Session# E09

The CICS Transaction Gateway: Web and Java access to CICS

Chris Smith
smithch@uk.ibm.com

Orlando, 31 May - 3 June 2000

RETURN TO INDEX

Trademarks

- The following terms are trademarks of International Business Machines Corporation in the United States and/or other countries:

AIX	DB2	OS/390	VisualAge
CICS	MVS/ESA	VSE/ESA	
CICS/VSE	OS/2	VTAM	

- Java and Solaris are trademarks of Sun Microsystems, Inc
- Windows, Windows 95, Windows 98, and Windows NT are trademarks of Microsoft Corporation, Inc
- Other company, product, and service names may be trademarks or service marks of others

Agenda

- The CICS Transaction Gateway
  - Overview
  - CICS Universal Clients
  - Gateway structure
  - Terminal Servlet
  - Network protocols
  - Connectivity to CICS
  - Security
  - Application Programming Interfaces
  - Common Connector Framework
- Summary

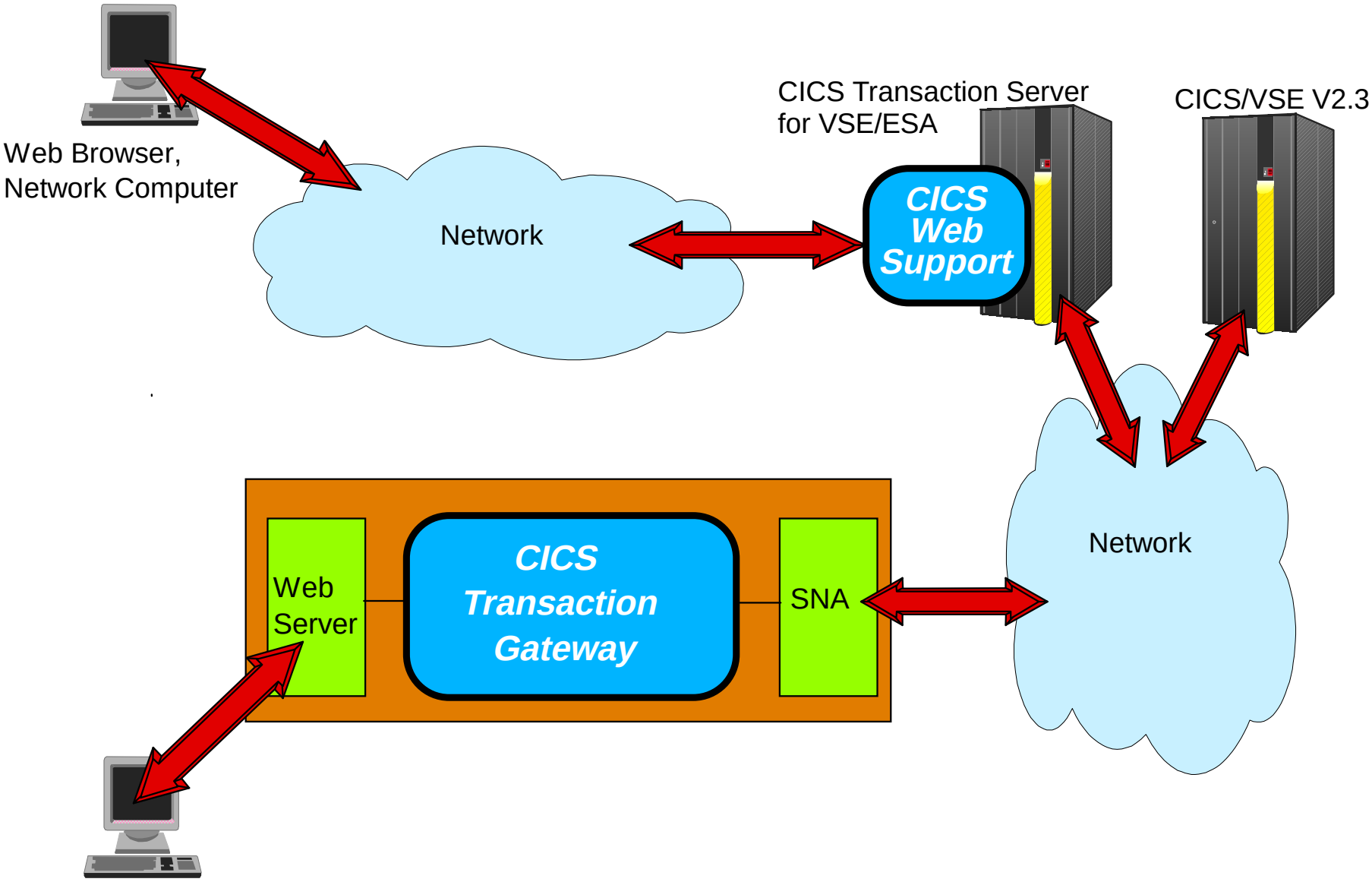
Overview

- Provides an interface to CICS from Java and the Web....
  - from a Web Browser or Network Computer
  - from any Java execution environment
- Provides access via....
  - Java programs
  - HTML pages
- Replaces....
  - CICS Gateway for Java
  - CICS Internet Gateway

Overview....

- Runs on Windows NT, AIX, Solaris, OS/2, OS/390
  - Also Windows 95/98 for Development purposes
- Supports multiple concurrent users and CICS conections
- Delivered with CICS Transaction Servers
  - Also delivered with VisualAge for Java Enterprise Edition
  - Also downloadable from the CICS web site

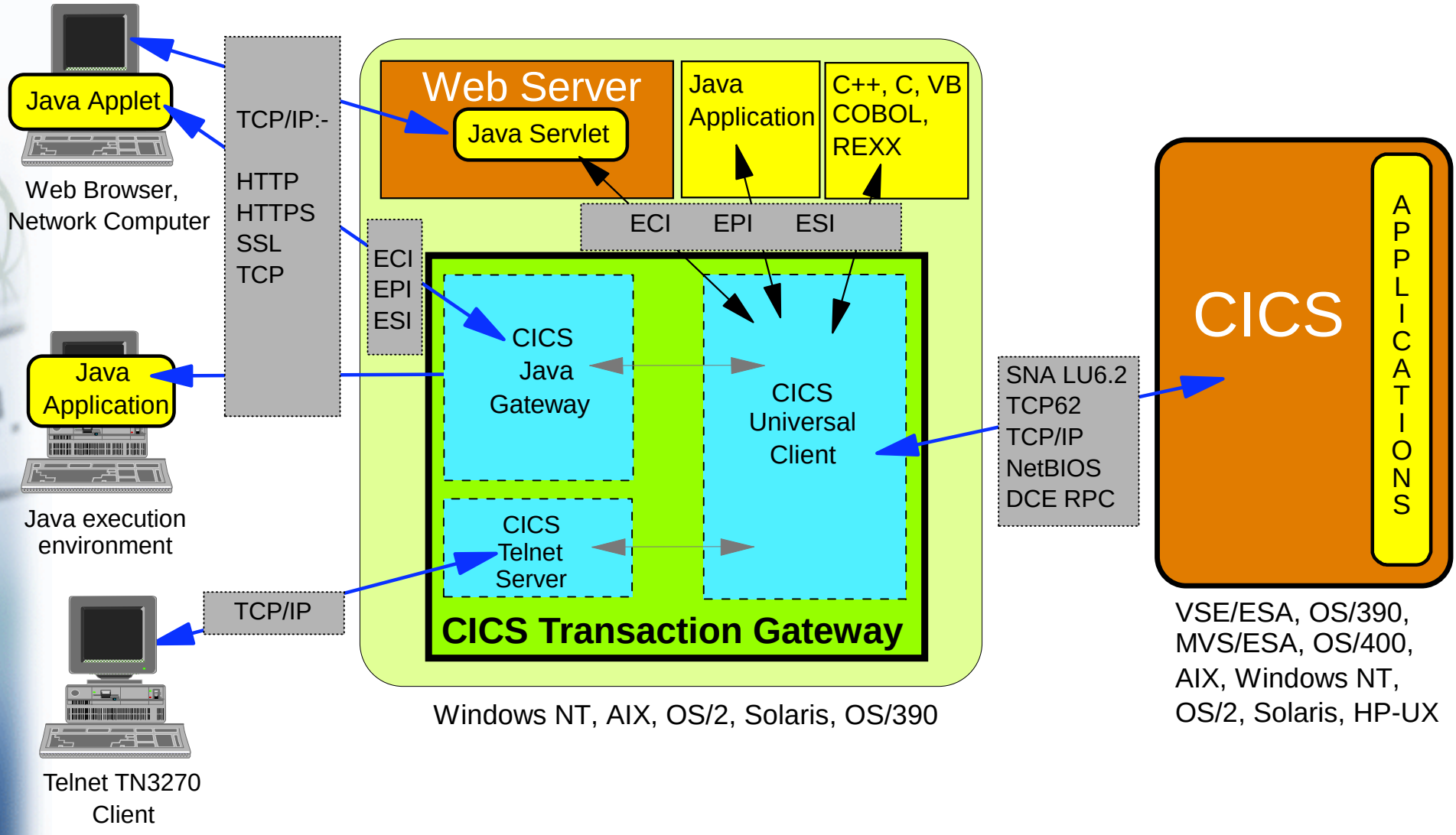
Overview....Relationship to CICS Web Support



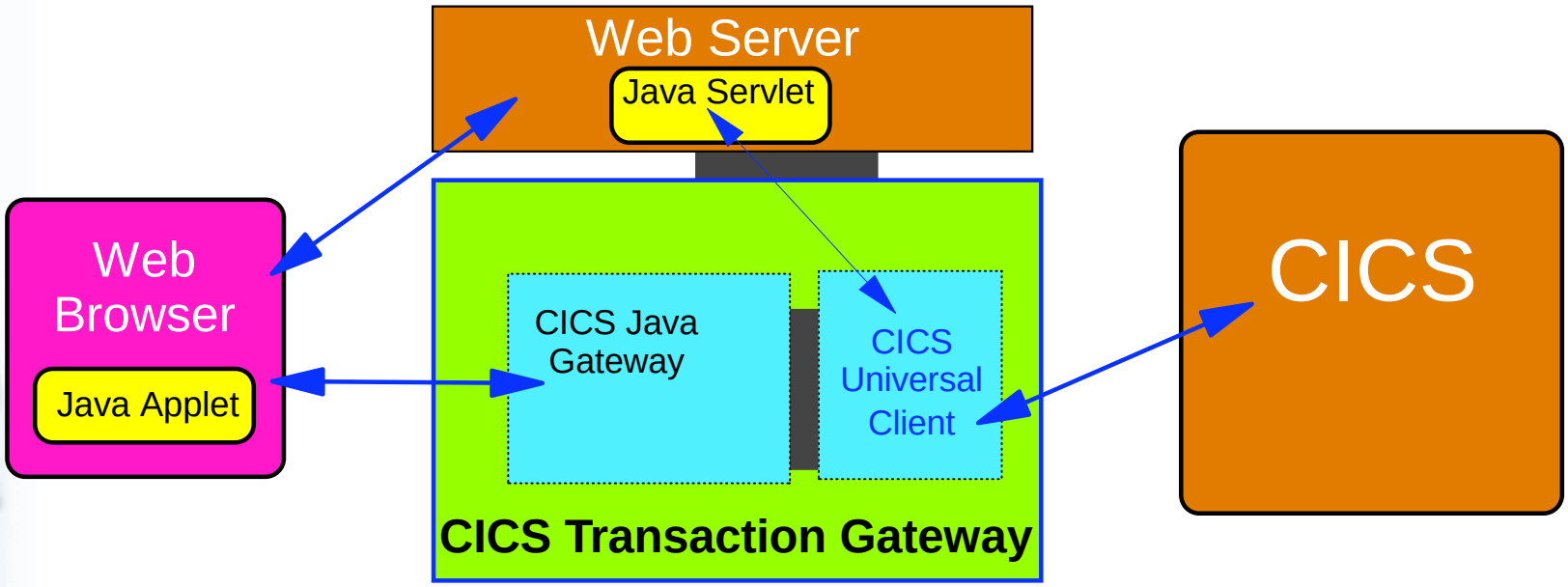
CICS Universal Clients

- The CICS Universal Clients
  - Integrated within the CICS Transaction Gateway
  - Also available separately
- Provides access to CICS systems
  - Client API's
  - Connectivity
- Includes Telnet TN3270 support
- Runs on Windows 95/98, Windows NT, AIX, Solaris, OS/2

The Gateway Structure

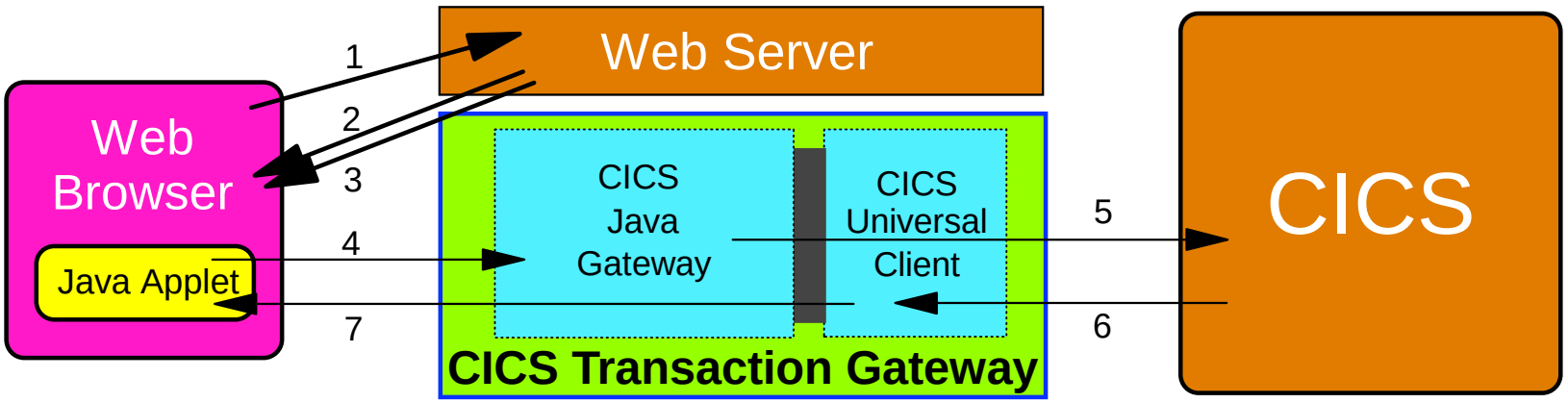


The Gateway Structure....



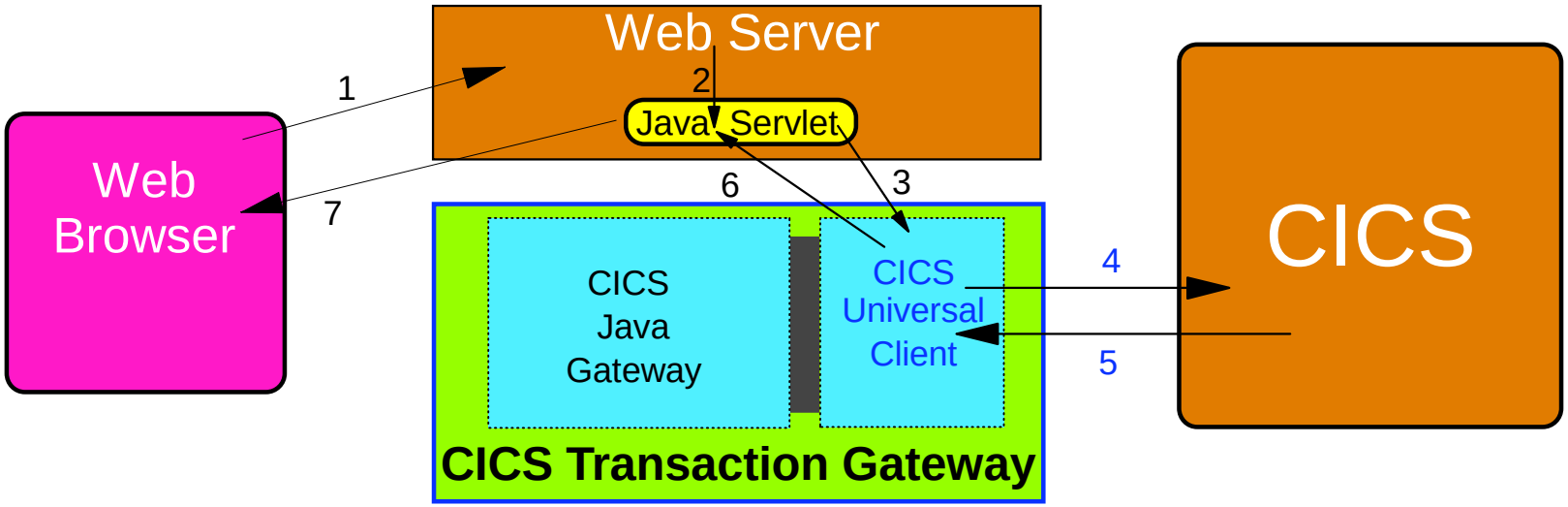
- The CICS Java Gateway component is a Java application
- Applets are Java applications that execute on web browsers
- Servlets are Java applications that execute on web servers

The Gateway Structure....Applet Flows



1. Web browser requests HTML page from the Web-server
2. Web server returns HTML page which identifes applet
3. Web browser downloads applet
4. Applet creates a CICS request and passes it to the Gateway
5. Gateway calls CICS Universal Client to pass request to CICS
6. CICS processes the request and returns result to the Client
7. Gateway gets result from the Client and sends to applet

The Gateway Structure....Servlet flows



1. Web browser requests an HTML page from the Web-server
2. Web server loads servlet identified in HTML page
3. Servlet creates a CICS request and passes to CICS Client
4. CICS Universal Client passes the request to CICS
5. CICS processes request and returns result to the Client
6. Servlet receives result from the Client
7. Servlet formats HTML page and web server sends to browser

The Terminal Servlet

- Provides access to CICS transactions from Web Browsers
- Supplied as part of the CICS Transaction Gateway
- Runs on Web Server
- Invoked via HTML pages
- Sends and receives CICS screen data
- Automatic conversion between HTML and 3270 datastreams
- Allows customisation

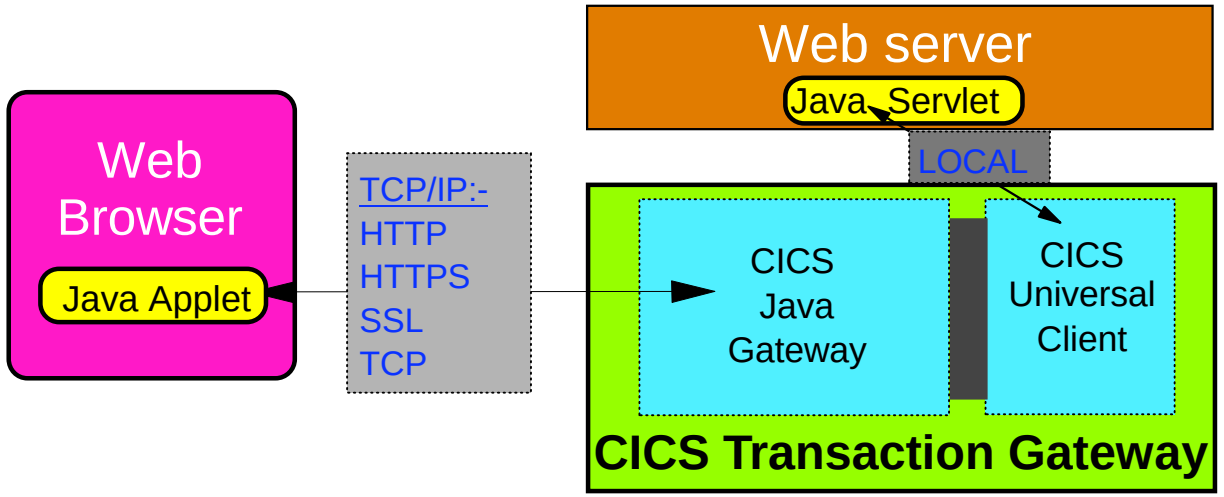
The Terminal Servlet....

- The Terminal Servlet can....
  - Behave like a simple terminal emulator
  - Substitute data from CICS into HTML template files
  - Display CICS screen data in server-side includes
  - Map specific CICS screens to HTML pages
- Can be invoked in three ways....
  - By URL
  - With an HTML FORM
  - With a server-side include

The Terminal Servlet....

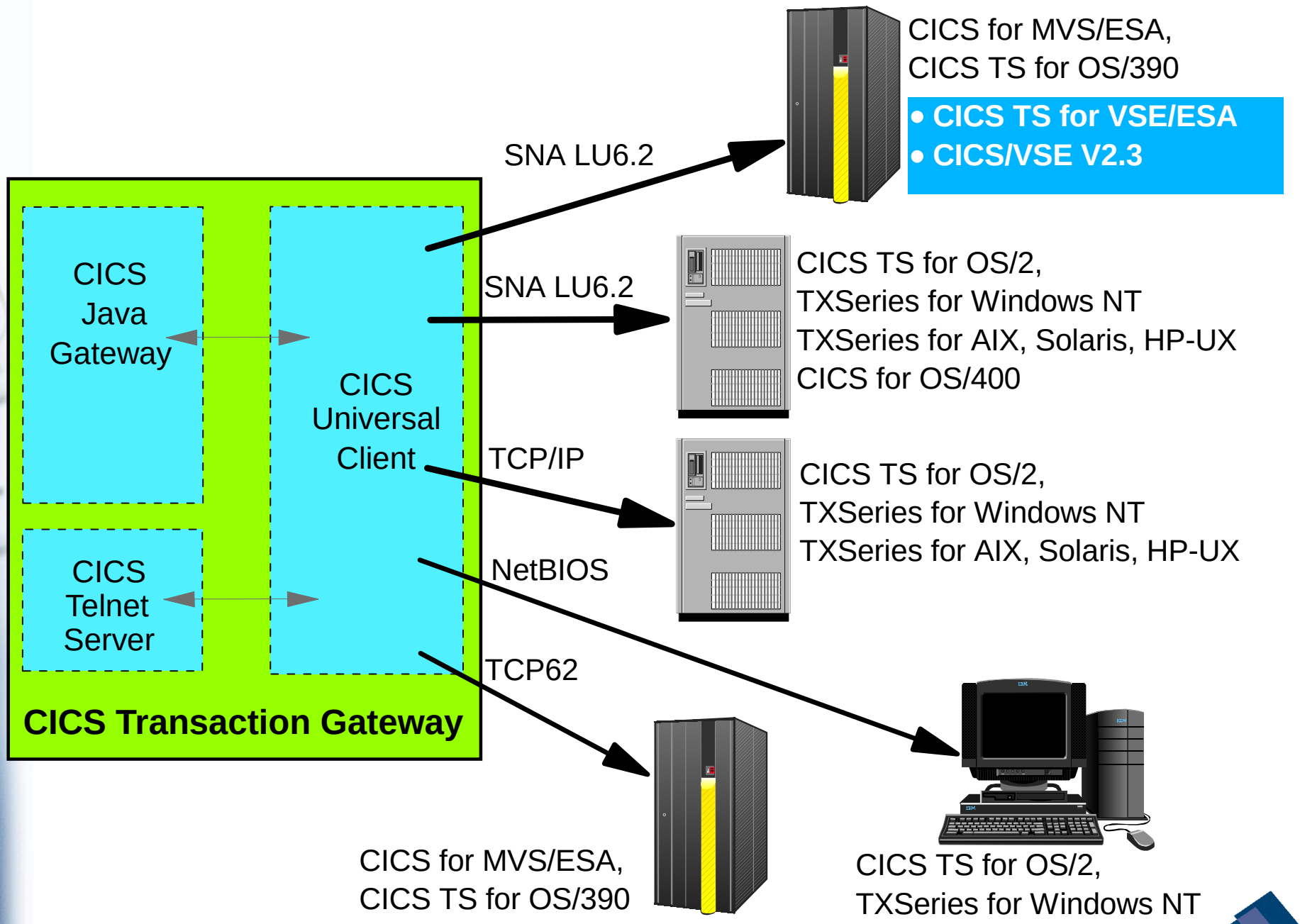
- Invoking the Terminal Servlet with a URL:
  - http://webserver/servlet/TerminalServlet?request=send&transaction=CECI
- Invoking the Terminal Servlet with an HTML FORM:
  - <FORM METHOD="GET" ACTION="/servlet/TerminalServlet">
    <INPUT TYPE="HIDDEN" NAME="REQUEST" VALUE="SEND"
    .....Text entry tags, buttons, etc.....
    </FORM>
- Invoking the Terminal Servlet with a server-side include:
  - <SERVLET NAME="TerminalServlet">
    <PARAM NAME="request" VALUE="send">
    <PARAM NAME="transaction" VALUE="CECI">
    <PARAM NAME="display" VALUE="none">

Network Protocols



- tcp
  - Private persistent connection protocol
- http
  - Standard protocol used for the Web
- ssl
  - Private persistent secure connection protocol
- https
  - Secure protocol used for the web
- local
  - Private protocol used on Gateway machine

Connectivity to CICS



Security

- Between end-user or client application and the Gateway....
  - Via Secure Sockets Layer
- Between the Gateway and CICS on S/390....
  - SNA LU6.2 security
  - CESN Signon transaction

Application Programming Interfaces

- Three API's
  - External Call Interface
  - External Presentation Interface
  - External Security Interface
- Java is the primary language
  - Applets
  - Servlets
  - Applications

NB: CICS Universal Clients interfaces also available on Gateway system

- C++, C, Visual Basic, COBOL, REXX

The External Call Interface

- Usually referred to as the ***ECI***
- Allows invocation of a COMMAREA-based ***application***
- CICS application invoked via
  - Program name
  - Userid and password
  - COMMAREA
- Like a CICS Distributed Program Link
- Calls may be extended to create one logical transaction
- Calls may be synchronous or asynchronous

The External Presentation Interface

- Usually referred to as the ***EPI***
- Provides access to CICS 3270 ***transactions***
- Acts as a logical terminal
- Used to control existing CICS 3270 applications
- No change to CICS application

The External Security Interface

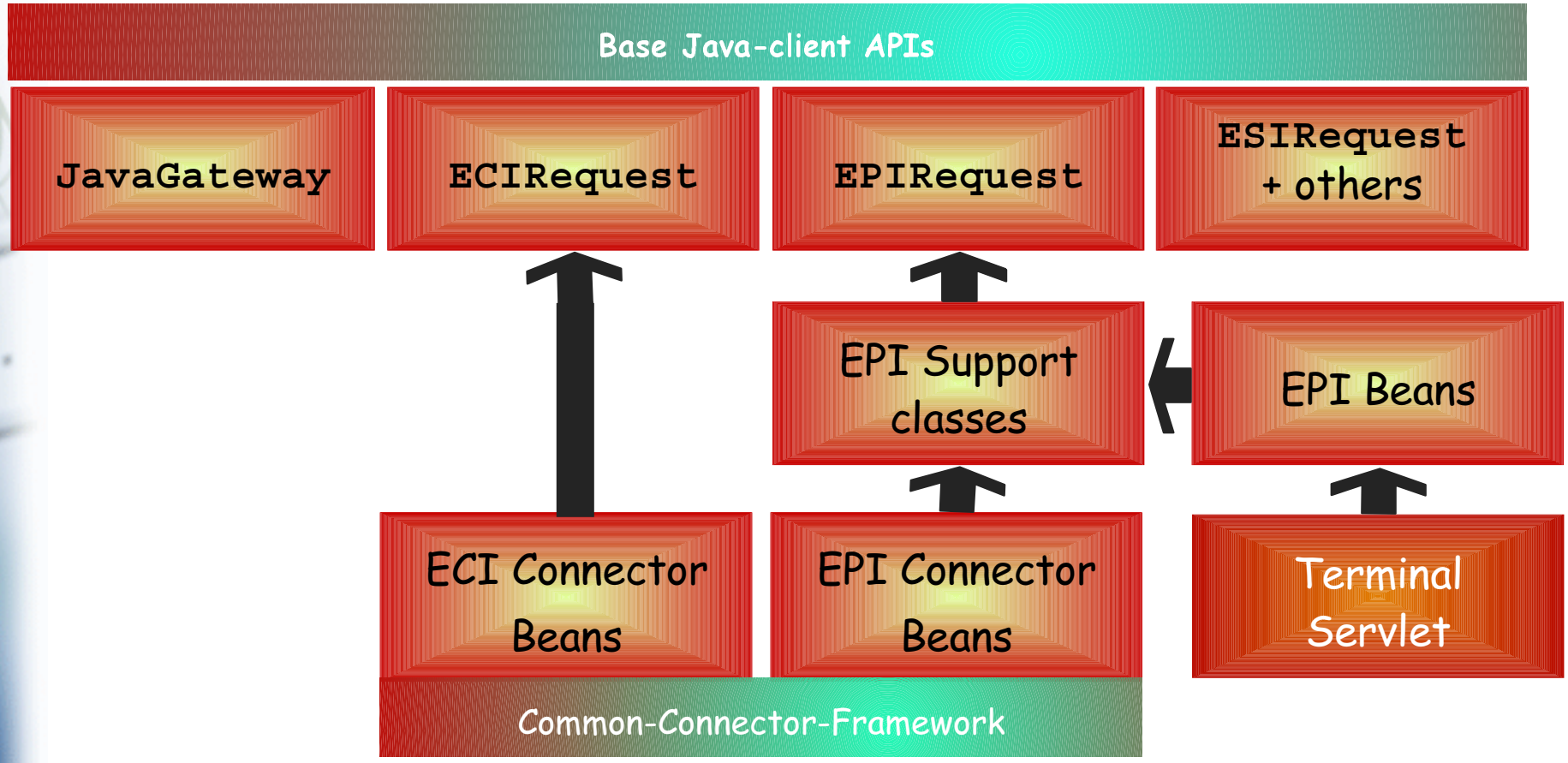
- Usually referred to as the ***ESI***
- Enables use of APPC ***Password Expiry Management (PEM)***
- Passwords can be verified or changed
- Provides audit trail information
- Requires an External Security Manager on S/390

The Java API

- Java class library
  - Contains all Java code and interfaces
- JavaBeans
  - No programming required
  - Use any JavaBean enabled visual application builder
    - e.g. IBM VisualAge for Java, Sun BDK BeanBox
  - JavaBeans supplied with the Gateway....
    - EPI Beans
    - Common Connector Framework (CCF) Beans

The Java API....

- Several layers of Java API supported



Core Java API's

- *JavaGateway* object
  - Represents connection to the CICS Transaction Gateway
  - Has various properties....
    - URL
    - Network address
    - Security classes to be used
  - Core method is *flow*
    - Sends requests to the Gateway
    - Synchronous or asynchronous

Core Java API's....

- ***ECIRequest*** object
  - Encapsulates all types of ECI request
- ***EPIRequest*** object
  - Encapsulates all types of EPI request
- ***ESIRequest*** object
  - Encapsulates all types of ESI request
- ***CicsCpRequest*** object
  - Queries code page in use
- ***Callbackable*** interface
  - Used with asynchronous calls

Java EPI Support Classes

- Based on C++ EPI classes in CICS Universal Client
- Hides programmer from 3270 datastreams
- Terminal class handles all interactions with CICS
- Terminal has associated ***Screen*** instance....
  - Contains a number of ***Fields***
    - accessed by index or screen position
- for BMS screens a ***Map*** class can be generated from BMS source and then fields accessed by name
- BMS Maps created using supplied utility

Simple Java EPI application

```
import com.ibm.ctg.client.*;                          // Need the client-side classes
import com.ibm.ctg.epi.*;                             // And the enhanced EPI classes
public class EPIAndy
{
   public static void main (String [ ] astrArgs)    //  Invoke program using :
   {                                                 //    java EPIAndy <Gateway_URL> <CICS_Server>
      try {
         JavaGateway jgate = new JavaGateway();              //Create a default JavaGateway
         jgate.setURL(astrArgs[0]);                          // Set URL of remote Gateway
         jgate.open();                                       // Open the connection

         Terminal terminal = new Terminal(jgate, astrArgs[1], null, null); // Add a terminal
         terminal.send(null, "CESN", null);                  // Start CESN on the terminal

         Screen screen = terminal.getScreen();               // Get the current screen
         for (int i = 1; i <= screen.fieldCount(); i++)      // Loop round all fields
         {
            if (screen.field(i).textLength() > 0) {          // Print non-empty fields
               System.out.println("Field " + i + ": " + screen.field(i).getText());
            }
         }

         screen.setAID(AID.PF3);                             // Set the AID key to send
         terminal.send();                                    // Return the screen to CICS
         terminal.disconnect();                              // Disconnect the terminal
      }
      catch (Exception e) {                                  // Handle any problems
         System.out.println(e.getMessage());
      }
   }
}
```

EPI JavaBeans

- The EPI Beans are ....
  - Built on top of the EPI Support classes
  - Fully compliant with Sun's JavaBeans API
- Use to quickly create front-ends that connect to CICS
- Four EPI Beans supplied

EPI JavaBeans....

- The *EPITerminal* bean
  - Acts as a 3270 terminal connected to CICS
  - Handles all interactions with CICS
- The *EPIBasicScreenHandler* bean
  - Simple default ScreenHandler
- The *EPIScreenButtons* bean
  - Displays and handles set of visual "buttons"
- The *EPIMonitor* bean
  - Mechanism to display basic terminal status
- Specific *ScreenHandler* beans
  - Can be created for specific BMS maps

The Common Connector Framework

- The IBM ***Common-Connector-Framework (CCF)*** provides a consistent means of interacting with Enterprise resources from any Java execution environment
- Consistent for User applications
  - Consistent CCF Client view, whatever the resource
- Consistent for the IBM Connectors
  - Consistent CCF Infrastructure view, whatever the runtime

The Common Connector Framework....

- The CCF Client API....
  1. **ConnectionSpec**
     - ➡ Contains properties to access an Enterprise resource
  2. ***InteractionSpec***
     - ➡ Contains properties for a single interaction
  3. ***Communication***
     - ➡ The conduit used to execute interactions
- The CICS Connectors....
  - ➤ CICSConnectionSpec
  - ➤ ECIInteractionSpec
  - ➤ EPIInteractionSpec

The Common Connector Framework....

- Expected that most people will use generator tools for CCF
- Enterprise Access Builder (EAB), part of VisualAge for Java Enterprise Edition V3, supplies CCF connector beans....
  - CICS, MQSeries, IMS, Encina
  - Host-on-Demand
  - SAP R/3
- New CCF based connectors planned for VSE/ESA V2.5
  - Access to VSE resources

Further Information

- CICS Website for general information and publications:
  - http://www.ibm.com/software/ts/cics/
- Red Books
  - Revealed! CICS Transaction Gateway with More CICS Clients Unmasked, SG24-5277
  - Revealed! Architecting Web Access to CICS, SG24-5466
  - Books downloadable from http://www.redbooks.ibm.com
- Samples provided with the Gateway....
  - Use of ECI, EPI, ESI
  - Use of EPI Beans with VisualAge for Java
  - Use of the Terminal Servlet

CICS Transaction Gateway Summary

- Enables access to CICS applications and transactions from:
  - Web Browsers or Network Computers
  - Java Applets, Servlets, Applications
- Provides the ECI, EPI and ESI programming interfaces
- Allows visual programming using supplied JavaBeans
- Terminal Servlet provides 3270 application access
- Supports the cross-product Common Connector Framework
- Provides network security via industry standard SSL
- Provided as part of the CICS TS for VSE/ESA package
- Supports CICS/VSE V2.3 as well as CICS TS for VSE/ESA