Connecting CICS with TCP/IP

Gus Kassimis - kassimis@us.ibm.com
Architect, z/OS Communications Server

William Yates – WYATES@uk.ibm.com
CICS Test Architect

Session: 17255
Friday, August 14, 2015: 10:00 AM-11:00 AM

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Trademarks, notices, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DataPower®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- GDPS®
- Geographically Dispersed Parallel Sysplex
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM eServer
- IBM (logo)®
- IBM®
- IBM zEnterprise™ System
- IMS
- InfiniBand ®
- IP PrintWay
- IPDS
- iSeries
- LANDP®
- Language Environment®
- MQSeries®
- MVS
- NetView®
- OMEGAMON®
- Open Power
- OpenPower
- Operating System/2®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- POWER®
- POWER7®
- PowerVM
- PR/SM
- pSeries®
- RACF®
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Sysplex Timer®
- System i5
- System p5
- System x®
- System z®
- System z9®
- System z10
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- z9®
- z10 BC
- z10 EC
- zEnterprise
- zSeries®
- z/Architecture
- z/OS®
- z/VM®
- z/VSE

* All other products may be trademarks or registered trademarks of their respective companies.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license there from.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
- IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Notes:

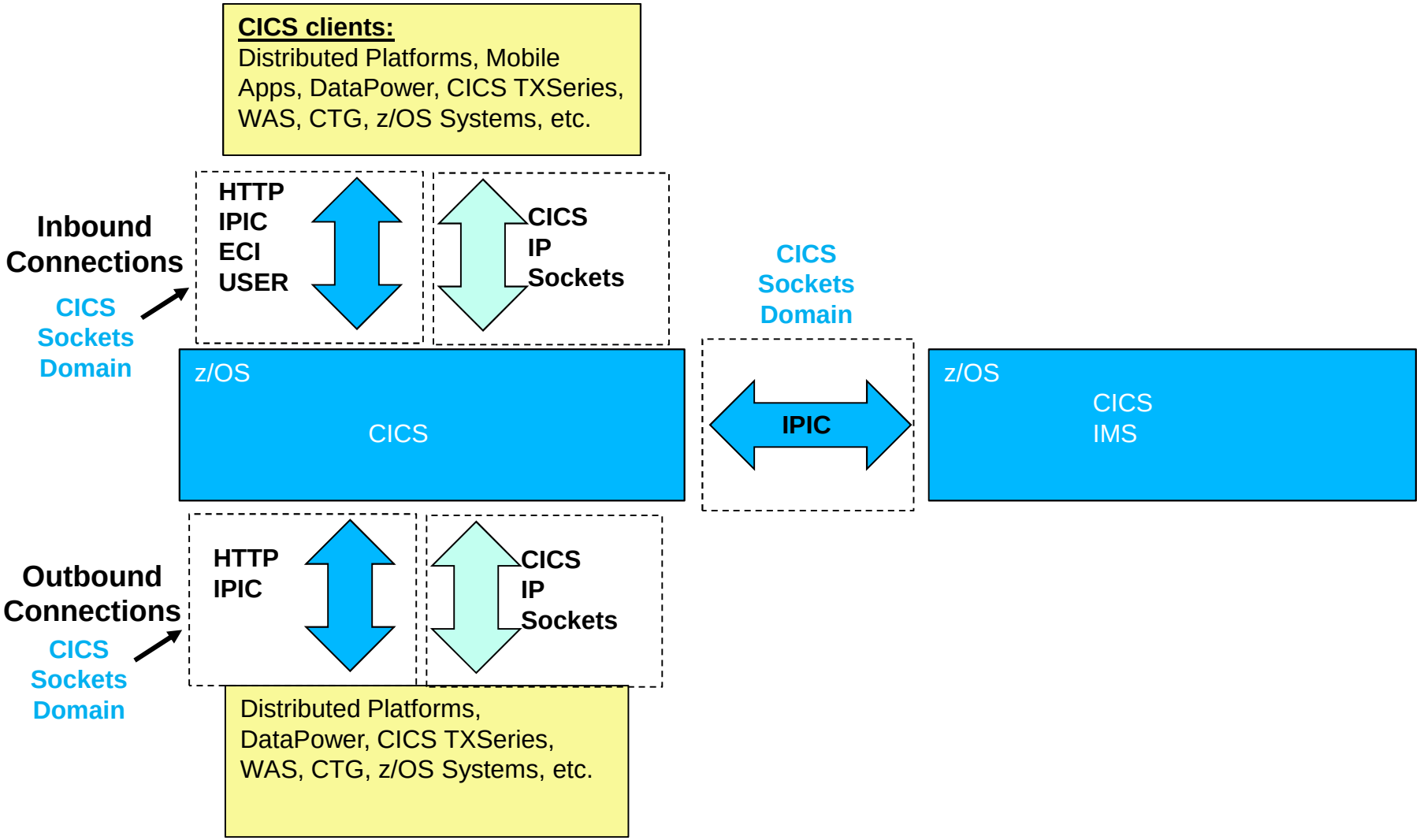
- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Refer to www.ibm.com/legal/us for further legal information.

Agenda

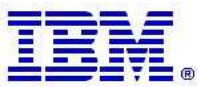
- CICS and TCP/IP Connectivity Overview
- CICS Sockets (aka “IP Sockets”) and the CICS Sockets Domain
- Configuration of CICS Sockets (“IP Sockets”)
- Configuration of TCPIPSERVICE for CICS Sockets Domain
- Monitoring TCP/IP connectivity (CICS Transaction Tracking, APPLDATA support in TCP/IP)
- Securing TCP/IP Communications in CICS
  - AT-TLS overview, Native SSL/TLS support, future directions
- CICS Connection Management
- In Appendix: High Availability Considerations (Sysplex Distributor, SHAREPORT, CP/SM)

CICS and TCP/IP Connectivity Overview



Connecting CICS with TCP/IP

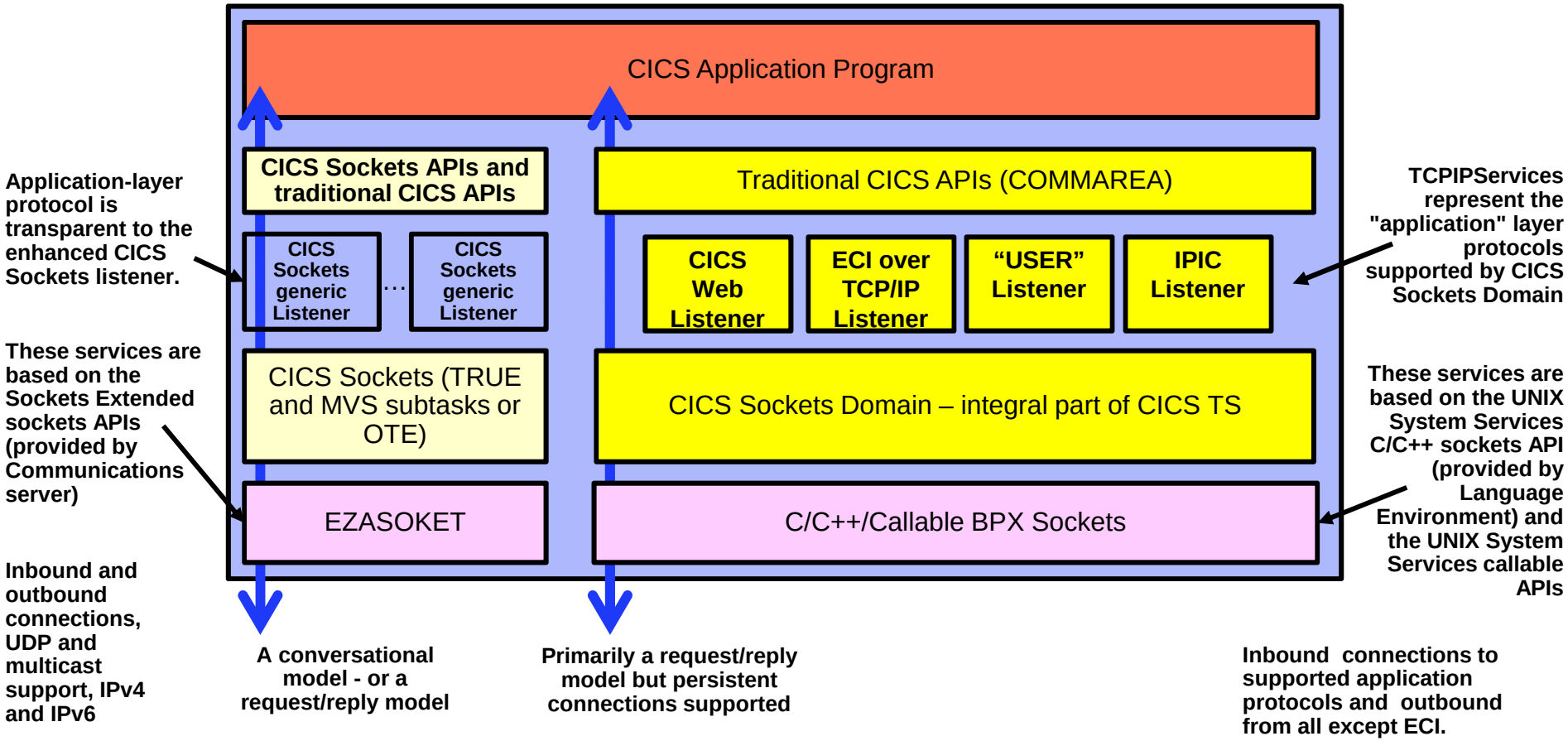
What is CICS Sockets and what is CICS Sockets Domain?



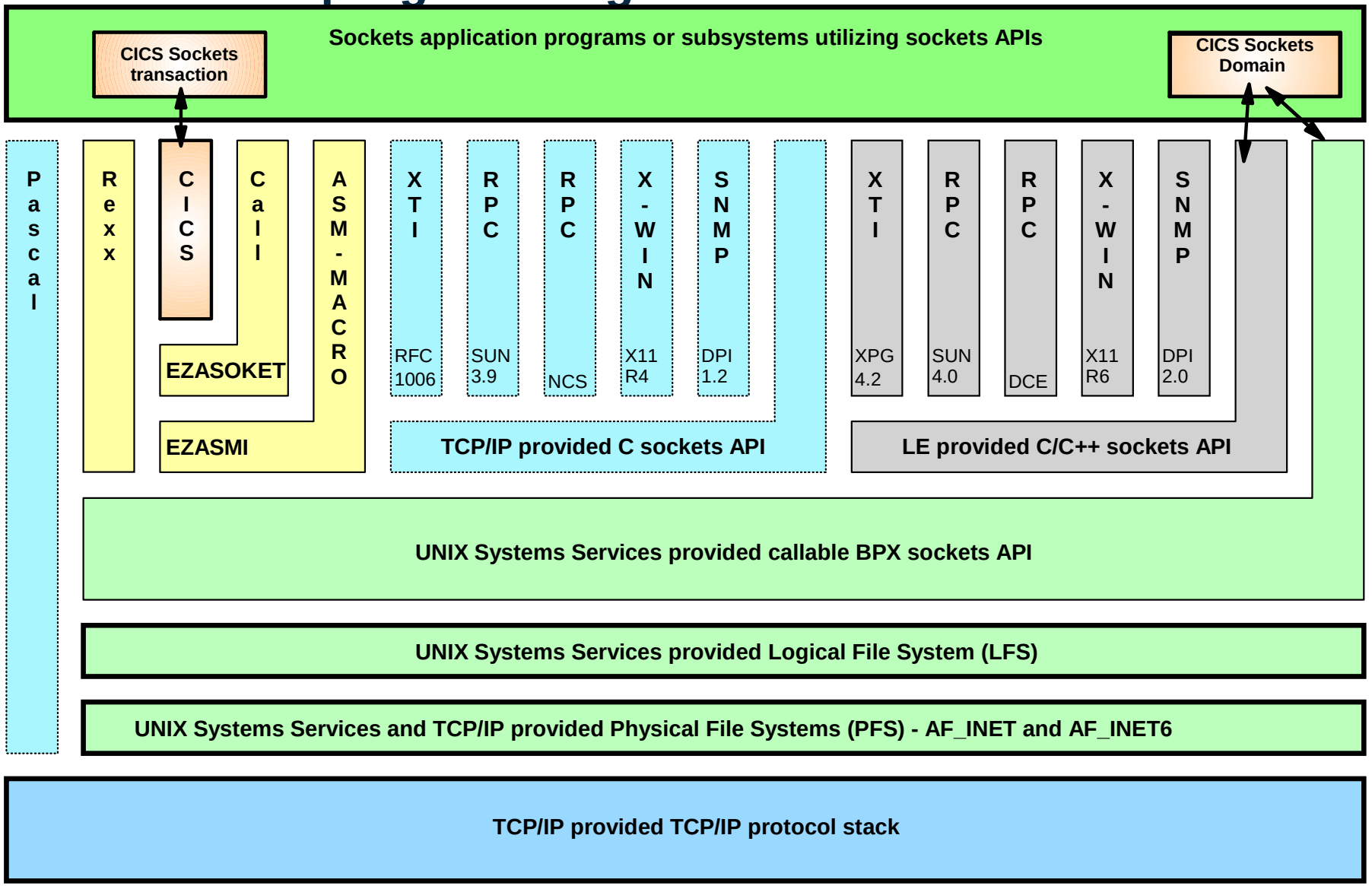
CICS Sockets (aka IP Sockets) vs. CICS Sockets Domain

A CICS Sockets transaction has direct access to the TCP/IP socket and can issue native sockets calls to receive and send data over the socket. Secure connectivity via AT-TLS support. No restrictions in application layer protocol.

A CICS Sockets Domain transaction does not have direct access to the socket, but communicates with CICS Sockets Domain services to receive a request and to send a reply over a socket. Secure connections are supported via native system SSL calls. Restricted to supported application layer protocols.



z/OS Sockets programming interfaces



An attempt at a comparison

Attribute	CICS Sockets	CICS Sockets Domain
Ease of use from a programmer perspective	Easy if you are a sockets programmer, otherwise very difficult	Easy if you are a CICS programmer
Development productivity	Low to medium	Very high if one of the CICS Sockets Domain application layer protocols can be used
Application layer protocol flexibility (message formats, code pages, interaction model, error processing, etc.)	Very high - this is the main reason for using CICS Sockets instead of CICS Sockets Domain – the user protocol needed is unique and not supported by CICS Sockets Domain	Low
Sysplex CICS transaction routing	Limited to CICS regions in an LPAR (sharing a TCP/IP stack)	No GIVE/TAKE Socket support, but DPL can be used across a Sysplex. Response must be sent from same CICS region into which the request arrived
IPv6 support	Yes	Yes from CICS TS 4.1
Web services support (REST, SOAP, XML, JSON)	No specific support	Yes

An attempt at a comparison (cont)

Attribute	CICS Sockets	CICS Sockets Domain
Secure connections	Yes (via AT-TLS)	Yes (via native system SSL usage)
OTE support	Yes	Yes
Application control over socket options in use (KEEPALIVE, TCP_NODELAY, etc.)	Yes	No
CICS as a client (outbound connections)	Yes	Yes for all services except ECI
Support for connectionless sockets (UDP including multicast)	Yes	No
Management (configuration), trace/debug, and monitoring integral part of CICS	No	Yes
Standard client support	No	Yes (HTTP, REST, etc.)
Connection persistence	Somewhat complicated – requires use of an iterative server design or home-written listener	Yes
Cost of high-volume transaction processing	*Perceived lower*	*Perceived higher*

Performance attributes of various TCP/IP connectivity options

- So which connectivity option performs best?
  - It depends!
  - Several factors:
    - Persistence of TCP connections
    - Protocol/Data Representation
    - Encryption requirements
    - Payload size
    - Etc.
  - The following Redpaper presents a comprehensive performance study of all major connectivity options into CICS – an excellent source of information if you are interested in this topic:

Redpaper

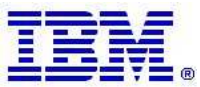
John Burgess
Arndt Eade

IBM CICS Performance Series: A Processor Usage Study of Ways into CICS

http://www.redbooks.ibm.com/redpapers/pdfs/redp4906.pdf

Connecting CICS with TCP/IP

Configuring TCPIPSERVICE



Explanation of a few of the TCPIPService options

```
OVERTYPE TO MODIFY                                        CICS RELEASE = 0690
 CEDA  ALter TCpipservice( TCP28904 )
  TCpipservice   : TCP28904
  GROup          : CMI28904
  DEScription  ==>
  Urm          ==> DFHWBAAX
  POrtnumber   ==> 28904              1-65535
  STatus       ==> Open               Open | Closed
  PROtocol     ==> Http               Http | Eci | User | IPic
  TRansaction  ==> CWXN
  Backlog      ==> 00020              0-32767
  TSqprefix    ==>
  Host         ==> ANY
  (Mixed Case) ==>
  Ipaddress    ==> ANY
  SPeciftcps   ==>
  SOcketclose  ==> No                 No | 0-240000 (HHMMSS)
  MAXPersist   ==> No                 No | 0-65535
  MAXDatalen   ==> 000032             3-524288
```

The TCP/IP port your service will operate on – value should be coordinated with your TCP/IP systems programmer to have him/her reserve that port in the TCP/IP profile for this purpose only

Backlog is used to specify the maximum number of connections waiting in TCP/IP to be serviced by your service, if the backlog is full then new connection requests will be rejected until the backlog queue falls below this value again

IP Address is used to turn your service into a bind specific service – only accepting requests from this IP address

When a client connects to your service, it is according to the underlying application protocol expected to send a request for the service to process. If the client is in error and doesn't send any input data after having connected how long should your wait before closing the connection down
Set to no to use persistent connections

How do you make your CICS Sockets Domain services bind-specific?

There are two ways you can do it:

1. **Specify the local IP address to bind to when defining your TCPIP service:**

```
OVERTYPE TO MODIFY                                        CICS RELEASE = 0690
CEDA  ALter TCpipservice( TCP28904 )
 TCpipservice   : TCP28904
 GROup          : CMI28904
 DEScription  ==>
 Urm          ==> DFHWBAAX
 POrtnumber   ==> 28904              1-65535
 STatus       ==> Open               Open | Closed
 PROtocol     ==> Http               Http | Eci | User | IPic
 TRansaction  ==> CWXN
 Backlog      ==> 00020              0-32767
 TSqprefix    ==>
 Host         ==> ANY
 (Mixed Case) ==>
 Ipaddress    ==> 10.1.1.161
 SPeciftcps   ==>
```

2. **Or have the TCP/IP systems programmer control it in the TCP/IP configuration data set (the TCP/IP Profile)**

```
PORT
     5081 TCP IMWEBSRV BIND 10.1.1.64   ; z/OS HTTP server
     5081 TCP CICSTS32 BIND 10.1.1.161  ; CICS HTTP service
```

How do you decide which IP address your server is listening on?

- The easiest way is to use the netstat command from either TSO or the UNIX shell (or the MVS console).

```
TSO: ALLCONN APPLDATA TCP TCPCS ( CLI  CICSTS32

MVS TCP/IP NETSTAT CS V1R11       TCPIP Name: TCPCS           13:22:46
User Id  Conn     State
-------  ----     -----
CICSTS32 000000A4 Listen
  Local Socket:   10.1.1.161..5081
  Foreign Socket: 0.0.0.0..0
  Application Data:  DFHICICS1A  CWXNHTTP    HTTP    ABC HTTP
CICSTS32 00000045 Listen
  Local Socket:   0.0.0.0..5082
  Foreign Socket: 0.0.0.0..0
  Application Data:  DFHICICS1A  CIEPECI     ECI     CICS ECI
CICSTS32 00000048 Listen
  Local Socket:   0.0.0.0..5084
  Foreign Socket: 0.0.0.0..0
  Application Data:  DFHICICS1A  CISSIPIC    IPIC    CICS IPI
```

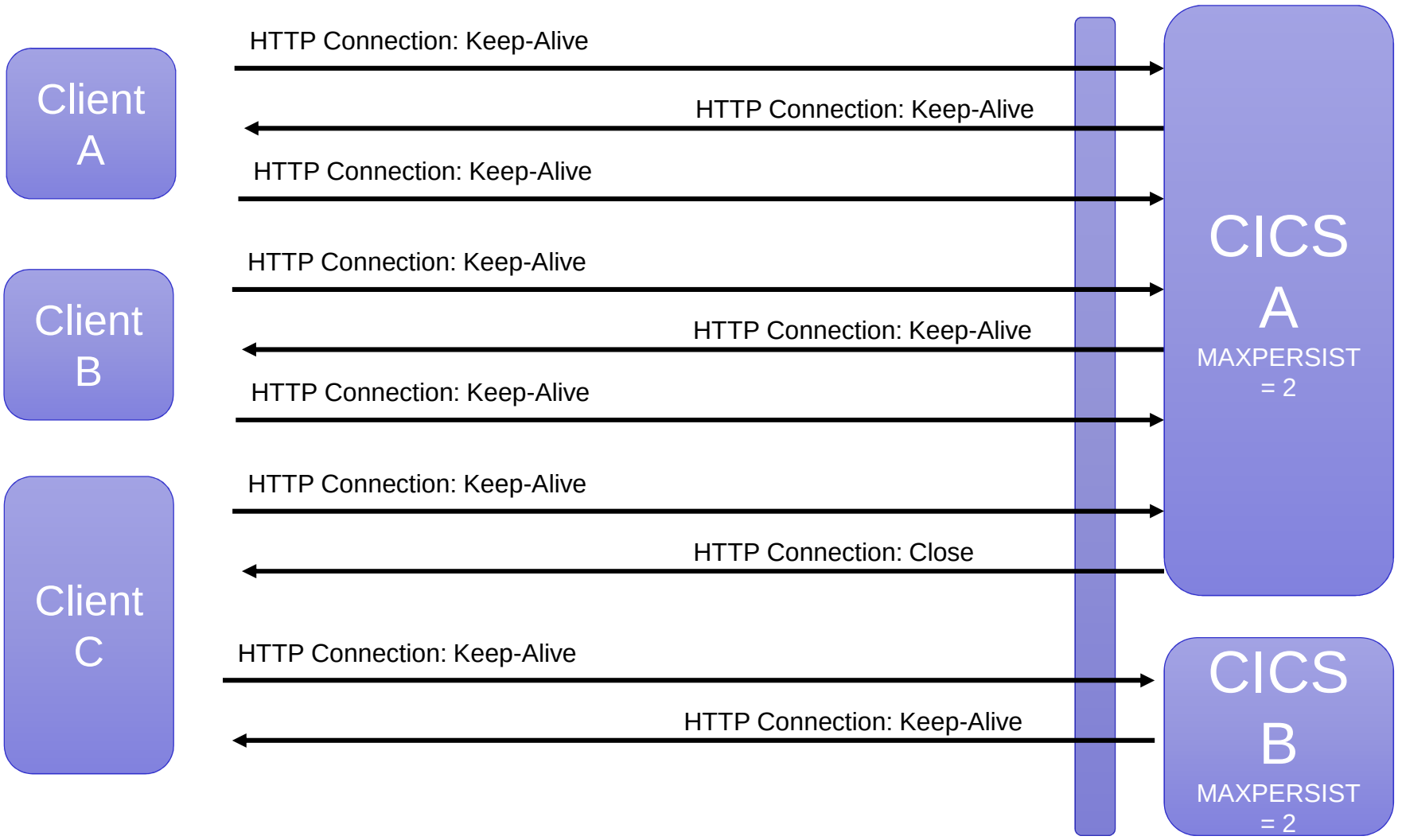
The services you did not make bind-specific - in this example ECI on port 5082, and IPIC on port 5084 show up in your netstat display with the local socket IP address as 0.0.0.0.
– They will receive connection requests that arrive on any of the IP addresses in the HOME list.

Which is better? Bind-specific or not?
– It depends! When using Dynamic VIPAs (DVIPA) bind-specific is typically preferred
  – Guarantees that clients only use DVIPA addresses
  – Allows multiple TCPIPServices to use use the same well known port

CICS Connection Management

HTTP Inbound Connection Throttling (V4.2)

- Connection: Keep-Alive HTTP header
  - Explicit option in 1.0, implied in 1.2
- Permit inbound HTTP connections to persist for future requests
- Provides a way to “throttle” incoming requests without leaving them queuing
- MAXPERSIST option on TCPIPSERVICE
  - Socket 'closed' if number exceeded
  - Default is NO
- SOcketclose attribute should be set to NO

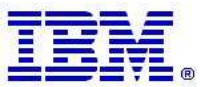


HTTP Outbound Connection Pooling

- Re-use connections which have the same properties, as defined by URIMAP
  - EP HTTP Adapter, Web Services, WEB API programs
- SOCKETCLOSE timeout option on client URIMAP
  - Non-zero value means HTTP requests using that URIMAP can use connections (sockets) from a pool
    - Applies to any HTTP requests using the same client URIMAP
    - No code changes needed to benefit, except when using CICS WEB interface
    - Benefits HTTP EP adapter
  - SOCKETCLOSE is timeout time for length of time socket remains available for reuse from the pool
  - Socket will be removed from pool if errors returned, or if any problems are detected

TCP/IP for CICS Systems Programmers

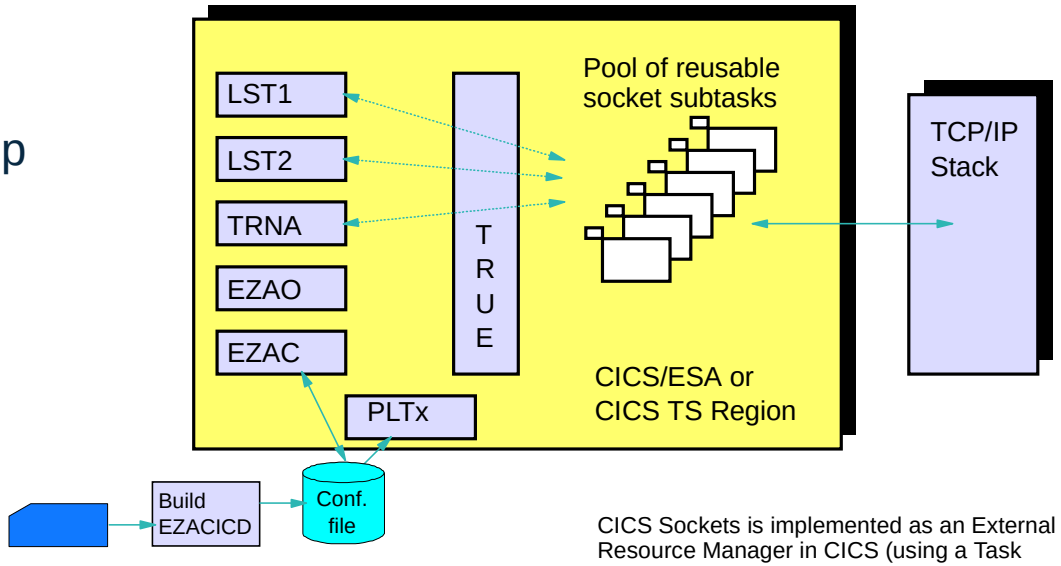
Introduction to CICS Sockets (aka IP sockets)



CICS Sockets overview

- Multiple listeners – each instance separately configurable
- Enhanced listener has no requirements on client input data
- Multiple listeners in many CICS regions can share listener port number
- User ID security
- SSL/TLS support by means of AT-TLS
- Configuration file and transaction (EZAC)
- Operations transaction to start/stop individual listeners (EZAO)
- PLT-enabled start and termination
- Reusable subtasks
- OTE enabled
- IPv6 support
- UDP and multicast support

- **CICS Sockets is a component of the Communications Server for z/OS, not CICS TS itself.**
- **It is a general-purpose sockets programming API to be used by CICS application programmers for implementing native (low-level) sockets communication in z/OS CICS transaction programs.**



CICS Sockets is implemented as an External Resource Manager in CICS (using a Task Related User Exit - a TRUE).

CICS entry in CICS Sockets configuration file - EZAC transaction

```
EZAC,ALTer,CICS                                          APPLID = CICS1A

Overtype to Enter

APPLID       ===> CICS1A          APPLID of CICS System
TCPADDR      ===> TCPCS           Name of TCP Address Space
NTASKS       ===> 100             Number of Reusable Tasks
DPRTY        ===> 010             DPRTY Value for ATTACH
CACHMIN      ===> 010             Minimum Refresh Time for Cache
CACHMAX      ===> 020             Maximum Refresh Time for Cache
CACHRES      ===> 005             Maximum Number of Resolvers
ERRORTD      ===> CSMT            TD Queue for Error Messages
SMSGSUP      ===> NO              Suppress Task Started Messages
TERMLIM      ===> 000             Subtask Termination Limit
TRACE        ===> YES             Trace CICS Sockets
OTE          ===> NO              Open Transaction Environment
TCBLIM       ===> 00000           Number of Open API TCBs
PLTSDI       ===> NO              CICS PLT Shutdown Immediately
APPLDAT      ===> YES             Register Application Data




PF 3 END                                                  12 CNCL
```

CICS Sockets always uses one TCP/IP stack only - which one is specified with the TCPADDR keyword.

To get APPLDATA in Netstat for CICS Sockets Sockets, you must specify YES to APPLDAT on the CICS entry

Listener entry in CICS Sockets configuration file - EZAC transaction - screen 1 of 2

```
EZAC,ALTer,LISTENER (standard listener.  screen 1 of 2)      APPLID = CICS1A

Overtype to Enter

APPLID       ===> CICS1A              APPLID of CICS System
TRANID       ===> CSKL                Transaction Name of Listener
PORT         ===> 03001               Port Number of Listener
AF           ===> INET                Listener Address Family
IMMEDIATE    ===> YES                 Immediate Startup   Yes|No
BACKLOG      ===> 040                 Backlog Value for Listener
NUMSOCK      ===> 100                 Number of Sockets in Listener
ACCTIME      ===> 060                 Timeout Value for ACCEPT
GIVTIME      ===> 000                 Timeout Value for GIVESOCKET
REATIME      ===> 000                 Timeout Value for READ
RTYTIME      ===> 015                 Stack Connection Retry Time
LAPPLD       ===> INHERIT             Register Application Data




Verify parameters, press PF8 to go to screen 2
                   or ENTER if finished making changes

PF 3 END                          8 NEXT                          12 CNCL
```

You specify if the listener is an IPv4 or an IPv6 listener (INET or INET6)

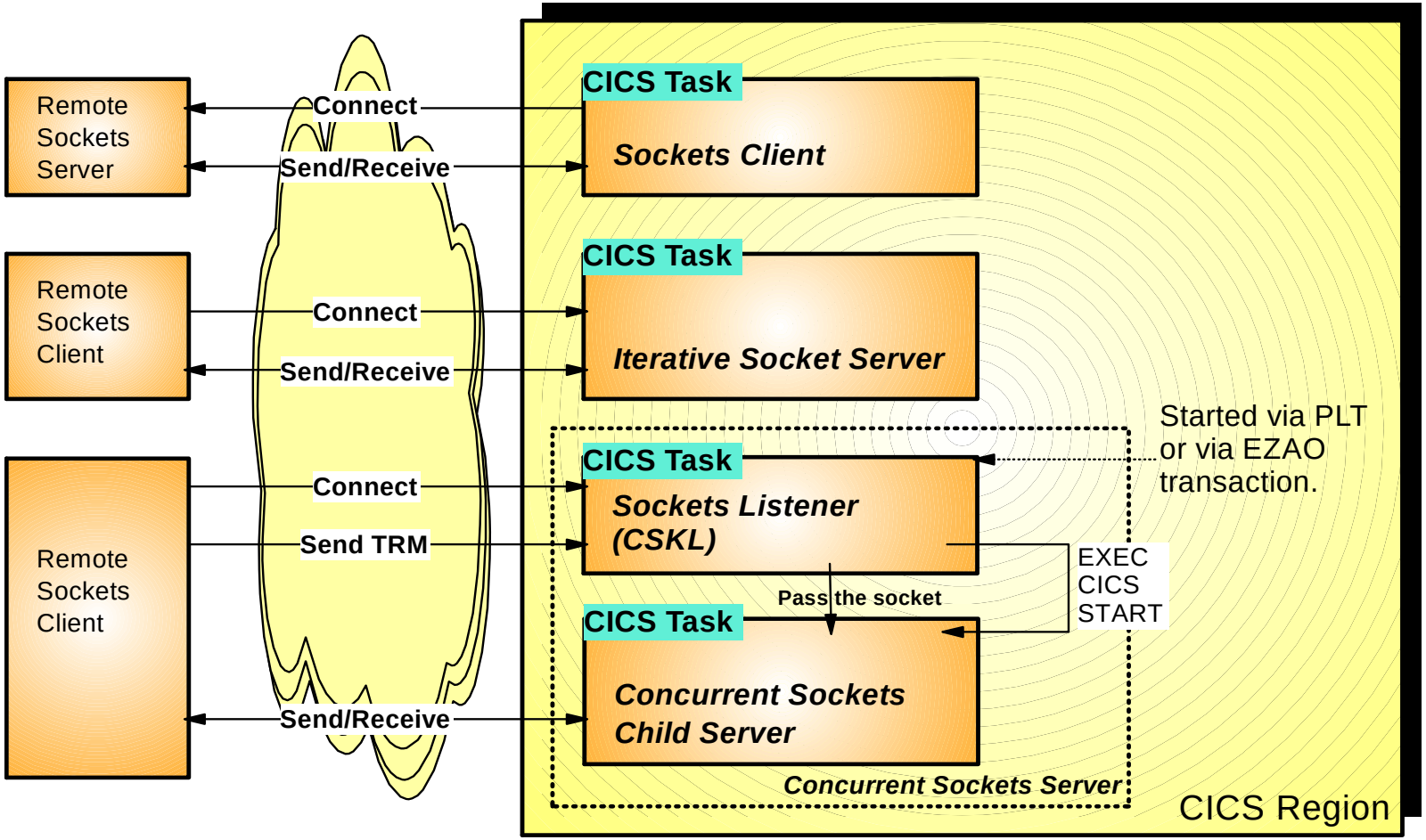
Similar comments about the backlog value. Ensure this is large enough to handle workload spikes

To get APPLDATA in Netstat for this listener, specify YES or INHERIT (inherit from the CICS entry)

Listener entry in CICS Sockets configuration file - EZAC transaction - screen 2 of 2

```
 EZAC,ALTer,LISTENER (standard listener.  screen 2 of 2)      APPLID = CICS1A

Overtype to Enter


MINMSGL      ===> 004                  Minimum Message Length
TRANTRN      ===> NO                   Translate TRNID     Yes|No
TRANUSR      ===> NO                   Translate User Data Yes|No
SECEXIT      ===>                      Name of Security Exit
GETTID       ===> NO                   Get AT-TLS ID  (YES|NO)
USERID       ===>                      Listener User ID









Verify parameters, press PF7 to go back to screen 1
                   or ENTER if finished making changes

PF 3 END           7 PREV                                          12 CNCL
```

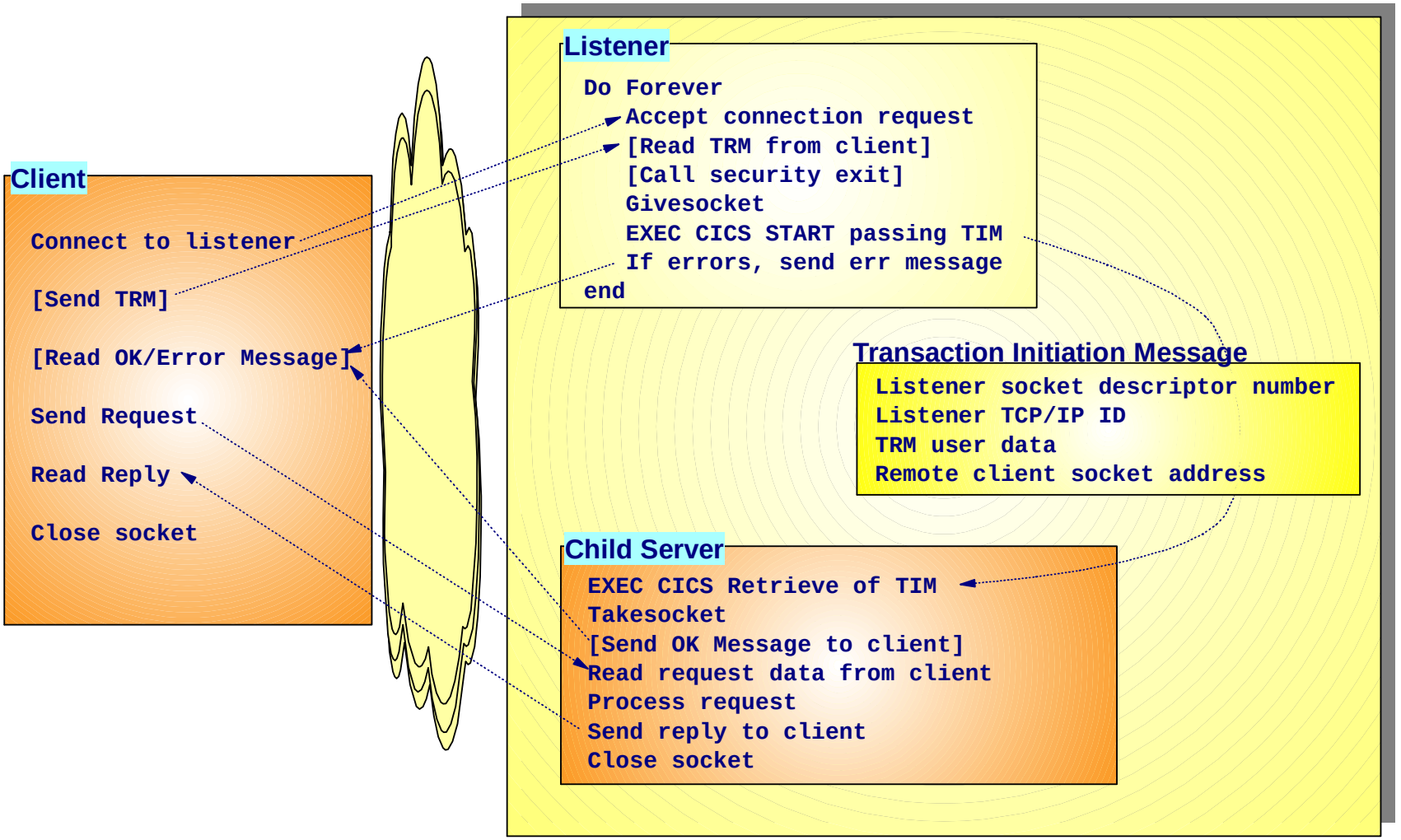
CICS Sockets program categories in CICS



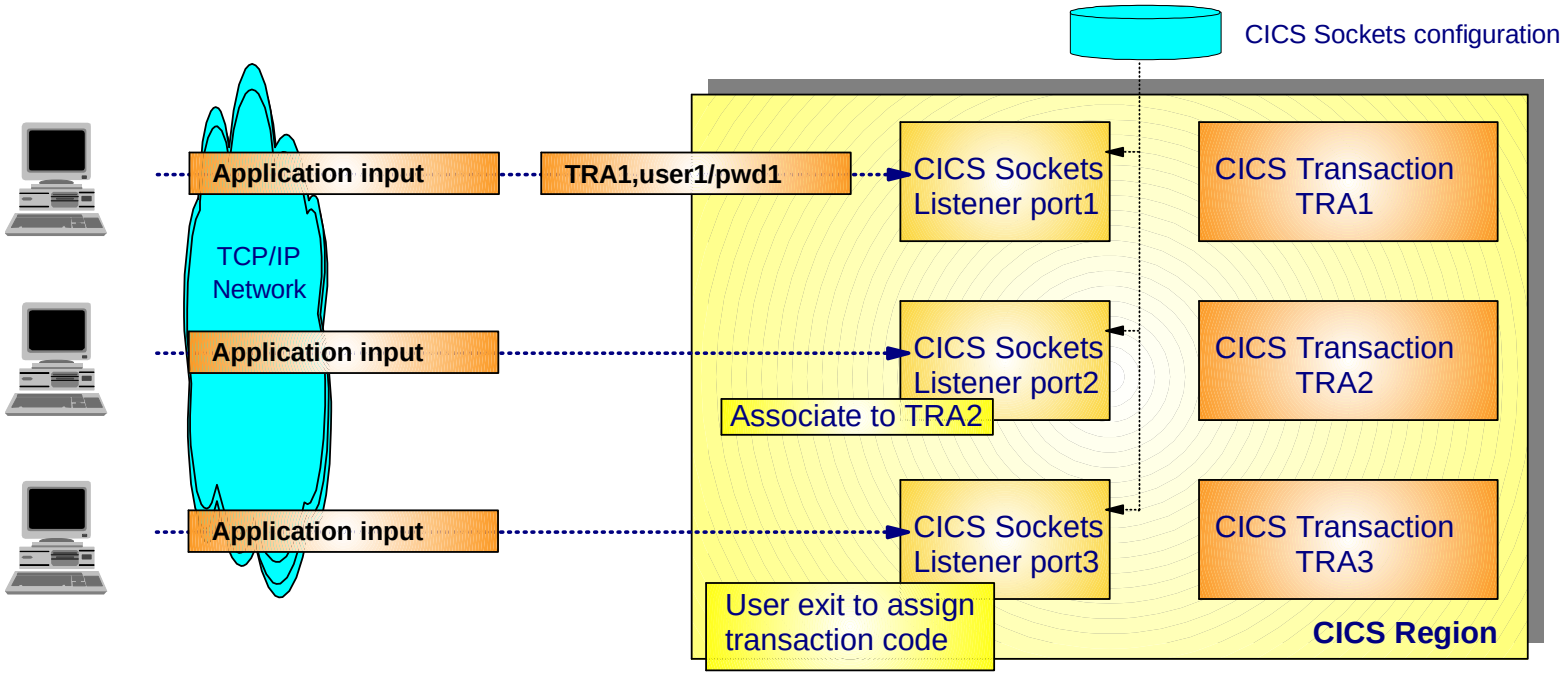
TRM: Transaction Request Message

Please note that use of the Enhanced Sockets Listener removes the requirement for the client sending a transaction request message - in reality removing any requirements from the CICS Sockets infrastructure on the application-level protocol between the client and the server running in CICS.

Concurrent CICS Sockets server - overview



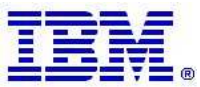
Client – Listener interactions



- Three ways to launch CICS transactions:
  - Via a Transaction Request Message – standard listener
  - Via a listener configuration option to associate listener instance (and port) with one specific CICS transaction code
  - Via the listener security user exit, driven by the listener
- With the last two options, data may be sent by the client in completely free format.

Connecting CICS with TCP/IP

Monitoring TCP/IP connectivity

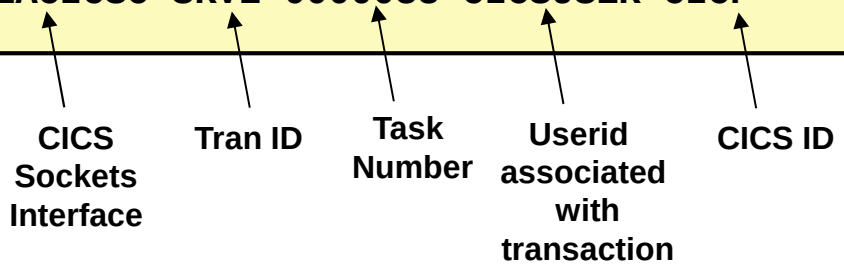


Providing CICS context to TCP Connections - APPLDATA

- APPLDATA is identification data a sockets application can associate with a sockets end point.
- APPLDATA can be displayed with Netstat, it is included in TCP/IP SMF records, and in the Network Management API.
  - Allows correlation of CICS transactions and TCP connections – bridges the gap between CICS and TCP/IP
  - Netstat also supports filtering using APPLDATA (can search through TCP connections using CICS context)
  - Enables better troubleshooting for CICS related TCP connections from the network side (e.g. Identify a problem TCP connection, debug problems, drop the connection, etc.)
- Both CICS IP Sockets and CICS Sockets Domain exploit APPLDATA to provide context information to TCP/IP
  - CICS IP Sockets provides varying information based on current state of the socket (Listen, Connect, GiveSocket, TakeSocket – details in the appendix)
  - CICS Sockets domain information varies depending on whether connection is associated with a TCPIPSERVICE or IPCONN resource

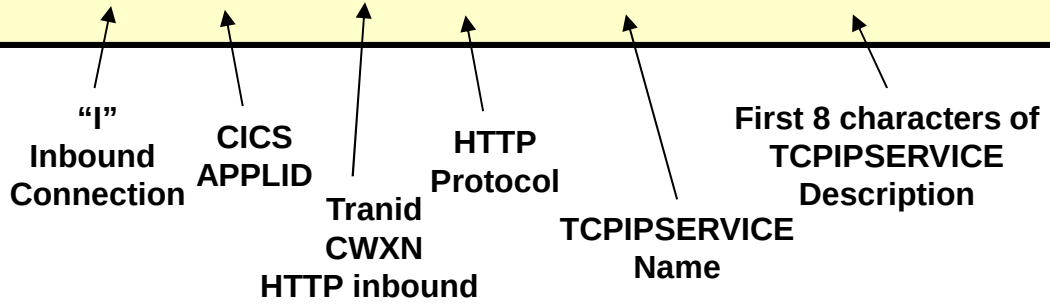
Sample Netstat ALL command for CICS sockets with APPLDATA

```
Client Name: TCPCS                    Client Id: 0000001E
  Local Socket: 9.67.115.5..23
  Foreign Socket: 9.27.11.182..4665
    BytesIn:            0000000000000001062
    BytesOut:           0000000000000000480
    SegmentsIn:         0000000000000000019
    SegmentsOut:        0000000000000000018
    StartDate:          01/09/2012       StartTime:          14:27:37
    Last Touched:       14:27:37         State:              Establsh
                          .
                          .
                          .
    Ancillary Input Queue: N/A
    Application Data:   EZACICSO SRV1 0000038 CICSUSER CICP
```



Sample Netstat ALL command for CICS Sockets Domain with APPLDATA

```
CLIENT NAME: CICSTORA                    CLIENT ID: 02921E93
  LCLSOCK: 10.1.1.5..5000                  FGNSOCK: 10.1.1.6..10915
    BYTESIN:             0000529787          BYTESOUT:            0001358551
    SEGMENTSIN:          0000000916          SEGMENTSOUT:         0000001198
    STARTDATE:           01/29/2015          STARTTIME:           14:25:05
    LAST TOUCHED:        14:25:47            STATE:               ESTABLSH
         .
         .
         .
         .
  ANCILLARY INPUT QUEUE: N/A
  APPLICATION DATA:   DFHICIPABC  CWXNHTTP    T5000N  PORT 500
```



CICS transaction tracking – Multiple ports of origin

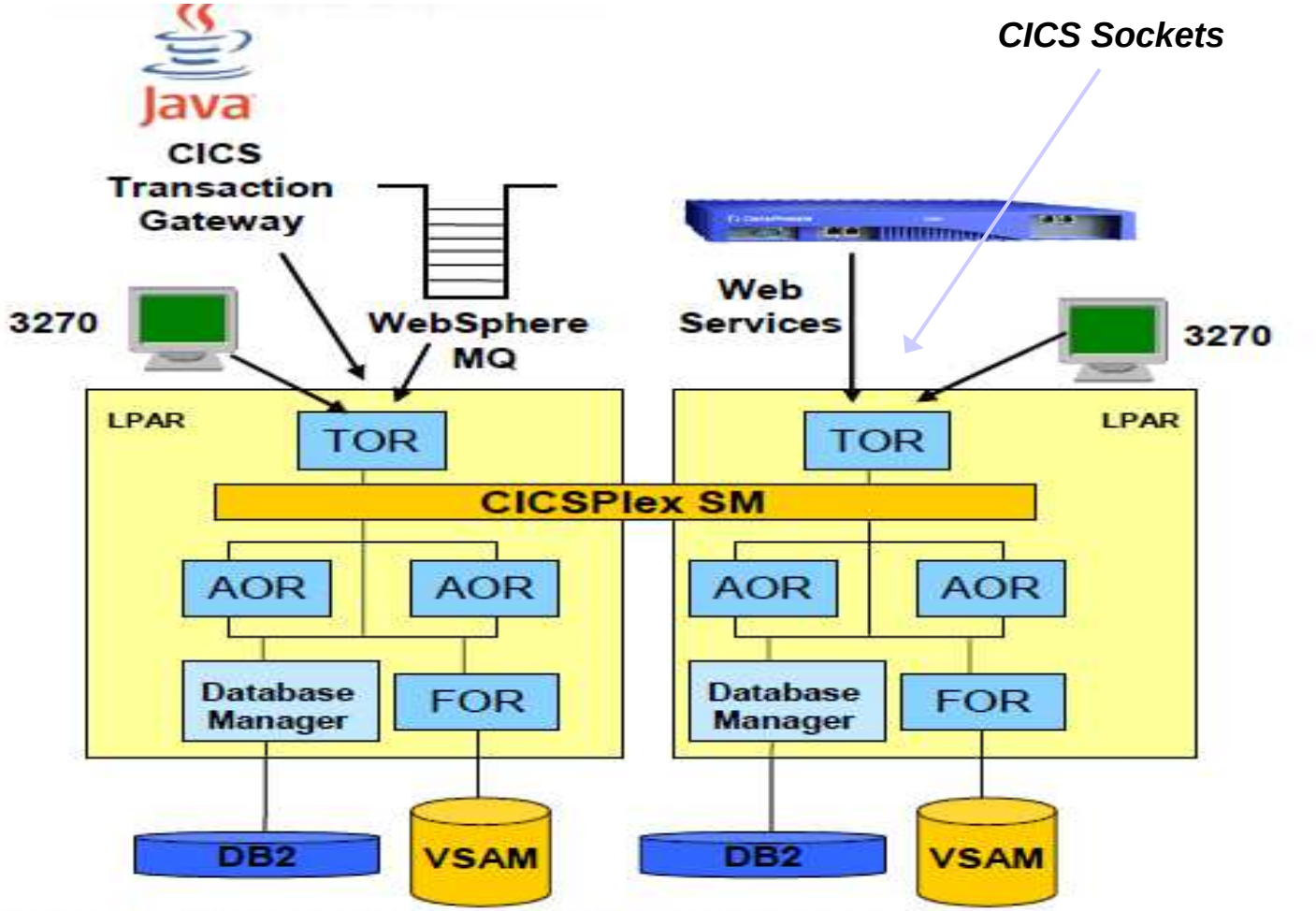
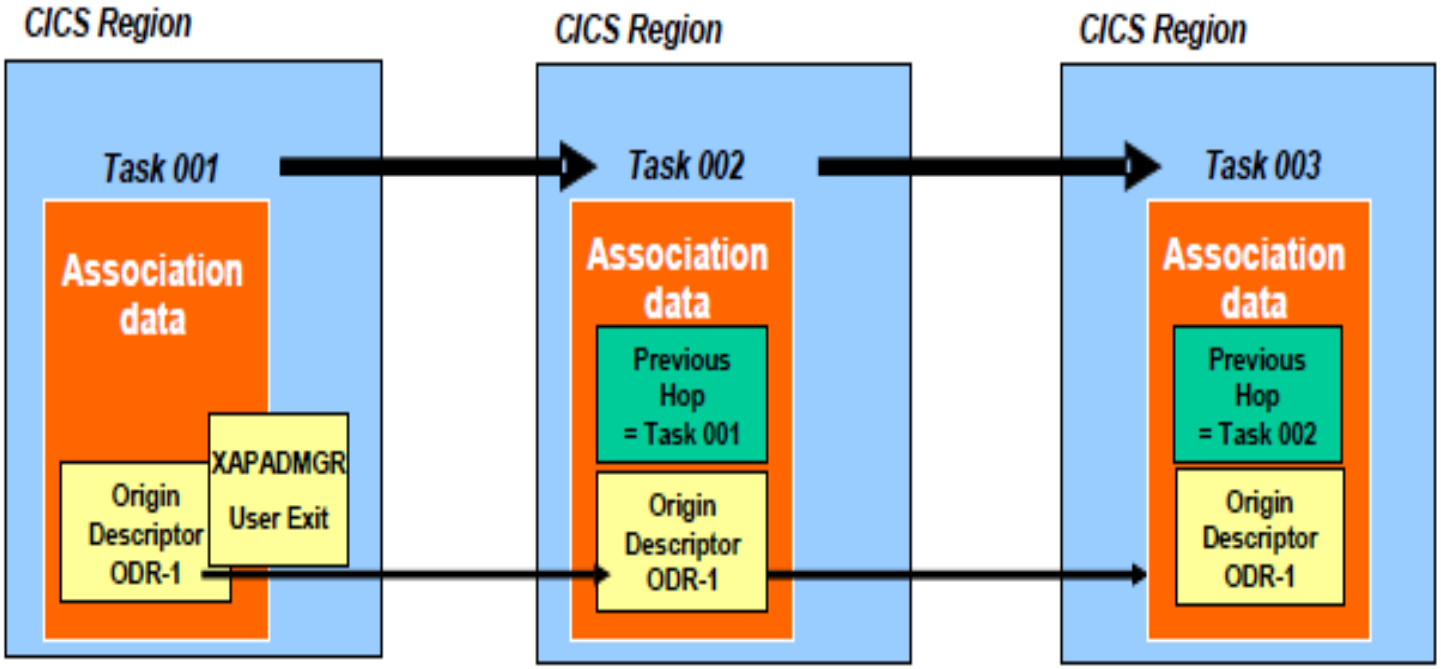


Figure 1. CICSplex with multiple front ends

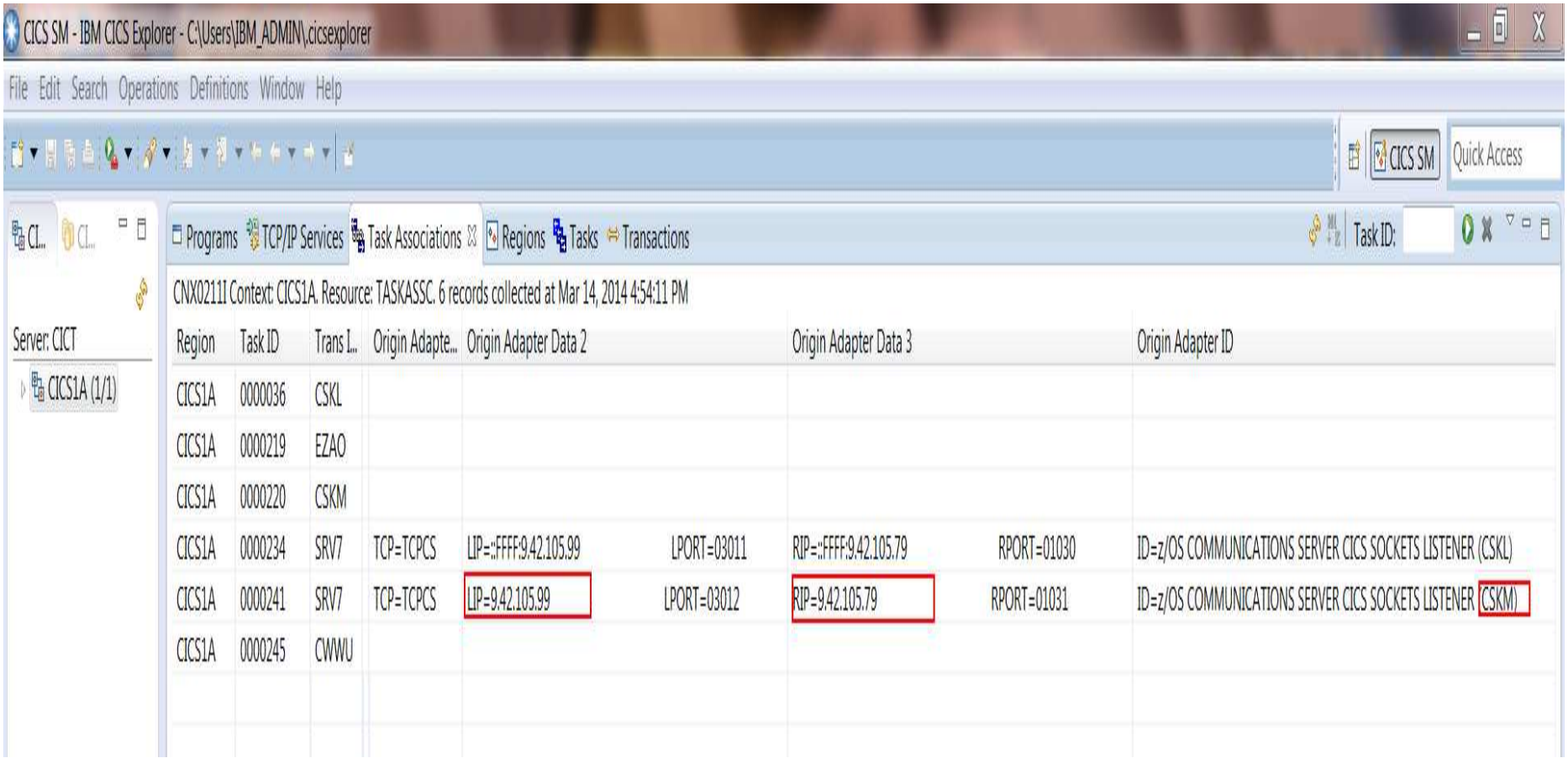
CICS transaction tracking – Propagating tracking info across CICS tasks/transactions (CICS TS 4.2)



- CICS Transaction tracking enables you to locate a transaction in CICS based on knowledge of the entry point, such as an IP address or queue name. With this information, it is possible to use new search functions in the CICS Explorer® to search the CICSplex to locate other active tasks that have been initiated from the originating task, and to build a picture of the relationships between the associated tasks.

CICS Sockets transaction tracking support for CICS TCP/IP IBM Listener

- In z/OS V2R2, the CICS Sockets Listener will provide to CICS the IP addresses and port numbers of the local and remote session partners for use by the CICS Explorer or Session Monitor.
  - This support is only for transactions that are started via the CSKL listener.



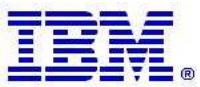
CICS transaction tracking support for CICS TCP/IP IBM Listener …

Parameters to be provided by the EZACIC01 TRUE:

Parameters	Value
ODAPTRID	ID=z/OS COMMUNICATIONS SERVER CICS SOCKETS LISTENER (CSKL)
ODAPTRDATA1	TCP=tcpip_name
ODAPTRDATA2	LIP=local IP address LPORT=local port number
ODAPTRDATA3	RIP=remote IP address RPORT=remote port number

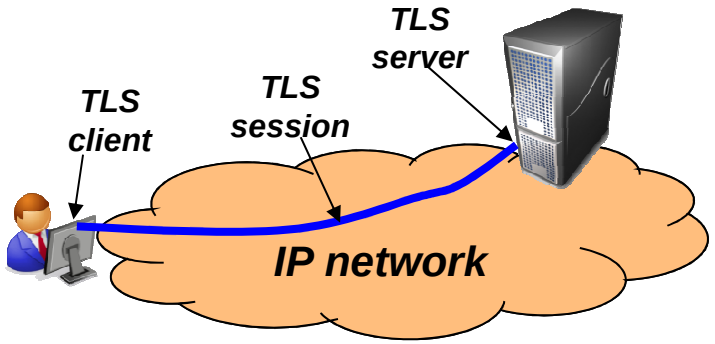
Connecting CICS with TCP/IP

Securing CICS TCP/IP Communications



Transport Layer Security (TLS/SSL) overview

- Transport Layer Security (TLS) is defined by the IETF **
  - Based on Secure Sockets Layer (SSL)
    - TLS defines SSL as a version of TLS for compatibility
- Provides secure connectivity between two TLS security session endpoints
  - TLS session
- Full application payload encryption and data authentication / integrity
- TLS security session endpoint plays either a client or server role
- Session endpoint authentication via X.509 certificates
  - Server authentication required
  - Client authentication optional (mutual authentication)



Full application payload encryption

TLS/SSL encryption:

SrcIP	DestIP	SrcPort	DestPort	Data
192.168.100.1	192.168.1.1	50002	443	@%$#*&&^^!:"J)*GVM><

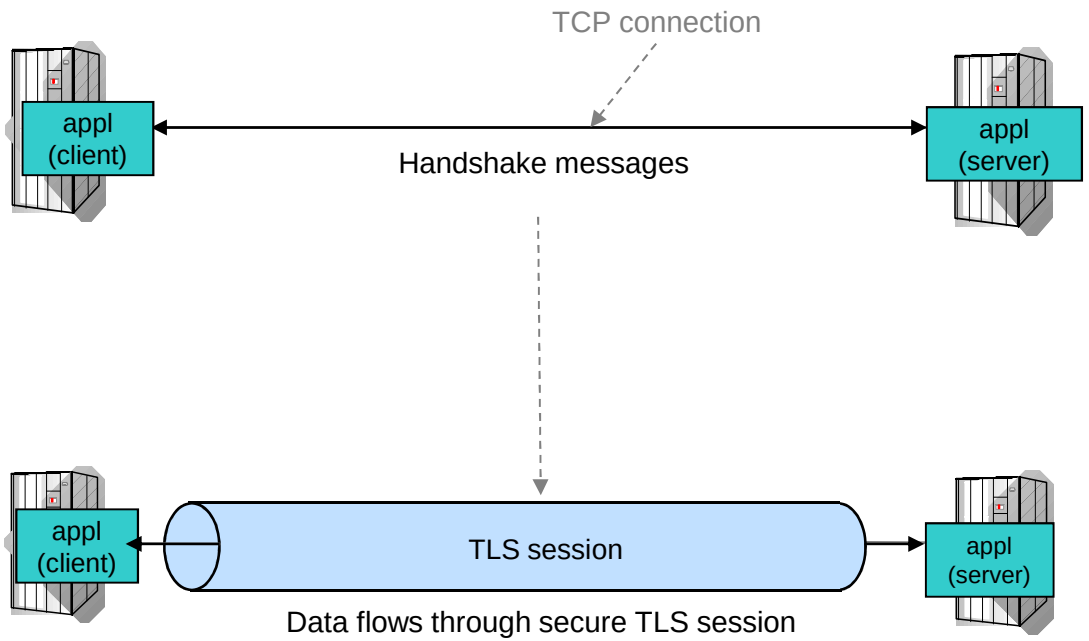
**** For our purposes, SSL and TLS are equivalent and one term implies the other**

TLS/SSL protocol basics

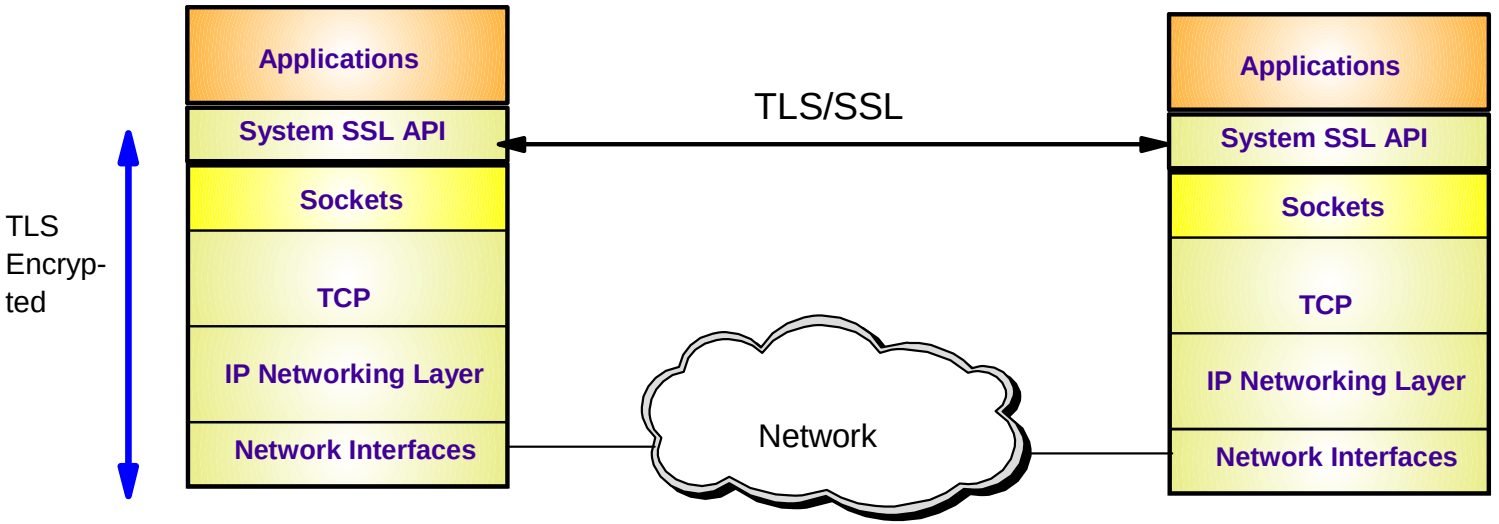
1 Client application initiates TLS handshake which authenticates the server (and, optionally, client) and negotiates a cipher suite to be used to protect data

Upon successful completion of the handshake, a secure TLS session exists for the application partners

2 Data flows through secure session using symmetric encryption and message authentication negotiated during handshake



Transport Layer Security enablement

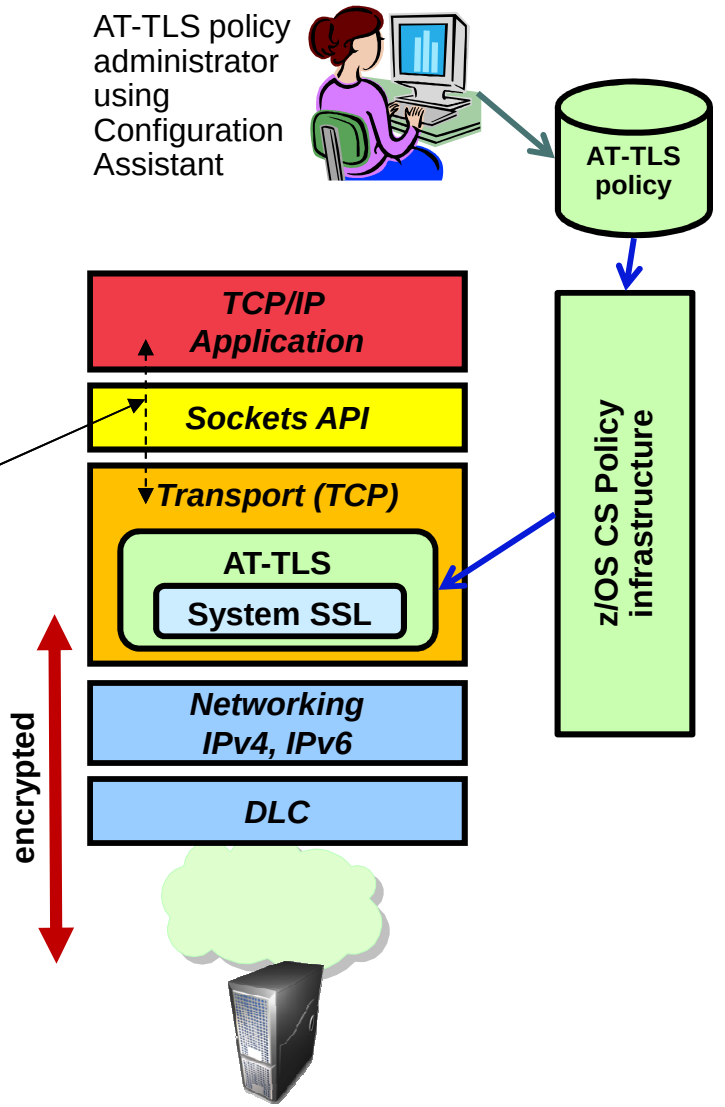


- TLS traditionally provides security services as a socket layer service
  - TLS requires reliable transport layer,
    - Typically TCP (but architecturally doesn't have to be TCP)
  - UDP applications cannot be enabled with traditional TLS
    - There is now a TLS variant called Datagram Transport Layer Security (DTLS) which is defined by the IETF for unreliable transports
- On z/OS, System SSL (a component of z/OS Cryptographic Services) provides an API library for TLS-enabling your C and C++ applications
- Java Secure Sockets Extension (JSSE) provides libraries to enable TLS support for Java applications
  - However, there is an easier way…

… Application Transparent TLS!

z/OS Application Transparent TLS overview

- **Stack-based TLS**
  - TLS process performed in TCP layer (via System SSL) without requiring any application change (transparent)
  - AT-TLS policy specifies which TCP traffic is to be TLS protected based on a variety of criteria
    - Local address, port
    - Remote address, port
    - Connection direction
    - z/OS userid, jobname
    - Time, day, week, month
- **Application transparency**
  - Can be fully transparent to application
  - An optional API allows applications to inspect or control certain aspects of AT-TLS processing – "application-aware" and "application-controlled" AT-TLS, respectively
- **Available to TCP applications**
  - Includes CICS Sockets
  - Supports all programming languages except PASCAL
- **Supports standard configurations**
  - z/OS as a client or as a server
  - Server authentication (server identifies self to client)
  - Client authentication (both ends identify selves to other)
- **Uses System SSL for TLS protocol processing**
  - Remote endpoint sees an RFC-compliant implementation
  - interoperates with other compliant implementations

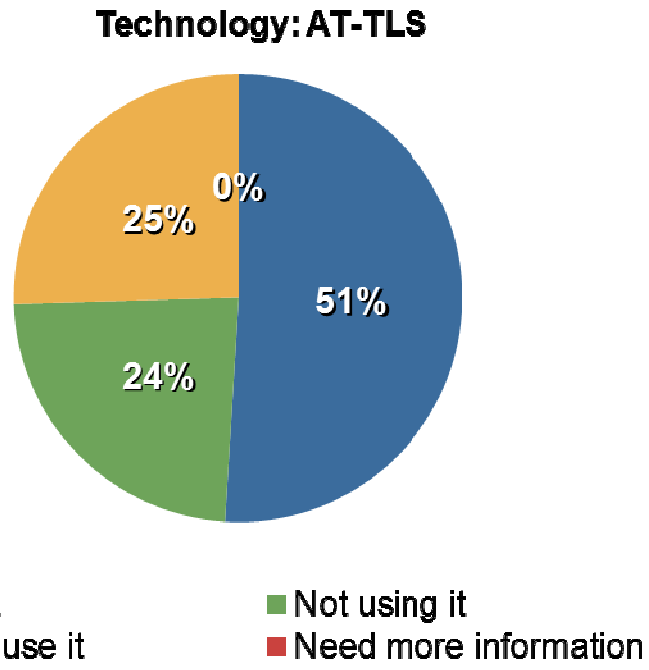


Some z/OS applications that use AT-TLS

- Comm Server applications
  - TN3270 Server
  - FTP Client and Server
  - CSSMTP
  - Load Balancing Advisor
  - IKE NSS client
  - NSS server
  - Policy agent
  - DCAS server
- DB2 DRDA
- IMS-Connect
- JES2 NJE

- IBM Multi-Site Workload Lifeline
- Tivoli Netview applications
  - MultiSystem Manager
  - NetView Management Console
- RACF Remote Sharing Facility
- CICS Sockets applications
- InfoSphere Guardium S-TAP
- 3rd Party applications
- Customer applications

Customer Survey Results
1Q2014
AT-TLS Adoption and Plans for Adoption



Advantages of using AT-TLS

- **Reduce costs**
  - Application development
    - Cost of System SSL integration
    - Cost of application's TLS-related configuration support
  - Consistent TLS administration across z/OS applications
  - Gain access to new features with little or no incremental development cost

- **Complete and up-to-date exploitation of System SSL features**
  - AT-TLS makes the vast majority of System SSL features available to applications
  - AT-TLS keeps up with System SSL enhancements – as new features are added, your applications can use them by changing AT-TLS policy, not code

- **Ongoing performance improvements**
  Focus on efficiency in use of System SSL

- **Great choice if you haven't already invested in System SSL integration**
  Even if you have, consider the long-term cost of keeping up vs. short term cost of conversion

AT-TLS support for TLS v1.2 and Related Features

…Added in z/OS V2R1

- TLS Protocol Version 1.2 (RFC 5246):
  - Twenty-one new cipher suites
    - 11 new HMAC-SHA256 cipher suites
    - 10 new AES-GCM cipher suites
- ***Addresses NIST SP800-131a requirements***
- Support Elliptic Curve Cryptography (ECC)
  - Twenty new ECC cipher suites
    - ECC cipher suites for TLS (RFC 4492)
- Support for Suite B cipher suites (RFC 5430)
  - TLS 1.2 is required
  - ECC is required
  - Suite B has two levels of cryptographic strength that can be selected
    - 128 or 192 bit
- Transport Layer Security (TLS) Renegotiation Extension (RFC 5746):
  - Provides a mechanism to protect peers that permit re-handshakes
  - When supported, it enables both peers to validate that the re-handshake is truly a continuation of the previous handshake

… Planned for z/OS V2R2

- Support retrieval of revocation information through the Online Certificate Status Protocol (OCSP)
- Support HTTP retrieval of CRLs
- Support for RFC 5280 certificate validation mode

AT-TLS application types

- **Not enabled**
  - No policy or policy explicitly disables AT-TLS for application traffic
  - Application may optionally use System SSL directly

- **Basic**
  - Policy enables AT-TLS for application traffic
  - Application is unchanged and unaware of AT-TLS
  - Application protocol unaffected by use of AT-TLS (think HTTP vs. HTTPS)

- **Aware**
  - Policy enables AT-TLS for application traffic
  - Application uses the SIOCTTLSCTL ioctl to extract AT-TLS information such as partner certificate, negotiated version and cipher, policy status, etc.

- **Controlling**
  - Policy enables AT-TLS and specifies ApplicationControlled ON for application traffic
  - Application protocol may negotiate the use of TLS in cleartext with its partner
  - Application uses the SIOCTTLSCTL ioctl to extract AT-TLS information (like an aware application) and to control TLS operations:
    - Start secure session
    - Reset session
    - Reset cipher

CICS IP Sockets & CICS Sockets Domain – TLS/SSL considerations

CICS Sockets (IP Sockets)

- Depends exclusively on AT-TLS for its TLS/SSL encryption processing
- Works for inbound and outbound connections
- Is an AT-TLS Aware Application
- Listener Configuration options (GETTID=YES) allow the Listener to extract the userid associated with the client certificate)
  - The listener can then associated that userid with the started child server transaction (Requires that the userid associated with the Listener transaction has SAF CICS Surrogate Authority)

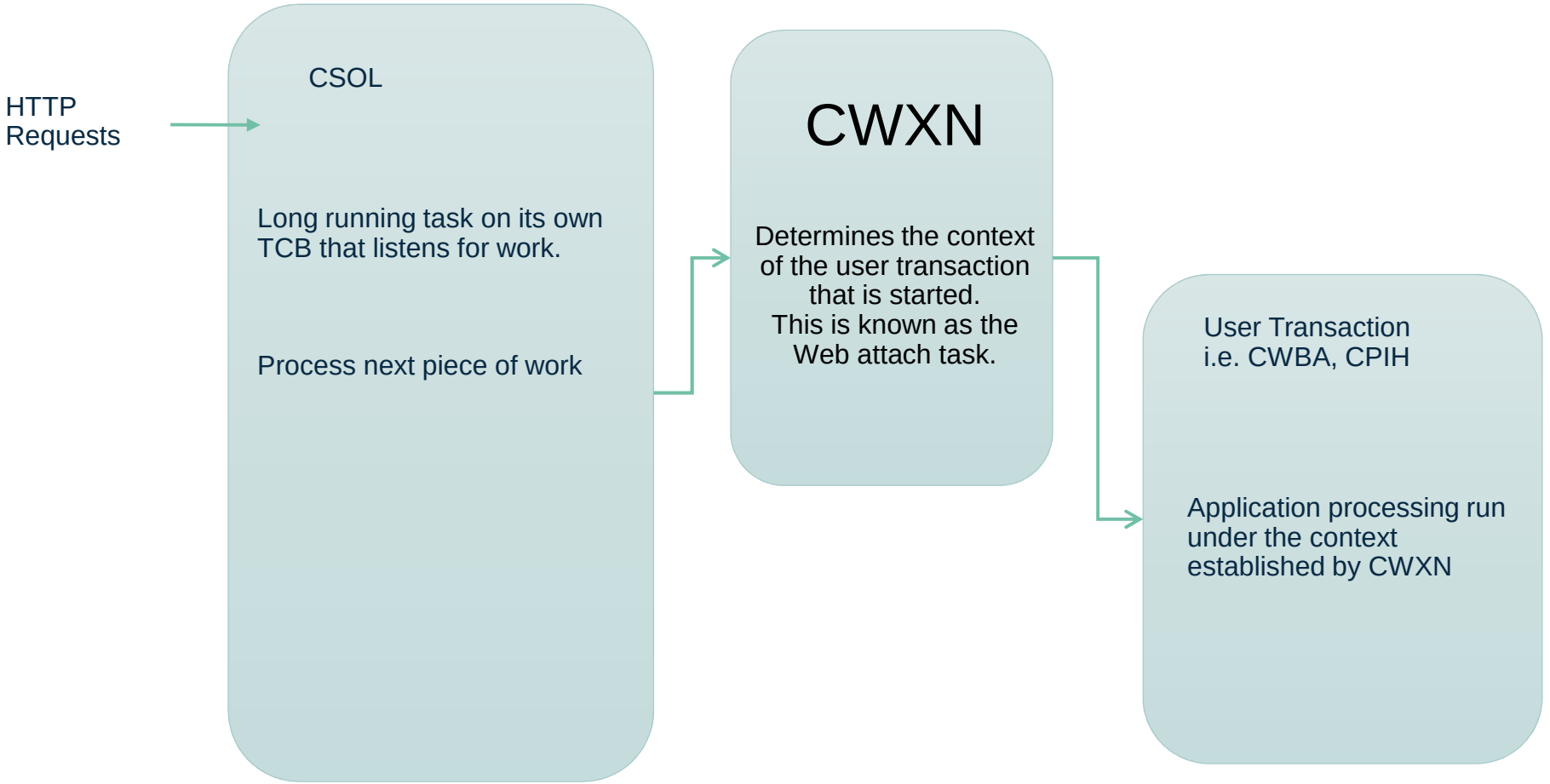
CICS Sockets Domain

- Current support:
  - Imbedded TLS/SSL support built into the CICS Sockets domain
  - Direct invocation of System SSL services
  - Configuration options to indicate various TLS/SSL encryption criteria
  - Works for inbound and outbound connections
- Future direction:
  - Become AT-TLS aware application (5.3 Beta)
    - Allows CICS to extract client certificate and userid information
  - Inbound (server-side) support initially
    - Allows CICS Sockets domain to optimize communications performance by minimizing context switches
    - Allows CICS to pick up latest TLS/SSL enhancements transparently
  - Outbound (client-side) enablement for AT-TLS is a future objective

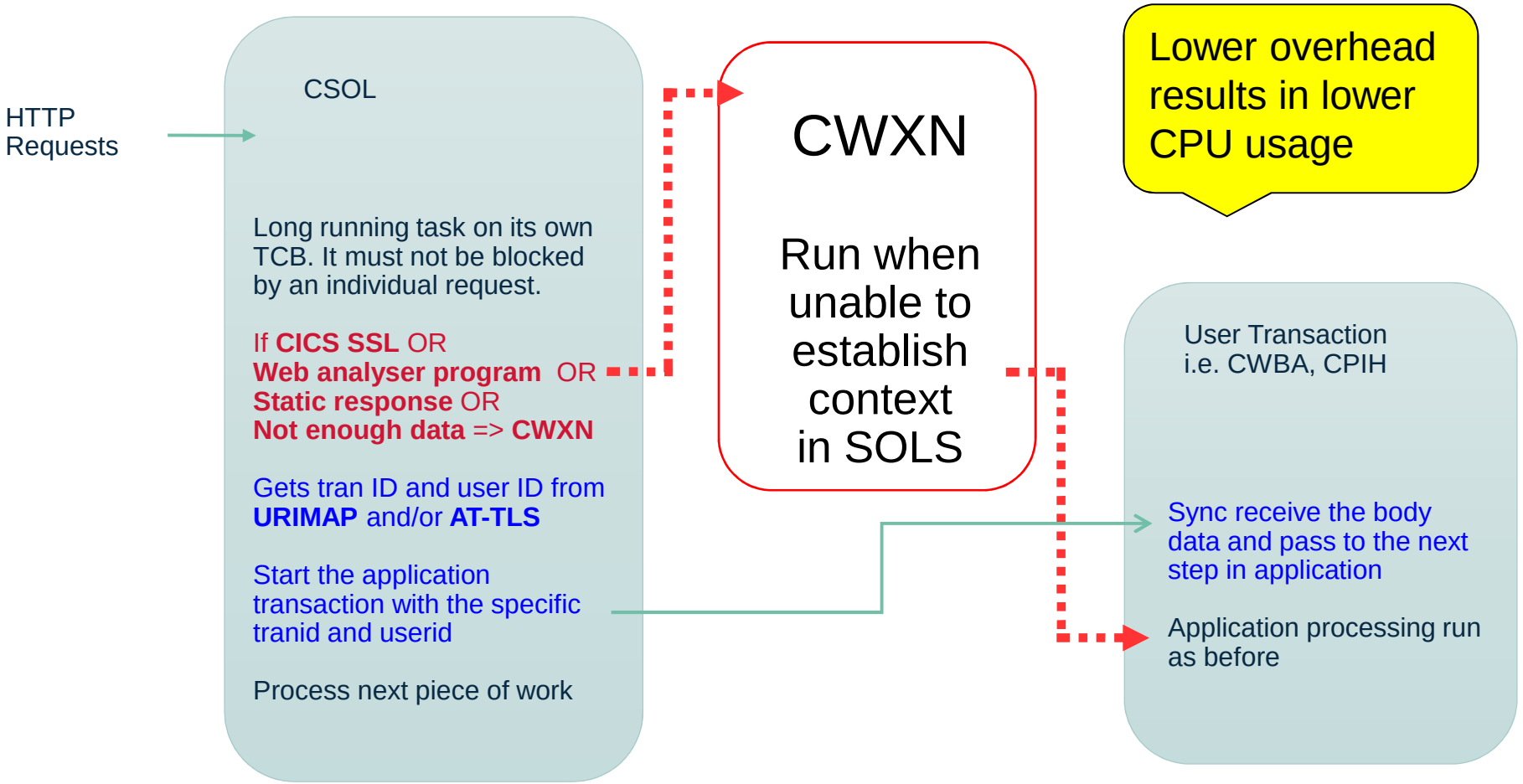
CICS TS V5.3 open beta – Performance Improvements

- HTTP efficiency, including for web services
- SSL/TLS improvements
- Other areas of improvement
- Some numbers from a CICS TS V5.3 open beta development level

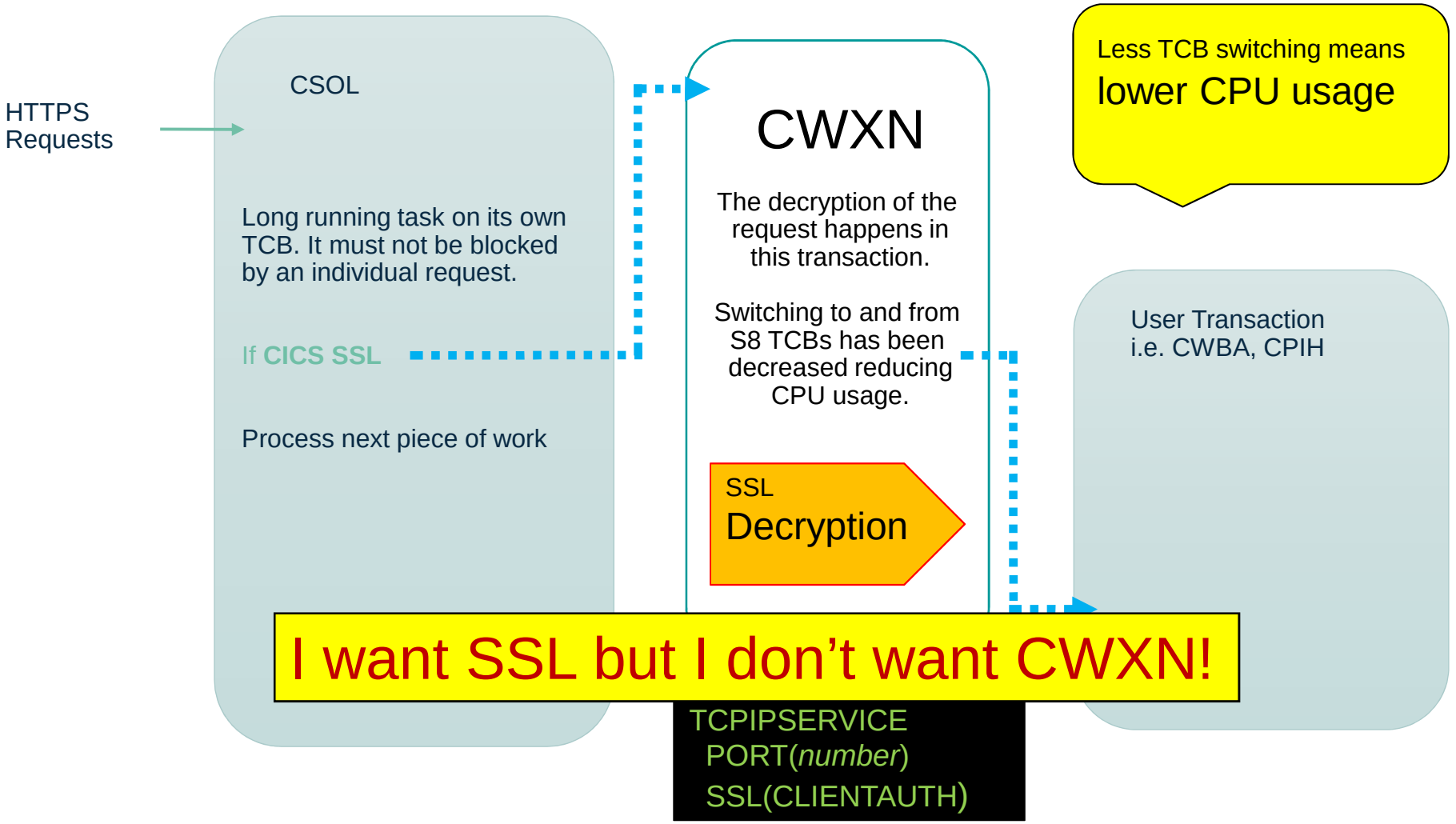
HTTP Pre open beta



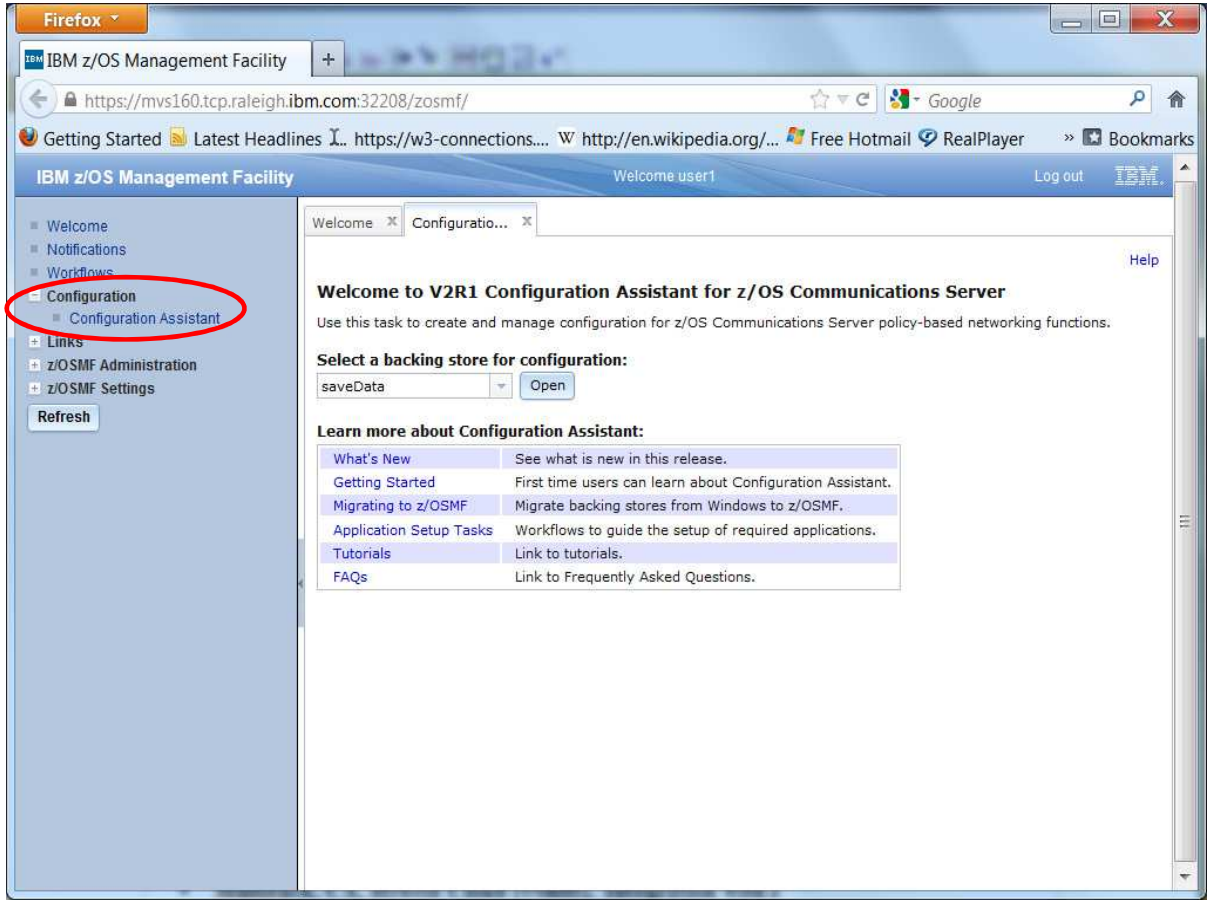
HTTP in CICS TS V5.3 open beta



SSL in CICS TS V5.3 open beta

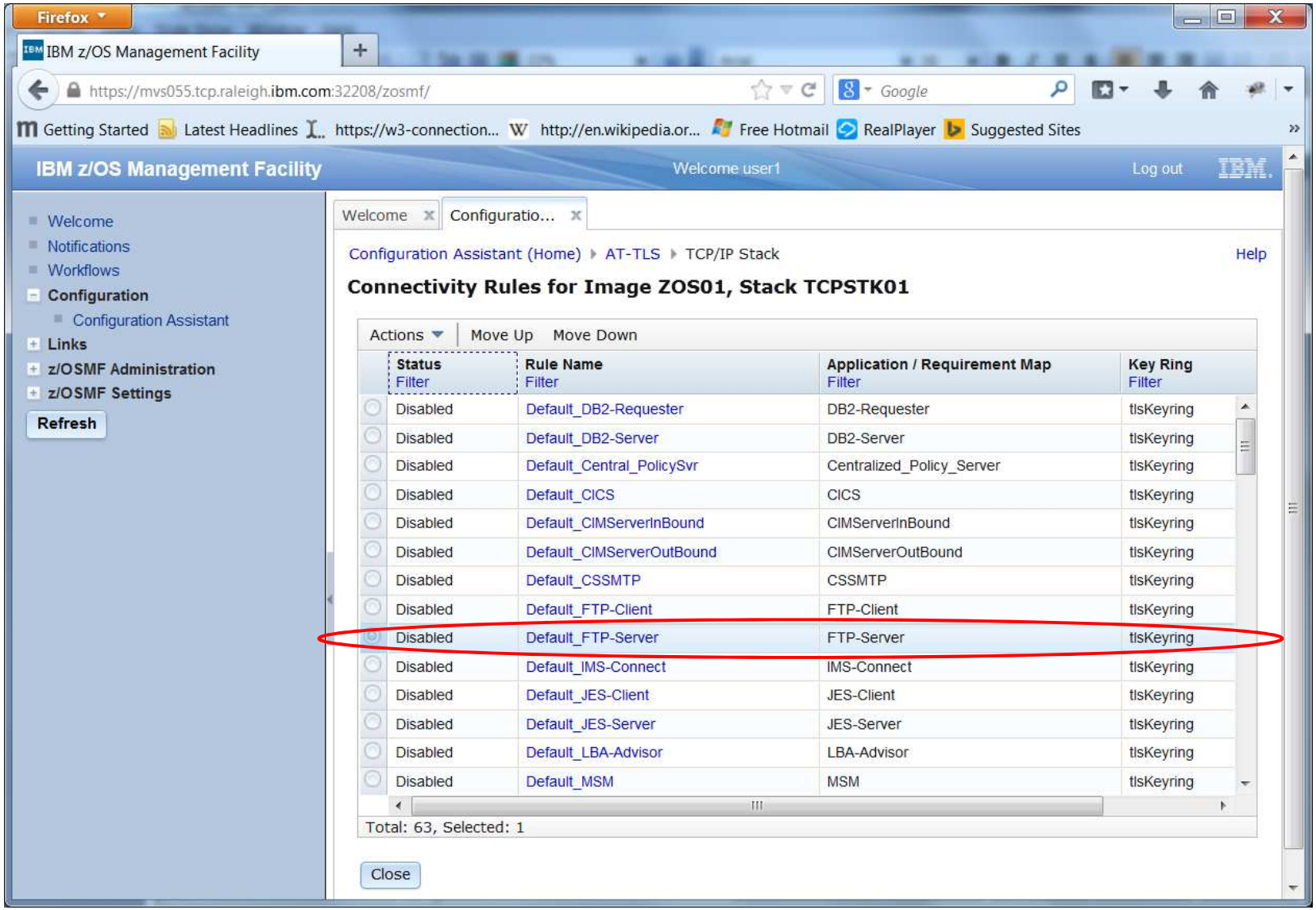


Policy-based network security on z/OS: Configuration Assistant

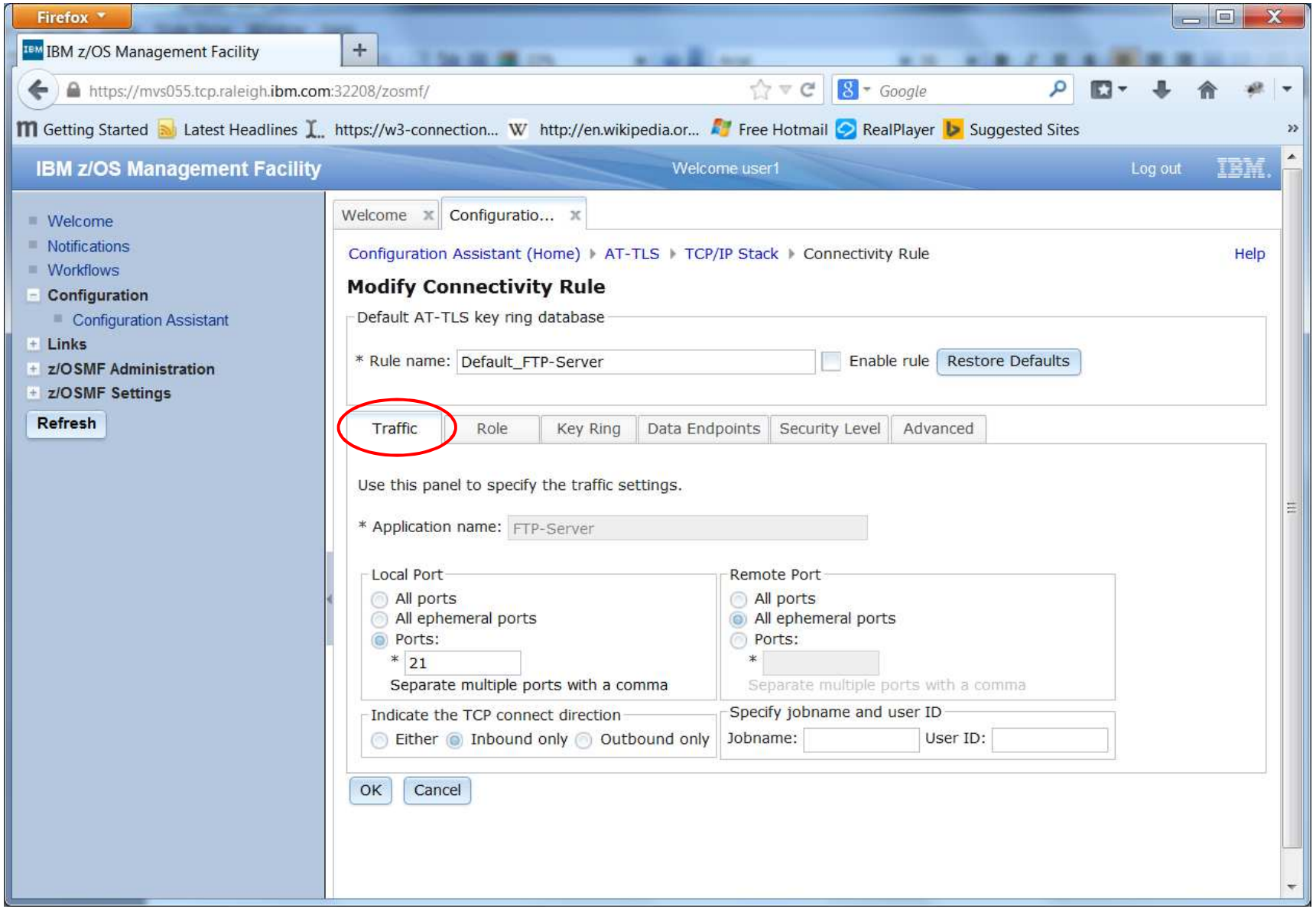


- **Configures:**
  - AT-TLS
  - IPSec and IP filtering
  - IDS
  - Quality of Service
  - Policy-based routing
- **Separate perspectives but consistent model for each discipline**
- **Focus on concepts, not details**
  - what traffic to protect
  - how to protect it
  - De-emphasize low-level details (though they are accessible through advanced panels)
- **z/OSMF-based web interface**
  - Standalone Windows application
    - Not supported after z/OS V1R13
- **Builds and maintains**
  - Policy files
  - Related configuration files
  - JCL procs and RACF directives
- **Supports import of existing policy files**

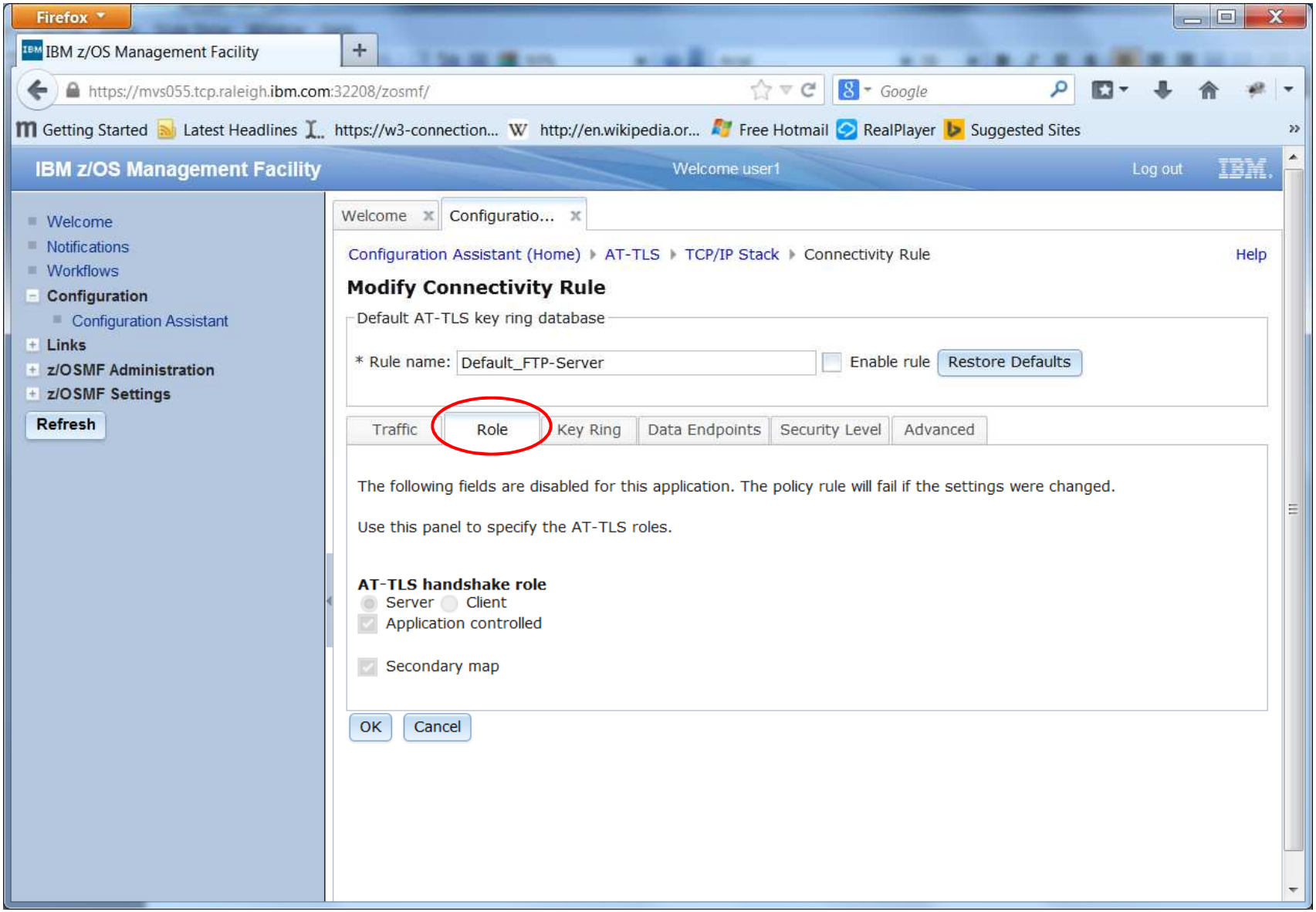
Examining the FTP server pre-defined connectivity rule



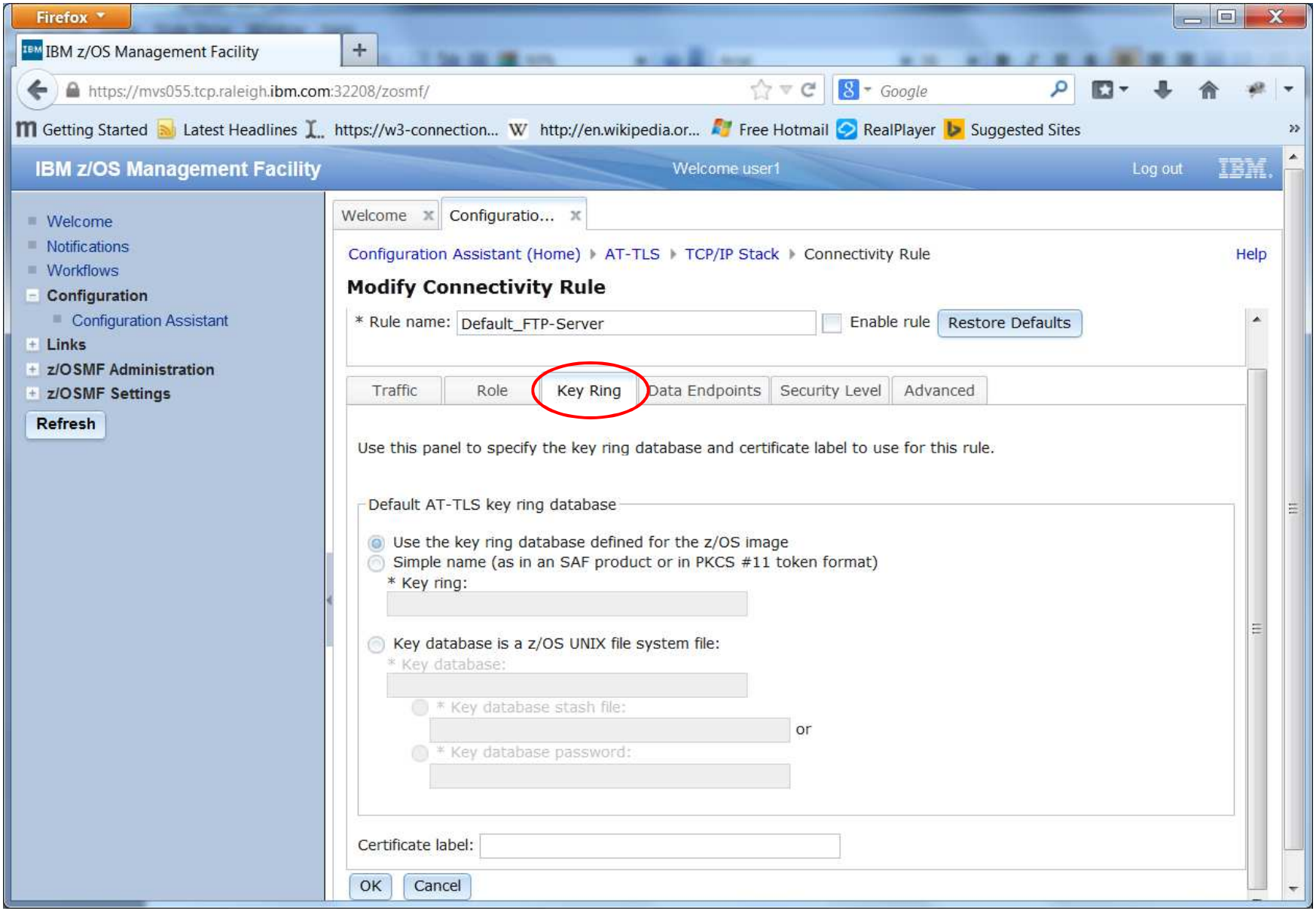
Describe traffic



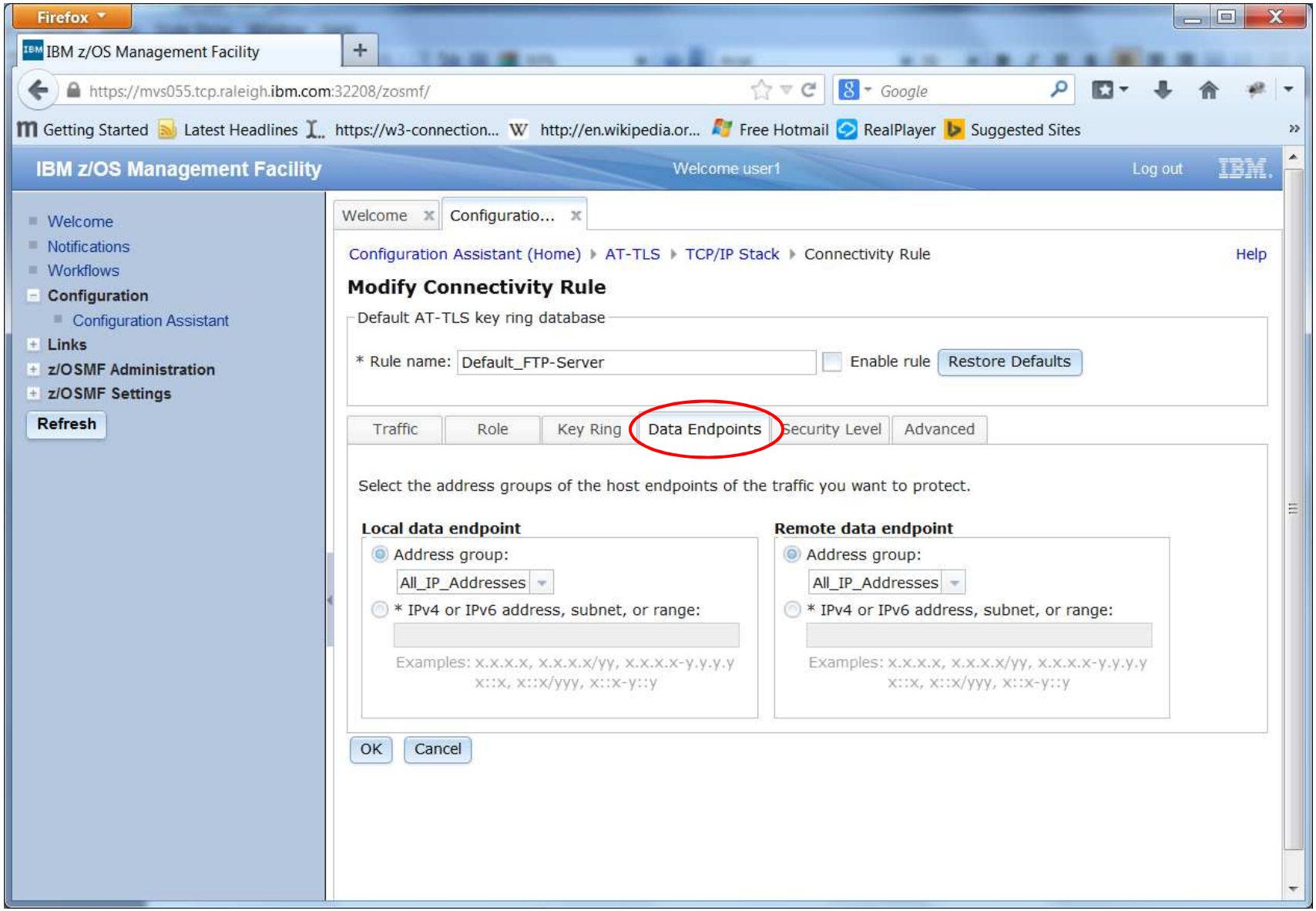
Describe role – Not changeable



Define key ring – in this case use the z/OS image level key ring



Describe data endpoints – in this case apply rule to all endpoints



For the latest news on z/OS Communications Server

Please fill out your session evaluation

Connecting CICS with TCP/IP

Session # 17255

QR Code:

Thank you!

Appendix – Backup

APPLDATA – CONNECT (Client socket in CICS)

Table 204. Registered application data - CONNECT

Bytes	Description
1-8	The component ID of the IP CICS socket interface. For an outbound IP CICS socket client, this data always comprises the characters EZACICSO.
9	Blank
10-13	The CICS/TS transaction identifier. This is the CICS/TS transaction ID that is assigned to the program that issued the CONNECT socket command.
14	Blank
15-21	The task number of the transaction identifier in bytes 10-13.
22	Blank
23-30	The user ID that is assigned to the transaction identifier in bytes 10-13.
31	Blank
32-35	The CICS system name where the transaction is running.
36-40	Blank

APPLDATA – GIVESOCKET (Socket given by listener to child server)

Table 205. Registered application data - GIVESOCKET

Bytes	Description
1-8	The component ID of the IP CICS Socket listener. For the IP CICS Sockets listener, this data always comprises the characters EZACIC02.
9	Blank
10-13	The CICS/TS transaction identifier. This is the transaction ID that the listener starts that the listener expects to take the specified socket.
14	Blank
15-21	This data is the task number of the CICS task that gives the accepted socket to a child process.
22	Blank
23-30	The user ID to be assigned to the transaction identifier in bytes 10-13.
31	Blank
32-35	The CICS system name where the transaction is to be assigned.
36-40	Blank

APPLDATA – TAKESOCKET (Socket taken by child server)

Table 207. TAKESOCKET

Bytes	Description
1-8	The component ID of the IP CICS Socket interface. For the IP CICS Sockets interface and listener, this data always comprises the characters EZACICSO.
9	Blank
10-13	The CICS/TS transaction identifier. This is the transaction ID that now owns the socket.
14	Blank
15-21	The task number of the transaction identifier in bytes 10-13.
22	Blank
23-30	The user ID that is assigned to the transaction identifier in bytes 10-13.
31	Blank
32-35	The CICS system name where the transaction is running.
36-40	Blank

APPLDATA – LISTEN (Listener socket)

Table 206. Registered application data - LISTEN

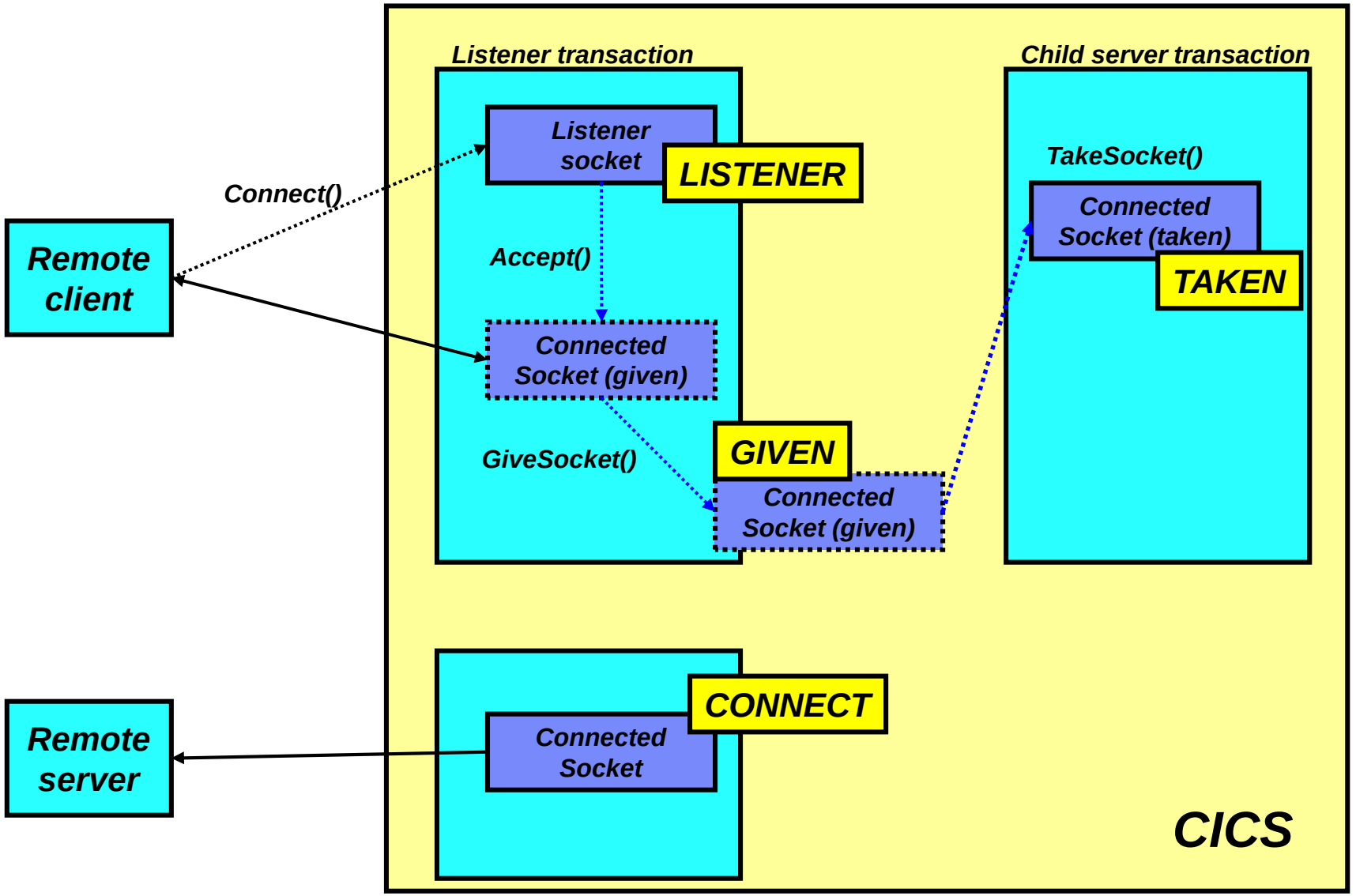
Bytes	Description
1-8	The component ID of the IP CICS socket interface. For the IP CICS sockets listener, this data always comprises the characters EZACICSO.
9	Blank
10-13	The CICS/TS transaction identifier. This is the CICS/TS transaction ID assigned to the EZACIC02 program or a user-designed listener transaction program.
14	Blank
15-21	The task number of the transaction identifier.
22	Blank
23-30	The user ID that is assigned to the transaction identifier in bytes 10-13.
31	Blank
32-35	The CICS system name where the transaction is executing.
36-40	Blank

What you can do with APPLDATA in Netstat – CICS Sockets

- APPLDATA is identification data a sockets application can associate with a sockets end point.
- CICS Sockets uses that feature to associate CICS-specific identification data with sockets that are used by the CICS Sockets.
- APPLDATA can be displayed with netstat, it is included in TCP/IP SMF records, and in the Network Management API.

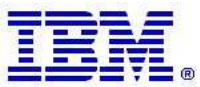
```
*-------------------- MVS TCP/IP NETSTAT CS z/OS V1R10 ----------------------*
Command ===>

Please enter optional selection criteria for CICS Sockets connection overview -
or press END to continue without any selection criteria.

   Remote IP address         ==>
   Local IP address          ==>
   CICS Sockets server port  ==>          CICS listener server port
   CICS address space name   ==>          CICS address space that owns socket
   CICS user ID              ==>          CICS assigned user ID
   CICS transaction code     ==>          CICS transaction identifier
   CICS task number          ==>          CICS internal task number
   CICS system name          ==>          CICS name transaction assigned to
   CICS Sockets type         ==>          Listener, Given, Taken, Connect

If you want a display of all your CICS Socket connections, leave all
selection fields above blank.
```

APPLDATA socket states



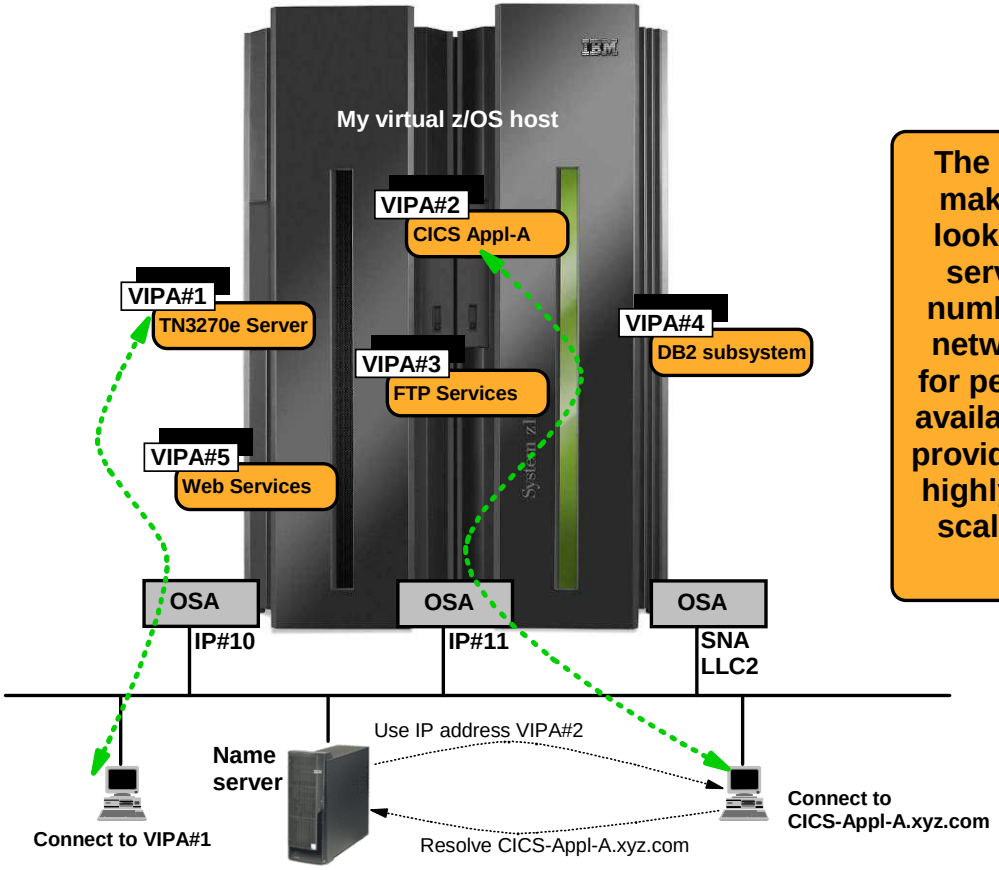
Connecting CICS with TCP/IP

CICS, TCP/IP and High Availability considerations



The network view of a Parallel Sysplex - a single large server with many network interfaces and many application services

- The promises of the Parallel Sysplex cluster environment are:
  - Application location independence
  - Ability to shift application workload between LPARs
  - Application single system image from the network
  - Application capacity on-demand
  - Component failure does not lead to application failure
- Gaining the benefits, depend on:
  - Carefully designed redundancy of all key hardware and software components in symmetric configurations
  - Supporting functions in z/OS and middleware
  - Cooperation by applications
  - Operations procedures



The objective is to make the Sysplex look like one large server that has a number of physical network interfaces for performance and availability - and that provides a number of highly available and scalable services.

SNA and TCP/IP

✓Single-system image (SSI)
✓Scalable
✓Highly available
✓Secure

A summary of the different types of z/OS VIPA addresses

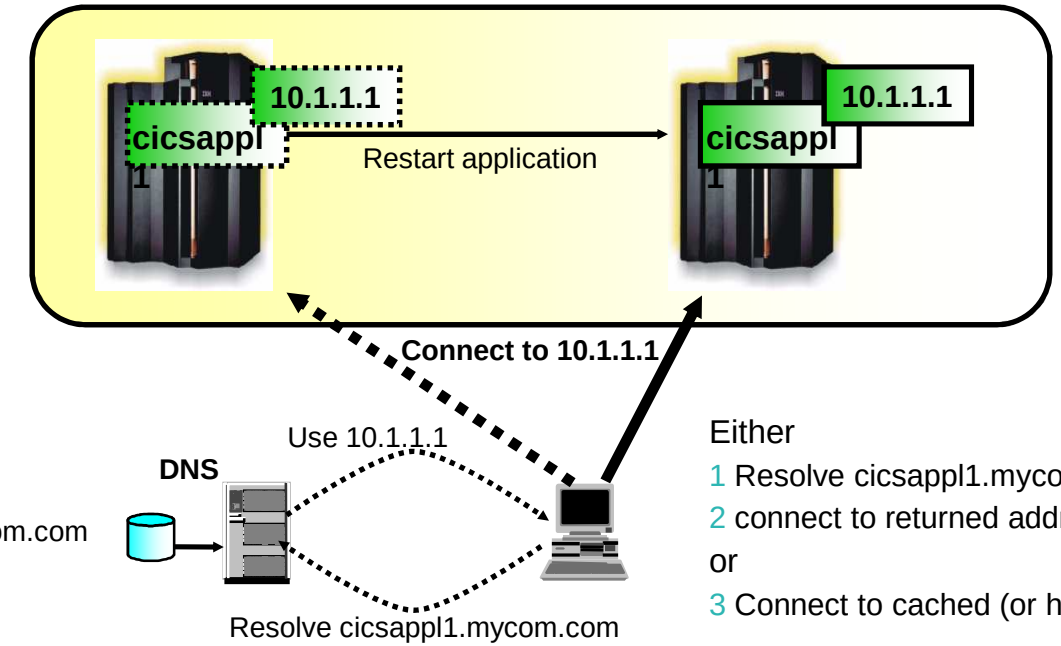
- **Static VIPA**
  - Belongs to one TCP/IP stack. Manual configuration changes are needed to move it.
    - No dependencies on Sysplex functions – can be used in non-Sysplex LPARs
    - Required for certain functions such as Enterprise Extender
    - Beneficial for interface resilience, source IP addressing, etc.

- **Dynamic VIPA (DVIPA)**
  - **Stack-managed (VIPADEFINE/VIPABACKUP)**
    - Belongs to one TCP/IP stack, but backup policies govern which TCP/IP stack in the Sysplex takes it over if the primary TCP/IP stack leaves the Sysplex
    - Individual stack-managed dynamic VIPAs can be moved between primary and backup stacks using MVS operator commands
  - **Application-specific also known as bind-activated (VIPARANGE)**
    - Belongs to an application. Becomes active on the TCP/IP stack in the Sysplex where the application is started. Moves with the application.
  - **Command- or utility activated (VIPARANGE)**
    - Belongs to whatever TCP/IP stack in the Sysplex on which a MODDVIPA utility to activate the address has been executed.
    - Moves between TCP/IP stacks based on execution of the MODDVIPA utility.
  - **Distributed also known as a DRVIPA or sometimes DDVIPA (VIPADEFINE/VIPABACKUP + VIPADISTRIBUTE)**
    - Used with Sysplex Distributor as a cluster IP address that represents a cluster of equal server instances in the Sysplex.
    - From a routing perspective it belongs to one TCP/IP stack.
    - From an application perspective it is distributed among the TCP/IP stacks in the Sysplex where an instance of the server application is executing.

Basic principles for recovery of single-instance IP application in a Sysplex

- Single-instance applications are applications that only run in one instance in the Sysplex. Either because the application needs exclusive access to certain resources, or because there is no need to start it in more than one instance.
- Availability from an IP perspective then becomes an issue of being able to restart the application on the same LPAR or on another LPAR with as little impact to end-users as possible.
- Speed of movement - ARM or automated operations procedures
- Retain identity from a network perspective (its IP address) - ***Application Instance DVIPAs***

New for z/OS V2R2 Up to 4096 Application Instance DVIPAs supported

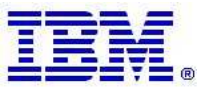


Application-specific dynamic VIPA addresses come in very handy for this purpose.

Either
1 Resolve cicsappl1.mycom.com
2 connect to returned address
or
3 Connect to cached (or hardcoded!) address

Connecting CICS with TCP/IP

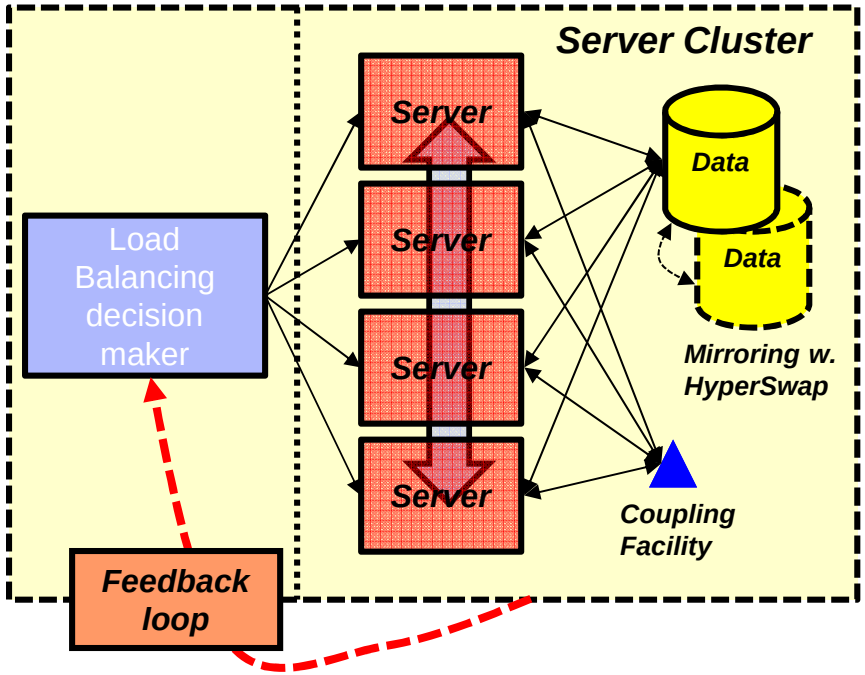
Workload Balancing Considerations



What are the main objectives of network workload balancing?

- **Performance**
  - Workload management across a cluster of server instances
  - One server instance on one hardware node may not be sufficient to handle all the workload
- **Availability**
  - As long as one server instance is up-and-running, the "service" is available
  - Individual server instances and associated hardware components may fail without impacting overall availability
- **Capacity management / horizontal growth**
  - Transparently add/remove server instances and/or hardware nodes to/from the pool of servers in the cluster
- **Single System Image**
  - Give users one target hostname to direct requests to
  - Number of and location of server instances is transparent to the user

All server instances must be able to provide the same basic service. In a z/OS Sysplex that means the applications must be Sysplex-enabled and be able to share data across all LPARs in the Sysplex.



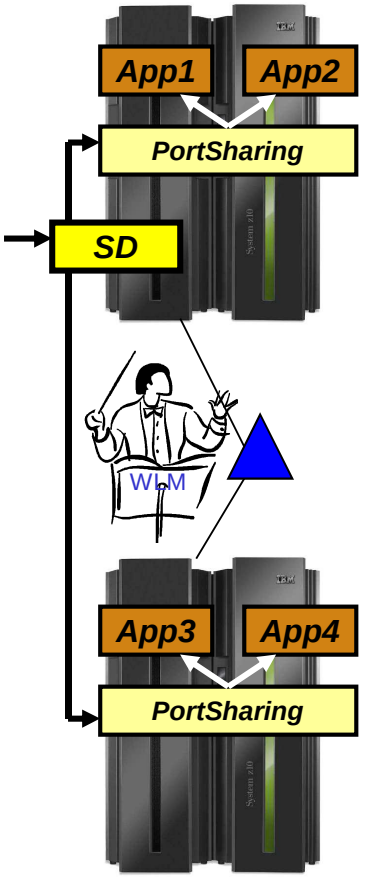
In order for the load balancing decision maker to meet those objectives, it must be capable of obtaining feedback dynamically, such as server instance availability, capacity, performance, and overall health.

z/OS IP network workload balancing overview

- Two main technologies:
  - Sysplex Distributor
  - Port sharing

- Sysplex Distributor
  - Sysplex Distributor is a layer-4 load balancer
    - It makes a decision when it sees an inbound SYN segment for one of the Distributed Dynamic VIPA (DDVIPA) IP address/port combinations it load balances for
  - Sysplex Distributor uses MAC-level forwarding when connection routing takes place over XCF
  - Sysplex Distributor uses GRE when connection routing takes place over any network between the z/OS images
    - Based on definition of VIPAROUTE
  - All inbound packets for a distributed connection must be routed through the Sysplex Distributor LPAR
    - Only the Sysplex Distributor LPAR advertises routing ownership for a DDVIPA, so downstream routers will forward all inbound packets for a given DDVIPA to the distributing LPAR
  - All outbound packets from the server instances can take whatever route is most optimal from the server instance node back to the client

- Port sharing
  - PORTSHARING can be used within a z/OS node to distribute connections among multiple server address spaces within that z/OS node
    - SHAREPORT – TCP/IP Server Efficiency Factor (SEF) value used to perform a weighted round robin distribution to the server instances
    - SHAREPORTWLM – WLM input is used to select server for new connection



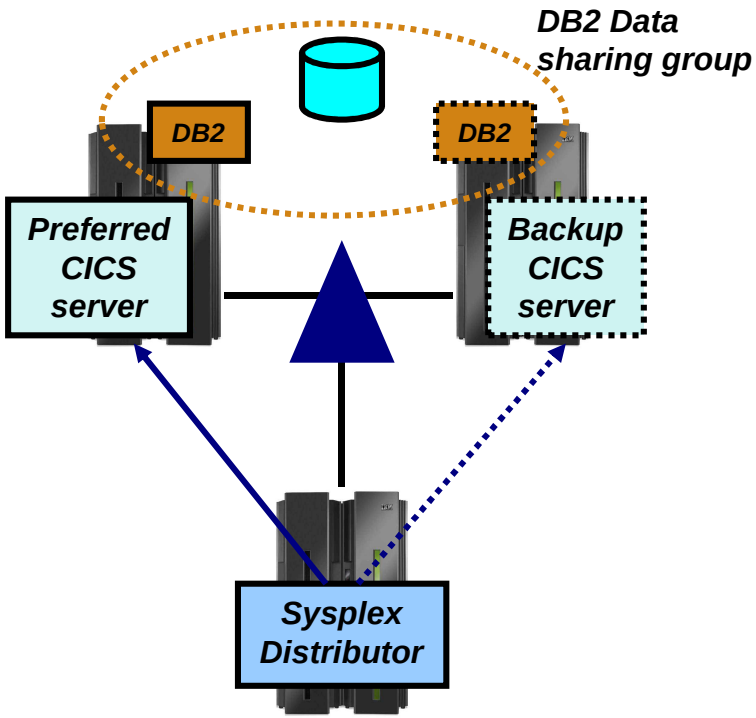
Sysplex Distributor distribution method overview

- z/OS targets without WLM recommendations
  - ROUNDROBIN
    - Static distribution of incoming connections, does not account for target system capacity to absorb new workload
  - WEIGHTEDACTIVE
    - Incoming connections are distributed so the available server instances' percentage of active connections match specified weights

- z/OS targets with WLM recommendations
  - BASEWLM
    - Based on LPAR level CPU capacity/availability and workload importance levels
  - SERVERWLM
    - Similar to BASEWLM but takes into account WLM service class and how well individual application servers are performing (i.e. meeting specified WLM goals) and how much CPU capacity is available for the specific workload being load balanced
    - Enhanced to account for WLM provided server health
    - Supports autonomic TCP/IP health detection metrics
    - ***Generally, the recommended distribution method for Sysplex Distributor***

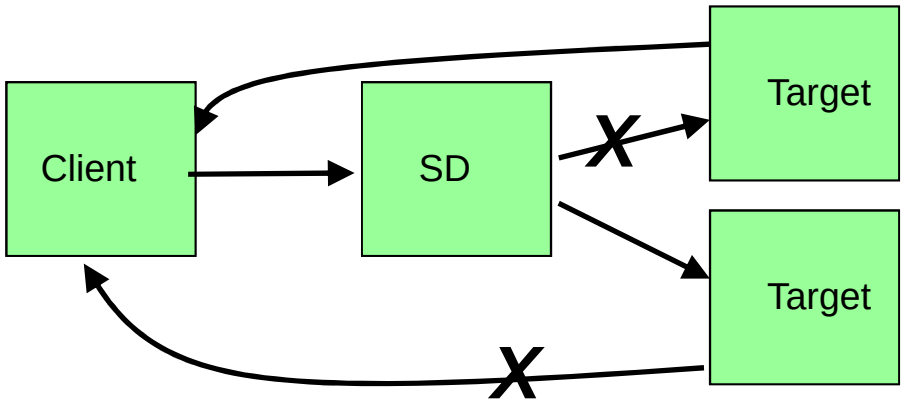
Sysplex Distributor distribution method overview …

- HOTSTANDBY
  - Incoming connections are distributed to a primary server instance and only rerouted to a backup server instance (the “hot standby”) when the primary server instance is not ready, unreachable, or unhealthy.
  - Method added in ***z/OS V1R12***



Sysplex Distributor built-in awareness of abnormal conditions

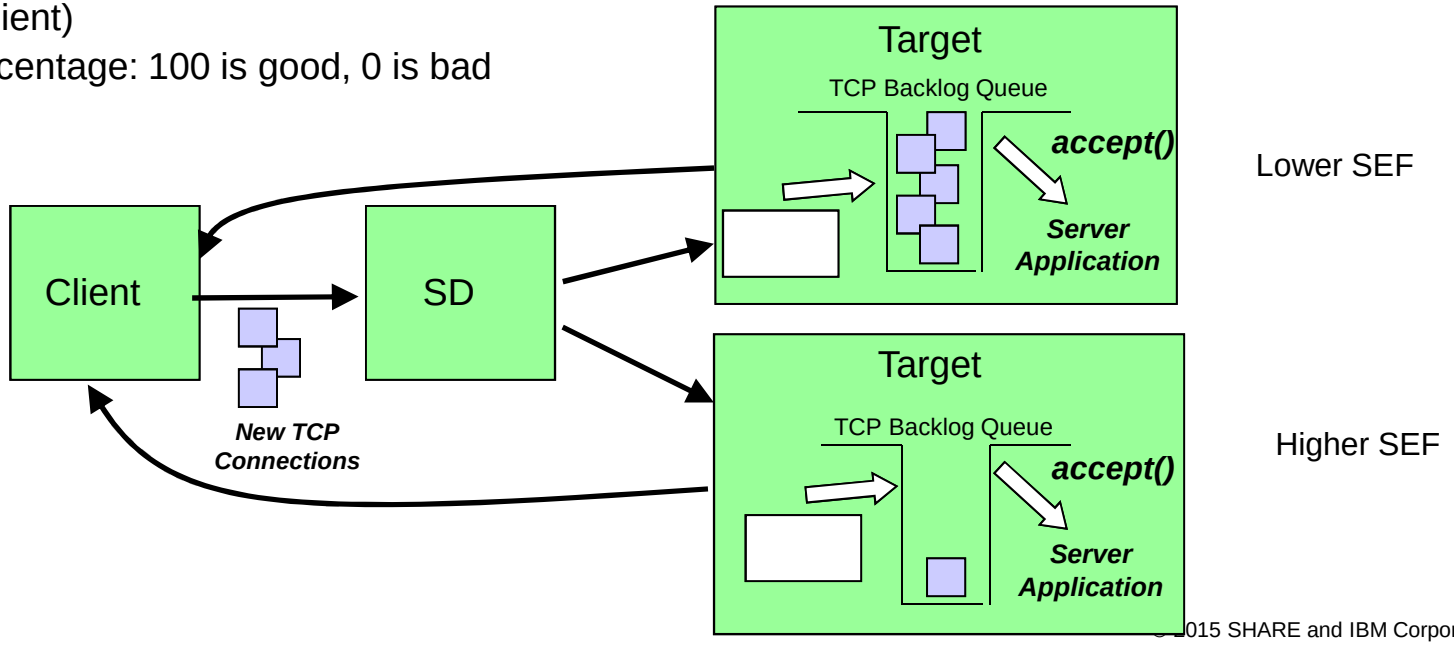
- TSR – Target Server Responsiveness
  - How healthy is the target system and application from an SD perspective? A percentage, 0-100%
  - Comprised of several individual health metrics:
    - TCSR – Target Connectivity Success Rate
      - Are connections being sent to the Target System making it there?
      - A Percentage: 100 is good, 0 is bad



    - CER – Connectivity Establishment Rate
      - Is connectivity between the target system and the client ok?
      - By monitoring TCP Connection Establishment state (requires 3 way handshake between client and server) we can detect whether a connectivity issue exists
      - A percentage: 100 is good, 0 is bad
      - Note: CER no longer part of TSR directly but is included in SEF and continues to be calculated and reported separately

Sysplex Distributor built-in awareness of abnormal conditions

- TSR – Target Server Responsiveness (cont)
  - SEF – Server Efficiency Fraction
    - Is the target server application server keeping up with new connections in its backlog queue?
      - > Is the new connection arrival rate higher than the application accept rate? (i.e. is backlog growing over time)
      - > How many connections in the TCP backlog queue? How close to maximum backlog queue depth? Did we have to drop any new connections because the backlog queue max was exceeded?
      - > Is the server application hung? (i.e. not accepting any connections)
      - > Are the number of half-open connections on the backlog queue growing? (Similar to CER – One such scenario is when the target system does not have network connectivity to the client)
    - A Percentage: 100 is good, 0 is bad



Middleware/Application Issues and the "Storm Drain Problem"

- TCP/IP and WLM are not aware of all problems experienced by load balancing targets (middleware/applications) – Examples:
  - The server application needs a resource such as a database, but the resource is unavailable
  - The server application is failing most of the transactions routed to it because of internal processing problems
  - The server application acts as a transaction router for other back-end applications on other system(s), but the path to the back-end application is unavailable
- In each of these scenarios, the server may appear to be completing the transactions quickly (using little CPU capacity) when they are actually being failed
- This is sometimes referred to as the *Storm Drain* Problem
  - The server is favored by WLM since it is using very little CPU capacity
  - As workloads increase, the server is favored more and more over other servers
  - All this work goes "down the drain"

Improving WLM awareness of Application Health - Avoiding "Storm Drain" Issues

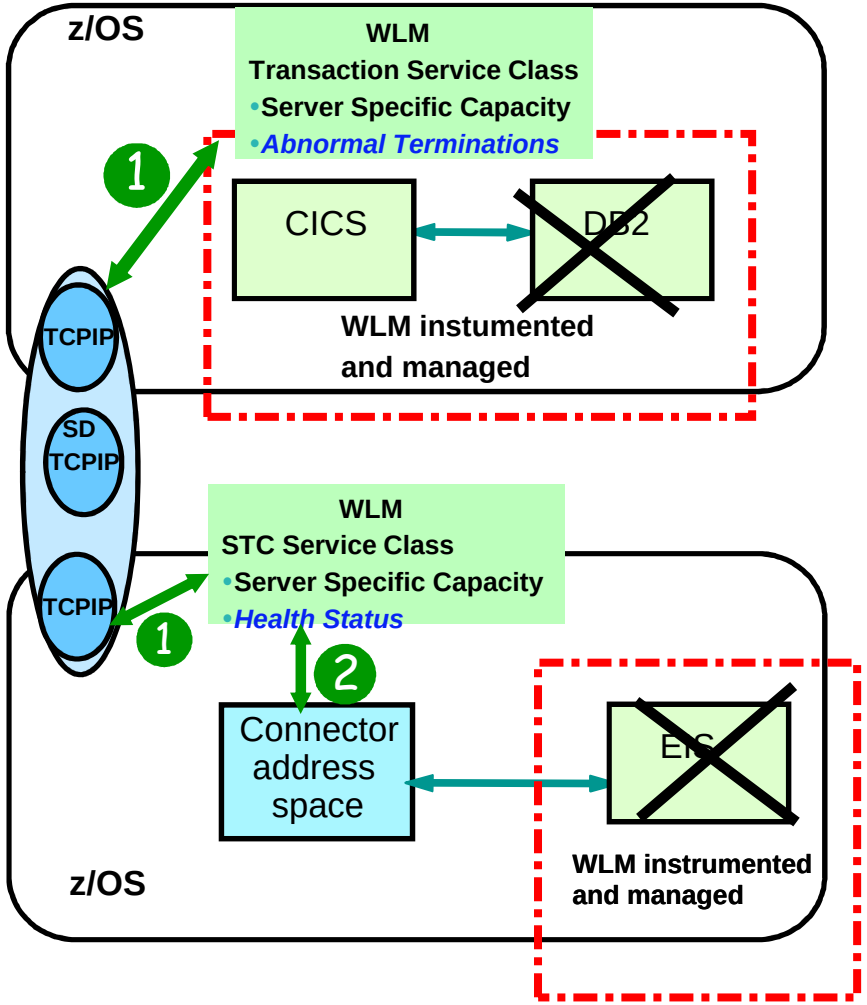
Server Scenarios

1 IWM4SRSC WLM Service

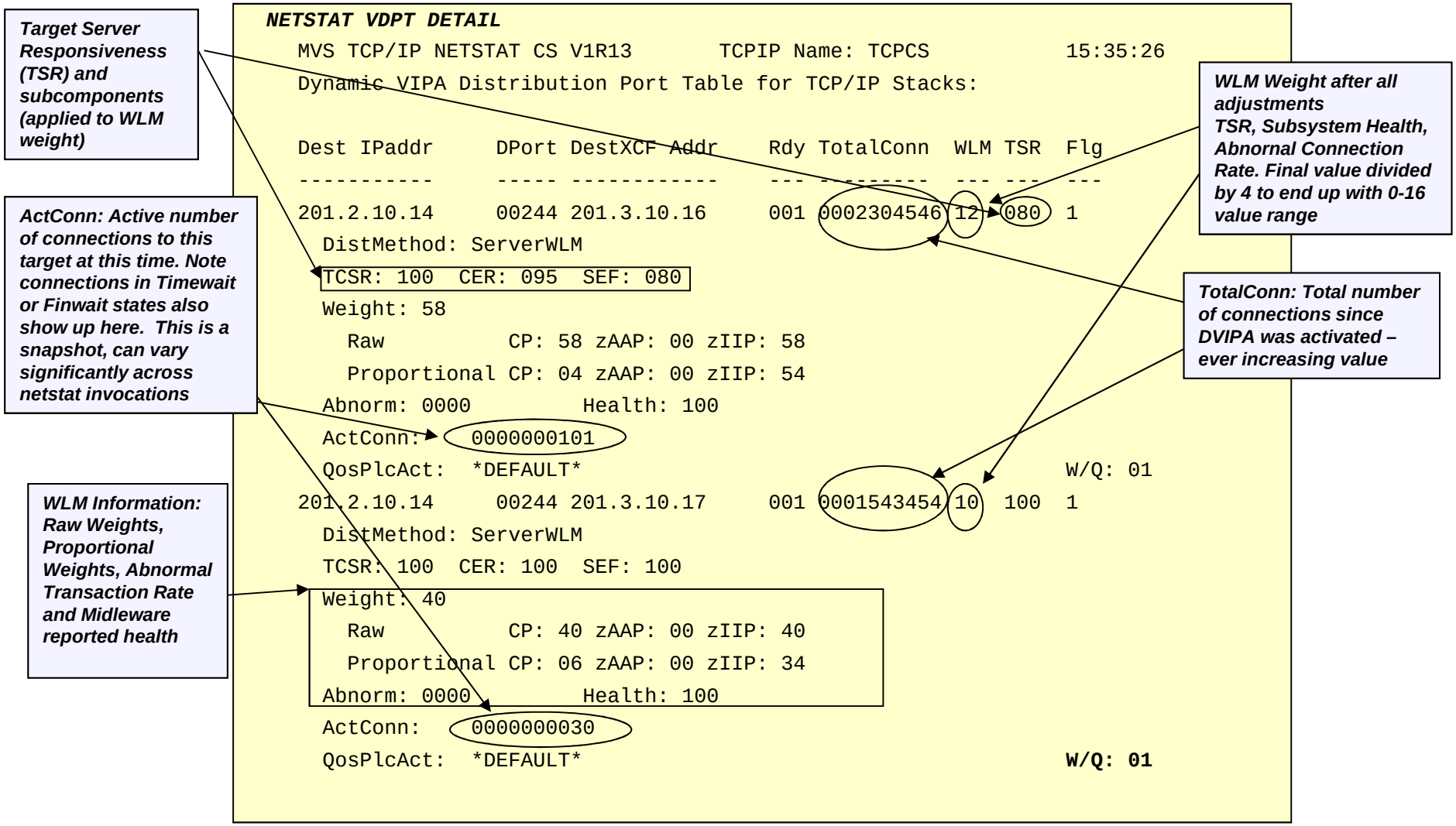
- Used by Sysplex Distributor to obtain WLM recommendations
- Abnormal Termination information: Reported by 1st tier server when transactions can not complete because back end resource managers are not available
  - WLM uses this information to reduce the recommendation for ailing server

2 IWM4HLTH WLM Service

- Allows address spaces which are not instrumented with WLM to set a a health status which is also returned by IWM4SRSC
- The ServerWLM recommendations are reduced when the health is <100%
- Exploited by CICS Transaction Gateway, DB2 and LDAP



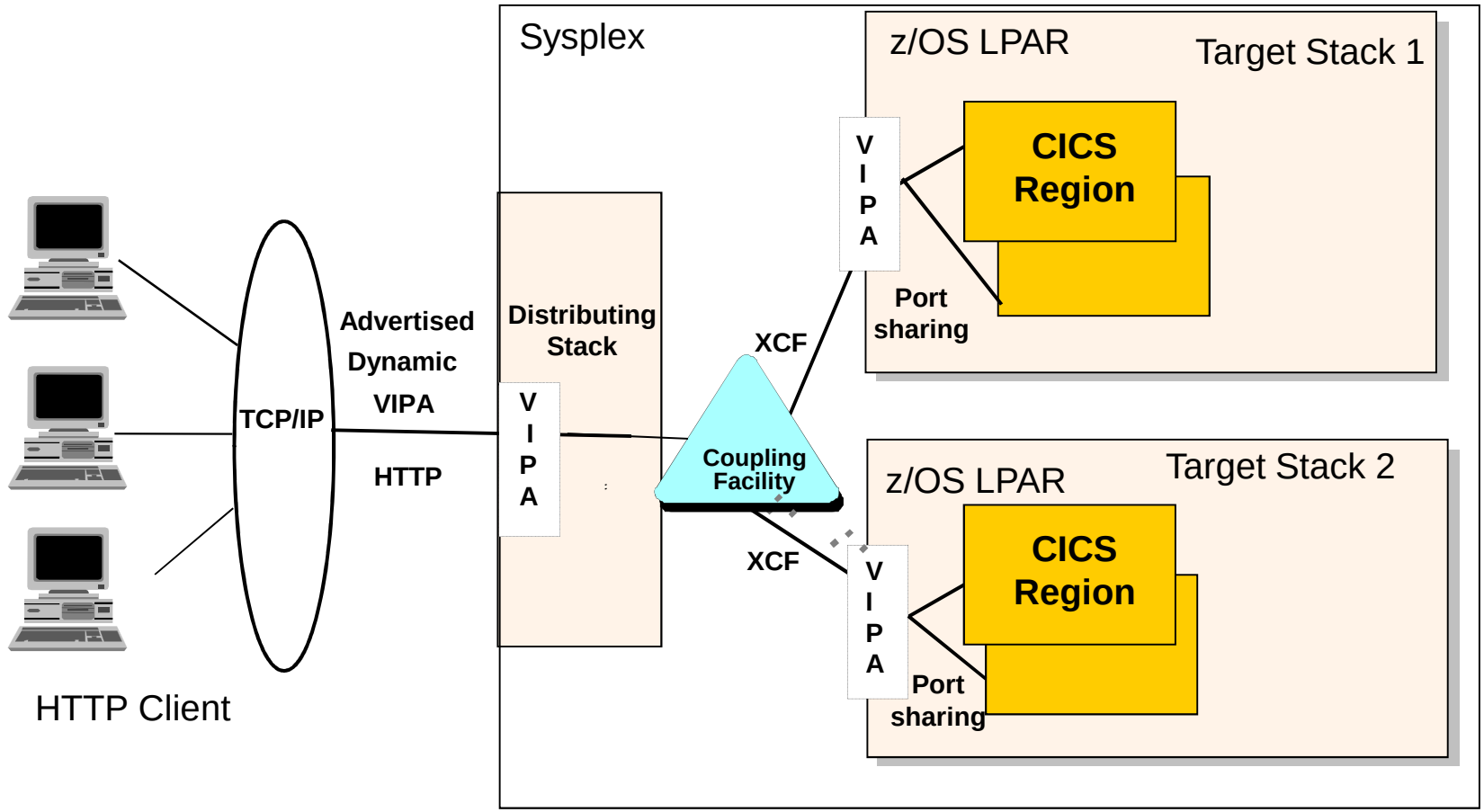
Using Netstat VDPT Detail display to monitor Sysplex Distributor



What impacts the final selection of a target server instance?

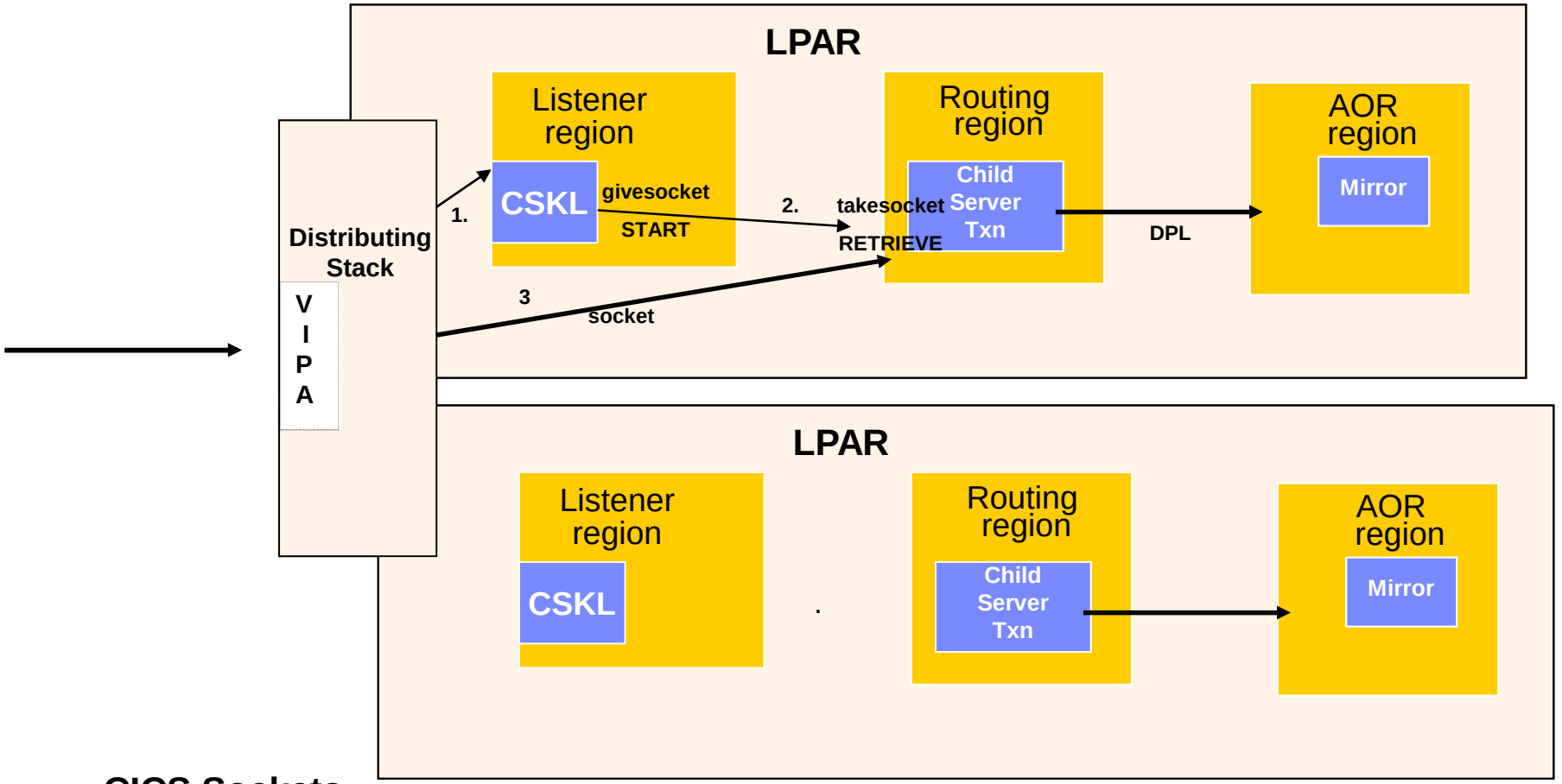
Technology	Target LPAR displaceable capacity as seen by WLM	Server instance performance as seen by WLM	Server instance self-perceived health (as reported to WLM)	Server instance TCP/IP perceived health (the TSR value)	QoS perceived network performance (the QoS fraction)
SD ROUNDROBIN	No	No	No	Yes (if TSR=zero)	No
SD WEIGHTEDACTIVE	No	No	Yes	Yes	No
SD BASEWLM	Yes	No	No	Yes	Yes
SD SERVERWLM	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
SD TARGETCONTROLLED	Yes (SD agent)	No	No	No	No
SD HOTSTANDBY	No	No	Yes	Yes	No
PORT SHAREPORT	No	No	No	Yes (Only SEF value)	No
PORT SHAREPORTWLM	No	Yes	Yes	Yes (Only SEF value)	No

CICS Web support - HTTP



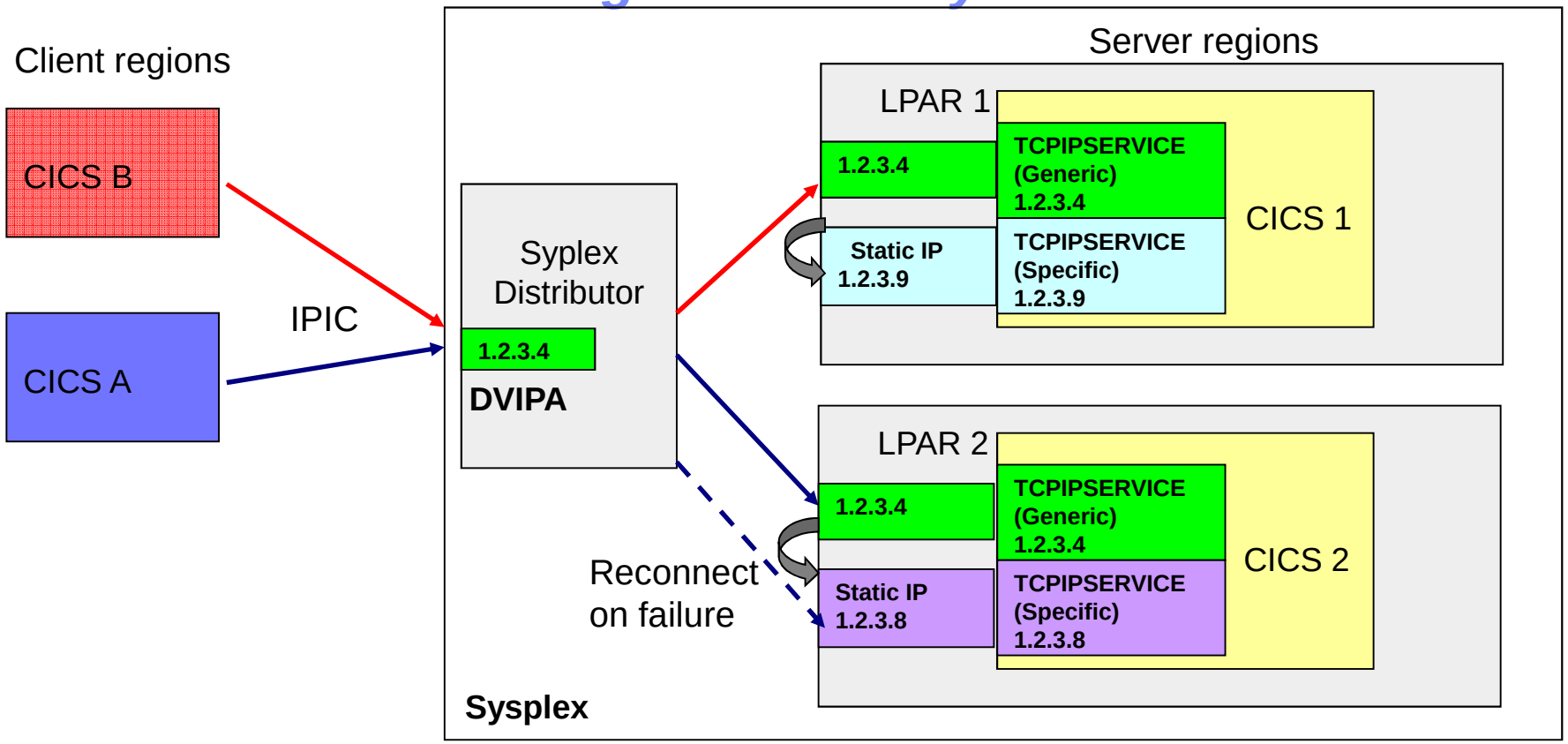
- HTTP requests used for HTML and SOAP requests to CICS
- HTTP 1.0 (with Keep Alives) and HTTP 1.1 supported
- Can be used with either TCP/IP port sharing or Sysplex Distributor
- Requires that any session data is in shared storage (i.e. RLS or shared TS)

CICS Sockets



- **CICS Sockets**
  - Child server transaction can be defined in a remote CICS region
  - CICS dynamic routing can be used to route remote START to AOR
  - Routing region must be on the same LPAR and share the same TCP/IP stack
  - Can exploit TCP/IP port sharing or Sysplex Distributor

CICS TS V5.2 – IPIC High Availability



- CICS server regions listen on a generic and a specific TCPIPService
- Client region reconnects to specific TCPIPService if connection terminated leaving UOW affinities
- Supports Sysplex Distributor DVIPAs and Port Sharing