FIX Protocol
One Day Course

By

Khader Shaik

Agenda – Part 1

- FIX Protocol Introduction
  - Overview
  - History
  - Usage / Players
  - Message Types
  - Message Format
  - Communication Model
  - Anatomy of sample message
  - Sample Flow
  - Understandng Specs

Agenda – Part 2

- FIX Protocol & Equity Trading
  - Electronic Trading Overview
  - Types of FIX Messages
  - Trading Scenarios

Agenda – Part 3

- Technical Implementation
  - Architecture
  - Buy-side Vs Sell-Side
  - Integration Methods
  - Commercial Engines
  - Free Engines
  - Implementation Steps
  - Testing Tips
  - Troubleshooting Production Issues
  - Roles & Responsibilities

Part - 1

What is FIX?

- FIX - Financial Information Exchange
- FIX Protocol is an industry driven messaging standard for exchange of Trading related information between financial institutions.
- FIX Protocol specification provides format for electronic messages and communication model
- FIX can be used by financial institutions like Broker-dealers, exchanges, Institutional investors and others in the industry to communicate among each other
- It is widely used protocol in the Financial Markets Industry today

FIX History

- FIX Protocol introduced in 1992
- FIX Specifications are developed and managed by the members of organization known as "FIX Protocol Limited (FPL)“
  - http://www.fixprotocol.org
- FPL is formed by major industry players
- FIX Versions:
  - Latest FIX version is 5.0
  - Most of the current production versions are 4.1 to 4.4

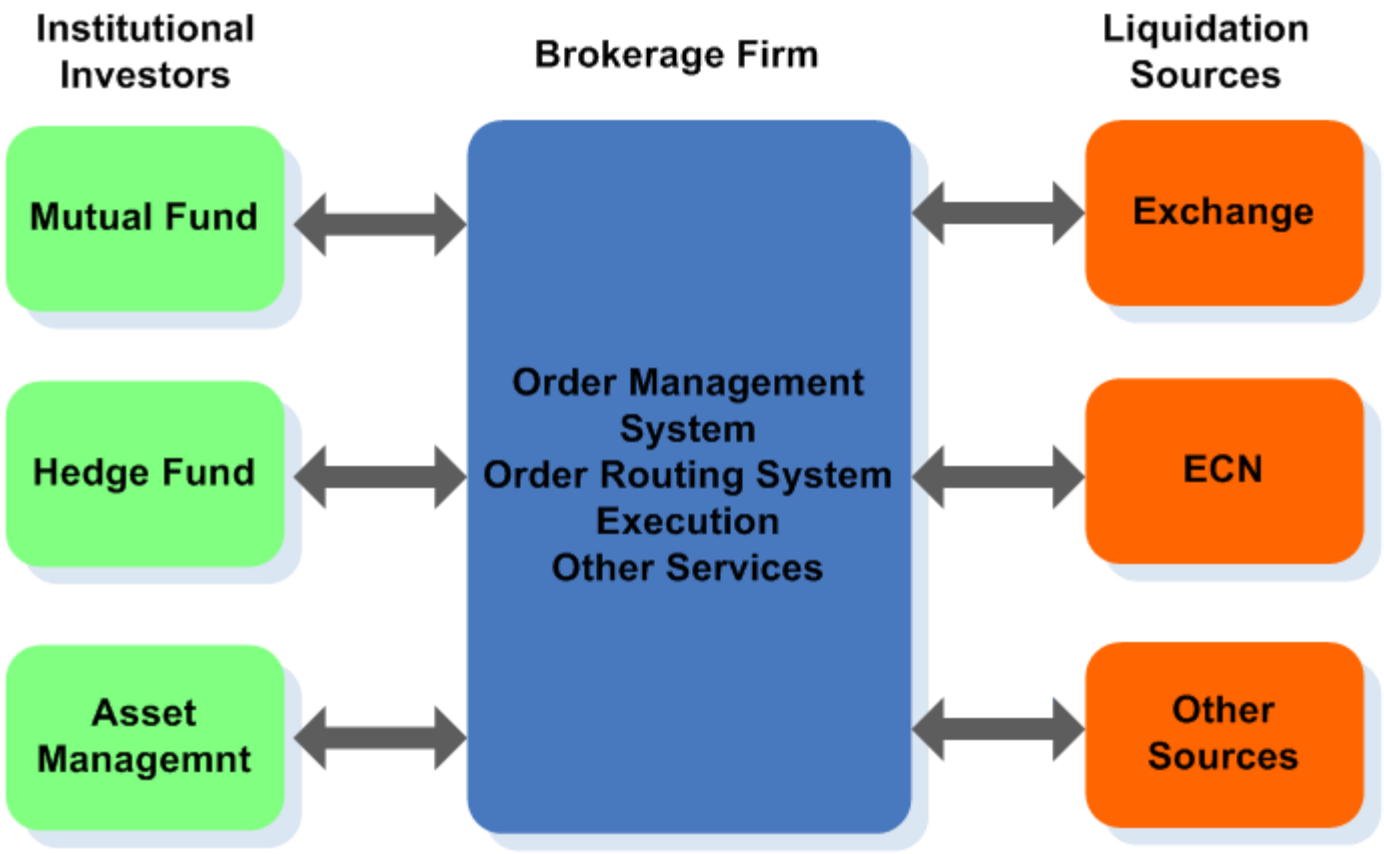
Benefits of FIX Protocol

- FIX is Open Protocol – brings all players together
- Electronic protocol – streamlines and increases the efficiency of communication among industry players
- Wide support and software packages availability – easy to get on board
- Simple and platform independent protocol
- Designed by users and extensible – addresses all industry needs

Industry Players & Usage

- Exchanges
  - Use FIX to receive trades from their members and send executions back and other trading related messages
- Buys-side firms
  - Use FIX to send and receive pre-trade, trade and post-trade messages to and from Sell-side firms
- Sell-side firms
  - Use FIX to receive and send pre-trade, trading, post-trade messages from and to buy-side firms
  - Use to communicate with Exchanges and other OTC markets

Industry Players & Usage



Industry Players – FIX Usage

Supported Product Classes

- Equities
- Fixed Income
- Derivatives (Options, Futures, IR Swaps etc)
- FX etc.

Message Categories

- Admin Messages
  - Used to maintain the different aspects of FIX session (connection)
- Application Messages
  - Messages used for transmission business messages

Admin Messages

- Logon – Client Authentication Message
- Logout – Normal Termination of Session
- Heartbeat- Used to check communication link between two parties
- Test Request – used to test the health of the communication link
- Resend Request – Request to retransmit the certain application messages
- Reject (Session Level) – session level validation failure (different from application level validation)
  - Example invalid version, msg type etc
  - RejectReason is populated with error info
- Sequence Reset/Gap Fill
  - In case of communication problems missing messages recovered or sequence is reset to ignore the missing messages

Application Messages

- Pre-trade messages
  - IOIs, Quotes, News, Email, Market Data, Security Info etc
- Trade Messages
  - Single Orders, Basket/List Orders, Multi-leg orders, Executions, Order Cancel, Cancel/Replace, Status etc
- Post-Trade Messages
  - Allocations, Settlement Instructions, Positions Mgmt etc

FIX Message Format

- FIX is a platform independent protocol
- Message contains 3 parts:
  - Header
  - Body
  - Trailer/Footer
- Message is a collection of fields
- Each field is a tag-value pair
  - <tag>=<Value>
  - Eg: 55=IBM (symbol=IBM)
- All fields are terminated by delimiter character
- Delimiter character is a non-printing ASCII - "SOH" (#001) (in print '^' is used)
- Delimiter cannot be used in the message anywhere else except in DATA field
- 9=0235^35=D^34=10^43=N^

FIX Message Format ..cont

- Tag
  - FIX uses predefined Tags
  - Each Tag represent the specific field
  - Each tag is given a predefined number
  - FIX Field dictionary provides the list of Fields and corresponding Tag numbers (Supplied with Spec)
  - Dictionary is available at the end of specification (by number and by tag name)
- Value
  - Values represent the value of the Tag assigned to
  - Supported Data Types are:
    - int, float, char, time, date, data, string

FIX Message Format ..cont

- All messages start with "8=FIX.*x.y*"
  - Indicates the FIX version of the message being transmitted
  - Useful to support multiple versions
- All messages terminate with "10=*nnn*<SOH>"
  - *nnn* represents the Checksum of the data
  - Checksum is the sum of all the binary values in the message
  - Checksum helps to identify the transmission problems

Sample Message

- New Order Single Message

8=FIX.4.1^9=0235^35=D^34=10^43=N^49=VENDOR^50=CUSTOME^56=BROKER^52=19980930-09:25:58^1=XQCCFUND^11=10^21=1^55=EK^48=277461109^22=1^54=1^38=10000^40=2^44=76.750000^59=0^10=165

Communication Model

- Session based communication
- Session is communication between two parties
- Initiator / Client
  - party who initiates the communication
- Acceptor /Server
  - party who receives connection request from Initiator
  - Server validates client request using login message

FIX Session

- FIX is a session protocol
  - Each session maintains the bi-directional messages between two parties
  - Session can spread across multiple physical connections
  - Session is maintained using sequence number
  - Both parties rely on sequence numbers to maintain the orderly communication
  - Every new session starts with sequence number 1
  - Missing messages are re-transmitted with bi-lateral agreement between both parties

Typical Session Flow

- Client starts the session with LOGON Message
- Exchange Business/Application Messages with Server
  - Sending new orders, receiving fills etc
- Ends with LOGOUT message

FIX Message Categories

- Admin/Session Messages
  - Used for Session maintenance
  - Eg: Logon, Logout, Heartbeat etc
- Application Messages
  - Used for exchange of business messages
  - Eg: New Order, Cancel Order etc
- Security
  - Authentication - Login and Logout between parties
  - Data Security - Options for data encryption (PGP)

Application Messages

- Trade Messages
  - New Order (Single)
  - Execution Report
  - Order Cancel Request
  - Order Cancel/Replace Request
  - Order Status Request etc

New Order (Single)

- Used to send a new (buy, sell etc) order to broker or an exchange.
- New order message provides numerous tags to support all possible information required with New order
- Has some mandatory fields that are common to every New order
  - Eg: ClientOrderID, Symbol etc
- Sample New Order Message:

8=FIX.4.1^9=0235^35=D^34=10^43=N^49=VENDOR^
50=CUSTOMER^56=BROKER^52=19980930-09:25:58^
1=XQCCFUND^11=10^21=1^55=EK^48=277461109^
22=1^54=1^38=10000^40=2^44=76.750000^10=165

New Order (Single) …cont

- FIX Version (8) = 4.1
- Message Body Length(9) = 0235
- Message Type (35) = D (New Order single)
- Message Seq (34) = 10
- PossDupFlag (43) = N (no)
- SenderCompID(49) = VENDOR (unique id of the sender firm)
- SenderSubID(50) = Vendor Sub id like desk etc (Optional)
- TargetCompID(56) = BROKER (value used to identify receiving firm)
- SendingTime (52) = Time of message transmission
- Account(1) = Account number
- ClOrdID(11) = Client Order Id

New Order (Single) …cont

- HandlInst(21) = Order Handling Instructions to Broker
- Symbol(55) = Security Identifier - Ticker
- SecurityID (48) = CUSIP or other alternate security identifier
- Side (54) = side of the order

IDSource(22) = Identifies class of alternative SecurityID

1 = CUSIP
2 = SEDOL
3 = QUIK
4 = ISIN number
5 = RIC code

Side (54) = Side of Order. Values are:

1 = Buy
2 = Sell
3 = Buy minus
4 = Sell plus
5 = Sell short
6 = Sell short exempt

New Order (Single) …cont

- OrderQty (38) = Order Quantity (eg: Number of shares ordered)
- OrdType(40) = Order Types
  - 1 = Market
  - 2 = Limit
  - 3 = Stop
  - 4 = Stop limit etc
- Price(44) = Price of order if the order is Limit etc
- Checksum(10) – used for data integrity check

Few Other Application Messages

- Order Cancel Request
  - This message is used to request the cancellation of full or part of the remaining quantity of the existing order.
- Order Cancel/Replace Request
  - This message is used to modify the existing order
- Order Status Request Message
  - This message is used to request the status of existing order

Execution Report

- Used by the recipient (Broker or Exchange)
- Used for various needs like
  - Used to confirm the receipt of an order
  - Confirm changes to an existing order (in response to order cancel request etc)
  - Relay order status information
  - Reject orders
  - Relay Fill (execution) information etc

Message Type Codes (Tag 35)

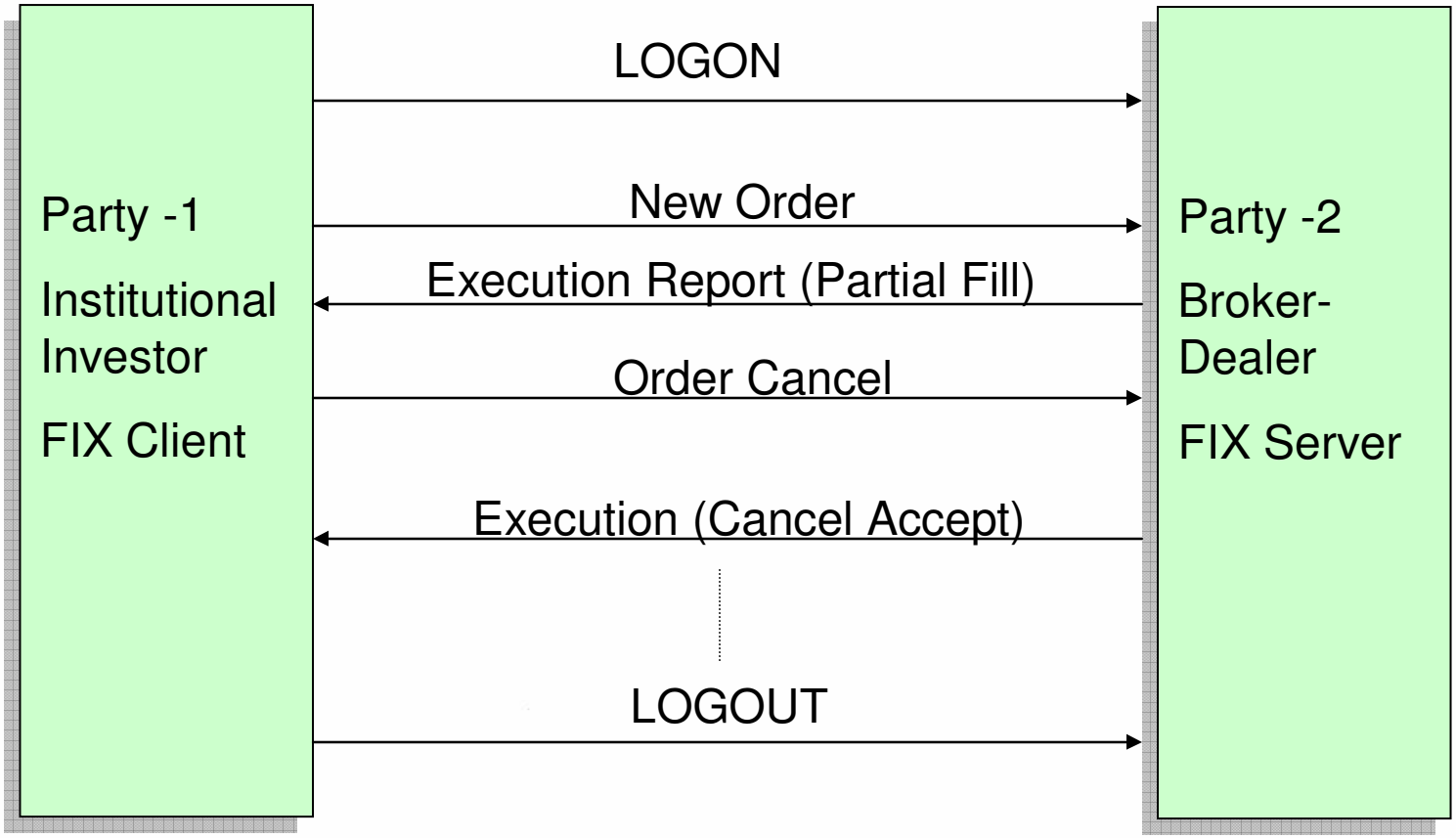
- 0 = Heartbeat
- 1 = Test Request
- 2 = Resend Request
- 3 = Reject
- 4 = Sequence Reset
- 5 = Logout
- 6 = Indication of Interest
- 7 = Advertisement
- 8 = Execution Report
- 9 = Order Cancel Reject
- A = Logon
- B = News
- C = Email
- D = Order - Single
- E = Order - List
- F = Order Cancel Request
- G = Order Cancel/Replace Request
- H = Order Status Request
- J = Allocation
- K = List Cancel Request
- L = List Execute
- M = List Status Request

and more

Administrative Messages

- Logon - Starts the Session
- Heartbeat – Used to check the health in case of idle
- Test Request
- Resend Request
- Logout etc

Sample Flow



Part - 2

FIX Protocol Equity Trading

By

Khader Shaik

Agenda

- Electronic Trading Overview
- Types of FIX Messages
- Trading Scenarios

Trade Life Cycle

- Pre-Trade – Trade prep
  - IOI – Market the Interest/positions
  - Quote Request and Quote
- Trade
  - Trading Activity Messages
- Post-Trade – Settlement etc
  - Allocations – Distribution of executions etc
  - Settlement Instructions
  - Street Side reporting (contra/counterparty info)

Execution report Message

- Simply known as Execution
- Interpreted using ExecType, ExecTransType and OrdStatus fields
- ExecTransTypes

NEW	Order Acknowledgement
CANCEL	Cancel previously reported execution due to error etc.
CORRECT	Correction to the previously reported execution.
STATUS	Reports the status of the orders.

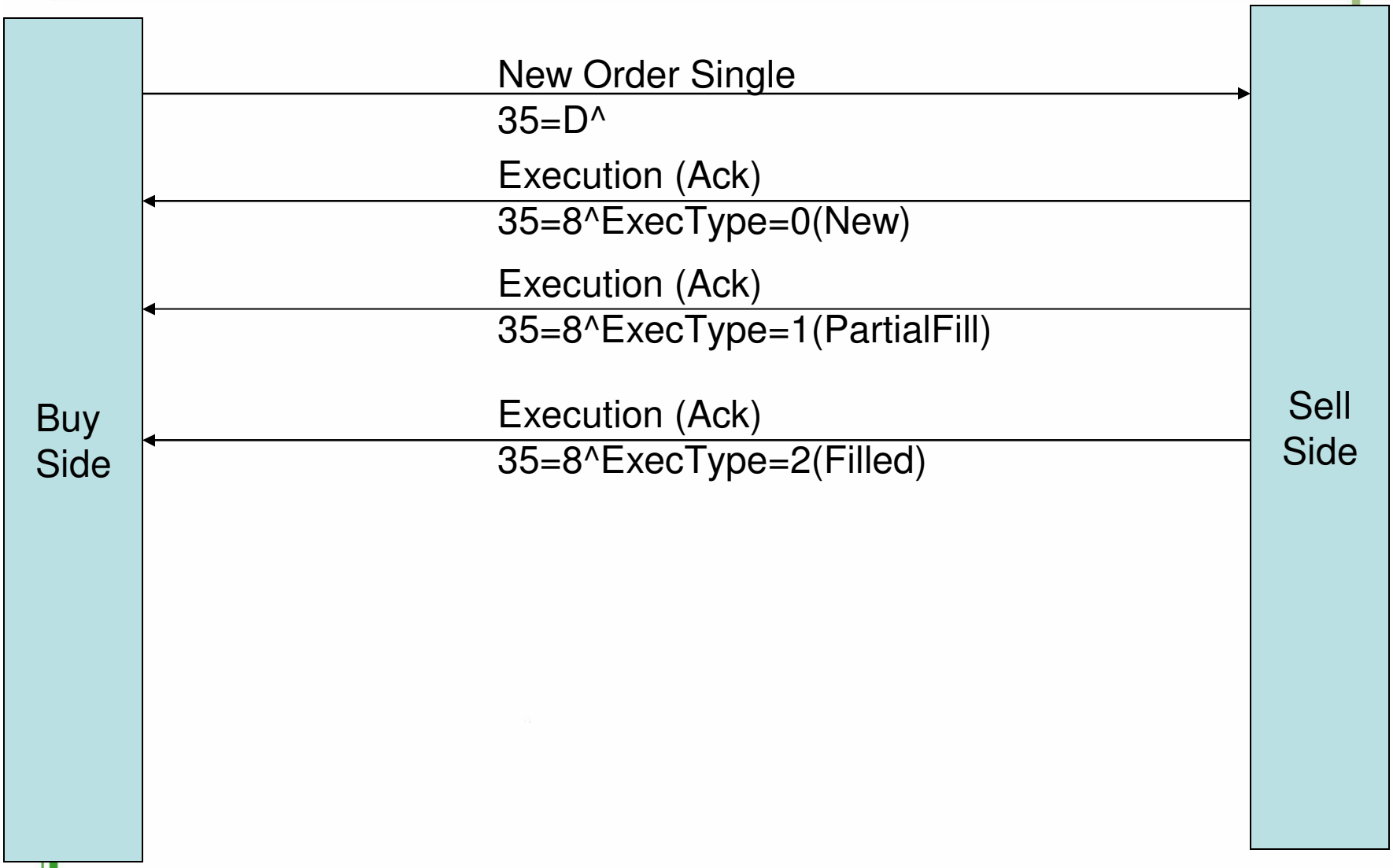
ExecTypes

- 0 = New
- 1 = Partial fill
- 2 = Fill
- 3 = Done for day
- 4 = Canceled
- 5 = Replace
- 6 = Pending Cancel (e.g. result of Order Cancel Request)
- 7 = Stopped
- 8 = Rejected
- 9 = Suspended
- A = Pending New
- B = Calculated
- C = Expired
- D = Restated (ExecutionRpt sent unsolicited by sellside, with ExecRestatementReason set)
- E = Pending Replace (e.g. result of Order Cancel/Replace Request)

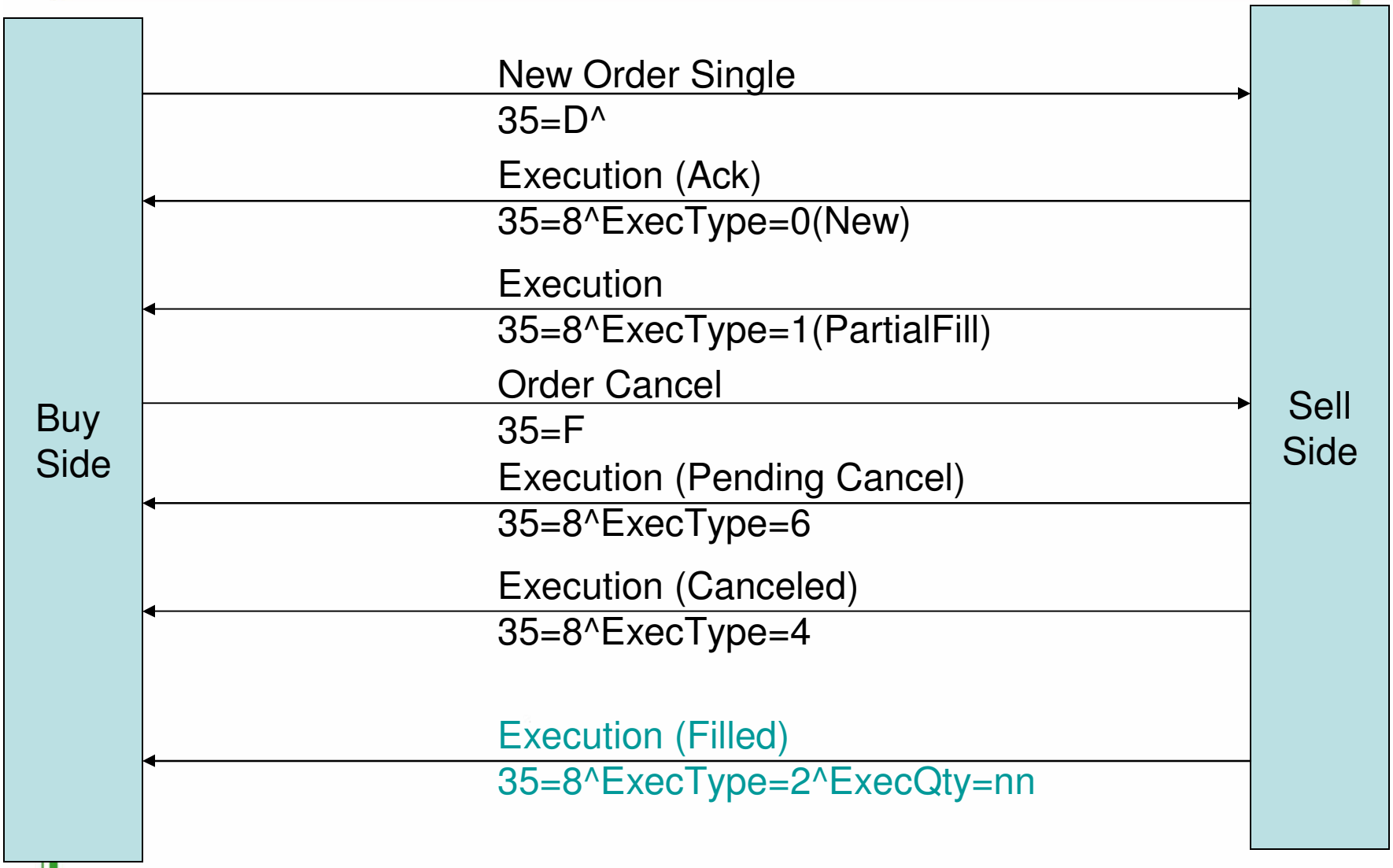
OrdStatus

- 0 = New
- 1 = Partially filled
- 2 = Filled
- 3 = Done for day
- 4 = Canceled
- 5 = Replaced
- 6 = Pending Cancel (e.g. result of Order Cancel Request)
- 7 = Stopped
- 8 = Rejected
- 9 = Suspended
- A = Pending New
- B = Calculated
- C = Expired
- D = Accepted for bidding
- E = Pending Replace (e.g. result of Order Cancel/Replace Request)

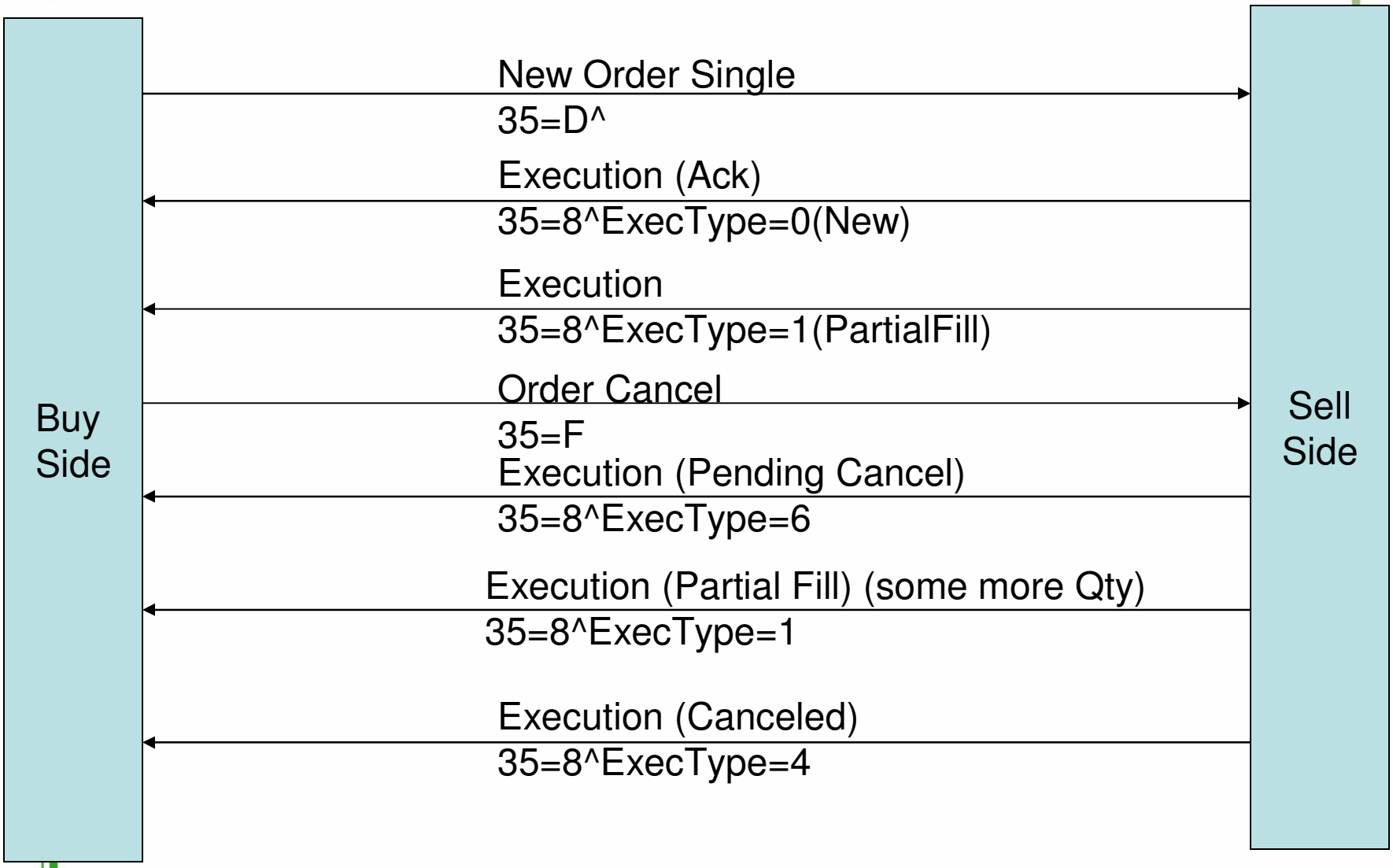
Scenario – 1 – Single Order



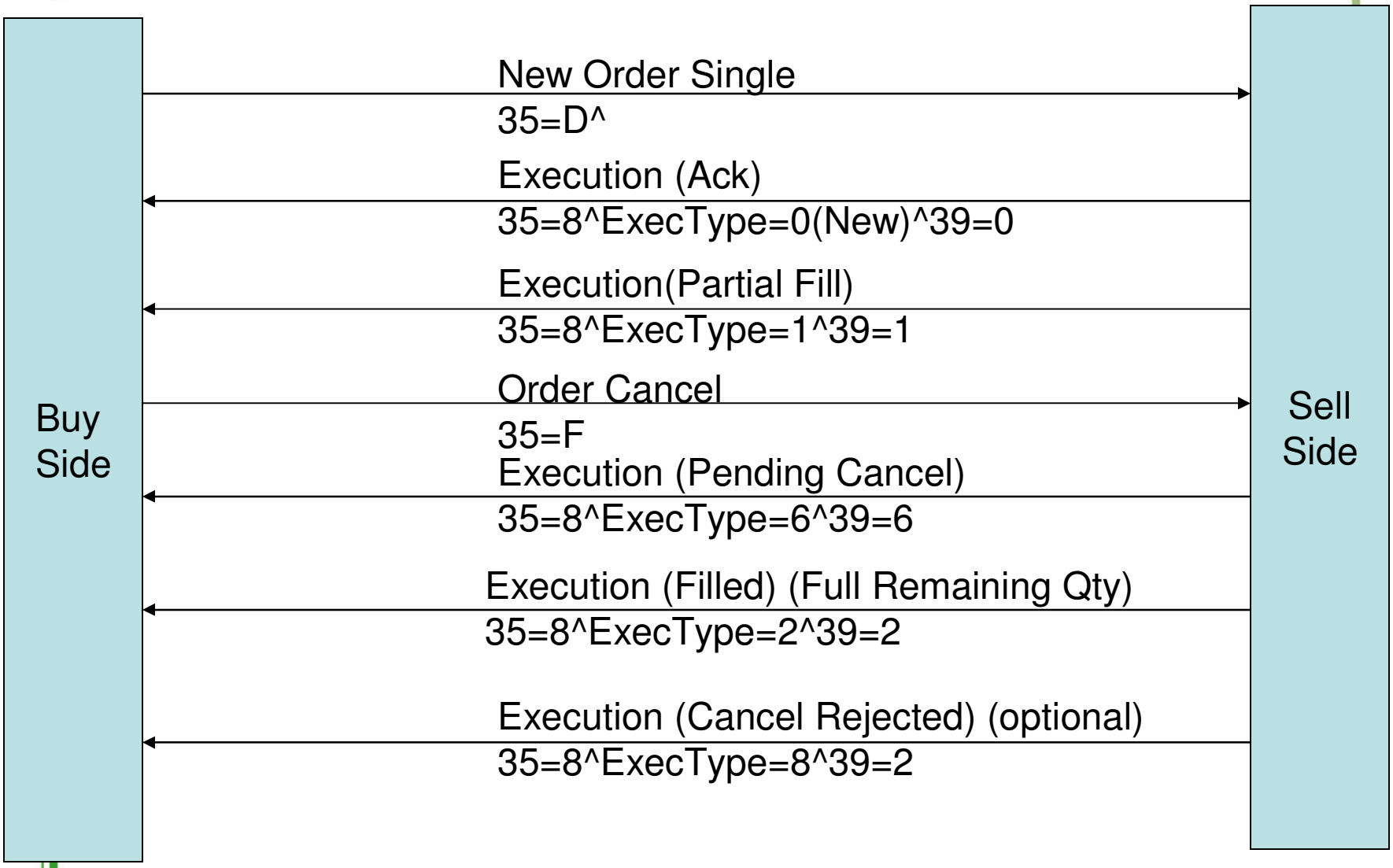
Scenario – 2 – Single Order



Scenario – 3 – Single Order



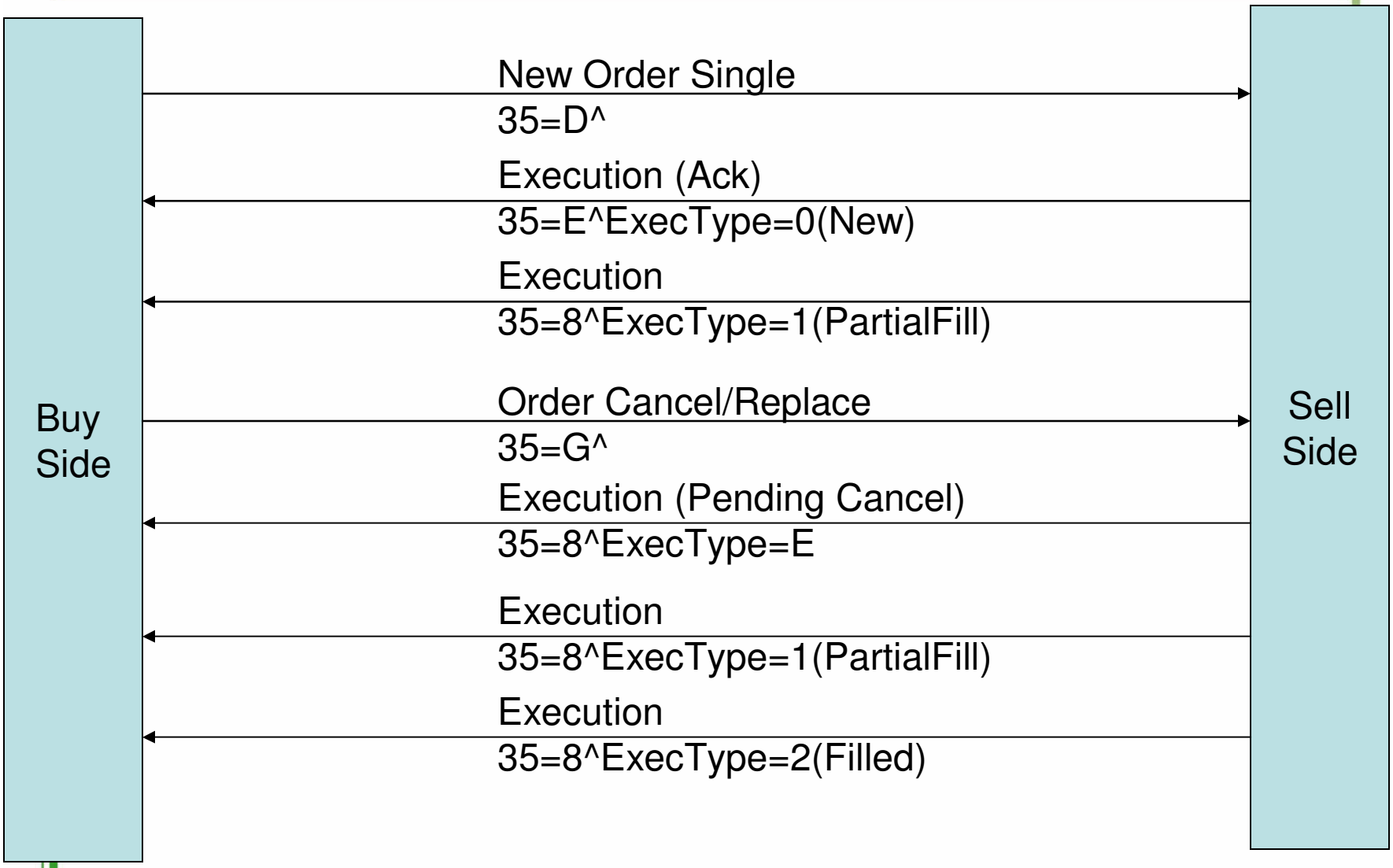
Scenario – 4 – Single Order



Order Cancel/Replace

- Also known as Order Modification
- Only modifiable properties can be changed
  - Order Qty, Order Price etc
- Filled order can also be reinstated by increasing the qty
- Used to increase the qty instead of sending new order
  - To holds the position / seniority

Scenario – 5 – Single Order



Key Fields

- OrderID – unique id used to identify the order
- ClOrdId – OrderID used to identify the order in all communication
- Order Type - Day Order Vs GTC (good till cancel)
- FK (FillOrKill) Vs IOC (Immediate or Cancel)
- Trade Bust – canceling previously issued execution (can be send by executing party)

ExecType Vs Order Status

- ExecType(150) – States the Execution Message type
  - New, Filled etc
  - Response to the request
- OrdStatus(39) – States the current orders status
  - New, Filled etc
  - May hold the same value as ExecType

Allocation

- Allocation and AllocationAck
- Used for
  - Distribution of executions/Orders among clients
  - Commission Calculations etc
- Also used in pre-trade to transmit the client allocation information

Part - 3

FIX Protocol Technical Section

By

Khader Shaik

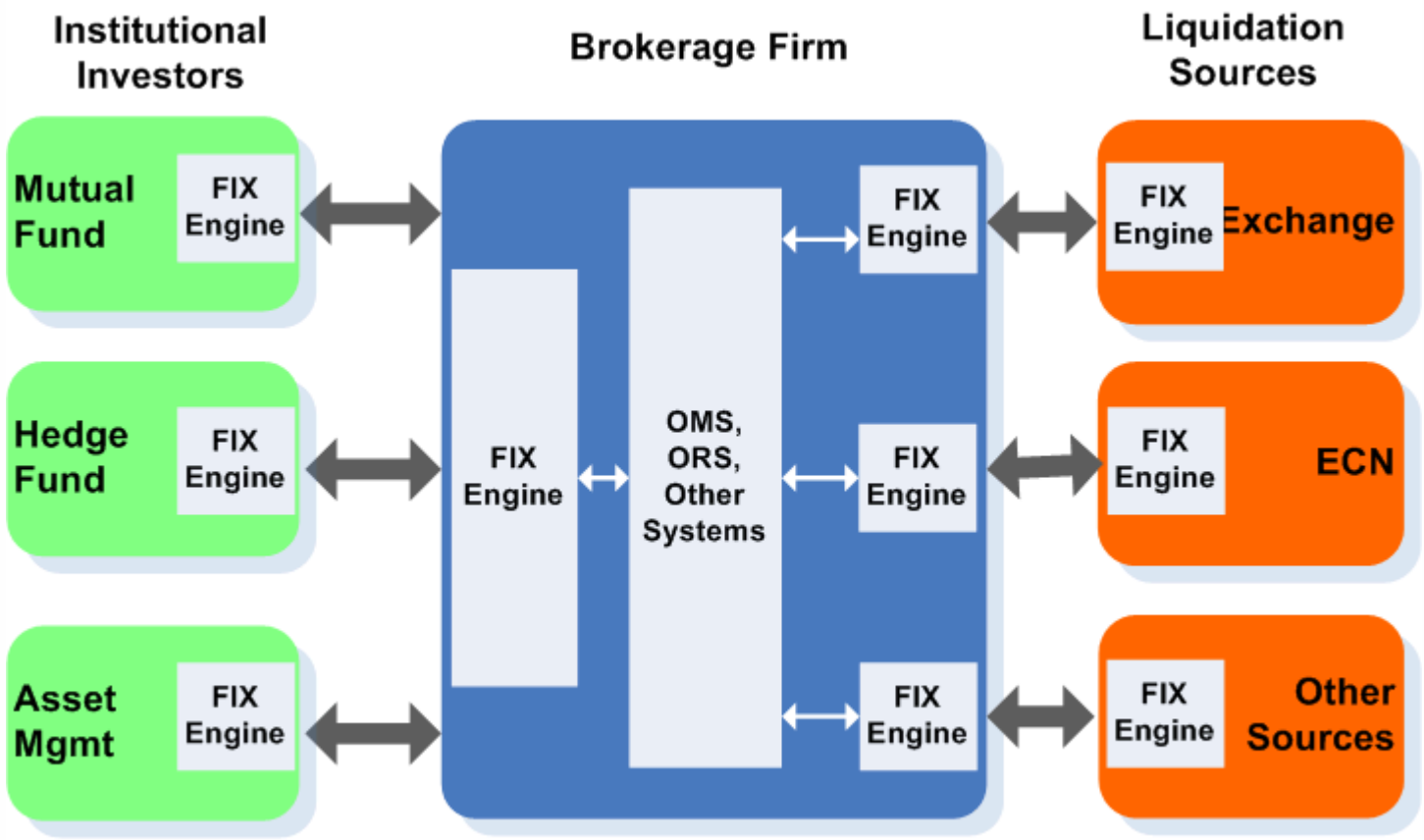
Agenda

- Architecture
- Buy-side Vs Sell-Side
- Integration Methods
- Commercial Engines
- Free Engines
- Implementation Steps
- Testing Tips
- Troubleshooting Production Issues
- Roles & Responsibilities

Architecture

- Client Side /Buy Side
  - OMS, Order Routing System, Trading Desks etc
  - FIX Engine
- Server Side / Sell Side / Exchange
  - FIX Engine
  - OMS, Order Matching Engine / Execution Engine
  - Settlement/Back-office System etc

Architecture



Basic Architecture

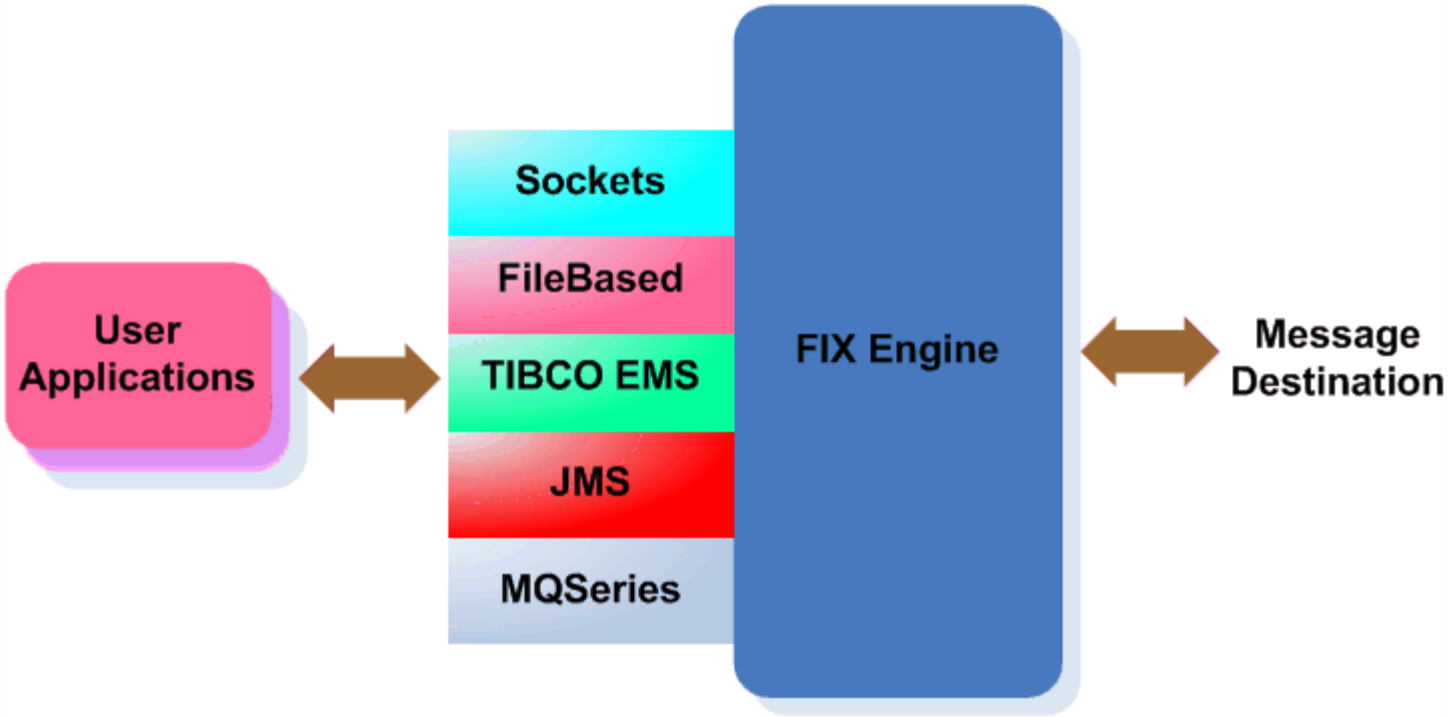
Commercial FIX Engines

- NYFIX – Appea
- Aegisoft – Aethna
- Reuters – Traid
- Financial Fusion (Sybase) - TradeForce
- CameronFIX (Orc Software)
- Other Major vendors support as part of their trading platforms
  - Fidessa
  - GL Trade etc
- QuickFix – Open Source FIX Engine (C++/Java)

Integration

- MQSeries
  - Input Queue
  - Output Queue
- Tibco Messaging (Rendezvous/EMS)
- Sockets
- JMS, RMI, .Net remoting
- Application (multi-threaded/different services)
  - Sender Module
  - Receiver Module

Integration



Pic: FIX Engine Integration Options

Implementation

- Trading Need– Requirement Identification
  - Financial Product
  - Supported Features
- Design the flow
  - Identify the messages
- Develop/Implement the systems
- Perform Internal testing
- Perform UAT with other party
- Rollout & Pre-Production testing

UAT

- Testing with other party testing is the must
  - Each client Implementation could be different
  - Messages are built using multiple tags, hence every tag is important
  - Create all scenarios and test

Production Issues and Troubleshooting

Message Validation

- Session Level
  - Invalid Protocol Level data
  - Eg: Invalid Message Type, Invalid TargetCompId etc
  - Rejected by FIX Engine
  - Will not be send to application
  - REJECT FIX message will be sent back
- Users may not see any sign; watch log files
- File watchers are used to monitor such errors

Message Validation

- Application Level
  - Validated by application (not FIX Engine)
  - Any missing or invalid custom fields
  - Rejected by application
  - BusinessReject FIX message will be sent back as response with a reason
- Applications must process these messages and appropriate actions
  - Displaying in the front end or send emails etc

Message Recovery

- Rarely happening situations, but expect couple of times a year at least
  - Transmission problems may occur some times
  - Messages may go out of sequence
  - FIX Engine will keep rejecting messages until sequence is fixed – serious issues
  - Usually manual intervention required to coordinate and fix the issues
- Very important to have measures in place to identify the situation the earliest possible
- Since its between FIX Engines, client application may not see any problem other than no flow of messages
- Using File Watchers is a standard solution

File Watchers

- Simple File Watcher – Shell or Batch script looking for certain text patterns in FIX Engine Log file
- Commercial Tools (Veritas and other) to monitor process log files

Application Level Rejects

- Developers must take care to avoid as much as possible
- Thorough UAT is required to identify these scenarios
- Must handle it gracefully
  - Generate BusinessReject message with clear RejectReason and send out to other party
  - Inform to the message originator the best possible way
  - Try to avoid dropping Business Reject Messages

Process Monitoring tools

- Veritas
- Tivoli
- CA – scheduler etc

Roles & Responsibilities

- Business Analysts
- Developers
- QA
- UAT
- Production Support – Level 1
- Production Support – Level 2

Business Analyst

- Understanding of Product Trading
- Understanding of the FIX protocol Design
- Understanding of the each FIX message and the usage of tags
- Development of detailed Test Cases
- Helping the Implementation/Development Team
- Coordination of UATs etc

Developers

- Must understand the Integration with FIX Engine
- Understand the various messages and tags to be used
- Flow of messages in various trading scenarios

QA

- Must be as good as BAs
- Must understand the operations of FIX Engines
- Must read and understand the log files
- Must be able to interpret format and different tag combinations etc

UAT

- Business Users
- Must understand the Trading (scenarios)

Production Support – Level 1

- Receives Calls from users
- Try to troubleshoot and identify the issues
- Pass on the issue to Level 2 as needed

Reading FIX Specification Doc

- FIX Version 4.2
  - One Volume
- FIX Version 4.3 and above
  - Multi-volume
- Walk-through of one of these version for basic understanding
- Current FIX Version is 5.0

Thank You

Khader Shaik

khader@orbitra.com

or

khaderv@yahoo.com

www.ksvali.com