

# 조민서 | 현실을 데이터로 전환하는 엔지니어

Phone: +82 10-5023-9977

Email: lging12@naver.com

Blog: <https://virtualworld.tistory.com/>

GitHub: <https://github.com/minseojo>

포트폴리오 다운로드 링크 : [GitHub](#)

---

## 1. 소개

단순히 작동하는 코드를 넘어서, 명확한 목적 아래 트레이드오프를 고려하며 시스템을 만드는 일에 흥미를 느낍니다. 기능 구현 자체보다 '왜 이 구조를 택했는가', '어떤 병목이 발생할 수 있는가', '어떻게 확장성과 유지보수성을 확보할 것인가', '이 코드 한 줄이 어떤 의미를 가지는가'와 같은 질문을 스스로에게 던지며 개발합니다.

성능 최적화, 병렬·분산 처리, 효율·안정적인 데이터 파이프라인 구축에 관심이 많으며, 시스템 전반을 깊이 있게 이해하고 설계하는 역량을 키우기 위해 실무와 학습을 병행하고 있습니다. 복잡한 문제를 단순하게 구조화하고, 명확하게 설명할 수 있는 시스템을 지향하며, 기술을 목적이 아닌 도구로 삼아 문제의 본질에 집중하는 문제 해결형 엔지니어를 지향합니다.

---

## 2. 기술 스택

- **Languages:** Java, C#, Python, JavaScript, TypeScript
  - **Backend:** Spring Boot, Spring MVC, JPA/Hibernate
  - **Database / Storage:** MySQL, PostgreSQL, SQLite, Amazon S3, SeaweedFS
  - **Messaging:** Kafka, Amazon SQS
  - **Cloud Infrastructure:** AWS (EC2, RDS, S3, CloudFront, Lambda, Route 53)
  - **DevOps:** Docker, GitHub Actions
  - **Monitoring / Testing:** Grafana, k6
  - **Frontend:** React, React Native
- 

## 3. 경력

아이씨티웨이(주) | 정보기술연구소 | 사원 (정규직)

2024.12 ~ 현재

- 공간 정보 자료구조 개발 (**Implicit Tiling**)
  - 모놀리식 GIS → 멀티노드 GIS 엔진 전환
  - 공간 정보 버전 관리 시스템 설계·구현 (= 공간 정보 Git-LFS)
  - 오프라인 기반 3D GIS 엔진을 온라인 서비스로 전환 (on-the-fly)
  - 파일 포맷 변환 코드 개발 (b3dm → glb)
- 

## 4. 경력 프로젝트

### 4.1. 대규모 공간 데이터 처리 자료구조 개발

2025.03 ~ 2025.06 | 인원: 2인 | 기여도: 90% | C#, Cesium 3D Tiles 1.1

- 타임 탐색 구조 최적화:  $O(\log_4 N) \rightarrow O(1)$ , FPS 20~30% 향상
- 데이터 경량화: JSON 크기 30~40% 감소, 렌더링/네트워크 비용 절감
- Random Access 지원: 좌표 기반 **URI 직접 접근**
- 부분 빌드·장애 복구: 기존에는 빌드 중 장애 시 수십~수백 시간 소요되는 전체 Rebuild 필요  
→ 좌표 기반 URI를 통해 빌드에 실패한 파일을 선별하여 **실패한 파일만 Rebuild**
- 확장성: 국가·도시 단위 부분 갱신 및 전지구 일관성 유지 구조 확보

블로그 포스팅: <https://virtualworld.tistory.com/23> - 전 세계 공간 정보를 관리하는 자료구조 설계/구현기

Implicit Tiling 시작화 데모: <https://minseojo.github.io/implicit-tiling/>

## 4.2. 공간 정보 버전관리 시스템 설계/개발 (모놀리식 → 멀티노드 전환)

2025.06 ~ 진행 중 | 인원: 2인 | 기여도: 50% | Java, C#, PostgreSQL, SeaweedFS, Kafka

- **아키텍처 전환**
  - 기존 모놀리식 기반 전체 재빌드 구조를 **Kafka 기반 Saga 패턴 MSA**로 재설계
  - 빌드 병렬화, 장애 격리, 선택적 재처리 구조 확보
- **모듈화**
  - 타일 빌드 단계(소스 수집 → 3D 모델 생성 → 타일 배치)를 **SRP 기반 모듈 단위로 분리**
  - Choreography 방식 **비동기 이벤트 전파 파이프라인 구축**
- **병렬 처리 구조 구현**
  - 타일 빌드 메시지 키를 Auto-Increment 정수형 PK로 설계하고 **PK 범위 기반 Index Range Scan 적용**
  - 각 빌드 범위를 Kafka 메시지로 발행해 **라운드Robin 분산 처리만으로 서버 증설 시 처리량 선형 확장** 달성
- **트랜잭션 일관성 확보**
  - PostgreSQL Outbox Table + Polling 구조로 1 Saga 단위 트랜잭션 보장
- **증복 처리 최소화**
  - SHA-1 해시 기반 변경 감지로 **변경된 타일만 선별 빌드**, 불필요한 재처리 제거

## 4.3. 파일 포맷 변환 파이프라인 개발 (B3DM → glTF 2.0)

2024.05 ~ 2024.06 | 인원: 2명 | 기여도: 70% | C#, B3DM, glTF 2.0

- **변환 로직 자체 구현:** C# 기반 B3DM 파서 직접 개발 Batch Table 파싱 및 glTF 2.0 직렬화 구조 설계
- **좌표계 변환 및 메타데이터 통합:** Y-up → Z-up 좌표계 변환, 텍스처·메타데이터를 손상 없이 보존하는 glTF 2.0 통합 구조 설계
- **성능 최적화:** 기존 Node.js 기반 3d-tiles-tools 대비 변환 속도 **20분 → 5초(약 240배)** 단축
- **품질 보존:** Draco 압축 및 불필요한 리소스 제거 없이 원본 텍스처·비즈니스 메타데이터 유지
- **신뢰성 확보:** 단위 테스트 코드 작성으로 변환 안정성 검증, 유지보수성 향상

## 5. 주요 프로젝트 (팀/개인)

### 5.1. Text-to-Image 생성형 AI 기반 네컷 만화 생성 앱

2024.04 ~ 2024.12 | 인원: 3인 | 기여도: 백엔드 70% · 프론트 25% · AI 30%

Java, Spring Boot, AWS (EC2, RDS, S3, CloudFront, Lambda, Route 53), React Native, Python Stable Diffusion, LoRA

- **AI 이미지 생성 파이프라인 구축**
  - Lambda → SQS → GPU Spot 인스턴스로 구성된 비동기 아키텍처 설계
  - CPU-GPU 리소스 간섭 제거 및 AI 인프라 비용 **56% 절감 (약 44만원/월 → 19만원/월)**
- **조회 및 캐싱 최적화**
  - 반정규화된 1:N 구조로 단일 쿼리 조회 → 응답 속도 **46% 향상**
  - Presigned URL + CloudFront 캐싱으로 이미지 접근 트래픽 최소화
- **다국어·자동 번역 시스템**
  - 커스텀 애노테이션 기반 번역 자동화 → 신규 언어 추가 시 코드 수정 없이 확장 가능
- **실시간 상태 안내 및 UX 유지**
  - FCM을 활용한 생성 상태 알림
  - 캐시 무효화 및 플레이스홀더 처리로 지역 시에도 UX 안정성 확보
- **배포 자동화 및 무중단 운영**
  - AMI 기반 Auto Scaling + GitHub Actions Rolling 배포
  - 배포 시간 **70% 단축**, 무중단 서비스 운영

## 5.2. MySQL 기반 모듈러 샤딩 조회 성능 분석

2024.03 ~ 2024.06 | 인원: 4인 | 기여도: 35% (팀장) | Java, Spring Boot, MySQL, JMeter, Thymeleaf

YouTube: <https://www.youtube.com/watch?v=zAhyplsdYAA>

GitHub: <https://github.com/minseojo/modular-sharding>

- 약 1,000만 건 데이터를 대상으로 1→3샤드 분산 및 인덱스 전략을 비교하는 **Web/테스트 환경 구축**
- **쿼리 유형별 성능 분석 (샤딩 관점, 샤드 1개→3개 확장)**
  - 테이블 풀 스캔 쿼리: **최대 3배** 응답 시간 단축
  - 인덱스 레인지 쿼리: **최대 2배** 응답 시간 단축
  - 커버링 인덱스 쿼리: **최대 1.5배** 응답 시간 단축
- **쿼리 유형별 성능 분석 (인덱스 관점)**
  - 인덱스 레인지 스캔은 테이블 풀 스캔 대비 **최대 2.5배** 빠른 응답
  - 커버링 인덱스는 인덱스 레인지 스캔 대비 **최대 5.5배** 빠른 응답
  - 커버링 인덱스는 테이블 풀 스캔 대비 **최대 14배** 빠른 응답
- **데이터 업로드/다운로드 개선**
  - **Bulk Insert**(JDBC Batch) + **Batch Commit**으로 **데이터 적재 시간 63% 단축** (11시간 → 4시간)
  - **JVM Heap 메모리 튜닝**으로 대용량 배치 로딩 중 발생하는 **OOM 제거** → **제한된 자원으로 실험 환경 확보**

## 5.3. 병렬 채용 정보 스크래퍼 개발

2023.09 ~ 2023.11 | 인원: 1인 | 기여도: 100% | Java, Selenium, Multi-threading, Apache POI

YouTube: <https://www.youtube.com/watch?v=a0PJ3KzdYwk>

GitHub: <https://github.com/minseojo/job-information-scraper>

- **멀티스레딩 성능 최적화**
  - CPU 코어 기반 Thread Pool 및 페이지 균등 분할로 멀티스레드 스크래퍼 구현
  - 단일 스레드 대비 **최대 3배** 처리 속도 향상 (**80페이지, 1,600건 기준: 2분 → 40초**)
- **동시성 및 데이터 무결성**
  - 멀티스레드 쓰기 충돌(경쟁 조건) 해결을 위해 Split/Merge 패턴을 적용
  - 공유 자원 접근 없이 결과 **데이터 무결성 확보**
- **리소스 효율성 증대**
  - 스크래핑 웹 로드 대기시간 고정 Sleep 대신 조건 대기(WebDriverWait) 메커니즘 적용
  - 불필요한 **CPU 점유율과 I/O 대기 시간 최소화**
- **호환성 및 UX 개선**
  - 크로스 **브라우저 호환성 확보** 및 스레드별 스크래핑 진행 상황 **시각화 기능** 구현

## 6. 학력 · 대내외활동

- 전북대학교 | IT정보공학과 · 2019.03 ~ 2025.02 (졸업)
- SW마에스트로 15기 연수생 · 2024.04 ~ 2024.11
- DB 연구실 학부연구생 · 2023.09 ~ 2024.06
- 운영체제/네트워크 스터디 리드 · 2023.01 ~ 2023.04
- 교내 알고리즘 동아리 · 2020.04 ~ 2021.06

## 7. 자격증

- TOPCIT 수준4 (755점) · 2024.11 · 정보통신기획평가원
- 정보처리기사 · 2024.06 · 한국산업인력공단
- SQLD (SQL 개발자) · 2021.12 · 한국데이터산업진흥원

---

## 8. 수상

- **공군참모총장상** · 2024.12 · 정보통신기획평가원 · 제22회 TOPCIT 전국 18위 (**전국 상위 0.002%**)
- **총장상** · 2024.11 · 전북대학교 · 제22회 TOPCIT 교내 1위
- **동상** · 2023.11 · 전북대학교 · ACM ICPC 교내 3위
- **동상** · 2021.11 · 전북대학교 · ACM ICPC 교내 3위
- **동상** · 2021.06 · 전북대학교 · 기초설계 경진대회 1위