



Beyond Agentforce

How Salesforce professionals should think about
agentic systems

Shivanath Devinarayanan

Chief Digital Labor and Technology Officer, Asymbi

[Salesforce MVP Hall of Fame](#) | Founder and former CEO, Dazeworks



Three **movements** for the next 15 minutes

A customer says “We want Agentforce.” Here is how I turn that into a system-design conversation.

1

Reframe the boundary

The Salesforce org stops being the whole application. It becomes one critical node in a larger work system.

2

Design the system

Five planes, the right execution pattern per step, and the difference between RAG, working context and durable memory.

3

Decide what to build next

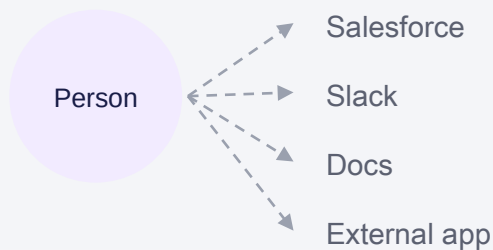
Leave with a repeatable first-project method you can run on Monday: one bounded workflow, designed and tested the same way every time.

Reframe, then design, then walk out with a first project you can start on Monday.

The org is no longer the **application**

OLD MODEL

People navigate systems



NEW MODEL

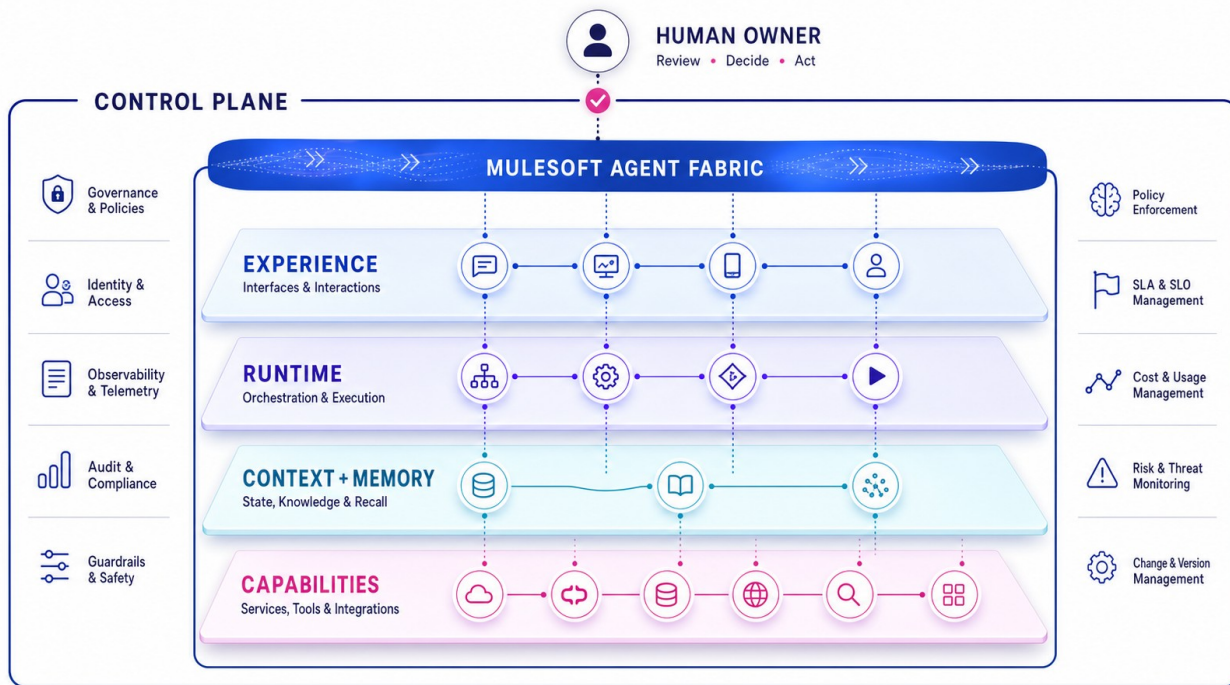
Work moves across agents, tools and people



“I built a Salesforce company around the org. Today, the org is one node in the system.”

The org stays important. It becomes one node in a larger work system, not the boundary.

The agentic system has **five planes**



Experience, runtime, context and memory, and capabilities, all bounded by one control plane, with a human owner above it.

Follow the **work**, not the product boundary

One escalation, seven steps, one accountable outcome.

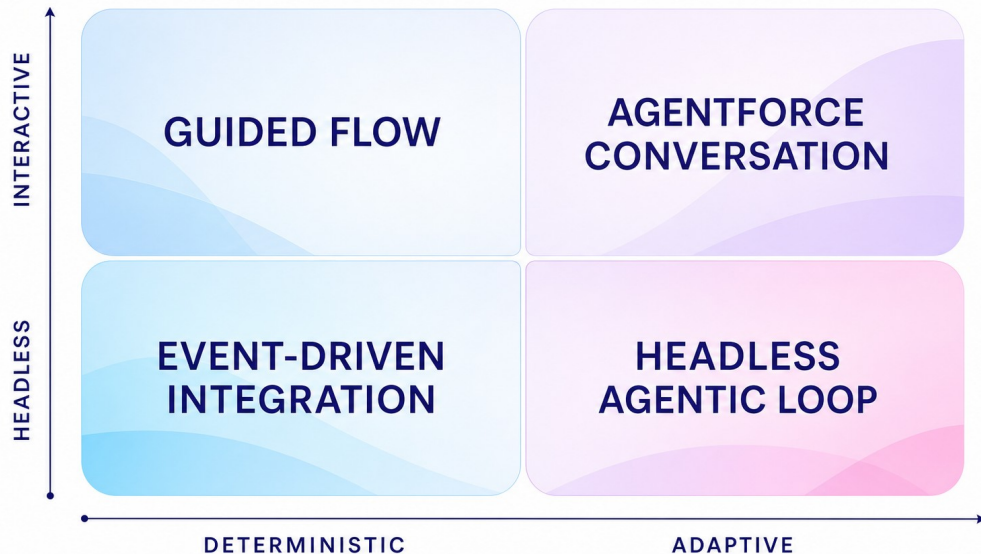
Same pattern: lead qualification · case resolution · employee onboarding · candidate screening · quote approval



Use the right execution pattern at each step. Human approval at step 6 keeps one owner accountable.

Choose the **execution pattern** before the product

Deterministic or adaptive is the first question. Interactive or headless is the second. Products come last.



IMPLEMENTATION OPTIONS

Flow

Apex

Agentforce

MuleSoft

Claude Agent SDK

Options, not fixed positions.

RAG is **not** memory



What does the source say?

What matters right now?

What happened or was decided?

⚠ A vector database is an implementation choice, not a memory strategy.

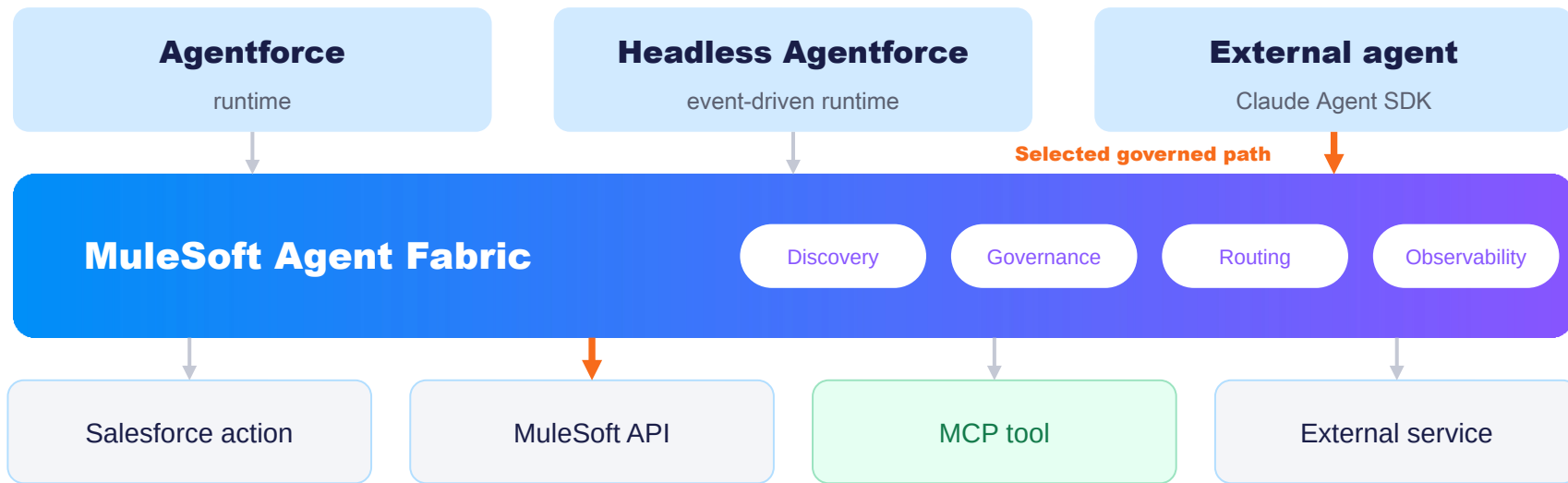
Govern the network. Agent Fabric is one

Illustrated implementation: **MuleSoft Agent Fabric**. The control-plane role is the constant; the product shown is one way to fill it.

● **Agentforce** = runtime

● **MCP** = tool protocol

● **MuleSoft Agent Fabric** = control plane



The agent is rarely the scary part. The scary part is a write with unclear authority and no reversal path.

Agent Fabric is one implementation. The design obligations remain without it: **identity, policy, routing, evidence, observation, rollback.**

What you can do on Monday



Six steps to take one bounded workflow from idea to a safe, testable first project.

1

Choose one bounded workflow

Pick a single job with a clear start and end.

2

Define the outcome and owner

Name the result and the one person accountable.

3

Map the full work path

Trace every system and hand-off the job touches.

4

Classify each step

Deterministic or adaptive, interactive or headless.

5

Mark the risk points

Writes, identity, evidence, approval, reversal.

6

Test in shadow mode

Watch it run before it acts, then allow controlled action.

The shared destination: **Agentic systems designer** · Judgment · Orchestration · Validation · Ownership



Let's keep **thinking** together

I share ongoing thoughts and operating lessons on building and running agentic systems on Substack.

Shivanath Devinarayanan

Chief Digital Labor and Technology Officer, Asymbl



[Scan to follow](#)