Message-based architecture with RabbitMQ

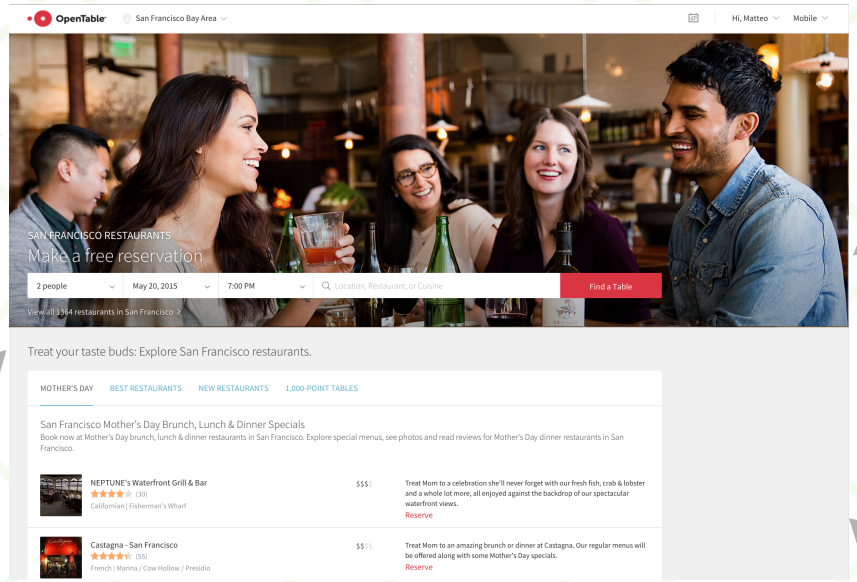
Łukasz Łysik

www.4developers.org.pl

@lukasz_lysik

Developers Festival

OpenTable™

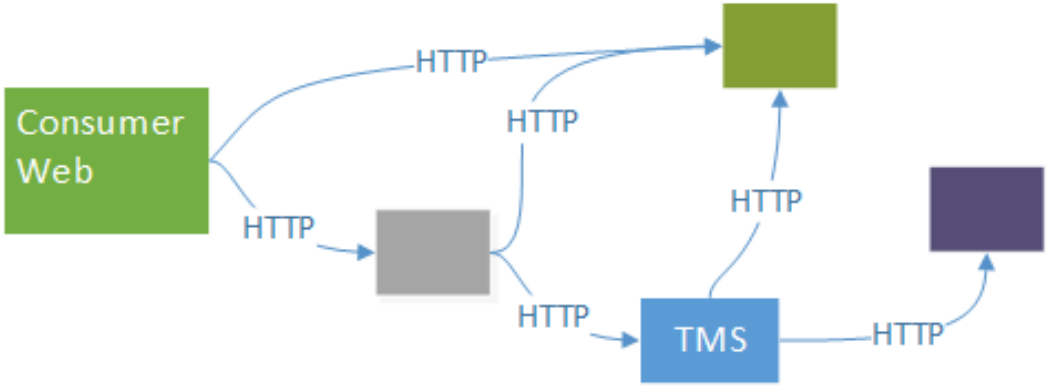


Developers Festival

Transactional Messaging System (TMS)

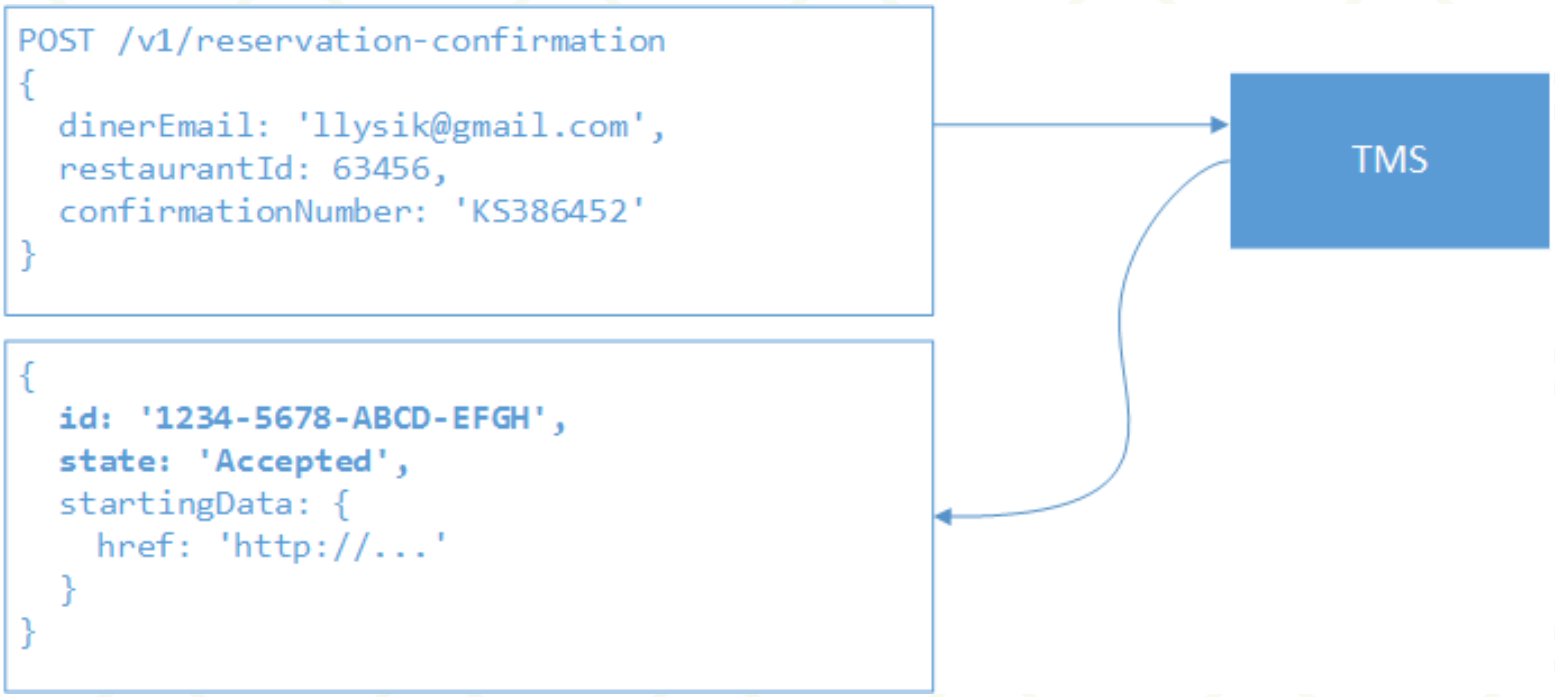
Developers Festival

Transactional Messaging System (TMS)

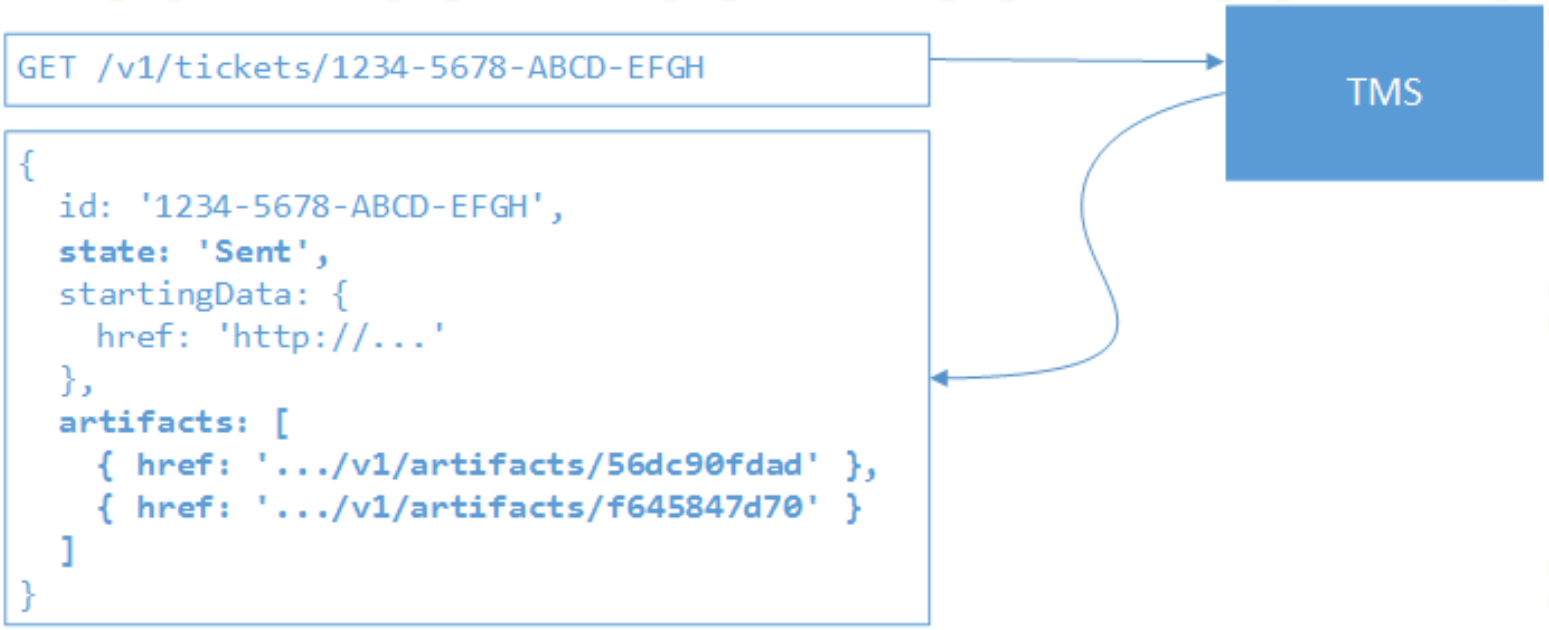


Developers Festival

TMS – Creating a ticket

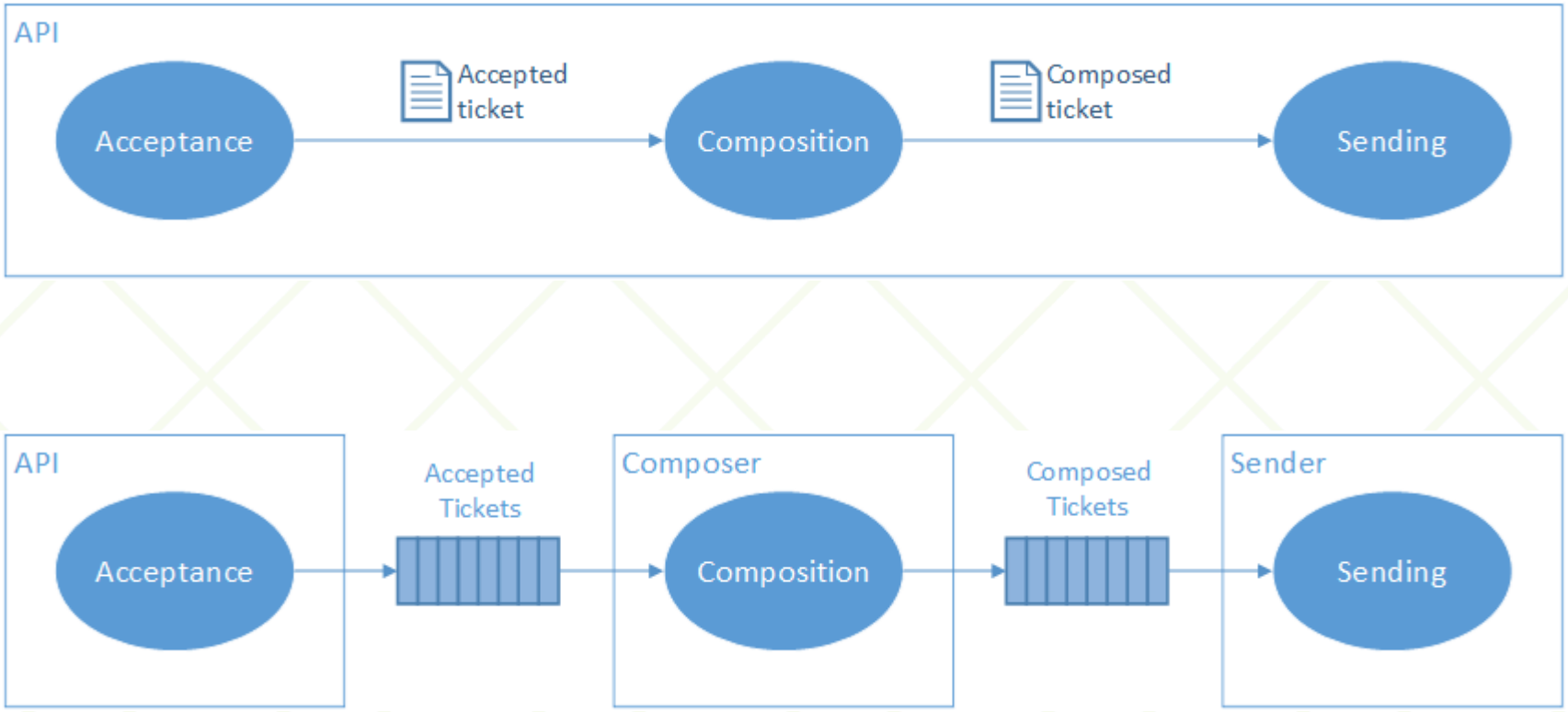


TMS – Getting ticket state



Developers Festival

Transactional Messaging System (TMS)



Developers Festival

RabbitMQ

Developers Festival

RabbitMQ

- Message broker
- Open Source
- Implements AMQP standard (Advanced Messaging Queueing Protocol)
- Written in Erlang

Developers Festival

RabbitMQ Simulator

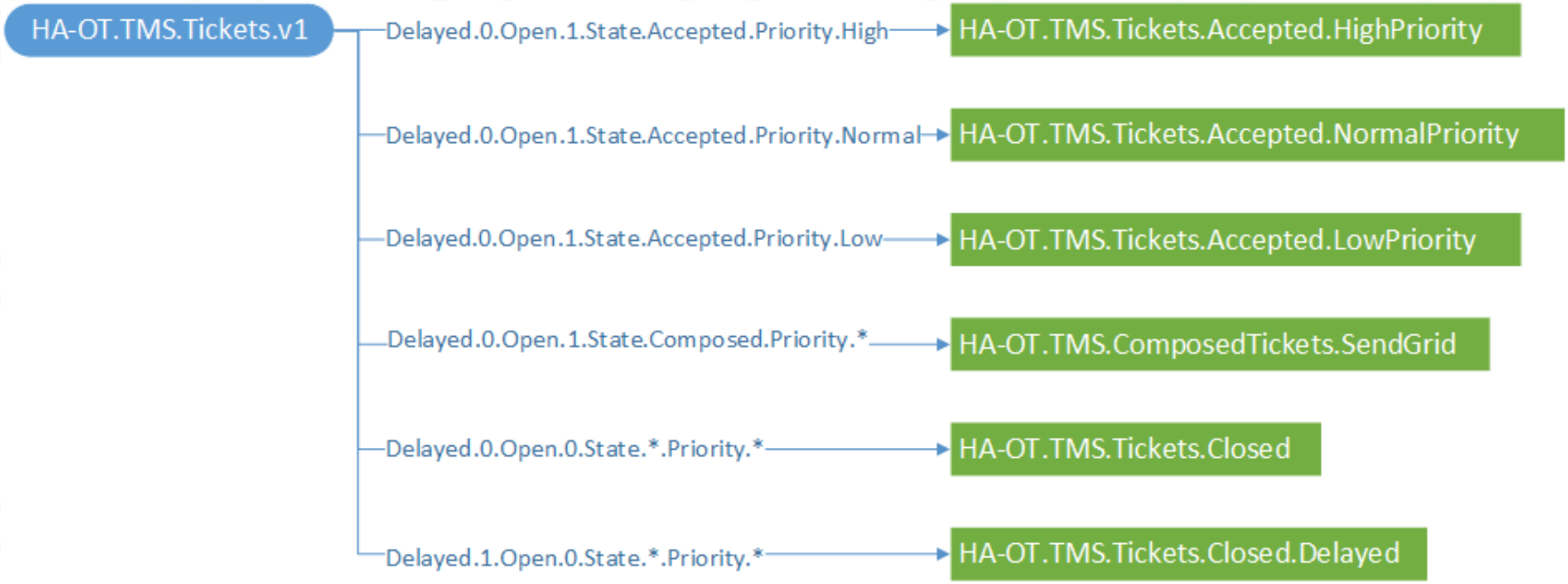
Demo

http://tryrabbitmq.com/

Developers Festival

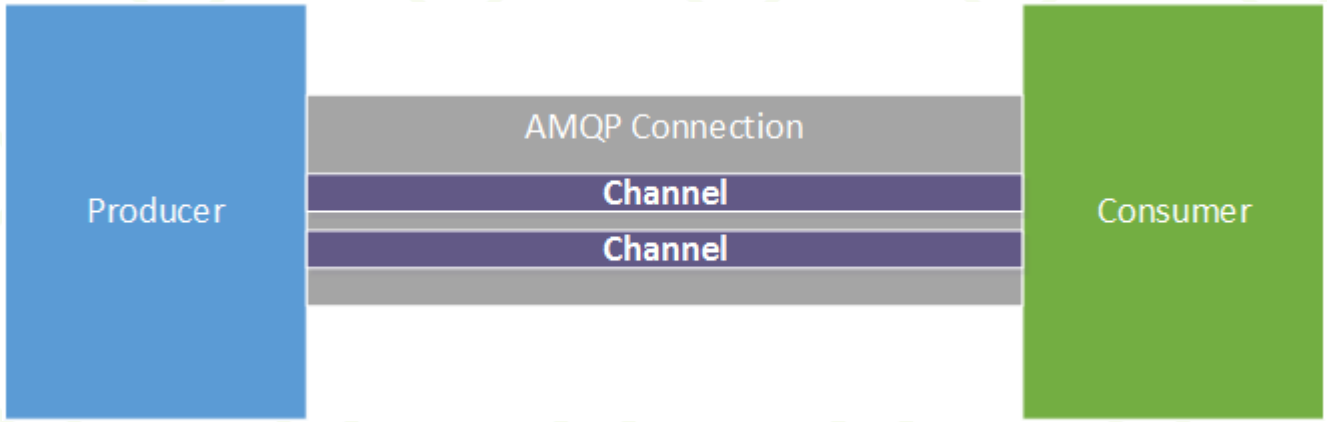
Queues in TMS

Routing key: Delayed.0.Open.1.State.Composed.Priority.Low



Developers Festival

AMQP Connection




```
var factory = new ConnectionFactory { HostName = "localhost" };
var connection = factory.CreateConnection();  // Thread safe
var channel = connection.CreateModel();       // Thread unsafe
```

Developers Festival

Exchanges and queues declarations

```
channel.ExchangeDeclare(
                exchange: "logs",
                type: "fanout");
```

```
channel.QueueDeclare(
                queue: "hello",
                durable: false,
                exclusive: false,
                autoDelete: false,
                arguments: null);
```

```
channel.QueueBind(
                queue: "hello",
                exchange: "logs",
                routingKey: "");
```

Developers Festival

Sending a message

```
var message = "Hello World!";
var body = Encoding.UTF8.GetBytes(message);
```

```
var properties = channel.CreateBasicProperties();
properties.SetPersistent(true);
properties.ContentType = "application/vnd.ticket_changed.v1+json";
properties.Headers = new Dictionary<string, object>();
properties.Headers.Add("anything", "here");
```

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

Developers Festival

Receiving a message

```
var consumer = new QueueingBasicConsumer(channel);
channel.BasicConsume(queue: "hello", noAck: true, consumer: consumer);
```

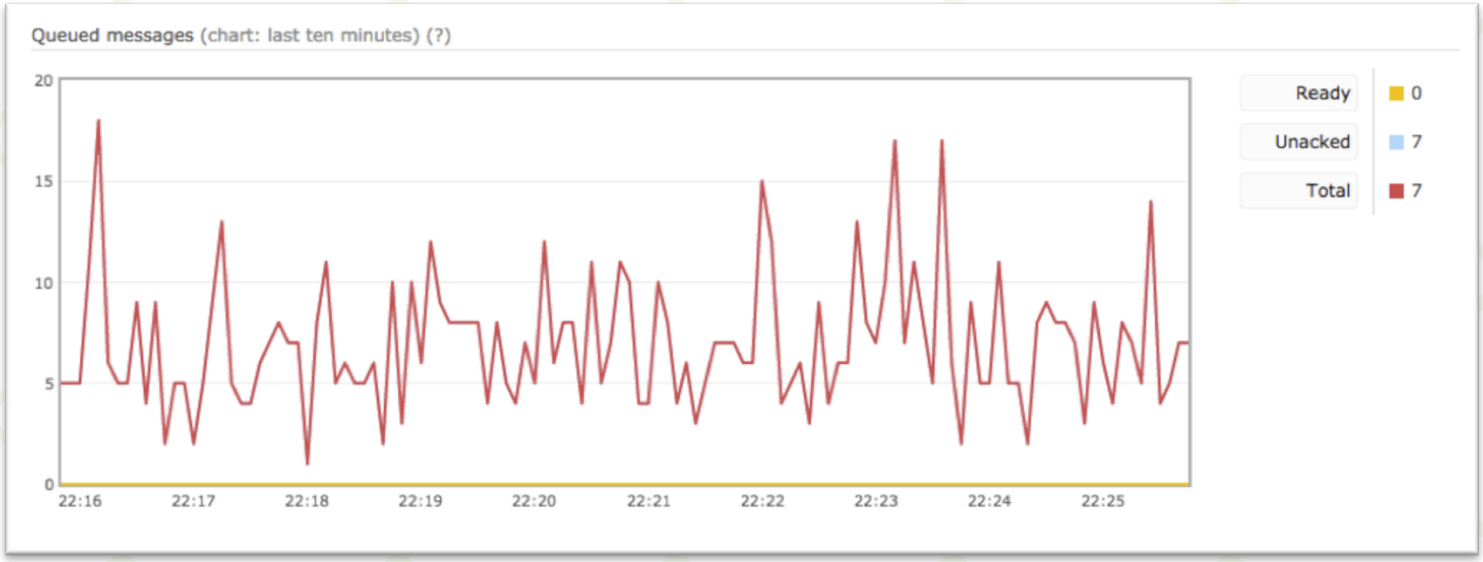
```
while (true)
{
        var eventArgs = consumer.Queue.Dequeue();
        var message = Encoding.UTF8.GetString(eventArgs.Body);
        Console.WriteLine("Received: {0}", message);
}
```

```
// eventArgs.BasicProperties
// eventArgs.RoutingKey
```

Receiving a message – manual ack

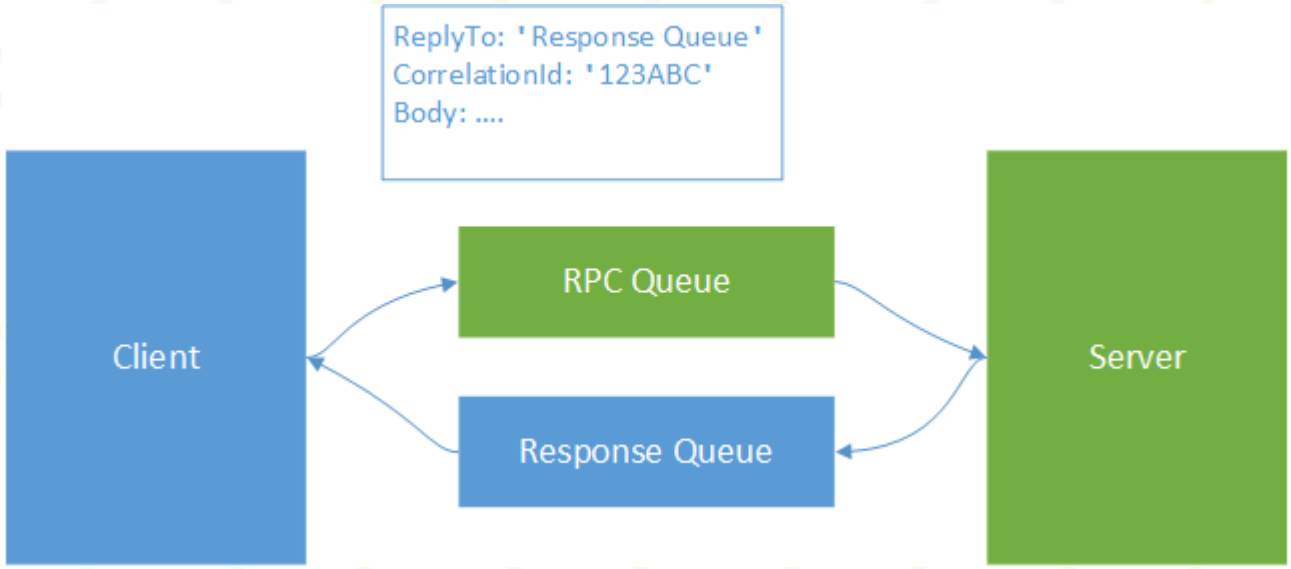
```
// noAck - "no manual acks"
channel.BasicConsume(queue: "hello", noAck: false, consumer: consumer);
```

```
// Process message
channel.BasicAck(eventArgs.DeliveryTag, multiple: false);
```



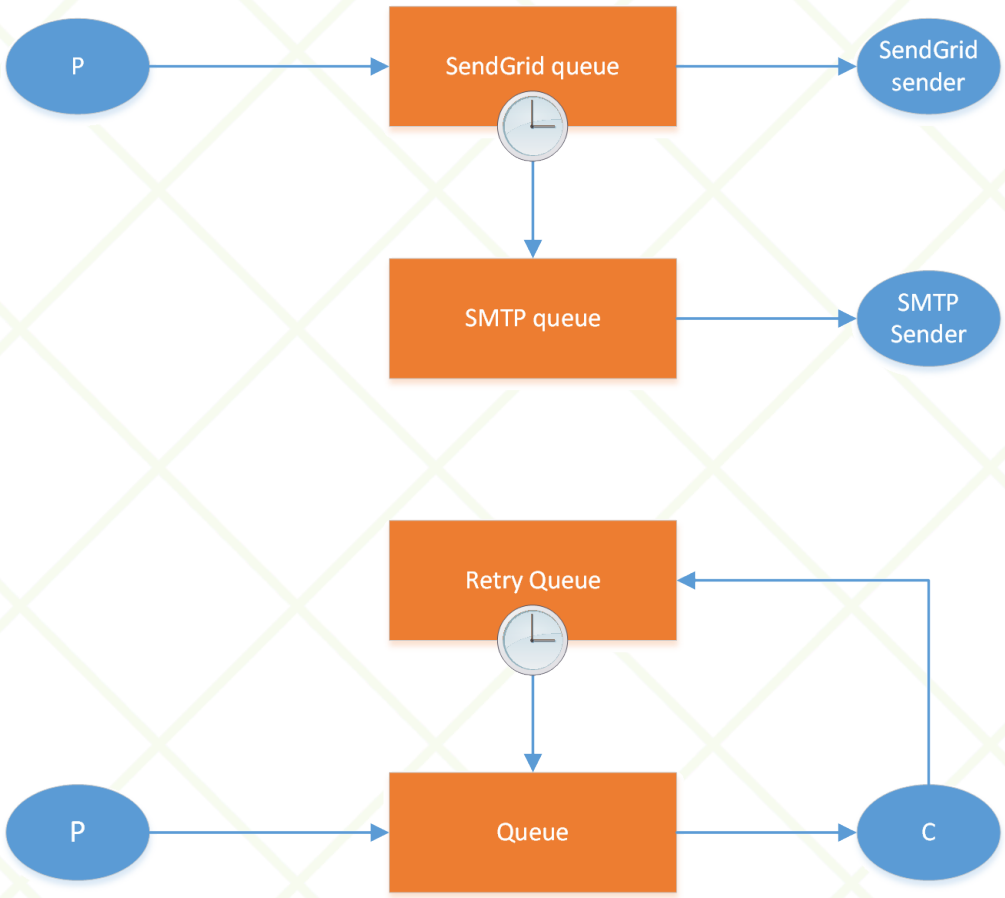
Developers Festival

Remote Procedure Calls (RPC)



Developers Festival

Time-To-Live extensions



Developers Festival

Time-To-Live extensions
Dead Letter exchange

```
// Per-Queue Message TTL
var args = new Dictionary<string, object>();
args.Add("x-message-ttl", 60000);
channel.QueueDeclare("myqueue", false, false, false, args);
```

```
// Per-Message TTL
var properties = new BasicProperties();
properties.Expiration = "60000";
channel.BasicPublish("my-exchange", "routing-key", properties, body);
```

```
// Per-Queue Message TTL
var args = new Dictionary<string, object>();
args.Add("x-message-ttl", 60000);
args.Add("x-dead-letter-exchange", "some.exchange.name");
args.Add("x-dead-letter-routing-key", "some-routing-key");
channel.QueueDeclare("myqueue", false, false, false, args);
```

Developers Festival

Questions?

Łukasz Łysik
llysik@gmail.com

Examples: https://github.com/lukasz-lysik/rabbitmq-brown-bag

Developers
Festival

www.4developers.org.pl

llysik@gmail.com / @lukasz_lysik