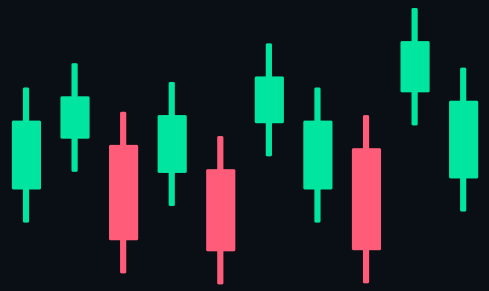
// COLLECTIVE EDUCATION · AI FOUNDERS TRACK

Building Live Trading Agents

From Prototype to Production



Stephan Rihs · Founder, Rihs Ventures

$ research >> signals >> testing >> execution >> controls

// STAY IN TOUCH

Continue the conversation

// EMAIL

stephan@rihs-ventures.com

// LINKEDIN

linkedin.com/in/dr-stephan-rihs

// SCHEDULE TIME

calendly.com/stephan-rihs-ventures

// NYC FOUNDERS & OPERATORS

Meet the community

Curated events and a weekly newsletter for founders and senior operators in New York.

foundersandoperators.nyc

bit.ly/NYCFANDO (Luma Community Calendar)

pulse.rihs-ventures.com (Newsletter)

Stephan Rihs · Rihs Ventures · New York City

// WHY BOTHER

Why algorithmic trading?

01

SYSTEMATIC

Rules run the same way every time. Repeatable, testable, improvable.

02

OBJECTIVE

No fear, no greed, no revenge trades. The rule has no feelings and no bias.

03

HANDS-OFF

The system watches the market so you don't have to. No staring at charts all day.

04

FEWER DECISIONS

Decide once, in the rules. Not two hundred times a day under stress.

05

DISCIPLINED EXITS

Stops and limits fire automatically. Losses get cut while you sleep.

06

SCALE

Screens hundreds of tickers and thousands of data points. No human can.

> *The machine brings discipline and scale.* ***You still supply the thesis and the judgment.***

// WHERE AI COMES IN

AI is the co-builder, not the trader

THINK

Brainstorm and develop your thesis

Pressure-test the edge, explore the universe, challenge your assumptions before a line of code exists.

BUILD

Write the nuts and bolts

Python scripts, API calls, backtests, cloud setup, Telegram bots. The plumbing that used to take months now takes evenings.

EXPLAIN

Commentary, synthesis and research

Summarize what moved and why, ground the numbers in narrative, surface what deserves a closer look.

// NET_EFFECT

AI democratizes the building of smart trading systems. What used to need a quant desk now needs a laptop, a thesis, and discipline. The caution and the judgment remain yours.

ABOUT ME

Markets, strategy and hands-on systems

Stephan Rihs

Founder, Rihs Ventures

New York City

I work where market structure, AI and implementation meet.

Wharton University of Pennsylvania	MBA focused on finance, entrepreneurship and investing
coinbase	Digital assets and market infrastructure
BCG The Boston Consulting Group	Structured strategy and operating transformation
Bundesministerium Finanzen	Public finance and economic policy (Austrian Treasury)
confidential · confidential	Model performance review and evaluation for leading AI labs

Logo assets: official organization sources and Wikimedia Commons

// TODAY'S AGENDA

From zero to a working trading system in five stages

01	**Observe**	Market data and portfolio state
02	**Interpret**	Alerts, summaries and research
03	**Decide**	Signals, sizing and exit rules
04	**Test**	Backtests and paper trading
05	**Execute**	Production infrastructure and controls

// AUDIENCE PULSE · SHOW OF HANDS

How far have you taken AI into the trading loop?

01

Research

Used an AI model to analyze a company, filing or market theme?

02

Signals

Defined a rule that produces a buy, sell or hold decision?

03

Bots

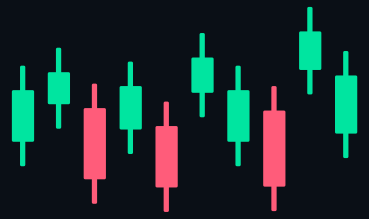
Built a script that monitors prices or sends an alert?

04

Execution

Allowed software to place a paper or live order?

No experience required. We will move from a single API request to a controlled execution system.



// SYSTEM OVERVIEW

A trading agent is a stack, not a single model

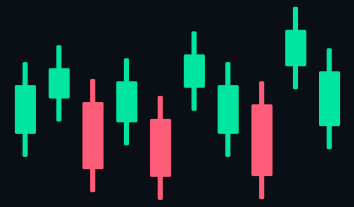
// MINIMUM_VIABLE_STACK

MARKET DATA	BOT DEV	NOTIFICATIONS	CLOUD INFRA	BROKER EXEC
yfinance for prototypes; Alpaca / Polygon for live	Python + Claude / ChatGPT / Codex	Telegram alerts, approvals, summaries	AWS EC2 + Docker or systemd, 24/7	Alpaca / IBKR / Robinhood Agentic

// RELIABILITY_LAYER

RESEARCH MODEL · SIGNAL ENGINE · RISK GATE · SECRETS · LOGS + STATE · KILL SWITCH

The LLM is one component. Reliability comes from the surrounding data, controls and infrastructure.



// STEP 0 · THESIS

Every strategy starts with a thesis

01

EDGE

Why should this make money? An observation, a pattern, an inefficiency

02

UNIVERSE

What you trade: 10 liquid large caps, one ETF, or a single asset

03

METHOD

Momentum, mean reversion, breakout or event-driven

04

HORIZON

Day trade, swing or long-term; sets data, costs and cadence

// EXAMPLE_THESIS

"Liquid tech stocks that gap down 3%+ on no news tend to recover within five sessions.
I trade the 10 most liquid names on a swing horizon."

// STEP 1 · OBSERVE

A trading agent is only as reliable as its data

// PROTOTYPE

Yahoo Finance via yfinance

Fast to explore. Convenient for notebooks. Treat it as a prototyping source, not your production contract.

// PRODUCTION_PATH

Authenticated market data

Alpaca, Polygon and other licensed feeds provide defined APIs, access controls and documented limits.

first_request.py

```python
import os, requests

headers = {
  "APCA-API-KEY-ID": os.getenv("ALPACA_KEY"),
  "APCA-API-SECRET-KEY": os.getenv("ALPACA_SECRET")
}

url = "https://data.alpaca.markets/v2/stocks/AAPL/bars"
params = {"timeframe": "1Day", "limit": 30}
bars = requests.get(url, headers=headers,
                    params=params).json()
```

docs.alpaca.markets · github.com/ranaroussi/yfinance

// STEP 1 · CLOUD SETUP

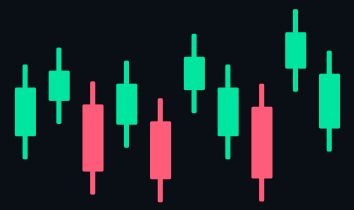
Run the algorithm 24/7, not just when your laptop is open

// AWS_EC2_EXAMPLE

LAUNCH	SET UP	SECURE	KEEP ALIVE	OBSERVE
Small Linux instance	Python, Git and dependencies	Environment variables for API keys	systemd or Docker restarts the process	CloudWatch logs and alerts

24/7 RUNTIME · $20-100 / MO · ON-DEMAND · AUTO-RESTART · REMOTE ACCESS · START SMALL

Recommended once the script matters outside working hours. Illustrative cost: $20-100/month.



// LEVERAGE THE ECOSYSTEM

Borrow the plumbing. Verify the assumptions.

// REUSE

Official SDKs + examples

alpaca-py for market data and broker integration

Research libraries

yfinance for exploration and rapid prototypes

Backtesting engines

Backtesting.py or vectorbt before building your own

// BEFORE_YOU_TRUST_A_REPO

[01] License and permitted use

[02] Recent maintenance and open issues

[03] Tests, examples and dependency quality

[04] Credential handling and network calls

[05] Data assumptions and execution logic

> *Clone it. Run the tests. Read the path that touches money. Change one thing at a time.*

github.com/alpacahq/alpaca-py · ranaroussi/yfinance · kernc/backtesting.py · polakowo/vectorbt

// STEP 2 · INTERPRET

Let the model explain the market, not control the account

// DETERMINISTIC_ALERT

"NVDA moved more than 3% since yesterday's close."

Code decides WHEN to speak.

// AI_COMMENTARY

"Summarize what changed, what is uncertain, and what deserves further research. Do not recommend a trade."

The model decides WHAT it means.

Notification layer: Telegram Bot API · Model layer: a fast, low-cost research model

// STEP 3 · CHOOSE THE GAME

Time horizon determines the strategy you can build

01

DAY TRADING

1m / 5m candles; minutes to hours; execution costs dominate

02

SWING TRADING

1h / 1d candles; days to weeks; overnight gaps matter

03

TECHNICAL

Price, volume, momentum and volatility

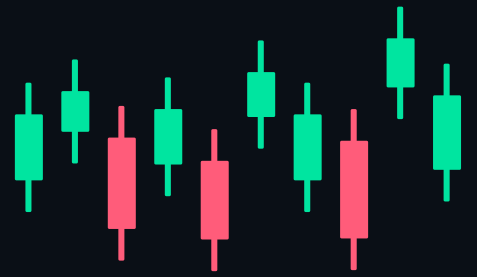
04

FUNDAMENTAL

Earnings, filings, news and macro catalysts

// CANDLE_LANGUAGE

Each candle summarizes open, high, low and close for one interval. Match the interval to the holding period.



// STEP 3 · DECIDE

A signal must be explicit enough to test

```
// TOY_SIGNAL
BUY if
  MA20 > MA50
  and volatility < 30%

SELL if
  trend reverses
  or risk limit is hit
```

[ENTRY] What must be true before a position opens?

[EXIT] What closes it, and when is the thesis invalid?

[SIZE] How much capital and risk may one position consume?

[COOLDOWN] What prevents repeated or duplicate orders?

> *The LLM may propose a hypothesis. Code must turn it into an auditable rule.*

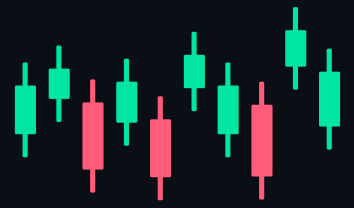
// STEP 3 · RISK MANAGEMENT

Risk management turns a signal into a tradable strategy

// RISK_MANAGEMENT_STACK

POSITION SIZE	STOP LOSS	TAKE PROFIT	TRAILING TP	PORTFOLIO LIMITS
Fractional Kelly; cap risk per trade	Hard, ATR or time-based exit	Fixed target or risk-reward multiple	Trail the peak; never loosen	Exposure, correlation and drawdown caps
1/4-1/2 KELLY	ATR / VOL	R:R TARGET	TIME STOP	MAX DRAWDOWN	KILL SWITCH

Set size, exits and failure rules before entry. Keep hard controls outside the model conversation.



// STEP 4 · TEST

A backtest is an attempt to disprove your idea

01

HISTORY

Clean prices and events

02

RULES

Same logic every time

03

FRICTION

Fees, slippage and latency

04

VALIDATION

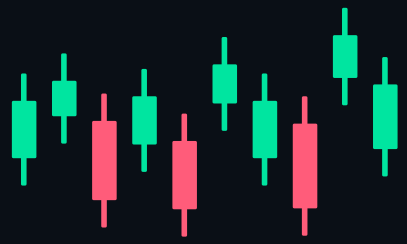
Out-of-sample and walk-forward

05

EVIDENCE

Return, drawdown, turnover, stability

If the strategy only works after repeated parameter tuning, it may be fitting history rather than finding a durable signal.



// STEP 4 · EVALUATE

A profitable backtest is not enough

01

SHARPE

Excess return / volatility

02

MAX DD

Worst peak-to-trough decline

03

WIN RATE

Share of profitable trades

04

PAYOFF

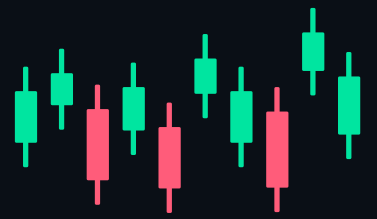
Average win / average loss

05

PROFIT FACTOR

Gross profit / gross loss

Also inspect: annualized return · drawdown duration · turnover · fees · slippage · out-of-sample stability



// PORTFOLIO INTELLIGENCE

Connect the portfolio in stages

01	**Observer** RECOMMENDED DEFAULT	Read positions, balances, watchlists and order history. Summarize exposures and concentration.
02	**Adviser**	Flag a breached rule. Explain the context. Draft an action for human approval.
03	**Executor**	Place an order only after deterministic risk checks and account-level limits pass.

> *Observe → recommend → approve. Stop-loss logic should not depend on a conversational model noticing a price move.*

Source: Robinhood Agentic Trading overview

// PUTTING IT TOGETHER

Paper trading tests the system before the capital

MARKET DATA › SIGNAL › RISK GATE › ALPACA PAPER › FILL + ALERT

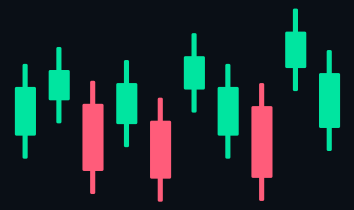
▶ LIVE DEMO

Submit one small paper order. Inspect the order state, simulated fill, resulting position and Telegram notification.

// PAPER_IS_NOT_LIVE

It cannot fully reproduce slippage, market impact, queue position or network failures. Your paper P&L will always be more generous than live.

Source: Alpaca Paper Trading documentation · docs.alpaca.markets



// FROM SCRIPT TO SYSTEM

Production is mostly controls, state and failure handling

// CLOUD_RUNTIME

SCHEDULER	DATA	SIGNAL ENGINE	RISK GATE	BROKER
When should the system run?	What is current and complete?	What rule fired, and why?	Is this order permitted?	Was it accepted or filled?

SECRETS · STATE · LOGS · ALERTS · RETRIES · KILL SWITCH

> *Every decision, request, response and order ID should be traceable after the fact.*

// STAGED ROLLOUT

Earn the right to automate real capital (3-9 months timeline)

01
RESEARCH
No orders

02
ALERTS
Human acts

03
PAPER
Simulated fills

04
SMALL LIVE
Approval required

05
BOUNDED AUTO
Hard limits

Before live execution: maximum position · daily loss limit · approved symbols · order limit · emergency stop

// WHERE TO START

Start small and simple

01 **Watchlist alerts**

Smart alerts and summaries for your favorite stocks, in Telegram or email

```
$ every day at 9:45 AM and 3:45 PM, send me a summary of the day's price action in [STOCK]
```

02 **Portfolio pulse**

The same rhythm, for the positions you actually hold

```
$ every day at 9:45 AM and 3:45 PM, send me an update on the performance of my Robinhood portfolio
```

03 **First backtest**

Test a rule you are curious about, on a few names you like

```
$ for [STOCKS], enter when the 30-day average > 50-day average; exit at +4% take profit or -2% stop loss
```

> *Then expand from here. Each of these is one prompt, one script, one afternoon.*

// Q&A

Three things to remember

01 AI is useful for research and orchestration.

02 Signals and risk limits must be explicit and deterministic.

03 You can’t ‘one-shot’ production quality. Iterate, and take your time.