INF 385T · FALL 2026

TEXAS
The University of Texas at Austin

INF385T: Special Topics in Information Science: Language Models and Agentic Workflows

Week 10: Human Agent Interaction

Dr. Abhijit Mishra

Announcements

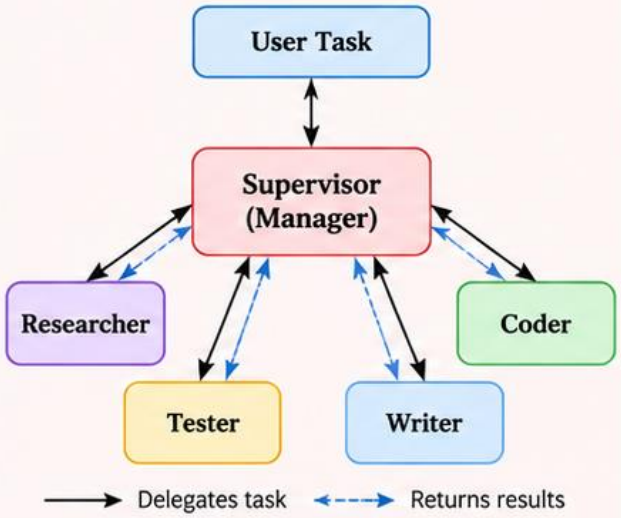
- **Group Project 1** – Package upload (Nov 5)
  - No class on November 5, instead prepare your presentation material for project 1
  - Video presentation, slides, report
  - QA sharing in the upcoming two days
- **Group Project 2** – Proposal submission
  - Deadline: 10/30
- **Industry Keynote:** Nov 19
- **Group Project 2** – Final Presentation (Dec 3)
- **Group Project 2** – Final Report (Dec 10)

W9 Recap: Single and Multi-agent Systems

- Independent subtasks → parallel specialists.
- Clear sequential dependencies → pipeline.
- Debuggability critical → centralized supervisor.
- Resilience at scale → hierarchy or sparse peer network.
- Example: software team: architect → coder → tester → reviewer.
- Decision gate: retain extra agents only if measured gains

Centralized

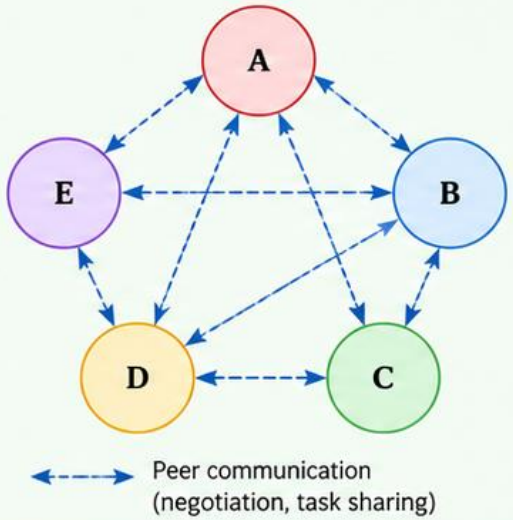
Supervisor owns global task and aggregation.



Delegates	Supervisor to workers
Speaks	Mainly via supervisor
State	Centralized (supervisor)
Failure	Supervisor detects and re-routes
Example	Research report with planning, coding, testing, and writing

Decentralized

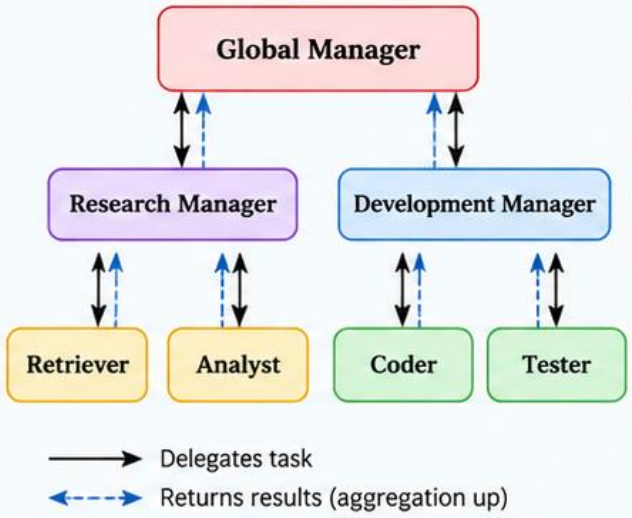
Peers negotiate without one global coordinator.



Delegates	Peers to peers
Speaks	Direct peer-to-peer
State	Distributed (shared/replicated)
Failure	Peers detect and recover locally (e.g., re-negotiate)
Example	Multi-agent research where agents bid for subtasks

Hierarchical

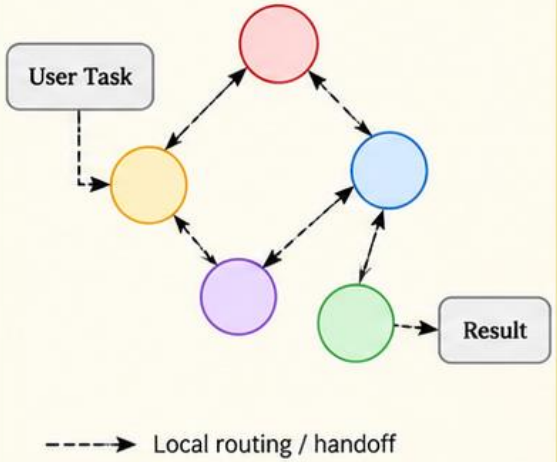
Domain managers supervise local teams.



Delegates	Top-down (multi-level)
Speaks	Within teams and to parent
State	Hierarchical (local at each level + summaries up)
Failure	Local manager handles and escalates if needed
Example	Enterprise workflow with domain-specific teams

Swarm

Local routing through handoffs, without global state.



Delegates	Dynamically via handoffs
Speaks	Direct with next agent
State	Local (no global state)
Failure	Next agent can take over or reroute
Example	Open-ended task solved by a chain of specialized agents

Recap: Popular multi-agent paradigms

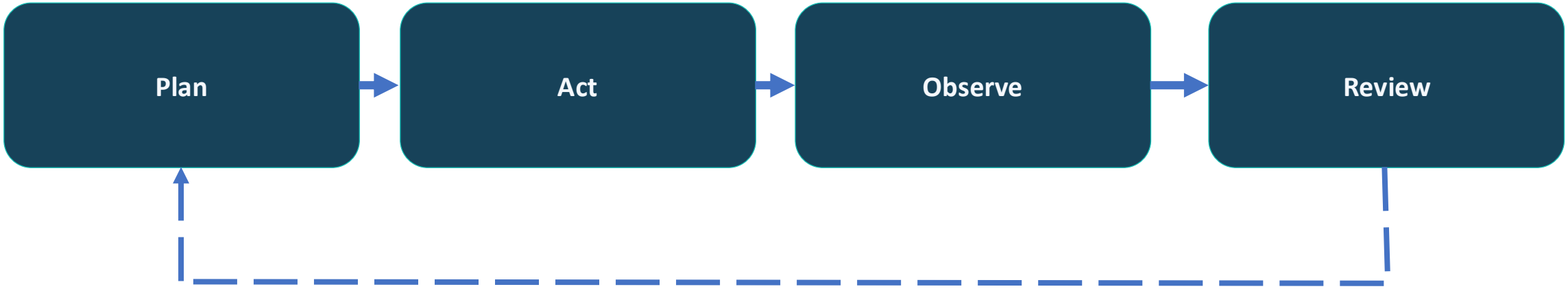
Week 10: Roadmap

- **Part 1: Human Agent Interactions**
  - Human-in-the-loop and human-on-the-loop systems
  - Approval boundaries and escalation rules
  - Levels of agent autonomy
  - Transparency and explainability
  - Interruptibility, user correction and trust calibration.
- **Part 2: Project 1 Discussions**

Human-in-the-loop and human-on-the-loop systems

Human in the loop

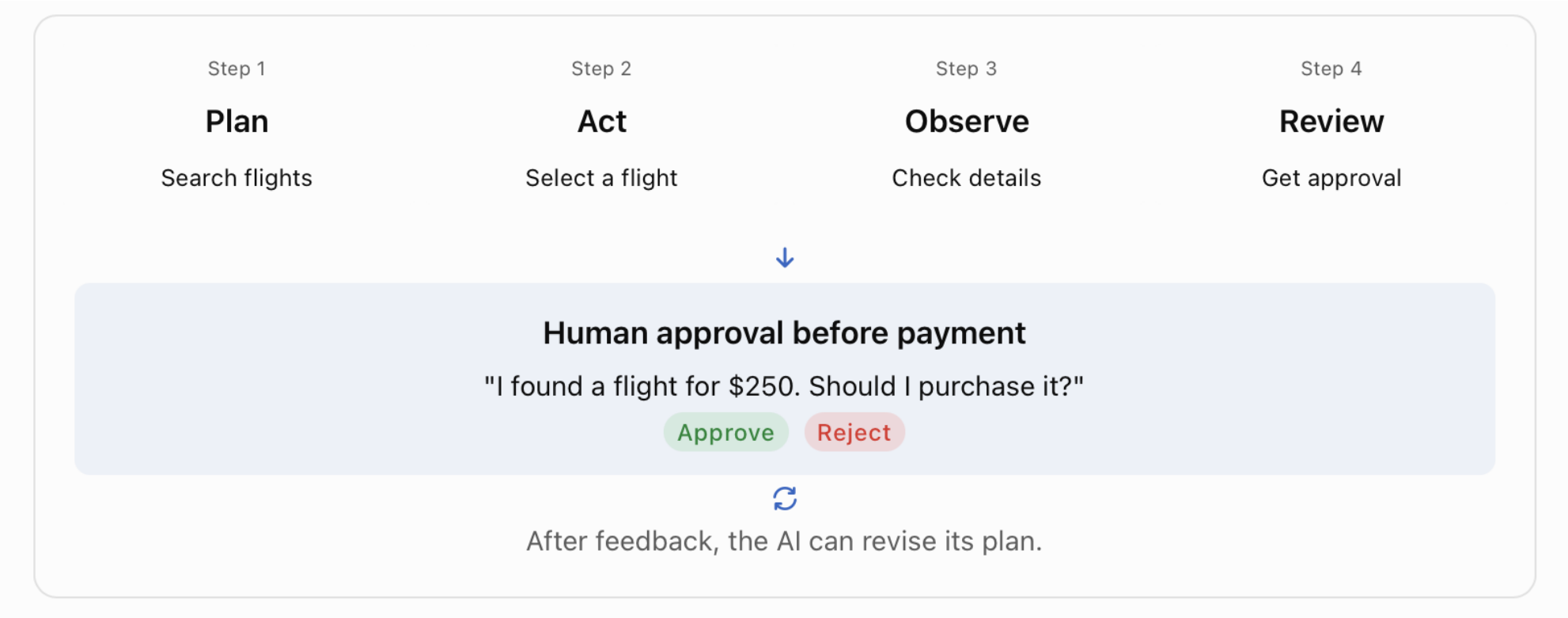
Autonomous agents can observe, decide and act across tools. Errors can propagate beyond a single answer. Oversight must match consequence and reversibility.



HITL DESIGN PRINCIPLE : Oversight should be proportional to risk.

“Book me a flight from Austin to New York for tomorrow, under $300.”

The AI agent follows the four steps



Why oversight matters

- Suppose the AI accidentally selects a flight for the wrong date.
  - **Without human oversight:** The AI purchases the wrong ticket, potentially wasting money.
  - **With human oversight:** You review the details, identify the mistake, and tell the AI to correct it before purchasing.

What does "oversight should be proportional to risk" mean?

AI action	Risk	Human oversight needed?
Search for flights	?	?
Compare ticket prices	?	?
Purchase a $250 ticket	?	?
Cancel a nonrefundable ticket	?	?

What does "oversight should be proportional to risk" mean?

AI action	Risk	Human oversight
Search for flights	Low	AI can do it independently
Compare ticket prices	Low	AI can do it independently
Purchase a $250 ticket	Higher	Human approval required
Cancel a nonrefundable ticket	Higher	Human approval required

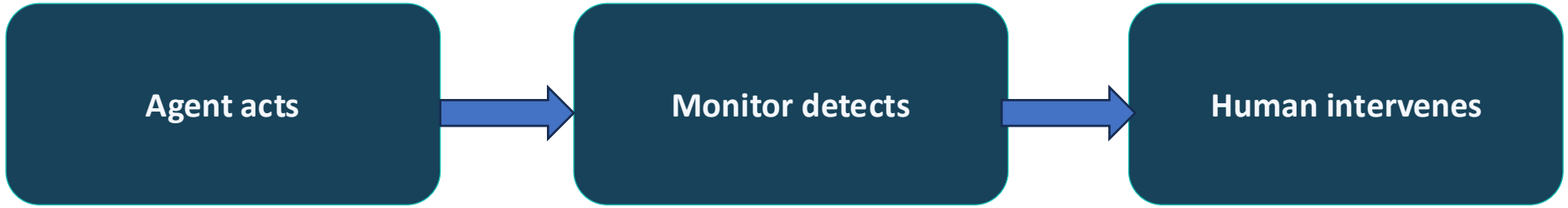
Human-in-the-loop (HITL)

- A person must approve or modify a proposed action before execution. Best for irreversible, high-impact or sensitive decisions.



Human-on-the-loop (HOTL)

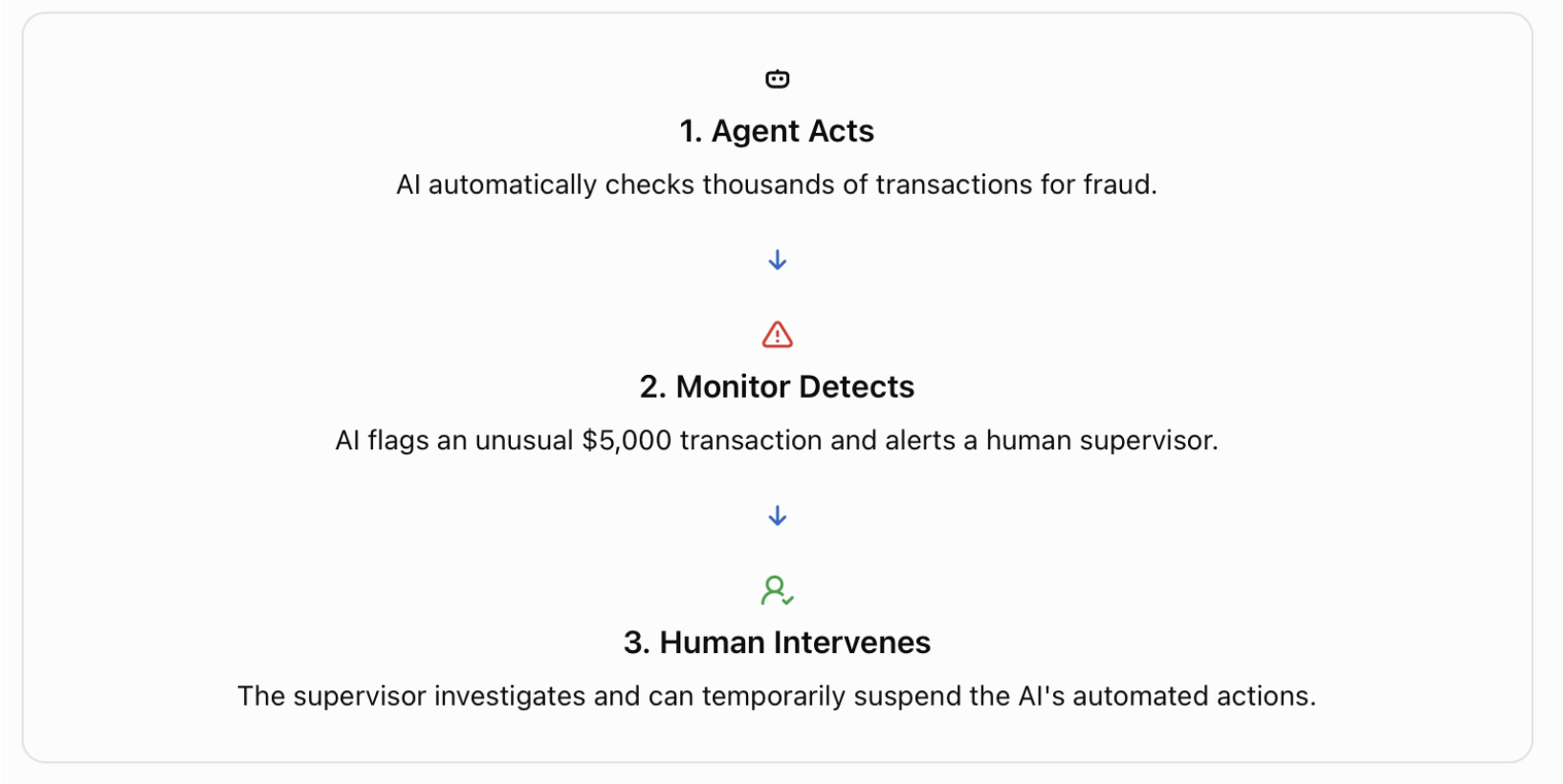
- Agent acts inside preauthorized bounds; a person monitors exceptions and can intervene. Requires visibility, latency budgets and a reliable stop path.



HOTL DESIGN PRINCIPLE : Human Controls should be technically enforceable

Example: Simple Example: AI Monitoring a Bank Account

Imagine a bank uses an AI agent to detect suspicious transactions.



Agent is allowed to monitor transactions automatically, while a human supervisor watches f or unusual situations and can intervene.

Why is this Human-on-the-Loop (HOTL)?

- The AI does not need human approval for every transaction it checks. It operates continuously within predefined limits.

- The human monitors its performance and intervenes when necessary.

What does "Human controls should be technically enforceable" mean?

- Imagine the supervisor clicks **STOP AI.**

- The system must actually stop the AI from taking further actions—not merely display a warning or ask the AI to stop.

- This requires a reliable technical mechanism that can block actions even if the AI attempts to continue.

HITL vs HOTL

Human in the loop

- Human approves important actions before execution
- Human is part of the decision process
- Example: Approving a flight purchase

Human-on-the-Loop (HOTL)

- AI executes permitted actions automatically
- Human supervises and can intervene
- Example: Monitoring AI fraud detection

What’s a monitor?

- Does it have to involve an LLM call?

What’s a monitor?

- A monitor can be another LLM, but it doesn't have to be.
- Three common types:

Rule-based monitor	Checks predefined rules	Alert if a transaction exceeds $5,000
LLM-based monitor	Another LLM evaluates the agent's decisions	Detect whether an AI is about to disclose confidential information
Hybrid monitor	Combines rules, LLM judgments, and other signals	Check transaction limits and evaluate suspicious behavior

LLM 1 — AI Agent

Receives a customer request and prepares to issue a $2,000 refund.

↓

Proposed action is inspected

LLM 2 — AI Monitor

Evaluates the refund justification and flags it as potentially unauthorized.

↓

Alert and temporary hold

Human Supervisor

Reviews the case and decides whether to approve, reject, or investigate.

Important distinction

- The monitor can detect a problem, but detection alone does not guarantee safety.
- For example, if the AI agent tries to issue a $10,000 refund when the maximum authorized amount is $500:
  - A rule-based monitor can reliably identify the limit violation.
  - An LLM monitor can help explain why the refund looks suspicious.
  - A separate technical control must actually block the unauthorized refund.

Human-out-of-the-loop (HOuTL)

- Automation may proceed without routine human involvement only where risk is bounded, validation is strong, and recovery is reliable. Not a blanket autonomy target.
- **Example?**
  - An AI system automatically sorts incoming emails into spam or inbox without human involvement, because errors are low-risk and users can easily recover misplaced emails.

Design principle: Choose autonomy by “risk” and not reward

- Evaluate impact, reversibility, uncertainty, data sensitivity, exposure and time to intervene. Higher risk calls for tighter boundaries and human authority.

- Autonomy is contextual
  - The same agent may summarize freely, draft with review, send only after approval, and never authorize transfers without designated controls.

Approval boundaries and escalation rules

Approval boundaries

- Define exactly what an agent may read, recommend, write or execute; distinguish preapproval, per-action approval and prohibited actions.



Exercise: (Allowed / Need Approval / Forbidden)

- **Scenario 1: AI Cybersecurity Agent**
  - Isolate a suspicious device for 5 minutes under a preapproved emergency policy **Allowed**
  - Delete forensic evidence to hide a breach **Forbidden**
  - Shut down an entire corporate network to stop a cyberattack

Need Approval

Exercise: (Allowed / Need Approval / Forbidden)

- **Scenario 2: AI Healthcare Agent**
  - Prescribe medication using another doctor's credentials **Forbidden**
  - Analyze patient records and flag possible drug interactions **Allowed**
  - Change a patient's medication or dosage **Need Approval**

Exercise: (Allowed / Need Approval / Forbidden)

- **Scenario 3:** AI Recruitment Agent
  - Predict from resume if an applicant belongs to a certain age group based on experience and invalidate candidates **Forbidden**
  - Reject candidates automatically without human review when hiring policy requires review **Need Approval**
  - Screen applications against job-related criteria and recommend candidates **Allowed**

Policy gate architecture

- Every proposed tool action is checked for identity, permissions, data class, monetary/operational limits and human signoff before execution.



Risk-tiered approval matrix

- Example policy:
  - low-risk internal summaries **auto**;
  - external messages require **review**;
  - destructive changes require elevated approval;
  - prohibited actions are blocked.



Designing a useful approval request

- Show intended action, target, scope, evidence, uncertainty, expected effects, alternatives, and an explicit approve/edit/reject choice.
- Escalate when confidence is inadequate, policy conflicts arise, scope changes, sensitive data appears, tools fail, or the outcome becomes difficult to reverse.

Designing a useful approval request

Give the human enough context to make an informed decision.

Agent's plan

AI agent proposes an action

Example:
Delete old customer records to reduce storage costs.

Approval request shown to human

Intended action	Delete customer records older than 5 years.
Target	1.2 million records in the Marketing database.
Scope	Marketing database only (does not include Sales or Finance).
Evidence	Records are inactive for > 5 years, no open support cases, confirmed by data retention policy v2.1.
Uncertainty	Some records may still be needed for ongoing legal investigations (est. 2% risk).
Expected effects	Free ~500 GB storage, reduce costs by $2,000/year.
Alternatives	(1) Archive instead of delete (2) Delete only records older than 7 years.

Human decision

Choose an explicit action with optional comments.

Approve

Proceed with the proposed action as described.

Edit

Modify the scope, conditions, or parameters.

Reject

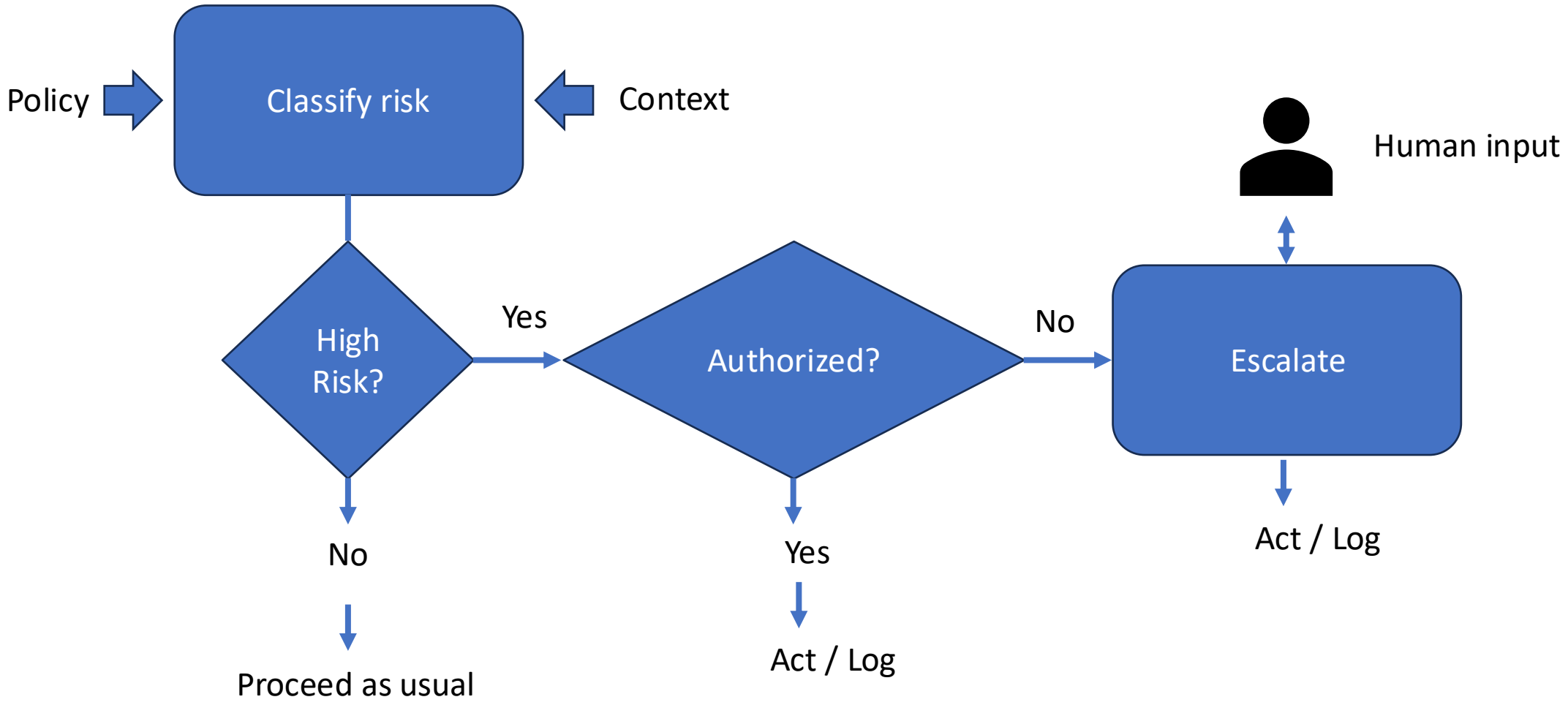
Do not proceed.

Escalation triggers

- Escalate when confidence is inadequate, policy conflicts arise, scope changes, sensitive data appears, tools fail, or the outcome becomes difficult to reverse.
- Example:
  - **AI Agent Managing Customer Data**
  - **Imagine an AI agent is instructed to delete duplicate customer records from a company's database.**
  - **The agent should stop and escalate to a human supervisor in these situations:**

Escalation Trigger	Example	Why Escalate?
1. Low Confidence	AI is only 60% sure two records are duplicates.	Risk of deleting valid data.
2. Policy Conflict	Policy requires 7-year retention; user requests deletion after 3 years.	Violates retention policy.
3. Scope Change	Authorized to clean 100 records; discovers 10,000 duplicates.	Exceeds authorized scope.
4. Sensitive Data	AI encounters medical records or Social Security numbers.	Requires additional privacy controls.
5. Tool Failure	Database backup fails before deletion.	Data may become unrecoverable.
6. Irreversible Outcome	Deletion would permanently erase transaction histories.	Cannot reliably undo the action.

Escalation decision tree



Transparency and explainability

Transparency: what to reveal

- Disclose that AI is involved, the goal, tool use, data accessed, constraints, material uncertainty and actions taken.
- Transparency must be timely and actionable.

Explainability versus transparency

- **Transparency** tells users what the system did and its limits;
- **Explainability** provides understandable reasons for a recommendation or action.
- Neither guarantees correctness.

Explanation design patterns

Use

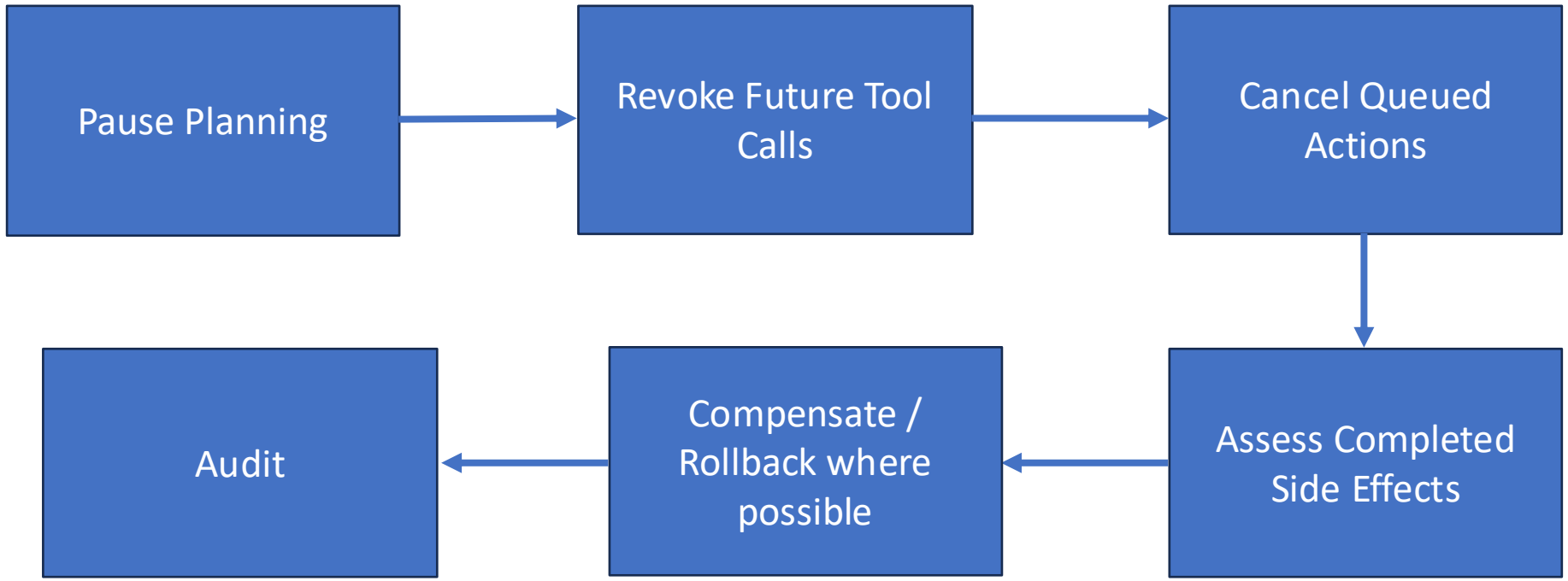
- short rationale,
- supporting evidence,
- uncertainty labels,
- counterfactuals

Drill-down audit trails. Match detail to user expertise and decision stakes.

Interruptibility

- A stop control must prevent new actions, cancel pending work where feasible, preserve state and report what has already happened.
- Safe shutdown is a system property.

Interruptibility: Stop and rollback protocol



Trust calibration

- Good trust matches actual capability
- Undertrust wastes useful automation
- Overtrust causes uncritical reliance
- Design for verified reliance, not maximal trust.

Oversight metrics and testing

- Track override rate,
- Escalation precision,
- Intervention latency
- Approval quality
- Unsafe-action prevention
- Recovery success
- Operator workload.

W10: Summary

Autonomous Agents should:

- Match autonomy to risk
- Keep approval authority explicit
- Make escalation actionable
- Explain decisions
- Preserve stop and correction paths
- Measure real oversight effectiveness.

Thank you !!!