Enterprise Multi-Agent Decision Systems with Claude

- Hands-on

Course Overview

This intensive, hands-on workshop teaches participants to engineer a working, enterprise-pattern multi-agent decision system directly with Claude. Using a realistic consulting/procurement scenario, teams build a workflow that gathers evidence, validates calculations, challenges assumptions, controls cost through model routing, and produces an executive-ready recommendation supported by traceable evidence and a clear path to production consumption.

The session walks participants through the complete lifecycle: from workflow audit and operating contracts to specialist subagent delegation, controlled tool integration, cost-aware model routing, deterministic validation, adversarial review, production hand-off, and reviewable delivery.

Working in teams of 3 to 4 from a prepared but intentionally incomplete repository, participants complete and test a Claude-native decision system. Project instructions are configured so the primary Claude workflow delegates specialized work to four subagents (Contract Analysis, Policy Compliance, Alternative Recommendation, and Red Team). The agents exchange structured JSON artifacts for sources, claims, calculations, caveats, and open questions rather than relying on unstructured conversational handoffs.

Participants leave with a fully functional Claude workspace, reusable project assets, a packaged skill, a working validation pipeline, basic cost-routing rules, and a clear production hand-off pattern that can be extended to real enterprise decision and repetitive-task workflows.

Who Should Attend

- People who want to build multi-agent decision systems in Claude but prefer not to write traditional control-plane code
- Technical practitioners and team leads who are comfortable with Claude Projects, structured outputs, and prompt-based orchestration
- Engineers who need a fast, Claude-native pattern for internal decision workflows and are willing to accept probabilistic agents constrained by contracts and validation rather than fully code-enforced agents

What You'll Learn

- Define Operating Contracts & Schemas: Translate an audited workflow into an operating contract (`contracts/decision-os.md`), repository instructions, a reusable skill, and machine-verifiable JSON schemas.
- Build Controlled Tool Integrations: Configure native tools and adapters for approved retrieval, document access, and deterministic calculation verification inside the Claude execution environment.
- Orchestrate Native Subagents: Configure project instructions and delegation boundaries so specialist subagents operate in separate execution contexts with explicit return contracts.
- Implement Deterministic Validation Gates: Build multi-layered acceptance tests enforcing schema compliance, citation metadata resolution, mathematical accuracy, and policy rules.

- Produce Reviewable Deliverables: Generate an evidence-backed deliverable containing the decision, categorical confidence assessment, verified calculations, explicit caveats, and flagged items requiring human review.
- Apply Cost-Aware Model Routing: Route sub-agent work by task criticality so expensive models are used only where necessary and cheaper/faster models handle extraction, classification, or preliminary checks.
- Design Production Hand-off: Define how the structured delivery package is consumed by existing systems or human reviewers and establish basic ownership and escalation after the workshop.

Outline

1. Orientation & Claude-Native Setup
   - Map of the day and how this builds on prompting + basic agents
   - Claude Projects, Artifacts, structured outputs, and tool-use patterns used throughout the workshop
   - Repository walkthrough and role assignments

2. Architecture Briefing & Baseline Run
   - Walkthrough of the decision-system reference architecture and team role assignments
   - Executing initial baseline test to observe failing validation assertions

3. Operating Contracts & Schemas
   - Teams complete `contracts/decision-os.md` defining system boundaries and rules (including a lightweight cost/token consideration)
   - Complete `schemas/delivery.schema.json` to enforce machine-readable output contracts
   - Author the reusable skill definition and update project instructions

4. Specialist Subagents & Controlled Tools
   - Configure the four specialist subagents to operate in separate contexts with explicit return contracts
   - Extend `tools/calculations.py` with custom verification logic
   - Test subagents independently against sample data fixtures

5. Cost-Aware Model Routing
   - Define simple routing rules by task criticality (stronger model for final recommendation and Red Team; faster/cheaper model for extraction, classification, or preliminary checks)
   - Add a lightweight cost or token awareness note into the workflow
   - Tool Engineer and Workflow Architect lead this segment

6. Primary Workflow Delegation & Handoffs
   - Design the orchestration
   - Configure project instructions so the primary Claude workflow delegates to subagents
   - Implement structured handoffs: Input Payload → Specialist Subagents → Merged Draft
   - Enforce strict return contracts so each specialist returns only required results

7. Deterministic Gates & Red-Team Review
   - Configure the Red-Team subagent for adversarial review
   - Extend `validators/validate_delivery.py` to enforce schema validation, citation resolution, and math re-computation
   - Explicitly surface one recovery / partial-result path
   - Route unresolved findings, unsupported claims, or policy violations to human review

8. Adversarial Fixture Testing & Repair

- Run the workflow against `conflicting/` and `missing-sources/` data fixtures
- Iterate on agent instructions and validator assertions until failure handling paths execute cleanly

9. Production Hand-off & Consumption

- What happens to the generated delivery package after it is produced?
- How an existing approval system, ticket queue, or human reviewer consumes the structured output
- Basic ownership, monitoring, and escalation rules after the workshop ends
- Teams sketch the concrete hand-off for their procurement scenario

10. Evidence-Pack Review, Cost Snapshot & Close

- Inspect generated delivery artifacts across teams
- Quick observations on cost/routing choices made during the day
- Wrap-up on transferring the completed skill and pattern to future client engagements

Prerequisites

- Students should have their own Claude Pro / Team / Enterprise access
- Basic experience with prompting and agents (builds on prompting + basic agents)