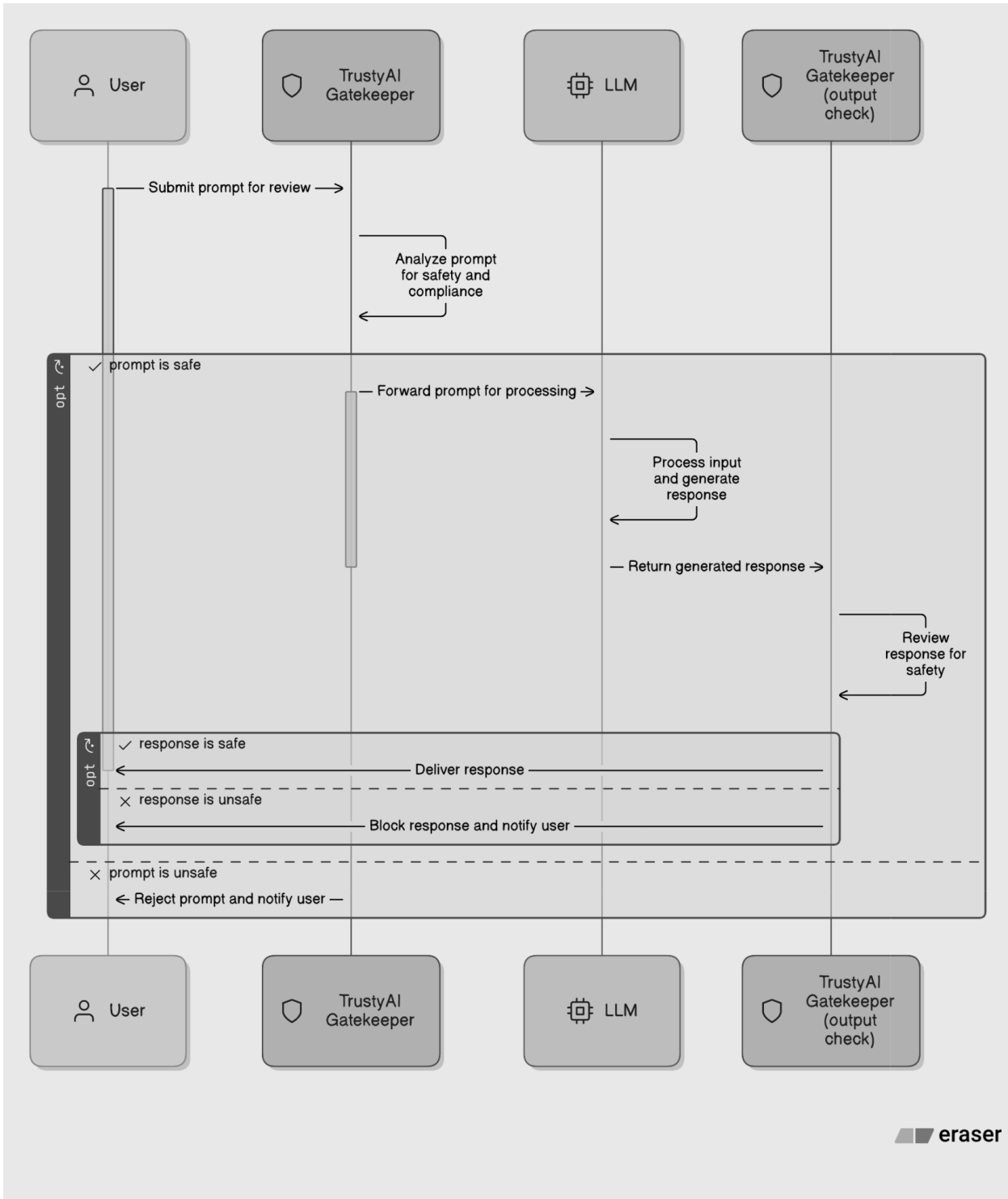
Using TrustyAI + RGAF for Safer GenAI and Faster Standards Compliance

TrustyAI is not another LLM—it is the open-source “safety layer” that sits between users and any generative model. The two key components of the AI Safety layer are focussing on guardrails of LLM) and Evaluation (focussing on quantifying LLM performance. I t includes: runtime guardrails, bias & drift metrics for predictive models that update while in production, regulatory explanations for every blocked or modified response, and evident logs that auditors can use. TrustyAI provides these four core features as general purpose-built tooling for the checklist items inside every Responsible Generative AI Framework (RGAF) dimension—turning high-level responsible-AI prose into measurable, enforceable, and loggable runtime behaviour. All exposed as Kubernetes-native micro-services so you can add them to an existing KServe, OpenShift AI or plain REST pipeline without touching the model weights.

The Linux Foundation AI & Data’s Responsible Generative AI Framework (RGAF) lists nine non-functional dimensions that every generative AI system must be: 1) human-centered & aligned, 2) accessible & inclusive, 3) robust, reliable & safe, 4) transparent & explainable, 5) accountable & rectifiable, 6) private & secure, 7) compliant & controllable, 8) ethical & fair, and 9) environmentally sustainable. TrustyAI’s guardrails, live-bias metrics, explainability logs and privacy filters directly instrument the first seven dimensions today, while its roadmap adds inclusive-evaluation datasets and carbon-per-inference tracking to close the remaining two which giving teams a single open-source toolkit that satisfies the full RGAF checklist out of the box.

TrustyAI Architecture


User
TrustyAI Gatekeeper
LLM
TrustyAI Gatekeeper (output check)
Submit prompt for review
Analyze prompt for safety and compliance
opt
prompt is safe
Forward prompt for processing
Process input and generate response
Return generated response
Review response for safety
opt
response is safe
Deliver response
response is unsafe
Block response and notify user
prompt is unsafe
Reject prompt and notify user
User
TrustyAI Gatekeeper
LLM
TrustyAI Gatekeeper (output check)
eraser

Guardrails

TrustyAI's guardrails functionality acts as a policy-enforcing gatekeeper that inspects every prompt and every model-generated response in real time. Implemented as lightweight, Kubernetes-native micro-services, the guardrails chain together pluggable detectors—hate speech, personally identifiable information (PII), prompt injection, hallucination, toxicity, copyright—each with user-tunable thresholds. If an incoming prompt or outgoing token stream violates the rule set, the gatekeeper either blocks the transaction outright or rewrites the offending snippet, then emits a JSON explanation (SHAP + counterfactual) and an immutable audit log entry. Because the framework loads new policies via ConfigMap hot-reload and exposes Prometheus metrics for every intervention, operators can tighten rules, measure false-positive rates, and demonstrate compliance without ever touching the underlying LLM weights or retraining data.

TrustyAI's guardrails translate the high-level obligations in the RGAF into enforceable runtime controls. By blocking, biased, off-topic or non-compliant content before it reaches the user, the guardrail layer directly satisfies RGAF's "Robust, Reliable & Safe" dimension and the "Ethical & Fair (unbiased)" requirement for active mitigation. SHAP explanations are not simultaneous and are not generated as part of the guardrails systems out-of-the-box. In short, each guardrail intervention produces a measurable, logged action that auditors can replay, turning RGAF principles into live, quantitative evidence.

Under ISO/IEC 42001:2023 the TrustyAI's guardrail bundle gives us evidence for multiple AIMS clauses in one shot:

- **6.1 (risk assessment)** – each detector is a pre-identified risk control; its config file is the documented risk-criteria.
- **8.2 (risk treatment)** – blocking / rewriting is the treatment action; Prometheus alerts prove the treatment triggers in real time.
- **9.1 (monitoring & measurement)** – every intervention increments a metric (guardrail_violations_total{rule="hate"}) that feeds KPI dashboards auditors can sample.
- **9.3 (management review)** – immutable JSON logs supply time-stamped records of non-conformities and corrective actions (raising threshold, adding new rule, etc.).
- **A.5.2 (AI-specific controls)** – the guardrail YAML itself is the "automated control for harmful outputs" the standard expects to see documented and tested.

Because rules load via ConfigMap and metrics export automatically, an organisation can demonstrate that its AI Management System continuously measures, controls and improves generative-AI risk which is exactly what an ISO 42001 external auditor wants to trace.

Bias & Drift

TrustyAI embeds fairness and drift monitoring capabilities appropriate to the model type. For predictive ML models, fairness calculators operate within the inference path, emitting real-time group-fairness metrics—demographic parity and equal-opportunity differences across protected

attributes—so alerts fire when decision rates for protected groups deviate beyond configured thresholds. Counter-factual fairness tests can temporarily perturb sensitive attributes in embedding space to measure outcome deltas. A parallel drift-detector tracks distribution shifts between live data and reference baselines using measures such as Kullback-Leibler divergence; when thresholds are breached, the service can trigger remediation workflows including canary rollbacks or traffic throttling.
For generative AI systems, TrustyAI applies adapted monitoring approaches suited to open-ended text generation, including toxicity detection, content policy enforcement, and output quality metrics, rather than the predictive fairness gauges described above.
These capabilities map to RGAF requirements across multiple dimensions. Under the Ethical & Fair dimension, RGAF demands "continuous or adaptive bias mitigation" and "quantitative evidence that protected groups are not systematically disadvantaged." For predictive use cases, TrustyAI's fairness gauges and counter-factual tests provide observable, metric-based controls per inference. Under Robust, Reliable & Safe, the drift monitor addresses requirements to "detect performance degradation promptly and trigger remediation." By embedding these calculators in the inference path where architecturally appropriate, TrustyAI translates high-level RGAF obligations into concrete, enforceable controls with audit trails regulators can review.
Similarly, these functions align with ISO/IEC 42001:2023 clauses 6.1 (risk assessment), 8.2 (risk treatment), 9.1 (monitoring & measurement), and 9.3 (management-review evidence). Streaming fairness metrics and drift indicators give the AI Management System objective KPIs for bias and performance risks. Automated alerts and rollback capabilities satisfy ISO 42001's requirement to "take action when risk criteria are exceeded" (8.2), while immutable logs provide auditors with documented evidence of continuous risk control (Annex A.5.2).

Regulatory Explanations

When a guardrail blocks or rewrites a generative-AI reply, TrustyAI captures structured audit data that supports regulatory transparency requirements including GDPR Article 22. The system records which policy triggered the intervention, the specific content characteristics that caused the block or rewrite, and the resulting action. These records are time-stamped, hashed, and stored in a tamper-evident audit log, giving legal teams a ready-made explanation of automated decisions without exposing full model weights or proprietary training data.

TrustyAI tools mapping to RGAF dimensions

RGAF Dimension	TrustyAI Tooling That Delivers It
Robust, Reliable & Safe	• Guardrails Framework (input/output filters, jail-break detection) • Bias & Drift REST service (real-time safety metrics)

Ethical & Fair (unbiased)	• Bias metrics API • LM-Eval harness (Winogender, StereoSet, custom fairness tasks)
Transparent & Explainable	• SHAP/LIME explainer service • Explainable Decision Logging (why was a prompt blocked?)
Accountable & Rectifiable	• Immutable log of every intervention (who, what, why) • KServe integration stores raw + modified prompt pairs for replay
Compliant & Controllable	• Dynamic policy engine (GDPR, HIPAA, NIST rules as code) • RBAC + Keycloak plug-in
Private & Secure	• PII scrubber / data-sanitiser guardrail • TLS-by-default, token-redaction in logs
Accessible & Inclusive	• LM-Eval diverse-language tasks • Performance dashboards split by demographic attributes
Environmental Sustainability	*Gap today* – CO_2-per-inference metrics on the roadmap for Q3-26

Using TrustyAI and RGAF for ISO/IEC 42001 conformity

ISO/IEC 42001:2023 i is AI Management System (AIMS) standard. The following chart shows clause-by-clause, here how TrustyAI + RGAF provide evidence of artefacts that auditors expect.

ISO 42001 Clause	Auditor Ask	TrustyAI Single Command Answer
4.1 – Context	“Show me your AI risk criteria.”	Export the YAML guardrail policy file → instant documented risk appetite.
6.1 – Actions to address R&O	“Prove you detect bias.”	Point to live Prometheus metric `trustyai_bias_dp{model="foo"}`.
7.2 – Competence	“How do operators know	Screenshot the SHAP explanation returned to the user.

	why a model was blocked?”	
8.2 – Risk assessment	“Demonstrate continuous monitoring.”	Grafana dashboard auto-refreshes drift & fairness scores every 30 s.
8.3 – Risk treatment	“Show treatment when drift > threshold.”	Alertmanager webhook triggers canary rollback; log entry is immutable.
9.1 – Monitoring & measurement	“Provide 12-month KPI trend.”	CSV export from Prometheus or the embedded TrustyAI **report generator**: `kubectl exec trustyai-reporter -- java -jar reporter.jar --period 12m --format iso42001`.
9.3 – Management review	“Minutes where guardrail false-positive rate was reviewed.”	Use the logged `intervention_type="fp"` filter to produce a review-ready table.
10.2 – Non-conformity & corrective action	“Show CAPA for a jail-break that succeeded.”	Search audit log for `jailbreak_score > 0.8 AND action=allowed`; link to Git commit that raised threshold.

Conclusion

TrustyAI provides an easy way to turn Responsible Generative AI Framework dimensions from a policy document into production code. Its guardrails, live fairness metrics, SHAP explanations and tamper-evident logs map one-to-one to every dimension of the RGAF principles while simultaneously generating the real-time evidence for ISO/IEC 42001:2023 auditors who demand risk assessment, treatment, monitoring and management review. In short, by using TrustyAI and RGAF principles you get a safer GenAI and a faster path to regulatory compliance to standards like ISO 42001 AIMS certificate.