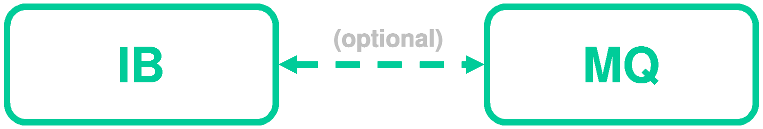
Flexible MQ Topologies in IBM Integration Bus V10

Christopher Frank

IBM Systems Middleware

Agenda

- **Flexibility for MQ**
- **MQ Environments**
  - Cloud, Clients and Connectivity
- **Administration**
- **Migration**
- **Installation changes**
- **MQ Node changes**
- **Changes to other MQ dependent function**
  - Nodes: Collector, Aggregate, Sequence, Timeout
  - Pub-sub changes
- **High availability**
- **Administration Security**
- **Transactionality**
- **CMP, Web Admin Ports and the Toolkit**



Simplify relationship

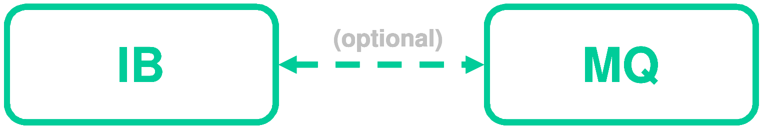
Remove limitations

Platforms

Platform Support

MQ Flexibility

MQ Flexibility

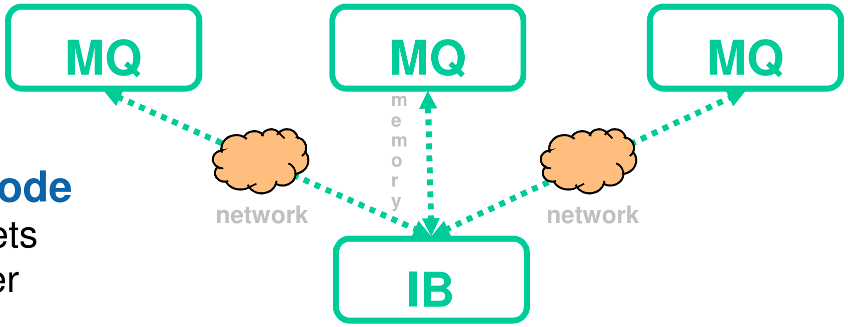


Simplify relationship

Remove limitations

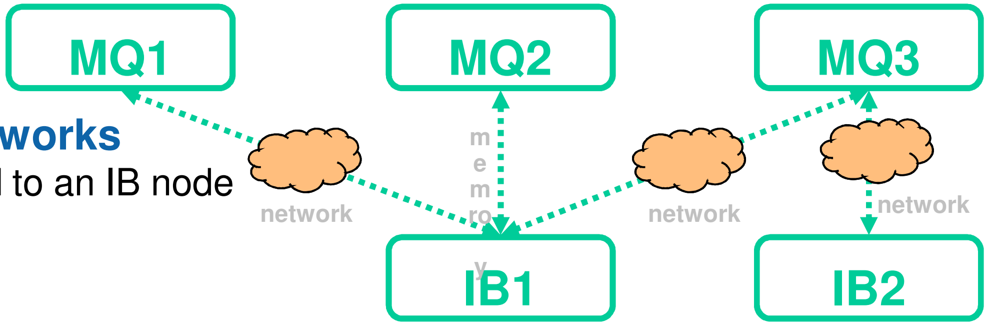
- **Message Broker and Integration Bus have always required a local queue manager**
- **Local Queue Manager provided:**
  - Persistence for resources (Aggregate nodes, SAP, etc)
  - Publish-subscribe capability (Statistics, Business Activity Monitoring / Record-Replay)
  - Global Transaction Management
  - Inter-process communication (IPC) between IIB Components and CMP-based applications
  - Administration Security Management
  - High-availability
- **MQ Nodes only work with local queue manager, therefore the messages “have to come to us!”.**
- **Actual business requirement: “I want to be able to get/put MQ messages to my queue managers”!**

MQ Environments & Topologies



- **Existing MQ infrastructure, new Integration Node**
  - Requirement to be able to tap into existing MQ assets
  - Multiple IB Nodes can use the same queue manager
- **V9 Migration Environment – existing Queue Manager assigned to IB**
- **Cloud Environment**
  - MQ assets separate from IB nodes, and dynamically discoverable via cloud services
- **Ability to "go where the messages are!"**
  - IB node can access messages in multiple queue managers – also enables Active-Active HA scenarios.
- **No MQ Requirement?**
  - Some customers don't use MQ!
  - Provides a lighter footprint
  - Still require full administrative control
  - Not available on zOS (MQ still required).

MQ connectivity options in v10



- **Developers can access existing MQ networks**
  - No longer require queue manager dedicated to an IB node
  - Multiple IB nodes can access the same QM
  - MQ v7.1, v7.5 and v8 supported

- **New MQ Input/Output/Get/Reply node options**
  - Can now use any form of MQ connectivity
  - Local, remote (client), or both in same flow

- **Reconnection supported**
  - Both local and remote connections will reconnect if link is broken
  - Node and Servers stay up, allowing flows to continue running against available MQ systems and other unaffected transports and resources (databases, Web Services, etc)

- **MQ Input Nodes can receive messages from queues on multiple queue managers**
  - Transactional behavior will require MQ Input and MQ Output nodes to use same Local Queue Manager if globally coordinated.
  - Only QM associated with Integration Node can be globally coordinated, other QMs are 1PC (except on zOS where all QMs are global).

- **Enabled for MQ Policy changes!**

Integration Bus and MQ Administration

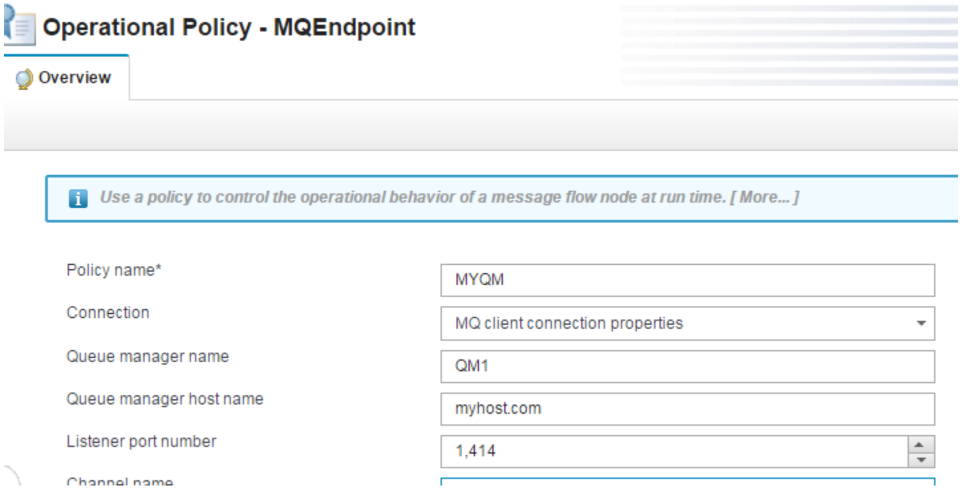
- **IB nodes no longer tied to a local Queue Manager**
  - Reduces complexity in local administration
  - No MQ admin knowledge needed for IB administration

- **QM links defined by policy/configurable services**
  - Allows for easy management of IB to MQ interactions

- **Internal queues replaced by other technologies**
  - IB can be administered without requiring MQ for internal processes.

- **IBX retired**
  - Web Admin UI enhanced
  - Capabilities merged

- **MQ nodes fully support v10 Policy to allow management of QM connections**



Use a policy to control the operational behavior of a message flow node at run time. [More...]

Policy name*	MYQM
Connection	MQ client connection properties
Queue manager name	QM1
Queue manager host name	myhost.com
Listener port number	1,414
Channel name	

Installation changes

- **IB V10 will no longer require MQ to be installed as a prerequisite (except on zOS)**
  - No requirement to have MQ on the box at any point
  - Similar to database connectivity, where there is no requirement (since WMB 7.0) to have a database installed.
  - IIB License still provides full MQ entitlement
- **IB will pick up MQ libraries for automatic use**
  - If MQ (Client or Server) is installed, then IB will configure the environment appropriately
    - Will pick up MQ install added after IB itself has been installed
- **Some IB functionality requires MQ**
  - Product will install without MQ and leave the functionality unavailable
  - Similar to database connectivity.
- **Automated installation simplified**
  - MQ resources not installed at the same time
  - Reduces dependency management
  - Simplifies cloud-based installs

Queue Manager associated with the Integration Node

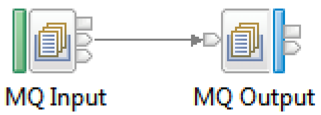
- **mqsicreatebroker BrokerName -q [qmgr]**
  - Creates an Integration Node, with a default queue manager
  - No queues created now
  - Queue Manager is not started when “mqsistart” is run.

- **mqsichangebroker BrokerName -q [qmgr]**
  - Queue Manager definition can now be changed

- **IIB Default Queue Manager is used for MQ nodes (MQInput, MQOutput…) whenever WMQ is being used, and no explicit queue manager has been defined.**
  - E.g. For migrated MQ nodes

- **Associating the queue manager will default a number of distinct things:-**
  - Queue Manager to use for MQ nodes
  - Queue Manager to use for internal use (EDA nodes, etc)
  - Queue Manager to act as Transaction Manager for global coordination (distributed)
  - IIB will forward pub-sub messages to queue manager

MQ-specific nodes



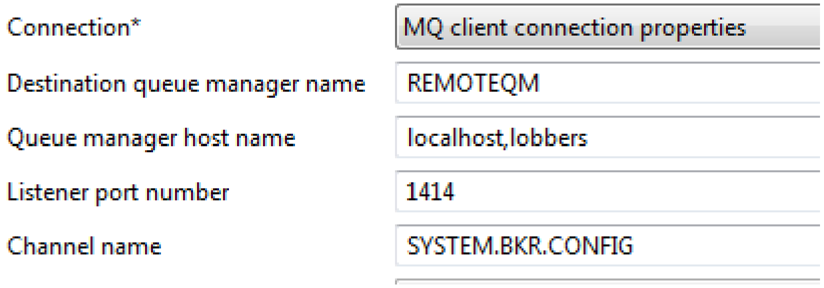
- **More flexible options, enabling simpler interactions with existing MQ networks**
  - Administratively define MQ links
  - No need to put client parameters on nodes!

- **Defaults for migrated Integration Nodes will be to use existing local QM**

- **All MQ nodes support**
  - Client and CCDT-based links (Apart from z/OS)
  - Server connections to default QM for the Integration Node (as per v9)
  - Server connections other local QMs
  - Automatic reconnect to queue managers

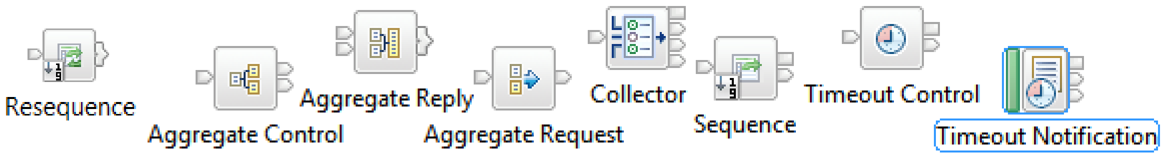
- **MQInput and MQGet will be able to receive message from multiple queue managers using MQ failover (CCDT or multiple hostnames)**
  - Specified via policy/configurable services, avoiding need for redeploy when MQ network changes
  - Transactionality limitations depending on configuration
    - Not all configurations will support XA initially

- **MQOutput/MQReply able to put to remote queues directly (as well as local)**



Connection*	MQ client connection properties
Destination queue manager name	REMOTEQM
Queue manager host name	localhost,lobbers
Listener port number	1414
Channel name	SYSTEM.BKR.CONFIG

IIB MQ usage



- **Some message flow nodes require a persistent store for messaging-related data and use local QM for this**
  - Collector, Aggregate, Sequence, etc

- **These message flow nodes will require local QM to store data**
  - QM connections specified via policy/configurable services as with MQ nodes

- **HTTP and SOAP nodes can use a MQ-based proxy servlet**
  - Will continue to be able to do so for compatibility

IIB MQ Usage

- **Certain function requires MQ resource definition (‘SYSTEM.BROKER.xx’ queues):**
  - Collector / Aggregate / Sequence / Timeout nodes (“EDA” Nodes)
  - Broker-wide HTTP Listener
  - SAP Input node (if transactional)
  - Record-Replay
- **This functions still rely on the presence QM associated with the Integration Node.**
- **Necessary resources can be defined via scripts [install\server\sample\wmq\iib_createqueues.sh/.cmd]**

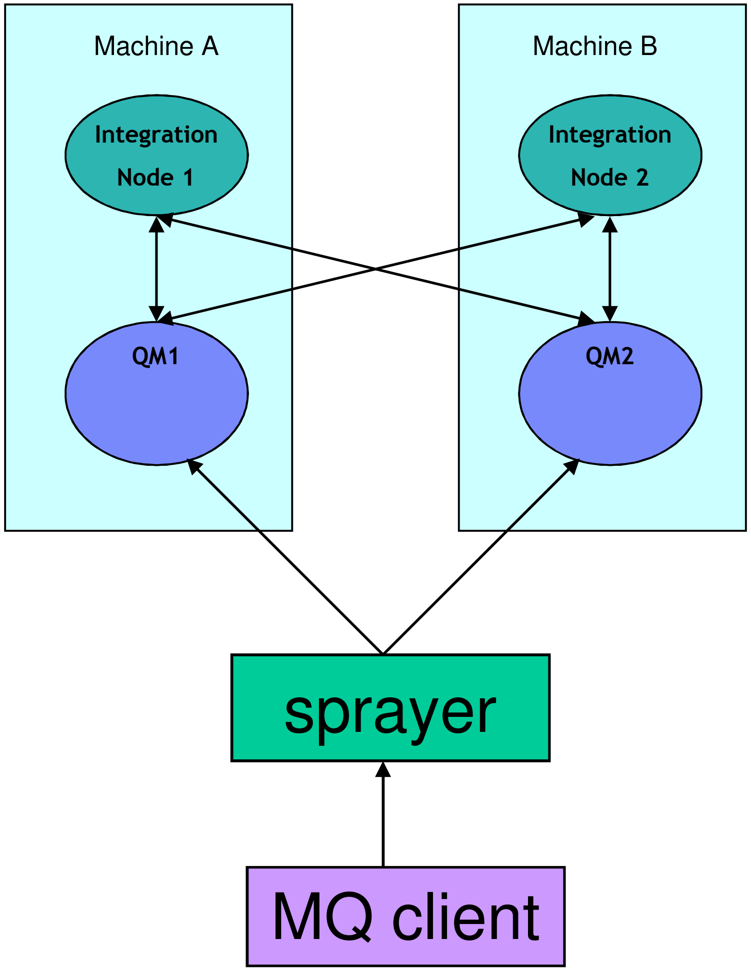
SSL Connectivity

- HTTPS based auth for remote administration and toolkit connections using web administration port.

- MQ Nodes can be configured using broker wide Keystore for SSL (‘mqsireportproperties <Node> -o BrokerRegistry –r’ now has a ‘mqKeyRepository’ and ‘mqCCDT’ property)

- SSL Peer name and SSL CipherSpec can be set in policy or MQ node properties

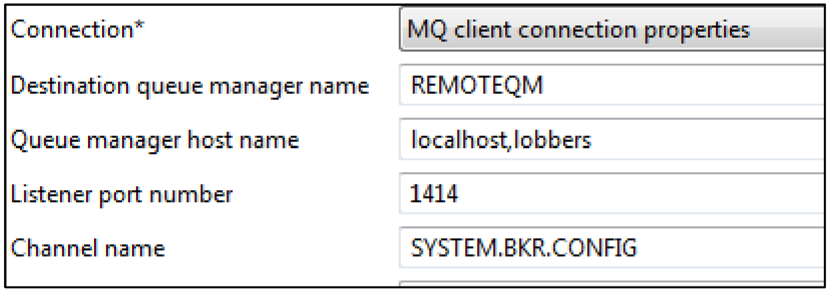
High Availability



- MQ CCDT Definition allows multiple target QM endpoints and will failover
- Comma-separated hostnames in MQ nodes.
- Creates possibility of an active-active model.
- IB nodes no longer tied to a QM, and can access messages on QM using either local or remote connections
- Failure of one QM or one node still allows messages to flow through the system
- QM1 and QM2 could be multi-instance queue managers, with multiple systems providing active/passive configuration
  - IB would access messages from which ever instance is up, reconnecting as needed.
- Multi-QM access also enables easier cloud integration
  - Cloud-based MQ services accessible from any IB node

Multi-instance QMs

- 2 Patterns: Client connections, and local connections.
- If local, then broker instance must failover with the queue manager (QM assoc. with integration node).
- If client, then failover can be achieved using either CCDT or comma-separated hostname on the node properties or policy.



Connection*	MQ client connection properties
Destination queue manager name	REMOTEQM
Queue manager host name	localhost,lobbers
Listener port number	1414
Channel name	SYSTEM.BKR.CONFIG

Publish Subscribe

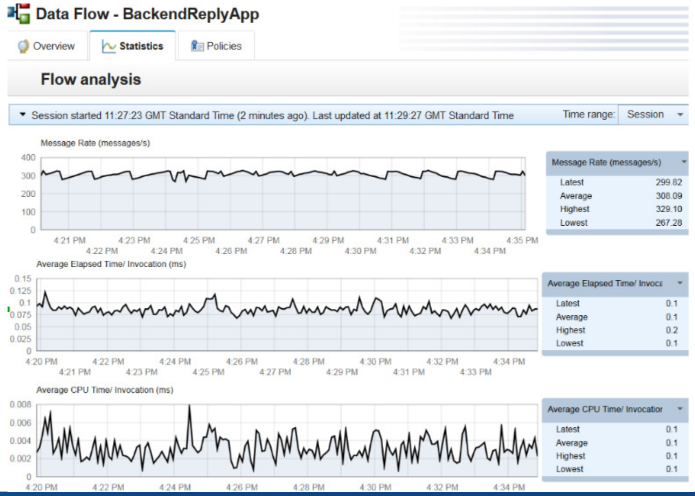
- **MQTT Broker embedded in IIB**
  - Provides OOTB pub-sub for events emitted by integration servers.
  - Enabled by default (can be disabled)
  - MQTT publications can be routed to an external MQTT broker such as MessageSight.

- **Publication emitted by the Integration node classified as :-**
  - Accounting and Flow Statistics
  - Resource Statistics
  - Business Activity Monitoring

- **Events can be routed to MQ (QM associated with integration node) or MQTT broker**

- **Events also published to QM associated with IIB Node for backward compatibility**

```
mqsireportproperties TESTNDE_xyz -b pubsub -o ...

OperationalEvents
  MQ
    policyUrl=''
    enabled='true'
    format=''
  MQTT
    policyUrl=''
    enabled='true'
AdminEvents
  MQ
    policyUrl=''
    enabled='true'
    format=''
MQTT
    policyUrl=''
    enabled='true'
BusinessEvents
  MQ
    policyUrl=''
    enabled='true'
    format=''
  MQTT
    policyUrl=''
    enabled='false'
MQTTServer
  enabled='true'
  port='11884'
```



Data Flow - BackendReplyApp

Flow analysis

Session started 11:27:23 GMT Standard Time (2 minutes ago). Last updated at 11:29:27 GMT Standard Time

Time range: Session

Message Rate (messages/s)

Latest	299.82
Average	308.09
Highest	329.10
Lowest	267.28

Average Elapsed Time/ Invoca

Latest	0.1
Average	0.1
Highest	0.2
Lowest	0.1

Average CPU Time/ Invocatior

Latest	0.1
Average	0.1
Highest	0.1
Lowest	0.1

Transactionality

- **IB can manage local (1PC) transactions, or use MQ to provide two-phase (XA) coordination**
  - IB-managed transactions will continue to support all resource managers
- **Global Two-phase Commit (2PC) support provided by MQ will also continue to be supported (distributed platforms)**
  - Only queue manager associated with Integration Node is global resource.
  - Other MQ QMs involved will be 1PC resources.
- **Global Two-phase Commit (2PC) support provided by RRS will also continue to be supported (z/OS)**
  - All QMs are global resources on zOS

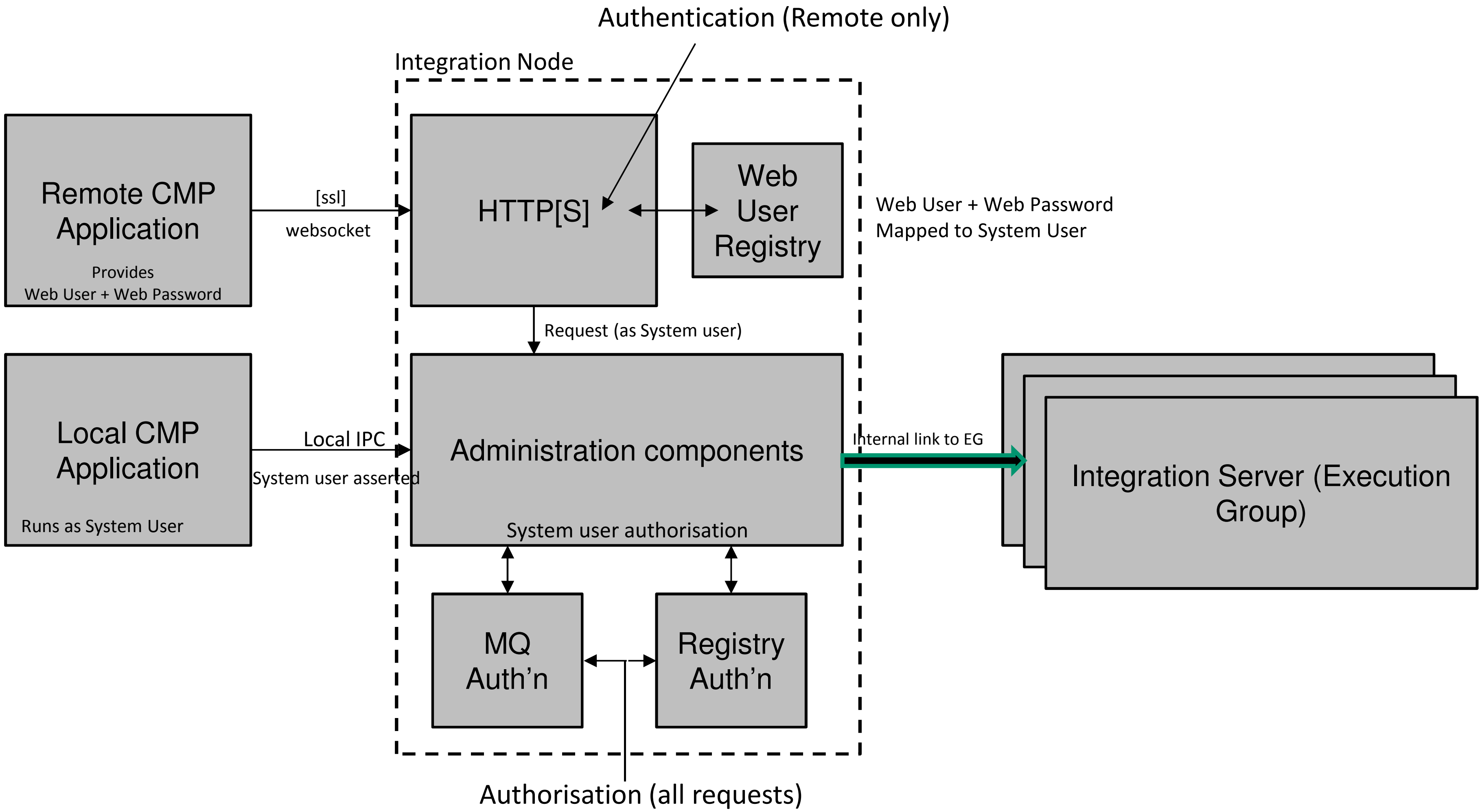
Administration Security

- 2 Methods of connection: Remote (Websocket) and local connections.
- Administrative security currently relies on access control lists held as permissions on MQ queue objects
  - Dependent on local QM to check group memberships, etc
- Will retain MQ authorization, but add a new broker registry-based authorization

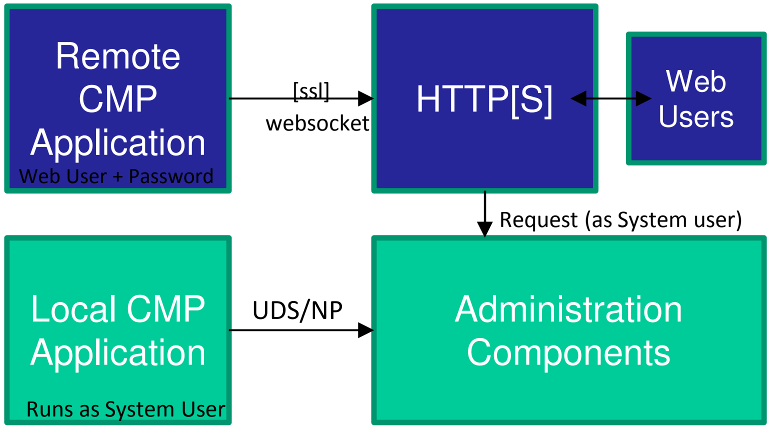
MQ Flow ("Runtime") Security

- MQ nodes can use SSL for Client connections
- Security identity propagation fully supported in new environments.
- Should continue to support MQ AMS for message security

System Diagram



Types of connection:



- **Remote connections will use Web Admin Port**
  - Note: Web Admin port can be disabled
  - Can be secured using user:password (Web Users)
  - Can be secured using SSL (all traffic on port 4414 will share this setting)
  - Client certificate can be used to map identity

- **Local connections will use new local transport (via Named Pipes or Unix Domain Sockets)**
  - Local clients cannot specify a userid. Userid is taken to mean the system account running the source process attempting to connect to the integration node.
  - Local transport will not rely on websocket/http and so will work when port is disabled.
  - System accounts in the IIB node’s primary group (mqbrkrs) will be treated as “superuser” to prevent self-lockout (authorisation).

Remote client authentication

- **Enabling admin security (“mqsichangebroker <iibnode> -s active”) causes user/password security to be required for:**
  - REST
  - Web UI
  - Integration API (remote mode)

- **In all cases, CMP-provided user+password are webuser accounts, and map to a system account. (‘mqsiwebuseradmin’).**

- **Commands which can connect to a remote integration node can now take a URI format:-**
  - tcp://user:password@hostname:port (user and password optional).
  - If user specified without password, user is prompted.

Authorisation

- **Retaining MQ authorisation**
  - SYSTEM.BROKER.AUTH queue, etc
  - IIB will need ALTERNATE_USER_AUTHORITY for this to work
  - Provides simple migration and retains RACF commands for zOS users

- **New file registry-based authorisation**
  - New command to administer access for single integration node.

- **Object structure in both cases unchanged**
  - i.e. READ/WRITE/EXECUTE on a Integration Node / Integration Server / DataCapture resource type.

Registry Authorisation

- **System Resource [Broker / Execution Group / Data Capture]**
  - System User
    - Read Access
    - Write Access
    - Execute Access

```
mqsichangefileauth <broker>
                        -r <role/systemUser>
                        -o <object>
                        -e <server name>
                        -p [read+/-] [write+/-] [execute+/-] [all+/-]
```

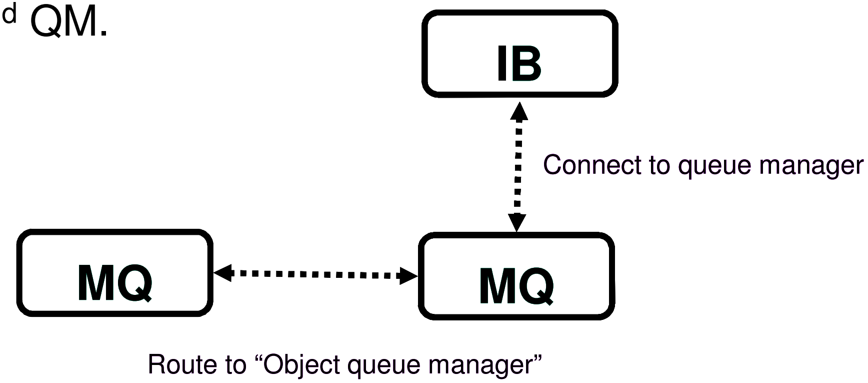
- **Examples:**
  - mqsichangeauth IB10NODE –r pmasters –e default +read
  - mqsichangeauth IB10NODE –r pmasters –o datacapture +all

```
mqsireportfileauth <broker> [-r <role>] [-e <server name>]
```

Other considerations

- **Default codepage for message parsing currently set to the QM CCSID**
  - Will be independent in v10
  - Migrated Integration Nodes will carry the QM CCSID forward into the new configuration

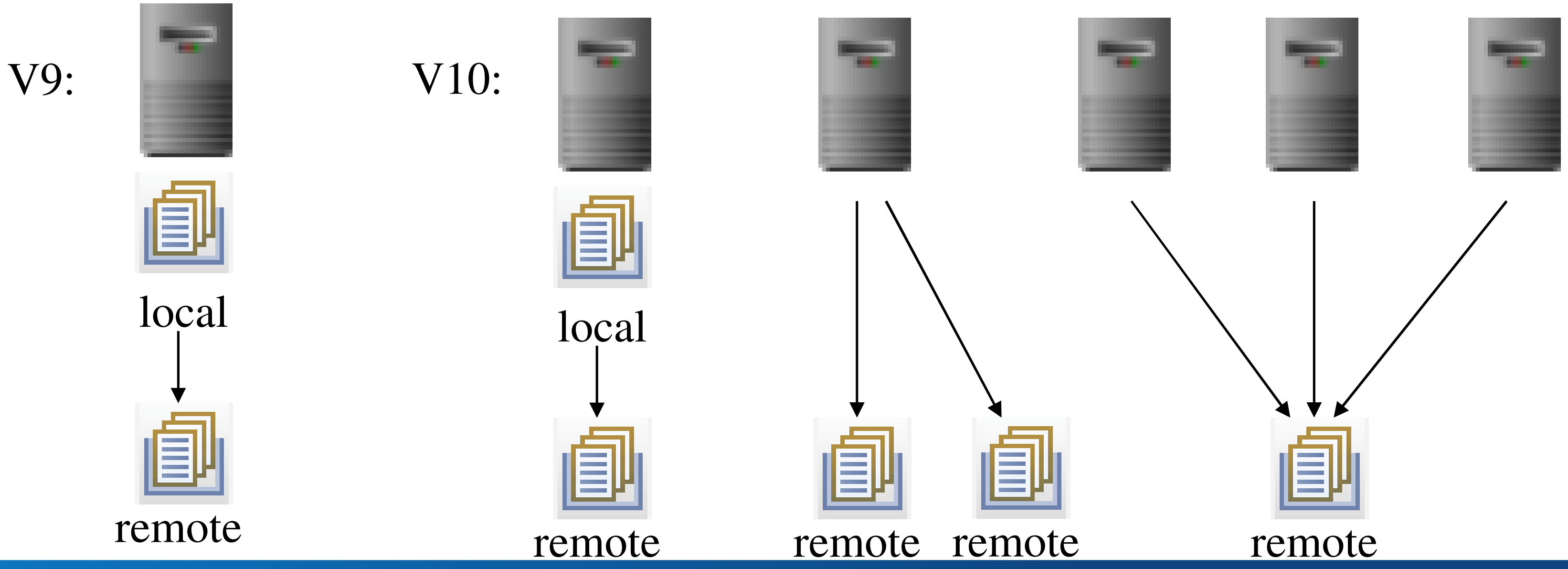
- **MQ Output node already has a “queue manager” property:**
  - In v9, acts as a routing from IB queue manager to target
  - In v10, will route from connected QM to 2nd QM.



Recap: Flexible MQ Topologies

- **V10 introduces flexibility in relationship between Integration Nodes and Queue Managers**
  - Connect to multiple remote QMs
  - Connect to local QM
  - Connect to no QMs

- **This requires a flexible, dynamic definition of these relationships**
  - For this we use policy



IIB Policies

- **Policies define a common approach to performing some action**
  - e.g. health and safety policies, WS-SecurityPolicy
  - A shared, managed definition
  - Policy for **reuse**

MQ Connection

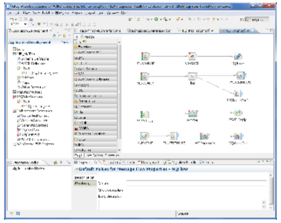
- **Policies separate the description of how something is done from where it is used**
  - Extract properties from code and runtime into objects with their own lifecycle
  - Allow updates to the policy independent of where its used
  - Policy for **visibility** and **dynamicity**

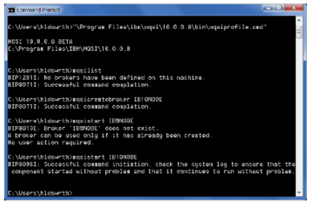
- **IIB Policies aim to provide these capabilities**
  - For Developer: reuse configuration data held in policies in multiple places
  - For Administrator: define key configuration data for each environment in policies
  - For Operator, view and dynamically modify configuration data in policies

- **IIB v9 has Workload Management Policy, applies to a flow**

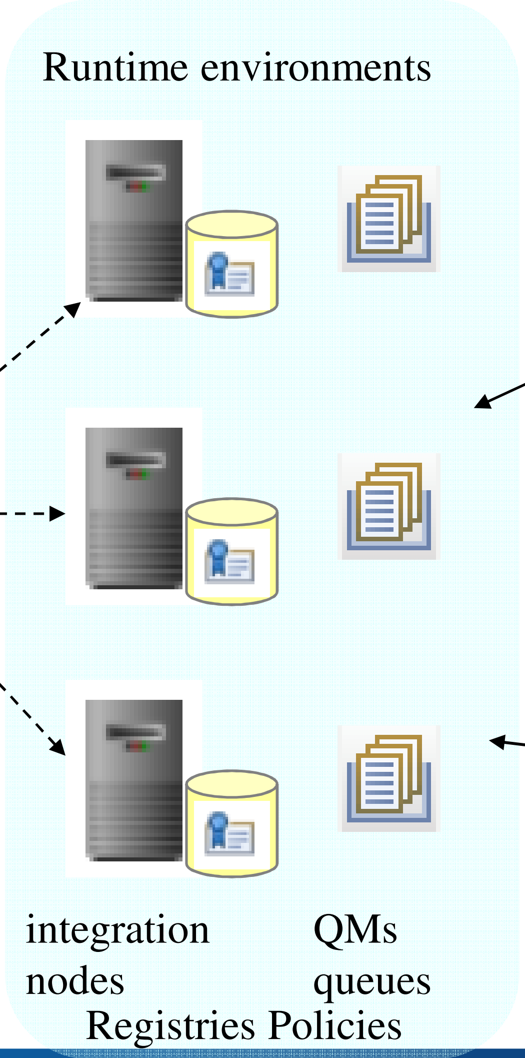
- **For V10 we are designing the general use of policies for nodes, flows and connectors, in particular MQ and MQTT**

IIB Roles for Policy





```
C:\Users\hldswrth>"\Program Files\ibm\mqsi\10.0.0.0\bin\mqsiprofile.cmd"

MQSI 10.0.0.0 BETA
C:\Program Files\IBM\MQSI\10.0.0.0


C:\Users\hldswrth>mqsilist
BIP1281I: No brokers have been defined on this machine.
BIP8071I: Successful command completion.

C:\Users\hldswrth>mqsicreatebroker IB10NODE
BIP8071I: Successful command completion.

C:\Users\hldswrth>mqsistart IB9NODE
BIP8013E: Broker 'IB9NODE' does not exist.
A broker can be used only if it has already been created.
No user action required.

C:\Users\hldswrth>mqsistart IB10NODE
BIP8096I: Successful command initiation. check the system log to ensure that the
 component started without problem and that it continues to run without problem.


C:\Users\hldswrth>
```



Developer

creates

BAR file

Runtime environments

Administrator

installs
configures

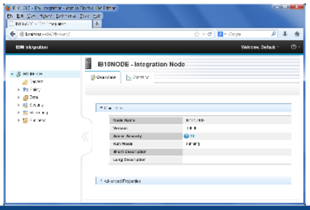
monitors
modifies

Operator

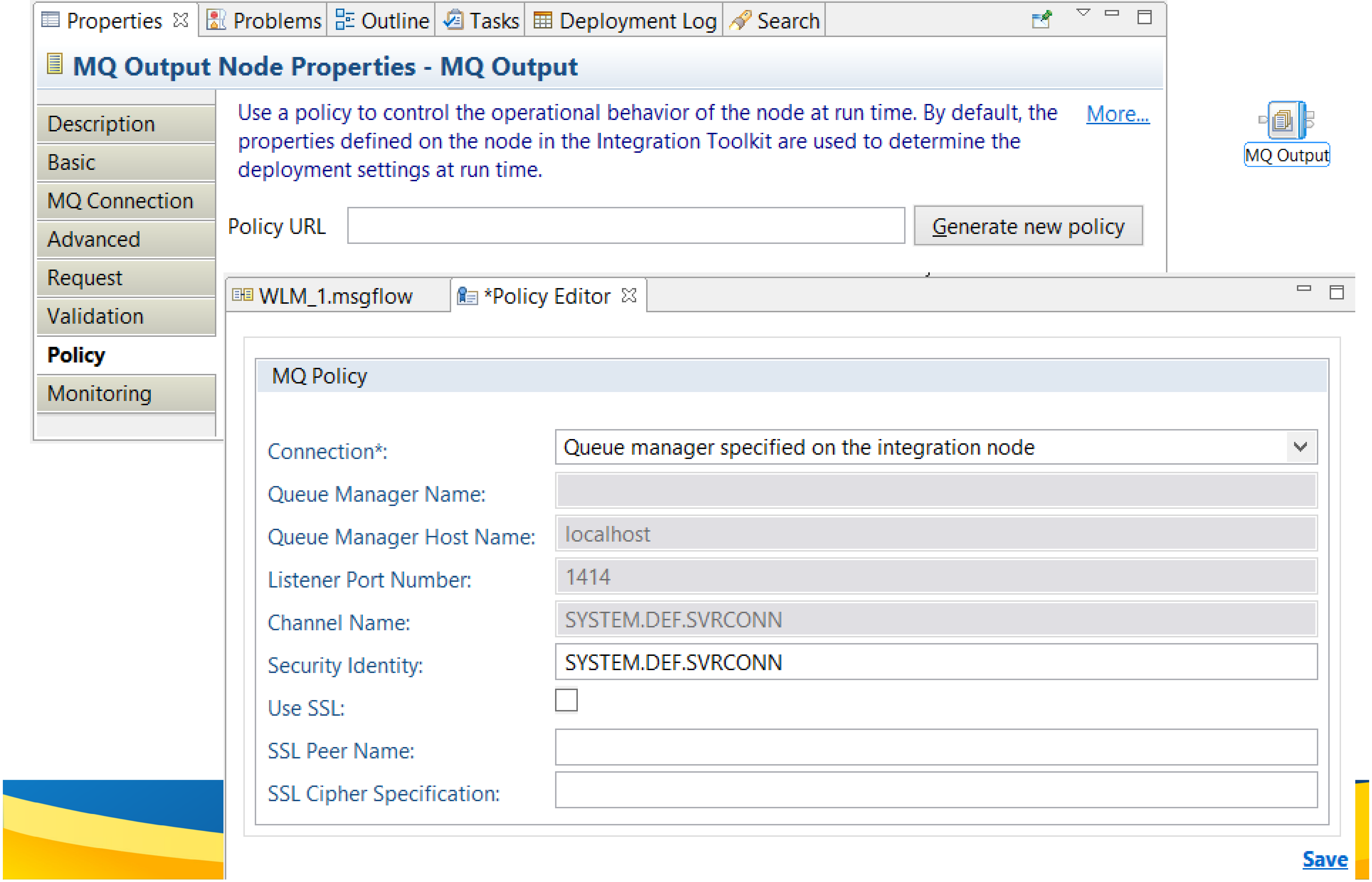
integration
nodes

QMs
queues

Registries Policies



Defining a policy



MQ Output Node Properties - MQ Output

Use a policy to control the operational behavior of the node at run time. By default, the properties defined on the node in the Integration Toolkit are used to determine the deployment settings at run time.

Policy URL:

MQ Policy

Connection*: Queue manager specified on the integration node

Queue Manager Name:

Queue Manager Host Name: localhost

Listener Port Number: 1414

Channel Name: SYSTEM.DEF.SVRCONN

Security Identity: SYSTEM.DEF.SVRCONN

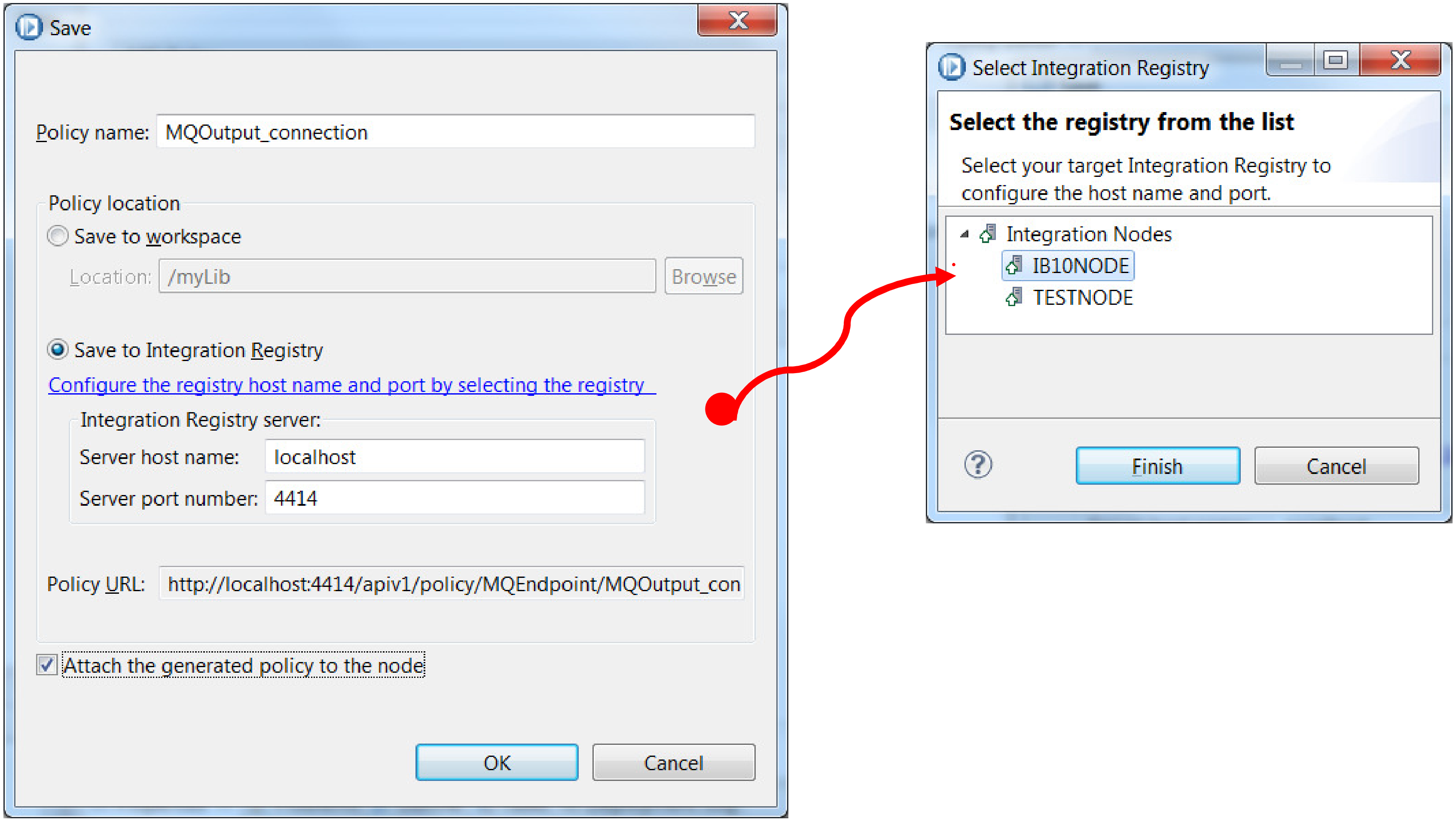
Use SSL: □

SSL Peer Name:

SSL Cipher Specification:

Toolkit: Policy generation wizard

- By default a policy is saved to the workspace with the option to publish to an Integration Node's registry



Save

Policy name: MQOutput_connection

Policy location

○ Save to workspace

Location: /myLib

Browse

● Save to Integration Registry

Configure the registry host name and port by selecting the registry

Integration Registry server:

Server host name: localhost

Server port number: 4414

Policy URL: http://localhost:4414/apiv1/policy/MQEndpoint/MQOutput_con

☑ Attach the generated policy to the node

Select Integration Registry

Select the registry from the list

Select your target Integration Registry to configure the host name and port.

- Integration Nodes
  - IB10NODE
  - TESTNODE

Integration Bus V10 MQ Enhancements

Benefits Summary

Development

- **Simplicity**
  - Single Install for all Integration Tools
    - Fully functional for design & test
  - Administer Integration component only
    - Align with appropriate skills base
  - Maintain Integration component only
    - No extra upgrade dependencies
  - MQTT Publish-Subscribe capability
- **Flexibility**
  - Exploit existing MQ network as-is
  - Direct connect via MQ client libraries
  - Many-to-1 and 1-to-many topologies
    - Maintain existing 1:1 if desired
  - System or MQ based security
  - SSL and CCDT Connections

Operations

- **Availability**
  - Single component for higher reliability
  - Client connection failover
  - Higher availability via multiple MQs
    - Simultaneous MQ connections
    - Single instance has many MQs!
- **MQ for when you need**
  - XA transactions
  - Persistence for state-based features
    - Aggregations, Collections, Sequencing, Timer…

Questions & Answers