Document title

EURONEXT DERIVATIVES MARKETS - OPTIQ® KINEMATICS SPECIFICATIONS

Revision number
4.4.0

Date
2 Nov 2020

Number of pages
148

Related SBE version
304

PREFACE

PURPOSE

The purpose of this document is to detail Kinematics for Optiq Order Entry Gateway and Market Data Gateway messages.

TARGET AUDIENCE

This document should be read by Euronext and Members using Optiq.

REVISION HISTORY

Version	Change Description
4.4.0	Introduction of SBE 304 – no impacts

ASSOCIATED DOCUMENTS

The following list of the associated documents, which either should be read in conjunction with this document or which provide other relevant information for the user:

- Euronext Markets – OEG Client Specifications – SBE Interface
- Euronext Markets – OEG Client Specifications – FIX 5.0 Interface
- Euronext Markets – Optiq & TCS Error List
- Euronext Markets – Optiq MDG Client Specifications
- Euronext Markets – Optiq File Specifications

Clients are advised to also refer to the Euronext Rules and Regulations documents for more details.

For the latest version of documentation please visit the IT Documentation page.

SUPPORT

Optiq Support Desk

Tel: +33 1 70 48 25 55

Email: optiq@euronext.com

CONTENTS

1. OVERVIEW

1.1 INTRODUCTION

This document provides an overview of the exchange of messages between the Optiq Order Entry Gateway (OEG), the clients' systems and the Market Data Gateway (MDG) for the Euronext Derivatives markets. It includes:

- Typical trading scenarios and the corresponding public and private messages for these scenarios, and the different cases they may cover;
- The names and IDs of the messages sent;
- The events that trigger the transmission of messages.

This overview is meant to provide a description of the main structures and concepts used within this document, to facilitate the review of the individual topics and cases covered within.

The messages that are sent between trading members and Optiq are referred to as private messages whereas the messages that are sent by the external broadcasting systems are referred to as public messages.

Private messages are exchanged exclusively between the clients' system and the Optiq matching engine via order entry gateways, for example to request information from the system or to issue a command (e.g., enter an order). Private messages are also sent back by Optiq via order entry gateways (OEGs) to the client's system to provide the information requested, or confirm that a command has been successfully executed (or not), as well as to notify of trades, etc.

Public messages are sent by Optiq via MDG to provide to all subscribing clients with anonymized Market Data, such as orders entered, best limits, executed trades, market events, etc.

The diagrams in this document express representative examples of message sequences and other scenarios can be figured out from the ones described inside that document. The details of the message contents may vary depending on the example.

The diagrams also endeavour to represent as close to reality as possible the sequence in which events and steps occur, and messages are sent. This introduction provides indication when such cases are not feasible to represent faithfully due to complexity of graphical representation.

For a complete description of the messages and their fields, please refer to the associated document:

- Optiq Order Entry Gateway Messages Specifications SBE;
- Optiq Order Entry Gateway Messages Specifications FIX;
- Optiq Market Data Gateway Messages Specifications.

1.2 MESSAGE CODES AND NAMES

1.2.1 Private Messages

Possible Direction:
Inbound - Client ▶OEG (From Client To OEG)
Outbound - Client ◀OEG (To Client From OEG)

Order Entry Gateway message identifiers, which include message codes and names, are provided throughout the message kinematics section as shown below:

- For Inbound messages (example for **NewOrder** message):

01 represents the SBE Bin Code.
[D] represents the FIX Code.

- For Outbound messages (example for **Ack** message):

03 represents the SBE Bin Code.
[8] represents the FIX Code.

- When there is a difference of kinematics between SBE Bin and FIX protocols, the flows are distinguished as shown below (example of **Logout** message):

The SBE Bin message Code is represented alongside the FIX kinematic.

- The exhaustive list of SBE Bin and FIX message codes that and names, which are used on the Derivatives markets, is provided in the table below:

SBE Bin Message Code	SBE Bin Message Name	FIX Message Code
01	New Order	**D**
03	Ack	**8**
04	Fill	**8**
05	Kill	**8**
06	Cancel Replace	**G**
07	Reject	**9**
08	Quotes	**i**

SBE Bin Message Code	SBE Bin Message Name	FIX Message Code
09	Quote Ack	**b**
10	Quote Request	**R**
	QuoteRequestReject	**AG**
12	Cancel Request	**F**
13	Mass Cancel	**q**
14	Mass Cancel Ack	**r**
15	Open Order Request	**AF**
17	Ownership Request Ack	**U29**
18	Ownership Request	**U18**
19	Trade Bust Notification	**8**
39	User Notification	**CB**
	RequestAckMessage	**Uy**
47	MM Sign In	
48	MM Sign In Ack	
50	Instrument Synchronization List	**U50**
51	Synchronization Time	**U51**
60	Security Definition Request	**c**
61	Security Definition Ack	**d**
62	MM Protection Request	
63	MM Protection Ack	
64	New Wholesale Order	**U64**
65	Wholesale Order Ack	**U65**
66	Request For Implied Execution	**U66**
67	Cross Order	**U67**
100	Logon	**A**
101	Logon Ack	
102	Logon Reject	**3**
103	Logout	**5**
106	Heartbeat	**0**
107	TestRequest	**1**
108	TechnicalReject	
	ERGCommand	**U68**
	ERGCommandAck	**U69**
	GetRiskControls	**U70**
	RiskControlsDetails	**U71**

1.2.2 Public Messages

Possible Direction:
Outbound - MDG ▶Client (From MDG To Client)

Market Data Gateway message identifiers, which include message codes and names, are provided throughout the message kinematics section as shown below:

- For public messages sent to the Market:

- The exhaustive list of message codes and names is provided in the table below:

Message Code	Message Name
1001	Market Update
1003	Price Update
1004	Full Trade Information
1005	Market Status Change
1006	Timetable
1008	Real Time Index
1009	Statistics
1011	Index Summary
1012	Strategy Standing Data
1013	Contract Standing Data
1014	Outright Standing Data
1016	LIS Package Structure
1101	Start Of Day
1102	End Of Day
1103	Health Status
1106	Technical Notification
2101	Start Of Snapshot
2102	End Of Snapshot

- For readability purposes on MDG side, multiple channels are not considered. Diagrams show only a single set of channels.

1.2.3 Graphical representations

The diagrams in this document represent the following components:

- The overall Optiq system which is the new integrated trading platform for the Euronext markets, shown as below:

- The Order Entry Gateway which is the private interface between clients and the matching engine:

- The Market Data Gateway (MDG) which sends public messages to the Market:

- The clients' systems, used by the client to send and receive private messages to and from the matching engine, here referred to as Member:

- And the Market represents all the publicly available data sent by the exchange to all subscribers of the public feeds:

Note: for readability purposes the field names in the graphs are abbreviated, e.g. *Order Quantity* is referred to as *OrderQty*, etc.

- Some diagrams are preceded by an order book to facilitate understanding.

  T# indicates the sequence in time for the submission of messages to the order book.

Symbol Index of the Instrument

At the start of kinematics, the order book contains a resting order with Time T0. T1 and T2 should be ignored.

In diagram, some private messages are followed by T1 (or T2,..). It means that at this stage, T1 (or T1 and T2) is added in the order book at the time T1 and/or T2.

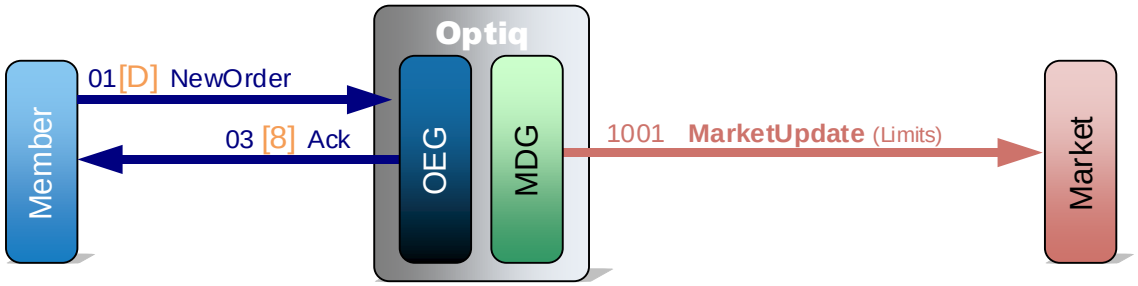
If diagrams are not preceded by an order book, it means that the order book is empty.

1.2.4 Main Principles

A request sent by a client will usually:

- Trigger an outbound acknowledgment message from the matching engine which is exclusively sent to this client, and in some cases this can be followed by other notification messages;
- Trigger one or several market data messages if the request has a direct impact on the Central Order Book (COB).

Below is an abbreviated, generic example of the interaction of messages, for the submission of a **NewOrder** (01) (FIX D) message:



When required diagrams may include division into steps of the scenarios displayed, that are delineated by dotted lines, and are denoted by the number of the step. Numbers denoting the steps in the diagram correspond to the numbers used in the explanation below the diagram.

More detailed diagrams may include additional details for the individual messages, such as, Side, Order Priority, Price, Quantity, etc.

1.2.5 Important Notes

1.2.5.1 Full Trade Information generation

A public message **FullTradeInformation** (1004) is sent in the dedicated Trade and Referential (REFT) channel each time a **MarketUpdate** (1001) following a trade is disseminated to the market by MDG. But for readability purposes it is not shown on the kinematics diagrams.

1.2.5.2 Order Update generation

There is **no** MDG **OrderUpdate** (1002) message dissemination for Derivatives. For Derivative markets, updates of Market Data are provided by price level only, not by individual orders.

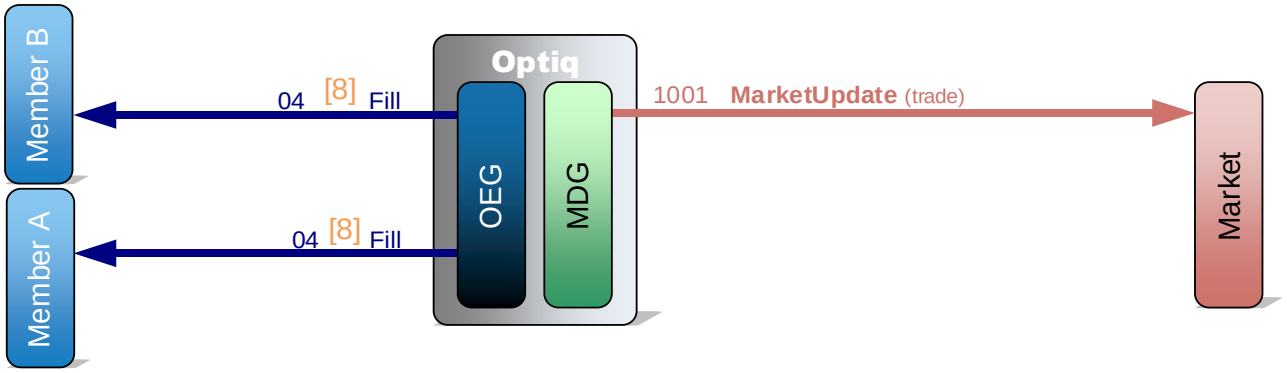
1.2.5.3 Implied prices

Implieds are not considered as orders however the associated Implied prices volume are displayed on the market only if they contribute to the Best Limit.

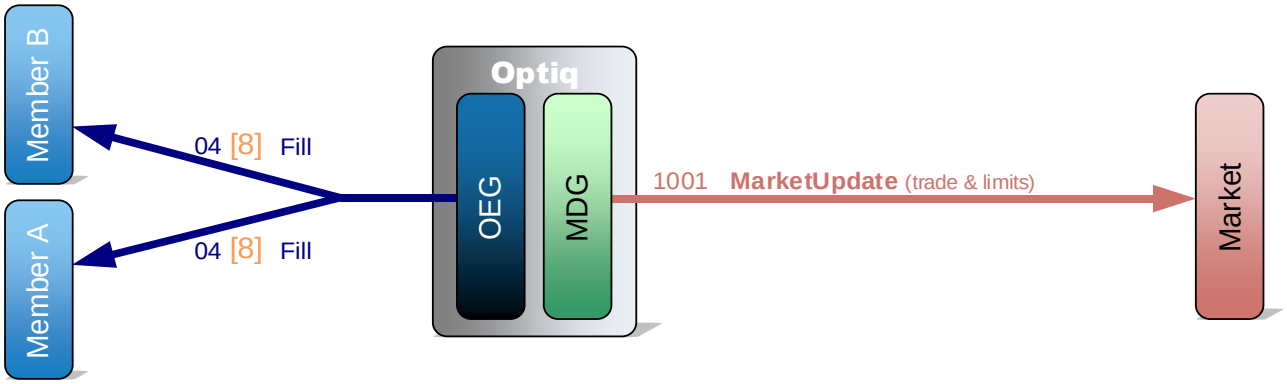
When an implied price contributes to a Best limit, the volume available on the market at that price increases without incrementing the number of orders. This logic allows client to distinguish volumes of implied prices vs. those of explicit orders. As such it is possible to have a Best Limit displayed with price and volume but with a number of orders equal to zero (when relying exclusively on implied prices).

1.2.5.4 Simultaneity of Private Messages

In all the diagrams of this document multiple private messages resulting from the same event (e.g. **Fill** (04) (FIX 8) messages due to a trade execution) are represented as if they were sent one by one:



This is done to reduce complexity of the graphical representation and to improve readability. In reality such messages are sent at the same moment to the different members:



For the rest of this document please assume that messages resulting from the same event and sent to different clients are sent at the same moment.

2. COMMON KINEMATICS

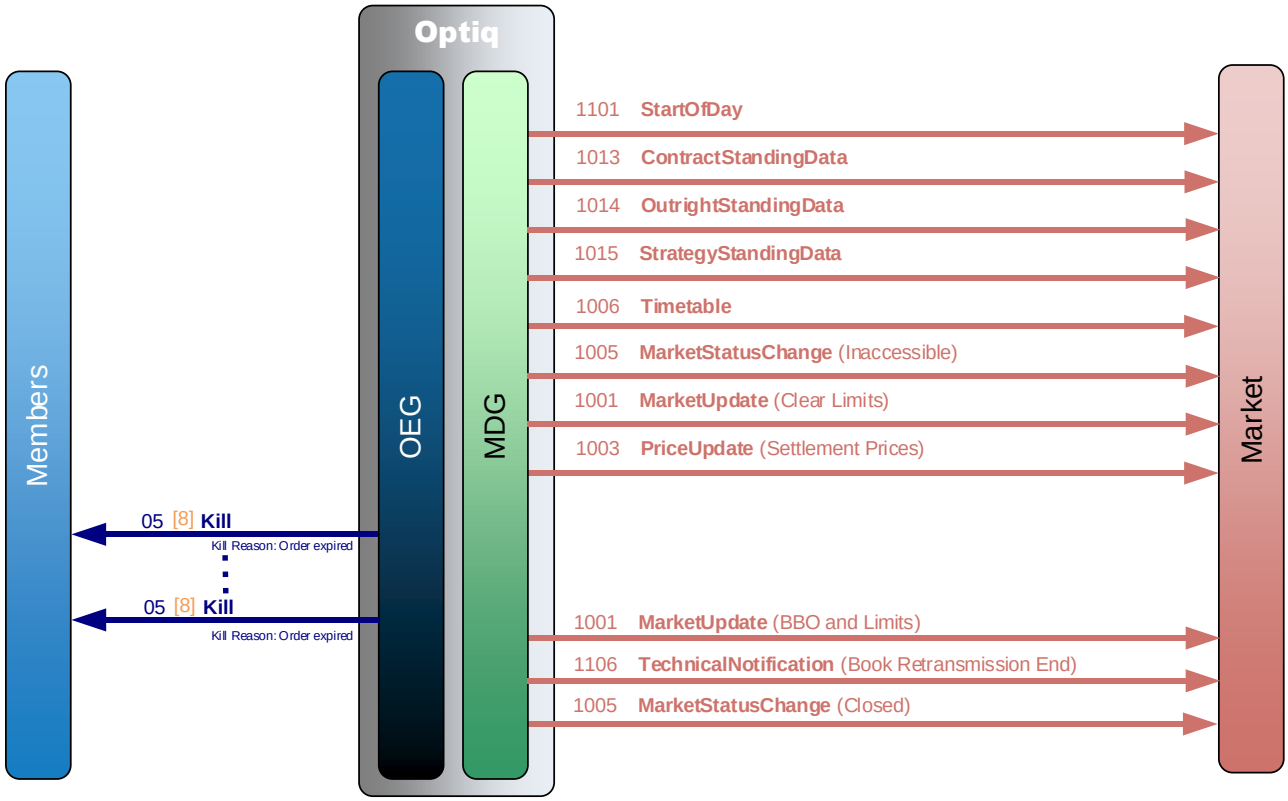
The following public messages contain repeating sections:

- PriceUpdate;
- MarketUpdate;
- MarketStatusChange;
- FullTradeInformation.

Detailed information regarding repeating sections can be found in the *Euronext Markets – MDG Client Specifications* document.

2.1 TRADING SESSION MANAGEMENT

2.1.1 Initialisation of a New Trading Day



At the initialization of each new trading day the Exchange sends the following public messages (the generation sequence is guaranteed to always be the same):

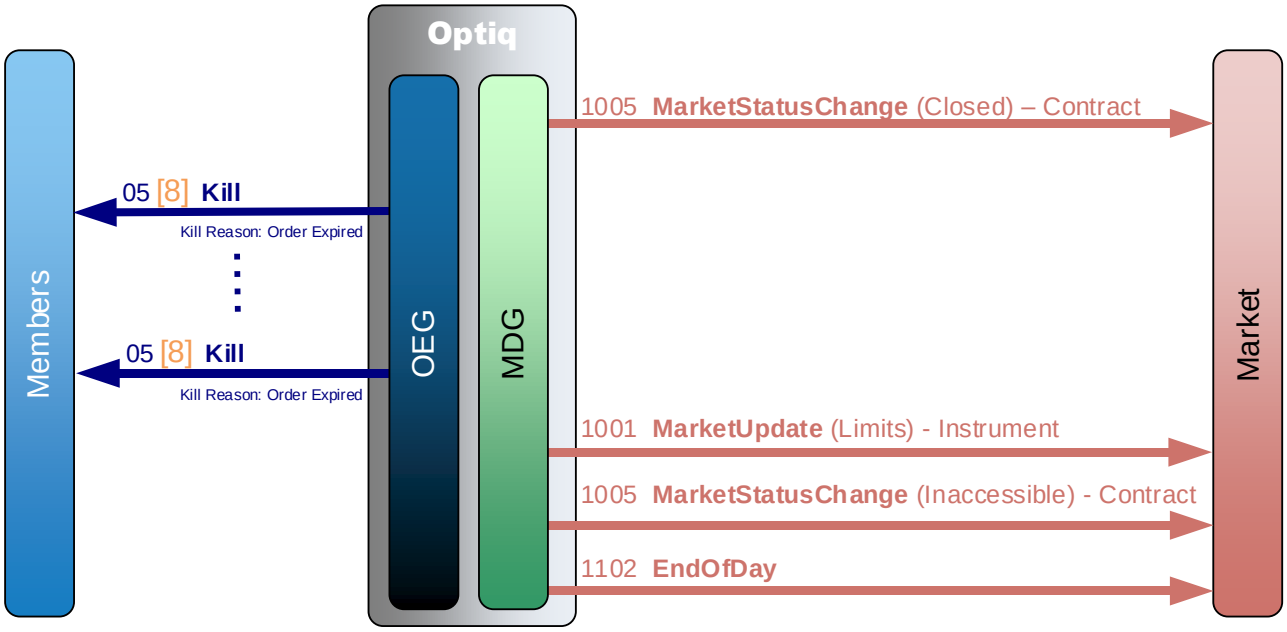
- **StartOfDay** (1101) message: It is always the first message of the day, which indicates the date of the trading session.

- **ContractStandingData** (1013) message: For every single contract it provides to the members all the necessary Contract data for the trading day.

- **OutrightStandingData** (1014) message: For every single outright it provides to the members all the necessary Outright data for the trading day.

- **StrategyStandingData** (1015) message: For every active strategy (i.e. having GTC / GTD orders) it provides to the members all the necessary Strategy data for the trading day.
- **Timetable** (1006) message: It provides all the trading patterns that are used across all the contracts.
- **MarketStatusChange** (1005) message: For every single contract it is sent with *Book State* set to 'Inaccessible', *Trading Period* set to 'Opening' and *Rebroadcast Indicator* set to '0'.
- **MarketUpdate** (1001) message: For every single instrument the limits are cleared at the beginning of the day.
- **PriceUpdate** (1003) message: For every single Outright instrument it provides the previous day's daily Settlement Price.
- **MarketUpdate** (1001) message: For every single instrument it provides both BBO and depth of the order book for order-driven markets with *Rebroadcast Indicator* set to '1'.
- **TechnicalNotification** (1106) message: For every single instrument it notifies the end of the book retransmission.
- **MarketStatusChange** (1005) message: For every single contract a *Book State* set to 'Closed' is sent at the scheduled time.

2.1.2 End Of Day



At the end of the trading day, when the contract is in 'Closed' State with *Trading Period* of 'Closing', expired orders are killed, thus a private **Kill** (05) (FIX 8) message will be sent for each expired order. The orders killed at the end of the day include Day orders, and unexecuted orders in Delta-neutral strategies.

Then, **MarketUpdate** (1001) messages are sent per instrument to update the Limits.

Note: Updates and cancellation of orders during Closed phase in private feed generate **MarketUpdate** (1001) to update limits.

At the scheduled time a **MarketStatusChange** (1005) message is sent for the 'Inaccessible' phase.

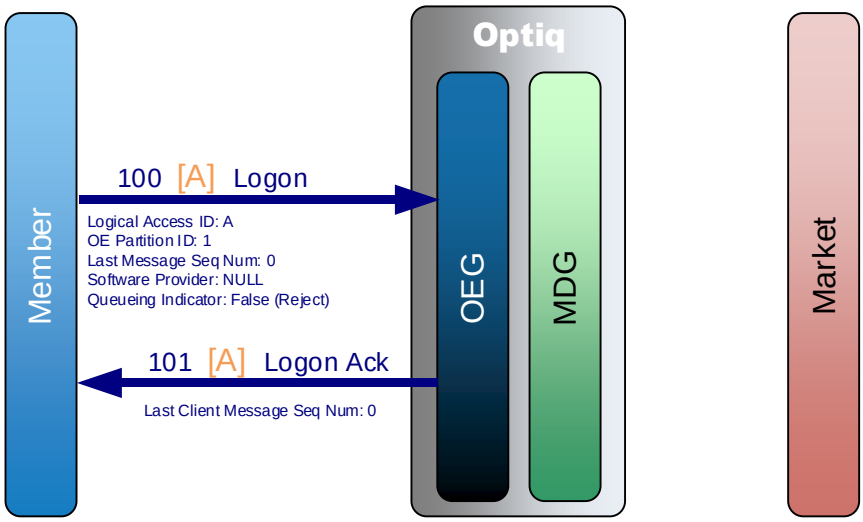
The public **EndOfDay** (1102) message is always the last message sent by the Exchange, it notifies that the platform and its network are now closed (members cannot send nor receive messages until the next trading day).

Note: Clients should be aware that orders eliminated at the end of the session will not be re-broadcast at the start of the next trading session. In case of disconnection at the end of the sessions, clients are advised to remove any expired day orders from their book.

2.2 ADMIN MESSAGES

Please note that all administrative messages exchanged between the client and the exchange are issued per OE session (physical connection).

2.2.1 Successful Logon



At the beginning of each trading day the members must log on to the Order Entry Gateway prior to send any other message.

In order to initiate the connection, the member sends a **Logon** (100) (FIX A) message. If the logon is successful the OEG sends back a **LogonAck** (101) message (only in SBE Bin protocol).

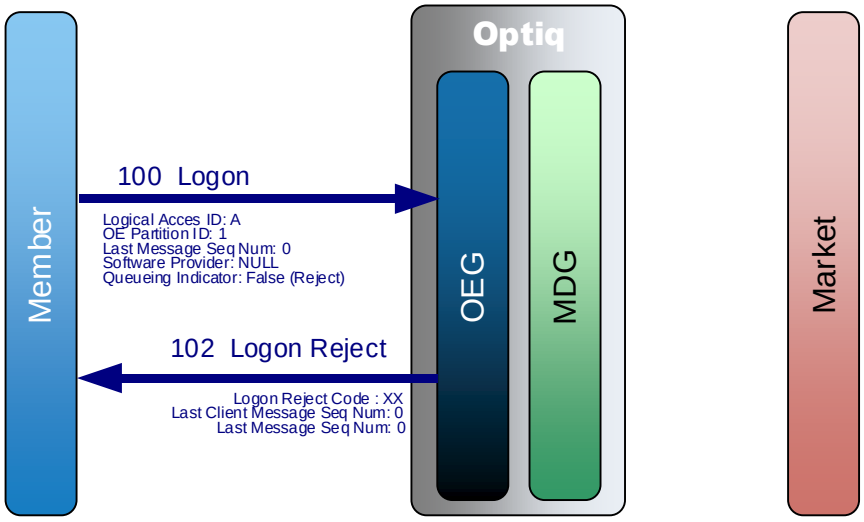
In FIX protocol, if the logon is successful the OEG sends back a **Logon** (A) message. While in SBE the sequence numbers start from 0, in FIX the sequence numbers start from 1.

No message is sent to the Market.

2.2.2 Logon Rejection

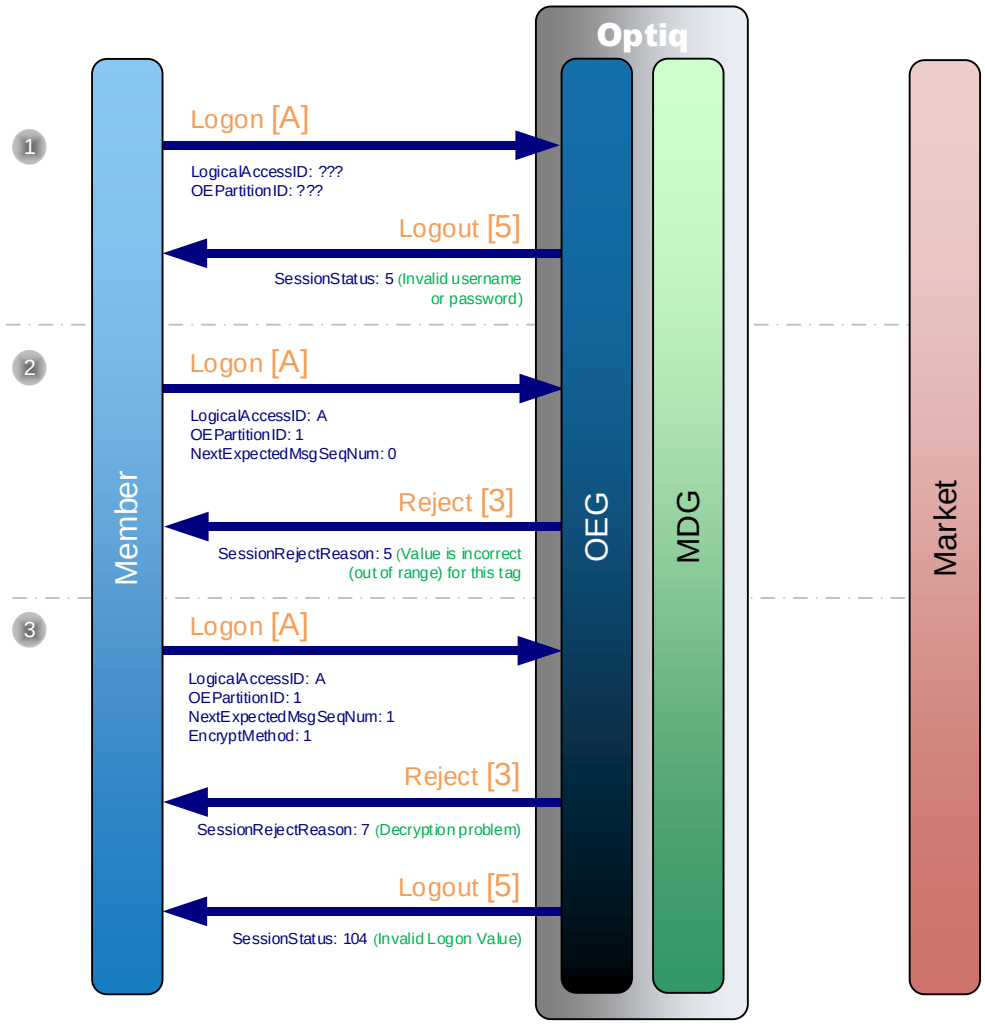
Logon rejection behaviour prescribed by the FIX protocol is different from that adopted for SBE, and for this case two different diagrams are provided, each one specific to the protocol. While at high level the behaviour might be different, the result of the Logon Rejection will be the same.

2.2.2.1 Logon Rejection in SBE



A member sends a **Logon** (100) message in order to initiate the connection with the OEG. If for any reason the **Logon** (100) message is not accepted, the OEG sends back a **LogonReject** (102) message. No message is sent to the Market.

2.2.2.2 Logon Rejection in FIX



A member sends a **Logon** (A) message in order to initiate the connection with the OEG. If for any reason the **Logon** (A) message is not accepted, the OEG sends back a **Logout** (5) message.

Additionally, OEG sends a **Reject** (3) message if the **Logon** (A) is poorly formatted.

① A member sends a **Logon** (A) message in order to initiate the connection with the OEG. If the fields *LogicalAccessID* and *OEPartitionID* are wrong or not recognized for the associated *SenderCompID*, OEG sends back a **Logout** (5) message with *SessionStatus* set to '5' (Invalid username or password).

② A member sends a **Logon** (A) message in order to initiate the connection with the OEG. If the field *NextExpectedMsqSeqNum* is set to '0', OEG sends back a **Reject** (3) message with SessionRejectReason set to '5' (Value is incorrect (out of range) for this tag).

③ A member sends a **Logon** (A) message in order to initiate the connection with the OEG. If the value of the field *EncryptMethod* is different than '0', OEG sends back a **Reject** (3) message with SessionRejectReason set to '7' (Decryption Problem).

No messages are sent to the Market.

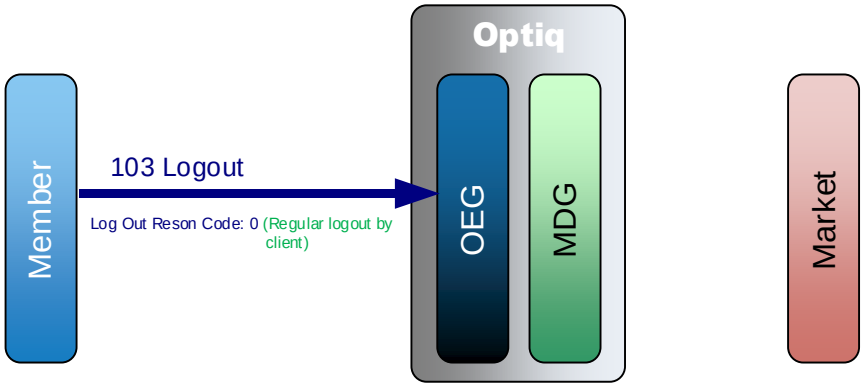
2.2.3 Logout

Logout behaviour prescribed by the FIX protocol is different from that adopted for SBE, and for this case two different diagrams and descriptions of steps are provided, each one specific to the protocol. While at high level the behaviour might be different, the result of the Logout from the system will be the same.

Logout is used to improve session management processes. This message identifies to the exchange if the client has disconnected on purpose or because of technical issue.

Note: This will trigger the Cancel on Disconnect mechanism (only on the specific orders on which it is enabled).

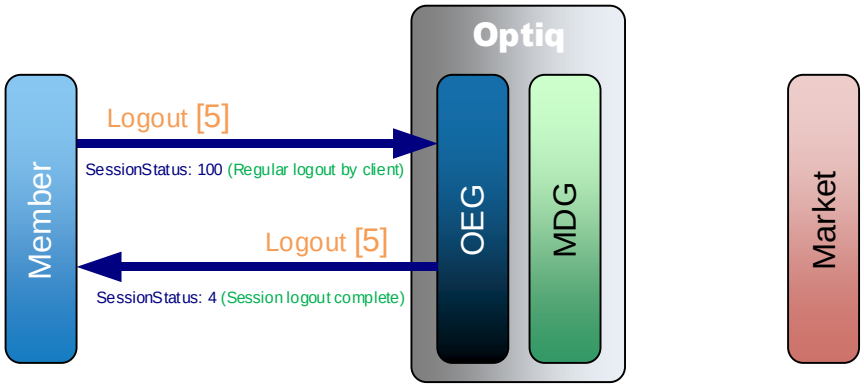
2.2.3.1 Logout in SBE



In order to log out the member sends a **Logout** (103) message, OEG immediately closes the physical connection.

No message is sent to the Market.

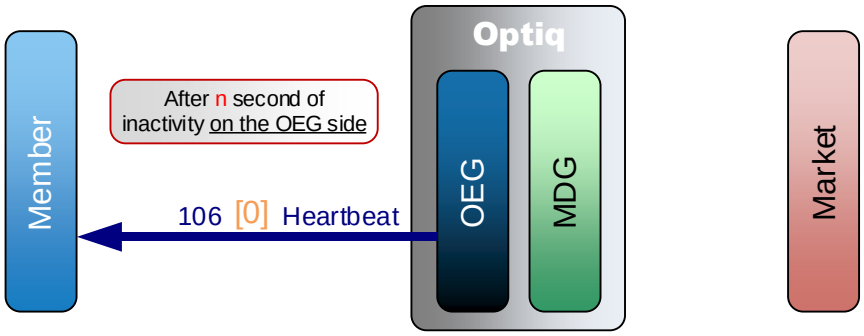
2.2.3.2 Logout in FIX



In order to log out the member sends a **Logout** (5) message with *SessionStatus* set to '100' (Regular logout by client). In response OEG firstly sends back a **Logout** (5) confirmation message with *SessionStatus* set to '4' (Regular logout complete) and then closes the physical connection.

No message is sent to the Market.

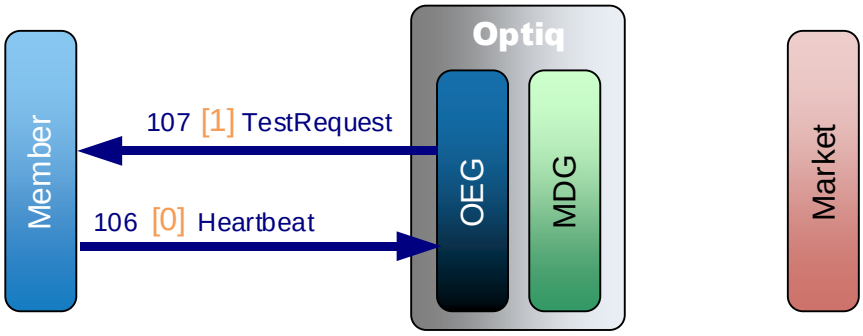
2.2.4 Heartbeat



After *n* second(s) of inactivity on Optiq OEG session side (i.e. when the OEG has not sent any message since *n* second(s)) the OEG sends a **Heartbeat** (106) (FIX 0) message to the member to signify that the connection is still alive from Optiq perspective. The member does not have to respond; it is only a notification from the OEG.

Note: The value of *n* will be provided for each Optiq Segment in the *Connectivity Specifications*.

2.2.5 Test Request



Test Request from Exchange to Client

After *n* second(s) of inactivity on the member side (i.e. when the OEG has not received any message since *n* second(s)) the OEG sends a **TestRequest** (107) (FIX 1) message to the member to request confirmation that the connection is still alive on member side.

The parameter *n* is identified per Optiq Segment in the Connectivity specifications as the period of inactivity.

For SBE:

- If the member issues a message in the following *n* second(s), the **TestRequest** (107) is ignored. Note the message issued by the member can be a **Heartbeat** (106) message or any other application message such as **NewOrder** (01), **CancelReplace** (06).
- If the member does not issue any message in the following *n* second(s), the OEG closes the connection. (This triggers the Cancel on Disconnect mechanism on the orders for which it is enabled.)

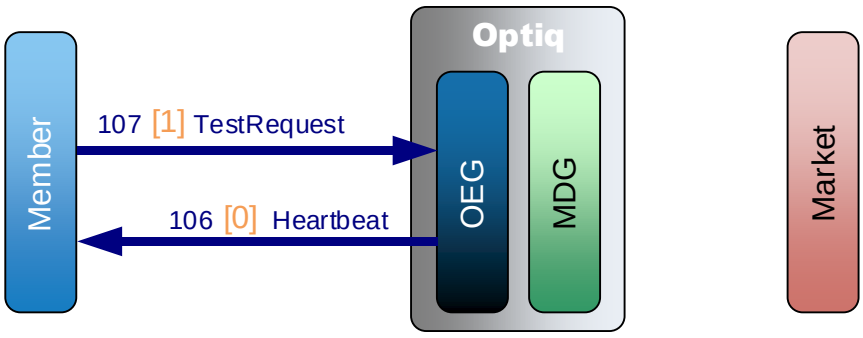
For FIX:

Member has *n* seconds to answer with a **HeartBeat** (0) messages, containing the same value in *TestReqID* (112), as the one sent in the original **TestRequest** (1) message sent by the OEG.

- Following receipt of the **TestRequest** (1) message, and for the duration of the inactivity period member may send other messages, including application messages and **HeartBeat** (0) messages. The application messages (such as **NewOrderSingle** (D), **CancelReplace** (G)) will be processed by OEG
- At the end of the period of inactivity if the member has not answered with a **HeartBeat** (0) message that contains the expected value of *TestReqID* (112), the client will be disconnected. (This triggers the Cancel on Disconnect mechanism on the orders for which it is enabled.)

Test Request from Client to Exchange

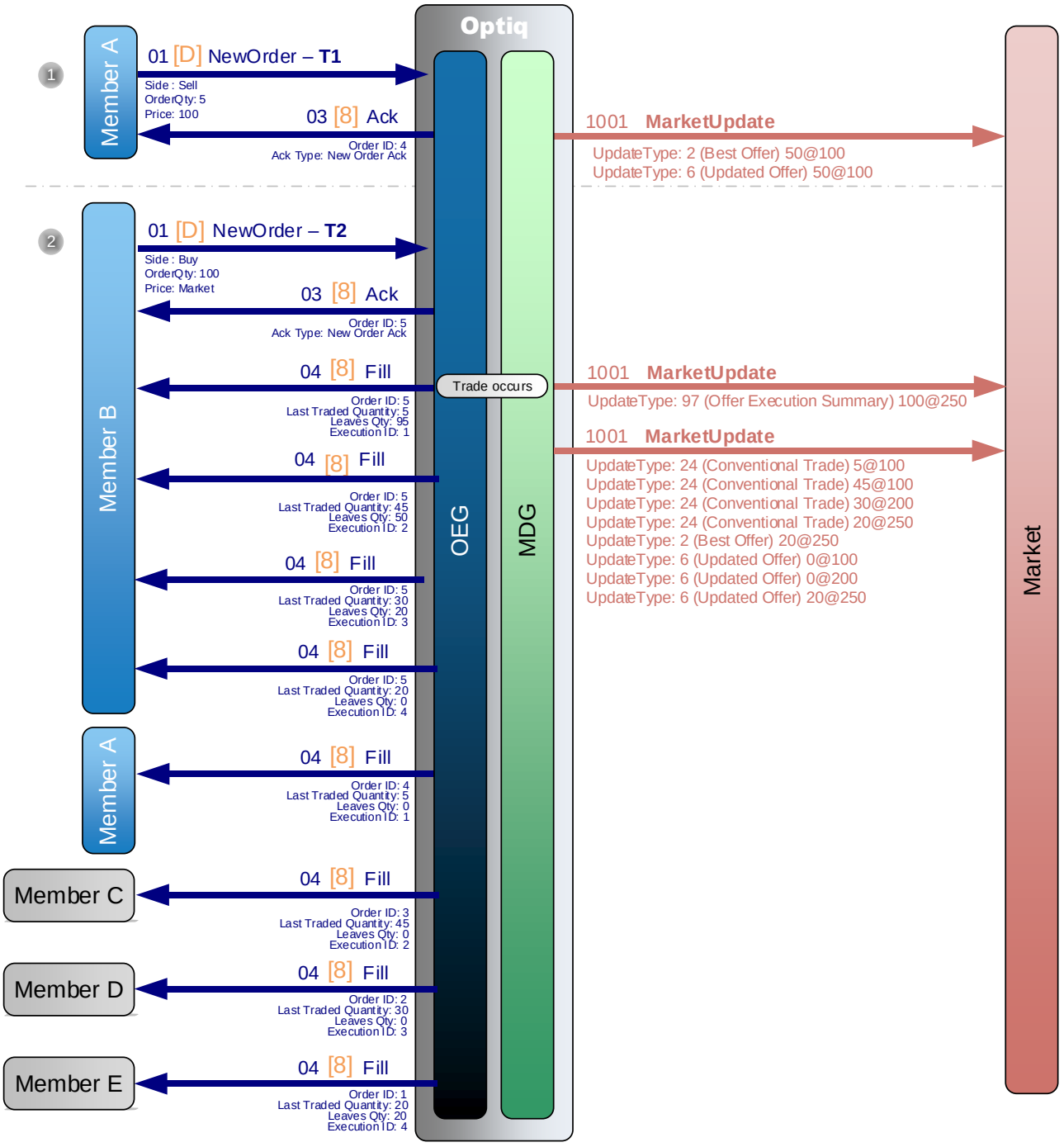
The **TestRequest** (107) (FIX 1) message can also be sent by the Member, in this case the OEG will respond with a **Heartbeat** (106) (FIX 0) message:



2.3 ENTERING AN ORDER

2.3.1 Incoming Order Matched Fully

Symbol Index: M1					
		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
T2	100	Market	100	45	T0
			100	5	T1
			200	30	T0
			250	40	T0



① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 5 and a price of 100.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) is sent to the market to update the limit.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 100 and a Market order type.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately, and fully, matches the four orders that are in the order book at this time, and the OEG generates a private **Fill** (04) (FIX 8) message to each member involved in the trade, for each leg of the trade. All the Fill messages are sent simultaneously.

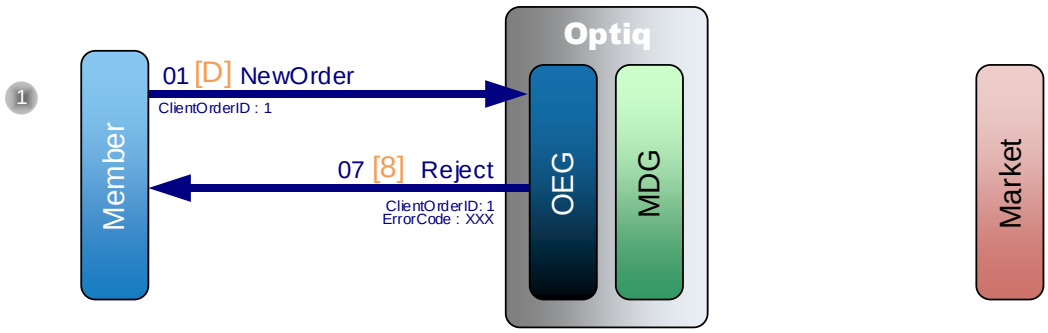
A public **MarketUpdate** (1001) message is immediately sent to the market for the Execution Summary.

Only then, public **MarketUpdate** (1001) messages are sent to the market for the Trades and update of the Limits.

Note: Market Data kinematics are based on the matching scenario Explicit vs Explicit in an Outright. For the full list of scenarios see section "Trading Kinematics"

No dedicated **MarketUpdate** (1001) message is sent for the entry of the second order as it is immediately matched.

2.3.2 New Order Rejected



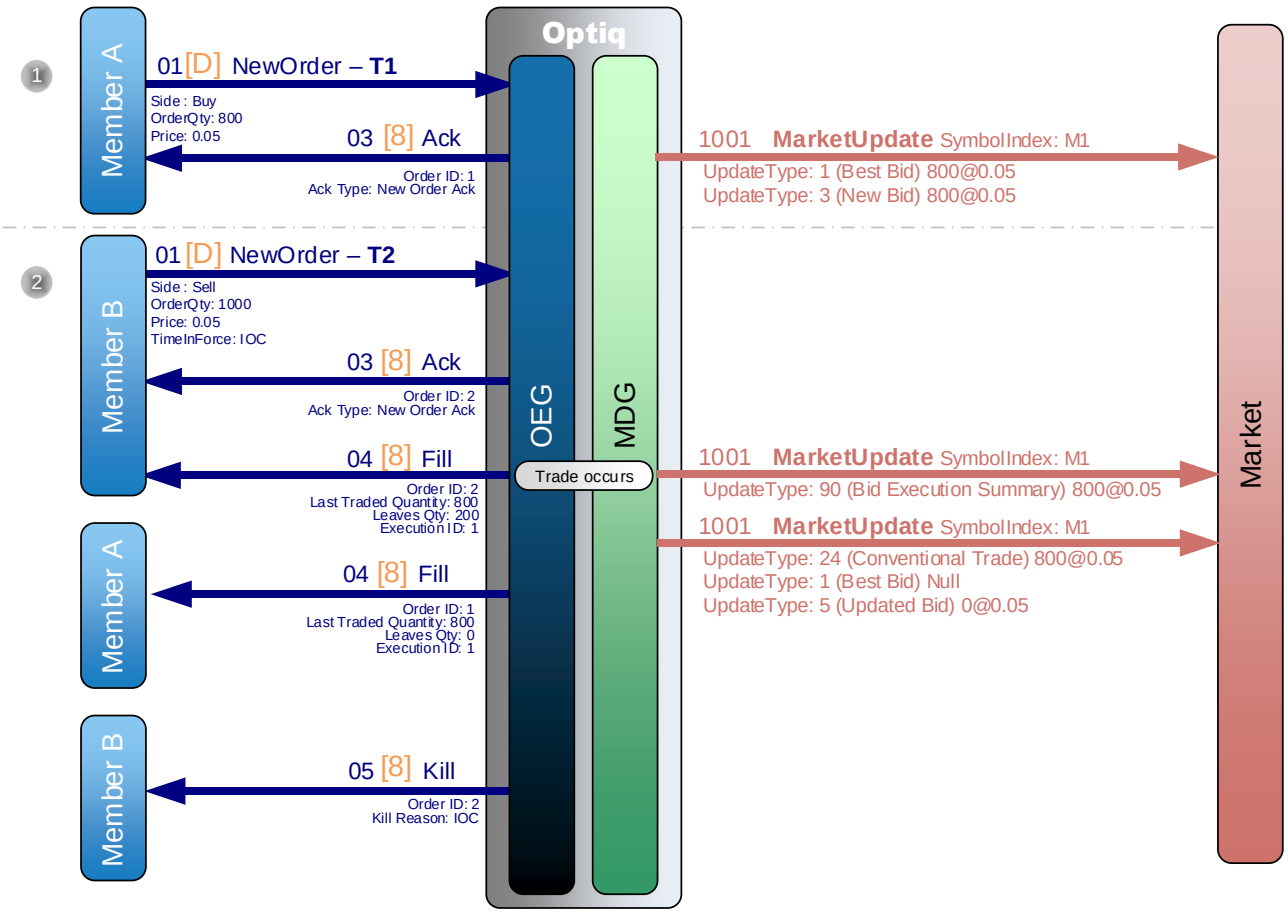
① A Member sends a private **NewOrder** (01) (FIX D) message to enter an order.

If the order is rejected OEG sends back a private **Reject** (07) (FIX 8) message with an *Error Code*. The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*.

No message is sent to the Market.

2.3.3 Immediate Or Cancel Order Partially Filled

SymbolIndex: M1					
		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
T1	800	0.05	0.05	1000	T2



① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 800 and a price of 0.05.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to update the limit.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 1,000, a price of 0.05 and a validity condition of Immediate or Cancel (IOC).

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately matches the first order for a quantity of 800 and OEG sends back a private **Fill** (04) (FIX 8) message to each member to notify the trade execution. As the remaining quantity cannot be immediately filled, OEG sends back to the Member B a **Kill** (05) (FIX 8) message to cancel the remaining quantity of that order.

A public **MarketUpdate** (1001) message is immediately sent to the market for the Execution Summary.

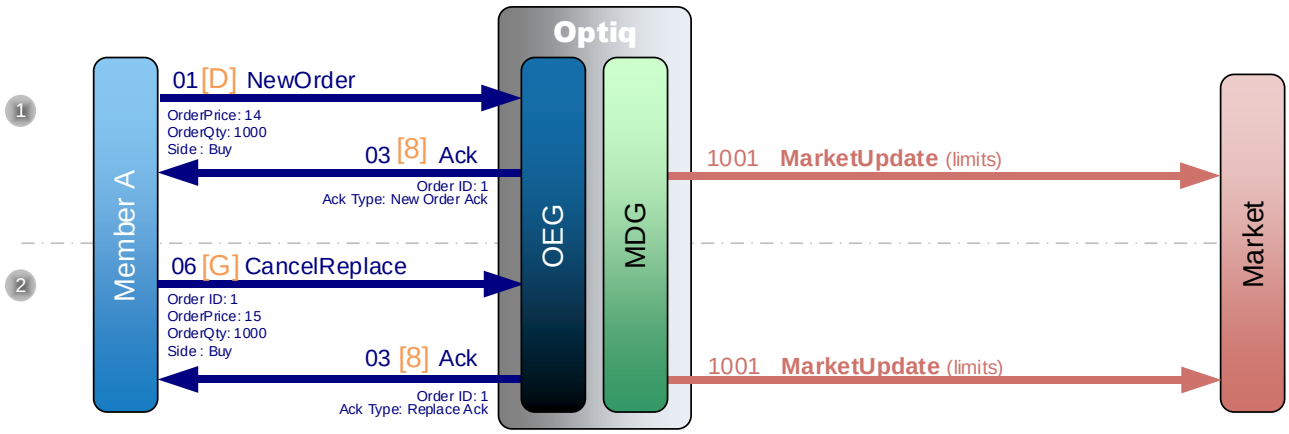
Only then, public **MarketUpdate** (1001) messages are sent to the market for the Trades and the Limits.

Note: Market Data kinematics are based on the matching scenario Explicit vs Explicit in an Outright. For the full list of scenarios see section "Trading Kinematics".

As IOC never enters in the book, there is no dedicated **MarketUpdate** (1001) message sent to the market.

2.4 MODIFYING AN ORDER

2.4.1 Modifying a Resting Order



① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new buy order with a quantity of 1,000 and a price of 14.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to update the limit.

② The same Member sends a private **CancelReplace** (06) (FIX G) message to modify the order by increasing the price up to 15.

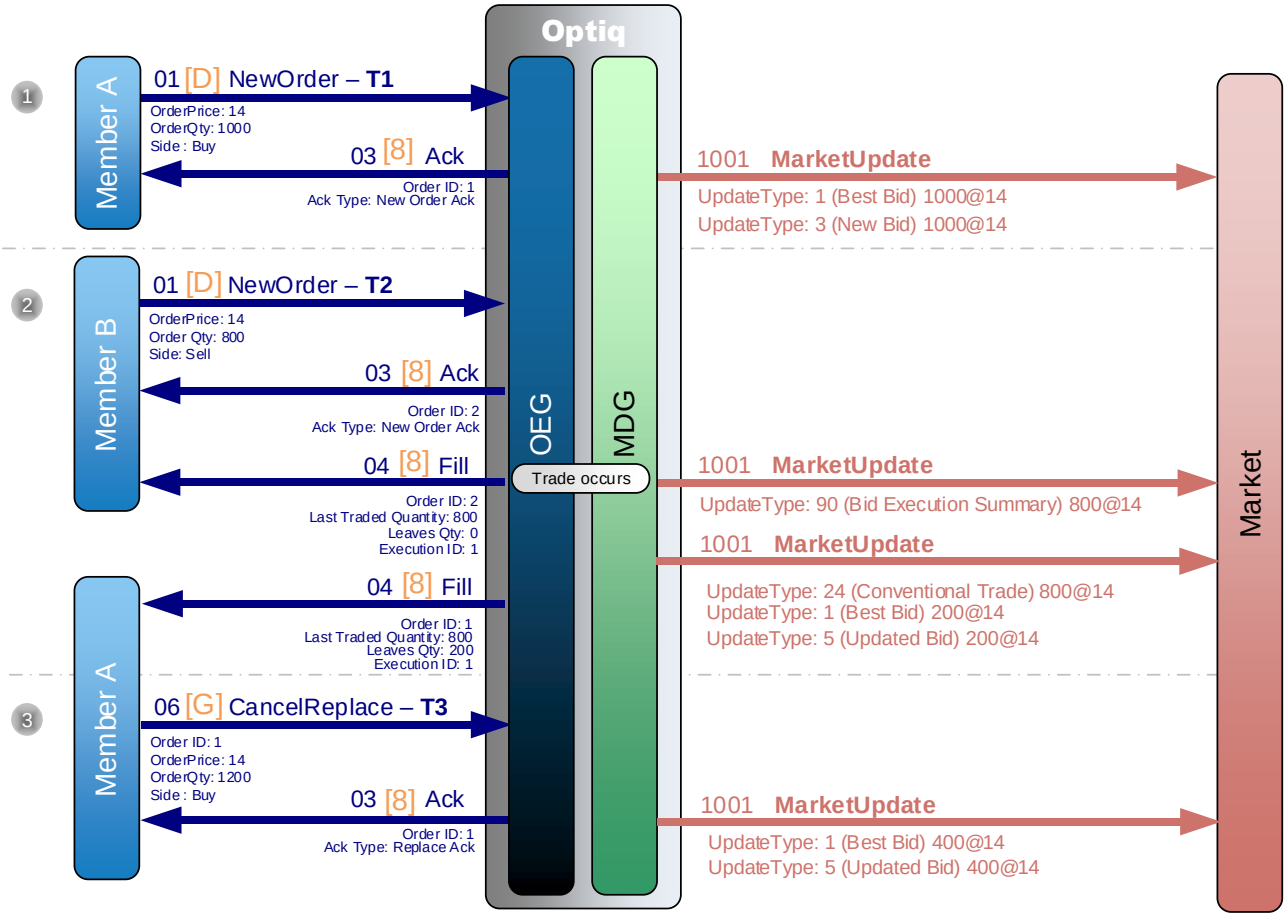
OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order modification.

MDG sends a public **MarketUpdate** (1001) message to update the limits.

Note: In case of a change of an order ownership, i.e. when the **CancelReplace** (06) (FIX G) message is sent from another OE Session, it will follow the same kinematic (no message is sent to the previous owner of the order). For more information about Ownership, please see section Ownership Request

2.4.2 Modifying a Partially Matched Order

Symbol Index: M1					
		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
T1	1000	14	14	800	T2



① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 1,000 and a price of 14.
OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to update the limit.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 800 and a price of 14.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately matches the first order, and OEG sends back a private **Fill** (04) (FIX 8) message to each member to notify of the trade execution.

A public **MarketUpdate** (1001) message is immediately sent to the market for the Execution Summary.

Only then, public **MarketUpdate** (1001) messages are sent to the market for the Trades and the Limits.

③ Later, Member A sends a private **CancelReplace** (06) (FIX G) message to modify the quantity of the original Buy order. As the member wants the leaves quantity to be equal to 400 after the modification, the member indicates a quantity of 1,200 (as 800 have already matched, and leaves quantity is 200).

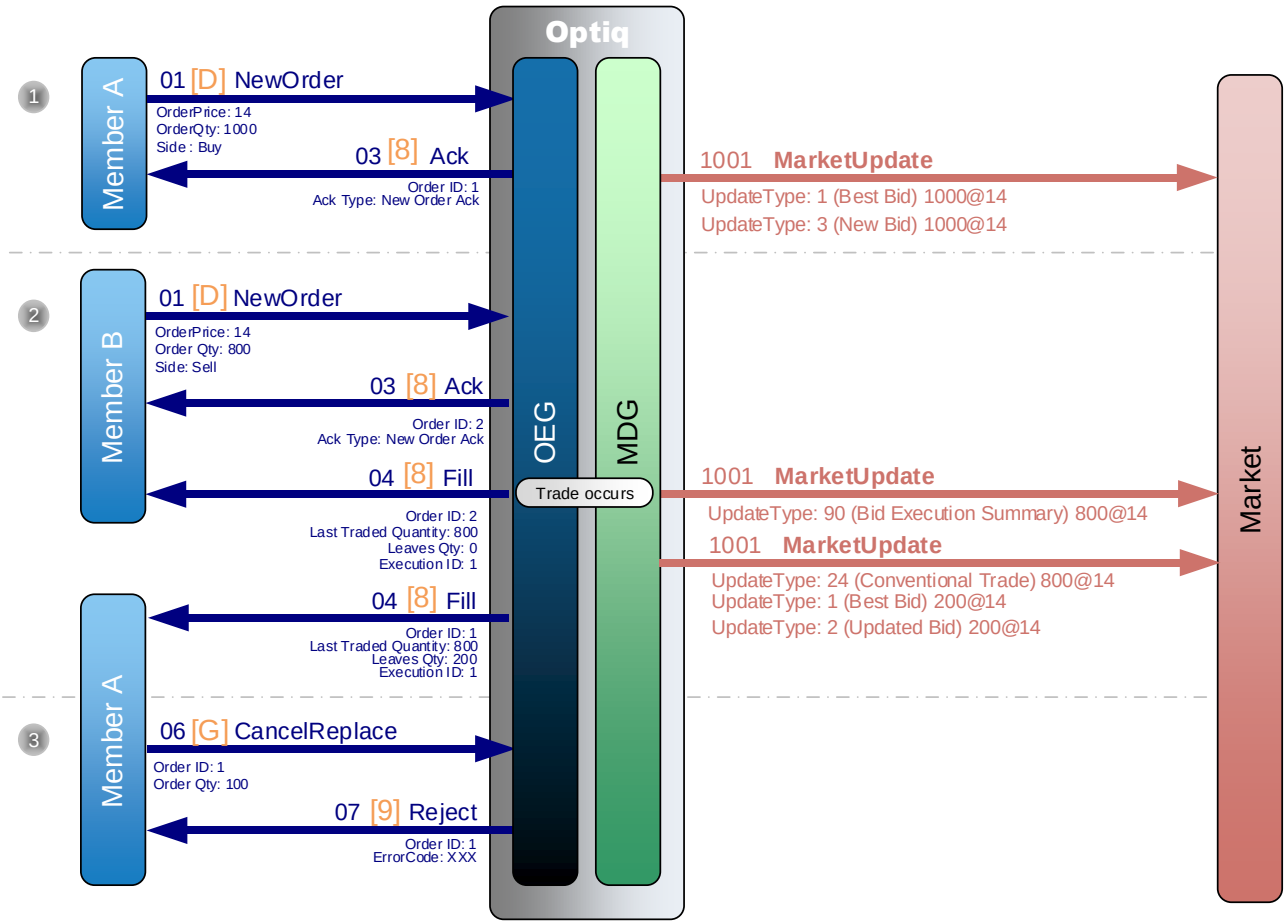
OEG sends back a **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order modification.

A public **MarketUpdate** (1001) message is sent to update the limit.

Note: Market Data kinematics are based on the matching scenario Explicit vs Explicit in an Outright. For the full list of scenarios see section "Trading Kinematics".

There is **no** **MarketUpdate** (1001) for the entry of the second order as it is immediately matched.

2.4.3 Rejected Modification



① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 1000 and a price of 14.
OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to update the limits.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 800 and a price of 14.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately matches the first order and OEG sends back a private **Fill** (04) (FIX 8) message to each member to notify the trade execution.

A public **MarketUpdate** (1001) message is immediately sent to the market for the Execution Summary.

Only then, public **MarketUpdate** (1001) messages are sent to the market for the Trades and the Limits.

③ Later, Member A sends a private **CancelReplace** (06) (FIX G) message to modify the quantity of the original Buy order. The member indicates a quantity of 100.

OEG sends back a private **Reject** (07) (FIX 9) message to reject the replace operation as the quantity to be modified for an individual order must always be larger than the one already present in the book. As such, the remaining quantity of 200 stays in the order book.

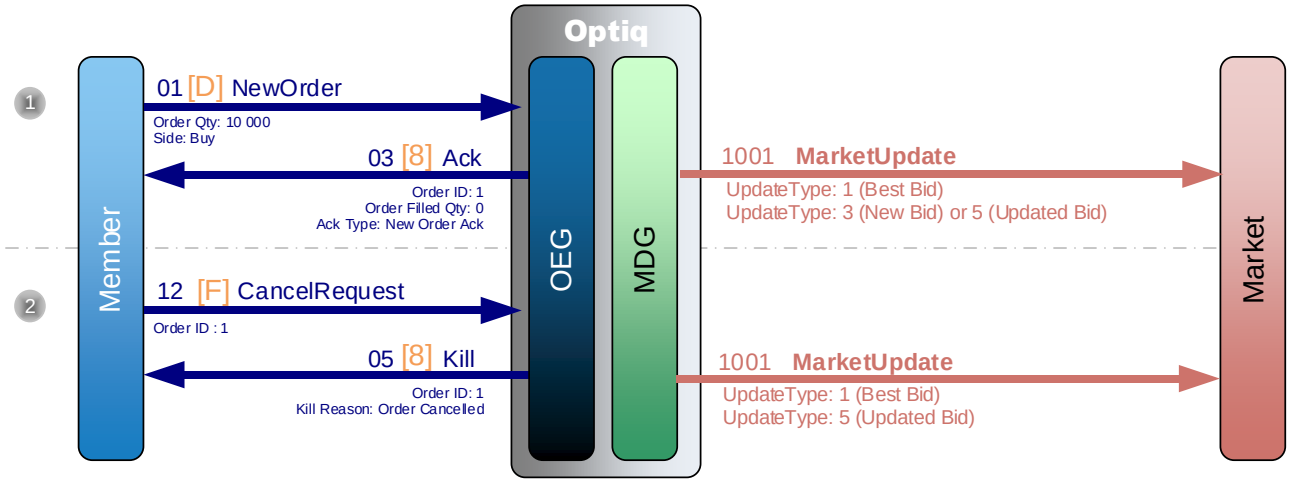
Note: If the member attempts to change the quantity of an order to a value less or equal to the quantity already traded, the order modification will be rejected. In this example, new quantity of 800 will be rejected, a new quantity of 801 will be accepted.

Market Data kinematics are based on the matching scenario Explicit vs Explicit in an Outright. For the full list of scenarios see section "Trading Kinematics".

There is **no** **MarketUpdate** (1001) for the entry of the second order as it is immediately matched.

2.5 CANCELLING AN ORDER

2.5.1 Cancelling an Unmatched Order



① A Member sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 10,000.
OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

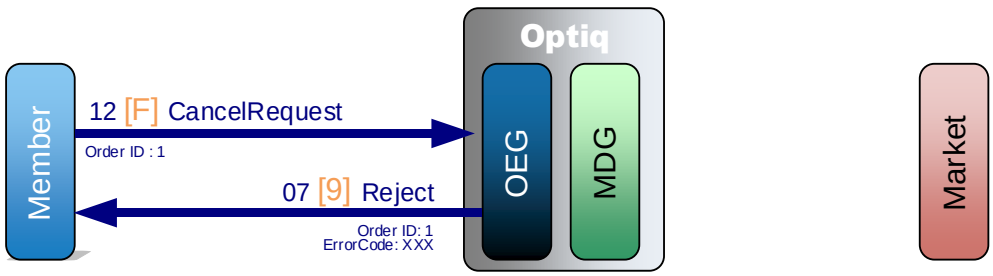
The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limit.

② Later the same Member sends a private **CancelRequest** (12) (FIX F) message to cancel the previously entered order.

OEG sends back a private **Kill** (05) (FIX 8) message to confirm that the order request has been cancelled.

A public **MarketUpdate** (1001) message is sent to the market to update the limits.

2.5.2 Rejected Order Cancellation



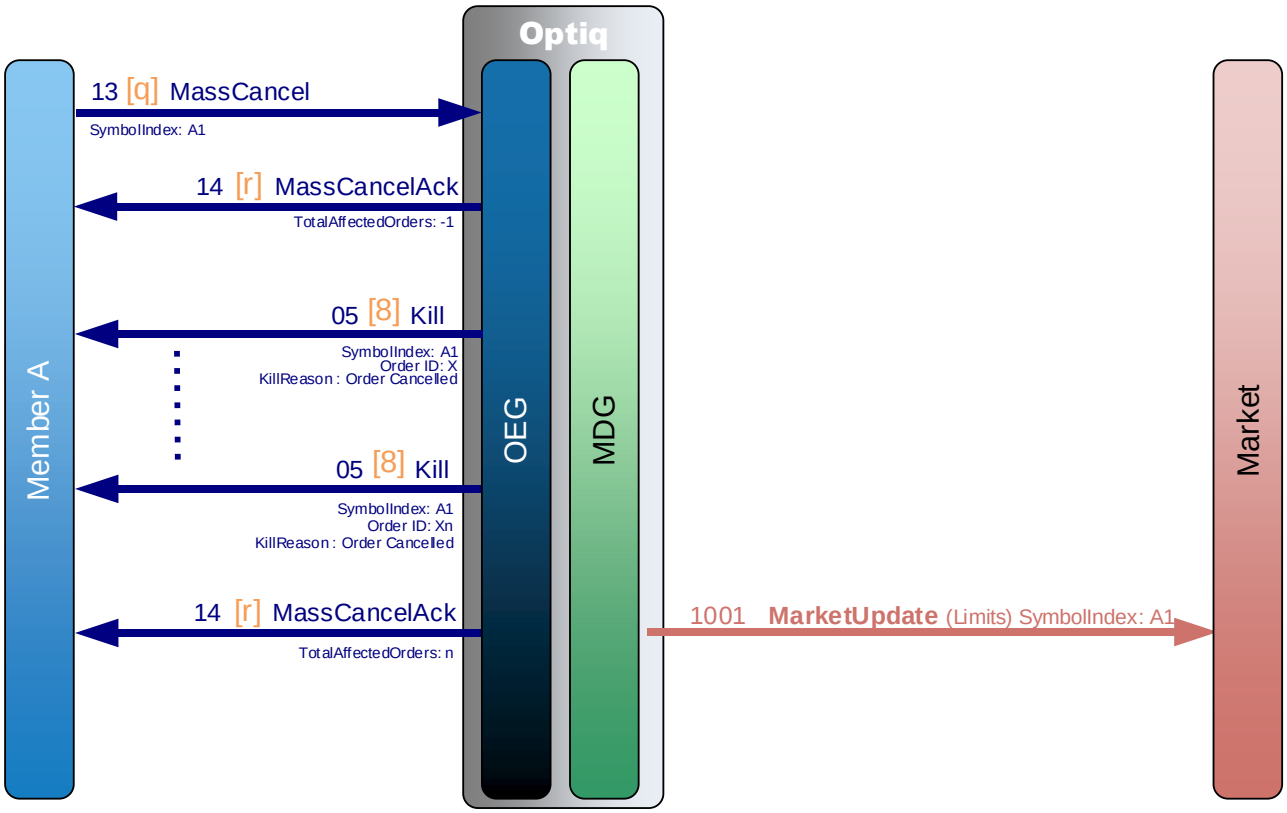
A Member sends a private **CancelRequest** (12) (FIX F) message to cancel an order that has already matched.

OEG sends back a private **Reject** (07) (FIX 9) message to reject the cancellation with an *Error Code*. The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*. As this is a case of functional rejection, **Reject** (07) message contains a system assigned Order ID.

No message is sent to the Market.

2.5.3 Mass Cancellation

2.5.3.1 Mass Cancellation for an Instrument



Member A sends a private **MassCancel** (13) (FIX q) message to cancel orders for an instrument A1.

OEG sends back a private **MassCancelAck** (14) (FIX r) message followed by a private **Kill** (05) (FIX 8) message for each killed order; the mass cancellation process ends with the sending of a new private **MassCancelAck** (14) (FIX r) message identifying the total number of orders cancelled.

A public **MarketUpdate** (1001) message is sent to the market to update the limits.

2.5.3.2 Mass Cancellation for a Contract

A1

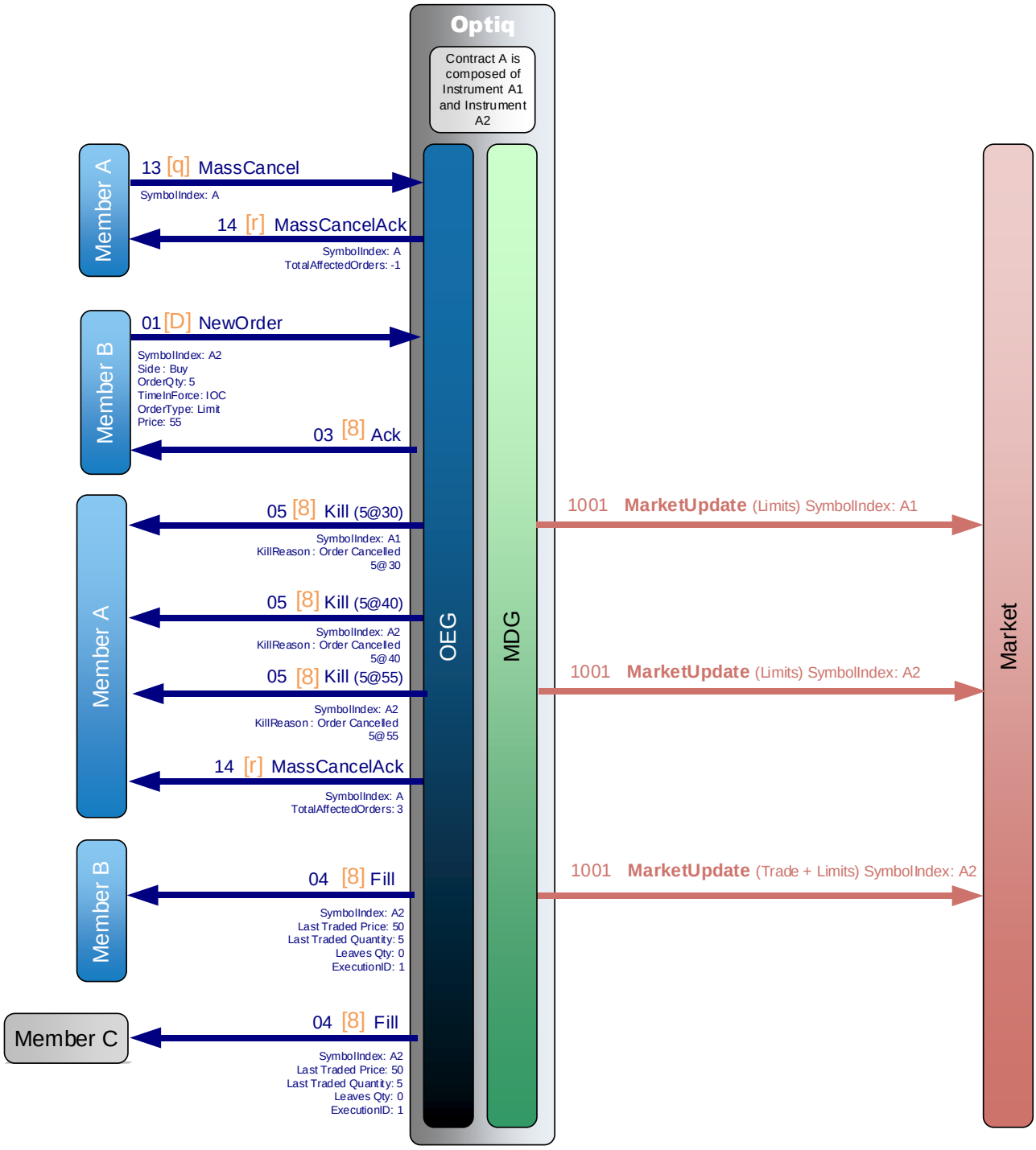
Outright Instrument

		Bid	Offer		
Member	Qty	Price	Price	Qty	Member
D	5	15	20	5	C
			30	5	A

A2

Outright Instrument

		Bid	Offer		
Member	Qty	Price	Price	Qty	Member
D	5	35	40	5	A
			50	5	C
			55	5	A



Member A sends a private **MassCancel** (13) (FIX q) message to cancel orders for a Contract A, which contains instruments A1 and A2. Member A has orders in both instruments on the Offer side.

Immediately following this Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order on instrument A2, which under other conditions would match one of the orders of Member A.

As Mass Cancellation for Derivatives is processed first and as a complex synchronous instruction, the cancellation of orders for Member A is fully processed first, with all orders on the affected instruments cancelled.

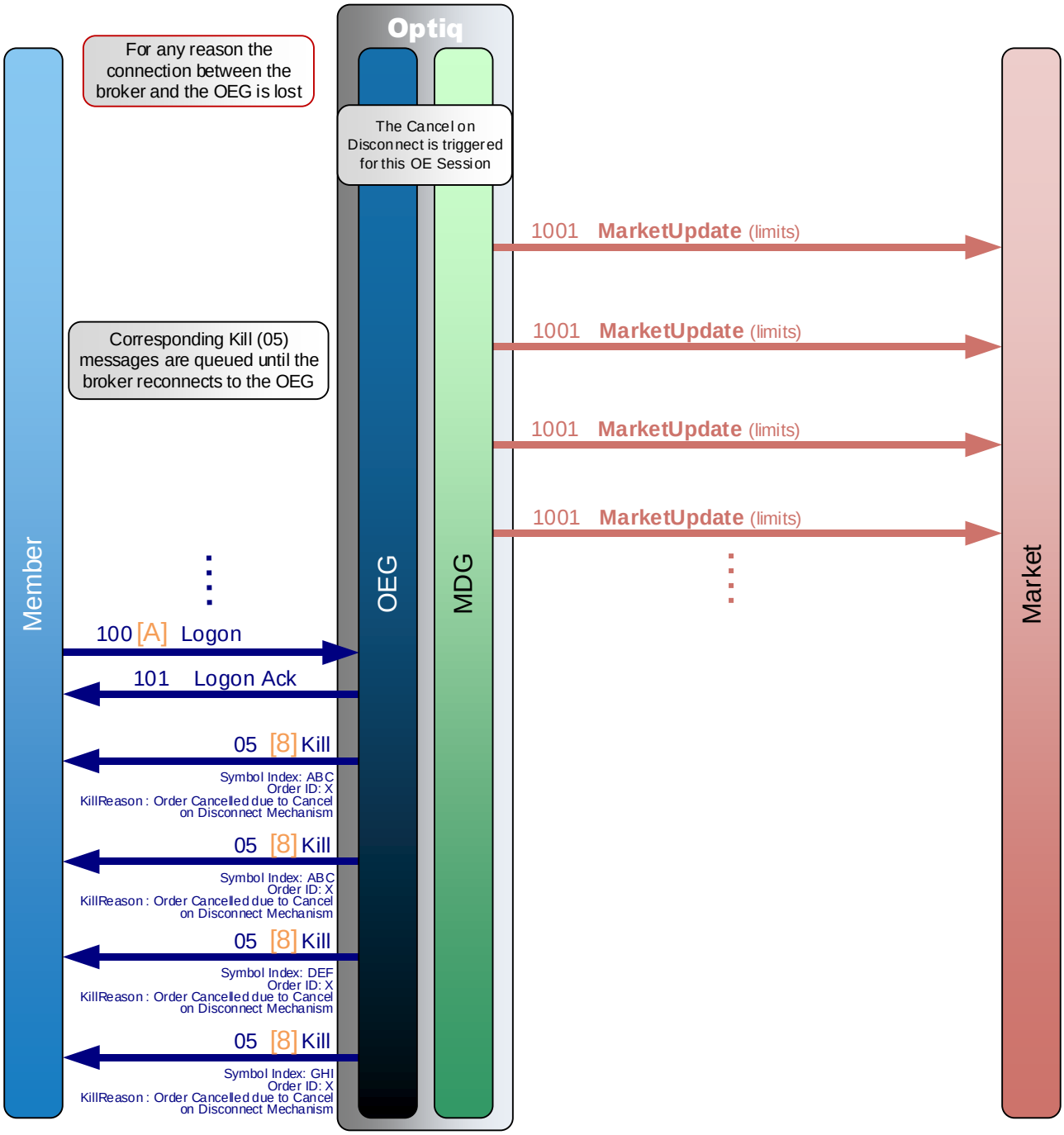
OEG sends back a private **MassCancelAck** (14) (FIX r) message followed by a private **Kill** (05) (FIX 8) message for each killed order. The mass cancellation process ends with the sending of a new private **MassCancelAck** (14) (FIX r) message identifying the total number of orders cancelled.

As Member A no longer has any orders in the book, matching of the newly entered Buy order for Member B occurs with the order of Member C. For which the **Fill** (04) (FIX 8) messages are sent to each participant of the executed trade.

As a result of both the cancellation of orders of Member A, and the trade between Member B and C - public **MarketUpdate** (1001) messages are sent to the market to update the limits for instruments A1 and A2.

Note: MassCancel (13) FIX (q) message applies to all the orders sent by the Firm ID or the combination of Firm ID & Execution Within Firm ShortCode, independent of the Logical Access that submitted the Mass Cancel request. (E.g.: A MassCancel message sent from a Logical Access 1 to cancel orders for Short Code 1 will cancel all the orders of the firm for that Short Code submitted from all the Logical Accesses of that firm).

2.5.4 Cancel on Disconnect Mechanism



The diagram represents a generic case of loss of connection (physical) between a client and a partition.

When a connection is lost between the member and OEG, for any reason, the Cancel on Disconnect (CoD) mechanism is triggered for all OE Sessions concerned by the connection outage. Once the mechanism is triggered, all live orders not flagged to be persisted and belonging to the corresponding OE Session(s) are immediately cancelled for their remaining quantity, regardless of order type and validity type.

For each order cancelled a public **MarketUpdate** (1001) message is sent to the market to update the limits.

For each cancelled order a **Kill** (05) (FIX 8) message is generated and queued until the client reconnects during the trading phases of the same trading day.

When the Member reconnects with a **Logon** (100) (FIX A) message, if the logon is successful, the OEG sends back a **LogonAck** (101) (FIX A) message.

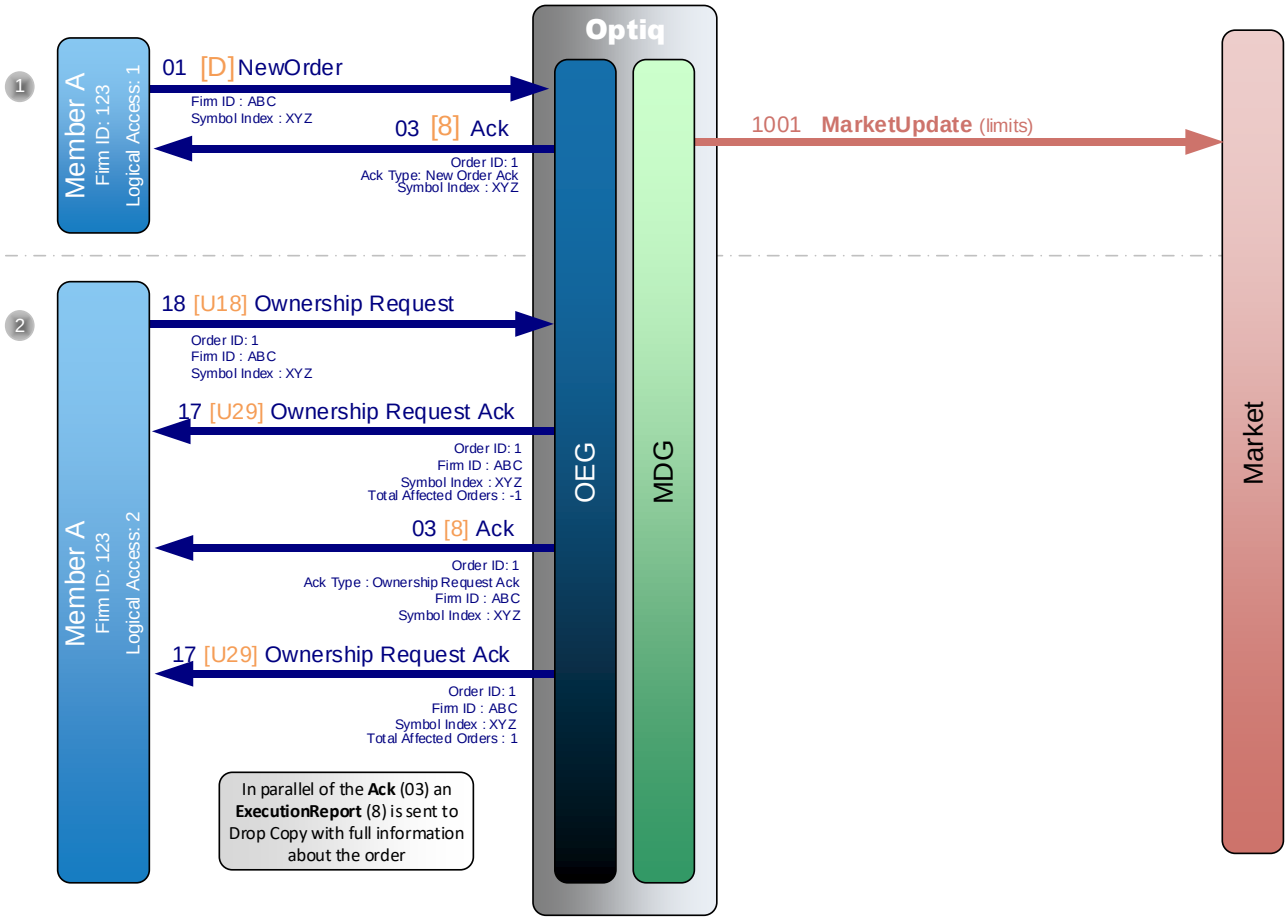
Once the connection is re-established, the Member immediately receives the **Kill** (05) (FIX 8) messages that have been queued.

Kill (05) (FIX 8) messages for the day orders are not persisted from one day to another.

Note: Scope of Cancel on Disconnect only includes orders sent during the current day, whether through single order submission or through 'Quotes'. Orders entered during a previous business day are not in scope of Cancel on Disconnect and are not impacted.

2.6 OWNERSHIP REQUEST

2.6.1 Ownership request for a specified order ID



① Member A from Logical Access 1 sends a private **NewOrder** (01) (FIX D) message to enter a new order.

OEG sends back an **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order (*Order ID* = 1).

The order enters into the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limits.

② Member A from Logical Access 2 sends an **OwnershipRequest** (18) (FIX U18) to request the ownership of the previous order (*Order ID* = 1) sent by Logical Access 1.

OEG sends back an **OwnershipRequestAck** (17) (FIX U29) message to Logical Access 2, to confirm the reception of the request (with *Total Affected Orders* = -1).

OEG sends back an **Ack** (03) (FIX 8) message to Logical Access 2, to give the detail of the order (*Order ID* = 1). In parallel of the **Ack** (03) (FIX 8) an **ExecutionReport** (8) is sent to Drop Copy with full information about the order.

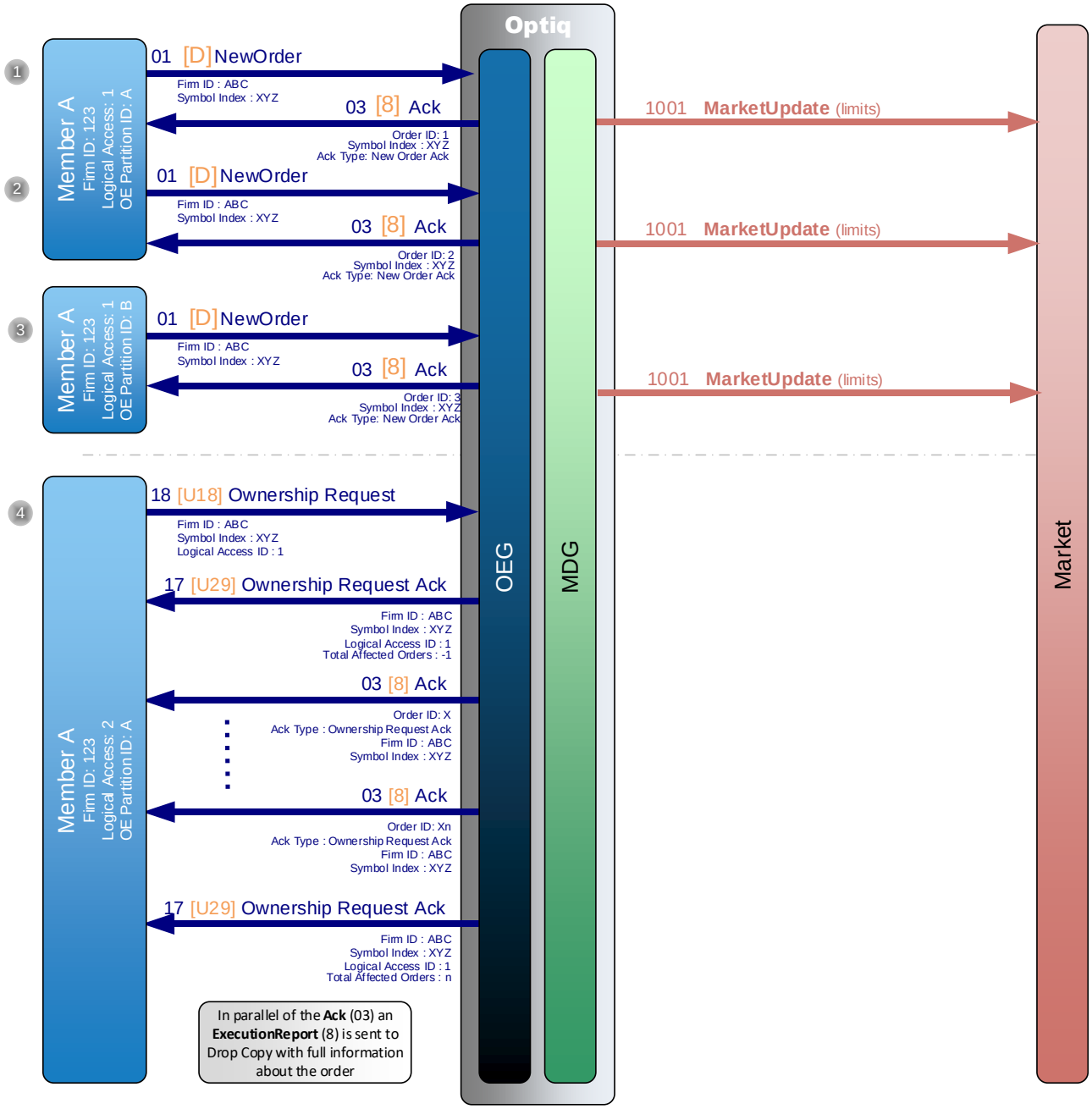
OEG sends back another **OwnershipRequestAck** (17) (FIX U29) message to Logical Access 2, to confirm the successful change of ownership of the order (*Order ID* = 1) from Member A's Logical Access 1 to

Member A's Logical Access 2 (*Total Affected Orders* = 1). Logical Access 1 does not receive any messages of this exchange and following the transfer of ownership all unsolicited messages for the affected order are sent to Logical Access 2.

Note: OwnershipRequest (18) does not apply for Quotes.

All specified Logical Access IDs and OE Sessions must belong to the same Firm.

2.6.2 Ownership request for all orders belonging to a Logical Access or OE Session



Logical Access ID and *OE Session ID* are provided by clients in the **Logon** (100) message.

OwnershipRequest (18) does not apply for Quotes.

① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new order on instrument XYZ.

The order is entered through the OE session 1A (Logical Access ID = 1, OE Partition ID = A).

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order (*Order ID* = 1).

The order enters into the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limits.

② Member A sends another private **NewOrder** (01) (FIX D) message to enter a new order instrument XYZ.

The order is entered through the OE session 1A (Logical Access ID = 1, OE Partition ID = A).

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order (*Order ID* = 2).

The order enters into the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limits.

③ Member A sends a private **NewOrder** (01) (FIX D) message to enter a new order instrument XYZ.

The order is entered through the OE session 1B (Logical Access ID = 1, OE Partition ID = B).

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order (*Order ID* = 3).

The order enters into the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limits.

④ Member A sends an **OwnershipRequest** (18) (FIX U18) to request the ownership of the orders of the *Logical Access ID* = 1 for the instrument XYZ.

The request is entered through the OE session 2A (Logical Access ID = 2, OE Partition ID = A).

OEG sends back an **OwnershipRequestAck** (17) (FIX U29) message to OE session 2A, to confirm the reception of the request (with *Total Affected Orders* = -1).

OEG sends back an **Ack** (03) (FIX 8) message to OE session 2A for each order (*Order ID* = 1, 2 and 3) for the instrument XYZ that are owned by the Logical Access 1.

In parallel of each **Ack** (03) (FIX 8) message an **ExecutionReport** (8) is sent to Drop Copy with full information about the order, with the field *ExecType* (150) set to k = Ownership Request Ack

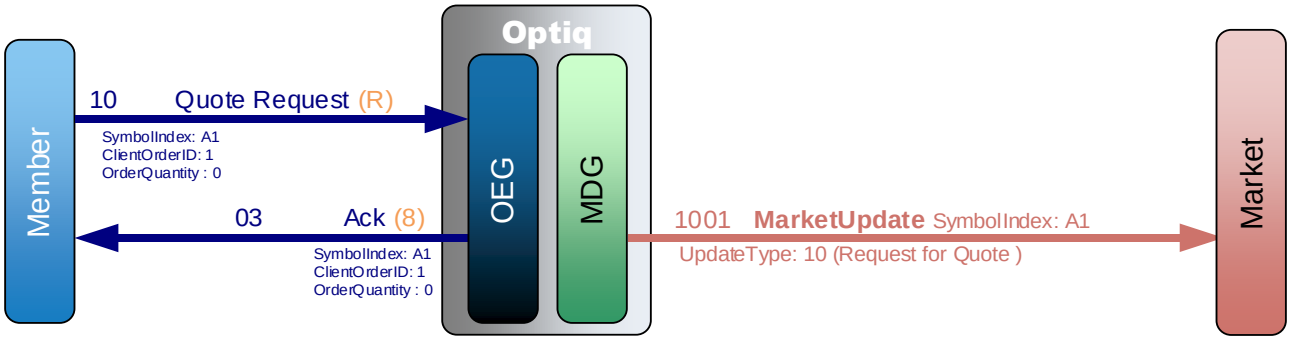
OEG sends back another **OwnershipRequestAck** (17) (FIX U29) message to OE session 2A to confirm the successful change of ownership of the orders belonging to the Logical Access ID = 1 for the instrument XYZ. The ownership of *Order ID* = 1 and 2 from Member A's OE session 1A and *Order ID* = 3 from Member A's OE session 1B transfer to Member A's OE session 2A (*Total Affected Orders* = 3).

OE session 1A and 1B do not receive any messages of this exchange, and following the transfer of ownership all unsolicited messages for the affected orders are sent to OE session 2A.

Note: All specified Logical Access IDs and OE Sessions must belong to the same Firm.

2.7 REQUEST FOR QUOTE

2.7.1 Request For Quote Accepted

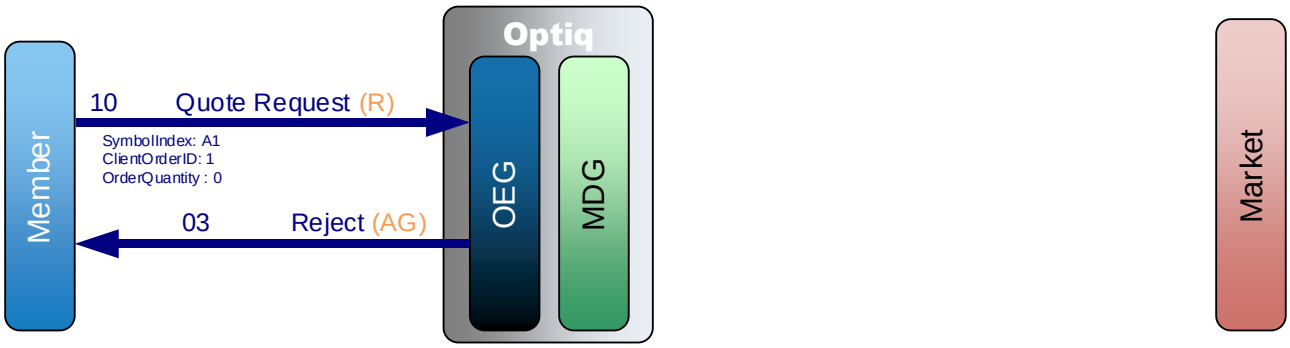


A member sends a private **QuoteRequest** (10) (FIX R) message to broadcast a request for liquidity to the market via MDG.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the request.

A public **MarketUpdate** (1001) message is immediately sent to the market with the Update type 10 = 'Request for Quote'. As the Order quantity in the OEG message was set to zero, the value is set to Null value.

2.7.2 Request For Quote Rejected



A member sends a private **QuoteRequest** (10) (FIX R) message to broadcast their request for liquidity to the market via MDG.

If the request is rejected OEG sends back a private **Reject** (07) (FIX AG) message with an Error Code. The reason of the rejection can be found using the Error Code within the Error Code List file (.csv).

No message is sent to the Market.

3. UNSOLICITED MESSAGES

3.1 ASYNCHRONOUS MESSAGES

3.1.1 Statistics Message

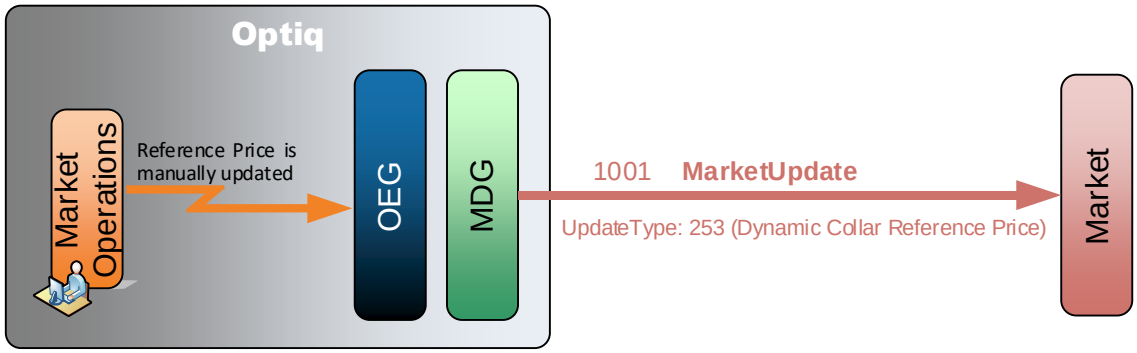
The public **Statistics** (1009) message is sent to the market after each trade, it includes only the information that needs to be updated. It can include minimum and maximum traded prices for daily, yearly and lifetime periods along with the cumulative volume since the start of the trading day and the percentage of variation of the traded price versus the last reference price.

3.1.2 Automatic IMP Calculation

A public **PriceUpdate** (1003) message is sent during the order collection period in real time to the market whenever the IMP Price or quantity have changed.

3.2 ACTIONS PERFORMED BY MARKET OPERATIONS

3.2.1 Reference Price Update

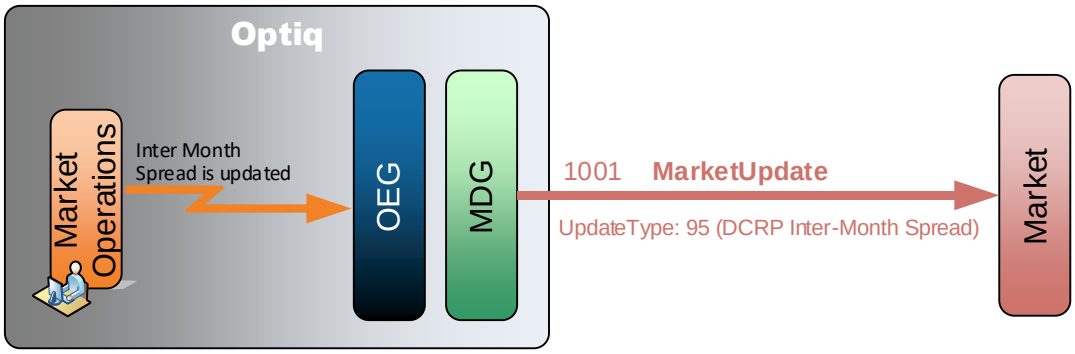


Market Operations send a private command to Optiq to update the reference price on the given instrument.

Optiq sends a public **MarketUpdate** (1001) message to broadcast the new prices.

Note: Only applicable for a contract where the reference price origin is set to Opening Call Price. Data available in **ContractStandingData** (1013).

3.2.2 Inter-Month Spread Update

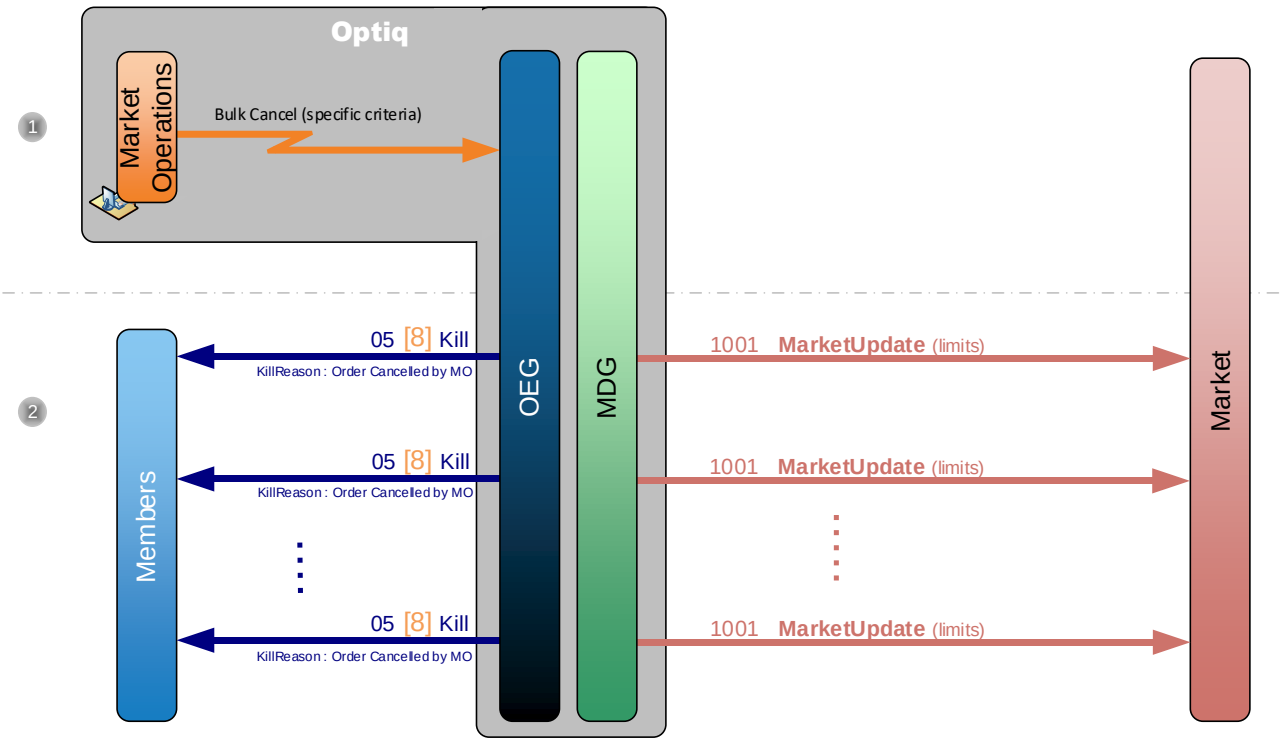


Market Operations send a private command to Optiq to update the Inter-Month Spread on a given instrument.

Optiq sends a public **MarketUpdate** (1001) message to broadcast the new Inter-Month Spread.

Note: Only applicable for a contract where the reference price origin is set to Future Market Price. Data available in **ContractStandingData** (1013).

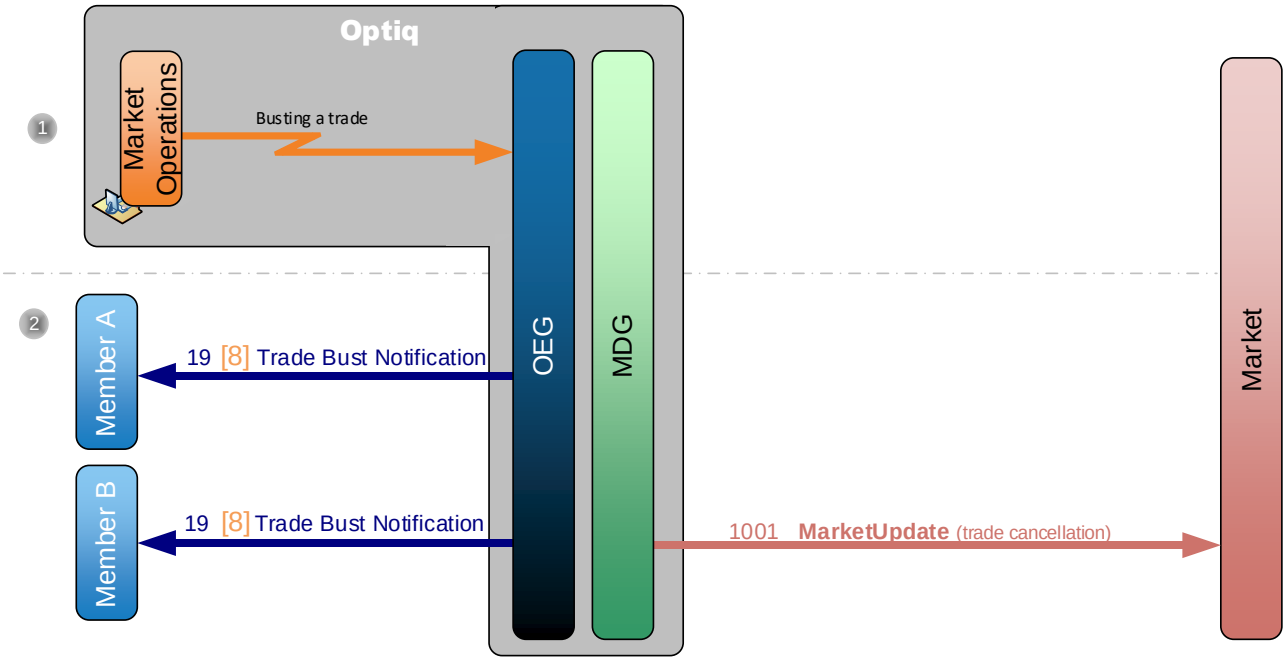
3.2.3 Bulk Order Cancellation by Market Operations



① Market Operations cancel orders matching a specified criterion.

② Optiq sends a private **Kill** (05) (FIX 8) message for each cancelled order to the member who entered the order, and public **MarketUpdate** (1001) messages to the market to update the limits.

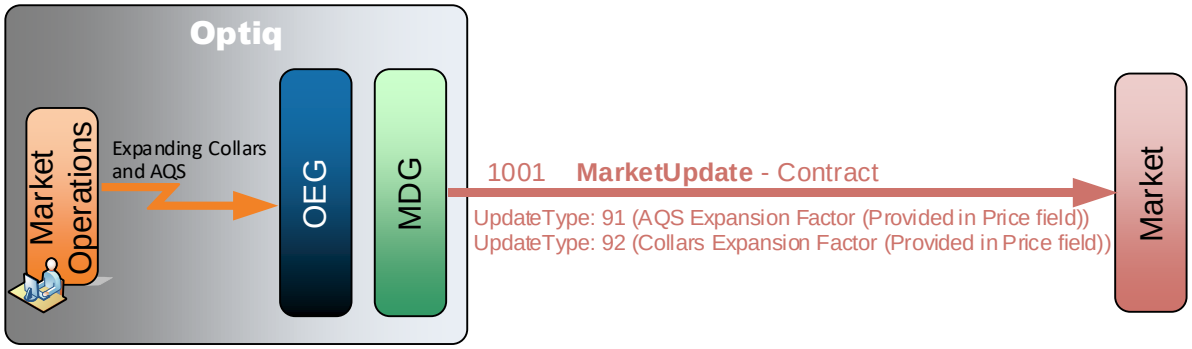
3.2.4 Trade Cancellation



① Market Operations busts a trade.

② Optiq sends a private **TradeBustNotification** (19) (FIX 8) message for the cancelled trade to the members who entered the orders and a public **MarketUpdate** (1001) message to remove the cancelled orders.

3.2.5 Triggering of Fast Market



In case of Fast Market declared by Market Operations, a command is sent to Optiq to expand Collars and AQS spreads for a contract.

Optiq sends a public **MarketUpdate** (1001) message to the market to indicate the expansion factor to apply for Collars and AQS.

Note: Once Fast Market is withdrawn by Market Operations, a public **MarketUpdate** (1001) message is sent to the Market with Expansion Factors set to one (1).

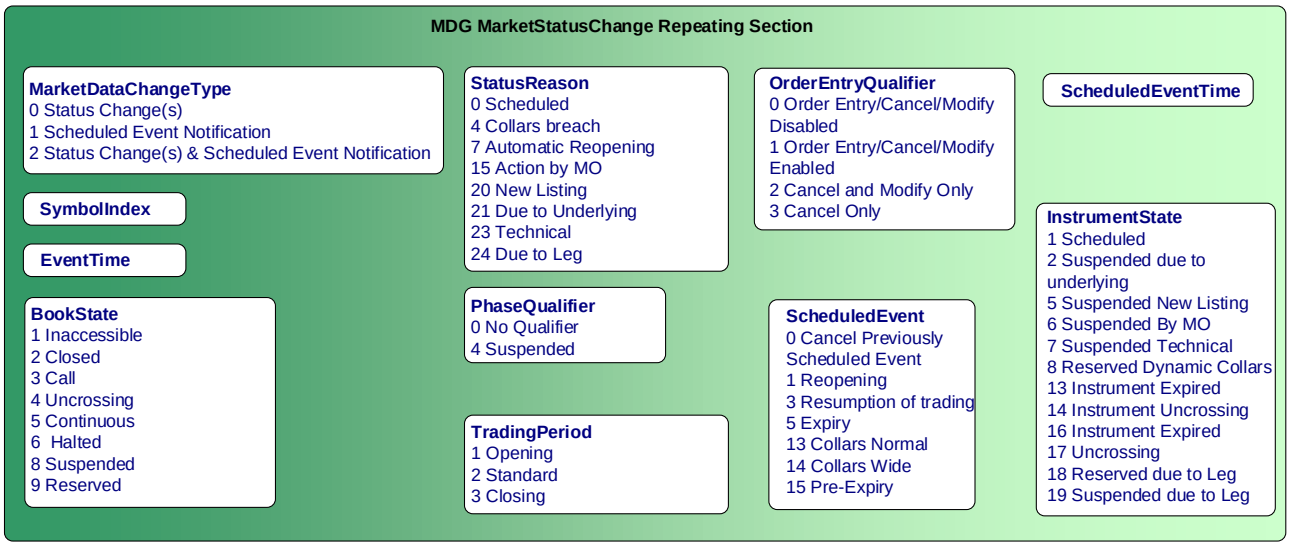
4. MARKET STATUS CHANGES

This section is dedicated to all market publications that deal with changes of Market Status on Euronext Derivatives markets, which are communicated via the **MarketStatusChange** (1005) message.

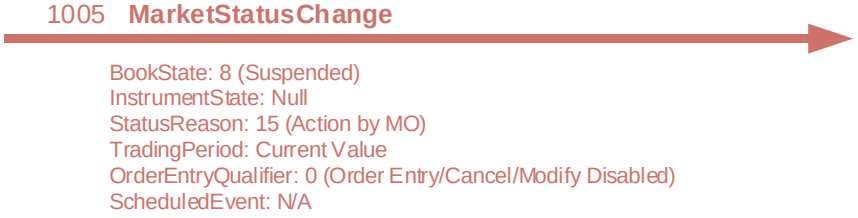
The Market Status of a contract or an instrument can be determined using the following fields:

- **Book State:** Market State of the Contract
- **Instrument State**: Market State of the Instrument
- **Status Reason**: Reason of the state change
- **Phase Qualifier**: Specifics during a trading phase that do not impact the Instrument State or Book State
- **Trading Period**: indicates the trading period
- **Order Entry Qualifier**: Describes whether order entry is allowed for the instrument or the contract
- **Scheduled Event**: Market Event notification
- **Scheduled Time**: Scheduled Event associated time if required

The possible Market Status values on Euronext Derivatives are as follows:



In the following Market Status change example, a contract is manually suspended by Market Operations with Order entry disabled:

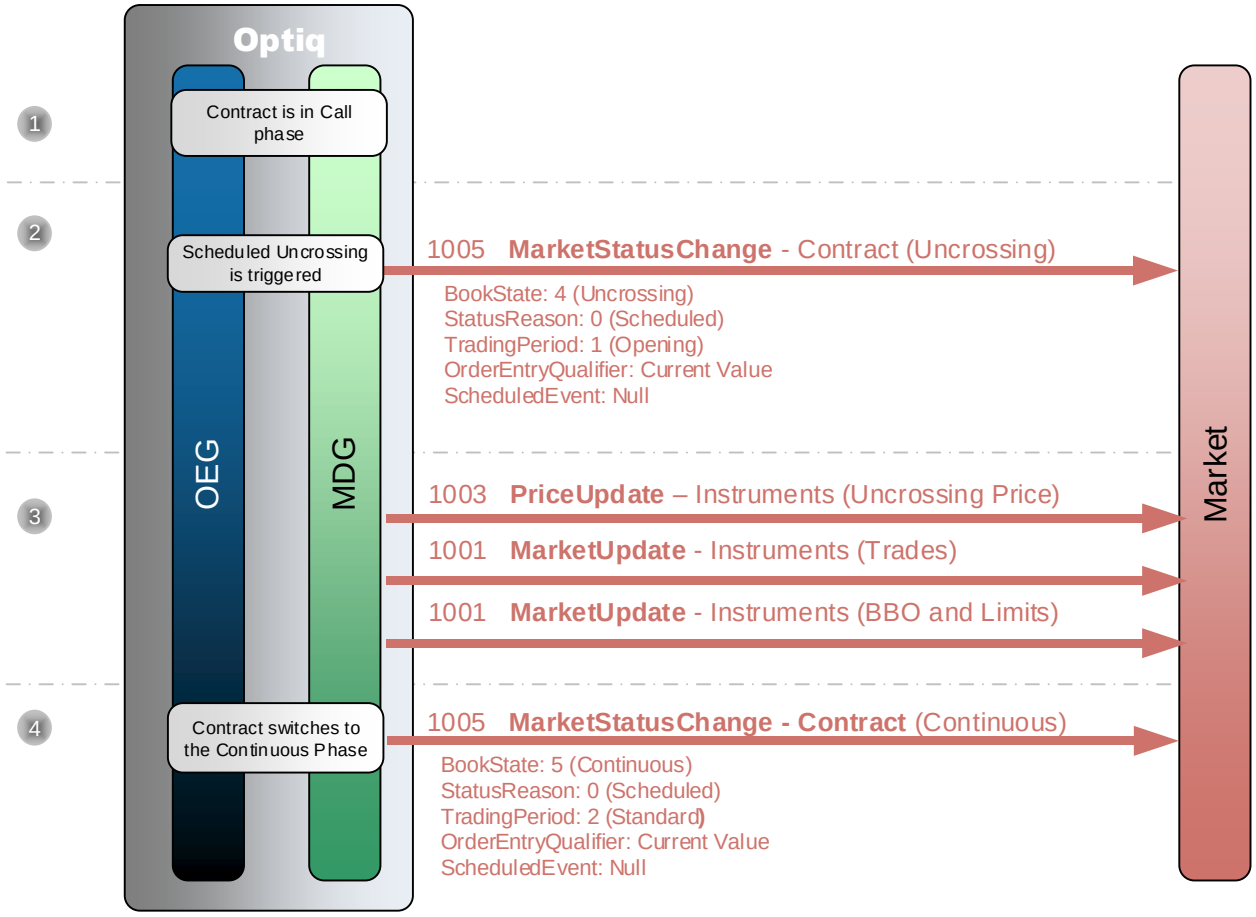


- InstrumentState is set to Null. It means that the State of the instrument should follow the contract state which is stored in book state.
- Also, some values of the **MarketStatusChange** (1005) message are set to 'Current Value'. It means that the value is the same as the one sent in the previous **MarketStatusChange** (1005) message.

4.1 AUTOMATIC MARKET STATUS CHANGES

For readability purposes, order entry messages are not shown in the diagrams.

4.1.1 Scheduled Uncrossing



① The Contract is in a Call trading phase as defined in the **TimeTable** and by the pattern associated to this Contract.

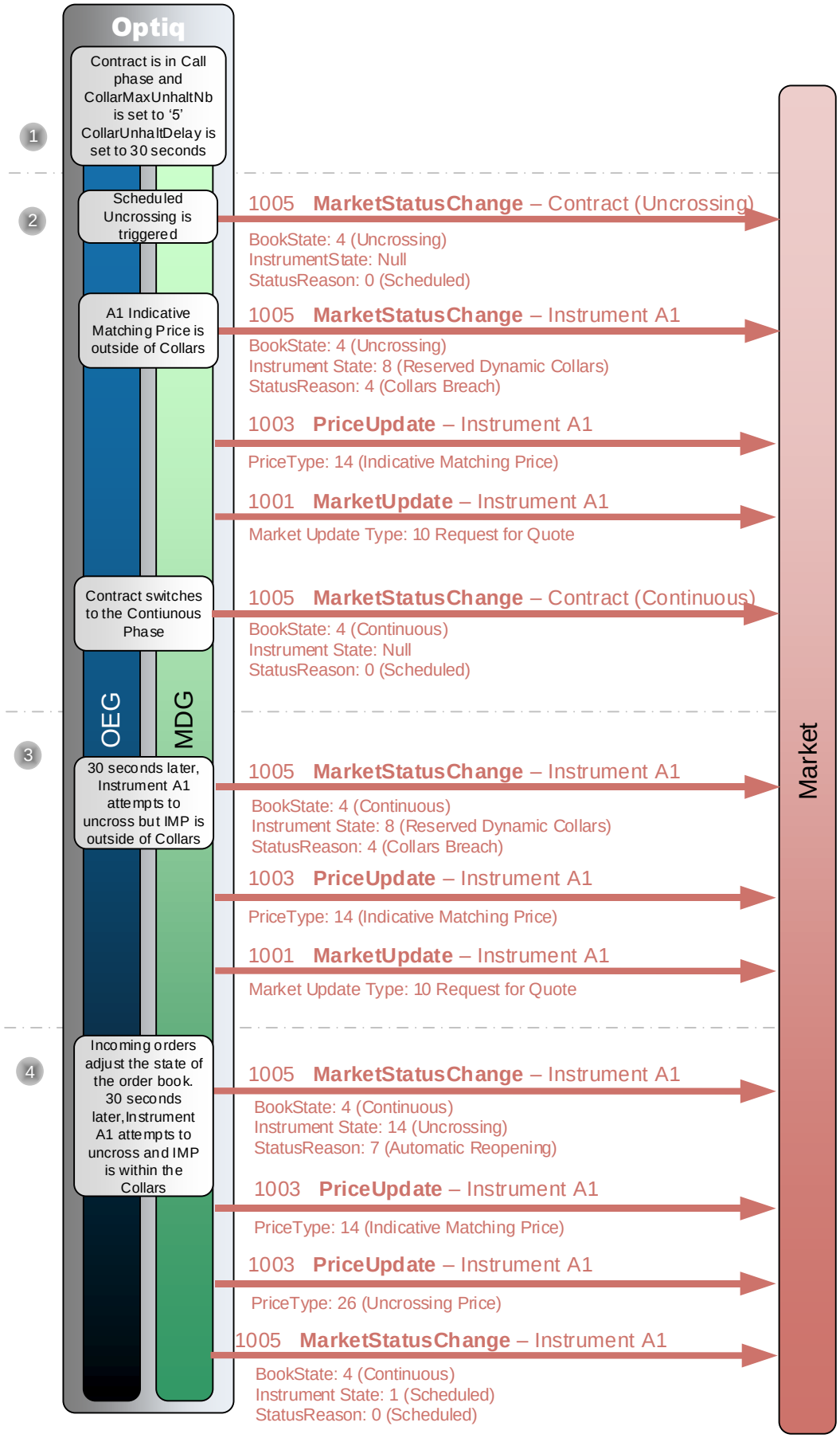
② When the Uncrossing is triggered for the contract a public **MarketStatusChange** (1005) message is disseminated to the market.

③ Right after the status change, a public **PriceUpdate** (1003) message is sent to the market for each Instrument of the contract with the uncrossing price and the quantity at which the uncrossing is performed.

MarketUpdate (1001) message is sent accordingly for each Instrument for Trades, BBO and Limits.

④ Upon contract entering into the Continuous phase, a public **MarketStatusChange** (1005) message is sent to the market to indicate that the Contract is now in a continuous phase.

4.1.2 Trade Price Validation (TPV) triggered at Uncrossing



① The contract is in a Call trading phase as defined in the **TimeTable** and by the pattern associated to this contract. For this example, the number of uncrossing attempts when TPV is triggered is set to 2 and the duration between each attempt is set to 30 seconds as defined in the Contact Standing Data. For the values set for use in Trade Price Validation for number of attempts and duration of TPV period clients should refer to the Contract Standing Data.

② When the Uncrossing is triggered for the Contract, the Uncrossing Price for the instrument A1 lies outside Dynamic Collars.

A public **MarketStatusChange** (1005) message is disseminated to the market to indicate the start of the Uncrossing for the Contract.

A second public **MarketStatusChange** (1005) message is immediately sent to the market to indicate the reservation of Instrument A1.

Right after the status change, a public **PriceUpdate** (1003) is sent to the market for instrument A1 with the Indicative Matching Price.

Then, a public **MarketUpdate** (1001) message is sent to the market to request Liquidity in instrument A1 (Request for Quote).

A public **MarketStatusChange** (1005) is sent to the market for the Contract, to inform that it switched to Continuous phase, as defined by the pattern. At that moment instrument A1 remains in the Reserved state.

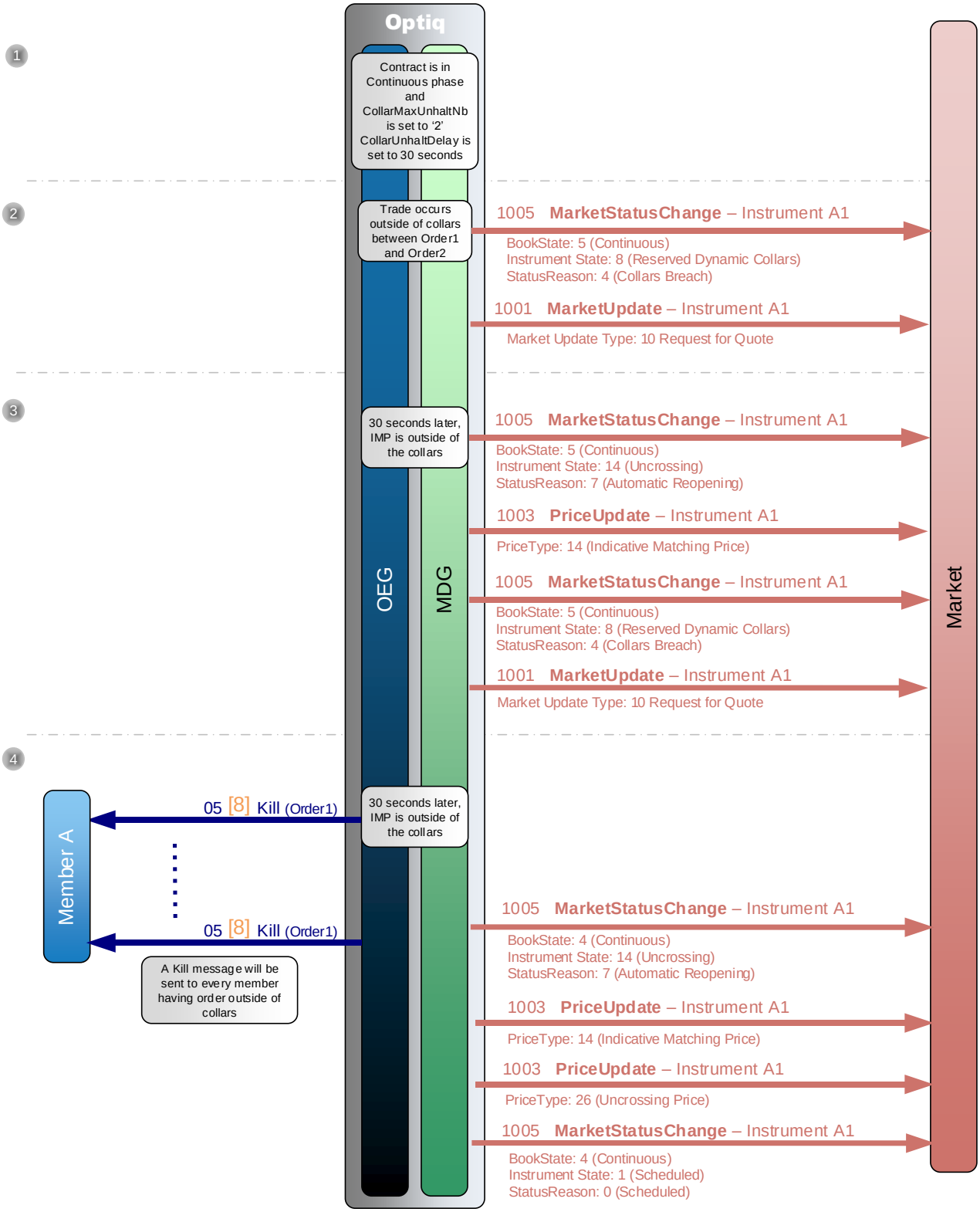
③ Instrument A1 attempts to uncross but the Uncrossing Price is still outside of Collars, a public **MarketStatusChange** (1005) is sent to the market to inform that the Instrument is still Reserved, following by a public **PriceUpdate** (1003) with the Indicative Matching Price and a **MarketUpdate** (1001) to request for Liquidity (Request for Quote).

④ Incoming orders adjust the Indicative Matching Price which fall now into Collars. Instrument A1 attempts to uncross and the Uncrossing Price lies within Collars, as such it exits its Reserved state. A public **MarketStatusChange** (1005) message is sent to the market to indicate that Instrument A1 switches to Uncrossing. Minor note, due to the state of the book, following this uncrossing no trades took place.

A public **PriceUpdate** (1003) is sent to the market for instrument A1 with the Uncrossing Price.

Then, a public **MarketStatusChange** (1005) is sent when the Instrument A1 switches to Continuous. From this point on, instrument re-joins the TimeTable of the Contract.

4.1.3 Trade Price Validation (TPV) triggered at Continuous



① The contract is in a Continuous trading phase as defined in the **TimeTable** and by the pattern associated to this contract. For this example, the number of uncrossing attempts when TPV is triggered is set to 2 and the duration between each attempt is set to 30 seconds as defined in the Contact Standing Data.

For the values set for use in Trade Price Validation for number of attempts and duration of TPV period clients should refer to the Contract Standing Data.

② A trade occurs outside of collars, TPV mechanism is triggered and a **MarketStatusChange** (1005) is immediately sent to the market to indicate the reservation of Instrument A1.

Then, a public **MarketUpdate** (1001) message is sent to the market to request Liquidity in instrument A1 (Request for Quote).

③ Three minutes later (as defined in the field CollarUnhaltDelay from Contract Standing Data) the IMP is outside of collars. A public **MarketStatusChange** (1005) is sent to the market to indicate the Uncrossing attempt of Instrument A1 following by a public **PriceUpdate** (1003) with the Indicative Matching Price (IMP).

As the IMP is outside of collars, a public **MarketStatusChange** (1005) is sent to the market to indicate the reservation of Instrument A1 following by a public **MarketUpdate** (1001) to request for Liquidity (Request for Quote).

④ Three minutes later (as defined in the field CollarUnhaltDelay from Contract Standing Data), the IMP is still outside of collars. The number of maximum uncrossing attempts is reached (defined in the field CollarMaxUnhaltNb from Contract Standing Data).

Consequently, all orders outside of collars are cancelled and **Kill** (05) (FIX 8) messages are sent to members for these orders.

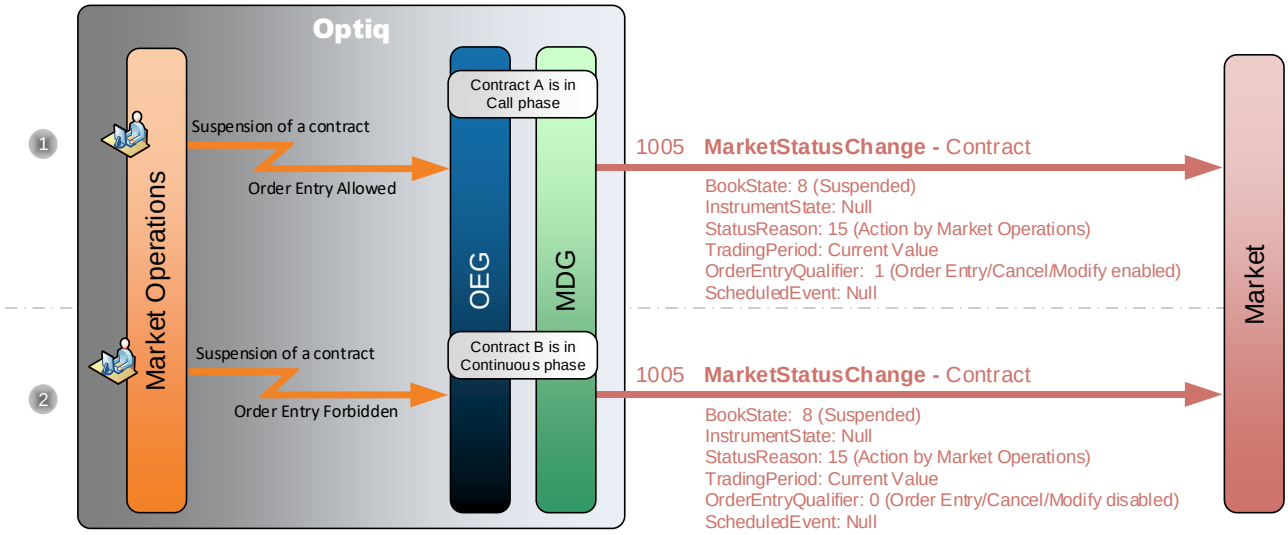
A public **MarketStatusChange** (1005) is sent to the market to indicate the Uncrossing attempt of Instrument A1 followed by a public **PriceUpdate** (1003) with the Indicative Matching Price.

Then a public **PriceUpdate** (1003) is sent to the market with the Uncrossing Price following by a **MarketStatusChange** (1005) to indicate that the instrument is now in Continuous phase.

Note: For readability purposes, incoming orders during Reservation are not in the diagrams.

4.2 Market Status Changes Due To Manual Intervention

4.2.1 Contract Suspended by Market Operations



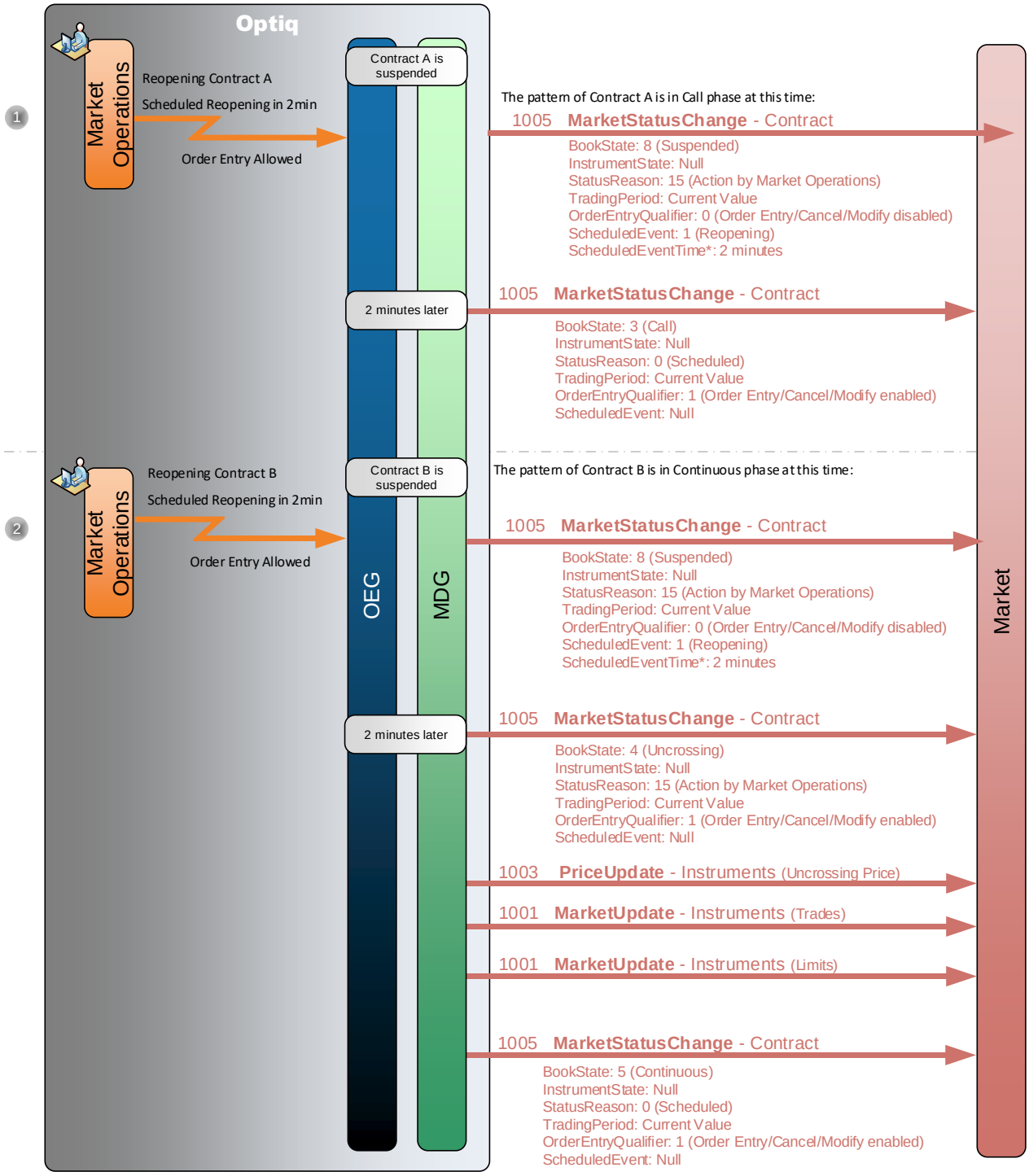
*Contract A is in a Call trading phase, and Contract B is in Continuous trading phases, according to the **TimeTable** and by the pattern associated to this contract.*

① Market Operations suspends the Contract A and set the order entry as enabled. This is communicated to the market by a public **MarketStatusChange** (1005) message for the Contract A.

② Market Operations suspends the Contract B and disables the order entry. This is communicated to the market by a public **MarketStatusChange** (1005) message for the Contract B. Order Entry Qualifier in this case is not related to the phase, and is populated based on the command by Market Operations.

Note: Book State is the State of the Contract. Instrument State is sent only if the State of the Instrument has been updated else it is set to Null.

4.2.2 Contract Reopened by Market Operations



** ScheduledEventTime: For readability purposes it is expressed as a duration in minutes but in reality this field is expressed as nanoseconds since Epoch.*

*Contract A is in a Call trading phase, and Contract B is in Continuous trading phase, according to the **TimeTable** and by the pattern associated to these contracts. Both contracts are suspended.*

① Market Operations schedule the reopening of Contract A in two minutes. As a result, a public **MarketStatusChange** (1005) message is sent to the market to notify the reopening of Contract A in two minutes.

Two minutes later, the Contract comes back to the Call phase, as defined by its pattern for this time. This is communicated to the market by a public **MarketStatusChange** (1005) message.

② Market Operations schedule the reopening of Contract B in two minutes. As a result, a public **MarketStatusChange** (1005) message is sent to the market to notify the reopening of Contract B in two minutes.

Two minutes later, the Contract goes to the Continuous phase as defined by its pattern at this time. Before going into Continuous phase an Uncrossing is performed. When the Uncrossing is triggered a public **MarketStatusChange** (1005) message is disseminated to the market.

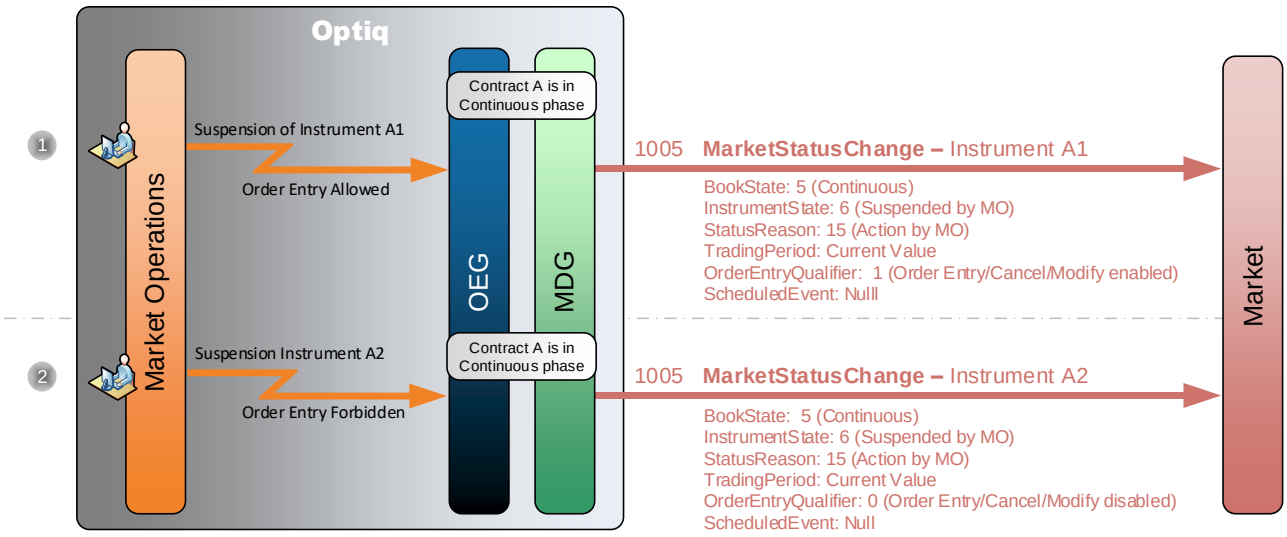
For each Instrument, if the Uncrossing Price lies within the collars, a public **PriceUpdate** (1003) message is sent to the market with the uncrossing price and the quantity at which the uncrossing is performed.

For each trade generated a public **MarketUpdate** (1001) is sent.

At the end of the uncrossing process a public **MarketUpdate** (1001) message is sent to update the values of each limit that has changed.

When the uncrossing is fully performed the Contract B switches to a Continuous phase. This is communicated to the market by a public **MarketStatusChange** (1005) message.

4.2.3 Instrument Suspended by Market Operations



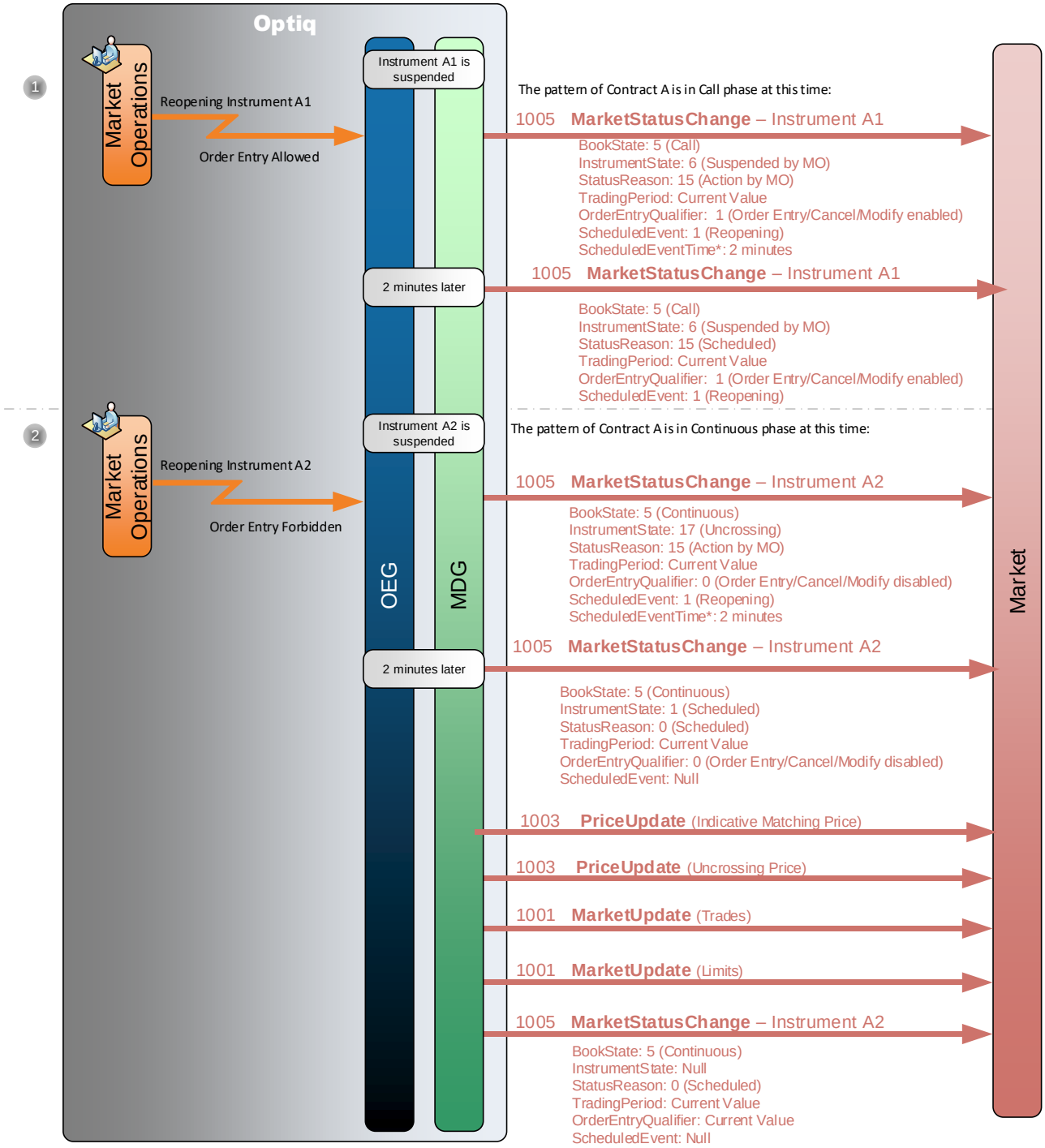
*Contract A is in a Continuous trading phase as defined in the **TimeTable** and by the pattern associated to this contract. Instruments A1 and A2 are part of the contract.*

① Market Operations suspends Instrument A1 and set the Order Entry as enabled. This is communicated to the market by a public **MarketStatusChange** (1005) message for the instrument.

② Market Operations suspends the instrument and disables the Order Entry. This is communicated to the market by a public **MarketStatusChange** (1005) message for the instrument.

Note: No **MarketStatusChange** (1005) is disseminated for the Contract.

4.2.4 Instrument Reopened by Market Operations



**ScheduledEventTime: For readability purposes it is expressed as a duration in minutes but in reality this field is expressed as nanoseconds since Epoch.*

*Contract A is in a Call trading phase as defined in the **TimeTable** and by the pattern associated to this contract. Instruments A1 and A2 are part of the Contract, and both are suspended.*

① Market Operations reopens the instrument A1. The instrument goes to the Call trading phase as defined by the pattern of its Contract at this time. This is communicated to the market by a public **MarketStatusChange** (1005) message.

② Market Operations reopens the instrument A2. The instrument goes to a Continuous trading phase as defined by the pattern of its Contract at this time. Before coming back to Continuous an Uncrossing is performed.

When the Uncrossing is triggered for the instrument and if the uncrossing price lies within the collars the uncrossing is performed and a public **MarketStatusChange** (1005) message is disseminated to the market for the instrument.

Right after the status change a public **PriceUpdate** (1003) message is sent to the market with the uncrossing price and the quantity at which the uncrossing is performed.

For each trade generated a public **MarketUpdate** (1001) is sent for the trade.

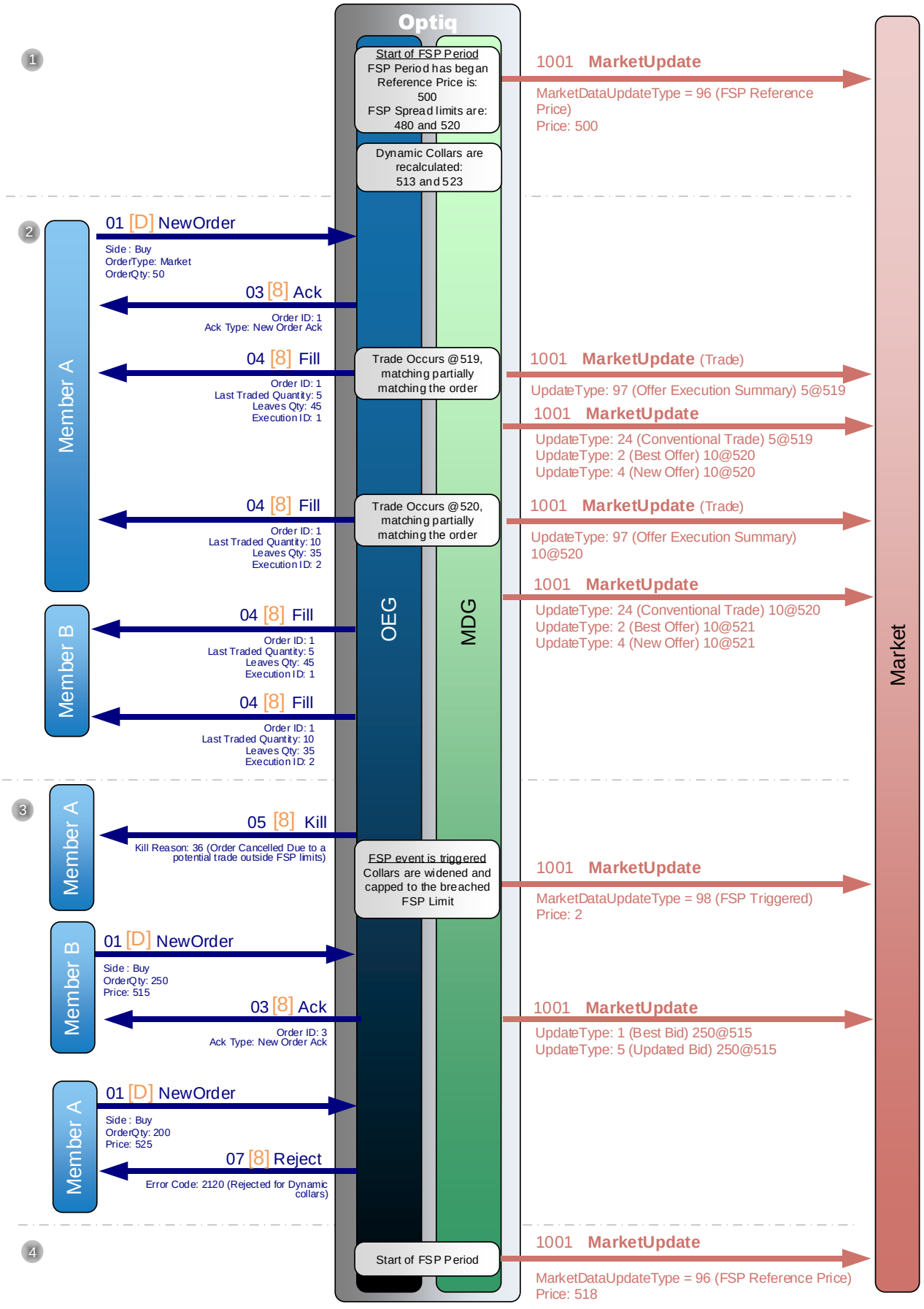
At the end of the uncrossing process a public **MarketUpdate** (1001) message is sent to update the values of each limit that has changed.

When the uncrossing is fully performed the instrument switches to the Continuous trading phase.

This is communicated to the market by a public **MarketStatusChange** (1005) message for the instrument.

4.3 FUTURE SPIKE PROTECTION

M1					
Outright Instrument					
		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
			519	5	T0
			520	10	T0
			521	10	T0



① A **MarketUpdate (1001)** message with *Market Data Update Type* set to 96 = 'FSP Reference Price' is sent to indicate start of the FSP Period. This message is sent at the start of each FSP Period. The message contains the associated FSP Reference Price of 500, with FSP limits being 480 and 520 around it.

② Member A sends a private **NewOrder** (01) (FIX D) message to enter a Buy Market order with quantity of 50. OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Part of the order immediately matches two orders already present in the book, @519 inside the FSP Limits, and @520 exactly on FSP Limit. OEG sends back private **Fill** (04) (FIX 8) message to each member to notify of the trade executions.

A public **MarketUpdate** (1001) messages are sent to the market for the Execution Summary and update of limits.

③ Remainder of the order would match orders outside of the FSP Limits. This triggers an FSP event, which is processed as following:

The remaining quantity is cancelled, and OEG sends back a private **Kill** (05) (FIX 8) message to inform Member A of this.

At the same time a **MarketUpdate** (1001) message is disseminated to the Market in order to inform of the FSP Event being triggered with the *Market Data Update Type* set to 98 = FSP Triggered.

Collars are widened, and the Price field of the MarketUpdate message contains the expansion factor. While widened, the collars are capped to the breached FSP Limit.

While FSP event is triggered, order entry and trading may still continue.

Member B enters an order within the collars and FSP limits. OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order and a public **MarketUpdate** (1001) message is sent to the market for the updated limits.

Member A enters an order which is outside of the collars. OEG sends back a private **Reject** (07) (FIX 8) message to reject this order with *ErrorCode* set to 2120 = Rejected for dynamic collar.

No messages are sent to the market.

④ At the end of the FSP event a new FSP Period starts, with a newly taken reference price and the collar expansion factor is reset. A **Market Update (1001)** message with *Market Data Update Type* set to 96 = 'FSP Reference Price' is sent to indicate start of the FSP Period, and the associated FSP Reference Price of 518.

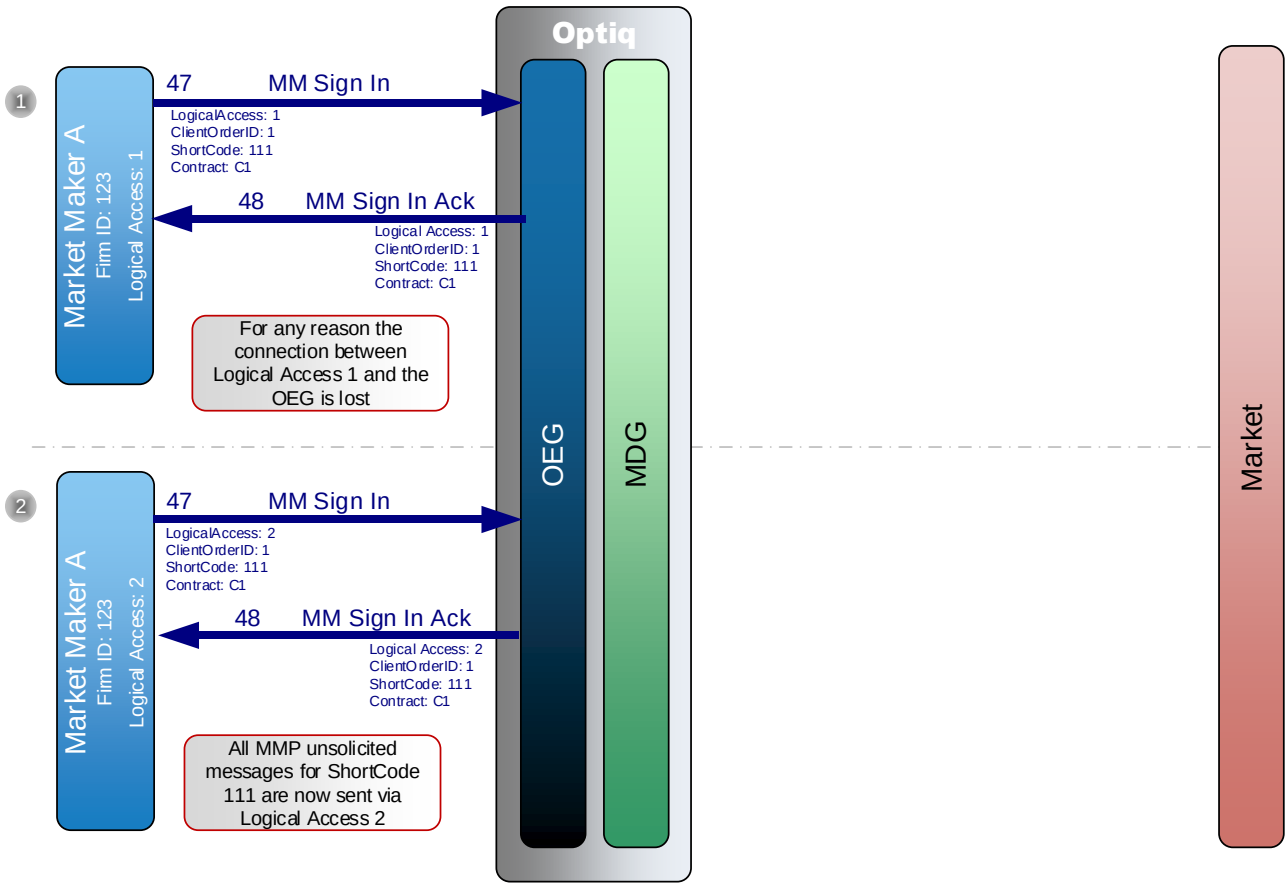
Note: Client should use the 13 = 'FSP Spread' value provided in the parameters in the standing data file to identify what spread is used, to obtain the FSP limits.

5. MARKET MAKER MESSAGES

Market Maker messages for Derivatives are available only in SBE format.

5.1 MM SESSION MESSAGES

5.1.1 Successful MM Sign-in & Unsolicited Messages



A Market Maker "trading key" is defined by the combination of Firm ID and Execution Within Firm ShortCode. Both Logical Access 1 and 2 are already logged onto the OEG.

① Market Maker A sends a **MMSignIn** (47) message to declare their Execution Within Firm ShortCode 111 via the Logical Access 1 on Contract C1.

OEG sends back a **MMSignInAck** (48) message to confirm the successful receipt and technical processing of the message. All associated unsolicited messages are sent to Logical Access 1.

Market Maker A loses the connection with the OEG via the Logical Access 1.

② In order to retrieve future unsolicited messages, Market Maker sends a private **MMSignIn** (47) with the same Execution Within Firm ShortCode 111 as in step 1.

All MMP unsolicited messages for Market Maker A in relation to Contract C1 will be sent to Logical Access 2.

Note: Market Maker A can lose the connection in two cases:

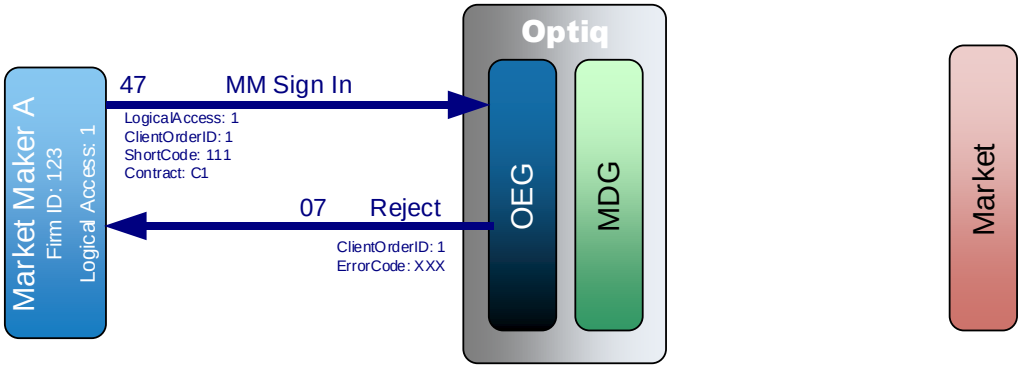
(1) Logical Access is still technically up, no technical issue between member and Optiq but In-House issue on member side
(2) Connection is technically lost, and Logical Access is down

In the first case, messages are sent by OEG to member in any case.

In the second case, all MMP messages behave like any other messages and are queued by OEG to be sent once the Logical access reconnects.

A single Market Making Logical Access may establish multiple MM sessions, by submitting different short codes, on the same or different contracts.

5.1.2 MM Sign-in Rejection



*A Member Firm 123 sends a private **MMSignIn** (47) message to declare an Execution Within Firm ShortCode 111, on Contract C1. For this example, the member 123 is not authorized as a Market Maker on contract C1.*

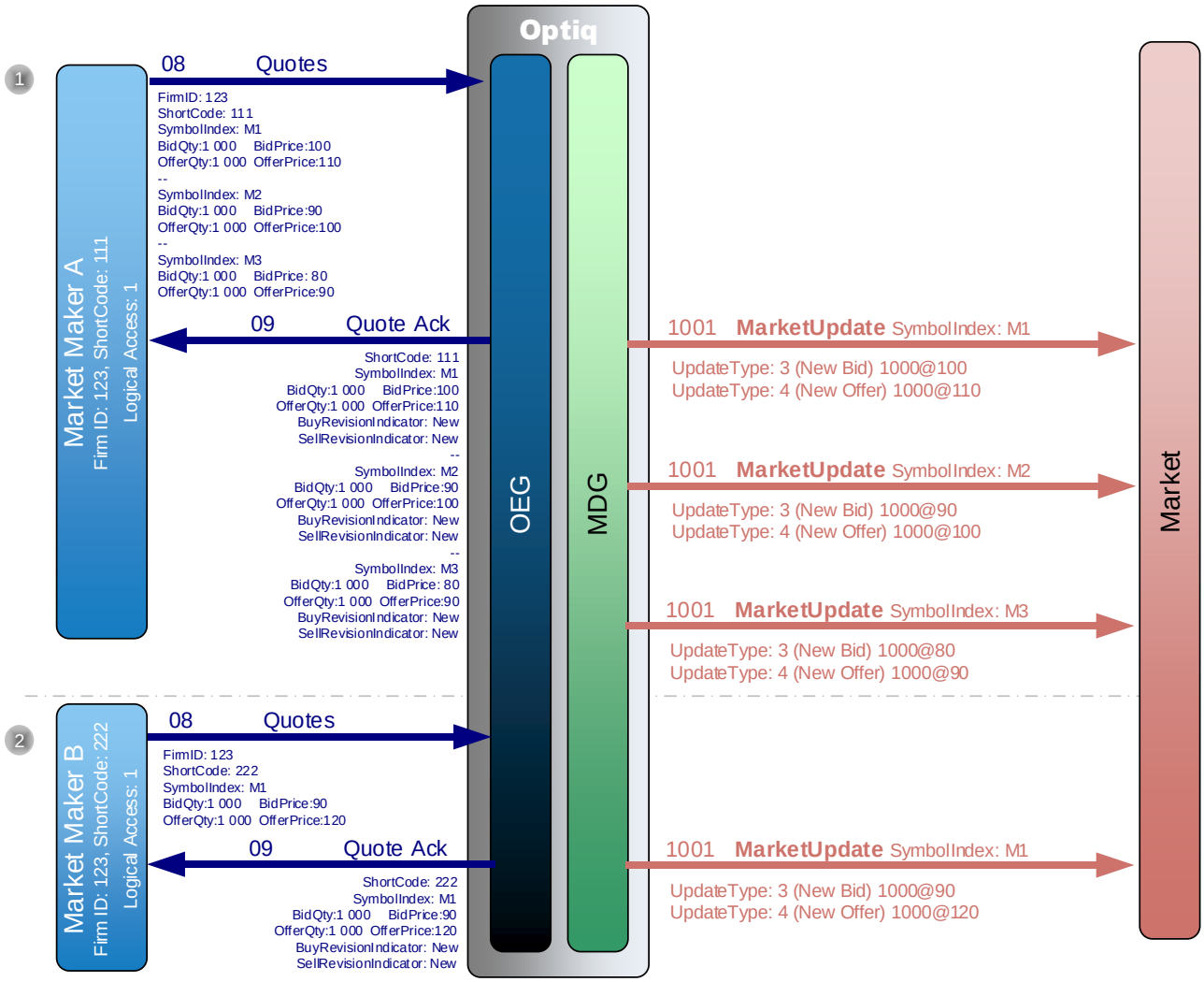
The MM Sign-in message is rejected and OEG sends back a private **Reject** (07) message with an *Error Code*. The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*.

No message is sent to the Market.

5.2 ENTERING QUOTES

Market Maker messages (i.e. Quotes) for Derivatives are available only in SBE format.

5.2.1 Mass Quote Accepted



① A Member Firm 123 previously signed-in for Execution Within Firm ShortCodes 111 and 222.

Firm sends a private **Quotes** (08) message for Execution Within Firm ShortCode 111 to enter new quotes for 3 instruments.

- For Instrument M1: Buy quote with a quantity of 1,000 and price of 100; Sell quote with a quantity of 1,000 at a price of 110.
- For Instrument M2: Buy quote with a quantity of 1,000 and price of 90; Sell quote with a quantity of 1,000 at a price of 100.
- For Instrument M3: Buy quote with a quantity of 1,000 and price of 80; Sell quote with a quantity of 1,000 at a price of 90.

OEG sends back a private **QuoteAck** (09) message to confirm the successful receipt and technical processing of the quotes.

The quotes enter the order book without matching and public **MarketUpdate** (1001) messages are sent to the market to update the BBO (if any) and the limits.

② A second Market Maker belonging to the same Firm ID sends a private **Quotes** (08) message for Execution Within Firm Shortcode 222 to enter a new Buy quote with a quantity of 1,000 at a price of 90 along with another Sell quote with a quantity of 1,000 at a price of 120 for Instrument M1.

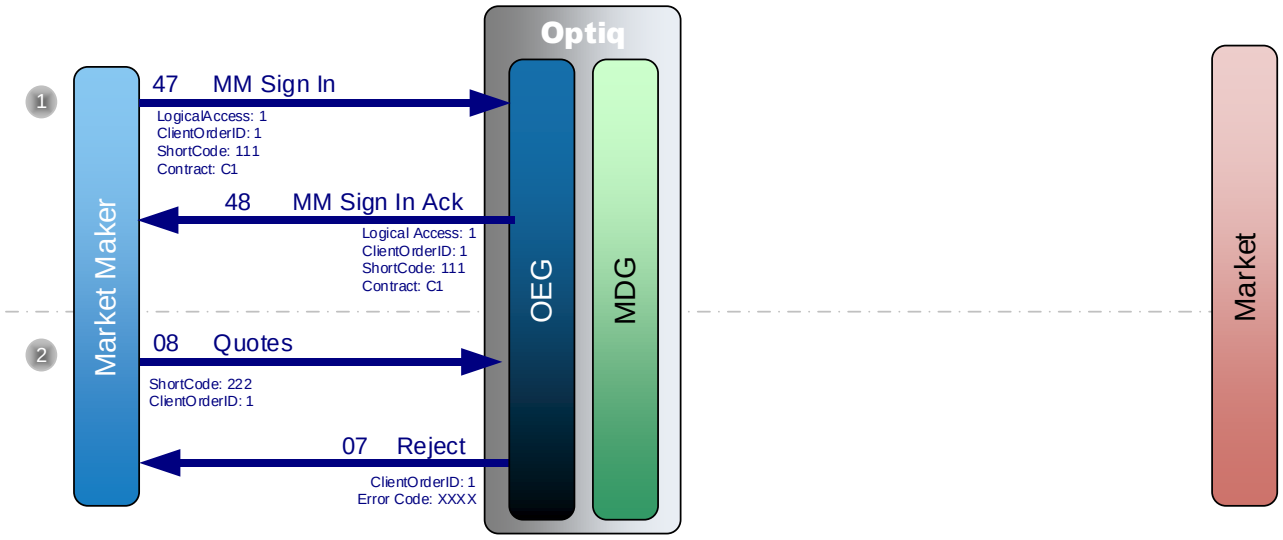
As the Quote is sent from a different Execution Within Firm ShortCode than the existing Quote in the book M1, the Quote enters the order book.

Note: In this example, entering quotes create new limits and do not update BBO.

The contract identified to route the message is the contract of the first Instrument (M1).

All following instrument inside the **Quotes** (08) message must be related to the same Contract, otherwise the Quotes will be individually rejected.

5.2.2 Mass Quote Fully Rejected



① A Member sends a private **MMSignIn** (47) message to declare the Execution Within Firm Shortcode 111 on Contract C1.

OEG sends back a private **MMSignInAck** (48) message to confirm the successful receipt and technical processing of the message.

② The same Member (same Firm ID) sends a private **Quotes** (08) message using an Execution Within Firm Shortcode that is not declared via MM Sign-in message to Optiq.

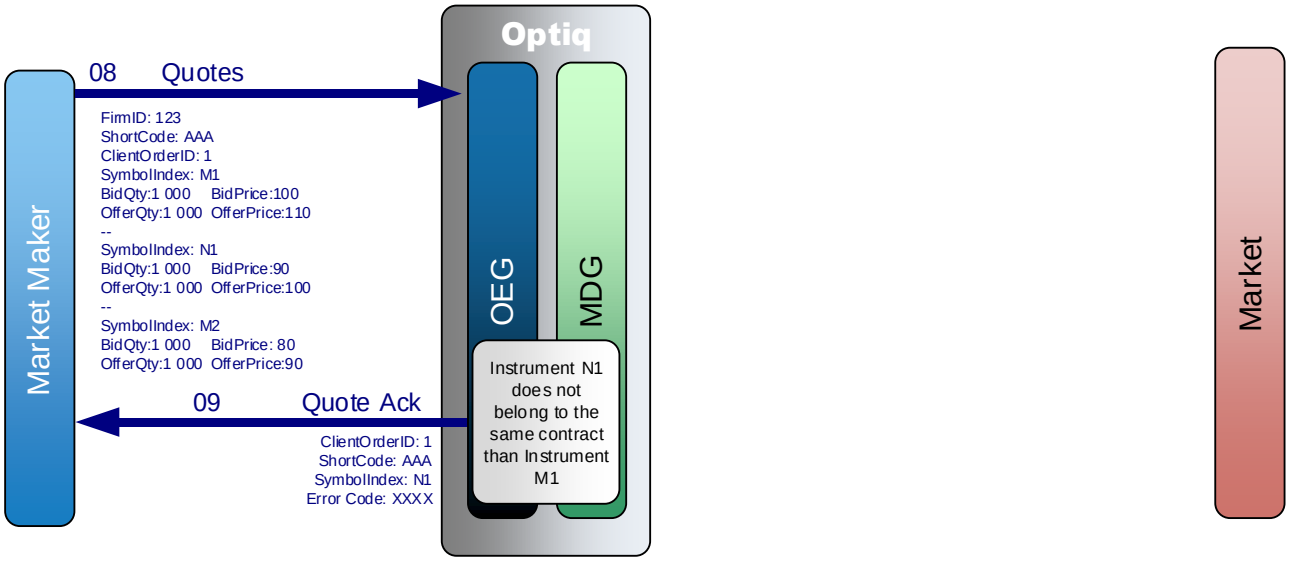
The entire **Quotes** (08) message is rejected and the OEG sends back a private **Reject** (07) message. The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv).*

No message is sent to the Market.

Note: The **Reject** (07) message is sent to fully reject the **Quotes** (08) message.

The **QuoteAck** (09) message is sent to individually reject a specific quote.

5.2.3 Mass Quote Individually Rejected



A Market Maker sends a **Quotes** (08) message for three instruments: M1, N1, M2. The second instrument N1 does not belong to the same contract as the first instrument in the Quote, i.e. M1.

The second Quote is rejected so OEG sends back a private **QuoteAck** (09) to reject this quote.

For the submission in this example **QuoteAck** (09) will respond with repeating groups for the three submitted instruments as following:

Repeating group 1 for instrument M1

Bid – Accepted, with the Bid Error Code being set to zero (0) – meaning no errors

Offer – Accepted, with the Offer Error Code being set to zero (0) – meaning no errors

Repeating group 2 for instrument N1

Bid – Rejected, with Bid Error Code being set to 1153 – “Quote must be sent on the same contract as the first valid quote”.

Offer – Rejected, with Offer Error Code being set to 1153 – “Quote must be sent on the same contract as the first valid quote”.

Repeating group 3 for instrument M2

Bid – Accepted, with the Bid Error Code being set to zero (0) – meaning no errors

Offer – Accepted, with the Offer Error Code being set to zero (0) – meaning no errors

Note: The **QuoteAck** (09) message is sent to individually reject a specific quote.

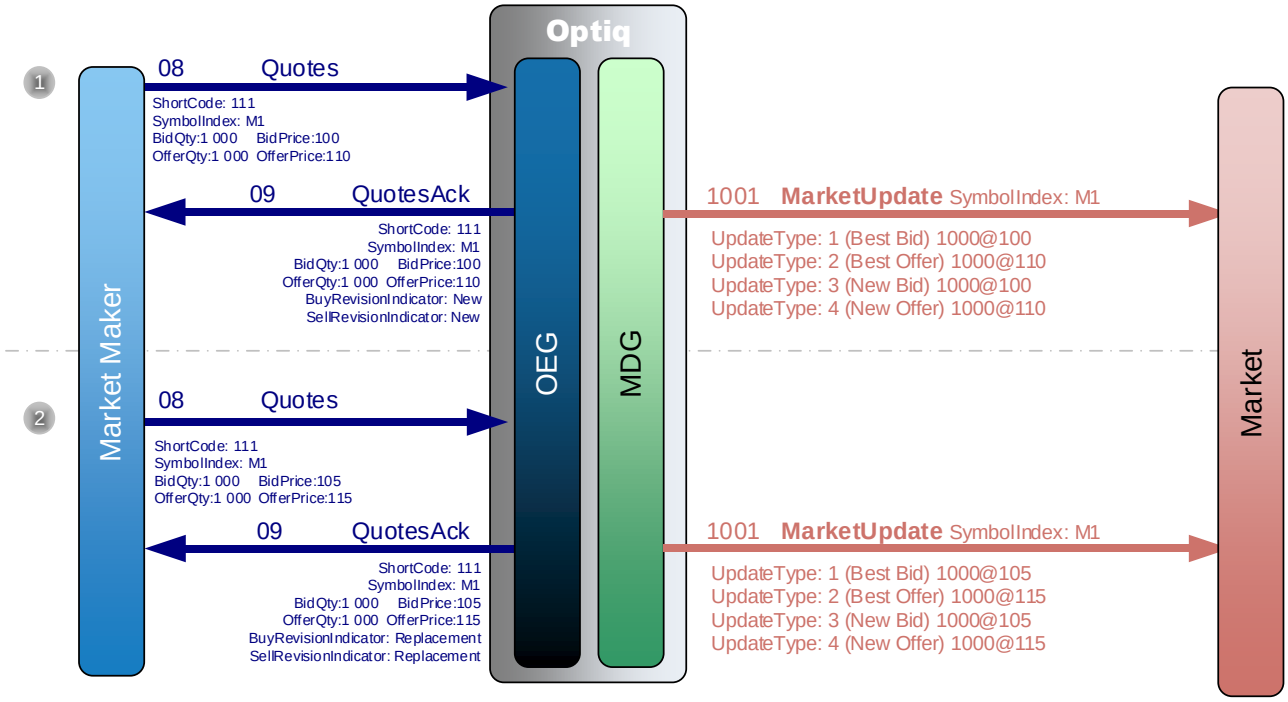
The **Reject** (07) message is sent to fully reject the **Quotes** (08) message.

All Quotes within a **Quotes** (08) message for Derivatives markets must belong to the same Contract.

Optiq identifies the contract of refence based on the first valid instrument in the **Quotes** (08) message.

5.3 MODIFYING A QUOTE

5.3.1 Modifying an Unmatched Quote



Firm previously declared Execution Within Firm ShortCode 111 for market making on Contract M, which includes instrument M1.

① A Market Maker sends a private **Quotes** (08) message to enter a new Buy quote with a quantity of 1,000 at a price of 100 along with another Sell quote with a quantity of 1,000 at a price of 110.

OEG sends back a private **QuotesAck** (09) message to confirm the successful receipt and technical processing of the quotes.

The quotes enter the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the BBO and the limits.

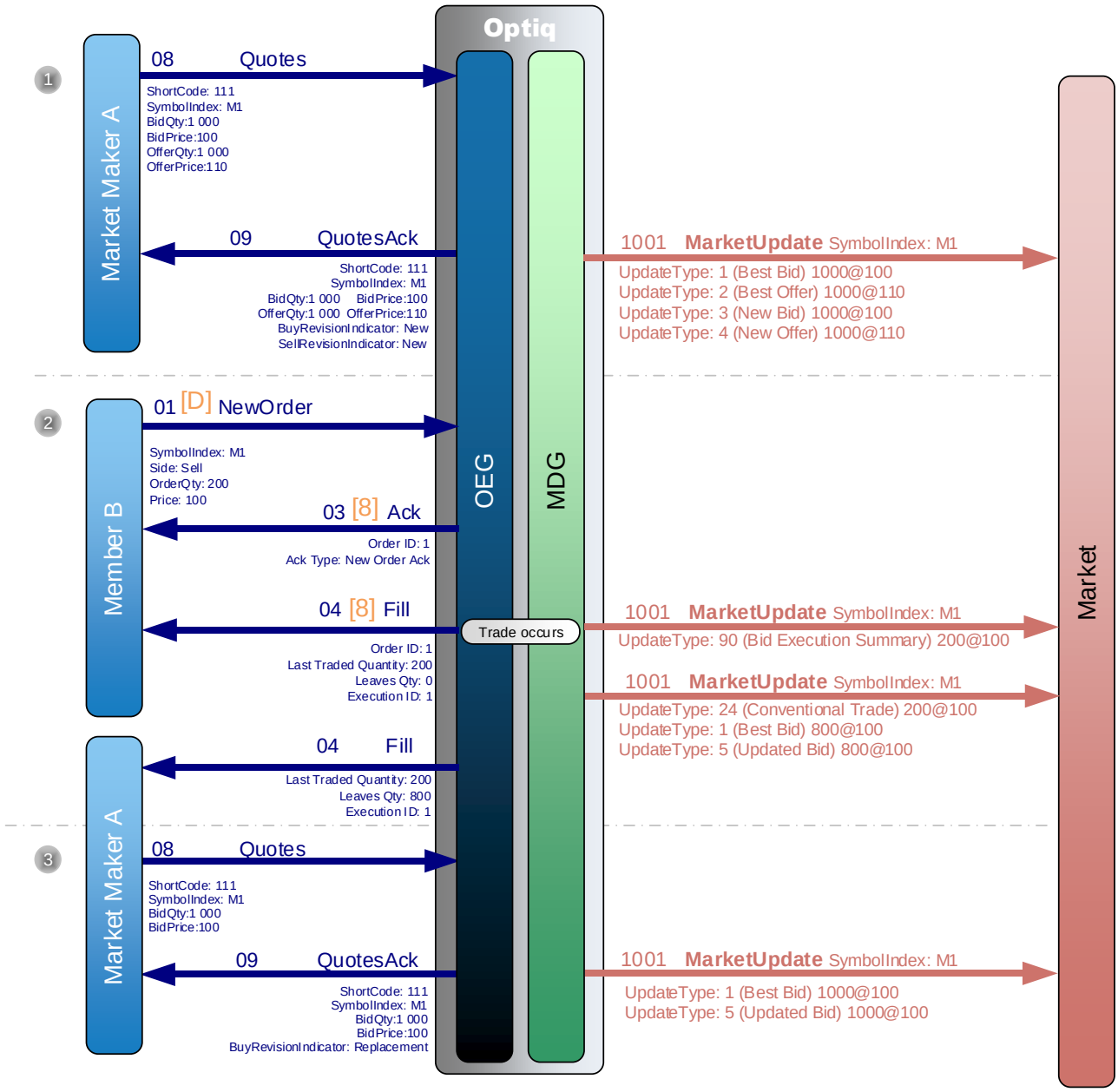
② The same Market Maker (with the same Execution Within Firm ShortCode as in Step 1) sends a private **Quotes** (08) message to revise the Buy quote with a new price of 105 and the Sell quote with a new price of 115.

OEG sends back a private **QuoteAck** (09) message to confirm the successful receipt and technical processing of the quotes. The existing quotes are replaced because new quotes are sent from the same Execution Within Firm ShortCode. The *Revision Indicator,* for the both sides, is set to 'Replacement' as these new quotes are replacing the old ones.

The quotes enter the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the BBO and the limits.

Note: For Derivatives, Quantity set in the **Quotes** (08) message is the Quantity the Market Maker wants to display in the Order Book.

5.3.2 Modifying the Volume of a Partially Matched Quote



① A Market Maker sends a private **Quotes** (08) message to enter a new Buy quote with a quantity of 1,000 at a price of 100 along with another Sell quote with a quantity of 1,000 at a price of 110.

OEG sends back a private **QuoteAck** (09) message to confirm the successful receipt and technical processing of the quotes.

The quotes enter the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the BBO and the limits.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 200 and a price of 100.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately matches the order from the Quote submitted by the Market Maker A and OEG sends back a private **Fill** (04) (FIX 8) message to each member to publish the trade execution.

A public **MarketUpdate** (1001) message is immediately sent to the market for the Execution Summary.

Then public **MarketUpdate** (1001) messages are sent to the market for the Trades and the Limits.

③ Market Maker A sends a private **Quotes** (08) message to revise the Buy quote without modify the Sell quote. The quantity of the Buy quote is set to 1,000 meaning that the member wants a displayed quantity of 1,000 in the order book. The quantity of the Sell quote is set to *Null* meaning that the existing Sell Quote should not be updated.

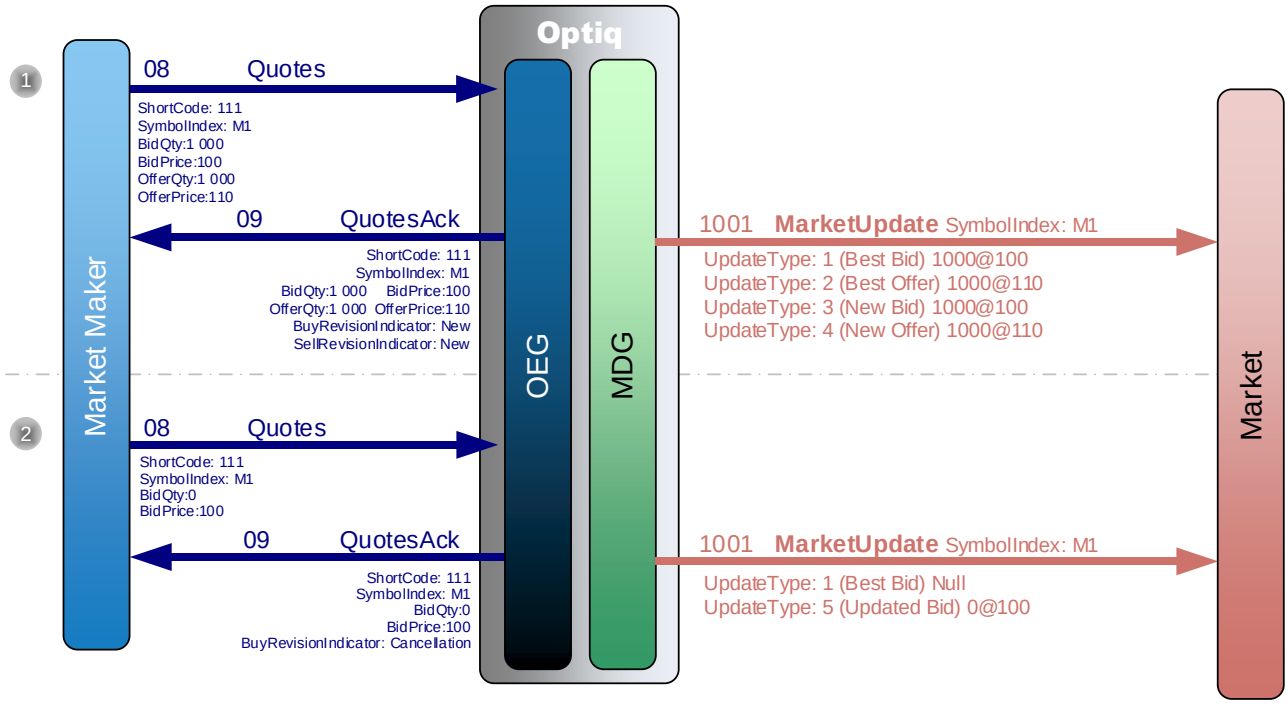
OEG sends back a private **QuoteAck** (09) message to confirm the successful receipt and technical processing of the quote. The *Revision Indicator* for the Buy side is set to 'Replacement' as the new Buy quote is replacing the old one.

A public **MarketUpdate** (1001) message is sent to the market to update the BBO and the Limits.

Note: For Derivatives, Quantity set in the **Quotes** (08) message is the Quantity the Market Maker wants to display in the Order Book.

Modifying the price of a partially matched quote leads to the same behaviour.

5.4 CANCELLING QUOTES



① A Market Maker sends a private **Quotes** (08) message to enter a new Buy quote with a quantity of 1,000 at a price of 100 along with another Sell quote with a quantity of 1,000 at a price of 110.

OEG sends back a private **QuoteAck** (09) message to confirm the successful receipt and technical processing of the quotes.

The quotes enter the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limits.

② The same Market Maker (with the same Execution Within Firm ShortCode as in step 1) sends a private **Quotes** (08) message with a quantity of 0 to cancel the Buy quote.

OEG sends back a private **QuoteAck** (09) message to confirm the successful receipt and technical processing of the quotes. The *Revision Indicator* for the Buy side is set to 'Cancellation'.

A public **MarketUpdate** (1001) message is sent to the market to update the BBO and the Limits.

Note: CancelReplace (06) and **CancelRequest** (12) messages are not applicable for Quotes.

5.5 MM PROTECTION MESSAGES

For readability purposes for MM Protection, private and public messages related to trades are not displayed in the diagram.

Market Maker Protection messages for Derivatives are available only in SBE format.

Unsolicited messages sent by the OEG for the market maker protection facility are sent to the Logical access identified in the sign-in message.

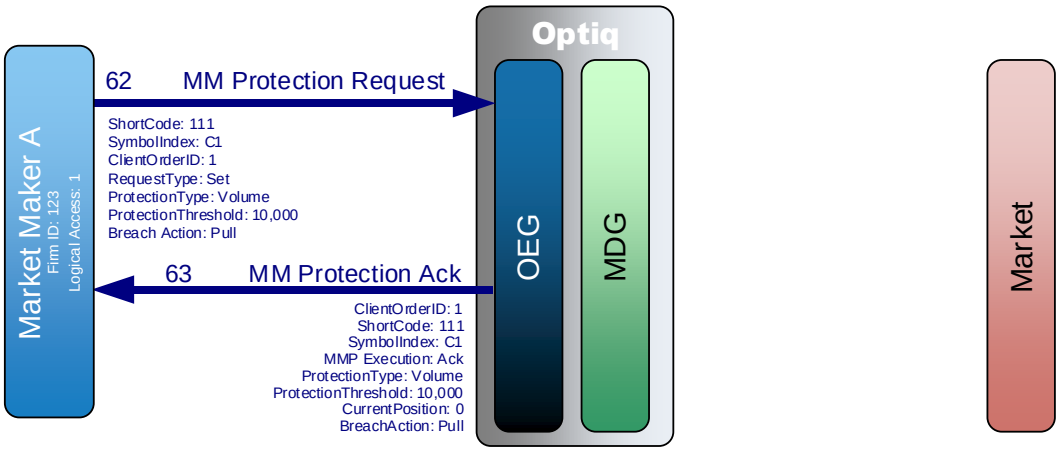
- Ack for Update of exposure

Solicited messages are replied back to the Logical Access that initiated the request.

- Ack for Set / Get / Update
- Reject

Kill messages triggered by breach of Market Maker protection are sent to the Logical Access that originally sent the order.

5.5.1 Setting the MM Protection



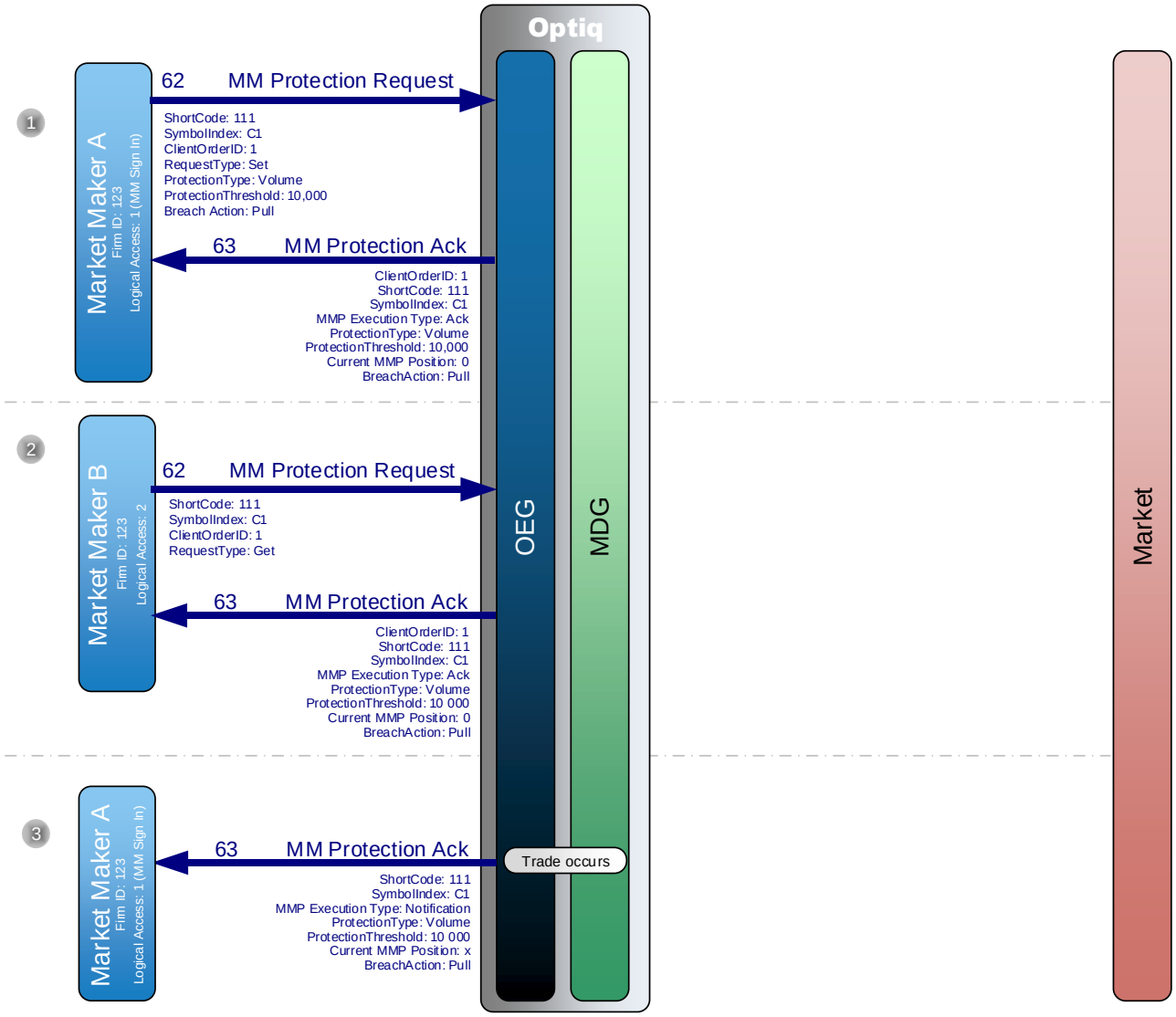
Market Maker A sends a private **MMProtectionRequest** (62) message to set a Protection Threshold with a volume of 10,000 for the Contract C1.

OEG sends back a private **MMProtectionAck** (63) message to confirm the successful receipt and technical processing of the setting.

No message is sent to the Market.

Note: A private **MMProtectionRequest** (62) message with *RequestType* = 'Set' resets the *Current MM Position*, unlike the same request set with a *RequestType* = 'Adjust'. Submitting messages to Adjust MM protection only adjusts the configuration of the protection.

5.5.2 Requesting the MM Protection State



① Market Maker A from Logical Access 1 send a **MMProtectionRequest** (62) message to set the MM Protection for Execution Within Firm ShortCode 111.

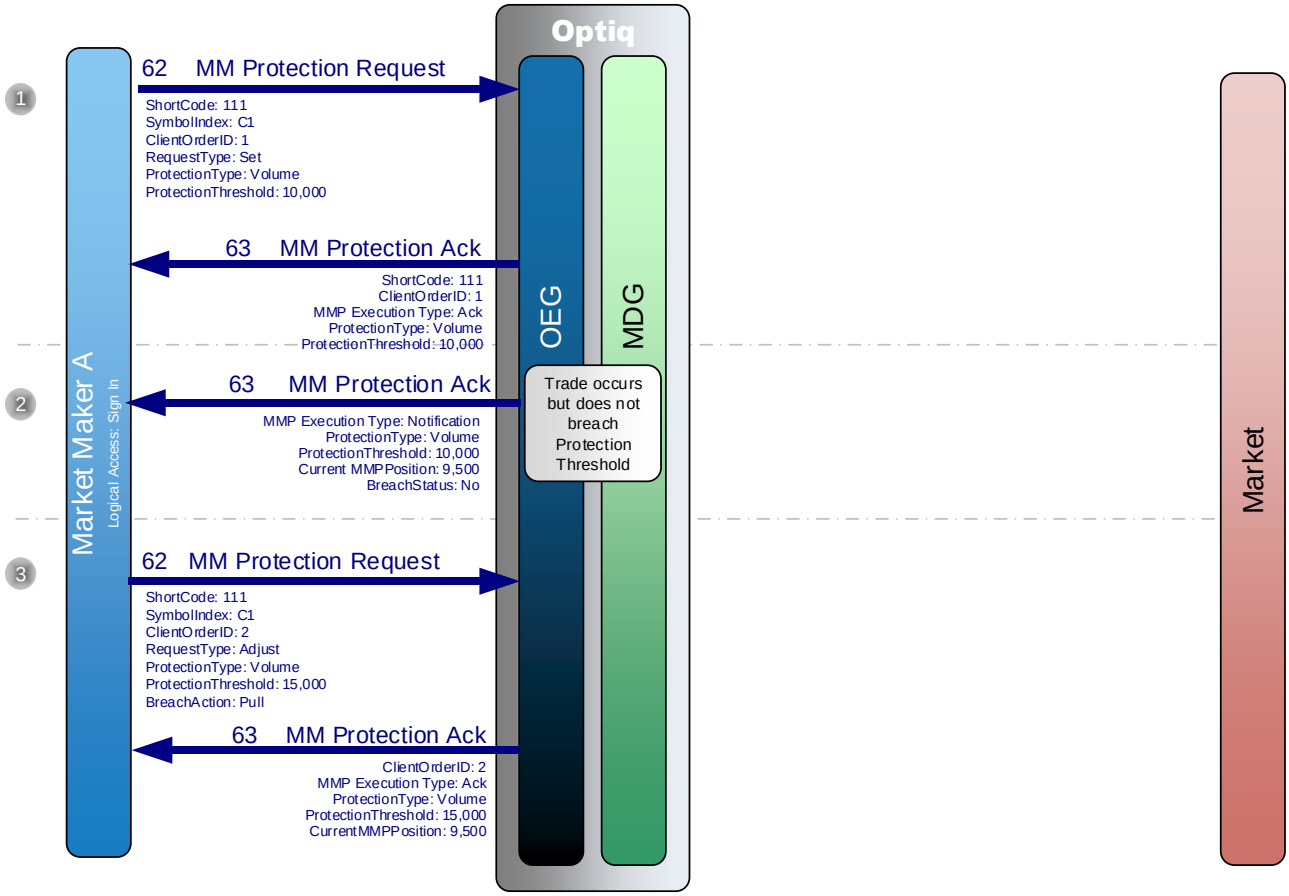
OEG sends back a private **MMProtectionAck** (63) message to the Logical Access 1, to confirm the successful receipt and technical processing of the message.

② Market Maker B from a Logical Access different than Market Maker A (but same Firm ID) sends a **MMProtectionRequest** (62) to get the status of the current MM Protection for Execution Within Firm ShortCode 111.

OEG sends back a private **MMProtectionAck** (63) message to provide the current position for Execution Within Firm ShortCode 111, which is routed to Logical Access 2.

③ A trade occurs without breaching a threshold, the Current Position of Execution Within Firm ShortCode 111 is updated and a private **MMProtectionAck** (63) is sent to the Logical Access defined in the **MMSignIn** (47) message.

5.5.3 Adjusting the MM Protection



① Market Maker A sends a private **MMProtectionRequest** (62) message to set a Protection Threshold with a volume of 10,000 for the Contract C1.

OEG sends back a private **MMProtectionAck** (63) message to confirm the successful receipt and technical processing of the setting.

② A trade occurs involving Market Maker A (with same Execution Within Firm ShortCode as in step 1) without breaching of the Protection Threshold. OEG sends a private **MMProtectionAck** (63) to Market Maker A to notify market maker of the new current MM protection position.

③ Market Maker A sends a private **MMProtectionRequest** (62) to adjust the Protection volume Threshold from 10,000 to 15,000.

OEG sends back a private **MMProtectionAck** (63) to Market Maker A to confirm the successful receipt and technical processing of the message. The MM protection position is not reset.

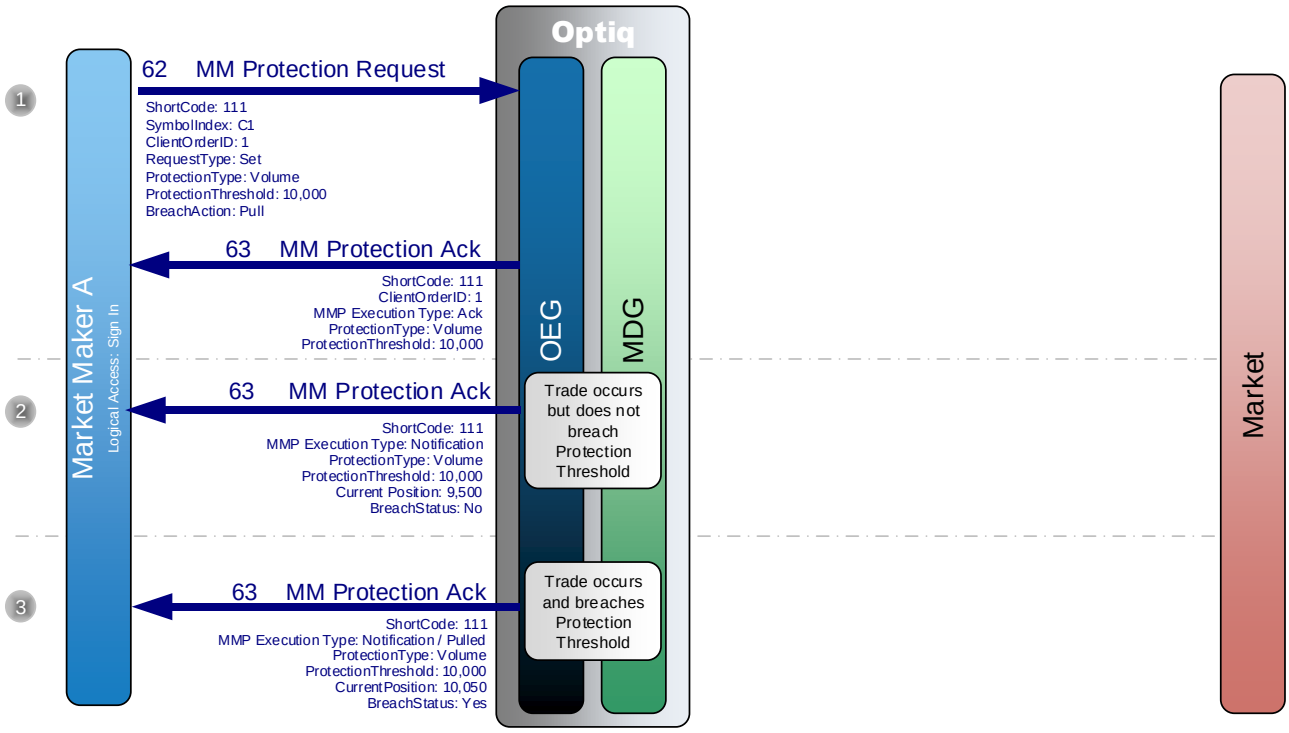
Note: A **MMProtectionRequest** (62) with *RequestType* = 'Adjust' does not reset the *Current MM Position*, unlike the same request set with a *RequestType* = 'Set'. Submitting messages to Adjust MM protection only adjusts the configuration of the protection.

Updates of the MM Protection values can only be done via the Logical Access that submitted the sign-in and set the MM Protection for the Firm and short code.

Upon setup of MM protection OEG will send the updated MM protection level for every trade, even if the Breach Action is not set.

5.5.4 Breach of MM Protection

For readability purposes the individual Fill and Market Update messages are not explicitly shown.



① Market Maker A sends a private **MMProtectionRequest** (62) message to set a Protection for a Volume of 10,000 for the Contract C1.

OEG sends back a private **MMProtectionAck** (63) message to confirm the successful receipt and technical processing of the setting.

② A trade occurs involving Market Maker A (with same Execution Within Firm ShortCode as in step 1), OEG sends a private **MMProtectionAck** (63) to Market Maker A to notify the new current position.

③ Another trade occurs involving Market Maker A (with same Execution Within Firm ShortCode as in step 1). This trade breaches the Protection Threshold.

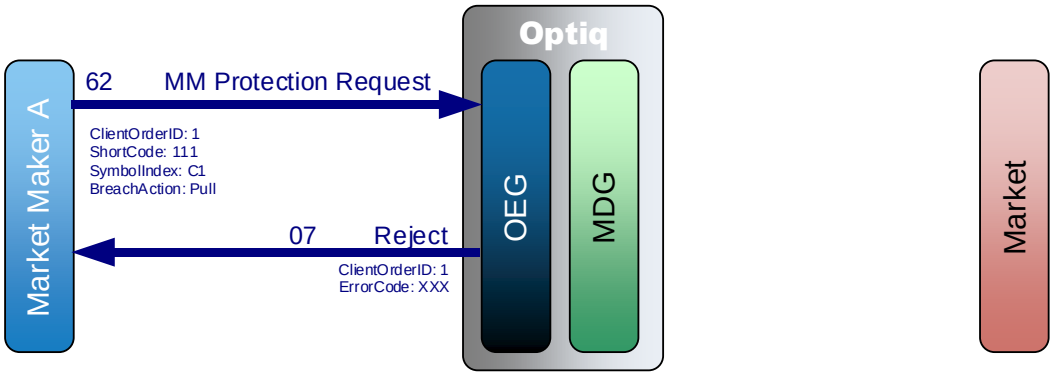
OEG sends a private **MMProtectionAck** (63) to Market Maker A to notify of the breach of MM Protection and the associated Breach Action.

Note: Following the breach, OEG will behave according to the Breach Action set.

In case it was set to "Pull" – all open orders for that Firm, Short Code and Contract combination will be cancelled. Market Maker will receive an individual **Kill** (05) (FIX 8) message for each cancelled order.

In case breach action is not set - all orders remain in the book, and market maker will continue receiving updates of their MM position upon occurrence of any trade.

5.5.5 MM Protection Rejected



Market Maker A sends a private **MMProtectionRequest** (62) message to set a MM Protection.

If the message is rejected OEG sends back a private **Reject** (07) message with an *Error Code*. The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*.

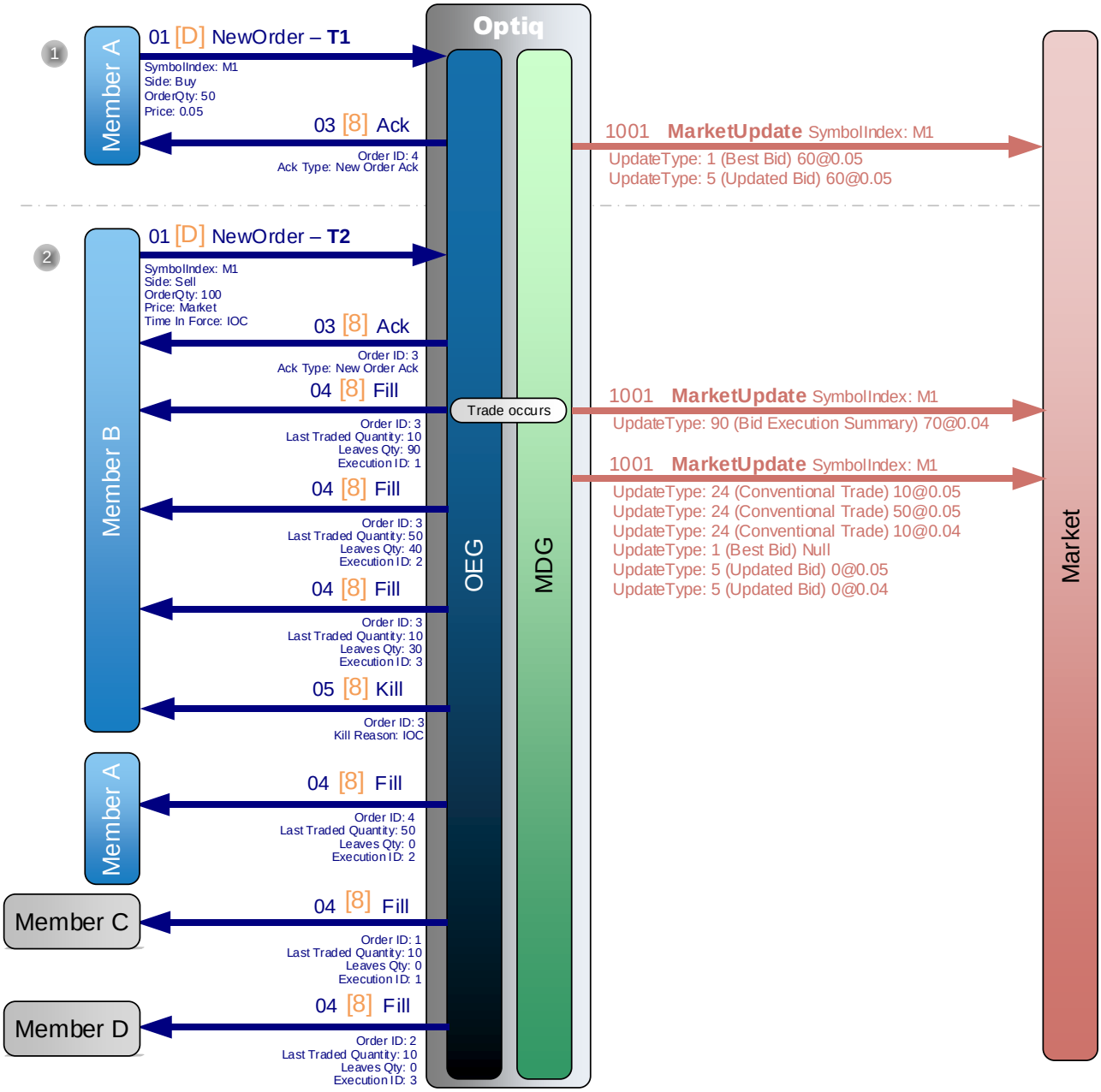
Note: Cases leading to a rejection are: Execution Within Firm ShortCode is not declared; Member is not a

Market Maker; Attempt to adjust MMP Threshold below the current MM Position (i.e. breaching the threshold).

6. TRADING KINEMATICS

6.1 EXPLICIT VERSUS EXPLICIT IN AN OUTRIGHT (NO IMPLIED PRICING)

M1 Outright Instrument					
		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
T0	10	0.05	-	100	T2
T1	50	0.05			
T0	10	0.04			



① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 50 and a price of 0.05.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to update the Limit.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 100 and a price to Market and a validity condition of Immediate or Cancel (IOC).

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately matches the three orders and OEG sends back a private **Fill** (04) (FIX 8) message to each member involved in the trade to notify the trade execution.

OEG sends back to Member B a private **Kill** (05) (FIX 8) message to kill the remaining quantity.

A public **MarketUpdate** (1001) message is immediately sent to the market for the Execution Summary.

Only then, public **MarketUpdate** (1001) messages are sent to the market for the Trades and the Limits.

Note: No dedicated **MarketUpdate** (1001) message is sent for the entry of the second order as it is immediately matched.

6.2 EXPLICIT VERSUS EXPLICIT IN STRATEGY (NO IMPLIED)

M1
Outright Instrument

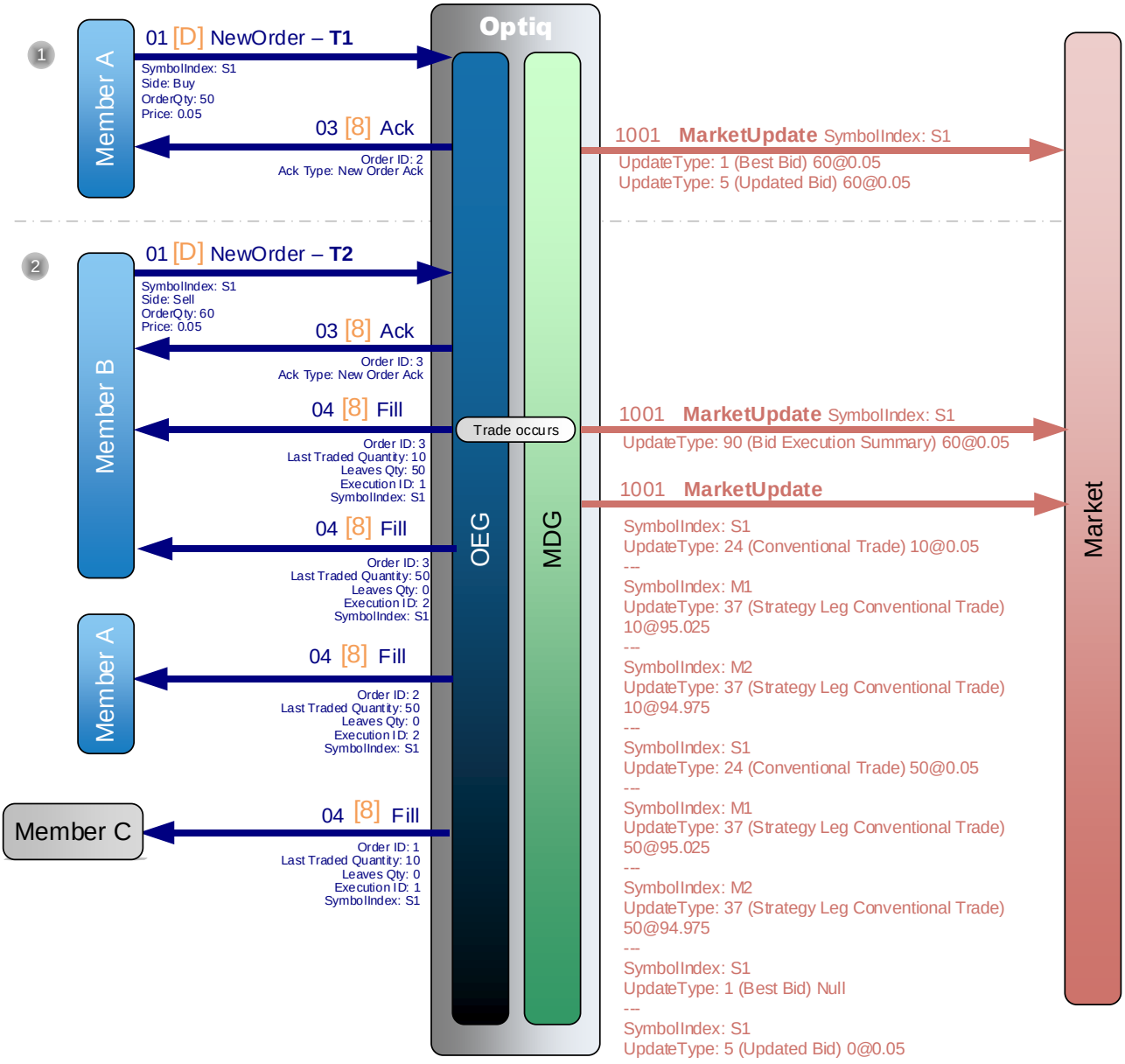
Bid			Offer		
Time	Qty	Price	Price	Qty	Time

M2
Outright Instrument

Bid			Offer		
Time	Qty	Price	Price	Qty	Time

S1
Strategy Instrument
Calendar Spread (**M1 – M2**)

Bid			Offer		
Time	Qty	Price	Price	Qty	Time
T0	10	0.05	0.05	60	T2
T1	50	0.05			



① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 50 and a price of 0.05.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to update the limit.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 60 and a price of 0.05.

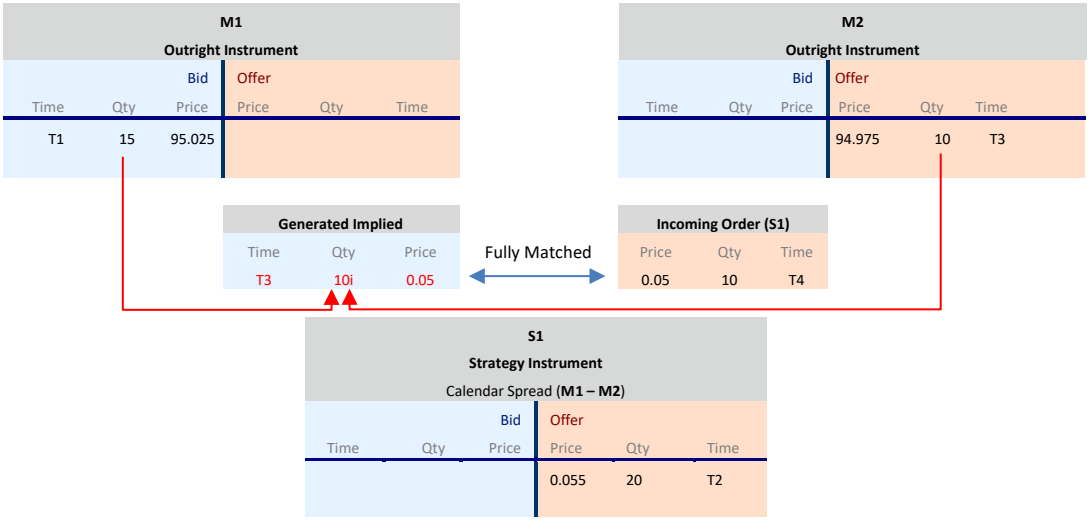
OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately matches the first order and OEG sends back a private **Fill** (04) (FIX 8) message to each member to publish the trade execution.

A public **MarketUpdate** (1001) message is immediately sent to the market for the Execution Summary of the Strategy

Only then, public **MarketUpdate** (1001) messages are sent to the market for the Trade in the Strategy (S1) and trades for each leg of the strategy (i.e. the Trades for the individual Outrights) that are flagged as the Strategy Leg Conventional Trade.

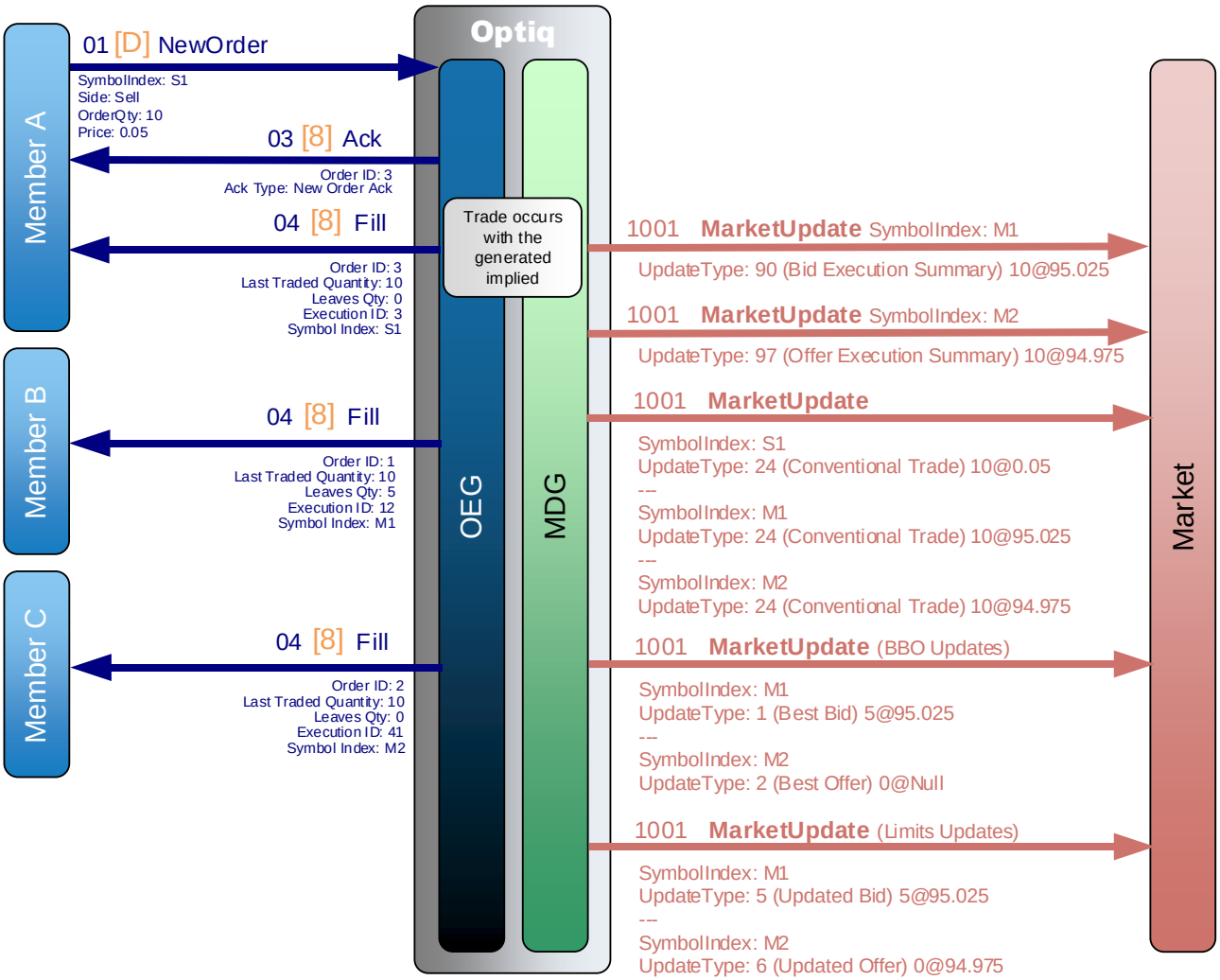
Following publication of updates for the strategy and strategy legs, another set of **MarketUpdate** (1001) messages are sent for BBO and Limits updates.



Note: In this case, the **Execution Summary** is sent for the strategy only, and not for individual legs of the strategy.

6.3 IMPLIEDS WITH EDIM: SUBMISSION OF A PRIORITY ORDER ON A STRATEGY BOOK

6.3.1 Strategy Priority Order Fully matches against Legs



Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 10 and a price of 0.05 in strategy book S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The new order automatically triggers generation of Implieds because it improves the BBO (Priority order).

The generated Implied immediately matches the Strategy's offer order and OEG sends back a private **Fill** (04) (FIX 8) message to each member to publish the trade execution.

The **Fill** (04) (FIX 8) message sent to Member A provides the execution on the Strategy S1, and the details of the execution of the outright legs.

Members B and C receive **Fill** (04) (FIX 8) messages for the execution of their individual orders in the associated outrights M1 and M2.

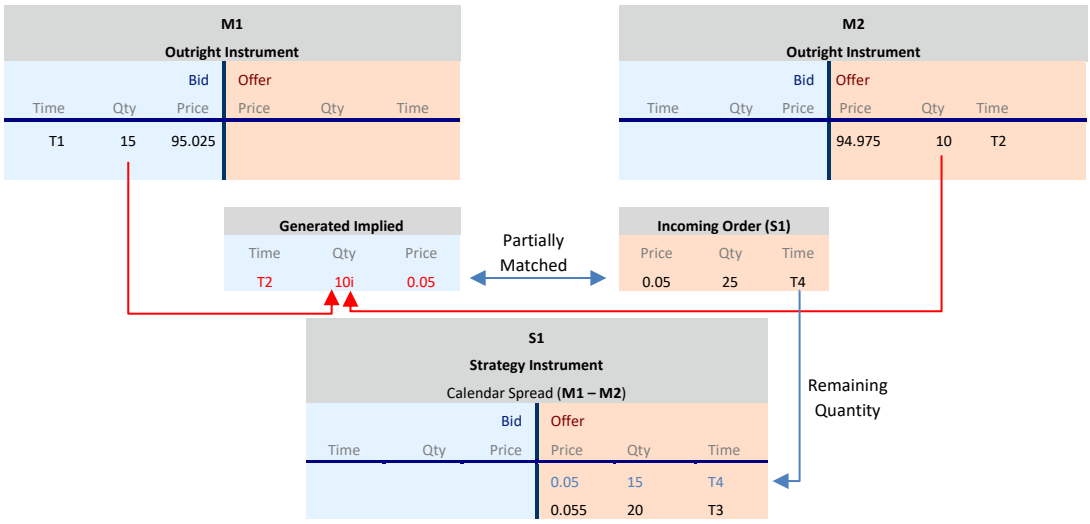
A public **MarketUpdate** (1001) message is sent to the market for the Execution Summary for each outright.

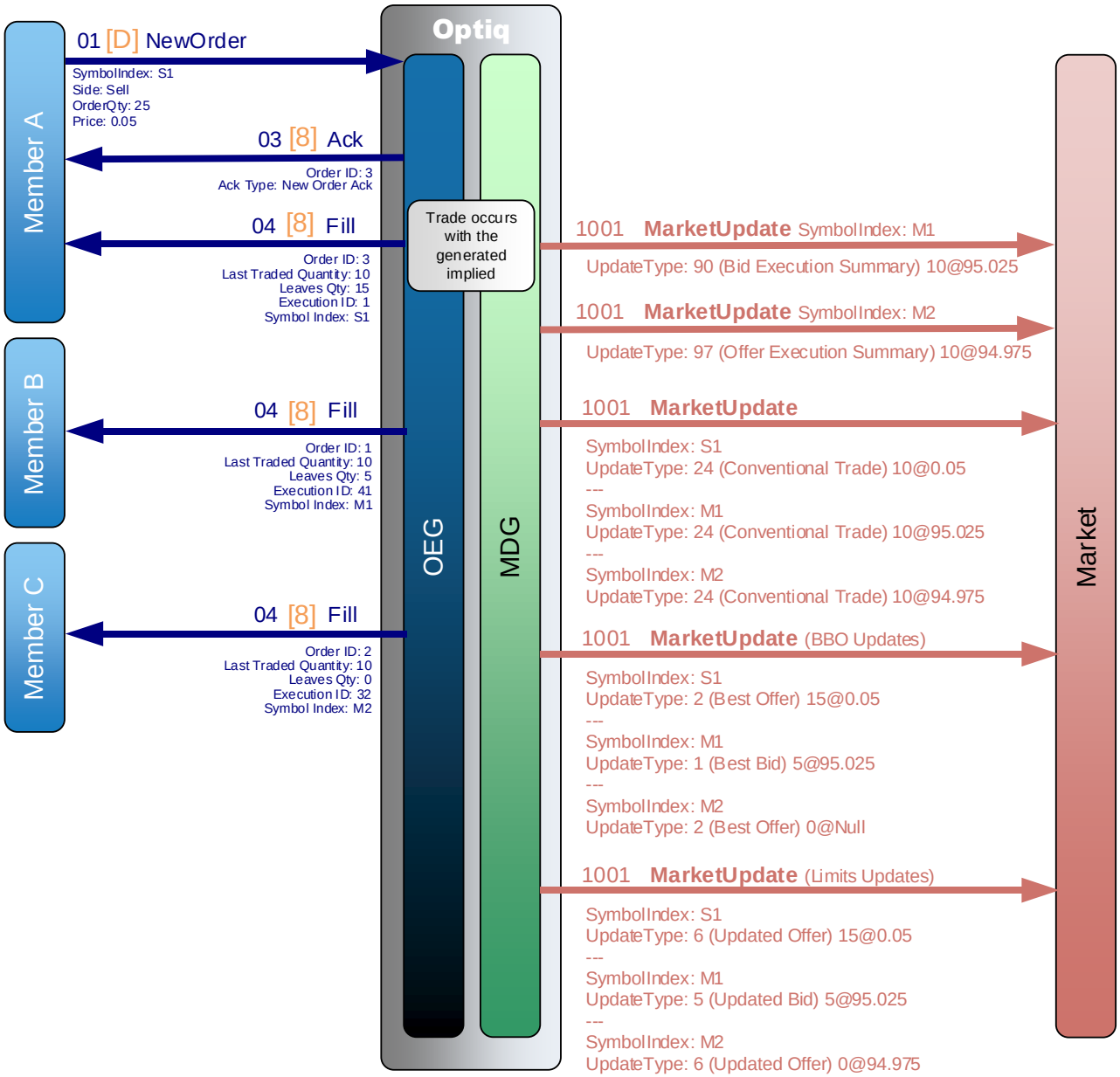
No Execution Summary is sent for the strategy.

Following this, public **MarketUpdate** (1001) messages (Conventional Trade) are sent to the market for the Trades in the Strategy (S1) and for each Outright.

Following publication of updates for the strategy and strategy legs, another set of **MarketUpdate** (1001) messages are sent for BBO and Limits updates.

6.3.2 Strategy Priority Order Partially matches against Legs





Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 25 and a price of 0.05 in strategy book S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The new order automatically triggers generation of Implieds because it improves the BBO (Priority order).

The generated Implied partially matches the strategy's offer order and OEG sends back a private **Fill** (04) (FIX 8) message to each member to publish the trade execution.

Remaining quantity of the order enters into the order book of the strategy.

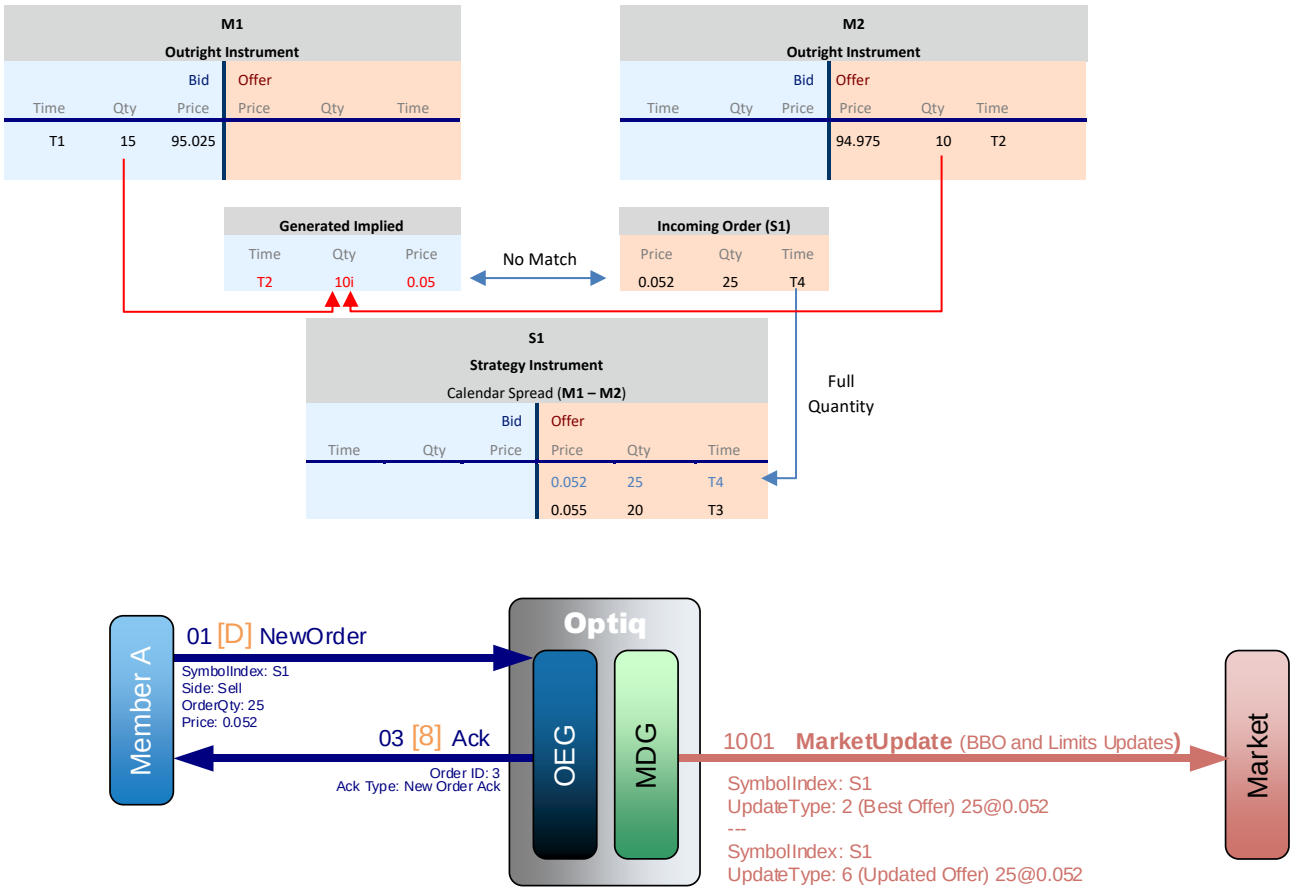
A public **MarketUpdate** (1001) message is sent to the market for the Execution Summary for each outright.

No Execution Summary is sent for the strategy.

Following this, public **MarketUpdate** (1001) message (Conventional Trade) are sent to the market for the Trade in the Strategy (S1) and trades for each Outright involved in the strategy.

Following publication of updates for the Strategy and its Legs, another set of **MarketUpdate** (1001) messages are sent for BBO and Limits updates.

6.3.3 Strategy Priority Order Does Not Match



Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 25 and a price of 0.052 in strategy book S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

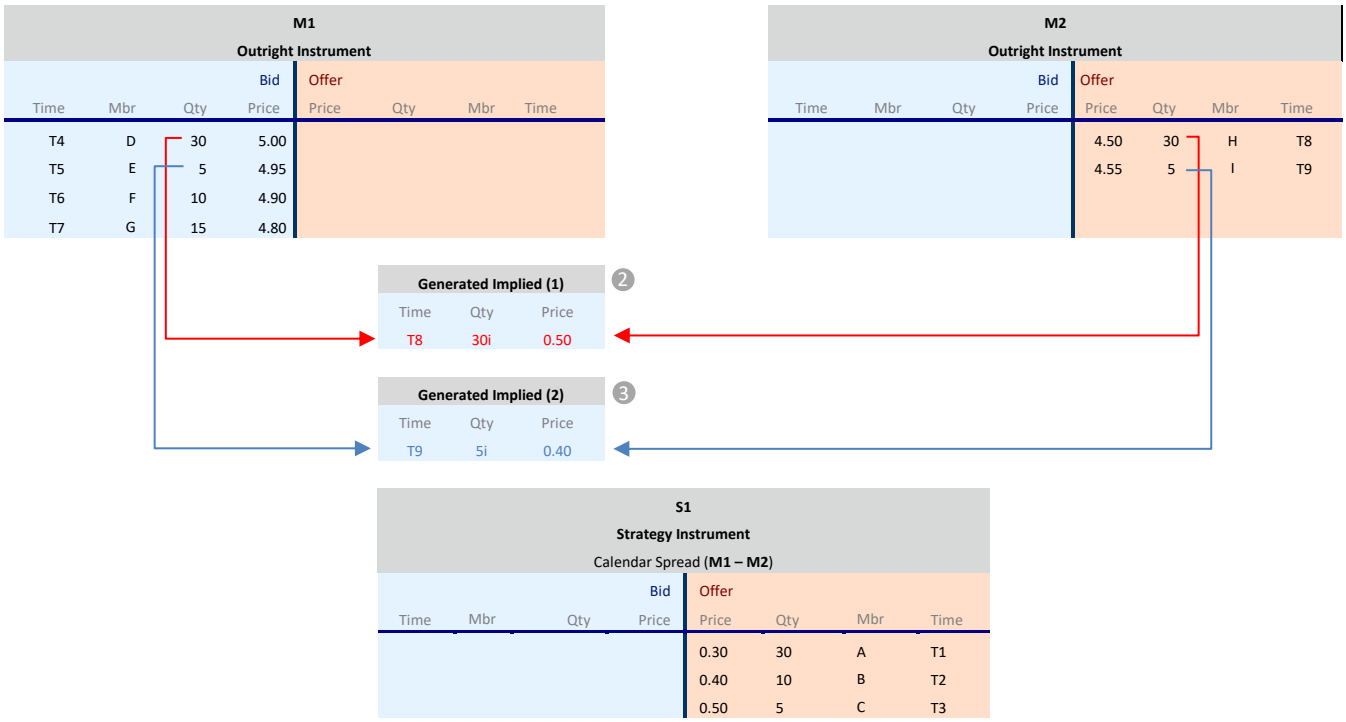
The new order automatically triggers generation of Implieds in strategy S1 because it improves the BBO (Priority order).

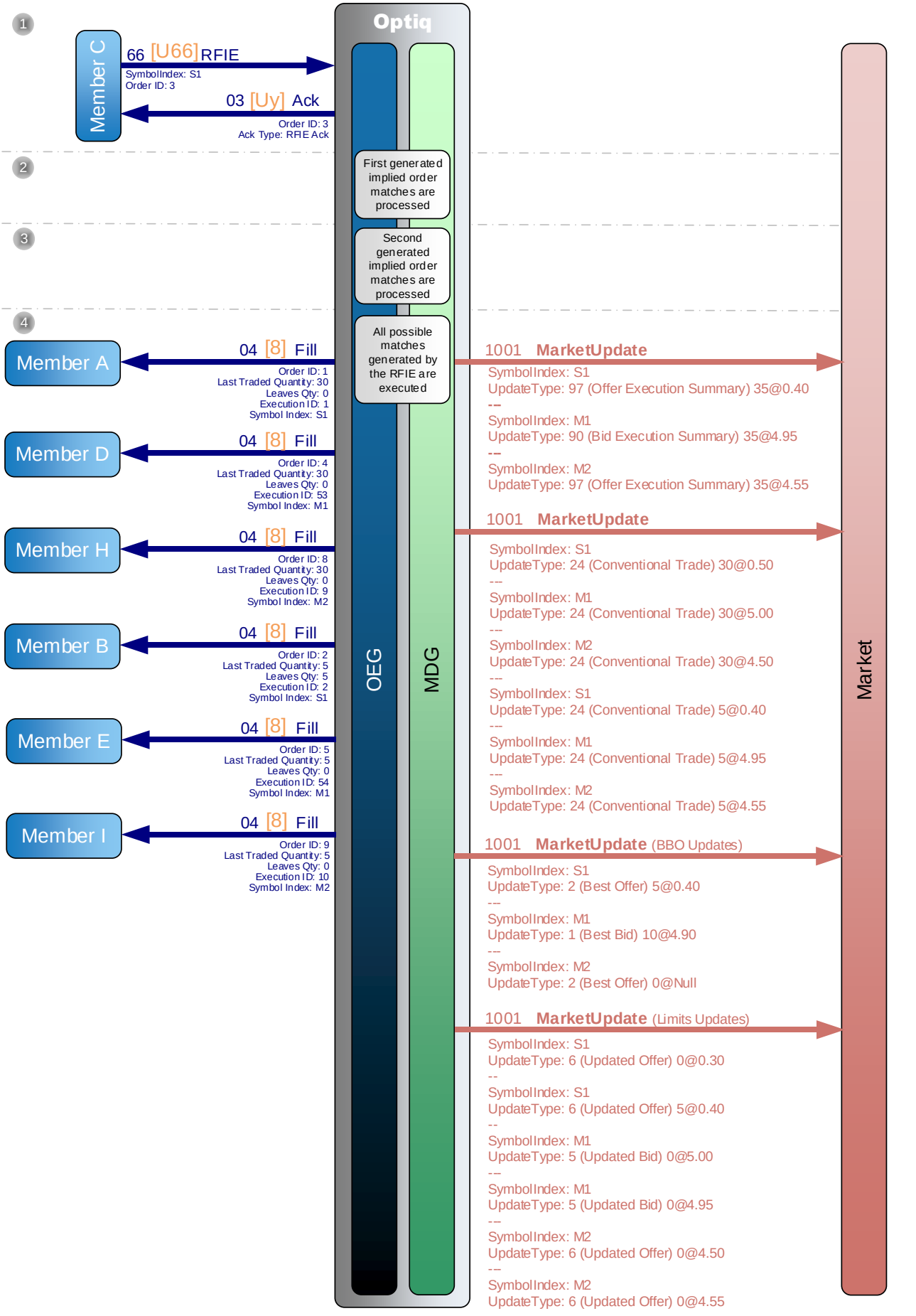
The order itself does not match any orders within the strategy, and the generated Implieds do not identify matching possibilities in the associated outright order books.

No match occurs, and the incoming order enters the strategy book.

Public **MarketUpdate** (1001) messages are sent to the market to update the BBO and the Limits.

6.4 IMPLIEDS WITH EDIM: SUCCESSFUL REQUEST FOR IMPLIED EXECUTION (RFIE)





① Member C sends a private **RequestForImpliedExecution** (66) (FIX U66) message to request the recalculation of implieds in order book of strategy S1.

OEG sends back a private **Ack** (03) (FIX Uy) message to confirm the successful receipt and technical processing of the request.

In this example, two Implieds are sequentially generated.

② The first generated Implied matches the order of the Member A for a quantity of 30 (full match).

③ The second generated Implied matches the Member B's order for a quantity of 5 (partial match).

④ Upon completion of processing of transactions, OEG sends back a private **Fill** (04) (FIX 8) message to each member for all the trade executions.

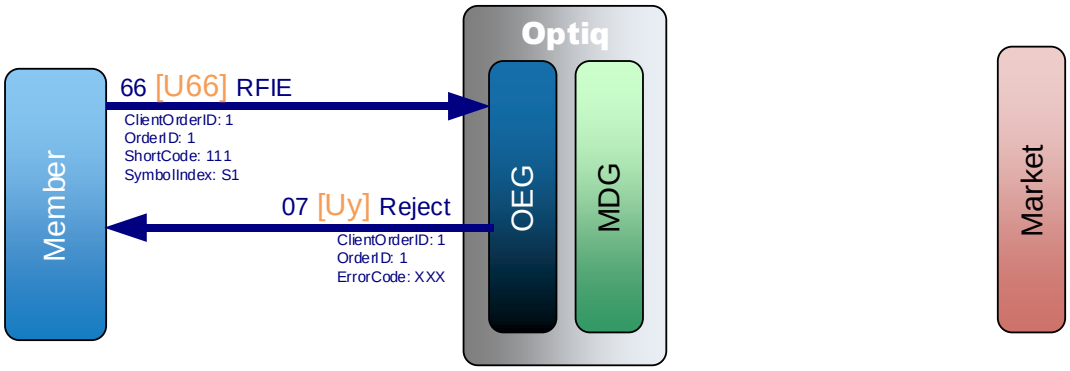
Public **MarketUpdate** (1001) messages are immediately sent to the market for the Execution Summary of each Outright (M1 and M2) and for the Strategy S1.

Following this, public **MarketUpdate** (1001) messages (Conventional Trade) are sent to the market for the trade on the Strategy (S1) and trades for each Outright involved in the Strategy.

Following publication of updates for the Strategy and Strategy Legs, another set of **MarketUpdate** (1001) messages are sent for BBO and Limits updates.

Note: The ordering of BBO and limits updates between instruments sent in the same **MarketUpdate** (1001) is not guaranteed.

6.5 IMPLIEDS WITH EDIM: REJECTION OF REQUEST FOR IMPLIED EXECUTION (RFIE)



A Member sends a private **RequestForImpliedExecution** (66) (FIX U66) message to request recalculation of implieds, for an order on the book of Strategy S1.

If the Request for Implied Execution message is rejected then OEG sends back a private **Reject** (07) (FIX Uy) message with the associated Error Code.

The request can be rejected due to one of the following reasons:

- The EDIM model is not activated for the Derivative Contract or the type of Strategy;
- The member has no order in the order book of the Strategy;
- If the effective state of the Strategy book is not Continuous.

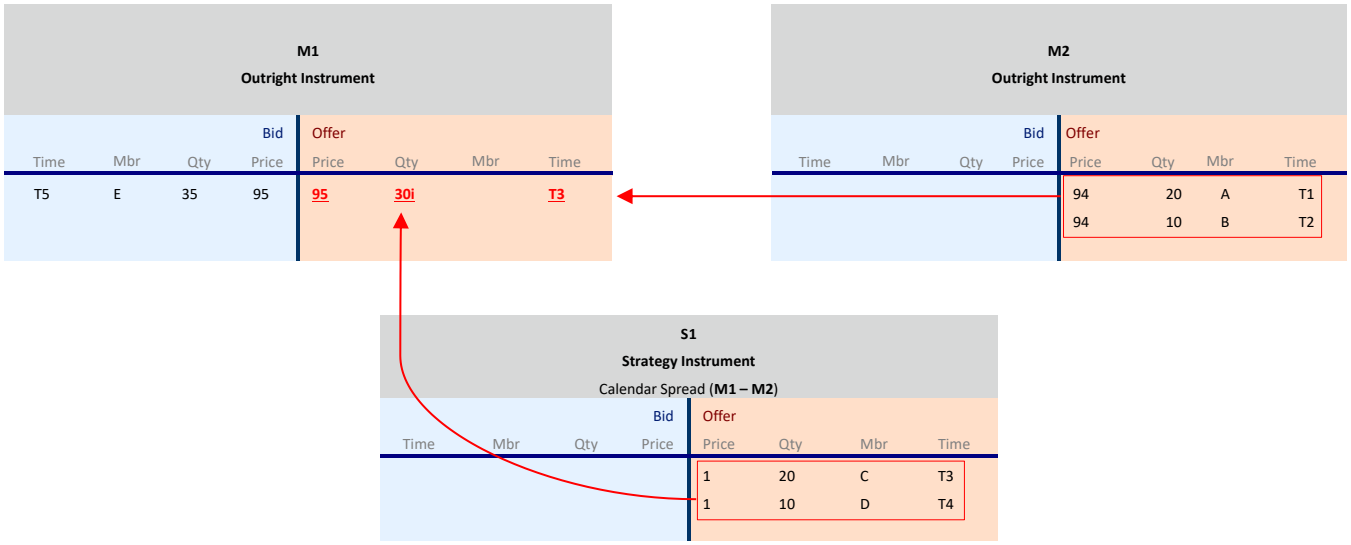
The exact reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list* (.csv) file.

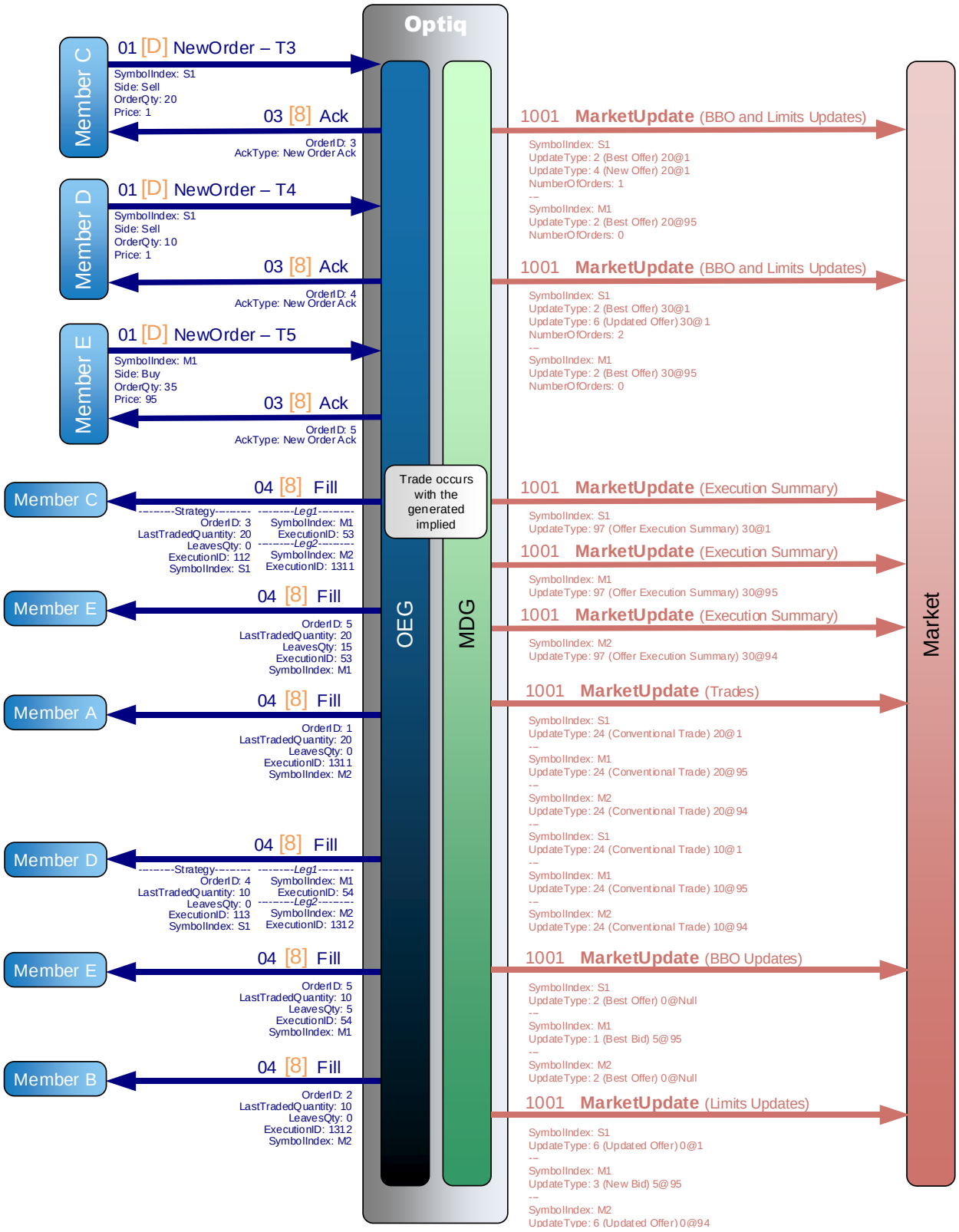
No message is sent to the market.

6.6 IMPLIEDS WITH SIM: COMPONENT IMPLIED VERSUS EXPLICIT ORDER

In this scenario, the SIM Model is activated for both contract and strategy S1.

The Books of this example are represented in their final states just before the incoming explicit order of the Member E triggers trades.





Member C sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 20 and a price of 1 in the strategy book S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Since the explicit order enters the order book without matching, a public **MarketUpdate** (1001) message is sent to create the new Limit and update the BBO of the strategy book S1.

The new explicit order automatically triggers generation of a component Implied in the order book M1, based on the state of order books S1 and M2 at that time.

The generated Implied creates volume of 20 at the price of 95 which is published to the Market via a public **MarketUpdate** (1001) message. Since there is no explicit Offer order within the outright book M1, the field "number of orders" of the **MarketUpdate** (1001) message is set to zero, even if there is an Implied volume. No additional messages are sent for instrument M2.

Member D sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 10 and a price of 1 in the strategy book S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

A public **MarketUpdate** (1001) messages are sent to update the Limit and the BBO of the strategy book S1.

The new explicit order automatically triggers generation of a component Implied in the outright book M1, based on the state of order books S1 and M2 at that time.

The generated Implied is equal to the Best Offer in the book of outright M1, and thus increases the existing Implied volume by 10 at the price of 95, for which a new public **MarketUpdate** (1001) message is published.

Member E sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 35 and a price of 95 in the outright book M1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The new explicit Buy Order from Member E partially matches the Implied Offer (identified in the book diagram by underlined T3) and OEG sends back a private **Fill** (04) (FIX 8) message for each order that participate in all generated trades, to each member that participated in the trade execution.

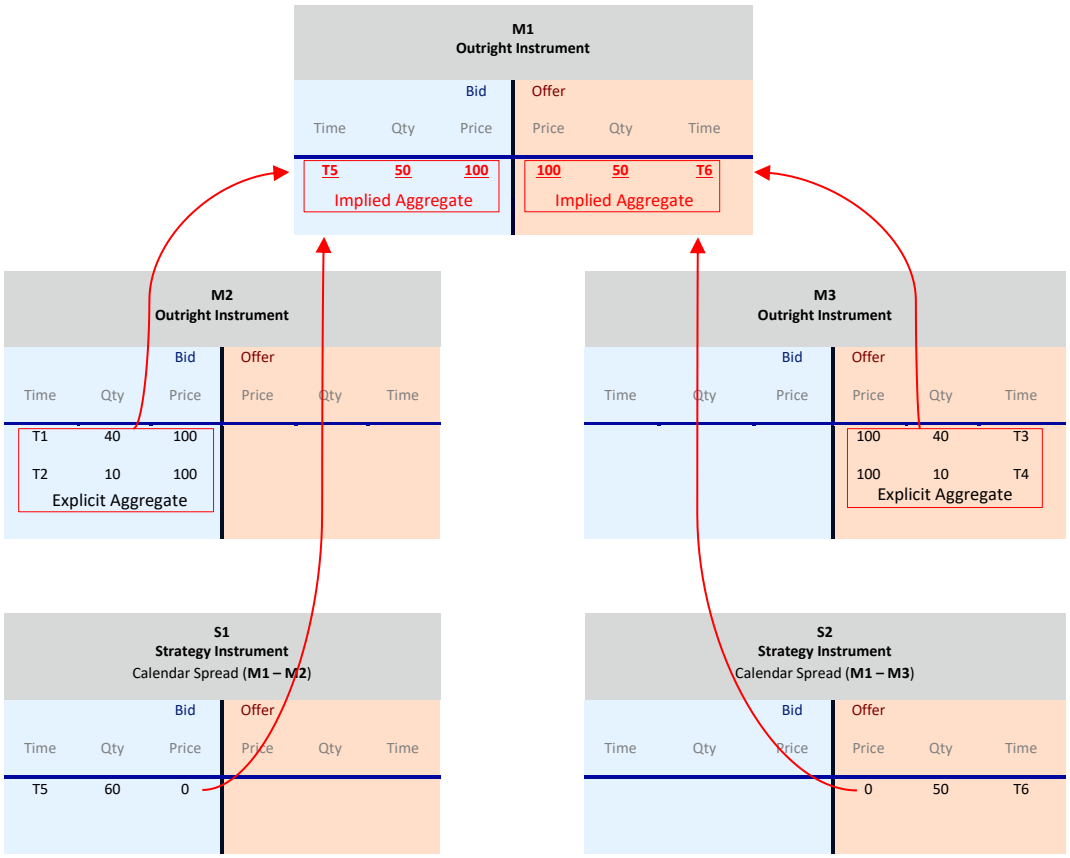
Public **MarketUpdate** (1001) messages for Execution Summary is sent to the market for each outright and for the strategy.

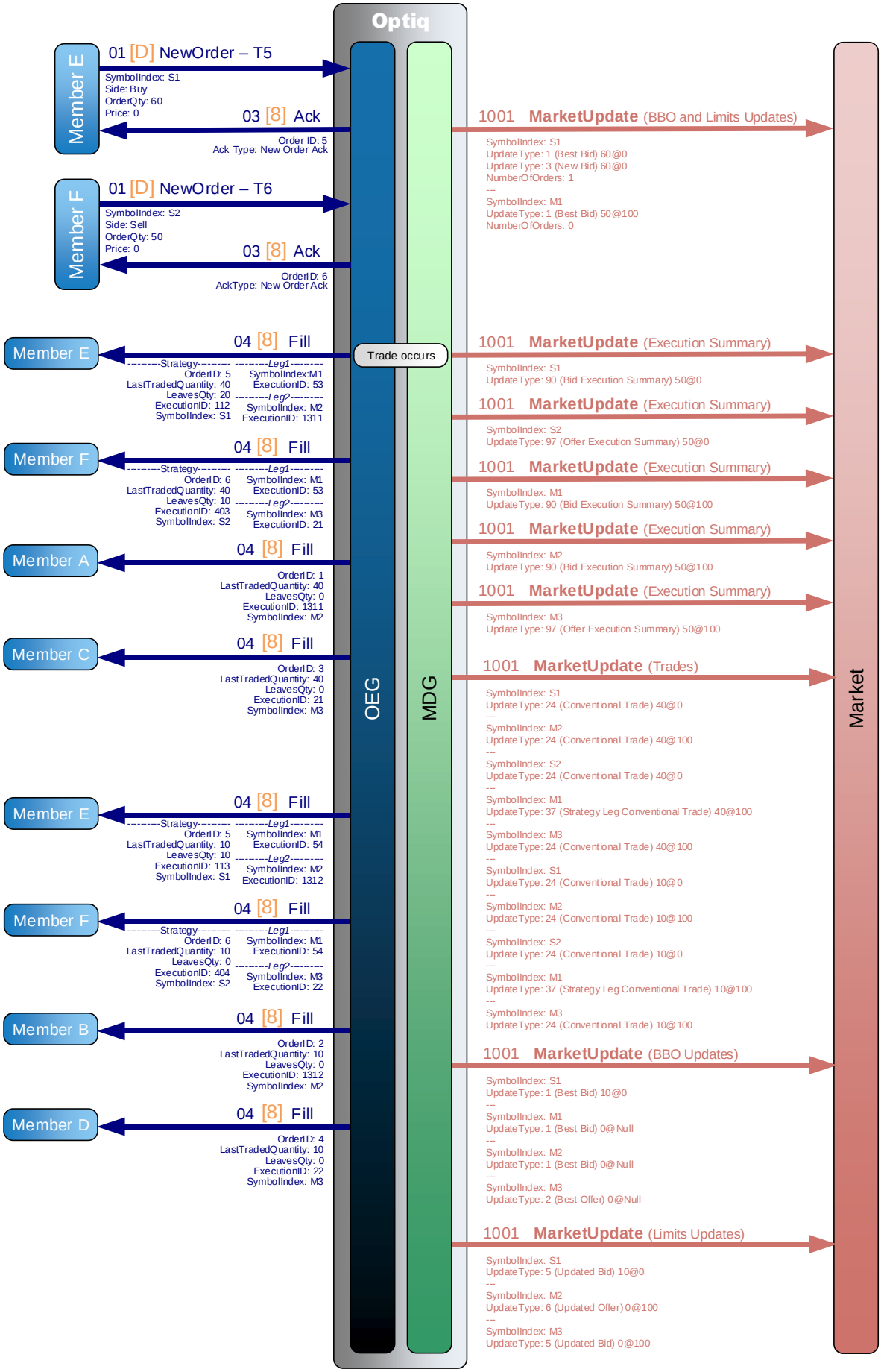
Following this, a public **MarketUpdate** (1001) message (Conventional Trade) is sent to the market for the Trade in the Strategy S1 and Trades for each Outright (M1 and M2).

Following publication of updates for the Strategy and Outrights, another set of **MarketUpdate** (1001) messages are sent for BBO and Limits updates.

Note: The ordering of BBO and limits updates between instruments sent in the same **MarketUpdate** (1001) is not guaranteed.

6.7 IMPLIEDS WITH SIM: COMPONENT IMPLIED VERSUS COMPONENT IMPLIED





Member E sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 60 and a price of 0 in the strategy book S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The new explicit order automatically triggers generation of a component Implied in the outright book M1, based on the state of order books S1 and M2 at that time.

The generated Implied is the Best Bid in the outright book M1, and thus creates a volume of 50 at the price of 100.

Since the explicit order enters the order book without matching, a public **MarketUpdate** (1001) message is sent to create the new Limit and update the BBO of the strategy book S1.

A public **MarketUpdate** (1001) message is sent in order to materialize the increase of volume due to the generation of the implied in the outright book M1. Since there is no explicit order within the outright book M1, the field number of orders of the **MarketUpdate** (1001) message is set to "0", even if there is an Implied volume.

Member F sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 50 and a price of 0 in the strategy book S2.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

A public **MarketUpdate** (1001) message is sent to create the new Limit and update the BBO of the strategy book S2.

The new explicit order automatically triggers generation of a component Implied in the outright book M1, based on the state of order books S2 and M3 at that time.

The generated Implied is the Best Offer in the outright book M1, and thus creates a volume of 50 at the price of 100.

The new generated Implied (Offer Implied) fully matches the Bid Implied and OEG sends back a private **Fill** (04) (FIX 8) message per order and per trade the order is involved in to each member in order to publish the trade execution.

Public **MarketUpdate** (1001) messages for Execution Summary are sent to the market for each outright and strategy.

Following this, public **MarketUpdate** (1001) message (Conventional Trade) is sent to the market for the Trades in each Strategy (S1 and S2) and Trades for each Outright (M2 and M3) excepted for M1 where it is a Strategy Leg Conventional Trade which is sent instead.

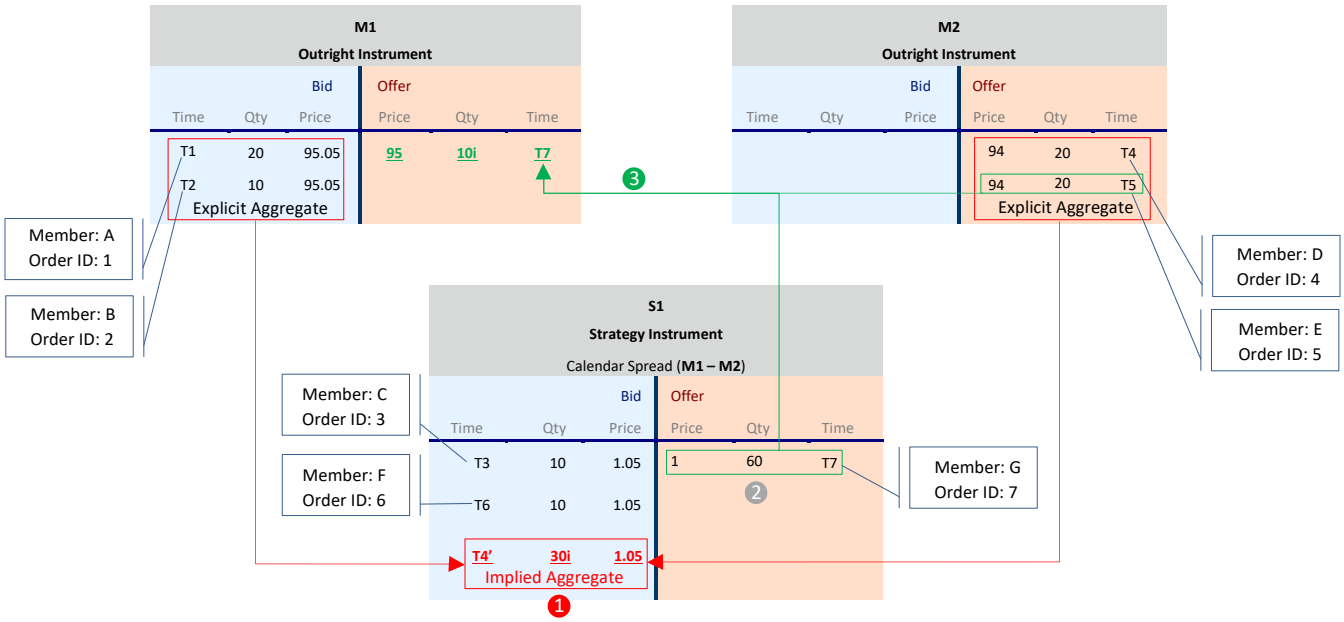
Following publication of updates for the Strategies and Outrights, another set of **MarketUpdate** (1001) messages are sent for BBO and Limits updates.

Note: The ordering of BBO and limits updates between instruments sent in the same **MarketUpdate** (1001) is not guaranteed.

6.8 IMPLIEDS WITH SIM: STRATEGY IMPLIED VERSUS EXPLICIT ORDER ON A STRATEGY BOOK

For a readability purpose, the steps of this kinematic scenario are split into multiple diagrams.

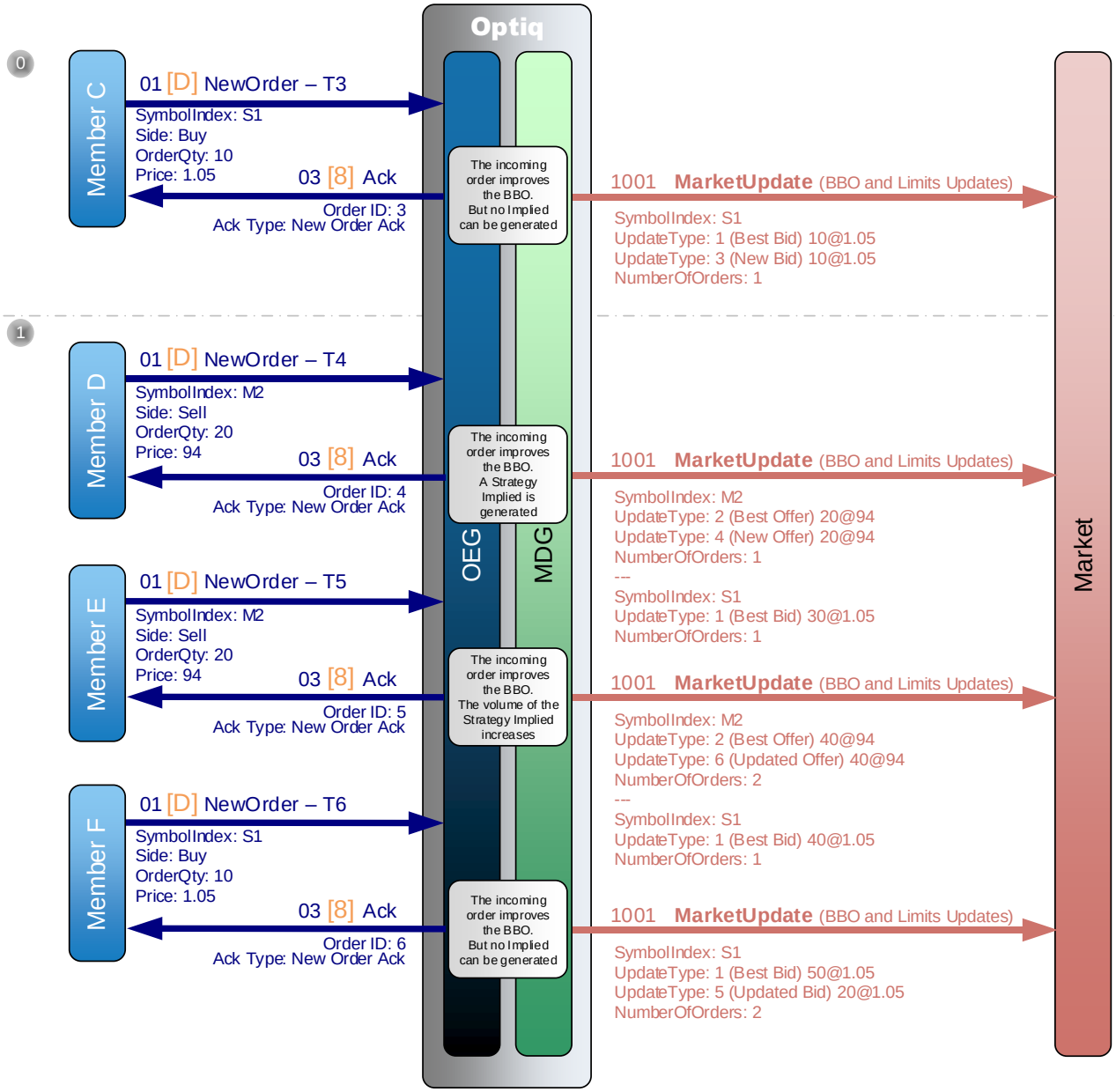
Additionally, a table provides a summary of the main events of this example to clarify the origin and rule for the timestamp used.



Within the table, we distinguish two kinds of timestamps:

- The timestamp related to an order entry (e.g. T1, …, Tn);
- The timestamp related to an Implied preceded by the formula from which it has been calculated.

			Order				
	Timestamp	Member	Order Category	Order ID	Price	Quantity	Implied generated due to the order entry
Phase 0	T1	A	Explicit Order	1	95.05	20	**No**
	T2	B	Explicit Order	2	95.05	10	**No**
	T3	C	Explicit Order	3	1.05	10	**No**
Phase 1	T4	D	Explicit Order	4	94	20	**Yes**
	Max(Min(T1,T2);Min(T4)) = T4'		Strategy Implied		1.05	**20i**	
	T5	E	Explicit Order	5	94	20	**Yes**
	Max(Min(T1,T2);Min(T4,T5)) = T4'		Strategy Implied		1.05	10i + **20i**	
	T6	F	Explicit Order	6	1.05	10	**No**
Phase 2	T7	G	Explicit Order	7	1	60	**Yes**
Phase 3	Max(Min(T5);Min(T7)) = T7		Component Implied		95	10i	



⓪ Member C sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 10 and a price of 1.05 in the strategy book S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Because it improves the BBO, the new explicit order automatically triggers the calculation of Implieds. At this stage, no implied can be generated.

Since the explicit order enters the order book without matching, a public **MarketUpdate** (1001) message is sent to create the new Limit and update the BBO of the strategy book S1.

① Member D sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 20 and a price of 94 in the Outright book M2.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The new explicit order automatically triggers generation of a strategy Implied in the book S1, based on the state of order books M1 and M2 at that time.

The generated Implied is at the BBO in the strategy book S1, and thus creates a volume of 20 at the price of 1.05.

Since the explicit order enters the order book without matching, a public **MarketUpdate** (1001) message is sent to create the new Limit and update the BBO of the order book M2.

A public **MarketUpdate** (1001) message is sent in order to materialize the increase of volume due to the generation of the implied in the strategy book S1. At this step, since there is only one explicit order within the Bid side of the order book S1, the field “number of orders” of the **MarketUpdate** (1001) message is set to “1”, despite the presence of the Bid strategy Implied.

Member E sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 20 and a price of 94 in the Outright book M2.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The new explicit order automatically triggers generation of a strategy Implied in the book S1, based on the state of order books M1 and M2 at that time.

The generated Implied is at the BBO in the strategy book S1, and thus creates an additional volume of 10 at the price of 1.05.

Since the explicit order enters the order book without matching, a public **MarketUpdate** (1001) message is sent to update the Limit and the BBO of the book outright M2.

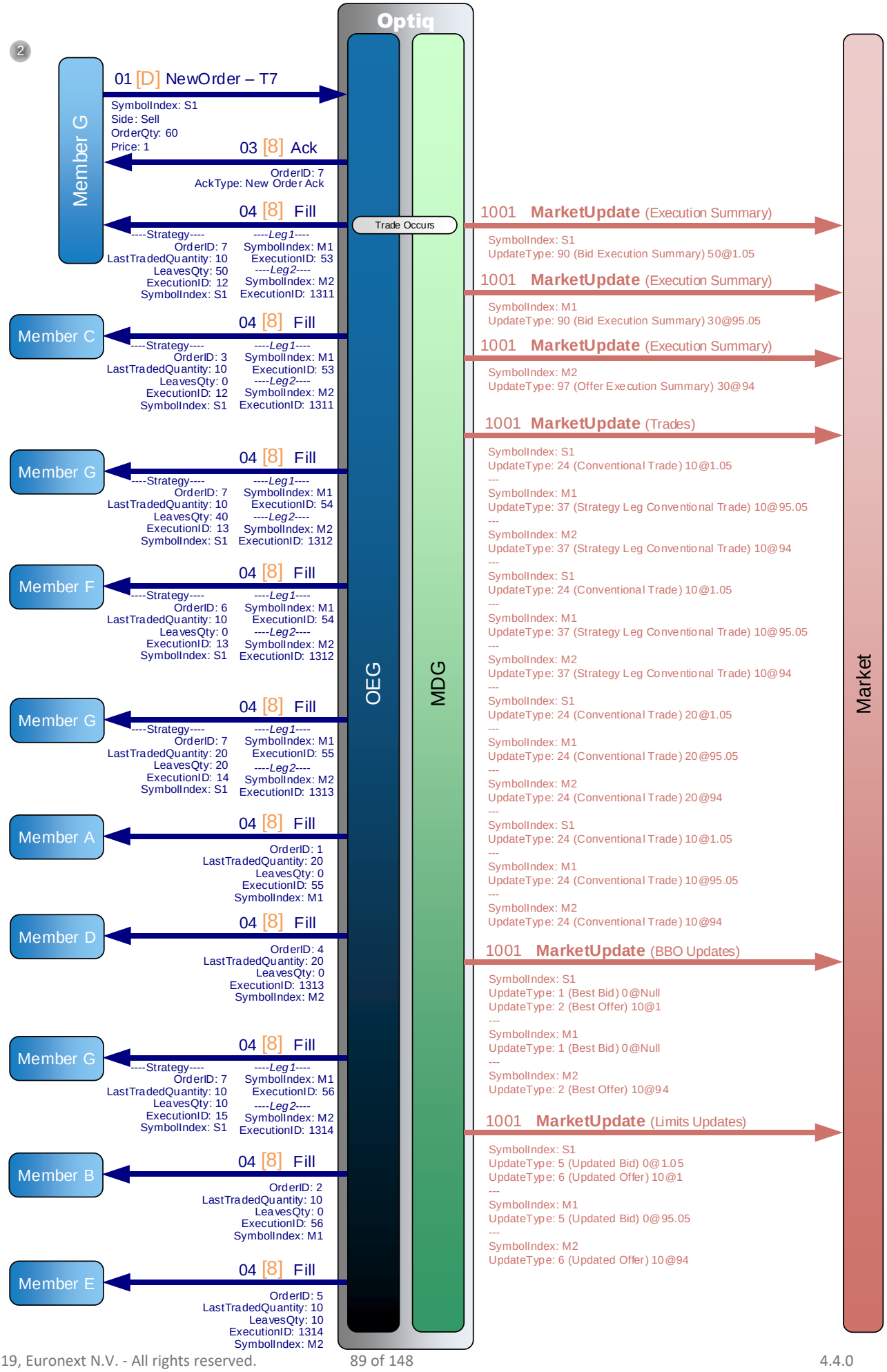
A public **MarketUpdate** (1001) message is sent in order to materialize the increase of volume due to the generation of the implied in the strategy book S1. At this step, despite the additional volume due to the aggregate Bid strategy Implied, the value of the field “number of orders” of the **MarketUpdate** (1001) message remains “1”.

Member F sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 10 and a price of 1.05 in the strategy book S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Because it improves the BBO, the new explicit order automatically triggers the calculation of Implieds. However, no new Implied can be generated.

A public **MarketUpdate** (1001) message is sent to update the Limit and the BBO of the order book S1 and the value of the field “number of orders” of the **MarketUpdate** (1001) message is incremented to “2”.



② Member G sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 60 and a price of 1 in strategy book S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The new Offer explicit order (Order ID = 7) immediately matches:

- the Strategy's Bid orders:
  - for a quantity of 10 with the Member C's explicit order (Order ID = 3)
  - for a quantity of 10 with the Member F's explicit order (Order ID = 5)

- the aggregate strategy Implied:
  - for a quantity of 20 and then for a quantity of 10

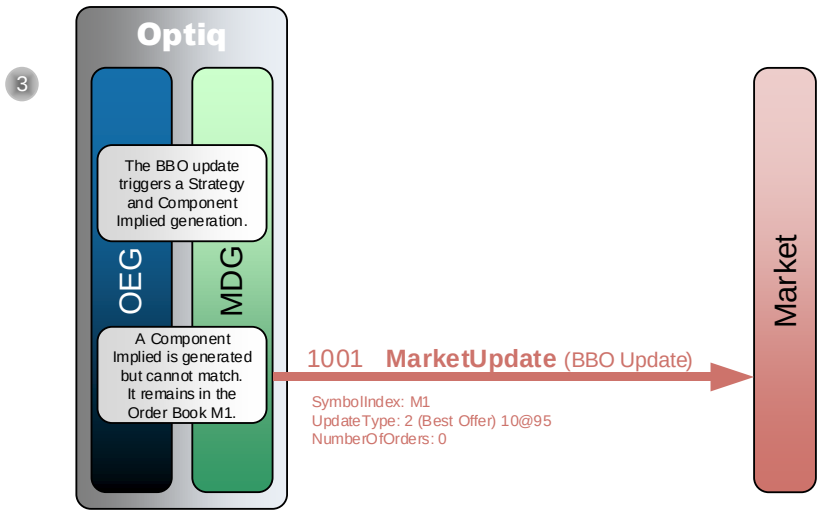
and OEG sends back a private **Fill** (04) (FIX 8) message to each member to publish the trade execution.

Public **MarketUpdate** (1001) messages of Execution Summary are sent to the market for the strategy and for each outright.

For explicit versus explicit orders, public **MarketUpdate** (1001) messages (Conventional Trade) are sent to the market for the Trade in the strategy (S1) and trades for each leg of the strategy (i.e. the Trades for the individual outrights) that are flagged as "Strategy Leg Conventional Trade".

For implied versus explicit order, public **MarketUpdate** (1001) messages (Conventional Trade) are sent to the market for the Trades in the strategy (S1) and for each outright.

Following publication of trades for the strategy and outrights, another set of **MarketUpdate** (1001) messages are sent for BBO and Limits updates.



③ The BBO update triggers an Implied calculation. One Offer component Implied is created in the order book M1.

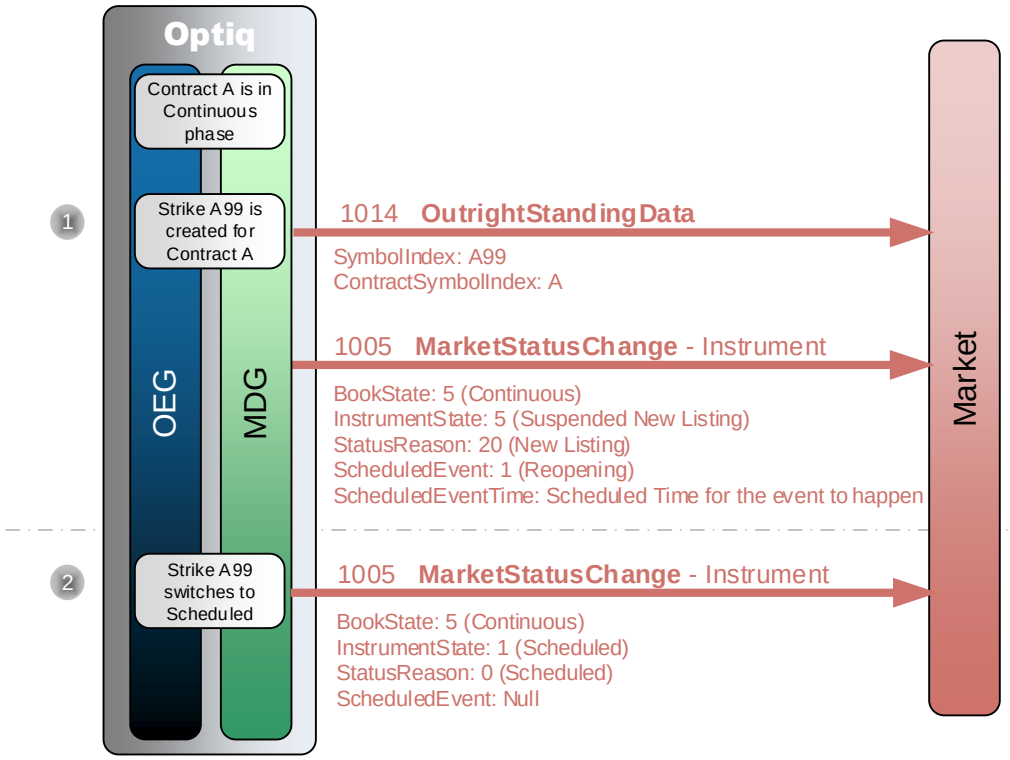
The generated Implied is the Best Offer in the order book M2, and thus creates a volume of 10 at the price of 95.

A public **MarketUpdate** (1001) message is sent to inform the market of the increase of volume due to the new Implied. The value of the field "number of orders" of the **MarketUpdate** (1001) message is set to "0".

Note: The ordering of BBO and limits updates between instruments sent in the same **MarketUpdate** (1001) is not guaranteed.

7. INTRADAY INSTRUMENT CREATION

7.1 INTRADAY STRIKE CREATION

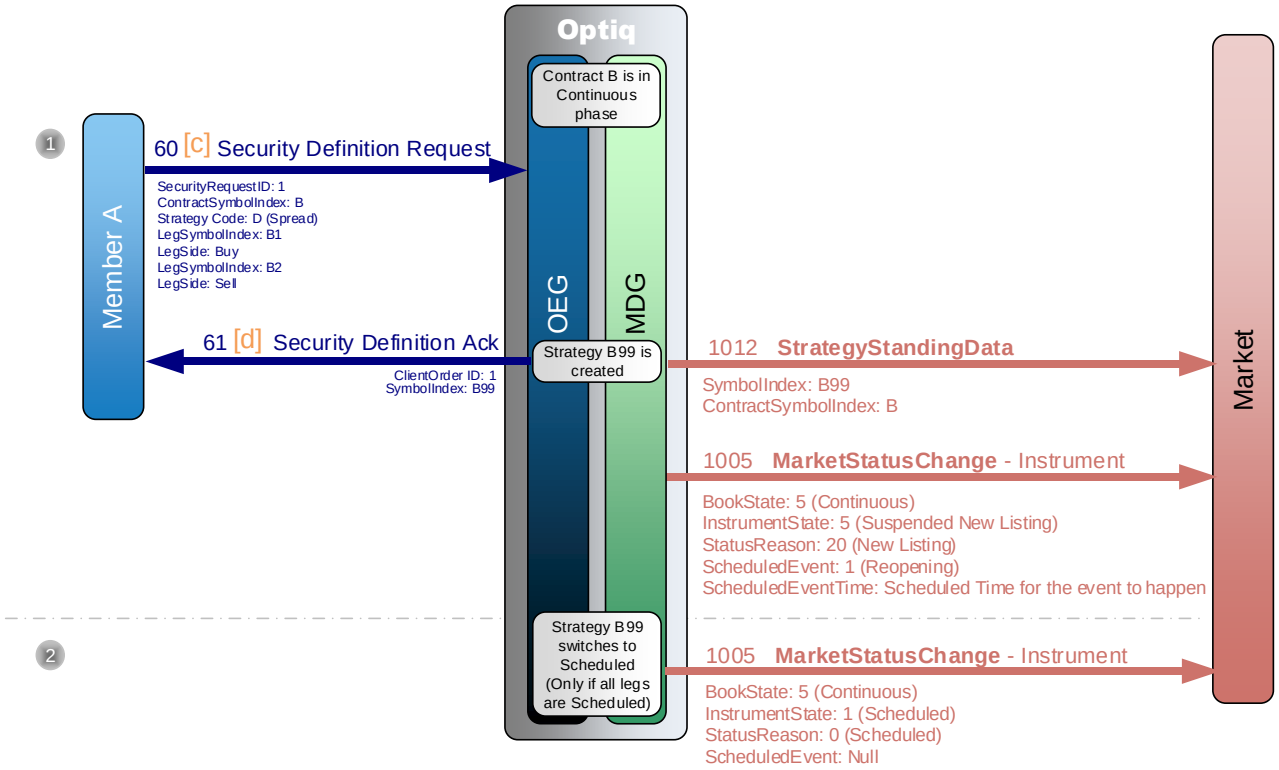


① Optiq creates a new Strike (A99) for Contract A. A public **OutrightStandingData** (1014) message is sent to the market, with the referential details associated to this new Strike. Creation of new instruments requires a brief period of new instrument configuration. As such, creation message is followed by a public **MarketStatusChange** (1005), as the strike is created in suspended state "Suspended New Listing", and the expected time of the strike becoming available for trading within the *ScheduledEventTime* field.

② Upon the strike A99 becoming available for trading, a public **MarketStatusChange** (1005) is sent to the market to indicate that Strike A99 is now in state Scheduled and follows the pattern and state of the Contract A.

7.2 INTRADAY STRATEGY CREATION

7.2.1 Intraday Strategy Creation Accepted



① Intraday strategy creation is initiated by the members by submitting a **SecurityDefinitionRequest** (60) (FIX c) message to the OEG.

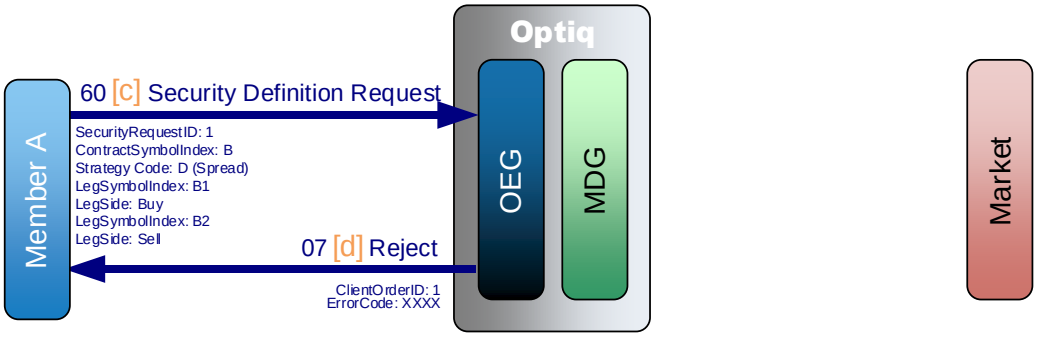
OEG responds with the **SecurityDefinitionAck** (61) (FIX d) message, to provide member with the Symbol Index of the strategy.

If the strategy does not yet exist, creation is communicated to the market by sending a public **StrategyStandingData** (1012) message. Creation of new instruments requires a brief period of price and leg status identification. As such, creation message is followed by the **MarketStatusChange** (1005) message for that strategy, with the state of "Suspended New Listing", and the expected time of the strategy becoming available for trading within the *ScheduledEventTime* field.

② If the strategy does not yet exist, upon identification of the appropriate price conditions for the strategy it will be made available for trading, at which point a new public **MarketStatusChange** (1005) message for that strategy is sent to the market with the state of the strategy being set to "Scheduled", which will follow the overall state of the contract.

Note: In case a member submits a **SecurityDefinitionRequest** (60) FIX (c) for a strategy that already exists in Optiq, OEG sends back a private **SecurityDefinitionAck** (61) FIX (d) with the Symbol Index of the existing Strategy. No messages are disseminated to the Market.

7.2.2 Intraday Strategy Creation Rejected



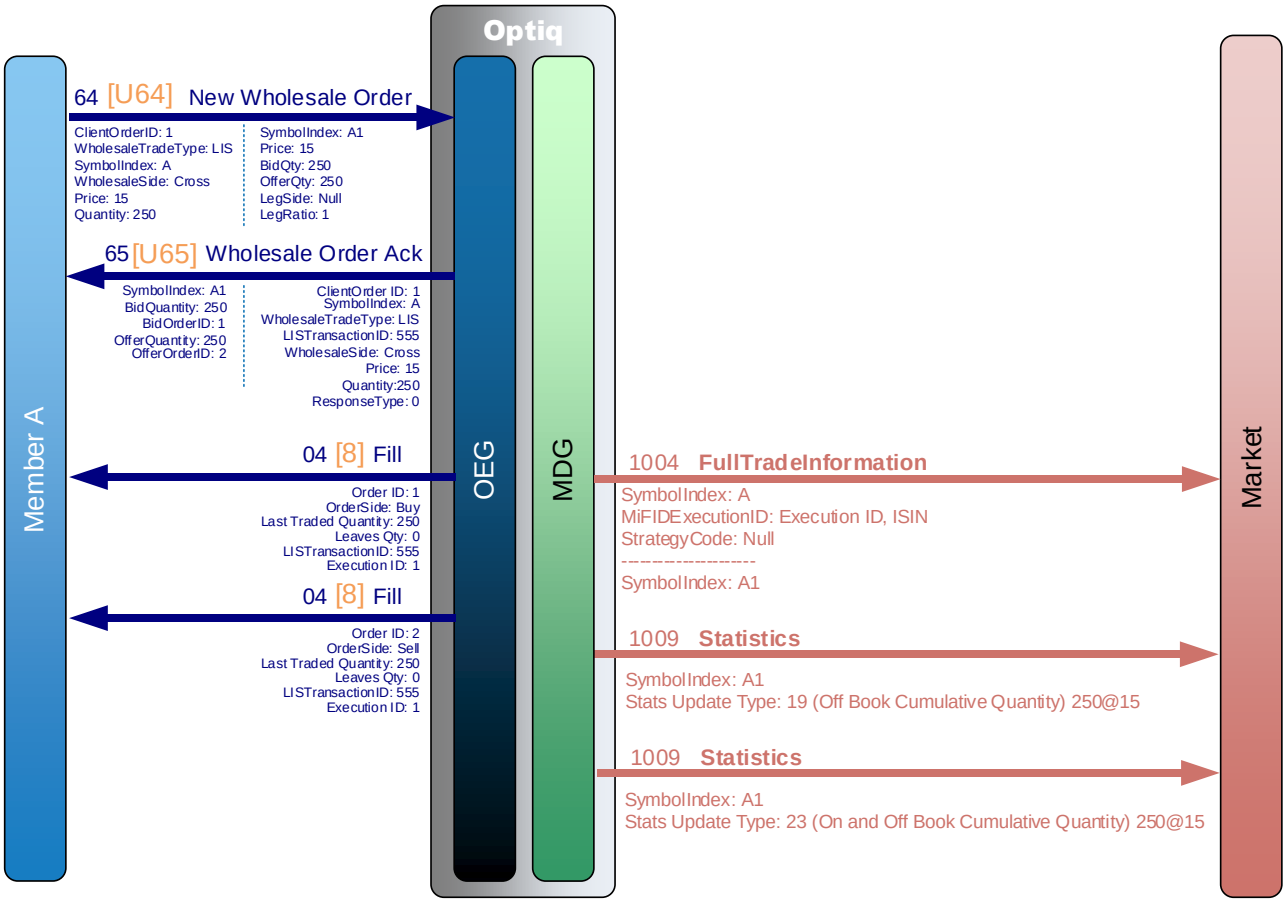
Member A sends a private **SecurityDefinitionRequest** (52) (FIX c) message to create a strategy.

If the message is rejected OEG sends back a private **Reject** (07) (FIX d) message with an *Error Code*. The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*.

No message is sent to the market.

8. WHOLESALES

8.1 CROSS ON AN OUTRIGHT



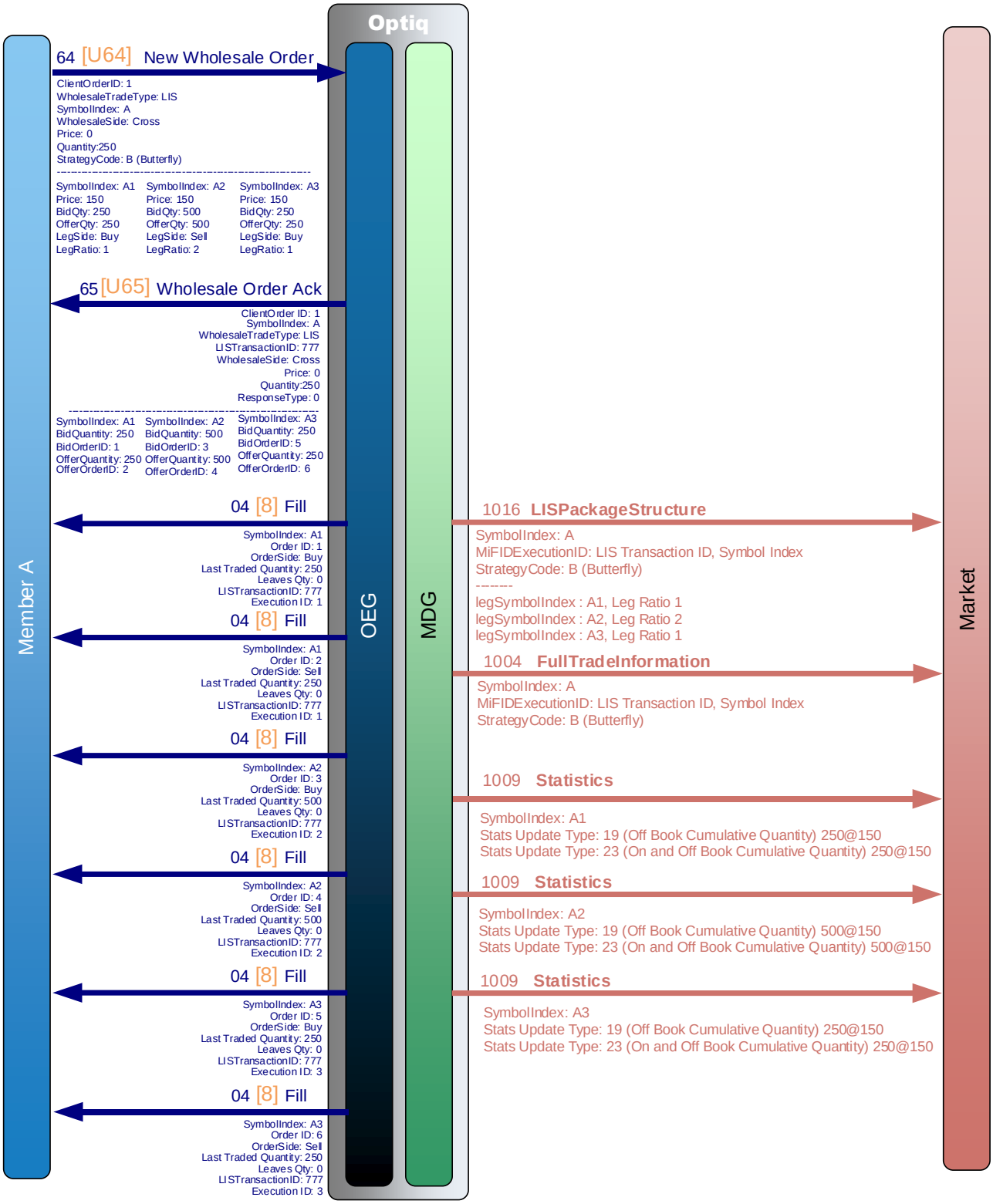
Member A sends a private **NewWholesaleOrder** (64) (FIX U64) message to initialize a new Wholesale transaction on the instrument A1, providing both sides of the transaction, as a Cross order.

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Wholesale Order, with the field *Response Type* set to 0 = Accept, and provides the system generated *LISTransactionID*.

The transaction results in immediate match and the OEG generates a private **Fill** (04) (FIX 8) message for each leg of the trade.

A public **FullTradeInformation** (1004) message is sent to the market for the transaction. This is followed by two **Statistics** (1009) messages sent to update the statistics of the Cumulative Quantity.

8.2 CROSS ON A STRATEGY



Member A sends a private **NewWholesaleOrder** (64) (FIX U64) message to initialize a new Wholesale transaction to Instrument in Contract A, on a strategy. The submission contains the information to setup the strategy for the transaction both sides of the transaction.

In this example the strategy submitted is a Butterfly, and the details of the strategy submitted match the defined structure and characteristics of the strategy.

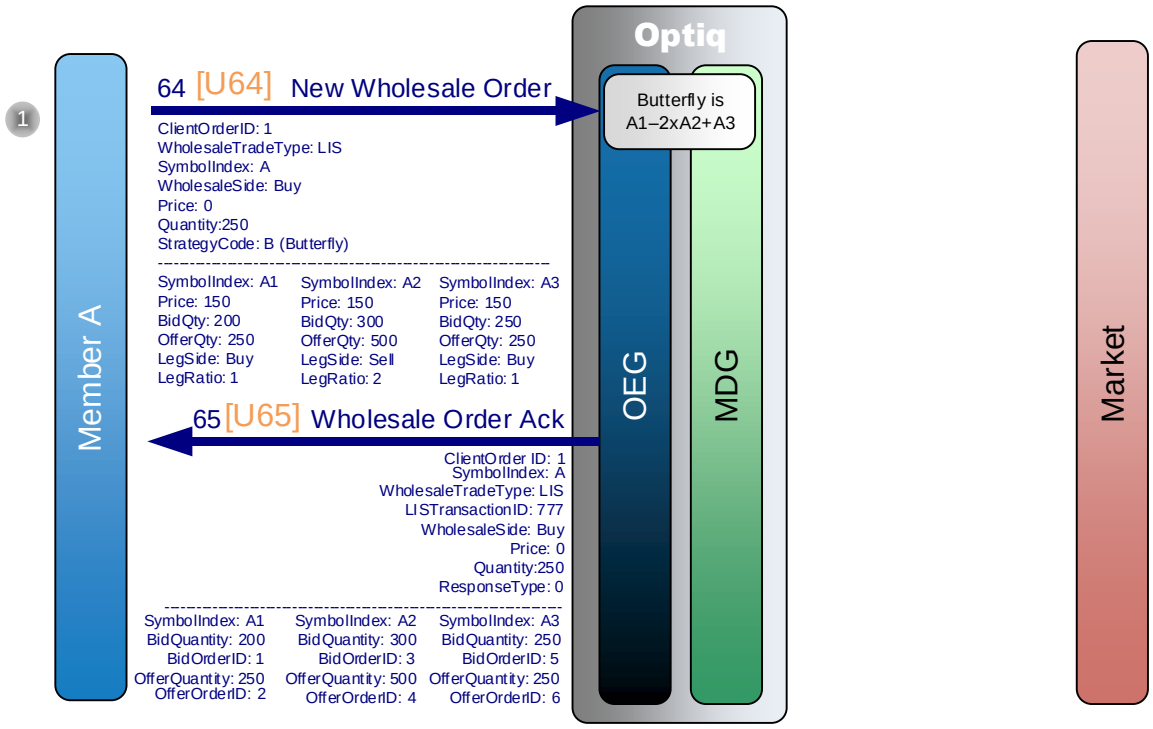
OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Wholesale Order, with the field *Response Type* set to 0 = Accept, and provides the system generated *LISTransactionID*.

The transaction results in immediate match in the strategy and the OEG generates a private **Fill** (04) (FIX 8) message for each Outright leg of the trade.

A public **LISPackageStructure** (1016) associated to its **FullTradeInformation** (1004) message are sent to the market for the transaction for instrument in Contract A. This is followed by **Statistics** (1009) messages sent to update the statistics of the Cumulative Quantity, for each individual Outright leg of the Strategy.

8.3 NEW WHOLESALE ORDER ON STRATEGY FOR OPTIONS

The steps of this kinematics scenario are split into multiple diagrams. Also, for readability purposes, **Fill** (04) messages seems to be sent first for Member B and then for Member A. In reality those messages are sent in the same time for Member A and Member B.

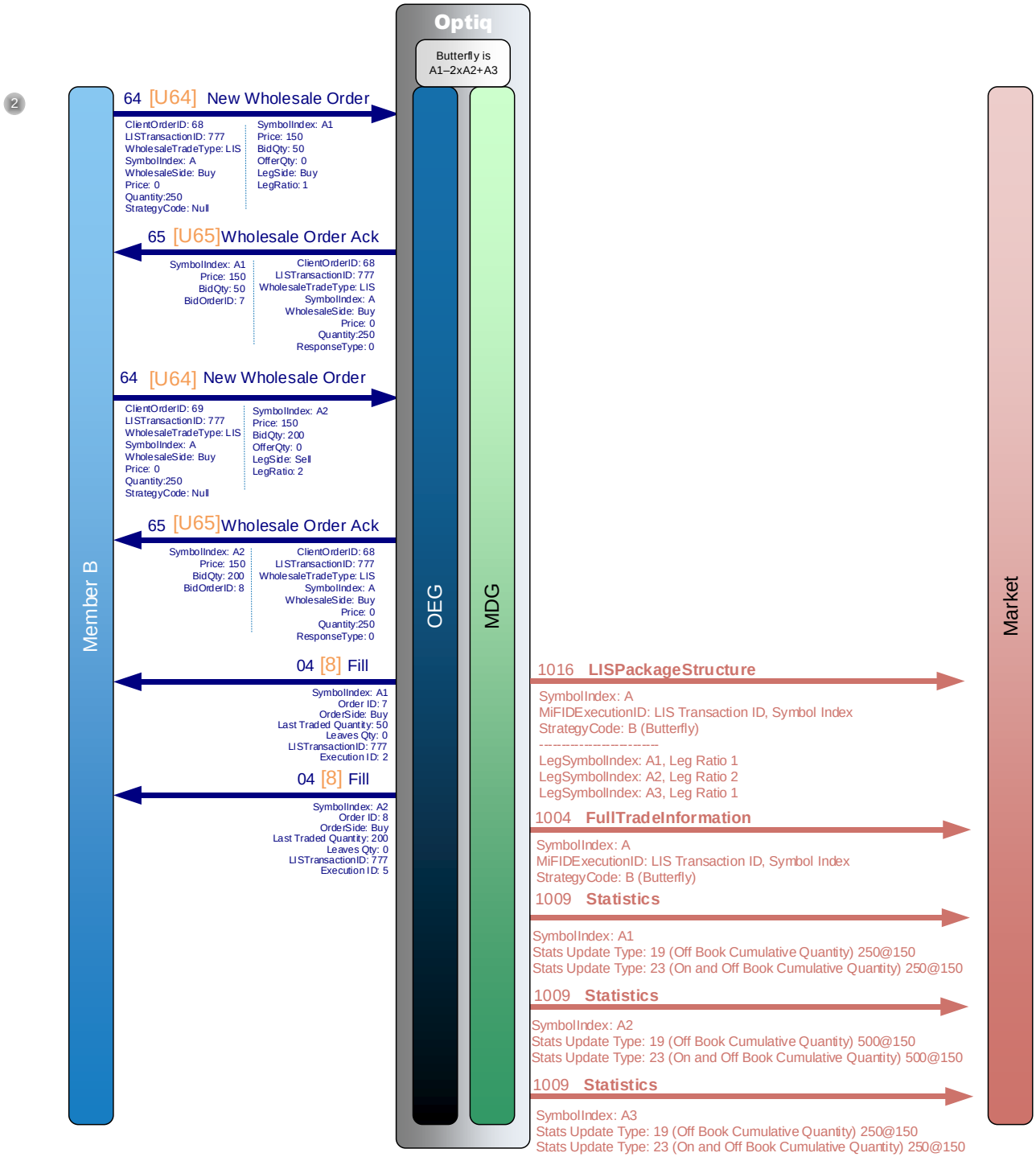


① Member A sends a private **NewWholesaleOrder** (64) (FIX U64) message to initialize a new Wholesale transaction on Instrument in Contract A, on a strategy for an Option. The submission contains the information to setup the strategy for the transaction both sides of the transaction.

In this example the strategy submitted is a Butterfly, and the details of the strategy submitted match the defined structure and characteristics of the strategy. Part of the submitted transaction contains crossed quantity on instrument A3, and the remaining legs are awaiting a Reactor to complete the transaction.

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Wholesale Order, with the field *Response Type* set to 0 = Accept, and provides the system generated *LISTransactionID*.

No message is sent to the market.



② Member B sends a private **NewWholesaleOrder** (64) (FIX U64) message to respond to a Wholesale transaction on Instrument in Contract A, on a strategy. The submission contains the information to setup the strategy for the transaction both sides of the transaction.

In this example the strategy submitted is a Butterfly, and the details of the strategy submitted match the defined structure and characteristics of the strategy. This response is targeted as a Reaction to the declaration done in Step 1 by Member A, and as such contains the *LISTransactionID* that was generated by system and sent to Member A, and was communicated between the two members.

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to Member B to confirm the successful receipt and technical processing of the Wholesale Order.

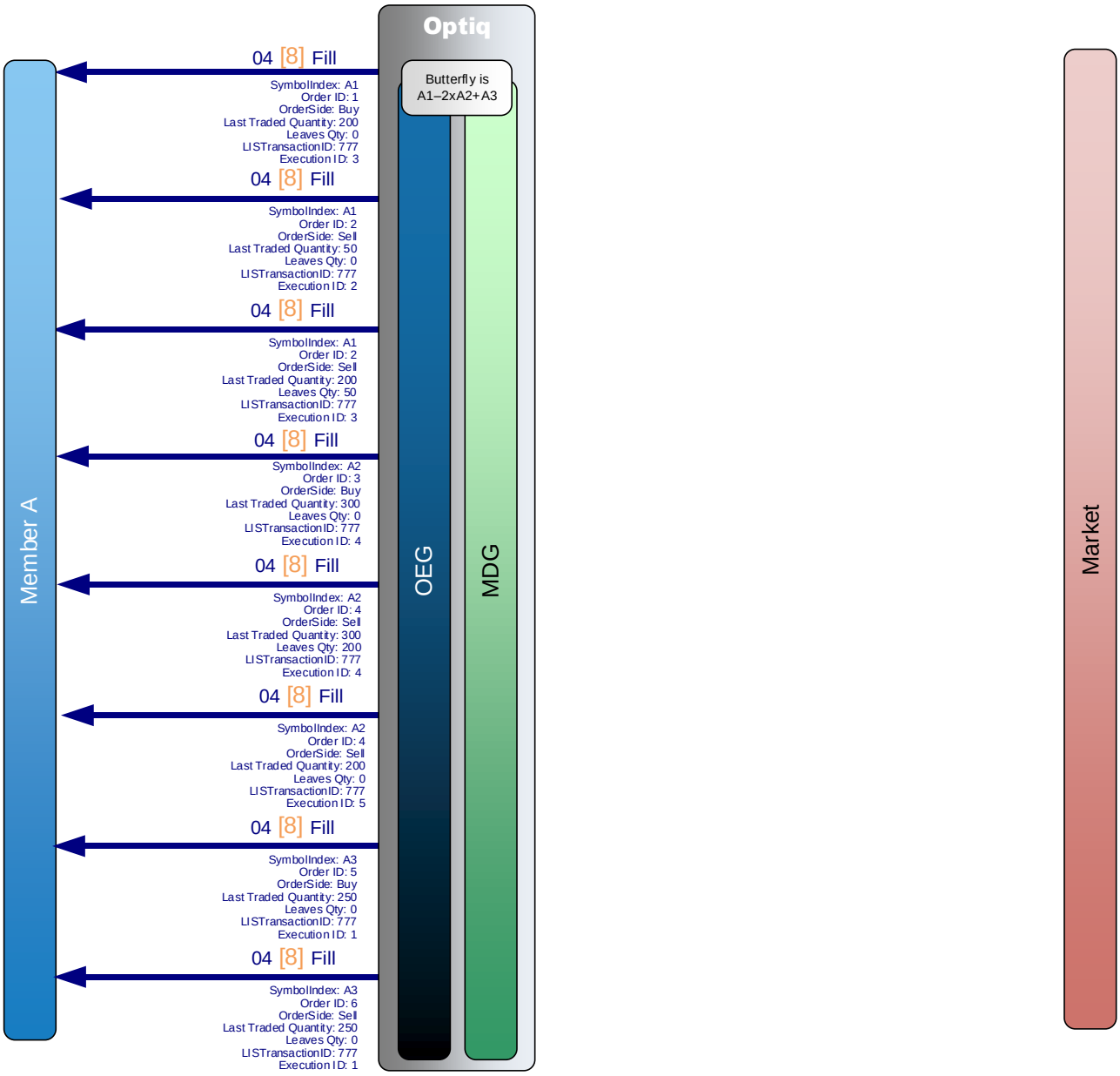
The submission by Member B completes the transaction and results in immediate match for all the legs of the strategy.

OEG generates a private **Fill** (04) (FIX 8) message for each Outright leg of the strategy.

The diagram above displays the two Fill messages sent to Member B, for the match in leg A1 and A2.

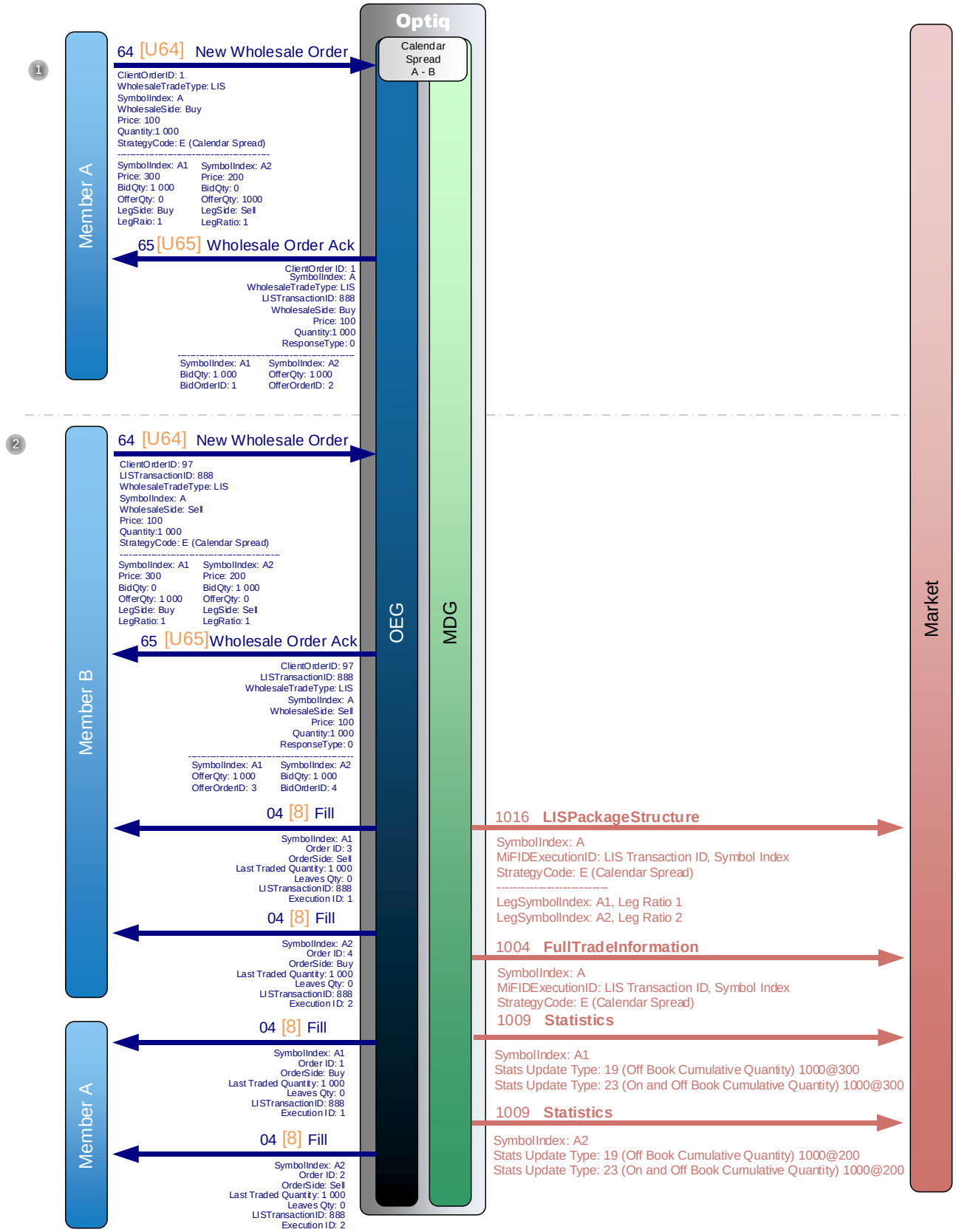
For readability the Fill messages to Member A are shown in the following diagram.

A public **LISPackageStructure** (1016) message associated to its **FullTradeInformation** (1004) message are sent to the market for the transaction on Instrument for Contract A. This is followed by **Statistics** (1009) messages sent to update the statistics of the Cumulative Quantity, for each individual Outright leg of the Strategy. This covers all the public messages for the transaction.



The diagram above displays the **Fill** (04) (FIX 8) message sent to Member A for the crossed quantity on all the submitted legs, as well as the matches against Member B on the A1 and A2. The associated public messages are described in the diagram above.

8.4 NEW WHOLESALE ORDER ON STRATEGY FOR FUTURES



① Member A sends a private **NewWholesaleOrder** (64) (FIX U64) message to initialize a new Wholesale transaction on Instrument in Contract A, on a strategy for a Future. The submission contains the information to setup the strategy for the transaction both sides of the transaction.

In this example the strategy submitted is a Calendar Spread, and the details of the strategy submitted match the defined structure and characteristics of the strategy.

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Wholesale Order, with the field *Response Type* set to 0 = Accept, and provides the system generated *LISTransactionID*.

No message is sent to the market.

② Member B sends a private **NewWholesaleOrder** (64) (FIX U64) message to respond to a Wholesale transaction on Instrument in Contract A, on a strategy. The submission contains the information to setup the strategy for the transaction both sides of the transaction.

This response is targeted as a Reaction to the declaration done in Step 1 by Member A, and as such contains the *LISTransactionID* that was generated by system and sent to Member A, and was communicated between the two members.

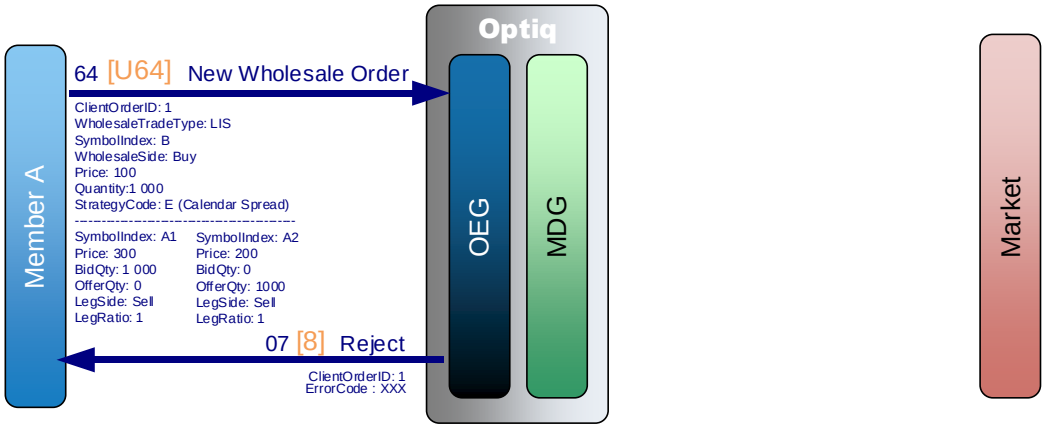
OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to Member B to confirm the successful receipt and technical processing of the Wholesale Order.

The submission by Member B completes the transaction and results in immediate match for all the legs of the strategy.

OEG generates a private **Fill** (04) (FIX 8) message for each leg of the strategy.

A public **LISPackageStructure** (1016) message, associated to its **FullTradeInformation** (1004) message are sent to the market for the transaction on Instrument in Contract A. This is followed by **Statistics** (1009) messages sent to update the statistics of the Cumulative Quantity, for each individual Outright leg of the Strategy.

8.5 REJECTION OF A NEW WHOLESALE ORDER



Member A sends a private **NewWholesaleOrder** (64) (FIX U64) message to initialize a new Wholesale transaction on a Calendar Spread strategy, providing the information to setup the strategy for the transaction, that does not match the expected structure of the strategy.

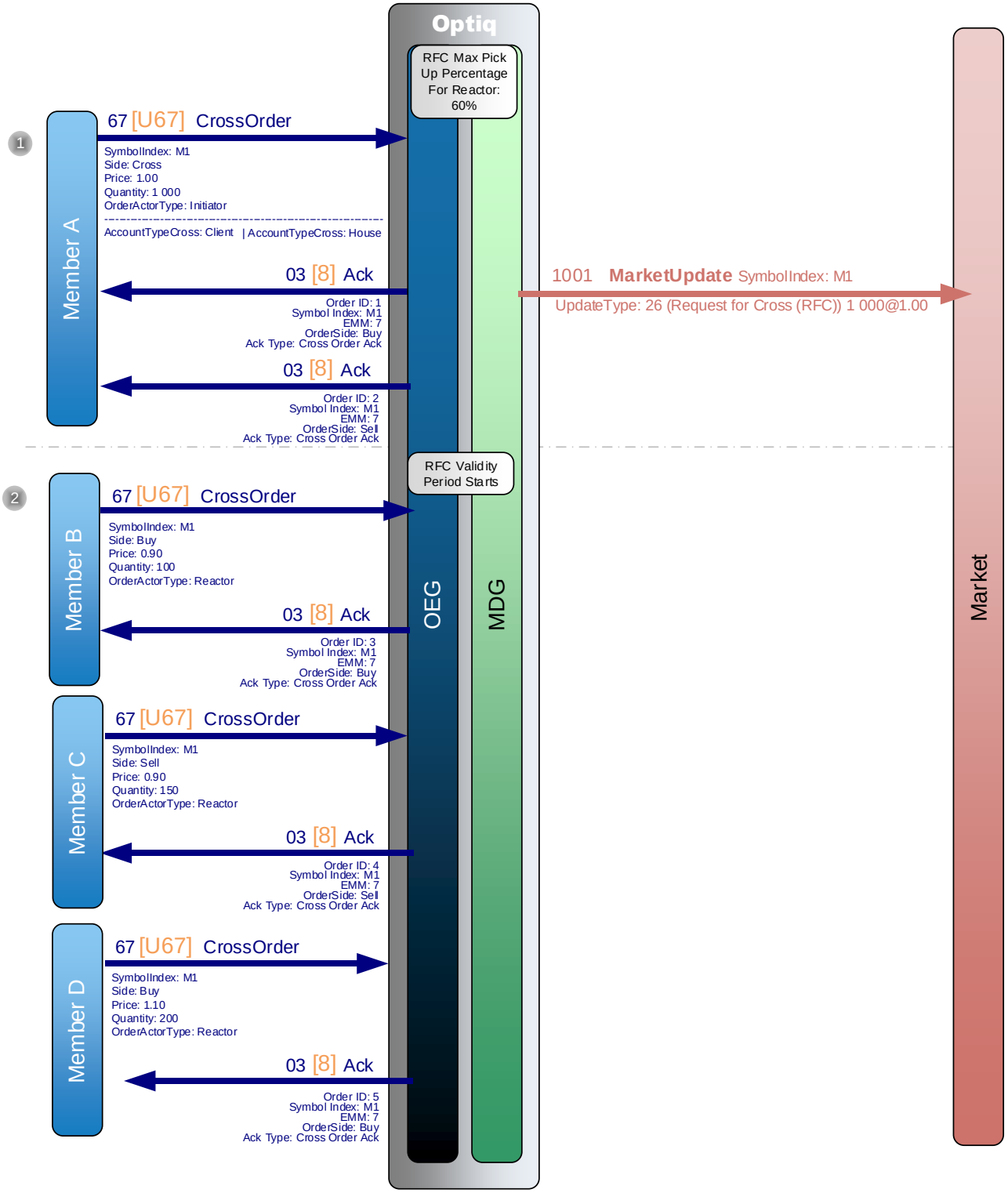
OEG sends back a private **Reject** (07) (FIX 8) message to reject the creation of the Wholesales transaction with an Error Code. The reason of the rejection can be found using the *Error Code* value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*.

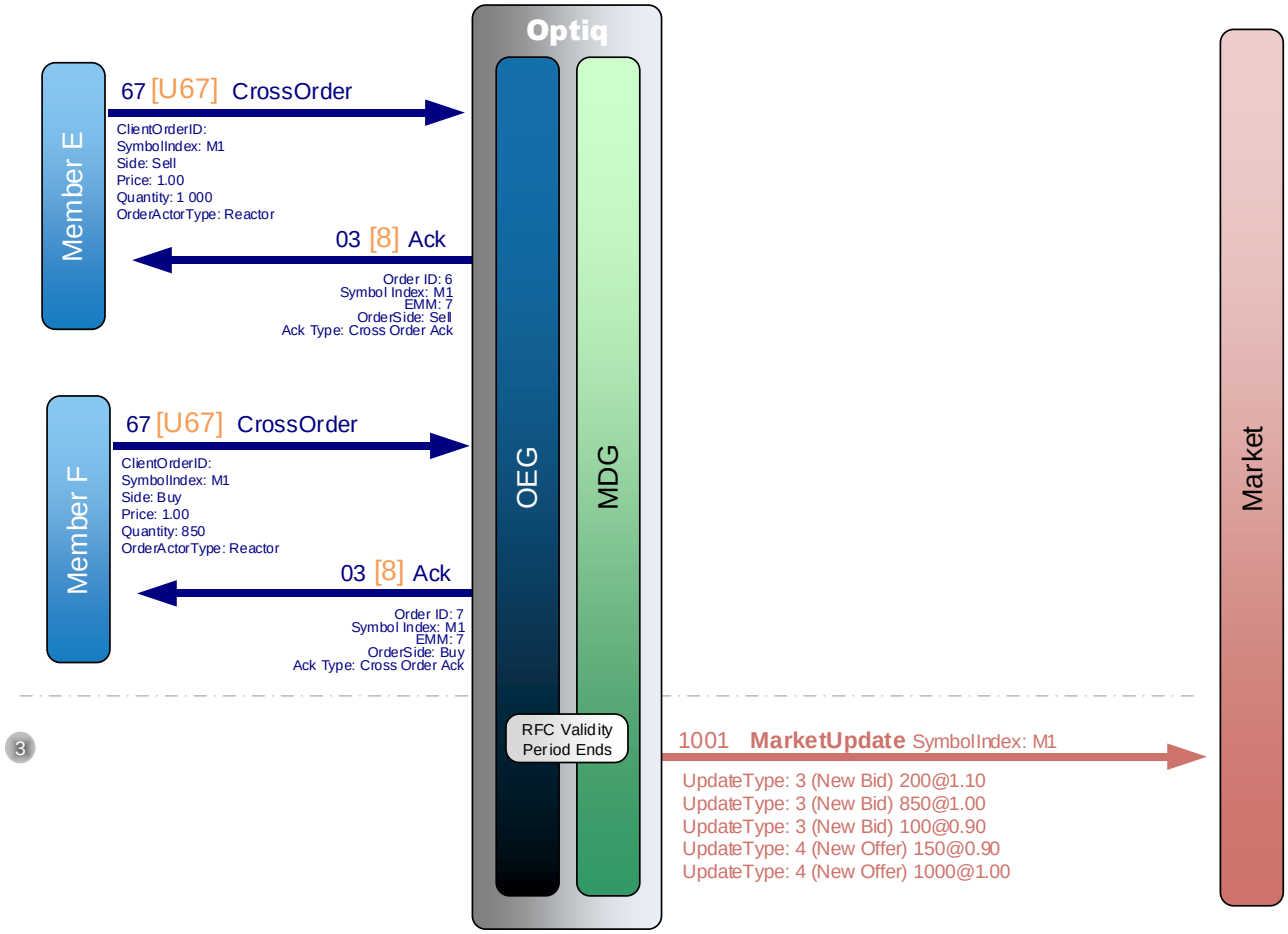
No message is sent to the Market.

9. REQUEST FOR CROSS

9.1 CLIENT BEST EXECUTION RFC – CLIENT VS HOUSE

For readability the scenario is presented in multiple diagrams.





Assumptions for this kinematics scenario:

- RFC Algorithm (parameter in standing data file): Client Best Execution RFC
- RFC Publication (parameter in standing data file): Yes
- RFC Max Pick Up Percentage For Reactor (parameter in standing data file): 60%
- First side provided in a Cross order always represents the Buy side, followed by the information for the Sell side
- Initiator sends a RFC with account type Client for Buy side and account type House for Sell side

① Member A sends a private **CrossOrder** (67) (FIX U67) message to initialize a new RFC transaction on the instrument M1 between a Client account type and a House account type.

OEG sends back two private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of both order.

A public **MarketUpdate** (1001) message is sent to the market with *Update Type* set to 26 = Request for Cross (RFC) which informs the market that there is a new RFC request that is available for improvement and matching.

② The RFC Validity Period starts.

Member B sends a private **CrossOrder** (67) (FIX U67) message to enter a new Buy order with a quantity of 100 and a price of 0.90.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Member C sends a private **CrossOrder** (67) (FIX U67) message to enter a new Sell order with a quantity of 150 and a price of 0.90.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Member D sends a private **CrossOrder** (67) (FIX U67) message to enter a new Buy order with a quantity of 200 and a price of 1.10.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Member E sends a private **CrossOrder** (67) (FIX U67) message to enter a new Sell order with a quantity of 1 000 and a price of 1.00.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Member F sends a private **CrossOrder** (67) (FIX U67) message to enter a new Buy order with a quantity of 850 and a price of 1.00.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Nothing is published for these reactor submissions to the market via public market data messages until RFC Improvement Period ends.

③ The RFC Validity Period end.

Public **MarketUpdate** (1001) messages are sent to the market to update the Limits for all of the reactor submissions.

System starts matching of all the reactor orders to the initiated RFC according to the steps of (*) Client Best Execution RFC allocation.

To illustrate the complex processing of the RFC algorithm of this case, and for readability purposes tables below provide details of how the reactor submissions are allocated against the initiator. This processing is completed instantaneously at the end of the RFC validity period.

The RFC book starts with the following state:

Initiator				Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**		**Price**	**Quantity**	**Firm**	**Timestamp**
T1	Member A **(Client)**	1000	1.00		1.00	1000	Member A **(House)**	T1
Reactors				**Bid**	**Ask**			
Timestamp	**Firm**	**Quantity**	**Price**		**Price**	**Quantity**	**Firm**	**Timestamp**
T4	Member D	200	1.10		0.90	150	Member C	T3
T6	Member F	850	1.00		1.00	1000	Member E	T5
T2	Member B	100	0.90					

Step 1: RFC Initiator Client improvement

A first allocation is done involving Member C and Member A (Client) of 150@0.90 as it improves price of a part of the Client side of the initiator RFC.

Initiator			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T1	Member A **(Client)**	~~1000~~ 850	1.00	1.00	1000	Member A **(House)**	T1
Reactors			**Bid**	**Ask**			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T4	Member D	200	1.10	0.90	~~150~~ 0	Member C	T3
T6	Member F	850	1.00	1.00	1000	Member E	T5
T2	Member B	100	0.90				

Step 2: Execution of reactors at RFC Price with RFC Initiator Client

This step takes into consideration the pick-up quantity for orders at the same price as the initiator. The resulting quantity is = 850 * 60% = 510.

As such the second allocation is done involving Member E and Member A (Client) of 510@1.00 .

Initiator			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T1	Member A **(Client)**	~~850~~ 340	1.00	1.00	1000	Member A **(House)**	T1
Reactors			**Bid**	**Ask**			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T4	Member D	200	1.10	1.00	~~1000~~ 490	Member E	T5
T6	Member F	850	1.00				
T2	Member B	100	0.90				

Step 3: RFC Initiator Non-Client (House) Improvement

Maximum possible quantity to match in this step for Non-Client (House) side of the initiated RFC for Member A against Reactors is the total quantity matched in previous steps 1 and 2 for the Client side of the initiated RFC (for Member A), which results in: 150 + 510 = 660.

This step of matching takes into consideration any reactor orders that improve or would be executed at the same price as the Non-Client RFC initiator side. This results in:

The third allocation is done involving Member D and Member A (House) of 200@1.10 .

A fourth allocation is done involving Member F and Member A (House) of 460@1.00 .

Initiator			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T1	Member A **(Client)**	340	1.00	1.00	~~1000~~ 340	Member A **(House)**	T1
Reactors			**Bid**	**Ask**			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T4	Member D	~~200~~ 0	1.10	1.00	490	Member E	T5

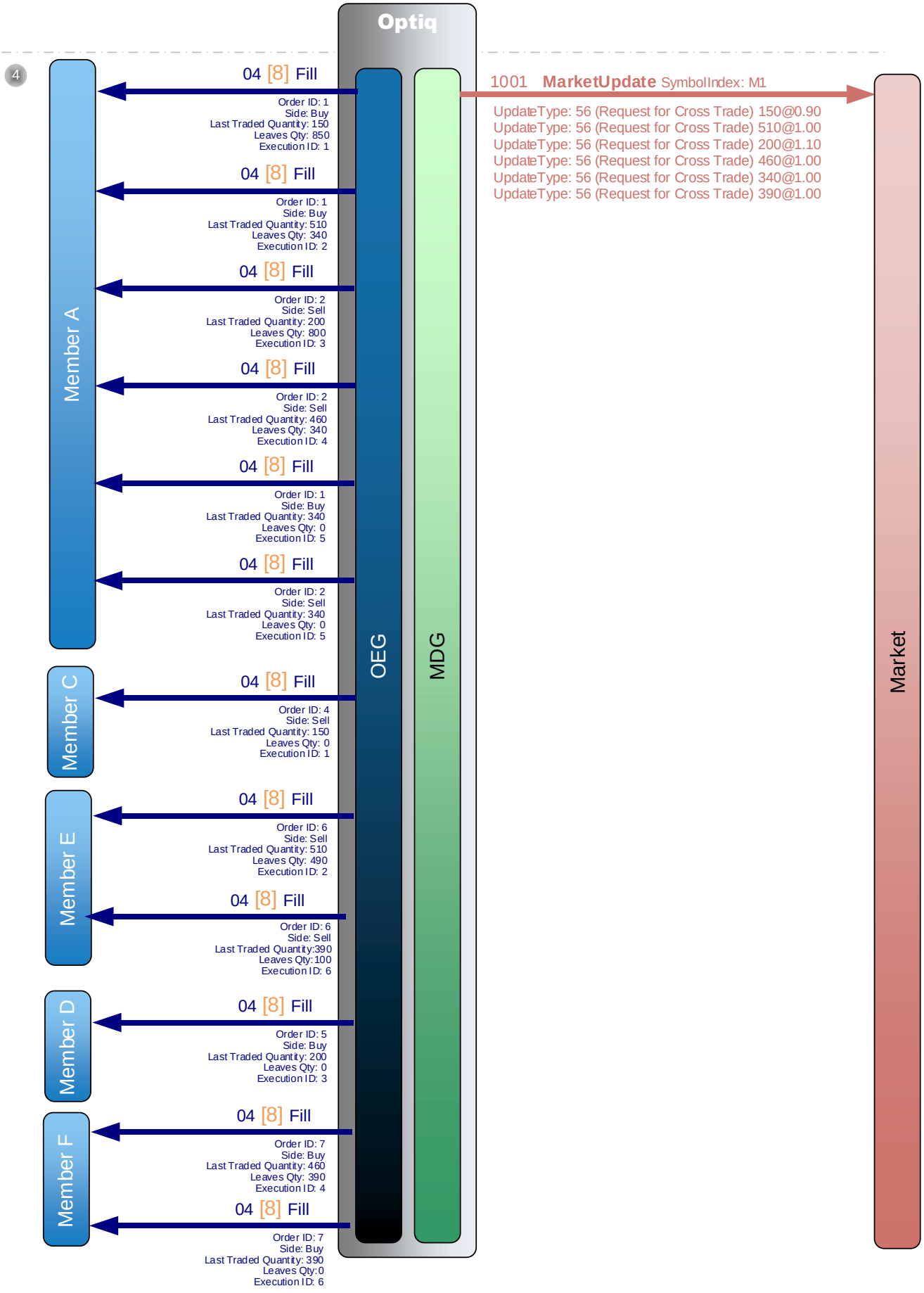
T6	Member F	~~850~~ 390	1.00				
T2	Member B	100	0.90				

Step 4: Final Cross Execution

Remaining RFC Initiator Client quantity is matched as a cross trade against the RFC Initiator Non-client quantity, resulting in the fifth allocation involving Member A Client and House sides for 340@1.00 to finalize matching of the initiator volume.

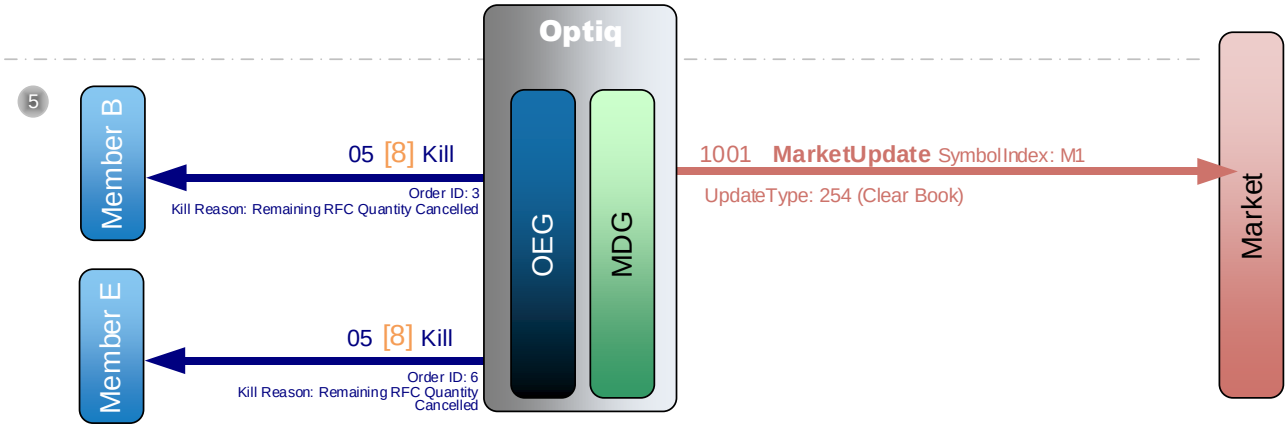
Step 5: Uncrossing of the RFC Reactor book

Remaining RFC response quantity with prices crossed is matched amongst themselves according to the Price Explicit Time matching policy. This results in the sixth allocation is done involving Member F and Member E of 390@1.00 to complete matching.



④ Following matching step done according to the **Client Best Execution RFC allocation (*)** the associated private and public messages are sent.
Members receive **Fill** (04) (FIX 8) messages for each individual part matched during the execution of their individual orders.
Public **MarketUpdate** (1001) messages for trades are sent out to the market in the order in which they were matched.

Note: For readability purposes, **FullTradeInformation** (1004) messages and **Statistics** (1009) messages are not in the diagrams.

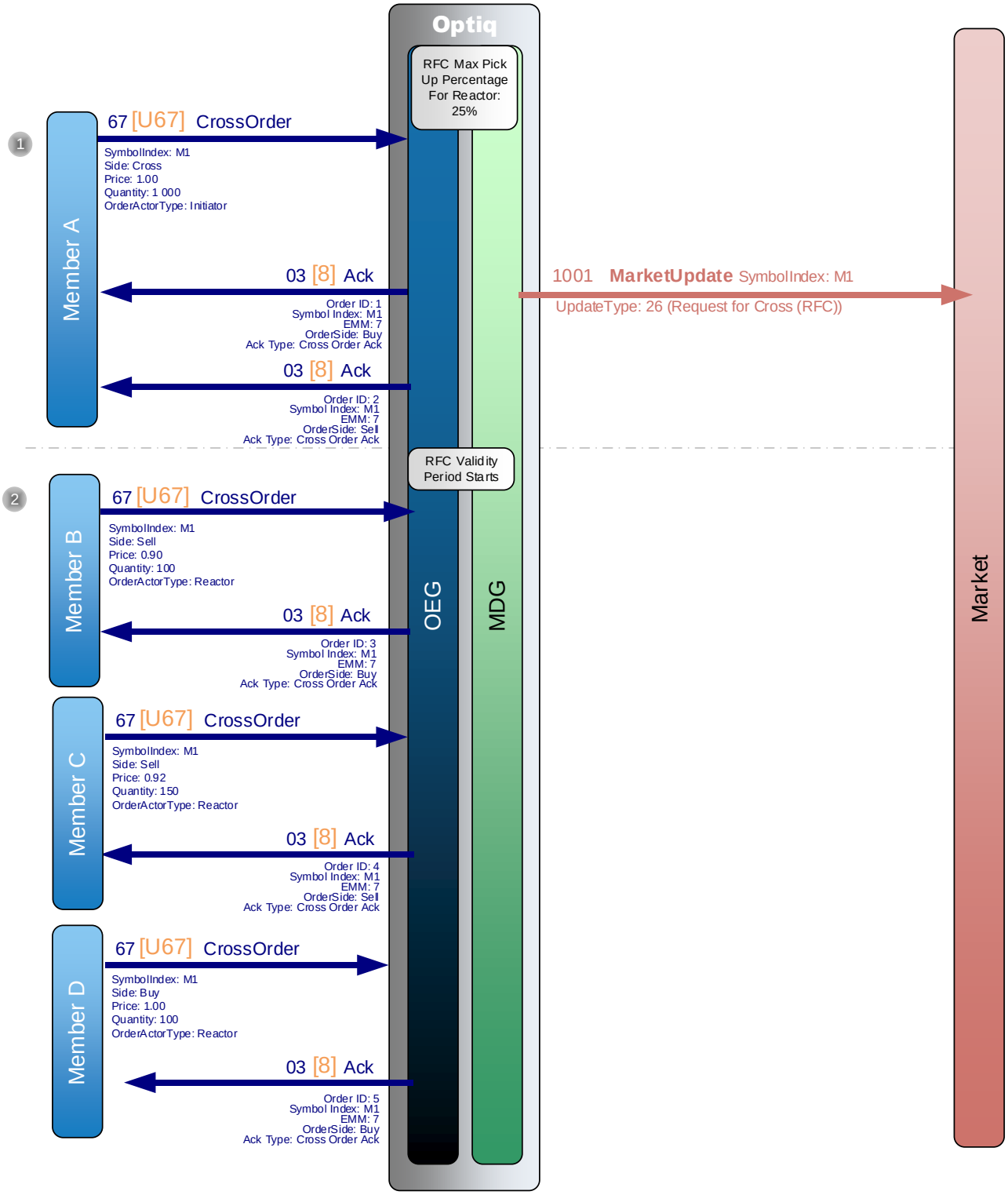


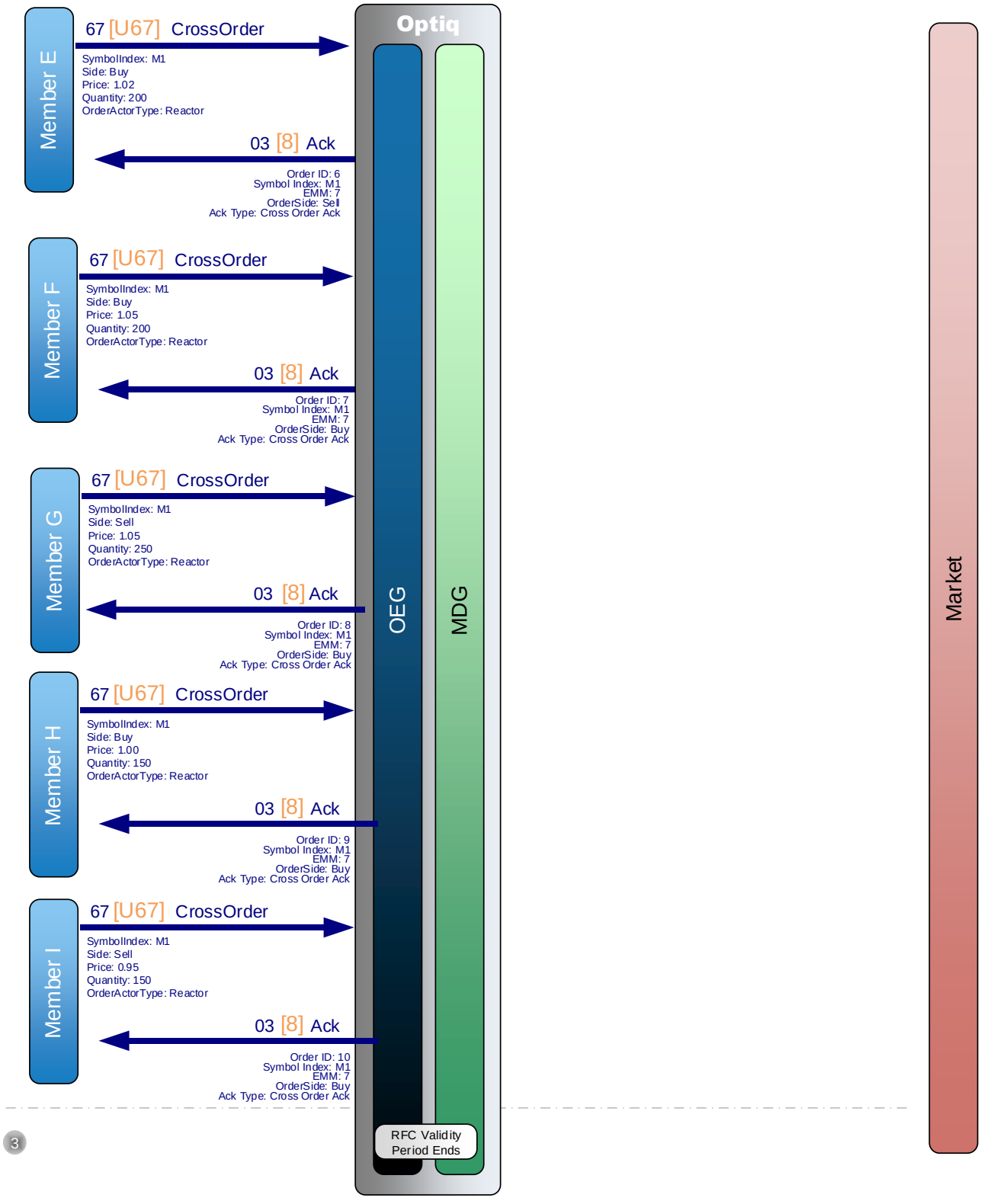
⑤ Cancellation of remaining orders:
As the last step, OEG sends to the Member B and Member E a private **Kill** (05) (FIX 8) message to cancel the remaining quantity of their reactor orders.
A public **MarketUpdate** (1001) message is sent to the market to clear the RFC book.

Note: This RFC is processed without any interaction with COB orders.

9.2 STANDARD RFC

For readability the scenario is presented in multiple diagrams.





Assumptions for this kinematics scenario:

- RFC Algorithm (parameter in standing data file): Standard RFC
- RFC Publication (parameter in standing data file): No
- RFC Max Pick Up Percentage For Reactor (parameter in standing data file): 25%
- First side provided in a Cross order always represents the Buy side, followed by the information for the Sell side

① Member A sends a private **CrossOrder** (67) (FIX U67) message to initialize a new RFC transaction on the instrument M1.

OEG sends back two private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of both order.

A public **MarketUpdate** (1001) message is sent to the market with *Update Type* set to 26 = Request for Cross (RFC) without price and quantity (RFC Publication parameter is set to No) which informs the market that there is a new RFC request that is available for improvement and matching.

② The RFC Validity Period starts.

Member B sends a private **CrossOrder** (67) (FIX U67) message to enter a new Sell order with a quantity of 100 and a price of 0.90.

Member C sends a private **CrossOrder** (67) (FIX U67) message to enter a new Sell order with a quantity of 150 and a price of 0.92.

Member D sends a private **CrossOrder** (67) (FIX U67) message to enter a new Buy order with a quantity of 100 and a price of 1.00.

Member E sends a private **CrossOrder** (67) (FIX U67) message to enter a new Buy order with a quantity of 200 and a price of 1.02.

Member F sends a private **CrossOrder** (67) (FIX U67) message to enter a new Buy order with a quantity of 200 and a price of 1.05.

Member G sends a private **CrossOrder** (67) (FIX U67) message to enter a new Sell order with a quantity of 250 and a price of 1.05.

Member H sends a private **CrossOrder** (67) (FIX U67) message to enter a new Buy order with a quantity of 150 and a price of 1.10.

Member I sends a private **CrossOrder** (67) (FIX U67) message to enter a new Sell order with a quantity of 150 and a price of 0.95.

Nothing is published for these reactor submissions to the market via public market data messages until RFC Validity Period ends.

③ The RFC Validity Period ends.

As the parameter RFC Publication is set to No, no public **MarketUpdate** (1001) messages are sent to the market for limits.

System starts matching of all the reactor orders to the initiated RFC according to the steps of (*) Standard RFC allocation.

To illustrate the complex processing of the RFC algorithm of this case, and for readability purposes tables below provide details of how the reactor submissions are allocated against the initiator. This processing is completed instantaneously at the end of the RFC validity period.

The RFC book starts with the following state:

Initiator	Bid	Ask

Timestamp	Firm	Quantity	Price	Price	Quantity	Firm	Timestamp
T1	Member A	1 000	1.00	1.00	1 000	Member A	T1
Reactors			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T8	Member H	150	1.10	0.90	100	Member B	T2
T6	Member F	200	1.05	0.92	150	Member C	T3
T5	Member E	200	1.02	0.95	150	Member I	T9
T4	Member D	100	1.00	1.05	250	Member G	T7

Step 1: RFC Initiator improvement

A first allocation is done involving Member B and Member A of 100@0.90 as it improves price of the buy side of the initiator RFC.

A second allocation is done involving Member C and Member A of 150@0.92 as it improves price of the buy side of the initiator RFC.

A third allocation is done involving Member I and Member A of 150@0.95 as it improves price of the buy side of the initiator RFC.

A fourth allocation is done involving Member H and Member A of 150@1.10 as it improves price of the sell side of the initiator RFC.

A fifth allocation is done involving Member F and Member A of 200@1.05 as it improves price of the sell side of the initiator RFC.

A sixth allocation is done involving Member E and Member A of 200@1.02 as it improves price of the sell side of the initiator RFC.

Initiator			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T1	Member A	~~1000~~ 600	1.00	1.00	~~1000~~ 450	Member A	T1
Reactors			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T8	Member H	~~150~~ 0	1.10	0.90	~~100~~ 0	Member B	T2
T6	Member F	~~200~~ 0	1.05	0.92	~~150~~ 0	Member C	T3
T5	Member E	~~200~~ 0	1.02	0.95	~~150~~ 0	Member I	T9
T4	Member D	100	1.00	1.05	250	Member G	T7

Step 2: Execution of reactors at RFC Price with RFC Initiator

This step takes into consideration the pick-up quantity for orders at the same price as the initiator. The resulting quantity is = 450 * 25% = 112.5

As such the seventh allocation is done involving Member D and Member A of 100@1.00.

Initiator			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T1	Member A	600	1.00	1.00	~~450~~ 350	Member A	T1
Reactors			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T4	Member D	~~100~~ 0	1.00	1.05	250	Member G	T7

Step 3: Final Cross Execution

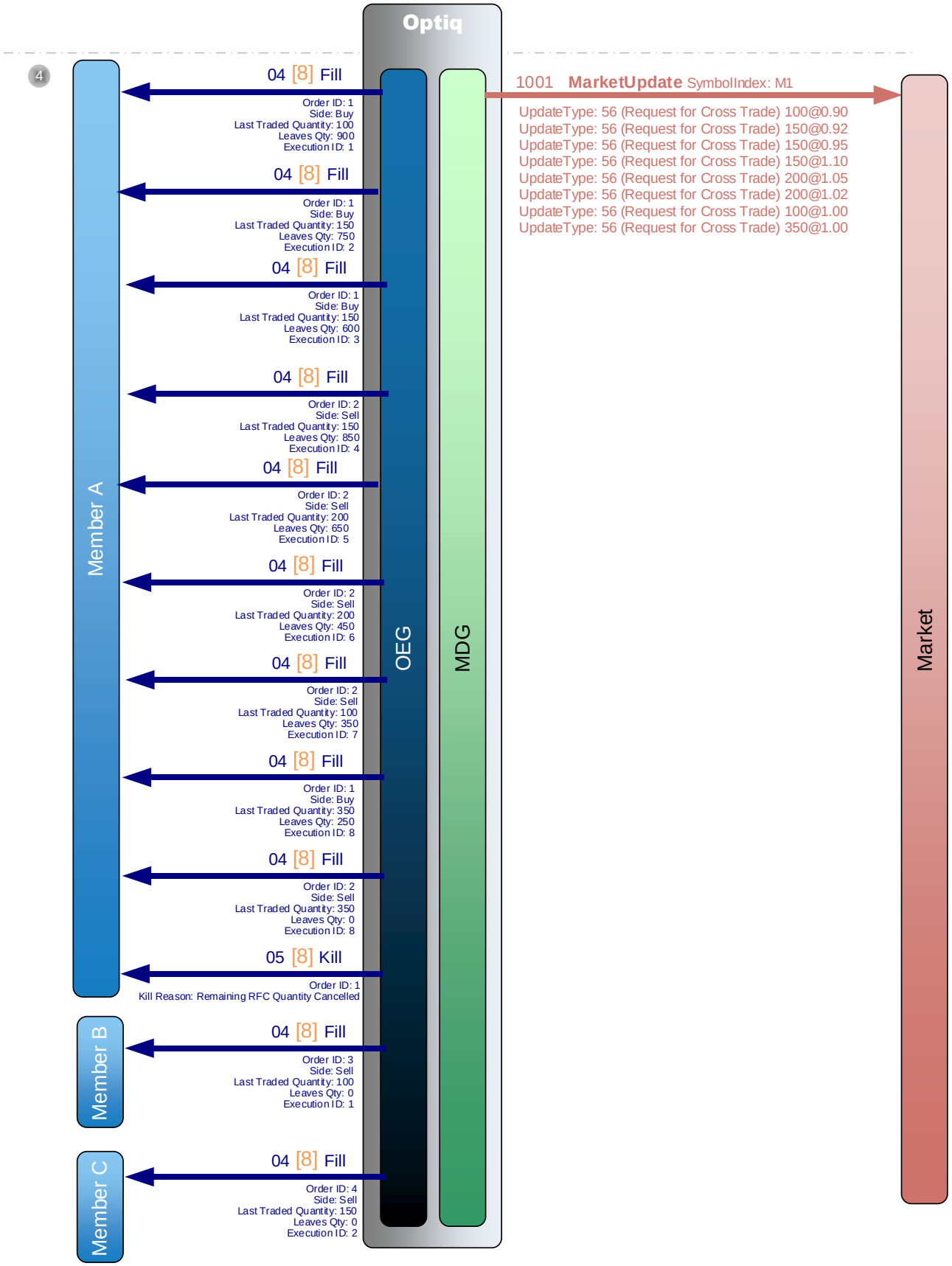
Remaining RFC Initiator quantity is matched as a cross trade for Member A, resulting in the eighth allocation of 350@1.00.

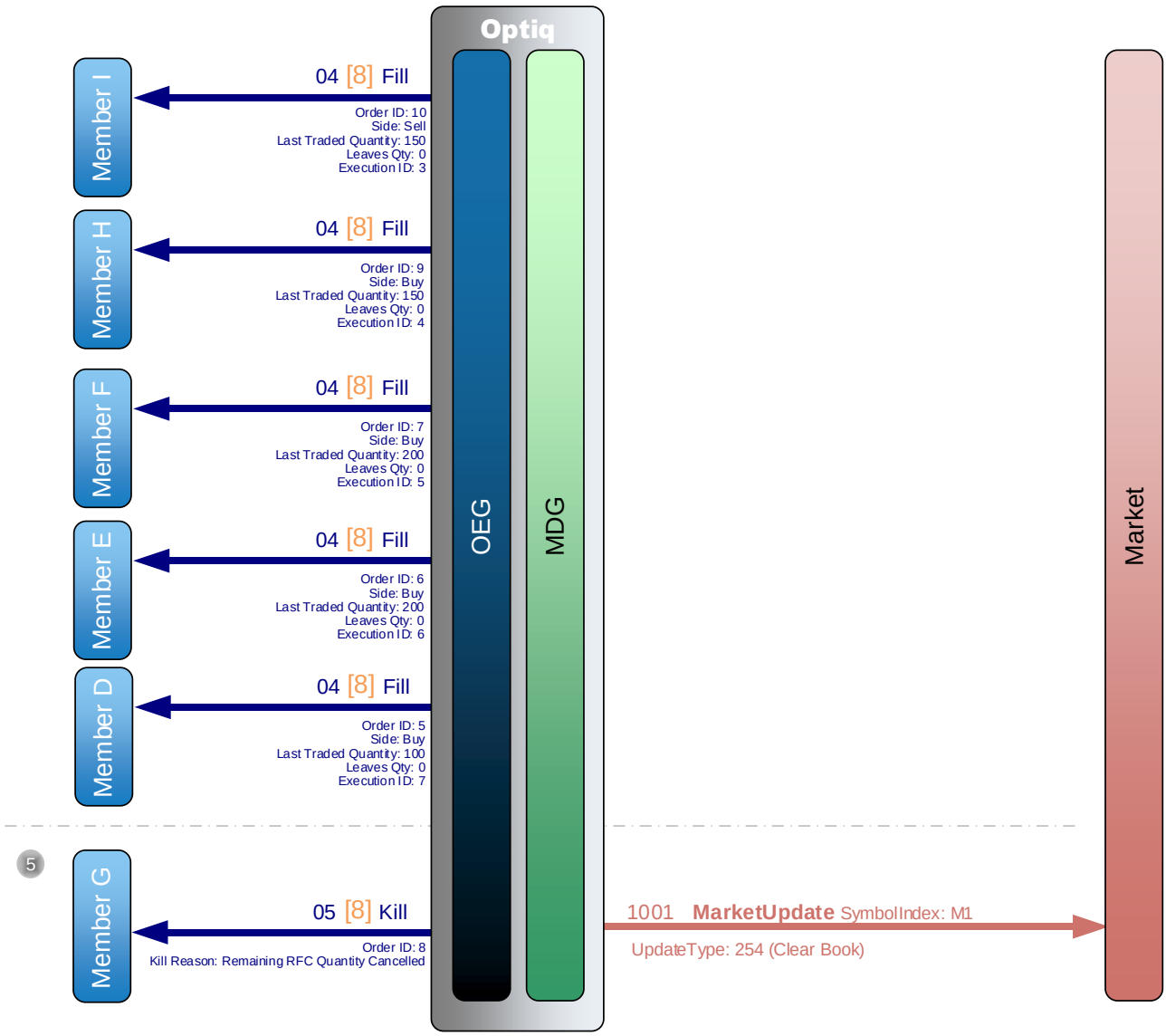
Remaining RFC initiator quantity of 250@1.00 is cancelled.

Initiator			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
T1	Member A	~~600~~ 250	1.00	1.00	~~350~~ 0	Member A	T1
Reactors			Bid	Ask			
Timestamp	**Firm**	**Quantity**	**Price**	**Price**	**Quantity**	**Firm**	**Timestamp**
				1.05	250	Member G	T7

Step 4: Uncrossing of the RFC Reactor book

This step is triggered only in the case where the RFC Reactor book is crossed. In this example, the RFC book is uncrossed.





④ Following matching step done according to the **Standard RFC allocation (*)** the associated private and public messages are sent.

Members receive **Fill** (04) (FIX 8) messages for each individual part matched during the execution of their individual orders.

Public **MarketUpdate** (1001) messages for trades are sent out to the market in the order in which they were matched.

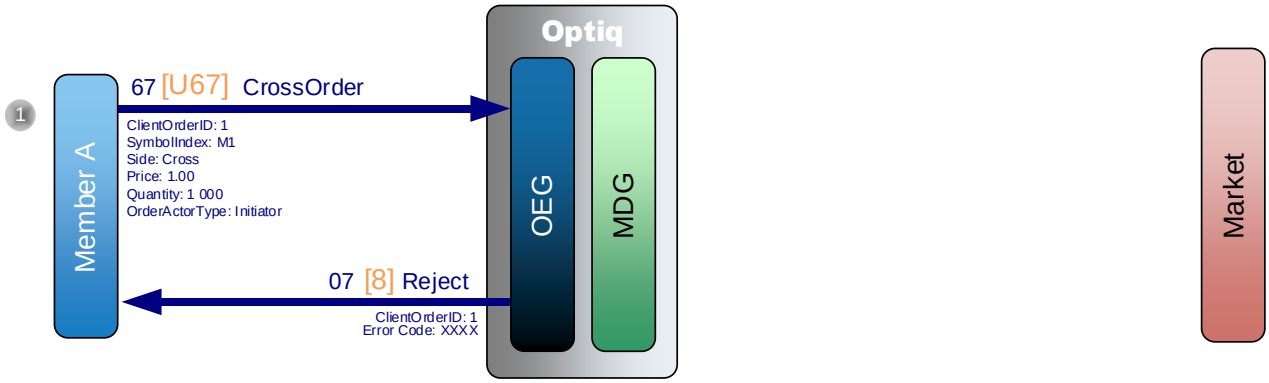
Note: For readability purposes, **FullTradeInformation** (1004) messages and **Statistics** (1009) messages are not in the diagrams.

⑤ As the last step, OEG sends to the Member G a private **Kill** (05) (FIX 8) message to cancel the remaining quantity of their reactor orders.

A public **MarketUpdate** (1001) message is sent to the market to clear the RFC book.

Note: This RFC is processed without any interaction with COB orders.

9.3 RFC REJECTED

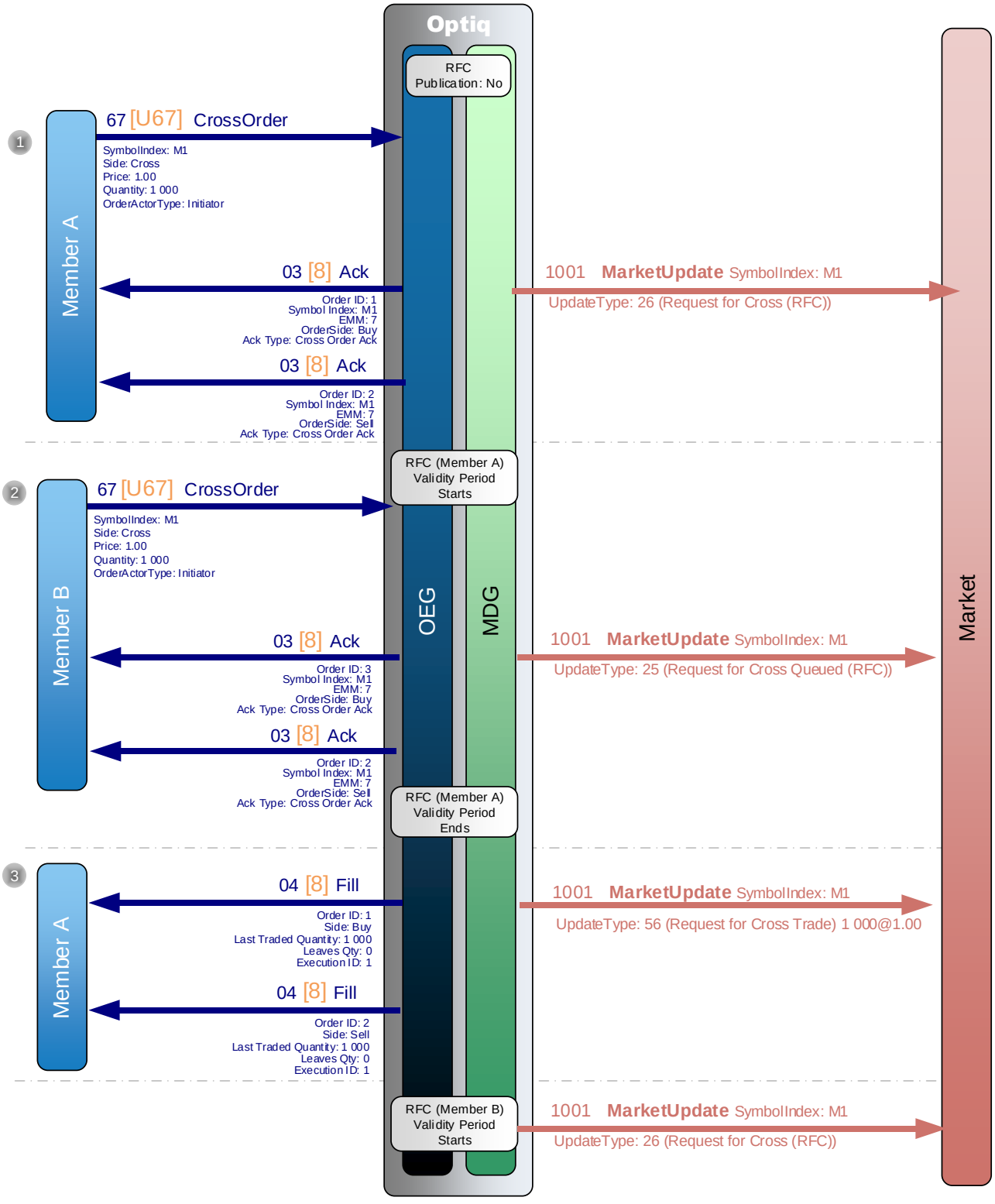


① Member A sends a private **CrossOrder** (67) (FIX U67) message to enter a new Request for Cross.

If the message is rejected OEG sends back a private **Reject** (07) (FIX 8) message with an *Error Code*. The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*.

No message is sent to the Market.

9.4 RFC EXPIRED AND RFC QUEUED



Assumptions for this kinematics scenario:

- RFC Algorithm (parameter in standing data file): Standard RFC
- RFC Publication (parameter in standing data file): No
- RFC Max Pick Up Percentage For Reactor (parameter in standing data file): 25%
- First side provided in a Cross order always represents the Buy side, followed by the information for the Sell side

① Member A sends a private **CrossOrder** (67) (FIX U67) message to initialize a new RFC transaction on the instrument M1.

OEG sends back two private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of both order.

A public **MarketUpdate** (1001) message is sent to the market with *Update Type* set to 26 = Request for Cross (RFC) without price and quantity which informs the market that there is a new RFC request that is available for improvement and matching.

② The RFC Validity Period starts for RFC from Member A.

Member B sends a private **CrossOrder** (67) (FIX U67) message to initialize a new RFC transaction on the instrument M1.

OEG sends back two private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of both order.

As the RFC from Member A is activated, this new RFC from Member B is queued.

A public **MarketUpdate** (1001) message is sent to the market with *Update Type* set to 25 = Request for Cross Queued (RFC) without price and quantity which informs the market that there is a RFC request queued.

The RFC Validity Period end for RFC from Member A and no reactors have reacted.

③ System matches the cross order from Member A.
Member A receives **Fill** (04) (FIX 8) messages for each side of his cross order.
A public **MarketUpdate** (1001) message for the trade is sent out to the market.

④ The RFC sent by Member B is activated.

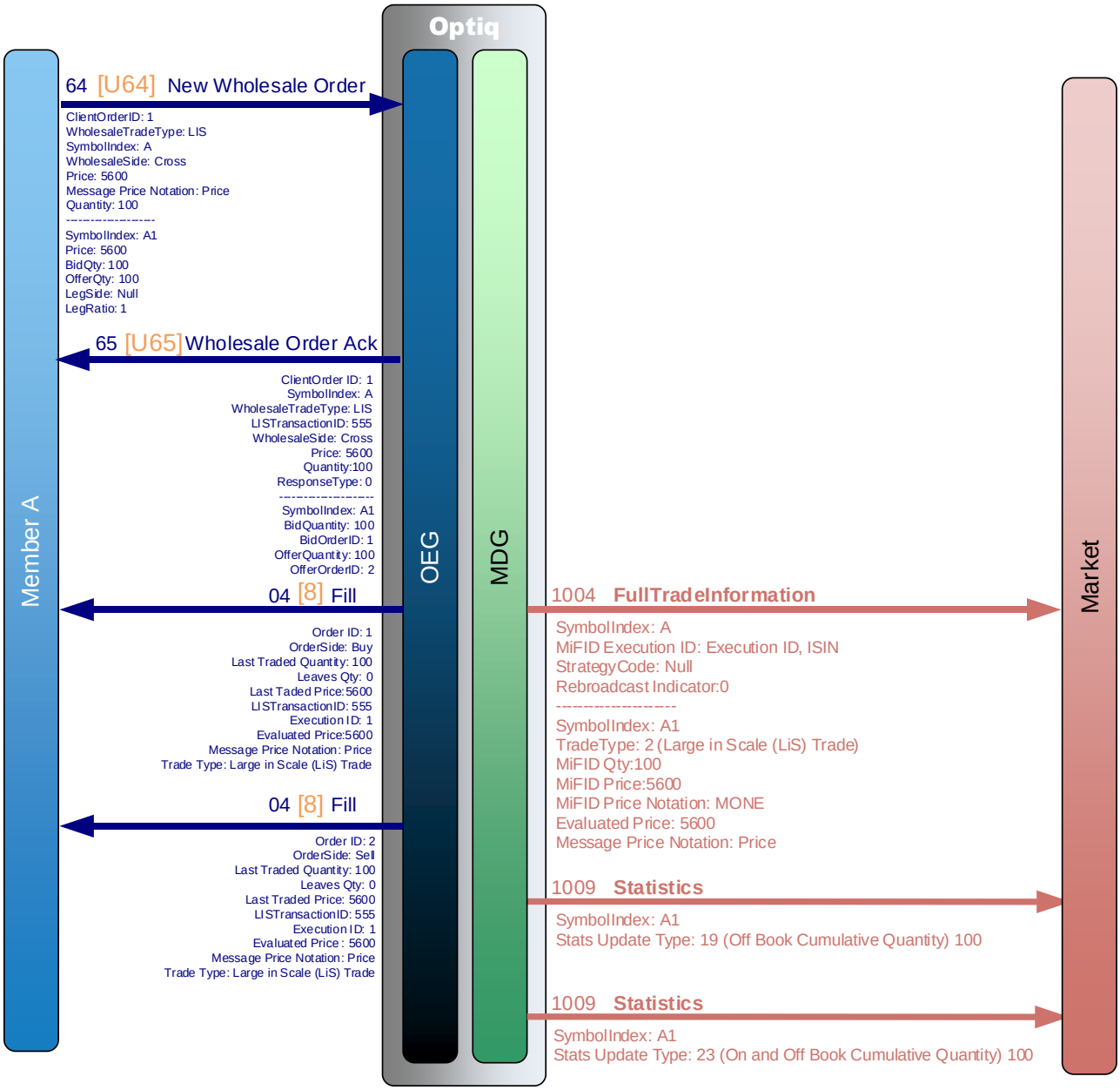
A public **MarketUpdate** (1001) message is sent to the market with *Update Type* set to 26 = Request for Cross (RFC) without price and quantity which informs the market that there is a new RFC request that is available for improvement and matching.

Note: As no reactors were involved in the RFC, no **MarketUpdate** (1001) was sent to the market to clear the book.

10. TOTAL RETURN FUTURE (TRF) AND MARKET ON CLOSE (MOC)

10.1 TRF WHOLESALE TRANSACTION – TAM TRADING

A1					
		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
T1	100	5600	5600	100	T2



Member A sends a private **NewWholesaleOrder** (64) (FIX U64) message to initialize a new Wholesale transaction on the instrument A1, an outright, part of a Total Return Future (TRF). The message includes both sides of the transaction, as a Cross order, and is submitted with the field *Message Price Notation* set to "**Price**". For a Wholesale transaction in a TRF contract this means trading type is Trade At Market (TAM).

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Wholesale Order, with the field *Response Type* set to 0 = Accept, and provides the system generated *LISTransactionID*.

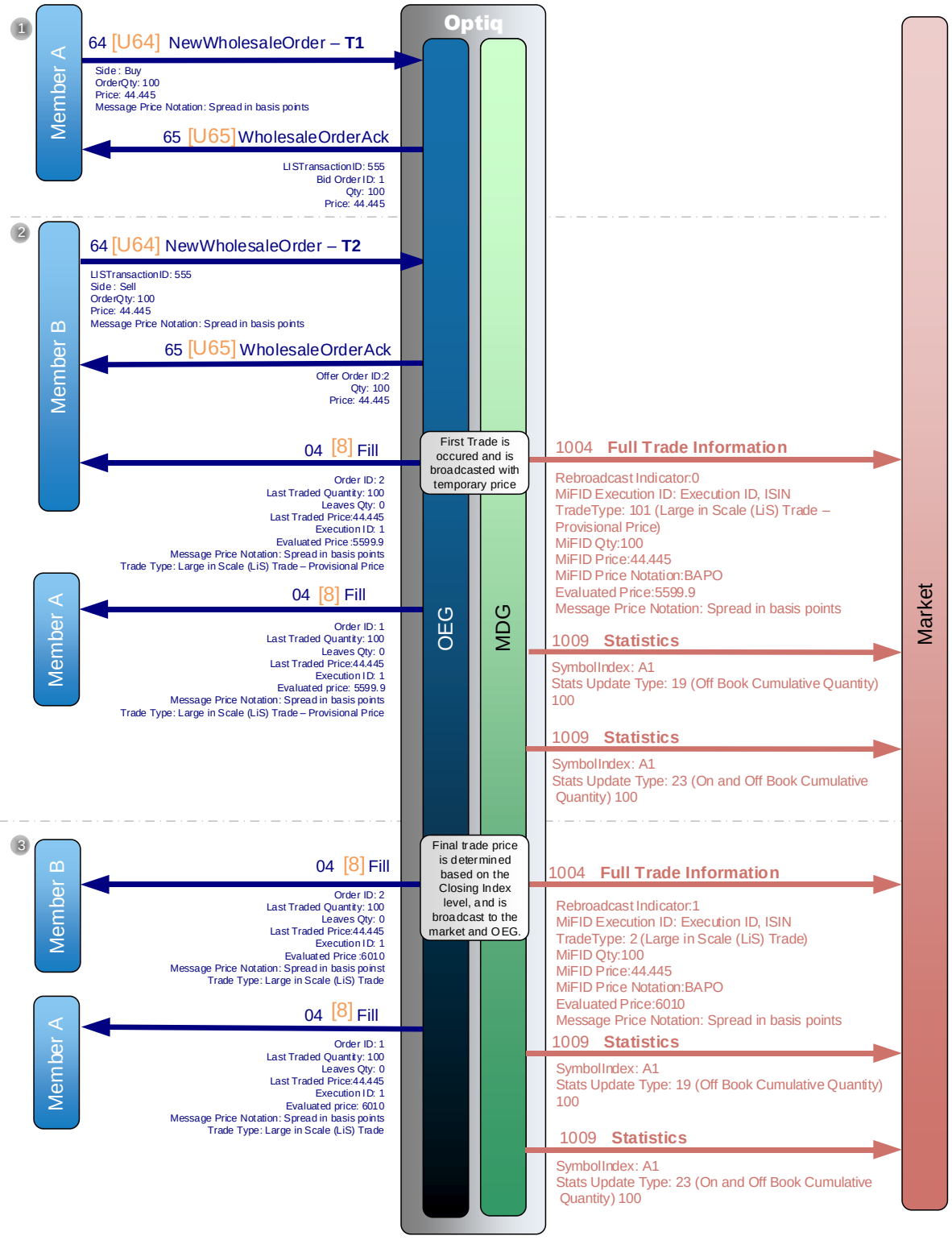
The transaction results in immediate match and the OEG generates a private **Fill** (04) (FIX 8) message for each leg of the trade. For each trade message the fields are set as following: *Evaluated Price* is equal to the entered price and *Trade Type* is set to "Large in Scale (LiS) Trade". As this transaction is managed as TAM - the price is "Final Confirmed Price".

A public **FullTradeInformation** (1004) message is sent to the market for the transaction with the field *TradeType* set to "2" = Large in Scale (LiS) Trade, and the *Rebroadcast indicator* set to 0.

This is followed by two **Statistics** (1009) messages sent to update the statistics of the Cumulative Quantity.

10.2 TRF WHOLESALE TRANSACTION – TAIC TRADING

A1					
		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
T1	100	44.445	44.445	100	T2



① Member A sends a private **NewWholesaleOrder** (64) (FIX U64) message to initialize a new Wholesale transaction on the instrument A1, an outright, part of a Total Return Future (TRF). The message is

provided with the Buy side of the transaction, and is submitted with the field *Message Price Notation* set to "Spread in Basis Points". For a Wholesale transaction on TRF contract this means trading type is Trade At Index Close (TAIC).

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Wholesale Order for the Buy side with system generated *LISTransactionID* for this order.

② Member B submits a private **NewWholesaleOrder** (64) (FIX U64) message to respond to the initial submission with the Sell side of the transaction. The submission contains the *LISTransactionID* that was provided by the system to Member A, which allows to map the two sides of the transaction.

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Sell side order.

Upon acceptance of the second order the quantity is fully met, the transaction results in an immediate match and the OEG generates a private **Fill** (04) (FIX 8) message for each leg of the trade to each Member that participated in the transaction where the *Trade Type* is equal to "Large in Scale (LiS) Trade – Provisional Price".

The price provided in the entering order messages is provided in the private messages in the field *Last Traded Price*. For TRF (and MOC) contracts the price provided in the field *Evaluated Price* is always expressed in Price index points notation and its calculation differs from the entered price. In this case the Evaluated Price is considered as Provisional (Temporary) Price.

A public **FullTradeInformation** (1004) message is immediately sent to the market for the provisional trade. The message is populated with the following values in fields: *TradeType* set to "101" = Large in Scale (LiS) Trade – Provisional Price, *Evaluated Price* set to the calculated price, based on the index levels identified at start of day and the field *Rebroadcast indicator* set to 0.

This is followed by two **Statistics** (1009) messages sent to update the statistics of the Cumulative Quantity.

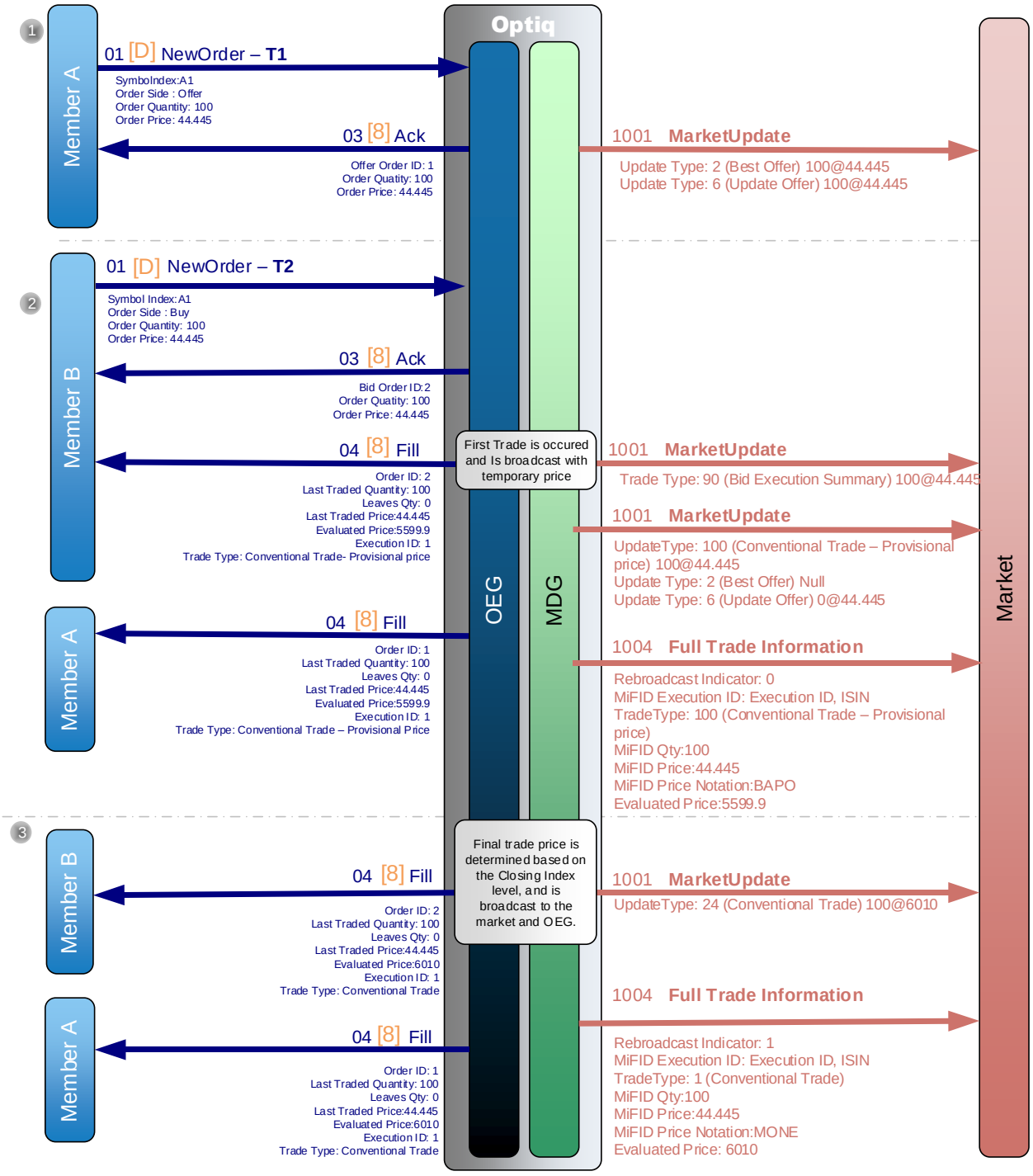
③ Upon receipt of the Closing level of the Index Optiq recalculates the trades that were executed during the day with the provisional prices.

Upon completion of this calculation OEG sends a new **Fill** (04) (FIX 8) messages, for each leg of the trade, to each participant of the transaction with the recalculated Final price. The OEG messages contain the same *Execution ID* as well as instrument information, as the ones provided for the earlier sent Provisional trade.

A public **FullTradeInformation** (1004) message is immediately sent to the market for the Final trade. The message is populated with the following values in fields: *TradeType* set to "2" = Large in Scale (LiS) Trade, *Evaluated Price* set to the calculated price, based on the index levels identified at end of day and the field *Rebroadcast indicator* set to 1.

10.3 TRF CENTRAL ORDER BOOK – TAIC TRADING

A1					
		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
T2	100	44.445	44.445	100	T1



① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 100 and a price of 44.445 (Expressed in Spread basis points).

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to update the Limit.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 100 and a price of 44.445 (Expressed in spread basis points).

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately, and fully, matches and the OEG generates a private **Fill** (04) (FIX 8) message to each member involved in the trade, for each leg of the trade. All the Fill messages are sent simultaneously with the *Evaluated Price* calculated in "Price index points" and a *Trade Type* of "Conventional Trade – Provisional Price".

A public **MarketUpdate** (1001) message is immediately sent to the market for the Execution Summary.

Only then, public **MarketUpdate** (1001) messages are sent to the market for the Trades and update of the Limits with an *Update Type* equal to "Conventional Trade – Provisional Price".

Then, a public **FullTradeInformation** (1004) message is sent to the market for the transaction with *Trade Type* of "Conventional Trade – Provisional Price" and the *Rebroadcast indicator* set to 0.

③ Upon receipt of the Closing level of the Index Optiq recalculates the trades that were executed during the day with the provisional prices.

Upon completion of this calculation OEG sends a new **Fill** (04) (FIX 8) messages, for each leg of the trade, to each participant of the transaction with the recalculated Final price. The OEG messages contain the same *Execution ID* as well as instrument information, as the ones provided for the earlier sent Provisional trade. The *Evaluated Price* should be recalculated.

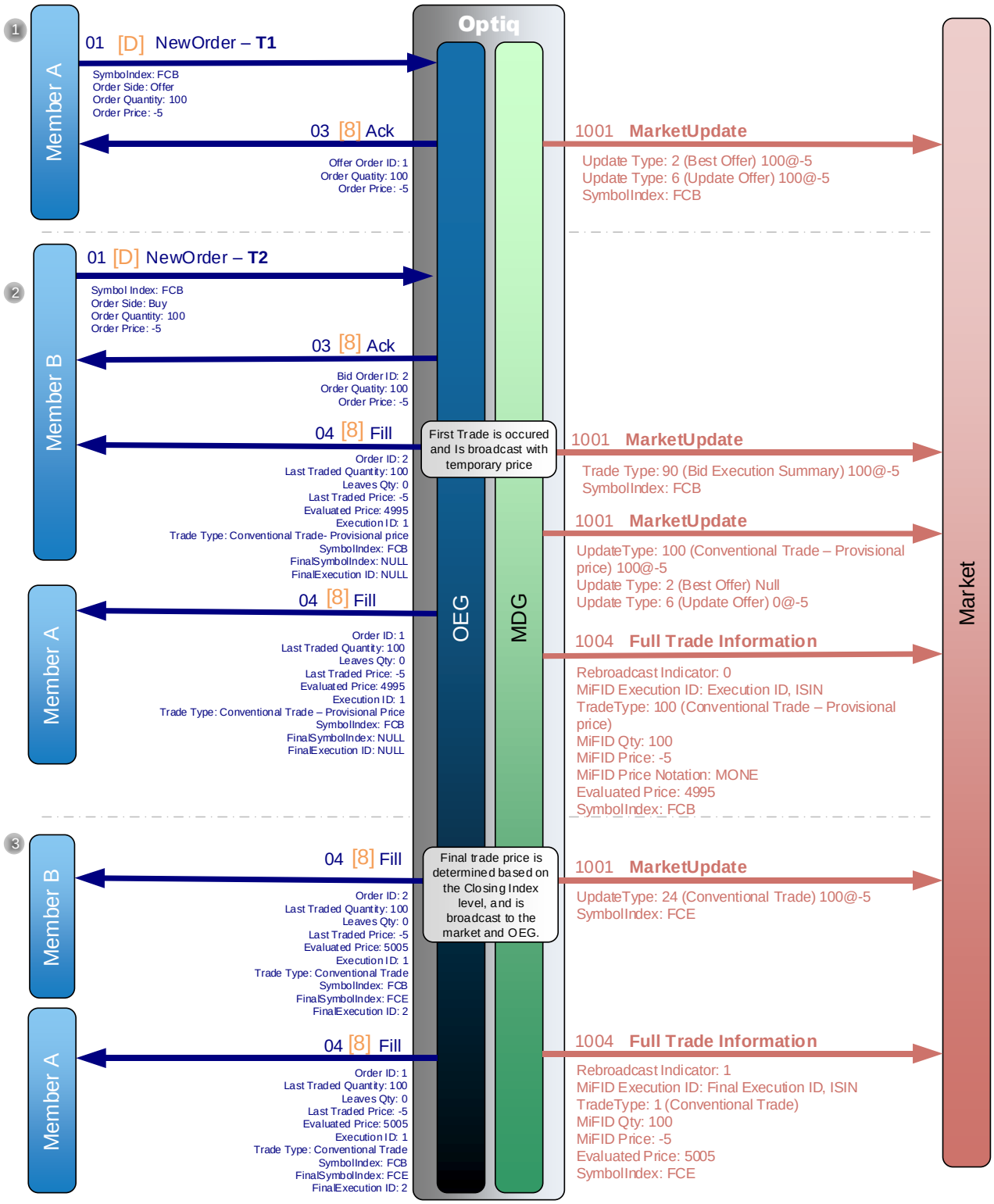
A public **MarketUpdate** (1001) message is immediately sent to the market for the occurred trade with *Update Type* of "Conventional Trade".

Then, a public **FullTradeInformation** (1004) message is submitted to the market for the Final trade. The message is populated with the following values in fields: *TradeType* set to "1" = Conventional Trade, *Evaluated Price* set to the calculated price, based on the index levels identified at end of day. The field *Rebroadcast indicator* should be set to 1.

10.4 MOC CENTRAL ORDER BOOK – TAIC TRADING

FCB

Bid			Offer		
Time	Qty	Price	Price	Qty	Time
T2	100	-5	-5	100	T1



Codes used in this example are: Market-on-Close Future Contract: FCB; and Final Future Contract: FCE

① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new sell order with a quantity of 100 and a price of -5 (Expressed in Spread basis points), with the symbol index of the Market-on-Close Future instrument (contract code FCB).

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 100 and a price of -5 (Expressed in Spread basis points), with the symbol index of the Market-on-Close Future instrument (contract code FCB).

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately, and fully, matches and the OEG generates a private **Fill** (04) (FIX 8) message to each member involved in the trade, for each leg of the trade. All the Fill messages are sent simultaneously with the *Evaluated Price* calculated in "price index points" and a *Trade Type* of "Conventional Trade – Provisional Price", with the symbol index of the Market-on-Close Future instrument (contract code FCB).

The *FinalSymbolIndex* and the *FinalExecutionID* in the fill are populated with null value.

A public **MarketUpdate** (1001) message is immediately sent to the market for the Execution Summary.

Following this, public **MarketUpdate** (1001) messages are sent to the market for the Trades and update of the Limits with an *Update Type* equal to "Conventional Trade – Provisional Price".

Then, a public **FullTradeInformation** (1004) message is sent to the market for the transaction with *Trade Type* of "Conventional Trade – Provisional Price" and the *Rebroadcast indicator* set to 0.

③ Upon receipt of the Closing level of the Index Optiq recalculates the trades that were executed during the day with the provisional prices.

Upon completion of this calculation OEG sends a new **Fill** (04) (FIX 8) messages, for each leg of the trade, to each participant of the transaction with the recalculated Final price. The OEG messages contain the same *Execution ID* as well as instrument information, as the ones provided for the earlier sent Provisional trade. The *Evaluated Price* should be recalculated.

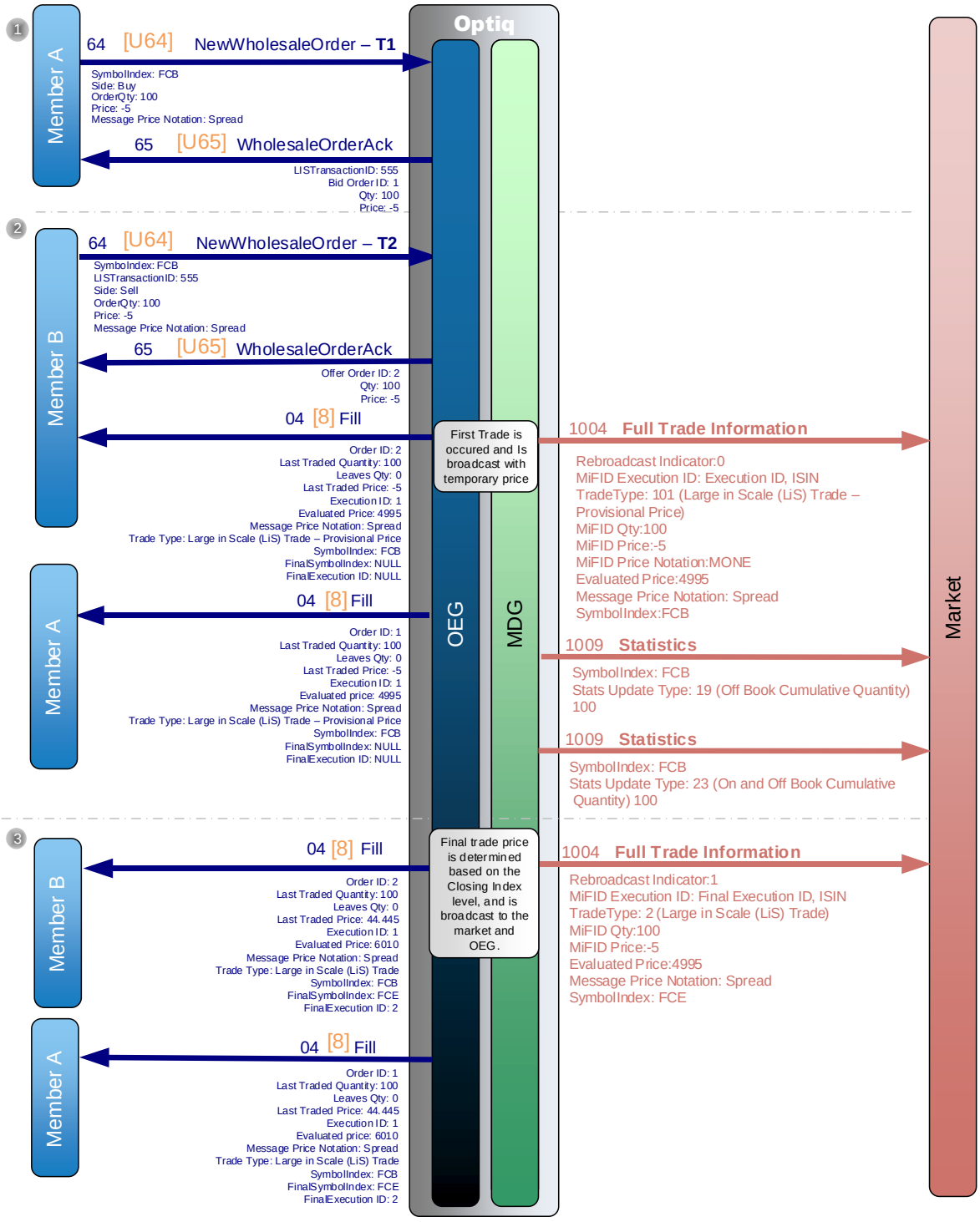
The *FinalSymbolIndex* and the *FinalExecutionID* are populated in the fill with the Final Futures instrument information (Contract Code FCE).

A public **MarketUpdate** (1001) message is immediately sent to the market for the trade with an *Update Type* of "Conventional Trade".

Then, a public **FullTradeInformation** (1004) message is immediately sent to the market for the Final trade. The message is populated with the following values in fields: *TradeType* set to "1" = Conventional Trade, *Evaluated Price* set to the calculated price, based on the index levels identified at end of day. The field *Rebroadcast indicator* should be set to 1, with the symbol index of the Final Future instrument.

10.5 MOC WHOLESALE TRANSACTION – TAIC TRADING

FCB					
		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
T1	100	-5	-5	100	T2



Codes used in this example are: Market-on-Close Future Contract: FCB; and Final Future Contract: FCE

① A Member A sends a private **NewWholesaleOrder** (64) (FIX U64) message to initialize a new Wholesale transaction on the instrument A1, providing bid side of the transaction with *Message Price Notation*

equal to "Spread" (Which means in this case Spread in index points), with the symbol index of the Market-on-Close Future instrument (contract code FCB).

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Wholesale Order for the Buy side with system generated *LISTransactionID* for this order.

② A Member B sends a private **NewWholesaleOrder** (64) (FIX U64) message to initialize submission with the Sell side of the transaction. The submission contains the *LISTransactionID* that was provided by the system to Member A, which allows to map the two sides of the transaction, with the symbol index of the Market-on-Close Future instrument (contract code FCB).

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Sell order.

Upon acceptance of the second order the quantity is fully met, the transaction results in immediate match and the OEG generates a private **Fill** (04) (FIX 8) message for each leg of the trade to each Member that participated in the transaction where the Trade Type is equal to "Large in Scale (Lis) Trade – Provisional Price", with the symbol index of the Market-on-Close Future instrument (contract code FCB).

The FinalSymbolIndex and the FinalExecutionID in the fill are populated with null value.

The price provided in the entering order messages is provided in the private messages in the field *Last Traded Price*. For MOC (and TRF) contracts the price provided in the field *Evaluated Price* is always expressed in Price index points notation and its calculation differs from the entered price. In this case the Evaluated Price is considered as Provisional (Temporary) Price.

A public **FullTradeInformation** (1004) message is immediately sent to the market for the provisional trade. The message is populated with the following values of fields: *TradeType* set to "101" = Large in Scale (LiS) Trade – Provisional Price, *Evaluated Price* set to the calculated price, based on the index levels identified at start of day and the field *Rebroadcast indicator* set to 0.

This is followed by two **Statistics** (1009) messages sent to update the statistics of the Cumulative Quantity.

③ Upon receipt of the Closing level of the Index Optiq recalculates the trades that were executed during the day with the provisional prices.

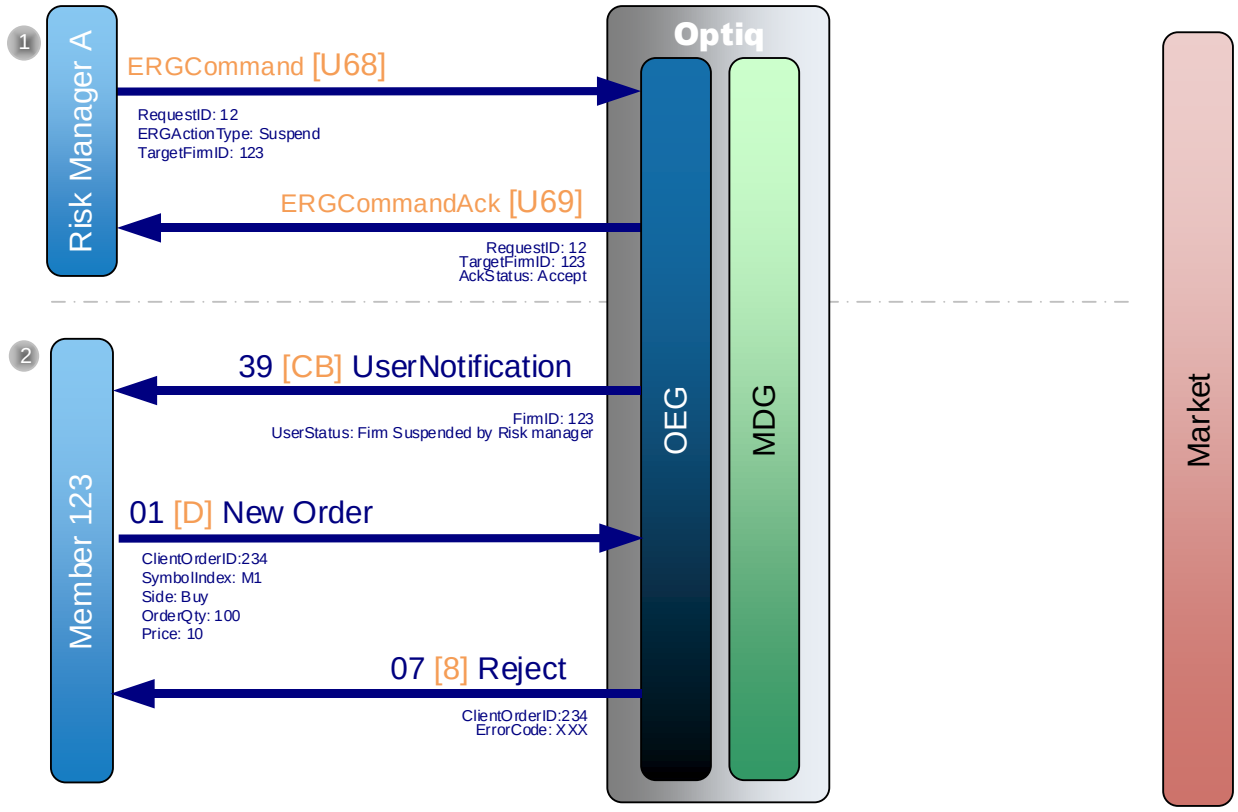
Upon completion of this calculation OEG sends a new **Fill** (04) (FIX 8) messages, for each leg of the trade, to each participant of the transaction with the recalculated Final price. The OEG messages contain the same *Execution ID* as well as instrument information, as the ones provided for the earlier sent Provisional trade. The *Evaluated Price* should be recalculated regarding closing prices.

The *FinalSymbolIndex* and the *FinalExecutionID* are populated in the fill with the Final Future instrument information (Contract Code FCE)

A public **FullTradeInformation** (1004) message is immediately sent to the market for the Final trade. The message is populated with the following values in fields: *TradeType* set to "2" = Large in Scale (LiS) Trade, *Evaluated Price* set to the calculated price, based on the index levels identified at end of day. The field *Rebroadcast indicator* should be set to 1, with the symbol index of the Final Future instrument.

11. EURONEXT RISKGUARD (ERG)

11.1 ERG: SUSPEND A FIRM WITHOUT CANCELLATION OF ORDERS



For this example: *Both the Risk Manager A and Member 123 are logged on to an OEG on the Equity Derivatives segment, and Risk Manager A is setup as the risk manager for this firm. ERG messages allow for other granularities for this functionality that are listed in the message specifications.*

① Risk Manager A sends an **ERGCommand** (U68) message to suspend Member 123 (identifier provided within *TargetFirmID*).

OEG sends back an **ERGCommandAck** (U69) message to confirm the successful receipt and technical processing of the message.

② Member 123 is notified of the suspension via **UserNotification** (39) (FIX CB) message. The **UserNotification** (39) (FIX CB) message is sent to all OE Sessions via which the Member 123 is connected, for the given Optiq Segment.

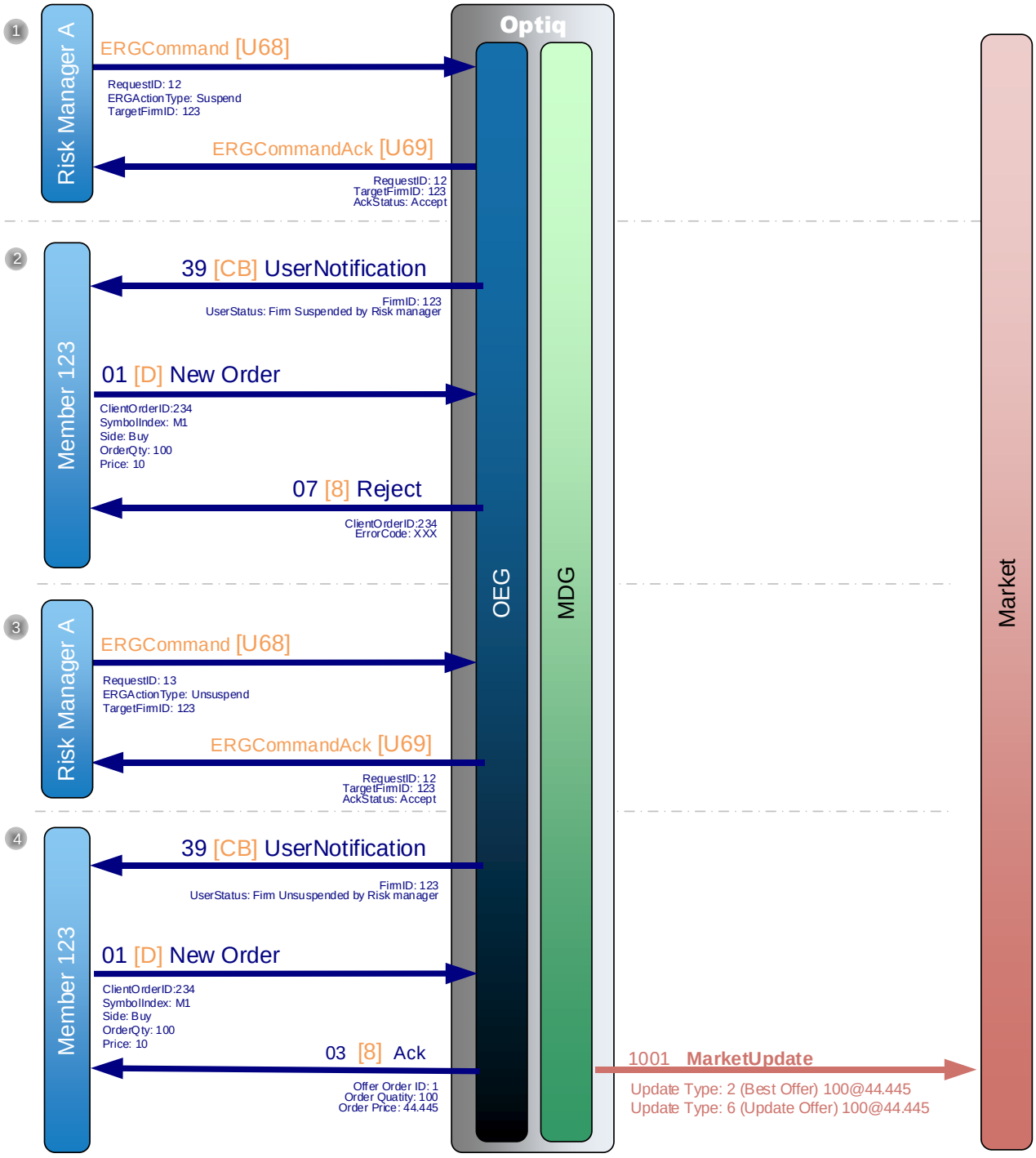
As the suspension was done without cancellation of orders, no other messages are sent to the Member at this moment. No messages are sent to the market.

Member 123 submits a **NewOrder** (01) (FIX D) messages. As member 123 is suspended OEG sends back a private **Reject** (07) (FIX 8) message to reject the creation of the new order with an Error Code.

The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*. No messages are sent to the market.

Note: Maximum scope of any RiskGuard messages is the Optiq segment. If an action from the Risk Manager requires to be effective on multiple segments, a message needs to be sent to each Optiq segment.

11.2 ERG: UNSUSPEND A FIRM



For this example: *Both the Risk Manager A and Member 123 are logged on an OEG on the Equity Derivatives segment, and Risk Manager A is setup as the risk manager for this firm, and has previously suspended Member 123. ERG messages allow for other granularities for this functionality that are listed in the message specifications.*

① Risk Manager A sends an **ERGCommand** (U68) message to suspend Member 123 (identifier provided within *TargetFirmID* field).

OEG sends back a private **ERGCommandAck** (U69) message to confirm the successful receipt and technical processing of the message.

② Member 123 is notified that their suspension has been lifted via **UserNotification** (39) (FIX CB) message. The **UserNotification** (39) (FIX CB) message is sent to all OE Sessions on which the Member 123 is connected, for the given Optiq Segment.

Member 123 sends a **NewOrder** (01) (FIX D) messages. As the status of the Member 123 is suspended, the message is rejected. OEG sends back a private **Reject** (07) (FIX 8) message to reject the creation of the new order with an Error Code.

The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*.

③ Risk Manager A sends an **ERGCommand** (U68) message to request the Unsuspension of Member 123 (identifier provided within *TargetFirmID* field).

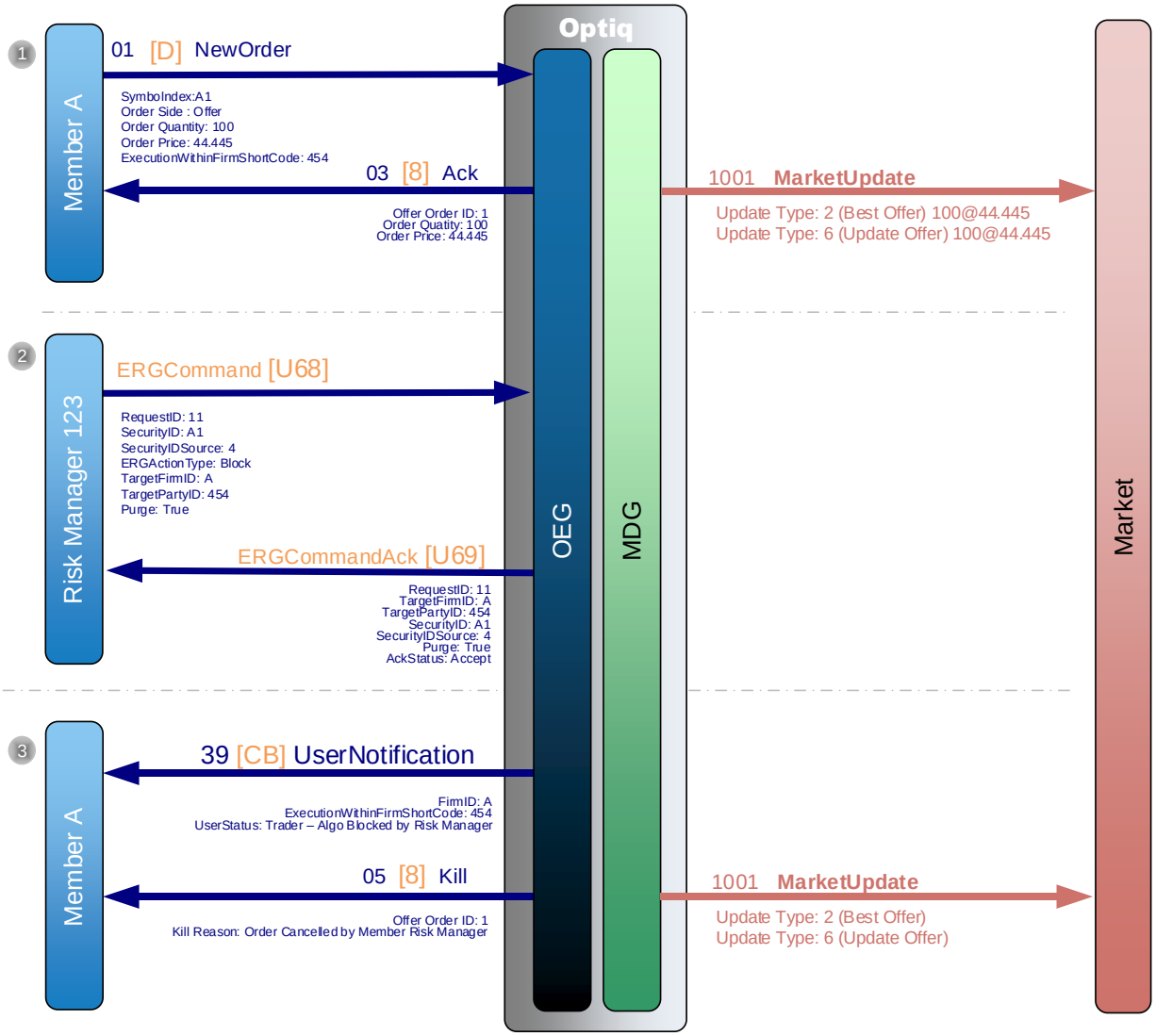
OEG sends back a private **ERGCommandAck** (U69) message to confirm the successful receipt and technical processing of the message.

④ Member 123 is notified of the unsuspension through **UserNotification** (39) (FIX CB) message. The **UserNotification** (39) (FIX CB) message is sent to all OE Sessions on which the Member 123 is connected, for the given Optiq Segment.

Upon submission of a **NewOrder** (01) (FIX D) message OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

Note: Maximum scope of any RiskGuard messages is the Optiq segment. If an action from the Risk Manager requires to be effective on multiple segments, a message needs to be sent to each Optiq segment.

11.3 ERG: BLOCK A TRADER OR AN ALGORITHM WITH ORDER CANCELLATION



For this example: *Both the Risk Manager 123 and Member A are logged on to an OEG on the Equity Derivatives segment, and Risk Manager 123 is setup as the risk manager for this firm. ERG messages allow for other granularities for this functionality that are listed in the message specifications.*

① Member A submits a **NewOrder** (01) (FIX D) message. OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limit.

② Risk Manager 123 sends an **ERGCommand** (U68) message to block a Trader (or an Algorithm) identified by the Short Code '454' (provided within *TargetPartyID* field) of the Member A (provided within *TargetFirmID* field), on a specific contract A1 (provided within *SecurityID* field), selecting to cancel all of the active orders in the book for this action.

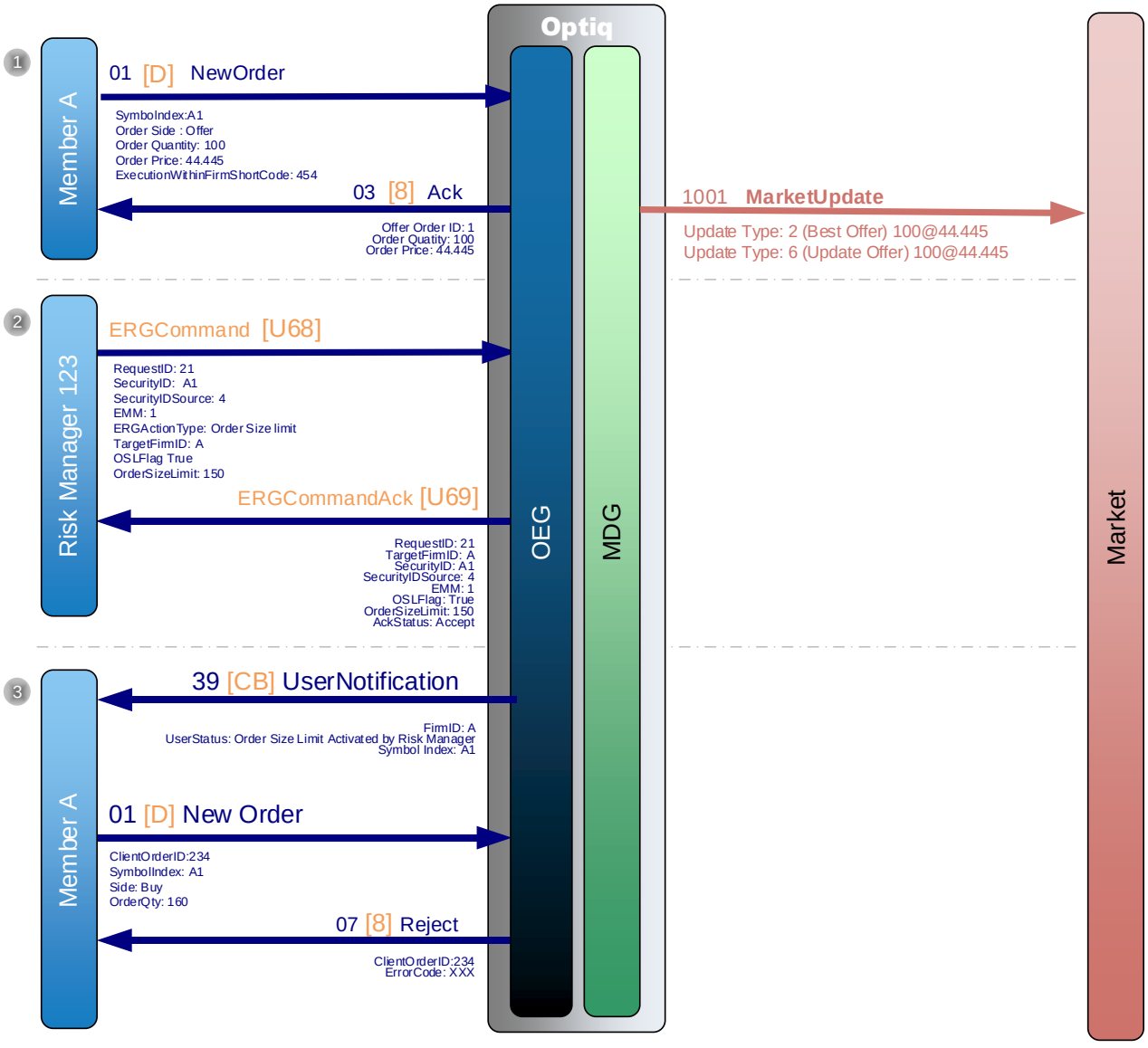
OEG sends back a private **ERGCommandAck** (U69) message to confirm the successful receipt and technical processing of the message.

③ OEG then sends a private **UserNotification** (39) (FIX CB) message to all the OE Sessions on which the Member A is connected, to notify the member that the identified trader (or an algorithm) is currently blocked for contract A1.

OEG sends back a private **Kill** (05) (FIX 8) message to notify the member of the cancellation of the active order in the book for the combination of the identified Target Firm ID + Target Party ID. Only the OE session that owns the order receives the message of the order cancellation.

A public **MarketUpdate** (1001) message is sent to the market to update the limits.

11.4 ERG: ORDER SIZE LIMIT ACTIVATED



Both the Risk Manager 123 and Member A are logged on to an OEG on the Equity Derivatives segment, and Risk Manager 123 is setup as the risk manager for this firm.

① Member A submits a **NewOrder** (01) (FIX D) message. OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limit.

② Risk Manager 123 sends an **ERGCommand** (U68) message to activate the Order Size Limit control for Member A, on contract A1, with the maximum size of 150.

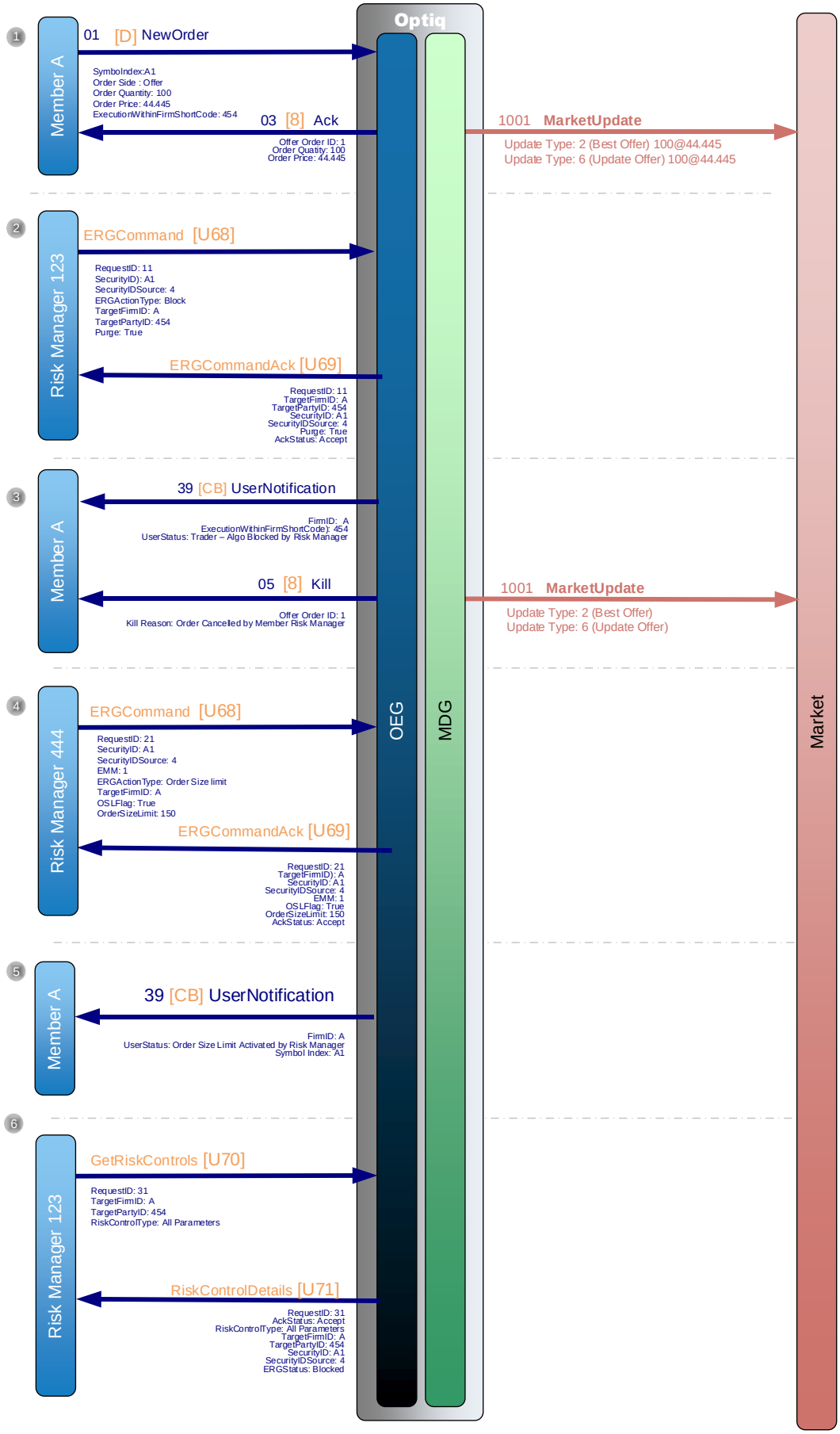
OEG sends back a private **ERGCommandAck** (U69) message to confirm the successful receipt and technical processing of the message.

③ Then OEG sends a private **UserNotification** (39) (FIX CB) message to all OE Sessions on which the Member A is connected, to notify the member that Order Size Limit is currently activated for a given contract, with the maximum size of 150.

Member A submits a **NewOrder** (01) (FIX D) message, with Order Qty of 160, which is higher than the set Order Size Limit. This order will be rejected as it breaches the set Order size limit, and OEG sends back a private **Reject** (07) (FIX 8) message to reject the creation of the new order with an Error Code.

The reason of the rejection can be found using the Error Code value within the *Euronext Markets - Optiq & TCS Error list file (.csv)*.

11.5 ERG: GET RISK CONTROL DETAILS FOR A TRADER OR AN ALGORITHM



Both the Risk Manager 123, Risk Manager 444 and Member A are logged on to an OEG on the Equity Derivatives segment, and Risk Managers 123 and 444 are setup as the risk managers for this firm.

① Member A submits a **NewOrder** (01) (FIX D), OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limit.

② Risk Manager 123 sends a **ERGCommand** (U68) message to block Trader (or an Algorithm) identified by the Short Code '454' (provided within *TargetPartyID* field) and Member A (provided within *TargetFirmID* field), on a specific contract A1 (provided within *SecurityID* field), selecting to cancel all of the active orders in the book for this action.

OEG sends back a private **ERGCommandAck** (U69) message to confirm the successful receipt and technical processing of the message.

③ OEG sends a private **UserNotification** (39) (FIX CB) message to all OE Sessions on which the Member A is connected, to notify the member that the identified trader (or an algorithm) is currently blocked for the contract A1.

OEG then sends a private **Kill** (05) (FIX 8) to notify the member of the cancellation of the only active order in the book for the combination of Target Firm ID + Target Party ID.

A public **MarketUpdate** (1001) message is sent to the market to update the limits.

④ Risk Manager 444 sends an **ERGCommand** (U68) message to activate the Order Size Limit control, for Member A, on the contract A1.

OEG sends back a private **ERGCommandAck** (U69) message to confirm the successful receipt and technical processing of the message.

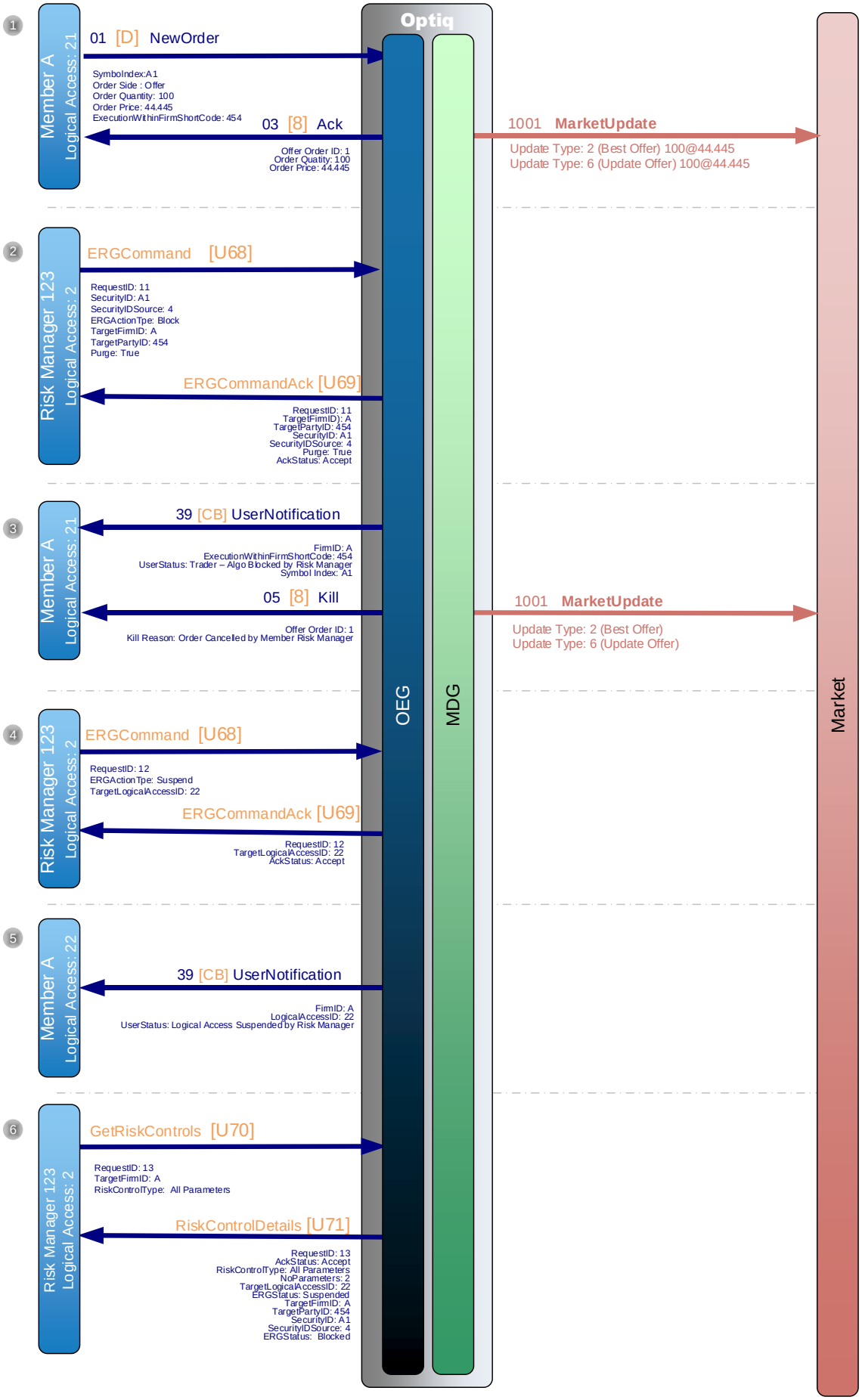
⑤ OEG sends a private **UserNotification** (39) (FIX CB) message to all OE Sessions on which the Member A is connected, to notify the member that Order Size Limit is currently activated for contract A1.

⑥ Risk Manager 123 sends a **GetRiskControls** (U70) message to request all parameters set by them, for Member A + Target Party ID 454.

OEG sends back a **RiskControlDetails** (U71) message to provide all the current Risk Guard controls set by Risk Manager 123 for Member A + Target Party ID 454.

Note: OEG sends to the Risk Manager 123 only the controls activated by that risk manager, meaning, in the **RiskControlDetails** (U71) there is no mention of the activation of Order Size Limit command sent by Risk Manager 444.

11.6 ERG: GET RISK CONTROL DETAILS - ALL PARAMETERS



Both Risk Manager 123 and Member A are logged on to an OEG on the Equity Derivatives segment, and Risk Manager 123 is setup as the risk manager for this firm. Member A is connected through 2 different Logical Accesses to the Equities segment.

① Member A submits a **NewOrder** (01) (FIX D), OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to the market to update the limit.

② Risk Manager 123 sends an **ERGCommand** (U68) message to block Trader (or an Algorithm) identified by the Short Code '454' (provided within *TargetPartyID* field) and Member A (provided within *TargetFirmID* field), on a specific contract A1 (provided within *SecurityID* field), selecting to cancel all of the active orders in the book for this action.

OEG sends back a private **ERGCommandAck** (U69) message to confirm the successful receipt and technical processing of the message.

③ OEG sends a private **UserNotification** (39) (FIX CB) message to all OE Sessions on which the Member A is connected, to notify the member that the trader (or an algorithm) is currently blocked on the contract A1.

OEG sends back a private **Kill** (05) (FIX 8) to notify the member of the cancellation of the only active order in the book for the combination of Target Firm ID + Target Party ID.

A public **MarketUpdate** (1001) message is sent to the market to update the limits.

④ Risk Manager 123 sends an **ERGCommand** (U68) message to suspend for the Logical Access ID 22.

OEG sends back a private **ERGCommandAck** (U69) message to confirm the successful receipt and technical processing of the message.

⑤ OEG sends a private **UserNotification** (39) (FIX CB) message to all OE Sessions on which the Member A is connected, to notify the member that the Logical Access 22 has been suspended by the Risk Manager.

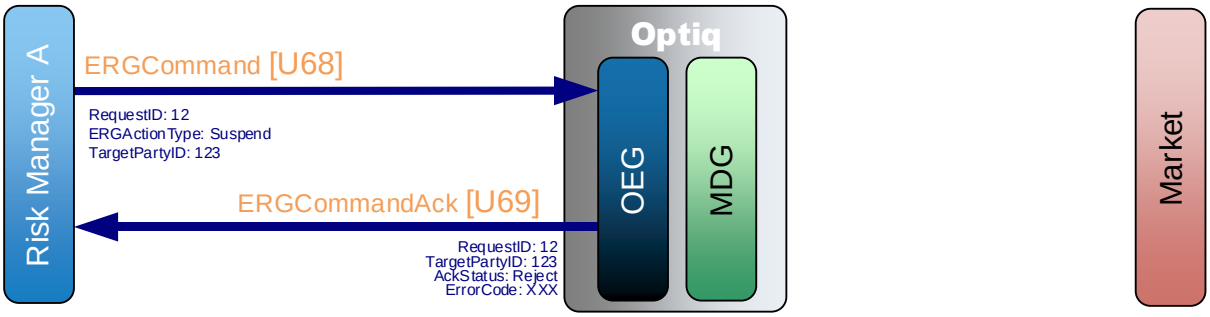
⑥ Risk Manager 123 sends a **GetRiskControls** (U70) message to request all parameters for Member A.

OEG sends back a **RiskControlDetails** (U71) message to provide all current RiskGuard controls for the Member A set by the Risk Manager 123. The provided details contain the following records: (i) Logical Access ID 22 is Suspended and (ii) Firm ID A is blocked on contract A1.

Note:

- OEG sends to the Risk Manager 123 only the controls activated by that risk manager.
- The details provided for Member A cover all Logical Access for the Equities segment.

11.7 ERG: SUSPEND COMMAND REJECTED FOR FUNCTIONAL REASONS



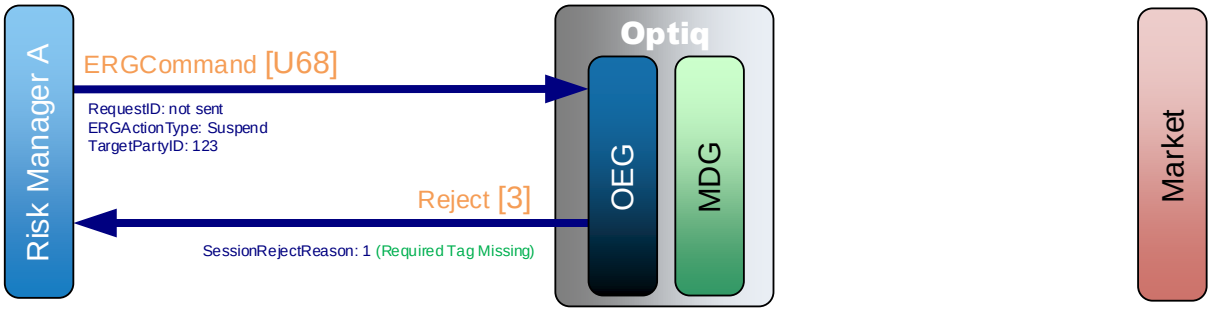
Risk Manager A sends a private **ERGCommand** (U68) message to suspend a trader or an algorithm using a short code, without providing the Firm ID.

OEG sends back a private **ERGCommandAck** (U69) message to reject the suspension with an Error Code.

The reason of the rejection can be found using the *ErrorCode* value within the *Euronext Markets - Optiq & TCS Error list file (.csv).*

No message is sent to the Market.

11.8 ERG: COMMAND MESSAGE REJECTED FOR TECHNICAL REASONS

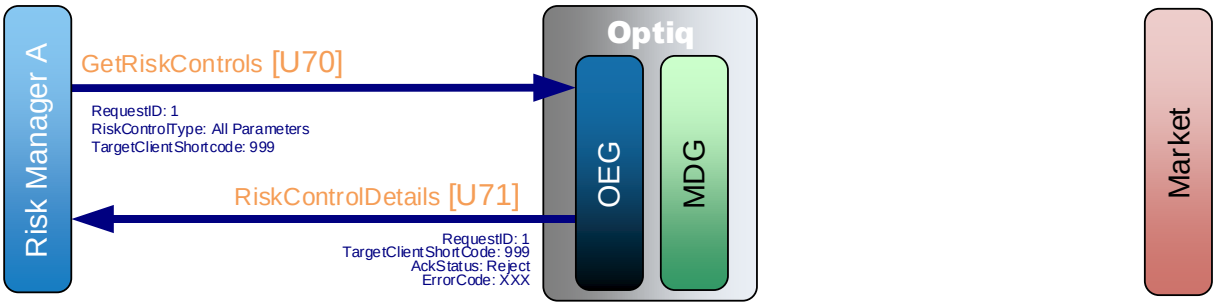


Risk Manager A sends a private **ERGCommand** (U68) message to suspend a trader or an algorithm, without populating a mandatory field *Request ID*.

OEG sends back a private **Reject** (3) message with the field *SessionRejectReason* set to '1' (Required Tag Missing) .

No message is sent to the Market.

11.9 ERG: RISK MANAGER'S REQUEST FOR SETUP DETAILS REJECTED FOR FUNCTIONAL REASONS

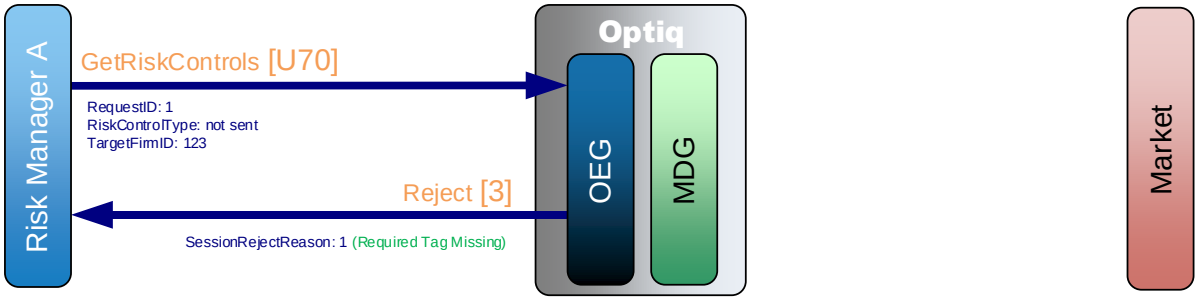


Risk Manager A sends a private **GetRiskControls** (U70) message to request details on all parameters for a given Client Identified Short Code, without providing the Firm ID.

OEG sends back a private **RiskControlDetails** (U71) message to reject the request with an Error Code. The reason of the rejection can be found using the *ErrorCode* value within the *Euronext Markets - Optiq & TCS Error list file (.csv).*

No message is sent to the Market.

11.10 ERG: RISK MANAGER'S REQUEST FOR SETUP DETAILS FOR TECHNICAL REASONS



Risk Manager A sends a private **GetRiskControls** (U70) message to request details for a given firm, without providing the Risk Control Type.

OEG sends back a private **Reject** (3) message with the field *SessionRejectReason* set to '1' (Required Tag Missing).

No message is sent to the Market.

12. DELTA NEUTRAL STRATEGY

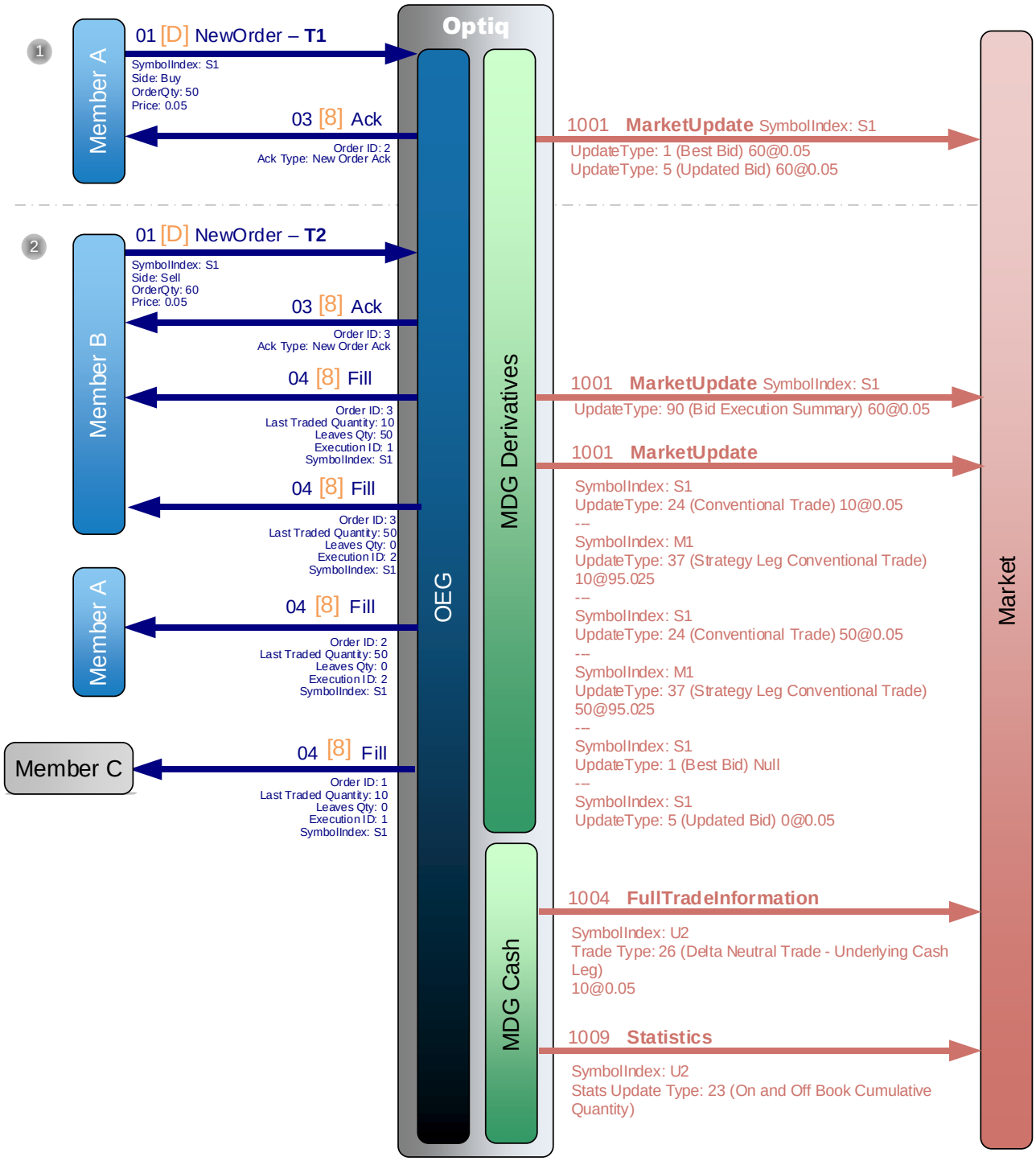
12.1 DELTA NEUTRAL STRATEGY - ORDER ON AN OPTION WITH A CASH UNDERLYING

M1
Outright Instrument

		Bid	Offer		
Time	Qty	Price	Price	Qty	Time

S1
Strategy Instrument
Call Or Put Spread vs Underlying (**M1 – M2**)

		Bid	Offer		
Time	Qty	Price	Price	Qty	Time
T0	10	0.05	0.05	60	T2
T1	50	0.05			



For submission of an order in a Delta Neutral Strategy it is mandatory to create such strategy using **SecurityDefinitionRequest** (60) (FIX c) message.

In this example the Symbol Index of the strategy created prior to the steps of this kinematic is S1.

① Member A sends a private **NewOrder** (01) (FIX D) message to enter a new Buy order with a quantity of 50 and a price of 0.05 in strategy S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The order enters the order book without matching and a public **MarketUpdate** (1001) message is sent to update the limits.

② Member B sends a private **NewOrder** (01) (FIX D) message to enter a new Sell order with a quantity of 60 and a price of 0.05 in strategy S1.

OEG sends back a private **Ack** (03) (FIX 8) message to confirm the successful receipt and technical processing of the order.

The entering order immediately matches the first order and OEG sends back a private **Fill** (04) (FIX 8) message to each member to publish the trade execution.

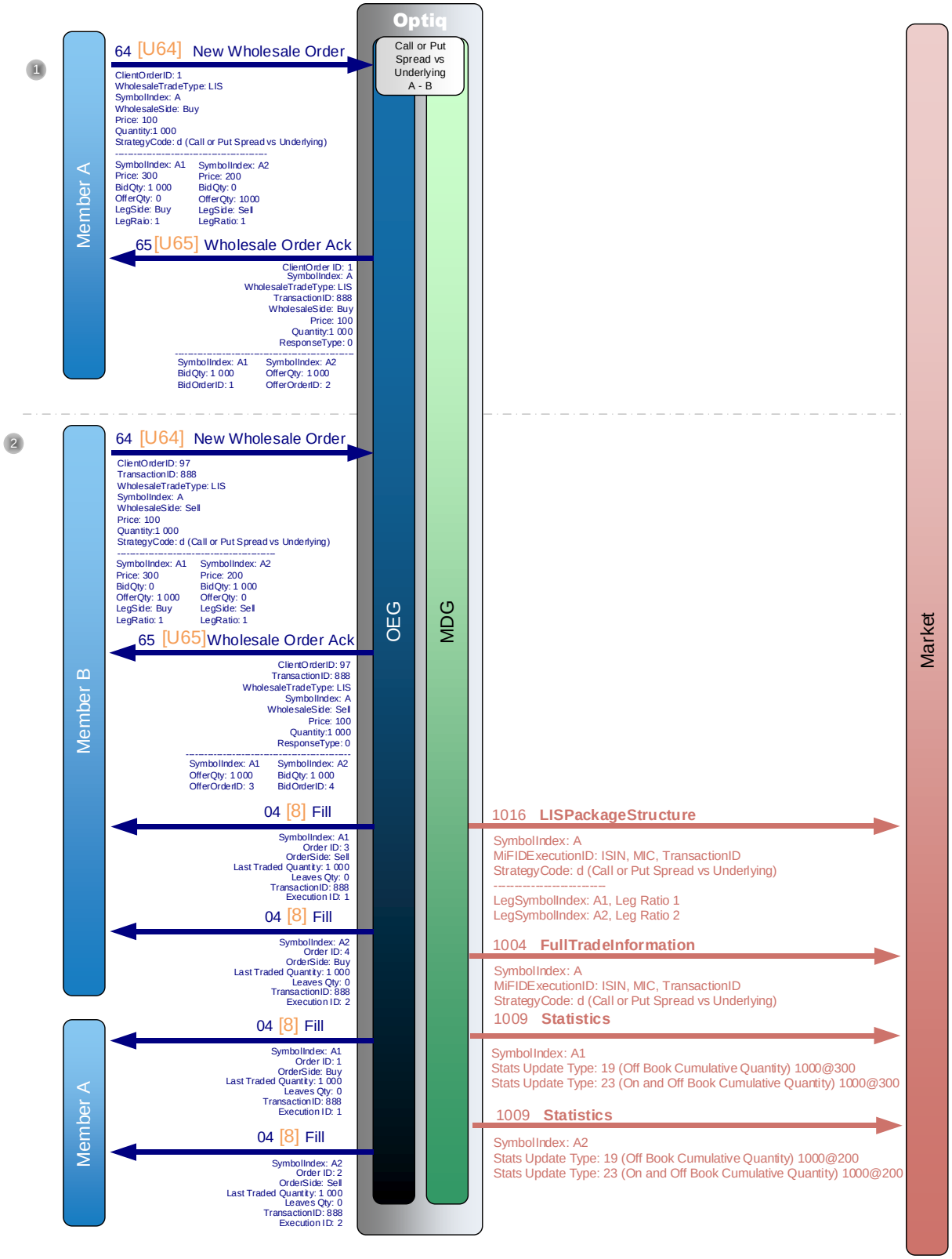
A public **MarketUpdate** (1001) message is sent at the same time to the market for the Execution Summary of the Strategy.

Following this public **MarketUpdate** (1001) messages are sent to the market for the Trade in the Strategy (S1) as a Conventional Trade, and trades for each leg of the strategy (i.e. the Trades for the individual Derivatives Outrights) that are flagged as the Strategy Leg Conventional Trade.

Following publication of updates for the strategy and strategy legs, another set of **MarketUpdate** (1001) messages are sent for BBO and Limits updates.

In parallel a public **FullTradeInformation** (1004) and **Statistics** (1009) messages are sent through MDG Cash to report the trade for the Cash Underlying leg of the trade (U2).

12.2 DELTA NEUTRAL STRATEGY - WHOLESALE SUBMISSION ON AN OPTION WITH A FUTURE UNDERLYING



For submission of an order in a Delta Neutral Strategy it is mandatory to create such strategy using **SecurityDefinitionRequest** (60) (FIX c) message.

In this example the Symbol Index of the strategy created prior to the steps of this kinematic is S1. The creation is for a strategy type of Call (or Put) Spread vs. Underlying, and within the message submitting the wholesale

transaction the details of the strategy submitted must match the defined structure and characteristics of the strategy.

① Member A sends a *private* **NewWholesaleOrder** (64) (FIX U64) message to initialize a new Wholesale transaction on Contract A, on a strategy for a Future. The submission contains the symbol index of created Delta-neutral strategy (S1) in the field *Contract Symbol Index / SecurityID* (48). The message also contains the details of the transaction on both sides.

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to confirm the successful receipt and technical processing of the Wholesale Order, with the field *Response Type* set to 0 = Accept, and provides the system generated *LIS Transaction ID / LISTransactionID* (21085).

No message is sent to the market.

② Member B sends a private **NewWholesaleOrder** (64) (FIX U64) message to respond to the Wholesale transaction on Contract A, on a strategy. The submission contains the information to setup the strategy for the transaction both sides of the transaction.

This response is targeted as a Reaction to the declaration done in Step 1 by Member A, and as such contains the *LIS Transaction ID / LISTransactionID* (21085) that was generated by system and sent to Member A, and was communicated between the two members.

OEG sends back a private **WholesaleOrderAck** (65) (FIX U65) message to Member B to confirm the successful receipt and technical processing of the Wholesale Order.

The submission by Member B completes the transaction and results in immediate match for all the legs of the strategy.

OEG generates a private **Fill** (04) (FIX 8) message for each leg of the strategy.

A public **LISPackageStructure** (1016) message, associated to its **FullTradeInformation** (1004) message are sent to the market for the transaction on Symbol Index A. This is followed by **Statistics** (1009) messages sent to update the statistics of the Cumulative Quantity, for each individual Outright leg of the Strategy.

Note: No additional messages are sent for the Future underlying. It is listed as a component of the strategy within the **LISPackageStructure** (1016).

In case of a wholesale submission in a Delta Neutral strategy with a Cash Underlying, a message **FullTradeInformation** (1004) on the Cash transaction is sent to the market via MDG Cash.

APPENDIX A: REVISION HISTORY

A.1 CHANGE SUMMARY

Version	Change Description
4.4.0	Introduction of SBE 304 – no impacts
1.1.1	The following changes have been made to this version of the document: ■ In Trading Kinematics, a note is added on the sequencing unguaranteed for BBO and limits updates between instruments in the same MarketUpdate message (case of explicit order generating implied order) ■ In Implieds with SIM: Component Implied Versus Component Implied, removal of public sending of BBO and limits updates for S2 as the original aggressive order totally matched.
1.1.0	The following changes have been made: ■ Removed section "Work in Progress" ■ In section 2.1.1 "Initialisation of a New Trading Day" Timetable is sent after Standing Data ■ In section 1.2.1 "Private Messages" added new SBE/FIX messages ■ Added section 2.7 "Request for Quote" ■ Added section 4.3 "Future Spike Protection" ■ Added sections from 6.3 to 6.5 "Implieds with EDIM…" ■ Added sections from 6.6 to 6.8 "Implieds with SIM…" ■ In section 8 "Wholesales" updated format of MiFID Execution ID in FullTradeInformation; Replaced *TransactionID* by *LISTransactionID* ■ Added section 9 "Request For Cross" ■ Added section 10 "Total Return Future (TRF) and Market on Close (MOC) ■ Added section 11 "Euronext RiskGuard (ERG)" ■ Added section 12 "Delta Neutral Strategy" ■ Updated section 2.3.1, 2.3.3, 2.4.2, 2.4.3, 5.3.2, 6.1, 6.2: Execution summary is now sent for the aggressed order side (and not the aggressive side) ■ For all kinematics having a "trade occurs": MarketUpdate message aligned with Fill message
1.0.1	The following changes have been made: ■ Addition of the LIS Package Structure message in sections 8.3 and 8.4 ■ Correction on section 4.1.3 : no 'Uncrossing' flag is sent in case the IMP was still outside of collars, but a new reservation notification and the associated Indicative Matching Price
1.0.0	First release for migration of Derivatives onto Optiq

A.2 DOCUMENT HISTORY

REVISION NO.	DATE	AUTHOR	CHANGE DESCRIPTION
4.4.0	2 Nov 2020	IT Market Services – WMA	Introduction of SBE 304 – no impacts
1.1.1	20 Sep 2020	IT Market Services – WMA	Minor update on generation of BBO and Limits with implied order
1.1.0	6 Sep 2019	IT Solutions - WMA	Second release for migration of Derivatives onto Optiq
1.0.1	13 May 2019	IT Solutions - WMA	Finalized Version for Opening of Optiq Derivatives Test Platform (EUA) and member conformance
1.0.0	30 Apr 2019	IT Solutions - WMA	First release for migration of Derivatives onto Optiq

