

# MQ for z/OS: Better Performance from knowing a bit about the internals

## Session 28144



Lyn Elkins – WSC – [elkinsc@us.ibm.com](mailto:elkinsc@us.ibm.com)  
Mitch Johnson – WSC – [mitchj@us.ibm.com](mailto:mitchj@us.ibm.com)



# Agenda – MQ for z/OS Internals Overview

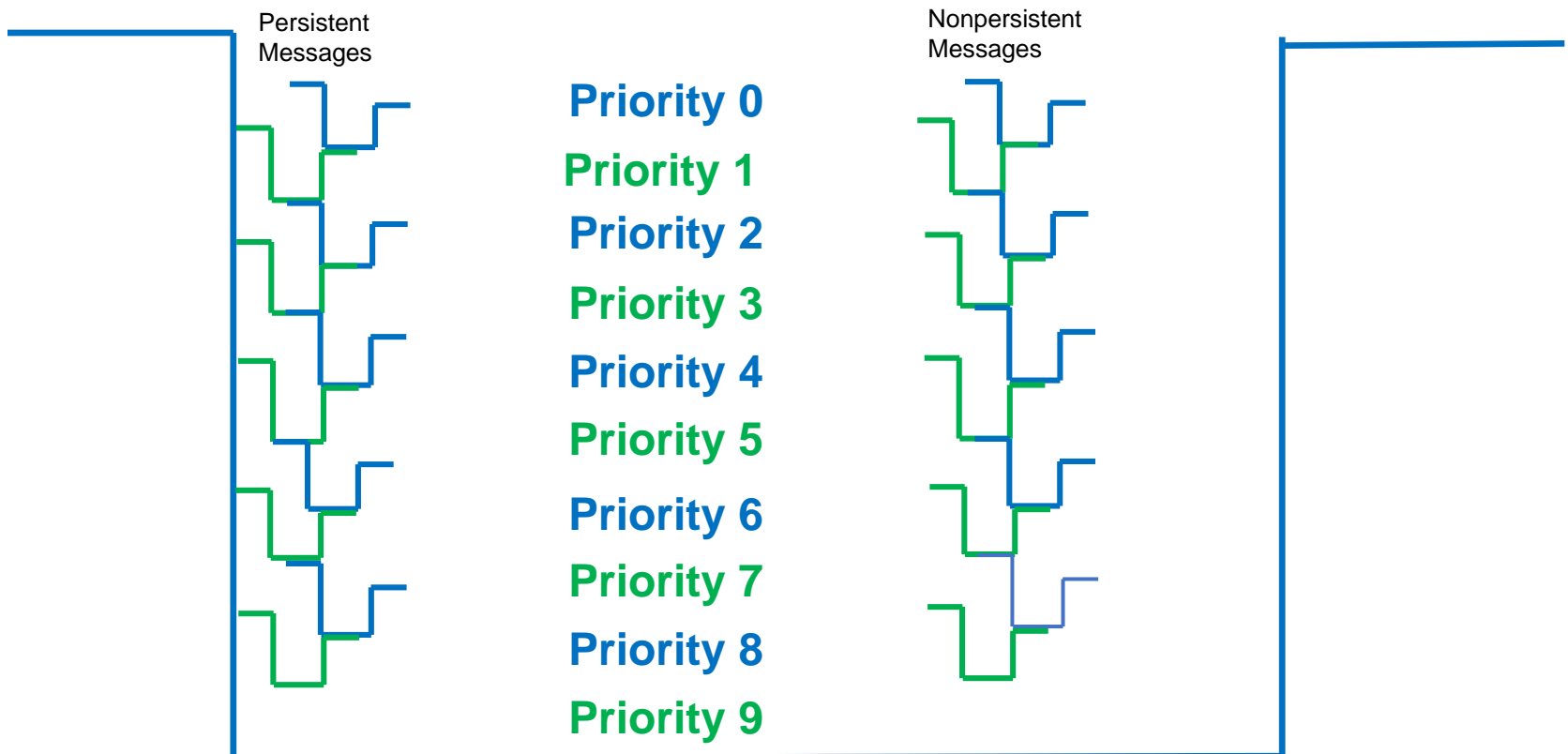
- Why is this important to me?
- Physical Storage - How are messages stored?
  - Private Queues
  - Shared Queues
- First line managers - the components of a z/OS queue manager
- Simple flow of an MQPUT
- Some CHIN Tasks
- Summary

# Why is this topic important to me?

- Queue manager performance
  - Knowing how the messages are stored and the internal tasks fit together
- Application performance
  - If the queue manager is not tuned, responsiveness can be affected
- Problem resolution

# The Internal Representation of a Private Queue

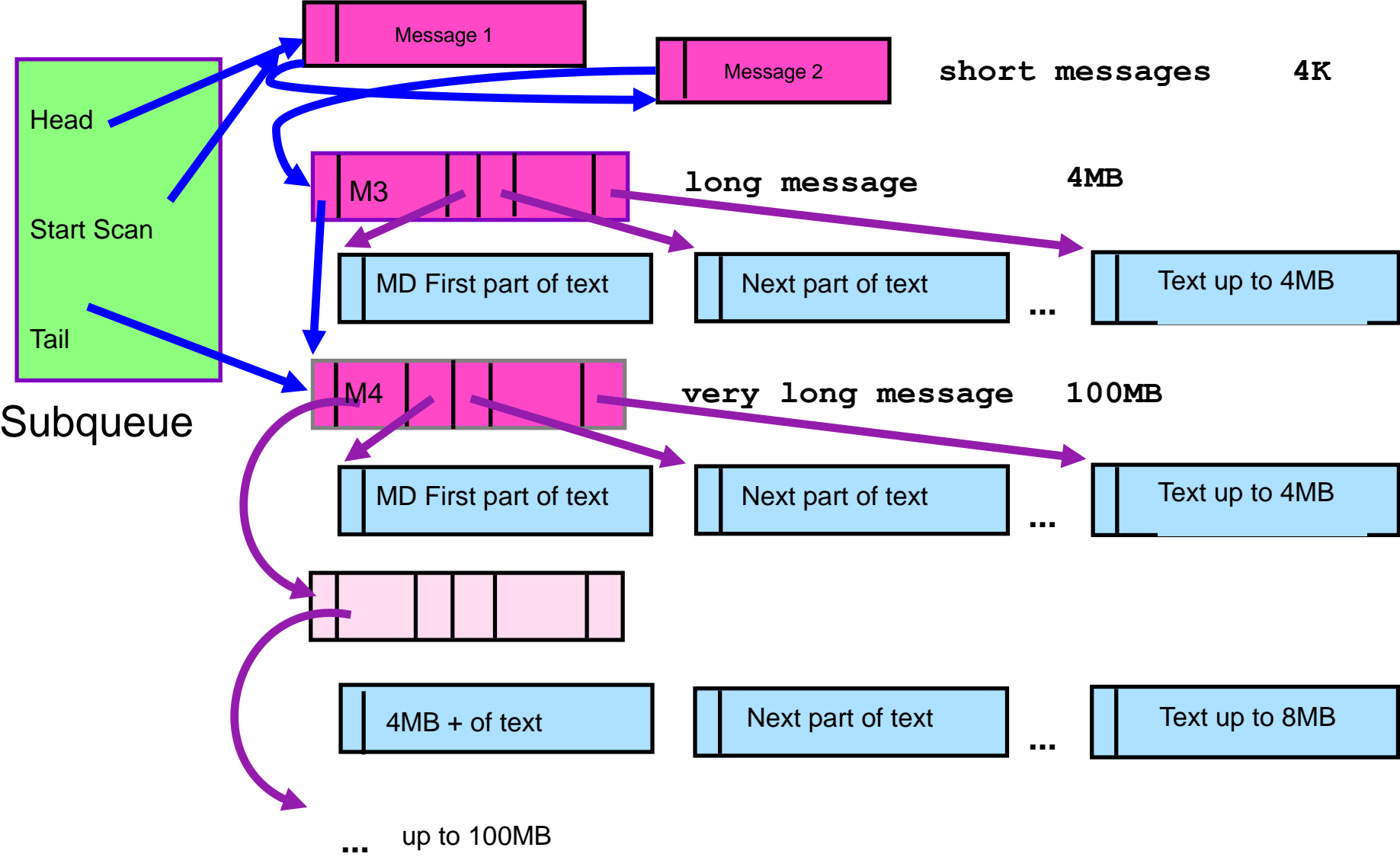
- Sub-queues within a queue



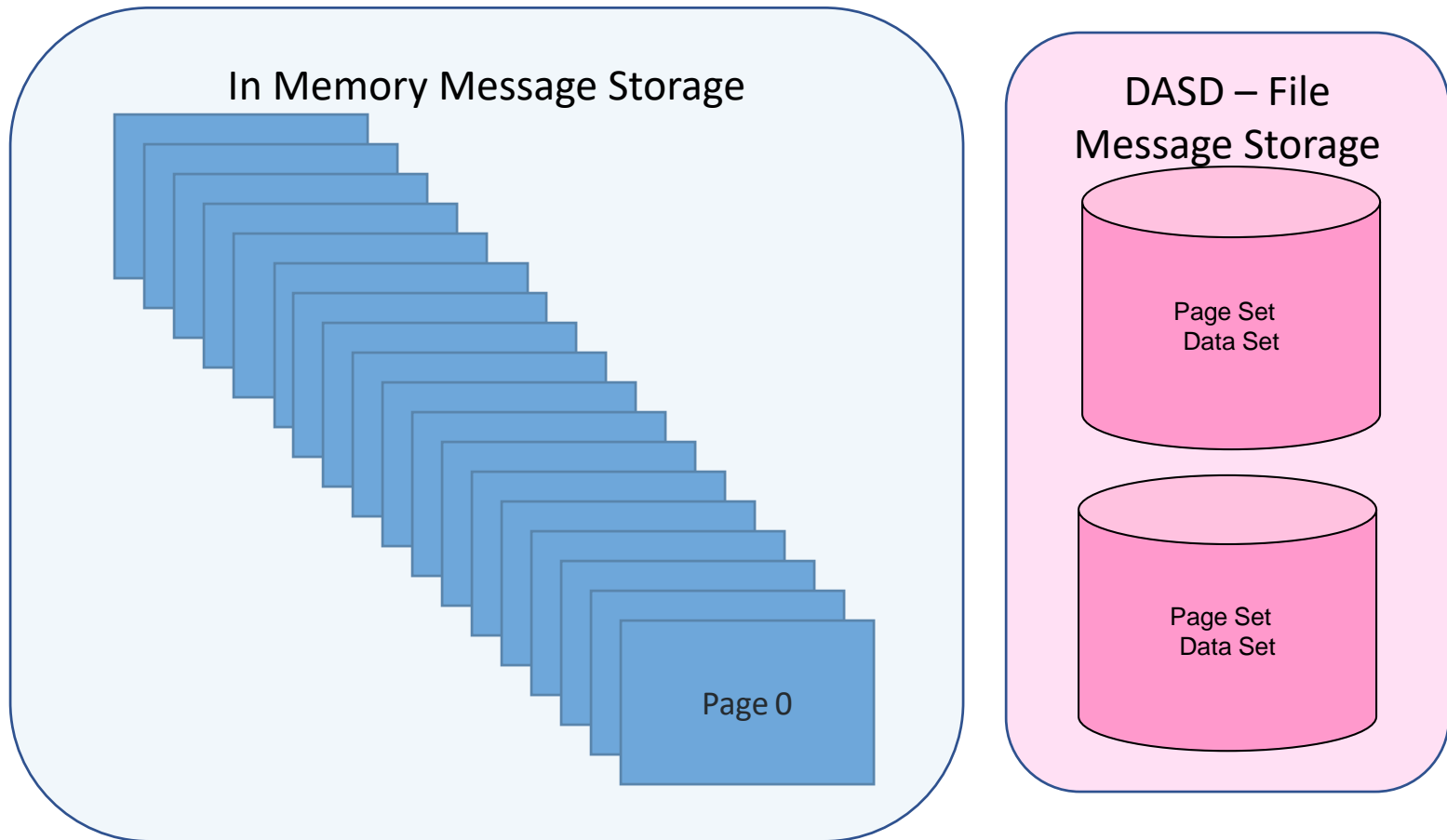
# The Internal Representation of a Queue

- A queue is implemented as 20 subqueues.
  - These subqueues correspond to the persistence and priority of messages. This improves performance of messages of a given priority at put time, though typically all messages on a queue have the same priority.
  - Having separate persistent and non-persistent message pages improves restart time, since non-persistent message pages can be marked as deallocated in the space map at restart.

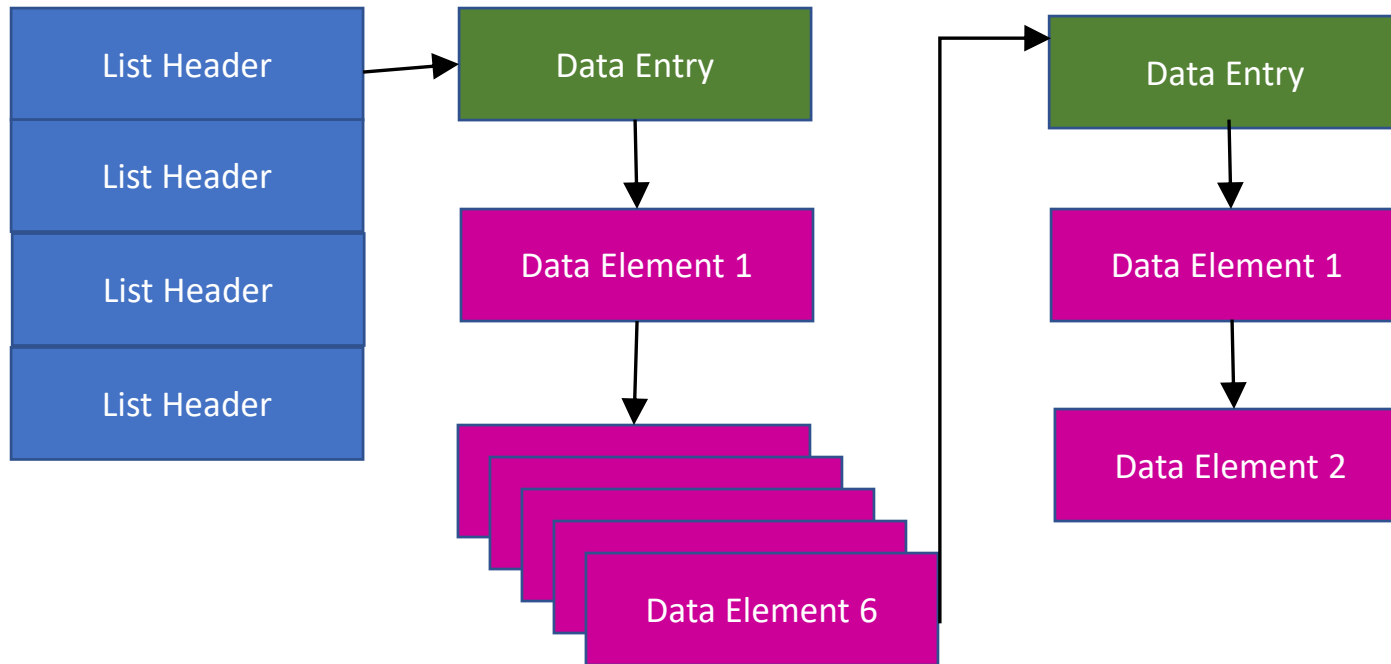
# Private Queue Message Storage



# Private Queues – Buffer pools and Page Sets



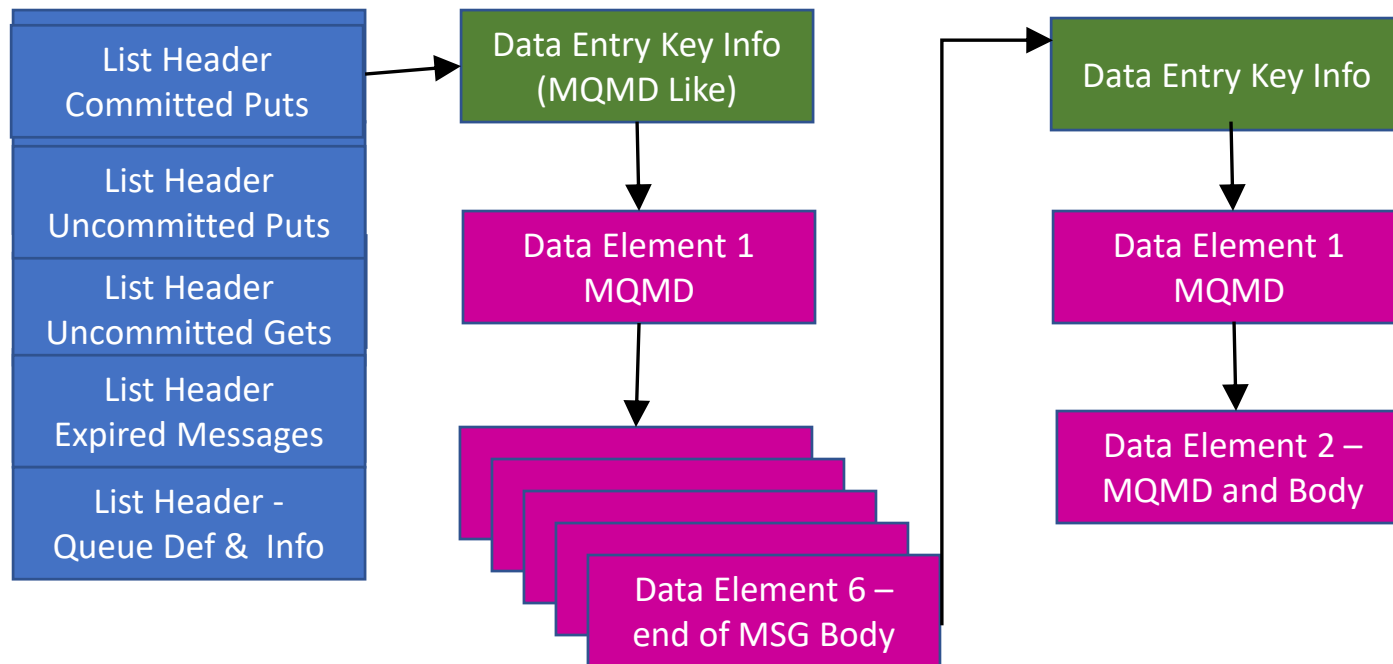
# Coupling Facility List Structure





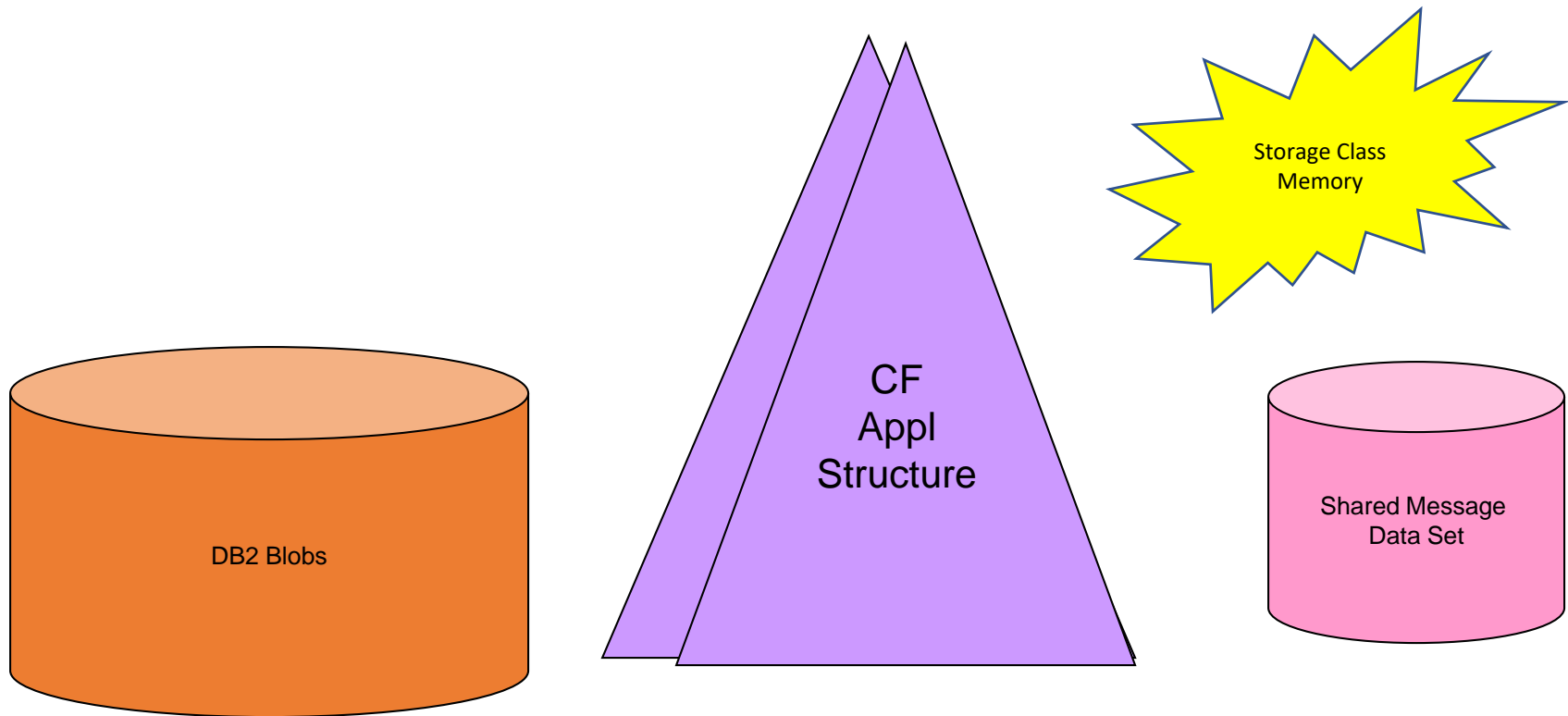
# Coupling Facility List Structure

## What does a queue look like?



# Shared Queue Message Storage

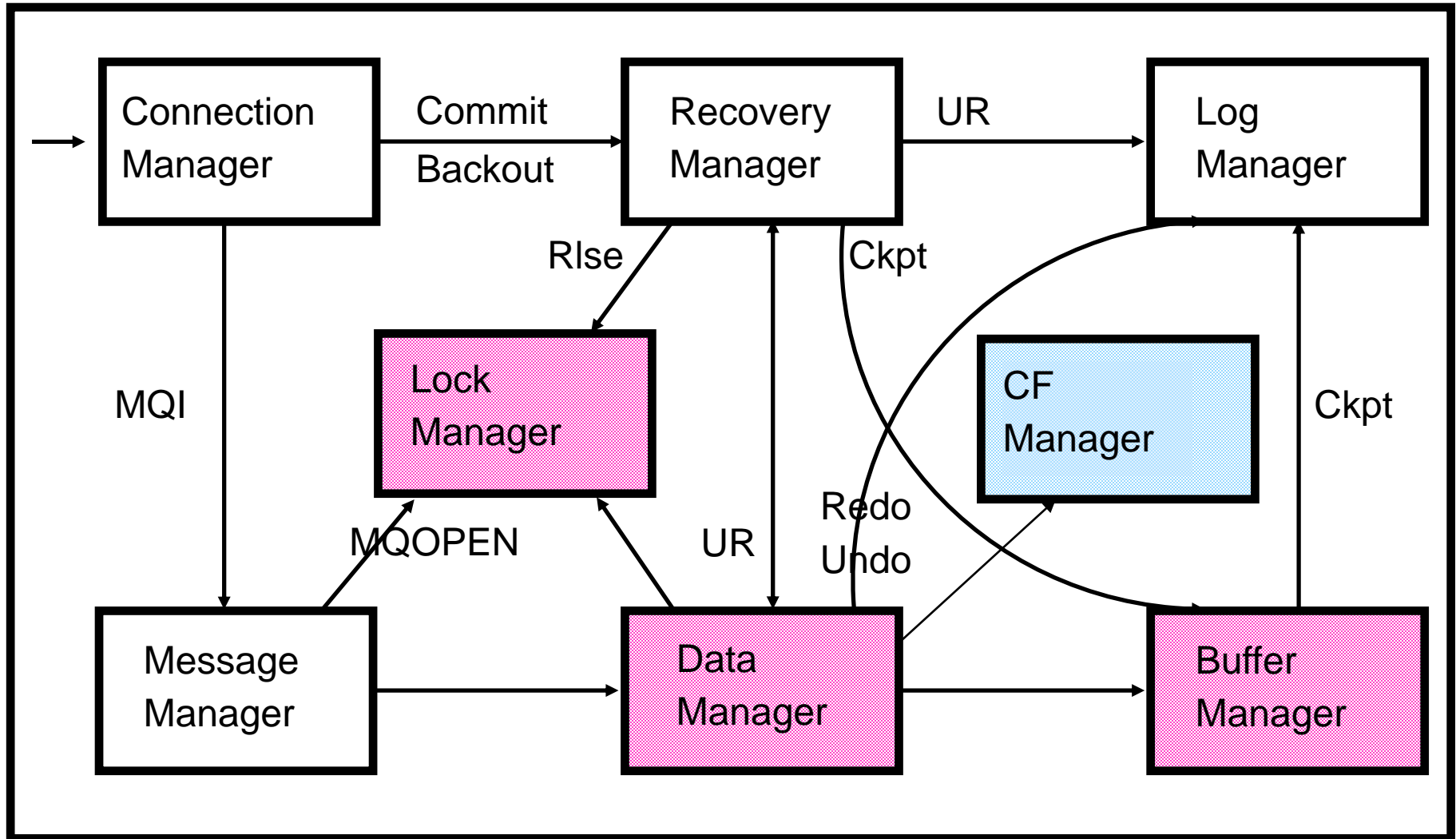
- Messages are stored in one of several ways:
  - Entirely within the list structure
  - Control information (CI) on list structure, message body in DB2
  - CI on list structure, message on Shared Message Data Sets
  - CI or/and message moved to Flash Memory



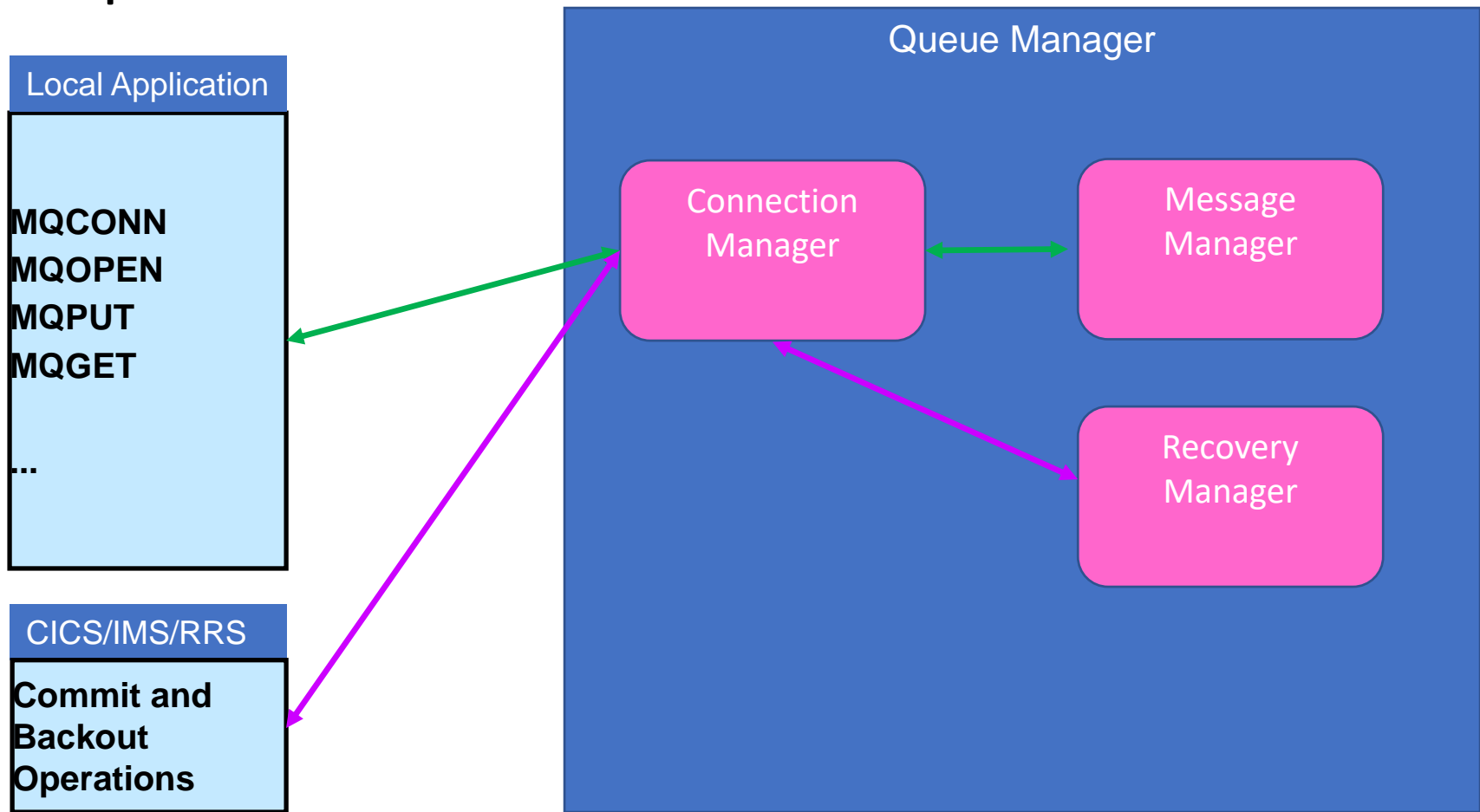
# First Line Managers – who does the real work

- 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.....

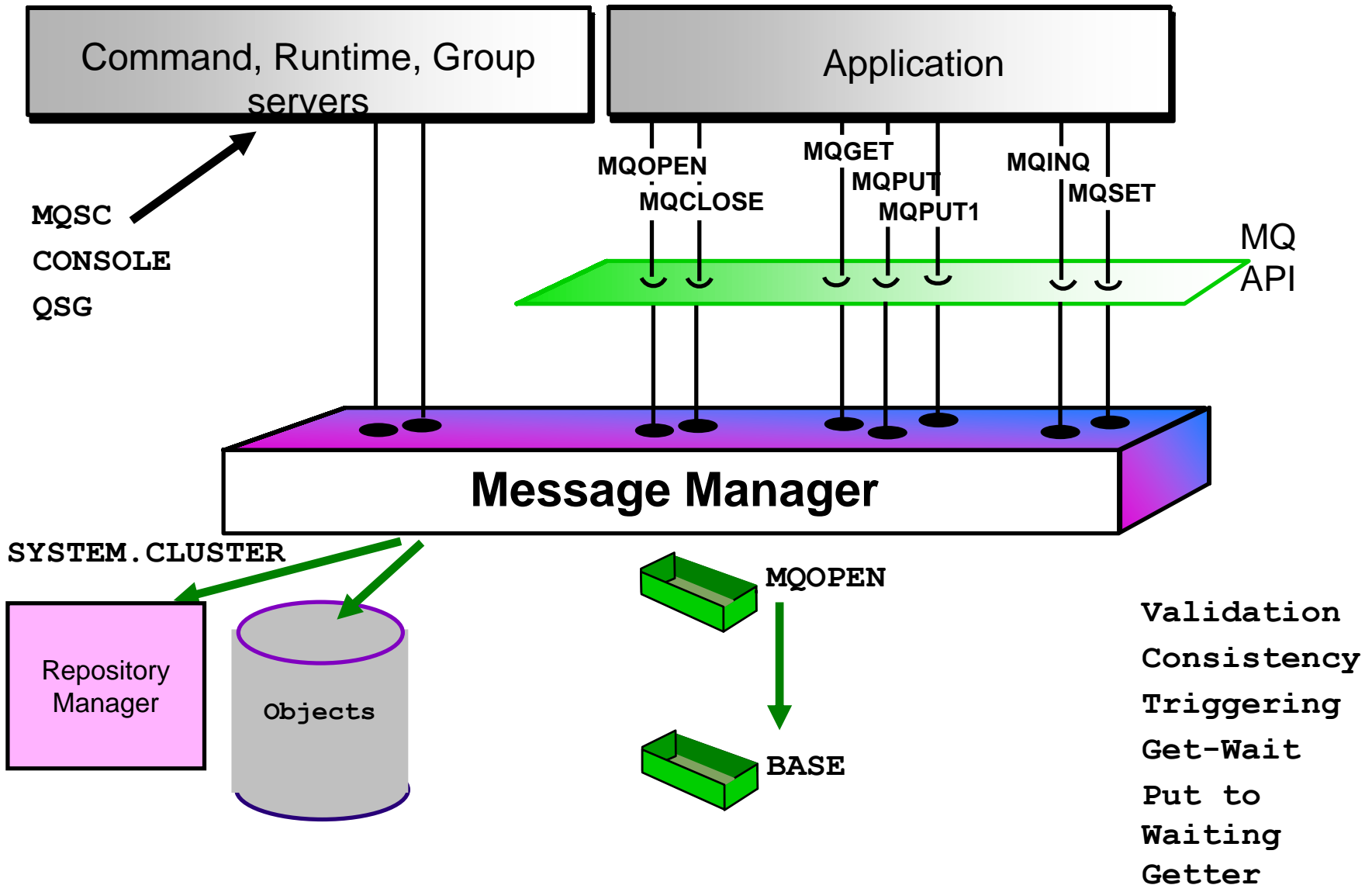
## Building Blocks - Resource Managers



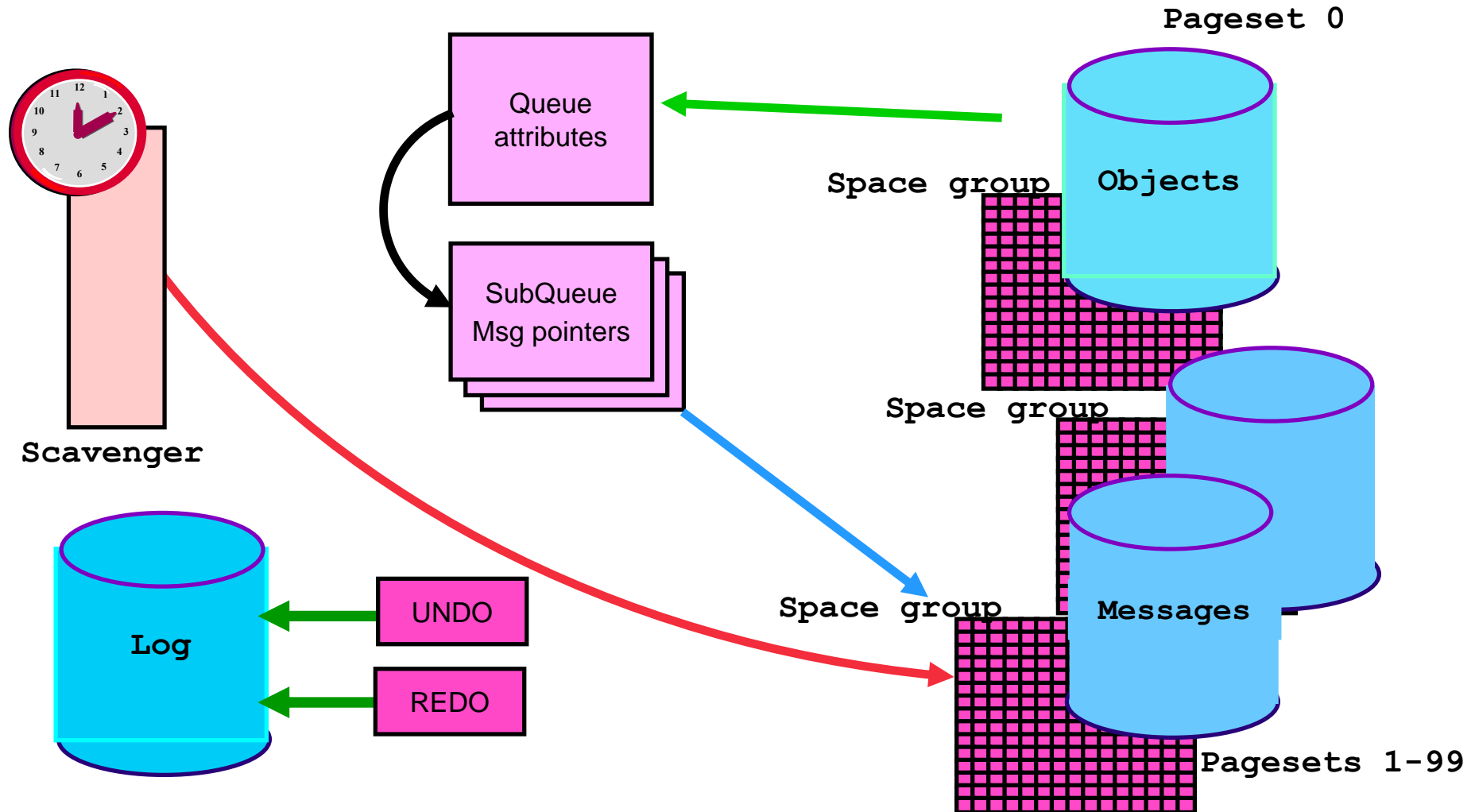
# Connection Manager – the entry point



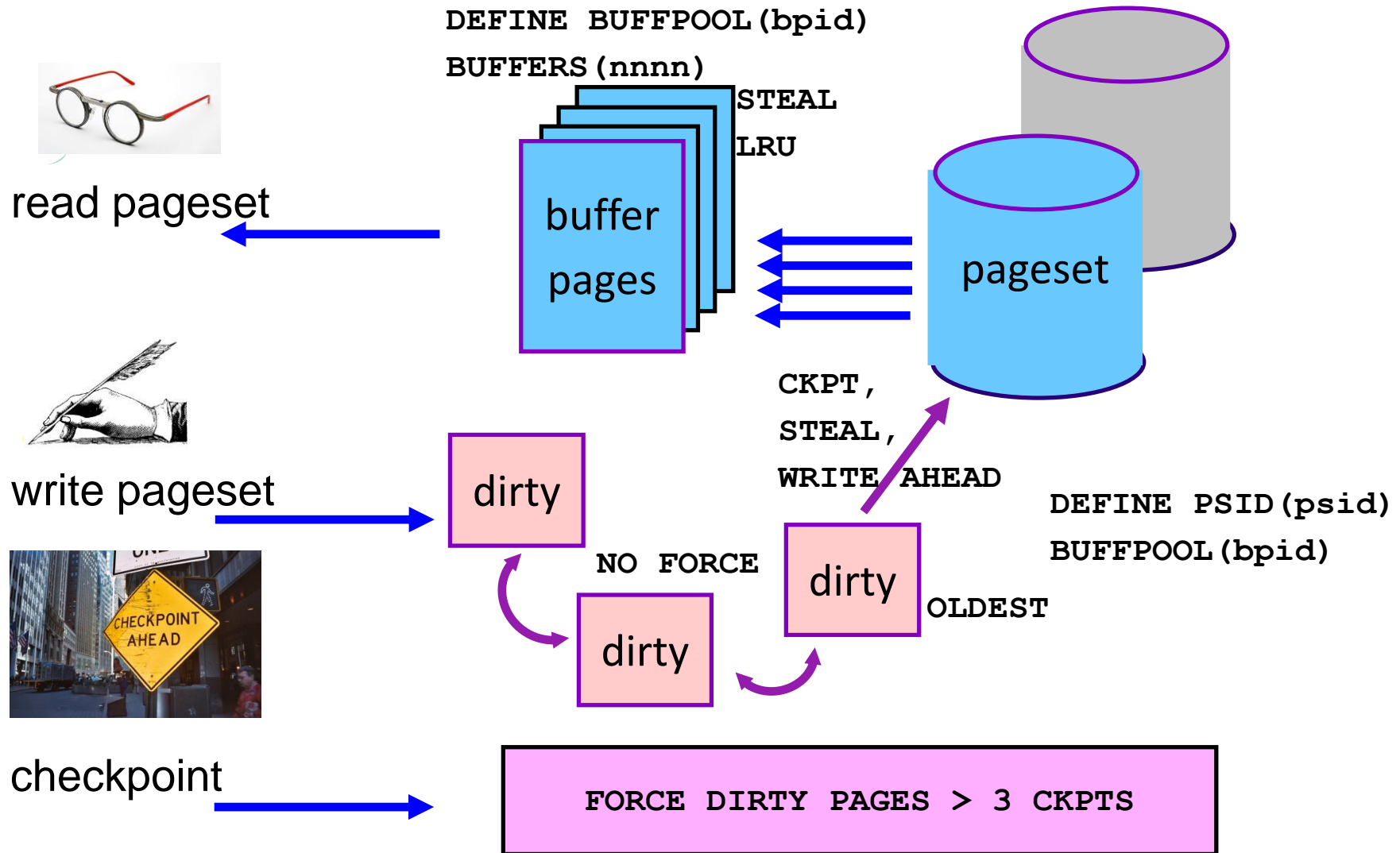
# Controlling the MQI and MQSC - Message Manager



# Controlling Messages and Objects - Data Manager



# Buffer Manager – High Performance storage and retrieval

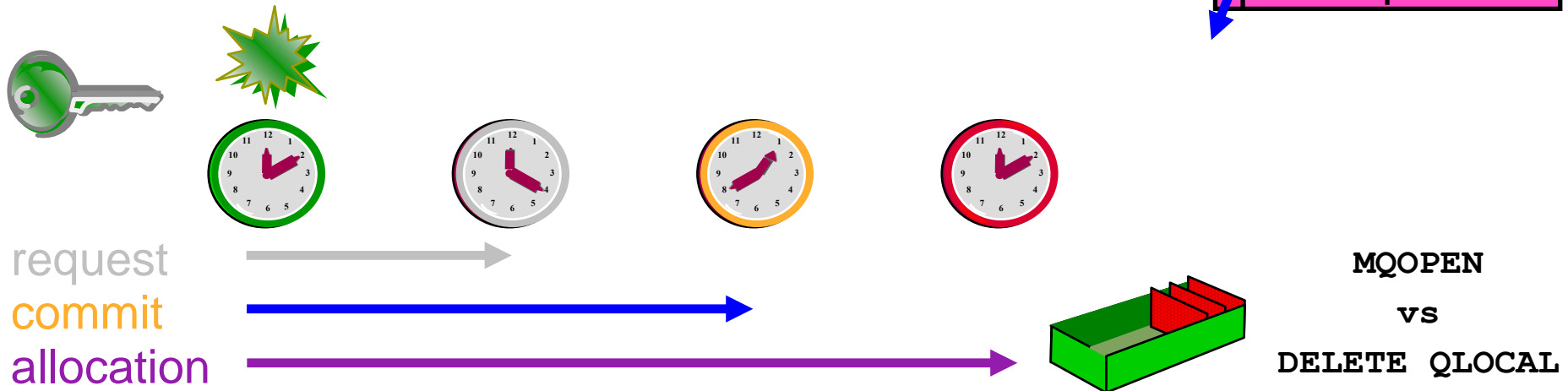
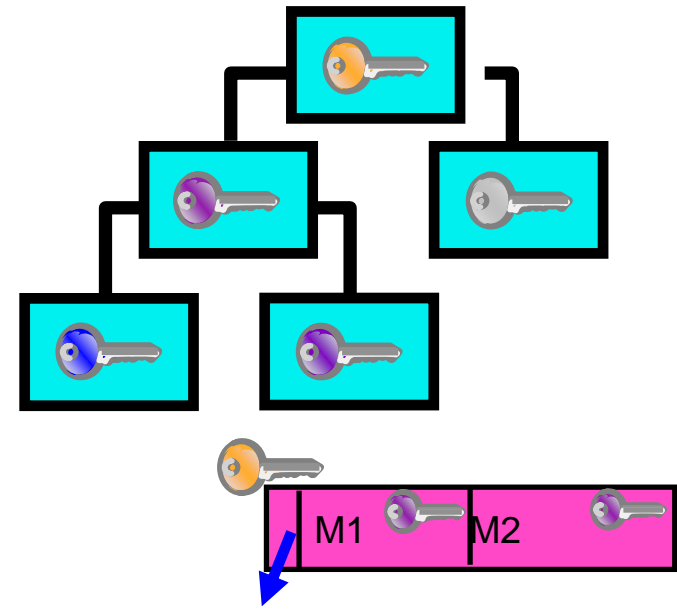
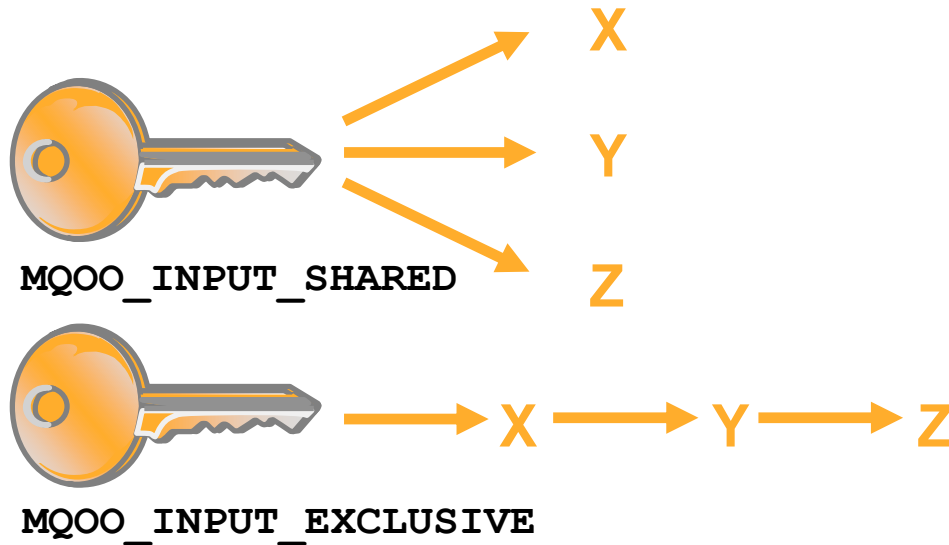








# Providing Logging Interfaces - Log Manager

- 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



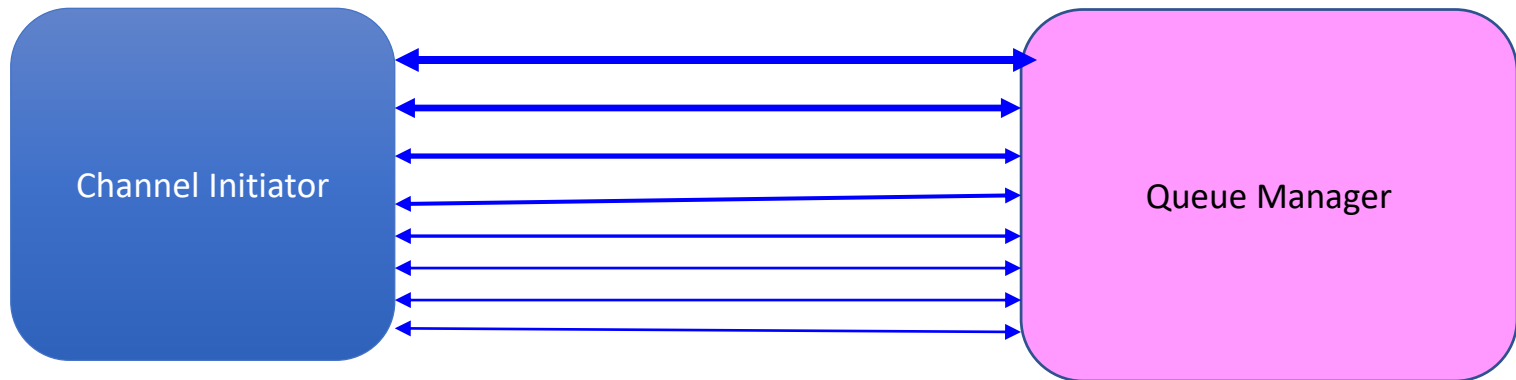
# Scenario – Persistent MQPut to a Triggered Queue

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	
MQCLOSE						RELEASE LOCKS

# A Bit about the CHIN

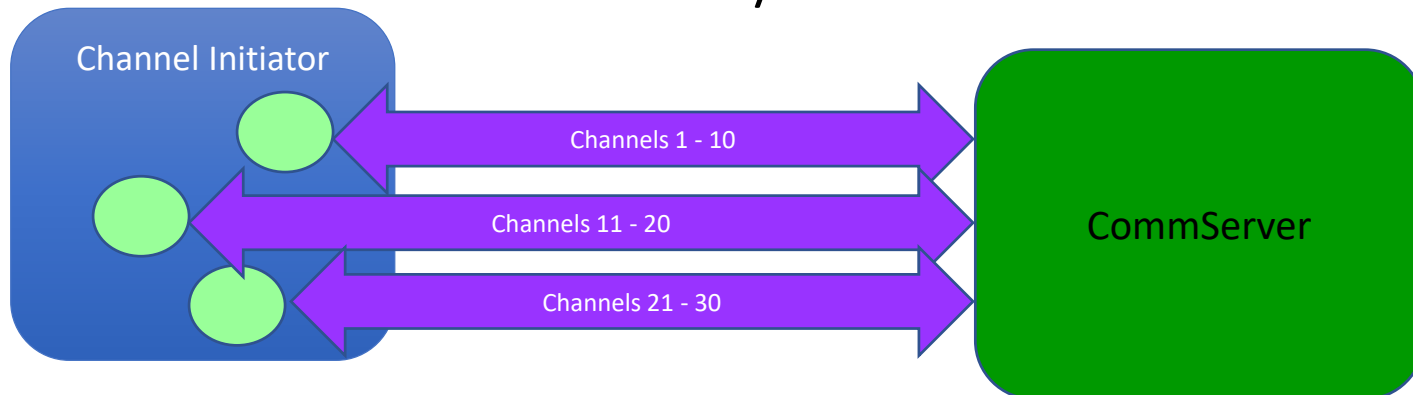
# Channel Initiator – Adapter Tasks

- Adapter tasks
  - Communication between CHIN to MSTR address spaces
  - Number defined by the CHIADAPS attribute
    - Default is 8, recommended for production is 30
  - Used as needed



# Channel Initiator – Dispatcher Tasks

- Dispatcher Tasks
  - Communication between CHIN and network protocol
  - Number defined by the CHIDISPS attribute
    - Default value is 20, recommendation is based on number of active channels\*
  - Tasks are CPU intensive, keep as few as busy as possible
  - Maximum of 100 for TCP/IP



# Channel Initiator – Other Tasks

- SSL Tasks
  - Behave like Dispatcher tasks
    - A channel is associated with a task and will remain there unless restarted
  - The attribute is SSLTASKS
- DNS Task
  - Single task

# Summary

- The physical container for the messages is critical for the overall queue manager performance.
- The log manager and I/O responsiveness is often the next largest area of opportunity for performance improvement.
- The CHIN offers tuning opportunities in tuning the number of tasks and the maximum number of channels.



# MQ for z/OS: Better Performance from knowing a bit about the internals

## Session 28144



Lyn Elkins – WSC – [elkinsc@us.ibm.com](mailto:elkinsc@us.ibm.com)  
Mitch Johnson – WSC – [mitchj@us.ibm.com](mailto:mitchj@us.ibm.com)

