MSTR OPTIONS RECOMMENDATION ENGINE

7-Day Self-Build Sprint

Built by You + OpenClaw Agents — No Developer Required

Prepared for Greg McKelvey
February 19, 2026
Version 2.0

CONFIDENTIAL

Contents

1. How This Works (Read This First)

Your Build Process

You already know how to do this. You built the entire PurelyBlu agent enterprise by talking to AI agents on Discord through OpenClaw. This project works exactly the same way.

You will create a new agent team on your existing OpenClaw server — an MSTR Chief Strategist on Opus 4.6, plus four sub-agents on Sonnet 4.6 for specialized analysis roles. You talk to them on Discord. They write code, build scripts, test integrations, and iterate based on your feedback. You are the product manager. They are the engineering team.

Claude (me, in this chat) serves as your AI advisor throughout the build. When you hit a wall, need architectural guidance, or want to sanity-check an approach your agents suggest, you come back here. Think of this as the strategy layer above the build layer.

Your Role vs. The Agents' Role vs. My Role

Who	What They Do
You (Greg)	Product manager. Define what you want. Test outputs. Make trading decisions. Deploy scripts to cron. Manage API keys.
MSTR Chief Strategist (OpenClaw, Opus 4.6)	Lead engineer + portfolio manager. Designs the system, writes the code, builds all scripts, and — once live — synthesizes analysis into trade recommendations.
Sub-Agents (4) (OpenClaw, Sonnet 4.6)	Specialist analysts. Macro, Technical, Options, and SRI Stage agents that produce focused analysis reports. Built and coordinated by the Chief Strategist.
Claude Advisor (This chat)	Architecture advisor. Helps you plan, troubleshoot infrastructure issues, review what your agents build, and make strategic decisions.

What You Need to Be Comfortable With

You already have all these skills from the PurelyBlu project:

- SSH into your VPS and run terminal commands
- Edit OpenClaw configuration files (openclaw.json)
- Restart services with systemctl
- Talk to agents on Discord and evaluate their output
- Copy API keys into config files or environment variables
- Set up cron jobs (your agents will write the exact crontab entries for you)

▶ **MINDSET:** You are not learning to code. You are managing a team of AI agents who code for you. The same way a CEO doesn't write marketing copy — they tell marketing what they want and review the output.

2. What This System Does

The One-Paragraph Version

An AI-powered research team that monitors MSTR and BTC markets around the clock, analyzes Bitcoin on-chain data, macro indicators, technical price action, institutional options flow, and your custom stage-based indicator framework, then delivers specific trade recommendations with exact strikes, expirations, and expected premium — every morning before market open and in real-time when opportunities appear during the trading day. You make the final call and execute trades yourself. It never trades for you.

What You Get Every Day

Morning Brief (before 7 AM ET): A concise analysis covering overnight BTC movement, MSTR pre-market action, macro conditions, current IV environment, SRI stage designation with stage transition probability, your portfolio status, and 2–3 specific trade ideas. Delivered via Telegram and viewable on a simple web dashboard.

Intraday Alerts (9:30 AM – 4 PM ET): Real-time Telegram notifications when actionable conditions appear. Example: "MSTR hit $125 support. IV at 89th percentile. SRI showing Stage 1 precursors. Recommend: Sell $120/$100 bull put spread, Mar 21 expiry, ~$3.80 credit. Probability of profit: 72%."

End-of-Day Recap (after 4:15 PM ET): Summary of what happened, how your positions changed, P&L attribution, SRI indicator state changes, and what to watch tomorrow.

On-Demand Strategy Discussion (anytime): Message your MSTR Chief Strategist on Discord anytime. "Should I roll my March puts?" "What happens to my positions if BTC drops 15%?" "Give me an SRI stage report." It has full context of your portfolio and market conditions.

What It Will NOT Do

- Place trades automatically — every trade requires your manual execution
- Guarantee profits — it improves your odds, it does not eliminate risk
- Replace your judgment — you are always the final decision-maker

3. The Hybrid Architecture: OpenClaw + Cron

This is the most important architectural concept in the entire project. Your system has two layers that serve different purposes and run differently.

Layer 1: OpenClaw Agents (Interactive)

These are your Discord-based agents. They serve two purposes:

During the build phase: They are your engineering team. You tell the MSTR Chief Strategist what you want, and it writes Python scripts, designs the database, builds API integrations, and tests everything in its sandbox workspace.

After launch: They are your on-demand advisors. You message them on Discord for strategy discussions, what-if scenarios, portfolio reviews, or ad-hoc analysis. They can read from the system's database to give you informed, context-aware answers.

OpenClaw agents run inside ephemeral Docker containers. This means they cannot run persistent background processes, maintain long-lived database connections, or wake themselves up on a schedule. They activate when you (or another agent) messages them and shut down when the conversation ends.

Layer 2: Cron Scripts (Automated)

These are Python scripts that run directly on your VPS as scheduled cron jobs — outside of OpenClaw entirely. Your agents write these scripts for you during the build phase. Once deployed, they run automatically without any human or agent intervention.

What the cron scripts handle:

- Pre-market data collection (6:00 AM ET daily): Pull overnight data from ORATS, Glassnode, FRED, Unusual Whales
- Pre-market analysis (6:15 AM ET): Send data to Claude API for each sub-agent's analysis, then synthesize via Chief Strategist prompt, save Morning Brief to database
- Morning Brief delivery (6:45 AM ET): Push brief to Telegram and web dashboard
- Intraday monitoring loop (9:30 AM – 4:00 PM ET, every 5 min): Check MSTR price levels, IV changes, SRI indicator state, and alert thresholds
- After-hours recap (4:15 PM ET): Generate end-of-day summary, update portfolio P&L

These scripts call the Claude API directly using your Anthropic API key — they do not go through OpenClaw. This means the automated pipeline runs independently. Even if OpenClaw is restarting or your Discord is offline, the cron scripts still collect data, run analysis, and send you alerts.

How the Two Layers Connect

Both layers read from and write to the same SQLite database on your VPS. The cron scripts write market data, analysis results, and trade recommendations. The OpenClaw agents read this data when you ask them questions on Discord. This shared database is the bridge between the automated pipeline and your interactive agents.

Layer	Runs On	Triggered By	Purpose
OpenClaw Agents	Docker containers via OpenClaw	You messaging on Discord, or agent-to-agent delegation	Build the system, on-demand strategy discussion, ad-hoc analysis
Cron Scripts	Directly on VPS (bare metal)	Cron schedule (time-based)	Automated data collection, scheduled analysis, alert delivery
SQLite Database	VPS filesystem	Read/written by both layers	Shared state: market data, portfolio, analysis, trade log, SRI indicator history
Claude API	Anthropic's cloud	Called by both layers	AI reasoning for all analysis

▶ **REMEMBER:** OpenClaw = your command center and build tool. Cron scripts = your autopilot. They work together through a shared database, but run independently.

4. Your Agent Team (5 Agents)

Five agents, each with a specific job. The Chief Strategist is the only one that produces final trade recommendations. The four sub-agents produce specialized analysis that feeds into the Chief Strategist's synthesis. The SRI Stage Analyst is deliberately isolated from the other three sub-agents so that its framework does not constrain their independent analysis.

Agent	Role	Model	OpenClaw Name
Chief Strategist	Synthesizes all reports, makes final trade calls, holds portfolio context	Claude Opus 4.6	mstr-strategist
Macro & Crypto Analyst	BTC on-chain, macro regime, Fed policy, risk-on/risk-off	Claude Sonnet 4.6	mstr-macro
Technical Analyst	MSTR price action, support/resistance, trend, momentum	Claude Sonnet 4.6	mstr-technical
Options Yield Strategist	IV analysis, spread construction, Greeks, trade structures	Claude Sonnet 4.6	mstr-options
SRI Stage Analyst	Stage designation via replicated Pine Script indicators, transition signals	Claude Sonnet 4.6	mstr-sri

Chief Strategist (mstr-strategist)

This is the brain of the operation. It has two modes:

Automated mode (via cron scripts): Receives structured JSON analysis from all four sub-agents plus your portfolio state. Produces the Morning Brief, evaluates intraday alert triggers, and generates the End-of-Day Recap. In this mode, it's called via the Claude API with a detailed system prompt that includes your trading rules, risk parameters, and portfolio context.

Interactive mode (via Discord/OpenClaw): You message it directly for strategy conversations. It reads the latest data from the database and gives you an informed answer.

What it knows (injected into every automated call):

- Your current MSTR positions: shares, options (with strikes, expirations, entry prices)
- Your STRK/STRF preferred share holdings if applicable
- Cash available for new trades and current margin usage
- Your risk tolerance rules (max position size, max loss per trade, max portfolio delta)
- Every past recommendation and its outcome (for self-improvement)
- All four sub-agent reports from the current analysis cycle
- The current SRI stage designation and transition probability

What it outputs:

- Morning Brief: Market thesis, SRI stage context, 2–3 specific trade ideas (exact strikes, expirations, expected credit), position management suggestions, key watch levels
- Intraday Alerts: Concise, actionable — what happened, what it recommends, risk/reward numbers
- Position Management: Roll, close, or adjust suggestions when conditions change
- End-of-Day Recap: P&L summary, SRI state changes, what to watch tomorrow

Macro & Crypto Analyst (mstr-macro)

MSTR is a leveraged Bitcoin proxy. This agent determines whether the macro environment favors risk-on or risk-off positioning.

Source	Specific Data	Frequency
Glassnode	STH MVRV, exchange net flows, SOPR, miner revenue, realized profit/loss	Hourly
FRED	10Y Treasury yield, 2Y yield, DXY, Fed funds rate	Daily
SEC EDGAR	MSTR 8-K filings (BTC purchases), convertible note updates	Real-time polling
Claude Web Search	Saylor statements, BTC ETF flow headlines, FOMC commentary	Pre-market daily

Outputs (structured JSON):

- Macro Regime: Risk-On / Risk-Off / Neutral with confidence score (0–100)
- BTC Directional Bias: Bullish / Bearish / Neutral with top 3 supporting signals
- Key Risk Factors: Events that could flip the thesis in 24–48 hours
- Saylor Activity Flag: New 8-K filings, BTC purchase announcements, convertible note activity

Technical Analyst (mstr-technical)

Focuses purely on MSTR's price chart to identify where to place option strikes.

Source	Specific Data	Frequency
ORATS	MSTR underlying price, daily OHLCV history	Intraday (1-min)
Computed locally	RSI, MACD, Bollinger Bands, 50/100/200 MAs, ATR, volume profile	Derived from price data

Outputs:

- Trend Direction: Uptrend / Downtrend / Sideways with strength rating
- Key Support Levels: 2–3 specific prices where put selling makes sense
- Key Resistance Levels: 2–3 specific prices where call selling makes sense

- Momentum: Oversold / Neutral / Overbought with RSI and context

Options Yield Strategist (mstr-options)

The quantitative core. Looks at actual options chain data and constructs specific trades.

Source	Specific Data	Frequency
ORATS Live Intraday	Full chain: bid/ask, smoothed Greeks, IV for every strike/expiry, IV forecasts, skew, term structure	Minute-by-minute
ORATS Core Data	IV rank, IV percentile, 500+ proprietary indicators	Daily + intraday
Unusual Whales	MSTR options flow: sweeps, blocks, dark pool prints	Real-time
Technical Analyst output	Support/resistance levels for strike anchoring	Per analysis cycle
SRI Stage Analyst output	Stage designation for directional bias	Per analysis cycle

Outputs:

- IV Environment: IV rank/percentile, premium rich or cheap, recommended strategy bias
- Specific Trade Ideas: 2–3 fully specified spreads with exact strikes, expirations, credits, max profit/loss, breakeven, probability of profit, Greeks
- Institutional Flow Summary: Smart money activity in MSTR options
- Roll/Adjustment Triggers: Specific conditions for managing existing positions

Strategy library:

- Bull Put Spreads — income when bullish
- Bear Call Spreads — income when bearish/neutral
- Iron Condors — income when range-bound
- Cash-Secured Puts — sell puts where you'd buy shares
- Covered Calls — sell calls against existing shares
- Big Lizards / Jade Lizards — advanced premium collection
- LEAP Hedges — long-dated put protection
- Calendar/Diagonal Spreads — term structure exploitation

5. The SRI Stage Analyst — Deep Dive

Why This Agent Is Separate

The SRI framework is a proprietary, stage-based indicator system built by your brother in TradingView. It represents a specific analytical thesis about how markets move through four stages: Accumulation (Stage 1), Markup (Stage 2), Distribution (Stage 3), and Markdown (Stage 4). It uses three interlocking indicators to identify where you are in this cycle and when transitions are likely.

This agent is deliberately isolated from the other three sub-agents because you do not want the SRI framework to constrain their independent analysis. The Technical Analyst does its own RSI/MACD work without knowing about stages. The Macro Analyst reads Glassnode without SRI bias. The Options Strategist builds spreads based on IV and Greeks independently. The Chief Strategist then synthesizes all five perspectives — including the SRI agent's stage read — and makes the final call.

Over time, as backtesting validates the SRI framework's accuracy, this agent's signal will carry increasing weight in the Chief Strategist's synthesis. But it earns that authority through data, not assumption.

The Three Indicators (Replicated in Python)

Indicator 1: Stage & Reversal Indicator (SRI)

The primary overlay indicator. Plots directly on the price chart and combines two sub-components:

Reversal Bands: Linear regression over a configurable length (default 50 bars) with standard deviation channels. Plots three lines: Support (red, regression minus deviation), Robust Fit (blue, the regression line itself), and Resistance (green, regression plus deviation). Key asymmetry: when RSI drops below 50, the Resistance line collapses down to the Robust Fit line, creating a bearish squeeze effect. These bands define the mean-reversion envelope and stage boundaries.

Fast & Slow Tracklines: ATR-based adaptive trailing stops. The Fast Trackline captures short-term momentum; the Slow Trackline identifies the persistent trend. Each trackline flips between bull mode (trailing below price, stepping up) and bear mode (trailing above price, stepping down) based on whether RSI is above or below 50. Colors indicate direction: green = rising, red = falling, orange = flat.

The SRI runs at four timeframe settings simultaneously:

Timeframe	Reversal Band	Fast Trackline	Slow Trackline	Purpose
Very Short	2 hours	2 hours	1 day	Precision entry/exit timing

Short	4 hours	2 hours	1 day	Trade construction/strategy (PRIMARY for execution)
Long	1 day	4 hours	1 week	Stage designation (PRIMARY for direction)
Very Long	2 days	8 hours	2 weeks	Lagging stage confirmation

Stage transition detection (from the Pine Script boolean rulesets):

Transition	Key Conditions (ALL required)	Pine Script Confidence
Stage 4→1 (Bottoming)	Slow sideways/down + Fast/Slow crossover + Stoch K/D crossover + Support > Robust Fit + Price bounces off support + No lingering below support + MTF RSI < 30	87%
Stage 1→2 (Bull Entry)	Slow sideways + Fast > Slow + Stoch K > 80 + Robust/Support > Slow TL + Close crosses above resistance + MTF RSI > 30 and rising	89%
Stage 2→3 (Topping)	Slow decelerating/sideways + Fast/Slow crossunder + Stoch D/K crossunder + Resistance < Robust + Price rejects resistance + MTF RSI > 70	88%
Stage 3→4 (Bear Entry)	Slow sideways + Fast < Slow + Stoch D < 20 + Resistance/Robust < Slow TL + Close crosses below support + MTF RSI falling and < 70	93%

▶ **IMPORTANT:** The confidence percentages are estimates from the Pine Script code, not statistically validated. Validating them through backtesting is one of this agent's primary Week 2 objectives.

Indicator 2: SRI Bias Oscillator (SRIBI)

A non-overlay histogram that distills the SRI's signals into a composite score ranging from approximately -110 to +110. It aggregates five scoring categories:

Category	Max Points	What It Measures
Trackline Colors	±30	Fast + Slow Trackline direction (green = +15, red = -15 each)
Slopes	±35	Slow slope magnitude (+25/-25) + Fast slope relative to Slow (+10/-10)
Crosses	±30	Fast/Slow crossover (+20/-20) + Price bounce/reject at bands (+10/-10)
Band Position	±15	Close vs. Robust Fit (+10/-10) + Band curve direction (+5/-5)
Linger Penalty	±5	Penalizes extended time below support or above resistance

Key signal interpretation (note: colors are inverted from conventional meaning):

- Bright red bars: Bearish exhaustion — favorable entry prices approaching (buy signal territory)
- Bright green bars: Bullish exhaustion — profit-taking and risk reduction warranted (sell signal territory)

- Zero-line cross (negative to positive): Confirms bullish trend reversal, especially after bright red bars
- Zero-line cross (positive to negative): Confirms bearish trend reversal, especially after bright green bars
- Bullish divergence (higher SRIBI lows while price makes lower lows): Weakening bearish trend
- Bearish divergence (lower SRIBI highs while price makes higher highs): Weakening bullish trend

SRIBI has its own timing table (offset from SRI settings):

Timeframe	Reversal Band	Fast Trackline	Slow Trackline
Very Short	8 hours	2 hours	1 day
Short	1 day	8 hours	3 days
Long (Primary)	3 days	1 day	1 week
Very Long	1 day	2 days	2 weeks

Indicator 3: STH-MVRV Replica

A price-based proxy for the Short-Term Holder Market Value to Realized Value ratio. It divides the current price by a 155-day SMA (approximating the average cost basis of recent buyers). This provides macro cycle context that frames the shorter-term SRI and SRIBI signals.

Zone interpretation:

Zone	MVRV Range	Color	Stage Implication
Deep Discount	< 0.5	Dark Green	Strong Stage 1 bottoming signal. Accumulate aggressively.
Moderate Loss	0.5 – 1.0	Light Green	Stage 4 markdown or early Stage 1 recovery.
Breakeven	~1.0	Green	Cross above 1.0 often signals Stage 1→2 transition.
Moderate Profit	1.0 – 1.5	Orange	Stage 2 markup territory. Healthy gains.
High Profit	1.5 – 2.0	Red-Orange	Stage 2 late / Stage 3 early. Caution warranted.
Extreme Profit	> 2.0	Dark Red	Stage 3 topping. High drawdown risk. Protect profits.

Critical rules from the tutorial:

- Stage 1 must begin in a dark or light green zone (MVRV < 1.0)
- Stage 2 can begin in green but ends in at least yellow/orange
- The higher the MVRV peak, the longer Stages 3 and 4 will take to resolve
- Stage 2 Consolidation patterns show muted MVRV peaks and troughs vs. true Stage 3
- Stage 4 Consolidation can masquerade as Stage 1 — watch for MVRV not reaching dark green

How the Indicators Work Together

The three indicators form a hierarchy: STH-MVRV provides the macro cycle context (where are we in the big picture?), the SRI provides the structural stage designation (which specific stage are we in?), and the SRIBI provides momentum confirmation (is the current stage accelerating, exhausting, or reversing?). The case studies from your brother's tutorial demonstrate this interplay:

- Case 1 (Stage 1 confirmation): MVRV dips into dark green → SRI shows price recovering across support and robust fit lines with Reversal Bands rounding → SRIBI confirms with zero-line cross. Double-bottom pattern typical.
- Case 2 (Stage 2 breakout): Price and Reversal Band lines cross above Slow Trackline → Slow Trackline begins green-orange alternation instead of red-orange.
- Case 3 (Stage 2 consolidation vs. Stage 3): Key tell is Slow Trackline color. Green-orange alternation = consolidation (bullish). First red appearance = likely Stage 3 (bearish). MVRV shows muted peaks during consolidation.
- Case 4 (Stage 3 confirmation): Only confirmed when consolidation fails and Stage 4 begins. Price rejected by Slow Trackline, Reversal Band lines fail to break above it, SRIBI shows multiple zero-line crossovers while price stays below Slow TL.
- Case 5 (Stage 4 confirmation): Reversal Band lines cross below Slow Trackline → price rejected by Slow TL on attempt to recover → Reversal Bands show rounding (further weakness) → SRIBI confirms with zero-line cross.
- Case 6 (Stage 4 consolidation): Mimics Stage 1 but breaks down instead of up. Clues: no recovery above Slow TL, no SRIBI zero-line cross, MVRV hasn't reached dark green.

Current MSTR Status (per Feb 16, 2026 Report)

The latest market structure report (generated by Grok from your brother's framework) places MSTR in Stage 4 (Markdown), entered approximately December 2025. Key readings:

- MSTR price: ~$133–$134, down from $450+ peaks. Sharp correction.
- SRI: 120–155 range, declining from January peaks. Stabilizing near 123–155 with some smart-money retention signals.
- SRIBI: 96–162, structurally positive longer-term but waning. SRI frequently below SRIBI = bearish short-term overlay.
- STH-MVRV: 113–181, compressed from 340+ peaks. Moving toward breakeven — typical of markdown exhaustion but not yet confirmed bottom.
- Stage 1 entry requires 4–5 of 6 conditions met. Currently NONE fully met.
- Your assessment: Between Stage 4 and Stage 1 (watching for bottoming confirmation).

What This Agent Does (Three Phases)

Phase A — Replication (Sprint Week, Day 3): Replicate all three indicators in Python using ORATS MSTR price data and Polygon/public BTC price data. The Pine Script math (linreg, stdev, RSI, ATR, RMA, stochastic, SMA) has direct Python equivalents in pandas-ta or ta-lib. The agent writes the code, you compare outputs against your TradingView charts to verify they match. Once verified, the system calculates indicators independently — no TradingView dependency.

Phase B — Backtesting (Week 2+): Replay 3–5 years of historical BTC and MSTR daily data. Calculate all indicators at each historical point. Apply the stage transition rules algorithmically. Compare predicted stages against actual subsequent price movement. Output: validated hit rates for each transition, average return by stage, average stage duration, false positive rates, and which of the six Stage 1 entry conditions matter most.

Phase C — Live Signals (Ongoing): Generate real-time stage designations and clear regime signals. Daily structured output: current stage, transition probability (now based on actual backtest data), direction of each sub-indicator, and a simple regime classification: Accumulate / Hold / Reduce / Hedge. This feeds into the Chief Strategist alongside the other three sub-agents' reports.

Python Replication Map

Every Pine Script function has a direct Python equivalent:

Pine Script Function	Python Equivalent	Library
ta.linreg(close, length, 0)	numpy.polyfit or scipy.stats.linregress	numpy / scipy
ta.stdev(close, length)	pandas .rolling(length).std()	pandas
ta.rsi(close, length)	ta.rsi() or pandas_ta.rsi()	ta-lib / pandas-ta
ta.rma(source, length)	pandas .ewm(alpha=1/length).mean()	pandas
ta.tr(true)	ta.true_range() or manual calc	ta-lib / pandas-ta
ta.stoch(close, high, low, length)	ta.stoch() or pandas_ta.stoch()	ta-lib / pandas-ta
ta.sma(source, length)	pandas .rolling(length).mean()	pandas
ta.mfi(hlc3, length)	ta.mfi() or pandas_ta.mfi()	ta-lib / pandas-ta
ta.crossover(a, b)	(a > b) & (a.shift(1) <= b.shift(1))	pandas
ta.crossunder(a, b)	(a < b) & (a.shift(1) >= b.shift(1))	pandas
request.security(ticker, tf, expr)	Resample OHLCV to target timeframe, compute, merge back	pandas resample
nz(value[1])	value.shift(1).fillna(0)	pandas

▶ **HARDEST PART:** The most complex piece to replicate is request.security() — Pine Script's multi-timeframe function. In Python, this means resampling your OHLCV data to the target timeframe (e.g., 1D → 1W), computing the indicator on that resampled data, then merging the result back to your base timeframe. Your SRI agent will handle this, but verify the outputs carefully against TradingView.

Structured Output Format

The SRI Stage Analyst outputs this JSON on every analysis cycle:


```
{
  "timestamp": "2026-02-19T06:15:00Z",
  "ticker": "MSTR",
  "current_stage": "Stage 4 (Markdown)",
  "stage_confidence": 0.82,
  "transition_watch": {
    "next_likely": "Stage 4 → Stage 1",
    "conditions_met": 1,
    "conditions_required": 5,
    "conditions_detail": {
      "structural_price_confirmation": false,
      "sri_sribi_momentum_turn": false,
      "sth_mvrv_capitulation": "approaching",
      "volume_volatility_profile": false,
      "macro_liquidity_backdrop": "mixed",
      "mstr_specific_catalysts": false
    }
  },
  "indicators": {
    "sri_long": { "support": 128.5, "robust_fit": 135.2,
      "resistance": 135.2, "fast_tl": 131.0,
      "slow_tl": 142.8, "slow_tl_color": "red" },
    "sribi_long": { "score": -22, "zone": "mild_bear",
      "zero_cross_recent": false },
    "sth_mvrv": { "value": 0.94, "zone": "light_green",
      "trend": "compressing_toward_breakeven" }
  },
  "regime": "Hedge",
  "regime_rationale": "Stage 4 dominant. MVRV approaching\n    breakeven but no
confirmed bottom. Stay defensive.\n    Monitor for Stage 1 precursors."
}
```

6. Data Stack (Locked)

These decisions are final. Every provider has been selected for data quality, API accessibility, and relevance to MSTR options trading.

Provider	Category	Plan	Monthly Cost	API Type
ORATS	Options analytics + MSTR price	Live Intraday API	$399	REST, JSON, API key
Glassnode	BTC on-chain metrics	Advanced	$29	REST, JSON, API key
Unusual Whales	Institutional options flow	Annual ($448/yr)	$37	REST, JSON, API key
FRED	Macro indicators	Free tier	$0	REST, JSON, free key
SEC EDGAR	MSTR corporate filings	Free	$0	REST, JSON, user-agent
Anthropic Claude API	AI reasoning engine	Pay-per-token	$75–$150	REST, JSON, API key
		TOTAL	$540–$615	

ORATS — Options Analytics Engine ($399/month)

Your primary and most important data source. The Live Intraday plan gives you minute-by-minute options data with ORATS' proprietary smoothing and the underlying stock price.

What you get:

- 1,000,000 API requests per month
- Intraday strikes chain with smoothed bid/ask, Greeks, theoretical values for every strike and expiry
- IV rank and IV percentile (current vs. trailing 30-day and 1-year)
- Proprietary IV forecasts and slope forecasts
- Skew and kurtosis indicators for the IV surface
- Core data: 500+ indicators per ticker
- Full history back to 2007 for backtesting
- MSTR underlying stock price included in options data responses

Signup: orats.com/data-api → Live Intraday API ($399/mo). Sign live data agreements. Docs at docs.orats.io

Glassnode — BTC On-Chain Intelligence ($29/month)

Gold standard for Bitcoin on-chain analytics. Advanced plan gives hourly resolution.

Key metrics:

- Short-Term Holder MVRV — market price vs. average purchase price of recent buyers
- Exchange Net Position Change — BTC moving on/off exchanges
- SOPR — are people selling at profit or loss?
- Miner Revenue & Hash Rate — miner economics and selling pressure
- Realized Profit/Loss — aggregate network profit-taking or loss realization

Signup: glassnode.com → Advanced ($29/mo). API at studio.glassnode.com/settings/api

Unusual Whales — Institutional Options Flow ($37/month)

Shows what institutional money is doing in MSTR options in real-time.

- MSTR-filtered options flow with direction, size, premium
- Dark pool prints, sweep alerts, net call/put premium flow
- Annual plan required for full API access ($448/year)

Signup: unusualwhales.com/pricing → Annual plan. API docs at api.unusualwhales.com/docs

FRED — Macro Data (Free)

Key series: DGS10, DGS2, DTWEXBGS, FEDFUNDS, CPIAUCSL. Signup: fred.stlouisfed.org/docs/api/

SEC EDGAR — MSTR Filings (Free)

EDGAR Full-Text Search API (efts.sec.gov). Filter for MSTR CIK. 10 req/sec. No key needed.

Claude API — Reasoning Engine ($75–$150/month)

Model	Used For	Input/1M Tokens	Output/1M Tokens
Claude Opus 4.6	Chief Strategist	$5.00	$25.00
Claude Sonnet 4.6	All four sub-agents	$3.00	$15.00

▶ **COST SAVER:** Prompt caching is critical. Your sub-agents use the same system prompt every call — cache it once and subsequent calls cost 90% less on the input side.

7. MSTR-Specific Intelligence

The mNAV Ratio

mNAV = MSTR Market Cap ÷ (BTC Holdings × BTC Price). Tells you whether MSTR is trading at a premium or discount to its Bitcoin value.

- High mNAV (e.g., 3.0x) = stock expensive vs. BTC → favor selling calls, reduce long exposure
- Low mNAV (e.g., 1.2x) = stock cheap vs. BTC → favor selling puts, add long exposure
- Calculated real-time using MSTR shares outstanding × price ÷ (BTC holdings × BTC price)
- BTC holdings updated when MSTR files 8-K purchases (detected via SEC EDGAR)

Saylor Purchase Patterns

Predictable sequence: announce convertible note → price offering → buy BTC over 1–2 weeks → file 8-K. Each stage creates tradeable dynamics. System monitors EDGAR and Saylor's public statements.

Convertible Note & Preferred Share Mechanics

Multiple convertible notes with specific conversion prices create delta hedging dynamics. STRK/STRF preferred shares have fixed dividend requirements tied to Saylor's BTC yield strategy. System tracks conversion prices, maturities, and dividend coverage ratios.

IV Regime Adaptation

- Ultra-high IV (90th+ percentile): Aggressively sell premium. Wide spreads. Iron condors.
- High IV (70th–90th): Standard credit spreads at key levels.
- Normal IV (30th–70th): Selective selling. Calendars/diagonals for term structure exploitation.
- Low IV (below 30th): Buy protective puts cheaply. LEAP hedges. Reduce selling.

8. Daily Workflow: What You Experience

6:45 AM — Your Phone Buzzes

Telegram notification. Morning Brief:

```
┌──────────────────────────────────────────
MSTR MORNING BRIEF — Feb 19, 2026

STAGE: Stage 4 (Markdown). 1 of 5 Stage 1
conditions met. MVRV at 0.94 (light green).
Watching for capitulation + structural recovery.

THESIS: Cautiously bullish. BTC reclaimed $97K.
Glassnode STH-MVRV at 1.08. DXY weakening.

MSTR pre-market: $136 (+1.8%). mNAV: 1.6x.
IV Rank: 78th percentile (rich, sellable).

TRADE IDEAS:
1. Sell $120/$100 bull put spread, Mar 7 expiry
   Credit: ~$3.80 | Max loss: $16.20 | PoP: 74%
   Stage rationale: Support zone holds if Stage 1
   develops. Aggressive if Stage 4 continues.
2. Sell $160/$180 bear call spread, Mar 7 expiry
   Credit: ~$2.90 | Max loss: $17.10 | PoP: 79%

FLOW: Unusual Whales shows large Mar 21 $115 put
selling yesterday ($1.8M premium). Bullish signal.
└──────────────────────────────────────────
```

1:47 PM — Intraday Alert

```
⚠ MSTR ALERT: Price dropped to $128. SRI Long TF
support at $128.5. SRIBI score shifted to -35.
MVRV now 0.91 — approaching deep discount zone.

Your $120/$100 put spread safe but monitor.
Stage 1 condition watch: STH-MVRV compression
is progressing. No structural confirmation yet.
Next alert trigger: $125 or SRIBI zero-cross.
```

4:20 PM — End-of-Day Recap

MSTR closed at $131. Your put spread is comfortable. SRI indicators: Slow TL still red, SRIBI at -28, MVRV at 0.92. No stage change. Theta decay earned ~$95 on your positions. Tomorrow: watch FOMC minutes release at 2 PM.

Anytime — On-Demand Chat

> **YOU SAY TO YOUR AGENT ON DISCORD:**
> "Give me a full SRI stage report. Where are we on each timeframe and how close are we to Stage 1?"

The agent reads all indicator values from the database and gives you a comprehensive breakdown of every timeframe setting, every sub-indicator, and a structured assessment of Stage 1 proximity.

9. 7-Day Build Sprint

This is your construction sequence. You are working 24/7 and building aggressively. Each day has a clear goal, specific Discord prompts you'll give your agents, and a milestone to verify before moving on.

▶ **DO THIS NOW:** PREREQUISITE: Sign up for ORATS, Glassnode, Unusual Whales, and FRED API keys TONIGHT. Some providers take hours to activate. Do not wait until Day 1.

Day 1: Foundation

Goal: MSTR agent team running on OpenClaw. All API keys verified. Directory structure ready.

Morning: Agent setup

- Edit /home/openclaw/.openclaw/openclaw.json to add five agents: mstr-strategist (Opus 4.6), mstr-macro (Sonnet 4.6), mstr-technical (Sonnet 4.6), mstr-options (Sonnet 4.6), mstr-sri (Sonnet 4.6)
- Create Discord channels: #mstr-strategy, #mstr-macro, #mstr-technical, #mstr-options, #mstr-sri
- Create workspace directories for each agent
- Mount Python, shared libs, SSL certs (same bind mounts as PurelyBlu agents)
- Restart OpenClaw: systemctl restart openclaw
- Test each agent by messaging on Discord

Afternoon: Infrastructure

- Create directory structure: mkdir -p /home/openclaw/mstr-engine/{scripts,config,logs,data}
- Create .env file with all API keys
- Install Python packages: pip install anthropic httpx pandas pandas-ta numpy scipy schedule pydantic python-telegram-bot --break-system-packages
- Set up Telegram bot (talk to @BotFather, get token and chat ID)
- Write system prompts for all five agents with help from Claude advisor (this chat)

Evening: Verify API connections

> **YOU SAY TO YOUR AGENT ON DISCORD:**
> "Write me a Python test script that verifies connectivity to all five data providers: ORATS, Glassnode, Unusual Whales, FRED, and SEC EDGAR. For each one, make a simple API call, print the response status, and confirm we're getting data back. Use the API keys from /home/openclaw/mstr-engine/config/.env"

☑ **Milestone:** All five agents responding on Discord. All five API connections verified. Telegram bot sends you a test message.

Day 2: Data Pipeline

Goal: All five data sources flowing into SQLite on automated schedules.

> **YOU SAY TO YOUR AGENT ON DISCORD:**
> "I need you to build a complete data collection system. Here are my API keys: [from .env]. Build me: (1) A SQLite database schema at /home/openclaw/mstr-engine/data/mstr.db with tables for: MSTR OHLCV prices, MSTR options chain snapshots, BTC on-chain metrics, macro indicators, SEC filings, Unusual Whales flow, SRI indicator values, analysis results, portfolio positions, and trade log. (2) A data collection script for each provider that fetches data and inserts it into the database. (3) The crontab entries to run each script on schedule: ORATS every 5 min during market hours, Glassnode hourly, FRED daily at 6 AM, Unusual Whales every 15 min during market hours, SEC EDGAR every 30 min."

Your agent will produce 5-6 Python scripts. Test each one individually, then deploy to cron.

☑ **Milestone:** Query the database and see fresh data from all five sources. Run sqlite3 /home/openclaw/mstr-engine/data/mstr.db and verify tables have recent rows.

Day 3: SRI Indicator Replication

Goal: All three SRI indicators calculating in Python and matching TradingView output.

This is the most technically demanding day. You're replicating the Pine Script math.

> **YOU SAY TO YOUR AGENT ON DISCORD:**
> "I need you to replicate three TradingView Pine Script indicators in Python. Here is the Pine Script code for each one: [paste the three .txt files]. The indicators are: (1) Stage & Reversal Indicator (SRI) — Reversal Bands + Fast/Slow Tracklines at four timeframe settings. (2) SRI Bias Oscillator (SRIBI) — composite score from trackline colors, slopes, crosses, band position, linger penalties. (3) STH-MVRV Replica — price / 155-day SMA with color zones. Build a Python module called sri_engine.py that: (a) Takes OHLCV DataFrame as input, (b) Computes all three indicators at all four timeframe settings, (c) Returns structured results matching the JSON output format I'll specify. Use pandas, numpy, and pandas-ta. Pay special attention to the multi-timeframe logic — Pine Script's request.security() needs to be replicated with pandas resample/merge. Also replicate the stage transition boolean rulesets (stage4_to1, stage1_to2, stage2_to3, stage3_to4) exactly as coded in the Pine Script."

After the agent produces the code, compare its outputs against your TradingView charts at multiple dates. The numbers should be close (minor floating-point differences are fine). If they diverge significantly, iterate with the agent until they match.

☑ **Milestone:** sri_engine.py produces SRI, SRIBI, and MVRV values that match your TradingView charts for the current date and at least 3 historical reference points.

Day 4: All Five Analysis Agents

Goal: All five sub-agents producing structured JSON analysis from live data.

Build all five analysis scripts. Start with the simplest, end with the most complex:

Script 1: technical_analyst.py

> **YOU SAY TO YOUR AGENT ON DISCORD:**
> "Build technical_analyst.py that reads MSTR price data from SQLite, calculates RSI, MACD, Bollinger Bands, 50/100/200 MAs, support/resistance levels, sends it all to Claude Sonnet via the Anthropic API with a Technical Analyst system prompt, and saves the structured JSON output to the database. Use prompt caching."

Script 2: macro_analyst.py (reads Glassnode + FRED + EDGAR data)

Script 3: options_strategist.py (reads ORATS + Unusual Whales + Technical output)

Script 4: sri_stage_analyst.py (runs sri_engine.py + sends to Claude for interpretation)

For the SRI Stage Analyst, the Claude prompt should include the full tutorial context — the stage definitions, transition rules, case studies, and current conditions. The agent receives the raw indicator values and produces the structured stage assessment with transition probability.

☑ **Milestone:** Run all four scripts manually. Each produces a JSON report you can read and that makes sense given current market conditions. Compare the SRI agent's output against the Feb 16 market structure report for consistency.

Day 5: Morning Brief + Chief Strategist

Goal: Full Morning Brief pipeline running. You wake up Day 6 to your first real brief.

> **YOU SAY TO YOUR AGENT ON DISCORD:**
> "Build morning_brief.py that: (1) Runs all four sub-agent scripts and collects their JSON outputs, (2) Reads my current portfolio state from the database, (3) Sends everything to Claude Opus 4.6 with a Chief Strategist system prompt that includes my trading rules, risk parameters, and all four sub-agent reports, (4) Saves the Morning Brief to the database, (5) Sends a formatted summary to Telegram. The Chief Strategist prompt should weight the SRI Stage Analyst's output as one input among five — important context but not the sole decision driver until we've validated it through backtesting. Set up the cron job to run at 6:15 AM ET every weekday."

☑ **Milestone:** You wake up on Day 6 to a Morning Brief on Telegram. Read it. Does it make sense? Would you trade on this information?

Day 6: Intraday Monitoring + Alerts

Goal: Real-time monitoring running during market hours. Targeted alerts when conditions fire.

> **YOU SAY TO YOUR AGENT ON DISCORD:**
> "Build intraday_monitor.py that runs continuously from 9:30 AM to 4 PM ET. Every 5 minutes it should: (1) Pull latest MSTR price from ORATS, (2) Recalculate SRI indicators for the current bar, (3) Compare against morning's support/resistance levels and SRI stage thresholds, (4) Check Unusual Whales for any large MSTR flow, (5) If ANY trigger fires (price hits key level, IV spikes/crashes, SRI stage transition signal, SRIBI zero-cross, large institutional trade): run the Options Strategist and Chief Strategist analysis, send Telegram alert. (6) If no triggers fire, stay silent. I do NOT want noise."

☑ **Milestone:** Run during live market hours on Day 6. You receive alerts ONLY when something actionable happens. Test by setting a trigger at a level MSTR is likely to hit.

Day 7: Portfolio Tracking + EOD + Go Live

Goal: Complete daily loop operational. Trade logging. EOD recap. System is live.

> `YOU SAY TO YOUR AGENT ON DISCORD:`
> "Build me: (1) A way to log trades via Discord — I tell mstr-strategist what I traded and it saves to the database with proper structure. (2) eod_recap.py that calculates daily P&L on open positions using ORATS data, generates a summary including SRI state changes, and sends via Telegram. Cron at 4:15 PM ET. (3) A !portfolio command for Discord that shows all open positions and current status. (4) A !stage command that shows current SRI stage designation across all timeframes."

Afternoon: End-to-end testing and polish

- Run the full cycle manually: data collection → analysis → Morning Brief → intraday monitoring → EOD recap
- Fix any bugs or formatting issues
- Tune alert thresholds (too many alerts = adjust triggers, too few = loosen them)
- Verify all cron jobs are scheduled correctly: crontab -l
- Set up log rotation so logs don't fill your disk

☑ **Milestone:** Complete system running in production. Morning Brief, intraday alerts, EOD recap, portfolio tracking, SRI stage monitoring. You are live.

Sprint Summary

Day	Goal	Key Deliverable
Day 1	Foundation	5 agents on Discord + all APIs verified + Telegram bot
Day 2	Data Pipeline	All data sources flowing into SQLite on cron schedules
Day 3	SRI Replication	sri_engine.py matching TradingView output
Day 4	Analysis Agents	4 sub-agent scripts producing structured JSON analysis
Day 5	Morning Brief	Full pipeline running. First brief delivered via Telegram.
Day 6	Intraday Monitoring	Real-time alerts during market hours
Day 7	Portfolio + EOD + Go Live	Complete daily loop. Trade logging. System is live.

10. OpenClaw Configuration

Agent Definitions

Agent ID	Model	Discord Channel	Workspace
mstr-strategist	anthropic/claude-opus-4-6	#mstr-strategy	workspace-mstr-strategist/
mstr-macro	anthropic/claude-sonnet-4-6	#mstr-macro	workspace-mstr-macro/
mstr-technical	anthropic/claude-sonnet-4-6	#mstr-technical	workspace-mstr-technical/
mstr-options	anthropic/claude-sonnet-4-6	#mstr-options	workspace-mstr-options/
mstr-sri	anthropic/claude-sonnet-4-6	#mstr-sri	workspace-mstr-sri/

Sandbox Tool Mounts

Same as existing agents. All MSTR agents need Python 3, shared libs, and SSL certs:

- /usr/bin/python3, /usr/lib/python3, /usr/lib/python3.12, /lib/x86_64-linux-gnu, /usr/local/lib/python3.12 (all :ro)
- /home/openclaw/ssl-certs:/etc/ssl/certs:ro
- /home/openclaw/mstr-engine/data/:/mnt/mstr-data/:rw (shared database access)

Python Packages (Install on VPS)

pip install anthropic httpx pandas pandas-ta numpy scipy schedule pydantic python-telegram-bot sqlalchemy ta-lib --break-system-packages

Directory Structure

- /home/openclaw/mstr-engine/scripts/ — all cron scripts
- /home/openclaw/mstr-engine/config/.env — API keys
- /home/openclaw/mstr-engine/data/mstr.db — SQLite database
- /home/openclaw/mstr-engine/logs/ — cron job logs

Environment Variables (.env file)

- ANTHROPIC_API_KEY=sk-ant-...
- ORATS_API_TOKEN=...
- GLASSNODE_API_KEY=...
- UNUSUAL_WHALES_API_KEY=...
- FRED_API_KEY=...
- TELEGRAM_BOT_TOKEN=...

- TELEGRAM_CHAT_ID=...

11. Monthly Operating Costs

Item	Provider	Monthly Cost
Options Analytics + MSTR Price	ORATS Live Intraday	$399
BTC On-Chain Metrics	Glassnode Advanced	$29
Institutional Options Flow	Unusual Whales Annual	$37
Macro Data	FRED	$0
MSTR Filings	SEC EDGAR	$0
AI Reasoning	Claude API (Opus + Sonnet)	$75–$150
VPS Hosting	Existing server	$0*
Alerts	Telegram Bot API	$0
	TOTAL	$540–$615/month

* Existing VPS has capacity. If upgrade needed, add $12–$24/month.

Developer cost: $0. You are the product manager. Your agents are the engineers.

Break-even: One successful 10-contract bull put spread collecting $4.00 credit = $4,000 income. That covers 6–7 months of operating costs.

12. Week 2+: Backtesting & Signal Development

This is where the SRI framework transforms from an interpretive tool into a quantitative signal generator. The sprint gets the system live. Week 2 makes it smart.

Backtesting Objectives

- Validate or invalidate the confidence percentages (87%, 89%, 88%, 93%) from the Pine Script code
- Determine which of the six Stage 1 entry conditions have the highest predictive value
- Measure average return and duration for each stage across BTC and MSTR histories
- Calculate false positive rates for each stage transition signal
- Identify which timeframe settings (Very Short through Very Long) carry the most signal vs. noise
- Establish baseline performance metrics: win rate, Sharpe ratio, max drawdown by stage

Backtesting Methodology

Your SRI Stage Analyst agent runs the backtest using ORATS historical data (available back to 2007). The process:

- Pull 3–5 years of daily MSTR and BTC OHLCV data
- Calculate all SRI/SRIBI/MVRV indicators at every historical bar (walk-forward, no lookahead)
- Apply the stage transition boolean rulesets at each bar
- When a transition fires: record the date, price, direction, and all indicator states
- Track the subsequent 5-day, 10-day, 30-day, and 60-day price movement after each signal
- Compare against a simple buy-and-hold baseline

Signal Development Roadmap

Level 1 (Week 2): Binary signals. Each stage transition produces a simple Accumulate / Hold / Reduce / Hedge classification. No position sizing, no specific strikes. Just directional bias based on validated stage designation.

Level 2 (Week 3–4): Graded signals. Stage transitions carry a confidence score based on backtest statistics (not the Pine Script estimates). The score determines how aggressively the Options Strategist sizes its recommended spreads. High-confidence Stage 1 = wider credit spreads, more contracts. Low-confidence Stage 3 = tighter hedges, smaller positions.

Level 3 (Month 2+): Integrated regime model. The SRI framework's stage designation anchors the directional thesis. The Options Strategist selects structures based on IV regime + SRI stage. The

Macro Analyst's risk-on/off classification modulates position sizing. This is the "world-class hedge fund" configuration, but it's earned through validated performance, not assumed from Day 1.

13. Risks & Limitations

What This System Does Well

- Removes emotion from analysis — no panic, no FOMO, no revenge trading
- Processes more data simultaneously than any human can
- Maintains perfect memory of every trade, thesis, and outcome
- Never misses a setup because you were busy or distracted
- Enforces systematic risk management through consistent analysis
- Replicates and extends your brother's proprietary indicator framework with backtesting rigor

What It Cannot Do

- Predict the future — it improves probability, not certainty
- React instantly to black swan events — flash crashes may produce delayed recommendations
- Replace your judgment — you evaluate and execute every trade
- Guarantee profits — options trading involves risk of loss

Sprint Risks

- API provisioning delays — sign up for everything tonight
- SRI replication accuracy — Day 3 may take extra hours if multi-timeframe logic diverges from TradingView. Budget extra time.
- Prompt engineering iteration — your agents' analysis quality depends on system prompts. Plan to refine Week 2.
- Agent code bugs — always test scripts manually before deploying to cron. Check logs daily.
- Single point of failure — if VPS goes down, everything stops. Enable DigitalOcean backups ($1–2/month).

Financial Disclaimer

This system is a decision-support tool, not financial advice. Options trading involves substantial risk of loss. Past performance (including backtesting results) does not guarantee future results. Never risk money you cannot afford to lose. Consider consulting a licensed financial advisor before implementing any trading strategy.

14. Glossary

API: A way for software systems to talk to each other. Your system uses APIs to get market data and talk to Claude.

ATR: Average True Range. Measures price volatility. Used by the SRI Tracklines to set trailing stop distances.

Cron Job: A scheduled task on a Linux server. Like an alarm clock for a script.

Greeks: Metrics describing how an option's price changes: Delta (price sensitivity), Theta (time decay), Vega (volatility sensitivity), Gamma (rate of change of Delta).

Implied Volatility (IV): The market's expectation of future price movement. Higher IV = more expensive options.

IV Rank / Percentile: Where current IV sits vs. history. 90th percentile = IV is higher than 90% of the past year.

Iron Condor: Selling both a bull put spread and bear call spread. Profits when stock stays in a range.

Linear Regression: A statistical method fitting a straight line to price data. Used by SRI's Reversal Bands.

mNAV: MSTR market cap ÷ (BTC held × BTC price). Shows if MSTR is cheap or expensive relative to its Bitcoin.

MVRV: Market Value to Realized Value. BTC metric comparing current price to average holder cost basis.

OpenClaw: Your AI agent platform. Runs agents on Discord that can write code and hold conversations.

Reversal Bands: SRI component: linear regression channel with support/resistance lines that define mean-reversion boundaries.

Robust Fit: The center line of the Reversal Bands. The linear regression line itself.

RSI: Relative Strength Index. Oscillator measuring momentum. Above 50 = bullish, below 50 = bearish.

Slow Trackline: SRI component tracking the persistent trend. Green = rising, red = falling, orange = flat. Its color is the key tell for Stage 2 consolidation vs. Stage 3.

Spread: An options strategy using 2+ contracts to limit risk.

SQLite: A lightweight database stored as a single file. No server needed.

SRIBI: SRI Bias Indicator. Histogram scoring market bias from -110 to +110. Bright red = bearish exhaustion (buy territory). Bright green = bullish exhaustion (sell territory).

Stage 1 (Accumulation): Bottoming phase. Smart money accumulates. MVRV in green zones. SRI shows price recovering above Reversal Band support.

Stage 2 (Markup): Bull phase. Rising prices. Consolidation/continuation sub-patterns common. Slow TL shows green-orange alternation.

Stage 3 (Distribution): Topping phase. Only confirmed retroactively when Stage 4 begins. Slow TL loses green, shows red. MVRV peaked in red zone.

Stage 4 (Markdown): Bear phase. Declining prices. Current MSTR stage as of Feb 2026. Can include consolidation sub-patterns mimicking Stage 1.

STH-MVRV: Short-Term Holder MVRV. Price ÷ 155-day SMA. Proxy for recent buyer profitability.

Tracklines: SRI components: ATR-based adaptive trailing stops that flip direction based on RSI. Fast captures short-term momentum, Slow captures persistent trend.

VPS: Virtual Private Server. A remote computer running your system 24/7.

Start Tonight

Tonight: Sign up for ORATS, Glassnode, Unusual Whales, FRED. Create Telegram bot.

Day 1: Configure agents in OpenClaw. Verify all APIs. Write system prompts.

Day 2: Build data pipeline. All five sources into SQLite on cron.

Day 3: Replicate SRI indicators in Python. Verify against TradingView.

Day 4: Build all four sub-agent analysis scripts.

Day 5: Wire Morning Brief pipeline. First real brief delivered.

Day 6: Intraday monitoring. Real-time alerts during market hours.

Day 7: Portfolio tracking. EOD recap. Go live.

Week 2: Backtest SRI framework. Validate signals. Develop regime model.

Total cost to build: $0
Monthly operating cost: ~$540–$615
Break-even: One good trade
7 days to live