

Ilya Mois

Senior Software Engineer · Remote or on-site in Tbilisi, Georgia (GMT+4)

job.offers@mois.pro · github.com/mois-ilya · linkedin.com/in/moisilya · Russian (native), English (professional)

SUMMARY

I take an area whole and make it dependable for the people on the other side of it: at Tonkeeper that was the TonAPI developer console and the TypeScript SDK, mine alone for 21 months, plus the API schema and documentation developers built against; this year it is Diwy Split, a Telegram Mini App taken from idea to production.

Where a system can hold a guarantee, I would rather it did than rely on anyone's attention — mine, a colleague's or a model's. So things that must agree are generated from one source instead of kept in step by hand, the states a thing can be in are catalogued instead of remembered, and what would otherwise be judged by eye is compared against a reference or scored. The same habit now goes into tooling for AI agents: skills that hand them freshness-checked documentation, and LLM pipelines that are measured and cost-capped.

EXPERIENCE

Diwy Split — side project, shipped

Mar 2026 – Present · diwy.me · Telegram Mini App, live in beta · solo; no company or funding behind it

- Photograph a restaurant receipt in a group chat; anyone in the chat taps what they ate, in their own language, and sees their share — nothing to install, no sign-up, no names to type.
- LLM recognition in a single linear pass: each stage names its model and fallbacks behind a gateway, spend is capped per user and per day, and a receipt whose arithmetic fails goes to the user for review rather than to another model.
- Built a benchmark before settling on that design: every test receipt has hand-checked ground truth, and each pipeline variant is scored against it. A second model with an arbiter caught nothing the single pass missed, so it was dropped.
- Every request, including the WebSocket, carries Telegram's signed init data, and a bill is visible only to the people in it. CI runs unit tests and screen snapshots across four device profiles on every change.

Tonkeeper — Software Engineer, Developer Platform & DX

Jun 2024 – Mar 2026 · Remote · TON Apps, the 23-person team behind Tonkeeper, a wallet with 12M+ MAU. Most of this work is public and linked below.

- **Developer console** — sole developer for 21 months. Moved billing from TON to USDT without forcing existing users through a cutover, and [replaced the MobX state layer](#) with React Query to stop duplicate events and server requests.
- **Console product work** — shipped webhook management and pricing, API tiers, billing history, the analytics query builder and airdrop tooling.
- **TypeScript SDK, ~16k weekly downloads** — sole maintainer from mid-2024. Rebuilt its packages and generator to preserve int64 values as bigint, handle OpenAPI serialization edge cases, and emit one client for browsers and Node with a single runtime dependency.
- **API schema and integration docs** — changed the parts generated clients exposed as ambiguous: integer and date formats, response types and operation grouping. Wrote guides on transaction tracking, data signing and hash normalization.
- **Wallet** — shipped non-liquid staking flows, SSE-backed swap work and TON Connect fixes across web, desktop and extension; made third-party API failures recoverable instead of letting them crash the app.

Quintegro — Frontend Developer

Jul 2023 – May 2024 · Remote

- Led development of the booking-change module for a reservation platform connecting hosts and guests.
- Defined frontend-backend API contracts, generated TypeScript types from them and set up CI.

Minacu — Full Stack Engineer

Apr 2023 – Jul 2023 · Part-time, Remote

- Built streaming speech-to-text over gRPC, text analysis through the OpenAI API and a Telegram integration through tdlb — Python and NestJS on the backend, Next.js on the frontend.

Upper Echelon Products — JavaScript Developer, then Technical Team Lead

Apr 2021 – Mar 2023 · Remote

- Promoted to lead; took 3–5 engineers and 1–2 QA through about 11 releases and mentored five junior developers.
- Built an Amazon Seller Central ingestion pipeline and moved the backend from a VPS to AWS with CI/CD. Standardised the frontend architecture and moved API documentation to Swagger.

Apptech — Software Engineer

Jul 2020 – Nov 2020 · Part-time, St Petersburg

- Built push notifications and real-time messaging on Firebase for the second version of a messenger app.

GET Information Technology — Software Engineer

Mar 2018 – Feb 2020 · St Petersburg

- Built Backbone/Marionette and ExtJS interfaces and the ColdFusion services that supplied their data.
- Introduced ESLint and Babel, including tooling that tracked their rollout across the codebase.

SELECTED OPEN SOURCE

- **TON Connect specification** — clarified two places where compliant implementations could sign different messages for the same request: the address format in a sign-data response and `appDomain` encoding. Both changes are in the [protocol specification](#).
- **Data signing** — implemented `SignData` in `@tonconnect/sdk`, covering text, binary and TON cell payloads, and published a readable [TypeScript reference implementation](#).
- **Wallet capabilities** — added `requiredFeatures` and `preferredFeatures` to the SDK, so apps can declare what they need and incompatible wallets are filtered before connection.

SKILLS

- **Languages & frontend** — TypeScript, JavaScript, Python; Go (reading and small fixes); React, Next.js, React Native
- **Backend & APIs** — Node.js, Fastify, NestJS, PostgreSQL, OpenAPI, REST, gRPC, webhooks, event sourcing
- **Cloud & testing** — Cloudflare Workers, Durable Objects, AWS, Docker, CI/CD; Vitest, fast-check, Testcontainers
- **Domains** — developer platforms and SDKs · payments and billing · Telegram Mini Apps and bots · blockchain

EDUCATION

MSc, High Technology and Economics of Innovation

2019 – 2021 · ITMO University

BSc, Information Systems and Technologies

2015 – 2019 · Herzen State Pedagogical University