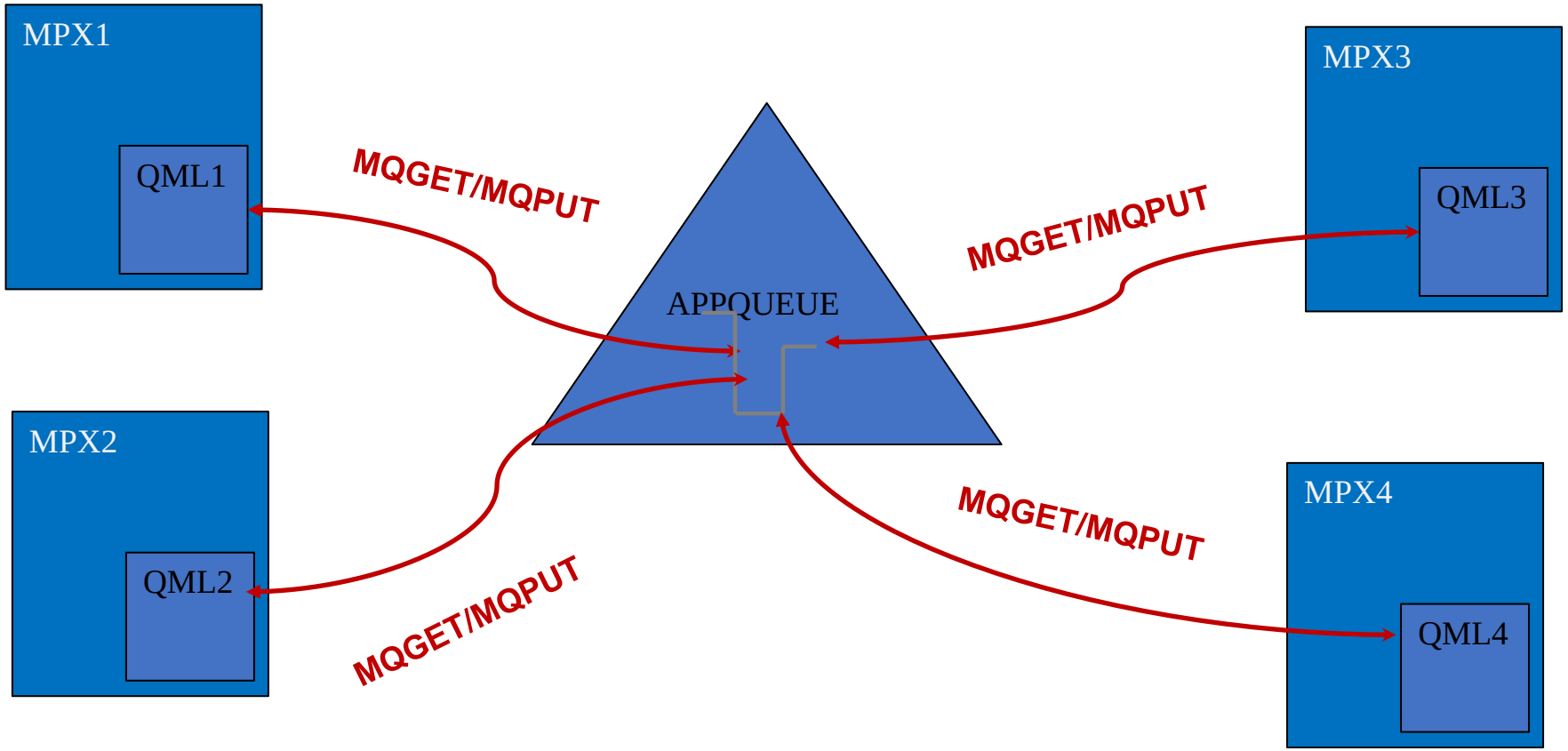
Shared Queues – where is my workload running

Steve Bohn – bohns@us.ibm.com
Mitch Johnson – mitchj@us.ibm.com
Lyn Elkins – elkinsc@us.ibm.com

Agenda

- What are shared queues – briefly
- What is workload skewing and why is it a problem?
  - What are the symptoms and causes
    - Asymmetrical Sysplex
    - Connection Skewing
    - Put to Waiting Getter
  - ‘ Local’ favoritism
- Mitigation Techniques:
  - Queue Manager Clustering
  - Gateway queue managers
  - CICS CPSM options

Briefly – What is a Shared Queue

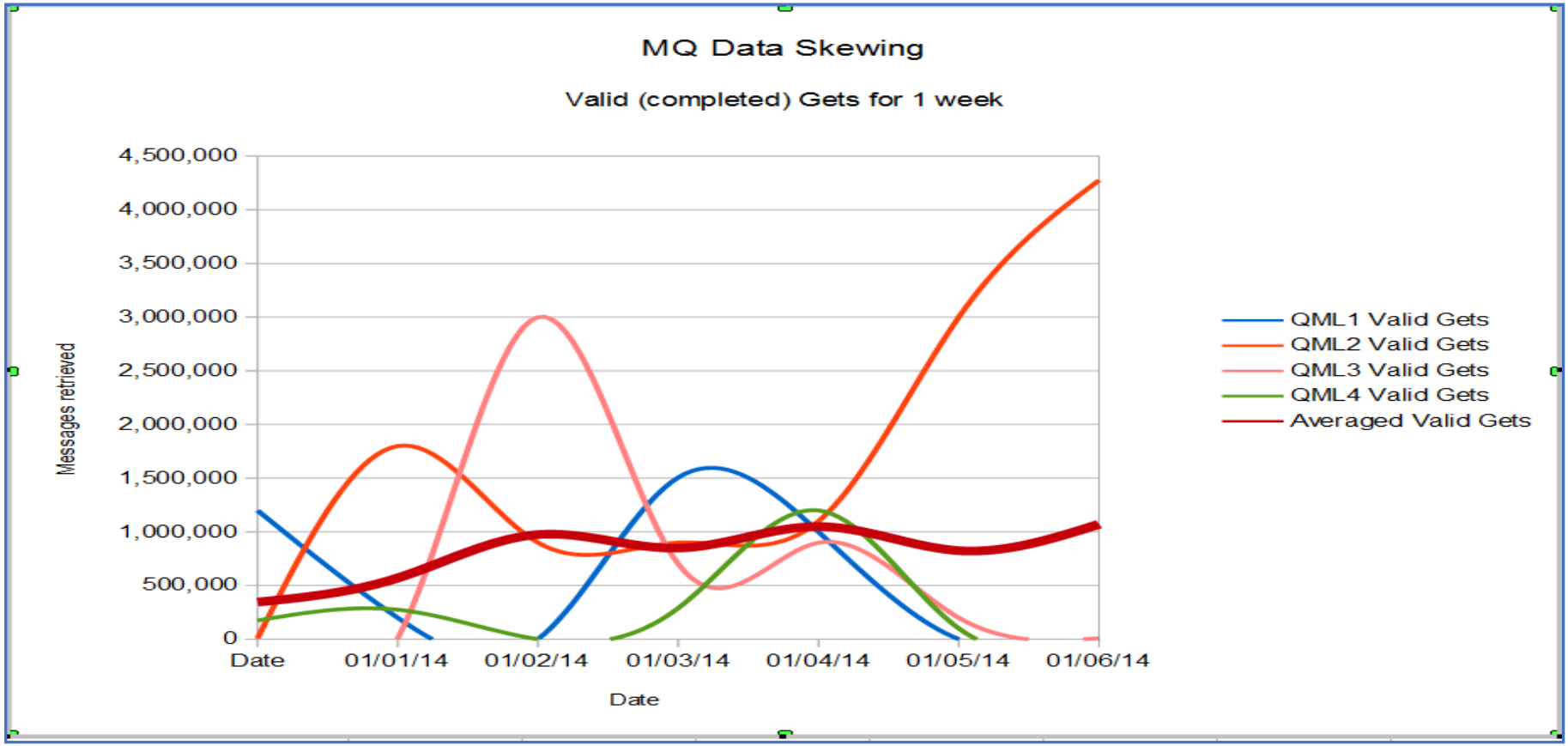


What is a shared queue?

- Unique to z/OS
  - Requires a coupling facility to host the queues
  - DB2 data sharing for the queue definitions
  - The gold standard for message availability
- Treats a shared queues as local to each queue manager in the QSG
  - Applications can PUT and GET
- Messages are available as long as one QMGR in the QSG can access the Coupling Facility Structure.
- Nonpersistent messages are only ‘lost’ if the structure or CF itself are lost.

What is MQ Workload Skewing?

- Workload skewing is detected when MQ driven work, typically transactions, is not close to being evenly distributed across the queue managers.

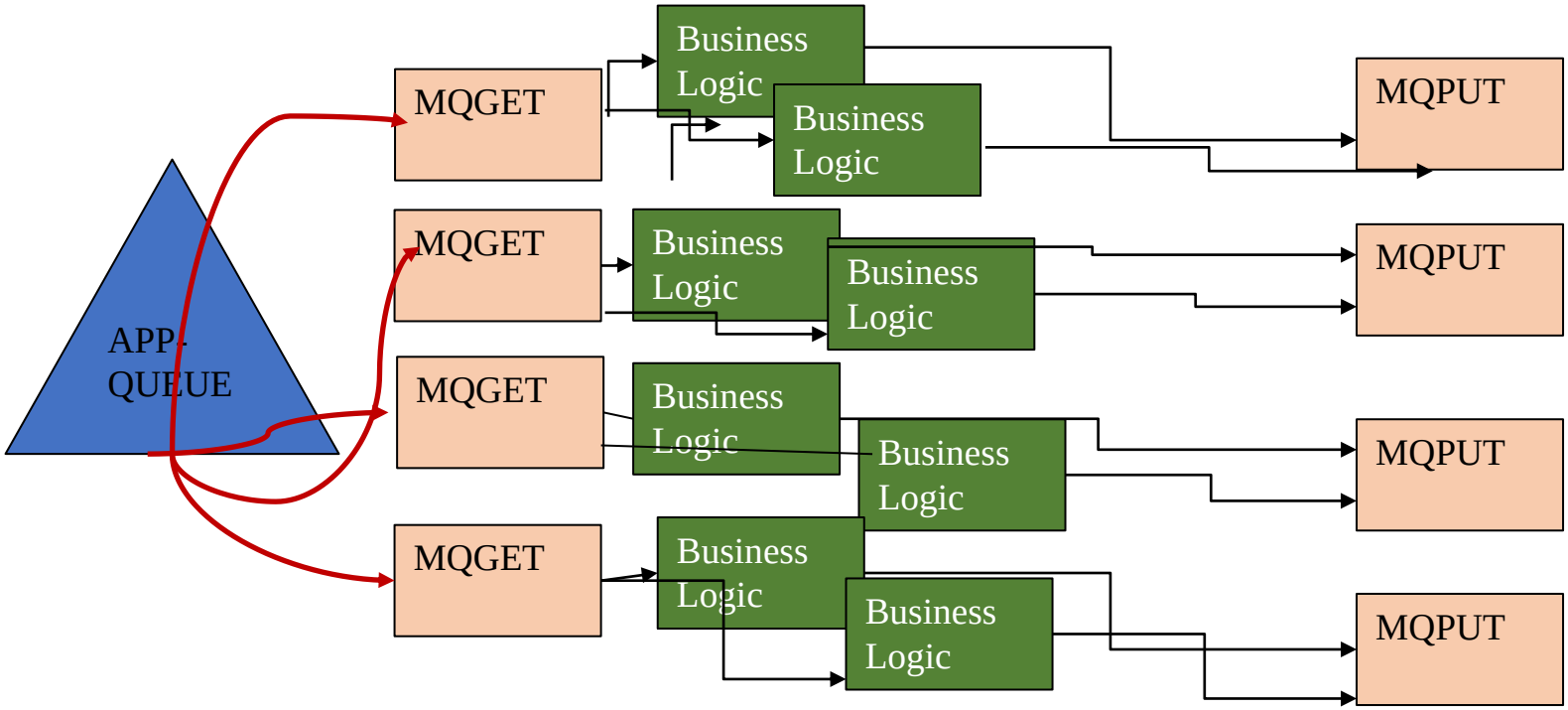


Why is MQ Workload Skewing a problem?

- This is often less a technical problem, more of a COST problem
  - If the MLC ‘rolling average’ is taken from the LPAR that is heavily favored, usage pricing is not going to reflect reality
  - Technical solutions to this problem may prove to be less efficient overall - lower throughput, slower response

Why is MQ Workload Skewing a problem?

- Can cause increased capacity demands in downstream workload
  - Known to produce responsiveness problems
    - Overloading the processing programs
  - Again, this can contort MLC charges

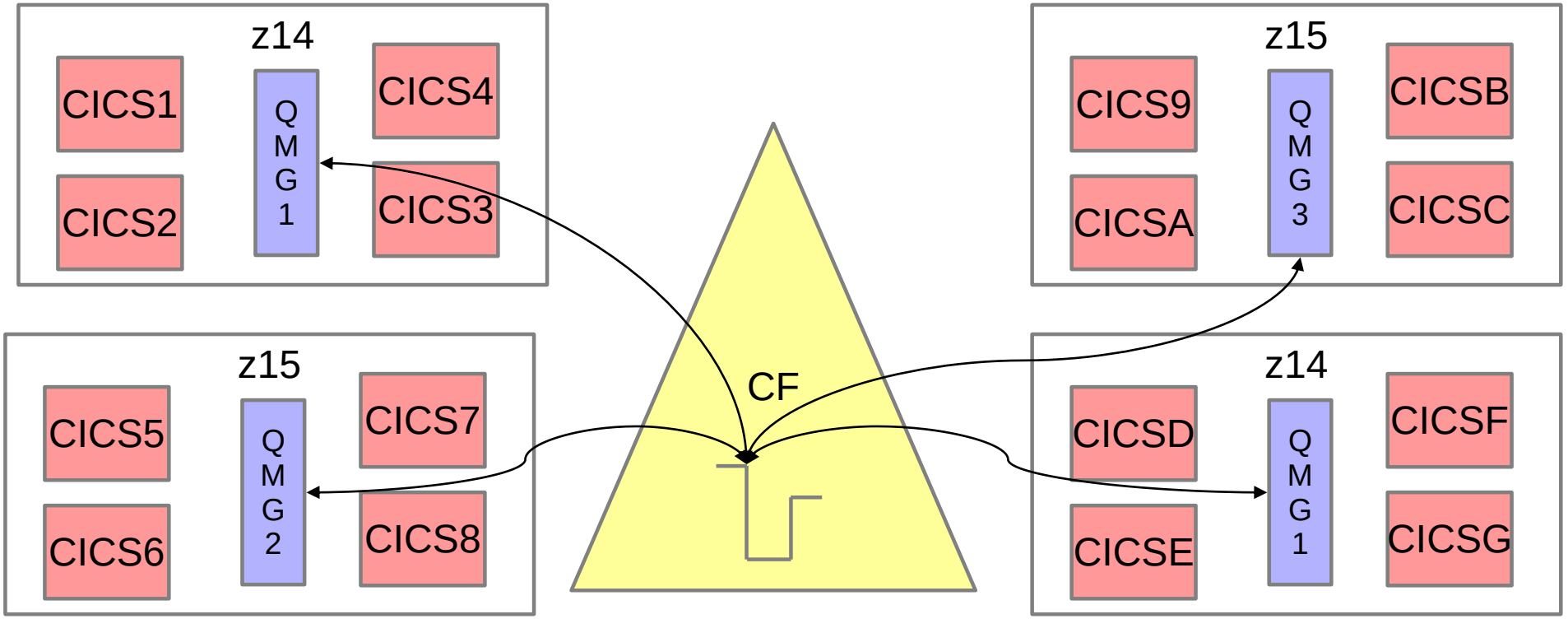


MQ Workload Skewing Causes

- Workload skewing in a QSG is often a result of the efficiencies of working locally
  - z/OS, and all subsystems, try to process requests locally to take advantage of CPU efficiency
- Workload skewing may be intentional
  - Some applications may be affinity bound to an LPAR, but are using shared queues for the additional availability
  - Some applications are not yet Sysplex enabled
  - Software licensing agreements requiring LPAR restriction
  - Recommendation – know when this is the case and document.

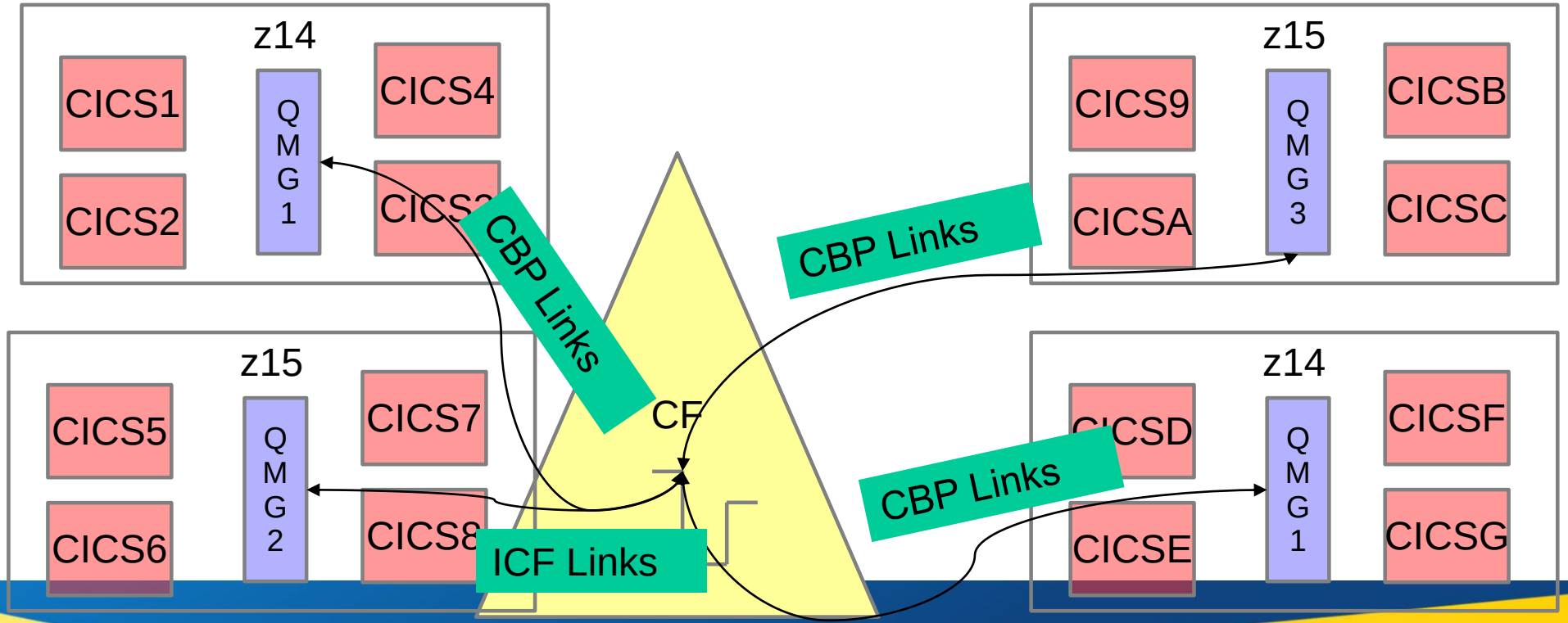
MQ Workload Skewing Causes - Hardware

- Asymmetric Sysplex
  - When the LPARs in the Sysplex are not equally weighted
    - Examples include:
      - Two LPARs have z14s, two have z15s
      - Two LPARs have 12 dedicated engines, two have 12 shared
      - All links to the CF are equally distant and are the same type



MQ Workload Skewing Causes - Hardware

- **Asymmetric Sysplex**
  - Most common example - One LPAR is co-located with the primary coupling facility, the others are on different CECs
  - ICF links give much better service times than CBP



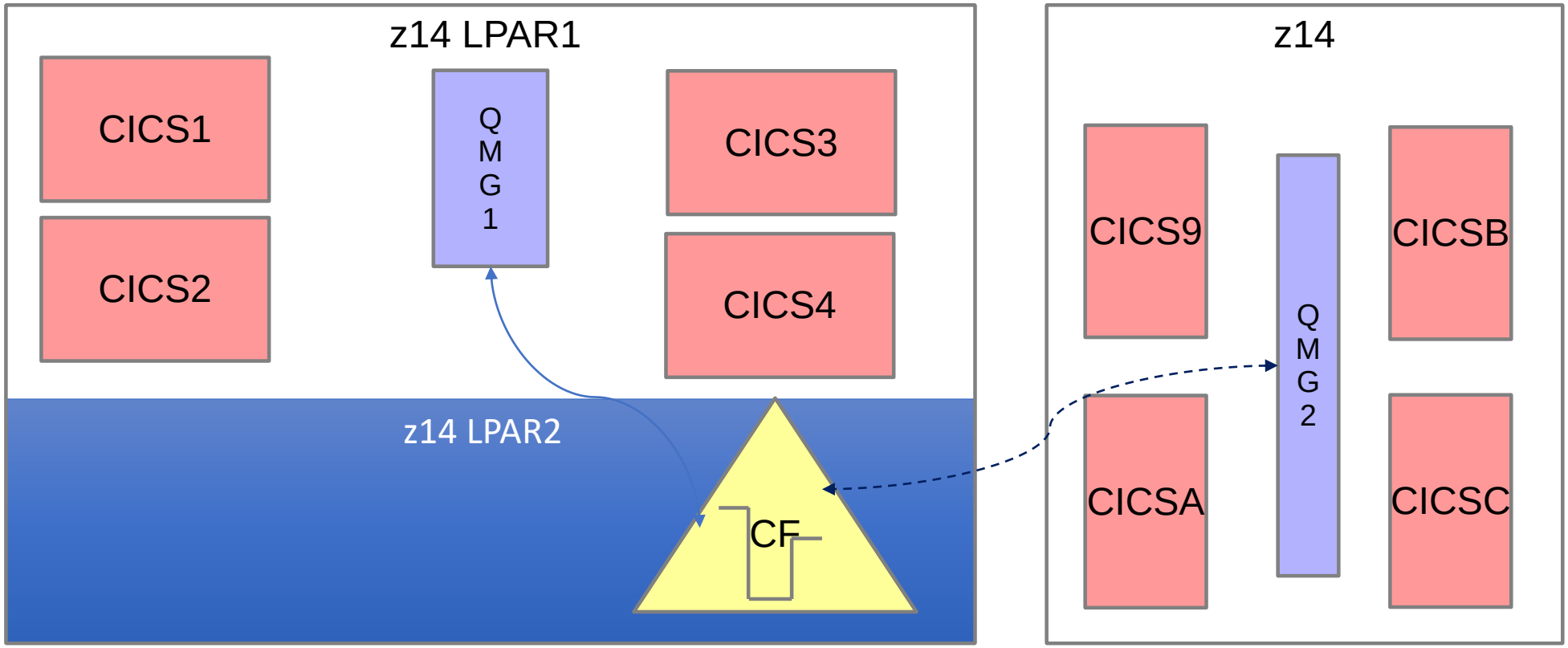
Physical Skewing – CF Activity Report

```
STRUCTURE NAME = QSGBUSER          TYPE = LIST    STATUS = ACTIVE
          # REQ    -------------- REQUESTS -------------   -------------- DE
SYSTEM    TOTAL                 #      % OF  -SERV TIME(MIC)-    REASON     #    %
NAME      AVG/SEC               REQ     ALL     AVG    STD_DEV              REQ     R

MPX1       295K   SYNC      295K    26.9      4.3       1.2     NO SCH      0     0
          492.1   ASYNC        0     0.0      0.0       0.0     PR WT       0     0
                  CHNGD        0     0.0  INCLUDED IN ASYNC     PR CMP      0     0
                  SUPPR        0     0.0                        DUMP       0     0

MPX2       802K   SYNC      802K    73.1     17.8       2.5     NO SCH      0     0
          1339    ASYNC        0     0.0      0.0       0.0     PR WT       0     0
                  CHNGD        0     0.0  INCLUDED IN ASYNC     PR CMP      0     0
                  SUPPR        0     0.0                        DUMP       0     0
```

- We (the WSC) first look at the CF Activity report – usually before the MQ Statistics when looking at shared queue usage
- In the example shown above it is easy to see that the MPX2 LPAR is getting a much longer service time (almost 4 times!) than the MPX1 LPAR and that MPX2 is making many more requests.
  - In this particular case, this exposed some internal workload skewing that was not apparent to the customer - except that they were missing SLAs consistently!

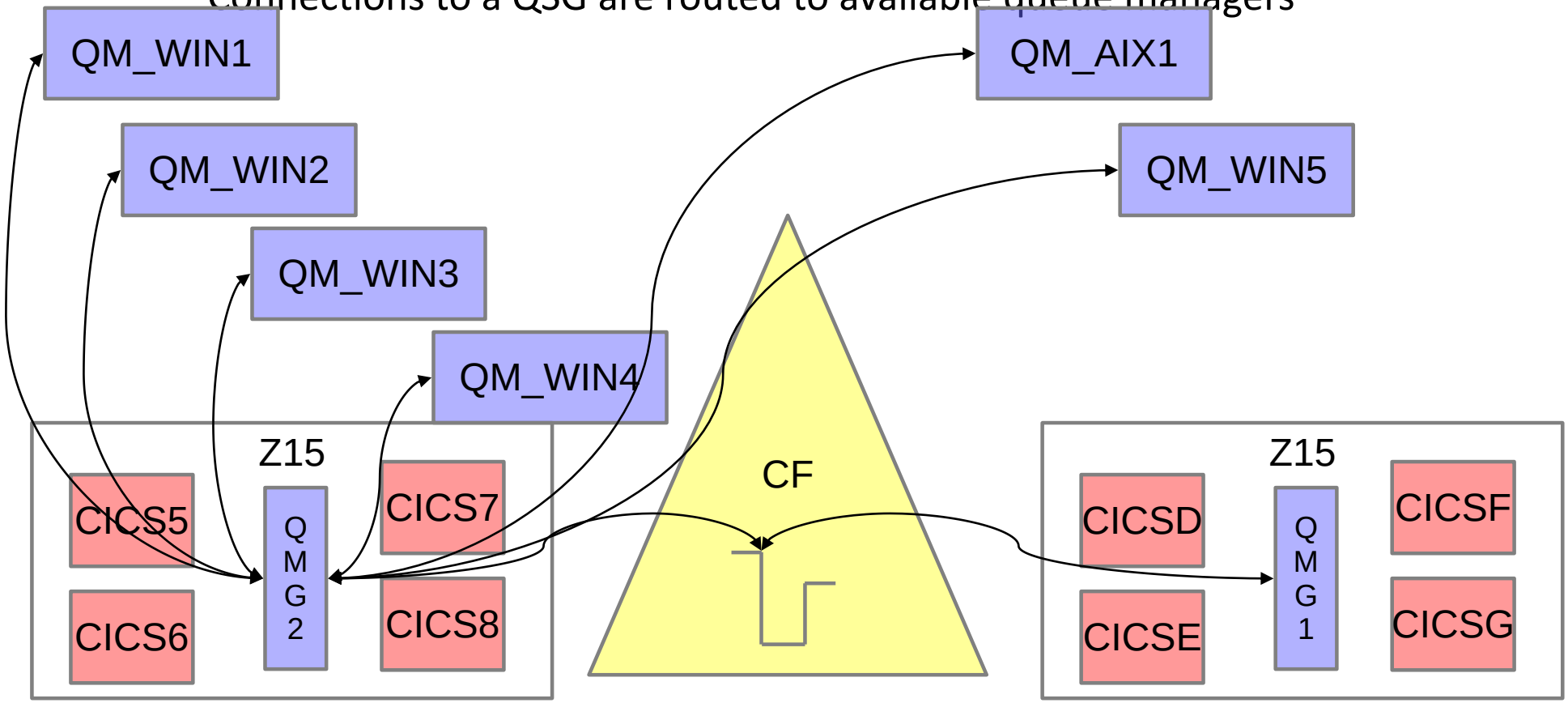
MQ Workload Skewing Causes - Hardware

- Location of the Coupling Facility
  - When the coupling facility is internal, LPARs on the same CEC tend to get faster response
  - When the coupling facility is external and one LPAR has more, faster, or less heavily used links it will get faster service



Connection Skewing

- Connection skewing may be historical
  - Hard-coded connections to specific queue managers
- Connection skewing may be the result of a queue manager outage
  - Connections to a QSG are routed to available queue managers

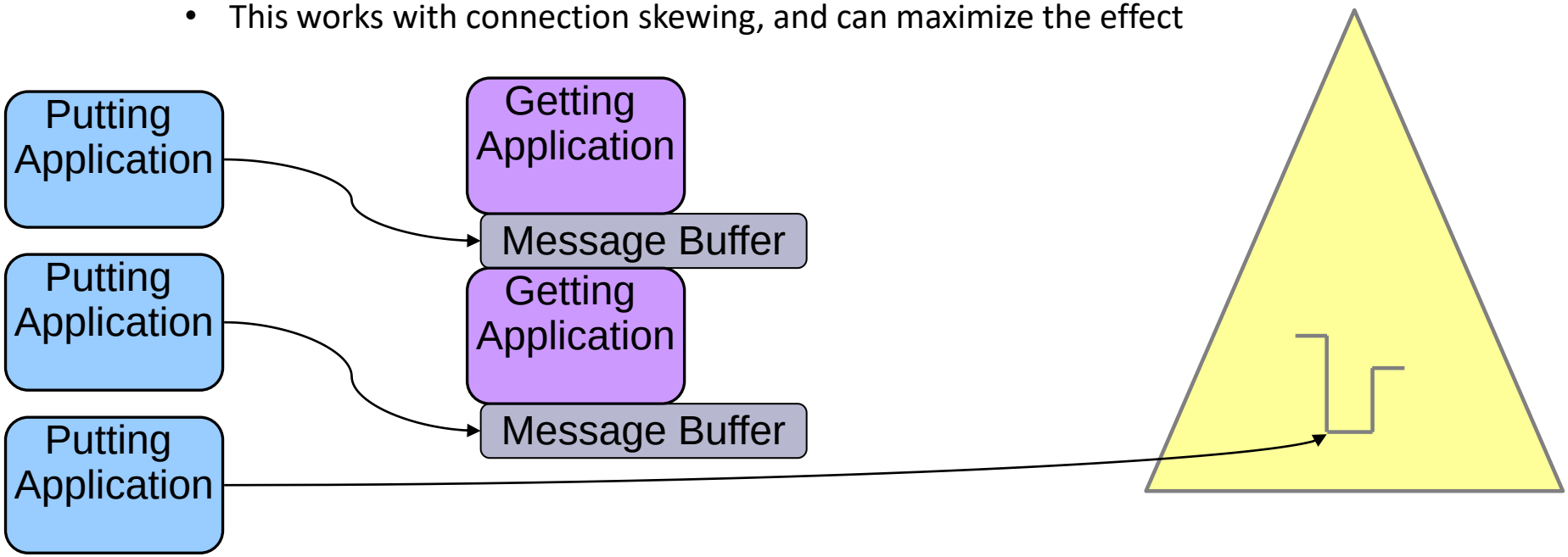


‘Downstream’ consequences

- We’ve talked about the MLC impact
- Resource use
  - Not every queue manager is sized to absorb the entire workload
  - Log impact of skewing has been seen
    - Rapid Log switches due to heavier workload – increasing I/O and CPU costs
  - Bufferpool/Pageset impact
    - Filling the bufferpool, forced into I/O
  - SMDS impact
    - One queue manager in QSG gets all offloaded messages

MQ Workload Skewing Causes

- Put to waiting getter
  - In V6 a performance feature was added called ‘put to waiting getter’
  - If a local put, from an application or message channel agent, is done and there is a getting application waiting the message is moved directly to the getting applications buffer
    - There is no posting to a shared queue
    - There is no notification to other available waiting applications
    - The CPU savings can be substantial
    - This works with connection skewing, and can maximize the effect



Put to Waiting Getter – SMF

- This shows messages flowing across a channel taking advantage of P2WG

Base_Name	CF Struct	Total_Valid_Gets	Total_Bytes_Put	Total_Valid_Puts	Total_Put2_Waiting_Getter	Puts not to Waiting Getter
SYSTEM.QSG.CHANNEL.SYNCQ	CSQSYSAP	0	0	0	0	0
SHARED.INPUT.QUEUE	APP1	0	4501092223	2095814	2012394	83420

- The CPU comparison shows why it can be a good thing!

BASE_NAME	VALID_PUTS	PUT_ELAPSED_TIME	PUT_CPU_TIME	PUT2_WAITING_GETTER	Average PUT ET	Average PUT CT
QLOCAL.PUT2WG	14879	127753	117956	14793	8.59	7.93
QLOCAL.NO.PUT2WG	41547	1025028	1010038	0	24.67	24.31

- The CPU costs can be 3 times as high!

MQ Workload Skewing Causes

- Local Favoritism
  - When a message is posted to a shared queue, the queue manager where the message is put is typically notified FIRST about the availability.
  - Normal processing by XCF, taking advantage of the efficiency of local processing.

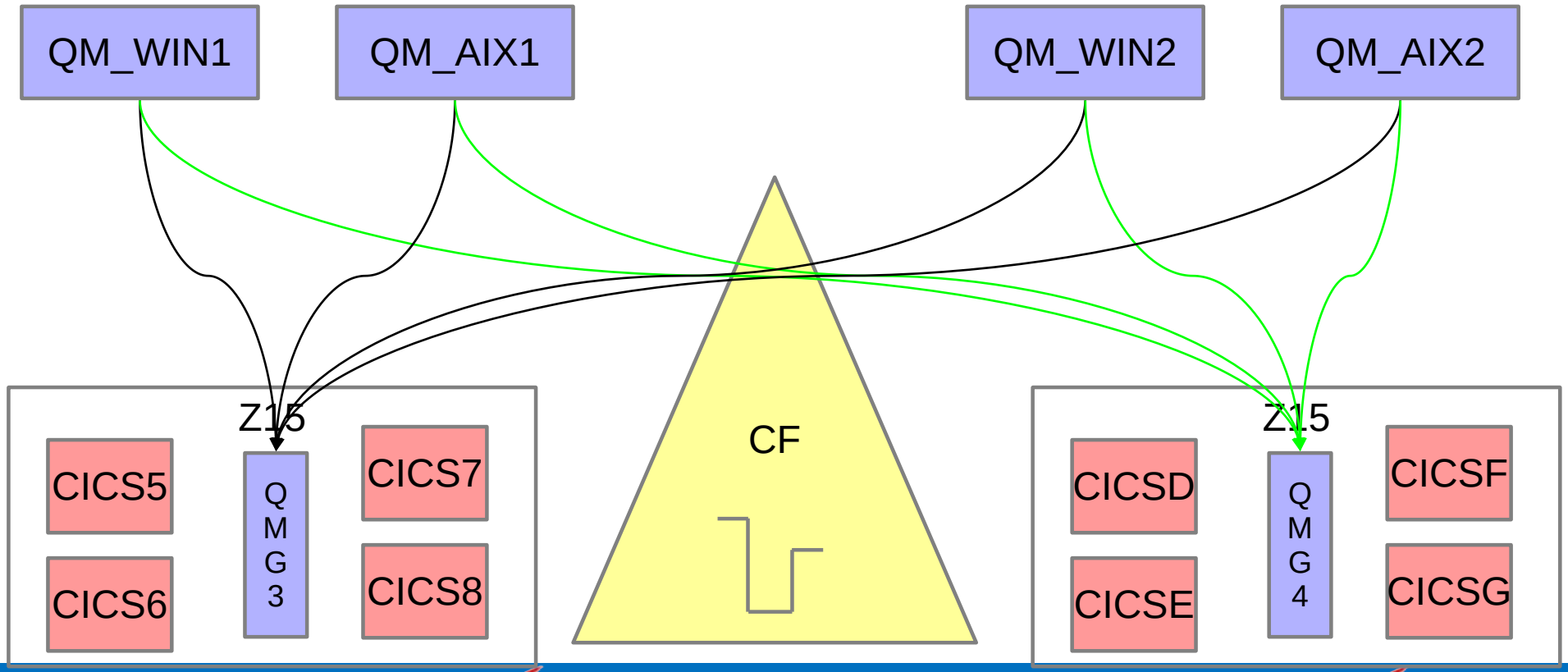
Skewing Mitigation Techniques

- Queue Manager Clusters
  - Clusters provide workload balancing across queue managers
  - Works with shared queues to distribute message ‘puts’ across queue managers in the QSG
- Connection skewing mitigation
  - Gateway queue managers
  - Re-driving connections
- CPSM mitigation

Queue Manager Clustering

•When messages are not bound to a specific queue manager (‘bind not fixed’), the messages are routed evenly across the receiving queue managers

- Black arrows show the first message put to the clustered queue
- Green arrows show the second message

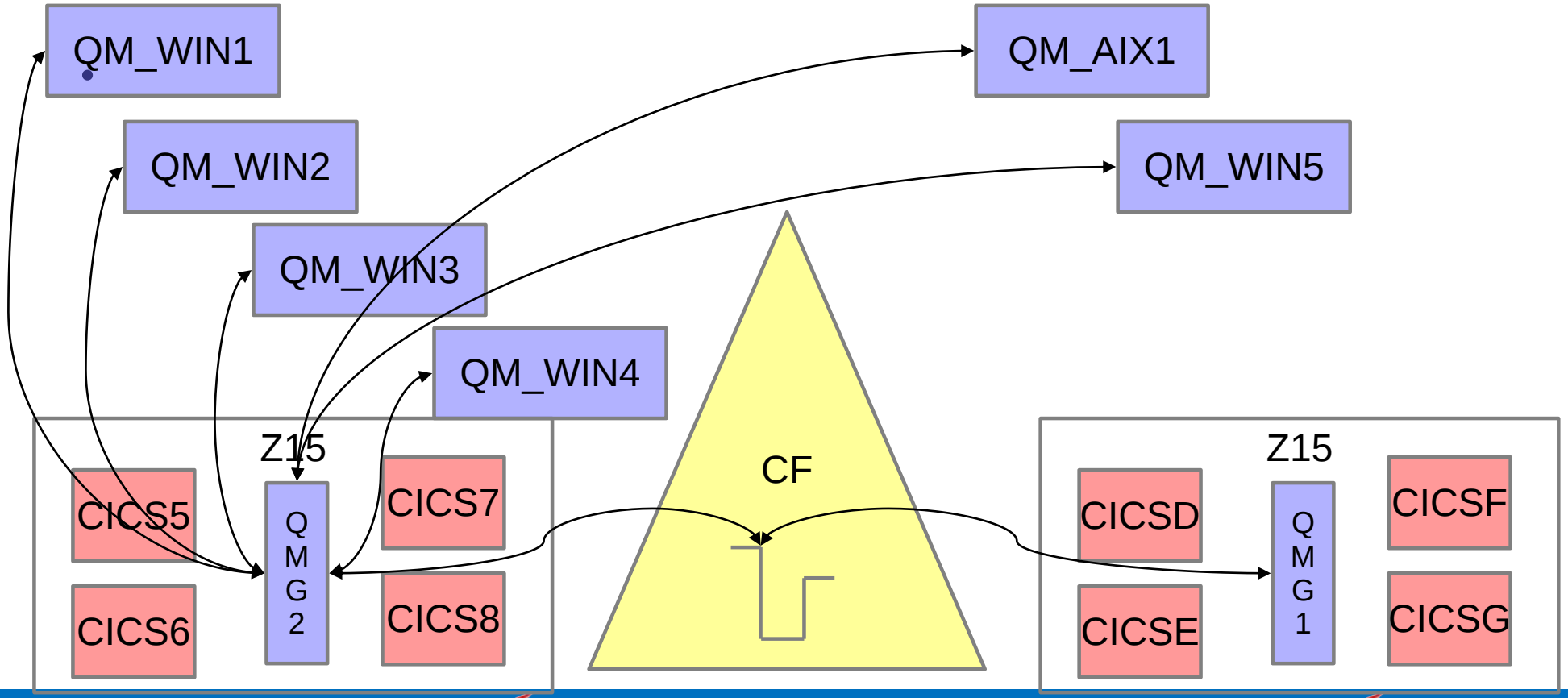


Connection Skewing Mitigation

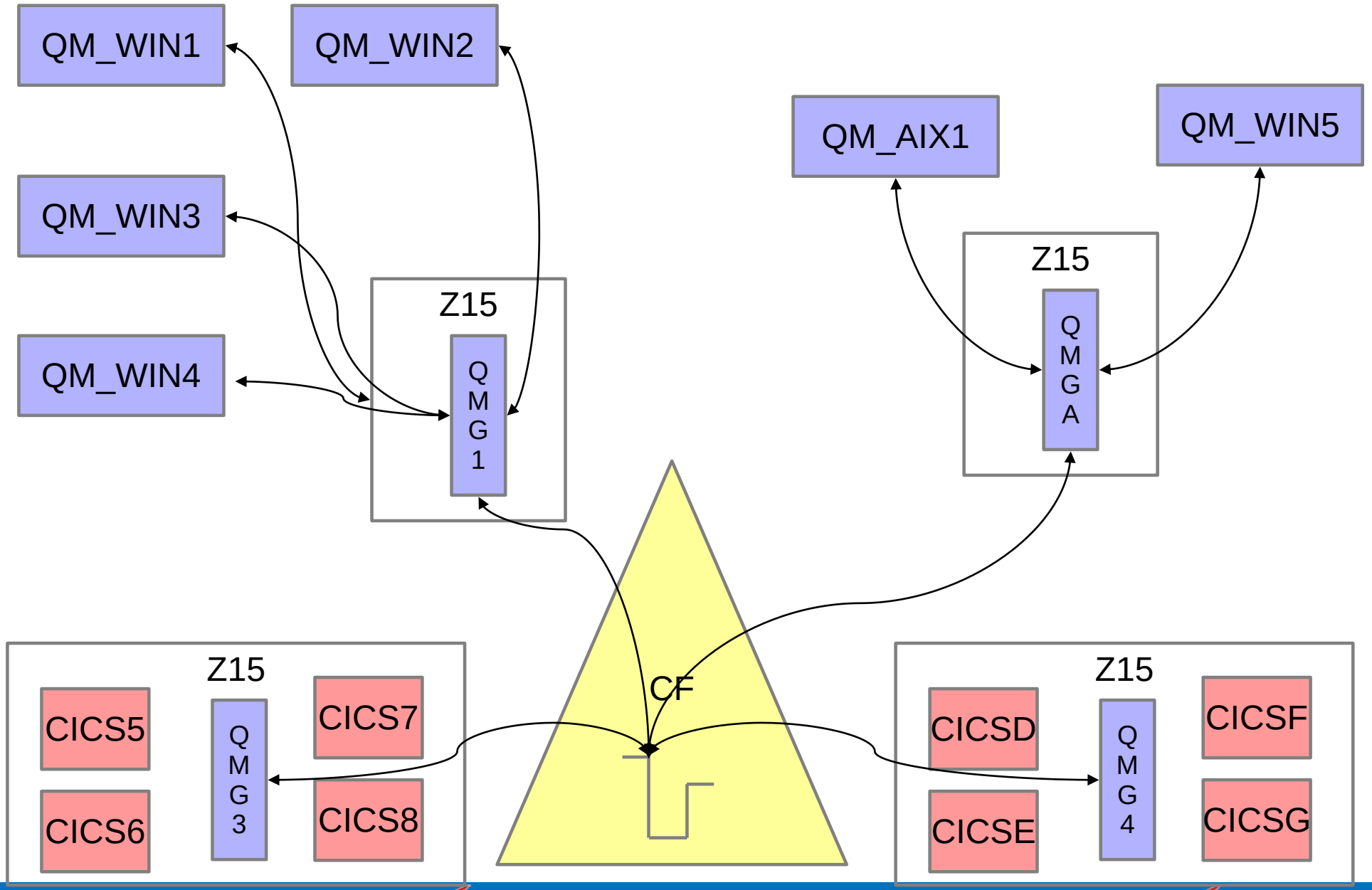
- The slides that follow outline two mitigation techniques for connection skewing:
  - Gateway queue managers
  - Re-driving connections

Connection Skewing – No Gateway queue managers

- When external queue managers or clients are passing work directly to application hosting queue managers, every attempt is made to process the work locally
- Environments that use gateway queue managers into the Queue Sharing group often eliminate connection skewing.

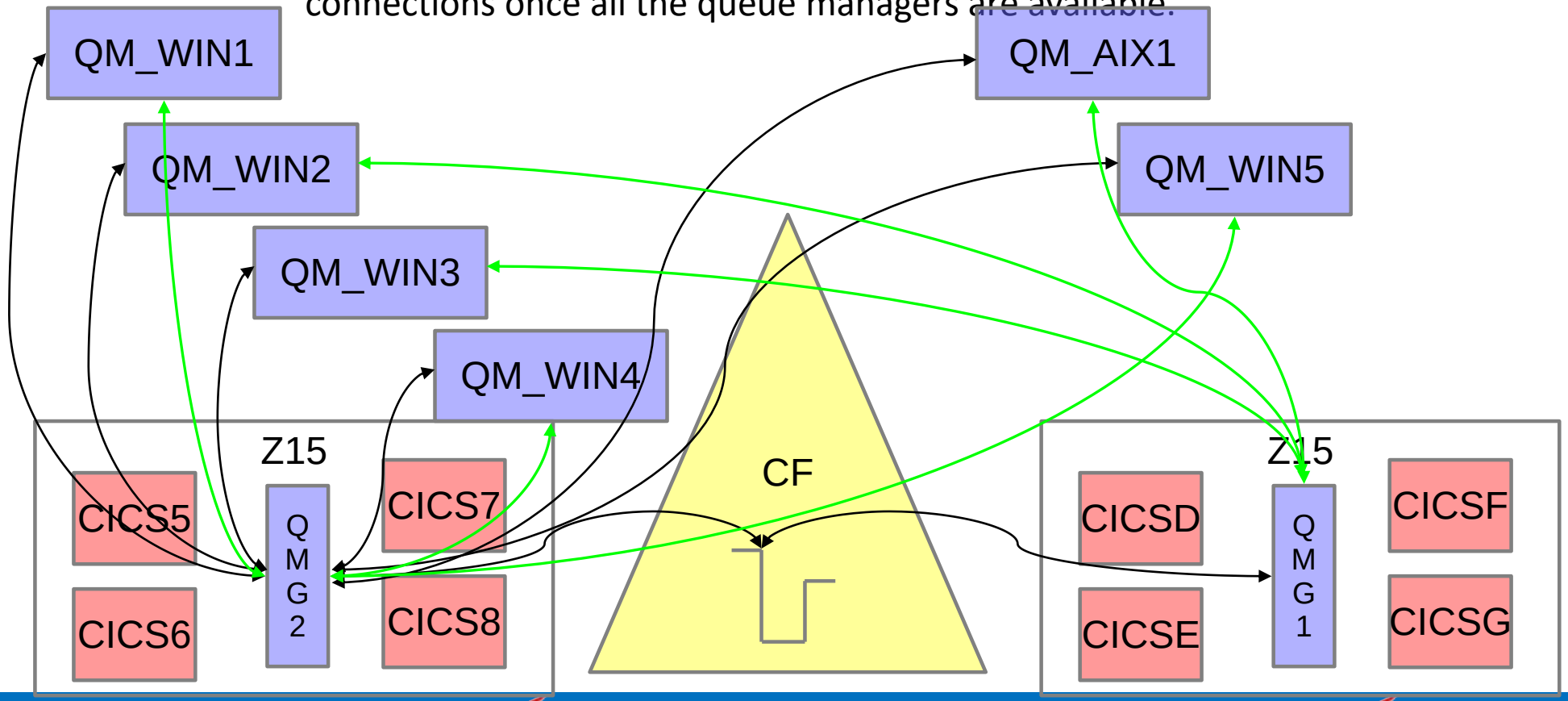


Gateway queue managers – the mitigation



Re-driving Connections

- When a queue manager is unavailable, inbound connections can get skewed to the other queue manager(s) in the group.
  - This is normal availability processing!
  - Once a connection is live and active, no attempt is made to balance the connections once all the queue managers are available.

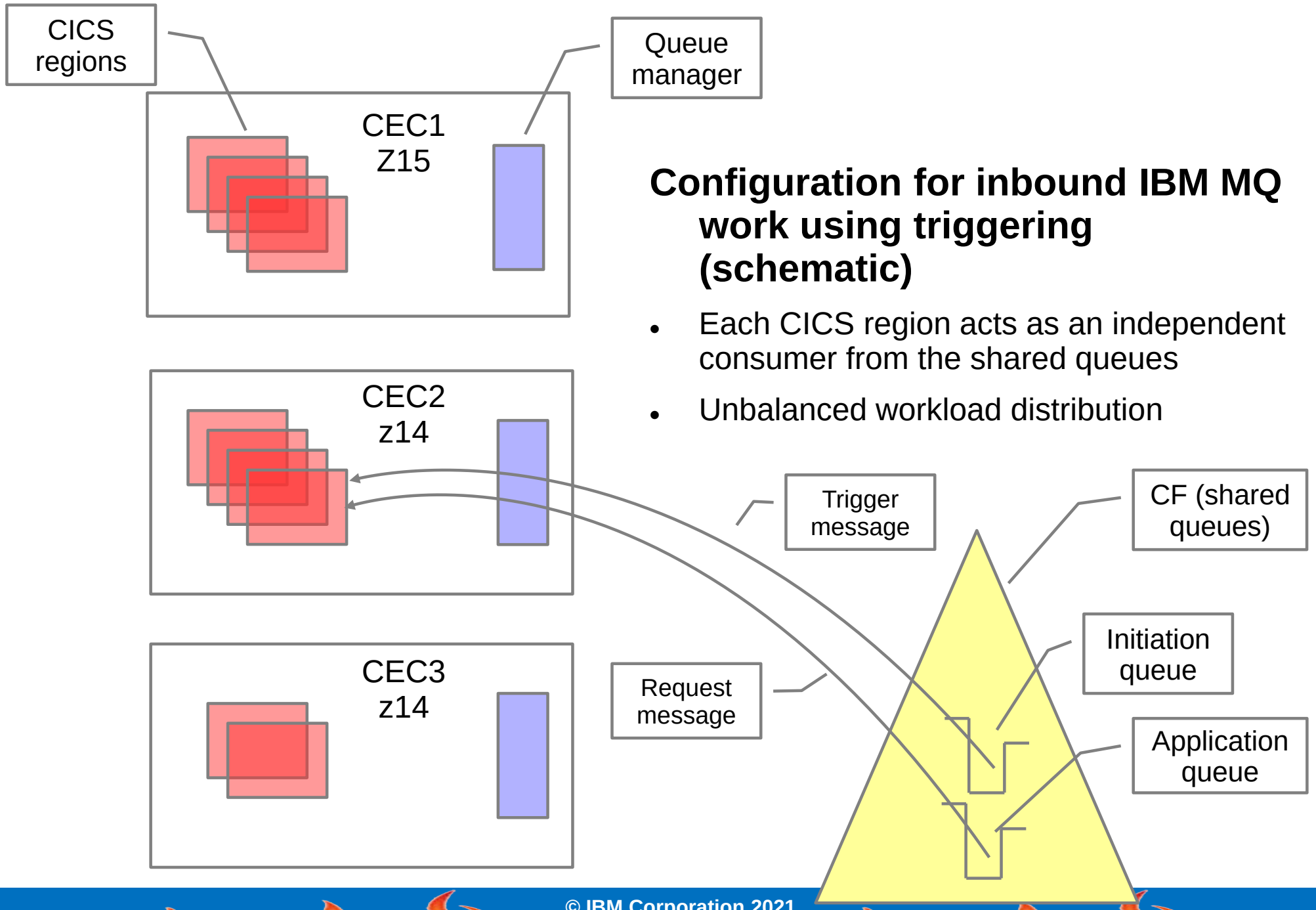


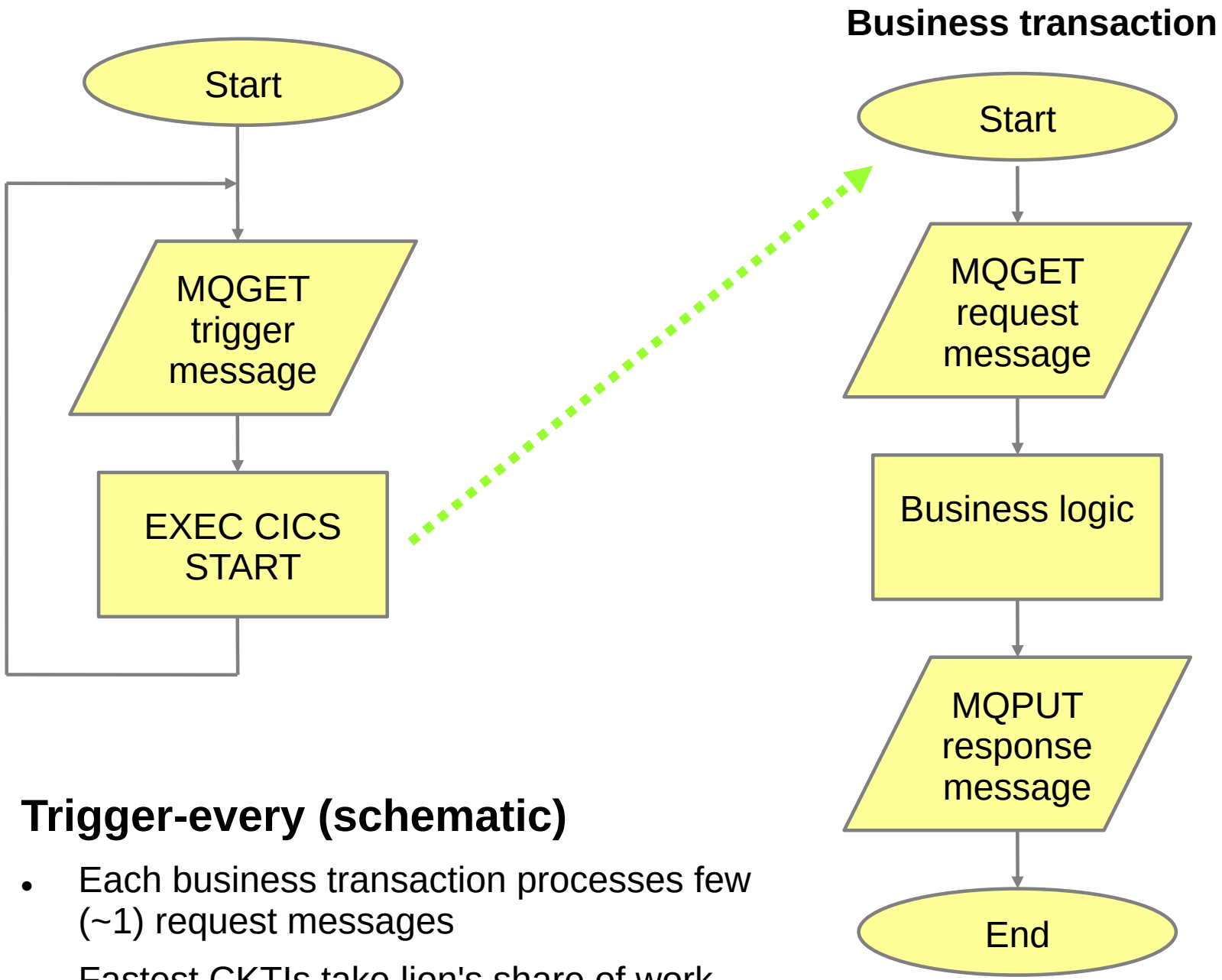
CICS – CPSM Mitigation

- The slides that follow outline a CPSM solution to the skewing problem based on the interaction between MQ triggering (CKTI) and CICS

Configuration for inbound IBM MQ work using triggering (schematic)

- Each CICS region acts as an independent consumer from the shared queues
- Unbalanced workload distribution



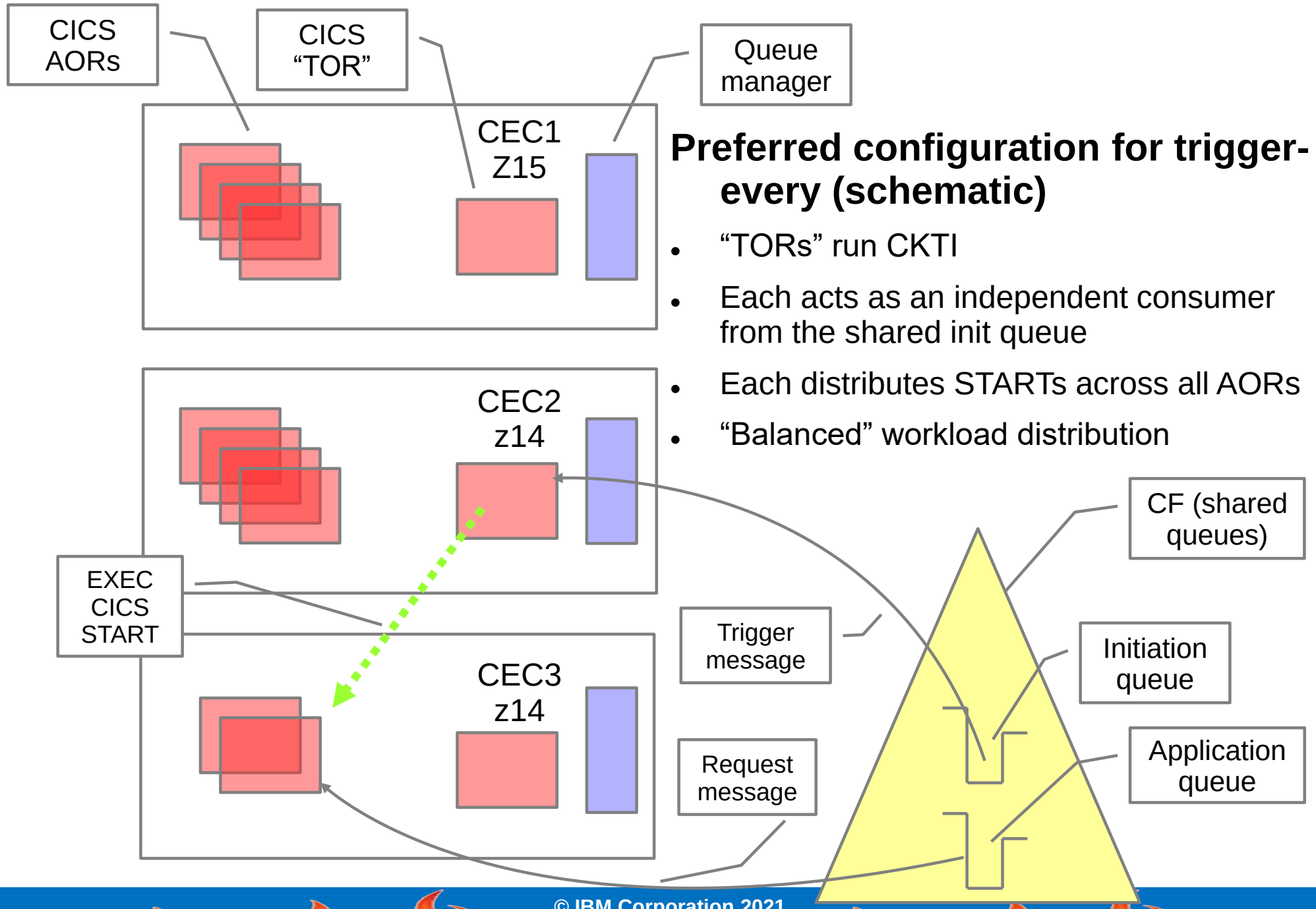


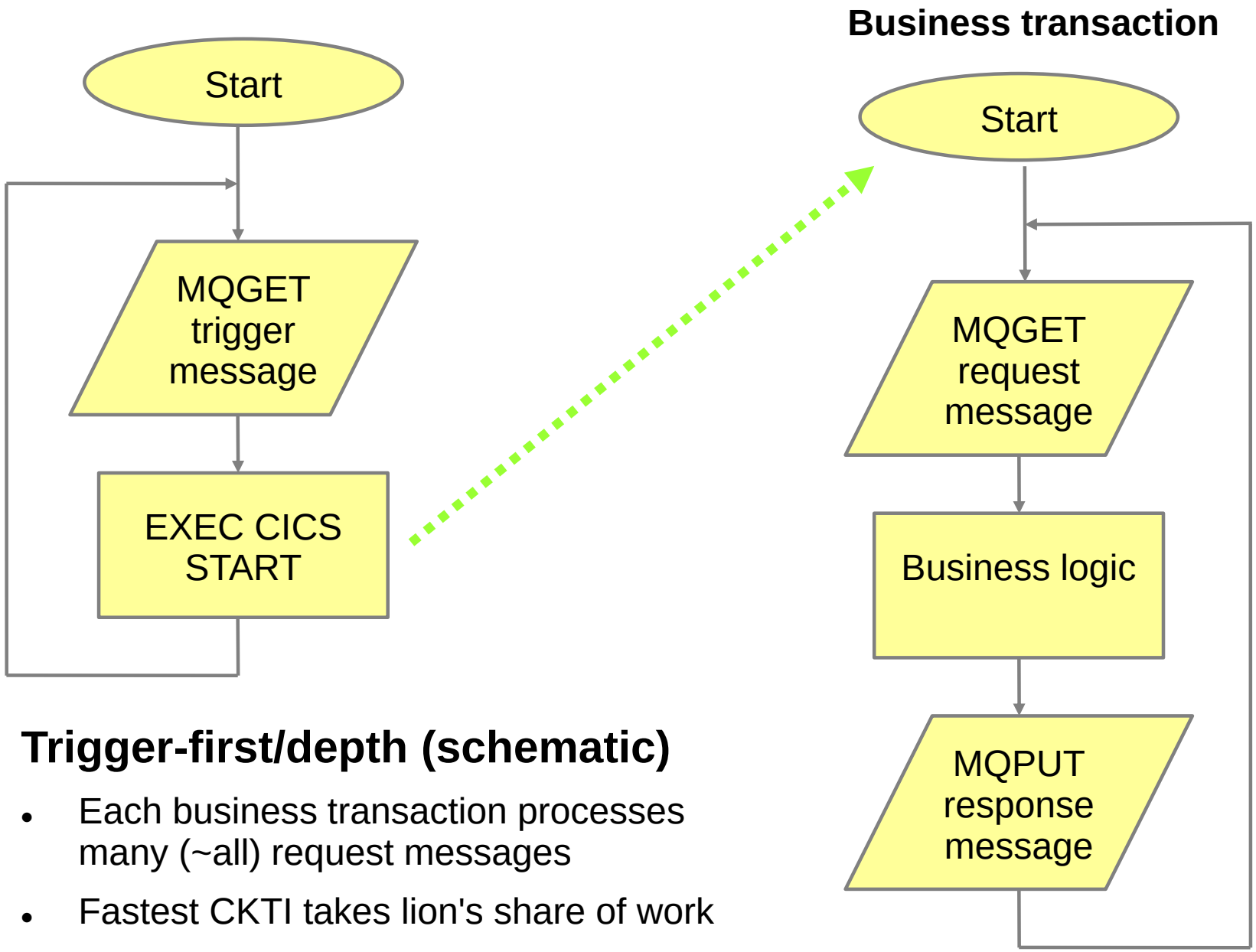
Trigger-every (schematic)

- Each business transaction processes few (~1) request messages
- Fastest CKTIs take lion's share of work

Preferred configuration for trigger-every (schematic)

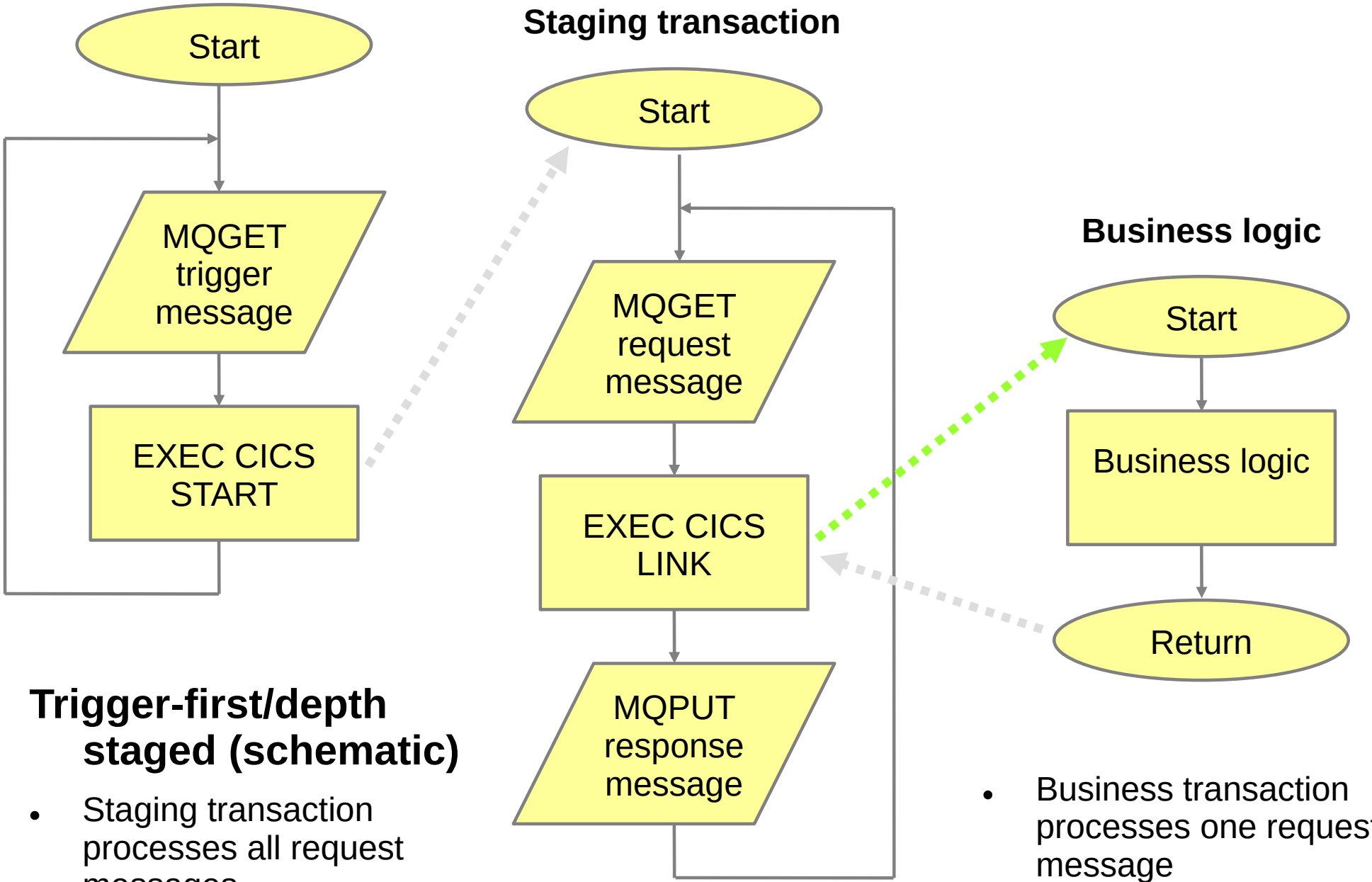
- “TORs” run CKTI
- Each acts as an independent consumer from the shared init queue
- Each distributes STARTs across all AORs
- “Balanced” workload distribution





Trigger-first/depth (schematic)

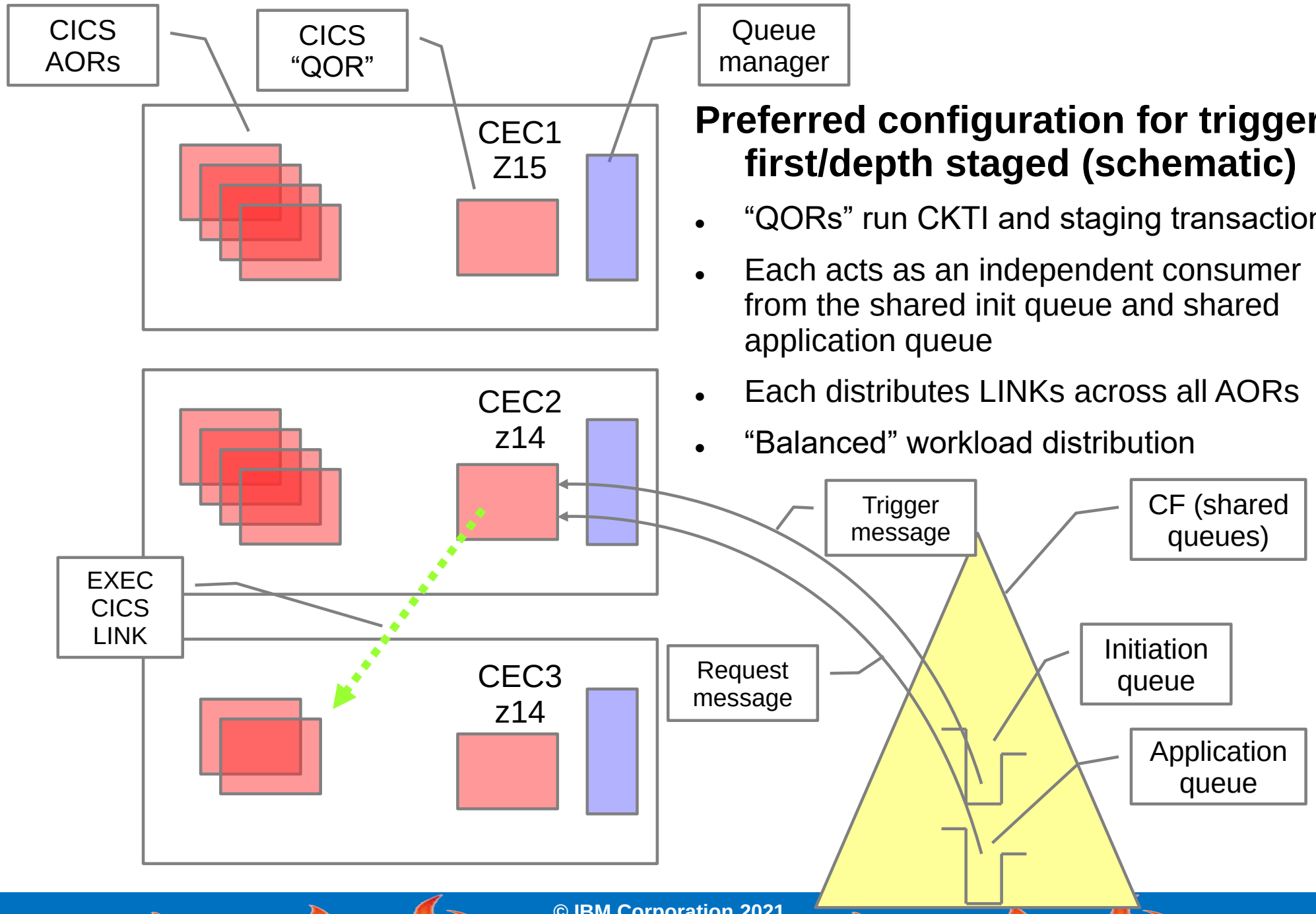
- Each business transaction processes many (~all) request messages
- Fastest CKTI takes lion's share of work
- Corresponding business transaction takes lion's share of the work



Trigger-first/depth staged (schematic)

- Staging transaction processes all request messages

- Business transaction processes one request message



Preferred configuration for trigger-first/depth staged (schematic)

- "QORs" run CKTI and staging transaction
- Each acts as an independent consumer from the shared init queue and shared application queue
- Each distributes LINKs across all AORs
- "Balanced" workload distribution

Highlights

- Solution uses proven technology for CPSM routing:
  - Each TOR/QOR uses link-neutral goal algorithm
    - Selects target AOR based on AOR load and health
    - Does not “prefer” local (= same LPAR) AORs
    - Even distribution across AORs, but ...
    - … responds to transient load/health variation
  - XCF MRO for “remote” STARTs or LINKs
    - High-performance System z sysplex technology
    - Uses coupling facility (CF) instead of TCP/IP stack
  - Sysplex-optimised workload routing
    - Highly responsive to transient variations
    - Uses CF to maintain current status for AORs
- Continuous operation and high availability through IBM MQ shared queues:
  - “Glitchless” recovery from region/LPAR/CEC outage
  - “Instant” redistribution of workload
  - In-flight messages backed-out, restart in another CICS region
- High throughput:
  - Exploits all available capacity
  - Highly responsive to transient spare capacity

MQ Workload Balance Summary

- MQ is a message delivery system, it does not try to balance workload
- Balancing the workload is attempting a technical solution for what is often a pricing problem
  - Beware spending a lot of effort for a solution to a temporary problem as well!
  - Turning off performance improvements like put to waiting getter will impact all applications, not just the skewed ones
- There are some mitigation techniques that can help the overall environment
  - Clustering!
  - Gateway queue managers
  - Using CPSM to make appropriate routing decisions

Additional Resources

- The following links are to additional information about IBM MQ
  - Queue Sharing Groups:
    https://www.ibm.com/docs/en/ibm-mq/9.2?topic=groups-what-is-queue-sharing-group
  - Clustering:
    https://www.ibm.com/docs/en/ibm-mq/9.2?topic=explorer-queue-manager-clusters
  - Shared Channels:
    https://www.ibm.com/docs/en/ibm-mq/9.2?topic=groups-shared-channels
  - Redbooks:
    - IBM WebSphere MQ V7.1 and V7.5 Features and Enhancements
      http://www.redbooks.ibm.com/abstracts/sg248087.html?Open
    - High Availability in WebSphere Messaging Solutions
      http://www.redbooks.ibm.com/abstracts/sg247839.html?Open
    - First ‘Redpiece’ about Shared queues
      http://www.redbooks.ibm.com/abstracts/sg246864.html?Open
  - Lyn’s first YouTube video:
    http://www.youtube.com/playlist?list=PL9N7JP2yU3T8JycrCOvEPM8c-0UdE97VT