

Hyeonjun Kil

Backend Engineer / Platform Part Leader

kboxstar@gmail.com · Seoul, South Korea

[GitHub](#) · [LinkedIn](#) · [Blog](#) · [Resume Web](#) · [Portfolio](#)

Professional Summary

Backend engineer designing and operating shared backend platforms across 6 live game environments (12 branches), currently leading the Platform Part across real-time game chat, marketing data pipelines, and common package systems. At GameDuo, reduced **BigQuery** read cost per TB by **82%** via Storage Read API migration, expanded pipeline coverage from 60 to 360 days through serverless migration, validated capacity for a real-time chat server targeting **6,000 CCU**, and presented the case study at [Games on AWS 2025](#).

Over 4 years and 11 months across education, healthcare, clinical trials, and gaming, I have consistently delivered DB optimization, concurrency control, legacy modernization, and event-driven architecture. Recently I have integrated AI tools into an engineering quality system that verifies outputs through **TDD**, integration tests, review gates, and evidence search rather than unchecked code generation.

Core Competencies

DB Optimization, Concurrency & Distributed Systems

- Normalized and indexed a 100+ column marketing table, reducing query latency from 18s to 0.5s
- **Deadlock root cause analysis and distributed locking** — diagnosing and resolving data-layer bottlenecks
- Advanced **Redis Streams WAL** with a 4-shard **Valkey** cluster and serverless fanout instrumentation to preserve real-time chat ordering and horizontal scale (accept rate 99.39 → 99.99%, ack p95 767ms → 6.6ms)

Legacy Modernization & Reliability

- Express monolith to **NestJS** via 3-stage progressive migration, **TDD**-driven incident-free deployments during major releases
- Shared package system unifying 7 projects, cutting deployment from 3h to 15min

Serverless Architecture & Cost Optimization

- Batch-to-**EDA** migration expanding throughput 270x, **Lambda** hybrid pipeline design
- **BigQuery pricing model analysis with read-method switch** — **82%** cost reduction

Engineering Productivity & Quality Verification

- Established team quality gates that verify AI-assisted development through **TDD**, integration tests, review checklists, and Git hooks
- Automated pre-review domain-knowledge and rule checks through Git-based shared context plus local RAG/MCP search
- Presented an Amazon Q Developer-based data pipeline scaling case at [Games on AWS 2025](#)

Experience

DEV Team Platform Part Leader

2025.01 — Present

[GameDuo \[Detail\]](#)

Mobile game development and publishing company. Shared backend platform design for 6 live game environments (12 operating branches), data pipeline engineering, and common package system operations

DEV Team Platform Part Leader

2026.04 — Present

- **After the Platform Part was formed, prioritization and verification criteria for shared backend work were fragmented across games** — Standardized platform work from backlog to verification to completion with task-type criteria, review gates, and templates, **Established an initial operating system for tracking shared infrastructure, package, and chat-server work under the same criteria**
- **After AI-assisted development adoption, output quality and verification responsibility depended on each engineer's usage style** — Defined a five-dimension sizing framework with TDD/review-based verification principles across scope, complexity, uncertainty, collaboration, and validation, **Established the basis for evaluating deliverables by tests and reviews regardless of AI-tool usage**
- **Development-automation productivity varied significantly by team member, while individually discovered context and practices did not propagate, causing repeated trial-and-error** — Chose Git-based config-as-code over individual documentation — docs must be re-read by humans, but codified config integrates directly into the runtime and applies automatically. Built a shared team development-context platform unifying principles, coding standards, domain knowledge, specialist agents, and hooks in one git repo, **Enabled all team members to start with the same AI session context via git pull — converting shared rules, domain knowledge, and specialist agents into searchable team assets**

DEV Team Server Developer

2025.01 — 2026.03

- **Marketing data query cost surge (full-scan billing model)** — Evaluated partitioning but rejected due to Write IOPS increase; adopted gRPC distributed reads (Storage Read API), **Reduced BigQuery read cost per TB by 82% (\$6.25 → \$1.1)**
- **Batch processing collection limits (60 days)** — Migrated from batch processing to EDA-based on-demand pipeline — distributing peak load and enabling throughput scaling, **Expanded metrics range 6x (60 → 360 days), processing time 2h → 5min**
- **Code duplication and convention inconsistency across 7 projects** — Replaced manual code sync with shared NestJS package system (10 utility modules + Game Server Kit, auto-deploy), **Automated upgrades cutting deployment time 3hrs → 15min**
- **Marketing query latency (18s) from fragmented ad platform data (Google Ads/Meta/TikTok)** — Normalized read path + index tuning to avoid full-scan on 100-column table, **Performance improved 97% (18s → 0.5s)**
- **Regulatory risk from game probability disclosure requirements** — Integrated DynamicModule-based probability package across multi-game projects + CDK-based audit log pipeline, **Achieved 94%+ integration test coverage (12 suites, 115 tests)**

Projects: Real-Time Game Chat Server (Multi-Tenant), Marketing Integrated Platform, Internal Common Library System

JNPMedi [Detail]

Clinical trial data management solution startup with KRW 16B cumulative investment (MSA-based)

- **5+ monthly e-signature failures and clinical data-integrity risk caused by deadlock patterns** — Reproduced and diagnosed FK shared-lock deadlocks, then adjusted transaction boundaries to prevent recurrence, **Eliminated 5+ monthly e-signature processing failures**
- **VDR project schedule delay under MVP launch pressure** — 3-week emergency sprint (2 BE + 1 FE), Event-Driven PDF conversion + permission-based workflow, **MVP launched on schedule**
- **Delayed email failure detection (hours)** — Constructed SES+SNS event pipeline, **Bounce/complaint detection cut to seconds, Slack instant alerting (published tech blog)**

Projects: Maven Docs System Reliability Improvement, Maven VDR MVP Development

Dev Team Backend Engineer

2022.04 — 2023.06

Medgo [Detail]

Telemedicine and medication delivery platform startup with 300K cumulative downloads

- **Regression risk increased as auth, appointment, and notification changes accumulated around an Express.js single-file app.js structure** — 3-stage progressive migration (single-file → layered → NestJS) + JS → TS, **TDD 80% coverage, zero incidents during major releases**
- **Manual prescription entry bottleneck (3-5min per case)** — Naver Clova OCR + MFDS (Korea FDA) API integration for full workflow automation, **Pharmacist processing time cut 90% (3-5min → 30sec)**
- **Status reflection delay in consultation/dispensing/delivery (tens of seconds)** — Socket.IO real-time sync + Web Push notifications, **Reduced to under 1 second, eliminated patient wait complaints**

Projects: Doctor/Pharmacist Web Service Enhancement, Backend Architecture Migration

Dev Team Backend Engineer

2021.07 — 2022.03

SimpleHAN [Detail]

Custom software consulting firm

- **Manual learning management (attendance, completion, progress tracking)** — Built LMS full-stack (QR attendance, PDF certificates, Excel bulk registration), **Digitized operations for 300+ students per semester**
- **New matching and counseling features had to be added to a legacy PHP system, while direct changes carried high regression risk** — Integrated an independent Spring Boot app through a constrained iframe/postMessage boundary and implemented a weighted-average matching algorithm, **Launched the JOB Agent service while isolating the legacy-system impact surface**

Projects: Learning Management System (LMS) Development

Platform / Tooling Engineering

2026.04 — Present

Personal Projects (Side) [Detail]

Self-directed platform/tooling engineering outside work — development workflow automation framework and side projects

Projects: Public AI CLI Runtime Launcher

Presentations & Writing

Boosting Developer Productivity with Amazon Q Developer [Review]

Conference Talk — 2025.10
Games on AWS 2025 Customer Session

Email Delivery Monitoring through AWS SES Event Logs

Internal Seminar → Tech Blog — 2024.03
Presented internally then published as tech blog

Technical Skills

Backend: Node.js, TypeScript, JavaScript, NestJS, Express.js, TypeORM, Java, Spring Boot, Jest, TDD, REST API, gRPC, Testcontainers, Integration Test, OAuth2

Database: MySQL, BigQuery, ElastiCache (Redis), Redis, Redis Streams, Valkey, ClickHouse

Cloud & Infra: AWS (Lambda, ECS/Fargate, S3, SQS/SNS, CDK, EventBridge, Kinesis, Athena), GCP (BigQuery, Pub/Sub), Docker, GitHub Actions, CI/CD, Terraform, ARM64 (Graviton)

Quality & AI Verification: TDD Governance, Integration Test, Code Review, Git Hooks, Evidence Search, AI Output Verification

Professional Development

System Design Interview Study [GitHub] — System Design Interview: An Insider's Guide Vol. 1 & 2

2025.08 ~ 2026.04

Open Source

oh-my-openagent [GitHub]

- Fix cached model context limits for Anthropic providers [#2664](#) — 2026-03

opencode [GitHub]

- Fix macOS sidecar process hang on desktop app [#15136](#) — 2026-02

CodexBar [GitHub]

- Support kiro-cli 1.24+ Q Developer format [#288](#) — 2026-02

Education

Inha Technical College

Bachelor's (Advanced Major Program, Concurrent Employment) • Computer Information Engineering
2022.02 — 2023.02

Inha Technical College

Associate (Concurrent Employment) • Computer Information
2017.02 — 2022.02

Certifications

SQLD (Structured Query Language Developer)
Korea Data Agency • 2025.12

Engineer Information Processing
Human Resources Development Service of Korea • 2023.06