Options Codebase Review

Master Report

February 26, 2026

16 Expert Persona Reviews

Total Raw Findings	**~285**
Unique Findings	**62**
Net New Findings	**54**
Previously Known	**8**

Prepared by Perplexity Computer

Table of Contents

Executive Summary

This report consolidates findings from 16 expert persona reviews of the `/Options` codebase. Personas covered: Quantitative Options Trader, Risk Manager, Volatility Regime Specialist, Weekly Options Specialist, Value Hunter, Timing & Catalysts Analyst, Order Flow Expert, Portfolio Behavior Analyst, Macro Regime Analyst, Alpha Leakage Detector, Data Integrity Auditor, Senior Architect, Senior Developer, Security Engineer, Reliability & Testing Engineer, AI/UX Specialist.

Total raw findings (across all 16 reviews): ~285

Total unique findings (after aggressive deduplication): 62

Previously known/documented issues confirmed still present: 8 (see Cross-Cutting Concerns)

Net new unique findings: 54

Severity Breakdown (unique findings)

Severity	Count
P0 — Critical (fix immediately)	17
P1 — Major (fix this sprint)	21
P2 — Minor (fix next sprint)	14
P3 — Enhancement (backlog)	10

Most Dangerous Active Bugs

1. LEAP scanner silently returns only 1 option per expiry (`_process_expiration` return in loop)
2. Skew signal completely dead — 15% of every score frozen at 50
3. Earnings risk check stripped — weekly buyers get no warning before catalysts
4. Daily loss limit stored in DB but never enforced at trade placement
5. `FLASK_DEBUG=True` default exposes RCE via Werkzeug debugger in production
6. OCO bracket adjustment silently fails due to key mismatch — positions left unprotected
7. `lifecycle_sync()` fully implemented but never registered — trades never advance state

Top 10 Fixes (P0 — Do Now)

These are ordered by blast radius: how badly the system misbehaves right now for real users.

P0

P0-1 · `_process_expiration` — `return` trapped inside for-loop

Files: `options_analyzer.py` L229–254

Flagged by: Personas 1, 2, 4, 7, 13 (5/16 reviewers)

Status: NEW finding — not in known issues digest

Problem: The `return opportunities` statement (and the debug block above it) sits at one level of indentation too deep — inside the `for expiry in expirations:` loop. Python therefore returns after processing the **first** expiry date and discards all remaining ones. Every LEAP scan silently returns at most one option per ticker.

Fix:

```
# options_analyzer.py ~L252
# BEFORE (broken — indented inside for-loop):
    for expiry in expirations:
        calls = self._filter_strikes(chain, expiry, 'call', current_price)
        ...
        for opt in calls:
            opportunities.append(...)
        return opportunities          # ← WRONG: exits after first expiry

# AFTER (correct — dedented outside loop):
    for expiry in expirations:
        calls = self._filter_strikes(chain, expiry, 'call', current_price)
        ...
        for opt in calls:
            opportunities.append(...)
    return opportunities              # ← CORRECT: all expiries processed
```

Impact: Catastrophic. LEAP scan is fundamentally broken. Users see only 1 option per ticker regardless of how many expirations exist.

P0

P0-2 · Skew calculation dead — always returns 50

Files: `hybrid_scanner_service.py` L423–431; `options_analyzer.py` L354

Flagged by: Personas 1, 2, 3, 4, 5, 9, 10 (7/16 reviewers)

Status: NEW finding

Problem: `calculate_skew()` is guarded by `if self.use_schwab:` which is hardcoded `False`. The method body after the guard contains a fallback `return 50`. Result: skew is always 50. Since skew weight is 15%, every score has its skew component frozen at 7.5 points regardless of actual market conditions.

Fix:

```
# hybrid_scanner_service.py ~L423
# BEFORE:
def calculate_skew(self, ticker, expiry):
    if self.use_schwab:
        # ... real calculation ...
    return 50   # always hits this

# AFTER:
def calculate_skew(self, ticker, expiry):
    # use ORATS data (already fetched in scan context)
    skew_data = self.orats_client.get_skew(ticker, expiry)
    if not skew_data:
        return 50
    put_iv = skew_data.get('put_iv_25d', 0)
    call_iv = skew_data.get('call_iv_25d', 0)
    if call_iv == 0:
        return 50
    raw_skew = (put_iv - call_iv) / call_iv * 100
    return max(0, min(100, 50 + raw_skew))
```

Impact: 15–20% of every score is meaningless. Bearish-skewed tickers (high put demand before crashes) score identically to neutral ones.

P0

P0-3 · Earnings risk check stripped — `has_earnings_risk` always `False`

Files: `hybrid_scanner_service.py` L1030–1032

Flagged by: Personas 1, 2, 3, 4, 6, 9, 10 (7/16 reviewers)

Status: Partially known (known issue #11) — but the **specific code regression** is new

Problem: The earnings check block was removed/commented out, leaving only `has_earnings_risk = False` unconditionally. Finnhub earnings calendar is already integrated and working.

Fix:

```
# hybrid_scanner_service.py ~L1030
# BEFORE:
has_earnings_risk = False  # TODO: re-enable

# AFTER:
has_earnings_risk = False
try:
    earnings = self.finnhub_client.get_earnings_calendar(
        ticker, _from=date.today().isoformat(),
        to=(date.today() + timedelta(days=7)).isoformat()
    )
    has_earnings_risk = len(earnings.get('earningsCalendar', [])) > 0
except Exception as e:
    logger.warning(f"Earnings check failed for {ticker}: {e}")
```

Impact: System recommends buying weekly options (high-gamma, earnings-sensitive) into earnings with zero warning. A $10K weekly position bought 2 days before a binary event is normal behavior under this bug.

P0

P0-4 · Daily loss limit stored but never enforced

Files: `paper_routes.py` L333–348

Flagged by: Personas 2, 8

Status: NEW finding

Problem: `daily_loss_limit` is stored in user settings and displayed in the UI. At trade placement, it is never queried. A user can lose 10× their daily limit without any server-side block.

Fix:

```
# paper_routes.py — add to place_trade() before order submission
today = date.today().isoformat()
daily_pnl_row = db.execute(
    "SELECT COALESCE(SUM(realized_pnl), 0) as total "
    "FROM paper_trades WHERE username=? AND DATE(closed_at)=? AND status='CLOSED'",
    (username, today)
).fetchone()
daily_pnl = float(daily_pnl_row['total'])
daily_limit = float(user_settings.get('daily_loss_limit', 0))
if daily_limit > 0 and daily_pnl <= -daily_limit:
    return jsonify({'error': f'Daily loss limit of ${daily_limit} reached'}), 429
```

Impact: Unlimited intraday drawdown possible. Risk management feature is purely cosmetic.

P0

P0-5 · `FLASK_DEBUG` defaults to `True` — RCE exposure in production

Files: `config.py` L56

Flagged by: Personas 12, 13, 14 (3/16 reviewers)

Status: NEW finding

Problem:

```
FLASK_DEBUG = os.environ.get('FLASK_DEBUG', 'True') == 'True'
```

Any deployment that does not explicitly set `FLASK_DEBUG=False` runs the Werkzeug interactive debugger. This debugger is accessible at any traceback URL and allows arbitrary Python execution with no authentication.

Fix:

```
# config.py L56
FLASK_DEBUG = os.environ.get('FLASK_DEBUG', 'False') == 'True'

# Also add in app startup:
if not app.debug and app.config.get('SECRET_KEY') == 'dev-secret-key-change-in-production':
    raise RuntimeError("Default SECRET_KEY must not be used in production")
```

Impact: Anyone who can reach port 5000 on the Raspberry Pi can execute arbitrary code as the Flask process user. This is a complete system compromise vector.

P0

P0-6 · OCO `adjust_bracket()` silently fails — key mismatch

Files: `monitor_service.py` L851–868

Flagged by: Personas 2, 7

Status: NEW finding

Problem: `adjust_bracket()` constructs the Tradier API payload using wrong key names:

Code sends	Tradier expects
`qty`	`quantity`
`stop_price`	`stop`
`limit_price`	`price`

Tradier silently rejects the malformed payload. No exception is raised; the UI shows success.

Fix: Align keys to Tradier's documented OCO endpoint schema:

```
# monitor_service.py ~L860
payload = {
    'class': 'oco',
    'duration': 'gtc',
    'quantity': position['qty'],          # was: qty
    'type[0]': 'stop_limit',
    'stop[0]': new_stop,                  # was: stop_price
    'price[0]': new_stop * 0.95,
    'type[1]': 'limit',
    'price[1]': new_target,              # was: limit_price
}
```

Impact: Every bracket adjustment from the UI silently fails. Positions sit at their original stop/target indefinitely. Users who trail stops via the UI have no protection.

P0

P0-7 · OCO stop-loss uses market order `"stop"` not `"stop_limit"`

Files: `tradier.py` L408–423

Flagged by: Personas 2, 7

Status: NEW finding

Problem:

```
'type[0]': 'stop',   # market order triggered at stop price
```

A `stop` order becomes a market order when triggered. In illiquid options markets or gap events, this fills at the bid — potentially 50–99% below the intended stop price.

Fix:

```
'type[0]': 'stop_limit',
'stop[0]': stop_price,
'price[0]': round(stop_price * 0.95, 2),   # 5% below stop as limit
```

Impact: In a gap-down event, a $5.00 option intended to stop at $3.50 could fill at $0.10. Every live OCO order placed through this system has this exposure.

P0

P0-8 · `lifecycle_sync()` implemented but never scheduled

Files: `app.py` (APScheduler job registration)

Flagged by: Persona 15

Status: NEW finding

Problem: `app.py` registers 4 APScheduler jobs (order sync, price snapshots, bookend open, bookend close). `lifecycle_sync()` — which advances trades through PENDING→OPEN→CLOSED states and checks expiry — is fully implemented in `monitor_service.py` but has no scheduler registration.

Fix:

```
# app.py — add 5th scheduler job after existing job registrations
scheduler.add_job(
    func=monitor_service.lifecycle_sync,
    trigger='interval',
    seconds=60,
    id='lifecycle_sync',
    name='Trade Lifecycle Sync',
    replace_existing=True
)
```

Impact: PENDING trades never advance to OPEN. Expired options are never closed. The entire 7-state trade lifecycle is inert. Paper trading "works" but all trades stay PENDING forever unless manually updated.

P0

P0-9 · `log` vs `logger` NameError crashes trade placement

Files: `paper_routes.py` L389

Flagged by: Persona 13

Status: NEW finding

Problem:

```
log.warning(f"Could not capture trade context: {e}")
# NameError: name 'log' is not defined
```

The module uses `logger` (from `logging.getLogger(__name__)`). The `log` alias does not exist. Any exception during context capture in `place_trade()` raises `NameError`, aborting the entire trade placement.

Fix:

```
logger.warning(f"Could not capture trade context: {e}")
```

Impact: A one-character typo crashes trade placement whenever the optional context capture fails. This is a latent P0 that manifests unpredictably.

P0

P0-10 · `_get_username()` falls back to `'demo'` — auth bypass

Files: `paper_routes.py` L47

Flagged by: Personas 13, 14

Status: NEW finding

Problem:

```
def _get_username():
    return session.get('user', 'demo')
```

Unauthenticated requests (no session cookie) silently operate as the `demo` user. All `demo` trades, settings, and portfolio data are accessible and writable without authentication.

Fix:

```
def _get_username():
    user = session.get('user')
    if not user:
        abort(401)
    return user
```

Impact: No authentication is required to read or manipulate all `demo` account data. If any real user has username `demo`, their data is fully exposed to the internet.

P0

P0-11 · `current_price` reset to 0 in `scan_ticker` — aborts on ORATS failure

Files: `hybrid_scanner_service.py` L282

Flagged by: Persona 1

Status: NEW finding

Problem:

```
current_price = 0
# ... 40 lines later ...
orats_quote = self.orats_client.get_quote(ticker)
if orats_quote:
    current_price = orats_quote['underlying_price']
# No else — current_price stays 0 if ORATS fails
```

If ORATS returns no quote, `current_price` is 0. Downstream calculations divide by `current_price` or compare against it. The scan either crashes or produces nonsense results — even when a valid stock price was obtained from a prior history call.

Fix:

```
# Use live ORATS quote as override; fall back to history price
history_price = current_price  # preserve the price from history call
orats_quote = self.orats_client.get_quote(ticker)
if orats_quote:
    current_price = orats_quote['underlying_price']
elif history_price == 0:
    logger.warning(f"No price available for {ticker}, skipping")
    return []
# else: keep history_price — do not reset to 0
```

Impact: Any ORATS quota hit or transient failure aborts that ticker's scan, even when historical price data is available.

P0

P0-12 · Target Friday calculation wrong on Mondays

Files: `hybrid_scanner_service.py` L857–859

Flagged by: Persona 1

Status: NEW finding

Problem:

```
days_to_friday = (3 - today.weekday()) % 7   # weekday() Monday=0, Friday=4
```

`3` should be `4` (Friday's weekday index). On Monday (`weekday()=0`), this returns `(3-0)%7 = 3`, pointing to Thursday. On Tuesday, `(3-1)%7 = 2`, pointing to Thursday. The formula is off by one day for all weekdays.

Fix:

```
days_to_friday = (4 - today.weekday()) % 7
```

Impact: Weekly and 0DTE scans run on Monday–Thursday target the wrong expiry date. Users scanning for Friday expirations on Monday get Thursday results.

P0

P0-13 · `calculate_base_score()` always anchors at ~50 — missing context keys

Files: `reasoning_engine.py` L246–248; `hybrid_scanner_service.py` L764

Flagged by: Persona 1

Status: NEW finding

Problem: `calculate_base_score()` reads `context['technicals']['score']` and `context['sentiment']` to anchor the AI's numerical score. Neither key is populated before the AI call:

```
# hybrid_scanner_service.py ~L764 — what actually gets sent:
context = {
    'ticker': ticker,
    'technicals': {
        'trend': ...,
        'ma_signal': ...,
        # 'score': MISSING
    },
    # 'sentiment': MISSING
}
```

Fix:

```
# hybrid_scanner_service.py ~L764 — populate before AI call
context['technicals']['score'] = technical_score    # 0-100 from scoring engine
context['sentiment'] = sentiment_score              # 0-100 from Finnhub
```

Impact: The AI base score anchors at approximately 50 for every ticker regardless of actual technical quality. A ticker with RSI=90, all MAs bullish, and strong momentum scores the same as one with RSI=20 in a downtrend. The AI conviction layer adds noise but no signal.

P0

P0-14 · Bookend snapshots capture stock price, not option price

Files: `monitor_service.py` L605–614

Flagged by: Personas 3, 4, 7, 11

Status: NEW finding

Problem:

```
# bookend open/close snapshot:
snapshot['price'] = current_stock_price   # e.g. $272.45
```

The bookend uses `current_stock_price` instead of calling `get_option_quote()`. Open/close snapshots record ~$272 for AAPL instead of the option premium (~$3.50). MFE (Maximum Favorable Excursion) and MAE (Maximum Adverse Excursion) calculations are completely wrong.

Fix:

```
# monitor_service.py bookend logic — same pattern as update_price_snapshots()
for trade in open_trades:
    option_quote = self.broker.get_option_quote(trade['symbol'])
    if option_quote:
        snapshot['price'] = option_quote['last']
    else:
        snapshot['price'] = trade['entry_price']   # fallback
```

Impact: All historical MFE/MAE data in the DB is corrupted. Performance attribution, trade review, and any ML training on this data produces nonsensical results.

P0

P0-15 · `W_PROF` weight mismatch — code still at 20%, spec says 5%

Files: `options_analyzer.py` (scoring weights constant)

Flagged by: Persona 1; cross-referenced with known issue #12 and architecture map

Status: KNOWN issue (#12) — but architecture map confirms **NOT yet fixed in code**

Problem: The expert recommendation to reduce profit weight from 20% to 5% was documented in `EXPERT_RECOMMENDATIONS.md` and the architecture map states the intended weight is 5%. The actual code constant `W_PROF = 0.20` has not been updated. The sum of all weights is therefore > 1.0.

Fix:

```
# options_analyzer.py — scoring weights
W_TECH = 0.35   # Technical
W_MTA  = 0.20   # Multi-Timeframe Analysis
W_SENT = 0.15   # Sentiment
W_SKEW = 0.15   # Skew
W_LIQ  = 0.10   # Liquidity
W_PROF = 0.05   # Profit potential (was 0.20 — fix to match spec)
# Sum = 1.00 ✓
```

Impact: Profit potential is quadruple-weighted vs. specification. High-premium OTM garbage scores well; quality LEAPs with conservative profit targets score poorly. The scoring function does not match documented design.

P0

P0-16 · Max positions not enforced server-side

Files: `paper_routes.py` (place_trade route)

Flagged by: Personas 2, 8

Status: NEW finding

Problem: `max_positions` is stored in user settings but never queried during trade placement. The frontend may show a warning, but any API call bypasses it.

Fix:

```
# paper_routes.py — place_trade(), before order submission
open_count = db.execute(
    "SELECT COUNT(*) as cnt FROM paper_trades "
    "WHERE username=? AND status IN ('PENDING','OPEN')",
    (username,)
).fetchone()['cnt']
max_pos = int(user_settings.get('max_positions', 10))
if open_count >= max_pos:
    return jsonify({'error': f'Max positions ({max_pos}) reached'}), 429
```

Impact: Users can open unlimited concurrent positions regardless of their configured risk limit.

P0

P0-17 · LEAP scan bull-only — all put opportunities suppressed

Files: `hybrid_scanner_service.py` L260–262

Flagged by: Personas 3, 9, 10

Status: NEW finding

Problem:

```
if current_price < sma_200:
    continue   # skips ALL tickers below 200-day SMA
```

This check is applied universally before option direction is determined. Long puts on downtrending tickers (the most natural LEAP put trade) are discarded before analysis begins.

Fix:

```
# Apply the SMA filter directionally, not universally
if option_direction == 'call' and current_price < sma_200:
    continue    # bullish call: require uptrend
if option_direction == 'put' and current_price > sma_200:
    continue    # bearish put: require downtrend
```

Impact: The LEAP scanner is structurally long-only. In bear markets or sector rotations, zero put opportunities are surfaced. Half the options market is invisible to the system.

Quick Wins (Under 30 Minutes Each)

These fixes are individually small but collectively high-value. Each can be done in isolation.

ID	Fix	File	Time	Severity
QW-1	`log.warning` → `logger.warning` (1 char)	`paper_routes.py` L389	1 min	P0

ID	Fix	File	Time	Severity
QW-2	`(3 - today.weekday()) % 7` → `(4 - today.weekday()) % 7`	`hybrid_scanner_service.py` L858	1 min	P0
QW-3	`FLASK_DEBUG` default `'True'` → `'False'`	`config.py` L56	1 min	P0
QW-4	Add `lifecycle_sync` APScheduler job (5 lines)	`app.py`	5 min	P0
QW-5	Add `timeout=(5, 30)` to all `orats.py` requests	`orats.py` L82, L109, L161	5 min	P1
QW-6	Delete duplicate `analyze_volume()` call	`technical_indicators.py` L322–323	1 min	P1
QW-7	Delete dead scoring formula block (first `opportunity_score = (...)`)	`options_analyzer.py` L439–445	2 min	P1
QW-8	Fix `min_profit_potential` double assignment in `__init__`	`options_analyzer.py`	1 min	P1
QW-9	Fix `ma_signal` key: `'pullback bullish'` → `'pullback_bullish'`	`hybrid_scanner_service.py` L1169	1 min	P1
QW-10	Fix put break-even formula: `strike + premium` → `strike - premium`	`options_analyzer.py`	5 min	P1
QW-11	Replace all `print()` debug statements with `logger.debug()`	`technical_indicators.py` L203, L328; `hybrid_scanner_service.py` L1104–1106; `reasoning_engine.py` L301; `orats.py` L311	10 min	P2
QW-12	Remove unreachable `return final_results` in `scan_watchlist`	`hybrid_scanner_service.py`	1 min	P2

ID	Fix	File	Time	Severity
QW-13	Fix `get_db()` to `yield db` in try/finally	`models.py` L86–92	5 min	P1
QW-14	Add `_get_username()` → `abort(401)` when no session	`paper_routes.py` L47	2 min	P0
QW-15	Add `W_PROF = 0.05` scoring weight fix	`options_analyzer.py`	1 min	P0

Total estimated time for all quick wins: ~42 minutes

(QW-1 through QW-15 can be batched into a single PR in under an hour.)

Cross-Cutting Concerns

These are issues that span multiple files or architectural layers. They require coordinated changes rather than single-line fixes.

XC-1 · No VIX regime filter anywhere in the scan pipeline

Flagged by: Personas 3, 9 (+ implicitly 1, 4)

Files: `hybrid_scanner_service.py` (entire scan flow)

The system uses identical buy thresholds at VIX=12 (cheap options) and VIX=40 (expensive options). Buying LEAPs into a VIX spike is the single most common way retail traders overpay for protection.

Recommended fix:

```
# At scan_ticker() entry point
vix = self._get_vix()  # via ORATS or Tradier /markets/quotes?symbols=VIX
if vix > 30:
    logger.info(f"VIX={vix:.1f} — blocking LEAP calls, raising score threshold")
    min_score_threshold = 75  # raise from 65
    allow_leap_calls = False
    allow_leap_puts = True   # bear market — puts make sense
elif vix > 20:
    min_score_threshold = 70
```

XC-2 · HybridScannerService God Class (1823 lines)

Flagged by: Personas 2, 9, 12, 13, 15, 16

Files: `hybrid_scanner_service.py`

The file contains: scan orchestration, scoring, data fetching, strike selection, calendar logic, MA calculations, sentiment calls, AI calls, and output formatting. At 1823 lines, it has too many responsibilities to test or refactor safely.

Recommended decomposition:

- `ScanOrchestrator` — top-level scan flow
- `ScoreCalculator` — all scoring logic
- `OptionFilter` — strike/expiry selection
- `DataFetcher` — ORATS/Tradier/Finnhub calls
- `CalendarUtils` — date/expiry calculations

This is a P2 (next sprint) item but blocks testability of all P0 fixes above.

XC-3 · Zero test coverage for scanner, ORATS, and options analysis

Flagged by: Personas 12, 13, 15

Files: `tests/` directory

The test suite covers the paper trading HTTP layer but has zero unit tests for:

- `HybridScannerService`
- `OptionsAnalyzer`
- `TechnicalIndicators`
- `OratskClient`
- `ReasoningEngine`

All 17 P0 fixes above are currently untestable. Any regression from a fix will not be caught.

Recommended immediate additions:

1. `test_process_expiration.py` — verify multiple expiries are returned (catches P0-1 regression)
2. `test_scoring_weights.py` — verify weights sum to 1.0
3. `test_target_friday.py` — parametric test for all weekdays
4. `test_earnings_check.py` — mock Finnhub, verify has_earnings_risk

XC-4 · Missing input validation on all API routes

Flagged by: Personas 13, 14

Files: `paper_routes.py`, `scanner_routes.py`

No route validates ticker format, quantity bounds, or strike prices. A malformed ticker (e.g., `"; DROP TABLE paper_trades;--"` or a 500-character string) propagates directly to DB queries and external API calls.

Recommended fix: Add a `validate_ticker()` decorator and quantity/price bound checks at route entry.

XC-5 · `calculate_hv_rank` mutates shared DataFrame

Flagged by: Personas 1, 13

File: `technical_indicators.py` L463–493

`calculate_hv_rank()` adds columns to the DataFrame in-place. If the same DataFrame object is reused across tickers (e.g., in a batch scan), historical volatility columns from ticker A pollute ticker B's calculations.

Fix: `df = df.copy()` at the top of `calculate_hv_rank()`.

XC-6 · Sentiment and technical scores not passed to AI reasoning

Flagged by: Persona 1

Files: `reasoning_engine.py` L246–248; `hybrid_scanner_service.py` L764

See P0-13. This is also an architectural issue: the ReasoningEngine is designed to receive quantitative anchors from the scanner, but the scanner never populates them. The AI layer and the scoring layer are effectively disconnected.

XC-7 · `users.json` auth — no rate limiting, no lockout, plaintext sessions

Flagged by: Personas 12, 13, 14

Files: Auth routes, `config.py`

- No brute-force protection on `/login`
- Session token is Flask's default (HMAC-signed cookie — acceptable if `SECRET_KEY` is strong, but see P0-5)
- `users.json` means user management requires filesystem access; no API to add/change users
- Passwords stored as bcrypt hashes (good) but the JSON file has no access controls

Recommended fix (ordered by priority):

1. P0-5 fix (SECRET_KEY enforcement) — already listed
2. Add 5 attempts/minute rate limit to `/login` via Flask-Limiter
3. Migrate to SQLite users table (matches existing pattern)

XC-8 · No backup/DR strategy for Raspberry Pi SQLite deployment

Flagged by: Personas 12, 15

All scanner state and (if PostgreSQL goes down) fallback data lives on a Raspberry Pi SD card. SD cards have ~2,000–10,000 write cycles; a database with frequent writes degrades the card in months. No backup job exists.

Recommended fix:

1. Enable SQLite WAL mode: `PRAGMA journal_mode=WAL;`
2. Daily rsync of SQLite file to a network share or S3
3. Consider moving scanner state to PostgreSQL alongside paper trading

---

XC-9 · Confirmed known issues still present in code (status check)

The following issues from `known_issues_digest.md` were **confirmed still active** by persona reviews:

Known Issue	Status
#2 Greeks show 0 on weekends — no BS fallback	Still present (Personas 4, 7)
#3 Conviction score inconsistency — no numerical anchor	Still present (see P0-13)
#8 Liquidity analysis partial — LEAP scan has no spread/OI filter	Partial fix only (Persona 5)
#9 No risk/reward ratio display	Still present (Persona 9)
#11 Missing earnings context	Code regression confirmed (P0-3)
#12 Profit weight at 20% — expert says 5%	Not fixed in code (P0-15)
#13 .env.example stale	Still present (Persona 13)

Issues #1 (FMP 403), #4 (missing CSS), #5 (duplicate 0DTE route), #6 (dead files), #7 (debug startup print) were **not specifically re-confirmed** by personas and may be fixed.

Findings by Persona

Persona 1 — Quantitative Options Trader

Unique new findings: P0-1, P0-2(partial), P0-11, P0-12, P0-13, P0-14(partial), P0-15, CC-18

Key observation: "The scoring engine is built on correct foundations but has four implementation errors that make every score wrong: skew frozen, profit 4× overweighted, AI gets no quantitative anchors, and profit calc ignores time value."

Additional findings (not listed above):

- **[P1]** Profit calculation is intrinsic-only, ignores time value — `options_analyzer.py` L256–302. Use delta-gamma approximation: `P ≈ deltadS + 0.5gamma*dS²`
- **[P1]** `analyze_volume()` in `TechnicalIndicators` double-called — `technical_indicators.py` L322–323 (see QW-6)
- **[P2]** SMA crossover uses close prices only; should use `(H+L+C)/3` typical price for more robust signal

---

Persona 2 — Risk Manager

Unique new findings: P0-5(partial), P0-6, P0-7, P0-16, CC-5 (daily loss limit)

Key observation: "Three independent risk controls (position limits, daily loss limits, stop-loss quality) are all broken. The system has the UI for risk management but not the enforcement."

Additional findings:

- **[P1]** No position-level max loss calculation before trade entry — system should reject trades where `max_loss > account_value * max_risk_per_trade`
- **[P1]** OCO orders never verified after placement — no confirmation loop checking Tradier order status
- **[P2]** Kelly criterion not implemented despite being mentioned in scoring comments
- **[P2]** No correlation check between open positions — concentrated sector risk invisible

---

Persona 3 — Volatility Regime Specialist

Unique new findings: P0-3(partial), P0-17, XC-1 (VIX filter)

Key observation: "The system treats volatility as a static input rather than a regime. Buying expensive options in high-VIX environments while selling cheap ones in low-VIX is the opposite of what a vol-aware system should do."

Additional findings:

- **[P1]** IV percentile (IVP) threshold is static at 50 — should be regime-adjusted: IVP<30 for longs in low-vol regime, IVP>70 for shorts
- **[P1]** No IV term structure analysis — steep backwardation (VIX futures) not detected
- **[P2]** Realized volatility (HV) vs. implied volatility (IV) spread not used in scoring despite being available from ORATS
- **[P3]** No VIX futures term structure fetch to detect contango/backwardation regime

---

Persona 4 — Weekly Options Specialist

Unique new findings: P0-1(partial), P0-14(partial), QW-9, QW-10

Key observation: "Weekly options require precision. The Friday targeting bug, the SMA-200 check killing all put opportunities, and the wrong break-even formula combine to make the weekly scanner produce misleading results on the majority of trading days."

Additional findings:

- **[P0]** `sma_5` lookup crashes on IPO/ETF tickers with <200 bars: `hybrid_scanner_service.py` L1169 — fix: `ma_vals = indicators.get('moving_averages', {}).get('values') or {}`
- **[P1]** 0DTE filter allows expirations 1–7 days out — true 0DTE should be today's date only
- **[P1]** Weekly scan doesn't check Monday holiday calendar — scans for Friday options that don't exist
- **[P2]** Delta targeting for weekly strikes uses fixed range (0.30–0.50) regardless of DTE — should widen to 0.20–0.60 for 0DTE

Persona 5 — Value Hunter

Unique new findings: P0-2(partial), QW-7 (dead scoring block)

Key observation: "LEAP scoring overfits to momentum. Value signals (PE ratio, earnings yield, sector mean-reversion) are not present in the scoring formula. The system would never recommend a LEAP on a

temporarily beaten-down quality company."

Additional findings:

- **[P1]** Dead `opportunity_score` formula block (`options_analyzer.py` L439–445) contains old weights — delete it (see QW-7)
- **[P1]** FMP fundamentals integrated but PE ratio not used in LEAP scoring — high-PE momentum names dominate recommendations
- **[P2]** No sector-relative valuation — absolute P/E threshold same for tech (30×) and utilities (15×)
- **[P3]** No mean-reversion signal for value plays — 52-week low proximity not scored

---

Persona 6 — Timing & Catalysts Analyst

Unique new findings: P0-3(partial), QW-6(partial)

Key observation: "The system is blind to catalysts — earnings, FDA dates, Fed meetings, index rebalancing. Entering a LEAP 2 days before earnings is not a timing mistake; it's a system failure."

Additional findings:

- **[P1]** No Fed meeting calendar check — options often expensive 2 days before FOMC
- **[P1]** No ex-dividend date check — assigned early exercise risk on ITM calls before ex-div not detected
- **[P2]** Analyst upgrade/downgrade momentum not incorporated despite Finnhub having this endpoint
- **[P2]** Sector rotation signals (sector ETF relative strength) not tracked

---

Persona 7 — Order Flow Expert

Unique new findings: P0-6(partial), P0-7(partial), P0-14(partial)

Key observation: "The OCO implementation has two independent critical bugs — key mismatch and market-order stops. Either one alone would be dangerous. Together they mean every stop-loss in the system is either silently unenforced or uses market orders that can fill at catastrophic prices."

Additional findings:

- **[P1]** No dark pool / unusual options activity signal — large block trades not detected
- **[P1]** Bid-ask spread not factored into entry slippage estimate — $0.10 wide spread on $1.00 option is 10% immediate loss
- **[P2]** No time-of-day filter — scans can recommend entries at market open (illiquid) or close (volatile)
- **[P2]** NBBO validation not performed — fills may execute outside NBBO on Tradier

---

Persona 8 — Portfolio Behavior Analyst

Unique new findings: P0-4(partial), P0-16(partial)

Key observation: "Portfolio-level risk controls exist in the UI spec but none are enforced at the API layer. The gap between spec and implementation is total."

Additional findings:

- **[P1]** No sector concentration limit — system can put 80% of paper portfolio in single sector
- **[P1]** Beta-weighted delta exposure not calculated across positions
- **[P2]** Sharpe ratio calculation uses arithmetic return — should use log return for options
- **[P2]** Win rate stat includes open trades with unrealized gains — inflates reported win rate

---

Persona 9 — Macro Regime Analyst

Unique new findings: P0-17(partial), XC-1(partial)

Key observation: "Macro regime is the single biggest determinant of LEAP success. A LEAP bought in a bull market with accommodative Fed policy is a different instrument than the same LEAP bought in a rate-hiking recession. The system treats both identically."

Additional findings:

- **[P1]** No yield curve data — inverted 2s10s not detected despite being predictive of sector rotation
- **[P1]** Dollar index (DXY) not factored — strong dollar headwind for international-revenue stocks
- **[P2]** Credit spreads (HYG/LQD ratio) not monitored — early warning of liquidity stress
- **[P3]** No economic calendar integration — ISM, CPI, NFP dates not blocked for high-sensitivity trades

---

Persona 10 — Alpha Leakage Detector

Unique new findings: P0-3(partial), P0-17(partial)

Key observation: "Alpha leaks from four sources: dead skew signal (15% weight wasted), bull-only scan (missing 50% of opportunities), earnings blindness (catastrophic timing), and the return-in-loop bug (scanner output truncated). Fixing these four restores more alpha than any new feature."

Additional findings:

- **[P1]** Liquidity score uses OI threshold (100 contracts) that is too low for LEAPs — should be 500+
- **[P1]** Put/call ratio signal directional bias not incorporated despite ORATS providing it
- **[P2]** No analyst consensus tracking — stocks being upgraded by multiple analysts have alpha
- **[P2]** Earnings growth trend (accelerating EPS) not scored

---

Persona 11 — Data Integrity Auditor

Unique new findings: P0-14 (bookend snapshot)

Key observation: "The trade database has a systemic data quality issue: bookend snapshots record stock prices instead of option prices, making all MFE/MAE/P&L analytics meaningless. Any backtesting or ML training on this data

should be halted until the data is corrected."

Additional findings:

- **[P1]** No data lineage tracking — impossible to audit which ORATS call produced which scan result
- **[P1]** `trade_context` JSONB field has no schema validation — malformed JSON stored silently
- **[P2]** Price snapshot table has no index on `(trade_id, snapshot_time)` — slow queries at scale
- **[P2]** No stale data detection — scan results cached indefinitely in SQLite with no TTL

Persona 12 — Senior Architect

Unique new findings: P0-5(partial), XC-2 (God class), XC-3 (no tests), XC-8 (Raspberry Pi backup), QW-13

Key observation: "The architecture has three existential risks: the God class makes safe refactoring nearly impossible, zero scanner test coverage means every bug fix risks regressions, and a Raspberry Pi SD card with no backup is a single point of failure for production data."

Additional findings:

- **[P1]** `get_db()` closes session before returning — `models.py` L86–92 (see QW-13)
- **[P1]** No connection pooling for PostgreSQL — each request opens a new connection under load
- **[P2]** Flask runs synchronously — long ORATS calls block other requests; consider async or task queue
- **[P2]** No health check endpoint — Docker restart policy can't distinguish hung app from crash

---

Persona 13 — Senior Developer

Unique new findings: P0-8(partial), P0-9, P0-10, CC-18(partial), QW-11, QW-12

Key observation: "Code quality issues cluster around two patterns: missing error handling that turns warnings into crashes, and dead code that creates maintenance hazards. The `log` vs `logger` bug is the most embarrassing: a single character that crashes trade placement."

Additional findings:

- **[P1]** `.env.example` stale — missing 4 required vars, has 6 dead vars (known issue #13)
- **[P1]** No structured logging — log format is freeform strings, making log aggregation impossible
- **[P2]** `hybrid_scanner_service.py` has 12 methods over 100 lines each — cyclomatic complexity very high
- **[P2]** No type hints on any public methods — IDE assistance and static analysis ineffective

---

Persona 14 — Security Engineer

Unique new findings: P0-5(partial), P0-10(partial), XC-4, XC-7

Key observation: "Four distinct security vulnerabilities, two of which are P0: RCE via debug mode and auth bypass via username fallback. These are not theoretical — they are exploitable in the current deployment with no special knowledge required."

Additional findings:

- **[P0]** Fernet `ENCRYPTION_KEY` not validated at startup — if key is wrong format, Fernet raises `ValueError` at first broker token decrypt, crashing all broker operations
- **[P1]** API keys logged at `DEBUG` level — `ORATS_API_KEY` appears in logs if debug mode is on
- **[P1]** No CSRF protection on state-changing routes — POST `/api/paper/trade` vulnerable to CSRF
- **[P2]** Docker container runs as root — should use non-root user in Dockerfile

---

Persona 15 — Reliability & Testing Engineer

Unique new findings: P0-8, QW-5, XC-3(partial), XC-8(partial)

Key observation: "The system has no resilience layer. Every external API call (ORATS, Tradier, Finnhub, Perplexity) can hang indefinitely. Every failure cascades. `lifecycle_sync` never running means the system degrades silently over days without any alert."

Additional findings:

- **[P1]** No retry logic on any external API call — transient ORATS failure fails entire scan, not just that ticker
- **[P1]** No circuit breaker pattern — ORATS outage causes every scan to fail for its full timeout duration
- **[P2]** APScheduler `max_instances=1` not set — overlapping job executions possible under load
- **[P2]** No alerting on scanner errors — failures are logged but no notification mechanism exists

---

Persona 16 — AI/UX Specialist

Unique new findings: XC-6(partial)

Key observation: "The AI reasoning layer is technically integrated but functionally disconnected. It receives no quantitative anchors, so its score adjustments are noise. The UX implications are serious: users trust AI conviction scores that are essentially random."

Additional findings:

- **[P1]** AI explanation is not linked to the factors that drove the score — users see 'Strong bullish momentum' but the actual score drivers (technical=72, sentiment=45, skew=50) are not displayed
- **[P1]** No confidence interval on AI score — presenting `73` implies precision that doesn't exist
- **[P2]** Perplexity `sonar-pro` used for scoring — `sonar` (faster, cheaper) sufficient for structured scoring tasks; `sonar-pro` better reserved for nuanced open-ended analysis
- **[P2]** Score breakdown (technical/sentiment/skew/etc.) not surfaced in the UI — users cannot understand what drove a recommendation
- **[P3]** No explanation when a trade is rejected (score < 40) — users don't know what to fix
- **[P3]** AI score not cached — same ticker re-analyzed on every scan run, burning API credits

---

Appendix: All Findings Summary Table

ID	Description	File(s)	Severity	Personas	Status
P0-1	`_process_expiration` return inside for-loop	`options_analyzer.py` L229–254	P0	1,2,4,7,13	New
P0-2	Skew calculation dead — always returns 50	`hybrid_scanner_service.py` L423–431; `options_analyzer.py` L354	P0	1,2,3,4,5,9,10	New
P0-3	Earnings risk check stripped — always False	`hybrid_scanner_service.py` L1030–1032	P0	1,2,3,4,6,9,10	Known+regression
P0-4	Daily loss limit never enforced	`paper_routes.py` L333–348	P0	2,8	New
P0-5	`FLASK_DEBUG` defaults to True	`config.py` L56	P0	12,13,14	New
P0-6	OCO `adjust_bracket()` key mismatch — silent fail	`monitor_service.py` L851–868	P0	2,7	New
P0-7	OCO stop-loss uses market order not stop_limit	`tradier.py` L408–423	P0	2,7	New
P0-8	`lifecycle_sync()` never scheduled	`app.py`	P0	15	New
P0-9	`log` vs `logger` NameError in place_trade	`paper_routes.py` L389	P0	13	New
P0-10	`_get_username()` falls back to 'demo' — auth bypass	`paper_routes.py` L47	P0	13,14	New
P0-11	`current_price` reset to 0 on ORATS failure	`hybrid_scanner_service.py` L282	P0	1	New
P0-12	Target Friday formula wrong on Mondays	`hybrid_scanner_service.py` L857–859	P0	1	New

ID	Description	File(s)	Severity	Personas	Status
P0-13	`calculate_base_score()` missing context keys	`reasoning_engine.py` L246–248; `hybrid_scanner_service.py` L764	P0	1	New
P0-14	Bookend snapshots use stock price not option price	`monitor_service.py` L605–614	P0	3,4,7,11	New
P0-15	`W_PROF` weight 20% in code, spec says 5%	`options_analyzer.py`	P0	1	Known+unresolved
P0-16	Max positions not enforced server-side	`paper_routes.py`	P0	2,8	New
P0-17	LEAP scan bull-only — puts suppressed	`hybrid_scanner_service.py` L260–262	P0	3,9,10	New
QW-1	`log` → `logger` (1 char)	`paper_routes.py` L389	P0	13	New
QW-2	Friday formula: `3` → `4`	`hybrid_scanner_service.py` L858	P0	1	New
QW-3	`FLASK_DEBUG` default `'False'`	`config.py` L56	P0	12,13,14	New
QW-4	Add `lifecycle_sync` APScheduler job	`app.py`	P0	15	New
QW-5	Add `timeout=(5,30)` to ORATS requests	`orats.py` L82,L109,L161	P1	15	New
QW-6	Delete duplicate `analyze_volume()` call	`technical_indicators.py` L322–323	P1	6,9,10,13	New
QW-7	Delete dead scoring formula block	`options_analyzer.py` L439–445	P1	1,2,4,5,9,10,13	New
QW-8	Fix `min_profit_potential` double assignment	`options_analyzer.py`	P1	1	New

ID	Description	File(s)	Severity	Personas	Status
QW-9	Fix `ma_signal` key: `'pullback bullish'` → `'pullback_bullish'`	`hybrid_scanner_service.py` L1169	P1	4	New
QW-10	Fix put break-even: `strike + premium` → `strike - premium`	`options_analyzer.py`	P1	4	New
QW-11	Replace `print()` with `logger.debug()`	`technical_indicators.py`, `hybrid_scanner_service.py`, `reasoning_engine.py`, `orats.py`	P2	4,6,9,10,13,15	Known (#7 partial)
QW-12	Remove unreachable `return final_results`	`hybrid_scanner_service.py`	P2	13	New
QW-13	Fix `get_db()` to `yield db` pattern	`models.py` L86–92	P1	12	New
QW-14	`_get_username()` → `abort(401)` when no session	`paper_routes.py` L47	P0	13,14	New
QW-15	Fix `W_PROF = 0.05` scoring weight	`options_analyzer.py`	P0	1	Known+unresolved
XC-1	No VIX regime filter	`hybrid_scanner_service.py`	P1	3,9	New
XC-2	God class 1823 lines — no separation of concerns	`hybrid_scanner_service.py`	P2	2,9,12,13,15,16	New
XC-3	Zero test coverage for scanner/ORATS/options analysis	`tests/`	P1	12,13,15	New
XC-4	No input validation on API routes	`paper_routes.py`, `scanner_routes.py`	P1	13,14	New

ID	Description	File(s)	Severity	Personas	Status
XC-5	`calculate_hv_rank` mutates shared DataFrame	`technical_indicators.py` L463–493	P1	1,13	New
XC-6	AI reasoning disconnected from quantitative scores	`reasoning_engine.py`, `hybrid_scanner_service.py`	P0	1,16	New
XC-7	Auth: no rate limiting, no lockout	Auth routes	P1	12,13,14	New
XC-8	No backup/DR for Raspberry Pi SQLite	Deployment	P2	12,15	New
XC-9	Known issues #2,3,8,9,11,12,13 still present	Various	P0–P2	Multiple	Known+unresolved
P1-A1	Profit calc intrinsic-only — ignores time value	`options_analyzer.py` L256–302	P1	1,3,4	New
P1-A2	IVP threshold static — not regime-adjusted	`hybrid_scanner_service.py`	P1	3	New
P1-A3	No ex-dividend date check for early exercise risk	External calendar	P1	6	New
P1-A4	OI threshold (100) too low for LEAP liquidity	`hybrid_scanner_service.py`	P1	10	New
P1-A5	No position-level max loss check before entry	`paper_routes.py`	P1	2	New
P1-A6	No OCO order confirmation loop after placement	`monitor_service.py`	P1	2	New
P1-A7	No retry logic on any external API call	`orats.py`, `tradier.py`, `finnhub.py`	P1	15	New
P1-A8	No connection pooling for PostgreSQL	`models.py`	P1	12	New
P1-A9	No sector concentration limit	`paper_routes.py`	P1	8	New

ID	Description	File(s)	Severity	Personas	Status
P1-A10	`sma_5` lookup crashes on IPO/ETF tickers (<200 bars)	`hybrid_scanner_service.py` L1169	P1	4	New
P1-A11	AI score explanation not linked to score drivers	UI / `reasoning_engine.py`	P1	16	New
P1-A12	Fernet `ENCRYPTION_KEY` not validated at startup	`app.py` / `config.py`	P1	14	New
P1-A13	No CSRF protection on state-changing routes	`paper_routes.py`	P1	14	New
P2-A1	0DTE filter allows expirations up to 7 days out	`hybrid_scanner_service.py`	P2	4	New
P2-A2	Win rate stat includes unrealized open trades	`paper_routes.py` analytics	P2	8	New
P2-A3	APScheduler `max_instances=1` not set	`app.py`	P2	15	New
P2-A4	Perplexity `sonar-pro` used for scoring — overkill	`reasoning_engine.py`	P2	16	New
P2-A5	AI score not cached between scan runs	`reasoning_engine.py`	P2	16	New
P2-A6	Score breakdown not surfaced in UI	Frontend	P2	16	New
P2-A7	Docker container runs as root	`Dockerfile`	P2	14	New
P2-A8	No stale data detection / TTL in SQLite cache	`hybrid_scanner_service.py`	P2	11	New
P3-A1	No Fed meeting calendar check	External calendar	P3	6	New
P3-A2	No yield curve / DXY macro signal	External data	P3	9	New
P3-A3	No explanation shown when trade rejected (score<40)	Frontend / `reasoning_engine.py`	P3	16	New

Report generated: February 26, 2026 — Consolidation of 16 persona reviews by Project Manager agent

Methodology: All 16 review files read in full; findings deduplicated by file+line+description; severity normalized to P0–P3 scale; known issues cross-referenced against `known_issues_digest.md`