AI Portal - Web Surfing Studios

Unified Visibility + Access Control for AI Usage Across WSS

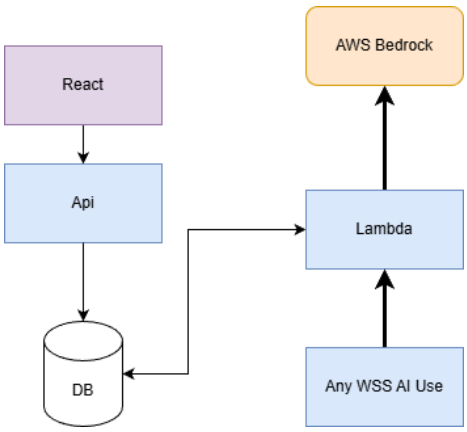
The Problem

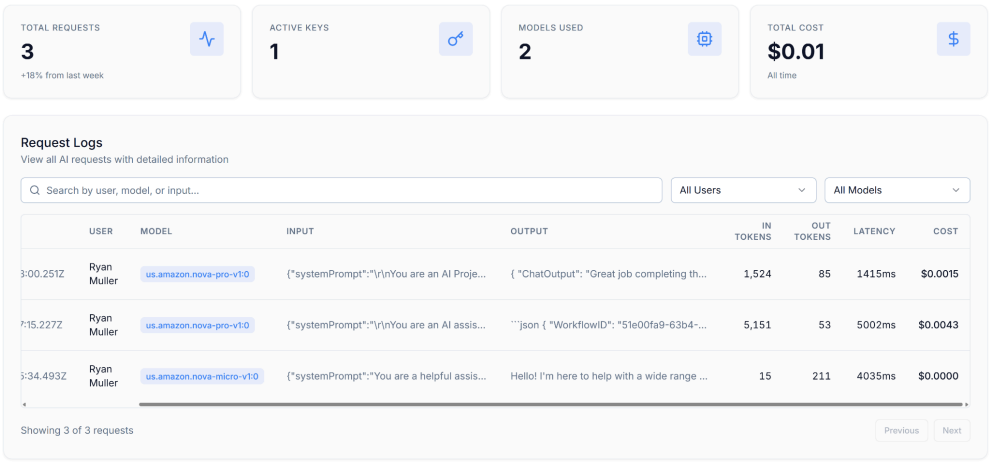
Multiple WSS apps call multiple AI models — but nothing shows who is using what, or how much spend each person drives.

Today we have:

- No unified location where access keys live
- No history of prompts or responses
- No way to link usage (and cost) back to a specific person

Because of that, we cannot measure usage, ROI, waste, or abuse.





The Idea

One Key Per User
Each developer receives a personal access key — simple to issue, simple to rotate, simple to revoke.

Automatic Expiration
Keys expire every 30 days by default — this enforces rotating discipline without manual policing.

Centralized Gateway
All AI traffic flows through the Portal. External apps call the Portal, and the Portal calls the AI model. This guarantees 100% visibility.

Unified Usage Logs
Every request is logged in one place — inputs, outputs, model ID, token counts, latency, and cost. Nothing is lost in random services.

Full Transparency
No private log modes. All members can view all requests. This teaches real-world cost management and audit discipline.

Instant Offboarding
Disabling one key disables a user everywhere — no hunting down hidden config files.

The Portal turns AI usage from a black box into a governable resource.

Architecture

React App + API
Front-end portal and REST API layer used for:

- Generating & managing access keys
- Viewing usage logs
- Managing users and revocation

AWS Lambda Gateway
Serverless execution layer that:

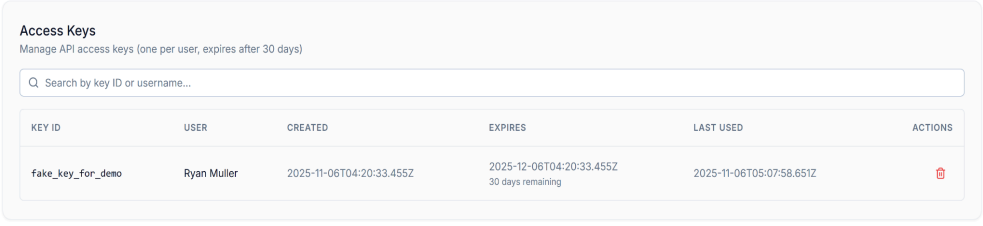
- Receives AI requests from WSS systems
- Validates keys + permissions
- Calls AWS Bedrock models
- Logs full request/response into the DB

Strategic Goal

The AI Portal gives WSS a scalable foundation for AI usage at any size. As more systems, bots, and developers start calling models, we don't have to worry about governance falling apart — because identity, access, and cost are all controlled and visible in one place. This lets WSS grow AI usage confidently, without losing governance or control..

Future Ideas

- Automated cost alerts and quotas per user or per team
- Anomaly detection to spot sudden spikes or abuse
- Model performance comparisons and ranking
- Usage-based billing or chargeback for project budgets
- Quality-of-output scoring to evaluate which prompts or models perform best
- Automatic detection of repeated prompt patterns that can be optimized into reusable templates
- Per-model or per-domain access control (e.g. only PMs can call certain models)
- Key rotation automation every 30 days without manual admin
- Latency / reliability benchmarking across providers to identify best-fit models
- Centralized model version tracking for historic reproducibility and audit



Created by Web Surfing Studios — a professional volunteer-driven software organization building real production systems. https://websurfingstudios.com