

# Арсений Вязников

Python Backend / Full Stack Developer

Специализация: Разработка сложных веб-приложений и бот-систем с нуля

Ключевое отличие: Глубокое понимание фундаментальных технологий (Native JS, Async Python, Linux System), что позволяет мне не зависеть от фреймворков и строить оптимизированные, безопасные решения



## Обо мне

Обладаю опытом полного цикла: от настройки «голого» Debian сервера и написания демонов на Python до реализации сложной клиентской логики на чистом JavaScript и интеграции 3D-графики

## Ссылки

### Контакты

[WhatsApp](#) — 7(993)961-31-12

[Telegram](#) — @SHAMAN\_LESNOY

## Backend (Python Core & Web)

### Languages

Python 3.10+ (Deep understanding of asyncio, threading, OOP, typing)

### Web Frameworks

FastAPI

Quart (Async Flask)

Hypercorn

Aiohttp

### Discord Ecosystem

Disnake

Discord.py (Sharding, Voice Protocol, Interaction API)

### Database

PostgreSQL (SQLAlchemy Async)

SQLite (Aiosqlite, raw queries)

Проектирование реляционных схем

### Cryptography

Cryptography (Fernet, RSA)

Passlib (Argon2)

HMAC

Hashlib

### Libraries

Bleach (sanitization)

Pydantic

Aiofiles

SMTPlib

## Frontend (Vanilla JS & Modern Web)

### Core JavaScript (ES6+)

Глубокое знание DOM API

Event Loop

Promises/Async-Await

Fetch API

### Browser APIs

Web Crypto API (нативная криптография в браузере)

Web Audio API (визуализация звука)

MediaStream Recording API (запись голоса/видео)

WebSocket API

### UI/UX

HTML5

CSS3 (Flexbox, Grid, Adaptive Layout)

Canvas API (обработка изображений на клиенте)

### Frameworks

Vue 3 (Composition API) + Three.js (для специфических задач по 3D)

## DevOps & Infrastructure

### OS

Linux (Debian 12/13) — уверенное администрирование через терминал (Bash)

### Server Config

Nginx (Reverse Proxy, Load Balancing, SSL/TLS)

Systemd (написание кастомных сервисов)

UFW/Iptables

### Tools

Docker

Git

FFmpeg (сложная обработка медиа)



## Разработка защищенного мессенджера (Full Stack: Python + Vanilla JS)

Создание SPA-приложения с архитектурой Zero-Knowledge и E2EE, где упор сделан на нативные технологии браузера для максимальной безопасности и производительности.

### Frontend (Native JavaScript):

Реализовал End-to-End шифрование используя только нативный `window.crypto.subtle` (PBKDF2 для деривации ключей, RSA-OAEP для обмена, AES-GCM для сообщений), без сторонних тяжелых библиотек.

Написал кастомный модуль записи голосовых и видеосообщений через MediaRecorder API с визуализацией звуковой волны в реальном времени через AudioContext и AnalyserNode.

Реализовал загрузку больших файлов с отображением прогресса (Progress Bar) через XMLHttpRequest.

Оптимизировал работу с изображениями: сжатие и конвертация в WebP на клиенте через Canvas API перед отправкой на сервер.

### Backend (Python/Quart):

Написал асинхронный API на Quart + Hypercorn для обработки HTTP/2 запросов.

Внедрил защиту от XSS (через Bleach) и SQL-инъекций (параметризованные запросы).

Реализовал безопасную систему сессий на основе HMAC-подписи и Argon2 хеширования паролей.



## Музыкальная экосистема и Административные боты (Python/Discord.py)

Разработка комплекса микросервисов для управления Discord-сообществом.

### Архитектура и База данных:

Использовал Aiosqlite для высокопроизводительной асинхронной работы с БД (хранение конфигураций, логов, профилей, инвентаря).

Реализовал систему балансировки нагрузки (Load Balancing): один "мозг" управляет пулом ботов-воркеров для обхода лимитов Discord API при воспроизведении музыки.

### Аудио-процессинг:

Глубокая работа с FFmpeg и Youtube-DLP для обработки потокового аудио из разных источников (YouTube, Spotify, SoundCloud).

Реализовал механизм кэширования ссылок и метаданных для ускорения поиска треков.

### Функционал:

Система тикетов, приватные голосовые комнаты с управлением правами, система достижений (генерация карточек через Pillow), модерация.



## Интеграция API: AI и Платежные системы

Написание надежных (robust) клиентов для взаимодействия с внешними сервисами.

### AI Integration (Google Gemini):

Разработал скрипты взаимодействия с LLM, включая управление контекстом диалога (история сообщений) и обработку мультимодального контента (текст + файлы).

Реализовал обработку ошибок и Rate Limiting для стабильной работы с API.

### Fintech API:

Написал универсальный Python-адаптер для работы с несколькими платежными шлюзами.

Реализовал проверку цифровых подписей (Signature Validation) входящих вебхуков для защиты от фрода.

Использовал aiohttp с кастомными политиками повторных попыток (Retry Policy) для работы в нестабильных сетях.



## Telegram Mini Apps (Web3D)

Разработка гибридных приложений: Backend на FastAPI/Aiogram + Frontend на Vue.js/Three.js.

Создание интерактивных 3D-сцен, оптимизированных для мобильных устройств (работа с GLTF моделями, PBR материалами, HDRI освещением).