Konstantin Zamyatin

Senior Software Engineer and Architect

Backend and platform systems | AI-orchestrated engineering | Hosting and domain-registrar systems

career1@zamyatin.org - Telegram: @zkremote
linkedin.com/in/konstantin-zamyatin - zamyatin.org

PROFILE

24 years building production systems and 20+ years architecting systems for a European hosting / domain-registrar provider. In recent years AI-driven engineering has become my primary working surface, materially expanding what I can do as a solo operator: concurrent multi-perspective design review, parallel adversarial verification of work, sustained context across multi-day projects. The same practice can scale beyond solo work: I can build the workflows, conventions, and quality gates that let a team adopt AI-orchestrated delivery. Recent production work shipped through this practice includes library extraction and adoption, modernization passes on legacy core systems, and new service delivery across multiple codebases.

Commercial work has been primarily in PHP, mostly in long-running custom-framework systems rather than Symfony-first projects, but the polyglot background (Java for Android, JavaScript, C++, Bash, and other stacks at lighter depths) plus the agent-driven workflow makes language a smaller constraint, and I pick up new stacks quickly when needed. Strong fit for AI-orchestrated engineering at production scale, modernization of long-running platform stacks (code + infrastructure together), and async, written-channel cultures.

AI ENGINEERING PRACTICE

AI-orchestrated engineering has shifted from supplement to my primary surface for shipping production work. The practice covers:

- **Concurrent multi-perspective review on every piece of work**: each design or change gets independent design review, parallel adversarial verification, and alternative synthesis before completion. A real capability expansion over what I could do single-handed in sequence; implementation work also runs through this practice rather than through a separate team
- **Spec-driven and task-driven delivery** with explicit quality gates at each step (spec authoring, planning, breakdown, implementation, review, completion); every task ends with mandatory static analysis (PHPStan level 8 baseline for new code) and the full test suite
- **Sustained multi-day orchestration**: persistent orchestrators with snapshot, resume, and handoff; long-context session management; per-task focused-context construction. Work continues across days without context loss

This practice is supported by personal tooling I maintain on my own infrastructure for daily use - not a separate product, but methodology and skills I bring to my engineering work. It is also team-transferable: I have built shared engineering conventions consumed by both developers and agents, and can set up, support, and evolve the same workflows, tooling, and quality gates for a team.

Representative work shipped through this practice (across multiple PHP codebases - a mix of long-running custom-framework legacy systems and newer projects built on Symfony components through a shared in-house library layer):

- Shared PHP framework library, built to modernize a portfolio of long-running legacy projects. First adopter: the e-commerce platform (2025). The library then co-evolved with the new DNS control-panel REST backend - first by adding building blocks the backend needed during its construction, then by extracting reusable modules back from the backend once it stabilized, into the library for the next consumers. Now in progress: Twig integration to support the e-commerce platform's template migration; migration of the core ERP to the same modernized stack
- New multi-tenant DNS control-panel REST backend designed and delivered: hardened authentication (JWT + two-factor + token blacklist + login brute-force protection), external authoritative-DNS integration, comprehensive acceptance and integration test coverage
- PSR-4 namespace restructuring of a 20-year-old core ERP codebase, alongside its ongoing migration to the modernized library stack
- Architectural refactors, security hardening, and test-infrastructure expansion across a portfolio of shared internal libraries (e.g. a DNS-records library brought to 94% test coverage)

Tools: Claude Code (daily driver), OpenAI Codex CLI; GitLab APIs and CLI for project integration.

SELECTED EXPERIENCE

Software architect, remote (2024 to present)

Remote engineering work for a European hosting / domain-registrar provider (name confidential, available on direct request).

- Continued architectural ownership of the provider's core systems (ERP, e-commerce, hosting control panel) and the evolving library stack underneath them - business-critical platforms serving tens of thousands of customers
- Designed and delivered a new DNS control-panel REST backend with reseller-aware multi-tenancy: tenant-scoped authentication, automatic per-tenant data scoping, and integration with the existing ERP reseller model
- Driving the shared PHP framework library through multi-version evolution and rolling it out across consumer projects
- Modernization passes on legacy core systems: PSR-4 namespace restructuring, PHP 8.x upgrades, Symfony components adoption, PHPStan level 8 baseline for new code
- Applied AI-orchestrated engineering across planning, implementation, review, verification, and multi-day delivery work

Founder / CEO / lead developer - Medvedomen LLC (2020 to 2024)

Founded and operated a development company delivering engineering work for a long-term European client.

- Hired and led a small distributed development team for client project work; set quality standards, dev processes, and tech-stack choices
- Introduced auto-testing and Docker as the dev-environment standard for the client's projects
- Established the team's internal knowledge base and documentation practice
- Cross-system PHP modernization on long-running platforms: PHP 7 to 8.x version migrations across the hosting control panel, ERP, and e-commerce stack
- Led design and delivery of a two-factor authentication system integrated into a hosting control panel
- Led design and delivery of an internal BIND zone-management subsystem
- Provided multi-team architectural review and design guidance: ~150 merge requests reviewed, ~1,400 review comments authored (peak 2023); API designs, code conventions, and architecture decisions worked out through written review threads

Lead developer / technical lead - European hosting / domain-registrar provider (2006 to 2020)

Name confidential, available on direct request.

- Owned overall architecture, databases, and ongoing development for the provider's product suite: a core ERP, an e-commerce platform, and a hosting control panel
- Set the technical direction for a small development team (typically 4-6 engineers): technology selection, development methods, deployment approaches - while remaining the primary hands-on developer
- 2006 to 2010: rewrote the legacy ERP as a distributed system with reseller support; designed the core database that has carried the business for many years since
- 2009 to 2011: built v2 of the hosting control panel with a seamless migration scheme from v1
- 2013 to 2014: led a multi-month migration project moving the hosting fleet to new infrastructure
- 2017: integrated a SaaS site-builder homepage product into the control panel; integrated Let's Encrypt SSL automation across hosted customers
- Migrated end-user bases from acquired hosting businesses into the receiving platform

Lead developer - European hosting / domain-registrar provider (2002 to 2006)

- Architected and built the first ERP (domain ordering, hosting orders, user management) within a tight initial deadline
- Architected and built the first end-user hosting control panel
- Wrote C-language patches for Apache web-server modules to meet hosting-specific needs
- Linux services administration alongside development: Apache, BIND, MySQL, Qmail, sqwebmail

Independent contractor (1999 to 2002)

Independent web development for international clients (UK, Russia, others). LAMP stack. Mix of solo and team work, several projects included end-to-end architecture and database design.

SIDE PROJECTS

- **DartScore** (`net.zk_labs.dartscore` on Google Play): sole-author Android app for darts scoring. Originally released in 2016; modernized periodically over the years to current Android SDKs (most recently compileSdk 34 / Android 14). Java; ~110 source files; Firebase Analytics; MPAndroidChart
- **Personal homelab and infrastructure**: Proxmox VE hypervisor with a mix of LXC and KVM guests (Linux Debian / Ubuntu / Alpine), Proxmox Backup Server, Synology NAS, Docker across multiple personal services, Node-RED and Home Assistant for home automation. Several years of self-managed home infrastructure plus assorted personal scripting (home-server automation, media-library tools)
- **Open-source era (1998 to 2003)**: Linux / KDE applications published on SourceForge during university and early-career years - Akkord (a Norton-Commander-style two-panel file manager in C++/Qt, actively developed 1999-2000), Kife (KDE/Qt frontend for kinstall), kinstall (Perl utility for Linux), Btlog (MS DOS C++ log processor). Sole author across all four

SKILLS

Hosting / domain-registrar industry (20+ years): domain ordering and management, hosting provisioning and control-panel UX, DNS subsystems (BIND zone management, authoritative DNS), SSL/TLS automation (Let's Encrypt), reseller-aware multi-tenant architecture, customer migrations between acquired hosting platforms, ERP for a hosting business

Programming languages: PHP (24y, expert), JavaScript (legacy systems, integration work), SQL (24y, strong), Java (Android, ~10y), Bash (competent, used regularly), C++ (early career, ~5y), Perl (early career), C# (occasional), Python (light). Polyglot background; comfortable picking up additional stacks via agent-driven workflows

Frameworks and tooling: long-running in-house PHP framework, Symfony components, Doctrine ORM, Twig, PHPUnit 12, PHPStan (level 8), Composer, OpenAPI

Architecture and platform thinking: reseller-aware multi-tenancy, shared-library extraction and versioning, live-data migration schemes, code + infrastructure modernization, test-infrastructure design, exception-hierarchy and error-key conventions, OpenAPI-spec-driven REST design, event-driven and SOAP integrations

Infrastructure: Docker (strong, daily, dev-environment standard across multi-project portfolio; also primary surface for personal homelab services), GitLab CI, Linux (Debian, Ubuntu, Alpine; 25+ years), Proxmox VE / virtualization, shell scripting, BIND, Apache, MySQL / MariaDB, SOAP (long-running production integrations), Let's Encrypt automation. Self-deployment of all my own projects throughout my career

AI / agent tooling: Claude Code (daily driver), OpenAI Codex CLI; multi-agent orchestration patterns (concurrent multi-perspective design and code review, parallel adversarial verification, alternative synthesis, persistent orchestrators with snapshot / resume / handoff, pipeline and parallel agent harnesses); structured-output / schema-validated agent calls; long-context session management; quality-gate-enforced agent loops; GitLab APIs and CLI

Mobile: Android (Java); long-running published personal app

Languages: English (strong written, primary working language; spoken less practiced), Russian (native)

WORKING PREFERENCES

- **Fully remote, no relocation.** Based in Nizhny Novgorod, Russia (UTC+3, Moscow time zone)
- **Strong preference for async, written-channel collaboration:** GitLab / Jira issues, MR or PR reviews, email, chat. 25+ years of distributed work with European teams through written channels; my strongest working mode is written English and documentation-heavy engineering. I do not use video or voice calls during the working day. Best fit: async-first organizations, written-first documentation cultures, distributed-by-default teams
- **Engagement type:** available as contractor (B2B) or full-time-equivalent

EDUCATION

1996 to 2001: Master's degree in Computer Science, Ryazan State Radio Engineering Academy (now Ryazan State Radio Engineering University). Diploma with distinction. Specialty: "Information Systems in Economics". GPA 4.75 / 5