Access to CICS with CICS Transaction Gateway

Phil Wakelin   CICS Transaction Gateway Technical Planner   Phil_Wakelin@uk.ibm.com

Hursley Lab, IBM United Kingdom Limited

CICS Transaction Gateway 7.2
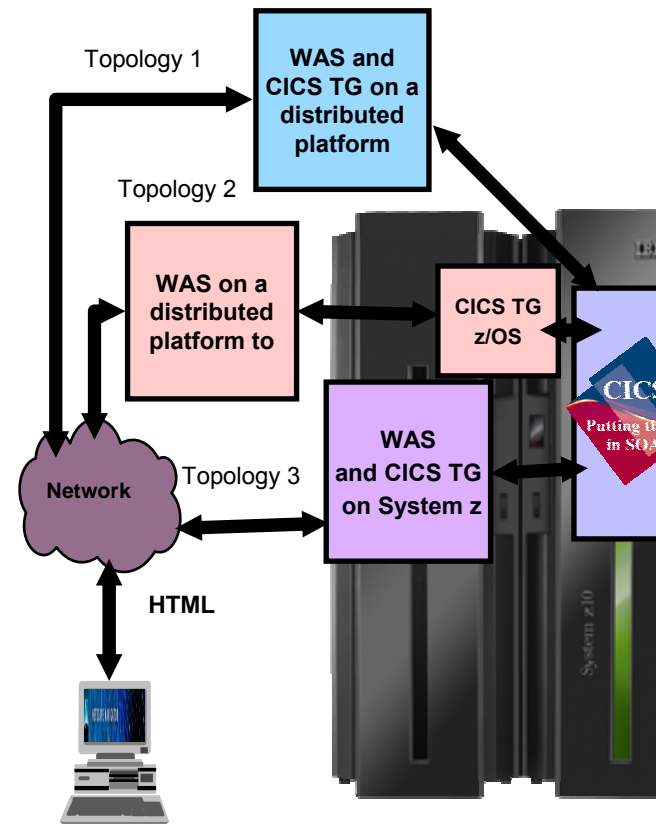## Secure, Scalable and Flexible SOA Access to CICS

CICS Transaction Gateway is IBM's strategic connector from WebSphere to CICS, production proven in over a thousand customer environments

Key Characteristics:

- High performing, security-rich and scalable

- Standard JCA interface is strategic and provides best Qualifies of Service

- Minimal or no changes required for CICS applications to leverage value in an SOA

CICS Transaction Gateway 7.2

- Improved business flexibility, delivered through transactional support across a balanced CICS environment

- New CICS connectivity for remote clients in C runtime, enabled for exploitation within .NET environment

- Enhanced systems monitoring API, providing for integration with CICS Explorer

Topology 1

WAS and CICS TG on a distributed platform

Topology 2

WAS on a distributed platform to

CICS TG z/OS

CICS

Putting in SOA

Network

Topology 3

WAS and CICS TG on System z

HTML

System z10

CICS Transaction Gateway - products

CICS Transaction Gateway for Multiplatform v7.2
Priced per processor
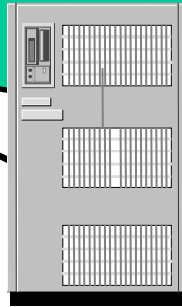**Multi-user server gateway for remote clients**
Includes CICS UC technology
Complements WebSphere App Server
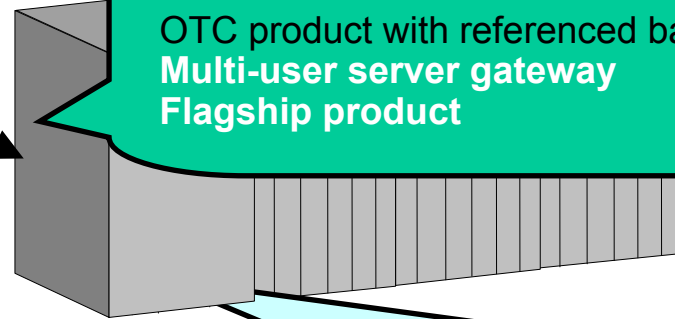Win32, AIX, Solaris, HP-UX, Linux on Intel/zSeries/POWER

Web Browser

Web App Server

CICS TG v7.2 for z/OS
OTC product with referenced based pricing
Multi-user server gateway
Flagship product

Desktop PC

CICS UC v7.1 for Windows/Linux/UNIX

Remote clients:

Java clients

J2EE applications (JCA)

C applications, .NET

Supported CICS servers
CICS TS for z/OS v2.3, v3.1, v3.2
CICS TS for VSE v1.1.1, CICS/VSE v2.3
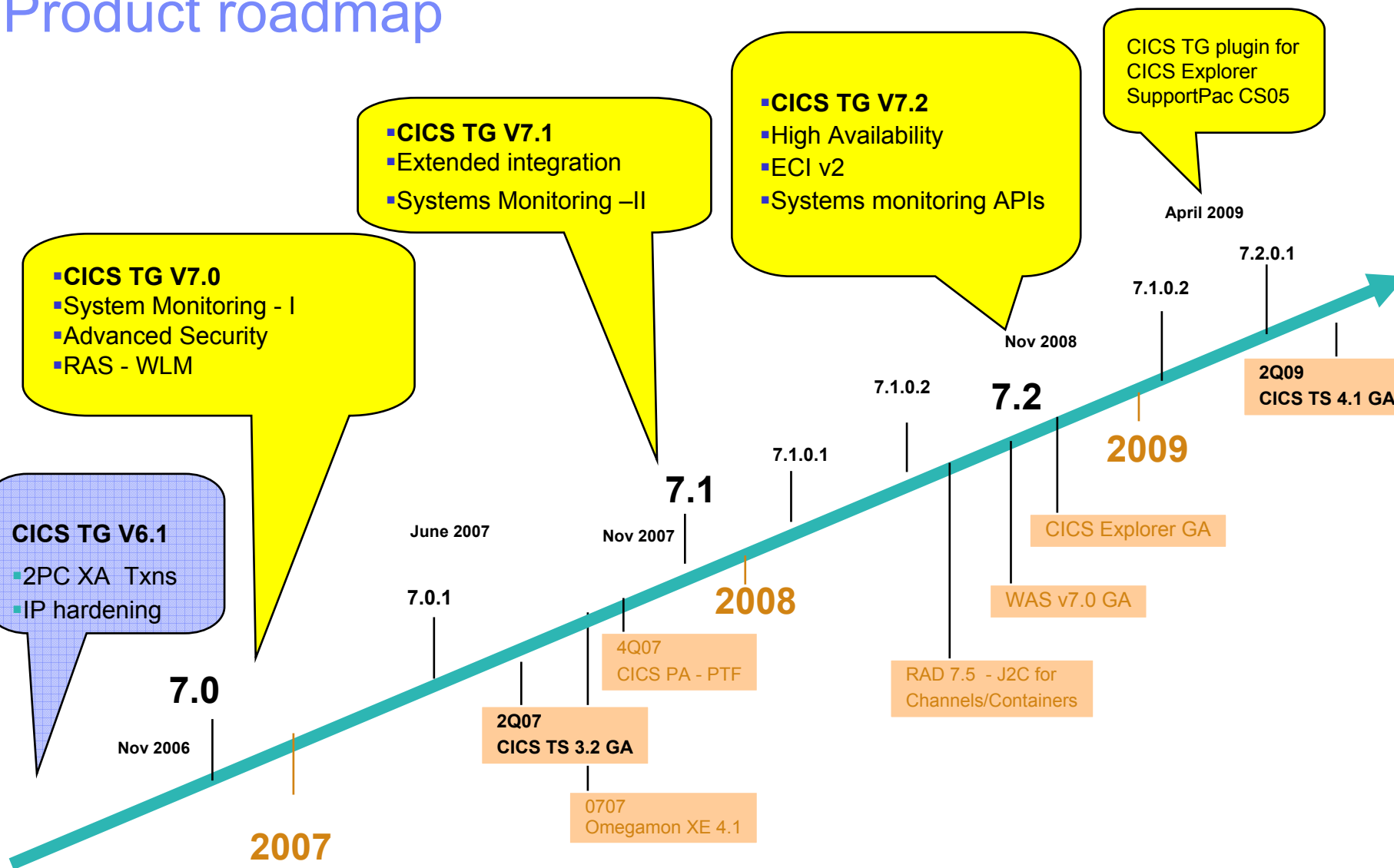TxSeries v5.1, v6.0, v6.1, v6.2, v7.1
CICS TS for iSeries (AS/400) v5.3, V5.4, V6.1

Why choose CICS TG on z/OS?

- **High Availability –** *Maximize confidence in system uptime*
  - Unique capabilities to exploit Parallel Sysplex
  - Support for TCP/IP Port sharing, Sysplex distributor, WLM, and RRS (for two-phase commit support)

- **High scalability/Performance –** *Superior performance and scalability on system z*
  - Benchmarked performance figures of 6,000 tps per Gateway with min. payload and >1,000 tps with a full 32KB payload
  - Cloning capability provides high level scaling across the sysplex

- **Advanced Systems Management** – *Improve capacity planning and system optimization*
  - SMF recording of statistics, enabling integration in CICS Performance Analyzer
  - OMEGAMON monitoring solution

- **Transaction support –** *Enterprise wide application integration*
  - XA (two-phase commit) support for EXCI and IPIC based requests
  - Integration with our HA support when cloning Gateways in a Sysplex

- **Advanced security options –** *Utilize system z security capabilities*
  - Asserted identity support is provided for EXCI and IPIC protocols
  - Pass tickets and Tivoli Federated Identity Manager integration options
  - Thread identity support for WAS on z/OS

Product roadmap

CICS TG plugin for CICS Explorer SupportPac CS05

April 2009

CICS TG V7.2
- High Availability
- ECI v2
- Systems monitoring APIs

Nov 2008

CICS TG V7.1
- Extended integration
- Systems Monitoring –II

CICS TG V7.0
- System Monitoring - I
- Advanced Security
- RAS - WLM

CICS TG V6.1
- 2PC XA Txns
- IP hardening

7.2.0.1

7.1.0.2

7.2

7.1.0.2

7.1.0.1

2009

CICS Explorer GA

7.1

WAS v7.0 GA

Nov 2007

2Q09 CICS TS 4.1 GA

June 2007

RAD 7.5 - J2C for Channels/Containers

7.0.1

2008

4Q07
CICS PA - PTF

7.0

2Q07 CICS TS 3.2 GA

Nov 2006

0707
Omegamon XE 4.1

2007

© 2008 IBM Corporation

CICS Transaction Gateway V7.0

Systems Monitoring

- **Real time monitoring** of CICS TG provides the ability to analyse system utilisation metrics and perform online problem determination.
- Access to key statistics about Gateway daemon, CICS Status, connections, threads and protocol handlers
- **Console message support** enabling automation via Tivoli SA for z/OS

Extended Networking

- On z/OS, **WLM support** improves distribution of IP connections reducing the likelihood of any storm-drain style scenarios
- **IPv6** support for remote Java clients can provide better routing, enhanced security, and global scalability
- **Extensions to SNA** support including the Linux and Sun Solaris environments, allowing migration away from TCP62 environments
- **Server idle-timeout** reduces network overheads

Advanced Security

- **TLS security standard** enables more stringent SSL encryption capabilities and better interoperation with a variety of secure clients.
- Further integration with **RACF** and System z **cryptographic hardware** allows for higher levels of security and increased throughput of security requests
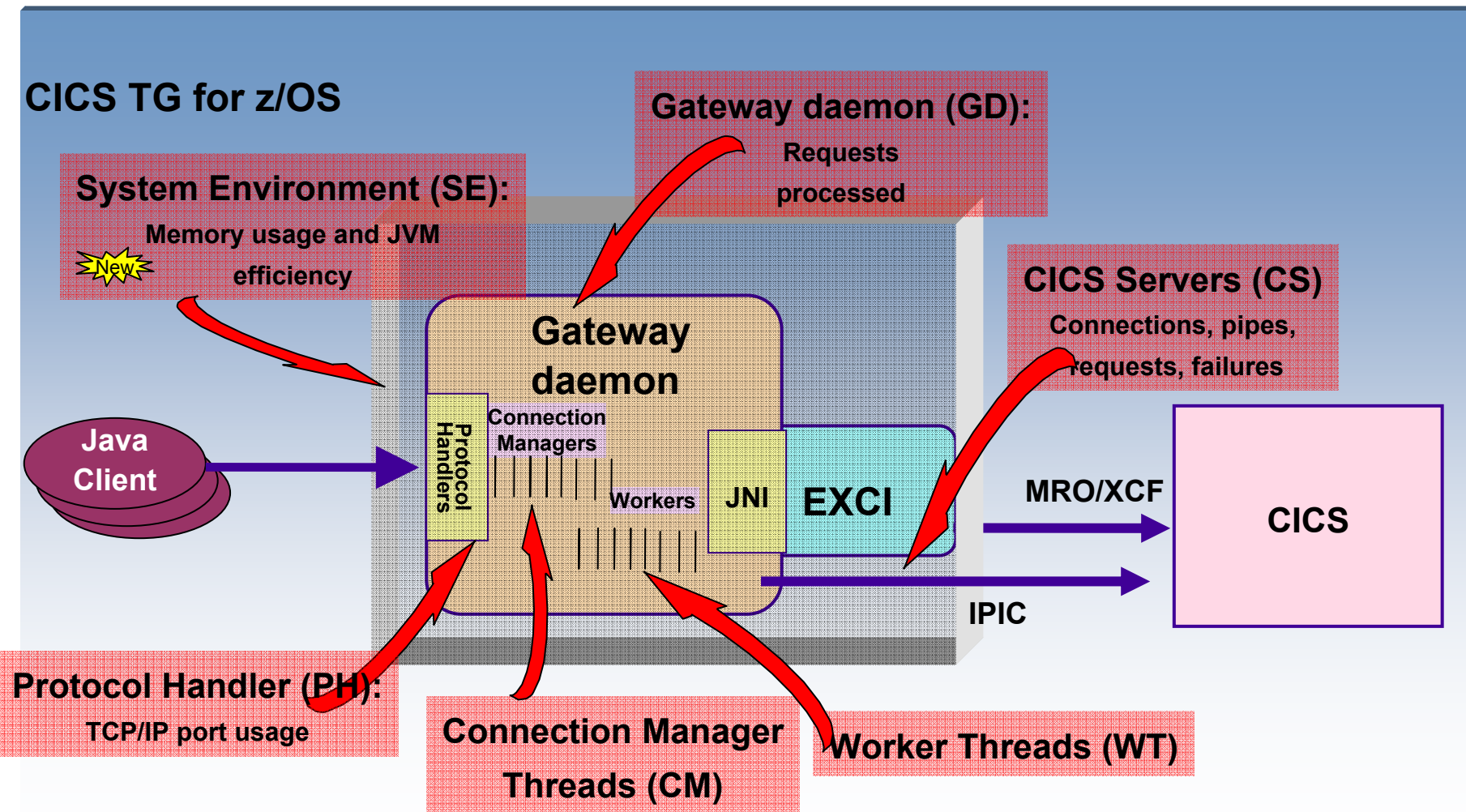
Systems Monitoring – A window into the black box

CICS TG for z/OS

Gateway daemon (GD):
Requests processed

System Environment (SE):
Memory usage and JVM efficiency

New

CICS Servers (CS)
Connections, pipes, requests, failures

Java Client

Gateway daemon

Protocol Handlers

Connection Managers

Workers

JNI

EXCI

MRO/XCF

IPIC

CICS

Protocol Handler (PH):
TCP/IP port usage

Connection Manager Threads (CM)

Worker Threads (WT)

CICS Transaction Gateway V7.1

Advanced Systems Monitoring

- Extended **real time monitoring** of CICS TG provides advanced capacity planning and problem determination facilities
- **Interval based statistics**, and off-line recording to **SMF**, and CICS PA support, provides for off-line monitoring and trend analysis
- **Request monitoring exits** provide simple and efficient infrastructure for advanced problem determination

Extended Integration

- Support of CICS TS V3.2 IPIC connectivity provides:
  - Exchange of large data areas by **containers/channels**
  - Simplified topologies for **SSL** and **XA** connectivity
- Improved update of **health** to WLM reducing liklehood of storm-drain style scenarios

Interoperability

- **64-bit operating system** toleration for Windows and Linux
- **Extensions to SNA** support assisting migration from TCP62 environments to Enterprise Extender
- Support of time change protocols

CICS TS V7.1 - Request monitoring

> **Aim**:
> Advanced problem determination for request tracking and response time analysis, of ECI requests as they flow the Java client into CICS applications and back
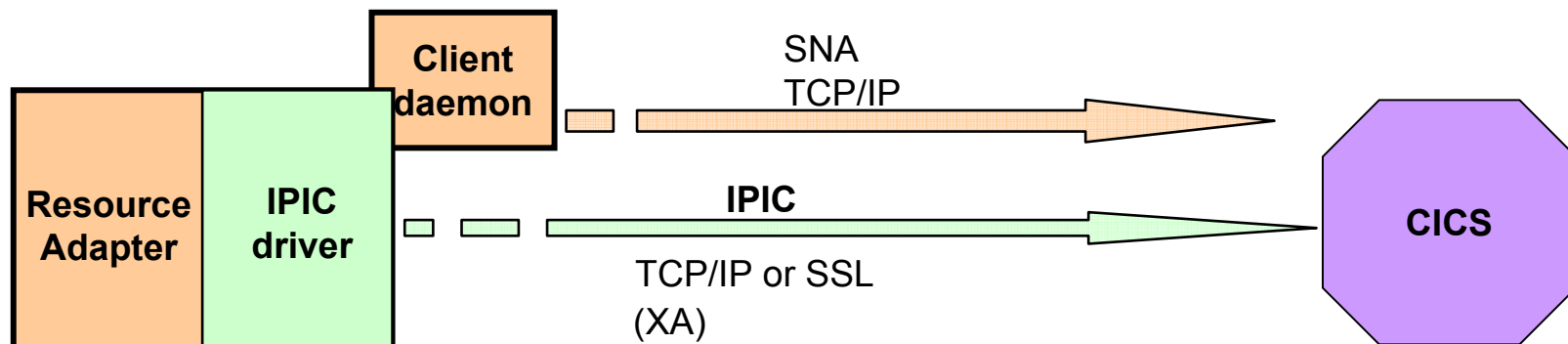

J2EE Application Server

Resource Adapter

ECI request

Gateway daemon

CICS Region

CICS Region

z/OS LPAR

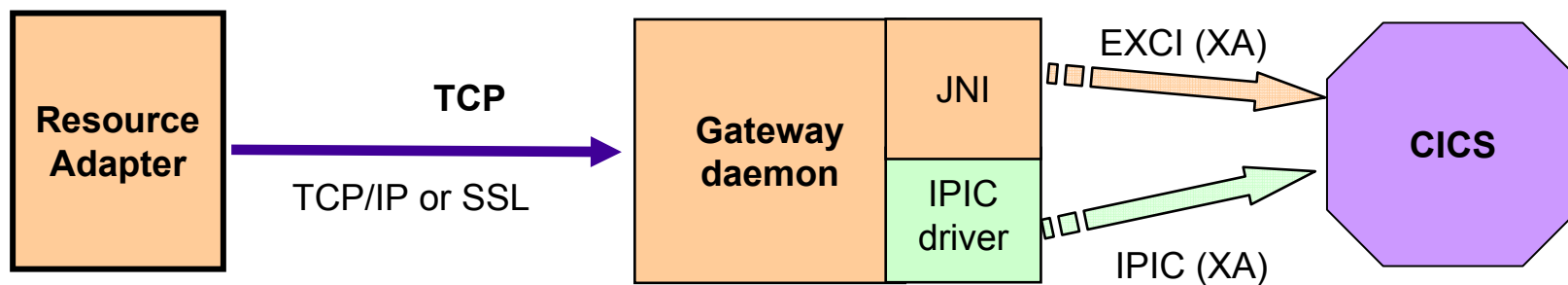Exit points

xit based infrastructure

- Request level transaction analysis and response time breakdown within the CICS TG
- Common set of exit points for all topologies
- Functions in Java client and Gateway daemon, and in local and remote modes, and on all platforms
- SupportPac CH51 provides *SimpleMonitor* with dynamic control of logging levels
- Vendor tools - further exploit the infrastructure
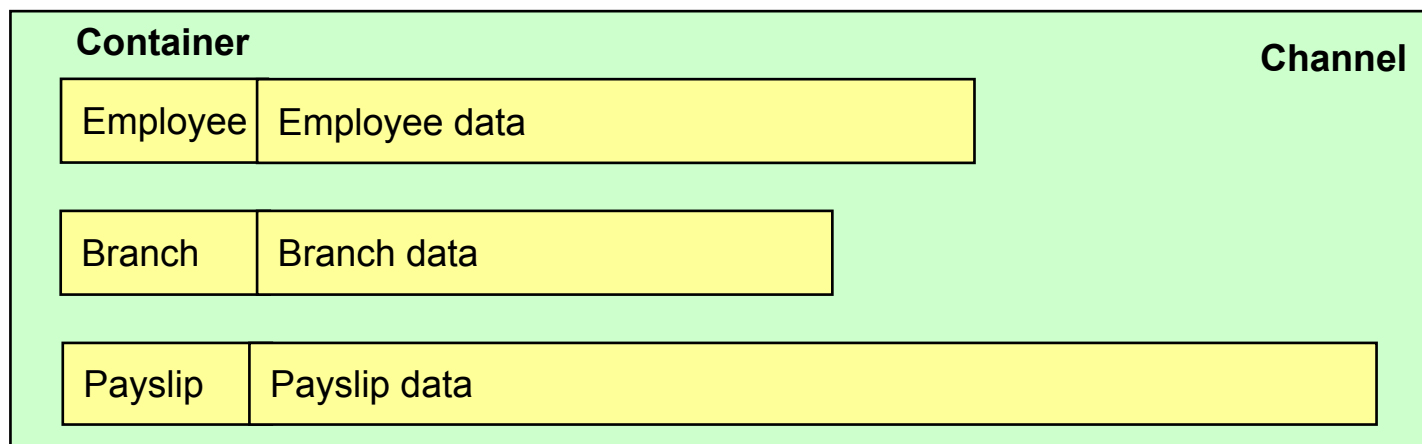
CICS TS V7.1 - IPIC

1. Local mode (2-tier)

```
┌─────────────────────────────┐
│ Resource │ IPIC          │──┐  Client daemon
│ Adapter  │ driver        │  │
└─────────────────────────────┘
```

Client daemon

SNA
TCP/IP → CICS

IPIC

TCP/IP or SSL
(XA) → CICS

2. Remote mode (3-tier)

Resource Adapter

TCP

TCP/IP or SSL →

Gateway daemon — JNI / IPIC driver

EXCI (XA) → CICS

IPIC (XA) → CICS

CICS TS V7.1 - Channels and Containers

- "Modern day COMMAREAs" >32K

- Recommended usage is to have more containers with less data inside each

- Containers:
  - One of many in a channel
  - No CICS enforced size limitation
    - Containers are stored above the bar (64 bit storage) in CICS TS v3.2
  - Are either binary (BIT) or of a specified codepage/ccsid (CHAR)
  - ccsid is numeric value (37) equivalent to a code page string (IBM-037) and is used as internal meta-data for each container

- CICS TG container data built in default JVM encoding (i.e UTF-8)
  - CICS converts received data to EBCDIC when GET CONTAINER is issued
  - CICS converts response data to code page of the channel when PUT CONTAINER is issued

Container   **Channel**

| Employee | Employee data |

| Branch | Branch data |

| Payslip | Payslip data |

CICS Transaction Gateway V7.2
Delivers major enhancements in three key value areas

High Availability

- **Default server selection** – Simple mechanism to default server
- **Server name remapping** Static definitions and user exit for CICS server name remapping – CICS TG z/OS only
- **Sysplex XA recovery** – Gateway group infrastructure expanded to support peer recovery across the sysplex – CICS TG z/OS only

New Remote connectivity

- **Remote C clients**
  - **Simple but powerful function to extend CICS integration**
  - **ECI v2 API with commarea access**
  - **.NET support**
  - **Migration path for CICS Universal Client**
  - **Redistributable zip file**

CICS Explorer Integration

- **Remote statistics API support**
- **Java statistics API**

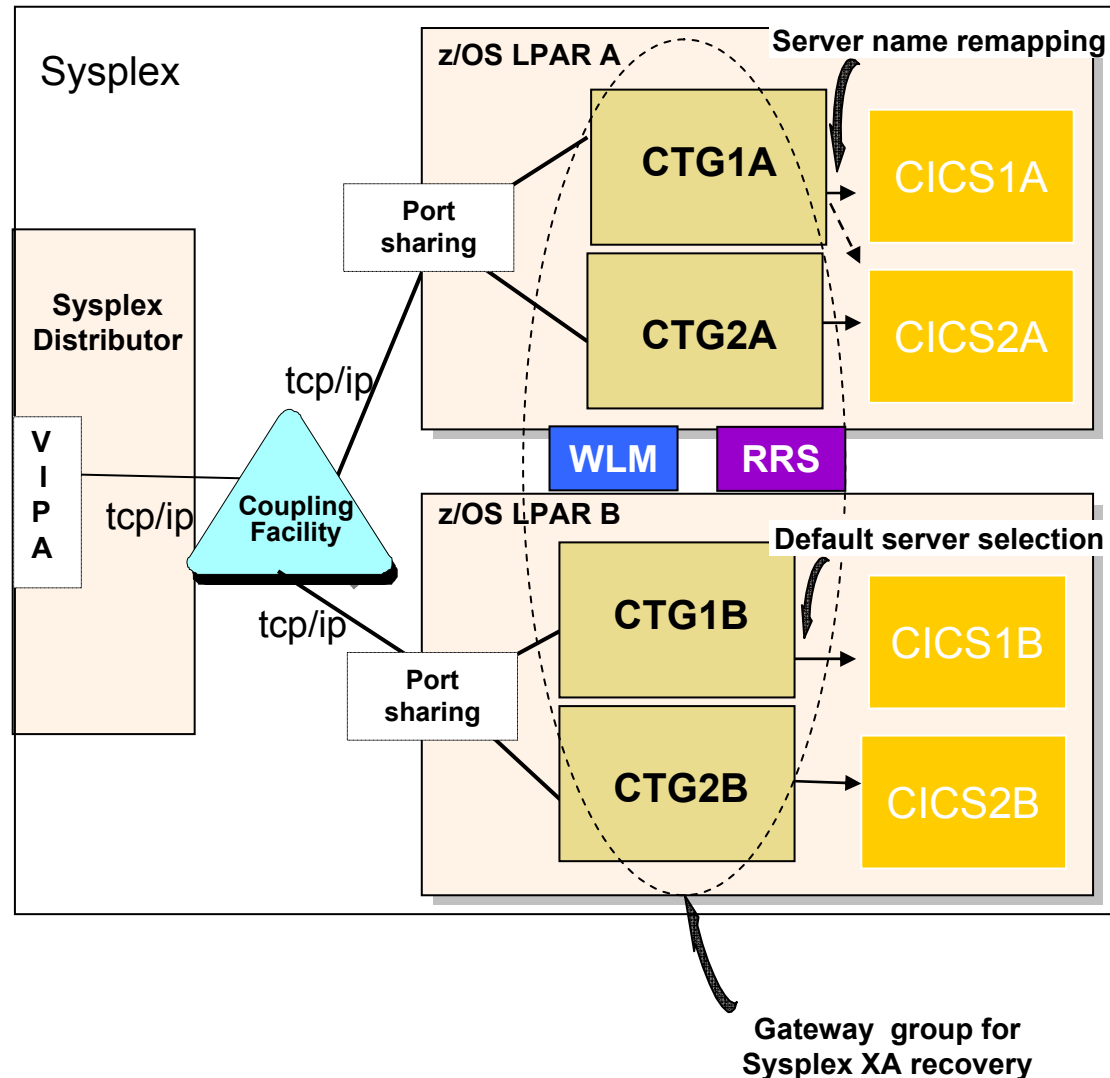# CICS TG V7.2 – New High Availability Functions

Default server
- Supported on all platforms
- Replacement for DFHJVSYSTEM_00 supporting IPIC and EXCI servers
- Supported for synconreturn and extended ECI requests (not XA/2pc)
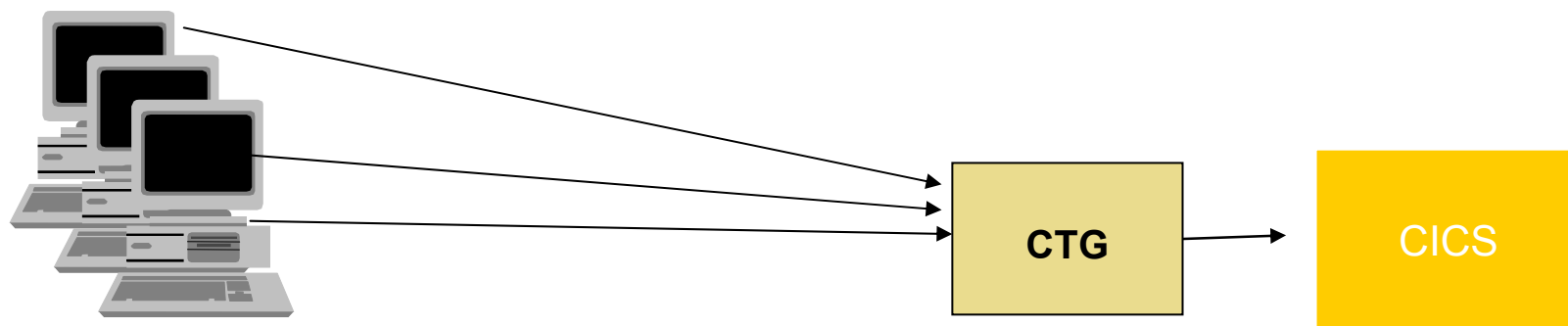
Server name remapping (z/OS only)
- Redirects ECI requests to a local CICS server when using IP load balancing
- Supported for synconreturn and extended ECI requests (not XA/2pc)
- Replacement for DFHXCURM supporting IPIC and EXCI servers
- Two supported options:
  1. Logical server definitions
  2. CICS request exit SupportPac CA1T - RoundRobin and Failover exits using HFS file

Sysplex XA recovery (z/OS only)
- Provides ability to build HA configuration when using XA 2pc transaction from WAS
- Gateway cloning supported across the sysplex
- Gateway group defined using common CICS TG APPLID qualifier

Sysplex

Server name remapping

z/OS LPAR A

CTG1A → CICS1A

Port sharing

CTG2A → CICS2A

Sysplex Distributor

VIPA

tcp/ip

Coupling Facility

tcp/ip

WLM   RRS

z/OS LPAR B

Default server selection

CTG1B → CICS1B

Port sharing

CTG2B → CICS2B

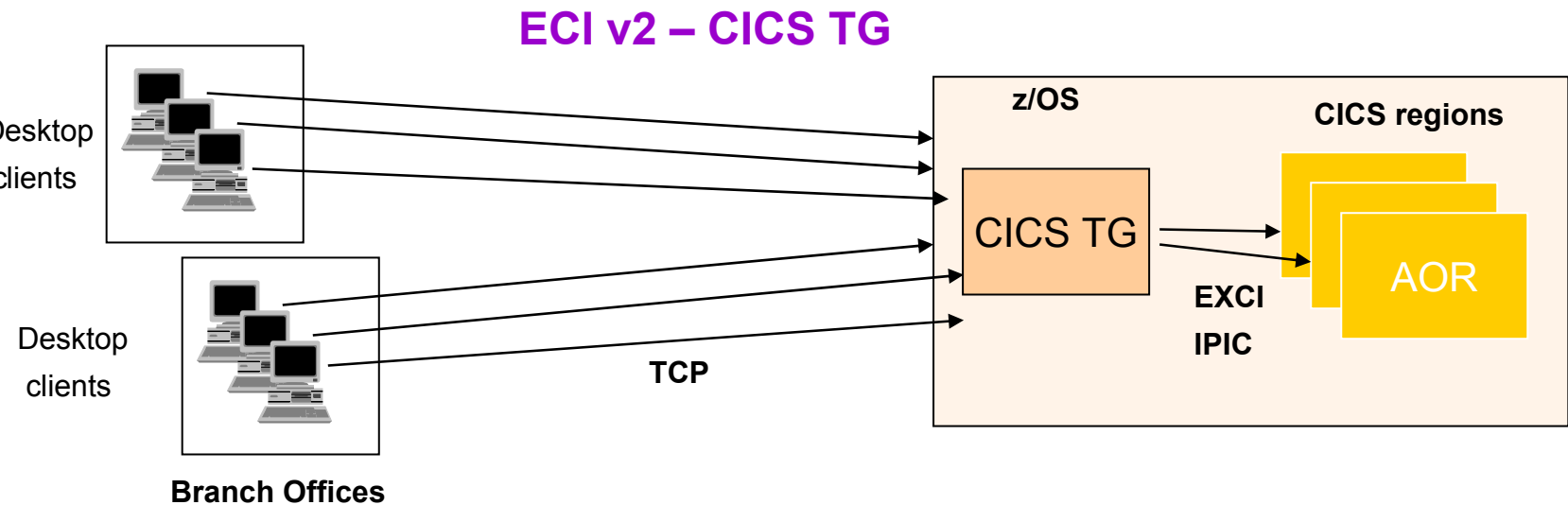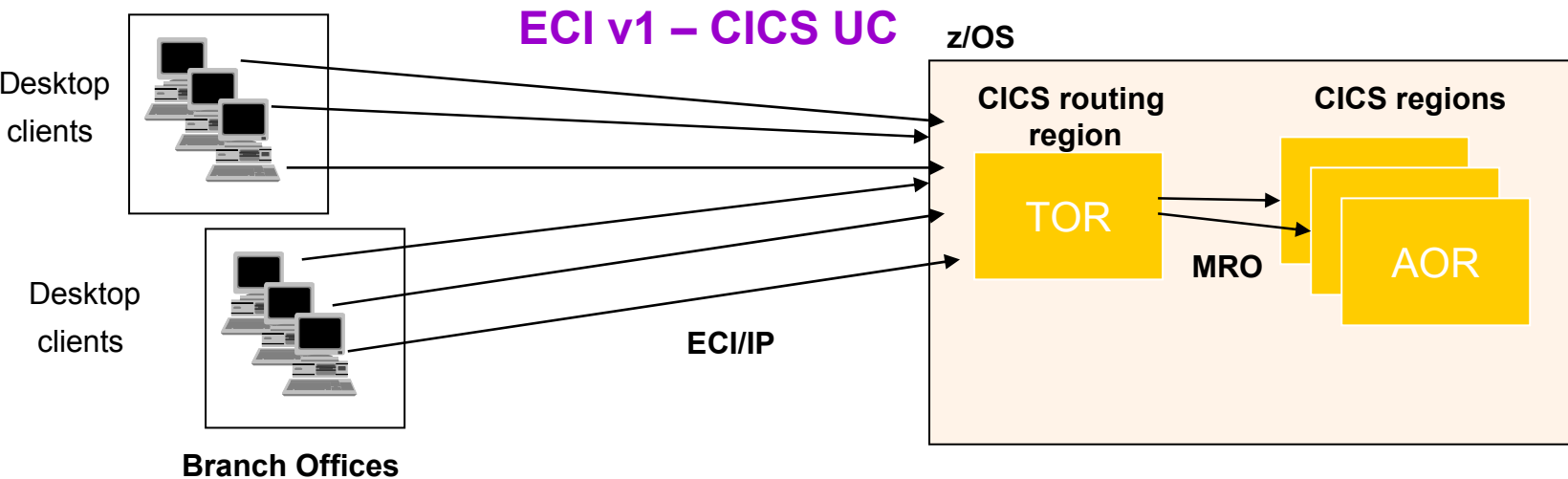**Gateway group for Sysplex XA recovery**
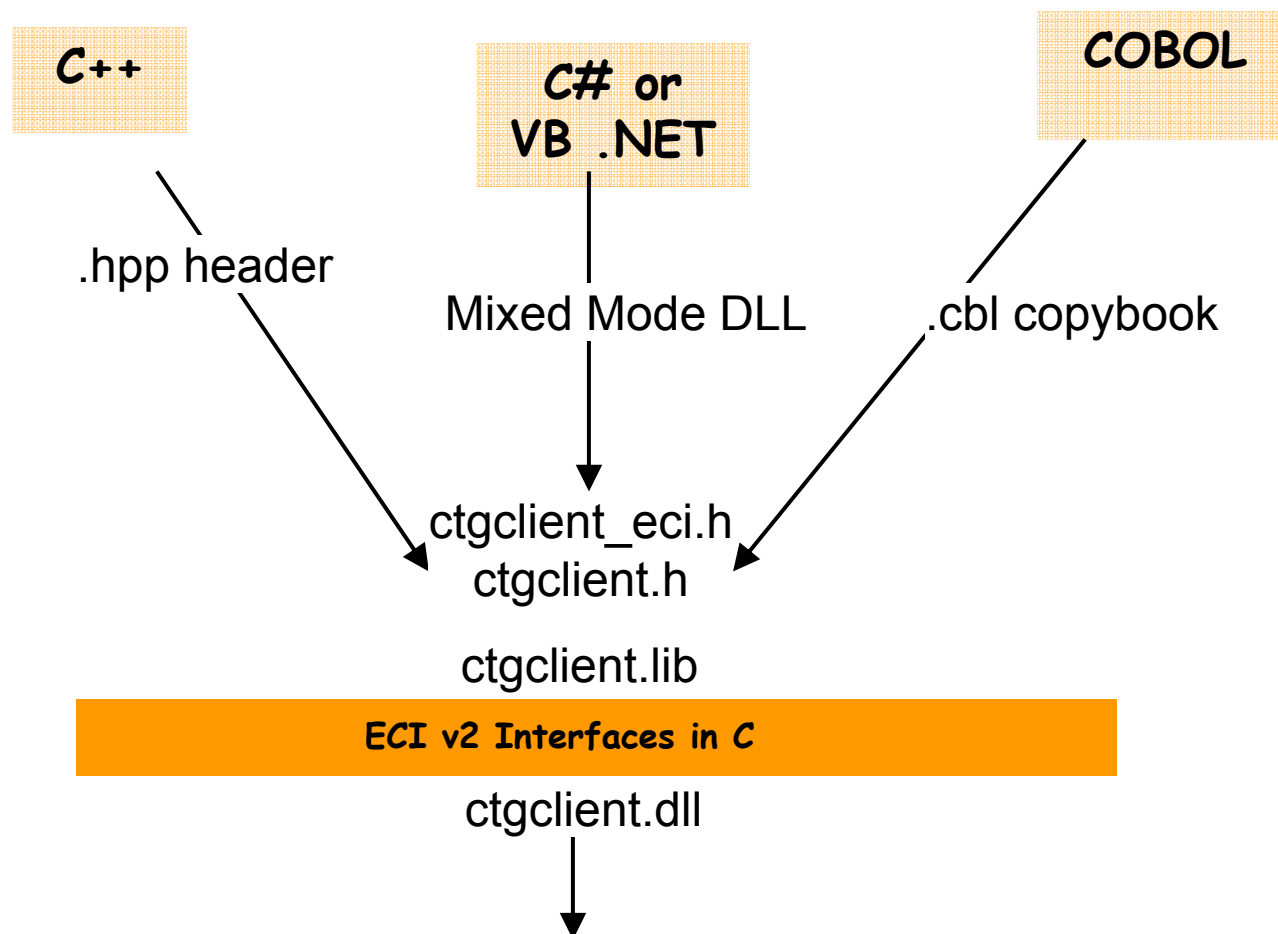
ECI v2 - Remote C client API

- ECI v2 C language bindings
  - AIX, Linux, HP-UX, Solaris, Windows client platform support (ctgredist.zip file)
  - Doxygen HTML based programming reference
  - Usage from COBOL and .NET environments
  - CA73 SupportPac provides Supported .NET API

- Migration path from CICS Universal Client
  - Lightweight client footprint
  - Simple code changes from ECI v1

- Exploit CICS TG QoS
  - Performance
  - High availability
  - Security
  - One phase commit

ECI v2 - Client migration from CICS UC

ECI v1 – CICS UC

z/OS

CICS routing region

CICS regions

TOR

AOR

MRO

ECI/IP

Desktop clients

Desktop clients

Branch Offices

ECI v2 – CICS TG

z/OS

CICS regions

CICS TG

AOR

EXCI

IPIC

TCP

Desktop clients

Desktop clients

Branch Offices

COBOL and .NET usage

C++

C# or
VB .NET

COBOL

.hpp header

Mixed Mode DLL

.cbl copybook

ctgclient_eci.h
ctgclient.h

ctgclient.lib

ECI v2 Interfaces in C

ctgclient.dll

ECI v2 - C language bindings

- **Synchronous API (no async)**
  - CTG_ECI_PARMS.eci_call_type
    - ECI_SYNC
    - ECI_SYNC_TPN

- **Extended LUWs (1pc)**
  - CTG_ECI_PARMS.eci_extend_mode
    - ECI_NO_EXTEND
    - ECI_EXTENDED
    - ECI_COMMIT
    - ECI_BACKOUT

- **Request timeout**
  - CTG_ECI_PARMS.eci_timeout

- **Data transmission optimisations**
  - Trailing nulls in commarea removed by CICS/CICS TG runtime
  - Manual data length transmission control via
    - CTG_ECI_PARMS.eci_commarea_length = n;
    - CTG_ECI_PARMS.commarea_outbound_length = n;
    - CTG_ECI_PARMS.commarea_inbound_length = n;

- **CTG_listSystems**
  - Returns defined CICS servers from ctg.ini

Where are the files?

- **ctgredist.zip** available in *deployable* directory
  - i.e.: /opt/ibm/cicstg/deployable
  - Containers headers/samples and DLLs for all client platforms
  - Licensed for free redistribution for AD and runtime usage

- **ctgclientdoc.zip** available in *doc* directory
  - Doxygen programming reference

- **Headers** - /opt/ibm/cicstg/include
  - ctgclient.eci.h
  - ctgclient.h

- **DLLs** (platform specific)
  - AIX   ../lib/libctgclient.a
  - HP-UX on PA-RISC   .//lib/libctgclient.sl
  - Windows <install_path>   ..\bin\ctgclient.dll
  - Solaris/HP-UX/IA64   ../lib/libctgclient.so

- **Samples**
  - ctgecib1.c - samples/c/eci_v2
  - samp.mak – makefile

- **Help**
  - Doxygen C programming references
  - Available on-line in Information Centre

Using the API

```c
/* 1. Open Gateway connectio n*/
rc = CTG_openRemoteGatewayConnection(hostname, port, gatewayTokenOut,connTimeout);

/* 2. Initialise ECI param. Block */
CTG_ECI_PARMS eciParms;                                   /* ECI parameter block                    */
char commarea[COMMAREA_SIZE];                             /* Commarea                               */
char program[ECI_PROGRAM_NAME_LENGTH + 1] = "EC01"; /* Program to call, 8 characters maximum */
char transId[ECI_TRANSID_LENGTH + 1] = "";          /* Transaction ID, 4 characters maximum   */
char tpn[ECI_TPN_LENGTH + 1] = "";                        /* TPN name, 4 characters maximum         */
char userId[ECI_USERID_LENGTH + 1] = "";            /* CICS user ID, 16 characters maximum    */
char passWd[ECI_PASSWORD_LENGTH + 1] = "";          /* CICS password, 16 characters maximum   */
memset(&eciParms, 0, sizeof(CTG_ECI_PARMS));
memset(commarea, 0, COMMAREA_SIZE);
eciParms.eci_version = ECI_VERSION_2;           /* ECI version 2 includes remote support     */
eciParms.eci_call_type = ECI_SYNC;              /* Synchronous ECI call                      */
eciParms.eci_commarea = commarea;               /* Commarea                                  */
eciParms.eci_commarea_length = COMMAREA_SIZE;   /* Actual length of the commarea             */
eciParms.eci_extend_mode = ECI_NO_EXTEND;       /* Non-extended call                         */
eciParms.eci_luw_token = ECI_LUW_NEW;           /* Zero for a new LUW or non-extended call   */
memcpy(&eciParms.eci_program_name, program, strlen(program));
memcpy(&eciParms.eci_system_name, serverName, strlen(serverName));

/* 3. Make ECI call */
rc = CTG_ECI_Execute(gatewayToken, &eciParms);

/* 4. Close CTG connection */
rc = CTG_closeGatewayConnection(gatewayTokenPtr);
```
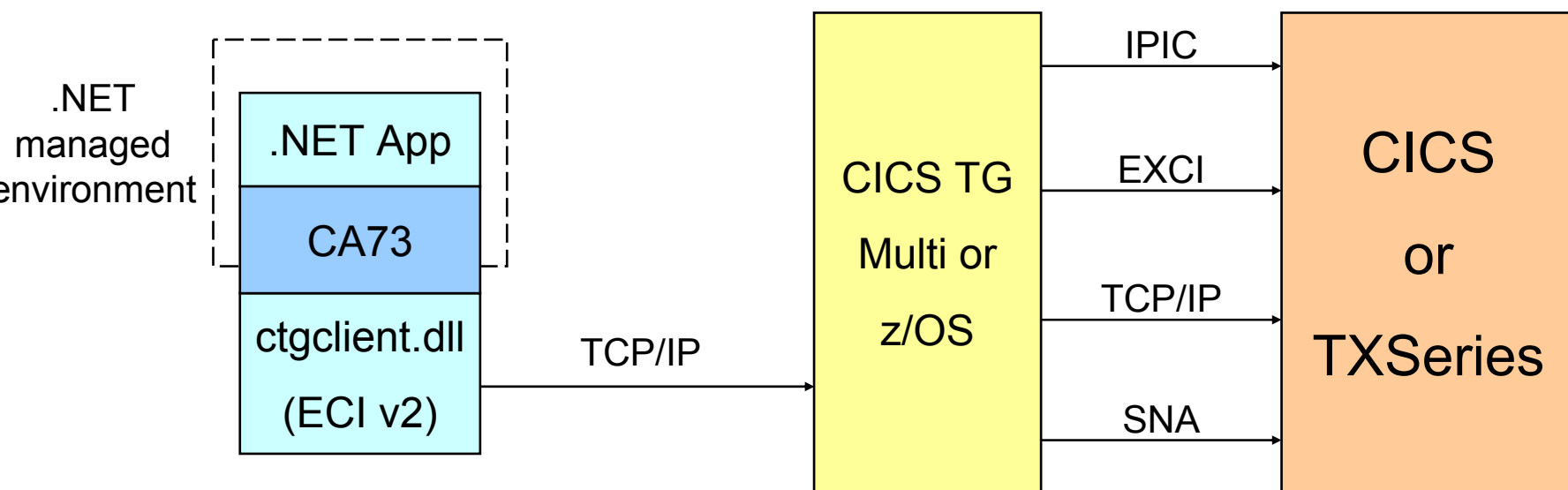
What is .NET?

- An application platform

- Potentially cross architectures

  – Mono project: http://www.mono-project.com/Main_Page

- Controlled execution environment

  – JIT code compilation

  – Garbage collection

  – Large base class library

  – Very similar to a JVM

- Language independence

  – C#, VB.NET, C++/CLI, J#, F#, PowerShell

.NET Application support

- **CA73 – ECI v2 .NET API**

    - Mixed mode DLL providing supported .NET API interface
    - Allows CICS TG ECI v2 APIs to be used in .NET managed runtime for C#, C++ or VB.NET applications
    - Doxygen HTML programming reference
    - Requires CICS TG V7.2 or later (any platform)
    - Category3 SupportPac (fully supported API)

CA73 - ECI v2 .NET API

- **Requirements:**
  - .NET 2.0 runtime or later
  - Any .NET supported language:
    - C#, VB.NET, C++/CLI, G#, F#,

- **Setup:**
  - Global assembly cache
  - Application cache

- **Development environment**

  - MS Visual Studio …

CICS TG V7.2 - Extended Systems Management

- **Remote client support for Statistics API**
  - Stats API handler allows connection from any IP address
  - (APAR PK57718 – available in V7.1.0.2)

    ```
    protocol@statsapi.handler=com.ibm.ctg.server.RestrictedTCPHandler
    protocol@statsapi.parameters=connecttimeout=2000;port=2980;maxconn=5;
    ```

    - Allows multi TCP/IP stack support for OMEGAMON CTG monitoring

  - New C function for connection to remote Gateway
    - openRemoteGatewayConnection()

- **Java API for Statistics collection**
  - Java API for statistics collection
  - Allows development of Java/Web clients for statistics collection
  - ctgstats.jar – available in classes directory

- **Command function for control of request monitoring exits**
  - The eventFired() method now be called with a new RequestEvent of *Command*
  - Allows dynamic control of logging levels in request monitor exits (see SupportPac CH51)
  - Driven using modify command syntax: /F <>,APPL=CMD
  - Also available in CTG V7.1.0.2 (PK68937)

> **Interoperability:**
> Statistics client programs are backwards **and forwards** compatible with remote CTG versions
> V7.2 ctgstats.jar will work with CTG V7.1 or CTG V7.2 or CTGV7.2 +1.

CICS TG plugin for CICS Explorer

- **SupportPac CS05**
  - Released April 09
  - Supports
    CICS TG V7.2 & 7.1
  - Provides operational views
    - Gateway Status
    - Configuration
    - CICS connectivity

  - Two views
    - Gateway daemons
    - CICS connections

V7.2 – Other items

- New install process (Multiplatform)

- APPC support for HP-UX
  - Provides migration from TCP62/AnyNet
  - Available in v7.1.0.2
  - Supports HP Itanium platform
  - Configuration document in information centre for TCP62 to Enterprise Extender migration

- New statistics and monitoring fields
  - Monitoring:
    - RetryCount, CICSServer

  - Statistics
    - LS – Logical server resource group
    - LSx_IALLREQ - Number of requests processed
    - LSx_SLIST – List of logical CICS servers
    - GD_IHAEXIT – Number of CICS request exit calls
    - GD_SDFLTSRV – Default server name
    - GD_SHOSTNAME – CICS TG host name
    - GD_SPLATFORM – CICS TG platform
    - CS_xPROTOCOL – CICS Server protocol
    - CS_ICOMMSFAIL – Number of CICS communication failures

CICS TG V7.2 - get ready

- CICS TS V3.2 (for IPIC/containers)
  - APARs: (PK49017, PK49015, PK49116, PK49490, PK49021, PK51587, PK55495, PK55494)

- WAS V6.1/7.0 (for containers)
  - (Java5 platform required for CICS TG V7/7.1 components)

- Weblogic 9.x
  - Weblogic JCA 1.5 support

- z/OS V1R8 or later
  - RRS APAR OA23163 for XA with V1R8/V1R9

- Solaris V9/10

- AIX V5.3/6.1

- HP-UX 11i v2/3 (RISC and Itanium)

- Linux: RHEL 4/5, SLES 9/10

- Windows: Vista, 2003, XP

Deprecation

- **Withdrawn function**
  - **EWLM**
    - Withdrawn from marketing: http://www.ibm.com/support/docview.wss?uid=swg21305697

  - **TCP62**
    - Withdrawn in V7.0
    - Migration to APPC and Enterprise Extender required

  - **Gateway master process**
    - No longer required for XA load balancing

  - **CTG_RRMNAME Environment variable**
    - Use APPLID definition

- **Consider migration for**
  - **Asynchronous ECI Java APIs**
    - Generic replies and non-validated message qualifiers

  - **AutoJavaGateway**
    - Use local, TCP or SSL protocols instead

  - **Statsport**
    - Use Stats API protocol handler

  - **DFHJVSYSTEM_00**
    - Use DEFAULTSERVER definition

Extra information

- **Whitepapers** -http://www.ibm.com/software/htp/cics/ctg/library/#Whitepapers
  - Integrating WebSphere Application Server and CICS using the CICS TG , G224-7218

- **Developer works**
  - Transactional integration of WebSphere Application Server and CICS with the JCA
  - Exploiting CICS Channels and Containers from Java clients

New

- CICS TG software support page
  - http://www.ibm.com/support/docview.wss?uid=swg21239203

- CICS TG on-line information centre
  - http://publib.boulder.ibm.com/infocenter/cicstg/v7r2m0/index.jsp

- CICS TG SupportPacs - http://www.ibm.com/software/htp/cics/ctg/support/
  - **CA1T – CICS Request exit HA samples**
  - **CC03** - JCA resource adapters
  - **CE51** - CTG Security Exit Samples
  - **CE52** - CCF compatibility JAR for CTG V6.0/WAS 6.0
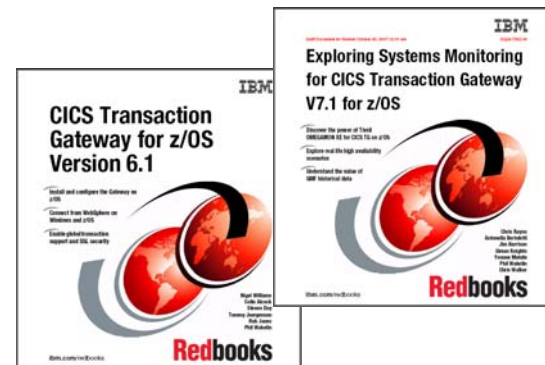 - **CC12** - CTG z/OS environment health check
  - **CH50 – ctgping utility**
  - **CH51 – Simple request monitor**
  - **CA72** - .NET tutorial
  - **CA 73 - .NET API for ECI v2**
  - **CR01** – ctgclean, IPC cleanup utility for Client daemon
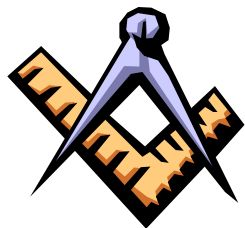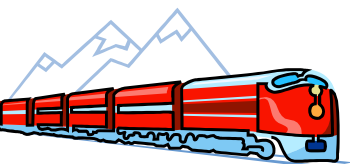 - **CS05 – CTG Explorer**

New

Updated

New

Redbooks
- CICS TG V7.1, SG24-7562 (Systems Monitoring)
- CICS TG v6.1, SG24-6171 (XA and security)
- CICS TG v5, The WebSphere Connector for CICS, SG24-6133-01
- Java Connectors for CICS, SG24-6401
- Developing Connector applications for CICS – SG24-7714

New

CICS TG and JCA – Trends and directions

1. Interoperation and standards

   - Maintain position as de-facto connector for CICS

   - Support IBM and industry wide standards

   - Exploitation of containers/channels

   - Extended integration - new application server environments

2. Address architectural limits

   - High availability/Performance

   - End to end security – ID propagation

   - End to end Transaction tracking

3. Customer value

   - Ease of configuration - CICS Explorer Integration

Connectivity Options: CICS TG and CICS Web Services

CICS TG and CICS Web Services deliver enterprise SOA connectivity to your CICS business logic, including

- Security
- Transactional scope
- Data conversion

- Tooling for Application Developers
- Performance
- Reliability, availability and scalability (RAS)

Both deliver integration opportunities with WebSphere Application Server

Choosing the right connectivity solution

Decision Point	CICS Transaction Gateway	Web Services
Matching your CICS application investment	CICS TG is **non-invasive**, requiring little or no changes to your CICS system. Commonly used when customer does not wish to alter their CICS environment.	The customer must be running on **CICS TS V3** or later. Some changes are required to your CICS system in order to enable Web Services for your applications
Communication methodology	CICS TG is typically deployed in **synchronous** environments where **tight coupling** of applications are required for high performing transactions. Best suited to **high volume, conversational data exchange**.	Web Services provide a **loosely coupled environment** with **maximum flexibility** in data exchanges. Best suited for rich data sets delivered for maximum flexibility and interoperability.
Inbound and outbound capability	CICS TG provides **inbound only** support to CICS from your application server environment. Best suited to modernising interaction with existing assets without altering source CICS business logic	Web Services provides **both inbound & outbound** support, allowing for CICS initiated integration with your application server. Best suited for **composite applications** and direct CICS application modernisation
Organisation Skills	Best suited to organisations wishing to **invest in application server technologies** skills, without needing detailed insight into to CICS applications	Best suited for organisations with **strong CICS skill base** and willingness to invest directly in improving CICS assets for deployment in an SOA.

Different parts of .NET