

How AI agents run a trading operation

RESEARCH TOOLS, TRADING AUTOMATION AND THE BUSYWORK, HANDLED BY A TEAM OF AGENTS WHILE YOU DECIDE

Not one big chatbot with a broker key. A desk of small workers: one reads the news, one keeps the bars clean, one tests an idea on years of history and thousands of histories that did not happen, one applies the limits, one logs why. A signal is a file that waits for you. This guide shows the shape of that desk and how to build the first piece tonight.

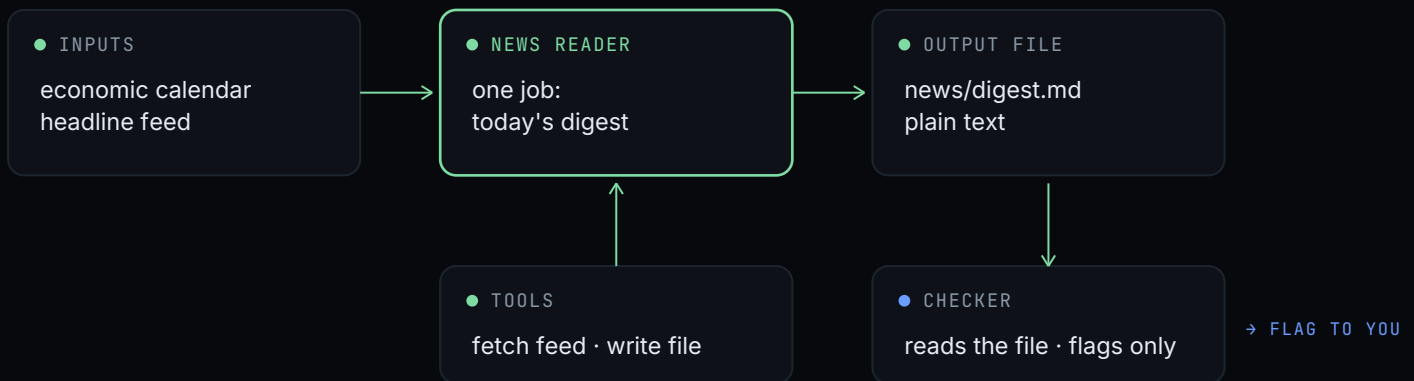


SEVEN AGENTS PROPOSE. ONE PERSON DECIDES. NOTHING SENDS AN ORDER ON ITS OWN.

PRINCE · data products and AI agents for trading · daily build logs, trading systems and research

What an agent actually is

Strip the word of its magic and an agent is five small things. A job it does, the inputs it may read, the tools it may use, a file it writes, and a checker that reads that file and can only raise a flag. Below, the five things for the first agent most traders build: the news reader.



● PART 01

One job

"Every morning, list today's scheduled events for my instruments and mark the ones inside my session." If the job needs the word "and" twice, it is two agents.

● PART 02

Inputs it may read

The calendar feed and the headline feed. Not the broker, not the bars, not your inbox. A short list, and nothing outside it.

● PART 03

Tools it may use

Fetch a feed, write a file. It cannot place, size or cancel anything. Add a tool only when the job cannot be done without it.

● PART 04

An output file

news/digest.md, same name every day. Ten lines: what is scheduled, what to avoid, what is already priced in. Files can be compared. Chat cannot.

● PART 05

A checker

A second agent reads the digest against the job: are the events real, are the times in your session's clock, is anything missing. It cannot edit. It can only flag.

The desk as a team

One big prompt that reads, tests, sizes and sends will do all four badly on a bad day and you will not know which step failed. A desk of single-job agents fails in one place, and that place has a name.



01	news reader	Reads the calendar and the headlines. Writes a short digest and marks the days to stay out.
02	data loader	Pulls the bars and keeps them clean. A missing bar or a bad tick gets flagged, not traded on.
03	researcher	Takes one idea and writes it as rules. Runs those rules on years of real bars and lists the trades.
04	stress tester	Reshuffles that trade list thousands of times. Reports the worst drop and the pass rate against your limits.
05	checker	Applies the hard limits: daily loss, max drawdown, size, session hours. Anything outside them never reaches the inbox.
06	trading floor inbox	A file, not a button. A signal lands here with the checker's note and waits for a person. It expires at session close.
07	journal	Logs every accept and every reject with the reason. Nothing is edited after. The weekly review reads this, not you.

How a trading day runs without you

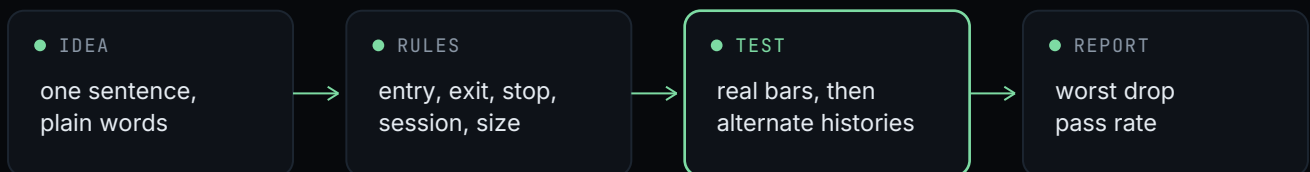
A working day on the desk. Green rows are agents. Blue rows are the only moments that need a person: the morning read and the accept step.

05:30	NEWS READER	Pre-market digest Calendar and headlines for your instruments. Ten lines. Marks the events inside your session.
06:00	DATA LOADER	Bars check Pulls overnight bars, checks gaps and timestamps, writes a one-line status. A bad feed stops the desk here.
06:30	RESEARCHER	Watchlist regimes Labels each instrument on the watchlist: trending, ranging, quiet, event day. Same words every day.
07:00	YOU	Morning read Digest, bars status, regimes. Fifteen minutes. You mark what is in play and what is off today.
session	CHECKER	Signals to inbox When your rules fire, the checker tests the signal against the limits and writes it to the inbox with a note.
session	YOU	Accept or reject You read the note, accept or reject. If you accept, you place it. Nothing else does. Unread signals expire at close.
16:30	REPORTER	Daily report Signals seen, accepted, rejected, and the limit counters against your rules. One page, same shape every day.
20:00	JOURNAL	Evening journal Every decision with your reason appended. Tomorrow's weekly review and tomorrow's reader both start here.

Nothing on this page sends an order. Agents write files. You decide what reaches the market.

AI research tools: what happens to an idea

You give the researcher one sentence. It comes back with two numbers and a report. Here is the path in between, and why the second test matters more than the first.



STEP	WHAT THE RESEARCHER DOES
Idea → rules	"Buy the first pullback after a session-open breakout" becomes entry, invalidation, exit, session window and size. If a rule cannot be written, the idea is not testable yet.
Rules → real bars	The rules run on years of real daily bars. Out comes the list of trades as they would have happened, with dates. That list is one sample of the past.
Real bars → alternate histories	The trade list is reshuffled thousands of times. Each shuffle is a past that could have happened. For each one, record the deepest fall from a high.
Histories → report	Two numbers first: the worst drop at the bad end of the list, and the pass rate: how many histories stayed inside your limits all the way. Then the details.

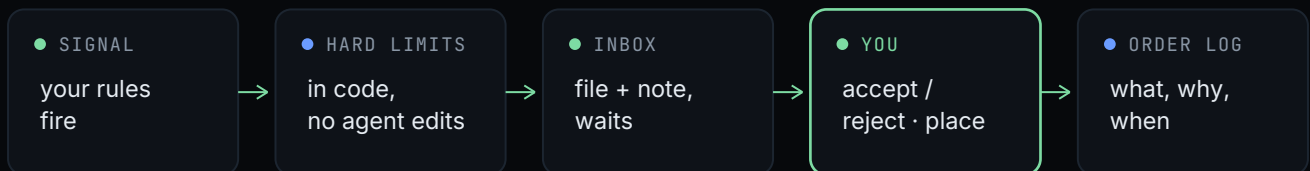
● THE PAST IS ONE SAMPLE

A backtest on real history tells you what happened once. An idea that looks fine on one history and falls apart on a thousand of them was never fine. It was lucky. The researcher's job is to find out which, before you do.

A test is a description of the past, not a forecast. It tells you how an idea would have behaved. It says nothing about next week.

Trading automation: the only path a signal can take

Automation on this desk means the busywork between a rule firing and you deciding. It does not mean an agent with a broker key. Five stops, one of them is you.



01 · SIGNAL

Your written rules fire on the clean bars. The signal is written as a file: instrument, direction, entry, stop, target, and the rule that fired. Plain text.

02 · HARD LIMITS IN CODE

Daily loss, max drawdown, size and session hours live in a config no agent can write to. The checker tests the signal against it. A signal that breaks a limit is logged and dropped. It never reaches the inbox.

03 · INBOX

The signal lands with the checker's note: what limit headroom is left, what the news reader flagged, whether this rule has failed recently. It waits. It expires at session close.

04 · YOU

You read it. Accept, reject, or ask for more. If you accept, you place the order yourself. That step is a person, every time, by design.

05 · ORDER LOG

Whatever you decided is written with the reason. Accepted and rejected alike. The journal and the weekly review read this file.

Nothing I build can send an order on its own.

That is not a setting. There is no order-sending code in the build. The market risk stays yours; the software risk is the part I remove.

The three rules that keep it safe

Every desk I build runs on the same three rules. They are not settings that can be toggled. They are the shape of the build.

● RULE 01

The human decides, agents propose

An agent's last step is always a file, never an order. Place, size, cancel, close: those verbs belong to you. The agents' job is to make your decision a two-minute read instead of a two-hour dig. If a workflow "needs" an agent to act on its own, the workflow is not ready.

AGENT → FILE → YOU → ORDER

● RULE 02

Keys and data stay on your machine

Your data keys, any broker key, your bars and your journal live in a local folder on a computer you own. Agents read from it and write to it. Nothing is uploaded to a dashboard, a third-party store or a builder's server. When I hand a build over, I hand over the folder and I keep nothing.

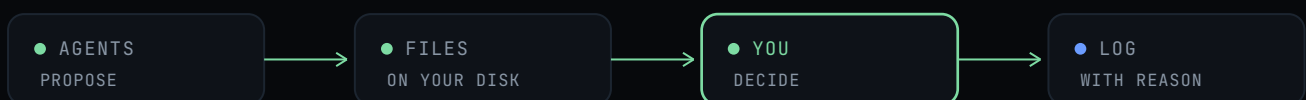
LOCAL FOLDER · YOUR KEYS · NO COPY

● RULE 03

Every action is logged with its reason

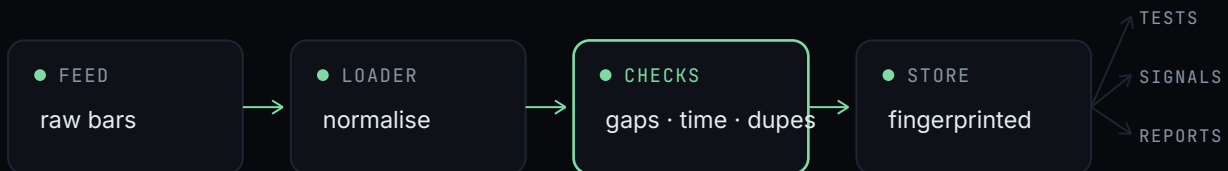
Each agent writes what it did and why to a log that no agent can edit. When a signal looks wrong you read the log, find the line, and fix that one agent. A desk you cannot audit is a desk you cannot trust, and a desk you cannot trust is one you will stop using.

WHO · WHAT · WHY · WHEN



Data products: what a clean data layer looks like

Every test, signal and report on the desk reads from one place. If that place is dirty, everything above it is fiction. Four properties make it clean.



● PROPERTY 01

Bars checked for gaps

Every session is counted. A missing bar, a duplicate bar or a bar with a zero range is written to a gaps file before anything reads the data. The desk does not trade through a hole.

● PROPERTY 02

Timestamps in one clock

Every bar carries its exchange time and your session time. Session windows are tested against the right clock, and a daylight-saving week does not silently move your open.

● PROPERTY 03

Fingerprints for every test

Each dataset carries a hash of its contents. A test report records that hash, the rules file hash and the seed. Any test can be re-run months later and produce the same numbers.

● PROPERTY 04

A journal you can export

Accepts, rejects, reasons and the limit counters live in a plain table you own. Export it to a spreadsheet, hand it to a reviewer, or feed it to the weekly review.

A test you cannot reproduce is an anecdote. The fingerprint is what turns it into a record.

Ten things a trader hands to agents first

Start at the top. Each one is a single agent with a single output file. None of them place, size or cancel anything. Tick them off as you build.

☐	01	News digest	Two lines each morning per instrument: what is scheduled, what to avoid.
☐	02	Journal	Each accept and reject written with a reason, without opening a notebook.
☐	03	Daily report	Signals seen, accepted, rejected, limit counters. One page at close, same shape every day.
☐	04	Alerts	A message when price reaches a level you wrote down, not a screen you stare at.
☐	05	Backtest queue	Drop a rules file in a folder; the report appears in another one.
☐	06	Data cleanup	A flag when a bar is missing, a timestamp is off or a feed went stale, before you trade on it.
☐	07	Weekly review	What repeated this week, good and bad, in ten lines. Reads the journal, not you.
☐	08	Watchlist regime labels	Each instrument tagged trending, ranging, quiet or event day, same words every morning.
☐	09	Research memos	One question in, one memo out, sources at the bottom. Never an opinion without a number.
☐	10	Follow-ups	Rules you said you would review, ideas parked, tests you meant to re-run. Listed, dated.

● ORDER MATTERS

The first three read your world and write it down. Once those files exist, every later agent has something to read. Build the readers before anything that touches a signal.

Your first trading agent in one evening

Seven steps, one evening, no framework needed. Any capable model with file access will do. The skeleton on the right is the whole prompt for a news-digest agent.

01 Pick one job

The news digest. It reads the world and touches nothing. Every desk starts here.

02 Write the job in one line

If you need "and" twice, split it. One line, one verb, one file.

03 List the inputs

The calendar feed and one headline feed. Not the bars, not the broker.

04 List the tools

Fetch a feed, write a file. Two tools. No browse, no send.

05 Name the output file

news/digest.md, same name every day. Say what shape it has.

06 Write the stop conditions

When it must stop and write
NEEDS-HUMAN: feed down, event time unclear, anything it cannot date.

07 Add a checker

A second prompt that reads the digest against steps 2 to 6 and writes flags only.

PROMPT SKELETON · COPY THIS

ROLE

You are the news reader on a small trading desk. You do one job and nothing else.

ONE JOB

Each morning, list today's scheduled events for my instruments and mark the ones inside my session.

INPUTS

calendar feed, headline feed. Instruments and session hours from config/setup.md. Nothing else.

TOOLS

fetch_feed, write_file. No browse. No orders. No bars.

OUTPUT

Write news/digest.md: max ten lines. Each = time (session clock) · event · instrument · in-session yes/no · one-line why it matters.

STOP WHEN

A feed is down, an event time cannot be converted to session clock, or an instrument is not in setup.md → write NEEDS-HUMAN.md and end.

CHECKER

A second agent reads digest.md against the rules above and writes flags.md. It may not edit the digest.

Run it on yesterday's calendar first. Read digest.md and flags.md side by side. Where the checker is wrong, the rules were unclear. Fix the rules, not the checker. Then run it tomorrow at 05:30.

Mistakes I see every week

None of these are model problems. They are shape problems, and every one has a small fix.

● MISTAKE 01

One prompt that trades

Reads, tests, sizes and sends in one context. Nobody can test it and nobody knows where it broke. Split it.

● MISTAKE 02

An agent with a broker key

If the last step is an order instead of a file, you have removed yourself. Put yourself back.

● MISTAKE 03

Trusting one history

A backtest on real bars is one sample. Without the alternate histories you are reading luck as skill.

● MISTAKE 04

Limits in a note, not in code

A limit the agent is asked to respect is a suggestion. A limit in a config it cannot write to is a wall.

● MISTAKE 05

Dirty bars

A gap or a wrong clock makes every test above it wrong. Check the data before you test the idea.

● MISTAKE 06

No journal

Rejects are where the learning is. If only accepts are logged, the weekly review reads half a story.

● MISTAKE 07

Starting with the signal

Build the news reader and the data loader first. The boring ones that run daily teach you the most.

● MISTAKE 08

Tests you cannot re-run

No fingerprint, no seed, no rules hash. Six months later the numbers are different and nobody knows why.

If you only fix two: put the limits in code, and give every signal a file that waits for you.

Want one built?

A research tool for one idea, a single agent for one job, or a full desk that runs your day. Built in the open, on your machine, with the three rules baked in. Here is what to send, and where.

SEND	WHAT I NEED TO HEAR
What you want	The job in plain words. "Test this idea on five years of bars." "Sort my mornings." No jargon needed.
What you use	Your data feed, platform, calendar, spreadsheets. Names of the tools, not logins.
When	Which session you trade and when you want to read the output. Pre-market is the usual answer.

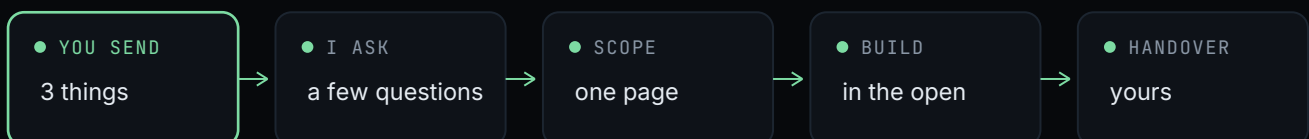
● TWO WAYS IN

1. Send the three things through the site

<https://princeofagents.app>

2. Or DM the word BUILD on Instagram

@theprinceofagents. I read every message and reply with questions, not a pitch.



No promises about results. What I promise is the build: it will do the job you described, it will run on your machine, and it will never send an order on its own.