CICS High Availability (HA), Continuous operation and Disaster Recovery (DR) overview

Arndt Eade (Arndt.eade@uk.ibm.com)
Catherine Moxey (Catherine_moxey@uk.ibm.com)

Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Terminology

- High availability
  - Minimizing the number of unplanned outages and the impact of unplanned outages. Unplanned outages are service interruptions caused by unexpected events such as hardware failures, software failures, and accidents.
- Continuous operation
  - Minimizing the number of planned outages and the impact of planned outages. Planned outages are service interruptions required for expected events such as running batch processes, upgrading hardware or software, and applying preventive maintenance.
- Disaster recovery
  - Capability to resume operations following a disaster in a manner which minimizes the time to recovery and the loss of data.

Agenda

- CICS High Availability
  - Overview and Objectives
  - Parallel SYSPLEX overview and options
  - Routing work to CICS
  - Routing work within CICS
  - Accessing data
  - Application affinities and Single points of failure (SPOF)
  - Tooling
- CICS continuous operation
  - Overview and objectives
- CICS Disaster Recovery
  - Overview and objectives

CICS high availability (HA)

Overview and objectives

- Highly availability objectives
  - Minimizes number and impact of unplanned outages
  - Minimizes number and impact of planned outage
  - Maximise availability of CICS applications
  - Ensure CICS application availability meets/exceeds business SLA requirements
- Components of highly available systems
  - Reliable components
  - Spare capacity
  - Operational processes, procedures, technologies
- Holistic approach
  - You need to look at the environment as a whole
  - End to end HA with above characteristics throughout

Reliable components

- Hardware
  - IBM servers excellent reliability record
  - Built in problem detection and automated failover / RAID devices
- z/OS operating system
  - RAS characteristics built into core design
- Sysplex
  - Backbone of Highly Available infrastructure
- CICS
  - Premier Transaction processing system on platform
  - Reliability, availability, scalability designed into product
- Data
  - DB2, VSAM, MQ
- Security
  - Protecting systems / data from unauthorised access
- Applications
  - Good design is key

Spare capacity

- IBM provides on demand capacity
- But need to consider CICS infrastructure
  - What happens if you lose an LPAR
    - Can other LPARs absorb capacity?
  - What Happens if you lose a CF / CF LINK
    - System managed duplexing is the CICS recommended best practice for CICS structures
  - What happens if you lose a CICS TOR / AOR
    - LPAR and SYSPLEX level capacity to absorb work from lost TOR / AOR
    - Single points of failure?
  - What happens if I lose a DB2 or MQ
    - Do you have DSG / QSG with multiple instances?
    - What impact will failover having on remaining system?

Operational processes, procedures, …

- Its not just about having the infrastructure
- Well versed operational procedures are key to a good HA story
  - Automation
  - Change control
  - Monitoring
  - Capacity planning
  - Maintenance
  - Security
- When did you last review these?
- DevOps(*) adoption helps this

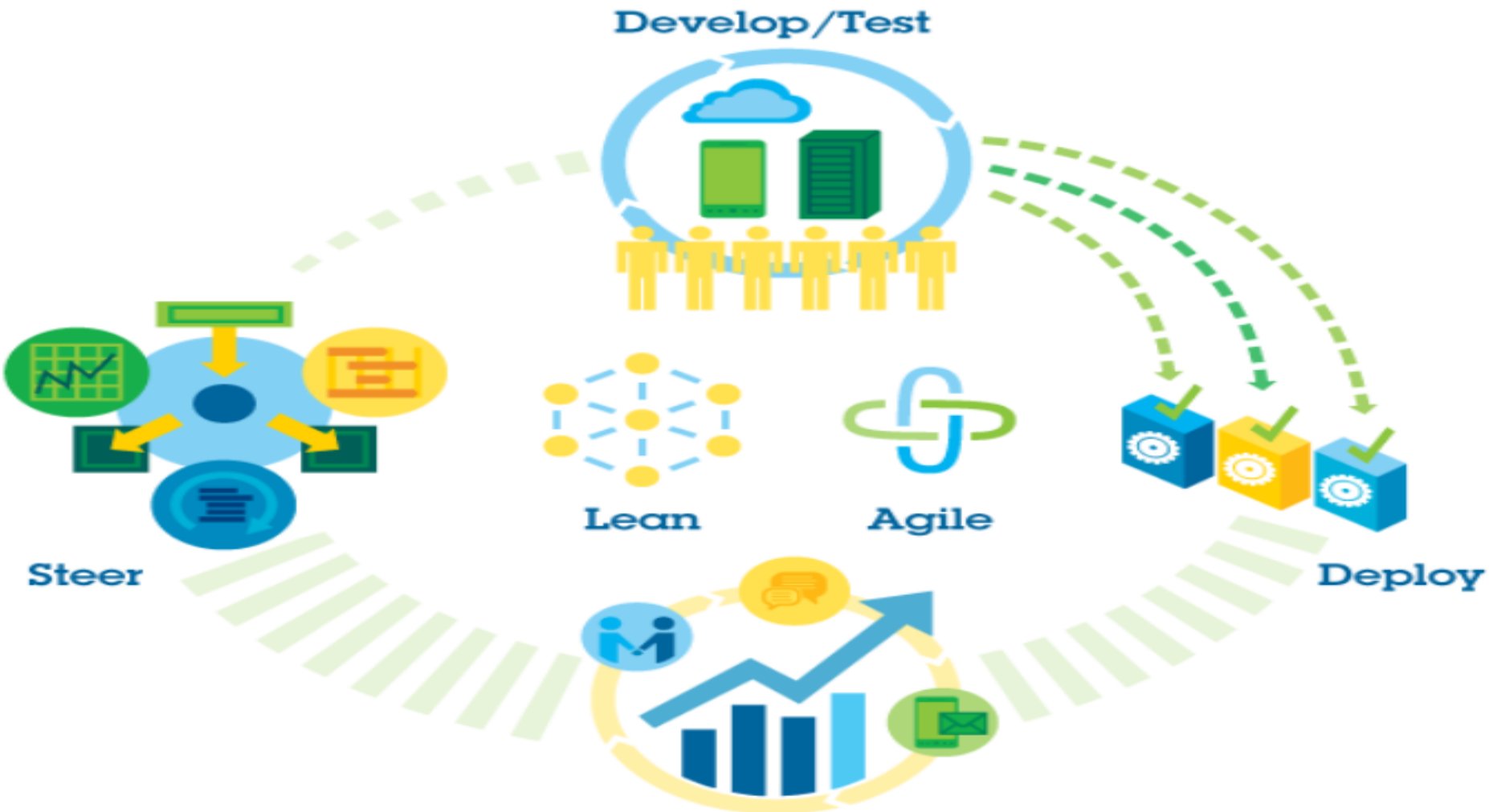
Integrate DevOps into your software lifecycle

Steer
Alignment of software portfolio decisions with business goals.

Develop/test
Innovation of ideas through collaborative development and testing.

Deploy
Delivery of innovation by automating processes and eliminating waste.

Operate
Improvement via feedback from internal and external customers.



Parallel Sysplex overview and options

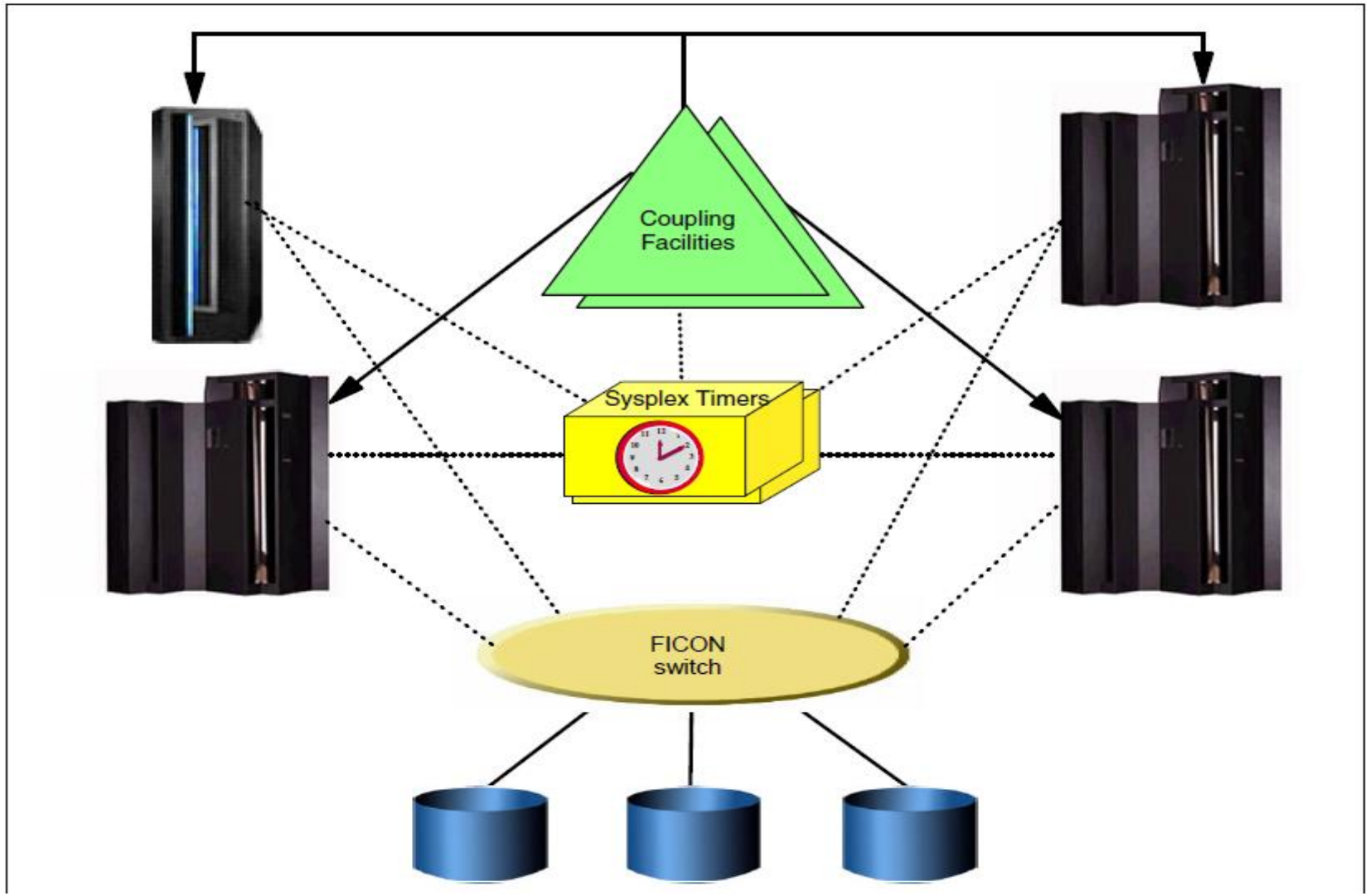
- What is a sysplex
  - Overview
  - Value

- Key components for HA
  - VIPA
  - Sysplex Distributor
  - Port Sharing
  - DNS Connection optimization
  - VTAM Generic resources
  - z/OS WLM
  - CF

What is a Parallel SYSPLEX

- Up to 32 z/OS systems connected to behave as a single, logical computing facility
  - Base Sysplex – “Systems Complex” – has been around since 1990.
  - Parallel Sysplex introduced in 1994 with addition of Coupling Facility (CF)
- Underlying structure remains virtually transparent to user, networks, application and even operations
- Primary objective is to provide data sharing capabilities. Benefits of which are:
  - Reduction/removal of Single points of failure within server, LPAR, subsystems
  - Application availability
  - Single systems image
  - Dynamic session balancing
  - Dynamic transaction routing
  - Scalable capacity

IBM Parallel SYSPLEX



Value of the Parallel Sysplex

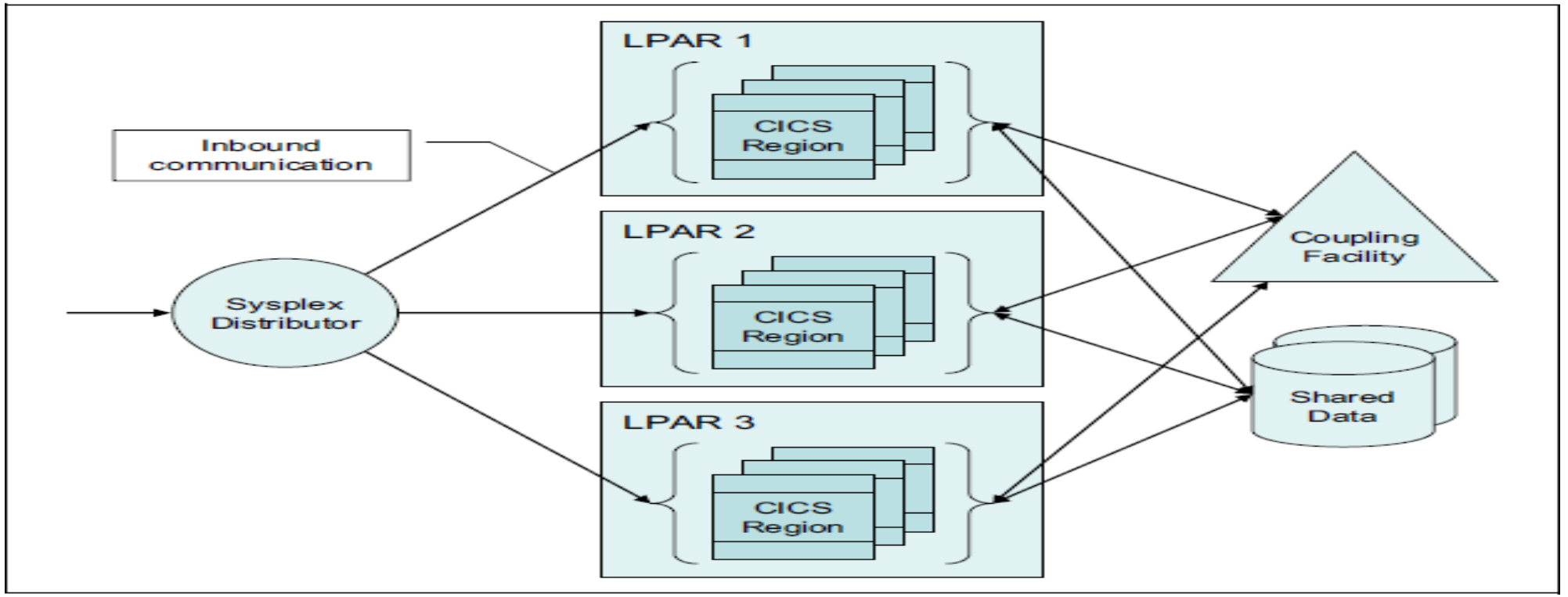
- Continuous availability of applications
- Reduction or elimination of planned application outages
- Scalability to virtually unlimited capacity to meet the high transaction volumes and response times of today and tomorrow
- Investment protection of existing applications by providing functions that allow these applications to operate in the e-business environment without a complete rewrite.
- A secure environment for existing and e-business transactions
- A development environment that provides the tools and languages to develop new applications for today and tomorrow
- A platform for server consolidation, to reduce the cost and complexity of having a large server farm to manage
- A simple growth path delivered with low incremental and total cost of computing

Key components – VIPA (Virtual IP Addressing)

VIPA (Virtual IP Addressing):

- Traditionally IP address associated with physical link.
- While failures in intermediate links in network can be rerouted the endpoints are points of failure.
- VIPA Introduced to provide fault-tolerant network connections to a TCP/IP for z/OS system stack.
- Allows installation to define virtual interface that is not associated with hardware components.
  - and thus cannot fail
  - The IP address that is defined for a virtual interface is always available.
- To the routing network VIPA appears to be host address indirectly attached to z/OS.
- Name servers configured to return VIPA of TCP/IP stack - not physical interface. If physical interface fails, dynamic route updates are sent out to update IP routing tables to use alternate path.
- IP connections are not broken but non-disruptively recovered through remaining physical interfaces.

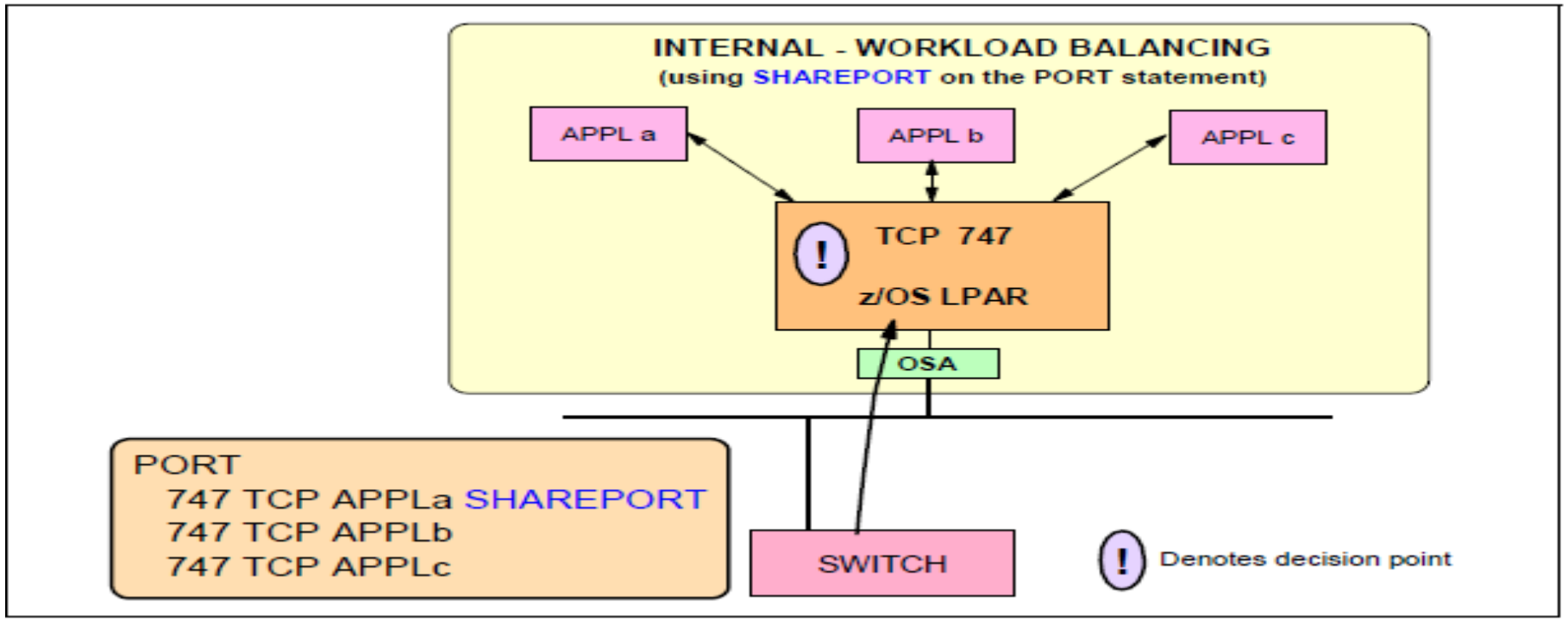
Key components - SYSPLEX Distributor



Distributed DVIPA (sysplex Distributor) :

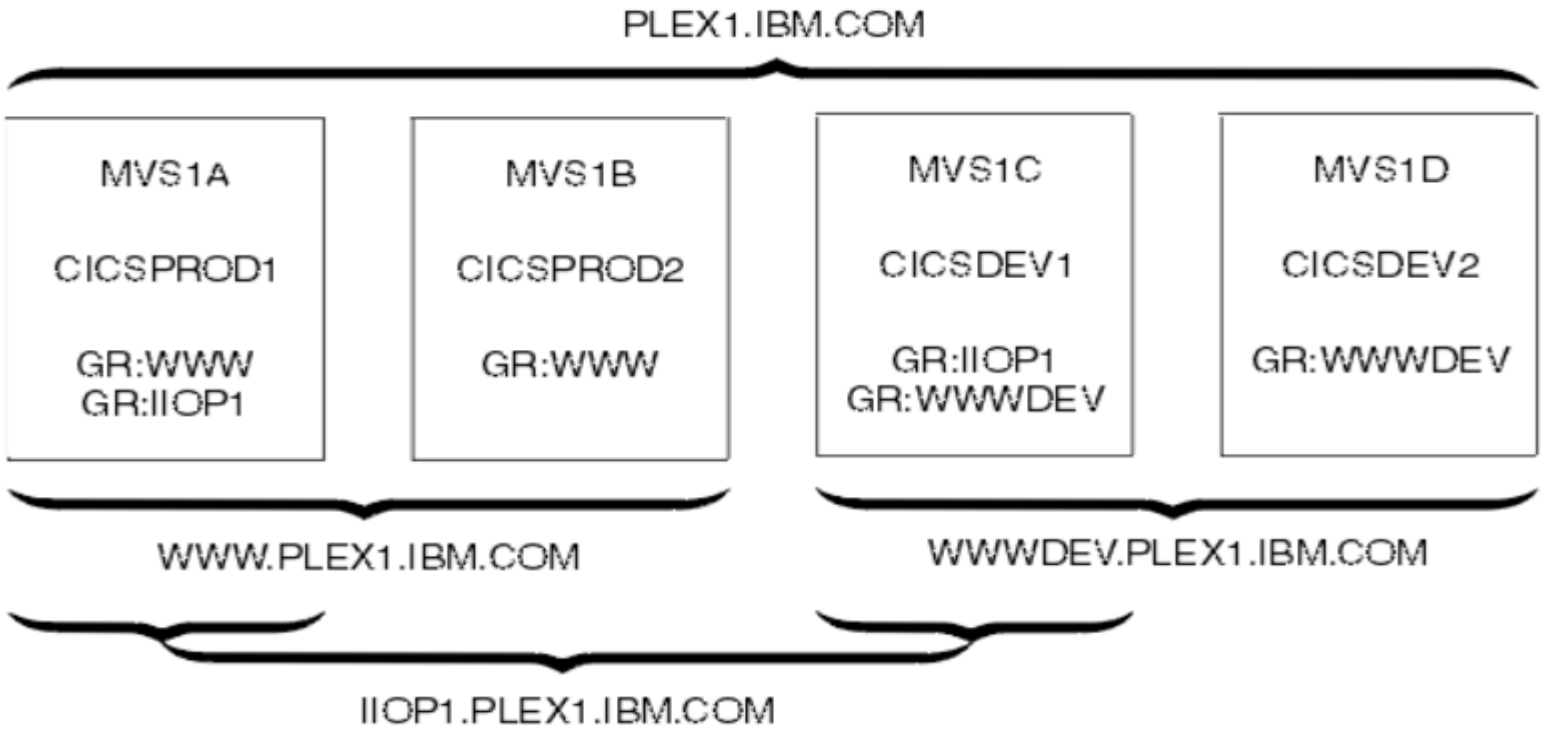
- allows an IP workload to be distributed to multiple server instances within the sysplex without requiring changes to clients or networking hardware andwithout delays in connection setup.
- allows you to implement a dynamic VIPA as a single network-visible IP address that is used for a set of hosts belonging to the same sysplex cluster.
- A client on the IP network sees the sysplex cluster as one IP address, regardless of the number of hosts in the cluster.
- Receive benefits of workload distribution provided by WLM and QOS policy agent.
- Ensure HA by providing continued operations in event of LAN failure, loss of entire IP stack or single z/OS image

Key components – Port Sharing



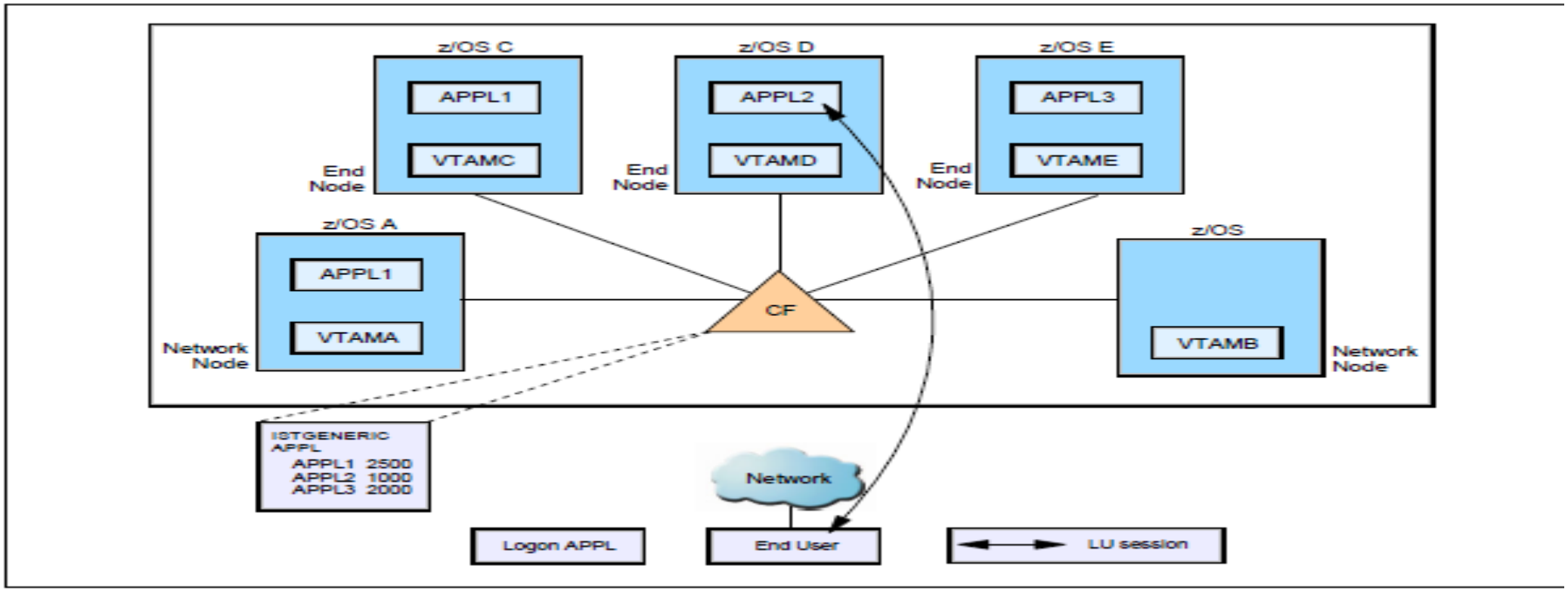
- Port sharing is a method to distribute workload for IP applications within a z/OS LPAR. TCP/IP allows multiple listeners to listen on the same combination of port and interface.
- Workload destined for this application can be distributed among the group of servers that listen on the same port. Port sharing does not rely on an active sysplex distributor implementation; it works without sysplex distributor.
- However, you can use port sharing in addition to sysplex distributor operation.
- Two flavours: SHAREPORT and SHAREPORTWLM

Key components – DNS connection optimization



- Connection optimization is a mechanism to balance IP connections and workloads in a sysplex domain.
- In DNS terms a sysplex is a subdomain that you add to you DNS name space.
- DNS optimization extends the concept of a “DNS Host name” to cluster or groups of server hosts or applications.
- Server applications within same group are considered to provide same service.
- Connection optimization used load-based ordering to determine which addresses to return for a given cluster.
- Servers register to WLM
- Generic host name decouples client application from specific host or CICS region

Key components – VTAM Generic Resources



- VTAM Generic Resources allows SNA applications to easily connect to application that may be available on multiple systems within a sysplex.
- Provides a single system image of application no matter where it runs in the sysplex
- User accesses application using the generic resource name of the application
- VTAM determines actual application instance based on performance and workload criteria
- Allows application instances to be added, removed and moved without impacting the user
- Supports LU types - LU0, LU1, LU2, LU3, LU6, LU6.1 and LU6.2

Key components – z/OS WLM

- Key strength of System z platform and z/OS operating system is ability to run multiple workloads at the same time within one image or across multiple images
- The Key function that makes this possible is z/OS Workload management (WLM)
  - Not to be confused with CICS WLM which we cover later
- Classification of work into Service classes which define expectations of how work should perform
- Using these goals WLM manages the work across all systems in the sysplex
- Provides information to components mentioned earlier as the where to best route work

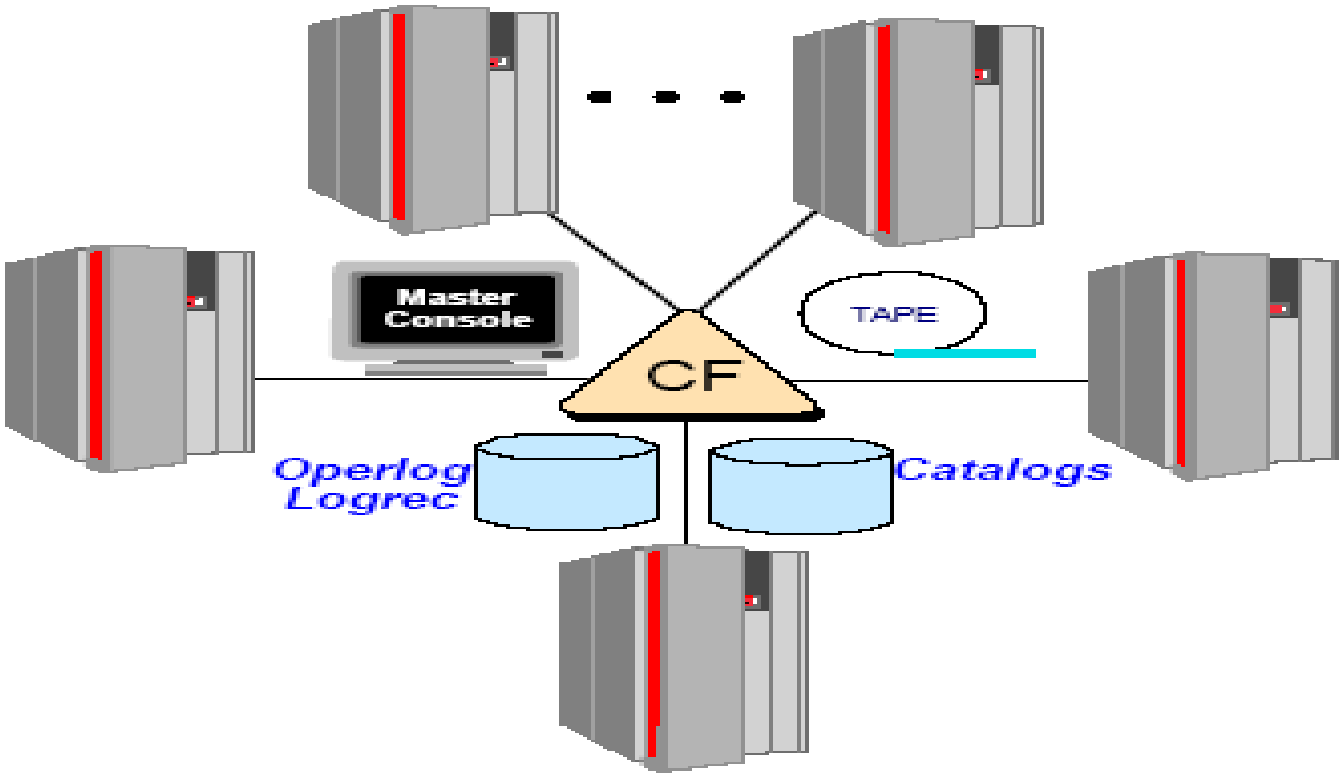
Key components – z/OS WLM

All the business performance requirements of an installation are stored in a *service definition*. There is one service definition for the entire sysplex.

The service definition contains the elements that WLM uses to manage the workloads. These include:

- One or more *service policies*.
- *Workloads*: Arbitrary names used to group various service classes together for reporting and accounting purposes. At least one Workload is required.
- *Service classes*: Where you define the goal for a specific type of work.
- *Report classes*: Aggregate set of work for reporting purpose.
- *Performance goals*: The desired level of service that WLM uses to determine the amount of resource to give to a unit of work.
- *Classification rules* and *classification groups*: Used to assign the incoming work to a service class and, if needed, to a report class.
- *Resource Groups:* Used to assign a minimum and a maximum amount of CPU SU/sec to one or more service classes.
- *Application Environments*: Set of address spaces with the same characteristics suitable for transactions requested by a client and executed in a set of server address spaces.
- WLM can dynamically manage the number of server address spaces to meet the performance goals of the work making the requests.

Key components – Coupling facility (CF)



- Coupling facility (CF) is central component for cross system data communications
- A combination of hardware and software (CFCC)
- Can be stand-alone or logical partition of a System z Server
- High speed links known as Coupling facility channels connect systems together
- Used by many components and applications to share data and provide single system view
- Dynamically partitioned into structures – three types – Lock, list and cache

Key components – Coupling facility (CF) exploiters

- Batch Pipes – multi system pipes
- Catalog – Enhanced catalogue sharing
- CICS – Logger, TS, CFDT, Name counter, oWLM
- DB2 – Data sharing
- GRS – Star, shared tape
- IMS – data sharing
- JES – checkpoint
- z/OS – operlog and logrec
- RACF – shared DB
- RRS – Logger
- DFSMS – HSM, RLS
- VTAM – Generic resources
- WebSphere MQ – Shared queues
- WLM – IRD, Enclaves
- XCF - Signaling

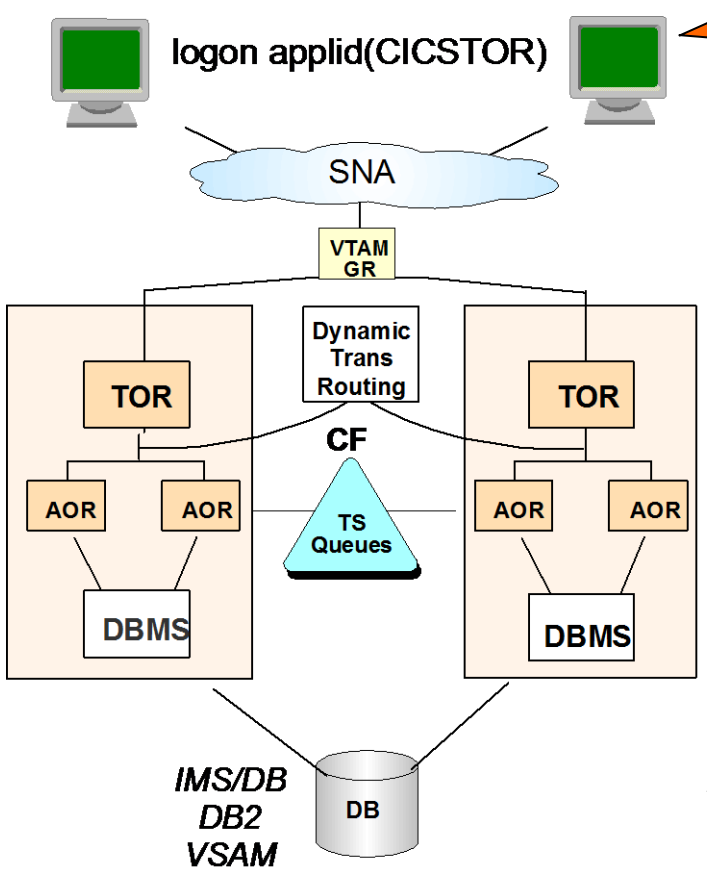
Routing work to CICS

- These are the typical mechanisms by which work arrives in CICS:
  - VTAM
  - Web / Web Services
  - CICS TG
  - Comm Server IP sockets
  - WebSphere MQ

CICS – VTAM

- CICS HA support for SNA devices provided by:
- CICS use of VTAM Generic Resources (gets you into CICS)
  - Supports LU types - LU0, LU1, LU2, LU3, LU6, LU6.1 and LU6.2
  - Requires CF
  - One z/OS communications server must be Advanced Peer to peer networking (APPN) network node
  - SIT GRNAME parameter defines generic name under which CICS will register with VTAM
  - VTAM APPL statement
  - Cannot use VTAM GR with XCF (XCF support being deprecated)
- CICSPlex SM workload management (once you are in CICS)

VTAM Generic resources



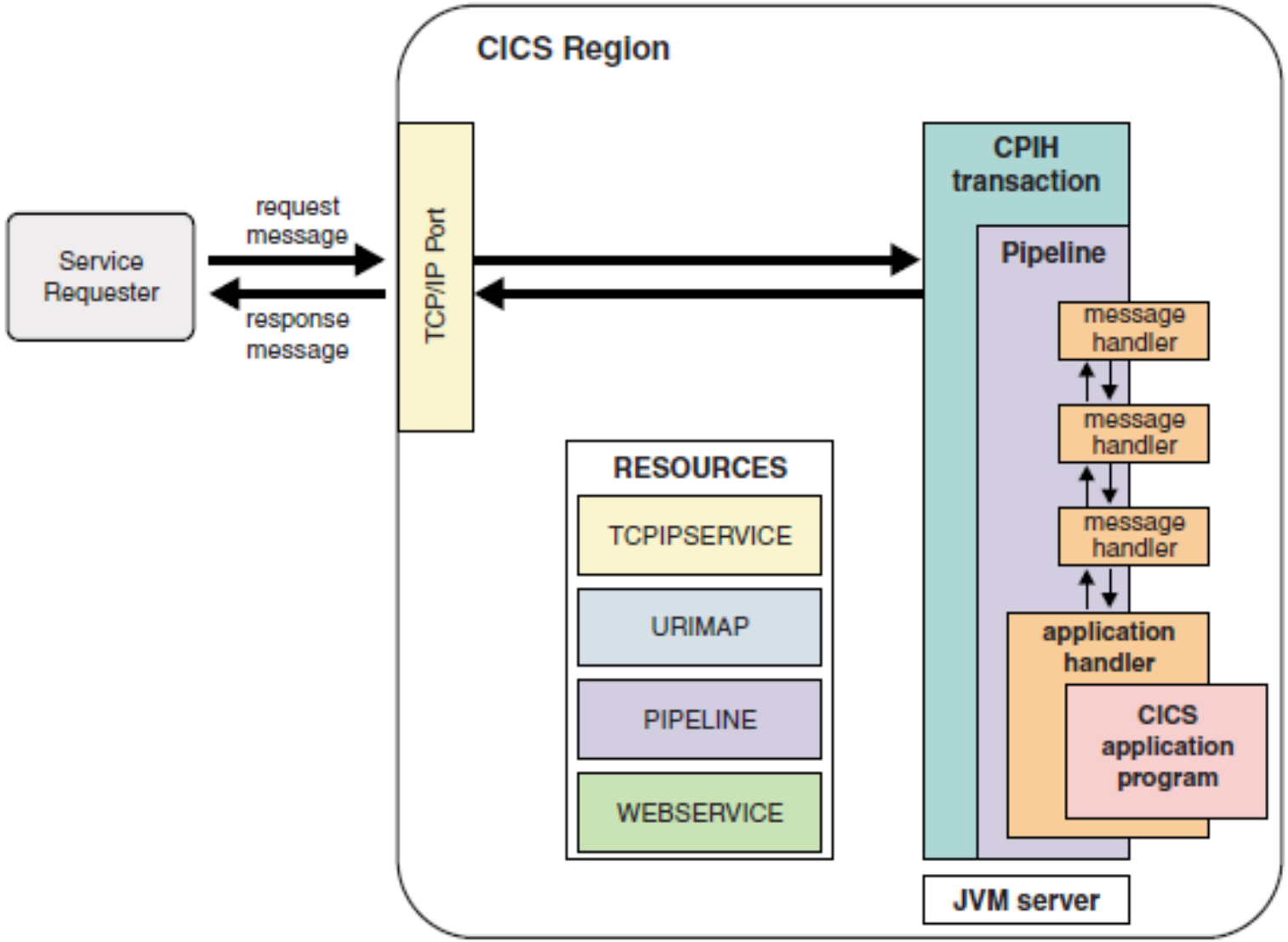
Dynamic session balancing across TORs using VTAM Generic Resource

Dynamic transaction routing across AORs using CICSPlex SM

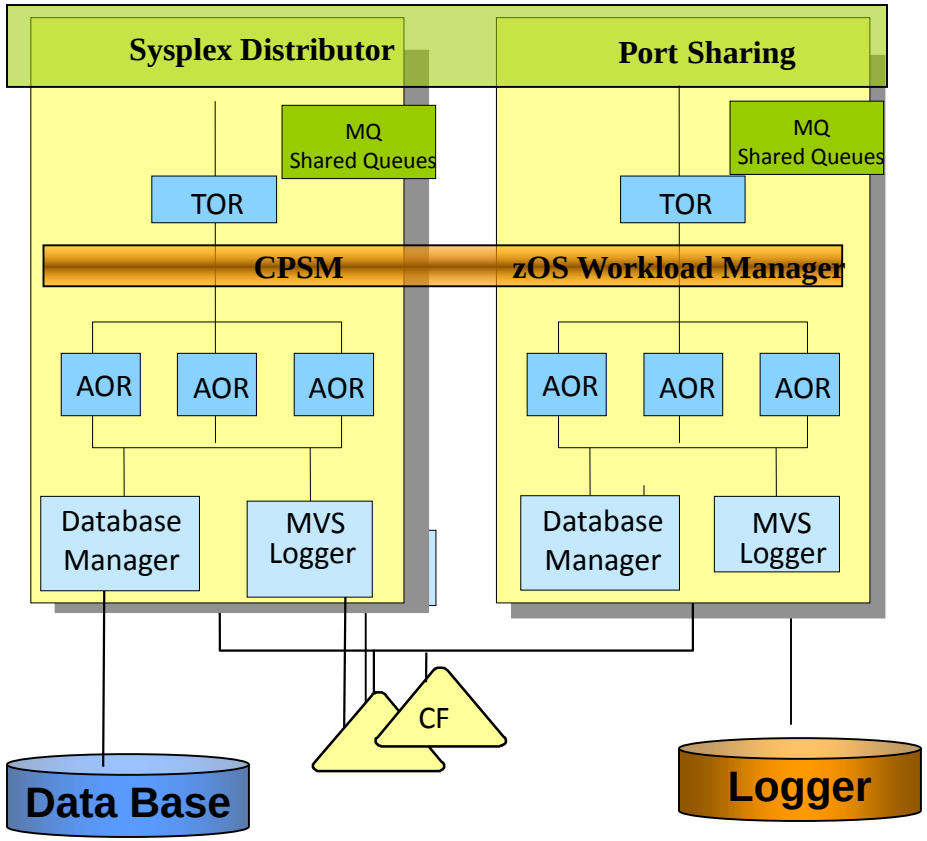
File system sharing via DB2 group attach, VSAM RLS or CICS function shipping

CICS – Web / Web Services

- CICS support for Web / Web Services uses common set of CICS components
- TCPIPSERVICE
  - Defines port to be listened on
  - Monitored by CICS CSOL transaction
- URIMAP
  - Provides information on how to process URI of HTTP, ATOM Feed or Web service request. Also maps requests to specific transaction identifiers. CPIH by default.
- PIPELINE
  - For web Service request provides details of handler programs to process request
- WEBSERVICE
  - Provide mapping information for WSBIND, WSDL and PIPELINE


CICS Region
Service Requester
request message
response message
TCP/IP Port
CPIH transaction
Pipeline
message handler
message handler
message handler
application handler
CICS application program
RESOURCES
TCPIPSERVICE
URIMAP
PIPELINE
WEBSERVICE
JVM server

Web / Web services



Session balancing across TORs/listeners

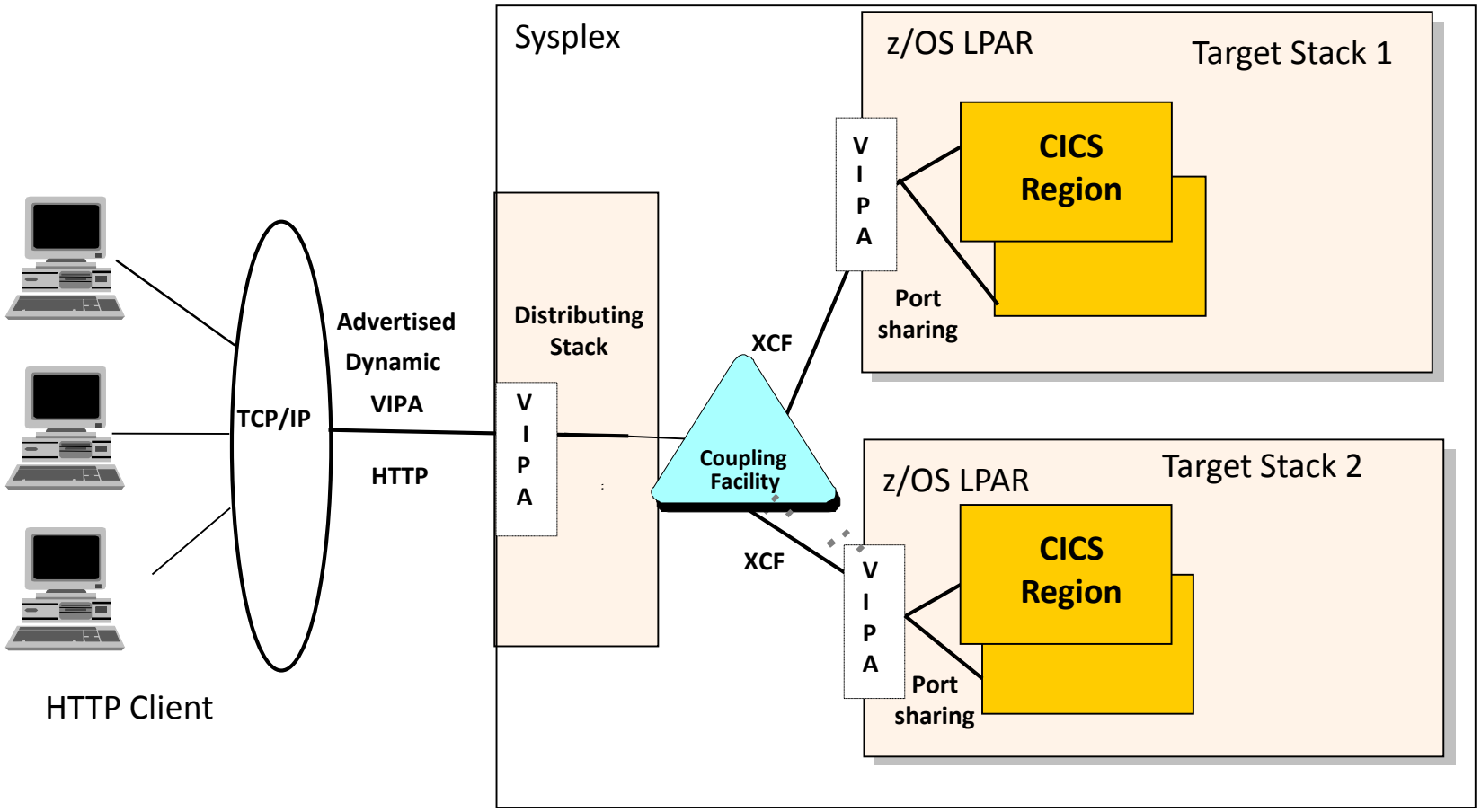
CICSPlex SM dynamic routing

Replicated Applications

Data Sharing with integrity
VSAM
DB2
IMS DB

29

CICS Web support - HTTP



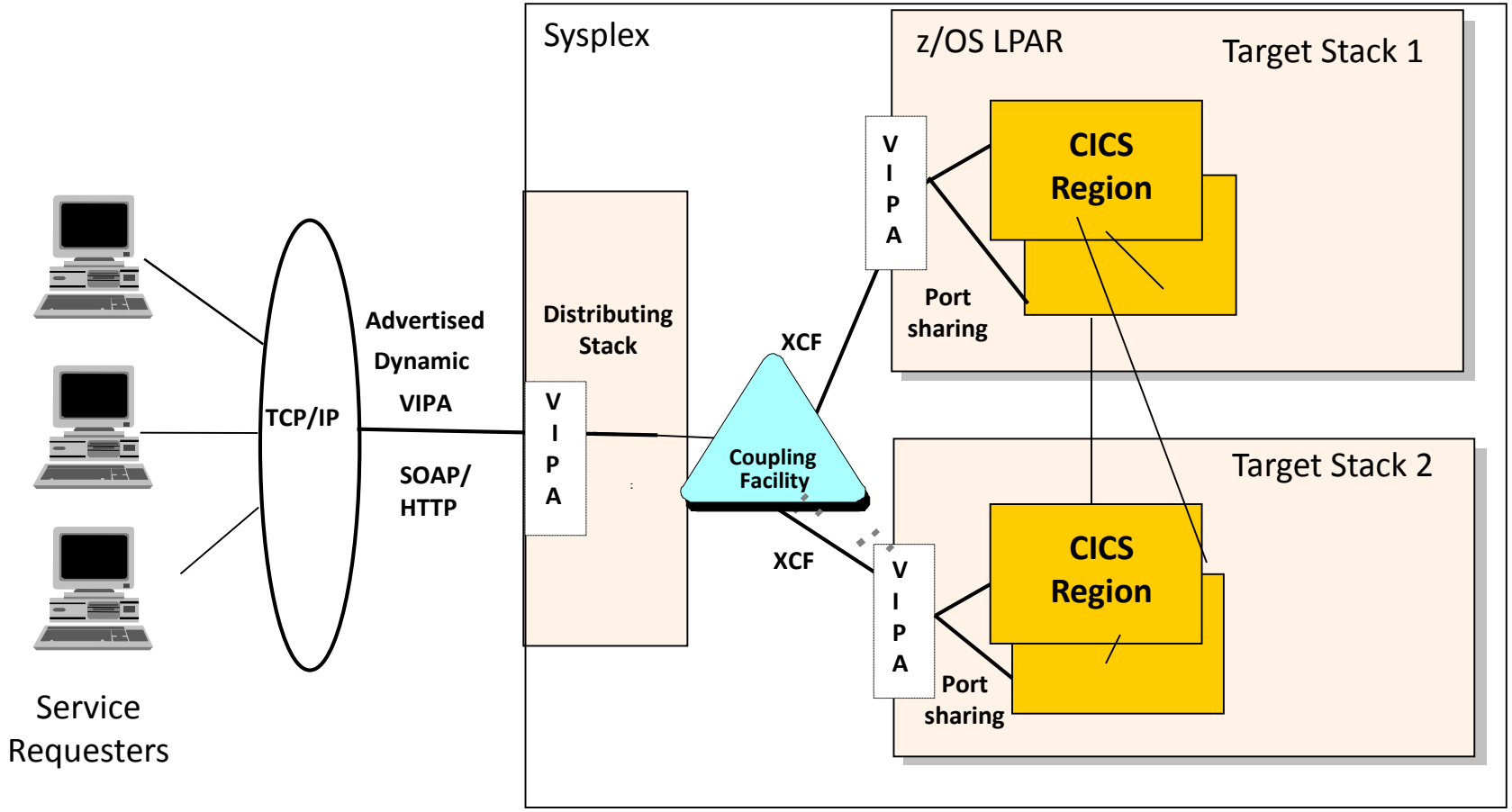
HTTP requests used for HTML and SOAP requests to CICS

HTTP 1.0 (with Keep Alives) and HTTP 1.1 supported

Can be used with either TCP/IP port sharing or Sysplex Distributor

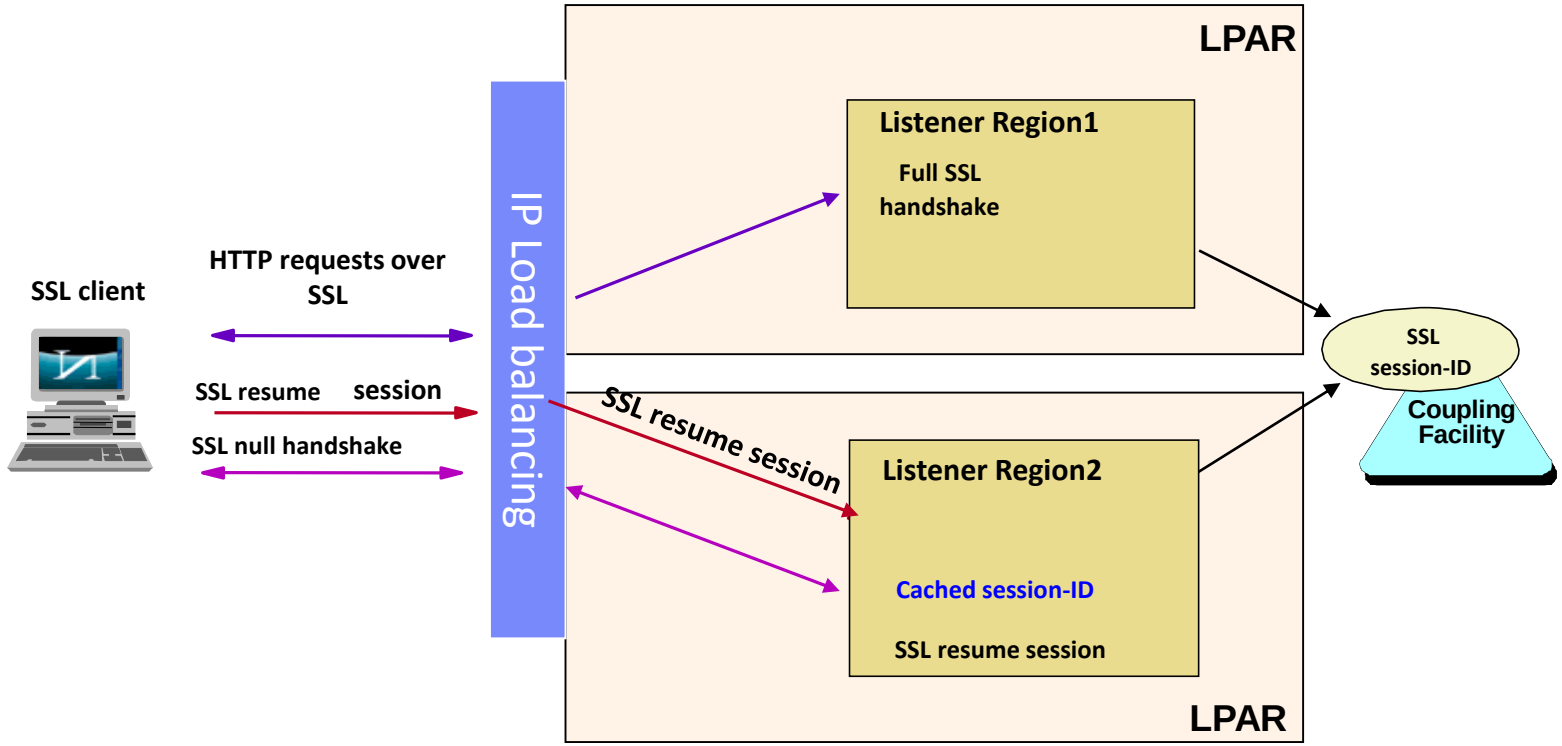
Requires that any session data is in shared storage (i.e. RLS or shared TS)

CICS Web services – SOAP/HTTP



WS-Transaction (2pc) support requires connections between all CICS regions

CICS TS v3.1 – Sysplex SSL session-ID caching

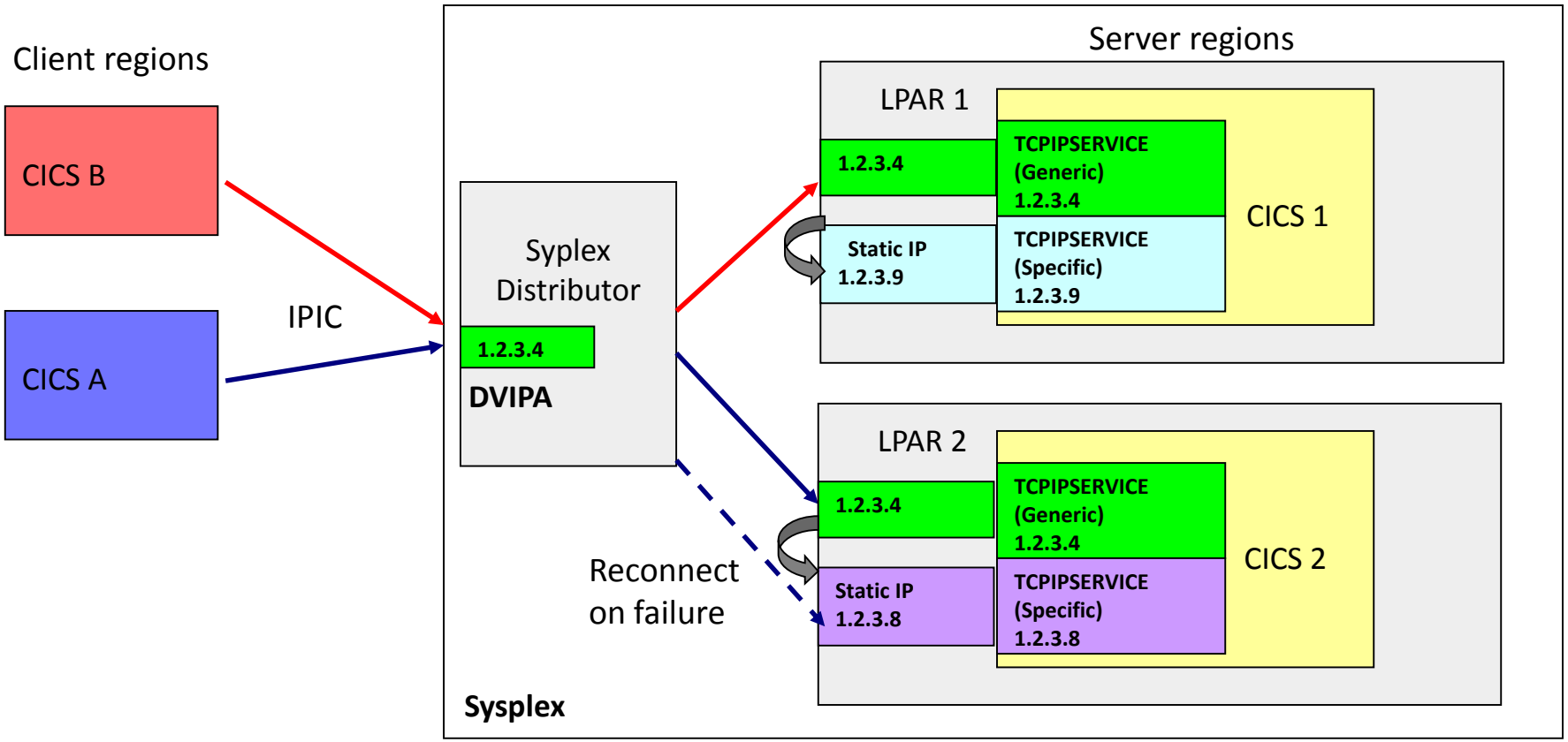


SSL session-IDs now stored in sysplex wide cache

SSL resume operation can lead to significant decrease in CPU usage when new SSL connections are created

Prior to CICS TS v3.1 SSL session-IDs were stored local to a CICS region

CICS TS V5.2 – IPIC High Availability



CICS server regions listen on a generic and a specific TCPIPService

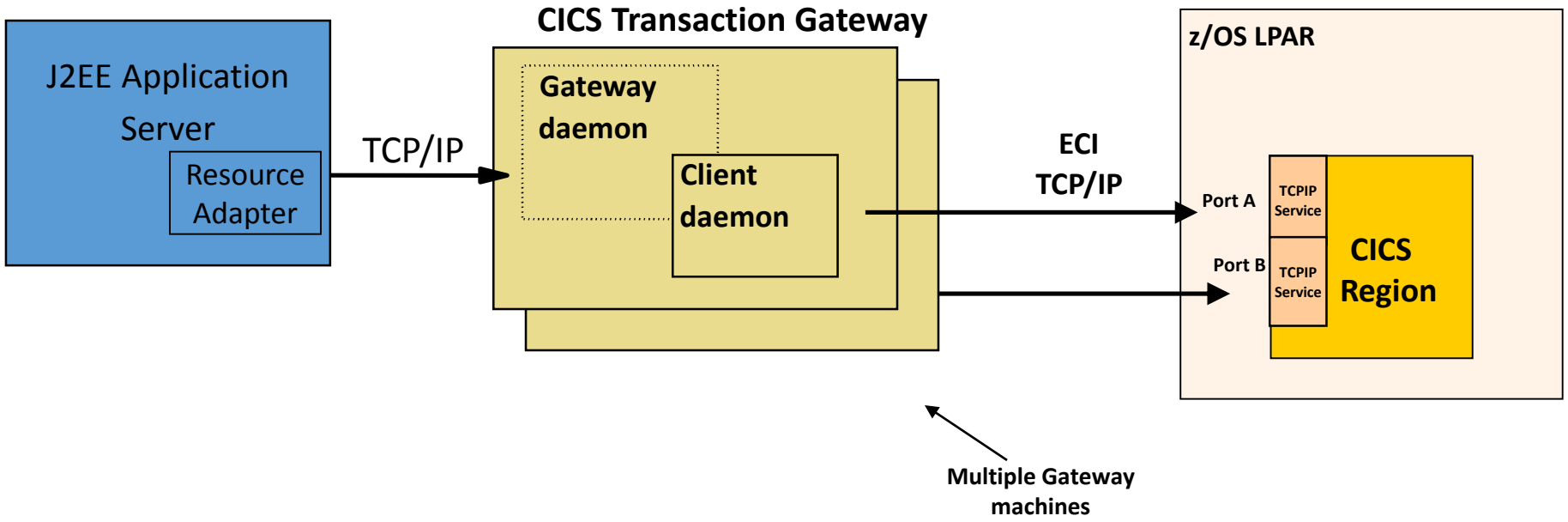
Client region reconnects to specific TCPIPService if connection terminated leaving UOW affinities

Supports Sysplex Distributor DVIPAs and Port Sharing

CICS – Transaction Gateway

- Leading connector for CICS
- Provides easy, secure integration to CICS for applications running on:
  - z/OS, Windows, Unix and Linux
  - Not only JEE certified application servers
  - But wide range of other applications:
    - Java applications, servlets, Microsoft .NET framework, C and COBOL applications

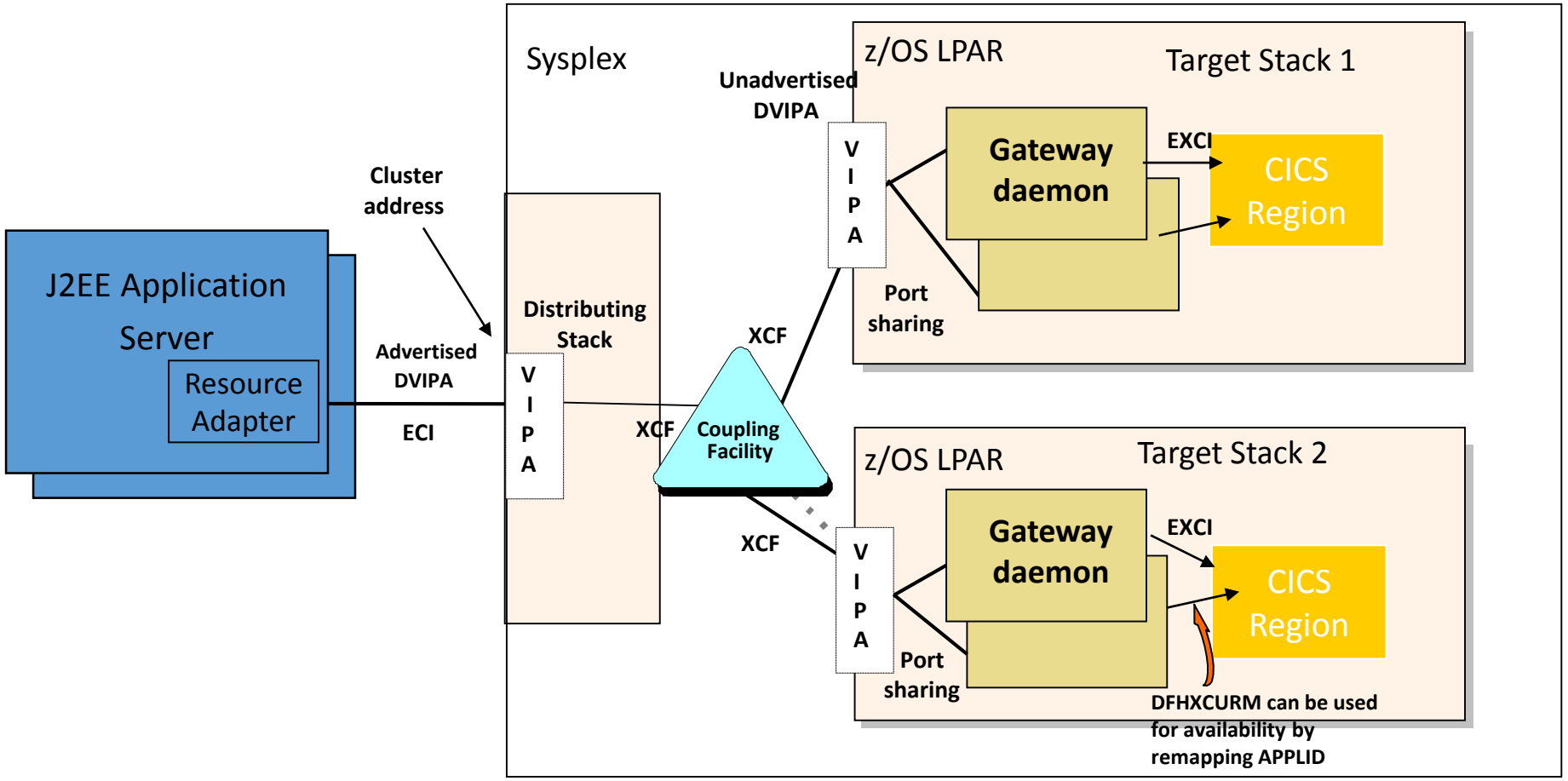
CICS TG - ECI/IP to CICS TS V2



- ECI/IP is a single socket based protocol
  - Multiple sessions multi-plexed over a single socket

- Multiple TCPIPService definitions can be used to provide failover
  - Can be exploited via multiple remote Gateway servers

CICS TG z/OS – load balancing (1pc)



Gateway daemons can be cloned to provide failover and increased throughput
Allows exploitation by TCP/IP port sharing and Sysplex Distributor
Integrated with WebSphere JCA connection pooling and local transaction support
DFHXCURM can be used to provide failover of the CICS routing region

Exploiting DFHXCURM – EXCI high availability

DFHXCURM – EXCI User replaceable module
Invoked on an EXCI Allocate and for retryable errors
- Invoked for initial ECI call for each thread
- On pipe re-allocation (if using CTG_PIPE_REUSE=ONE)
- For EXCI errors (such as CICS failure)

Sample URM exploiting CICSPlex SM API

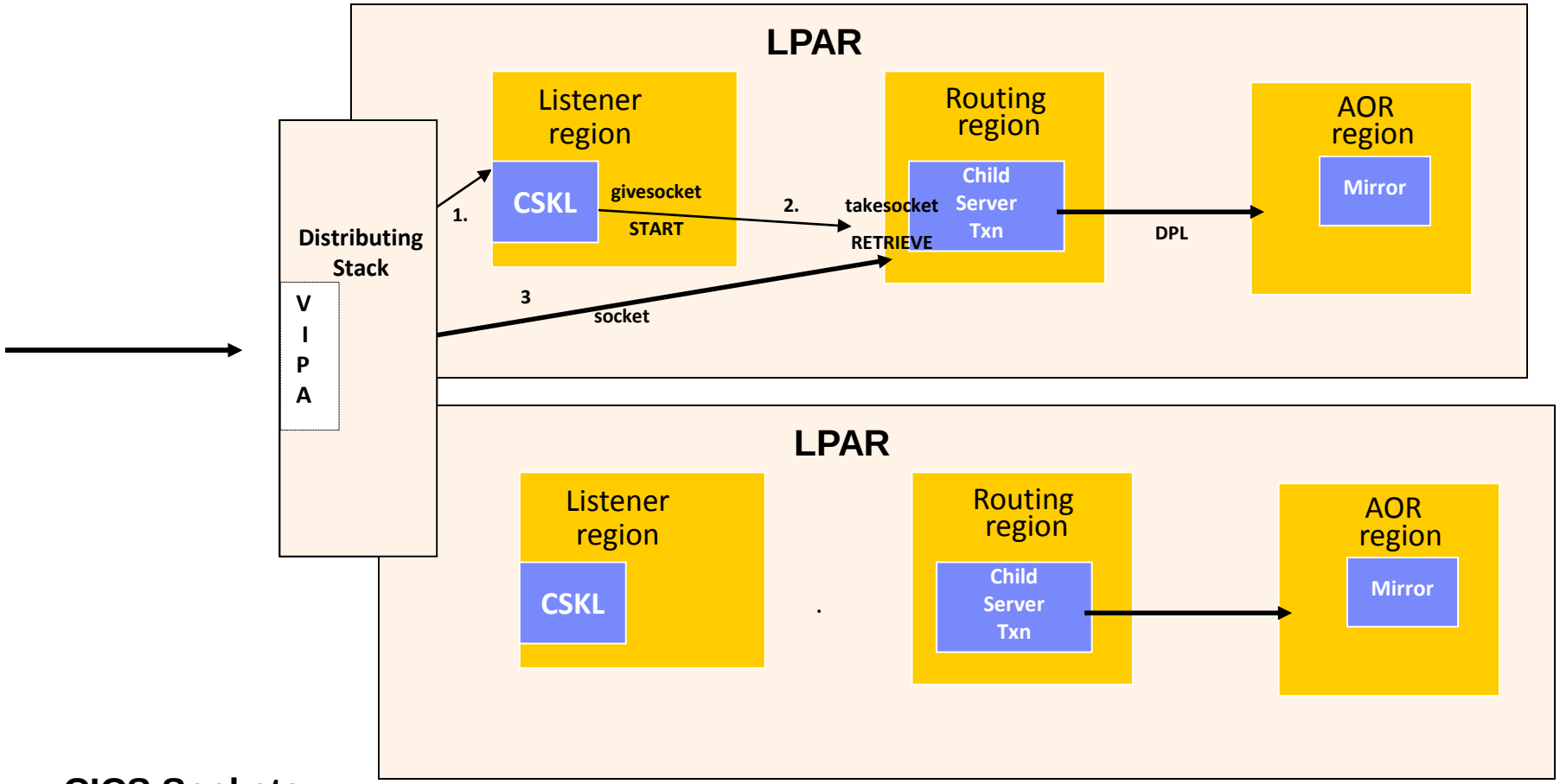
```
Invoke DFHXCURM
  CPSM CONNECT CONTEXT(PLEXNAME)                -> Connect to the CICSplex required
  CPSM GET OBJECT(MAS) TOKEN(MTOKEN)
  CPSM FETCH TOKEN(MTOKEN)                      -> returns list of MAS records which
                                        contain system name, status, location
  ...................
  CPSM GET OBJECT(CSYSGRP) CRITERIA(CTGSET)-> returns list of system names in CTGSET
  TOKEN(CTOKEN)                         system group

  CPSM FETCH TOKEN(CTOKEN)              -> Build a list of system names which
                              are in CTGSET and are active & local
```

To install the URM the following actions need to be completed by the user:
- Define a group in CPSM table CSYSGRP called CTGSET
- Define the group (i.e CTGSET) as the servername in the JCA connection factory
- Deploy the DFHXCURM into the CTG STEPLIB
- Note sample assumes that system name = CICS applid

CICS – Comm server IP sockets

- Comm Server IP sockets for CICS
  - Part of the z/OS Communications server component
  - Runs as a Task Related User Exit (TRUE) inside CICS
  - Listens on one or more PORTS
  - Two mechanism:
    - Iterative (fast, single threaded processing of multiple sockets)
    - Concurrent child (spawns child transaction passing socket to child)
  - Can run thread safe for scalability

Comm server IP Sockets for CICS



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 – WebSphere MQ

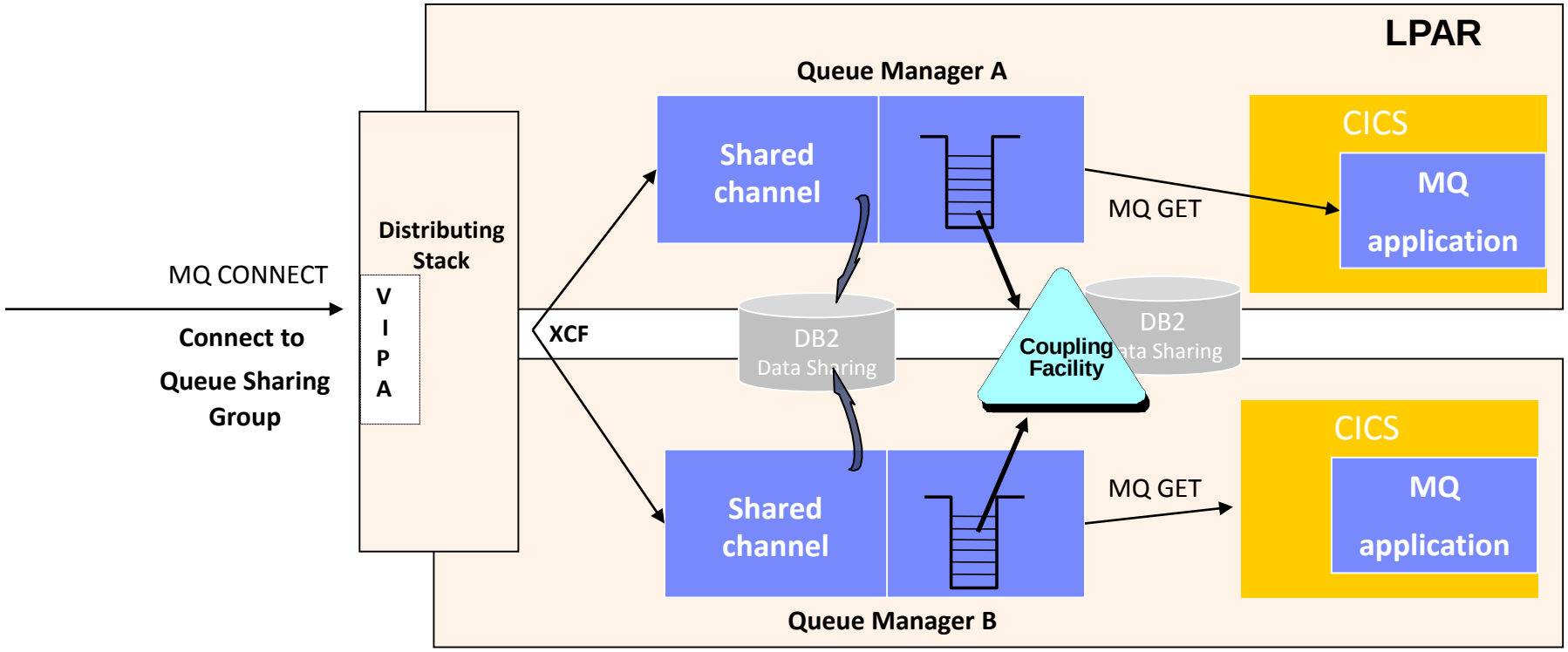
- WebSphere MQ clusters
- WebSphere MQ shared queues
- CICS connectivity

CICS TS and WebSphere MQ High Availability

- **CICS TS V4.1 - Queue Sharing Groups**
  - MQNAME on MQCONN can specify name of **QSG** or QM
  - Shared queue - centrally stored queue
    - Available through multiple QMs within the sysplex
    - Data stored in coupling facility
    - QM definitions shared in DB2 to allow cloning
    - Typically one QM per LPAR
- **MQ 7.0.1**
  - Peer recovery for XA clients
  - Support for in-doubt resolution
- **CICS TS V4.2 – Group UR recovery**
  - `RESYNCMEMBER(GROUPRESYNC)` on `MQCONN`
    - CICS can attach to any QM in the QSG
    - CICS can recover in-doubt URs after a CICS or MQ failure
    - Requires new version of WebSphere MQ

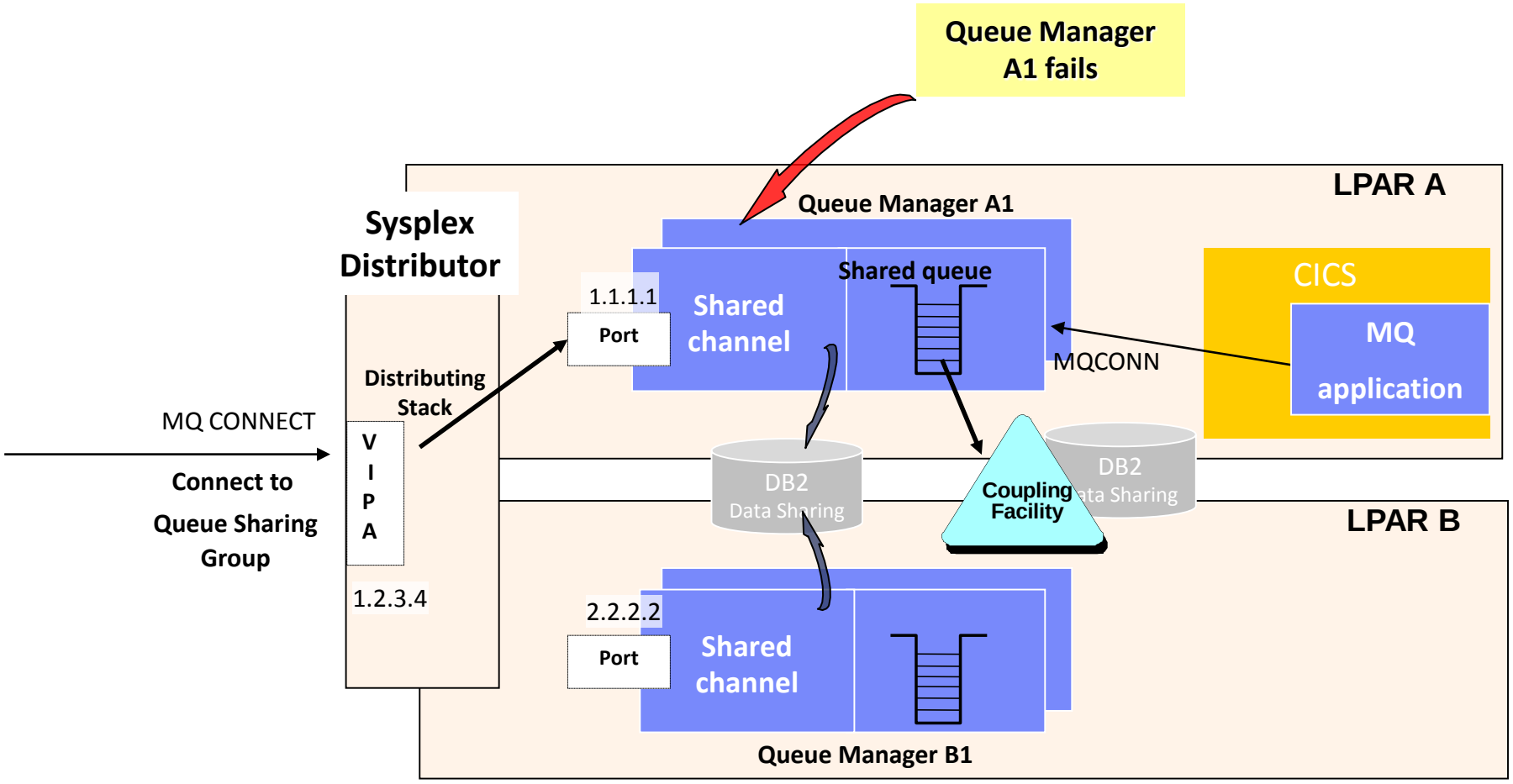
41

WebSphere MQ – shared queues



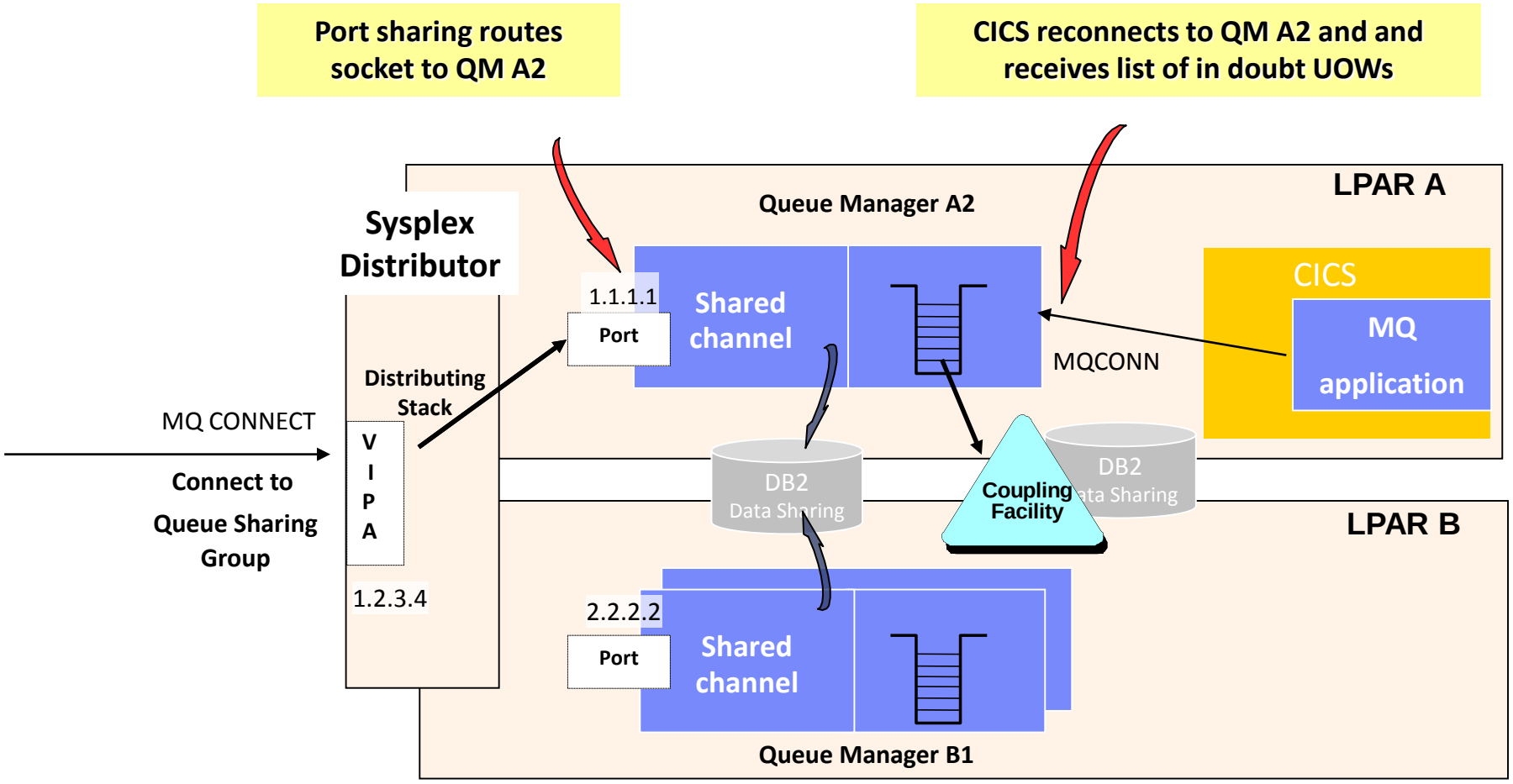
- **Queue sharing group**
  - Centrally stored queue, available through multiple queue managers within the sysplex
  - Data stored in coupling facility
  - Payload of large msgs (>63K) can now be stored in DB2 (MQ V6)
  - Queue manager definitions usually shared in DB2 to allow cloning
  - XA clients (such as WAS/JMS or TXSeries), have no support for shared queues
  - Typically one QM per LPAR

WebSphere MQ – shared queues



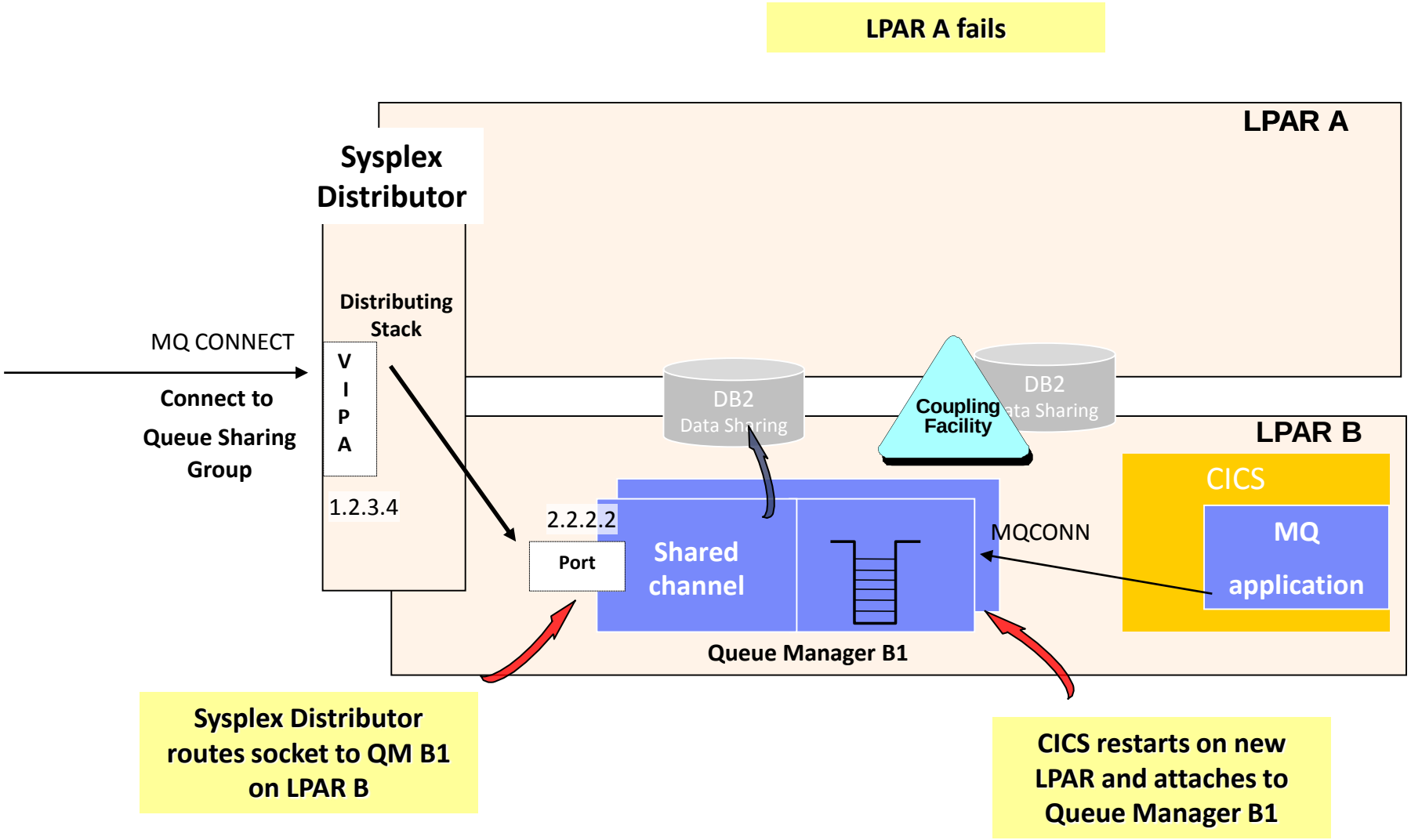
43

WebSphere MQ – shared queues



44

WebSphere MQ – shared queues

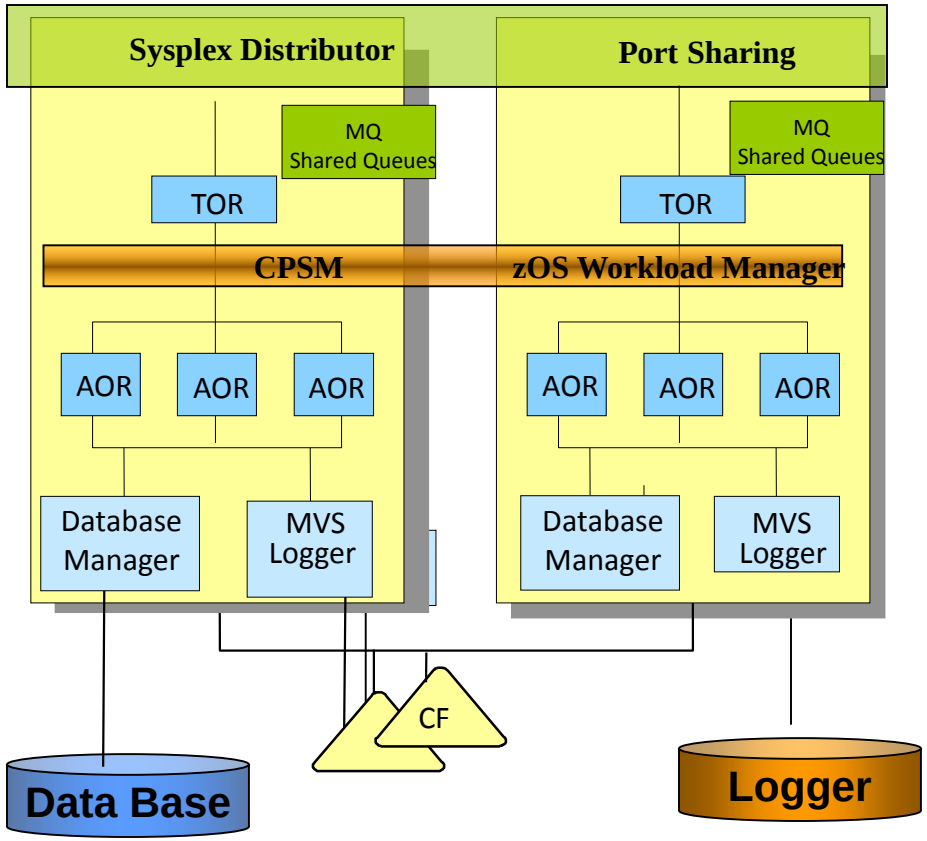


45

Routing work within CICS

- Once work arrives in CICS a number of things can happen
  - The request can run entirely within a single CICS region
  - But typically, in HA environments some or all of request will be routed.
- Need to consider the following:
  - CICS configurations
  - CICSPlex/SM
  - CICS Workload management
  - Cloning and application isolation

CICSPlex SM



Session balancing across TORs/listeners

CICSPlex SM dynamic routing

Replicated Applications

Data Sharing with integrity
VSAM
DB2
IMS DB

47

CICS Workload Distribution Options

- CICSPlex SM Workload Management provides the following options:

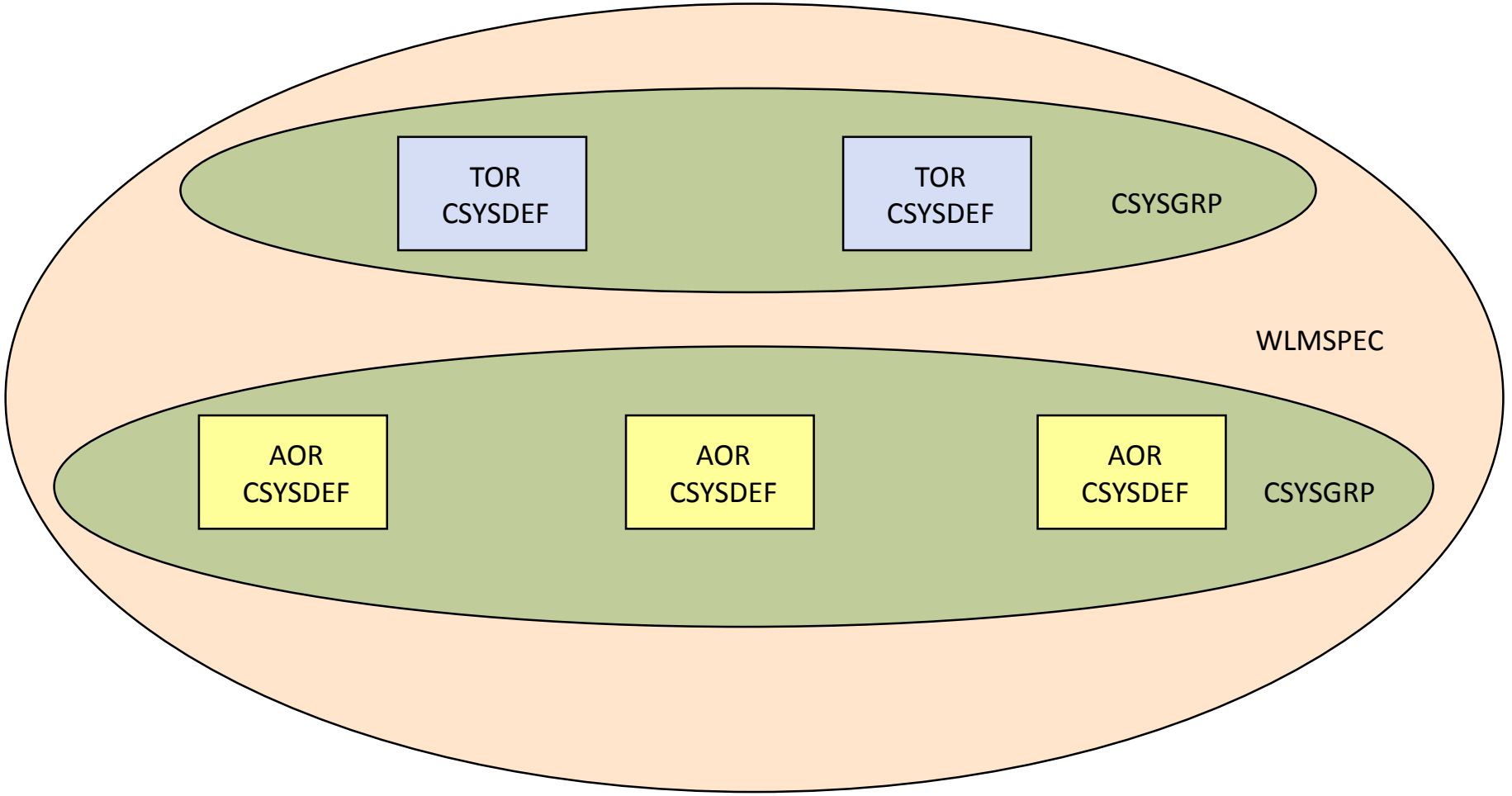
  - **Routing**
    - The process which decides which of the Target Regions is considered to be the most suitable candidate to route to, assuming:
      - Work ‘could’ be sent to any of the Target Regions
      - Work does not impose any constraints on the distribution

  - **Workload Separation**
    - The process of distributing specified work to a specific set of Target Regions
      - e.g. separation of work by application type

  - **Affinity management**
    - Routing workloads which have existing dependencies declared for them
      - e.g. State data maintained in memory in a single region
    - Affinities should be eliminated if at all possible!

48

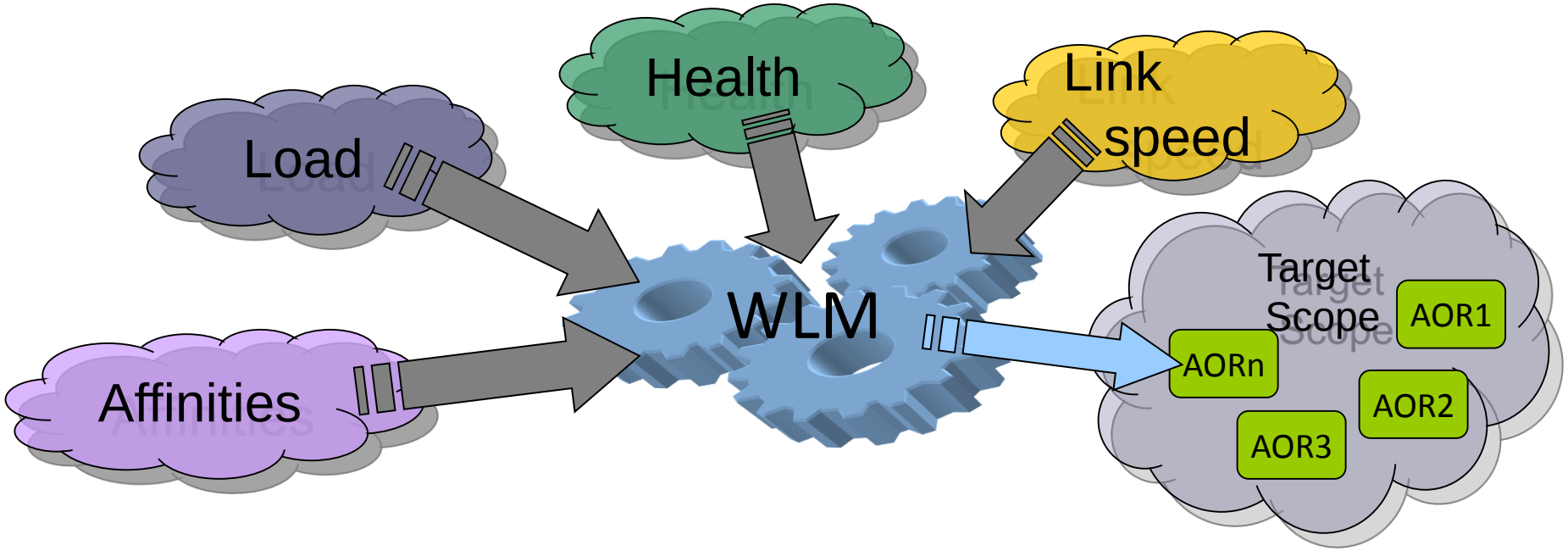
CICSPlex SM Routing…



49

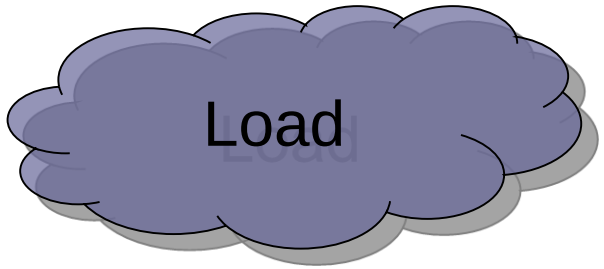
CICSPlex SM Routing…

- **Workload Management is about providing CICS with the ‘best’ target region at the moment the request is made from all of the possible region candidates.**



50

CICSPlex SM Routing…



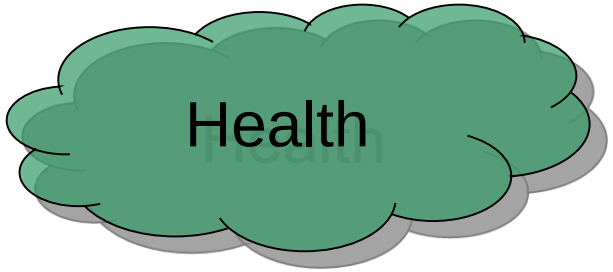
- **QUEUE** algorithm
  - LOAD = Current Tasks / MAXTASK
    - May be influenced by:
      - Task load health threshold on the CSYSDEF
        - When LOAD exceeds this higher LINK weights will apply
      - Task load queue mode on the CSYSDEF
        - ALL: LOAD includes active tasks, MAXTASK and TRANCLASS queued tasks
        - MAXTASK: LOAD includes active tasks and MAXTASK queued tasks

- **GOAL** Algorithm
  - QUEUE plus z/OS WLM average transaction response time goals

New **LNQUEUE**
  - As QUEUE but link weights not used
- **LNGOAL**
  - As GOAL but link weights not used

51

CICSPlex SM Routing…

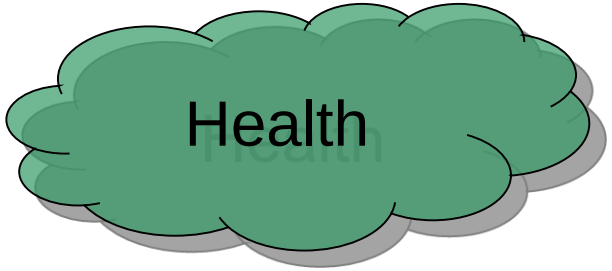


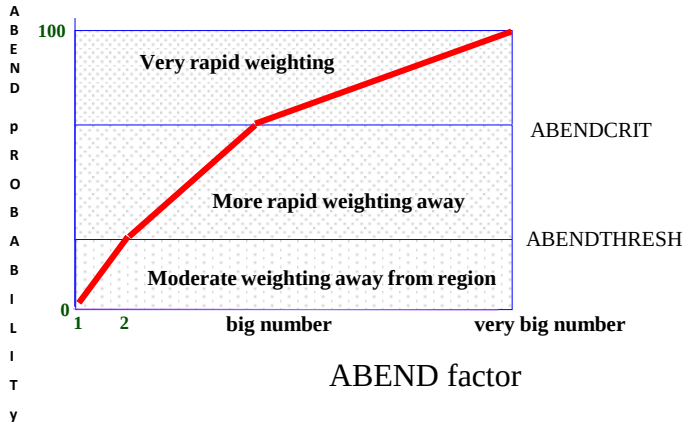
- Combination of:
  - Target Region Health
    - System Availability Monitoring
      - MAXTASK
      - Short on Storage
      - TRANDUMP or SYSDUMP
      - Region Stall condition
      - Non-responsive
    - Real Time Analysis
      - RTADEF associated with WLMSPEC
      - STATDEF associated with WLMSPEC
  - Target Region being quiesced
  - Abend compensation

52

CICSPlex SM Routing…

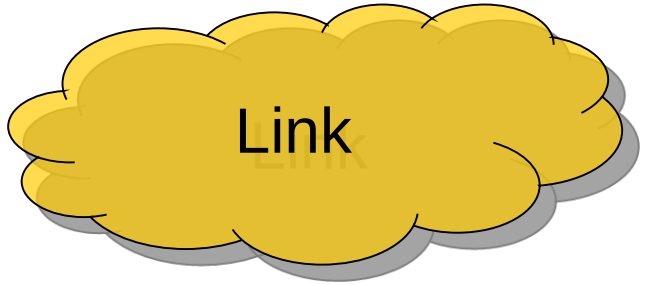
- Abend Compensation
  - Used to avoid regions where the transaction might abend
    - CICSPlex SM will calculate an abend probability (AP)
      - Based on past executions of the transaction
    - Parameters on the WLMSPEC or TRANGRP influence the routing
      - ABENDCRIT
        - If AP if greater than ABENDCRIT region will be considered unhealthy
      - ABENDTHREASH
        - If AP is greater than ABENDTHRESH region LOAD will be doubled





53

CICSPlex SM Routing…



- GOAL/QUEUE preference based on type of link to the target region
  1. No link -> Run in local region
  2. MRO XM (same LPAR)
  3. MRO XCF (same Sysplex)
  4. IPIC (same LPAR)
  5. IPIC
  6. APPC

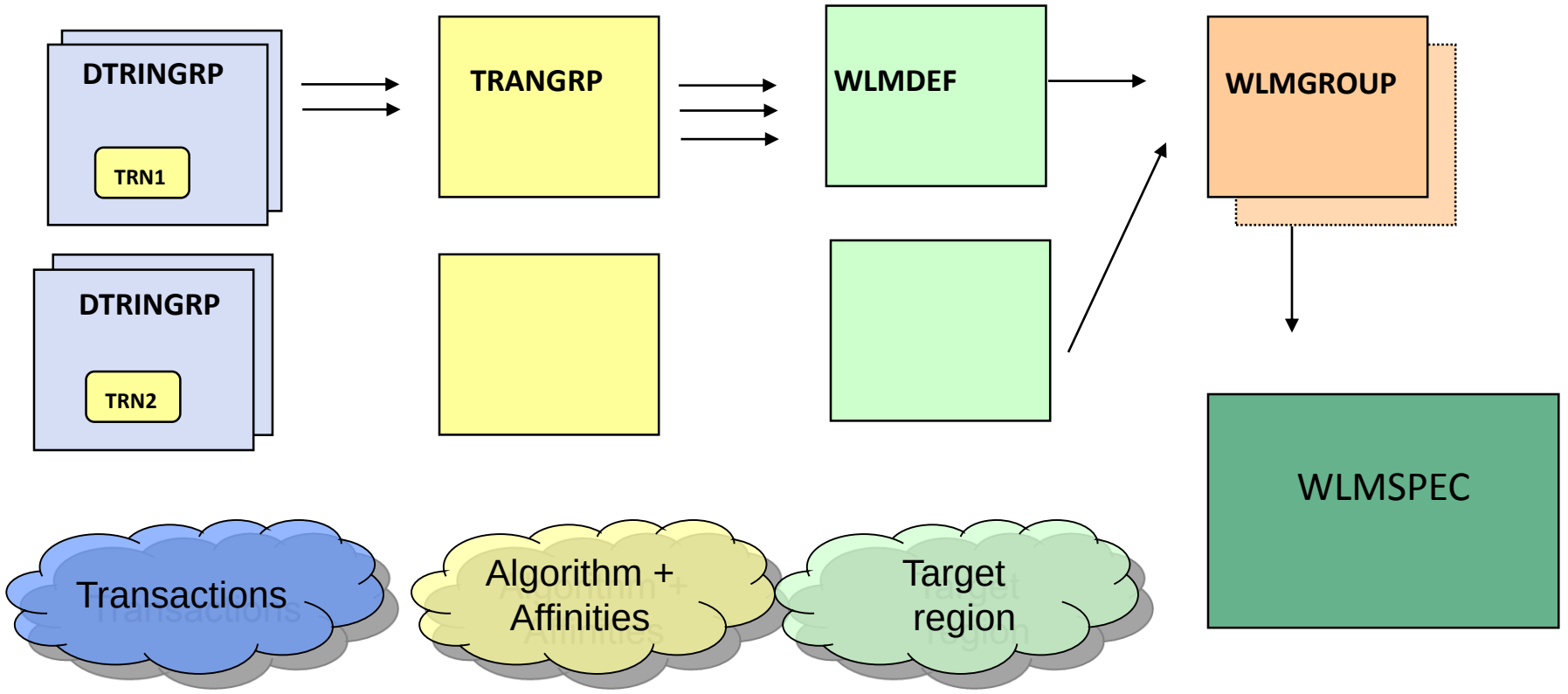
New

- New in CICS TS V4.2
  - LNQUEUE/LNGOAL
    - Excludes link weights
    - Provides flat routing model across specified target regions
    - Can provide more even CPU utilization in sysplex

54

CICSPlex SM Workload Separation…

- Workload definitions and groups



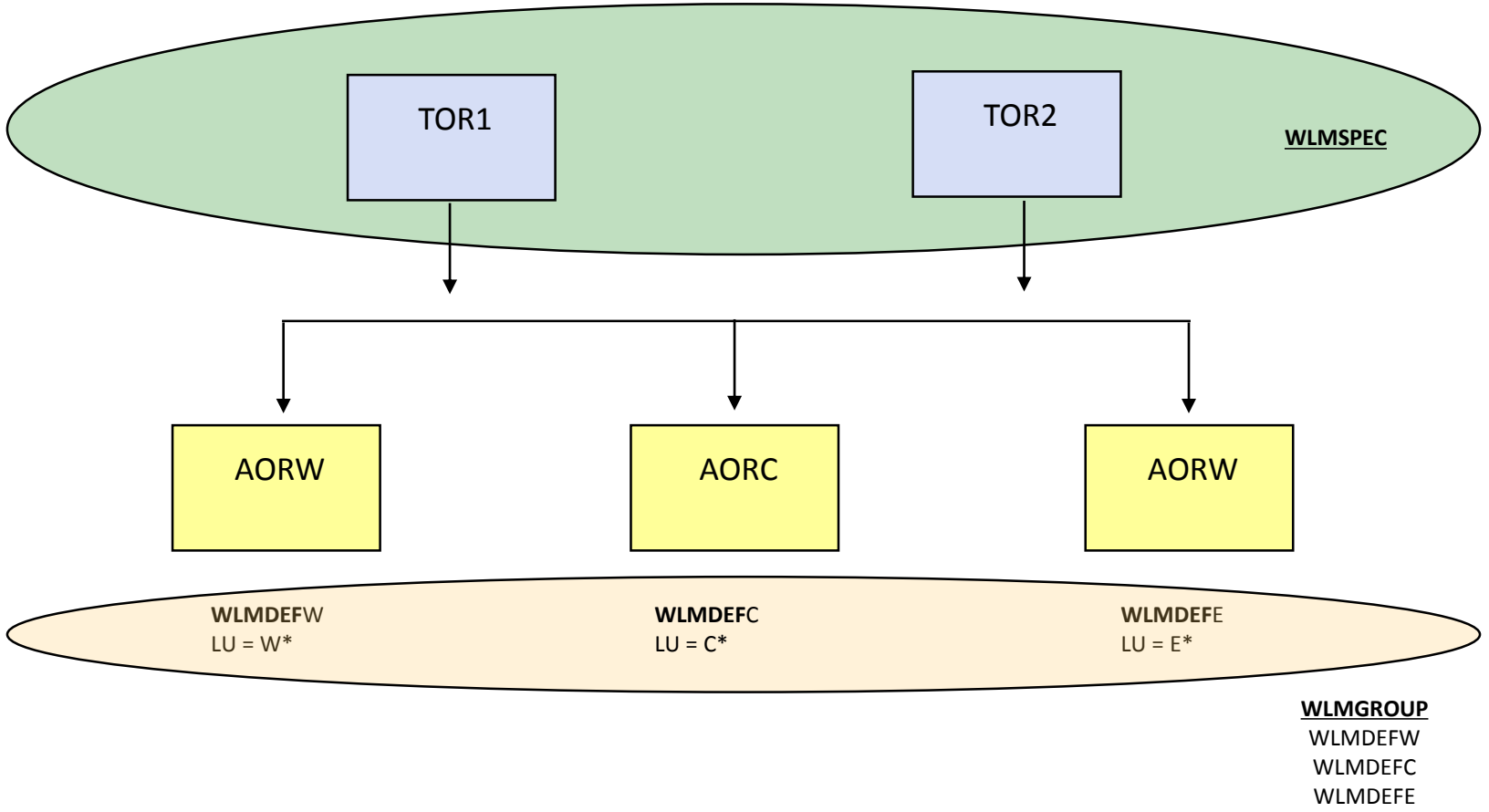
55

CICSPlex SM Workload Separation…

- WLMSPEC
  - Identifies a workload associated with 1 or more regions
- WLMGROUP
  - Used to associate WLMDEFs with a WLMSPEC
  - Can have multiple groups associated with WLMSPEC
- WLMDEF
  - Defines scope and separation and affinity requirements
- TRANGRP
  - A list of txns
  - Routing algorithm (LNQUEUE etc)
  - Can override algorithm in WLMSPEC

56

CICSPlex SM Workload Separation…



57

Application affinities and SPOF

- Application affinities
  - What are they?
  - Affinity lifetime
  - Why they affect availability
- Identifying affinities
- Removing affinities
- Identifying Single Points of Failure (SPOF)
- Removing single Points of Failure

CICSPlex SM Affinities

- What is an Affinity?
  - *A characteristic of an application that constrains transactions to run in specific regions and prevents dynamic workload distribution.*

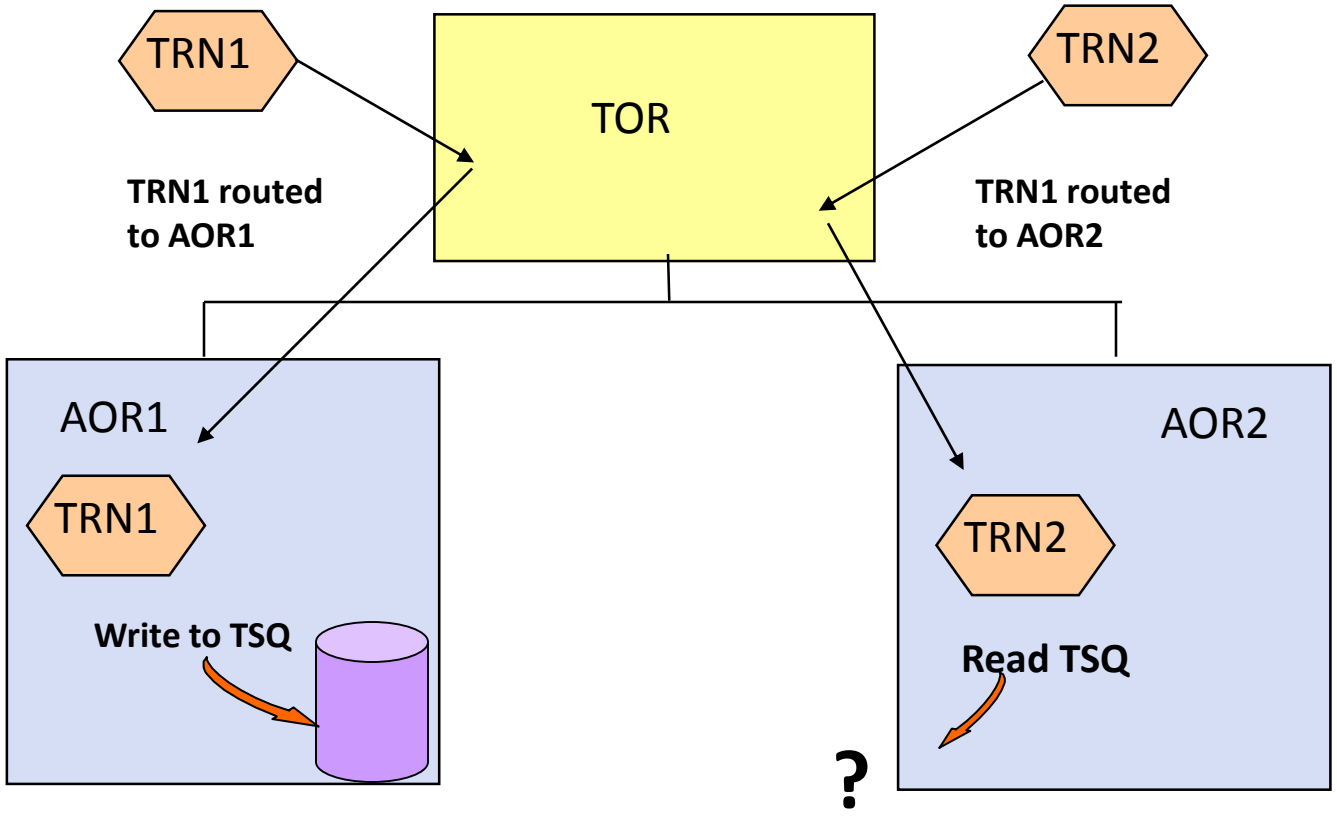
- Are there different types of Affinities?
  - Inter-transaction affinity
    - Transaction requires local state data (i.e. TS)
    - Two transactions synchronize (using ECB or ENQ)
  - Transaction-system affinity
    - Dependency on global user exit program
    - Use of INQUIRE and SET commands
    - Dynamic use of memory storage areas
  - Unit of Work
    - Multiple requests in same transaction (UOW) update a shared resource (DB2 row, VSAM record, recoverable TS/TD)

New

59

CICSPlex SM Affinities…

- Transaction Affinity example



60

CICSPlex SM Affinities…

- Affinity Relationships
  - Global
    - All instances of all transactions must be executed in the same target CICS region
  - LUNAME
    - All instances of a transaction from the same LUNAME must be executed in the same target CICS region
  - USERID
    - All instances of a transaction from the same USERID must be executed in the same target CICS region
  - BAPPL
    - All instances of all transactions within a BTS application must run in the target CICS region
  - LOCKED
    - Lifetime of UOW

61

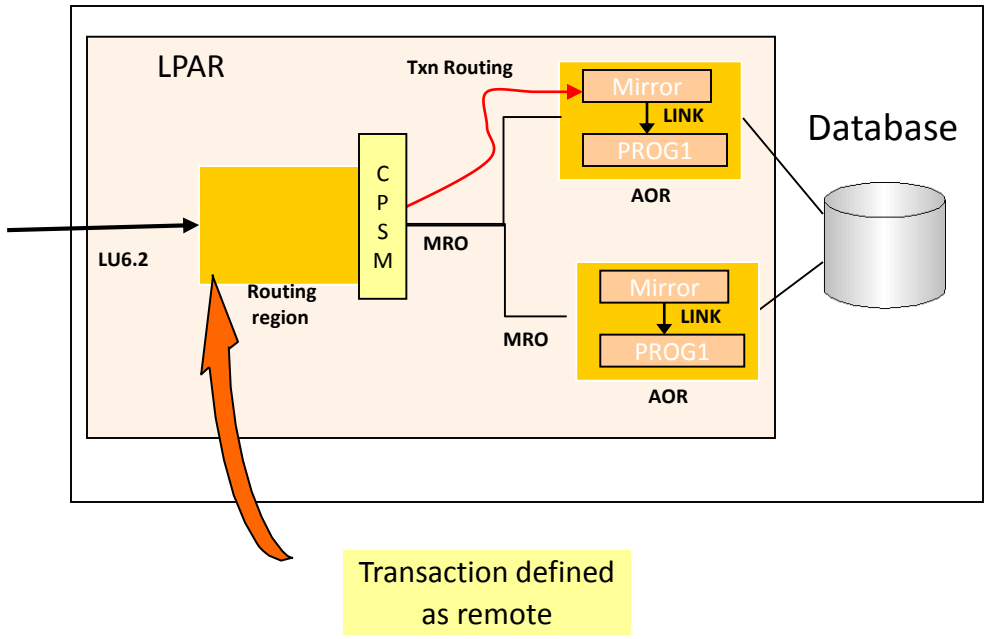
CICSPlex SM Affinities…

- Lifetime specified on WLMSPEC or TRANGRP
  - Lifetime expiration
    - PERMANENT
      - Workload termination
    - SYSTEM
      - Region termination
    - LOGON
      - LUNAME logs off from the routing region
    - SIGNON
      - USERID signs off from the routing region
    - PROCESS
      - BTS process ends
    - ACTIVITY
      - BTS activity ends
    - PCONV
      - When the transaction issues EXEC CICS RETURN with no NEXTTRANSID
    - UOW
      - Affinity has lifetime of the Global transaction
      - Applicable only to dynamic DPLs

62

CICS Dynamic routing – Transaction Routing

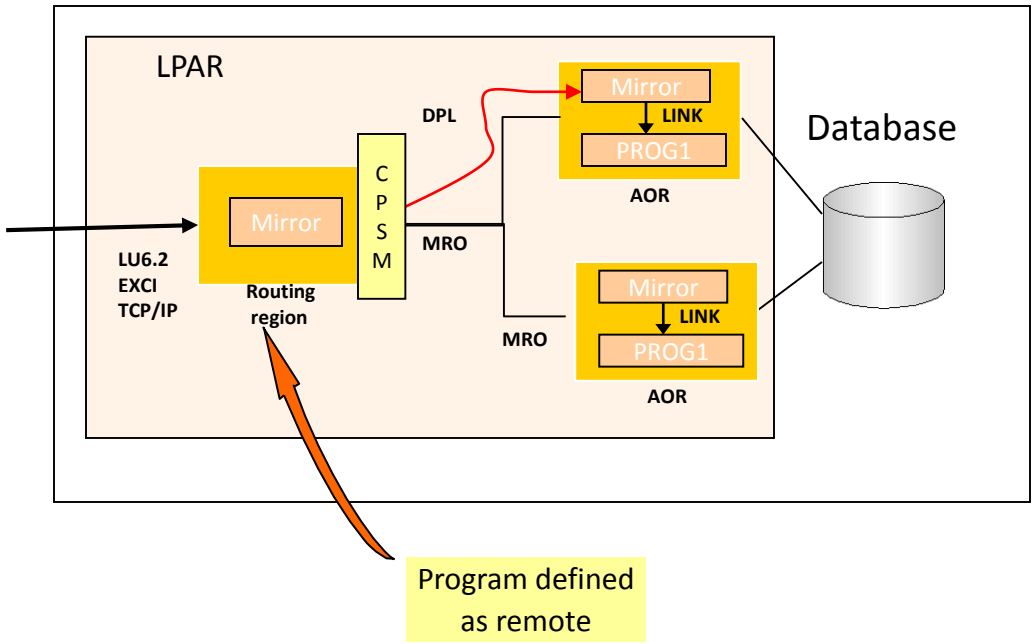
- Used to separate routing (i.e TOR) from business logic (AOR)
  - Listener regions do not run application code
  - 3270 routing supported for IPIC, APPC and MRO connections
  - Mirror routing supported only for LU6.2 connections
  - Web services pipeline routing supported for MRO connections (not IPIC)
  - Transaction is routed for entire lifetime of task
  - CICSPlex SM routing can by exploited using DYNAMIC(YES) on Transaction definition



63

CICS Dynamic routing – Dynamic Link

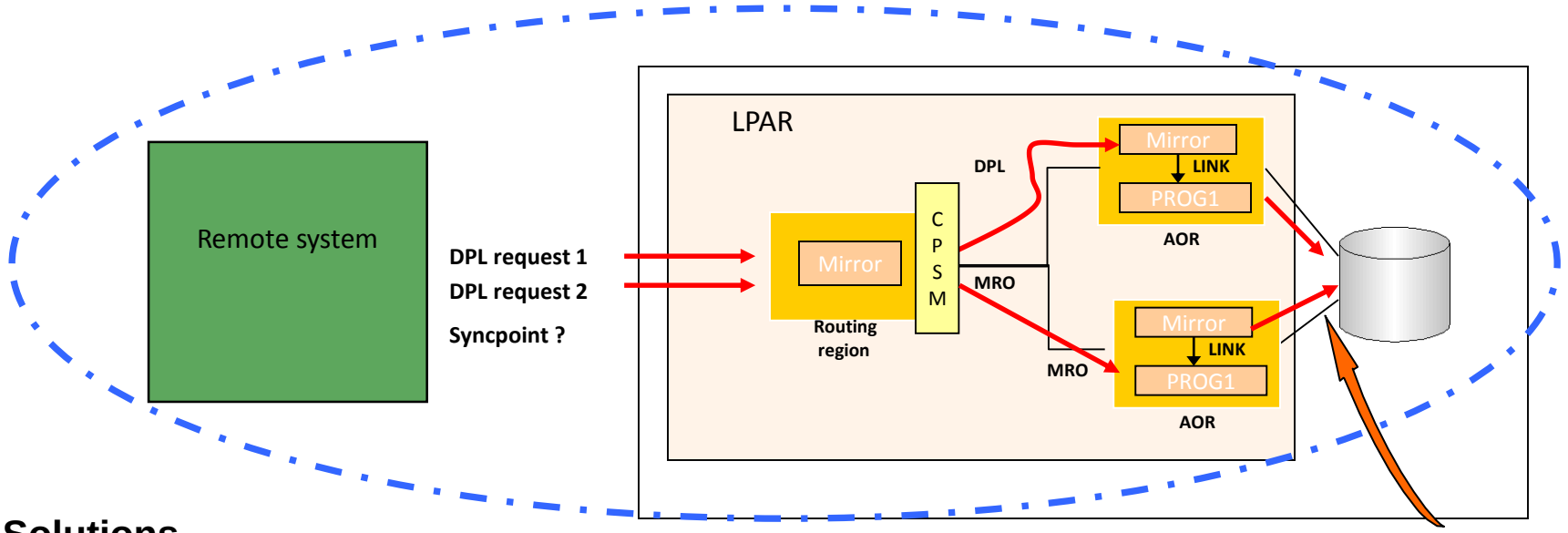
- **Dynamic Link**
  - Used for dynamic routing of LINK requests (from Listener to AOR)
  - Simple to configure
  - Supported for any connection type (IPIC, EXCI, APPC, ECI/IP)



- CICSPlex SM routing
  - CICSPlex SM routing controlled using DYNAMIC(YES) on PROGRAM defn
  - CICSPlex SM manages affinities for transactions

64

UOW affinities – deadlock



Solutions

- Removing Routing region/TOR
- Transaction route mirror
- Use custom wrapper program in routing region to call CICSPlex SM
- CICS TS V4.2 UOW affinity support
  - TRANGRP, affinity lifetime has a new value of UOW

Deadlock when second request updates file

65

CICS Applications

- For HA environment
  - Looking to avoid affinities and therefore maximise choice for CICSPlex SM
- Convert or develop threadsafe applications
  - Maximises concurrency
  - Reduces task switching
  - Relief on QR TCB which can
    - reduce need to clone regions
    - Give potential for regions consolidation

Accessing data

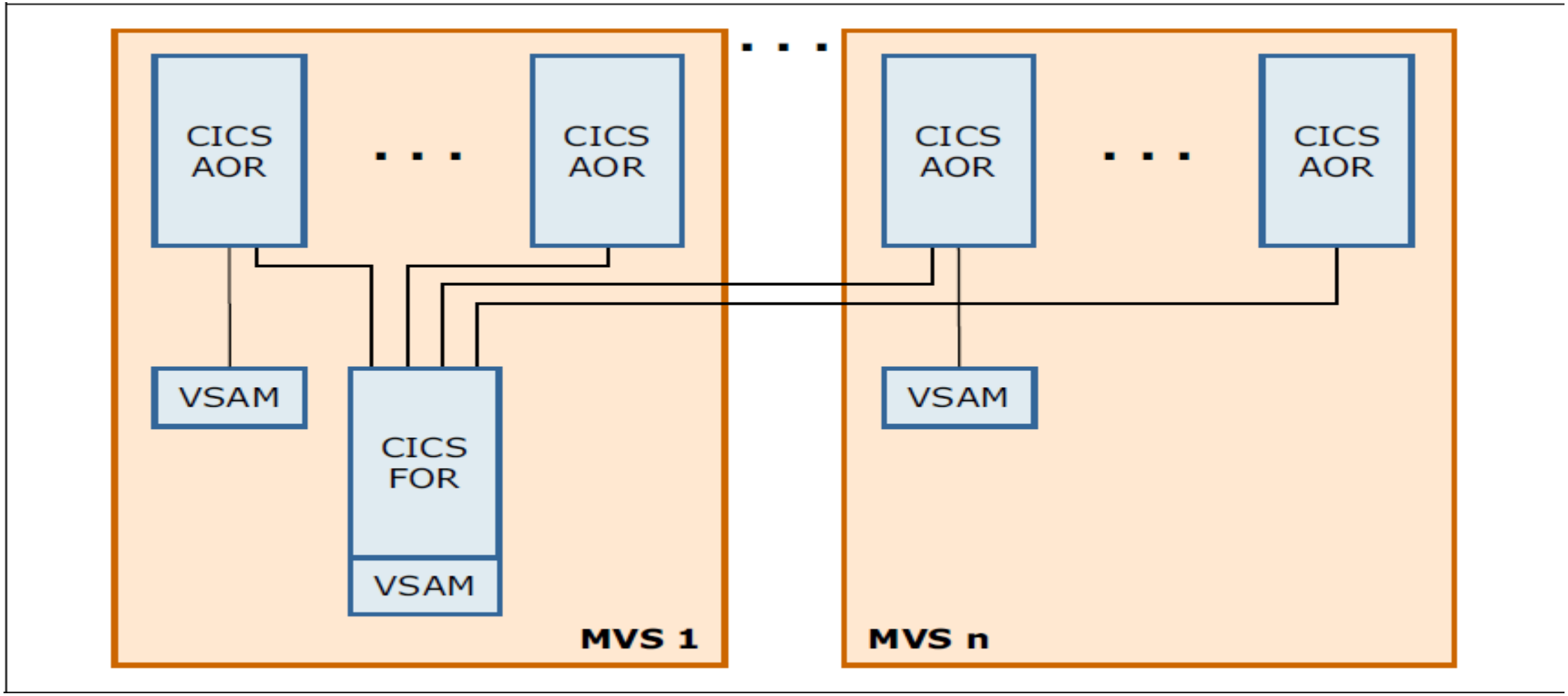
- VSAM
  - RLS
- CICS DATA TABLES
  - CMT|UMT
- DB2
  - Data sharing groups
- WebSphere MQ
  - Clusters
  - Shared queues
- Coupling Facility resources
  - Name counter servers
  - Shared Temporary Storage
  - Coupling Facility Data Tables
  - Optimized workload management
  - Logger

VSAM support

- CICS support for VSAM
  - CICS has supported KSDS, ESDS, RRDS, ESDS for many releases
  - Files defined locally or remotely
  - Access between CICS using CICS connections
  - Function Shipping used to access remote files

- CICS supports RLS and non-RLS
  - Can have RLS and non-RLS files within a single CICS region
  - But can’t mix access type for a single file across SYSPLEX

CICS and VSAM Non-RLS access



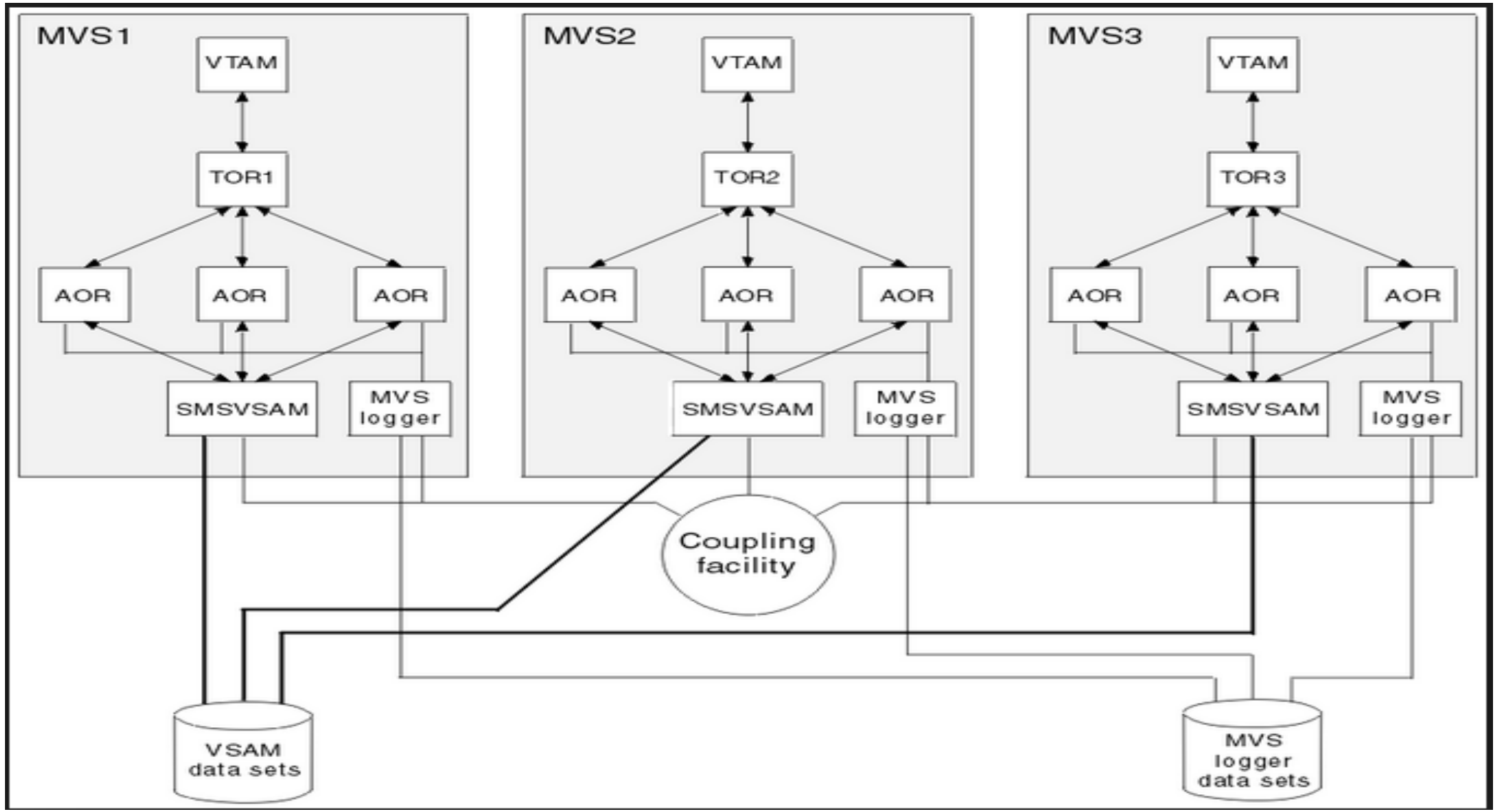
Cross system access via CICS connections : MRO/XCF, APPC, IPIC

VSAM support

- CICS and VSAM non-RLS
  - Concurrent ‘Read only’ access supported across multiple CICS system
  - Concurrent update limited to single CICS region
  - Leads to FS requests to FOR regions – Typically become SPOF
  - CMT/UMT are options for access/performance
  - Myths around RLS performance creates reluctance to adopt technology (*)

(*) See REDPAPER: CICS and VSAM RLS for details
http://www.redbooks.ibm.com/redpapers/pdfs/redp4905.pdf

IBM Parallel SYSPLEX – CICS and RLS



VSAM RLS Components

- DFSMSvsam provides
  - N-way data sharing
    - Store-through cache with cross-invalidation of buffers
  - Sysplex-wide record-level locking
  - SMSVSAM address space in each MVS, with shared buffer pool
  - Recovery attributes in catalog entry – LOG(NONE|UNDO|ALL) and LOGSTREAMID for LOG(ALL) – recovery becomes a data set property rather than a file property
- CICS provides
  - Logging of VSAM data set changes for backout and recovery
  - Backout of uncommitted changes at transaction abort, AOR restart, etc.
  - Forward recovery logging and file autojournalling
- CICS VR (or equivalent forward-recovery product) provides
  - Inventory of image copies and CICS logs
  - Automated data set restore and forward recovery
  - Uses required RLS commands to unbind and rebind any unresolved locks etc.

New Functions with VSAM RLS

- Read Integrity
  - **Uncommitted**, the default (aka “dirty read” – only option for non-RLS and data tables)
  - **Consistent** (Do not read data that is currently being updated)
  - **Repeatable** (Data this task has read cannot be updated)
  - Readinteg specified on file, can be overriden by specifying on API request
  - Consistent or Repeatable cause additional locking, and potential deadlocks
- Conditional Locking
  - NOSUSPEND option on READs – return RECORDBUSY if record is locked, rather than waiting
- Browse for Update
  - EXEC CICS READNEXT | READ PREV UPDATE TOKEN()
  - REWRITE | DELETE TOKEN()

Exception conditions introduced with RLS

- LOCKED
  - Request encounters a retained lock
  - Can also occur for non-RLS
    - VSAM retains locks for RLS, CICS (NQ domain) retains locks for non-RLS
  - Programs must already handle this for non-RLS requests
    - So should not be an issue moving to RLS
- RECORDBUSY
  - Conditional request encounters an active lock
  - RLS only
  - Can only be seen by programs which use NOSUSPEND
    - So not an issue for existing programs

VSAM RLS and High Availability

- Survive failure of CICS region
  - VSAM retains locks for any incomplete work, to maintain data integrity
  - Work can be routed to other regions which can access the data set
- Survive failure of the SMSVSAM server
  - VSAM retains locks for any incomplete work, to maintain data integrity
  - CICS regions automatically reconnect to the server and resolve any incomplete work when it becomes available again (Dynamic RLS Restart)
- Survive failure of CF structure
  - Ideally, CF Duplexing is in use
  - SMSVSAM can usually transparently recover CF structures
    - If not, for cache structure failure, requests will back out and retry later
    - If not, for Lock structure failure, leads to “Lost Locks” situation
      - When new Lock structure accessible, Lost Locks Recovery carried out – on a per-data set basis, but can take some time
- Survive failure of the data center
  - Offsite recovery
  - GDPS

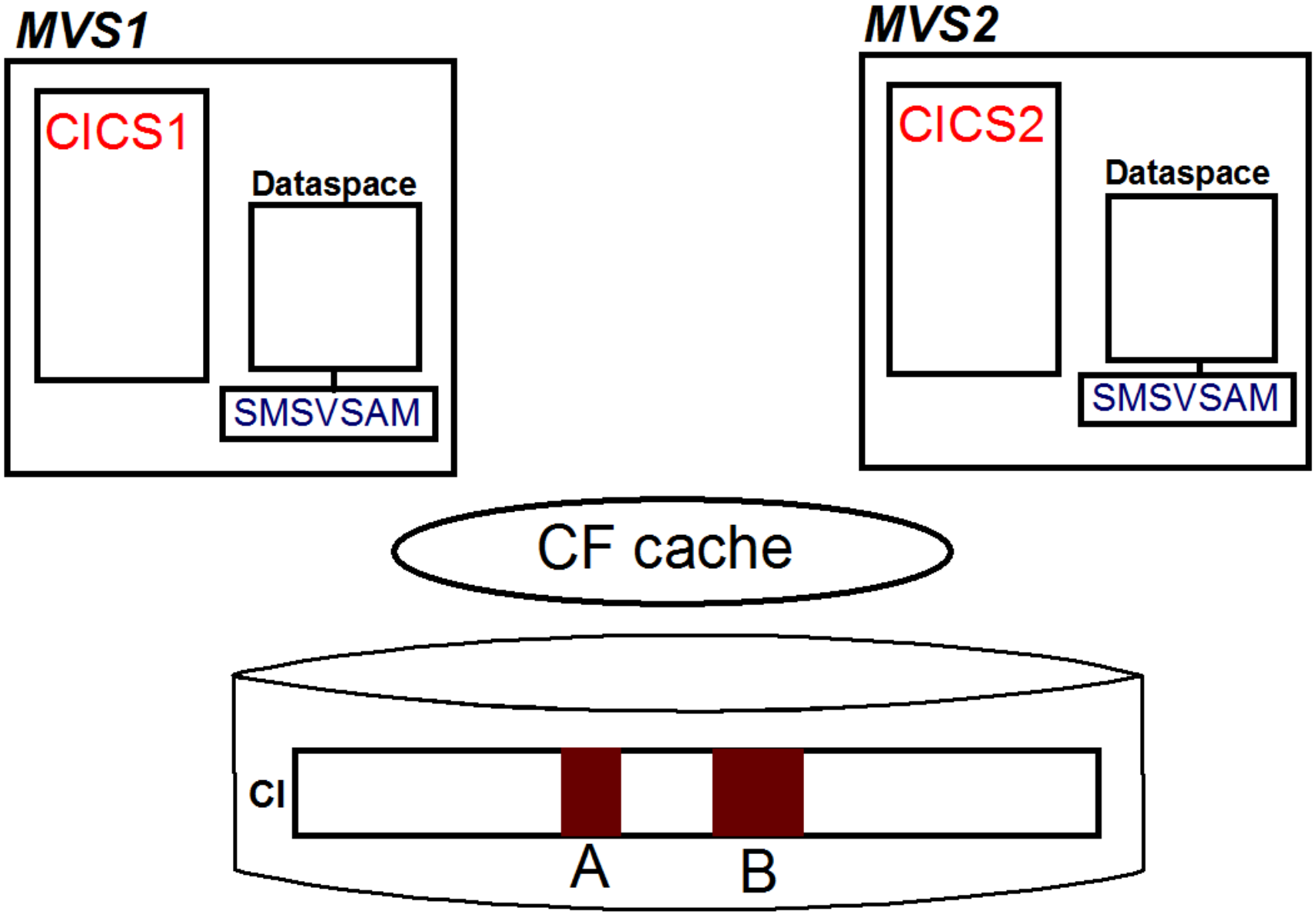
Backout failures, and indoubt support

- Failures, such as CICS transaction abend (or explicit rollback request), attempt to access RLS when SMSVSAM server unavailable, etc.) are backed out
- If the backout fails
  - Record locks are converted to retained locks
  - Means other records in the data set remain available
  - Backout is retried when failure is repaired
- If a coordinator fails during syncpoint processing, an in-doubt condition arises
  - Record locks are converted to retained locks
  - When contact is re-established, UOW can be driven for the correct resolution
- Options exist to ‘throw away’ retained locks for continued data access, but at risk of data integrity
- Retained locks also supported by CICS itself for data sets accessed in non-RLS mode

VSAM RLS, and sharing with batch

- Non-recoverable data sets
  - Can be read and updated from CICS and batch
- Recoverable data sets
  - Can be read and updated by CICS and read (only) by batch
  - RLS=NRI or RLS=CR in JCL for batch applications
- A batch job cannot open a recoverable data set for update in RLS mode
  - To update in batch
    - Switch to non-RLS mode (Resolve retained locks, Quiesce data set, open non-RLS)
    - Use DFSMStvs – Transactional VSAM

RLS Buffer Consistency Example



RLS Buffer Consistency Example

1. CICS1 issues READ UPDATE for record A
   - Reads buffer containing record A into CF cache and dataspace on MVS1
2. CICS2 issues READ UPDATE for record B
   - Reads the buffer containing B from the CF cache into the dataspace on MVS2. No need to get buffer from disk.
3. CICS2 issues READ for record A
   - Reads the record from the buffer in the dataspace on MVS2. No need to go to the CF cache.
4. CICS2 issues REWRITE of record B
   - Updates the buffer in the dataspace in MVS2 and in the CF cache. Buffer usually hardened to disk at this point, but depends on the request. Updating the buffer in the cache causes MVS1 to be marked to indicate the buffer has been updated. This is via hardware invalidation.
5. CICS1 issues REWRITE of record A
   - Because the buffer in the dataspace is now invalid, it is read back in from the CF cache, and record merge-redo is performed to merge change to record A into rest of buffer.

CICS and VSAM RLS Redpaper

Summary of performance measurements

- Best performance, in terms of CPU cost per transaction, is local VSAM LSR files
- When data is made available for sharing (via function shipping or RLS) the cost necessarily increases
- However, the net effect of migrating to RLS will depend on the original configuration
  - If migrating is from MRO, with a high proportion of requests being function-shipped across XCF links, could well see a **reduction** in overall CPU cost per transaction
  - If migrating from MRO/XM files or local files, then expect an increase in CPU cost per transaction (but relatively small compared with MRO/XM)
- RLS has better scaling capabilities than CICS function shipping because it is not limited to a single FOR that is constrained to the speed of a CP due to its single-TCB architecture.

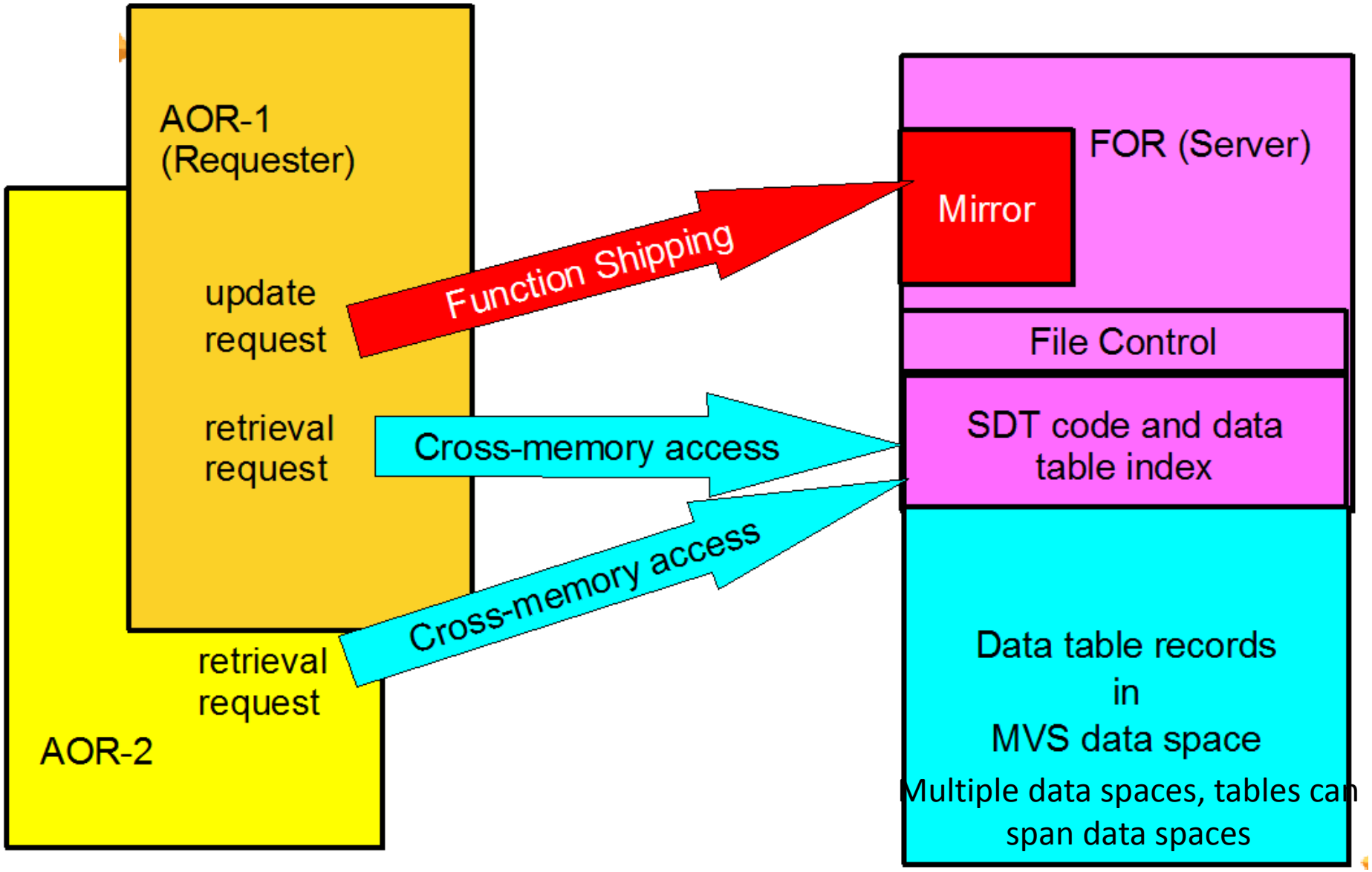
CICS Data Tables

- Shared Data Tables
  - Optimized for shared access within an MVS image
    - CICS-Maintained Data Table (CMT): high-performance for read-only operations; update operations via function shipping and VSAM
    - User-Maintained Data Table (UMT): high-performance for read-only operations across MVS image; high-performance for all allowed operations within owning region
- Coupling Facility Data Tables (CFDT)
  - Optimized for shared access to ‘informal’ data within a parallel sysplex
    - Somewhat akin to sysplex-shared UMT, but with update integrity
  - See also Shared Temporary Storage and Named Counters

Shared Data Tables

- In the beginning (CICS/ESA V3) was Data Tables – for direct access to keyed data within CICS region
- Shared Data Tables extended support to more of the File Control API, and across an MVS image
- Nowadays ‘data table’ = ‘shared data table’ = a table which supports the extended API and has the ***capability*** to be shared (but doesn’t have to be)
- CICS-Maintained data table
  - Updates always persisted to the underlying VSAM KSDS source data set
  - Automatically loaded from the source data set on open
- User-Maintained data table
  - Independent of source data set once loaded
  - Updates are made only to the UMT (giving speed and transaction recovery, but not persistence)
  - Persistence could be provided by the user
- User exits provided to control loading and decide which records are included in the table (XDTRD, XDTLC, XDTAD)

Shared Data Tables concepts



Shared Data Tables API

- SDT exploiting commands
  - API commands which exploit SDT support use cross-memory sharing for remote access, and use the data table for local access
    - READ (without UPDATE or RBA)
    - STARTBR, RESETBR, READNEXT, READPREV (without RBA option)
    - ENDBR (unless data table is a CMT and browse has already accessed the source data set)
- SDT supported commands
  - Supported but non-exploiting commands need to be function shipped for remote access, and need to access the source data set for a local CMT
    - READ UPDATE, REWRITE, DELETE (without RBA)
    - WRITE (without RBA or MASSINSERT)
    - Reads and browses to a CMT for records which are not in the data table
- Commands not in the SDT subset
  - A small subset of file control API commands are “not supported” – these access the underlying source data set for a CMT, and are rejected for a UMT
    - Any command with the RBA option
    - WRITE with MASSINSERT

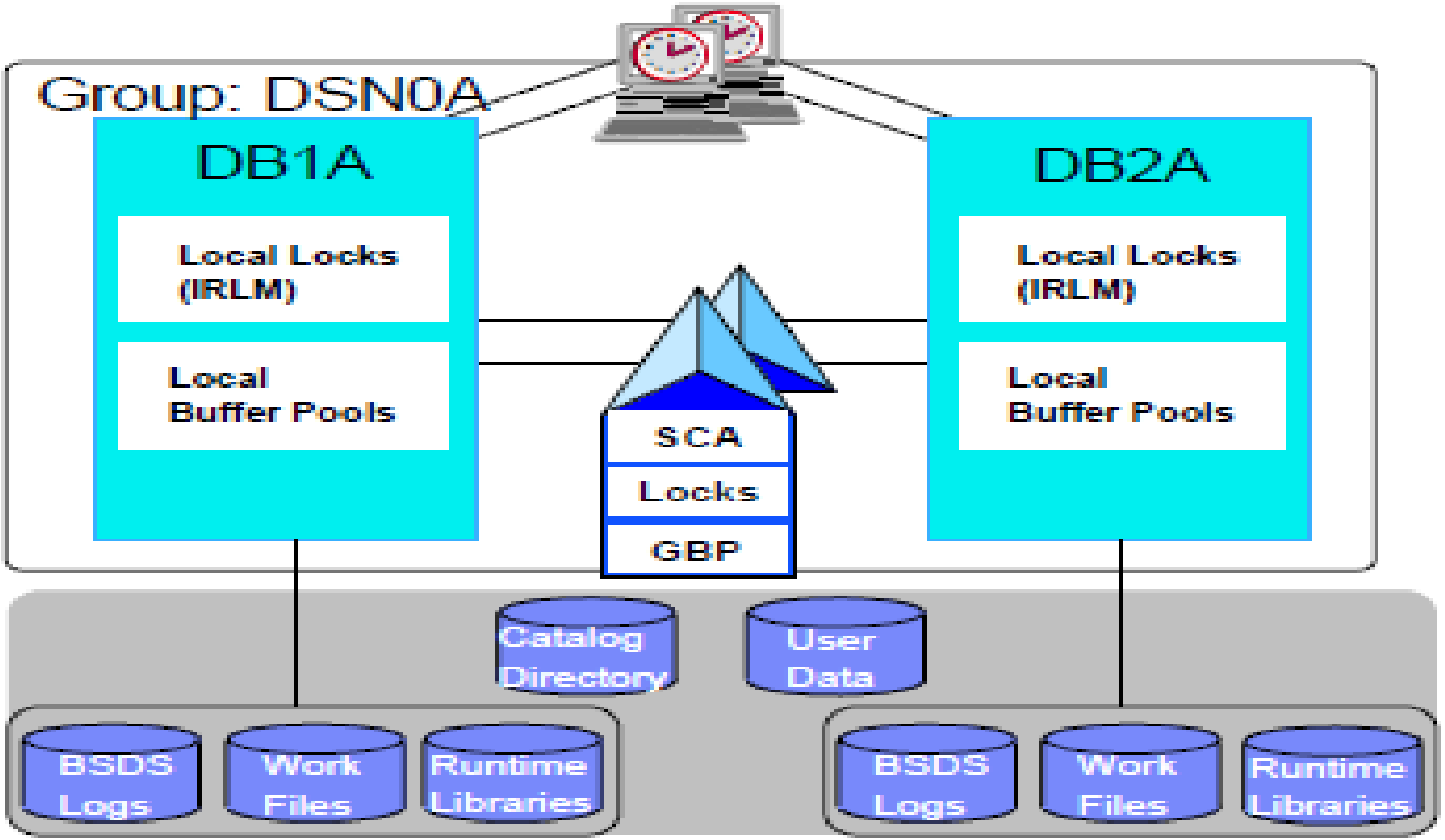
Candidates for data tables support

- Good candidates:
  - Files frequently accessed from one or more other regions **within a single MVS**, especially if predominantly reads or browses
  - For CMTs: Files accessed within a single region, but where the read-to-update ratio is high
  - UMTs: Files accessed within a single region with a large proportion of updates, but which do not require updates to be permanently saved
  - Files for which read performance is more critical than update performance
  - Files accessed in a sysplex which are either readonly, or updates do not need to be seen immediately in all MVS systems – could use UMTs with one per LPAR – see ‘Using Shared Data Tables in a Sysplex’ in Knowledge Center
- Poor candidates
  - Files accessed locally with a lot of update and/or browse requests – local VSAM browsing is quite efficient
  - Files accessed in a Sysplex – use VSAM RLS or CF data tables

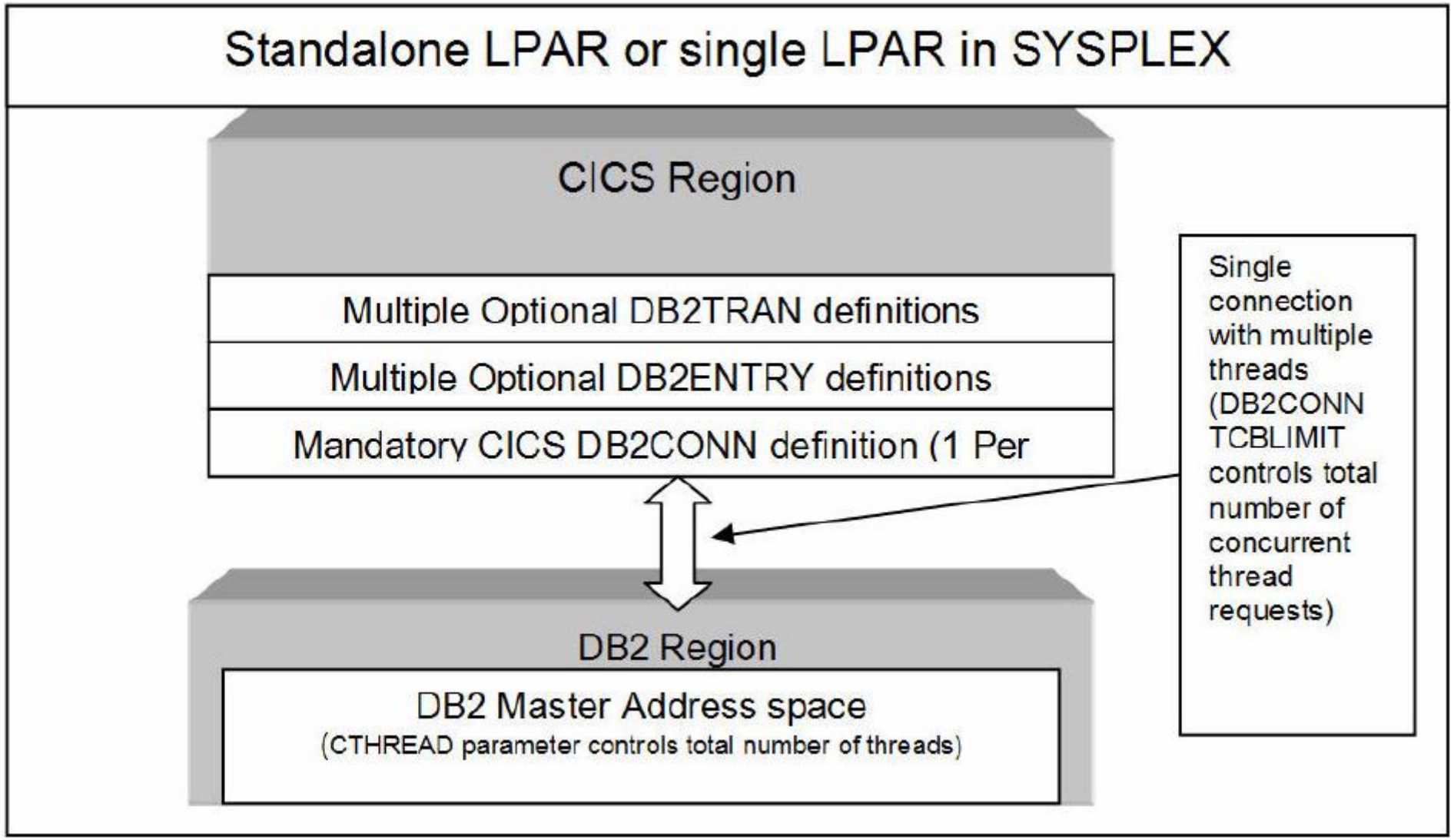
DB2 data sharing

- CICS connects to single DB2 instance at any one time
- Connection handled by DB2CONN resource
  - Ability to specify Group ID of data sharing group rather than single DB2 instance
  - Available DB2 automatically selected from available members in group
  - In event of DB2 connection loss CICS will attempt to reconnect to same instance
  - Can force reconnection to lost DB2 if in-doubts UOWs exist

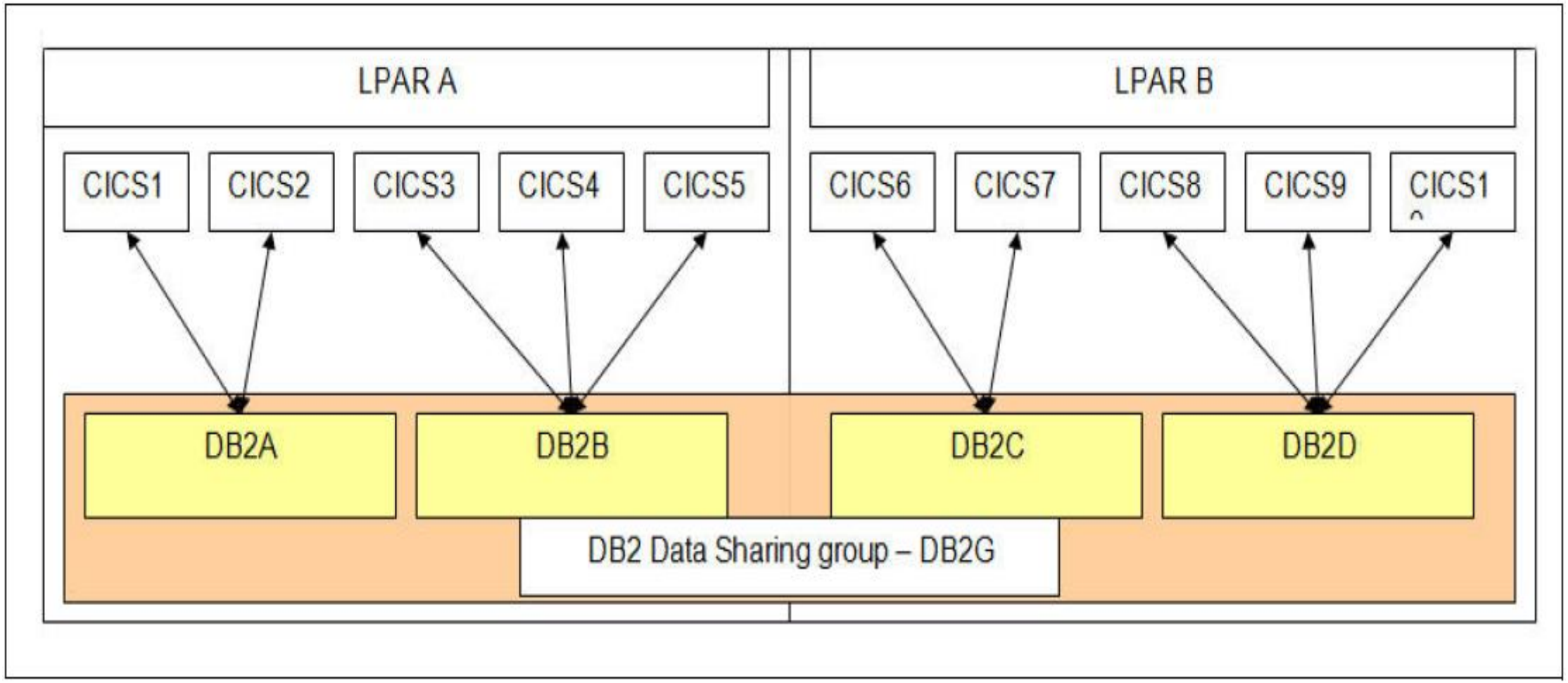
DB2 data sharing components



CICS and DB2 Stand-alone



CICS data sharing and CICS group attach



Coupling Facility Resources

- Named Counters
  - Efficiently generate unique sequence numbers for use in a Parallel Sysplex
  - Can also be used for sharing simple named pieces of data
- Shared Temporary Storage
  - CICS TS queues shared in a sysplex
  - Persist across CICS restart, but do not support CICS recovery
- Coupling Facility Data Tables
  - File data shared in a sysplex
  - Persist across CICS restart, can have recovery (from failures other than CF)
- CICSPlex SM Optimized Workload Management
  - Uses CF data tables, managed by a Region Status Server
- Logger - CICS uses MVS Log Streams for
  - Backout log: DFHLOG and DFHSHUNT
  - Forward recovery logs – logstreams named on data sets or files and DFHLGLOG (the log-of-logs)
  - User journals, including file autojournalling
  - SMF data, if configured to use log streams

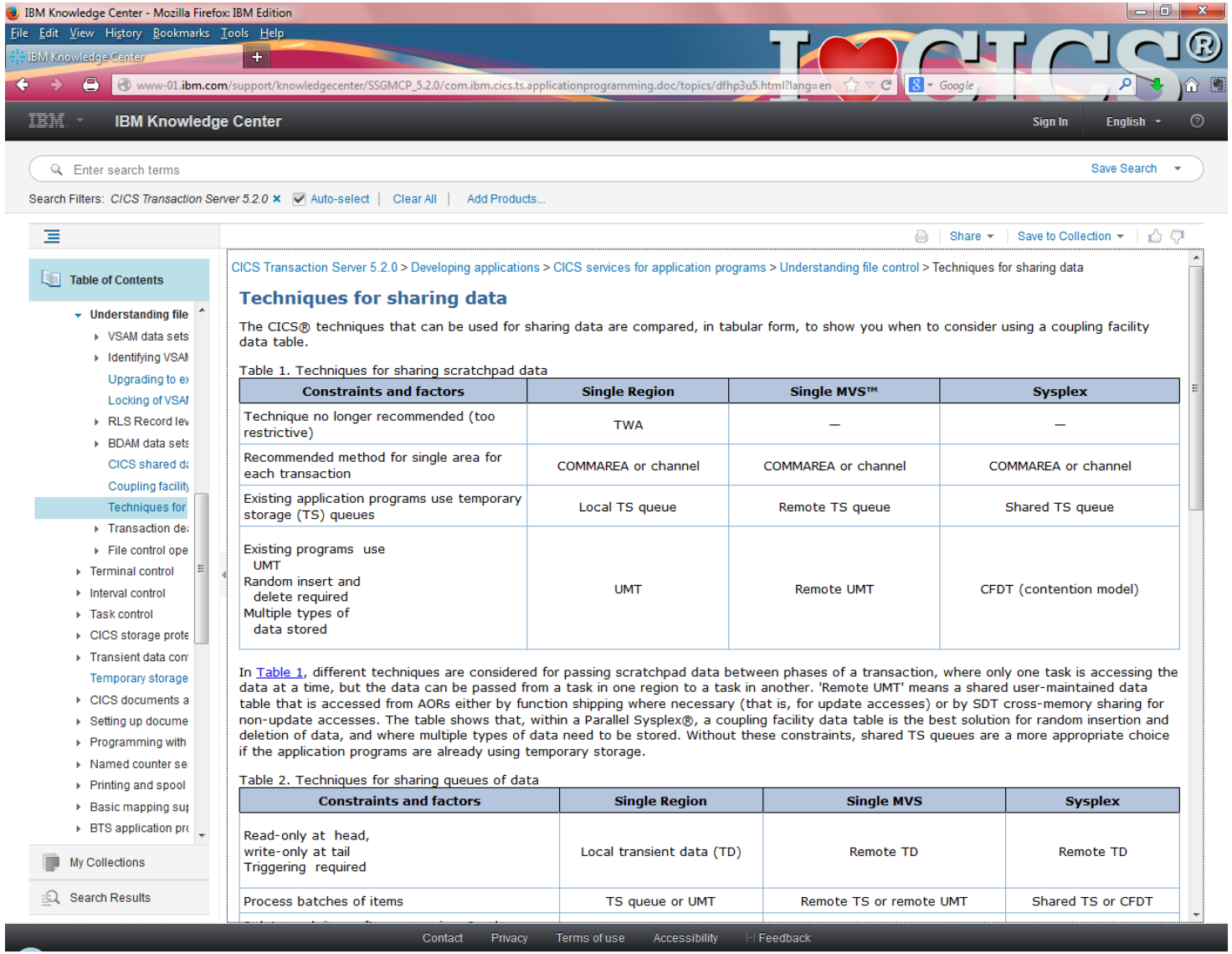
Data-Sharing Servers

- Shared TS queues, CF Data Tables and Named Counters each use their own data-sharing servers
- All use AXM system services
- Each server manages a Pool of data-sharing resources: a CF list structure
- CFDTs, TS queues, or named counters are allocated in a pool
  - Can have multiple pools for each type of resource e.g. to separate test from production
- Data-sharing resources survive CICS failure, but not failure of CF structure
  - Typical use is for non-permanent data
  - Can use CF Duplexing

Coupling Facility Data Tables

- Efficient sharing of informal data in a sysplex with update integrity
  - For scratchpad data, lookup-table, subset of data from larger file, data “belonging” to the user or terminal, etc.
- Accessed via File Control API
- Like UMTs, except
  - Can opt not to load from source data set
  - NOT deleted when all files closed
  - Speed is similar for reading and updating
- Two update models
  - Locking – records locked when read for update; can be recoverable (except across CF failure)
  - Contention – records not locked for update, but update rejected if record has been changed
- Restriction
  - Key length limited to 16 bytes (CF restriction)

When to use CFDT, TSq, NC



Topic 'Techniques for sharing data' in CICS InfoCenter or Knowledge Center

http://www.ibm.com/support/knowledgecenter/SSGMCP_5.2.0/com.ibm.cics.ts.applicationprogramming.doc/topics/dfhp3u5.html

Data Sharing Choices – a few things to consider

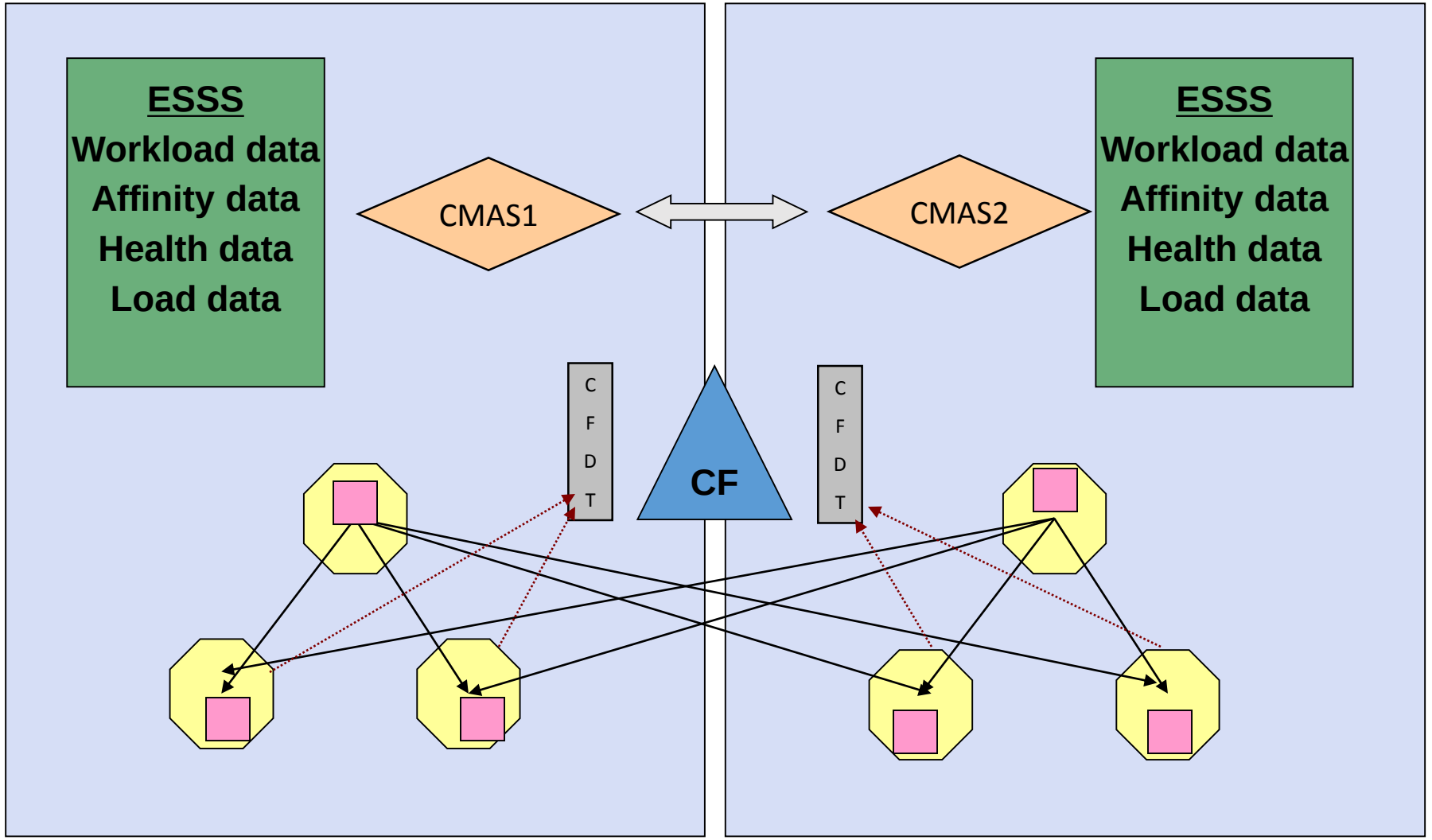
- DB2 has the most flexible and complete data sharing capabilities, especially for structured relational data
  - CICS VT can help with a transition from file-based applications to DB2 back-end data
- VSAM RLS allows data sharing of file data across CICS regions and sysplex-wide
  - Avoids the need for a single FOR
  - Good option for data sets to be accessed from multiple regions
  - Also for applications wishing to make use of more granular locking (for reduced contention) and/or read integrity and other API options
- Potential candidates for shared data tables
  - Require optimal performance for predominantly read access
  - Shared within single MVS
- Candidates for CF data tables, Named Counters, Shared TS
  - Scratchpad data, control records, keyed items, shared queues

CICSPlex SM Sysplex Optimized Workload- CICS TS V4.1

- Exploitation of the z/OS coupling facility to improve cross LPAR routing of CICS workloads
- Improved recognition of CICS region status for more efficient WLM routing decisions
- Additional WUI views that can assist with problem determination in route selection
- Video on routing performance between CICS V3.2 and V4.1
  - http://www.youtube.com/watch?v=RpS2h4jZdsU

95

CICSPlex SM Sysplex Optimized Workload…



96

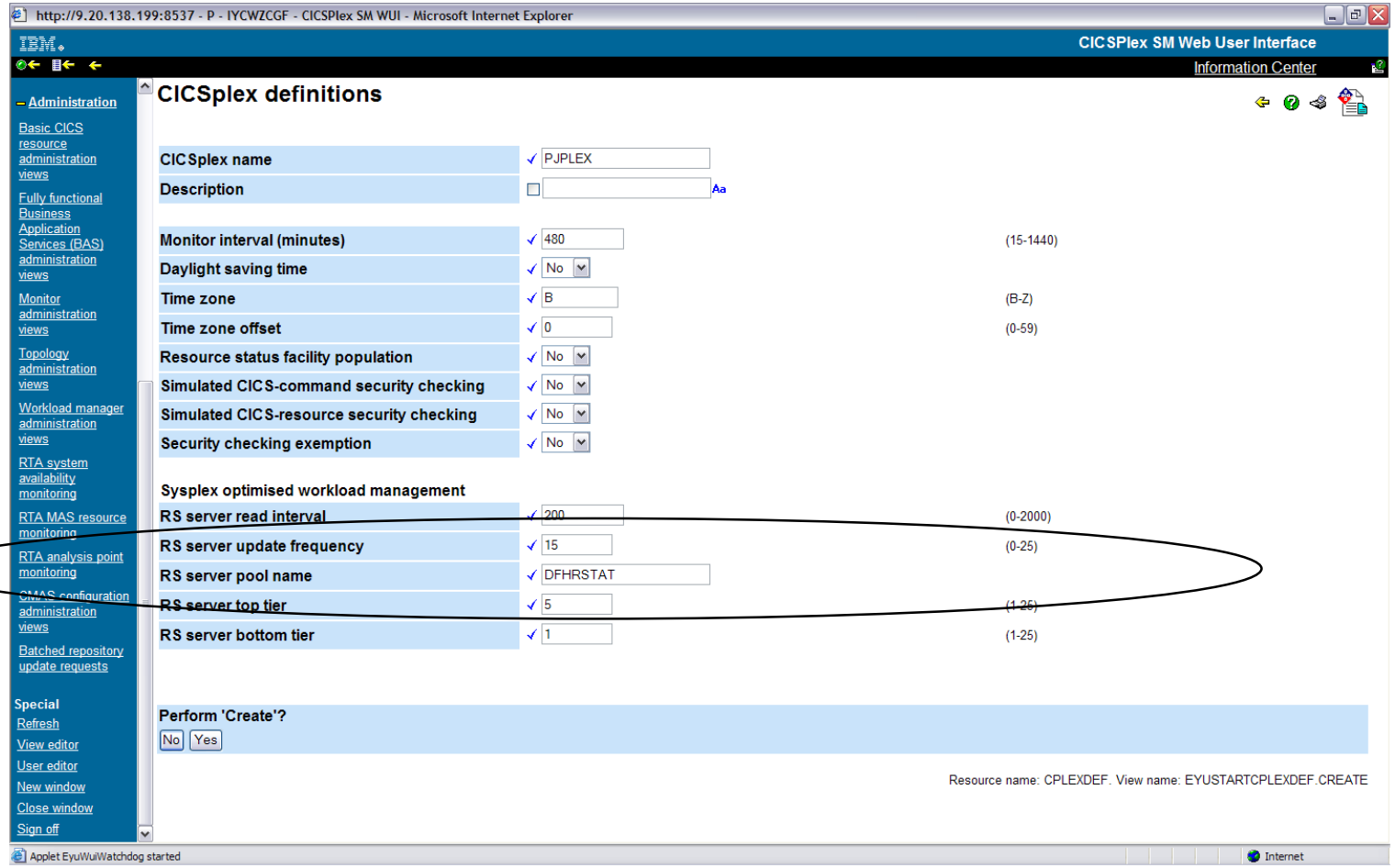
CICSPlex SM Sysplex Optimized Workload…

- CICS Region Status Domain
  - Exploitation of z/OS coupling facility
    - “Near real time” Sysplex-wide focus on target region status
    - No impact to “non-optimized” WLM
  - Optimized WLM routing enabled by configuring a Region Status Server
    - Uses CF Data Table to hold Region Status information
      - SOS, MaxTask, System or transaction dump in progress, Current task band
    - Shared by all routing regions (in the Sysplex)
  - Can be activated/deactivated dynamically
  - Called by XM (Transaction Manager), SM (Storage Manager) and DU (Dump Domain)

97

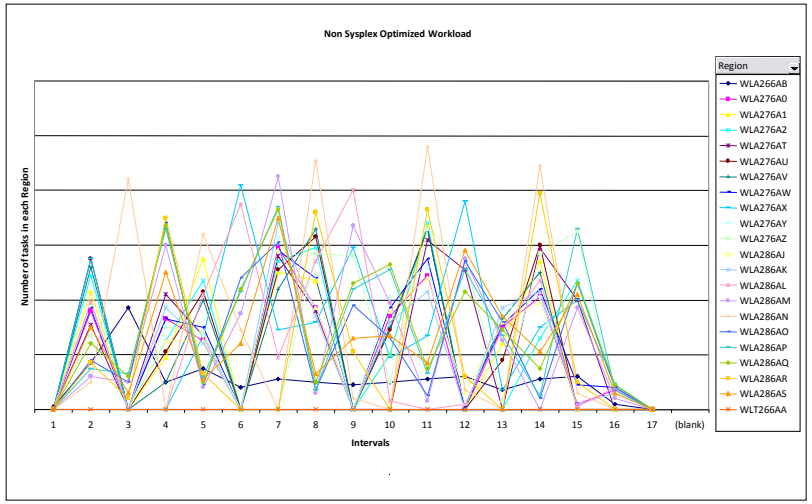
CICSPlex SM Sysplex Optimized Workload…

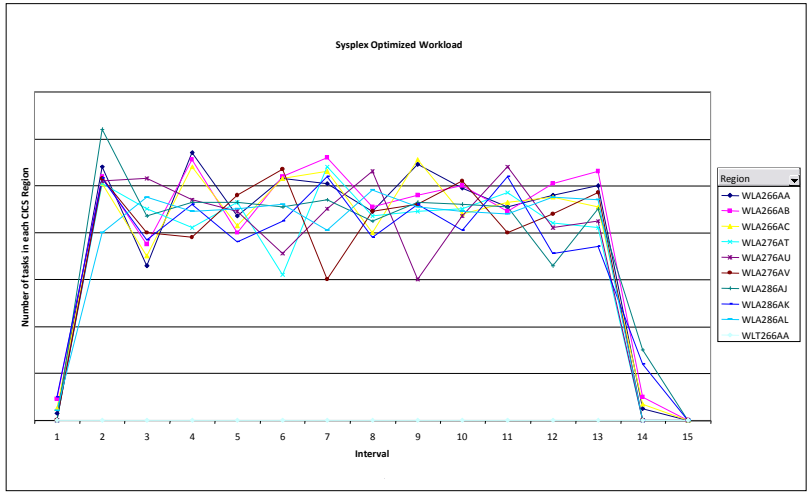
- CPLEXDEF



98

CICSPlex SM WLM: Optimized versus Non-optimized Routing





- Non-optimized Routing
  - 27 CICS regions on 3 LPARs
- Optimized Routing
  - 9 CICS regions on 3 LPARs
- Comparison
  - Workload is 10K started transactions
  - Number of tasks in each region measured every 10 second interval
- Results
  - Optimized results shows smoother distribution of work
  - Higher throughput esp. for async/STARTs
  - Better resilience for CMAS failures

Tooling

- Automation
- Real time monitoring
  - System level monitoring
  - Application level monitoring
- Post processing performance data
  - RMF, Tivoli Omegamon, CICS Performance Analyser (PA)
- Affinity identification and elimination
  - CICS Interdependency Analyser (IA)
- Multi-processor exploitation
  - OTE – threadsafe analysis and conversion (IA|PA)
- Resource management and control
  - CICS Configuration manager (CM)
- CICS visualisation
  - CICS Deployment Assistance (DA)
- VSAM and DB2
  - CICS VR | CICS VT

Continuous Operation

Continuous operations

- Cloud
- CICS continuous operation
- Maintenance
- Upgrade

CICS Cloud

- Based on CICSPlex/SM
  - Flexible management of CICS across Sysplex
  - Dynamic workload management
- Applications, platforms and policies
  - New model for packaging of applications
  - management of application lifecycle
- Policies allows non-invasive monitoring and actioning
  - At Operation, Application, CICS or platform level
  - Action – message, abend, event or program initiation

CICS continuous operation

- Near continuous operation of CICS provides good HA story
- Ability to quiesce CICS regions from workload for maintenance
- Large number of operations possible without stopping CICS
  - More online commands to adjust environment
  - Stats and monitoring provide ability to analyse systems
  - VSCR and 64 bit exploitation allows more to run in single CICS region

CICS maintenance

- CICS follows the RSU model
  - RSU (Recommended Service Upgrade)
- CICS participates in the CST testing cycle
  - CST (Consolidated Service Test)
- Service can normally be applied using rolling ‘upgrade’
  - Minimises or removes planned outages
- When special action required PTF will be have hold action
  - Extremely rare that hold action will require sysplex wide outage

CICS Upgrade

- **CICS Editions**
  - CICS TS Base
  - CICS TS VUE – Value Unit Edition
    - Runs on zNALC LPAR
    - Created to run specific workloads using different pricing model
  - CICS TS Developer trial
    - Try before you buy version
    - Does not start the dual license charge ‘clock’
    - Full function version of CICS with ‘kneecapped’ MXT – Only for use in Dev, Test
    - Changed in CICS TS 5.2 to allow ‘switch over’ to TS BASE / VUE edition without reinstall

- CICS TS 5.2 introduces new ‘activation’ mechanism
  - APF authorised libraries contain code activation modules
    - SDFHLIC for CICS TS for z/OS V5.2 - activation module
    - SDFHDEV for CICS TS for z/OS Developer Trial V5.2 - activation module
    - SDFHVUE for CICS TS for z/OS Value Unit Edition V5.2 - activation module
  - Switch between versions by changing the activation module library

CICS TS V5 Vision

IBM CICS Transaction Server V5

is the premier enterprise grade mixed language application server

Service Agility

Runtime support for production ready web applications.

Integration with mobile applications.

Operational Efficiency

Increased system capacity and capability to achieve more with less.

Automatic control of critical resources using policies.

Cloud Enablement

Simplified system management and rapid application deployment.

On premise pattern based cloud deployment.

CICS Transaction Server V5.1

Service Agility

A production-ready web container, built on IBM WebSphere Application Server Liberty Profile, for the deployment of lightweight Java servlets and Java Server Pages (JSPs).

Operational Efficiency

Greater capacity is delivered through significant vertical and horizontal scalability enhancements. Fewer regions can now run the same workload.

Cloud Enablement

First-class applications enable the creation of agile services from existing assets. First-class platforms enable applications to be rapidly promoted from development and test environments into production.

CICS Transaction Server V5.2

1. Integrates mobile capabilities and broadens the web programming model, ***without requiring additional middleware***

2. Support distributed security standards and automatically safeguard critical runtime resources, ***without requiring additional system code***

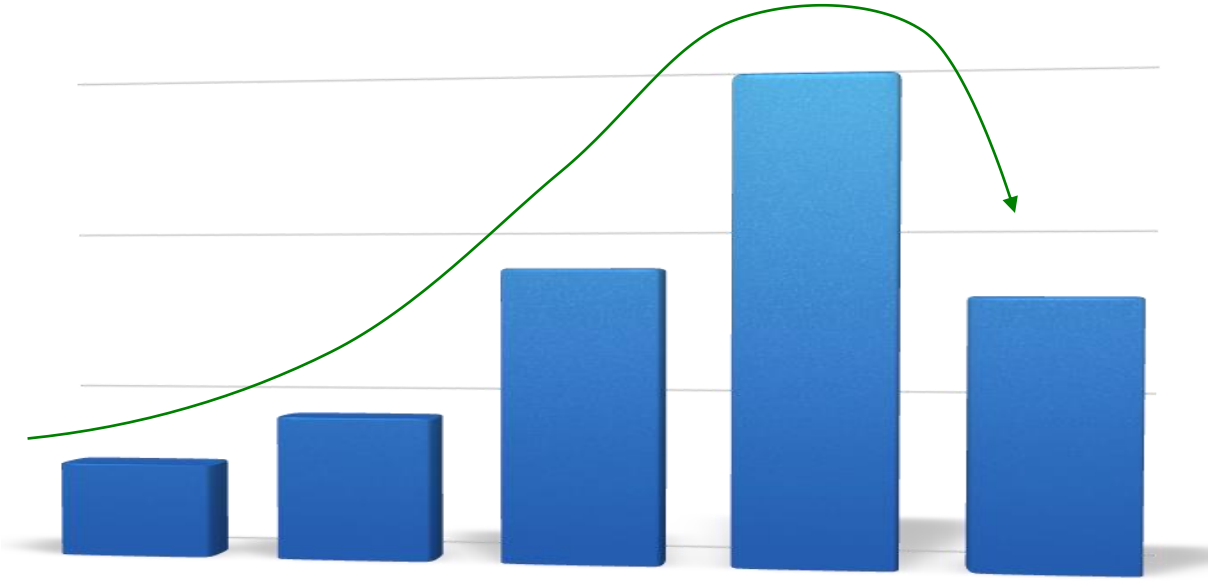
3. Deploy and manage several versions of a multi-program application and run them concurrently, ***without requiring additional regions***

* Continued enhancements to the Technical Foundation

Service Agility

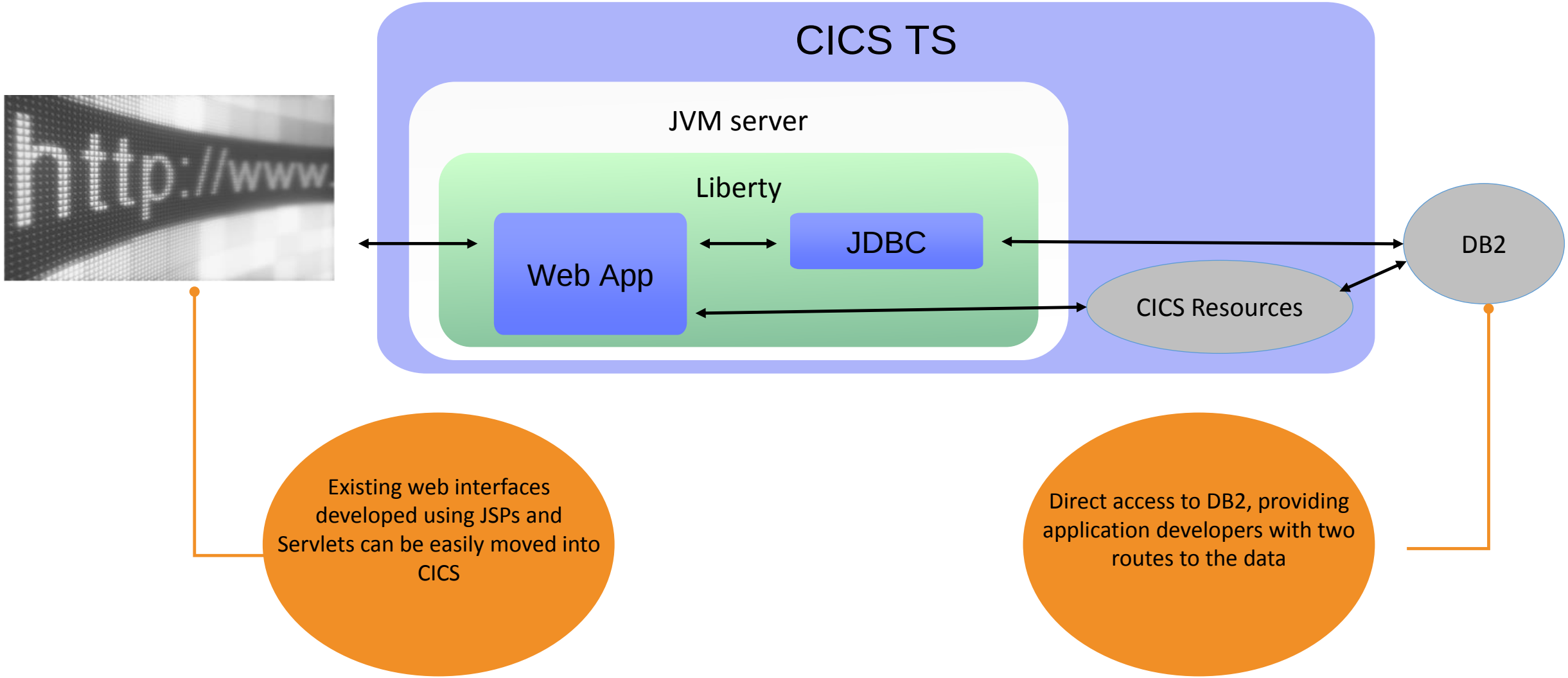
1

Integrates mobile capabilities and broadens the web programming model, ***without requiring additional middleware***



Number of systems

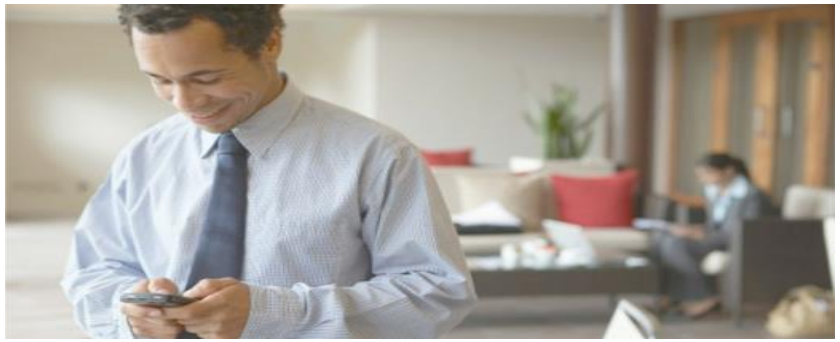
Liberty Web Apps in CICS TS V5.2



Mobile scenario

Connect directly to CICS from Worklight using end-to-end JSON

```
function getCustomerDetails(custNum) {
    var pathURL = "GENAPP/getCustomerDetails";
    var request=
        {
            "LGICUS01Operation":{
                "ca" : {
                    "ca_request_id" : "01ICUS",
                    "ca_return_code" : "00",
                    "ca_customer_num" : custNum,
                    "ca_num_policies" : ""
                    }
            }
        };
```



A

Step 1

DFHLS2JS to generate the JSON artefacts for the target CICS service.

B

Step 2

Mobile developer uses JSON schema to build a Worklight adapter.

C

Step 3

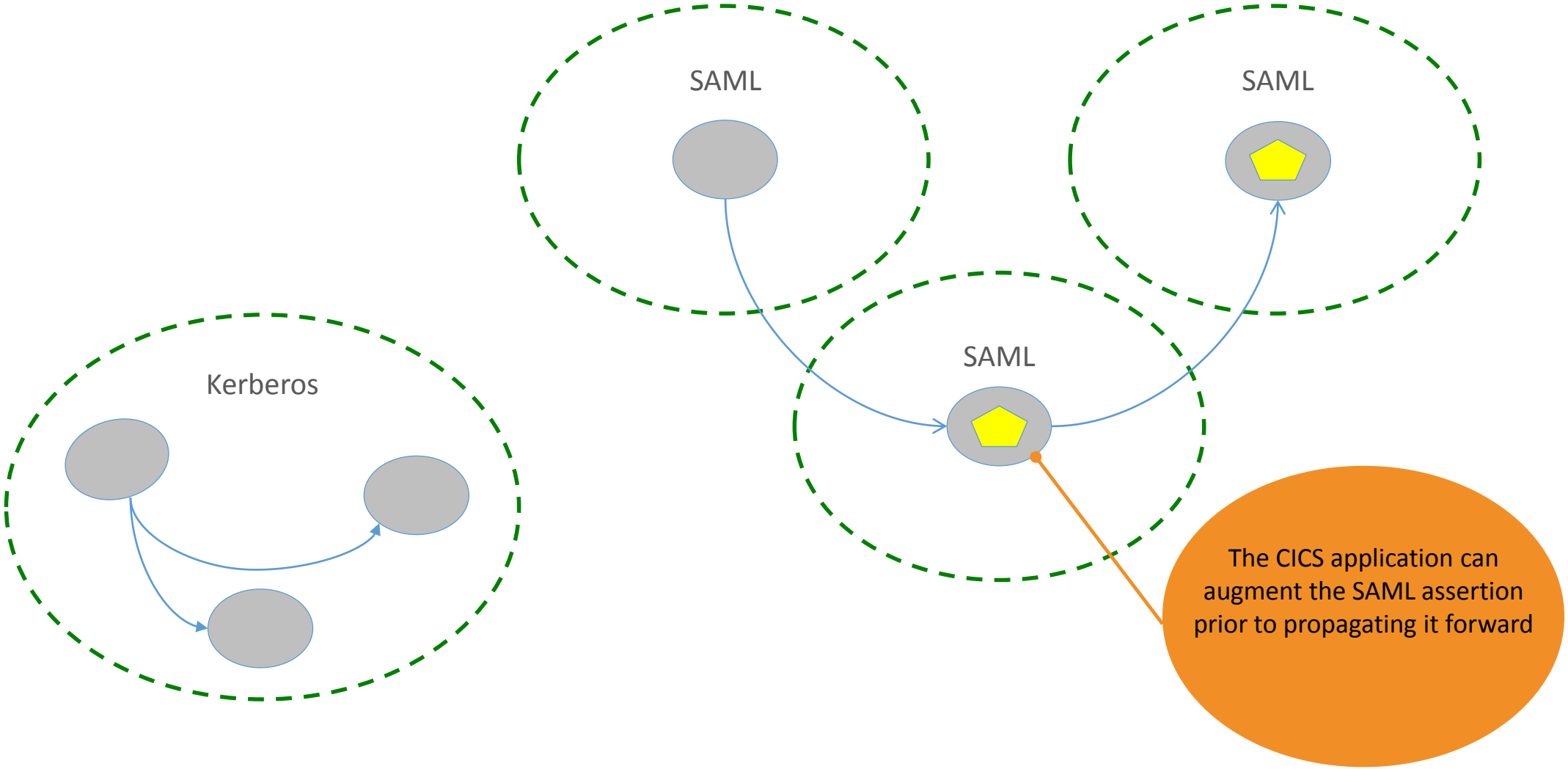
Frontend mobile developer calls the Worklight adapter which calls the service hosted in CICS.

Operational Efficiency

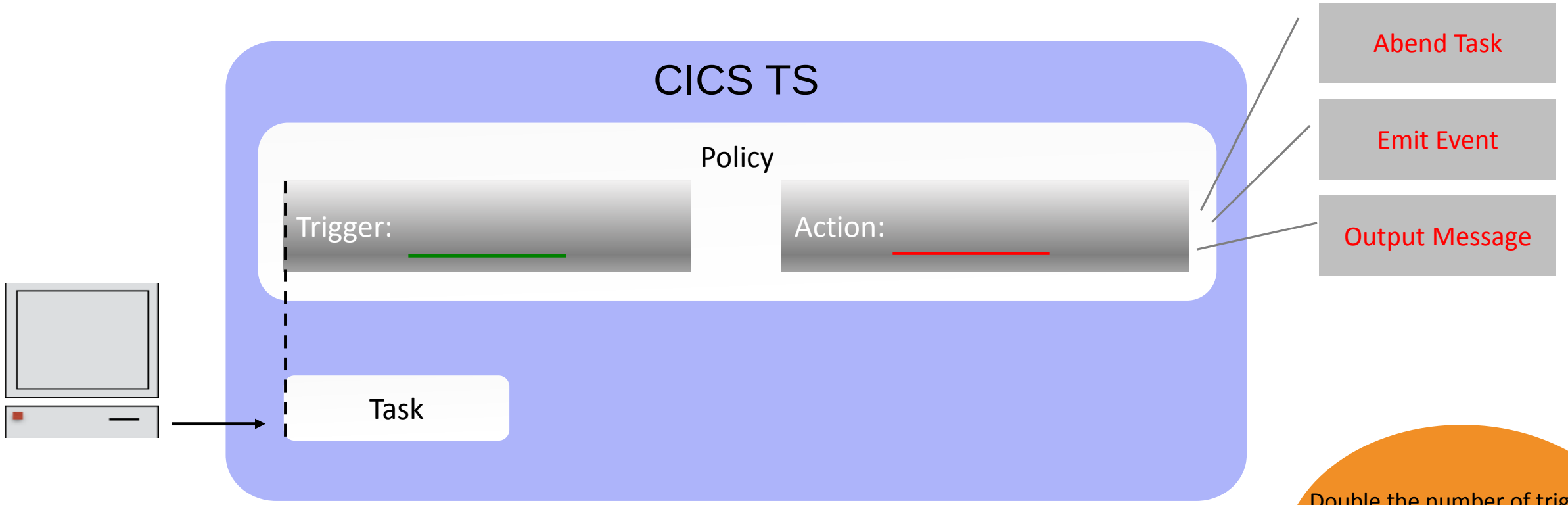
2

Support distributed security standards and automatically safeguard critical runtime resources, ***without requiring additional system code***

SAML and Kerberos security



Double the number of policy triggers



New Triggers	Cumulative Time	Syncpoint	Start	TSQ/TDQ
CICS V5.1	DB2 SQL	Files	Storage	CPU Time

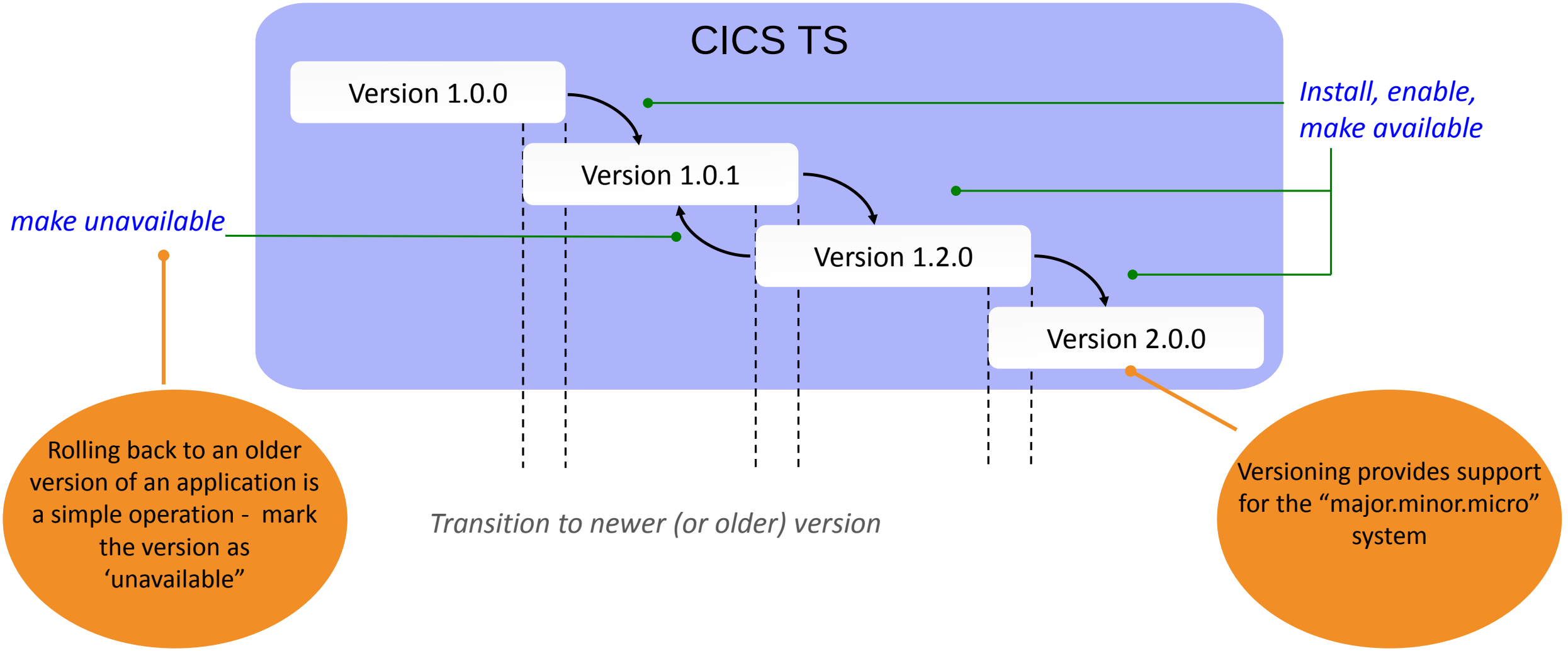
Double the number of triggers are available to provide greater level of control to CICS

Cloud Enablement

3

Deploy and manage several versions of a multi-program application and run them concurrently, ***without requiring additional regions***

Multi-versioning



Multi-versioning

Application packaging helps you consolidate onto fewer regions by providing private program definitions

CICS TS

Private programs

PROG1

URIMAP

Payroll App

LIBRARY

URIMAP

Customer App

LIBRARY

PROG1

Private programs

X

Y

Additional material

- Redbooks
  - **Improving z/OS Application Availability by Managing Planned Outages**
  - http://www.redbooks.ibm.com/redpieces/abstracts/sg248178.html?Open

CICS Disaster Recovery (DR)

IT Resilience

- IBM defines *IT resilience* as the ability to rapidly adapt and respond to any internal or external disruption, demand, or threat, and continue business operations without significant impact.
- IT resilience is related to, but broader in scope, than *disaster recovery.* Disaster recovery concentrates solely on recovering from an *unplanned* event.

Two things must be considered:

- Recovery Time Objective (RTO)
  - This term refers to *how long* your business can afford to wait for IT services to be resumed following a disaster.
  - If this number is not clearly stated now, think back to the last time you had a significant service outage. How long was that outage, and how much pain did your company suffer as a result?
  - This will help you get a sense of whether to measure your RTO in days, hours, or minutes.
- Recovery Point Objective (RPO)
  - This term refers to *how much data* your company is willing to have to recreate following a disaster.
  - In other words, what is the acceptable time difference between the data in your production system and the data at the recovery site?

Typical achievable RPO and RTO for some common DR options

Description	Typically achievable Recovery Point Objective (RPO)	Typically achievable Recovery Time Objective (RTO)
No disaster recovery plan	N/A - all data lost	N/A
Tape vaulting	Measured in days since last stored backup	Days
Electronic vaulting	Hours	Hours (hot remote location) to days
Active replication to remote site (w/o recovery automation)	Seconds to minutes	Hours to days (dependent on availability of recovery hardware)
Active storage replication to remote "in-house" site	Zero to minutes (dependent on replication technology and automation policy)	1 or more hours (dependent on automation)
Active software replication to remote 'active' site	Seconds to minutes	Seconds to minutes (dependent on automation)

GDPS

- GDPS is actually a collection of several offerings, each addressing a different set of IT resiliency goals, that can be tailored to meet the RPO and RTO for your business.
- Each offering leverages a combination of server and storage hardware or software-based replication and automation and clustering software technologies

GDPS offerings

- GDPS/PPRC
  - Near-CA or DR solution across two sites separated by metropolitan distances. The solution is based on the IBM PPRC synchronous disk mirroring technology.
- GDPS/PPRC HyperSwap Manager
  - Near-CA solution for a single site or entry-level DR solution across two sites separated by metropolitan distances. The solution is based on the same technology as GDPS/PPRC, but does not include much of the systems automation capability that makes GDPS/PPRC a more complete DR solution.
- GDPS/XRC
  - DR solution across two sites separated by virtually unlimited distance between sites. The solution is based on the IBM XRC asynchronous disk mirroring technology (also branded by IBM as z/OS Global Mirror).
- GDPS/Global Mirror
  - DR solution across two sites separated by virtually unlimited distance between sites. The solution is based on the IBM System Storage® Global Mirror technology, which is a disk subsystems-based asynchronous form of remote copy.

GDPS offerings

- GDPS Metro/Global Mirror
  - A three-site solution that provides CA across two sites within metropolitan distances and DR to a third site at virtually unlimited distances. It is based on a cascading mirroring technology that combines PPRC and Global Mirror.
- GDPS Metro/z/OS Global Mirror
  - A three-site solution that provides CA across two sites within metropolitan distances and DR to a third site at virtually unlimited distances. It is based on a multitarget mirroring technology that combines PPRC and XRC (also known as z/OS Global Mirror on IBM storage subsystems).
- GDPS/Active-Active
  - A multisite CA/DR solution at virtually unlimited distances. This solution is based on software-based asynchronous mirroring between two active production sysplexes running the same applications with the ability to process workloads in either site.