DOI 10.37882/2223-2966.2025.05-2.16

КОНЦЕПЦИЯ СИСТЕМЫ ИНТЕЛЛЕКТУАЛЬНОГО УПРАВЛЕНИЯ И БАЛАНСИРОВКИ НАГРУЗКИ ДЛЯ API-СЕРВИСОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

CONCEPT OF INTELLIGENT CONTROL AND LOAD BALANCING SYSTEM FOR ARTIFICIAL INTELLIGENCE API SERVICES

P. Matsipudra
V. Shmakov

Summary. This article presents an innovative middleware software concept designed for centralized management of access to artificial intelligence API services under modern geopolitical restrictions and challenges. The paper thoroughly examines complex issues, encompassing both the significant lack of effective tools for managing multiple API keys and the critical need to ensure reliable uninterrupted access to international AI services for organizations located in regions with limited access to global technological resources. As a solution, a multi-component system is proposed, organically combining advanced load balancing mechanisms with innovative methods of bypassing regional restrictions through specialized servers located in countries with open access. The developed solution includes comprehensive components for monitoring, control, and multi-level security assurance. The study conducts a comprehensive comparative analysis of existing market analogues, identifies their significant limitations in the context of solving the problem of geopolitical accessibility, and presents a convincing justification for the need to develop a qualitatively new solution. The technical implementation of key system components is described in detail, utilizing modern technologies and approaches to developing enterprise-grade fault-tolerant software. The obtained research results demonstrate the high effectiveness of the proposed solution in ensuring stable access to AI services and optimizing their use in organizations of various scales, regardless of their geographical location and current geopolitical situation.

Keywords: API management, middleware, artificial intelligence, enterprise security, scalability, geopolitical restrictions, centralized management, load balancing, cost optimization, fault tolerance, proxy server, API security, LLM services, API access management, OpenAI API, Claude API.

Маципудра Петр Евгеньевич
Санкт-Петербургский политехнический университет Петра Великого
matsipudra.pe@edu.spbstu.ru
Шмаков Владимир Эдуардович
к.т.н., доцент, Санкт-Петербургский политехнический университет Петра Великого
shmakov_ve@spbstu.ru

Аннотация. В данной статье представлена инновационная концепция программного обеспечения промежуточного уровня, предназначенного для централизованного управления доступом к API-сервисам искусственного интеллекта в условиях современных геополитических ограничений и вызовов. В работе детально рассматривается комплексная проблематика, охватывающая как существенный недостаток эффективных инструментов управления множественными API-ключами, так и критическую необходимость обеспечения надежного бесперебойного доступа к международным ИИ-сервисам для организаций, находящихся в регионах с ограниченным доступом к глобальным технологическим ресурсам. В качестве решения предложена многокомпонентная система, органично сочетающая передовые механизмы балансировки нагрузки с инновационными методами обхода региональных ограничений через специализированные серверы, расположенные в странах с открытым доступом. Разработанное решение включает в себя комплексные компоненты мониторинга, контроля и обеспечения многоуровневой безопасности. В рамках исследования проведен всесторонний сравнительный анализ существующих аналогов на рынке, выявлены их существенные ограничения в контексте решения проблемы геополитической доступности и представлено убедительное обоснование необходимости разработки качественно нового решения. Подробно описана техническая реализация ключевых компонентов системы с использованием современных технологий и подходов к разработке отказоустойчивого программного обеспечения корпоративного уровня. Полученные результаты исследования демонстрируют высокую эффективность предложенного решения для обеспечения стабильного доступа к ИИ-сервисам и оптимизации их использования в организациях различного масштаба, независимо от их географического положения и текущей геополитической обстановки.

Ключевые слова: API-менеджмент, middleware, искусственный интеллект, корпоративная безопасность, масштабируемость, геополитические ограничения, централизованное управление, балансировка нагрузки, оптимизация расходов, отказоустойчивость, прокси-сервер, API-безопасность, LLM-сервисы, управление доступом API, OpenAI API, Claude API.

Актуальность

В современных условиях организации, использующие API-сервисы искусственного интеллекта, сталкиваются с отсутствием эффективных инструментов для централизованного управления и контроля использования различных ИИ-сервисов. Существующие решения не предоставляют возможности гибкого распределения нагрузки между API-ключами, что приводит к неэффективному использованию ресурсов, превыше-

нию установленных лимитов запросов и, как следствие, к незапланированному увеличению операционных расходов. В условиях сложившихся геополитических ограничений доступа к международным ИИ-сервисам [1], организации нуждаются в надежном решении, способном обеспечить бесперебойный доступ к этим сервисам. Отсутствие комплексного решения для мониторинга, логирования и контроля расходов при использовании API различных ИИ-сервисов существенно затрудняет процесс принятия управленческих решений, снижает прозрачность использования ресурсов и создает дополнительные риски для информационной безопасности организации. Данная ситуация особенно критична для предприятий, где использование ИИ-сервисов интегрировано в ключевые бизнес-процессы и требует строгого контроля над расходами и соблюдением политик безопасности.

Формулировка задачи

В данной работе предлагается разработка программного обеспечения промежуточного уровня (middleware), выступающего в роли интеллектуального прокси-сервера для взаимодействия с различными провайдерами ИИ-сервисов. Данное ПО должно обеспечивать централизованное управление API-ключами, оптимизировать их использование посредством автоматической балансировки нагрузки и предоставлять администраторам гибкие инструменты контроля. В рамках предлагаемого решения необходимо реализовать систему мониторинга и журналирования всех операций, а также внедрить многоуровневую систему ограничений, позволяющую контролировать расходы, устанавливать лимиты по языковым единицам LLM моделей и управлять доступом с различных IP-адресов. Особое внимание уделяется безопасности и отказоустойчивости системы, обеспечивающим стабильную работу сервиса даже в условиях ограничения доступа к провайдерам ИИ-сервисов, вызванных геополитическими факторами, техническими сбоями в работе провайдеров, изменениями в региональном законодательстве или временными перебоями в работе телекоммуникационной инфраструктуры. Система должна обеспечивать непрерывность бизнес-процессов организации.

Анализ целевой аудитории

Разрабатываемое программное обеспечение ориентировано преимущественно на средние и крупные компании, активно внедряющие технологии искусственного интеллекта в свои бизнес-процессы, а также на разработчиков программного обеспечения, интегрирующих ИИ-сервисы в свои продукты. Данные организации характеризуются высоким уровнем использования API различных ИИ-сервисов и потребностью в оптимизации связанных с этим расходов. Особый интерес решение представляет для компаний, находящихся в регионах с потенциальными ограничениями доступа к международным ИИ-сервисам. Вторичный сегмент целевой аудитории включает образовательные учреждения, исследовательские центры и технологические стартапы, которым необходим контролируемый и безопасный доступ к ИИ-сервисам. Ключевыми потребностями целевой аудитории являются оптимизация расходов, повышение безопасности при работе с API, улучшение контроля над использованием ресурсов и получение детальной аналитики использования сервисов. Важным фактором является готовность целевой аудитории инвестировать в решения, обеспечивающие стабильность и безопасность их бизнес-процессов, связанных с использованием ИИ-технологий.

Сравнение с существующими аналогами

Анализ существующих решений для управления доступом к API искусственного интеллекта выявил существенные ограничения в текущих предложениях на рынке. Большинство доступных решений не обеспечивает комплексного подхода к управлению API-ключами, имеет ограниченные возможности масштабирования и не предоставляет достаточных инструментов для мониторинга и обеспечения безопасности. В ходе исследования были проанализированы следующие основные решения: OpenAI API (как эталонная реализация) [2], ProxyAPI [3] и LiteLLM [4]. Каждое из этих решений имеет свои особенности и ограничения, что создает потребность в более совершенном инструменте для управления доступом к API искусственного интеллекта.

OpenAI API, являясь стандартным решением, не предоставляет возможностей для управления множественными ключами и распределения нагрузки, что существенно ограничивает возможности масштабирования и оптимизации расходов.

ProxyAPI предлагает доступ к различным провайдерам ИИ через единый интерфейс, однако имеет существенные недостатки: отсутствие возможности самостоятельного хостинга, наличие дополнительной наценки за использование и необходимость подписки для некоторых функций. Кроме того, решение не предоставляет возможностей для тонкой настройки ограничений и управления доступом.

LiteLLM, будучи открытым решением, предлагает хороший функционал для работы с различными провайдерами ИИ и имеет систему команд (teams). Однако данное решение не обеспечивает распределение нагрузки между ключами одного провайдера и имеет ограниченные возможности по установке лимитов и мониторингу безопасности. Большое количество функций привязано к Enterprise версии данного ПО.

Таблица 1.

Сравнение характеристик аналогов реализации

Характеристика	Наша реализация	OpenAI API	ProxyAPI	LiteLLM
Самостоятельный хостинг	ДА	НЕТ	НЕТ	ДА
Наценка за использование	НЕТ	НЕТ	ДА	НЕТ
Необходимость подписки	НЕТ	НЕТ	Частично	НЕТ
Распределение нагрузки между ключами одного провайдера	ДА	НЕТ	НЕТ	НЕТ
Ограничение общего количества IP для ключа	ДА	НЕТ	НЕТ	НЕТ
Лимиты по токенам/денежным средствам	ДА	ДА	НЕТ	ДА
Периодическое восстановление лимитов для ключа	ДА	НЕТ	НЕТ	ДА
Собственные лимиты на частоту запросов	ДА	НЕТ	НЕТ	ДА
Открытый исходный код	ДА	НЕТ	НЕТ	ДА
Поддержка множества провайдеров	ДА	НЕТ	ДА	ДА
Система команд (teams)	НЕТ	НЕТ	НЕТ	ДА
Логирование запросов	ДА	НЕТ	НЕТ	ДА
Настраиваемая модерация логов для безопасности	ДА	НЕТ	НЕТ	НЕТ
Автоматическое ограничение ключей при мошенничестве	ДА	НЕТ	НЕТ	НЕТ
Особые предложения для предприятий	НЕТ	ДА	НЕТ	ДА

Реализация компонентов

Для предлагаемой системы интеллектуального управления ключевыми компонентами являются: система управления API-ключами, балансировщик нагрузки, система контроля и ограничений, система мониторинга и логирования, административный интерфейс, а также система автоматической обработки и модерации запросов.

Система управления API-ключами:

- Реляционная база данных

— Классическое решение с использованием PostgreSQL или MySQL для хранения API-ключей, их метаданных и настроек. Обеспечивает надежность и целостность данных, поддерживает сложные связи между сущностями.

Балансировщик нагрузки:

- Round-robin распределение

— Простой алгоритм циклического перебора доступных API-ключей. Подходит для базового распределения нагрузки, но не учитывает текущую загруженность и лимиты ключей.

- Weighted round-robin

— Усовершенствованный алгоритм с учетом весов ключей, основанных на их лимитах и текущей загрузке. Позволяет более эффективно распределять запросы.

Система контроля и ограничений:

- In-memory счетчики

— Хранение метрик использования в оперативной памяти с периодической синхронизацией в базу данных.

Система мониторинга и логирования:

- Локальное журналирование

— Запись логов в файловую систему с ротацией. Простое решение, подходящее для небольших систем.

Административный интерфейс:

- Web-интерфейс

— Классическое web-приложение с использованием современного фреймворка (React, Vue.js). Обеспечивает доступ через браузер с любого устройства.

- API-интерфейс

— REST API для программного управления системой. Позволяет интегрировать управление в существующие системы администрирования.

Система автоматической обработки и модерации запросов:

- Regex-фильтрация

— Использование регулярных выражений для базовой фильтрации контента. Подходит для простых проверок на наличие запрещенных слов, паттернов или форматов данных.

- Локальные LLM

— Использование легковесных LLM моделей (например, LLaMA, GPT-2) для локального анализа контента.

- API инструменты

— Использование специализированных API для модерации (например, OpenAI Moderation API, PerspectiveAPI).

Архитектура

Архитектура разрабатываемого программного обеспечения построена на принципах микросервисной архитектуры с разделением ответственности между

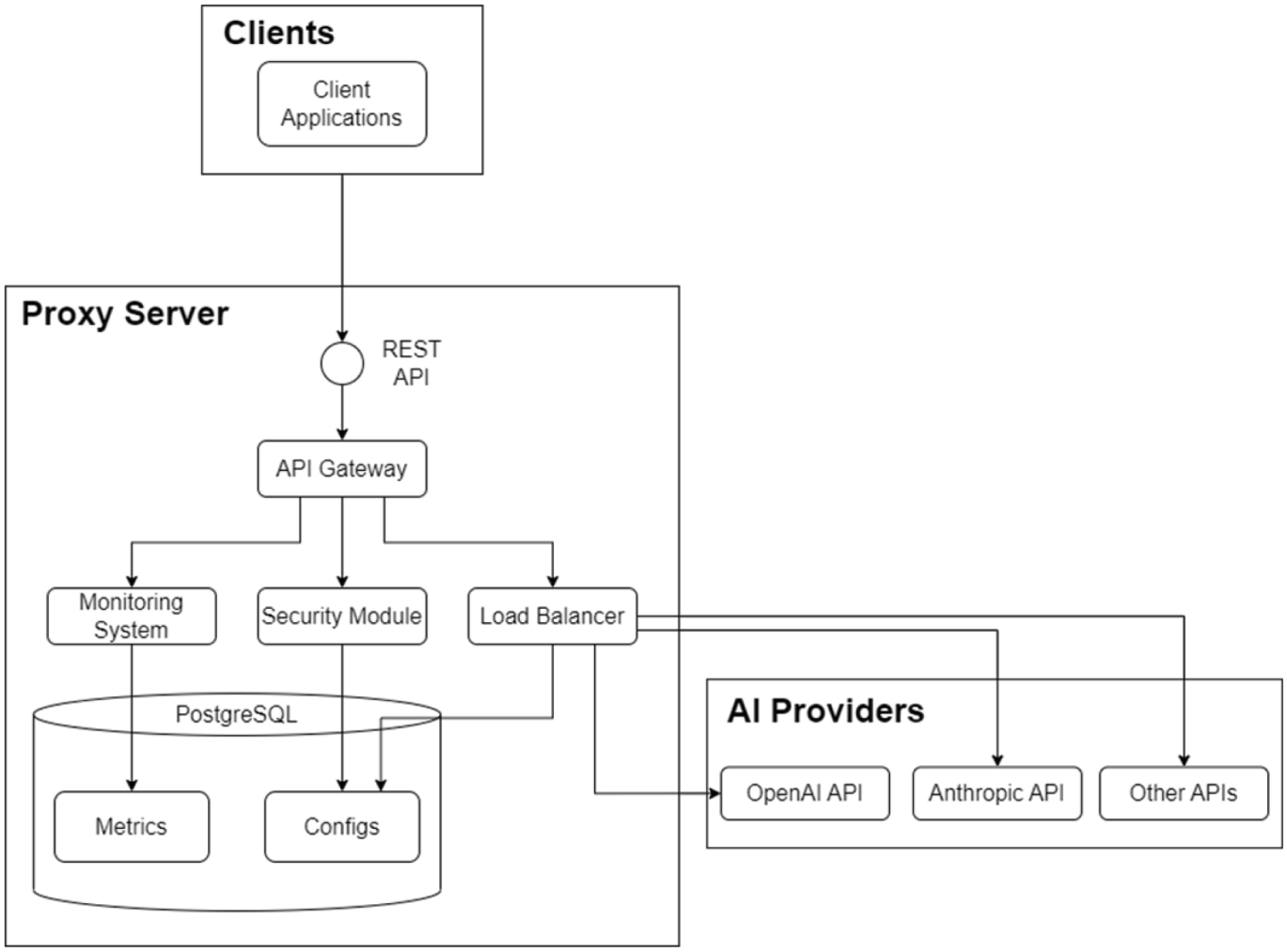

Рис. 1. Архитектура решения

компонентами. Это обеспечивает гибкость при модификации отдельных модулей и возможность горизонтального масштабирования. Система спроектирована с учетом требований масштабируемости, отказоустойчивости и безопасности.

- Архитектура включает следующих компонентов:
- Система управления API-ключами
- Балансировщик нагрузки
- Система контроля и ограничений
- Система мониторинга и логирования
- Административный интерфейс
- Система автоматической обработки и модерации запросов

Система управления API-ключами

Система управления API-ключами представляет собой критически важный компонент разрабатываемого программного обеспечения, реализованный на базе реляционной системы управления базами данных PostgreSQL[5]. Выбор PostgreSQL обусловлен его надежностью, производительностью и широкими возможностями для обеспечения целостности данных. В основе данной системы лежит структура таблиц, позволяющая эффективно хранить и управлять информацией о API-ключах различных провайдеров, их текущем состоянии и установленных ограничениях.

Особое внимание в системе уделено безопасности хранения и обработки API-ключей. Все чувствительные данные должны хранится в зашифрованном виде с использованием алгоритма AES-256, а доступ к операциям с ключами строго контролируется через систему ролей и разрешений.

Балансировщик нагрузки

Балансировщик нагрузки реализован как интеллектуальная система распределения запросов между доступными API-ключами, использующая комбинированный подход на основе алгоритмов Round-robin и Weighted round-robin. Базовый механизм Round-robin обеспечивает равномерное распределение запросов между всеми доступными ключами, в то время как weighted-компонент учитывает дополнительные факторы, такие как оставшиеся лимиты ключей, их приоритет и истори-

ческую производительность. Это позволяет динамически адаптировать распределение нагрузки в зависимости от текущей ситуации и установленных ограничений.

Система контроля и ограничений

Система контроля и ограничений реализована на основе высокопроизводительного механизма in-memory счетчиков. Данный подход позволяет эффективно отслеживать и контролировать множество параметров, включая денежные лимиты, количество использованных токенов, частоту запросов и активность с различных IP-адресов. Периодическая синхронизация данных с основной базой данных обеспечивает надежное сохранение истории использования и возможность последующего анализа.

Система мониторинга и логирования

Система мониторинга и логирования реализована на базе локального механизма записи логов с использованием библиотеки logging. Данное решение обеспечивает надежное сохранение информации о всех запросах, их параметрах и результатах выполнения, при этом автоматическая ротация логов предотвращает чрезмерное использование дискового пространства и облегчает последующий анализ данных. Несмотря на свою простоту, такой подход полностью удовлетворяет требованиям системы на начальном этапе разработки и может быть легко масштабирован при необходимости.

Административный интерфейс

Административный интерфейс представлен в двух взаимодополняющих форматах: веб-интерфейс на базе React и программный REST API [6] интерфейс. Веб-интерфейс предоставляет удобный визуальный доступ к функциям управления системой, включая мониторинг активности, управление ключами и настройку ограничений, в то время как REST API обеспечивает возможность программной интеграции с существующими системами администрирования и автоматизации. Оба интерфейса используют единую систему аутентификации на основе токенов и поддерживают детальное разграничение прав доступа для различных критериев [7].

Система автоматической обработки и модерации запросов

Система автоматической обработки и модерации запросов реализована как многоуровневый механизм фильтрации и анализа контента. На первом уровне производится базовая фильтрация с использованием регулярных выражений для выявления очевидных нарушений и запрещенного контента. Второй уровень включает локальную обработку с помощью легковесной LLM модели, которая анализирует семантику запросов на предмет потенциальных угроз безопасности [8]. Для особо важных случаев задействуется третий уровень — специализированные API модерации, такие как OpenAI Moderation API, обеспечивающие дополнительную проверку контента. Такой многоуровневый подход позволяет эффективно выявлять и блокировать потенциально опасные или неприемлемые запросы, сохраняя при этом высокую производительность системы.

Технологический стек

Backend:
- Python 3.11+ в качестве основного языка программирования
- FastAPI[9] как основной веб-фреймворк
- PostgreSQL для хранения данных
- Docker + Docker Compose[10] для контейнеризации

Ключевые библиотеки Python:
- httpx для асинхронных HTTP-запросов к API провайдеров
- pydantic для валидации данных и конфигурации
- python-jose для работы с JWT-токенами
- uvicorn: ASGI сервер

Frontend:
- React.js с TypeScript для разработки пользовательского интерфейса
- Material-UI или Tailwind CSS для стилизации компонентов
- React Query для эффективной работы с API

Заключение

Предлагаемая концепция программного обеспечения предоставляет уникальное решение для централизованного управления доступом к различным провайдерам ИИ-сервисов. Анализ существующих аналогов показал, что предлагаемое решение обладает рядом преимуществ, включая возможность самостоятельного хостинга, отсутствие дополнительных наценок и наличие расширенных функций безопасности и мониторинга. Разработанная концепция демонстрирует возможность создания надежной системы управления API-ключами с интеллектуальной балансировкой нагрузки, многоуровневой системой безопасности и детальным мониторингом использования ресурсов. Особую ценность представляет реализованный механизм автоматической модерации контента, сочетающий локальные и облачные решения для обеспечения максимальной защиты от потенциальных угроз.

Реализация данного проекта позволит организациям оптимизировать использование ИИ-сервисов, повысить

безопасность и эффективность работы с API, а также обеспечить стабильный доступ к сервисам в условиях возможных региональных ограничений. Дальнейшее развитие системы предполагает интеграцию с новыми провайдерами ИИ-сервисов и расширение функциональности для поддержки других типов API-сервисов, что значительно расширит возможности применения решения в различных бизнес-контекстах. Планируемое внедрение корпоративного уровня поддержки с расширенными инструментами администрирования и интеграции сделает систему еще более привлекательной для крупных предприятий, нуждающихся в надежном и масштабируемом решении для управления API-сервисами.

ЛИТЕРАТУРА

1. Wang, S.H. OpenAI — Explain Why Some Countries Are Excluded From ChatGPT / S.H. Wang // Nature. — 2023. — ISSN: 0028-0836. — eISSN: 1476–4687.
2. OpenAI API Documentation [Электронный ресурс]. — Режим доступа: https://platform.openai.com/docs/ (дата обращения: 24.01.2025).
3. ProxyAPI Документация [Электронный ресурс]. — Режим доступа: https://proxyapi.ru/docs (дата обращения: 24.01.2025).
4. LiteLLM Documentation [Электронный ресурс]. — Режим доступа: https://docs.litellm.ai/docs/ (дата обращения: 24.01.2025).
5. PostgreSQL Documentation [Электронный ресурс]. — Режим доступа: https://www.postgresql.org/docs/ (дата обращения: 24.01.2025).
6. REST API Документация [Электронный ресурс]. — Режим доступа: https://docs.github.com/ru/rest?apiVersion=2022-11-28 (дата обращения: 22.01.2025).
7. Аникин, Д.А. Анализ методов авторизации и аутентификации REST API / Д.А. Аникин // Информационные технологии и системы. — 2023. — С. 122. — eISSN: 2500-1752.
8. Денисов, Д.А. Искусственный интеллект как инструмент модерации контента / Д.А. Денисов // Современные информационные технологии. — 2024. — С. 19–22.
9. Docker Documentation [Электронный ресурс]. — Режим доступа: https://docs.docker.com/ (дата обращения: 24.01.2025).
10. FastAPI Documentation [Электронный ресурс]. — Режим доступа: https://fastapi.tiangolo.com/ (дата обращения: 22.01.2025).

Журнал «Современная наука: актуальные проблемы теории и практики»