

# Time-Traveling Agents: Event-Driven AI Systems That Rewind, Retry, Recover

Danish Rehman

November 25, 2025

- LLM agents are increasingly used in real-time decisioning (marketing, support, personalization)
- But first-pass decisions are often:
  - Wrong, unsafe, or off-policy
  - Hard to debug, audit, or reproduce
- This talk shows how to combine event-sourcing + replay to:
  - **Rewind** to unsafe decisions
  - **Retry** with explicit corrections
  - **Recover** into deterministic, compliant outcomes

# Problem: First-Pass Agents in Production

- Deployed into:
  - Marketing automation, customer journeys, ticket triage
- Typical failure modes:
  - No reliable memory across long workflows
  - No notion of time, frequency, or caps
  - Weak alignment with domain rules and constraints
  - Hallucinated or off-brand output
- Business impact: revenue loss, brand risk, legal exposure, user churn

# Idea: Time-Traveling Agents

## Core concept

Treat the agent as **stateless** and the event log as the **source of truth**.

- Every decision is an **event**, not just an RPC result
- On violation:
  - Rewind state to a safe boundary using event replay
  - Inject explicit corrections (caps, constraints, overrides)
  - Re-run the agent to derive a **corrected decision**
- Preserve both naive and corrected decisions for audit and learning

# Concrete Use Case: Marketing Decision Engine

- Stream of user events (ViewedProduct, AddedToCart, BecameVIP, ...)
- Orchestrator issues commands like `GenerateOffer(User123)`
- Agent proposes:
  - Discount level, template, send time, copy
- Validators enforce:
  - Discount limits, quiet hours, frequency caps, legal wording
- If a rule is violated, the system:
  - Rewinds to pre-decision state
  - Applies corrected constraints
  - Replays history to produce a safe, deterministic decision

# What Attendees Will Learn

- How to pair LLM agents with event-sourcing and replay instead of treating them as one-shot APIs
- Patterns for:
  - Modeling decisions as events
  - Rewinding and replaying only the affected slice of history
  - Injecting corrections without polluting state
- How this design improves:
  - Safety and compliance of AI decisions
  - Debuggability and auditability
  - Experimentation and counterfactual analysis

# Why This Matters Now

- Organizations are shipping LLM-driven features faster than safety mechanisms
- Simple guardrails and prompt tweaks are not enough for high-stakes decisions
- Time-Traveling Agents offer:
  - A practical blueprint that leverages existing event infrastructure
  - A path to move from “best-effort AI” to **replayable, explainable, reversible** systems

**Goal: Help practitioners design AI systems that can safely rewind, retry, and recover.**

## Time-Traveling Agents

Event-Driven AI Systems That Rewind, Retry, Recover