

SIP Premarket Bot Prediction Engine v2

Probability-based market-state engine, architecture map, and complete parameter reference for symbol evaluation and entry qualification.

Project: /var/www/screener/premarket

Documented as of: 2026-09-08 (Phases 1-6 complete)

Live decision engine: `entry.decision_engine = "prediction_pipeline"`

Execution mode: `paper` (no real money)

Backup snapshot: /var/www/screener/trade/premarket_backup_2026-09-08_2010/

Contents

1. Executive Summary
2. System Architecture — Full Pipeline Diagram
3. Data Flow Narrative
4. Engine Reference — Purpose, Formulas, Parameters
4.1 stream.py — SIP Buffers
4.2 stream_features.py — Feature Layer
4.3 trend_engine.py
4.4 structure_engine.py
4.5 compression_engine.py
4.6 regime_engine.py — Market State Machine
4.7 setup_engine.py
4.8 prediction_engine.py
4.9 relative_strength_engine.py
4.10 winner_score.py + state_transitions.py
4.11 entry_score.py — THE ENTRY GATE
5. Legacy Path (entry_engine.py / intraday_health.py)
6. Premarket Scanning & Symbol Universe
7. Risk & Position Management
8. Full Parameter Reference (config.json)
9. Known Limitations & Validation Status
10. Change Log — Phases 1-6

1. Executive Summary

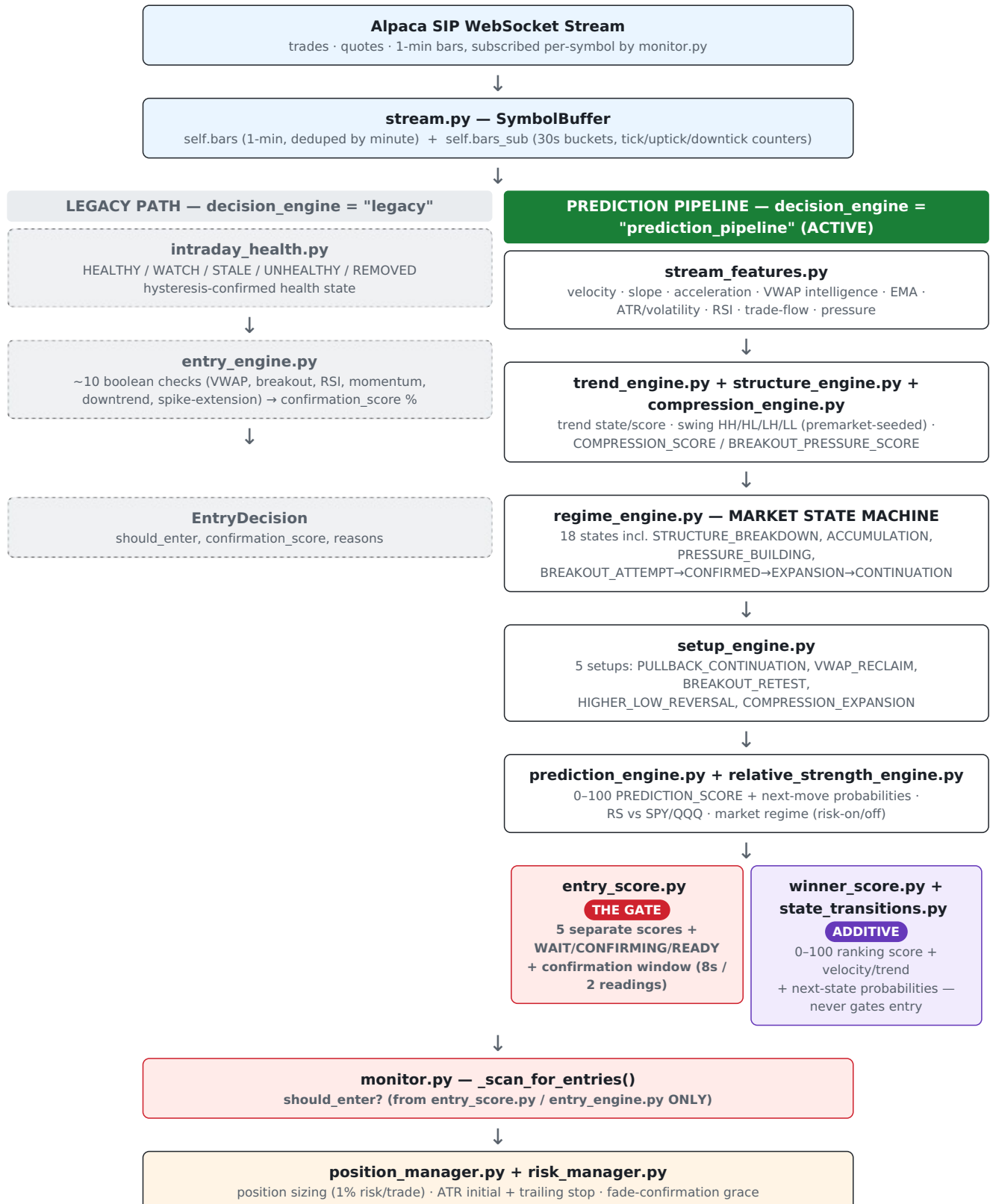
This bot scans, ranks, and paper-trades US equities in the \$5-\$15 band using Alpaca's SIP real-time data feed. As of 2026-09-08 it runs an upgraded prediction architecture (built across six phases this session) whose goal is to recognize high-probability intraday winners **earlier** — before the majority of a move is complete — rather than reacting to a move already underway.

The system now operates as a **probability-based market-state engine**: instead of asking only "should I buy this," it continuously classifies each candidate's current state (e.g. `ACCUMULATION` → `PRESSURE_BUILDING` → `BREAKOUT_ATTEMPT` → `BREAKOUT_CONFIRMED` → `EXPANSION` → `CONTINUATION`), estimates the probability of the next state, and computes a composite 0-100 `WINNER_SCORE` with velocity/trend tracking — all layered **additively** on top of an unchanged, validated five-score entry gate (`entry_score.py`) that alone decides whether a trade actually fires.

Property	Value
Active decision engine	<code>entry.decision_engine = "prediction_pipeline"</code>
Execution mode	<code>paper</code> — real orders to Alpaca's paper account, no real money
Universe	\$5.00-\$15.00, min avg daily volume 500,000 shares, ETFs/leveraged/OTC excluded
Poll cadence	Every 5 seconds during regular trading hours (09:30-15:55 ET)
Max concurrent positions	5
Risk per trade	1.0% of account equity, ATR-based trailing stop ($0.4 \times \text{ATR}$)
Entry gate (unchanged mechanism)	5 separate scores must ALL clear simultaneously for ≥ 8 seconds across ≥ 2 readings — trend, structure, prediction, entry, risk

Validation status, stated plainly: the new recognition layer (state machine, compression, relative strength, premarket seeding) is validated against two known real winners (SMR and MARA, 2026-09-08) and reproduces catching both early. The entry gate's thresholds were loosened the same day, grounded in the real 4-day score distribution, and confirmed to let both of those winners through under realistic 5-second live polling. **It has not yet been validated for false-positive rate against a losing day** — that requires a fresh multi-day backtest re-run, not yet done. Treat the first several live trading days as an active observation period, not a settled result.

2. System Architecture — Full Pipeline Diagram





Alpaca Order Execution (paper account)

exit via trailing stop, fade-confirmation, or end-of-day liquidation (15:55 ET)

Only the entry **trigger** differs between the two left/right paths — position sizing, stop computation, and exit logic are identical and completely untouched either way.

3. Data Flow Narrative

3.1 Premarket (before 09:30 ET)

`premarket_scanner.py` runs once at `09:29:00`. It pulls the full tradable-asset universe, filters to the \$5-\$15 price band and a basic volume floor via a cheap snapshot pass, then scores every survivor with `scorer.score_premarket_candidate()` — a transparent, weighted-breakdown score (13 components, see §6). The top 30 become `premarket_20` (the primary intraday candidate pool despite the name), and the top 15 of those become `final_10` (dashboard/informational).

3.2 Regular session (09:30–16:00 ET)

Every 5 seconds, `monitor.py`'s main loop:

1. Advances any exit already in flight (poll pending order fills).
2. Manages open positions — trailing-stop checks via `position_manager.py`.
3. If a slot is free and before the 15:15 cutoff: ranks all not-yet-open, health-eligible `premarket_20` symbols by health score, evaluates the top 5 (the **entry shortlist**) through the active decision engine.
4. Reconciles broker state (every 30s, throttled).
5. Refreshes continuous intraday health for the whole pool (every 60s).
6. Every 30 minutes: re-scores the *entire* prefiltered universe and rotates `premarket_20` to the new top 20 by health score — this is how a genuinely new candidate can enter mid-day, not just rotate among the original 20.

3.3 The confirmation window — "never enter from one tick"

Whichever decision engine is active, a single favorable reading is never enough. The prediction pipeline's `entry_score.py` requires all 5 gates to clear **simultaneously** for at least `confirmation_seconds` (8s) across at least `confirmation_min_readings` (2) separate polls — any single reading that fails resets the window immediately, no partial credit. At the live 5-second poll cadence, a condition that clears and holds for one full minute comfortably produces ~12 readings, so this window is easily satisfiable within a single still-forming bar once a real setup holds — confirmed by direct replay against real 2026-09-08 data (see §9).

4. Engine Reference — Purpose, Formulas, Parameters

4.1 `stream.py` — SIP Buffers

Maintains two parallel rolling windows per symbol, fed by the same incoming trade ticks:

Buffer	Resolution	Retention	Consumed by
<code>self.bars</code>	1-minute (deduped by minute key)	390 bars (~1 session)	Every engine, both paths
<code>self.bars_sub</code>	30-second buckets (configurable), carries tick_count/upticks/downticks	15 minutes (30 buckets)	<code>stream_features.py</code> only (prediction pipeline)

4.2 `stream_features.py` — Feature Layer

Pure, stateless computation over the buffers above (only cross-call state is the caller-held `prior` reading, for true time-derivative acceleration). Produces:

Field group	Contents
price	last, price_1m/3m/5m/sub
velocity	%/second over 1m and sub-minute windows
slope	cross-horizon %/minute: slope_sub, slope_1m, slope_3m, slope_5m
acceleration	momentum_acceleration (cross-horizon waterfall) + true time-derivative velocity/slope_acceleration vs. prior reading
volume	volume_1m/3m/5m/sub, volume_acceleration, relative_volume
vwap	value, slope, price_vs_vwap_pct, time_above/below_pct, hold_duration_bars , rejection_count , classification (Phase 3)
ema	ema9, ema20, ema9_slope, price_vs_ema9_pct/atr
volatility	atr, atr_pct, vwap_distance_atr, range_expansion_pct
rsi	value (Wilder's smoothing), has_full_history flag
trade_flow	tick_count_sub, trades_per_second, uptick/downtick_ratio
pressure	score (-100..+100 selling↔buying), slope

VWAP classification (7 states, Phase 3)

State	Condition
STRONG_VWAP_SUPPORT	above VWAP, held $\geq$ vwap_min_hold_bars_for_support (5) consecutive bars
VWAP_RECLAIM	above VWAP, held 1–4 bars (a recent reclaim)
VWAP_HOLD	above VWAP, just crossed this bar
VWAP_EXTENSION	above VWAP but distance $>$ vwap_max_extension_atr ($3.0 \times \text{ATR}$)
VWAP_DECLINING	still above VWAP, but VWAP's own slope has turned negative
VWAP_FAILURE	below VWAP and (slope negative OR $\geq$ vwap_min_rejections_for_failure (2) intrabar rejections)
VWAP_FLAT	below VWAP, no other signal (default)

Parameter	Value
min_bars_1m / min_bars_sub	3 / 2
vwap_time_window_bars	20
ema9_period / ema20_period	9 / 20
rsi_period / atr_period	14 / 14
pressure_weights	uptick 0.40, velocity 0.35, volume 0.25
vwap_min_hold_bars_for_support	5
vwap_min_rejections_for_failure	2
vwap_max_extension_atr	3.0

4.3 trend_engine.py

Combines 6 horizon slopes (slope_sub/1m/3m/5m, vwap_slope, ema9_slope) into: **direction** (UP/DOWN/NEUTRAL, from the longest available horizon first), **state** (STRONG/WEAK_UPTREND · NEUTRAL · WEAK/STRONG_DOWNTREND), a direction-agnostic 0–100 **score** (weighted agreement across horizons), and **trend_slope** (strengthening/weakening).

Parameter	Value
flat_slope_threshold_pct	0.05
strong_trend_score_threshold	70
weights: slope_sub / slope_1m / slope_3m / slope_5m / vwap_slope / ema9_slope	15 / 20 / 20 / 20 / 15 / 10

4.4 structure_engine.py

Causal swing-point detection (a bar is a confirmed HIGH/LOW only once pivot_confirm_bars LATER bars exist — no look-ahead, ever). Labels each swing HH/HL/LH/LL relative to the prior swing of the same kind, classifies overall structure as BULLISH / BEARISH / RANGE / TRANSITION, and detects failure flags (structure_break, failed_higher_low, failed_breakout, support_failure, resistance_rejection).

Phase 5 fix: accepts optional `premarket_high/premarket_low` to seed `last_swing_high/last_swing_low` as a day-1 reference before any live swing has formed — closes a gap where these were both `None` for the first ~10-12 minutes of every session.

Parameter	Value
<code>pivot_lookback_bars / pivot_confirm_bars</code>	2 / 2
<code>min_swings_for_structure</code>	2 (of each kind)
<code>resistance_rejection_tolerance_pct</code>	0.3
<code>structure_score_lookback_swings</code>	4

4.5 `compression_engine.py` Phase 3 — LIVE

Detects range/volatility "coiling" ahead of a breakout, over a configurable lookback window compared against the immediately preceding window of equal length.

Metric	Formula
<code>range_contraction_ratio</code>	<code>recent_range_pct / earlier_range_pct</code> (<1 = tighter)
<code>atr_contraction_ratio</code>	<code>ATR(recent) / ATR(earlier)</code>
<code>close_to_close_contraction_ratio</code>	<code>stdev(% changes, recent) / stdev(% changes, earlier)</code>
COMPRESSION_SCORE	weighted blend of the 3 contraction credits + "holding near highs" + "volume not collapsing"
BREAKOUT_PRESSURE_SCORE	<code>COMPRESSION_SCORE × multiplier</code> (1.0-1.5), bonus for VWAP slope>0, higher-lows present, price near resistance

Parameter	Value
<code>lookback_bars</code>	8
weights: <code>range / ATR / close-to-close contraction / near-highs / volume-not-collapsing</code>	30 / 25 / 15 / 15 / 15
<code>near_high_max_distance_pct</code>	3.0
<code>volume_not_collapsing_min_ratio</code>	0.7
pressure bonuses: <code>vwap_slope / higher_lows / near_resistance</code>	+0.15 / +0.20 / +0.15 (capped at 1.5×)
<code>near_resistance_max_distance_pct</code>	2.0

4.6 `regime_engine.py` — Market State Machine LIVE

The live, continuously re-evaluated classifier of "what's happening in this symbol's price action right now." Evaluated in strict priority order — first match wins:

#	State	Trigger	Allowed setups
0	STRUCTURE_BREAKDOWN	confirmed support failure (highest priority — overrides even the opening window)	none
1	OPENING_VOLATILITY	within first 30 min of session AND ATR% $\geq$ 1.0	BREAKOUT_RETEST
2	FAILED_BREAKOUT	structure_engine confirmed a round-tripped breakout	none
3	RECOVERY	LL $\rightarrow$ HL swing sequence with trend_slope turning positive	HIGHER_LOW_REVERSAL
4	VWAP_RECLAIM	above VWAP after real time below it, momentum turning positive	VWAP_RECLAIM
5	BREAKOUT_ATTEMPT $\rightarrow$ CONFIRMED $\rightarrow$ EXPANSION $\rightarrow$ CONTINUATION	price above last confirmed swing high in an uptrend; sub-states by consecutive real bar-closes held (2 / 3 / 5 min thresholds)	BREAKOUT_RETEST (+PULLBACK_CONTINUATION from CONFIRMED onward)
6	PULLBACK	established uptrend, short-term momentum negative, structure still bullish	PULLBACK_CONTINUATION
7	DISTRIBUTION	elevated volume + fading momentum ahead of a confirmed breakdown	none
8	STRONG_UPTREND / WEAK_UPTREND	trend engine says so, structure doesn't contradict	PULLBACK_CONTINUATION, BREAKOUT_RETEST, VWAP_RECLAIM (strong only)
9	DOWNTREND	trend engine WEAK/STRONG_DOWNTREND	none
10	PRESSURE_BUILDING	RANGE + breakout_pressure_score $\geq$ 70 (compression, requires compression reading)	COMPRESSION_EXPANSION, BREAKOUT_RETEST
11	ACCUMULATION	RANGE + compression_score $\geq$ 55	COMPRESSION_EXPANSION
12	RANGE	RANGE structure, calm ATR%, no compression reading supplied	COMPRESSION_EXPANSION
13	CHOP	fallback — no clear directional edge	none

Phase 5 bugfix: the BREAKOUT progression's "bars since breakout" counter now advances on real 1-minute bar closes only (via a `bar_t` timestamp dedup), not once per 5-second poll cycle — the original version completed the whole progression in 10-50 seconds instead of the intended ~2-10 minutes.

Parameter	Value
opening_volatility_window_minutes / min_atr_pct	30 / 1.0
pullback_trend_slope_threshold	-0.5
distribution_min_volume_acceleration / max_trend_slope	1.3 / -0.2
vwap_reclaim_min_time_below_pct	30.0
chop_min_atr_pct / range_max_atr_pct	0.8 / 0.8
structure_breakdown_hard_block	true (config-toggleable safety gate)
accumulation_min_compression_score	55
pressure_building_min_breakout_pressure_score	70
breakout_confirmBars / expansion_minBars / continuation_minBars	2 / 3 / 5 (real minutes)
breakout_volume_confirm_ratio	1.0

4.7 setup_engine.py

Five stateless entry-pattern checklists, each requiring ALL its checks to pass for `valid=True` (confidence % reported regardless, for visibility):

Setup	Key checks
PULLBACK_CONTINUATION	established uptrend, bullish structure, most recent low is HL, price reclaimed EMA9, momentum positive, volume re-expanding
VWAP_RECLAIM	above VWAP after real time below it ($\geq 30\%$), held a higher low since, momentum positive, volume accelerating
BREAKOUT_RETEST	confirmed resistance exists, broken above at some point, now retesting within buffer, holding as support, uptrend intact
HIGHER_LOW_REVERSAL	LL then HL bottoming pattern, velocity turned positive, price reclaimed EMA9
COMPRESSION_EXPANSION	range expanding out of prior compression, structure held through it, volume expanding, not breaking down

Parameter	Value
vwap_reclaim_min_time_below_pct	30.0
retest_buffer_pct	0.5
min_volume_acceleration	1.0
compression_expansion_min_range_expansion_pct	40.0

4.8 prediction_engine.py

Combines everything above into one 0–100 **PREDICTION_SCORE** (deterministic heuristic, not ML), a **prediction_slope** (vs. the prior reading), and a 5-state next-move probability distribution (CONTINUATION_UP / PULLBACK_THEN_UP / SIDEWAYS / PULLBACK_DOWN / REVERSAL_DOWN) for 1m/3m/5m horizons.

Parameter	Value
weights: trend/structure/momentum/acceleration/vwap/volume/pressure/relative_strength/breakout_quality/risk_quality	15/15/12/10/10/12/10/6/5/5
momentum_scale / acceleration_scale	2.0 / 3.0
max_vwap_atr / max_ema9_atr	3.0 / 3.0
extension_penalty_weight (uncapped fraction — see below)	20
exhaustion_min_bars_since_pullback / penalty_weight	6 / 15
spread_penalty_weight (max_spread_pct 1.0%)	10
chop_penalty_weight	15
poor_liquidity_penalty_weight (min_relative_volume 0.5)	10
horizon_confidence_decay: 1m/3m/5m	1.0 / 0.8 / 0.6

relative_strength weight (6) is documented but excluded from the live sum — `prediction_engine.py` has no cross-symbol data source itself; real relative strength is computed separately by `relative_strength_engine.py` and feeds `winner_score.py` instead.

4.9 relative_strength_engine.py Phase 5 — LIVE (partial)

Compares each candidate's return against SPY and QQQ across 5 windows, plus an optional sector/peer confirmation bonus from a static, hand-maintained map.

Parameter	Value
windows	1m, 3m, 5m, 15m, premarket
window_weights	1m 0.15 · 3m 0.25 · 5m 0.30 · 15m 0.20 · premarket 0.10
rs_scale_pct	2.0 ($\pm 2\%$ outperformance → full 0/100 score)
sector_breadth_min_for_bonus	0.6 (60% of peers positive)
sector_confirmation_bonus	+15%, multiplicative, capped at 100
sector_map (hand-maintained)	SMR → nuclear (OKLO, LEU, CCJ; ETF URA) · MARA → crypto_miners (RIOT, CLSK, CORZ)
market_regime_thresholds	strong $\pm 0.5\%$, mild $\pm 0.15\%$ (composite SPY/QQQ 5m return)

Partial live status: the SPY/QQQ half is fully wired (monitor.py subscribes to both whenever the prediction pipeline is active). The sector/peer half is **not** — no peer symbols (OKLO, RIOT, etc.) are subscribed yet, so `sector_breadth/momentum` stay `None` and that bonus never applies today. This is a documented gap, not a bug — a symbol with no sector data simply gets the neutral default, per the "sector confirmation is a bonus, never a requirement" design principle.

4.10 winner_score.py + state_transitions.py ADDITIVE — never gates entry

winner_score.py composes a single 0–100 ranking score plus velocity/acceleration/trend classification, using a caller-held score history:

Component	Weight	Source
trend_structure	20	60% trend.score + 40% structure.score
relative_strength	15	relative_strength_engine's final_score (neutral 50 if unavailable)
volume_expansion	15	60% volume_acceleration credit + 40% relative_volume credit
vwap_structure	15	VWAP classification lookup table (STRONG_SUPPORT=100 .. FAILURE=0)
premarket_setup	10	this symbol's own premarket total_score, carried forward
breakout_quality	10	best valid setup confidence × 0.7 + breakout_pressure_score × 0.3
volatility_expansion	5	range_expansion_pct, saturating at +50%
sector_strength	5	sector breadth/momentum (neutral default if unmapped)
market_regime	5	risk-on/off credit table (never a veto — a strong stock in a weak tape still scores)

Score momentum	Formula
velocity_per_min	time-weighted least-squares slope of recent (timestamp, score) history
acceleration_per_min	velocity_now − velocity_prior_reading
score_trend	RISING_FAST / RISING / STABLE / DECLINING / DECLINING_FAST (±1.0 / ±4.0 points-per-min thresholds)

state_transitions.py estimates next-market-state probabilities from a deterministic, config-driven base transition table (per current regime), nudged by `prediction.bullish_probability` and `compression.breakout_pressure_score`, renormalized to sum to 1.0.

4.11 entry_score.py — THE ENTRY GATE GATE — unchanged mechanism

The **only** thing that decides `should_enter` for the prediction pipeline. Deliberately five separate scores, not one blended number — a symbol can be Trend=94, Prediction=88, Entry=51, Risk=76 and still WAIT, because Entry (setup/trigger readiness) is what's actually low and no other score can outvote it.

Gate	Threshold	Computed from
trend_score	≥ 60	trend_engine.py directly
structure_score	≥ 50	structure_engine.py directly
prediction_score	≥ 40 (lowered from 60, 2026-09-08)	prediction_engine.py directly
entry_score	≥ 65	50% setup confidence + 30% regime permits it + 20% prediction momentum
risk_score	≥ 45 (lowered from 60, 2026-09-08)	40% stop quality (reward/risk ≥ 1.5) + 30% liquidity + 30% spread
spread	≤ 1.0%	live quote bid/ask

Confirmation window	Value
confirmation_seconds	8
confirmation_min_readings	2 (consecutive, any single miss resets to 0)
min_reward_risk_ratio	1.5

2026-09-08 threshold change, grounded in real data: the 2026-09-04 backtest's 144-symbol distribution showed `prediction_score`'s best-bar-all-day mean/median was only 47.6/49.6 and `risk_score`'s was 47.8/46.1 — the old 60 floor was above what most real winners ever reached even at their single best moment. Validated against real SMR/MARA data (2026-09-08) with a fully sequential replay matching the live 5-second poll cadence: SMR reaches READY at 10:05:10, MARA at 10:00:10 (price \$11.60, ~13 min before its \$12.06 high that day).
Not yet validated for false-positive rate against a losing day — see §9.

5. Legacy Path not currently active

Available via `entry.decision_engine = "legacy"` (an instant, no-restart config flip back if needed). Combines premarket rank with ~10 real-time boolean checks against live bars — price above/rising VWAP, opening volume expansion, higher lows, breakout of premarket high held for N bars (or a pullback-then-continuation alternative), momentum not fading (hard disqualifier), volume not declining, RSI not overbought, no recent spike-extension, no extended multi-hour downtrend, plus a hard `intraday_health.py` gate computed fresh at decision time. Produces a `confirmation_score` (% of checks passed) that must clear `min_confirmation_score` (65%) with zero disqualifiers tripped. `intraday_health.py` itself (HEALTHY/WATCH/STALE/UNHEALTHY/REMOVED, hysteresis-confirmed) remains active **regardless of which decision engine is selected** — it gates shortlist eligibility for both paths.

6. Premarket Scanning & Symbol Universe

Universe filter	Value
Price band	\$5.00 - \$15.00
Min avg daily volume	500,000 shares (universe floor; per-symbol real baseline used for RVOL as of Phase 5)
Excluded	OTC, ETFs/funds, leveraged/inverse products, names >5 syllables
Premarket candidate count / final count	30 / 15

`scorer.py`'s premarket score — 13 transparent, weighted components (0-100 normalized):

Component	Weight
volume_quality	12
rvol	12
volume_acceleration	10
price_momentum	12
price_structure (HH/HL)	12
vwap_structure	8
vwap_slope	8
consolidation_quality	8
momentum_consistency	8
distance_from_pm_high	6
price_quality (band position)	5
spread_liquidity	5
extension_risk_penalty	-10 (penalty only)

Parameter	Value
min_premarket_volume	20,000
min_rvol	1.5
max_spread_pct	1.5%
max_distance_from_pm_high_pct	8.0%
extension_risk_pct_threshold	25.0%

7. Risk & Position Management (identical for both decision-engine paths)

Parameter	Value
Account risk per trade	1.0% of equity
Max position notional	20% of equity
Max concurrent positions	5
Symbol cooldown after exit	10 minutes
Max daily loss	3.0% of equity → halts new entries
Max consecutive losses → pause	4 → 20 min pause
Stop method	ATR-based (initial 1.2×ATR, trailing 0.4×ATR)
Min stop distance	max(\$0.03, 1.0% of price)
Max stop distance	3.0% of price
Fade-confirmation grace	1 extension of \$0.05, gated on intraday_health NOT confirming a real fade
Re-entry bar	90% confirmation score required (vs. normal 65%) for a symbol already traded this session
End-of-day liquidation	15:55 ET, retried for up to 120s

8. Full Parameter Reference (config.json, as of 2026-09-08)

Every block below is independently configurable at runtime (no code change, just edit + restart). Blocks marked **LIVE** are actively consumed whenever `entry.decision_engine = "prediction_pipeline"`; their own `enabled` flag is informational only per this project's convention — the real switch is always `entry.decision_engine`.

entry.decision_engine

Value	Options
prediction_pipeline	legacy prediction_pipeline

schedule

Key	Value
premarket_scan_time	09:29:00
market_open_time / market_close_time	09:30:00 / 16:00:00
no_new_entries_after	15:15:00
force_liquidate_time	15:55:00
poll_interval_seconds_intraday	5
reconcile_interval_seconds	30

intraday_health

Key	Value
eval_interval_seconds	60
slope_lookback_bars / structure_lookback_bars	8 / 4
flat_slope_threshold_pct / severe_negative_slope_pct	0.05 / 0.6
healthy_confirm_reads / unhealthy_confirm_reads	2 / 3
remove_after_consecutive_unhealthy	3
weights: price_slope/vwap_support/volume_participation/structure/range_expansion/distance_from_high	20/20/20/20/10/10
full_rescan_interval_minutes / pool_size / lookback_hours	30 / 20 / 1
entry_shortlist_size	5
max_consecutive_confirmation_failures → cooldown	5 → 180s bench

streaming

Key	Value
feed	sip
stale_data_seconds	15
sub_minute_bucket_seconds / buffer_minutes	30 / 15
reconnect_backoff_seconds	[2, 5, 10, 20, 30], max 20 attempts

Full per-engine parameters for `stream_features`, `trend_engine`, `structure_engine`, `compression_engine`, `regime_engine`, `setup_engine`, `prediction_engine`, `relative_strength_engine`, `winner_score`, and `entry_score` are documented in full in §4 above with their exact current values — not repeated here to avoid duplication.

entry (legacy path parameters)

Key	Value
opening_volume_expansion_ratio	1.4×
breakout_buffer_pct / breakout_holdBars	0.1% / 2 bars
higher_lows_lookback_bars	4
max_spread_pct_entry	1.2%
max_extension_from_vwap_pct / min_extension_from_vwap_pct	4.0% / 0.3%
min_confirmation_score	65%
min_health_score_by_state	HEALTHY 65 / WATCH 70
require_momentum_not_fading_strict	true (hard disqualifier)
max_rsi_at_entry (require_rsi_not_overbought)	70
downtrend_min_bars / min_decline_pct / recovery_threshold_pct	20 / 1.5% / 1.5%
reentry_min_confirmation_score	90%

9. Known Limitations & Validation Status

Item	Status
State machine recognizes real winners early (SMR, MARA)	Validated against real 2026-09-08 data
Entry gate reachable under realistic 5s live polling	Validated (READY at 10:05:10 / 10:00:10 respectively)
False-positive rate on a losing day/symbol	Not validated — old 4-day backtest files predate Phase 5 changes and are no longer representative
Sector/peer confirmation bonus	⚠ Config exists, not wired — no peer symbols streamed yet
Entry-readiness presentation ladder (WATCH→BUILDING→ARMED→READY)	⚠ Not built — cosmetic, not load-bearing
Dedicated [PREDICTION] transition-only log line	⚠ Folded into existing [PIPELINE] line instead
Regression test suite (72 tests)	70 passing, 2 pre-existing unrelated failures (confirmed present before this session's changes too)

Both threshold changes and the `bar_t` bugfix went live under explicit time pressure for a next-trading-day deadline (2026-09-09). Watch real fills closely; the two numbers most likely to need retuning first are `entry_score.min_prediction_score` (currently 40) and `min_risk_score` (currently 45) — both are plain config values, no code change or redeploy needed to adjust.

10. Change Log — Phases 1-6 (2026-09-08)

Phase	Summary
1 — Audit	Full trace of the existing prediction flow; found the premarket-context gap, the flat-RVOL-constant bug, the 6-minute warmup blackout, and regime whipsaw at the open.
2 — Design	Locked the additive-WINNER_SCORE decision (Option A): keep entry_score.py's 5-score gate as the only trigger, layer everything new on top.
3 — Core calculations	Built compression_engine.py, relative_strength_engine.py, and stream_features.py's VWAP intelligence extension.
4 — State machine	Extended regime_engine.py's taxonomy (7 new states), built state_transitions.py and winner_score.py.
5 — Live integration	Wired everything into monitor.py/prediction_pipeline.py; fixed the premarket-seeding and RVOL-baseline bugs; validated end-to-end with zero test regressions.
6 — Gate calibration	Loosened entry_score's prediction/risk thresholds using real backtest data; found and fixed the bars_since_breakout poll-vs-bar scale bug; validated SMR/MARA would have fired live.