adaptcx

TECHNICAL PAPER

The Crystallization Loop

An architectural pattern for making generative AI safer in regulated CX by turning recurring model variance into governed, deterministic playbooks.

A production architecture pattern for consistency, repeatability and auditability in regulated AI. AdaptCX builds on an earlier, smaller version. The expanded platform has not yet been deployed in full at a customer site.

GUIDE MAP

Contents

The problem with pure RAG The serving loop in detail The promotion loop in detail Audit logging of every retrieval and response Months 4-6 - Capture loop:

SECTION

The problem with pure RAG

Retrieval-augmented generation is a powerful default for grounding AI responses in source material. You retrieve the relevant chunks, you stuff them into the prompt, you let the model write the answer. The answer is grounded - every fact has a source - and the architecture scales gracefully as the corpus grows. It also has two failure modes that don't show up until production - Same question, different answer: the retrieval pulls slightly different chunks on different calls, and the model phrases the answer differently each time. Substance is consistent. Wording is not. - The auditor's question: 'Why did the system say what it said?' has the same answer every time - 'because the model produced it from these chunks.' That answer is true and not useful. There's no policy decision to point at, no playbook version, no human in the chain. For commercial CX where the cost of variance is low, pure RAG is fine. For regulated CX where the cost of variance is high - eligibility determinations, member benefit explanations, policy citations, hearing procedures - it is not. What crystallization means Crystallization is the process of taking a recurring generative response pattern and converting it into a deterministic playbook. The generative response is the seed; the playbook is the crystal. Once a playbook exists for a given question class,

SECTION

The problem with pure RAG (continued)

the system uses it. The model is invoked only when no playbook matches. The architecture has three loops, running at different cadences The serving loop Handles every inbound interaction. Runs in milliseconds. For each incoming question, it tries deterministic match first against the playbook library, falling back to generative-with-RAG only if nothing matches. This is the hot path; everything else is cold. The capture loop in detail Watches the generative responses. Clusters them by intent, retrieval signature, and surface form. A cluster of responses that consistently answer the same question - across different agents, different calls, different days - becomes a candidate for crystallization. The promotion loop Takes a candidate cluster and produces a draft playbook: a deterministic response template with the key facts, the citations, and the canonical phrasing. The draft goes to a human reviewer (a domain expert at the customer agency) who edits, approves, and

SECTION

The serving loop in detail

assigns a version number. The approved playbook enters the library. From the next call forward, that question class hits the deterministic path. The serving loop is conceptually simple but operationally precise. The order of operations matters - Intent classification: a small, fast model classifies the inbound question into one of the agency's intent categories. Use an embedding-based classifier with a calibrated similarity threshold - letting the LLM classify is too slow and too variable for the hot path. - Playbook lookup: the intent category is the key into the playbook library. If a versioned playbook exists, the system instantiates it with case-specific variables and returns the response. The lookup is sub-millisecond. - Generative fallback: if no playbook matches, fall back to RAG. The retrieval pulls relevant policy documents; the model produces a response with citations; the response is returned. - Telemetry: every interaction emits structured telemetry - intent classification result, playbook hit/miss, retrieval signature, response, latency - to the capture loop's input stream. Playbook coverage depends on the workflow and knowledge library. Evaluate coverage, exceptions and quality; no customer outcome is claimed here. The capture loop in detail The capture loop runs every six hours on the previous window's generative

SECTION

The serving loop in detail (continued)

responses. It does two things: cluster, and score. Clustering can combine intent classification, response embeddings and retrieval signatures. Calibrate thresholds against representative human-reviewed data. Scoring Ranks clusters by crystallization value. A cluster scores high if it (a) appears frequently - the question is recurring, (b) has high within-cluster consistency - the answer is stable across instances, and (c) contains responses that the reviewing agents accepted without modification - the AI got it right repeatedly. High-scoring clusters become candidates.

SECTION

The promotion loop in detail

Promotion is the human-in-the-loop step. We deliberately keep it human. The system drafts the playbook; a domain expert at the customer agency approves it. The draft includes the canonical response, the variable slots filled at runtime, the citations to source policy, and a metadata block with version number, approving reviewer, and effective date. The reviewer edits in a structured editor that enforces the playbook schema; a published playbook cannot be missing required fields. Once approved, the playbook is versioned (semver, with breaking-change discipline) and deployed to the playbook library. Every interaction served by the playbook records the playbook version in the audit chain. If a regulator ever asks 'what was the policy on X on date Y,' the answer is in the audit log - playbook version, approving reviewer, effective date - every variable named The crystallization loop is an approach to evaluating generative AI in regulated work. The model produces variety; the loop converts variety into consistency; consistency is what an auditor can trust. Why this matters beyond Assist The crystallization loop is part of the expanded AdaptCX platform design. AdaptCX builds on an earlier, smaller version. The expanded platform presented here has not yet been deployed in full at a customer site. Capabilities and illustrative workflows

SECTION

The promotion loop in detail (continued)

Plan the implementation in phases. Start with representative questions and approved source material. Check retrieval quality, source references and staff review before adding playbook suggestions. Agree evaluation criteria and responsibilities for each phase; no fixed delivery schedule or response-time result is promised.

SECTION

Audit logging of every retrieval and response Months 4-6 - Capture loop:

- - Telemetry pipeline emitting structured per-interaction events - Clustering job running on rolling windows - Manual review surface for inspecting clusters Months 7-9 - Promotion and serving: - Playbook editor with schema validation - Versioned playbook library with semver discipline - Serving loop with playbook lookup before generative fallback Months 10-12 - Operations: - Reviewer dashboards showing playbook performance over time - Cluster-to-playbook conversion as standard operations work - Automated regression testing on every playbook change

NEXT STEP

Continue the conversation

If this guide surfaced questions about your architecture, compliance posture, deployment model or AI operating model, talk with an AdaptCX engineer or solutions architect.

Talk to AdaptCX →