Pavel Pro.

📧 job@mankey.ru 🔗 linkedin.com/in/pavel-pro

Senior/Lead Software Engineer, 10+ years in HL/HA enterprise.

Technical Skills

- **Frontend:** Vue.js v3 and v2 (since 2016), Vuex, Vue Router, TypeScript, SCSS, Less, TipTap, ProseMirror, D3.js, Chart.js, Vuetify, Quasar
- **Backend:** Node.js (since 2017), PostgreSQL/Oracle, Elasticsearch, Kafka, Redis, Express.js/Nest.js
- **Desktop:** Electron (since 2017)
- **Infrastructure & Delivery:** Vite, Rollup, Webpack, Docker, GitLab CI/CD, GitHub Actions
- **Testing:** Jest, Cypress (unit, component, E2E);
- **Other:** Metadata-driven systems, Micro-frontends, NPM package publishing, Legacy modernization, AI tools and LLMs integration, TypeScript migrations;

Professional Experience

Progressed from Junior to Senior/Lead roles at Russian Railways (RZD.ru), Nov 2011 – Apr 2026

- Frontend
  - ▸ High-Load Ticket Platform
  - ▸ UARM-FE: RZD Admin Dashboard
  - ▸ wFlow: Kanban-style project management app
  - ▸ MCC: deployment/delivery manager
- Backend
  - ▸ CSM API Adapter
  - ▸ Version Informers (IVI + RVI)
- Using AI
- Education
- Language Skills
- Location
- My Priorities
- Architecture
- Infrastructure
- Testing
- NPM Packages
- TypeScript Coverage
- Desktop

Frontend

Built 30+ frontend apps only with modern stack:

- Vue.js v2 and v3, Vuex, Vue Router;
- Typescript;
- D3.js, Chart.js;
- Vuetify, Quasar;
- Scss, Less;

High-Load Ticket Platform Migration

Scale: 20+ million visits per month, ~150K *paid* tickets per day.

Critical frontend migration from custom jQuery-based framework to Vue.js

- Impact: one SPA instead of multiple pages. ~3x less traffic and server load, improved performance, stability, UX, maintainability, shipment speed, and a modern tech stack for the future
- Complexity handled:
  - Business logic implemented in frontend via JS and metadata-driven server templates;
  - Thousands of stations across 11 timezones; intercity, international, and commuter trains and buses;
  - Tens of extra services; 3 languages support;
  - Hundreds of scenarios;

UARM-FE: Russian Railways Admin Dashboard Frontend App

Modern replacement for legacy admin dashboard

- Before: JSP-based IBM WebSphere stateful app (built by Java developers);
- After: Modern async Vue.js application;
- Architecture: Dynamic component rendering based on declarative metadata;
- Result: Significantly improved performance and developer experience
- Adoption: 1000+ users;
- Complex build pipeline;

wFlow: Kanban-style project management app

Enterprise-grade project management frontend

Adoption & Scale:

- Adopted company-wide: 70+ users migrated from Jira;
- Deployed at Russian University of Transport: 200+ users;

Features:

- Drag-and-drop Kanban boards;
- Custom workflow configurations;
- Custom issue fields;
- Issue linking and file attachments;
- Custom WYSIWYG JSON editor (built on TipTap/ProseMirror);

Learn more about wFlow [RU]

MCC: deployment/delivery manager

MCC stands for Mission Control Center

An internal app, which is responsible for CI/CD: rollout of container images, launching and validating automated test results, releasing software distributions, reviewing and approving Merge Requests (MRs), and generating release documentation together with two Node.js services mentioned below. It is a critical part of the company's software delivery process, and is used by multiple teams across the organization.

Backend

Have strong experience in backend development and infrastructure management.

Developed 7+ Node.js services with:

- Elasticsearch (indexing, search, logging);
- Apache Kafka;
- PostgreSQL;
- Redis;
- Express.js / Axios;
- "Retry/backoff" strategies;
- JWT and LTPA authentications;
- Logging;
- DOCx and XLSx generation;
- Integrations with various third-party APIs: JSON and XML;
- All components are dockerized

CSM API Adapter

The mission-critical Node.js backend service for RZD (Russian Railways) CSM (Support center for passengers with reduced mobility). A hub for processing four distinct types of handicapped passengers applications with complex data models; integrates the following systems:

- RZD website https://www.rzd.ru/ru/9290 — Kafka
- RZD administrative interfaces — Kafka
- EKMP ACS (Unified Customer Mobile Application) — JSON REST API
- TTK ACS (TransTeleCom) — JSON REST API
- SIOP FPK ACS (Passenger Identification and Service System, Federal Passenger Company) — JSON REST API

Bidirectional synchronization across the full application lifecycle, notifications to users and administrators, and more.

Version Informers (IVI + RVI)

Two backend services: IVI (company side) and RVI (deployed on the customer’s server — Russian Railways). The services fully automate the software release lifecycle — from merge request acceptance to release publication and installation across the server environment chain (staging environments through to production).

Key Features & Responsibilities:

- Developed and maintained integrations with: Wflow (task tracker), MCC (installation manager), RVI, PostgreSQL, FTP, SVN, GitLab, Mattermost, and Kibana.
- Implemented structured logging to Elasticsearch with real-time monitoring.
- Configured smart notifications in Mattermost across multiple channels, including links to rerun forms, Kibana logs, and error details.
- Achieved high test coverage with comprehensive end-to-end and unit tests (Jest).
- MR Acceptance Handler: Validates issue fields in the task tracker, checks for duplicates, gathers contract/epic data, identifies changed repositories, records data in IVI database, and forwards it to RVI via REST API.
- Release Publication Handler: Detects “lost” issues, downloads and unpacks the distribution, generates customer-facing artifacts (6 document types) and a release composition JSON for reliable data transfer to RVI (fallback when REST API is unavailable).
- Installation Event Handler: Records successful installations/rollbacks on server, performs bulk status updates for releases and linked issues via Bulker API, and adds rollback comments.

Architecture

Designed various solutions on multi-domain basis: backend (app layer + DB layer) and frontend.

Major Redesign of RZD.RU National Portal (Russian Railways)

Led the complete technological redesign and migration of the RZD.RU national portal as part of a large-scale transition from legacy Oracle + IBM WebSphere stack to a modern Postgres, Kubernetes, and Docker-based architecture.

Project Scale: ~1,000 user-facing pages across ~50 subdomains and ~500 editorial CRUD interfaces / admin workstations.

Key Contributions:

- Architecture: Designed the frontend architecture and collaborated closely with Java architects on new metadata formats, enabling a modern headless metadata-driven CMS [RU] approach.
- Metadata Migration: Orchestrated the migration of 1,500+ legacy XML files (heavily burdened with technical debt and workarounds) into clean, maintainable JSON structures. Developed conversion scripts and built MetaWizard — a powerful metadata editor that became the central tool for the entire migration and ongoing maintenance.
- User-Facing Portal: Implemented the foundational frontend solutions, including the new site theme and complete redesign of the most critical and complex pages. This baseline enabled the rest of the frontend team to efficiently build out the remaining functionality.
- Editorial Interfaces: Designed and developed the new admin application (UARM-FE), delivering the majority of UI controls along with several of the most complex editorial interfaces.

Desktop

MetaWizard Electron App

Complex internal desktop application (Windows/Mac/Linux).

- Designed and developed in 2017;
- Maintained for 9 years then, while keeping up with the latest Electron and Node versions since 1.x and v.0.x respectively

Key Features:

- Metadata editors and lists;
- Database connectivity (Oracle, PostgreSQL);
- SSH/SFTP operations;
- Complex file operations;
- Multi-repository structure with integrated admin dashboard app;

Infrastructure

Developed and maintained complex full-cycle frontend and backend pipelines:

- **Build:** Vite & Webpack (incl. Code-Splitting/Module Federation/DLL), Babel, tsc/tsx, SCSS/Less,Eslint, Prettier, Husky, Lint-Staged etc
- **Delivery:** Docker, GitLab CI/CD, GitHub Actions

Testing

Implemented testing for multiple projects (frontend, backend, desktop):

- Unit, component and E2E
- Jest and Cypress

NPM Packages

- Published 10+ custom packages to the company's private registry
- Frontend and backend, used across multiple company projects:
  - Frontend/Backend libs for auth in different systems (JWT, Blitz, LTPA)
  - Unified (company standard) libs for:
    - Error handling
    - Logging (ELK)
    - Kafka integration
    - Mattermost notifications
  - Frontend and backend utilities for common tasks
  - Shared configs for ESLint, Prettier, etc.
  - Frontend/Backend libs for integrating with various systems
- Provided various build pipelines and CI/CD

TypeScript Coverage

Led TypeScript migrations for multiple projects, improving code quality and maintainability, reducing runtime errors and enhancing DX. Usually step-by-step, with gradual migration and strict typing enforcement.

Using AI

Using various LLMs with various purposes and approaches.

Maintaining balance between code control and development performance.

When applicable - practicing agentic workflows and Spec-Driven Development (Openspec).

For other purposes - sure: suggestions, configuration, documentation, debugging, testing, quick prototyping, etc.

Education

Bachelor’s Degree in Geological Engineering

Peoples’ Friendship University of Russia (RUDN University), Moscow, Russia, 2005 – 2009

Language skills

Fluent English, native Russian, beginner German, French and Ukrainian.

Location

Remote-first since 2013, now from Thailand. *Very* flexible with timezones (up to GMT+1).

Open to hybrid jobs if needed, as well as relocation/on-site.

My Priorities

- Teamwork - Collaborating, constantly learning while sharing knowledge with others as mentoring and leading
- Clear design - Architecture, tasks, code etc.
- Building solutions - Either maintainable apps for years ahead OR quick PoC/MVPs
- Balance - Business purposes (first priority), developer experience, and managing tech debt