

BATYRKHAN SHAKENOV — PORTFOLIO

+7(707)775-4575 ♦ Almaty, KZ

batr1996@gmail.com ♦ <https://shakenov.com> ♦ github.com/devbatyr

B-CONNECT — BANKING INTEGRATION PLATFORM

Role: Senior Software Engineer / Tech Lead

May 2021 – Present

Company: Bereke Bank (formerly Sberbank Kazakhstan)

Stack: Java 17/21, Go, Spring Boot, Apache Kafka, Kubernetes, Docker, Helm, OpenSearch, Prometheus, Grafana, GitLab CI/CD, Redis, SMPP

Overview:

B-Connect is the central integration platform of Bereke Bank, responsible for handling all communication between internal banking systems and external services. The platform consists of **87+ active microservices** (Java and Go) deployed across DEV, STAGE, and PROD Kubernetes clusters. Every transaction, notification, and inter-system request flows through B-Connect.

Key Responsibilities:

- Led a team of 4–6 developers in designing, building, and maintaining the integration platform.
- Architected microservices for integrations with card processing systems, credit scoring platforms, core banking, anti-fraud and AML systems, CRM, collection, onboarding, and 30+ other internal and external systems.
- Designed and implemented communication protocols: REST, SOAP, SFTP, SMPP, SMTP, and Kafka-based async messaging.
- Built centralized CI/CD pipelines in GitLab CI for automated build, test, Docker image creation, and Helm deployment across environments.
- Implemented comprehensive observability: OpenSearch for log aggregation, Prometheus + Grafana for metrics, and per-service health endpoints.
- Conducted code reviews, enforced quality standards, and mentored team members on microservices architecture and integration patterns.

Impact:

- Improved system throughput by 20% through performance tuning of high-load services and optimization of core processing workflows.
- Cut memory footprint by 50% by redesigning service architecture and introducing efficient memory management strategies.
- Migrated the integration platform to Apache Kafka for asynchronous messaging, replacing legacy message queue systems.
- Maintained 99.9%+ uptime for critical banking operations under peak load conditions.

Scale: 87+ microservices, 4–6 developers, 3 environments (DEV/STAGE/PROD), processing millions of transactions daily across all banking channels.

COMMUNICATION GATEWAY — SMS & PUSH DELIVERY

Role: Senior Software Engineer / Tech Lead

2023 – Present

Company: Bereke Bank

Stack: Java, Spring Boot, Apache Kafka, SMPP, Firebase Cloud Messaging, Kubernetes

Overview:

A sub-platform within B-Connect responsible for delivering SMS and push notifications to bank customers. Handles routing, traffic management, provider failover, and delivery tracking for all customer-facing communications (OTP, transaction alerts, marketing campaigns).

Components:

- **Inbox-Hub** — central notification hub that receives messages from internal systems, normalizes them, and routes to the appropriate delivery channel (SMS or push).

- **Router-Service** — intelligent routing service that determines the optimal delivery channel based on message type, customer preferences, provider availability, and cost.
- **Integration-SMS-Traffic** — SMPP-based integration with SMS providers, handling traffic shaping, rate limiting, retry logic, and delivery receipts.

Key Contributions:

- Designed the multi-provider SMS failover architecture, ensuring delivery even when primary providers are unavailable.
- Implemented retry-marker logic for card processing system notification retries, with automatic marker detection and message routing (SMS vs push).
- Built traffic monitoring and alerting for SMS delivery rates, provider costs, and failure thresholds.

LOGIX — LOG COLLECTION & ROUTING PLATFORM

Role: Creator / Solo Developer

2026

Stack: Java 25, Quarkus, Apache Kafka, Elasticsearch, React, Docker, Kubernetes, Helm

GitHub: github.com/devbatyr/logix-app

Overview:

Logix is a high-performance, strongly-typed platform for log collection, validation, and routing. It prevents invalid or unstructured data from reaching Elasticsearch by enforcing JSON Schema contracts at the entry point. Designed as a lightweight, scalable alternative to heavy log-aggregation pipelines.

Architecture:

- **lgx-ingestor** — edge gateway that accepts logs via HTTP, validates against JSON Schema, routes valid logs to Kafka and invalid logs to a Dead Letter Queue (DLQ). Includes WAL fallback and backpressure handling.
- **lgx-sink** — background worker that consumes from Kafka, batches and bulk-indexes to Elasticsearch with retry and DLQ support.
- **lgx-dashboard** — React SPA control plane with health metrics, throughput charts, schema viewer, DLQ inspector with replay, and live log streaming.

Performance: Benchmarked at 17K+ logs/second with zero data loss, no duplicates, and sub-10s delivery latency. Ingestor memory footprint 150 MB, sink 267 MB.

Key Features:

- JSON Schema Draft 2020-12 validation at ingestion point.
- Kafka-based decoupling with retry and DLQ topics.
- Idempotent Elasticsearch indexing via content-based `_id`.
- Horizontal scaling: 1 instance handles 20K logs/s; 10+ instances for 500K+ logs/s.
- Full Helm chart for production Kubernetes deployment.

GITLAB REVIEWER — AUTOMATED CODE ANALYSIS

Role: Creator / Solo Developer

2025

Stack: Java 21, Quarkus, PMD 7.x, Ollama (DeepSeek), Docker, GitLab API v4

GitHub: github.com/devbatyr/gitlab-reviewer

Overview:

An automated static analysis tool for GitLab merge requests. Uses PMD to analyze Java code and posts AI-powered recommendations with contextual suggestions directly as MR comments. Designed as a centralized solution to avoid integrating linters into dozens of existing repositories.

Key Features:

- Receives GitLab webhooks and automatically analyzes MR diffs.
- PMD 7.x rule engine with configurable rulesets and suppressions.
- AI-powered recommendations via Ollama (DeepSeek models) — posts contextual improvement suggestions alongside violations.

- Docker support for both JVM and native-image modes.
- Single-point analysis: no need to modify each repository's CI pipeline.

Motivation: At Bereke Bank, the B-Connect platform has 87+ repositories. Integrating PMD into each repo's build pipeline would be impractical. GitLab Reviewer provides a single, centralized service that analyzes all MRs without touching individual project configurations.

BITBUCKET REVIEWER — AUTOMATED CODE ANALYSIS

Role: Creator / Solo Developer

2025

Stack: Java 21, Quarkus, PMD 7.x, Ollama (DeepSeek), Docker, Bitbucket Server API v1.0

GitHub: github.com/devbatyr/bitbucket-reviewer

Overview:

A counterpart to GitLab Reviewer, designed for Bitbucket Server. Uses PMD to analyze Java code in pull requests and posts AI-powered recommendations as PR comments. Same centralized analysis approach — no per-repository CI modifications needed.

Key Features:

- Bitbucket Server REST API v1.0 integration.
- PMD 7.x static analysis with configurable rulesets.
- AI-powered contextual suggestions via Ollama (DeepSeek models).
- Webhook-driven: automatically analyzes PRs on creation or update.
- Docker support for JVM and native-image modes.

Use Case: Deployed in environments using Bitbucket Server (Atlassian) as the primary code repository, providing the same automated code review capabilities as GitLab Reviewer but for a different VCS platform.