MQ for z/OS: Better Performance from knowing a bit about the internals
Session 27053

Lyn Elkins – WSC – elkinsc@us.ibm.com

Mitch Johnson – WSC – mitchj@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 MILEAGE WILL VARY

Agenda – QMGR Internals Overview

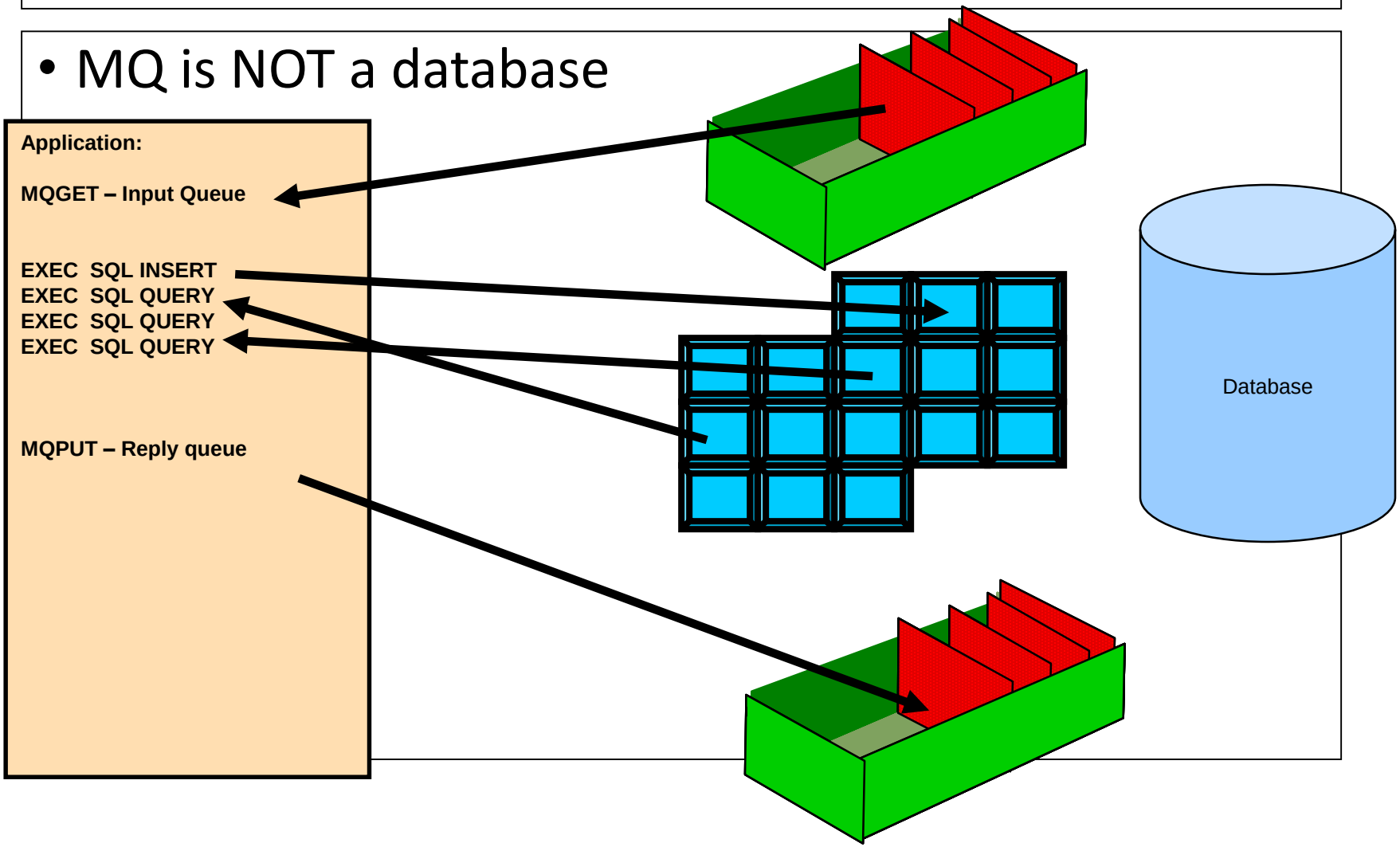
- Why is this important to me?
- One of these things is not like the other
- How are messages stored?
  - Private Queues
  - Shared Queues
- First line managers - the components of a z/OS queue manager
- What happens on a API call?
- Summary

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

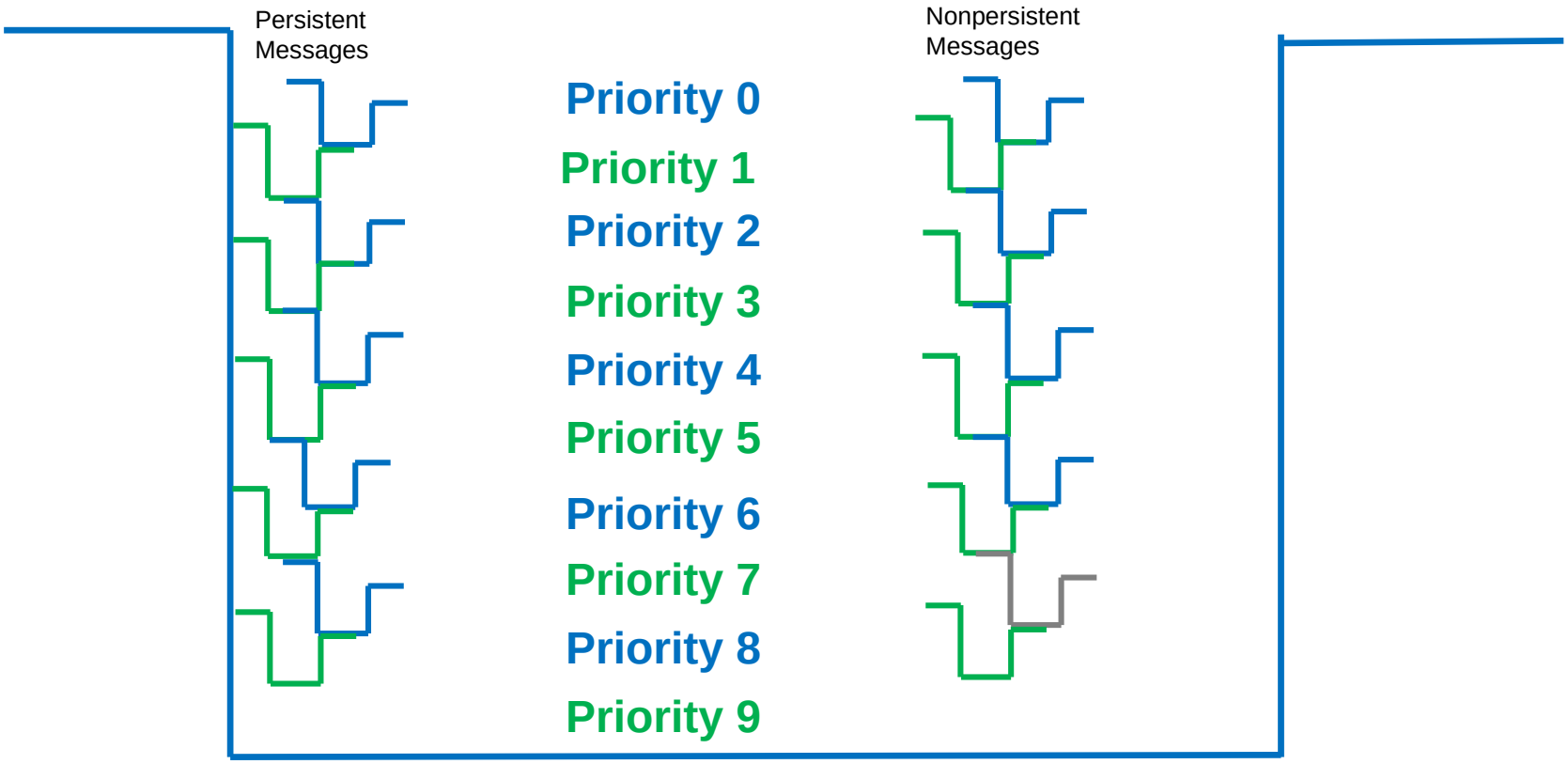
One of these things is not like the other

- MQ is NOT a database

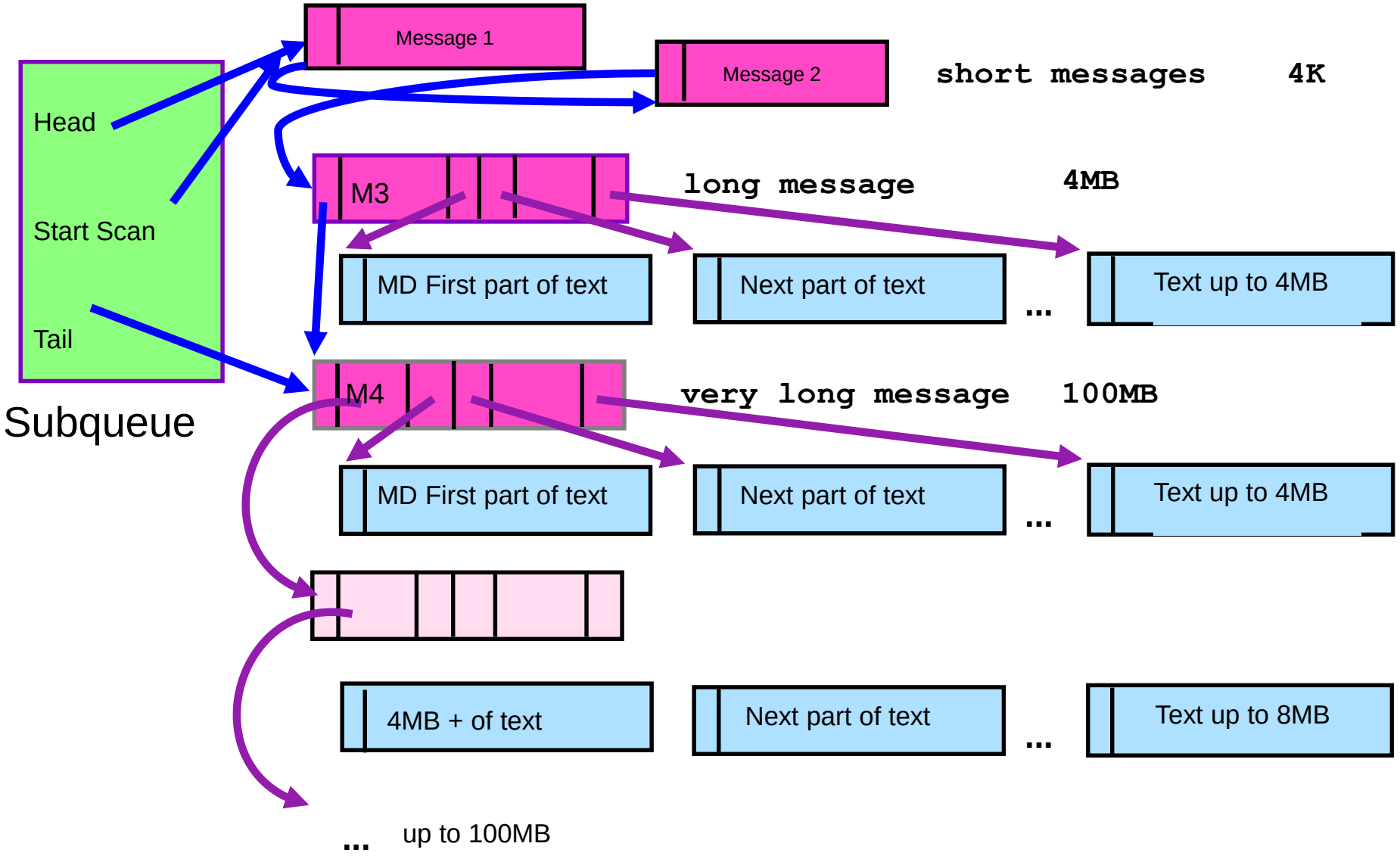


The Internal Representation of a Queue

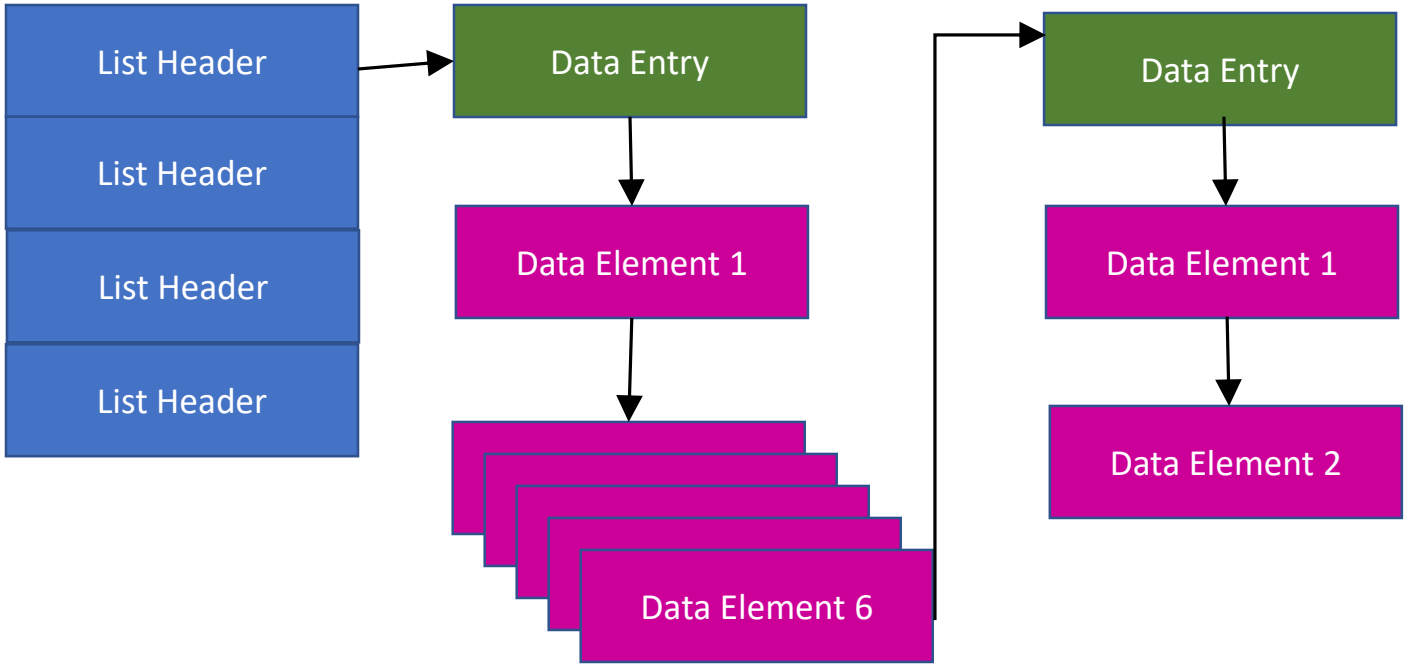
- Sub-queues within a queue



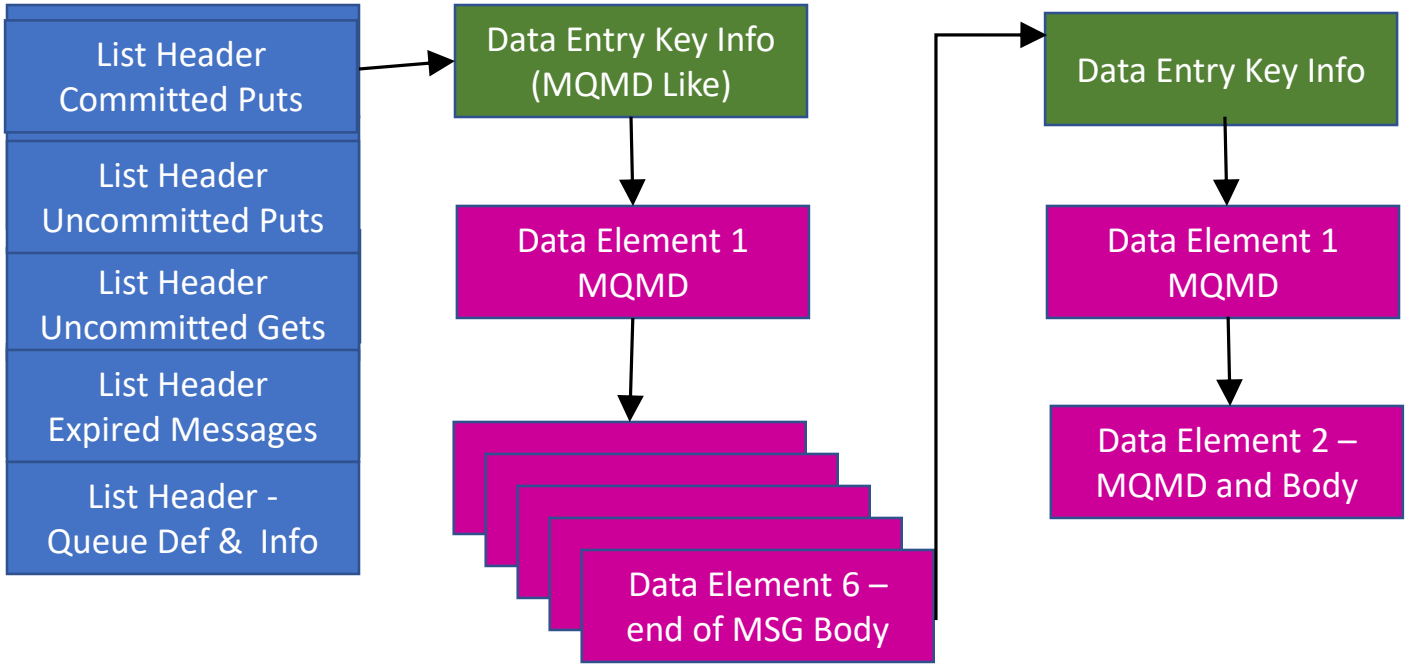
Private Queue Message Storage



Coupling Facility List Structure

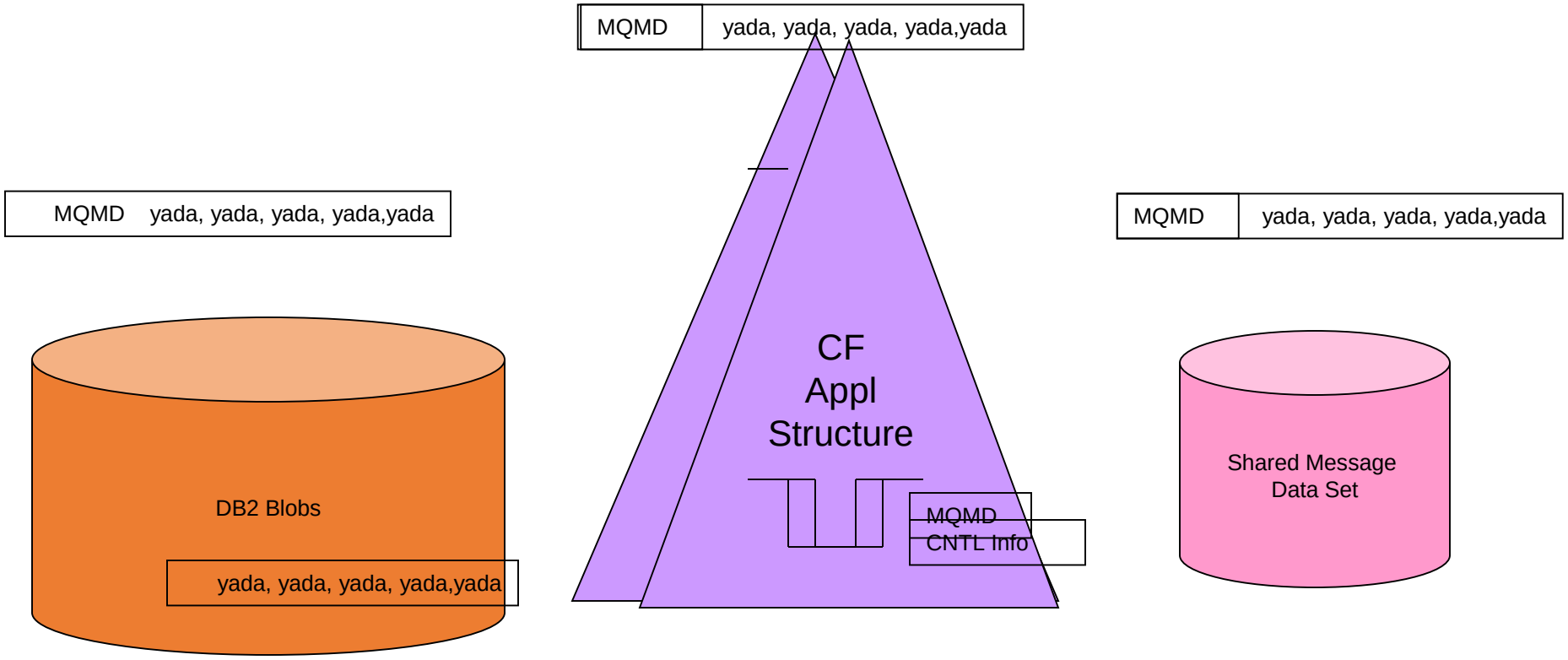


Coupling Facility List Structure What does a queue look like?



Shared Queue Message Storage

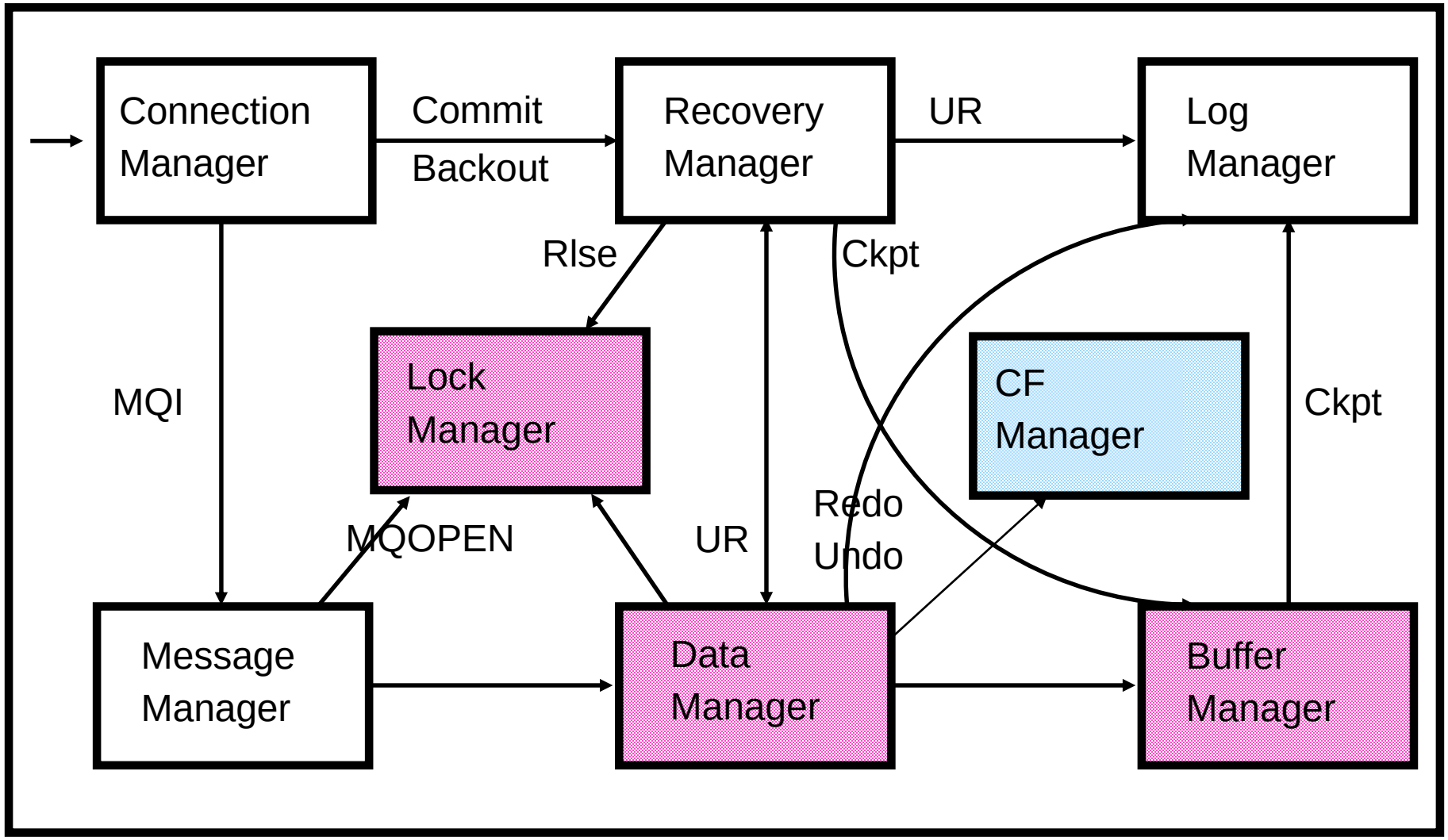
- Messages are stored in one of several ways:
  - Entirely within the list structure
  - Control information (CI) on list structure, message body in DB2
  - CI on list structure, message on Shared Message Data Sets
  - CI and/or message moved to Flash Memory (not shown)



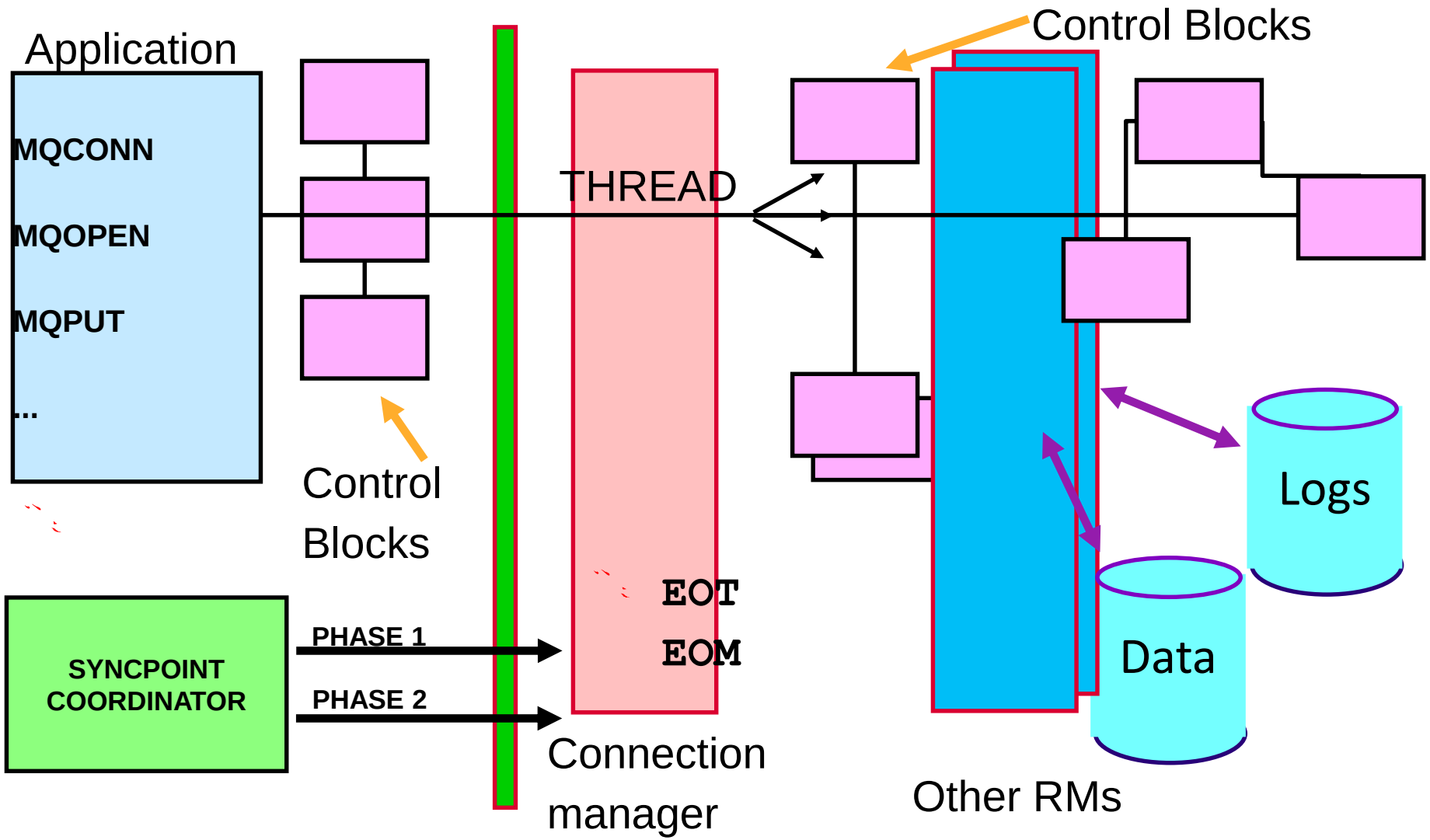
First Line Managers – who does the real work

- 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……

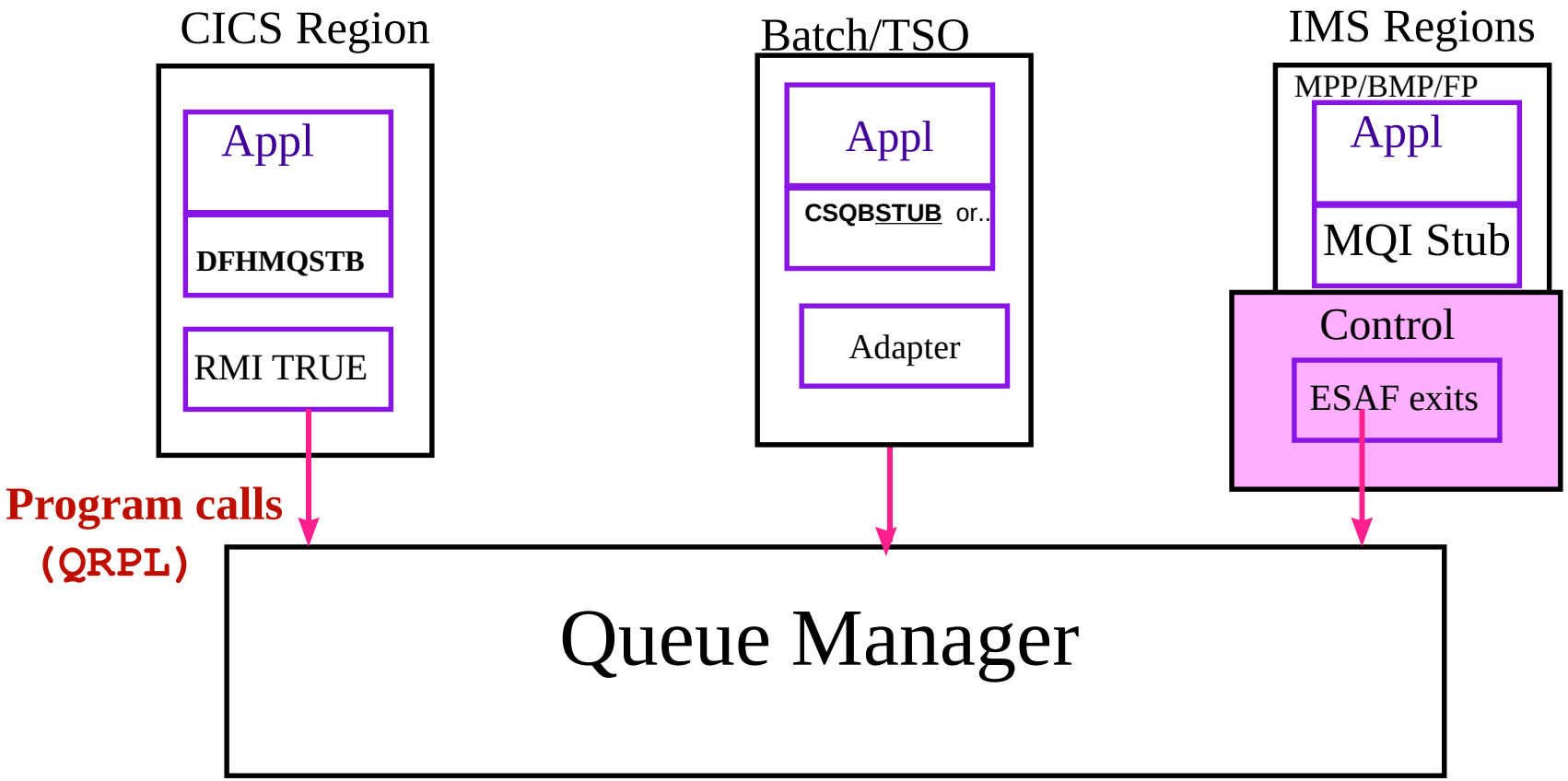
Building Blocks - Resource Managers



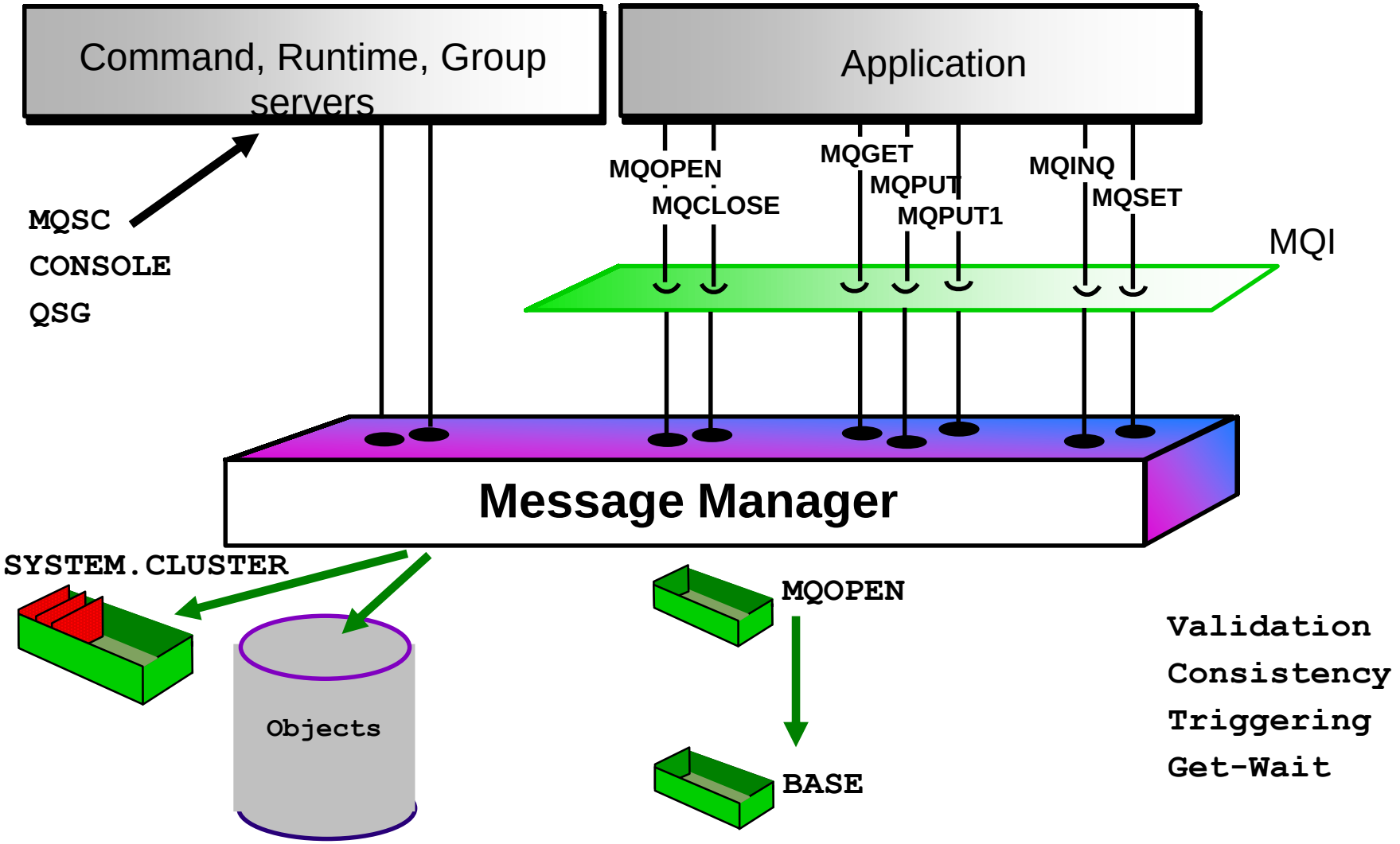
Handling Applications - Connection Manager



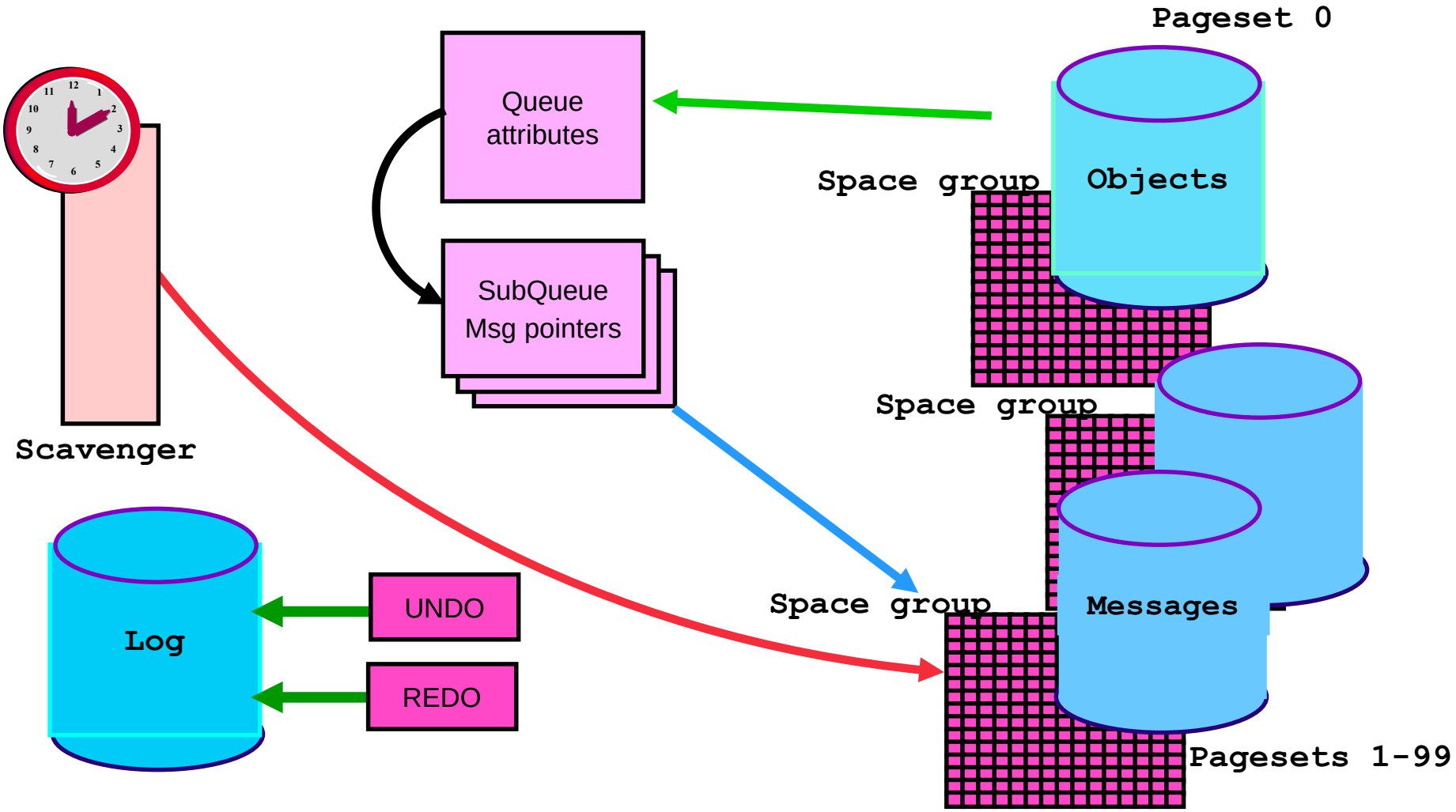
Getting requests into WMQ - Stubs and Adapters



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

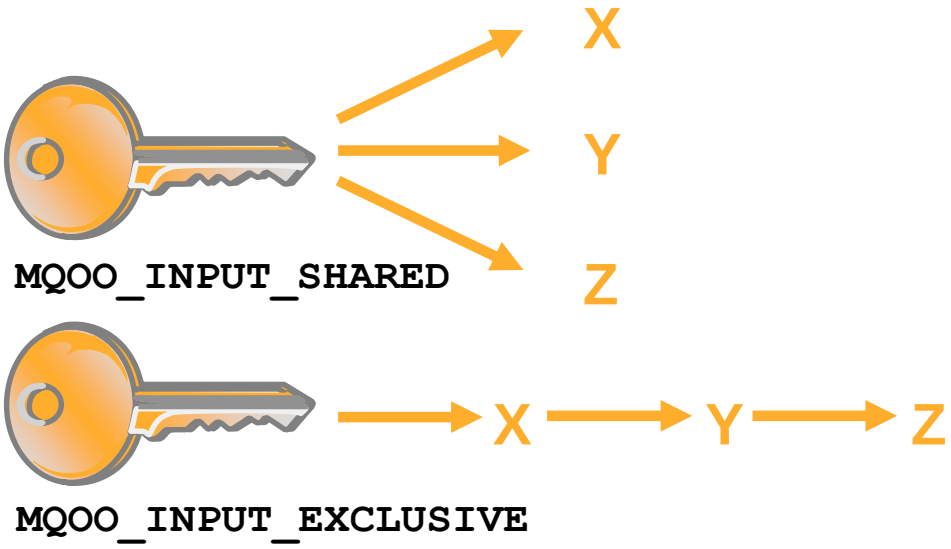
checkpoint

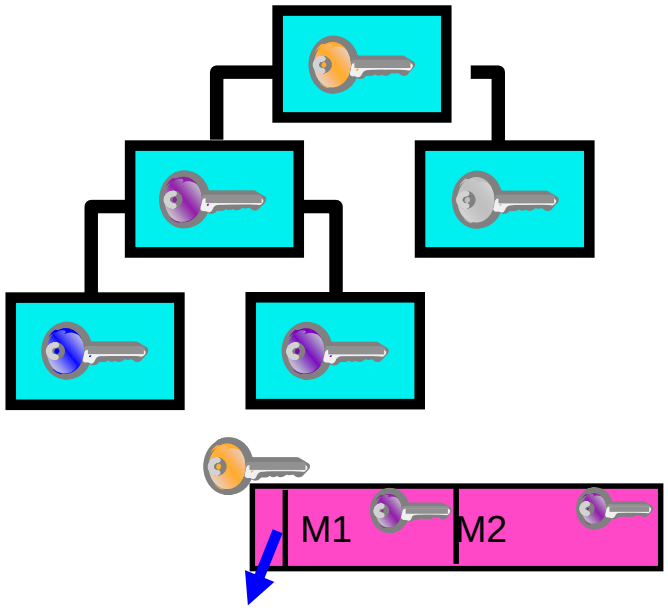
FORCE DIRTY PAGES > 3 CKPTS

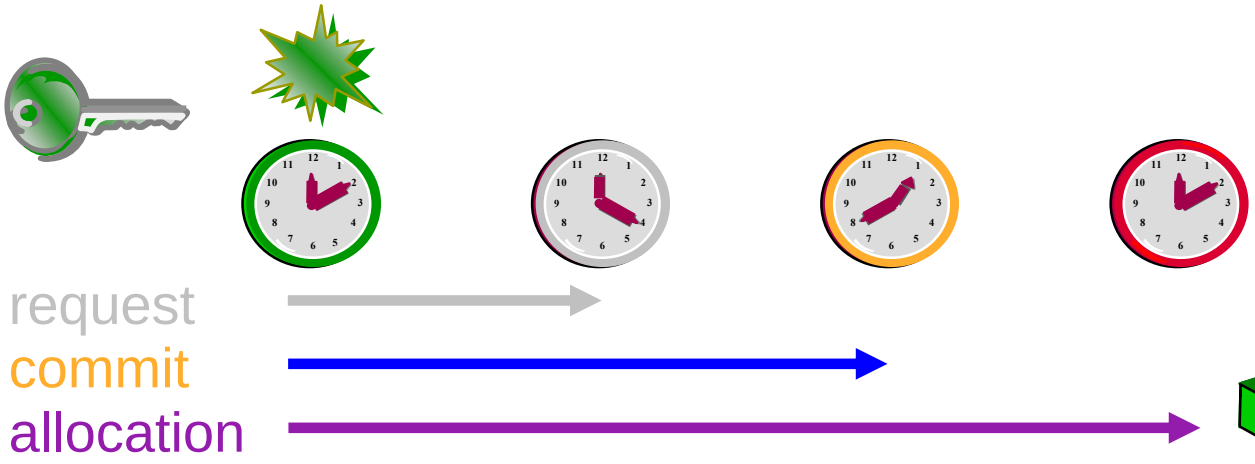
Providing Logging Interfaces - Log Manager

- 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







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

MQ & ACE Sessions (Room 201B unless stated)

Day	Monday	Tuesday	Wednesday	Thursday	Friday
8:30			27054: Hybrid Integration - App Connect, Connecting Cloud and on-Prem **David Coles**	27060: Using MQ in Multiple Cloud Environments - Docker, Kubernetes, OpenShift, AWS, Azure, and more **Mark Taylor**	27083: CICS and MQ - New and Vintage **Room 201A** **Lyn Elkins & Mitch Johnson**
9:45	27055: Introduction to IBM MQ - Enterprise Messaging That Makes Your Life Easier [z/OS & Distributed] **Simon Page**		27061: Welcome to IBM Event Streams (Apache Kafka) **Subhajit Maitra**	27065: z/OS Container Extensions - Running MQ on a Container on z/OS and Integrating with MQ on zOS **Subhajit Maitra**	27068: Where's My Message in MQ and in IBM Event Streams (Kafka)? **Neil Johnston**
11:00	27064: A techn... IBM App Conn... & Distributed] **David Coles**	...uted]	27053: Better MQ z/OS Performance Through Understanding the Internals [z/OS] **Lyn Elkins**	26226: Lab: MQ - Hands-on Labs for MQ 9.1 ~or~ z/OS Connect - Hands on Lab Experience **Hands On-Lab: Room 102** **Lyn Elkins, Mitch Johnson & Mark Taylor** **…. The Lab Continues …..**	27069: Scalable, Fault Tolerant Messaging with MQ Shared Queues on z/OS and Uniform Clusters on Distributed [z/OS & Distributed]
12:25 PM					
1:45 PM	27063: What's New in the Messaging Family - MQ v9.1 and More [z/OS & Distributed] **Mark Taylor**				
2:15 PM			27057: Making sense of queues and event streams : IBM MQ vs IBM Event Streams (Apache Kafka) **Subhajit Maitra**		
3:00 PM	27062: What's New in IBM App Connect Enterprise [z/OS & Distributed]	27059: Security: Everything You Wanted to Know about SSL/TLS Principles but Were Unsure Who to Ask [z/OS & Distributed]		27067: Monitoring z/OS and Distributed Queue Managers with Open Source Tools	
3:30 PM	**David Coles**	**Neil Johnston**		**Mark Taylor**	
4:15 PM	27066: MQ SMF Data: What We Continue to Learn about Statistics and Lies [z/OS] **Lyn Elkins**	27070: Implementing RESTful Services Using IBM MQ and z/OS Connect **Mitch Johnson**		27058: Securing your Enterprise: Understanding MQ Security through Scenarios and Roles [z/OS & Distributed] **Neil Johnston**	

You are Here!

MQ for z/OS: Better Performance from knowing a bit about the internals
Session 27053

Lyn Elkins – WSC – elkinsc@us.ibm.com

Mitch Johnson – WSC – mitchj@us.ibm.com