Silver Expert Core - Implementation Plan

A modular, rule-based trading workflow for Silver (XAGUSD / MCX Silver). Non-repainting, structured as a pipeline with clear IF/ELSE gates.

User Review Required

IMPORTANT

Target Platform Clarification Needed

- What is the target platform for this workflow?
  - **TradingView (Pine Script)**
  - **MetaTrader (MQL4/MQL5)**
  - **Python (standalone analysis/backtesting)**
  - **Other platform** (please specify)

This implementation plan assumes **TradingView Pine Script v5** as the default. Please confirm or specify your preferred platform.

WARNING

Configuration Parameters Please review and confirm these default values:

- Max Spread Threshold: **0.05** (0.5 points for XAGUSD)
- Minimum Candle Range: **0.10** (ATR-relative)
- ADX Trend Threshold: **22** (higher = stronger trend)
- ADX Range Threshold: **18** (lower = ranging market)
- Minimum Trade Quality Score: **60** (out of 100)
- Minimum Risk-Reward Ratio: **1:2**
- Key Level Tolerance: **0.50** (distance from key level)

Proposed Changes

Project Structure

```
d:\Silver Brains\
├── SilverExpertCore.pine          # Main indicator/strategy file
├── config/
│   └── settings.pine              # Configuration constants
├── blocks/
│   ├── 01_market_validation.pine   # Block 1: Market Validation
│   ├── 02_volatility_session.pine  # Block 2: Volatility & Session
│   ├── 03_htf_bias.pine            # Block 3: HTF Bias Detection
│   ├── 04_regime_detection.pine    # Block 4: Market Regime
│   ├── 05_key_levels.pine          # Block 5: Key Level Engine
│   ├── 06_trend_setup.pine         # Block 6: Trend Setup Filter
│   ├── 07_entry_confirmation.pine  # Block 7: Entry Confirmation
│   ├── 08_trap_filter.pine         # Block 8: Manipulation Filter
│   ├── 09_risk_management.pine     # Block 9: Risk Management
│   ├── 10_trade_scoring.pine       # Block 10: Trade Quality
│   ├── 11_signal_output.pine       # Block 11: Final Signal
│   └── range_mode.pine             # Range Mode Pipeline
├── utils/
│   ├── indicators.pine             # Technical indicator helpers
│   ├── structure.pine              # Market structure utilities
│   └── time_utils.pine             # Session/time utilities
└── docs/
    └── README.md                   # Documentation
```

NOTE

In TradingView Pine Script, modular files are combined into a single script. The folder structure above represents logical organization for development. The final output will be a single `.pine` file.

Core Infrastructure

[NEW]

SilverExpertCore.pine
Main workflow orchestrator that:

- Imports all block modules
- Manages pipeline flow with IF/ELSE gates
- Outputs final BUY/SELL/NO_TRADE signals
- Handles alerts and visual display

Block 1: Market Validation

[NEW]

01_market_validation.pine

Gate	Logic
Symbol Check	`syminfo.ticker` contains "XAG" or "SILVER"
Spread Check	Current spread ≤ max threshold
Volatility Check	Candle range ≥ min volatility
Output	`VALID` or `REJECT` with reason

Block 2: Volatility & Session

[NEW]

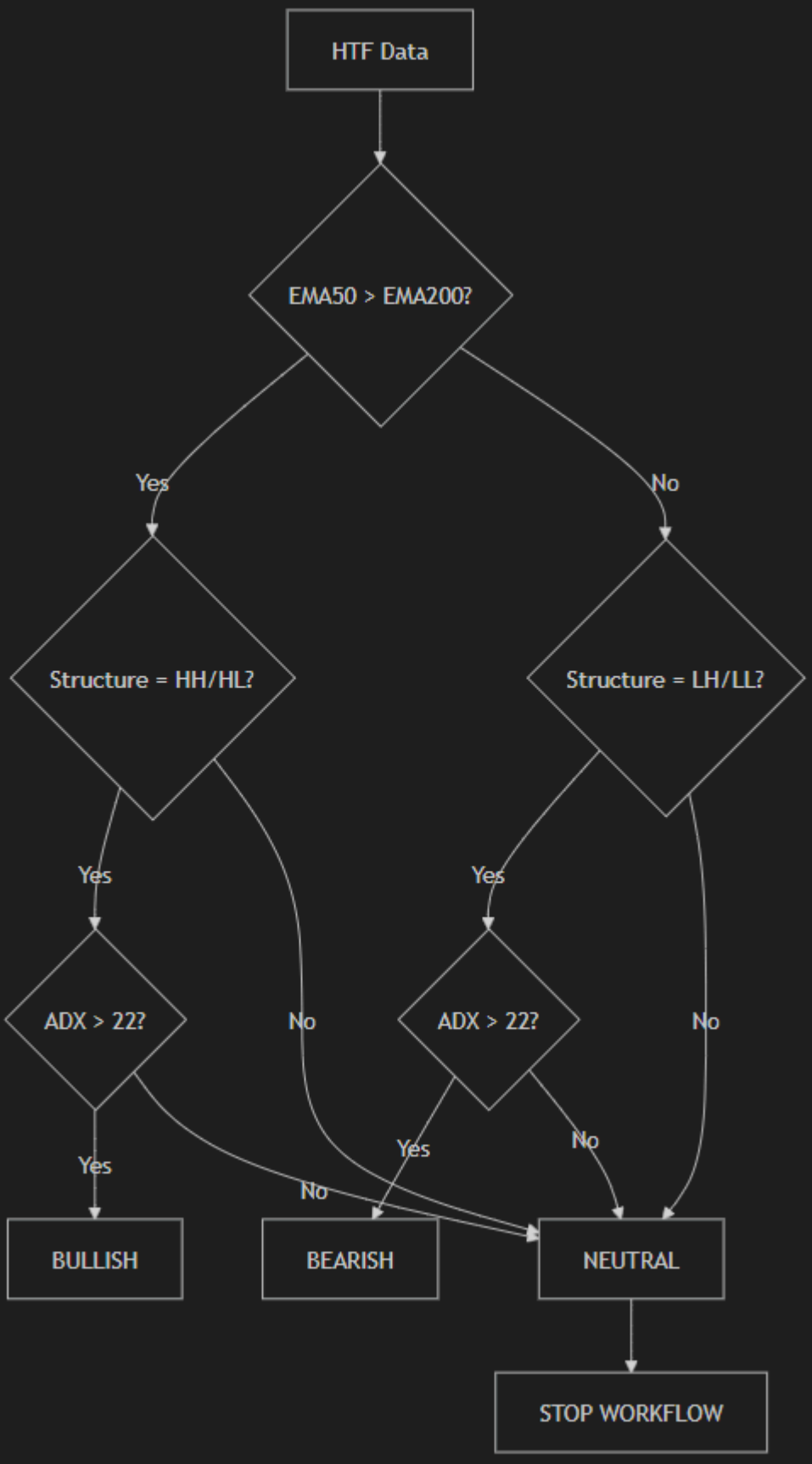
02_volatility_session.pine

Component	Implementation
ATR(14)	`ta.atr(14)`
Volatility State	LOW (<0.5×ATR), NORMAL (0.5-1.5×ATR), HIGH (>1.5×ATR)
Session Detection	Asian: 00:00-08:00 UTC, London: 08:00-16:00 UTC, NY: 13:00-22:00 UTC
Output	`volatility_state`, `current_session`, `allow_trade`

Block 3: HTF Bias Block

[NEW]

03_htf_bias.pine


HTF Data
EMA50 > EMA200?
Yes
No
Structure = HH/HL?
Structure = LH/LL?
Yes
Yes
ADX > 22?
No
ADX > 22?
No
Yes
No
Yes
No
BULLISH
BEARISH
NEUTRAL
STOP WORKFLOW

Block 4: Market Regime Detection

[NEW]

04_regime_detection.pine

Condition	Result
ADX < 18	RANGE MODE → Route to Range Pipeline
ADX ≥ 18 and < 22	Transitional → NEUTRAL (no trade)
ADX ≥ 22 + valid structure	TREND MODE → Route to Trend Pipeline

---

Block 5: Key Level Engine

[NEW]

05_key_levels.pine
Auto-detected levels:

- Previous Day High/Low (PDH/PDL)
- Previous Week High/Low (PWH/PWL)
- Session Open Price
- Psychological levels (e.g., 23.00, 23.50, 24.00)

Proximity check: Entry must be within `level_tolerance` of at least one key level.

Block 6: Trend Setup Filter

[NEW]

06_trend_setup.pine

Check	Logic
Direction alignment	Trade direction matches HTF_BIAS
Pullback required	Price retraced to EMA zone or key level
Opposite level block	No HTF resistance/support between entry and TP

Block 7: Entry Confirmation Engine

[NEW]

07_entry_confirmation.pine

Signal	Weight
RSI rebound (>30 for buy, <70 for sell)	25
RSI divergence	30
Volume spike (>1.5× average)	20
Strong candle close (body >60% of range)	15
Micro structure break (LTF)	10
Minimum required:	**≥ 50**

Block 8: Manipulation/Trap Filter

[NEW]

08_trap_filter.pine
Trap detection:

- Long wick beyond key level (>60% of candle range)
- Immediate reclaim (close back inside level)
- Weak follow-through (next candle smaller range)

Action: Delay entry by 1 candle for confirmation.

Block 9: Risk Management Engine

[NEW]

09_risk_management.pine

Component	Calculation
Stop Loss	Structure swing + buffer OR wick extreme

Take Profit 1	Nearest opposite key level
Take Profit 2	2× risk distance (minimum 1:2 RR)
RR Validation	Reject if RR < 1:2

Block 10: Trade Quality Scoring

[NEW]

10_trade_scoring.pine

Factor	Max Points
HTF trend strength (ADX-based)	25
Key level quality (type + proximity)	25
Confirmation count	20
Volatility state (NORMAL preferred)	15
Session alignment (London/NY preferred)	15
Total	**100**

Threshold: Score ≥ 60 to allow trade.

Block 11: Final Signal Output

[NEW]

11_signal_output.pine
Output structure:
Signal: BUY | SELL | NO_TRADE
Entry: [price]
Stop Loss: [price]
Take Profit 1: [price]
Take Profit 2: [price]
Risk-Reward: [ratio]
Trade Quality: [score]%

Volatility: [LOW|NORMAL|HIGH]
Session: [Asian|London|NY]
Reason: [validation message]

Range Mode Pipeline

[NEW]

range_mode.pine

Component	Implementation
Range detection	20-period high/low consolidation
Entry condition	Price at range extreme (top 10% or bottom 10%)
Confirmation	Rejection wick + RSI divergence + volume absorption
Position sizing	Flag for 50% reduced size

Utility Modules

[NEW]

indicators.pine

- EMA calculations
- RSI with divergence detection
- ADX calculation
- Volume analysis

[NEW]

structure.pine

- Swing high/low detection
- Market structure (HH/HL/LH/LL) identification
- Structure break detection

[NEW]

time_utils.pine

- Session detection
- Time-based filters
- News window handling

Global No-Trade Conditions

Applied across all modes:

- Major HTF level between entry and target → **BLOCK**
- Extreme volatility (ATR > 2× normal) → **BLOCK**
- Ultra-low volume (< 50% average) → **BLOCK**
- News window (configurable times) → **BLOCK**

Verification Plan

Automated Tests

Since this is a TradingView Pine Script indicator, automated unit tests are not directly available. Instead, verification will be done through:

1. **Pine Script Compiler Validation**
    - Run Pine Script compiler to check for syntax errors
    - Verify no compilation warnings
2. **Backtesting on TradingView**
    - Apply indicator to XAGUSD chart
    - Verify signals are generated correctly
    - Check non-repainting behavior by comparing real-time vs historical signals

Manual Verification

TIP

For the user to verify:

1. **Symbol Validation Test**
    - Apply indicator to XAGUSD → should show signals
    - Apply indicator to EURUSD → should show "INVALID SYMBOL" message
2. **Session Detection Test**
    - Check indicator during Asian session → should show "Asian"
    - Check during London session → should show "London"
    - Check during NY session → should show "New York"
3. **Signal Quality Test**
    - Wait for a live signal
    - Verify all components are displayed (SL, TP, RR, Quality Score)
    - Check that blocked trades show reason
4. **Key Level Accuracy**
    - Verify PDH/PDL lines match actual previous day
    - Verify PWH/PWL lines match actual previous week

Next Steps After Approval

1. Create project folder structure
2. Implement core configuration and types
3. Build blocks 1-4 (foundation)
4. Build blocks 5-8 (trend mode)
5. Build blocks 9-11 (risk/scoring/output)
6. Implement range mode pipeline
7. Combine into single Pine Script file
8. Test and refine