# MQPERF1

## Intro

**Lyn Elkins –**
**elkinsc@us.ibm.com**

# Legal Disclaimer

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

# YOUR MILAGE WILL VARY

# Agenda – Day 1

| Day 1 | Time |
|---|---|
| Workshop intro and an overview of MQ for z/OS internals | 9:00 - 10:00 |
| Break | 10:00-10:15 |
| An introduction to MQ SMF | 10:15-11:15 |
| Lunch and Lab - Running MP1B | 11:15-12:45 |
| MQ SMF Statistics - The big three in detail | 12:45-1:30 |
| Break | 1:30-1:45 |
| MQ SMF Statistics - New Statistics (V8 & V9) | 1:45 - 2:15 |
| Break | 2:15-2:30 |
| MQ SMF - Introduction to Task Records | 2:30 - 3:15 |
| Break | 3:15-3:30 |
| Special session - MQ Monitoring for distribued (Mostly), with new z/OS features | 3:30-4:30 |

# Agenda – Day 2

| Day 2 | |
|---|---|
| Review, Q&A | 9:00 - 9:15 |
| What's New in MQ V9.X? | 9:15 - 10:15 |
| MQ SMF Accounting - Channel accounting | 10:15 - 10:4 |
| Break | 10:45-11:00 |
| MQSMFCSV - A different way to process | 11:00- 12:0( |
| Lunch & Lab - Downloading and processing MQSMFCSV | 12:15-1:30 |
| MQSMFCSV - Queries | 1:30 - 2:30 |
| Labs - The rest of the day | 2:30 - 4:30 |
| Lab - Creating And Loading Db2 Databases | |
| Lab - Runinf Basic Report Queries | |
| Lab - Running Other Queries | |
| Lab - Replay Messages from MQ logs | |

Additional disclaimer
This workshop is being done under the ‘continuous delivery model.’
You will find bugs, errors, typos
Please let me know
Please also let me know where we need more samples!

# Administration

You know the routine:

Phones on vibrate, please
Know where the exits are
Tray tables in upright and locked position

# Why are you here?

Please introduce yourself, give a brief description on what you want to get out of this session and tell us what your experience is with MQ for z/OS performance, tuning, MQ SMF data evaluation, and problem determination.

# A bit of background

To understand performance tuning, and in some measure problems determination – a bit about the internals of the queue manager is helpful.  So we are going to start there!

# Agenda – QMGR Internals Overview

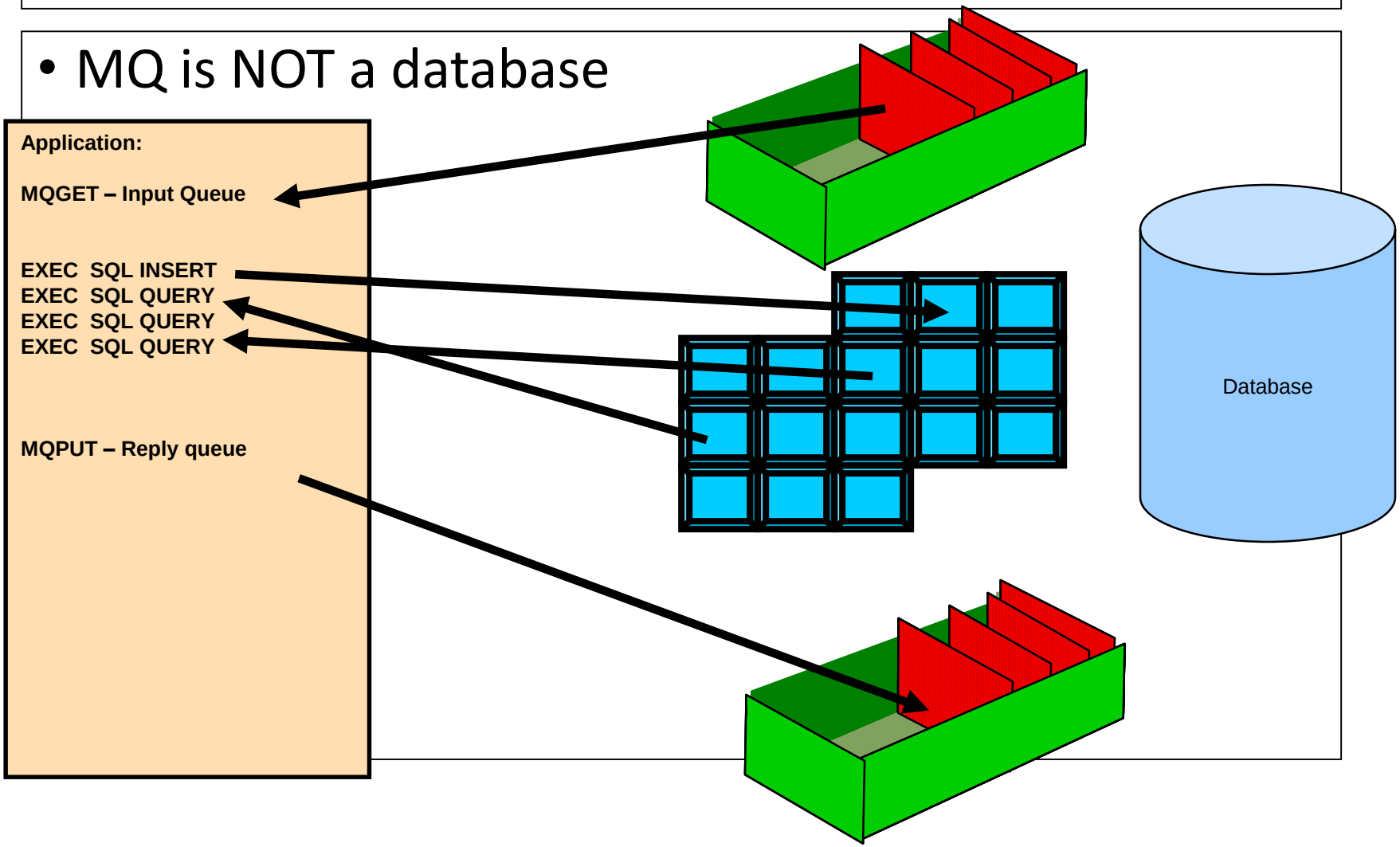
- Why is this important to me?
- One of these things is not like the other
- How are messages stored?
  - Private Queues
  - Shared Queues
- First line managers - the components of a z/OS queue manager
- What happens on a API call?
- Summary

# Why is this topic important to me?

- Queue manager performance
  - Knowing how the pieces fit together
- Application performance
  - If the queue manager is not tuned, responsiveness can be affected
- Problem resolution

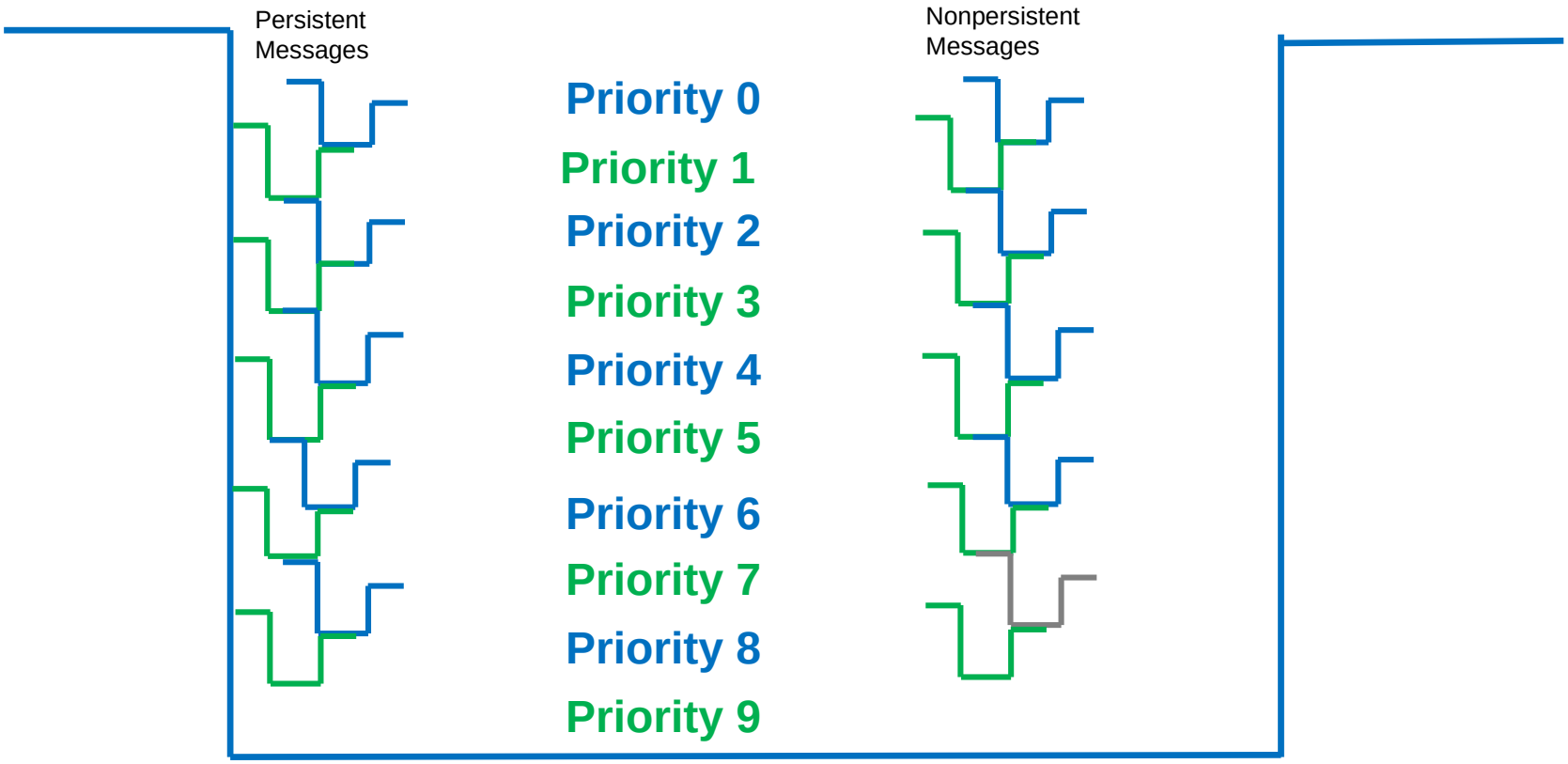
# One of these things is not like the other

- MQ is NOT a database

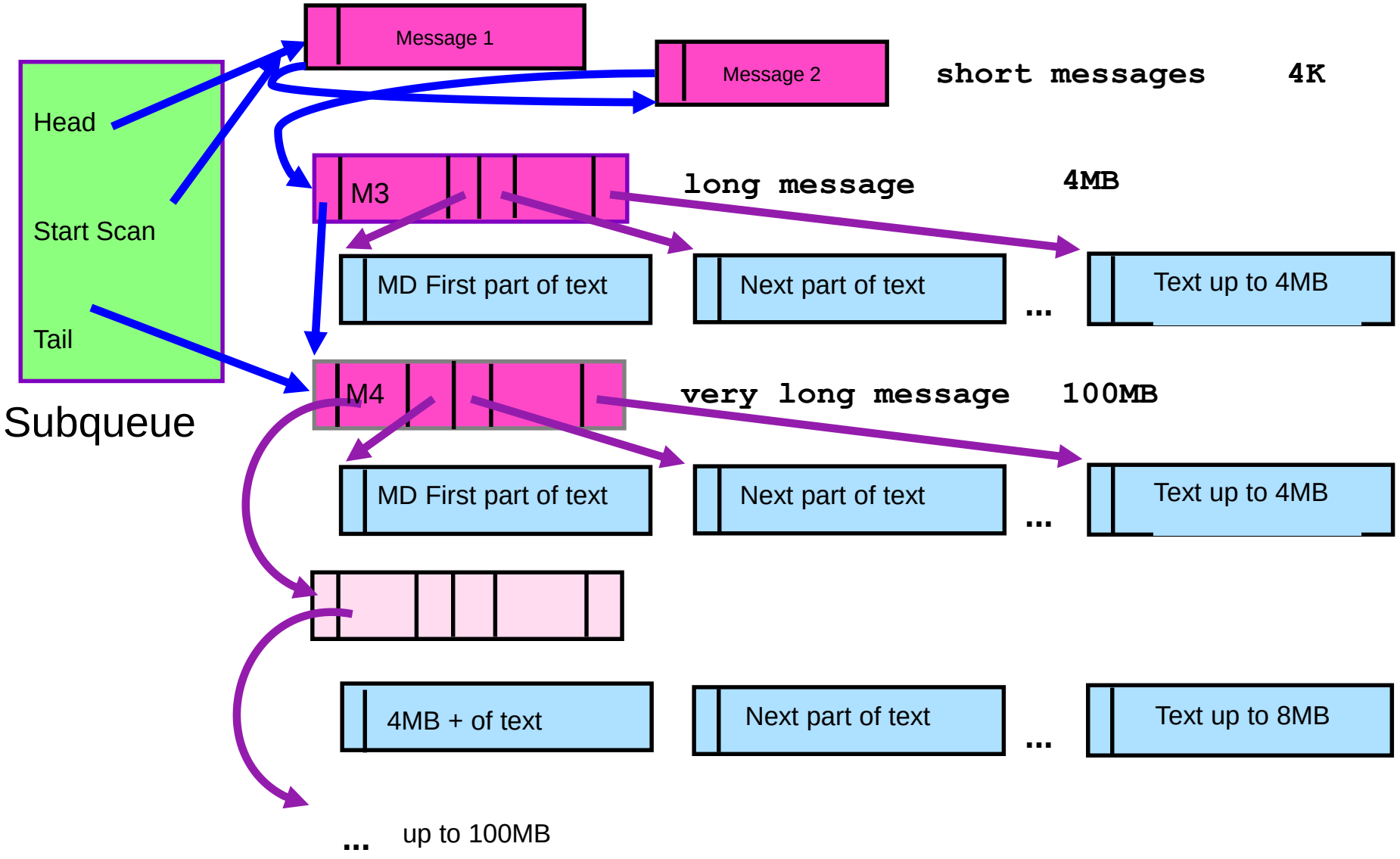


# The Internal Representation of a Queue

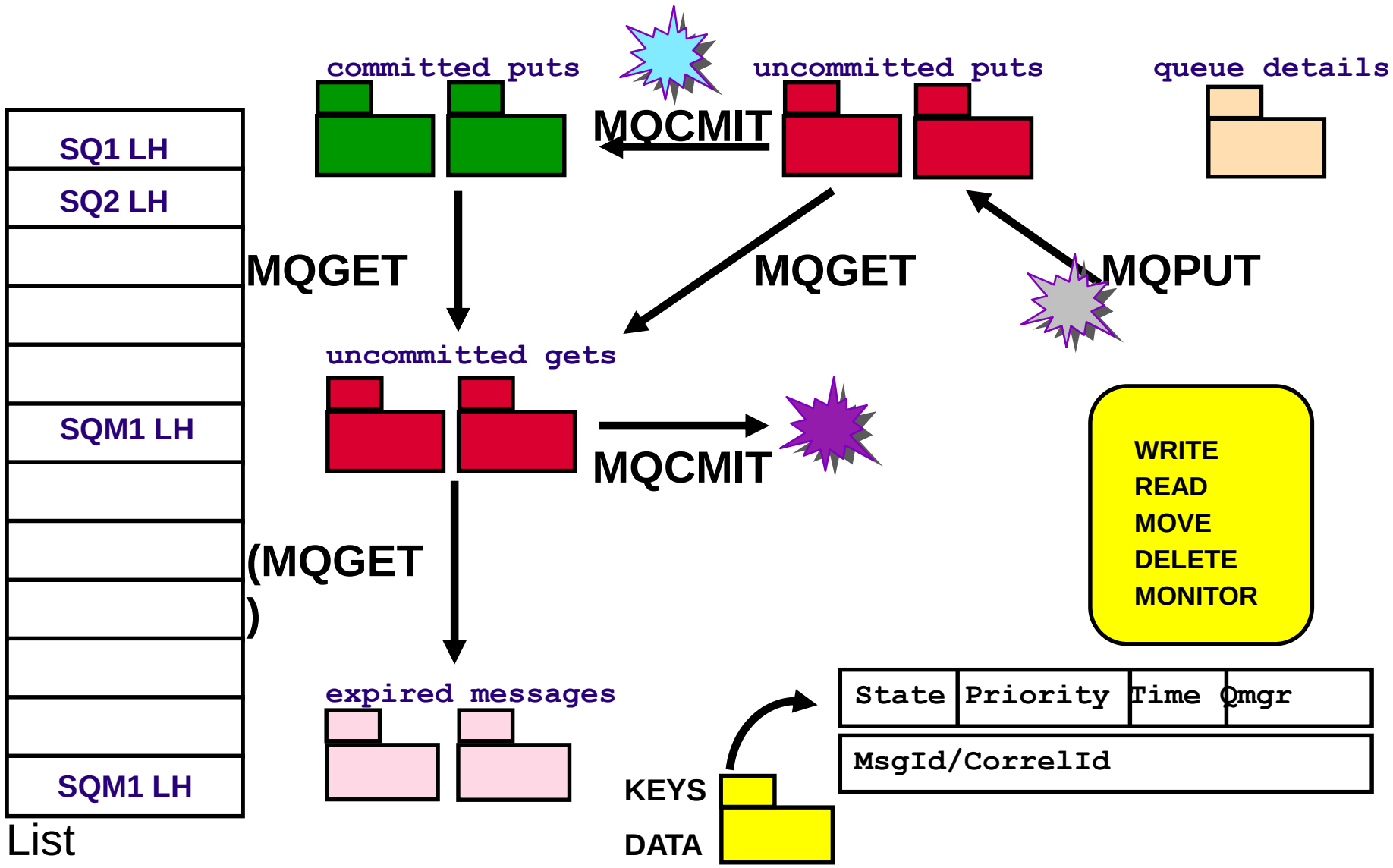
- Sub-queues within a queue



# Private Queue Message Storage

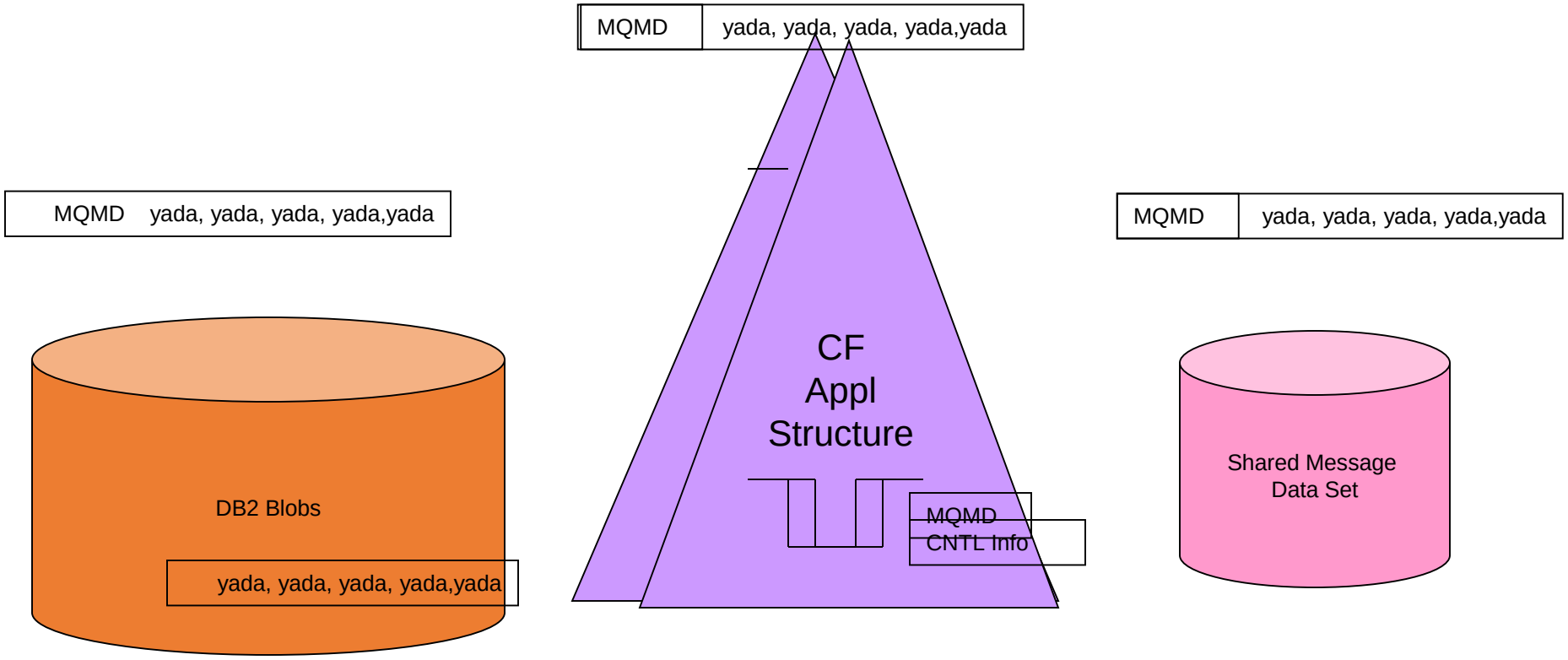


# Shared Message Queue Storage Using CF List Structures



# Shared Queue Message Storage

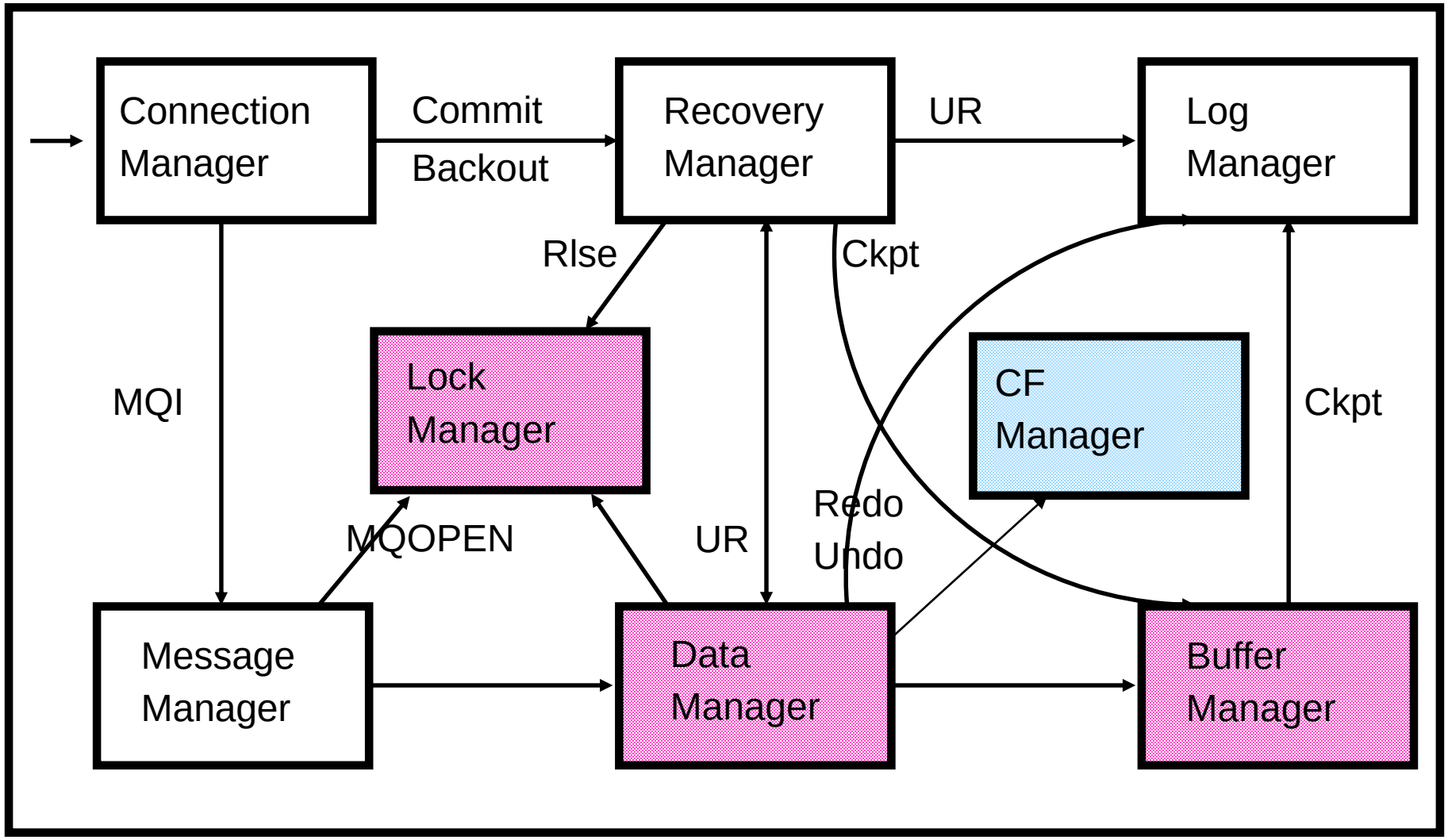
- Messages are stored in one of several ways:
  - Entirely within the list structure
  - Control information (CI) on list structure, message body in DB2
  - CI on list structure, message on Shared Message Data Sets
  - CI and/or message moved to Flash Memory (not shown)



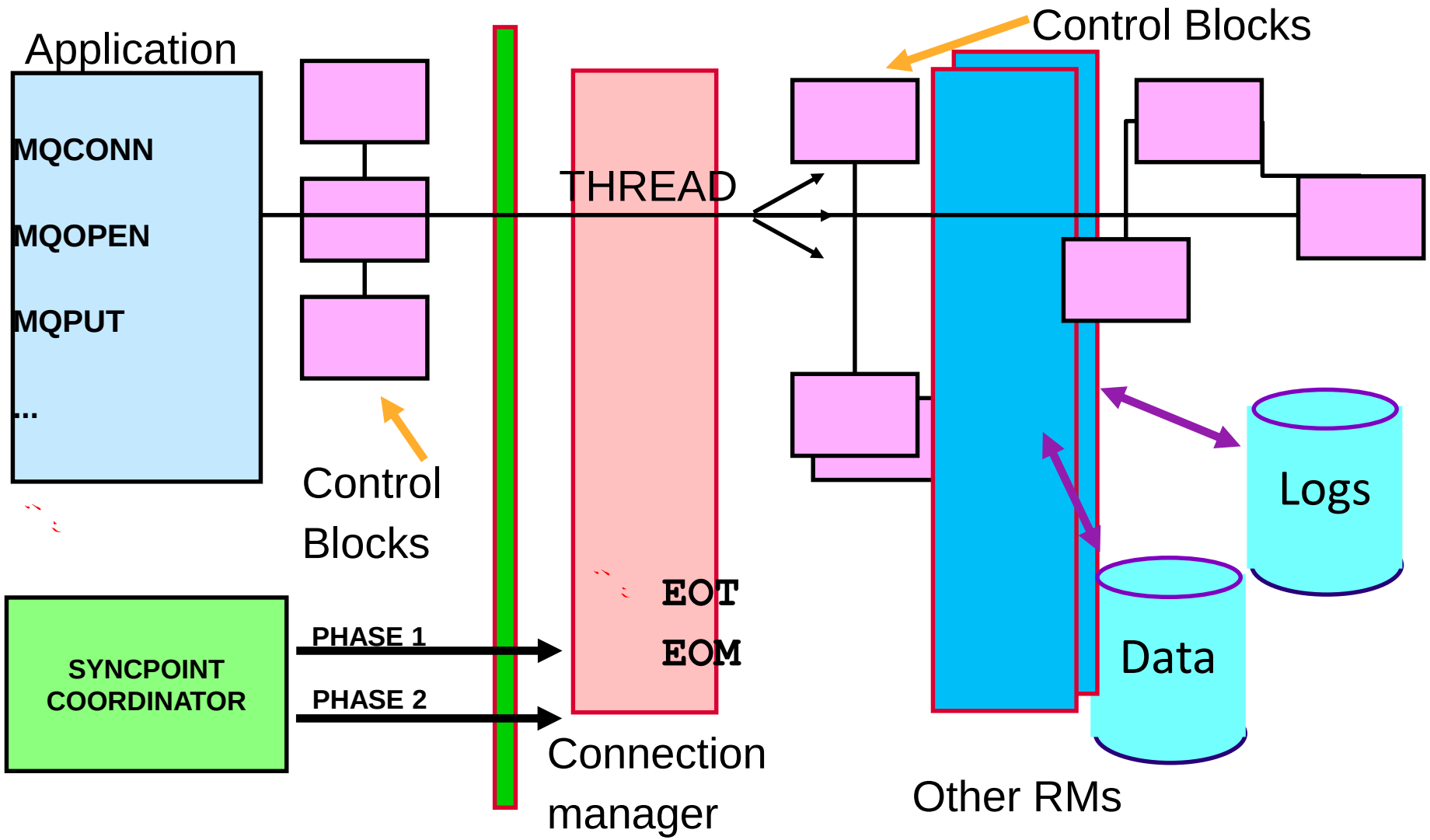
# First Line Managers – who does the real work

- To provide the qualities of service that are the basis for WMQ, the real work within the queue manager is divided into logical ‘workers’ or managers. They interact with the 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……

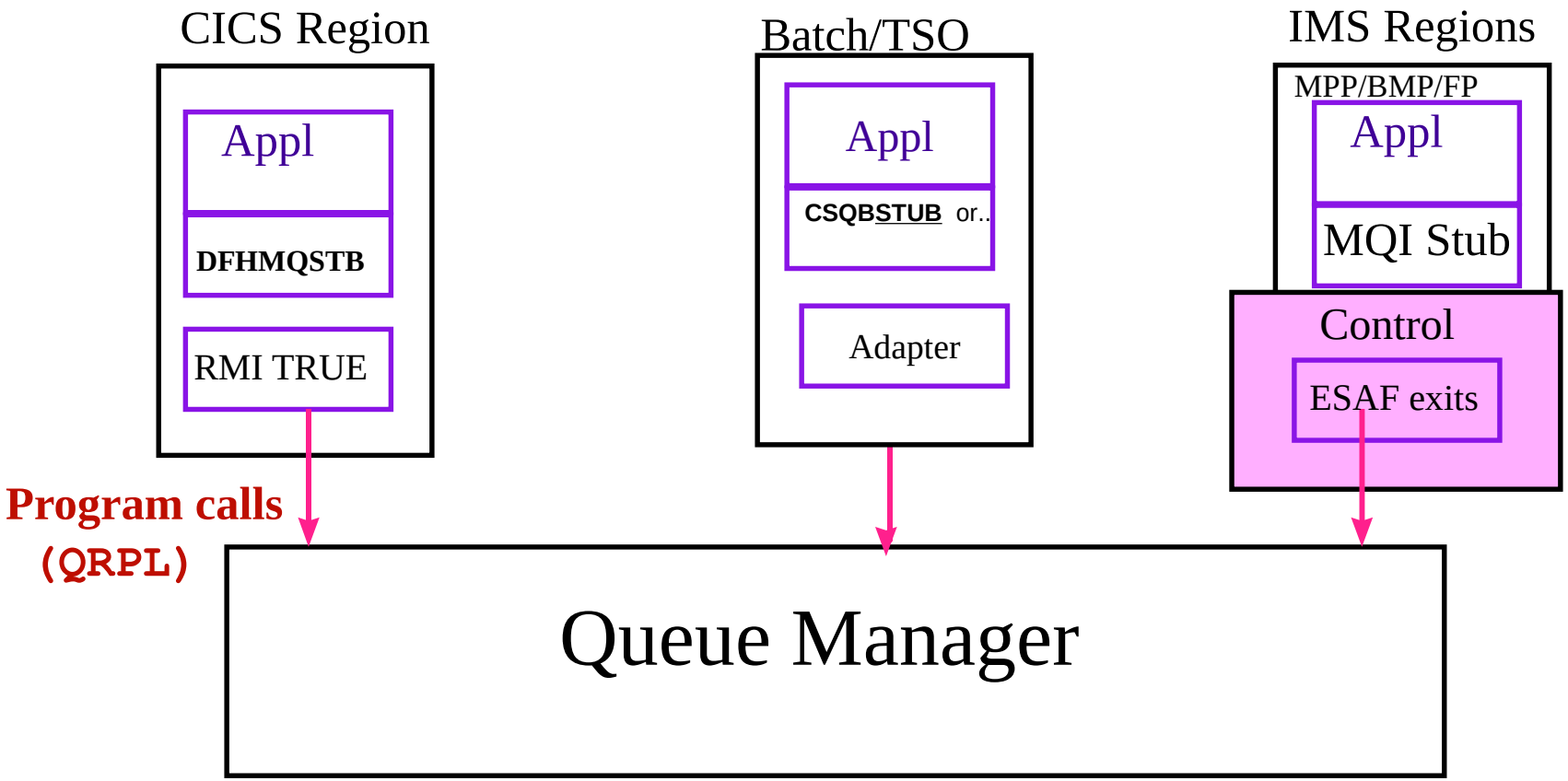
## Building Blocks - Resource Managers



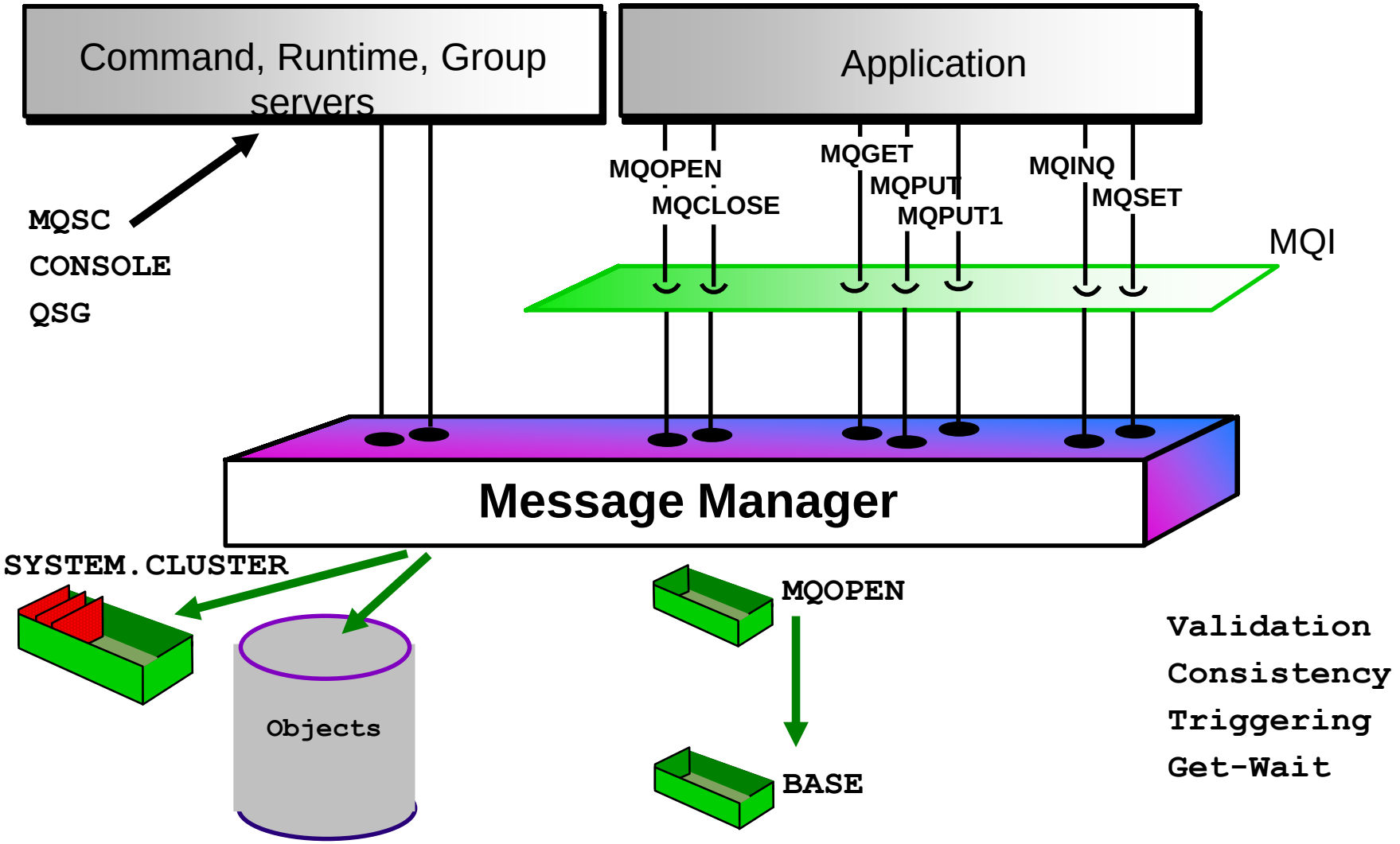
# Handling Applications - Connection Manager



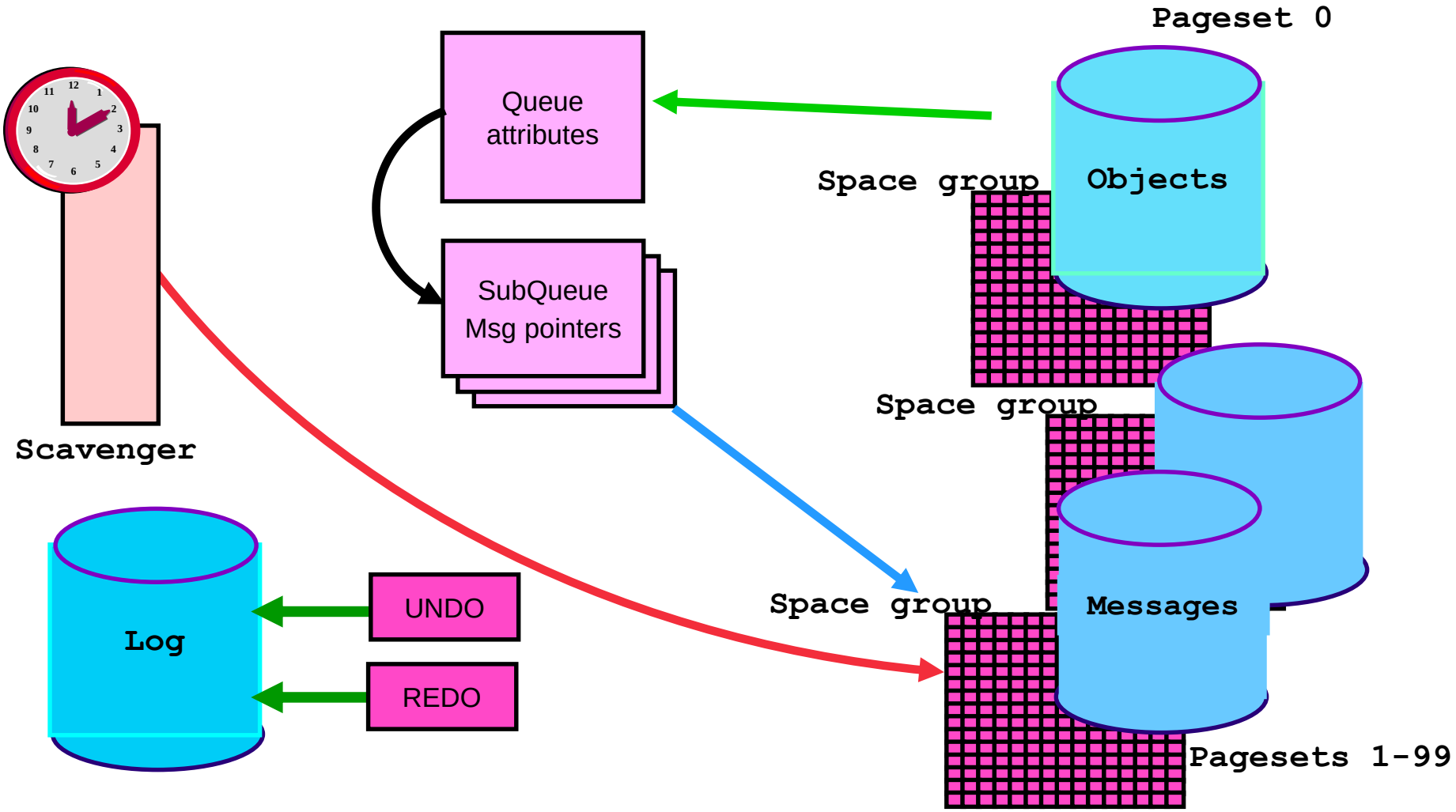
# Getting requests into WMQ - Stubs and Adapters



# Controlling the MQI and MQSC - Message Manager



# Controlling Messages and Objects - Data Manager



# Buffer Manager – High Performance storage and retrieval

read pageset

DEFINE BUFFPOOL(bpid)
BUFFERS(nnnn)
STEAL
LRU

buffer pages

pageset

write pageset

dirty

CKPT,
STEAL,
WRITE AHEAD

DEFINE PSID(psid)
BUFFPOOL(bpid)

NO FORCE

dirty

dirty

OLDEST

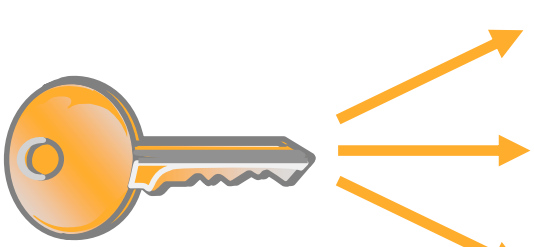

checkpoint

FORCE DIRTY PAGES > 3 CKPTS

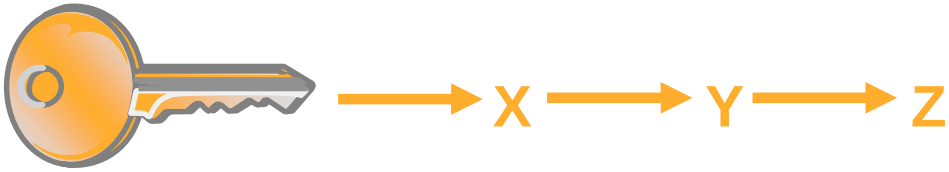
# Providing Logging Interfaces - Log Manager

- Log read and write functions
- Log Shunting
- Multiple active log data sets and archive
- Archive inventory management
- Duplexed for reliability
- “Bootstrap” file
  - End of log location
  - Archive inventory
- Various Utilities

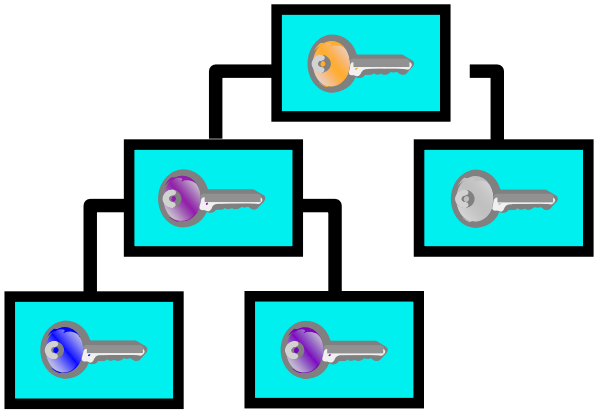
# Concurrency and Isolation - Lock Manager

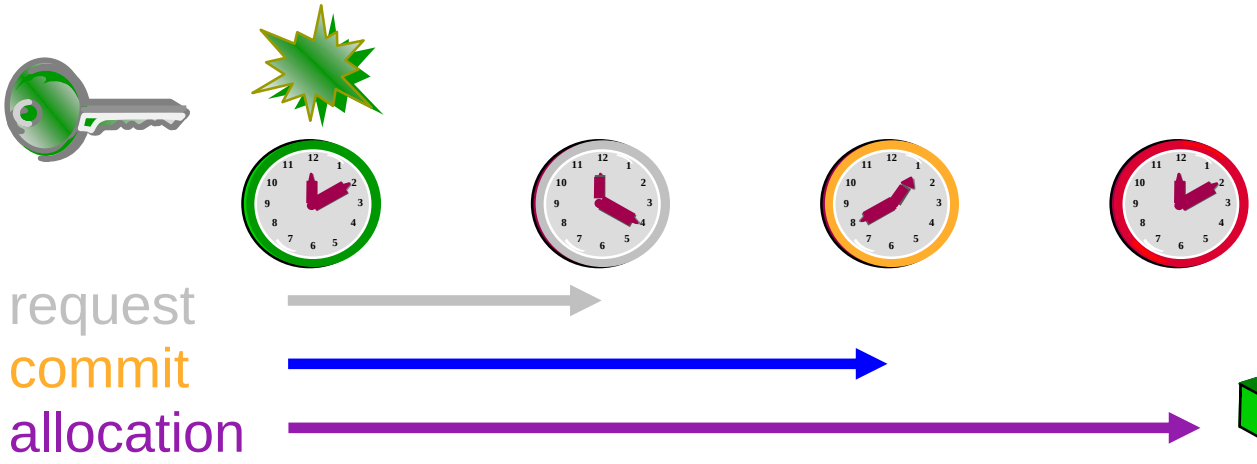



MQOO_INPUT_SHARED




MQOO_INPUT_EXCLUSIVE




MQOPEN

vs

DELETE QLOCAL

# Scenario – Persistent MQPut to a Triggered Queue

| Application | Message Manager | Data Manager | Buffer Manager | Recovery Manager | Log Manager | Lock Manager |
|---|---|---|---|---|---|---|
| MQOPEN | | | | | | |
| | → | | | | | ACQUIRE LOCK |
| | | LOCATE QUEUE IN HASH TABLE | | | | |
| | SECURITY | | | | | |
| | BASE NAME | | | | | |
| | ACQUIRE HANDLE | | | | | |
| MQPUT | | | | | | |
| | USE HANDLE | | | | | |
| | | LOCATE PAGE TO HOLD MSG | | | | |
| | | | BUFFER PAGE | | | |
| | | → | | START UR | LOG RECORDS | |
| | | → | | | LOG RECORDS | |
| | CHECK TRIGGER RULES | | | | | |
| MQCMIT | | | | | | |
| | | | | → | FORCE LOG | |
| | | | | | | RELEASE LOCKS |

# Scenario - MQGet from a Queue

| Application | Message Manager | Data Manager | Buffer Manager | Recovery Manager | Log Manager | Lock Manager |
|---|---|---|---|---|---|---|
| MQOPEN | | | | | | |
| | | | | | | ACQUIRE LOCK |
| | | LOCATE QUEUE IN HASH TABLE | | | | |
| | SECURITY | | | | | |
| | BASE NAME | | | | | |
| | ACQUIRE HANDLE | | | | | |
| MQGET | | | | | | |
| | USE HANDLE | | | | | |
| | | FIND MSG (INDEX / NEXT) | | | | |
| | | | BUFFER PAGE | | | |
| | | | | START UR | LOG RECORDS | |
| | | | | | LOG RECORDS | |
| MQCMIT | | | | | | |
| | | | | | FORCE LOG | |
| | | | | | | RELEASE LOCKS |

# Summary

- Delivers transactional messaging
  - Enables robust business applications
- Complex, but well organized
  - Adapters, Address spaces, Resource Managers
- Designed for throughput, availability and scalability
  - Logging, Buffering, Locking, Communications