

Ilya Mois

Senior Full-Stack Engineer · Удалённо или в офисе в Тбилиси, Грузия (GMT+4)

job.offers@mois.pro · github.com/mois-ilya · linkedin.com/in/moisilya · русский (родной), английский (профессиональный)

О СЕБЕ

Делаю веб-продукты с 2018 года, в последнее время платформы для разработчиков и платежи. В Tonkeeper 21 месяц владел Tonconsole, консолью разработчика TonAPI, включая биллинг и документацию, и поддерживал TypeScript SDK для TonAPI, у которого около 16 тыс. загрузок в неделю. В 2026 году в одиночку сделал Diwy Split, Telegram Mini App, которое делит счёт в ресторане по фото чека: от интерфейса и распознавания на LLM до сервера, на котором оно работает. От прототипа до запуска ушло меньше месяца активной работы.

Строю LLM-пайплайны и выпускаю изменения промптов и моделей только после того, как автоматические эвалы на эталонных данных не показали регрессии. Разрабатываю с агентами внутри процесса из тестовых сценариев, ревью несколькими моделями и скриншот-тестов в CI. К пользователям ничего не попадает, пока не пройдены все проверки.

ОПЫТ

Diwy Split — собственный проект

март 2026 — сейчас · diwy.me · Telegram Mini App для разделения счёта по фото чека · в одиночку, бета

- **Распознавание через LLM** — выбрал связку моделей из примерно 20 по эвалам на размеченных чеках: 0.991 при \$0.004 за чек. Модель только переписывает чек, арифметику делает код и отмечает чеки, в которых не сходится итоговая сумма.
- **Тесты** — правки хранятся как события, поэтому одновременные изменения не теряются, а счёт воспроизводится из истории. Из событий собираются 77 состояний счёта для попиксельных скриншот-тестов в CI; property-based тесты сверяют расчёт долей с эталонной реализацией.
- **Эксплуатация** — сохранил доступ к приложению из России. GeoDNS ведёт пользователей оттуда через релей с TLS passthrough, остальных через Cloudflare. Внешние пробы следят за обоими маршрутами и шлют алерты в Telegram, а данные непрерывно копируются в R2.
- **UX** — прототипировал сценарии до кода, чтобы бумажный чек стал экраном без тупиков, в том числе при выборе только части блюда. Добавил редактор чека для исправления ошибок распознавания.

Tonkeeper — Software Engineer, Developer Platform & Developer Experience

июнь 2024 — март 2026 · удалённо · TON Apps: команда из 23 человек, которая делает кошелёк Tonkeeper (12M+ MAU). Большая часть работы публична, ссылки ниже.

- **Tonconsole**, консоль разработчика TonAPI — перевёл биллинг на единый долларовый баланс по ценам тарифов с пополнением в TON или USDT. Постепенно перевёл состояние на TanStack Query и разделил кэши по пользователям, чтобы аккаунты в одном браузере не видели чужих данных.
- **Продукты консоли** — выпустил вебхуки, выделенные лайтсерверы, тарифы API и аналитику. Благодаря единой функции расчёта цены с тестами трёхкратное изменение цены вебхуков свелось к правке констант. Покупатели лайтсерверов скачивают global.config.json без ручных правок. При понижении тарифа консоль показывает неизрасходованный остаток, а не блокирует переход.
- **Бесплатные лимиты TonAPI для dApps** — спроектировал и сделал предложенную мне идею на всех платформах кошелёка, включая тестовый стенд для QA. Кошелёк встраивает window.tonapi.fetch, SDK подхватывает его без настройки, и dApps внутри Tonkeeper получают повышенные лимиты.
- **TypeScript SDK, ~16 тыс. загрузок в неделю** — единственный мейнтейнер с середины 2024-го. Добавил форматы bigint, адресов и ячеек в [OpenAPI-схему](#) TonAPI, так что любой генератор получает точные типы сумм. При перегруппировке схемы сохранил старые пути SDK как устаревшие обёртки.
- **Интеграционная документация** — генерировал справочник REST из той же схемы. Написал руководства по отслеживанию транзакций и подписи данных и сборник рецептов.
- **Кошелёк** — заменил опрос двух DEX-агрегаторов потоковыми котировками свопов Omniston. Сделал стейкинг в десктопе ([влит после моего ухода](#)). Во время сбоя TronGrid отправка TRC20 теперь сообщает «временно недоступно», а не «недостаточно средств». Ускорил SSR в Tonviewer.

- **Комиссии и расширения кошелька** — написал расчёт комиссий по формулам TON и проверил его на реальных транзакциях (**не закончен к моему уходу**). Работал над переводами, где комиссию оплачивает спонсор (gasless), и расширениями кошелька W5, включая 2FA.

Quintegro — Frontend Developer

июль 2023 — май 2024 · удалённо

- Вёл разработку модуля изменения брони для платформы, связывающей хозяев и гостей.
- Определил API-контракты фронтенда и бэкенда, генерировал из них типы TypeScript, настроил CI.

Minacu — Full Stack Engineer

апрель 2023 — июль 2023 · частичная занятость, удалённо

- Построил потоковое распознавание речи поверх gRPC, анализ текста через OpenAI API и интеграцию с Telegram через tdlb — Python и NestJS на бэкенде, Next.js на фронтенде.

Upper Echelon Products — JavaScript Developer, затем Technical Team Lead

апрель 2021 — март 2023 · удалённо

- Стал тимлидом: вёл 3–5 инженеров и 1–2 QA примерно через 11 релизов, был ментором пяти джунов.
- Построил пайплайн сбора данных из Amazon Seller Central и перевёз бэкенд с VPS на AWS с CI/CD. Привёл фронтенд-архитектуру к единому виду и перевёл документацию API на Swagger.

Apptech — Software Engineer

июль 2020 — ноябрь 2020 · частичная занятость, Санкт-Петербург

- Сделал пуш-уведомления и чат в реальном времени на Firebase для второй версии мессенджера.

GET Information Technology — Software Engineer

март 2018 — февраль 2020 · Санкт-Петербург

- Делал интерфейсы на Backbone/Marionette и ExtJS и сервисы на ColdFusion, отдававшие им данные.
- Внедрил ESLint и Babel и сделал инструменты, отслеживавшие их распространение по коду.

ИЗБРАННЫЙ OPEN SOURCE

- **TON Connect SDK**, библиотека, через которую dApps подключают кошельки TON — реализовал **подпись данных** для текста, бинарных данных и TON-ячеек в протоколе, SDK и UI. Добавил возможности кошельков: **обязательные** отсекают неподходящие кошельки, **предпочтительные** ранжируют их и предлагают переподключиться вместо молчаливой ошибки.
- **Спецификация TON Connect** — написал **эталонную реализацию подписи данных** с тестами. Они выявили два места, где кошельки, соответствующие спецификации, могли подписать разные сообщения для одного запроса. Оба исправлены в спецификации: **формат адреса** в ответе и **кодирование appDomain**, где ошибался даже пример из самой спецификации.
- **Telegram Bot Docs skill** — скилл, который делит документацию Telegram по ботам примерно на 1000 разделов с поиском и проверкой актуальности, чтобы агенты читали свежие доки, а не память.

НАВЫКИ

- **Языки и фронтенд** — TypeScript, JavaScript, Python; Go (чтение и мелкие правки); React, Next.js
- **Бэкенд и API** — Node.js, Fastify, NestJS, REST, OpenAPI, GraphQL, gRPC, вебхуки, OAuth/OIDC, JWT
- **Данные и очереди** — PostgreSQL, MySQL, MongoDB, Redis, RabbitMQ, event sourcing
- **Облако и эксплуатация** — Cloudflare (Workers, Durable Objects, R2, Pages), AWS, Azure AI, Docker, GitHub Actions, NixOS, Prometheus, Grafana
- **Тесты** — Vitest, Playwright (попиксельные скриншот-тесты), fast-check, Testcontainers
- **Области** — платформы и SDK для разработчиков · платежи и биллинг · Telegram Mini Apps и боты · блокчейн

ОБРАЗОВАНИЕ

MSc, High Technology and Economics of Innovation

2019 — 2021 · Университет ИТМО

BSc, Information Systems and Technologies

2015 — 2019 · РГПУ им. Герцена