

Sending and Receiving Messages with RabbitMQ



Roland Guijt

INDEPENDENT SOFTWARE DEVELOPER AND TRAINER

@rolandguijt www.rmgsolutions.nl



Module Overview



What is RabbitMQ?

Installing RabbitMQ

Terminology

Exchange types

Sending messages

Receiving messages

The management plugin

Reliability options

Virtual hosts and security





Message broker

Erlang

Open source

Commercially supported

Standardized protocols

Plugins

Many clients

Clustering



Installing

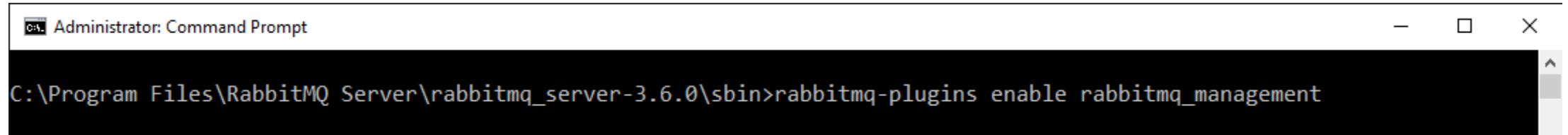
Support for various OS, cloud, and container platforms

Windows version has installer

Management plugin



Installing the Management Plugin



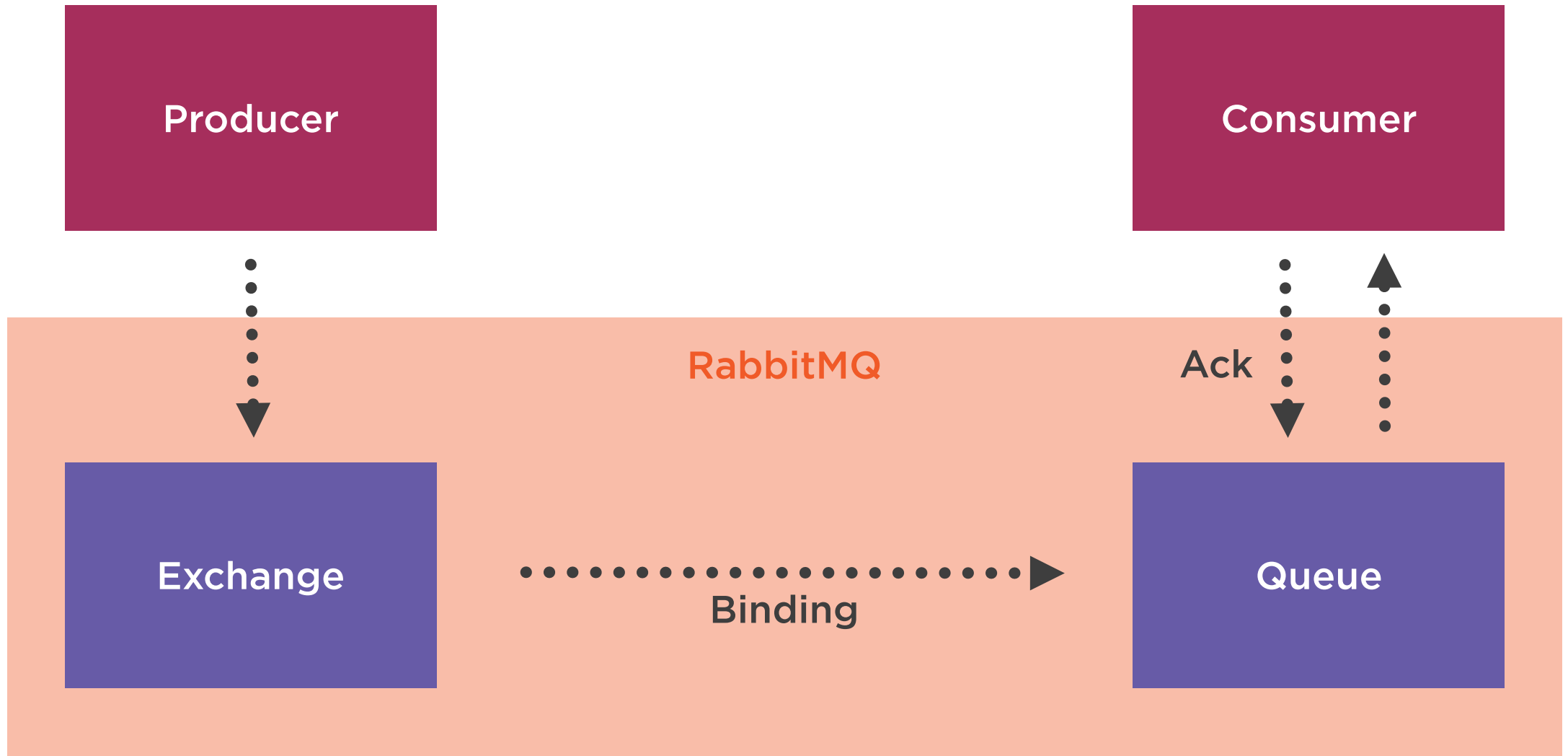
```
Administrator: Command Prompt
C:\Program Files\RabbitMQ Server\rabbitmq_server-3.6.0\sbin>rabbitmq-plugins enable rabbitmq_management
```



RabbitMQ Terminology



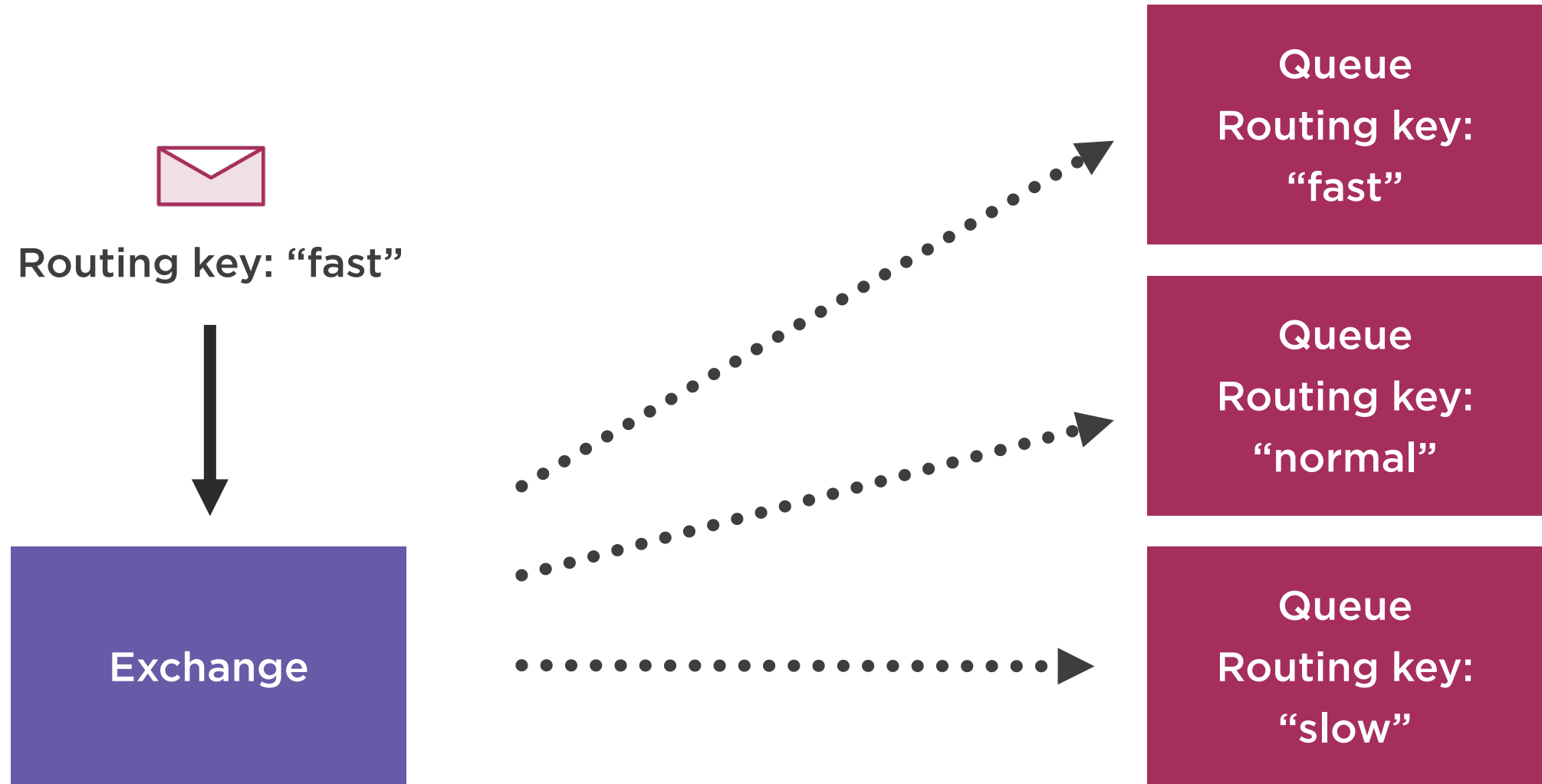
RabbitMQ Terminology



Exchange Types: Direct



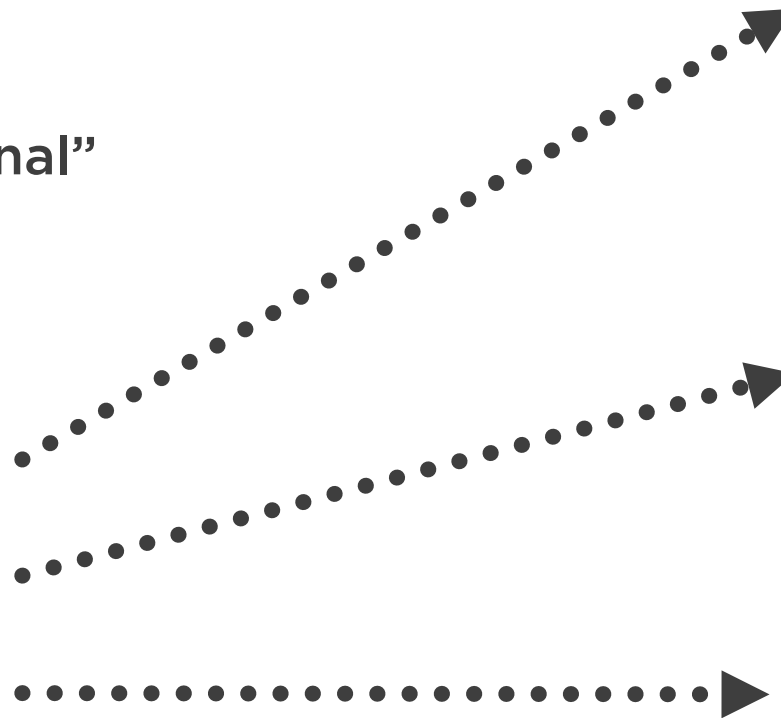
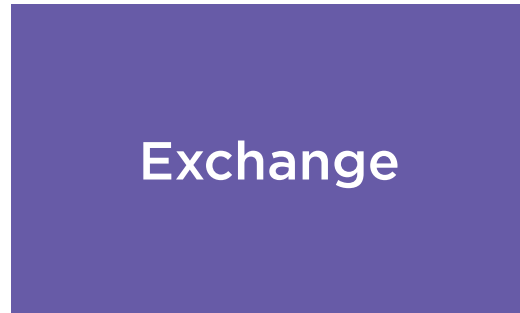
Routing Keys



Exchange Types: Topic



Routing key: "package.fast.national"



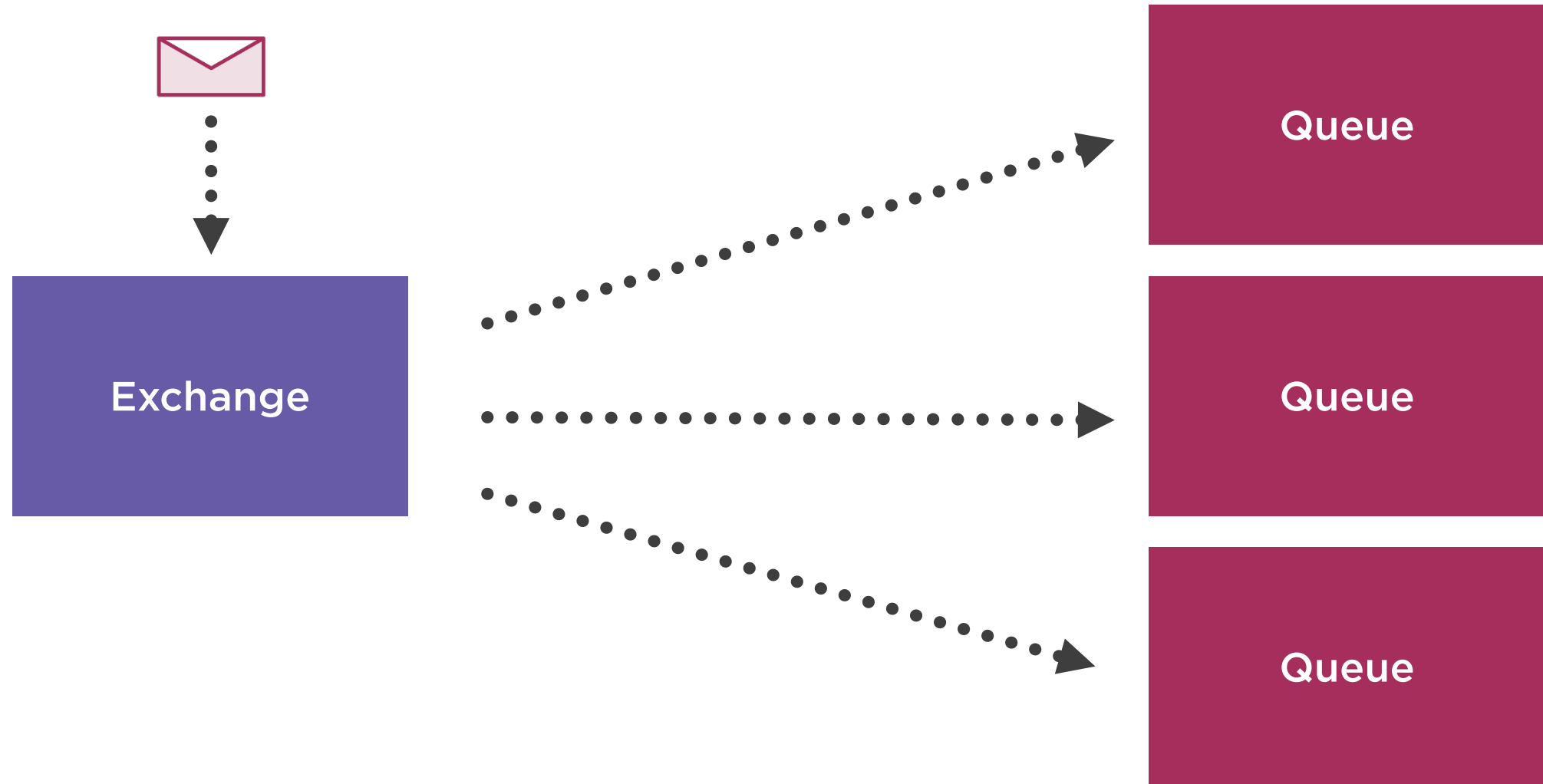
Queue
Routing key:
"package.*.international"

Queue
Routing key:
"#.national"

Queue
Routing key:
"package.fast.national"



Exchange Types: Fanout



```
var messageProperties =  
channel.CreateBasicProperties();  
  
messageProperties.CorrelationId =  
Guid.NewGuid().ToString();
```

◀ Correlating a message

- Process identifier
- Groups messages belonging to the same process
- Typically stored in the service database
- Can also be done using message property



Reliability Options

Acks

Publisher confirms

Mandatory

Reply to sender

**Connection and
topology recovery**



Acks

Consumer tells RabbitMQ to delete the message from the queue

NoAck bool when registering consumer

Unacked messages are requeued

Redelivered flag

Implement redelivered strategy yourself

Dead letter exchange



Publisher Confirms

Producer gets an acknowledgement when message is queued

Possible responses: Ack and Nack

Implement re-send strategy yourself



Publisher Confirms in Code

```
channel.ConfirmSelect();
```

```
channel.BasicAcks += (o, args) => HandleAck(args);
```

```
channel.BasicNacks += (o, args) => HandleNack(args);
```



Mandatory

Flag to set when doing a BasicPublish

Ensures that the message can be routed to a queue

If not the message is sent back to the producer



Mandatory in Code

```
channel.BasicPublish(exchange: "exchange", routingKey: "",  
    basicProperties: properties, body: body, mandatory: true);
```

```
channel.BasicReturn += (o, args) => HandleReturn(args);
```



Reply to Sender

When the producer has to be certain a message arrived at the consumer

ReplyTo in message properties

Implement yourself or use
SimpleRpcServer and SimpleRpcClient



Connection and Topology Recovery

Automatically recover when the network connection fails

Reconnect, re-open channels, restore channel listeners

Re-declare exchanges and queues, recover all bindings and consumers



Recovery in Code

```
ConnectionFactory factory = new ConnectionFactory();  
factory.AutomaticRecoveryEnabled = true;  
factory.TopologyRecoveryEnabled = true; //default  
factory.NetworkRecoveryInterval = TimeSpan.FromSeconds(10);  
IConnection conn = factory.CreateConnection();
```



Virtual Hosts

Logical container

Separation

Security

System exchanges/queues

Connect using URI extension



Security

Users

Globally

Configure, read, write

Regular expressions

Example: `^service.*`

Guest



Summary



Purpose of RabbitMQ

Terminology

Exchange types

.NET Client API

Management Plugin

Correlation

Reliability

Virtual Hosts

Security

