SOLUTION OVERVIEW

Semantic API Caching

Intelligent Caching for AI Applications

SOLUTION OVERVIEW FOR AKAMAI CONNECTED CLOUD

The Problem With Traditional Caching

Traditional API caching only hits on exact key matches, missing opportunities to reuse responses for semantically similar requests. Two differently-phrased questions seeking the same information result in redundant API calls, higher latency, and unnecessary backend load. This inefficiency is particularly costly for AI/LLM endpoints, where users frequently ask similar questions using different words.

Introducing AAM Semantic Caching

Akamai AI and API Manager (AAM) Semantic Caching uses advanced AI to cache by meaning, not just matching characters. When a new request arrives, the policy identifies semantically similar cache keys and returns cached responses when intent matches—even without exact duplication. This intelligent caching layer dramatically reduces repeated backend calls while maintaining response accuracy.

AI-Powered Similarity Matching

The policy uses LLM-generated embeddings to understand meaning behind each cache key, not just text. You define similarity tolerance to control how closely new requests must match cached keys.

Fully Customizable

AAM is fully programmable—create custom cache keys that fit your API, tune cache TTL, and customize response codes the policy applies to.

Proven In Production

Studies show Semantic Caching increases cache hit-rates by 50%+, significantly reducing latency and API traffic. Whether building search APIs or proxying LLMs, Semantic Caching delivers massive benefits.

Benefits of Semantic Cache Usage

Reduce API Costs

Avoid duplicate calls to backend services (especially expensive LLMs), dramatically lowering operational expenses.

Scale Your AI/LLM API

Handle traffic spikes and repeated queries without backend strain. High cache hit rates keep backend workload and DB queries low as usage grows.

Improved Response Times

Users get faster answers. Cache hits serve instantly from the edge, eliminating upstream call delays.

Consistent Experience

Deliver reliable responses for similar requests. Even when users rephrase questions, they get the same accurate answer quickly—providing a smooth, cohesive experience.