

EXECUTIVE EDUCATION SESSION

Beyond the Prototype

Architecting Self-Improving AI Systems for Production

A Case Study in Autonomous AI Management



Mario Facussé | Austin AI Alliance Summer Intensive | June 22-24, 2026

Case study provided by

MEF SOLUTIONS

The Problem We're All Facing

87%

**of AI projects never
make it past the pilot stage.**

- AI demos work. Production breaks.
- The gap between 'works in a demo' and 'runs reliably at scale' is where organizations get stuck.

Source: McKinsey 2024 AI Adoption Report



By the end of this session, you will have a framework for closing this gap.

Four capabilities every production AI system needs

1

Shared Memory

How agents learn from each other

2

Trust Scoring

How to know which outputs to trust

3

Prompt Evolution

How systems improve themselves weekly

4

Safe Governance

How to deploy changes without breaking production

» These are principles, not products. You can implement them on any stack.

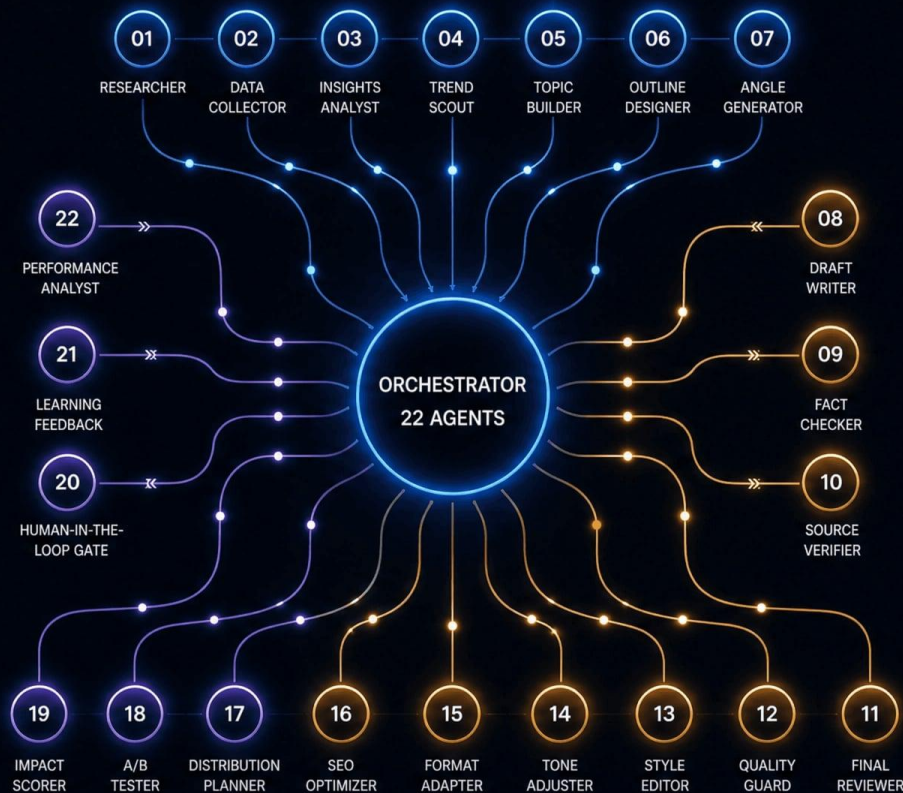
THE CASE STUDY

Real architecture, real data, real lessons.

- A production system running daily—not a prototype
- Orchestrated pipeline of 22 specialized AI agents
- Multi-model consensus & 3-pass automated verification



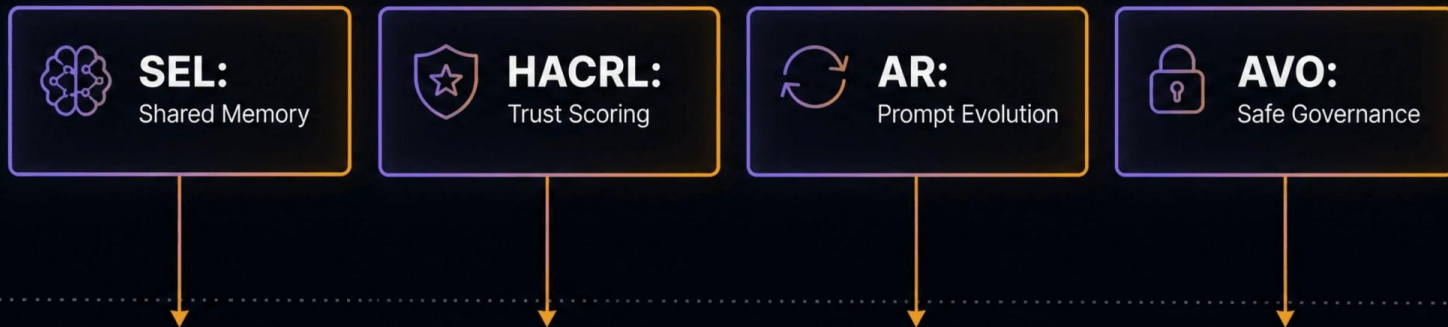
We will use this engine to illustrate
the four principles of self-improving AI.



The Architecture at a Glance

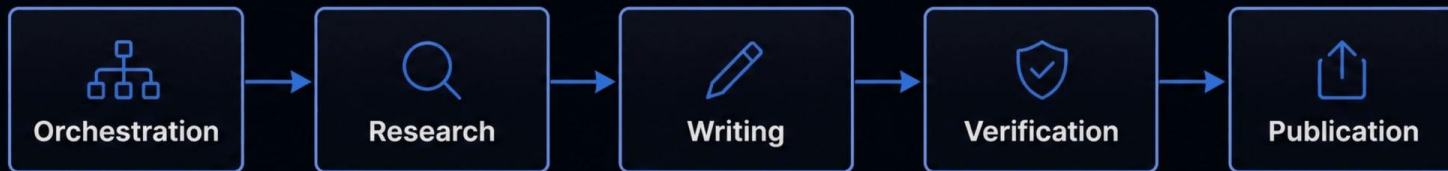
THE META-SYSTEMS

Self-improvement layer that manages the pipeline



THE PIPELINE

Execution layer that delivers content at scale



The pipeline itself is table stakes.

The four meta-systems that manage it are what make it self-improving.

Never trust a single model — the consensus approach



1. The Multi-Draft Pattern:

Claude (precise),
GPT-4.1 (creative),
GPT-mini (alternative).



2. Mathematical scoring

based on data density
(entities, statistics).



3. The Principle:

If the decision matters,
get multiple model perspectives.

```
production.log 2025-05-20 14:32:11Z

{
  "draft_a_claude": {"score": 0.88, "density": "high"},
  "draft_b_gpt4": {"score": 0.92, "density": "very_high"},
  "draft_c_gptmini": {"score": 0.75, "density": "medium"},
  "consensus_action": "Merge A and B"
}
```

STATUS: SUCCESS | MODELS: 3 | CONSENSUS: MERGE A AND B | VERSION: v1.2.0

Quality gates that catch what humans miss



Pass 1:

Factual accuracy scoring

(0.0-1.0 scale)



Pass 2:

Brand voice compliance and Anti-AI-Slop detection



Pass 3:

SEO and structural optimization



The Principle:

Verification must be **AUTOMATED**.
Define your quality criteria as code.

```
verifier.log 2025-05-20 14:32:11Z

[PASS 1] Factual Accuracy: 0.95
(DeepSeek R1)

[PASS 2] Anti-AI-Slop: FAILED
-> Flagged: "delve", "unleash", "synergy"
-> Action: Rewriting...

[PASS 3] SEO Optimization: 0.98

STATUS: Published

STATUS: PUBLISHED | RUN ID: 7f3c21a | DURATION: 00:00:04.21 | VERSION: v1.2.0
```

Your agents should learn from each other



THE SHIFT

The Problem: Most AI agents start with amnesia every single run.

The Solution: A Shared Experience Layer (SEL) that captures execution outcomes, scores them, and makes them retrievable by other agents.



KEY CONCEPTS

- Vector embeddings of task intent
- Automated quality scoring
- Cross-agent retrieval



Insight: If you have multiple AI agents or workflows, they should be building institutional knowledge.

Real retrieval output from a production system

```
1  ## Cross-Agent Experience Insights (HACRL Shared Experience Layer)
2  ### Experience 1 - GAM Researcher [HIGH TRUST] (Similarity: 87.4%)
3  - Domain: MEF | Category: Market Analysis
4  - Model Used: claude-sonnet-4.6 | Capability: deep_research
5  - Reasoning Steps: [Step 1... Step 2... Step 3...]
6  - Outcome Summary: Successfully synthesized 5 distinct
7  market reports into a unified thesis.
```



1. Trust Tiers:

The [HIGH TRUST] label ensures agents only learn from validated successes.



2. Semantic Matching:

Vector search finds the 87.4% most relevant past experience.



3. Cross-Pollination:

Agents are explicitly excluded from their *own* past experiences to force team learning.

Not all AI outputs deserve equal confidence.

- **The Problem:** How do you know which agent outputs to trust?
- **The Solution:** A continuous trust scoring system evaluating every agent on a rolling basis.
- **The Scale:** Runs every 6 hours, evaluating up to 5,000 experiences.
- **The Impact:** Trust scores cascade into the memory system — low-trust experiences get deprioritized.



You need a quantitative answer to: **'How reliable is this AI system right now?'**

The math behind confidence

THE TRUST SCORE FORMULA

	40%	Recent 7-Day Quality
+	30%	All-Time Quality
+	30%	Success Rate
×		Confidence Multiplier (Requires 20+ executions)

Key Insight: Recency-weighted scoring catches degradation early.

REAL JSON OUTPUT (TRUST SCORE)



```
{
  "agent_name": "mef-content-writer",
  "trust_score": 0.8421,
  "capability_ratio": 0.8105,
  "success_rate": 0.94,
  "avg_quality_recent": 0.88
}
```

Your prompts should get better every week without human intervention

THE SHIFT

The Problem: Prompts are static, but the world changes.

The Solution:

Weekly automated experiments that mutate prompts, run A/B tests, and promote winners.

KEY CONCEPTS

- Covers 9 agents with independent evolution cycles
- Uses an independent LLM Judge for scoring
- Autonomous Refinement (AR)



Insight: If you're still manually tuning prompts, you're doing it wrong.

A mutation that improved data completeness by 17 percentage points

AR Experiment Results — Real Data

- Baseline vs. Mutation A/B testing.
- Independent LLM Judge scoring against defined KPIs.
- Automatic promotion of winning prompts to production.



Key Insight: The system documents WHY improvements work, creating institutional knowledge about prompt engineering.

Canned Result (Real Data Output)

```
ar_experiment_result.json  PRODUCTION LOG
```

```
1  {
2    "status": "WINNER_PROMOTED",
3    "agent_name": "GAM Researcher",
4    "kpi": "data_completeness_percentage",
5    "baseline_score": 0.75,
6    "mutation_score": 0.92,
7    "delta": "+0.17",
8    "message": "Mutation improved data completeness
9               by explicitly requiring regulatory
10              landscape coverage."
  }
```

Autonomous improvement requires autonomous safety



THE SHIFT

The Problem: If AI systems can modify themselves, how do you prevent catastrophic changes?

The Solution:

A governance layer that diagnoses issues, proposes solutions, requires approval, validates safety, and maintains rollback capability. This is the 'adult in the room.'



KEY CONCEPTS

- Automated issue diagnosis
- Proposed solutions with required approval
- Safety validation checks
- Built-in rollback capability



Insight: The faster your systems move, the stronger your guardrails need to be.

What autonomous system monitoring looks like

AVO Daily Diagnostics — The Morning Report



System health status, success rate, execution time, and cost.



Top failing workflows identified automatically.



Optimization proposals with priority levels.



Key Insight: The system surfaces its own problems before humans notice them.



Canned Result: Microsoft Teams Message from AVO



AVO

APP

8:00 AM



AVO Daily Diagnostics Report

System Health: **HEALTHY**
Success Rate: **98.5%** (1,240 total, 18 failed)
Avg Execution Time: **12.4s**
Total Cost: **\$4.32**

Top Failing Workflows:

- Content Verifier: **12 failures** (OpenRouter timeout)

Optimization Proposals (2):

- **[HIGH]** Restructure Flow in Content Verifier
- **[LOW]** Clarify Instructions in Social Media Generator

AVO Autonomous Operations Platform | Daily Diagnostics

Trust but verify — the deployment safety net

PRE-DEPLOYMENT CHECKS

- 1 | Credential Integrity
- 2 | Webhook Preservation
- 3 | Node-Count Delta (Max 5)
- 4 | Error Handler Preservation
- 5 | Connection Validation
- 6 | Schema Consistency

THE ROLLBACK GUARDIAN

- Full snapshot taken before every change.
- **The Three-Strike Rule:** If a workflow rolls back 3 times, the system triggers an automatic lock.



**3 FAILURES = BLOCK
AUTOMATED CHANGES +
HUMAN ESCALATION**



Insight: Build your safety protocols **BEFORE** you need them.

The flywheel: each system makes the others smarter



Insight: The goal isn't four separate systems. It's one self-reinforcing loop. **This is the compound interest of AI.**

Your diagnostic checklist



1. MEMORY

How are our AI agents sharing what they learn?

If the answer is "they don't" → you need a SEL equivalent.



2. TRUST

How do we quantitatively measure AI output reliability?

If the answer is "we spot-check" → you need a HACRL equivalent.



3. EVOLUTION

How are our prompts improving over time?

If the answer is "manually" → you need an AR equivalent.

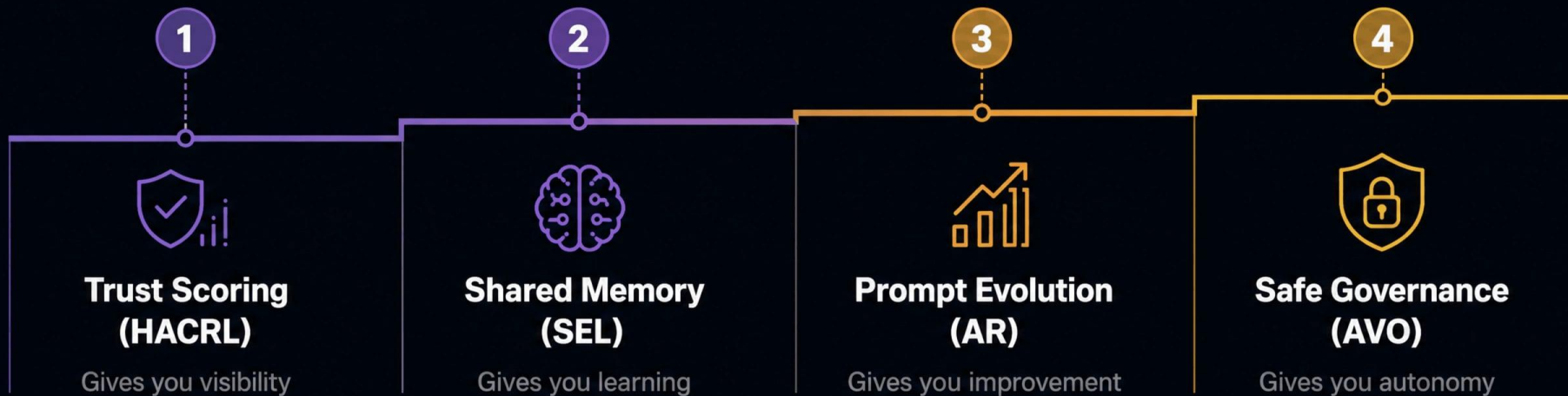


4. GOVERNANCE

What happens when an autonomous change breaks something?

If the answer is "we find out from users" → you need an AVO equivalent.

You don't need to build all four at once



Insight: Pick the one that solves your biggest pain point today.

What to remember from this session



1. Production AI requires management systems, not just better prompts.



2. Self-improving systems compound their gains over time (like compound interest).



3. Safety and governance are prerequisites for autonomy, not afterthoughts.



4. These are architectural patterns — implement them on whatever stack you use.



5. Start with measurement (trust scoring), then add learning, evolution, and governance.

Thank You *and* Q&A

*"We don't just build the engine.
We build the factory that improves the engine."*



Mario Facussé
Managing Director



MEF Solutions



Austin AI Alliance Summer Intensive
June 22-24, 2026

Case study provided by **MEF SOLUTIONS**



Connect with me on LinkedIn