Adaptive Recommenders in the Real World

Mallika Rao, Senior Engineering Manager, Zocdoc
(ex-Netflix, Walmart, Twitter)

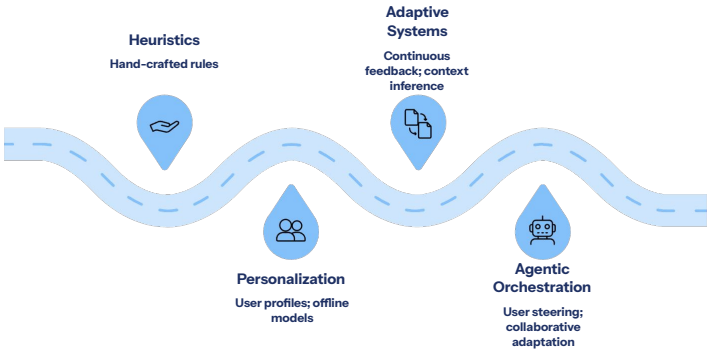
QCON AI 2026

MOTIVATION

Why This Talk Exists

Recommendation systems are sophisticated production AI systems.

The real challenges are:



Adaptation

Continuous learning under constraints

Evaluation

Metrics that reflect system health

Operational Trust

Reliable, debuggable at scale

Evolution

Improve without breaking

CONTEXT

From Hand-Tuned Rules to AI-Native Systems

"Many systems today are in both worlds."

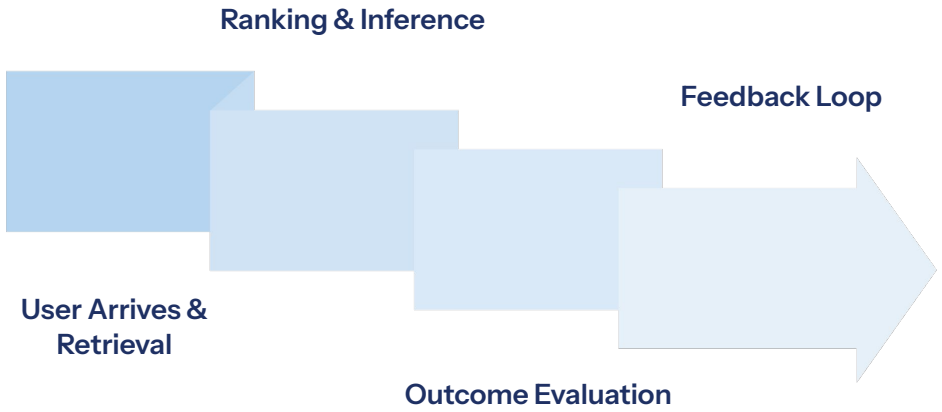
Classical Systems

- Hand-crafted rules
- Static ranking pipelines
- Offline tuning cycles
- Feature bottlenecks
- Fixed retrieval strategies

AI-Native Systems

- Dense embeddings
- Context-aware understanding
- Continuous adaptation
- Dynamic orchestration layers
- Learned representations

The Recommendation Lifecycle



A mental model for how recommendation systems operate end to end.

ROADMAP

What We'll Cover

01

Adaptive Architecture

Adaptive vs. batch

02

Real-Time Inference

Latency, fanout, scale

03

Evaluation Systems

Offline, online, judge metrics

04

Migration Strategies

Safe paths to AI-native

05

Operational Lessons

Debugging, observability, trust

06

Future Evolution

Ranking to adaptation systems

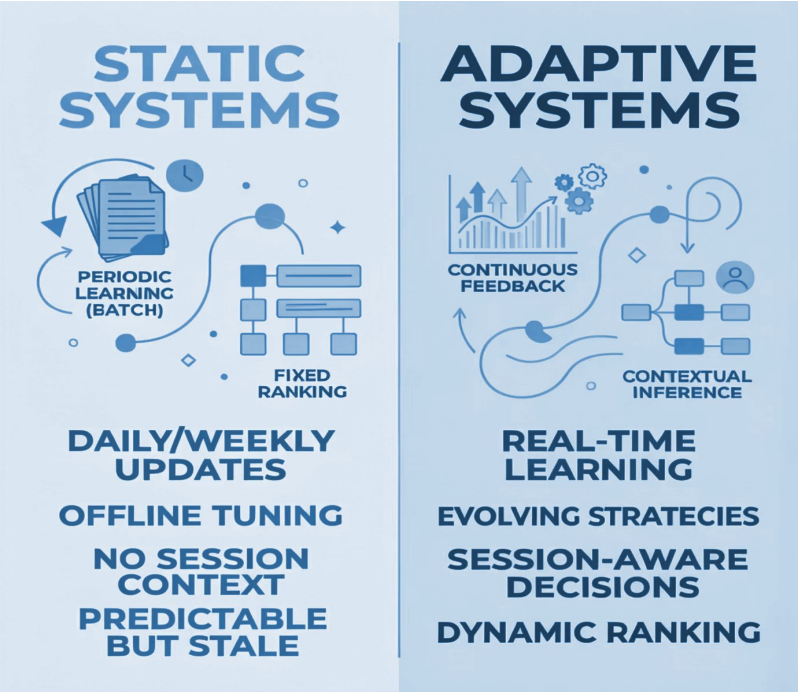
SECTION 2

What Makes Systems Adaptive

Properties that separate adaptive systems from batch pipelines.

CORE CONCEPT

Static vs. Adaptive Systems



Feedback latency

Not model sophistication, is what separates static from adaptive

Policy mutability

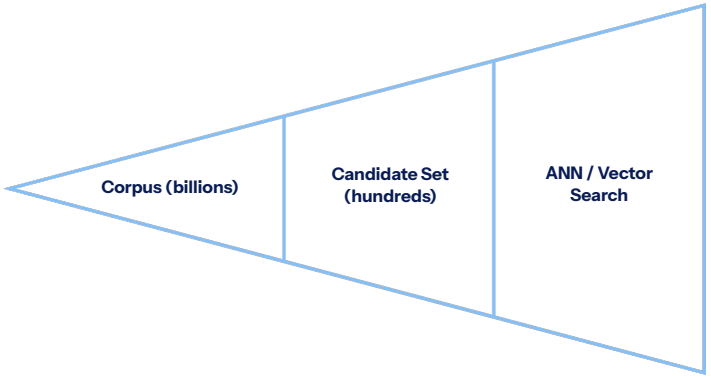
Adaptive systems update ranking weights per request; static systems don't

Production spectrum

Most systems sit between extremes; know where yours sits

RETRIEVAL

Candidate Generation Is a Breadth Problem



"Retrieval determines what the system is even allowed to consider." Ranking cannot rescue items retrieval never surfaced.

Embeddings

Dense vectors from interactions

ANN / Vector Search

Approximate nearest neighbor at scale

Hybrid Retrieval

Sparse + dense signals for recall

RANKING

Ranking Is a Precision Problem

Contextual Features

Time, device, location, and entry point shape weights.

Behavioral Features

Session history, dwell time, skip patterns, recency signal intent.

Multi-Stage Ranking

Coarse scoring plus fine re-ranking balances cost.

Adaptive Weighting

Weights shift by browsing or search mode.

ORCHESTRATION

Orchestration Is an Experience Problem

"Recommendation systems are evolving from ranking engines into orchestration systems."

Modern systems coordinate experiences across surfaces—not just rank items within one request. Optimization shifts from item to session to relationship.

Interaction Loops

Each action feeds orchestration—and future models.

The system becomes self-reinforcing.

Adaptive Surfaces

Layout and content adapt to context

User Steering

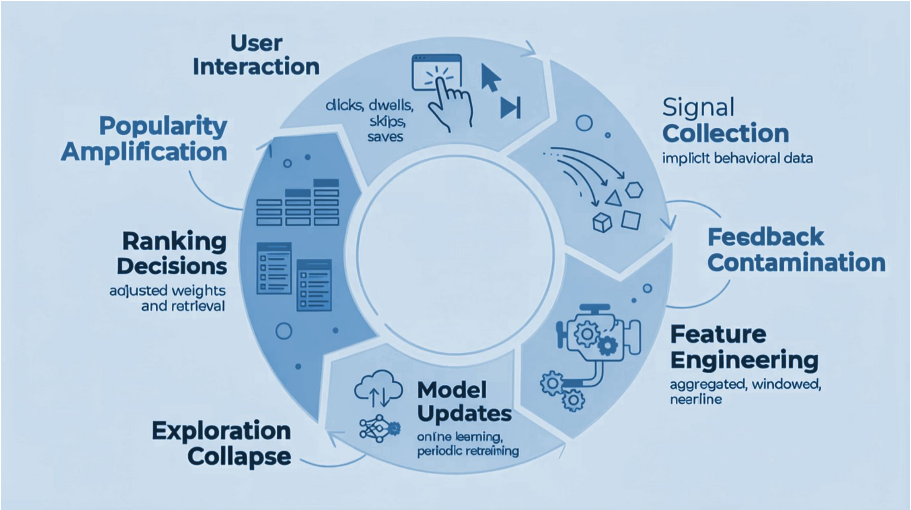
Systems respond to mid-session feedback

Session Evolution

Recommendations evolve as intent clarifies

SIGNALS

Feedback Loops — Everywhere



Every interaction becomes a training signal — including signals from the system's own decisions.

Feedback Contamination

Signals reflect system bias, not user truth

Popularity Amplification

Popular items get more exposure, more clicks

Exploration Collapse

System stops surfacing new content

SECTION 3

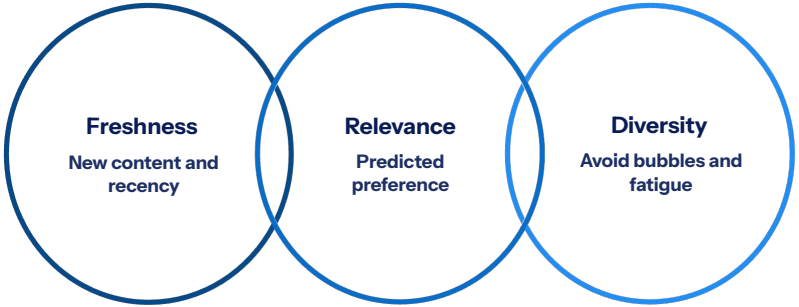
Case Study: Adaptive Homepage Systems

Adaptive theory meets user behavior.

CASE STUDY

Why Homepage Systems Are Hard

The homepage must serve unknown intent, respect fatigue, and balance freshness with familiarity.



Changing Intent

Different goals each session

Novelty vs. Fatigue

Familiar feels stale; novel feels irrelevant

Session Drift

Intent can shift mid-session

Diversity Constraints

Rules, policies, and fairness intersect

SESSION DYNAMICS

A Session Is a Moving Target

One evening can move from comedy to thriller to family content.

1

Session Start

Long-term preference dominates.

2

Early Signals

First interactions update short-term intent.

3

Mid-Session

Volatility peaks. Competing signals emerge.

4

Session Convergence

Strong signal established. Ranking narrows.

Static session context underserves evolving intent.

REAL-TIME ADAPTATION

Freshness Beats Sophistication

User Action

Event

Feature Update

Retrieval

Ranking

Recommendation

Users experience freshness.

They never experience model complexity.

Simple models with fresh data often beat complex models on stale data.

KEY INSIGHT

Exploration vs. Exploitation

"Optimization creates behavioral gravity."
Over-exploiting preferences narrows recommendations and trains users to want less.

The Core Tradeoff

Exploitation maximizes short-term reward. Exploration preserves long-term trust.

Neither pure strategy is correct. The ratio must be tuned.

Safe Exploration

Inject novel candidates with bounded risk

Uncertainty Handling

Treat low-confidence predictions as exploration

Recommendation Collapse

Exploitation shrinks the item universe

What Most Teams Underestimate

Freshness beats sophistication

Recent items often outperform stale, personalized ones.

Evaluation is harder than ranking

Building a model is easier than proving it works in production.

Observability is a product requirement

Unexplained decisions erode trust.

Failures are often systems failures

The root cause is often data, staleness, drift, or incentives.

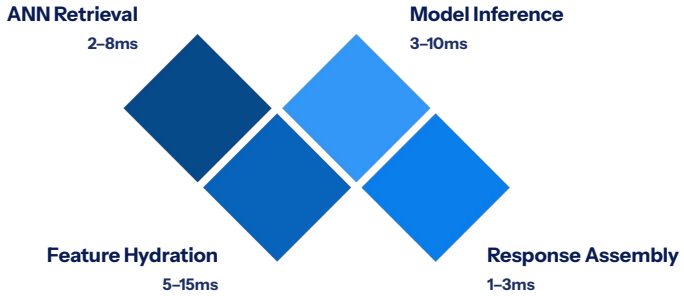
SECTION 4

Real-Time Inference & Operational Reality

Theory meets production.

INFERENCE

Real-Time Inference Constraints



Every millisecond has an owner. Latency budgets must be allocated deliberately.

Feature Hydration

Pre-computed vs. real-time resolution

Inference SLAs

P99 matters more than P50

Fanout Constraints

Parallel retrieval multiplies latency and cost

DISTRIBUTED SYSTEMS

Caching, Fanout & Async Pipelines

One slow dependency can break SLA.

01

Request Layer

Load Balancer → Request Shaping → Timeout Budgets

02

Async Fanout

Vector Store, Feature Store, Business Rules

03

Caching Layer

Pre-computed, Session Cache, Embedding Cache

Partial Degradation

Return cached results when budgets slip.

Request Shaping

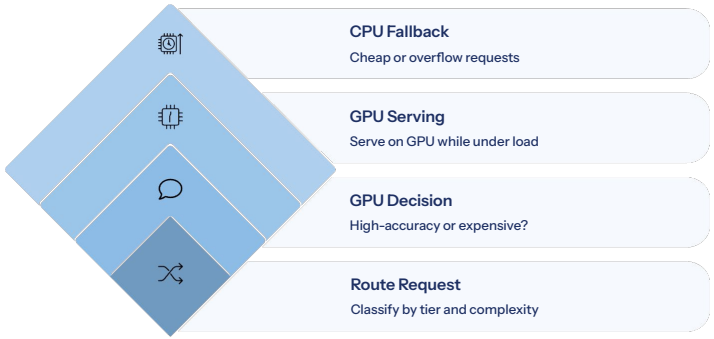
Shed low-priority requests before saturation.

Graceful Fallbacks

Use pre-ranked slates during degradation.

INFERENCE COST

GPU/CPU Blending & Traffic Routing



Inference Cost Optimization

Route cheap requests to CPU, expensive ones to GPU

Model Tiering

Lightweight models handle high-volume, low-stakes requests

GPU Saturation

Sheer overflow to CPU fallback

DEBUGGING

Online/Offline Skew

Training environments are cleaner than reality.

Stale Features

Training features may be hours old

Distribution Mismatch

User behavior and catalogs keep evolving

Schema Drift

Training pipelines lag production changes

Serving Inconsistencies

Training and serving logic can differ

Replay Limitations

Offline replay misses real-time context

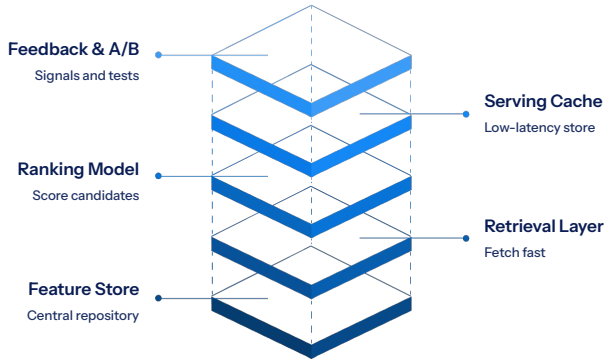
Feedback Loop Amplification

Feedback loops can amplify skew

OBSERVABILITY

Debugging Adaptive Systems

Adaptive systems that cannot be debugged lose trust.



Attribution Ambiguity — Root cause is hard to isolate

Ranking Regressions — Drops may start upstream

Hidden Coupling — Shared stores and caches

Observability Gaps — Inference logging is hard

Causal Uncertainty — Needs instrumentation and holdouts

SECTION 5

Evaluation Systems That Don't Lie

Measure what matters in production recommenders.

EVALUATION

CTR Is Not Enough

Clicks are easy. Outcomes are hard.

High CTR

Low Retention

Click Bait

Erodes Trust

Popular Picks

Starve Long-tail

Clicks ≠ Value

Diverging Signals

Engagement Traps — High CTR can hurt retention

Retention Degradation — Click optimization erodes trust over time

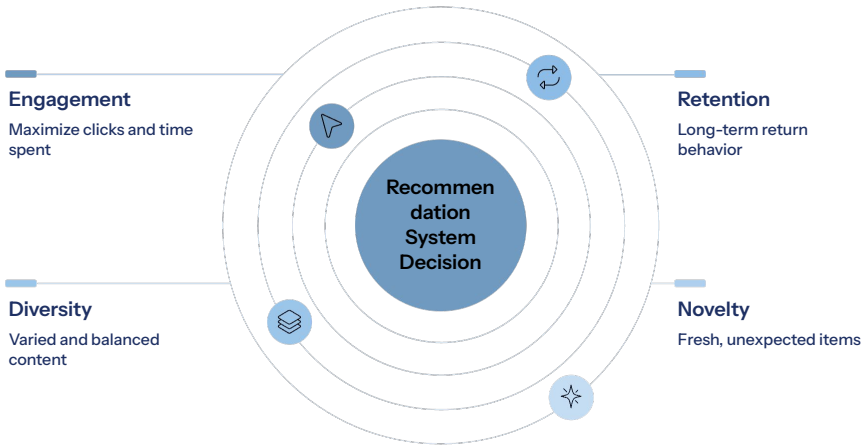
Ecosystem Health — Popular picks starve long-tail creators

Satisfaction vs. Clicks — Clicks and value often diverge

EVALUATION

Multi-Objective Optimization

There is no universally correct objective function.



Engagement — Maximize clicks and time spent

Retention — Long-term return behavior

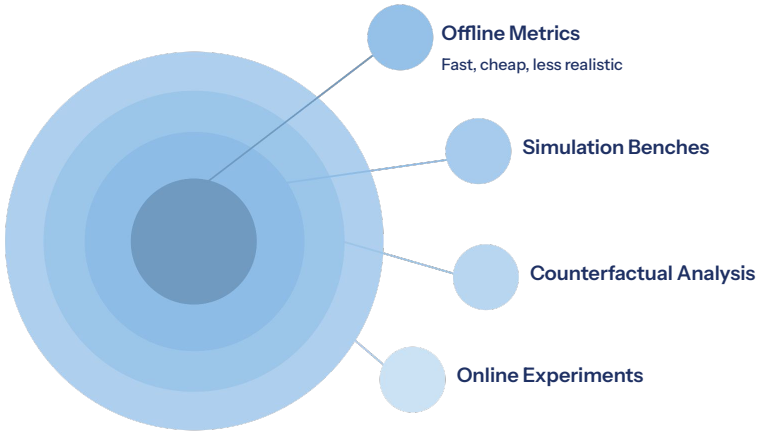
Diversity — Varied and balanced content

Novelty — Fresh, unexpected items

Every objective trade-off is a product decision disguised as a technical one.

EVALUATION

Offline Metrics vs Online Reality



Offline metrics are necessary but insufficient — they miss real behavior

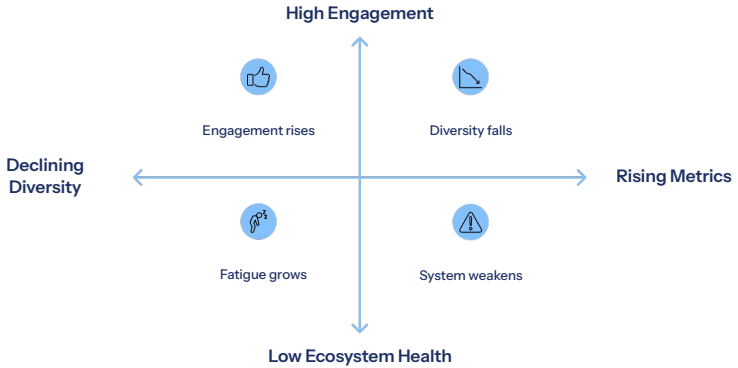
Counterfactual analysis bridges offline and online via logged decisions

LLM-as-judge is a scalable proxy for human signals

FAILURE STORY

When Metrics Improved but the System Weakened

Adaptive systems optimize exactly what you teach.



Engagement Improved — Dashboards looked strong

Diversity Collapsed — Winners crowded out exploration

Fatigue Increased — Repetition eroded satisfaction

Ecosystem Narrowed — Long-tail content shrank

SECTION 6

Migration & Organizational Scaling

Practicality and leadership maturity — shipping adaptive systems at scale.

MIGRATION

Migrating from Classical to AI-Native Systems

Most migrations fail operationally before they fail technically.

- Classical Rules Engine
- Shadow Traffic
- Progressive Rollout
- Dual Writes
- Embedding Bootstrap
- AI-Native System

Shadow Traffic — Score without serving results

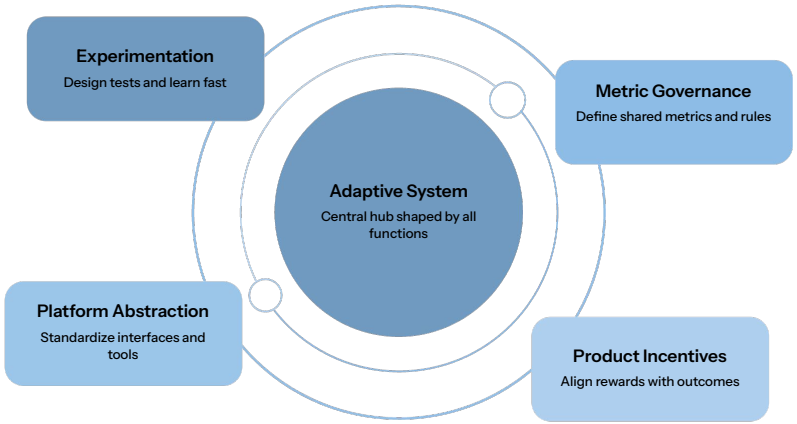
Progressive Rollout — Ramp traffic 1% → 100%

Safe Degradation — Define fallback behavior first

ORG DESIGN

Adaptive Systems Are Organizational Systems

Adaptive systems amplify organizational structure.



Experimentation Ownership — Who decides testing and interpretation?

Metric Governance — Shared metrics prevent teams competing

Platform Abstraction — Without shared infra, teams rebuild primitives

Product Incentives — Misaligned incentives harm the whole system

Coordination Complexity — Org charts miss system coupling

7 SECTION 7

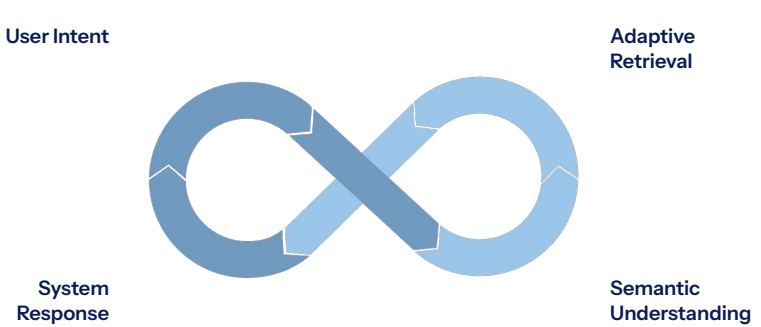
Future Evolution & Closing

Where adaptive recommenders are headed and why it matters.

FUTURE

From Ranking to Collaborative Adaptation

Adaptive systems are becoming collaborative systems.



User Steering

Conversational Systems

Adaptive Retrieval

Semantic Understanding

Memory Systems

Orchestration Layers

Final Thesis

AI's future depends less on model intelligence and more on systems that adapt, evaluate, and evolve in production.

Thank you.

Thank you

Q&A

Mallika Rao.

linkedin.com/in/mallikarao

Medium