elevenai.co

FINOS NY

Agentic workflows your auditors can live with

Why BPMN fits agentic orchestration, demonstrated with Fluxnova

Chris Miller · 11ai



11ai · FLUXNOVA FOUNDING PARTNER

elevenai.co

An agent needs three things that are hard to bolt on later

Discoverable tools

What can the agent call? With what inputs? What does each tool do? The answer has to come from somewhere the agent can read, and that humans maintain.

Bounded autonomy

Room to plan and act, but inside limits: which tools, for how long, and who decides when it stops. The model should never be the last line of defence.

An audit trail

Which tools ran, in what order, with what data, who overrode what. Reconstructable months later, by someone who was not in the room.

elevenai.co

BPMN in ninety seconds

A diagram standard

Business Process Model and Notation, an OMG standard since 2004. A small set of shapes that analysts, engineers and auditors all read the same way.

The diagram is executable

A process engine runs the model directly. Each shape is a step the engine starts, waits on, finishes and records.

One artefact for everyone

The picture the business signed off is the thing in production. Versioned, deployable, with history per element.

THE SHAPES YOU WILL SEE TODAY



Event
Something happens: a dispute arrives, a case ends

Task
A unit of work. A cog means a system does it

User task
A person does it, from a task list with a form

Business rule task
A DMN decision table decides; the result is a variable

Ad-hoc subprocess
A scope whose steps run in any order, or not at all

Gateway
A decision point that routes on data

Timer boundary event
Interrupts a step when time runs out

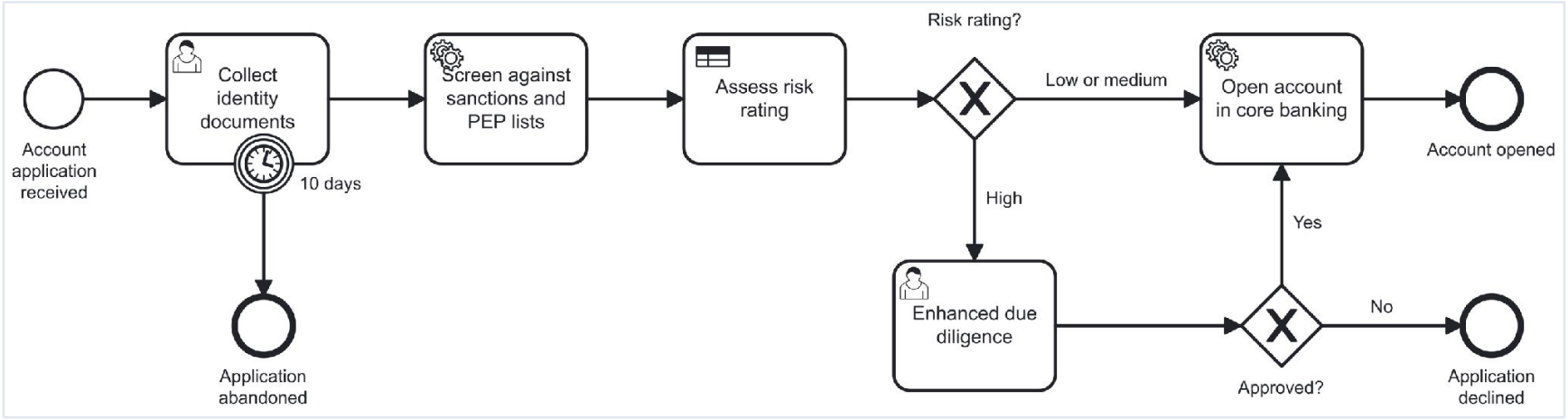
Message boundary event
Interrupts a step when an outside event arrives

Sequence flow
The order things happen in

elevenai.co

Client onboarding, as a bank would draw it

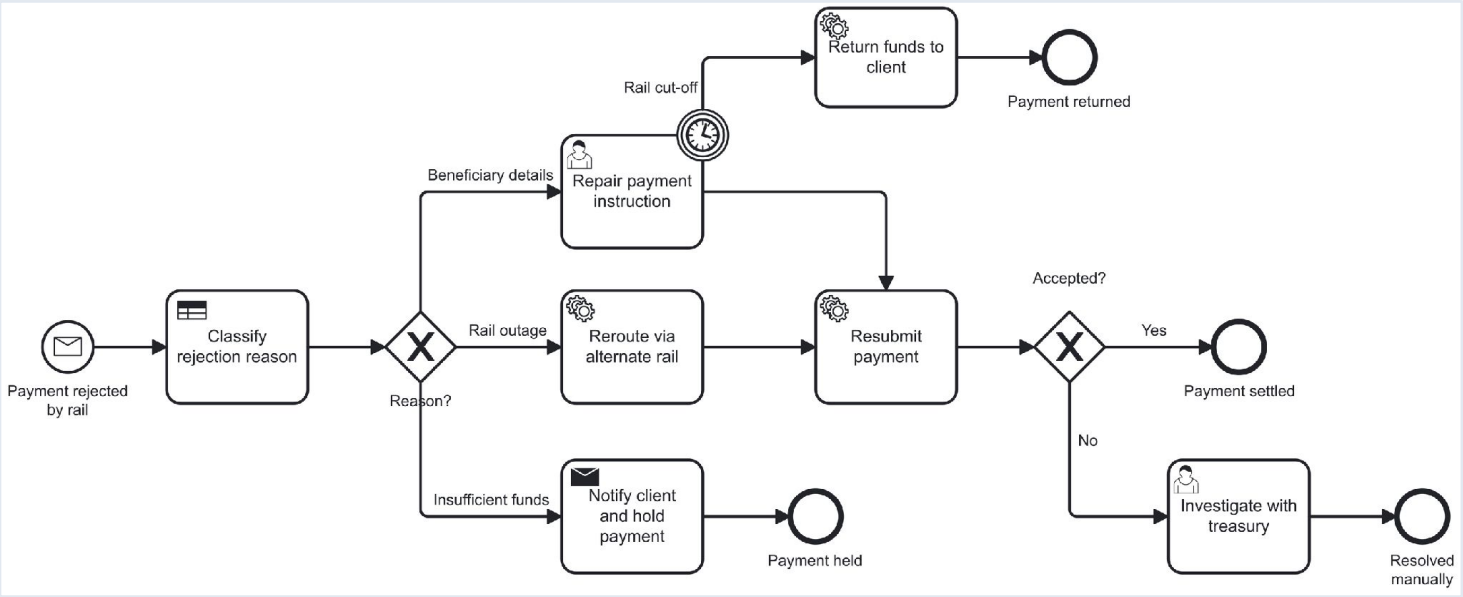
A human collects documents, a system screens against sanctions and PEP lists, a DMN table rates the risk, compliance reviews the high-risk cases. Ten days without documents and the application lapses.



elevenai.co

And a payment the rail rejected

A rules task classifies the rejection, operations repair bad beneficiary details, a service reroutes around a rail outage, the client is told when funds are short. Miss the cut-off and the money goes back.



elevenai.co

BPMN standardized all three, twenty years ago

A BPMN process is self-describing. Everything an agent needs to know is already in the model, because people needed it too.

Agent need	BPMN construct	What it gives you
Discoverable tools	Activities with ids, names, documentation; typed inputs and outputs	A tool catalogue that is also the diagram your analysts drew
Bounded autonomy	Subprocess scopes, gateways, boundary events, user tasks	Explicit limits on what, how long, and who decides
Audit trail	History of every activity instance, variable and decision	Replayable, queryable, already in your monitoring tools

Processes become tools agents can discover and invoke. Subprocesses become the scope an agent is allowed to act in.

elevenai.co

Two patterns, both plain BPMN

1. Processes as tools

A start event annotated for MCP becomes a tool: typed parameters in, process variables out. Any MCP client can discover and invoke it. The process is the tool implementation, with all its controls intact.

2. Ad-hoc subprocess as agent scope

Inside a governed scope, the model picks which activities to run and in what order. Each child activity is a tool. The process decides what happens before and after.

Today's demo is pattern 2. Pattern 1 is the same engine, the same plugin suite.

elevenai.co

Fluxnova: the open source process engine at FINOS

BPMN 2.0 and DMN engine

Java, embeds in Spring Boot, REST API, built on the Camunda 7 community codebase

Control Center and web apps

Monitoring, history, incidents, tasklist, a modeler for BPMN, DMN and forms

AI plugin suite

MCP server plugin, MCP start events, and the agentic subprocess plugins used today

FINOS governance

Apache 2.0, open roadmap, built by and for financial services firms

Part of FINOS Governance as Code. Fluxnova is the BPMN/DMN orchestration layer in the FINOS AI governance-as-code pipeline, next to the AI Governance Framework (AIGF). AIGF's human-in-the-loop approval gate (AIR-PREV-024) is the control the demo's gates map to.

fluxnova.finos.org · ai.finos.org/governance-as-code · air-governance-framework.finos.org

elevenai.co

Pattern 1: a start event becomes an MCP tool

```
<startEvent id="start"
    mcp:toolName="SubmitExpenseClaim"
    mcp:description="Submits a new expense claim
                     for management approval">
  <extensionElements>
    <mcp:parameters>
      <mcp:parameter paramName="claimantId"
paramType="string"/>
      <mcp:parameter paramName="amount"
paramType="number"/>
    </mcp:parameters>
    <mcp:returnVariables>
      <mcp:returnVariable paramName="claimId"
paramType="string"/>
    </mcp:returnVariables>
  </extensionElements>
</startEvent>
```

Discovered
The engine registers one MCP tool per annotated start event, at deployment and at startup.

Typed
Parameters are declared with names and types. The description is what the model reads.

Governed
Invoking the tool starts a process instance. Approvals, timers and history apply as always.

Answered
Return variables are collected up to the first commit point and sent back to the client.

SECTION 02

The ad-hoc subprocess as the agent's room to work

A governed, observable scope in which the model decides what to do next.



elevenai.co

Anatomy: four things turn a subprocess into an agent

agent:config

The system prompt and provider. This is where the business rules live, in plain language the compliance team can read.

agent:context

The process variables the model is allowed to see. Re-read before every turn, so it learns what tools produced.

Child activities are the tools

The activity id is the tool name; name and documentation are the description. No free-form arguments: tools read and write process variables.

The process keeps control

Entry and exit are sequence flows. What happens after the agent stops is a gateway, not the model's opinion.

elevenai.co

The loop the engine runs on the agent's behalf

Enter scope

Engine queues the first orchestration job

Call the model

System prompt + context variables + tool catalogue

Dispatch tools

Each requested tool starts its activity inside the scope

Activities run

Service tasks, user tasks, anything BPMN can run

Feed back

When all tools finish, context is re-read and the model is called again

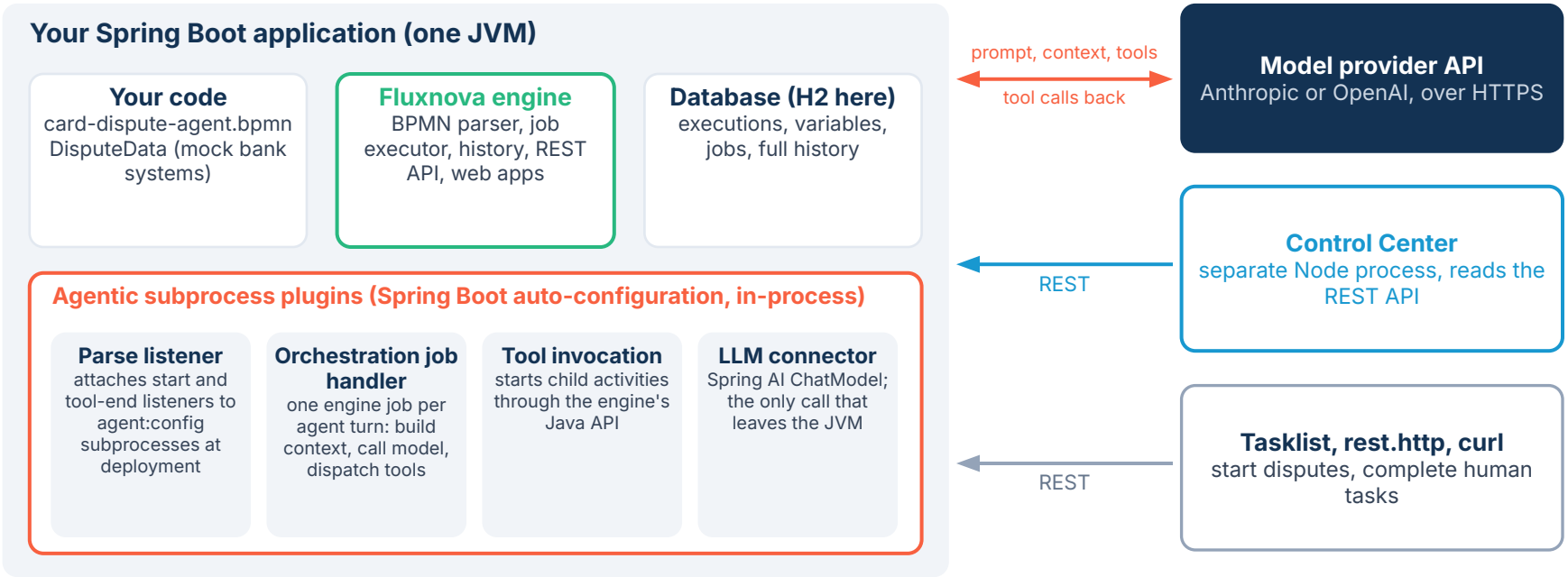
Stop

No tool calls returned: the scope completes and the sequence flow continues

The amber steps are the model; they repeat until it stops calling tools. Everything else is the engine, and it is what gets recorded.

elevenai.co

Where it all runs: one JVM, one transaction log

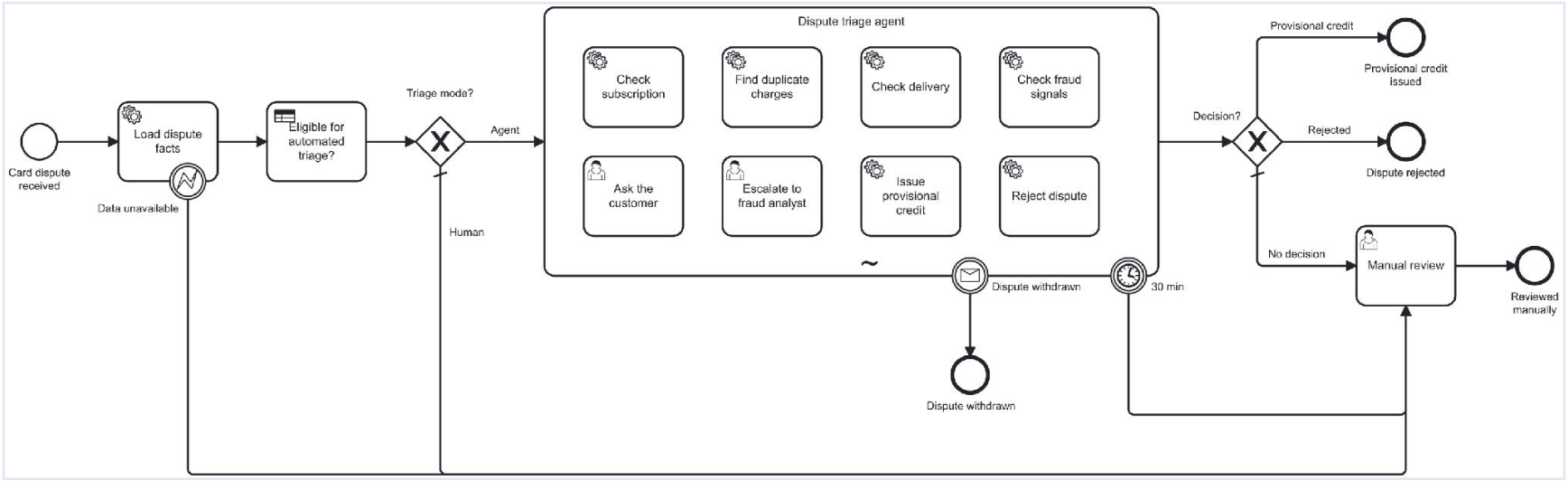


Every agent turn is an engine job in the same database transaction log as the rest of the process. The model never touches the engine; it only answers questions.

elevenai.co

Demo: a gate, then one agent with eight tools

Before the agent: the process loads the facts and a DMN table decides whether the agent gets the case at all. Inside: the prompt is a policy, not a script. Gather only the evidence the claim calls for, ask the customer if the evidence contradicts them, escalate on fraud or above the credit cap. Which tools, in what order, and when to stop is the model's call.



elevenai.co

What you are about to see

Dispute	The customer says	Likely path	Decided by
DSP-1001	"I cancelled my subscription but was charged again"	Check subscription: cancelled before the charge. Credit.	Agent
DSP-1002	"I did not make this purchase"	Check fraud signals: unknown device, wrong city, 2,450 USD. Escalate.	Analyst
DSP-1003	"The parking meter charged me twice"	Find duplicate charges: one 31 seconds later. Credit.	Agent
DSP-1004	"My jacket never arrived"	Check delivery: signed for by the front desk. Ask the customer, then decide or escalate.	Agent or analyst
DSP-1005	"Charged twice" (restricted account)	Never reaches the agent: the DMN gate routes restricted accounts to a person.	Person, by rule
DSP-1006	"I cancelled my gym membership" (899 USD)	Check subscription: cancelled. Credit refused by the cap in code. Escalate.	Analyst

03 · LIVE DEMO

Live demo

Six disputes → watch the paths diverge → play customer and analyst → withdraw one → read the trail

01

Control Center

Six disputes, six different paths; then withdraw one mid-investigation

02

Tasklist

The customer's answer and the analyst's decision are just tasks, with forms

03

Variables

The evidence each tool gathered, disputeDecision, decidedBy, and the full conversation

If the network fails: the recorded run is on the next two slides.

elevenai.co

Bounded autonomy, drawn on the diagram

Bounded before it starts
A DMN table decides whether the agent gets the case at all. Restricted accounts and frequent disputers go to a person; the rule is versioned with the model, not buried in a prompt.

The gateway decides, not the model
After the agent stops, a gateway routes on disputeDecision. The default path is a human review. The model cannot invent an outcome.

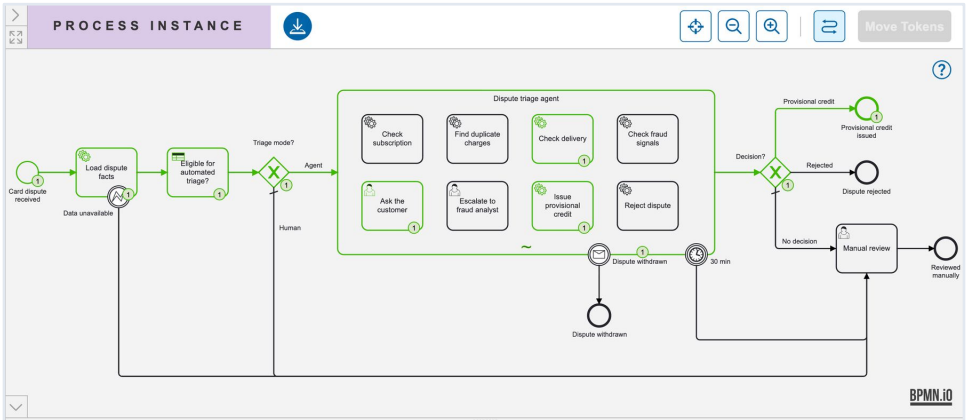
A human is a tool
Escalation is a user task inside the scope. The agent waits for it like any other tool, and the decision is recorded as the analyst's.

Bounded in time and by events
A timer hands the case to a person after thirty minutes. A message boundary cancels the agent the moment the customer withdraws. An error boundary catches a tool that cannot do its job.

Bounded in reach
Tools take no arguments from the model; they read and write named process variables. The credit tool enforces the bank's cap in code: above it, it refuses and the model has to escalate.

elevenai.co

The audit trail comes for free



Variables (19) Incidents (0) Called Process Instances (0) History (45) Decision Instances (1)

Filter on History Type: All

Start Time	End Time	Duration	Task/Event Name	History Type	Entity Type
2026-10-10 21:12:17	2026-10-10 21:12:17	0:00:00.0	Provisional credit issued	Activity Instance	noneEndEvent
2026-10-10 21:12:17	2026-10-10 21:12:17	0:00:00.0	Decision?	Activity Instance	exclusiveGateway
2026-10-10 21:12:17	2026-10-10 21:12:17	0:00:00.0		Detail	Variable
2026-10-10 21:12:16	2026-10-10 21:12:16	0:00:00.0		Detail	Variable

Every tool call is an activity instance
with start and end timestamps, in order, in the same history as everything else the engine runs.

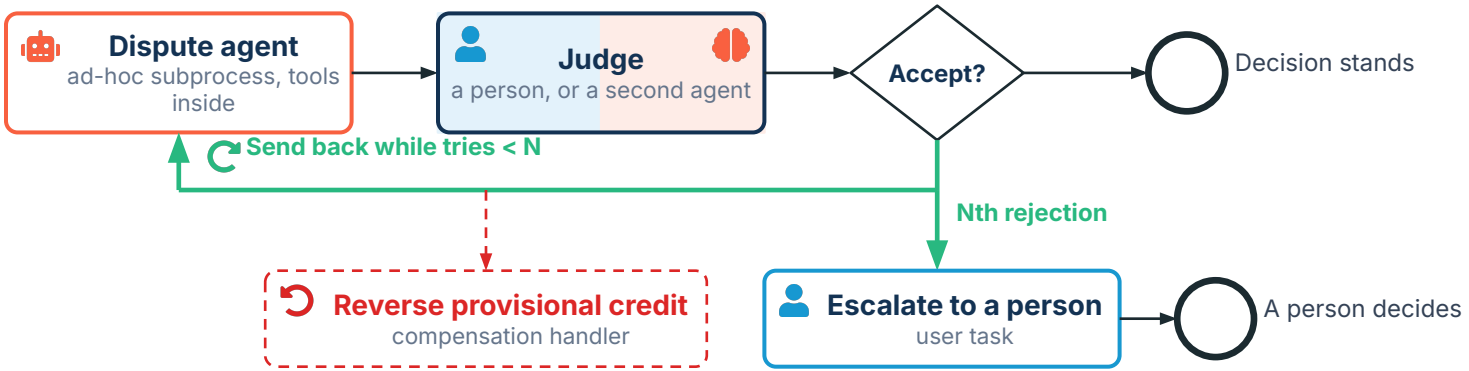
Every fact is a variable
customerProfile, transactionDetails, disputeDecision, decidedBy: agent or analyst.

The conversation is a variable too
_agentConversationHistory holds every model turn and tool call, queryable by the REST API.

Same tools your operations team already uses
Control Center, Monitoring, history REST API, database. No new audit system to build.

elevenai.co

Judging the agent: four patterns, one BPMN shape



Human as judge

The judge is a user task. The form limits the verdict to an enum, so the gateway can route on it. In the demo: escalate to analyst inside the loop, manual review after it.

LLM as judge

A second agent subprocess with one instruction, "does this decision follow policy?", and no side-effect tools. Separate prompt, separate version, cheaper model. Its verdict is just a variable.

Bounded reflection

The green loop: send back is a sequence flow and a counter. The model file decides how many tries, not the prompt, and the Nth rejection goes to a person. The engine records every attempt.

Compensation

If the judge rejects after the credit was already issued, a compensation handler reverses it. BPMN has had this since 2.0; agent frameworks make you write it by hand.

elevenai.co

Agentic patterns and what they are in BPMN

Pattern	In BPMN	In today's demo
Routing, classification	A DMN decision table and a gateway in front of the agent. Deterministic policy stays out of the prompt.	Triage eligibility gate
Tool use	Activities inside an ad-hoc subprocess. The tool list and descriptions come from the model file.	Eight tools
Human in the loop	A user task inside the agent's scope. The agent waits; the person answers through a form.	Ask the customer, escalate to analyst
Parallel evidence gathering	Several tool calls in one turn, or a parallel gateway inside a tool fragment.	DSP-1002 ran two tools at once
Orchestrator and workers	A call activity as a tool: the worker is its own agent process, with its own audit trail.	Not shown
Multi-agent handoff	Message events between process instances, correlated by business key.	Same mechanism as the withdrawal message
Memory	Process variables for the case; engine history for everything that ever happened.	agent:context, History tab
Budgets and termination	Timer boundary, completion condition, a cap on turns. The engine ends the loop, not the model.	30-minute timer
Guardrails on side effects	Hard limits in code or DMN. The tool refuses; the prompt only explains what to do next.	500 USD credit cap
Judge, reflection, compensation	Previous slide: user task or second agent, bounded send-back loop, compensation handler.	Analyst and manual review
Evals and regression	A scenario suite over the REST API asserting the end event and who decided.	Seven scenarios plus a withdrawal

elevenai.co

Where this goes next

Tool parameters

Today the model chooses tools but passes no arguments; tools read process variables. The next plugin version carries input mappings into the tool schema, so a tool can take what the model found.

The process as an MCP tool

Annotate the start event and an outside agent can file a dispute and get the decision back. Pattern 1 and pattern 2 in one process.

Join in

The agentic plugins are early, 1.0 snapshots, and the design is being worked out in the open. Join us and help shape the discussion.

github.com/finos/fluxnova-plugins · github.com/finos/fluxnova-examples

elevenai.co

Join the discussion

The work happens in the open, on the public issue tracker.

Feature requests, design discussions and bug reports for the agentic plugins live there, not on a slide. It is open source: the order of work is set by who shows up, so a comment or a pull request moves things.

github.com/finos/fluxnova-plugins/issues

Issues and pull requests welcome. The example you saw today is in finos/fluxnova-examples.

Scan to join the discussion

elevenai.co

Three things to take home

Discoverability, boundaries and audit trail are modelling problems. BPMN solved them before LLMs existed.

An ad-hoc subprocess gives a model room to act inside a scope the process controls and the engine records.

Fluxnova does this with plain BPMN and open plugins, hosted at FINOS. Come and build on it.

fluxnova.finos.org · github.com/finos/fluxnova-examples · Questions?



Thank You

Go to ELEVEN.

Chris Miller · 11ai.co

fluxnova.finos.org · github.com/finos/fluxnova-examples

Enterprise workflow orchestration. Open. Independent. Proven.