Market-Based Orchestration of Specialized AI Agents for Complex Knowledge Work

Stage 4A: Architecture, Skill Agents, and First Results

April 2026

Abstract

We present an architecture for autonomous AI agent economies in which specialized skill agents — each combining an LLM with a single real-world tool — collaborate to solve complex knowledge tasks. A market-based orchestrator decomposes tasks, routes subtasks to specialists using embedding-based reputation, and integrates results. We validate the architecture with real Claude API calls and real World Bank data, demonstrating that (1) skill teams with real tool access outperform single agents by +8.8pp on complex cross-domain tasks, (2) agents autonomously retrieve and cite real economic data through API tool use, and (3) market-based routing achieves parity with rule-based assignment within 10 tasks, with early evidence of adaptation advantages. We introduce *embedding-based reputation* — a non-parametric reputation estimator over continuous task space that requires no pre-defined categories and scales to arbitrary domains. The architecture is designed as a general-purpose system for complex work, not limited to specific subject areas.

Contents

1 Introduction

Current AI systems treat tool access as a configuration problem: a human administrator decides which tools an agent can use, and the agent operates within those constraints. There is no mechanism for agents to autonomously discover, evaluate, or acquire capabilities, and no market to coordinate which agent should handle which task.

This paper presents an architecture that replaces central configuration with market-based coordination. The central question is:

> *Can a market of specialized AI agents, each equipped with a single real-world tool, outperform centrally orchestrated systems on complex knowledge tasks?*

This work builds on three prior stages: incentive framing for single agents (Stage 1), persistent reputation in homogeneous settings (Stage 2), and decentralized markets with tool-based specialization (Stage 3). Stage 4A introduces real tool execution, universal reputation, and the skill-agent paradigm.

1.1 Contribution

1. **Skill-agent architecture**: each agent IS one tool — a specialized LLM instance with a domain-specific system prompt and a real API connection. Complex tasks are decomposed and routed to specialist agents.
2. **Embedding-based reputation**: a non-parametric reputation estimator that operates in continuous task-embedding space, requiring no pre-defined categories.
3. **Real tool validation**: agents call the World Bank API during task execution, retrieving and citing actual economic data — not simulated, not described in prompts, but executed through the Anthropic tool_use API.
4. **Market vs. rules comparison**: first empirical results comparing market-based orchestration with rule-based assignment on identical tasks.

2 Architecture

2.1 Design Principle: Agent = Skill

Rather than giving agents multiple tools, each agent IS one skill:

Table 1: Skill agent roster

Agent	Specialty	Tool	Type
`economist`	Macroeconomic data & analysis	World Bank API	Real API
`data_researcher`	Finding indicators & datasets	World Bank Search	Real API
`country_analyst`	Cross-country comparison	World Bank API	Real API
`policy_analyst`	Regulatory & policy analysis	—	LLM only
`risk_assessor`	Risk identification & frameworks	—	LLM only
`writer`	Professional document drafting	—	LLM only
`tech_architect`	Technical system design	—	LLM only
`integrator`	Synthesis of multiple analyses	—	LLM only

Three agents have real API access (World Bank), five operate as pure LLM specialists with domain-specific system prompts. All use Claude Sonnet as the base model.

2.2 Task Pipeline

1. **Task arrival** — a complex question or assignment enters the system
2. **Decomposition** — an LLM call breaks the task into 2–4 focused subtasks and suggests specialist assignments
3. **Routing** — each subtask is assigned to a specialist agent (via rules or market)
4. **Execution** — each specialist executes its subtask, potentially making real API calls
5. **Integration** — the integrator agent synthesizes all specialist outputs into a coherent response
6. **Review** — an LLM evaluates the final output on multiple quality dimensions
7. **Reputation update** — quality scores are recorded in each agent's embedding-based reputation

2.3 Real Tool Execution

Agents with API tools use the Anthropic `tool_use` feature. When Claude determines that data is needed, it generates a tool call:

```
worldbank_get_data(country_code="IT", indicator="GC.DOD.TOTL.GD.ZS", years=5)
→ Returns: Italy debt-to-GDP: 140.6% (2024), 137.3% (2023), ...
```

The system executes the call against the live World Bank API and returns real data to Claude, which incorporates it into its response. This is not simulated — the numbers come from the actual World Bank database.

2.4 Two Orchestration Modes

We compare two approaches to subtask routing:

Table 2: Orchestration modes

	Rule-Based ("Law & Order")	**Market-Based**
Assignment	Keyword matching	Embedding-based reputation
Adaptation	None (fixed rules)	Learns from outcomes
Exploration	None	Epsilon-greedy (decaying)
Cold start	Immediate (rules work from R1)	Needs calibration
Novel tasks	Fails if no keyword matches	Transfers from similar tasks

3 Embedding-Based Reputation

3.1 Motivation

Fixed-category reputation (e.g., separate scores for "economics", "policy", "risk") does not scale. A system that handles fiscal policy today may face Kubernetes debugging or music composition tomorrow. We need reputation that works across arbitrary domains without reconfiguration.

3.2 Design

Each agent maintains a set of observations $\mathcal{O} = \{(\mathbf{e}_t, q_t, r_t, \tau_t)\}$ where $\mathbf{e}_t$ is the task embedding, q_t is observed quality, r_t is the role, and τ_t is the round.

Predicted quality for a new task with embedding $\mathbf{e}^*$:

$$\hat{q}(\mathbf{e}^*) = \frac{w_0\mu_0 + \sum_t w_t \cdot q_t}{w_0 + \sum_t w_t}, \qquad w_t = \cos(\mathbf{e}^*, \mathbf{e}_t) \cdot e^{-\lambda(\tau_{\text{now}} - \tau_t)} \tag{1}$$

where $\mu_0 = 0.45$ is the prior mean, $w_0 = 2.0$ the prior weight, and $\lambda = 0.693/h$ with half-life $h = 50$.

3.3 Properties

- **No fixed categories** — reputation is defined over continuous embedding space
- **Automatic transfer** — performance on "EU fiscal policy" informs predictions for "UK monetary policy" but not "Kubernetes debugging"
- **Graceful cold-start** — prior dominates with few observations, data dominates with many
- **Recency weighting** — old observations decay, tracking capability changes

4 Results

4.1 Result 1: Real Tool Access Matters

We compared Claude's response to an economic question with and without real World Bank API access:

Table 3: Impact of real tool access (single task, Claude Sonnet)

Dimension	Without tools	With tools	Δ
Factual accuracy	0.85	0.90	+0.05
Completeness	0.60	**1.00**	**+0.40**
Data quality	0.75	0.80	+0.05
Overall	0.78	**0.91**	**+0.13**

Without tools, Claude responds: "I don't have access to real-time data." With tools, Claude calls `worldbank_get_data('US', 'UNRATE')` and provides current figures with citations.

4.2 Result 2: Skill Teams Outperform Single Agents on Complex Tasks

We compared a single agent (with all tools) against a decomposed skill team (specialist agents + integrator) on 5 diverse tasks:

Table 4: Single agent vs. skill team (5 tasks, real LLM calls)

Task type	**Single**	**Team**	**Winner**
AI regulation (cross-domain)	0.30	0.40	Team
Credit risk + compliance	0.50	**0.70**	Team
White paper (simple)	**0.80**	0.70	Single
Vendor risk framework	**0.66**	0.50	Single
Regulatory filing (cross-country)	0.30	**0.70**	Team
Average	0.512	**0.600**	**+0.088**

The team wins on complex, cross-domain tasks (+0.20 on regulatory filing). The single agent wins on focused, simple tasks. This suggests that the orchestration mechanism should be *adaptive* — decompose when beneficial, skip decomposition for simple tasks.

4.3 Result 3: Market vs. Rule-Based Orchestration

We compared two routing mechanisms on 10 tasks with real LLM calls:

Table 5: Market vs. rule-based routing (10 tasks, real LLM + real tools)

Metric	**Rules**	**Market**	**Δ**
Average quality	0.530	0.530	0.000
Wins	5	4	
First half quality	0.524	**0.560**	+0.036
Second half quality	**0.536**	0.500	−0.036

At 10 tasks, the market achieves parity with rules. The market shows early advantage in the first half (while exploring) but does not yet surpass rules in the second half. This is expected: 10 tasks provide insufficient data for the reputation system to calibrate.

4.4 Result 4: Orchestrated Task Execution (Case Study)

A concrete example demonstrates the full pipeline:

> *Task: "Analyze the debt-to-GDP trajectory of Italy compared to Germany and assess the implications for EU fiscal policy."*

The orchestrator decomposed this into 4 subtasks:

1. `data_researcher` (4 API calls) — retrieved World Bank debt indicators
2. `country_analyst` (18 API calls) — compared fiscal metrics for Italy and Germany
3. `policy_analyst` (0 API calls) — analyzed EU fiscal rule implications
4. `integrator` (0 API calls) — synthesized into coherent report

The final output correctly cited Italy's debt-to-GDP at ∼140% and Germany's at ∼68%, sourced from live World Bank data. Total: 22 real API calls, 4 specialists, 156 seconds.

5 Swarm Simulation (Supplementary)

To test market dynamics at scale, we ran a numerical simulation with 2,000 agents over 200 rounds. While the simulation uses computed quality (not real LLM outputs), it reveals emergent properties of the market mechanism.

5.1 ELO-Based Pairwise Evaluation

Using pairwise comparison (inspired by Chatbot Arena), the market condition starts below rule-based assignment but crosses over at round ∼20:

Table 6: ELO trajectory (2,000 agents, 200 rounds, numerical simulation)

Round	Baseline	Central	Market	Market > Central?
0	1,185	1,231	1,184	No
10	1,030	1,373	1,197	No
20	967	1,302	**1,331**	**Yes**
100	819	1,425	1,356	No
199	774	1,293	**1,533**	**Yes**

Final ELO: Market = 1,510, Central = 1,324, Baseline = 766.

5.2 Emergent Tool Adoption

In the simulation, 74.2% of agents independently converged on the same best tool (`fre4x-fred`), a stronger concentration than central planning's 50% allocation. All popular tools reached the +200% price cap through demand-driven dynamic pricing.

5.3 Endogenous Specialization

Specialists invested aggressively (average cash remaining: $0.80, 2.0 tools) while generalists hoarded cash ($5.30, 1.0 tools) — without any instruction to do so.

6 Discussion

6.1 What Works

1. **Real tool access** produces measurably better outputs (+13pp overall, +40pp on completeness). This is the clearest finding.
2. **Skill teams** outperform single agents on complex cross-domain tasks (+8.8pp). The architecture successfully decomposes, routes, and integrates.
3. **The agent = skill paradigm** creates genuine specialization. Each agent does one thing well, and the system combines them.

6.2 What Remains Open

1. **Market vs. rules at scale.** With 10 real tasks, the market achieves parity but does not clearly surpass rules. The reputation system needs more observations to calibrate. Longer runs with real LLM calls are needed but costly (∼$3–5 per 10 tasks).
2. **Cold-start problem.** The market's epsilon-greedy exploration (60% exploration early, 5% late) prevents winner-takes-all but does not yet demonstrate clear superiority over keyword matching.
3. **Evaluation.** We use LLM-as-judge for the primary quality metric, which introduces subjectivity and self-enhancement bias. Rubric-based evaluation exists but covers only a subset of tasks.

6.3 Limitations

1. Only one real API (World Bank). Production deployment would include FRED, government data APIs, compliance databases, and domain-specific tools.
2. The swarm simulation uses computed quality, not real outputs. The crossover finding should be validated with real LLM calls at scale.
3. All agents use the same base model (Sonnet). Cross-model markets (Haiku/Sonnet/Opus) would add another dimension of heterogeneity.

7 Architecture Summary

The complete system comprises:

Table 7: System modules

Module	Purpose
`skill_agents.py`	Skill agent definitions, execution, orchestration
`real_tools.py`	World Bank API integration (tool_use)
`reputation.py`	Embedding-based universal reputation
`task_generator.py`	Dynamic task generation with disruptions
`budget.py`	Complexity estimation and budget assignment
`elo.py`	ELO-based pairwise evaluation
`evaluation.py`	Rubric-based objective evaluation
`experiment_market_vs_rules.py`	A/B experiment framework

8 Next Steps

1. **More real tools** — integrate FRED, government APIs, compliance databases to create genuine tool heterogeneity across agents
2. **Longer market runs** — 50–100 tasks with real LLM calls to observe reputation calibration and market convergence
3. **Adaptive decomposition** — the system should learn when to decompose (complex tasks) and when to skip (simple tasks)
4. **Cross-model agents** — mix Haiku, Sonnet, Opus as base models to test whether cheap-but-well-tooled agents can compete with expensive-but-generic ones
5. **Dynamic tool discovery** — agents browse npm/MCP registries autonomously rather than choosing from a curated store

9 Conclusion

We present a working architecture for market-based orchestration of specialized AI agents. The key findings:

- **Real tools produce real improvement.** Agents with World Bank API access cite actual data (Italy debt-to-GDP: 140%, Germany: 68%) instead of saying "I don't have access to real-time data." Quality improves by +13pp.

- **Skill teams beat single agents on complex tasks** (+8.8pp), with the advantage concentrated on cross-domain questions requiring multiple forms of expertise.
- **Market-based routing achieves parity with rules within 10 tasks**, with the explore-exploit tradeoff as the key design challenge. The market's advantage is expected to emerge with more calibration data and novel task types.
- **Embedding-based reputation** provides a universal, category-free mechanism for tracking agent capability across arbitrary domains.

The architecture is designed as a general-purpose system: not limited to economics or policy, but extensible to any domain where specialized tools exist. The market does not yet clearly beat central planning on a small task set — but it provides the infrastructure for adaptation, learning, and emergent specialization that static rules cannot.