

Introducing WebSphere MQ

Elton Stoneman
geekswithblogs.net/eltonstoneman
@EltonStoneman



pluralsight 
hardcore dev and IT training

Introducing WebSphere MQ

The background of the slide features a series of stylized, overlapping human figures in shades of purple and pink, arranged in a line that recedes into the distance. On the left side, there is a large, faint '@' symbol, which is part of the IBM logo.

Enterprise-grade
messaging

Cross-platform
server: Windows,
Unix, Linux, IBM

Supports **complex**
configurations

Goals

WebSphere MQ overview

Feature set and usage

Installation and configuration

Deployment on Windows

.NET client library

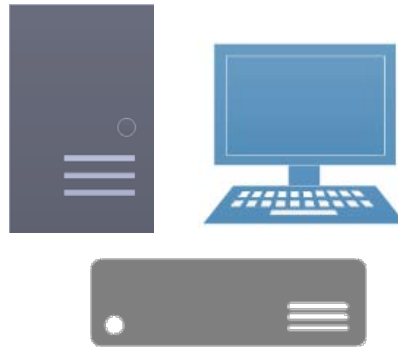
Structure and usage

Pattern support

Fire-and-forget, request-response, publish-subscribe

What is WebSphere MQ?

p



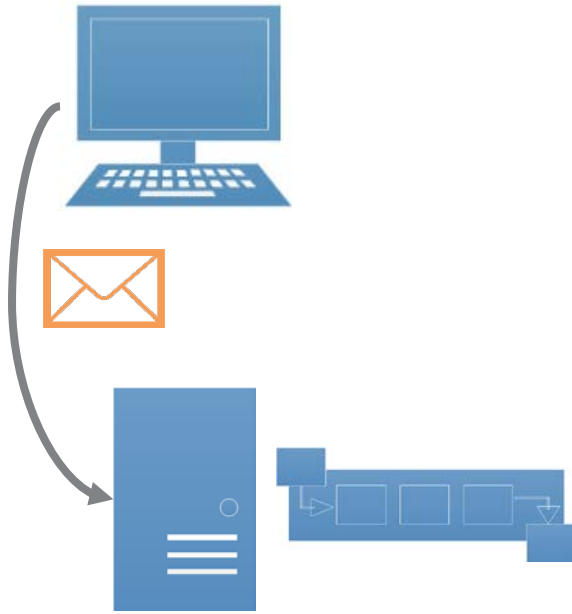
3

On-premise
messaging

Client-server
platform support

Connected queue
technology

How Configurable is WebSphere MQ?

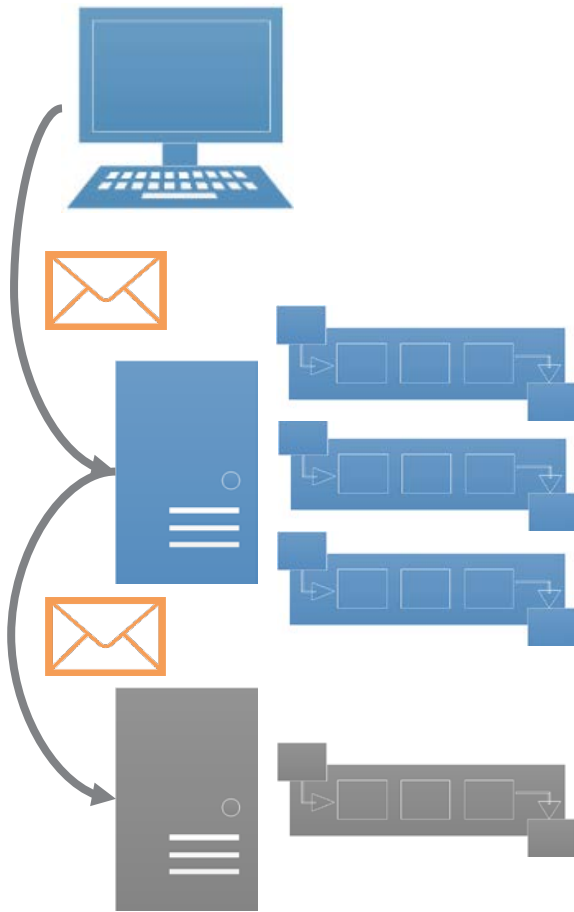


Queue Manager

Hosts queues

Manages client connections

How Configurable is WebSphere MQ?



Queue Manager

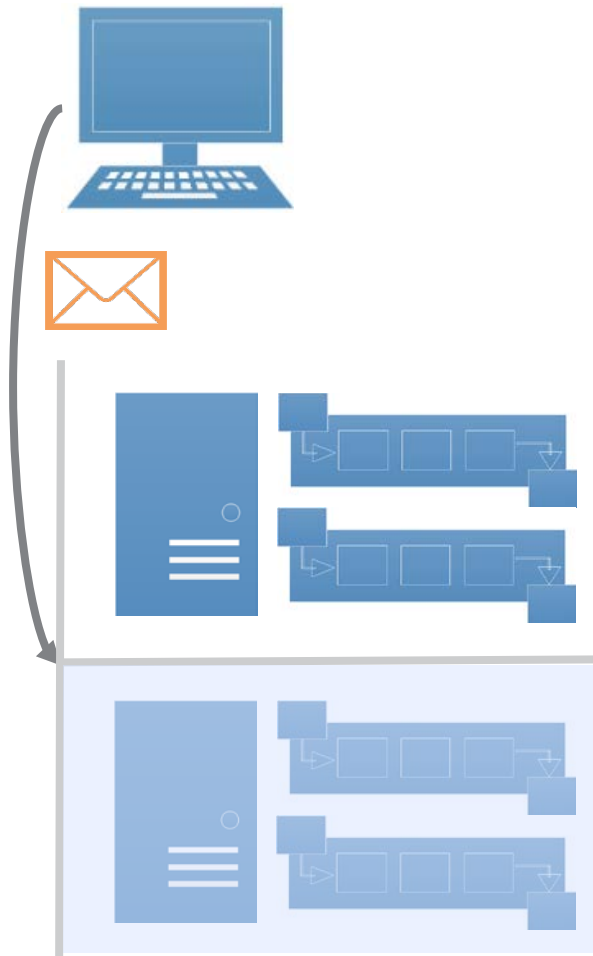
Hosts queues

Manages client connections

Remote queues

Remote Queue Manager

How Configurable is WebSphere MQ?



Queue Manager

Hosts queues

Manages client connections

Remote queues

Remote Queue Manager

Clusters

High availability

Scale

WebSphere MQ Features

Flexibility

- Queue managers, channels, queues
- Storage and transaction options
- Fine-grained security

Enterprise

- Scaling for performance/availability
- Secure remote communication
- File transfer component

WebSphere MQ Features

Practical

- Dynamic response queues
- Optimized receive by correlation ID
- Large messages & segmentation

Administration

- Rich UI portal (Eclipse)
- Honours security model
- Command line (MQSC) & PowerShell

Demo 1: Deploying WebSphere MQ

Feature

Install and
configure
WebSphere MQ

Task

Install WebSphere
MQ version 7.5
on Windows

Task

Use MQ Explorer
to create Queue
Manager, Channel
and Queue

Demo 1: Deploying WebSphere MQ

Demo 1: Deploying WebSphere MQ

- Create Queue Manager

Create Queue Manager

Queue Manager
Enter basic values

Queue manager name: * SC.UNSUB

☐ Make this the default queue manager

Default transmission queue:

Dead-letter queue:

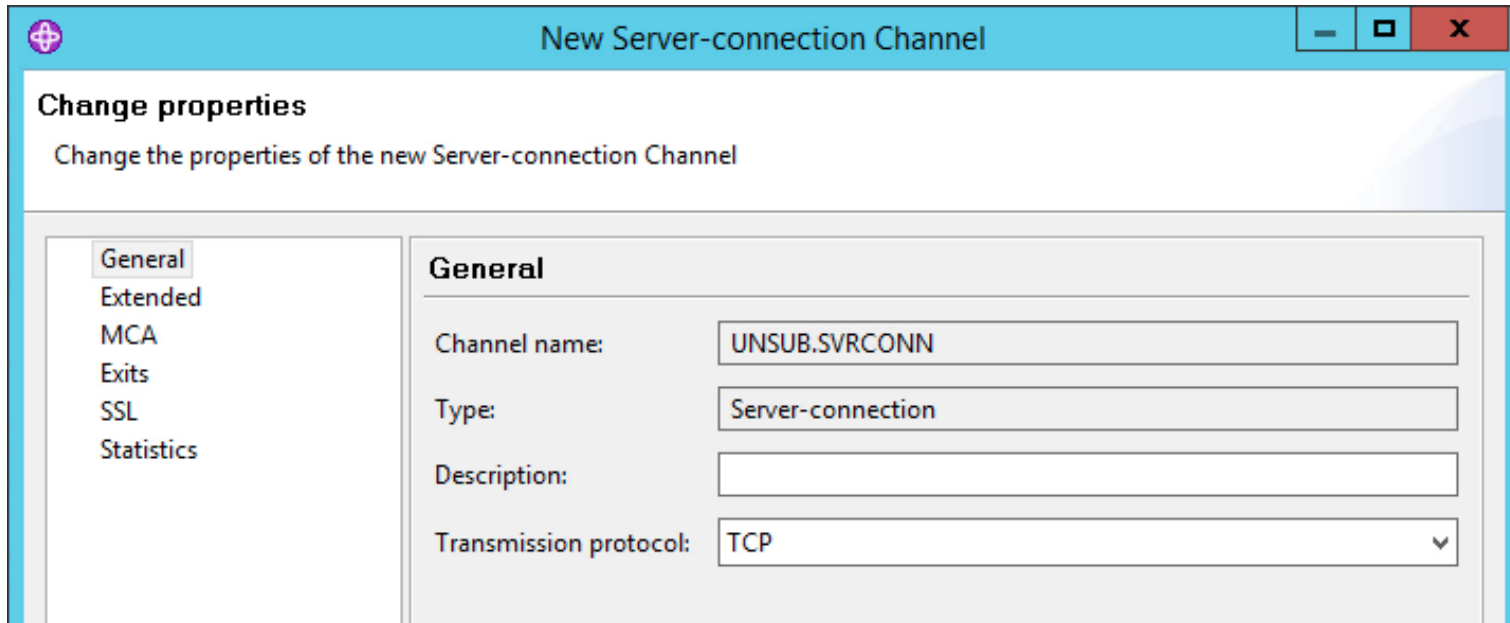
Max handle limit: 256

Trigger interval: 999999999

Max uncommitted messages: 10000

Demo 1: Deploying WebSphere MQ

- Channel Configuration



The screenshot shows a window titled "New Server-connection Channel" with a standard Windows-style title bar (minimize, maximize, close buttons). Below the title bar, the text "Change properties" is displayed, followed by the instruction "Change the properties of the new Server-connection Channel". The main area is divided into two panes. The left pane contains a list of tabs: "General" (selected), "Extended", "MCA", "Exits", "SSL", and "Statistics". The right pane, titled "General", contains four configuration fields: "Channel name:" with the value "UNSUB.SVRCONN", "Type:" with the value "Server-connection", "Description:" with an empty text box, and "Transmission protocol:" with a dropdown menu showing "TCP".

General	
Channel name:	UNSUB.SVRCONN
Type:	Server-connection
Description:	
Transmission protocol:	TCP

Demo 1: Deploying WebSphere MQ

- Channel Configuration

The screenshot shows a 'New Server-connection Channel' dialog box. The 'Extended' tab is selected, displaying the following configuration fields:

Extended	
Maximum message length (bytes):	4194304
Heartbeat interval:	300
Maximum instances:	999999999
Maximum instances per client:	999999999
Message compression:	None

The left sidebar of the dialog box lists the following tabs: General, Extended (selected), MCA, Exits, SSL, and Statistics.

Demo 1: Deploying WebSphere MQ

- Queue Configuration

New Local Queue

Change properties
Change the properties of the new Local Queue

General

Queue name: test

Queue type: Local

Description:

Put messages: Allowed

Get messages: Allowed

Default priority: 0

Default persistence: Not persistent

Demo 1: Deploying WebSphere MQ

- Queue Configuration

The screenshot shows the 'New Local Queue' configuration window. The window has a title bar with a plus icon and standard minimize, maximize, and close buttons. Below the title bar, there's a section titled 'Change properties' with the subtitle 'Change the properties of the new Local Queue'. On the left side, there's a vertical list of tabs: General, Extended (selected), Cluster, Triggering, Events, Storage, and Statistics. The main area is divided into two panes. The left pane is titled 'Extended' and contains a list of properties: Max queue depth, Maximum message length (bytes), Shareability, and Default input open option. The right pane shows the values for these properties: 5000, 4194304, Shareable, and Input shared. At the bottom, there's a 'Default persistence' dropdown menu set to 'Not persistent'.

Property	Value
Max queue depth:	5000
Maximum message length (bytes):	4194304
Shareability:	Shareable
Default input open option:	Input shared
Default persistence:	Not persistent

Client Libraries

C

Client libraries for
Java and .NET (etc.)



JMS-style API for
Java and .NET



Native API - **feature
parity** in .NET and Java

MQQueueManager



Queue Manager connection

Specified channel

Connection properties

Administration

Change configuration

Create queues

Indirectly via model queues

MQQueue



Messaging operations

Created from `MQQueueManager`

Context-bound to one queue

MQMessage

Define content format

Write body directly

Send and receive

Persistence options

Transactional options

Demo 2: Using WebSphere MQ with .NET

Feature

Explore
messaging with
WebSphere MQ
and .NET

Task

Send and receive
string messages
with MQQueue

Task

Check basic
performance for
different
reliability

Demo 2: Using WebSphere MQ with .NET

Demo 2: Using WebSphere MQ with .NET

- Send messages

- Build connection properties for Queue Manager

```
var properties = new Hashtable();  
properties.Add(MQC.HOST_NAME_PROPERTY, "127.0.0.1");  
properties.Add(MQC.CHANNEL_PROPERTY, "UNSUB.SVRCONN");
```

- Connect to Queue Manager & Queue
 - Specifying MQOO_OUTPUT option

```
var queueManager = new MQQueueManager("SC.UNSUB", properties);  
var queue = queueManager.AccessQueue("test", MQC.MQOO_OUTPUT);
```

Demo 2: Using WebSphere MQ with .NET

- Send messages

- Build message

```
var message = new MQMessage();  
message.Format = MQC.MQFMT_STRING;  
message.WriteString("message");
```

- Send

```
queue.Put(message);
```

Demo 2: Using WebSphere MQ with .NET

- **Receive messages**

- Build connection properties for Queue Manager

```
var properties = new Hashtable();  
properties.Add(MQC.HOST_NAME_PROPERTY, "127.0.0.1");  
properties.Add(MQC.CHANNEL_PROPERTY, "UNSUB.SVRCONN");
```

- Connect to Queue Manager & Queue
 - Specifying MQOO_INPUT* option

```
var queueManager = new MQQueueManager("SC.UNSUB", properties);  
var queue = queueManager.AccessQueue("test", MQC.MQOO_INPUT_AS_Q_DEF);
```


Demo 2: Using WebSphere MQ with .NET

- Receive messages

- Build message

```
var message = new MQMessage();  
message.Format = MQC.MQFMT_STRING;  
message.WriteString("message");
```

- Receive

```
queue.Get(message);
```

- Read body

```
message.ReadString(message.MessageLength).Dump("Body");
```

Demo 2: Using WebSphere MQ with .NET

- Transactional messaging

- Specify SYNCPOINT option on Put

```
var options = new MQPutMessageOptions()  
{  
    Options = MQC.MQPMO_SYNCPOINT  
};  
queue.Put(message, options);
```

- Commit

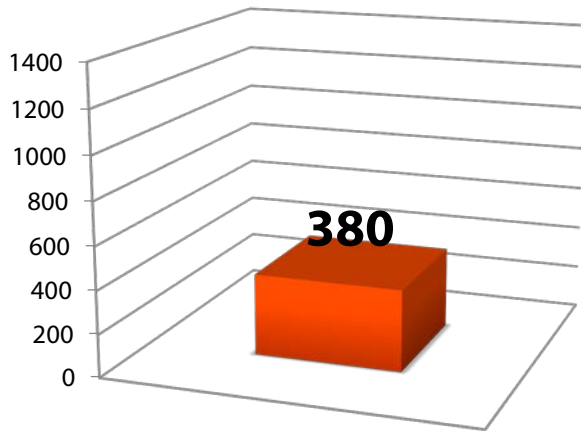
```
queueManager.Commit();
```

Demo 2: Using WebSphere MQ with .NET

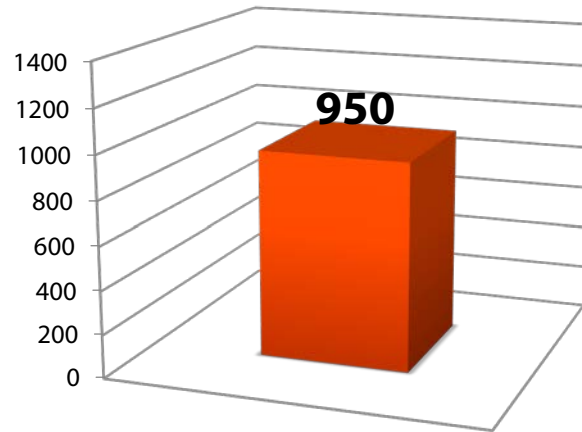
- **Durable messaging**
 - Specify Persistence option on message

```
var message = new MQMessage();  
message.Persistence = MQC.MQPER_NOT_PERSISTENT; //0;  
message.Persistence = MQC.MQPER_PERSISTENT; //1  
message.Persistence = MQC.MQPER_PERSISTENCE_AS_Q_DEF; //2;
```

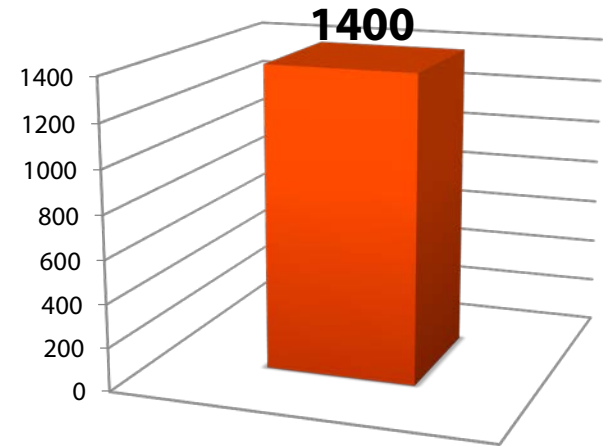
Demo 2: Using WebSphere MQ with .NET




Queue default
non-persistent



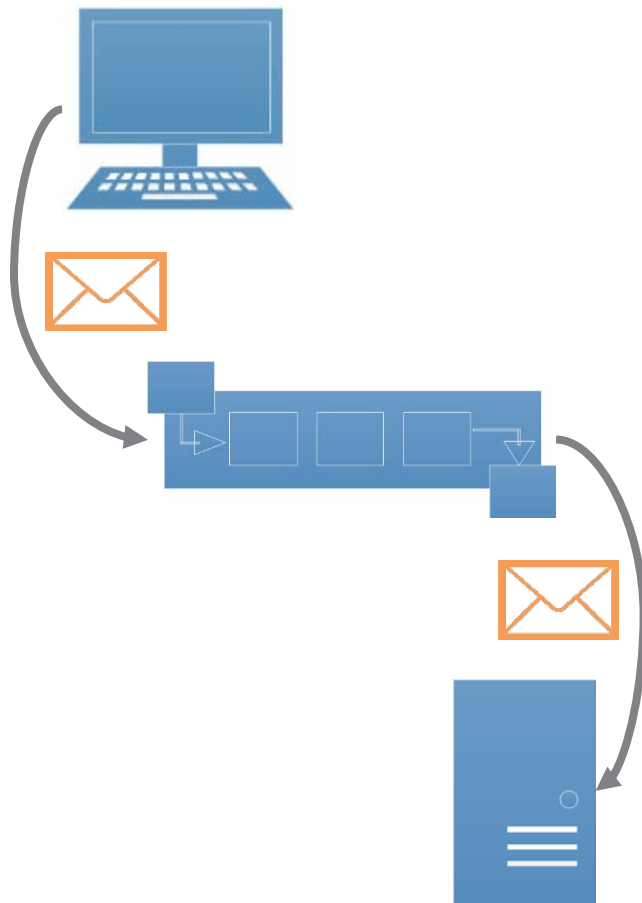
Persistent
messages



Syncpoint
queue access

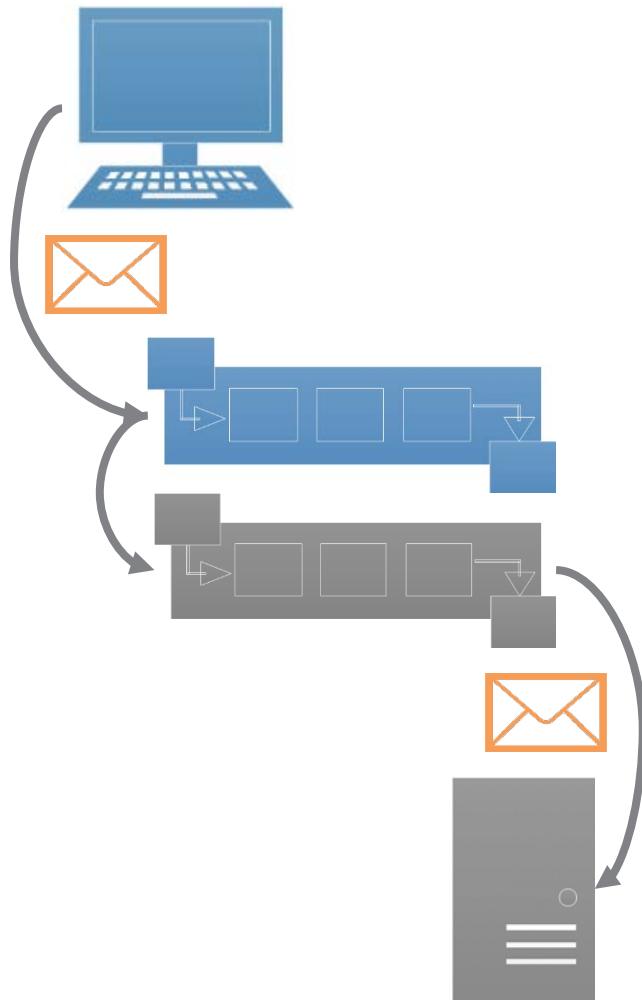
 - milliseconds to send 1,000 messages

Messaging Pattern Support



Fire-and-forget
Standard support

Messaging Pattern Support

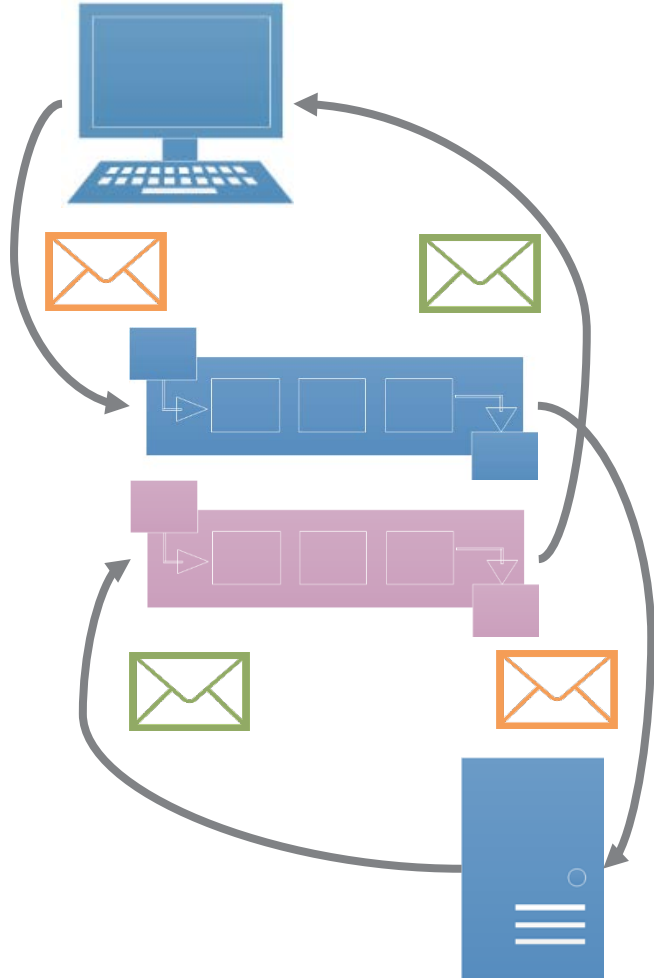


Fire-and-forget

Standard support

B2B with remote queues

Messaging Pattern Support



Fire-and-forget

Standard support

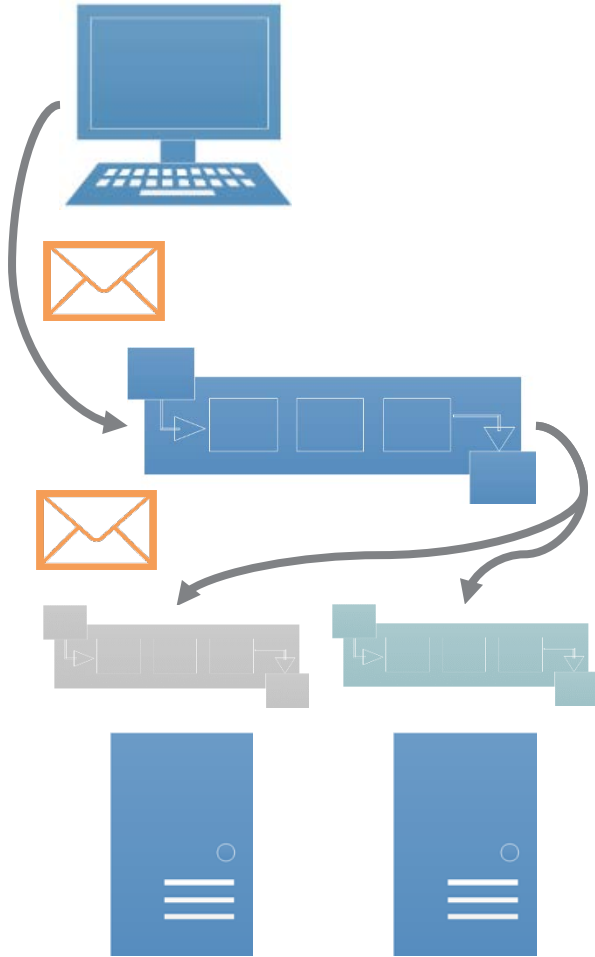
B2B with remote queues

Request-response

Shared response queue

Dynamic temporary response queue

Messaging Pattern Support



Fire-and-forget

Standard support

B2B with remote queues

Request-response




Shared response queue

Dynamic temporary response queue

Publish-subscribe

Topics coupled to queues

Summary

- **Introducing WebSphere MQ** 
 - Enterprise-grade messaging
 - Cross-platform client-server
- **Components** 
 - Queue Manager & channel
 - Queue – local, alias, model, remote
 - And lots more options
- **Usage** 
 - MQ Explorer
 - .NET client library
 - Performance



WebSphere MQ