

# IBM MQ V9.3 機能と構成

---

## 2. 構成

2023年09月

日本アイ・ビー・エム システムズ・エンジニアリング (株)

## ■ インストール

- ◆ リリース・モデル
- ◆ ライセンス情報
- ◆ 複数バージョンの共存

## ■ MQオブジェクト

- ◆ キュー・マネージャー
- ◆ キュー
- ◆ チャネル
- ◆ リスナー・オブジェクト
- ◆ サービス・オブジェクト
- ◆ トピック・オブジェクト
- ◆ サブ・オブジェクト
- ◆ プロセス定義
- ◆ 名前リスト
- ◆ デフォルト・オブジェクト、システム・オブジェクト

## ■ 可用性構成

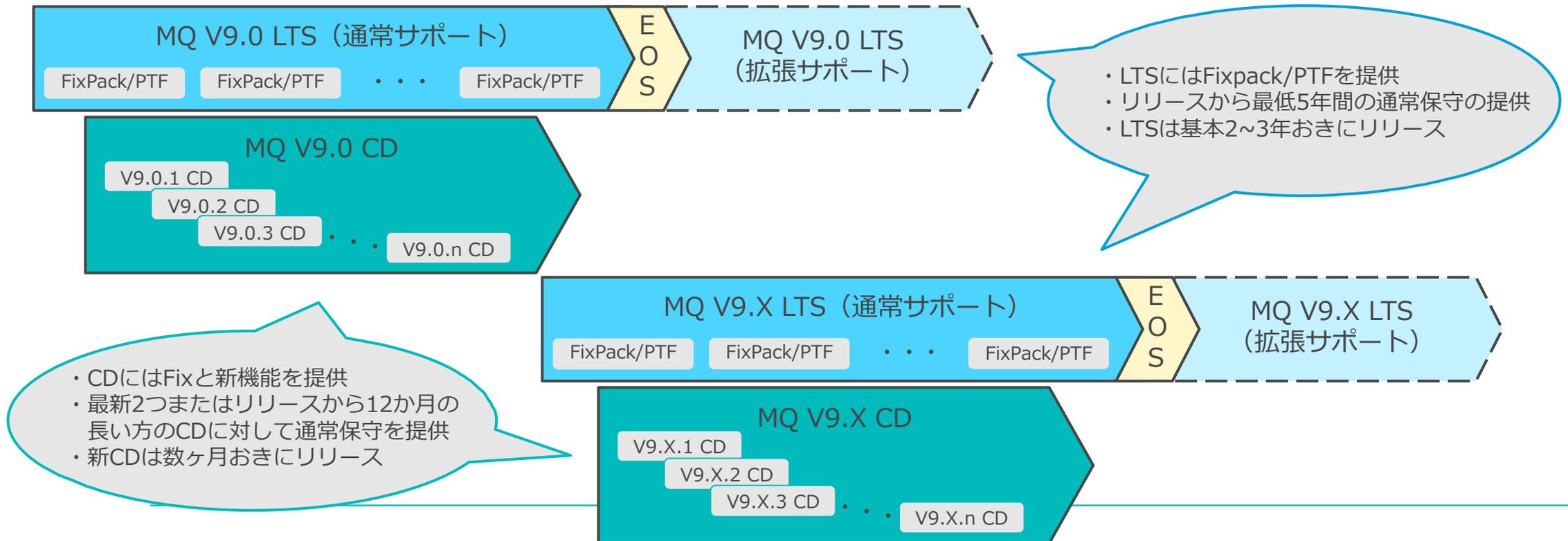
- ◆ MQの高可用性構成
- ◆ マルチインスタンス・キュー・マネージャー
- ◆ RDQM
- ◆ MQクラスター
- ◆ Uniform Cluster
- ◆ <参考> Native HA



# インストラクション

# リリース・モデル

- サポート期間や最新機能の利用有無によってリリース・モデルの検討が必要
- V9.0以降、2種類のリリース・モデルを提供
  - ◆ 継続デリバリー・リリース(Continuous Delivery Release : CDリリース)
    - 新しい機能を短いサイクルで提供
  - ◆ 長期サポート・リリース(Long Term Support Release : LTSリリース)
    - 安定した機能を長期間提供



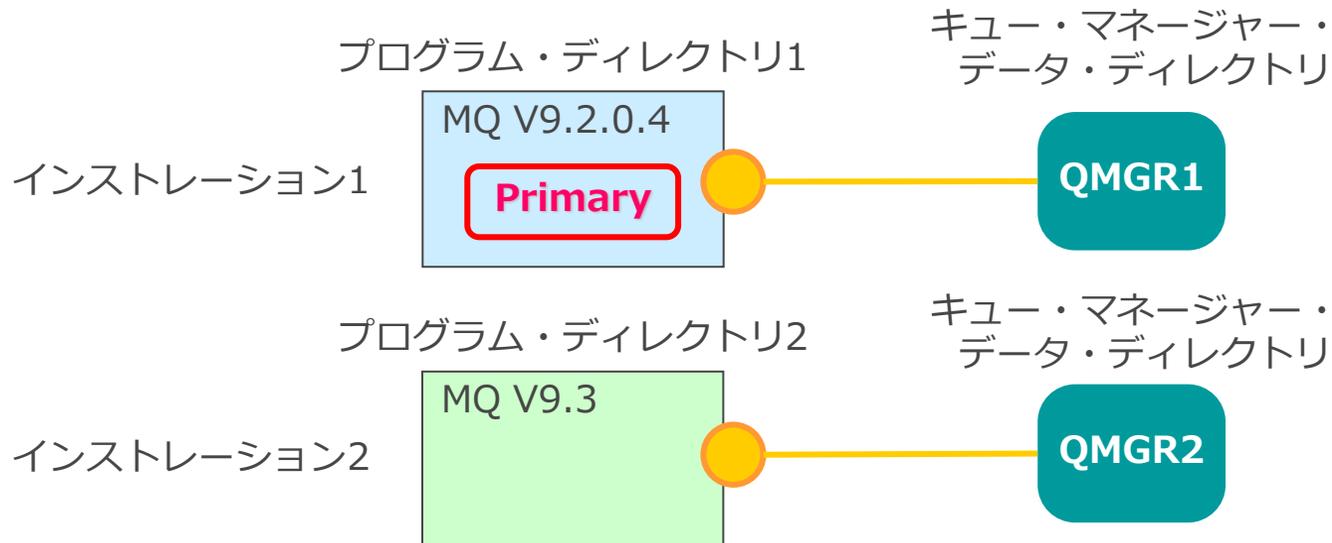
## ■ 導入するMQコンポーネントによってライセンスの検討が必要

### ◆ IBM MQもしくはIBM MQ Advanced

IBM MQ 製品機能		IBM MQ	IBM MQ Advanced
IBM MQ client	MQクライアント機能	✓	✓
Telemetry Service	MQTTプロトコルを使用した、軽量なパブリッシュ/サブスクライブ処理機能		✓
Advanced Message Security	エンドツーエンドの暗号化を実現する高水準のセキュリティ機能		✓
Managed File Transfer Service	MFT(MQのファイル転送管理機能)のサーバ機能		✓
Managed File Transfer Agent	MFTのエージェント機能		✓
Managed File Transfer Tools	MFTのコマンドラインツール機能		✓
Managed File Transfer Database Logger component	MFTのロガー機能		✓
Replicated Data Queue Manager (RedHat x86-64 only)	外部ディスクなしでMQの高可用性構成(フェイルオーバー構成)を実現する機能		✓
IBM MQ Advanced container	実稼動用のIBM MQイメージをデプロイするためのイメージとオペレーターを提供する機能		✓
IBM MQ Internet Pass-Thru	MQプロトコルをHTTPでトンネリングする機能	✓	✓
IBM Aspera faspio Gateway	遠隔地のメッセージングを高速化するためのゲートウェイ機能		✓
AMQP	メッセージング用のオープンなプロトコルであるAMQPとのブリッジ機能	✓	✓
その他基本機能	MQキュー・マネージャー機能および上記以外の機能	✓	✓

## 複数バージョンの共存

- 1つのOSに複数のバージョンのMQを導入可能
  - ◆ 既存のMQを停止せずに導入可能
  - ◆ 任意のディレクトリを指定して導入可能 (UNIX, Linux)
- 各導入環境に固有の名称(インストール名)が付与される
  - ◆ デフォルトのインストール名は「Installation**N**」(**N**: 数字を順次割り当て)
  - ◆ 任意のインストール名にする場合、規定のルールに従って指定
    - 16文字以内の英文字(a-z、A-Z、大文字/小文字の区別なし)及び数字(0-9)の組み合わせ



## ■ デフォルトで利用するMQインストレーション

### ◆ デフォルトの使用環境をプライマリー・インストレーションとして指定可能

- プライマリー・インストレーションの指定により、明示的なパス指定なくコマンドが実行可能になる
- プライマリー・インストレーションを指定する方法
  - プライマリー・インストレーションは明示的に指定する必要がある
  - プライマリー・インストレーションを指定するためのコマンド

制御コマンド	説明
setmqinst	導入環境の変更（プライマリー・インストレーションの設定、解除）
dspmqinst	導入環境の確認（プライマリー・インストレーションの表示）

- ※Windowsは、導入ウィザード中で「一次インストール」を指定可能

## ■ プライマリー・インストレーション以外

### ◆ プライマリー・インストレーション以外のMQ環境を指定するには、明示的にコマンド環境を初期化する必要がある

- 使用するインストレーションの導入ディレクトリ下で `setmqenv -s` コマンドを実行
  - `.$MQ_INSTALLATION_PATH/bin/setmqenv -s` (UNIX, Linux)
  - `"<MQ_INSTALLATION_PATH>%bin%setmqenv.cmd" -s` (Windows)

## キュー・マネージャとインストールの関連付け

- キュー・マネージャは一つのインストールに関連付けられる
  - ◆ デフォルトで `crtmqm` コマンドを実行したインストールに関連付けられる
  - ◆ 関連付けられていないインストールからキュー・マネージャは操作できない
- キュー・マネージャに関連付けられるインストールは変更可能
  - ◆ 関連付けられるインストールの変更は `setmqm` コマンドで実施
  - ◆ キュー・マネージャとインストールの関連付けは `dspmq -o installation` コマンドで確認可能
    - `dspmq` コマンドのオプション

dspmq オプション	説明
なし	キュー・マネージャの状況を表示 (QMNAME、STATUSのみ)
-o installation	キュー・マネージャのインストールの詳細を表示
-o all	キュー・マネージャの全ての状況情報を表示



# MQオブジェクト

# MQオブジェクト

- キュー・マネージャーによって管理されるIBM MQの構成要素
  - ◆ MQオブジェクトはそれぞれ属性を持ち、それらを変更することで動作の制御が可能
  - ◆ 属性の変更は、管理コマンド、管理REST API、GUI(Webコンソール)などから行うことができる
- MQオブジェクト一覧

MQオブジェクト名	説明
キュー・マネージャー	メッセージ・キューイング・サービスをアプリケーションに提供 MQオブジェクトを管理
キュー	メッセージを保持する論理的な箱
チャンネル	キュー・マネージャー間、MQクライアント~キュー・マネージャー間のコミュニケーション・リンク
リスナー・オブジェクト	チャンネル起動要求を受け取り、関連付けられたチャンネルを起動するコンポーネント
サービス・オブジェクト	キュー・マネージャーの起動/停止に合わせ、起動/停止するプログラムを定義
トピック・オブジェクト	Pub/Sub機能において、パブリッシャーとサブスクライバーを紐付ける共通の話題を定義
サブ・オブジェクト	Pub/Sub機能で使用するサブスクリプションを定義
プロセス定義	キュー・マネージャー上のトリガー・イベントに応答して開始されるプロセスを定義
名前リスト	MQオブジェクト名のリスト
認証情報オブジェクト	LDAPサーバへの接続情報(アドレス、ユーザーID、パスワード)を定義 SSL/TLSチャンネル使用時、LDAPに接続し証明書失効リスト(CRL)を参照する場合に必要

- メッセージ・キューイング・サービスをアプリケーションに提供するプロセス
- 主な役割
  - ◆ キュー、チャネルなどのMQオブジェクトを管理
    - 管理コマンドから要求を受け取り、MQオブジェクトの作成、属性の変更、削除処理などを行う
  - ◆ アプリケーションが発行したMQIを受け取り、キューに対する読み込み/書き込み処理を行う
  - ◆ 該当の条件が満たされた場合、トリガー・イベントなどの特殊イベントを生成
  - ◆ MQオブジェクトに対する権限をチェック
- 通常、1システムに1キュー・マネージャーを稼動
  - ◆ 複数キュー・マネージャーの稼動も可能
- 主な属性
  - ◆ CCSID属性
    - キュー・マネージャー間チャネル接続で用いられる文字コードを指定
  - ◆ MAXMSGL属性
    - キュー・マネージャーが扱うことができる最大メッセージ・サイズを指定
  - ◆ DEADQ属性
    - 送達不能メッセージを書き出すデッド・レター・キュー名

# キュー

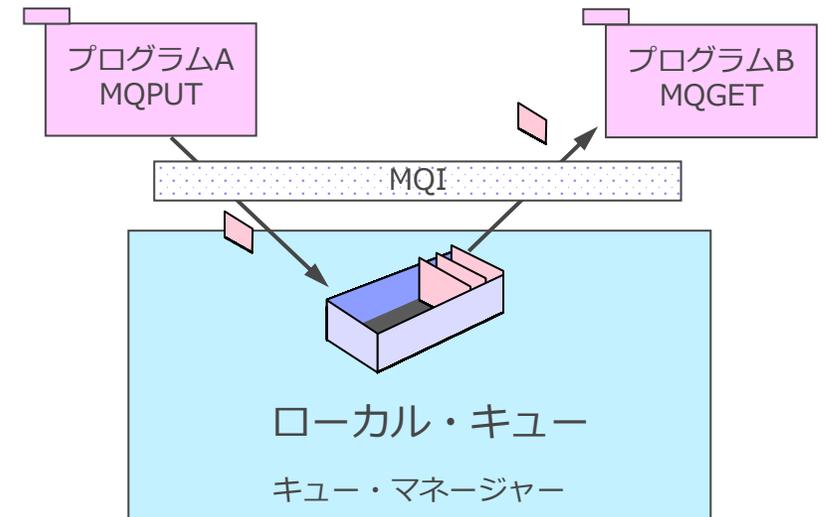
- メッセージを保持する論理的な箱
- アプリケーションはMQIを介し、キューに対してメッセージの読み(GET)/書き(PUT)を行う
- 1キュー・マネージャーに複数作成可能
- キューのタイプ

パラメータ	説明
ローカル・キュー	実際にメッセージを保持するキュー
リモート・キュー定義	リモート・キュー・マネージャー上のキューを指し示す定義 メッセージを書き込むと、トランスミッション・キューにメッセージが書き込まれる
別名キュー	ローカル・キュー、またはリモート・キューへの参照 他のキューを間接的にポイントすることができる
モデル・キュー	動的にローカル・キューを作成する場合のテンプレート

- ◆ ローカル・キュー以外は、実体の無い定義や参照
- ◆ ローカル・キューは、アプリケーションから書き込まれたメッセージを保持
  - 実体はメモリ、ファイルなどのストレージ

## ローカル・キュー

- アプリケーションよりPUTされたメッセージを保持するキュー
- 優先順位を付けてメッセージを保持することが可能
- キュー・マネージャーの再起動をまたがってメッセージを保持することが可能
- 主な属性
  - ◆ MAXDEPTH属性
    - 同時にキューに保持できるメッセージ数の最大数を指定
  - ◆ MAXLENGTH属性
    - アプリケーションがPUTできるメッセージの最大サイズを指定
  - ◆ PUT、GET属性
    - アプリケーションからのPUT、GETの可否を指定
  - ◆ DEFPRTY属性
    - メッセージのデフォルト優先順位を指定
  - ◆ DEFPSIST属性
  - ◆ メッセージのデフォルト永続性を指定



## ■ 特殊な特別な目的を持つローカル・キュー

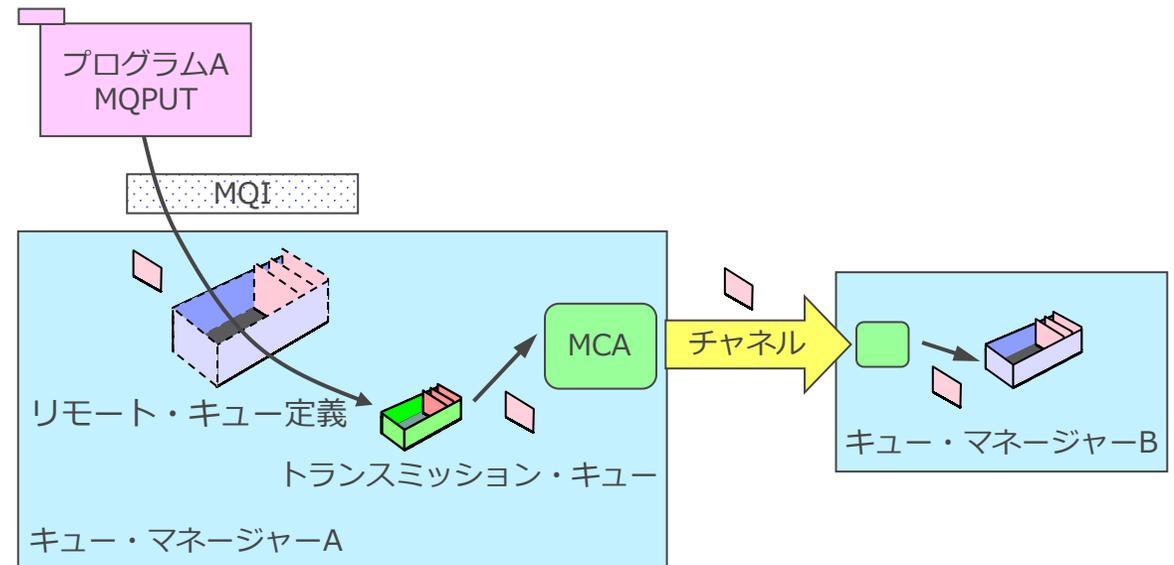
特別な目的をもつ ローカル・キュー名	説明
トランSMission・キュー	<ul style="list-style-type: none"><li>・リモート・キュー・マネージャー宛てのメッセージを一時的に保管する転送用のキュー</li><li>・宛先のリモート・キュー・マネージャーごとに、最低1つ定義することが必要</li><li>・通常、アプリケーションからは直接アクセスされない</li></ul>
イニシエーション・キュー	<ul style="list-style-type: none"><li>・トリガー機能に用いられるキュー</li><li>・トリガー条件が満たされた場合、トリガー・メッセージがキュー・マネージャーによって書き込まれる</li></ul>
デッド・レター・キュー (送達不能キュー)	<ul style="list-style-type: none"><li>・下記のような、何らかの問題でキューに書き込めないメッセージが破棄されるキュー<ul style="list-style-type: none"><li>- 転送可能な長さを超えているなどで、送信チャンネルがメッセージを転送できない</li><li>- 未定義、PUT禁止などの理由で受信チャンネルが宛先キューにメッセージをPUTできない</li></ul></li></ul>
システム・コマンド・キュー	<ul style="list-style-type: none"><li>・MQの管理コマンドを受け付けるためのキュー</li><li>・管理コマンドをメッセージとしてMQPUTすることにより、キュー・マネージャーにコマンドを実行させることが可能<ul style="list-style-type: none"><li>- 管理コマンドはPCF形式のメッセージで記述</li></ul></li></ul>
イベント・キュー	<ul style="list-style-type: none"><li>・MQに発生した様々なエラー、警告、その他の事象がMQメッセージとして書き込まれる</li><li>・MQの監視インターフェースとして利用することが可能</li></ul>
システム・デフォルト・キュー	<ul style="list-style-type: none"><li>・キュー・マネージャーのデフォルト属性をもつキュー</li><li>・新しいキューを作成すると、キューマネージャーは該当のシステムデフォルトキューから属性をコピー</li></ul>

## リモート・キュー定義

- リモート・キュー・マネージャー上のキューを指し示す定義
- リモートのキューを意識することなくメッセージをPUTすることが可能
  - ◆ アプリケーションは、ローカルのリモート・キューに対してメッセージをPUT
  - ◆ メッセージは一旦トランスミッション・キューに保管され、チャンネル経由で宛先キュー・マネージャーへ転送される
    - トランスミッション・キューと宛先キュー・マネージャー間のチャンネルが別途必要

### ■ 主な属性

- ◆ RQMNAME属性
  - 宛先キュー・マネージャー名を指定
- ◆ RNAME属性
  - 宛先キューを指定
- ◆ XMITQ属性
  - トランスミッション・キューを指定



# 別名キュー

## ■ 他キューへの別名定義

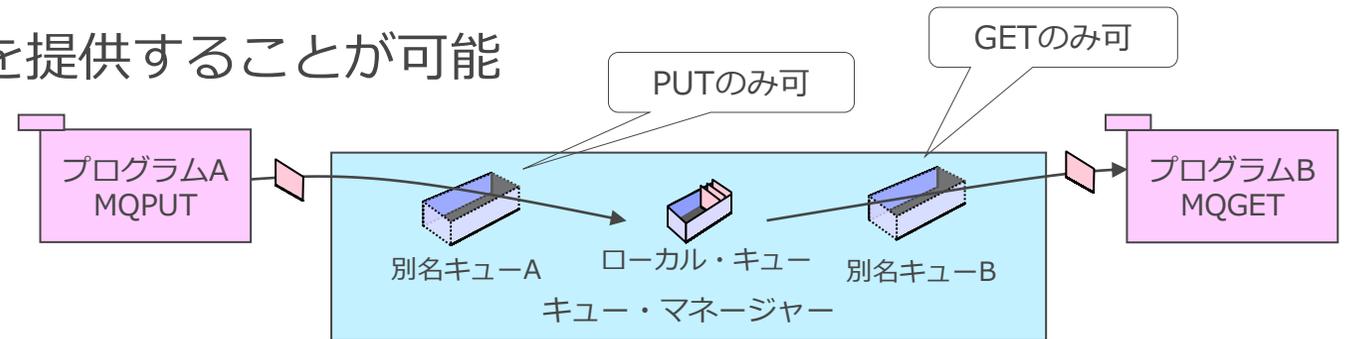
- ◆ 別名キューは、ローカル・キューまたはリモート・キュー定義に紐付けられる
- ◆ 別名キューに接続することで、アプリケーションは間接的にキューにアクセスすることが可能

## ■ 同一のキューに対して複数の別名キューを指定可

## ■ 別名キューに対する別名キューは作成不可

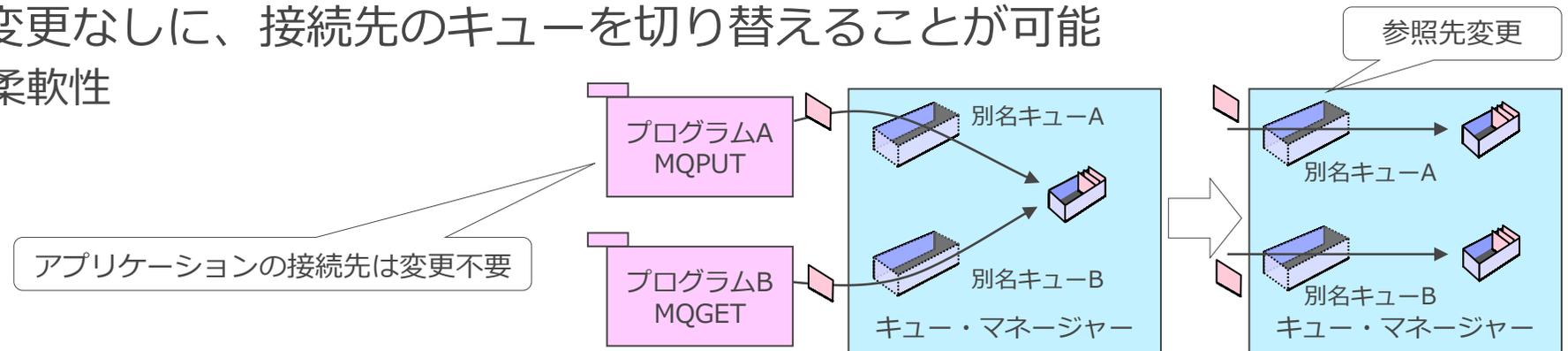
## ■ ユーザーまたは目的に応じて異なる名前を提供することが可能

- ◆ 異なる属性、異なるセキュリティ



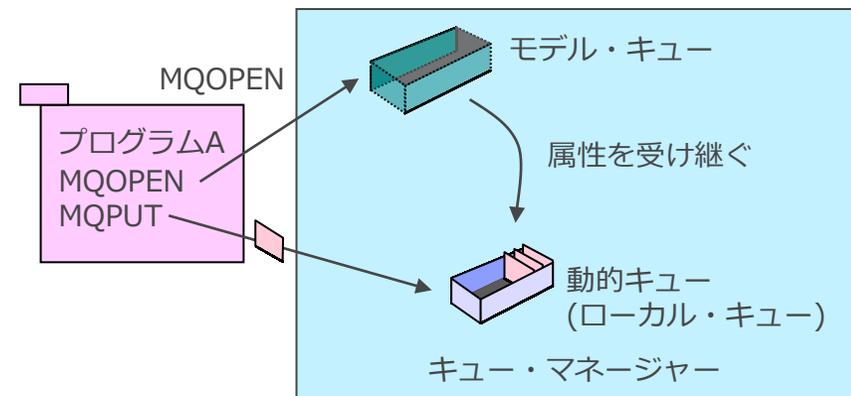
## ■ アプリケーションの変更なしに、接続先のキューを切り替えることが可能

- ◆ 将来的な負荷分散の柔軟性



## モデル・キュー

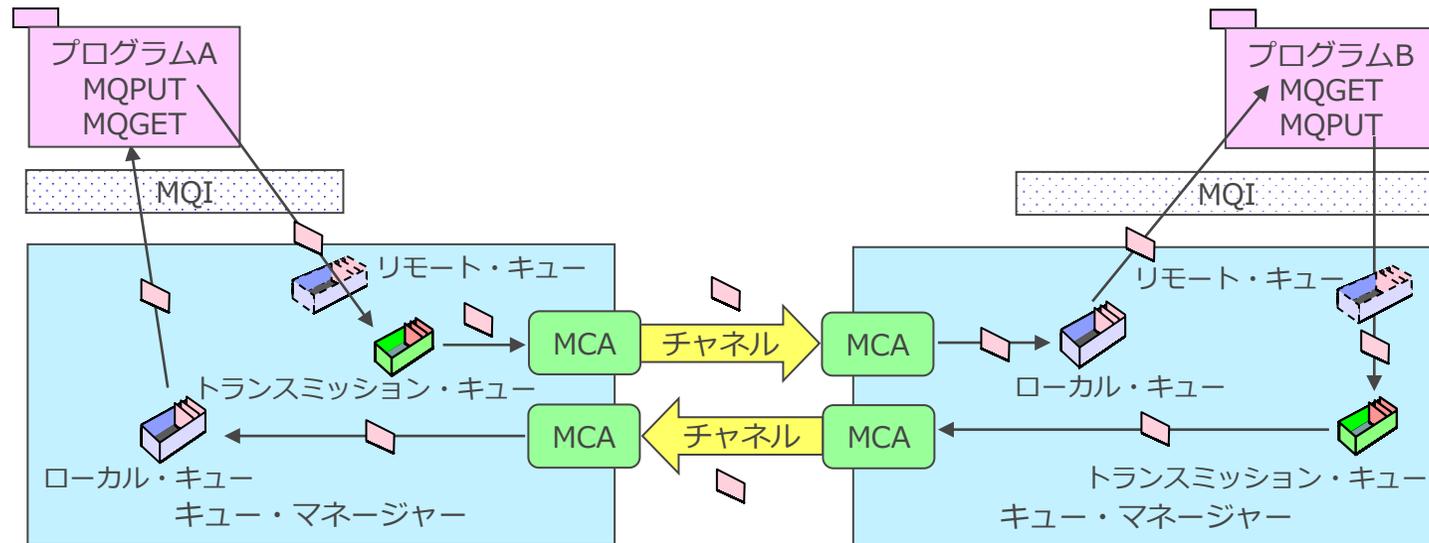
- 動的キューを作成する場合に使用されるキュー属性のテンプレート
  - ◆ モデル・キューをオープンすると、キュー・マネージャーによって動的キューが作成される
  - ◆ 作成されるキューはローカル・キューで、モデル・キューから属性が引き継がれる
- 動的キューの種類
  - ◆ 一時動的キュー
    - ノン・パーシステント・メッセージのみ保持可能
    - キュー・マネージャーの再起動時、アプリケーションがクローズした際にキューは削除される
  - ◆ 永続動的キュー
    - パーシステント、ノン・パーシステント・メッセージの両方を保持可能
    - アプリケーションのクローズ・オプションにより、クローズ時にキューを削除するかどうかを選択可能
- 要求に対する応答を待つキューなど、一時的にキューを使用する目的で利用可能
- モデル・キューの変更はテンプレートの変更となる
  - ◆ 既に作成した動的キューの属性は変更されない
- 主な属性
  - ◆ DEFTYPE属性
    - 一時動的キュー、永続動的キューのどちらかを設定



# メッセージ・チャンネル

## ■ メッセージ・チャンネル

- ◆ キュー・マネージャー間のコミュニケーション・リンク
- ◆ 送信側と受信側でそれぞれチャンネル・オブジェクトの定義が必要
  - チャンネル名は同一である必要がある
- ◆ メッセージの転送は一方方向
  - キュー・マネージャー間で送受信を行う場合は、上り下りで2対のチャンネル定義が必要
- ◆ 両端でMCA(メッセージ・チャンネル・エージェント)が稼動しチャンネルの制御を行う
  - 送信側MCAは、トランSMission・キューからメッセージを取り出し、受信側MCAに転送
  - 受信側MCAは、送信側MCAからメッセージを受け取り、宛先のキューに書き出す



## ■ チャンネル・タイプ

- ◆ MCAが送信側、受信側のどちらであるかをCHLTYPE属性で定義することが必要
- ◆ 4つのチャンネル・タイプがあり、それぞれ役割(送信側/受信側)、チャンネル操作(起動/停止)が異なる

チャンネル・タイプ	役割	チャンネル操作	
		起動	停止
SENDER	送信側	○	○
SERVER	送信側	○ (フル定義の場合)	○
RECEIVER	受信側	×	△*
REQUESTER	受信側	○	△*

\* 自分だけ停止  
送信側がSENDER、フル定義のSERVERの場合、  
送信側は接続再試行状態になる

- ◆ SENDER-RECEIVERの組合せが一般的

## ■ 主な属性

- ◆ CONNAME属性
  - 接続先のIPアドレス(ホスト名)、ポート番号を指定
- ◆ DISCINT属性
  - メッセージ転送が無い場合の自動切断時間を指定
- ◆ SHORTTMR/SHORTRTY/LONGTMR/LONGRTY属性
  - チャンネル切断時のチャンネルの再接続試行回数、再接続試行間隔を指定
- ◆ HBINT属性
  - 障害検知に用いるハートビートの送信間隔を指定

# MQクライアント・チャンネル

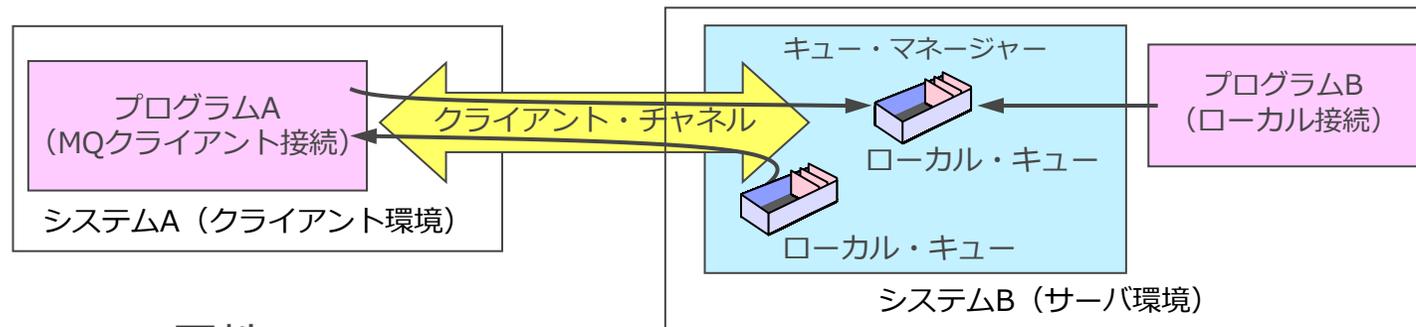
## ■ MQクライアント・チャンネル

- ◆ MQクライアントとキュー・マネージャー間のコミュニケーション・リンク
- ◆ サーバ側でMCAプロセスが稼動しチャンネルの制御を行う
- ◆ メッセージの転送は双方向
- ◆ MQクライアント・アプリケーションからのMQI(MQCONN/MQDISC)呼び出しにより起動/停止

## ■ サーバ側には、サーバ接続チャンネル定義が必要

## ■ MQクライアント・アプリケーションはキュー・マネージャーへの接続情報が必要

- ◆ MQSERVER環境変数、クライアント・チャンネル定義テーブル、プログラム内に記述のいずれか

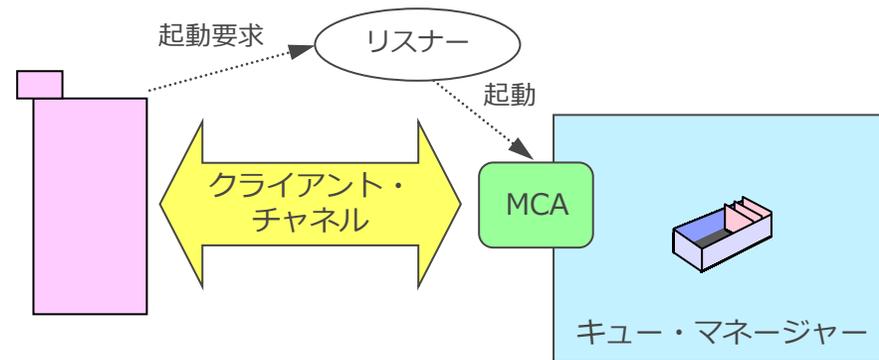


## ■ 主な属性

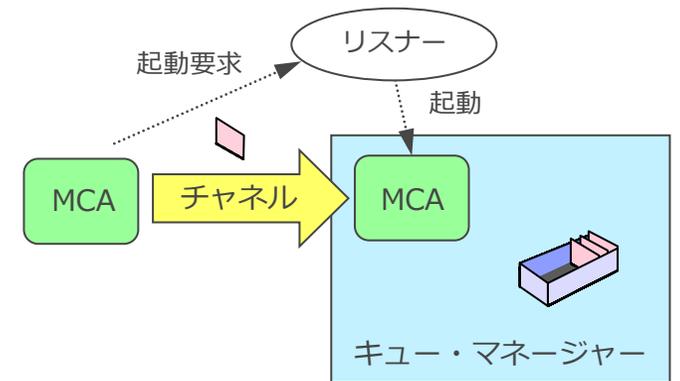
- ◆ MAXINST、MAXINSTC属性
  - 同時に稼動可能なMQクライアント・チャンネル数の上限を指定
- ◆ HBINT属性
  - 障害検知に用いるハートビートの送信間隔を指定

# リスナー・オブジェクト

- MQリスナーは、チャンネルへの接続要求を受け取りチャンネルの起動を行うプロセス
- 起動方法は2通り
  - ◆ runmqtsr 制御コマンドから起動
  - ◆ リスナー・オブジェクトより起動
- リスナー・オブジェクト
  - ◆ キュー・マネージャーの起動/停止時に合わせ、サービス起動することが可能
    - CONTROL属性で設定
    - 手動での起動も可能
- 主な属性
  - ◆ TRPTYPE属性
    - 通信プロトコルを指定
  - ◆ PORT属性
    - 待ち受けるポート番号を指定



<サーバ接続チャンネルの起動>



<メッセージ・チャンネルの起動>

# サービス・オブジェクト

## ■ キュー・マネージャーの起動/停止に合わせ、起動/停止するプログラムを定義

- ◆ 例) キュー・マネージャー起動時、複数のチャンネル起動処理を行う

## ■ 2種類のサービス・オブジェクト

### ◆ サーバ・サービス・オブジェクト

- 起動できるのは1つのインスタンスのみ
- 管理コマンド(DISPLAY SVSTATUS)で稼動状況を確認可能

### ◆ コマンド・サービス・オブジェクト

- 同時に複数のインスタンスを起動できる
- 管理コマンドでの稼動状況確認はできない

## ■ サービス・オブジェクト定義に変数を使用可能

- ◆ /var/mqm/service.env (UNIX) 、 mq\_inst¥service.env (Windows) に定義した変数が置き換えられる

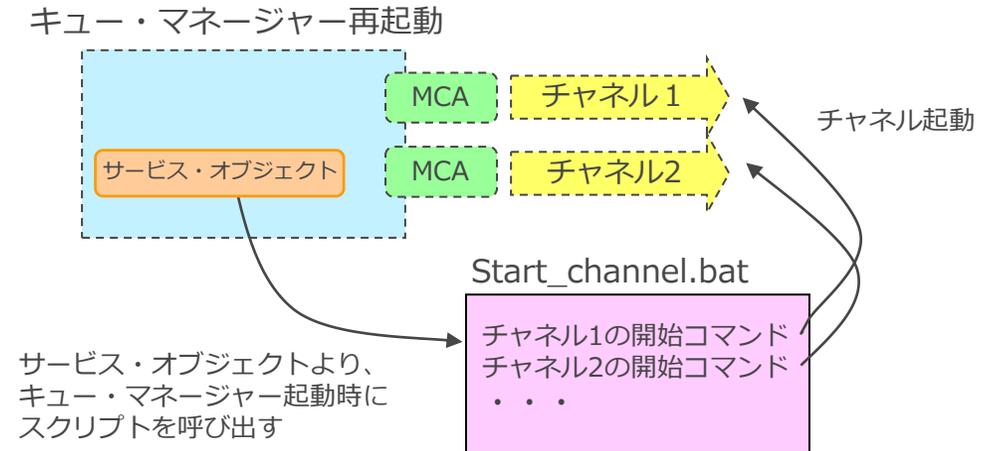
## ■ 主な属性

### ◆ SERVTYPE属性

- サービス・オブジェクトの種類を指定

### ◆ STARTCMD、STARTARG、STOPCMD、STOPARG、STDOUT、STDERR属性

- サービス・オブジェクトより起動するプロセスの起動/停止コマンド、引数、標準出力の出力先を指定



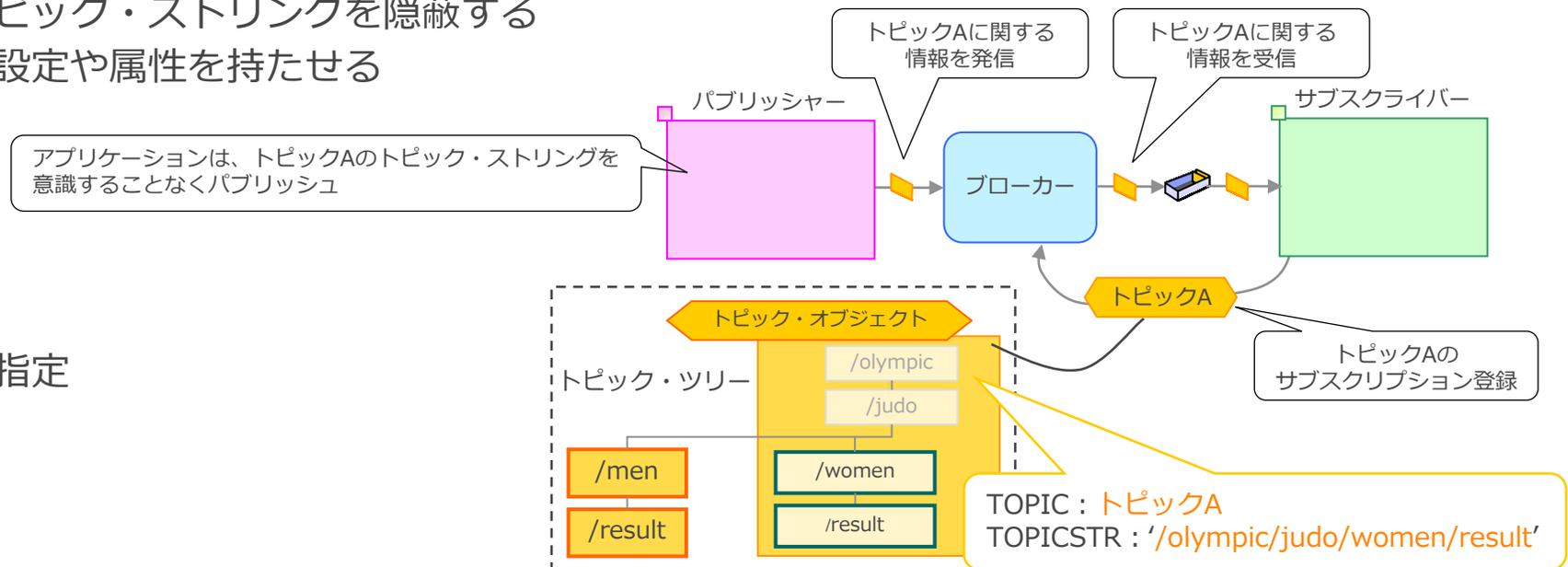
# トピック・オブジェクト

## ■ Pub/Sub機能で使用するトピックを定義

- ◆ Pub/Subとは、情報の提供者(パブリッシャー)と受信者(サブスクライバー)が、トピックを介して行うメッセージング形態の1つ
  - パブリッシャーが発信したパブリケーション(メッセージ)をブローカーが仲介し、サブスクライバーが受信
  - 疎結合なメッセージ連携が可能
- ◆ トピックとは、パブリッシャーとサブスクライバーを紐付ける共通の話題
  - トピックを表す文字列であるトピック・ストリング、もしくはトピック・オブジェクトで指定
- ◆ トピックをMQオブジェクトで定義することで、以下のことが可能になる
  - アプリケーションから宛先トピック・ストリングを隠蔽する
  - トピックにセキュリティの設定や属性を持たせる

## ■ 主な属性

- ◆ TOPICSTR属性
  - トピック・ストリングを定義
- ◆ DEFPSIST属性
  - パブリケーションの永続性を指定



# サブ・オブジェクト

## ■ Pub/Sub機能で使用するサブスクリプションを定義

- ◆ サブスクリプションとは、サブスクライバーがパブリケーションを受信するための購読要求
- ◆ 通常は、アプリケーション内からMQSUBを実行してサブスクリプション登録
- ◆ サブ・オブジェクトを定義することでアプリケーションを介さずにサブスクリプションを定義し、トピックに対してパブリッシュされたメッセージを任意のキューに投入することが可能

## ■ 主な属性

### ◆ DEST属性

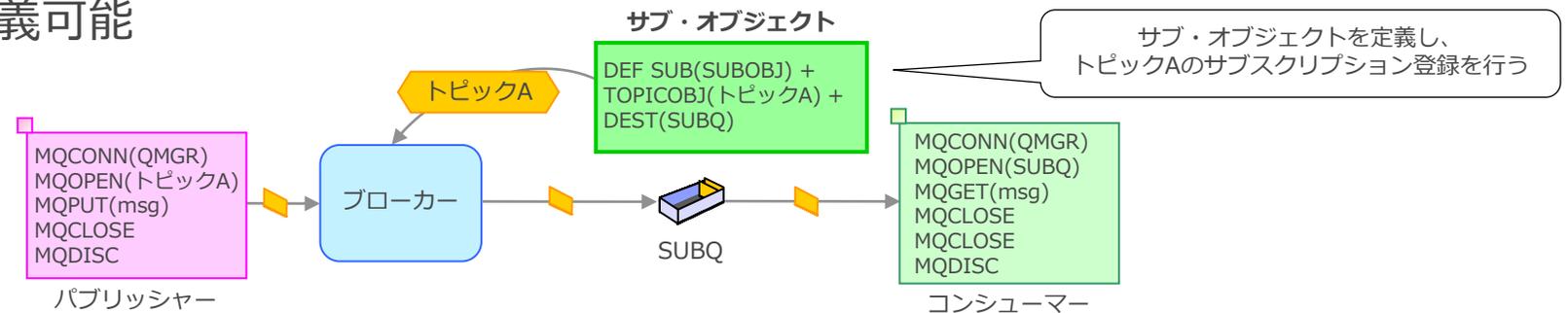
- サブスクライバーがパブリケーションを受信するためのキュー(サブスクライバー・キュー)を指定

### ◆ TOPICOBJ属性

- サブスクリプションが使用するトピック・オブジェクト名を指定

## ■ 考慮点

- ◆ Pub/Subブローカー非稼動時にはパブリッシュ/サブスクライブはできず、サブ・オブジェクト定義も不可
- ◆ 継続サブスクリプションのみ定義可能



# プロセス定義

- トリガー機能使用時、トリガー・イベントに反応して開始するアプリケーションを定義
- トリガー機能の動作
  - ◆ キュー・マネージャーは、トリガー設定のローカル・キューがトリガリング条件を満たすと、トリガー・メッセージをインシエーション・キューにPUT
  - ◆ プロセス情報がトリガー・メッセージにセットされる
  - ◆ トリガー・モニターはトリガー・メッセージ内のプロセス定義をもとにプロセスを起動
- 主な属性

- ◆ APPLICID属性

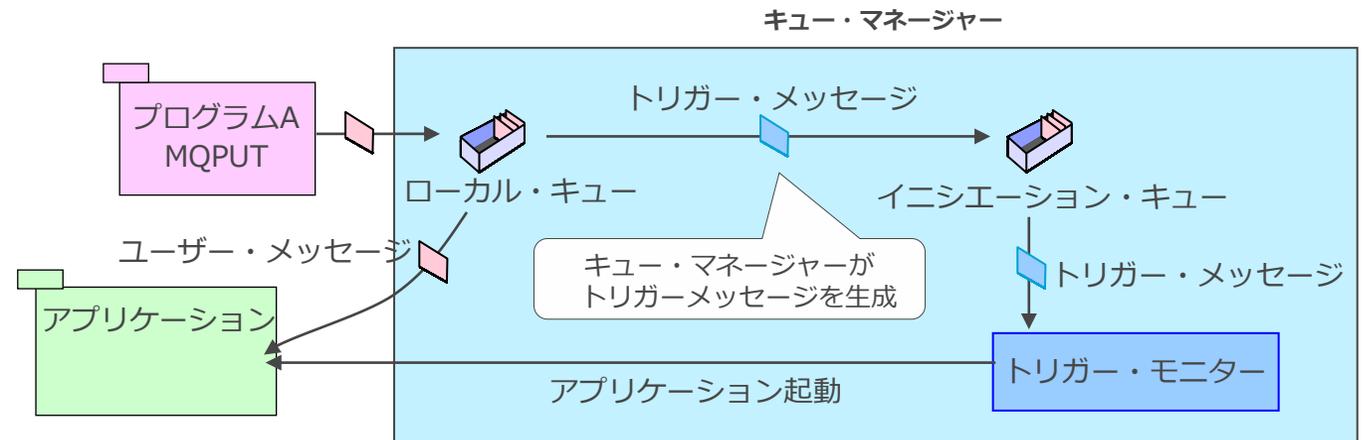
- 開始するアプリケーションの名前を指定

- ◆ ENVRDATA属性

- アプリケーションの環境を設定

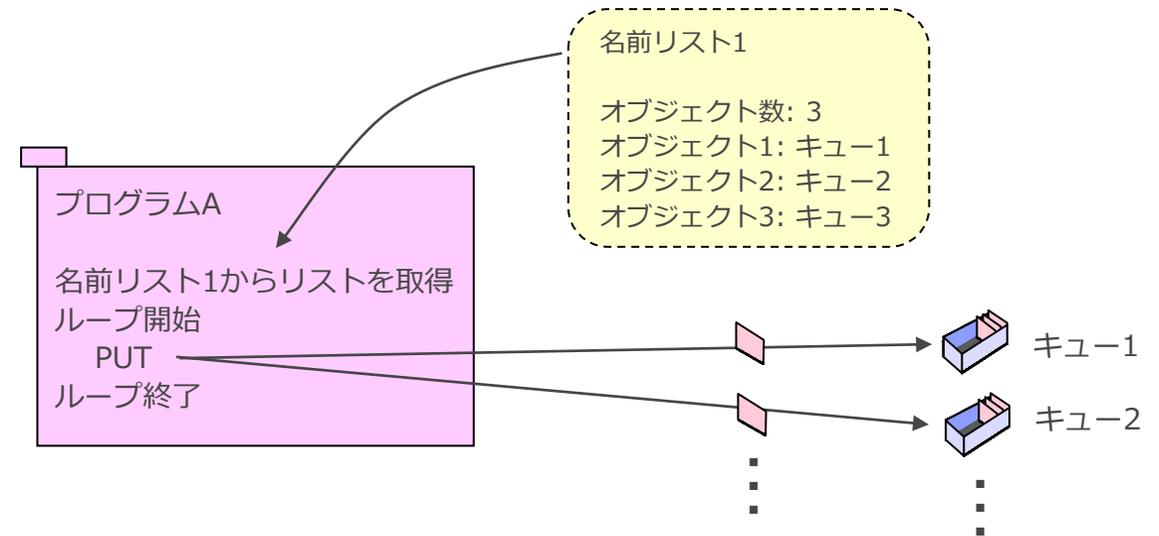
- ◆ USERDATA属性

- アプリケーションのユーザー情報を設定



# 名前リスト

- 複数のMQオブジェクト名をリスト化したオブジェクト
- 複数のキューをグループ化し、アプリケーションから利用するといった利用が可能
  - ◆ 名前リストを照会し、宛先キュー名を取得
  - ◆ 名前リストに含まれるキューにアクセスし、PUT/GETを行う
  - ◆ アプリケーションの変更をすることなく、宛先キューを変更することが可能
- キューマネージャーが複数のクラスターに属する場合、クラスター名の指定に使用
- 主な属性
  - ◆ NAMCOUNT属性
    - 名前リストに指定するオブジェクトの数を指定
  - ◆ NAMES属性
    - 名前リストに含まれるオブジェクト名を指定



# デフォルト・オブジェクト、システム・オブジェクト

---

## ■ デフォルト・オブジェクト

### ◆ MQオブジェクト作成時の雛形として使用される

- キュー・マネージャー作成時、それぞれのMQオブジェクトに対しデフォルト・オブジェクトが作成される
- MQオブジェクト作成時、指定のない属性はデフォルト・オブジェクトから引き継がれる

### ◆ デフォルト・オブジェクト名は SYSTEM から始まる

### ◆ デフォルト・オブジェクトの例

- ローカル・キュー：SYSTEM.DEFAULT.LOCAL.QUEUE
- SENDERチャンネル：SYSTEM.DEF.SENDER

### ◆ オブジェクト名を変更することはできない

### ◆ 属性値を変更することで、MQオブジェクト作成時のデフォルト値を変更可能

## ■ システム・オブジェクト

### ◆ キュー・マネージャーが内部的に使用するMQオブジェクト

### ◆ 主なシステム・オブジェクト

- SYSTEM.CHANNEL.SYNCQ：チャンネル間の同期に使用するキュー
- SYSTEM.ADMIN.QMGR.EVENT：キュー・マネージャーに関連するイベントが格納されるキュー

## <参考> キュー・マネージャ属性

属性	デフォルト値	説明
QMNAME	なし	キュー・マネージャ名
ACCTCONO	DISABLED(DISABLED/ENABLED)	アプリケーションでACCTMQI、ACCTQの設定を許可するか否か
ACCTINT	1800(1~604800)	会計情報を生成する時間間隔 (秒)
ACCTMQI	OFF(OFF/ON)	MQIの会計情報を生成するか否か
ACCTQ	OFF(OFF/ON/NONE)	キューの会計情報を生成するか否か
ACTIVREC	MSG(MSG/DISABLED/QUEUE)	アクティブ・レポートを生成するか否か
ACTVCONO	DISABLED(DISABLED/ENABLED)	アプリケーションでACTVTRCの設定を許可するか否か
ACTVTRC	OFF(OFF/ON)	MQIアプリケーションのアクティブ・トレース情報を生成するか否か
ADVCAP *	-	キュー・マネージャでIBM MQ Advanced拡張機能を使用できるか否か
ALTDATE *	-	最後に定義が変更された日付 (yyyy-mm-dd)
ALTTIME *	-	最後に定義が変更された時刻 (hh.mm.ss)
AMQPCAP *	-	キュー・マネージャでAMQP機能を使用できるか否か
AUTHOREV	DISABLED(ENABLED/DISABLED)	権限失敗のイベントを生成するか否か
CCSID	キュー・マネージャを作成した環境に従い設定される (1~65535)	キュー・マネージャのコード化文字セットID
CERTLABL	ibmwebspheremq<キュー・マネージャ名>	キュー・マネージャの証明書ラベル
CERTVPOL	ANY(ANY/RFC5280)	TLS証明書の認証ポリシー

\* DISPLAYコマンド実行時のみ表示される属性 (明示定義不可)

## <参考> キュー・マネージャ属性

属性	デフォルト値	説明
CHAD	DISABLED(DISABLED/ENABLED)	受信側チャンネルまたはサーバ接続チャンネルを自動定義するか否か
CHADEV	DISABLED(DISABLED/ENABLED)	チャンネル自動定義イベントを生成するか否か
CHADEXIT	なし(128文字以内)	チャンネル自動定義のEXIT名
CHLAUTH	ENABLED(ENABLED/DISABLED)	チャンネル認証レコードで定義された規則を使用するか否か
CHLEV	DISABLED(DISABLED/ENABLED/EXCEPTION)	チャンネル・イベントを生成するか否か
CLWLDATA	なし(32文字以内)	クラスター・ワークロードEXITデータ
CLWLEXIT	なし(128文字以内)	クラスター・ワークロードEXIT名
CLWLLEN	100(0~999999999)	クラスター・ワークロードEXITに渡されるメッセージデータの最大長
CLWLMRUC	999999999(1~999999999)	クラスター送信チャンネルの最大数
CLWLUSEQ	LOCAL(LOCAL/ANY)	クラスターで振り分け対象のキューがローカルとリモートに存在する場合、リモートのキューも振り分け対象にするか否か
CMDEV	DISABLED(DISABLED/ENABLED/NODISPLAY)	コマンドイベントを生成するか否か
CMDLEVEL *	-	キュー・マネージャによってサポートされるシステム制御コマンドのレベル
COMMANDQ *	-	システム・コマンド・キュー名
CONFIGEV	DISABLED(DISABLED/ENABLED)	構成イベントを生成するか否か
CONNAUTH	なし	認証情報オブジェクトの名前

\* DISPLAYコマンド実行時のみ表示される属性（明示定義不可）

## <参考> キュー・マネージャ属性

属性	デフォルト値	説明
CRDATE *	-	キュー・マネージャが作成された日付 (yyyy-mm-dd)
CRTIME *	-	キュー・マネージャが作成された時刻 (hh.mm.ss)
CUSTOM	なし	新たな属性が追加される際に、新機能の構成用として使用
DEADQ	なし(48文字以内)	送達不能メッセージを書き出すデッド・レター・キュー名
DEFCLXQ	SCTQ(SCTQ/CHANNEL)	クラスター送信チャネルに対してデフォルトで選択されるクラスター・トランスミッション・キューを制御
DEFXMITQ	なし(48文字以内)	省略時のトランスミッション・キュー名
DESCR	なし(64文字以内)	MQオブジェクトの説明
DISTL *	-	配布リストがパートナーQMgrでサポートされるか否か
IMGINTVL	60(1~999999999/OFF)	キュー・マネージャがメディア・イメージを自動で書き込む時間間隔 (分)
IMGLOGLN	OFF(OFF/1~999999999)	書き込まれるリカバリー・ログのサイズ (メガバイト)
IMGRCOVO	YES(YES/NO)	リニア・ログを使用する場合に、オブジェクトをメディア・イメージからリカバリー可能にする否か
IMGRCOVQ	YES(YES/NO)	リニア・ログを使用する場合に、動的キュー・オブジェクトをメディア・イメージからリカバリー可能にするか否か
IMGSCHEM	MANUAL(MANUAL/AUTO)	キュー・マネージャが自動でメディア・イメージを書き込むか否か
INHIBTEV	DISABLED(DISABLED/ENABLED)	GET/PUT禁止イベントを生成するか否か
INITKEY	なし	パスワード保護システム用の初期キー

\* DISPLAYコマンド実行時のみ表示される属性 (明示定義不可)

## <参考> キュー・マネージャ属性

属性	デフォルト値	説明
IPADDRV	IPV4(IPV4/IPV6)	チャンネル接続で使用するIPプロトコルのバージョン
KEYRPWD	なし	TLSキー・リポジトリにアクセスするためのパスワード
LOCALEV	DISABLED(DISABLED/ENABLED)	ローカル・エラーのイベントを生成するか否か
LOGGEREV	DISABLED(DISABLED/ENABLED)	ログ・イベントを生成するか否か
MARKINT	5000(0-999999999)	メッセージに付与されたマークが除去されるまでの時間 (ミリ秒)
MAXHANDS	256(0~999999999)	一つのタスクで同時にオープン可能なオブジェクト・ハンドルの最大数
MAXMSGL	4MB(32KB~100MB)	キュー・マネージャが許可するメッセージの最大長
MAXPROPL	NOLIMIT(0~100MB)	メッセージに関連付けられるプロパティの最大長を (バイト)
MAXPRTY *	-	最高優先順位(9)
MAXUMSGS	10000(1~999999999)	コミットされていないメッセージが1同期点内に存在可能な最大数
MONACLS	QMGR(QMGR/OFF/LOW/MEDIUM/HIGH)	自動定義クラスター送信チャンネルのオンライン・モニタリングを使用可能にするか否か
MONCHL	OFF(OFF/NONE/LOW/MEDIUM/HIGH)	チャンネルのオンライン・モニタリングを使用可能にするか否か
MONQ	OFF(OFF/NONE/LOW/MEDIUM/HIGH)	キューのオンライン・モニタリングを使用可能にするか否か
PARENT	なし	階層型Pub/Subで自身の親となるキュー・マネージャ名
PERFMEV	DISABLED(ENABLED/DISABLED)	パフォーマンス関連のイベントを生成するか否か QDPHIEV、QDPLOEV、QDPMAXEV利用のためENABLED指定

\* DISPLAYコマンド実行時のみ表示される属性 (明示定義不可)

## <参考> キュー・マネージャ属性

属性	デフォルト値	説明
PLATFORM *	-	キュー・マネージャが稼動しているプラットフォーム
PSCLUS	ENABLED(ENABLED/DISABLED)	クラスターにおいて、Pub/Sub機能を使用するか否か
PSMODE	ENABLED(ENABLED/COMPAT/DISABLED)	Pub/Subブローカーの稼動設定
PSNPMSG	DISCARD(DISCARD/KEEP)	Pub/Subブローカーがノンパースিসテントのパブリケーション・メッセージをサブスクライバークューやデッドレターキューに出力できなかった場合の処理
PSNPRES	NORMAL(NORMAL/SAFE/DISCARD/KEEP)	Pub/Subブローカーがノンパースিসテント・メッセージの応答メッセージを返せなかった場合の処理
PSRTYCNT	5(0-999999999)	Pub/Subブローカーが同期点付きコマンドメッセージの処理に失敗した場合にレポート・オプションに従って処理する前にコマンドをリトライする回数
PSSYNCPT	IFPER(YES/IFPER)	Pub/Subブローカーがコマンドメッセージを同期点付きで処理するか否か
QMID *	-	キュー・マネージャの固有ID
REMOTEEV	DISABLED(DISABLED/ENABLED)	リモート・エラーのイベントを生成するか否か
REPOS	なし(48文字以内)	リポジトリ・キュー・マネージャとして参加するクラスター名
REPOSNL	なし(48文字以内)	キュー・マネージャの提供するリポジトリ・マネージャサービスの対象となるクラスター名前リスト
REVDNS	ENABLED(ENABLED/DISABLED)	TRPTYPE(TCP)を指定した場合に、チャンネルの接続元であるIPアドレスに対してDNSからホスト名を逆引きするか否か
ROUTEREC	MSG(MSG/DISABLED/QUEUE)	トレース・ルート情報を生成するか否か
SCHINIT	QMGR(QMGR/MANUAL)	キュー・マネージャの起動と同時にチャンネル・イニシエーターを起動するか否か

\* DISPLAYコマンド実行時のみ表示される属性（明示定義不可）

## <参考> キュー・マネージャ属性

属性	デフォルト値	説明
SCMDSERV	QMGR(QMGR/MANUAL)	キュー・マネージャの起動と同時にコマンド・サーバを起動するか否か
SPLCAP *	-	AMS機能が使用可能か否か
SSLCRLNL	なし	CRLチェックに使用されるAUTHINFOオブジェクトの名前リスト名
SSLCRYP	なし(256文字以内)	暗号化装置の構成に必要なパラメータ・ストリング名
SSLEV	DISABLED(DISABLED/ENABLED)	SSLイベントを生成するか否か
SSLFIPS	NO(NO/YES)	FIPSで認定された暗号方式のみ使用可能にするか否か
SSLKEYR	/var/mqm/qmgrs/QMGR/ssl/key	SSLのキー・リポジトリ名
SSLRKEYC	0(0~999999999)	秘密鍵を再生成するまでに送受信するバイト数
STATACLS	QMGR(QMGR/OFF/LOW/MEDIUM/HIGH)	自動定義クラスター送信チャネルの統計情報を生成するか否か
STATCHL	OFF(OFF/NONE/LOW/MEDIUM/HIGH)	チャネルの統計情報を生成するか否か
STATINT	1800(1~604800)	統計情報を生成する時間間隔 (秒) MAXASログリーダー統計に合わせる
STATMQI	OFF(OFF/ON)	MQIの統計情報を生成するか否か
STATQ	OFF(OFF/ON/NONE)	キューの統計情報を生成するか否か
STRSTPEV	ENABLED(ENABLED/DISABLED)	キュー・マネージャ開始/終了のイベントを生成するか否か
SUITEB	NONE(NONE/128_BIT/192_BIT/128_BIT,192_BIT)	Suite B準拠の暗号方式を使用するか否か、またその強度を指定
SYNCPT *	-	同期点サポートが使用可能か否か (読み取り専用の属性)

\* DISPLAYコマンド実行時のみ表示される属性 (明示定義不可)

## <参考> キュー・マネージャ属性

属性	デフォルト値	説明
TREELIFE	1800(0-604000)	非管理トピックの持続時間（秒）
TRIGINT	999999999(0~999999999)	TRIGTYPE(FIRST)、かつキューが空でない時に次のメッセージが入ってきた場合、このインターバルを経過していればトリガー・メッセージを生成（ミリ秒）
VERSION *	-	キュー・マネージャが関連付けられているインストレーションのバージョン (VRRMMFF) : V(Version)、R(Release)、M(Maintenance level)、F(Fix level)
XRCAP *	-	MQ Telemetry機能がキュー・マネージャによってサポートされているか否か

\* DISPLAYコマンド実行時のみ表示される属性（明示定義不可）

## <参考> キュー属性

属性	ローカル・キュー	リモート・キュー定義	別名キュー	モデル・キュー
QUEUE	○	○	○	○
ACCTQ	○			○
ALTDATA *	○	○	○	○
ALTTIME *	○	○	○	○
BOQNAME	○			○
BOTHRESH	○			○
CAPEXPY	○	○	○	○
CLCHNAME	○			
CLUSNL	○	○	○	
CLUSTER	○	○	○	
CLWLPRTY	○	○	○	
CLWLRANK	○	○	○	
CLWLUSEQ	○			
CRDATE *	○			○
CRTIME *	○			○
CURDEPTH *	○			
CUSTOM	○	○	○	○
DEFBIND	○	○	○	
DEFPRESP	○	○	○	○

\* DISPLAYコマンド実行時のみ表示される属性（明示定義不可）

## <参考> キュー属性

属性	ローカル・キュー	リモート・キュー定義	別名キュー	モデル・キュー
DEFPRTY	○	○	○	○
DEFPSIST	○	○	○	○
DEFREADA	○		○	○
DEFSOPT	○			○
DEFTYPE				○
DESCR	○	○	○	○
DISTL	○			○
GET	○		○	○
HARDENBO/NOHARDENBO	○			○
IMGRCOVQ	○			○
INITQ	○			○
IPPROCS *	○			
MAXDEPTH	○			○
MAXFSIZE *	○			○
MAXMSGL	○			○
MONQ	○			○
MSGDLVSQ	○			○
NPMCLASS	○			○
OPPROCS *	○			

\* DISPLAYコマンド実行時のみ表示される属性（明示定義不可）

## <参考> キュー属性

属性	ローカル・キュー	リモート・キュー定義	別名キュー	モデル・キュー
PROCESS	○			○
PROPCTL	○		○	○
PUT	○	○	○	○
QDEPTHHI	○			○
QDEPTHLO	○			○
QDPHIEV	○			○
QDPLOEV	○			○
QDPMAXEV	○			○
QSVCIEV	○			○
QSVCINT	○			○
QTYPE *	○	○	○	○
RETINTVL	○			○
RNAME		○		
RQMNAME		○		
SCOPE	○	○	○	
SHARE/NOSHARE	○			○
STATQ	○			○
STREAMQ	○			○
STRMQOS	○			○

\* DISPLAYコマンド実行時のみ表示される属性（明示定義不可）

## <参考> キュー属性

属性	ローカル・キュー	リモート・キュー定義	別名キュー	モデル・キュー
TARGET			○	
TARGQ			○	
TARGETTYPE			○	
TRIGDATA	○			○
TRIGDPTH	○			○
TRIGGER/NOTRIGGER	○			○
TRIGMPRI	○			○
TRIGTYPE	○			○
USAGE	○			○
XMITQ		○		

\* DISPLAYコマンド実行時のみ表示される属性（明示定義不可）

## <参考> キュー属性

属性	デフォルト値	説明
QUEUE	なし(48文字以内)	キュー名
ACCTQ	QMGR(QMGR/ON/OFF)	キューの会計情報を生成するか否か
ALTDATE *	-	最後に定義が変更された日付 (yyyy-mm-dd)
ALTTIME *	-	最後に定義が変更された時刻 (hh.mm.ss)
BOQNAME	なし	バックアウト・リキュー名
BOTHRESH	0(0~999999999)	バックアウトの閾値
CAPEXPY	NOLIMIT(NOLIMIT/1~999999999)	オブジェクト・ハンドルを使用して書き込まれたメッセージの有効期限 (デシ秒)
CLCHNAME	なし(48文字以内)	転送キューとして使用するクラスター送信チャネル名
CLUSNL	なし	キューが所属するクラスター名前リスト名
CLUSTER	なし(48文字以内)	キューが所属するクラスター名
CLWLPRTY	0(0~9)	クラスターのプライオリティ (9が優先順位高)
CLWLRANK	0(0~9)	クラスターのランク (9が優先順位高)
CLWLUSEQ	QMGR(QMGR/ANY/LOCAL)	クラスターで振り分け対象のキューがローカルとリモートに存在する場合、リモートのキューも振り分け対象にするか否か
CRDATE *	-	キューが定義された日付 (yyyy-mm-dd)
CRTIME *	-	キューが定義された時刻 (hh.mm.ss)
CURDEPTH *	-	キューの現行数

\* DISPLAYコマンド実行時のみ表示される属性 (明示定義不可)

## <参考> キュー属性

属性	デフォルト値	説明
CUSTOM	なし(128文字以内)	新たな属性が追加される際に、新機能の構成用として使用
DEFBIND	OPEN(OPEN/NOTFIXED/GROUP)	MQOPEN時のデフォルトのBINDオプション
DEFPRESP	SYNC(SYNC/ASYN)	メッセージ送信の応答タイプのデフォルト
DEFPRTY	0(0~9)	キューのメッセージに与える省略時の優先順位 (9が優先順位高)
DEFPSIST	NO(NO/YES)	再起動をまたいでメッセージを保持するか否か
DEFREADA	NO(NO/YES/DISABLED)	ノン・パーシステント・メッセージの先読み機能の使用可否
DEFSOPT	SHARED(SHARED/EXCL)	キューのOPEN時の省略時の共用オプション
DEFTYPE	TEMPDYN(TEMPDYN/PERMDYN)	動的キューのタイプ
DESCR	なし(64文字以内)	MQオブジェクトの説明
DISTL	NO(NO/YES)	配布リストがパートナーQMgrでサポートされるか否か
GET	ENABLED(ENABLED/DISABLED)	メッセージのGETを認めるか否か
HARDENBO/ NOHARDENBO	HARDENBO(HARDENBO/NOHARDENBO)	メッセージの回復回数(Backout Count)の記録をハード化するか否か (分散系MQでは無視：指定値に関わらずHARDENBOになる)
IMGRCOVQ	QMGR(QMGR/NO/YES)	リニア・ログを使用する場合に、動的キュー・オブジェクトを メディア・イメージからリカバリー可能にするか否か
INITQ	なし	トリガー・メッセージを送るキューの名称
IPPROCS *	-	メッセージをGETするキューに接続されているアプリケーションの数
MAXDEPTH	5000(0~999999999)	キューに保持できるメッセージの最大数

\* DISPLAYコマンド実行時のみ表示される属性 (明示定義不可)

## <参考> キュー属性

属性	デフォルト値	説明
MAXFSIZE	DEFAULT(2MBに相当)	拡張可能なキュー・ファイルの最大サイズ (メガバイト)
MAXMSGL	4MB(32KB~100MB)	キューに保持できるメッセージの最大長
MONQ	QMGR(QMGR/OFF/LOW/MEDIUM/HIGH)	キューのオンライン・モニタリングを使用可能にするか否か
MSGDLVSQ	PRIORITY(PRIORITY/FIFO)	メッセージの配布順序
NPMCLASS	NORMAL(NORMAL/HIGH)	ノン・パーシステント・メッセージの信頼性のレベル
OPPROCS *	-	メッセージをPUTするキューに接続されているアプリケーションの数
PROCESS	なし	トリガー・モニターを起動するプロセス名
PROPCTL	COMPAT(COMPAT/ALL/FORCE/NONE)	メッセージ・プロパティの制御方法
PUT	ENABLED(ENABLED/DISABLED)	メッセージのPUTを認めるか否か
QDEPTHHI	80(0~100)	QueueDepthHighイベントが発生するメッセージの%
QDEPTHLO	20(0~100)	QueueDepthLowイベントが発生するメッセージの%
QDPHIEV	DISABLED(DISABLED/ENABLED)	QueueDepthHighイベントの発生可否：イベント利用のためEnabled
QDPLOEV	DISABLED(DISABLED/ENABLED)	QueueDepthLowイベントを発生可否：イベント利用のためEnabled
QDPMAXEV	ENABLED(ENABLED/DISABLED)	QueueDepthMaxイベントを発生可否
QSVCI EV	NONE(NONE/HIGH/OK)	QueueServiceIntervalイベントを発生させるか否か
QSVCI NT	999999999(0~999999999)	QueueServiceIntervalイベントが発生する間隔 (ミリ秒)
QTYPE *	-	キューのタイプ

\* DISPLAYコマンド実行時のみ表示される属性 (明示定義不可)

## <参考> キュー属性

属性	デフォルト値	説明
RETINTVL	999999999(0~999999999)	キューが作成されてから不要になるまでの時間（時）
RNAME	なし	接続先キュー・マネージャー上でのキュー名
RQMNAME	なし	接続先キュー・マネージャー名
SCOPE	QMGR(QMGR/CELL)	キューの定義が有効な範囲
SHARE/NOSHARE	SHARE(SHARE/NOSHARE)	キューのGET時の共用オプション
STATQ	QMGR(QMGR/OFF/ON)	キューの統計情報を生成するか否か
STREAMQ	なし	キューに書き込まれたメッセージを、指定した別のキューにコピー
STRMQOS	BESTEF(BESTEF/MUSTDUP)	メッセージをコピーする際のサービス品質
TARGET	なし(48文字以内)	別名キューが指すキュー/トピックの名称
TARGQ	なし	別名キューが指すキューの名称（TARGETとの同時指定不可）
TARGETYPE	QUEUE(QUEUE/TOPIC)	別名が解決するオブジェクトのタイプ
TRIGDATA	なし(0~64)	トリガー・メッセージに挿入するデータ
TRIGDPTH	1(1~999999999)	トリガー・メッセージを生成するメッセージの数
TRIGGER/NOTRIGGER	NOTRIGGER(NOTRIGGER/TRIGGER)	トリガー・メッセージを作成するか否か
TRIGMPRI	0(0~9)	トリガー・メッセージを生成する対象メッセージの優先順位選択
TRIGTYPE	FIRST(FIRST/EVERY/DEPTH/NONE)	トリガー・メッセージを生成する条件
USAGE	NORMAL(NORMAL/XMITQ)	キューの用途
XMITQ	なし	使用する転送キュー名

## <参考> チャネル属性

属性	SDR	SVR	RCVR	RQSTR	CLNTCONN	SVRCONN	CLUSSDR	CLUSRCVR
channel-name	○	○	○	○	○	○	○	○
AFFINITY					○			
ALTDATE *	○	○	○	○	○	○	○	○
ALTTIME *	○	○	○	○	○	○	○	○
AUTOSTART *		○	○	○		○		
BATCHHB	○	○					○	○
BATCHINT	○	○					○	○
BATCHLIM	○	○					○	○
BATCHSZ	○	○	○	○			○	○
CERTLABL	○	○	○	○	○	○	○	○
CHLTYPE	○	○	○	○	○	○	○	○
CLNTWGHT					○			
CLUSNL							○	○
CLUSTER							○	○
CLWLPRTY							○	○
CLWLRANK							○	○
CLWLWGHT							○	○
COMPHDR	○	○	○	○	○	○	○	○
COMPMSG	○	○	○	○	○	○	○	○

\* DISPLAYコマンド実行時のみ表示される属性（明示定義不可）

## <参考> チャネル属性

属性	SDR	SVR	RCVR	RQSTR	CLNTCONN	SVRCONN	CLUSDR	CLUSRCVR
CONNAME	○	○		○	○		○	○
CONVERT	○	○					○	○
DEFCDISP	○	○	○	○		○		
DESCR	○	○	○	○	○	○	○	○
DISCINT	○	○				○	○	○
HBINT	○	○	○	○	○	○	○	○
KAINT	○	○	○	○	○	○	○	○
LOCLADDR	○	○		○	○		○	○
LONGRTY	○	○					○	○
LONGTMR	○	○					○	○
MAXINST						○		
MAXINSTC						○		
MAXMSGL	○	○	○	○	○	○	○	○
MCANAME	○	○		○			○	○
MCTYPE	○	○		○			○	○
MCAUSER	○	○	○	○		○	○	○
MODENAME	○	○		○	○		○	○
MONCHL	○	○	○	○		○	○	○
MRDATA			○	○				○
MREXIT			○	○				○

## <参考> チャネル属性

属性	SDR	SVR	RCVR	RQSTR	CLNTCONN	SVRCONN	CLUSDR	CLUSRCVR
MRRTY			○	○				○
MRTMR			○	○				○
MSGDATA	○	○	○	○			○	○
MSGEXIT	○	○	○	○			○	○
NETPRTY								○
NPMSPEED	○	○	○	○			○	○
PASSWORD	○	○		○	○		○	○
PROPCTL								
PUTAUT	○	○					○	○
QMNAME			○	○		○		○
RESETSEQ *	○	○	○	○			○	○
RCVDATA	○	○	○	○			○	○
RCVEXIT	○	○	○	○	○	○	○	○
SCYDATA	○	○	○	○	○	○	○	○
SCYEXIT	○	○	○	○	○	○	○	○
SENDDATA	○	○	○	○	○	○	○	○
SENDEXIT	○	○	○	○	○	○	○	○
SEQWRAP	○	○	○	○	○	○	○	○
SHARECNV	○	○	○	○			○	○

\* DISPLAYコマンド実行時のみ表示される属性（明示定義不可）

## <参考> チャネル属性

属性	SDR	SVR	RCVR	RQSTR	CLNTCONN	SVRCONN	CLUSDR	CLUSRCVR
SHORTRTY					○	○		
SHORTTMR	○	○					○	○
SSLCAUTH	○	○					○	○
SSLCIPH	○	○	○	○	○	○	○	○
SSLPEER	○	○	○	○	○	○	○	○
STATCHL	○	○	○	○			○	○
TPNAME	○	○		○	○	○	○	○
TRPTYPE	○	○	○	○	○	○	○	○
USEDLQ	○	○	○	○			○	○
USERID	○	○		○	○		○	
XMITQ	○	○						

## <参考> チャネル属性

属性	デフォルト値	説明
channel-name	なし	チャネル名
AFFINITY	PREFERRED(PREFERRED/NONE)	同一マシンからの接続試行順序を保証するか否か
ALTDATE *	-	最後に定義が変更された日付 (yyyy-mm-dd)
ALTTIME *	-	最後に定義が変更された時刻 (hh.mm.ss)
AUTOSTART *	-	LU6.2応答側プロセスを開始するか否か
BATCHHB	0(0~999999)	バッチ処理前に事前検査の目的で行うハートビートの最小間隔 (ミリ秒)
BATCHINT	0(0~999999999)	バッチ処理前の時間間隔 (ミリ秒)
BATCHLIM	5000(0~999999)	チャネル転送の際に、1つの同期点に含める最大データ量 (キロバイト)
BATCHSZ	50(1~9999)	チャネル転送の際に、1つの同期点に含める最大メッセージ数
CERTLABL	なし	チャネルの証明書ラベル
CHLTYPE	なし	チャネルのタイプ
CLNTWGHT	0(1~99)	クラスター振り分け重み付け
CLUSNL	なし	クラスター名前リスト
CLUSTER	なし(0~48文字)	クラスター名
CLWLPRTY	0(0~9)	クラスタープライオリティ (0:低、9:高)
CLWLRANK	0(0~9)	クラスターチャネルのランク (0:低、9:高)
CLWLWGHT	50(1~99)	クラスター振り分け重み付け

\* DISPLAYコマンド実行時のみ表示される属性 (明示定義不可)

## <参考> チャネル属性

属性	デフォルト値	説明
COMPHDR	NONE(NONE/SYSTEM)	ヘッダーの圧縮方法
COMPMSG	NONE(NONE/RLE/ZLIBFAST/ZLIBHIGH/ANY)	メッセージの圧縮方法
CONNNAME	なし(ポート番号は1414が初期値)	接続先アドレス(ポート番号)
CONVERT	NO(NO/YES)	チャネルにて文字コード変換実施可否
DEFCDISP	PRIVATE(PRIVATE/FIXSHARED/SHARED)	デフォルト・チャネルの処理方法
DEFRECON	NO(NO/YES/QMGR/DISABLED)	クライアント接続が切断した場合に、自動的に再接続するか否か
DESCR	なし(64文字以内)	MQオブジェクトの説明
DISCINT	6000(0~999999)	メッセージ転送が無い場合の自動切断時間 (秒)
HBINT	300(0~999999)	ハートビート・フローの時間間隔 (秒)
KAINT	AUTO(AUTO/0~99999)	チャネル単位のTCPキープアライブ時間 (分散系MQでは無効)
LOCLADDR	なし	送信元インターフェースのアドレス・ポート
LONGRTY	999999999(0~999999999)	SHORTRTYの再接続回数試行後、LONGTMRインターバルで実施する再接続回数
LONGTMR	1200(0~999999999)	LONGRTY実施時の実施間隔 (秒)
MAXINST	999999999(0-999999999)	同時に接続可能なサーバ接続チャネル・インスタンス数
MAXINSTC	999999999(0-999999999)	1クライアントあたり同時に接続可能なサーバ接続チャネル・インスタンス数
MAXMSGL	4194304(0~104857600)	送受信できる最大のメッセージ長
MCANAME	なし	リザーブ値
MCTYPE	PROCESS(PROCESS/THREAD)	メッセージチャネルエージェントの実行タイプ

## <参考> チャネル属性

属性	デフォルト値	説明
MCAUSER	なし(12文字以内)	MCAのアクセス許可検査に使用するユーザー (WINでは64文字以内)
MODENAME	なし(8文字以内)	LU6.2モードネーム
MONCHL	QMGR(QMGR/OFF/LOW/MEDIUM/HIGH)	チャネルのオンライン・モニタリングを使用可能にするか否か
MRDATA	なし(32文字以内)	チャネルメッセージリトライ出口ユーザデータ
MREXIT	なし(128文字以内)	チャネルメッセージリトライ出口名
MRRTY	10(0~999999999)	チャネルメッセージリトライ回数
MRTMR	1000(0~999999999)	チャネルメッセージリトライ間隔(ミリ秒)
MSGDATA	なし(999文字以内)	チャネル・メッセージ出口に渡すデータ
MSGEXIT	なし(128文字以内)	チャネル・メッセージ出口名
NETPRTY	0(0~9)	チャネル転送パスの選択プライオリティ (0: 低、9: 高)
NPMSPEED	FAST(FAST/NORMAL)	非持続性メッセージの高速/標準送達
PASSWORD	なし	LU6.2セキュアセッションのパスワード値
PROPCTL	COMPAT(COMPAT/ALL/NONE)	メッセージ・プロパティの制御方法
PUTAUT	DEF(DEF/CTX)	宛先キューにメッセージを書き込む権限の設定
QMNAME	なし	MQIクライアント環境で実行するクライアント・アプリケーションからの接続要求の対象となるキュー・マネージャーの名前
RESETSEQ *	-	リセット未処理のシーケンス番号

\* DISPLAYコマンド実行時のみ表示される属性 (明示定義不可)

## <参考> チャネル属性

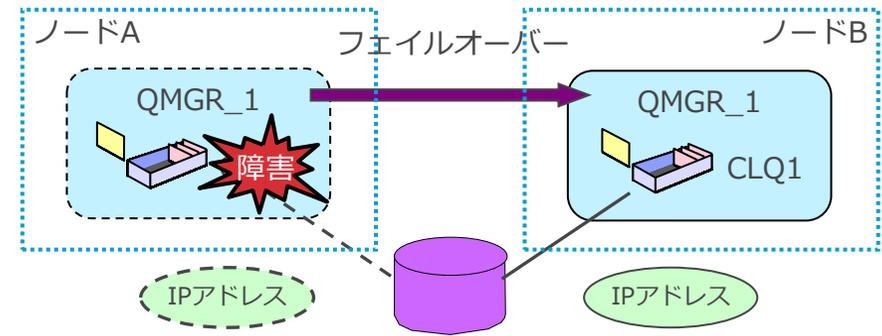
属性	デフォルト値	説明
RCVDATA	なし(32文字以内)	チャネル受信出口に渡すデータ
RCVEXIT	なし(999文字以内)	チャネル受信出口名
SCYDATA	なし(32文字以内)	チャネル・セキュリティー出口に渡すデータ
SCYEXIT	なし(128文字以内)	チャネル・セキュリティー出口名
SENDATA	なし(32文字以内)	チャネル送信出口に渡すデータ
SENDEXIT	なし(128文字以内)	チャネル送信出口名
SEQWRAP	999999999(100~999999999)	チャネルが管理する順序番号の最大値
SHARECNV	10(0-999999999)	1つのソケットが共用するチャネル・インスタンスの最大数
SHORTRTY	10(0~999999999)	チャネル再接続試行回数
SHORTTMR	60(0~999999999)	チャネル再接続試行間隔 (秒)
SSLCAUTH	REQUIRED(REQUIRED/OPTIONAL)	SSLクライアントの認証を行うか否か
SSLCIPH	なし(32文字以内)	SSLで使用する暗号方式
SSLPEER	なし(1024文字以内)	SSLで使用する証明書の識別名に対するフィルター
STATCHL	QMGR(QMGR/OFF/LOW/MEDIUM/HIGH)	チャネルの統計情報を生成するか否か
TPNAME	なし	LU6.2トランザクション・プログラム名 (最大長は 64 文字)
TRPTYPE	TCP(LU62/NETBIOS/SPX/TCP)	使用する転送プロトコル
USEDLQ	YES(YES/NO)	チャネルでメッセージを送信できない場合に、送達不能キューを使用するか否か
USERID	なし(12文字以内)	LU6.2セッション・ユーザーID
XMITQ	なし	転送キューの名前



## 可用性構成

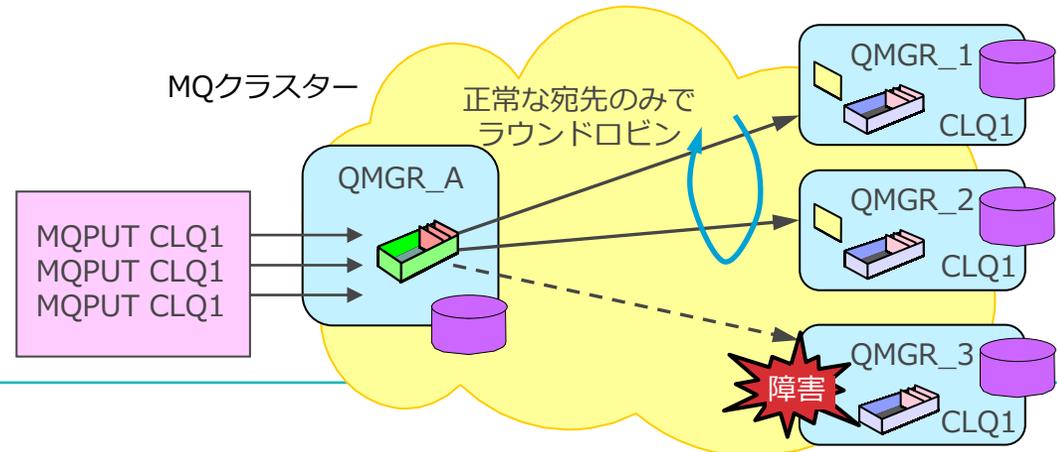
## ■ACTIVE-STANDBY構成(フェイルオーバー構成)

- ◆正常時は片系で稼働し、障害発生時は待機ノードでキュー・マネージャーを起動する構成
- ◆キュー・マネージャーの実体(キュー・ファイル、トランザクション・ログ)を共有
  - 共有ディスク/共有ファイルシステムに配置
  - データ・レプリケーションによる同期
- ◆主な構成方法
  - クラスター・ソフトウェアによる構成
  - マルチ・インスタンス・キュー・マネージャー
  - RDQM
  - Native HA (Cloud環境のみ利用可能)



## ■ACTIVE-ACTIVE構成

- ◆複数ノードでキュー・マネージャーを起動し、サービスを提供 (処理の負荷分散が可能)
- ◆障害発生時には自動で縮退運転に切替
- ◆主な構成方法
  - キュー・マネージャー接続
    - MQクラスター
  - クライアント接続
    - Uniform Cluster



## マルチ・インスタンス・キュー・マネージャー

---

### ■ マルチ・インスタンス・キュー・マネージャーとは

- ◆ HAコーディネーター製品(HACMPなど)を使用せずに、待機系のキュー・マネージャーをアイドル状態で待機させるフェイルオーバー機能
- ◆ キュー・マネージャー・データやログを共有ファイルシステム(NFS v4など)に配置することで可用性が向上

### ■ 自動フェイルオーバーが可能

- ◆ アクティブ・インスタンスで障害が発生した場合、自動でスタンバイ・インスタンスへフェイルオーバー

### ■ 2つ以上のキュー・マネージャー・インスタンスを構成することが可能

- ◆ 3つ目以降のキュー・マネージャー・インスタンスはスタンバイとして開始しておくことは不可

### ■ z/OS以外の全プラットフォームで対応

- ◆ strmqm、endmqm、dspmq のオプション使用でマルチ・インスタンス・キュー・マネージャーを制御
  - strmqm -x : マルチ・インスタンスとしてキュー・マネージャーを開始
  - endmqm -s : アクティブ・インスタンスを停止し、スタンバイ・インスタンスへフェイルオーバー
    - -s オプションを使用しない場合、アクティブ・インスタンスを停止するとスタンバイ・インスタンスも停止
  - dspmq -x : キュー・マネージャー・インスタンスに関する情報を表示

- ネットワークストレージにキュー・マネージャーのキューファイルやトランザクション・ログを配置
  - ◆ AIX、Linux : NFS v4以上
  - ◆ Windows : Windows共有、CIFS(Common Internet File System)など
  - ◆ ファイル・システムの要件 (IBM Documentation)
    - <https://www.ibm.com/docs/en/ibm-mq/9.3?topic=multiplatforms-requirements-shared-file-systems>
- 同一IDのmqmユーザー/グループを使用
  - ◆ AIX、Linuxでは同一UID、GIDのmqmユーザー/グループ
  - ◆ Windowsではmqmドメイン・ローカル・グループ
    - Windowsでドメイン・ローカル・グループを使用する理由
      - ドメインおよびフォレストのユーザー/グループを、同一メンバーとして所有することができる
      - ドメイン内のアクセス許可のみに使用することができる

## マルチ・インスタンス・キュー・マネージャーの考慮点および制約

---

- 2つ以上のキュー・マネージャー・インスタンスを構成することが可能
  - ◆ 可用性を考慮しても、3つ目以降はスタンバイとして開始しておくことができないため手動での起動が必要
- IPアドレスは引き継がないため、接続元のチャンネルのCONNNAMEには両ホストの情報登録が必要
- 最初に起動したキュー・マネージャー・インスタンスがアクティブとして稼動
  - ◆ 運用にて起動順序を決めておくことが望ましい
- マルチ・インスタンス・キュー・マネージャーの情報は、共用キュー・マネージャー・データ・ディレクトリにある qm.ini および qmstatus.ini ファイルに保管
  - ◆ Windowsもレジストリではない
- 同一マルチ・インスタンス・キュー・マネージャーは同一バージョンであることが必須

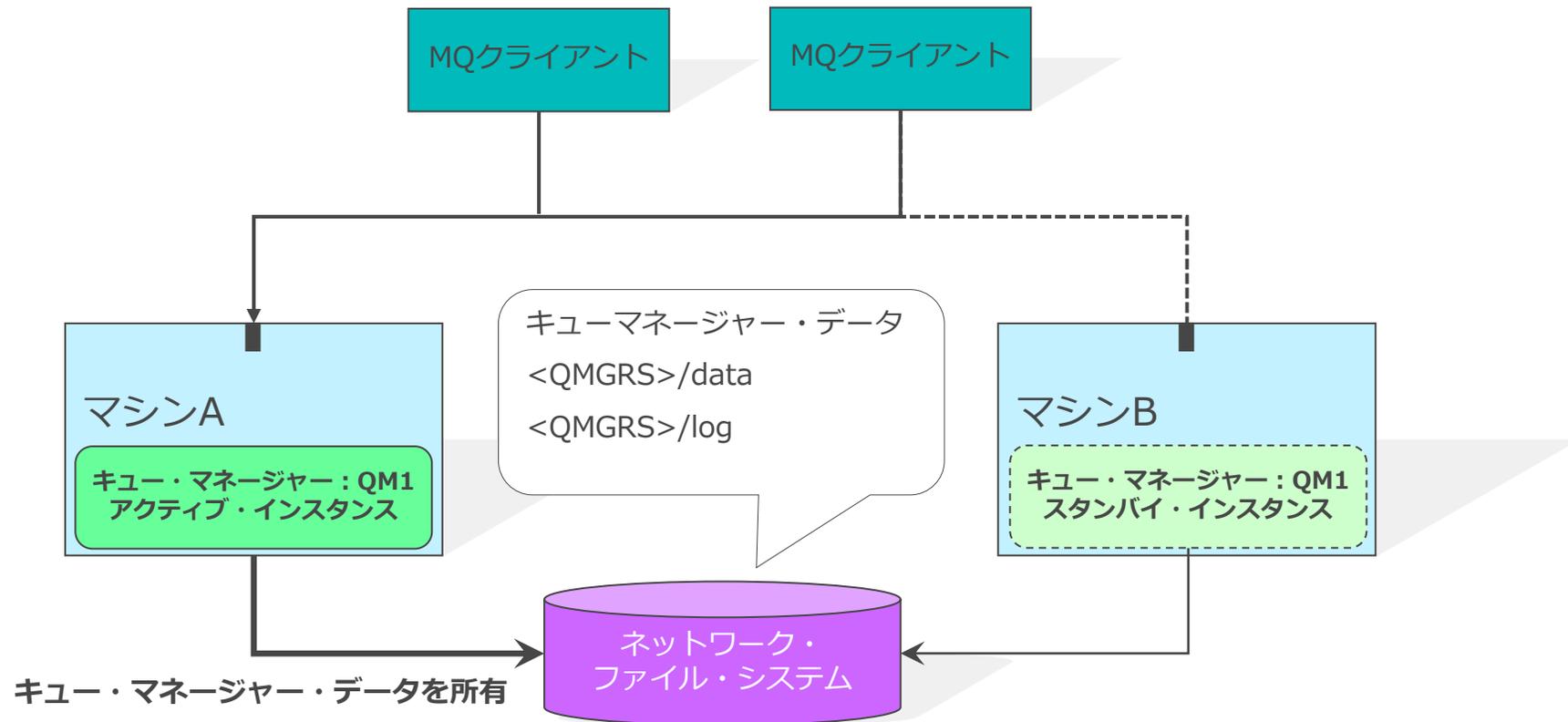
## マルチ・インスタンス・キュー・マネージャーの考慮点および制約

---

- Windowsにおいて、ログおよびデータ・ディレクトリはFATおよびNFS v3ファイル・システムには格納できない
- Windows環境でマルチ・インスタンス・キュー・マネージャーを稼働させる場合、ドメイン・コントローラ上に作成する必要がある
  - ◆ 通常、業務サーバ（MQやWASなどが稼働）と管理サーバ（ユーザ情報を管理するドメイン・コントローラなど）を同一ネットワークセグメントに配置することはないため、セキュリティへの対応も考慮する必要がある
- Windows環境でのマルチ・インスタンス・キュー・マネージャーの起動オプションは「手動」に設定
  - ◆ 起動オプションを「自動」で起動すると、-x オプションなし(STANDBYを許可しない)でキュー・マネージャーが開始されてしまうため

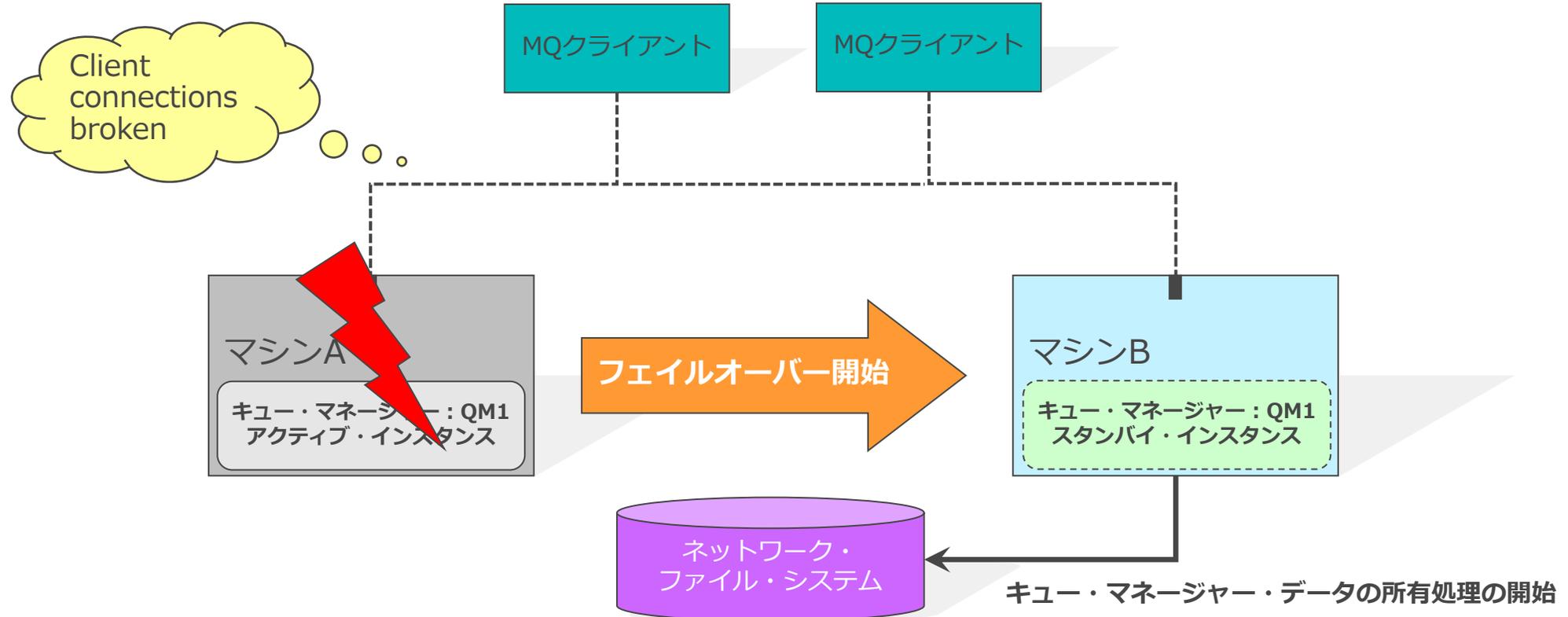
## 動作イメージ (通常稼動)

- マルチ・インスタンス・キュー・マネージャー通常稼動時
  - ◆ マシンAのキュー・マネージャー：アクティブ・インスタンスとして稼動
  - ◆ マシンBのキュー・マネージャー：スタンバイ・インスタンスとして稼動
- クライアントはマシンAへのCONNNAMEで接続



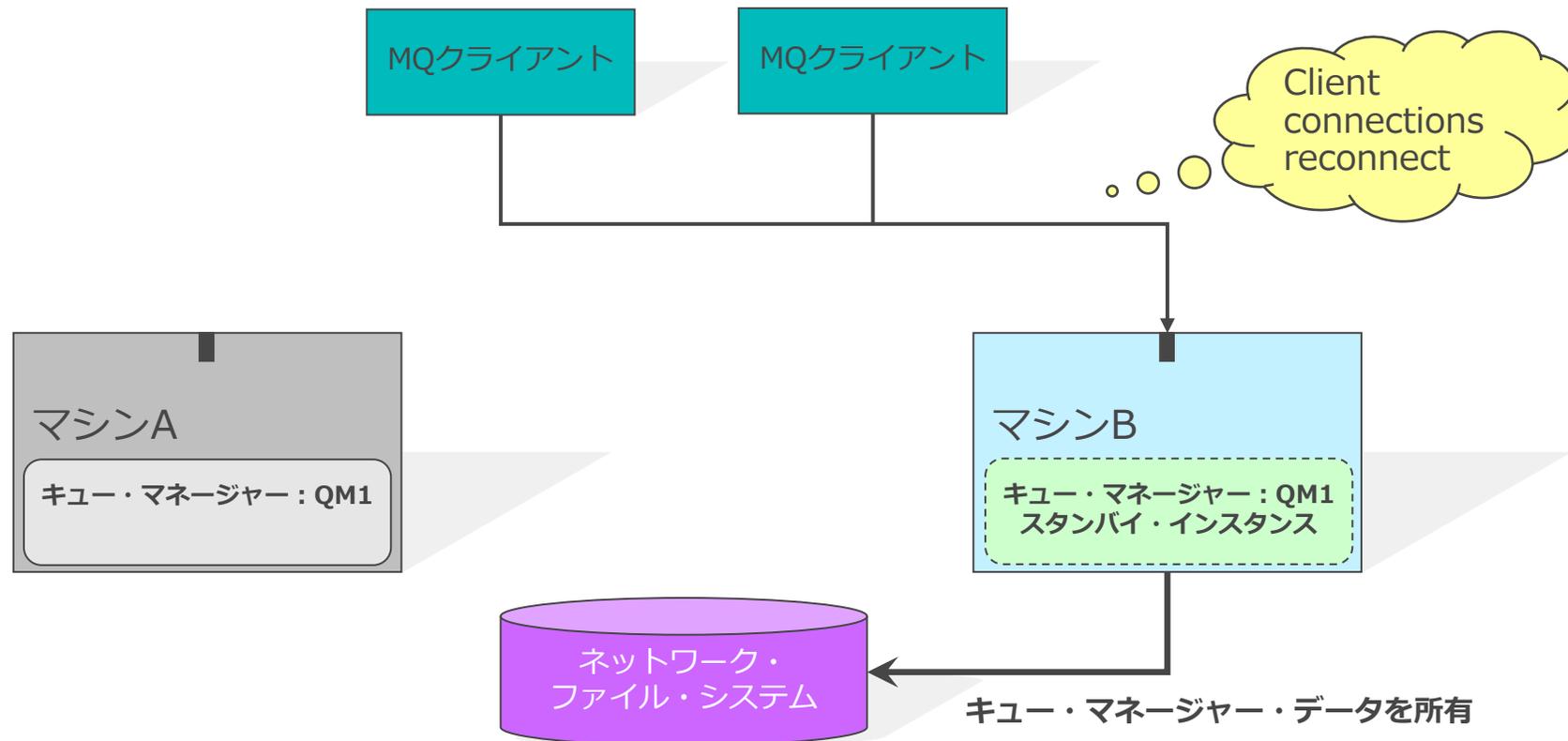
## 動作イメージ（フェイルオーバー）

- アクティブ・インスタンスがダウンし、スタンバイ・インスタンスへフェイルオーバー開始
  - ◆ マシンBのキュー・マネージャー・インスタンスがキュー・マネージャー・データの所有処理を開始する
- クライアントからの接続は切断
  - ◆ クライアント自動再接続を利用することで、アプリケーションにConnection Brokenを返さないようにすることも可能



## 動作イメージ (フェイルオーバー)

- スタンバイ・インスタンスがアクティブ・インスタンスとして稼動開始
  - ◆ キュー・マネージャー・データの所有
- クライアントはマシンBへのCONNNAMEで再接続

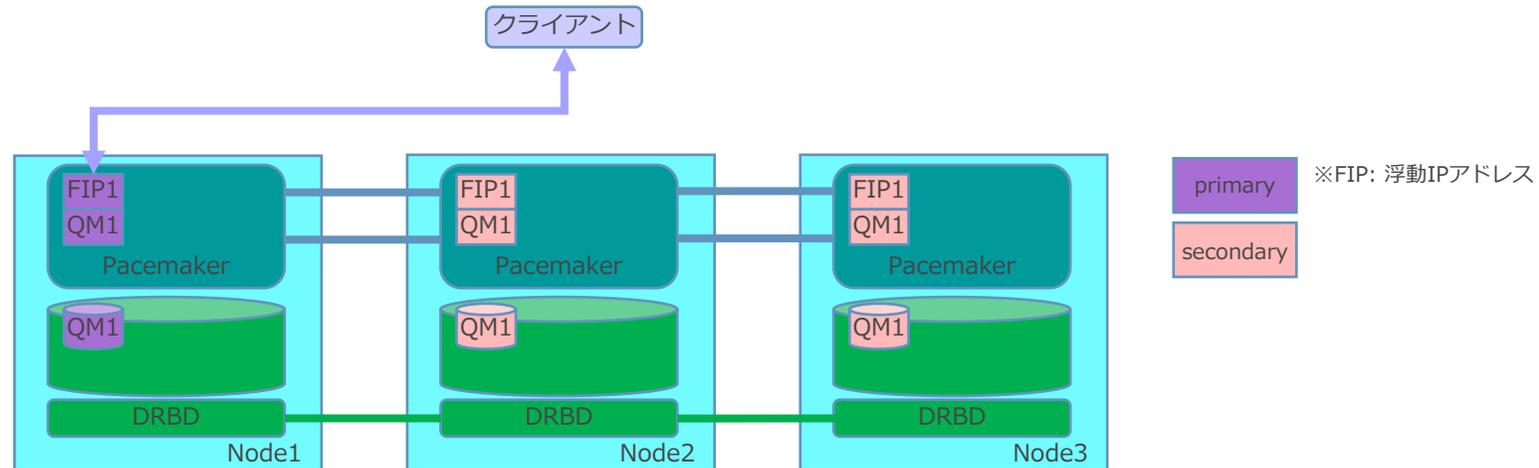


- RDQM(Replicated Data Queue Manager : 複製データ・キュー・マネージャー)とは
  - ◆ Linux RHELプラットフォームでフェイルオーバー構成を実現する機能
    - RHEL上で使用可能な、Pacemaker・DRBD(\*1)を利用
  - ◆ 外部ディスクなしで、オンプレミス環境およびクラウド環境で高可用性を実現可能
  - ◆ RDQMはMQ Advancedライセンスで使用可能
- 以下3種類の構成が可能
  - ◆ HA RDQM : V9.0.4~
  - ◆ DR RDQM : V9.0.5~
  - ◆ DR/HA RDQM : V9.1.5~

\*1)

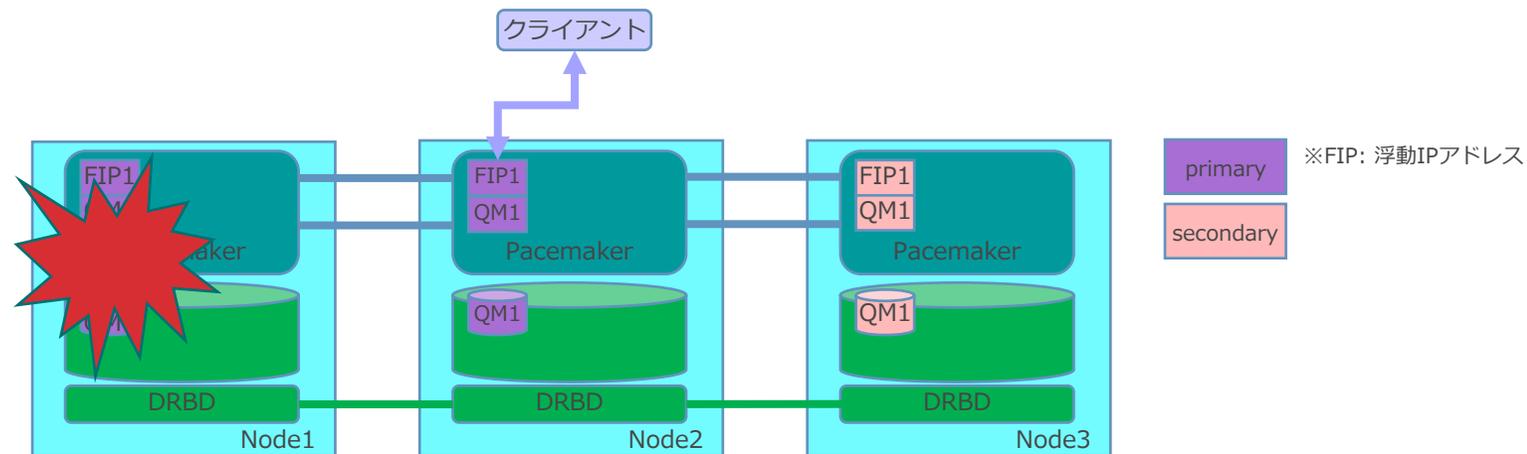
- ◆ Pacemaker
  - オープンソースソフトウェアのHAクラスタソフト(リソースマネージャクラスタ)
  - RDQMにおいては、ノードのモニタリングおよび浮動IPアドレスの管理を行う
  - 参考 : <https://clusterlabs.org/pacemaker/>
- ◆ DRBD (Distributed Replicated Block Device)
  - Linuxで使用可能な、ストレージのレプリケーションのためのソフトウェアで、シェアードナッシングを実現
  - DRBDはサーバ間でブロックデバイス(ハードディスク、パーティション、論理ボリュームなど)の内容をミラー
  - RDQMにおいては、ノード間のデータの複製を行う
  - 参考 : <https://docs.linbit.com/docs/users-guide-9.0/>

- HA RDQM(高可用性RDQM)は、MQ AdvancedのV9.0.4以降で使用可能な高可用性ソリューション
- HA RDQMは、高可用性(HA)グループとしてクォーラム構成された3つのサーバで構成
  - ◆ 個々のサーバにPacemaker、キュー・マネージャー・インスタンス、DRBDが必要
    - Pacemakerの役割：3つのノードのモニタリング及び浮動IPアドレスの管理を行う
    - DRBDの役割：3つのノード間のデータ複製を行う
  - ◆ 共用の外部ディスクは不要で、それぞれのノードのローカルディスクにデータを保持
  - ◆ インスタンスの1つはプライマリー-QMGRとして実行
    - QMGR稼働中の更新データを、他の2つのセカンダリー・インスタンスに同期的に複製



# HA RDQMのフェイルオーバー

- QMGRを実行しているノードの障害をPacemakerが検知すると、別ノードのQMGRインスタンスが自動的に開始
  - ◆ クォーラムを設定した3ノード・システムが前提となるため、1ノードの障害時には処理継続可能
  - ◆ 各QMGRが浮動IPアドレスを使用することにより、QMGRの稼動ノードに依存せずクライアントは同じIPアドレスを利用してアクセスを継続
- 他ノードと接続していない状態ではQMGRの稼動不可
  - ◆ クォーラム構成のため、二重障害などで他2ノードとの接続が失われた状態ではQMGRは使用不可



# HA RDQM構成の有効活用例

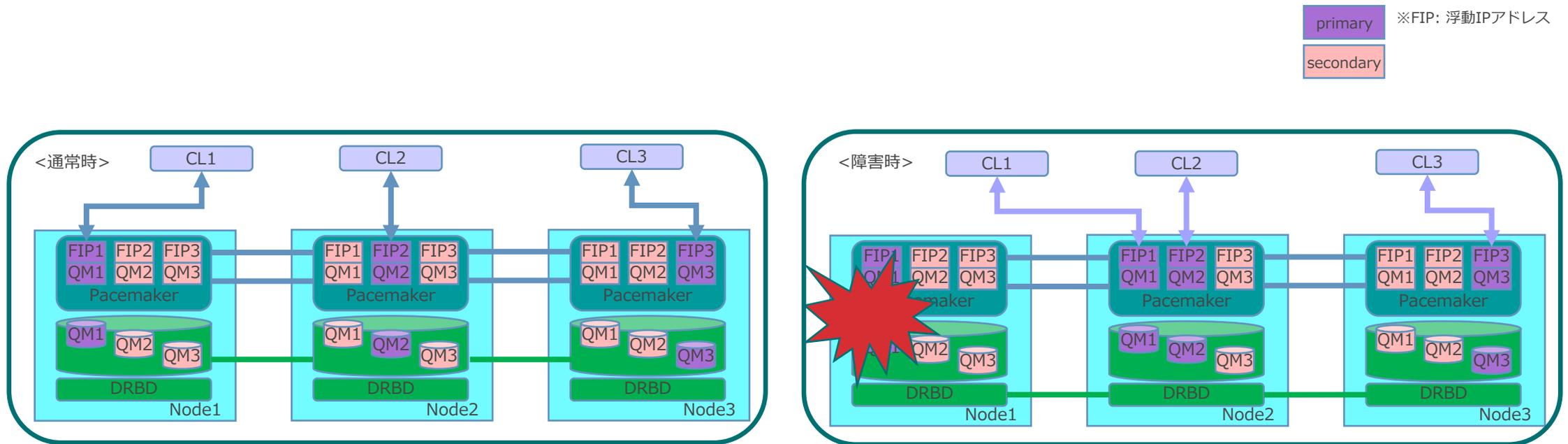
## ■ 3ノードの有効活用

◆ 3ノードにそれぞれ異なるQMGRを構成

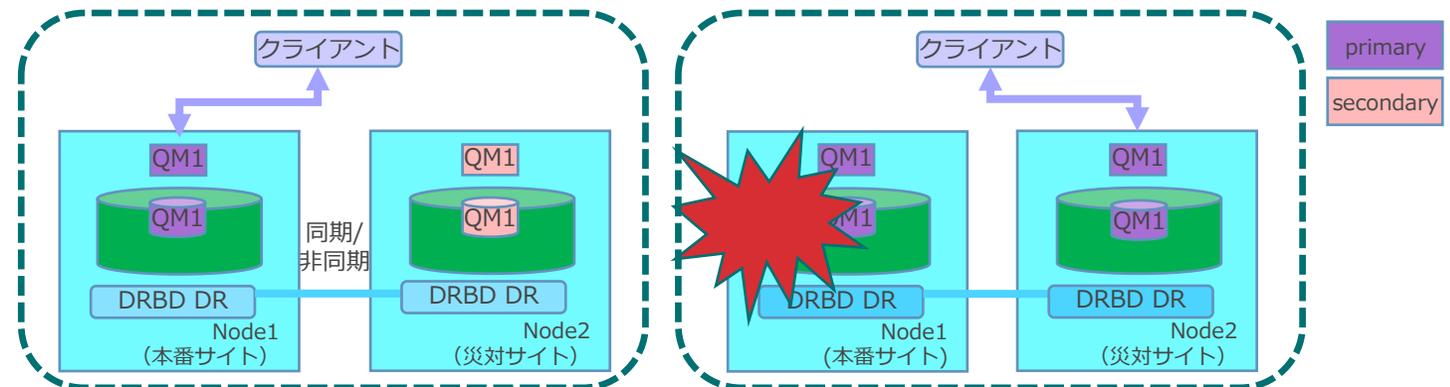
- 例) QM1、QM2、QM3

## ■ 別ノードに各QMGRのセカンダリー・インスタンスを構成

◆ 相互テイクオーバーが可能な構成とする



- DR RDQM(災対RDQM)は、MQ AdvancedのV9.0.5以降で使用可能な災害対策構築ソリューション
- DR RDQMは、本番サイトと災対サイトで稼動する1ペアのサーバで構成
  - ◆ それぞれのサーバにはQMGRインスタンス、DRBDを構成
  - ◆ DRBDがノード間のデータ複製を行う
    - データの複製は、同期/非同期を選択可能
      - 非同期複製では、プライマリー・ノードでのMQのPUTやGETなどの制御をアプリケーションに戻した後、そのイベントに関するデータがセカンダリー・ノードに複製される
      - 直前のデータ(メッセージ)がロストする可能性があるので注意
- 本番サイトでの障害発生時には、災対サイトのQMGRをプライマリーQMGRとして手動で起動
  - ◆ 災対サイトのQMGRのデータは整合性のある状態を保持しているため、QMGRの起動は可能
  - ◆ 本番サイトのQMGRが停止していることを確認してから、災対サイトのインスタンスをプライマリーQMGRとして起動させる
    - スプリット・ブレイン発生を防止するため



### ■ 複製(レプリケーション)・同期・スナップショット

#### ◆ 複製

- 2ノード間のネットワークが接続されているとき、プライマリー・ノードへの更新がセカンダリー・ノードに転送されること

#### ◆ 同期

- 2ノード間のネットワークが切断されるとプライマリー・ノードでの更新がトラッキングされ、ネットワーク再開後、複製とは別のプロセスによってプライマリー・ノードでの更新情報がセカンダリー・ノードに送信されること

#### ◆ スナップショット

- 同期処理の開始時にセカンダリー・ノードの状態を取得すること
- 同期処理中はセカンダリー・ノードのデータが不整合な状態になるため、フェイルオーバーに備えてQMGRが稼働できる状態を確保する

### ■ スプリット・ブレーンに関する考慮

#### ◆ DR RDQM環境では、セカンダリー・ノードへの切り替えを手動 (RDQM外の制御) で実施するため、スプリット・ブレーン(=同時に2ノードが稼働/接続)を発生させないように制御すること

- 例1) 障害によりプライマリー・ノードが完全に停止して、セカンダリーがプライマリーに昇格し処理を開始した場合
  - 元のプライマリーが回復後、まずセカンダリーとして起動してデータを複製・同期すること
- 例2) 2ノード間のレプリケーション用ネットワークやプライマリー・ノード側のネットワークが切断された場合
  - プライマリー側を停止させてから、セカンダリーをプライマリーに昇格させること
  - ネットワークの回復で両系に接続することを回避するため



- DR/HA RDQMが実行されているノードで障害が発生すると、自動的にその高可用性グループ内の別のノードにフェイルオーバー
  - ◆ HA RDQMと同様
- サイト全体で障害が発生すると、災対サイトのプライマリー・ノードでRDQM QMGRを手動で開始
  - ◆ DR RDQMと同様
- 本番サイト/災対サイトという関係ではなく、それぞれにアクティブQMGRを構成することが可能
  - ◆ 例) サイトAでQMGR1/QMGR2がアクティブ、サイトBでQMGR3がアクティブ
  - ◆ 災害・障害により1つのサイトが使用不可となった場合には、すべてが同じサイトで実行される

### ■ マイグレーション・サポート

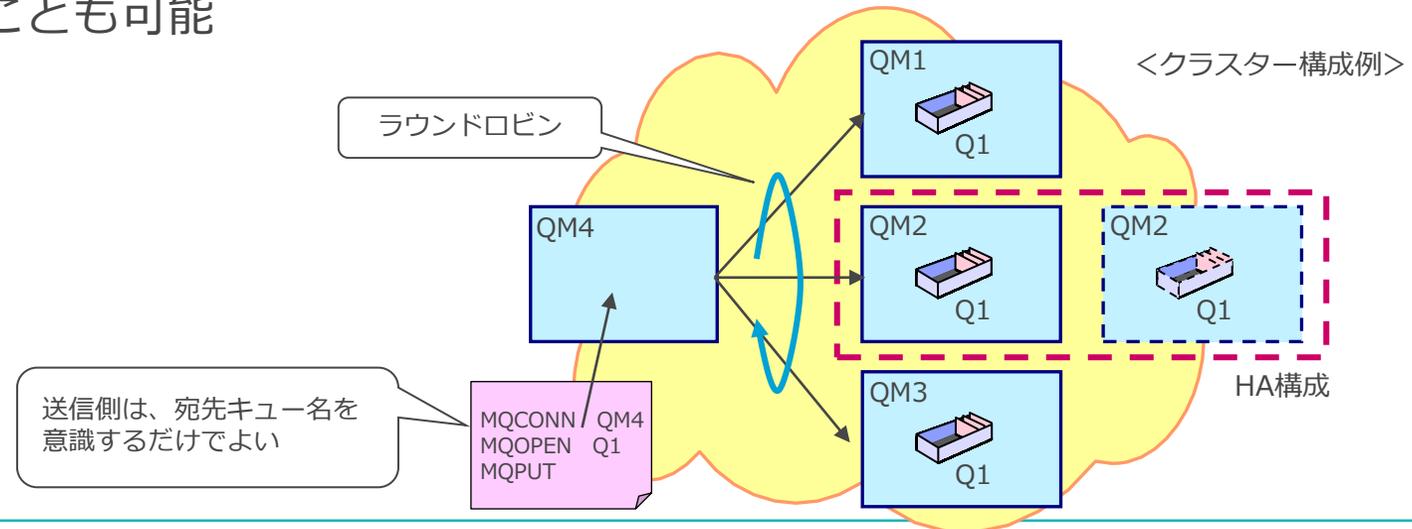
- ◆ バージョンアップに際してサポートされるマイグレーションパス（V9.0からV9.3など）においては、ローリング・アップグレードをサポート
  - ただし、下位のバージョンから上位のバージョンへのフェイルオーバーのみ可能
    - セカンダリー・ノードからアップグレードを行う
    - 可能な限り低いレベルのノードでQMGRが稼動するよう、優先ロケーションおよび第2優先ロケーションを設定する
- ◆ RHEL のメジャーバージョンアップなどにより Pacemaker のバージョンが変更される場合には、HAグループの再作成が必要となり、ローリングアップグレードが行えない可能性がある
  - RHEL 7~RHEL 9では、いずれのRHELのバージョンアップでもHAグループの再作成が必要
  - RHEL 7/8のHAグループをバックアップし、RHEL 9で作成した新HAグループにリストアする方法で移行可能

## ■ MQクラスターとは

- ◆ 論理的に関連付けられたキュー・マネージャーの集合

## ■ 特徴

- ◆ MQクラスターでは、キュー定義をMQクラスター内のキュー・マネージャーに公開することが可能
- ◆ 公開されたクラスター・キューへは、リモート定義なしにメッセージ送信が可能
  - 公開された情報は一元管理され、クラスター・キューへのメッセージ送信に必要なオブジェクトは自動定義される
- ◆ 送信側アプリケーションは、宛先キューがどのキュー・マネージャーにあるのか意識する必要がない
- ◆ MQクラスター内では、キュー・マネージャー間のメッセージ・チャンネルは自動作成される
- ◆ システムを停止させずにキュー・マネージャーやキューをクラスター環境に追加できる
- ◆ HA製品などと組み合わせることも可能
  - より高い可用性を実現



### ■ メッセージレベルの負荷分散

- ◆ ワークロード管理機能によって、同一名称のキュー間でのメッセージの振り分けが可能

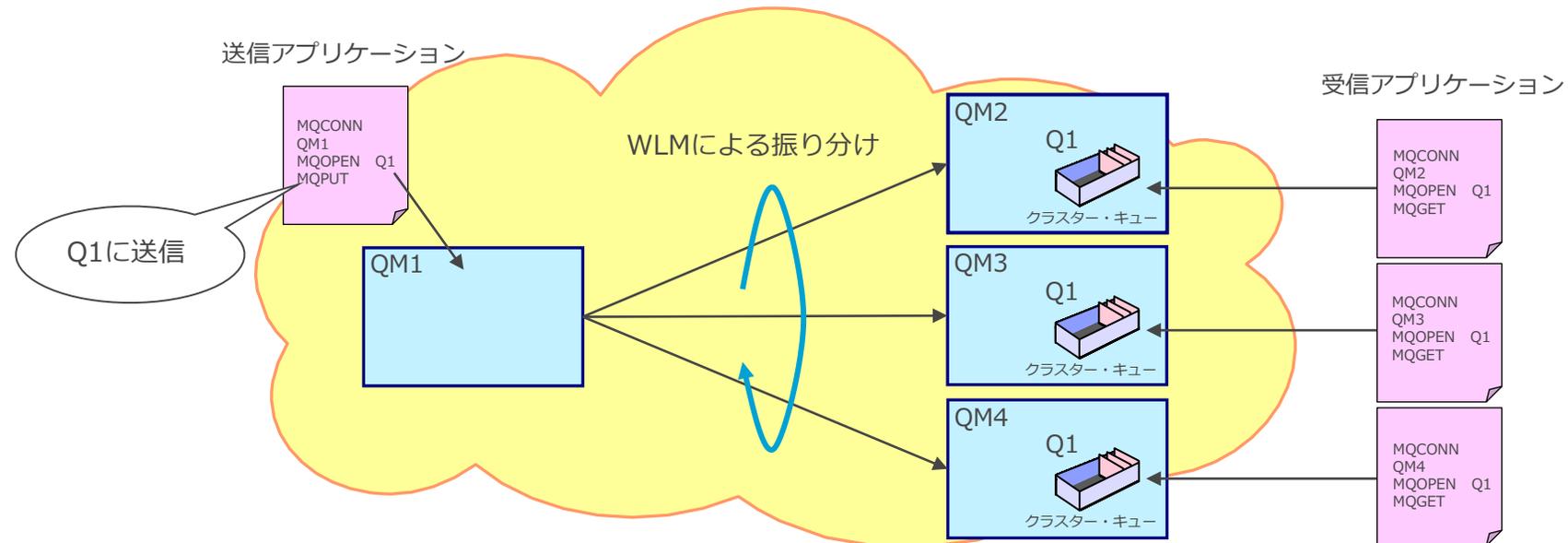
### ■ 高可用性の実現

- ◆ キュー・マネージャーが障害を受けると自動的にメッセージの宛先対象から除外され、残りの正常なキュー・マネージャー間で処理
  - キュー・マネージャーが復旧すると、再び自動的にメッセージの宛先対象となる
    - チャンネルの再接続は必要

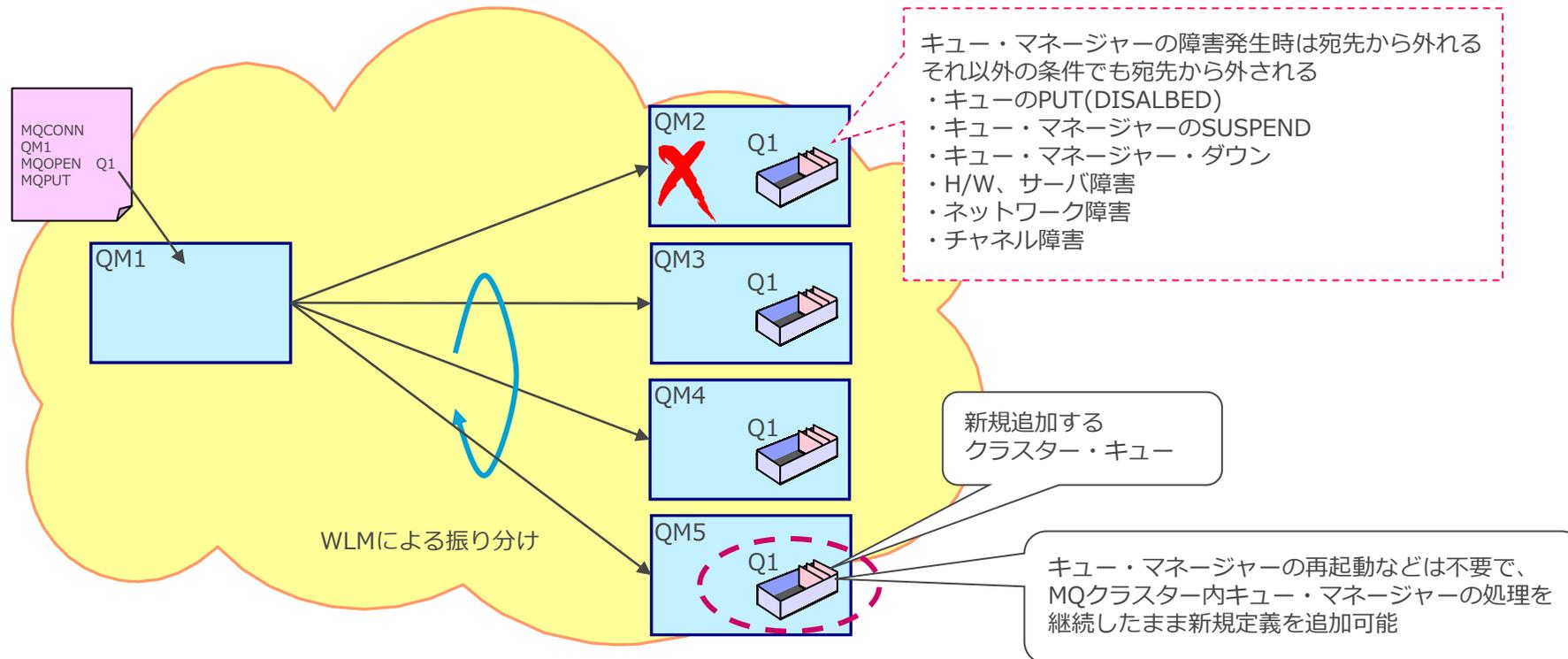
### ■ MQオブジェクト管理の簡素化

- ◆ メッセージ送信に必要なオブジェクトが自動定義される
- ◆ 構成の変更がより早く、簡単になり、定義ミスも削減
- ◆ クラスター情報が一元管理される

- WLM(ワークロード・マネージャー)によるメッセージの振り分けが可能
  - ◆ 複数キュー・マネージャー上の同一名称のクラスター・キューへ、ラウンドロビン方式で宛先を振り分け
    - メッセージの最終宛先キュー・マネージャーは、WLMにより決定
  - ◆ 振り分け方法は2つあり、送信アプリケーションのMQOPEN時のBINDオプションにて指定
    - MQPUT単位(メッセージ単位)で振分
    - MQOPEN単位で複数メッセージを同一キューに振り分け
  - ◆ 単純なラウンドロビンに加え、重み付けや優先順位を考慮した振り分けが可能 (後述)



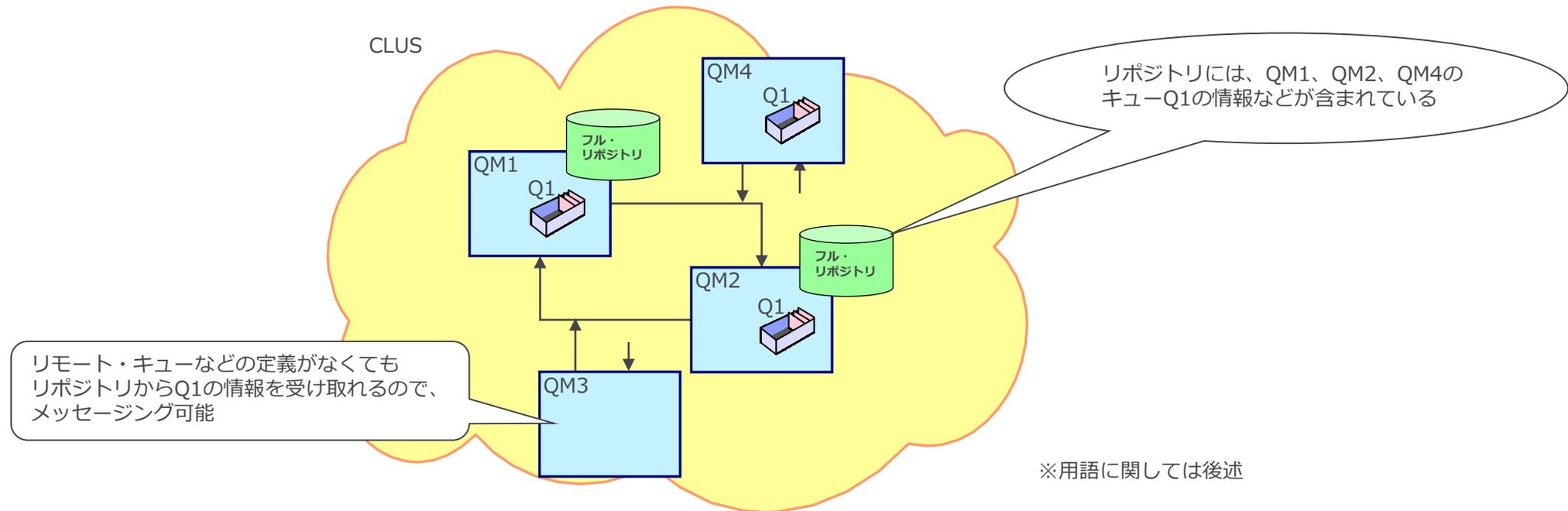
- 宛先キュー・マネージャーの障害発生時には、残りの正常稼動キュー・マネージャーでメッセージを振り分け
  - ◆ キュー・マネージャーの障害を検知後、宛先対象から外れる
  - ◆ 障害が復旧すれば、自動的にラウンドロビン対象に復帰
  - ◆ システム停止なしで、キュー・マネージャーやキューなどのリソースをクラスターに追加可能



# MQオブジェクト管理の簡素化

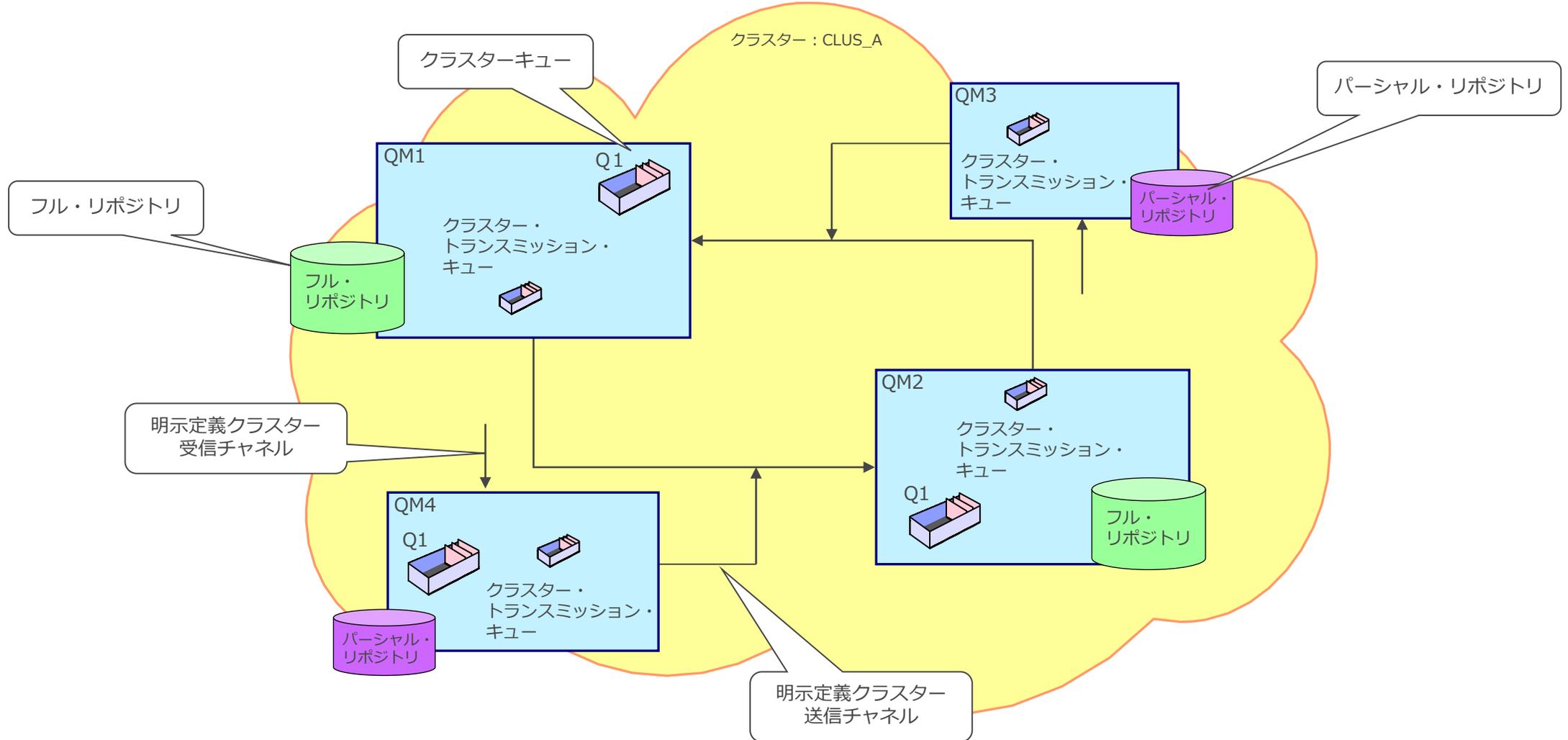
## ■ メッセージ送信に必要なオブジェクトを自動定義

- ◆ クラスタ内のオブジェクト定義情報はフル・リポジトリを保持するキュー・マネージャーが一元管理
- ◆ メッセージ送信に必要な情報は、リポジトリから取得
  - DQM(Distributed Queue Management : 分散キュー管理)構成で必要となるリモート・キュー、トランスマッション・キュー、宛先キュー・マネージャーに対するチャンネルなどの定義が不要
- ◆ 構成の定義/変更がより早く、簡単になり、定義ミスも削減



# MQクラスターの構成要素

## ■ MQクラスター構成例



# リポジトリ

## ■ リポジトリ

- ◆ クラスタに参加するキュー・マネージャーは必ず1つ保持
- ◆ MQクラスター内のキュー・マネージャー、キューに関する情報を保管
  - 実体はキュー、情報はパーシステント・メッセージ
    - SYSTEM.CLUSTER.REPOSITORY.QUEUE(システム・デフォルト・オブジェクト)

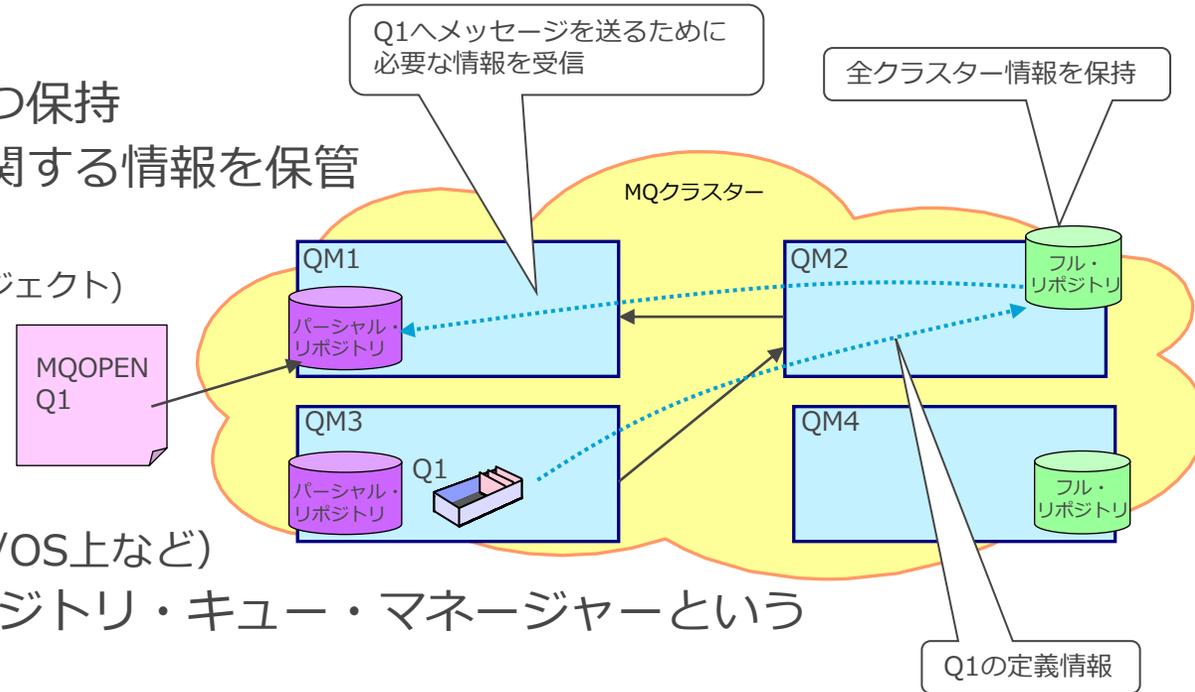
## ◆ フル・リポジトリとパーシャル・リポジトリがある

## ■ フル・リポジトリ

- ◆ MQクラスター内の全ての情報を保持するリポジトリ
  - 可用性の最も高いシステムにフル・リポジトリを配置 (z/OS上など)
- ◆ フル・リポジトリを持つキュー・マネージャーをリポジトリ・キュー・マネージャーという
  - 通常クラスター内に2つ構成する

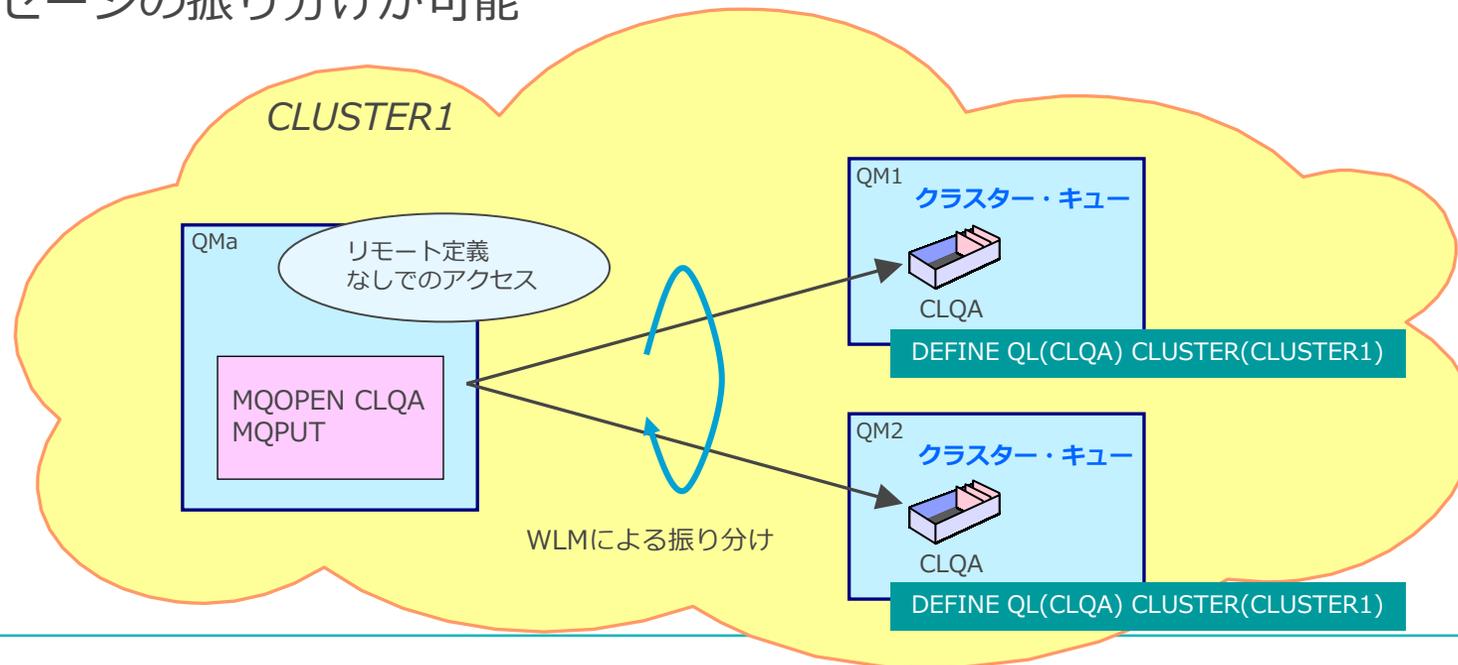
## ■ パーシャル・リポジトリ

- ◆ キュー・マネージャーが使用する情報のみを保持するリポジトリ
  - 必要な情報は、フル・リポジトリ経由で受信
- ◆ リポジトリ・キュー・マネージャー以外のキュー・マネージャーに存在



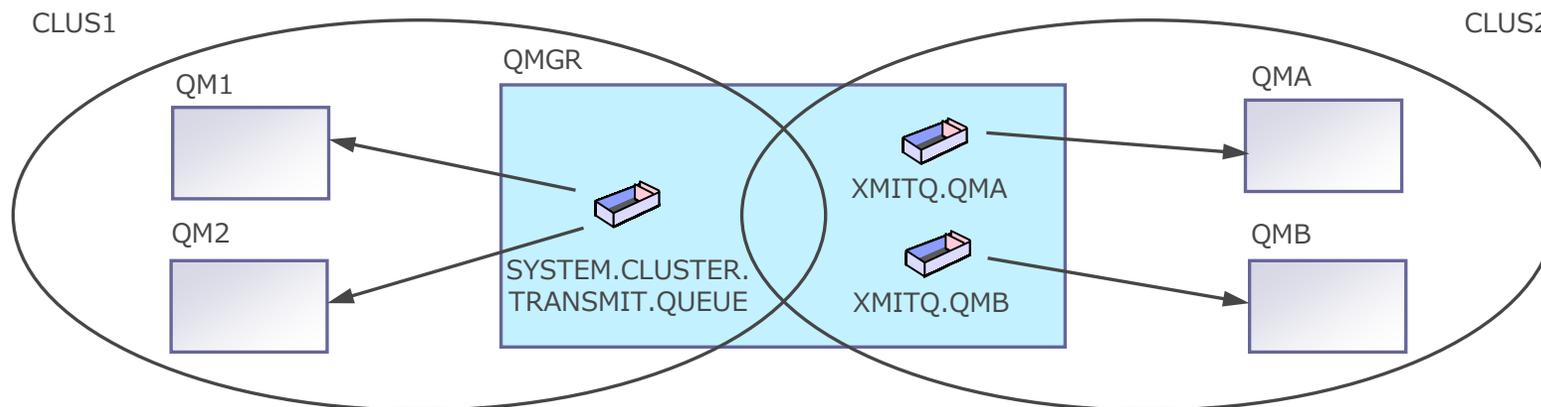
## クラスター・キュー

- クラスター内で、キュー名だけでアクセス可能なキュー
  - ◆ キュー属性CLUSTERにクラスター名を指定したキュー
  - ◆ メッセージ送信に必要となるリモート・キューなどのオブジェクト定義が不要
  - ◆ ローカル・キュー、別名キュー、リモート・キューをクラスター・キュー化可能
  - ◆ 送信アプリケーションは、キューが存在する宛先キュー・マネージャー名を知らなくてもよい
    - MQOPEN時にクラスター・キュー名を指定するだけ
- 複数キュー・マネージャー上に同一名のクラスター・キューを定義可能
  - ◆ WLMによるメッセージの振り分けが可能



## クラスター・トランスミッション・キュー

- クラスター内におけるキュー・マネージャー間のメッセージ送信の際に使用される転送キュー
- 1つのキュー・マネージャー上で複数のクラスター・トランスミッション・キュー(CTQ)を構成可能
  - ◆ デフォルトでは、SYSTEM.CLUSTER.TRANSMIT.QUEUE(SCTQ)を全てのクラスター送信チャンネルで共有
  - ◆ クラスター送信チャンネルごとに個別のCTQを割り当てることも可能
  - ◆ CTQは、自動指定する方法と明示指定する方法がある（後述）
- 複数のCTQを構成するメリット
  - ◆ クラスター送信チャンネルごとにメッセージのトラフィックをモニターできる
    - ゲートウェイ・キュー・マネージャーなど接続先キュー・マネージャーが多くある場合に特に有効
  - ◆ キュー障害の場合に、全てのキュー・マネージャーへの送信が停止することを回避できる

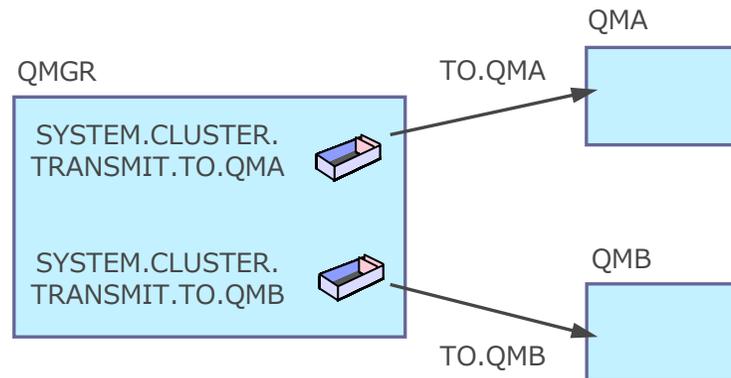


### ■ 個別クラスター・トランスミッション・キューを自動的に指定する方法

#### ◆ キュー・マネージャーのDEFCLXQ属性に「CHANNEL」を指定

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

- ※DEFCLXQのデフォルト値：SCTQ  
(従来通り、全てのクラスター送信チャンネルがSYSTEM.CLUSTER.TRANSMIT.QUEUE を使用)
- ◆ クラスター送信チャンネルごとに個別のトランスミッション・キューの割り当てが可能
  - 次回の送信チャンネル開始時から新たなトランスミッション・キューを使用する
  - CTQ名は、SYSTEM.CLUSTER.TRANSMIT.<クラスター送信チャンネル名> (固定)
    - モデルキュー SYSTEM.CLUSTER.TRANSMIT.MODEL の永続動的キューとして作成される
  - 個別のCTQが作成されるのは次回の送信チャンネル開始のタイミング



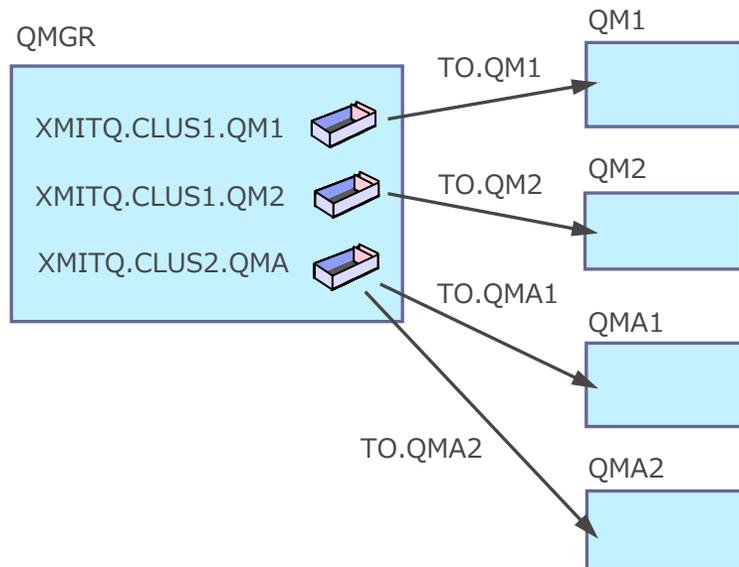
## 個別クラスター・トランスミッション・キューの明示指定

### ■ 個別トランスミッション・キューを明示定義し、利用するクラスター送信チャンネルを指定する方法

#### ◆ トランスミッション・キューのCLCHNAME属性にこのキューを使うクラスター送信チャンネルを指定

```
DEFINE QLOCAL(<任意のキュー名>) USAGE(XMITQ) CLCHNAME(<チャンネル名>)
```

- ※CLCHNAME：トランスミッション・キューを使う送信チャンネル名を指定
- ◆ CLCHNAMEは、個別のチャンネル名や「\*」による複数チャンネル名の指定が可能
  - 「\*」は任意の文字列で、「\*」指定ではチャンネルのネーミングの考慮が必要
- ◆ CLCHNAMEで指定されないクラスター送信チャンネルはSCTQを使用
- ◆ 回目の送信チャンネル開始時から新たなトランスミッション・キューを使用する



<トランスミッション・キューに対するクラスター送信チャンネルの指定>

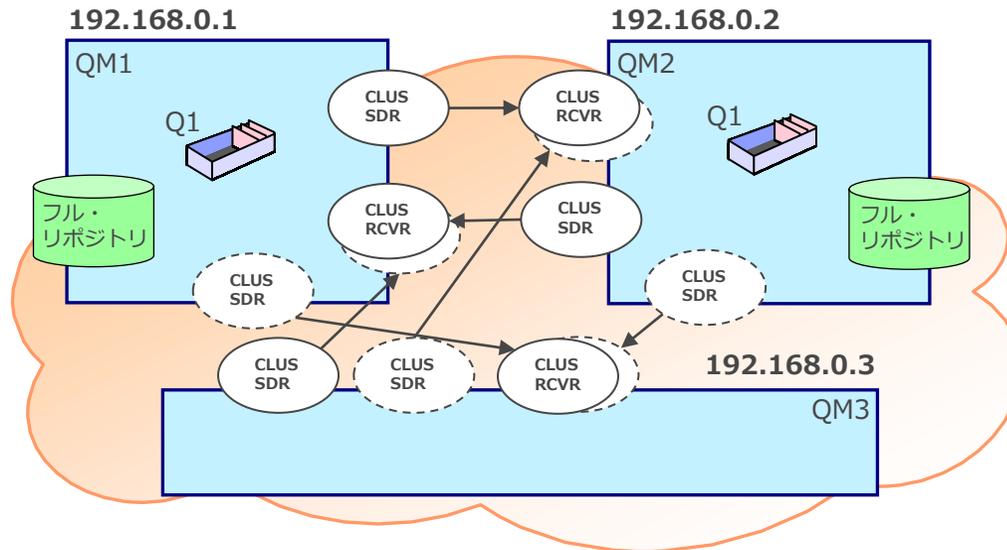
トランスミッション・キュー名	CLCHNAME
XMITQ.CLUS1.QM1	TO.QM1
XMITQ.CLUS1.QM2	TO.QM2
XMITQ.CLUS2.QMA	TO.QMA*

- クラスターに参加しているキュー・マネージャー間で、クラスター情報やユーザー・データをやりとりするために使われるチャンネル定義
  - ◆ 各々のキュー・マネージャーで、以下を明示的に定義
    - クラスター受信チャンネル(CLUSRCVR)
      - CONNAMEに自身へ接続するための情報を設定してMQクラスターに公開
    - クラスター送信チャンネル(CLUSDR)
      - いずれか一方のフル・リポジトリ・キュー・マネージャーに向けて作成
  - ◆ クラスター内のキュー・マネージャーは、該当キュー・マネージャーへの接続が必要になるとクラスター受信チャンネルの情報を入手し、クラスター送信チャンネルを自動定義して接続

# クラスター・チャンネル

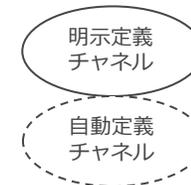
## ■ クラスター・チャンネルは、明示的に定義するものと自動定義されるものがある

	クラスター送信チャンネル (CLUSDR)	クラスター受信チャンネル (CLUSRCVR)
明示定義するチャンネル	複数あるリポジトリ・キュー・マネージャーのうち、1つに対して1本定義	Inbound用を1本定義
定義例	DEFINE CHL(TO.QM2) CHLTYPE(CLUSDR) CONNAME('192.168.0.2(1414)') CLUSTER(CLUS1)	DEFINE CHL(TO.QM1) CHLTYPE(CLUSRCVR) CONNAME('192.168.0.1(1414)') CLUSTER(CLUS1)
自動定義されるチャンネル	その他キュー・マネージャーに対してCLUSRCVRの情報を元に定義	クラスター内キュー・マネージャーからの接続要求に応じて、クラスター受信チャンネル定義と同名のインスタンスが起動 - 同名のチャンネル・インスタンスが複数起動することになる
備考	クラスター情報を送信するために、リポジトリ・キュー・マネージャーに対しては明示定義し、その他は自動定義	CLUSRCVRの情報がクラスター内に公開され、他キュー・マネージャーは、その情報を元にCLUSDRを自動定義可能



※動的に作成された送信チャンネルはDISPLAY CHANNELコマンドでは見えないので注意

- ✓ DISPLAY CLUSQMGRで確認する
- ✓ ステータスは参照可能 (DISPLAY CHSTATUS)
- ✓ START、STOPは可能



### ■ クラスター構成情報を確認するコマンド

#### ◆ DISPLAY CLUSQMGR

- クラスター内のキュー・マネージャー情報を出力

#### ◆ DISPLAY QCLUSTER

- クラスター内のキュー情報を出力

### ■ クラスター構成情報を管理するコマンド

#### ◆ 通常は、クラスター環境に問題が発生した場合やリポジトリ情報に不整合が発生した場合に使用

#### ◆ 通常のオペレーションでは使用しない

#### ◆ REFRESH CLUSTER

- 保持しているクラスター情報の削除、およびフル・リポジトリに対して自身の情報の再送信

#### ◆ RESET CLUSTER

- 特定キュー・マネージャーのクラスター情報の削除、および他のキュー・マネージャーへの使用不可通知
- フル・リポジトリ・キュー・マネージャーでのみ実行可能

### ■ クラスター・キュー・マネージャーを一時停止/再開するコマンド

#### ◆ SUSPEND QMGR

- キュー・マネージャーをクラスター内で一時停止状態にする

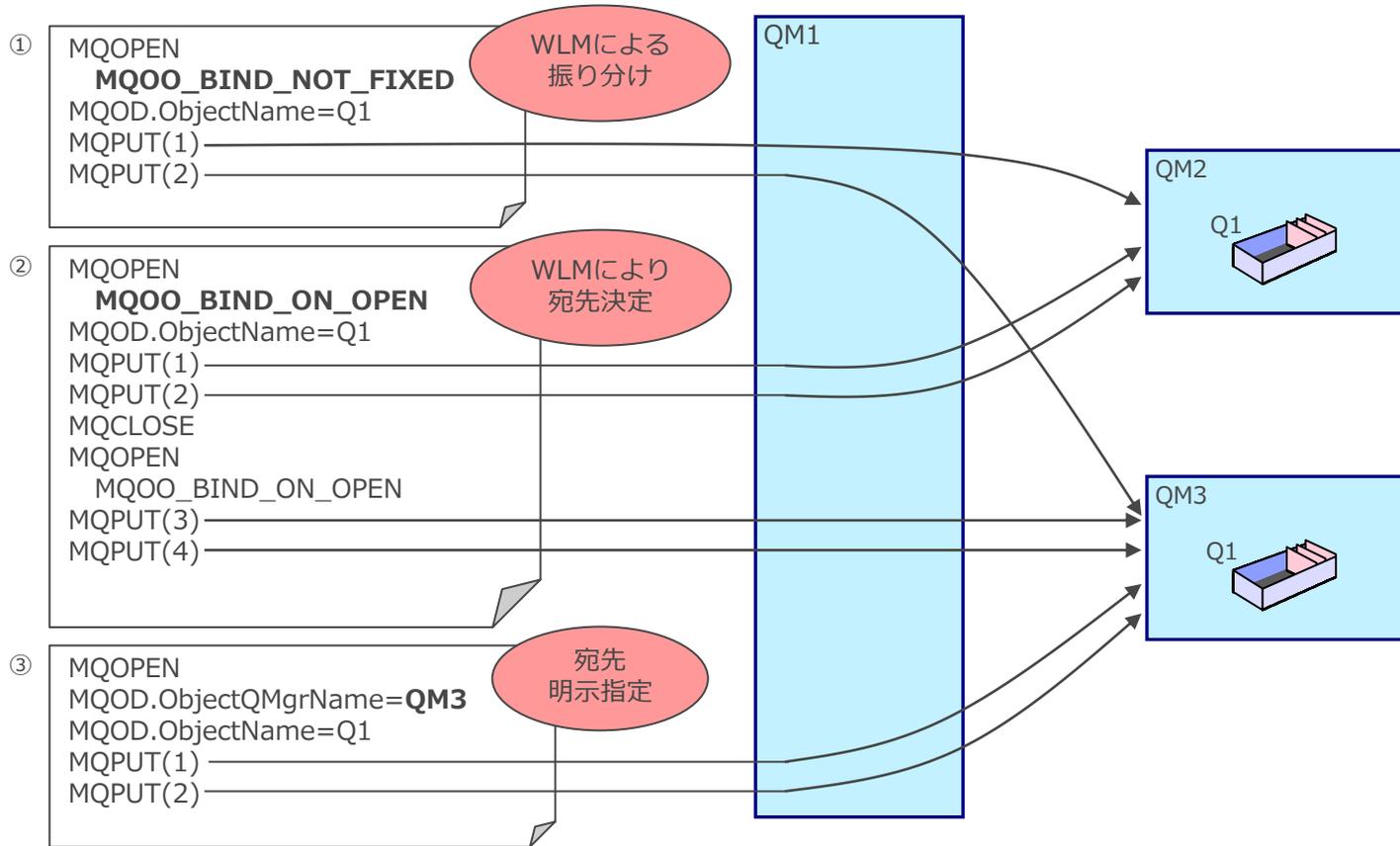
#### ◆ RESUME QMGR

- 一時停止状態から復旧する

# クラスター・キューへの送信方法

## ■ クラスター環境での、キューへの基本的な送信方法は3通り

1. 宛先の選択はQMGRにまかせ、各メッセージ(MQPUT)ごとにメッセージを振り分け(①)
2. 宛先の選択はQMGRにまかせ、MQOPEN単位でメッセージを振り分け(②)
3. 特定のQMGRのクラスター・キューにメッセージを送信する(③)



方法	MQOD. ObjectQMGrName	MQPMO BINDオプション
①	ブランク	NOTFIXED
②	ブランク	OPEN
③	指定(QM3)	-

# 送信方法の指定

## ■ 負荷分散の方法はアプリケーション内で指定

### ◆ 方法①、方法②：ラウンドロビン

- MQOPEN – MQOO\_BIND\_\* オプション で指定

オープン・オプション	選択されるキュー
MQOO_BIND_ON_OPEN	オープン時に宛先が決定され、同一オブジェクト・ハンドルを使用しPUTされた後続のメッセージはすべて同一キューにPUTされる
MQOO_BIND_NOT_FIXED	MQPUT時にWLMによって動的に宛先が決定され、別のクラスター・キューに割り振られる
MQOO_BIND_AS_Q_DEF (デフォルト)	キューの属性DEFBIND(OPEN/NOTFIXED)に従う

補足：MQOO\_BIND\_ON\_OPEN 指定でQMGRによって選択された宛先が返されるMQODのフィールド

- MQOO\_BIND\_ON\_OPEN指定時は、ResolvedQMGrNameに宛先Qmgr名がセットされる

フィールド名	入力	出力	説明
ResolvedQName (MQCHAR48)	×	○	最終宛先キュー名
ResolvedQMGrName (MQCHAR48)	×	○	最終宛先キュー・マネージャー名

- MQOO\_BIND\_NOT\_FIXED指定時は、ResolvedQMGrName は空白が返される  
=> 最終宛先はMQPUT時に動的に決定されるが、アプリケーションからは宛先の検知は不可

### ◆ 方法③：特定のキュー・マネージャーへ送信

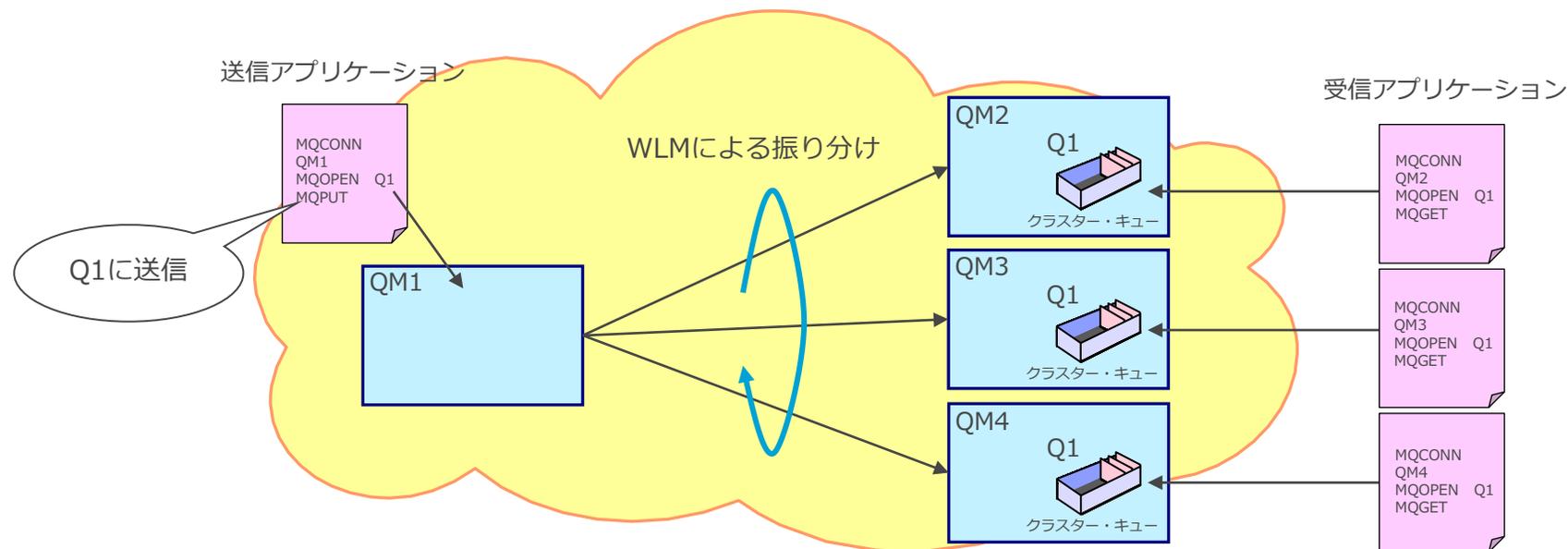
- MQOD.ObjectQMGrName で指定
  - 特定キュー・マネージャー上のクラスター・キューに送信する場合に指定
  - ラウンドロビン対象にする場合は空白

## MQODの指定と宛先の決定

- MQOD.ObjectName : クラスター・キュー名を指定
- MQOD.ObjectQMgrName : ブランク、または宛先キュー・マネージャーを指定
  - ◆ ブランクの場合
    - MQOO\_BIND\_ON\_OPENを指定すると、MQOPEN時に宛先キュー・マネージャーが決定される
    - MQOO\_BIND\_NOT\_FIXEDを指定すると、WLMによる振り分けが行われる
  - ◆ MQODのObjectQMgrNameに明示的にキュー・マネージャー名をセットした場合
    - クラスター・キュー・マネージャーであれば、すべてクラスター・トランスミッション・キューに入れられる  
=> ノン・クラスター・キューでもクラスター・チャンネル経由でメッセージが送られる

入力されたMQOD		実際に設定される名前		
ObjectQMgrName	ObjectName	ObjectQMgrName	ObjectName	転送キュー名
ブランク	ローカル・キュー (クラスター属性)	WLMが設定したCLQMGR名 or MQPUT時に設定されるCLQMGR名	入力されたキュー名	SYSTEM.CLUSTER.TRANSMIT.QUEUE
ローカル・キュー・マネージャー	ローカル・キュー (クラスター属性)	ローカル・キュー・マネージャー	入力されたキュー名	-
ブランク	ローカルにない クラスター・キュー	WLMが設定したCLQMGR名 or MQPUT時に設定されるCLQMGR名	入力されたキュー名	SYSTEM.CLUSTER.TRANSMIT.QUEUE
ローカルになく、クラスター・QM またはクラスター-QMALIAS	Not Resolved	ObjectQMgrName or MQPUT時に設定されるCLQMGR名	入力されたキュー名	SYSTEM.CLUSTER.TRANSMIT.QUEUE

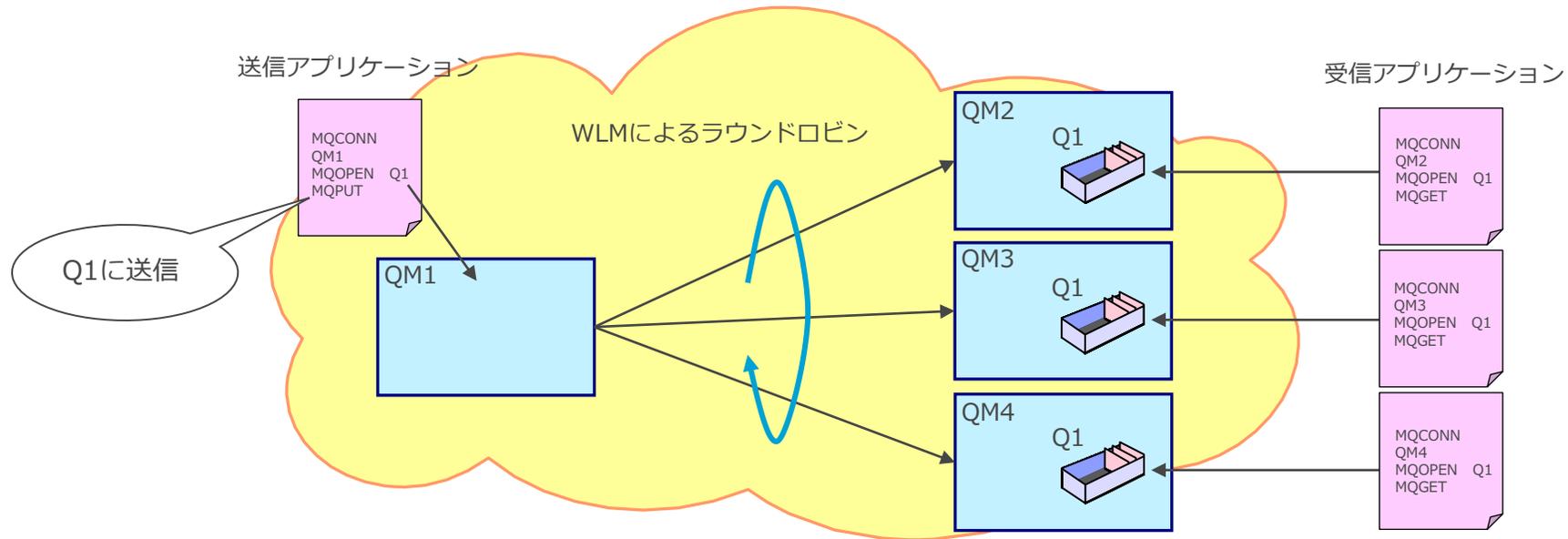
- WLM(ワークロード・マネージャー)によるメッセージの振り分けが可能
  - ◆ 複数キュー・マネージャー上の同一名称のクラスター・キューへ、ラウンドロビン方式で宛先を振り分け
    - メッセージの最終宛先キュー・マネージャーは、WLMにより決定
  - ◆ 振り分け方法は2つあり、送信アプリケーションのMQOPEN時のBINDオプションにて指定
    - MQPUT単位(メッセージ単位)で振分
    - MQOPEN単位で複数メッセージを同一キューに振り分け
  - ◆ 単純なラウンドロビンに加え、重み付けや優先順位を考慮した振り分けが可能 (後述)



# ワークロード・バランシングの制御

## ■ ラウンドロビンによる負荷分散（デフォルト）

- ◆ ローカルのキュー・マネージャーがキューを保持していればローカルのキューにPUT
- ◆ ローカルに存在しない場合、チャンネルのステータス、キュー・マネージャー、キューのステータスをもとに宛先を決定
- ◆ 宛先候補が複数ある場合はラウンドロビンで決定



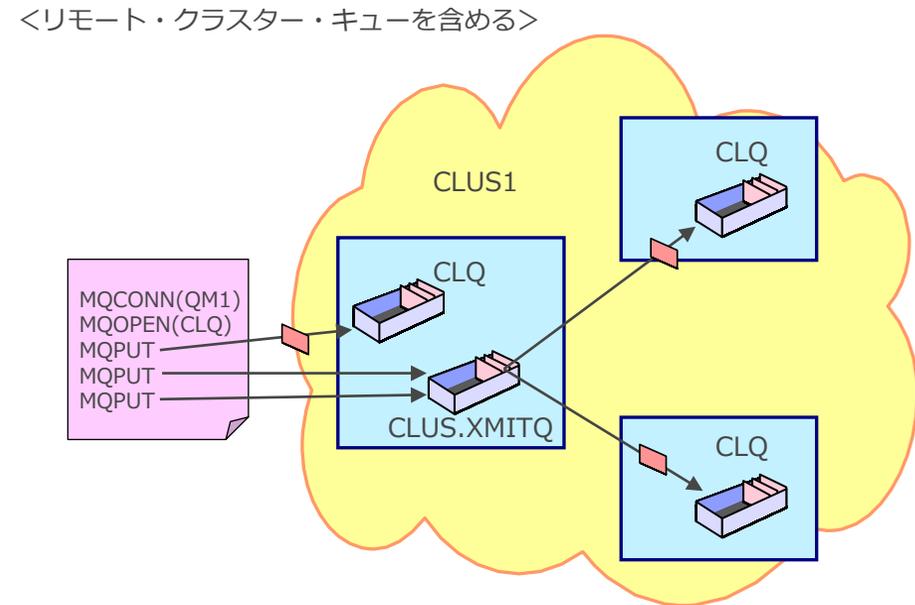
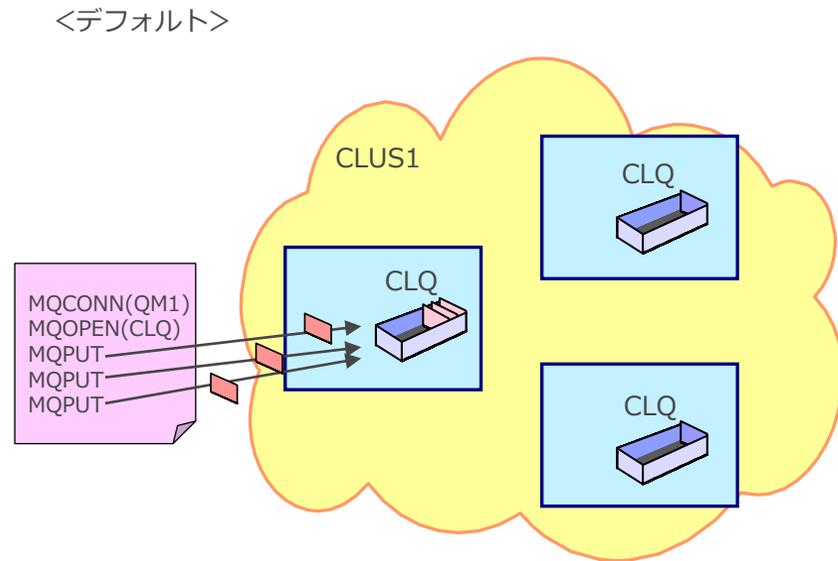
# リモート・クラスター・キューを含めたラウンドロビン

## ■ リモートにあるクラスター・キューも宛先として参加

### ◆ クラスター内に定義してある同一名称のキューを対称にラウンドロビン

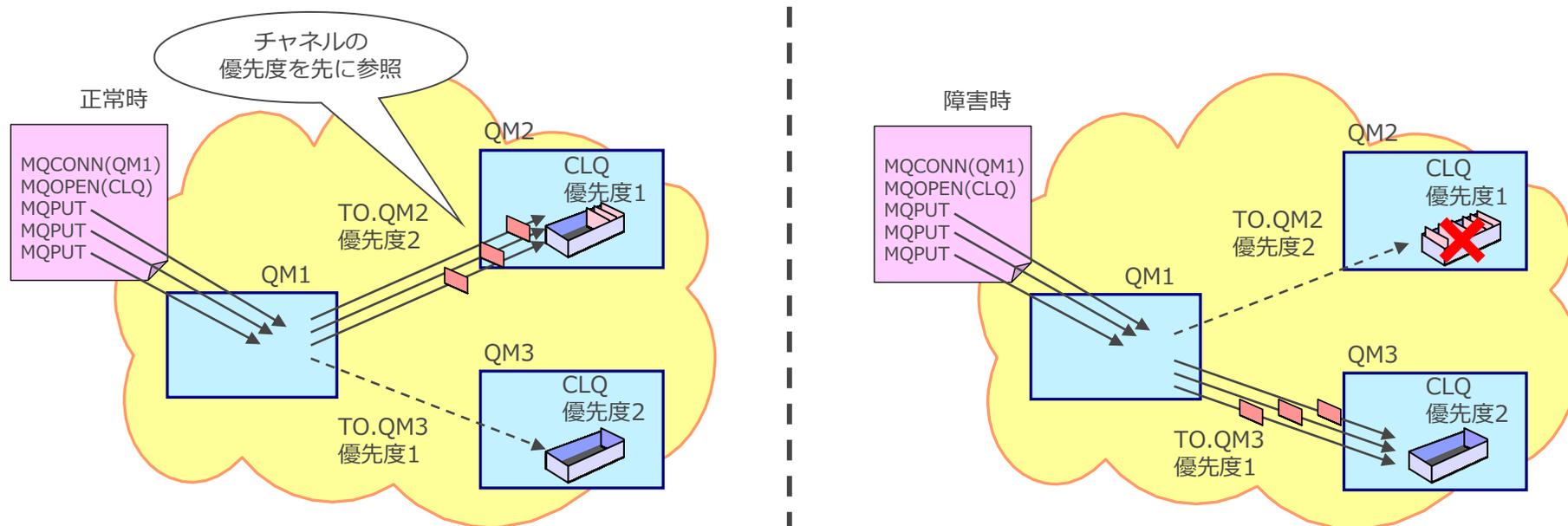
- デフォルトでは、接続するキュー・マネージャーに宛先クラスター・キューがあると、メッセージはすべてそのキューにPUT

### ◆ キューまたはキュー・マネージャーのCLWLUSEQ属性で設定



## ■ キューやチャンネルに設定された優先順位に従ってワークロード・バランシング

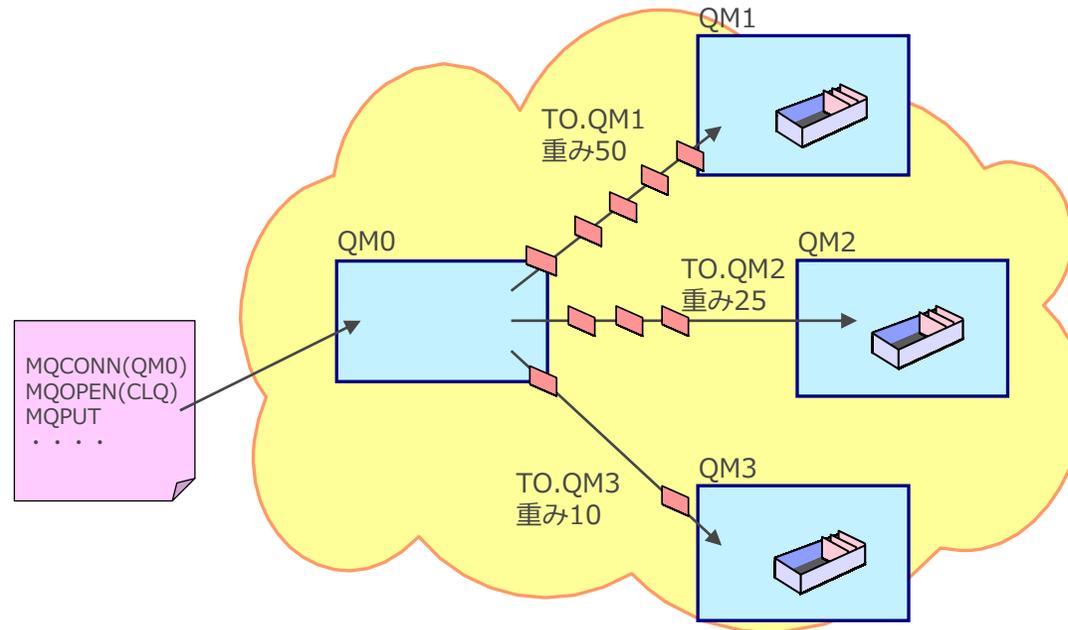
- ◆ 0(最低)~9(最高)の範囲で順位付け
- ◆ 最も高いプライオリティに設定されているチャンネルやキューが使用される
  - メッセージ送信時に、チャンネルやキューのステータスを確認する
  - 最もプライオリティの高いものが使用不可になると、次点でプライオリティの高いキューやチャンネルを使用
    - 復旧すると、再び振り分けの対象となる
  - 優先順位は、チャンネルの値が判断され、次に キュー の値が判断される
- ◆ チャンネルまたはキューのCLWLPRTY属性で設定



## 重み付け

### ■ チャンネルに設定された重み付けに従ってワークロード・บาลancing

- ◆ 1(最低)~99(最高)の範囲で重み付け
- ◆ チャンネルがどのくらいの比重で使用されたかを判断し、宛先を決定
  - 以下の例では、50:25:10 の割合で振分
- ◆ チャンネルのCLWLWGHT属性で設定



#### <補足>

メッセージが送信されれば、その内容に関わりなくチャンネルは使用されたとみなされます。

例えば、クラスター内部情報を送受信するためにチャンネルが使用されても、チャンネルが使用されたと判断されます。

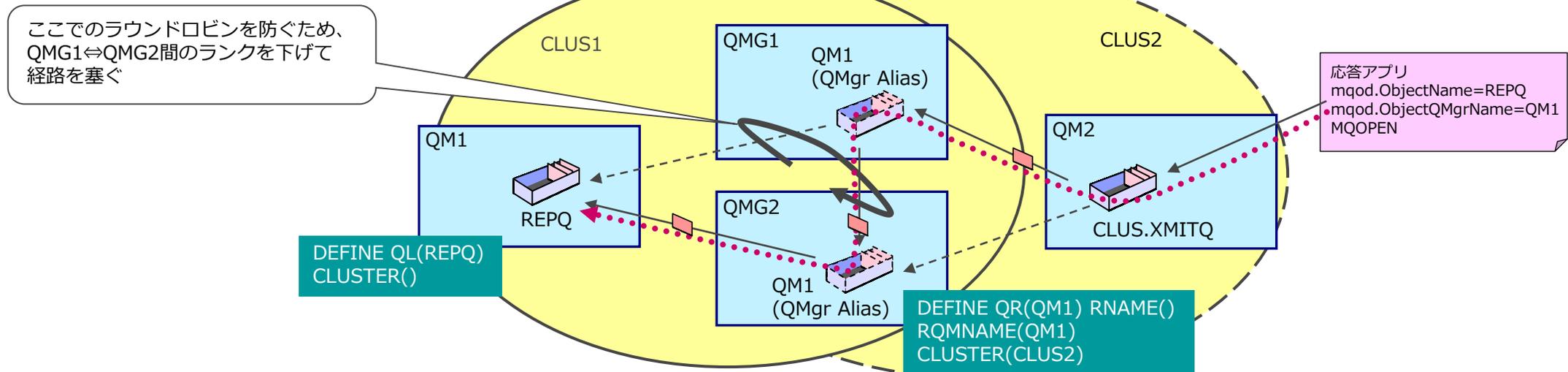
したがって、上記の例では重み付けが50:25:10と定義されていますが、必ずしもユーザー・メッセージが50:25:10で振り分けられるとは限りません。

# ランク

## ■ キューやチャネル設定されたランクに従ってワークロード・バランシング

- ◆ 0(最低)~9(最高)の範囲でランク付け
- ◆ MQクラスターをまたがってRequest/Reply型のメッセージを行うような環境で使用
- ◆ 常に最も高いランクに設定されているキューやチャネルにメッセージを振分
  - 一番ランクの高いものがすべて使用不可になっても、それよりも低いランクのものは使用されない
  - ランクを高く設定することで、クラスター内のルートを固定することが可能
  - ランクは、まずチャネルの値が判断され、次にキューの値が判断される
- ◆ チャネルやキューのCLWLRANK属性で設定

←----- RANKが同じ場合のREPLYメッセージの経路  
◀..... メッセージの経路



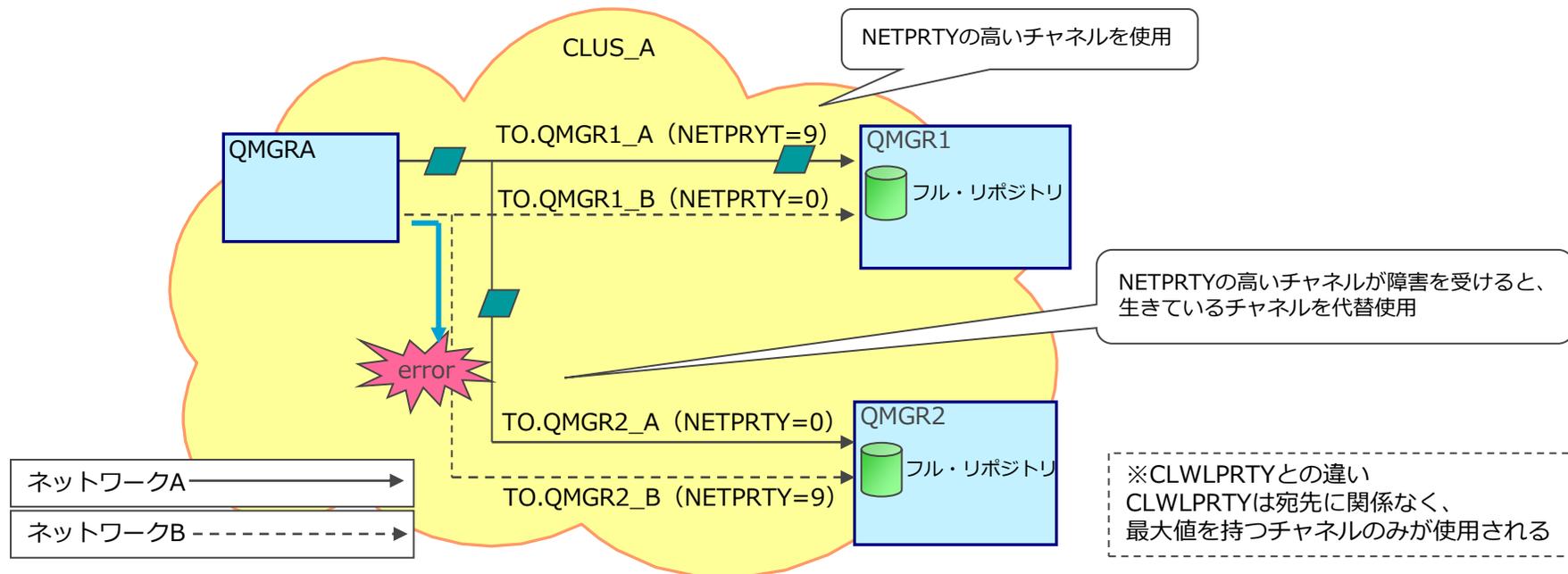
# ネットワークの優先順位

## ■ ネットワークの二重化

- ◆ ルーター障害などに備えて N/W を二重化

## ■ 考慮点

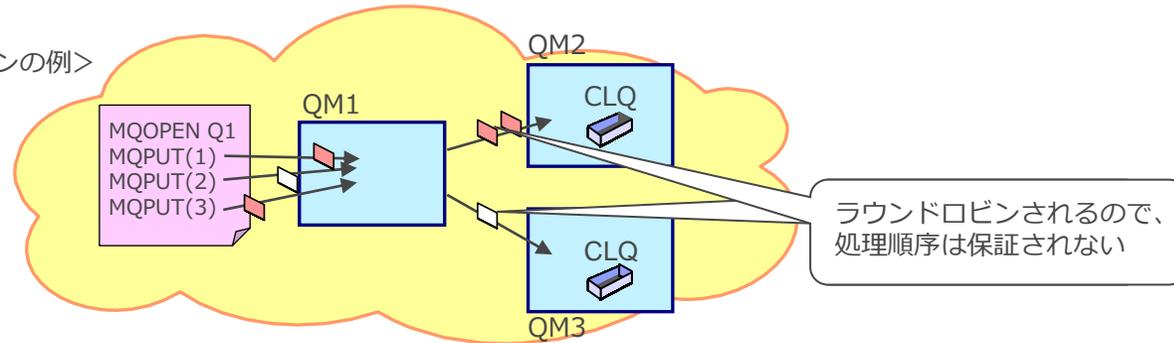
- ◆ デフォルトでは、定義されたチャンネル間でラウンドロビン
- ◆ 特定のネットワークを使用させる場合は、チャンネルのNETPRTY属性を使用
  - 0(最低)~9(最高)の範囲で順位付け
  - 同一宛先でNETPRTYの高いものが障害を受けると、次点のNETPRTYを使用



# MQクラスター構成での考慮点

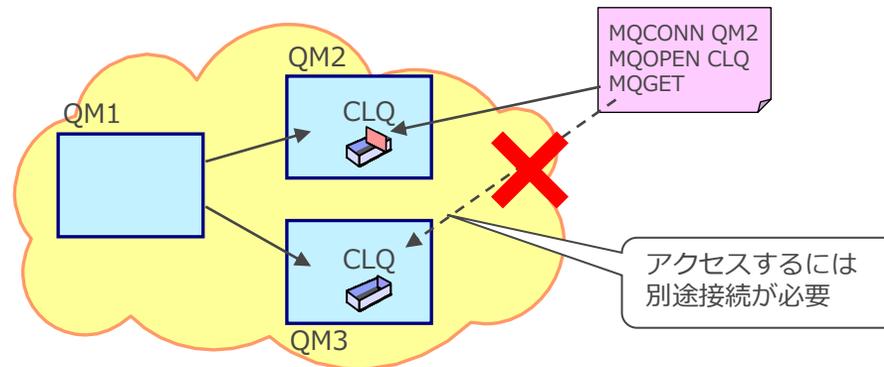
- 宛先キューが複数ある場合、メッセージは複数キュー・マネージャーにラウンドロビンされるため、メッセージの順序性は保証されない
  - ◆ MQOPEN単位であれば、複数メッセージを同一宛先に送ることは可能
    - MQOPEN単位での追い越しは発生する
  - ◆ 関連性のあるメッセージを同一宛先に送ったり、順序性を保証する必要がある場合は考慮が必要

<MQPUT単位でのラウンドロビンの例>



- 受信側アプリケーションから参照可能なのは、接続先キュー・マネージャーがローカルに保持するキューのみ

- ◆ クラスター・キューは、共用キューとは異なる

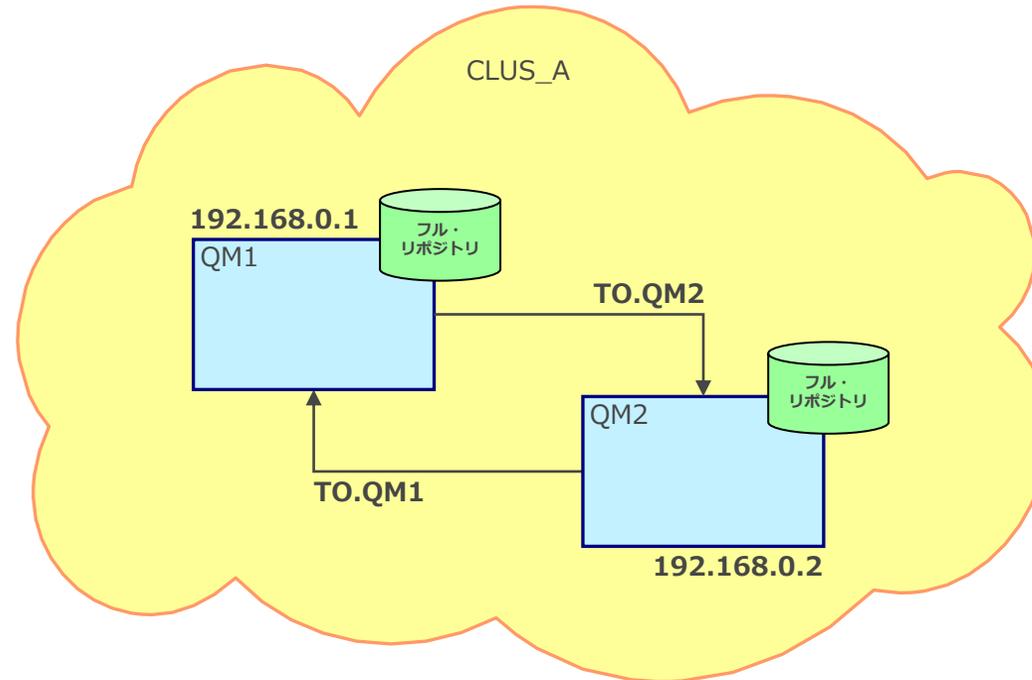


## <参考> MQクラスター環境の構成手順例

### ■ 新たなクラスター環境の作成

#### ◆ 作成する環境

- QM1とQM2で新たなクラスター環境 CLUS\_A を構成
- フル・リポジトリを持つキュー・マネージャーはQM1、QM2とする



## <参考> MQクラスター環境の構成手順例

### ■ 作成手順（前提：キュー・マネージャーは作成済み）

#### ◆ キュー・マネージャーをフル・リポジトリに変更

```
ALTER QMGR REPOS(CLUS_NAME)
```

CLUS\_NAME : クラスター名

#### ◆ クラスター受信チャネルの定義

```
DEFINE CHL(CHL_NAME) CHLTYPE(CLUSRCVR)  
        CLUSTER(CLUS_NAME) CONNAME(HOST_NAME)
```

CHL\_NAME : チャネル名  
CLUS\_NAME : クラスター名  
CONNAME : 自分のIPアドレスやホスト名(\*)

\*クラスター受信チャネルは、RECEIVERチャネルなどのin-boundチャネルとは異なり、自身のCONNAMEを定義する。  
この定義情報を元に、リモートのキュー・マネージャーはクラスター送信チャネルを自動定義する。

#### ◆ クラスター送信チャネルの定義

```
DEFINE CHL(CHL_NAME) CHLTYPE(CLUSSDR)  
        CLUSTER(CLUS_NAME) CONNAME(HOST_NAME)
```

CHL\_NAME : チャネル名  
CLUS\_NAME : クラスター名  
CONNAME : 自分のIPアドレスやホスト名

## <参考> MQクラスター環境の構成手順例

### ■ クラスター構成手順

作業内容	作業内容	作業場所
1. フル・リポジトリに変更	ALTER QMGR REPOS(CLUS_A)	QM1、QM2
2. クラスター受信チャンネルの作成	DEFINE CHL(TO.QM1) CHLTYPE(CLUSRCVR) CLUSTER(CLUS_A) CONNNAME('192.168.0.1(1414)')	QM1
3. クラスター受信チャンネルの作成	DEFINE CHL(TO.QM2) CHLTYPE(CLUSRCVR) CLUSTER(CLUS_A) CONNNAME('192.168.0.2(1414)')	QM2
4. クラスター送信チャンネルの作成	DEFINE CHL(TO.QM2) CHLTYPE(CLUSSDR) CLUSTER(CLUS_A) CONNNAME('192.168.0.2(1414)')	QM1
5. クラスター送信チャンネルの作成	DEFINE CHL(TO.QM1) CHLTYPE(CLUSSDR) CLUSTER(CLUS_A) CONNNAME('192.168.0.1(1414)')	QM2

- ◆ クラスター送信チャンネルから定義すると警告メッセージが出力されるため、クラスター受信チャンネルから定義することを推奨
  - 「AMQ9419：クラスター受信チャンネルがありません。」
- ◆ チャンネル(TO.QM1、TO.QM2)は自動起動される
- ◆ チャンネルが接続されるとクラスターに参加し、クラスター・キューを使用可能

## <参考> MQクラスター環境の構成手順例

### ■ 両方のキュー・マネージャーで、クラスター構成ができたことを確認

#### ◆ 以下は、QM1上での確認例

- MQクラスター(CLUS\_A)に属しているキュー・マネージャーの情報の表示

```
DISPLAY CLUSQMGR(*)
AMQ8441: クラスター・キュー・マネージャーの詳細を表示します。
  CLUSQMGR(QM1)           CHANNEL(TO.QM1)
  CLUSTER(CLUS_A)
AMQ8441: クラスター・キュー・マネージャーの詳細を表示します。
  CLUSQMGR(QM2)           CHANNEL(TO.QM2)
  CLUSTER(CLUS_A)
```

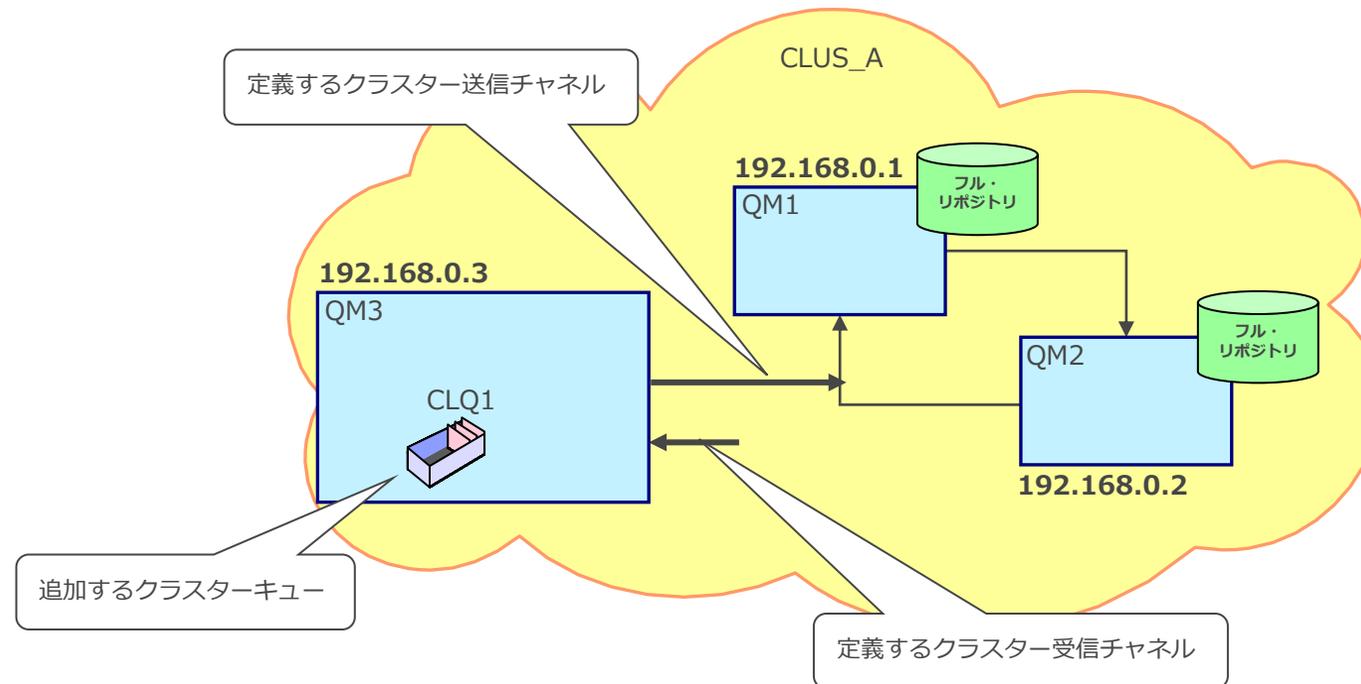
- チャンネル状況の稼動状況の確認

```
DISPLAY CHSTATUS(*)
AMQ8417: チャンネル状況の内容を表示します。
  CHANNEL(TO.QM2)           CHLTYPE(CLUSSDR)
  CONNAME(192.168.0.2 (1414))  CURRENT
  RQMNAME(QM2)              STATUS(RUNNING)
  SUBSTATE(MQGET)           XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)

AMQ8417: チャンネル状況の内容を表示します。
  CHANNEL(TO.QM1)           CHLTYPE(CLUSRCVR)
  CONNAME(192.168.0.1)       CURRENT
  RQMNAME(QM1)              STATUS(RUNNING)
  SUBSTATE(RECEIVE)         XMITQ()
```

## <参考> MQクラスター環境の構成手順例

- パーシャル・リポジトリを持つキュー・マネージャーをクラスターに追加
  - ◆ 追加する定義
    - QM3をクラスターCLUS\_Aに追加
- クラスター・キューをキュー・マネージャーQM3に追加
  - ◆ 追加する定義
    - CLQ1をクラスター・キューとして、CLUS\_Aに公開



## <参考> MQクラスター環境の構成手順例

### ■ パーシャル・リポジトリを持つキュー・マネージャーの追加手順

#### ◆ 明示的に接続するリポジトリ・キュー・マネージャーの決定

- 明示的に接続するキュー・マネージャーは1つでよい
  - ここではQM1とする

#### ◆ クラスター受信/送信チャネルの定義 => p.95参照

- 両チャネルを定義した時点でチャネルが自動起動
- チャネルが接続されると自分の情報を送信し、フル・リポジトリからクラスター情報を受信

### ■ クラスター・キューの定義

```
DEFINE QL(Queue_NAME) CLUSTER(CLUS_NAME)
```

QUEUE NAME : キュー名  
CLUS\_NAME : クラスター名

### ■ 実際の追加作業

#### ◆ クラスター受信/送信チャネルの定義 (QM3上で定義)

1. クラスター受信チャネルの作成	DEFINE CHL(TO.QM3) CHLTYPE(CLUSRCVR) CLUSTER(CLUS_A) CONNAME('192.168.0.3(1414)')
2. クラスター送信チャネルの作成	DEFINE CHL(TO.QM1) CHLTYPE(CLUSDR) CLUSTER(CLUS_A) CONNAME('192.168.0.1(1414)')

#### ◆ クラスター・キューCLQ1の追加 (QM3上で定義)

1. クラスター・キューの作成	DEFINE QL(CLQ1) CLUSTER(CLUS_A)
-----------------	---------------------------------

## <参考> MQクラスター環境の構成手順例

---

- キュー・マネージャーがクラスターに追加されたかを確認
  - ◆ p.97参照
- クラスター・キューが追加されたかを確認
  - ◆ リポジトリ・キュー・マネージャーQM1、QM2上で確認
    - リポジトリ・キュー・マネージャー上では、クラスター・キュー定義が見える

```
DISPLAY QCLUSTER(*) CLUSQMGR PUT
AMQ8409: キューの内容を表示します。
QUEUE(CLQ)           TYPE(QCLUSTER)
CLUSQMGR(QM3)        PUT(ENABLED)
```

# Uniform Cluster

## ■ V9.2(LTS)/V9.1.2(CD)から登場した、可用性と拡張性のための新たなクラスター構成

◆ MQ for z/OSは非対応

## ■ Uniform Cluster構成

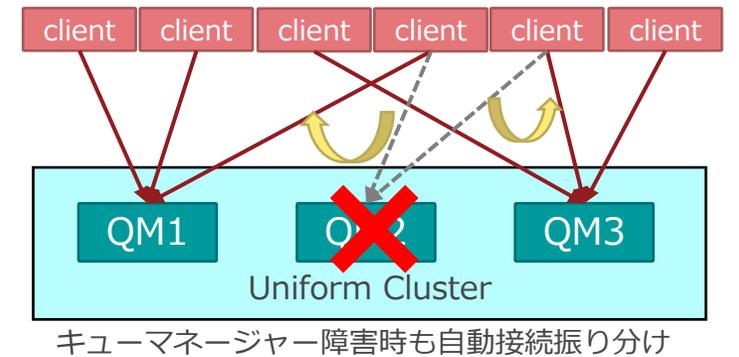
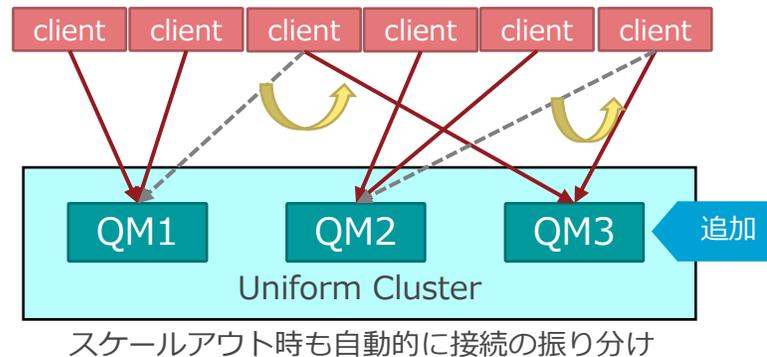
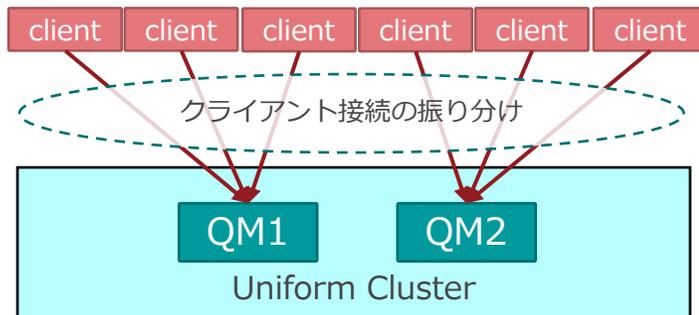
◆ 複数のキューマネージャーをグルーピングしてUniform Clusterを構成

- Uniform Clusterの構成要素は従来のMQクラスター構成とほぼ同等
  - クラスター送受信チャネルやフル/パーシャル・リポジトリ
  - qm.iniにUniform Cluster用のパラメータが追加

## ■ 自動balancing機能

◆ クライアント・アプリケーションからの接続をUniform Cluster内のキューマネージャーに分散

- アプリケーションには透過的に接続数を平準化
- キューマネージャーやクライアント・アプリケーションが増減した場合にも自動的に対応
  - ただし、クライアント・チャネル定義テーブル(CCDT)の変更は必要



## 従来のMQクラスターとの主な相違点

### ■ MQクラスター構成

#### ◆ キューマネージャー間接続

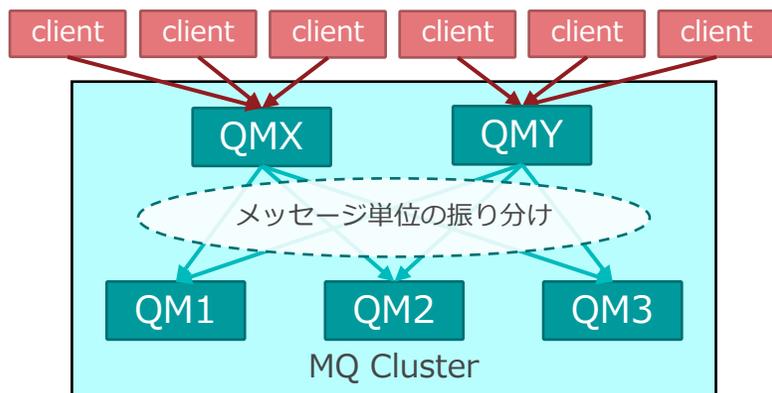
- クライアントサイドにもMQ導入が必要

### ■ メッセージの振り分け

#### ◆ クラスターキューへのMQPUT単位もしくはMQOPEN単位で振り分けが行われる

### ■ キューのPUT属性で振り分け制御可能

#### ◆ クラスターキューのPUT属性をDISABLEDにすることで振り分け対象から外すことができる



### ■ Uniform Cluster構成

#### ◆ クライアント接続

- クライアントサイドにMQ導入が不要

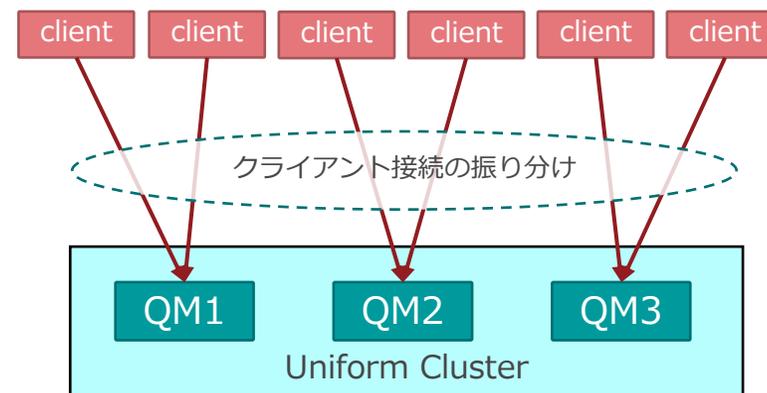
### ■ クライアント接続の振り分け

#### ◆ クライアント接続数が平準化するように自動的に接続の振り替えが行われる

### ■ キューのPUT属性では振り分け制御不可

#### ◆ 接続レベルの振り分けのため、キューの属性では制御できない

#### ◆ キューマネージャーのSUSPENDによる制御は可



### ■ Uniform Clusterを構成するためのキュー・マネージャーの前提

#### ◆ MQ for multiplatform V9.1.2以上（z/OS版は非対応）

- V9.1.2~V9.1.4のキューマネージャーとV9.1.5~のキューマネージャーでUniform Clusterを構成することは不可

#### ◆ 2つ以上のキューマネージャーで構成

- 厳密な制約ではないが、キューマネージャー数は10以下が推奨
  - キューマネージャー数が増えると内部の情報交換のためメッセージ量が増大し、性能に重大なインパクトを与えるため

#### ◆ QMGR属性のPSCLUS/PSMODEはENABLED(デフォルト)を設定する

- 内部の情報共有をPub/Subで行うため

#### ◆ 各キューマネージャーは同一構成とする

- 自動バランシング機能では、クラスター内の全てのキューマネージャーを対象に接続を振り分けるため、サーバ接続チャンネルやキュー構成、セキュリティ設定は同一構成としておく
  - クラスター送受信チャンネルはキューマネージャーごとに異なる設定となる

### ■ 考慮点

#### ◆ 1つのキューマネージャーが複数のUniform Clusterに同時に参加することはできない

#### ◆ Uniform Clusterに参加しているキューマネージャーが他のMQクラスターに参加することは可能

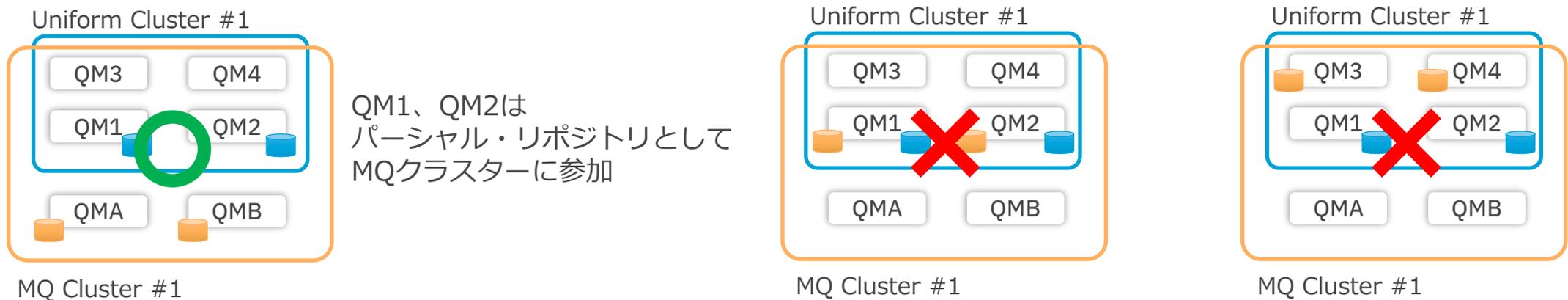
- 複数のMQクラスターにパーシャルリポジトリとして参加することができる
  - Uniform Cluster内のキューマネージャーが他のMQクラスターのフルリポジトリになることは不可
- アプリケーションからのメッセージを受信するキューは、クラスター・キューである必要はない
  - Uniform Clusterではメッセージ・レベルの振り分けは行わないため、ローカルキューでもよい

## <補足> キューマネージャの前提・考慮点

- 1つのキューマネージャが複数のUniform Clusterに同時に参加することはできない



- Uniform Clusterに参加しているキューマネージャが他のMQクラスターに参加することは可能
  - ◆ 複数のMQクラスターにパーシャル・リポジトリとして参加することができる
    - Uniform Cluster内のキューマネージャが他のMQクラスターのフル・リポジトリになることは不可



### ■ 自動バランシング機能を利用するためのアプリケーションの前提

- ◆ MQ for multiplatform V9.1.2以上（z/OS版は非対応）
- ◆ C、JMS、IBM MQ .NET、XMS .NETで書かれたクライアント・アプリケーションであること
  - バインディング(ローカル)接続はUniform Clusterの振り分け対象外
  - チャンネルのSHARECNV属性は1以上
- ◆ 自動再接続機能を利用
  - 実装言語によっては自動再接続機能がサポートされない点に注意
    - BaseJavaは未サポート、JMSもJava EEのWebコンテナ/EJBコンテナ環境ではサポートされない
  - アプリケーションには透過的に接続先が変わるケースがあるため、アプリケーションの実装には様々な考慮点がある
    - <https://www.ibm.com/docs/en/ibm-mq/9.3?topic=managers-application-recovery>
  - クライアント・チャンネル定義テーブル(CCDT)を利用
    - 従来からのバイナリ形式のCCDTも利用できるが、チャンネル構成が煩雑になるためJSON形式の利用を推奨
- ◆ 長時間接続し続けるアプリケーションであること
  - 接続の振り替えはアプリケーションの接続後に行われるため、短期間に接続/切断を繰り返すアプリケーションでは振り替えが行われない

### ■ 考慮点

- ◆ 要求・応答型のアプリケーションで利用するには注意が必要
  - 要求メッセージを送信後、応答メッセージを待つ間に接続先が切り替わってしまうと応答メッセージを受け取れない
    - V9.2.4以降、要求応答間での再接続を抑止するオプションパラメータが提供されている
      - <https://www.ibm.com/docs/ja/ibm-mq/9.3?topic=clusters-influencing-application-re-balancing-in-uniform>

## <参考> Native HA

### ■ MQ独自のログ・レプリケーション機能によるフェイルオーバー構成

◆ Kubernetes/OpenShift環境で構成可能

◆ キューマネージャーは3つのPodで稼動

- 1つがActiveインスタンスとして稼動、残りはReplicaインスタンスとして待機

◆ 各PodがPersistent Volume上にログのコピーを保有

- MQプロセスにてActiveからReplicaへログがコピーされる

◆ 障害時にフェイルオーバーしたキュー・マネージャーへの接続切り替えは、Kubernetes/OpenShiftのServiceで解決

- 接続元のチャンネルやクライアントは1つの宛先に対して再接続するだけでよい

◆ 追加のクラスタ製品やストレージの冗長構成が不要

### ■ Cloud Pak for Integrationライセンスに含まれるMQ Advancedでサポート

### ■ MQ V9.3からMQ Operatorを利用せずに構成することも可能

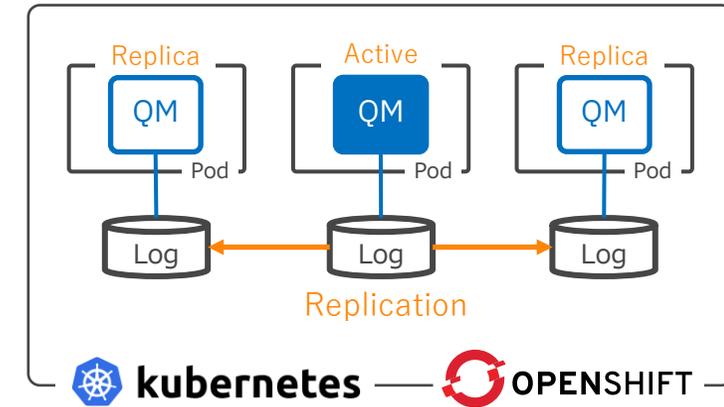
◆ OpenShift以外のKubernetes環境に構成することが可能に

- サンプルのHelmチャートを公開 (<https://github.com/ibm-messaging/mq-helm>)

◆ MQ Operatorを利用しない場合は、ユーザー実装のコンテナ・イメージを利用

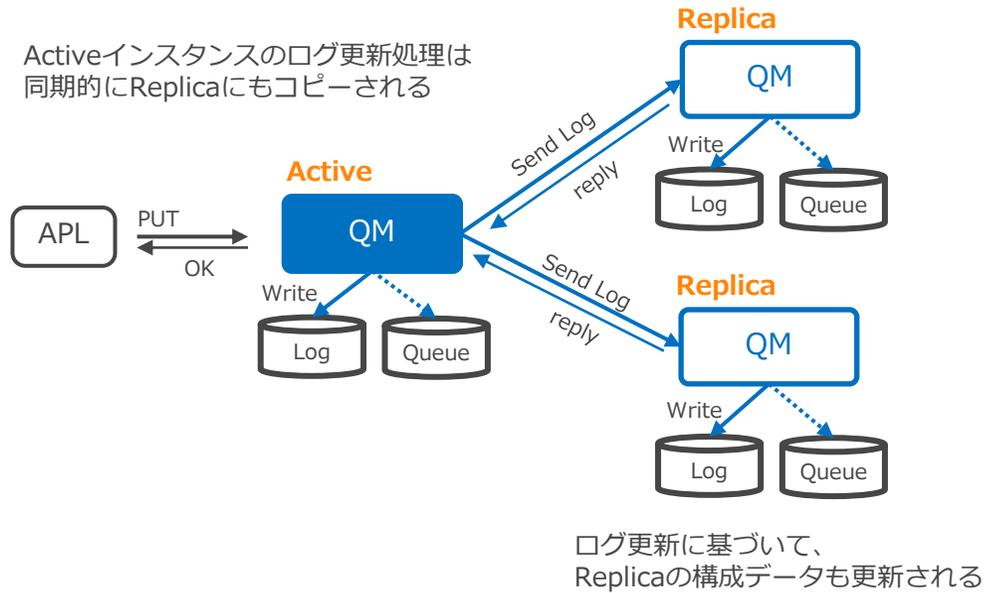
### ■ 構成手順

◆ IBM MQ Native HA構成ガイド : <https://www.ibm.com/support/pages/node/6840043>



# <参考> Native HAの動き

## ■ ログ・レプリケーション



## ■ 障害時

- Activeインスタンスは定期的にハートビートを送信し、自身のリーダーシップを主張
- 間隔はqm.iniのHeartbeatIntervalで設定可能 (デフォルト5秒)

