Guard

Producer:Manav Kapur

Why Agentic AI Needs a Guardrail Layer

*

Safety logic issues

Scattered logic
Inconsistent implementation
Duplicated across projects

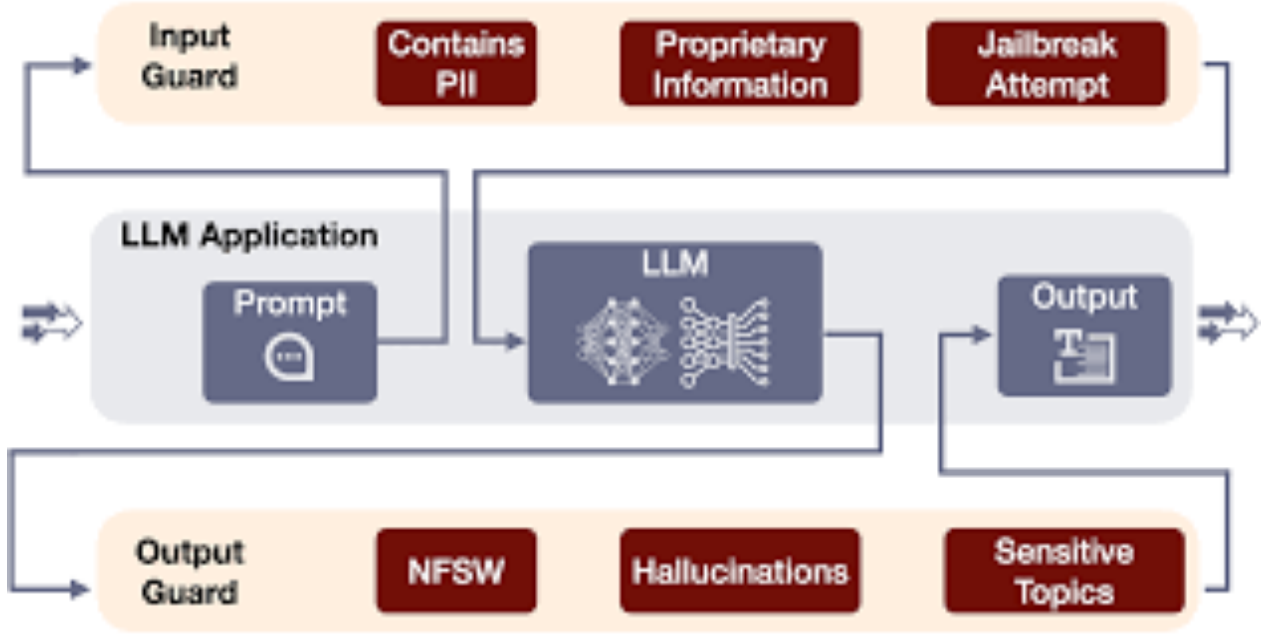
Agentic AI systems risks

Unsafe or malicious prompts
Hallucination or wrong actions
Leak of sensitive information
Jailbreaking

Increased risk and costs

Increased risk
Higher maintenance costs
Compliance issues

Noc agentic AI with Guardrail Layers



Guardrails interception points

Input prompts (before LLM)
Output responses (before delivery)

Enforced aspects

Safety
Compliance
Reliability

Comparison of Guardrail Technologies

Guardrail Technology Options

01

Guardrails AI

Pro : - Easy to embed, strong validator

Cons :- Limited jailbreak & scope control, not for agent behavior

02

NeMo Guardrails

Pro : - jailbreak prevention, agent behavior governance

Cons :- Requires project-level configuration, Less suitable as a lightweight

03

OpenAI Guardrails / Moderation

Pro : - No ML maintenance

Cons :- Limited customization

Our Proposal

Guardrail Framework SDK (Unified Interface)

Build a company Guardrail SDK
Acts as a single safety entry point
Any agent imports and uses it
Internally plugs different guardrail engines

How SDK solves problem

Instead of:

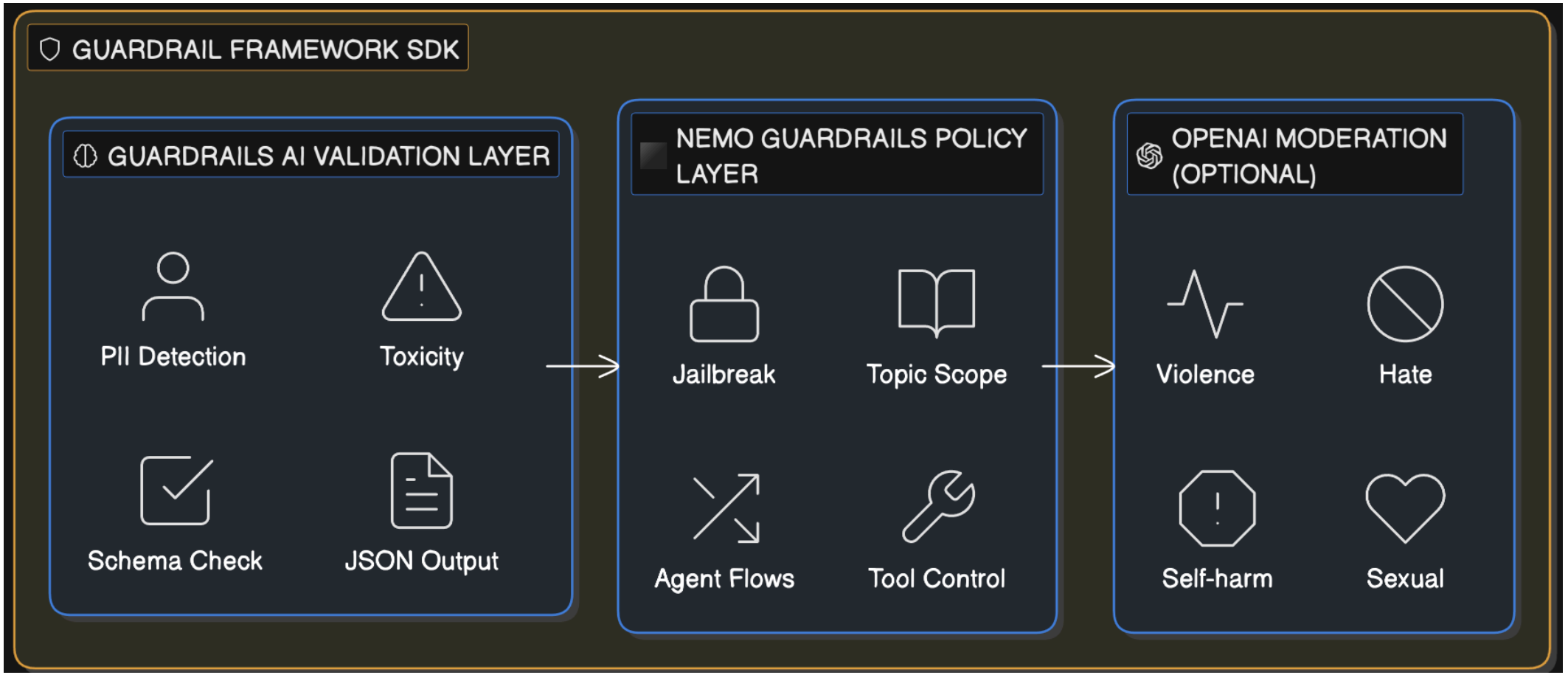
```
Client project → NeMo configs → NeMo runtime
```

You do:

```
Client project → Guardrail SDK → (internal)
NeMo runtime + configs
```

So NeMo becomes an **internal engine**, not a client dependency.

NeMo Guardrails requires configuration at the engine level, but in our architecture that configuration is owned by the Guardrail SDK, not by each project. Clients only select a profile, which makes the SDK plug-and-play


GUARDRAIL FRAMEWORK SDK
GUARDRAILS AI VALIDATION LAYER
PII Detection
Toxicity
Schema Check
JSON Output
NEMO GUARDRAILS POLICY LAYER
Jailbreak
Topic Scope
Agent Flows
Tool Control
OPENAI MODERATION (OPTIONAL)
Violence
Hate
Self-harm
Sexual

What Each layer Does

Layer	Purpose	Examples
Input Guard	Before LLM	Jailbreak, PII, topic scope
Output Guard	After LLM	Toxicity, hallucination, schema
Policy Guard	Agent behavior	Allowed tools, domains
Risk Guard	Compliance	Violence, hate, sexual, self-harm

How a Request Proceed

User Prompt
↓
Guardrail SDK
- NeMo: scope & jailbreak
- Guardrails AI: PII & validation
- OpenAI: risk classification
↓
LLM / Agent reasoning
↓
Guardrail SDK
- Guardrails AI: schema & toxicity
- NeMo: hallucination & policy
- OpenAI: compliance
↓
Safe Response

Why a Guardrail SDK (Not a Single Tool)

Centralized governance Plug-and-play for teams Avoids vendor lock-in
Modular and replaceable Consistent logging & audit Scales across all AI projects

Implementation Plan

Phase 1: Guardrail SDK interface

Phase 2: Guardrails AI integration

Phase 3: NeMo Guardrails integration

Phase 4: Optional OpenAI moderation

THANKS