Compliance AI Content Generation Platform

1. Problem We Are Solving

1.1 Background

Fintech and insurance organizations must generate large volumes of content such as marketing material, product descriptions, policy explanations, and digital campaigns. At the same time, they operate under strict regulatory and legal frameworks (e.g., IRDAI guidelines, financial advertising norms, internal brand and SEO rules).

This creates a constant conflict:

- Business teams require **speed and scale** in content creation
- Compliance teams require **accuracy, control, and auditability**

1.2 Limitations of Traditional Content Workflows

Most organizations follow a reactive workflow:

1. Content is written manually or generated using generic AI tools
2. Content is sent to compliance or legal teams for review
3. Violations are identified
4. Content is rewritten and re-reviewed

This approach leads to:

- Slow turnaround times
- Multiple review cycles
- High operational costs
- Inconsistent compliance decisions
- Limited or no deterministic audit trail

1.3 Risks of Generic AI Content Generation

Generic AI content generation tools are not designed for regulated domains. They:

- Can generate legally risky claims (e.g., “guaranteed returns”)
- Lack awareness of jurisdiction-specific regulations
- Produce non-deterministic outputs

- Do not provide explainable or auditable decisions

As a result, organizations face increased regulatory, legal, and brand risks.

1.4 Core Problem Statement

The core problem is not content generation alone.

The real challenge is:

How can organizations generate content at scale while ensuring every piece of content is compliant by design, explainable, auditable, and governed by human-defined rules — without relying on slow and expensive manual review cycles?

---

2. Solution Overview

The proposed solution is a **Compliance-First AI Content Generation Platform** that enables organizations to generate content at scale while ensuring strict regulatory, legal, and brand compliance.

Instead of generating content first and fixing violations later, the platform **embeds compliance directly into the content generation workflow**.

2.1 What the Platform Does

The platform:

- Generates marketing and informational content using AI
- Enforces **human-defined compliance rules** before and after generation
- Produces **deterministic, explainable, and auditable outcomes**
- Reduces reliance on slow and expensive manual compliance reviews

---

2.2 How the Solution Works

User submits a prompt or uploads content

1. The system enhances the prompt with compliance constraints
2. Active compliance rules are loaded from a rule engine
3. AI generates content within enforced boundaries

4. Generated content is reviewed and validated against rules
5. Final output is approved, blocked, or auto-corrected with clear explanations

---

2.3 Key Differentiators

- **Compliance-by-Design:** Rules shape content before it is generated
- **Rule-First Architecture:** Deterministic rules always override AI
- **Explainability:** Every decision references rule IDs and versions
- **Audit-Ready:** All actions are logged and traceable
- **Cost-Efficient:** Reduces repeated human review cycles

2.4 Outcome

The platform allows organizations to:

- Safely adopt AI for content generation in regulated domains
- Maintain full regulatory control and governance
- Scale content creation without scaling compliance risk or cost

---

3. Technology Stack & Approach

3.1 Architectural Approach

The platform follows a **compliance-first, rule-driven architecture** where deterministic systems govern all decisions and AI models are used strictly as language and analysis assistants.

Key architectural principles:

- **Multi-model AI orchestration** (generation + review)
- **Rule-first enforcement** with deterministic outcomes
- **Separation of authority** between rules, context, and AI
- **Auditability by design**

3.2 Multi-Model AI Strategy

The system uses multiple AI models, each with a narrowly defined responsibility:

- **Content Generator Model**
  Generates human-readable content within predefined constraints.
- **Strict Reviewer Model**
  Performs conservative risk analysis and outputs structured compliance signals.

No model has approval or enforcement authority.

3.3 Rule Engine

- Human-authored, versioned compliance rules
- Supports hard, soft, and conditional rules
- Enforced deterministically in code
- Rules override AI output at all times

3.4 Deep Analysis Layer (Pre-Enforcement)

- Breaks content into atomic claims
- Tags claims with risk and regulatory categories
- Maps claims to relevant regulations using semantic retrieval
- Feeds structured signals into the rule engine

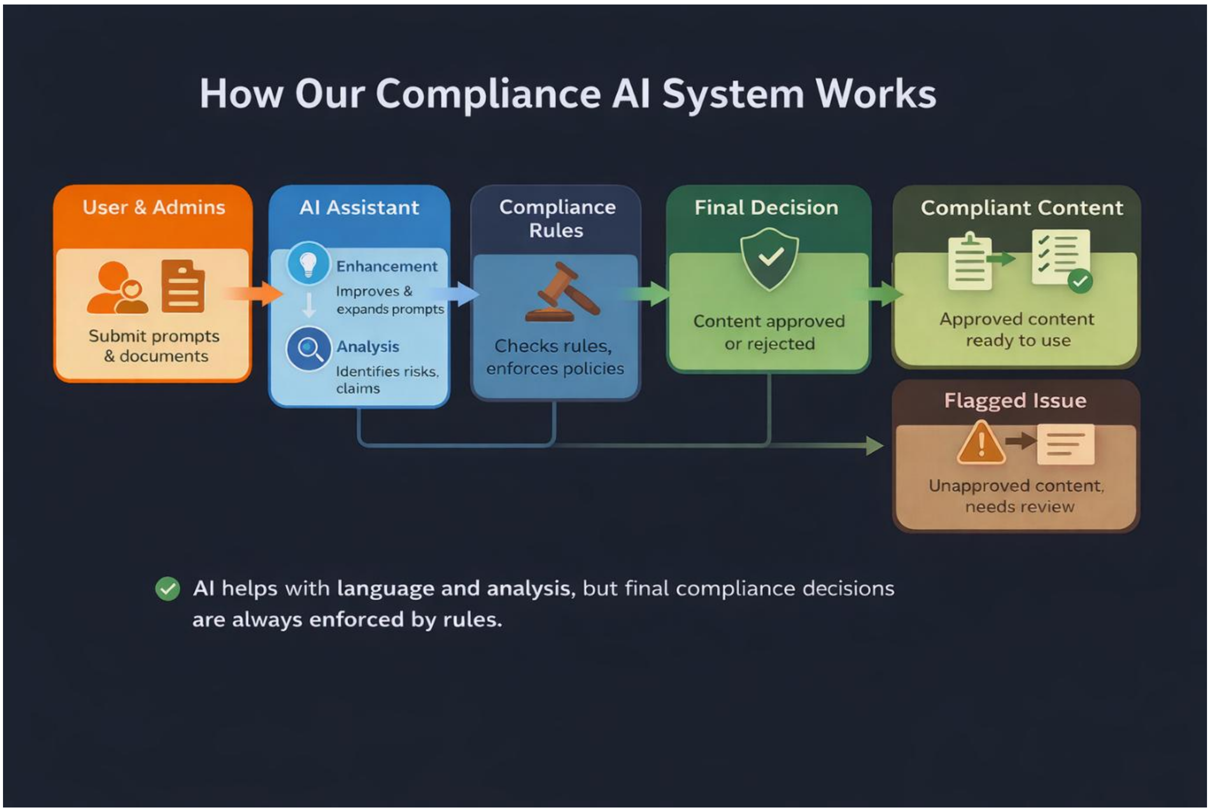
3.5 Data & Storage Stack

- **SQL Database (PostgreSQL):**
  Source of truth for rules, decisions, audit logs
- **Vector Database (OpenSearch):**
  Semantic retrieval of regulatory context only
- **Object Storage (S3):**
  Documents, generated content, and artifacts

3.6 Cloud & Infrastructure

- **AWS Bedrock:** LLM access and embeddings

- **AWS Lambda / ECS:** Orchestration and services
- **AWS IAM & Cognito:** Authentication and authorization
- **AWS CloudWatch:** Logging, monitoring, audit trails



How Our Compliance AI System Works

User & Admins

Submit prompts & documents

AI Assistant

Enhancement

Improves & expands prompts

Analysis

Identifies risks, claims

Compliance Rules

Checks rules, enforces policies

Final Decision

Content approved or rejected

Compliant Content

Approved content ready to use

Flagged Issue

Unapproved content, needs review

✓ AI helps with **language and analysis**, but final compliance decisions **are always enforced by rules.**

---

4. AWS Costing & Billing Analysis

4.1 AWS Services Used – Why and How They Are Billed

4.1.1 Amazon Bedrock (LLMs and Embeddings)

Why Used

- Managed access to multiple foundation models
- No model hosting, scaling, or training overhead
- IAM-controlled, enterprise-ready inference

Usage in This Platform

- Content generation model
- Strict compliance reviewer model
- Titan Text Embeddings for regulatory RAG

How AWS Bills

- Per 1,000 input tokens
- Per 1,000 output tokens
- Embeddings billed per 1,000 tokens embedded

Typical Pricing Ranges (Model Dependent)

- Input tokens: ~$0.0001 – $0.001 per 1K tokens
- Output tokens: ~$0.0004 – $0.003 per 1K tokens

Reference https://aws.amazon.com/bedrock/pricing/

Note: Exact pricing varies by model and region. The POC intentionally keeps models interchangeable to avoid vendor lock-in.

4.1.2 Amazon OpenSearch Service (Vector Database)

Why Used

- AWS-native vector search for regulatory grounding (RAG)
- Semantic similarity and metadata filtering
- Clean integration with Amazon Bedrock

What Is Stored

- Embeddings of regulatory clauses only
- No user data, decisions, or enforcement logic

How AWS Bills (Serverless)

- OpenSearch Compute Units (OCUs) per hour
- Vector storage (GB per month)

Important Cost Behavior

- Serverless collections may incur a baseline OCU cost even when idle
- This makes OpenSearch a semi-fixed cost at low traffic

References https://docs.aws.amazon.com/opensearch-service/latest/developerguide/serverless-overview.html https://aws.amazon.com/opensearch-service/pricing/

4.1.3 AWS Lambda (Orchestration)

Why Used

- Prompt enhancement
- Rule evaluation
- Decision aggregation

How AWS Bills

- Per invocation
- Per GB-second of execution time

Cost Characteristics

- Negligible at low to medium scale
- Linear growth with usage

Reference https://aws.amazon.com/lambda/pricing/

4.1.4 Amazon RDS (PostgreSQL)

Why Used

- Source of truth for compliance rules
- Storage of compliance results
- Audit logs and decision history

How AWS Bills

- Instance-hours
- Storage (GB per month)
- I/O operations

POC Guidance

- Small instance (e.g., db.t3.small) is sufficient
- Can be reserved later for cost optimization

Reference https://aws.amazon.com/rds/postgresql/pricing/

4.1.5 Amazon S3 (Object Storage)

Why Used

- Uploaded documents
- Generated content artifacts
- Long-term audit storage

How AWS Bills

- Storage per GB per month
- PUT / GET request counts

Reference https://aws.amazon.com/s3/pricing/

4.1.6 Amazon CloudWatch and IAM

CloudWatch

- Logs and metrics billed by ingestion and retention
- Used for auditability, monitoring, and observability

IAM

- No direct cost
- Required for access control and security

Reference https://aws.amazon.com/cloudwatch/pricing/

4.2 Usage Assumptions (POC Baseline)

The following assumptions are used to estimate costs:

- 1 request = 1 content generation or 1 document validation
- Average enhanced prompt size: ~1,500 tokens
- Average generated output: ~800 tokens
- Reviewer model usage: ~1,200 input / 200 output tokens
- Regulatory documents embedded once and amortized

- Region: ap-south-1 (India)

4.3 Cost Scenarios (Normalized per 1,000 API Calls)

To make costing concrete and easy to reason about, the following estimates normalize usage to **1,000 API calls**. This helps stakeholders directly understand cost per unit of usage, independent of scale.

An API call is defined as **one content generation or one document compliance check**.

4.3.1 Cost per 1,000 API Calls (Baseline)

LLM Usage Assumptions per 1,000 Calls

- Prompt (after enhancement): ~1,500 tokens
- Generated output: ~800 tokens
- Compliance reviewer: ~1,200 input / 200 output tokens
- Total LLM tokens per call: ~3,700 tokens
- Total tokens per 1,000 calls: ~3.7 million tokens

Estimated Cost Breakdown per 1,000 API Calls

- **Amazon Bedrock (Generation + Review)**: $14 – $18
- **Amazon OpenSearch (Vector RAG)**: $4 – $6
- **AWS Lambda (Orchestration)**: $1 – $2
- **Amazon RDS (Rules, Results, Audit Logs)**: $1 – $2
- **Amazon S3 (Documents & Artifacts)**: <$1
- **CloudWatch (Logs & Metrics)**: ~$1

Estimated Total per 1,000 API Calls: $20 – $21

4.3.2 Monthly Cost Mapping (Using 1,000 API Call Unit)

Monthly API Calls	Estimated Monthly Cost
1,000	$20 – $21

10,000	$200 – $210
50,000	$1,000 – $1,050
100,000	$2,000 – $2,100

4.3.3 Why This Model Is Predictable

- Cost scales **linearly** with API usage
- No hidden fixed AI training or fine-tuning costs
- Token usage is controlled by prompt enhancement and rule short-circuiting
- Vector embeddings are amortized, not re-generated per request

This makes cost forecasting straightforward for finance and leadership teams.

Estimated Total: $75 – $200 per month

4.3.4 Typical Usage (Pilot Deployment)

- ~10,000 requests per month
- Regular generation and review

Estimated Monthly Cost

- Bedrock: $150 – $350
- OpenSearch: $150 – $300
- Other infrastructure: $50 – $100

Total: $350 – $750 per month

4.3.5 Maximum Usage (Production-Like)

- ~100,000 requests per month
- Bedrock usage dominates

Estimated Monthly Cost

- Bedrock: $1,500 – $3,500

- OpenSearch: $300 – $800
- Other infrastructure: $100 – $200

Total: $2,000 – $5,000 per month

4.4 Cost Control and Optimization Levers

- Prompt enhancement reduces retries and token waste
- Rule-only short-circuiting avoids unnecessary LLM calls
- Smaller reviewer models for compliance checks
- Caching embeddings and enhanced prompts
- Periodic cleanup and cold storage in S3

4.5 References

- AWS Bedrock Pricing: https://aws.amazon.com/bedrock/pricing/
- OpenSearch Serverless Overview: https://docs.aws.amazon.com/opensearch-service/latest/developerguide/serverless

---

Technology Stack & Architectural Approach

---

1. Architectural Approach

The platform follows a **compliance-first, rule-driven architecture** where deterministic systems have final authority and AI models are used only as controlled assistants.

Key characteristics:

- Rule-first enforcement (rules override AI)
- Multi-model AI orchestration (generation + review)
- Clear separation of authority (rules, context, AI)
- Auditability and governance by design

Layer	**Technology**	**Purpose**
Backend Services	**FastAPI (Python)**	Core backend APIs, orchestration logic, rule engine, compliance workflows
AI Development & Orchestration	**AntiGravity (IDE + Framework)**	Used as a development IDE and orchestration framework for multi-model AI workflows, prompt control, and compliance-safe experimentation
Frontend	**React** (modern UI framework)	User, Admin, and Super Admin interfaces
API Layer / BFF	**Node.js**	Frontend-backend communication and API aggregation
Containerization	**Docker**	Initial local development, testing, and environment consistency before cloud deployment
AI Models	**Amazon Bedrock**	Content generation models and strict compliance reviewer models
Embeddings	**Titan Text Embeddings (Bedrock)**	Regulatory grounding and semantic retrieval (RAG)
Vector Database	**Amazon OpenSearch**	Semantic search over regulatory and policy clauses
Relational Database	**PostgreSQL (Amazon RDS)**	Source of truth for rules, compliance decisions, and audit logs
Object Storage	**Amazon S3**	Uploaded documents, generated content, and artifacts
Observability	**Amazon CloudWatch**	Logs, metrics, and audit monitoring
Security & Access	**AWS IAM & Cognito**	Authentication, authorization, and role-based access control