

KEY PROJECTS

PORTFOLIO

Java/Spring 기반 백엔드 개발 및 시스템 연계 전문가
전략 수립부터 실행까지 주도하는 개발자 김진용 입니다.

CONTACT

010-6302-2985

yong9976@naver.com

01

INTRODUCTION

소개

02

CAREER SUMMARY

경력 요약

03

KEY PROJECTS

주요 프로젝트

04

ADDITIONAL SKILLS

추가 역량

05

EDUCATION

학력

INTRODUCTION

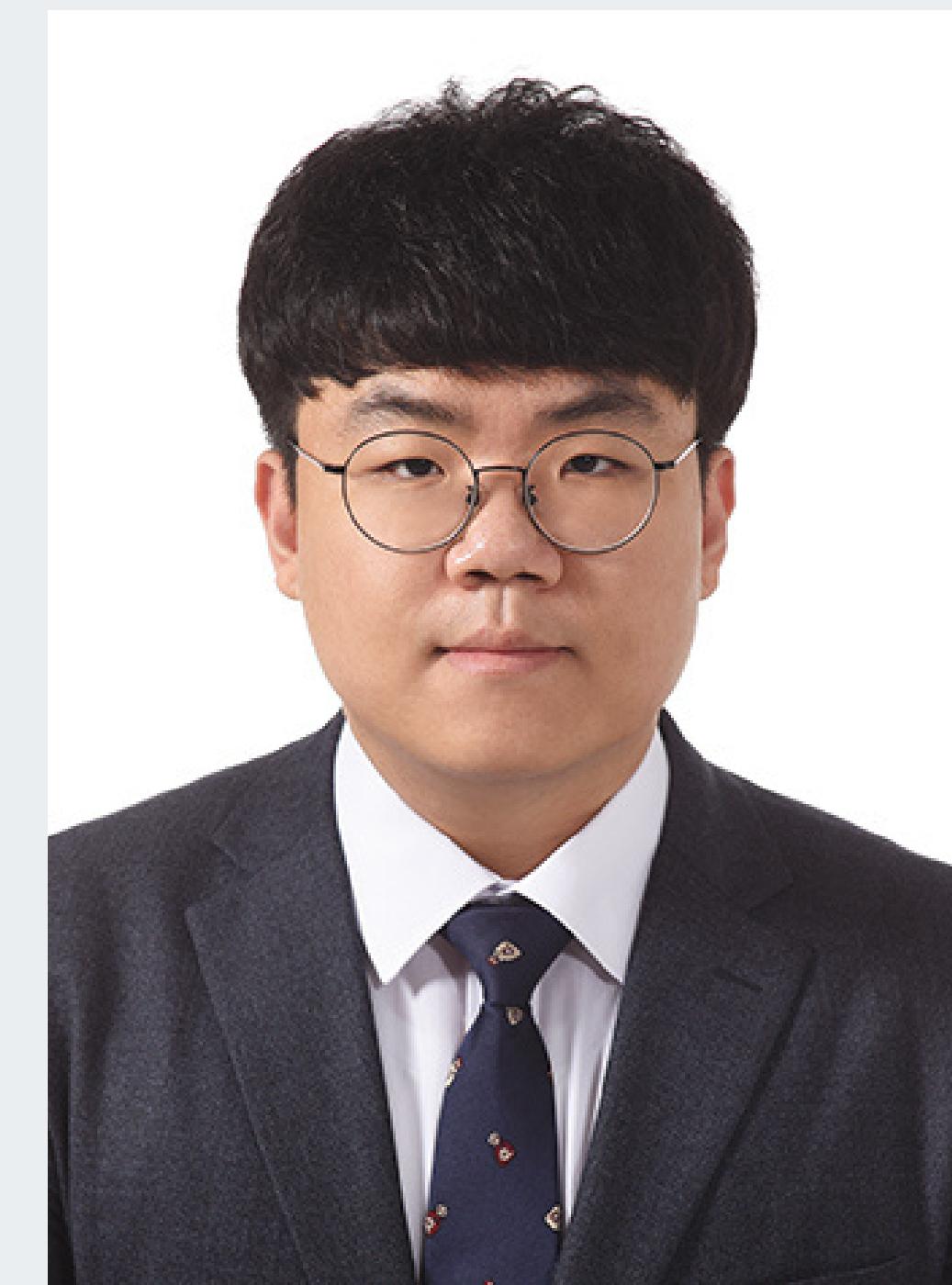
안녕하세요, 김진용입니다.

Java/Spring 기반의 백엔드 개발자로 데이터베이스 접근제어 솔루션 개발 및 금융권 프로젝트를 수행해왔습니다.

대규모 시스템 연계, 성능 최적화, 클라우드 자원 관리 등의 분야에서 다양한 경험을 쌓았으며, 최근에는 LLM 기반 MCP(Model Context Protocol) 서버 개발을 통해 AI 서비스 확장 영역까지 경험을 확장하고 있습니다.

특히 최근에는 경쟁사의 AI 제품 출시에 대응하여 자사 제품의 AI 전환 전략을 직접 수립하고 경영진 승인을 받아 실행에 옮긴 경험이 있습니다. 시장 분석부터 기술 검증, 단계별 로드맵 수립, 그리고 직접 개발까지 전 과정을 주도했습니다.

저는 단순히 주어진 업무를 수행하는 것을 넘어, 문제를 발견하고 해결 방안을 제시하며 실행까지 책임지는 개발자입니다. 2,000대 이상의 서버를 대상으로 한 자동 접속 시스템 개발, 멀티 클라우드 통합 관리, AI 플랫폼 연동 등의 경험을 통해 복잡한 시스템 문제를 해결하는 능력을 키워왔습니다.



CAREER SUMMARY

전략 기획부터 기술 검증, 실행까지 전 과정을 주도하며 비즈니스 가치를 창출하는 개발자입니다.
특히 AI, 클라우드, 고성능 시스템 등 최신 기술 트렌드를 빠르게 습득하고 실무에 적용하는 것을 강점으로 합니다.

피앤피시큐어

- 2021년 5월 - 현재
- 백엔드 개발 전임연구원

주요 업무

- AI 도입 전략 수립 및 실행
- LLM/MCP 기반 서비스 연동
- DB/AM 접근제어 솔루션 개발
- 클라우드 자원 통합 관리 시스템
- 금융권 대규모 시스템 구축
- 백엔드 아키텍처 고도화

주요 성과

- 경영진 대상 AI 전략 제안/승인
- 금융권 주요 업무 서버 자동접속 시스템 구축
 - 2,000대 서버 대상 적용 후, 추가 도입 성과 달성
- 접근제어 시스템 API 서버 테스트 자동화
 - 100% 자동화 달성 (1,331/1,331)

휴머스온

- 2020년 1월 - 2021년 5월
- 웹 서비스 개발 연구원

주요 업무

- 메시지 선발송 시스템 개발
- 실시간 메시지 처리 최적화
- RDBMS 컨버팅 작업
- Message Queue 아키텍처

주요 성과

- 메시지 발송 성능 대폭 향상
 - 기존 발송 (평균 3초) → 선발송 (평균 1초 내)

접근제어 시스템 AI 전략 수립

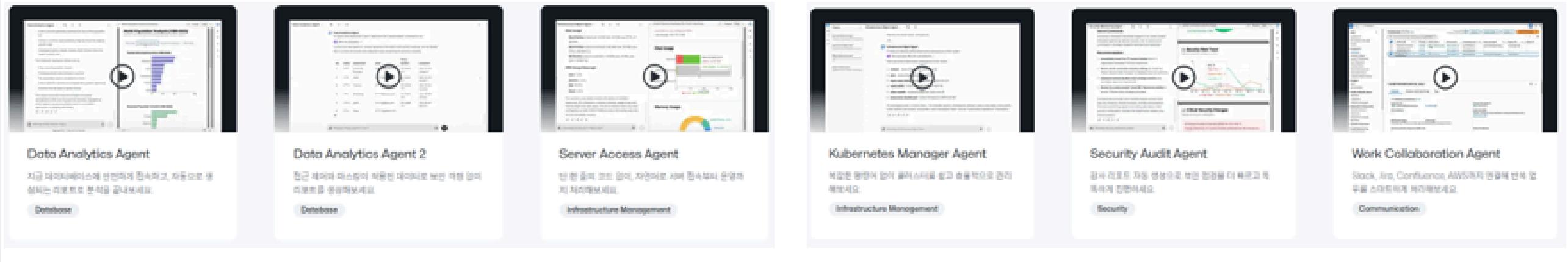
문제 > 접근 > 성과 > 배운 점

>>> 경쟁 환경 분석 및 4단계 AI 전환 로드맵 제시

■ 경쟁사 비교 (QueryPie AI Hub)

- QueryPie AI Hub는 MCP 서버를 조합하여 다양한 Agent를 만들고, 구동할 수 있는 AI Playground
- 외부 플랫폼이 아닌 사용자가 질의할 수 있는 자체 플랫폼 제공 ([LibreChat](#) 기반)
- 외부 MCP(Gmail, Slack)과 연동 가능 제공
- QueryPie 제품과 관련된 연동 기능 제공 (접근 가능한 DB 목록 조회, Query 실행, 권한 제어, 결과 마스킹 등)
- **단순 검색, 메뉴 구조뿐만 아니라, 접근 제어와 관련된 주요 기능들을 대상으로 범위가 확장되고 있음.**

* QueryPie에서 제안하는 Use Case 기반 Agent 목록



문제

- 2025년 초, 경쟁사 QueryPie가 MCP 기반 AI Hub을 출시하며 시장을 선점하려는 상황이었습니다. 자사 제품 DBSAFER는 AI 기능이 전무하여 경쟁력 약화가 우려되었고, 고객사들도 AI 기반 서비스를 요구하기 시작했습니다.
- 회사 내부에는 **AI 도입에 대한 명확한 전략이 없었고, “AI를 도입해야 한다”는 막연한 필요성만 있는 상태였습니다.**

접근제어 시스템 AI 전략 수립

문제 > 접근 > 성과 > 배운 점

>>> 경쟁 환경 분석 및 4단계 AI 전환 로드맵 제시

수행 역할 및 접근

- 문제를 해결하기 위해 AI 도입 전략을 수립하여 경영진(대표이사)에게 AI 전환 로드맵을 제안했습니다.

[1단계: 사용자 입력 기반 질의/탐색 구조]

[2단계: 외부 MCP 호스트 연동]

[3단계: 외부 AI API 연동]

[4단계: 자체 AI 모델 + 완전 통합]

기술 검증 및 분석

- 단순히 로드맵만 제시한 것이 아니라, 각 단계별로 기술적 타당성을 검증했습니다.
- MCP Protocol 표준 분석 및 활용 방안 검토
 - sLLM(Llama, Mistral 등) 도입 가능성 검토
 - 경쟁사 QueryPie AI Hub 기능 분석
 - 보안 요구사항 정의 (접근제어, 감사로그, 마스킹)



접근제어 시스템 AI 전략 수립

>>> DBSAFER AI 도입 전략 수립 및 실행

문제 > 접근 > 성과 > 배운 점

성과 및 임팩트

- ✓ 경영진 승인 및 프로젝트 진행
- ✓ 2단계 DBSAFER MCP 서버 개발 완료 (3개월)
- ✓ 향후 3~4단계 실행을 위한 기반 마련
- ✓ 회사의 AI 전환 방향성 수립

배운 점

- 첫째, **개발자도 비즈니스 관점이 필요**하다는 것을 깨달았습니다. 기술만 아는 것이 아니라 시장 상황, 경쟁사, 고객 요구를 분석하고 전략적으로 접근해야 한다는 것을 배웠습니다.
- 둘째, **경영진 설득을 위한 커뮤니케이션 능력의 중요성**을 느꼈습니다. 기술적 세부사항보다는 비즈니스 가치(매출, 경쟁력)와 실행 가능성을 중심으로 설명해야 효과적이라는 것을 배웠습니다.
- 셋째, **점진적 접근(Incremental Approach)의 가치를 이해**했습니다. 처음부터 완벽한 AI를 구축하려 하지 않고, 단계별로 가치를 제공하면서 발전시키는 전략이 현실적이고 리스크가 낮다는 것을 깨달았습니다.

KB 국민은행 클라우드 접근제어 프로젝트

문제 > 접근 > 성과 > 배운 점

>>> 멀티 클라우드 환경의 통합 관리 및 자산별 독립성 확보

문제

KB 국민은행에서는 AWS 등 다양한 클라우드 플랫폼을 사용하고 있었습니다. 각 플랫폼의 자원들이 분산되어 있어 통합 관리가 어려웠고, 특히 **자산별로 독립적인 접근제어를 제공하는 것이 불가능했습니다.**

기존에는 모든 클라우드 자원을 **하나의 접근제어 시스템으로 관리**했기 때문에, **특정 자산에 문제가 발생하면 전체 시스템에 영향을 주는 구조적 문제**가 있었습니다.

접근

- 클라우드 자원을 스캔하고 통합 관리하되, 자산별로 분산하여 독립적인 접근제어 서비스를 제공하는 **하이브리드 아키텍처**를 설계했습니다.

[1단계: 자산별 클라우드 자원 스캔]

- AWS SDK, Azure SDK, GCP Client Library 연동
- 클라우드 서비스 및 태그 정보 자동 수집
- Spring Scheduler 기반 주기적 스캔

[2단계: 통합 관리 서버 개발]

- 수집된 자원을 통합 조회하는 REST API
- Hazelcast 캐싱으로 대용량 메타데이터 조회 최적화 (**캐시 운영 부담 최소화와 확장성 고려**)
- 운영자용 통합 관리 인터페이스 제공

[3단계: 자산별 분산 배포]

- 자산 식별값 기준으로 접근제어 서비스 분산
- 독립된 컨테이너/VM으로 격리 배포
- 한 자산의 장애가 다른 자산에 영향 없음

KB 국민은행 클라우드 접근제어 프로젝트

문제 > 접근 > 성과 > 배운 점

>>> 멀티 클라우드 환경의 통합 관리 및 자산별 독립성 확보

성과

- ✓ 멀티 클라우드 플랫폼의 자원을 통합 조회 가능
- ✓ 자산별로 독립된 접근제어 서비스 제공
- ✓ 한 자산의 문제가 다른 자산에 영향 주지 않음
- ✓ 운영자의 관리 편의성 대폭 향상
- ✓ 서비스 확장성 및 유지보수성 향상

배운 점

- 첫째, **멀티 클라우드 환경의 복잡성을 이해**하게 되었습니다. 각 플랫폼마다 API 구조, 인증 방식, 데이터 형식이 다르기 때문에 이를 추상화하는 공통 인터페이스 설계가 핵심이었습니다.
- 둘째, **통합과 분산의 균형**을 배웠습니다. **중앙에서 통합 관리는 하되 실제 서비스는 분산하여 독립성을 보장하는 하이브리드 아키텍처**의 장점을 깨달았습니다.
- 셋째, 캐싱 전략의 중요성을 실감했습니다. **클라우드 자원 메타데이터가 방대하고, 빈번한 변화**로 인해 Hazelcast 캐싱 없이는 조회 성능을 확보할 수 없었습니다.

KDB 산업은행 접근제어 시스템 고도화 프로젝트

문제 > 접근 > 성과 > 배운 점

>>> 2,000대 이상 서버 대상 고성능 자동 접속 시스템

문제

KDB 산업은행에서는 약 2,000대의 주요 업무 서버에 대한 자동 접속 기능이 필요했습니다. 특히 **업무 시작 시간(오전 9시)에 트래픽이 집중되어 동시에 수백 건의 접속 요청이 발생하는 상황**이었습니다.

기존 동기 방식(Thread per Connection)으로는 이러한 대량 트래픽을 처리할 수 없었고, 타임아웃이나 접속 지연이 빈번하게 발생했습니다.

접근

- **트래픽 집중 문제를 해결하기 위해 비동기 이벤트 기반 아키텍처를 선택**했습니다. Netty 프레임워크를 활용하여 논블로킹 I/O 기반의 고성능 자동 접속 서버를 개발했습니다.

[아키텍처 설계]

- Netty 이벤트 루프: 적은 스레드로 대량 연결 처리
- 논블로킹 I/O: 네트워크 대기 시간 활용

[외부 시스템 연동]

- 계정권한 관리 시스템: 사용자 권한 실시간 조회
- 패스워드 관리 시스템: 암호화된 비밀번호 취득

기술 스택

- Netty (비동기 네트워크 프레임워크)
- Spring Boot (비즈니스 로직)

KDB 산업은행 접근제어 시스템 고도화 프로젝트

문제 > 접근 > 성과 > 배운 점

>>> 2,000대 이상 서버 대상 고성능 자동 접속 시스템

성과

- ✓ 약 2,000대 이상의 서버 동시 처리 가능
- ✓ 평균 접속 처리 시간 75% 단축 (3.5초 → 0.8초)
- ✓ 트래픽 집중 시간대에도 안정적인 성능 보장
- ✓ 타임아웃 및 접속 실패율 최소화
- ✓ 현재까지 특이사항 없이 운영 중
- ✓ 인터넷망으로도 확대 도입 성과

배운 점

- 첫째, 비동기 프로그래밍의 강력함을 체감했습니다. Netty의 이벤트 루프를 활용하여 **적은 스레드(10개)로도 대량의 동시 접속(2000개)을 처리**할 수 있었습니다.
- 둘째, 외부 시스템 연동 시 안정성의 중요성을 배웠습니다. 계정 정보를 가져오는 **외부 API에 장애가 발생할 경우를 대비한 재시도 로직, 서킷 브레이커 패턴 등을 적용**했습니다.
- 셋째, **좋은 아키텍처는 확장성을 자연스럽게 제공**한다는 것을 배웠습니다. 초기 2,000대를 목표로 설계했지만, 아키텍처가 탄탄해서 추가 도입하는데, 큰 수정 없이 확대 도입할 수 있었습니다.

TECH STACK

Language & Framework

- Backend
 - Java
 - Spring Boot, Spring Security
 - Spring WebFlux (Reactive)
 - MyBatis
- Frontend
 - TypeScript, Node.js

- Database
 - MySQL
 - Oracle
 - Redis
 - Hazelcast

Communication & Integration

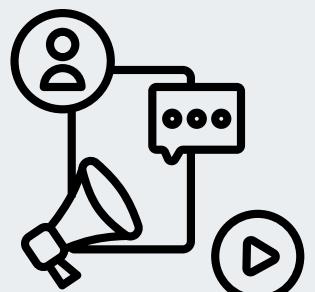
- gRPC
- Netty
- REST API
- Message Queue
- JSON-RPC 2.0

AI & Modern Tech

- MCP(Model Context Protocol)
- Claude, OpenAI API

Cloud & DevOps

- AWS (EC2, RDS, DynamoDB, Memory DB)
- Azure (VM, Storage, SQL Database)
- GCP (Compute Engine, Cloud Storage)
- Git, GitHub



ADDITIONAL PROJECTS

백엔드 아키텍처 고도화 프로젝트

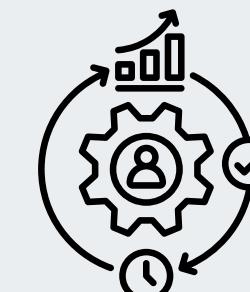
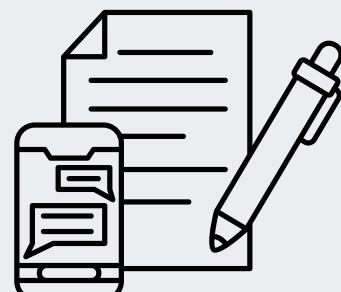
제품 버전별로 분리 관리하던 백엔드 형상을 통합 지원 아키텍처로 전환. 서버 기동 시 버전 기준으로 컴포넌트와 기능이 동적으로 세팅되는 구조 개발. Spring Boot Profile 및 동적 Bean 등록 활용. 개발 생산성 향상 및 유지보수 복잡도 감소.

카카오 DB 접근제어 시스템 원백 프로젝트

기존 타사 제품에서 자사 제품(DBSAFER)으로 전환. 완벽한 호환성 유지하면서 성능 개선. Refresh Token Rotation 및 탈취 감지 기능 구현으로 인증 강화. 카카오톡 API 연동한 자원 알림 기능 개발 (사전/당일 알림). 사용 편의성 및 운영 효율성 향상.

기타 주요 경험

- Spring Batch를 활용한 대용량 데이터 처리
- 외부 시스템 연동 (Active Directory, Okta)
- 보안 인증 강화 (JWT, OAuth 2.0)
- 테스트 자동화 (단위/통합 테스트)



EDUCATION

한세대학교

- 2014년 3월 - 2020년 2월
- 정보통신공학과 학사

한영고등학교

- 2010년 3월 - 2013년 2월
- 인문계

자격증

- 정보처리기사

병역

- 2014년 8월 - 2016년 5월
- 육군 병장 만기제대

김진용

THANK YOU

CONTACT

010-6302-2985

yong9976@naver.com