AGENT ORCHESTRATION

Agent Orchestration Blueprint

A worksheet for designing agentic workflows around triggers, tools, approvals, handoffs, observability, and recovery.

WHAT THIS TEMPLATE HELPS YOU DECIDE

Move from prompt-driven automation to tool-first orchestration. Define what starts the workflow, what the agent may call, what requires approval, and what evidence proves completion.

BEST FOR

- Teams planning agent workflows beyond a chat interface
- Operations with approvals, tool calls, routing, or cross-system handoffs
- Builders who need typed workflow contracts before implementation

OUTPUTS

- A tool-first orchestration map
- Approval and handoff boundaries
- Execution receipts and observability signals

STEP 1

Frame the workflow trigger

Agentic systems need explicit starts, stops, and handoff rules. The trigger defines when orchestration begins and what context enters the workflow.

Workflow frame

Trigger event
Examples: new lead, support request, renewal risk, report variance, document upload.

Initial context package
Name the records, documents, messages, or metrics the agent receives first.

Successful completion
Describe the evidence that proves the workflow ended correctly.

Actors and handoffs

Agent role	What does the agent coordinate, and what is outside its role?
Human owner	Who owns exceptions, approvals, and final accountability?
System handoff	Which tools, queues, databases, or documents may receive outputs?
Stop condition	When should the agent pause, ask for approval, or stop entirely?

STEP 2

Make tools deterministic

If a workflow step can be performed by a tool, service, rule, or database query, the agent should use that contract instead of improvising from natural language.

Tool registry map

Read tools	Which tools retrieve records, context, search results, metrics, or policy references?
Reasoning tools	Which classifiers, scorers, validators, or calculators structure the decision?
Write tools	Which tools can create, update, send, assign, publish, or trigger work?
Approval gate	Which write operations require preview and human approval before execution?
Fallback	What happens when a tool fails, times out, or returns incomplete state?

Agents should coordinate deterministic systems. The more important the action, the more explicit the input schema, validation rule, permission check, and receipt.

STEP 3

Define observability and receipts

Agent orchestration becomes operational infrastructure only when people can see what happened, why it happened, and what still needs review.

Execution visibility

- [ ] Each tool call has a trace ID or workflow ID
- [ ] Inputs are validated before execution
- [ ] Approval decisions are recorded with owner and timestamp
- [ ] Tool results are summarized as execution receipts
- [ ] Failures create a visible retry, fallback, or escalation path
- [ ] Operators can review recent activity by workflow and account

Receipt design

Receipt fields

Example: workflow ID, tool name, input summary, result, approver, timestamp, next action.

Operator view

Where should a person inspect status, approvals, failures, and outcomes?

Design agent workflows as operations, not demos.

IntelliSync helps teams build approval-aware, tool-first orchestration with typed contracts, visible receipts, and recovery paths.

Open Architecture Assessment