IBM MQ for z/OS SMF Skills Workshop

Dorothy Quincy
Washington Systems Center

IBM MQ for z/OS Client
Technical Specialist

IBM

Today's agenda

Introduction to Queue Manager Internals 45 min)

Walk through the internal functionality of a queue manager to understand the internal resource threads and storage facilities underpinning each queue manager.

Introduction to Interpreting SMF data for MQ on z/OS (1 hour)

Recognize key metrics and the associated impact in SMF 115 and 116 data to identify where tuning can occur to improve MQ's performance.

Demonstration of SMF processing (45 minutes)

Demonstration of the process IBM uses evaluate customer environments for IBM MQ health checks. IBM will demonstrate how the data gets from z/OS to readable and interpretable spreadsheets.

My goals for today

Make it clear how IBM processes your SMF data to make recommendations for your MQ environment during our health checks
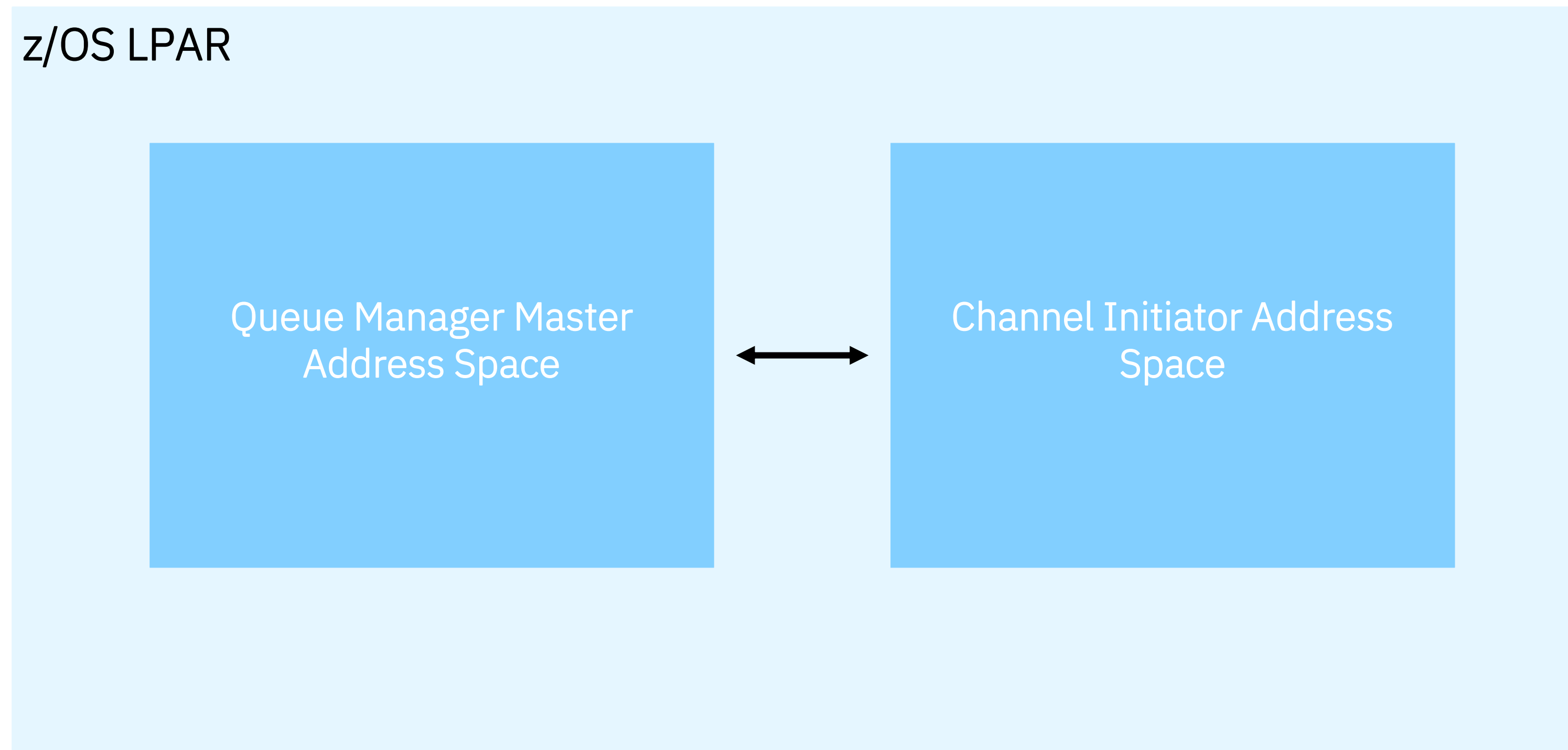
Give you a sense of the KPI's to pay attention to when evaluating SMF data for your environment
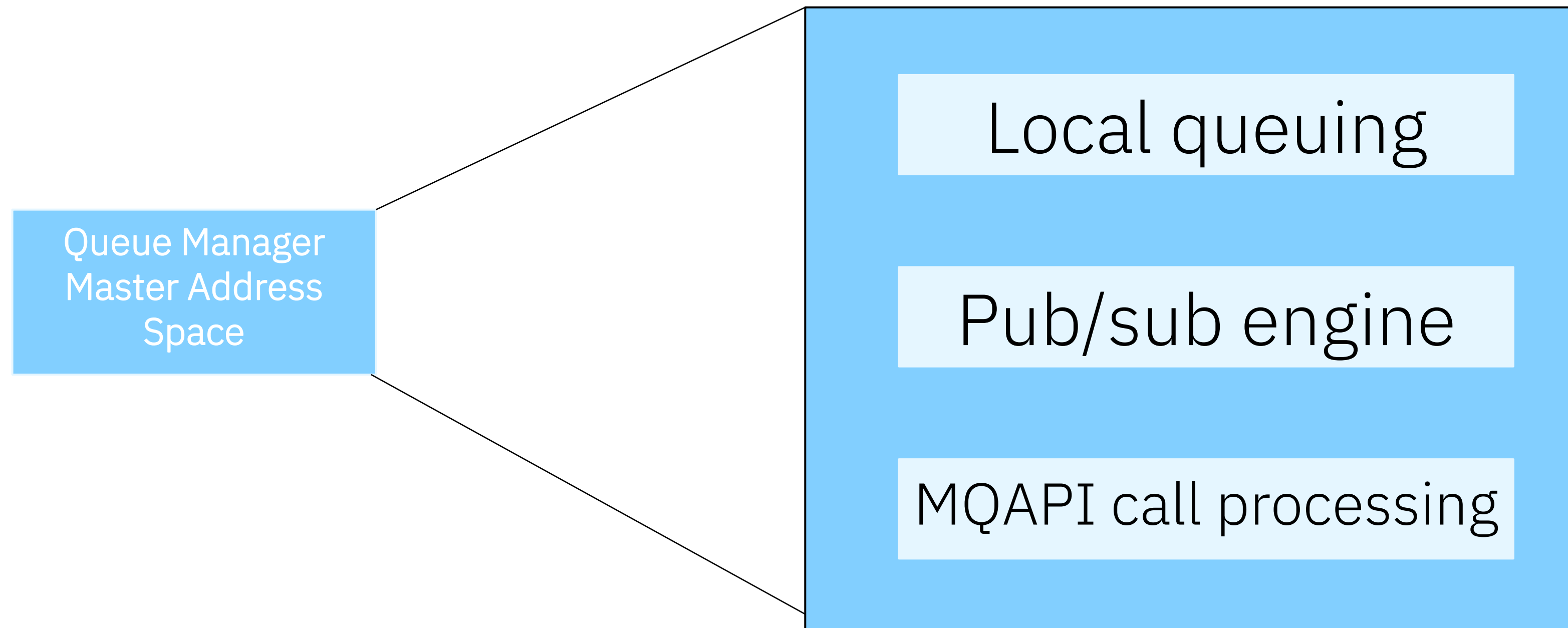
Show you various options for how to view SMF data

Connect the SMF data and its KPI's back to what is actually going on inside a queue manager running on IBM MQ for z/OS

Introduction to Queue Manager Internals

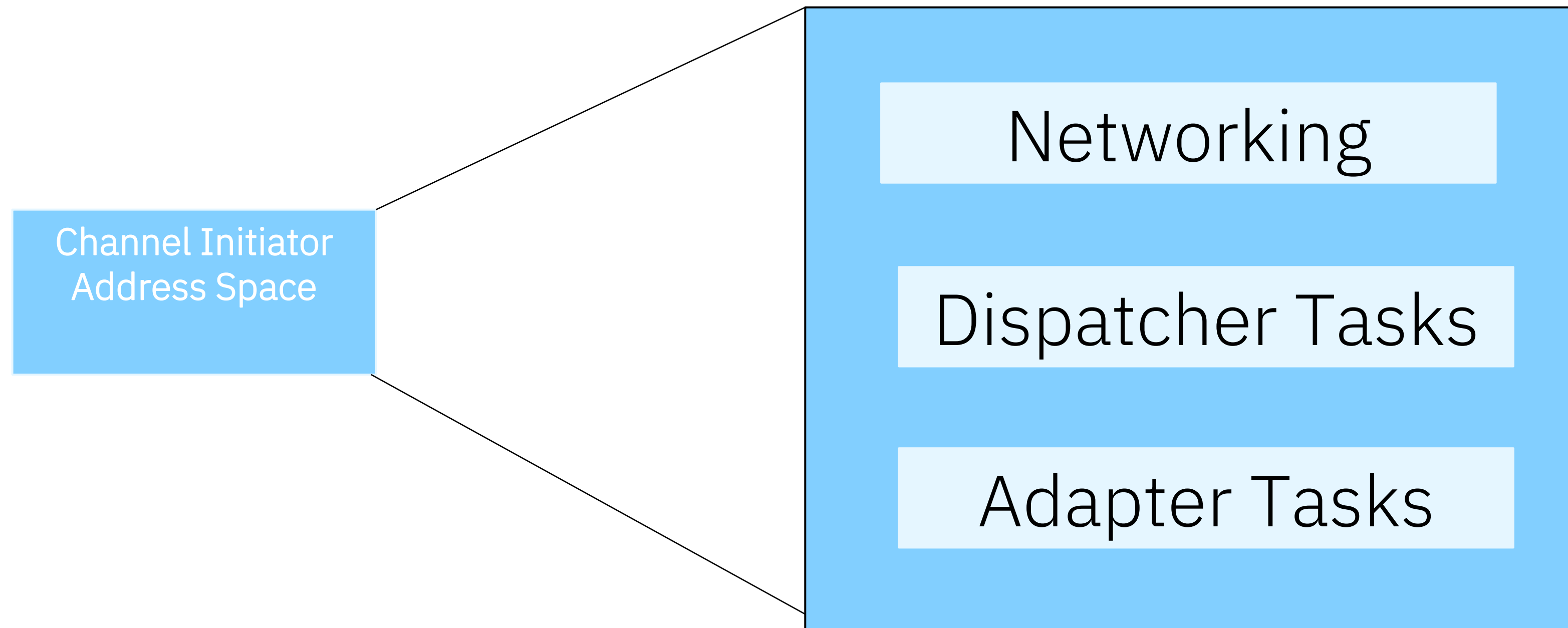# At a glance

Diving deeper:
Private queues

Queue Manager Master Address Space

Local queuing

Pub/sub engine

MQAPI call processing

Diving deeper: Private queues

Channel Initiator
Address Space

Networking

Dispatcher Tasks

Adapter Tasks

Digging into local queuing

Message A

```
Sent message:

  JMSMessage class: jms_text
  JMSType:          null
  JMSDeliveryMode:  2
  JMSDeliveryDelay: 0
  JMSDeliveryTime:  1585562399950
  JMSExpiration:    0
  JMSPriority:      4
  JMSMessageID:     ID:414d5120514d31202020202020202020200ac2815e024ce120
  JMSTimestamp:     1585562399950
  JMSCorrelationID: null
  JMSDestination:   queue:///DEV.QUEUE.1
  JMSReplyTo:       null
  JMSRedelivered:   false
    JMSXAppID: JmsPutGet (JMS)
    JMSXDeliveryCount: 0
    JMSXUserID: app
    JMS_IBM_PutApplType: 28
    JMS_IBM_PutDate: 20200330
    JMS_IBM_PutTime: 09595997
Your lucky number today is 926
```

Message details   ×

Messages properties

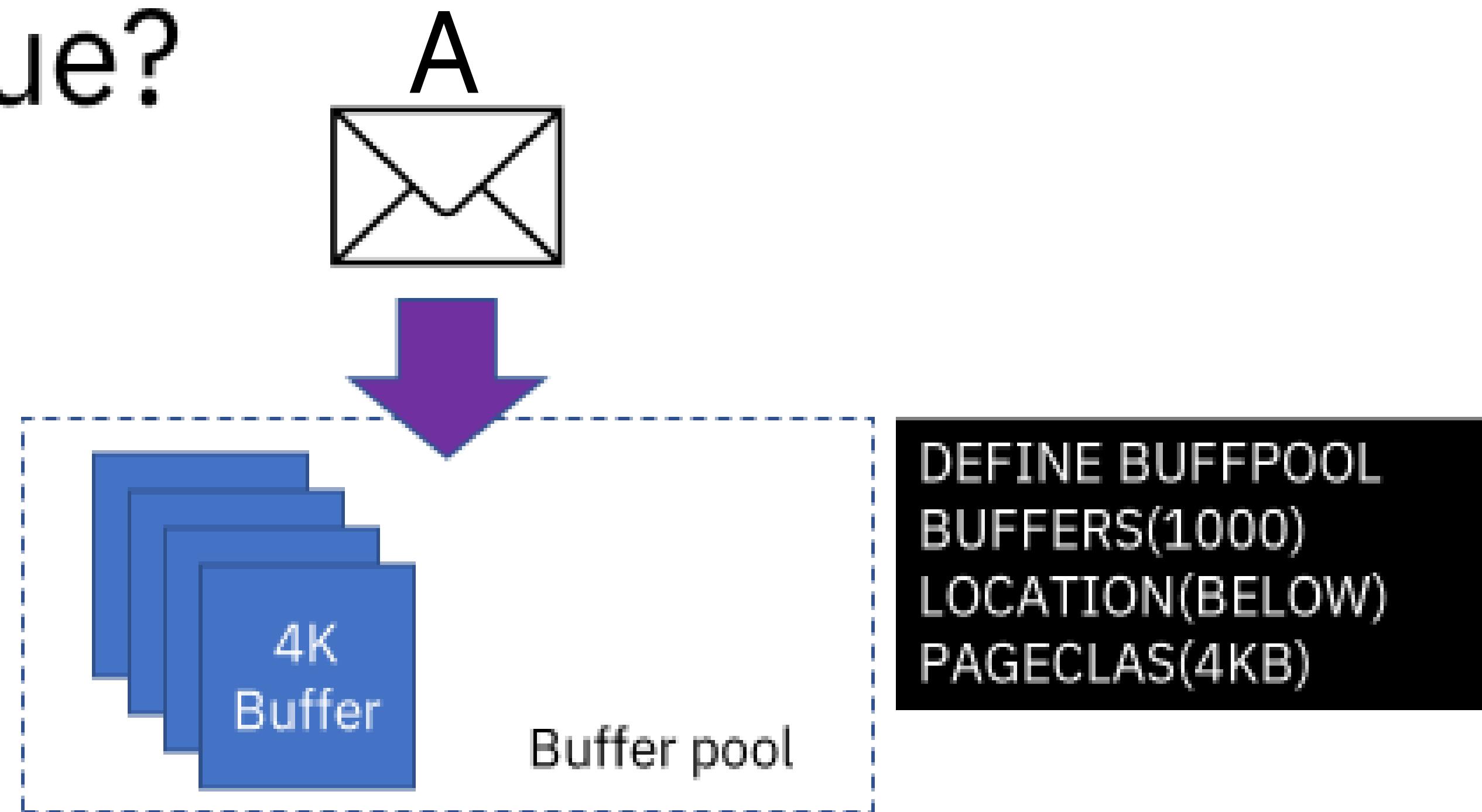|	
Message ID	ID:414d5120514d312020202020202020 0202020aec50b62010a0e40
Timestamp	2022-2-18 16:37:23
Character set	UTF-8
Delivery mode	Persistent
Application ID	JmsPutGet (JMS)
Format	MQSTR
Expiration	0
Priority	4
Encoding	273
User ID	app

Application data
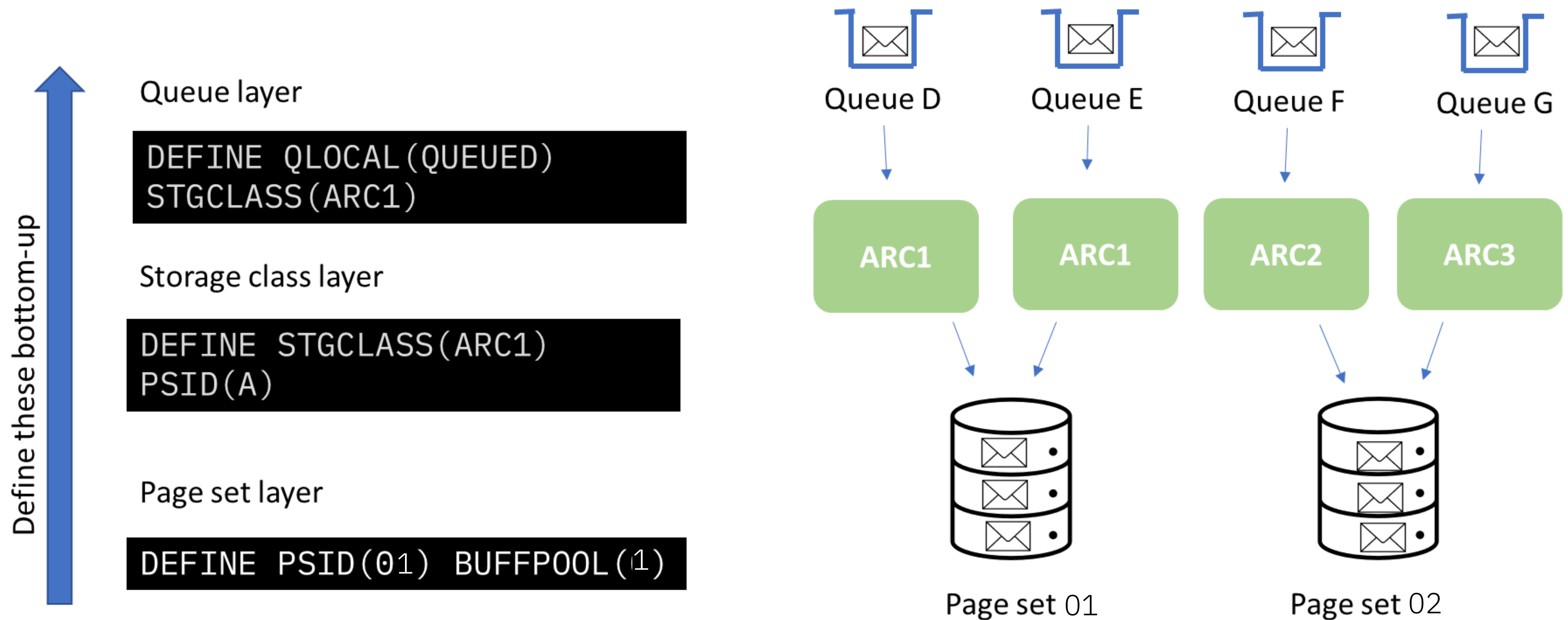
Your lucky number today is 369

9

How does physical storage work on a private queue?

A

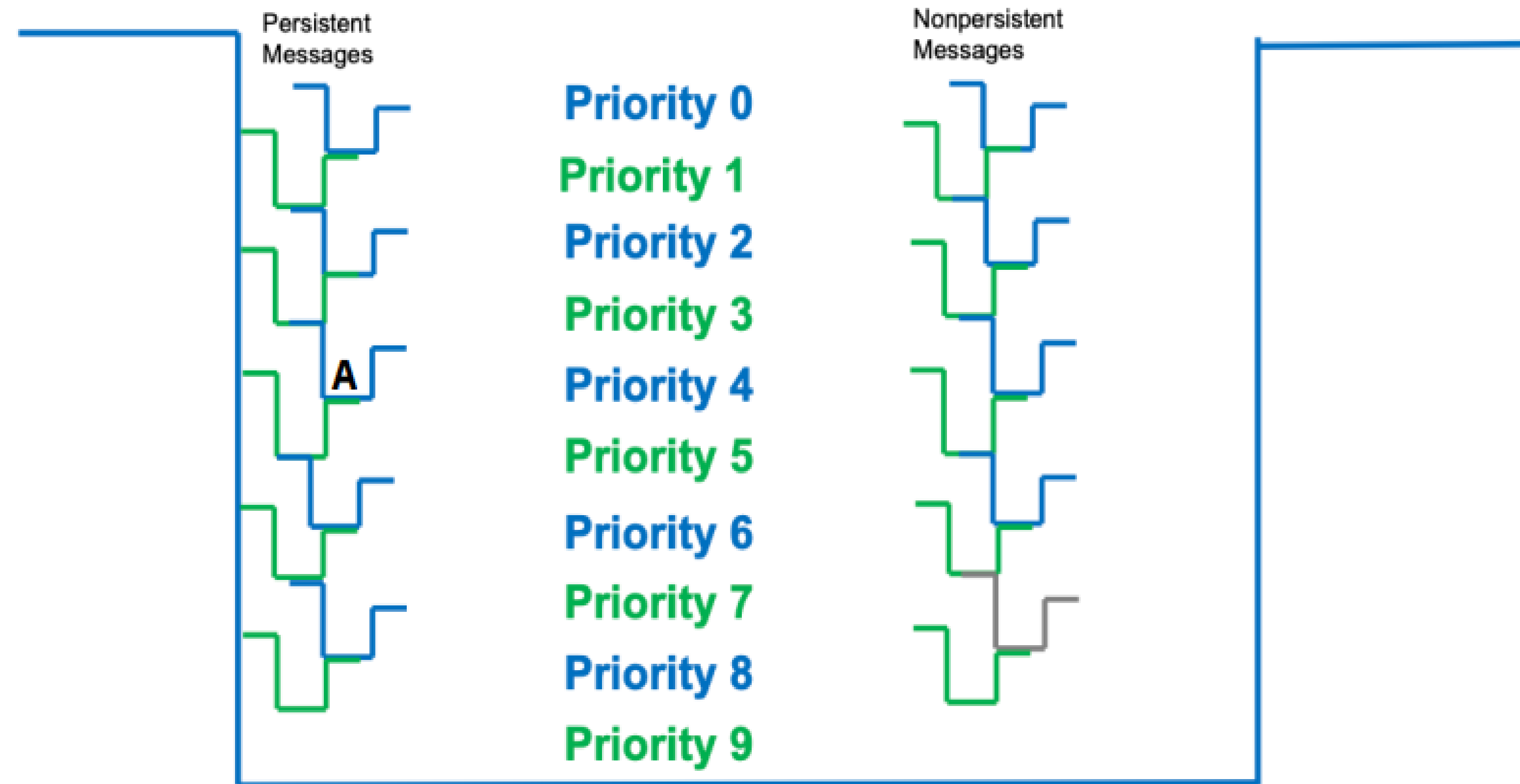Messages are written to buffer pools when any of these conditions are met:

1. When messages have been in the buffer pool for 2 log checkpoints
2. When buffer pool usage exceeds the deferred write threshold
3. When buffer pool usage exceeds the buffer pool threshold

4K
Buffer

Buffer pool

```
DEFINE BUFFPOOL
BUFFERS(1000)
LOCATION(BELOW)
PAGECLAS(4KB)
```

How are private queues associated with physical storage?

Define these bottom-up ↑

Queue layer

```
DEFINE QLOCAL(QUEUED)
STGCLASS(ARC1)
```

Storage class layer

```
DEFINE STGCLASS(ARC1)
PSID(A)
```

Page set layer

```
DEFINE PSID(01) BUFFPOOL(1)
```

Queue D   Queue E   Queue F   Queue G

ARC1   ARC1   ARC2   ARC3

Page set 01   Page set 02

Internal Representation of a
Private Queue

Persistent
Messages

Nonpersistent
Messages

Priority 0

Priority 1

Priority 2

Priority 3

Priority 4

A

Priority 5

Priority 6

Priority 7

Priority 8

Priority 9

IBM

Sub-queue Internal View

CHINIT Address Space Structure

Network

QMGR1.TO.QMGR2
QMGR3.TO.QMGR2
QMGR4.TO.QMGR2
QMGR2.TO.QMGR4

TLS interaction

Channel Initiator Address Space

Adapter Tasks

Dispatcher Tasks

SSL Tasks

Named server tasks

QM Master Address Space

QM, a channel needs to do a MQPUT
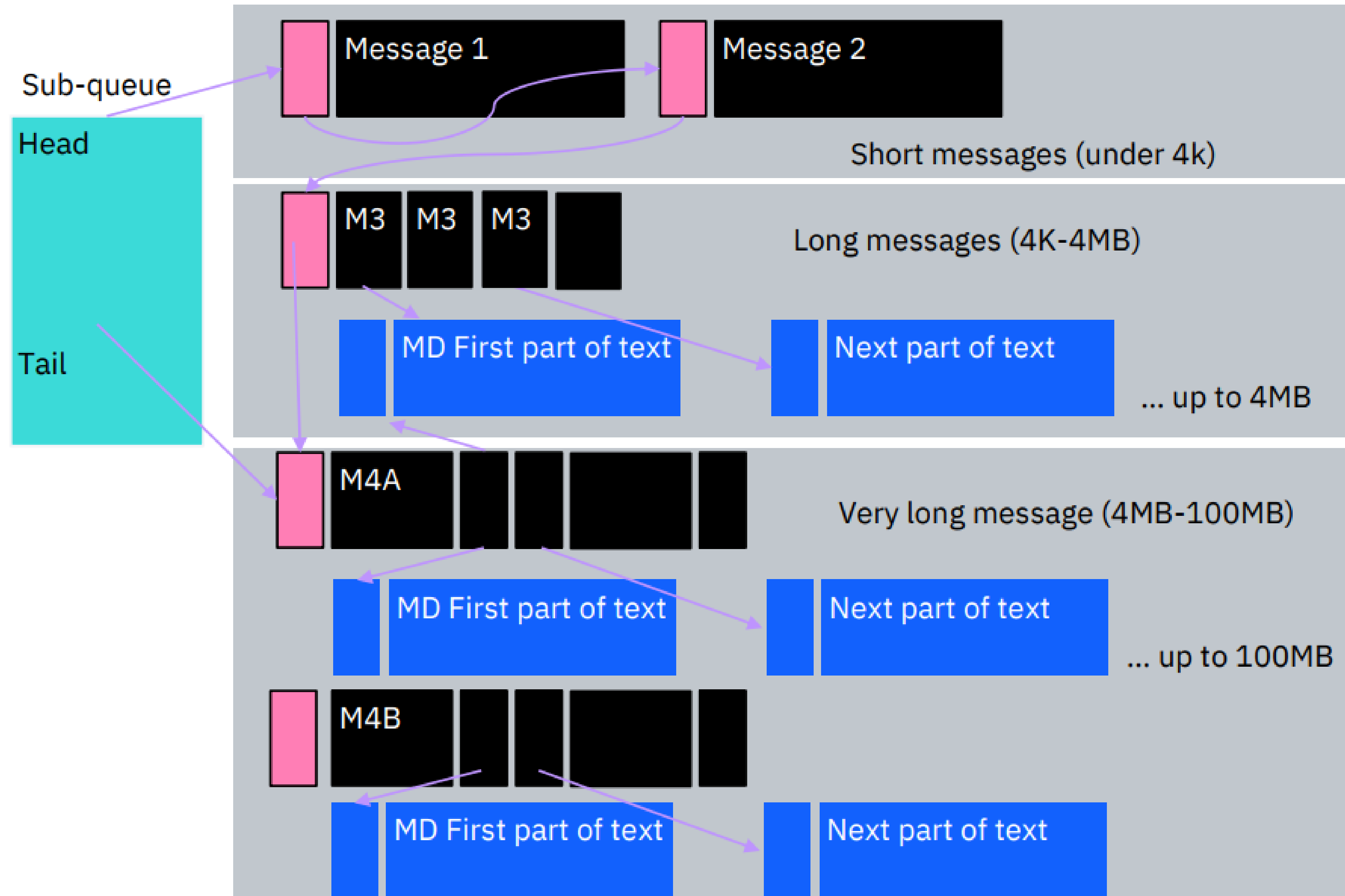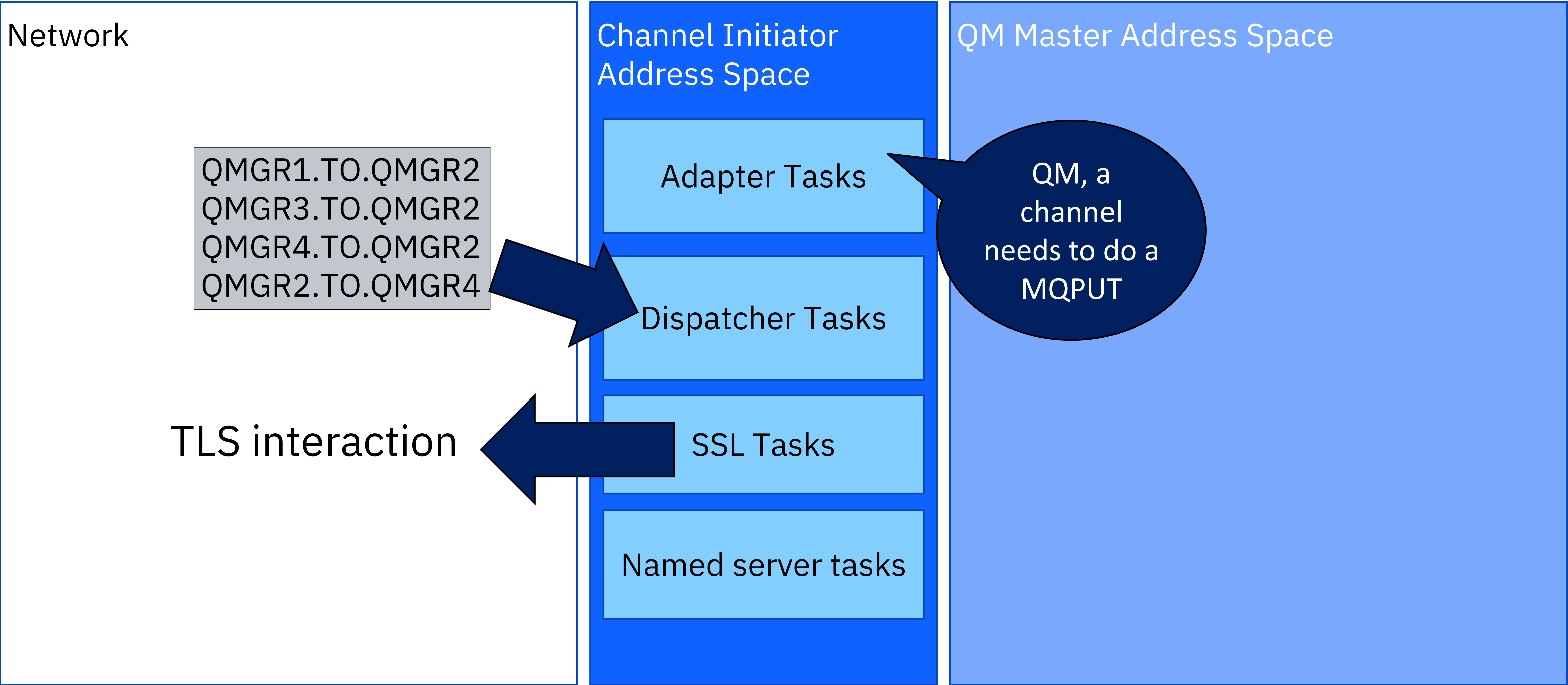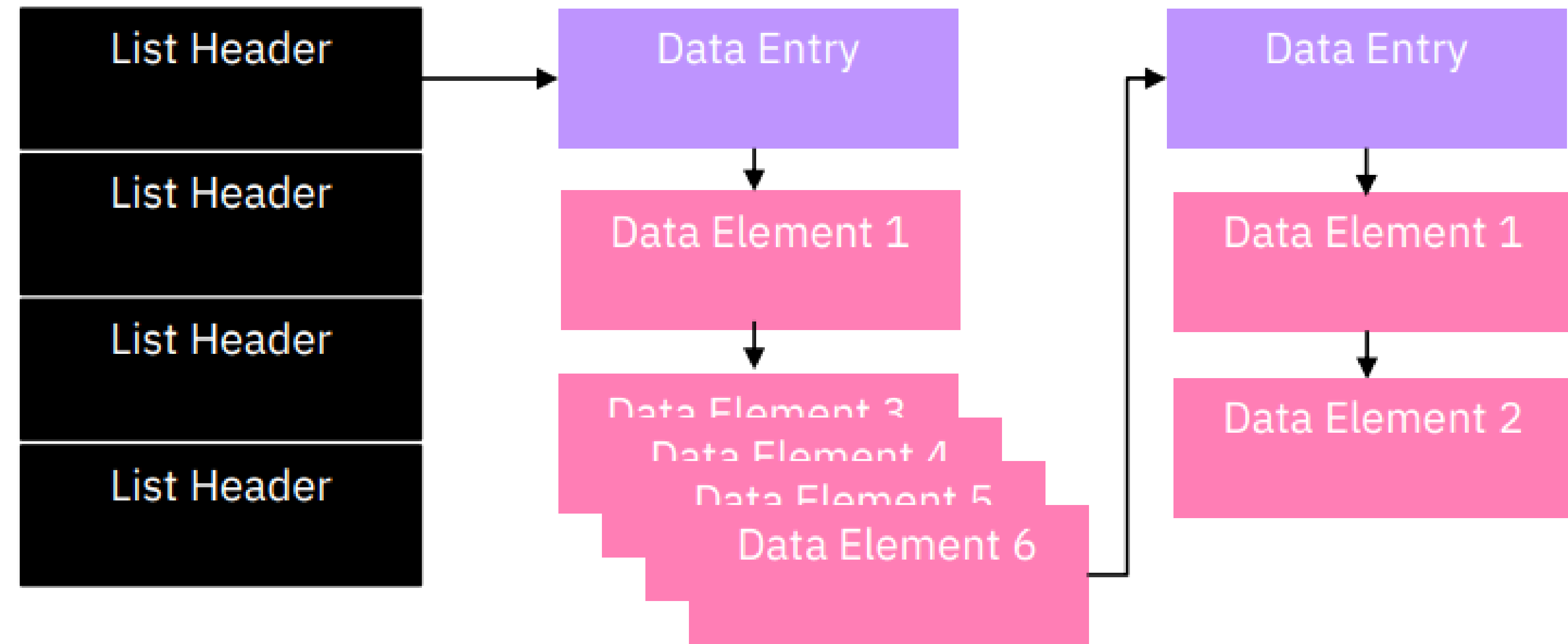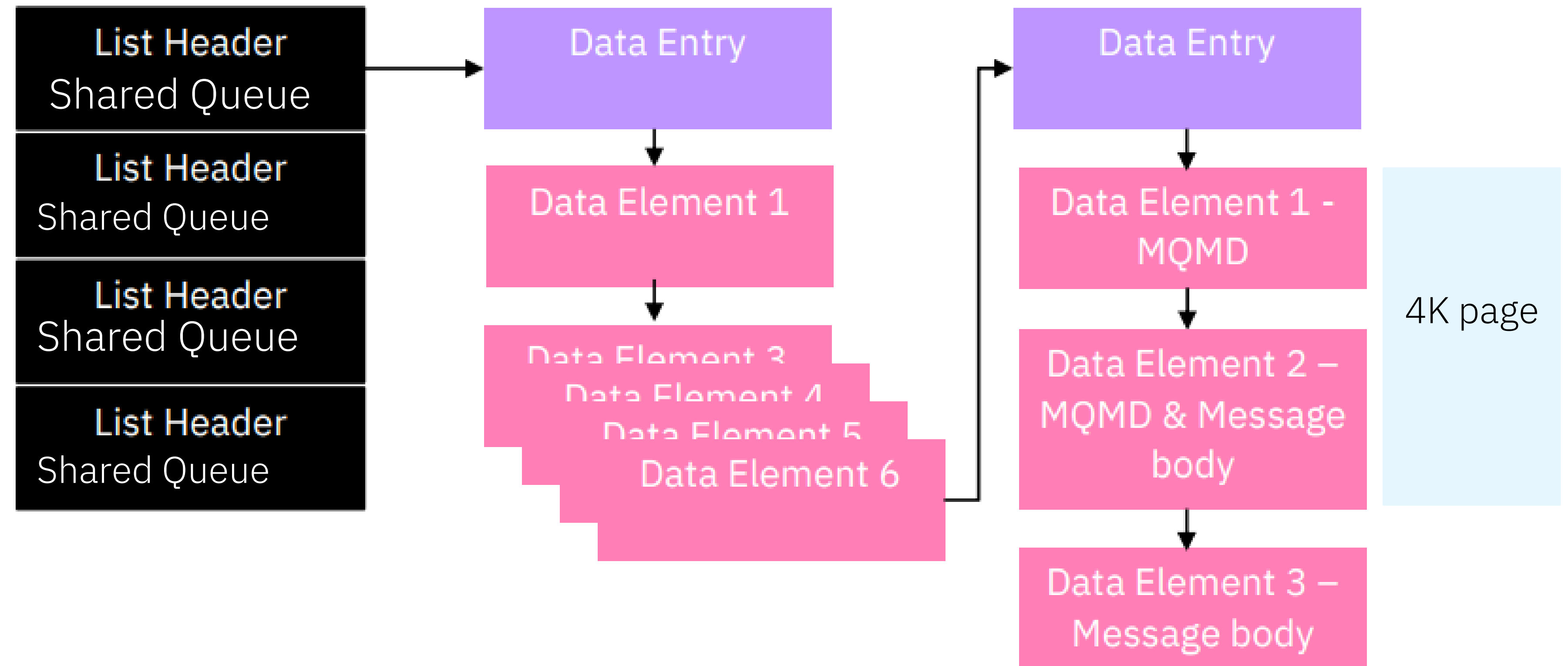
Internals of a shared queue

Internal Representation of a Coupling facility list structure

Internal Representation of a Shared Queue

List Header
Shared Queue
List Header
Shared Queue
List Header
Shared Queue
List Header
Shared Queue

Data Entry

Data Element 1

Data Element 3
Data Element 4
Data Element 5
Data Element 6

Data Entry

Data Element 1 - MQMD

Data Element 2 – MQMD & Message body

Data Element 3 – Message body

4K page

Shared Queue Message Storage
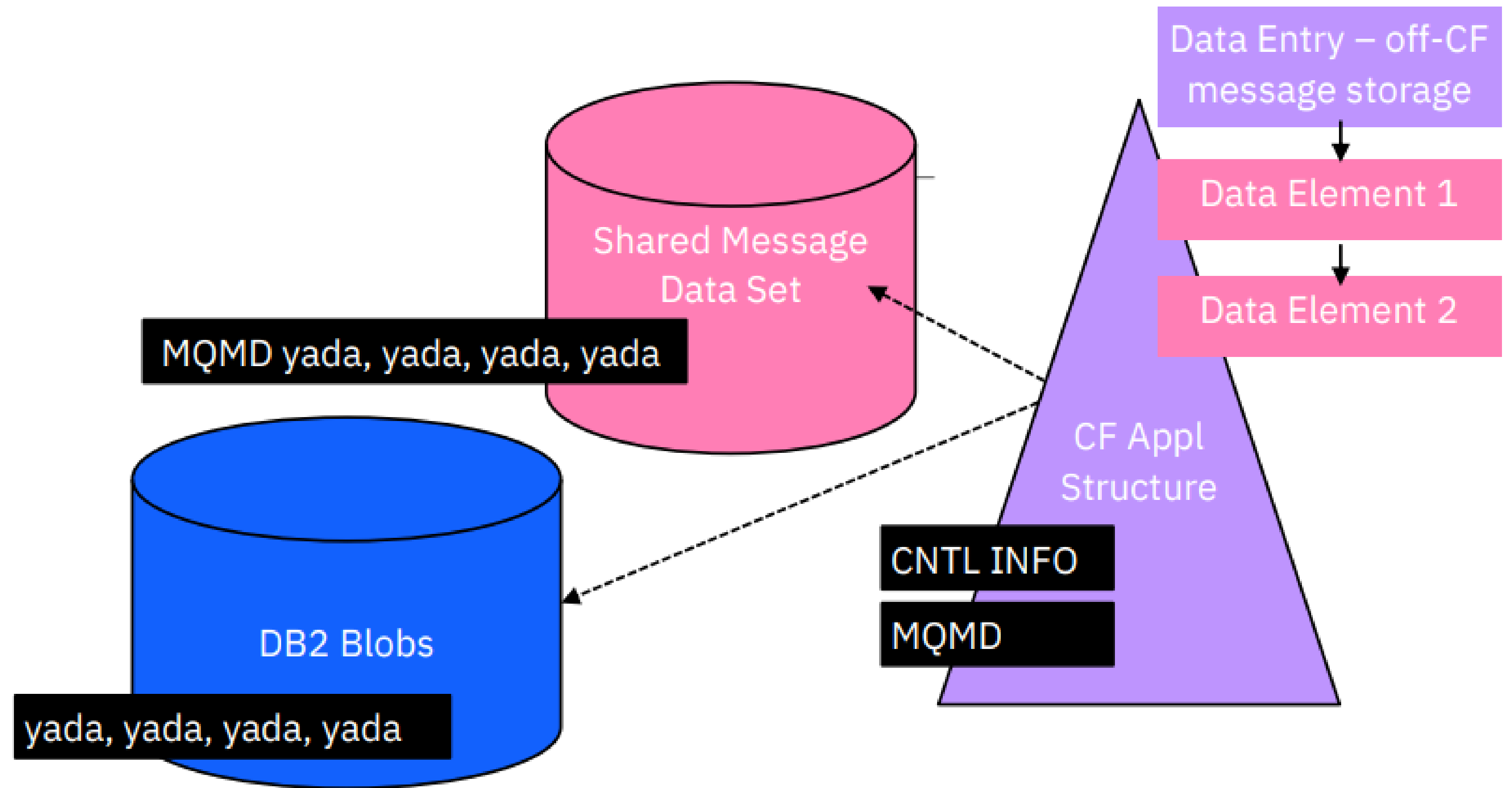
Where does logging come in?

1) Persistent
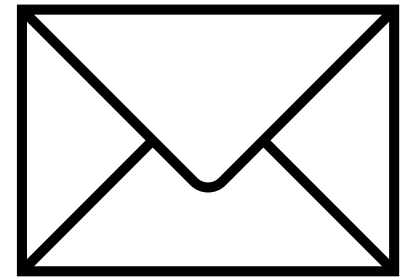
2) MQ Object

3) Queue Manager

IBM MQ

1. Unit of recovery log records
2. Checkpoint records
3. Page set control records
4. CF structure backup records

Bootstrap Data Set
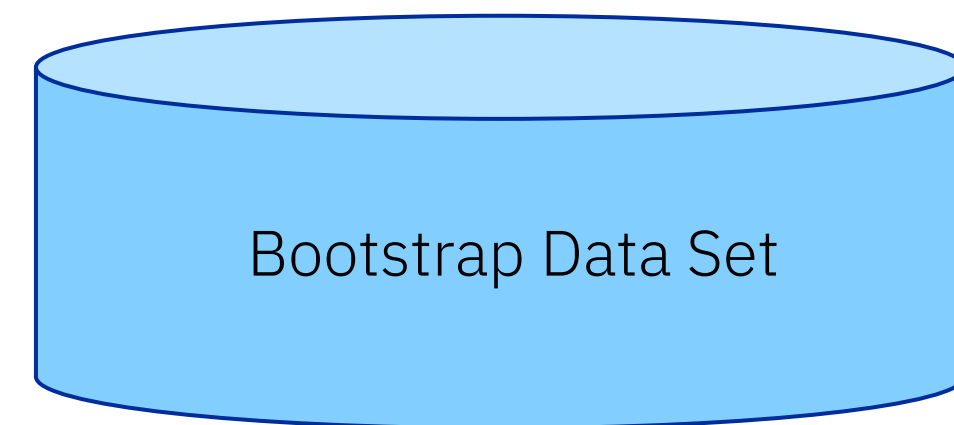
An inventory of all active and archived log data sets known to IBM MQ.

Circular Logs

Active Log File

Active Log Files
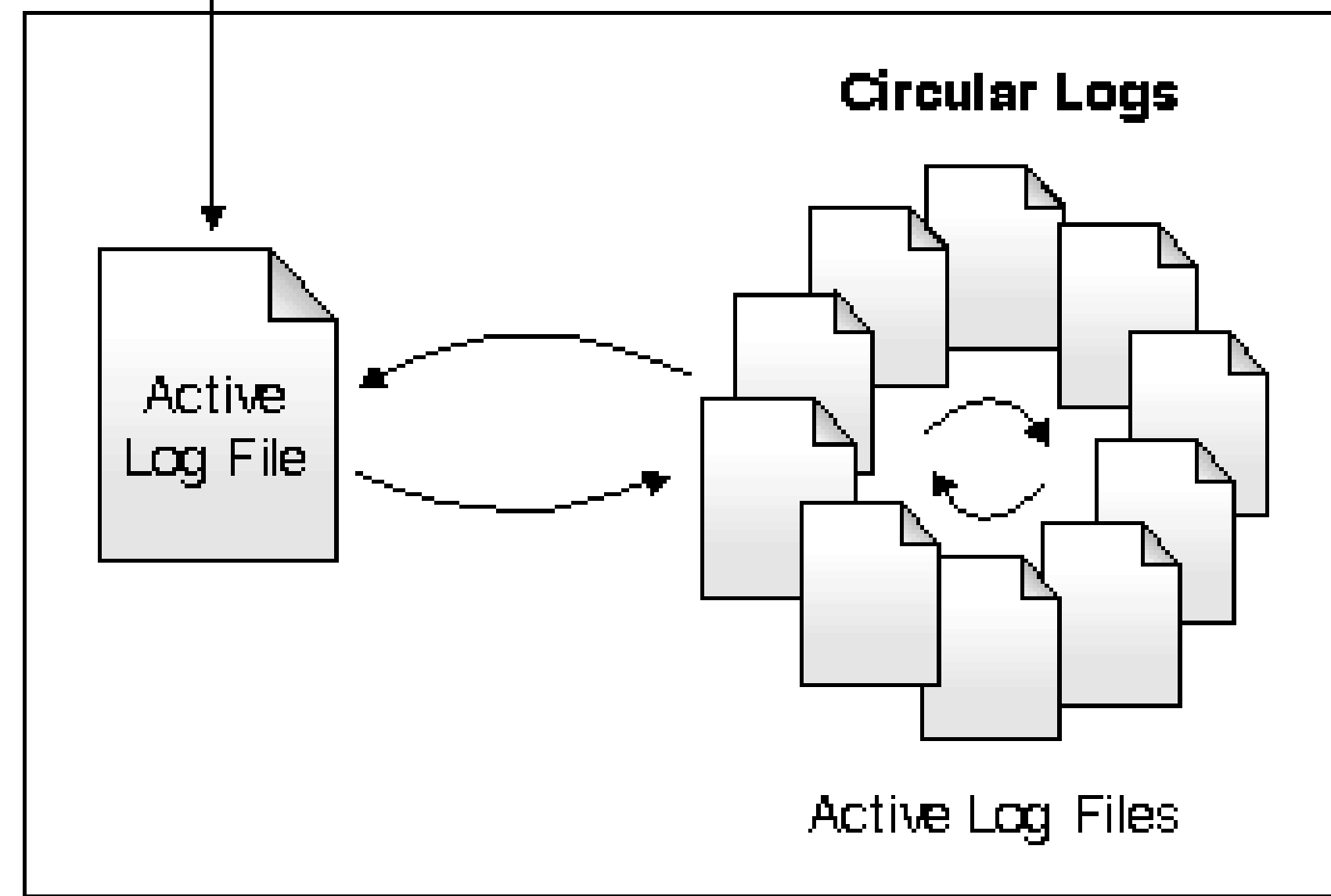
Offloading

Archive Log

What does a log file look like?
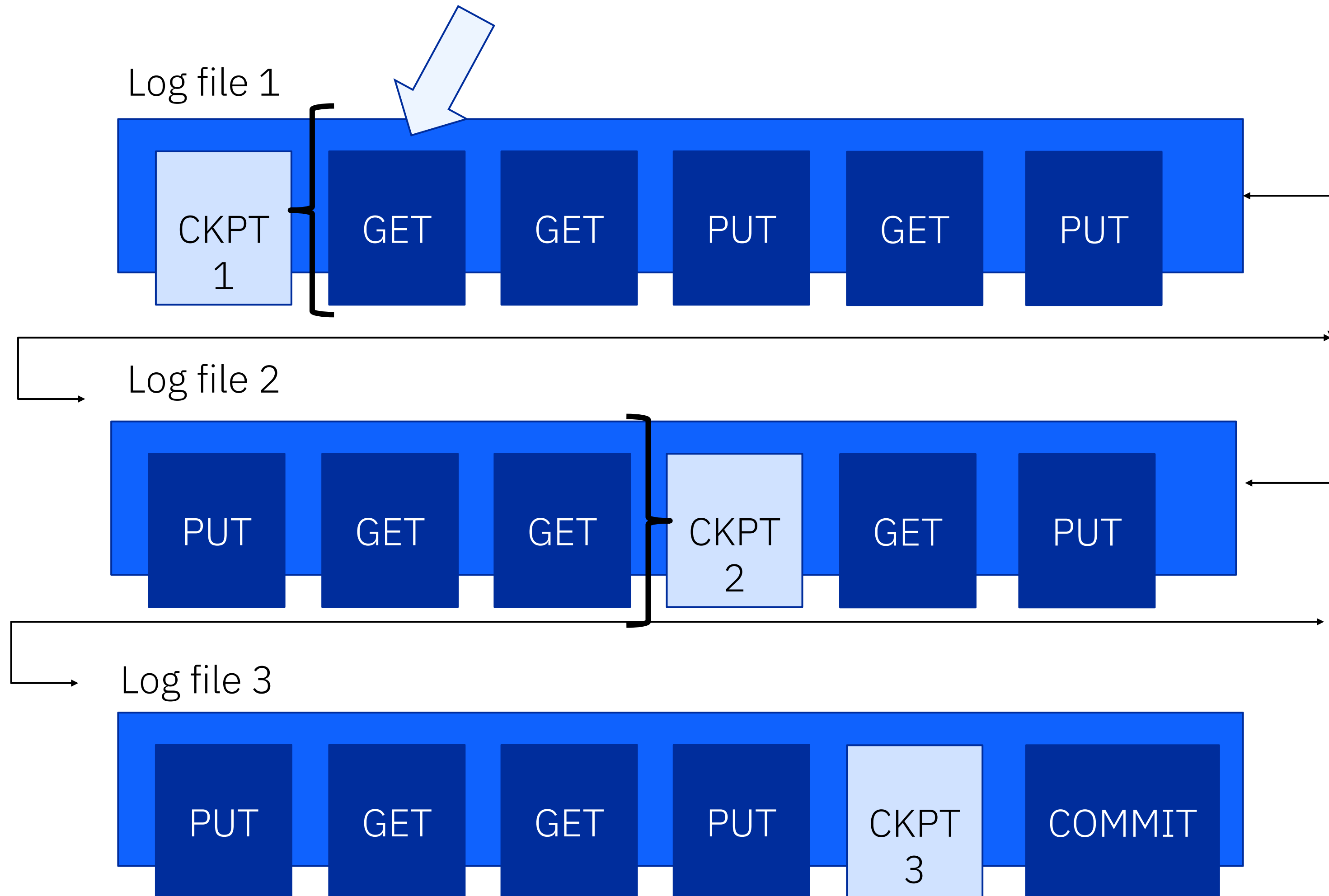
Log file 1

| CKPT 1 | GET | GET | PUT | GET | PUT |

Log file 2

| PUT | GET | GET | CKPT 2 | GET | PUT |

Log file 3

| PUT | GET | GET | PUT | CKPT 3 | COMMIT |

Queue layer

```
DEFINE QLOCAL(QUEUED)
STGCLASS(ARC1)
```

Storage class layer

```
DEFINE STGCLASS(ARC1)
PSID(A)
```

Page set layer

```
DEFINE PSID(01) BUFFPOOL(1)
```
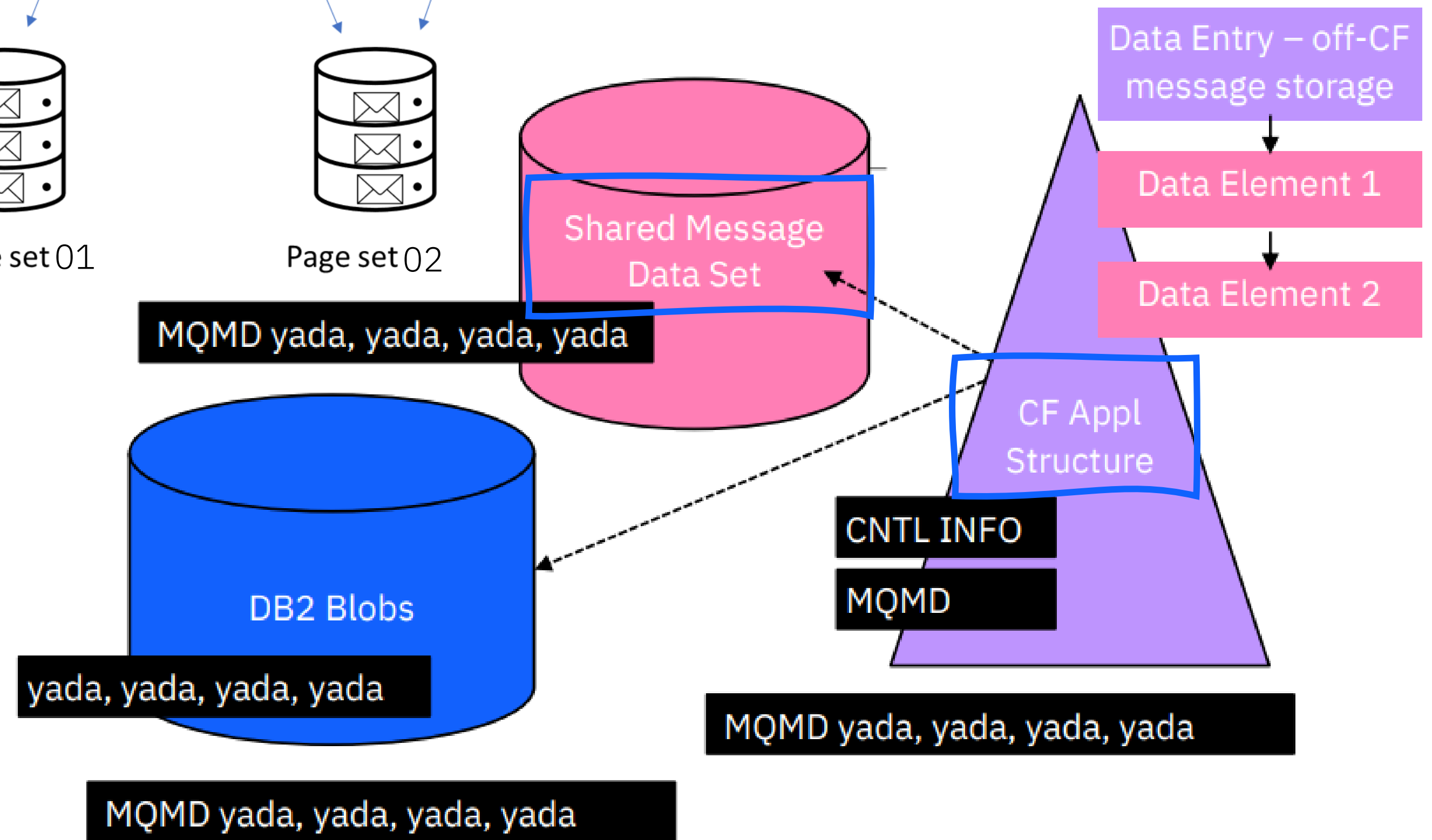
Define these bottom-up

Queue D   Queue E   Queue F   Queue G

ARC1   ARC1   ARC2   ARC3

Page set 01   Page set 02

Shared Message Data Set

MQMD yada, yada, yada, yada

DB2 Blobs

yada, yada, yada, yada

MQMD yada, yada, yada, yada

Data Entry – off-CF message storage

Data Element 1

Data Element 2

CF Appl Structure

CNTL INFO

MQMD

MQMD yada, yada, yada, yada

Circular Logs

Active Log File

Active Log Files

Concept check

When I want to offload messages from my list structure, I should use...

(a) DB2 Blobs

(b) Shared Message Data sets

(c) Page sets

Why might a short message be classified as a 4k or less?

Which address space is the Pub/Sub engine associated with?

(a) QMGR master address space

(b) CHIN address space

What is the size of an element in a list structure?

To recap...

Private queues use buffer pools, storage classes, and page sets to underpin queuing

Shared queues use CF list structures, shared message data sets, and BLOBs to underpin queuing

Both private and shared queues use **logging** for recovery

Understanding storage and logging sets the stage for our next presentation...

How to analyze your IBM MQ for z/OS SMF data

Dorothy Quincy
Client Technical Specialist

IBM MQ for z/OS
Washington Systems Center

IBM

There are two types of SMF records that are relevant to MQ for z/OS:

SMF 115:

Statistics data produced by an IBM MQ queue manager

SMF 116:

Accounting data produced by an IBM MQ queue manager

* You can look at this data in two ways – on z/OS and through exporting to CSV files

SMF 115

MQ Storage

SMF-QIS1.csv – Page Set Statistics

SMF-QPST.csv – Buffer Manager

SMF-QJST.csv – Log Manager

SMF-QSGM.csv - Storage

SMF-QSPH.csv - Storage

SMF-QSRS.csv - Storage

SMF-QSST.csv – Storage

SMF-QESD.csv – Shared Message Data Set

SMF-QEST.csv – Coupling Facility Statistics

SMF-Q5ST.csv – BLOB Statistics

MQ Requests

SMF-QLST.csv – Lock Manager

SMF-QMST.csv – Message Manager

SMF-QIST.csv – Data Manager Statistics

SMF-QCCT.csv – Channel Statistics

SMF-QCTADP.csv – Adapter Task Statistics

SMF-QCTDSP.csv – Dispatcher Task Statistics

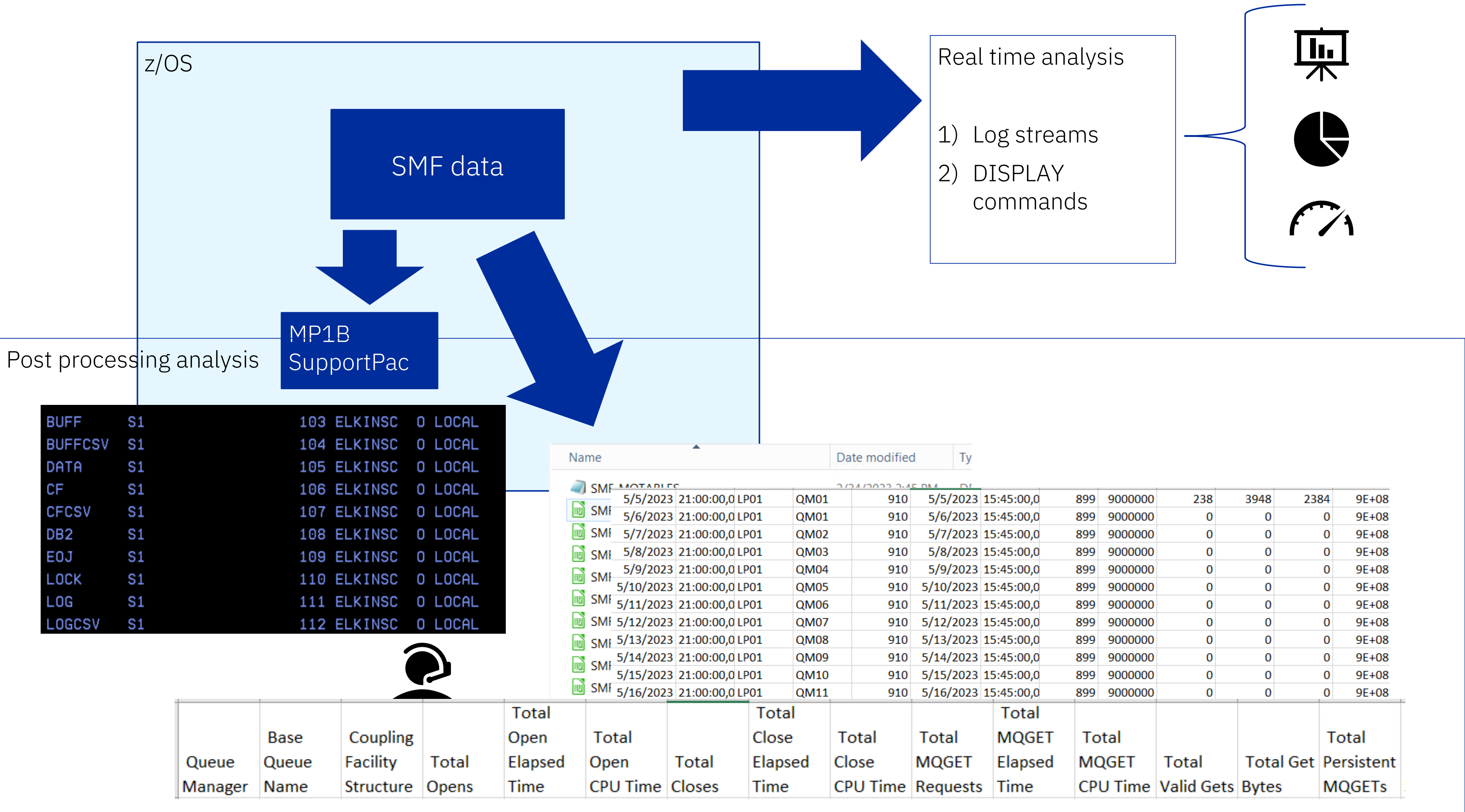SMF-QCTSSL.csv – SSL Statistics

SMF-QTST.csv – Publications Statistics

SMF 116

SMF-QCST.csv – Channel Accounting

SMF-WQ.csv – Task Queue Accounting

SMF-WTAS.csv – Task Accounting

SMF-WTID.csv – Task ID Accounting

MQ
Tasks

How do you look at SMF data?

z/OS

SMF data

Real time analysis

1) Log streams

2) DISPLAY commands

Post processing analysis

MP1B SupportPac

BUFF	S1		103 ELKINSC	0 LOCAL
BUFFCSV	S1		104 ELKINSC	0 LOCAL
DATA	S1		105 ELKINSC	0 LOCAL
CF	S1		106 ELKINSC	0 LOCAL
CFCSV	S1		107 ELKINSC	0 LOCAL
DB2	S1		108 ELKINSC	0 LOCAL
EOJ	S1		109 ELKINSC	0 LOCAL
LOCK	S1		110 ELKINSC	0 LOCAL
LOG	S1		111 ELKINSC	0 LOCAL
LOGCSV	S1		112 ELKINSC	0 LOCAL

Name		Date modified	Ty
SMF-MQTABLES		2/24/2023 2:45 PM	Pl...

5/5/2023 21:00:00,0	LP01	QM01	910	5/5/2023 15:45:00,0	899	9000000	238	3948	2384	9E+08
5/6/2023 21:00:00,0	LP01	QM01	910	5/6/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/7/2023 21:00:00,0	LP01	QM02	910	5/7/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/8/2023 21:00:00,0	LP01	QM03	910	5/8/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/9/2023 21:00:00,0	LP01	QM04	910	5/9/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/10/2023 21:00:00,0	LP01	QM05	910	5/10/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/11/2023 21:00:00,0	LP01	QM06	910	5/11/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/12/2023 21:00:00,0	LP01	QM07	910	5/12/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/13/2023 21:00:00,0	LP01	QM08	910	5/13/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/14/2023 21:00:00,0	LP01	QM09	910	5/14/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/15/2023 21:00:00,0	LP01	QM10	910	5/15/2023 15:45:00,0	899	9000000	0	0	0	9E+08
5/16/2023 21:00:00,0	LP01	QM11	910	5/16/2023 15:45:00,0	899	9000000	0	0	0	9E+08

Queue Manager	Base Queue Name	Coupling Facility Structure	Total Opens	Total Open Elapsed Time	Total Open CPU Time	Total Closes	Total Close Elapsed Time	Total Close CPU Time	Total MQGET Requests	Total MQGET Elapsed Time	Total MQGET CPU Time	Total Valid Gets	Total Get Bytes	Total Persistent MQGETs

32

CICS Transaction Rate Achieved

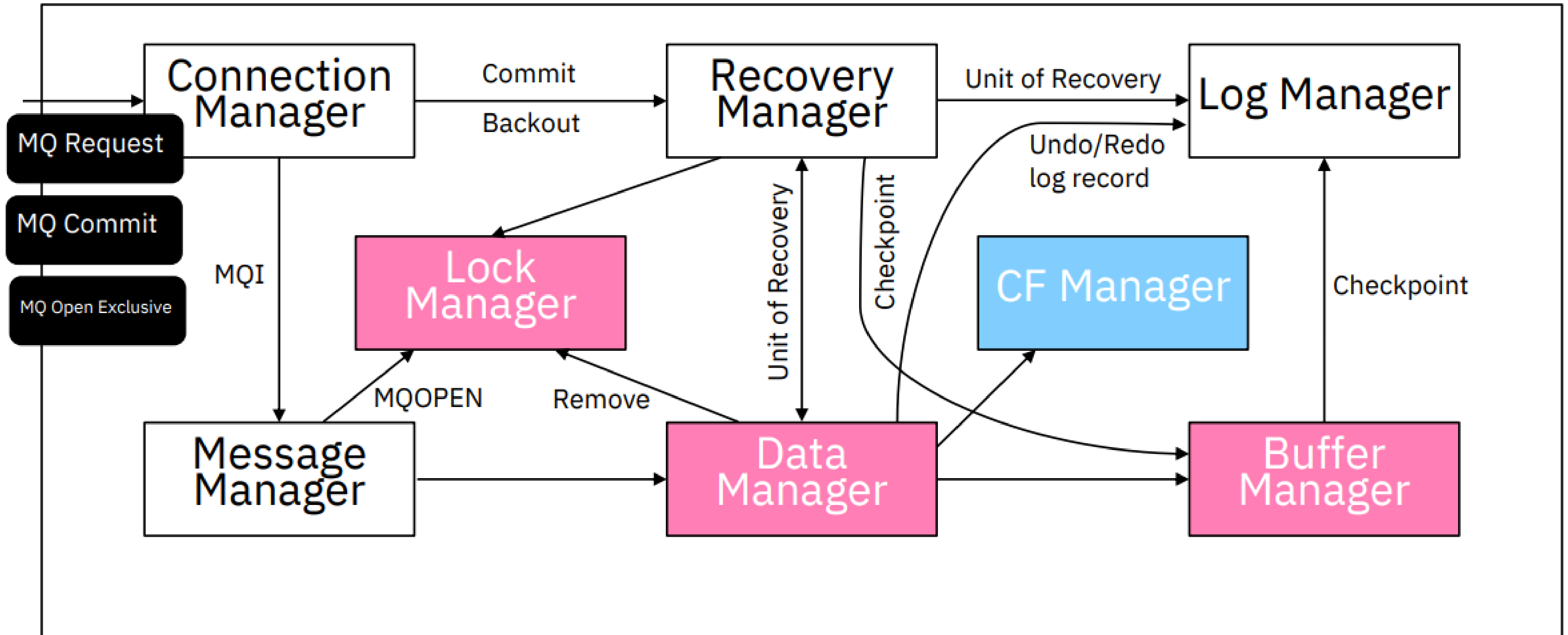All transactions monitored with MQ Accounting Class(3) enabled

■ SMF written to Dataset
■ SMF written to LogStream
■ SMF written to LogStream with hardware compression
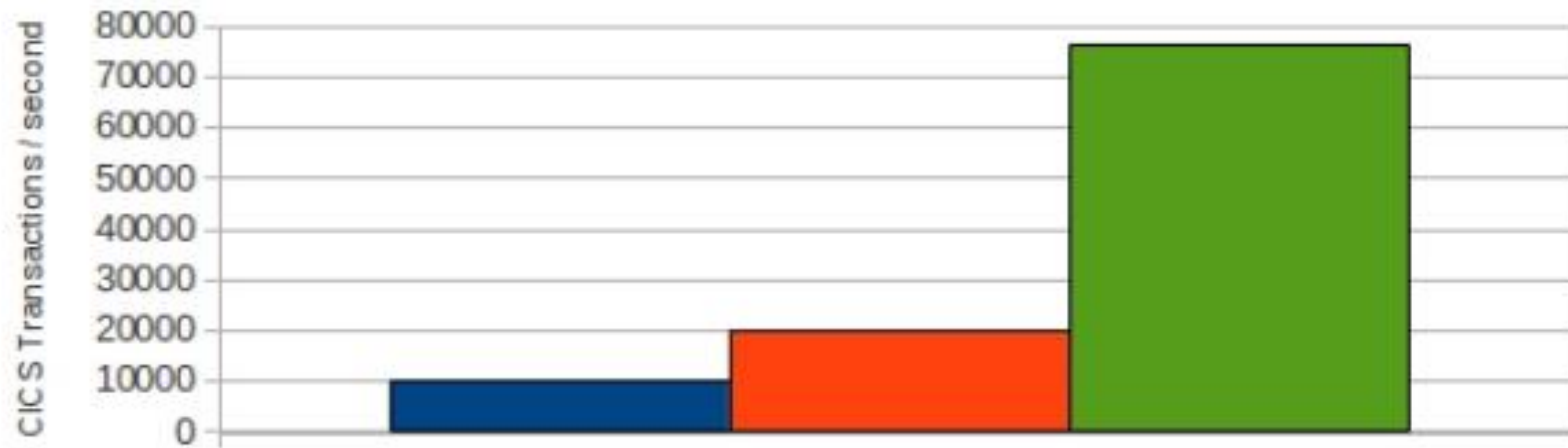
Interpreting SMF 115 data for private queues

Building Blocks
Resource Managers

First Line Managers

The threads within the QMMSTR address space who do the real work within each queue manager by interacting with applications and the underlying z/OS resource managers

They include:

Connection Manager – not the channel initiator, but local connections

Recovery Manager

Log Manager

Message Manager

Topic Manager

Data Manager

Buffer Manager

Lock Manager

Storage Manager

CF Manager

Security Manager

Buffer Manager

The buffer manager remains the largest area of opportunity for performance tuning for private queues.

1) Keeping private queue messages in-memory (buffer pools), especially for responsiveness driven workload, can be critical for meeting those goals.

2) Buffer pool contention, that is too much activity on a pool, can also cause delays due to the volume of work being done.

SMF-QPST.csv – Buffer Manager Statistics

- Buffer Pool
- Buffer Count
- Highest Used Percent
- Sync Writes
- Defer Write Thold Reached
- Sync Write Thold Reached
- Suspend No Buffers
- Pages Written

- Date
- Time
- LPAR
- QMgr
- MQ Version
- Interval Duration
- Location
- Pagefixed

- Current Stealable
- Getp Old Requests
- Getp New Requests
- DASD Read
- Set Write Pages
- DASD Write

- Lowest Stealable
- Highest Used
- Buffer Steals
- Buffer Steals Hash Changes

Log Manager

Traditionally, the act of logging persistent messages is the primary limiting factor of a queue manager. This is due to several constraints:

1. Single threaded work

2. Limited log buffers

3. Backing out work being done

4. MQ I/O limitations

5. Checkpointing

Log file 1

| CKPT 1 | GET | GET | PUT | GET | PUT |

Log file 2

| PUT | GET | GET | CKPT 2 | GET | PUT |

Log file 3

| PUT | GET | GET | PUT | CKPT 3 | COMMIT |

43
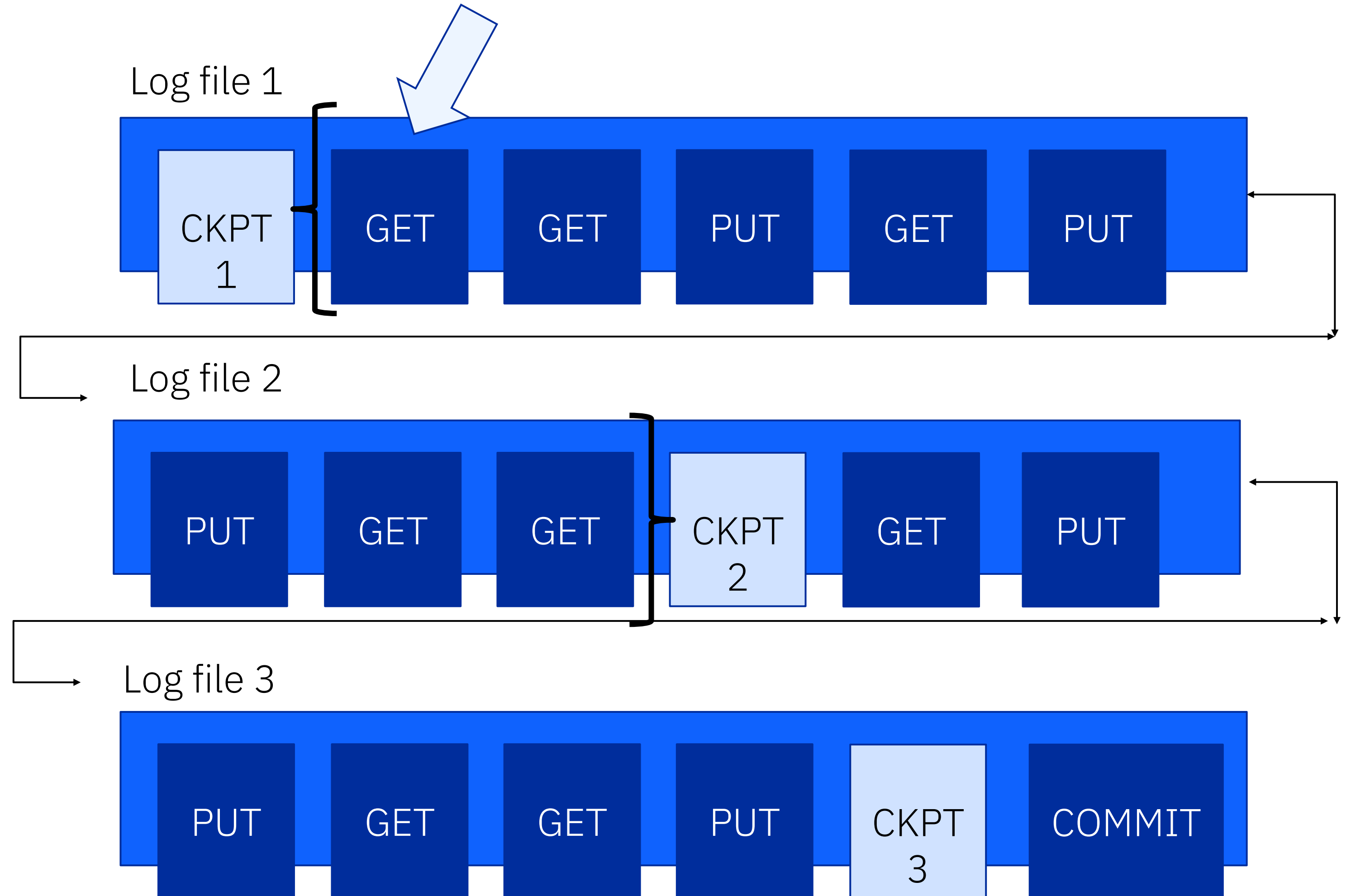
SMF-QJST.csv – Log Manager Statistics

- Log Reads

- Log Task Busy

- Checkpoints

- Unavailable buffer count

- I/O Max Duration Log Copy – how long it takes to do physical I/O
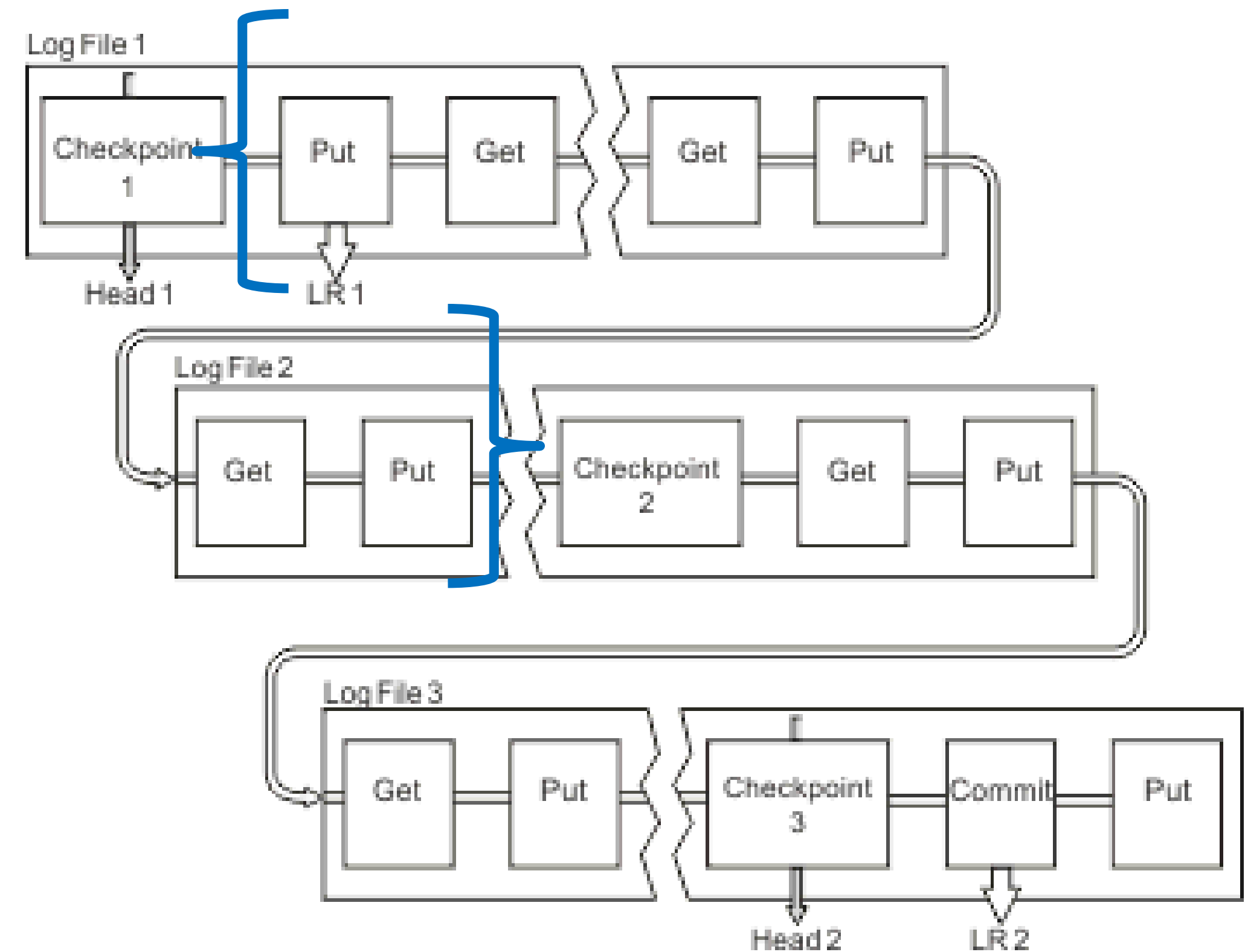
SMF-QMST.csv – Message Manager Statistics

- Records API requests

- Less useful for problem determination

- More useful for observing workload volume and peak periods

- Critical metrics here:

  - **MQPUT** – When an application made a put request onto the QM

  - **MQPUT1** – When an application has made a request to do an open and a put together

SMF-QTST.csv – Publications Statistics

- **Publications with no Subscriber to Topic**

- **Total Publication Request Count**

- Date Time   LPAR   QMgr

- Interval Duration

- Total Publication API Count

- Administrative Publications - Total   Proxy Publications - Total

- High point of Publications   Publications - Low Point

- Longest ET for publication in microseconds

- Total ET for publications in microseconds

SMF-QIST.csv – Data Manager Statistics

- **Message Mgr MQGETs**

- **Data Mgr MQGETs**

- **MQGET Difference**

- **Message Mgr MQPUTs; Message Mgr MQPUT1s; Total Message Mgr Puts**

- **Data Mgr MQPUTs**

- **PUT Difference**

- MSG_COUNT

- OBJECT CREATE; OBJECT DELETE;

- OBJECT PUT; OBJECT GET; OBJECT LOCATE

Implications:

GET difference can indicate scrolling

PUT difference can indicate put to waiting getter performance enhancement

Object creates and deletes shows temporary dynamic queue utilization

Positive put difference

MM puts > DM puts

MQPUT

Message Manager

Data Manager
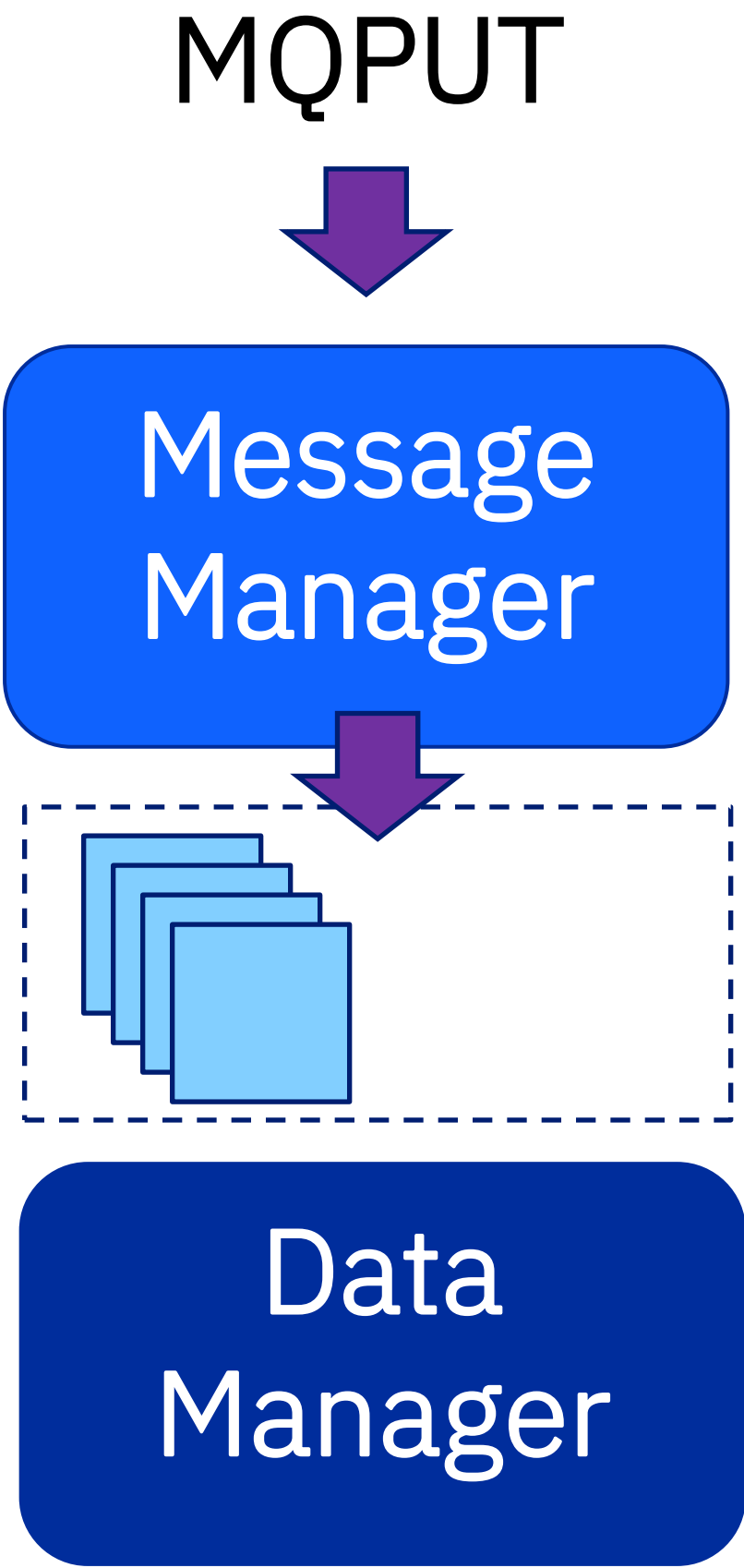
Put to waiting getter advantage

Negative put difference

DM puts > MM puts

MQPUT

Message Manager

Data Manager
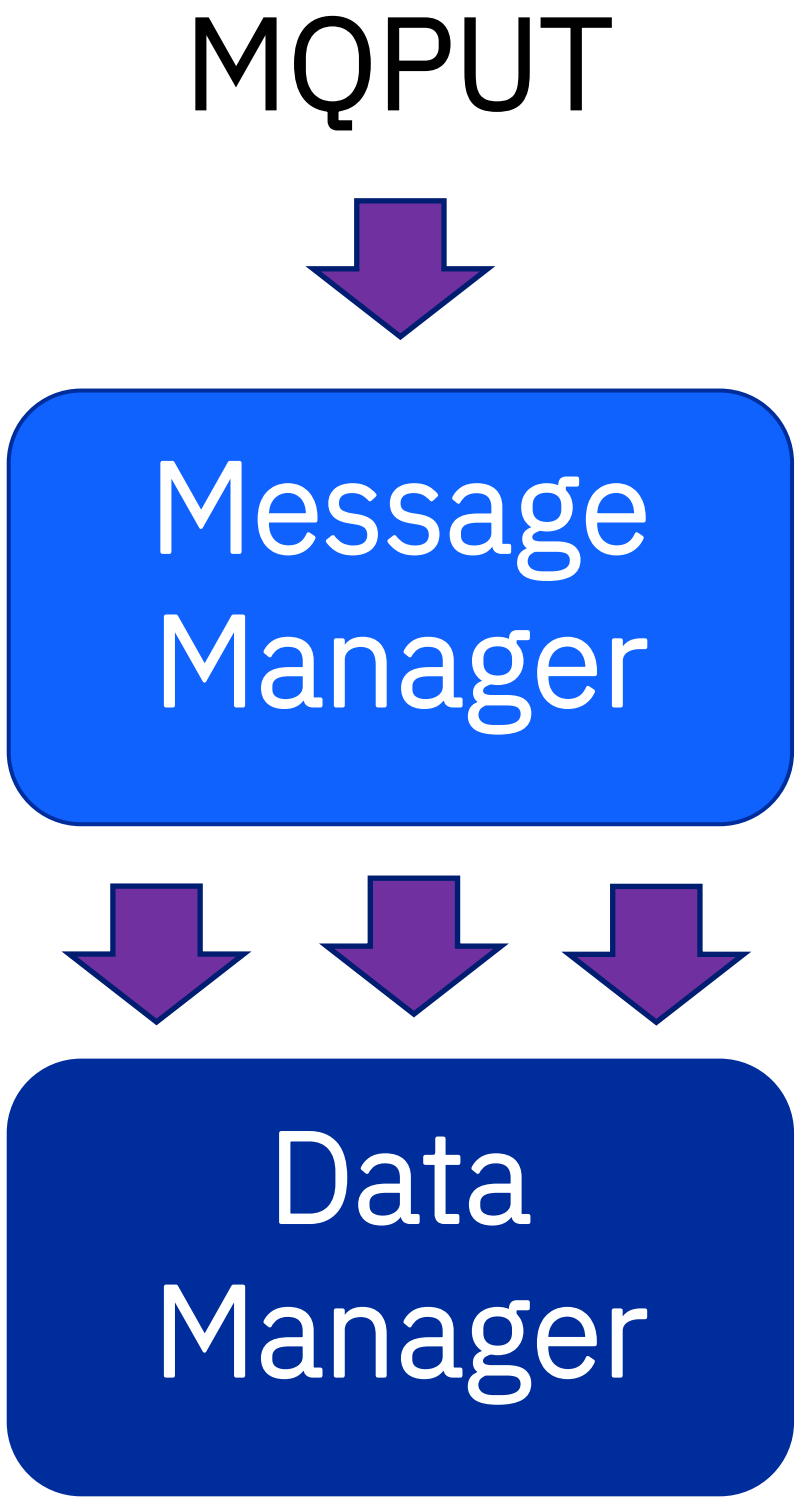
Publications or generated messages from triggering

Positive get difference

MM gets > DM gets

MQGET

Message Manager

Data Manager
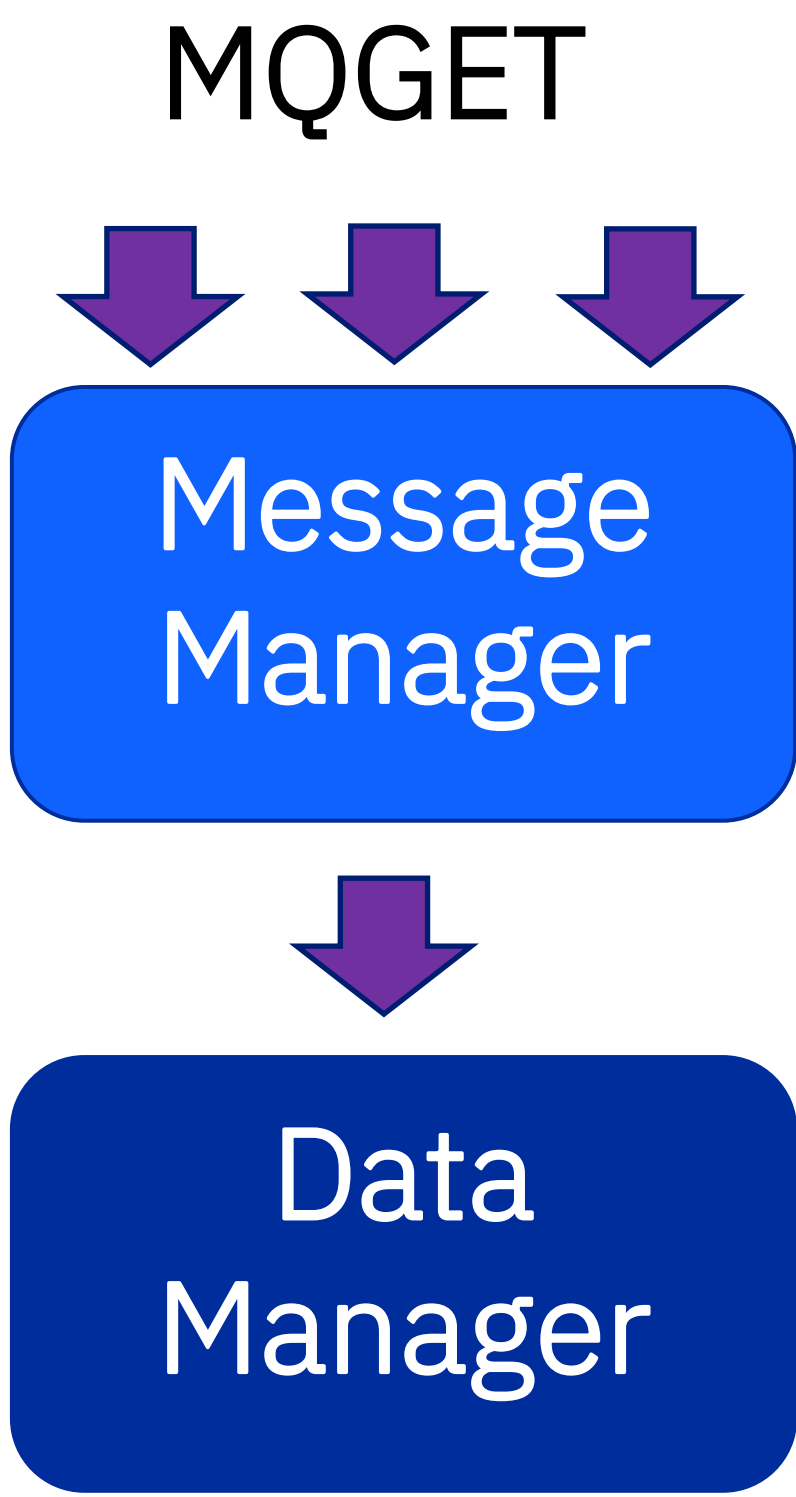
Target queue is empty

Not a problem

Negative get difference

DM gets > MM gets

MQGET

Message Manager

Data Manager
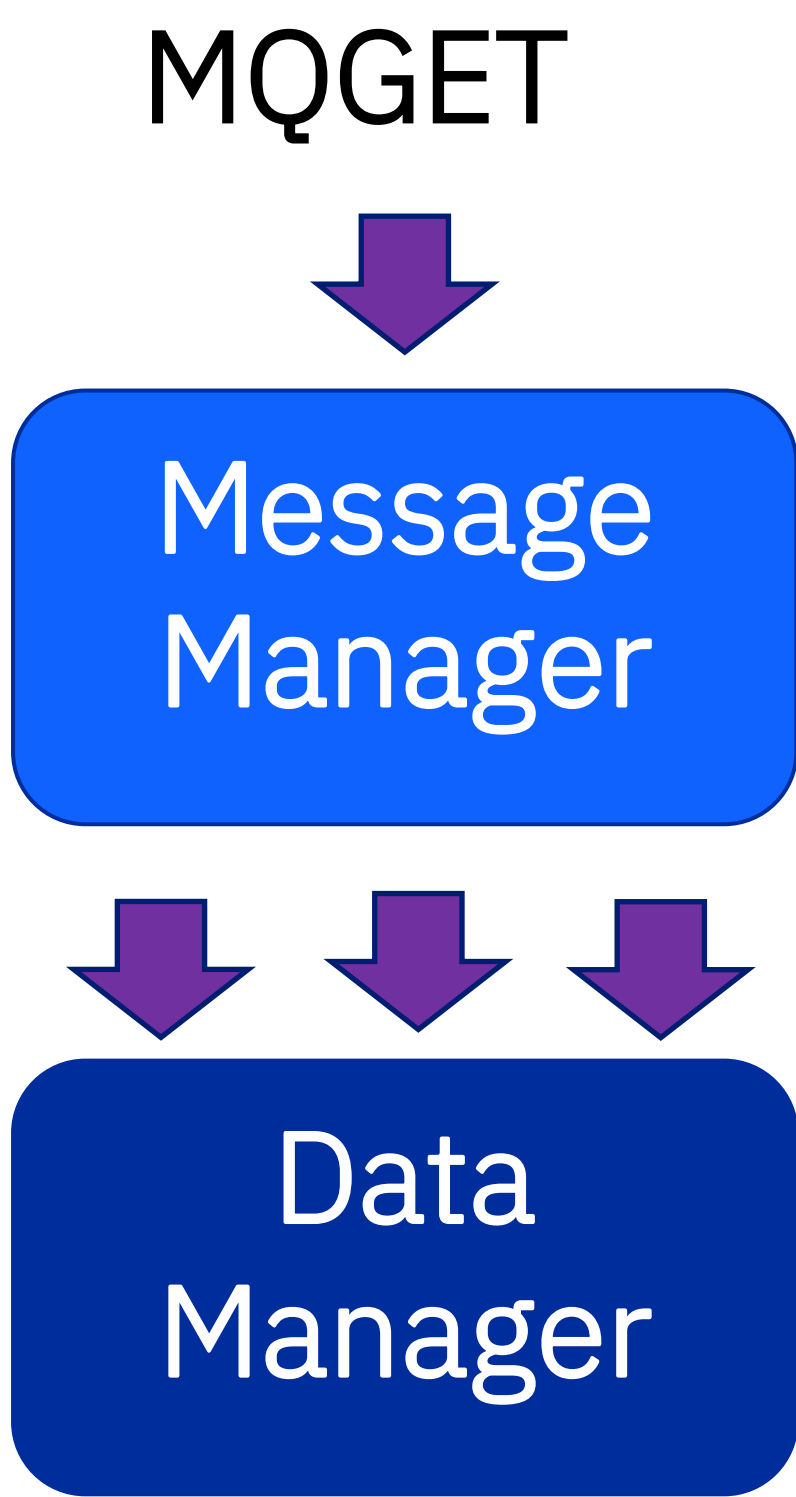
May indicate scrolling

Look for skipped messages

51

Interpreting SMF 115 channel information

CHINIT Address Space Structure

Network

QMGR1.TO.QMGR2
QMGR3.TO.QMGR2
QMGR4.TO.QMGR2
QMGR2.TO.QMGR4

Channel Initiator Address Space

Adapter Tasks

Dispatcher Tasks

SSL Tasks

Named server tasks

QM Master Address Space

QM, a channel needs to do a MQPUT
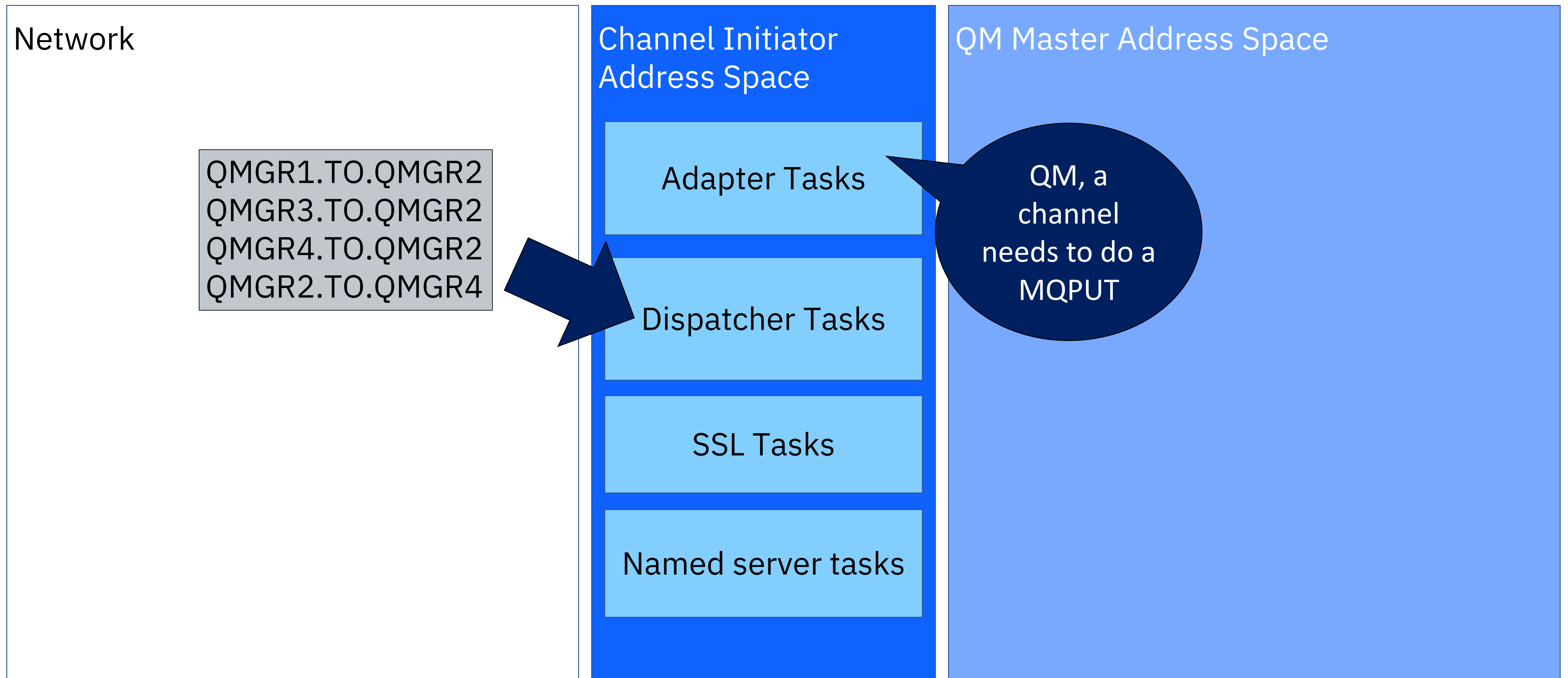
SMF-QCCT.csv – Channel Statistics

- CHINIT job name

- **QSG name if it is in a QSG**

- Peak number of current channels

- Peak number of active channels

- **MAXCHL - maximum permitted current channels**

- **ACTCHL - maximum permitted active channels**

- TCPCHL - maximum permitted

- TCP/IP channels

- LU62CHL - maximum permitted LU62 channels

- **Storage used by CHINIT**

SMF-QCTADP.csv – Adapter Task Statistics

- Date LPAR   QMgr

- **Adapter Task Number – always unique**

- **Total Requests for this Adapter Task**

- Total CPU for this Adapter Task

- Total Elapsed Time for this Adapter Task – could be in a wait state

- Total Wait Time for this Adapter Task

How adapter tasks are assigned
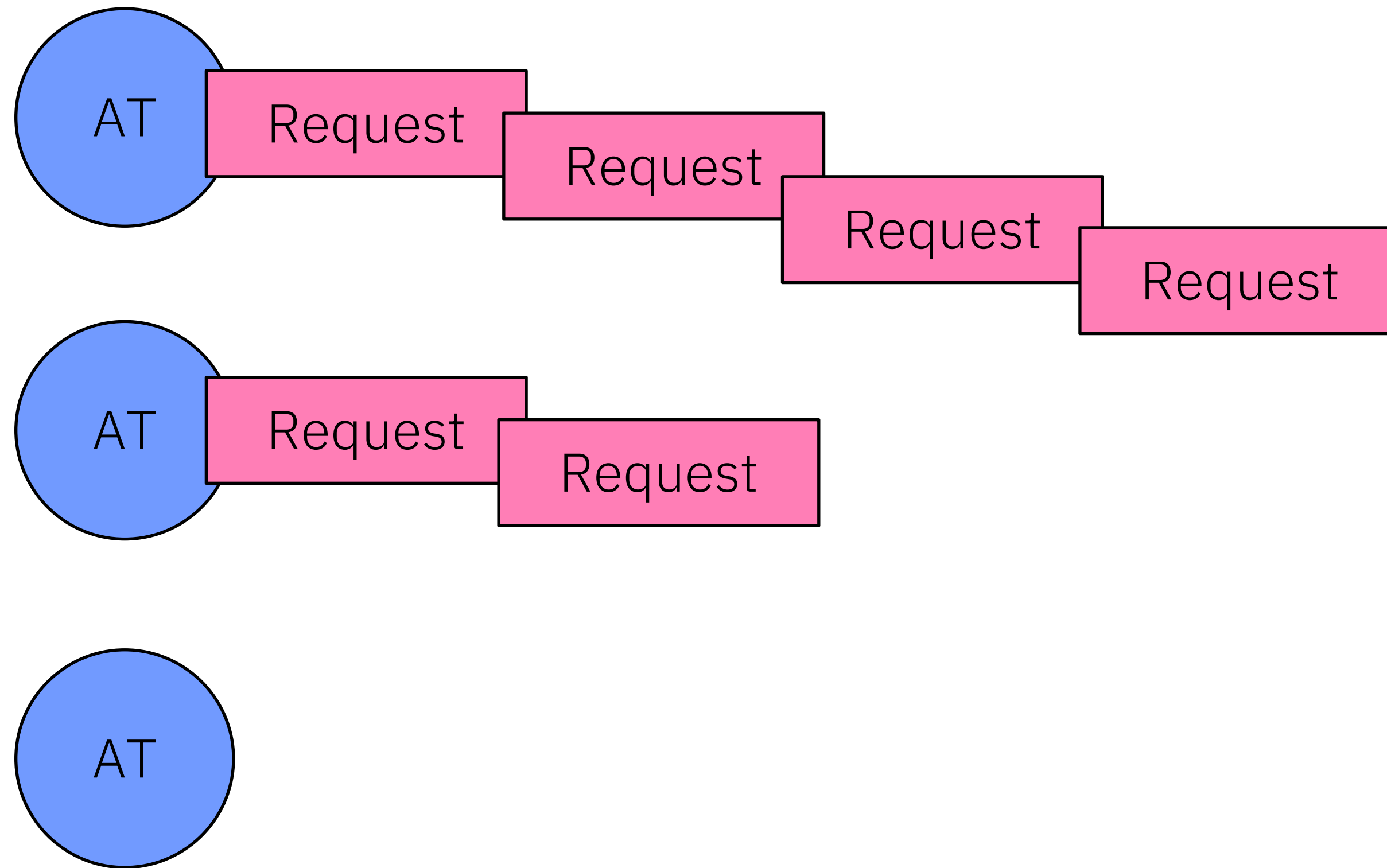
SMF-QCTDSP.csv – Dispatcher Task Statistics

- Date LPAR   QMgr

- **Dispatcher Task Number**

- **Total Requests for this Dispatcher Task**

- Total CPU for this Dispatcher Task

- **Total Elapsed Time for this Dispatcher Task**

- Total Wait Time for this Dispatcher Task

- **Maximum channels for this dispatcher task for Day**

How dispatcher tasks are assigned to channels

Scenario 1:
200 / 5 = 40 | 40 > 10 (from the rule of 10) | SO, 10 channels will be assigned to each dispatcher task

Active Channel Max of 200

n = 200

10 Channels
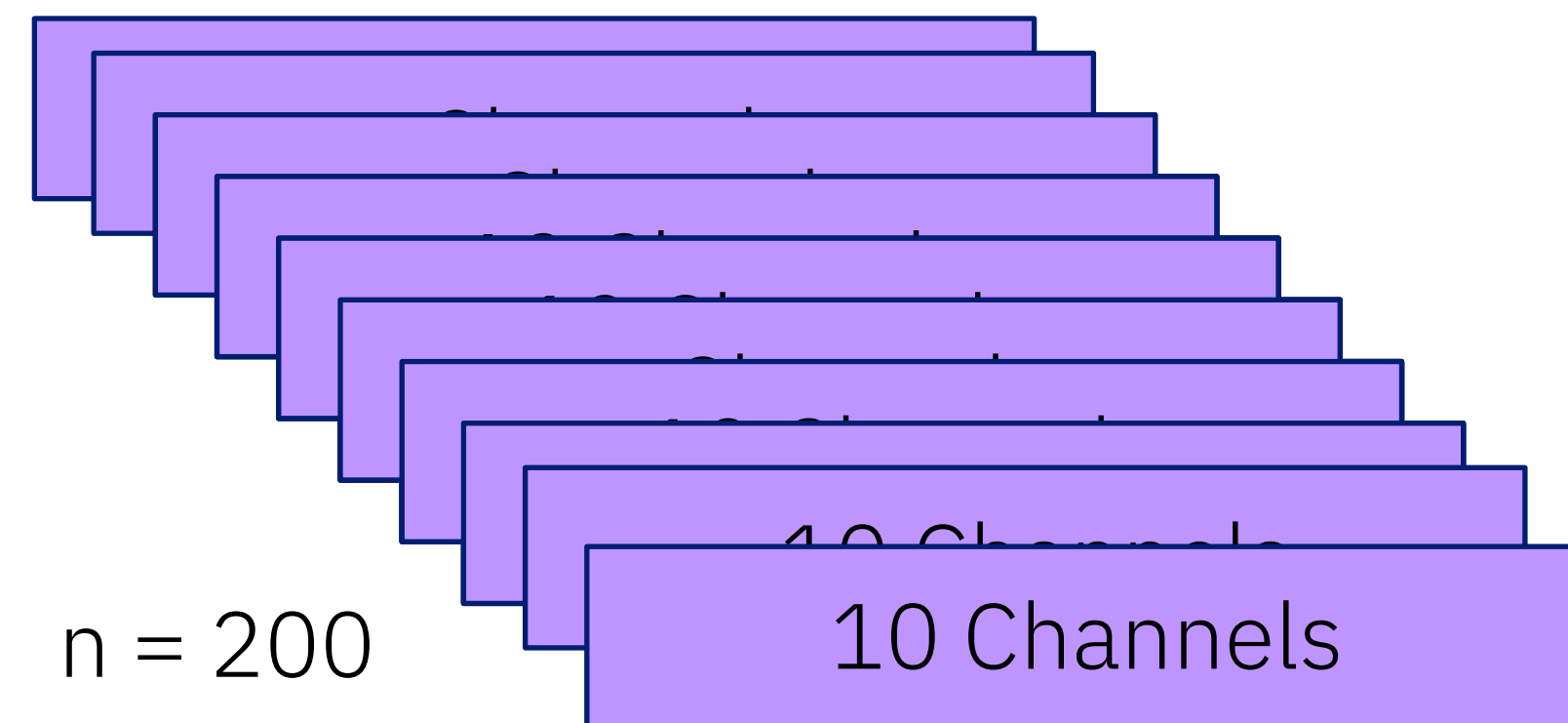
Dispatcher Tasks allocated

DP

n = 5
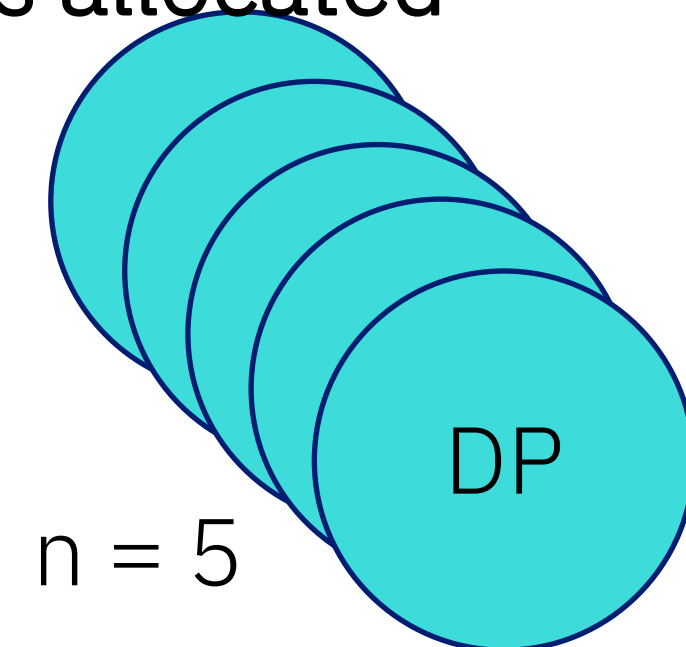
SMF statistics for shared queues

SMF-QEST.csv – Coupling Facility Statistics

- DATE TIME LPAR

- QMGR MQ_VERSION

- INTERVAL_START_DATE

- INTERVAL_START_TIME

- INTERVAL_DURATION

- **STRUCTURE_NAME**

- STRUCTURE_NUMBER

- **IXLLSTE_CALLS**

- **IXLLSTM_CALLS**

- **IXLLSTE_REDRIVES**

- **IXLLSTM_REDRIVES**

- STRUCTURE_FULL

SMF-Q5ST.csv – BLOB Statistics

- Date Time   LPAR   QMgr

- BLOB SELECTS

- BLOB INSERTS

- BLOB UPDATES

- BLOB DELETES

- BLOB LISTS

CF Application Structure

Db2 BLOBS

Shared Message Data Sets (SMDS)
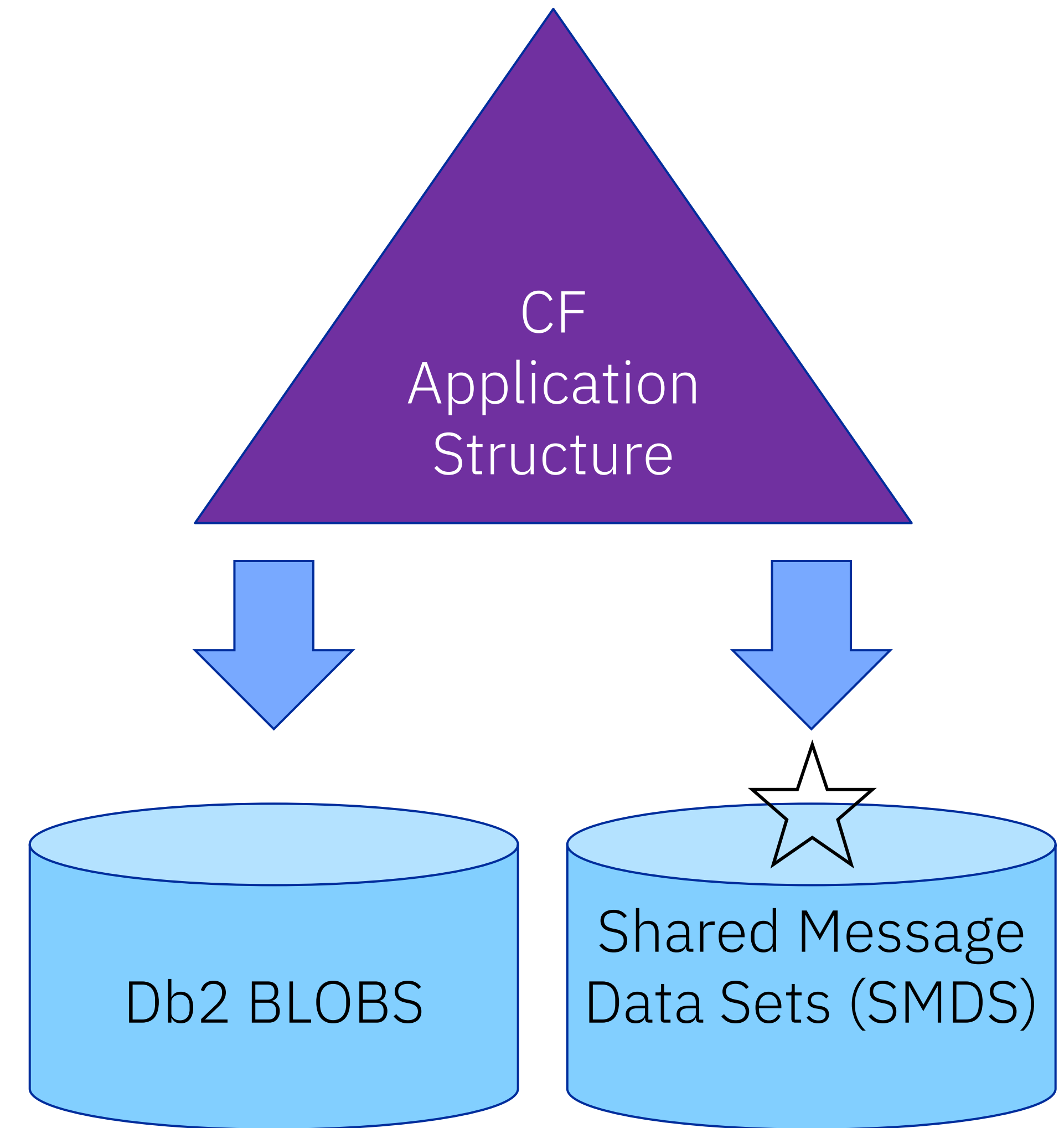
Single and multi-entry Redrives

Redrives, both single and multiple entry, are when the coupling facility detects that we need a larger buffer for returning a message.

Once queue manager gets the notification that it needs a larger buffer, it will use the buffer size necessary.

Using redrives allows for more flexibility in message sizes

Queue manager

1st try

x10

4K

8K

CF Application Structure
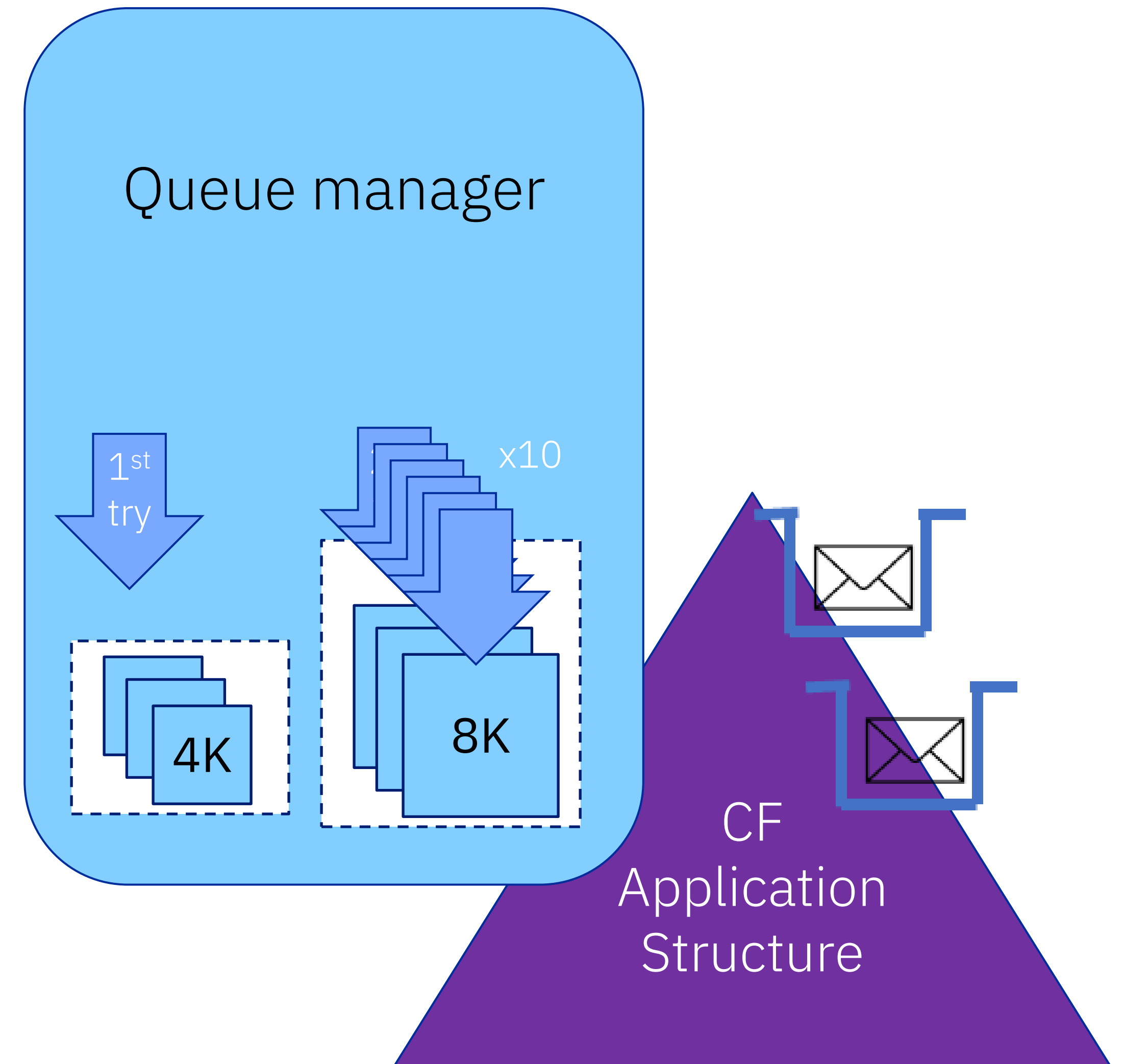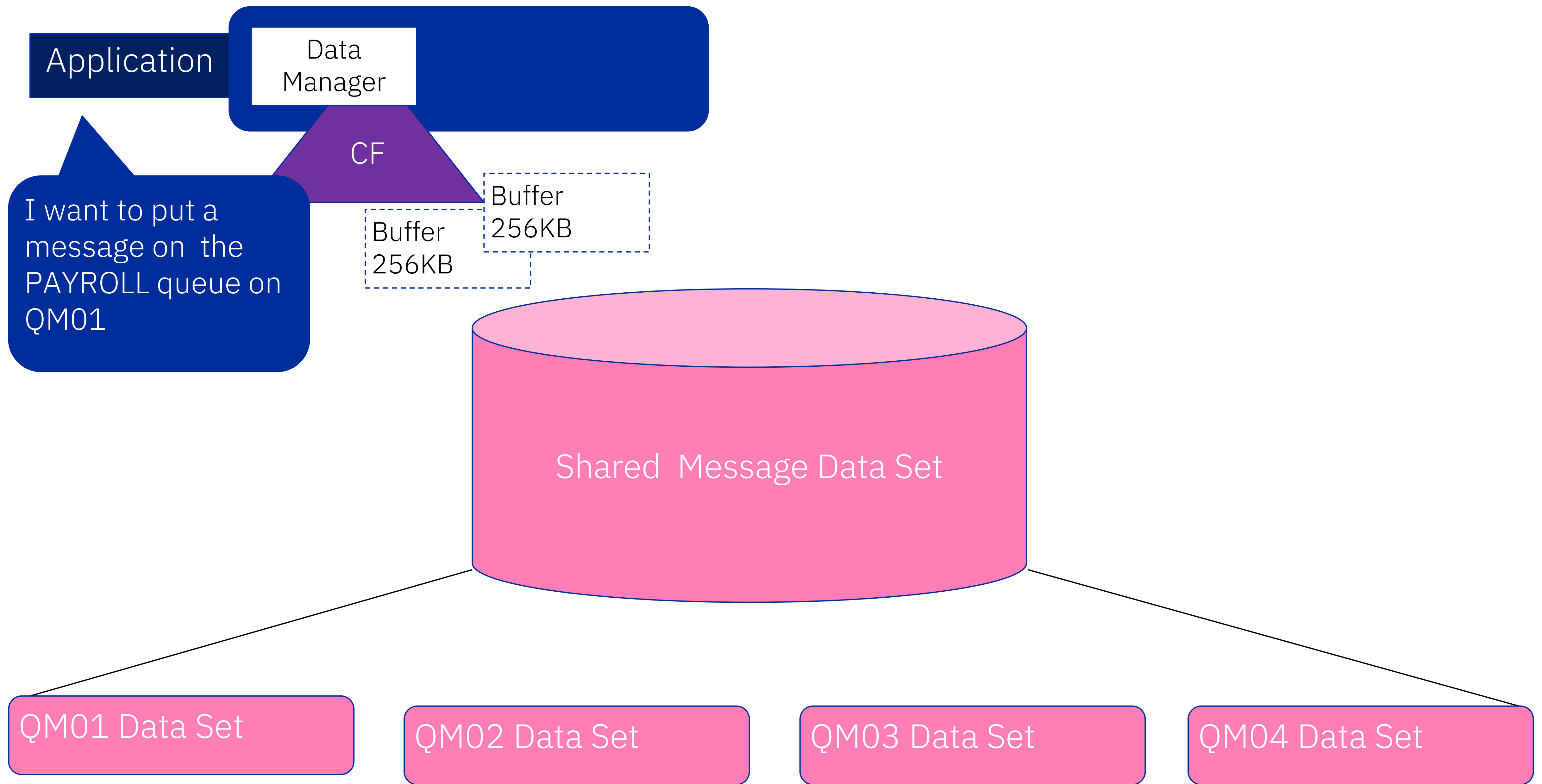
SMF 116 Accounting Data

SMF-QCST.csv – Channel Accounting

- **QMGR**

- **CHL_NAME**

- **TYPE**

- **CONNECTION_NAME**

- START_DATE

- DISPOSITION

- COLLECTION_TIME_DATE DATE

- COLLECTION_TIME_TIME CHAR(19)

- STATUS

- STAT_SETTING

- And more….

SMF-WTID.csv – Task ID Accounting

- DATE TIME LPAR QMGR
- MQ_VERSION
- **WTAS_CORRELATOR**
- **APPL_TYPE**
- **CONNECTION_NAME**
- OPERATOR_ID
- NID
- CORREL_HEX

- CORREL_CHAR
- UOW_ID
- ACCOUNTING_TOKEN
- **CHANNEL_NAME**
- CHANNEL_CONNECTION_NAME
- CONTEXT_TOKEN
- MVS_USERID

SMF-WTAS.csv – Task Accounting

- QMgr Correl

- Longest_Latch

- Max Latch Wait Microseconds

- Max Latch Wait Type

- Start Date

- Start Time

- QMgr

- Correl

- Type 11 Latch Wait Time (Over 5000 mics.)

- Type 11 Wait Count

- Task Start Date

- Task Start Time

Latching types

Type Number	What it means
11	Typically - ensuring serialization across API requests on a queue
12	Buffer pool latching
15	First-open Last closed effect on shared queues
16	Serializing access to a PSID
19	Serializing access to a buffer page
21	Log buffer latching
23	Serializing access to the BSDS
24	Serializing access across waiting getters
25	To serialize access to a particular lock AND for creating Accounting
30	Task switching and security checks
31	Often, queue scanning when a queue is not indexed, occasionally when checking security and freeing storage
32	Serializing requests to get or free storage

SMF-WQ.csv – Task Queue Accounting

- Base Queue Name

- Open Name

- Bufferpool ID

- Pageset ID

- Coupling Facility Structure

- Total Opens

- Total Open Elapsed Time

- Total Open CPU Time

- Total Closes

- Total Close Elapsed Time

- Total Close CPU Time

- Total MQGET Requests

- Total MQGET Elapsed Time

- Total MQGET CPU Time

- Total Valid Gets

- Total Get Bytes

- Total Persistent MQGETs

- **Total Messages Skipped**

- Total Messages Expired

- Total MQPUT Requests

- Total MQPUT Elapsed Time

- Total MQPUT CPU Time

- Total MQPUT1 Requests

- Total MQPUT1 Elapsed Time

- Total MQPUT1 CPU Time

- **Total Valid MQPUTs**

- **Total Bytes Put**

- Total Puts to Waiting Getter

- Total Put1s to Waiting Getter

- Total Generated Messages

- Total Persistent MQPUTs

- Total Persistent MQPUT1s

- **Max Depth on Queue**

- **Max Time on Queue**

- Min Time on Queue

- Total Inquiries

- Total Sets

- **Get percent unfulfilled**

- **Index Type**

To recap...

SMF 115 data

SMF 115 for shared queues specifically
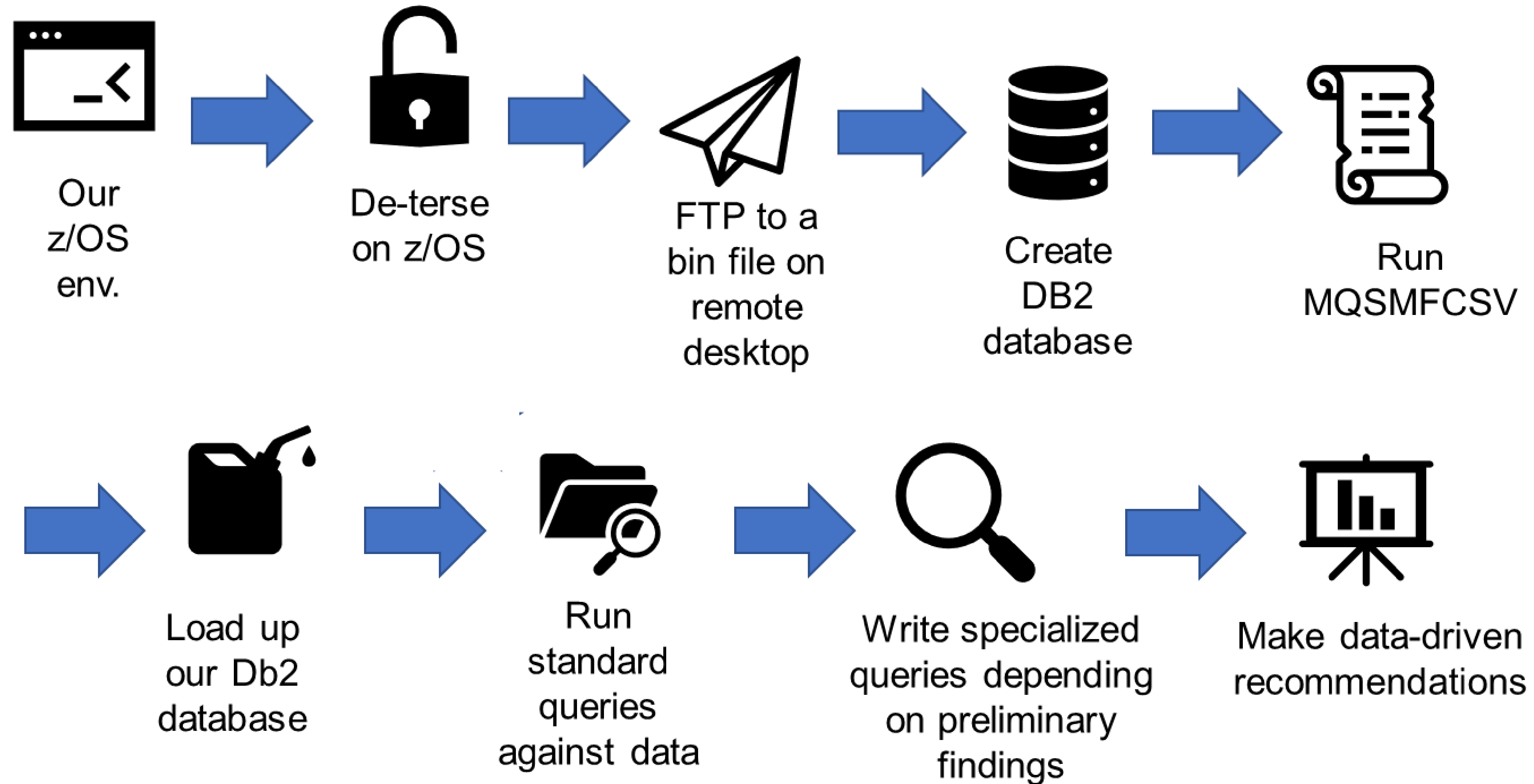
SMF 116 data

Concept check

What is a redrive?

1) When you have to drive the same route twice

2) When the queue manager makes a call to the CF and the CF responds saying you need a larger buffer

3) When an adapter and dispatcher task interfere with one another

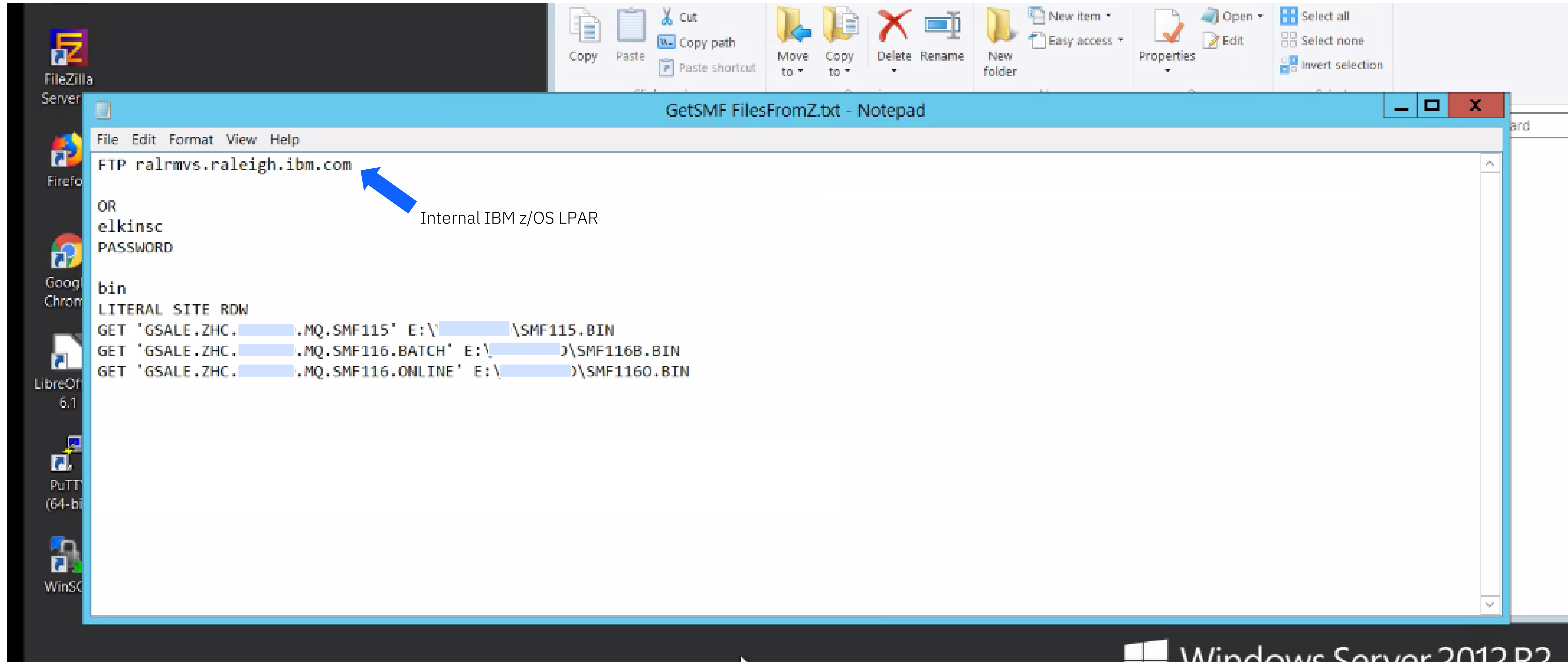What is the difference between assignment for adapter and dispatcher tasks?

1) Dispatcher tasks are worker tasks for channel code to run on. Adapter task issue MQ API calls on behalf of channels

2) Dispatcher tasks connect the queue manager to the CHINIT address space. Adapter tasks connect the network to the CHINIT address space.

3) Adapter tasks can become more evenly distributed by reducing the MAXACTCHL.
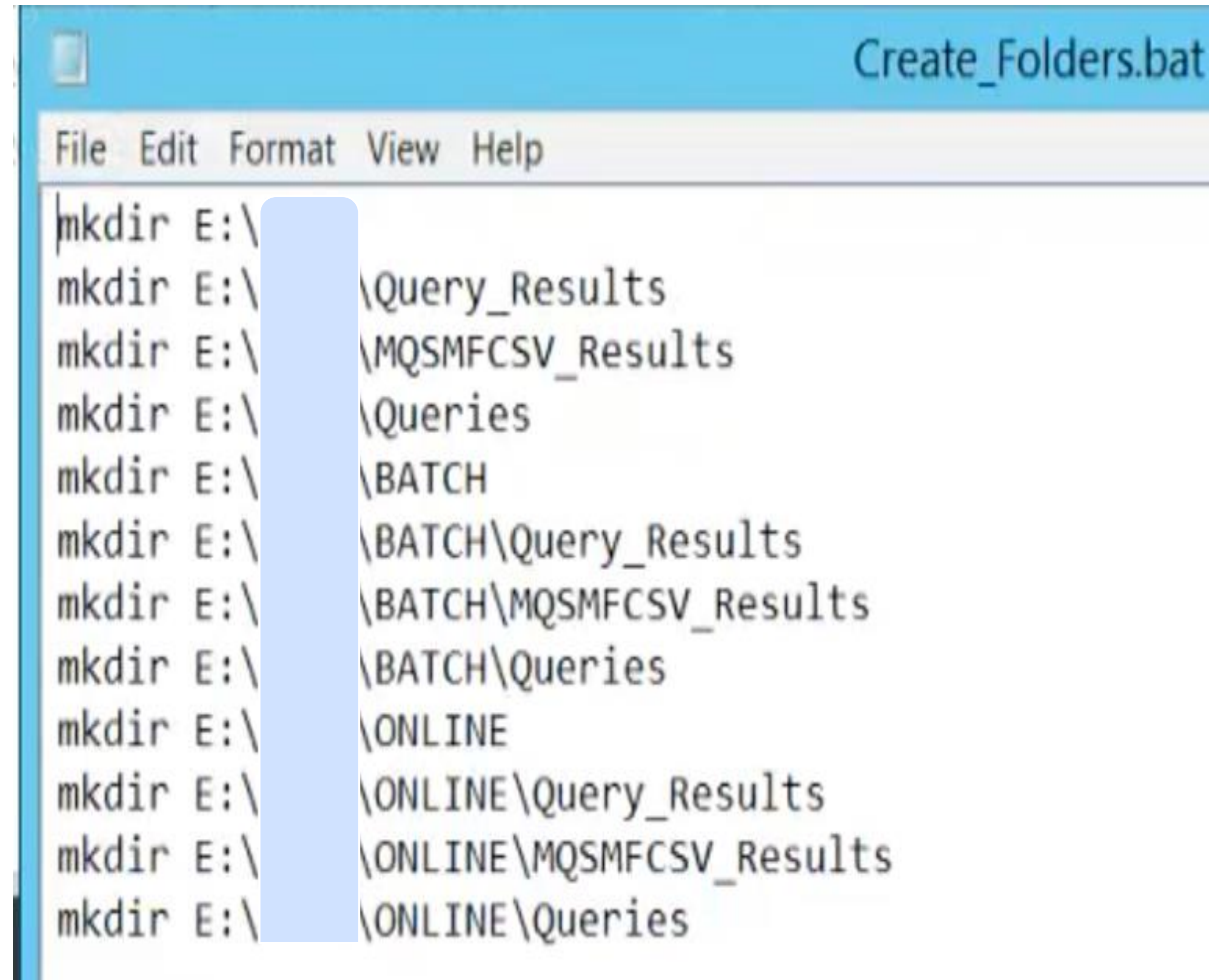
Demonstration

Our z/OS env. → De-terse on z/OS → FTP to a bin file on remote desktop → Create DB2 database → Run MQSMFCSV

→ Load up our Db2 database → Run standard queries against data → Write specialized queries depending on preliminary findings → Make data-driven recommendations

Pull data down from z/OS


```
FTP ralrmvs.raleigh.ibm.com          Internal IBM z/OS LPAR

OR
elkinsc
PASSWORD

bin
LITERAL SITE RDW
GET 'GSALE.ZHC.        .MQ.SMF115' E:\        \SMF115.BIN
GET 'GSALE.ZHC.        .MQ.SMF116.BATCH' E:\        )\SMF116B.BIN
GET 'GSALE.ZHC.        .MQ.SMF116.ONLINE' E:\        )\SMF116O.BIN
```

Create file system

Create_Folders.bat

File Edit Format View Help

```
mkdir E:\
mkdir E:\        \Query_Results
mkdir E:\        \MQSMFCSV_Results
mkdir E:\        \Queries
mkdir E:\        \BATCH
mkdir E:\        \BATCH\Query_Results
mkdir E:\        \BATCH\MQSMFCSV_Results
mkdir E:\        \BATCH\Queries
mkdir E:\        \ONLINE
mkdir E:\        \ONLINE\Query_Results
mkdir E:\        \ONLINE\MQSMFCSV_Results
mkdir E:\        \ONLINE\Queries
```
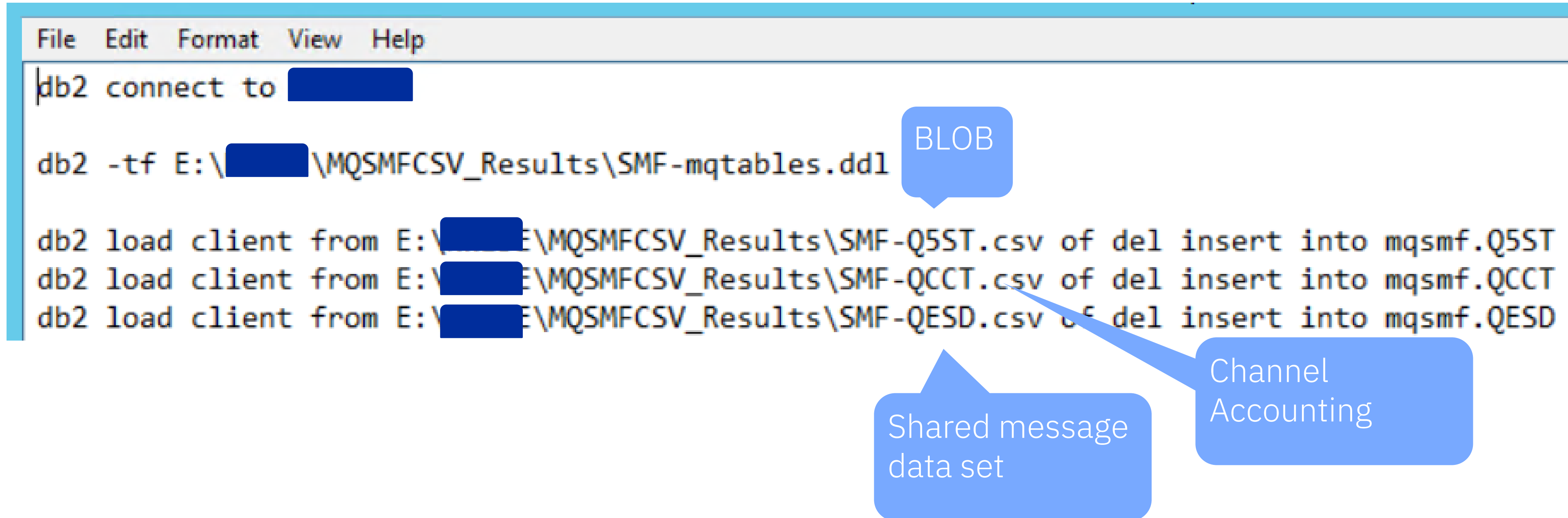
Create DB2 database(s)

```
Db2 CREATE DATABASE XXXXXX AUTOMATIC STORAGE YES ON 'E:\XXXXXX'
DBPATH ON 'C:' ;
```

Run MQSMFCSV

```
mqsmfcsv -f sql -i E:\customerdir\MQ115.bin  -o E:\customerdir\MQSMFCSV_Results
```

Load the database(s)

Our z/OS env. → De-terse on z/OS → FTP to a bin file on remote desktop → Create DB2 database → Run MQSMFCSV → Load up our Db2 database → Run standard queries against data → Write specialized queries depending on preliminary findings → Make data-driven recommendations

Check who is here

```
SELECT DISTINCT LPAR,
QMGR
FROM MQSMF.QMST (message manager)
 ORDER BY LPAR.
```

Customize some queries
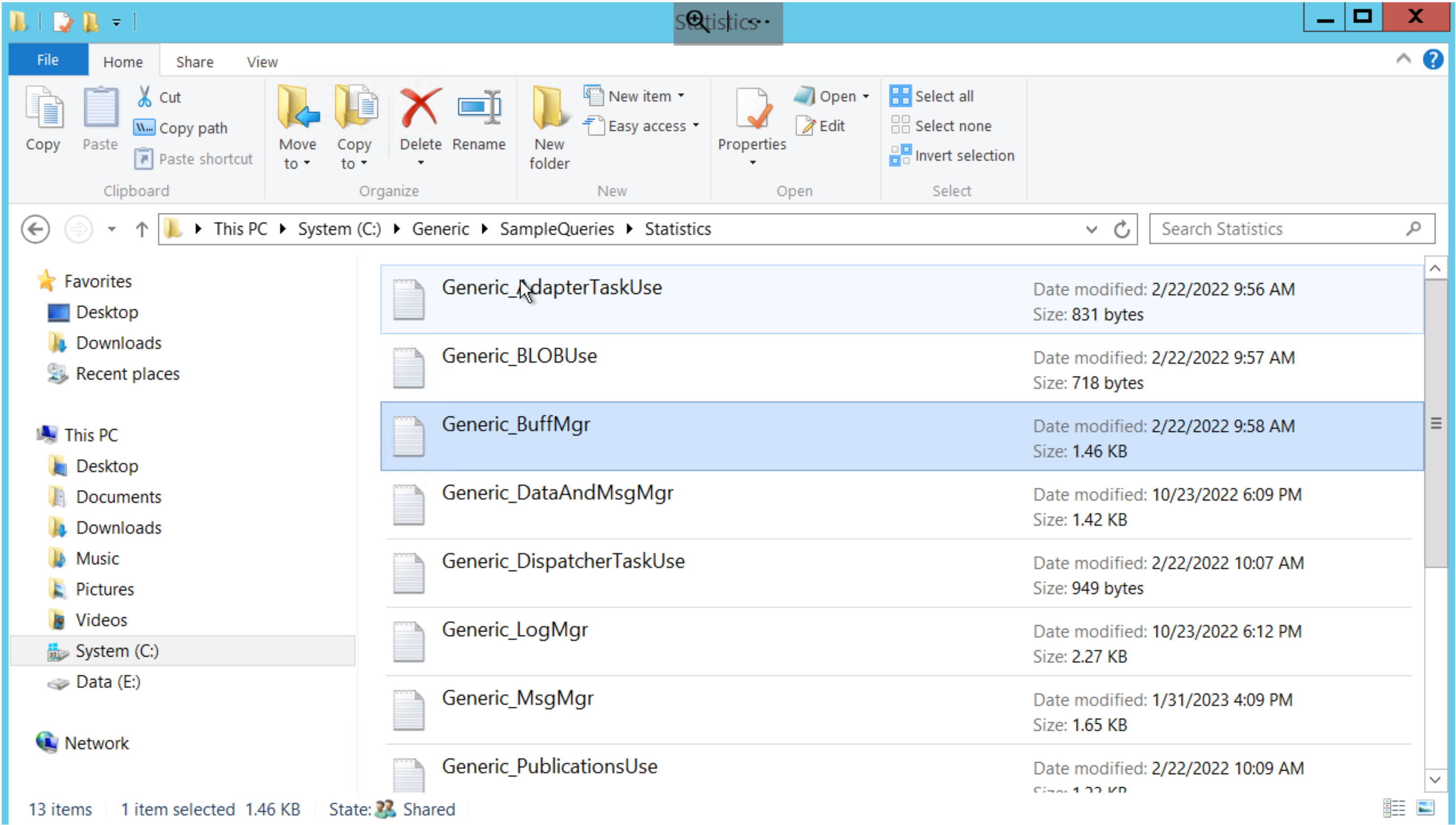
Run the queries against the data

```
DB2 -TVF E:\          \Queries\          1_AdapterTaskUse.txt
DB2 -TVF E:\          \Queries\          1_BuffMgr.txt
DB2 -TVF E:\          \Queries\          1_DataAndMsgMgr.txt
DB2 -TVF E:\          \Queries\          1_DispatcherTaskUse.txt
DB2 -TVF E:\          \Queries\          1_LogMgr.txt
DB2 -TVF E:\          \Queries\          1_LongLatches.txt
DB2 -TVF E:\          \Queries\          1_MsgMgr.txt
DB2 -TVF E:\          \Queries\          1_OpenCloseCF.txt
DB2 -TVF E:\          \Queries\          1_PublicationsUse.txt
DB2 -TVF E:\          \Queries\          1_QueueSumm.txt
DB2 -TVF E:\          \Queries\          1_SharedQueueSumm.txt
DB2 -TVF E:\          \Queries\          1_Type11Latches.txt
DB2 -TVF E:\          \Queries\          3_AdapterTaskUse.txt
DB2 -TVF E:\          \Queries\          3_BuffMgr.txt
DB2 -TVF E:\          \Queries\          3_DataAndMsgMgr.txt
DB2 -TVF E:\          \Queries\          3_DispatcherTaskUse.txt
DB2 -TVF E:\          \Queries\          3_LogMgr.txt
```

Good starting point scenarios

1. Check if buffer pools are highly utilized

2. If you see a highly utilized buffer pool, investigate which queues are most active for given buffer pool

3. Move queues to less busy buffer pool

1. Check for get difference on your data manager

2. Verify get difference is due to skipped messages

3. If get difference is negative, look at queue summary to identify which queues are busy and not indexed

4. Index queues

1. Check adapter task utilization

2. Check adapter tasks aren't filled up due to scrolling

3. If adapter tasks are all full, allocate more adapter tasks within reason

Further reading and resources:

Understanding MQ SMF Data: MP1B MQSMF pdf

Capacity planning and tuning guide for IBM MQ on z/OS: MP16 pdf

https://github.com/ibm-wsc/mq-wildfire-mqv9zos/

MQ and SMF - How might I process the data? (ibm.com)

GitHub - ibm-messaging/mq-smf-csv: Simple formatter for MQ's SMF records to assist with import to spreadsheets and databases

Dorothy-Quincy/generic_smf (ibm.com)

Dorothy-Quincy/smf_chinit_task_interpretation (ibm.com)

Thank you!

Dorothy Quincy
Dorothy.quincy@ibm.com