



Technical Deepdive into Flowise & Agents





Topic

Introduction & Context – From Strategy to Architecture

Workday's Agent Ecosystem Overview – Workday Agent Gateway, ASOR, and AI Agents

Building with Flowise – how it compares and a live build

HR Agent Reference Architecture Deep Dive – Layers, Data Flow, and Security

Flowise Patterns & Use Cases – Applying the Architecture (with Demos)

Wrap-Up & Next Steps

Q&A

REC

LIVE DEMOS



Any unreleased services, features, functionality or enhancements referenced in any Workday document, roadmap, blog, our website, press release or public statement that are not currently available are subject to change at Workday's discretion and may not be delivered as planned or at all.

Customers who purchase Workday services should make their purchase decisions based upon services, features, and functions that are currently available.



Rick Leunisse
Managing Partner



Matt Komendolowicz
Managing Partner



The Innovation Partner for Workday: Incubane



Customers

20+ Customers

- Incubane built apps for 20+ Customers and 500,000+ workers in 100+ countries
- Largest Customer: 150,000+ workers
- Excellent references



Project Expertise

200+ Projects experience

- Founders involved in Extend since 2017
- Founders who were involved in 200+ Extend projects
- 20+ Extend projects as Incubane in the first year of company



Innovation Champions

Agents, Flowise, ASOR

- First partner developing Flowise Agents (webinar for 2,000 people)
- 5 apps on marketplace
- Apps enhanced with your own custom LLM

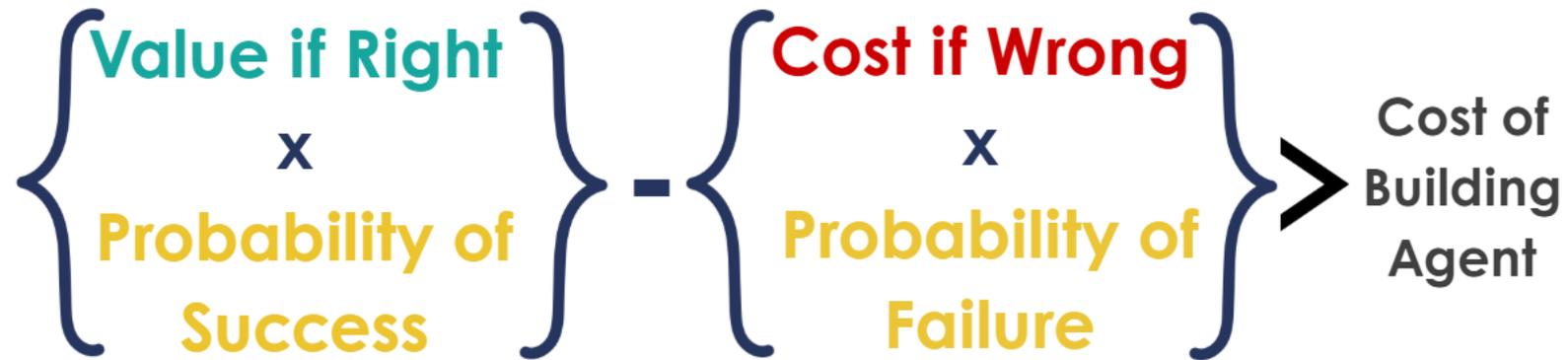


Free resources

LinkedIn & Community

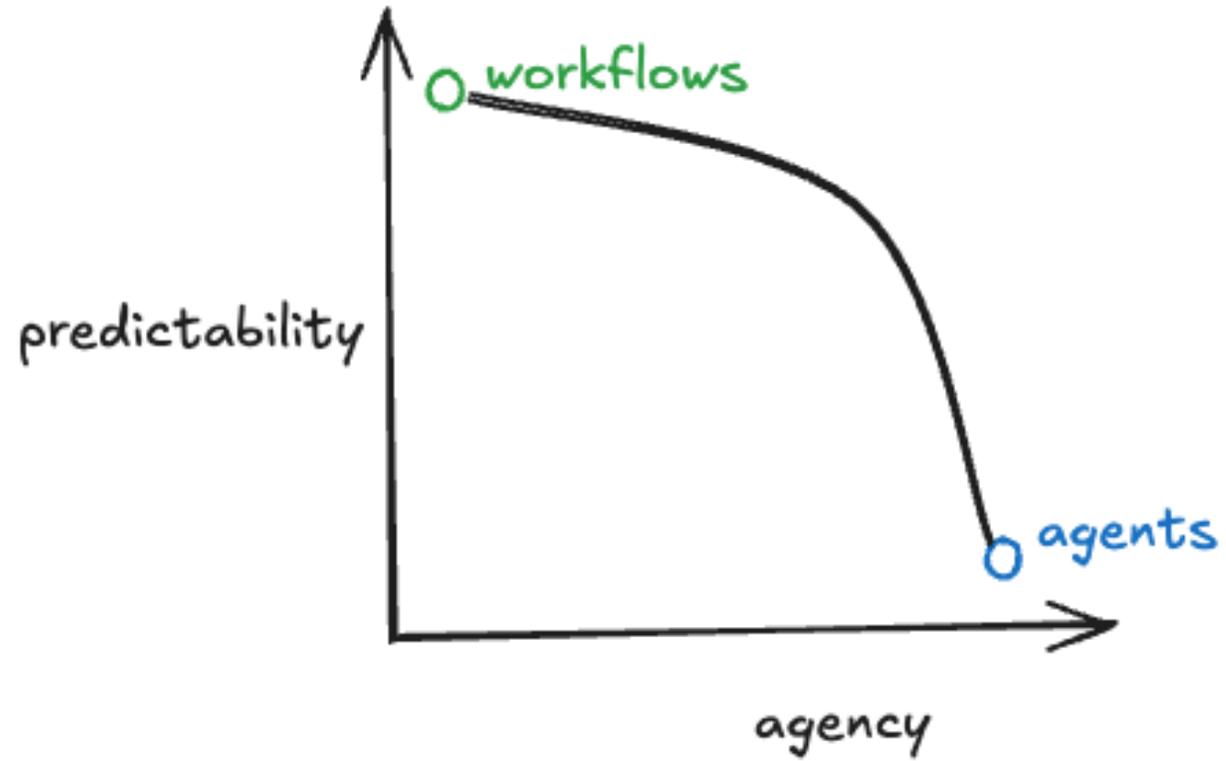
- Free resources like Extend Playbook, Custom Rising GPT, Rising Summary, DevCon summary
- 1,000,000+ people reached
- 20,000+ people following

Designing Agents That Actually Deliver



We work with HR and IT leaders to help you build agents that provide real enterprise value

Agents vs. Workflows



How you can make this formula work for the enterprise

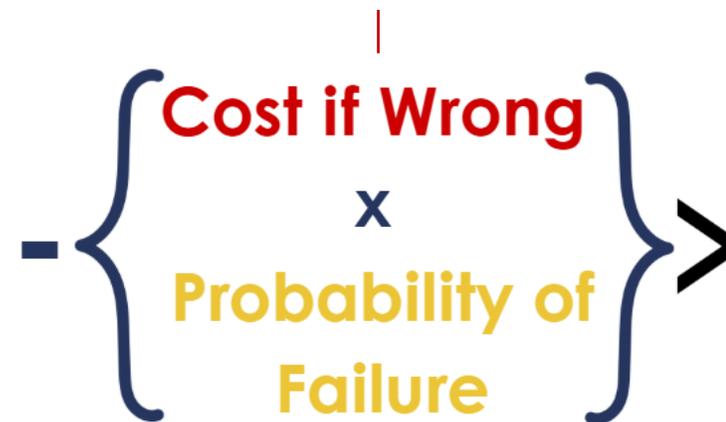


- Have agents focus on high value tasks
- Use multi-agent to improve quality & quantity of outputs



- Make parts of your agent more deterministic
- Observability and Evaluations

- Make it easy to reverse
- Distinguish between perceived and actual costs



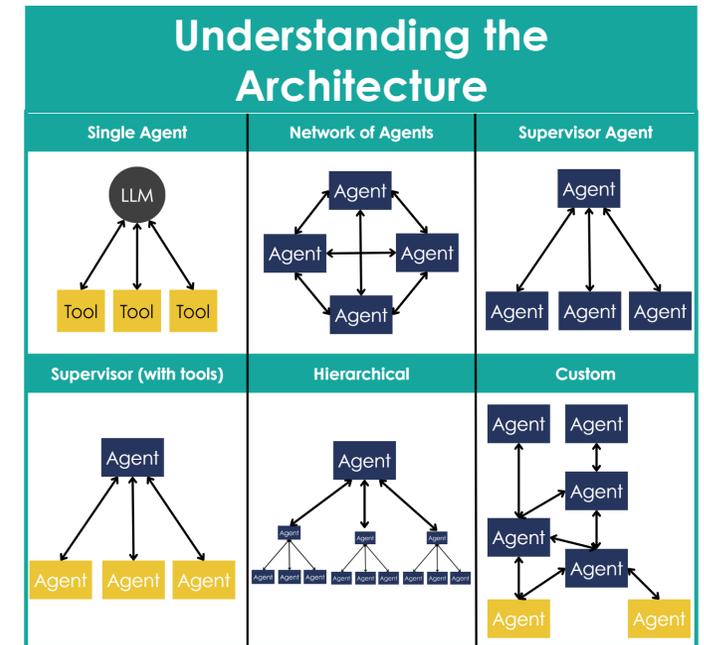
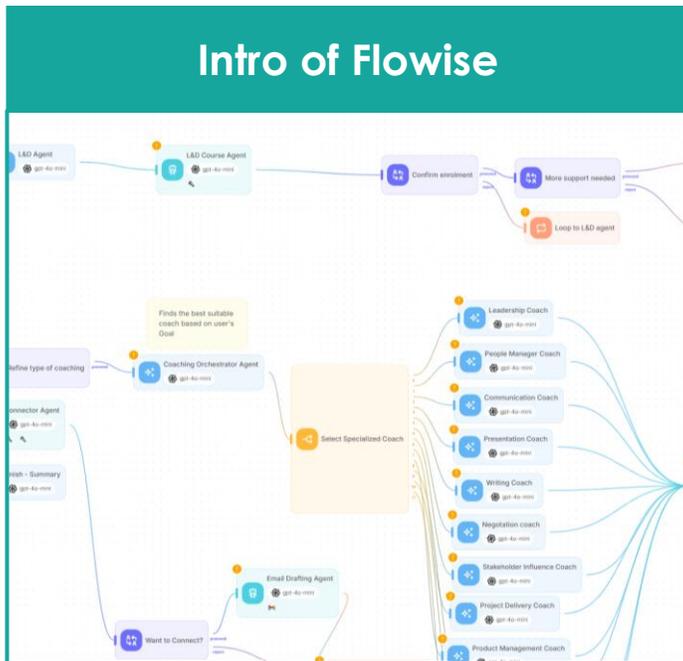
- Human-in-the-loop

Cost of Building Agent

From Strategy to Architecture



In the last two months we introduced you to Flowise, explained our A.G.E.N.T. strategy framework and today we will dive into the technology



Workday's Agent Ecosystem Overview



Workday's Agent Ecosystem

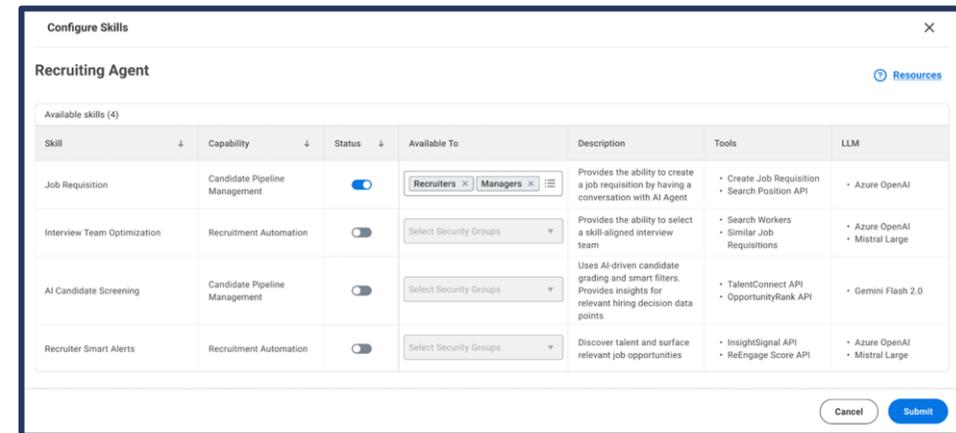
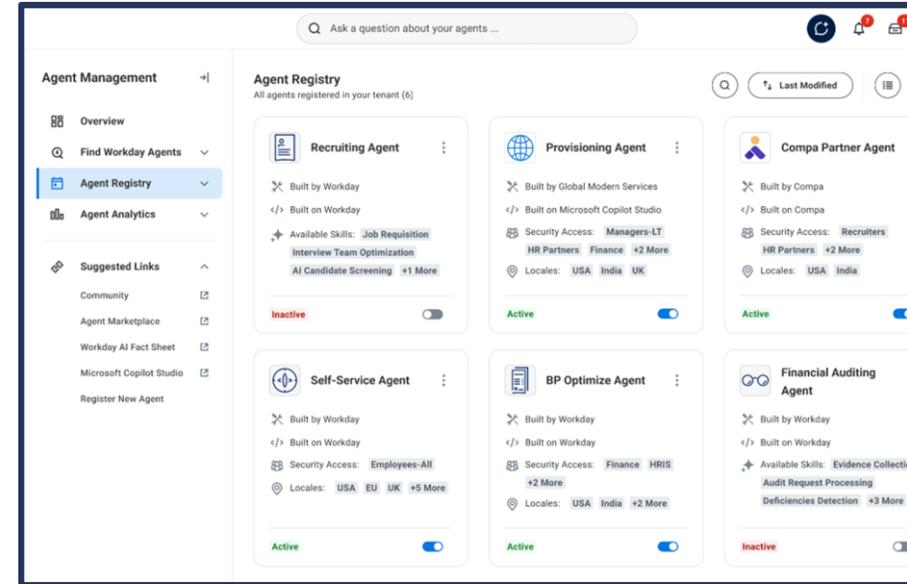


What is the Agent System of Record?



Agent System of Record (ASOR): Workday's central source of truth and control plane for enterprise AI agents. It catalogs all agents, including Workday-built, partner, and customer-built, manages their lifecycle and skills.

- **Management:** Source of truth for your enterprise agents
- **Security:** Agent identity, authentication, authorization permissions
- **Analytics:** Full observability of metrics, usage and value
- **Interoperability:** Multi-system and agent interactions with Agent Gateway



ASOR will offer multi-level analytics



Human & Machine Collaboration

This dashboard shows a central profile for Robert Chang, a VP of Talent Acquisition. It features a list of other agents like Joanna Casey Smith and Sokratis Papastathopoulos. A 'Recruiting Agent' card highlights its performance, showing an average time to fill role of 21 days compared to a historical average of 32 days. A sidebar lists various agent categories such as Recruiting Operations, Business Talent Partners, and Talent Acquisition Agents.

Agent Value

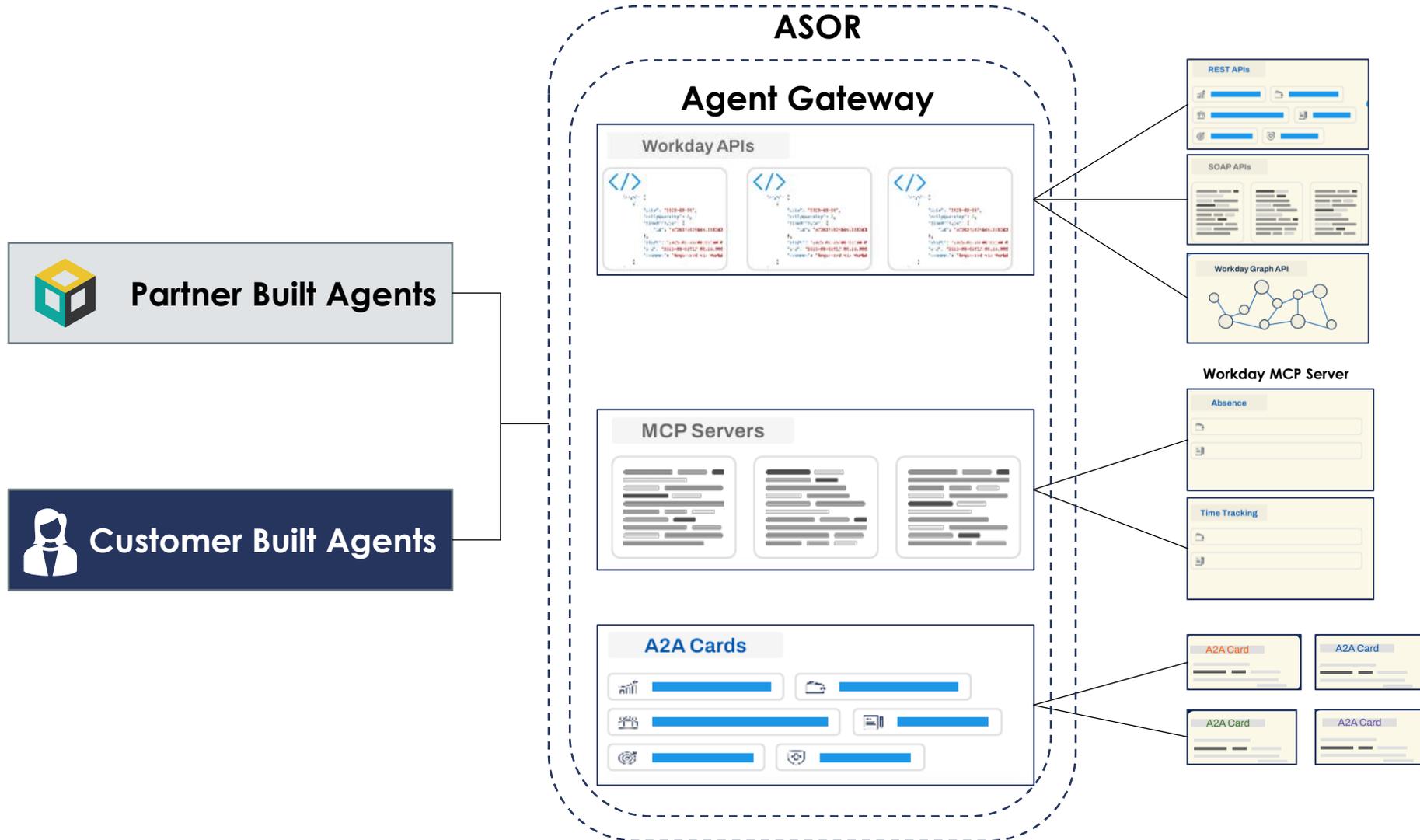
This dashboard provides a comprehensive overview of agent usage and performance. It includes a line chart for 'Agent Usage' over the last 90 days, showing metrics for various agent types like Self-Service Agent, Recruiting Agent, and Talent Mobility Agent. Key highlights include an agent engagement of 12,687 (up 14.2% since last week) and 18 active agents. A 'Most Used Agents' section lists top performers like the Recruiting Agent and Self-Service Agent.

Runtime & Operational

This dashboard focuses on the operational aspects of agent sessions. It displays key metrics: 12,583 total sessions, a 98.5% success rate, 12,395 complete sessions, and 342 unique users. A table below provides a detailed log of agent sessions, including session ID, start time, duration, status, user, execution type, and skills used.

Session ID	Start Time (PDT)	Duration	Status	User	Execution Type	Skill(s) Used	Audit Trail
SID-84561351	2025-05-12 10:31:15	3000 ms	Complete	Maria Garcia	Delegation	Job Requisition Creation	View
SID-68746851	2025-05-12 10:25:05	1850 ms	Complete	John Chen	Ambient	Interview Feedback Summarization	View
SID-68413539	2025-05-12 10:32:01	650 ms	Incomplete	Maria Garcia	Delegation	Job Requisition Creation	View

Introducing Workday's Agent Gateway



Building with Flowise



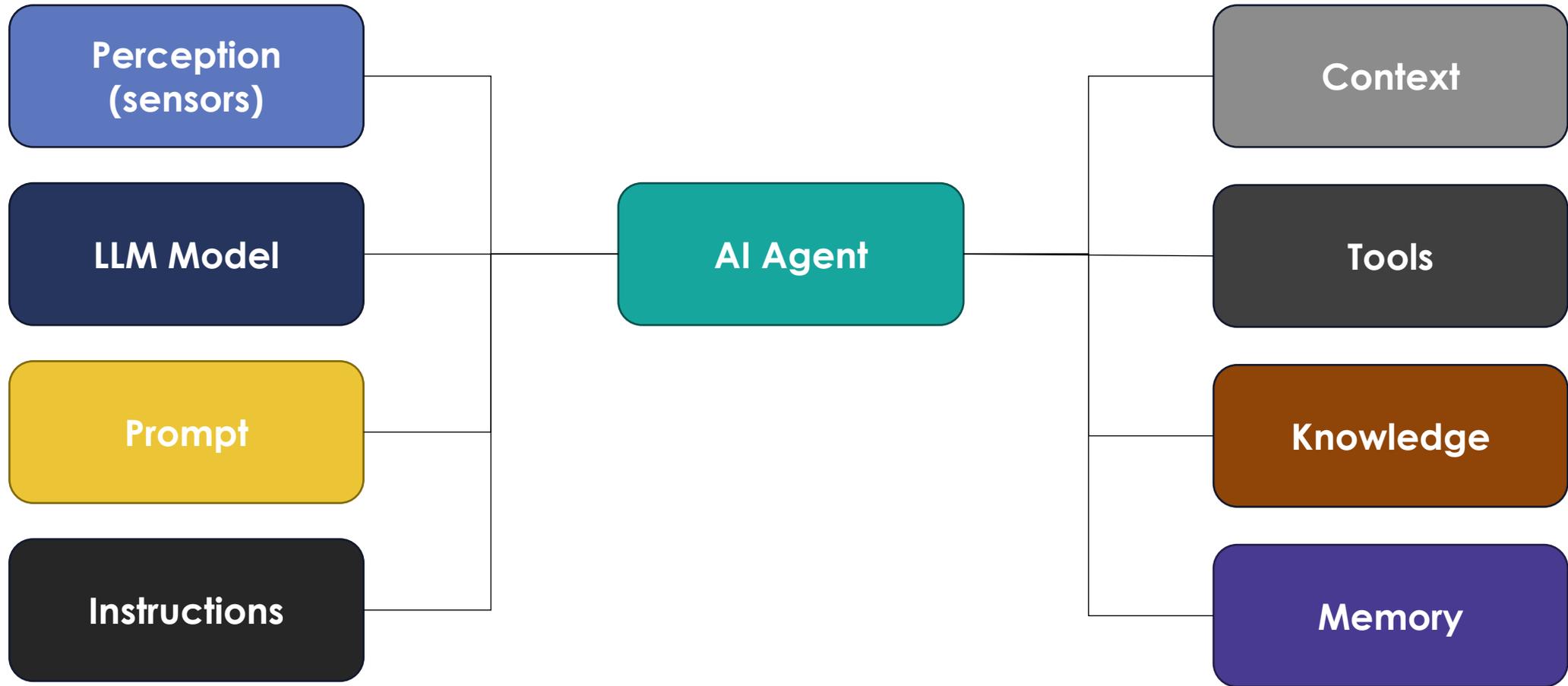
What Is An Agent?



An agent is an AI system powered by an LLM that can decide what to do next and call tools or APIs to get things done, instead of only replying with text.



Components of an Agent



What is Flowise?



Build Agents Visually

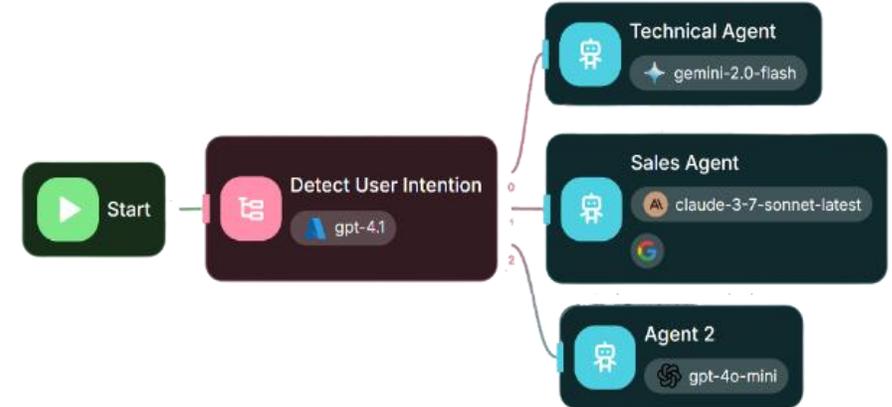
Build various AI workflows and agents from simple conversational to complex orchestrations all via an intuitive visual builder.

Rich AI Ecosystem

Connect with 150+ data sources, tools, LLMs and MCP to build deeply integrated AI workflows and agents.

API, SDK, Widget

Support API, SDK and embedded chat widget to extend and integrate custom-build agents into existing systems.

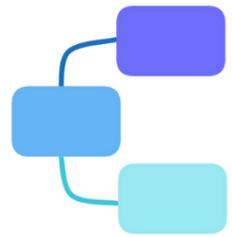


Flowise will be part of the Extend Pro SKU and is expected to become GA in 2026 H1

How does Flowise compare to other tooling?



Criteria	 Flowise	 n8n	 make	 zapier
Main Use	LLM agents, chatbots	General automation	Cloud workflows	Simple SaaS zaps
Visual Builder	Drag/drop, AI-focused	Node-based, flexible but complex	Flowchart-style, user-friendly	Wizard-style, very simple
Ease of Use	Built for AI, no-code friendly	Moderate, scripting for depth	Easy templates	Very easy, basic use
Logic & Loops	Full support incl. loops	Advanced branching	Filters, iterators	Basic branches only
Memory/State	Built-in (chat + vector DBs)	Partial (no persistent memory)	Partial (scenario-level)	Single-pass only
Custom Logic	JS/Python nodes, open plugins	High – JS/Python functions	HTTP/API, limited scripting	Minimal, closed platform
Integrations	Growing (~100+, mostly AI)	Wide (~400+)	Very wide (~3,000)	Huge (~8,000+)
AI Capabilities	Native, built for agents & LLMs	Available, extensible	Basic LLM support only	Basic, no memory/tools
Hosting & License	Open-source + SaaS, can also self host	Fair-code, self-hosted and SaaS	Closed SaaS only	Closed SaaS only
Maturity	New, fast-growing (AI niche)	Mature, strong OSS base	Mature, wide use	Long-standing, very stable



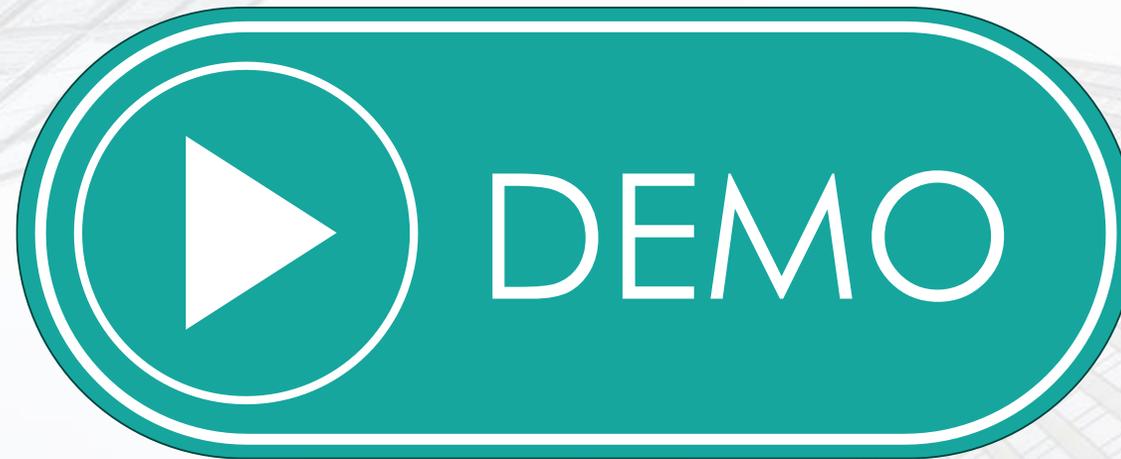
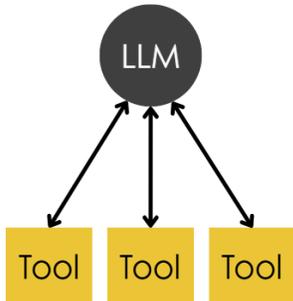
Flowise





Live Build of Course Advisor Agent

Single Agent



HR Agent Reference Architecture



Example of a HR Agent Reference Architecture



Experience & Entry Points

Agent Layer

Primary HR Platforms



What is Agent2Agent (A2A) and Model Context Protocol (MCP)

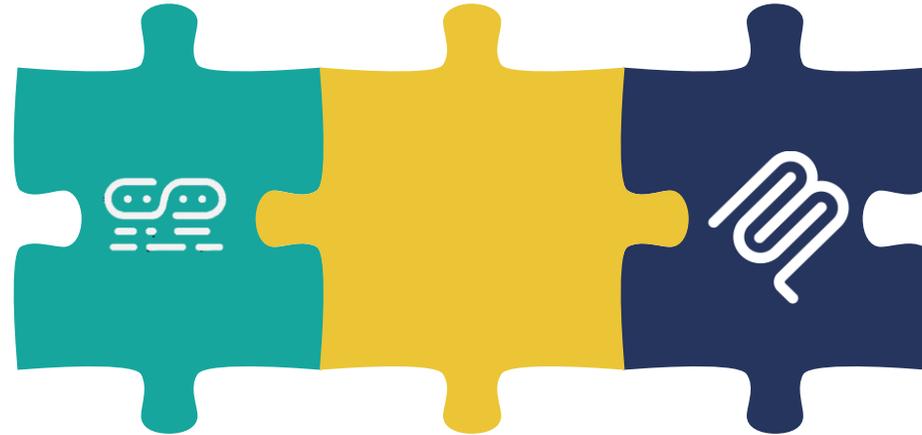


A2A: agent to agent collaboration.

An open standard that enables agents, regardless of their underlying frameworks or vendors, to securely communicate and collaborate on complex task

Focused On

- Discovery (which agents exist and what they can do).
- Capability advertising (what skills an agent exposes).
- Sending tasks and getting back results.
- Secure, auditable communication between agents from different vendors.



In Workday Context

- An HR Super Agent that cooperate with:
- Workday's Self Service Agent for simple employee questions.
 - The Grow Agent from Incubane for development plans.
 - The Galileo Agent from Josh Bersin for learning recommendations.

MCP: agent to tools and systems.

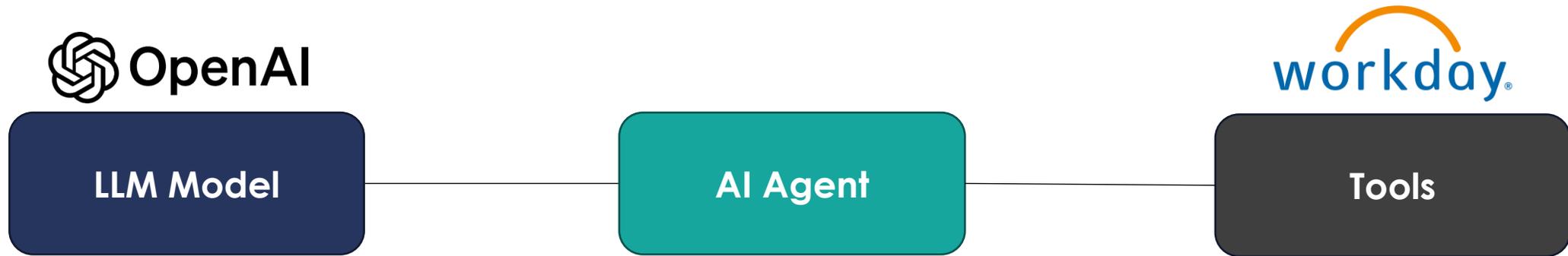
An open standard for connecting LLMs to external data, tools, and systems, acting like a "USB-C for AI". It enables AI models to access real-time info, perform actions

Focused On

- Standard way to expose tools and data from HR systems like Workday etc.
- Clear schemas for inputs and outputs, so agents know how to call each tool.
- How tools are discovered and configured for different agents.
- Shared rules for authentication, permissions, and secret handling.

In Workday Context

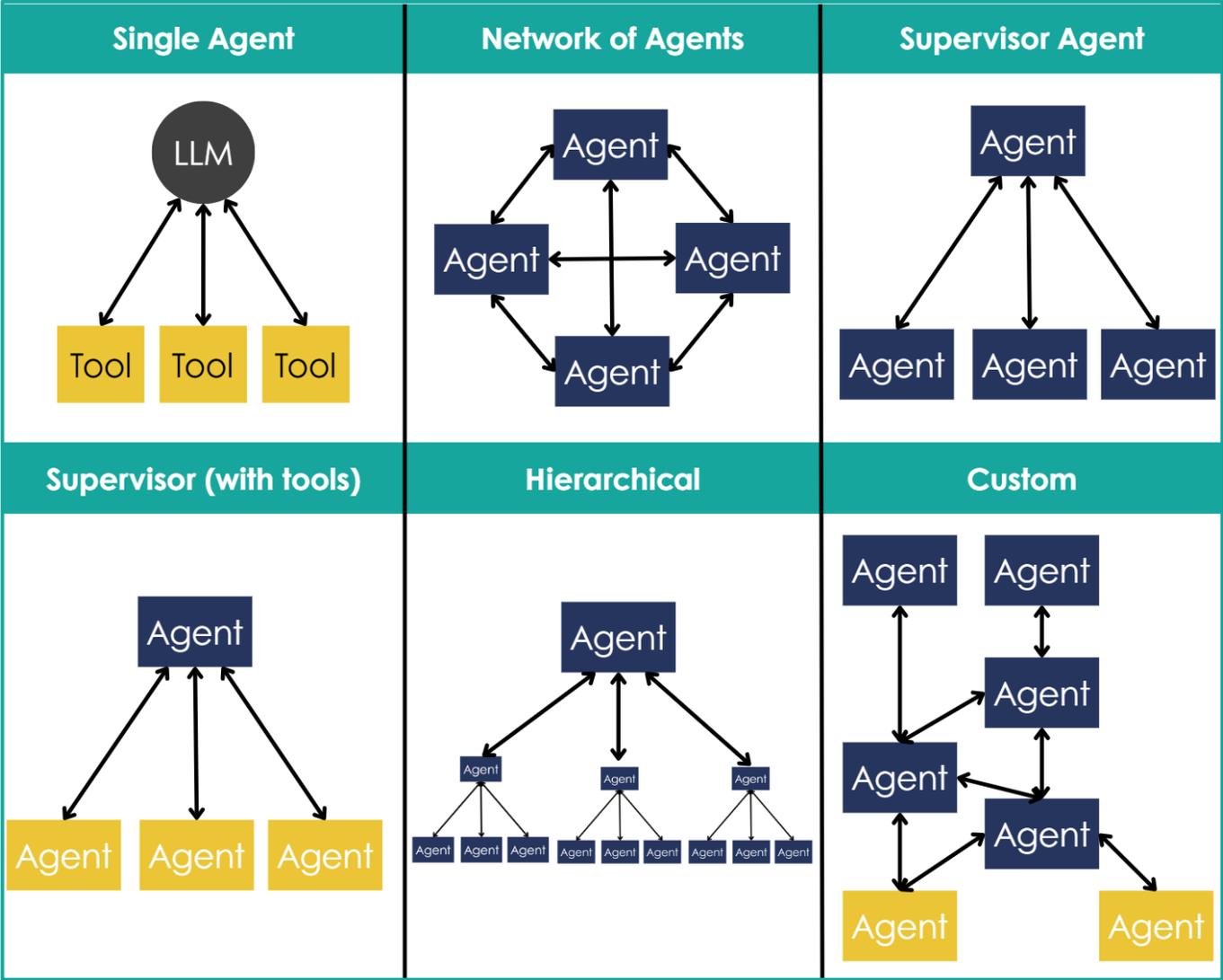
- Call Workday MCP servers that expose:
- Get employee details
 - Create a learning assignment in Learning.
 - Get open tickets from ServiceNow.
- Let your agents (in Flowise, or custom HR Agents) call these MCP tools in a consistent way.



Problem with single agents

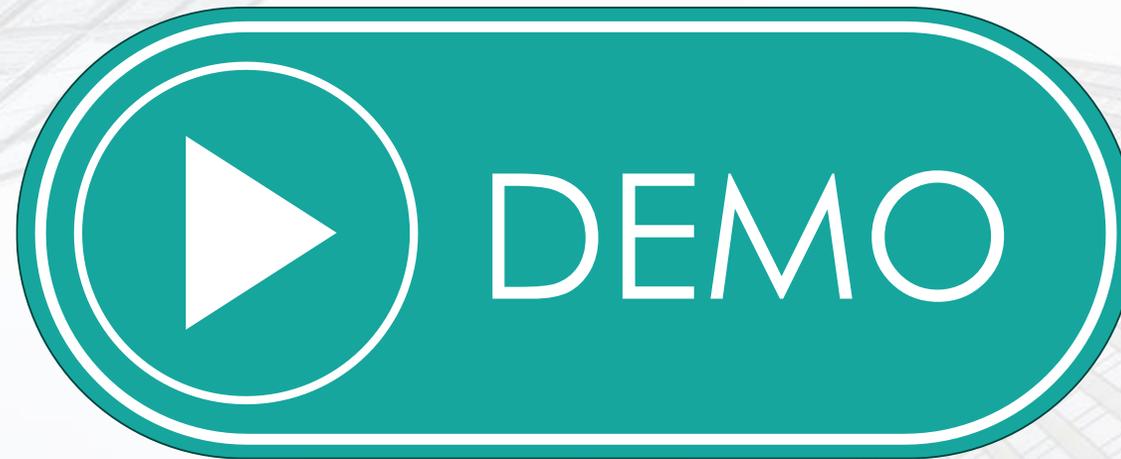
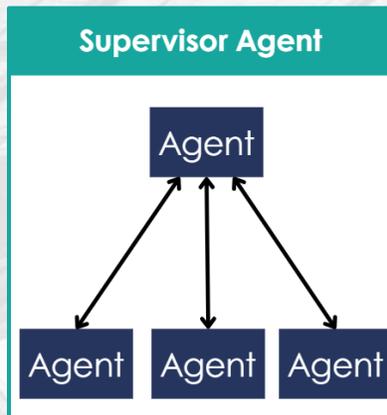
- Agent will have too many tools at it's disposal
- Context will grow too complex
- You cannot have specialization of an agent

Different Examples of Multi Agent Architectures



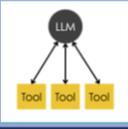
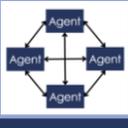
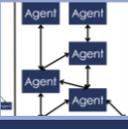


Onboarding Agent



The Pros and Cons of Different Multi Agent Architectures



Agent Pattern	Short description	When it fits best	Main pros	Main cons
Single Agent 	One LLM that talks to the user and calls tools directly.	Simple use cases, single domain, clear tasks like Q&A or simple workflows.	<ul style="list-style-type: none"> • Easiest to design and run • Low infra cost • Few failure modes • Easy to debug and monitor 	<ul style="list-style-type: none"> • Limited specialization • Context window becomes a bottleneck for bigger tasks • Harder to enforce strict policies and separation of concerns • Single point of failure
Network of Agents 	Many agents that can talk to each other in a peer-to-peer way.	Exploratory or R&D setups, creative problem solving, situations where you want many perspectives.	<ul style="list-style-type: none"> • Flexible, agents can self coordinate • Good for exploring different options in parallel • Easy to add or remove agents without redesigning the whole system 	<ul style="list-style-type: none"> • Hard to predict flows and outcomes • Debugging conversations between many agents is difficult • Governance and audit are complex • Risk of loops or "chatter" between agents that drives up cost
Supervisor Agent 	One top agent routes work to a small group of worker agents and merges the results.	Clear "manager with specialists" setup, for example planner plus domain experts.	<ul style="list-style-type: none"> • Single place for orchestration and policy • Workers can stay simple and specialized • Easier logging and monitoring through the supervisor • Good match with business team structures 	<ul style="list-style-type: none"> • Supervisor becomes a bottleneck and single failure point • If planning by the supervisor is poor, the whole system suffers • More latency than a single agent • Needs careful design of interfaces between supervisor and workers
Supervisor (with tools) 	One main agent that calls tool like components, which themselves are simple or "agent like" modules.	When most work is deterministic tooling, with a bit of LLM reasoning on top, for example data lookups, workflow triggers, APIs.	<ul style="list-style-type: none"> • Keeps logic mainly in tools, so behavior is more stable • Easier to test tools in isolation • Good control over data access through tools • Often cheaper and faster than many full agents 	<ul style="list-style-type: none"> • Less flexible than full worker agents • Harder to handle open ended tasks only with tools • Main agent still must know which tool to use when • Can grow into a large tool list that is hard to manage
Hierarchical 	Multiple levels of agents, top level breaks work into parts, mid level manages sub areas, lower level executes tasks.	Large or complex tasks that naturally split into projects, subprojects and tasks, for example big analyses or multi step workflows.	<ul style="list-style-type: none"> • Scales well to complex goals • Strong specialization at each level • Mirrors real world org charts so mapping to teams is natural • You can swap parts of the tree without breaking everything 	<ul style="list-style-type: none"> • Design is more complex and time consuming • Many points of failure and handoffs • Latency and token use go up quickly • Needs strong shared memory or state, otherwise context is lost between levels
Custom 	Any mix of the patterns above, tuned to a specific product or domain.	Mature systems that have learned what works, or products with special needs or strict constraints.	<ul style="list-style-type: none"> • Maximum flexibility • Can be tailored to your exact data, tools and risk profile • Lets you reuse the good parts of other patterns and drop the rest 	<ul style="list-style-type: none"> • Highest design and maintenance effort • Harder to explain and document for stakeholders • Testing, monitoring and safety can be complex • Risk of "architecture by accident" if changes are not controlled

Flowise Patterns & Use Cases





Concept: Chaining runs agent steps sequentially, where the output of one step becomes the input for the next.

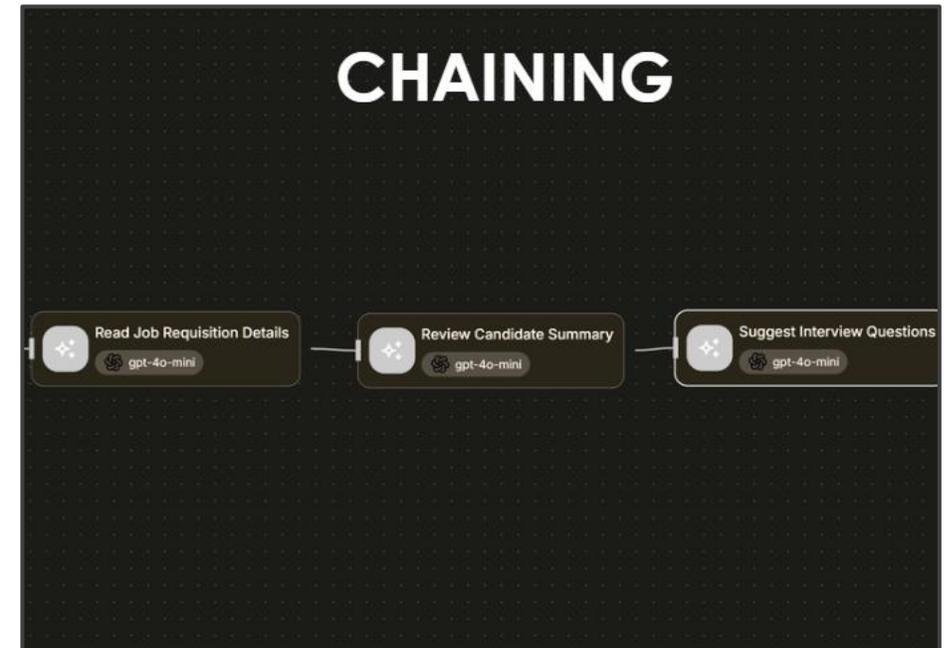
Benefits

- ✓ Easy to design and debug
- ✓ Transparent logic for HR/IT teams
- ✓ Ideal for use cases with linear reasoning or tasks that depend on earlier outputs

Workday Example: HR – Interview Prep Assistant

An agent helps a recruiter get ready for interviews:

1. Reads the job requisition from Workday
2. Summarizes the candidate profile
3. Generates tailored interview questions



Flowise Pattern: Looping



Concept: Looping allows an agent to revisit earlier steps until a condition is met. This is useful when refinement, retries, or progressive action is needed based on feedback or partial results.

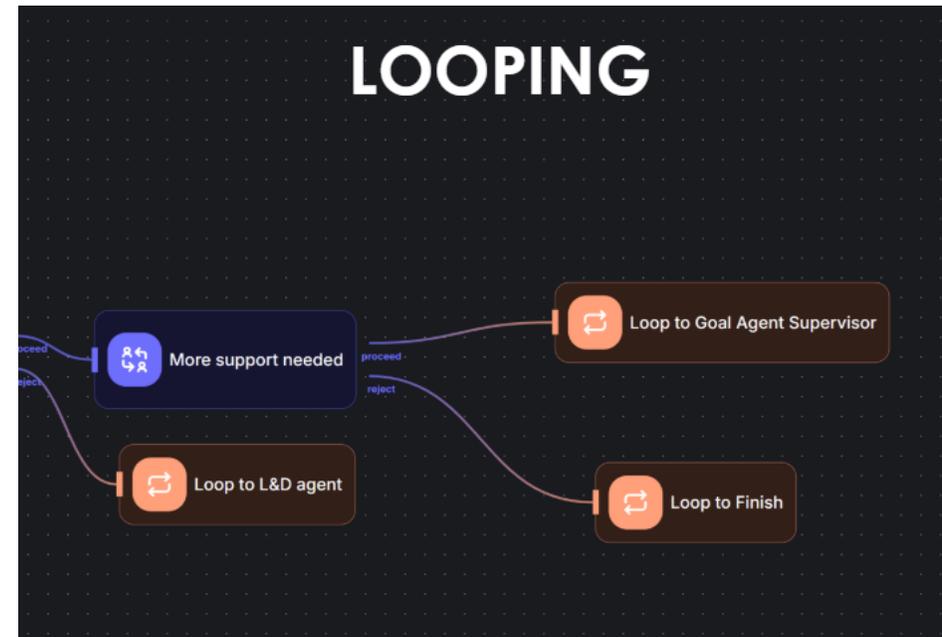
Benefits

- ✓ Enables iterative workflows (e.g. reviews, improvements, approvals)
- ✓ Supports more human-like reasoning cycles
- ✓ Helps avoid dead-ends in agent logic

Workday Example: Manager Coach

An agent helps a manager coach an underperforming employee:

1. Evaluates progress against a goal
2. If improvement is lacking, loops through suggested learning actions via L&D agent
3. Supervisor agent reviews and decides whether to continue support or escalate



Flowise Pattern: Routing



Concept: Routing uses a central logic node or agent to decide which path or agent to trigger next, based on dynamic input or context.

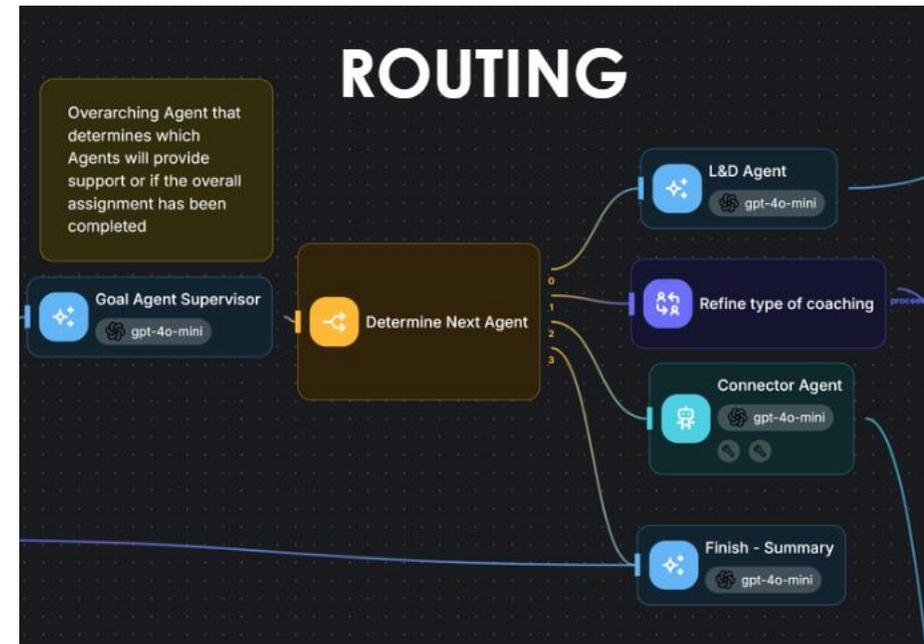
Benefits

- ✓ Enables adaptive, context-driven workflows
- ✓ Scales well across multiple agent types
- ✓ Reduces unnecessary steps by routing only what's needed

Workday Example: GROW Agent

A performance agent evaluates an employee's progress:

1. If coaching is needed, routes to an L&D agent
2. If goals are off track, routes to a Goal Supervisor agent
3. If support is complete, routes to a Summary agent



Flowise Pattern: Parallel



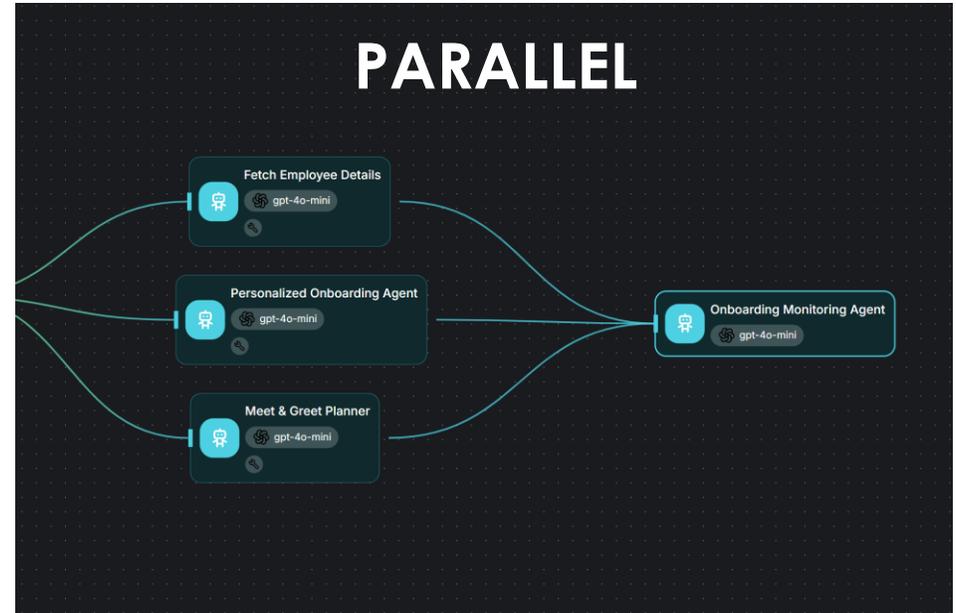
Concept: Parallel patterns run multiple agent tasks at the same time. These branches execute independently, then either return their own results or converge back into a single flow.

Benefits

- ✓ Speeds up workflows by avoiding sequential delays
- ✓ Enables agents to specialize (e.g. one reads, one writes, one calculates)
- ✓ Reduces user wait time in multi-part processes

Workday Example: Onboarding Assistant Agent

1. Agent A fetches employee details from Workday
2. Agent B generates a personalized onboarding checklist
3. Agent C sends intro messages or links via Teams/Slack



Flowise Pattern: Evaluator-Optimizer



Concept: This pattern separates the agent that does the work (**optimizer**) from the one that checks the quality or correctness (**evaluator**).

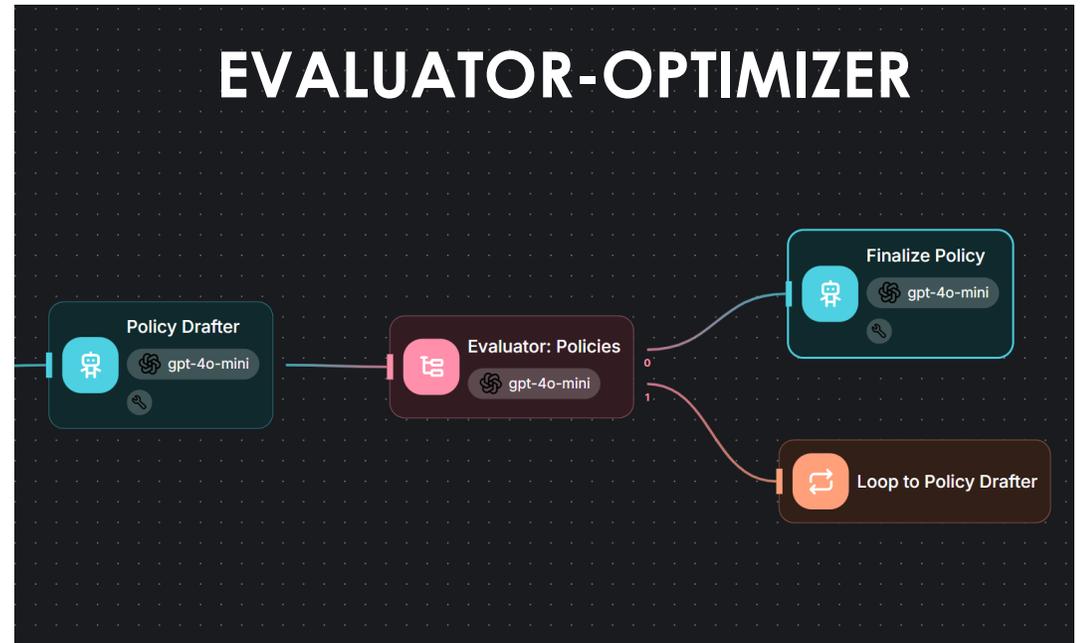
Benefits

- ✓ Raises reliability and trust in agent output
- ✓ Supports built-in validation and compliance logic
- ✓ Enables human-like self-review cycles before actions are taken

Workday Example: HR – Policy Drafting Agent

An HR agent drafts a new policy for remote work:

1. The **optimizer agent** writes the first draft based on templates and org inputs
2. A separate **evaluator agent** checks tone, legal compliance, and coverage gaps
3. If issues are found, feedback is sent back for redrafting
4. Once approved, the policy is routed for human review and publishing





HR Policy Writer Agent



DEMO

Flowise Pattern: Iteration



Concept: Iteration enables the agent to repeat a specific action multiple times until a desired output quality or goal is reached.

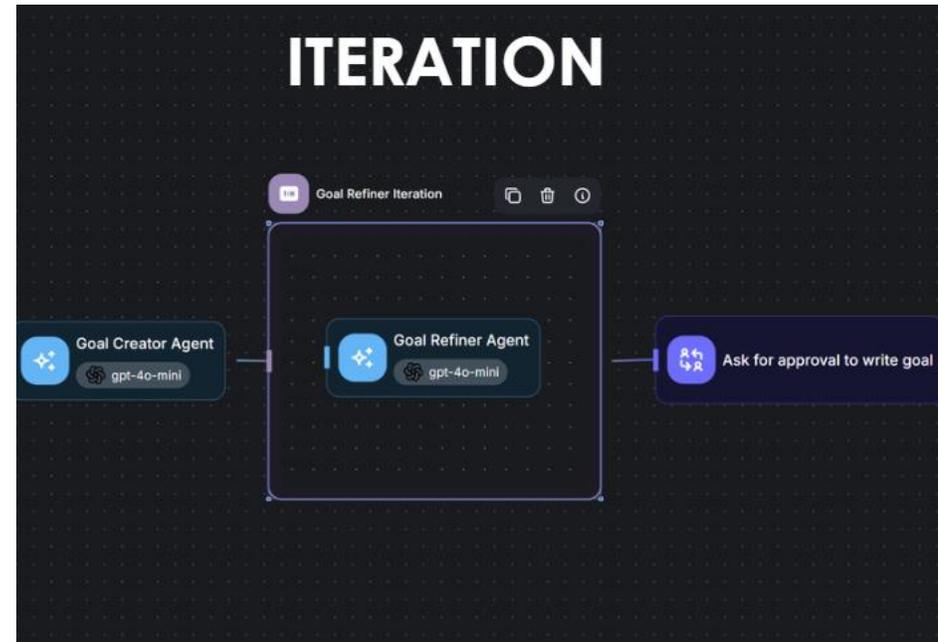
Benefits

- ✓ Supports refinement, drafts, rephrasing, or incremental progress
- ✓ Ideal for co-creation tasks with humans in the loop
- ✓ Boosts quality of generated output over time

Workday Example: GROW Agent

An agent helps an employee draft a goal:

1. Agent proposes a first version
2. User provides input like “make it more measurable” or “align with team goals”
3. Agent refines the goal wording in multiple iterations
4. Once confirmed, it submits in Workday



Flowise Pattern: Human-in-the-loop



Concept: This pattern inserts a human checkpoint at one or more points in the agent workflow. The agent pauses, waits for human input or approval, and only continues once confirmation is received.

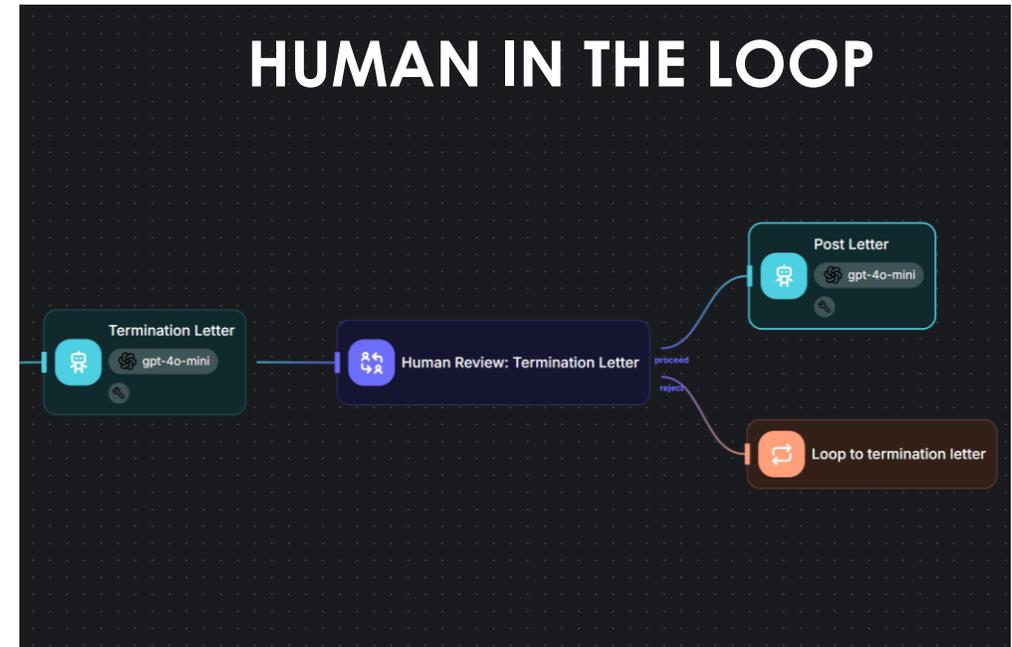
Benefits

- ✓ Adds oversight to sensitive or high-impact decisions
- ✓ Builds trust with stakeholders (HR, Legal, Finance)
- ✓ Ensures alignment with compliance, policy, and ethics

Workday Example: Termination Notification Agent

An HRBP asks the agent to generate a termination letter:

1. Agent gathers employee details and policy references from Workday
2. It drafts a letter using a language model
3. Before sending, it pauses and routes to the HRBP for review
4. Only after manual approval is the letter sent



Flowise Pattern: Orchestrator-Worker



Concept: Orchestrator dynamically creates or assigns tasks to goal-oriented worker agents based on the overall objective. Unlike classic Routing, the workers aren't predefined, they are instantiated or instructed on-the-fly.

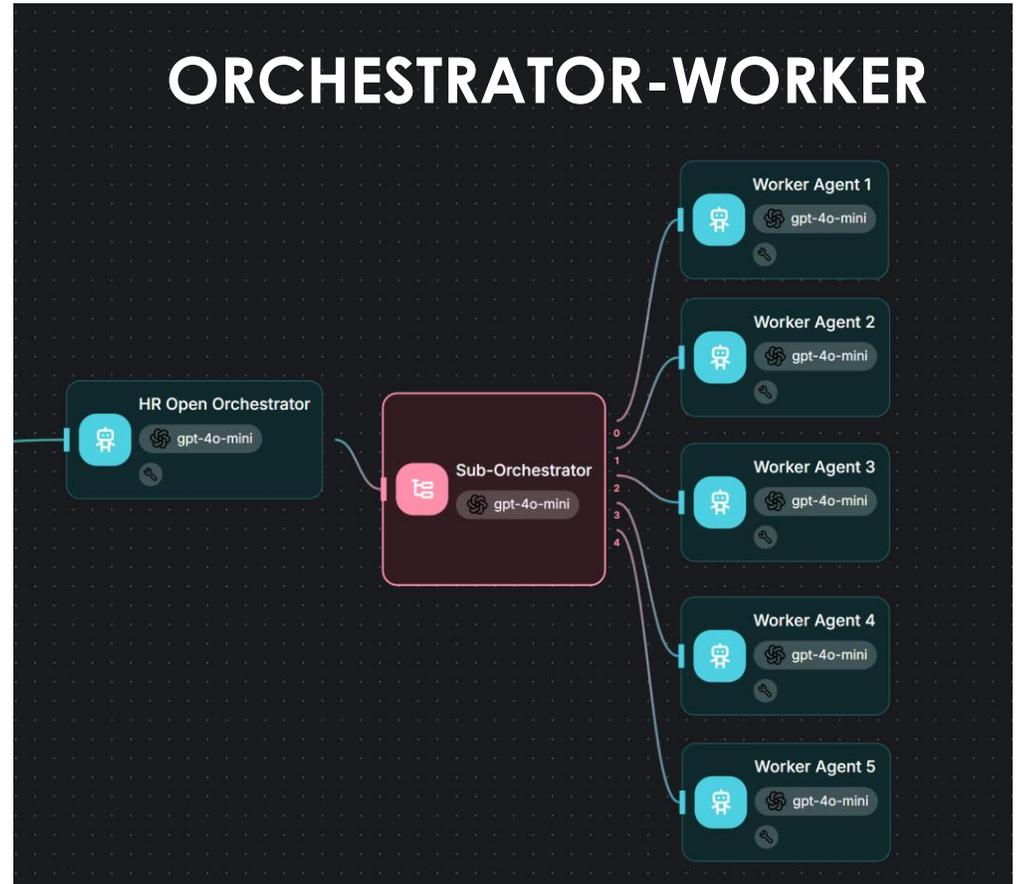
Benefits

- ✓ Enables adaptive workflows for open-ended objectives
- ✓ Reduces hardcoding; agents can specialize at runtime
- ✓ Supports autonomous task decomposition and delegation (multi-agent systems)

Workday Example: Workforce Strategy Agent

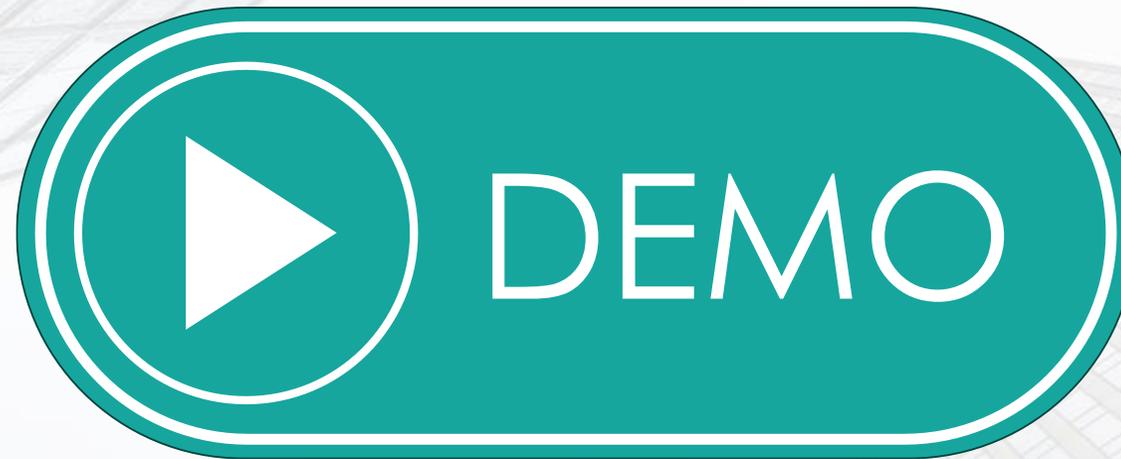
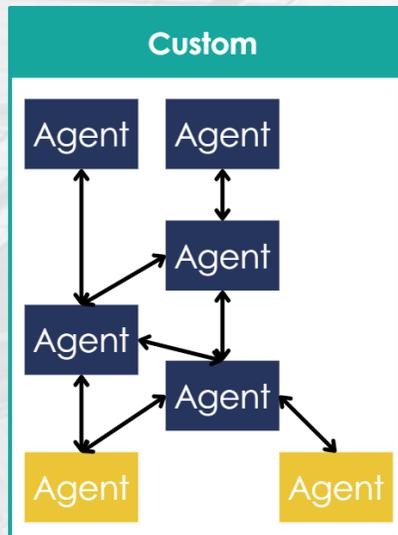
The CHRO wants to assess workforce readiness for a new expansion.

1. The Orchestrator receives the goal: *“Assess talent readiness across regions”*
2. It spawns or configures three dynamic worker agents:
 1. One for pulling skills coverage from Workday
 2. One for analyzing attrition risk
 3. One for summarizing training gaps





GROW Agent



Wrap-Up



We are building the Agents for your future needs



Reconciliation Agent

Matches transactions across ledgers, bank statements, and subledgers, then highlights gaps or duplicates. It explains variances, proposes journal entries or clearing actions, and routes exceptions to the right owner with context



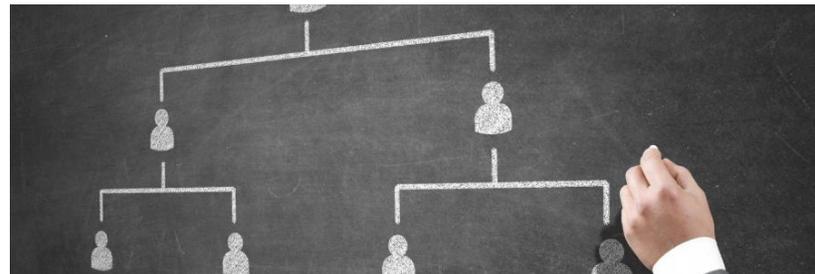
Security Agent

Users can request agent to access data; the agent analyzes the requests and asks for role approval. Can also identify security anomalies (mismatches between job profile, security access) and flag those for review.



Data Quality Agent

Give data quality assignments to collect, improve or audit data elements. The agent will flag anomalies, contact users for missing data and escalate where appropriate. Set your own data quality rules as guard rails.



Org Design Agent

Be able to request organizational changes using the org design agent. The org design agent will follow your master data management reasons, flag risks (e.g. span of control) and route the process for appropriate approvals.



Integration Agent

Proactively monitors your integration and flags any issues with integrations. Will do initial analysis on potential issues with broken integrations and suggest fixes.



Manager Assistant Agent

Supports managers with their recurring Workday tasks. Will be able to initiate complex MSS tasks. Supports manager in how to lead their team by giving advice, keeping an eye on sentiment and recurring teams within their team.

Launch of the HRIS Agent Strategy Blueprint



Strategy & Vision

Help HR tech lead define the vision and get buy in.

The slide displays the AGENT framework components: Ambitions & Outcomes, Governance & Guardrails, Experience & Platforms, Motive vs. Custom Agents, and Trajectory & Roadmap. It also features a 'Your AI Readiness Assessment Score' section showing a 'Developing - 62%' score, a 'What Your Score Means' section, and a 'Data Completeness' section.

- AGENT Framework & Templates
- Overview of all Workday Agents
- Agent Use Case Library for Workday
- Business case example for an agent
- AI Readiness Assessment
- Executive Pitch Deck template
- On demand webinars
- Workday AI Glossary for HRIS Teams



Architecture & Build

Show how to design and wire agents with Workday and Flowise.

The slide shows a 'Feedback Agent' workflow diagram, a table of 'Use Cases' for various agents, and a 'Super Agent' architecture diagram. The table lists use cases for HRIS, HR, and HRIS agents, such as 'HRIS Self-Service Agent' and 'HRIS Manager Assistant'.

- Flowise templates
- Early access to Flowise tutorials
- Reference Architecture Cheat Sheet
- Templates & guidance on MCP vs A2A protocols
- Overview of Multi Agent Architectures and advantages
- Overview of Flowise Design Patterns



Governance & People

Make it safe, compliant, and make sure people can run it.

The slide includes a 'Risk Matrix' table, an 'Agent Governance Board starter pack' graphic, an 'EU AI Act' risk level overview, and a 'Workday AI Governance' diagram. The risk matrix shows risk levels for different agents, and the EU AI Act diagram shows risk levels from Minimal to Unacceptable.

- Templates & guidance on RACI for agents
- EU AI Act overview for HR agents
- Risk and Controls checklist
- Agent Governance Board starter pack
- Templates & guidance on upskilling your HRIS team on agents

What is your next step in your agent journey?



10 Agents in 60 minutes

Intense Session with 10 unique custom Agents demos showed LIVE during webinar all within 1 hours. This is a must join session!

**Wednesday Jan 28th at
16:00 CET / 10:00 am ET**

You will see:

- Customer Agnostic AI Agents
- HCM and FIN Agents
- Geographic / Industry specific Agents
- Reactive and Proactive Agents



Scan to book

Contact Us

- If you want to build your agent strategy
- If you want to learn how to build agents inhouse
- If you have an agent use case you want to work together on



Rick Leunisse
rick.leunisse@incubane.com



Matt Komendolowicz
matt.komendolowicz@incubane.com

Thank you

