Clustering 101

Dorothy Quincy
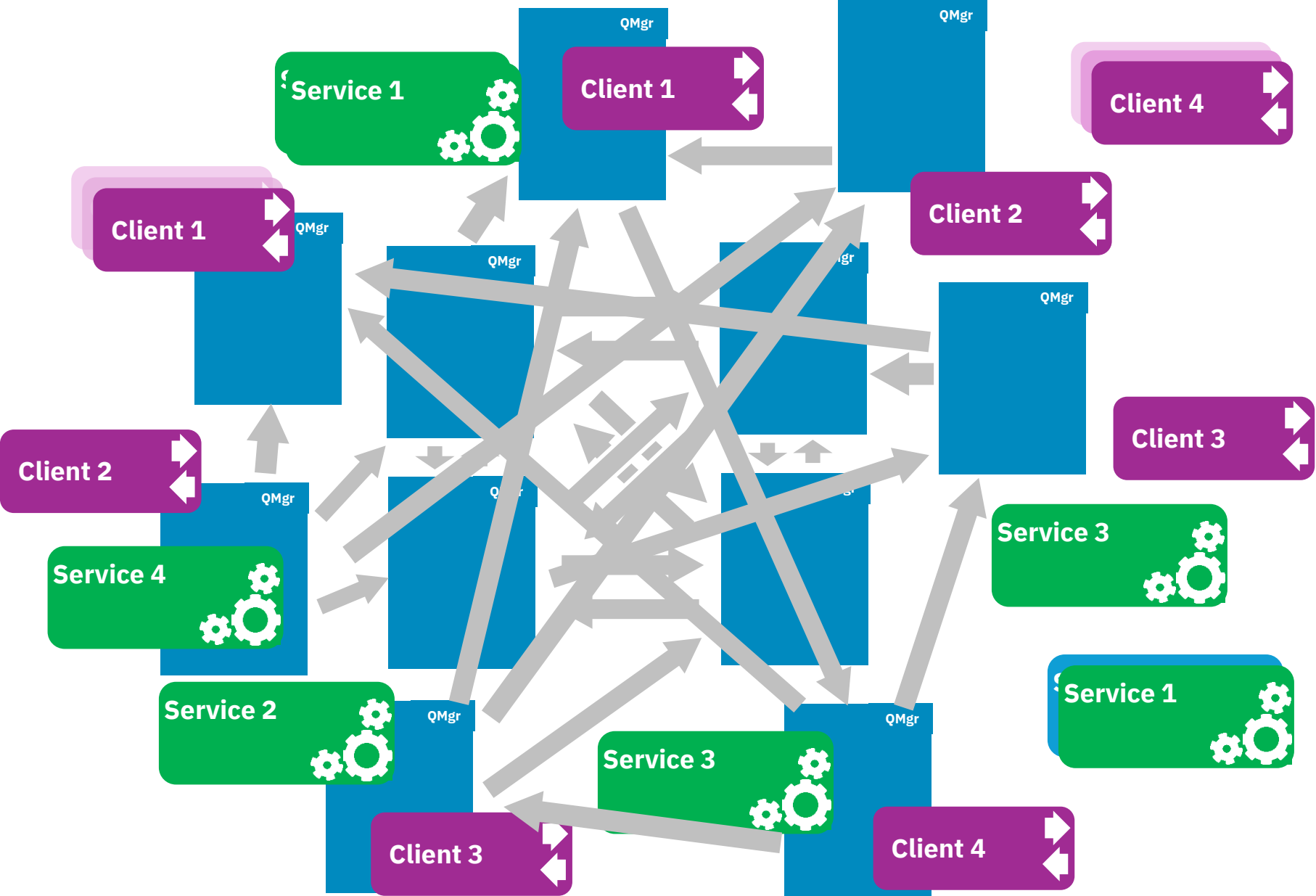Dorothy.quincy@ibm.com

Agenda

- Clustering, what is it and why is it needed?

- How Clustering works and basic setup

- Benefits of Clustering
  - Workload Balancing
  - Service Availability

- Uniform clusters
  - What are they?
  - How to configure them
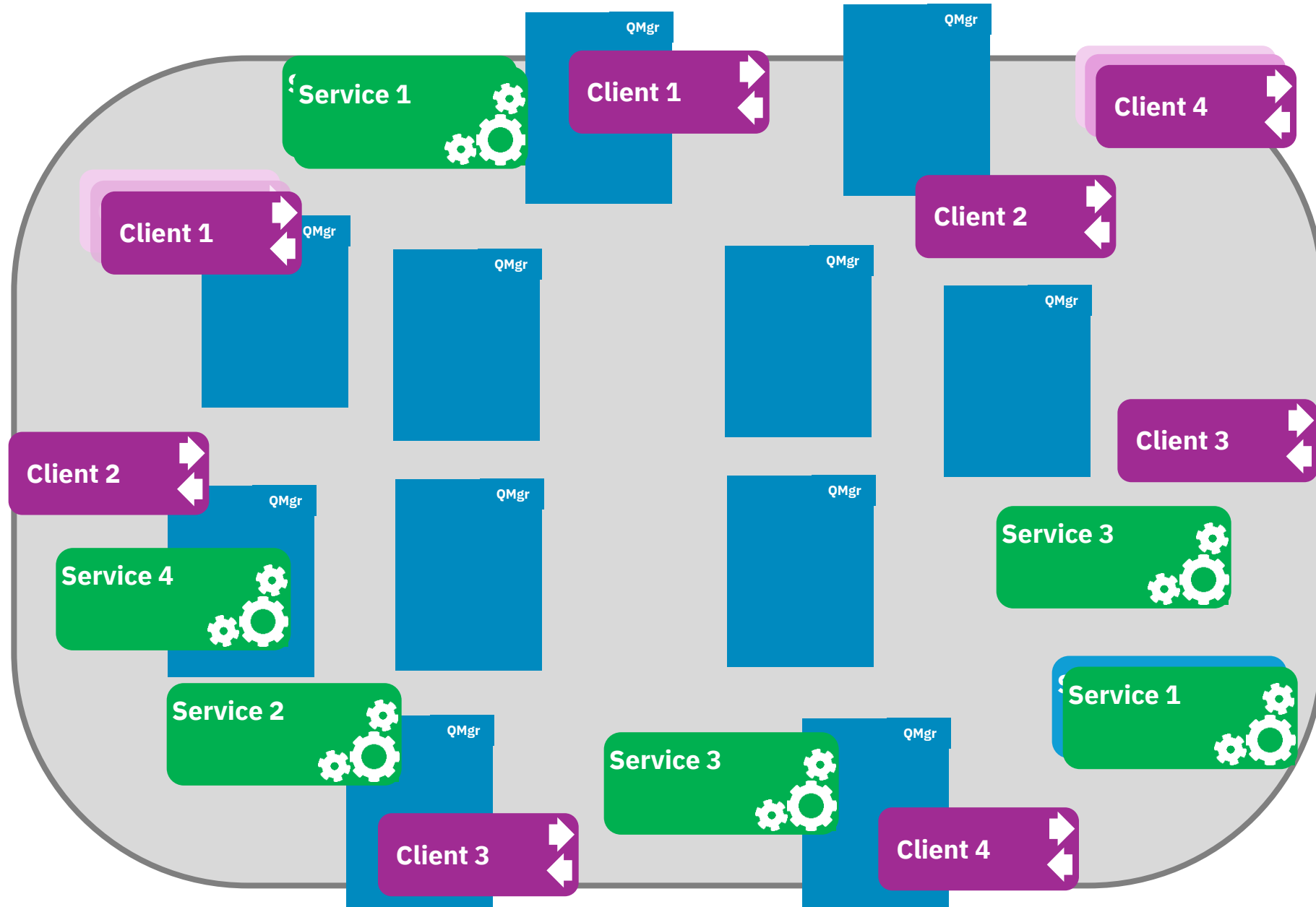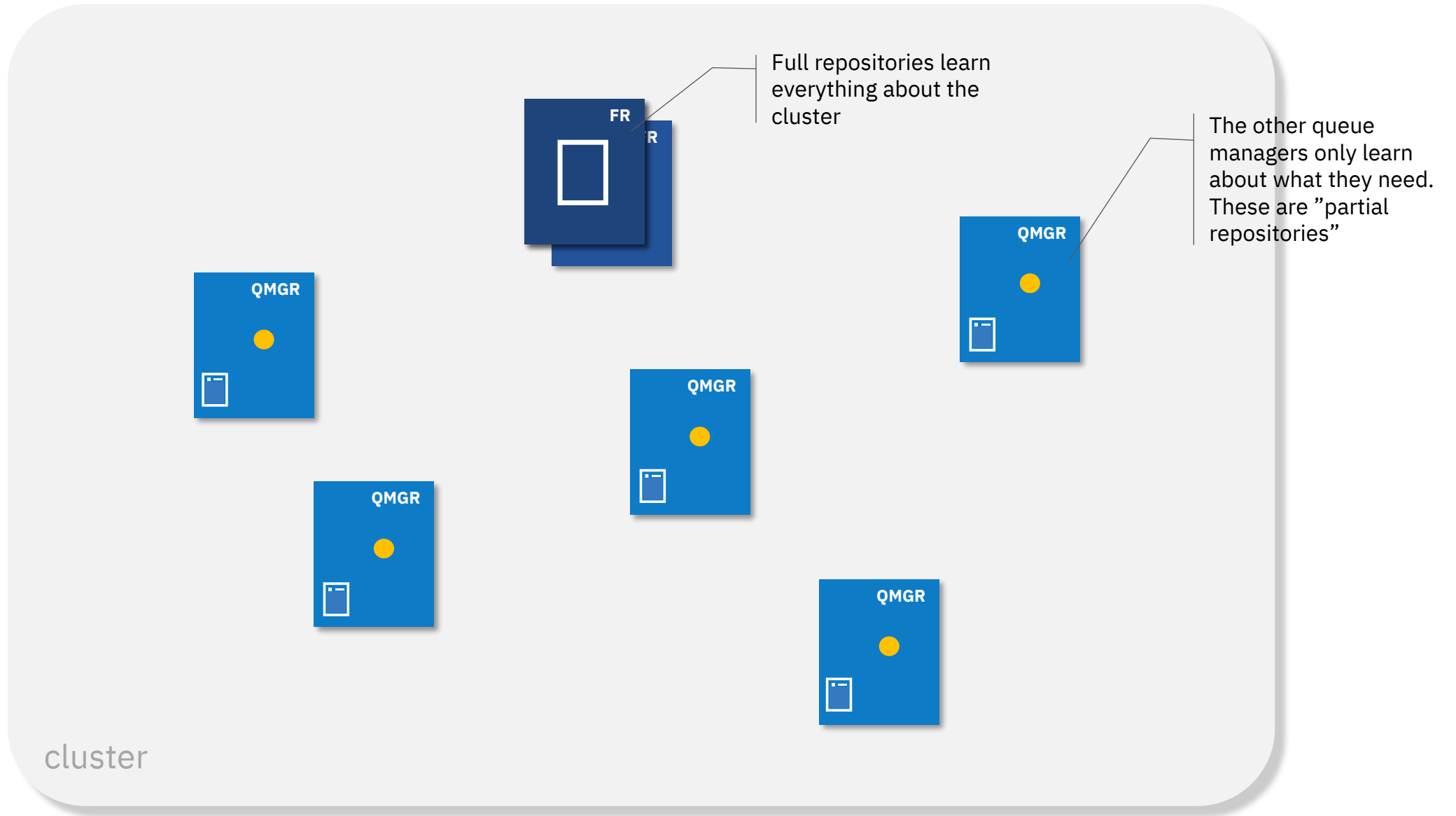
Why is clustering needed?

Where it all begins...

Client 1

QMgr

Service 1

Over time...

...Now you need a cluster

What is a Cluster?

Cluster overview

FR

Full repositories learn everything about the cluster

QMGR

QMGR

QMGR

QMGR

QMGR

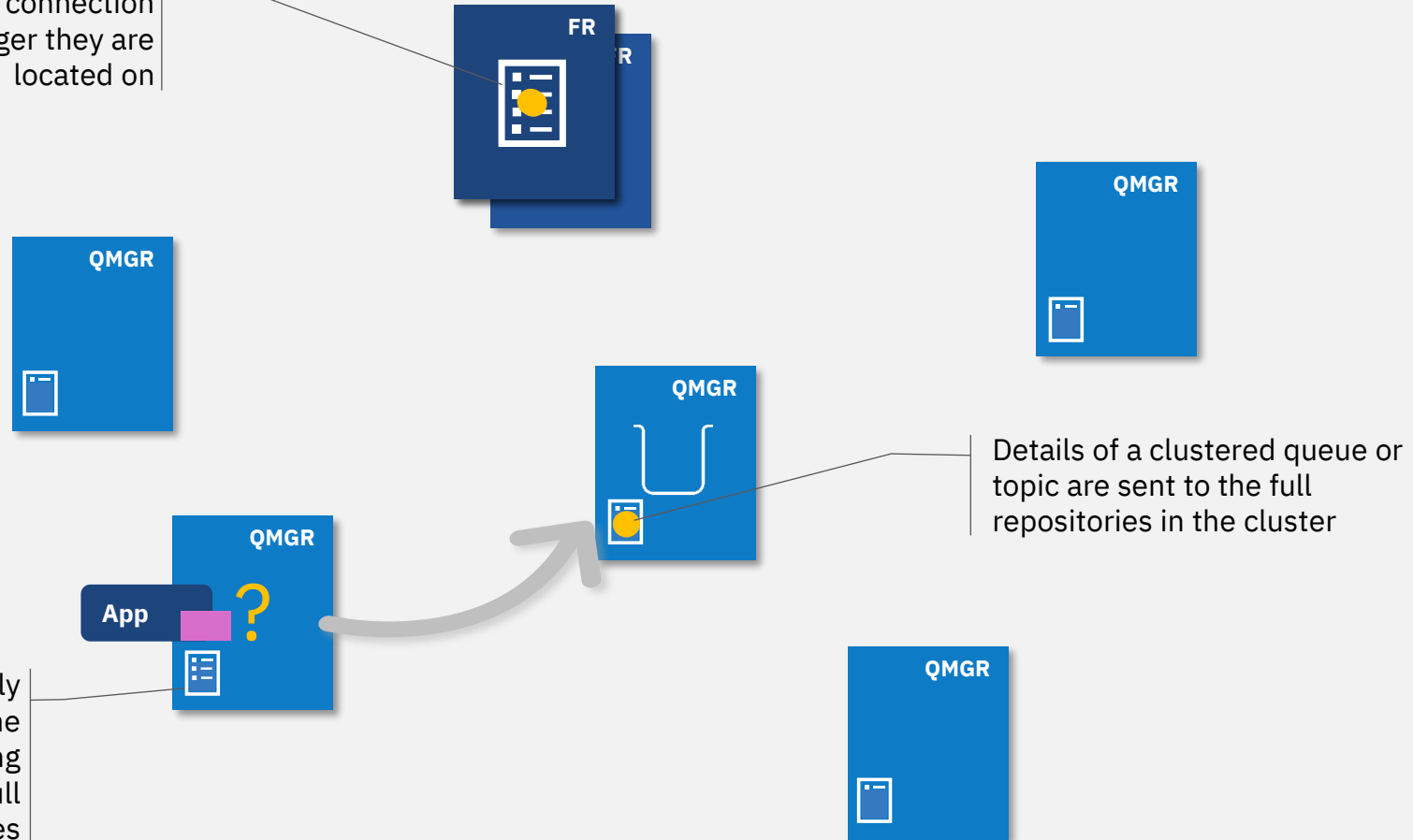The other queue managers only learn about what they need. These are "partial repositories"

cluster

Cluster management
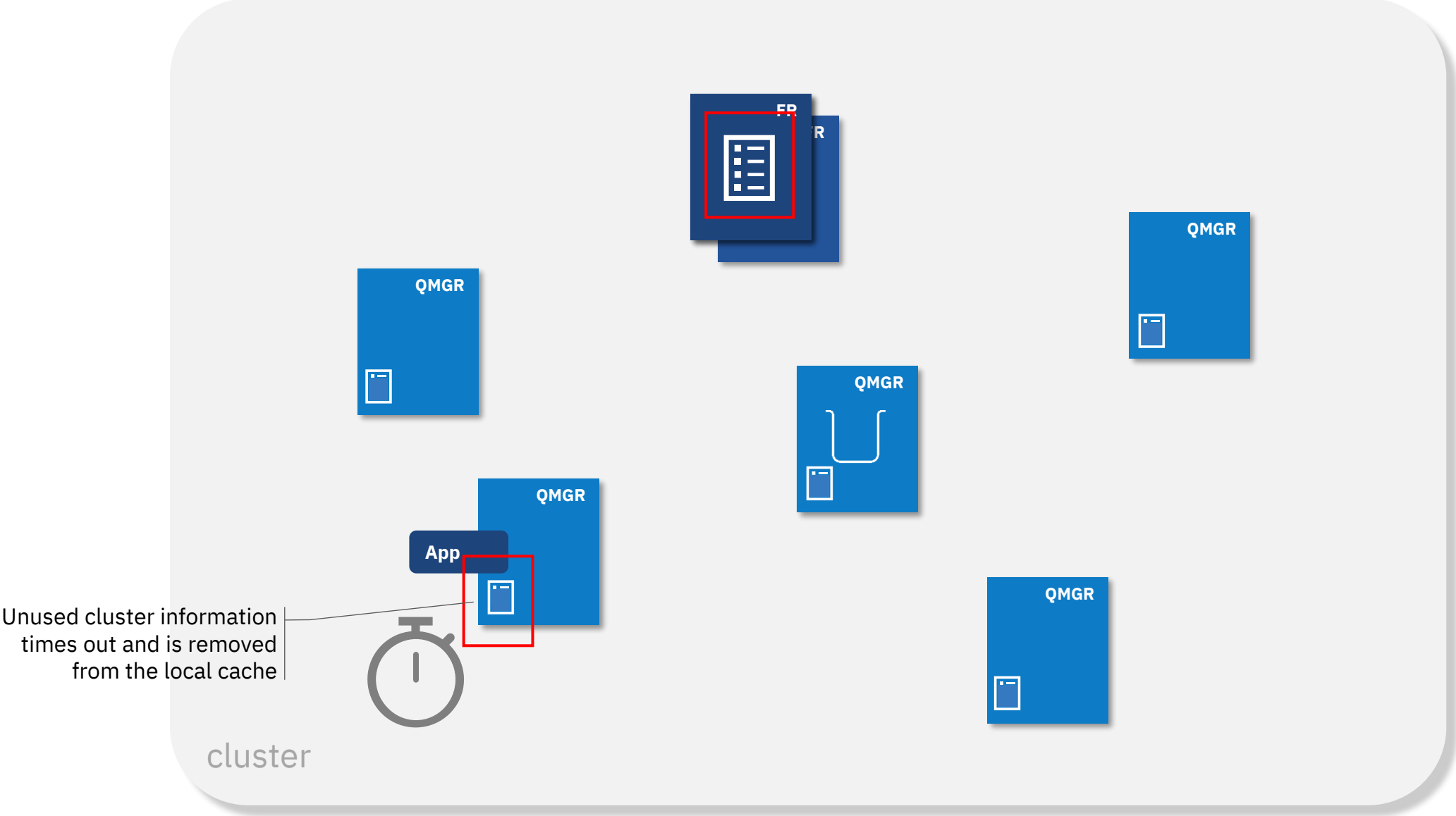
Full repositories will pass on the details of cluster queues and the connection details of the queue manager they are located on

FR

QMGR

QMGR

QMGR

Details of a clustered queue or topic are sent to the full repositories in the cluster

App

QMGR

Queue managers persistently cache their knowledge of the cluster resources, limiting interactions with the full repositories

QMGR

cluster

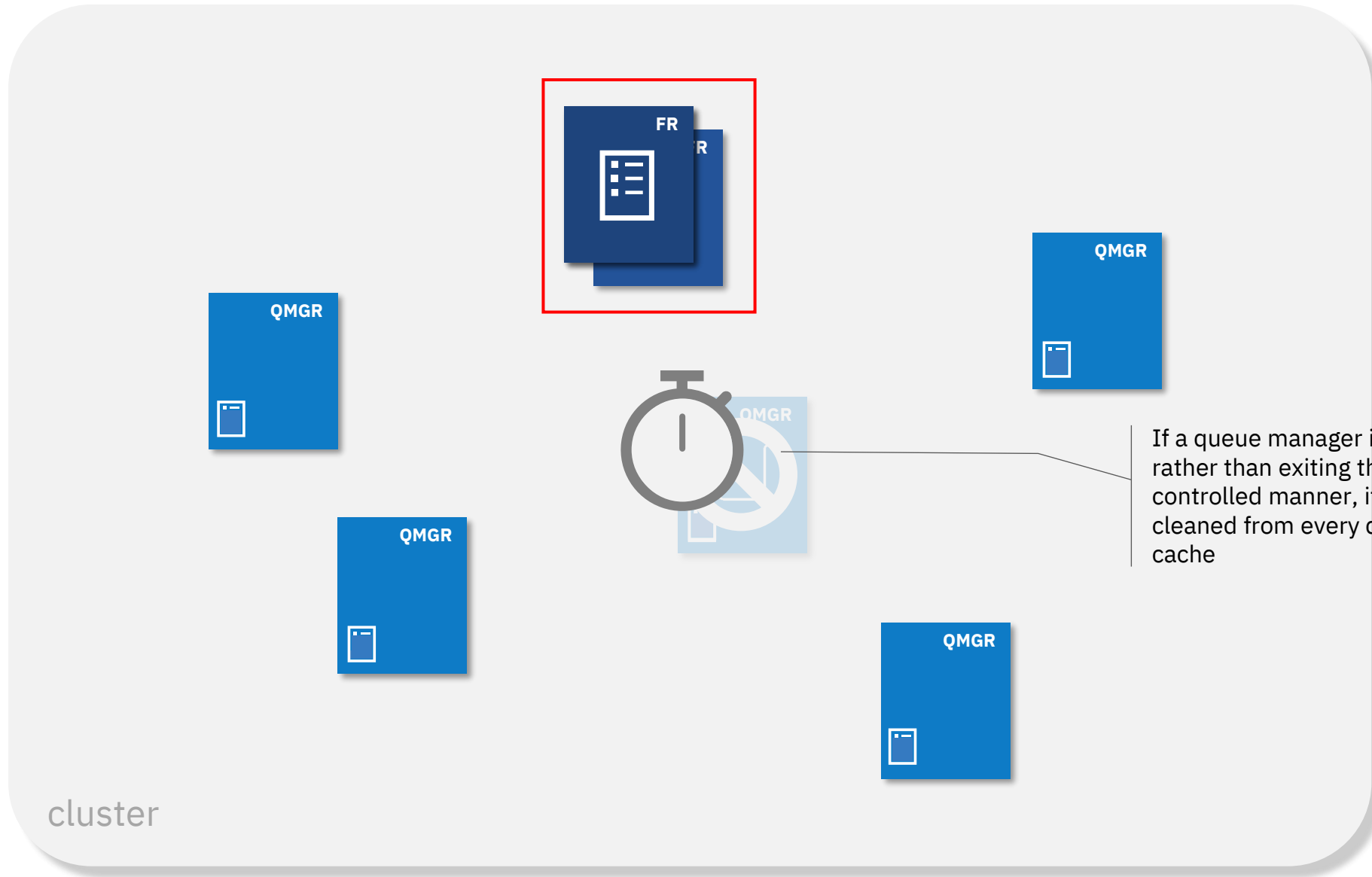# Cluster overview

FR

QMGR

QMGR

QMGR

QMGR

QMGR

QMGR

App

Unused cluster information times out and is removed from the local cache

cluster

Cluster overview

If a queue manager is simply switched off, rather than exiting the cluster in a controlled manner, it will eventually be cleaned from every queue manager's cache
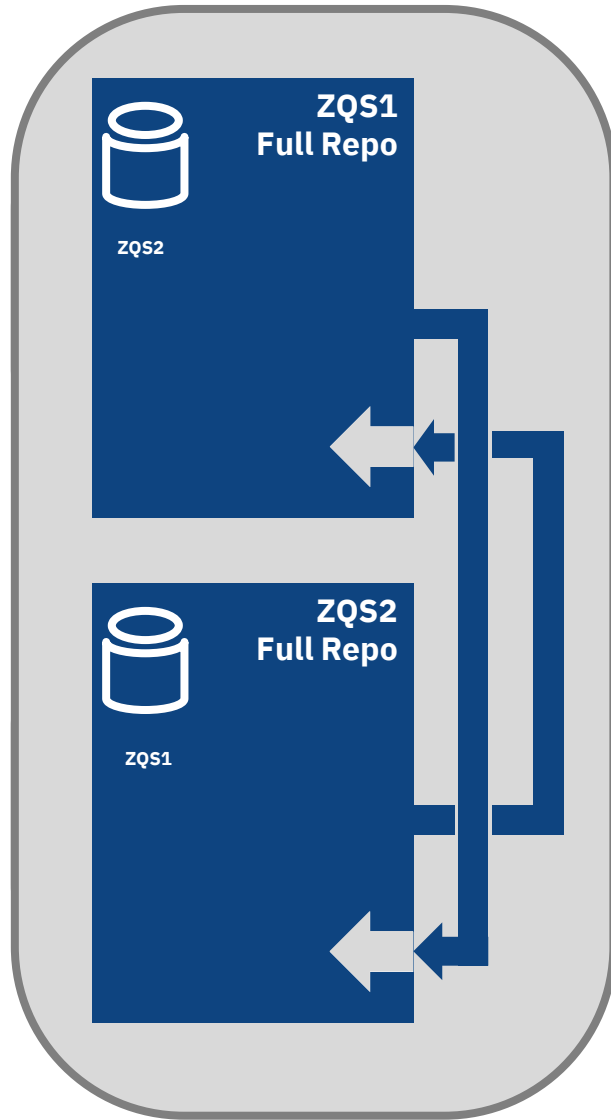
How to create a cluster

Step 1: Create your two full repositories


```
+ZQS1 ALTER QMGR REPOS('ZQCLUSTER')

+ZQS1 DEFINE CHANNEL('TO.ZQS1') CHLTYPE(CLUSRCVR)
      CLUSTER('ZQCLUSTER') CONNAME(FR1 location) TRPTYPE(TCP)

+ZQS1 DEFINE CHANNEL('TO.ZQS2') CHLTYPE(CLUSSDR)
      CLUSTER('ZQCLUSTER') CONNAME(FR2 location) TRPTYPE(TCP)
```

```
+ZQS2 ALTER QMGR REPOS('ZQCLUSTER')

+ZQS2 DEFINE CHANNEL('TO.ZQS2') CHLTYPE(CLUSRCVR)
      CLUSTER('ZQCLUSTER') CONNAME(FR2 location) TRPTYPE(TCP)

+ZQS2 DEFINE CHANNEL('TO.ZQS1') CHLTYPE(CLUSSDR)
      CLUSTER('ZQCLUSTER') CONNAME(FR1 location) TRPTYPE(TCP)
```

*This object creation is handled by MQ Explorer tooling

Step 2: Add in more queue managers


```
+ZQS3 DEFINE CHANNEL('TO.ZQS3')
       CHLTYPE(CLUSRCVR)
       CLUSTER('ZQCLUSTER')
       CONNAME(ZQS3 location)

+ZQS3 DEFINE CHANNEL('TO.ZQS1')
       CHLTYPE(CLUSSDR)
       CLUSTER('ZQCLUSTER')
       CONNAME(ZQS1 location)

+ZQS3 DEFINE QLOCAL(Q1)
       CLUSTER(ZQCLUSTER)
```

DISPLAY QCLUSTER(*)
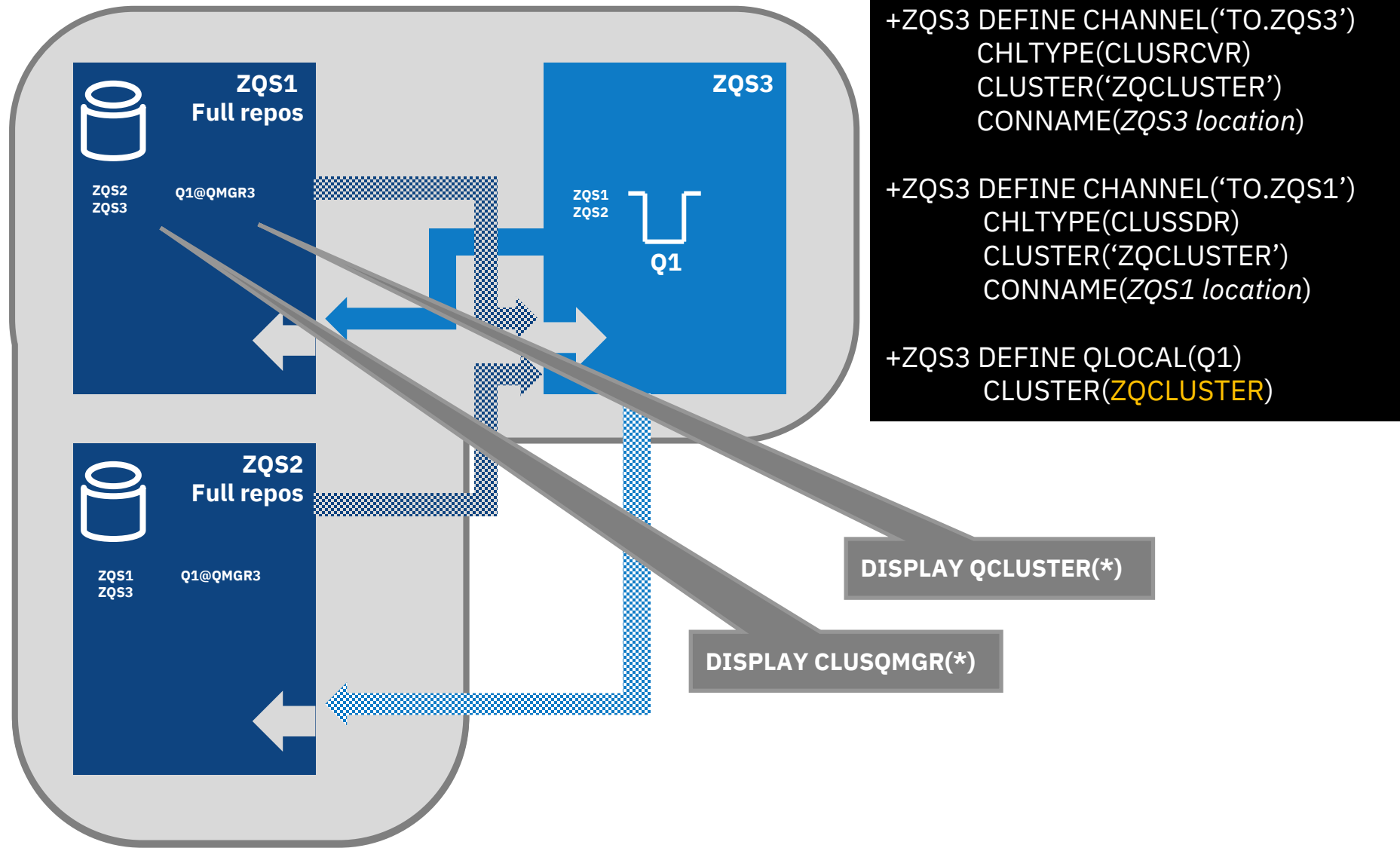
DISPLAY CLUSQMGR(*)

```
ZQS1 DISPLAY CLUSQMGR(*)
CSQM293I ZQS1 CSQMDRTC 2 CLUSQMGR FOUND MATCHING REQUEST CRITERIA
CSQM201I ZQS1 CSQMDRTC  DISPLAY CLUSQMGR DETAILS 769
CLUSQMGR(ZQS1)
CLUSTER(ZQCLUSTER)
CHANNEL(TO.ZQS1)
 END CLUSQMGR DETAILS
CSQM201I ZQS1 CSQMDRTC  DISPLAY CLUSQMGR DETAILS 770
CLUSQMGR(ZQS2)
CLUSTER(ZQCLUSTER)
CHANNEL(TO.ZQS2)
 END CLUSQMGR DETAILS
```

information about remote queue managers **and** the channel definitions by which they can be reached

```
ZQS1 DISPLAY QCLUSTER(*)
CSQM293I ZQS1 CSQMDRTC 1 QCLUSTER FOUND MATCHING REQUEST CRITERIA
CSQM201I ZQS1 CSQMDRTC  DISPLAY QCLUSTER DETAILS 811
QUEUE(Q1)
TYPE(QCLUSTER)
QSGDISP(QMGR)
 END QCLUSTER DETAILS
CSQ9022I ZQS1 CSQMDRTC ' DISPLAY QCLUSTER' NORMAL COMPLETION
```

information about the queue and topic objects that we can see via a cluster

Step 2: Add in more queue managers
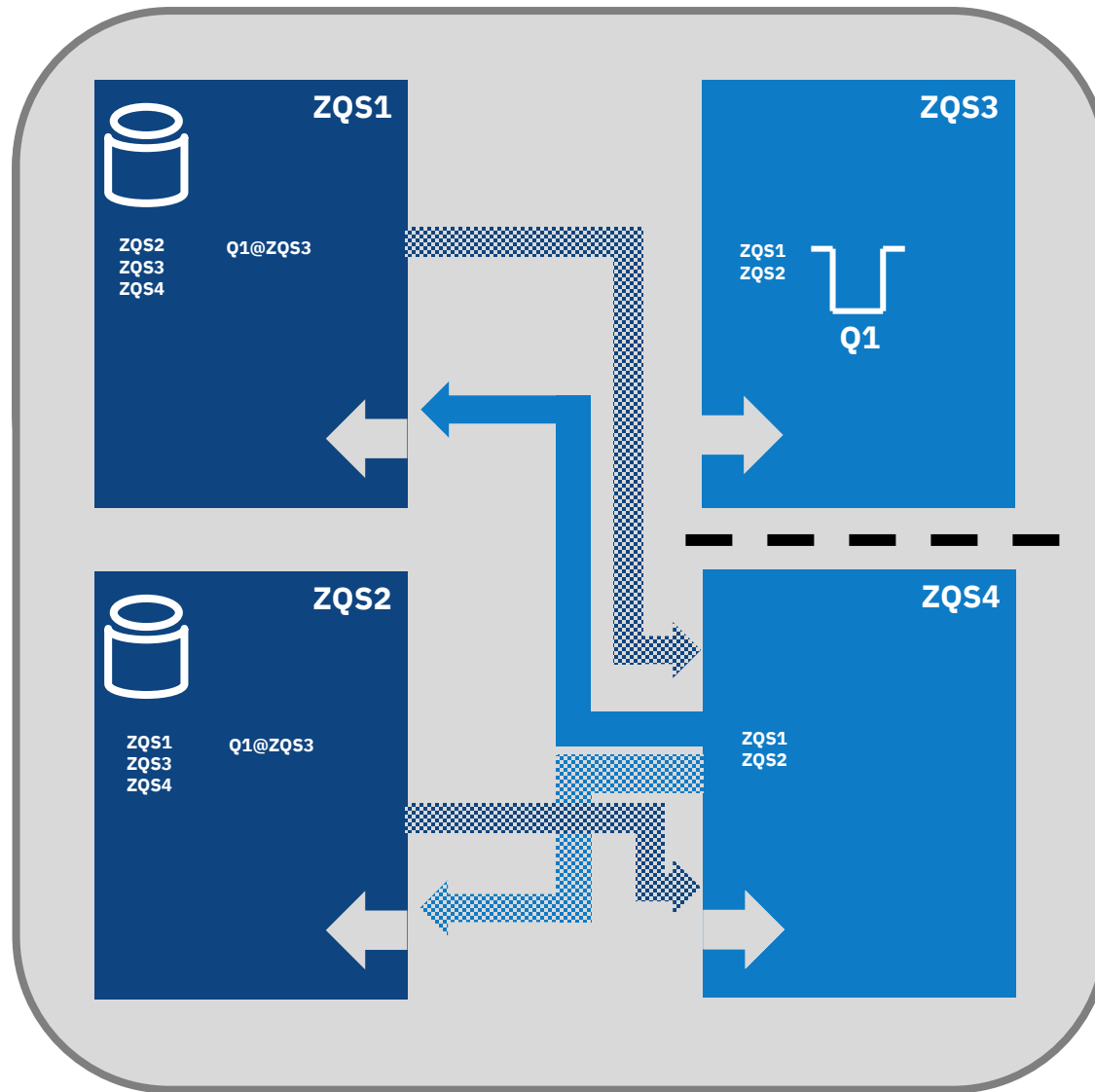

```
+ZQS3 DEFINE CHANNEL('ZQCLUSTER.ZQS3')
       CHLTYPE(CLUSRCVR)
       CLUSTER('ZQCLUSTER')
       CONNAME(QMGR1 location)

+ZQS3 DEFINE CHANNEL('ZQCLUSTER.ZQS1')
       CHLTYPE(CLUSSDR)
       CLUSTER('ZQCLUSTER')
       CONNAME(FR1 location)

+ZQS3 DEFINE QLOCAL(Q1)
       CLUSTER(ZQCLUSTER)
```
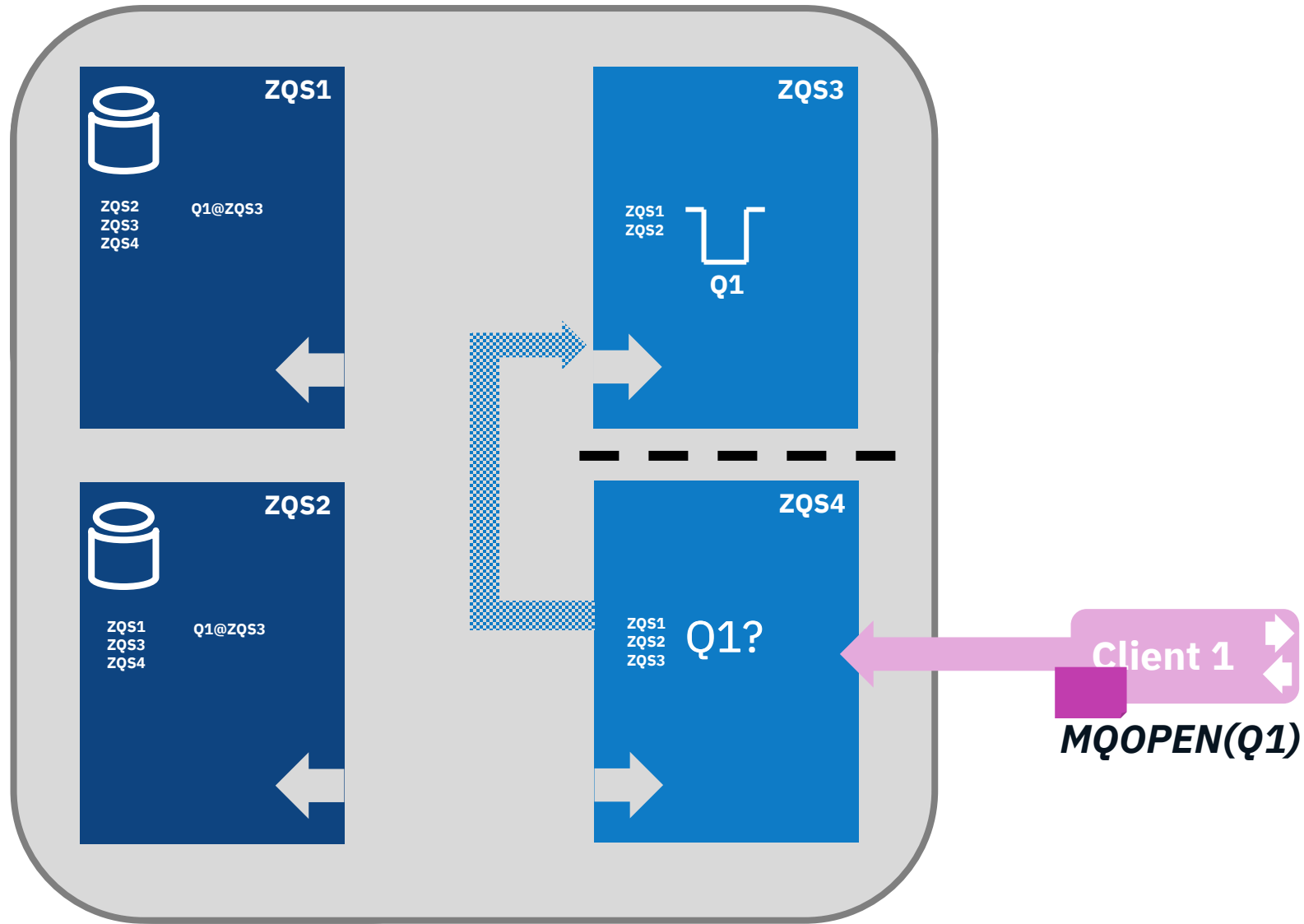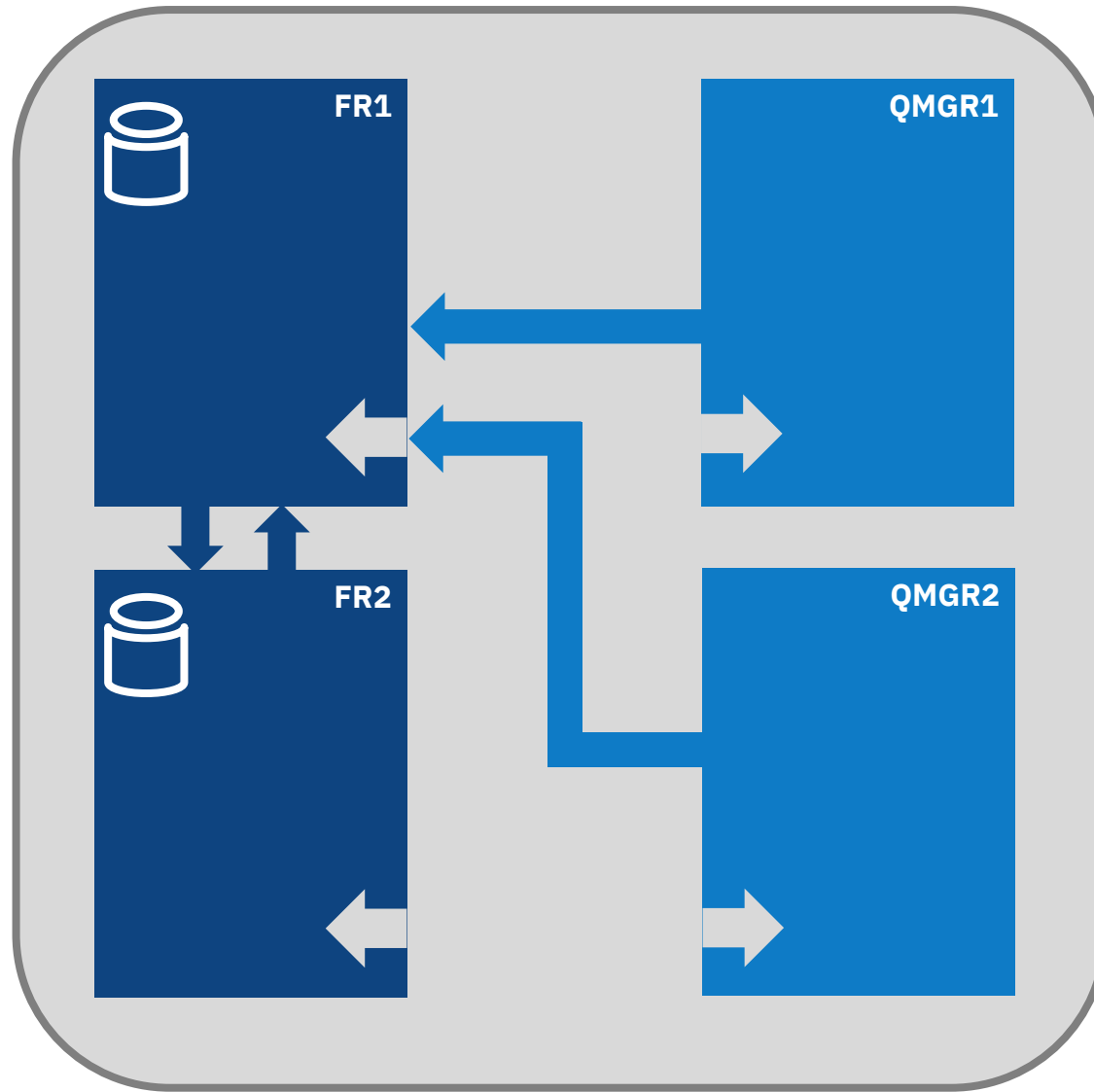
```
+ZQS4 DEFINE CHANNEL('ZQCLUSTER.ZQS4')
       CHLTYPE(CLUSRCVR)
       CLUSTER('ZQCLUSTER')
       CONNAME(ZQS4 location)

+ZQS4 DEFINE CHANNEL('ZQCLUSTER.ZQS1')
       CHLTYPE(CLUSSDR)
       CLUSTER('ZQCLUSTER')
       CONNAME(ZQS1 location)
```

Step 3: Start sending messages
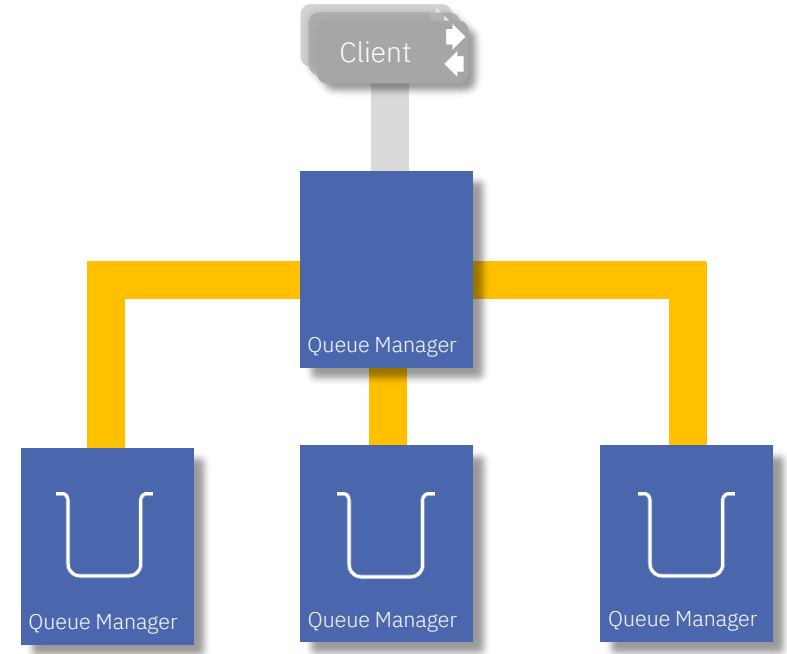
So all you needed...

- Two full repository queue managers

- A cluster receiver channel each

- A single cluster sender each

- No need to manage pairs of channels between each queue manager combination or their transmission queues

- No need for remote queue definitions
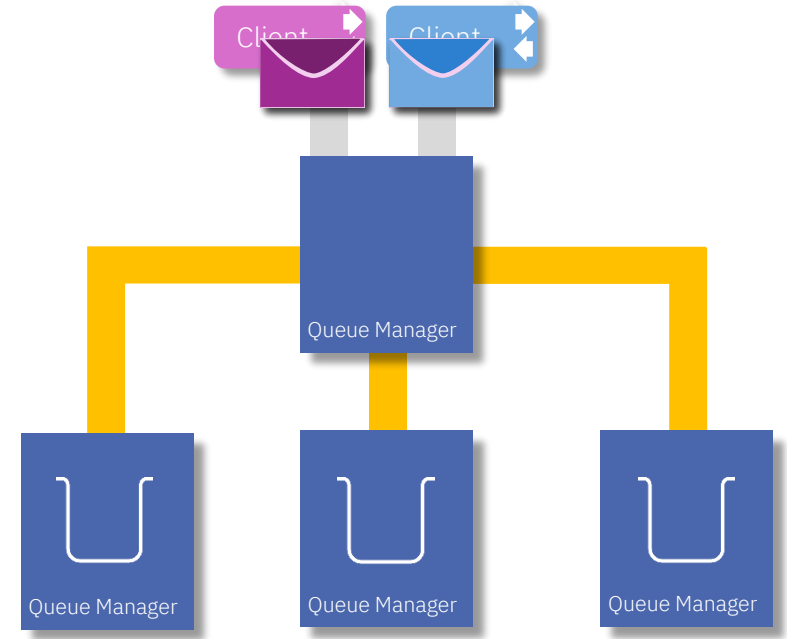
Shared Cluster Queues

Channel workload balancing

- Cluster workload balancing applies when there are
  multiple cluster queues of the same name

- Cluster workload balancing will be applied in one of three ways:
  - When the putting application opens the queue   - bind on open
  - When a message group is started   - bind on group
  - When a message is put to the queue   - bind not fixed

- When workload balancing is applied:
  - The source queue manager builds a list of
    all potential targets based on the queue name
  - Eliminates the impossible options
  - Prioritises the remainder
  - If more than one come out equal, workload balancing ensues ...

- Balancing is based on:
  - The **channel** – not the target queue
  - Channel traffic to all queues is taken into account
  - Weightings can be applied to the channel

- ... this is used to send the messages to the chosen target

Channel workload balancing

- Cluster workload balancing applies when there are **multiple cluster queues of the same name**

- Cluster workload balancing will be applied in **one of three ways**:
  - When the putting application opens the queue **- bind on open**
  - When a message group is started   **- bind on group**
  - When a message is put to the queue   **- bind not fixed**

- When workload balancing is applied:
  - The source queue manager builds a list of all potential targets based on the queue name
  - **Eliminates** the impossible options
  - **Prioritises** the remainder
  - If more than one come out equal, **workload balancing** ensues ...

- Balancing is based on:
  - The **channel** – not the target queue
  - **Channel traffic** to **all queues** is taken into account
  - **Weightings** can be applied to the channel

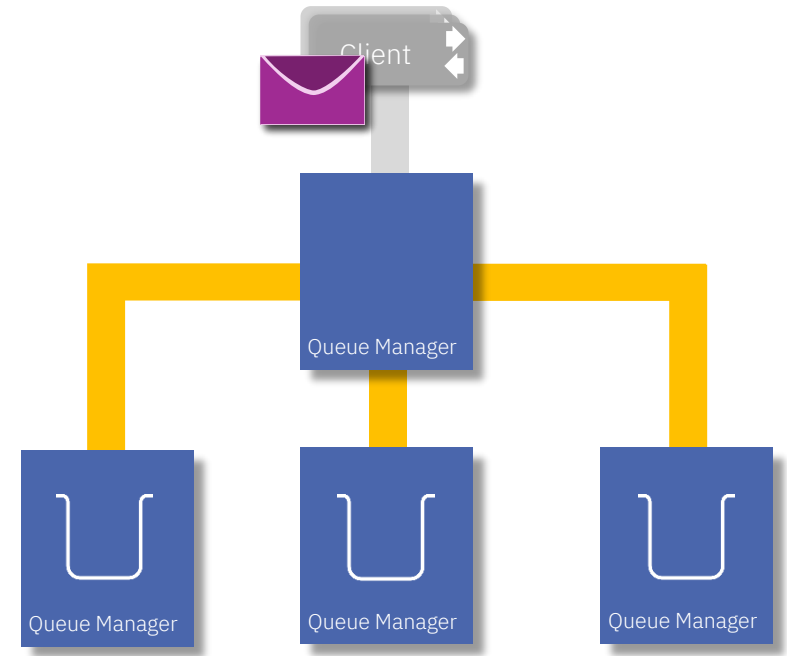- ... this is used to send the messages to the chosen target
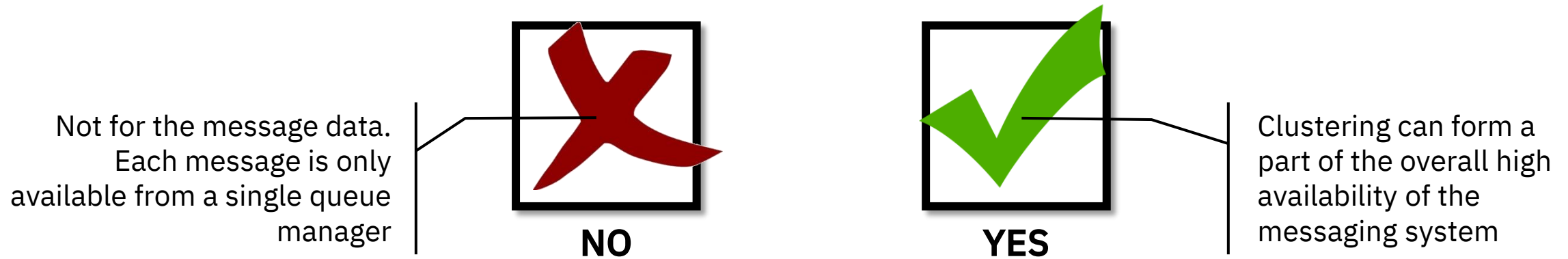
Channel workload balancing

- Cluster workload balancing applies when there are
  multiple cluster queues of the same name

- Cluster workload balancing will be applied in **one of three ways**:
  - When the putting application opens the queue   **- bind on open**
  - When a message group is started   **- bind on group**
  - When a message is put to the queue   **- bind not fixed**

- When workload balancing is applied:
  - The source queue manager builds a list of
    all potential targets based on the queue name
  - **Eliminates** the impossible options
  - **Prioritises** the remainder
  - If more than one come out equal, **workload balancing** ensues ...

- Balancing is based on:
  - The **channel** – not the target queue
  - **Channel traffic** to **all queues** is taken into account
  - **Weightings** can be applied to the channel

- ... this is used to send the messages to the chosen target

Availability routing

Clustering for availability

Is MQ Clustering a high availability solution?

Not for the message data. Each message is only available from a single queue manager

NO

YES

Clustering can form a part of the overall high availability of the messaging system

- Having multiple potential targets for any message can improve the availability of the solution, always providing an option to process new messages.
- A queue manager in a cluster has the ability to route new and old messages based on the **availability of the channels**, routing messages to running queue managers.
- Clustering can be used to route messages to active consuming applications.

Channel availability routing

- When performing workload balancing, the availability of the channel to reach the target is a factor
- All things being equal, messages will be routed to those targets with a working channel

Routing of messages based on availability doesn't just happen when they're first put, it also occurs for queued transmission messages every time the channel is retried
So blocked messages can be re-routed, if they're not prevented...

Things that can prevent routing
- Applications targeting messages at a specific queue manage (e.g. reply message)
- Using "cluster workload rank"
- Binding messages to a target
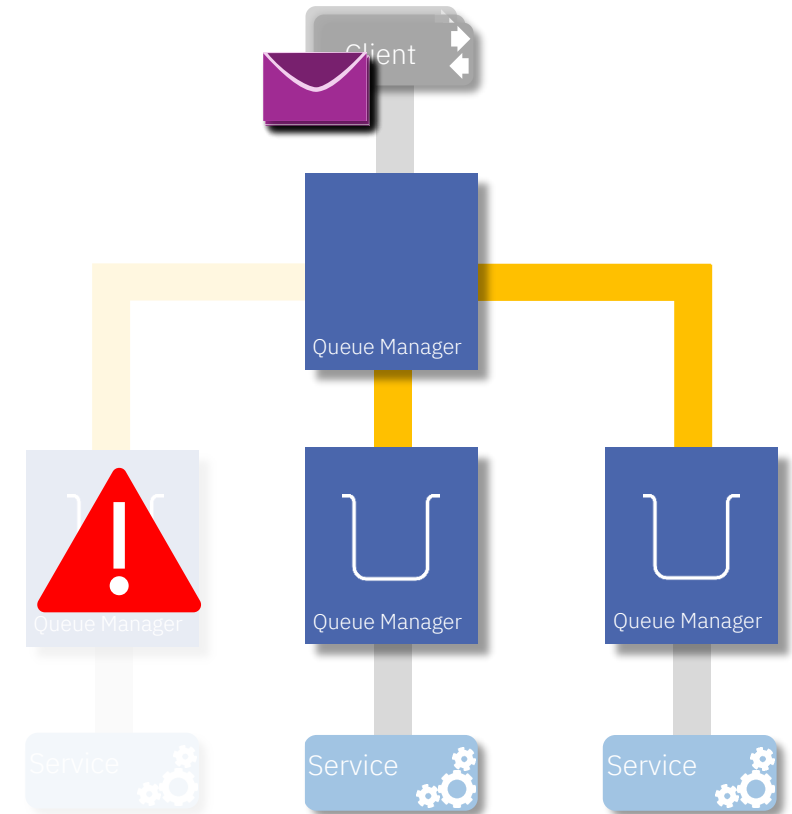
Pros and cons of binding

Bind on open	Bind on group	Bind not fixed

Bind context:
   Duration of an *open*
   Duration of *logical group*

- All messages put within the bind context will go to same target*
- Message order can be preserved**
- Workload balancing logic is only driven at the start of the context

- Once a target has been chosen it **cannot change**
  - Whether it's available or not
  - Even if all the messages could be redirected

Bind context:
   None

- Greater *availability,* a message will be redirected to an available target***

- Overhead of workload balancing logic for every message
- Message order may be affected

<u>Bind on open is the default</u>
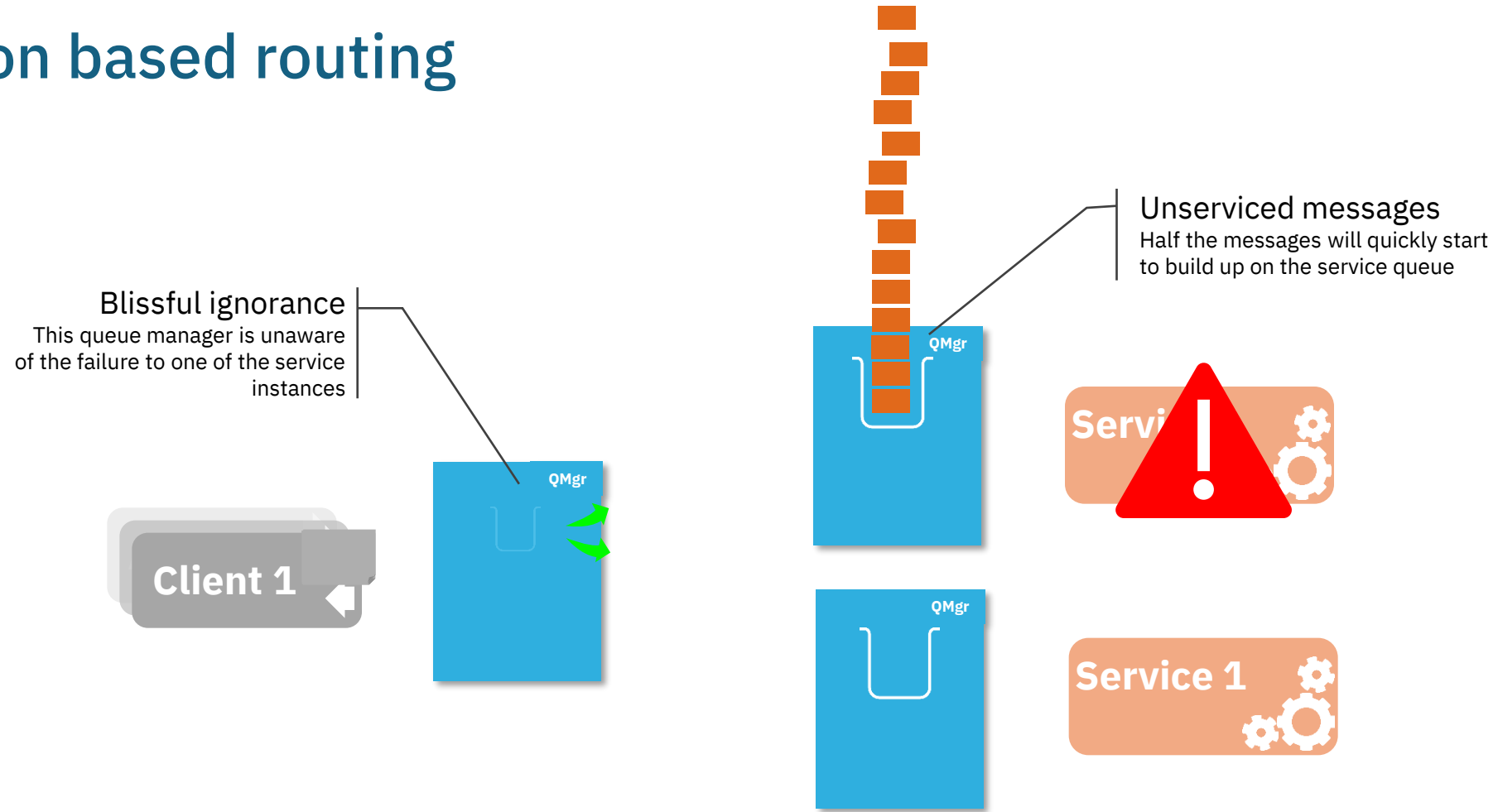It could be set on the cluster queue (don't forget aliases) or in the app

---

* While a route is known by the source queue manager, it won't be rebalanced, but it could be DLQd
** Other aspects may affect ordering (e.g. deadletter queueing)
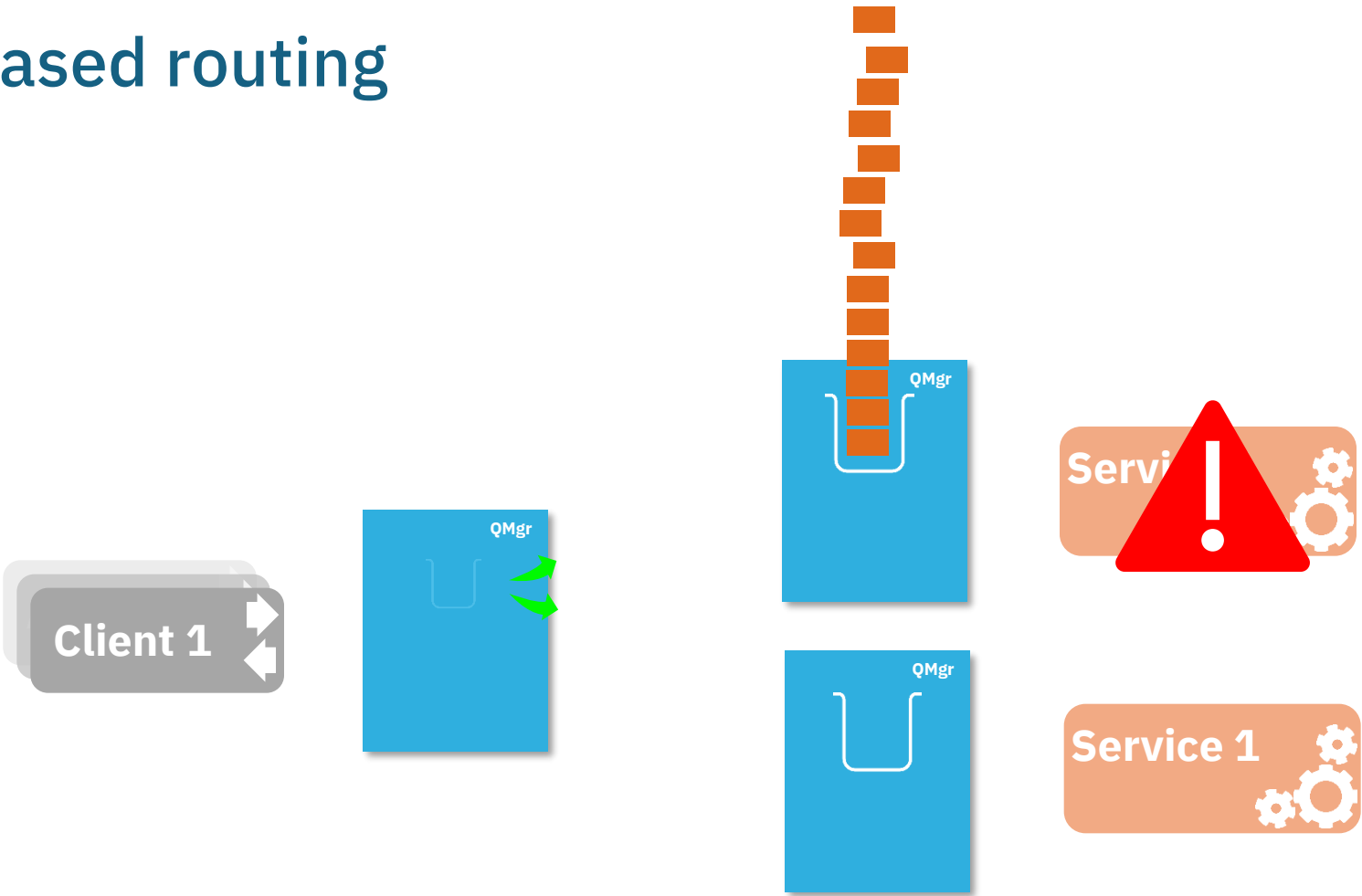*** Unless it's fixed for another reason (e.g. specifying a target queue manager)

Application availability routing

Application based routing

Blissful ignorance
This queue manager is unaware of the failure to one of the service instances

Unserviced messages
Half the messages will quickly start to build up on the service queue
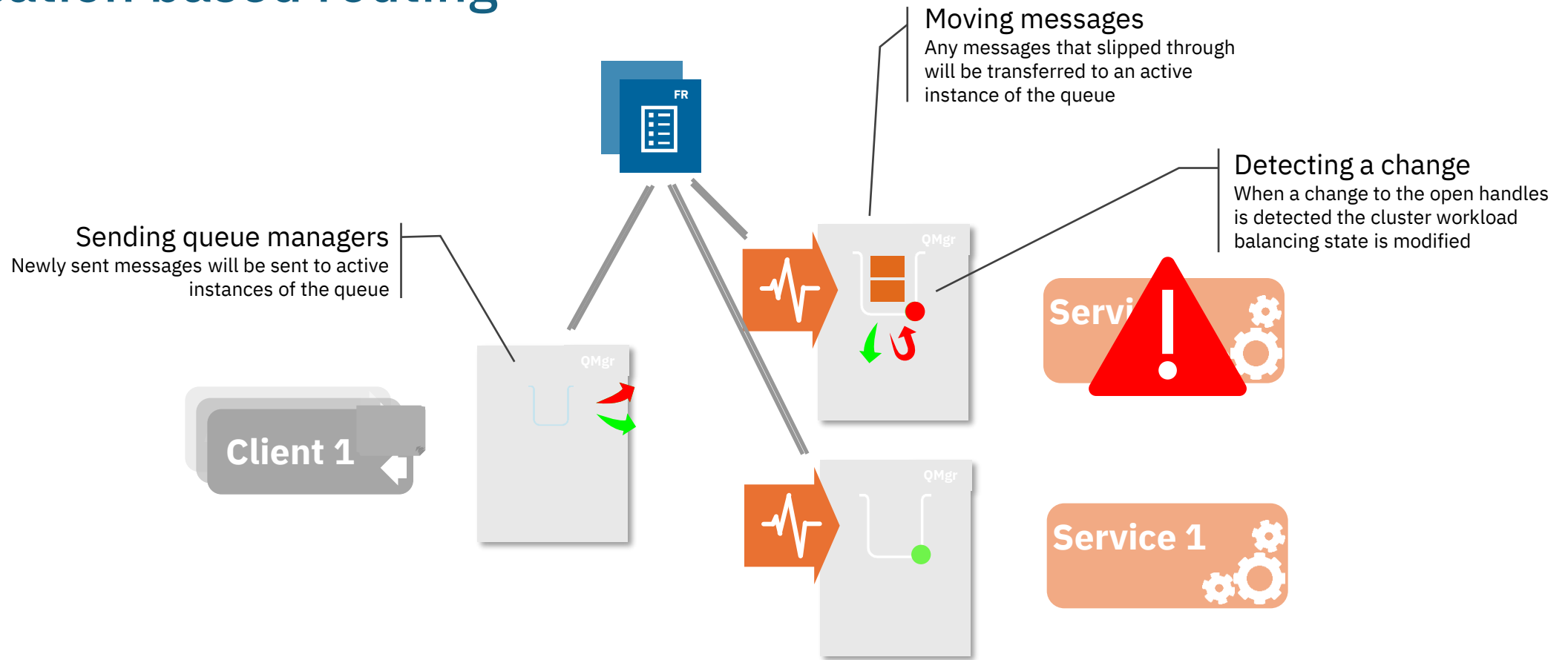
QMgr

Client 1

QMgr

Servi ⚠

QMgr

Service 1

- Cluster workload balancing does not take into account the availability of receiving applications
- Or a build up of messages on a queue

Application based routing

Application based routing

Moving messages
Any messages that slipped through will be transferred to an active instance of the queue

Detecting a change
When a change to the open handles is detected the cluster workload balancing state is modified

Sending queue managers
Newly sent messages will be sent to active instances of the queue

- MQ provides a sample monitoring service tool, **amqsclm**
- It regularly checks for attached consuming applications (IPPROCS)
- And automatically adjusts the cluster queue definitions to route messages intelligently (CLWLPRTY)
- That information is automatically distributed around the cluster

Thank You !

Any Questions Please ?

Your feedback is important!

Submit a session evaluation for each session you attend:

www.share.org/evaluation

www.share.org/evaluation