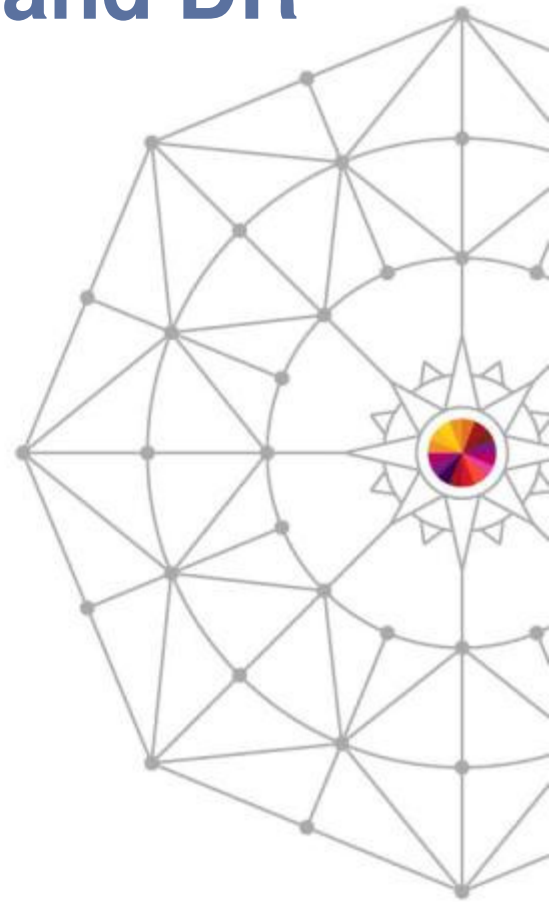
#SHAREorg

SHARE

15026: WebSphere MQ HA and DR



Ralph Bateman
ralph@uk.ibm.com

STSM Connectivity and Integration
Customer Support.

IBM Hursley

SHARE in Anaheim

Please Note

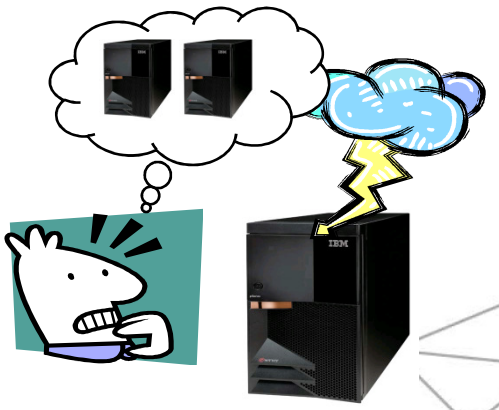
IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

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.

Introduction



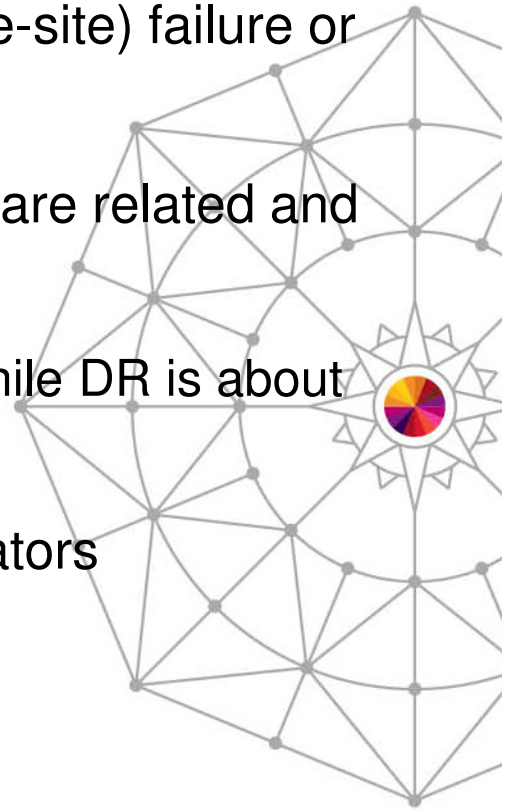
- Availability is a very large subject
- You can have the best technology in the world, but you have to manage it correctly
- Technology is not a substitute for good planning and testing!
- HA is not the same as DR.

What is DR – Wikipedia Version

- Disaster recovery is the process, policies and procedures related to preparing for recovery or continuation of technology infrastructure critical to an organization after a natural or human-induced disaster.
- Disaster recovery is a subset of business continuity. While business continuity involves planning for keeping all aspects of a business functioning in the midst of disruptive events, disaster recovery focuses on the IT or technology systems that support business functions.

What is DR

- Getting applications running after a major (often whole-site) failure or loss
- It is not about High Availability although often the two are related and share design and implementation choices
  - "HA is having 2, DR is having them a long way apart"
  - More seriously, HA is about keeping things running, while DR is about recovering when HA has failed.
- Requirements driven by business, and often by regulators
  - Data integrity, timescales, geography …
- One major decision point: cost
  - How much does DR cost you, even if it's never used?
  - How much are you prepared to lose



Disaster Recovery vs High Availability

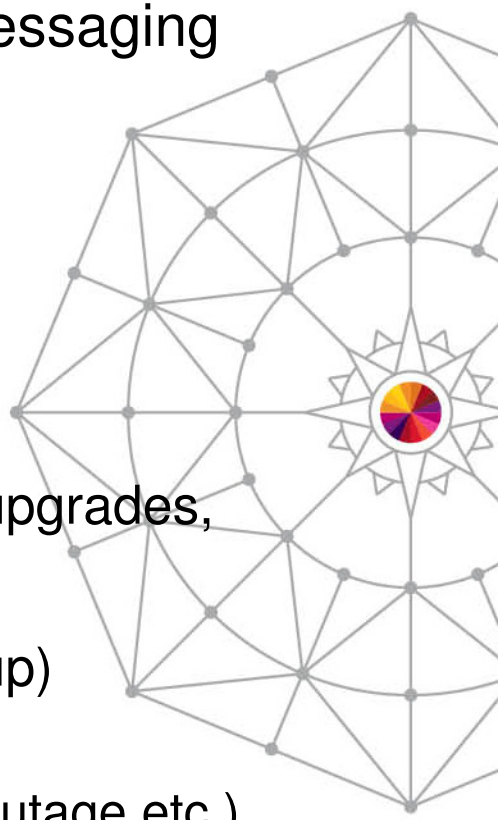
- Designs for HA typically involve a single site for each component of the overall architecture
- Designs for DR typically involve separate sites

- Designs for HA typically require no data loss
- Designs for DR typically can have limited data loss

- Designs for HA typically involve high-speed takeover
- Designs for DR typically can permit several hours down-time

Availability objective

- The objective is to achieve 24x7 availability of messaging
- Not always achievable, but we can get close
  - 99.9% availability = 8.76 hours downtime/year
  - 99.999% = 5 minutes
  - 99.9999% = 30 seconds
- Potential outage types:
  - 80% scheduled downtime (new software release, upgrades, maintenance)
  - 20% unscheduled downtime (source: Gartner Group)
    - 40% operator error
    - 40% application error
    - 20% other (network failures, disk crashes, power outage etc.)



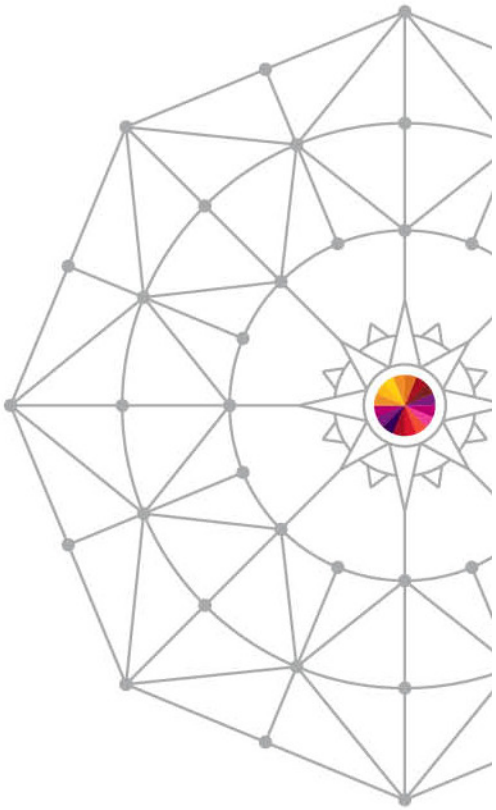
HIGH AVAILABILITY

Single Points of Failure

- With no redundancy or fault tolerance, a failure of any component can lead to a loss of availability
- Every component is critical. The system relies on the:
  - Power supply, system unit, CPU, memory
  - Disk controller, disks, network adapter, network cable
  - ...and so on
- Various techniques have been developed to tolerate failures:
  - UPS or dual supplies for power loss
  - RAID for disk failure
  - Fault-tolerant architectures for CPU/memory failure
  - ...etc
- Elimination of SPOFs is important to achieve HA

WebSphere MQ HA technologies

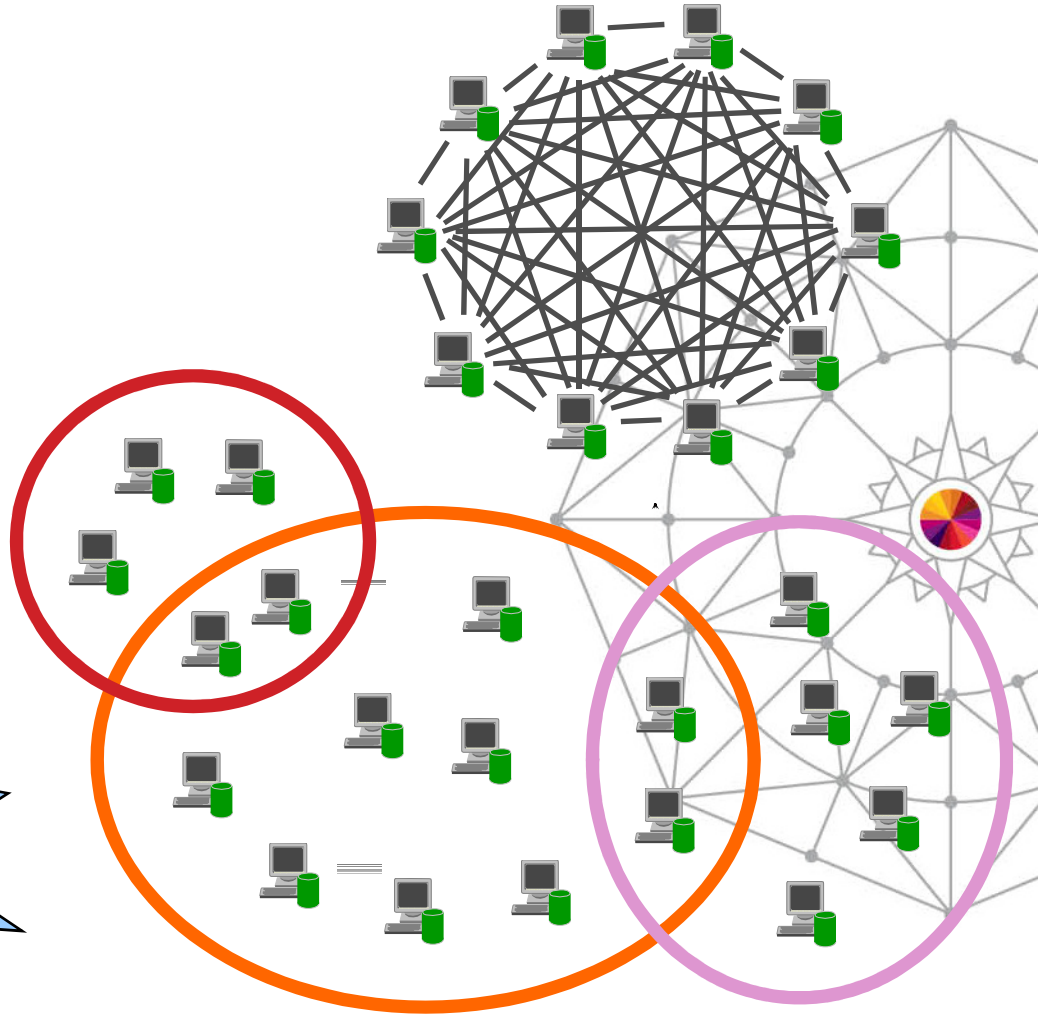
- Queue manager clusters
- Queue-sharing groups
- Support for networked storage
- Multi-instance queue managers
- HA clusters
- Client reconnection



Queue Manager Clusters

- Sharing cluster queues on multiple queue managers prevents a queue from being a SPOF

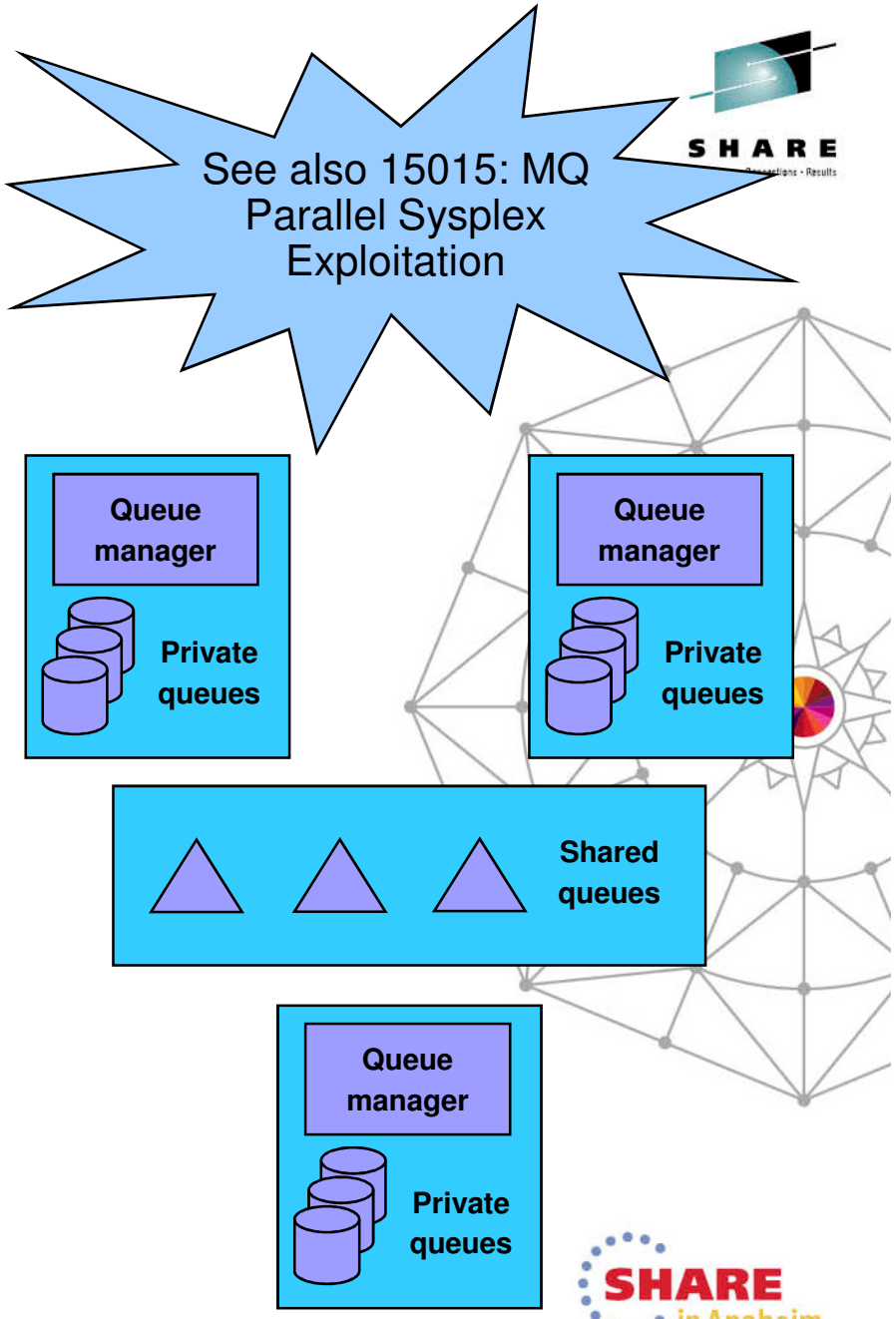
- Cluster workload algorithm automatically routes traffic away from failed queue managers



See also
15016: MQ
Clustering
On Thursday

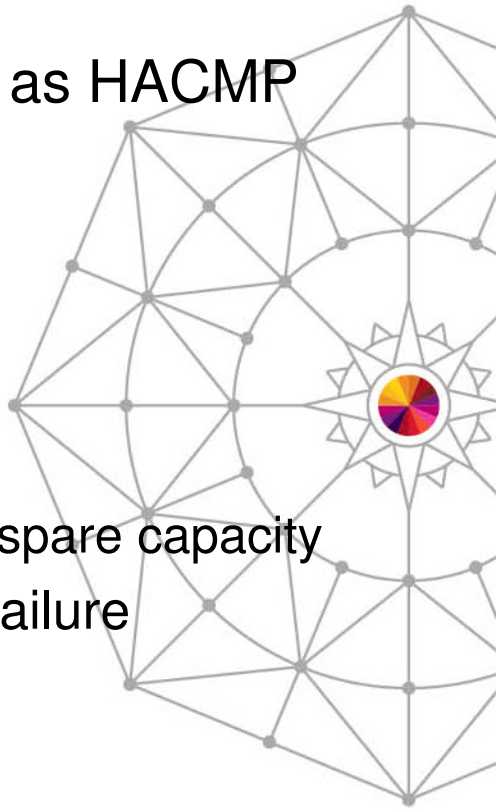
Queue-Sharing Groups

- On z/OS, queue managers can be members of a queue-sharing group
- Shared queues are held in a coupling facility
  - All queue managers in the QSG can access the messages
- Benefits:
  - Messages remain available even if a queue manager fails
  - Pull workload balancing
  - Apps can connect to the group



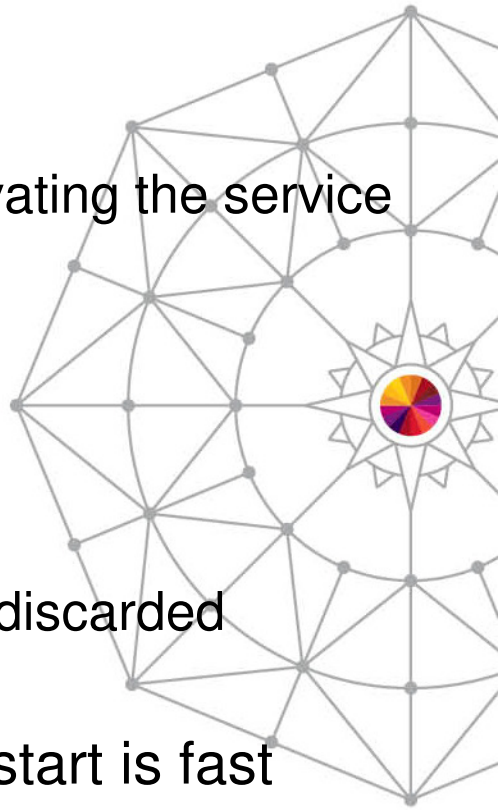
Introduction to Failover and MQ

- Failover is the automatic switching of availability of a service
  - For MQ, the “service” is a queue manager
- Traditionally the presense of an HA cluster, such as HACMP
- Requires:
  - Data accessible on all servers
  - Equivalent or at least compatible servers
    - Common software levels and environment
  - Sufficient capacity to handle workload after failure
    - Workload may be rebalanced after failover requiring spare capacity
  - Startup processing of queue manager following the failure
- MQ offers two ways of configuring for failover:
  - Multi-instance queue managers
  - HA clusters



Failover considerations

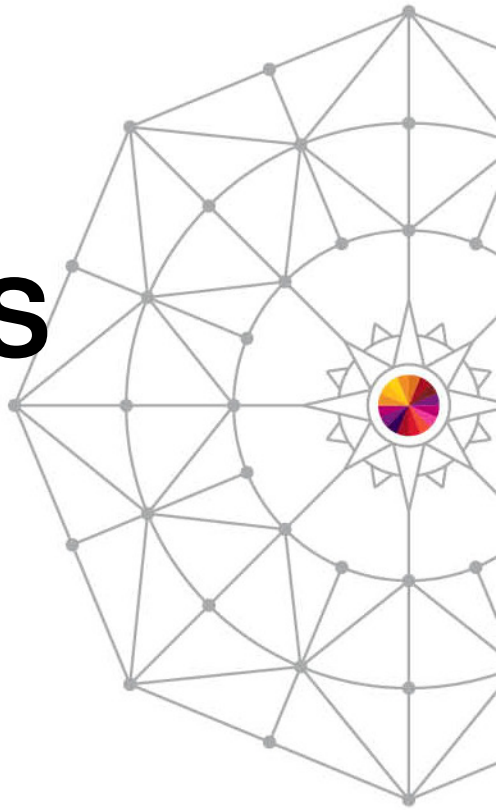
- Failover times are made up of three parts:
  - Time taken to notice the failure
    - Heartbeat missed
    - Bad result from status query
  - Time taken to establish the environment before activating the service
    - Switching IP addresses and disks, and so on
  - Time taken to activate the service
    - This is queue manager restart



- Failover involves a queue manager restart
  - Nonpersistent messages, nondurable subscriptions discarded

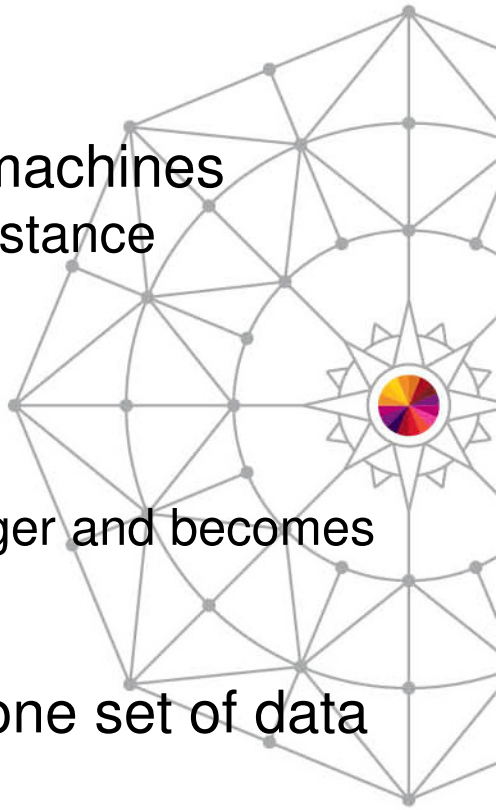
- For fastest times, ensure that queue manager restart is fast
  - No long running transactions, for example

MULTI-INSTANCE QUEUE MANAGERS



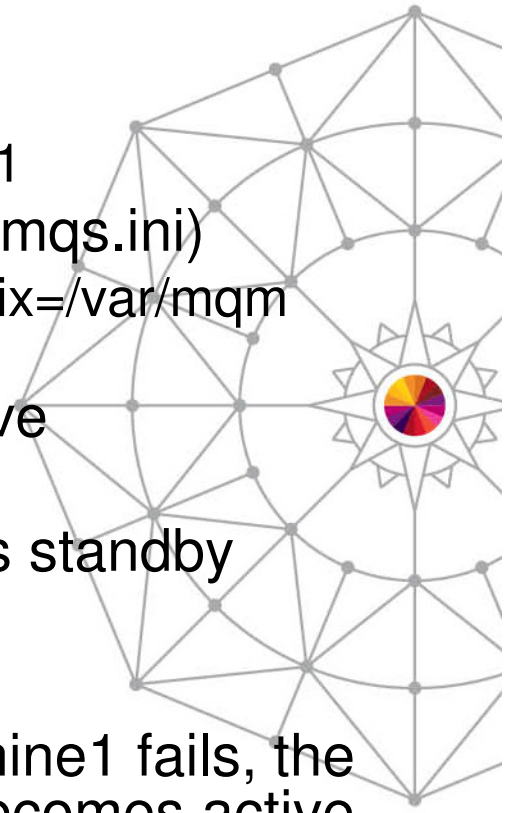
Multi-instance Queue Managers

- Basic failover support without HA cluster
- Two instances of a queue manager on different machines
  - One is the “active” instance, other is the “standby” instance
  - Active instance “owns” the queue manager’s files
    - Accepts connections from applications
  - Standby instance monitors the active instance
    - Applications cannot connect to the standby instance
    - If active instance fails, standby restarts queue manager and becomes active
- Instances are the SAME queue manager – only one set of data files
  - Queue manager data is held in networked storage



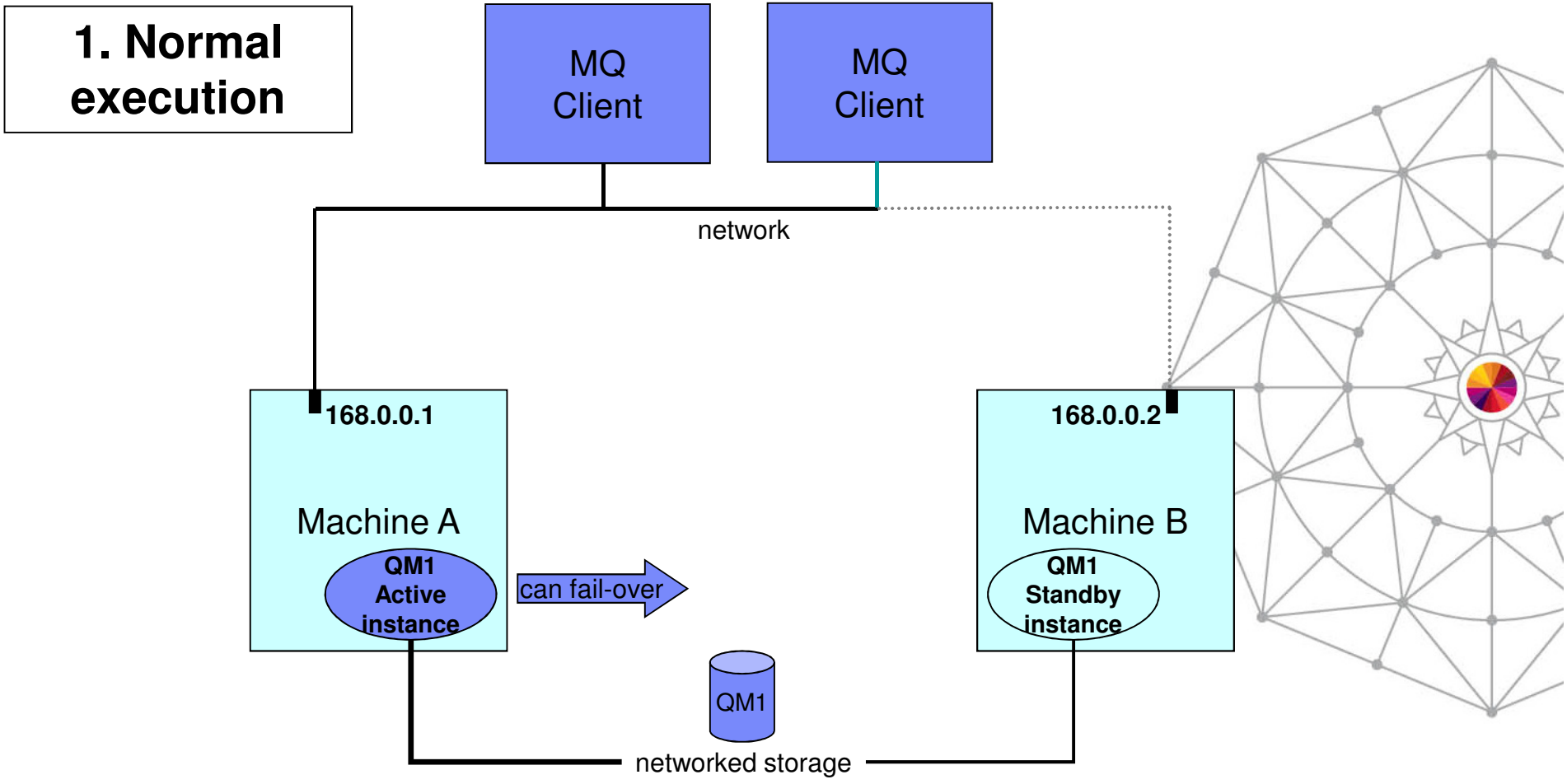
Setting up Multi-instance Queue Manager

- Set up shared filesystems for QM data and logs
- Create the queue manager on machine1
  - crtmqm –md /shared/qmdata –ld /shared/qmlog QM1
- Define the queue manager on machine2 (or edit mqs.ini)
  - addmqinf –v Name=QM1 –v Directory=QM1 –v Prefix=/var/mqm -v DataPath=/shared/qmdata/QM1
- Start an instance on machine1 – it becomes active
  - strmqm –x QM1
- Start another instance on machine2 – it becomes standby
  - strmqm –x QM1



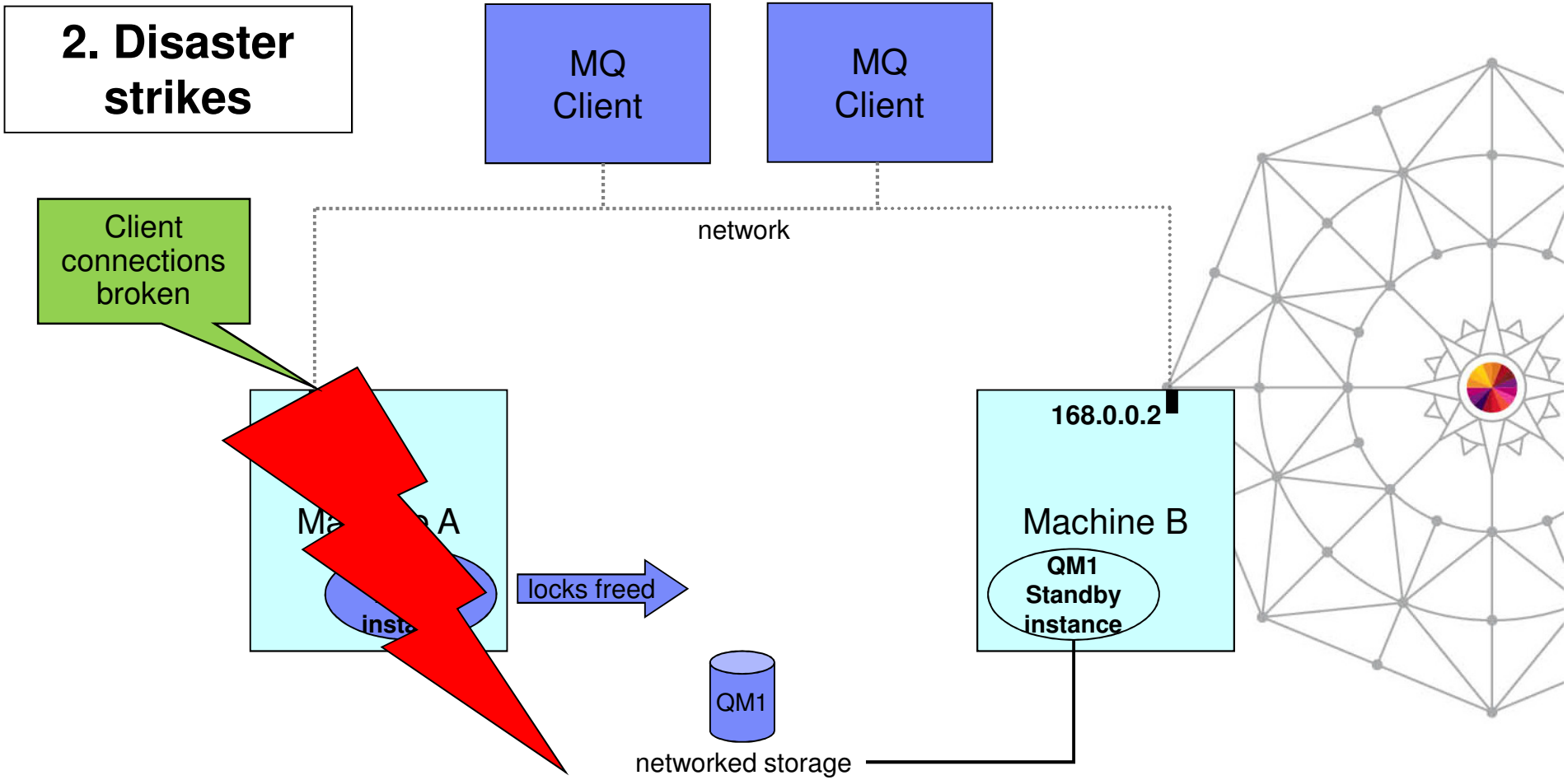
- That's it. If the queue manager instance on machine1 fails, the standby instance on machine2 takes over and becomes active

Multi-instance Queue Managers



Owns the queue manager data

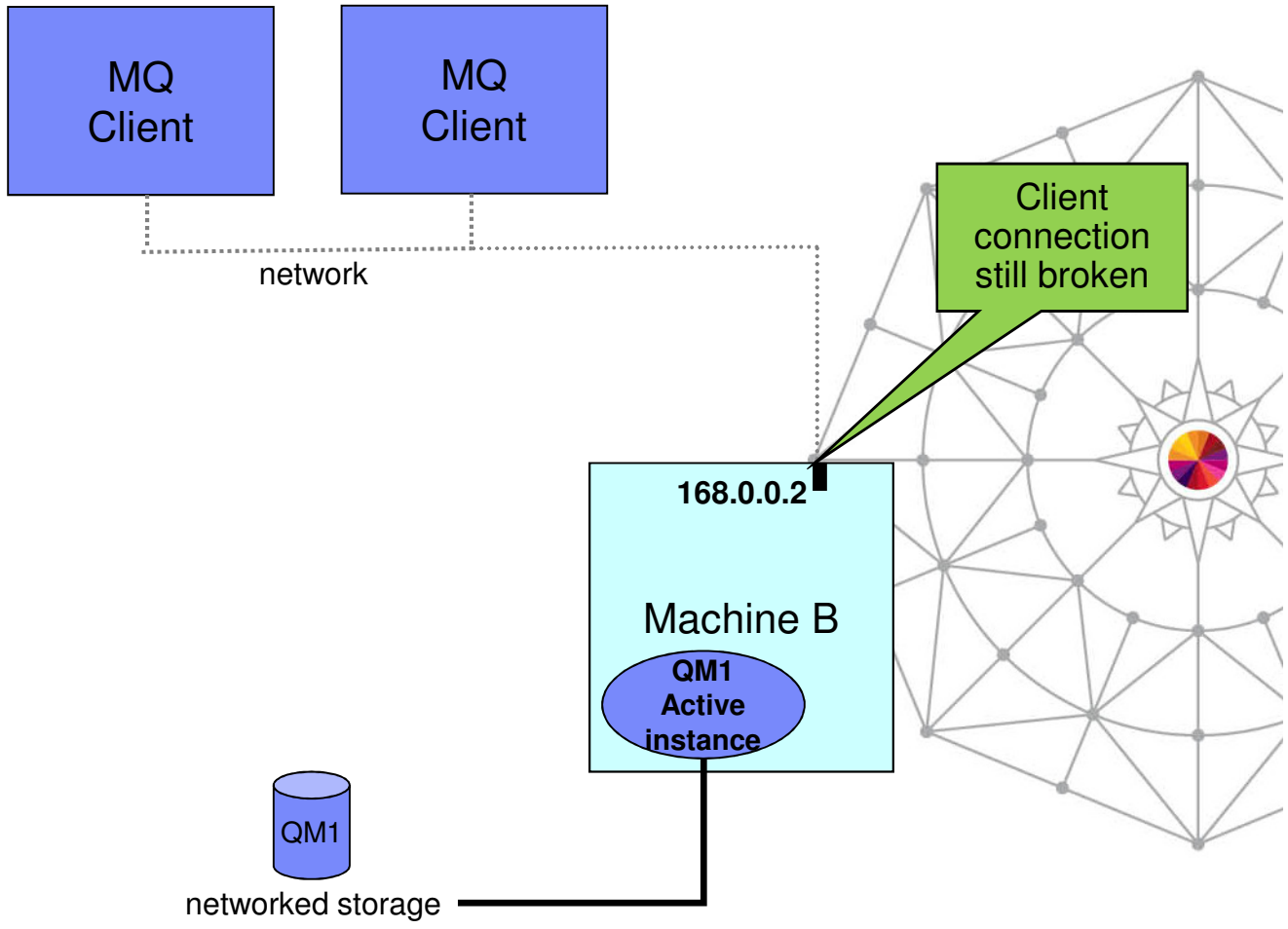
Multi-instance Queue Managers



Multi-instance Queue Managers

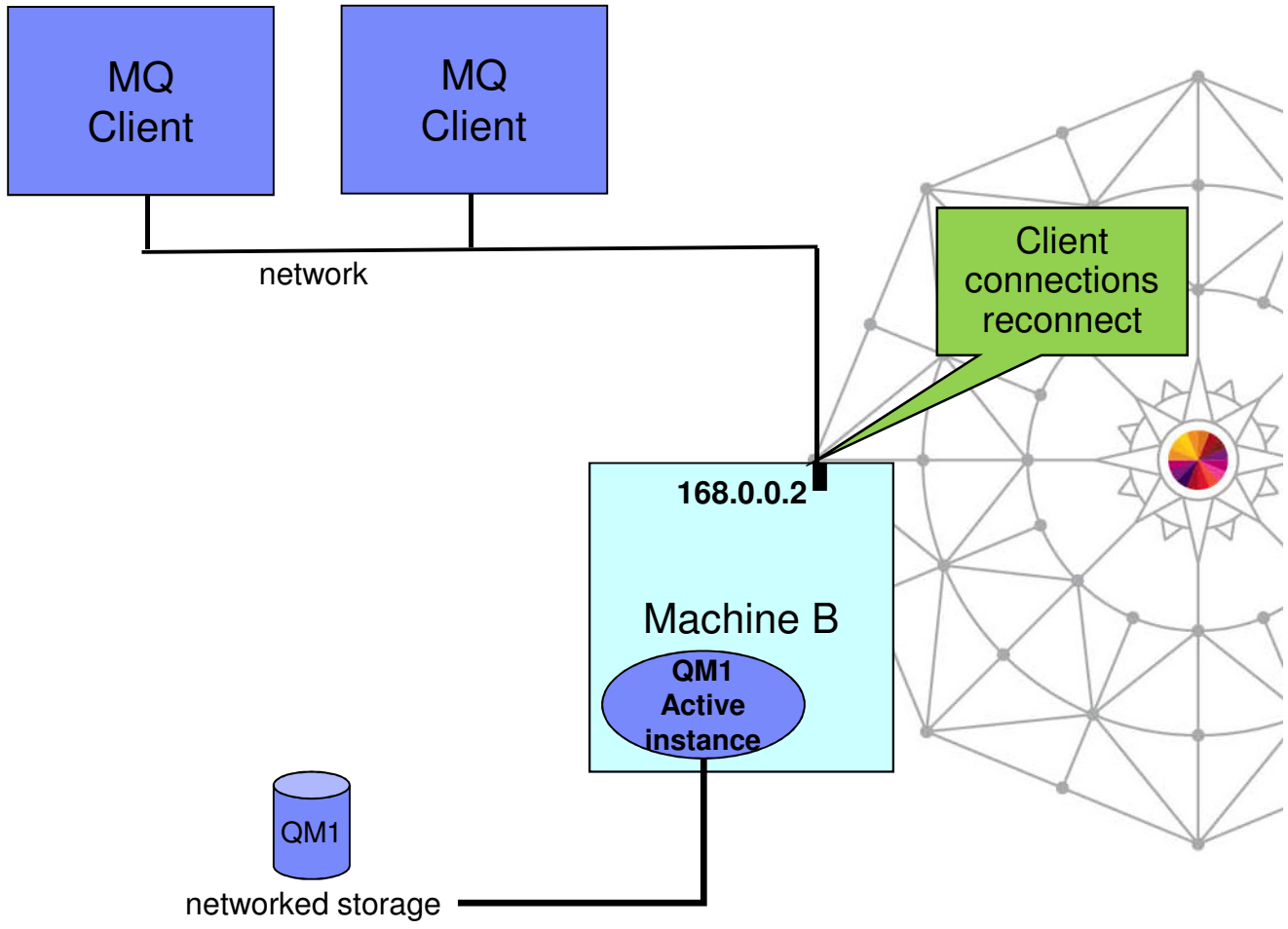
3. FAILOVER

Standby becomes active



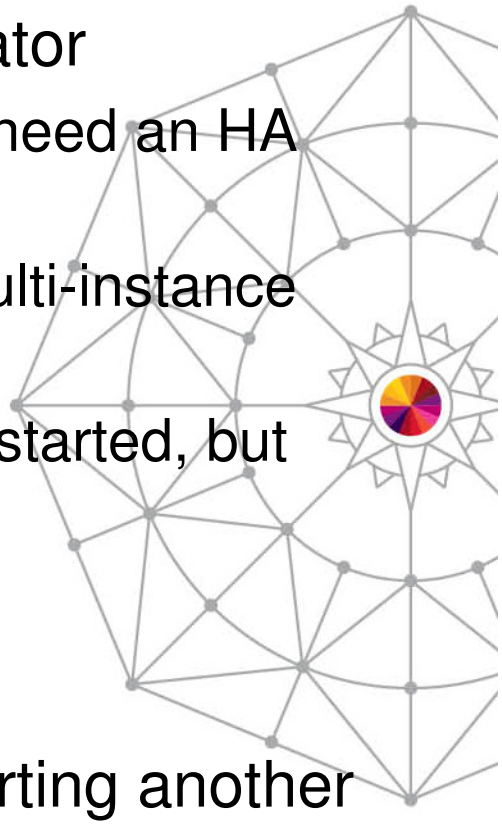
Multi-instance Queue Managers

4. Recovery complete



Multi-instance Queue Managers

- MQ is NOT becoming an HA cluster coordinator
  - If other resources need to be coordinated, you need an HA cluster
  - WebSphere Message Broker integrates with multi-instance QM
  - Queue manager services can be automatically started, but with limited control



- System administrator is responsible for restarting another standby instance when failover has occurred

Dealing with multiple IP addresses

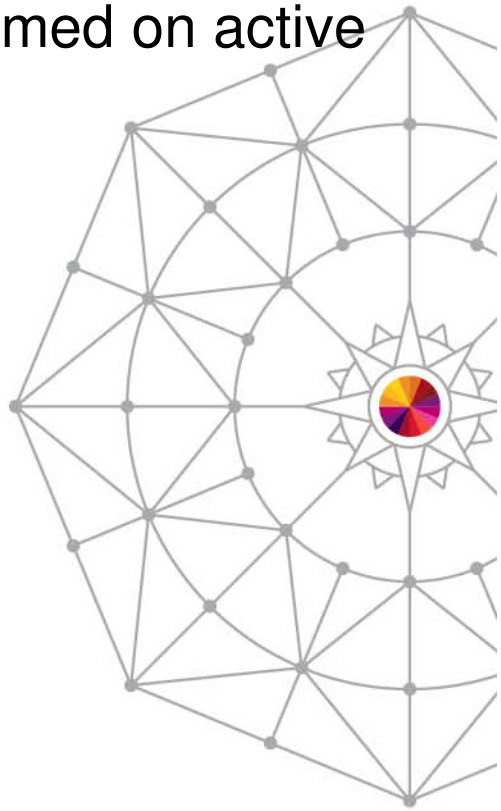
- The IP address of the queue manager changes when it moves
  - So MQ channel configuration needs way to select address
- Connection name syntax extended to a comma-separated list
  - CONNAME('168.0.0.1,168.0.0.2')
  - Needs 7.0.1 qmgr or client
- Unless you use external IPAT or an intelligent router or MR01
- WAS8 admin panels understand this syntax.
- For earlier levels of WAS
  - Connection Factories:
    - Set a custom property called XMSC_WMQ_CONNECTION_NAME_LIST to the list of host/port names that you wish to connect to
    - Make sure that the existing host and port values defined on the connection factory match the first entry in this property
  - Activation Specs:
    - Set a custom property called connectionNameList on the activation spec with the same format

Administering Multi-instance QMgrs

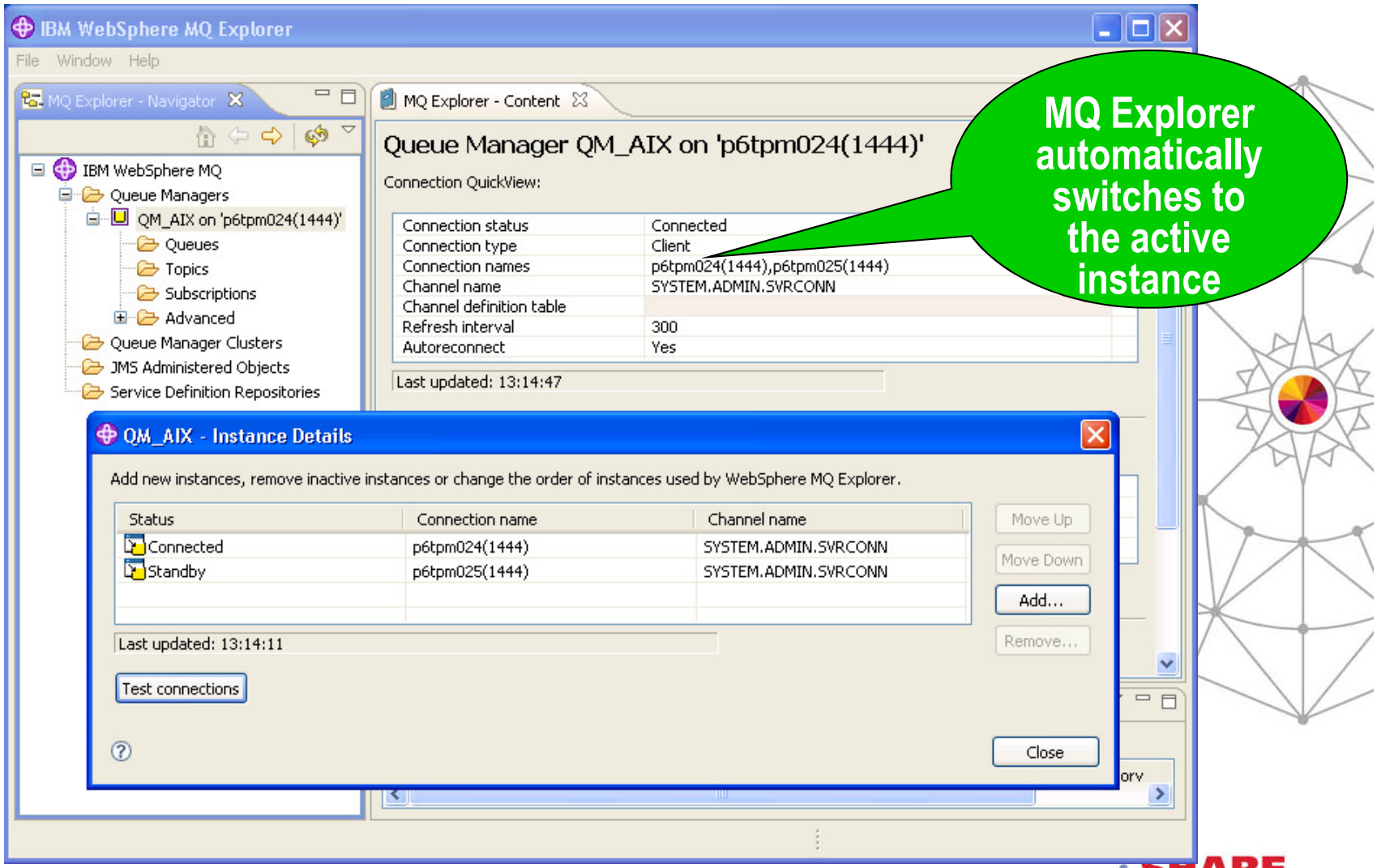
- All queue manager administration must be performed on active instance
- dspmq enhanced to display instance information

```
$ hostname
england
$ dspmq -x
QMNAME(MIQM)          STATUS(Running as standby)
    INSTANCE(wales)       MODE(Active)
    INSTANCE(england)     MODE(Standby)
```

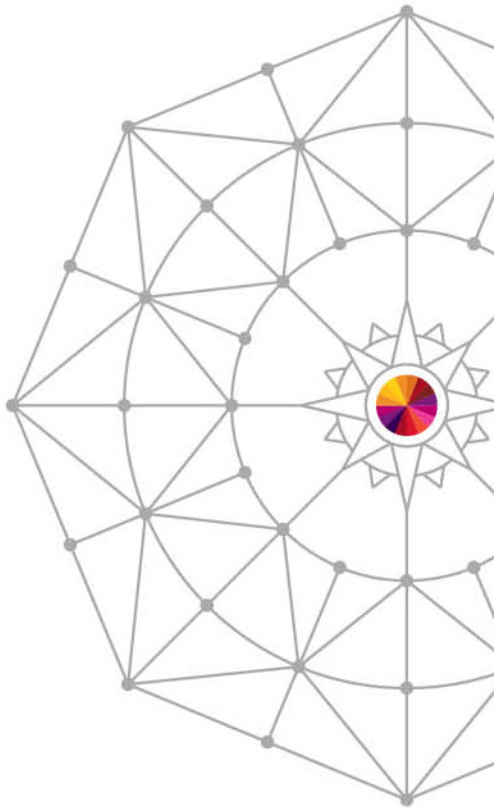
  - dspmq issued on “england”
  - On “england”, there’s a standby instance
  - The active instance is on “wales”



Multi-instance QMgr in MQ Explorer

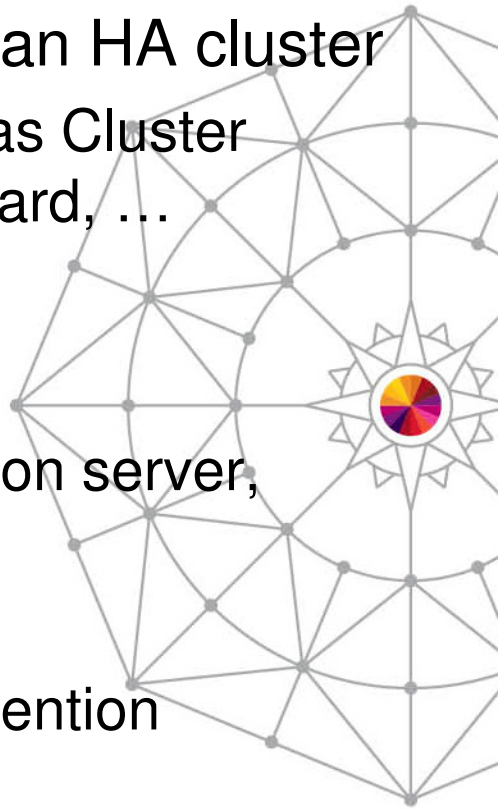


HA CLUSTERS



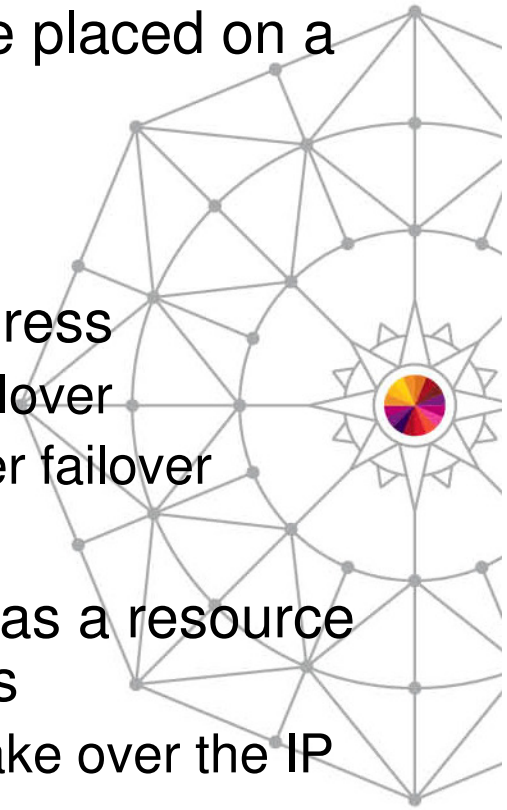
HA clusters

- MQ traditionally made highly available using an HA cluster
  - IBM PowerHA for AIX (formerly HACMP), Veritas Cluster Server, Microsoft Cluster Server, HP Serviceguard, …
- HA clusters can:
  - Coordinate multiple resources such as application server, database
  - Consist of more than two machines
  - Failover more than once without operator intervention
  - Takeover IP address as part of failover
  - Likely to be more resilient in cases of MQ and OS defects

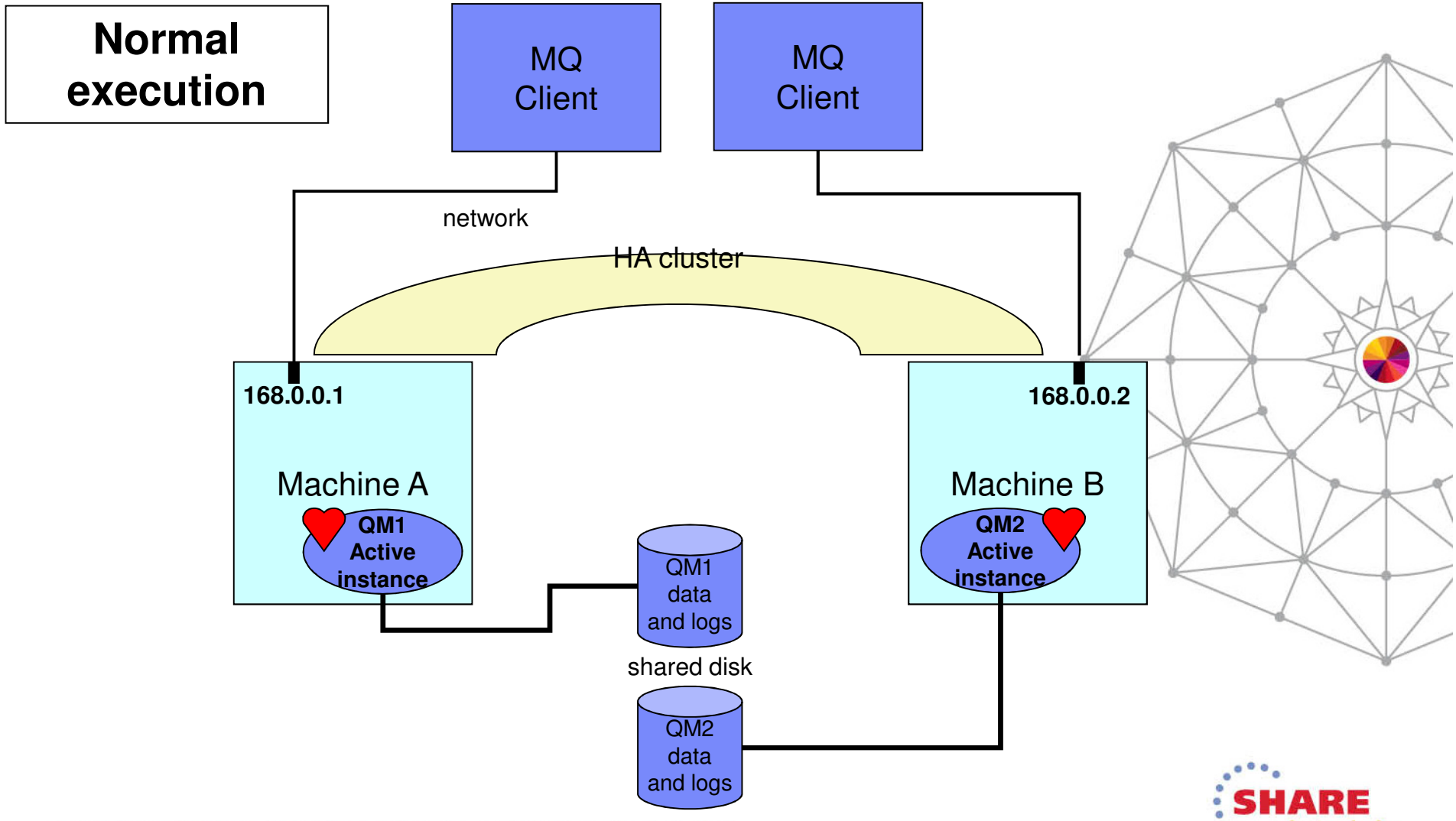


HA clusters

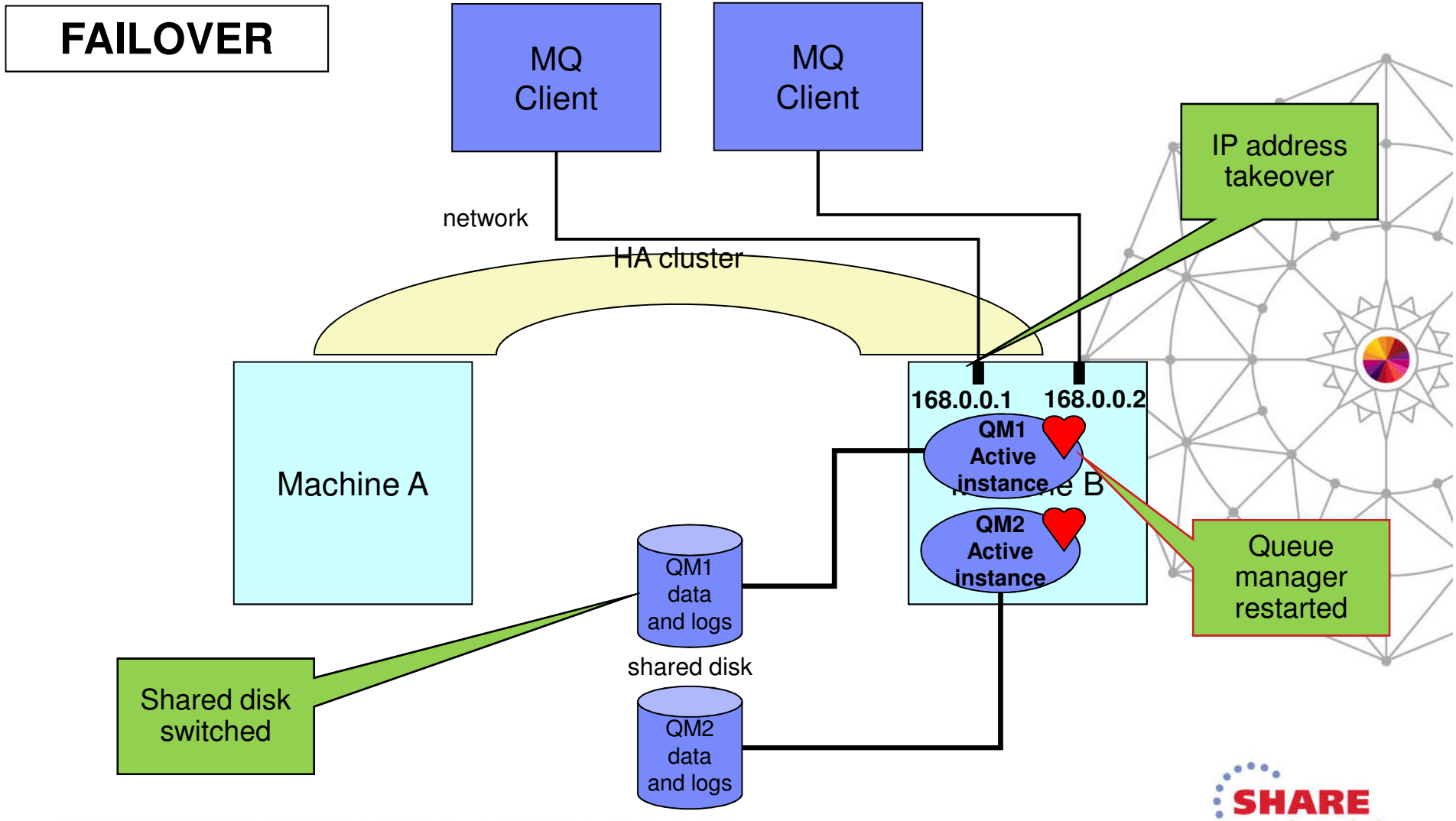
- In HA clusters, queue manager data and logs are placed on a shared disk
  - Disk is switched between machines during failover
- The queue manager has its own “service” IP address
  - IP address is switched between machines during failover
  - Queue manager’s IP address remains the same after failover
- The queue manager is defined to the HA cluster as a resource dependent on the shared disk and the IP address
  - During failover, the HA cluster will switch the disk, take over the IP address and then start the queue manager



MQ in an HA cluster – Active/active

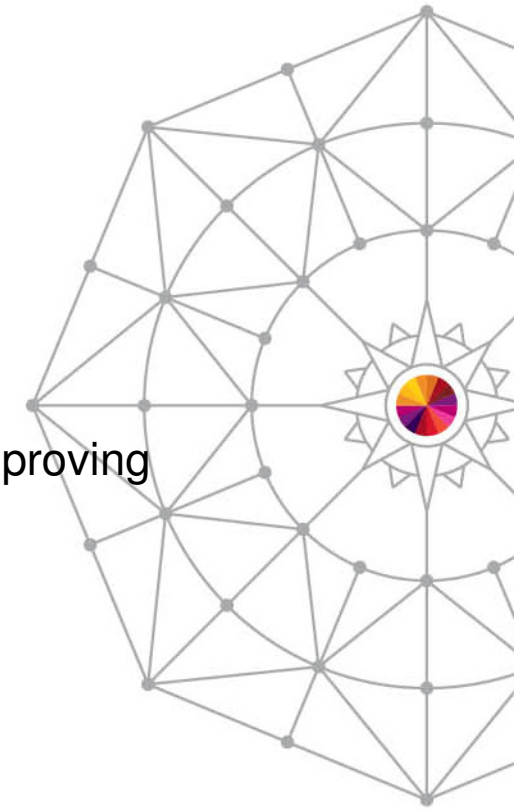


MQ in an HA cluster – Active/active



Multi-instance QM or HA cluster?

- Multi-instance queue manager
  - ✓ Integrated into the WebSphere MQ product
  - ✓ Faster failover than HA cluster
    - ✓ Delay before queue manager restart is much shorter
  - × Runtime performance of networked storage
  - × Suitable storage can sometimes be a challenge
- HA cluster
  - ✓ Capable of handling a wider range of failures
  - ✓ Failover historically rather slow, but some HA clusters are improving
  - ✓ Capable of more flexible configurations (eg N+1)
  - × Required MC91 SupportPac or equivalent configuration
  - × Extra product purchase and skills required



- Storage distinction
  - Multi-instance queue manager typically uses NAS
  - HA clustered queue manager typically uses SAN

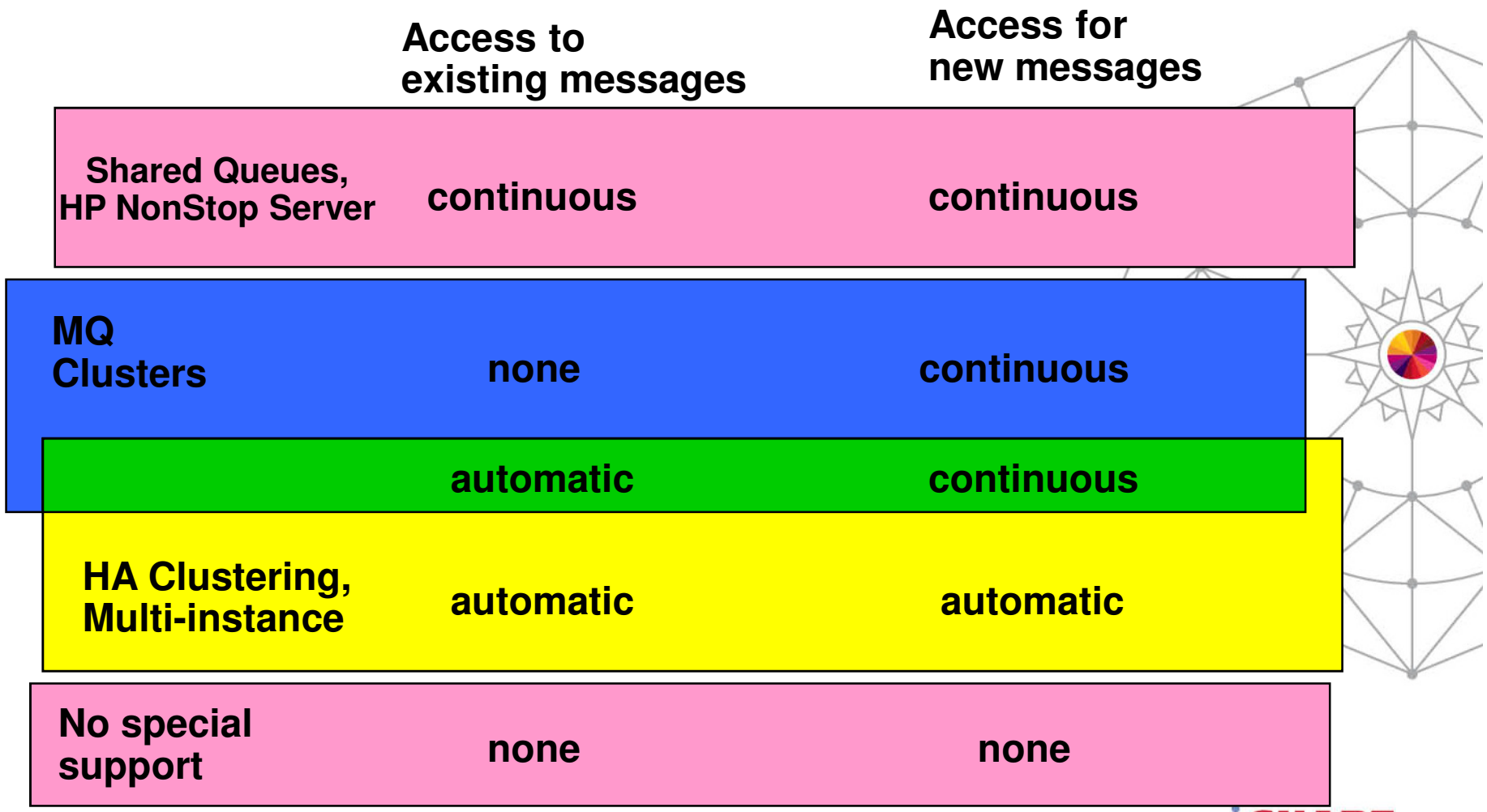
Creating QM for failover

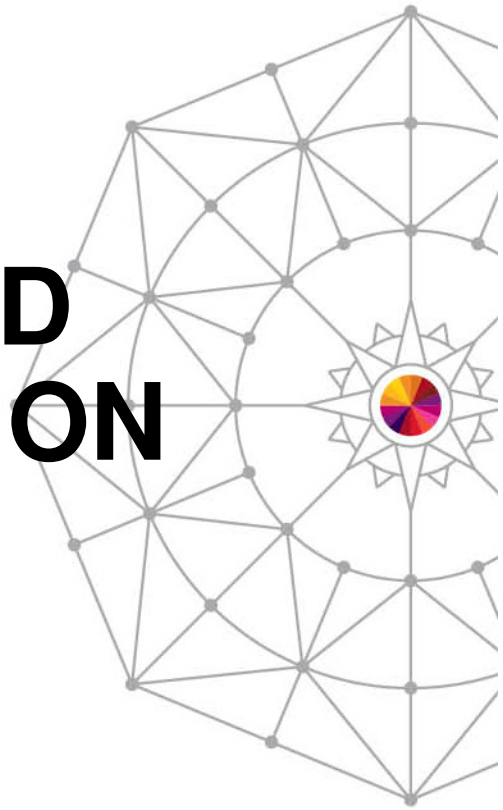
- Create filesystems on the shared disk, for example
  - /MQHA/QM1/data for the queue manager data
  - /MQHA/QM1/log for the queue manager logs

- On one of the nodes:
  - Mount the filesystems
  - Create the queue manager
    - crtmqm –md /MQHA/QM1/data –ld /MQHA/QM1/log QM1
  - Print out the configuration information for use on the other nodes
    - dspmqinf –o command QM1

- On the other nodes:
  - Mount the filesystems
  - Add the queue manager's configuration information
    - addmqinf –s QueueManager –v Name=QM1 –v Prefix=/var/mqm –v DataPath=/MQHA/QM1/data/QM1 –v Directory=QM1

Comparison of Technologies

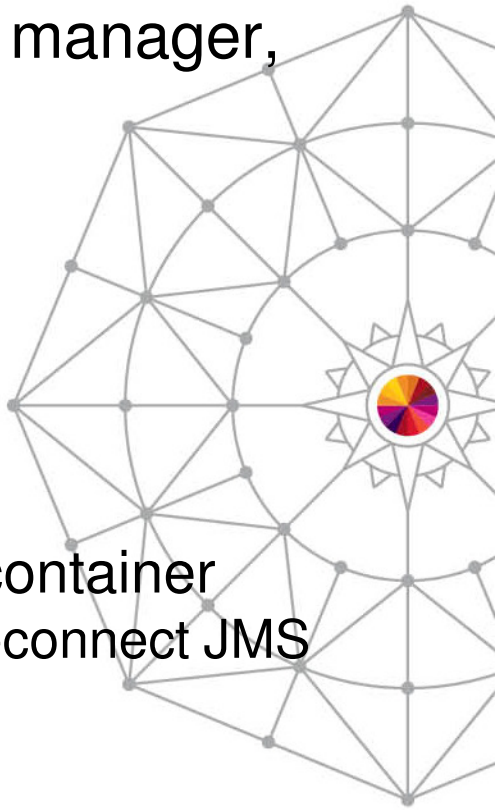




APPLICATIONS AND AUTO-RECONNECTION

HA applications – MQ connectivity

- If an application loses connection to a queue manager, what does it do?
  - End abnormally
  - Handle the failure and retry the connection
  - Reconnect automatically thanks to application container
    - WebSphere Application Server contains logic to reconnect JMS clients
  - Use MQ automatic client reconnection



Automatic client reconnection

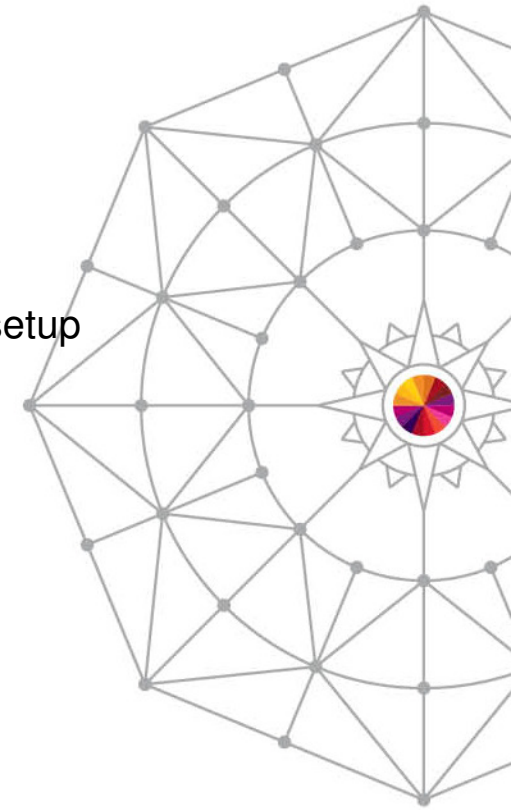
- MQ client automatically reconnects when connection broken
  - MQI C clients and standalone JMS clients
  - JMS in app servers (EJB, MDB) do not need auto-reconnect
- Reconnection includes reopening queues, remaking subscriptions
  - All MQI handles keep their original values
- Can reconnect to same queue manager or another, equivalent queue manager
- MQI or JMS calls block until connection is remade
  - By default, will wait for up to 30 minutes
  - Long enough for a queue manager failover (even a really *slow* one)

Automatic client reconnection

- Can register event handler to observe reconnection
- Not all MQI is seamless, but majority repaired transparently
  - Browse cursors revert to the top of the queue
  - Nonpersistent messages are discarded during restart
  - Nondurable subscriptions are remade and may miss some messages
  - In-flight transactions backed out
- Tries to keep dynamic queues with same name
  - If queue manager doesn't restart, reconnecting client's TDQs are kept for a while in case it reconnects
  - If queue manager does restart, TDQs are recreated when it reconnects
- Requires:
  - Threaded client
  - 7.0.1 server – including z/OS
  - Full-duplex client communications (SHARECNV >= 1)

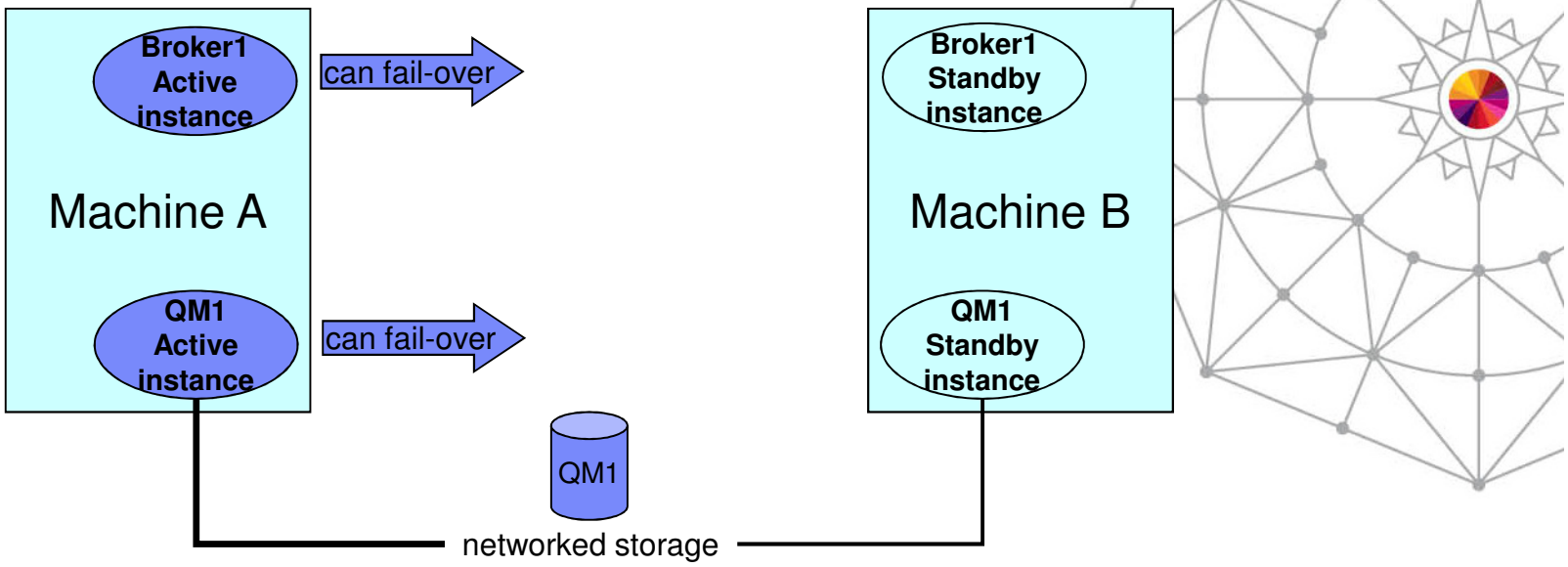
Client Configurations for Availability

- Use wildcarded queue manager names in CCDT
  - Gets weighted distribution of connections
  - Selects a “random” queue manager from an equivalent set
- Use multiple addresses in a CONNAME
  - Could potentially point at different queue managers
  - More likely pointing at the same queue manager in a multi-instance setup
- Use automatic reconnection
- Pre-connect Exit from V7.0.1.4
- Use IP routers to select address from a list
  - Based on workload or anything else known to the router
- Can use all of these in combination!



IBM Integration Bus – Multi-instance

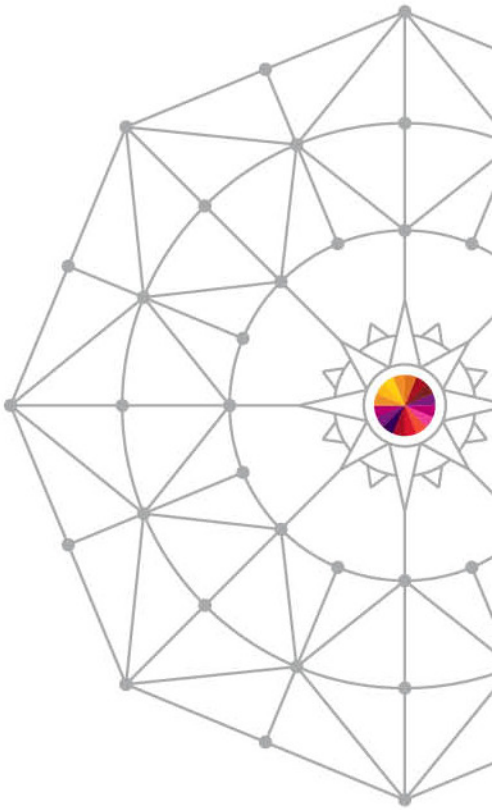
- IBM Integration Bus is designed to work well with multi-instance queue managers
  - Standby broker instance can be started on a standby QM instance
  - An alternative is to make the broker a queue-manager service



Owns the queue manager data

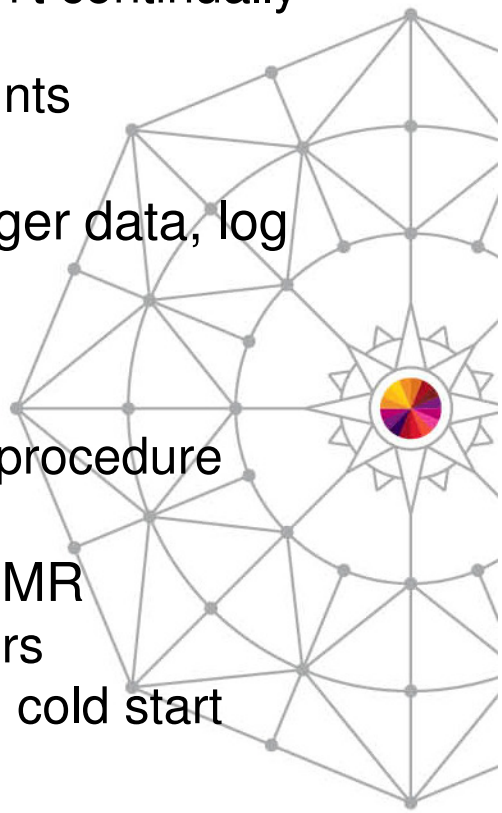
Disaster Recovery

Local Recovery



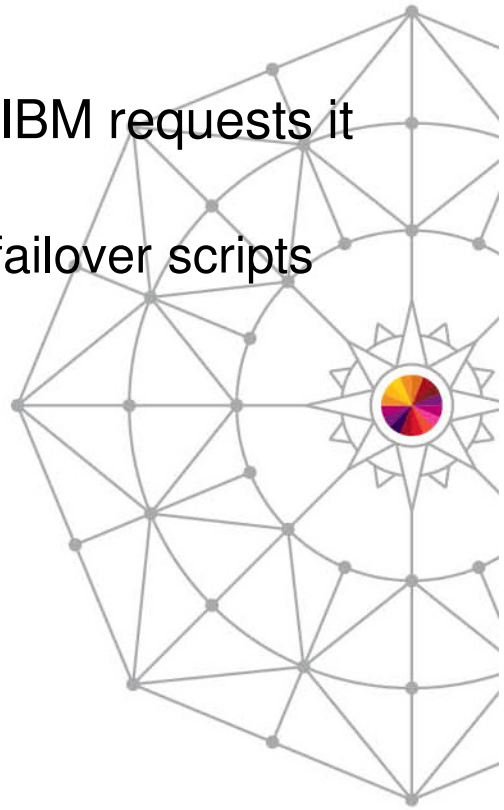
“MQ has failed”

- Don’t restart queue manager until you know why it failed
  - You can probably do one restart attempt safely, but don’t continually retry
  - When running under an HA coordinator, have retry counts
- At least ensure you take a backup of the queue manager data, log files, and any error logs/FDCs/dumps
  - So it can be investigated later
  - Might be possible for IBM to recover messages
  - Consider taking these copies as part of an HA failover procedure
- While trying restart/recovery procedures, consider a PMR
  - Often see “cold start” as first reaction at some customers
  - If you have a support contract, open PMR before trying cold start
    - IBM may have an alternative
    - IBM may ask you to start collecting documentation
  - Do not make everything a Sev1



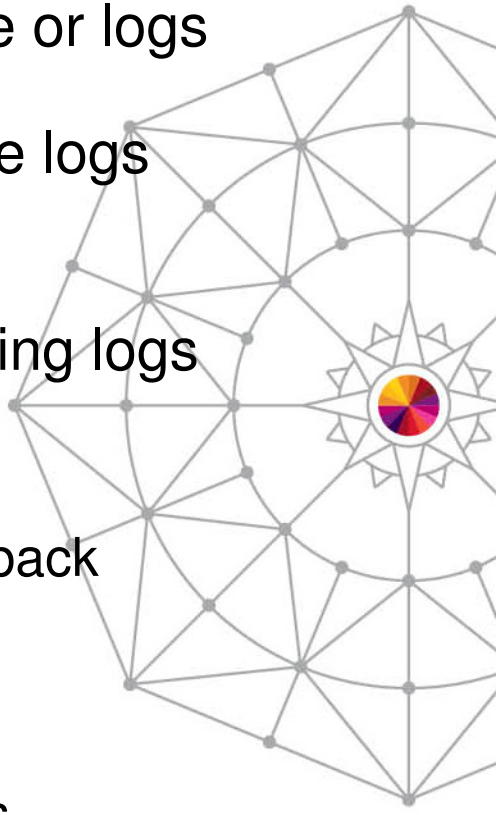
First Manual Restart

- Restart your queue manager
  - Only clean the IPC (shared memory/semaphores) if IBM requests it
    - This should never be necessary
    - Remove calls to ipcrm or amqiclen from any startup/failover scripts
  - Start as simply as possible
    - strmqm –ns QM1
  - Monitor the restart, look for FDC's
  - If OK, then end the qmgr and restart normally



- What if the restart fails?
  - Option to escalate to cold start
  - Further escalation to rebuilding queue manager

Cold Starting WMQ



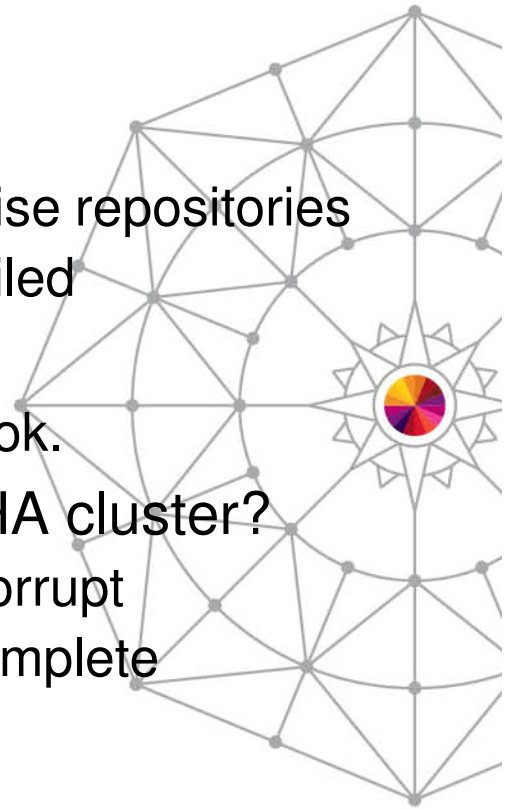
- Typical reason: hardware (most likely disk) failure or logs deleted by mistaken administrator
- Symptoms: Cannot start queue manager because logs unavailable or corrupt files

- "Cold start" is a technique to restart without needing logs

- What does a cold start cost you?
  - In-flight transactions will not be automatically rolled-back
  - In-doubt transactions will be forgotten
  - Ability to recover messages from the logs
  - Possible loss of messages
  - Possible duplication of already-processed messages

Cold Start QMGR

- Considerations
  - Is this queue manager part of a WMQ cluster?
    - This shouldn't matter, but may want to resynchronise repositories
    - In case any updates were in-flight when system failed
  - Is QMGR part of a Queue-Sharing group?
    - This shouldn't matter, but check that all connects ok.
  - Is this queue manager under the control of an HA cluster?
    - Failover will not help if the shared disks/files are corrupt
    - Disable failover in the HA system until recovery complete

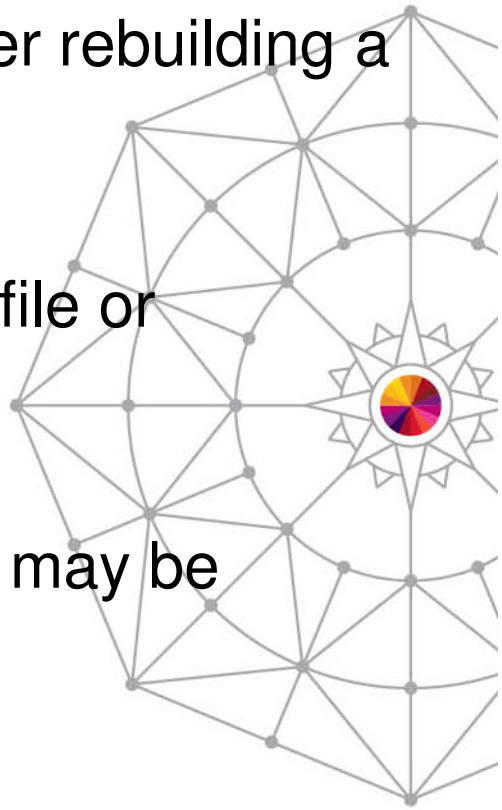


Rebuilding a Queue Manager

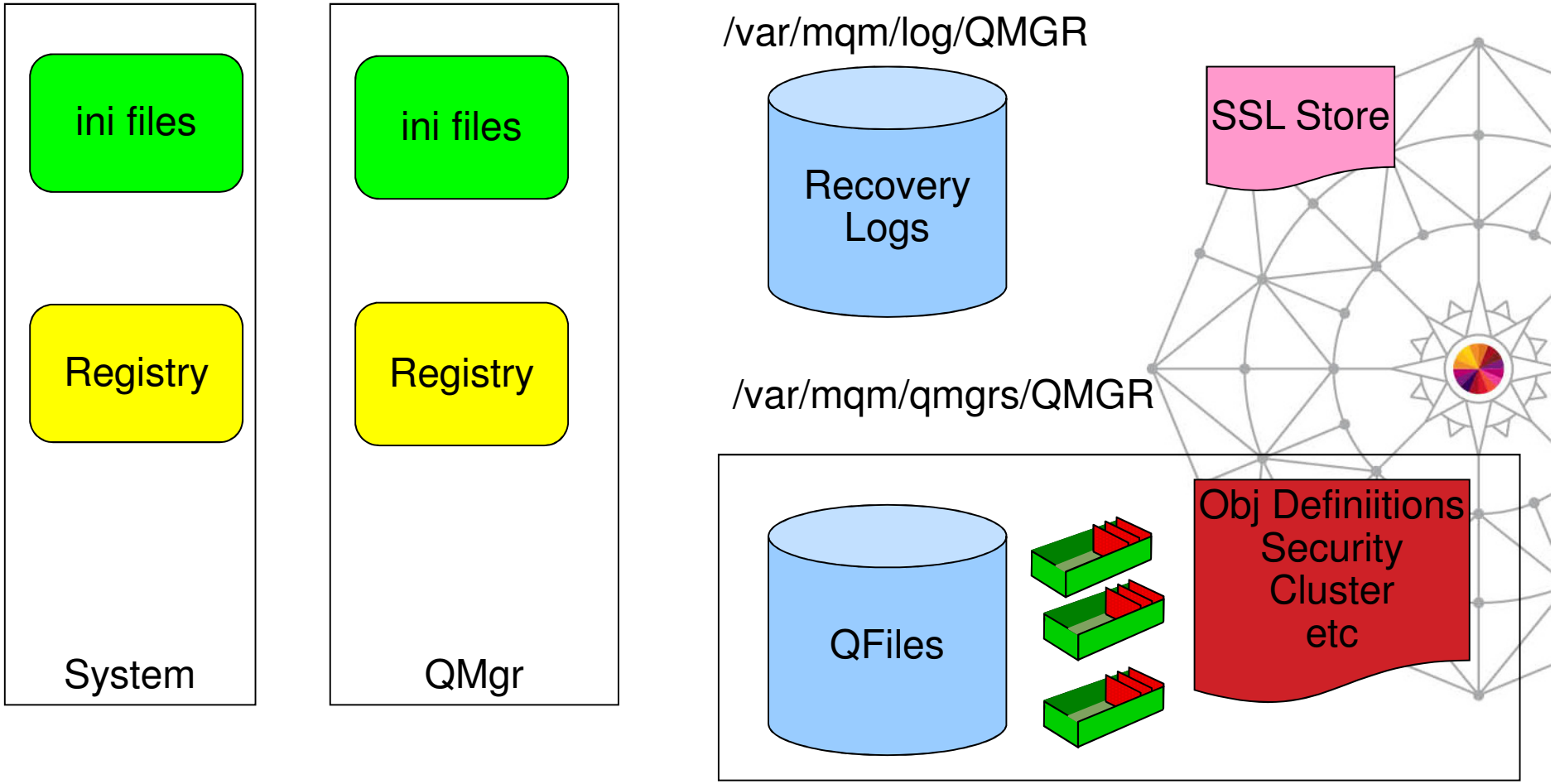
- A rebuild creates a replacement queue manager
  - Same object definitions
  - But loss of message data and channel sequence numbers
- Replacement queue manager has a new QMID
  - MQ Explorer saves QMID in its list of known queue managers
  - Will allow you to connect, but requires confirmation that the new qmid is expected
- Recommend issuing RESET CLUSTER at full repository to remove the old QMID before bringing the replacement online

Recovering Messages

- It might be possible to recover messages after rebuilding a queue manager
- While queue manager is stopped, copy the qfile or pagesets from the damaged system
- No guarantees, and transactional operations may be inconsistent
  - But it might be good enough



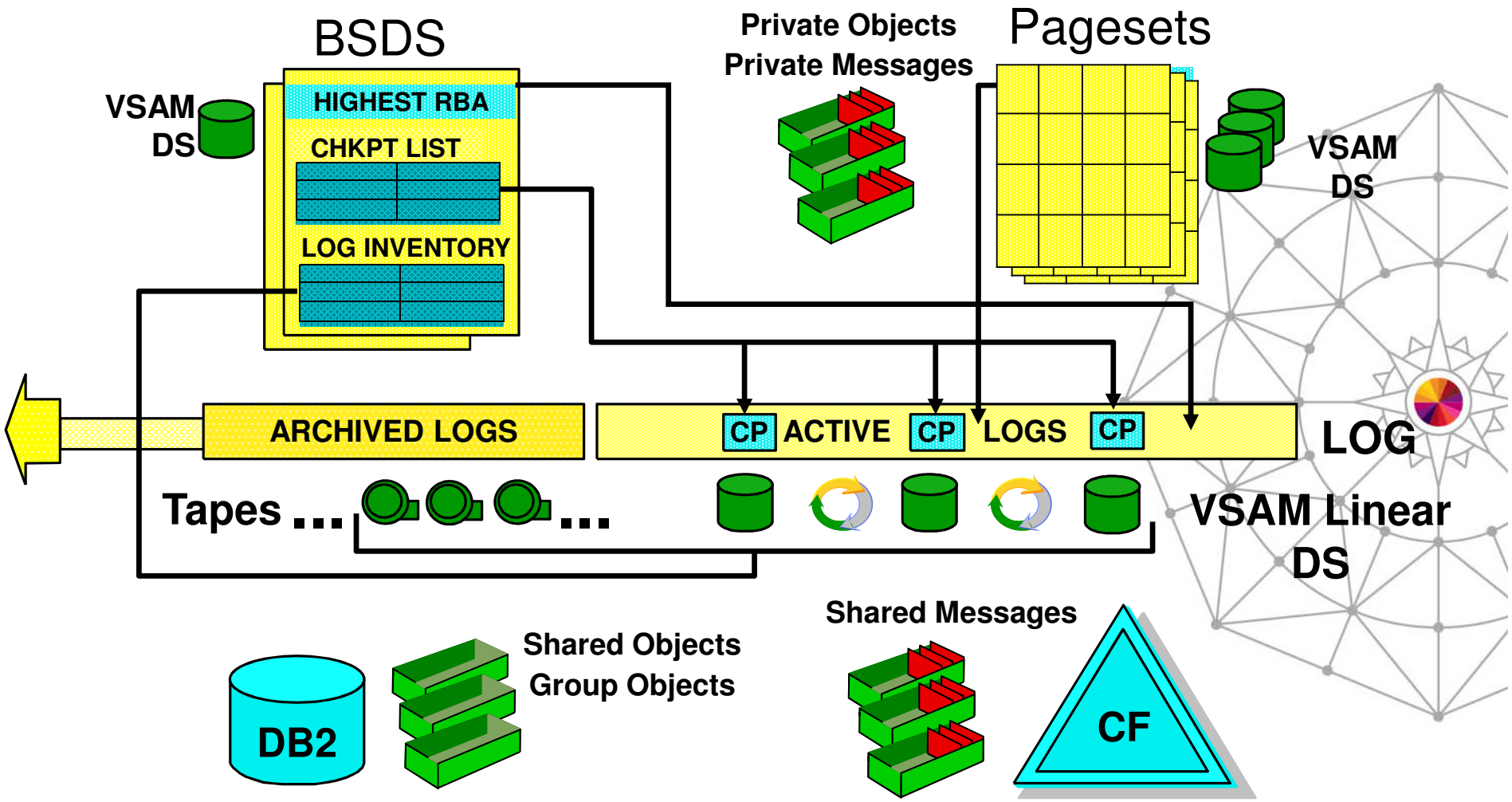
What makes a Queue Manager on Dist?



Backups

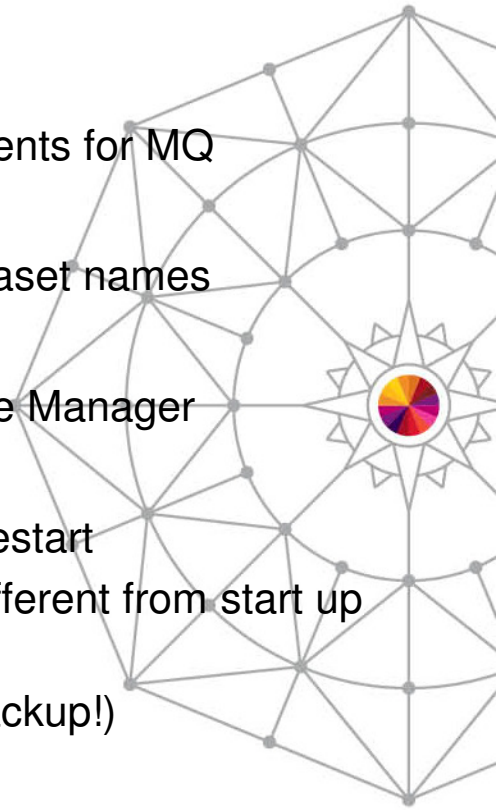
- At minimum, backup definitions at regular intervals
  - Include ini files and security settings
- One view is there is no point to backing up messages
  - They will be obsolete if they ever need to be restored
  - Distributed platforms – data backup only possible when qmgr stopped
- Use rcdmqimg on Distributed platforms to take images
  - Channel sync information is recovered even for circular logs
- Backup everything before upgrading code levels
  - On Distributed, you cannot go back
- Exclude queue manager data from normal system backups
  - Some backup products interfere with WMQ processing

What makes a Queue Manager on z/OS?



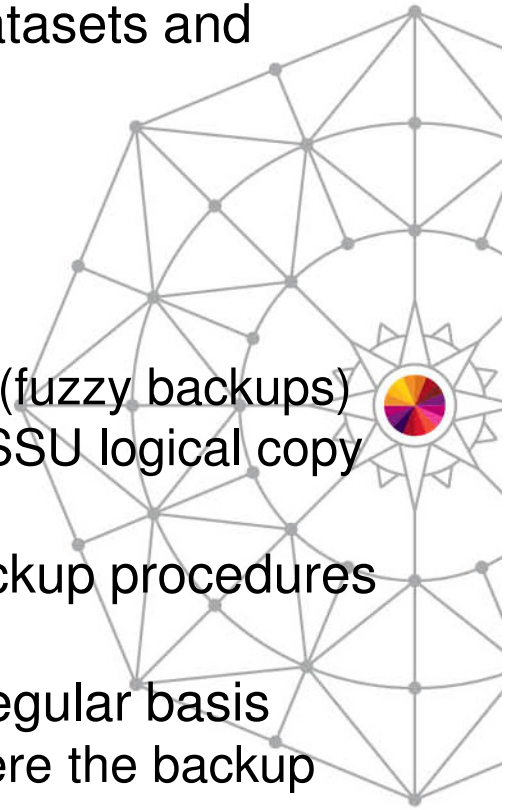
What makes up a Queue Manager?

- Queue manager started task procedure
  - Specifies MQ libraries to use, location of BSDS and pagesets and INP1, INP2 members start up processing
- System Parameter Module – zParm
  - Configuration settings for logging, trace and connection environments for MQ
- BSDS: Vital for Queue Manager start up
  - Contains info about log RBAs, checkpoint information and log dataset names
- Active and Archive Logs: Vital for Queue Manager start up
  - Contain records of all recoverable activity performed by the Queue Manager
- Pagesets
  - Updates made “lazily” and brought “up to date” from logs during restart
  - Start up with an old pageset (restored backup) is not really any different from start up after queue manager failure
  - Backup needs to copy page 0 of pageset first (don’t do volume backup!)
- DB2 Configuration information & Group Object Definitions
- Coupling Facility Structures
  - Hold QSG control information and MQ messages

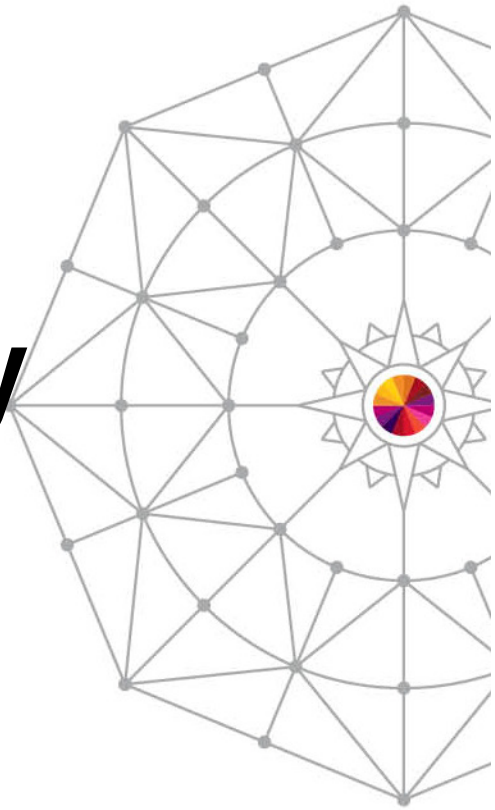


Backing Up a z/OS Queue Manager

- Keep copies of ZPARM, MSTR procedure, product datasets and INP1/INP2 members
- Use dual BSDS, dual active and dual archive logs
- Take backups of your pagesets
  - This can be done while the queue manager is running (fuzzy backups)
  - Make sure you backup Page 0 first, REPRO or ADRDSSU logical copy
- DB2 data should be backed up as part of the DB2 backup procedures
- CF application structures should be backed up on a regular basis
  - These are made in the logs of the queue manager where the backup was issued

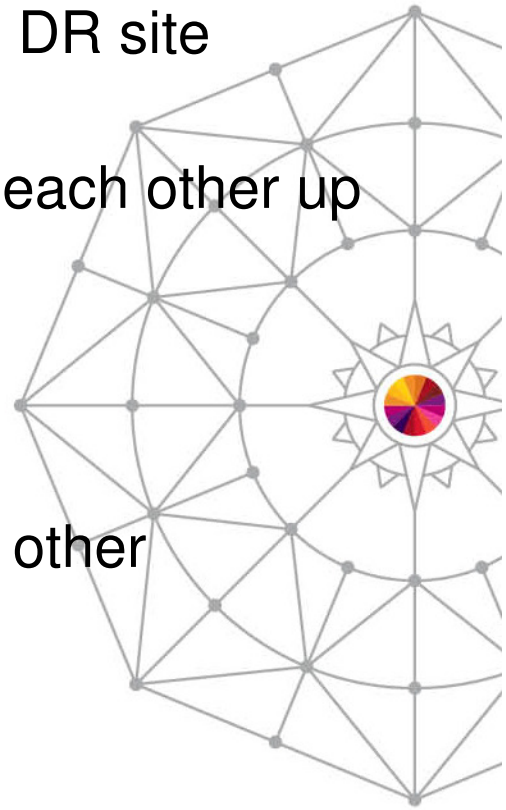


Remote Recovery



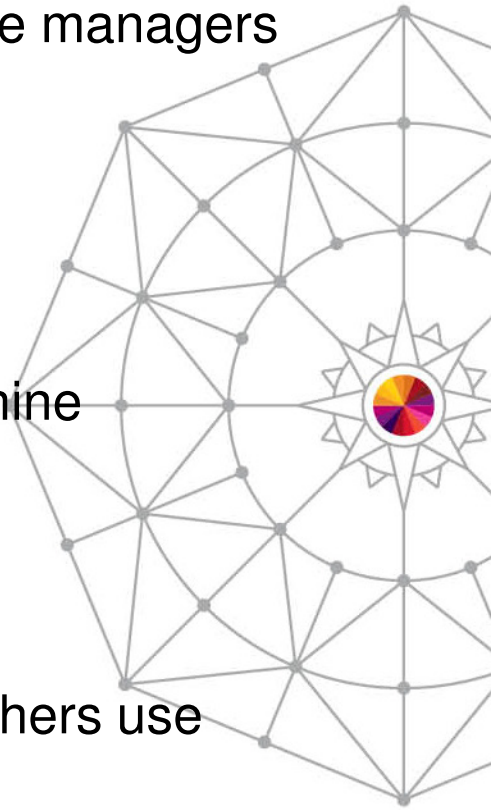
Topologies

- Sometimes a data centre is kept PURELY as the DR site
- Sometimes 2 data centres are in daily use; back each other up for disasters
  - Normal workload distributed to the 2 sites
  - These sites are probably geographically distant
- Another variation has 2 data centres “near” each other
  - Often synchronous replication
  - With a 3rd site providing a long-distance backup
- And of course further variations and combinations of these



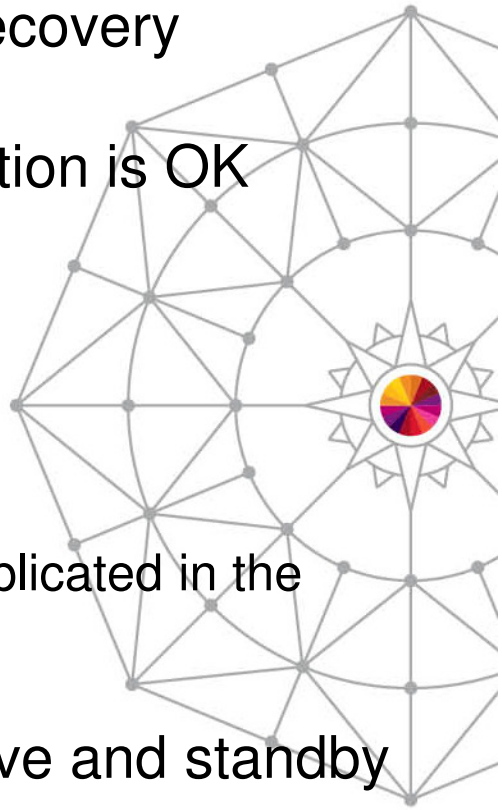
Queue Manager Connections

- DR topologies have little difference for individual queue managers
- But they do affect overall design
  - Where do applications connect to
  - How are messages routed
- Clients need CIntConn definitions that reach any machine
- Will be affected by how you manage network
  - Do DNS names move with the site?
  - Do IP addresses move with the site?
- Some sites always put IP addresses in CONNAME; others use hostname
  - No rule on which is better

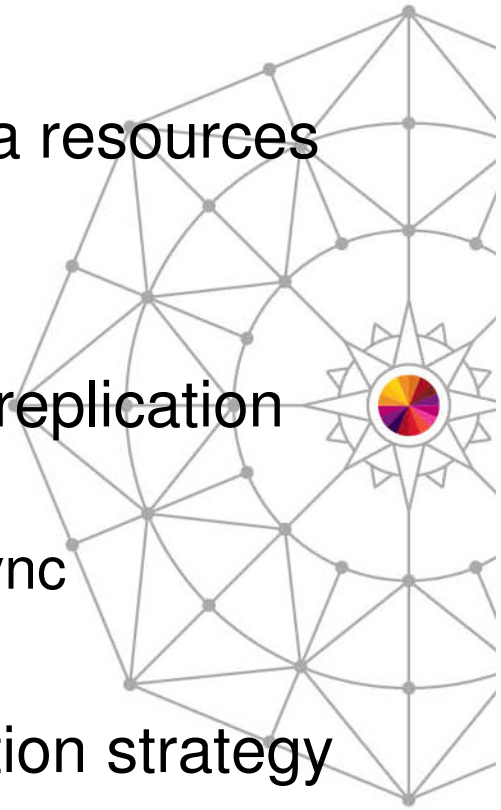


Disk replication

- Disk replication can be used for WMQ disaster recovery
- Either synchronous or asynchronous disk replication is OK
  - Synchronous:
    - No data loss if disaster occurs
    - Performance is impacted by replication delay
    - Limited by distance (eg 100km)
  - Asynchronous:
    - Some limited data loss if disaster occurs
    - It is critical that queue manager data and logs are replicated in the same consistency group if replicating both
- Disk replication cannot be used between the active and standby instances of a multi-instance queue manager
  - Could be used to replicate to a DR site in addition though



Integration with other products

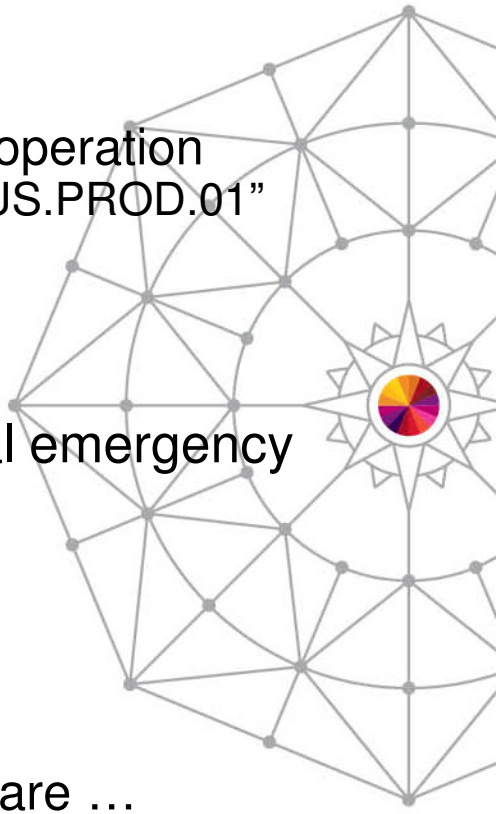


- May want to have consistency with other data resources
  - For example, databases and app servers
- Only way for guaranteed consistency is disk replication where all logs are in same group
  - Otherwise transactional state might be out of sync
- DB2 can use WMQ as part of its own replication strategy
  - InfoSphere Replication Server

Planning and Testing

Planning for Recovery

- Write a DR plan
  - Document everything – to tedious levels of detail
  - Include actual commands, not just a description of the operation
    - Not “Stop MQ”, but “as mqm, run /usr/local/bin/stopmq.sh US.PROD.01”
- And test it frequently
  - Recommend twice a year
  - Record time taken for each task
- Remember that the person executing the plan in a real emergency might be under-skilled and over-pressured
  - Plan for no access to phones, email, online docs …
- Each test is likely to show something you’ve forgotten
  - Update the plan to match
  - You’re likely to have new applications, hardware, software …
- May have different plans for different disaster scenarios



Example Exercises from Hursley

- Different groups have different activities that must continue
  - Realistic scenarios can help show what might not be available

- From the WMQ development lab …
- Most of the change team were told there was a virulent disease and they had to work from home
  - Could they continue to support customers

- If Hursley machine room was taken out by a plane missing its landing at Southampton airport
  - Could we carry on developing the WMQ product
  - Source code libraries, build machines, test machines …
  - Could fixes be produced

- (A common one) Someone hit emergency power-off button

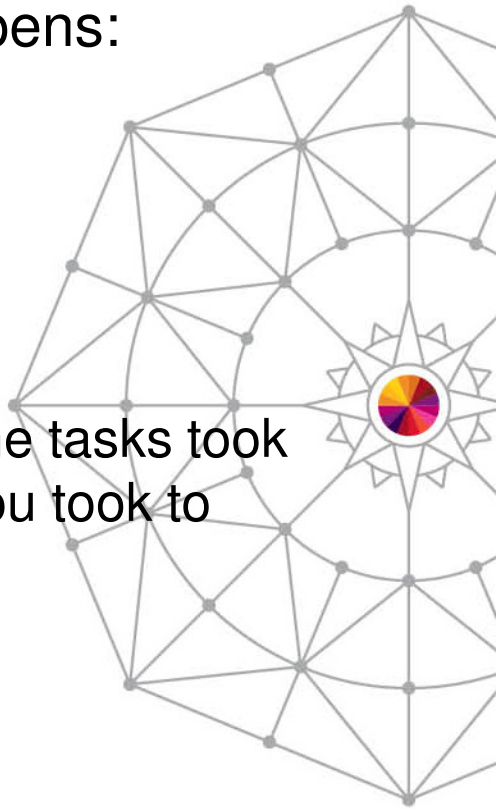
- Not just paper exercises

Other Resources

- Applications may need to deal with replay or loss of data.
  - Decide whether to clear queues down to a known state, or enough information elsewhere to manage replays
- Order of recovery may change with different product releases
  - Every time you install a new version of a product revisit your DR plan
- What do you really need to recover
  - DR site might be lower-power than primary site
  - Some apps might not be critical to the business
  - But some might be unrecognised prereqs

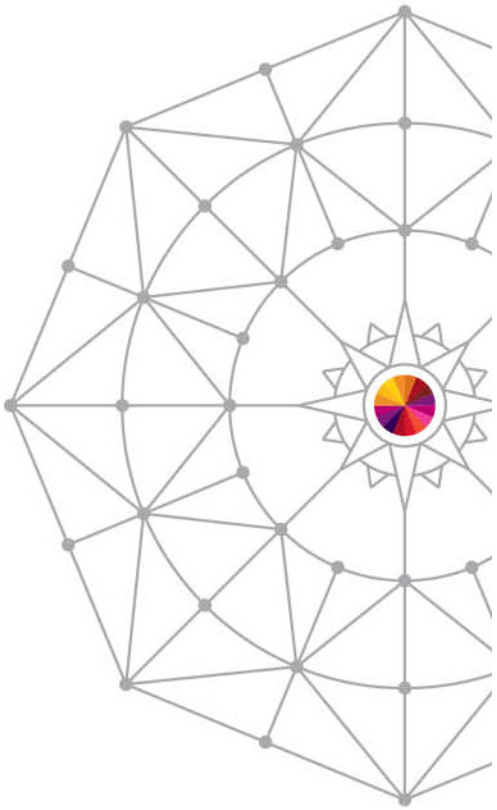
If a Real Disaster Hits

- Hopefully you never need it. But if the worst happens:
- Follow your tested plan
  - Don't try shortcuts
- But also, if possible:
  - Get someone to take notes and keep track of the time tasks took
  - Prepare to attend post mortem meetings on steps you took to recover
  - Accept all offers of assistance
- And afterwards:
  - Update your plan for the next time



Summary

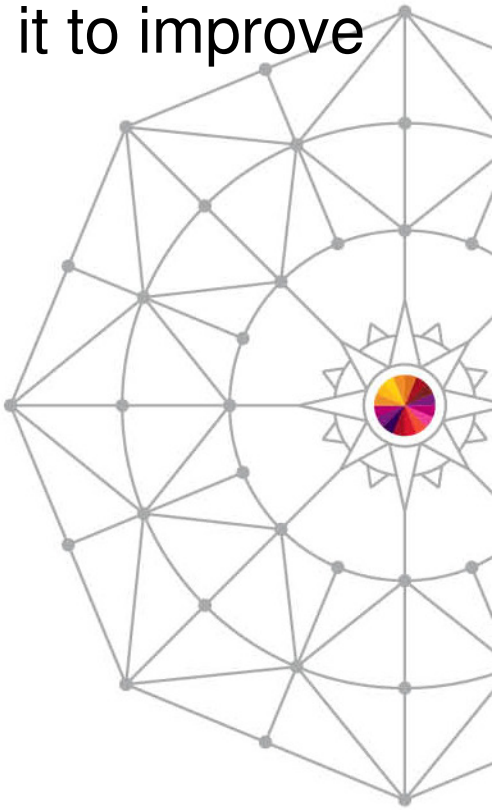
- Various ways of recovering queue managers
- Plan what you need to recover for WMQ
- Plan the relationship with other resources
- Test your plan



We love your Feedback!

Don't forget to submit your SHARE session and speaker feedback!

- Your feedback is very important to us – we use it to improve next year's conference
- Go to the SHARE mobile App



This was session 15026 - The rest of the week ……

	Monday	Tuesday	Wednesday	Thursday	Friday
08:00			What's Available in MQ and Broker for High Availability and Disaster Recovery?	Best Practices in Enhancing our Security with WebSphere MQ	MQ & CICS Workload Balancing in a 'Plexed' World
09:30				What's Wrong with MQ?	
11:00	The Dark Side of Monitoring MQ - SMF 115 and 116 Record Reading and Interpretation			IIIB - Internals of IBM Integration Bus	
12:15				Hands-on Labs for MQ - Take Your Pick!	
01:30		What's New in the MQ Family	MQ on z/OS – Vivisection	MQ Clustering - The Basics, Advances and What's New	
03:00	Introduction to MQ		WebSphere MQ CHINIT Internals	Using IBM WebSphere Application Server and IBM WebSphere MQ Together	
04:30	First Steps with IBM Integration Bus: Application Integration in the new world	What's New in IBM Integration Bus & WebSphere Message Broker	MQ & DB2 – MQ Verbs in DB2 & InfoSphere Data Replication (Q Replication) Performance	MQ Parallel Sysplex Exploitation, Getting the Best Availability From MQ on z/OS by Using Shared Queues	

Legal Disclaimer