

GPW WATS 2.01

Native Order Gateway Specification.

Date: 3.02.2025 | Version: GPW1.5

CONTENTS

Contents2

1. Disclaimer3

2. Preface4

2.1. Target Audience4

2.2. Document's Purpose4

2.3. Associated Documents4

3. Document History6

4. Service Description14

4.1. Native Service Description14

4.2. Conditionally Required Fields16

4.3. Security Identification16

4.4. Order Management17

4.5. Trade Capture26

4.6. Quote Handling29

4.7. Cancel on Disconnect29

4.8. Mass Cancellation29

4.9. Reject Messages30

4.10. Timestamps and Dates30

5. Connectivity31

5.1. Session Identification31

5.2. Failover and Recovery31

5.3. Message Throttling35

5.4. Connectivity Policy35

6. Session Layer36

6.1. Establishing a Session36

6.2. Maintaining a Session36

6.3. Terminating a Connection37

7. Messages38

7.1. Header39

7.2. BidOfferUpdate39

7.3. ConnectionClose40

7.4. Heartbeat40

7.5. Login41

7.6. LoginResponse41

7.7. Logout42

7.8. LogoutResponse42

7.9. MarketMakerCommand42

7.10. MarketMakerCommandResponse43

7.11. MassQuote43

7.12. MassQuoteResponse46

7.13. OrderAdd47

7.14. OrderAddResponse51

7.15. OrderCancel53

7.16. OrderCancelResponse54

7.17. OrderMassCancel56

7.18. OrderMassCancelResponse57

7.19. OrderModify58

7.20. OrderModifyResponse60

7.21. Reject60

7.22. RequestForExecution61

7.23. Trade62

7.24. TradeBust62

7.25. TradeCaptureReportDual62

7.26. TradeCaptureReportResponse69

7.27. TradeCaptureReportSingle70

7.28. Order Add Example75

8. Message Kinematics76

8.1. Clob77

8.2. Off-Book (Block or Cross)95

1. DISCLAIMER

This document is for information purposes only and does not form any part of contractual documentation.

Reasonable care has been taken to ensure details contained within are accurate and not misleading at the time of publication. Warsaw Stock Exchange is not responsible for any errors or omissions contained in this document.

Warsaw Stock Exchange reserves the right to treat information contained in this document subject to later change without prior notice.

This document contains confidential information to Warsaw Stock Exchange and may not be reproduced, disclosed, or used in whole or part, in any manner, without prior written consent from the owner of this document. Information included in this document shall be maintained and exercised with adequate security measures necessary to protect confidential information from unauthorized access or disclosure.

In case of sections of documentation at a **High** level work progress according to the current version of *GPW WATS Advancement of Documentation*, Warsaw Stock Exchange will endeavor to limit changes to these sections of documents to those related to:

1. correcting errors in the documentation or in the software;
2. clarification of the documentation content or removing ambiguity;
3. implementation of approved change requests or;
4. regulatory changes.

2. PREFACE

This document has been prepared by Warsaw Stock Exchange in order to help in the implementation process of GPW WATS trading platform.

2.1. TARGET AUDIENCE

This document has been prepared to development staff, Independent Software Vendors who produce software integrated with GPW WATS, analysts, market participants and all clients who want to deepen their knowledge about GPW WATS.

2.2. DOCUMENT'S PURPOSE

The purpose of this document is to provide a full description of Native Order Gateway, which is part of GPW WATS.

2.3. ASSOCIATED DOCUMENTS

GPW WATS 2.01 Native Order Gateway Specification is a part of GPW WATS documentation set.

Please check the following documents to learn about the construction of Trading System.

- GPW WATS 1.01 Trading System.

Please check the documentation of the trading protocols supported by GPW WATS.

- **GPW WATS 2.01 Native Order Gateway Specification** (this document),
- GPW WATS 2.02 FIX Order Gateway Specification.

Please check the description of the communication with Data Distribution Service.

- GPW WATS 3.01 Market Data Protocol.

Please check the description of the communication with Internet Data Distribution System.

- GPW WATS 3.02 Internet Data Distribution System,
- GPW WATS 3.03 Streaming Messages for IDDS,
- GPW WATS 3.04 Rest API Messages for IDDS.

Please check the additional documentation, which explains other services provided within GPW WATS.

- GPW WATS 4.01 Drop Copy Gateway,
- GPW WATS 4.02 Post Trade Gateway,
- GPW WATS 5.01 Risk Management Gateway.

Please check the additional documentation describing the following:

- GPW WATS 2.03 Rejection Codes,
- GPW WATS 2.04 BenDec Message Definition Format,
- GPW WATS 4.03 Contract Notes,
- GPW WATS 6.01 Connectivity,
- GPW WATS 6.02 (ENG) Short Code Record Keeping,
- GPW WATS 6.02 (PL) Mapowanie Short Code,
- GPW WATS 6.03 Short-Long Mapper User Guide.

It is recommended to read **GPW WATS 1.01 Trading System** document first.

Gateway messages are available in the Repository shared with ISV.

The contract is based on the BenDec library which is publicly available, see [Bendec library](#).

3. DOCUMENT HISTORY

Version	Date	Description
0.51	29.06.2023	The initial publication of the documentation.
0.52	26.07.2023	Messages that have been added or updated in version 0.52: Added new messages: • MassQuote • MassQuoteResponse • RiskLimitDefinition • RiskLimitDefinitionResponse • TradeBust Modified • OrderAddResponse - field source has new Value 7 - IcebergRefill • OrderCancelResponse - field source has new Value 7 - IcebergRefill • RiskLimitBreach - field limitType has new values (301-326) RiskLimitBreach - new fields: • riskWarningLevelAction • riskLimitUtilizationAmount • riskLimitUtilizationPercent. Cancelled: • OrderAddResponse field memo was deleted.
0.53	16.08.2023	Messages that have been updated accordingly to changes in messages v0.53: MassQuoteResponse: OrderBookRebuild value added in responses.source field. OrderAddResponse: OrderBookRebuild value added in source field. OrderCancelResponse: OrderBookRebuild value added in source field.
0.54	06.09.2023	Messages that have been updated accordingly to changes in messages v0.54: Changes in MassQuote message fields: • mifidFields.client.qualifier • mifidFields.executingTrader.qualifier MassQuoteResponse message field: • responses. OrderAdd • mifidFields.client.qualifier (values) • mifidFields.executingTrader.qualifier (values) • mifidFields.investmentDecisionMaker.qualifier (values) OrderAddResponse • reason (values) • source (values Activated added) OrderCancelResponse • reason (values) • source (values Activated added) OrderModifyResponse • reason (values) RiskLimitDefinition deleted: • elementType, side field added:

Version	Date	Description
		- marketSegmentId, capacity field RiskLimitDefinitionResponse - reason (value MissingMicCode) TradeCaptureReportDual - algorithmicTradeIndicator (values) - tcrPartyBuy.mifidFields.client.qualifier (values) - tcrPartyBuy.mifidFields.executingTrader.qualifier (values) - tcrPartyBuy.mifidFields.investmentDecisionMaker.qualifier (values) - tcrPartyBuy.mifidFields.investmentDecisionMaker.qualifier (values) - tcrPartySell.mifidFields.client.qualifier - tcrPartySell.mifidFields.executingTrader.qualifier - tcrPartySell.mifidFields.investmentDecisionMaker.qualifier TradeCaptureReportResponse - reason (values) TradeCaptureReportSingle - algorithmicTradeIndicator - tcrParty.mifidFields.client.qualifier - tcrParty.mifidFields.executingTrader.qualifier - tcrParty.mifidFields.investmentDecisionMaker.qualifier
0.55	16.10.2023	Messages that have been updated accordingly to changes in messages v0.55: Changes in OrderAdd message Listed values added in reason field: - InvalidPartyRoleForCLOB (1073) - InvalidPartyRoleQualifierForPartyId (1075) - RiskLimitNotDefined (7000) - RiskMaximumOrderVolumeExceeded (7001) - RiskMaximumOrderValueExceeded (7002) - RiskOrderPriceCollarExceeded (7003). Reject message Listed values added in rejectReason field: - NA (0) - InvalidSettlementDate (4). RiskLimitDefinition message riskLimitId field added. Modified values in clientRoleQualifier field: - NA (1) - Algorithm (2) - FirmOrLegalEntity (3) - NaturalPerson (4). TradeCaptureReportDual message Deleted fields: - tradeRequestId - lastParPx - tcrPartyBuy.side - tcrPartyBuy.counterpartyId - tcrPartySell.side - tcrPartySell.counterpartyId. TradeCaptureReportResponse message

Version	Date	Description
		Added values in reason field: - SettlementDateCannotBeEarlierThanMinimumSettlementDate (2015) - SettlementDateCannotBeLaterThanMaximumSettlementDate (2016). TradeCaptureReportSingle message Deleted fields: - tradeRequestId - lastParPx. tcrParty.counterpartyId changed into counterpartyId
0.56	09.11.2023	Messages that have been updated accordingly to changes in messages v0.56: New message TestEvent has been added. Changes in Header message: msgType length value changed from 1 to 2 bytes.
0.57	30.11.2023	Messages that have been updated accordingly to changes in messages v0.57: - Unused messages removed: RiskLimitDefinition, RiskLimitDefinitionResponse, Test and TestEvent. - Rejection codes updated. - Added the value 'Stp' to the enum 'OrderSource'. Cross transactions added in sections 4.1.2 and 4.5. Order source filed values updated – section 4.4.5. New chapter 4.6 describing quotes handling.
0.58	15.12.2023	Publication of v0.58.
0.59	25.01.2024	RegularTrade value has been removed from TradeType filed from TradeCaptureReport messages. New values have been added for responses (OrderRejectReason) field: - InvalidBidAskSpread – 1208, - RequestNotAllowedForBlockInstrument – 2026, - RequestNotAllowedForCrossInstrument – 2028. New value has been added for source (OrderSource) filed: - CorporateAction - 11. New values have been added for rejectReason field: - SettlementDateRequired – 5, - TradeReportIdRequired – 6, - MissingReportIdSecondaryTradeReportIdOrTradeReportRefId – 7. MassQuoteResponse message Fields status (MassQuoteStatus) and reason (MassQuoteRejectionReason) have been added. TradeCaptureReportDual message Values for matchStatus filed have been changed to: - NA – 0, - Matched – 1, - Unmatched – 2. Values for reason (TcrRejectionReason) field have changed to: Deleted: - CannotFindMatchingFirstLeg – 2025. Added: - InvalidMatchStatus – 2029, - CrossNotAllowedOutsideOfClobInstrumentSpread – 2030, - CrossPriceNotEqualToTheReferencePrice – 2031,

Version	Date	Description
		• CrossNotAllowedDuringClobInstrumentAuctionOrSuspension – 2032, • ForbiddenSecondaryTradereportId – 2033, • UnknownSecondaryTradereportId – 2034. TradeCaptureReportSingle message Added fields: • secondaryTradeReportId Values for matchStatus field have been changed to: • NA – 0, • Matched – 1, • Unmatched – 2. Correction of the throttling mechanism description. Division into business and technical limits.
0.62	25.03.2024	The document has been updated with: • modifications of orders in terms of GTD validity (4.4.3.3), • enhancement of the description in section 4.5 Trade Capture for Cross and Block trades, • new section 4.8 Mass Cancellation, The following messages in section 7 have been updated: The list of messages is now described not by type, but by Id. New messages: • InitiateState • InitiateStateResponse • OrderMassCancel • OrderMassCancelResponse • RequestForExecution MassQuote: • stpId length has changed to 1 • count field name has changed to quotes.count • quotes field name has changed to quotes.item; its type has changed to QuotesArray • count field name has changed to responses.count • responses field name has changed to responses.item; its type has changed to QuoteOrderResponseArray; new or changed values for reason field: ○ 1028 – Description amended to "lower", ○ 1032 – Name changed to InvalidOrdTypeForSelectedMarketMode, ○ 1039 – Name changed to InvalidTimeInForceForOrderType, ○ 1040 – Name changed to InvalidTimeInForceForCurrentMarketPhase, ○ 1041 – Name changed to InvalidTimeInForceForSelectedMarketModel, ○ 1043 – New value, ○ 1044 – New value, ○ 1046 – New value, ○ 1069 – New value, ○ 1201 – New value, ○ 1203 – New value, ○ 1301 – New value, ○ 1401 – New value. • reason field – 1210 – New value added. OrderAdd: • stpId length has changed to 1 OrderAddResponse

Version	Date	Description
		- New or changed values for reason field: 1028 – Description amended to "lower", 1032 – Name changed to InvalidOrdTypeForSelectedMarketMode, 1039 – Name changed to InvalidTimeInForceForOrderType, 1040 – Name changed to InvalidTimeInForceForCurrentMarketPhase, 1041 – Name changed to InvalidTimeInForceForSelectedMarketModel, 1043 – New value, 1044 – New value, 1046 – New value, 1201 – New value, 1203 – New value, 1301 – New value, 1401 – New value. source field – new value added: 12 MassCancel OrderModify - New field added: expire. OrderModifyResponse - New or changed values for reason field: 1028 – Description amended to "lower", 1032 – Name changed to InvalidOrdTypeForSelectedMarketMode, 1039 – Name changed to InvalidTimeInForceForOrderType, 1040 – Name changed to InvalidTimeInForceForCurrentMarketPhase, 1041 – Name changed to InvalidTimeInForceForSelectedMarketModel, 1043 – New value, 1044 – New value, 1046 – New value, 1069 – New value, 1201 – New value, 1203 – New value, 1301 – New value, 1401 – New value. TradeCaptureReportResponse: secondaryTradeReportId field added, - New or changed values for reason field: 1028 – Description amended to "lower", 2024- New Value, counterpartyCode field and its type amended.
1.0	30.04.2024	Changes propagating across multiple chapters: - Harmonized order statuses with FIX Order Gateway - Introduced a field indicating the execution of an order, execTypeReason, replacing the source field. 2.3. Associated Documents Updated the list of related documents. 4.1 Native Service Description /4.1.2 Business layer Enhanced the functional scope to include "mass quotes" and mass "cancellation" Figure 2 - updated the main scenario of message exchange with Gateway about currently used status fields and execTypeReason (the successor to the source field). 4.4.5 Order Status Harmonized order statuses with FIX Order Gateway. The previously used status Acknowledged corresponds to New (similarly as in FIX).

Version	Date	Description
		Included statuses Partially filled and Expired. There is no Modification status (Replaced in FIX), which in FIX 5.0 was also treated as deprecated. Information about order execution in the execTypeReason field. The new execTypeReason field contains information similar to the ExecType/ExecTypeReason field in FIX Order Gateway. 4.4.7. Modification Specified fields that can be subject to modification. Extended the scope of modifications e.g., the expire field. 4.5.1. Block Market Model Updated information on the execType field value for executing a Trade. 4.6. Quote Handling Indicated the possibility of using Mass Quote in the Hybrid market model 4.7 Cancel on Disconnect Changed the operation of the CoD mechanism to be fully manageable by Exchange Members (added a dedicated field execInst to orders) 5.2. Failover and Recovery Expanded the description of the section by separating Failover and Recovery modes. 7. Messages Added the BidOfferUpdate message related to IPO and Tender offer Added messages for Hybrid market model: • MarketMakerCommand • MarketMakerCommandResponse • RequestForExecution RiskLimitBreach message deleted (only for RMA purpose) Removed the AlgorithmicTrade flag from mifidFields.flags (a qualifier is used to indicate the use of an algorithm) Removed BIC code from ClearingIdentifier - LEI and Custom codes are used. Adopted new values for the OrderStatus field. Introduced a field indicating the execution of an order, execTypeReason, replacing the source field. Updated codes for OrderRejectionReason and TcrRejectionReason. In the OrderAdd message, added fields: • execInst - for the purpose of the CoD mechanism • feeStructureId - fee scheme for billing purposes. In the OrderAddResponse message, added fields displayQty and filled. In the OrderCancel message, added fields mifidFields and status. In the OrderMassCancel message, added field executingTrader. In the OrderModify message, expanded the scope of modifications e.g., the expire field. In the OrderModifyResponse message, added field filled. In the TradeCaptureReportDual and TradeCaptureReportSingle, corrected the execType and tradeReportType fields, added the feeStructureId field, removed the matchStatus field (using execType = Trade), added information about the clearingMember. Updated the kinematics for the gateway (Chapter 8)
1.1	28.06.2024	Changes propagating across multiple chapters Enums TimeInForce - Indicates the order's time in force (e.g. GTC). FOK corrected to: A Fill or Kill order must be immediately fully filled or canceled. VFA corrected to: Valid For Auction. VFC corrected to: Valid For Closing. Affected messages: OrderAdd. ClearingIdentifier- Clearing member identifier type.

Version	Date	Description
		BIC - Business Identifier Code, value removed. Affected messages: MassQuote, OrderAdd, TradeCaptureReportDual, TradeCaptureReportSingle. ExecType - Type of execution being reported. Uses subset of ExecType for trade capture reports. New - Value removed. Affected messages: TradeCaptureReportDual, TradeCaptureReportSingle From TradeCaptureReportDual message has been removed feeStructuredId field. Two fields: tcrPartyBuy.feeStructureId and tcrPartySell.feeStructureId have been added. From TradeCaptureReportSingle message has been removed feeStructuredId field. The field tcrParty.feeStructureId has been added.
1.1.2	9.08.2024	ConnectionClose message: New value added in reason field: CloseOps. 4.4.7 Order Modification - MiFID Fields modification are no longer included in the list of modifications. Short Codes modification has been added to the list.
1.2	18.09.2024	TradeReportRefId and TradeReportId field lengths have been increased to 21 characters.
1.3	17.10.2024	GapFill New message added. OrderAddResponse message New values added in execTypeReason field: - CancelonBuyOnlyStateEntry - CancelonKnockedOutStateEntry - CancelByRiskManagement OrderCancelResponse message New values added in execTypeReason field: - CancelonBuyOnlyStateEntry - CancelonKnockedOutStateEntry - CancelByRiskManagement Reject message New values added in rejectReason field: - InvalidTradeReportId - InvalidGapFillSeqNum
1.4	6.12.2024	Unpublished version. All changes in this version have been documented in v1.5.
1.5	3.02.2025	total size = number of Lots price = price per unit (smallest portion) of instrument ConnectionClose message - reason field: new value added: 8 - Disconnect MassQuote message - length changed to 1200 new fields added: - feeStructureId - interestedParty - quoteId onBehalOf field has been deleted. MassQuoteResponse message new field added: - feeStructureId

Version	Date	Description
		OrderAdd message - length changed to 167 new field added: - InterestedParty - clientOrderId onBehalOf field has been deleted. TradeCaptureReportDual message - length changed to 271 new fields added: - tcrPartyBuy.interestedParty - tcrPartySell.interestedParty TradeCapture ReportSingle message - length changed to 206 new field added: - tcrParty.interestedParty Dual Sided - Accepted kinematics has been updated

4. SERVICE DESCRIPTION

4.1. NATIVE SERVICE DESCRIPTION

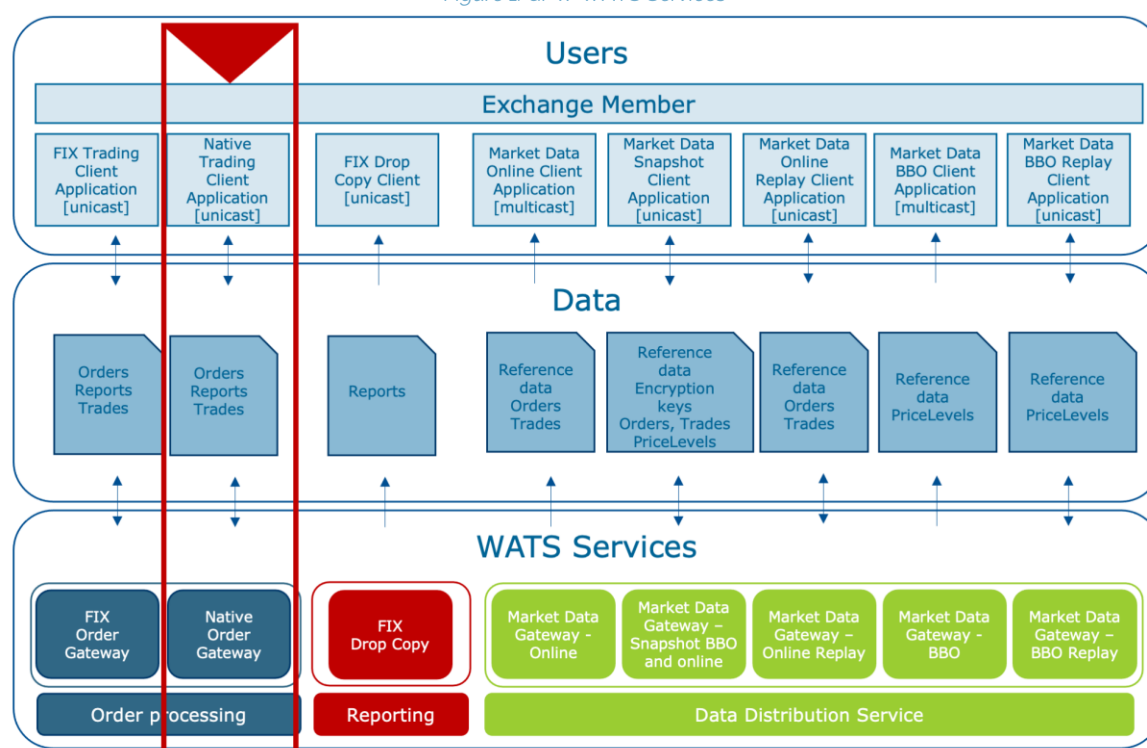
Native Order Gateway service provides a low-latency trading interface based on a proprietary binary protocol.

The main service functionalities:

- order submission,
- order modification,
- order cancellation and mass cancellation,
- transaction submission (block market),
- mass quotes.

Native Order Gateway is a point-to-point service based on the TCP-IP protocol.

Figure 1. GPW WATS Services



The following description is a brief introduction to the operation of the service, divided into the session and business layer. Detailed information can be found in the subsequent chapters of the document.

4.1.1. SESSION LAYER

To use the service, log in by sending a Login message. The message contains access parameters such as a token and a connection identifier. System confirms successful login with a LoginResult message which contains the result field set to "OK".

After logging in, a potential synchronization of missing messages takes place based on the sequential numbers exchanged during the login process (also called gap filling).

If no messages are sent within a specified time period, each party should send a Heartbeat message.

Participant can terminate the session at any time by sending a Logout message and closing the network connection. The Service responds to the Logout message with a LogoutResponse message.

Gateway can terminate the session by sending a ConnectionClose message and closes the network connection. The ConnectionClose message can be sent in a positive scenario similar to closing trading day, as well as in the case of significant session-level errors, e.g. communication protocol errors.

4.1.2. BUSINESS LAYER

Business communication is primarily based on the request - response pattern. We have the following main communication scenarios:

- Exchange Member places an order by sending an OrderAdd message. Gateway responds with an OrderAddResponse message, which confirms the successful submission of the order or otherwise rejects it,
- Exchange Member modifies an order by sending an OrderModify message. Gateway responds with an OrderModifyResponse message to confirm the accuracy of the modification or otherwise rejects it,
- Exchange Member cancels an order by sending an OrderCancel message. Gateway responds with an OrderCancelResponse message to confirm the accuracy of the cancellation or subsequently rejects it,
- Exchange Member reports a block transaction conducted outside Market by sending a TradeCaptureReportSingle or TradeCaptureReportDual message. Gateway responds with a TradeCaptureReportAck message and continues to handle the Blocked transaction,
- Exchange Member reports a Cross transaction using trade capture reports - similar to block transactions.
- Mass cancellation of orders carried out through the OrderMassCancel message.

In addition to the above request – response pattern, where Exchange Member initiates communication, it is also possible to receive ad-hoc messages from Gateway concerning the status of orders, transactions and order cancellations performed by System. Changes in the order status are sent to Exchange Member in the OrderAddResponse message. Information about transactions based on the submitted orders is provided in the Trade message. With regards to order cancellations performed by System, the OrderCancelResponse message is used. Below is a diagram illustrating the usage of key business messages.

```

sequenceDiagram
    participant NTCA as Native Trading Client  
Application -  
Exchange Member 1
    participant NTCM as Native Trading Client  
Application -  
Exchange Member 2
    participant NOG as Native Order  
Gateway
    participant SMDG as System and  
Market Data  
Gateway - Online
    participant MDCA as Market Data  
Online Client  
Application

    NOG->>NTCA: [TPC/IP]: connect (IP1, port1)
    NTCA->>NOG: Login
    NOG-->>NTCA: LoginResponse  
connectionId = 44, token = ABCDEFGH
    NOG->>NTCM: [TPC/IP]: connect (IP1, port1)
    NTCM->>NOG: Login
    NOG-->>NTCM: LoginResponse  
connectionId = 3, token = ZXCVBNML
    NTCA->>NOG: OrderAdd  
side = Buy (1)  
orderType = Limit (1), timeInForce = Day (1)  
quantity = 1000, price = 100
    NOG-->>NTCA: OrderAddResponse  
orderId = 3377768440004609, publicOrderId = 1  
status = New (1), reason = NA (1), execTypeReason = NA(1)
    NOG->>SMDG: OrderAdd
    SMDG->>MDCA: OrderAdd  
publicOrderId = 1, side = Buy  
quantity = 1000, price = 100
    NTCM->>NOG: OrderAdd  
side = Sell (2)  
orderType = Limit (1), timeInForce = Day (1)  
quantity = 500, price = 100
    NOG-->>NTCM: OrderAddResponse  
orderId = 28147566390542337, publicOrderId = 2  
status = PartiallyFilled (5), reason = NA (1), execTypeReason = FirstTradeOnAggressiveOrder (14)
    NOG->>SMDG: OrderAdd
    SMDG->>NOG: Trades
    SMDG->>MDCA: OrderExecute  
publicOrderId = 1  
quantity = 500, executionQuantity = 500  
executionPrice = 100  
OrderExecute  
publicOrderId = 2  
quantity = 0, executionQuantity = 500  
executionPrice = 100  
Trade
    NOG->>NTCA: Trade  
orderId = 3377768440004609  
price = 100, quantity = 500, leavesQty = 500
    NOG->>NTCM: Trade  
orderId = 28147566390542337  
price = 100, quantity = 500, leavesQty = 0
    NTCA->>NOG: OrderModify  
orderId = 3377768440004609  
quantity = 2000, price = 100
    NOG-->>NTCA: OrderModifyResponse  
orderId = 3377768440004609, status = PartiallyFilled (5),  
execTypeReason = Replaced(13), priorityFlag = Lost
    NOG->>SMDG: OrderModify
    SMDG->>MDCA: OrderModify  
publicOrderId = 1  
quantity = 2000, price = 100
    NTCA->>NOG: OrderCancel  
orderId = 3377768440004609
    NOG-->>NTCA: OrderCancelResponse  
orderId = 3377768440004609  
reason = NA
    NOG->>SMDG: OrderCancel
    SMDG->>MDCA: OrderDelete  
publicOrderId = 1
  
```

4.2. CONDITIONALLY REQUIRED FIELDS

One sets unused conditional and optional numeric fields should to zero. One sets unused conditional and optional enumerable fields to a value which indicates their unrequired use (e.g. not applicable).

System assigns the instrument identifier (e.i. instrumentId). On the basis of reference data, one obtains other identifiers, e.g. ISIN number.

4.4. ORDER MANAGEMENT

The chapter describes important information about orders, including their identifiers, types, and processing rules.

4.4.1. ORDER IDENTIFIERS

The following order identification fields are used in messages:

- orderId,
- publicOrderId.

4.4.1.1. orderId

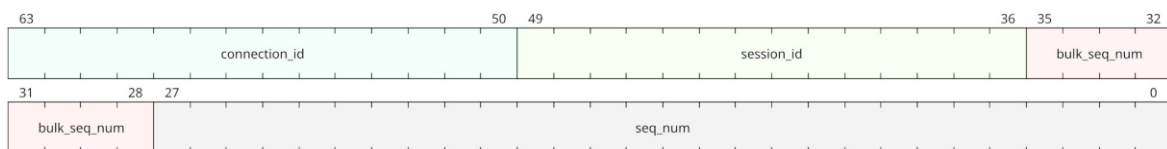
Trading System assigns a unique identifier for an order. The orderId field is unique across all trading days, all order books and all Participants. It is persistent for the entire life of an order (i.e. it does not change after an order modification). orderId is known only for Exchange Member who submits the order, Exchange and possibly for supervisory entities (e.g. Polish Financial Supervision Authority - KNF). Other Exchange Members do not know that orderId field.

An orderId field is generated for each order as a concatenation of connection id, session id, bulk sequence number and sequence number of the OrderAdd message sent by Exchange Member:

- connection Id – a connection number - connectivity parameter provided by GPW,
- session Id – a session number received during the login process,
- bulk sequence number – zero value for single orders (e.g. Market, Limit) and the value of subsequent orders for messages with multiple orders or quotes (e.g. MassQuote),
- sequence number - a sequence number of OrderAdd message.

All the above parts of the orderId field are known at the time of sending the OrderAdd message.

Figure3. orderId field construction.



Example

1. OrderAddResponse message with orderId (highlighted in bold blue):
 1b 00 05 01 00 00 00 **01 00 00 00 10 00 38 04** 01 00 00 00 00 00 00 02 01 00 01,
2. Raw bytes of order id: 01 00 00 00 10 00 38 04,
3. Order id value 0x0438001000000001 (little endian byte order) and parts value:
 - a) connection id = 0b00_0001_0000_1110 = 0x10E = 270,
 - b) session id = 0b00_0000_0000_0001 = 1,
 - c) bulk = 0b0000_0000 = 0,
 - d) sequence number = 0b0000_0000_0000_0000_0000_0000_0001 = 1.

Diagram illustrating the structure of the 64-bit connection_id, session_id, and bulk_seq_num fields:

- connection_id** (32 bits): 0 0 0 0 0 1 0 0 0 0 1 1 1 0
- session_id** (32 bits): 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
- bulk_seq_num** (4 bits): 0 0 0 0
- seq_num** (28 bits): 0 1

Then `orderId` is used across multiple messages throughout the entire life of the order.

4.4.1.2. publicOrderId

publicOrderId is unique across all order books and all Exchange Members, but only within a single trading day. On the next trading day, the numbering is reset and starts afresh (i.e. publicOrderId is not consistent for the entire life of an order). Moreover it also changes with each refill of an iceberg order or applies to transfer orders (e.g. orders with time in force set to Good Till Cancel or Good Till Date).

Gateway provides `publicOrderId` for an order only in the `OrderAddResponse` message. `publicOrderId` can be used to relate submitted orders to Market Data, however orders are modified or cancelled only using the `orderId` field.

4.4.2. ORDER TYPES

- Limit Order,
- Market Order,
- Market To Limit Order,
- Iceberg Order,
- Stop Limit Order,
- Stop Loss Order.

4.4.2.1. Limit Order

Relevant message fields:

- orderType = 1 (Limit),
- price = Limit Price.

4.4.2.2. Market Order

Market Order is an order to buy or sell without specified price. Market Order is executed at the best opposite prices with one or more counterpart resting orders.

During Continuous Trading Market Phase, Market Orders can only be submitted with IOC, FOK, VFA, VFC validity attributes, i.e. `timeInForce` = 3, 4, 5, 7.

During Auction Market Phase, Market Orders can only be submitted with VFA and VFC validity attributes, i.e. `timeInForce` = 5, 7.

Relevant message fields:

- `ordType` = 2 (Market),
- `timeInForce` = 3 (Immediate Or Cancel), 4 (Fill Or Kill), 5 (Valid For Auction), 7 (Valid For Closing).

4.4.2.3. Market To Limit

Market To Limit Order is an order to buy or sell without a specified price. Market To Limit Order is executed at the best opposite price level with one or more counterpart resting orders.

During Continuous Trading Market Phase, Market To Limit Orders can only be submitted with IOC, FOK, VFA, VFC validity attributes, i.e. `timeInForce` = 3, 4, 5, 7.

During Auction Market Phase, Market To Limit Orders can only be submitted with VFA and VFC validity attributes i.e. `timeInForce` = 5, 7.

Relevant message fields:

- `ordType` = 3 (Market To Limit),
- `timeInForce` = 3 (Immediate Or Cancel), 4 (Fill Or Kill), 5 (Valid For Auction), 7 (Valid For Closing).

4.4.2.4. Iceberg

Iceberg Order is an order, for which only a part of the total quantity is disclosed to Market via Market Data. The rest of the order is hidden (invisible) to the general market. Exchange Member submitting an Iceberg Order must specify additionally the quantity that is going to be displayed initially, i.e. `displayQty`, which:

- must be lower than the total order quantity,
- must be greater or equal to the Minimum Display Quantity set in Trading System configuration,
- may be randomized by Trading System within a specified range.

Once displayed quantity is completely filled, a new portion of the order is disclosed to Market according to the `displayQty` requested at the order entry (with possible randomization).

The value of each Iceberg Order (i.e. `price` x `orderQty`) at the moment of entry or during any modification must be equal or greater than the Minimum Iceberg Size set in Trading System configuration.

Iceberg Orders can only be submitted with DAY, GTC, GTD, GTT validity attributes, i.e. `timeInForce` = 1, 2, 6, 8.

Relevant message fields:

- `ordType` = 4 (Iceberg),
- `price` = Limit Price,
- `displayQty` = Initially Displayed Quantity,
- `timeInForce` = 1 (Day), 2 (Good Till Cancel), 6 (Good Till Date), 8 (Good Till Time).

4.4.2.5. Stop Limit

Stop Limit is an order to buy or sell at a stipulated limit price or a better one, which is activated and inserted into the order book upon reaching or exceeding pre-defined price level (called Trigger Price) by Last Traded Price on Market. Until activation, Stop Limit is hidden and cannot interact with the order book. Once activated, Stop Limit is processed the same way as a Limit Order with the same validity attribute that the original Stop Order had before its activation (nonetheless maintains its original Order Type = 5).

When submitting or modifying a Stop Limit order, the following conditions must be met:

- for buy order: Price $\geq$ Trigger Price $>$ Last Traded Price (LTP),
- for sell order: Price $\leq$ Trigger Price $<$ Last Traded Price (LTP).

Stop limit can only be submitted with DAY, GTC, GTD, GTT validity attributes, i.e. timeInForce = 1, 2, 6, 8.

The Trigger Price may be modified for inactive Stop Order, but once the order is activated, Trigger Price cannot be modified anymore.

Relevant message fields:

- ordType = 5 (Stop Limit),
- price = Limit Price,
- triggerPrice = Activation Price,
- timeInForce = 1 (Day), 2 (Good Till Cancel), 6 (Good Till Date), 8 (Good Till Time).

4.4.2.6. Stop Loss

Stop Loss is an order to buy or sell without specified price, which is activated and inserted into the order book upon reaching or exceeding pre-defined price level (called Trigger Price) by the Last Traded Price on Market. Until activation, Stop Loss is hidden and cannot interact with the order book. Once activated, Stop Loss is processed the same way as Market Order with Immediate Or Cancel (IOC) validity attribute (nonetheless maintains its original Order Type = 6).

When submitting or modifying a Stop Loss Order, the following conditions must be met:

- for buy order: Trigger Price $>$ Last Traded Price (LTP),
- for sell order: Trigger Price $<$ Last Traded Price (LTP).

Stop Loss can only be submitted with DAY, GTC, GTD, GTT validity attributes, i.e. timeInForce = 1, 2, 6, 8.

The Trigger price may be modified for inactive Stop order, but once the order is activated, Trigger Price cannot be modified anymore.

Relevant message fields:

- ordType = 6 (Stop Loss),
- triggerPrice = Activation Price,
- timeInForce = 1 (Day), 2 (Good Till Cancel), 6 (Good Till Date), 8 (Good Till Time).

4.4.3. ORDER VALIDITY TYPES

Exchange Members may submit orders with the following validity attributes (timeInForce field):

- Day (DAY),
- Good Till Cancel (GTC),
- Good Till Date (GTD),

- Good Till Time (GTT),
- Immediate Or Cancel (IOC),
- Fill Or Kill (FOK),
- Valid For Auction (VFA).
- Valid For Closing (VFC).

Not all validity attributes may be used with every order type throughout various Market Phases.

4.4.3.1. Day (DAY)

Day order is valid until the end of the current trading day only. Unexecuted Day orders are expired by Trading System at the end of the current trading day.

Day validity attribute may be used with all order types, except for Market Order and Market To Limit.

Relevant message fields:

- timeInForce = 1 (Day).

4.4.3.2. Good Till Cancel (GTC)

Good Till Cancel order is valid until it gets fully executed or canceled by the submitter or market operations (whichever comes first).

There is a maximum number of days threshold set up in Trading System configuration, that defines for how long a Good Till Cancel order may rest in the order book (ranging from 1 to 365 days). After that time elapses, GTC order gets expired as well.

Good Till Cancel validity attribute may be used with all order types, except for Market Order and Market To Limit.

Relevant message fields:

- timeInForce = 2 (Good Till Cancel).

4.4.3.3. Good Till Date (GTD)

Good Till Date order is valid until the end of the specified day.

If the expiry date specified for an order is not a trading day, the order gets expired at the end of the previous day.

The specified date cannot exceed the maximum number of days threshold defined by Market Operator (ranging from 1 to 365 days).

Good Till Date validity attribute may be used with all order types, except for Market Order and Market To Limit.

It is possible to modify orders with GTD validity type. Modification of the date is possible from the current date to the limit for GTD (otherwise the modification is rejected). Modified orders do not change priority.

Relevant message fields:

- timeInForce = 6 (Good Till Date),
- expire = Expiry Date.

4.4.3.4. Good Till Time (GTT)

Good Till Time order is valid until the specified time of the current trading day.

If the expiry time specified for an order is later than the end of the last Market Phase, the order gets expired at the end of the current trading day.

Good Till Time validity attribute must be used with the expiry field set to the current date and time in the future.

Good Till Time validity attribute may be used with all order types, except for Market Order and Market To Limit.

Relevant message fields:

- timeInForce = 8 (Good Till Time),
- expire = Expiry Time (UTC timestamp with current date).

4.4.3.5. Immediate Or Cancel (IOC)

Immediate Or Cancel order must be executed immediately, in full or partially, upon order entry. IOC orders may be matched against one or more opposite orders (with the price limits within Trade Price Collars). Any unexecuted part of an IOC order is automatically canceled by Trading System.

Immediate Or Cancel orders may be submitted during Continuous Trading only.

Immediate Or Cancel validity attribute may be used with all order types, except for Stop Limit, Stop Loss and Iceberg.

Relevant message fields:

- timeInForce = 3 (Immediate Or Cancel).

4.4.3.6. Fill Or Kill (FOK)

Fill Or Kill order must be executed in full, immediately upon order entry. FOK orders may be matched against one or more opposite orders (with the price limits within Trade Price Collars). An unexecuted FOK order is automatically canceled by Trading System.

Fill Or Kill orders may be submitted during Continuous Trading Market Phase only.

Fill Or Kill validity attribute may be used with all order types, except for Stop Limit, Stop Loss and Iceberg.

Relevant message fields:

- timeInForce = 4 (Fill Or Kill).

4.4.3.7. Valid For Auction (VFA)

Valid For Auction order is activated and inserted into the order book at the beginning of the next Auction Market Phase (whether scheduled or unscheduled). VFA order submitted in Continuous Trading is hidden until its activation (i.e. it is not published via Market Data) and cannot interact with the order book. Once activated, VFA order is valid until the end of Auction and gets expired by Trading System immediately after uncrossing (if left unexecuted). VFA orders are valid for the current trading day only. Activated VFA orders retain their original priority timestamp from the moment of submission.

Valid For Auction validity attribute may be used with all order types, except for Stop Limit, Stop Loss and Iceberg.

Relevant message fields:

- `timeInForce = 5` (Valid For Auction).

4.4.3.8. Valid For Closing (VFC)

Valid For Closing order is activated and inserted into the order book at the beginning of the Closing Auction Market Phase only. VFC order submitted in Continuous Trading or Auction other than Closing, is hidden until its activation (i.e. it is not published via Market Data) and cannot interact with the order book. Once activated, VFC order is valid until the end of the Closing Auction and gets expired by Trading System immediately after the uncrossing (if left unexecuted). VFC orders are valid for the current trading day only. Activated VFC orders retain their original priority timestamp from the moment of submission.

Valid For Closing validity attribute may be used with all order types, except for Stop Limit, Stop Loss and Iceberg.

Relevant message fields:

- `timeInForce = 7` (Valid For Closing).

4.4.4. ORDER CAPACITY

Order capacity defines the capacity of the company, which submits an order. The capacity field values:

- 1 - Agency mapped as AOTC (Any other capacity),
- 2 - Principal mapped as DEAL (Dealing on own account),
- 3 - Riskless Principal mapped as MTCH (Matched principal).

It is required to set the correct capacity field value by Exchange Member.

4.4.5. ORDER STATUS

The order status informs on the state in which it is in GPW WATS. With regards to orders submitted by Exchange Member, the status may take the values acknowledged, cancelled, rejected or filled. Successful modification of the order by Exchange Member results in the modified status. Similarly, a successful cancellation changes the status of the order to be cancelled.

Exchange Member may receive the following status for orders from Gateway:

- New (1) – status in which an order is accepted into the order book until there is any execution on it (trade) or until an order is cancelled or expired. In case aggressive order was partially or fully filled there is no New status assigned to order. To help Exchange Member to identify such a situation, the field `execTypeReason` is set to value `FirstTradeOnAggressiveOrder` (First trade on aggressive order),
- Partially filled (7) – status in which there was at least one execution on an order until the order is fully filled (there is no remaining quantity on it) or until the order is cancelled or expired,
- Filled (4) - status in which an order is fully executed (there is no remaining quantity on an order),
- Cancelled (2) - status in which an order is canceled by Exchange Member, Trading System (including situation when an incoming IOC/FOK order cannot be fully executed upon arrival) or

by the Market Operations. Additional information about the reason for unsolicited cancellation is provided in the `execTypeReason` field,

- Rejected (3) - status in which an order is rejected by Trading System upon entry due to various business validations,
- Expired (8) – status in which an order is automatically removed from the order book due to its validity condition specified in `timeInForce` field (including removal of unexecuted part of VFA/VFC order after the Auction).

Information about the status of orders is sent in `OrderAddResponse`, `OrderCancelResponse` and `OrderModifyResponse` messages.

The reason for the order status change may be an order modification or cancelation by the Exchange Member, orders matching, System actions (e.g. cancelation via Cancel on Disconnect mechanism) and session supervision team operations.

More information concerning the reason for the order state change can be found in the `execTypeReason` field description (see `execTypeReason` field description in `OrderAddResponse` message). The field informs why an order was executed and about order lifecycle events. It takes the following values:

- `CancelOnDisconnect` – cancelled order by Cancel on Disconnect mechanism,
- `Expired` - cancelled order by System in the case of expiry,
- `Triggered` - notification of Stop or VFA/VFC order activation,
- `CancelOnSuspension` - cancelled order by Trading System in the case of instrument suspension,
- `OrderRestatement` - reinstated order,
- `IcebergOrderRefill` - iceberg order refill,
- `CancelByStp` – cancelled order due to self-trade prevention,
- `CancelByCorporateAction` - cancelled order due to submitted Corporate Action,
- `CancelByMassCancel` – an order cancelled due to mass cancel,
- `CancellocFokOrder` – an order cancelled due to IOC/FOK,
- `CancelByMarketOperations` – an order cancelled due to market operations,
- `Replaced` – order modified,
- `FirstTradeOnAggressiveOrder` – new and aggressive order was filled or partially filled,
- `Rejected` - order rejected.

4.4.6. CANCELLATION

Cancellation of a previously submitted order by Exchange Member is only possible on the same connection id on which the order was submitted. For example, an order submitted on the connection id = 123 cannot be removed from the connection id = 242.

The order is cancelled by sending the `OrderCancel` message with the appropriate `orderId` field (i.e. the `orderId` received in the `OrderAddResponse` message)

Connection allocation management is the responsibility of the Exchange Member.

Deleting orders is subject to business validation, in particular, it is not possible to delete an order which has already been fully executed.

More information about connections is presented in the Connectivity chapter.

4.4.7. MODIFICATION

Modification of the order, similarly to its cancellation, is only possible on the connection on which the order was previously sent (i.e. the same connection id).

The order is modified by sending the OrderModify message with the appropriate orderId field (i.e. the orderId received in the OrderResponse message).

Exchange Members are allowed to modify the following order characteristics:

- total quantity,
- initial displayed quantity (in case of Iceberg Orders only),
- price,
- trigger price (in case of Stop Orders only),
- expire (in case of timeInForce = GTD (i.e. Good Till Date),
- Short Code.

MiFID fields shall be provided as in a OrderAdd message or modified. MiFID fields will be validated in the same way as in the case of the OrderAdd message.

4.4.8. ACCOUNT STRUCTURE

There are the following types of accounts (i.e. accountType field):

- Customer - account is carried on the customer side of the books,
- House – house trader.

If there is no account placed in the order, the accountType field should be set to "Missing" and the account field should be filled with zeros.

4.4.9. TEXT FIELDS

A character in the text field is treated as an unsigned byte type. In the trading_port.json contract, the above type is called AnsiChar. The string is created as an AnsiChar array. Examples of text fields are account, token, micCode etc.

If the length of the text is less than the length of the field, the empty space should be filled with zeros.

ASCII characters, in the range 0 – 127, should be used. Messages containing text fields with characters outside the above range is rejected.

4.4.10. ORDER BOOK RESTATEMENT

At the beginning of each trading day, Exchange Members receive incoming orders from Gateway which were placed during the previous session and are still valid (e.g., orders with a Time In Force equal to GTD or GTC), known as restated orders.

More information can be found in the document *GPW WATS 1.01 Trading System*, chapter Trading Schedules.

Restated orders are sent using an OrderAddResponse message with fields set to the values from the end of the previous day. An order status is also set to the last status from the previous day (e.g. modified, filled). The field execTypeReason is set to a Reinstated value.

Reinstated orders are not sent publicly in Market Data, only to Exchange Members which submitted them.

Expired orders are not transferred – reinstated (e.g. order with TIF = Day, or Good Till Date orders with a past expiry date).

UNSOLICITED ORDER CANCELLATIONS GENERATED BY TRADING SYSTEM

Unsolicited order cancellations generated by System is carried out using OrderCancelResponse messages with the following characteristics:

1. order execTypeReason (i.e. execTypeReason field) takes a different value than submitted. This value explains the source of the order cancellation e.g. Cancel on Disconnect mechanism,
2. reason – a reason for cancelation if available.

4.5. TRADE CAPTURE

There are several situations where transaction capture reports are exchanged:

1. Block transactions (field tradeType - BlockTrade):
 - o Exchange Member may send a dual transaction capture report in a situation where a transaction involving clients of a given Exchange Member was made on the block market. The TradeCaptureReportDual message is used by Exchange Member as a dual transaction capture report. Gateway response should be the TradeCaptureReportResponse message.
 - o Exchange Member may send a single transaction capture report which requires confirmation by the other party. System ensures correct message exchange between the transaction parties. The TradeCaptureReportSingle message is used by Exchange Member as a single transaction capture report. Gateway response should be the TradeCaptureReportResponse message.
2. Cross transactions (field tradeType - PrivatelyNegotiatedTrade):
 - o Exchange Member may send a dual transaction capture report (a TradeCaptureReportDual message) in a situation where a cross transaction is submitted.

FIX Order Gateway equivalent of the message about transaction capture reports are TradeCaptureReport (AE)/ TradeCaptureReportAck (AR) messages.

4.5.1. BLOCK MARKET MODEL

Block trading is a market model in which trades are concluded bilaterally in the process of the negotiations between the counterparties (beneficiary clients). The beneficiary clients may belong to one or a few Exchange Members. The negotiations usually take place beyond Trading System infrastructure. Block trading is concluded when the respective trade entry is recorded in Trading System based on the Trade Capture Report message/messages submitted by Exchange Members.

Block Trade facility enables Exchange Members to report pre-arranged large trades through Market Operator Trading System.

The value of Block Trades must not be lower than Large In Scale value. The price of the Block Trade must be within thresholds set by the exchange. Price thresholds can be set on different levels depending on the session Phase of the same instrument in the Central Limit Order Book. Only instruments traded in the Central Limit Order Book are subject to Block Trades.

Block Trade is reported to the exchange by entering "one-party" or "two-party" Trade Capture Report. In the "one-party for pass-through" Block Trade the beneficiary clients belong to different Exchange Members. The initiator who enters one-party Block Trade requires confirmation by the counterparty to achieve the Block Trade. In "two-party" Block Trade both beneficiary clients belong to the same Exchange Member. "Two-party" Block Trade is used to report internal trade where the trade can be concluded and reported between two clients of Exchange Member (Order Capacity; Agency vs Agency) or between the firm and the client (order Capacity: Principal vs Agency).

Block Trade features:

- Despite the fact that MiFID 2 tick size regime does not apply to Block Trades, simplified tick size tables for block instruments have been adopted,
- Both parties to the Block Trade must be the Exchange Member,
- Block Trades are reported by using the Trade Capture Reports messages.
- Block Trades are not pre-transparent thus not matched Trade Capture Reports (i.e. the `execType` field is not set to the value `Trade`) are not disseminated in Market data stream,
- Block Trades are post-trade transparent thus matched Block Trades (i.e. the `execType` field is set to the value `Trade`) are immediately disseminated in Market Data stream,
- Block Trades do not update the Last Trade Price.
- Block Trades are not taken into account when calculating the opening price or the closing price.

4.5.1.1. Trade Capture Report Dual for Block Trades

`TradeCaptureReportDual` is applied when both beneficiary parties to the trade belong to the reporting firm (Exchange Member). The `TradeCaptureReportDual` is accepted by Gateway by sending back `TradeCaptureReportResponse` with the field `status = 2` (Accepted). If `TradeCaptureReportDual` is rejected, then Exchange Member receives the `TradeCaptureReportResponse` with `status = 3` (Rejected) with the respective information about the reason for rejection provided in the **reason** field.

4.5.1.2. Trade Capture Report Single for Block Trades

`TradeCaptureReportSingle` is applied when the beneficiary parties to the trade belong to different Exchange Members. The applied one-sided model is the "one-party report for pass-through". The initiator submits a report (first leg to the trade), which is accepted by the counterparty to the trade.

This is the basic `TradeCaptureReportSingle` flow enabling Exchange Members to conclude the block trade:

- The initiator submits the first leg to the trade by sending `TradeCaptureReportSingle` with the values:
 - a) `tradeReportTransType = 1` (New),
 - b) `tradeReportType = 1` (Submit).
- The first leg is accepted by System by sending `TradeCaptureReportResponse` with the `status = 2` (Accepted).

- System forwards the TradeCaptureReportSingle to the counterparty by sending notification with the values:
 - a) tradeReportTransType = 1 (New),
 - b) tradeReportType = 2 (Alleged).
- The counterparty accepts the first leg by sending TradeCaptureReportSingle with the values:
 - a) tradeReportTransType = 3 (Replace),
 - b) tradeReportType = 3 (Accept).
- System sends the TradeCaptureReportSingle informing the initiator and the counterparty about the concluded trade with the values:
 - a) execType = 15 (Trade),

4.5.2. CROSS TRADES

For a given instrumentId, Cross trades can only be submitted if the corresponding instrumentId with the same ISIN code in the Order Book is in the Continuous Trading Phase (variable price and fixed price), i.e. in:

- Continuous price and time (ContinuousPriceTime),
- Continuous time at reference price (ContinuousPriceTime),
- Trade at last (ContinuousLastAuctionTime).

Cross trades do not change Last Trade Price for an instrument with the same ISIN code in the central order book.

Cross trades are transmitted to System via Trade Capture Report message. This is a dual-sided TradeCaptureReportDual.

The content of the TradeCaptureReportDual message for Cross trades is analogous to that for bilateral Block trades, if the following tags have the following fixed values:

- tradeType = 22 (PrivatelyNegotiatedTrade),
- settlementDate = T+2 .

The values in tradeReportType have been reduced to the requirements of a Cross transaction to:

- 1 = Submit,
- 7= Trade Report Cancel.

Cross transaction scenario:

1. Transmission of a TCR Cross to System with a tradeReportType field with the value 1 = Submit.
2. For a valid TCR Cross, System sends a TradeCaptureReportResponse with an Accepted status (status = 2 (Accepted)).
3. For an invalid TCR Cross, System sends a TradeCaptureReportResponse with status Rejected (status = 3 (Rejected)).

4.6. QUOTE HANDLING

Market makers send quotes into a market using a MassQuote message. For a given instrument, it is possible to send one quotation. The MassQuote message enables the sending of up to 30 quotations.

The Trading Platform sends a confirmation of the accepted and rejected quotations by transmitting a MassQuoteResponse message.

Quotations rules:

- Two-sided quotes are allowed,
- Only one market maker quotation is allowed per instrument,
- MassQuoteResponse message includes all submitted quotations along with their acceptance status (e.g. Accepted, Rejected),
- Modification of submitted quotations involves placing another MassQuote message with changed values for price or quantity,
- Cancellation of submitted quotations involves placing another MassQuote message with the quantity set to 0,
- Mass Quote messages can be used on CLOB and Hybrid market model.

4.7. CANCEL ON DISCONNECT

Cancel on Disconnect (CoD) mechanism prevents orders from remaining outside the Exchange Member's control after an incorrect disconnection from System, i.e. without logging out.

Disconnection from Gateway is understood as breaking the TCP/IP connection or lack of communication from the Client Application, including the lack of Heartbeat messages for a period of 12 seconds (so-called grace period).

The CoD is a sub-service of Gateway service and can be enabled for any Exchange Member connection. Elimination of orders follows as per the CoD mechanism rules on a given trading connection. If the user has more than one connection, then the mechanism can be activated on each of these connections or only on chosen connections. The GPW is responsible for the configuration of the service and performs it at Exchange Member request.

Orders in which the Exchange Member has set the CancelOnConnectionLoss flag in the execInst field are exclusively subject to the CoD mechanism.

4.8. MASS CANCELLATION

Mass Cancel functionality allows Exchange Member to cancel all active orders according to the selection criteria they have specified. In the case of a previously partially executed order, only the remaining unexecuted part of the order is cancelled. This means that Mass Cancel doesn't affect the already executed volume of the order. Mass Cancel functionality works only for CLOB market model (i.e. the Mass Cancel functionality does not include the cancellation of unmatched Trade Capture Reports).

This functionality can only be used by Exchange Member in respect of orders that they have sent themselves. Sending a Mass Cancel message is not possible via the DEA (Direct Electronic Access). Exchange Member must indicate the Executing Trader performing the operation (the executingTrader field) and provide shortcode.

Exchange Member performs Mass Cancel by sending the OrderMassCancel message. The OrderMassCancel message allows the cancellation of all active orders placed on all connections (ConnectionID) of a Participant. OrderMassCancel message contains fields that define the selection criteria for cancelled orders. These fields can be divided into two groups:

- General criteria specified in the massCancelRequestType field,
- Narrowing criteria contained in the body of the message (e.g. targetPartyId).

Selection Criteria

Three main selection criteria are available for the mass cancellation of orders. These main criteria are disjoint and are contained in the required massCancelRequestType field:

- 1 = CancelForSecurity,
- 7 = CancelAllOrders,
- 9 = CancelOrdersForMarketSegment,

In addition, a number of narrowing criteria is available that only work in combination with one of the above three main criteria for Mass Cancel. It is allowed to select only one narrowing criterion from the targetPartyRole, which automatically means that the value of the targetPartyId field should correspond to the selected role:

- None and LiquidityProvider – targetPartyId equals 0,
- SenderLocation - targetPartyId is set to connection_id,
- ClientID, ExecutingTrader and InvestmentDecisionMaker - targetPartyId is set to the short code of a party.

4.9. REJECT MESSAGES

Messages sent by Exchange Member can be rejected at the session level and at the application level.

The Reject message informs on the message rejection at the session level i.e. in a situation where the message cannot be considered valid and can no longer be processed. Reason for rejection are given in the rejectReason field (e.g. Invalid message type).

Session-level valid messages related to submitting, cancelling and modifying orders may be rejected at the application level. Relevant information about the rejection is provided in the status and reason fields of the message which are a Response to the Exchange Member's message.

4.10. TIMESTAMPS AND DATES

The time and date are stored on the Timestamp type, which is based on a 64-bit unsigned integer. On the Timestamp type, the time is given within nano second accuracy.

Below are the fields in which the time is given and the rules for filling them in:

- Expire – in the case of GTT orders, only the time is filled in, in the case of GTD orders, only the date is filled in.

5. CONNECTIVITY

This chapter provides information regarding: logon authentication, network parameters, failover and recovery.

5.1. SESSION IDENTIFICATION

5.1.1. LOGIN PARAMETERS

Establishing a session with GPW WATS requires logging into System. To do this, it is necessary to have a token associated with a unique connection ID. The token and connection ID are fields of the login message. Both parameters are obtained from GPW. You cannot log in with the same connection ID multiple times.

5.1.2. NETWORK CONNECTION

IP addresses and ports for Gateway are published in a separate configuration document. There are 2 IP addresses available for each connection. The primary address for typical use and secondary address used in the case of failure.

5.1.3. TECHNICAL SESSION IDENTIFICATION

The technical session begins when System starts up and ends when it is turned off. Thus, it typically covers a trading day. In special cases, such as an emergency system restart, another technical session is started. The technical session ID is published in the StartOfTechnicalSession market data message as a field sessionId. It is also possible to obtain the current session id when logging in. Then the session id is given in response to the login message.

The session ID is used to create an order ID (see section *Order Identifiers*).

5.2. FAILOVER AND RECOVERY

Due to a reduction to operational risk within GPW WATS, the failover and recovery mechanism has been designed.

Failover and recovery modes:

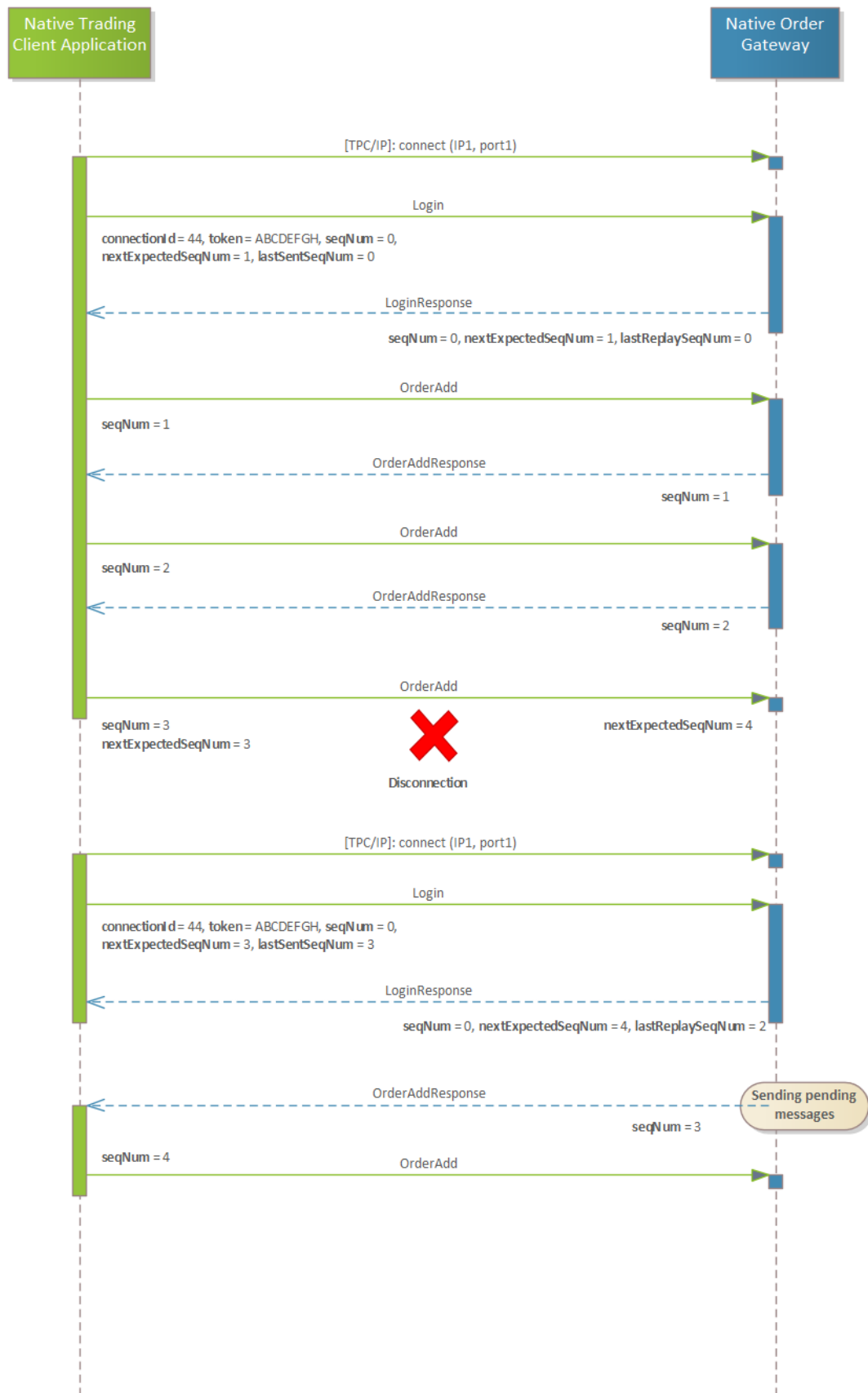
1. **Failover Mode:** This mode is activated in response to temporary connectivity losses. The system is designed to allow reconnection attempts within a timeframe of several seconds, up to approximately fifteen seconds. If the connection cannot be re-established after multiple attempts, users will be required to transition to Recovery Mode .
2. **Recovery Mode:** This mode is implemented when there is a need to shift sessions to an alternative gateway. An alternative gateway may be located at a different SITE.

5.2.1. FAILOVER MODE

In the case of temporary and short unavailability of Gateway, Exchange Member should attempt to re-establish the connection and login to Gateway. Subsequently the messages are resynchronized between Exchange Member and Gateway based on the sequence numbers included in the Login and LoginResponse messages. There are nextExpectedSeqNum and lastSentSeqNum in the Login message.

The `nextExpectedSeqNum` field means the next expected message sequence number to be received and the `lastSentSeqNum` field means the last sent sequence number. The `LoginResponse` message also contains the `nextExpectedSeqNum` field and, additionally, the `lastReplaySeqNum` field indicating the number of the last sent message or the last message currently waiting to be sent, if any. Any gap in the messages is filled by sending missing messages by each party. The diagram below shows such a situation.

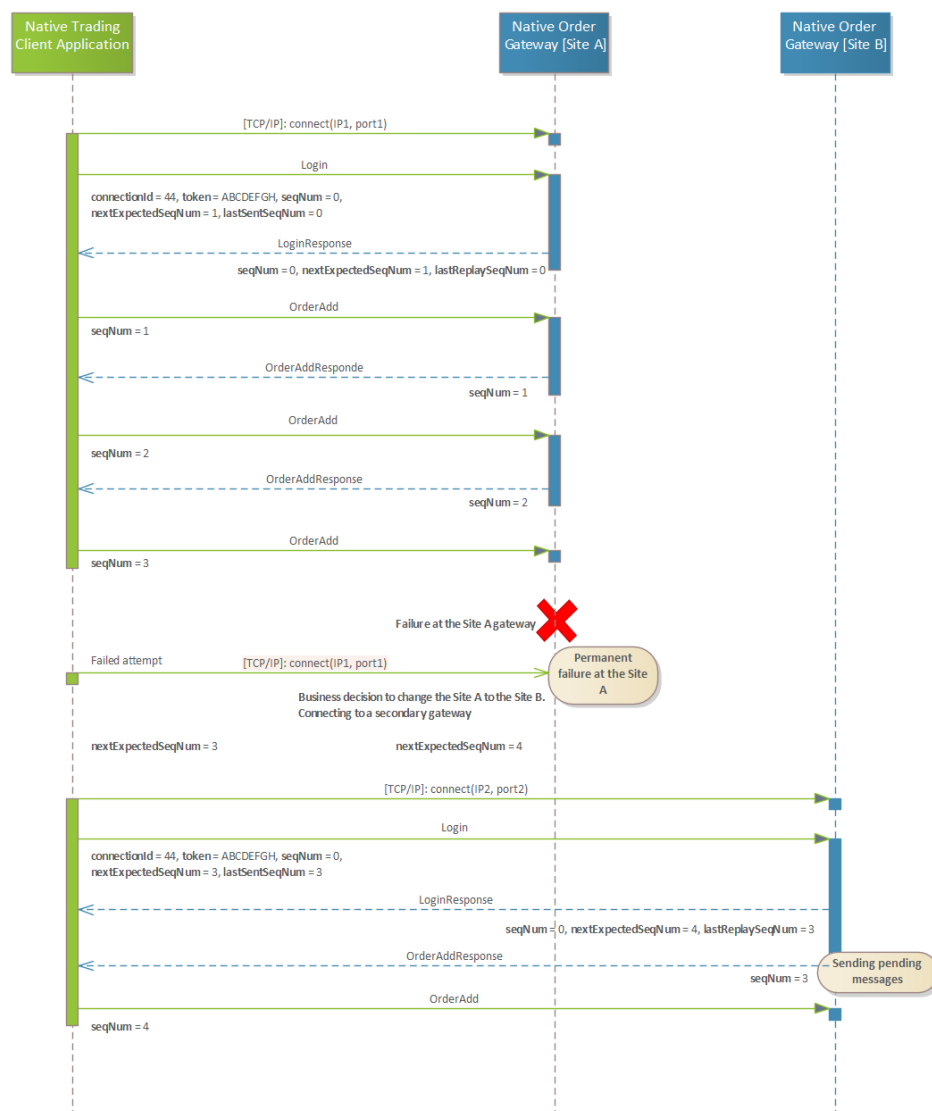
Figure 5. The synchronization after disconnection scenario.



5.2.2. RECOVERY MODE

When the primary Gateway is unavailable for a long period (minimum 15 seconds), the procedure to switch to a secondary could be launched. It is a business decision of Exchange Member. The connection is established with the same login parameters but on a new IP address and a port number. After logging in, business messages are synchronized in the same manner as in the first case (i.e. failover mode). The diagram below shows such a situation. Exchange Member should number subsequent application messages in accordance with the sequence number received in the LoginResponse (the nextExpectedSeqNum field).

Figure 6. Switching gateway location in the case of disconnection.



In the event that a full restoration of the message state is necessary on the side of the Exchange Member, the nextExpectedSeqNum field in the Login message will be set to 1 which will enable the recovery of the entire state of messages received from System.

In case of further connection issues, please contact the System Operator.

Please note that when using the CoD mechanism and disconnecting without a logout message, which is natural for most failures, orders are cancelled (see section Cancel on Disconnect).

5.3. MESSAGE THROTTLING

The throttling mechanism protects Gateway against overloading with an excessive number of input messages.

There are 2 limits set in Gateway:

1. business throttling - after exceeding the business limit level, consecutive messages are rejected using Reject messages with the reason set to MaxThroughputExceeded,
2. technical throttling - after exceeding the technical limit level, the connection is closed by sending a ConnectionClose message with the reason set to AntiFloodingThresholdExceeded and then the socket is closed.

Limit calculations take into account all messages. However, Logout and OrderCancel messages will not be rejected.

The MassQuote messages are treated differently. When calculating limits for MassQuote, each side of each quotation is counted as a separate message. For example, a MassQuote message containing 8 quotations is counted as 16 messages.

The above limits are calculated over a period of time configurable in Gateway (i.e., measurement period).

The throttling parameters are configurable at Gateway service as follows:

- The business limit level is configured for each connection and depends on the level of service provision agreed upon between System Operator and the Exchange Member. The measurement period - last 10 seconds.
- The technical limit level is configured as an internal gateway parameter and is set to 5000 messages per second. The measurement period - last 100 milliseconds.

In addition, after exceeding the technical limit level, it is not possible to log into System for 10 seconds.

Note: the measurement period means the period of time during which messages are counted and the limits which have possibly been exceeded are determined.

5.4. CONNECTIVITY POLICY

Connectivity policy is described in the document *GPW WATS 6.01 Connectivity*.

6. SESSION LAYER

6.1. ESTABLISHING A SESSION

Starting a session between the Native Trading Client Application and Gateway begins with establishing a TCP/IP connection and sending the correct Login message. The message should contain the connection parameters presented in the Connectivity chapter. In particular, the Login message should contain a token and a connection identifier. Both of the above parameters are provided by System Operator.

When successfully logging in, the client receives a LoginResponse message with the result field equals "OK". When there is a business level error, a reason is given in the result field, e.g. invalid token.

When there are protocol errors or it has not been possible to establish a session, Gateway sends the ConnectionClose message which contains the reason and then the TCP/IP connection would be closed.

Each user can only be logged in once with the same credentials. When the already logged user tries login again during another established session, Gateway sends the LoginResponse message with the result field set to "AlreadyLoggedIn".

If Participant exceeds the number of login attempts, Gateway blocks the account and the LoginResponse message is set to "AccountLocked". The limit of failed login attempts is set to 5.

The maximum limit of the login attempts with Gateway is set up as once every 3 seconds. In the situation of a failed login attempt, Exchange Member is advised to contact the GPW service desk for further assistance.

6.2. MAINTAINING A SESSION

6.2.1. SEQUENCE NUMBER

Sequence numbers are only assigned to business messages within a trading day. Each trading day is associated with a session id available after logging in (see. a LoginResponse message and the sessionId field) or in market data (see. StartOfTechnicalSession and EndOfTechnicalSession messages). The sequence numbers start with 1 and are incremented by 1 in each sent business message. Session messages have a sequence number set to 0.

The following is a list of session message types (i.e. messages with a sequence number of zero):

- Login,
- LoginResponse,
- Logout,
- LogoutResponse,
- ConnectionClose,
- Heartbeat,
- Reject.

Message types other than those listed above belong to the business message set and require setting a sequence number.

6.2.2. HEARTBEATS

The heartbeat message used by Gateway and Client Application to provide information about the communication line during time gaps of inactivity and to verify that interfaces at each end are accessible. The length of heartbeat interval is 4 seconds. The heartbeat send time (i.e. interval) is counted from the time of the last sent message.

The Logout procedure and closing the TCP/IP connection is done once the server noticed an inactivity for 3 heartbeat intervals.

6.2.3. REJECTIONS

Client Application message is rejected at the session level by sending a "Reject" message. The reason for rejecting is presented in the "reason" field and includes:

- exceeding throughput limits (MaxThroughputExceeded),
- sending the wrong type of a message (InvalidMsgType),
- wrong expire field format (InvalidExpireTimePrecision).
- invalid settlement date in trade capture report (InvalidSettlementDate).

In the case of communication protocol errors, instead of rejecting the message, the connection is terminated (see next section).

6.3. TERMINATING A CONNECTION

Client Application should terminate all used connections at the end of trading day by sending a Logout message. Then Gateway service closes TCP/IP connection.

If Client Application does not terminate a connection, Gateway does it automatically once it shuts down (a ConnectionClose message is sent).

During the trading day, in extraordinary cases, Gateway may initiate the termination of connection by sending the ConnectionClose message.

7. MESSAGES

Name	ID	Length	Description
BidOfferUpdate	31	45	Message send during the IPO to the sell side or during the Tender Offer to the buy side.
ConnectionClose	12	17	The ConnectionClose message confirms the termination of a session through the trading port service.
GapFill	34	16	Create a SeqNum gap between current SeqNum and header::seqNum
Heartbeat	13	16	Heartbeat message
Login	2	36	The login message authenticates a user establishing a connection to the trading port service. The login message must be the first message sent by the client application to request the initiation of a trading port session.
LoginResponse	3	27	The login response message. The result field describes the login status, indicating whether the login was successful or not (i.e. successful login).
Logout	11	16	The logout message is sent from the client application to terminate the communication session with the trading port.
LogoutResponse	14	16	The logout response message confirms the client logout message.
MarketMakerCommand	32	21	Market Maker command request.
MarketMakerCommandResponse	33	23	The response to the Market Maker command.
MassQuote	24	1200	Mass Quote
MassQuoteResponse	25	719	The response to a MassQuote message.
OrderAdd	4	167	Message used to add new orders to System.
OrderAddResponse	5	52	The message is a response to an OrderAdd message and includes the order execution status.
OrderCancel	6	40	Message used to cancel the previously submitted order.
OrderCancelResponse	7	28	The message is a response to an order cancel request and contains information about its execution, in particular whether the order to cancel was found or not.
OrderMassCancel	29	35	Message used to cancel multiple existing orders.
OrderMassCancelResponse	30	44	Response to message used to cancel multiple existing orders.
OrderModify	8	80	Message used to modify the submitted order.
OrderModifyResponse	9	36	The response message for an OrderModify.

Name	ID	Length	Description
Reject	15	21	The reject message is sent by the trading port service when receiving an erroneous message that cannot be further processed.
RequestForExecution	28	21	The information for the MM that one of the quotes has been crossed.
Trade	10	52	The message used to report trades between counterparties (i.e. generated when two or more orders are matched).
TradeBust	23	20	Message used to inform about cancellation of previously accepted Trade
TradeCaptureReportDual	19	271	Trade Capture Report - dual sided.
TradeCaptureReportResponse	20	56	The message is a response to an Trade Capture Report message, containing the state of TCR execution.
TradeCaptureReportSingle	18	206	Trade Capture Report - single side.

7.1. HEADER

Name	Type	Kind	Length	Description
length	MsgLength	Alias (u16)	2	Total length of the message.
msgType	MsgType	Enum	2	Type of the message (e.g. Login).
seqNum	SeqNum	Alias (u32)	4	Sequence number of the message added by the sender.
timestamp	Timestamp	Alias (u64)	8	Sending time.

7.2. BIDOFFERUPDATE

Name	Data type	Kind	Length	Description		
header	Header	Struct	16	Message header.		
instrumentId	ElementId	Alias (u32)	4	Instrument ID.		
updateType	BidOfferUpdateType	Enum	1	Indicates a type of the BidOfferUpdate message.		
				Name	Value	Description
				Ipo	1	IPO.
				TenderOffer	2	Tender Offer.

Name	Data type	Kind	Length	Description
totalBidSize	Quantity	Alias (u64)	8	Specifies the total bid size (number of Lots).
totalOfferSize	Quantity	Alias (u64)	8	Specifies the total offer size (number of Lots).
bidOrders	u32	Primitive	4	Number of bid orders.
offerOrders	u32	Primitive	4	Number of offer orders.

7.3. CONNECTIONCLOSE

Name	Type	Kind	Length	Description		
header	Header	Struct	16	Header.		
reason	ConnectionCloseReason	Enum	1	Connection close reason.		
				Name	Value	Description
				ProtocolError	1	Invalid message or frame length.
				InvalidSeqNum	2	Message came with an incorrect sequence number.
				EndOfDay	3	The session day has come to an end.
				SyncFail	4	The synchronization of messages has failed.
				AntiFloodingThresholdExceeded	5	The second level of the throttling limit has been exceeded.
				ConnectionConfigChanged	6	Connection configuration has changed
				CloseOps	7	Service closed the connection because of an exchange operation (e.g. supervision operation).
				Disconnect	8	Service closed the connection because of an internal operation (e.g. grace period end).

7.4. GAPFILL

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.

7.5. HEARTBEAT

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.

7.6. LOGIN

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.
version	MsgVersion	Alias (u16)	2	Indicates the version of the protocol in which the message is defined.
token	Token	Array (AnsiChar)	8	The security data required for authentication, which is a token received by Exchange Member during the registration process.
connectionId	ConnectionId	Alias (u16)	2	ID of the connection.
nextExpectedSeqNum	SeqNum	Alias (u32)	4	Next expected message sequence number value to be received.
lastSentSeqNum	SeqNum	Alias (u32)	4	Last sent sequence number.

7.7. LOGINRESPONSE

Name	Type	Kind	Length	Description		
header	Header	Struct	16	Header.		
result	LoginResult	Enum	1	Login response status code.		
				Name	Value	Description
				Ok	1	Successful login.
				NotFound	2	User not found.
				InvalidToken	3	Authorization failure.
				AlreadyLoggedIn	4	Already logged in.
				AccountLocked	5	Account locked.

Name	Type	Kind	Length	Description		
				LoginNotAllowed	6	Login is currently unavailable due to reasons such as service unavailability.
				InvalidLoginParameters	7	The login parameters, such as the connection ID, are invalid.
				ThrottlingTemporaryLock	8	The login attempt failed due to exceeding the anti-flooding threshold.
				Other	9	Other errors.
nextExpectedSeqNum	SeqNum	Alias (u32)	4	Next expected message sequence number value to be received.		
lastReplaySeqNum	SeqNum	Alias (u32)	4	Last replay sequence number.		
sessionId	SessionId	Alias (u16)	2	ID of the session.		

7.8. LOGOUT

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.

7.9. LOGOUTRESPONSE

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.

7.10. MARKETMAKERCOMMAND

Name	Data type	Kind	Length	Description
header	Header	Struct	16	Message header.
instrumentId	ElementId	Alias (u32)	4	Instrument ID.
action	CommandAction	Enum	1	The action for the Market Maker command request.

Name	Data type	Kind	Length	Description		
				Name	Value	Description
				ChangeToHybridBuyOnly	1	Transition of the instrument from the Hybrid phase to the Hybrid Buy Only phase.
				ChangeToKnockOut	2	Change of the instrument's phase to KnockOut.
				RevokeKnockOut	3	Cancellation of the KnockOut phase. New orders can be placed again

7.11.MARKETMAKERCOMMANDRESPONSE

Name	Data type	Kind	Length	Description		
header	Header	Struct	16	Message header.		
refId	u32	Primitive	4	A reference id to the Market Maker command request message.		
result	CommandResult	Enum	1	Confirmation of achieving the intended action. If the action (e.g., Buy Only state) has already been achieved before the command, the result will be positive.		
				Name	Value	Description
				Success	1	
				Failure	2	
rejectionCode	CommandRejectionCode	Enum	2	Please, see GPW WATS 2.03 Rejection Codes for a full list of rejection codes.		

7.12. MASSQUOTE

Name	Type	Kind	Length	Description
header	Header	Struct	16	Message header.
stpid	STPID	Alias (u16)	1	An ID assigned by the client used in Self Match Prevention mechanism.
capacity	Capacity	Enum	1	Capacity of the party making the order (either principal or agency).

Name	Type	Kind	Length	Description															
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>Agency</td><td>1</td><td>Agency (mapped to AOTC).</td></tr><tr><td>Principal</td><td>2</td><td>Principal (mapped to DEAL).</td></tr><tr><td>RisklessPrincipal</td><td>3</td><td>Riskless Principal (mapped to MTCH)</td></tr></table>	Name	Value	Description	Agency	1	Agency (mapped to AOTC).	Principal	2	Principal (mapped to DEAL).	RisklessPrincipal	3	Riskless Principal (mapped to MTCH)			
				Name	Value	Description													
				Agency	1	Agency (mapped to AOTC).													
				Principal	2	Principal (mapped to DEAL).													
RisklessPrincipal	3	Riskless Principal (mapped to MTCH)																	
account	Account	Array (AnsiChar)	16	Account mnemonic as agreed between buy and sell sides.															
accountType	AccountType	Enum	1	Type of account associated with the order. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>Missing</td><td>1</td><td>Account is missing. Account is expected to be filled with 0x00.</td></tr><tr><td>Customer</td><td>2</td><td>Account is carried on customer side of the books.</td></tr><tr><td>House</td><td>3</td><td>House trader.</td></tr></table>	Name	Value	Description	Missing	1	Account is missing. Account is expected to be filled with 0x00.	Customer	2	Account is carried on customer side of the books.	House	3	House trader.			
				Name	Value	Description													
				Missing	1	Account is missing. Account is expected to be filled with 0x00.													
				Customer	2	Account is carried on customer side of the books.													
House	3	House trader.																	
mifidFields.flags	MifidFlags	Enum	1	Flags raised on an order in compliance with the MiFID directive. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>None</td><td>0b0000</td><td></td></tr><tr><td>LiquidityProvisionActivity</td><td>0b0001</td><td>Order submitted by any firm providing liquidity to the market.</td></tr><tr><td>DirectOrSponsoredAccess</td><td>0b0010</td><td>Order submitted by Sponsored client using Sponsored Access connectivity.</td></tr><tr><td>MarketMakerOrSpecialist</td><td>0b0100</td><td>Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.</td></tr></table>	Name	Value	Description	None	0b0000		LiquidityProvisionActivity	0b0001	Order submitted by any firm providing liquidity to the market.	DirectOrSponsoredAccess	0b0010	Order submitted by Sponsored client using Sponsored Access connectivity.	MarketMakerOrSpecialist	0b0100	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.
				Name	Value	Description													
				None	0b0000														
				LiquidityProvisionActivity	0b0001	Order submitted by any firm providing liquidity to the market.													
DirectOrSponsoredAccess	0b0010	Order submitted by Sponsored client using Sponsored Access connectivity.																	
MarketMakerOrSpecialist	0b0100	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.																	
mifidFields.client.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.															
mifidFields.client.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>NA</td><td>1</td><td>Not Applicable.</td></tr><tr><td>Algorithm</td><td>2</td><td>Short code of the algorithm used when submitting an order.</td></tr><tr><td>FirmOrLegalEntity</td><td>3</td><td>Qualifier if an order is submitted by a client being legal entity.</td></tr><tr><td>NaturalPerson</td><td>4</td><td>Qualifier if an order is submitted by a client being natural person</td></tr></table>	Name	Value	Description	NA	1	Not Applicable.	Algorithm	2	Short code of the algorithm used when submitting an order.	FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity.	NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
				Name	Value	Description													
				NA	1	Not Applicable.													
				Algorithm	2	Short code of the algorithm used when submitting an order.													
FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity.																	
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person																	
mifidFields.executingTrader.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.															
mifidFields.executingTrader.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.															

Name	Type	Kind	Length	Description					
				Name		Value	Description		
				NA		1	Not Applicable.		
				Algorithm		2	Short code of the algorithm used when submitting an order.		
				FirmOrLegalEntity		3	Qualifier if an order is submitted by a client being legal entity.		
				NaturalPerson		4	Qualifier if an order is submitted by a client being natural person		
mifidFields.investmentDecisionMaker.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.					
mifidFields.investmentDecisionMaker.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.					
				Name		Value	Description		
				NA		1	Not Applicable.		
				Algorithm		2	Short code of the algorithm used when submitting an order.		
				FirmOrLegalEntity		3	Qualifier if an order is submitted by a client being legal entity		
NaturalPerson		4	Qualifier if an order is submitted by a client being natural person						
memo	Memo	Array (AnsiChar)	18	Free text.					
clearingMemberCode	ClearingCode	Array (AnsiChar)	20	Clearing member code.					
clearingMemberClearingIdentifier	ClearingIdentifier	Enum	1	Clearing member's clearing identifier.					
				Name		Value	Description		
				NotApplicable		33	Not Applicable.		
				Lei		78	Legal Entity Identifier.		
				Custom		68	Custom clearing identifier.		
quotes.count	u8	Primitive	1	The number of quotes this message contains.					
quotes.items	QuotesArray	Array (Quote)	1080	The array of quotes. Quote:					
				Name		Type	Kind	Length	Description
				instrumentId		ElementId	Alias (u32)	4	ID of the instrument being traded.
				bid.price		Price	Alias (Number)	8	Indicates the price of the given order.

Name	Type	Kind	Length	Description				
				bid.quantity	Quantity	Alias (u64)	8	Indicates the quantity of the instrument included in the order (number of Lots).
				ask.price	Price	Alias (Number)	8	Indicates the price of the given order.
				ask.quantity	Quantity	Alias (u64)	8	Indicates the quantity of the instrument included in the order (number of Lots).
feeStructureId	u8	Primitive	1	Optional identifier of a fee scheme for billing purposes. 1 to 99 - for internal purposes for Members 100 to 255 - for internal purposes for WSE (GPW)				
interestedParty	InterestedParty	Array (AnsiChar)	8	3rd party interested in this order or trade.				
quoteId	ClientOrderId	Array (AnsiChar)	20	Arbitrary user provided value associated with the mass quote.				

7.13. MASSQUOTE RESPONSE

Name	Type	Kind	Length	Description				
header	Header	Struct	16	Message header.				
massQuoteId	OrderId	Alias (u64)	8	Quote id.				
responses.count	u8	Primitive	1	How many responses this message contains.				
responses.items	QuoteOrderResponseArray	Array (QuoteOrderResponse)	690	The array of responses. QuoteOrderResponse:				
				Name	Data type	Kind	Length	Description
				instrumentId	ElementId	Alias (u32)	4	ID of the instrument being traded.
				bidOrderId	OrderId	Alias (u64)	8	Unique for each trading day order identifier based on the sequence number of order message, bulk sequence number, session ID and connection ID.
				askOrderId	OrderId	Alias (u64)	8	Unique for each trading day order identifier based on the sequence number of order

Name	Type	Kind	Length	Description						
								message, bulk sequence number, session ID and connection ID.		
				status	OrderStatus	Enum	1	Status of the given order.		
								Name	Value	Description
								New	1	New order.
								Cancelled	2	Order canceled.
								Rejected	3	Order rejected.
								Filled	4	Order filled.
Partially filled	5	Order partially filled.								
Expired	6	Order expired.								
reason	OrderRejectionReason	Enum	2	Reason for rejecting the given order. Please, see GPW WATS 2.03 Rejection Codes for a full list of rejection codes.						
status	MassQuoteStatus	Enum	1	Status of the given mass quote order.						
				Name			Value		Description	
				Accepted			1		Mass quote acknowledged by system.	
				Rejected			2		Mass quote rejected.	
reason	MassQuoteRejectionReason	Enum	2	Reason for rejecting the given mass quote order. Please, see GPW WATS 2.03 Rejection Codes for a full list of rejection codes.						
feeStructureId	u8	Primitive	1	Optional identifier of a fee scheme for billing purposes. 1 to 99 - for internal purposes for Members 100 to 255 - for internal purposes for WSE (GPW)						

7.14. ORDERADD

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.

Name	Type	Kind	Length	Description		
stpId	STPId	Alias (u16)	1	ID assigned by the client used in the Self Trade Prevention mechanism.		
instrumentId	ElementId	Alias (u32)	4	ID of the instrument being traded.		
orderType	OrderType	Enum	1	Indicates the order type.		
				Name	Value	Description
				Limit	1	Limit order type.
				Market	2	Market order type.
				MarketToLimit	3	Market to limit order type.
				Iceberg	4	Iceberg order type.
				StopLimit	5	Stop limit order type.
				StopLoss	6	Stop loss order type.
timeInForce	TimeInForce	Enum	1	Indicates the order's time in force (e.g. GTC).		
				Name	Value	Description
				Day	1	A day order is valid until the end of the trading day.
				GTC	2	A GTC order is good till canceled.
				IOC	3	An Immediate or Cancel order must be filled immediately or canceled.
				FOK	4	A Fill or Kill order must be immediately fully filled or canceled.
				VFA	5	Valid For Auction.
				GTD	6	A Good Till Date order must be filled before timestamp provided in 'Expire' field or canceled.
				VFC	7	Valid For Closing.
GTT	8	A Good Till Time order must be filled before timestamp provided in 'Expire' field or canceled within the day of submission.				
side	OrderSide	Enum	1	Indicates the order's side (buy or sell).		
				Name	Value	Description
				Buy	1	Indicates a buy-side order.
				Sell	2	Indicates a sell-side order.
price	Price	Alias (Number)	8	Indicates the price per unit (smallest portion) of instrument.		
triggerPrice	Price	Alias (Number)	8	Indicates the trigger price (Last Trade Price - LTP) after which the order should be added to the order book (per unit(smallest portion) of instrument).		

Name	Type	Kind	Length	Description																
quantity	Quantity	Alias (u64)	8	Indicates the quantity of the instrument included in the order (number of Lots).																
displayQty	Quantity	Alias (u64)	8	Used only for iceberg order. The quantity to be displayed (number of Lots).																
capacity	Capacity	Enum	1	Capacity of the party making the order (either principal or agency).																
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>Agency</td><td>1</td><td>Agency (mapped to AOTC).</td></tr><tr><td>Principal</td><td>2</td><td>Principal (mapped to DEAL).</td></tr><tr><td>RisklessPrincipal</td><td>3</td><td>Riskless Principal (mapped to MTCH)</td></tr></table>		Name	Value	Description	Agency	1	Agency (mapped to AOTC).	Principal	2	Principal (mapped to DEAL).	RisklessPrincipal	3	Riskless Principal (mapped to MTCH)			
				Name	Value	Description														
				Agency	1	Agency (mapped to AOTC).														
				Principal	2	Principal (mapped to DEAL).														
RisklessPrincipal	3	Riskless Principal (mapped to MTCH)																		
account	Account	Array (AnsiChar)	16	Account number.																
accountType	AccountType	Enum	1	Type of account associated with the order.																
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>Missing</td><td>1</td><td>Account is missing. Account is expected to be filled with 0x00.</td></tr><tr><td>Customer</td><td>2</td><td>Account is carried on customer side of the books.</td></tr><tr><td>House</td><td>3</td><td>House trader.</td></tr></table>		Name	Value	Description	Missing	1	Account is missing. Account is expected to be filled with 0x00.	Customer	2	Account is carried on customer side of the books.	House	3	House trader.			
				Name	Value	Description														
				Missing	1	Account is missing. Account is expected to be filled with 0x00.														
				Customer	2	Account is carried on customer side of the books.														
House	3	House trader.																		
mifidFields.flags	MifidFlags	Enum	1	Flags raised on an order in compliance with the MiFID directive.																
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>None</td><td>0b0000</td><td></td></tr><tr><td>LiquidityProvisionActivity</td><td>0b0001</td><td>Order submitted by any firm providing liquidity to the market.</td></tr><tr><td>DirectOrSponsoredAccess</td><td>0b0010</td><td>Order submitted by Sponsored client using Sponsored Access connectivity.</td></tr><tr><td>MarketMakerOrSpecialist</td><td>0b1000</td><td>Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.</td></tr></table>		Name	Value	Description	None	0b0000		LiquidityProvisionActivity	0b0001	Order submitted by any firm providing liquidity to the market.	DirectOrSponsoredAccess	0b0010	Order submitted by Sponsored client using Sponsored Access connectivity.	MarketMakerOrSpecialist	0b1000	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.
				Name	Value	Description														
				None	0b0000															
				LiquidityProvisionActivity	0b0001	Order submitted by any firm providing liquidity to the market.														
				DirectOrSponsoredAccess	0b0010	Order submitted by Sponsored client using Sponsored Access connectivity.														
MarketMakerOrSpecialist	0b1000	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.																		
mifidFields.client.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.																
mifidFields.client.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.																
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>NA</td><td>1</td><td>Not Applicable.</td></tr><tr><td>Algorithm</td><td>2</td><td>Short code of the algorithm used when submitting an order.</td></tr><tr><td>FirmOrLegalEntity</td><td>3</td><td>Qualifier if an order is submitted by a client being legal entity</td></tr><tr><td>NaturalPerson</td><td>4</td><td>Qualifier if an order is submitted by a client being natural person</td></tr></table>		Name	Value	Description	NA	1	Not Applicable.	Algorithm	2	Short code of the algorithm used when submitting an order.	FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity	NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
				Name	Value	Description														
				NA	1	Not Applicable.														
				Algorithm	2	Short code of the algorithm used when submitting an order.														
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity														
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person																		

Name	Type	Kind	Length	Description		
mifidFields.executingTrader.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
mifidFields.executingTrader.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person				
mifidFields.investmentDecisionMaker.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
mifidFields.investmentDecisionMaker.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person				
expire	Timestamp	Alias (u64)	8	Expiration time indicating the validity of the order - relevant only when timeInForce is set to GTD (Good Till Date) or GTT (Good Till Time).		
memo	Memo	Array (AnsiChar)	18	Free text.		
clientOrderId	ClientOrderId	Array (AnsiChar)	20	Arbitrary user provided value associated with the order.		
clearingMemberCode	ClearingCode	Array (AnsiChar)	20	Clearing member code.		
clearingMemberClearingIdentifier	ClearingIdentifier	Enum	1	Clearing member's clearing identifier.		
				Name	Value	Description
				NotApplicable	33	Not Applicable.
				Lei	78	Legal Entity Identifier.
				Custom	68	Custom clearing identifier.
execInst	ExecInst	Enum	1	Instructions for order handling on exchange trading floor.		
				Name	Value	Description
				None	0b00000000	

Name	Type	Kind	Length	Description		
				CancelOnConnectionLoss	ob00000001	Cancel on connection loss
feeStructureId	u8	Primitive	1	Optional identifier of a fee scheme for billing purposes. 1 to 99 - for internal purposes for Members 100 to 255 - for internal purposes for WSE (GPW)		
interestedParty	InterestedParty	Array (AnsiChar)	8	3rd party interested in this order or trade.		

7.15. ORDERADDRESPONSE

Name	Type	Kind	Length	Description		
header	Header	Struct	16	Header.		
orderId	OrderId	Alias (u64)	8	Unique for each trading day order identifier based on the sequence number of order message, bulk sequence number, session ID and connection ID.		
publicOrderId	PublicOrderId	Alias (u64)	8	Unique for single trading day order identifier assigned by the trading system and shared publicly in market data (public information).		
displayQty	Quantity	Alias (u64)	8	Used only for iceberg order. The quantity to be displayed (number of Lots).		
filled	Quantity	Alias (u64)	8	Indicates the quantity of the order which has already been filled (number of Lots).		
status	OrderStatus	Enum	1	Status of the given order.		
				Name	Value	Description
				New	1	New order.
				Cancelled	2	Order canceled.
				Rejected	3	Order rejected.
				Filled	4	Order filled.
				Partially filled	5	Order partially filled.
				Expired	6	Order expired.

Name	Type	Kind	Length	Description		
reason	OrderRejectionReason	Enum	2	Reason for rejecting the given order. Please, see GPW WATS 2.03 Rejection Codes for a full list of rejection codes.		
execTypeReason	ExecTypeReason	Enum	1	Describes why an order was executed and the events related to its lifecycle.		
				Name	Value	Description
				NA	1	Not applicable.
				CancelOnDisconnect	2	Order cancelled by the canceller on client disconnect.
				Expired	3	Order cancelled by the canceller due to expiration.
				Triggered	4	Notification of Stop or VFA/VFC order activation.
				CancelOnSuspension	5	Order cancelled by canceller due to instrument suspension.
				OrderRestatement	6	Order reinstated.
				IcebergOrderRefill	7	Iceberg order refill.
				CancelByStp	8	Order cancelled due to self-trade prevention.
				CancelByCorporateAction	9	Order cancelled due to submitted Corporate Action.
				CancelByMassCancel	10	Order cancelled due to mass cancel
				CancellocFokOrder	11	Order cancelled due to IOC/FOK.
				CancelByMarketOperations	12	Order cancelled due to market operations.
				Replaced	13	Order replaced.
				FirstTradeOnAggressiveOrder	14	First trade on aggressive order.
				Rejected	15	Order rejected.
				CancelonBuyOnlyStateEntry	16	Cancel on BuyOnly state entry
				CancelonKnockedOutStateEntry	17	Cancel on KnockedOut state entry
CancelByRiskManagement	18	Order cancelled due to exceeding RMA post trade limit.				

7.16. ORDERCANCEL

Name	Type	Kind	Length	Description															
header	Header	Struct	16	Header.															
orderId	OrderId	Alias (u64)	8	Unique for each trading day order identifier based on the sequence number of order message, bulk sequence number, session ID and connection ID.															
mifidFields.flags	MifidFlags	Enum	1	Flags raised on an order in compliance with the MiFID directive.															
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>None</td><td>0b0000</td><td></td></tr><tr><td>LiquidityProvisionActivity</td><td>0b0001</td><td>Order submitted by any firm providing liquidity to the market.</td></tr><tr><td>DirectOrSponsoredAccess</td><td>0b0010</td><td>Order submitted by Sponsored client using Sponsored Access connectivity.</td></tr><tr><td>MarketMakerOrSpecialist</td><td>0b0100</td><td>Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.</td></tr></table>	Name	Value	Description	None	0b0000		LiquidityProvisionActivity	0b0001	Order submitted by any firm providing liquidity to the market.	DirectOrSponsoredAccess	0b0010	Order submitted by Sponsored client using Sponsored Access connectivity.	MarketMakerOrSpecialist	0b0100	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.
				Name	Value	Description													
				None	0b0000														
				LiquidityProvisionActivity	0b0001	Order submitted by any firm providing liquidity to the market.													
				DirectOrSponsoredAccess	0b0010	Order submitted by Sponsored client using Sponsored Access connectivity.													
MarketMakerOrSpecialist	0b0100	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.																	
mifidFields.client.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.															
mifidFields.client.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.															
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>NA</td><td>1</td><td>Not Applicable.</td></tr><tr><td>Algorithm</td><td>2</td><td>Short code of the algorithm used when submitting an order.</td></tr><tr><td>FirmOrLegalEntity</td><td>3</td><td>Qualifier if an order is submitted by a client being legal entity</td></tr><tr><td>NaturalPerson</td><td>4</td><td>Qualifier if an order is submitted by a client being natural person</td></tr></table>	Name	Value	Description	NA	1	Not Applicable.	Algorithm	2	Short code of the algorithm used when submitting an order.	FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity	NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
				Name	Value	Description													
				NA	1	Not Applicable.													
				Algorithm	2	Short code of the algorithm used when submitting an order.													
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity													
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person																	
mifidFields.executingTrader.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.															
mifidFields.executingTrader.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.															
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>NA</td><td>1</td><td>Not Applicable.</td></tr><tr><td>Algorithm</td><td>2</td><td>Short code of the algorithm used when submitting an order.</td></tr></table>	Name	Value	Description	NA	1	Not Applicable.	Algorithm	2	Short code of the algorithm used when submitting an order.						
				Name	Value	Description													
				NA	1	Not Applicable.													
Algorithm	2	Short code of the algorithm used when submitting an order.																	

Name	Type	Kind	Length	Description		
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
				NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
mifidFields.investmentDecisionMaker.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
mifidFields.investmentDecisionMaker.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
				NaturalPerson	4	Qualifier if an order is submitted by a client being natural person

7.17. ORDERCANCELRESPONSE

Name	Type	Kind	Length	Description		
header	Header	Struct	16	Header.		
orderId	OrderId	Alias (u64)	8	Unique for each trading day order identifier based on the sequence number of order message, bulk sequence number, session ID and connection ID.		
status	OrderStatus	Enum	1	Status of the given order.		
				Name	Value	Description
				New	1	New order.
				Cancelled	2	Order canceled.
				Rejected	3	Order rejected.
				Filled	4	Order filled.
				PartiallyFilled	5	Order partially filled.
				Expired	6	Order expired.
reason	OrderRejectionReason	Enum	2	Reason for rejecting the given order. Please, see GPW WATS 2.03 Rejection Codes for a full list of rejection codes.		

Name	Type	Kind	Length	Description		
execTypeReason	ExecTypeReason	Enum	1	Describes why an order was executed and the events related to its lifecycle.		
				Name	Value	Description
				NA	1	Not applicable.
				CancelOnDisconnect	2	Order cancelled by the canceller on client disconnect.
				Expired	3	Order cancelled by the canceller due to expiration.
				Triggered	4	Notification of Stop or VFA/VFC order activation.
				CancelOnSuspension	5	Order cancelled by canceller due to instrument suspension.
				OrderRestatement	6	Order reinstated.
				IcebergOrderRefill	7	Iceberg order refill.
				CancelByStp	8	Order cancelled due to self-trade prevention.
				CancelByCorporateAction	9	Order cancelled due to submitted Corporate Action.
				CancelByMassCancel	10	Order cancelled due to mass cancel
				CancellocFokOrder	11	Order cancelled due to IOC/FOK.
				CancelByMarketOperations	12	Order cancelled due to market operations.
				Replaced	13	Order replaced.
				FirstTradeOnAggressiveOrder	14	First trade on aggressive order.
				Rejected	15	Order rejected.
				CancelonBuyOnlyStateEntry	16	Cancel on BuyOnly state entry
				CancelonKnockedOutStateEntry	17	Cancel on KnockedOut state entry
				CancelByRiskManagement	18	Order cancelled due to exceeding RMA post trade limit.

7.18. ORDERMASSCANCEL

Name	Type	Kind	Length	Description			
header	Header	Struct	16	Header.			
massCancelRequestType	MassCancelRequestType	Enum	1	Mass cancel request type.			
				Name		Value	Description
				CancelForSecurity		1	Cancel orders for a security.
				CancelAllOrders		7	Cancel all orders.
				CancelOrdersForMarketSegment		9	Cancel orders for a market segment.
targetPartyRole	TargetPartyRole	Enum	1	Target party role filter selection field.			
				Name		Value	Description
				None		0	No filtering.
				LiquidityProvider		35	Liquidity provider.
				SenderLocation		54	Sender location.
				MarketMaker		66	Market maker.
				ClientID		3	Client ID.
				ExecutingTrader		12	Executing trader.
				InvestmentDecisionMaker		122	Investment decision maker.
targetPartyId	u32	Primitive	4	Used to identify the party targeted for the action specified in the message.			
marketSegmentId	ElementId	Alias (u32)	4	Identifies Market segment for request type CancelOrdersForMarketSegment.			
instrumentId	ElementId	Alias (u32)	4				
executingTrader.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.			
executingTrader.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.			
				Name	Value	Description	

Name	Type	Kind	Length	Description		
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
				NaturalPerson	4	Qualifier if an order is submitted by a client being natural person

7.19. ORDERMASSCANCELRESPONSE

Name	Type	Kind	Length	Description		
header	Header	Struct	16	Header.		
totalAffectedOrders	u64	Primitive	8	Total number of orders affected by the mass cancel request.		
massCancelRequestType	MassCancelRequestType	Enum	1	Mass cancel request type.		
				Name	Value	Description
				CancelForSecurity	1	Cancel orders for a security.
				CancelAllOrders	7	Cancel all orders.
massCancelId	OrderId	Alias (u64)	8	Mass cancel request ID.		
targetPartyRole	TargetPartyRole	Enum	1	Target party role filter selection field.		
				Name	Value	Description
				None	0	No filtering.
				LiquidityProvider	35	Liquidity provider.
				SenderLocation	54	Sender location.
				MarketMaker	66	Market maker.
				ClientID	3	Client ID.
				ExecutingTrader	12	Executing trader.

Name	Type	Kind	Length	Description		
				InvestmentDecisionMaker	122	Investment decision maker.
targetPartyId	u32	Primitive	4	Used to identify the party targeted for the action specified in the message.		
marketSegmentId	ElementId	Alias (u32)	4	Identifies Market segment for request type CancelOrdersForMarketSegment.		
reason	OrderRejectionReason	Enum	2	Reason for rejecting the given mass order cancel request. Please, see GPW WATS 2.03 Rejection Codes for a full list of rejection codes.		

7.20. ORDERMODIFY

	Type	Kind	Length	Description		
header	Header	Struct	16	Header.		
orderId	OrderId	Alias (u64)	8	Unique for each trading day order identifier based on the sequence number of order message, bulk sequence number, session ID and connection ID.		
price	Price	Alias (Number)	8	Indicates the price per unit (smallest portion) of instrument		
triggerPrice	Price	Alias (Number)	8	Indicates the trigger price (Last Trade Price - LTP) after which the order should be added to the order book (per unit (smallest portion) of instrument).		
quantity	Quantity	Alias (u64)	8	Indicates the quantity of the instrument included in the order (number of Lots).		
displayQty	Quantity	Alias (u64)	8	Used only for iceberg order. The quantity to be displayed (number of Lots).		
expire	Timestamp	Alias (u64)	8	Expiration time indicating the validity of the order - relevant only when timeInForce is set to GTD (Good Till Date).		
mifidFields.flags	MifidFlags	Enum	1	Flags raised on an order in compliance with the MIFID directive.		
				Name	Value	Description
				None	ob0000	
				LiquidityProvisionActivity	ob0001	Order submitted by any firm providing liquidity to the market.
				DirectOrSponsoredAccess	ob0010	Order submitted by Sponsored client using Sponsored Access connectivity.

	Type	Kind	Length	Description		
				MarketMakerOrSpecialist	ob0100	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.
mifidFields.client.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
mifidFields.client.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person				
mifidFields.executingTrader.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
mifidFields.executingTrader.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person				
mifidFields.investmentDecisionMaker.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
mifidFields.investmentDecisionMaker.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person				

7.21. ORDERMODIFYRESPONSE

Name	Type	Kind	Length	Description		
header	Header	Struct	16	Header.		
orderId	OrderId	Alias (u64)	8	Unique for each trading day order identifier based on the sequence number of order message, bulk sequence number, session ID and connection ID.		
filled	Quantity	Alias (u64)	8	Indicates the quantity of the order which has already been filled (number of Lots).		
status	OrderStatus	Enum	1	Status of the given order.		
				Name	Value	Description
				New	1	New order.
				Cancelled	2	Order canceled.
				Rejected	3	Order rejected.
				Filled	4	Order filled.
				PartiallyFilled	5	Order partially filled.
Expired	6	Order expired.				
priorityFlag	PriorityFlag	Enum	1	Indicates whether the priority flag is lost or retained after modification.		
				Name	Value	Description
				Lost	1	The priority flag was lost.
				Retained	2	The priority flag was retained.
reason	OrderRejectionReason	Enum	2	Reason for rejecting the given order. Please, see GPW WATS 2.03 Rejection Codes for a full list of rejection codes.		

7.22. REJECT

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.

Name	Type	Kind	Length	Description		
refSeqNum	SeqNum	Alias (u32)	4	Sequence number of the rejected message.		
rejectReason	RejectReason	Enum	1	Reject reason.		
				Name	Value	Description
				NA	0	Not applicable
				MaxThroughputExceeded	1	Max throughput exceeded.
				InvalidMsgType	2	Invalid message type.
				InvalidExpireTimePrecision	3	Invalid expire time precision.
				InvalidSettlementDate	4	Invalid settlement date.
				SettlementDateRequired	5	Settlement date required.
				TradeReportIdRequired	6	Trade report ID required.
				MissingReportIdSecondaryTradeReportIdOrTradeReportRefId	7	Missing report id (SecondaryTradeReportId or TradeReportRefId).
				InvalidTradeld	8	Invalid trade ID.
				InvalidAlgorithmicTradeIndicator	9	Invalid algorithmic trade indicator.
				InvalidTradeReportId	10	Invalid Trade report ID.
				InvalidGapFillSeqNum	11	GapFill seqNum should be higher than current seqNum

7.23. REQUESTFOREXECUTION

Name	Type	Kind	Length	Description		
header	Header	Struct	16	Message header.		
instrumentId	ElementId	Alias (u32)	4	Instrument ID		
reason	RequestForExecutionReason	Enum	1	The reason for this RFE.		
				Name	Value	Description
				PassiveQuote	1	

Name	Type	Kind	Length	Description
				AggressiveQuote2

7.24. TRADE

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.
orderId	OrderId	Alias (u64)	8	Unique for each trading day order identifier based on the sequence number of order message, bulk sequence number, session ID and connection ID.
id	TradeId	Alias (u32)	4	ID of the trade.
price	Price	Alias (Number)	8	Price per unit (smallest portion) of instrument .
quantity	Quantity	Alias (u64)	8	Quantity of the instrument involved in the given trade (number of Lots).
leavesQty	Quantity	Alias (u64)	8	How much of the given security is left on Market after the trade is concluded (number of Lots).

7.25. TRADEBUST

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.
tradeId	TradeId	Alias (u32)	4	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.

7.26. TRADECAPTUREREPORTDUAL

Name	Type	Kind	Length	Description
header	Header	Struct	16	Header.
instrumentId	ElementId	Alias (u32)	4	ID of the instrument included in the order.
tradeReportId	TradeReportId	Array (AnsiChar)	21	Unique identifier of trade capture report. Last character is reserved for exchange-assigned identifiers.

Name	Type	Kind	Length	Description																					
tradeId	TradeId	Alias (u32)	4	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.																					
tradeReportTransType	TradeReportTransType	Enum	1	Identifies Trade Report message transaction type.																					
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>New</td><td>1</td><td>New.</td></tr><tr><td>Cancel</td><td>2</td><td>Cancel.</td></tr><tr><td>Replace</td><td>3</td><td>Replace.</td></tr></table>	Name	Value	Description	New	1	New.	Cancel	2	Cancel.	Replace	3	Replace.									
				Name	Value	Description																			
				New	1	New.																			
				Cancel	2	Cancel.																			
Replace	3	Replace.																							
tradeReportType	TradeReportType	Enum	1	Type of Trade Report.																					
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>Submit</td><td>1</td><td>Submit</td></tr><tr><td>Alleged</td><td>2</td><td>Alleged</td></tr><tr><td>Accept</td><td>3</td><td>Accept</td></tr><tr><td>Decline</td><td>4</td><td>Decline</td></tr><tr><td>TradeReportCancel</td><td>7</td><td>Trade Report Cancel</td></tr><tr><td>TradeBreak</td><td>8</td><td>Trade Break</td></tr></table>	Name	Value	Description	Submit	1	Submit	Alleged	2	Alleged	Accept	3	Accept	Decline	4	Decline	TradeReportCancel	7	Trade Report Cancel	TradeBreak	8	Trade Break
				Name	Value	Description																			
				Submit	1	Submit																			
Alleged	2	Alleged																							
Accept	3	Accept																							
Decline	4	Decline																							
TradeReportCancel	7	Trade Report Cancel																							
TradeBreak	8	Trade Break																							
tradeType	TradeType	Enum	1	Type of trade.																					
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>PrivatelyNegotiatedTrade</td><td>22</td><td>Privately negotiated trade.</td></tr><tr><td>BlockTrade</td><td>38</td><td>Block trade.</td></tr></table>	Name	Value	Description	PrivatelyNegotiatedTrade	22	Privately negotiated trade.	BlockTrade	38	Block trade.												
				Name	Value	Description																			
				PrivatelyNegotiatedTrade	22	Privately negotiated trade.																			
BlockTrade	38	Block trade.																							
algorithmicTradeIndicator	AlgorithmicTradeIndicator	Enum	1	Indicates algorithmic trade.																					
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>NA</td><td>1</td><td>Not applicable.</td></tr><tr><td>NonAlgorithmicTrade</td><td>1</td><td>Non-algorithmic trade.</td></tr><tr><td>AlgorithmicTrade</td><td>2</td><td>Algorithmic trade.</td></tr></table>	Name	Value	Description	NA	1	Not applicable.	NonAlgorithmicTrade	1	Non-algorithmic trade.	AlgorithmicTrade	2	Algorithmic trade.									
				Name	Value	Description																			
				NA	1	Not applicable.																			
				NonAlgorithmicTrade	1	Non-algorithmic trade.																			
AlgorithmicTrade	2	Algorithmic trade.																							
execType	ExecType	Enum	1	Type of execution being reported. Uses subset of ExecType for trade capture reports.																					

Name	Type	Kind	Length	Description		
				Name	Value	Description
				NA	1	Not applicable.
				Rejected	8	Rejected.
				Expire	12	Expire
				Trade	15	Trade.
				TradeCorrect	16	Trade Correct.
				TradeCancel	17	Trade Cancel.
tradeReportRefId	TradeReportRefID	Array (AnsiChar)	21	Reference identifier used with Cancel and Replace transaction types. The TradeReportID that is being referenced for trade correction or cancelation.		
lastQty	Quantity	Alias (u64)	8	Quantity (e.g. shares) bought/sold on this (last) fill.		
lastPx	Price	Alias (Number)	8	Price per unit(smallest portion) of instrument of this (last) fill.		
settlementDate	Date	Alias (u32)	4	Settlement date of the trade is equal to current date plus actual settlement offset calendar days.		
tcrPartyBuy.mifidFields.flags	MifidFlags	Enum	1	Flags raised on an order in compliance with the MiFID directive.		
				Name	Value	Description
				None	0b0000	
				LiquidityProvisionActivity	0b0001	Order submitted by any firm providing liquidity to the market.
				DirectOrSponsoredAccess	0b0010	Order submitted by Sponsored client using Sponsored Access connectivity.
tcrPartyBuy.mifidFields.client.shortCode	PartyRoleQualifier	Enum	1	MarketMakerOrSpecialist	0b0100	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.
				Short code of MiFID participant.		
				Qualifier of MiFID participant.		
				Name	Value	Description

Name	Type	Kind	Length	Description		
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
				NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
tcrPartyBuy.mifidFields.executingTrader.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
tcrPartyBuy.mifidFields.executingTrader.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
				NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
tcrPartyBuy.mifidFields.investmentDecisionMaker.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
tcrPartyBuy.mifidFields.investmentDecisionMaker.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
				NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
tcrPartyBuy.clearingMemberCode	ClearingCode	Array (AnsiChar)	20	Clearing member code.		
tcrPartyBuy.clearingMemberClearingIdentifier	ClearingIdentifier	Enum	1	Clearing member's clearing identifier.		
				Name	Value	Description
				NotApplicable	33	Not Applicable.

Name	Type	Kind	Length	Description		
				Lei	78	Legal Entity Identifier.
				Custom	68	Custom clearing identifier.
tcrPartyBuy.account	Account	Array (AnsiChar)	16	Account number.		
tcrPartyBuy.accountType	AccountType	Enum	1	Type of account associated with the order.		
				Name	Value	Description
				Missing	1	Account is missing. Account is expected to be filled with 0x00.
				Customer	2	Account is carried on customer side of the books.
				House	3	House trader.
tcrPartyBuy.orderCapacity	Capacity	Enum	1	Designates the capacity of the firm placing the order.		
				Name	Value	Description
				Agency	1	Agency (mapped to AOTC).
				Principal	2	Principal (mapped to DEAL).
				RisklessPrincipal	3	Riskless Principal (mapped to MTCH)
tcrPartyBuy.orderRestrictions	ElementId	Alias (u32)	4	Restrictions associated with an order.		
tcrPartyBuy.orderOrigination	ElementId	Alias (u32)	4	Identifies the origin of the order.		
tcrPartyBuy.feeStructureId	u8	Primitive	1	Optional identifier of a fee scheme for billing purposes.		
tcrPartyBuy.interestedParty	InterestedParty	Array (AnsiChar)	8	3rd party interested in this order or trade.		
tcrPartyBuy.memo	Memo	Array (AnsiChar)	18	Free text.		
tcrPartySell.mifidFields.flags	MifidFlags	Enum	1	Flags raised on an order in compliance with the MiFID directive.		
				Name	Value	Description

Name	Type	Kind	Length	Description		
				None	ob0000	Order submitted by any firm providing liquidity to the market.
				LiquidityProvisionActivity	ob0001	Order submitted by Sponsored client using Sponsored Access connectivity.
				DirectOrSponsoredAccess	ob0010	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.
				MarketMakerOrSpecialist	ob0100	Order submitted by any firm providing liquidity to the market.
tcrPartySell.mifidFields.client.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
tcrPartySell.mifidFields.client.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
				NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
tcrPartySell.mifidFields.executingTrader.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
tcrPartySell.mifidFields.executingTrader.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
				NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
tcrPartySell.mifidFields.investmentDecisionMaker.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		

Name	Type	Kind	Length	Description		
tcrPartySell.mifidFields.investmentDecisionMaker.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person				
tcrPartySell.clearingMemberCode	ClearingCode	Array (AnsiChar)	20	Clearing member code.		
tcrPartySell.clearing MemberClearingIdentifier	ClearingIdentifier	Enum	1	Clearing member's clearing identifier.		
				Name	Value	Description
				NotApplicable	33	Not Applicable.
				Lei	78	Legal Entity Identifier.
				Custom	68	Custom clearing identifier.
tcrPartySell.account	Account	Array (AnsiChar)	16	Account number.		
tcrPartySell.accountType	AccountType	Enum	1	Type of account associated with the order.		
				Name	Value	Description
				Missing	1	Account is missing. Account is expected to be filled with 0x00.
				Customer	2	Account is carried on customer side of the books.
				House	3	House trader.
tcrPartySell.orderCapacity	Capacity	Enum	1	Designates the capacity of the firm placing the order.		
				Name	Value	Description
				Agency	1	Agency (mapped to AOTC).
				Principal	2	Principal (mapped to DEAL).

Name	Type	Kind	Length	Description		
				RisklessPrincipal	3	Riskless Principal (mapped to MTCH)
tcrPartySell.orderRestrictions	ElementId	Alias (u32)	4	Restrictions associated with an order.		
tcrPartySell.orderOrigination	ElementId	Alias (u32)	4	Identifies the origin of the order.		
tcrPartySell.feeStructureId	u8	Primitive	1	Optional identifier of a fee scheme for billing purposes.		
tcrPartySell.interestedParty	InterestedParty	Array (AnsiChar)	8	3rd party interested in this order or trade.		
tcrPartySell.memo	Memo	Array (AnsiChar)	18	Free text.		

7.27. TRADECAPTUREREPORTRESPONSE

Name	Type	Kind	Length	Description		
header	Header	Struct	16	Header.		
instrumentId	ElementId	Alias (u32)	4	ID of the instrument included in the order.		
tradeId	TradeId	Alias (u32)	4	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.		
tradeReportId	TradeReportId	Array (AnsiChar)	21	Unique identifier of trade capture report. Last character is reserved for exchange-assigned identifiers.		
secondaryTradeReportId	OrderId	Alias (u64)	8	ID of the trade capture report.		
status	TcrStatus	Enum	1	Status of the given order.		
				Name	Value	Description
				New	1	New.
				Accepted	2	Accepted.
				Rejected	3	Rejected.

Name	Type	Kind	Length	Description		
				Cancelled	4	Cancelled.
reason	TcrRejectionReason	Enum	2	Reason for rejecting the given TCR. Please, see GPW WATS 2.03 Rejection Codes for a full list of rejection codes.		

7.28. TRADECAPTUREREPORTSINGLE

Name	Type	Kind	Length	Description		
header	Header	Struct	16	Header.		
instrumentId	ElementId	Alias (u32)	4	ID of the instrument included in the order.		
tradeReportId	TradeReportId	Array (AnsiChar)	21	Unique identifier of the trade capture report. Last character is reserved for exchange-assigned identifiers.		
secondaryTradeReportId	OrderId	Alias (u64)	8	ID of the trade capture report.		
tradeId	TradeId	Alias (u32)	4	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.		
tradeReportTransType	TradeReportTransType	Enum	1	Identifies Trade Report message transaction type.		
				Name	Value	Description
				New	1	New.
				Cancel	2	Cancel.
tradeReportType	TradeReportType	Enum	1	Replace	3	Replace.
				Type of Trade Report.		
				Name	Value	Description
				Submit	1	Submit
				Alleged	2	Alleged
				Accept	3	Accept
				Decline	4	Decline
				TradeReportCancel	7	Trade Report Cancel

Name	Type	Kind	Length	Description		
				TradeBreak	8	Trade Break
tradeType	TradeType	Enum	1	Type of trade.		
				Name	Value	Description
				PrivatelyNegotiatedTrade	22	Privately negotiated trade.
				BlockTrade	38	Block trade.
algorithmicTradeIndicator	AlgorithmicTradeIndicator	Enum	1	Indicates algorithmic trader.		
				Name	Value	Description
				NA	1	Not applicable.
				NonAlgorithmicTrade	2	Non-algorithmic trade.
				AlgorithmicTrade	3	Algorithmic trade.
execType	ExecType	Enum	1	Type of execution being reported. Uses subset of ExecType for trade capture reports.		
				Name	Value	Description
				NA	1	Not applicable.
				Rejected	8	Rejected.
				Expired	12	Expired
				Trade	15	Trade.
				TradeCorrect	16	Trade Correct.
				TradeCancel	17	Trade Cancel.
tradeReportRefId	TradeReportRefID	Array (AnsiChar)	21	Reference identifier used with Cancel and Replace transaction types. The TradeReportID that is being referenced for trade correction or cancelation.		
lastQty	Quantity	Alias (u64)	8	Quantity (e.g. shares) bought/sold on this (last) fill.		
lastPx	Price	Alias (Number)	8	Price per unit(smallest portion) of instrument of this (last) fill.		
settlementDate	Date	Alias (u32)	4	Settlement date of the trade is equal to current date plus actual settlement offset calendar days.		

Name	Type	Kind	Length	Description		
side	OrderSide	Enum	1	Side of order.		
				Name	Value	Description
				Buy	1	Indicates a buy-side order.
				Sell	2	Indicates a sell-side order.
counterpartyCode	CcpCode	Array (AnsiChar)	16	CCP code of the counterparty.		
tcrParty.mifidFields.flags	MifidFlags	Enum	1	Flags raised on an order in compliance with the MiFID directive.		
				Name	Value	Description
				None	0b0000	
				LiquidityProvisionActivity	0b0001	Order submitted by any firm providing liquidity to the market.
				DirectOrSponsoredAccess	0b0010	Order submitted by Sponsored client using Sponsored Access connectivity.
MarketMakerOrSpecialist	0b0100	Order submitted as part of Liquidity Provision Activity by firm which have an agreement with Market.				
tcrParty.mifidFields.client.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
tcrParty.mifidFields.client.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		
				Name	Value	Description
				NA	1	Not Applicable.
				Algorithm	2	Short code of the algorithm used when submitting an order.
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person				
tcrParty.mifidFields.executingTrader.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.		
tcrParty.mifidFields.executingTrader.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant.		

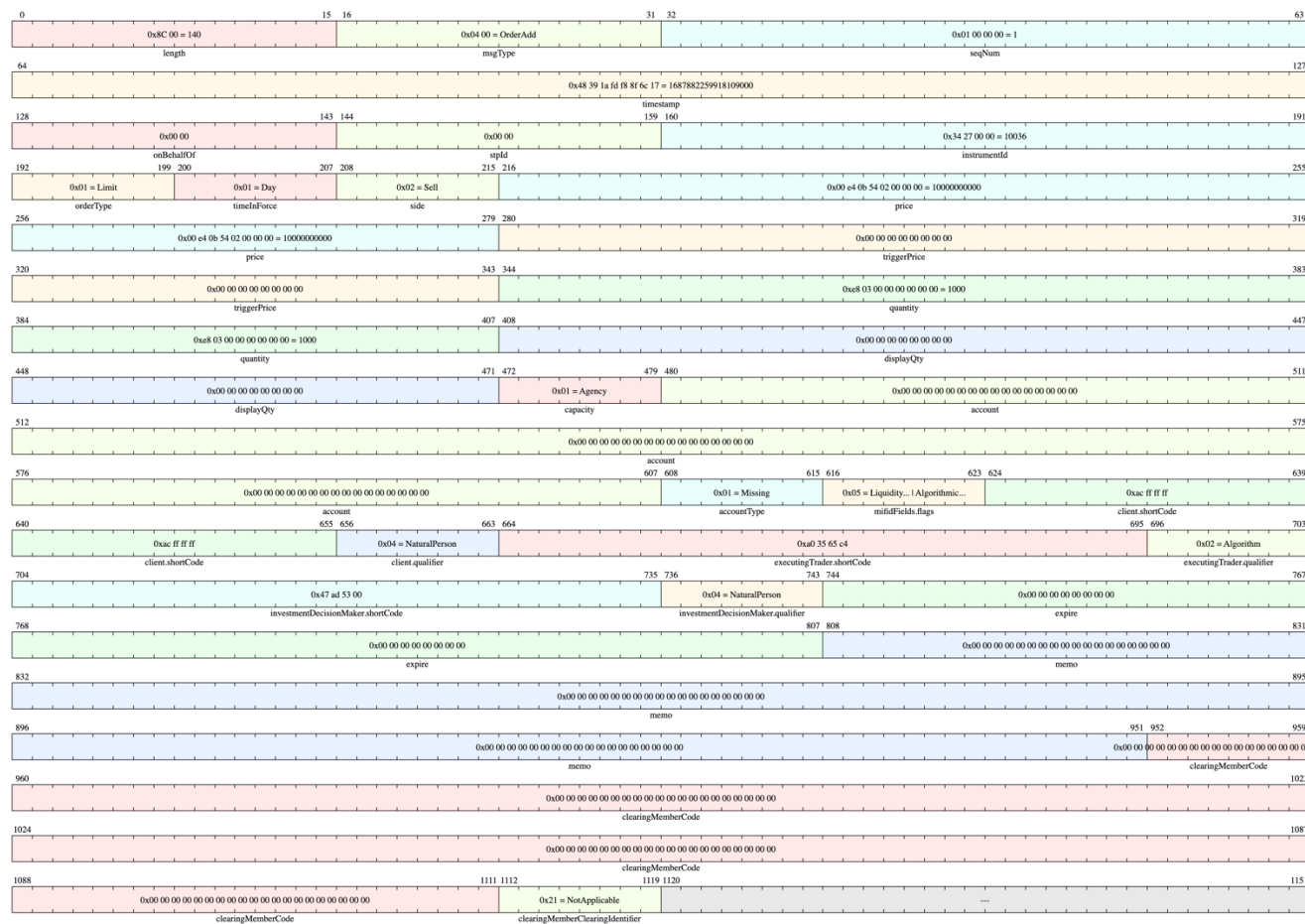
Name	Type	Kind	Length	Description															
				<table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>NA</td><td>1</td><td>Not Applicable.</td></tr><tr><td>Algorithm</td><td>2</td><td>Short code of the algorithm used when submitting an order.</td></tr><tr><td>FirmOrLegalEntity</td><td>3</td><td>Qualifier if an order is submitted by a client being legal entity</td></tr><tr><td>NaturalPerson</td><td>4</td><td>Qualifier if an order is submitted by a client being natural person</td></tr></table>	Name	Value	Description	NA	1	Not Applicable.	Algorithm	2	Short code of the algorithm used when submitting an order.	FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity	NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
				Name	Value	Description													
				NA	1	Not Applicable.													
				Algorithm	2	Short code of the algorithm used when submitting an order.													
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity													
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person																	
tcrParty.mifidFields.investmentDecisionMaker.shortCode	ShortCode	Alias (u32)	4	Short code of MiFID participant.															
tcrParty.mifidFields.investmentDecisionMaker.qualifier	PartyRoleQualifier	Enum	1	Qualifier of MiFID participant. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>NA</td><td>1</td><td>Not Applicable.</td></tr><tr><td>Algorithm</td><td>2</td><td>Short code of the algorithm used when submitting an order.</td></tr><tr><td>FirmOrLegalEntity</td><td>3</td><td>Qualifier if an order is submitted by a client being legal entity</td></tr><tr><td>NaturalPerson</td><td>4</td><td>Qualifier if an order is submitted by a client being natural person</td></tr></table>	Name	Value	Description	NA	1	Not Applicable.	Algorithm	2	Short code of the algorithm used when submitting an order.	FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity	NaturalPerson	4	Qualifier if an order is submitted by a client being natural person
				Name	Value	Description													
				NA	1	Not Applicable.													
				Algorithm	2	Short code of the algorithm used when submitting an order.													
				FirmOrLegalEntity	3	Qualifier if an order is submitted by a client being legal entity													
NaturalPerson	4	Qualifier if an order is submitted by a client being natural person																	
tcrParty.clearingMemberCode	ClearingCode	Array (AnsiChar)	20	Clearing member code.															
tcrParty.clearingMemberClearingIdentifier	ClearingIdentifier	Enum	1	Clearing member's clearing identifier. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>NotApplicable</td><td>33</td><td>Not Applicable.</td></tr><tr><td>Lei</td><td>78</td><td>Legal Entity Identifier.</td></tr><tr><td>Custom</td><td>68</td><td>Custom clearing identifier.</td></tr></table>	Name	Value	Description	NotApplicable	33	Not Applicable.	Lei	78	Legal Entity Identifier.	Custom	68	Custom clearing identifier.			
				Name	Value	Description													
				NotApplicable	33	Not Applicable.													
				Lei	78	Legal Entity Identifier.													
				Custom	68	Custom clearing identifier.													
tcrParty.account	Account	Array (AnsiChar)	16	Account number.															
tcrParty.accountType	AccountType	Enum	1	Type of account associated with the order.															

Name	Type	Kind	Length	Description		
				Name	Value	Description
				Missing	1	Account is missing. Account is expected to be filled with 0x00.
				Customer	2	Account is carried on customer side of the books.
				House	3	House trader.
tcrParty.orderCapacity	Capacity	Enum	1	Designates the capacity of the firm placing the order.		
				Name	Value	Description
				Agency	1	Agency (mapped to AOTC).
				Principal	2	Principal (mapped to DEAL).
tcrParty.orderRestrictions	ElementId	Alias (u32)	4	RisklessPrincipal	3	Riskless Principal (mapped to MTCH)
				Restrictions associated with an order.		
tcrParty.orderOrigination	ElementId	Alias (u32)	4	Identifies the origin of the order.		
tcrParty.feeStructureId	u8	Primitive	1	Optional identifier of a fee scheme for billing purposes.		
tcrParty.interestedParty	InterestedParty	Array (AnsiChar)	8	3rd party interested in this order or trade.		
tcrParty.memo	Memo	Array (AnsiChar)	18	Free text.		

7.29. ORDER ADD EXAMPLE

Below is an example of an OrderAdd message, illustrating sample values for each field along with their binary representation.

Figure 7 OrderAdd message frame



8. MESSAGE KINEMATICS

Scope:

The purpose of this appendix is to present the message flow within the Native communication protocol, as used in GPW WATS. The appendix does not include all available message flows in GPW WATS, but only those selected for presentation purposes.

Remarks:

Due to ensuring transparency of message flow diagrams, the scope of presented fields is limited to basic components and does not include the complete set of all fields that a given message may contain. The scope of the fields for the same messages may also vary depending on the specific example described in the diagram. The complete range of fields in the messages is available in the description of each individual message within the protocol.

Actors:

For the purpose of the presentations, the following actors have been utilized in the context of kinematics:

Native Trading Client Application – Exchange Member authorized to submit buy and sell orders, communicating with System represented by the Native Order Gateway.

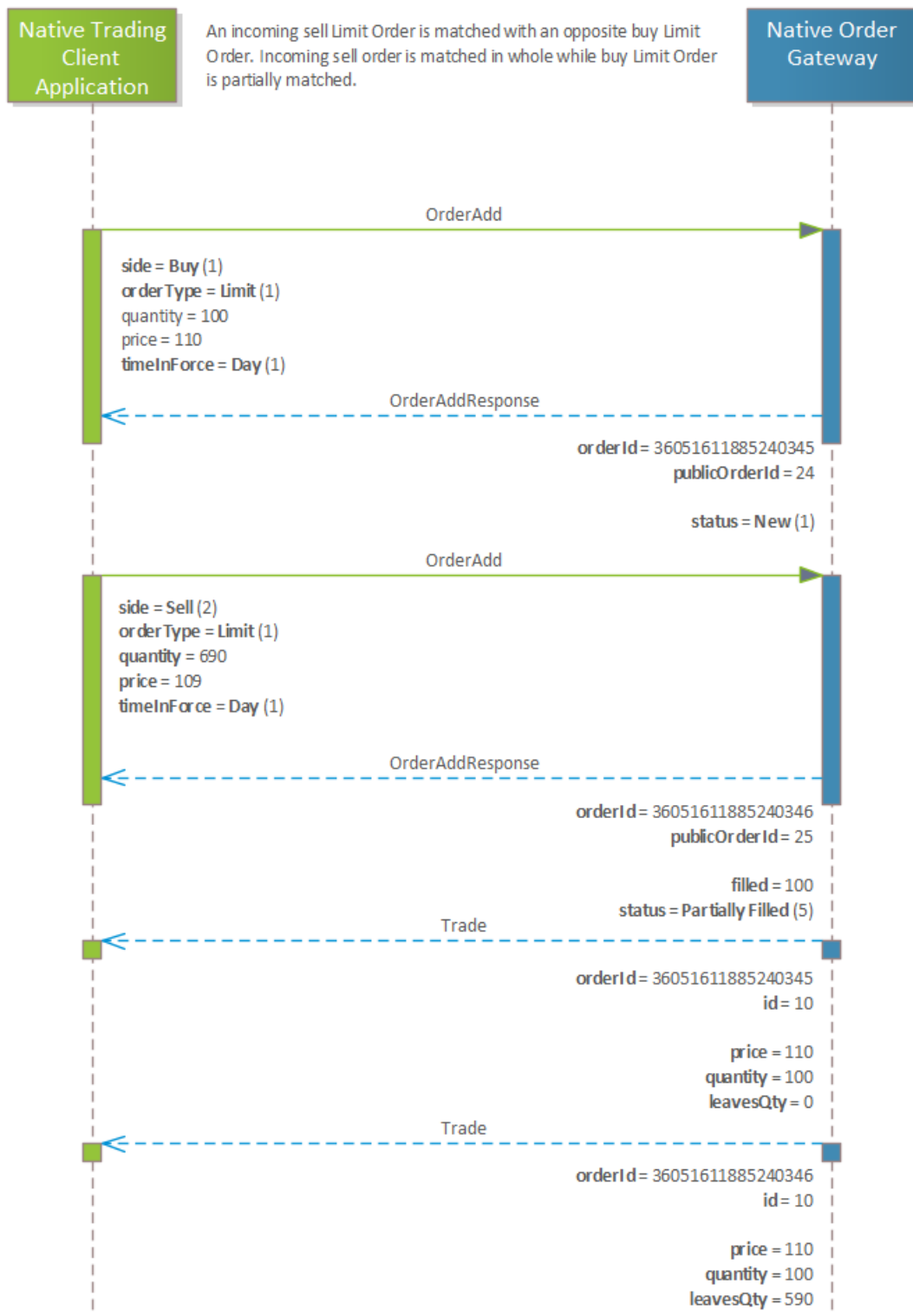
Native Order Gateway – a system that communicates with the Native Protocol Client Application.

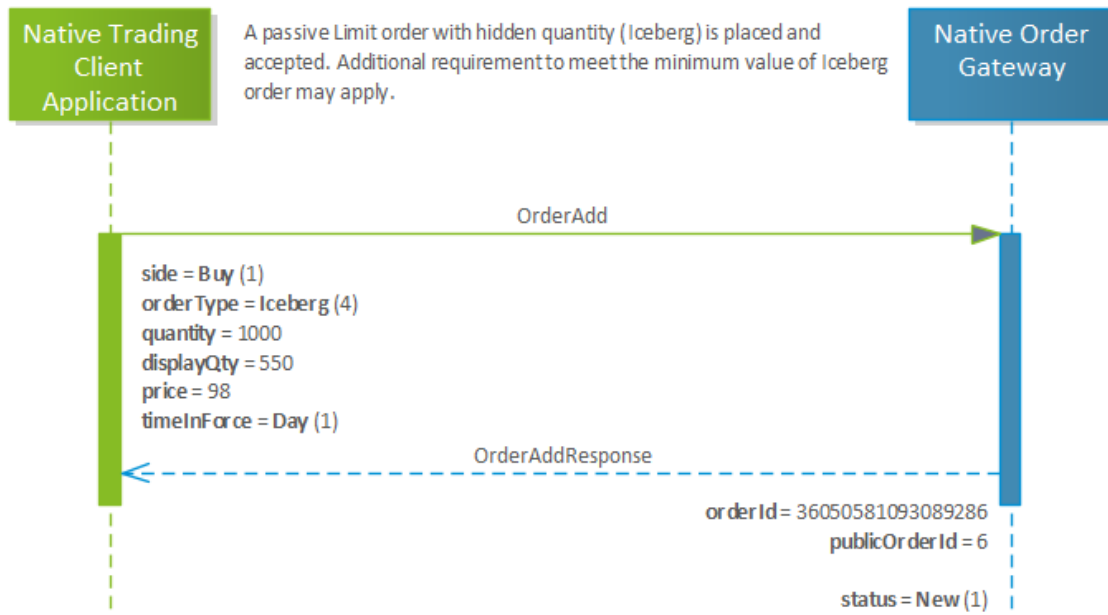
Note: presentation of simultaneous communication between multiple clients and Gateway was omitted. As a result, the Native Trading Client Application can either represent a single Exchange Member or simulate the exchange of messages between two different Exchange Members including buying, selling and conducting transactions.

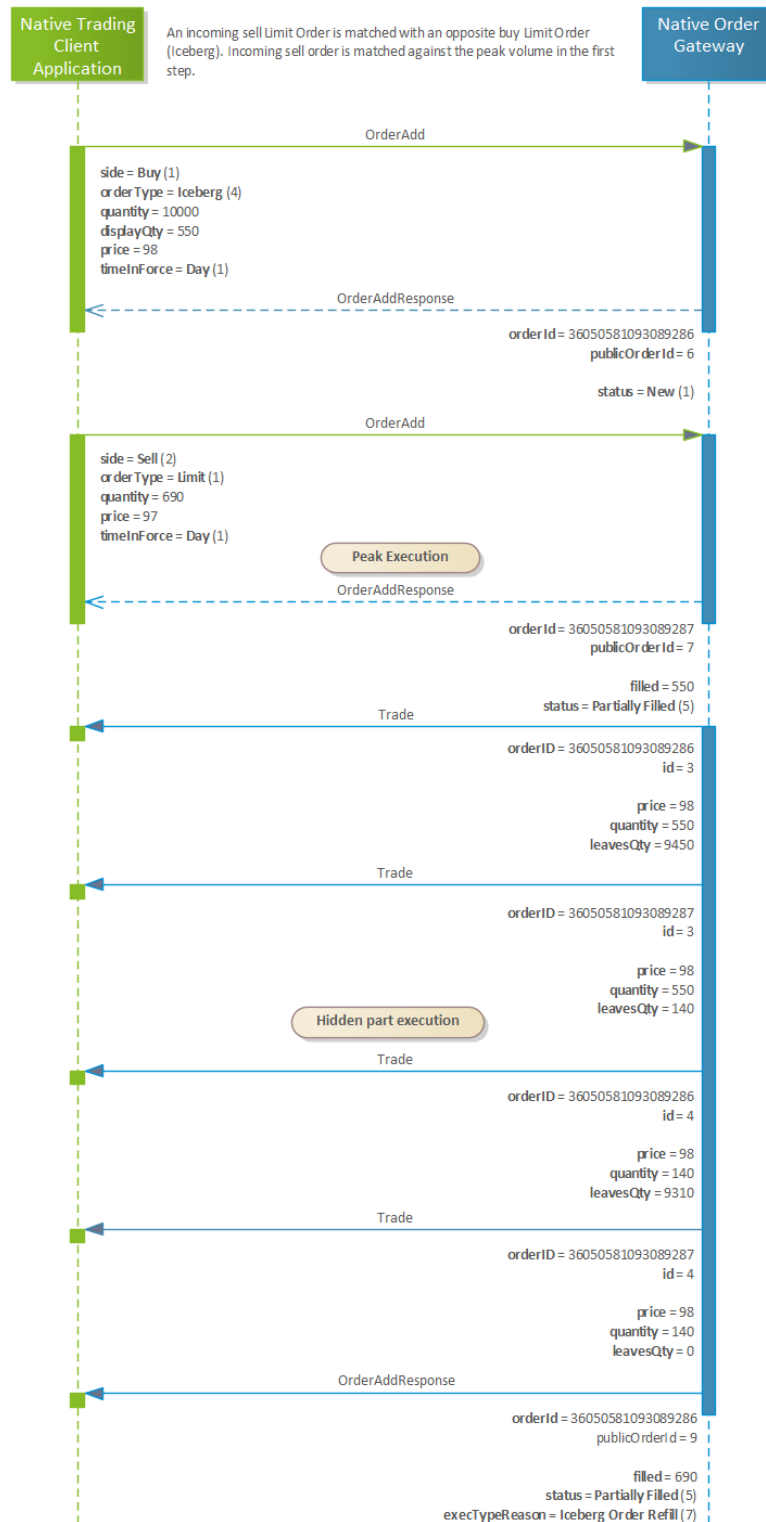
8.1. CLOB

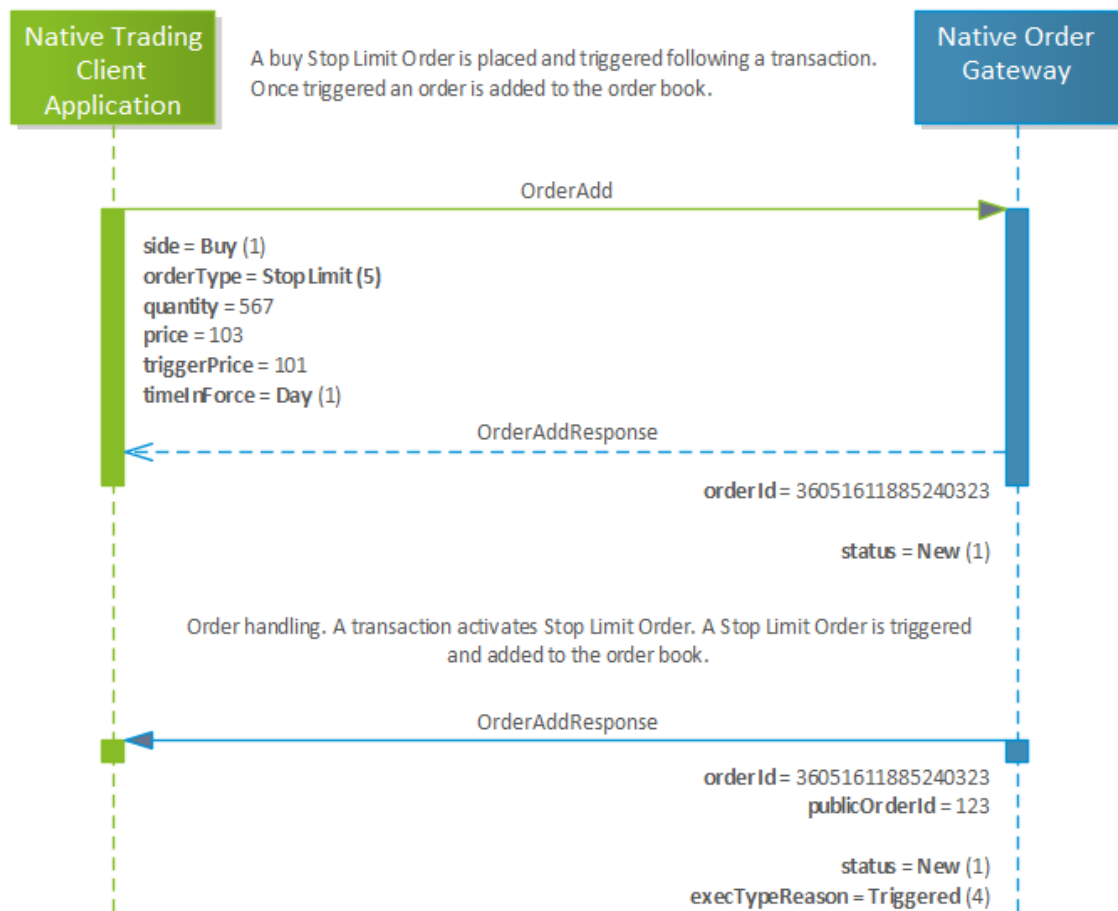
Order - Accepted

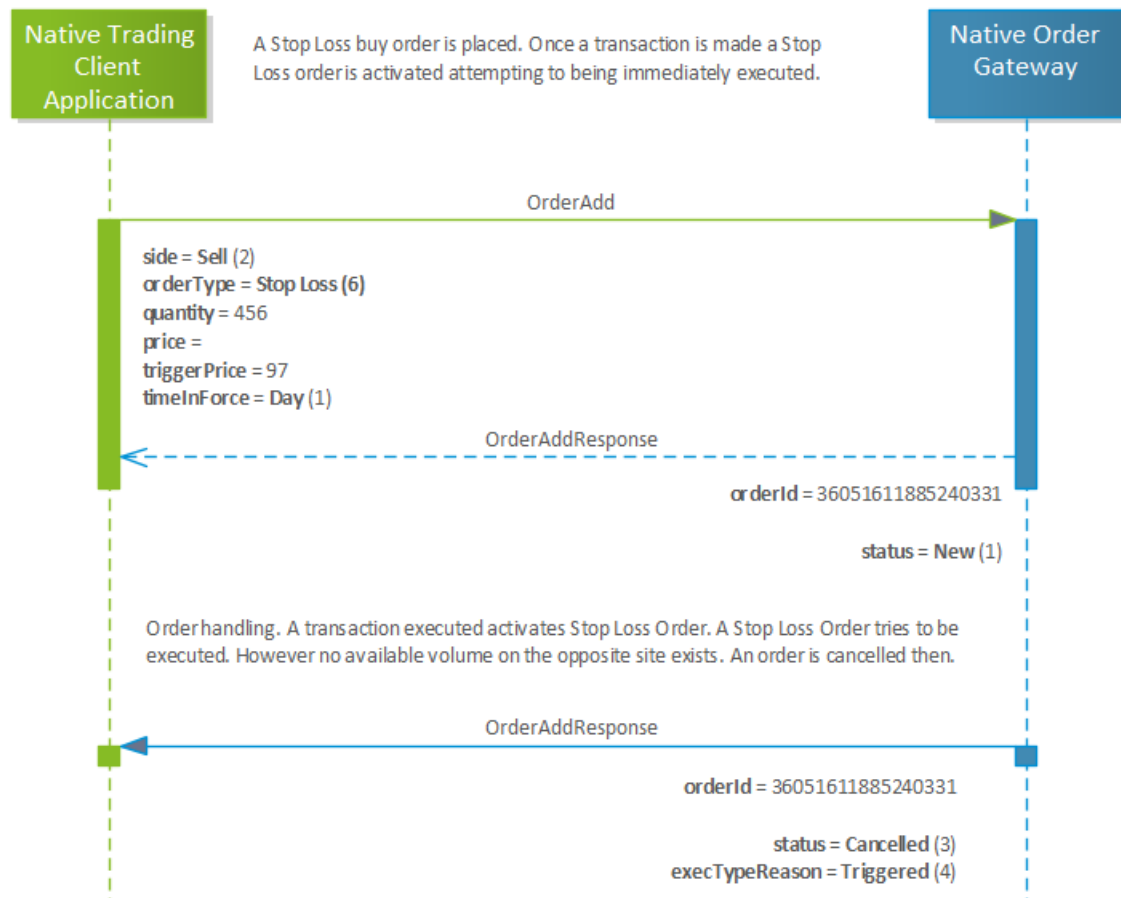


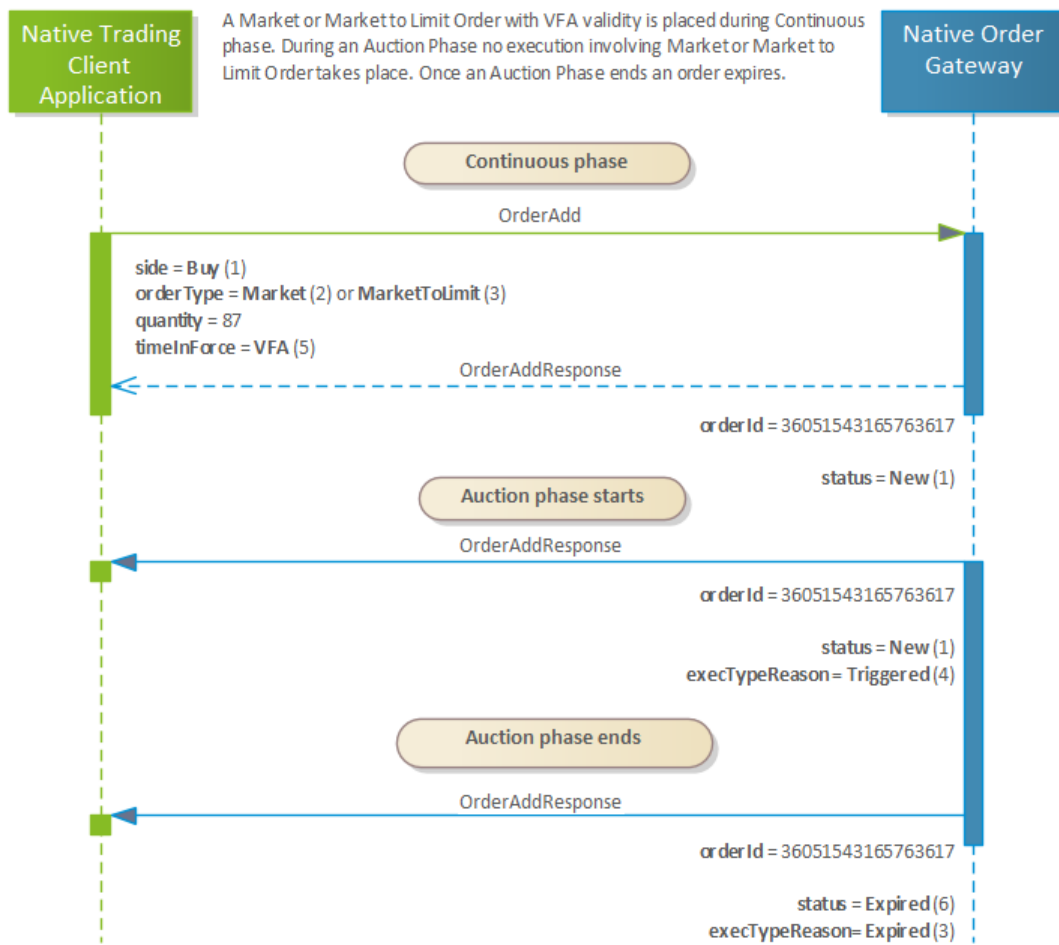
Order – Accepted and Filled – Full and Partially

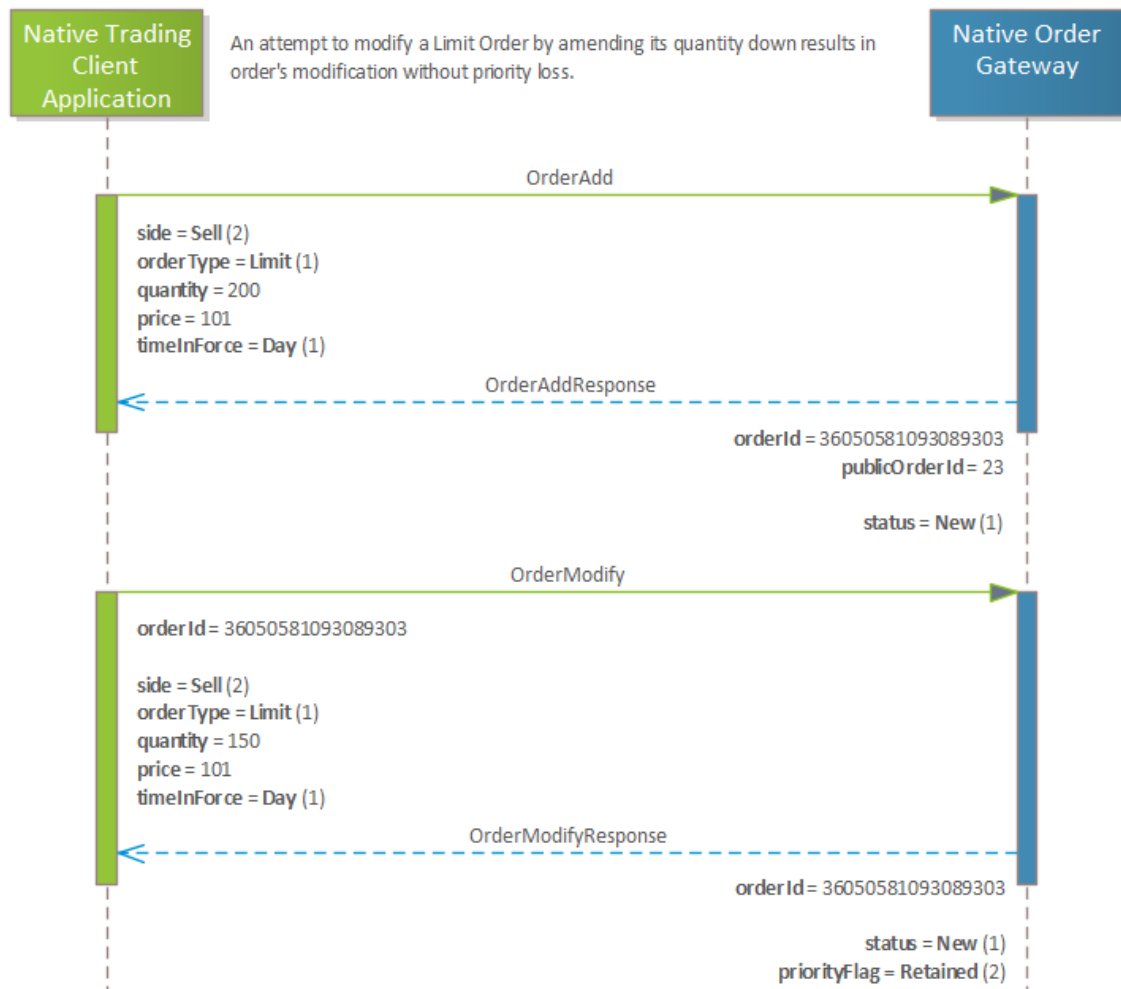
Iceberg Order – Accepted

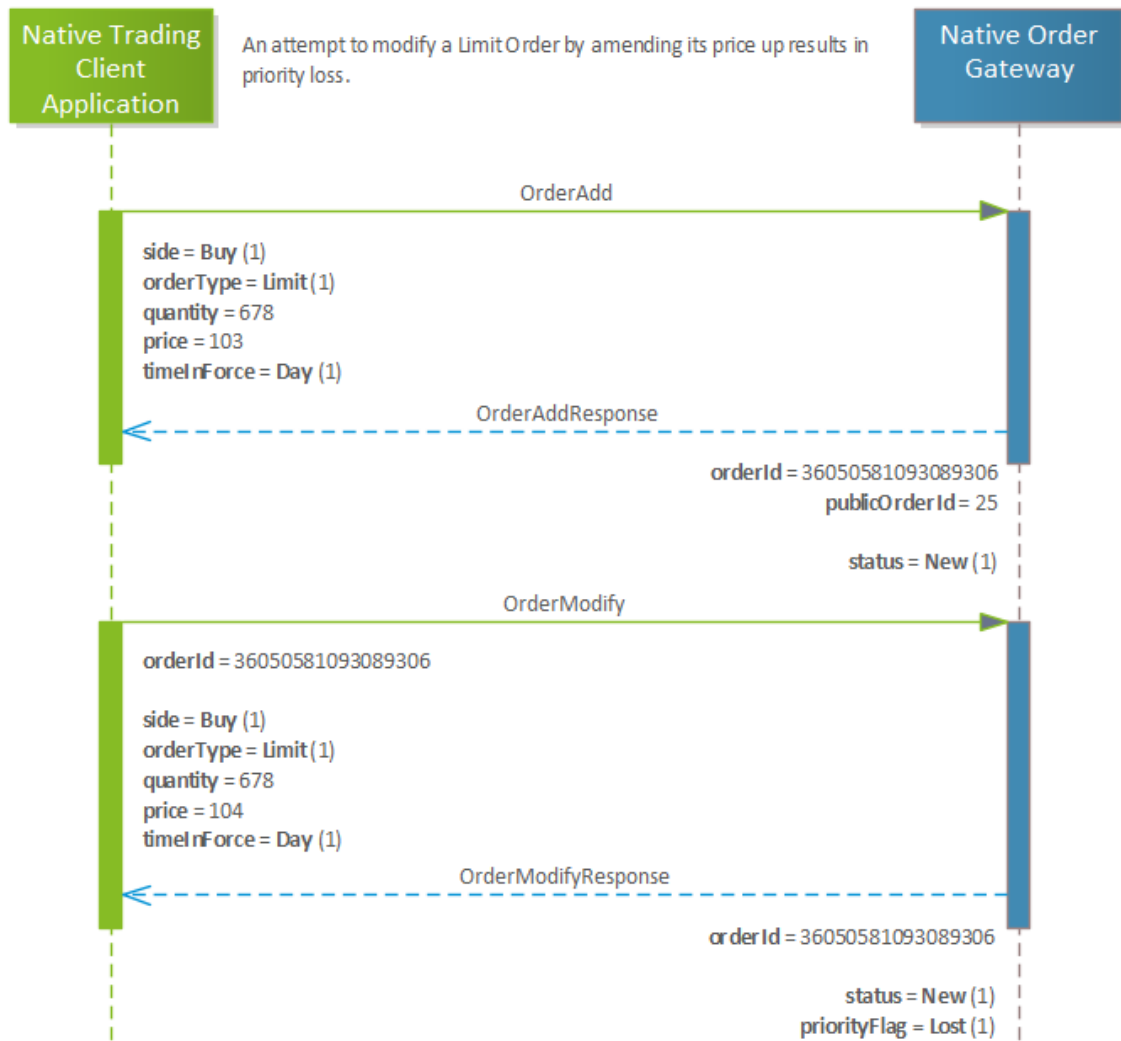
Iceberg Order – Accepted Partially Filled and Restated

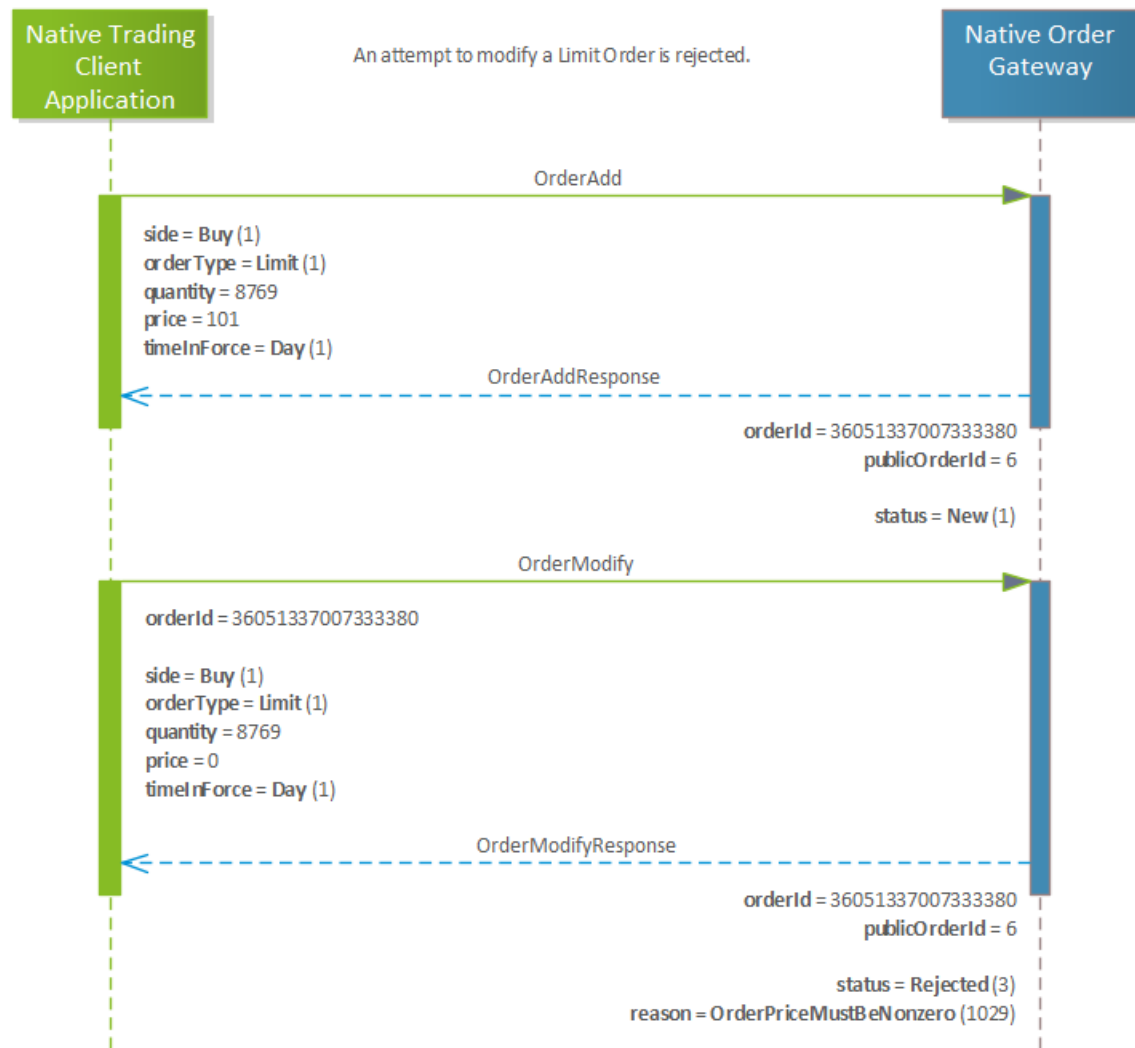
Stop Limit Order – Accepted and Triggered

Stop Loss Order – Accepted, Triggered and Cancelled

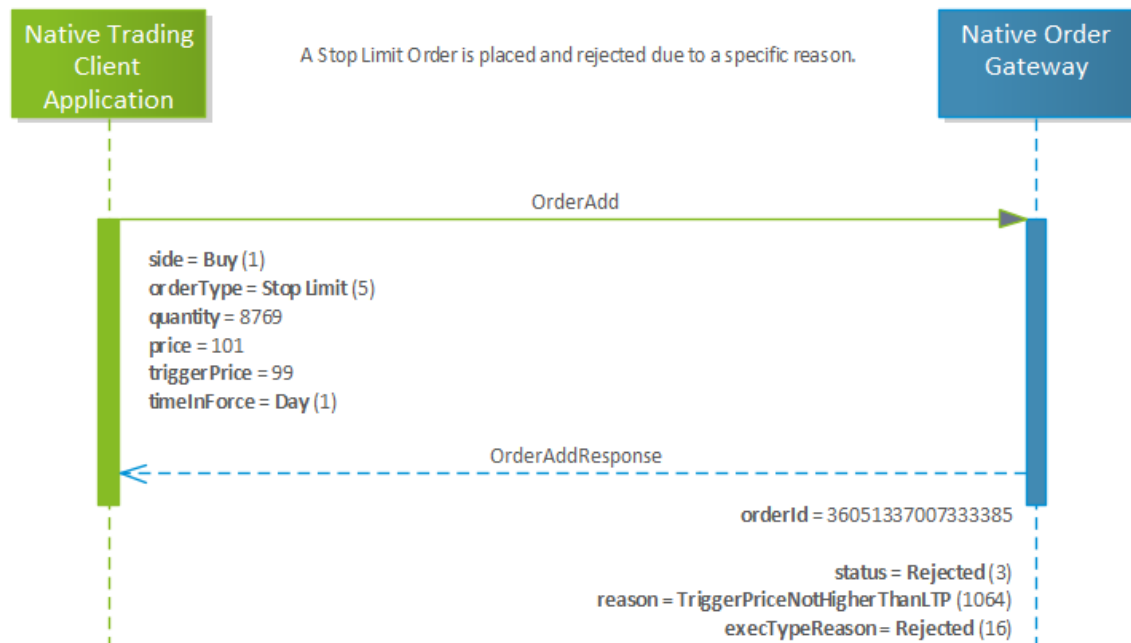
Market Order and Market to Limit Order with TIF VFA – Triggered during Auction

Order Modification without priority loss

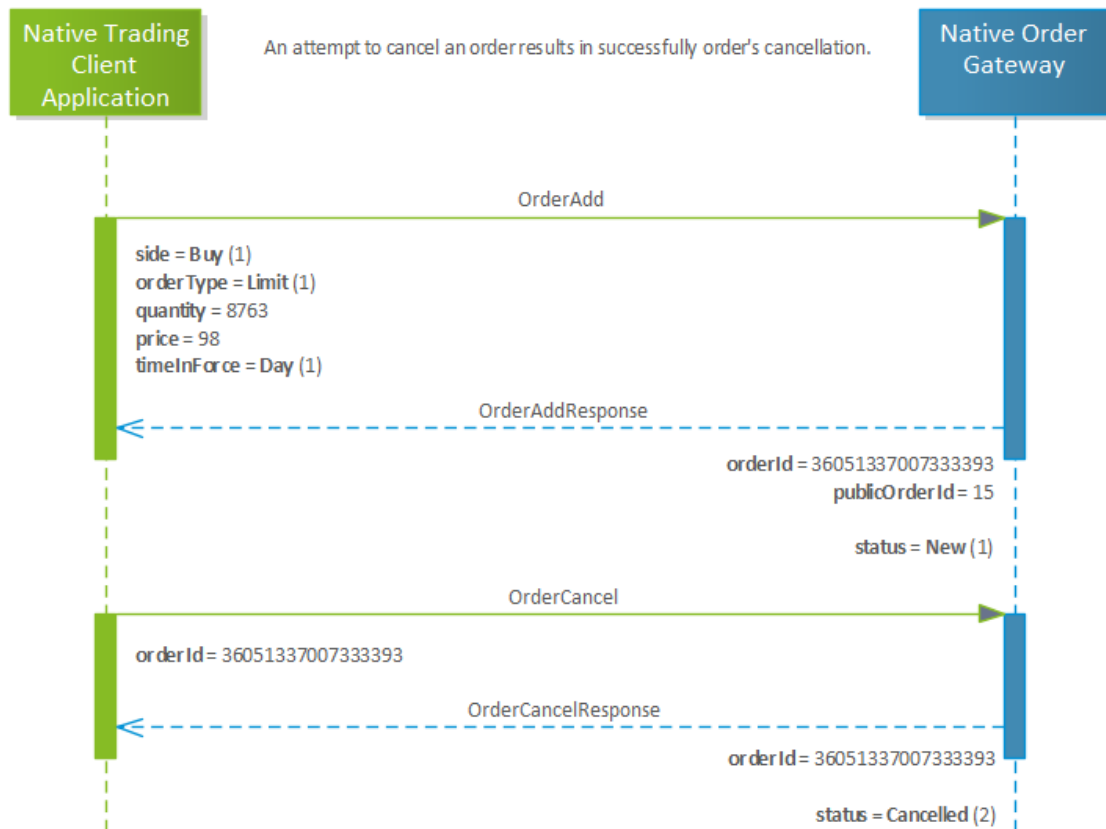
Order Modification with priority loss

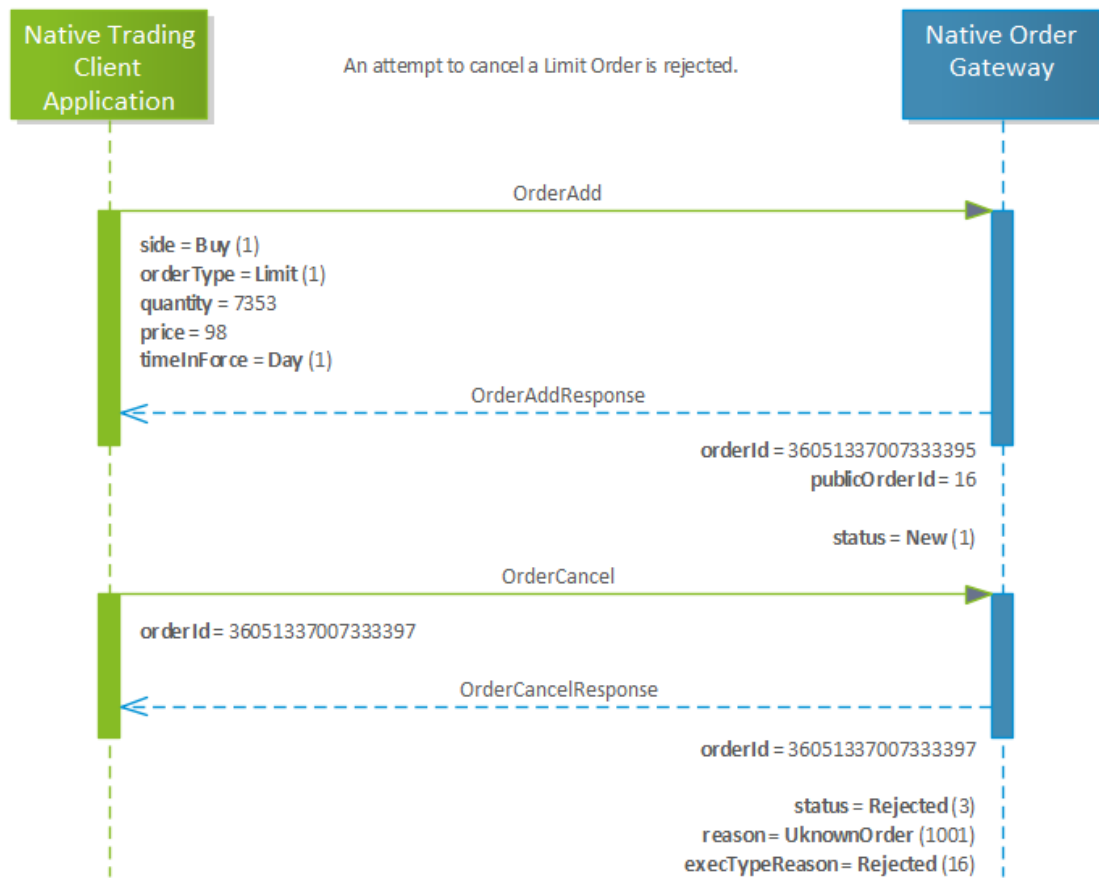
Order Modification – Rejected

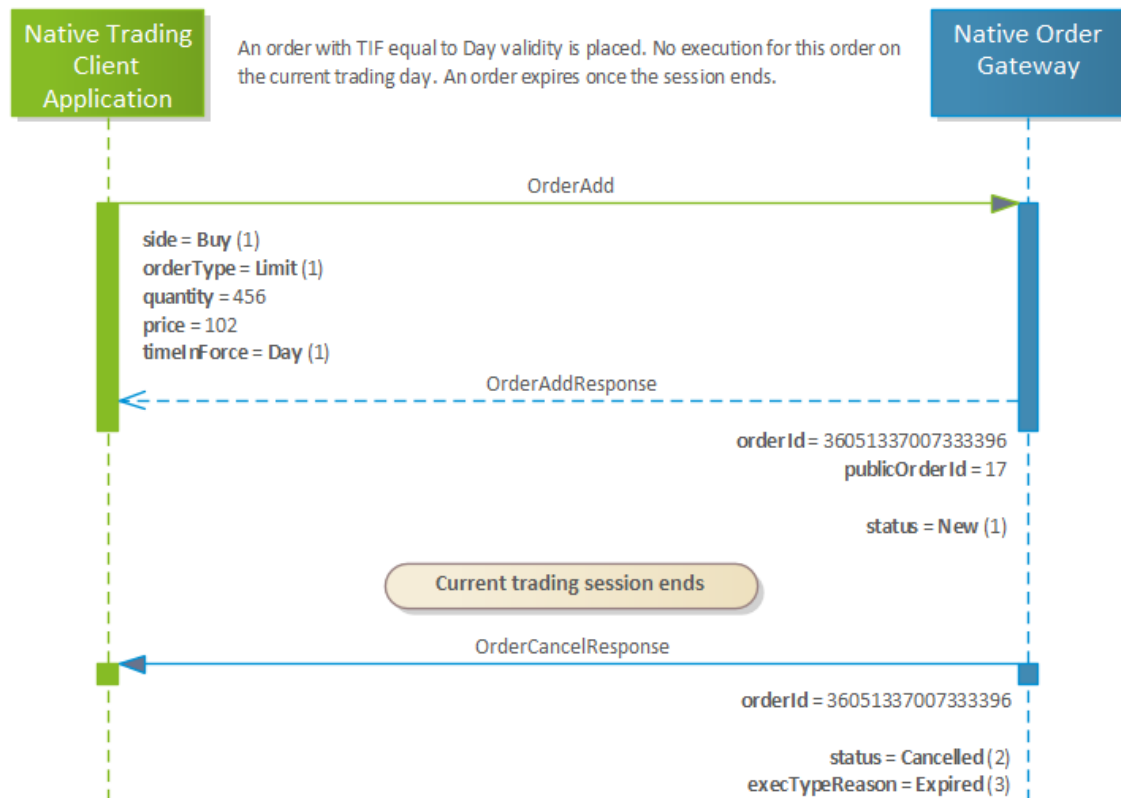
Order - Rejected

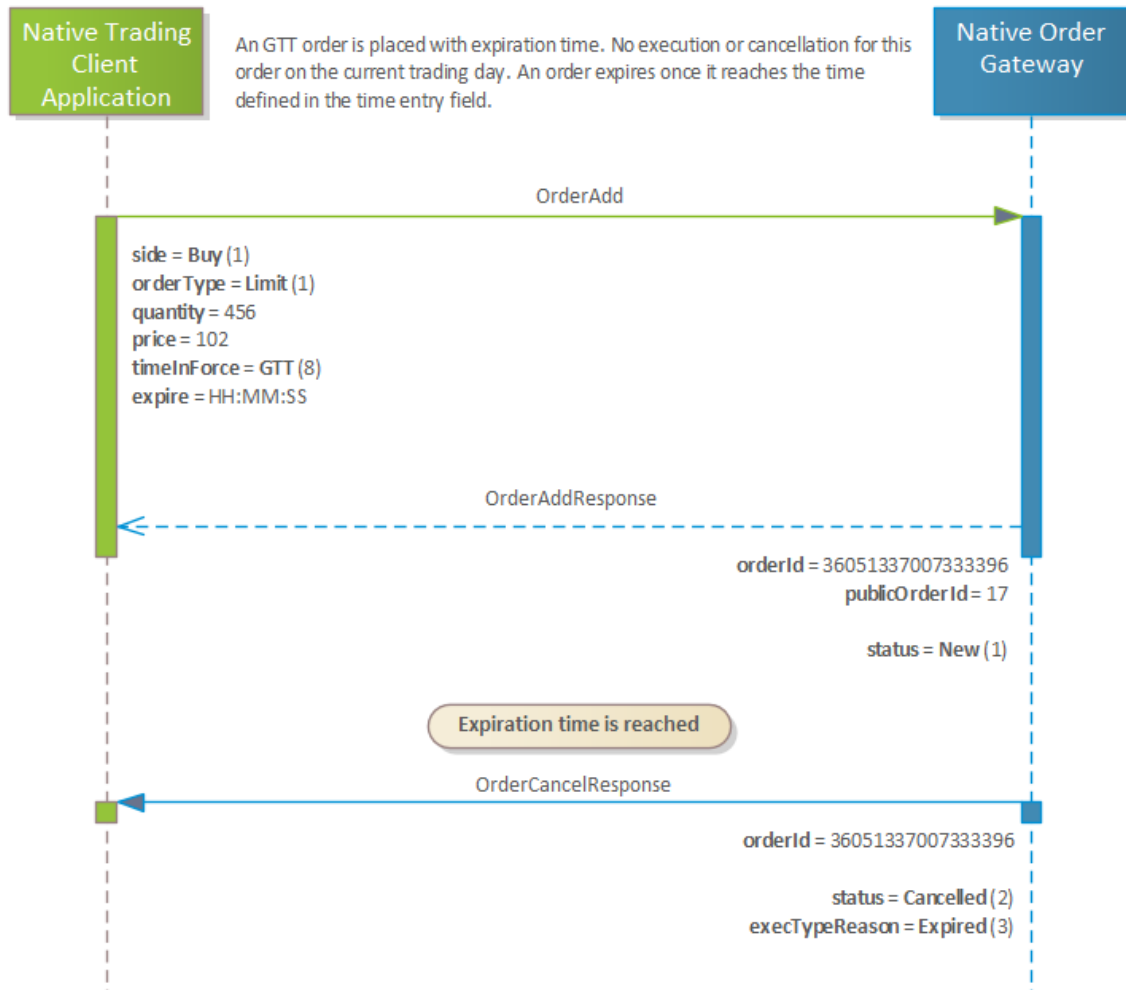
Stop Limit Order – Rejected

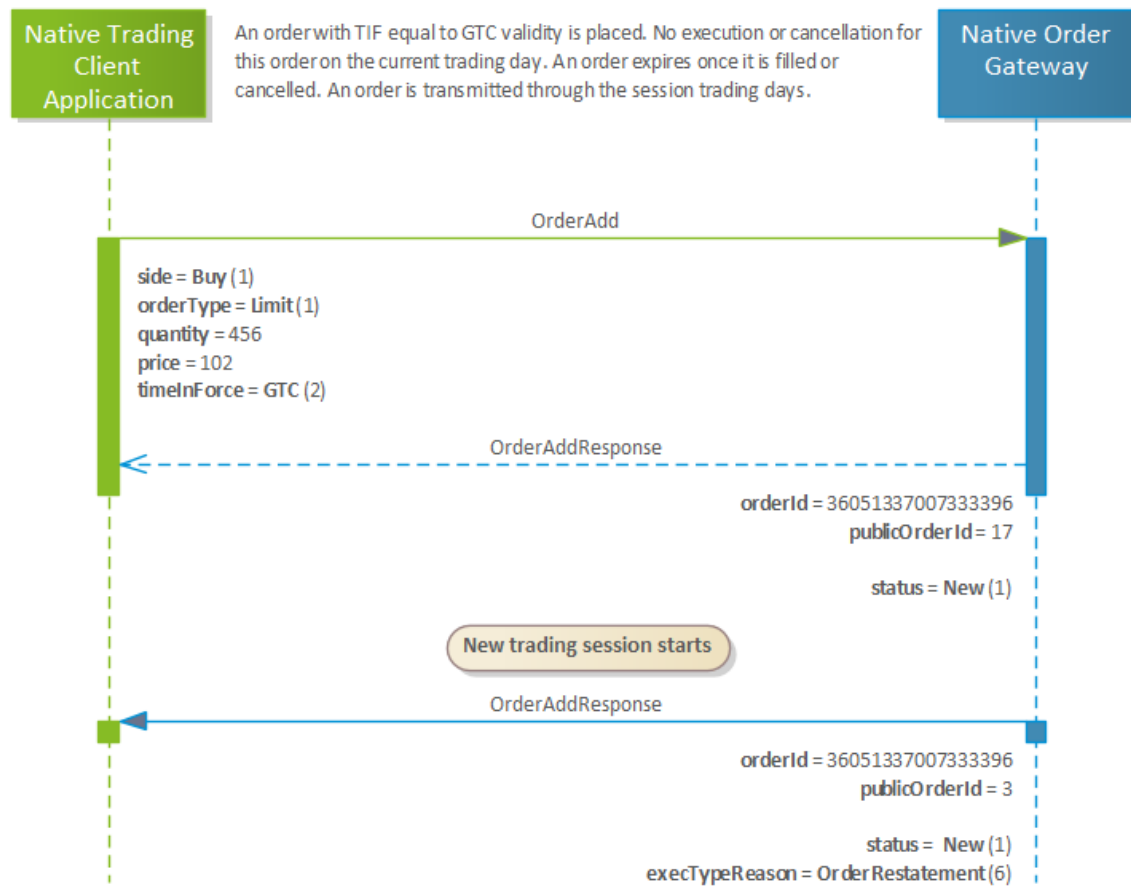
Market Order and Market to Limit Order ICO/FOK – Rejected during Auction

Order Cancellation – Accepted

Order Cancellation – Rejected

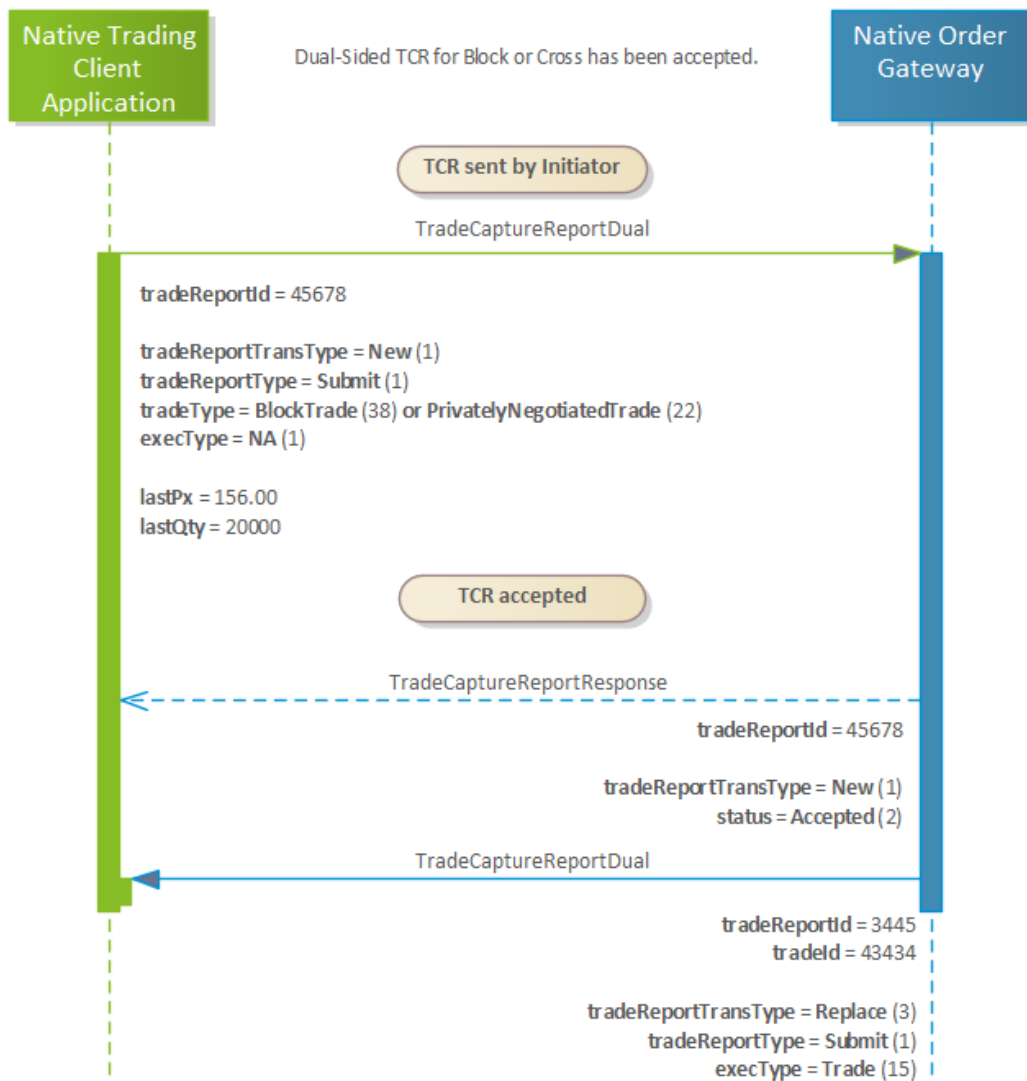
Day Order – Expired

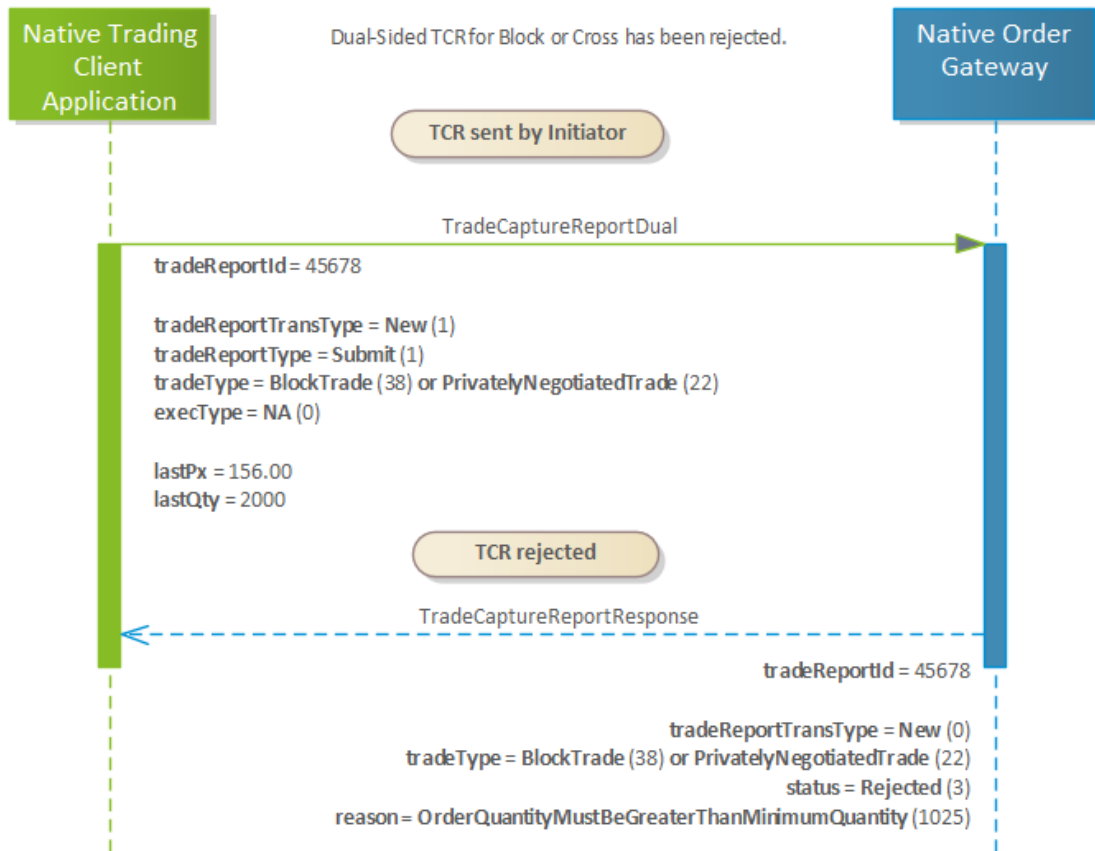
GTT Order – Expired

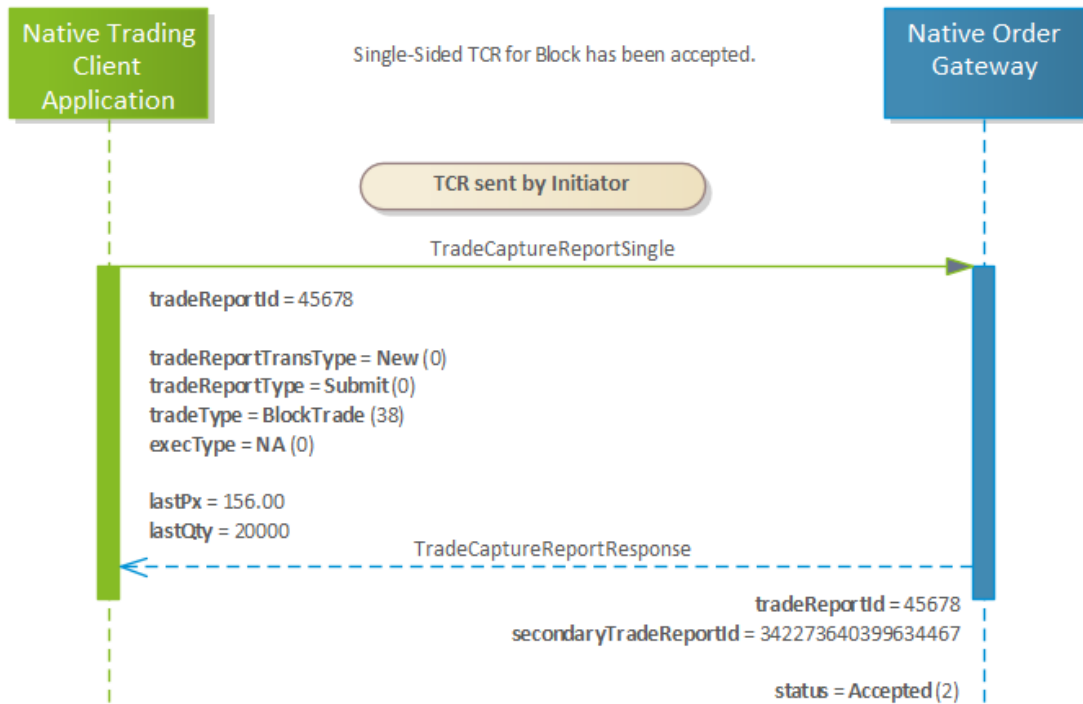
GTC Order – Reinstated

8.2. OFF-BOOK (BLOCK OR CROSS)

Dual Sided – Accepted

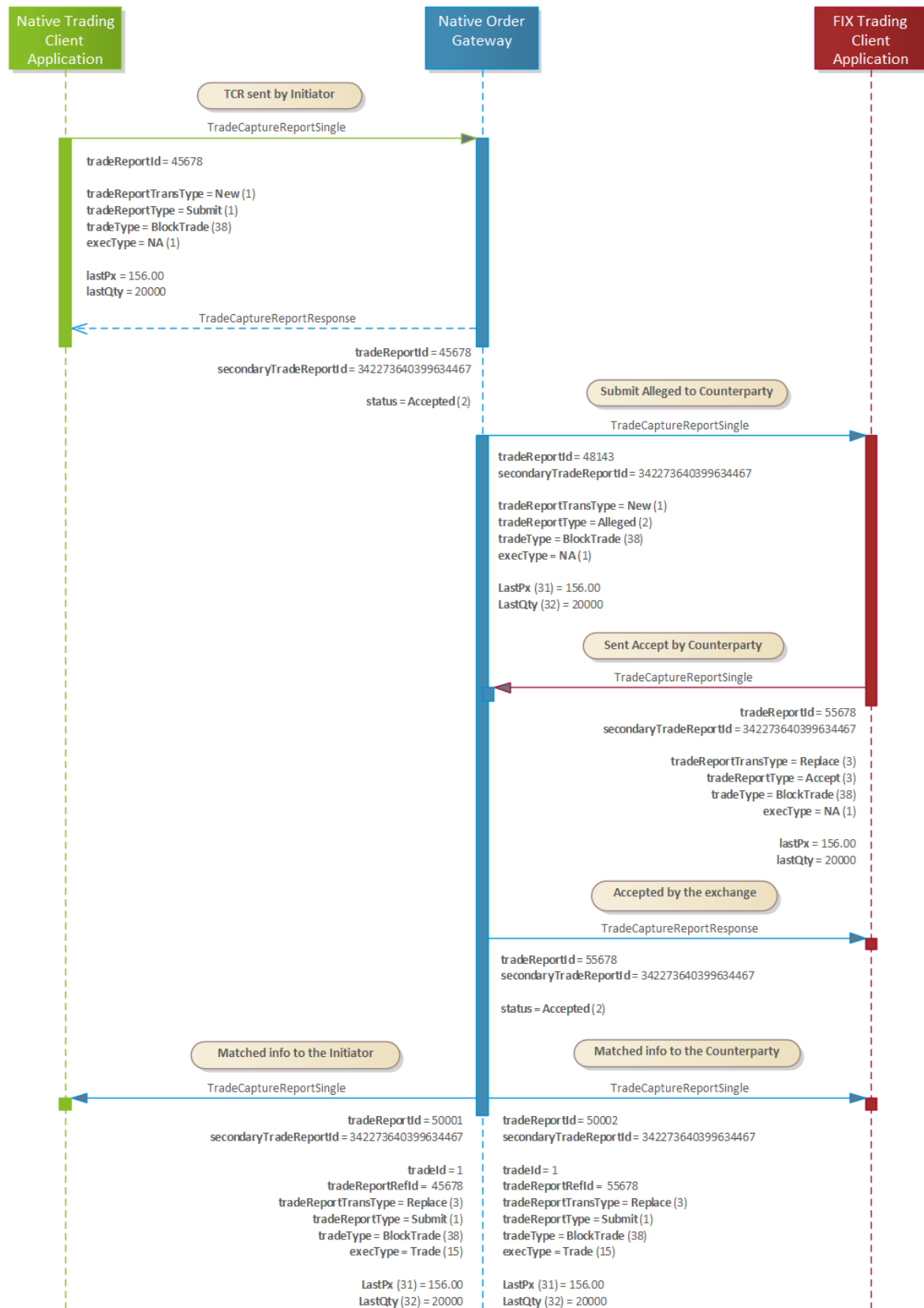


Dual Sided -Rejected

Single Sided – Accepted

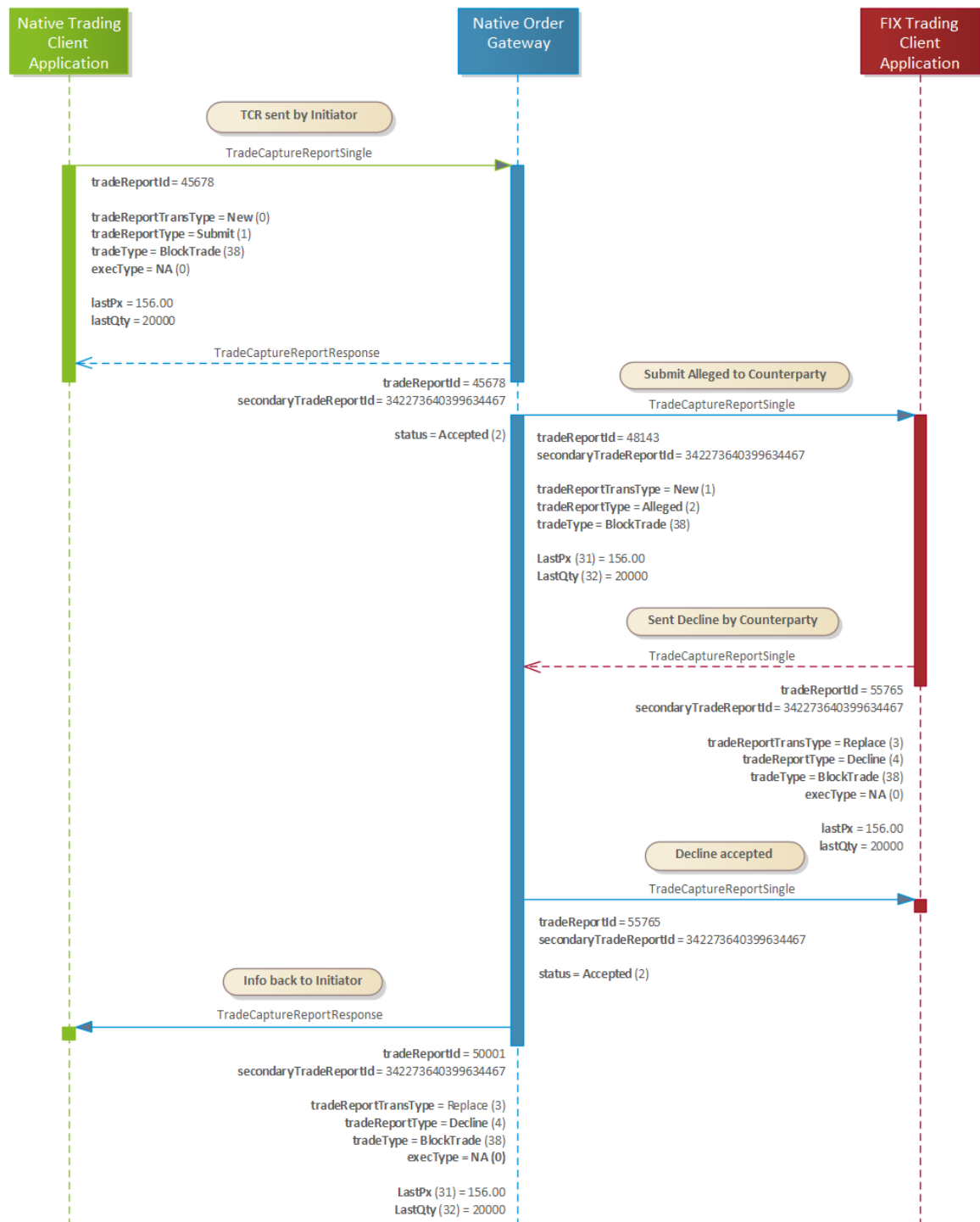
Single Sided – Accepted by Counterparty

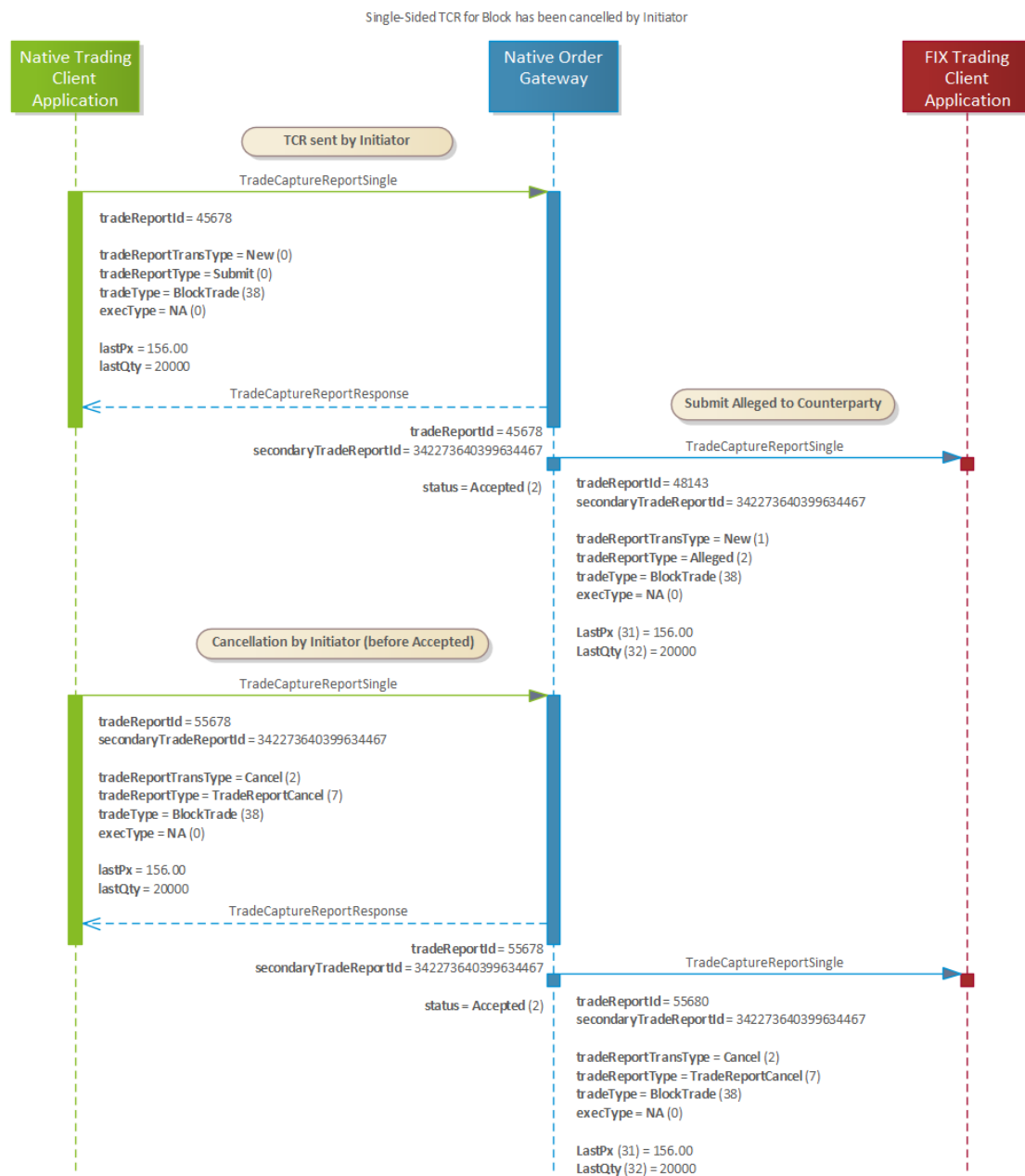
Single-Sided TCR for Block has been accepted by Counterparty.



Single Sided – Denied by Counterparty

Single-Sided TCR for Block has been declined by Counterparty (before being accepted).



Single Sided – Cancelled by Initiator

Single Sided – Rejected