

Byungho Choi

Someone who makes things work, and makes things

choeebh@gmail.com linkedin.com/in/byungho-choi-
4a43632a5 github.com/renechoi upcurvewave.tistory.com

INTRODUCE

Whatever I am making, something gets drawn first. Building a space it was **how people would move through it**. Writing a track it was **the emotional curve over time**. Building a system it is **what a person is left holding when it breaks**.

I studied English literature, built and ran a cafe on Jeju island, produced music, worked in publishing and commerce, and became an engineer past thirty. Five changes taught one thing: engineering is the work of **closing the distance between what people actually want and a system that actually runs**.

So the domain does not decide it. Networking, build systems, messaging, static analysis, front end: patches merged into **fourteen unrelated codebases**. These days I build systems where being wrong means money is wrong, and I spend my time **moving my own judgment into systems**, so that people are left with the calls and the exceptions.

I believe **the world belongs to the people who make things**.

EXPERIENCE

2024-08 ~

Nexon Korea

• current

Platform Core, Payments · maintainer of the payment verification system

- Own the design, development and operation of the **payment verification system** tied directly to company-wide revenue.
- Operate **5M+ monthly transactions** across 10+ markets (Apple, Steam and others) at **99.99%+ SLA**.
- Eliminated duplicate-grant incidents through event-driven rework, removed recurring CS with subscription automation.
- Led an **AI-native operations rebuild** that changed how the team works.

Java

Kotlin

Spring

Kafka

MySQL

Redis

Docker

2023-07 ~ 2024-08

ParkingCloud

Core team · owner of season-ticket and product services

- Maintained and owned **season tickets, free-vehicle and product services** end to end.
- Kept **1,000+ tests at 80%+ coverage** running in CI for the services I owned.
- Wrote **30+ internal articles** on refactoring, incidents and retrospectives; gave internal talks.
- Employee of the year 2024; company slogan contest, 2nd place.

Java

Spring Batch

JPA

Querydsl

JUnit 5

PROJECT

Problems I owned at work, written as situation, choice, outcome.

2025-04 ~ 2025-07

Event publishing reliability and transaction management

Nexon Korea · redesigned the whole payment-to-grant path · 2 engineers, 70% contribution

- **Situation.** Grant events could be lost or duplicated after successful payment, and progress was visible only in logs.
- **Choice.** DynamoDB and Kafka cannot share one transaction, so I introduced a state table that records **how far each grant has gone**, publishing only after a conditional write succeeds. Chose at-most-once publishing over an outbox to avoid double write cost, pushing final idempotency to the consumer contract.
- **Outcome.** **~6,000 TPS** stable under load (about **12x** the design target), duplicate grants **50/month → 0**, one-click re-grant for operators.

Kafka

DynamoDB

Spring

nGrinder

Related post: how far can you trust Kafka message handling (Korean) ↗

2026-01 ~ 2026-08

Discount coupon system backend

Nexon Korea · a coupon platform shared by three departments · led DB and domain design

- **Situation.** The platform had no concurrency control, and a legacy coupon event had already taken the DB down once.
- **Choice.** Split the API into **validate** → **use** → **rollback** with use ahead of the PG call. Correctness rests on conditional UPDATES and idempotency keys; the distributed lock is only a traffic damper. Folded currency and locale into JSON so **bulk registration of 1,000** works.
- **Outcome.** Three departments consume **one API**; recall, reissue and expiry covered.

MySQL

Spring

Redis

idempotency

concurrency

Related post: Couchbase CAS concurrency troubleshooting (Korean) ↗

2026-01 ~ 2026-04

AI-native operations

Nexon Korea · led the change in how the team works

- **Situation.** Repetitive operations kept eating human hours.
- **Choice.** Decomposed the work into **5 systems and 30+ agents**, leaving people only judgment and exceptions.
- **Outcome.** The team's way of working changed, and I wrote the journey up for the internal AI community.

LLM

agent design

ops automation

2025-02 ~ 2025-04

Subscription billing automation

Nexon Korea · removing recurring support tickets structurally

- Situation. Subscription activation was manual, so the same class of support ticket came back every month.
- Choice. Classified every exception case, moved the common path into an **automatic activation flow**, and left people only the residual exceptions.
- Outcome. Related support tickets went from **5-7 a month to 0**.

Spring

Batch

payments domain

2024-12 ~ 2025-03

Multi-store support and clean architecture refactor

Nexon Korea · ending the pattern of editing core logic per store

- Situation. Every new store meant touching core logic.
- Choice. Redrew the boundaries with **ports and adapters**, pushing per-store differences out into adapters.
- Outcome. Onboarding a new store is now **a new use case and nothing else**.

clean architecture

Kotlin

Spring

2025-04 ~ 2025-05

AWS Linux EOS migration and CI/CD rebuild

Nexon Korea · swapping the runtime with no downtime

- Situation. The server OS was approaching end of support and the deploy pipeline had aged with it.
- Choice. Replaced runtime and pipeline together on **Docker and ECR**.
- Outcome. Build and deploy time **cut in half, zero downtime** through the cutover.

Docker

ECR

CI/CD

2023-12 ~ 2024-03

New ticket service from zero

ParkingCloud · season tickets and free vehicles, greenfield

- Situation. The new platform had no ticket domain at all; legacy data ran to **100M rows**.
- Choice. Designed read paths for volume first, built **100+ APIs wrapped in 800+ tests**.
- Outcome. Search latency improved **7x**; stable at 80%+ coverage. The story is on the blog.

Java

JPA

Querydsl

MySQL

Related post: cutting 500ms and getting a 7x improvement (Korean) ↗

2024-03 ~ 2024-06

Legacy integration for the iParking app

ParkingCloud · message loss and concurrency arriving together

- Situation. Messages were lost between the legacy system and the new platform, and concurrent updates collided.
- Choice. Made completion explicit with **manual Kafka acks**, and scoped a **Redis distributed lock** narrowly to the contended section.
- Outcome. Loss and duplicate updates stopped; the concurrency root-cause work from the same period is written up on the blog.

Kafka

Redis

concurrency

Related post: Couchbase CAS concurrency troubleshooting (Korean) ↗

Other work

Smaller than the cases above, and the reason the platform runs every day. One line each, not inflated.

- **Apple purchase-date mismatch.** Added a receipt-issue-date field with alerting, and split out a consumer-only deploy lane.
- **Load testing for a major title launch.** Verified launch traffic ahead of time with a multi-threaded load profile.
- **DLQ handling.** Redrew the commit and rollback boundary to make failed-message recovery reliable.
- **Back-office access control.** Per-environment IP allowlists and a blocked-access page.
- **Three security-audit remediations.** Access control, stored XSS, authorization checks, against OWASP Top 10 guidance.
- **Partner payment methods.** New top-up types and marketplace determination logic.
- **Authentication and authorization system (ParkingCloud).** JWT based, 90% class coverage.
- **OCPP firmware verification automation (ParkingCloud).** 30 payload types verified automatically, 30%+ QA efficiency gain.

ACTIVITY · EDUCATION

2024-09 ~

Algorithms and system design study group lead

- **60+ offline sessions;** ran two cohorts of a system design deep-dive. Weekly talks, code review, retrospectives.

AWARDS · CERTS

- Korea MSIT Minister's Award (2025, Open Source Contribution Academy, Apache Zeppelin team) · Employee of the year 2024 · company slogan contest, 2nd place
- Engineer Information Processing (2023) · SQLD (2022)

2023-03 ~ 2025-08

Korea National Open University, Computer Science, B.S.

- GPA 4.35/4.5, completed while working full time.

2009-03 ~ 2014-07

Ajou University, English Literature, B.A.

- GPA 4.39/4.5 · exchange semester at Utrecht University.

CONTACT
