



IBM Software Group

# MQ Pub/Sub: introduction to distributed networks

<http://www.ibm.com/support/docview.wss?uid=swg27050244>

Angel Rivera ([rivera@us.ibm.com](mailto:rivera@us.ibm.com))

IBM MQ Distributed Level 2 Support

Date last updated: 20-Sep-2017



## Related presentations

This presentation is one of a series.  
For the complete list, please see:

<https://developer.ibm.com/answers/questions/402074/mq-pubsub-training-presentations.html>

MQ Pub/Sub: training presentations



# Agenda

## Motivation

There are three basic topologies for a Pub/Sub network:

- Direct routed cluster
- Topic host routed cluster
- Hierarchy

Each approach will be covered in detail in another presentation



# Motivation

This section is based on the following presentation, which provides more details on distributed MQ: remote queue definitions, transmission queues, sender channel, receiver channel, MCA.

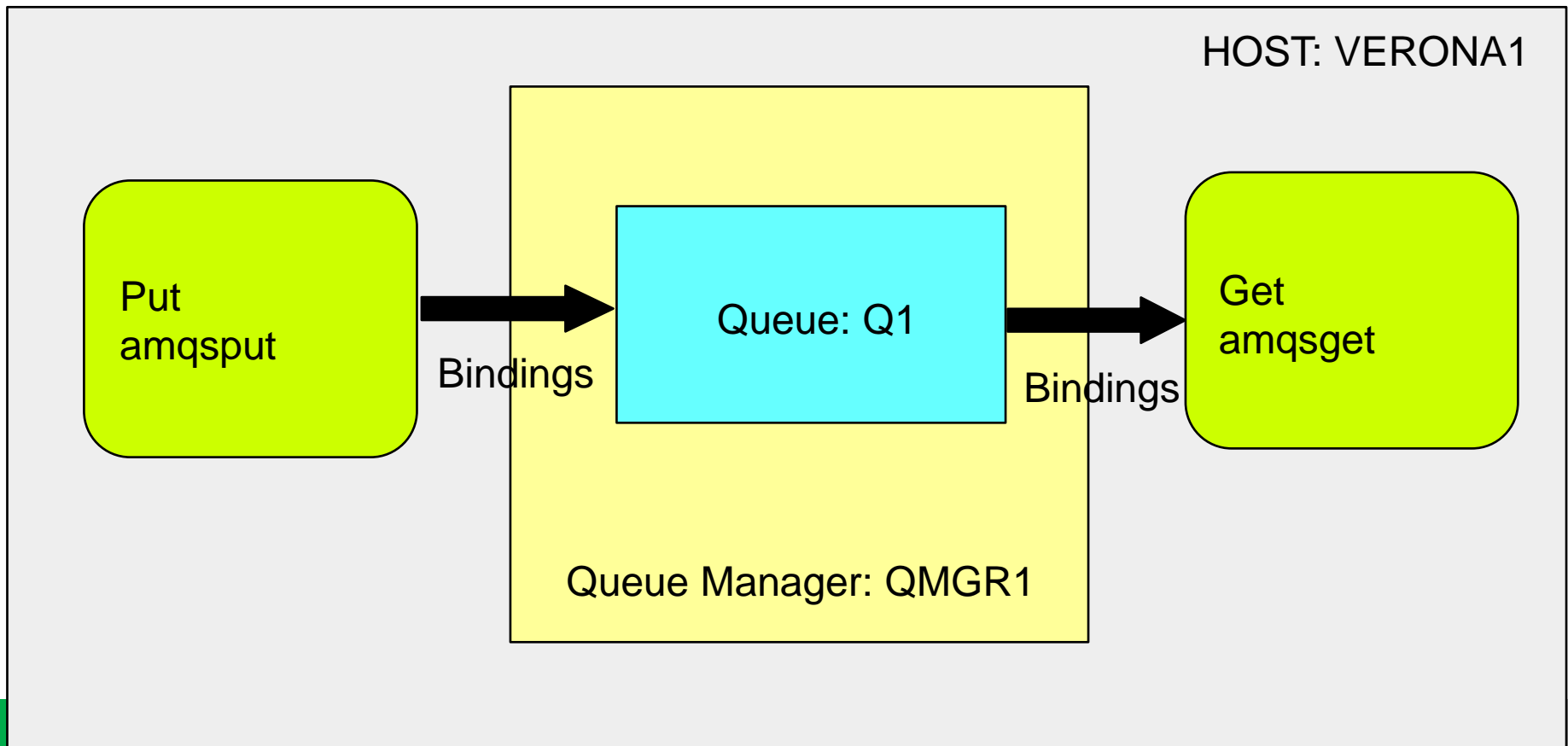
<http://www-01.ibm.com/support/docview.wss?uid=swg27021403>

A Day in the Life of a WebSphere MQ Transmission Queue



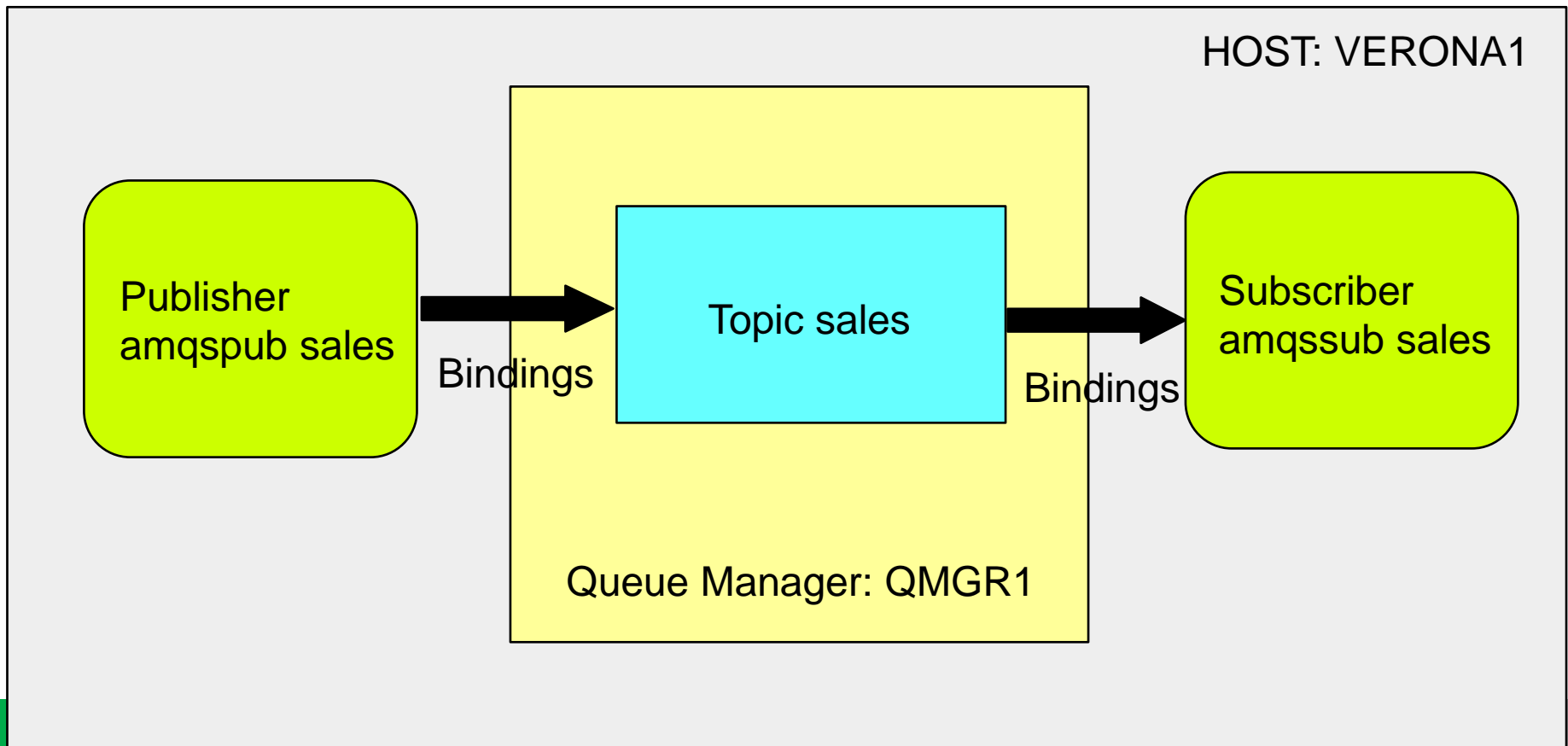
## Local queue, Put/Get using bindings

Local queue - Put and Get are done within the host  
(using “bindings”)



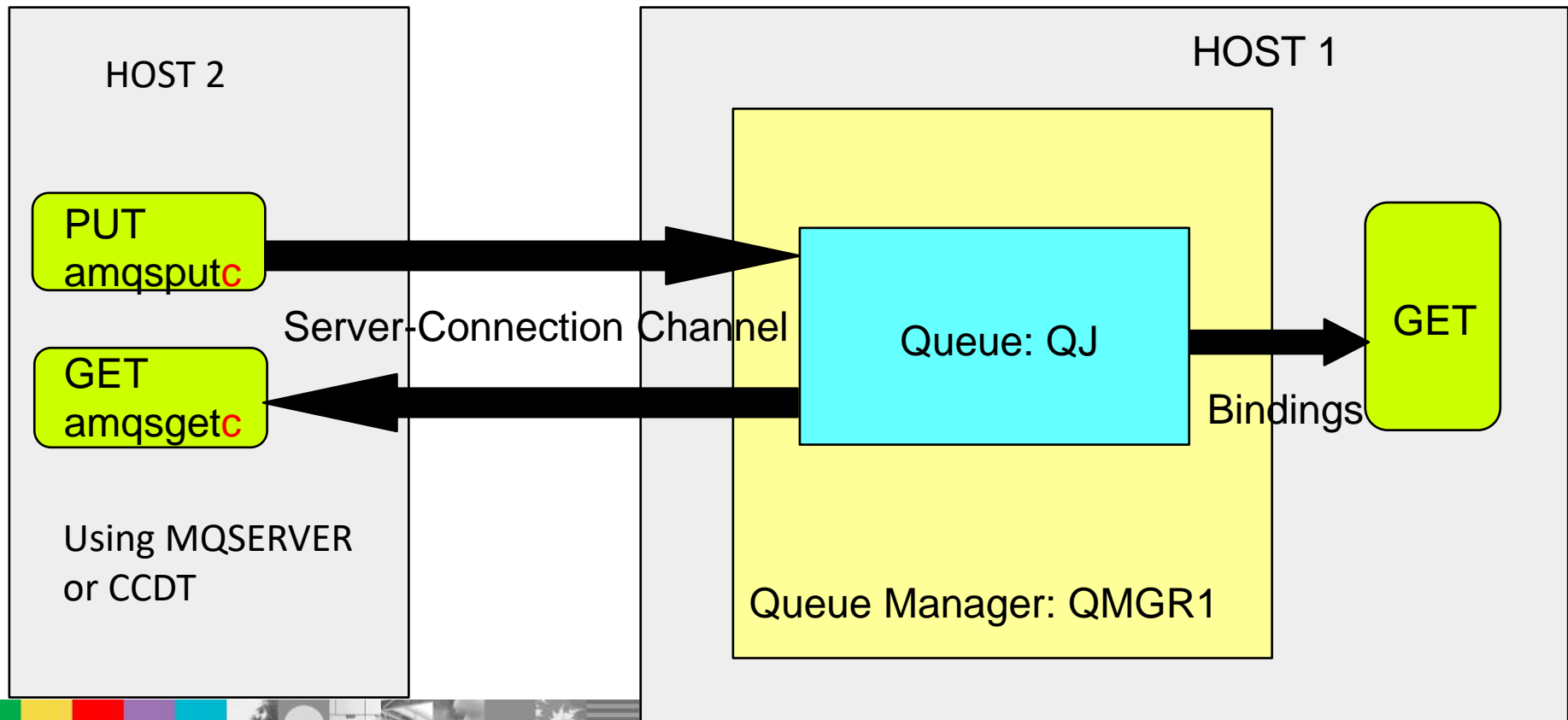
## Local topic, Pub/Sub using bindings

Local topic - Pub and Sub are done within same local host (using “bindings”)



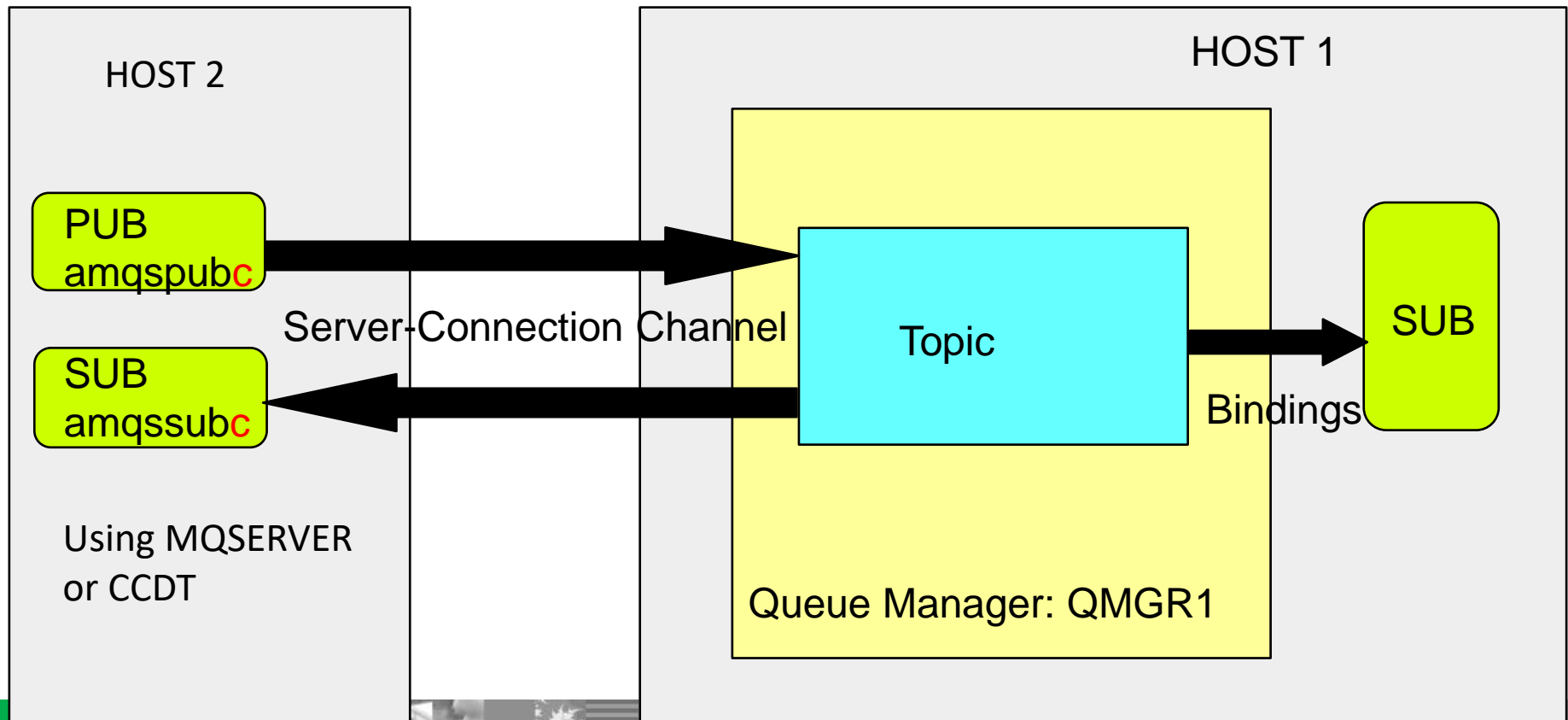
## Local queue, server-connection channel

Local queue (to QMGR1) - Put and Get are done from another host (using “client” or “network” conn)



## Local topic, server-connection channel

Local topic (to qmgr) - Pub and Sub are done from another host (using “client” or “network” connection)



## Using two queue managers

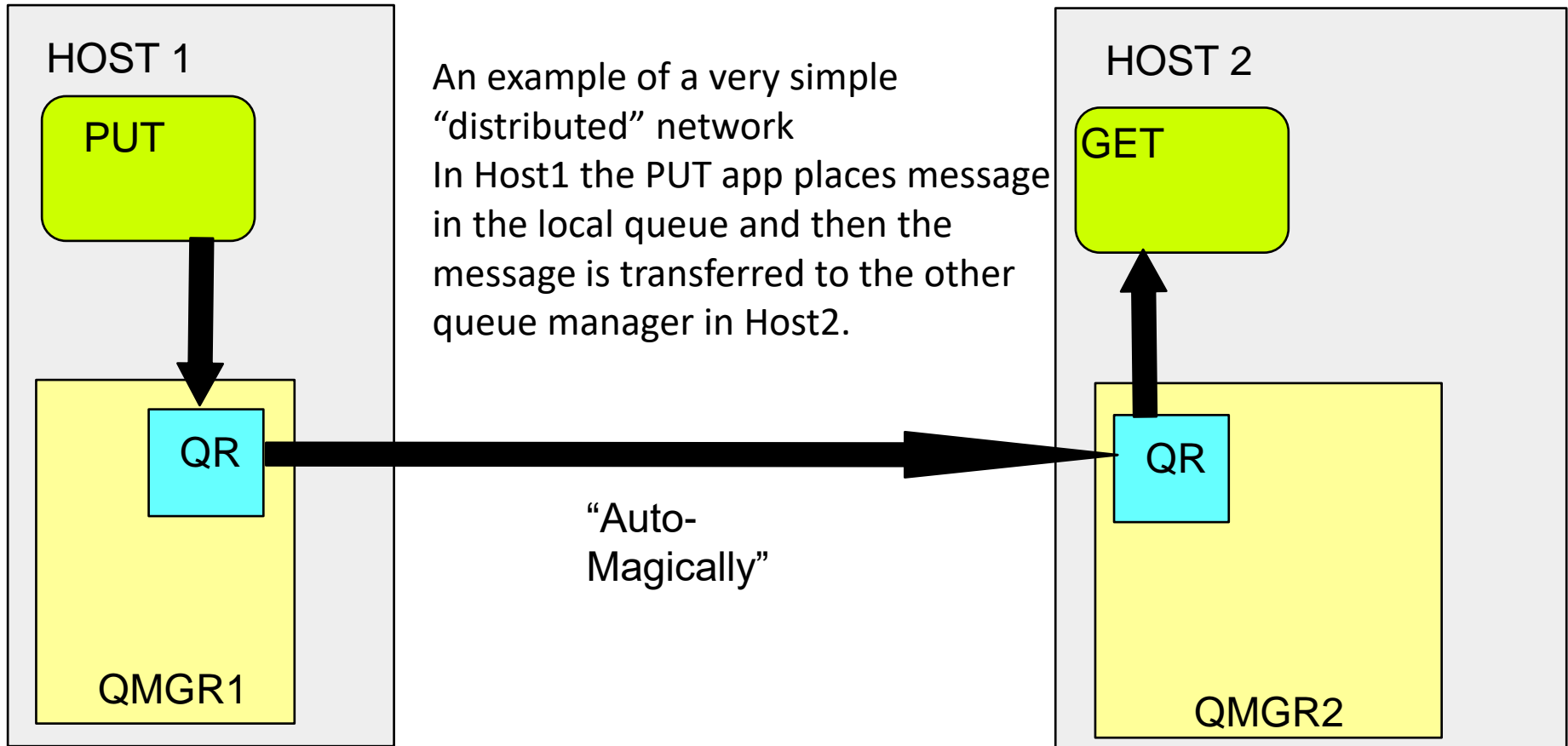
The previous slides showed only 1 queue manager, which is OK for activity that is local or relatively local (close proximity).

For occasional remote access from far away clients would be still ok.

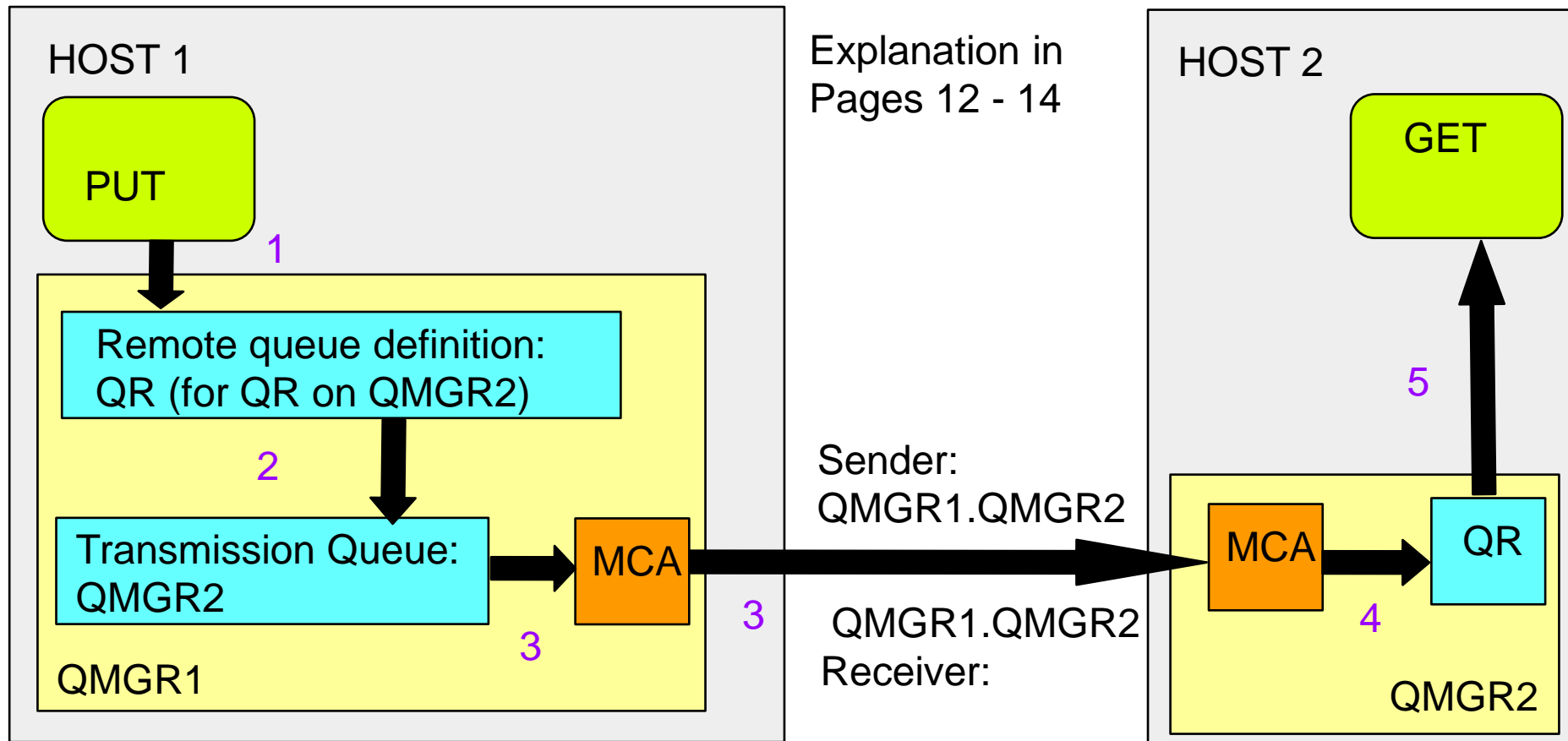
But for constant remote access from many clients from far places, this is not efficient.



## Using 2 queue managers



# Sending 1 message via XMITQ-MCA



# Notes: Technote with steps to connect 2 queue managers

<http://www-01.ibm.com/support/docview.wss?uid=swg21470997>

## **Commands to setup both ways communication between 2 queue managers via Sender and Receiver channels**

n  
o  
t  
e  
s

Let me tell you a suggestion on how I use the commands in the technote:  
I copy the commands from the technote into a text editor.

Then I do a global search-replace on the names of the queue managers, queues, etc.

I open 2 command prompts, one for each queue manager.

Then I copy each command from the text editor and paste them into the runmqsc session for each queue manager.

At the end of the process, all the necessary MQ objects will be created.

You NEED to test the setup!

The technote has instructions on how to test and verify that the objects were configured properly.



# Notes: Overview of the components of distributed queuing - 2

## notes

If both applications are connected to the same queue manager then no inter-queue manager communication is required, and the target queue is described as local to both applications.

However, if the applications are connected to different queue managers, two MCAs and their associated network connection are involved in the transfer, as shown in the figure. In this case, the target queue is considered to be a remote queue to the putting application.

The sequence of events is as follows:

1. The putting application issues MQOPEN and MQPUT calls to put messages to the target queue.
2. During the MQOPEN call, the name resolution function detects that the target queue is not local, and decides which transmission queue is appropriate. Thereafter, on the MQPUT calls associated with the MQOPEN call, all messages are placed on this transmission queue.

# Notes: Overview of the components of distributed queuing - 3

## notes

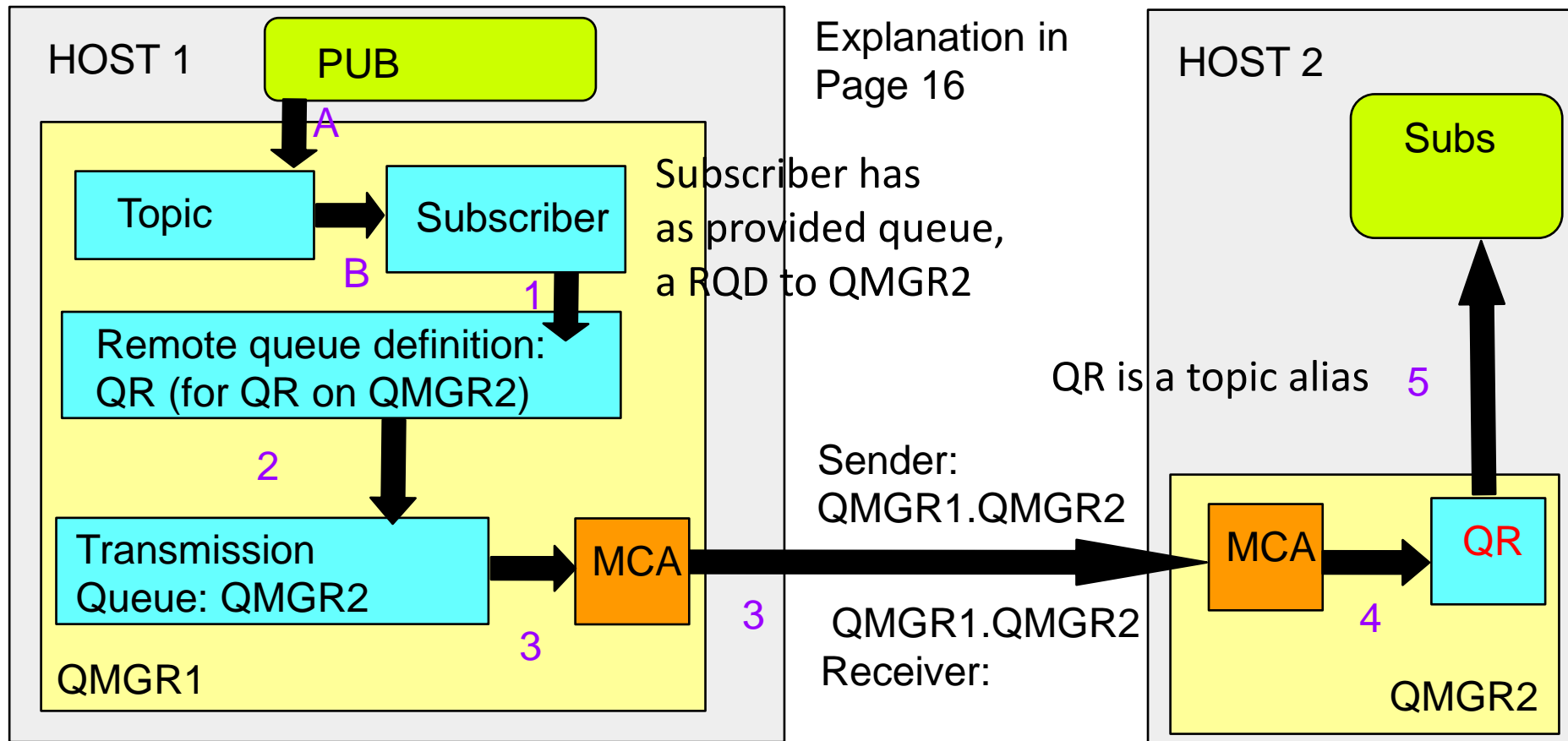
3. The sending MCA gets the messages from the transmission queue and passes them to the receiving MCA at the remote computer.
4. The receiving MCA puts the messages on the target queue, or queues.
5. The getting application issues MQOPEN and MQGET calls to get the messages from the target queue.

Note: Only step 1 and step 5 involve application code; steps 2 through 4 are performed by the local queue managers and the MCA programs. The putting application is unaware of the location of the target queue, which could be in the same processor, or in another processor on another continent.

**The combination of sending MCA, the network connection, and the receiving MCA, is called a message channel, and is inherently a unidirectional device.**

Normally, it is necessary to move messages in both directions, and two channels are set up for this, one in each direction.

# Pub/Sub 1 message via XMITQ-MCA



# Notes: Pub/sub

n  
o  
t  
e  
s

<https://developer.ibm.com/answers/questions/402841/publishing-in-one-queue-manager-and-subscribing-in.html>

Publishing in one queue manager and subscribing in another, without using clusters nor hierarchies.

I tested the setup described in the above dwAnswers article and it works!

Create a subscriber in Host1/Qmgr1 for the desired topic AND specify a Provided Queue.

The Provided Queue is really a Remote Queue Definition (RQD)

A. A Publisher publishes a message into a topic.

B. A copy of the message is sent to the subscribers.

One of the subscribers receives the message into the RQD.

1 – 3 and then the message is moved to the XMITQ and to the other queue manager.

4. At the receiving queue manager, instead of using a local queue to receive a message, use a TOPIC ALIAS!

Thus, when the message arrives to the queue manager, instead of arriving to a queue, it is treated as a published message for a topic.

5. Then the queue manager will send a copy of the message to the subscribers.

# Disadvantages

Even though, this method is doable, it is brute-force and it is not really recommended.

It requires a lot of administration in the queue managers.

It is not flexible.

It is not extendible.

New topics require their RQD in qmgr1 and topic alias in qmgr2



# Publish/subscribe hierarchies

An extension of the approach of connecting queue managers via server and receiver channels (using transmission queues) is to use:

## **publish/subscribe hierarchies**

A publish/subscribe hierarchy is a set of queue managers connected by channels into a **hierarchical structure**.

Each queue manager identifies its **parent** queue manager



## publish/subscribe hierarchies

For the 2 queue managers that we have already connected, we can form a hierarchy.

```
runmqsc QMGR2
```

```
ALTER QMGR PARENT(QMGR1)
```



QMGR1  
is the PARENT  
of QMGR2

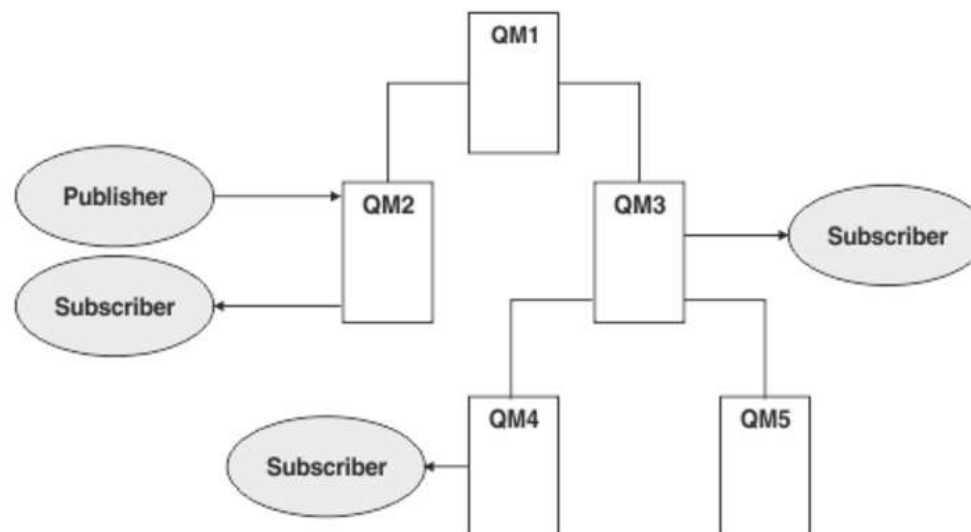
# Publish/subscribe hierarchies

[https://www.ibm.com/support/knowledgecenter/SSFKSJ\\_9.0.0/com.ibm.mq.pro.doc/q005120\\_.htm](https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.0.0/com.ibm.mq.pro.doc/q005120_.htm)

IBM MQ > Technical overview > Publish/subscribe messaging >  
Distributed publish/subscribe networks

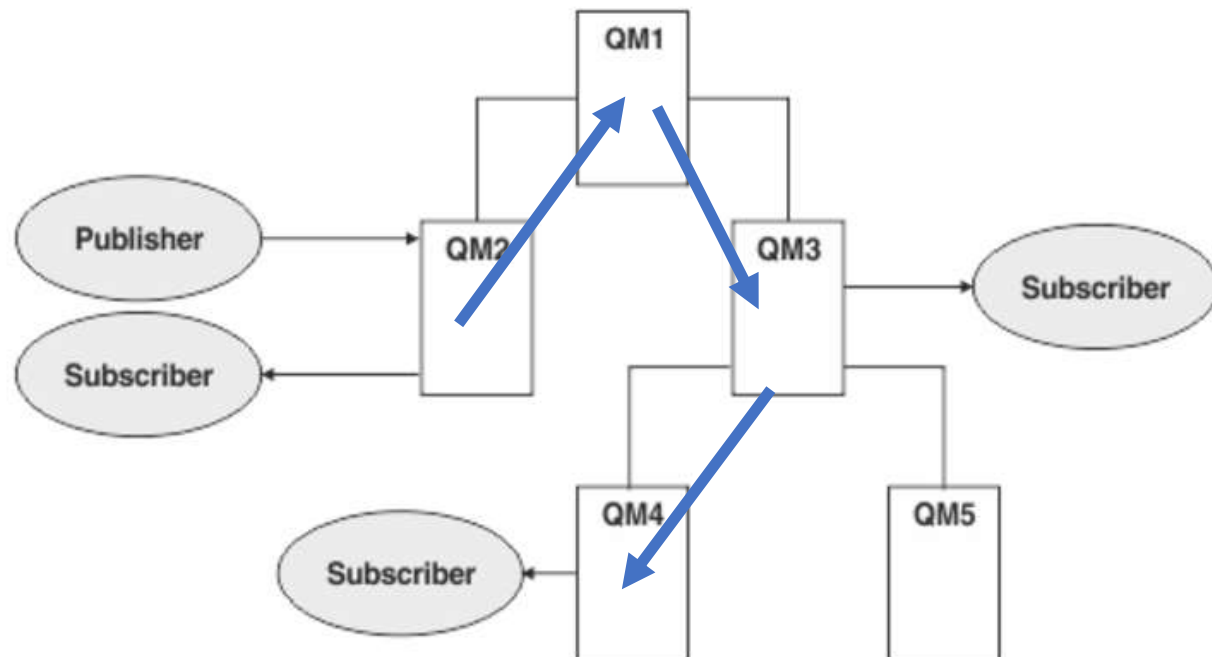
Publishers and subscribers to a topic can connect to any queue manager in the hierarchy.

Messages flow between the queue managers.



# Publish/subscribe hierarchies

Publications delivered to subscribers on QM3 / QM4 have been routed from QM2 to QM1, then onto QM3, and finally to QM4.



## Publish/subscribe hierarchies

Hierarchies give you direct control over the relationships between every queue manager in the hierarchy.

This allows fine-grained control over the routing of messages from publishers to subscribers.

It is especially useful when routing between queue manager networks with restricted connectivity.

**BUT**

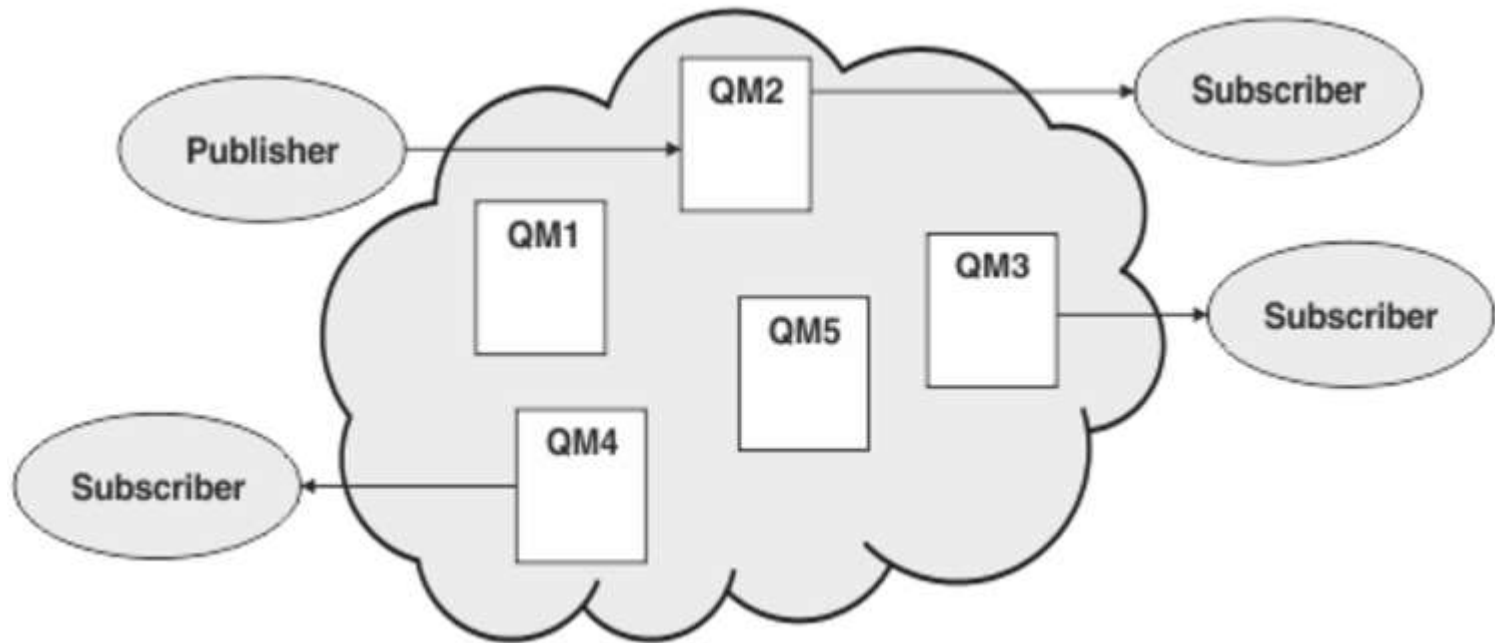
If an intermediary queue manager stops, then there is NO flow to subsequent qmgrs in the hierarchy

If QM3 is down, then QM4 will not receive messages



# Pub/Sub Clusters

A Pub/Sub cluster is a standard cluster with one or more topic objects added to the cluster.



# Pub/Sub Clusters

There are two ways to configure how publish/subscribe messages are routed in a cluster:

Direct routing

Topic host routing



## Pub/Sub Clusters: direct routing

When you configure a direct routed clustered topic, messages published on one queue manager are sent directly from that queue manager to every subscription on any other queue manager in the cluster.

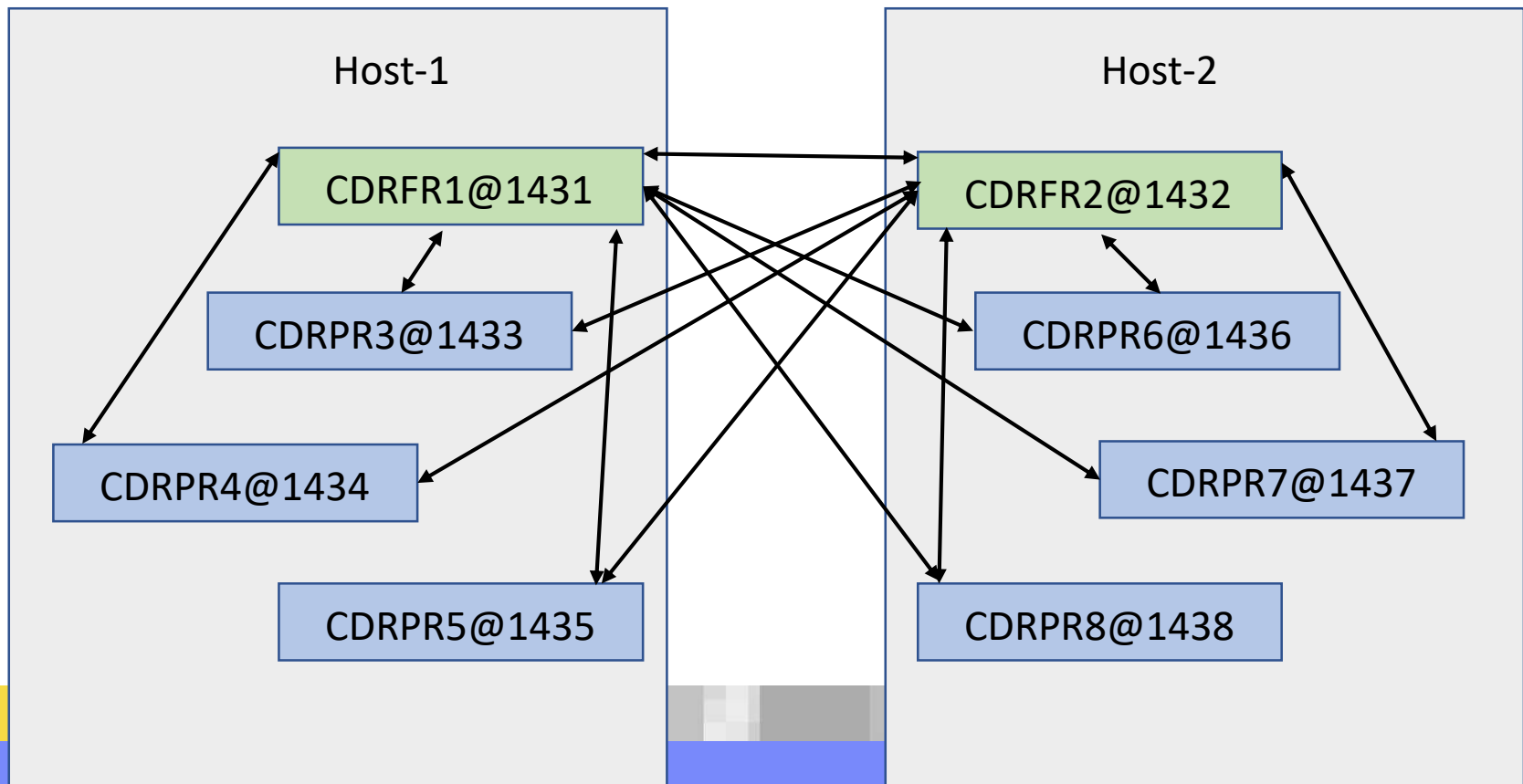
This can provide the most direct path for publications.

But does result in all queue managers in a cluster becoming aware of all other queue managers, each potentially having cluster channels established between them.



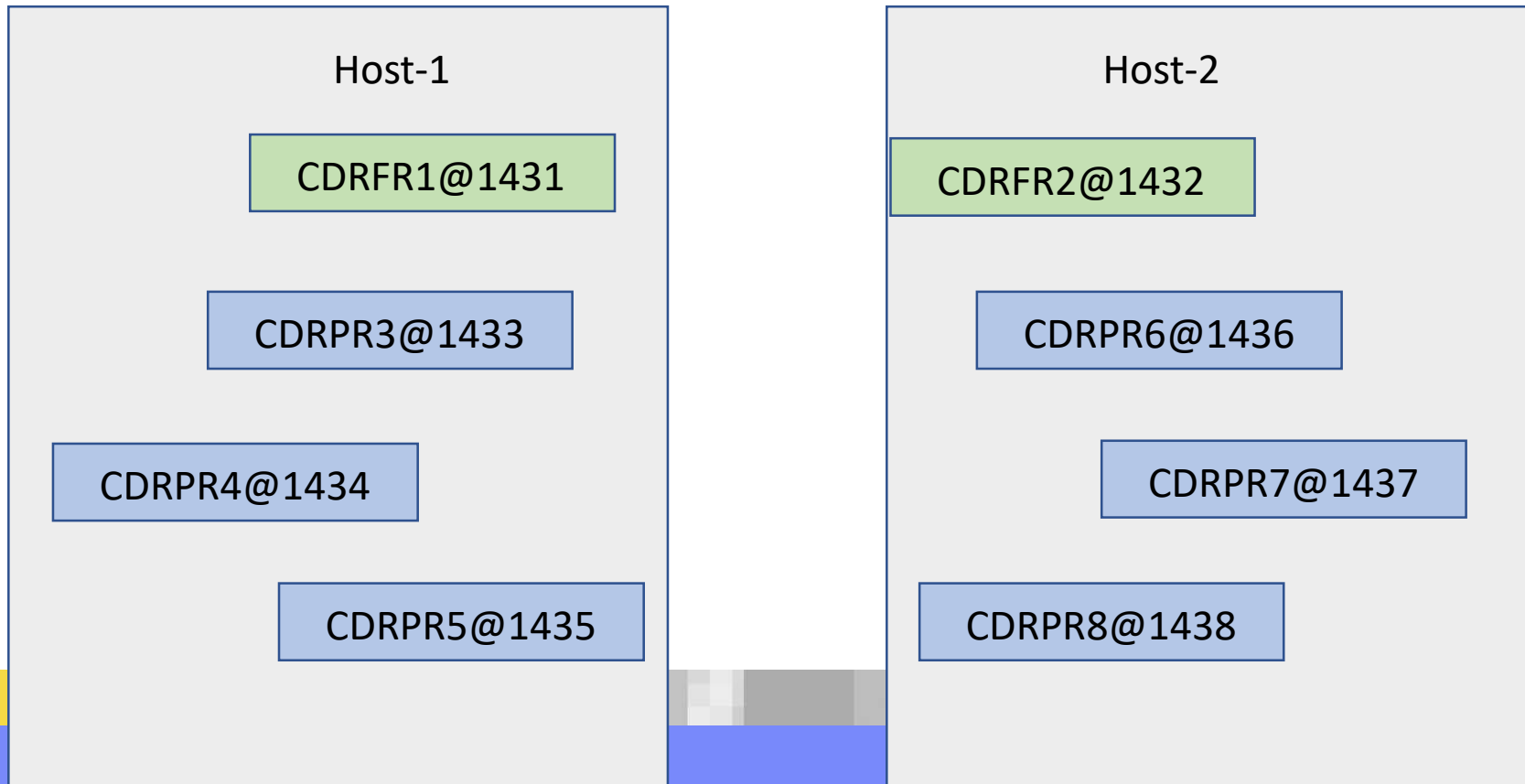
# Cluster topology

Initial configuration for a cluster. 2 FRs and 6 PRs.  
Showing cluster-sender and cluster-receiver channels



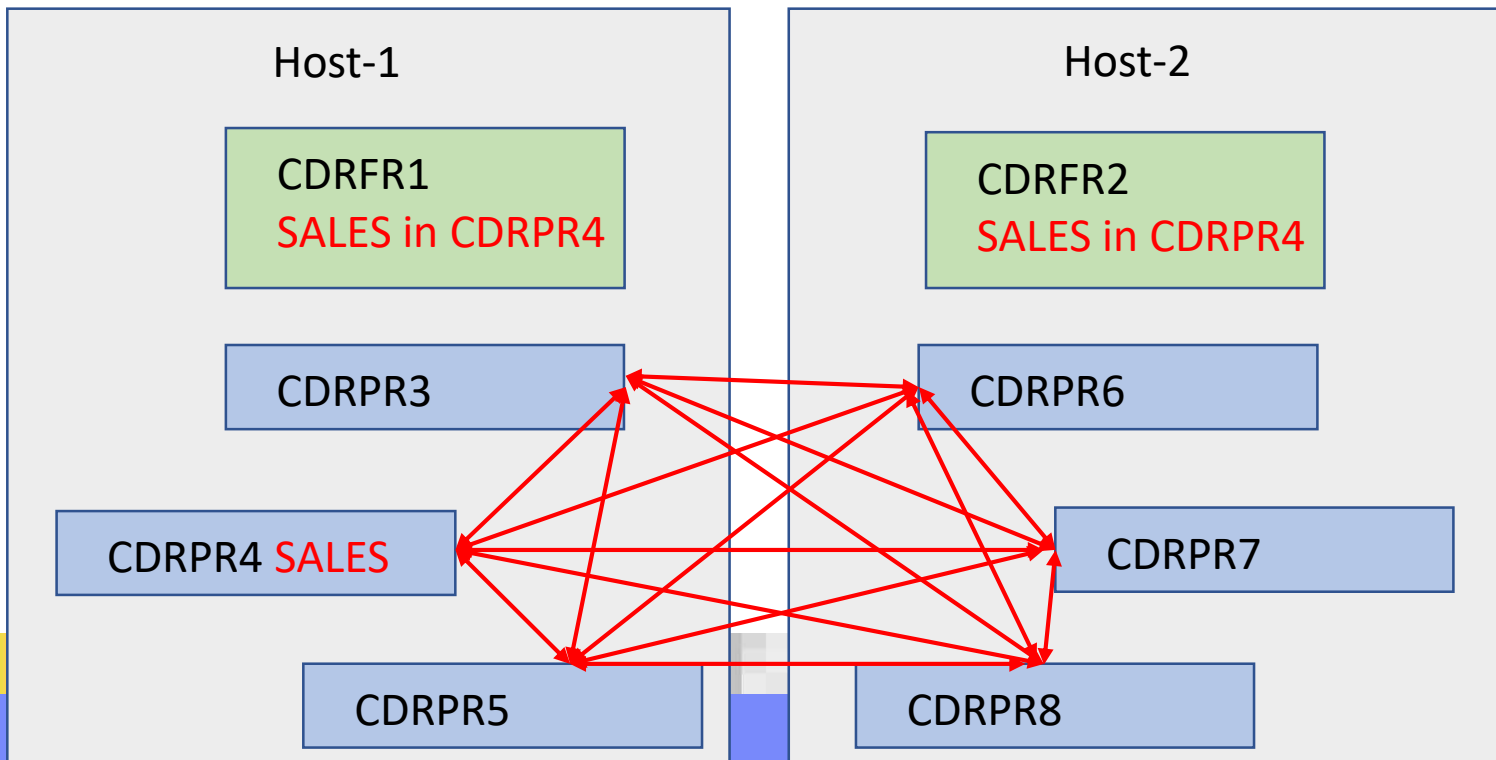
## Simpler view of the topology

To simplify the view,  
let's hide the cluster channels for the infrastructure



## Connections between all PRs

- A clustered topic (SALES) is added at PR CDRPR4
- Automatically, cluster channels are created between ALL the queue managers!



## Direct routing clusters

This configuration relies on MQ maintaining a high level of sharing of information and connectivity between every queue manager in the cluster.

For small and simple networks (that is, a small number of queue managers, and a fairly static set of publishers and subscribers) this could be acceptable.

However, when used in larger or more dynamic environments the overhead might be prohibitive.



# Topic Host routing

Introduced in MQ 8.0 as alternative for the potentially large performance impact of direct routing.

Messages published on one queue manager are sent to another queue manager that hosts a definition of the administered topic object.

That ***topic host queue manager*** routes the message on to every subscription on any other queue manager in the cluster.

Attribute for **Topic**: **CLROUTE(TOPICHOST)**



## Topic Host routing

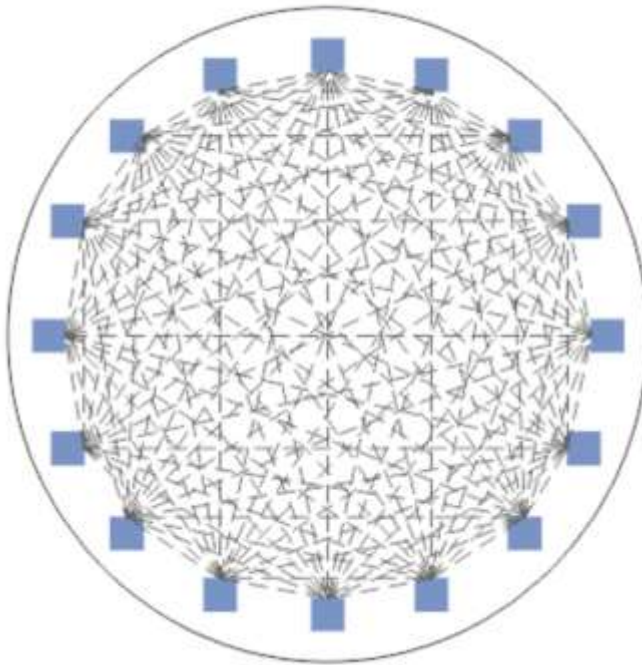
If the publishers or subscribers are not located on the topic host queue managers, this results in a longer route for publications.

The benefit is that only the topic host queue managers become aware of all other queue managers in the cluster, and potentially have cluster channels established with them.

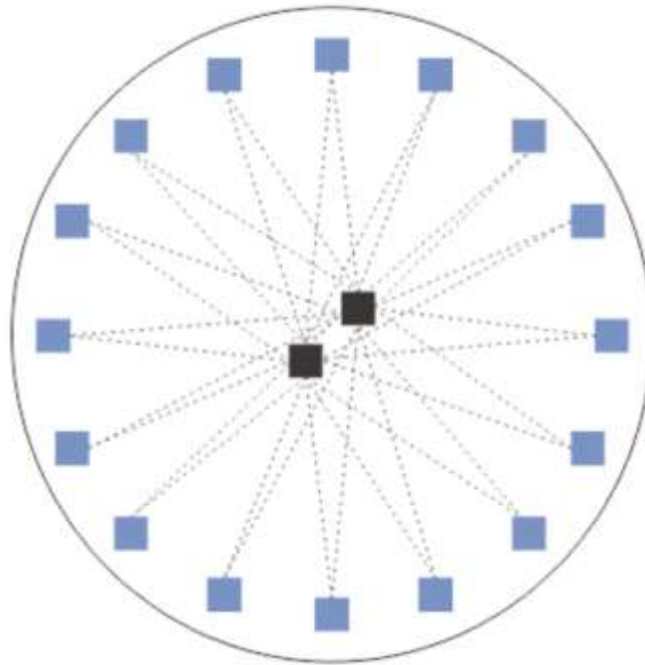


# Topic Host routing vs Direct routing

Left: direct routing



Right: topic host routing



# The End

This is the end of the presentation.

THANKS!!

