

# ФРОЛОВ АРТЁМ АЛЕКСЕЕВИЧ

Backend Developer (Go / Python / C++)

Белгород (готов к релокации в Москву / СПб или удаленной работе) • Сайт: [kazinagg.site](https://kazinagg.site) • Telegram: [@kazinagg\\_it](https://t.me/kazinagg_it) •

GitHub: [github.com/Kazinagg](https://github.com/Kazinagg) • Email: [artemfrolov04@gmail.com](mailto:artemfrolov04@gmail.com)

Статус: Магистратура в академическом отпуске — свободен для полной занятости (Full-time, 40 ч/нед), учебная нагрузка отсутствует.

## ПРОФИЛЬ

Бэкэнд-разработчик с академической базой в области фундаментальной математики и компьютерных наук (бакалавриат МОиАИС, магистратура 02.04.01). Проектирую высокопроизводительные сервисы и распределенные системы на **Go, Python и C++**. Опыт построения production-ready RAG-конвейеров на GPU с реляционно-векторными БД (**PostgreSQL + pgvector**), разработки системных сетевых сервисов на Go (**Sing-box, WireGuard, NET\_ADMIN**), а также низкоуровневой оптимизации параллельных алгоритмов (**CUDA shared memory, OpenMP, MPI**). Понимаю внутреннее устройство Linux, асинхронный I/O и сетевой стек (L4/L7).

## ТЕХНИЧЕСКИЕ НАВЫКИ

**Языки программирования:** **Go, Python, C/C++** (C++17/20, CUDA C), **SQL** (PostgreSQL, SQLite), Java, TypeScript.

**CS & Параллелизм:** Алгоритмы и структуры данных, оценка сложности  $O(N)$ , вычисления на GPU (**CUDA shared memory**, редукция), **OpenMP, MPI**, многопоточность и асинхронность (**Asyncio, Goroutines**), модель памяти.

**Базы данных и брокеры:** **PostgreSQL** (ACID-транзакции, индексы, EXPLAIN), **pgvector** (косинусное расстояние), **Redis** (сессии, кэширование, pub/sub), **Celery**, SQLite, GORM, SQLAlchemy, Django ORM, Flyway, Liquibase.

**Архитектура и фреймворки:** **Gin (Go), FastAPI**, Django REST Framework, Spring Boot, RESTful API, **RAG** (Two-Stage Retrieval, Cross-Encoder), **Model Context Protocol (MCP)**, Server-Driven UI (SDUI), SSE.

**Инфраструктура и DevOps:** **Docker** (Multi-stage сборка), **Docker Compose** (сетевые режимы **host, NET\_ADMIN**), **Linux, Nginx** (Stream SSL/TLS proxy), systemd демоны, Git, GitHub Actions, RouterOS API (MikroTik), Ansible.

## ОПЫТ И КЛЮЧЕВЫЕ ПРОЕКТЫ

**Pegas AI — Корпоративная RAG-платформа и портал института** [github.com/Kazinagg/pegas-ai](https://github.com/Kazinagg/pegas-ai) **Python 3.12, FastAPI, PostgreSQL + pgvector, Redis, Celery, PyTorch / CUDA, Docker Compose**

- Спроектировал реляционно-векторное хранилище на **PostgreSQL + pgvector** с RBAC-изоляция курсов Moodle: связал документы и права студентов через таблицу `user_access`, гарантировав строгую изоляцию контекста в единых ACID-транзакциях.
- Построил двухэтапный конвейер поиска (**Two-Stage Retrieval**) на GPU: первичный отбор векторов моделью **BGE-M3 (FP16, 1024-d, top\_k=40)**, точное реранжирование через Cross-Encoder **bge-reranker-base (top\_n=7)** и контекстное склеивание (**Sliding Window Expansion  $\pm 1$** ).
- Реализовал защиту GPU от OOM: контроль конкурентного доступа (**GPU\_LOCK** на `asyncio.Lock()`) и вынос ресурсоемкой индексации в очередь **Celery + Redis** с единым персистентным циклом событий воркера.
- Разработал сессионное хранилище с **Fernet**-шифрованием паролей в Redis (TTL 7 дней), мультиагентный роутер **AgentSupervisor** (DeepSeek) и **Server-Driven UI** (10 типов интерактивных блоков со стримингом этапов рассуждения по SSE).

**All-in-One VPN Network Backend — Сетевой шлюз и панель управления** [github.com/Kazinagg/all-in-one-vpn](https://github.com/Kazinagg/all-in-one-vpn)

**Go 1.26, Gin, GORM, Telebot v3, Sing-box, AmneziaWG, Nginx Stream, Supervisor, Docker**

- Разработал высокопроизводительный бэкэнд и встроенного Telegram-бота на **чистом Go** (Gin, GORM, telebot.v3, CGO-free SQLite) для управления конфигурациями сетевых туннелей **Sing-box** (VLESS Reality, Shadowsocks, Hysteria2) и **AmneziaWG**.
- Спроектировал 5-этапный **Multi-stage Dockerfile**, объединивший панель, Sing-box, AmneziaWG, скомпилированный из исходников Nginx (со `--with-stream` модулем) и Supervisor под единым управлением в режиме `network_mode: host` с правами **NET\_ADMIN**.
- Реализовал генерацию обфусцированных конфигураций AmneziaWG (Jc, Jmin, Jmax, S1, S2, H1-H4) и маршрутизацию Sing-box с поддержкой GeoIP-фильтрации и бесшовным перезапуском демонов через `supervisorctl`.

**IoT-комплекс сетевой автоматизации арготехники (Wiren Board + MikroTik)** [github.com/Kazinagg/mactoir](https://github.com/Kazinagg/mactoir) **Go 1.26, Python, FastAPI, RouterOS API, Ansible, SQLite, Linux, systemd**

- Разработал сервис `wiren-mikrotik-manager` на **Go** с паттерном Registry для взаимодействия с **RouterOS API** контроллеров комбайнов, упаковав его в отказоустойчивый системный демон **systemd** с автоперезапуском при сетевых сбоях.
- Создал REST-микросервис `mactoir` на **FastAPI** и **SQLAlchemy** для динамического сопоставления MAC-адресов и IP оборудования; покрыл контрактные точки **pytest-тестами** с использованием `in-memory SQLite (StaticPool)`.
- Автоматизировал конфигурацию беспроводных бортовых сетей комбайнов через **Ansible** (модуль `community.routeros.api_modify`).

**Высокопроизводительные и параллельные вычисления (CUDA, OpenMP, MPI)** **C++, CUDA C, OpenMP, MS-MPI / OpenMPI, Алгоритмы и структуры данных**

- Написал вычислительные ядра на **CUDA C** для матрично-векторного умножения  $s = U^T \cdot x \cdot A$  с ручным размещением тайлов в разделяемой памяти (`__shared__ float s_data[]`) и параллельной редукцией, снизив задержки доступа к глобальной VRAM.
- Реализовал параллельный алгоритм Флойда-Уоршелла на **C++ (OpenMP)**, достигнув задокументированного ускорения **7.99x  $\approx$  8x** на 16 потоках для матрицы 1024x1024 (с 39.5 с до 4.9 с) и **8.38x** для матрицы 256x256.
- Реализовал распределенный обмен сообщениями на **MPI (MPI\_Send/MPI\_Recv)**, зафиксировав масштабируемое ускорение **до 4.08x** на 4 процессах.

**Open-Source ReadmeKit & Native Model Context Protocol (MCP) Server** [github.com/Kazinagg/readme-kit](https://github.com/Kazinagg/readme-kit) **Python 3.10+, Zero-Dependency, AST Markdown Compiler, Stdio JSON-RPC 2.0, SVG Engine**

- Спроектировал **Zero-Dependency** движок векторной графики и AST-компилятор Markdown-директив на чистом Python (13 700+ LOC, 20 тестовых модулей на `pytest`) без внешних графических зависимостей с мгновенным ин-мемори рендерингом.
- Реализовал собственный **Native Stdio JSON-RPC 2.0 MCP-сервер** (Model Context Protocol) и FastMCP-адаптер для вызова инструментов AI-агентами.

## ОБРАЗОВАНИЕ

**НИУ «БелГУ», Институт инженерных и цифровых технологий (ИИиЦТ) — Магистратура, 02.04.01**

2025 — 2028 (академ. отпуск)

«Математика и компьютерные науки»

**НИУ «БелГУ», ИИиЦТ, Кафедра МПОИС — Бакалавриат, 02.03.03 «Математическое обеспечение и администрирование ИС»**

2021 — 2025

**Тема ВКР:** «Разработка веб-портала и информационной подсистемы Института инженерных и цифровых технологий» (Django, React, TypeScript, Docker).