XAUUSD Confluence & Risk Management Plan

1. Confluence Logic (Split as Primary & Secondary)

Primary Confluences (must have for all strategies)

* Market Structure: Presence of BOS (Break of Structure) or CHoCH (Change of Character) in the trade direction on your entry timeframe.
* Candle Pattern: Confirming pattern at the entry (Engulfing, Pin Bar, Morning/Evening Star, Doji, Hammer).
* Support/Resistance (S/R) Zone: Trade must be at, or reacting to, a mapped S/R zone or order block.
* Trend Regime: Entry must align with the higher timeframe trend or be a fully confirmed counter-trend fade.

Secondary Confluences (boost confidence, not mandatory)

* Psychological Level: Entry reacts to a round number (e.g., 1950, 2000) or $5/$10 minor level (especially for scalping).
* Fibonacci Zone: Entry is near a key Fibonacci retracement/extension level.
* Volume Confirmation: Entry occurs with a volume spike (breakout or rejection), confirming participation.
* Liquidity Zone: Recent sweep, fair value gap, or other visible liquidity event.
* Spread Condition: For scalps, spread must be <0.2%.

Confluence Scoring

* Requirement:  
  *All 4 primary confluences* are required to enter a trade.  
  *At least 2 secondary confluences* should be present for the highest conviction, but if all primary confluences are met, you can still take the trade even if no secondary confluence is present. Secondary confluences are a bonus, not mandatory.
* Risk scaling: More confluences (including secondary) allow higher risk tiers within your risk management grid.

2. Risk & Position Management Logic

Risk per Trade

* Base risk: 0.50% of account equity per trade.
* Adjustment: Reduce risk/lot size by 20% whenever the account is down by 20%. Increase risk incrementally as the account recovers, returning to normal once initial balance is restored.
* Formula:  
  position\_risk = account\_equity × 0.005
* Confluence Adjustment: Only risk the full 0.5% if all minimum confluence conditions are met.

Position Size Calculation

* Steps:
  1. Calculate dollar risk: account\_equity × risk%
  2. Determine stop loss distance: abs(entry price - stop loss price)
  3. Add spread to stop loss distance
  4. Compute lot size:  
     lot\_size = position\_risk / ((stop\_loss\_distance + spread) × pip\_value)  
     (Where pip\_value is the dollar value per point/pip for XAUUSD, usually $100 per $1 move per standard lot)

SL/TP Logic (1:2 RRR)

* Stop Loss (SL):  
  Set using ATR (Average True Range) to adapt to volatility:
  + For Longs: SL = entry price - ATR × multiplier (e.g., multiplier = 1)
  + For Shorts: SL = entry price + ATR × multiplier
* Take Profit (TP):
  + Always set at 2 × stop loss distance from entry (1:2 RR).
  + For Longs: TP = entry price + 2 × (entry price - SL)
  + For Shorts: TP = entry price - 2 × (SL - entry price)

Breakeven Rule

* Trigger:  
  When price has moved 70% of the distance from entry to TP in your favor, move stop loss to entry price (breakeven).
* Example:  
  Entry at 1950, SL at 1948, TP at 1954:
  + TP distance = 4 points
  + 70% move = 2.8 points
  + Once price hits 1952.8, adjust SL to 1950 (breakeven)

Risk Management Tiers

* In profit: 1% risk (max confluence only)
* At baseline: 0.5% risk (standard mode)
* Drawdown > 50%: 0.2% risk (A+ setups only, avoid noise)
* *Risk tier always respects current confluence count and account status.*

3. Operational Discipline (Execution Rules)

* No trade without all 4 primary confluences. Secondary confluences are a bonus and only increase risk tier if present.
* Never exceed risk per trade—no exceptions.
* All entries, exits, SL/TP, and breakeven adjustments must be logged.
* Monthly performance and risk review required.

**1. INPUTS (per new bar or trade signal)**

* Current account equity
* Entry price
* ATR value (for current timeframe)
* Spread (in $)
* pip\_value (e.g., $100 per $1 move per standard lot for XAUUSD)
* All confluence signals (as bool/int for each)
  + Market Structure (BOS/CHoCH)
  + Candle Pattern (Engulfing, Pin Bar, Doji, Hammer, etc.)
  + S/R Zone
  + Trend Regime
  + Psychological Level
  + Fibonacci Zone
  + Volume Confirmation
  + Liquidity Zone
  + Spread Condition

**2. CONFLUENCE SCORING**

**Step 2.1: Compute Primary Confluence Score**

* Score 1 for each of:
  + Market Structure
  + Candle Pattern
  + S/R Zone
  + Trend Regime
* *Primary met if all 4 = True.*

**Step 2.2: Compute Secondary Confluence Count**

* Score 1 for each of:
  + Psychological Level
  + Fibonacci Zone
  + Volume Confirmation
  + Liquidity Zone
  + Spread Condition
* *Secondary count = sum of True values*

**Step 2.3: Confluence Logic**

* **IF all 4 primary confluences are True:**
  + Trade is allowed, even if all secondary confluences are False.
  + If 2 or more secondary confluences are also True, allow for higher risk tier (see below).

**3. RISK TIER SELECTION**

* If account is in profit: risk = 1%
* If account at baseline: risk = 0.5%
* If account down >50%: risk = 0.2%
* **Adjustment:**
  + Reduce risk/lot size by 20% if account is down by 20%.
  + Return to normal as account recovers.
* **Final risk % is further limited if not all primary/secondary confluences are met.**

**4. POSITION SIZE CALCULATION**

**Step 4.1: Determine Position Risk**

* position\_risk = account\_equity × risk%

**Step 4.2: Calculate SL/TP distances**

* **Stop Loss (SL):**
  + For long: SL = entry\_price - ATR × 1
  + For short: SL = entry\_price + ATR × 1
  + stop\_loss\_distance = abs(entry\_price - SL)
* **Take Profit (TP):**
  + TP = entry\_price + 2 × (entry\_price - SL) (long)
  + TP = entry\_price - 2 × (SL - entry\_price) (short)
  + RRR always 1:2

**Step 4.3: Account for Spread**

* effective\_stop\_loss\_distance = stop\_loss\_distance + spread

**Step 4.4: Compute Lot Size**

* lot\_size = position\_risk / (effective\_stop\_loss\_distance × pip\_value)
* *Lot size must be rounded to broker’s minimum allowed step.*

**5. TRADE ENTRY RULES**

* Only enter a trade if all 4 primary confluences are True.
* Use risk % as determined by risk tier and confluence scoring.
* Log all entry data (timestamp, price, confluence scores, risk %, lot size, SL, TP).

**6. BREAKEVEN LOGIC (POST-ENTRY MANAGEMENT)**

**Step 6.1: Monitor Trade**

* After entry, monitor price movement.

**Step 6.2: Calculate BE Trigger**

* BE\_trigger\_level = entry\_price + 0.7 × (TP - entry\_price) (long)
* BE\_trigger\_level = entry\_price - 0.7 × (entry\_price - TP) (short)

**Step 6.3: Adjust Stop to Breakeven**

* When price reaches BE\_trigger\_level, move SL to entry\_price.

**Step 6.4: Log Breakeven Action**

* Log time, price, and reason for SL adjustment.

**7. EXIT MANAGEMENT**

* Exit at SL or TP as hit by market.
* After BE move, if price returns to entry, exit trade for 0 loss.
* All exits (SL, TP, BE) must be logged.

**8. MONTHLY/MANDATORY REVIEWS**

* Review win rate, R-multiple, trade logs.
* Adjust risk tiers, confluence logic, or strategy as required.

**SUMMARY**

* **Inputs:** Equity, prices, ATR, spread, pip\_value, confluences
* **Decisions:** Only trade if all 4 primary; more secondaries = more risk allowed
* **Risk:** 0.5% base, dynamic by tier, confluence, drawdown
* **Stops:** ATR-based, 1:2 RRR, spread always included
* **BE:** Move SL to entry after 70% move to TP
* **Logging:** All entries, exits, BE moves, and risk reviews mandatory

**Below: A complete, modular Python system (pure Python, pandas, NumPy; ready for integration with your MT5 bridge, feature pipeline, or backtest engine).  
Covers: confluence scoring, position sizing, ATR-based SL/TP, breakeven logic, full risk tiering, and logs every key action.**

**1. Main Confluence & Risk Engine**

import numpy as np

import pandas as pd

from datetime import datetime

class XAUUSDConfluenceRiskEngine:

# Primary and secondary confluences

PRIMARY\_CONFS = ['conf\_structure', 'conf\_bos\_or\_choch', 'conf\_candle', 'conf\_sr\_zone']

SECONDARY\_CONFS = ['conf\_psych\_level', 'conf\_fib\_zone', 'conf\_volume', 'conf\_liquidity', 'conf\_spread']

def \_\_init\_\_(self, account\_equity, account\_status='baseline', drawdown\_pct=0.0, spread=0.5, pip\_value=100):

self.account\_equity = account\_equity

self.account\_status = account\_status # 'profit', 'baseline', or 'drawdown'

self.drawdown\_pct = drawdown\_pct # e.g., 0.20 for 20% drawdown

self.spread = spread # in $ for gold (e.g., 0.5)

self.pip\_value = pip\_value # $100 per $1 move per standard lot XAUUSD

def get\_risk\_pct(self, primary\_count, secondary\_count):

# Risk management tiering

if self.account\_status == 'profit':

risk\_pct = 0.01

elif self.account\_status == 'drawdown' and self.drawdown\_pct > 0.5:

risk\_pct = 0.002

else:

risk\_pct = 0.005

# Risk adjust for confluence: only allow max risk if all 4 primary

if primary\_count < 4:

return 0.0 # no trade

# If secondary confluences present, allow full risk; if none, stay at base risk

if secondary\_count >= 2:

return risk\_pct

return risk\_pct # still risk allowed if all primary, secondaries are bonus

def compute\_confluence(self, row):

primary\_score = sum(int(row.get(c, 0)) for c in self.PRIMARY\_CONFS)

secondary\_score = sum(int(row.get(c, 0)) for c in self.SECONDARY\_CONFS)

return primary\_score, secondary\_score

def position\_sizing(self, risk\_pct, entry\_price, sl\_price, direction='long'):

position\_risk = self.account\_equity \* risk\_pct

stop\_loss\_distance = abs(entry\_price - sl\_price)

effective\_stop = stop\_loss\_distance + self.spread

if effective\_stop == 0:

return 0.0

lot\_size = position\_risk / (effective\_stop \* self.pip\_value)

return np.round(lot\_size, 2) # rounded to 0.01 lots

def atr\_sl\_tp(self, entry\_price, atr, direction='long'):

# 1:2 RRR, multiplier = 1 for SL, TP is 2x SL distance

if direction == 'long':

sl = entry\_price - atr \* 1

tp = entry\_price + 2 \* (entry\_price - sl)

else:

sl = entry\_price + atr \* 1

tp = entry\_price - 2 \* (sl - entry\_price)

return round(sl, 2), round(tp, 2)

def breakeven\_level(self, entry\_price, tp, direction='long'):

# Move SL to BE after price has moved 70% toward TP

if direction == 'long':

be\_trigger = entry\_price + 0.7 \* (tp - entry\_price)

else:

be\_trigger = entry\_price - 0.7 \* (entry\_price - tp)

return round(be\_trigger, 2)

def generate\_trade\_signal(self, row, entry\_price, atr, direction='long'):

primary, secondary = self.compute\_confluence(row)

risk\_pct = self.get\_risk\_pct(primary, secondary)

if risk\_pct == 0.0:

return {'signal': False, 'reason': 'Not all primary confluences met'}

sl, tp = self.atr\_sl\_tp(entry\_price, atr, direction)

lot\_size = self.position\_sizing(risk\_pct, entry\_price, sl, direction)

be\_level = self.breakeven\_level(entry\_price, tp, direction)

log = {

'timestamp': datetime.now().isoformat(),

'primary\_confs': primary,

'secondary\_confs': secondary,

'risk\_pct': risk\_pct,

'lot\_size': lot\_size,

'entry\_price': entry\_price,

'stop\_loss': sl,

'take\_profit': tp,

'breakeven\_trigger': be\_level,

'signal': True,

'reason': 'All primary met, secondary: {}'.format(secondary)

}

return log

**2. Batch Processing Usage**

# Example DataFrame of signals for current bar (each row = a potential trade)

signals = pd.DataFrame([

{'conf\_structure':1, 'conf\_bos\_or\_choch':1, 'conf\_candle':1, 'conf\_sr\_zone':1, 'conf\_psych\_level':1, 'conf\_fib\_zone':1, 'conf\_volume':0, 'conf\_liquidity':0, 'conf\_spread':1},

{'conf\_structure':1, 'conf\_bos\_or\_choch':1, 'conf\_candle':1, 'conf\_sr\_zone':1, 'conf\_psych\_level':0, 'conf\_fib\_zone':0, 'conf\_volume':0, 'conf\_liquidity':0, 'conf\_spread':1},

# Add more rows as needed

])

engine = XAUUSDConfluenceRiskEngine(

account\_equity=10000,

account\_status='baseline', # or 'profit', or 'drawdown'

drawdown\_pct=0.0,

spread=0.5, # adjust for your broker

pip\_value=100 # for XAUUSD

)

results = []

for idx, row in signals.iterrows():

# You would supply the real entry price and ATR for each trade setup:

entry\_price = 1950.00

atr = 3.0

direction = 'long' # or 'short'

result = engine.generate\_trade\_signal(row, entry\_price, atr, direction)

results.append(result)

results\_df = pd.DataFrame(results)

print(results\_df)

**3. Trade Management (Breakeven and Logging Example)**

def manage\_trade(current\_price, trade\_log, direction='long'):

# Check if price has hit breakeven trigger

be\_trigger = trade\_log['breakeven\_trigger']

entry\_price = trade\_log['entry\_price']

sl = trade\_log['stop\_loss']

# Only adjust if SL is not already at entry

be\_moved = False

if direction == 'long' and current\_price >= be\_trigger and sl != entry\_price:

trade\_log['stop\_loss'] = entry\_price # Move SL to breakeven

be\_moved = True

elif direction == 'short' and current\_price <= be\_trigger and sl != entry\_price:

trade\_log['stop\_loss'] = entry\_price

be\_moved = True

if be\_moved:

print(f"Breakeven moved at price {current\_price}.")

return trade\_log

**You now have:**

* **Confluence logic (primary/secondary, strict logic)**
* **Dynamic risk tiering and position sizing (equity, risk, spread, pip\_value)**
* **ATR-based SL/TP, exact 1:2 RRR**
* **Breakeven trigger at 70% TP distance**
* **Full signal/execution log per trade**
* **Batch processing for research/backtest**