



**BITS Pilani**  
Pilani | Dubai | Goa | Hyderabad

# Message Construction

---

**Srikanth Gunturu**

Guest Faculty  
BITS, WILP

# In this segment

## Message Construction

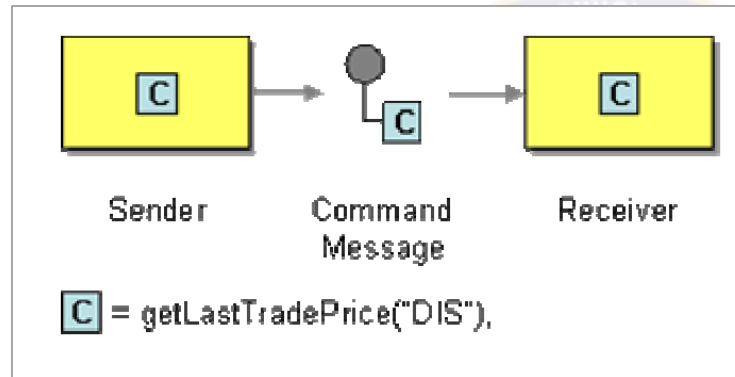
- Command Message
- Document Message
- Event Message
- Request-Reply
- Return Address
- Correlation Identifier
- Message Sequence
- Message Expiration
- Format Indicator



# Message Construction

## Command Message

- Command Message is used to invoke a procedure in another application, reliably.
- Asynchronous invocation of procedures, as opposed to RPC

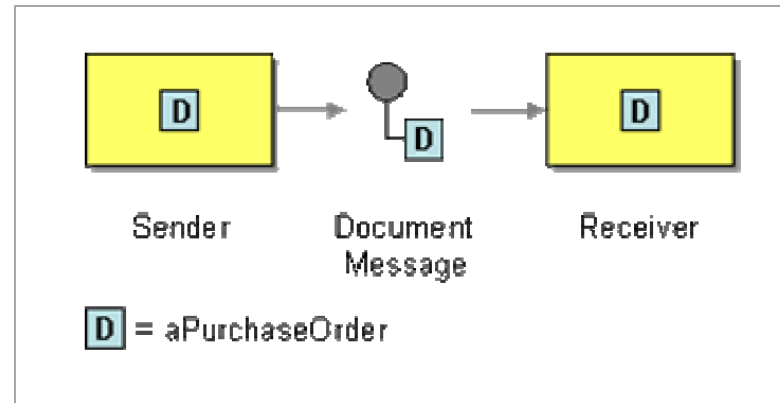


- Message contains command for the other application, there's no other specific indication on message
- Usually sent on point-to-Point channel
- Ex: SOAP messages

# Message Construction

## Document Message

- Document Message is used to reliably transfer a data structure between applications

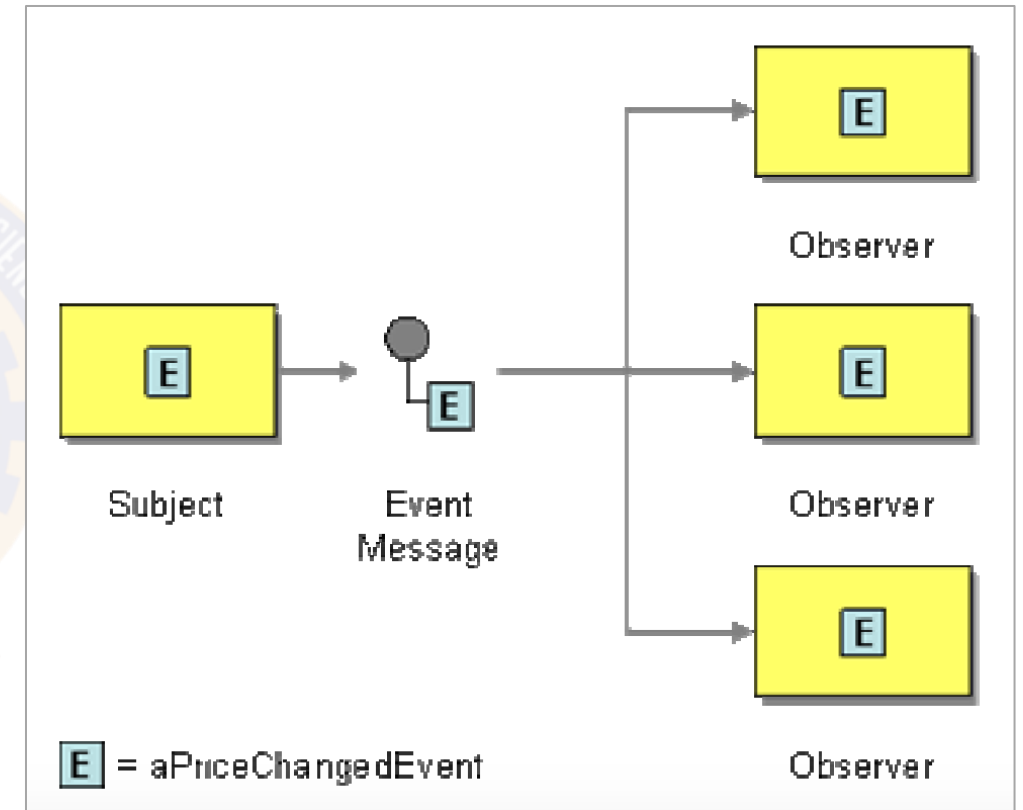


- Can contain any kind of message in the system, except that it represents a data object
- Usually sent on a point-to-point channel
- Ex: XML documents sent across the network

# Message Construction

## Event Message

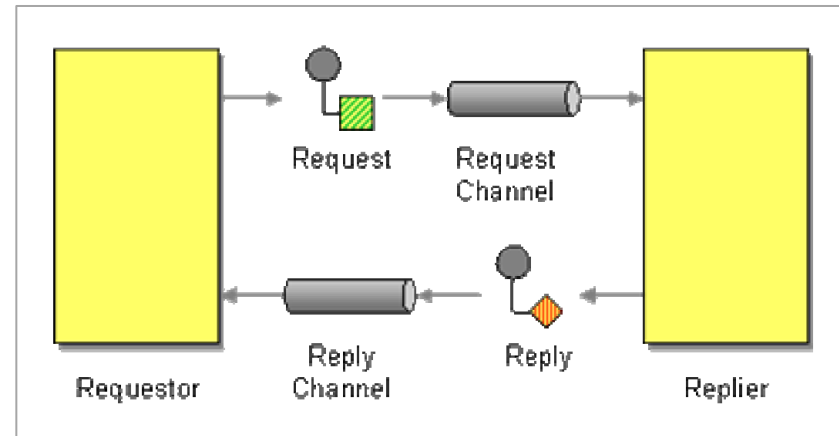
- Document Message is used to reliably send event notifications between application
- Sender will create an event object and sends it in a message (no special indicator used)
- Push model – sends event and the related state change info
- Pull model – sends only the event details and optionally additional info to obtain state details
  - Update
  - State Request
  - State Reply



# Message Construction

## Request-Reply

- Facilitates two way communication between sender/receiver using two-way channel

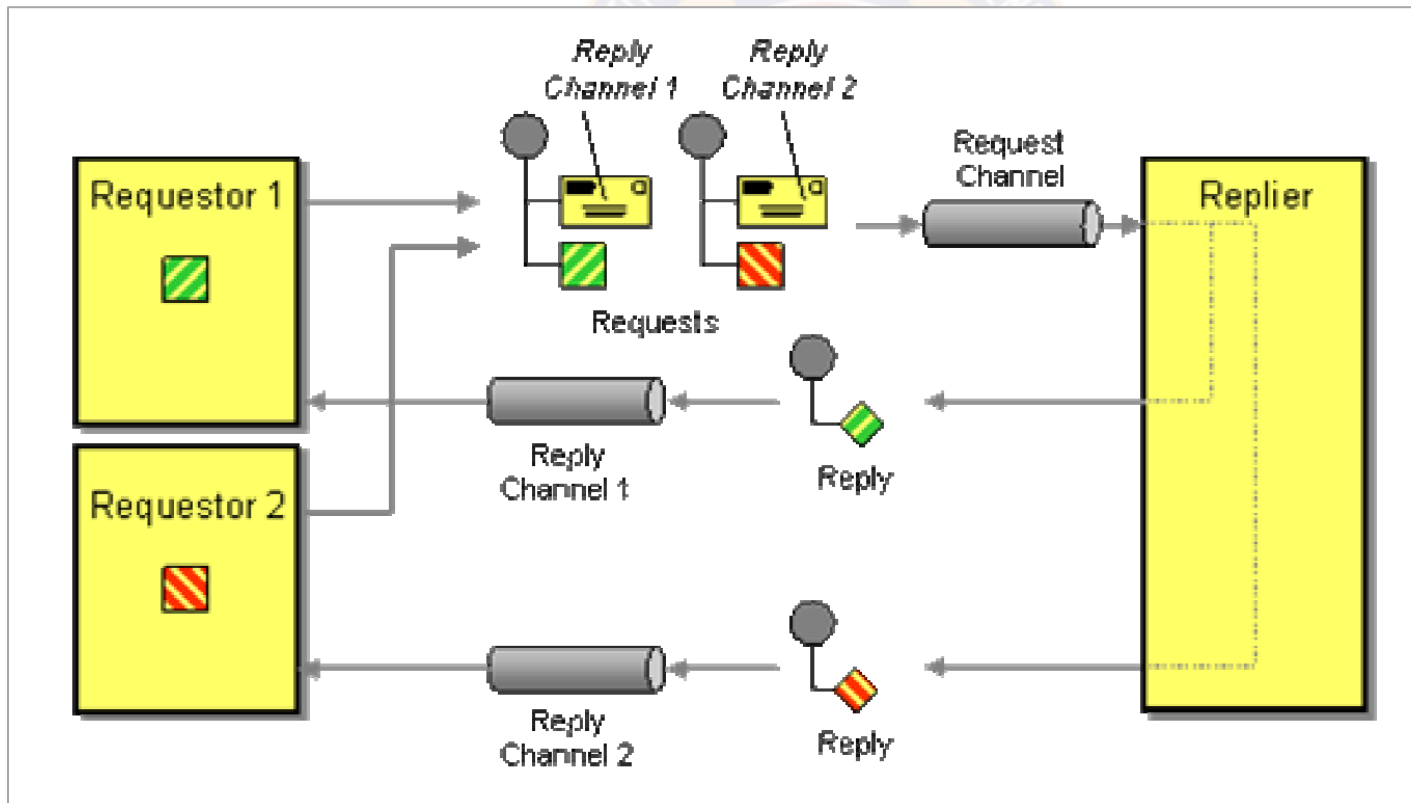


- Requestor – sends request message and waits for reply
- Replier – receives request message and sends the reply
- Request channel could be point-to-point or publish-subscribe
- Reply channel is almost always point-to-point
- Reply can be:
  - Synchronous (blocking)
  - Asynchronous (callback)

# Message Construction

## Return Address

- Asynchronous request-reply model has callback mechanism, which needs the receiver to know where to send the response, especially with multiple senders in question
- Return Address is supplied along with the Request Message, that will be used by receiver to send the reply back (Ex: JMS – JMSReplyTo field on request object)

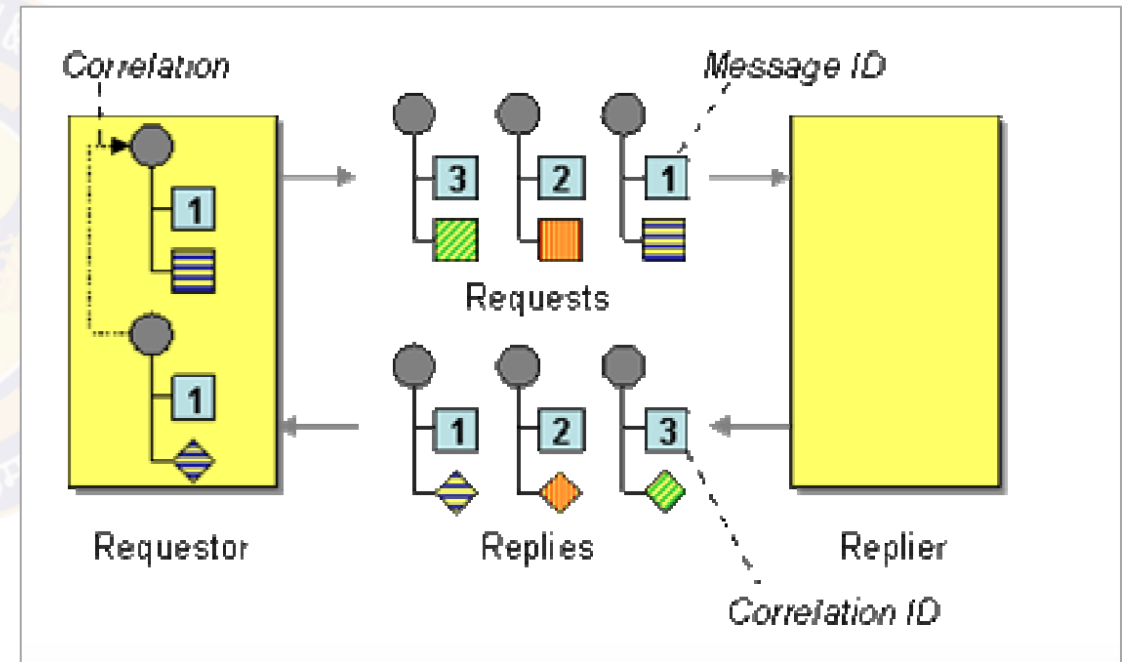




# Message Construction

## Correlation Identifier

- Each reply in request-reply model should contain a Correlation Identifier that uniquely indicates a which request this reply is for.
- Correlation Identifier has 6 parties to it:
  - Requestor – Sender App
  - Replier – Replier App
  - Request – Contains Request ID
  - Reply – Contains Correlation ID
  - **Request ID – A unique token from the request**
  - **Correlation ID – Unique token in response copied from Request ID**
- Correlation ID (and even Request ID) are specified in respective message headers
- Ex: JMS - JMSCorrelationID and JMSMessageID

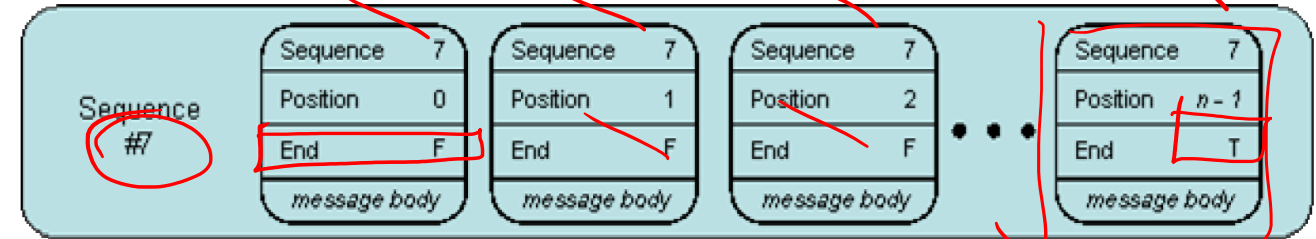
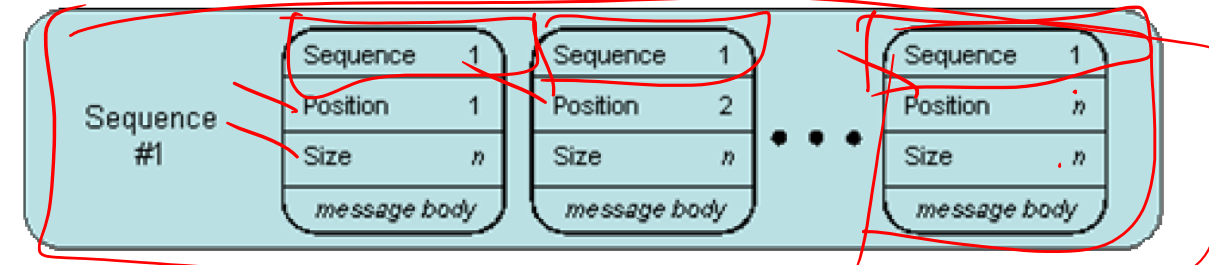




# Message Construction

## Message Sequence

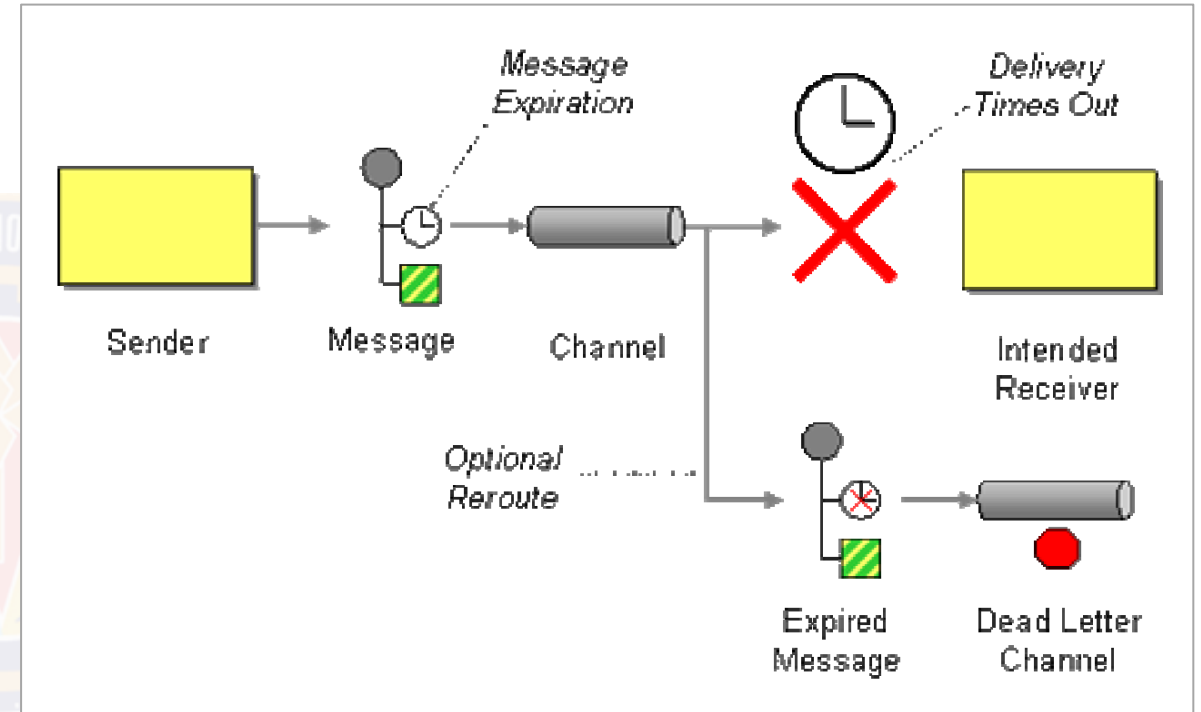
- Message Sequence is used whenever a large set of data may need to be broken into message-size chunks
- Each message in Message Sequence is marked with sequence identification fields
- Message Sequence identification fields
  - Sequence identifier
  - Position identifier
  - Size or End indicator
- Example Use cases:
  - Large document transfer
  - Multi item query
  - Distributed query



# Message Construction

## Message Expiration

- Message Expiration sets the time limit after which the message becomes stale
- Once the time limit expires, the message is not considered by consumers for processing, moved to Invalid Message Channel
- It could be configured to be rerouted to Dead Letter Channel for further handling
- Message Expiration is a timestamp
  - Relative (longevity), using sent time property
  - Absolute
- JMS implementation -  
`MessageProducer.setTimeToLive(long)`



# Message Construction

## Format Indicator

- The format indicator enables the sender to tell the receiver the format of the message
- This handles versioning and future changes
- Format indicator can be implemented as:
  - Version Number
  - Foreign Key
  - Format Document

```
<?xml version="1.0"?>
```

```
<!DOCTYPE greeting SYSTEM "hello.dtd">
```

```
<!DOCTYPE greeting [  
  <!ELEMENT greeting (#PCDATA)>  

```



# Thank You!

In our next session:  
Message Routing