RabbitMQ for .Net Developers – Part 2

Messages & Data

Michael Stephenson
@michael_stephen
Michael_stephensonuk@yahoo.co.uk

pluralsight
hardcore developer training

Agenda

- Serialization & Real Data
- Message Format & Serialization
- Messages & Message Identification

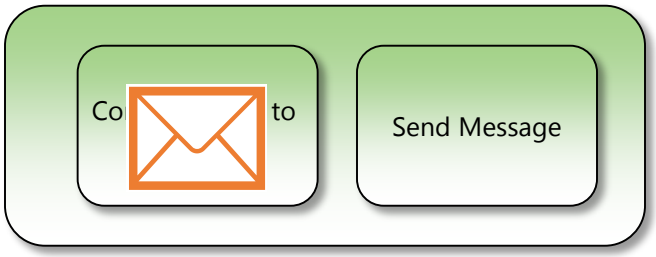
Serialization & Real Data

What & Why

- **Real world != “Hello World”**
- **Real World**
  - Applications work with objects
  - Messaging systems work with messages
  - Objects are converted to messages
  - Messages are converted back to objects

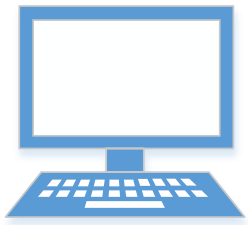
Object → byte array = Serialization
Byte array → object = Deserialization

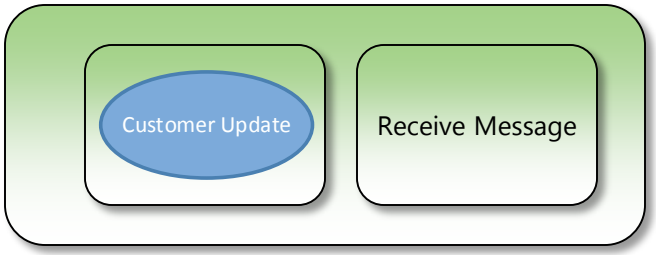
Application Message Processing



Sender Application

Queue





Receiving Application

Demo

Serialization & Real Data

Message Format & Serialization

What & Why

- **Types of serialization**
  - XML
  - JSON
  - Binary

- **How do I communicate what the format of the message body is?**

ContentType property!!!

Application Message Processing

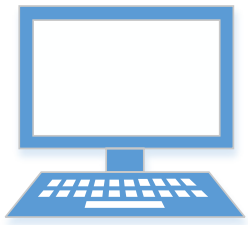
Con to

Indicate the format Of message

Send Message

Sender Application

Queue



Workout the Message format

Receive Message

Receiving Application

Demo

Message Format & Serialization

Messages & Message Identification

What & Why

- **Real world**
  - Many different types of messages
  - Data structures can use inheritance or polymorphism
  - Messages aren't always sent by the same technology as processes it

- **To process a message the receiver needs to workout what the message is**
  - From the content?
  - The sender indicates the message type?

BasicProperties has a Type property intended for the message type

Application Message Processing

Co to

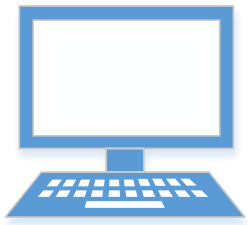
Indicate the type of message

Indicate the format Of message

Send Message

Sender Application

Queue



Workout the Message type

Workout the Message format

Receive Message

Receiving Application

How Can We Do It

- **Message Type = BasicProperties.Type**
  - Custom String?
  - .net Fully Qualified Name
  - XSD namespace + Root Element

- **Considerations**
  - Should be application agnostic (if you want Interoperability)
  - What about versioning?

Demo – Shared .net Type

Messages & Messages Identification

Demo – Interoperable

Messages & Messages Identification

Summary

- **Messaging Challenges**
  - Serialization & Real Data
  - Message Format & Serialization
  - Messages & Message Identification

This is one of the keys to interoperability