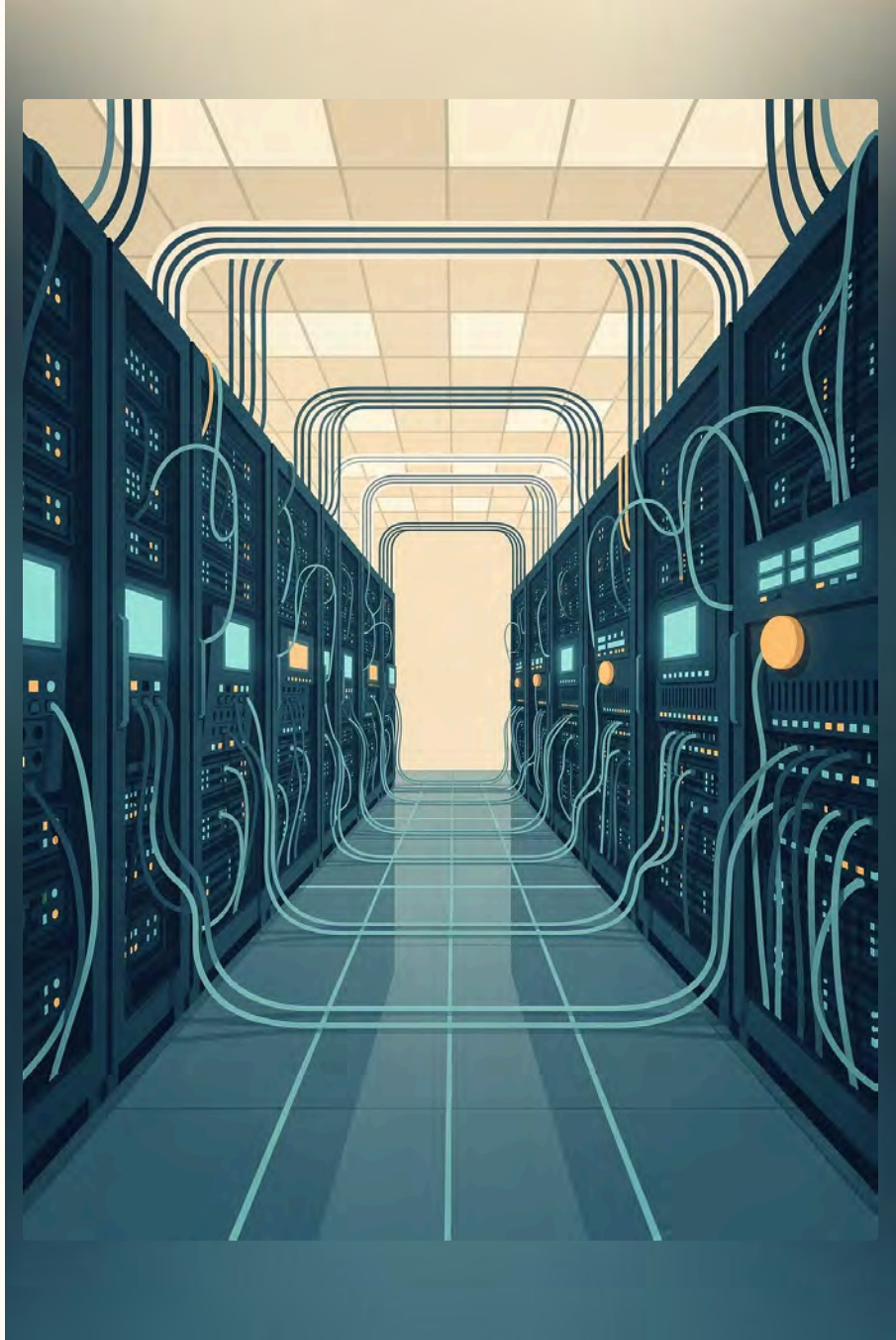
Designing Compliant Digital Onboarding Cloud-Native Architectures for Trustworthy, Audit-Ready Financial Services

MANASA UPPULA

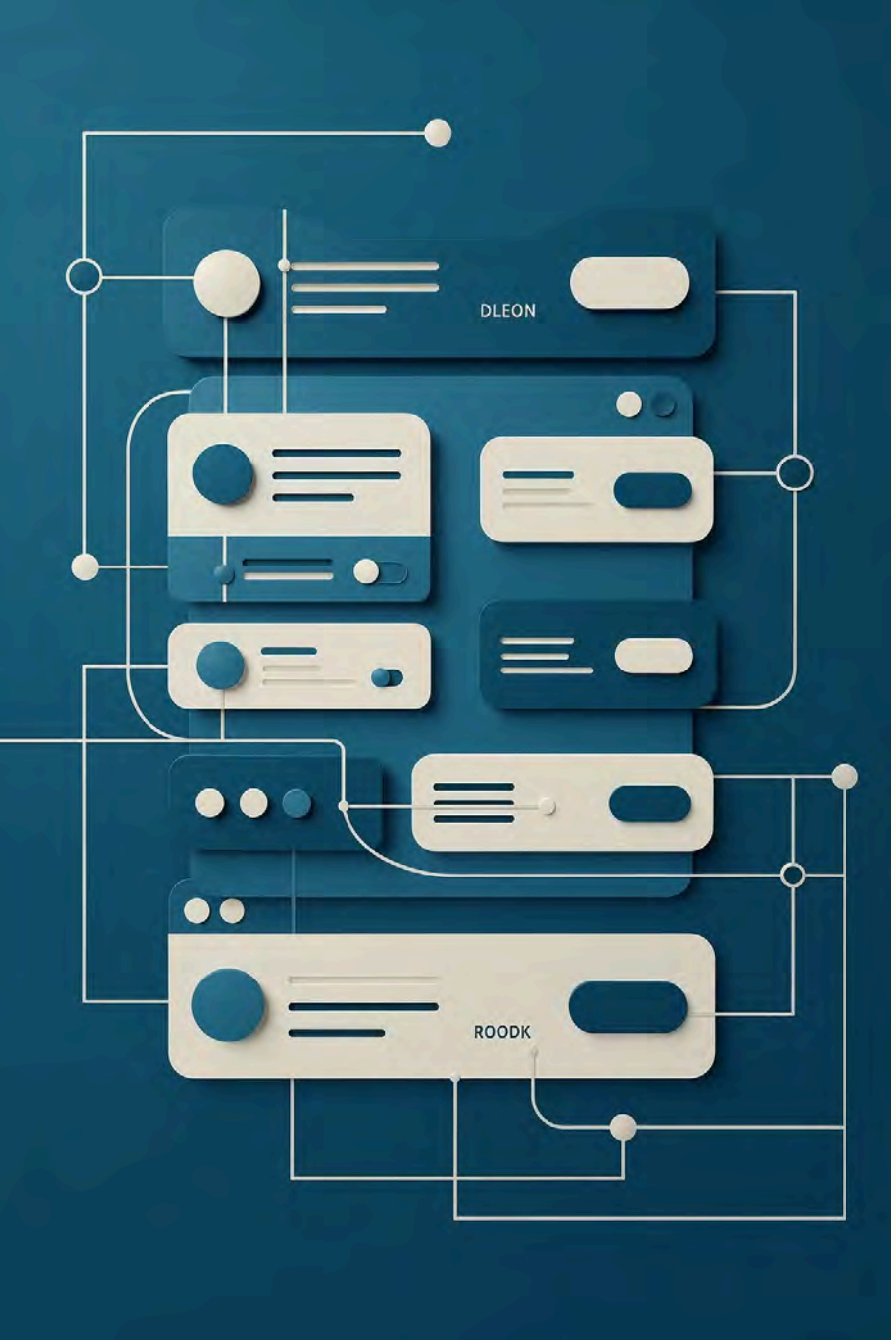
CONF42 CLOUD NATIVE 2026

INDEPENDENT RESEARCHER, USA



Speaker Bio

Manasa Uppula is an enterprise web architect with nearly 14 years of experience designing and modernizing large-scale digital platforms in regulated financial environments. She specializes in audit-ready frontend architectures, evidence-centric interaction design, and translating complex regulatory requirements into clear architectural constraints. Manasa brings a field-level perspective on building customer-facing systems that are compliant, explainable, and built to last.



The Stakes: Why Onboarding Is Different

Legal Relationship

Establishes consent boundaries and grants access to regulated services

UX + Legal Obligation

Simultaneously a design challenge, a compliance obligation, and a governance problem

Real Consequences

Errors here aren't cosmetic, they carry regulatory and legal consequences

The Regulatory Landscape

What Regulators Expect

Not only that data was collected, but **how**, **when**, and **by whom**.

A complete audit trail is essential: what was presented, what was consented to, and the exact system state at the moment of interaction.

Core Frameworks

KYC

Customer identity verification requirements

AML

Anti-money laundering controls and monitoring obligations

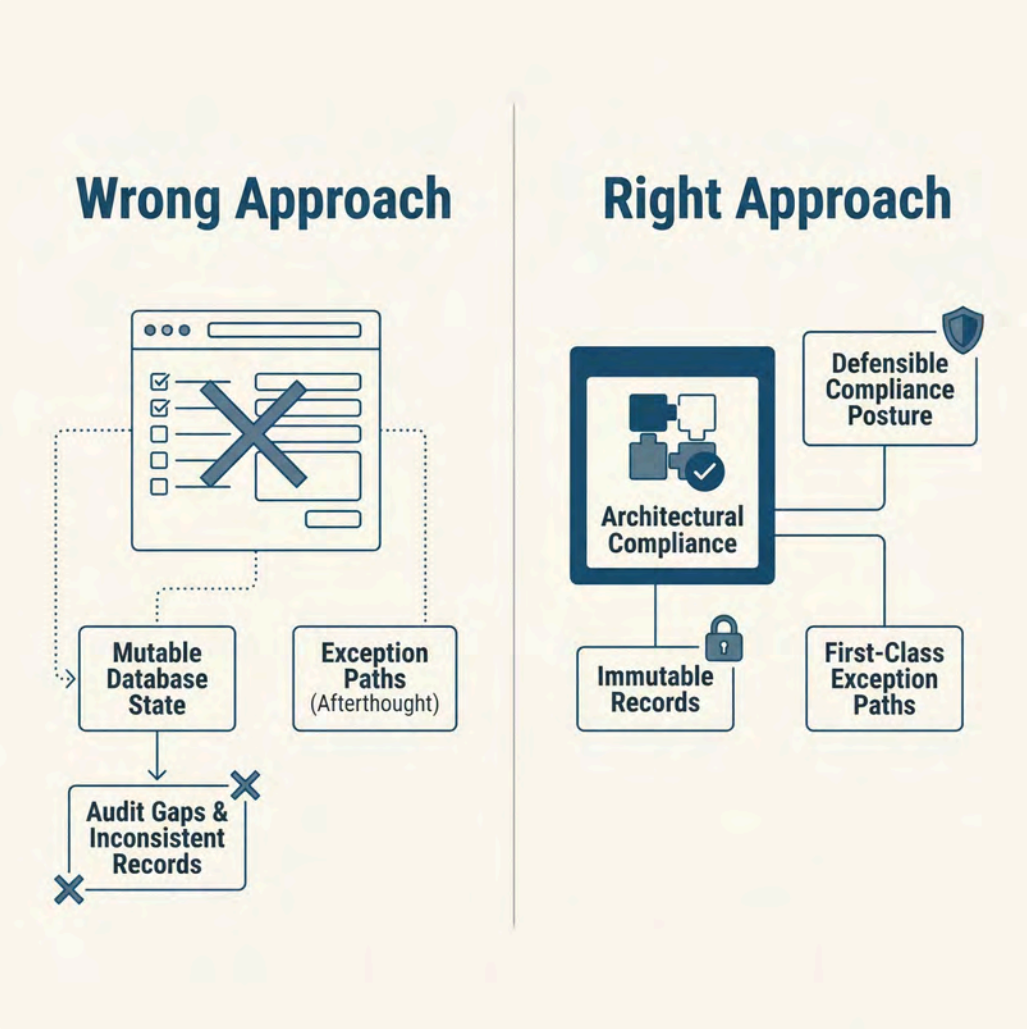
GDPR / Privacy Laws

Requirements under GDPR, CCPA, and related privacy laws

CFPB

Consumer Financial Protection Bureau oversight, including UDAAP and fair lending obligations

The Core Problem: Surface-Layer Compliance Fails



Common Failure Mode

- Compliance treated as a UI layer (checkboxes, modals, form validations)
- Consent data stored as **mutable** database state
- Exception paths designed as afterthoughts

The Consequences

- Audit gaps and inconsistent records
- Distributed services diverging in onboarding state
- Indefensible compliance posture under scrutiny

The Fix: Embed compliance into the architecture itself not applied on top of it.

Introducing: Interaction Integrity

Definition: The property of onboarding flows in which every material interaction is structured, versioned, and evidence-producing.

1

Consent & Disclosures

Formal compliance artifacts versioned, timestamped, immutable

2

Identity Assurance

Progressive, risk-calibrated enforcement across service boundaries

3

Exception Handling

Interaction continuity and auditability under failure conditions

Pillar 1: Consent & Disclosures as Compliance Artifacts

The Problem with Current Practice

Consent is captured as thin UI logic disconnected from audit infrastructure and legally fragile.

The Compliant Design

Each disclosure and consent interaction is a formally defined artifact carrying:

- Version of the disclosure presented
- Timestamp of presentation
- User's explicit response
- System state at the time of interaction

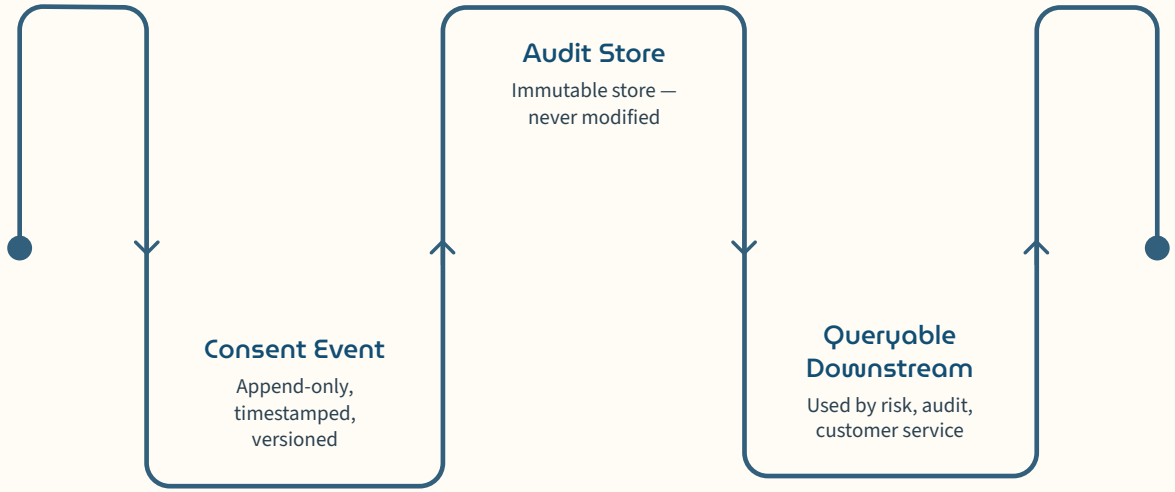
APPEND-ONLY EVENT LOG

Building an Immutable Consent Record

> Consent transforms from a UX checkpoint into a **durable compliance record**.

Versioned Disclosure Content

Disclosure content is managed as uniquely addressable versioned artifacts. Regulatory updates are tracked without overwriting prior versions.



Append-Only Event Log

Consent events are emitted as immutable records never modified, only appended. Any historical interaction can be resolved to the exact disclosure text shown at that moment.

Pillar 2: Progressive Identity Assurance

Identity verification is not binary it is **risk-calibrated** and incremental.

Low-Risk Access

Customers engage with limited features while identity is being established

Incremental Verification

Evidence gathered progressively across the onboarding journey

Full Capability Unlocked

Higher-risk features enabled only with stronger verification evidence

In microservices architectures, identity state must be **consistently communicated across service boundaries** without this, services act on stale or inconsistent state, creating policy gaps.

Identity Assurance as a First-Class Attribute

Principled Propagation

Declared Context

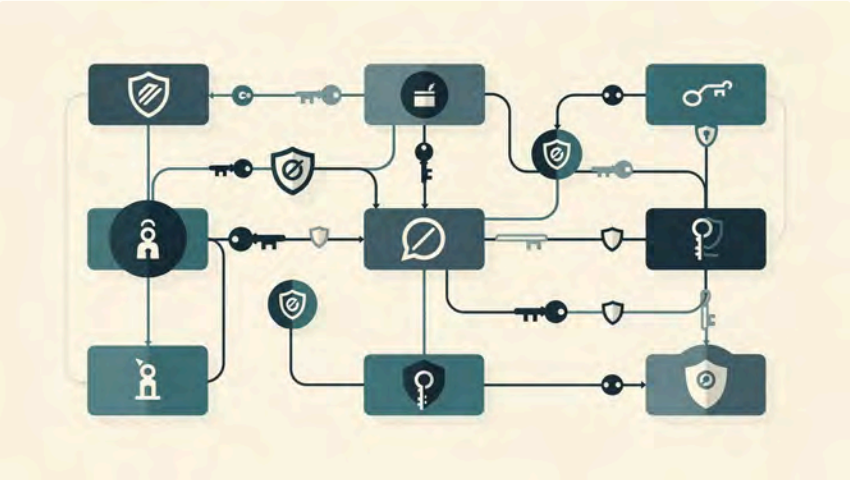
Assurance level is a first-class attribute of the auth context propagated explicitly across all service calls

Observable State Change

Successful verification emits an observable event; downstream services consume and update enforcement posture

Interface Co-Design

UI must faithfully reflect the underlying assurance model mismatches create support burden and erode trust



Pillar 3: Exception Handling & Interaction Continuity

Production Realities

- Network interruptions and session timeouts
- Third-party identity verification failures
- User abandonment mid-flow

The Compliance Risk

Loosely coupled services diverge in their understanding of state when a flow is interrupted. Partial completions create **legal ambiguity** about the customer relationship.

Example Failure Scenario

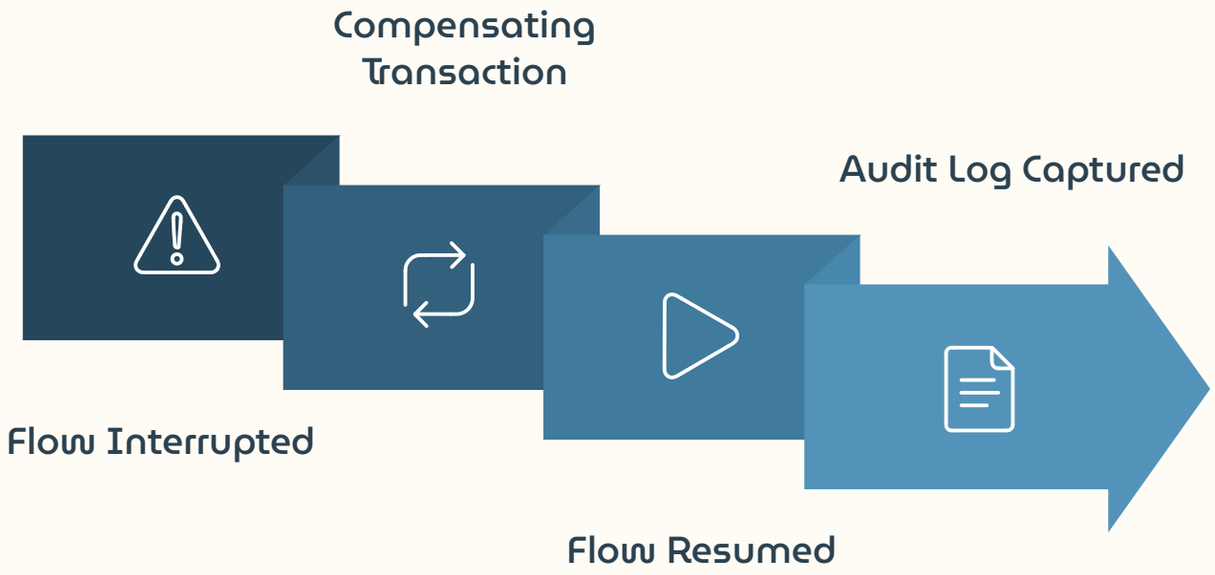
Identity data collected →

Required disclosure **NOT shown** due to downstream service failure →

Legally undefined onboarding state

This is not a UX edge case. It is a compliance incident.

Designing for Continuity: Sagas & Orchestration



Recommended Patterns

Audit Requirements for Exceptions

Capture the fact of interruption, its cause, and the resolution taken. Every exception path must be **explainable and defensible**.

Saga Pattern

Coordinates multi-step distributed workflows with compensating transactions

Process Orchestration

Maintains consistency even under failure conditions across services

What Audit-Ready Architecture Looks Like

Layer	Design Decision	Cloud-Native Example
Consent	Versioned, immutable, append-only event records	Apache Kafka (append-only event log), AWS S3 with Object Lock
Identity	Assurance level carried in propagated auth context	AWS Cognito / OAuth 2.0 JWT claims, SPIFFE/SPIRE for workload identity
Orchestration	Saga-based, exception-aware workflow management	AWS Step Functions, Temporal.io, or Conductor for saga workflows
Audit Log	Captures successes, failures, and resolutions with equal rigor	AWS CloudTrail, OpenTelemetry + Elasticsearch, or Datadog
Disclosure Content	Versioned, resolvable artifacts never overwritten	S3 versioned buckets, Git-backed content store, or AWS DynamoDB Streams

REAL-WORLD SCENARIO

Digital Bank Customer Onboarding

A customer opens a savings account in the app, and the compliance trail starts right away.

Pillar 1 — Consent & Disclosures

The customer reviews the Terms of Service and GDPR disclosure. Every action is captured as an immutable event in Apache Kafka, then stored in S3 with Object Lock. If a regulator asks what the customer consented to, the answer is immediate.

Pillar 2 — Progressive Identity Assurance

Email verified → low-risk access granted (view-only). ID document uploaded → assurance level updated in JWT claims → funds transfer unlocked. Each state change emits an event, so no service uses old identity data.

Pillar 3 — Exception Handling

When the third-party KYC provider times out, the Saga orchestrator steps in with a compensating transaction. The session pauses, partial state is kept, and the customer is notified. The cause and fix are both logged, so there is no audit gap.

The outcome: A clear, defensible onboarding record because compliance is built into the architecture, not added later.

Governance for Long-Term Compliance Resilience

Cloud-native systems are designed for change, compliance must survive it.

Key Risks as Systems Evolve

- Disclosure artifacts must remain resolvable across service updates
- Identity context must be consistently interpreted across service versions
- Exception handling must be verifiable as infrastructure changes

What Governance Requires

→ Compliance constraints treated as **explicit architectural inputs** documented, versioned, governed

→ Not assumed to be preserved implicitly by individual teams

→ Architectural decisions must be **legible** enough to inform future ones

Key Takeaways

1 Compliance must be structural

Not a surface layer applied to existing architecture, embedded from the ground up

2 Consent is a record, not a checkbox

Version it, timestamp it, and make it immutable

3 Identity assurance is continuous

Design it as a first-class, propagated attribute across all services

4 Exceptions are compliance events

Audit them with the same rigor as success paths

5 Governance must evolve with the system

Compliance integrity is not a one-time achievement

Thank you!

Manasa Uppula
Conf42 Cloud Native, April 23 2026.