PROJECT OSKAR 2.0

Production-Ready Architecture for Transparent, Scalable Content Moderation

Technical Specification v2.1

Architecture Team

February 21, 2026

Abstract

PROJECT OSKAR 2.0 is a comprehensive content moderation decision-support ecosystem addressing critical gaps in existing systems: explicit uncertainty quantification, network-level intelligence, and ethical governance. This document presents a production-hardened architecture incorporating MLOps governance, privacy-preserving data handling, real-time scalability modeling, adversarial red-team frameworks, and human-cognitive optimization. We introduce novel contributions including Bayesian trust scoring, narrative intelligence tracking, and faithfulness-validated explainability. The specification includes capacity planning, economic modeling, and deployment strategies suitable for enterprise adoption.

Contents

1 Introduction

Content moderation systems face a deployment crisis: research prototypes fail in production due to inadequate MLOps, missing privacy safeguards, and ignorance of human moderator psychology. OSKAR 2.0 addresses these gaps through a seven-layer architecture spanning from infrastructure to behavioral optimization.

1.1 Architecture Evolution

Table 1: OSKAR 2.0 Layer Stack

Layer	Name	Function
7	Behavioral Economics	A/B testing, trust impact, moderator cognition
6	Decision Interface	Pre-post warnings, appeals, dashboards
5	Cognitive Engine	Calibration, uncertainty, risk fusion
4	Multimodal Intelligence	Text, image, audio, context graphs
3	Knowledge Infrastructure	Versioned graphs, caching, privacy
2	MLOps & Governance	Model registry, drift detection, canary deploys
1	Security & Compliance	Encryption, RBAC, audit, disaster recovery

2 Layer 1: Security, Privacy & Compliance

2.1 Data Privacy Architecture

2.1.1 PII Handling Strategy

- **Detection**: Microsoft Presidio / Google DLP API for automatic PII identification
- **Anonymization**: Hash user IDs with HMAC-SHA256 (platform-specific keys)
- **Pseudonymization**: Replace usernames with consistent tokens per analysis session
- **Redaction**: Automatic removal of emails, phone numbers, addresses from logs

2.1.2 Data Retention Policy

Table 2: Data Lifecycle Management

Data Type	Retention	Encryption	Access
Raw content	90 days	AES-256-GCM	System only
Feature embeddings	1 year	AES-256-GCM	Model training
Decision logs	7 years	AES-256-GCM	Audit, legal
Moderator actions	3 years	AES-256-GCM	HR, performance
Model checkpoints	Indefinite	AES-256-GCM	MLOps team

2.1.3 Encryption Layers

```
# Encryption in Transit
TLS 1.3 for all API communications
mTLS for inter-service communication

# Encryption at Rest
```

```
Database: AES-256-GCM with platform-managed keys
Object Storage: Client-side encryption before upload
Backups: Encrypted with separate key hierarchy

# Key Management
AWS KMS / Azure Key Vault / HashiCorp Vault
Automatic key rotation every 90 days
HSM-backed root keys
```

2.2 Access Control Model

2.2.1 Role-Based Access Control (RBAC)

Table 3: RBAC Matrix

Role	Permissions	Scope
System Admin	Full infrastructure access	Deployment, scaling, security config
MLOps Engineer	Model registry, training pipelines	Cannot access raw user content
Moderator Supervisor	Override decisions, view analytics	Platform-specific, time-bounded
Content Moderator	Review flagged content, appeal decisions	Cannot see model internals
Auditor	Read-only logs, bias reports	Cross-platform, compliance-focused
API Consumer	POST /analyze endpoint	Rate-limited, scoped to organization

2.3 Disaster Recovery

- **RPO (Recovery Point Objective)**: 5 minutes (continuous replication)
- **RTO (Recovery Time Objective)**: 15 minutes (automated failover)
- **Backup Strategy**: Cross-region replication, daily snapshots, 7-year archive
- **Chaos Engineering**: Monthly simulated region failures, botnet attacks

3 Layer 2: MLOps & Model Governance

3.1 Model Registry Structure

```
model_registry/
        production/
             gemma_base_v2.1.0/          # Base model, frozen
             hate_lora_2026_02_15_r16/   # LoRA adapter, weekly
   ↪ update
             claim_classifier_v3.2.1/    # Claim type model
             verification_nli_v2.1.0/    # NLI verification
             bot_gnn_v1.5.0/             # Graph neural network
        staging/
             [candidate models]
```

```
            shadow/
                  [A/B test variants]
            archived/
                [deprecated versions]

model_registry.json:
{
  "production": {
    "hate_speech": {
      "version": "hate_lora_2026_02_15_r16",
      "base_model": "gemma_base_v2.1.0",
      "training_data_hash": "sha256:abc123...",
      "validation_f1": 0.923,
      "deployment_date": "2026-02-15T00:00:00Z",
      "rollback_threshold": 0.02
    }
  }
}
```

3.2 Deployment Strategy

3.2.1 Canary Deployment Logic

Algorithm 1 Safe Model Rollout

Require: New model M_{new}, Current model M_{curr}, Traffic split $\alpha = 0.05$
1: Deploy M_{new} to 5% traffic (canary)
2: Monitor for 24 hours:
3: **for** each metric $m \in \{\text{latency}, \text{error\_rate}, \text{drift\_score}\}$ **do**
4:     **if** $m_{new} > 1.5 \times m_{curr}$ **then**
5:         **Rollback**: Route 100% to M_{curr}
6:         Alert MLOps team
7:         **exit**
8:     **end if**
9: **end for**
10: Gradually increase: $\alpha \leftarrow 0.25, 0.5, 0.75, 1.0$
11: At each step, monitor for 6 hours
12: Upon full deployment, archive M_{curr}

3.3 Drift Detection

captionDrift Monitoring Dashboard

Drift Type	**Detection Method**	**Threshold**	**Action**
Data drift	Embedding space KL-divergence	$D_{KL} > 0.1$	Trigger retraining
Concept drift	Performance decay on gold set	ΔF1 > 0.03	Immediate roll-back
Feature drift	PSI (Population Stability Index)	PSI > 0.25	Feature engineering
Label drift	Class distribution shift	χ^2 p < 0.01	Resample training data

4 Layer 3: Knowledge Infrastructure

4.1 Versioned Knowledge Graph

Neo4j schema for temporal fact tracking:

```
// Node: Claim
CREATE (c:Claim {
  id: 'claim_12345',
  text: 'Vaccines cause autism',
  first_seen: '2024-01-15',
  embedding: [...],
  version: '2024-06-15'
})

// Node: Evidence
CREATE (e:Evidence {
  source: 'CDC',
  url: 'https://cdc.gov/...',
  verdict: 'REFUTED',
  publication_date: '2023-03-10',
  credibility_score: 0.95
})

// Relationship: Temporal validity
CREATE (c)-[:REFUTED_BY {valid_from: '2023-03-10', valid_to: NULL}]->(e
  ↪ )
```

4.2 Privacy-Preserving Graph Processing

User IDs hashed before graph construction:

$$\text{user\_token} = \text{HMAC-SHA256}(\text{user\_id}, \text{platform\_key}) \tag{1}$$

Graph analysis performed on anonymized tokens; re-identification only possible with platform key.

5 Layer 4: Multimodal Intelligence Core

[Previous Sections 4.1-4.4 maintained with additions:]

5.1 Additions: Narrative Intelligence Layer

Track narrative evolution through time-series graph analysis:

Algorithm 2 Narrative Drift Detection

Require: Claim embedding stream $\{\mathbf{c}_t\}_{t=1}^T$, Time window w

1: Compute topic coherence: $\text{coh}_t = \frac{1}{w}\sum_{i=t-w}^{t}\cos(\mathbf{c}_i, \mathbf{c}_{i-1})$
2: Detect framing shifts: $\Delta_t = \|\mathbf{c}_t - \text{EMA}(\mathbf{c}_{t-w:t})\|$
3: Measure emotional escalation: $e_t = \text{SentimentIntensity}(\text{context}_t)$
4: **if** $\Delta_t > \tau_{\text{shift}} \wedge e_t > \tau_{\text{emotion}}$ **then**
5:     **Alert**: Narrative manipulation detected
6:     Trigger network analysis for coordinated spread
7: **end if**

Polarization Index:

$$\mathcal{P} = 1 - \frac{\text{Between-cluster connectivity}}{\text{Total connectivity}} \tag{2}$$

where clusters are communities in the reply graph.

6 Layer 5: Cognitive Engine

[Previous calibration and uncertainty sections maintained with:]

6.1 Additions: Bayesian Trust Scoring

Maintain longitudinal user trust priors:

$$P(\text{trustworthy}|\text{history}) = \frac{\alpha_0 + \text{correct\_claims}}{\alpha_0 + \beta_0 + \text{total\_claims}} \tag{3}$$

where $(\alpha_0, \beta_0) = (2, 2)$ is the prior. Updates after each verified claim.

Trust-Adjusted Risk:

$$\text{Risk}_{\text{adjusted}} = \text{Risk}_{\text{base}} \times (1.5 - \text{trust\_score}) \tag{4}$$

High-trust users get reduced scrutiny; low-trust users (frequent misinformation sharers) get elevated review.

7 Layer 6: Decision Interface

[Previous sections maintained with:]

7.1 Additions: Moderator Cognitive Optimization

Table 4: Human-Centered Design Features

Feature	**Implementation**
Decision assistance	Top-3 evidence snippets pre-fetched, confidence highlighted
Cognitive load management	High-confidence cases auto-approved; moderators see uncertain cases only
Exposure rotation	Moderators rotated between hate speech, misinformation, bot clusters every 2 hours
Burnout prevention	Daily exposure limits, mandatory breaks, trauma counseling resources
Skill calibration	Periodic gold-set testing; feedback on accuracy vs. model

8 Layer 7: Behavioral Economics & Experimentation

8.1 A/B Testing Framework

```
# Experiment configuration
experiment = {
    "warning_message_variant": ["A", "B", "C"],
    "traffic_split": [0.33, 0.33, 0.34],
    "success_metrics": [
        "repost_rate_reduction",
        "correction_acceptance",
        "appeal_rate",
        "user_satisfaction"
    ],
    "duration": "14_days",
    "min_sample_size": 10000
}
```

8.2 Trust Impact Metrics

- **Warning Efficacy**: % of users who edit post after soft warning
- **Backfire Effect**: % who double-down on false claim (indicates overreach)
- **Platform Trust**: Survey-based Likert scale (quarterly)
- **Churn Rate**: User deletion correlation with flagging

9 Adversarial Robustness: Red Team Framework

9.1 Adversarial Lab Structure

```
adversarial_lab/
          synthetic_bot_generator.py       # LLM-powered bot behavior simulation
          dialect_stress_test.py           # AAVE, Spanglish, Hinglish robustness
          claim_mutation_engine.py         # Semantic-preserving perturbations
          coordinated_attack_sim.py        # Multi-account campaign simulation
          visual_adversarial.py            # OCR-resistant image generation
          evaluation/
              robustness_report.py
              mitigation_effectiveness.py
```

9.2 Stress Test Scenarios

Table 5: Red Team Exercise Calendar

Frequency	**Attack Type**	**Success Criteria**
Weekly	Character-level obfuscation	F1 drop < 5% vs. clean text
Monthly	Coordinated bot swarm (100 accounts)	Detection rate > 95%
Quarterly	Election interference simulation	Harm reduction > 80%
Annually	Full platform penetration test	No critical vulnerabilities

10 Explainability Validation

10.1 Faithfulness Testing

1. **Comprehensiveness**: Does removing highlighted features reduce prediction confidence?
2. **Sufficiency**: Do highlighted features alone reproduce the prediction?
3. **Consistency**: Are explanations stable across similar inputs?

10.2 Counterfactual Evaluation

Generate minimal perturbations that flip predictions:

$$\mathbf{x}_{\text{cf}} = \arg\min_{\mathbf{x}'} \|\mathbf{x} - \mathbf{x}'\| \text{ s.t. } f(\mathbf{x}') \neq f(\mathbf{x}) \tag{5}$$

If counterfactuals are semantically implausible, the model relies on spurious correlations.

11 Scalability & Capacity Planning

11.1 Throughput Modeling

Table 6: Capacity Requirements

Component	**Latency**	**Throughput**	**Infrastructure**
Text analysis (P95)	150ms	10,000 QPS	8x A100 GPUs
Image OCR + analysis	800ms	2,000 QPS	4x A100 + Tesseract cluster
Audio transcription	2s	500 QPS	Whisper large-v3 cluster
Bot detection (GNN)	300ms	5,000 QPS	GPU + CPU hybrid

11.2 Cost Modeling

Per-1M Posts Analyzed:

- Text-only: \$450 (compute) + \$50 (storage) = **\$500**
- With images (30%): \$450 + \$135 (images) + \$50 = **\$635**

- With video (10%): \$450 + \$45 (video) + \$50 = **\$545**

Election Day Spike (10x normal load):

- Auto-scaling to 20x capacity
- Pre-warmed caches for trending claims
- Estimated cost: \$5,000/hour at peak

12 Economic Model

12.1 SaaS Pricing Tiers

Table 7: Pricing Structure

Tier	Volume	Features	Price
Starter	100K posts/month	Text-only, API access	\$499/month
Professional	1M posts/month	+Images, dashboard, email support	\$2,499/month
Enterprise	10M+ posts/month	+Video, on-prem option, SLA, dedicated support	Custom
Government	Unlimited	+Classified air-gap, FedRAMP, audit support	Custom

12.2 On-Premise Licensing

- **License**: Annual subscription per node
- **Support**: 24/7 critical, business-hours standard
- **Updates**: Quarterly feature, monthly security
- **Custom training**: Additional fee per domain adaptation

13 Observability & Monitoring

13.1 Three-Pillar Observability

Table 8: Monitoring Stack

Pillar	Tools	Key Metrics
Metrics	Prometheus + Grafana	Latency, throughput, error rates, GPU utilization
Logs	ELK Stack (Elasticsearch)	Decision audit trails, security events, errors
Traces	Jaeger / OpenTelemetry	Request latency breakdown, dependency mapping

13.2 Alerting Thresholds

```
alerts:
  - name: high_latency
    condition: p95_latency > 500ms
    duration: 5m
    severity: warning

  - name: model_drift
    condition: validation_f1 < 0.85
    duration: 0m
    severity: critical

  - name: bot_surge
    condition: bot_detection_rate > 5x baseline
    duration: 10m
    severity: critical
    action: trigger_incident_response
```

14 Conclusion

OSKAR 2.0 represents a production-ready evolution from research prototype to enterprise system. The seven-layer architecture addresses critical deployment gaps: MLOps governance, privacy engineering, scalability modeling, adversarial robustness, and human-cognitive optimization. With explicit economic modeling and comprehensive observability, OSKAR 2.0 is positioned for immediate platform integration while maintaining the transparency and ethical safeguards essential for democratic discourse.

OSKAR Online Safety & Knowledge Authenticity Resolver

MLOps Machine Learning Operations

RBAC Role-Based Access Control

PII Personally Identifiable Information

RPO Recovery Point Objective

RTO Recovery Time Objective

LoRA Low-Rank Adaptation

GNN Graph Neural Network

GAT Graph Attention Network

ECE Expected Calibration Error

KL Kullback-Leibler

PSI Population Stability Index

SLA Service Level Agreement

QPS Queries Per Second