

Максим Елисеев

Senior Backend Engineer

Python · Go · Rust

7 лет опыта · Открыт к предложениям

Senior Backend Engineer с 7 годами коммерческого опыта разработки backend-систем на Python и Go. Специализируюсь на архитектуре распределённых сервисов, обработке данных в реальном времени и геопространственных вычислениях. Работал над системой мониторинга 90 000+ транспортных средств и агротехнологической платформой; отвечал за архитектурные решения, оптимизацию производительности и надёжность сервисов. Дополнительно развиваю независимый open-source проект на Rust.

Опыт работы

Senior Backend Engineer Python / Go — Magnifio

2023 — 2026

- Устранил критическую проблему производительности: расчёты обработки полей превышали 40-минутный лимит задачи Celery; после оптимизации время выполнения сократилось примерно до 5 минут
- Выявил квадратичную сложность O(n²) при попарной проверке пересечений полигонов; использовал пространственный индекс STRtree и предварительную фильтрацию геометрий
- Спроектировал и внедрил в production Go-микросервис для ресурсоёмких геопространственных вычислений, выделив их из Django/Celery в самостоятельный компонент backend-архитектуры
- Разрабатывал новые модули Django-backend, оптимизировал работу PostgreSQL/PostGIS, настроил CI/CD и Docker; реализовал аналитику по полям и севообороту

Backend-разработчик Python — GISDEV

2019 — 2023

- Разрабатывал backend системы мониторинга 90 000+ транспортных средств и сервисы обработки потоковой GPS-телеметрии на Python, aiohttp и asyncio (клиенты — «Газпромнефть» и др.)
- Реализовал обновление координат и статусов автомобилей в браузере через WebSockets для отображения движения на карте в реальном времени
- Оптимизировал пространственные запросы в PostgreSQL/PostGIS, разрабатывал фоновые задачи на Celery/Redis, поддерживал и рефакторил Django-монолит

Независимый проект

Backend-разработчик Rust — Konstruct

Ноябрь 2025 — н.в.

Независимый проект в свободное время. Открытый мессенджер с постквантовым шифрованием, который знает о пользователе ровно столько, сколько нужно для доставки сообщений: аккаунт — это пара ключей, без номера телефона и email.

- Реализовал протокольную логику защищённого обмена сообщениями на Rust: X3DH, Double Ratchet и постквантовое согласование ключей (PQXDH, ML-KEM); использовал ChaCha20-Poly1305 и гибридные подписи Ed25519/ML-DSA-65
- Разделил backend на сервисы identity, messaging, signaling и media; доставка сообщений через двунаправленный gRPC-поток по QUIC с автоматическим переходом на HTTP/2
- Опубликовал исходный код и выпустил клиенты для iOS (TestFlight) и Android (alpha)

github.com/konstruct-msg · спецификация протокола

Ключевые навыки

PYTHON
Django · FastAPI · asyncio · aiohttp · Celery

GO
микросервисы, георасчёты

RUST
Tokio · gRPC · async

DATA
PostgreSQL · PostGIS · Redis

ARCHITECTURE
микросервисы · распределённые системы · системный дизайн

PROTOCOLS
REST · WebSockets · gRPC · QUIC

INFRA
Docker · CI/CD · Linux · Git

Контакты

TELEGRAM	@maxeliseyev	EMAIL	maxim.elis@icloud.com
LINKEDIN	in/maximeliseyev	GITHUB	maxeliseyev