### AI Agent Architecture: Examples for Key Elements

Building on the overview of AI agent architecture, below I'll provide 2-3 practical examples for each major section, drawn from real-world applications in 2025. These illustrate how components, models, layers, and patterns are implemented in domains like business, healthcare, and autonomous systems. Examples are based on established frameworks (e.g., LangChain, AutoGen) and deployments (e.g., in enterprises like Wells Fargo for financial agents). Focus is on modularity, with agents often powered by LLMs like Gemini or GPT-4o for reasoning.

#### Key Components of AI Agent Architecture

Each component forms part of the perception-plan-act-learn loop.

| Component | Examples |

|-----------|----------|

| \*\*Perception/Input\*\* | 1. A customer service agent in banking (e.g., Wells Fargo's Fargo assistant) ingests user queries via chat or voice, pulling in account data from APIs.<br>2. An autonomous vehicle agent perceives road conditions via cameras and sensors, detecting obstacles in real-time.<br>3. A research agent scans emails or Slack for triggers, like "summarize latest market report," to initiate workflows. |

| \*\*Memory\*\* | 1. Short-term: A coding agent remembers code snippets from the current session to debug iteratively.<br>2. Long-term: A personalized shopping agent stores user preferences in a vector database for future recommendations.<br>3. Hybrid: A medical diagnostic agent recalls patient history from past interactions to inform current analyses. |

| \*\*Reasoning & Planning\*\* | 1. Chain-of-thought in trip planning: An agent decomposes "plan a vacation" into subtasks like budgeting, flights, and hotels.<br>2. Fraud detection: A financial agent analyzes transaction patterns to plan alerts or blocks.<br>3. Game AI: A chess agent simulates moves ahead using minimax algorithms integrated with LLMs. |

| \*\*Execution/Tools\*\* | 1. Calendar booking: An agent calls Google Calendar API to schedule meetings based on availability.<br>2. Data query: A business intelligence agent executes SQL queries on databases for report generation.<br>3. External integration: A weather agent invokes APIs like OpenWeatherMap to fetch forecasts and act on them (e.g., reschedule events). |

| \*\*Feedback/Learning\*\* | 1. Self-reflection: An essay-writing agent critiques its draft and refines based on quality scores.<br>2. User feedback: A recommendation agent adjusts suggestions after thumbs-up/down inputs.<br>3. Reinforcement: A robotic agent learns from trial-and-error in navigation, optimizing paths over time. |

#### Core Models of AI Agents

These models dictate the agent's responsiveness and intelligence level.

- \*\*Reactive Model\*\*: 1. Thermostat agent: Adjusts temperature based on current readings without history.<br>2. Simple chatbot: Responds to keywords like "help" with predefined menus.<br>3. Alert system: Triggers notifications on stock price drops via rule-based thresholds.

- \*\*Deliberative Model\*\*: 1. Route optimizer: Builds a mental map of traffic to plan the best path.<br>2. Investment advisor: Models market scenarios to recommend portfolios.<br>3. Legal review agent: Analyzes contracts against regulations before approving.

- \*\*Hybrid Model\*\*: 1. Virtual assistant (e.g., enhanced Siri): Reacts to voice commands while planning multi-step tasks like "book dinner and remind me."<br>2. Healthcare triage: Quickly assesses symptoms reactively, then deliberates on diagnoses with patient history.<br>3. Supply chain agent: Reacts to inventory alerts and plans restocking dynamically.

#### Essential Layers for Real-World AI Agents

The seven-layer framework ensures end-to-end functionality.

| Layer | Examples |

|-------|----------|

| \*\*Experience Layer\*\* | 1. Chat interface: Slack bot for team queries in project management.<br>2. Voice UI: Amazon Alexa agent handling home automation commands.<br>3. Multimodal: AR glasses agent overlaying info on real-world views. |

| \*\*Discovery Layer\*\* | 1. Web search: Agent retrieves news articles for market analysis.<br>2. Sensor fusion: Drone agent gathers environmental data via cameras and GPS.<br>3. Database query: HR agent pulls employee records for performance reviews. |

| \*\*Agent Composition Layer\*\* | 1. Role definition: A multi-agent team with "planner," "executor," and "critic" roles for software development.<br>2. Ethical constraints: Banking agent programmed to avoid biased lending advice.<br>3. Goal setting: E-commerce agent composed to maximize cart value while respecting user budgets. |

| \*\*Reasoning & Planning Layer\*\* | 1. LLM-based: Using GPT for hypothesizing solutions in troubleshooting.<br>2. Symbolic: Rule engines in compliance agents ensuring regulatory adherence.<br>3. Hybrid: Monte Carlo simulations in gaming agents for strategy planning. |

| \*\*Tool & API Layer\*\* | 1. Payment gateway: Agent integrates Stripe for transaction execution.<br>2. CRM tools: Salesforce API calls for updating customer leads.<br>3. Safeguarded actions: Email agent with approval loops before sending sensitive info. |

| \*\*Memory & Feedback Layer\*\* | 1. Vector store: Pinecone database for recalling similar past queries.<br>2. Feedback loop: Post-task surveys refining a tutoring agent's explanations.<br>3. Long-term adaptation: Evolving fraud patterns stored for future detections. |

| \*\*Infrastructure Layer\*\* | 1. Cloud scaling: AWS Bedrock for handling peak loads in customer service agents.<br>2. Security: Encryption and auditing in healthcare agents via HIPAA-compliant platforms.<br>3. Orchestration: Kubernetes for managing multi-agent deployments in enterprises. |

#### Major Architecture Patterns

These patterns provide blueprints for agent setups.

| Pattern | Examples |

|---------|----------|

| \*\*Single Agent + Tools\*\* | 1. Personal finance tracker: Agent uses budgeting tools to categorize expenses.<br>2. Research assistant: Queries Wikipedia APIs for fact-checking.<br>3. Code reviewer: Invokes GitHub tools to suggest improvements. |

| \*\*Sequential Agents\*\* | 1. Content pipeline: Writer → Editor → Publisher agents for blog posts.<br>2. Order fulfillment: Picker → Packer → Shipper in e-commerce.<br>3. Diagnostic chain: Symptom collector → Analyzer → Recommender in health apps. |

| \*\*Single Agent + MCP Servers + Tools\*\* | 1. Cross-platform integrator: Agent accesses tools via standardized protocols for IoT control.<br>2. Enterprise search: Queries multiple databases through a unified server.<br>3. API aggregator: Combines weather and traffic tools for commute planning. |

| \*\*Agents Hierarchy + Parallel Agents + Shared Tools\*\* | 1. Research team: Supervisor delegates parallel data gathering to workers.<br>2. Fraud network: Hierarchical agents analyzing transactions in parallel.<br>3. Game development: Lead agent oversees parallel testers sharing bug tools. |

| \*\*Single Agent + Tools + Router\*\* | 1. Customer support router: Directs queries to billing or tech tools.<br>2. Multi-tool selector: Chooses between search engines based on query type.<br>3. Workflow router: Routes tasks to email or calendar tools dynamically. |

| \*\*Single Agent + Human in the Loop + Tools\*\* | 1. Medical advisor: Pauses for doctor approval before recommendations.<br>2. Legal drafter: Human reviews contract clauses generated via tools.<br>3. Investment bot: Seeks user confirmation for high-risk trades. |

| \*\*Single Agent + Dynamically Call Other Agents\*\* | 1. Master orchestrator: Calls specialist agents for niche tasks like translation.<br>2. Project manager: Invokes coder or designer agents on-demand.<br>3. Virtual concierge: Dynamically engages booking or info agents. |

| \*\*Agents Hierarchy + Loop + Parallel Agents + Shared RAG\*\* | 1. Complex simulation: Hierarchical agents looping feedback in climate modeling.<br>2. Collaborative R&D: Parallel researchers sharing retrieved knowledge.<br>3. Crisis response: Looping agents adapting plans with shared data sources. |

#### Design Patterns for Agentic Behaviors

Reusable behaviors enhancing agent capabilities.

- \*\*Reflection\*\*: 1. Code debugger: Self-critiques syntax errors.<br>2. Content refiner: Iterates on ad copy for engagement.<br>3. Strategy optimizer: Reviews game moves post-match.

- \*\*Tool Use\*\*: 1. Browser tool: Searches web for current events.<br>2. Calculator API: Handles math in financial projections.<br>3. Email sender: Automates notifications.

- \*\*ReAct (Reason + Act)\*\*: 1. Query resolver: Reasons population stats then acts to compare.<br>2. Troubleshooting: Thinks through errors before tool calls.<br>3. Navigation: Plans steps then executes moves.

- \*\*Planning\*\*: 1. Event organizer: Decomposes party planning into venues and invites.<br>2. Supply chain: Subtasks for sourcing and delivery.<br>3. Learning tutor: Breaks lessons into modules.

- \*\*Multi-Agent Collaboration\*\*: 1. Software team: Planner + Coder + Tester for app building.<br>2. Debate panel: Agents argue pros/cons for decisions.<br>3. Research consortium: Specialists collaborate on reports.

#### Orchestration Patterns

For coordinating workflows.

- \*\*Sequential\*\*: 1. Approval flow: Draft → Review → Sign-off.<br>2. Recipe executor: Prep → Cook → Serve steps.<br>3. Onboarding: Welcome → Training → Assignment.

- \*\*Concurrent\*\*: 1. Multi-source analysis: Parallel news scraping from sites.<br>2. Testing suite: Simultaneous unit tests.<br>3. Brainstorming: Independent idea generation then merge.

- \*\*Group Chat\*\*: 1. Consensus builder: Agents discuss best vacation spots.<br>2. Risk assessment: Debate potential threats.<br>3. Idea validation: Vote on product features.

- \*\*Handoff\*\*: 1. Escalation: Basic bot hands off to expert agent.<br>2. Phase transition: Planning agent to execution agent.<br>3. Specialization: General query to domain-specific handler.

- \*\*Magentic\*\*: 1. Open-ended explorer: Builds subgoals for adventure games.<br>2. Innovation driver: Iterates ideas in product design.<br>3. Adaptive learner: Evolves strategies in unknown environments.

These examples demonstrate versatility across industries. For Wells Fargo, adapt patterns like Sequential Agents for compliance workflows or Tool Use for API integrations in fraud detection. If implementing, prototype with tools like AutoGen for quick testing.