agents_design (Updated)

This document defines each agent with: (1) purpose, (2) inputs/outputs & topics, (3) algorithms/mathematics, and (4) how downstream components should interpret the output.

> **Key addition:** `PricePredictionAgent (LSTM)` and `ValueFundamentalsAgent`, plus an expanded `DebateOrchestratorAgent` protocol.

Agent Catalog (Updated)

Agent	Primary Outputs	Primary Consumers
IngestionAgent	`data.market.quote`, `data.portfolio.state`, raw text feeds	Most agents
RiskMetricsAgent	`data.risk.snapshot`	RiskEvent, UI, Explanation
RiskEventAgent	`data.risk.event`	MCDA, UI, Explanation, Debate gating
MCDAAgent	`data.risk.prioritized`	DebateOrchestrator, Explanation
PricePredictionAgent (LSTM)	`data.forecast.lstm`	RiskEvent, MCDA, Debate, Explanation
ValueFundamentalsAgent	`data.fundamentals.value`	RiskEvent, MCDA, Debate, Explanation
EarningsCallSentimentAgent	`data.earnings.sentiment`	RiskEvent, Explanation, Debate
RegimeShiftDetector	`data.market.regime`	MCDA, Debate, Explanation
ExplanationAgent (RAG)	`thought.risk.explanation`	UI, Debate
DebateOrchestratorAgent	`thought.debate.conclusion`, proposals, critiques	ExecutionPlanner, UI

Agent	Primary Outputs	Primary Consumers
ExecutionPlannerAgent	`data.execution.plan`	UI approval, ExecutionEngine
ExecutionEngineAgent	`data.execution.status`, orders	Monitor, UI
ExecutionMonitorAgent	`thought.execution.summary`, feedback events	Risk loop, UI

---

1) PricePredictionAgent (LSTM Forecast)

Purpose

Produce a **next-day predictive signal** per symbol (return or price), including confidence/uncertainty diagnostics. This is a forecast input to risk orchestration—not a direct trading command.

Inputs

- `data.market.quote`: prices, volume, returns
- `data.features.technical`: precomputed feature vector (recommended), or the agent computes features internally
- Model artifacts: LSTM weights, scaler/normalizer parameters, training metadata

Output topic + schema

`data.forecast.lstm`

```
{
  "schema_version":"1.0",
  "msg_id":"01J...",
  "produced_at":"2025-12-15T00:00:00Z",
  "symbol":"NVDA",
  "horizon":"1d",
  "target":"return",
  "yhat": 0.0123,
  "yhat_price": 152.40,
  "uncertainty": {
    "sigma": 0.018,
    "p10": -0.010,
    "p50": 0.012,
```

```
      "p90": 0.035,
      "method": "mc_dropout"
    },
    "feature_snapshot": {
      "return_1d": 0.004,
      "return_5d": 0.031,
      "return_20d": 0.082,
      "return_60d": 0.140,
      "return_120d": 0.220,
      "return_252d": 0.410,
      "volatility_5d": 0.55,
      "volatility_20d": 0.42,
      "volatility_60d": 0.38,
      "volume_ratio_5d": 1.4,
      "volume_ratio_20d": 1.1,
      "dollar_volume": 3.2e10,
      "market_beta": 1.6,
      "market_return": 0.002,
      "market_volatility": 0.18,
      "rsi_14": 62.0,
      "macd": 1.8,
      "sma_50_200_cross": 1,
      "bollinger_position": 0.72,
      "atr": 4.1
    },
    "diagnostics": {
      "data_freshness_sec": 45,
      "lookahead_checks_passed": true,
      "model_version": "lstm_v2.1",
      "train_window":"2016-01-01..2025-11-30",
      "last_retrain":"2025-12-01"
    }
  }
```


Algorithm / mathematics

Feature computation (typical definitions)

- Returns: `return_kd = (P_t / P_{t-k}) - 1`
- Volatility (rolling): `vol_kd = std(r_{t-k+1..t}) * sqrt(annualization_factor)`
- Volume ratio: `volume_ratio_kd = vol_t / mean(vol_{t-k+1..t})`
- RSI (14): standard Wilder RSI on returns

- MACD: EMA(12) - EMA(26) (optionally signal/ histogram)
- SMA cross: indicator `1` if SMA50 > SMA200 else `0`
- Bollinger position: normalized position of price within bands (0..1)
- ATR: average true range (volatility-of-range measure)

LSTM inference

The LSTM maps a sequence of feature vectors `X_{t-L+1..t}` to a next-day prediction:

```
ŷ_{t+1} = f_LSTM(X_{t-L+1}, ..., X_t)
```

Uncertainty (recommended): provide confidence bounds rather than a single point estimate. Options:

- **MC Dropout:** run N stochastic forward passes with dropout enabled; use mean and std:

  ```
  ŷ = mean(ŷ_i),  sigma = std(ŷ_i),  i=1..N
  ```

- **Quantile regression head:** predict p10/p50/p90 directly.
- **Ensemble:** multiple LSTMs with different seeds, average predictions, compute dispersion.

How to interpret and use the output downstream

- **Do not treat as a trade signal by itself.** Treat as an evidence input with uncertainty.
- **RiskEventAgent:** can fire an event if forecast tail (p10) implies large downside beyond threshold (e.g., “Forecasted -3% with high confidence”).
- **MCDAAgent:** incorporate forecast into “immediacy” / “severity” terms, but penalize low confidence.
- **DebateOrchestrator:** use forecast to trigger “hedge vs hold” debate when forecast conflicts with value fundamentals or regime.
- **UI:** display yhat + uncertainty band + diagnostics; highlight when confidence low or data stale.

---

2) ValueFundamentalsAgent (Value Investing Perspective)

Purpose

Compute valuation and “business quality” signals that support **long-horizon decisions**: intrinsic value range, margin-of-safety, and quality/moat proxies from EDGAR fundamentals.

Inputs

- SEC EDGAR filings (10-K/10-Q): revenue, operating income, FCF, shares, debt, etc.
- Optional: AlphaVantage fundamentals endpoints (if used as a convenience layer)
- Market price (`data.market.quote`) to compute margin-of-safety vs current price

Output topic + schema

`data.fundamentals.value`


```
{
  "schema_version":"1.0",
  "msg_id":"01J...",
  "produced_at":"2025-12-15T00:05:00Z",
  "symbol":"NVDA",
  "valuation": {
    "method":"dcf_multi_stage",
    "intrinsic_value": {"p10": 120.0, "p50": 155.0, "p90": 210.0},
    "assumptions": {
      "fcf_growth_5y": {"p10": 0.08, "p50": 0.18, "p90": 0.30},
      "discount_rate": {"p50": 0.11},
      "terminal_growth": {"p50": 0.03}
    }
  },
  "margin_of_safety": {
    "price": 152.0,
    "mos_p50": 0.02,
    "mos_p10": -0.21,
    "classification":"FAIRLY_VALUED"
  },
  "quality": {
    "roic": 0.22,
    "gross_margin": 0.68,
    "operating_margin": 0.32,
    "fcf_margin": 0.25,
    "debt_to_fcf": 1.2,
    "interest_coverage": 18.0
  },
  "red_flags": ["NONE"],
  "diagnostics": {
    "filing_used":"10-K",
    "filing_date":"2025-02-21",
    "model_version":"value_v1.0"
```

```
    }
  }
```

Algorithm / mathematics

Core ratios

- ROIC (proxy): `ROIC = NOPAT / InvestedCapital`
- FCF margin: `FCF / Revenue`
- Debt-to-FCF: `TotalDebt / FCF`
- Interest coverage: `EBIT / InterestExpense`

DCF (multi-stage) valuation

```
PV = Σ_{t=1..T} (FCF_t / (1+r)^t) + (TV / (1+r)^T)
TV = FCF_{T+1} / (r - g)
```

Use scenario/quantile assumptions for growth and discount rate to create an intrinsic value distribution (p10/p50/p90).

Margin-of-safety

```
MoS = (IntrinsicValue - Price) / Price
```

Classification example: UNDERVALUED if MoS_p50 > 0.20; OVERVALUED if MoS_p50 < -0.20; else FAIR.

How to interpret and use the output downstream

- **Debate is the primary consumer:** value signals define “thesis alignment” and provide a counterweight to short-term forecast/volatility.
- **MCDA:** value can reduce urgency of selling (if strongly undervalued) but cannot override hard risk constraints (margin, VaR breaches).
- **UI:** show intrinsic value band + MoS + quality metrics; explicitly label assumptions and filing date.

---

3) DebateOrchestratorAgent (Expanded Flow + Coordination + Conclusion)

Purpose

Coordinate a structured multi-agent debate to reconcile conflicting signals and produce an auditable, constrained recommendation.

Inputs

- `data.risk.prioritized` (ranked events + regime)
- `data.risk.snapshot` (current exposure, VaR, drawdown, margin)
- `data.forecast.lstm` (forecast + uncertainty)
- `data.fundamentals.value` (intrinsic value + MoS + quality)
- `thought.risk.explanation` (RAG explanation + uncertainties)

Outputs

- `thought.debate.context` (frozen packet)
- `thought.action.proposal` (per strategy agent)
- `thought.action.critique` (cross-exam)
- `thought.debate.conclusion` (final recommendation + runner-up + reasons + confidence)

Protocol (how it coordinates)

Step 0: Build frozen context packet


```
{
  "debate_id":"deb_01J...",
  "account_id":"acct_123",
  "top_events":[...],
  "risk_snapshot_id":"01J...",
  "forecast_summary":{"symbol":"NVDA","p10":-0.01,"p50":0.012,"p90":0.035,"sig
  "value_summary":{"symbol":"NVDA","mos_p50":0.02,"intrinsic_p50":155},
  "hard_constraints":{"max_var_pct":0.04,"max_concentration_pct":0.25},
  "uncertainties":[...]
}
```


Step 1: Proposal round (parallel)

Each strategy agent must output a *schema-valid* proposal:


```
{
  "debate_id":"deb_01J...",
  "agent":"HedgeAgent",
  "proposal":{
```

```
    "actions":[{"type":"HEDGE","instrument":"QQQ_PUT_SPREAD","notional":250000
    "expected_impact":{"var_reduction_pct_points":0.6,"net_exposure_reduction"
    "estimated_cost":{"premium_usd":4200,"fees_usd":35},
    "failure_modes":["vol crush","gap risk"],
    "assumptions":["liquidity ok","spreads normal"]
  }
}
```


Step 2: Cross-exam (structured critiques)


```
{
  "debate_id":"deb_01J...",
  "agent":"TrimAgent",
  "critiques":[
    {"target_agent":"HedgeAgent","issue":"Cost too high for small VaR gain","e
    {"target_agent":"HoldAgent","issue":"Violates concentration constraint unc
  ]
}
```


Step 3: RiskOfficer validation (hard constraints + feasibility)

RiskOfficer returns PASS/REVISE/REJECT with reasons. REVISE loops back to Step 1 for the affected proposals only.

Step 4: Deterministic conclusion synthesis

Orchestrator ranks proposals with a fixed scoring model (weights versioned):

```
score = 0.35*risk_reduction + 0.20*feasibility + 0.15*cost_efficiency
      + 0.15*robustness + 0.15*thesis_alignment - 0.10*tail_risk
```

Conclusion payload:


```
{
  "debate_id":"deb_01J...",
  "recommendation":{
    "winner":"HedgeAgent",
    "runner_up":"TrimAgent",
    "why":["Var breach resolved with minimal thesis damage","Feasible within ]
```

```
    "confidence":0.74
  },
  "required_user_action":"APPROVE_PLAN",
  "next_steps":["generate execution plan","simulate slippage","await approval'
}
```

> **Key governance rule:** Debate output cannot bypass “hard constraints” checks; RiskOfficer is the gatekeeper. The system can be advisory-only until you explicitly enable auto-execution.

---

4) Minimal updates needed in existing agents (to consume forecast + value)

- **RiskEventAgent:** add rule types:
  - `FORECAST_DOWNSIDE_TAIL`: if p10 < -X% and uncertainty small
  - `VALUE_DISLOCATION`: if MoS_p50 > +Y% (undervalued) or < -Y% (overvalued)
  - `SIGNAL_CONFLICT`: if LSTM bullish but regime bearish / risk breaches high
- **MCDAAgent:** incorporate forecast and value in criteria:
  - Severity/Immediacy boosted by forecast downside tail
  - Thesis alignment boosted by undervaluation + quality
  - Penalty for low confidence forecast (high sigma)
- **ExplanationAgent:** explicitly cite:
  - forecast band (p10/p50/p90) and uncertainty method
  - intrinsic value band and key assumptions + filing date
  - why debate was triggered (conflicts, breaches)

> **Critical correctness note:** Ensure your LSTM feature pipeline is strictly “t and earlier” (no lookahead leakage), and log `lookahead_checks_passed` in every forecast.