# RLC Hybrid Intelligence Architecture

## From Automated Scripts to Intelligent Agents: A Migration Plan

**Version**: 0.1 (Draft)
**Date**: February 4, 2026
**Author**: Architecture planning session with Claude

---

## Executive Summary

This document outlines a phased approach to evolving RLC's analytical infrastructure from deterministic automation to a hybrid system that combines:

- **Statistical rigor** (econometric models, ML forecasting)
- **AI reasoning** (LLMs for synthesis, judgment, and content)
- **Agentic coordination** (goal-oriented agents working together)

The architecture preserves existing investments, adds capabilities incrementally, and includes clear decision points for hardware purchases.

---

## Part 1: Current State Assessment

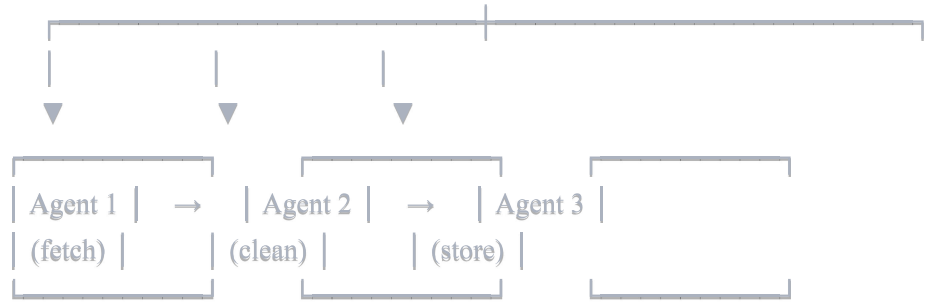
### Existing Infrastructure

| Component | Status | Notes |
|---|---|---|
| MSI Aegis ZS2 (RTX 5080, 16GB VRAM) | ✅ Operational | Primary compute, runs 24/7 |
| Lenovo Yoga 9 | ✅ Operational | Secondary/mobile |
| PostgreSQL Database | ✅ Operational | Bronze/Silver/Gold layers |
| 46 Agent Scripts | ✅ Operational | Mostly deterministic |
| Power BI Dashboards | ✅ Operational | Visualization layer |
| Notion "RLC OS" | ✅ Operational | Knowledge management |

**Current Agent Inventory (Estimated Categories)**

```
Data Collection Agents (~15 agents)
├── USDA PSD API
├── USDA NASS
├── USDA AMS
├── NOAA Weather
├── Census Trade Data
├── FGIS Export Inspections
└── Various market feeds

Data Processing Agents (~12 agents)
├── Bronze → Silver transformations
├── Data validation
├── Schema normalization
├── Marketing year alignment
└── Time series alignment

Analytics Agents (~8 agents)
├── Basis calculations
├── Spread monitoring
├── Seasonal patterns
└── Trend detection

Output Agents (~6 agents)
├── Report generation
├── Power BI refresh
├── Alert dispatch
└── Content formatting

Utility Agents (~5 agents)
├── Error handling
├── Logging
├── Scheduling
└── Health checks
```

**Current Architecture Pattern**




Each agent: Single task, explicit rules, no reasoning

**Strengths of Current System:**

- Reliable and predictable
- Fast execution
- Low resource usage
- You understand every line of code

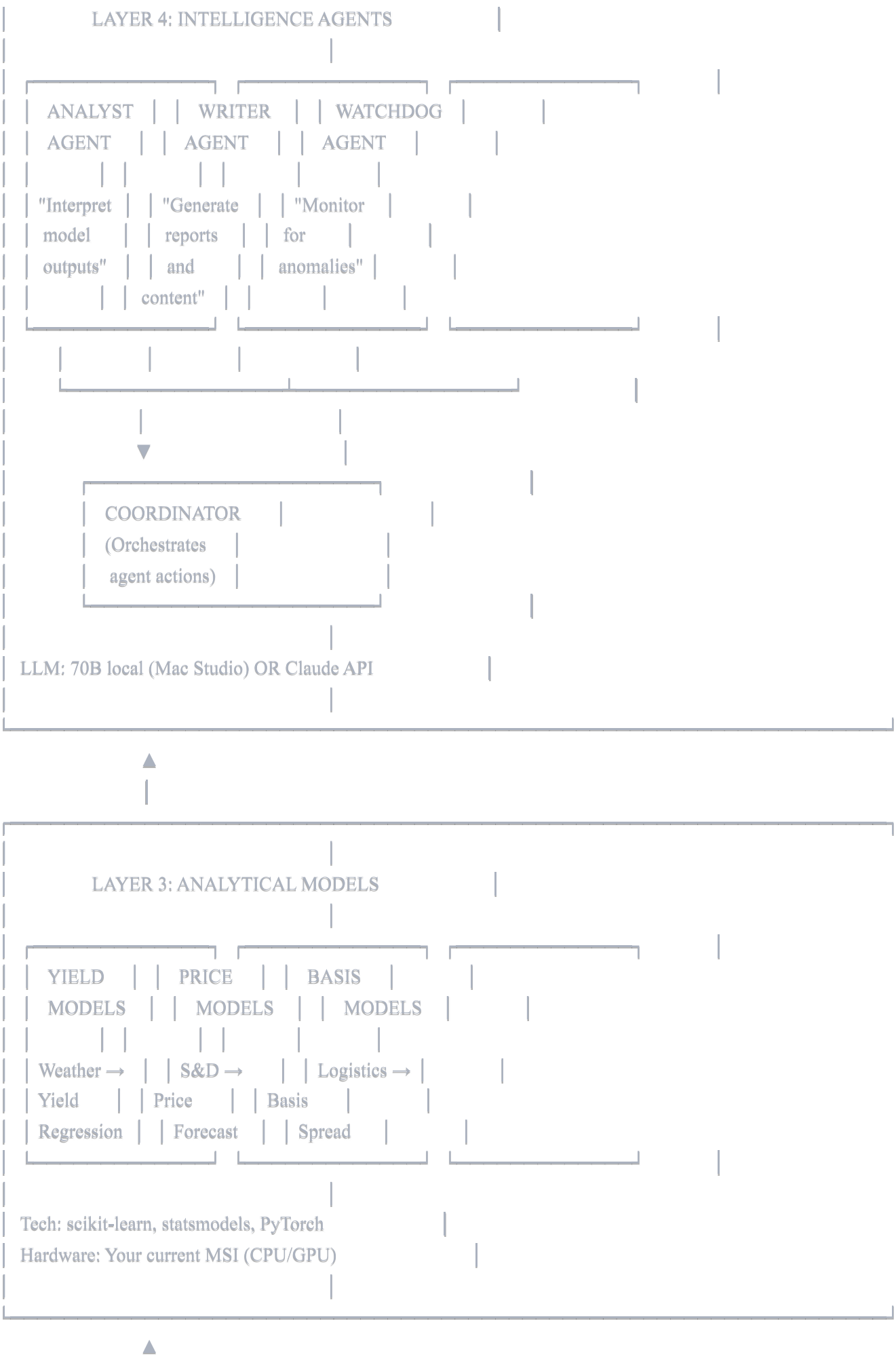
**Limitations:**

- Cannot handle unexpected situations
- Cannot synthesize across data sources
- Cannot explain "why" something is happening
- Cannot adapt without code changes
- Cannot generate nuanced analysis

---

## Part 2: Target Architecture

### The Layered Hybrid Model


LAYER 4: INTELLIGENCE AGENTS
ANALYST AGENT
"Interpret model outputs"
WRITER AGENT
"Generate reports and content"
WATCHDOG AGENT
"Monitor for anomalies"
COORDINATOR
(Orchestrates agent actions)
LLM: 70B local (Mac Studio) OR Claude API
LAYER 3: ANALYTICAL MODELS
YIELD MODELS
Weather → Yield Regression
PRICE MODELS
S&D → Price Forecast
BASIS MODELS
Logistics → Basis Spread
Tech: scikit-learn, statsmodels, PyTorch
Hardware: Your current MSI (CPU/GPU)


LAYER 2: DATA PROCESSING
SILVER LAYER
Cleaned, normalized, aligned, validated data
Transform Agent (rules)
Validate Agent (rules + LLM edge)
Align Agent (rules)
Mostly deterministic, LLM only for edge cases
LAYER 1: DATA COLLECTION
BRONZE LAYER
Raw data exactly as received from sources
USDA Agent
NOAA Agent
Census Agent
FGIS Agent
News Agent
100% deterministic - no LLM needed

## Intelligence Agent Specifications

### ANALYST AGENT

```yaml
name: Analyst Agent
goal: Interpret quantitative model outputs and market data to form market views

tools:
  - query_database: Access Gold layer data
  - get_model_output: Retrieve latest forecasts
  - get_historical_context: Pull comparison periods
  - get_news_summary: Recent relevant news
  - update_market_view: Store current thesis

triggers:
  - New model output available
  - Significant data revision
  - Price moves beyond threshold
  - Scheduled daily review

example_reasoning: |
  "The yield model updated with new NOAA data showing 2" below-normal
  precip for western Iowa through mid-August. Model now projects
  state yield at 195 bu/acre vs 198 last week. This is a 1.5%
  reduction. Combined with similar revisions in Illinois, national
  production estimate drops ~150M bushels. Current futures appear
  to have priced in ~100M of this. Recommend: monitor for further
  weather deterioration; if precip deficit persists, market has
  room to rally $0.15-0.20."

llm_requirement: 70B+ for nuanced reasoning, or Claude API
```

### WRITER AGENT

```yaml
```

```yaml
name: Writer Agent
goal: Generate market reports and content in RLC's voice and style

tools:
  - get_market_view: Current analyst thesis
  - get_data_tables: Formatted S&D tables
  - get_template: Report templates
  - get_style_guide: RLC writing standards
  - publish_report: Output to destination

triggers:
  - Scheduled report times (daily, weekly)
  - Analyst agent flags significant development
  - Manual request

output_types:
  - Daily market brief (300-500 words)
  - Weekly S&D summary (1000-1500 words)
  - Flash alerts (50-100 words)
  - Social media posts (LinkedIn, Twitter)
  - Client emails

llm_requirement: 7B adequate for templated content, 70B for original analysis
```

**WATCHDOG AGENT**

yaml

```
name: Watchdog Agent
goal: Continuously monitor for anomalies, errors, and significant changes

tools:
  - query_database: Check latest data
  - compare_historical: Statistical anomaly detection
  - check_data_quality: Validation rules
  - send_alert: Notify on findings
  - log_observation: Record for review

monitors:
  - Data quality issues (missing, invalid, outliers)
  - Unexpected price moves (> 2 std dev)
  - Unusual basis behavior
  - Model forecast changes (> threshold)
  - News keyword triggers
  - Source availability (API health)

alert_levels:
  - INFO: Log only, include in daily digest
  - WATCH: Include in next analyst review
  - ALERT: Immediate notification
  - CRITICAL: Interrupt current work

llm_requirement: 7B adequate for pattern matching, 70B for interpretation
```

---

## Part 3: Migration Phases

### Phase 0: Foundation (Current → Week 4)

**Goal**: Organize existing infrastructure, no new AI yet

**Tasks**:

1. Document all 46 current agents (inputs, outputs, dependencies)
2. Ensure Bronze/Silver/Gold data layers are clearly separated
3. Create comprehensive logging for all agent activities
4. Set up monitoring dashboard for system health
5. Identify which agents are truly needed vs. legacy

**Hardware**: Current MSI only
**LLM**: None (or existing small model usage)
**Cost**: $0 (time investment only)

**Success Criteria**:

- Complete agent inventory document
- All agents logging to central location
- System health dashboard operational
- Clear data lineage documentation

---

**Phase 1: Analytical Models (Weeks 4-12)**

**Goal**: Build the quantitative foundation that AI will interpret

**Tasks**:

1. Implement yield forecasting models
    - Weather → Yield regression (corn, soybeans by state)
    - Input: NOAA data, historical yields
    - Output: Point forecast + confidence interval
2. Implement price forecasting models
    - S&D → Price relationships
    - Seasonal patterns
    - Basis models by location
3. Create model output tables in Gold layer
    - Standardized format for all model outputs
    - Version tracking (what changed, when)
    - Forecast vs. actual tracking
4. Build model performance monitoring
    - Track forecast accuracy over time
    - Identify when models need retraining

**Hardware**: Current MSI (CPU for training, GPU optional)
**LLM**: None yet
**Cost**: $0 (time investment only)

**Success Criteria**:

- At least 3 working forecast models
- Model outputs flowing to Gold layer
- Accuracy tracking operational
- Documentation of model methodology

**Key Deliverable**:

> When the ANALYST AGENT later asks "what does the model say about Illinois soybean yields?", there's a real model with a real answer—not an LLM hallucination.

---

**Phase 2: LLM Integration - Read Only (Weeks 12-16)**

**Goal**: Add LLM reasoning that observes but doesn't act

**Tasks**:

1. Set up LLM inference
    - Option A: Claude API (recommended for this phase)
    - Option B: Local 7B/13B on RTX 5080 via Ollama
2. Create WATCHDOG AGENT (read-only)
    - Monitors data quality
    - Flags anomalies
    - Generates daily digest
    - NO automated actions, just observations
3. Create prototype ANALYST AGENT (advisory only)
    - Given: Latest model outputs + recent data
    - Produces: Draft market interpretation
    - Human reviews before any use
4. Evaluation period
    - Track: How often is the LLM's judgment correct?
    - Track: How often does it catch things you missed?
    - Track: How often does it hallucinate or miss obvious issues?

**Hardware**: Current MSI + API access
**LLM**: Claude API (~$50-150/month estimated)
**Cost**: ~$200-600 total for phase

**Success Criteria**:

- WATCHDOG running daily, generating useful observations
- ANALYST producing draft interpretations you find valuable
- Quantified accuracy: "LLM was helpful X% of the time"
- Clear understanding of LLM limitations in your domain

**Decision Point**:

```
At the end of Phase 2, you'll know:
- Is the LLM reasoning valuable enough to continue?
- Is 7B/13B local sufficient, or do you need 70B?
- Are API costs acceptable, or should you go local?

If LLM adds little value → Stay with Phase 1 architecture
If LLM is valuable + API costs OK → Continue with API
If LLM is valuable + want local → Consider hardware purchase
```

---

**Phase 3: LLM Integration - Write Access (Weeks 16-24)**

**Goal**: Allow AI agents to take limited automated actions

**Tasks**:

1. WRITER AGENT implementation
    - Generates draft reports from templates + model outputs
    - Human approval required before publishing
    - Gradually reduce approval requirements for routine content
2. WATCHDOG AGENT upgrades
    - Can auto-log issues to tracking system
    - Can trigger data re-fetch on quality failures
    - Still requires human approval for alerts to you
3. Implement agent coordination
    - WATCHDOG detects issue → triggers ANALYST review

- ANALYST forms view → triggers WRITER for report
- Simple workflow, human checkpoints

4. Build feedback loop
    - When you correct agent output, log the correction
    - Use corrections to improve prompts/fine-tuning

**Hardware**: Current MSI + API (or local if decided in Phase 2)
**LLM**: Same as Phase 2 decision
**Cost**: Depends on Phase 2 decision

**Success Criteria**:

- WRITER producing usable first drafts 70%+ of the time
- WATCHDOG catching issues before you notice them
- Clear escalation paths working
- Reduced time spent on routine analysis

---

**Phase 4: Autonomous Operations (Weeks 24+)**

**Goal**: Agents operate with minimal supervision for routine tasks

**Tasks**:

1. Define "routine" vs "exceptional" clearly
    - Routine: Daily data refresh, standard reports, normal market conditions
    - Exceptional: Data anomalies, unusual market moves, breaking news
2. Implement tiered autonomy
    - TIER 1 (Full auto): Data collection, validation, standard reports
    - TIER 2 (Log + proceed): Minor anomalies, routine analysis updates
    - TIER 3 (Ask first): Significant findings, non-routine content
    - TIER 4 (Human only): Trading signals, client communications, model changes
3. Implement COORDINATOR agent
    - Orchestrates other agents
    - Decides which agent to invoke based on situation
    - Manages priorities and resources
4. Continuous improvement

- Agents suggest improvements to their own prompts
- You approve/reject, system learns

**Hardware**: Possibly Mac Studio if local 70B needed
**LLM**: 70B local or continued API
**Cost**: $0-6,000 depending on hardware decision

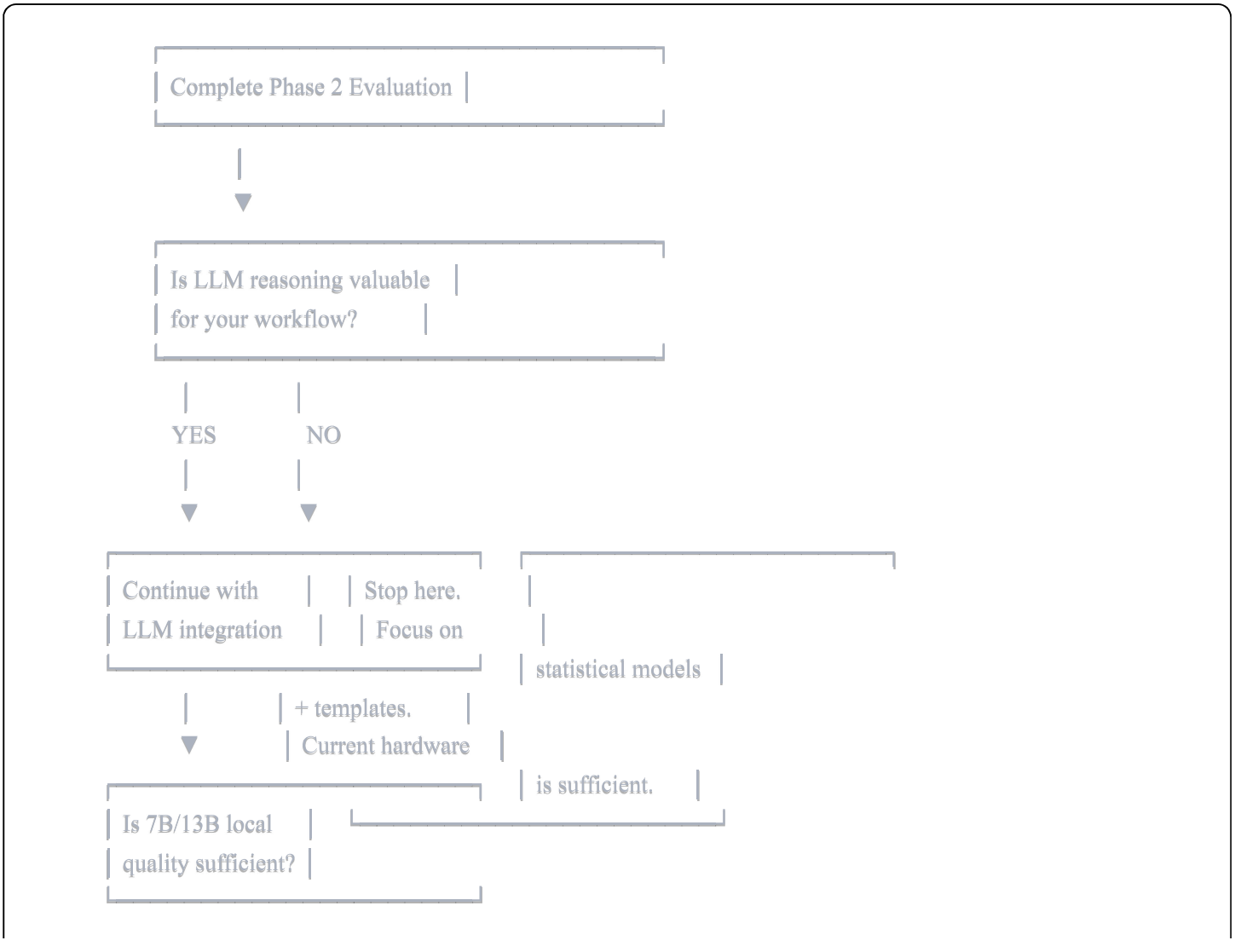
**Success Criteria**:

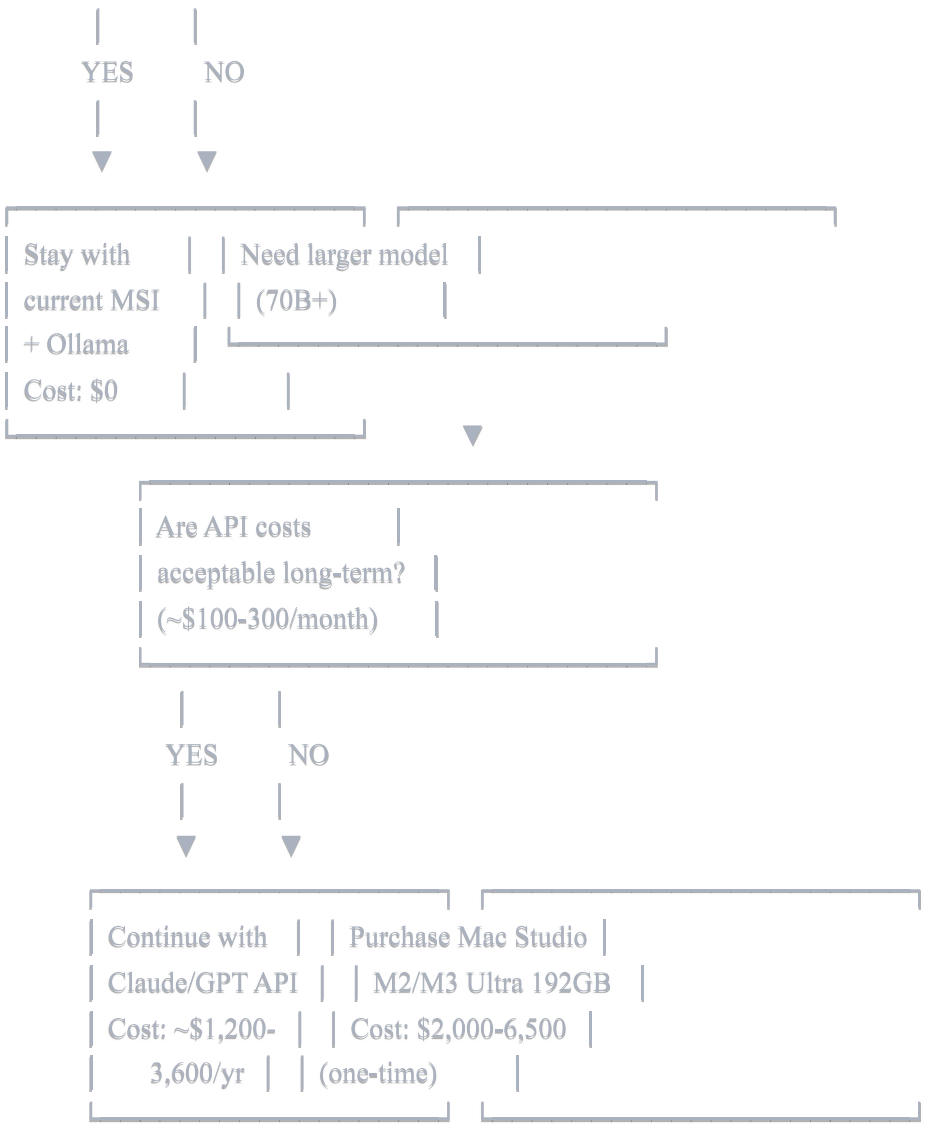
- System runs for days without intervention (for routine operations)
- You're notified only of things that need your attention
- Report quality indistinguishable from your manual work
- Clear audit trail of all agent decisions

---

## Part 4: Hardware Decision Framework

### Decision Tree






### Hardware Options Summary

| Option | Upfront Cost | Monthly Cost | Model Size | Best For |
|---|---|---|---|---|
| Current MSI + 7B Ollama | $0 | $0 | 7B-13B | Simple tasks, testing |
| Current MSI + Claude API | $0 | $100-300 | Frontier | Best quality, low volume |
| Mac Studio M2 Ultra 192GB (used) | $2,000-2,500 | $0 | 70B | Good quality, high volume, budget |
| Mac Studio M3 Ultra 192GB (new) | $6,000-6,500 | $0 | 70B | Good quality, high volume, warranty |
| Mac Studio M3 Ultra 512GB (new) | $10,000-14,000 | $0 | 180B+ | Maximum future-proofing |

## Part 5: Specific Agent Designs

### Example: Southern Illinois Soybean Yield Agent

This addresses your specific example: "forecast the yield of soybeans in southern Illinois based on weather/yield models and write an analysis"

yaml

```
name: IL_Soybean_Yield_Agent
type: Composite (Statistical Model + LLM Interpretation)

components:

  1_data_collection:
    type: Deterministic Script
    schedule: Daily at 6am CT
    actions:
      - Fetch NOAA weather data for IL climate divisions 6, 7, 8, 9
      - Fetch soil moisture from NLDAS
      - Fetch crop condition from NASS (when available)
      - Store in Bronze layer

  2_model_execution:
    type: Statistical Model (Python)
    schedule: After data collection completes
    model_type: Ridge regression with weather features
    features:
      - Cumulative precipitation (Apr-Aug)
      - GDD accumulation
      - Days above 95°F in July-Aug
      - Soil moisture at key growth stages
      - Trend yield adjustment
    output:
      - Point estimate (bu/acre)
      - 80% confidence interval
      - Comparison to 5-year average
      - Comparison to last week's estimate
    store: Gold layer (model_outputs.il_soybean_yield)

  3_interpretation:
    type: LLM Agent
    trigger: Model output changes by > 1 bu/acre OR weekly schedule
    prompt: |
      You are an agricultural analyst specializing in Midwest soybean production.

      Current model output for Southern Illinois soybeans:
      - Yield estimate: {yield_estimate} bu/acre
      - Last week: {last_week_estimate} bu/acre
      - Change: {change} bu/acre ({change_pct}%)
      - 5-year average: {avg_5yr} bu/acre
      - Confidence interval: {ci_low} to {ci_high}
```

```
    Weather factors driving change:
    {weather_summary}

    Recent observations:
    {crop_condition_notes}

    Tasks:
    1. Explain what's driving the yield estimate change in plain language
    2. Assess whether the model's reaction seems appropriate given conditions
    3. Note any factors the model might be missing
    4. Provide a 2-3 sentence summary suitable for a client report

  output: Structured interpretation stored to Gold layer

4_content_generation:
  type: LLM Agent (WRITER)
  trigger: Interpretation complete + scheduled report time
  inputs:
    - Interpretation from step 3
    - Template: weekly_crop_update.md
    - Style guide: rlc_writing_standards.md
    - Historical examples: sample_crop_reports/
  output:
    - Draft report section (200-400 words)
    - Suggested charts/tables
    - Confidence flag (high/medium/low)
  review: Human approval required before publishing
```

### Example: Data Quality Watchdog

yaml

```yaml
name: Data_Quality_Watchdog
type: Hybrid (Rules + LLM for ambiguous cases)

monitors:

  price_data:
    rules:
      - Price must be positive
      - Daily change < 15% (configurable)
      - No gaps > 2 trading days
      - Timestamp must be valid
    llm_escalation:
      trigger: Rule violation detected
      prompt: |
        A price data anomaly was detected:

        Commodity: {commodity}
        Date: {date}
        Reported price: {price}
        Previous price: {prev_price}
        Change: {change_pct}%

        Historical context:
        - 30-day average: {avg_30d}
        - 30-day std dev: {std_30d}
        - Recent news: {news_summary}

        Is this likely:
        A) A data error that should be flagged for manual review
        B) A real market move that should be accepted
        C) Unclear - need more information

        Explain your reasoning.

  fundamental_data:
    rules:
      - Production values within historical range (±50%)
      - Stocks cannot be negative
      - Exports + domestic use ≤ total supply
      - Marketing year dates must align
    llm_escalation:
      trigger: Rule violation OR data revision > 5%
      prompt: |
```

```
    A fundamental data change was detected:

    Report: {report_name}
    Field: {field_name}
    Previous value: {old_value}
    New value: {new_value}
    Change: {change_pct}%

    This change is {larger/smaller} than typical revisions.

    Historical revision patterns for this field:
    {revision_history}

    Is this revision:
    A) Within normal bounds - accept automatically
    B) Unusual but plausible - log for review
    C) Likely an error - flag for immediate attention

    Explain your reasoning.

alert_routing:
  A_accept: Log only, continue processing
  B_review: Add to daily digest, continue processing
  C_flag: Send immediate alert, pause downstream processing
```

---

## Part 6: Risk Assessment

### Things That Could Go Wrong

| Risk | Likelihood | Impact | Mitigation |
|---|---|---|---|
| LLM hallucinates market data | Medium | High | Always ground in database queries, never trust LLM for numbers |
| LLM gives confidently wrong analysis | Medium | Medium | Human review for all external-facing content |
| Model drift (forecasts degrade) | High | Medium | Continuous accuracy tracking, scheduled retraining |
| Over-reliance on AI judgment | Medium | High | Maintain your own market view, use AI as input not oracle |

| Risk | Likelihood | Impact | Mitigation |
|---|---|---|---|
| API costs exceed budget | Low | Low | Set spending caps, monitor usage |
| Hardware failure during critical period | Low | High | Redundancy plan, cloud backup capability |
| Data source changes format/breaks | High | Medium | Robust error handling, multiple source validation |

**The Honest Limitations**

**What this architecture CAN do:**

- Automate routine data processing and validation
- Generate first drafts of reports and analysis
- Catch anomalies and flag them for your attention
- Apply your analytical frameworks consistently
- Work 24/7 without fatigue

**What this architecture CANNOT do:**

- Predict prices better than your statistical models
- Replace your market intuition and experience
- Handle truly novel situations without guidance
- Make trading decisions you'd trust without review
- Understand context it hasn't been taught

**The 1+1=3 you're looking for:**

- Your expertise + AI synthesis = faster, more comprehensive analysis
- Statistical models + LLM interpretation = numbers with narrative
- Continuous monitoring + intelligent filtering = you see only what matters
- Templated output + LLM flexibility = consistent quality, less drudgery

---

## Part 7: Success Metrics

**Phase 1 Success (Statistical Models)**

- [ ] 3+ forecast models operational

- [ ] Model accuracy tracked and documented
- [ ] Gold layer contains all model outputs

**Phase 2 Success (LLM Read-Only)**

- [ ] WATCHDOG catching issues before manual discovery (50%+ of issues)
- [ ] ANALYST interpretations rated "useful" 70%+ of time
- [ ] Clear quantification of LLM value-add

**Phase 3 Success (LLM Write Access)**

- [ ] WRITER drafts accepted with minor edits 60%+ of time
- [ ] Time spent on routine reports reduced by 50%+
- [ ] No significant errors in published content

**Phase 4 Success (Autonomous Operations)**

- [ ] System runs 7+ days without intervention (routine operations)
- [ ] Alert signal-to-noise ratio > 3:1 (useful alerts vs. false alarms)
- [ ] Your weekly time on routine analysis reduced by 70%+

---

## Part 8: Next Steps

**Immediate Actions (This Week)**

- [ ] Review and refine this architecture document
- [ ] Complete inventory of current 46 agents
- [ ] Identify which current agents map to which layer
- [ ] Set up centralized logging if not already done

**Phase 0 Deliverables (Next 4 Weeks)**

- [ ] Complete agent documentation
- [ ] System health dashboard
- [ ] Data lineage documentation
- [ ] Decision: Which statistical models to build first?

**Key Decision Points**

- **Week 4**: Ready to start Phase 1? (statistical models)
- **Week 12**: Ready to start Phase 2? (LLM integration)
- **Week 16**: LLM value assessment → hardware decision

- **Week 24**: Autonomous operations readiness assessment

---

## Appendix A: Technology Stack

### Recommended Stack

| Layer | Technology | Notes |
|---|---|---|
| Data Collection | Python + requests/aiohttp | Keep existing |
| Data Storage | PostgreSQL | Keep existing |
| Data Processing | Python + pandas | Keep existing |
| Statistical Models | scikit-learn, statsmodels | Add |
| LLM Inference (local) | Ollama + llama.cpp | Add |
| LLM Inference (API) | Claude API / OpenAI API | Add |
| Agent Framework | LangChain or custom | Evaluate |
| Orchestration | Prefect or existing scheduler | Evaluate |
| Monitoring | Grafana or custom dashboard | Add |
| Version Control | Git | Keep existing |

### File Structure (Proposed)

```
rlc-agent/
├── agents/
│   ├── collection/      # Layer 1: Data fetching
│   ├── processing/      # Layer 2: Transformation
│   ├── models/          # Layer 3: Statistical models
│   └── intelligence/    # Layer 4: LLM agents
├── data/
│   ├── bronze/          # Raw data
│   ├── silver/          # Cleaned data
│   └── gold/            # Analytical outputs
├── config/
│   ├── agents.yaml      # Agent configurations
│   ├── models.yaml      # Model parameters
```

```
│   └── prompts/        # LLM prompt templates
├── templates/
│   └── reports/        # Report templates
├── monitoring/
│   └── dashboards/     # Monitoring configs
└── docs/
    └── architecture.md  # This document
```

---

## Appendix B: Glossary

- **Agent**: A software component that performs tasks, ranging from simple scripts to AI-powered decision makers
- **Bronze Layer**: Raw data stored exactly as received from sources
- **Silver Layer**: Cleaned, validated, normalized data
- **Gold Layer**: Analytical outputs, model results, aggregations
- **LLM**: Large Language Model (e.g., Claude, GPT-4, Llama)
- **Frontier Model**: State-of-the-art LLM (Claude, GPT-4) with best reasoning capability
- **Local Model**: LLM running on your own hardware
- **Agentic**: Software that can decide what actions to take based on goals, not just explicit instructions

---

*Document Status: Draft for Review Last Updated: February 4, 2026*