PROTOPIA + aws

www.protopia.ai

Powering Private Inference in AWS Environments

Unlock Your Sensitive Enterprise Data for AI

Available through Amazon SageMaker and AWS Marketplace, Protopia Stained Glass Transform (SGT) adds a drop-in privacy layer to protect critical data in enterprise AI applications. SGT enables secure inference on open-weight models, starting with Llama 3.1 8B, and is designed to support high-impact use cases like RAG, document processing, and customer interaction, especially in industries where sensitive or proprietary data would otherwise block adoption.

Unlock New Use Cases

Enable private inference on data that was previously too sensitive to use due to regulatory considerations, customer privacy, and data leakage concerns.

Lower total cost of ownership

Run sensitive AI workloads on shared infrastructure. No need for isolated GPUs, enclaves, compute-intensive obfuscation techniques.

Accelerate time-to-impact

Use sensitive data safely in RAG and document workflows, without brittle token filtering or format-specific preprocessing.

How Stained Glass Transforms Work

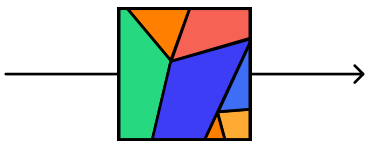
Stained Glass Transform (SGT) converts plaintext inputs into target-model-compatible stochastic embeddings through an OpenAI-compatible API, eliminating raw data exposure risks while maintaining full model accuracy and near-zero performance impact.

Unprotected Prompt (Without Protopia)

Context & Instruction

Create a SOAP note. Patient: John Doe | DOB: 05/12/1973 | MRN: 00432879 | Insurance: FutureCare PPO #FC-832-99213 Clinic: Acme Family Medicine | Chief Complaint: “Fatigue and shortness of breath on exertion.” | Past Medical History: T2DM (...

Stained Glass Transform™



Protected Prompt (With Protopia)

Context & Instruction

}, // ', ');){ '); }); a>?[<}); "> });];]; >?[<});)); >?[< } s}; '); >?[< */ },]); }; >?[< '); KS}); }); '); ', ', }); */ // ', for'); our', "> s'); ', */ */ >?[< */ G }; >?[< '); };]); }; */ ', ',){ '; }); //){ ', */ >?[<}; ', });){ ', /** //){ >?[< */ ', ";]); */ ()); }, } '; }; ',]); }); ()); // }; }, ', (); } ', (); };){ };...

Maintain High Performance While Eliminating Plain-Text Data Exposure

Model	Sentence Completion (HellaSwag)	Language Understanding (MMLU)	Model Truthfulness	Abstraction Reasoning (ARC)
Llama 3.1 8B W/ SGT*	64.38%	50.13%	49.02%	67.63%
Llama 3.1 8B without SGT	67.2%	56.06%	52.99%	67.72%

Model weights, accuracy, performance **remains intact**

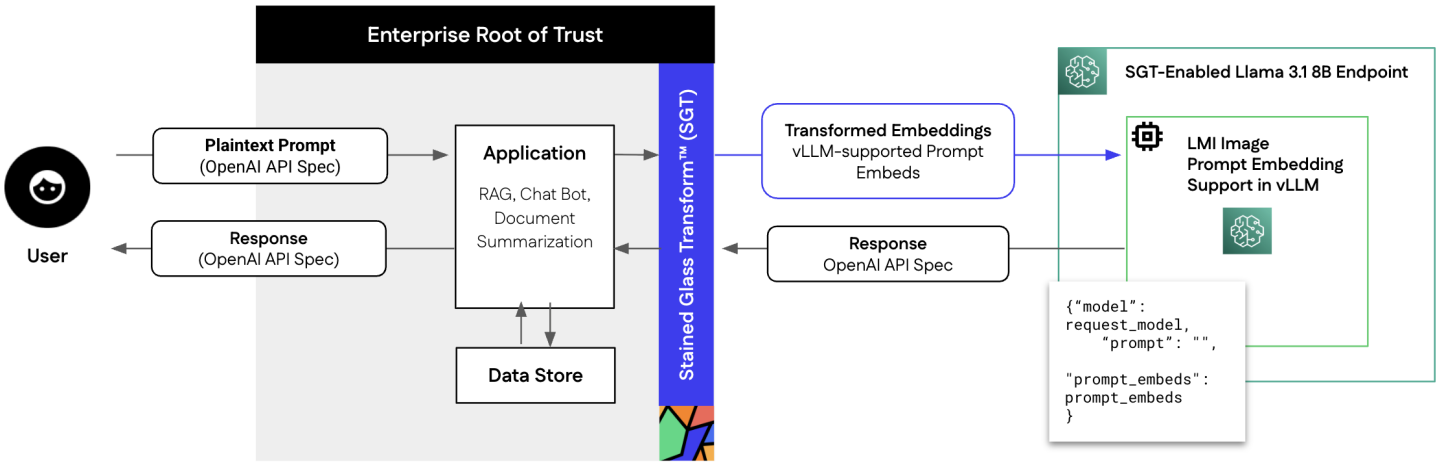
<1% Added to inference time

~25 milliseconds latency for Llama 70B SGT for ~200 tokens

**98.44% average tokens transformed to stochastic representations*

Streamlined & Secure AI Deployment on AWS

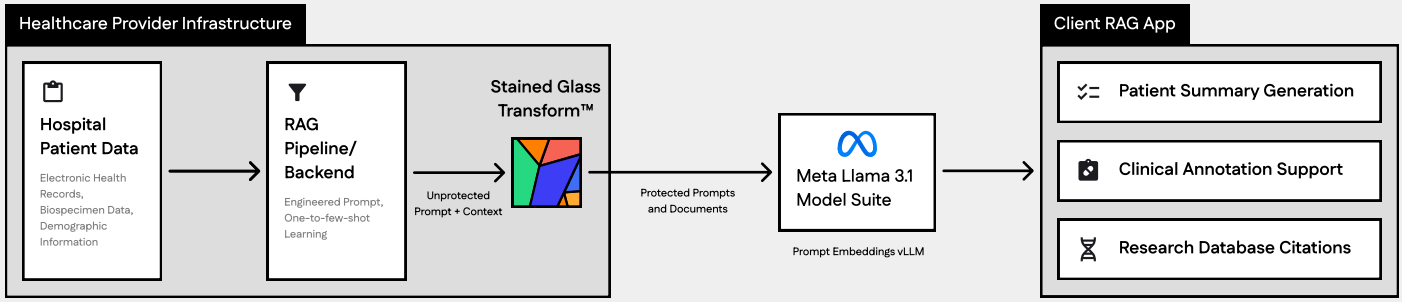
Stained Glass Transform ensures that sensitive information never exists in plaintext outside of the enterprise zone of trust. Customers maintain full data ownership without sacrificing accuracy, performance, or cost. SGT works seamlessly with AWS-native services like EKS, S3, and IAM as well as AWS's built-in procurement, billing, and governance tools.



Stained Glass Proxy sits within your enterprise zone of trust, transforming model inputs into unreadable, irreversible stochastic representations *before* reaching the SageMaker endpoint. This approach preserves security by keeping raw data local while eliminating model management overhead and accelerating time-to-value. SGT-enabled Llama 3.1 8B is available now in SageMaker with support for larger models, including Llama 3.3 70B, and Amazon Bedrock coming soon.

RAG Case Study:
Advancing Medical Research While Protecting Patient Data

Protopia partnered with Meta and a leading Major Health System to securely implement a RAG application for oncology research. By using Protopia SGT, the institution overcame strict privacy constraints, enabling secure AI-powered collaboration with doctors and researchers.



Technical Resources

[Read the Docs ↗]

Follow a step-by-step walkthrough to deploy SGT Proxy in AWS.

[Sample Notebook ↗]

Configure your AWS endpoint using Stained Glass Transform.

[vLLM Deep-Dive ↗]

Technical overview of how SGT leverages vLLM's prompt_embed parameter.