

FINANCIAL MARKET INFRASTRUCTURE

EXCHANGE CONCEPTS

PART II - Orders & Matching

A conceptual guide to financial exchange architecture, order books & matching engines.

46	273	10	NO
CHAPTERS	GLOSSARY TERMS	REAL EXCHANGES	MATHS REQUIRED

J. Persson

v1.10.1 · 2026

Exchange Concepts An Introduction
Book Version: 1.10.1
Copyright © Johan Persson
Generated on October 16, 2026
Prepared with LaTeX from Markdown sources for duplex printing. Sources for documentation available at
https://github.com/johan162/EduMatcher

Exchange Concepts

An Introduction for Beginners

Johan Persson <johan162@gmail.com>

Generated on October 16, 2026

Overview

Preface to the Second Edition ix

Preface to the First Edition xi

I Part II: Orders, Matching, and the Trading Day 1

1 The Order: The Fundamental Unit 5

2 Order Types, The Vocabulary of Intent 7

3 Time-In-Force, How Long Should the Order Live? 13

4 The Order Book, The Exchange's Memory 17

5 Price-Time Priority, The Fairness Rule 27

6 Tick Sizes and Fractional Ticks 31

7 The Matching Engine, The Heart of the Exchange 35

8 The Life of a Trade 41

9 Market Makers, The Providers of Liquidity 45

10 The Opening and Closing Auction 51

11 Trading Sessions, The Day in the Life of a Market 57

12 Putting It All Together 61

13 Implied Orders and Synthetic Liquidity 63

14 Multi-Layer Implied Orders and the Combinatorial Wall 77

15 Indexes, The Market's Single Number 85

16 Benchmark Integrity: LIBOR, the FX Fix, and the Anatomy of a Riggable Number 93

Glossary **97**

References **117**

Exchanges own technical references **121**

Back-Cover Text **125**

About the Author **127**

Contents

Preface to the Second Edition ix

Preface to the First Edition xi

I Part II: Orders, Matching, and the Trading Day 1

1 The Order: The Fundamental Unit 5
1.1 Round Lots and Odd Lots . . . 6

2 Order Types, The Vocabulary of Intent 7
2.1 Limit Orders . . . 7
2.2 Market Orders . . . 7
2.3 Stop Orders . . . 8
2.4 Stop-Limit Orders . . . 8
2.5 Trailing Stop Orders . . . 9
2.6 How a Trailing Stop Actually Executes . . . 9
2.7 Iceberg Orders . . . 9
2.8 Hidden Liquidity, Priority Rules, and Midpoint Orders . . . 10
2.9 OCO Orders (One-Cancels-Other) . . . 12
2.10 Combo Orders . . . 12

3 Time-In-Force, How Long Should the Order Live? 13
3.1 DAY . . . 13
3.2 GTC, Good-Till-Cancelled . . . 13
3.3 GTD, Good-Till-Date . . . 13
3.4 ATO, At-The-Open . . . 13
3.5 ATC, At-The-Close . . . 14
3.6 IOC, Immediate-Or-Cancel . . . 14
3.7 FOK, Fill-Or-Kill . . . 14
3.8 TIF Quick Reference . . . 14

4 The Order Book, The Exchange's Memory 17
4.1 A Note on Implementation . . . 17
4.2 The Data Structure in Detail: Three Operations, Step by Step . . . 19
4.3 The Spread . . . 22
4.4 Depth . . . 22
4.5 Measuring Depth . . . 23
4.6 Price Levels . . . 23
4.7 The Order Book Is Not the Market Price . . . 24
4.8 What the World Sees vs What the Engine Knows . . . 25

5 Price-Time Priority, The Fairness Rule 27
5.1 How Order Amendments Affect Priority . . . 28

5.2 Ticks: The Minimum Price Increment . . . 28

6 Tick Sizes and Fractional Ticks **31**
6.1 Recap: Why Integer Tick Counts . . . 31
6.2 Tick Size Is Not One Number Per Market . . . 32
6.3 Sub-Penny Quoting and Rule 612 . . . 32
6.4 Fractions, Not Decimals: A Short History . . . 32
6.5 The Tick Size Pilot Program (2016–2018) . . . 33
6.6 A Worked Example: Rounding a Midpoint to a Valid Tick . . . 33

7 The Matching Engine, The Heart of the Exchange **35**
7.1 The Core Loop . . . 35
7.2 The Sweep . . . 37
7.3 Single-Threaded by Design . . . 38
7.4 One Book Per Symbol . . . 38
7.5 The Role of Data Structures . . . 39

8 The Life of a Trade **41**
8.1 Order Submission . . . 41
8.2 The Match . . . 41
8.3 Partial Fills . . . 42
8.4 Publication . . . 42
8.5 Order Status Lifecycle . . . 42

9 Market Makers, The Providers of Liquidity **45**
9.1 Market Maker Obligations . . . 45
9.2 The Two-Sided Dilemma: When One Side Is Taken . . . 46
9.3 Quote Refresh Policies . . . 46
9.4 Market Maker Protection (MMP) . . . 46
9.5 The Full Lifecycle of a Market Maker Quote . . . 48

10 The Opening and Closing Auction **51**
10.1 The Problem an Auction Solves . . . 51
10.2 Finding the Equilibrium Price . . . 51
10.3 Auction-Only Order Types . . . 52
10.4 The Named Crosses: Opening Cross and Closing Cross . . . 53
10.5 The Closing Auction . . . 54
10.6 Manipulating the Close, and Why the Auction Is Policed . . . 54

11 Trading Sessions, The Day in the Life of a Market **57**
11.1 Pre-Open / Pre-Market . . . 57
11.2 Opening Auction . . . 57
11.3 Continuous Trading (the main session) . . . 57
11.4 Intraday Auction (optional) . . . 57
11.5 Post-Close . . . 58
11.6 State Machine . . . 58

12 Putting It All Together **61**

13 Implied Orders and Synthetic Liquidity **63**
13.1 Level 0: Futures in One Page . . . 63
13.2 Level 1: The Spread Instrument and Its Sign Convention . . . 64
13.3 Level 2: The Developer's Model, Orders as Vectors . . . 65
13.4 Level 3: Implied OUT, a Complete Worked Example . . . 67
13.5 Level 4: Implied IN, Two Outrights Produce a Spread Quote . . . 68
13.6 Level 5: The Promise Kept, Implied Orders Create No Liquidity . . . 69
13.7 Level 6: Quantity Rules, min(), Shared Legs, Depth, and Recomputation . . . 69
13.8 Level 7: Priority, Real Orders Before Implied Orders . . . 71

13.9 Level 8: Price Improvement and Closing Cross-Book Gaps . . . 71
13.10 Level 9: Tick Alignment and Leg Prices, Where Arithmetic Meets Reality . . . 72
13.11 Level 10: How Implied Liquidity Appears in Market Data . . . 73
13.12 Engineering Deep-Dive . . . 73
13.13 Real-World Implementations . . . 75
13.14 Self-Check Exercises . . . 75

14 Multi-Layer Implied Orders and the Combinatorial Wall **77**
14.1 Generations: Counting the Links in the Chain . . . 77
14.2 A Real Multi-Generation Example (ICE) . . . 77
14.3 The Graph View: Implied Pricing Is a Path Problem . . . 79
14.4 The Combinatorial Wall . . . 80
14.5 How Real Venues Draw the Line . . . 82
14.6 What This Means When You Build One . . . 82
14.7 Self-Check Exercises . . . 83

15 Indexes, The Market's Single Number **85**
15.1 What an Index Is, and Why Exchanges Maintain Them . . . 85
15.2 How Indexes Are Weighted . . . 85
15.3 Corporate Actions and the Divisor . . . 87
15.4 Getting Into the Index: The S&P 500 Example . . . 88
15.5 The Consequences of Being Included . . . 89
15.6 Why Correct Index Calculation Is Critical . . . 89
15.7 What to Watch Out For . . . 90
15.8 Indexes in Risk Control . . . 90
15.9 Key Takeaways . . . 91

16 Benchmark Integrity: LIBOR, the FX Fix, and the Anatomy of a Riggable Number **93**
16.1 What Makes a Benchmark Riggable . . . 93
16.2 Case One: LIBOR, a Benchmark Built on Self-Reported Estimates . . . 94
16.3 Case Two: The 4pm FX Fix, a Benchmark Built on a Narrow Window . . . 95
16.4 What the Developer Should Take From This . . . 95
16.5 References . . . 96

Glossary **97**
A . . . 97
B . . . 98
C . . . 99
D . . . 101
E . . . 102
F . . . 102
G . . . 103
H . . . 104
I . . . 104
K . . . 105
L . . . 106
M . . . 106
N . . . 108
O . . . 108
P . . . 109
Q . . . 110
R . . . 110
S . . . 111
T . . . 113
U . . . 114
V . . . 114
W . . . 114
Z . . . 115

References **117**

Exchanges own technical references **121**
CME Globex Matching Engine & Implied Logic 121
Cboe US Options Complex Book Architecture 121
Intercontinental Exchange (ICE) Order Books 121
NYSE Pillar (Equity Trading Platform) 122
Further Sources Cited in the Text 122
Notes on Sources Not Cited Inline 123

Back-Cover Text **125**

About the Author **127**

Preface to the Second Edition

This book is still for the same person: the developer who gets dropped into an exchange codebase full of words like “NBBO”, “LULD”, “implied in/out”, “ATC”, “SIP”, “MMP”, and quietly wonders whether they have accidentally joined a cult. You have not. You have joined a domain with very old vocabulary, very modern latency budgets, and very expensive failure modes.

The first edition focused on giving the reader a map. This second edition keeps the same promise, almost no code-first deep dives, no equations for their own sake, just concepts that make the system legible, but the map is now wider, denser, and more practical. Compared with the first edition, this edition adds well over a thousand lines of new material and rewrites key chapters that were a bit too brief in the first pass where obviously my aspiration to keep it short sometimes conflicted with the need for completeness and understandability. Some pseudo-code has in addition been added where it helps clarify the mechanics, but the focus is still on concepts, not implementation. The goal is still to give you a mental model that lets you reason about the system, not just translate jargon.

The largest expansion is in the implied-orders material, which has been rebuilt from a short overview into two full chapters placed at the end of Part II, once the order book and the matching engine are familiar. The first is a complete walkthrough: the spread sign convention, a vector model from which every implied price formula follows, quantity, depth and priority rules, tick alignment, market-data publication, a first-generation algorithm, engineering invariants, and self-check exercises with worked solutions. The second follows chains of implied orders across many contract months and shows why every venue must limit how far implication is allowed to reach. The order-book chapter has also been expanded from concept to implementation shape: cache behavior, memory layout, single-threaded matching partitions, pool allocation, and step-by-step insert/cancel/sweep mechanics, including determinism concerns.

Several missing bridge topics are also now included explicitly. Part I this time around closes with listing (including the part black-magic art of IPO valuation) and delisting mechanics (initial and continued standards, cure periods, reverse splits, voluntary delisting, direct listings, and SPAC pathways). Part II adds round-lot vs odd-lot treatment and a dedicated tick-size chapter (Rule 612, historical fraction pricing, Tick Size Pilot, and midpoint rounding). The auction material has been upgraded with auction-only order types, opening/closing cross mechanics, and manipulation-at-the-close context.

Beyond core matching, this edition also adds the operational and structural topics developers eventually run into: benchmark integrity (LIBOR and FX fix case studies), trade busting and clearly erroneous handling, conformance testing and onboarding drift, market-data economics (SIP vs proprietary feeds, locked/crossed markets), fixed-income microstructure, crypto venue differences, and options expiry mechanics (exercise, assignment, settlement type, and pin risk).

So yes, the document is longer now. That is deliberate. The first edition tried to keep every section short; this one tries to keep every section useful when things get messy in production. If the first edition was the “day one” guide, this one aims to be the “month six” guide, the point where you are no longer asking what a term means, but why the system was designed that way and what breaks if it is wrong.

As before, this is an introduction, not a rulebook. Exchange rulebooks and venue technical specifications remain the final authority. But if this second edition helps you move from terminology to reasoning faster, it has done its job. The reference section includes the publicly available reference material many details in this books draws from to be as exact as possible.

Reasonable efforts have been made to ensure accuracy, but, it would be foolish of me to believe that I during some late night/early morning while editing this book didn't swap some bid/ask columns or simply forgot a whole section I had in my notes or my mind.

Enjoy the pleasure of finding things out and the triumph of being sure that the author was proveable wrong! So, any typos, missing signs, bad explanations/grammar or faults otherwise, are mine and mine alone. I have no-one to blame!

If you find any mistakes or have other suggestions, please let me know at *johan162@gmail.com*. I genuinely appreciate it.

Johan Persson, Second Edition, Autumn 2026, Järnboås Socken

Preface to the First Edition

This is a book for anyone who finds themselves having been asked to work on a financial exchange system, and yet, has no financial background. Scary. The codebase is full of words like "bid," "ask," "FOK," "the book," "OCO," "circuit breaker," "drop copy," and "kill switch," and many more. These are not arbitrary names; each one represents a concept that evolved over decades of real-world market operation, regulation, and hard-won lessons about what can go wrong when large amounts of money change hands at high speed. You just happened to be "unlucky" enough to start when all these terms have already been defined. Had you instead started 50-ish years ago the terms did not yet exist and you would have seen them come to life organically.

My goal with this book is simple: give you the conceptual vocabulary so that when someone says "GW01 submitted a GTC iceberg order that triggered a circuit breaker halt," you nod instead of panic. No code. No math. Just the map of the territory your code inhabits.

I have tried to keep things brief: each concept gets only as much space as it needs to be useful, and not a page more. Where I have necessarily oversimplified, you will find pointers at the end to the proper academic treatments (hundreds of pages of the hard stuff) for those rare evenings when you feel brave (or masochistic) enough to dive deeper. I make that promise with a clear conscience, because I spent rather too many evenings, and far too many weekends, working through the classic finance texts listed in the references, which served as the primary sources for this little book. If I am honest, this booklet is really nothing more than my own "lecture notes" from all that reading, tidied up and handed to you so you can skip the long way round. In the reference section you can find micro-reviews of all the books I used and taken inspiration from. You can also discover my own personal favourite!

This is an introduction, not a specification. Real exchanges are far more complex, and their official rulebooks are the final word. Think of this as the document I wish someone had handed me on day one I started to work in this world. I hope it helps you get up to speed faster than I did.

Reasonable efforts have been made to ensure accuracy, but as usual any remaining errors are entirely my fault. If you find mistakes or have suggestions, please let me know at *johan162@gmail.com* I genuinely appreciate it!

Johan Persson, First Edition, Winter 2025/2026, Järnboås Socken

PART I

Part II: Orders, Matching, and the Trading Day

How orders work, how the matching engine processes them, and how a complete trading day unfolds from open to close.

Historic Notes

In the summer of 1929, **Jesse Livermore**, then the most famous speculator in America began quietly selling short. He had been watching the tape for weeks, reading the continuous stream of prices and volumes that printed on the stock ticker, and he had seen something that troubled him: large blocks of stock were appearing at the top of rallies, absorbed by relentless selling pressure that the public could not see. The great bull market felt unstoppable, but the tape told a different story. Livermore increased his short positions through September and October. When the crash came in late October 1929, he is widely reported to have made about $100 million, in 1929 dollars, in a matter of weeks [Richard Smitten, *Jesse Livermore: World's Greatest Stock Trader*, 2001]. His methods had already been immortalised, thinly disguised as the fictional "Larry Livingston", in Edwin Lefèvre's *Reminiscences of a Stock Operator* (1923), still one of the most widely read books about trading.
Livermore did not know about matching engines, FIX protocols, or price-time priority. But he understood, intuitively and empirically, exactly what this entire Part formalises: that every trade is the intersection of a buyer's intent and a seller's intent, encoded in an order; that the order book is a record of unresolved intentions; that the sequence and size of fills reveals information about who is doing what; and that the rules governing when and how orders match determine the character of the market.
Part II is the technical formalisation of what Livermore read in the tape.

Part Summary:

Move from concepts to mechanics: how participant intent is encoded in order types, how matching logic enforces fairness, and how the trading session progresses from open through close.

Learning Objectives:

- Read an order ticket and understand each field's execution implications.
- Compare major order types and time-in-force instructions by risk and behavior.
- Trace how price-time priority and order book state changes produce trades.
- Describe the end-to-end lifecycle of a trade across a full trading day.
- Compute, execute, and test implied (synthetic) orders across related futures books, and explain why implication depth must be limited.

Content:

- The Order: The Fundamental Unit
- Order Types, The Vocabulary of Intent
- Time-In-Force, How Long Should the Order Live?
- The Order Book, The Exchange's Memory
- Price-Time Priority, The Fairness Rule
- Tick Sizes and Fractional Ticks
- The Matching Engine, The Heart of the Exchange
- The Life of a Trade
- Market Makers, The Providers of Liquidity
- The Opening and Closing Auction
- Trading Sessions, The Day in the Life of a Market
- Putting It All Together
- Implied Orders and Synthetic Liquidity
- Multi-Layer Implied Orders and the Combinatorial Wall
- Indexes, The Market's Single Number
- Benchmark Integrity: LIBOR, the FX Fix, and the Anatomy of a Riggable Number

CHAPTER 1

The Order: The Fundamental Unit

Everything in an exchange system revolves around the **order**. An order is an instruction from a participant to the exchange: "I want to buy (or sell) a certain quantity of a certain instrument, subject to certain conditions."

Every order carries several key pieces of information:

Field	What it specifies	Notes
Symbol	Which instrument	"AAPL" = Apple; "ES" = E-mini S&P 500 futures. Each symbol has its own independent order book.
Side	BUY or SELL	Defines everything about how the order interacts with the book.
Quantity	How many units to trade	Typically a positive integer. The **lot** is the smallest unit an order can be submitted in; for US equities the matching engine will accept an order for as little as one share. Asian markets often require order sizes to be a whole multiple of a larger minimum lot, 100 or 1,000 shares. This minimum tradeable unit is a separate concept from the **round lot** used to define an odd lot, covered just below.
Price	Limit price (for limit orders)	Maximum a buyer will pay, or minimum a seller will accept. Market orders carry no price.
Time-In-Force	How long the order remains valid	So important it gets its own section below.
Arrival timestamp	When the exchange received the order	Recorded to nanosecond precision. Not just metadata, it is the tiebreaker in price-time priority.
Identity	Which gateway (participant) submitted it	Used for self-match prevention, kill switches, and regulatory reporting.

1.1 Round Lots and Odd Lots

Most reference to "the lot" assumes a **round lot**, the standard trading unit for a symbol (100 shares for most US equities, though this varies by price tier and market). An order for any quantity that is not a whole multiple of the round lot size is an **odd lot**, for example, an order for 37 shares of a stock whose round lot is 100.

Odd lots trade normally, the matching engine has no difficulty filling them, but they have a surprising history in US market data: for decades, odd-lot orders and executions were excluded entirely from the consolidated tape and from the NBBO calculation described in the *Smart Order Routing* section of Part IV. The reasoning at the time was that odd lots were assumed to come from small retail investors and were not representative of the "real" market price. This had a real consequence: a stock could have an odd-lot bid or offer that was better than the displayed round-lot NBBO, and that better price would simply not appear in public quote feeds. As retail trading in high-priced stocks grew (a single share of some stocks costs more than many investors want to commit per trade), odd lots became a much larger share of total volume, and regulators revisited this exclusion. Under the SEC's 2020 market-data-infrastructure rules, odd-lot information is being phased into the consolidated data feeds. For exchange developers, the practical lesson is that "the best displayed price" and "the best price actually available" have not always been the same thing, and reference data (round lot size per symbol) drives which orders count as odd lots and are handled differently by downstream reporting and NBBO logic.

CHAPTER 2

Order Types, The Vocabulary of Intent

The type of an order describes the conditions under which it should execute. Understanding order types is fundamental to understanding exchange system code, because the matching engine has different logic for each type.

2.1 Limit Orders

A **limit order** says: "I am willing to trade at this price or better, but no worse."

- A buy limit order at $150.30 says: "Fill me at $150.30 or cheaper, but never at $150.31 or higher."
- A sell limit order at $150.35 says: "Fill me at $150.35 or higher, but never at $150.34 or lower."

The word "limit" refers to the price limit the participant is imposing. If an incoming limit order cannot immediately find a counterparty at an acceptable price, it **rests** in the order book, waiting. Resting orders are also called **passive orders**, they are not actively seeking to trade; they are waiting to be found.

Limit orders are by far the most common order type in most markets.

Price Improvement

An important and often surprising behaviour: **a limit order executes at the best available price, not necessarily at its own limit price**. This is called **price improvement**.

Example: the best ask is $150.30 (someone is selling at $150.30). You submit a buy limit order at $150.40. Your order says "I am willing to pay up to $150.40", but since the best available seller is only asking $150.30, you trade at $150.30. You receive $0.10 per share of price improvement relative to your limit. Your limit of $150.40 was the *worst* price you were willing to accept, not the target.

This applies in both directions: - A buy limit at $150.40 crossing a $150.30 ask → fills at **$150.30** (better for the buyer) - A sell limit at $150.25 crossing a $150.30 bid → fills at **$150.30** (better for the seller)

The limit price is a *floor* for sellers and a *ceiling* for buyers. The actual execution price is the best price available at the time, which will always be at least as good as the limit.

Price improvement matters for execution quality analysis. In the US, SEC Rule 605 requires market centres (and, since the SEC's 2024 amendments, larger broker-dealers too) to publish monthly statistics on execution quality, including how often orders received price improvement versus the quoted price. Large retail brokers like Fidelity and Charles Schwab report that a significant percentage of their retail orders receive price improvement, particularly for small orders routed to market makers who can beat the NBBO [1].

2.2 Market Orders

A **market order** says: "Fill me immediately at whatever the current market price is." There is no price constraint. The exchange executes it against the best available resting orders immediately.

Market orders maximise the probability of immediate execution, but execution is still subject to available liquidity, the current exchange state, risk controls, and regulatory protections. A market order submitted during a trading halt, against an empty book, or at a price outside a circuit-breaker band will not fill. In normal continuous trading conditions, market orders can be treated as effectively execution-guaranteed, but not price-guaranteed. Market orders are used when certainty of execution matters more than certainty of price.

The slippage danger in thin markets. If you submit a market buy for 10,000 shares but only 100 shares are available at the best ask, your order sweeps through every available seller in order of price until 10,000 shares are filled. Each level you sweep through costs you more. In an extreme case, you may end up paying dramatically more than intended, this is called **market impact** or **slippage**.

The most consequential real-world example occurred on 6 May 2010, the **Flash Crash**. In the 13 minutes between 2:32 and 2:45:28pm Eastern time, the E-mini S&P 500 futures contract (then, as now, one of the most liquid futures products in the world) fell 5.1%, as selling swept through a book from which many buyers had temporarily withdrawn. The joint SEC/CFTC investigation found that a single mutual fund had started an algorithm to sell 75,000 E-mini contracts (about $4.1 billion) at a rate of 9% of the previous minute's volume, "without regard to price or time"; as the price fell, other algorithms also sold, creating a feedback loop. The fall stopped at 2:45:28pm, when CME's *stop logic* functionality paused E-mini trading for just five seconds, and by 3:08pm prices were back close to their pre-drop level [SEC/CFTC, *Findings Regarding the Market Events of May 6, 2010*]. Some individual equities briefly traded at $0.01 or $100,000 during the chaos, because market orders crossed nearly empty books [SEC/CFTC Flash Crash Report, September 2010]. The circuit breaker mechanisms introduced after 2010 were specifically designed to prevent a recurrence.

Because they have no price to wait at, market orders cannot rest in the book. On most equity venues, any part that cannot immediately fill (for example, because there are no sellers at all) is cancelled. Some futures venues instead protect the market order with a price limit: CME Globex, for example, converts market orders into limit orders at a "protection" price a set distance from the best price, and any unfilled remainder rests at that limit.

2.3 Stop Orders

A **stop order** sits dormant until the market price reaches a specified trigger level (the **stop price**). When triggered, it converts to another order type and enters the book.

- A **stop-loss order** is a sell stop: "If the price falls to $145, automatically sell for me." Used by investors to automatically exit a losing position and limit further losses.
- A **buy stop** triggers when the price rises to the stop price. Used to enter a position on upward momentum: "If the stock breaks above $155, buy it, that confirms the uptrend."

Stop orders do not sit in the regular bid/ask book, they are held in a separate dormant queue and injected into the book only when triggered. It is worth noting that stop orders are not universally supported natively at the exchange level: some exchanges handle stops internally as described here; others leave stop order logic to brokers or client-side systems, sending only the resulting limit or market order to the exchange once the trigger is reached. The exchange system you are building handles stops natively, this is a deliberate design choice.

2.4 Stop-Limit Orders

A **stop-limit order** is a stop order that, when triggered, converts to a **limit order** rather than a market order. This gives the trader price protection but introduces the risk of non-execution.

Example: A sell stop-limit with a stop price of $145 and a limit price of $144. When the market falls to $145, the order converts to a sell limit at $144. It will only execute at $144 or better. If the market drops sharply past $144 (a "gap"), the order sits unfilled. Compare with a plain stop order, which converts to market and fills at whatever price is available, guaranteed execution, but potentially at $140 instead of $144.

Neither is universally better; the choice depends on whether the trader prioritises execution certainty (stop-market) or price certainty (stop-limit).

2.5 Trailing Stop Orders

A **trailing stop** is a stop order with a twist: the stop price automatically adjusts as the market moves in a favourable direction, but freezes when it moves against the position.

Example: You bought shares at $150. You set a sell trailing stop with an offset (also called the **trail distance**) of $5.00. The stop starts at $145. If the price rises to $160, the stop rises to $155. If the price then rises to $170, the stop rises to $165. But if the price then falls back to $167, the stop freezes at $165, it cannot move down, and will trigger a sale if the price continues falling and reaches $165.

The mechanism that advances the stop in the favourable direction is called the **ratchet**, like a mechanical ratchet that only moves in one direction. The trail offset is stored as a fixed distance, and the ratchet advances by subtracting this offset from each new high-water-mark price.

2.6 How a Trailing Stop Actually Executes

This is a point of frequent confusion, so it deserves careful explanation.

The trailing stop is not a resting sell order in the book. Before it fires, it is completely invisible to the market, held in a separate dormant queue inside the matching engine, with no presence in the bid or ask side of the order book. It does not compete with other sell orders. It does not queue behind them. It simply waits, monitoring the last trade price as each fill occurs.

The question "how can a trailing stop ever execute if there is always a better sell order in the book?" contains a false premise: the trailing stop is not *in* the book. Other sellers at higher prices are irrelevant to it.

What happens when the stop triggers. When the last traded price falls to or below the stop price, the trailing stop converts into a **market sell order** and is injected into the book. A market sell order goes directly to the *buy side*, it matches against the best available bid. Other sell orders are completely irrelevant at this point. The market sell does not queue behind limit sells; it sweeps across to the buyers.

To make this concrete with the example above: the stop price has ratcheted up to $165. The price starts falling. At $165.50, nothing. At $165.01, nothing. At $165.00, the stop triggers. A market sell order is born and immediately matches against whatever buyer is sitting at the top of the bid queue. If the best bid is $164.95, the fill occurs at $164.95. The trailing stop has executed.

The common mental model that causes confusion. People sometimes imagine the trailing stop as a limit sell order sitting in the book at $165, waiting in the queue behind other sellers who are already there. Under that mental model, yes, it seems like it could never execute, because better-priced sellers would always be ahead of it. But that mental model is wrong. A trailing stop is a trigger and a market order, not a limit order competing for queue position.

The trade-off. Because the trailing stop fires as a market order, execution is guaranteed (assuming buyers exist) but price is not. In a fast-moving market the price may have dropped well below the stop price by the time the market order fills. This is the same gap risk that applies to all stop orders: the stop triggers at $165 but if the market is falling quickly and the best bid is $162, that is where the fill occurs.

Trailing stop logic is not universally implemented at the exchange level. On some venues it is handled client-side (the trading algorithm tracks the high-water mark and adjusts the stop price manually), on others broker-side, and on others natively within the matching engine. Wherever it is implemented, the ratchet behaviour and the dormant-then-market-order execution model are the same.

2.7 Iceberg Orders

An **iceberg order** (also called a **reserve order** or **hidden order**) is a large order that conceals most of its size. It shows only a small visible portion, the **peak** or **tip**, to other participants. When the visible portion is consumed by fills, the order automatically replenishes from its hidden reserve, showing a new peak.

Why would someone want this? If you need to buy 100,000 shares, showing the full size in the book signals your intention to the entire market. Other participants may raise their ask prices or front-run your order before you can fill it. By showing only 1,000 shares at a time, you hide your true intent and reduce your

market impact.

The trade-off is **queue priority**: each replenishment gets a fresh timestamp and goes to the back of the queue at that price level, rather than retaining the position of the original order. The exchange's other participants can see that an iceberg is present (the order keeps replenishing at the same price), but not how large the hidden reserve is.

Iceberg orders are widely used by institutional investors and market makers on exchanges like the LSE, Euronext, and Deutsche Börse.

2.8 Hidden Liquidity, Priority Rules, and Midpoint Orders

Different exchanges treat hidden liquidity differently, and the rules matter for developers building order management systems.

Displayed vs hidden priority: On some exchanges, fully displayed (non-iceberg) orders at a given price level have strict priority over hidden or iceberg orders at the same price, even if the iceberg arrived earlier. The reasoning: participants who take the risk of displaying their intentions publicly should be rewarded with better queue position. On other exchanges, FIFO applies equally to hidden and displayed orders, first in, first out regardless of visibility. Knowing which rule applies on a given venue determines how institutional participants choose between iceberg and fully disclosed orders.

Reserve refresh priority: When an iceberg replenishes, a new peak appears from the hidden reserve, most exchanges treat the replenishment as a new order submission for queue purposes: it goes to the back of the queue at that price. This "back-of-queue on refresh" rule means icebergs that replenish many times lose priority relative to new participants arriving at the same price. Some venues implement partial priority preservation, but "back of queue on refresh" is the most common.

Midpoint peg orders: A **midpoint peg order** is an order whose price continuously tracks the current mid price (the average of the best bid and best ask). It never displays at a price, it sits invisibly, adjusting to the mid in real time. It executes only when a counterparty arrives whose order is also willing to trade at the midpoint. Midpoint peg orders are common in dark pools and supported by some lit venues, most notably IEX.

Why would any counterparty accept this when there are better-priced limit orders in the book?

This is the question that stops most people when they first encounter midpoint pegs, and the answer dismantles the confusion immediately: *the midpoint is a better price than the quoted ask for a buyer, and a better price than the quoted bid for a seller.* Both parties benefit compared to a standard trade at the quoted prices. There is no sacrifice involved.

Consider the numbers:

	Price
Best bid (highest buyer)	$150.30
Best ask (lowest seller)	$150.35
Mid price	**$150.325**

A standard market buy fills at the **ask**: $150.35. A midpoint peg buy fills at the **mid**: $150.325. The midpoint peg buyer pays $0.025 *less* per share than if they had bought at the quoted ask.

A standard market sell fills at the **bid**: $150.30. A midpoint peg sell fills at the **mid**: $150.325. The midpoint peg seller receives $0.025 *more* per share than if they had sold at the quoted bid.

Neither party is giving anything up. Both are doing better than the quoted market price by half the spread, they are sharing the spread between them rather than each paying it in full to the market maker. This is not a compromise; it is a joint saving.

A worked example.

An institutional investor wants to buy 200,000 shares of AAPL. The book shows: - Best bid: $150.30 / 10,000

shares - Best ask: $150.35 / 10,000 shares - Mid: $150.325

Option A, aggressive market order: Fill the entire 200,000 shares by sweeping the ask side. The first 10,000 fill at $150.35, then the next level, and so on. The average fill price will be well above $150.35 due to market impact. Transaction cost: spread paid on every share, plus significant slippage.

Option B, midpoint peg in a dark pool: The buyer submits a midpoint peg buy for 200,000 shares in a dark pool. The order sits invisible, pegged at $150.325. Separately, a pension fund that wants to sell 200,000 shares submits a midpoint peg sell in the same dark pool.

The dark pool matches them: 200,000 shares trade at $150.325.

- Buyer paid $150.325 instead of $150.35, saving $0.025 × 200,000 = **$5,000**.
- Seller received $150.325 instead of $150.30, gaining $0.025 × 200,000 = **$5,000**.
- The spread of $0.05 was split equally. Total joint saving: $10,000 compared to trading at the quoted prices.

Neither party showed their intention in the lit book, avoiding the price impact of a visible 200,000-share order. Both traded at a price inside the spread. This is the proposition that makes midpoint pegs attractive.

Who uses midpoint pegs?

Midpoint pegs are primarily used by institutional investors, mutual funds, pension funds, hedge funds, who have large orders, are price-sensitive, and are not urgently time-pressured. They have enough time to wait for a natural counterparty at the midpoint. High-frequency traders and market makers generally do not use midpoint pegs because they need execution certainty, not price improvement at the cost of uncertain timing.

The trade-offs.

The critical limitation is that **there is no guarantee of execution**. A midpoint peg buy will only fill if a seller willing to trade at the midpoint appears. In a liquid market this may happen quickly. In a thin or one-sided market the order may wait indefinitely. If the market moves strongly against you while you wait, the price rises while you hold a midpoint peg buy, the opportunity to buy at the original mid price may have passed by the time a counterparty appears. The participant must accept that improved price comes at the cost of execution certainty.

The mid price itself also moves continuously. An institution submitting a midpoint peg at $150.325 may find the mid has moved to $150.50 by the time it fills, if the market has drifted. The pegged price tracks the mid; the participant does not control the final execution price precisely, only that it will be the mid whenever the fill occurs.

Midpoint pegs in dark pools vs lit venues.

In a dark pool, the entire book is hidden, so a midpoint peg buyer and a midpoint peg seller can find each other without either revealing their interest to the lit market. The dark pool operator runs a separate matching process against the midpoint.

On IEX (a lit exchange), midpoint pegs work slightly differently. The midpoint peg sits passively at the mid. An *aggressive* order, a sell order willing to accept a price at or below the mid, arrives from a participant routing to IEX. That aggressive sell is priced at or below $150.325 and matches the resting midpoint peg buy at the mid. The aggressive seller "crosses to mid" by accepting the mid price instead of demanding the full ask. They receive $150.325, still better than the bid ($150.30), and the midpoint peg buyer gets their fill at $150.325 instead of paying the ask ($150.35). IEX's speed bump (350 microseconds) is relevant here: it prevents fast-moving quotes from making the midpoint stale before the fill occurs, ensuring the mid price used in the match is genuinely current.

The developer perspective.

The matching engine must handle midpoint pegs as a separate priority queue that sits parallel to the regular price-time priority book. The mid price must be recalculated on every book update (every time the best bid or best ask changes). The midpoint peg queue must be checked against incoming orders and against each other. The fill price for a midpoint peg match is not a stored order price, it is computed dynamically at match time as (best_bid + best_ask) / 2, requiring care with rounding to the nearest valid tick. Because the sum of two tick-aligned prices is not always an even number of ticks, this rounding step has a specific, easy-to-get-wrong correct answer, worked through in full in the *Tick Sizes and Fractional Ticks* section below.

2.9 OCO Orders (One-Cancels-Other)

An **OCO order** is a pair of orders linked together by a rule: if either order fills (or is cancelled), the other is automatically cancelled.

Classic use case: You own shares bought at $200. You want to take profit if the price rises to $215, but also automatically cut your losses if it falls to $185. You submit: - Order A: Sell limit at $215 (take-profit) - Order B: Sell stop at $185 (stop-loss) - Linkage: these are an OCO pair

Only one of A or B will ever execute. Whichever triggers first cancels the other. This is called a **bracket order**, the position is "bracketed" between a profit target above and a loss limit below.

OCO orders are a standard feature of most professional trading platforms. Relatively few exchanges support them natively in the matching engine; more often the linkage is maintained by the broker or the trading front end, which cancels the surviving order itself when the other one fills. (That design has a race: for a brief moment both orders are live, and in a fast market both can fill. Native exchange support closes that gap.)

2.10 Combo Orders

A **combo order** (also called a **spread order** or **strategy order**) is a single instruction to simultaneously execute orders in multiple instruments. Each component is called a **leg**. [12]

Example: A **pairs trade**, buy 100 shares of AAPL and sell 50 shares of MSFT simultaneously, treating the combined position as a single trade. The trader believes AAPL will outperform MSFT relative to each other, and wants to be exposed only to the relative difference between them, not to overall market direction.

Combo orders are critical in derivatives markets. On CME, a futures trader might simultaneously buy a March contract and sell a June contract on the same underlying, a **calendar spread**. Eurex offers a rich set of combination strategies for options, including straddles (buy a call and buy a put at the same strike), strangles (buy a call and buy a put at different strikes), and butterflies (three strikes, two legs long and one short in a specific ratio).

The execution challenge is **leg risk**: if one leg fills and the other does not, the trader is left with an unintended one-sided position. Production exchange systems handle this with sophisticated combo matching engines; simpler systems accept the leg risk explicitly.

CHAPTER 3

Time-In-Force, How Long Should the Order Live?

Every order must specify how long it should remain active if not immediately filled. This is called **Time-In-Force (TIF)** and is a standard attribute on every order in every major exchange system.

3.1 DAY

The order is valid only for the current trading session. At the end of the session, it is automatically cancelled. This is the default and most common TIF.

3.2 GTC, Good-Till-Cancelled

The order remains active until it is completely filled or explicitly cancelled by the participant. It survives overnight, across weekends, and across holidays. This requires special handling by the exchange: GTC orders must be persisted to durable storage at the end of each session and reloaded at the start of the next.

The risk for the participant: a GTC order placed when a stock was at $100 might unexpectedly fill at a different point in the market cycle if the participant forgets about it.

3.3 GTD, Good-Till-Date

A variant of GTC with a specified expiry date and time. The order remains active until it is completely filled, explicitly cancelled, or the specified expiry date arrives, whichever comes first. On the expiry date the order is automatically cancelled regardless of fill status.

GTD is standard on most professional trading platforms and widely used by institutional investors who want the persistence of GTC without the open-ended risk of an order remaining active indefinitely. A fund manager expecting a news event on a specific date might place limit orders to accumulate a position and use GTD to ensure they expire automatically if not filled by the event date.

From an engineering perspective GTD behaves like GTC with an additional scheduled cancellation task: the exchange's session scheduler must track each GTD order's expiry and issue the cancellation at the right moment, even if no other event in the system triggers it.

3.4 ATO, At-The-Open

The order is valid only during the **opening auction** (the special matching procedure that establishes the first price of the day). If not filled in the opening auction, it is cancelled. On NYSE, these are sometimes called **Market-At-Open (MOO)** or **Limit-At-Open (LOO)** orders depending on whether they carry a price limit.

An important restriction: ATO orders submitted after the opening auction has concluded are rejected, the TIF cannot be satisfied after the moment it targets has passed. Some exchanges accept ATO orders throughout

the pre-open period; others impose a cutoff time before the auction begins.

3.5 ATC, At-The-Close

Valid only during the **closing auction**. Extremely common among institutional investors who need to trade at or near the official closing price, many funds are benchmarked against closing prices. On a typical day in 2024, the US closing auctions matched roughly 9% of the whole day's volume in a single instant [BMLL, *Into the Close*, 2024]; see the *Opening and Closing Auction* chapter.

Like ATO, ATC orders submitted after the closing auction has already run are rejected. Unlike DAY orders, ATC orders cannot accumulate position through the continuous session, they are specifically targeting the single closing price that emerges from the auction uncross.

3.6 IOC, Immediate-Or-Cancel

Fill as much as possible immediately; cancel any unfilled remainder instantly. Unlike a market order, an IOC can carry a limit price: "Fill whatever you can at $150.30 or better right now; cancel the rest." IOC orders never rest in the book.

3.7 FOK, Fill-Or-Kill

Fill the **entire** quantity immediately, or cancel the entire order without any partial fill. Used when partial fills are unacceptable, for instance, an arbitrage strategy that only works if the full quantity executes simultaneously across legs.

The exchange must verify available liquidity before executing any fills. In practice, the engine performs a **dry-run sweep**: it walks through the book checking whether the full quantity can be matched at acceptable prices, without committing any fills. Only if the full quantity is satisfiable does the engine execute the actual fills. If at any point the available depth is insufficient, including because part of the available liquidity belongs to the same participant (which SMP would cancel), the entire order is cancelled without a single fill occurring. This is more complex than a standard market order sweep: the engine must complete a full hypothetical assessment of the book before touching it.

IOC vs FOK in one sentence: IOC accepts a partial fill and cancels the rest; FOK requires a full fill or cancels entirely. Both execute immediately and neither rests in the book.

3.8 TIF Quick Reference

TIF	Rests in book?	Accepts partial fill?	Valid period	Typical use case
DAY	Yes	Yes	Current session	Most orders; default
GTC	Yes	Yes	Until cancelled	Long-term limit orders
GTD	Yes	Yes	Until specified date	Event-driven; expires automatically
ATO	Opening auction only	Yes	Opening auction	Participate in the open price
ATC	Closing auction only	Yes	Closing auction	Benchmark to the close

TIF	Rests in book?	Accepts partial fill?	Valid period	Typical use case
IOC	No	Yes (partial OK)	Immediate only	Aggressive, price-limited; remainder cancelled
FOK	No	No (all or nothing)	Immediate only	Arbitrage, multi-leg strategies

CHAPTER 4

The Order Book, The Exchange's Memory

The **order book** (also called the **limit order book** or **LOB**) is the central data structure of a matching engine [1] [2]. It is the live record of every resting order in the market, all the buyers waiting to buy and all the sellers waiting to sell, organised by price.

Figure 1: *The most important data structure in an Exchange - the book.*

> **Key idea:** The order book contains only *resting* orders, those waiting for a counterparty. The current "market price" is derived from the book (as the mid of best bid and ask) or from the last trade, not from a stored field.

Think of it as two sorted lists:

The bid side, all resting buy orders, sorted from the highest price (most attractive for sellers) down to the lowest. A buyer offering $150.34 is at the top of the bid side if no one else is offering more.

The ask side (also called the **offer side**), all resting sell orders, sorted from the lowest price (most attractive for buyers) up to the highest. A seller asking $150.35 is at the top of the ask side if no one is asking less.

What a real order book looks like. At any given moment, a simplified snapshot of the AAPL book might be:

Bid Qty	Bid Price		Ask Price	Ask Qty
2,000	$150.34	← **best bid** | **best ask** →	$150.35	1,500
1,500	$150.33		$150.36	2,800
3,200	$150.32		$150.37	1,000
800	$150.31		$150.38	4,200

The **spread** here is $150.35 − $150.34 = $0.01. The **mid price** is ($150.34 + $150.35) / 2 = $150.345. If a market sell order for 3,500 shares arrives, it sweeps bid size: 2,000 shares at $150.34 (exhausting that level), then 1,500 at $150.33 (also exhausting that level). The new best bid after the sweep is $150.32 with 3,200 shares remaining at that price level.

4.1 A Note on Implementation

It is probably safe to say that no other data structure in an exchange is as heavily optimised as the order book. A modern exchange may maintain tens of thousands of order books simultaneously (one per tradeable symbol) and process millions of operations per second across them. Shaving a microsecond (10^{-6}s) from each operation or reducing the per-order memory footprint by a few bytes can translate directly into measurable throughput and latency gains at scale. Understanding *why* involves looking at how software architecture is shaped by hardware constraints.

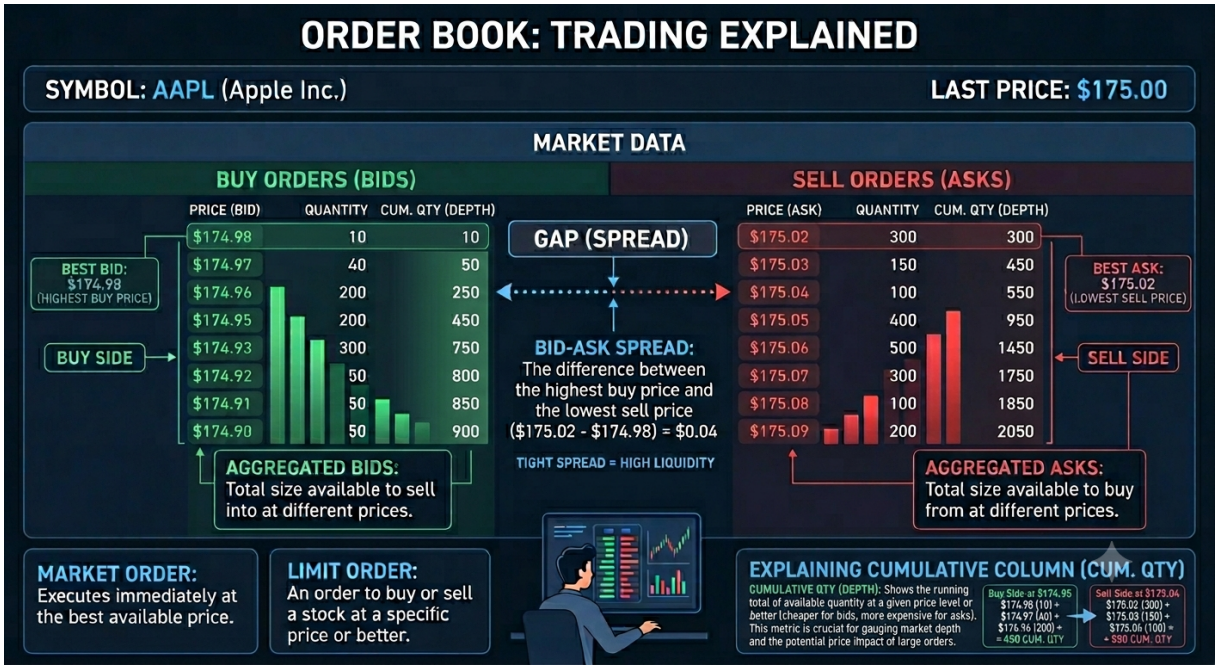


Figure 4.1: The Order Book

Principles of order book design

Constant-time best price access. The single most frequent operation is reading or modifying the best bid or best ask. Any design that requires traversal to find the top of book is immediately disqualified. Real implementations maintain direct pointers or indices to the best price level on each side, updated as levels are created or exhausted.

O(1) insertion at an existing price level. Once the correct price level is located, appending an order to the back of the queue at that level must be constant time. A doubly-linked list per price level is the classic choice: it gives O(1) append and O(1) removal by pointer (important for cancellations, which are the most common message type in modern markets).

Efficient price-level lookup. Finding the correct price level for a new limit order requires a structure keyed on price. Options include sorted arrays, red-black trees, skip lists, and direct-indexed arrays (when the price range is bounded and the tick size is fixed). Direct indexing by price offset is $O(1)$ and is preferred when applicable, at the cost of pre-allocated memory for the entire price range.

Minimise allocations on the hot path. Dynamic memory allocation (malloc/new) is unpredictable in latency due to fragmentation and system calls. High-performance engines pre-allocate pools of order objects and price-level nodes at startup, then dispense and recycle from the pool during trading, achieving deterministic allocation latency.

Aligning software architecture with hardware

Modern CPUs are fast enough that raw instruction throughput is rarely the bottleneck. Instead, the limiting factor is **memory access latency**: an L1 cache hit takes ~1ns, an L3 hit takes ~10ns, but a main-memory fetch costs 50–100ns. A single cache miss during a match can dominate total processing time. This hardware reality drives several architectural choices:

Cache-line-friendly data layout. Order book structures are laid out so that the data accessed together during a match (the top-of-book price level, the first few orders in the queue, the quantity and price fields) resides in adjacent cache lines. This often means using arrays of structs (AoS) or structs of arrays (SoA) tuned so that the hot fields pack into 64-byte cache-line boundaries.

Hot-path fits in L3 (or even L2). Engineers measure the working set of the critical matching path, the code and data touched for every single incoming order, and ensure it fits within the processor's L2 or L3 cache. If the hot path spills to main memory on every invocation, latency degrades dramatically. This constrains

both the code size (keeping the matching loop tight and branchless where possible) and the data footprint per book.

NUMA awareness. On multi-socket servers, accessing memory attached to a remote socket costs 2–3x more than local memory. Exchange engines pin each matching thread to a specific CPU core and ensure that the order books it manages reside in the same NUMA node's memory.

Branch prediction and prefetching. Critical paths are written to minimise unpredictable branches. Where future memory accesses are known (e.g. walking a price-level queue), software prefetch instructions are inserted manually so data arrives in cache before it is needed.

How real exchanges achieve speed

Single-threaded-per-book design. Rather than using locks to protect a shared book from concurrent access, most production exchanges assign each order book to exactly one thread (or one core). All messages for that symbol are routed through a single sequencer thread. This eliminates lock contention entirely, which is the single largest source of latency variance in concurrent systems.

Kernel bypass networking (DPDK / FPGA NICs). The operating system's network stack adds 5–15 s of latency per packet. Exchanges bypass the kernel entirely using user-space networking frameworks (like DPDK or Solarflare's OpenOnload) or offload protocol parsing to FPGA-based network cards. Messages arrive directly in user-space memory, often with hardware timestamps accurate to nanoseconds.

Busy-polling instead of interrupts. Rather than sleeping and waiting for an interrupt when no message is pending, the matching thread continuously polls the network ring buffer. This trades CPU power for lower latency: when a message arrives, processing begins within nanoseconds rather than waiting for an interrupt-to-thread-wake cycle (~2–5 s).

FPGA and ASIC acceleration. Some exchanges (and many trading firms) implement parts of the matching logic or the entire order book in FPGAs, achieving sub-microsecond matching latency. The trade-off is development complexity and reduced flexibility for protocol changes.

Huge pages and locked memory. Using 2MB or 1GB huge pages reduces TLB (Translation Lookaside Buffer) misses, which are another source of unpredictable latency. Critical memory regions are also locked (mlock) to prevent the OS from swapping them to disk.

Co-location and deterministic networking. Exchanges offer co-location services where participants place their servers in the same data centre, with equalised cable lengths to ensure fair, low-latency access. The exchange's own matching infrastructure is connected via cut-through switches with sub-microsecond forwarding latency.

The cumulative effect of these techniques is that a modern exchange can process an order, match it against resting liquidity, update the book, generate execution reports, and publish market data, all in well under 10 microseconds from the moment the network packet arrives.

4.2 The Data Structure in Detail: Three Operations, Step by Step

The principles above describe *what* properties the structure must have. This section shows *how* they fit together, by walking through the three operations that every order book must support, against one concrete arrangement of data structures. The goal is not to prescribe the only correct design, it is to make the abstract complexity claims ("O(1) cancel by pointer", "O(1) best-price access") tangible enough to implement.

The concrete layout. We assume the classic arrangement the previous sections described:

- Each **side** (bid, ask) owns a price-indexed map from price to a **price-level** object. For a bounded, fixed-tick instrument this map is a direct-indexed array (price offset → slot); for an unbounded one it is a balanced tree or skip list. Either way, we treat "find the level for price P" as its own step, and note its cost.
- Each **price level** holds a **doubly-linked list** of resting orders in strict time priority: head = oldest (front of queue), tail = newest (back of queue). It also caches its own `total_quantity` so depth queries need not walk the list.
- Each **side** caches a pointer to its **best** price level (`best_bid_level` / `best_ask_level`), so top-of-book

access never searches.

- A global **order-ID → order-node** hash map lets a cancel or amend find the exact node in O(1) without knowing its price. Each order node stores back-pointers to its price level and its neighbours in the linked list.

```
Side (ASK)
 ├── best_ask_level ──────────────┐
 ├── price_map: {                 ▼
 │     15035 ──► PriceLevel(150.35, total=1500)
 │                 head► [O:501 q=1000 t=09:31:02] ◄──► [O:502 q=500 t=09:31:40] ◄tail
 │     15036 ──► PriceLevel(150.36, total=2800)
 │                 head► [O:503 q=2800 t=09:30:59] ◄tail
 │   }
 └── ...

order_index: { 501 ► node@150.35, 502 ► node@150.35, 503 ► node@150.36, ... }
```

Every claim of speed in the previous sections is a claim about one of the three walk-throughs that follow.

Operation 1: Insert a resting limit order

A limit buy for 400 shares at $150.33 arrives and does **not** cross the ask (we handle the crossing case as Operation 3). It must be filed at the back of the queue for its price level, creating that level if it does not yet exist.

```
function insert_resting(order):                # order: {id, side, price, qty, ts}
    level = side(order.side).price_map.find(order.price)
    if level is null:
        level = pool.acquire_level(order.price)  # from pre-allocated pool, no malloc
        side(order.side).price_map.insert(order.price, level)
        update_best_pointer_on_insert(order.side, level)   # see note below
    node = pool.acquire_node(order)              # from pre-allocated pool
    list_append_tail(level, node)                # O(1): level.tail.next = node; ...
    level.total_quantity += order.qty            # keep cached depth correct
    order_index[order.id] = node                 # O(1) handle for later cancel/amend
    publish_market_data_add(order.price, order.qty)
```

Cost analysis, tying back to the design principles:

- `price_map.find` / `insert` is the *only* step whose cost depends on the price-map choice: O(1) for a direct-indexed array, O(log n) for a tree. Everything else is strictly O(1).
- `list_append_tail` is O(1) because the level caches its tail pointer, this is why time priority (append at back) is cheap while price improvement (a new best level) is handled purely by the best-pointer update.
- `update_best_pointer_on_insert` compares the new level's price to the current best on that side and, for a buy, replaces `best_bid_level` only if the new price is higher (for a sell, only if lower). This is the entire cost of a new order improving the top of book, one comparison, no search.
- Both `acquire_level` and `acquire_node` come from startup-allocated pools, honouring "no allocation on the hot path."

Operation 2: Cancel a resting order

Cancellation is the most common message type in modern markets, so it must be the cheapest. This is exactly what the order-ID index and the doubly-linked list buy us:

```
function cancel(order_id):
    node = order_index.get(order_id)
    if node is null: return REJECT_UNKNOWN_ORDER     # already filled/cancelled
    level = node.level
    list_unlink(node)                 # O(1): node.prev.next = node.next; node.next.prev = node.prev
    level.total_quantity -= node.qty
    order_index.remove(order_id)
    publish_market_data_delete(level.price, node.qty)
    if list_is_empty(level):
        side(node.side).price_map.remove(level.price)
        if side(node.side).best_pointer == level:
            recompute_best_pointer(node.side)   # only when the *best* level empties
        pool.release_level(level)
    pool.release_node(node)
```

The critical detail is `list_unlink`: because the list is *doubly* linked and the node holds pointers to both neighbours, removal is a couple of pointer writes with no traversal, regardless of where in the queue the order sat. A singly-linked list would force an O(queue-length) walk to find the predecessor, which is why the doubly-linked list is not a stylistic preference but a requirement. Note also that `recompute_best_pointer`, potentially the most expensive step, runs *only* when the level that emptied was itself the best on its side; a cancel deep in the book never touches the best pointer.

Operation 3: An aggressive order sweeps the book

Now the case that generates trades. A market (or marketable-limit) buy for 3,500 shares arrives against the ask side. It must consume resting liquidity from best price outward, in time priority within each level, until filled or until it runs out of book (or, for a limit, out of acceptable price).

```
function match_aggressive_buy(incoming):        # incoming: {qty, limit_price or ∞ for market}
    while incoming.qty > 0:
        level = ask_side.best_ask_level
        if level is null: break                       # book exhausted on this side
        if incoming.limit_price < level.price: break  # limit no longer marketable
        node = level.head                             # oldest order first: time priority
        while node is not null and incoming.qty > 0:
            traded = min(incoming.qty, node.qty)
            emit_trade(buyer=incoming, seller=node, price=level.price, qty=traded)
            incoming.qty     -= traded
            node.qty         -= traded
            level.total_quantity -= traded
            if node.qty == 0:                         # resting order fully filled
                next_node = node.next
                list_unlink(node)
                order_index.remove(node.id)
                pool.release_node(node)
                node = next_node
            else:
                break                                 # resting order partly filled: incoming is done
        if list_is_empty(level):
            ask_side.price_map.remove(level.price)
            pool.release_level(level)
            recompute_best_pointer(ASK)               # advance to next-best level
    if incoming.qty > 0 and incoming.is_limit:
        insert_resting(incoming)                      # unfilled remainder rests (Operation 1)
```

Trace it against the book from the start of this chapter, best ask \$150.35 × 1,500, next \$150.36 × 2,800:

1. `level` = \$150.35. `traded = min(3500, 1500) = 1500`. Trade printed at **\$150.35**. Incoming now needs 2,000; the level empties, is released, best pointer advances.
2. `level` = \$150.36. `traded = min(2000, 2800) = 2000`. Trade printed at **\$150.36**. Incoming is filled. The \$150.36 level keeps 800 shares and becomes the new best ask.

Two properties of this loop are worth making explicit because both were asserted earlier in the chapter without

proof:

- **Every fill in a sweep can print at a different price** (here $150.35 then $150.36). The single-price rule is a property of *auctions* (covered later), never of continuous sweeps, this loop is precisely where "market impact" and "slippage" come from.
- **The inner loop preserves time priority**: it always starts at `level.head`, the oldest order, and moves to the next order only when the one in front is completely filled. A partial fill of a resting order (`node.qty > 0` after trading) can only happen when the incoming order has run out of quantity, so the sweep ends there, and the partly filled order keeps its place at the front of the queue. If instead the sweep ends because the book or the limit price is exhausted, any remaining incoming quantity either rests (limit) or is dropped (market).

  Key idea: The three operations, insert, cancel, sweep, are the entire contract of an order book, and each one's cost is dominated by a single step: the price-map lookup on insert, the pointer unlink on cancel, and the level-by-level walk on sweep. Every optimisation in the preceding sections (best-price pointers, doubly-linked queues, the order-ID index, pooled allocation) exists to make exactly one of those steps constant time. If you can implement these three functions with the costs annotated above, you have implemented the core of a matching engine; everything else is order types, risk checks, and protocol.

A note on determinism

Nothing in the three functions above reads a clock, consults a random source, or depends on iteration order of a hash map during matching (the `order_index` is used only for direct point lookups, never iterated during a match). This is deliberate, and it is what makes the engine replayable in the sense the *Determinism, Replay, and Persistence* chapter requires: given the same book state and the same ordered input, these functions produce identical trades in identical order. Introducing, say, a `map` whose iteration order varies between runs, or reading wall-clock time to break a tie, would silently destroy that guarantee.

4.3 The Spread

The **spread** is the gap between the best bid (highest buy offer) and the best ask (lowest sell offer). If the best bid is $150.30 and the best ask is $150.35, the spread is $0.05.

The spread represents the immediate cost of trading: if you need to buy right now, you pay the ask price; if you need to sell right now, you receive the bid price. The round-trip cost of buying and immediately selling is the spread.

A **tight spread** (small gap) indicates a liquid, efficiently-priced market. A **wide spread** indicates illiquidity, less competition between market participants, higher trading costs. Market makers earn the spread: they buy at the bid and sell at the ask, pocketing the difference.

The **mid price** is the arithmetic average of the best bid and best ask: (150.30 + 150.35) / 2 = $150.325. This is often used as the "current price" of an instrument when no trade has occurred recently.

4.4 Depth

Depth refers to how much quantity is resting at each price level. A market with 50,000 shares resting within $0.05 of the best bid is "deep", you can trade a large size without moving the price much. A market with only 100 shares available near the best price is "shallow", a single large order will sweep through multiple price levels.

Level 1 data shows only the best bid price, best ask price, and quantities. **Level 2 data** (also called **market depth** or **market-by-price**) shows the aggregated quantity at several (sometimes all) price levels. **Level 3 data** (**market-by-order**) goes further and shows every individual resting order, still without revealing who placed it. Professional traders subscribe to depth data because it reveals information about near-term price pressure.

4.5 Measuring Depth

Depth is not a single number, it is a shape. Practitioners and algorithms use several derived measures to quantify it for different purposes.

Quantity at a price level. The simplest measure: the total resting quantity at a single specific price. If there are three sell orders at $150.35 for 200, 500, and 300 shares respectively, the quantity at $150.35 is 1,000 shares. This is what Level 2 data shows at each row. (Note that this is exactly the `level.total_quantity` field the implementation above maintains incrementally, precisely so this query never walks the queue.)

Cumulative depth within N ticks. More useful than a single level is knowing how much total quantity is available within a price range. If AAPL has a best ask of $150.35 and you sum all resting ask quantities from $150.35 to $150.45 (10 ticks), you get the total shares you could buy while moving the price at most 10 cents. A large cumulative depth within a few ticks indicates a resilient, liquid market; a small cumulative depth means a single large order will sweep many levels quickly.

Bid-ask imbalance (depth ratio). Compare the total resting quantity on the bid side to the total on the ask side, within some symmetric window around the mid price:

```
Imbalance = bid_depth / (bid_depth + ask_depth)
```

A value of 0.5 means the book is balanced, roughly equal buying and selling interest. A value near 1.0 means the bid side is much heavier: many buyers, few sellers. This is often interpreted as short-term upward price pressure. A value near 0.0 means the ask side is dominant: selling pressure. Market microstructure research consistently finds that order book imbalance is a short-term predictor of price direction. Many trading algorithms compute it as a continuous signal.

Market impact estimation. Given a target order size S, you can compute the *average price* you would pay by sweeping through the book level by level. If you want to buy 5,000 shares and the book is:

Ask price	Qty
$150.35	2,000
$150.40	1,500
$150.45	2,000

Your 5,000-share buy sweeps all three levels: 2,000 at $150.35, 1,500 at $150.40, 1,500 at $150.45. The volume-weighted average price is:

```
VWAP = (2000 × 150.35 + 1500 × 150.40 + 1500 × 150.45) / 5000 = $150.395
```

The **market impact** is the difference between this average and the initial best ask: $150.395 − $150.35 = $0.045 per share. Depth data lets a trader estimate their market impact before submitting, which is critical for execution strategy: split into smaller orders, use an iceberg, route to a dark pool, or simply accept the impact if time pressure is high. (This calculation is Operation 3 above, run as a read-only simulation against a copy of the book, rather than an actual match.)

Available depth at cost. The inverse of the above: given a maximum acceptable average price (or maximum price movement), how large an order can you execute within that budget? This is how automated execution algorithms compute optimal slice sizes.

Volume-at-touch vs total book depth. A useful distinction: *volume at touch* is only the best bid and ask (Level 1). *Total book depth* includes all visible levels. An iceberg order contributes only its visible peak to displayed depth, so total book depth may understate available liquidity if icebergs are present. This is why dark pool liquidity (invisible until matched) and iceberg reserves (invisible until refreshed) are relevant even to participants who believe they can read the full book.

4.6 Price Levels

A **price level** is a single specific price at which one or more orders are resting. All orders at $150.30 form one price level on the bid side. When all orders at a given price level have been filled or cancelled, that price

level disappears from the book. (In the implementation above, this is the `pool.release_level` step, and it is also the moment the best pointer may need to advance.)

4.7 The Order Book Is Not the Market Price

This is a subtle but important point: the order book shows only **resting orders**, orders that have not yet traded. The current market price, as quoted in news tickers and trading apps, is typically the price of the **most recent actual trade**, not the price of any resting order. After a trade happens, the market price updates. Between trades, the price is conventionally shown as the mid of the book.

This means there are actually several distinct "prices" in play at any moment, each used for a different purpose:

Last trade price. The price at which the most recent fill occurred. This is what scrolling tickers and trading screens display as the "current price" during the session. It updates with every fill, potentially many times per second in a liquid market.

Mid price. The average of the best bid and best ask: (best_bid + best_ask) / 2. Used as a proxy for fair value between trades, particularly when no trade has occurred for a while. Derived from the book, not from any actual transaction.

Previous day's closing price. Once the session ends, the **official closing price** from the closing auction becomes the reference price for the entire period the market is closed, typically overnight and across weekends. This is the price used to value portfolios at end of day, to calculate overnight P&L, to set the reference for the next day's static price collars, and to publish the figures that appear in newspapers, financial reports, and fund valuations.

> **Closing Auction and Static Price Collar**
>
> Two terms in that paragraph will be unfamiliar at this point in the document, so a brief preview is warranted.
>
> The **closing auction** is a special matching procedure that runs at the end of the trading day: rather than matching orders one at a time as they arrive, the exchange collects all outstanding buy and sell interest over a short accumulation period and then computes the single price at which the greatest number of shares can trade simultaneously, matching all eligible orders at that one price. This produces a more authoritative closing price than simply taking the last trade of continuous trading, which might have been a small or unusual transaction. The closing auction is covered in full in the *Opening and Closing Auction* section of Part II.
>
> A **static price collar** (also called a fat-finger filter) is a pre-trade risk control that rejects any incoming order whose submitted price strays too far from the previous closing price, protecting against obvious entry errors such as a misplaced decimal point. Because it uses the closing price as its benchmark, it must be recalculated at the start of each new session. Static price collars are covered in full in the *Trading Sessions* section of Part II.

The closing price carries particular weight precisely because it is independently determined by a transparent auction process rather than by a single trade that could be anomalous or thin. A portfolio worth $10 million at 3:59pm might be marked at a slightly different value at 4:00pm if the closing auction produced a different price, but that closing price is considered more authoritative because it reflects the broadest simultaneous expression of supply and demand at that moment of the day.

For exchange developers, the closing price has several concrete implications. It is the reference that the static price collar compares each new day's orders against. It is the benchmark that performance reports are measured against. It is the number that triggers overnight margin calls if positions have moved far enough. And it is the price that must be persisted at end of session, broadcast to all downstream systems, and made available when the exchange reopens the following morning.

4.8 What the World Sees vs What the Engine Knows

Most market participants see only an **aggregated view** of the book: total quantity at each price level, without knowing how many individual orders make up that quantity or who placed them. The exchange itself knows the full detail, every individual order, its owner, its arrival time, its type. Publishing the aggregated view is part of the exchange's **market data** service; it's how participants observe the market. (The engine's private, per-order detail is exactly the doubly-linked queue of nodes from the implementation section; the public aggregated view is the `total_quantity` per level. The two are kept consistent by updating the cached total on every insert, cancel, and fill.)

CHAPTER 5

Price-Time Priority, The Fairness Rule

One of the most fundamental questions an exchange must answer is: when multiple resting orders are at the same price, which one gets filled first?

The universal answer on mainstream exchanges is **price-time priority**:

1. **Better price goes first.** A buyer offering $150.40 gets priority over a buyer offering $150.30, because the seller gets a better deal. A seller asking $150.25 gets priority over a seller asking $150.35.
2. **At the same price, earlier arrival goes first.** Among all buy orders at $150.30, the one that arrived first gets filled first. This is **first-in, first-out (FIFO)** ordering within a price level.

This seems simple, but it has profound implications. It means:

- Participants are incentivised to quote good prices, better prices move you to the front of the queue.
- Participants are incentivised to act quickly, at any given price, being early is an advantage.
- It is completely transparent and deterministic, the exchange's matching rules are known in advance, equally to everyone.

Price-time priority is the default on NYSE, NASDAQ, CME, Eurex, LSE, and virtually every major exchange [1]. Price-time FIFO is dominant in equities markets, but derivatives markets sometimes use alternative allocation rules. The most common alternative is **pro-rata**, where fills at a price level are distributed proportionally to the size of each resting order. To make the difference concrete:

Example: 60 lots are available to sell at a price. Two buy orders are resting at that price: Order X for 100 lots (arrived first) and Order Y for 40 lots (arrived second).

- *Under FIFO:* Order X arrived first and has priority. It receives all 60 lots. Order Y receives nothing.
- *Under pro-rata:* The 60 lots are distributed in proportion to order size. Total resting demand = 100 + 40 = 140 lots. Order X receives 60 × (100/140) 43 lots; Order Y receives 60 × (40/140) 17 lots. Both orders fill partially.

Pro-rata rewards size over speed: a large order gets more even if it arrived later. It is common in interest-rate futures markets (where orders can be very large and the marginal value of nanosecond speed is lower) and in some options markets. CME Group, for example, matches its SOFR (Secured Overnight Financing Rate) futures with an "allocation" algorithm that is largely pro-rata (with priority for the order that first improves the price), matches 2-year Treasury note futures with a configurable FIFO/pro-rata blend, and uses plain FIFO for its equity index futures (E-mini S&P 500, E-mini Nasdaq 100), crude oil, and 10-year Treasury note futures [CME Group Client Systems Wiki, *Supported Matching Algorithms*; Databento, *CME matching algorithms explained*]. The algorithm is configured per product, and occasionally changes, so always look it up. Knowing which rule a venue uses matters enormously for anyone building order-routing or execution logic: an HFT firm optimising for FIFO markets (be first) must use entirely different strategies than one optimising for pro-rata markets (be large).

For the exchange to implement price-time priority correctly, two things must be true: prices must be comparable without ambiguity (hence integer tick counts), and order arrival must be sequenced deterministically. Modern exchange systems use **nanosecond-precision timestamps** as one component of this sequencing.

However, the timestamp alone does not fully guarantee fairness, what matters is the order in which messages are **accepted into the exchange system**, which also depends on network routing, gateway sequencing, and in high-performance systems, hardware timestamping infrastructure. The timestamp records when the exchange received the order; the sequencing infrastructure ensures that two orders arriving within the same nanosecond are ordered consistently and reproducibly.

5.1 How Order Amendments Affect Priority

Orders do not always stay unchanged from submission to fill. Participants modify their orders, adjusting price, changing quantity, or occasionally altering other attributes. The question of how an amendment affects queue priority is important, and the rules are consistent across almost all major exchanges.

The guiding principle: priority is earned by making a commitment at a specific price at a specific moment in time. When that commitment changes in a way that is beneficial to the participant at the expense of others in the queue, the priority is lost. When the change is a concession, giving up something, priority is retained.

Applied to the three common amendment types:

Price change → priority is lost. If a resting buy order at $150.30 is amended to $150.35, it receives a new timestamp and goes to the back of the queue at $150.35. The logic is straightforward: the order at $150.30 was competing for queue position against others who committed to $150.35 earlier. If it could simply "upgrade" its price while retaining the earlier timestamp, it would jump ahead of participants who made the $150.35 commitment first, which is unfair to them.

Example: At 10:00:00.000 you submit a buy limit at $150.30 and join the queue behind five other orders at that price. At 10:00:05.000 the market is moving and you amend your price up to $150.31. Your order now has a new timestamp of 10:00:05.000 and joins the back of the queue at $150.31, behind every order that was already resting at $150.31, including those that arrived between 10:00:00 and 10:00:05. Your five seconds of seniority at $150.30 count for nothing at the new price.

Quantity increase → priority is lost. If a resting order is amended upward from 500 shares to 1,000 shares, it receives a new timestamp and goes to the back of the queue. The logic: the order now claims twice as much of the available execution as it originally committed to. Participants who arrived later but were waiting behind the 500-share order should not now find themselves behind a 1,000-share version of the same order.

Quantity decrease → priority is retained. If a resting order is amended downward from 1,000 shares to 200 shares, the timestamp and queue position are unchanged. The logic: the order is conceding some of its claim, not expanding it. This is a concession that favours other participants (more liquidity becomes available behind it), so there is no fairness reason to penalise it.

These rules hold on NYSE, NASDAQ, CME, Eurex, LSE, and virtually every other major exchange. They are sometimes informally summarised as: *increasing your claim loses priority; decreasing your claim does not.*

Cancel-and-replace. Some participants, particularly market makers who update quotes frequently, do not amend orders but instead cancel the old order and submit a new one. The new order always gets a fresh timestamp at the back of the queue, there is no way to retain priority through a cancel-replace. Some systems provide an **order modify** message specifically to allow quantity reductions without losing priority; using a cancel-replace when you only want to decrease size is a mistake that costs queue position unnecessarily.

Implications for software: the exchange engine must apply the correct priority rule based on the type of amendment. A modify message that contains only a quantity decrease should leave the order's timestamp unchanged. Any other modification, price change, quantity increase, TIF change, or unknown field change, should assign a new timestamp. The simplest safe implementation: treat all amendments other than quantity decrease as cancel-plus-new-order.

5.2 Ticks: The Minimum Price Increment

Prices in financial markets do not move continuously. They move in discrete steps called **ticks**. The minimum price movement is the **tick size**. Tick size is the reason two prices can be safely compared for priority at all, without it, "better price" would be a question with infinitely many possible answers between any two quotes.

For most US equities, the tick size is $0.01 (one cent); for US equity futures on CME it varies by product; EUR/USD in the interbank market moves in **pips** (0.0001 of the exchange rate). This is only the surface of the topic, tick size interacts with price representation, regulation, and market structure in ways that cause real numerical bugs when misunderstood, so it gets a full chapter of its own immediately following this one: *Tick Sizes and Fractional Ticks*.

CHAPTER 6

Tick Sizes and Fractional Ticks

The previous section introduced the tick as the minimum price increment and showed why prices are stored as integer tick counts rather than floating-point decimals. That was the minimum you needed to read the rest of Part II. This section goes deeper, because tick size is one of the most common sources of subtle numerical bugs in exchange software, and because it has a genuinely interesting history that explains why the rules look the way they do today.

The Odd-Eighths Scandal: The Study That Moved a Market (1994–97)

In 1994, finance professors William Christie and Paul Schultz published a deceptively simple observation: for a large set of heavily traded NASDAQ stocks, market makers almost never quoted prices in odd eighths ($, , ,). With the even eighths alone in use, the effective minimum spread was a quarter dollar, not the eighth the tick regime allowed, consistent with tacit coordination among dealers to keep spreads wide. The market's reaction was itself a natural experiment: on 27 May 1994, the day after the findings were reported in the press, quoted spreads on several of the named stocks collapsed by roughly half, overnight, with no rule change of any kind, documented in a companion paper by Christie, Harris, and Schultz in the same issue of the *Journal of Finance*. The Department of Justice and the SEC investigated; the SEC's Section 21(a) report (August 1996) found that NASDAQ dealers had followed a "quoting convention" and that the NASD had failed to police its own market; the follow-on civil litigation settled in 1997 for approximately $1.03 billion, then the largest antitrust settlement on record. The regulatory aftermath, the SEC's Order Handling Rules (1997), Regulation ATS (1998), and momentum toward decimalization (2001), reshaped US equity market structure. It remains the cleanest documented case of an academic paper, standing on public quote data, exposing and ending a market-wide pricing convention.

References: William G. Christie and Paul H. Schultz, "Why Do NASDAQ Market Makers Avoid Odd-Eighth Quotes?" *Journal of Finance* 49(5), 1994; William G. Christie, Jeffrey H. Harris, and Paul H. Schultz, "Why Did NASDAQ Market Makers Stop Avoiding Odd-Eighth Quotes?" *Journal of Finance* 49(5), 1994

6.1 Recap: Why Integer Tick Counts

In software, storing prices as binary floating-point numbers (like Python's `float` or C's `double`) is dangerous because most decimal fractions have no exact binary representation. The double nearest to 150.30 is actually 150.3000000000000011368683772161602973937988281250, and the classic demonstration, `0.1 + 0.2 == 0.3`, evaluates to `False` in almost every programming language (the sum is 0.30000000000000004). Two prices that "should" both be $150.30, one typed in and one computed by adding a tick to $150.29, can therefore differ in their last bit and be treated as different price levels, a subtle but serious bug.

The standard solution is to store prices as **integer tick counts**: the number of minimum increments from zero. $150.30 with a tick size of $0.01 is stored as the integer 15030. Integers are exact. 15030 always equals 15030. All arithmetic, comparison, addition, subtraction, is exact with integers. Prices are only converted back to decimal notation when displaying them to humans, at the output boundary.

What happens when a non-aligned price is submitted? If a participant submits a limit buy at $150.305

when the tick size is $0.01, the price is not a valid tick multiple. The pre-trade validation layer rejects the order before it reaches the matching engine and returns an error message identifying the nearest valid prices on either side ($150.30 and $150.31) so the participant can resubmit correctly. The order receives REJECTED status and never enters the book. This becomes especially important at larger tick sizes, a futures product with a $0.25 tick makes it easy to accidentally submit a price like $150.10 that falls between valid multiples.

6.2 Tick Size Is Not One Number Per Market

A common assumption among developers new to this domain is that a market has *a* tick size, one cent for US stocks, done. In practice tick size is reference data attached to each instrument (see the *Reference Data* section of Part IV), and it varies along several axes:

By price band. Several major venues use a **banded** or **variable tick regime**, where the tick size itself changes as the price of the instrument changes. The European tick size regime, harmonised across EU venues under MiFID II (and retained, with local adjustments, in the UK), is the clearest example: in the same liquidity band, a share priced at €0.80 might trade in €0.0005 increments, while a share priced at €80 trades in €0.05 increments. The logic is that a fixed absolute tick would be economically meaningless across a 100x price range, one cent is a huge relative increment on a 50-cent stock (2% of the price) and a trivial one on a $500 stock (0.002% of the price). The EU's tick size table under MiFID II defines around twenty price ranges crossed with six liquidity bands (based on how many trades a share sees per day), and each cell of that table has its own tick.

By product, in derivatives markets. CME Group sets tick size per contract based on the economics of the underlying: the E-mini S&P 500 futures contract moves in increments of 0.25 index points (worth $12.50 per contract given its $50 multiplier), while a US Treasury futures contract ticks in fractions of 1/32nd of a point, a direct holdover from how Treasury bonds themselves have quoted in fractions for generations (more on this below).

By listing venue for the same underlying stock. Because reference data is defined per venue, the same company's shares can carry a different tick size on different exchanges, this is one of several reasons a smart order router (see Part IV) must load tick size per venue per symbol rather than assuming a single global value.

A software system that hardcodes "$0.01" as the tick size, rather than reading it from reference data per instrument and per price band, will work correctly for the common case and then fail unpredictably the first time it encounters a low-priced stock, a bond, or a European listing.

6.3 Sub-Penny Quoting and Rule 612

In the US, **Rule 612 of Regulation NMS** (adopted 2005) prohibits any exchange, ECN, or market maker from displaying, ranking, or accepting a quotation in an NMS stock priced at $1.00 or more in an increment finer than $0.01. This is why a US equity above a dollar cannot be quoted at $150.305, it isn't merely that the reference exchange in this book chooses a penny tick, it is a regulatory floor that applies market-wide. Stocks priced below $1.00 are permitted sub-penny increments (as fine as $0.0001), reflecting the same price-band logic described above: a penny is a huge relative increment on a $0.50 stock. The rule exists because, before it, some venues and market makers competed for order flow by quoting fractions of a cent ahead of a competitor's whole-cent quote, a practice regulators judged to add negligible price improvement while degrading the reliability and comparability of quotes across venues, exactly the "fairness and transparency" promise described in Part I. (The penny floor is not frozen, either: in September 2024 the SEC adopted amendments introducing a half-penny ($0.005) tick for heavily traded stocks whose spreads are routinely pinned at one cent, roughly 70% of listed securities by one estimate. The rule survived a court challenge in October 2025, but its implementation timetable has since been under review. A tick size, in other words, is a configurable parameter even at the level of national regulation.)

6.4 Fractions, Not Decimals: A Short History

For almost the entire history of the NYSE, from 1792 until 2001, US stocks did not trade in cents at all. They traded in **fractions of a dollar**, initially eighths ($0.125), inherited from the Spanish colonial "piece of eight"

that circulated as US currency in the exchange's early decades and was itself divisible into eight reales, hence "pieces of eight" and the fractional quoting convention that outlived the coin by two centuries. A stock might be quoted "50 and 1/4" (meaning $50.25) or move by "an eighth" ($0.125). By the late twentieth century, the minimum tick on NYSE had been narrowed once, to sixteenths of a dollar ($0.0625, informally a "teenie"), but the fractional system itself persisted.

This mattered enormously for spreads and trading costs. A market maker quoting a one-eighth-wide spread ($0.125) on an actively traded, high-volume stock was capturing a spread that, in relative terms, was often far wider than the genuine cost of providing liquidity justified, precisely because $0.125 was the smallest increment available; there was no way to compete down to a narrower spread even if a market maker wanted to. Academic microstructure research through the 1990s (see Harris [1] and Hasbrouck [2] in the references), not least the odd-eighths study in the note above, built a substantial body of evidence that the fractional system was inflating trading costs for ordinary investors.

Decimalization, the switch from fractions to decimal cents, was mandated by the SEC and phased in across US markets between August 2000 and April 2001, with NYSE and NASDAQ fully converted by April 2001. The effect was immediate: quoted spreads, especially on actively traded stocks, fell sharply, because market makers could now compete for order flow in one-cent increments instead of being locked into sixteenths [Bessembinder, "Trade Execution Costs and Market Quality after Decimalization", *Journal of Financial and Quantitative Analysis* 38(4), 2003]. It remains one of the most-studied natural experiments in market microstructure, precisely because it changed exactly one variable, the tick size, while leaving the rest of market structure intact, and the resulting narrower spreads are the baseline every example of tick size and spread in this book assumes. Decimalization is also the direct reason **Rule 612** exists: having gone through the disruption of moving to a one-cent tick, regulators wanted to prevent competition from immediately re-fragmenting into sub-penny increments, undoing the benefit just achieved.

6.5 The Tick Size Pilot Program (2016–2018)

Narrower ticks are not unambiguously good for every stock. A one-cent tick works well for a heavily traded, liquid large-cap stock, but critics argued that for small, thinly traded companies, a one-cent spread gives market makers too little compensation to justify committing capital and quoting size, potentially *reducing* liquidity for exactly the companies that need it most.

To test this empirically, the SEC approved the **Tick Size Pilot Program**, which ran from October 2016 to October 2018. Eligible stocks were small companies (market capitalisation of $3 billion or less, average daily volume of 1 million shares or less, price of at least $2). From them, 1,200 were randomly assigned to three test groups of 400, quoting (and, in two of the groups, also trading) in five-cent increments under varying rules; the remaining eligible stocks formed a control group that kept the one-cent tick. The results were genuinely mixed. Quoted spreads widened (in the strictest group by about 24%), and so did trading costs for small orders. Displayed size at the best prices, on the other hand, rose dramatically, by several hundred percent, and market makers' profits per share rose too. What did *not* materialise was the hoped-for benefit that justified the experiment: better overall liquidity and more capital formation for small companies [SEC DERA, *Assessment of the Plan to Implement a Tick Size Pilot Program*, 2018]. The SEC let the pilot expire without adopting a wider tick for small caps. The pilot is a useful case study for exchange developers precisely because it treats tick size as an empirical, testable market-structure parameter rather than a fixed constant, and because it shows how one change can improve one metric (displayed depth) while worsening another (spread).

6.6 A Worked Example: Rounding a Midpoint to a Valid Tick

The *Order Types* section described **midpoint peg orders**, whose fill price is computed dynamically as (best_bid + best_ask) / 2, and flagged that this calculation "requires care with rounding to the nearest valid tick." Here is exactly why, worked through in ticks rather than dollars so the arithmetic is exact.

Suppose the tick size is $0.01, so prices are stored as integer cent-counts. The best bid is $150.30 (tick count 15030) and the best ask is $150.33 (tick count 15033):

```
sum        = 15030 + 15033 = 30063
midpoint   = 30063 / 2      = 15031.5   ← not an integer!
```

The true mathematical midpoint, \$150.315, is **not a whole number of ticks**: it falls exactly halfway between \$150.31 (tick 15031) and \$150.32 (tick 15032). What happens next depends on the market. US equity rules are, perhaps surprisingly, relaxed about this particular case: Rule 612 restricts sub-penny *quotes and orders*, not executions, so midpoint trades at half-penny prices such as \$150.315 are permitted and common, and an engine that stores prices as integer cents must therefore represent them in a finer unit (for example half-cents or hundredths of a cent). In many other markets, including futures, every trade must print on a valid tick, and the matching engine must round 15031.5 to one of its two neighbours. *Which* neighbour it chooses is a real economic decision, not a cosmetic one: rounding up gives the extra half-cent to the seller, rounding down gives it to the buyer.

This is exactly the kind of detail that must be a single, explicit, documented rule rather than "whatever the rounding function in the standard library happens to do." A system with two components that both round tick-and-a-half prices, say, the matching engine and a downstream P&L reconciliation service, must use the *identical* rounding rule, or the two will silently disagree by one cent on what price a given midpoint fill occurred at. In isolation a one-cent discrepancy looks trivial; multiplied across millions of midpoint fills a day, it becomes a real, hard-to-trace reconciliation break between systems that are supposed to be describing the same trade. Common conventions include round-half-up, round-half-to-even ("banker's rounding," which avoids systematically favouring either side over many trades), or a rule that explicitly favours the resting order as compensation for the queue-priority risk it took by waiting. Whichever rule is chosen, the requirement is that it lives in exactly one place and every component that touches a midpoint price calls the same function, the same principle of a single authoritative source that governs reference data generally (see Part IV).

Note that this specific problem, an odd sum of tick counts, arises exactly when the bid and ask are an odd number of ticks apart (here, 3 ticks: $15033 - 15030 = 3$), because bid + ask $= 2 \times$ bid + spread, which is odd precisely when the spread is odd. When the spread is an even number of ticks (for example, bid 15030 and ask 15032, a two-tick spread), the midpoint lands exactly on a valid tick with no rounding decision required. And a one-tick spread, the tightest a market can be and the normal state of a liquid stock, *always* has the problem: (15030 + 15031) / 2 = 15030.5. In other words, the rounding rule is not an edge case; in liquid markets it is exercised on nearly every midpoint trade.

> **Key idea:** Tick size is reference data, not a universal constant, it varies by price band, by product, and by venue, and must be read from configuration rather than hardcoded. Historically, US markets operated in fractions for two centuries before decimalization in 2001 proved that narrower ticks meaningfully reduce trading costs, which is precisely why Rule 612 now puts a regulatory floor under how fine a tick can go. And because a midpoint (or any other computed, non-stored price) does not always land on a valid tick, every system component that rounds one must use the same rounding rule, or fills that are supposed to describe the same trade will silently disagree.

CHAPTER 7

The Matching Engine, The Heart of the Exchange

The **matching engine** is the software that receives orders, manages the order book, and executes trades when buy and sell orders are compatible. It is the exchange's most critical, performance-sensitive component.

Figure 1: *The matching engine receives orders from participants, manages the order book, and executes trades when compatible orders are found. It publishes trade and order status events to participants and subscribers.*

7.1 The Core Loop

At its simplest, the matching engine runs an endless loop:

1. Receive an incoming order.
2. Check if it can immediately match against resting orders on the opposite side.
3. If yes, execute the match (create a trade, update quantities, notify participants).
4. If there is remaining unfilled quantity, decide what to do with it (rest it in the book, or cancel it, depending on order type and TIF).
5. Check if any dormant stop orders have now been triggered by the new trade price.
6. Publish the results (trades, order status changes) to participants and subscribers.

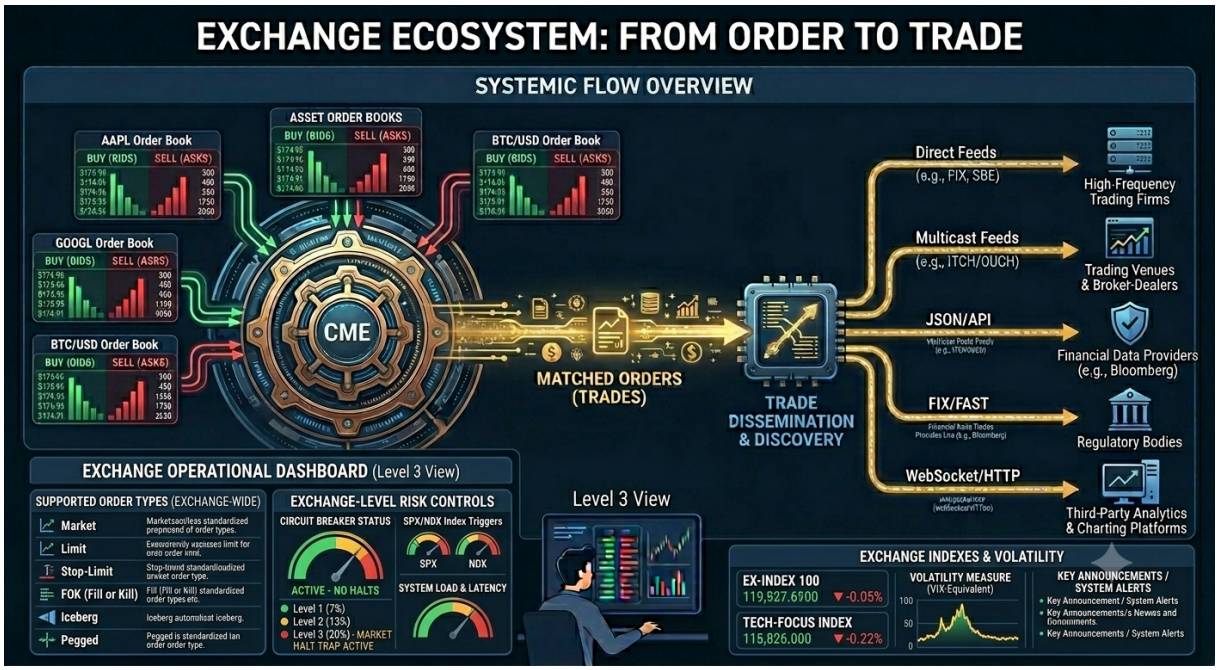
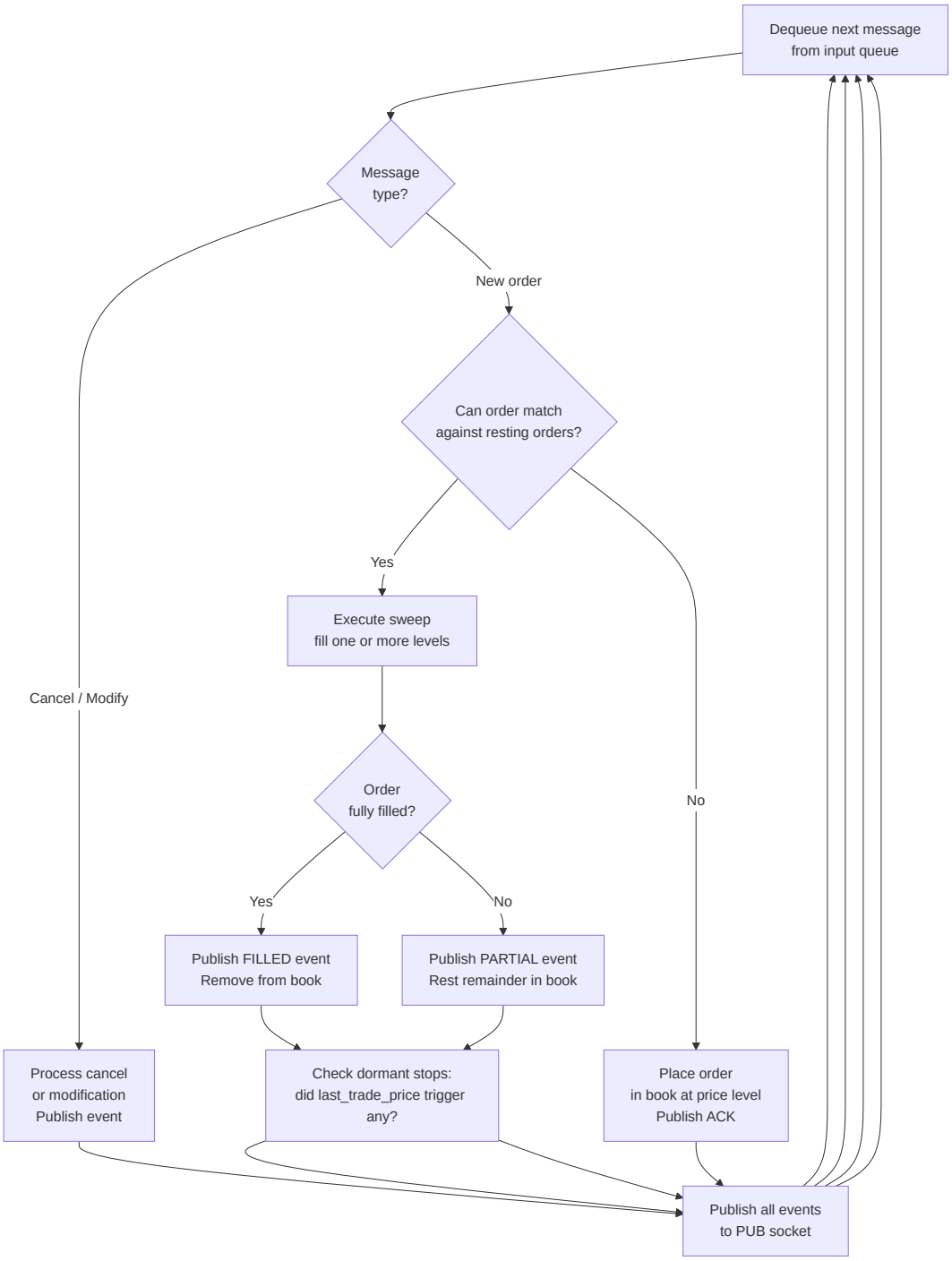


Figure 7.1: Matching Engine Diagram



7.2 The Sweep

When an aggressive order arrives and begins filling against the resting book, this process is called **sweeping** the book. The aggressive order works through price levels one by one, from best to worst, until either it is fully filled or it reaches its limit price (or the book runs out of orders).

For example, a market buy order for 5,000 shares arrives: - Takes 2,000 shares at the best ask of $150.35. - Takes 1,500 shares at the next level, $150.40. - Takes 1,200 shares at $150.45. - Takes 300 shares at $150.50. - Fully filled. Total cost: a weighted average of these prices.

The price impact of sweeping through multiple levels is called **slippage** or **market impact**, the large order moved the effective price away from the initial best ask.

7.3 Single-Threaded by Design

Perhaps counterintuitively, most matching engines are **single-threaded**, they process orders one at a time, in strict sequence, with no parallel processing of orders. This is not a performance limitation; it is a deliberate design choice.

The order book is a shared, stateful data structure. If two threads tried to modify it simultaneously, you would need complex locking mechanisms to prevent corruption, and you might still end up with non-deterministic outcomes. In a system where fairness and determinism are legally required, this is unacceptable.

By processing orders in a single thread, the engine guarantees that the outcome is perfectly deterministic: given the same sequence of orders, the same sequence of trades will always result.

> **Key idea:** Single-threaded design is a feature, not a limitation. It makes the matching engine auditable, replayable, and legally defensible. Performance comes from algorithmic efficiency, not parallelism.

The performance requirement is achieved through algorithmic efficiency (correct data structures, avoiding unnecessary computation) rather than parallelism. The world's fastest matching engines process an order in a few microseconds or less. For scale: in 2025 CME Group's venues traded a record average of 28.1 million *contracts* per day [CME Group press release, 5 January 2026], and the number of *messages* (new orders, modifications, cancellations) behind those trades is far larger still, because most orders are cancelled or modified without ever trading. Major equity exchanges typically acknowledge an order within tens of microseconds, and the fastest co-located trading firms, often using FPGA hardware, can react to a market data update with an order in well under a microsecond. These numbers drive the entire hardware and architecture design: no garbage-collected language, no dynamic memory allocation in the critical path, no operating system calls that can introduce variable latency.

7.4 One Book Per Symbol

The matching engine maintains one order book per tradeable symbol. AAPL trades in one book, MSFT in another. These books are entirely independent, an order in the AAPL book cannot interact directly with an order in the MSFT book (combo orders handle cross-symbol interaction at a higher level).

Conceptually, one logical order book per symbol is the correct mental model. In production systems, the implementation may be distributed differently, books may be sharded across cores, replicated for high availability, or partitioned by instrument group across multiple machines, but the logical behaviour is identical: each symbol has its own independent price-time priority queue.

Where symbol independence breaks down: Several order types and matching mechanisms require coordination *across* symbols, and developers should not over-generalise the "symbols are independent" rule:

- **Spread orders and calendar spreads** (CME): a single order to buy a March futures contract and sell a June contract simultaneously. The exchange must evaluate both legs together, filling only one is leg risk.
- **Implied matching** (derivatives markets): if there is a resting spread order to buy March and sell June, and a resting outright order to sell March, the two together are a synthetic offer to sell June, and the exchange can publish it as an *implied* order in the June book and fill an incoming June buyer against both of them at once. CME Globex implements implied matching across many contract months. This topic gets two full chapters at the end of this Part.
- **Multi-leg options strategies** (Eurex, CBOE): a straddle (buy call + buy put at the same strike) or a strangle requires co-ordination between two different option series, each with its own symbol.
- **Basket orders and index rebalancing**: trading a portfolio of dozens of stocks as a single instruction requires cross-symbol scheduling and coordination.

For exchange system developers, this means the "separate process per symbol" architecture that works cleanly for single-instrument matching must be extended, or wrapped with a higher-level combinator, to handle any multi-leg or implied matching requirements.

7.5 The Role of Data Structures

The order book needs to answer one question extremely fast: "What is the best available price right now?" Priority queues and heap-like structures are conceptually useful for understanding this, a heap gives O(1) access to the minimum or maximum element, and O(log n) insertion and deletion, which suits the matching engine's primary operations well.

Production exchange engines, however, often use more specialised data structures optimised for FIFO ordering within price levels, cache locality, and deterministic latency: balanced trees, skip lists, or intrusive linked lists indexed by price level are common. The conceptual model of "best price always accessible in O(1)" is correct regardless of the underlying structure; the heap is a pedagogically useful approximation of what these structures achieve.

CHAPTER 8

The Life of a Trade

Let us trace a single trade from start to finish.

8.1 Order Submission

Participant A, connected through Gateway GW01, submits a limit buy order: "Buy 500 shares of AAPL at $150.30, DAY." The order arrives as a **FIX message** (Financial Information eXchange, the standard protocol for order submission in financial markets, covered in full in the Technology Architecture section). The gateway validates the basic format and forwards it to the matching engine.

The engine receives the order and assigns it: - A **unique order ID** (a system-wide identifier, discussed in detail in the architecture section) - A **timestamp** (nanosecond precision) - An initial status of **NEW** (the order has been accepted and registered)

The engine publishes an **acknowledgment (ACK)** back to GW01 confirming the order is live. The order now rests in the book.

8.2 The Match

Moments later, Participant B, connected through Gateway GW02, submits a sell limit order: "Sell 500 shares of AAPL at $150.30, DAY." Or perhaps a market sell order. Either way, the engine determines that this sell order can match the resting buy order.

The match happens: 500 shares trade at $150.30. Two events occur simultaneously: - Order A transitions from status NEW to **FILLED**. - Order B transitions from status NEW to FILLED.

Both GW01 and GW02 receive **fill notifications** (also called **execution reports**) detailing the trade: quantity, price, and any remaining unfilled quantity.

A **trade record** is created, capturing: - The trade ID - The symbol - The price - The quantity - The IDs of both orders involved - The IDs of both gateways (participants) - The timestamp - Which side was the **aggressor** (which order arrived and "took" the fill)

The aggressor field is more than just a label. It matters for several reasons. First, for **fee calculation**: most exchanges charge takers (aggressors) a fee and pay makers (resting orders) a rebate, so correctly identifying which side was aggressive determines the billing for each trade. Second, for **regulatory reporting**: in some jurisdictions, trades must be classified as "buy-initiated" or "sell-initiated" based on which side was the aggressor. Third, for **market analysis**: the sequence of buyer-initiated and seller-initiated trades is a standard input to microstructure models that infer order flow direction and predict short-term price movement. Fourth, for **clearing**: in some clearing architectures, the aggressor side may have different margin or settlement obligations.

8.3 Partial Fills

If Order A was for 500 shares but only 300 shares were available at $150.30, a **partial fill** occurs: 300 shares trade, Order A's status becomes **PARTIAL** with 200 shares remaining, and it continues resting at $150.30 waiting for more sellers.

8.4 Publication

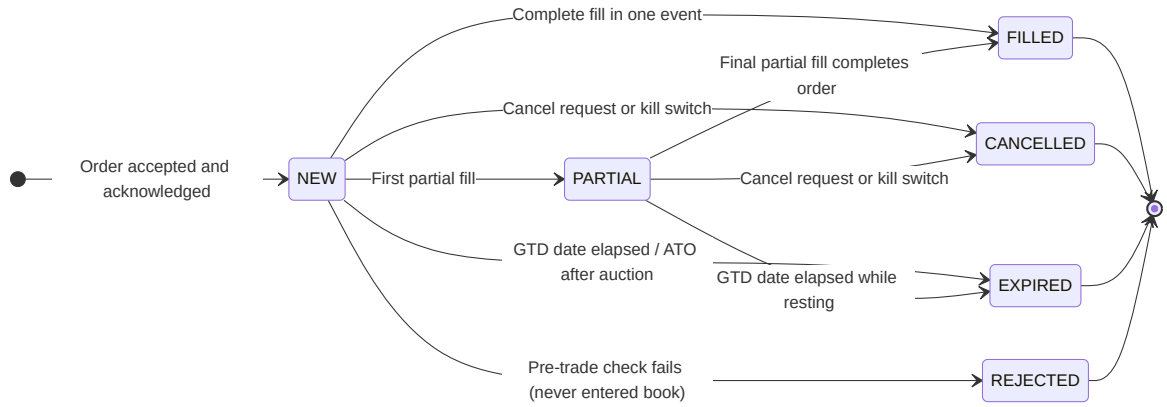
Every trade is immediately published over the market data feed to all subscribers. The viewer, board, stats database, clearing system, and any other subscriber all receive the trade notification within microseconds of it occurring.

8.5 Order Status Lifecycle

An order passes through a defined set of statuses during its lifetime. Understanding these is essential for anyone building order management, reporting, or audit systems.

Status	Meaning
NEW	The order has been accepted and acknowledged by the exchange. It is live, either resting in the book (if a limit order) or being matched (if aggressive).
PARTIAL	At least one fill has occurred but the order quantity is not yet fully satisfied. The remaining quantity continues to rest or be eligible for matching.
FILLED	The entire order quantity has been executed. The order is complete and leaves the book.
CANCELLED	The order was withdrawn before it was fully filled, either by the participant, by the exchange (e.g., kill switch), or by a system rule (e.g., end-of-day cancellation of DAY orders).
REJECTED	The order was refused by the gateway or the engine before entering the book, for example, failing a pre-trade risk check, containing an invalid price, or submitted during a halt. A rejected order never entered the book.
EXPIRED	The order's time-in-force condition was not met. GTD orders that reach their expiry date without filling receive this status. ATO orders unfilled after the opening auction expire automatically.

The key transitions: an order starts as NEW. A partial fill moves it to PARTIAL. A final fill completes the transition to FILLED. A cancel message (from the participant) or a kill switch (from the exchange) moves a NEW or PARTIAL order to CANCELLED. A pre-trade rejection produces REJECTED without the order ever becoming NEW. An elapsed GTD date produces EXPIRED.



Key idea for developers: REJECTED is fundamentally different from CANCELLED. A rejected order never existed in the book; a cancelled order did. Audit trails must record both, but they have different implications for position tracking (a rejected order has no position impact) and for regulatory reporting.

CHAPTER 9

Market Makers, The Providers of Liquidity

Earlier we introduced market makers conceptually as liquidity providers. In this section we examine their operational interaction with the exchange in much greater detail: quote lifecycles, formal obligations, protection mechanisms, inventory risk, and the software implications of supporting them.

> **Key idea:** Market making is not a passive activity. The market maker's position, risk, and obligations change with every fill, every second of elapsed quoting time, and every price movement in the market. The exchange system must enforce all of this automatically, in real time, without manual intervention.

9.1 Market Maker Obligations

Being a designated market maker is not a free lunch. The exchange grants privileges, reduced fees, faster data access, and in some markets priority access to certain order flow, but in return imposes binding contractual obligations. These are monitored continuously in real time. Failing to meet them results in financial penalties or loss of market maker status.

The typical obligations, in descending order of fundamentality, are:

1. Two-sided quoting. The market maker must always have both a live bid and a live ask resting in the book at the same time. Quoting only one side defeats the purpose of providing liquidity and is a contractual breach.

2. Maximum spread. The ask price must not be more than a specified number of ticks above the bid. If the market maker agreement specifies a maximum spread of 5 ticks on AAPL (tick size \$0.01), the market maker can never quote a bid at \$150.30 and an ask at \$150.36 simultaneously, the gap of 6 ticks is too wide.

3. Minimum size. Both the bid and ask must offer at least a specified minimum quantity. Quoting 50 shares when the minimum is 200 is insufficient; the liquidity must be meaningful.

4. Maximum distance from mid. Quotes must remain within a specified distance of the current mid price. Quoting a bid at \$100 and an ask at \$200 on a \$150 stock satisfies the letter of two-sided quoting but not its spirit.

5. Presence obligation. The market maker must be actively quoting for at least a specified percentage of the trading session, commonly 85% or more. This prevents market makers from only quoting on easy days and disappearing on volatile ones when liquidity is most needed.

6. Re-quoting obligation. After either side of a quote is filled, the market maker must post a fresh replacement quote within a specified maximum delay, typically a few hundred milliseconds to a few seconds depending on the product and exchange. The exchange system must track fill events and monitor whether replacement quotes arrive in time.

9.2 The Two-Sided Dilemma: When One Side Is Taken

The obligations above sound manageable until you think carefully about what happens at the moment of a fill. This is where market making becomes genuinely complex, and where the concept of **adverse selection** becomes central.

Adverse selection occurs when the participant trading against a market maker is doing so because they have information that the market maker lacks. The informed trader knows the price is about to move; the market maker does not. The market maker ends up buying at $150.30 just before the price falls, or selling at $150.35 just before the price rises. This is not bad luck; it is the systematic consequence of always being available to trade against anyone who shows up.

The arithmetic makes the danger concrete. Suppose a market maker is quoting AAPL: - **Bid:** Buy 500 shares at $150.30 - **Ask:** Sell 500 shares at $150.35

An aggressive sell order arrives and hits the bid. The market maker has just bought 500 shares at $150.30. But their ask, "sell 500 shares at $150.35", is still live. The problem: this situation is no longer neutral. The seller was aggressive, they had a reason to sell urgently. If that reason is information (the price is about to fall), the market maker has been adversely selected.

The arithmetic: if the true price falls to $150.10 after the fill, the market maker now holds 500 shares worth $150.10 that they paid $150.30 for. Loss: $0.20 × 500 = **$100**. What did the market maker expect to earn? Buying at the bid, $0.025 below the mid price of $150.325, earns the *half-spread*: $0.025 × 500 = **$12.50** per fill. The full spread, $0.05 × 500 = **$25**, is earned only on a complete round trip, when a matching sale at the ask follows the purchase at the bid. A single adverse selection event of $100 therefore wipes out the revenue of four complete round trips, or eight ordinary fills. If the informed seller keeps hitting the bid repeatedly before the market maker can react, the losses compound rapidly.

The exchange system must have a defined policy about what to do with the surviving side, because these events occur in microseconds with no time for manual decisions.

9.3 Quote Refresh Policies

The **quote refresh policy** is the rule governing what happens to the surviving leg of a two-sided quote when the other leg fills.

Policy 1, Cancel both sides immediately (most conservative). The moment either the bid or ask is filled, the entire quote is automatically cancelled. The market maker starts fresh and must submit a new two-sided quote within their re-quoting window. This is the safest policy: no stale inventory, no unintended exposure.

Policy 2, Cancel only the filled side, leave the other active. The surviving side remains live. Faster for market liquidity but riskier for the market maker, if prices are moving, the surviving quote may already be stale.

Policy 3, Leave both sides active. No automatic action. Riskiest; used only in tightly controlled situations.

The exchange system must implement all variants, configurable per market maker per product, typically via a quote refresh policy setting checked each time a quote leg fills.

9.4 Market Maker Protection (MMP)

Even with careful refresh policies, market makers face a specific danger: **adverse selection at high speed**.

When an algorithm with access to breaking news starts aggressively selling, it does not submit one large order, it sends many small orders in rapid succession, each hitting the market maker's bid before they can react. In milliseconds, the market maker has bought far more inventory than intended at prices that are about to move against them.

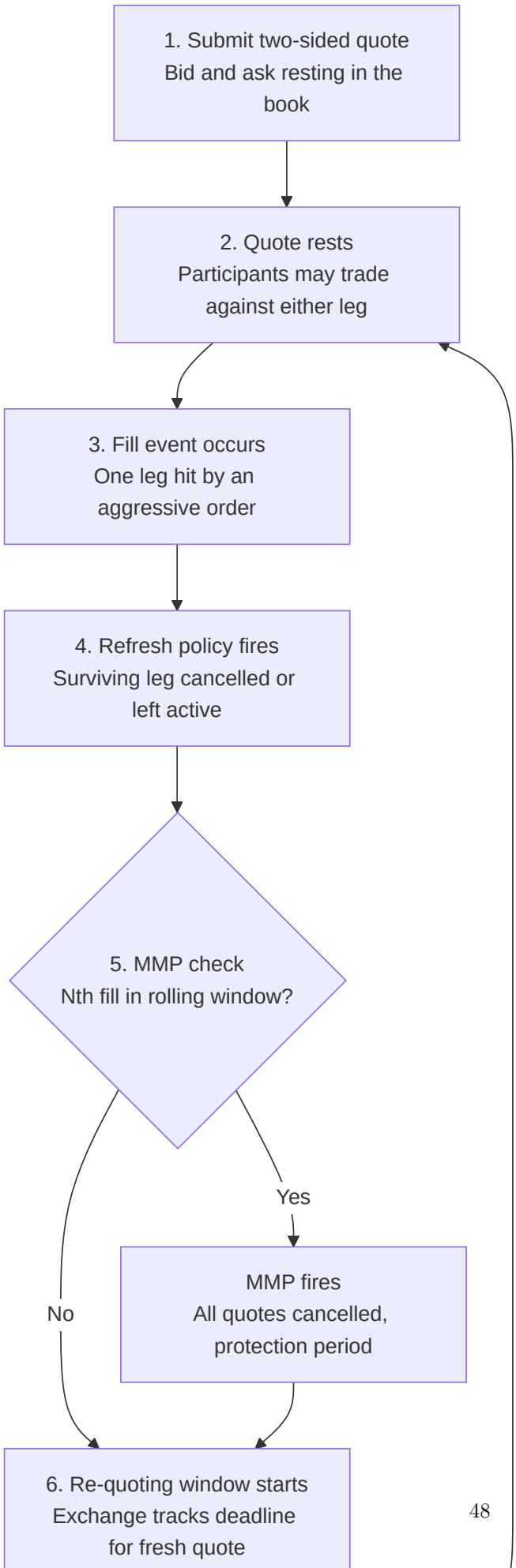
Market Maker Protection (MMP) is the automated countermeasure. The exchange maintains a rolling time window per market maker and counts fills within it. If fills arrive faster than a configured threshold, MMP fires automatically, cancelling all of that market maker's quotes without waiting for human intervention.

After MMP fires, the market maker enters a **protection period** to assess and decide before re-quoting. MMP parameters (fill count, time window) are configured per market maker per symbol and are formally specified in market maker agreements. An illustrative configuration might be "5 fills within 500 milliseconds" for a liquid equity option: if five fills occur faster than that, the system assumes the market maker is being picked off by an informed algorithm and activates protection automatically. Real implementations usually measure more than a simple fill count. Eurex's T7, for example, lets market makers set "percentage, volume, delta and vega limits", so protection can trigger on how much risk has been accumulated rather than on how many trades produced it, and it *inactivates* the quotes (which the market maker can then reactivate) rather than deleting them [Eurex, *T7 Functional and Interface Overview*]. The parameters are different for each product: a highly liquid product like a major equity index option might allow more activity in a given window than an illiquid single-stock option.

Why exchanges provide MMP: Without it, market makers would either quote much wider spreads (to compensate for adverse selection risk) or withdraw entirely. MMP is not a favour to market makers, it is infrastructure that makes tight spreads and deep liquidity sustainable for the whole market.

In a typical implementation a fill-counter function tracks fills against the rolling window, and an MMP activation function cancels all resting quotes for that market maker when the threshold is exceeded.

9.5 The Full Lifecycle of a Market Maker Quote



This cycle, repeating dozens of times per minute per symbol, is what "providing liquidity" means operationally. A professional market maker runs sophisticated algorithms that evaluate every fill, update pricing models, and decide within microseconds whether and at what prices to re-quote.

CHAPTER 10

The Opening and Closing Auction

Trading does not simply start at 9:30am and stop at 4:00pm (for US equities). The transition between closed and open trading is managed through a **call auction**, also called a **fixing** or **uncross**.

10.1 The Problem an Auction Solves

Imagine the exchange has been closed overnight. Many participants have new orders to submit. If the exchange simply opened and started matching immediately, the first few orders would define the opening price, which would be arbitrary and potentially far from "fair value" based on overnight news.

Instead, exchanges run an **opening auction**:

1. During a pre-open period (on NYSE, from early morning until the 9:30am open), participants submit orders. These orders are accepted but not matched. The book quietly accumulates interest on both sides.
2. At the close of the auction period, the exchange's **equilibrium algorithm** finds the single price at which the maximum quantity can trade. This is the **equilibrium price** (also called the **clearing price** or **auction price**).
3. All orders that can trade at the equilibrium price are **uncrossed** (matched) simultaneously, all at that one price. There is no sweeping through levels, everyone trades at the same price.
4. Any remaining unfilled orders transition to the continuous book.

The result is a fair opening price that reflects the overnight information available to all participants simultaneously, rather than favouring whoever happened to submit their order a few milliseconds earlier.

10.2 Finding the Equilibrium Price

The equilibrium price is the price that maximises total traded volume, the **maximum executable volume rule**. For each candidate price, the algorithm calculates: - How many buy orders would trade at that price or better (buyers willing to pay at least that much) - How many sell orders would trade at that price or better (sellers willing to accept at most that much)

The executable volume at each candidate price is `min(cumulative_bids, cumulative_asks)`. The price that maximises this is the equilibrium.

A worked example. During the pre-open period, the following orders have accumulated:

Buy orders: 500 shares at \$152, 1,000 shares at \$151, 800 shares at \$150

Sell orders: 600 shares at \$150, 900 shares at \$151, 800 shares at \$152

The equilibrium algorithm evaluates each candidate price:

Candidate Price	Buyers willing to pay price	Sellers willing to accept price	Executable (min)
$150	500+1,000+800 = 2,300	600	**600**
$151	500+1,000 = 1,500	600+900 = 1,500	**1,500** ← maximum
$152	500	600+900+800 = 2,300	**500**

The equilibrium price is **$151**, the single price at which the most volume (1,500 shares) can trade. All buyers who bid $151 or more (1,500 shares total) and all sellers who asked $151 or less (1,500 shares total) trade simultaneously at $151, regardless of their individual limit prices. The buyer who bid $152 still pays only $151, receiving $1 per share of price improvement. The seller who asked $150 still receives $151, getting $1 above their minimum. Two orders do not execute: the 800-share buy order at $150 (it will not pay $151) and the 800-share sell order at $152 (it will not accept $151). Both transition to the continuous book, which therefore opens with a best bid of $150 and a best ask of $152.

Tie-breaking when multiple prices produce the same volume: If two candidate prices both yield 500 shares executable, the algorithm must choose between them. The standard tie-breaking rules, applied in order, are:

1. **Minimise the imbalance.** The imbalance is the unfilled quantity on the heavier side after the uncross. Prefer the price that leaves the smallest imbalance. If at $151 the buy side has 500 executable but 200 additional buy orders remain, the imbalance is 200. A price with a smaller imbalance is preferred.
2. **Match market pressure.** If imbalances remain equal, prefer the price in the direction of the remaining pressure, if there is a surplus of buys, choose the higher of the tied prices; if a surplus of sells, the lower.
3. **Proximity to reference price.** If all else is still equal, choose the price closest to a reference price (typically the previous close or the last traded price from the previous session).

Indicative pricing during the accumulation period: While the auction is still accumulating orders and before uncrossing, many exchanges continuously publish an **indicative uncross price**, the price that *would* result if uncrossing happened at that moment. This lets participants adjust their orders as the indicative price evolves. The indicative price is recalculated after every order arrives.

Auction imbalance messaging: Exchanges often publish the **imbalance**, how many more shares are on the buy side than the sell side at the indicative price, to help participants decide whether to submit offsetting orders before the uncross. NYSE publishes closing auction imbalance information from 3:50pm, updated as often as every second, giving participants ten minutes to respond before the 4:00pm uncross [NYSE, *Closing Auction Fact Sheet*].

10.3 Auction-Only Order Types

The order types introduced in the *Order Types* chapter are the vocabulary of *continuous* trading. Auctions add a small, specialised vocabulary of their own, order types that exist specifically to participate in an uncross and that behave differently, or do not exist at all, during continuous matching. A developer implementing an auction must handle these as distinct types with their own eligibility rules, not as ordinary limit orders that happen to arrive during the call period.

Market-on-Open / Market-on-Close (MOO / MOC). An instruction to trade at whatever price the auction produces, with no price limit. An MOC order says, in effect, "I want to be part of the closing print, at the closing price, whatever it is." Because it names no price, it is always willing to trade and therefore always counts toward executable volume at the eventual equilibrium; its only risk is the auction price itself. MOO is the opening-auction equivalent.

Limit-on-Open / Limit-on-Close (LOO / LOC). The same idea with a price limit. An LOC to buy at $151 participates in the closing auction only if the closing price is $151 or lower. It expresses "I want the close, but not above (or below) this price."

Imbalance-only orders (IO). This is the auction-specific type most likely to surprise a developer, because it is defined *relative to the auction's own imbalance* rather than as a standalone buy or sell. An imbalance-only

order provides liquidity **only on the side that offsets a published imbalance**, and only at prices that do not disturb the auction's price discovery. NASDAQ's Imbalance-Only order is the canonical example: an IO buy order is eligible to execute only against a sell-side imbalance, and NASDAQ constrains its effective price so that it can absorb imbalance without pushing the cross price through the prevailing quote. The purpose is to invite offsetting liquidity into a lopsided auction without letting that liquidity itself become a new source of price distortion.

Closing Offset orders (CO). NYSE's Closing Offset order is a close cousin: a CO order is a limit order that "executes only against the opposite side of an imbalance; never adds to an imbalance", and is the lowest-priority interest in the auction, it executes only after other eligible closing interest, and only to the extent it reduces a published imbalance. It lets a participant say "use me only if I am genuinely needed to balance the book," which is attractive to a trader who wants closing-price execution but does not want to add to a one-sided pileup.

Why these types need dedicated handling. Each of these interacts with the equilibrium calculation differently:

- MOO/MOC are unconditionally executable and simply add to both the "willing to buy at any price" and "willing to sell at any price" tallies.
- LOO/LOC behave like ordinary limit orders *within* the auction's cumulative-volume calculation.
- IO and CO orders are **conditional on the imbalance itself**, which means they can only be evaluated *after* a provisional imbalance has been computed from the other order types, and they must not be allowed to reverse the imbalance or move the price outside a permitted band. In implementation terms, the uncross is therefore not a single pass: the engine computes the equilibrium and imbalance from price-carrying and market orders first, then admits imbalance-offsetting interest to the extent it reduces (never inverts) the imbalance, then re-derives the final print.

A note on cut-off times. Auction order types have entry and cancellation deadlines that differ from continuous-session rules, for example, on NYSE, MOC and LOC orders must be entered by 3:50pm Eastern, after which they may be entered or modified only to offset a published imbalance, and not freely cancelled. These exact times are set by exchange rule, are periodically revised, and differ between venues; treat any specific time in this book as illustrative and verify against the current rulebook of the venue you are building for.

10.4 The Named Crosses: Opening Cross and Closing Cross

Real venues brand and specify their auctions as named mechanisms, and an engineer reading exchange documentation will meet the names rather than the generic term "uncross." The mechanics are the equilibrium algorithm described above; the names denote the specific rule set, order types, and imbalance dissemination each venue attaches to it.

The NASDAQ Opening Cross and Closing Cross. NASDAQ runs its opening and closing auctions as the **Opening Cross** and **Closing Cross**. In the run-up to the cross, NASDAQ disseminates order imbalance information, the indicative clearing price, the size of any imbalance, the paired-off quantity, at defined intervals so participants can respond with offsetting or imbalance-only interest. The Closing Cross establishes the **NASDAQ Official Closing Price (NOCP)**, the number used to value NASDAQ-listed securities at end of day. NASDAQ's cross is a fully electronic, rules-based uncross: the equilibrium price is chosen to maximise executable volume, with documented tie-breaks analogous to those above.

The NYSE close and the Designated Market Maker. NYSE's closing auction reaches the same kind of single-price print but retains a human-supervised element that reflects NYSE's floor heritage: the **Designated Market Maker (DMM)** for each security is responsible for facilitating the close, and may, within strict rules and electronic constraints, help set the closing auction price when conditions are unusual. In ordinary conditions the process is electronic and formulaic; the DMM's role is a controlled backstop for the difficult cases (large imbalances, news, illiquidity), not a discretionary override of price discovery. NYSE publishes closing imbalance information in the final minutes (as noted above, from 3:50pm) so that offsetting interest can be attracted before the print.

Why the closing print is worth this much machinery. The reason both venues invest so heavily in a robust, well-policed close is the same reason the next section on manipulation exists: an enormous quantity

of economic activity references the closing price specifically. Index funds must trade at the close to track their benchmark; derivatives settle against it; fund NAVs are struck at it; performance is measured against it. A closing price is not merely the last number of the day, it is a contractually and financially load-bearing benchmark, which is exactly what makes it a target.

10.5 The Closing Auction

The closing auction works identically but at the end of the day, establishing the official **closing price**. This is one of the most important prices in the market, it is used to benchmark fund performance, price derivatives, and compute official valuations of positions. The closing auctions are among the most liquidity-rich events in the US equity market: on a typical day in 2024, US exchanges matched about $50 billion in their closing auctions, roughly 9% of the whole day's trading volume, in a single instant. On index rebalancing days the proportion is far higher, because index funds must trade at exactly the closing price to match their benchmarks: on the Russell reconstitution day of 28 June 2024, over 34% of the day's notional volume executed in the close [BMLL, *Into the Close*, 2024].

10.6 Manipulating the Close, and Why the Auction Is Policed

Because so much value references the closing price, the close is a standing target for manipulation. The generic abuse is called **marking the close** (or, for a benchmark fixing more generally, **banging the fix**): entering orders near the end of the auction period not to trade on their merits but to push the official price in a direction that benefits a position held elsewhere, a large derivatives position expiring against the close, an index-tracking obligation, a fund's month-end valuation, or simply a book that is marked to the closing price.

The mechanics of the defence are already present in the design above. Imbalance dissemination lets honest offsetting interest arrive to counter an artificial push. The imbalance-only and closing-offset order types exist precisely to attract price-stabilising liquidity. The single-price uncross means a manipulator cannot cherry-pick; they must move the whole equilibrium, which requires committing real, exposed size. And every order into the auction is captured in the audit trail, timestamped and attributed, which is what makes after-the-fact enforcement possible.

> **Historic Note: Marking the Close by Algorithm, Athena Capital, 2014**
>
> In October 2014 the U.S. Securities and Exchange Commission settled charges against **Athena Capital Research**, a high-frequency trading firm, for manipulating the closing prices of thousands of NASDAQ-listed stocks over a six-month period in 2009, the first SEC enforcement action for manipulation brought against a high-frequency trading firm. Athena's algorithm, known internally as **"Gravy,"** concentrated a large volume of aggressive orders in the final seconds before the close, in the securities where Athena held imbalance positions, in order to push the NASDAQ Closing Cross price in its favour. Internal communications, quoted in the SEC's order, described the intent in plain terms and even referred to the tactic in ways that made the manipulative purpose explicit, evidence drawn directly from the kind of audit trail this Part describes. Athena, without admitting or denying the findings, paid a $1 million penalty. The case is instructive for exchange developers for three reasons: it demonstrates that the closing auction's economic importance makes it a manipulation target; that the very imbalance information published to *improve* the auction can be gamed by a participant willing to trade against it artificially; and that the defence is ultimately the completeness of the timestamped, attributed audit trail, the same infrastructure the *Regulatory Surveillance* and *Determinism and Replay* chapters treat as first-class engineering requirements. [U.S. Securities and Exchange Commission, *In the Matter of Athena Capital Research, LLC*, Administrative Proceeding File No. 3-16199, 16 October 2014.]

For the developer, the lesson is that an auction is not only a price-discovery algorithm but a *policed* one. The obligations this places on the system are concrete: disseminate imbalance and indicative-price information accurately and on schedule; enforce the entry and cancellation cut-offs for auction order types exactly; constrain imbalance-offsetting order types so they cannot themselves distort the print; and capture every auction message in the audit trail with the same rigour as continuous-session orders, because the closing print is precisely the number a regulator is most likely to ask you to reconstruct.

Key idea: The opening and closing auctions run the same maximum-executable-volume equilibrium algorithm as any call auction, but the closing print's role as a load-bearing financial benchmark drives everything distinctive about them: dedicated auction-only order types (MOC/LOC, imbalance-only, closing-offset), venue-specific named crosses with imbalance dissemination, a human backstop at NYSE, and an explicit surveillance and audit-trail burden aimed at the ever-present incentive to mark the close.

CHAPTER 11

Trading Sessions, The Day in the Life of a Market

A trading day is structured into distinct phases, each with different rules about what is allowed.

11.1 Pre-Open / Pre-Market

Before the formal opening, some exchanges accept orders for the day (limit orders, ATO orders, GTC orders from previous days). No matching occurs. This is the accumulation phase for the opening auction.

11.2 Opening Auction

As described above, orders collected, equilibrium computed, uncross executed, continuous trading begins.

11.3 Continuous Trading (the main session)

The normal matching mode. Orders arrive continuously, the engine matches them in real time, trades happen as soon as compatible orders meet. This is the primary trading session, NYSE runs from 9:30am to 4:00pm Eastern time, LSE from 8:00am to 4:30pm London time.

11.4 Intraday Auction (optional)

Some exchanges run a brief scheduled auction during the continuous session, typically at a fixed time such as noon or 1:00pm. These **intraday auctions** serve several purposes.

For exchanges that handle both liquid and illiquid instruments, an intraday auction gives participants a chance to trade illiquid stocks against a concentrated pool of interest, rather than relying on sparse continuous liquidity throughout the day. Large institutional orders that are difficult to execute in continuous trading can be "parked" to the intraday auction where counterparties are known to aggregate.

For derivatives markets, intraday auctions are also used as **volatility auctions**, if a futures contract moves rapidly during continuous trading and triggers a volatility interruption (not a full halt, but a brief pause), the exchange may resume through a short auction rather than returning immediately to continuous matching. Eurex, for example, uses *volatility interruptions*, short auctions triggered when a price moves outside a predefined range, as a softer alternative to a full circuit breaker halt; the London Stock Exchange runs a scheduled intraday auction at midday for its most liquid shares.

From the exchange system perspective, an intraday auction is mechanically identical to the opening or closing auction: orders accumulate during the auction call period, the engine computes the equilibrium price, and the uncross executes simultaneously.

11.5 Post-Close

After the formal close, several important processes run:

After-hours trading. Some venues, particularly US equity exchanges and alternative trading systems, run extended-hours sessions before the open and after the close. Volume and liquidity are substantially lower than during the regular session, spreads are wider, and price moves can be exaggerated. Not all order types are supported. The exchange may operate a separate "extended hours" trading mode with different rules from the main session.

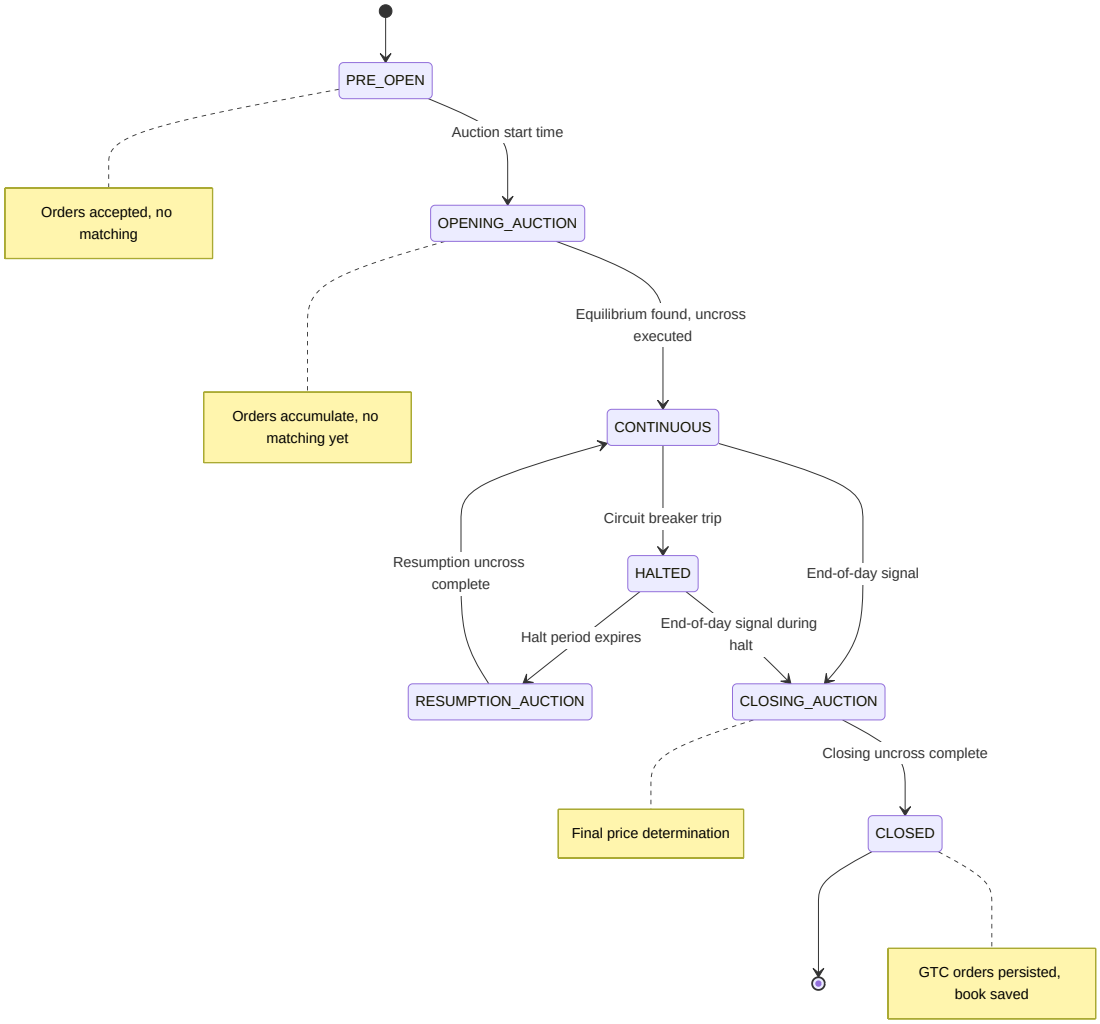
GTC order persistence. Good-Till-Cancelled orders that did not fill during the session must be saved to durable storage so they can be reloaded into the order book when the next session begins. This is a non-trivial operation: each GTC order must be written to a persistent store with all its parameters (symbol, side, price, quantity, participant ID, original submission timestamp). At start-of-day, these orders are reloaded before the pre-open period begins, and each is re-acknowledged to the originating participant. If a GTC order's participant has disconnected overnight, the exchange must decide whether to hold the order or cancel it.

End-of-day batch processes. The close of session triggers a cascade of batch processes: the official closing price is published and broadcast to downstream systems; daily P&L is calculated for all participants; clearing reports are generated; statistics (OHLCV, Open, High, Low, Close, Volume) are finalised and archived; surveillance reports are run against the day's audit trail. These processes are not in the matching engine's critical path, but they must complete before the next session begins. Their failure can delay the next day's open.

Book state save. The complete state of the order book, including all resting limit orders that will carry forward, is written to disk. This is the snapshot that will be used as the starting point for warm restart if needed.

11.6 State Machine

A well-designed exchange system models these transitions as a **state machine**, a finite set of states with explicit rules about which transitions are allowed. You cannot jump from PRE_OPEN directly to CLOSED without going through OPENING_AUCTION and CONTINUOUS. This prevents bugs where the engine enters an impossible state due to a software error or out-of-order messages. If the transition is not in the allowed list, the system rejects it.



Each arrow represents a permitted transition. Any attempt to trigger a transition not shown is rejected as invalid. This state machine is enforced in the exchange's session scheduler, which issues state change commands to the matching engine at the appropriate times.

CHAPTER 12

Putting It All Together

Let us trace the life of a complex event through the entire system using the vocabulary we have now built.

Scenario: It is 10:47am. Trading is in the CONTINUOUS session. A large institutional investor decides to sell 50,000 shares of AAPL aggressively using a limit order priced at $150.20, while the best bid in the book is $150.35.

Step 1: The investor's trading system sends the order to the exchange gateway (GW03). The gateway parses the FIX message, validates the format, and forwards it to the matching engine.

Step 2: The engine receives the order, assigns it a unique order ID (say, `f3a7-...` UUID) and a nanosecond timestamp. Status: NEW. The engine acknowledges to GW03.

Step 3: The engine checks the order type: LIMIT SELL at $150.20. Since the best bid is $150.35 (above the limit price), this order will immediately sweep the buy side.

Step 4: The engine sweeps. It fills against resting buy orders, starting at $150.35, then $150.30, then $150.25, and it would continue down to $150.20 until either the 50,000 shares are exhausted or all bids at $150.20 and above are consumed, unless something stops it first (see Step 7). Along the way: - A market maker's GTC bid for 1,000 shares at $150.35 is completely consumed by the sweep. Status: NEW → FILLED. - An iceberg order's visible 500-share peak at $150.30 gets filled, triggering replenishment from the hidden reserve. - Several DAY limit orders at $150.25 get partially or fully filled.

Step 5: After each fill, the engine updates `last_trade_price`. After sweeping $150.35 bids (aggressive side is SELL), `last_sell_price` is also updated to $150.35.

Step 6: The engine checks dormant stops. Does the new `last_trade_price` trigger any waiting stop orders? Suppose a buy stop at $150.40 was waiting, but the price just moved down, not up, so it does not trigger. A sell stop at $150.30 was waiting, the last trade price of $150.25 is now below $150.30, so this sell stop triggers, converting to a market sell order and immediately joining the sweep.

Step 7: Before each fill, the engine checks the price against the stock's price band. Suppose yesterday's closing auction cleared at $153.30 and the band for this stock is ±2%. The lower band is $153.30 × 0.98 = $150.234, which rounds *up* to $150.24, the lowest valid price inside the band. The fills at $150.35, $150.30 and $150.25 were inside the band, but the next resting bids, at $150.20, lie below it. Rather than print a trade outside the band, the engine stops the sweep and a circuit breaker trips. The engine transitions to HALTED state, all market maker quotes are cancelled (stale quotes cannot be maintained during a halt), and a halt notification is published.

Step 8: All fill events are published to the PUB socket. GW03 receives fill notifications for the institutional investor's order (now showing PARTIAL status, 50,000 shares ordered, 38,000 filled before the halt, 12,000 remaining). Each market maker whose bid was hit receives their fill notification on their respective gateways.

Step 9: The drop copy feed receives a copy of all these fill events, tagged with sequence numbers, and forwards them to the clearing broker's risk system.

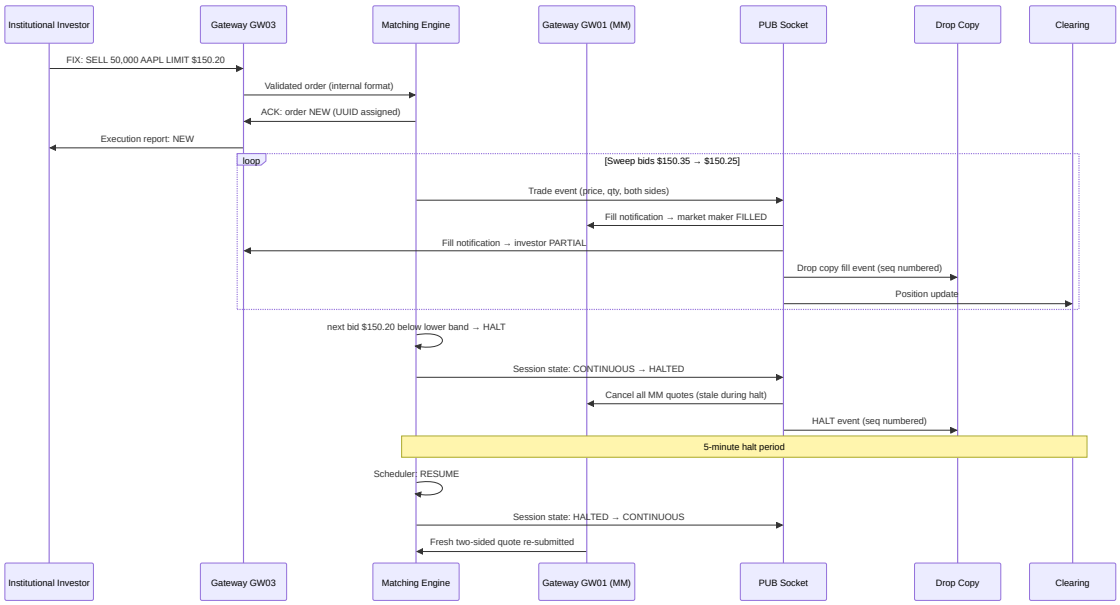
Step 10: The clearing process updates positions. The institutional investor's AAPL position decreases by 38,000. The various market makers' positions increase accordingly. P&L is updated for all parties.

Step 11: After the halt period (say, 5 minutes), the scheduler sends a RESUME command. The engine transitions back to CONTINUOUS trading (or, if specified, initiates a brief resumption auction first). Market makers re-quote. Trading resumes.

Step 12: The 12,000-share remainder of the institutional investor's sell order continues resting on the ask side at $150.20 and will fill when buyers willing to pay $150.20 or more appear.

In this single scenario, we used: limit orders, GTC orders, iceberg orders, fill notifications, last trade price, last sell price, stop orders, circuit breakers, collar bands, automatic market maker quote cancellation on halt, drop copy, clearing, P&L, trading sessions, and the state machine. Every piece of vocabulary we have discussed had a role to play.

The message flow between components in steps 1–10 can be visualised as follows:



CHAPTER 13

Implied Orders and Synthetic Liquidity

This concept trips up almost every developer who meets a derivatives exchange for the first time, and for good reason: it is the first place in this book where a single incoming order can trade against orders resting in *other* order books. The arithmetic is school-level, but it arrives wrapped in an unusual sign convention, the quantity rules are less obvious than they look, the priority rules are subtle, and the engineering consequences are severe. This chapter therefore climbs in explicit levels. Each level introduces one new idea and works it through with numbers before the next level begins. The numbers carry over from level to level, so read every example even when the idea feels obvious.

By the end of the chapter you should be able to write a correct first-generation implied-matching module: compute implied prices and quantities, execute them atomically, keep them consistent as the books change, and test them. The chapter after this one then asks the obvious follow-up question, *why not imply from implied orders, and from those, and so on?*, and shows why every exchange stops after a few steps.

One promise before we begin, which the rest of the chapter proves over and over: **implied orders do not create liquidity out of nothing.** An implied order is a different *view* of commitments that already rest in other books. It never adds a single lot to the total that can be traded; it only makes existing lots reachable from one more place. Hold on to that sentence. Every level below is, in the end, a demonstration of it.

13.1 Level 0: Futures in One Page

You do not need a finance background for this chapter, but you do need four facts about futures.

1. A futures contract is a promise, not a purchase. A futures contract is a standardised agreement to buy or sell a fixed quantity of something (crude oil, wheat, a stock index) at a price agreed *today*, for delivery on a fixed date in the *future*. Nothing changes hands at the moment of the trade except a good-faith deposit (the *margin*, covered in the clearing chapter of Part III). If you agree to buy, you are **long**; if you agree to sell, you are **short**. If the price later rises, the long side gains and the short side loses the same amount, and vice versa. Futures are a zero-sum game between the two sides of each trade.

2. Everything in this chapter uses crude oil. The running example is modelled on NYMEX WTI Light Sweet Crude Oil futures (symbol CL) at CME Group. One contract, also called one **lot**, covers **1,000 barrels**. Prices are quoted in **US dollars per barrel**, and the minimum price step (the **tick**, see the *Tick Sizes* chapter) is **$0.01 per barrel, worth $10 per contract** [CME Group, NYMEX Rulebook Chapter 200]. So when this chapter says "a price of $75.00", it means $75.00 per barrel, and one lot at that price represents $75,000 worth of oil.

3. Every delivery month is a separate instrument with its own order book. January crude and February crude are *different products*: different delivery dates, different buyers and sellers, different prices. An exchange lists many months at once, and each month has its own independent order book exactly like the ones in the *Order Book* chapter. Throughout this chapter we use two of them, **January** (the *near* month, delivered sooner) and **February** (the *far* month).

4. Traders often care about the *difference* between two months more than about either price. Three common reasons:

1. **Rolling a position.** A trader who is long January and wants to stay long oil beyond January's expiry must sell January and buy February. Doing that as two separate trades exposes them to the market moving between the two fills.
2. **Relative-value trading.** A trader may believe February is too expensive *relative to* January without having any opinion at all on where oil itself is going. The only thing they want exposure to is the difference.
3. **Margin efficiency.** Long January plus short February is largely protected against the overall oil price moving (if oil rises, one leg gains roughly what the other loses), so clearing houses require much less margin for the pair than for two unrelated positions. CME's margin methodology grants explicit *intra-commodity spread credits* for exactly this reason.

The naive way to get exposure to the difference is to submit two ordinary orders and hope both fill. That carries **leg risk**: one leg fills, the other does not, and the trader is left holding an outright position they never wanted. Exchanges solved this by listing the difference itself as a tradeable instrument, with its own order book. That instrument is where our story starts.

13.2 Level 1: The Spread Instrument and Its Sign Convention

A **calendar spread** is a single listed instrument whose price is *defined* as the difference between two delivery months. Throughout this chapter we use the convention CME and most futures venues use for calendar spreads:

```
spread price   S  =  Near-month price  −  Far-month price
                    (here: S = January − February)
```

Buying one spread means buying one lot of the near month *and* selling one lot of the far month, simultaneously, as one instruction. **Selling one spread** is the mirror image: sell the near month, buy the far month. CME's own description of Treasury calendar spreads states this directly: "buying the calendar spread entails buying the nearby month … and selling the deferred month" [CME Group, *Treasury Futures Calendar Spreads*, 2016].

Why this convention makes sense

The convention is chosen so that **buying the spread is a bet that the number S goes up**, exactly as buying anything else is a bet that its price goes up. Work it through:

- You buy one Jan/Feb spread at S = −$2.00. Concretely, suppose you buy January at $75.00 and sell February at $77.00 (75.00 − 77.00 = −2.00).
- A week later January is $76.00 and February is $77.50, so S = 76.00 − 77.50 = **−$1.50**. The spread price *rose* from −2.00 to −1.50.
- Your January long gained $1.00 per barrel. Your February short lost $0.50 per barrel. Net: **+$0.50 per barrel**, which is exactly the change in S: (−1.50) − (−2.00) = +0.50.
- Per lot that is $0.50 × 1,000 barrels = **$500**.

Notice that the outright price of oil rose too, but that did not matter; only the difference did. That is the whole point of trading the spread.

The four things that confuse every beginner

The convention is logical, but it produces four effects that look wrong the first time you see them. Name them now, and they will not ambush you later.

Confusion 1: spread prices are routinely negative. When the far month is more expensive than the near month (a market in **contango**, common in oil when storage is plentiful), S is negative. When the near month is more expensive (**backwardation**), S is positive. Both are completely normal:

Spread price	Meaning	A spread *buyer* does, for example
S = −$2.00	January trades $2.00 **below** February	buys Jan at $75.00 and sells Feb at $77.00

Spread price	Meaning	A spread *buyer* does, for example
S = +$0.50	January trades $0.50 **above** February	buys Jan at $76.50 and sells Feb at $76.00

A negative price is not an error code, and your engine must not treat it as one. Nor are negative prices limited to spreads. On 20 April 2020, with storage at the Cushing, Oklahoma delivery point nearly full and the May 2020 WTI contract one day from expiry, May crude *settled at −$37.63 per barrel* [CFTC, Interim Staff Report, November 2020], while the June contract closed at $20.43 [Congressional Research Service, IN11354]. Holders of May contracts were, in effect, paying people to take oil off their hands, and the May/June spread was roughly −$58 per barrel. CME had warned its clearing members in early April that some energy contracts could trade below zero and had opened a test environment for negative prices [CME Clearing Advisories 20-152 and 20-160, April 2020]. Not every system was ready. The CFTC later found that at one large retail broker "negative prices were not displayed to customers and customers were unable to place orders with negative-priced limit orders", and ordered it to pay a $1.75 million penalty plus about $82.6 million in restitution to customers [CFTC press release 8432-21, 28 September 2021]. An assumption of `price > 0`, buried in one validation function, turned out to be a nine-figure bug.

Confusion 2: "higher" and "better" need care. −1.50 is *higher* than −2.00. A spread bid of −1.90 is therefore a *better* (more generous) bid than −2.00, and a spread ask of −2.10 is a better (cheaper) ask than −2.00. Price-time priority works exactly as in any other book, provided your comparison operators are signed. Storing spread prices as signed integer tick counts (−200 ticks for −$2.00) makes this automatic.

Confusion 3: buying the spread *sells* the far month. A spread *buyer* is a *seller* of February. So a resting spread **bid** is, among other things, a standing offer to **sell** February. This side-flip is the single most important fact in this chapter, and it is where most implementation bugs come from.

Confusion 4: subtracting a negative number adds. Because S = Jan − Feb, it follows that Feb = Jan − S. With S = −2.00 and Jan = 75.00, Feb = 75.00 − (−2.00) = **77.00**. Every time you see "minus a spread price" in this chapter, check the sign twice.

> **Convention warning.** Near-minus-far is the norm for calendar spreads on CME, and most futures venues use the same convention, but it is not a law of nature. Inter-commodity spreads, ratio spreads, and some venues' products define their price differently (some as a ratio of legs, some with legs weighted 2:3, and so on). Always read the product's definition before writing a single formula.

Leg prices

When a spread trades at −$2.00, the clearing house still needs a concrete price for each leg, because positions, profit and loss, and margin are tracked per month. The exchange assigns leg prices using a documented rule. On CME, for example, one leg is anchored to a recent reference price for that month (its latest trade price, a current price indication, or its settlement price) and the other leg is derived so that the two legs differ by exactly the spread price [CME Group, *Treasury Futures Calendar Spreads*]. The spread traders' economics depend only on the difference; the leg prices exist so every system downstream can do per-instrument bookkeeping. We will return to leg prices at Level 9, where they stop being a formality.

13.3 Level 2: The Developer's Model, Orders as Vectors

The rest of the chapter becomes mechanical, and much easier to implement correctly, once you describe every order with two numbers-with-structure.

The position vector v. For each instrument, how many lots does *one unit* of this order buy (+1) or sell (−1)?

The cash amount c. How much money (per barrel) does one unit of this order *pay*? A negative c means the order *receives* money.

For the four basic kinds of order:

Order	Position vector v	Cash paid c (per barrel)
Buy January at price P	(Jan +1)	+P
Sell January at price P	(Jan −1)	−P
Buy Jan/Feb spread at price S	(Jan +1, Feb −1)	+S (pays the Jan price, receives the Feb price; Jan − Feb = S)
Sell Jan/Feb spread at price S	(Jan −1, Feb +1)	−S

The one rule that powers everything else is this:

> **Combining orders adds their vectors and adds their cash.** If a set of resting orders is executed together, the combined effect on the world is the sum of their position vectors, and the combined money flow is the sum of their cash amounts.

An **implied order** is simply a set of resting orders whose vectors add up to something that *looks like a single order in some other book*. Here is the canonical case. Take one resting January ask at $75.00 and one resting spread bid at −$2.00:

```
  Sell Jan at 75.00          v = (Jan −1)            c = −75.00
+ Buy spread at −2.00        v = (Jan +1, Feb −1)    c = −2.00
  ------------------------------------------------------------
  Combined                   v = (Feb −1)            c = −77.00
```

The January legs cancel. What remains sells one February lot and receives $77.00: that is precisely **an offer to sell February at $77.00**, even though nobody has placed an order in the February book. This is the complete theory of implied pricing. Everything else in this chapter (quantities, priority, rounding, market data) is engineering around this addition.

Two consequences follow immediately and should be memorised:

1. **Implied prices come from adding cash amounts**, so they are *exact* sums and differences of real order prices, no averaging, no estimation.
2. **Legs must cancel against an order on the opposite side.** The January leg of the spread bid (+1) cancels only against a January *sell* (−1). A January *bid* would add +1 to +1 and produce a two-lot January purchase, which is not an order in any book. This is the formal version of "an implied order needs every leg to be executable".

The six first-generation implied prices

Apply the rule to every pair of orders that can cancel a leg, and exactly six implied quotes appear. This table is the reference for the whole chapter.

Implied quote	Ingredients (both must be resting)	Vector check	Formula
Feb **ask**	Jan ask + Spread **bid**	(Jan −1) + (Jan +1, Feb −1) = (Feb −1)	Jan_ask − S_bid
Feb **bid**	Jan bid + Spread **ask**	(Jan +1) + (Jan −1, Feb +1) = (Feb +1)	Jan_bid − S_ask
Jan **ask**	Feb ask + Spread **ask**	(Feb −1) + (Jan −1, Feb +1) = (Jan −1)	Feb_ask + S_ask
Jan **bid**	Feb bid + Spread **bid**	(Feb +1) + (Jan +1, Feb −1) = (Jan +1)	Feb_bid + S_bid
Spread **ask**	Jan ask + Feb **bid**	(Jan −1) + (Feb +1) = (Jan −1, Feb +1)	Jan_ask − Feb_bid
Spread **bid**	Jan bid + Feb **ask**	(Jan +1) + (Feb −1) = (Jan +1, Feb −1)	Jan_bid − Feb_ask

*(Reading the "Spread ask" row: the combined vector is (Jan −1, Feb +1), which is what one unit of a spread **sell** looks like; a resting set of orders that together behave like a seller is an **ask**. Its price is $-c = -(-Jan_ask + Feb_bid) = Jan_ask - Feb_bid$.)*

Check the sign-flip from Confusion 3 in the table: an implied February **ask** is built from a spread **bid**. Near-month quotes keep their side; far-month quotes flip it.

Terminology. CME defines the two families as follows [CME Group Client Systems Wiki, *Implied Orders*]:

- **Implied IN**: "an order CME Globex identifies as existing in the spread market based on orders in the outright market" (the bottom two rows: two outrights imply a spread quote).
- **Implied OUT**: "an order CME Globex identifies as existing in the outright market based on orders in the spread market" (the top four rows: a spread plus an outright imply an outright quote).

A mnemonic: implied **in**to the spread book, implied **out** of the spread book. Informal industry usage is not always consistent, so when reading another venue's documentation, check its definitions rather than assuming.

How to never get a formula wrong. Do not memorise the table. When you need a formula, write down the two vectors and add them. If the unwanted legs cancel, you have an implied order and its price falls out of the cash sum. If they do not cancel, there is no implied order, however tempting the arithmetic looks. Exercise E2 at the end of the chapter is a deliberate trap built on exactly this point.

13.4 Level 3: Implied OUT, a Complete Worked Example

A spread order plus an outright order jointly produce a quote in a book that may otherwise be completely empty.

The books before anything happens:

Book	Side	Price	Lots	Who
January (outright)	Ask	\$75.00	50	Trader A
Jan/Feb spread	Bid	−\$2.00	30	Trader B
February (outright)	–	*empty*	–	–

Trader A will sell up to 50 January lots at \$75.00 or more. Trader B will buy up to 30 spreads at −\$2.00 or less, meaning B will buy January and sell February whenever February can be sold for at least \$2.00 more than January costs.

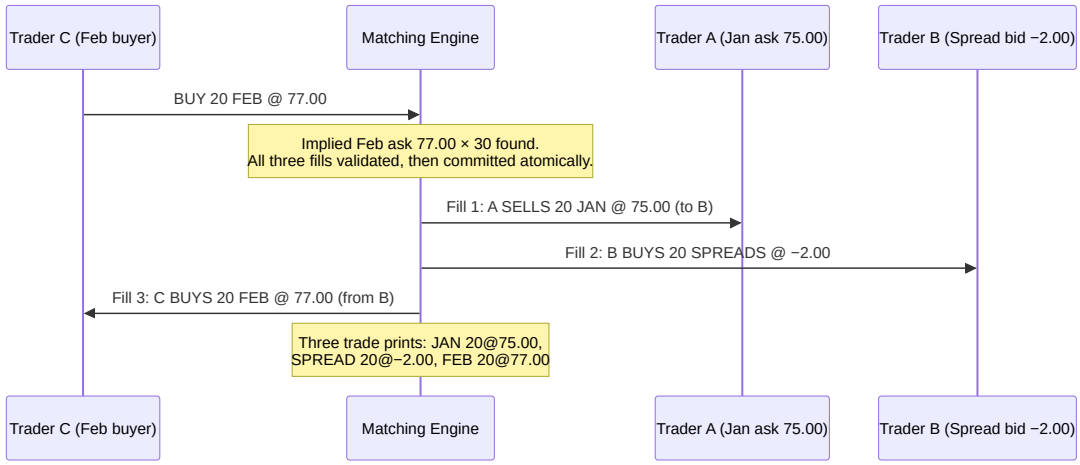
The operational question. Could anyone, right now, firmly *buy* February, even though the February book is empty? Yes. Trader B is willing to sell February, *provided* B can simultaneously buy January \$2.00 cheaper. Trader A is willing to sell January at \$75.00. Chain the two commitments: B buys January from A at \$75.00, so B can sell February at \$75.00 − (−\$2.00) = **\$77.00**. This is row one of the table: implied Feb ask = Jan_ask − S_bid = 75.00 − (−2.00) = 77.00.

Quantity is capped by the scarcer ingredient: min(50, 30) = **30 lots**. B only wants 30 spreads, however much January A offers.

The engine therefore publishes in the February book:

Book	Side	Price	Lots	Source
February	Ask	\$77.00	30	**Implied** (A's January ask + B's spread bid)

Execution. Trader C submits an order to buy 20 February at \$77.00. Matching it against the implied quote requires three fills that happen *together or not at all*:



Check that every party got exactly what its order asked for, and nothing it did not ask for:

- **A** sold 20 January at \$75.00, precisely A's limit.
- **B** bought 20 spreads at −\$2.00: long 20 January at \$75.00 and short 20 February at \$77.00; $75.00 - 77.00 = -2.00$. B's margin is computed on the much smaller spread risk.
- **C** bought 20 February at \$77.00, precisely C's limit. C need not know, and on most venues cannot tell from the fill, that the counterparty's February sale was part of a spread.

And check the vector bookkeeping: A's (Jan −20), B's (Jan +20, Feb −20) and C's (Feb +20) sum to zero in every instrument, and the cash sums to zero too. Using "cash paid" per barrel: A pays $-(20 \times 75.00) = -1{,}500$, C pays $+(20 \times 77.00) = +1{,}540$, and B pays $20 \times (-2.00) = -40$, that is, B *receives* the \$2.00 difference on each of its 20 spreads. $-1{,}500 + 1{,}540 - 40 = 0$ (multiply by 1,000 barrels per lot for real dollars). Nothing was created and nothing was lost. Every correct implied execution balances to zero like this, and your test suite should assert it.

The books after the match:

Book	Side	Price	Lots remaining
January (outright)	Ask	\$75.00	30 *(was 50)*
Jan/Feb spread	Bid	−\$2.00	10 *(was 30)*
February (implied)	Ask	\$77.00	**10** = min(30, 10)

13.5 Level 4: Implied IN, Two Outrights Produce a Spread Quote

Now the other direction: the *spread* book is empty, and the outright books are live.

The books:

Book	Side	Price	Lots	Who
January (outright)	Bid	\$74.40	25	Trader D
February (outright)	Ask	\$76.60	40	Trader E
Jan/Feb spread	–	*empty*	–	–

Add the vectors: D's January bid is (Jan +1) with c = +74.40; E's February ask is (Feb −1) with c = −76.60. Together: (Jan +1, Feb −1), which is the shape of a spread **buy**, paying $74.40 - 76.60 =$ **−\$2.20**. The two outrights jointly *bid* −\$2.20 for the spread.

In words: anyone who wants to **sell** the spread (sell January, buy February) can do it right now by selling January to D at \$74.40 and buying February from E at \$76.60, which is a spread sale at $74.40 - 76.60 = -2.20$.

```
Implied S_bid = Jan_bid - Feb_ask = 74.40 - 76.60 = -2.20
Quantity      = min(25, 40)       = 25 lots
```

Trader F submits an order to sell 10 spreads at −$2.20. The engine executes atomically:

1. **January fill:** F sells 10 January at $74.40 to D.
2. **February fill:** F buys 10 February at $76.60 from E.
3. **Spread print:** F sold 10 spreads at 74.40 − 76.60 = −$2.20 .

D and E each received an entirely ordinary outright fill at their own limit price. Afterwards the January bid has 15 lots left, the February ask has 30, and the implied spread bid is −$2.20 × min(15, 30) = 15.

The *opposite* implied quote needs the opposite ingredients. An implied spread **ask** requires a January *ask* and a February *bid* (Jan_ask − Feb_bid). With only D and E in the books, no implied spread ask exists: add the vectors of D and E any other way and nothing cancels.

13.6 Level 5: The Promise Kept, Implied Orders Create No Liquidity

It is time to make the introduction's promise precise, because it is the single most important idea in the chapter, and because misunderstanding it leads directly to the most dangerous bugs.

Argument 1: an implied trade is a trade anyone could have done by hand. Go back to Level 3. Trader C wanted February and could have assembled it manually, without any implied functionality:

1. Sell 20 spreads to Trader B at −$2.00. (C now: sell 20 Jan, buy 20 Feb, and pays −2.00, i.e. *receives* $2.00 per barrel on the difference.)
2. Buy 20 January from Trader A at $75.00. (C's January sale and purchase cancel.)

C's net result: long 20 February, having paid $75.00 + $2.00 = $77.00 per barrel. Exactly the implied trade, same counterparties, same prices. The implied engine did not find any liquidity that was not already reachable. It did the legwork **atomically**, eliminating C's leg risk (the danger that A's January offer disappears between C's two trades) and letting C trade with a single ordinary order.

Argument 2: the same lot must never be counted twice. In Level 3's books, A's 50 January lots appear on screens in *two places*: as 50 lots in the January book, and (30 of them) inside the 30-lot implied February ask. It is tempting to add up what the screens show and conclude that 80 lots are for sale. That is wrong. If a January buyer takes all 50 of A's lots, the implied February ask **disappears**, because one of its ingredients is gone. The displayed quantities across related books are *alternative* uses of the same underlying commitments, not additional ones.

Argument 3: the removal test. Cancel A's January order and the implied February ask vanishes in the same processing step: there is nothing left to build it from. Cancel B's spread order and it vanishes too. An implied quote that survives the cancellation of one of its ingredients is not a harmless display glitch; it is the engine advertising a price it cannot honour.

Argument 4: conservation in the arithmetic. In Level 3, before the match, the system held 50 committed January lots and 30 committed spread lots. After C's trade, it held 30 January lots and 10 spread lots, and C held 20 February lots that each consumed *exactly one* of A's lots and *exactly one* of B's spreads. The vector sum of every trade is zero (Level 3). No lot was ever used twice, and none was conjured.

> **Implied orders move liquidity between books; they never add to it.** Every implied lot is a real lot from a real order, reachable by one more route. When it trades through one route, it is gone from all the others.

13.7 Level 6: Quantity Rules, min(), Shared Legs, Depth, and Recomputation

The min() rule from Level 3 generalises, and the generalisation is where implementations start to acquire bugs.

Shared legs. Suppose a second spread bidder joins Level 3's original books:

Book	Side	Price	Lots	Who
January	Ask	\$75.00	50	Trader A
Spread	Bid	−\$2.00	30	Trader B
Spread	Bid	−\$2.00	15	Trader H

Both spread bids combine with the *same* January ask to imply February asks at \$77.00. The correct implied quantity is min(50, 30 + 15) = **45**. Here that happens to equal the naive calculation min(50, 30) + min(50, 15) = 45, which is exactly why the naive calculation survives code review. Now shrink A's order to 40 lots. The correct answer is min(40, 45) = **40**. The naive "pairwise min, then sum" still says 30 + 15 = 45 and over-advertises by 5 lots that do not exist. A's January order is a **shared leg**: every construction that uses it draws from the same 40 lots.

The general statement: *the implied quantity is the largest quantity that can be routed through the contributing orders simultaneously, with no order used beyond its own size.* For a single chain that is the smallest ingredient; when chains share an order, the shared order caps their total. (Readers who know graph algorithms will recognise a maximum-flow problem; for first-generation implieds between two months it reduces to the simple sums and mins shown here.)

Depth beyond the best price. Books have more than one price level, and so do implied books. Suppose:

Book	Side	Price	Lots
January	Ask	\$75.00	10
January	Ask	\$75.01	20
Spread	Bid	−\$2.00	15
Spread	Bid	−\$2.02	10

Each implied February lot costs (January ask) − (spread bid), so the cheapest February lots come from pairing the cheapest January lots with the most generous spread bids. The correct implied February ask ladder is built by walking both ladders together, best against best, exactly like a merge of two sorted lists:

Implied Feb ask	Lots	Built from
\$77.00	10	Jan 75.00 (all 10) with spread −2.00 (10 of 15)
\$77.01	5	Jan 75.01 (5 of 20) with spread −2.00 (remaining 5)
\$77.03	10	Jan 75.01 (10 of remaining 15) with spread −2.02 (all 10)

Total: 25 lots, which equals min(30 January lots, 25 spread lots), as conservation demands. Notice what is *missing*: a naive implementation that pairs every January level with every spread level independently would also publish **\$77.02 × 10** (Jan 75.00 with spread −2.02). That quote is fictitious: the January lots at \$75.00 are already committed to the \$77.00 level. Buy everything the naive ladder shows and the engine runs out of January oil before the order is filled. Because a correct implied depth ladder is *path-dependent* in this way, venues typically compute and publish only the top one or two implied levels (see Level 10).

Recompute, never decrement. After any event that touches a contributing order (a fill, a cancel, an amendment, a new order at a better price), the engine re-derives the affected implied quotes from the current state of the real books. Level 3's post-trade figure (implied 10 = min(30, 10)) is a recomputation, not "30 minus 20". The two happen to agree there, but Exercise E6 shows a case where they do not.

13.8 Level 7: Priority, Real Orders Before Implied Orders

Suppose the February book contains both a real resting ask and an implied ask at the same price:

February book	Price	Lots	Source
Ask	\$77.00	5	**Real**, Trader G, resting since 09:31
Ask	\$77.00	30	Implied (A + B, as in Level 3)

Trader C's order to buy 20 at \$77.00 arrives. Who fills?

On CME, the answer for futures is explicit: "Outrights (generation 0) always trade first, then followed by spreads (1st generation)" [CME Group Client Systems Wiki, *Futures Implied Order Matching Priority*], and "implied quantity in futures markets does not have time priority" [CME, *Implied Orders*]. So at the same price, **real orders trade before implied orders**, no matter how long the implied quote has been displayed. C's order therefore fills 5 lots against G (an ordinary two-party trade) and 15 lots through the implied construction (the three-fill atomic execution from Level 3).

The rationale mirrors the displayed-before-hidden logic from the *Order Types* chapter: a participant who committed directly in this book is rewarded ahead of a price that is only a reflection of commitments elsewhere. There is also an engineering benefit: a two-party fill is cheaper and simpler than a multi-book atomic one, so using real liquidity first minimises the expensive path. When several *implied* constructions compete at the same price, CME breaks the tie with a documented sequence of criteria (strategy type, product codes, leg expirations, number of legs, and finally instrument identifiers), not by time. Treat "real before implied at the same price" as the strong default, and each product's rulebook as the authority.

Two further rules on CME are worth building in from day one [CME, *Implied Orders*]:

- **"Implication requires at minimum two orders in related markets in the proper combination."** An implied quote is never built from a single order.
- **"Implied bids do not trade against implied offers."** Every trade needs at least one real, *incoming* order as the aggressor. Implied quotes are only ever passive: they wait to be hit. We will see why this rule is natural in Level 8.

13.9 Level 8: Price Improvement and Closing Cross-Book Gaps

Implication does not just fill empty books; it can *beat* the real book. Suppose:

Book	Side	Price	Lots
January	Ask	\$74.95	40
Spread	Bid	−\$2.00	25
February	Ask (real)	\$77.00	60

The implied February ask is $74.95 - (-2.00) = \mathbf{\$76.95}$, which is 5 ticks (\$0.05, or \$50 per lot) *better* than the real February ask. An incoming order to buy 20 February at \$77.00 fills all 20 at \$76.95 through the implied construction, and the real \$77.00 seller is not touched. The buyer receives price improvement exactly in the sense of the *Order Types* chapter: they were willing to pay \$77.00 and paid \$76.95.

Closing gaps that would otherwise be free money. Now add one more resting order: Trader K bids \$76.97 for 10 February lots. Is anything wrong with these books?

Book	Side	Price	Lots
January	Ask	\$74.95	40
Spread	Bid	−\$2.00	25
February	Bid (real, Trader K)	\$76.97	10

Yes: they are **crossed across books**. K will buy February at $76.97, while the January seller and B together will sell February at $76.95. In a single book, a bid above an ask can never rest (the incoming one would have traded). Across books, without implied functionality, it can. And then it is free money for whoever notices first. An arbitrageur could:

1. Sell 10 spreads to B at −$2.00 (sell January at some price p, buy February at p + 2.00),
2. buy 10 January from A at $74.95, and
3. sell 10 February to K at $76.97.

The arbitrageur's January trades cancel, the February trades cancel, and the cash is p − 74.95 + 76.97 − (p + 2.00) = **+$0.02 per barrel**, $20 per lot, with no risk, provided all three fills happen before anyone else acts.

With implied functionality this state can never arise, because it could only be reached through an *arrival*, and the arriving order is matched when it arrives. Suppose K's bid rested first and B's spread bid is the last of the three orders to arrive. At that moment the engine already publishes an implied spread **ask** built from A's January ask and K's February bid: Jan_ask − Feb_bid = 74.95 − 76.97 = **−$2.02**. B's incoming bid of −$2.00 is willing to pay up to −2.00, and −2.02 is cheaper, so B trades immediately:

1. B buys 10 January from the January seller at $74.95.
2. B sells 10 February to K at $76.97.
3. B's spread fill prints at 74.95 − 76.97 = **−$2.02**, two ticks better than B's limit.

K's bid is now used up, so the implied spread ask disappears, and B's remaining 15 spreads rest as a bid at −$2.00 (where, together with the 30 January lots still offered, they again imply a February ask at $76.95 × 15).

The $0.02 that an arbitrageur would have pocketed goes instead to B, a real participant, as price improvement. This is the deeper purpose of implied functionality: it keeps the whole *family* of related books mutually consistent, so that cross-book gaps are closed by the exchange on behalf of resting orders rather than harvested by the fastest outsider.

It also explains the rule "implied bids do not trade against implied offers". If an implied bid ever crossed an implied offer, some set of *real* orders would be crossed through a cycle of books. But every real order was matched against everything it could reach at the moment it arrived. So a crossed cycle can only exist if the engine missed a match at an earlier arrival, a bug, not a situation the matching rules need to handle.

13.10 Level 9: Tick Alignment and Leg Prices, Where Arithmetic Meets Reality

The formulas of Level 2 are exact arithmetic; real books live on price grids. Two complications follow, both descendants of the *Tick Sizes and Fractional Ticks* chapter.

Spread grids can differ from outright grids. Some venues list a spread on a finer tick grid than its legs, because spreads move less than outright prices. The moment the grids differ, an implied outright price computed from a spread price may land *between* valid outright ticks. For illustration, take a hypothetical product whose outright tick is 0.25 and whose spread tick is 0.05:

```
January ask        100.00   (valid outright tick)
Spread bid          −0.35   (valid spread tick)
Implied Feb ask  = 100.00 − (−0.35) = 100.35   ← not a multiple of 0.25
```

The engine cannot print a February trade at 100.35. The standard remedy is to **round outwards**, meaning in the direction that is worse for the party trading against the implied quote: an implied ask is rounded *up*, an implied bid *down*. So the implied Feb ask is published at **100.50**. Carruthers' analysis of CME's Treasury inter-commodity spreads describes exactly this rule: "For price-priority purposes, implied prices are rounded outwards to the nearest tick", with direct liquidity at the rounded price keeping priority over implied liquidity [Carruthers, *Inter-Commodity Spreads and Implied Pricing*, 2018].

Who gets the 0.15 difference when someone buys February at 100.50? Follow the money through the construction: the February buyer pays 100.50; A sells January at 100.00; so B's spread executes at 100.00 − 100.50 = **−0.50**. B was willing to buy the spread at −0.35 or lower, and −0.50 is lower, so B receives the

0.15 as price improvement. Every party trades at or better than its own limit, and all prints are on their grids (−0.50 is a valid 0.05 spread tick). Rounding outwards is what guarantees this: it never forces any real order to trade worse than its limit.

Whatever the rounding rule, it must be documented and implemented in **exactly one function**, shared by matching, market-data publication, and clearing. Two components that round differently will disagree about which prices exist.

Leg prices must land on-tick and inside limits. Every leg of every implied execution needs a price that is (a) a valid tick for its instrument, (b) consistent with every order involved, so that each spread's legs still differ by exactly its trade price, and (c) inside that instrument's price limits for the session. In the first-generation examples above, the real outright orders pin down the leg prices naturally. In the multi-layer constructions of the next chapter, some legs trade only *between two spread orders*, where no real outright price exists, and the engine must manufacture one by rule. Venues publish these leg-pricing algorithms. An engine that assigns an off-tick or out-of-limits leg price produces trades that clearing systems reject, a far worse failure than refusing the implied match up front.

13.11 Level 10: How Implied Liquidity Appears in Market Data

Participants need to know that implied liquidity exists, and sophisticated participants need to know *which part* of a displayed quantity is implied, because an implied quote can vanish for reasons invisible in its own book (one of its ingredients in another book was cancelled or filled). Venues handle this in one of three ways:

1. **Publish implied quotes as a separate book** next to the real book. CME's MDP 3.0 market data feed disseminates a "2-deep best bid and ask … for each implied-eligible futures contract" as a distinct implied book [CME Group Client Systems Wiki, *MDP 3.0 – Implied Book*], and documents how a client should consolidate it with the real book.
2. **Aggregate real and implied quantity** into one displayed number per price level.
3. **Show only the real book**, and let implied liquidity surface only at execution.

A market-data consumer must know which policy a venue uses, or its reconstructed book will disagree with the venue's, a classic source of "our depth doesn't match the exchange's" support tickets. And recall Level 6: implied depth beyond the first level or two is path-dependent, which is one reason venues keep the published implied book shallow.

13.12 Engineering Deep-Dive

With the mechanics established, the engineering can be stated precisely.

A minimal first-generation algorithm

For one pair of months (January, February) and their spread, the whole of implied matching fits in a page of pseudocode. Prices are signed integer ticks throughout.

```
# Real books: jan, feb, spr. Each side is a price-time-ordered queue of real orders.

function implied_quotes():                    # recomputed after EVERY event in the group
    return {
      FEB_ASK: pair(jan.asks, spr.bids, price = a.price - b.price),   # Jan_ask - S_bid
      FEB_BID: pair(jan.bids, spr.asks, price = a.price - b.price),   # Jan_bid - S_ask
      JAN_ASK: pair(feb.asks, spr.asks, price = a.price + b.price),   # Feb_ask + S_ask
      JAN_BID: pair(feb.bids, spr.bids, price = a.price + b.price),   # Feb_bid + S_bid
      SPR_ASK: pair(jan.asks, feb.bids, price = a.price - b.price),   # Jan_ask - Feb_bid
      SPR_BID: pair(jan.bids, feb.asks, price = a.price - b.price),   # Jan_bid - Feb_ask
    }
    # pair() walks both ladders best-first (the merge of Level 6), emitting
    # (implied_price, qty, [ingredient orders]) and never using an order's quantity twice.
    # Round each implied price outwards to its book's grid (Level 9).

function on_incoming(order):                  # order is always REAL: implied never aggresses
    while order.qty > 0:
        real = best_real_opposite(order.book)
        impl = best_implied_opposite(order.book, implied_quotes())
        best = better_price(real, impl, tie -> real)   # real before implied (Level 7)
        if best is none or not marketable(order, best): break
        if best is real:
            fill_two_party(order, best)
        else:
            q = min(order.qty, impl.qty)
            validate_all_legs(impl.ingredients, q)      # ticks, limits, risk: all or nothing
            commit_atomically(order, impl.ingredients, q)  # N fills, one sequence number
        # loop re-reads books and recomputes implied quotes: never decrements them
    rest_or_cancel_remainder(order)
    publish_real_and_implied_market_data()
```

Everything else in this section explains why each line of that pseudocode is the way it is.

Recalculation triggers and fan-out. Every order event on any instrument in a related group (new order, cancel, amendment, partial fill) can change implied quotes in every *other* book of the group. One cancelled spread order can move implied prices in two outright books at once, each change generating market data. Implied-enabled products therefore have a structurally higher ratio of market-data messages to order messages, which must be budgeted in the publishing path, not discovered in production.

Atomicity. All legs of an implied match commit together or not at all. The single-threaded-per-book design from *The Matching Engine* chapter now shows its limits: an implied match spans several books. Either the whole related group of instruments is assigned to one sequencing thread (the common production choice: "partition by symbol" becomes "partition by *related instrument group*"), or a cross-book commit protocol is required, with all the latency and complexity that implies.

Double-execution prevention. Trader A's January order participates in the January book *and*, simultaneously, in the implied February ask. If a January buyer and a February buyer arrive in adjacent events, both paths claim A's lots, and only one may win. Serialised processing within the instrument group resolves this naturally: whichever event is sequenced first consumes the lots, and recomputation (Level 6) shrinks or removes the implied quote before the second event is processed. Any design that evaluates implied quotes against a stale copy of the contributing books reintroduces the race.

Determinism. Implied recomputation must itself be deterministic: given the same event sequence, the same implied quotes must appear, in the same order, with the same rounding. Iterating over candidate combinations in hash-map order, or letting floating-point arithmetic creep into spread prices, breaks the replay guarantees of the *Determinism, Replay, and Persistence* chapter.

Testing invariants. Implied logic is an ideal target for property-based testing, because its correctness conditions are crisp global invariants rather than example-shaped assertions:

1. *Conservation:* every implied execution's fills sum to zero in every instrument and in cash (Level 3), and no real order's quantity is ever consumed twice or driven negative.
2. *Firmness:* every published implied quote is executable at that instant: each ingredient exists with

sufficient quantity, including shared legs (Level 6).
3. *Grid consistency:* no published implied price and no assigned leg price is off-grid for its book (Level 9).
4. *No cross-book crossing:* after every event, no combination within the enabled implication depth is crossed (Level 8).
5. *Removal:* cancelling any single ingredient removes or correctly shrinks every implied quote built on it, within the same event.
6. *Limits respected:* every party to every implied execution trades at or better than its own limit price.

A fuzzer that generates random order flow across three related books and asserts these six properties after every event will find more implied-matching bugs than any hand-written example suite.

> **Key idea:** Implied orders are not free liquidity. They are the engine expressing, in one book, commitments already resting in others, so that related markets stay consistent, empty months stay tradeable, and cross-book price gaps are closed by the exchange rather than harvested by the fastest participant. When an implied order matches, real orders in real books are consumed, exactly once, atomically. Everything difficult about implementing implied orders (shared-leg quantities, depth, priority, tick rounding, fan-out, double-execution races) follows from taking that atomic, exactly-once consumption seriously.

13.13 Real-World Implementations

CME Globex is the reference implementation for futures: implied IN and implied OUT per product, implied depth disseminated on MDP 3.0, and per-product documentation of which implied types are enabled. CME has extended implied functionality steadily for two decades; in 2004, for example, it announced that its revised implied functionality would "link contract liquidity in butterfly spread trades on GLOBEX with that in underlying markets" for Eurodollar futures, then among the most actively traded futures in the world [CME press release, 8 March 2004]. **Eurex T7** offers the same idea under the name *synthetic matching*: "orders of different simple and complex instruments are executed against each other", supported for futures spreads and inter-product spreads [Eurex, *Matching principles*]. **ICE Futures** runs implied pricing across its energy futures and their spreads, including implied prices derived from other implied prices, the subject of the next chapter [ICE, *ICE Futures Implied Prices* FAQ]. The vocabulary differs from venue to venue; the three-book mechanics of this chapter are the common core.

13.14 Self-Check Exercises

Answers follow; attempt each before reading on. Convention throughout: $S = Near - Far$, and all prices are per barrel with a \$0.01 tick.

E1. January: bid \$74.20, ask \$74.60. February: bid \$76.10, ask \$76.55. The spread book is empty. Compute the implied Jan/Feb spread bid and ask.

E2. The only two orders in the market are a February bid at \$76.40 × 12 lots and a Jan/Feb spread ask at −\$1.80 × 20 lots. What implied quote do they create, at what price and quantity?

E3. In Level 3's original books, Trader C instead submits an order to buy 40 February at \$77.00. Describe the outcome, including every book afterwards.

E4. January ask \$75.00 × 10; two spread bids at −\$2.00 of 8 and 6 lots. What implied February quantity is published at \$77.00, and why is $8 + 6 = 14$ the wrong answer?

E5. You bought 3 Jan/Feb spreads at +\$0.40. The spread now trades at +\$0.15. What is your profit or loss, in dollars? (1 lot = 1,000 barrels.)

E6. A colleague's implementation decrements implied quantities on fills in the implied book instead of recomputing them, "for performance". Give a concrete sequence of events in which this publishes a quote the engine cannot honour.

E7. The books hold: January ask \$75.10 × 10 and a spread bid of −\$2.00 × 10, plus a real February bid of \$77.05 × 10 that arrived last. What happens when the February bid arrives, and at what prices?

Solutions

E1. Implied S_bid = Jan_bid − Feb_ask = 74.20 − 76.55 = **−$2.35**. Implied S_ask = Jan_ask − Feb_bid = 74.60 − 76.10 = **−$1.50**. Sanity check: the implied spread market is 0.85 wide, which is the January bid-ask width (0.40) plus the February width (0.45). Trading the spread through the outrights means paying both outright spreads at once.

E2. This is a trap, and the trap is the point: **these two orders imply nothing.** Add the vectors. The February bid is (Feb +1). The spread ask is a committed spread *seller*: (Jan −1, Feb +1). Their sum is (Jan −1, Feb +2), which cancels nothing and is not the shape of any single order. In words: the spread seller needs to *buy* February to complete its legs, and the only February order present is another *buyer*. Cross-check against the Level 2 table: Feb_bid pairs with S_**bid** (implied Jan bid), and S_ask pairs with Feb_**ask** (implied Jan ask) or Jan_**bid** (implied Feb bid); "Feb_bid + S_ask" appears in no row. If you mechanically computed 76.40 + (−1.80) = $74.60 and published it as an implied January quote, you manufactured a price with nothing executable behind it, precisely the class of bug that testing invariant 2 exists to catch.

E3. The implied ask at $77.00 is 30 lots (min(50, 30)). C fills 30 lots through the three-fill construction: A sells 30 January at $75.00, B's spread fills 30 at −$2.00 (B's order is now complete), and C buys 30 February at $77.00. With B's order gone, the implied February ask recomputes to nothing, so C's remaining 10 lots rest as a real February bid at $77.00. Afterwards: January ask $75.00 × 20 (A); spread book empty of real orders; February bid $77.00 × 10 (C). And one more thing happens, which is easy to miss: C's new February bid is itself an ingredient. Together with A's January ask it implies a spread **ask** at Jan_ask − Feb_bid = 75.00 − 77.00 = **−$2.00**, quantity min(20, 10) = **10**. Every new real order is a candidate ingredient in every related book.

E4. Published quantity = min(10, 8 + 6) = **10**. The January ask is a shared leg: both spread bids route their February selling through the same 10 January lots, so together they can deliver at most 10. Pairwise-min-then-sum (min(10, 8) + min(10, 6) = 14) counts the shared leg twice and over-advertises by 4 lots that the engine could not deliver if a buyer took them.

E5. The spread price fell from +0.40 to +0.15, a change of −0.25 per barrel. You are long the spread, so you lose: −0.25 × 1,000 barrels × 3 lots = **−$750**. (Check with legs: suppose you bought January at $76.40 and sold February at $76.00; later January is $76.00 and February $75.85. January: −0.40; February short: +0.15; net −0.25 per barrel.)

E6. Books: January ask $75.00 × 20 (A); spread bid −$2.00 × 20 (B), so implied February ask $77.00 × 20. Event 1: an ordinary January buyer takes 15 of A's lots. A has 5 left, so the true implied quantity is min(5, 20) = 5. The decrement-only implementation saw no fill *in the February implied book* and still advertises 20. Event 2: a February buyer sends an order to buy 12 at $77.00. The engine can construct only 5 lots: it must either reject a fill against a quote it published, fill against a phantom, or (worst of all) begin an atomic commit that cannot complete. Recomputation on every event in the group is not an optimisation opportunity; it is the correctness mechanism.

E7. Before the February bid arrives, the engine publishes an implied February ask of 75.10 − (−2.00) = $77.10 × 10. The incoming bid of $77.05 is below $77.10, so it does **not** cross: nothing trades, and the bid rests. Now look for implied quotes involving the new order: January ask plus February bid imply a spread ask of 75.10 − 77.05 = **−$1.95**, and B's resting spread bid is −$2.00, which does not reach −1.95 either. The books are consistent and nothing trades. (Had the February bid been $77.10 or higher, it would have traded 10 lots at $77.10 through the implied construction. The price is always the *resting* side's price, and here the resting side is the implied ask.)

CHAPTER 14

Multi-Layer Implied Orders and the Combinatorial Wall

The previous chapter combined exactly two real orders at a time. A January ask plus a spread bid implied a February ask; a January bid plus a February ask implied a spread bid. A curious developer immediately asks the next question: *if an implied February ask can be traded against, can it also be used as an ingredient?* Combine it with a Feb/Mar spread and you get an implied March ask. Combine that with a Mar/Apr spread and you get an implied April ask. Why stop?

This chapter answers that question. It extends the vector model of the previous chapter to chains of any length, shows a real multi-generation example published by ICE, re-derives the "no liquidity from nothing" promise for chains, and then walks into what this chapter calls **the combinatorial wall**: the reason every exchange that offers implied matching restricts it, whether by the number of steps, by the set of instruments, by the depth it publishes, or by the time it is switched on.

The short version, for readers in a hurry: finding the *single best* implied price through arbitrarily long chains of simple calendar spreads is a well-behaved shortest-path problem. Everything else an exchange must do with implied liquidity (allocate quantity, publish depth, break ties fairly, execute atomically, assign leg prices, support three- and four-legged strategies, and do all of it within microseconds after *every single event*) grows explosively with the number of steps allowed. The explosion, not the arithmetic, is what forces the limit.

14.1 Generations: Counting the Links in the Chain

We need a precise way to say how "deep" an implied quote is. This chapter uses the convention implied by the ICE example below:

- **Generation 0**: a real order, resting in its own book.
- **Generation 1**: an implied quote built from **two** real orders (everything in the previous chapter).
- **Generation 2**: an implied quote that uses a generation-1 implied quote as an ingredient, so it is built from **three** real orders.
- **Generation k**: built from $k + 1$ real orders.

Venues word this differently. CME's matching-priority documentation, for instance, labels outright orders "generation 0" and spread orders "1st generation" when ordering sources for allocation [CME Group Client Systems Wiki, *Futures Implied Order Matching Priority*]. What matters is the underlying quantity: *how many real orders must execute together for one implied lot to trade.* That number drives every cost in this chapter.

14.2 A Real Multi-Generation Example (ICE)

ICE Futures publishes the following example of implied prices across generations in a crude-oil-style strip of September, October and November contracts [ICE, *ICE Futures Implied Prices* FAQ]. The real orders are:

Book	Side	Price
September	Bid	75.90
October	Ask	76.83
Oct/Nov spread	Bid	−0.52

(As in the previous chapter, spread price = near month − far month, and a spread bid buys the near month and sells the far month.)

ICE derives, and we can check with the vector rule:

Generation 1, implied IN: a Sep/Oct spread bid at −0.93.

```
  Buy Sep at 75.90          v = (Sep +1)             c = +75.90
+ Sell Oct at 76.83         v = (Oct −1)             c = −76.83
  ------------------------------------------------------------------
                            v = (Sep +1, Oct −1)     c =  −0.93   → Sep/Oct spread BID at −0.93
```

Generation 1, implied OUT: a November ask at 77.35.

```
  Sell Oct at 76.83         v = (Oct −1)             c = −76.83
+ Buy Oct/Nov at −0.52      v = (Oct +1, Nov −1)     c =  −0.52
  ------------------------------------------------------------------
                            v = (Nov −1)             c = −77.35   → November ASK at 77.35
```

Generation 2: a Sep/Nov spread bid at −1.45, built from the real September bid and the *implied* November ask:

```
  Buy Sep at 75.90          v = (Sep +1)             c = +75.90
+ implied Sell Nov at 77.35 v = (Nov −1)             c = −77.35
  ------------------------------------------------------------------
                            v = (Sep +1, Nov −1)     c =  −1.45   → Sep/Nov spread BID at −1.45
```

Unpack the implied November ask and the generation-2 quote is simply the sum of all **three** real orders: $75.90 - 76.83 - 0.52 = -1.45$, with the October legs cancelling. The same number appears if you chain the other generation-1 quote instead: implied Sep/Oct bid (-0.93) plus real Oct/Nov bid (-0.52) = -1.45. That is not a coincidence. **An implied quote is a property of the set of real orders it uses, not of the order in which you combine them.** The "generation" is just a count of how many real orders are in the set, minus one.

Executing a generation-2 trade

Give the three real orders some sizes and owners:

Book	Side	Price	Lots	Who
September	Bid	75.90	10	Trader P
October	Ask	76.83	8	Trader Q
Oct/Nov spread	Bid	−0.52	5	Trader R

The implied Sep/Nov spread bid is -1.45 for $\min(10, 8, 5) = $ **5 lots**, bottlenecked by R, the smallest ingredient. Trader X now submits an order to **sell** 5 Sep/Nov spreads at -1.45 (sell September, buy November). The engine must commit **four** participants' fills atomically:

Participant	Does	Position change	Price(s)
X (incoming)	sells 5 Sep/Nov spreads at −1.45	Sep −5, Nov +5	Sep 75.90, Nov 77.35
P	buys 5 September	Sep +5	75.90
Q	sells 5 October	Oct −5	76.83

Participant	Does	Position change	Price(s)
R	buys 5 Oct/Nov spreads at −0.52	Oct +5, Nov −5	Oct 76.83, Nov 77.35

Check the columns. **Positions** sum to zero in every month: September $-5 + 5 = 0$; October $-5 + 5 = 0$; November $+5 - 5 = 0$. **Leg prices** are consistent with every order: X's legs differ by $75.90 - 77.35 = -1.45$; R's legs differ by $76.83 - 77.35 = -0.52$; P and Q trade exactly at their limits. **Cash**, using the "cash paid" convention of the previous chapter, per barrel: X pays $5 \times 1.45 = 7.25$ (selling a spread at −1.45 means paying 1.45 per unit), P pays $5 \times 75.90 = 379.50$, Q pays $-5 \times 76.83 = -384.15$, and R pays $5 \times (-0.52) = -2.60$. Sum: $7.25 + 379.50 - 384.15 - 2.60 = \mathbf{0}$.

As before, X could have done all of this by hand: sell September to P, buy October from Q, and sell the Oct/Nov spread to R. Three orders, three chances for the market to move in between, and one very unhappy trader if the third fill fails. Implied matching removes that leg risk. It does not add one lot of liquidity: the 5 lots X traded are 5 of P's lots, 5 of Q's lots and 5 of R's spreads, and every one of those is now gone from every book where it used to be visible. **Multi-layer implication, like single-layer implication, only adds routes.**

When leg prices must be manufactured

In the ICE example, every month in the chain had a real outright order, so every leg price was pinned down by a real limit price. That is not always the case. Take two spread orders and nothing else:

Book	Side	Price	Lots
Jan/Feb spread	Bid	−2.00	30
Feb/Mar spread	Bid	−1.50	20

The vectors (Jan +1, Feb −1) and (Feb +1, Mar −1) add up to (Jan +1, Mar −1): an implied **Jan/Mar spread bid** at $(-2.00) + (-1.50) = \mathbf{-3.50}$, for $\min(30, 20) = 20$ lots. Only two real orders are involved, so it is a generation-1 quote, but of a third kind that is neither CME's "implied IN" (spread from outrights) nor "implied OUT" (outright from spread): a spread implied from two other spreads.

When a Jan/Mar spread seller hits it for 20 lots, three participants take **six** leg positions (the seller: Jan −20, Mar +20; the Jan/Feb bidder: Jan +20, Feb −20; the Feb/Mar bidder: Feb +20, Mar −20). The two February legs trade *between the two spread bidders*, and nothing in any order says what the February price is. Only differences are fixed: Jan − Feb = −2.00 and Feb − Mar = −1.50. The engine must manufacture the actual prices by rule, typically anchoring one month to a reference price (its last trade or settlement, as in the previous chapter's leg-price discussion). With February anchored at 77.00, the legs become January 75.00, February 77.00, March 78.50, and every spread's legs differ by exactly its price. Every leg price must also be a valid tick and inside that month's price limits, and the anchor rule must be deterministic. The longer the chain, the more legs have no real price of their own and must be manufactured this way.

14.3 The Graph View: Implied Pricing Is a Path Problem

To reason about chains of any length, the vector model becomes a graph.

- Draw one **node** per delivery month, plus one special node **$** that stands for "cash".
- Every resting order becomes a **directed edge** u → v, meaning "this order is willing to go long u and short v". A buy of January is the edge Jan → $; a sell of January is $ → Jan; a Jan/Feb spread bid is Jan → Feb; a Jan/Feb spread ask is Feb → Jan.
- Give each edge a **weight** equal to the order's cash amount c from the previous chapter: the most it is willing to pay per unit. A January bid at 74.40 has weight +74.40; a January ask at 75.00 has weight −75.00; a spread bid at −2.00 has weight −2.00; a spread ask at −1.80 has weight +1.80.

Now the whole theory of implied pricing fits in one sentence:

A path of resting orders from node x to node y is an implied order to go long x and short y, and the most it will pay is the sum of the weights along the path.

The intermediate nodes cancel, exactly as the October legs cancelled above. Some checks against examples you already know:

- Path \$ → Jan → Feb (Jan ask 75.00, weight −75.00; spread bid −2.00, weight −2.00): long nothing, short February, paying −77.00, that is, *receiving* 77.00. An implied **February ask at 77.00** (previous chapter, Level 3).
- Path Jan → \$ → Feb (Jan bid 74.40, weight +74.40; Feb ask 76.60, weight −76.60): long January, short February, paying −2.20. An implied **spread bid at −2.20** (previous chapter, Level 4).
- Path Sep → \$ → Oct → Nov (Sep bid +75.90, Oct ask −76.83, Oct/Nov bid −0.52): long Sep, short Nov, paying −1.45. The ICE generation-2 **Sep/Nov bid at −1.45** .

The generation of an implied quote is the number of edges in its path, minus one.

The best implied bid for "long x, short y" is therefore the maximum-weight path from x to y. Equivalently, negate every weight and it becomes a *shortest-path* problem, one of the most thoroughly studied problems in computer science.

Cycles are crossed markets. A cycle of resting orders (a path from a node back to itself) has a position vector of zero: its orders could trade with each other and nobody else. They *can* do so exactly when the total they are willing to pay is at least zero, since the cash must balance. A January bid at 75.00 (weight +75.00) and a January ask at 74.99 (weight −74.99) form a cycle of weight +0.01: a crossed book. Because a matching engine trades every crossing it can see the moment it arises (the previous chapter's Level 8), a correctly running engine never has a cycle of non-negative weight at rest *within the scope of implication it has enabled*. (Cycles longer than the enabled depth can stay crossed; closing them is left to arbitrageurs.) In shortest-path terms, **an uncrossed family of books is a graph with no negative cycles** (after negating the weights), which is precisely the condition under which classical algorithms such as Bellman–Ford find optimal paths in polynomial time.

This is a genuinely useful result, and it matches what at least one exchange advertises: ICE states that its engine "will determine the best bid and offer price in each market regardless of the number of generations required", within a given strip type [ICE, *ICE Futures Implied Prices* FAQ]. For the *best price* in a world of simple two-legged calendar spreads, the number of generations is not the problem.

So where is the wall?

14.4 The Combinatorial Wall

The best price is one number per book side. An exchange needs a great deal more than that, and each additional requirement is where the growth hides.

1. The number of routes grows factorially

Consider a model strip of $\mathbf{M = 12}$ consecutive monthly contracts, with every outright and every calendar spread between any two months listed: 12 outright books plus $12 \times 11 / 2 = 66$ spread books. (Real venues list fewer spreads, but many list a lot of them.) In the graph that is 13 nodes (12 months plus \$), every pair joined by orders on both sides.

How many distinct *routes* (simple paths) lead from \$ to, say, the December node, i.e. how many different constructions could produce an implied December ask? A route with L real orders visits L − 1 intermediate months chosen in order from the 11 others, so there are 11! / (12 − L)! routes of length L:

Generation (real orders − 1)	Routes of exactly this length	Cumulative routes up to this generation
0 (the real December book)	1	1
1	11	12
2	110	122

Generation (real orders − 1)	Routes of exactly this length	Cumulative routes up to this generation
3	990	1,112
4	7,920	9,032
5	55,440	64,472
11 (no limit)	39,916,800	108,505,112

That is for **one** side of **one** of 78 books, with a single price level per book. Every additional resting price level multiplies the possibilities further. A best-price search avoids enumerating these routes, but several of the exchange's other obligations do not.

2. Fair allocation needs the ties, not just the best

When several constructions offer the same implied price, the engine must decide which real orders trade, in a documented and deterministic order. CME sorts competing implied sources by strategy type, product code, leg expirations, number of legs, and instrument identifiers [CME, *Futures Implied Order Matching Priority*]. A shortest-path algorithm returns *a* best path. Fair allocation needs *all* the best paths, ranked, and the number of equal-price paths can be as large as the route counts above.

3. Quantity and depth turn a path problem into a flow problem

The previous chapter showed that quantity is a bottleneck (the smallest ingredient) and that shared orders cap the total. Across many routes that share many orders, "how much can trade at this price, and at the next price, and the next?" is a **minimum-cost flow** problem: route as many lots as possible through the graph, cheapest routes first, never exceeding any order's size. That is still polynomial in theory, but it is far heavier than one shortest-path search, and it must be redone after every event because every fill and cancel changes the capacities. The previous chapter's warning about naive depth (a fictitious $77.02 level) applies with far more force when routes can be several orders long and share orders in the middle.

4. Every event can change every book

With implication depth unlimited, a single cancel of a single order in the middle of the strip can change the best implied price in *every* book whose best route passed through it: potentially all 78 books, both sides. Each change is a market-data message. The previous chapter's "fan-out" becomes a flood, and it hits hardest precisely when markets are busiest.

5. Atomic execution grows with the chain

A generation-k trade commits $k + 1$ real orders plus the incoming one, in $k + 1$ or more books, as one indivisible unit. Every leg must pass validation (price limits, tick grid, risk checks, self-match prevention) before anything commits; every internal leg with no real outright price must have its price manufactured; and every participant must receive a correct execution report. The longer the chain, the larger the unit that can fail, and one failed leg voids the whole trade.

6. Multi-legged strategies leave the world of paths

Everything so far assumed two-legged instruments: edges connect exactly two nodes. Real venues also list **butterflies** (buy 1 near, sell 2 middle, buy 1 far: CME launched implied IN *and* implied OUT functionality for Eurodollar butterflies in 2004, describing the new implied OUT functionality as "an industry first", and covered only the first three years of the Eurodollar curve [CME press release, 18 October 2004]), **condors**, **packs and bundles** (a strip of four or more consecutive contracts traded as one), and **ratio inter-commodity spreads** (for example, CME Treasury spreads in which legs trade in ratios such as 3:2, where implied prices can even fall on a grid finer than the direct spread's ticks [Carruthers, 2018]).

A butterfly's vector has entries $(+1, -2, +1)$. It is not an edge between two nodes; it touches three at once, with a weight of 2 in the middle. Finding a set of resting orders whose vectors, with integer multiplicities, sum

exactly to a target vector is no longer a path problem. In general it is an **integer programming** problem, a class for which no efficient general algorithm is known, and whose best-known methods can take exponential time in the worst case. An exchange cannot afford "exponential in the worst case" inside a matching loop that must answer in microseconds, deterministically, after every event.

7. The benefit shrinks while the cost explodes

Remember the promise: implication creates no liquidity. Each extra generation adds only more *routes* to the same real orders. A generation-5 route is bottlenecked by the smallest of six real orders and is usually beaten on price by a shorter one, because every extra link adds another bid-ask spread to cross (the previous chapter's Exercise E1 showed a single implied spread already costs both outright spreads). The marginal value of the sixth link is small; its marginal cost, in latency, fan-out, and failure modes, is large.

14.5 How Real Venues Draw the Line

Every venue that offers implied matching restricts it along one or more of four axes. The examples below are all from the venues' own documentation.

Limit the number of generations. CME's implied options functionality is limited to generation 1: "all implied options orders must be built directly from customer orders; implied orders cannot be used to create another implied order (second generation implied)" [CME Group Client Systems Wiki, *Implied Options*].

Limit the set of instruments. CME's 2004 implied butterfly functionality covered only the first three years of the Eurodollar curve [CME, 18 October 2004]. ICE implies prices "regardless of the number of generations", but only within a given strip type (month, quarter, season or calendar) [ICE, *ICE Futures Implied Prices* FAQ]. Restricting the instrument set shrinks M, and the route counts in the table above shrink factorially with it.

Limit what is published. CME's MDP 3.0 feed carries only a "2-deep best bid and ask" for each implied-eligible futures contract [CME, *MDP 3.0 – Implied Book*]. Two levels of implied depth are cheap to compute and difficult to get wrong; forty would be neither.

Limit when it runs. CME implied options are "available *on demand* for a configurable time interval": implication for an options instrument is switched on by a request for quote and switched off again when the timer expires, with "no extension to the specified time under any circumstances" [CME, *Implied Options*]. The engine pays the cost of implication only when a participant has asked for a price, and only briefly.

Two further practical restrictions are common: implied prices must usually beat the direct market by at least a whole tick before implied liquidity is accessed (Carruthers documents this for CME Treasury inter-commodity spreads), and all instruments in the group must be in an open trading state for implication to run at all (CME, *Implied Options*).

14.6 What This Means When You Build One

If you are designing an implied-capable engine, the lesson is not "implement Bellman–Ford and you are done". It is:

1. **Make the implication depth an explicit, per-product configuration value**, defaulting to 1, and give every increase a measured latency budget. Depth is a product decision with an engineering price tag, not a free feature.
2. **Bound the instrument graph.** Decide which spreads participate in implication and keep the group small enough to live on one sequencing thread, so that atomicity comes from serialisation rather than from a distributed commit protocol.
3. **Compute what you publish, and publish little.** Top-of-book or two levels of implied depth, recomputed after every event, beats a deep ladder that is expensive, path-dependent, and occasionally fictitious.
4. **Keep two-legged and multi-legged implication separate.** Paths are tractable; hyperedges are not. Support butterflies and packs only through explicitly enumerated, fixed construction templates, never through a general search.

5. **Reuse the invariants from the previous chapter unchanged.** Conservation, firmness, grid consistency, no cross-book crossing, removal, and limits respected are all statements about *sets of real orders*, so they hold for generation 5 exactly as for generation 1. A property-based test that checks them after every event is the only practical way to trust a multi-generation engine.

   Key idea: Chains of implied orders are sums of real orders, just like single implied orders, so they create no liquidity, only more routes to it. Finding the single best price through chains of simple calendar spreads is a tractable shortest-path problem. But fair allocation among equal prices, quantity and depth, market-data fan-out, atomic execution, manufactured leg prices, and multi-legged strategies all grow explosively with the number of links allowed. That is why every venue bounds implication, by generation, by instrument set, by published depth, or by time, and why a well-designed engine makes that bound an explicit, measured configuration choice.

14.7 Self-Check Exercises

Convention: spread price = near − far; all spreads are calendar spreads of one lot per leg.

E1. Real orders: March bid 80.10 × 6; March/April spread ask −0.40 × 9; April/May spread ask −0.35 × 4. What is the best implied May quote, which side is it on, what generation is it, and what quantity?

E2. Using the edge-weight rules of the graph view, write down the weights of: a June ask at 81.00; a June/July spread bid at −0.30; a July bid at 81.25. Is there a cycle among these three orders, and if so, is the family of books crossed?

E3. In the ICE example, suppose Trader R's Oct/Nov bid is cancelled. Which of the three implied quotes survive?

E4. Using the route table, how many routes of generation 2 or less lead from $ to one month in a fully connected strip of M = 12 months? If an exchange restricts implication to adjacent-month spreads only (Jan/Feb, Feb/Mar, … , Nov/Dec), how many routes lead from $ to December, and why?

Solutions

E1. Chain the orders so that the unwanted legs cancel. The March bid is (Mar +1), weight +80.10. A March/April spread *ask* is a spread seller, (Mar −1, Apr +1), weight +0.40. An April/May spread ask is (Apr −1, May +1), weight +0.35. Sum: (May +1), weight 80.10 + 0.40 + 0.35 = **80.85**. The combination is willing to go long May paying up to 80.85: an implied **May bid at 80.85**. It uses three real orders, so it is **generation 2**, and its quantity is min(6, 9, 4) = **4** lots. (Check in words: someone selling May to this construction delivers May to the Apr/May spread seller, who buys May and sells April 0.35 lower; April goes to the Mar/Apr spread seller, who buys April and sells March 0.40 lower; March goes to the March bidder at 80.10. 80.10 + 0.40 + 0.35 = 80.85.)

E2. June ask: edge $ → Jun, weight −81.00. June/July spread bid: edge Jun → Jul, weight −0.30. July bid: edge Jul → $, weight +81.25. They form the cycle $ → Jun → Jul → $, with total weight −81.00 − 0.30 + 81.25 = **−0.05**. The cycle's weight is negative, so the books are **not** crossed. (In prices: the implied July ask from the first two orders is 81.00 + 0.30 = 81.30, which is above the July bid of 81.25: a five-cent-wide, uncrossed market.)

E3. The generation-1 Sep/Oct bid at −0.93 survives, because it uses only the September bid and the October ask. The implied November ask at 77.35 disappears, because R's order was one of its two ingredients. The generation-2 Sep/Nov bid at −1.45 disappears too, because it used R's order (through the implied November ask). This is the removal test from the previous chapter, applied to a chain: remove any real order, and every implied quote whose set of real orders contains it must go.

E4. From the table: 1 + 11 + 110 = **122** routes. With adjacent-month spreads only, the graph from $ is: $ to every month (the outright books), plus a line Jan – Feb – … – Dec. A simple path from $ to December leaves $ once, enters the line at some month, and walks along the line to December without revisiting a node. It can enter at any of the 12 months, so there are exactly **12** routes: one direct (generation 0) and one for each

possible entry month from January to November. Restricting which spreads participate turned a factorial explosion into a straight line, which is exactly why venues restrict the instrument set.

CHAPTER 15

Indexes, The Market's Single Number

Every concept so far has been about a single instrument: one order, one book, one matching engine, one symbol's price. But participants, journalists, regulators, and pension trustees all ask a broader question every day: *how did the market do?* Answering that question with one number is the job of a **market index**.

> **Key idea:** An index is a single number computed from a basket of instruments that summarises the collective movement of a market or a slice of it. It is not itself tradable, it is a *measurement*, but vast sums of real money are invested, settled, and risk-managed against that measurement. Getting it right matters far more than its "just a number" appearance suggests.

15.1 What an Index Is, and Why Exchanges Maintain Them

An index reduces hundreds or thousands of individual prices to one figure that rises and falls with the market it tracks. The S&P 500 summarises 500 large US companies; the FTSE 100 tracks the largest 100 on the London Stock Exchange; the DAX follows 40 major German firms; the Nikkei 225 covers Japan. When the news says "the market was up 1.2% today," it is quoting an index.

Exchanges and specialist index providers maintain indexes for several reasons:

- **A benchmark.** Investors measure their own performance against an index. A fund that returned 8% when its benchmark returned 12% underperformed, even though it made money.
- **A tradable underlying.** You cannot trade the index directly, but you can trade *derivatives* of it: index futures, index options, and exchange-traded funds (ETFs) that hold the basket. These are some of the most heavily traded instruments in the world.
- **A risk-control reference.** As we will see, market-wide safety mechanisms such as circuit breakers are triggered by index moves, not by individual stocks.
- **An economic signal.** Central banks, governments, and businesses watch indexes as a barometer of economic confidence.

A single exchange or index provider typically maintains *many* indexes at once: a broad market index, sector sub-indexes (technology, financials, energy), size-segmented indexes (large-cap, mid-cap, small-cap), and themed indexes. Each is just a different basket and a different set of rules applied to the same underlying prices.

15.2 How Indexes Are Weighted

The central design question for any index is: *when AAPL moves 1% and a small company moves 1%, should they affect the index equally?* The answer is the index's **weighting methodology**, and there are three common approaches.

Price-weighted. Each constituent contributes in proportion to its *share price*. The Dow Jones Industrial Average works this way: a $400 stock moves the index four times as much as a $100 stock, regardless of company size. This is a historical accident from the era when adding up prices by hand was the only practical method, and it produces distortions, a stock's influence depends on the arbitrary number of shares it has

issued, not on its economic importance.

Equal-weighted. Every constituent gets the same weight regardless of size or price. Simple, but it requires constant rebalancing as prices drift apart, and it gives tiny companies the same influence as giants.

Market-capitalisation-weighted. Each constituent contributes in proportion to its **market capitalisation**, the total value of its shares. This is how the S&P 500, FTSE 100, DAX, and almost every modern serious index works, and it is the method worth understanding in depth.

Market Capitalisation and Normalisation

A company's **market capitalisation** ("market cap") is simply:

```
market cap = share price × shares outstanding
```

A company trading at $200 with one billion shares is worth $200 billion. In a cap-weighted index, that company's influence is its market cap as a fraction of the *total* market cap of all constituents. A $400 billion company in an index whose constituents total $4 trillion drives 10% of the index's movement; a $40 billion company drives 1%. Large companies dominate, which is usually what we want, the index reflects where the market's money actually sits.

But there is a problem. If you just add up the market caps of 500 companies, you get a number in the tens of trillions, an unwieldy figure that means nothing to a human and that jumps every time a company issues shares or is swapped out of the index. The solution is **normalisation against a divisor**.

> **Key idea:** An index level is the aggregate market cap of its constituents divided by a **divisor**, a scaling number chosen so the index reads as a friendly figure (say, 1000 at launch) and, crucially, so the level stays continuous when the basket itself changes.

```
index level = aggregate market cap / divisor
```

At launch, the divisor is set so the index equals its chosen **base value**. If the constituents total $7 trillion and you want the index to start at 1000, you set the divisor to $7 trillion ÷ 1000 = $7 billion. From then on, as prices move, the aggregate cap moves, and dividing by the (mostly fixed) divisor produces the published level.

A worked example. Suppose a small index has three constituents:

Company	Price	Shares outstanding	Market cap
Alpha	$209.50	15,000,000,000	$3,142,500,000,000
Beta	$415.00	7,400,000,000	$3,071,000,000,000
Gamma	$248.00	3,200,000,000	$793,600,000,000
Total			**$7,007,100,000,000**

To launch at a base value of 1000:

```
divisor = 7,007,100,000,000 / 1000 = 7,007,100,000
index level = 7,007,100,000,000 / 7,007,100,000 = 1000.00
```

Now if Alpha rises from $209.50 to $230 (up 9.8%), its market cap grows by $20.50 × 15 billion = $307.5 billion, the aggregate cap rises to $7,314,600,000,000, and the index reads 7,314,600,000,000 / 7,007,100,000 1043.9, up about 4.4%. Notice that Alpha, the largest constituent, moved the index more than an identical percentage move in tiny Gamma would have. That is cap weighting doing its job.

Most serious indexes refine this further with **free-float adjustment**: instead of *all* shares outstanding, they count only the shares actually available to public trading, excluding blocks held by founders, governments, or strategic owners that never trade. A company that is 60% owned by its founding family has only 40% of its shares influencing supply and demand, so only that 40% should influence the index. The principle is identical; only the share count used in the market-cap formula changes.

15.3 Corporate Actions and the Divisor

Here is where index maintenance becomes genuinely delicate. Companies do things to their share structure that change the raw numbers without changing their economic value, and a naive index would lurch wildly every time.

Stock splits. A company does a 2-for-1 split: every share becomes two, and the price halves. A $200 stock becomes a $100 stock with twice as many shares. The market cap is *unchanged*, the company is worth exactly what it was a second earlier. But if the index simply plugged in the new $100 price without noticing the doubled share count, it would appear to halve that constituent's contribution, and the index would drop for no real reason.

Dividends. When a company pays a cash dividend, its share price typically drops by roughly the dividend amount on the **ex-dividend date** (the buyer no longer receives that payout). Again, no value was destroyed, it was handed to shareholders, but the price, and therefore the raw market cap, falls.

Share issuance, buybacks, mergers, and spin-offs. Each of these changes the share count or the set of constituents. A company issuing new shares raises its market cap without its price moving. A constituent being acquired must be removed entirely.

> **Key idea:** Whenever a corporate action changes the *aggregate market cap* of the basket for a reason unrelated to genuine price movement, the divisor is adjusted to absorb the change, so the index level does not jump at that instant. Only *future* price movements then change the level.

The mechanism is a simple ratio. If a corporate action changes the aggregate cap from an old value to a new value, the divisor is rescaled by the same ratio:

```
new divisor = old divisor × (new aggregate cap / old aggregate cap)
```

Because the level is `aggregate cap ÷ divisor`, multiplying both the cap *and* the divisor by the same factor leaves the level unchanged at the moment of adjustment. The index glides through the event seamlessly. This is precisely the technique the S&P 500 has used for decades, its famous divisor is a closely tracked number that evolves continuously as constituents split, pay dividends, get added, and get removed [S&P Dow Jones Indices, *Index Mathematics Methodology*].

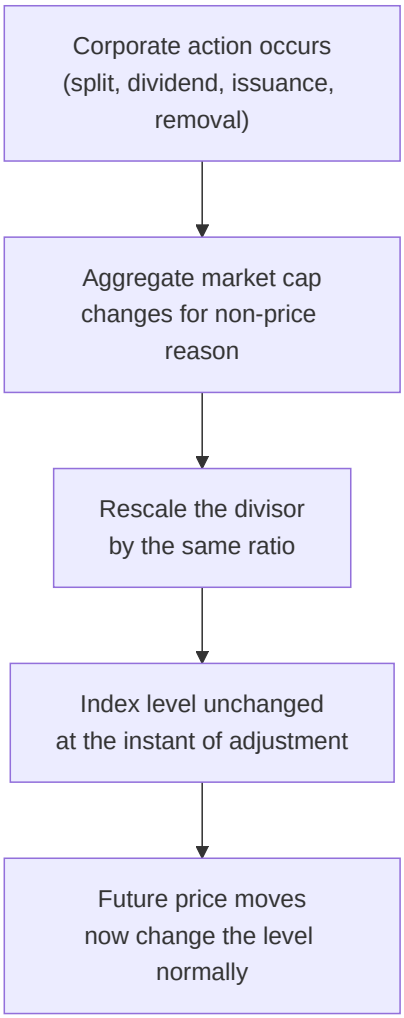


Figure 1: *The divisor is adjusted to absorb changes in aggregate market cap that are not genuine price moves, so the index level remains continuous.*

The continuity this provides is not a nicety, it is the whole point. An index is only meaningful if a move from 4000 to 4040 always means "the market rose 1%," never "a constituent happened to split last night."

15.4 Getting Into the Index: The S&P 500 Example

Membership in a major index is coveted, and the criteria are strict. The S&P 500 is instructive because it is *not* purely mechanical, a committee makes the final call, which surprises many people who assume it is simply "the 500 biggest US companies."

To be eligible for the S&P 500, a company must broadly satisfy rules such as [S&P Dow Jones Indices, *U.S. Indices Methodology*]:

- **Domicile.** It must be a US company.
- **Market capitalisation.** It must exceed a minimum threshold that is raised periodically to keep pace with the market (in the billions of dollars, and revised upward over the years).
- **Liquidity.** Its shares must trade actively enough that index funds can buy and sell without distorting the price; there are minimum requirements on float-adjusted value traded.
- **Public float.** A sufficient proportion of its shares must be publicly available for trading, not locked up by insiders.
- **Profitability.** The sum of its most recent four quarters of earnings must be positive, *and* the most recent quarter must be positive. This single rule has kept some very large, fast-growing-but-unprofitable companies out for years.
- **Committee approval.** Even meeting every numeric rule does not guarantee entry. An index committee

selects constituents to keep the index representative of the US large-cap economy.

This contrasts with rules-based indexes like the Russell or FTSE families, which reconstitute largely mechanically, on a published schedule, according to published rules. Both philosophies exist; the trade-off is between predictability (rules) and representativeness (committee judgement).

Indexes are **reconstituted** periodically, additions and removals are batched and announced in advance, so that the funds tracking them have time to adjust. And this advance announcement is where the drama begins.

15.5 The Consequences of Being Included

Being added to a major index is one of the most valuable non-operational events that can happen to a public company, because of a structural feature of modern markets: **passive investing**.

Trillions of dollars sit in **index funds** and **ETFs** whose mandate is to hold exactly the index basket, in exactly the index proportions. These funds are not allowed to exercise judgement, when a stock joins the S&P 500, every S&P 500 index fund is *contractually obliged* to buy it, in proportion to its index weight, regardless of price. Collectively that can mean tens of billions of dollars of forced, price-insensitive demand arriving on a known date.

> **Key idea:** Index inclusion converts a stock from something investors *may* buy into something a huge cohort of funds *must* buy. That forced demand, and the front-running of it by other traders, is the **index inclusion effect**.

The textbook illustration is Tesla's addition to the S&P 500 in December 2020. Because of its size, index funds needed to buy on the order of tens of billions of dollars of Tesla stock to mirror the index, and traders who anticipated that demand bought ahead of them. Tesla's share price rose dramatically between the announcement and the actual inclusion date, an enormous move driven substantially by the mechanics of index membership rather than by any change in the company's business that week.

The consequences of inclusion (and its mirror image, *removal*) include:

- **Forced buying or selling** by index funds on the reconstitution date.
- **Anticipatory trading** by others, often pushing the price before the funds even act.
- **A permanently broader shareholder base** and usually improved liquidity and visibility.
- **A lower cost of capital** for the company, since constant index-fund demand supports its share price.
- **Removal pain**, conversely, being dropped from an index forces mass selling and can depress a struggling company further.

This is also why the integrity of index *membership decisions* matters so much: a single inclusion can move billions of dollars, so the rules and the committee must be transparent and free from manipulation.

15.6 Why Correct Index Calculation Is Critical

It is tempting to treat an index as a cosmetic summary. It is anything but. Consider everything that depends, in legally and financially binding ways, on the published number:

- **Derivatives settle against it.** Index futures and index options pay out based on the index level at expiry. If the closing index print is wrong by a fraction of a percent, the settlement payments on billions of dollars of contracts are wrong.
- **ETFs are priced against it.** An ETF's fair value is derived from the basket; errors propagate directly into what investors pay and receive.
- **Performance is judged against it.** Fund manager bonuses, fund inflows, and investor decisions all hinge on the benchmark.
- **Risk controls fire on it.** Market-wide circuit breakers (below) are triggered by index moves.

Because so much rests on it, index calculation must be **deterministic, auditable, and reproducible**. The same inputs must always produce the same output; every divisor adjustment must be recorded so the history can be reconstructed and explained; and the **closing value**, "the print", must be computed with special care, because that is the value that settles derivatives and strikes fund valuations. A calculation error in an index is not a cosmetic blemish; it is a chain reaction that can mis-settle contracts, misprice funds, and trigger or

fail to trigger safety mechanisms.

This is exactly why, in real systems, index calculation is kept *separate* from the matching engine. The engine's one job is to match orders deterministically with minimal latency; index calculation involves aggregation, persistence, and historical record-keeping that have no business adding delay to order processing. The index is computed by a dedicated subscriber that *listens* to the trade feed and never interferes with matching.

15.7 What to Watch Out For

Indexes carry a number of subtle hazards that anyone building or relying on them should respect.

Stale prices. If a constituent has not traded recently, the index must use its last known price. In a fast-moving or thinly traded market, parts of the index can be "stale," reflecting prices that are minutes old. During stress, this can make the index lag reality.

The closing auction and "marking the close." Because the closing print settles so much money, the final minutes of trading attract enormous volume and, occasionally, manipulation attempts, traders pushing a constituent's price at the very close to influence the settlement. Surveillance pays special attention here.

Reconstitution turbulence. On the days indexes add and drop members, the forced flows described above create unusual volume and volatility. "Triple witching" days, when index futures, index options, and stock options all expire together, are notorious for this.

Reflexivity and concentration. When a huge fraction of investment is passive, the index can start to *drive* the market rather than merely *measure* it. Money flows into index funds, which mechanically buy the largest constituents, which raises their weight, which attracts more flows. The largest few names can come to dominate the index to a degree that worries regulators about concentration risk.

Survivorship and continuity bias. Because failing companies are removed and replaced by thriving ones, a long-running index quietly flatters itself, the losers are dropped from the historical record. An index's long-term return is partly an artefact of this constant pruning.

Free-float and corporate-action edge cases. Splits with awkward ratios, special dividends, spin-offs that create new entities, rights issues, all are fiddly to handle, and each is an opportunity to introduce a discontinuity if the divisor is adjusted incorrectly.

15.8 Indexes in Risk Control

Finally, indexes are not just measured, they are wired directly into the market's safety machinery. The most important example connects back to the circuit breakers discussed in Part III.

Market-wide circuit breakers halt *all* trading on an exchange when the market falls too far, too fast, and in the United States these are triggered by the **S&P 500 index**, not by any single stock. The thresholds are defined as percentage declines from the previous day's closing level [SEC / NYSE / Nasdaq market-wide circuit breaker rules]:

Level	S&P 500 decline from prior close	Action
Level 1	7%	15-minute halt (if before 3:25pm)
Level 2	13%	15-minute halt (if before 3:25pm)
Level 3	20%	Trading halted for the remainder of the day

The logic is the same single-symbol circuit breaker idea you already understand, monitor a reference price, halt when it moves too far, but applied to the *whole market* through its index. The index is the trigger that pauses thousands of individual books at once, giving participants time to absorb information and preventing a cascading crash. (Individual stocks have their own, separate Limit Up-Limit Down protections; the index drives the *market-wide* halt.)

Indexes feed risk control in other ways too:

- **Volatility indexes** such as the VIX are themselves derived from index-option prices and serve as a market-wide "fear gauge" that risk managers and regulators watch closely.
- **Portfolio risk limits** are frequently expressed relative to an index ("our tracking error against the benchmark must stay below 2%").
- **Margin and stress testing** for index derivatives depend on the index level and its volatility.

Key idea: An index is simultaneously a *measurement*, a *tradable reference*, and a *safety trigger*. The same number that tells a journalist "the market is down 2%" can, a few percent lower, automatically halt every order book on the exchange.

15.9 Key Takeaways

- An **index** condenses a basket of instruments into one number; it is a measurement, but enormous sums are invested, settled, and risk-managed against it.
- Modern serious indexes are **market-capitalisation-weighted**, often **free-float adjusted**, so each constituent's influence matches its economic size.
- The index level is the **aggregate market cap divided by a divisor**; the divisor normalises the level to a friendly base value and, when rescaled, keeps the index continuous through changes to the basket.
- **Corporate actions** (splits, dividends, issuance, mergers) change raw numbers without changing value; the **divisor is adjusted** to absorb them so the level does not jump.
- Major indexes like the **S&P 500** have strict eligibility rules and, in some cases, committee discretion; inclusion is not automatic.
- **Index inclusion** forces passive funds to buy and triggers anticipatory trading, the **index inclusion effect**, with large, real price consequences.
- **Correct, deterministic, auditable** index calculation is critical because derivatives settle on it, funds are priced on it, and risk controls fire on it.
- Watch for **stale prices, closing-print manipulation, reconstitution turbulence, reflexivity, and corporate-action edge cases**.
- Indexes are wired into **risk control**: in the US, **market-wide circuit breakers** halt all trading based on S&P 500 declines of 7%, 13%, and 20%.

CHAPTER 16

Benchmark Integrity: LIBOR, the FX Fix, and the Anatomy of a Riggable Number

The *Indexes* chapter made a structural argument: because so many instruments and obligations reference a single published number, the integrity of that number is a systemic concern, not a cosmetic one. That chapter treated the failure mode as *accidental*, a stale price, a wrong divisor, a miscalculation propagating downstream. This chapter treats the same structural fact from the other direction. When a widely referenced benchmark is computed from inputs that interested parties can *influence*, the benchmark is not merely fragile, it is **riggable**, and the history of the last two decades contains two enormous, exhaustively documented demonstrations of exactly that: the manipulation of **LIBOR** and the manipulation of the **4pm WM/Reuters foreign-exchange fix**. Together they produced tens of billions of dollars in fines, criminal convictions, the wholesale replacement of one of the most important numbers in finance, and a global re-engineering of how benchmarks are constructed. For anyone building systems that produce or consume reference prices, they are the definitive case studies in benchmark design.

16.1 What Makes a Benchmark Riggable

A benchmark is a number, published on a schedule, that summarises the price of something so that contracts can reference it instead of specifying a price directly. "The interest rate on this loan resets every quarter to three-month LIBOR plus 2%." "This fund is valued at the 4pm London fix." "This structured note pays out based on the closing level of the index." The reference is a convenience of staggering scale: at its peak, LIBOR was referenced by an estimated **$300+ trillion** in financial contracts worldwide, from multi-billion-dollar interest-rate swaps down to individual adjustable-rate mortgages and student loans.

Three properties, taken together, make a benchmark vulnerable to manipulation. They are worth stating precisely, because a benchmark that avoids all three is very hard to rig, and every reform described later in this chapter is an attack on one of them.

1. **The inputs are submitted or influenceable, not observed from arm's-length transactions.** If the benchmark is computed from numbers that participants *report*, or from a market thin enough that a participant can *move* it with their own trading, then the people who benefit from a particular outcome are, at least partly, the same people who determine the inputs.
2. **The people who set the inputs have positions that the benchmark settles.** Manipulation requires a motive. If a submitting bank also runs a large derivatives book that pays out against the benchmark, every basis point of the published number is worth money to the submitter directly.
3. **The calculation window is narrow and predictable.** If the benchmark is a snapshot taken over a short, known interval, a manipulator knows exactly when to concentrate their influence, and a small amount of activity at the right instant moves the published number by more than the same activity spread across the day would.

LIBOR failed the first two conditions catastrophically. The FX fix failed all three. Neither failure was subtle in retrospect; both were, for years, simply not looked at closely enough.

16.2 Case One: LIBOR, a Benchmark Built on Self-Reported Estimates

How LIBOR was constructed. The London Interbank Offered Rate was, in its classic form, not a measurement of transactions at all. Each business day, a panel of major banks was asked a hypothetical question: at what rate could *you* borrow unsecured funds from another bank, in a given currency, for a given tenor, right now? Each panel bank *submitted an estimate*. The administrator (historically the British Bankers' Association) discarded the highest and lowest quarter of the submissions and averaged the rest, a **trimmed mean**, to produce the day's published rate for each currency and tenor. Rates were published across several currencies and many maturities.

Every one of the three vulnerability conditions was present, and the first two acutely so. The inputs were *self-reported estimates*, not transactions (condition 1, in its strongest possible form, there was frequently no underlying trade to anchor a submission to at all, especially once interbank lending dried up in the crisis). And the submitting banks ran vast derivatives books whose value depended on where LIBOR fixed (condition 2). A trader whose swap position gained value if three-month LIBOR were a basis point higher could, and as the record established, sometimes did, simply ask a colleague on the submissions desk to nudge the estimate.

The two distinct abuses. The investigations uncovered two overlapping but separable forms of manipulation:

- **Derivatives-driven skewing.** Traders requested specific submission levels to benefit their own trading positions, sometimes across banks, coordinating so that several panel members skewed the same way and moved the trimmed mean. Internal messages, later quoted in regulatory findings and criminal trials, recorded these requests with damning candour.
- **Crisis-era "lowballing."** During 2007–09, a bank's LIBOR submission was read by the market as a signal of its own creditworthiness: a bank that admitted it could only borrow at a high rate was advertising distress. Several banks submitted artificially *low* rates to appear healthier than they were, distorting the benchmark for reasons of self-preservation rather than trading profit.

The consequences. The enforcement wave was among the largest in financial history. **Barclays** was the first to settle, in June 2012, paying roughly **$450 million** across the U.S. CFTC, the U.S. Department of Justice, and the U.K. FSA; the surrounding scandal cost the bank its chairman and chief executive. **UBS** settled in December 2012 for about **$1.5 billion**, and **RBS** in early 2013 for around **$610 million**, both with criminal components attached to subsidiaries. Further settlements followed against other panel banks and interdealer brokers. Aggregate LIBOR-related fines across the industry exceeded **$9 billion**, and several individual traders were criminally convicted and imprisoned. Crucially, the manipulation was possible for years despite the enormous stakes, precisely because no one was checking the submissions against transactions, there were often no transactions to check against.

The structural fix: replace the benchmark. LIBOR could not be patched, because its fatal flaw was its foundation, self-reported estimates of an activity (unsecured interbank term lending) that had largely stopped happening. The response was to move the world's benchmarks onto **transaction-based, secured, overnight rates** computed from observed trades in deep markets. In the United States, the Alternative Reference Rates Committee selected **SOFR** (the Secured Overnight Financing Rate), calculated by the Federal Reserve Bank of New York from actual overnight Treasury repurchase transactions, hundreds of billions of dollars of real trades every day, essentially impossible for any single participant to move or misreport. The United Kingdom moved to a reformed **SONIA**; other jurisdictions adopted their own transaction-based overnight rates. Administration of LIBOR itself had already been taken from the BBA and given to a regulated administrator (ICE Benchmark Administration) in 2014, but administration reform was not enough; the number had to be retired. Most USD LIBOR settings ceased to be published after **mid-2023**, ending a benchmark that had underpinned global finance for decades.

The lesson is stated most sharply as a design principle: *a benchmark should be computed from observed transactions in a market too deep for any participant to move, not from the self-interested reports of the parties it enriches.* Every property that made LIBOR riggable is inverted in SOFR.

16.3 Case Two: The 4pm FX Fix, a Benchmark Built on a Narrow Window

If LIBOR is the case study in riggable *inputs*, the WM/Reuters 4pm London fix is the case study in a riggable *window*, condition 3 above, and it shows that even a benchmark computed from *real transactions* can be manipulated if the calculation interval is short and predictable enough.

How the fix was constructed. Enormous quantities of foreign exchange, especially the passive, index-tracking flows of pension funds and asset managers, are transacted not at a negotiated price but at a daily **benchmark rate**: the WM/Reuters fix, of which the **4pm London fixing** is the most important. Historically the fix for major currency pairs was computed from actual trades and quotes captured over a short window, on the order of **one minute**, centred on 4:00pm London time. A client who instructed their bank to trade "at the 4pm fix" would receive whatever rate the fix produced, and the bank undertook to fill them at that rate.

Here is the structural problem. The banks knew, in advance, the direction and often the rough size of the client orders they had agreed to fill at the fix, because clients placed those orders ahead of time. That advance knowledge, combined with a one-minute calculation window, created both the motive and the mechanism. A bank sitting on a large "buy euros at the fix" client order could buy euros aggressively *during* the fixing window to push the fix rate up, fill the client at that inflated rate, and pocket the difference between what it paid to accumulate the position and the higher fix at which it sold to the client. The tactic was known, self-descriptively, as **"banging the close."**

The coordination. What turned a structural vulnerability into a scandal was collusion. Traders at different banks, holding offsetting or reinforcing fix orders, coordinated in electronic chat rooms, given names by their own participants that read as confessions: **"The Cartel," "The Bandits' Club," "The Mafia."** They shared confidential information about the size and direction of client orders and agreed how to trade the window jointly, so that their combined activity would move the fix reliably in the direction that maximised their collective profit at their own clients' expense.

The consequences. In November 2014 a coordinated set of regulators, the U.K. FCA, the U.S. CFTC, the U.S. OCC, and the Swiss FINMA, announced settlements with six major banks totalling roughly **$4.3 billion**, with the FCA's portion (about **£1.1 billion** across five banks) the largest it had ever imposed at that time. Further settlements and, in the United States, criminal pleas followed in 2015, pushing global FX-manipulation penalties well above **$10 billion**. As with LIBOR, individual traders faced prosecution, and the chat-room transcripts became the evidentiary core of the cases, again, an attributed, timestamped audit trail proving intent.

The structural fix: widen the window and change the incentives. The FX fix, unlike LIBOR, did not have to be abolished, because it was at least based on real transactions; it had to be made *harder to bang*. Following recommendations from the Financial Stability Board (2014–15), the calculation window for the major fixes was **widened from one minute to five minutes**. The arithmetic of manipulation is unforgiving of this change: moving a five-minute volume-weighted benchmark requires roughly five times the committed capital, and exposes the manipulator to five times the market risk, of moving a one-minute one, for the same effect on the published number. Alongside the wider window came clearer conduct rules separating the handling of client fix orders from proprietary trading, and greater transparency around fixing methodology. The reform is a clean illustration of attacking condition 3 directly: you cannot always change what a benchmark measures, but you can often change *how narrowly in time* it measures it, and a wider window is a structurally more manipulation-resistant one.

16.4 What the Developer Should Take From This

These are conduct-and-regulation stories, but they rest on system-design facts, and the engineering lessons generalise well beyond LIBOR and FX to any system that produces or consumes a reference price, including the closing auction of the *Opening and Closing Auction* chapter and the index calculation of the *Indexes* chapter.

Prefer observed transactions over reported estimates. A benchmark anchored to actual, arm's-length trades in a deep market removes the "riggable input" vulnerability at its root. If you are designing a reference

price, the first question is whether it can be computed from things that happened rather than things people said.

Prefer wide, and ideally unpredictable, calculation windows. The narrower and more predictable the snapshot, the cheaper it is to bang. Volume-weighting over a longer interval raises the capital cost of manipulation in direct proportion. Where a snapshot is unavoidable (an auction *is* a snapshot), the compensating controls are the ones the auction chapter described: imbalance dissemination, offsetting order types, single-price uncrossing, and audit capture.

Separate the parties who set the inputs from the parties who profit from the output, or, where you cannot separate them, surveil the boundary relentlessly. The LIBOR submissions desk sitting near the derivatives desk, and the FX fixing desk knowing the client order book, are the same structural conflict that self-match prevention and information-barrier controls exist to police inside an exchange.

The audit trail is what makes benchmarks enforceable after the fact. In every one of these cases, the decisive evidence was the timestamped, attributed record, chat logs, submission histories, order timings, that reconstructed intent. This is the same completeness-and-immutability requirement that the *Regulatory Surveillance* and *Determinism, Replay, and Persistence* chapters place on the exchange itself. A benchmark you cannot reconstruct is a benchmark you cannot defend.

> **Key idea:** A benchmark is riggable to the exact extent that interested parties can influence its inputs, hold positions that it settles, and know when its narrow calculation window falls. LIBOR failed on inputs (self-reported estimates by the banks the rate enriched) and was abolished in favour of transaction-based rates like SOFR; the 4pm FX fix failed on its window (one predictable minute, tradeable by banks who knew the client flow) and was hardened by widening the window to five minutes. Both produced fines in the billions and criminal convictions, and both teach the same design rule the *Indexes* chapter began: a number that the whole market leans on must be engineered, measured from deep observed markets, over windows too wide to bang, with the input-setters walled off from the output-beneficiaries, and fully captured in an audit trail, because its importance is precisely what makes it a target.

16.5 References

- U.S. Commodity Futures Trading Commission, U.S. Department of Justice, and U.K. Financial Services Authority: orders and statements of facts in the Barclays LIBOR settlement (June 2012); UBS (December 2012); RBS (February 2013).
- U.K. Financial Conduct Authority, U.S. CFTC, U.S. Office of the Comptroller of the Currency, and Swiss FINMA: final notices and orders in the November 2014 foreign-exchange benchmark settlements; U.S. Department of Justice FX-related guilty pleas (May 2015).
- Financial Stability Board, *Foreign Exchange Benchmarks* (final report, September 2014) and *Reforming Major Interest Rate Benchmarks* (July 2014 and subsequent progress reports), including the recommendation to widen the fix calculation window.
- Martin Wheatley, *The Wheatley Review of LIBOR* (final report, HM Treasury, September 2012), on transferring administration and reforming submission practice.
- Federal Reserve Bank of New York and the Alternative Reference Rates Committee: publications on SOFR and the transition away from USD LIBOR (2017–2023); ICE Benchmark Administration assumption of LIBOR administration (2014); cessation of most USD LIBOR settings (2023).

Glossary

The following definitions are written to be concise and standalone, useful as a quick reference after reading the document.

A

Automated Collar Expansion (ACE): This book's name for rules that, when an instrument's likely price lies outside its protective collar, pause only that instrument, keep collecting orders, and widen the collar in published steps until the indicative auction price fits, instead of either blocking a genuine repricing or trading at an absurd price. Real venues use their own names (Nasdaq's Halt Cross price protections, NYSE's auction-collar extension after LULD pauses, Xetra's extended volatility interruption). See *Price Collars*, Part III.

Adverse Selection: The risk a market maker faces when their quote is hit by a counterparty who has superior information about where prices are heading, causing the market maker to trade at a price that will quickly move against them. A market maker who is repeatedly adversely selected will lose money even while quoting at "correct" prices. MMP exists specifically to limit adverse selection damage.

Aggressive order: (also called a **taker**), an incoming order that immediately trades against resting orders. It "sweeps" the book.

Aggressor / Taker: The participant whose incoming order triggered a match by crossing the spread to meet a resting order. The aggressor "takes" liquidity that was already available. Contrast with *Maker*, the participant whose resting order was already in the book.

Assignment: The process by which an options clearing house (the OCC, in the US) selects which option writer(s) must fulfil the obligation created when a holder exercises. Assignment passes from the OCC to a clearing member and from the clearing member to a specific client, typically via random or FIFO allocation. See *Exercise*, *Pin Risk*.

Automated Market Maker (AMM): A smart-contract-based pricing mechanism used by decentralised cryptocurrency exchanges in place of an order book. Prices are set algorithmically from the ratio of two assets held in a liquidity pool, commonly via a constant-product formula, rather than by matching a buy order against a sell order. See *Decentralised Exchange (DEX)*.

Automatic Exercise: The OCC's practice of exercising any in-the-money option on a holder's behalf at expiry (by a threshold as small as $0.01 in-the-money) unless the holder has filed a contrary instruction. See *Exercise*.

Alternative Trading System (ATS): A trading venue that matches orders but is not a registered national securities exchange. Includes ECNs and dark pools. Subject to lighter regulation than exchanges in some jurisdictions.

Arbitrage: The simultaneous purchase and sale of the same (or equivalent) asset on different markets to profit from a price difference. Pure arbitrage is risk-free (buy cheap here, sell expensive there at the same moment); in practice, most arbitrage involves some timing or execution risk. Arbitrage is the mechanism by which prices on different venues are kept consistent, arbitrageurs quickly eliminate price discrepancies, making markets more efficient. **Latency arbitrage** is a specific form exploiting the time it takes for price changes

to propagate between venues.

Ask / Offer: The price at which a seller is willing to sell. The best (lowest) ask is the top of the sell side of the order book. The collection of all active asks forms the sell side of the book.

ATC (At-The-Close): An order TIF valid only during the closing auction.

ATO (At-The-Open): An order TIF valid only during the opening auction.

Auction / Call Auction / Uncross / Fixing: A trading mechanism where orders are collected over a period and then matched simultaneously at a single equilibrium price. Contrasts with continuous matching where orders match one at a time as they arrive.

Auction Imbalance: The excess of buy orders over sell orders (or vice versa) at the indicative uncross price during an auction. Published by exchanges to encourage offsetting orders before uncrossing.

Audit Trail: The complete, immutable, time-ordered record of every event in the exchange system, every order, fill, cancellation, and rejection, with precise timestamps. Required for regulatory reporting and replay-based recovery.

B

Big Bang: The 27 October 1986 deregulation of the London Stock Exchange, which abolished fixed commissions, ended the broker/jobber separation, permitted outside and foreign ownership of member firms, and replaced floor trading with the screen-based SEAQ system, all on a single day. The UK counterpart to the US's 1975 "May Day." See *May Day*.

Backwardation: A futures market in which near-dated contracts are more expensive than later-dated ones, so a calendar spread priced as near minus far is positive. The opposite of *contango*.

Basis: In derivatives and futures markets, the price difference between two related instruments, most commonly between the spot/cash price and a futures price on the same underlying, or between two futures contracts with different expiry dates. In the implied-order examples in this document (the *Implied Orders and Synthetic Liquidity* chapter of Part II), the spread price of −$2 between the January and February crude oil contracts is the basis between those two contract months: January is $2 cheaper than February. The basis is not fixed; it changes continuously as supply and demand for the near and far months evolve independently. **Basis risk** is the risk that the expected relationship between two instruments diverges unexpectedly, a spread trader who expected the two legs to move together may find they do not, leaving an unintended net exposure. See *Implied Order*, *Calendar Spread*.

Bear Market: A sustained decline of 20% or more from a recent market peak. Originates from the image of a bear swiping its paws downward. Contrast with *Bull Market*.

Best Ask: The lowest resting sell price in the order book, the price at which a buyer can trade immediately.

Best Bid: The highest resting buy price in the order book, the price at which a seller can trade immediately.

Best Execution: The regulatory obligation for brokers and investment firms to take all reasonable steps to achieve the best possible result for their clients when executing orders. "Best" considers price, execution costs, speed, likelihood of execution, and market impact simultaneously, not just the quoted price in isolation. Mandated by MiFID II in the EU and the SEC's duty of best execution in the US. Best execution is the regulatory foundation that makes smart order routing (SOR) necessary: brokers must demonstrate their routing decisions genuinely serve clients' interests, not just the broker's economics.

Bid: The price at which a buyer is willing to buy. The best (highest) bid is the top of the buy side of the order book.

Bid-Ask Imbalance: A measure of depth asymmetry: bid_depth / (bid_depth + ask_depth), computed within a symmetric price window around the mid. Values near 1 indicate heavy buying interest; values near 0 indicate heavy selling interest. Used as a short-term price direction signal.

Blue Chip: A large, financially sound, well-established company, the most prestigious tier of the equity market. The term originates from poker, where blue chips represent the highest denomination.

Bond: A standardised debt instrument. The issuing company borrows money and promises to pay periodic

interest (coupon payments) and return the face value (principal) at maturity. Bondholders are creditors, not owners, and are paid before equity holders in a bankruptcy. Bonds trade on secondary markets after issuance, allowing investors to exit before maturity. The financial term "bond" traces its roots back to the 14th century from the Old English word bindan and band, which eventually evolved into the standard English word "bind". The word was originally used to describe something that literally ties, fastens, or holds things together. Because a financial bond fundamentally operates as a formal, legally binding contract where an issuer promises to repay a loan with interest, the term stuck.

Book-Building: The customary way a conventional IPO is priced: the issuer and its underwriters publish a price range, collect indications of interest from institutional investors during a roadshow, revise the range, and set the final offer price the evening before trading starts. A process of price discovery, not a legal requirement.

Book Depth: The total volume (quantity of shares) resting near the top of the order book. A deep book absorbs large orders without significant price impact. A thin (or shallow) book moves quickly on even moderate order flow.

Book Snapshot: A periodic complete view of the current order book state (all price levels and their quantities) published by the exchange for subscribers who need to get up to speed.

Borrow Rate: The annualised fee charged to a short seller for borrowing shares from a lender. Easy-to-borrow large-cap stocks typically have borrow rates near zero. Hard-to-borrow, heavily-shorted, or illiquid stocks can carry rates of 5–50% or more per annum, applied daily to the short position. Rising borrow rates signal increasing short interest and decreasing availability of lendable shares, and are themselves a market signal.

Bracket Order: A strategy combining an entry order with both a take-profit LIMIT and a stop-loss STOP, typically implemented as an OCO pair.

Broker: An intermediary who executes orders on behalf of clients without trading for their own account. The word traces to Old French/Norse roots meaning someone who opens a cask and sells by the cup, a retail intermediary.

Bull Market: A sustained rise of 20% or more from a recent market trough. Originates from the image of a bull thrusting its horns upward. Contrast with *Bear Market*.

Buy-In: A remedy for failed settlement: if a seller cannot deliver securities, the buyer purchases them in the open market at the seller's expense.

C

Calendar Spread: A derivatives strategy involving the simultaneous purchase and sale of contracts on the same underlying instrument but with different expiry dates, for example, buy the March futures contract, sell the June futures contract. Used to profit from expected changes in the price difference (the "spread") between contract months. On CME and most futures venues it is also a listed instrument with its own order book, priced as near month minus far month; *buying* the spread means buying the near month and selling the far month, so the buyer profits when that difference rises. Spread prices are routinely negative. Requires cross-symbol coordination in the matching engine. See *Implied Order*.

Call auction: Synonym for auction, "call" refers to the fact that all participants are "called" to submit orders before a single price is determined.

Capital Appreciation: An increase in the market price of a share (or other asset) above the price paid. The primary way equity investors generate returns.

Capital Gain: The profit realised when an asset is sold for more than it was purchased for.

Cascade Cancel: When a COMBO order fails (any leg is cancelled or rejected), all remaining unfilled legs of the combo are automatically cancelled. Previously executed fills are not reversed.

CCP (Central Counterparty Clearing House): An entity that stands between buyer and seller in a trade, becoming the legal counterparty to both, eliminating direct counterparty credit risk.

Circuit Breaker: An automatic trading halt triggered when price movement exceeds a threshold within a time window, designed to prevent cascading automated trading crashes.

Clearly Erroneous Execution (CEE): A trade whose price deviates from a reference price by more than a published, price-tiered percentage threshold, making it eligible for cancellation (busting) or price adjustment under exchange rules. Reviewed execution-by-execution within a short time window (commonly 30 minutes); multi-venue events require coordinated rulings across exchanges. Distinct from a trade that is merely unwelcome to the participant who sent it, the test is an objective price deviation, not the sender's intent. See *Trade Busting and Clearly Erroneous Trades*, Part III.

Conformance Testing: The process by which a prospective exchange participant's trading system is verified, against a prescribed test script in a certification (UAT) environment, to correctly implement the exchange's protocol and expected behaviours before being granted production credentials.

Contango: A futures market in which later-dated contracts are more expensive than near-dated ones (common when the cost of storing the underlying is high), so a calendar spread priced as near minus far is negative. The opposite of *backwardation*.

Continued Listing Standards: The ongoing, ordinarily automated compliance thresholds (minimum bid price, market value of public float, shareholder count) a listed company must keep satisfying after its initial listing, distinct from the one-time *Initial Listing Standards*. Breach triggers a deficiency notice and cure period before involuntary delisting.

Crossed Market: A condition where the best bid on one trading venue is higher than the best ask on another venue for the same instrument, a transient artefact of fragmentation and quote-propagation latency across independent venues, and a theoretical risk-free arbitrage opportunity for whoever can act on it fastest. See *Locked Market*.

Cure Period: The window (commonly 180 days, sometimes extended) an exchange gives a listed company to regain compliance with a continued listing standard after a deficiency notice, before involuntary delisting proceedings begin.

Clearing: The process of confirming and recording a trade, and (via a CCP) guaranteeing its settlement.

Co-location: Placing a participant's trading servers physically inside the same data centre as the exchange's matching engine, minimising network round-trip latency to microseconds.

Collar Band / Price Collar: A price filter that rejects orders whose submitted price deviates too far from a reference price. Two distinct variants: the *static price collar* (compares against the last close) and the *dynamic price collar* (compares against the last trade). See those entries for details.

Combo Order: A single instruction to execute orders in multiple instruments simultaneously. Each component is a leg.

Conflation (Market Data): Merging consecutive market data updates for the same price level into a single message under high-load conditions. Reduces message volume at the cost of losing intermediate states.

Continuous Trading: The normal trading mode where orders match as soon as they arrive and a compatible counterparty is present. Contrasts with auction mode.

Contract Multiplier: For futures and options, the dollar (or other currency) value of a one-unit price move per contract. The E-mini S&P 500 has a multiplier of $50, a 1-point move in the index is worth $50 per contract. Essential reference data: a wrong multiplier affects every P&L and margin calculation.

Corporate Action: An event affecting a company's share structure, stock splits, dividends, mergers, delistings. Requires adjustments to all open orders, positions, and reference data in exchange systems.

Counterparty Risk: The risk that the other party to a trade defaults before settlement is complete. Cleared via CCP novation, which replaces bilateral counterparty risk with exposure to the CCP.

Coupon: The periodic interest payment on a bond. Named after the physical coupon strips that bond holders historically detached and redeemed for payment.

Crossing the Spread: Submitting an order at a price that matches an existing resting order, triggering an immediate fill.

Cum-Dividend: The period during which a share trades *with* entitlement to a declared dividend, the buyer receives the upcoming payment. A share trades cum-dividend from the dividend announcement up to (but not including) the ex-dividend date. After the ex-dividend date the share trades *ex-dividend* and the buyer is not

entitled to the payment. The price typically drops by approximately the dividend amount on the ex-dividend date.

CUSIP: A 9-character identifier for US and Canadian securities used in clearing and settlement. Analogous to ISIN.

Custody: The safekeeping of securities on behalf of investors by a broker or custodian bank, with records maintained at a central securities depository.

D

Dark Fibre: Unused optical fibre cable already laid in the ground but not yet carrying data traffic ("lit"). Trading firms and data providers lease dark fibre from infrastructure companies and run their own optical transceivers, creating private, dedicated network links between financial data centres. Dark fibre links are used between major exchange co-location sites (e.g., NYSE in Mahwah and NASDAQ in Carteret) to reduce cross-venue latency for arbitrage and hedging strategies.

Decentralised Exchange (DEX): A cryptocurrency trading venue that executes trades directly via smart contracts on a blockchain, with no central operator holding custody of participant assets and, commonly, no traditional order book, matching instead via an *Automated Market Maker*. Contrast with a *Centralised Exchange (CEX)*, which operates a conventional matching engine and custodies participant assets itself.

Decimalization: The switch from fractional dollar price quoting (eighths, then sixteenths) to decimal cent quoting in US equity markets, phased in between August 2000 and April 2001. Quoted spreads on liquid stocks roughly halved as a direct result. The direct historical motivation for Rule 612's sub-penny quoting floor. See *Tick Sizes and Fractional Ticks*, Part II.

De-SPAC: The merger transaction in which a Special Purpose Acquisition Company (SPAC) combines with a private operating company, resulting in the operating company becoming publicly traded under the (renamed) shell's listing. See *SPAC*.

Direct Listing: A method of going public in which a company's existing shares begin trading on an exchange via the opening auction mechanism, with no new shares issued, no underwriter guarantee, and no pre-set offer price. Contrast with a traditional underwritten IPO. Pioneered at scale by Spotify (2018) and Slack (2019), both on NYSE.

Dark Pool: A private trading venue where orders are not displayed in a public order book. Participants submit orders and are matched against other participants' orders, typically at the midpoint of the NBBO, with the trade only reported publicly after it occurs. Used by large institutional investors to trade without revealing their intentions and causing adverse price movement. Contrast with *lit markets* (public exchanges) where resting orders are visible to all. See *Regulatory Surveillance*.

DAY: An order TIF valid only for the current trading session.

Delivery versus Payment (DvP): The settlement principle that the transfer of securities and cash happen simultaneously and conditionally, neither is released without the other. Eliminates principal risk in settlement.

Depth (Market Depth): The total quantity of resting orders available at each price level. Deep markets can absorb large orders with minimal price impact.

Depth-of-Book (Level 2): Market data showing multiple price levels and their resting quantities, rather than only the best bid and ask.

Designated Market Maker (DMM): A firm assigned by an exchange to maintain fair and orderly markets in specific securities, with both privileges and obligations. NYSE's term; Nasdaq calls them Market Makers.

Determinism (Exchange): The property that given the same initial state and the same ordered input messages, the matching engine always produces identical outputs. Essential for audit trail replay and disaster recovery.

Discounted Cash Flow (DCF): A valuation method that estimates a company's future free cash flows and discounts each one back to today's money at a rate reflecting risk and the time value of money. For young, fast-growing firms most of the result usually sits in the *terminal value*, the lump that stands for all years beyond the forecast.

Dilution: The reduction in existing shareholders' ownership percentage when new shares are issued. Selling 20% of a company to new investors reduces existing owners to 80%.

Dividend: A cash payment made by a company to its shareholders, distributed from retained profits. Not guaranteed, the board of directors decides whether, when, and how much to pay. Companies that pay regular dividends are typically mature, profitable businesses with stable cash flows. High-growth companies often pay no dividend, choosing to reinvest profits in expansion. The key dates are: the *declaration date* (when the board announces the dividend), the *ex-dividend date* (buyers on or after this date do not receive the payment), the *record date* (the register of shareholders eligible to receive payment), and the *payment date* (when the cash is distributed).

Drop Copy: A real-time copy of all order lifecycle events for a specific participant, sent to a designated third party such as a clearing broker or risk system.

DTCC (Depository Trust and Clearing Corporation): The central clearing and settlement infrastructure for US equity markets. Its subsidiary DTC is the US central securities depository.

Dynamic Price Collar (Reference Price Filter): A pre-trade risk control that rejects any order whose submitted price deviates more than a configured percentage from the most recent traded price. Protects against algorithms that walk the price incrementally through a series of small trades. Contrast with *Static Price Collar*, which uses the last close as the reference. Together, the two collars provide overlapping protection against both sudden errors and gradual drift.

E

ECN (Electronic Communication Network): An electronic trading system that automatically matches buy and sell orders, typically with narrower spreads than exchanges but less regulatory oversight.

Equilibrium Price (Auction Price): The single price at which the maximum volume can trade in a call auction.

Equity: Ownership stake in a company. Equity holders (shareholders) have a residual claim on the company's assets and earnings after all creditors are paid.

ETF (Exchange-Traded Fund): A basket of securities (shares, bonds, commodities) that itself trades as a single share on an exchange.

Ex-Dividend Date: The date from which a stock trades without entitlement to a declared dividend. Buyers on or after this date do not receive the dividend.

Execution Algorithm (Algo): Software that breaks a large institutional order into smaller pieces and executes them over time to minimise market impact and achieve a target average price. Common types include VWAP (match the day's volume-weighted average price), TWAP (distribute uniformly over time), Implementation Shortfall (minimise the gap between decision price and execution price), and POV/In-Line (participate at a fixed percentage of market volume). Execution algorithms generate the majority of order flow on modern exchanges.

Exercise: An option holder's invocation of their right to buy (call) or sell (put) the underlying at the strike price. American-style options may be exercised any time up to expiry; European-style options only at expiry. See *Automatic Exercise*, *Assignment*.

F

Face Value (Bond): The principal amount of a bond, the amount the issuer promises to repay at maturity, and the basis on which coupon payments are calculated.

Failover: The process of switching from a failed primary system to a secondary (backup) system.

Fat Finger: Industry slang for an accidental trading error caused by mistyping an order parameter (quantity, price, etc.).

FIFO (First-In, First-Out): Within a price level, orders are filled in the order they arrived. Earlier arrivals have priority.

Fill: An execution, a match between a buy and sell order resulting in a trade. Partial fills consume part of an order; a complete fill consumes all of it.

FOK (Fill-Or-Kill): An order that must be completely filled immediately or cancelled entirely.

FIX (Financial Information eXchange): The industry-standard protocol for electronic order submission, used by most exchanges and brokers globally.

Flat: Having no open position in a symbol (net quantity = 0). Neither long nor short.

Floor Broker: Historically, a broker who executed trades on the physical exchange floor. The term persists in regulations and informal speech.

FPGA (Field-Programmable Gate Array): Reconfigurable hardware chips that can implement market data parsing or order routing in dedicated logic circuits, achieving nanosecond-level latency.

Front Running: Acting on advance knowledge of a pending client order to trade for one's own account before executing the client's order, then profiting from the price movement the client's order causes. Illegal under market manipulation regulations and a serious breach of fiduciary duty. Front running is detected by surveillance systems that flag cases where a participant's own trades immediately precede large client executions in the same direction.

Futures Contract: A standardised agreement to buy or sell a specified quantity of an underlying asset (commodity, index, currency, interest rate) at a predetermined price on a specified future date. Obligations are legally binding. Futures are exchange-traded, margined, and marked-to-market daily. Unlike equity ownership, futures do not represent ownership of an underlying asset, they represent an obligation about a future transaction.

G

Greenshoe (Over-Allotment Option): An option letting IPO underwriters sell up to 15% more shares than the base deal. If the price sags after listing they buy shares back in the market to support it; if it rises they exercise the option and buy the extra shares from the issuer at the offer price.

Gap (Market Data): A missing sequence number in a market data feed, indicating a lost message. Subscribers must request retransmission or resynchronise from the next full snapshot.

Gateway: The participant-facing interface to the exchange. Handles authentication, session management, and message translation.

Going Long: Owning an asset in expectation that its price will rise, so it can be sold later at a profit. You are "long" the asset. The term originates in physical commodity trading: a merchant who had purchased durable goods, grain, spices, metal, cloth, and was holding them in a warehouse was described as "long" in those goods. The word captured both *possession* (you own something tangible) and *duration* (durable goods last a long time and can be held through seasons or years). Being long in modern finance carries exactly the same meaning: you hold the asset and are exposed to its price over time. A long position profits when the price rises and loses when it falls.

Going Short / Short Selling: Selling an asset you do not currently own, typically by borrowing it from another party, with the intention of buying it back later at a lower price and returning it, pocketing the difference. The term originates in physical commodity trading: a merchant who had sold goods they did not yet possess, committing to deliver more than their warehouse held, was "short" of the goods. The same word family as "falling short," "short of supplies," or "shortage", deficient, insufficient, lacking. In the early forward markets of Amsterdam and London, a short seller would sell forward (promise future delivery), then purchase the goods in the market before delivery, hoping prices had fallen. Isaac Le Maire's 1609 campaign against VOC shares is the earliest recorded large-scale short selling operation. Today, a short position profits when the price falls (you buy back cheaper than you sold) and loses when it rises, in theory without limit, since a rising price has no ceiling.

GTC (Good-Till-Cancelled): An order TIF that persists across trading sessions until filled or cancelled.

GTD (Good-Till-Date): An order Time-In-Force variant that keeps an order active until it fills, is explicitly cancelled, or reaches a specified expiry date, whichever comes first. On the expiry date the order is automat-

ically cancelled. Provides the persistence of GTC without the open-ended risk of an order remaining active indefinitely. Requires the exchange scheduler to track each GTD order's expiry and issue the cancellation at the right time.

H

Heap: A data structure that gives O(1) access to the minimum (or maximum) element. Used in matching engines to efficiently maintain best price access.

High-Frequency Trading (HFT): Automated trading strategies that execute very large numbers of orders at extremely high speeds, typically holding positions for milliseconds or seconds rather than minutes or days. HFT firms invest heavily in low-latency infrastructure (co-location, microwave links, FPGAs) to gain speed advantages. Strategies include latency arbitrage, market making, and statistical arbitrage. Controversial because speed advantages can disadvantage slower participants; the subject of the IEX speed bump.

I

Iceberg Order / Hidden Order / Reserve Order: An order that shows only a small visible peak while hiding a large reserve. Each time the visible portion fills, the order replenishes from the reserve.

Imbalance: The difference between total buy and sell quantity at the equilibrium price. Expressed as a side (BUY or SELL) and a quantity.

Immediate-Or-Cancel (IOC): An order that fills as much as possible immediately and cancels any unfilled remainder. Never rests in the book.

Implementation Shortfall (IS) / Arrival Price Algorithm: An execution algorithm that minimises the difference between the *decision price* (the market price when the investment decision was made) and the actual average execution price. IS algos trade faster when the price moves against the position (urgency increases to avoid further shortfall) and slower when price moves favourably. More adaptive and sophisticated than VWAP or TWAP; preferred by quantitative funds whose models predict short-term price moves.

Implied Generation: A count of how many real orders an implied quote combines, minus one. A first-generation implied quote is built from two real orders; a second-generation quote uses a first-generation implied quote as an ingredient and so combines three. Venues restrict how many generations they compute because the work grows explosively. See *Multi-Layer Implied Orders and the Combinatorial Wall*, Part II.

Implied Matching: A technique used in derivatives markets where the exchange combines resting spread and outright orders (in different books) into a synthetic order in a third book, and lets incoming orders trade against it, all contributing orders executing together. For example: a resting spread order to buy March/sell June combined with a resting outright June bid creates an implied March bid. Implemented on CME Globex, ICE, and (as "synthetic matching") Eurex. Implied matching creates no new liquidity: it makes existing orders reachable from another book.

Implied Order: An order derived by the matching engine from a combination of existing spread and outright orders. Not submitted by any participant, computed from pre-existing orders and published in the relevant outright or spread book. When an implied order matches, all contributing orders execute simultaneously and atomically. If any contributing order is cancelled or filled by other means, the implied order shrinks or disappears in the same processing step. Implied orders never act as the aggressor, and at the same price real orders trade before them.

Implied IN: In CME's terminology, an implied order in the *spread* book created from orders in the outright books. Example: a January bid at \$74.40 and a February ask at \$76.60 together imply a Jan/Feb spread *bid* at $74.40 - 76.60 = -\$2.20$.

Implied OUT: In CME's terminology, an implied order in an *outright* book created from a spread order plus an outright order. Example: a January ask at \$75.00 and a Jan/Feb spread *bid* at −\$2.00 together imply a February *ask* at $75.00 - (-2.00) = \$77.00$.

Inactivation (Quote): Automatically cancelling or suspending the surviving leg of a two-sided quote the moment the other leg is filled. The market maker must re-quote both sides afresh within their re-quoting

window. Used in conservative quote refresh policies (Eurex-style). Protects the market maker from being left with a stale unhedged quote after their inventory position has changed.

Indicative Uncross Price: During an auction accumulation period, the price at which uncrossing *would* occur if matching happened at that moment. Recalculated after every order arrival and published as a signal to help participants decide whether to adjust their orders before the actual uncross.

Information Asymmetry: A situation where some market participants have access to information (faster data, better analytics) that others do not, giving them a trading advantage. Auctions reduce information asymmetry by letting all participants submit orders before any execution occurs.

Informed Flow: Order flow that originates from participants who have an informational advantage about near-term price direction, for example, participants who have seen relevant news before others. When a market maker is repeatedly hit by informed flow, they accumulate inventory at prices that are about to move against them. Informed flow is the root cause of adverse selection.

Initial Listing Standards: The quantitative (minimum share price, public float value, shareholder count) and qualitative thresholds a company must meet before an exchange will admit its shares to trading for the first time. Distinct from *Continued Listing Standards*, which apply on an ongoing basis thereafter.

Initial Margin: Collateral deposited when opening a position to cover potential losses. Calculated by the CCP based on the instrument's historical volatility and the position size.

Insider Trading: Trading on material, non-public information, for example, buying shares before a publicly unannounced merger is announced. Illegal in all major jurisdictions. The exchange's audit trail is a key source of evidence in insider trading investigations, as regulators can reconstruct exactly who traded, when, and at what prices relative to when news became public.

Instrument / Symbol: The specific financial product being traded. Each instrument has its own order book.

Internaliser: A broker-dealer that matches client orders internally against its own inventory or other clients' orders, rather than routing to an exchange. A form of OTC secondary trading.

IOC (Immediate-Or-Cancel): A time-in-force (and order type) where an order sweeps available liquidity at the limit price and any unfilled remainder is immediately cancelled. Never rests on the book

IPO (Initial Public Offering): The first sale of a company's shares to the general public, after which the shares are listed and traded on a stock exchange.

ISIN (International Securities Identification Number): A 12-character global identifier for a security, independent of the exchange it trades on. A single company has one ISIN regardless of where its shares are listed.

ITCH (NASDAQ TotalView-ITCH): NASDAQ's binary, message-based market data feed protocol for order book dissemination. ITCH publishes sequenced events such as add order, modify/replace, cancel/delete, executions, and auction/trade messages so subscribers can reconstruct the full limit order book deterministically from the stream. Official specification: NASDAQ TotalView-ITCH 5.0.

K

Kernel Bypass: Technology that allows a network application to communicate directly with network hardware, bypassing the operating system kernel and reducing latency from microseconds to sub-microsecond.

Kill Switch: An emergency control that immediately cancels all resting orders and quotes for a specific participant AND marks their gateway as inactive, preventing new order submission until the participant reconnects and re-authenticates. Triggered by the participant (emergency self-halt), the exchange operator, or automatically on gateway disconnect. Mandatory under most exchange regulations including MiFID II. Contrast with *Mass Cancel*, which cancels orders but leaves the gateway active.

L

Latency: The time delay between an action and its effect. In exchanges, latency is measured in microseconds or nanoseconds. Low latency is a competitive advantage.

Latency Arbitrage: Trading that profits from the brief period between when a price changes on one venue and when that change propagates to another venue.

Layering: A market manipulation practice: placing large orders to create a false impression of market depth, then cancelling them once the desired price movement occurs.

LCH: A major international clearing house, part of the London Stock Exchange Group. Clears interest rate swaps, equities, foreign exchange, and other products.

Leg: One component order within a combo order or spread strategy.

Leg Risk / Legging Risk: The risk that one leg of a combo order fills while another does not, leaving an unintended single-instrument position.

Level 1 Data: The best bid and ask prices and quantities only.

Level 2 Data (Market Depth): The full order book, all price levels and their quantities.

Limit Order: An order to trade at a specified price or better. May rest in the book if not immediately matched.

Limit Up-Limit Down (LULD): A US regulatory circuit breaker mechanism that pauses trading in an individual stock if its price moves outside a percentage band (the "price band") calculated from a reference price. The bands are tighter for liquid large-cap stocks and wider for smaller, more volatile ones. Trades outside the band are not allowed. If the best bid or offer sits exactly at the band (a "limit state") and does not recover within 15 seconds, a 5-minute trading pause begins. Implemented across all US equity trading venues.

Liquidity: The ease with which an asset can be bought or sold without significantly affecting its price. High liquidity = tight spread, large depth, easy trading.

Locate: The process of finding and reserving a source of borrowable shares before executing a short sale. Under US Regulation SHO, broker-dealers must have reasonable grounds to believe shares can be borrowed before accepting a short sale order. "Failing to locate" before shorting is called naked short selling, which is illegal in most jurisdictions.

Lock-Up: A contractual agreement, typically 180 days, under which insiders and pre-IPO investors promise not to sell their shares after an IPO. It is a contract with the underwriters, not a legal rule, and its expiry can put pressure on the share price.

Locked Market: A condition where the best bid on one venue equals the best ask on another venue for the same instrument, a transient artefact of fragmentation and quote-propagation latency. See *Crossed Market*.

Long Position: Owning a positive quantity of shares. You profit if the price rises. Expressed as a positive number P&L ledger.

Lot: The standard unit of quantity for an instrument.

M

Maintenance Margin: The minimum collateral balance that must be maintained in a margin account. Falling below this level triggers a margin call.

Maker: The participant whose resting order was already in the book when a fill occurred. Contrast with *Aggressor/Taker*.

Margin Call: A demand from a clearing house or broker for a participant to deposit additional collateral to restore their margin account to the required level.

Mark-to-Market: The daily revaluation of positions at current market prices, with cash payments made to reflect daily gains and losses. The mechanism by which variation margin is calculated.

Market capitalisation (market cap): The total market value of all a company's outstanding shares, calculated as share price multiplied by the total number of shares in existence. If Apple has approximately 15.4 billion shares outstanding and each trades at $190, Apple's market cap is roughly $2.9 trillion [1]. Market cap is the most widely used shorthand for a company's size. When rankings refer to "the world's largest exchange by listed market cap," they are summing the market caps of every company listed there.

Market Fragmentation: The distribution of trading volume for a single instrument across multiple competing venues, exchanges, dark pools, ATSs, and internalisers. Modern equity markets are highly fragmented; in the US, AAPL may trade simultaneously across a dozen registered exchanges plus numerous alternative venues. Fragmentation improves competition and reduces fees but complicates order routing and best-execution analysis. Regulatory frameworks like Regulation NMS (US) and MiFID II (EU) were introduced partly in response to fragmentation.

Market Impact / Slippage: The adverse price movement caused by a large order sweeping through multiple price levels.

Market Impact Estimation: The process of computing, before order submission, the average price you would pay (or receive) by sweeping through available book levels for a given order size. Derived from the cumulative depth profile. Used by execution algorithms to size order slices and choose between aggressive execution and passive accumulation.

Market Maker: A professional participant obligated to continuously provide both buy and sell prices, ensuring liquidity for other participants.

Market Maker Protection (MMP): An exchange-provided mechanism that automatically cancels all of a market maker's resting quotes when fills arrive at a rate exceeding a configured threshold within a rolling time window. Protects market makers from being rapidly picked off by participants with superior information (adverse selection). After MMP fires, the market maker enters a protection period before re-quoting. Parameters (fill count, time window) are configured per market maker per symbol.

Market Order: An order that attempts immediate execution against available liquidity at the best obtainable prices. It prioritises execution certainty over price certainty, but execution is still subject to exchange state, available liquidity, price collars, and risk controls. A market order submitted during a halt, against an empty book, or rejected by a pre-trade risk check will not fill.

Mass Cancel: A participant-initiated message type that cancels multiple resting orders simultaneously, all orders for a symbol, all orders of a given type, or all orders system-wide. The gateway connection remains active after a mass cancel; the participant can submit new orders immediately. Contrast with *Kill Switch*, which inactivates the gateway and requires re-authentication.

Matching Engine: The core software that manages order books and matches buy and sell orders to produce trades.

Maturity (Bond): The date on which a bond's issuer must repay the face value to the bondholder.

May Day: 1 May 1975, the date the SEC abolished fixed brokerage commissions on the NYSE, ending nearly two centuries of fixed-rate trading and creating the discount-brokerage industry. The US counterpart to the UK's 1986 "Big Bang." See *Big Bang*.

Microwave Link: A radio communications link using microwave frequencies. Between financial data centres, microwave links provide lower latency than fibre-optic cables because radio waves travel faster through air than light through glass.

Mid Price: The average of the best bid and best ask: (best_bid + best_ask) / 2.

Midpoint Peg Order: An order whose price continuously tracks the current mid price (the average of the best bid and best ask). Executes only when a counterparty arrives whose order is also willing to trade at the midpoint. The mid is a *better* price for both parties than the quoted bid or ask, the buyer pays less than the ask, the seller receives more than the bid; the spread is shared rather than fully paid. Provides price improvement at the cost of uncertain execution timing: there is no guarantee a counterparty will appear. Common in dark pools; also supported on IEX on the lit market. Never competes in the regular price-time priority queue, maintained as a separate midpoint matching layer.

MMP Threshold: The fill count and time window that together trigger Market Maker Protection. For

example, "5 fills within 1 second." If this threshold is exceeded, the exchange automatically cancels all of the market maker's resting quotes. The threshold is negotiated per market maker, per symbol, and finding the right value is a key parameter of market making strategy.

Monotonic: Strictly non-decreasing. A monotonic sequence is one where each value is equal to or greater than all previous values. A strictly monotonic sequence is one where each value is strictly greater.

N

Naked Short Selling: Selling shares short without first locating and arranging to borrow them. The seller delivers nothing at settlement, creating a "fail to deliver." Illegal in most jurisdictions under short sale regulations (Regulation SHO in the US). Distinguished from covered short selling, where shares have been located and borrowed.

Nanosecond: One billionth of a second (10 seconds). Modern matching engines record timestamps at nanosecond precision.

NBBO (National Best Bid and Offer): The best available bid and ask prices aggregated across all US exchanges, as required by Regulation NMS. Brokers must offer clients execution at or better than the NBBO.

NAV (Net Asset Value): The total value of a fund's holdings divided by the number of shares outstanding. For mutual funds, NAV is computed daily using the official **closing price** of each holding, one reason the closing price matters so much.

Novation: The legal process by which a CCP replaces the original buyer-seller relationship: the CCP becomes buyer to the seller and seller to the buyer, eliminating bilateral counterparty risk.

NTP (Network Time Protocol): A protocol for synchronising computer clocks over a network. Basic NTP achieves millisecond accuracy; higher-precision variants achieve sub-millisecond.

O

OCC (Options Clearing Corporation): The central clearing house for equity options in the US.

Odd Lot: An order or execution for a quantity that is not a whole multiple of the standard round lot size for that instrument (commonly 100 shares for US equities). Historically excluded from consolidated NBBO and tape data in US equities, a practice being revised under the SEC's market-data-infrastructure rules. See *Round Lot*.

OCO (One-Cancels-Other): Two linked orders where filling either automatically cancels the other.

OHLCV: Open, High, Low, Close, Volume, the five standard summary statistics recorded for each trading session (or any time period). Open: the first traded price. High: the highest traded price. Low: the lowest traded price. Close: the last traded price (the official closing price in a closing auction). Volume: total shares traded. OHLCV data forms the basis of almost all price charts and historical market analysis.

Open Outcry: The historical method of trading on exchange floors, where traders shouted bids and asks and used hand signals in a sunken circular area called a pit. Largely replaced by electronic trading.

Options: Contracts that give the buyer the right, but not the obligation, to buy (a **call option**) or sell (a **put option**) an underlying asset at a specified price (the **strike price**) before or on a specified date (the **expiry date**). The seller of an option takes on the corresponding obligation. Options are priced using models (Black-Scholes being the most famous) that account for time to expiry, volatility, interest rates, and the distance between the current price and the strike. Cboe is the world's largest options exchange.

Order: An instruction to buy or sell a quantity of an instrument at a specified price or better.

Order Book: The central data structure of a matching engine, all resting buy and sell orders for a symbol, organised by price.

Order ID: A unique identifier assigned to each order upon submission. Must be unique across the entire system.

OTC (Over-the-Counter): Trading that occurs directly between two parties without going through an

exchange. OTC trades are less transparent and subject to different regulations than exchange-traded transactions.

OUCH (NASDAQ OUCH): NASDAQ's high-performance binary order entry protocol used by member firms and low-latency trading systems to submit, modify, cancel, and manage orders directly at the matching engine. OUCH is session-based and message-oriented, with exchange acknowledgments and execution/cancel responses that let clients track the full order lifecycle deterministically. Official specification: NASDAQ OUCH 5.0.

P

P&L (Profit and Loss): The financial gain or loss on a trading position.

Passive Order: See *Resting Order*

Payment for Order Flow (PFOF): A practice in which retail brokers receive payment from wholesale market makers in exchange for routing client order flow to those market makers for internalisation rather than to lit exchanges. Market makers pay for retail flow because retail orders have lower adverse selection risk. Effectively banned in the UK since 2012, and banned EU-wide from 30 June 2026 under the 2024 MiFIR review. Legal but controversial in the US, where it subsidises zero-commission brokerages but raises concerns about conflicts of interest and reduced price discovery. See *Best Execution*, *Internaliser*.

Pin Risk: The risk faced by an option writer when the underlying closes at expiry very close to the strike price, leaving the writer uncertain whether automatic exercise and assignment will occur until after markets have closed, with any resulting delivery obligation priced at the strike regardless of where the underlying opens the next session. See *Exercise*, *Assignment*.

Pit: The sunken, octagonal or circular physical trading area on an exchange floor where open outcry trading happened. CME Group operated prominent futures pits in Chicago. Largely dormant today.

Portfolio: A collection of investments held by an investor, typically diversified across asset types (equities, bonds, etc.) to manage risk and return.

Position: A participant's current holding in an instrument, positive (long) if they own more than they have committed to deliver; negative (short) if they have committed to deliver more than they own; zero (flat) if both sides balance. Position tracking is the clearing system's primary responsibility.

Pre-Trade Risk Controls: Checks applied to orders before they reach the matching engine, quantity limits, notional limits, fat-finger filters, credit checks, and rate limiting. The gateway is typically the enforcement layer.

PRE_OPEN: Session phase before the opening auction. Orders are accepted and rest on the book, but no matching occurs.

Presence Obligation: A market maker's contractual requirement to have live two-sided quotes resting in the book for a minimum percentage of the trading session, commonly 85% or more. Prevents market makers from only quoting on easy days and disappearing on volatile ones when liquidity is most needed.

Price Discovery: The process by which the market determines the current fair price of an asset through the interaction of buyers and sellers. A well-functioning exchange enables efficient price discovery: prices reflect all currently available information and update rapidly as new information arrives. Price discovery is one of the three core promises every exchange makes. Poor price discovery, where the traded price diverges significantly from fair value for extended periods, is a sign of market dysfunction.

Price Level: All resting orders at the same price on the same side of the book.

Price-Time Priority: The matching rule: better prices fill first; at the same price, earlier arrivals fill first.

Primary Market: Where new securities are issued and sold for the first time. The company (or government) receives the proceeds.

Primary Site: The active, live data centre running the production matching engine.

Private Company: A company whose shares are not publicly traded, held only by founders, employees, and private investors.

Prospectus: The disclosure document a company must publish before offering securities to the public. In the EU it is governed by the Prospectus Regulation (EU) 2017/1129 and approved by a national competent authority; in the US it forms part of the S-1 registration statement.

Pro-Rata Allocation: An alternative to price-time priority where fills at a price level are distributed proportionally among all resting orders at that level. Common in some futures markets.

Protection Period: The window of time after MMP fires during which a market maker has no active quotes and is expected to assess market conditions before deciding whether and at what prices to re-enter. The duration is typically a few seconds and is defined in the market maker agreement.

PTP (Precision Time Protocol, IEEE 1588): A network clock synchronisation protocol achieving sub-microsecond accuracy. Used in exchange infrastructure to ensure consistent timestamps across distributed systems.

Public Company: A company whose shares are listed on a stock exchange and available to any investor.

Q

Quote: A two-sided order submitted by a market maker containing both a bid (buy) and an ask (sell) simultaneously. Unlike a regular order, the two legs are linked, what happens to one (a fill, a cancellation) triggers a defined response on the other according to the quote refresh policy.

Quote Leg: One side (bid or ask) of a two-sided market maker quote. Filling one leg triggers the quote refresh policy for the other leg.

Quote Refresh Policy: The rule governing what happens to the surviving leg of a two-sided quote when the other leg is filled. Three main variants: (1) cancel both sides immediately, the most conservative, used by Eurex-style inactivation; (2) cancel only the filled side, leave the other active; (3) leave both active. The choice reflects the market maker's balance between speed of re-entry and risk of holding stale inventory.

Quote Stuffing: A market manipulation practice where a participant rapidly submits and cancels a very large number of orders to consume exchange bandwidth, slow down competitors' systems, and create artificial confusion in market data. A form of denial-of-service abuse in electronic markets.

R

Ratchet: The mechanism in a trailing stop that advances the stop price in the favourable direction but freezes it when the market reverses.

Rate Limiting / Throttling: A control that limits the number of orders a participant can submit per unit time. Prevents denial-of-service conditions from algorithmic misfires.

Re-quoting Obligation: A market maker's contractual requirement to post a fresh two-sided quote within a specified maximum delay after either leg of their previous quote is filled or after MMP fires. Failure to re-quote within this window counts against the presence obligation and may trigger penalties. The window is typically measured in milliseconds to low seconds.

Realized P&L: Profit or loss that has been "locked in" through a trade that reduced an open position. Once realized, it cannot be reversed by subsequent price moves.

Reference Data / Instrument Master: The configuration data describing a tradeable instrument, tick size, price scale, session schedule, contract multiplier, trading status, expiry date, position limits. Read by every exchange component; an error propagates system-wide. Treat it with the same rigour as code: version-controlled, tested, applied atomically.

Regulation NMS: A US Securities and Exchange Commission regulation requiring that equity orders be executed at the best available price across all exchanges (the NBBO). A primary driver of US market fragmentation.

Regulation SHO: A US Securities and Exchange Commission rule governing short sales in equity markets. Key requirements: broker-dealers must have a reasonable belief shares can be borrowed before accepting a short sale order (the "locate" requirement); persistent fail-to-deliver situations must be closed out; short sales

must be marked as "short" in order records. Adopted in 2005 to address concerns about abusive naked short selling.

Replay (Exchange): Replaying a recorded sequence of input messages through the matching engine to reconstruct past market states. Requires the engine to be deterministic.

Reserve Refresh Priority: The queue position rule for iceberg replenishment. Most exchanges place a newly replenished iceberg peak at the back of the queue at its price level, equivalent to a brand-new arrival, rather than preserving the original queue position.

Resting Order / Passive Order: An order that has been accepted by the exchange and is waiting in the book for a counterparty.

Retained Earnings: Profits kept within the business rather than distributed to shareholders. A source of self-funding for company growth.

Rolling Window (MMP): The time period over which fills are counted for Market Maker Protection purposes. Fills older than the window are discarded from the count; only fills within the current window matter. For example, a 1-second rolling window means: "if more than N fills have arrived in any sliding 1-second period, MMP fires."

Round Lot: The standard trading unit for an instrument (commonly 100 shares for US equities), used as the basis for distinguishing a normal order from an *Odd Lot*.

Rule 612 (Sub-Penny Rule): The Regulation NMS provision, adopted 2005, prohibiting exchanges, ECNs, and market makers from displaying or accepting quotations in NMS stocks priced at $1.00 or more in increments finer than $0.01. Stocks priced below $1.00 may be quoted in sub-penny increments. See *Tick Sizes and Fractional Ticks*, Part II.

S

Secondary Market: Where existing securities are traded between investors. The issuing company does not receive proceeds. Stock exchanges are secondary markets.

Secondary Site / Backup Site: A standby data centre that can take over exchange operation if the primary site fails.

Securities Lending / Stock Borrow: The temporary transfer of securities from a lender (typically a long-term investor or custodian) to a borrower (typically a short seller) in exchange for collateral and a lending fee (borrow rate). Short sellers must borrow shares before selling short. The lender retains economic ownership (receives manufactured dividends and can recall the shares) but transfers legal title temporarily. Recall risk, the lender demanding shares back at an inconvenient time, is a key risk for short sellers.

S-1 (Form S-1): The registration statement a US company files with the SEC under the Securities Act of 1933 before an IPO. It contains the prospectus, audited financials, risk factors and the *use of proceeds*. Foreign issuers use the equivalent Form F-1.

SEDOL: A 7-character UK security identifier used in settlement and clearing. Analogous to CUSIP.

Self-Match Prevention (SMP): A mechanism that detects when a participant would trade against their own orders and applies a cancellation policy.

Sequence Number: A monotonically increasing counter attached to events (in a drop copy or message feed) to enable detection of missed events and ordered replay.

Session State Machine: The formal model of an exchange's trading day phases (PRE_OPEN, OPENING_AUCTION, CONTINUOUS, CLOSING_AUCTION, CLOSED) and the permitted transitions between them.

Securities Information Processor (SIP): The consolidated, publicly distributed feed aggregating trade and quote data from every registered US exchange into a single tape. Contrast with a *proprietary direct feed*, which an individual exchange sells separately, typically with lower latency and richer content than the SIP. See *Market Data Economics*, Part IV.

Settlement: The final transfer of securities from seller to buyer and cash from buyer to seller. Legally

completes the trade. In the US, equity settlement currently occurs on T+1.

Settlement Method (Derivatives): Whether a futures or options contract settles by cash payment or physical delivery of the underlying asset. Determines post-expiry processing required by the exchange and clearing house.

Shadow Replication / Shadow Mode: An architectural pattern for high-availability exchange systems. The secondary site receives and processes all input messages in parallel with the primary, maintaining an up-to-date copy of the order book, but does not publish outputs or accept participant connections. If the primary fails, the secondary's in-memory state is already current and it can begin accepting inputs and publishing outputs with minimal failover time. More common in production than true active-active matching.

Share / Stock: A single unit of equity ownership in a company. Each share represents a fractional claim on the company's assets, earnings, and votes.

Slippage: The difference between the expected fill price and the actual average fill price, caused by executing against multiple price levels in the book. Large orders in thin books experience more slippage.

Smart Order Router (SOR): Software that evaluates multiple trading venues and routes orders to achieve the best overall execution, balancing price, fees, available depth, and speed.

SmartNIC / Ultra-Low-Latency NIC: A specialised network interface card designed for trading and other latency-sensitive applications. Unlike commodity Ethernet cards, SmartNICs include an on-card kernel bypass networking stack (such as Solarflare's OpenOnload or Mellanox's RDMA), hardware timestamping of incoming packets to nanosecond precision, and CPU offload for networking tasks. Examples include the AMD Solarflare SFN series and NVIDIA Mellanox ConnectX series, widely deployed in exchange co-location environments.

Snapshot (Persistence): A complete dump of the matching engine's state (all resting orders, positions, key data) taken periodically to reduce recovery time by limiting the amount of write-ahead log that must be replayed.

SPAC (Special Purpose Acquisition Company): A shell company with no operating business that completes its own IPO to raise cash held in trust, with the purpose of later merging with a private operating company to take it public without a traditional IPO process. See *De-SPAC*.

Specialist: NYSE's historical term for the designated firm responsible for maintaining fair and orderly markets in a specific stock, including maintaining the physical order book. The modern equivalent is the Designated Market Maker (DMM).

Speed Bump: A deliberate artificial delay applied to all incoming orders to eliminate the advantage of being marginally faster than other participants. Pioneered by IEX.

Split-Brain: A failure mode in distributed systems where two nodes both believe themselves to be the active primary, leading to conflicting actions.

Spoofing: A market manipulation practice: placing orders with no intention of trading to move prices, then cancelling before execution. Illegal under market abuse regulations.

Spread: The difference between the best ask and the best bid. Market makers earn the spread; participants pay it as the cost of immediate trading.

State Machine: A formal model consisting of a finite set of states and explicit rules about which transitions between states are allowed.

Static Price Collar (Fat-Finger Filter): A pre-trade risk control that rejects any order whose submitted price deviates more than a configured percentage from the last official close price. Designed to catch obvious fat-finger errors, an order at $15.00 on a stock trading at $150.00 is almost certainly a decimal error. Because the reference is the previous close, the static collar is stable throughout the trading day. Contrast with *Dynamic Price Collar*, which tracks the most recent trade instead.

Stop Order: A conditional order that is dormant until the market price reaches the stop price, then converts to another order type.

Stop Price: The trigger price on a stop or stop-limit order. The order remains dormant until the market's last trade price reaches or crosses the stop price, at which point it activates and enters the book (as a market

or limit order, depending on type). A buy stop has a stop price above the current market; a sell stop has a stop price below it.

Stop-Limit Order: A stop order that converts to a limit order when triggered. Provides price protection but may not execute if the market gaps.

Stop-Loss: A sell stop order used to automatically exit a losing long position if the price falls to a specified level.

Surplus: The absolute difference between cumulative buy and sell quantity at the equilibrium price, the quantity that cannot be matched.

Sweeping: The process of an aggressive order filling against resting orders at successive price levels.

T

T+1 / T+2 Settlement: Settlement occurring 1 or 2 business days after the trade date. The US moved from T+3 to T+2 in 2017 and to T+1 in 2024.

T+2 Settlement: Settlement occurring two business days after the trade date. The US standard until 2024; most US equities now settle T+1. T+2 remains the standard in many other markets.

Tick: The minimum price movement for an instrument.

Tick Count / Tick-Based Price: A price represented as an integer number of minimum increments, rather than a decimal. Tick counts are exact; floating-point decimals are not.

Tick Size: The value of one tick, the minimum price increment for an instrument.

Tick Size Pilot Program: A US SEC study (October 2016 – October 2018) that widened the tick size to five cents for roughly 1,200 small-capitalisation stocks to test whether coarser ticks improve market-making economics and liquidity for thinly traded names. Results were mixed: quoted spreads and costs for small trades rose, displayed size at the best prices rose sharply, but the hoped-for improvement in overall liquidity and capital formation did not appear, and no permanent wider-tick regime was adopted. See *Tick Sizes and Fractional Ticks*, Part II.

Tick-to-Trade Latency: The time from when a market data message leaves the exchange to when an order based on that message is received by the exchange. A key performance metric for market makers and HFT firms.

Ticker / Ticker Symbol: The abbreviated code identifying a tradeable instrument (e.g., AAPL for Apple Inc.). Named after the mechanical stock ticker machine that printed trade prices on paper tape using telegraph signals in the late 19th and early 20th centuries.

Ticker Tape: The narrow paper strip printed by the stock ticker machine, showing a continuous stream of trade symbols, prices, and volumes. The source of the term "tick" (minimum price movement) and related concepts.

Time-In-Force (TIF): The attribute specifying how long an order remains valid (DAY, GTC, IOC, FOK, ATO, ATC).

Top-of-Book (Level 1): Market data showing only the best bid price, best ask price, and their quantities. The minimum information needed to understand the current market.

Total Addressable Market (TAM): The total revenue available if a company captured 100% of its market. Often narrowed to the serviceable addressable market (SAM) and the share realistically obtainable (SOM). Popular in growth-company IPO pitches, and notorious for being stretched.

Trade: An executed match between a buy and sell order. Also called a fill or execution.

Trailing Stop: A stop order whose trigger price automatically advances in the favourable direction as the market moves, protected by a fixed trail offset.

TWAP (Time-Weighted Average Price Algorithm): An execution algorithm that distributes a large order evenly across equal time intervals, regardless of trading volume. Simple and predictable, but less adaptive than VWAP. Used when a stock has no reliable volume profile or when the trader wants mechanically

consistent participation. Performance is measured against the average of prices during the execution window, not the overall day's VWAP.

Two-Sided Quote: A market maker's simultaneous bid and ask for the same instrument. Both sides must be live at the same time to fulfil the two-sided quoting obligation. A one-sided quote (only bid or only ask) is a contractual breach of market maker obligations.

U

UUID (Universally Unique Identifier): A 128-bit identifier designed to be unique without central coordination. UUID v4 is randomly generated; UUID v1 incorporates the current time and network address.

Underpricing: Setting an IPO offer price below what the market will pay, visible as a first-day price jump. The gap multiplied by the number of shares sold is called money *left on the table*.

Uncross The single atomic batch execution that occurs at the end of an auction phase. All crossable orders execute at the **equilibrium price** simultaneously.

Unrealized P&L The theoretical profit or loss on an open position if it were closed at the current market price. Changes continuously as the market moves. Becomes realized P&L when the position is closed.

V

Variation Margin: Daily cash payments reflecting mark-to-market gains and losses on open positions. Prevents losses from accumulating to unmanageable levels before settlement.

Venture Capital (VC): Private investment firms that provide funding to early-stage companies in exchange for equity stakes, before those companies are large enough to go public.

VIX (Volatility Index): A real-time index calculated by Cboe from the prices of S&P 500 options, representing the market's expectation of 30-day price volatility in the S&P 500. Commonly called the "fear gauge", when markets are uncertain or falling sharply, participants buy put options for protection, driving up option prices and therefore the VIX. Values above 30 typically indicate high market stress; values below 20 indicate relative calm. Derivatives on the VIX itself trade on Cboe.

VWAP (Volume-Weighted Average Price): The average price of a position weighted by the quantity traded at each price. Used as cost basis for P&L calculation. VWAP $= \frac{\sum(\text{price}_i \times \text{qty}_i)}{\sum \text{qty}_i}$\$

VWAP Algorithm: An execution algorithm that slices a large order and executes it throughout the day in proportion to expected trading volume, more during high-volume periods (open, close) and less during low-volume periods (mid-day). The goal is to achieve an average execution price close to or better than the day's volume-weighted average price. The most widely used institutional execution benchmark. Distinct from *VWAP (Volume-Weighted Average Price)*, which is a measurement; the VWAP algorithm is the execution strategy aimed at matching that measurement.

W

Wash Trading / Wash Trade: A transaction where the same participant is on both sides, generating artificial trading volume with no real change of ownership. Illegal in all major jurisdictions under market manipulation regulations (the EU's Market Abuse Regulation, the US SEC's Rule 10b-5, and others). Also called a wash sale. Sometimes performed accidentally, see *Self-Match Prevention (SMP)*, but when deliberate, it is a form of market abuse used to inflate trading volume figures or create false impressions of liquidity.

Write-Ahead Log (WAL): A persistent, append-only log of all input messages to the matching engine. Written before processing; used for recovery and replay. The source of truth from which all derived state (the in-memory order book) can be reconstructed.

Z

ZeroMQ (ZMQ): A messaging library providing efficient pub/sub, push/pull, and other communication patterns. Commonly used as the message bus in exchange systems. Uses a broker-less PUSH/PULL and PUB/SUB topology. No external message queue server is required.

References

The following sources were drawn upon in writing this document.

[1] Harris, Larry. *Trading and Exchanges: Market Microstructure for Practitioners.* Oxford University Press, 2003. ISBN 978-0-19-514470-3.

The standard academic and practitioner reference for exchange mechanics and market microstructure. Covers order types, price-time priority, the order book, spreads, market makers, adverse selection, and much of the vocabulary in this document. Recommended as the primary companion text for anyone building exchange software.

[2] Hasbrouck, Joel. *Empirical Market Microstructure: The Institutions, Economics, and Econometrics of Securities Trading.* Oxford University Press, 2007. ISBN 978-0-19-530164-3.

A rigorous academic treatment of the order book and matching process. Covers the econometrics of market data, information in order flow, and the theory underlying price-time priority. More mathematical than [1] but an authoritative source on how order books actually behave in practice.

[3] Brealey, Richard A., Stewart C. Myers, and Franklin Allen. *Principles of Corporate Finance.* 13th edition. McGraw-Hill Education, 2020. ISBN 978-1-260-01390-0.

The leading corporate finance textbook, used in business schools worldwide. Covers equity, bonds, dividends, IPOs, cost of capital, and the relationship between primary and secondary markets. The source for concepts in Part I of this document (equity, bonds, how companies raise capital).

[4] De la Vega, Joseph. *Confusión de Confusiones* (written 1688). Translated by Hermann Kellenbenz. Harvard University Graduate School of Business Administration, Kress Library of Business and Economics, 1957.

The oldest known book about stock market trading, written in Amsterdam in 1688 by a Spanish merchant named Joseph de la Vega. Describes, in dialogues, the practices of the Amsterdam VOC share market, including resting orders, market making, and speculative trading. A remarkable primary source demonstrating that many modern concepts are centuries old. The Kellenbenz translation includes a scholarly introduction providing historical context.

[5] Petram, Lodewijk. *The World's First Stock Exchange.* Columbia Business School Publishing, 2014. ISBN 978-0-231-16714-9. (Originally published in Dutch as *De wereld's eerste beurs.* Uitgeverij Atlas Contact, 2011.)

The definitive modern history of the Amsterdam Exchange and the VOC share market. Covers the founding of the VOC (1602), the development of the first secondary market in shares, Isaac Le Maire's 1609 short selling campaign against VOC stock, the 1610 Amsterdam edict attempting to ban short selling, and the broader financial innovations that originated in 17th-century Amsterdam. Directly cited for the Le Maire material in this document.

[6] Lewis, Michael. *Flash Boys: A Wall Street Revolt.* W.W. Norton & Company, 2014. ISBN 978-0-393-24413-3.

A narrative account of the founding of IEX by Brad Katsuyama and colleagues, describing the practice of high-frequency trading, its effects on institutional investors, and the design of the speed bump as a countermeasure.

The primary source for the IEX material in the *Speed Bumps* section of Part IV. Accessible to non-specialist readers; recommended as background reading for anyone working on exchange latency and fairness.

[**7**] U.S. Securities and Exchange Commission and Commodity Futures Trading Commission. *Findings Regarding the Market Events of May 6, 2010.* Joint Report. 30 September 2010.

The official regulatory investigation into the 2010 Flash Crash, which saw the Dow Jones Industrial Average fall approximately 1,000 points in minutes on May 6, 2010. Analyses the sequence of events, the role of automated trading, and the breakdown of circuit breakers.

- **Available at:** https://www.sec.gov/news/studies/2010/marketevents-report.pdf

[**8**] U.S. Securities and Exchange Commission. *SEC Adopts T+2 Settlement Cycle for Securities Transactions.* Press Release No. 2017-68. 22 March 2017.

The formal SEC announcement of the rule change shortening US equity settlement from T+3 to T+2, effective September 5, 2017. Contains the rationale for the change and the text of the amended rules.

- **Available at:** https://www.sec.gov/news/press-release/2017-68

[**9**] CME Group. *CME Group to Close Open Outcry Trading in Most Futures Pits.* Press Release. 4 February 2015. Available at: https://www.cmegroup.com

The formal announcement that CME Group would cease open outcry operations in most of its Chicago futures pits, transitioning those products to fully electronic trading. Marks the practical end of an era of physical floor trading that had defined futures markets for over a century.

[**10**] Michie, Ranald C. *The Global Securities Market: A History.* Oxford University Press, 2006. ISBN 978-0-19-928065-3.

A comprehensive scholarly history of securities markets worldwide from the 17th century to the present. Covers the origins of the NYSE and NASDAQ, the development of settlement practices, short selling regulation, and the transition from physical to electronic trading. Cited for historical facts about exchange foundations and the evolution of market structure.

[**11**] Goetzmann, William N., and K. Geert Rouwenhorst, eds. *The Origins of Value: The Financial Innovations that Created Modern Capital Markets.* Oxford University Press, 2005. ISBN 978-0-19-517571-4.

An edited volume of scholarly essays examining the historical development of financial instruments and markets. Includes detailed coverage of the Dutch East India Company, the Amsterdam Exchange, and the origin of tradeable shares, futures, and options. The chapter by Gelderblom and Jonker, "Amsterdam as the Cradle of Modern Futures and Options Trading, 1550–1650," is particularly relevant to the Amsterdam material in the *Language of the Market* section of Part I.

[**12**] Natenberg, S. *Option Volatility and Pricing.* McGraw-Hill, 2nd edition, 1994. ISBN 978-1-55738-486-7.

The practitioner standard. Used as the internal training text at options market making firms for decades. Covers every significant multi-leg strategy, spreads, straddles, strangles, ratio spreads, butterflies, condors, with worked examples from first principles. It explains synthetic positions in depth: synthetic long stock (long call + short put at the same strike), synthetic short stock (short call + long put), synthetic calls and puts, and how put-call parity makes them equivalent. It explains why these constructions exist, delta hedging, volatility trading, arbitrage, not just what they are. The section on volatility spreads directly connects to why exchanges need to support combo orders natively.

[**13**] Hull, John C. *Options, Futures, and Other Derivatives.* Pearson/Prentice Hall, (13th edition, 2021). ISBN 978-0-13-813423-3.

The authoritative academic and practitioner reference for all derivatives. Chapter 12 covers trading strategies involving options in detail, spreads (bull, bear, box, butterfly), combinations (straddles, strangles, strips, straps), with numerical examples and payoff diagrams. The chapters on futures also cover calendar spreads and the basis. Hull grounds everything in put-call parity and arbitrage relationships, which is what makes synthetics work. More mathematical than Natenberg but more complete.

[**14**] *Oliver Gingold and "blue chip" (1923-1924):* The attribution of the first financial use of "blue chip stocks" to Dow Jones employee *Oliver Gingold* circa 1923-1924 is widely repeated in financial dictionaries (including Merriam-Webster). The reference used here comes from **Philadelphia University**, Blue Chip term. Note

the $200/share figure attached to Gingold's original usage is a nominal 1923 dollar amount; it has no bearing on "blue chip" classification today and should not be read as a price threshold (see the discussion in *Before the Exchange*, Part I).

[15] *Edward Calahan and the stock ticker (1867):* Widely documented in the history of telegraphy and American industry. The invention is discussed in Calahan's own patents of the period U.S. Patent No 76,157, Edward A. Calahan,

[16] *Mizuho Securities fat-finger incident (2005):* The December 2005 incident in which a Mizuho trader submitted an order to sell 610,000 shares at ¥1 rather than 1 share at ¥610,000 was extensively reported in media, e.g. FinExtra - Mizuho sues TSE over 'fat finger' trade botch-up, Guardian. It has become a standard illustrative example in financial risk management literature.

- Available at Guardian Newspaper: https://www.theguardian.com/business/2005/dec/09/japan.internationalnews1
- Available at FinExtra: https://www.finextra.com/newsarticle/16081/mizuho-sues-tse-over-fat-finger-trade-botch-up

[17] Lovelock, Mendel, Wright. *An Introduction to the Mathematics of Money.* Springer Verlag, 2007. ISBN 978-0387-34432-4.

A personal favourite, because it quietly dismantles concepts you were certain you understood. Take interest: you think you know what it is? Probably not. At the very least you will come away knowing how the *effective interest rate* on a loan ought to be calculated, and why an effective rate of a few million percent is rather less alarming than it sounds. In the EU this figure must, by law, be quoted on every instalment-based payment, which has inspired businesses to develop a remarkable repertoire of entirely legal "tricks" for arriving at the lowest quotable number.

[18] U.S. Securities and Exchange Commission. *Decimalization of Equity Markets.* Phased implementation order, effective through April 2001. See also Bacidore, Battalio, and Jennings, "Order Submission Strategies, Liquidity Supply, and Trading in Pennies on the New York Stock Exchange," and related Rule 612 (sub-penny) rulemaking under Regulation NMS, 2005.

The formal record of the switch from fractional to decimal quoting in US equities. The source for the decimalization history and Rule 612 material in the *Tick Sizes and Fractional Ticks* section of Part II.

[19] U.S. Securities and Exchange Commission. *Tick Size Pilot Program*, adopted under Rule 612, effective October 2016 to October 2018. Final assessment reports prepared for the SEC by academic and industry researchers.

The primary source for the Tick Size Pilot Program discussion in the *Tick Sizes and Fractional Ticks* section of Part II, including the pilot's design (control group plus three five-cent test groups across roughly 1,200 small-cap securities) and its mixed empirical findings.

- Available at: https://www.finra.org/rules-guidance/key-topics/tick-size-pilot-program
- Assessment report available at: https://www.sec.gov/files/TICK%20PILOT%20ASSESSMENT%20FINAL%20Aug
- Submission of Tick Size Pilot Plan: https://www.govinfo.gov/content/pkg/FR-2014-06-30/pdf/2014-15205.pdf
- SEC Landing Page: https://www.sec.gov/data-research/tick-size-pilot-program/tick-size-pilot-data-resources
- SEC Approval: https://www.sec.gov/files/rules/sro/nms/2015/34-74892-exa.pdf

[20] U.S. House of Representatives, Committee on Financial Services. *Game Stopped: Who Wins and Loses When Short Sellers, Social Media, and Retail Investors Collide.* Majority Staff Report, June 2021.

The primary congressional record of the January 2021 GameStop episode, including the NSCC margin call to Robinhood and other brokers and the resulting trading restrictions. Cited in the *Clearing and Settlement* section of Part III.

- Available at: https://democrats-financialservices.house.gov/uploadedfiles/6.22_hfsc_gs.report_hmsmeetbp.irm.nlr

[21] U.S. Securities and Exchange Commission and U.S. Commodity Futures Trading Commission, civil complaints against Samuel Bankman-Fried and FTX Trading Ltd., December 2022; U.S. Department of Justice, *United States v. Samuel Bankman-Fried*, S.D.N.Y., 2023 conviction.

The primary legal record of the FTX collapse, cited in the *Cryptocurrency and Digital Asset Venues* section of Part IV as an illustration of counterparty risk when matching, custody, and clearing are not institutionally

separated.

- **Wikipedia:** https://en.wikipedia.org/wiki/Trial_of_Sam_Bankman-Fried
- **SEC:** https://www.sec.gov/news/press-release/2022-219
- **CFTC:** https://www.cftc.gov/PressRoom/PressReleases/8638-22

- **DOJ:** https://www.justice.gov/archives/opa/pr/samuel-bankman-fried-sentenced-25-years-his-orchestration-multiple-fraudulent-schemes

Exchanges own technical references

Publicly available technical specifications and documentation provide exact structural logic and algorithmic details for combo and implied/synthetic orders across major electronic matching engines.

CME Globex Matching Engine & Implied Logic

CME Group provides open-access engineering wikis that map out their continuous multi-leg matching models.

- **Implied Orders Functionality:** To read the explicit state constraints, calculation boundaries, and engine generation rules for "Implied-In" and "Implied-Out" routing structures, visit the CME Globex Implied Orders Documentation.
- **Options-Specific Implied Generation:** For data regarding RFQ-triggered timers and second-generation restriction models, see the CME Globex Implied Options Documentation.
- **Algorithmic Match Priorities:** To inspect how implied quantities interact with Pro Rata or FIFO allocation passes when competing with outright resting orders, review the CME Globex Matching Algorithms Reference.
- **Implied Orders:** CME Group, *Client Systems Wiki, Implied Orders* (overview of implied IN / implied OUT functionality, per-product enablement, and implied dissemination on MDP3): https://cmegroupclientsite.atlass (search: "Implied Orders")

Cboe US Options Complex Book Architecture

Cboe details its execution logic for multi-leg derivative allocations, net premium sorting, and atomicity verification.

- **Technical Specifications Library:** Access the complete engineering documentation library, covering binary order entry and auction layout parameters, via the Cboe U.S. Options Technical Specifications.
- **Complex Book Fundamentals:** For an structural layout of multi-leg order handling constraints and leg ratio normalization, review the Cboe Titanium Complex Order Basics.
- **Complex Book Process Guide:** To dive into the complete structural manual—detailing Signed Values on the Complex Book (debit vs. credit), order sorting mechanisms, and Complex Order Auctions (COA)—view the comprehensive Cboe Titanium Complex Book Process Document.

Intercontinental Exchange (ICE) Order Books

ICE manages highly nested energy and financial spreads that utilize synthetic order generation.

- **Implied Prices and Generations:** ICE's FAQ on implied prices, including the Sep/Oct/Nov multi-generation example used in Part II: ICE Futures Implied Prices.
- **Market Data & Connectivity Controls:** For foundational parameters on data fields, systemic electronic metrics, and electronic volume calculations within the ICE infrastructure, refer to the Intercontinental Exchange Connect User Guide.

NYSE Pillar (Equity Trading Platform)

The core documentation for understanding order behavior, priority categories, and amendment rules across the NYSE (New York Stock Exchange) platform

- Pillar is NYSE technology platform which enables members to access NYSE equities and options markets using standard protocols, NYSE Pillar
- The structural and architectural details are broken down in the official NYSE Pillar Functional Differences Guide. This resource outlines how market collars, auction imbalances, and continuous trading interactions operate inside the core matching engine.
- Order types Matrix (XLSX) supported by NYSE

Further Sources Cited in the Text

Implied orders and spreads

- CME Group Client Systems Wiki, Futures Implied Order Matching Priority (real orders before implied orders; tie-breaks between implied sources).
- CME Group Client Systems Wiki, MDP 3.0 – Implied Book (two-deep implied book dissemination).
- CME Group, Treasury Futures Calendar Spreads, 2016 (spread buy/sell convention; leg-price assignment).
- Isaac Carruthers, Inter-Commodity Spreads and Implied Pricing, 2018 (outward rounding of implied prices; finer implied grids for ratio spreads).
- Eurex, Matching principles (synthetic matching) and Passive Liquidity Protection.
- CME press releases on implied functionality for Eurodollar butterflies, 8 March 2004 and 18 October 2004.

Crude oil and negative prices

- CME Group, NYMEX Rulebook Chapter 200: Light Sweet Crude Oil Futures (1,000 barrels; $0.01 tick).
- CFTC, Interim Staff Report on NYMEX WTI Crude Oil Futures Trading around April 20, 2020, November 2020.
- Congressional Research Service, Crude Oil Futures Prices Turn Negative (IN11354), 2020.
- CFTC press release 8432-21, 28 September 2021 (Interactive Brokers: negative-price handling failures).

Market events, enforcement and statistics

- SEC and CFTC, Findings Regarding the Market Events of May 6, 2010, September 2010.
- SEC, In the Matter of Knight Capital Americas LLC, Release No. 34-70694, 16 October 2013.
- SEC Division of Economic and Risk Analysis, Assessment of the Plan to Implement a Tick Size Pilot Program, 2018.
- FINRA, Rule 11892: Clearly Erroneous Transactions in Exchange-Listed Securities.
- NYSE, Closing Auction Fact Sheet; BMLL, Into the Close: Unpacking U.S. Closing Auction Dynamics, 2024.
- Federal Reserve Bank of St. Louis, FRED series SP500 (S&P 500 closing levels, March 2020).
- CME Group, Record annual ADV of 28.1 million contracts in 2025, 5 January 2026.
- Databento, CME matching algorithms explained.
- Eurex, T7 Functional and Interface Overview (Market Maker Protection limits).
- NYSE, Enhancement to Reopening Process after LULD Trading Pauses, 2017 (auction collar widening in 5% steps).
- SEC, Order approving Nasdaq Halt Cross price protections, Release No. 34-102336, 10 February 2025.

- Deutsche Börse, Protective Mechanisms in Auctions (Xetra volatility interruptions); CME Group,
- Deutsche Börse, Price Limits.
- Deutsche Börse, Market Model for the Trading Venue Deutsche Börse Xetra
- Wikipedia, Initial public offering of SpaceX; CoinDesk, SpaceX prices shares at $135 in largest IPO ever, 11 June 2026.
- CME Group, CME Group to Permanently Close Most Open Outcry Trading Pits, 4 May 2021.
- SEC, In the Matter of The NASDAQ Stock Market, LLC (Facebook IPO), Release No. 34-69655, 29 May 2013.
- CCP Global, The Lehman Case (LCH SwapClear close-out of Lehman's portfolio).

IPO pricing and valuation

- Jay R. Ritter, IPO Data, University of Florida (first-day returns, money left on the table, gross spreads).
- Aswath Damodaran, interview with The Acquirer's Multiple, 2024 (pricing versus valuing).
- Investor.gov (SEC), Lockup Agreements.
- SEC, Regulation S-K Item 504: Use of Proceeds.
- European Union, Prospectus Regulation (EU) 2017/1129 and Listing Act Regulation (EU) 2024/2809; Bird & Bird, 2025 (Listing Act overview).
- S&P Dow Jones Indices, S&P U.S. Indices Methodology (IPO seasoning, profitability rule); S&P DJI announcement, 4 June 2026.
- WeWork Companies Inc., Form S-1 registration statement, 14 August 2019.
- News coverage used for deal details: *Forbes* (Facebook, 17 May 2012; WeWork, 22 April 2019 and 7 November 2023), *The Register* (Google auction, 19 August 2004), CNBC (Snowflake, 15 September 2020; Robinhood, 2 July 2021; Rivian, 9 November 2021), CoreWeave press release and Reuters (27 March 2025), *Fortune* (SpaceX S-1, 20 May 2026).

History and etymology

- Christopher Steiner, "How the Nasdaq Got Hacked", *Forbes*, 30 August 2012 (Thomas Peterffy's robot typist); see also his book *Automate This* (2012).
- Online Etymology Dictionary, short ("sell short" attested by 1852).
- Hendrik Bessembinder, "Trade Execution Costs and Market Quality after Decimalization", *Journal of Financial and Quantitative Analysis* 38(4), 2003.
- Richard Smitten, *Jesse Livermore: World's Greatest Stock Trader* (Wiley, 2001).

Notes on Sources Not Cited Inline

Some specific facts in this document are well established in the historical and financial literature but could not be attributed to a single verifiable primary academic document within this text, but rather second hand sources such as NYSE own publications and posts. For example:

Closing auction volume: The share of daily volume executed in US closing auctions varies by year, instrument, and calendar (it spikes on index rebalancing days); the figures in the text are from BMLL's 2024 analysis cited above. See The shifting dynamics of the NYSE Closing Auction

Back-Cover Text

You can write production-grade software in many domains without ever asking what a "spread" really is, why a matching engine must be deterministic, or what happens when a market suddenly halts.

Financial exchanges are not one of those domains.

How a Financial Exchange Works is a practical conceptual guide for developers, architects, product owners, QA engineers, and operations teams who need to understand market structure fast, without wading through rulebooks before they can ship. It explains the language, mechanics, and failure modes of modern markets in plain English: from bids, asks, and order books to auctions, implied matching, risk controls, clearing, settlement, surveillance, and market data economics.

This second edition is significantly expanded and updated, with deeper treatment of exchange microstructure, implementation-relevant architecture, and the real operational realities teams face in live systems.

Inside you will learn:

- How exchanges create price discovery, liquidity, and fairness
- How orders, priority rules, and matching logic produce trades
- Why tick sizes, auctions, and market fragmentation matter in practice
- How risk controls, conformance testing, and post-trade processes keep markets stable
- Where real-world edge cases appear, and what they imply for system design

No hype. No hand-waving. No prerequisite finance degree.

Just the map you need to stop translating jargon and start reasoning clearly about the system you are building.

About the Author

Johan Persson is a software engineer and musician with over 30 years of experience in high-performance, safety-critical systems. He has worked in multiple fields in software design and among them modern financial trading systems, as well as with thruster design for satellite control systems. Johan is passionate about making complex systems understandable and accessible and actually enjoys writing documentation. Johan lives in Sweden with his family and enjoys playing banjo (5-string, of-course!), cycling, wood-working, and when time allows, doing electronics design for analog computers.

As a companion to this book Johan has also written an educational exchange platform `EduMatcher` which implements many of the techniques discussed in this book. While out of necessity it is simplified compared with real exchanges it is still a surprisingly complete multi-process, message-bus based trading platform that can run on a laptop. It uses its own protocol suit with names that is a wink to the fintech industry; ALF, BALF, CALF, RALF, and LALF, with ALF=**AL**most **F**ix, BALF=**B**inary **ALF**, CALF=**C**hannel **ALF**, RALF=**R**econcilliation **ALF**, LALF=**L**ogging **ALF**. It makes an excellent companion to run while reading this book!