Algorithmic Trading: The Speed Paradox

We built machines to trade for us – now they trade around us, at speeds we cannot supervise, with consequences we cannot predict

Digital Finance

Why Did We Build Machines That Trade Faster Than We Can Think?

The Speed Paradox

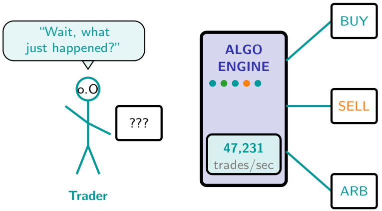
A portfolio manager needs to buy one million shares without moving the market price. Doing it by hand would take days, cost a fortune in market impact, and leak information to competitors. So we built algorithms to slice the order into thousands of small pieces and execute them automatically.

What algorithms were built to do:

- Execute large orders without moving the market price
- Reduce transaction costs by optimizing timing and venue
- Remove human emotion from execution decisions
- Operate across multiple markets simultaneously

What algorithms now actually do:

- Make autonomous trading decisions in microseconds
- React to market events faster than any human can perceive
- Interact with other algorithms in feedback loops no one designed
- Occasionally amplify crashes that wipe out billions in minutes



You asked for speed.
Speed stopped asking for you.

Algorithmic trading began as a tool to help humans execute decisions faster – but the machines now make decisions of their own, at speeds no human can follow.

How Many Algorithms Touched Your Last Investment Before You Even Saw the Price?

Reflection Prompt

Open your brokerage app. You see CHF 142.37 – simple, objective, final. But before it reached your screen: **1.** How many algo market makers set that bid/ask? **2.** Which algo routed your order – best for you, or for your broker? **3.** Do algos front-run the index fund rebalancing you never see?

At least six algorithmic layers sit between your “buy” button and execution:

1. Broker’s **smart order router** selects the venue
2. Venue’s **matching engine** pairs your order with a counterparty
3. **Market makers** (algorithms) provide the liquidity you trade against
4. **Dark pools** may internalize your order before it reaches the exchange
5. **Index rebalancing** algorithms determine what your fund holds
6. **Risk management** algorithms set your broker’s margin requirements

You chose none of these. You cannot observe them. But they determine the price you pay and whether your order moves the market.

Bring your count to class. How many intermediaries between your “buy” click and settlement?

Every price you see, every trade you make, and every index fund you hold is shaped by algorithms you never consented to and cannot observe.

What Separates a Helpful Execution Tool from an Autonomous Trading Machine?

Dimension	Gen 1: Execution Algos	Gen 2: Systematic Strategies	Gen 3: AI/ML Trading
Goal	Execute human decision	Generate trading signals	Discover patterns autonomously
Speed	Milliseconds	Milliseconds–seconds	Microseconds
Decision-maker	Human decides what; algo decides when	Human designs rules; algo follows	Algo learns rules from data
Strategy type	TWAP, VWAP, Implementation Shortfall	Momentum, mean reversion, stat arb	Deep learning, NLP sentiment, reinforcement learning
Regulation	MiFID II best execution	Position limits, market abuse	Emerging (EU AI Act, SEC proposals)
Primary risk	Poor execution	Model failure, regime change	Overfitting, unexplainable behavior

The generational shift

Read from left to right. With each generation, the human moves further from the decision:

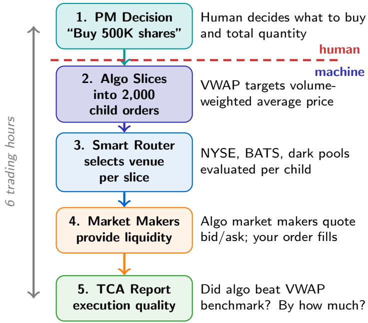
- **Gen 1 (Execution):** The human decides to buy. The algorithm decides how to slice and route the order. The human retains full strategic control.
- **Gen 2 (Systematic):** The human designs the trading rules. The algorithm applies them without hesitation. The human controls the logic but not the speed of execution.
- **Gen 3 (AI/ML):** The algorithm discovers its own rules from data. The human sets constraints but may not understand why the algorithm trades as it does. The machine has operational autonomy.

The paradox in each generation: More automation means lower costs and faster execution – but also less human oversight and harder-to-predict failures.

Three generations of algorithmic trading – each one gave more decision-making power to the machine and less to the human.

(c) Joerg Osterrieder 2026

Follow One VWAP Order from Portfolio Manager to Execution Across Six Hours



Anatomy of a VWAP execution

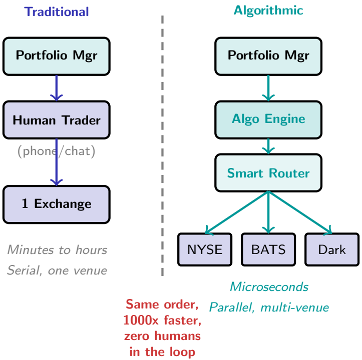
- **VWAP** (Volume-Weighted Average Price) is the benchmark: the algorithm tries to buy at the average price weighted by volume throughout the day
- **Why slice?** Executing 500,000 shares at once would move the price against you (market impact). Slicing into 2,000 child orders of 250 shares each spreads the impact across the trading day
- **Smart order routing** evaluates each venue in real time: Which has the best price? Which has depth? Which charges the lowest fee? Decisions are made in microseconds
- **Market makers** are themselves algorithms. Your algorithm trades against their algorithm. Neither involves a human in the execution loop
- **Transaction cost analysis (TCA)** measures whether the algo achieved its benchmark. MiFID II requires brokers to report execution quality

The one human decision was "buy 500K shares." Everything else – timing, venue, counterparty, slice size – was decided by machines talking to machines.

A single VWAP order can touch dozens of venues and thousands of counterparties – all invisible to the human who clicked "buy."

(c) Joerg Osterrieder 2026

How Does a Human-Driven Trading Desk Differ from an Algorithmic One?



Two architectures, same goal

- **Traditional:** PM tells a human trader what to buy. Trader uses judgment and relationships to execute on one exchange. Errors are human; so is oversight.
- **Algorithmic:** PM enters an order into an EMS. Algo engine selects a strategy (VWAP, TWAP, IS). Smart router evaluates multiple venues and routes child orders to the best destination.

Why the right side wins on cost:

- Multi-venue access captures better prices
- Smaller slices reduce market impact
- No human latency between decision and execution

Why the right side worries regulators:

- No human can supervise microsecond decisions
- Algorithms interact unpredictably with each other
- When algorithms fail, they fail at algorithmic speed

The architectural shift from serial human execution to parallel algorithmic execution is why algo trading dominates – it is not just faster, it is structurally different.

What Happens When an Algorithm Crashes the Market in 36 Minutes?

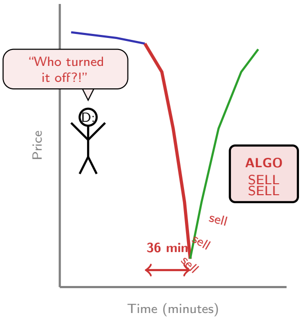
When speed becomes a weapon
On May 6, 2010, the US stock market lost nearly $1 trillion in 36 minutes – then recovered. The cause: a single algorithm selling 75,000 E-mini futures contracts without regard to price or time, triggering a cascade of algorithmic responses.

The Flash Crash anatomy:

- A mutual fund's algorithm sold a massive futures position targeting volume, not price
- Other algorithms detected downward momentum and joined the selling
- Market makers withdrew liquidity, widening spreads enormously
- Some stocks traded at $0.01; others at $100,000
- Human circuit breakers eventually halted trading

Other algorithmic failures:

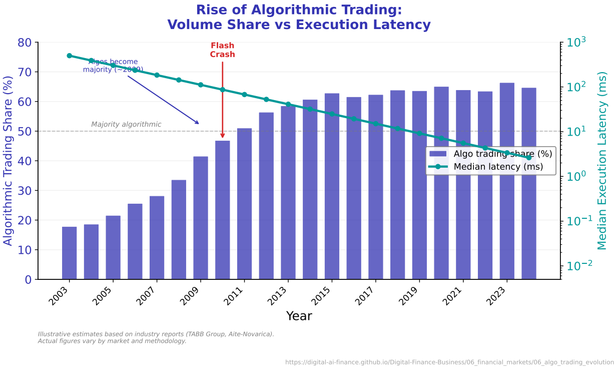
- **Knight Capital (2012):** Software error caused $440M in losses in 45 minutes – bankrupting the firm
- **Spoofing:** Fake orders to manipulate prices, cancelled before execution (illegal under Dodd-Frank)
- **Quote stuffing:** Flooding exchanges to slow competitors' systems



The algorithm did exactly what it was told. That was the problem.

Flash crashes, rogue algorithms, and feedback loops are not bugs – they are the predictable consequences of systems that operate faster than humans can intervene.

How Did Algorithms Go from Executing Orders to Dominating Markets?



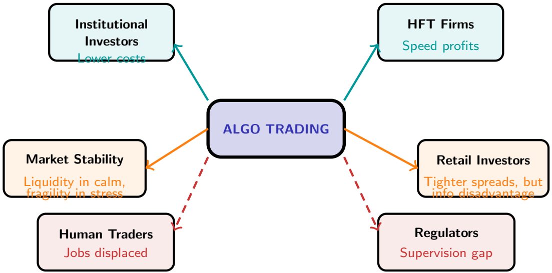
Reading the two trends together

- **The bars** show algorithmic trading's share of total US equity volume. It rose from under 20% in the early 2000s to over 60% by the mid-2010s – and has stabilized there, suggesting a mature market
- **The line** shows median trade execution latency. It collapsed from hundreds of milliseconds to single-digit microseconds – a millionfold speedup in two decades
- **The inflection point** (around 2007–2009) is where both trends crossed a threshold: algorithms became the majority of trading volume, and latency dropped below the threshold of human perception
- **Post-2015 plateau:** Volume share stabilized but latency kept falling – the arms race shifted from market share to speed advantage among algorithms competing with each other

The implication: Markets are now primarily algorithm-to-algorithm. Human traders are a minority by volume, interacting with a system designed by and for machines.

Illustrative estimates based on industry reports. As algorithmic share rose from under 20% to over 60%, execution latency fell from seconds to microseconds – a millionfold speedup.

Who Wins and Who Loses When Machines Replace Human Traders?



Winners:

- **Institutional investors** pay 30–50% lower execution costs since 2005
- **HFT firms** profit from speed, earning small margins on enormous volume

Losers:

- **Human traders** displaced from execution roles; floors that employed hundreds now run with a few engineers
- **Regulators** cannot supervise microsecond systems; flash crashes are over before they are detected

Mixed:

- **Retail investors** get tighter spreads but face info asymmetry: algos can front-run their orders
- **Market stability** improves in calm (constant liquidity) but deteriorates in stress (simultaneous withdrawal)

Algorithmic trading lowers costs for large institutions but creates a two-tier market – those with algorithmic infrastructure and those without.

Four Questions That Reveal Whether an Algorithmic System Helps or Harms the Market

When evaluating any algorithmic trading system – whether as a regulator, investor, or engineer – ask these four questions:

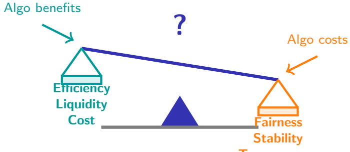
1. Does it provide or consume liquidity?
Market-making algorithms that post bid/ask quotes *provide* liquidity and tighten spreads. Momentum algorithms that chase trends *consume* liquidity and widen spreads. The same technology can help or harm depending on the strategy.

2. How does it behave under stress?
Many algorithms are designed to withdraw from the market when volatility spikes – precisely when liquidity is most needed. An algorithm that provides liquidity only in calm markets is a fair-weather friend.

3. Who bears the cost of its errors?
When Knight Capital lost $440 million in 45 minutes, the firm went bankrupt – but the market absorbed the price distortions. Algorithmic errors create externalities.

4. Can it be supervised at the speed it operates?
If an algorithm makes decisions in microseconds but the kill switch requires a human to notice, evaluate, and act, the supervision is structurally inadequate.

The framework: These four questions map to the key regulatory debates: market making obligations, circuit breakers, liability rules, and real-time surveillance.



Algo trading tips toward efficiency.
The question is the cost in fairness.

Algorithmic trading is not inherently good or bad – but its benefits accrue to those with capital and infrastructure, while its risks are shared by everyone.

(c) Joerg Osterrieder 2026

Your Challenge: Evaluate an Algorithmic Trading Strategy

Mini-Challenge (15 minutes)

A Swiss pension fund pays a human desk 8 bps/trade to rebalance quarterly over 2–3 days. A vendor offers VWAP algo execution at 3 bps with same-day completion via smart order routing across European venues. The board must decide: algorithm, human, or hybrid?

Apply the four evaluation questions:

1. **Provide or consume liquidity?** VWAP blends with natural volume – but is the rebalancing large enough to move prices? How does the algo handle illiquid Swiss mid-caps vs liquid blue chips?
2. **Behavior under stress?** What if a flash crash occurs mid-execution? Does the vendor guarantee quality in volatile markets?
3. **Who bears error costs?** If the algo misprices a trade, who absorbs the loss – vendor or fund? Human desk made judgment calls; the algorithm follows rules. Which is safer for retirees?
4. **Supervisable at operating speed?** Does the fund have expertise to monitor real-time algo execution, or are they trusting a black box they cannot inspect?

Discuss: What would you recommend? Human, algorithm, or hybrid – and why?

The four-question framework works for any algorithm – from simple VWAP execution to complex ML-driven strategies. Practice it once, and you can apply it to any system.

(c) Joerg Osterrieder 2026