

Paulius Labanauskis

Principal Engineer & Technical Architect

paulius@labanauskis.is | +370 681 34437 | [LinkedIn](#) | [GitHub](#) | Lithuania

Professional Summary

Fintech-specialized technical leader with 15+ years of experience building financial infrastructure across banking, banking-as-a-service, and digital assets. Transforms systems into high-performance platforms and builds regulated financial services from scratch, including compliance infrastructure, transaction engines, and multi-provider integrations. Builds lean, high-efficiency engineering teams focused on delivery over process.

Career Highlights

Fintech Domain Expertise: 6+ years building critical financial infrastructure across digital assets, banking-as-a-service, and traditional banking, serving platforms with millions of users in FCA and EU-regulated environments

Architecture Leadership: Designed event-sourced financial platforms from scratch, including multi-custody, multi-bank architectures with integrated compliance screening and audit logging

Performance Impact: Resolved critical database failures reducing API latency from 60s timeouts to under 200ms; designed architectures scaling throughput from 9 to 1,000 TPS

Delivery Under Pressure: Led emergency banking migration in four weeks with zero customer disruption; took StablR platform to production within six months of joining

Experience

Principal Software Engineer | *StablR* | 2023 - Present

Impact: Designed and built an EU-regulated institutional digital asset platform from scratch, event-sourced transaction engine, multi-custody and multi-bank architecture, full compliance stack. Individual transactions regularly in the tens of millions of euros.

- **Platform Architecture:** Designed the full system using event sourcing, CQRS, and DDD, bounded contexts for identity, financial data, transaction processing, compliance, and governance. Personally architected and implemented the majority of the codebase.
- **Transaction Engine:** Built event-sourced order aggregate orchestrating the full transaction lifecycle across fiat AML screening, crypto KYT screening, mint/burn execution, and fiduciary transfers. Migrated from legacy Step Functions orchestration to this architecture using a strangler fig pattern.
- **Compliance & Integration:** Integrated KYT (Chainalysis), AML screening (ComplyAdvantage), three digital asset custodians, three banking partners, and four blockchain networks. Built unified webhook gateway and full audit logging pipeline.
- **Delivery & Team:** Took the platform to production within 6 months of joining, establishing all engineering practices, CI/CD, infrastructure-as-code, and monitoring. Built and scaled the engineering team from early stage.
- **Technical Stack:** C#, .NET 8, AWS (Lambda, API Gateway, Step Functions, SQS/SNS, DynamoDB, S3), PostgreSQL, Marten (event sourcing), Terraform

Senior Software Engineer | *CloudKitchens* | 2022 - 2023

Impact: Built developer automation infrastructure for incident management and platform operations at Travis Kalanick's post-Uber venture

- Created automation tools for incident management, SLO compliance, and platform monitoring as part of an internal developer tools team
- Designed deployment automation reducing operational overhead for the platform engineering group
- **Technical Stack:** Java, Kubernetes, Docker, CockroachDB, GCP

Technical Lead | *Railsr (Railsbank)* | 2019 - 2022

Impact: Early engineering hire at FCA-regulated BaaS platform serving 500K daily active users. Company grew from ~20 to nearly 500 employees during this period.

- **Emergency Banking Migration:** When banking partner terminated partnership with minimal notice, led migration of customer sort codes and account numbers onto Bank of England payment schemes (Faster Payments, BACS, CHAPS). Completed in four weeks with zero customer disruption.
- **Performance Turnaround:** Diagnosed critical Datomic database bottlenecks by studying internals independently, reducing API response times from 60s+ timeouts to under 200ms at p95. Designed event-driven read-replica pipeline to PostgreSQL, bringing read latency under 100ms and eliminating cascading failures.
- **Scalability:** Created and led the Platform Performance team. Designed transaction processing architecture scaling from 9 TPS (Datomic-constrained baseline) to 1,000 TPS, validated through benchmarking.
- **Compliance & Process:** Implemented KYC/AML integrations (Onfido, ComplyAdvantage) and transaction monitoring. Introduced Kanban methodology, established API standardisation, and interviewed majority of early engineering hires during rapid growth.
- **Technical Stack:** Clojure, Datomic, AWS (ECS, SQS, SNS, DynamoDB), PostgreSQL, Terraform

Technical Lead | *Adform* | 2017 - 2019

Impact: Led architecture modernisation of MRC-licensed campaign management platform

- Adopted event-sourcing and event-driven patterns to resolve performance bottlenecks, building on the DDD/CQRS foundation established in the previous role
- Managed platform, frontend, and backend teams through the technical transition
- Ensured platform met Media Rating Council accreditation requirements
- **Technical Stack:** C#, Microsoft SQL Server, RabbitMQ, Kafka, Puppet

Software Architect | *Danske Bank* | 2016

Impact: Designed cloud service architecture for mobile banking platform serving ~3.5M users across Finland, Sweden, Norway, and Denmark

- Created cloud-based abstraction layer over core banking mainframe, enabling distributed development across multiple teams
- Led backend team of 15 engineers, coordinating with frontend, mainframe engineering, document management, and compliance groups as part of the bank's broader modernisation programme
- **Technical Stack:** C#, RabbitMQ, IBM API Connect, Oracle DB

Senior Software Engineer | *Adform* | 2013 - 2016

Impact: Transformed online banner management platform through domain-driven design implementation

- **Architecture Transformation:** Key contributor to DDD and CQRS transformation of the campaign management platform, decomposing the monolith into bounded contexts with separated read and write models
- **Deployment Innovation:** Team achieved shift from single weekly release to up to 30 releases per day across a 6-developer team
- Implemented fully automated test suite ensuring reliable continuous deployment
- **Technical Stack:** C#, Microsoft SQL Server, RabbitMQ, Kafka, Puppet

Software Engineer | *Bentley Systems* | 2010 - 2013

Impact: Developed extensions and complementary products for MicroStation CAD system

- Built extensions for enterprise CAD software used in engineering and construction
- Transitioned from desktop to mobile/web development with rigorous TDD approach
- **Technical Stack:** C#, C, SQLite, Android, Web APIs

Education

Master of Science, Computer Science | *Vilnius University* | 2013

Graduated Cum Laude - Research focus: Concurrent algorithms for theoretical homogeneous many-core CPU architectures (up to 1024 cores)

Core Technical Expertise

Languages: C# (.NET 6/8), Clojure, Java, Go, Python, SQL

Architecture: Event Sourcing, CQRS, Domain-Driven Design, Microservices, Event-Driven Architecture

Cloud & Infrastructure: AWS (Lambda, API Gateway, Step Functions, SQS/SNS, ECS, DynamoDB, S3), GCP, Docker, Kubernetes, Terraform

Fintech & Compliance: KYC/AML (Onfido, ComplyAdvantage, Chainalysis KYT), UK/EU payment schemes, transaction monitoring, audit logging, regulatory reporting, multi-bank payment routing

Blockchain & Custody: Ethereum, Solana, Concordium, multi-custody architecture (Fireblocks, Dfns, Hadron)

Data: PostgreSQL, Datomic, DynamoDB, CockroachDB, RabbitMQ, Kafka

What Sets Me Apart

Fintech Domain Depth: 6+ years across three segments of financial services, traditional banking (Danske Bank, 3.5M users), BaaS (Railsr, 500K DAU, FCA-regulated), and institutional digital assets (StabIR, EU-regulated). Hands-on with compliance infrastructure, payment schemes, and multi-provider integration, not just at awareness level.

Building From Scratch in Regulated Environments: Twice built financial platform architecture from zero to production under regulation (StabIR, Railsr), including compliance infrastructure and external provider integrations where there was no existing playbook.

Performance Under Pressure: Led an emergency banking migration in four weeks. Resolved database performance failures that had resisted prior attempts. Designed architectures scaling throughput by two orders of magnitude.

Lean Engineering Philosophy: Builds small, focused teams that deliver high impact with minimal overhead, shipping production systems, not processes.