

WINKLER TRADING STATION

DIE 4-TEILIGE TRADING-REIHE TEST-READER

24 Seiten Orientierung, Leseproben und Schwierigkeitscheck



WINKLER
TRADING STATION

01 BASICS

Fundament und Marktverständnis

02 MODELS

Orderflow-Strategien und manuelle Ausführung

03 ADVANCED ORDERFLOW

Level 3, HFT und institutionelle Mikrostruktur

04 AI / TRADING & ANALYTICS

Python, Quant Research, LLM und Data Analytics

WILLKOMMEN

Was dieser Test-Reader leistet

Ein realistischer Blick auf Inhalt, Sprache, Arbeitsweise und Anspruch der vierbändigen Reihe.

Dies ist keine Kurzfassung zum schnellen Nachhandeln. Der Reader zeigt, wie die vier Bände denken, welche Vorarbeit sie verlangen und welcher Lernweg für einen kompletten Anfänger sinnvoll ist.

- Du erkennst den Unterschied zwischen Grundlagenwissen, Strategieentwicklung, professioneller Mikrostruktur und AI-gestütztem Research.
- Du bekommst pro Band eine echte inhaltliche Leseprobe: Begriffe, Entscheidungslogik, typische Aufgaben und Qualitäts Gates.
- Die Schwierigkeitswerte bewerten nicht nur das Lesen, sondern das selbstständige Anwenden unter realistischen Marktbedingungen.
- Zeitangaben sind Lernkorridore, keine Versprechen. Vorwissen, Datenzugang, Programmiererfahrung und Replay-Praxis verändern den Aufwand stark.
- Alle Chartbilder in den Bänden sind Lehr- oder synthetische Visualisierungen. Sie stellen keinen Track Record und keine Gewinnprognose dar.

STUFE A

Lesen

Begriffe wiedererkennen, Argumentation nachvollziehen und Beispiele erklären. Das ist die erste Stufe.

STUFE B

Meistern

Regeln selbst definieren, Daten prüfen, Fehler erkennen und robuste Entscheidungen reproduzieren. Das ist die eigentliche Arbeit.

WT-72869680

SERIENKARTE

Vier Baende - vier verschiedene Aufgaben

Zusammen bilden sie eine Kette vom Marktgrundwissen bis zum professionellen Forschungs- und Produktionssystem.

01

Winkler Trading - Basics

176 Seiten. Futures, Charttechnik, Markttechnik, Volume Profile, Footprint, CVD, DOM, ES/NQ, Risiko und NinjaTrader.

02

Models - Trading Strategies

328 Seiten. Sieben Orderflow-Modelle, 70 Chartbeispiele, manuelles Scalping, Research, Statistik und Risikomanagement.

03

Advanced Orderflow

328 Seiten. Level 3/MBO, Queue, Latenz, HFT, Market Making, TCA, ML auf Orderbüchern und Firmenarchitektur.

WT-32087331

04

AI / Trading & Analytics

252 Seiten. Python, Data Engineering, Analytics, Quant Research, ML, lokale LLMs, Backtesting, MLOps und ES/NQ-Orderflow.

SCHWIERIGKEIT

Wie hart ist jeder Band für einen totalen Anfänger?

Lesbarkeit und Meisterschaft werden getrennt bewertet. Zehn bedeutet: ohne mehrstufige Vorbildung nicht sinnvoll beherrschbar.

01 BASICS



02 MODELS



04 AI / TRADING & ANALYTICS



03 ADVANCED ORDERFLOW



WT-54110060

Warum das Ranking so ausfällt

- Basics ist breit, aber didaktisch von null aufgebaut. Die Schwierigkeit entsteht beim Übertragen in Replay und Risikodisziplin.
- Models verlangt schnelle visuelle Erkennung, regelkonforme Ausführung und statistische Validierung gleichzeitig.
- AI verlangt zusätzlich Python, Datenlogik, Statistik, Modellkritik und Produktionsdenken. Für Data Analysts ist der Einstieg deutlich leichter.
- Advanced Orderflow beginnt oberhalb der Retail-Ebene. Exchange-Protokolle, Kausalität, L3-Rekonstruktion und Latenzmodelle machen ihn zum härtesten Band.

LERNROUTE

Der empfohlene Weg durch die Reihe

Ein kompletter Anfänger sollte nicht versuchen, alle 1.084 Seiten linear und ohne Praxisblock zu konsumieren.

1

BASICS

Markt, Instrument, Orders, Profile und Risiko verstehen.

8-16 Wochen

2

AI FUNDAMENT

Python, Datenqualität, EDA und Backtesting aus Band 4.

8-20 Wochen

3

MODELS

Ein bis zwei Setups im Replay testen, nicht alle gleichzeitig.

16-32 Wochen

4

ADVANCED

L3/HFT/Quant erst nach belastbarer Tickdaten- und Statistikpraxis.

9-24 Monate

5

AI PRODUCTION

ML, LLM, MLOps und kontrollierte Automation vertiefen.

laufend

SELBSTCHECK

Welche Voraussetzungen bringst du mit?

Null Punkte sind kein Problem. Sie zeigen nur, wo dein Einstieg beginnt.

Kompetenz	Basics	Models	Advanced	AI
Futures/Orders	lernen	sicher	sehr sicher	sicher
Footprint/CVD	lernen	sicher	vorausgesetzt	Featurequelle
Statistik	Basis	anwenden	fortgeschritten	fortgeschritten
Python	optional	hilfreich	erforderlich	Kern
Tick/L2/L3-Daten	Einblick	Replay	Kern	Pipeline
Live-Ausführung	Simulation	manuell	institutionell	kontrolliert

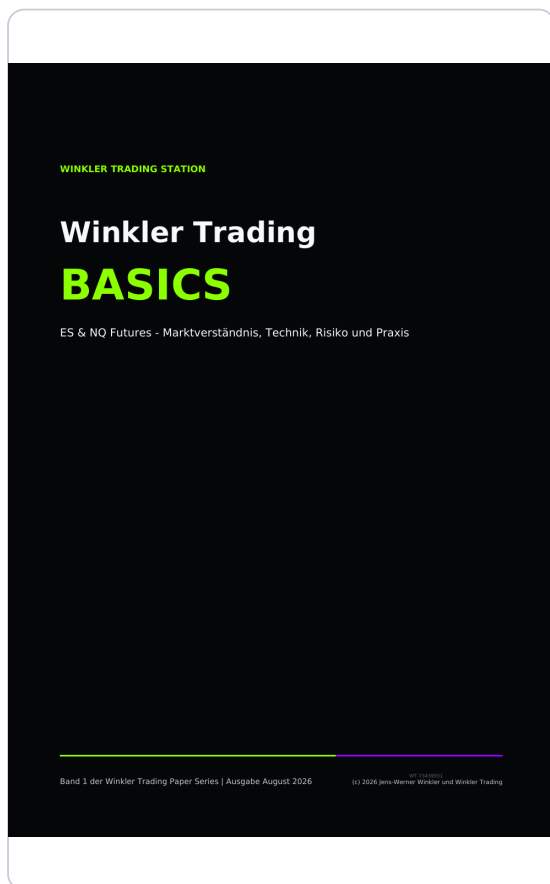
Startentscheidung

- 0-2 Bereiche sicher: mit Basics beginnen und mindestens 20 Replay-Sessions dokumentieren.
- 3-4 Bereiche sicher: Models parallel zu Statistik- und Datenarbeit lesen.
- Python plus Statistik, aber wenig Marktmechanik: zuerst Basics, dann AI und Models.
- L2/Tickdaten plus belastbare Researchpraxis: Advanced Orderflow ist fachlich sinnvoll.
- Wer nur Signale sucht, wird in keinem Band das eigentliche Ziel finden: Die Reihe trainiert Entscheidungsprozesse.

BAND 1

Winkler Trading - Basics

Das Fundament: von Futures und Plattformen bis Orderflow, Quant-Auswertung und kontrollierter Umsetzung.



Was der Band beantwortet

- Was ist ein Future und wie wirken Tickwert, Margin, Session, Rollover und Kontraktwahl?
- Wie werden Limit- und Market Orders gematcht, und was zeigen DOM, Level 2 und Tape wirklich?
- Wie verbinden sich Dow-Theorie, Markttechnik, Chartstruktur, Volume Profile und Footprint?
- Wie unterscheiden sich ES und NQ in Volatilität, Tempo, Tickrisiko und Ausführung?
- Wie wird aus Beobachtung ein testbares Setup mit Risiko-, Journal- und Reviewprozess?

EINSTIEG

Zielgruppe

Komplette Anfänger, Trading-Umsteiger und Leser, die Wissenslücken systematisch schließen wollen.

BESTANDEN

Mastery Gate

Du kannst eine Session vorbereiten, Ordertypen erklären, Profizonen markieren und einen Trade vollständig in R dokumentieren.

INHALTSKARTE

Die 15 Phasen von Basics

Der Band wird anspruchsvoller, ohne den roten Faden vom Instrument zur kontrollierten Entscheidung zu verlieren.

01 Fundament

Wissen -> Anwendung -> Kontrolle.

02 Marktzugang

Wissen -> Anwendung -> Kontrolle.

03 Arbeitsumgebung

Wissen -> Anwendung -> Kontrolle.

04 Marktmechanik

Wissen -> Anwendung -> Kontrolle.

05 Charttechnik

Wissen -> Anwendung -> Kontrolle.

06 Dow-Theorie & Markttechnik

Wissen -> Anwendung -> Kontrolle.

07 Volumen & Volume Profile

Wissen -> Anwendung -> Kontrolle.

08 Orderflow & Footprint

Wissen -> Anwendung -> Kontrolle.

09 CVD, Tape & DOM

Wissen -> Anwendung -> Kontrolle.

10 ES-/NQ-Playbooks

Wissen -> Anwendung -> Kontrolle.

11 Trading-Systeme

Wissen -> Anwendung -> Kontrolle.

12 Quantitative Analyse

Wissen -> Anwendung -> Kontrolle.

13 Risiko & Performance

Wissen -> Anwendung -> Kontrolle.

14 NinjaTrader & NinjaScript

Wissen -> Anwendung -> Kontrolle.

15 Simulator zum Business

Wissen -> Anwendung -> Kontrolle.

LESEPROBE

Vom Footprint zur sauberen Aussage

Eine Zahl im Footprint ist zunächst Beobachtung. Erst Kontext und Preisreaktion machen daraus eine Hypothese.

WINKLER TRADING - BASICS

PRAXISBLATT / LEKTION 29

Werkzeugkasten und Anwendung

Fachliche Vertiefung

- Orderflow beschreibt ausgeführte Markttorders relativ zu passiver Liquidität. Er ist eine Mikroskopansicht und ersetzt keinen übergeordneten Kontext.
- Delta ist Ask-Volumen minus Bid-Volumen. Positives Delta bedeutet aggressive Kaufaktivität, aber nicht zwangsläufig steigenden Preis.
- Imbalances und Stacked Imbalances benötigen instrument- und feedbezogene Schwellen; starre Universalwerte sind nicht belastbar.
- Absorption oder Erschöpfung werden erst durch fehlenden Preisfortschritt und anschließende Reaktion plausibel.

Standardprozess

- Begrenze die Fragestellung auf: Darstellung ausgeführter Kontrakte am Bid und Ask je Preis.
- Definiere die beobachtbaren Größen aus: Zellengröße, Bar-Delta, Gesamtvolumen und Darstellungsfiler.
- Lege Instrument, Kontraktmonat, Session, Zeitzone, Feed und Chartaggregation fest.
- Führe die Aufgabe aus: eine Footprint-Bar aus Rohprints nachvollziehen.
- Prüfe ausdrücklich das Fehlerbild: Bid-Ask-Zahlen mit ruhenden Limitorders zu verwechseln.
- Speichere als Ergebnis: eine verständliche Footprint-Lesereihenfolge.

Kontrollfragen

? Welche Information ist direkt beobachtet und welche nur interpretiert?

? Welche Daten oder Bedingungen würden die Hypothese widerlegen?

? Kann ein zweiter Leser dieselbe Regel ohne Rückfrage anwenden?

AUFGABE

Abgabe: eine verständliche Footprint-Lesereihenfolge. Versioniere das Ergebnis und ergänze mindestens ein ES- sowie ein NQ-Beispiel.

WT-19202982

Ausbildungsunterlage | Keine Anlageberatung (c) 2026 Jens-Werner Winkler und Winkler Trading 91

So wird gelesen WT-72396198

- Bid x Ask zeigt ausgeführtes Volumen je Preis, nicht wartende Liquidität.
- Delta misst aggressive Kauf- minus aggressive Verkaufsaktivität.
- Eine Imbalance ist relativ zur Vergleichszelle; Schwelle und Mindestvolumen müssen definiert sein.
- Aggression ohne Preisfortschritt kann auf Absorption hinweisen, beweist sie aber nicht.
- Der Trade entsteht erst mit Profilzone, Marktstruktur, Invalidierung und genügend Raum zum Ziel.

Testfrage: Viele Käufer treffen am Tageshoch ein, der Preis steigt aber nicht. Welche drei Alternativerklärungen musst du vor einem Short prüfen?

Originalseite aus Band 1, verkleinert dargestellt.

EHRLICHER CHECK

Wie schwer ist Basics wirklich?

Leicht zu beginnen, deutlich schwerer konsequent anzuwenden.

Bereich	Lesen	Anwenden	Meistern
Futuresmechanik	3/10	5/10	6/10
Chart & Markttechnik	3/10	6/10	7/10
Orderflow	5/10	7/10	8/10
Risiko & Statistik	4/10	6/10	8/10

Mini-Prüfung

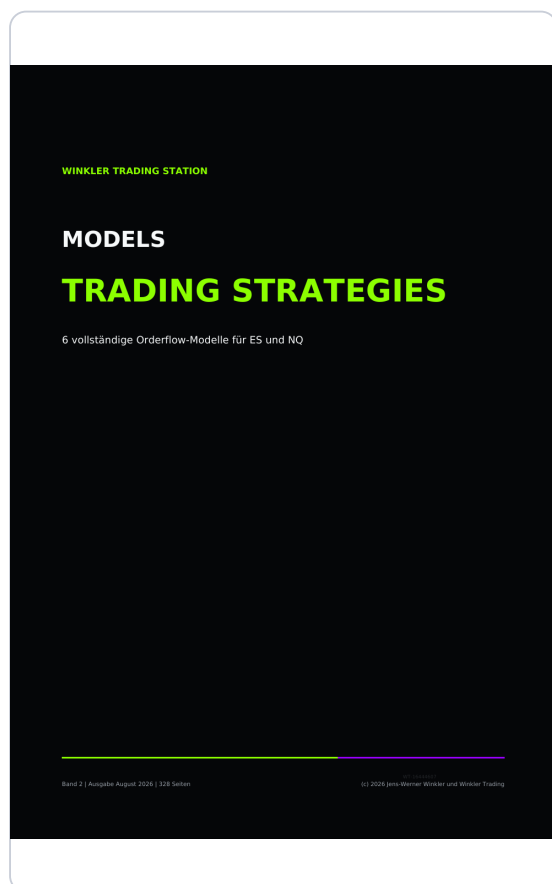
- Kannst du Tickgröße, Tickwert, Stopdistanz und Kontorisiko für ES und NQ ohne Verwechslung berechnen?
- Kannst du VAH, VAL, VPOC und LVN erklären, ohne daraus automatisch Support oder Resistance zu machen?
- Kannst du Market Buy, Limit Sell, CVD und sichtbare DOM-Liquidität sauber unterscheiden?
- Kannst du für einen Trade Kontext, Trigger, Entry, Invalidierung, Ziel, Größe und No-Trade-Regel schriftlich festhalten?
- Kannst du nach 50 Replay-Trades sagen, in welchem Regime dein Setup nicht funktioniert?

Empfehlung: Erst wenn mindestens vier Antworten belastbar mit eigenen Beispielen belegt sind, in Band 2 wechseln.

BAND 2

Models - Trading Strategies

Sieben detaillierte Orderflow-Modelle für ES und NQ - inklusive manueller Ausführung, Research und Risiko.



Was sich gegenüber Basics ändert

- Nicht mehr: Was bedeutet Footprint? Sondern: Welche exakte Footprint-Sequenz ist ein gültiger Trigger?
- Nicht mehr: Wo liegt die Value Area? Sondern: Welche Profilkante liefert Kontext, Raum und Invalidierung?
- Nicht mehr: Ist Delta positiv? Sondern: Passt Aggression, Preisantwort und CVD zum erwarteten Auktionspfad?
- Jedes Modell wird nach ES, NQ, Session und Regime getrennt ausgewertet.
- Die manuelle Umsetzung setzt Stops und Ziele als Obergrenzen: ES maximal 20/40 Ticks, NQ maximal 60/120 Ticks.

PRAXIS

Umfang

328 Seiten, sieben Strategien, 70 synthetische Chartbeispiele sowie Setups, Fehlerkataloge, Journal- und Researchteile.

MODELLKARTE

Die sieben Strategie-Familien

Alle nutzen dieselben Datenquellen, aber eine andere Marktlogik.

01 Liquidity Pullback

Retest einer Profilkante nach Impuls; Einstieg erst bei erneutem Orderflow-Support.

02 Acceptance Breakout

Ausbruch wird nur gehandelt, wenn außerhalb der Balance Akzeptanz entsteht.

03 Orderflow Continuation

Kurze Rebalancierung im Trend, gefolgt von neuer Aggression und Preisfortschritt.

04 Absorption Reversal

Extremzone, hohe Aggression ohne Resultat und bestätigte Rückkehr in die Auktion.

05 Tape Momentum Follow

Tempo, Big Trades und geringe Gegenliquidität begleiten einen echten Drive.

06 Profile-to-Footprint Rotation

Tagesprofil bestimmt Start und Ziel; Footprint taktet den Entry.

07 NQ Value Migration & Effort Ladder

Value Drift, Effort/Result, MBO und drei Regimemodi in einem Forschungsmodell.

WT-95783558

LESEPROBE

Ein Beispieltrade ist eine Entscheidungskette

Das Bild dient nicht zum Kopieren. Es zwingt den Leser, Kontext, Trigger, Risiko und Ergebnis getrennt zu prüfen.

Manueller Ablauf

- Vorbereitung: Session, Referenzzonen und Regime notieren.
- Beobachtung: Value, Delta, Big Trades und Preisantwort erfassen.
- Trigger: Nur die vorab definierte Sequenz erlaubt einen Entry.
- Risiko: Stop gehört hinter die logische Invalidierung, nicht an eine bequeme Geldsumme.
- Management: Keine spontane Regelveraenderung nach dem Fill.
- Review: Auch ein Gewinner kann regelwidrig und damit ein schlechter Trade sein.



Originalseite aus Band 2, verkleinert dargestellt.

RESEARCH-GATE

Wann ist ein Modell mehr als eine Idee?

Ein Setup wird nicht durch zehn schoene Charts validiert.

Gate	Mindestfrage	Typischer Fehler
Definition	Ist jeder Trigger messbar?	Rückschau
Daten	Sind Feed, Session und Roll korrekt?	Mischdaten
Simulation	Werden Queue und Slippage realistisch?	perfekte Fills
Statistik	Genug Ereignisse und Unsicherheit?	kleine Stichprobe
Stabilitaet	Bleibt der Edge out-of-sample?	Overfitting
Risiko	Ist Drawdown tragbar?	zu grosse Größe

Band 2 ist gemeistert, wenn du ein Modell in eigenen Worten operationalisieren, 100-300 Ereignisse sauber testen und klare No-Trade-Regeln verteidigen kannst.

Anfänger-Rating

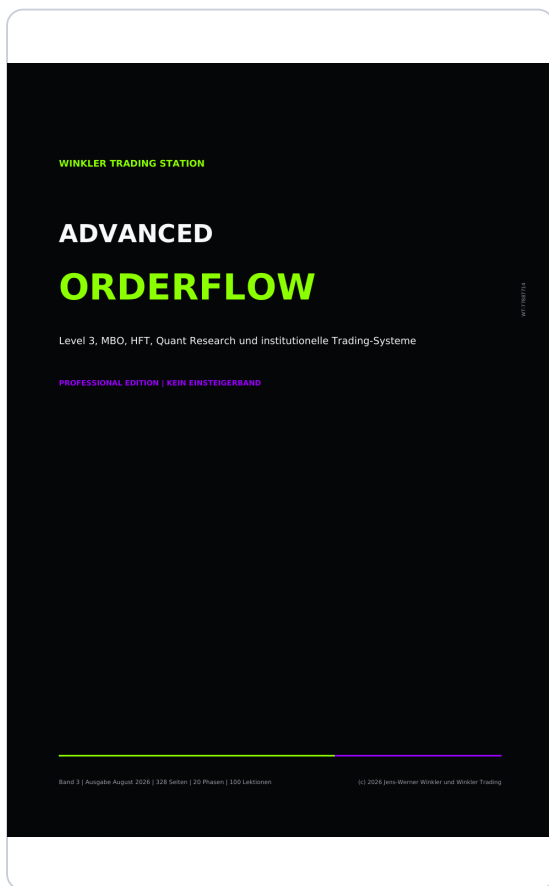
MODELS

Lesen		6/10
Meistern		8/10

BAND 3

Advanced Orderflow

Professional Edition: Exchange Events, Level 3/MBO, HFT, Quant Research und institutionelle Systeme.



WT-40041293

Dieser Band startet nicht bei null

- Tick, Market Order, VPOC, Footprint und CVD werden vorausgesetzt, nicht erneut elementar erklärt.
- Level 3 wird als Venue-spezifischer order-by-order- oder Market-by-Order-Feed behandelt.
- Queue-Position, Fill-Wahrscheinlichkeit, Latenz und Adverse Selection ersetzen naive DOM-Interpretation.
- HFT bedeutet Feed Handling, Book Building, Entscheidungsloop, Gateway, Risk und Recovery - nicht einfach nur schnelles Klicken.
- Die Firmenskapitel untersuchen öffentlich erkennbare Prinzipien erfolgreicher Tradingorganisationen, keine geheimen Strategien.

PROFESSIONAL

Klare Grenze

Keine Anleitung zu Manipulation, Kontrollumgehung oder missbräuchlichen Marktpraktiken. Surveillance bleibt evidenzbasiert und regelkonform.

SYSTEMARCHITEKTUR

Orderflow als institutionelle Produktionskette

Der Edge ist ein Produkt mehrerer Systeme. Ein gutes Signal kann durch schlechte Daten, Ausführung oder Betrieb wertlos werden.

01

EXCHANGE

Messages, Matching, Queue

02

DATA

Capture, Clock, Book

03

RESEARCH

Features, Statistik, ML

04

EXECUTION

Gateway, Fill, TCA

05

RISK

Limits, Stress, Kill

06

FIRM

Teams, Kapital, Governance

Professioneller Edge = Informationsqualität x Modellgüte x Execution x Risikodisziplin x Betriebsstabilität. Nahe null in einem Faktor kann das Gesamtsystem zerstören.

LESEPROBE

Von HFT-Architektur zu Orderbuch-ML

Zwei Seiten zeigen die Spannweite: deterministische Echtzeitsysteme und statistische Modelle auf ereignisbasierten Daten.

WINKLER TRADING - ADVANCED ORDERFLOW

PHASE 10 | LEKTION 01 | KONZEPT

Feed Handler und Decoder

Feed Handler und Decoder wird als überprüfbarer Bestandteil von HFT Systems Architecture operationalisiert.

- 01** Kernel Bypass: Kernel Bypass reduziert Scheduling- und Copy-Overhead, verschiebt aber Verantwortung für Polling, Backpressure, Isolation und Fehlerbehandlung in die Anwendung. Als autoritativer Input wird die genaue Definition einschließlich Venue- und Feedvariante festgeschrieben. Die konkrete Implementierung wird für Feed Handler und Decoder nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.
- 02** Zero Copy: Zero Copy vermeidet Datenkopien im Hot Path; Ownership, Lifetime und Cache-Kohärenz müssen dafür explizit und race-frei sein. Für die zeitliche Kopplung werden Event Time, Receive Time, letzte Eingangszeit und Unsicherheitsfenster gespeichert. Die konkrete Implementierung wird für Feed Handler und Decoder nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.
- 03** Binary Decode: Der Decoder validiert Länge, Template, Version und Sequenz vor Feldzugriff und schreibt Integer-Ticks statt verlustbehafteter Fließkommazahlen. Als Zustandsvariable erhält die Größe Einheit, Normalisierung, Nullmodell und eine mechanisch einfachere Baseline. Die konkrete Implementierung wird für Feed Handler und Decoder nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.
- 04** Sequence Check: Sequence Checks laufen vor Book Mutation. Gap, Duplicate und Reset führen in unterschiedliche, getestete Recovery-Pfade. Im Stresspfad werden Gap, Reset, Latenzspitze, Spreadwechsel, Teillfill und Regimebruch explizit injiziert. Die konkrete Implementierung wird für Feed Handler und Decoder nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.
- 05** Backpressure: Kann ein Consumer nicht folgen, wird Degradation sichtbar gemacht; stilles Dropping erzeugt einen scheinbar schnellen, aber falschen Marktstate. Die Promotion verlangt Out-of-sample-Evidenz, Kostenstress, Online-Parität, Owner und dokumentiertes Rollback-Kriterium. Die konkrete Implementierung wird für Feed Handler und Decoder nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.

KERN

Feed Handler und Decoder wird als überprüfbarer Bestandteil von HFT Systems Architecture operationalisiert.

Professional Research | Keine Anlageberatung (c) 2026 Jens-Werner Winkler und Winkler Trading 154

HFT Feed Handler

Pakete decodieren, Sequenzen prüfen, Gaps erkennen, deterministisch rekonstruieren und Fehler ohne stillen Ersatzpfad behandeln.

Originalseiten aus Band 3, verkleinert dargestellt.

WINKLER TRADING - ADVANCED ORDERFLOW

PHASE 14 | LEKTION 01 | KONZEPT

Label Design ohne Leakage

Label Design ohne Leakage wird als überprüfbarer Bestandteil von Machine Learning on Order Books operationalisiert.

- 01** Horizon: Ein LOB-Modell ist ein kalibrierter Wahrscheinlichkeitsdienst innerhalb eines Latenz- und Risikobudgets. Prediction Score allein ersetzt weder Execution Value noch Driftkontrolle. Als autoritativer Input wird die genaue Definition einschließlich Venue- und Feedvariante festgeschrieben. Die konkrete Implementierung wird für Label Design ohne Leakage nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.
- 02** Mid Move: Ein LOB-Modell ist ein kalibrierter Wahrscheinlichkeitsdienst innerhalb eines Latenz- und Risikobudgets. Prediction Score allein ersetzt weder Execution Value noch Driftkontrolle. Für die zeitliche Kopplung werden Event Time, Receive Time, letzte Eingangszeit und Unsicherheitsfenster gespeichert. Die konkrete Implementierung wird für Label Design ohne Leakage nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.
- 03** Fill Condition: Ein LOB-Modell ist ein kalibrierter Wahrscheinlichkeitsdienst innerhalb eines Latenz- und Risikobudgets. Prediction Score allein ersetzt weder Execution Value noch Driftkontrolle. Als Zustandsvariable erhält die Größe Einheit, Normalisierung, Nullmodell und eine mechanisch einfachere Baseline. Die konkrete Implementierung wird für Label Design ohne Leakage nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.
- 04** Triple Barrier: Ein LOB-Modell ist ein kalibrierter Wahrscheinlichkeitsdienst innerhalb eines Latenz- und Risikobudgets. Prediction Score allein ersetzt weder Execution Value noch Driftkontrolle. Im Stresspfad werden Gap, Reset, Latenzspitze, Spreadwechsel, Teillfill und Regimebruch explizit injiziert. Die konkrete Implementierung wird für Label Design ohne Leakage nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.
- 05** Availability: Ein LOB-Modell ist ein kalibrierter Wahrscheinlichkeitsdienst innerhalb eines Latenz- und Risikobudgets. Prediction Score allein ersetzt weder Execution Value noch Driftkontrolle. Die Promotion verlangt Out-of-sample-Evidenz, Kostenstress, Online-Parität, Owner und dokumentiertes Rollback-Kriterium. Die konkrete Implementierung wird für Label Design ohne Leakage nach Markt, Session und Regime getrennt und mit einem Unsicherheitsintervall berichtet.

KERN

Label Design ohne Leakage wird als überprüfbarer Bestandteil von Machine Learning on Order Books operationalisiert.

Professional Research | Keine Anlageberatung (c) 2026 Jens-Werner Winkler und Winkler Trading 218

ML auf Orderbüchern

Labels ohne Leakage, sequenzielle Repraesentation, Kalibrierung, Latenzbudget und Regimestabilitaet gemeinsam prüfen.

MASTERY GATE

Warum Advanced Orderflow 10/10 erreicht

Der Band verlangt gleichzeitig Mikrostruktur, Datenengineering, Statistik, Software, Risiko und Betrieb.

Ebene	Was beherrscht werden muss
Feed	Message-Semantik, Sequenz, Recovery, Snapshot/Incremental
Book	MBO-Lifecycle, Queue, Priority Loss, Hidden Liquidity
Clock	Event/Receive/Decision Time, Synchronisation, Kausalitaet
Quant	Bootstrap, multiple Tests, Kalibrierung, Kosten
Execution	Fill-Modell, Impact, TCA, Adverse Selection
Production	Risk Gates, Observability, Incident und Governance

Nicht geeignet als erster Tradingband. Ein totaler Anfänger sollte Advanced erst beginnen, wenn Basics sicher sitzt, mindestens ein Modell sauber getestet wurde und Python-/Tickdatenarbeit kein Neuland mehr ist.

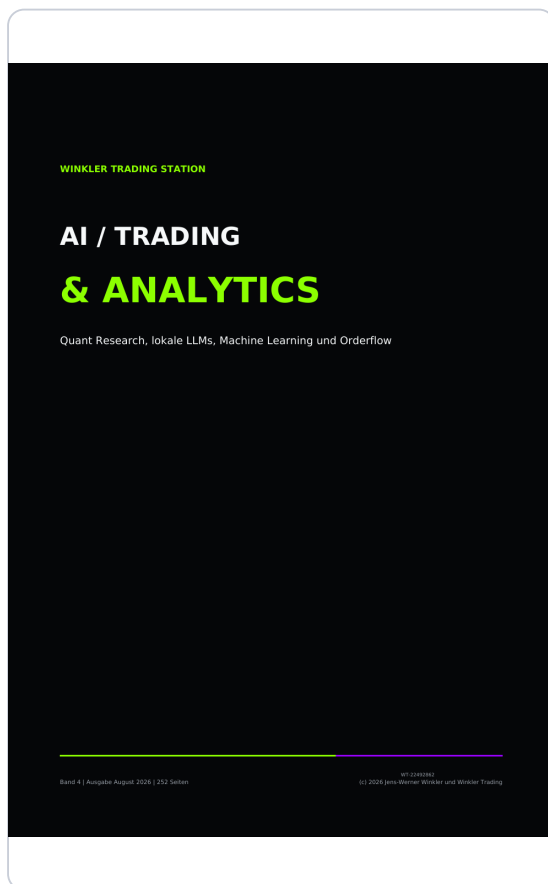
ADVANCED ORDERFLOW

Lesen		9/10
Meistern		10/10

BAND 4

AI / Trading & Analytics

Von Python und Data Analytics bis lokalen LLMs, Machine Learning, MLOps und kontrollierter Trading-Automation.



WTF-13220520

Wofür AI hier eingesetzt wird

- Daten aufbereiten, Qualitätsfehler finden und reproduzierbare Research-Datensätze bauen.
- Trade-Ergebnisse lesen: Expectancy, Drawdown, MAE/MFE, bedingte Performance und Regime.
- Hypothesen, Features, Backtests und Modelle mit Baselines und Unsicherheit prüfen.
- Lokale LLMs für Strukturierung, Research, Codeassistentz und RAG nutzen - nicht als unkontrolliertes Orakel.
- ES/NQ-Orderflow mit Footprint, CVD, Big Trades, Tape Speed, DOM, Level 2 und Heatmaps in Features übersetzen.

CONTROL

Kernidee

AI ist ein Verstärker für Datenqualität und Disziplin. Ein Modell bekommt keinen ungeprüften direkten Brokerzugriff.

ANALYTICS

Vom Ergebnis zur besseren Strategie

Data Analytics beantwortet nicht nur, ob ein System verdient hat, sondern wann, warum und mit welchem Risiko.

17839020

1. BESCHREIBEN

Verteilung, Winrate, Payoff, Profit Factor, Drawdown, MAE/MFE.

2. SEGMENTIEREN

Markt, Session, Wochentag, Volatilität, Trend und Setupqualität.

3. DIAGNOSTIZIEREN

Welche Bedingung erklärt Verlustcluster oder schwache Exits?

4. HYPOTHESE

Eine kleine, messbare Regelveränderung formulieren.

5. VALIDIEREN

Walk-forward, Kosten, Bootstrap und Out-of-sample.

6. ENTSCHEIDEN

Behalten, begrenzen, überarbeiten oder verwerfen.

LESEPROBE

Lokales LLM und ES/NQ-Orderflow

WT-36263127

Der Band verbindet Sprachmodelle mit harten Datenprozessen und getrennten Freigaberechten.

WINKLER TRADING - AI / TRADING & ANALYTICS

PHASE 13 | LEKTION 1 | METHODE

Arbeitsprozess und Qualitätsgates

Cloud-LLM versus lokales LLM

SCHRITT 1

Daten klassifizieren. Ergebnis und Prüfkriterium werden unmittelbar im Research Log gespeichert.

SCHRITT 2

Aufgabe benchmarken. Ergebnis und Prüfkriterium werden unmittelbar im Research Log gespeichert.

SCHRITT 3

Kosten messen. Ergebnis und Prüfkriterium werden unmittelbar im Research Log gespeichert.

SCHRITT 4

Fallback definieren. Ergebnis und Prüfkriterium werden unmittelbar im Research Log gespeichert.

Vollständiger Ablauf

- 1 Daten klassifizieren. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.
- 2 Aufgabe benchmarken. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.
- 3 Kosten messen. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.
- 4 Fallback definieren. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.
- 5 Providerwechsel testen. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.

Typische Fehlentscheidungen

- Sensible Daten ungefiltert senden. Dieser Fehler wird durch einen automatisierten Test oder ein dokumentiertes Review-Gate abgefangen.
- Modellnamen als Qualitätsgarantie. Dieser Fehler wird durch einen automatisierten Test oder ein dokumentiertes Review-Gate abgefangen.
- Keine Versionierung. Dieser Fehler wird durch einen automatisierten Test oder ein dokumentiertes Review-Gate abgefangen.

WT-47817780

Ausbildungsunterlage | Keine Anlageberatung (c) 2026 Jens-Werner Winkler und Winkler Trading 164

Lokale LLM-Pipeline

Ollama, llama.cpp, GGUF, Quantisierung, strukturierte Ausgaben und RAG werden nach Hardware, Datenschutz und Prüfbarkeit bewertet.

Originalseiten aus Band 4, verkleinert dargestellt.

WINKLER TRADING - AI / TRADING & ANALYTICS

WT-30284543

PHASE 17 | LEKTION 1 | METHODE

Arbeitsprozess und Qualitätsgates

ES/NQ-Mikrostruktur und Sessionkontext

SCHRITT 1

Kontrakt prüfen. Ergebnis und Prüfkriterium werden unmittelbar im Research Log gespeichert.

SCHRITT 2

Session labeln. Ergebnis und Prüfkriterium werden unmittelbar im Research Log gespeichert.

SCHRITT 3

Events markieren. Ergebnis und Prüfkriterium werden unmittelbar im Research Log gespeichert.

SCHRITT 4

Features normalisieren. Ergebnis und Prüfkriterium werden unmittelbar im Research Log gespeichert.

Vollständiger Ablauf

- 1 Kontrakt prüfen. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.
- 2 Session labeln. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.
- 3 Events markieren. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.
- 4 Features normalisieren. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.
- 5 Regime separat validieren. Vor dem nächsten Schritt werden Input, Output, Zeitbezug und Prüfsumme protokolliert. Ein Fehler stoppt den Lauf, statt stillschweigend mit Ersatzwerten fortzufahren.

Typische Fehlentscheidungen

- ES-Schwelle auf NQ übertragen. Dieser Fehler wird durch einen automatisierten Test oder ein dokumentiertes Review-Gate abgefangen.
- RTH und ETH mischen. Dieser Fehler wird durch einen automatisierten Test oder ein dokumentiertes Review-Gate abgefangen.
- Rolltag ignorieren. Dieser Fehler wird durch einen automatisierten Test oder ein dokumentiertes Review-Gate abgefangen.

Ausbildungsunterlage | Keine Anlageberatung (c) 2026 Jens-Werner Winkler und Winkler Trading 216

Orderflow-Feature

Aggression, CVD, Big Trades, Tape Speed und DOM-Zustände werden punktgenau berechnet und gegen Preisantwort sowie Regime getestet.

Ausbildungsübersicht | Keine Anlageberatung

(c) 2026 Jens-Werner Winkler und Winkler Trading

21

MASTERY GATE

Was AI-Kompetenz im Trading wirklich bedeutet

Ein Notebook, das laeuft, ist noch kein belastbares Research-System.

- Du kannst Rohdaten mit Zeitzone, Kontrakt, Session und Point-in-Time-Logik reproduzierbar aufbereiten.
- Du liest Expectancy, Payoff, Drawdown, MAE/MFE und bedingte Ergebnisse, ohne einzelne Kennzahlen zu überhöhen.
- Du erkennst Leakage, Daten-Snooping, Overfitting, nicht stationaere Regime und unrealistische Kostenannahmen.
- Du vergleichst komplexe Modelle gegen einfache Baselines und berichtest Kalibrierung sowie Unsicherheit.
- Du begrenzt LLM-Werkzeuge durch Schemas, Tests, Rechte, Logging und menschliche Freigabe.
- Du kannst ein Research-Ergebnis in eine klare Regel, einen Backtest, Paper Trading und ein kontrolliertes Deployment überfuehren.

MASTER**Für totale Anfänger**

9/10, weil Trading, Statistik, Python und Software Engineering gemeinsam gelernt werden müssen.

TRANSFER**Für Data Analysts**

6-8/10: Daten- und Pythonwissen helfen stark; Marktmechanik und Execution bleiben eigenstaendige Lernfelder.

ENTSCHEIDUNG

Welcher Band ist jetzt der richtige?

Die beste Reihenfolge folgt deiner größten Wissenslücke, nicht dem spannendsten Titel.

Ausgangslage	Start	Danach
Totaler Anfänger	Basics	AI-Basis -> Models
Trader ohne Datenpraxis	Basics Quant-Teil	AI -> Models
Data Analyst ohne Futures	Basics	AI -> Models
Orderflow-Trader	Models	AI -> Advanced
Quant mit L2/L3	Advanced	AI Production
Nur Strategie gesucht	Models nach Basics	ein Modell testen

Gesamt-Ranking für komplette Anfänger

1. Basics	6/10	Pflichtfundament
2. Models	8/10	Praxis und Statistik
3. AI / Trading & Analytics	9/10	Daten, Code und Modelle
4. Advanced Orderflow	10/10	institutionelles Niveau

WT-2

ABSCHLUSS

Die Reihe in einem Satz

Erst verstehen, dann messen, anschliessend testen und erst zuletzt kontrolliert ausführen.

Basics baut Sprache und Marktverständnis. Models verwandelt Beobachtungen in testbare Playbooks. Advanced Orderflow rekonstruiert die institutionelle Produktionskette. AI / Trading & Analytics macht Daten, Forschung und Automation reproduzierbar.

Was der Leser nach diesem Test-Reader wissen sollte

- Keiner der vier Bände ist ein Signalservice oder Gewinnversprechen.
- Der Schwierigkeitsgrad steigt nicht nur durch Fachbegriffe, sondern durch die Pflicht, Entscheidungen zu messen und zu widerlegen.
- Für einen Anfänger ist ein langsamer, dokumentierter Weg schneller als das Springen zwischen Setups.
- Simulation, Replay, konservative Kosten und kleine Risikobudgets gehören vor jede Live-Umsetzung.
- Die gesamte Reihe umfasst 1.084 Seiten und ist als mehrstufige Ausbildung gedacht, nicht als Wochenendlektüre.

WINKLER TRADING

Dank und Ausblick

Danke an alle Leserinnen und Leser, die Trading als Handwerk behandeln. Der Wert der Reihe liegt nicht im schnellsten Signal, sondern in besseren Fragen, saubereren Daten und kontrollierbaren Entscheidungen.

WT-23910176

Grundlage dieses Readers sind die vier finalen Winkler-Trading-PDFs, Ausgabe August 2026.

30-SEITIGES VERTIEFUNGSKAPITEL

AUCTION THEORY + FOOTPRINT

Auktionszustände als eigene Markt-, Footprint- und Delta-Logik



Balance -> Imbalance -> Acceptance -> Rejection -> Failed Auction

CORE LOGIC

Aggression wird erst durch Preisreaktion und anschließende Akzeptanz interpretierbar.

Jeder Zustand besitzt Beobachtung, Alternativerklärung, Invalidierung und No-Trade-Regel.

ZIEL Beobachtung, Inferenz und Handelsentscheidung strikt trennen.

SYSTEMKARTE

Die Kernbilder vollständig neu aufgebaut

Fünf Marktpfade werden mit denselben Ebenen verglichen: Preis, ausgeführtes Bid/Ask-Volumen, Delta, Profil und Preisreaktion.



01

Preisstruktur

Rotation, Drive, Hold, Rückkehr oder gescheiterter Ausbruch.

WT-53613150

02

Footprint

Bid x Ask je Preis sowie relative Imbalances und Cluster.

03

Delta

Nettoaggression wird getrennt vom tatsächlichen Preisresultat gemessen.

04

Volume Profile

Value, VPOC, HVN und LVN liefern Ort und möglichen Auktionsraum.

05

Entscheidung

Kontext -> Aggression -> Reaktion -> Hypothese -> Risiko -> Review.

06

Grenze

Kein einzelnes Farbfeld beweist Kontrolle, Absorption oder Teilnehmerabsicht.

GRUNDMODELL

Auction Market Theory ohne Mythos

Ein Markt ist kein statisches Angebot-Nachfrage-Bild. Er ist ein fortlaufender Suchprozess, in dem Orders zu Preisen gematcht werden.

1. EXPLORATION

Preis verlässt den bekannten Bereich und prüft, ob dort Gegeninteresse existiert.

2. RESPONSE

Limit- und Market-Orders bestimmen, ob der Preis weiterläuft oder zurückkehrt.

3. ACCEPTANCE

Zeit, Volumen und wiederholte Ausführung etablieren einen neuen Handelsbereich.

4. REJECTION

Schnelle Rückkehr und fehlender Anschluss zeigen mangelnde Akzeptanz.

5. ROTATION

In Balance wechselt die Kontrolle kurzfristig; der Preis kehrt häufig zur Mitte zurück.

Wichtig: Die Auktionstheorie erklärt einen Prozess. Sie garantiert weder Richtung noch Ziel und ersetzt keine Invalidierung.

WT-52

FOOTPRINT-ANATOMIE

Was bedeuten Bid, Ask, Preis und Delta?

Der Footprint aggregiert ausgeführte Trades je Preis. Er zeigt keine vollständige Absicht und nicht automatisch die wartende Orderbuchgröße.

BID x ASK			
BID		PREIS	ASK
18		5254.00	74
35		5253.75	91
68		5253.50	55
104		5253.25	42
82		5253.00	39
47		5252.75	28
DELTA			-25

Berechnung

Delta je Preis = Ask - Bid

Bar Delta = Summe aller Preislevel

Bar Delta = -25

- Bid-Spalte: Trades, die typischerweise gegen ruhende Kauf-Limits ausgeführt wurden; häufig aggressive Verkäufer.
- Ask-Spalte: Trades gegen ruhende Verkauf-Limits; häufig aggressive Käufer.
WT-61824569
- Die Klassifikation hängt vom Datenfeed und der Plattformlogik ab.
- Positives Delta bedeutet mehr aggressive Kauf- als Verkaufsmenge - nicht automatisch steigenden Preis.
- Erst Preisreaktion, Position im Profil und Folgeverhalten liefern Interpretation.

Die stärkste Information ist oft nicht das Delta selbst, sondern Preisfortschritt pro Einheit aggressiven Volumens.

FARBLEGENDE

Was die Farben zeigen - und was nicht

Farben sind Anzeigeparameter. Schwellen, Vergleiche und Normalisierung müssen vor jeder Interpretation bekannt sein.

Anzeige	Saubere Aussage	Nicht automatisch
Hohes Bid-Volumen	viel Ausführung am Bid	Verkäufer kontrollieren
Hohes Ask-Volumen	viel Ausführung am Ask	Käufer kontrollieren
Hohes positives Delta	Ask minus Bid stark positiv	Preis muss steigen
Hohes negatives Delta	Ask minus Bid stark negativ	Preis muss fallen
Imbalance	relatives Verhältnis überschreitet Schwelle	sicherer Entry
Imbalance ohne Weg	Aggression erzielt wenig Resultat	bewiesene Absorption

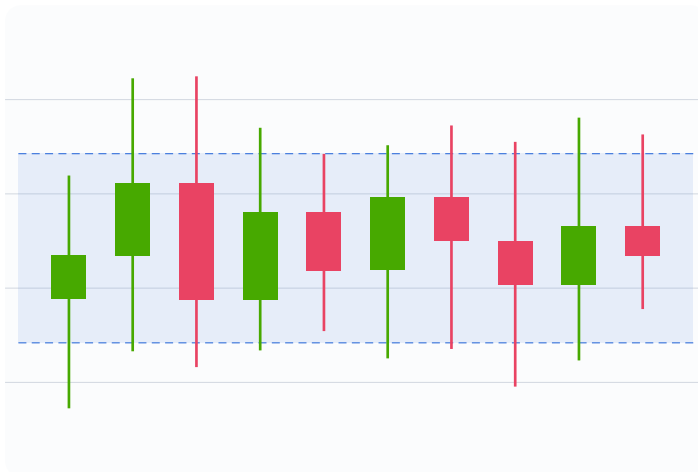
Prüfe in NinjaTrader, ATAS oder anderer Software immer: diagonal oder horizontal, Ratio, Mindestvolumen, Clustergröße, Tick-Aggregation und Sessionfilter.

ZUSTAND 1

BALANCE

Käufer und Verkäufer finden vorübergehend einen fairen Bereich.

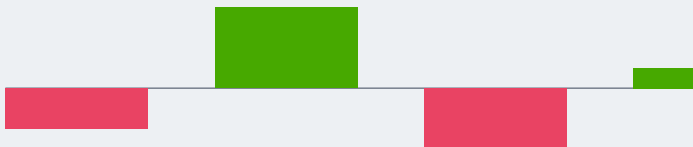
WT-80435097



BALANCE		
BID	PREIS	ASK
84	2353.50	82
95	2353.00	88
87	2352.50	90
96	2352.00	95
110	2351.50	106
98	2351.00	93
DELTA		-16



BAR DELTA



BEOBACHTUNG

Überlappende Kerzen, wechselndes Delta, beidseitige Ausführung und wiederholte Rückkehr zur Mitte.

Beobachtung

Überlappende Kerzen, wechselndes Delta, beidseitige Ausführung und wiederholte Rückkehr zur Mitte.

Inferenz

Die Auktion rotiert. Keine Seite erzielt nachhaltigen Preisfortschritt.

Handelslogik

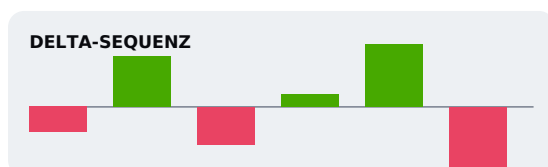
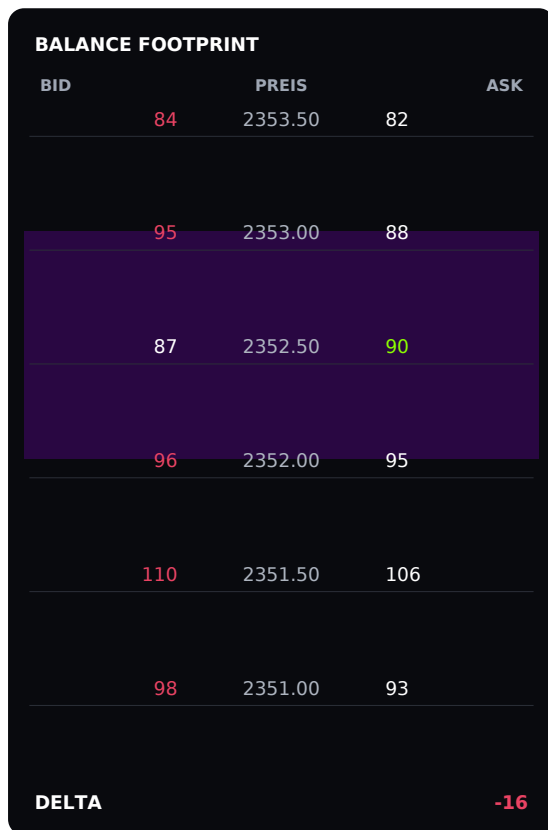
Ränder nur mit bestätigter Ablehnung handeln oder auf einen akzeptierten Ausbruch warten.

Nur die Beobachtung kommt direkt aus den Daten. Inferenz und Trade benötigen Alternativerklärungen, Kontext und Invalidierung.

ZUSTAND 1 - DEEP DIVE

BALANCE im Footprint lesen

Die gleiche Form kann in einem anderen Session- oder Profilkontext eine andere Bedeutung besitzen.



Prüfreihenfolge

- Kontext: Wo liegt das Ereignis relativ zu Value, VPOC, VAH/VAL, Range und Sessionextrem?
- Aggression: Welche Seite handelt am Bid oder Ask, mit welchem Volumen und Tempo?
- Resultat: Wie viele Ticks Preisweg entstehen pro Volumen- oder Deltaeinheit?
- Persistenz: Bleibt die Reaktion über mehrere Events bestehen oder wird sie sofort negiert?
- Invalidierung: Welches beobachtbare Ereignis macht die Arbeitshypothese falsch?

NICHT HANDELN

Gegenbeispiel

BALANCE kann durch News, dünne Liquidität, Roll- oder Sessionwechsel, Datenlücken oder kurzfristige Hedging-Flows nur vorgetäuscht werden.

Arbeitsauftrag: Markiere 20 Beispiele für BALANCE. Speichere Kontext, Delta, Preisweg, Folgereaktion und Gegenbeispiel - ohne die Gewinner zuerst auszuwählen.

ZUSTAND 2

IMBALANCE

Aggressive Orders treffen auf zu wenig Gegenliquidität; der Preis sucht schnell ein neues Niveau.



IMBALANCE		
BID	PREIS	ASK
32	2357.50	120
35	2357.00	102
28	2356.50	95
22	2356.00	84
18	2355.50	72
12	2355.00	53
DELTA		+379



BAR DELTA



BEOBACHTUNG

Mehr Ask-Ausführung, steigende Geschwindigkeit, dünne Gegenliquidität und klare Preisprogression.

Beobachtung

Mehr Ask-Ausführung, steigende Geschwindigkeit, dünne Gegenliquidität und klare Preisprogression.

Inferenz

Aggressive Käufer erzielen Resultat. Das unterscheidet echten Drive von bloß positivem Delta.

Handelslogik

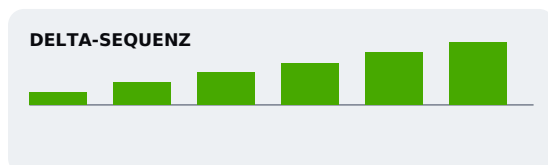
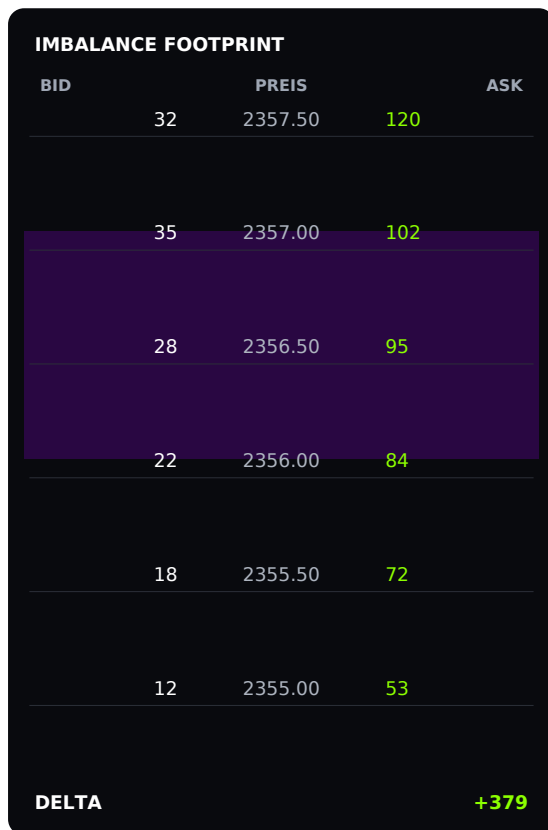
Nicht in den vertikalen Impuls springen; Retest, kurze Balance oder Pullback mit gehaltenem Level abwarten.

Nur die Beobachtung kommt direkt aus den Daten. Inferenz und Trade benötigen Alternativerklärungen, Kontext und Invalidierung.

ZUSTAND 2 - DEEP DIVE

IMBALANCE im Footprint lesen

Die gleiche Form kann in einem anderen Session- oder Profilkontext eine andere Bedeutung besitzen.



WT-9

Prüfreihenfolge

- Kontext: Wo liegt das Ereignis relativ zu Value, VPOC, VAH/VAL, Range und Sessionextrem?
- Aggression: Welche Seite handelt am Bid oder Ask, mit welchem Volumen und Tempo?
- Resultat: Wie viele Ticks Preisweg entstehen pro Volumen- oder Deltaeinheit?
- Persistenz: Bleibt die Reaktion über mehrere Events bestehen oder wird sie sofort negiert?
- Invalidierung: Welches beobachtbare Ereignis macht die Arbeitshypothese falsch?

NICHT HANDELN

Gegenbeispiel

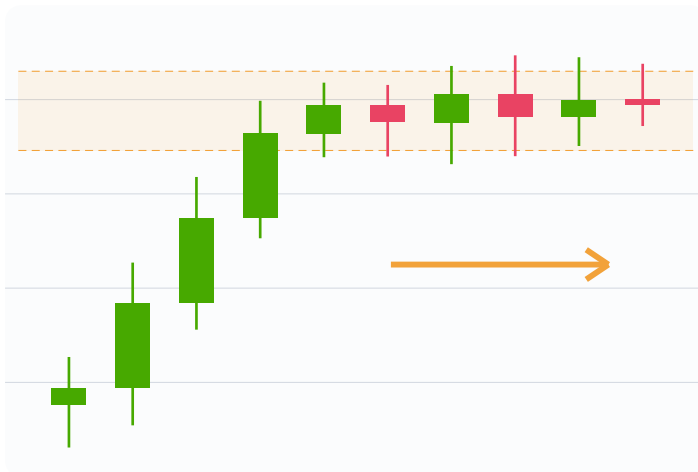
IMBALANCE kann durch News, dünne Liquidität, Roll- oder Sessionwechsel, Datenlücken oder kurzfristige Hedging-Flows nur vorgetäuscht werden.

Arbeitsauftrag: Markiere 20 Beispiele für IMBALANCE. Speichere Kontext, Delta, Preisweg, Folgeaktion und Gegenbeispiel - ohne die Gewinner zuerst auszuwählen.

ZUSTAND 3

ACCEPTANCE

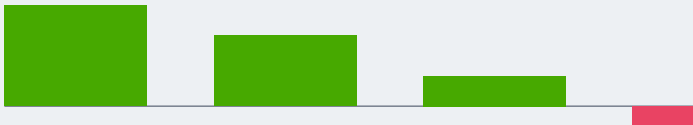
Der Markt verbringt Zeit und Volumen in einem neuen Preisbereich.



ACCEPTANCE		
BID	PREIS	ASK
68	2359.50	62
72	2359.00	69
75	2358.50	60
80	2358.00	61
82	2357.50	75
71	2357.00	69
DELTA		-52



BAR DELTA



BEOBACHTUNG

Mehrere Rotationen oberhalb der alten Range, wachsendes Volumen und kein sofortiger Rückfall.

Beobachtung

Mehrere Rotationen oberhalb der alten Range, wachsendes Volumen und kein sofortiger Rückfall.

Inferenz

Das neue Niveau wird gehandelt und verteidigt; Balance beginnt auf höherer Ebene.

Handelslogik

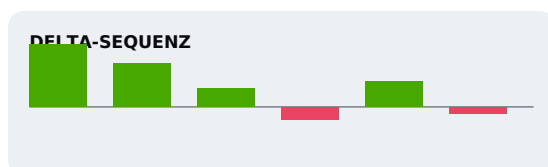
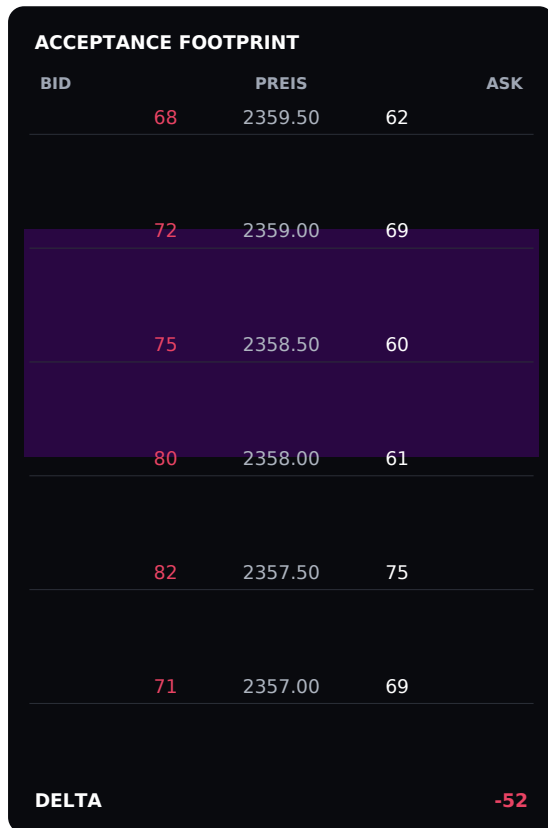
Fortsetzung eher am Rand der neuen Value suchen, nicht in der Mitte des akzeptierten Bereichs.

Nur die Beobachtung kommt direkt aus den Daten. Inferenz und Trade benötigen Alternativerklärungen, Kontext und Invalidierung.

ZUSTAND 3 - DEEP DIVE

ACCEPTANCE im Footprint lesen

Die gleiche Form kann in einem anderen Session- oder Profilkontext eine andere Bedeutung besitzen.



Prüfreihenfolge

- Kontext: Wo liegt das Ereignis relativ zu Value, VPOC, VAH/VAL, Range und Sessionextrem?
- Aggression: Welche Seite handelt am Bid oder Ask, mit welchem Volumen und Tempo?
- Resultat: Wie viele Ticks Preisweg entstehen pro Volumen- oder Deltaeinheit?
- Persistenz: Bleibt die Reaktion über mehrere Events bestehen oder wird sie sofort negiert?
- Invalidierung: Welches beobachtbare Ereignis macht die Arbeitshypothese falsch?

WT-21739545

NICHT HANDELN

Gegenbeispiel

ACCEPTANCE kann durch News, dünne Liquidität, Roll- oder Sessionwechsel, Datenlücken oder kurzfristige Hedging-Flows nur vorgetäuscht werden.

Arbeitsauftrag: Markiere 20 Beispiele für ACCEPTANCE. Speichere Kontext, Delta, Preisweg, Folgereaktion und Gegenbeispiel - ohne die Gewinner zuerst auszuwählen.

ZUSTAND 4

REJECTION

Ein Preisbereich wird getestet, aber rasch verlassen; die Auktion kehrt zurück.



REJECTION

BID	PREIS	ASK
15	2361.50	62
18	2361.00	70
26	2360.50	72
30	2360.00	85
40	2359.50	78
55	2359.00	36
DELTA		+219

VOLUME PROFILE



BAR DELTA



BEOBACHTUNG

Test des Hochs, fehlender Anschluss, Gegenaggression und schneller Rücklauf unter die Testzone.

Beobachtung

Test des Hochs, fehlender Anschluss, Gegenaggression und schneller Rücklauf unter die Testzone.

Inferenz

Höhere Preise fanden keine dauerhafte Akzeptanz. Der Rücklauf bestätigt die Ablehnung.

Handelslogik

Entry erst nach Rückkehr unter das Level oder nach fehlgeschlagenem Retest; Stop hinter dem abgelehnten Extrem.

Nur die Beobachtung kommt direkt aus den Daten. Inferenz und Trade benötigen Alternativerklärungen, Kontext und Invalidierung.

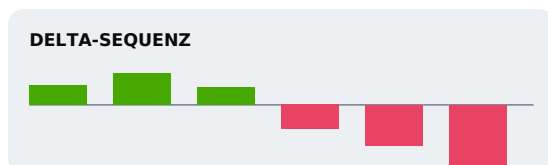
ZUSTAND 4 - DEEP DIVE

REJECTION im Footprint lesen

Die gleiche Form kann in einem anderen Session- oder Profilkontext eine andere Bedeutung besitzen.

WT-25920932

REJECTION FOOTPRINT			
BID	PREIS	ASK	
15	2361.50	62	
18	2361.00	70	
26	2360.50	72	
30	2360.00	85	
40	2359.50	78	
55	2359.00	36	
DELTA			+219



Prüfreihenfolge

- Kontext: Wo liegt das Ereignis relativ zu Value, VPOC, VAH/VAL, Range und Sessionextrem?
- Aggression: Welche Seite handelt am Bid oder Ask, mit welchem Volumen und Tempo?
- Resultat: Wie viele Ticks Preisweg entstehen pro Volumen- oder Deltaeinheit?
- Persistenz: Bleibt die Reaktion über mehrere Events bestehen oder wird sie sofort negiert?
- Invalidierung: Welches beobachtbare Ereignis macht die Arbeitshypothese falsch?

NICHT HANDELN

Gegenbeispiel

REJECTION kann durch News, dünne Liquidität, Roll- oder Sessionwechsel, Datenlücken oder kurzfristige Hedging-Flows nur vorgetäuscht werden.

Arbeitsauftrag: Markiere 20 Beispiele für REJECTION. Speichere Kontext, Delta, Preisweg, Folgeaktion und Gegenbeispiel - ohne die Gewinner zuerst auszuwählen.

ZUSTAND 5

WT-32838378

FAILED AUCTION

Ein Ausbruch scheitert und die Gegenseite übernimmt nach der Rückkehr in die vorherige Value.



FAILED AUCTION

BID	PREIS	ASK
10	2362.00	85
14	2361.50	92
18	2361.00	95
36	2360.50	90
55	2360.00	68
80	2359.50	42
DELTA		+259

VOLUME PROFILE



BAR DELTA



BEOBACHTUNG

Breakout durch ein Extrem, keine Akzeptanz, Rückkehr in die Range und anschließende Bewegung zur Gegenseite.

Beobachtung

Breakout durch ein Extrem, keine Akzeptanz, Rückkehr in die Range und anschließende Bewegung zur Gegenseite.

Inferenz

Die Breakout-Auktion ist gescheitert; gefangene Positionen können den Rücklauf verstärken.

Handelslogik

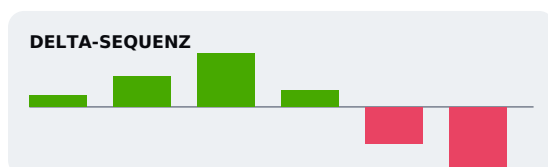
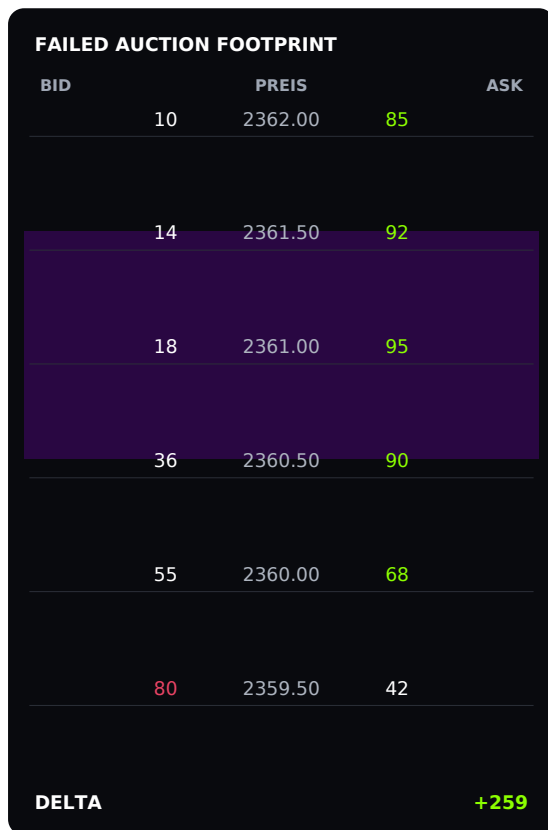
Nicht jeden Docht als Failed Auction handeln. Rückkehr, Halten innerhalb der Range und Gegenorderflow müssen zusammenpassen.

Nur die Beobachtung kommt direkt aus den Daten. Inferenz und Trade benötigen Alternativerklärungen, Kontext und Invalidierung.

ZUSTAND 5 - DEEP DIVE

FAILED AUCTION im Footprint lesen

Die gleiche Form kann in einem anderen Session- oder Profilkontext eine andere Bedeutung besitzen.



Prüfreihenfolge

- Kontext: Wo liegt das Ereignis relativ zu Value, VPOC, VAH/VAL, Range und Sessionextrem?
- Aggression: Welche Seite handelt am Bid oder Ask, mit welchem Volumen und Tempo?
- Resultat: Wie viele Ticks Preisweg entstehen pro Volumen- oder Deltaeinheit?
- Persistenz: Bleibt die Reaktion über mehrere Events bestehen oder wird sie sofort negiert?
- Invalidierung: Welches beobachtbare Ereignis macht die Arbeitshypothese falsch?

NICHT HANDELN

Gegenbeispiel

FAILED AUCTION kann durch News, dünne Liquidität, Roll- oder Sessionwechsel, Datenlücken oder kurzfristige Hedging-Flows nur vorgetäuscht werden.

Arbeitsauftrag: Markiere 20 Beispiele für FAILED AUCTION. Speichere Kontext, Delta, Preisweg, Folgereaktion und Gegenbeispiel - ohne die Gewinner zuerst auszuwählen.

WT-37808261

VERGLEICH

Die fünf Zustände in einer Matrix

Der Screenshot zeigt eine logische Folge. In realen Märkten können Zustände übersprungen, wiederholt oder falsch klassifiziert werden.

Zustand	Preis	Delta	Volumen	Bestätigung
Balance	Rotation	wechselnd	Mitte/HVN	Rückkehr
Imbalance	schneller Weg	gerichtet	mitlaufend	Progression
Acceptance	neue Range	normalisiert	baut sich auf	Zeit + Hold
Rejection	schneller Rücklauf	dreht oft	am Extrem	unter/über Level
Failed Auction	Break + Return	erst Drive, dann Flip	Breakout scheitert	alte Value hält

Typische Zustandsübergänge



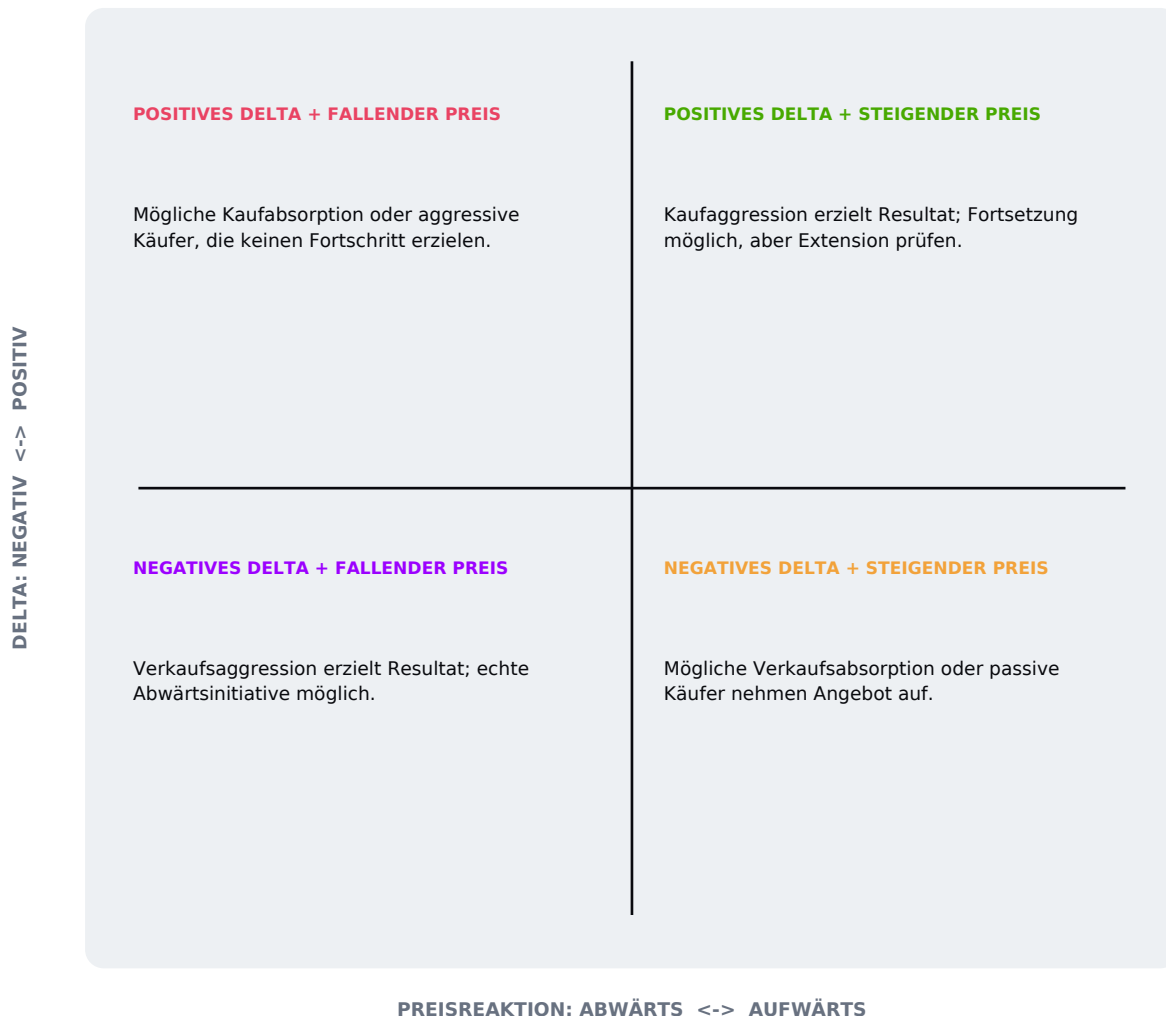
Alternative Wege: Balance -> Rejection -> Balance; Imbalance -> Rejection -> Failed Auction; Acceptance -> neue Balance. Deshalb ist eine starre Fünf-Schritt-Schablone gefährlich.

WT-65393701

EFFORT VERSUS RESULT

Delta und Preisreaktion als Vier-Felder-Modell

Die Kernaussage des Screenshots wird professioneller, wenn Aggression und Resultat getrennt gemessen werden.



Delta-Divergenz ist eine Diagnose, kein Entry. Sie benötigt Level, Reaktion, Zeitfenster und klare Invalidierung.

ENTSCHEIDUNGSPROZESS

Context -> Aggression -> Reaction -> Decision

Die Apply-this-logic-Box des Screenshots wird zu einem reproduzierbaren Ablauf erweitert.

01 CONTEXT

Markt, Kontrakt, Session, Regime, Value, Extrem, News und verfügbare Liquidität.

02 AGGRESSION

Bid/Ask-Volumen, Delta, Imbalances, Big Trades, Tape Speed und Sweep-Verhalten.

03 REACTION

Preisweg, Zeit bis Reaktion, Rücklauf, Hold, Replenishment und Gegenaggression.

04 HYPOTHESE

Balance, Imbalance, Acceptance, Rejection oder Failed Auction - mit Alternativerklärung.

05 DECISION

Entry, Ordertyp, Invalidierung, Ziel, Positionsgröße und No-Trade-Regel.

06 REVIEW

Screenshot, Rohdaten, Slippage, Regelkonformität, MAE/MFE und Ergebnis in R.

PRAXIS

ES/NQ-Umsetzung und Risikogrenzen

Die Auktionslogik ist marktübergreifend, aber Tempo, Tickwert, Slippage und sinnvolle Stopdistanz unterscheiden sich.

Punkt	ES	NQ
Charakter	tiefer, oft ruhiger	schneller, sprunghafter
Tickgröße	0,25 Indexpunkte	0,25 Indexpunkte
Screenshot-Logik	Profil + Reaktion	Tempo + Reaktion
Hauptgefahr	zu frühes Fade	Slippage/Überschießen
Manual-Setup	max. 20 Tick SL	max. 60 Tick SL
Zielrahmen	max. 40 Tick TP	max. 120 Tick TP

Vor jedem Trade

- Ist der Zustand wirklich bestätigt oder nur im Nachhinein sichtbar?
- Liegt der Entry an einer logischen Kante und ist der Stop hinter der Invalidierung?
- Reicht der freie Raum bis zum nächsten VPOC, HVN, LVN oder Sessionextrem?
- Sind Nachrichten, Sessionwechsel oder Datenprobleme ausgeschlossen?
- Ist die Positionsgröße aus Geldrisiko und tatsächlicher Stopdistanz berechnet?

Die genannten Tickgrenzen sind Obergrenzen aus dem Strategienband, keine Empfehlung für jeden Marktstatus. Wenn die logische Invalidierung weiter entfernt liegt: kein Trade oder kleinere Größe.

ABSCHLUSS

Was aus dem Screenshot gelernt werden soll

Die Grafik ist eine gute Prozesskarte, aber erst präzise Definitionen machen sie research- und handelstauglich.

WT-74539984

Aggression zeigt, wer sofort handeln will. Preisreaktion zeigt, ob diese Aggression Wirkung erzielt. Zeit und Volumen zeigen, ob ein Bereich akzeptiert wird.

Zehn Kontrollfragen

1

Was ist im Chart direkt beobachtbar?

2

Welche Aussage ist nur Interpretation?

3

Wie wurde Bid/Ask klassifiziert?

4

Welche Delta- und Imbalance-Schwellen gelten?

5

Wo liegt das Ereignis im Profil?

6

Erzielt Aggression messbaren Preisfortschritt?

7

Wie lange hält Acceptance oder Rejection?

8

Welche Alternativerklärung ist plausibel?

9

Wo ist die Hypothese invalidiert?

10

Wie wird das Ergebnis netto und in R dokumentiert?

Danke an alle Leserinnen und Leser, die Footprint nicht als Farbsignal, sondern als Messinstrument innerhalb eines kontrollierten Prozesses verwenden.

Ausgangsidee: vom Nutzer bereitgestellte Auction-Theory-/Footprint-Grafik; alle Lehrbilder wurden neu aufgebaut.

SYSTEMARCHITEKTUR

WT-13204468

Hot Path in C, Research Path in Python

Professionelle Geschwindigkeit entsteht aus deterministischer Arbeitsteilung, nicht aus beliebigem Low-Level-Code.

01 NIC / CAPTURE

Pakete erfassen, Sequenz und Hardware-/Receive-Zeit speichern.

02 DECODE

Bounds-Check, Endianness, Template und Instrument validieren.

03 L3 BOOK

Order-Lifecycle, Preispriorität, Menge und Level-Aggregat aktualisieren.

04 FEATURES

Microprice, OFI, Queue, Replenishment und Preisantwort inkrementell.

05 HANDOFF

SPSC-Ring übergibt unveränderliche Snapshots oder Events.

06 PYTHON

Deep Analytics, Backtest, Statistik und Strategy Promotion.

Hot Path: keine Allokation, kein Logging, kein Python, kein Brokerzugriff. Jeder Fehlerpfad stoppt oder markiert den Replay-Zustand deterministisch.

CPU CACHE

Cache-bewusstes Level-3-Datenmodell

Hot und Cold Data werden getrennt. Structure of Arrays verbessert lineares Scannen; 64-Byte-Ausrichtung verhindert nicht automatisch Cache Misses.

C11 | HOT/COLD SPLIT

```
01 #include <stdint.h>
02 #include <stdalign.h>
03 #define MAX_ORDERS (1u << 20)
04 #define NIL UINT32_MAX
05
06 typedef struct {
07     alignas(64) uint64_t id[MAX_ORDERS];
08     int32_t px_ticks[MAX_ORDERS];
09     uint32_t qty[MAX_ORDERS];
10     uint32_t next[MAX_ORDERS];
11     uint32_t prev[MAX_ORDERS];
12     uint8_t side[MAX_ORDERS];
13     uint8_t live[MAX_ORDERS];
14 } order_hot_soa;
15
16 typedef struct {
17     uint64_t exch_ts_ns;
18     uint64_t recv_ts_ns;
19     uint32_t instrument;
20     uint32_t flags;
21 } order_cold;
22
23 _Static_assert(alignof(order_hot_soa) >= 64,
24                "cache alignment required");
```

Entscheidung	Nutzen	Risiko
SoA	nur benötigte Felder laden	komplexere Mutation
Index statt Pointer	kompakt und serialisierbar	Generation nötig
Hot/Cold Split	kleiner Working Set	zweiter Lookup
64-B Alignment	False Sharing steuerbar	plattformabhängig

WT-49749306

Messregel: Layout nur übernehmen, wenn perf/PMU-Zähler, p50/p99-Latenz und Replay-Hash auf Zielhardware besser werden.

FEED DECODER

Bounds-sicheres binäres Decoding

Wire-Strukturen niemals blind auf C-Structs casten. Länge, Template, Sequenz und Endianness werden vor dem Book-Update geprüft.

C11 | CHECKED DECODER

```

01 #include <stdint.h>
02 #include <string.h>
03 #include <stdbool.h>
04
05 typedef struct {
06     uint64_t seq, order_id;
07     int32_t px_ticks;
08     uint32_t qty;
09     uint8_t action, side;
10 } mbo_event;
11
12 static bool read_u32le(const uint8_t **p,
13                      const uint8_t *end, uint32_t *out) {
14     if ((size_t)(end - *p) < 4) return false;
15     uint32_t v; memcpy(&v, *p, 4); *p += 4;
16     #if __BYTE_ORDER__ == __ORDER_BIG_ENDIAN__
17     v = __builtin_bswap32(v);
18     #endif
19     *out = v; return true;
20 }
21
22 bool decode_mbo(const uint8_t *p, size_t n, mbo_event *e) {
23     const uint8_t *end = p + n; uint32_t lo, hi;
24     if (!read_u32le(&p, end, &lo) || !read_u32le(&p, end, &hi))
25         return false;
26     e->seq = ((uint64_t)hi << 32) | lo;
27     /* remaining fields: same checked pattern */
28     return p <= end;
29 }

```

- Sequenzlücke: Book nicht still weiterführen; Recovery-Zustand aktivieren.
- Unbekanntes Template oder Instrument: Event quarantänisieren und Metrik erhöhen.
- memcpy vermeidet unaligned access und Strict-Aliasing-Probleme beim Lesen.
- Der Decoder kennt Protokollsemantik; die Strategie erhält nur validierte Events.

ORDER TRACKING

O(1)-Lookup mit Open Addressing

Order-ID wird auf einen stabilen Slot abgebildet. Tombstones, volle Tabellen und doppelte Adds sind explizite Zustände.

C11 | ORDER-ID INDEX

```

01 #include <stdint.h>
02 #include <stdbool.h>
03 #define CAP (1u << 20)
04 #define EMPTY 0ull
05 #define TOMB 1ull
06
07 typedef struct { uint64_t key; uint32_t slot; } bucket;
08 static bucket map[CAP];
09
10 static inline uint64_t mix64(uint64_t x) {
11     x ^= x >> 30; x *= 0xbf58476d1ce4e5b9ULL;
12     x ^= x >> 27; x *= 0x94d049bb133111ebULL;
13     return x ^ (x >> 31);
14 }
15
16 bool find_order(uint64_t id, uint32_t *slot) {
17     uint32_t i = (uint32_t)mix64(id) & (CAP - 1);
18     for (uint32_t probe=0; probe<CAP; ++probe) {
19         const bucket *b = &map[i];
20         if (b->key == EMPTY) return false;
21         if (b->key == id) { *slot=b->slot; return true; }
22         i = (i + 1) & (CAP - 1);
23     }
24     return false;
25 }

```

I1

Invariant

Jede Live-ID verweist auf genau einen Live-Slot; Slotgeneration verhindert stale references.

I2

Load Factor

Vor Session statisch dimensionieren; kein Rehash im Hot Path.

I3

Recovery

Snapshot neu aufbauen und Hash des rekonstruierten Books vergleichen.

Worst Case bleibt O(n). Kapazität, Probeverteilung und maximale Probezahl müssen unter realer Order-ID-Verteilung gemessen werden.

LIFECYCLE ENGINE

Add, Modify, Delete deterministisch anwenden

Die Book-State-Machine akzeptiert nur die nächste Sequenz und protokolliert einen reproduzierbaren Fehlerzustand.

C11 | DETERMINISTIC APPLY

```

01 typedef enum { ADD=0, MODIFY=1, DELETE=2 } action_t;
02 typedef enum { OK=0, GAP, DUP, UNKNOWN, BAD_QTY } status_t;
03
04 status_t apply_event(book *b, const mbo_event *e) {
05     if (e->seq <= b->last_seq) return DUP;
06     if (e->seq != b->last_seq + 1) return GAP;
07     b->last_seq = e->seq;
08
09     uint32_t slot;
10     switch ((action_t)e->action) {
11     case ADD:
12         if (find_order(e->order_id, &slot)) return DUP;
13         return add_order(b, e) ? OK : UNKNOWN;
14     case MODIFY:
15         if (!find_order(e->order_id, &slot)) return UNKNOWN;
16         if (e->qty == 0) return BAD_QTY;
17         return modify_order(b, slot, e) ? OK : UNKNOWN;
18     case DELETE:
19         if (!find_order(e->order_id, &slot)) return UNKNOWN;
20         return delete_order(b, slot) ? OK : UNKNOWN;
21     }
22     return UNKNOWN;
23 }

```

Status	Aktion	Trading-Freigabe
OK	Book fortführen	möglich
DUP	zählen/verwerfen	nur wenn spezifiziert
GAP	Recovery starten	gesperrt
UNKNOWN	Quarantäne	gesperrt
BAD_QTY	Protokollfehler	gesperrt

THREAD HANDOFF

SPSC-Ring mit C11-Atomics

Ein Produzent und ein Konsument vermeiden Locks. Acquire/Release schützt Sichtbarkeit; mehrere Produzenten benötigen ein anderes Design.

C11 | SINGLE-PRODUCER / SINGLE-CONSUMER

```

01 #include <stdatomic.h>
02 #include <stdalign.h>
03 #include <stdbool.h>
04 #define RING_CAP 4096u
05
06 typedef struct {
07     alignas(64) _Atomic uint32_t head;
08     alignas(64) _Atomic uint32_t tail;
09     mbo_event data[RING_CAP];
10 } spsc_ring;
11
12 bool push(spsc_ring *r, const mbo_event *e) {
13     uint32_t h = atomic_load_explicit(&r->head,
14                                     memory_order_relaxed);
15     uint32_t n = (h + 1u) & (RING_CAP - 1u);
16     if (n == atomic_load_explicit(&r->tail,
17                                 memory_order_acquire)) return false;
18     r->data[h] = *e;
19     atomic_store_explicit(&r->head, n, memory_order_release);
20     return true;
21 }
22
23 bool pop(spsc_ring *r, mbo_event *out) {
24     uint32_t t = atomic_load_explicit(&r->tail,
25                                     memory_order_relaxed);
26     if (t == atomic_load_explicit(&r->head,
27                                 memory_order_acquire)) return false;
28     *out = r->data[t];
29     atomic_store_explicit(&r->tail, (t+1u)&(RING_CAP-1u),
30                           memory_order_release); return true;
31 }

```

Backpressure ist eine fachliche Entscheidung: Drop ist für Orderbuch-Rekonstruktion meist unzulässig. Bei vollem Ring Replay stoppen oder kontrolliert degradieren.

LATENCY DISCIPLINE

CPU-Affinität, Zeitmessung und Histogramme

Ein schneller Median verdeckt Tail-Latency. p99, p99.9, Maximum, Cache Misses und Kontextwechsel gehören in denselben Bericht.

LINUX C | AFFINITY + CLOCK

```

01 #define _GNU_SOURCE
02 #include <sched.h>
03 #include <time.h>
04 #include <stdint.h>
05 #include <stddef.h>
06
07 void decode_and_apply(const uint8_t *, size_t);
08 void latency_hist_record(uint64_t);
09
10 int pin_current_thread(unsigned cpu) {
11     cpu_set_t set; CPU_ZERO(&set); CPU_SET(cpu, &set);
12     return sched_setaffinity(0, sizeof set, &set);
13 }
14
15 uint64_t mono_raw_ns(void) {
16     struct timespec ts;
17     if (clock_gettime(CLOCK_MONOTONIC_RAW, &ts) != 0) return 0;
18     return (uint64_t)ts.tv_sec * 1000000000ull +
19           (uint64_t)ts.tv_nsec;
20 }
21
22 void on_event(const uint8_t *buf, size_t n) {
23     uint64_t t0 = mono_raw_ns();
24     decode_and_apply(buf, n); /* no printf in hot path */
25     latency_hist_record(mono_raw_ns() - t0);
26 }

```

Metrik	Warum
p50	typischer Pfad
p99/p99.9	Tail und Burst
Cache Misses	Layout-/Working-Set-Problem
Context Switches	Scheduling-/IRQ-Störung

Affinität und MONOTONIC_RAW machen Messungen nicht automatisch börsensynchron. Exchange-, NIC-, Receive- und Decision-Time bleiben getrennte Clock Domains.

PYTHON DEEP ANALYTICS

Orderflow-Features aus Eventdaten

Python liest unveränderliche Replay-Dateien. Features werden ausschließlich aus Informationen berechnet, die zum Entscheidungszeitpunkt verfügbar waren.

PYTHON | MEMMAP + POINT-IN-TIME FEATURES

```

01 import numpy as np
02 import pandas as pd
03
04 DT = np.dtype([
05     ("ts", "<u8"), ("px", "<i4"), ("qty", "<u4"),
06     ("side", "i1"), ("kind", "i1"), ("bid_q", "<u4"),
07     ("ask_q", "<u4")
08 ])
09 ev = np.memmap("mbo.bin", dtype=DT, mode="r")
10 df = pd.DataFrame(ev)
11 df["signed_qty"] = np.where(df.side > 0, df.qty, -df.qty)
12 df["mid"] = df.px.astype("float64")
13 depth = (df.bid_q + df.ask_q).clip(lower=1)
14 df["ofi"] = (df.bid_q - df.ask_q) / depth
15 df["delta_1s"] = (df.set_index(pd.to_datetime(df.ts, unit="ns"))
16                  .signed_qty.rolling("1s", closed="left").sum()
17                  .to_numpy())
18 df["fwd_20"] = df.mid.shift(-20) - df.mid
19
20 # Diagnose: niemals fwd_20 als Feature verwenden
21 report = (df.groupby(pd.qcut(df.ofi, 10, duplicates="drop"))
22          .agg(n=("fwd_20", "size"),
23              mean_move=("fwd_20", "mean"),
24              hit=("fwd_20", lambda x: (x>0).mean())))

```

- closed='left' verhindert, dass das aktuelle Event unbemerkt in das Rolling Feature fließt.
- Forward Move ist nur Label/Diagnose und wird strikt von Features getrennt.
- OFI-Skalierung, Eventhorizont und Sessionfilter werden pro Markt validiert.
- Konfidenzintervalle und abhängiger Bootstrap begleiten jede Decile-Tabelle.

EVENT BACKTEST

Queue, Latenz und Partial Fills simulieren

Ein Bar-Backtest kann MBO-Strategien nicht validieren. Die Simulation muss Ereignisreihenfolge und eigene Orderzustände abbilden.

PYTHON | EVENT-DRIVEN QUEUE MODEL

```

01 from dataclasses import dataclass
02
03 @dataclass
04 class SimOrder:
05     side: int; px: int; qty: int
06     ahead: int; filled: int = 0
07
08 def on_trade(o: SimOrder, trade_px: int, trade_qty: int,
09             aggressor: int) -> None:
10     hits = (o.side > 0 and aggressor < 0 and trade_px <= o.px) or \
11           (o.side < 0 and aggressor > 0 and trade_px >= o.px)
12     if not hits or o.filled >= o.qty:
13         return
14     consumed_ahead = min(o.ahead, trade_qty)
15     o.ahead -= consumed_ahead
16     residual = trade_qty - consumed_ahead
17     o.filled += min(o.qty - o.filled, residual)
18
19 def submit(signal_ts, decision_ns, gateway_ns, book):
20     live_ts = signal_ts + decision_ns + gateway_ns
21     level = book.at(live_ts) # state after latency
22     return SimOrder(+1, level.bid_px, 3, level.bid_qty)
23
24 def pnl(fill_px, exit_px, side, fees, slippage):
25     return side * (exit_px - fill_px) - fees - slippage

```

Komponente	Konservativ	Fehler
Latency	Verteilung/Stress	Null
Queue Ahead	sichtbare Menge	sofortiger Fill
Partial Fill	Restmenge bleibt	alles oder nichts
Cancel Ahead	Venue-Regel	100% Vorteil
Costs	Fee + Slippage	Brutto-PnL

STRATEGY BUILDING

Vom Feature zur freigegebenen Strategie

Ein schneller Book Builder ist Infrastruktur. Edge entsteht erst durch testbare Hypothese, realistische Fills, stabile Statistik und kontrolliertes Risiko.

1 HYPOTHESE

Mechanismus, Markt, Session, Horizont und Nullhypothese.

2 FEATURES

Point-in-time, Einheit, Baseline und Datenvertrag.

3 SIMULATOR

Sequenz, Queue, Latenz, Partial Fill, Fees und Slippage.

4 VALIDATION

Walk-forward, Purging, Bootstrap, Regime und Kapazität.

5 PROMOTION

Replay -> Paper -> klein live -> Review; unabhängige Risk Gates.

Primärquellen

CME MDP 3.0 MBO: cmegroup.com/tools-information/webhelp/autocert-mdp3-with-mbo/
 Intel Optimization Manual: cdrdv2-public.intel.com/821614/356477-Optimization-Reference-Manual-V2-050.pdf
 Linux sched_setaffinity: man7.org/linux/man-pages/man2/sched_setaffinity.2.html
 GCC aligned attribute: gcc.gnu.org/onlinedocs/gcc/Common-Variable-Attributes.html
 NumPy memmap: numpy.org/doc/stable/reference/generated/numpy.memmap.html
 pandas rolling: pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rolling.html

WT-29473427

Codebeispiele sind kompakte Replay-/Research-Bausteine, keine zertifizierte Exchange-Anbindung und keine Live-Trading-Freigabe.