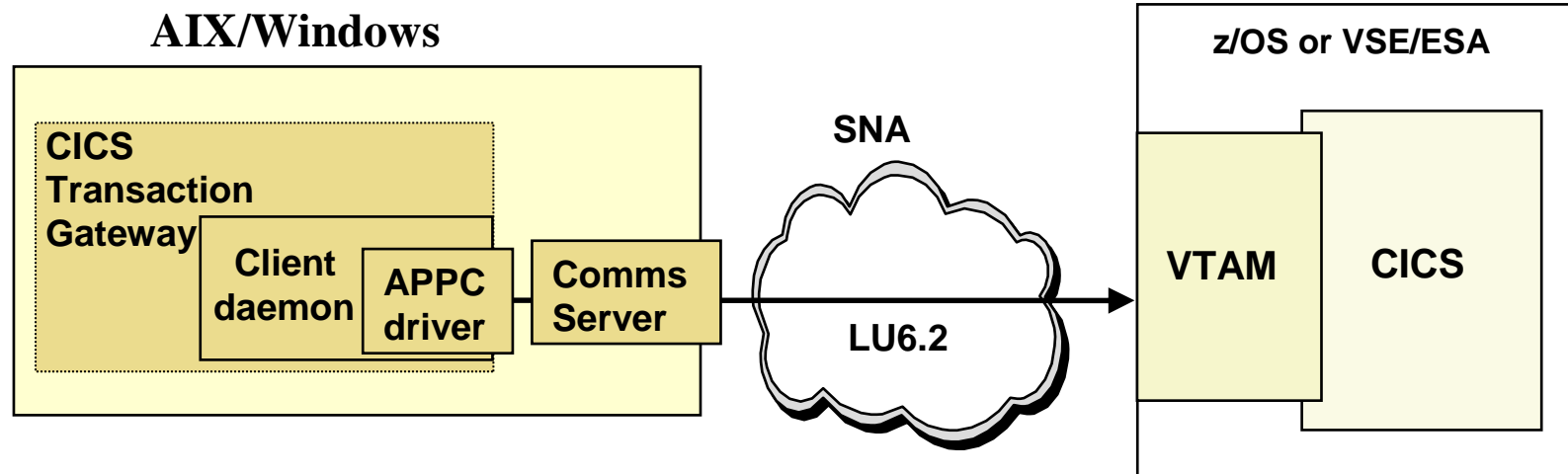
- Replaces function of Client daemon on z/OS
- ECI support only (no ESI or EPI)
- Can use XCF from LPAR to LPAR with Coupling Facility
- Transactional support (2PC) using RRS but:
 - Requires Gateway and CICS to be in same LPAR
- 100 pipe limit per calling address space
- V. high performance when using the MRO cross memory protocol

TCP/IP



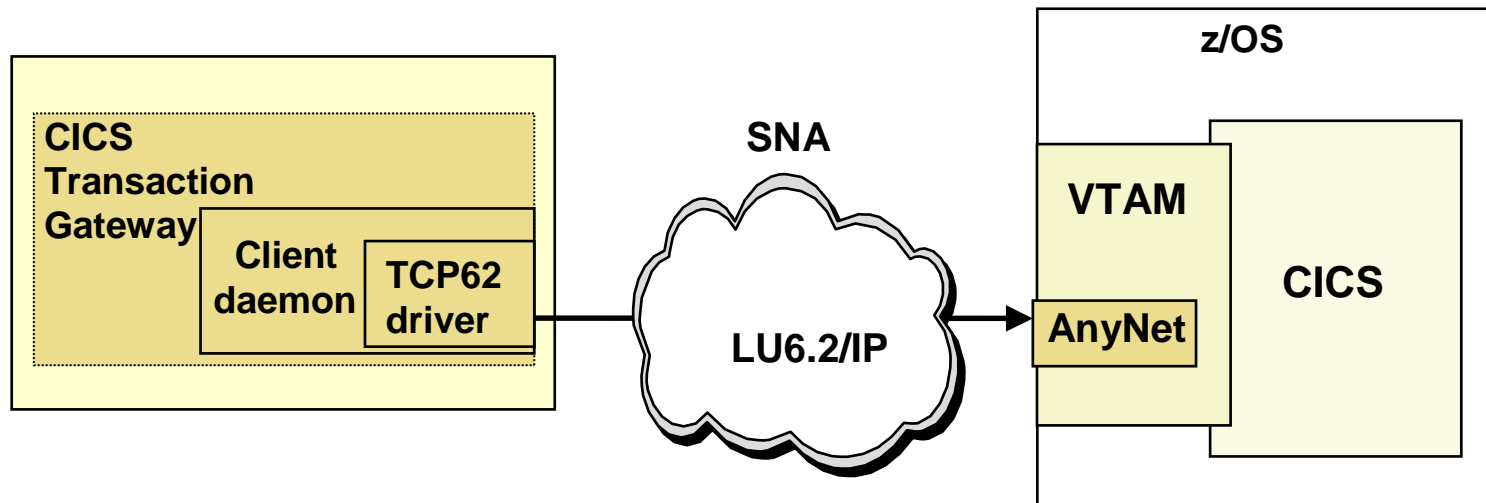
- Feature of CICS TS V2.2 & VSE/ESA 1.1.1 (not a feature of CICS TG V5)
- Backwards compatible with CICS TG V5, V4 and V3
- CICS TS for z/OS & VSE/ESA requires:
 - ECI TCPIP SERVICE
 - z/OS Communications Server, TCP/IP support
 - CICS TCP/IP listener (CSOL)
 - CIEP transaction
- CICS TS for z/OS and VSE/ESA supports ECI only
- TxSeries/CICS TS for Windows/CICS TS for iSeries supports EPI and ECI
- Very high performance when using Hipersockets from Linux on zSeries

APPC



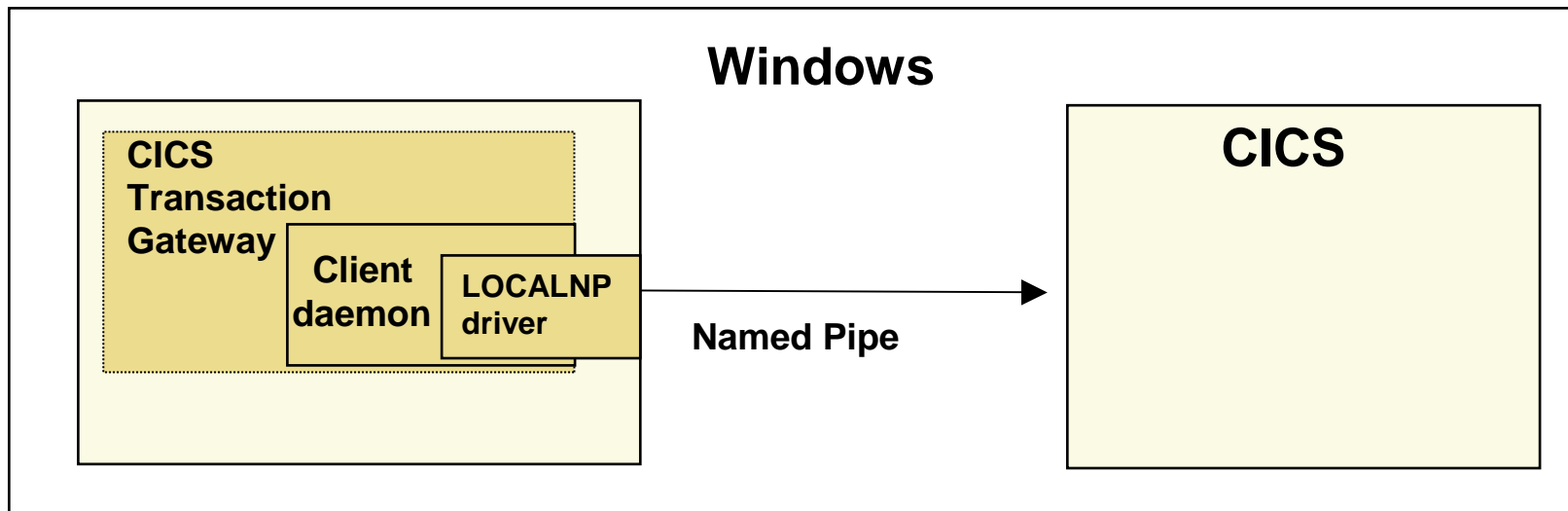
- LU6.2 connection into CICS
- Requires:
 - VTAM
 - LU6.2 parallel sessions in CICS
- Workstation software:
 - IBM Personal Communications
 - IBM Communications Server (Windows/AIX)
 - Microsoft Host Integration Server
- Supports EPI, ECI and ESI Interfaces

TCP62



- LU6.2 encapsulation via TCP/IP
- Requires:
 - VTAM + AnyNet on z/OS
 - LU6.2 parallel sessions in CICS
- All function provided by CICS TG
 - No requirement for IBM Personal Communications
 - Configured using CTG.INI and etc/hosts file
- Supports EPI, ECI and ESI Interfaces

Named Pipes

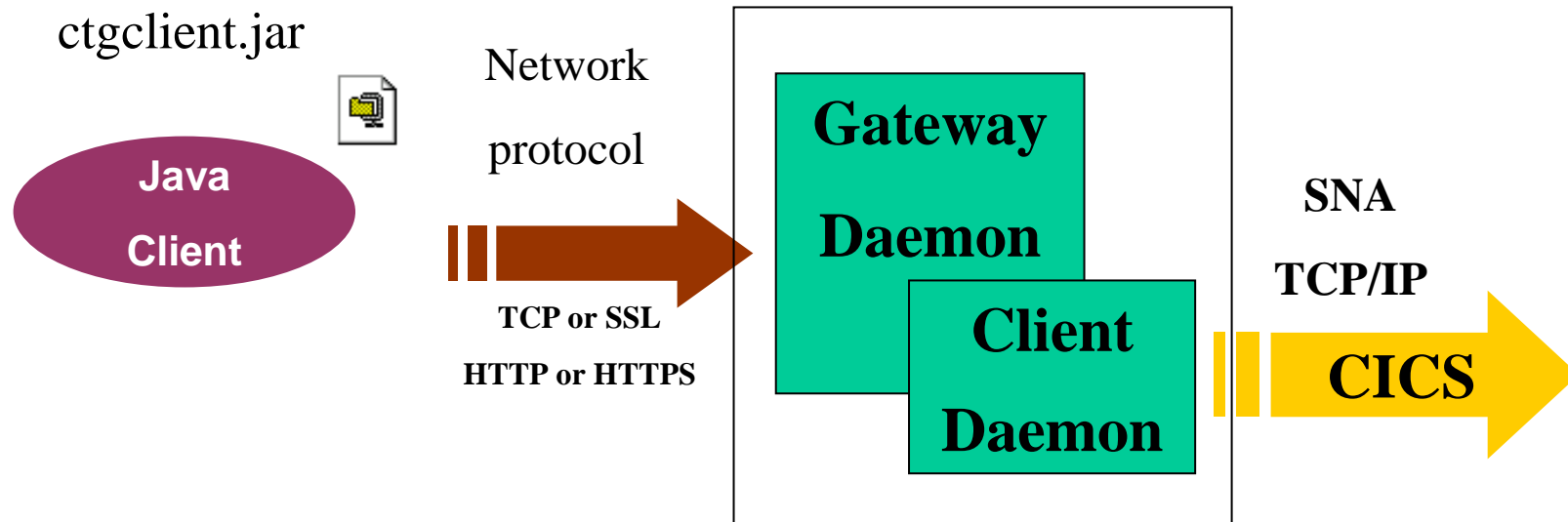


- TxSeries and CICS TS regions can use Windows named pipes
- Provides a fast communications method independent of any network protocol

Part VI

Deployment

Java Client

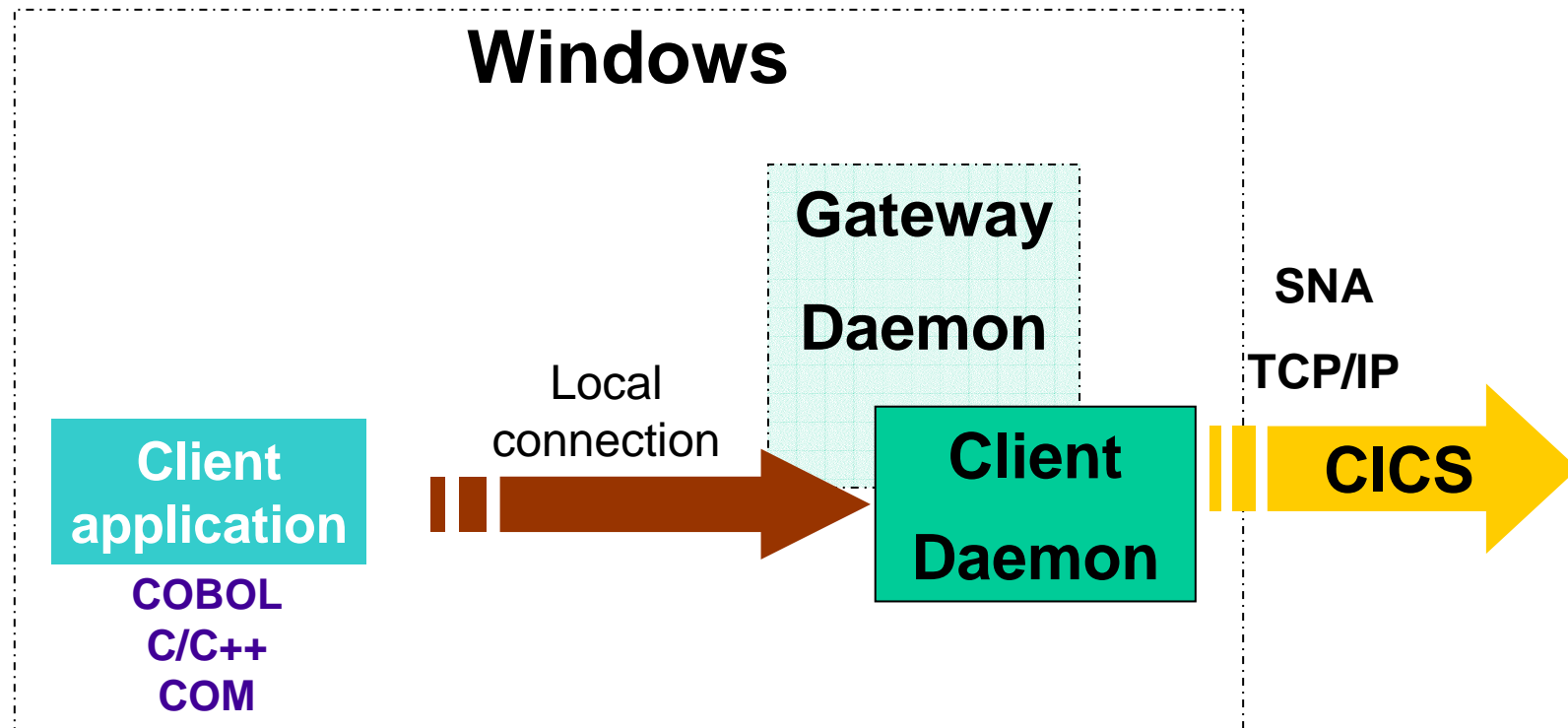


Q. What do I need

A. Supported JVM & the Java class libraries

- Standalone Java application (fat client)
- Applet
- Servlet
- Session bean (J2EE connector/JCA)

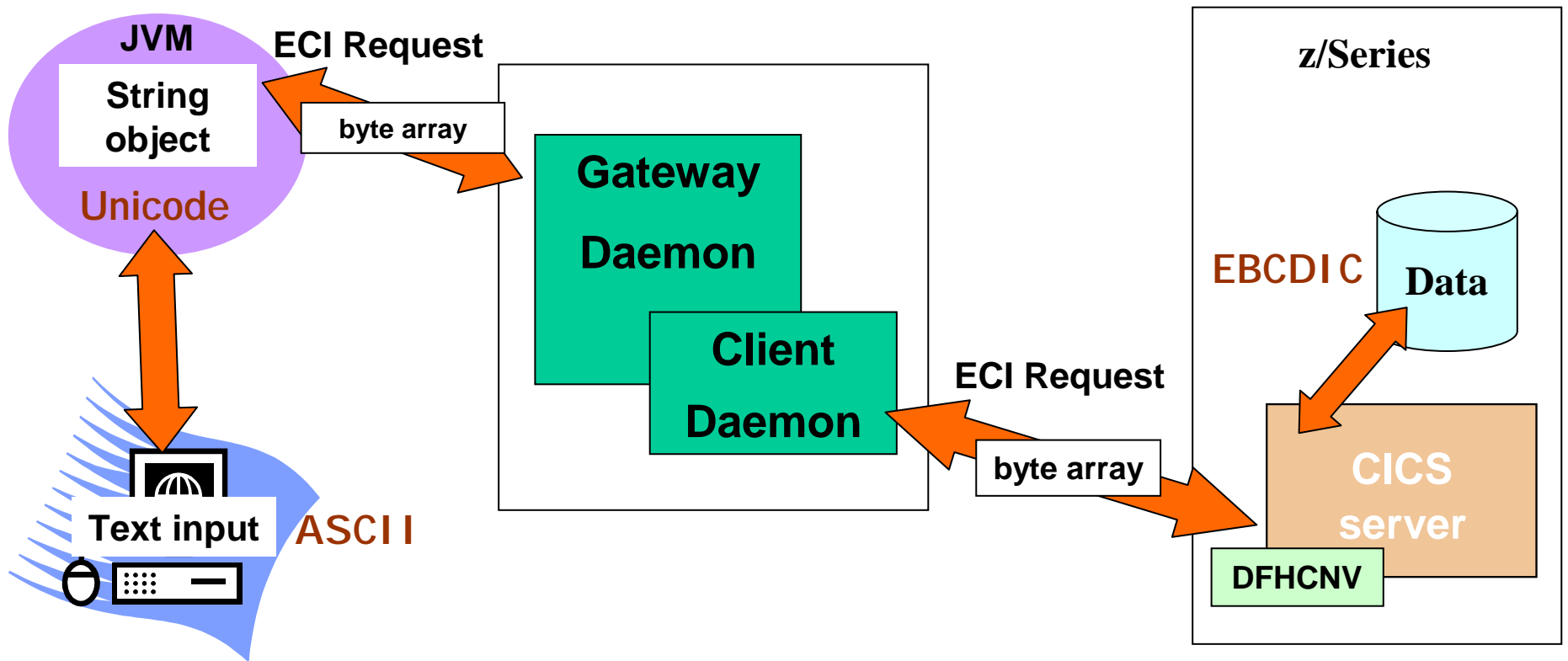
C/COBOL/COM applications



Windows: COBOL and C
UNIX or Windows: C/C++:

Note on z/OS: COBOL, C, PL/I, Assembler (via EXCI)

Data conversion



Problem:

Need to convert ASCII character data from Windows PC/Web browser to EBCDIC data used on z/Series, via Unicode Java objects.

Solution:

1. Flow EBCDIC to CICS: Convert in Java – `getBytes()`, or EAB or WSAD-IE
2. Flow ASCII to CICS, convert to EBCDIC in CICS – DFHCNV templates

For further details refer to Redbook SG24-6401 Appendix B

Part VII

CICS TG V5 and beyond

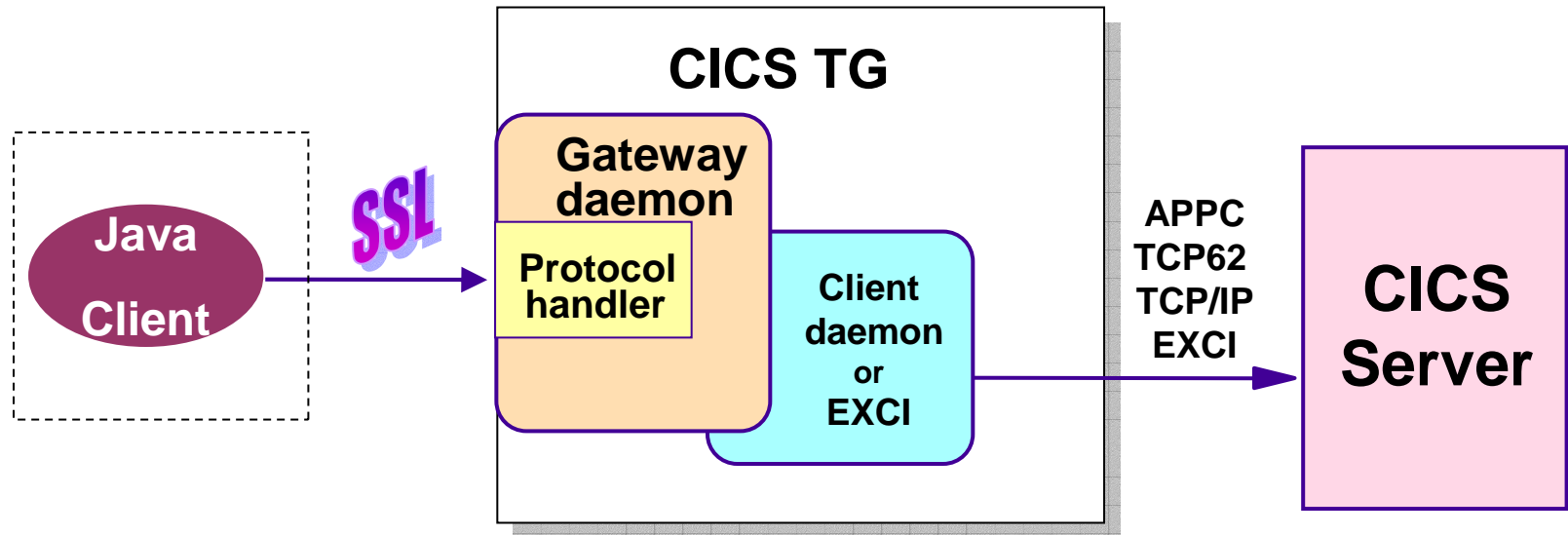
Highlights of CICS TG V5.0

- Strategic connector from WebSphere to CICS
- Evolution from CCF to J2EE Connector Architecture
- Packaged with WebSphere Studio Application Developer/ Integration Edition
- Separately priced package - enables standalone sales with any app server
- Same pricing on distributed & mainframe platforms
- In-line CICS UC V5 on Windows platforms
- Updated platforms:
 - z/OS V1R4
 - AIX v5.2
 - Windows XP
 - Linux kernel 2.4

New in V5.0

- 128 bit SSL
- Dynamic control of tracing
- ARM support on z/OS
- Performance enhancements for data transfers
- Extended EXCI logging
- Support for the J2EE Connector Architecture
- Support for WebSphere V5

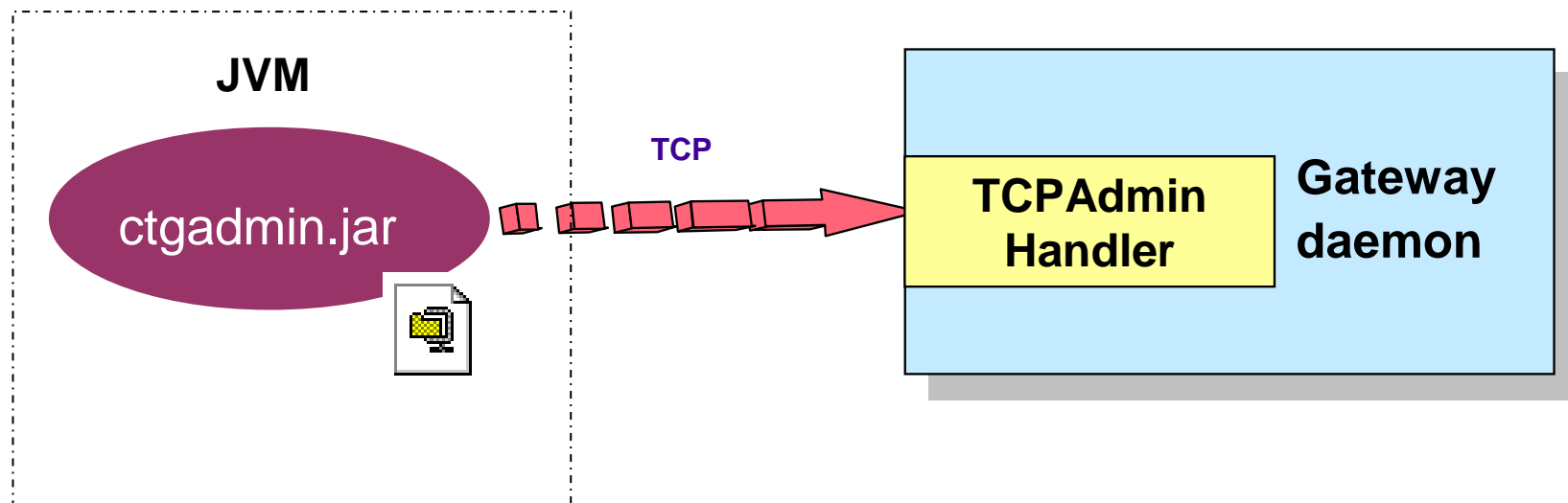
128 bit SSL - JSSE



- 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
- SystemSSL 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



Dynamic Tracing - output

```
C:\Program Files\IBM\IBM CICS Transaction Gateway\classes>java -jar ctgadmin.jar  
-ctg=tcp:\\winmvs2c:2008 -a=qtrace
```

```
CTGCtrl - CTG Control Program, version 5.0.1  
(C) Copyright IBM Corporation 2002. All rights reserved
```

Gateway trace settings:

```
tlevel=3  
truncationsize=80  
dumpoffset=0  
tfile=/u/wakelin/ctg501/logs/ctg.trc  
tfilesize=0
```

JNI trace settings:

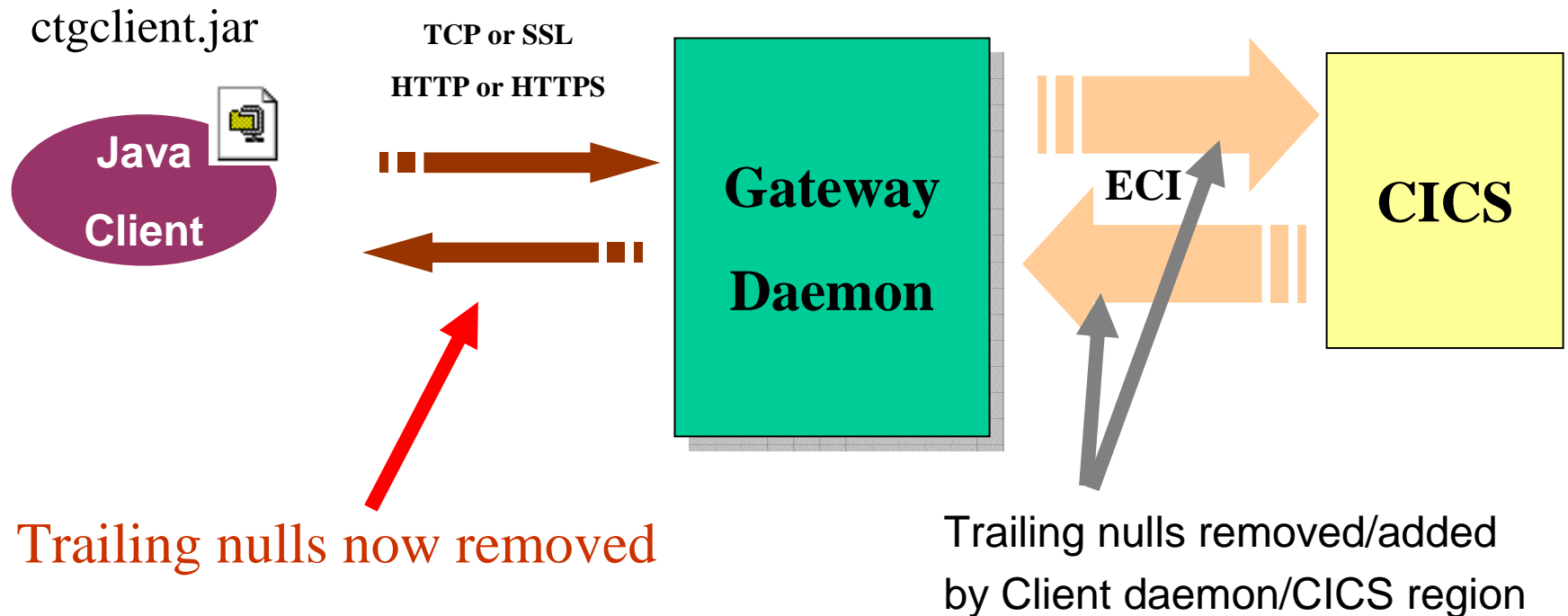
```
tlevel=1  
tfile=/u/wakelin/ctg501/logs/jni.trace
```

The command completed successfully

ARM support

- MVS Automatic Restart Manager
- Allows MVS to restart the Gateway automatically if it fails
- `ctgarm` command used to register the Gateway daemon with z/OS before it starts
- Note: ARM deregistration not possible if Gateway daemon running as a started task

Improved data handling



- 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
```

CTG v5.01

- **GA Aug 1,2003**
- **WAS z/OS V5 support** (including z/OS.e)
- **Other function**
 - Full accessibility (samples and graphical interfaces)
 - Includes discontinuation of EPIBeans, VB/VBscript samples, and Terminal Servlet
 - AIX v5.2 and z/OS v1.4 support
 - J2EE ECIInteractionSpec methods: setTPNName() & setTranName()
 - Control of SSL cipher suite (enforcement of 128 bit SSL)
 - Memory mapped 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 retrievable failed EXCI allocates
 - Non-swappable CTG address space
 - Improved performance of compression exits

V5.1

- GA March 2004
- Linux on Intel 32-bit architecture
 - CICS Universal Client
 - CICS Transaction Gateway
- Java 1.4.1 support
 - Support for WebSphere App. Server V5.1
 - Support for native JSSE (supplied by Java SDK)
- Up-levelling of supported operating systems
 - Windows 2003
 - Solaris 9
- Service refresh

CICS Connectors ... beyond V5

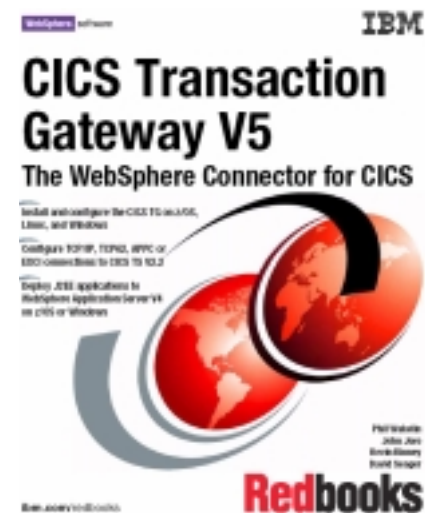
- So what's been achieved?
 - JCA connector (JCA 1.0) support
 - ECI over TCP/IP for CICS TS for z/OSV2 & CICS TS for VSE/ESA V1.1.1
 - LINKable 3270 bridge for ECI (z/OS)
 - Support of connectors in CICS TS
 - Transactional support in WebSphere Application Server
 - Full accessibility
 - Common SSL on all platforms (JSSE)
- Future directions
 - Interoperation:
 - Support of WebSphere and JCA environment
 - Integration with the latest IBM technologies
 - Support for new operating systems
 - Customer value
 - Enhanced management
 - Monitoring interface
 - Architectural limits
 - Improved performance
 - Standards:
 - Accessibility
 - JCA 1.5

More Information

- CICS TG Configuration PDFs for WebSphere
 - <http://www-3.ibm.com/software/htp/cics/library/cicstgv5.html>
- WebSphere v5 for distributed platforms: SC34-6294
 - Deploying the ECIDateTime and EPIPlay J2EE samples into WebSphere Application Server Version 5.0
- WebSphere v5 for z/OS: SC34-6306
 - Deploying the ECIDateTime J2EE sample into WebSphere Application Server Version 5.0 for z/OS

Redbooks

- Java Connectors for CICS SG24-6401
- CICS TG v5 , The WebSphere Connector for CICS - SG24-6133-01



Other CICS TG talks

- **CT07 JCA and CICS: An Intro to the J2EE Connector Architecture**
 - Introduction to the JCA
 - Nigel Williams
- **CT08 Using the J2EE Connector Architecture to access CICS from WebSphere Application Server for z/OS V5**
 - Setup, performance and security information for using the CICS ECI resource adapter with WebSphere z/OS V5
 - Nigel Williams
- **C52 CICS Transaction Gateway Problem Determination**
 - Problem determination techniques for use with the CICS TG for z/OS
 - Richard Johnson
- **C53 Implementing CICS Transaction Gateway on z/OS**
 - How to configure the CICS TG on z/OS
 - Leigh Compton
- **C54 Connecting Remote Gateway and Clients to CICS TS**
 - How to configure the CICS TG on midrange platforms
 - Leigh Compton
- **C55 CICS Transaction Gateway – The WebSphere Connector for CICS**
 - Integration with WebSphere and analysis of security and transactions available
 - Phil Wakelin