



## Alexander Strizhakov

Senior Backend Developer

11.06.1986

Dubai, UAE

+971 52 799 0581

[LinkedIn](#) [GitHub](#) [alex.strizhakov@gmail.com](mailto:alex.strizhakov@gmail.com)

Seasoned Elixir Backend Developer with a decade-long track record of success in crafting cutting-edge web applications, enterprise systems, and high-performance distributed systems across diverse industries, such as FinTech, AdTech, and social networking. Boasting extensive experience in all phases of the software development lifecycle, I excel at concept creation, development, code optimization, testing, and delivery. Equipped with the ability to efficiently self-manage during solo projects and collaborate seamlessly as part of a high-performing team. Committed Open Source Software contributor.

## Skills

With a primary focus on web development, I have honed my expertise in Elixir and Phoenix in recent years, solidifying my position as a seasoned Elixir developer.

Proficient in: Elixir, OTP, Phoenix, Phoenix LiveView, Test Driven Development, Git

Actively utilizing: PostgreSQL, Docker, Tailwind CSS, Alpine.js

Familiar with technologies: MySQL, Redis, ClickHouse, Vue.js, React, Amazon AWS

## Employment History

### MiPasa (Jul 2021 - Jun 2022)

As a key member of the MiPasa team, I contributed to the development of a web-based data science platform built on the PETAL stack, streamlining collaboration for individual developers, teams, and organizations. The platform's central feature is a sophisticated notebook editor, enabling users to create and execute Python scripts, import and export data from various file types, and visualize data. It supports real-time data processing in multiple languages and allows for scheduling background executions for intensive computations or time-sensitive results. My accomplishments at MiPasa include:

- Developing and refining the notebook editor, which facilitates code editing, execution, data import and export, chart creation, and other visualizations
- Ensuring the smooth operation, performance enhancement, and bug resolution of connections between the application and distributed execution nodes
- Augmenting testability and test coverage while reducing test execution time by employing ExUnit and asynchronous tests
- Crafting comprehensive integration tests with complex test setups, utilizing the Playwright framework
- Enhancing a robust event system featuring real-time notifications
- Improving data and output processing in both real-time and background modes by leveraging Oban

### Pleroma (Mar 2019 - Mar 2021)

As an integral member of the Pleroma team, I contributed to the development of an open-source social networking platform compatible with Misskey, Pixelfed, Mastodon, and others, built on Elixir and Phoenix. Pleroma employs the ActivityPub protocol for federation, ensuring seamless interoperability between networks. My key accomplishments at Pleroma include:

- Expanding and maintaining the Mastodon-compatible JSON API, as well as developing and refining Pleroma client and Pleroma admin JSON APIs using Elixir, Phoenix, and Ecto
- Designing and implementing dynamic application configuration through the database, enabling process restarts to apply start-time settings using OTP and DynamicSupervisor
- Developing an admin API for database configuration CRUD operations, including Elixir type-to-JSON conversion and vice versa
- Implementing migration functionality for importing and exporting configuration settings between files and databases
- Creating and maintaining two open-source adapters for the Tesla HTTP client library, facilitating federation between nodes

- Enhancing data exchange between federation nodes through a custom connection pool and optimizing memory usage for HTTPS connections
- Developing the Pleroma Installer to streamline the creation of new Pleroma nodes for instance administrators

**Paysale** (Jul 2015 - Mar 2019)

Paysale is a high-performance affiliate network featuring a high-load redirect API and sophisticated targeting rules, developed using Elixir and Phoenix. The redirect API demands low latency and quick response times to function effectively.

During my tenure at Paysale, I accomplished the following:

- Transitioned the Redirect API to the Elixir/Phoenix stack, redesigning it for enhanced scalability and fault tolerance
- Boosted redirect API throughput by up to 10 times, achieving an average latency of 100ms with Elixir and Phoenix, enabling a single server (2 vCPU and 4GB RAM) to effortlessly handle over 15k+ concurrent requests per second (RPS)
- Managed cloud-based infrastructure and oversaw DevOps processes
- Configured deployment to Kubernetes via Rancher, later transitioning to Edelivery and multi-regional server deployment
- Designed and implemented temporary storage solutions for data prior to batch insertion into ClickHouse