CICS **INSIGHT** Series

August 9th , 2023

11:00 – 13:00 EST

Lifting the Lid on CICS Temporary Storage

Andy Wright and Darren Beard

For enrollment, visit
https://ibm-zcouncil.com/events/insight-aug-9/

IBM

Notices and disclaimers

- This presentation is provided by IBM Corporation. Copyright © IBM Corporation, 2023. Redistribution is prohibited.
- U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. This document is distributed "as is" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.
- Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and disclaimers continued

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of 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. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.

- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM, the IBM logo, ibm.com, AIX, BigInsights, Bluemix, CICS, Easy Tier, FlashCopy, FlashSystem, GDPS, GPFS, Guardium, HyperSwap, IBM Cloud Managed Services, IBM Elastic Storage, IBM FlashCore, IBM FlashSystem, IBM MobileFirst, IBM Power Systems, IBM PureSystems, IBM Spectrum, IBM Spectrum Accelerate, IBM Spectrum Archive, IBM Spectrum Control, IBM Spectrum Protect, IBM Spectrum Scale, IBM Spectrum Storage, IBM Spectrum Virtualize, IBM Watson, IBM Z, IBM z Systems, IBM z13, IBM z14, IMS, InfoSphere, Linear Tape File System, OMEGAMON, OpenPower, Parallel Sysplex, Power, POWER, POWER4, POWER7, POWER8, Power Series, Power Systems, Power Systems Software, PowerHA, PowerLinux, PowerVM, PureApplication, RACF, Real-time Compression, Redbooks, RMF, SPSS, Storwize, Symphony, SystemMirror, System Storage, Tivoli, WebSphere, XIV, z Systems, z/OS, z/VM, z/VSE, zEnterprise and zSecure are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

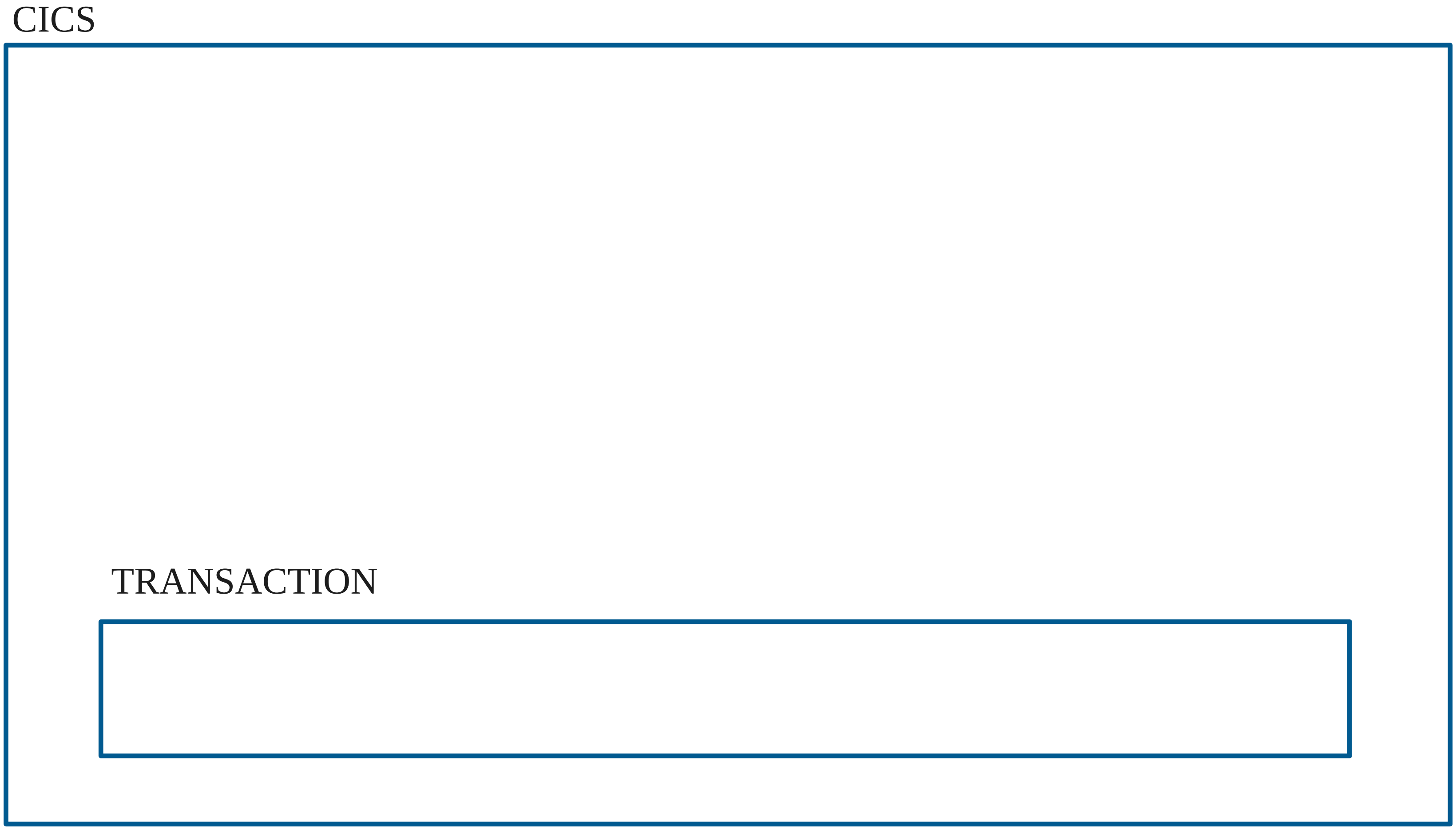
Agenda

- What is Temporary Storage?
- Main Temporary Storage
- Auxiliary Temporary Storage
- Shared Temporary Storage
- Definitions – TSMODEL
- Local and remote Temporary Storage queue support
- Temporary Storage recovery considerations
- 64 bit enhancements for Temporary Storage
- Temporary Storage automated cleanup
- Temporary Storage diagnostics
- Historical problems
- Temporary Storage performance considerations

What is Temporary Storage?

- CICS has provided a temporary storage component "forever"
- Originally written in a hybrid of assembler and PLS
  - DFHTSP and DFHTSPA
- Restructured and rewritten as OO PLX in CICS TS 1.1
  - The TS domain
- It is provided as a scratchpad for short-lived data
- 1-n items (records) are held on unique queues
- It is used by applications and by CICS
  - The EXEC CICS API and the internal domain calls
- It can be recoverable and may be shared between CICS regions

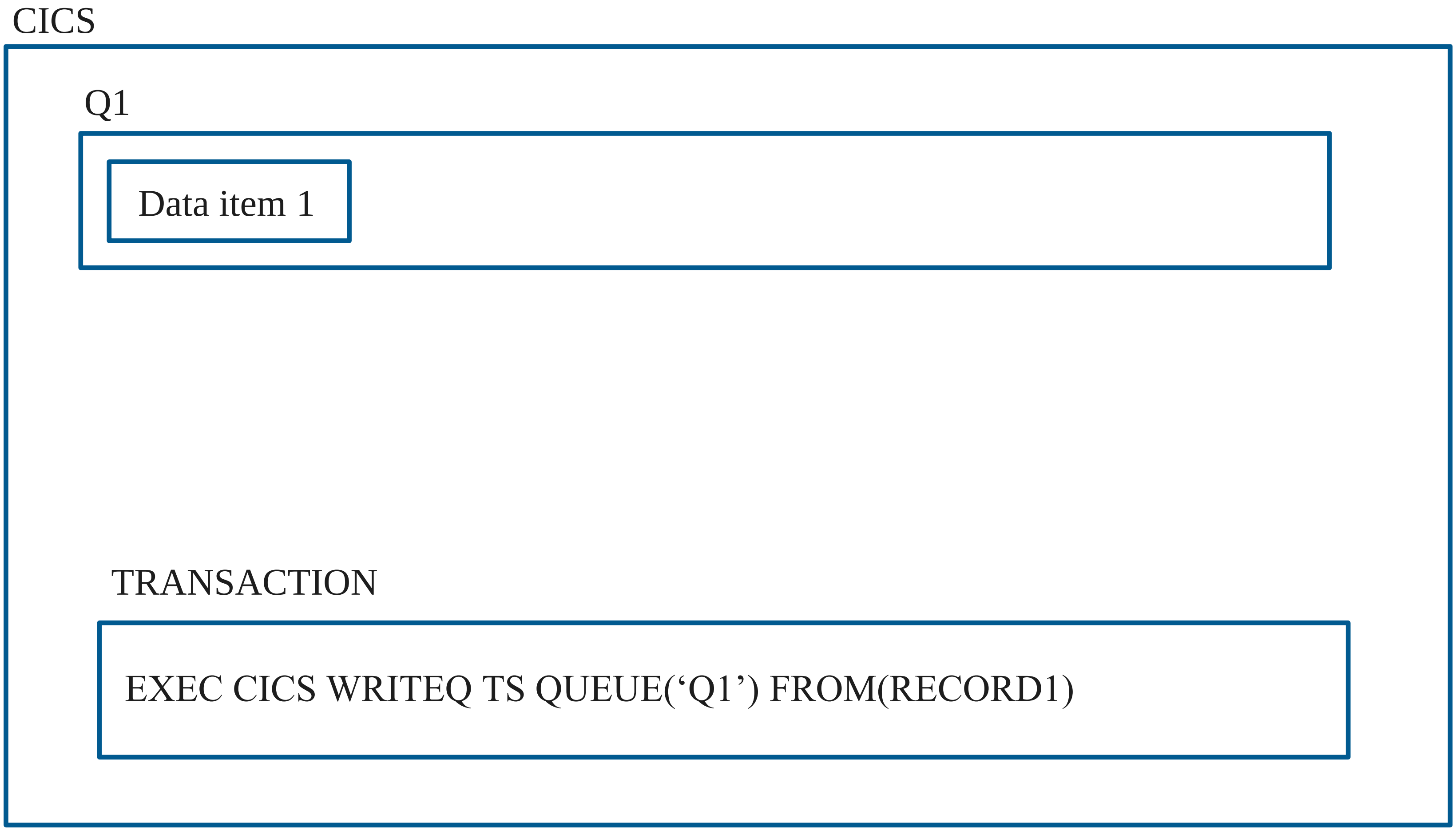
A Temporary Storage queue



CICS

TRANSACTION

A Temporary Storage queue



CICS

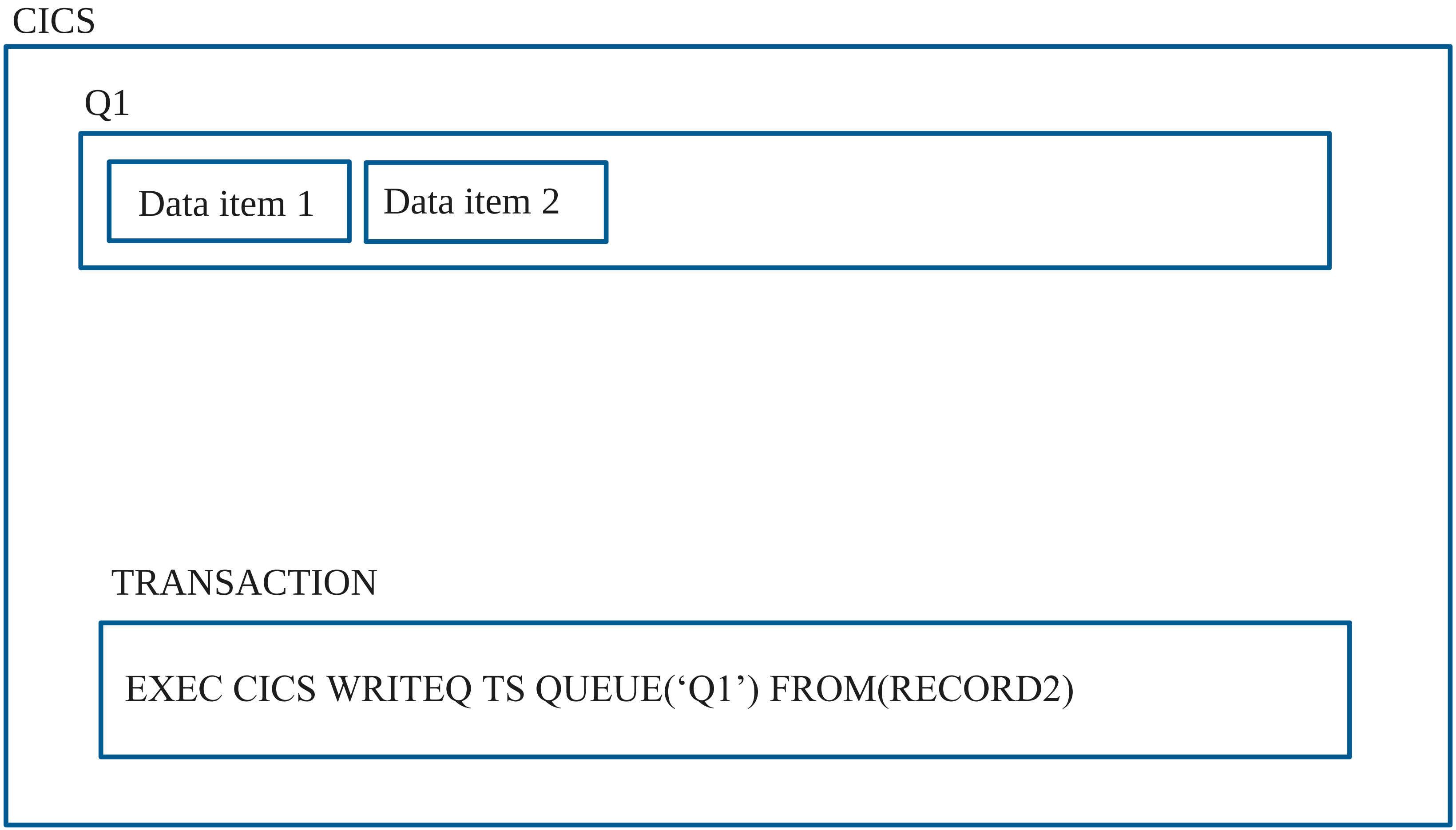
Q1

Data item 1

TRANSACTION

```
EXEC CICS WRITEQ TS QUEUE(‘Q1’) FROM(RECORD1)
```

A Temporary Storage queue



CICS

Q1

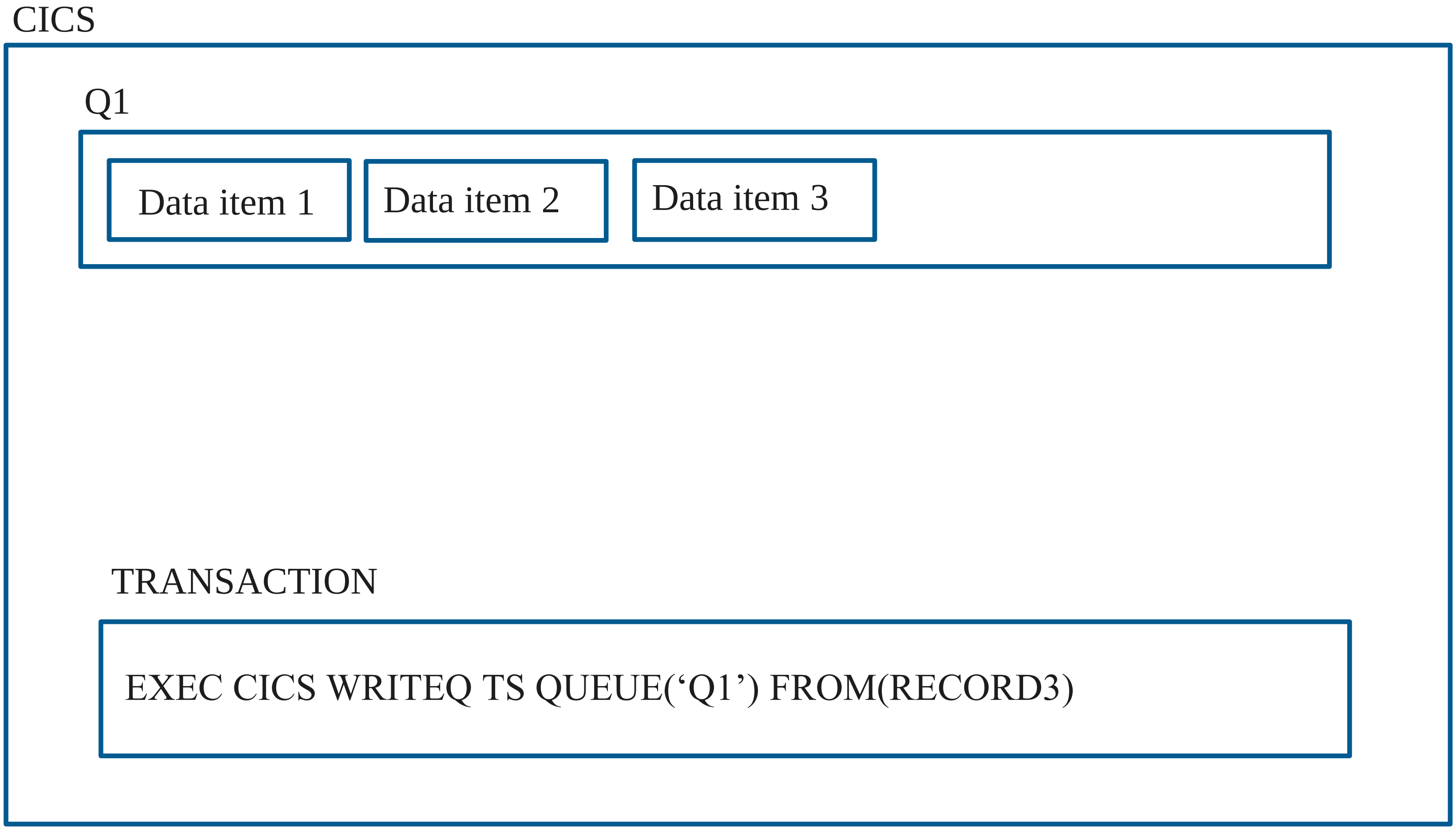
Data item 1

Data item 2

TRANSACTION

```
EXEC CICS WRITEQ TS QUEUE(‘Q1’) FROM(RECORD2)
```

A Temporary Storage queue



CICS

Q1

Data item 1

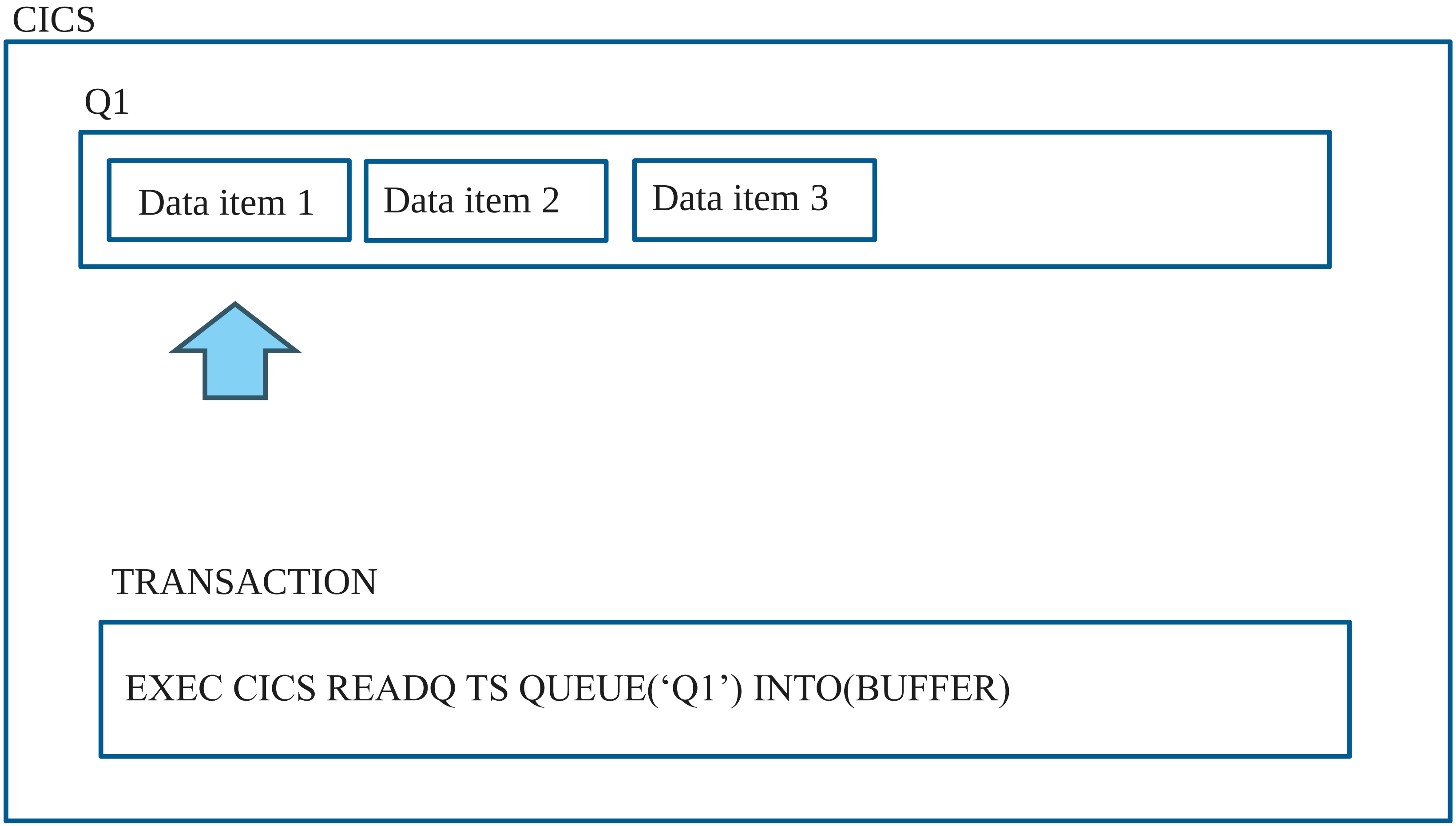
Data item 2

Data item 3

TRANSACTION

```
EXEC CICS WRITEQ TS QUEUE(‘Q1’) FROM(RECORD3)
```

A Temporary Storage queue



CICS

Q1

Data item 1

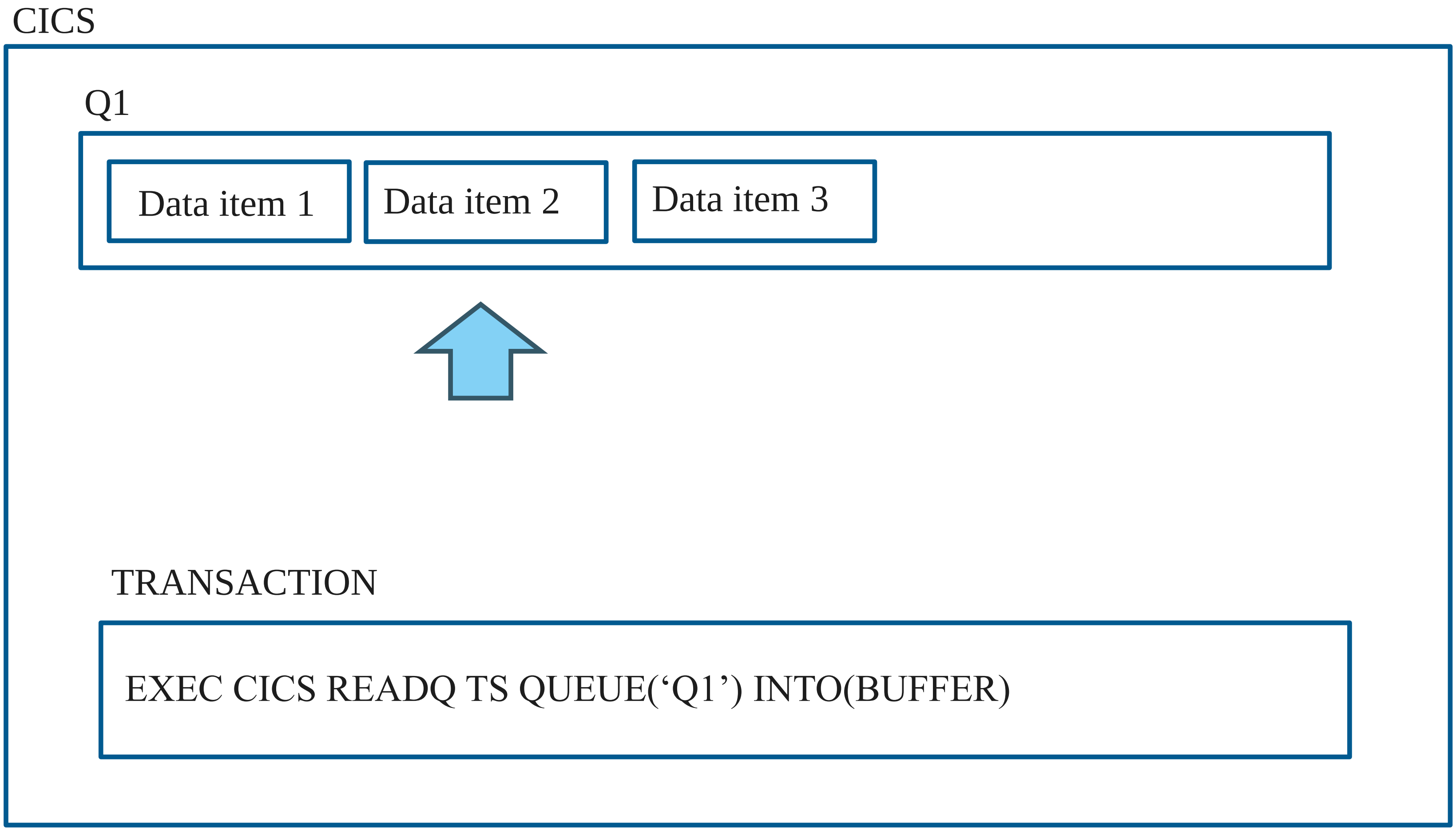
Data item 2

Data item 3

TRANSACTION

```
EXEC CICS READQ TS QUEUE(‘Q1’) INTO(BUFFER)
```

A Temporary Storage queue



CICS

Q1

Data item 1

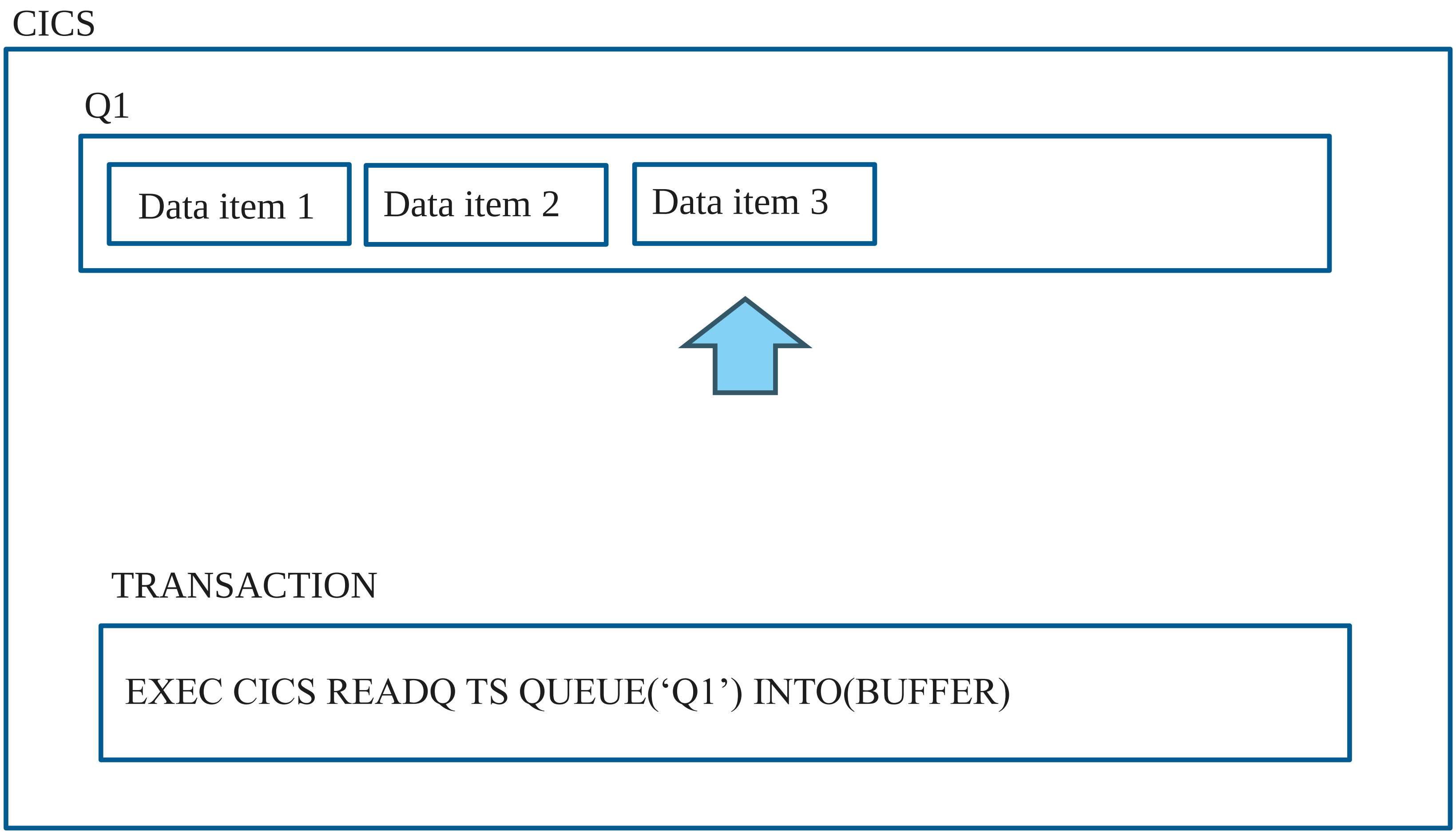
Data item 2

Data item 3

TRANSACTION

EXEC CICS READQ TS QUEUE(‘Q1’) INTO(BUFFER)

A Temporary Storage queue



CICS

Q1

Data item 1

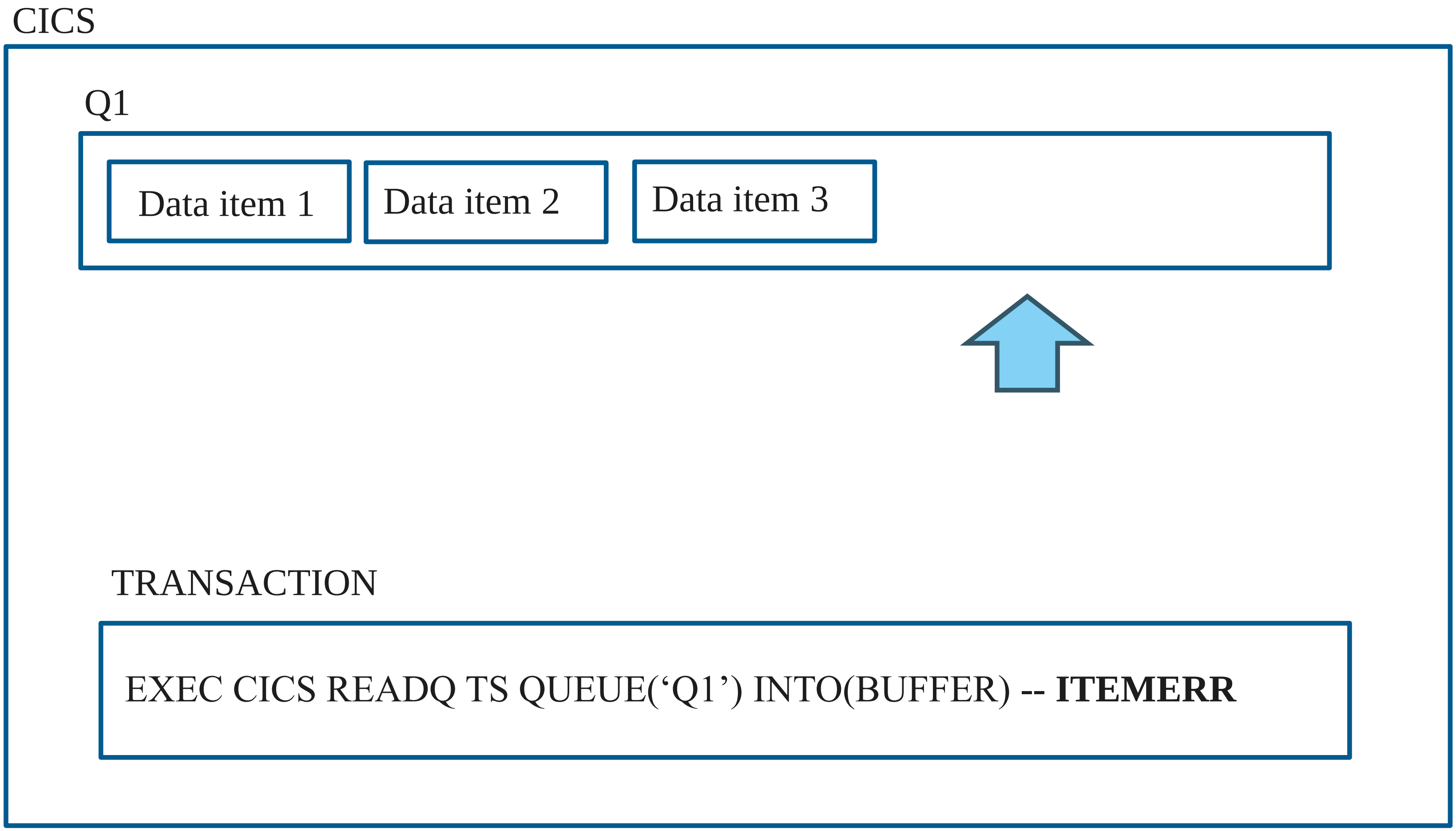
Data item 2

Data item 3

TRANSACTION

```
EXEC CICS READQ TS QUEUE(‘Q1’) INTO(BUFFER)
```

A Temporary Storage queue



CICS

Q1

Data item 1

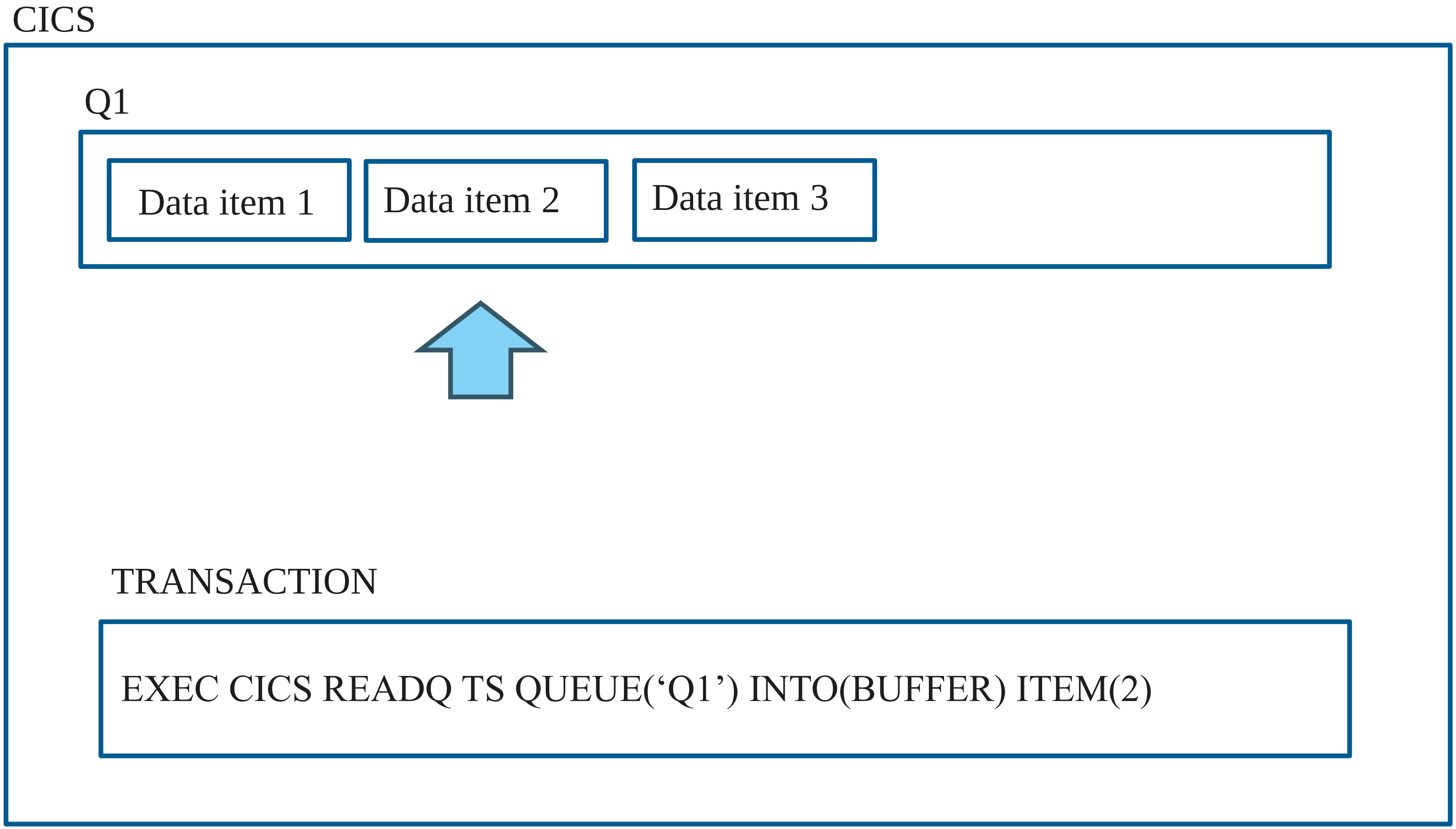
Data item 2

Data item 3

TRANSACTION

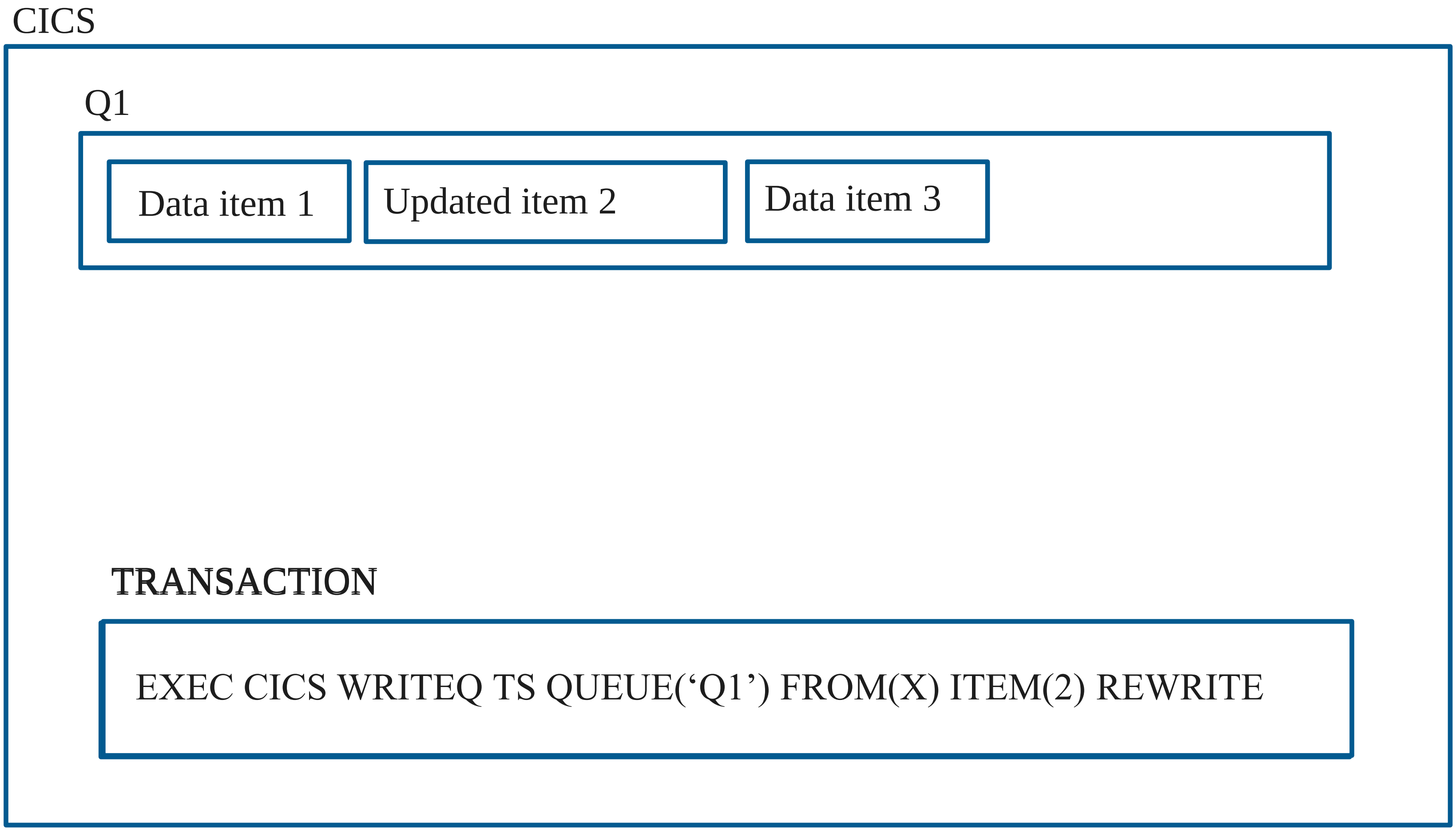
EXEC CICS READQ TS QUEUE(‘Q1’) INTO(BUFFER) -- **ITEMERR**

A Temporary Storage queue



```
EXEC CICS READQ TS QUEUE(‘Q1’) INTO(BUFFER) ITEM(2)
```

A Temporary Storage queue



CICS

Q1

Data item 1

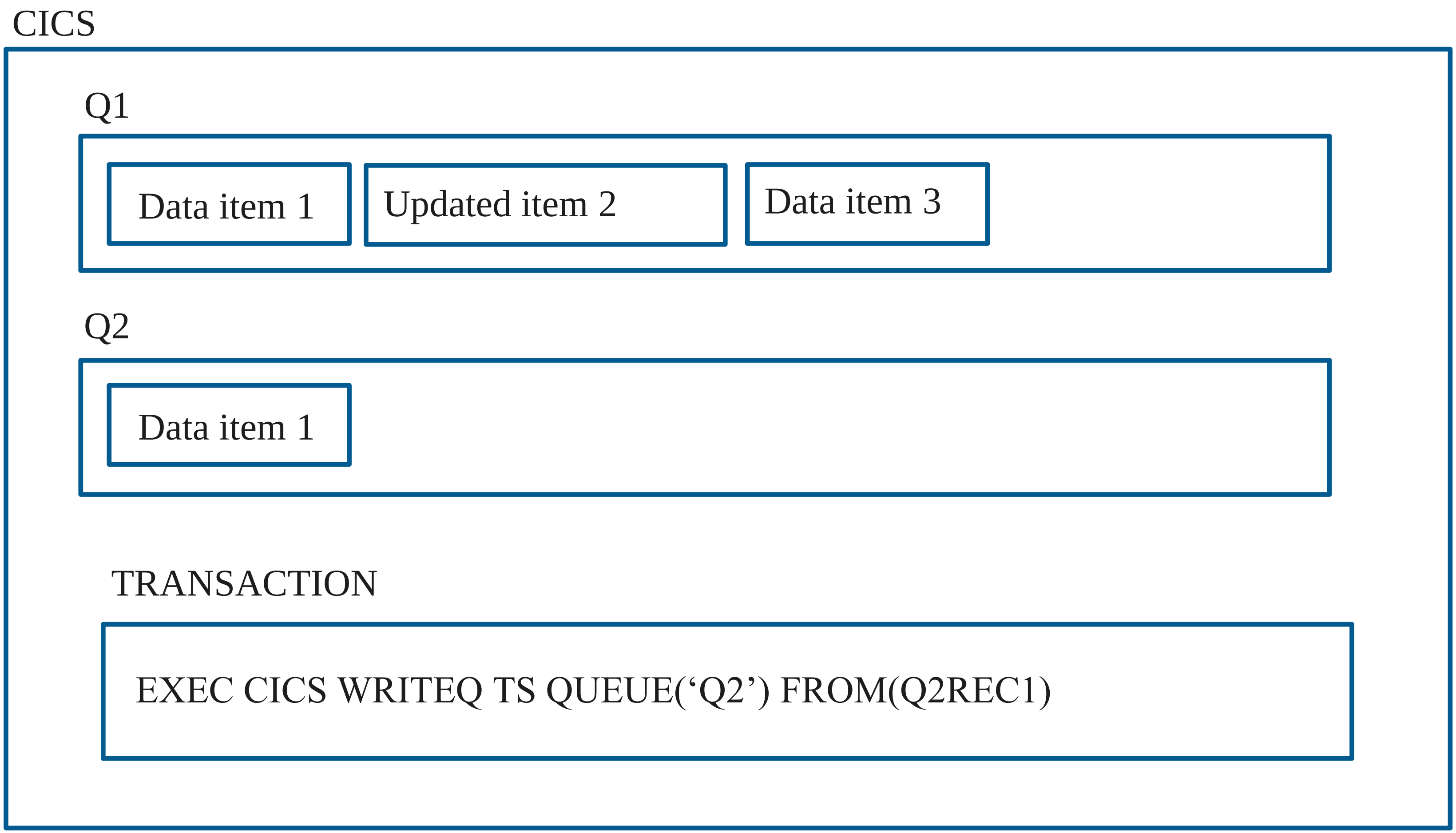
Updated item 2

Data item 3

TRANSACTION

```
EXEC CICS WRITEQ TS QUEUE(‘Q1’) FROM(X) ITEM(2) REWRITE
```

A Temporary Storage queue



CICS

Q1

Data item 1

Updated item 2

Data item 3

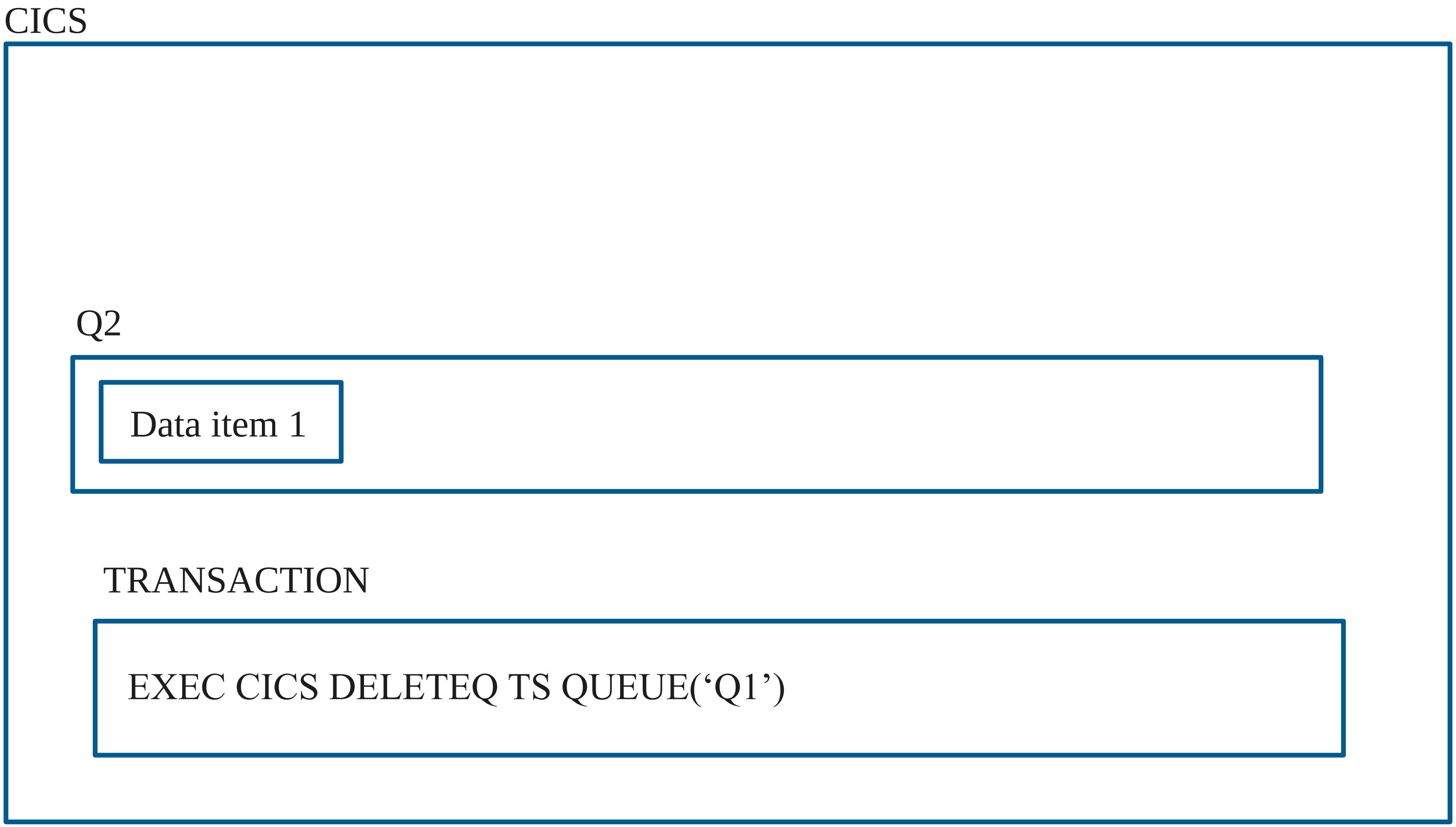
Q2

Data item 1

TRANSACTION

```
EXEC CICS WRITEQ TS QUEUE(‘Q2’) FROM(Q2REC1)
```

A Temporary Storage queue



CICS

Q2

Data item 1

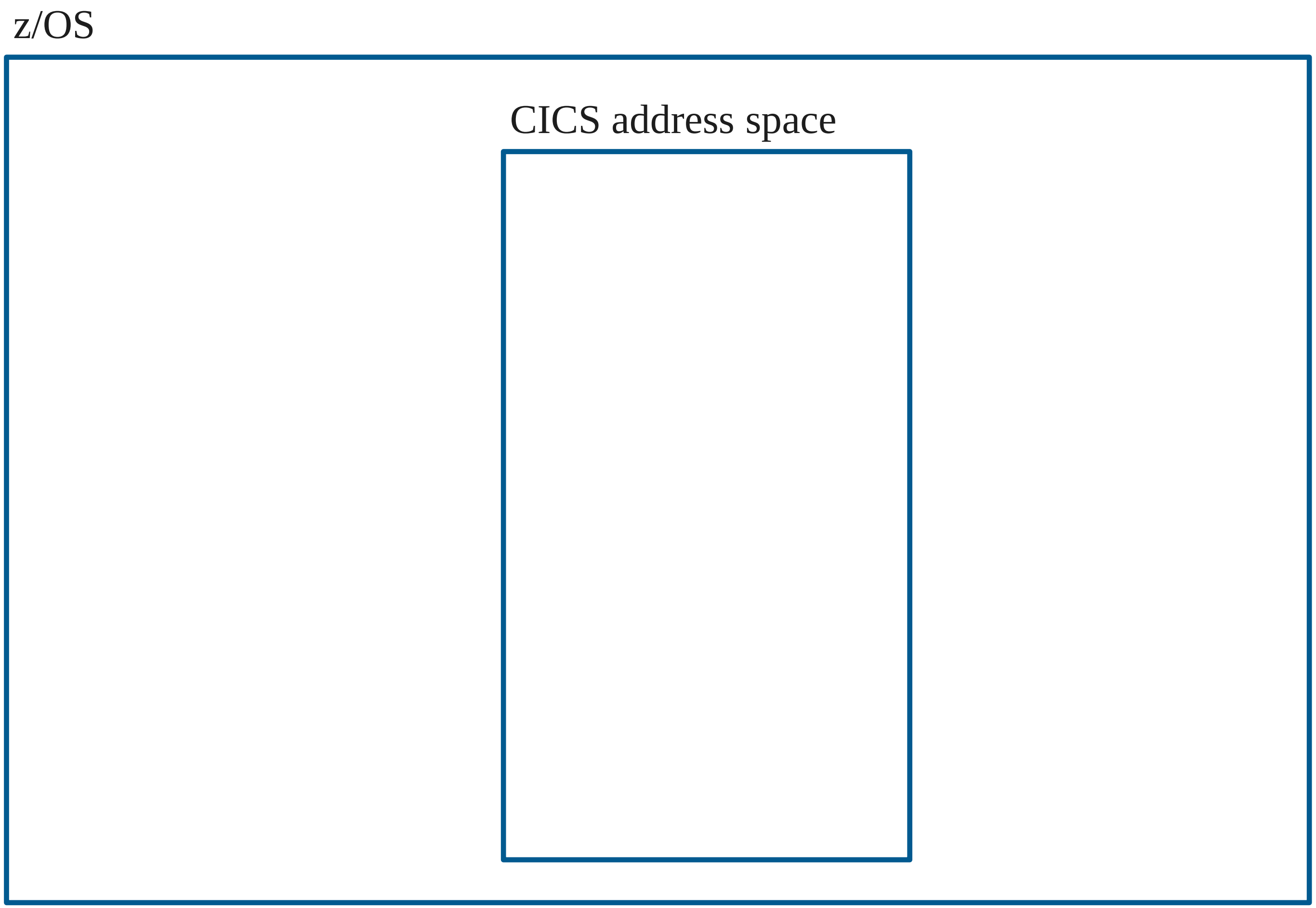
TRANSACTION

EXEC CICS DELETEQ TS QUEUE(‘Q1’)

A Temporary Storage queue

- A given queue exists in main, auxiliary or shared storage
- Each queue has a unique 16-character queue name
- Allocated on the EXEC CICS API command that creates it, or by CICS internally when creating a queue
- Each queue can have up to 32767d items (records) on it
- Each record can be up to 32763d bytes long
- For applications:
  - Queues are created or appended to via EXEC CICS WRITEQ TS
  - Queues are updated via EXEC CICS WRITEQ TS REWRITE
  - Queues are read via EXEC CICS READQ TS
  - Queues are deleted via EXEC CICS DELETEQ TS
  - Individual records are referenced via the ITEM(n) adverb
  - You cannot delete individual items, only an entire queue

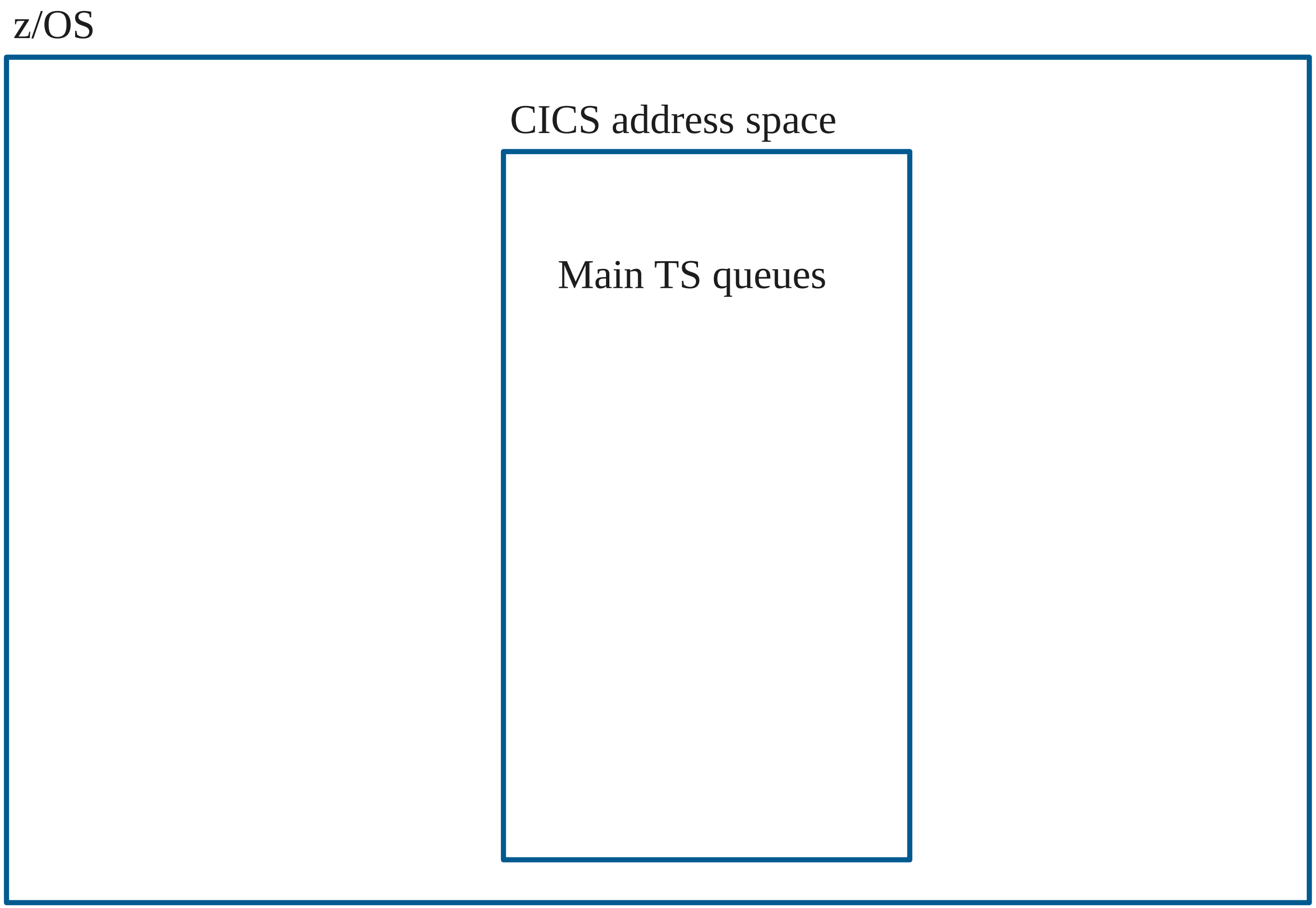
Temporary Storage locations



z/OS

CICS address space

Temporary Storage locations

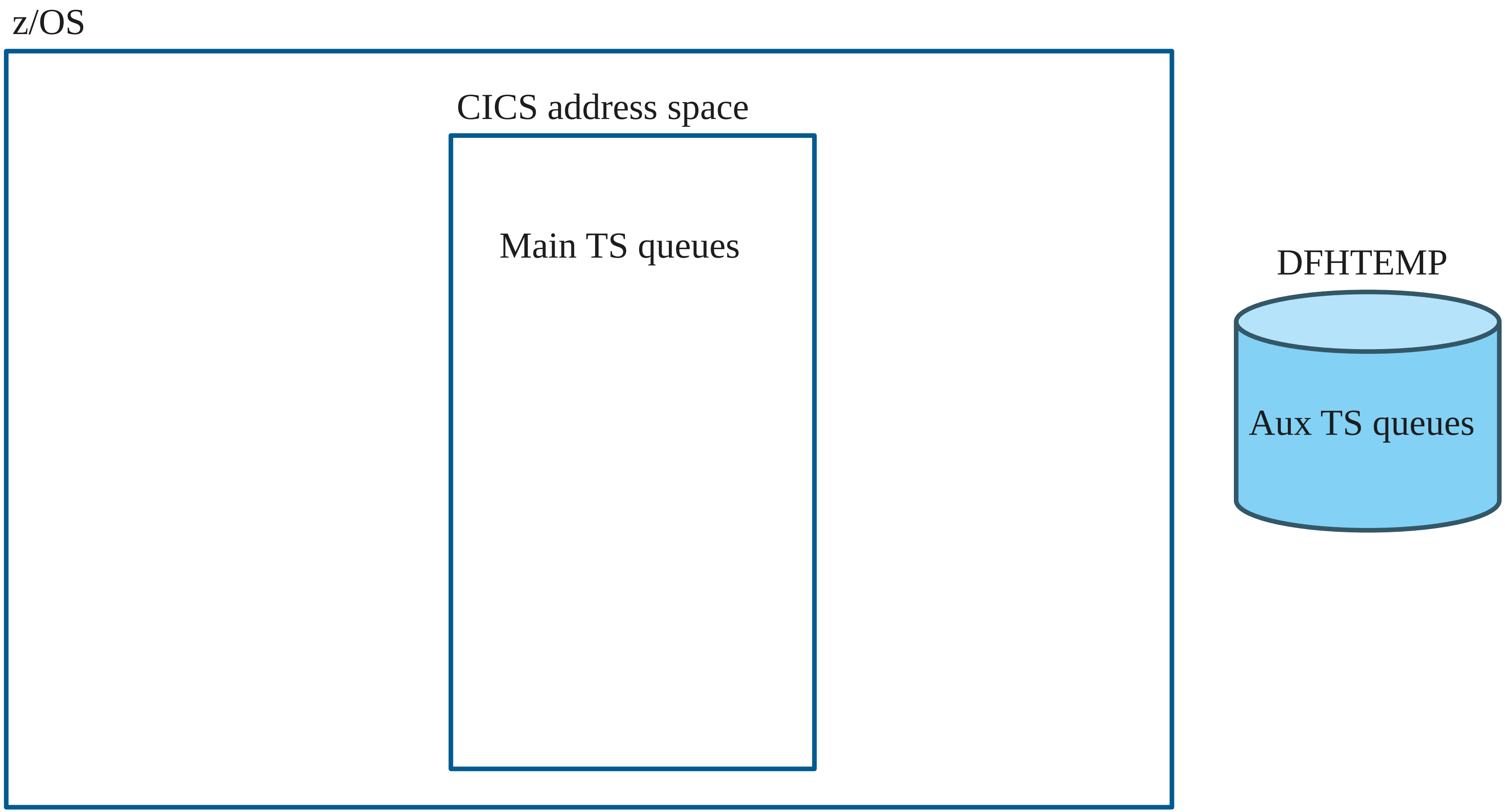


z/OS

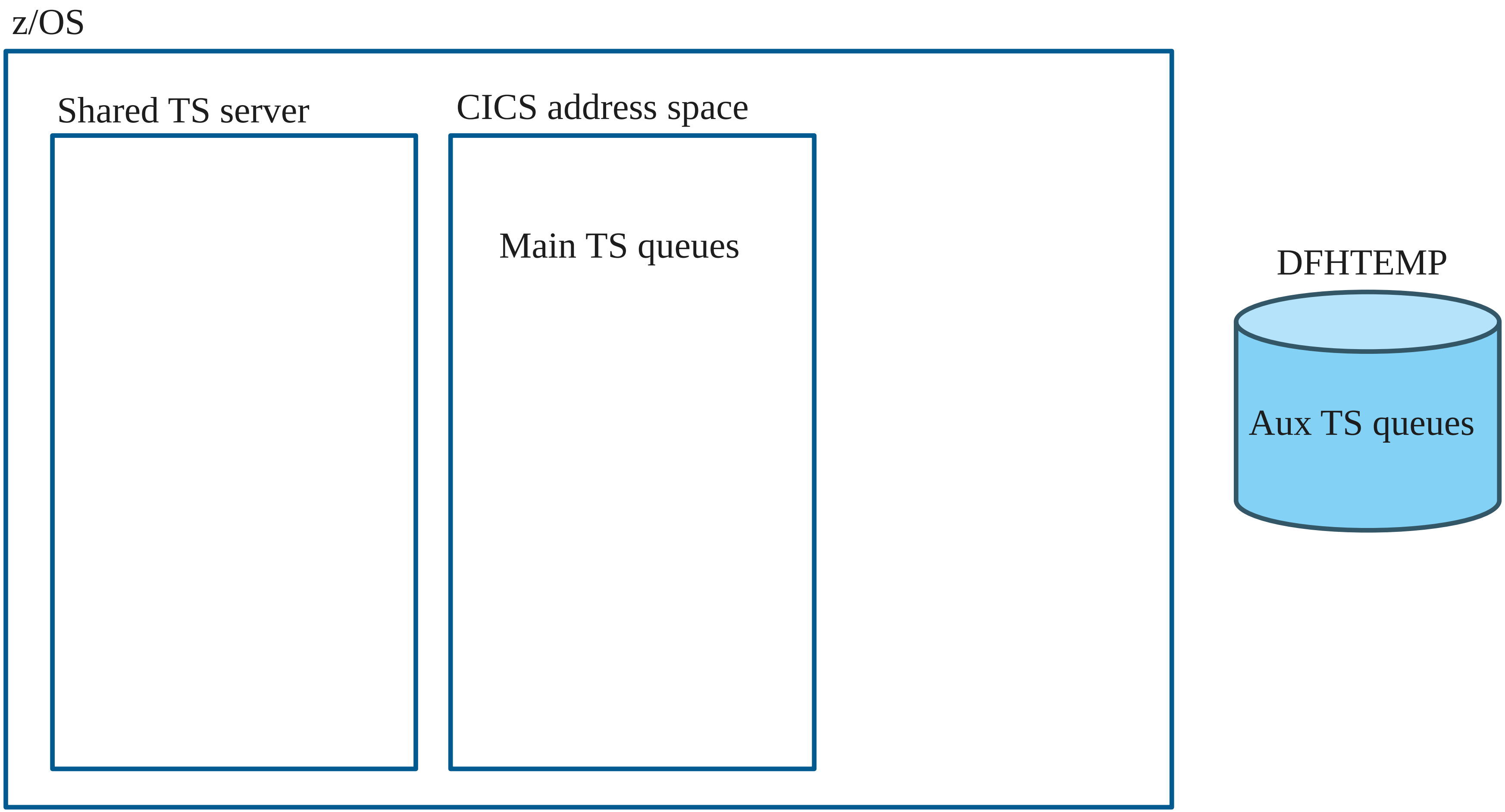
CICS address space

Main TS queues

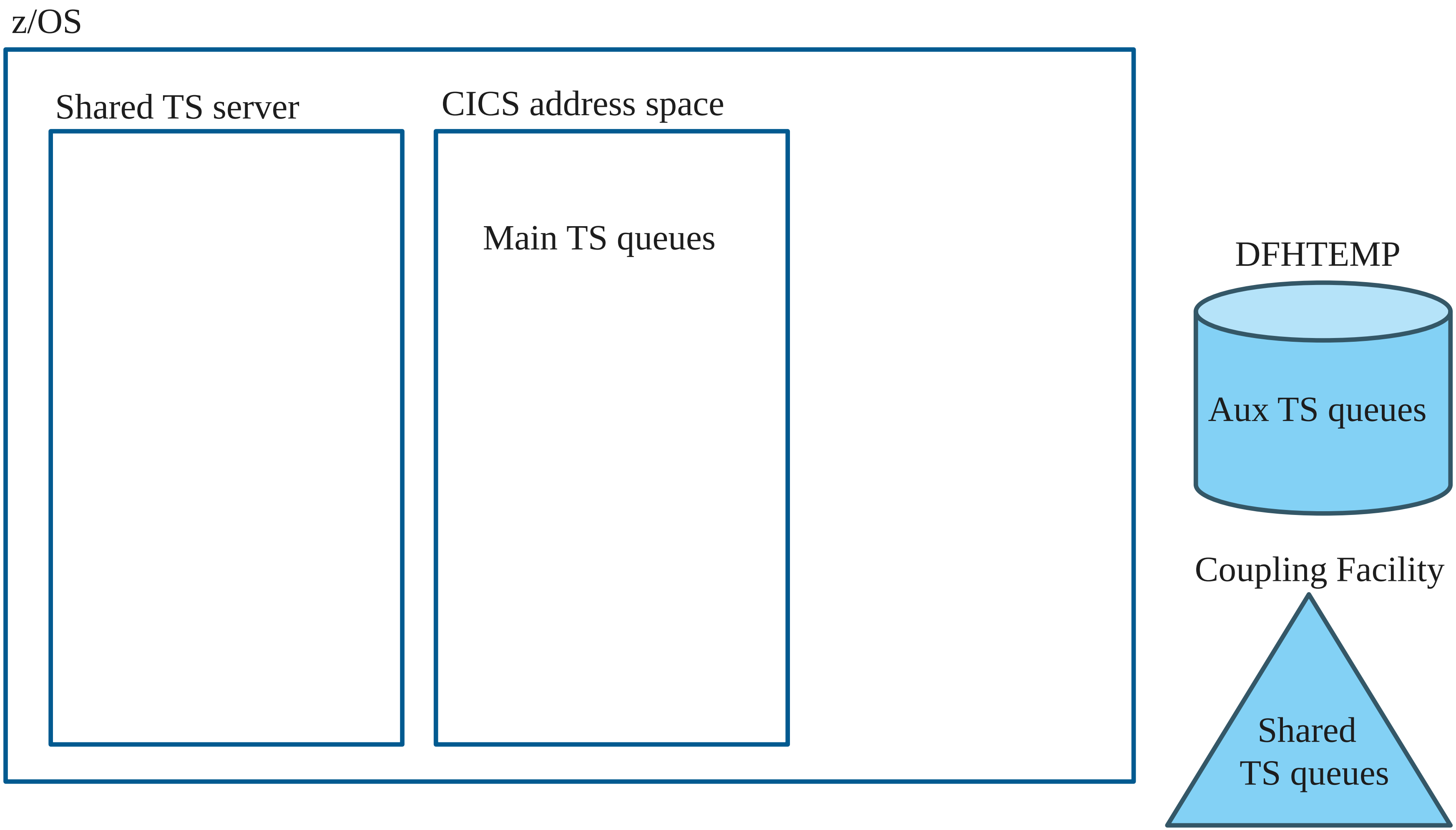
Temporary Storage locations



Temporary Storage locations



Temporary Storage locations



Main Temporary Storage

- Data is held within virtual storage within the CICS region
  - Held above the bar (in 64-bit storage) within the address space
- Main temporary storage data is not recoverable, so changes made to it by applications cannot be automatically backed out
  - Does not get undone following an abend or an EXEC CICS SYNCPOINT ROLLBACK
- It does not survive across any form of CICS restart
  - Data is persistent but not recoverable within a given run of CICS
- Requires a sensible curation and management policy
  - To ensure old queue data is deleted when no longer required

Auxiliary Temporary Storage

- Queues can also exist outside of CICS, within DFHTEMP
  - A CICS-maintained VSAM entry sequence data set (ESDS)
- Data is passed at the CI level
  - CICS will read in and write out an entire control interval as required
- Multiple queues can exist across multiple CIs
- New CIs are allocated as required up to the limit
- Can be defined as recoverable if required
  - Non-recoverable auxiliary queues survive a warm restart
  - Recoverable auxiliary queue work is backed out at DTB or E/R
- Again, requires sensible management to tidy up old data

Shared Temporary Storage

- Data can be shared across z/OS images within a parallel sysplex environment
- Shared temporary storage queues are held within pools
  - Each pool corresponds to a list structure within the coupling facility
- Pool access is via a separate server address space
- Shared temporary storage is not recoverable
  - Changes are not logged and do not get backed out
- They are persistent
  - They are preserved across CICS restarts and z/OS IPLs
- Again, requires sensible management to tidy up old data

Definitions - TSMODELs

- Temporary storage queues are self-defining
  - No pre-existing definitions are required to create one
- The first EXEC CICS WRITEQ command for a given queue will cause CICS to create it
  - By default, as a non-recoverable auxiliary queue, local to the CICS system
- The RDO TSMODEL resource may be used if there is the need to predefine certain queue characteristics
- Each TSMODEL affects a subset of queues whose names begin with a specific character prefix
- TSMODELs allow queuenames of a given prefix to be:
  - Recoverable
  - Remote
  - Main, auxiliary or shared

Local and Remote TS Queue Support

- Temporary storage was originally local to a given CICS
- With CICS function shipping, commands could be routed
  - From application to queue owning regions (AORs and QORs)
  - Applications could specify a static value for the remote system with the SYSID adverb on a command
  - Queues could be statically defined as remote to another CICS, with REMOTESYSTEM specified via a TSMODEL
- A modern approach is selecting the destination at run-time
- Simple scratchpad data is likely to stay local to its CICS
- Customers first used QORs where data needed sharing
  - Shared temporary storage is now heavily used for data sharing

TS Recovery Considerations

- Recap:
  - Main temporary storage is always non-recoverable
  - Auxiliary temporary storage can be recoverable or non-recoverable
  - Shared temporary storage is always non-recoverable
- Temporary storage is often used as a scratchpad to share data between transactions, so most queues are not defined as recoverable
  - Well-designed applications ensure that existing queues of interest are deleted before new data is written to them
- For main temporary storage queues, any form of CICS restart will automatically delete them
- For auxiliary queues, initial and cold restarts will delete them. They are rebuilt on a warm or emergency restart

64 bit Enhancements for TS

- Most of TS domain was moved above the bar in CICS TS 4.2
  - Provided virtual storage constraint relief
- TS queue data was moved above the bar
  - API for using temporary storage was not changed, so no application changes were required
- New SIT parameter, TSMAINLIMIT, introduced to limit how much storage can be used by TS queues
  - Can set limits from 1 MB to 32 GB. Default is 64 MB.
    - But cannot be set to more than 25% of MEMLIMIT

Changing TSMAINLIMIT

- If you increase the TSMAINLIMIT setting, the value is set as follows:
  - If the new value is not greater than 25% of the value of the z/OS parameter MEMLIMIT, the value that you choose is set.
  - If the new value is greater than 25% of the MEMLIMIT value, TSMAINLIMIT remains unchanged.

- If you decrease the TSMAINLIMIT setting, CICS attempts to maintain at least 25% free space
  - If there is currently less than 25% free space, TSMAINLIMIT remains unchanged.
  - If at least 25% of the new limit will be free space, the setting is decreased to the value that you choose.
  - If less than 25% of the new limit would be free space, the setting is decreased to the current utilization plus 33% of that utilization.

64 bit storage warnings

- When TS queues are using 75% of TSMAINLIMIT, message DFHTS1601 is issued.

```
DFHTS1601  APPLID MAIN TEMPORARY STORAGE USAGE HAS REACHED
XX% OF TSMAINLIMIT STORAGE.
```

  - First issued at 75% usage and then at each subsequent 5%.

- When limit is reached, attempts to write TS queues will fail with message DFHTS1602.

```
DFHTS1602  APPLID MAIN TEMPORARY STORAGE HAS ATTEMPTED
TO EXCEED THE TSMAINLIMIT STORAGE LIMIT.
```

TSMAINLIMIT

- Setting TSMAINLIMIT too high in the CICS startup JCL results in
  - `DFHTS1608 APPLID TS domain initialization has failed because an attempt was made to set TSMAINLIMIT to a value greater than 25% of MEMLIMIT.`
  - CICS initialisation is terminated

Messages for Shared and AUX TS

- Messages have been added to warn when Shared TS or Auxiliary TS is approaching storage limits
  - Similar to the TS Main messages
    - Initial message when threshold crossed
    - Subsequent messages as limit approached
    - Messages when capacity is available again

Messages summary

TS queue type	MAIN	Shared pool	AUX
Messages for threshold being crossed	DFHTS1601 applid Main temporary storage usage has reached xx% of TSMAINLIMIT storage. DFHTS1604 applid Main temporary storage usage has fallen below 70% of TSMAINLIMIT.	DFHXQ0422I CF structure strname has risen to percentage% of entries in use. DFHXQ0423I CF structure strname has risen to percentage% of elements in use. DFHXQ0420I CF structure strname has fallen to percentage% of entries in use. DFHXQ0421I CF structure strname has fallen to percentage% of elements in use.	DFHTS1316 applid Auxiliary temporary storage data set usage has reached xx% of the capacity. DFHTS1317 applid Auxiliary temporary storage data set usage has fallen below 70% of the capacity.
When the first threshold messages are issued	When 75% or more of the maximum storage is in use.	Default is from 80%, the warning threshold is increased to the next higher level, or decreased to the previous lower level.	When 75% or more of the maximum storage is in use.
Increment step	5% increment	5% if less than 95%; otherwise 1%	5% increment
Message issued when the storage is full	DFHTS1602 applid Main temporary storage has attempted to exceed the TSMAINLIMIT storage limit. (It is not issued more often than once every 5 minutes)	DFHXQ0442 CF structure strname request failed, structure is full. DFHXQ0443 CF structure strname request failed, all lists are in use. (It is not issued again for further failures until the used numbers of elements and entries fall well below the warning threshold)	DFHTS1311 applid Temporary storage data set is full and cannot be extended. (It is not issued more often than once every 5 minutes)

* Blue messages are new in CICS TS 6.1 and in 5.6 with PH28145

TS Automated Cleanup

- Putting TS queues in 64 bit storage has the potential for large amounts of data to be stranded there
  - Applications could write lots of TS queues and not delete them
- Mechanism needed for automatic deletion of unused TS queues
  - TS queue automatic cleanup was introduced in CICS TS 4.2
- TS queue deletion is controlled by the EXPIRYINTMIN parameter on TSMODELs
  - Specifies the number of minutes after which a TS queue which has not been used becomes **eligible** for deletion
    - Actual deletion will occur on next cleanup scan

Time of queue deletion

09:59 Cleanup scan runs.

10:00 Queue with expiry interval 30 minutes is read. If this queue uses the only TSMODEL with an EXPIRYINTMIN specified, the cleanup scan interval will be 3 minutes.

10:02 Cleanup scan runs. Queue not deleted.

⋮

10:29 Cleanup scan runs. Queue not deleted.

10:30 Queue eligible for deletion

10:32 Next cleanup scan, queue is deleted

TS cleanup scan

- Scan interval is 10% of smallest specified expiry interval or 1 minute, whichever is the larger value
  - Prior to CICS TS 6.1 the same interval was used for both local and shared TS queues

- From CICS TS 6.1, separate intervals are used for local and shared TS queues
  - For the shared TS scan, the smallest interval used is 5 minutes

- If all installed TSMODELs specify EXPIRYINTMIN as zero or there are no TSMODELs installed, no cleanup scan is performed

TS Cleanup messages

- If queues are not being deleted but the cleanup scans are running, message DFHTS1605 is issued every 30 minutes. This will report that 0 queues have been deleted.

```
DFHTS1605  DATE TIME APPLID SCAN OF TEMPORARY STORAGE QUEUES
           COMPLETED. XXXX TEMPORARY STORAGE QUEUES WERE
           SCANNED AND YYYY WERE DELETED.
```

- If queues are deleted, the messages are issued on the next cleanup scan and will report how many queues have been deleted.

Shared TS cleanup scan

- With many CICS regions using the same coupling facility, potential exists for many regions to scan shared TS queues
  - This is not required. If CICS 1 has just scanned the shared queues, CICS 2 does not need to do so a few seconds later etc.

- CICS writes a shared queue
  - DFSHARED concatenated with the pool name
  - Contains STCK of when shared TS queues were last scanned

- The cleanup task checks the interval between last shared TS scan and current time
  - Do not scan again if interval is less than 2 minutes.

Temporary storage diagnostics

- Temporary storage domain provides a full set of diagnostics
  - System dump verbexit
  - Standard trace component settings
- For auxiliary temporary storage, a consistency checking mechanism is available
  - This is activated by the TS=3 (or TS=ALL) trace selection
- It drives additional checks on entry and exit to the class
  - This is much like the traditional “DFHTS1310” DFHTRAP
- If there is corruption or an overlay to control blocks or buffers, the consistency checker will detect this now
  - As opposed to getting a potential DFHTS1310 abend later

Historical Problems

- Insufficient space in DFHTEMP for auxiliary queues
  - Larger data sets now typically avoid this
- Insufficient space in address space for main queues
  - Could lead to cloning of QORs to accommodate sufficient space
  - 64 bit storage use avoids this issue
- General build-up of unwanted queues and data
  - Customers now using automated clean up mechanism
- Performance issues when accessing queues
  - Older table mechanism now replaced with a tree structure
- DFHTS1310 abends
  - “Temporary storage data set does not match bit map”
  - Very common in the 1980s and beyond
  - Typically due to overlays of state data and control blocks
  - Resolved with TS domain restructure and storage protection

Temporary Storage Performance Considerations

- Auxiliary temporary storage can out-perform main
  - If sufficient buffers and strings are defined via TS=(x,x,x)
- This can keep CIs in memory and avoid the need to perform I/O
  - It also avoids the need to getmain storage for the data
- Choose appropriate CI sizes and number of CIs for DFHTEMP, to reduce the need for I/O
- Actively delete unwanted temporary storage queues
  - Explicitly, or via the cleanup mechanism
- TS=ALL or TS=3 tracing drives the consistency checker
  - If there are no problems with auxiliary Temporary Storage, this is an unnecessary overhead
  - IBM's recommendation is all standard tracing be active and set to level 1 by default

Questions
Answers