김윤기

Backend Developer

- 010-3385-2507
- kalsagansteve@gmail.com
- github.com/kalsteve
- velog.io/@kalsteve98

Summary

전 스택을 아우르며 실시간 스트리밍과 대규모 검색 시스템을 최적화한 경험이 있습니다.
데이터 흐름·검색·무중단 배포를 설계하여 사용자 경험과 운영 효율을 동시에 향상시켜본 경험이 있습니다.
I/O 병목·DB 부하·모바일 제약을 해결해 서비스 성능과 가용성을 높이는 문제 해결형 개발자입니다.

Project

Brain Washer (2024.06 - 2024.08) - github

실존하는 특정 인물이 독설로 동기 부여해주는 서비스

FastAPI, MySQL, AWS(CloudFront, EC2, RDS, CodeDeploy), GitHub Actions, Docker, Redis, Nginx

비동기 스트리밍 파이프라인 재설계 - (Blog)

- **문제 :** 텍스트 응답과 TTS 음성이 순차 전송되어 최대 **180초의 지연**이 발생하는 I/O 동기화 문제가 있었습니다.
- **시도 :** asyncio 기반 텍스트·음성 동시 **스트리밍 코루틴**을 구현하고, ProcessPoolExecutor에서 TTS 생성 작업을 병렬 처리한 뒤 asyncio.Queue로 패킷 순서를 보존하며 SSE으로 실시간 Push 하였습니다.
- **결과 :** 텍스트-음성 시작 지연을 **180초 → 12초로 93 % 단축**하고, 동시 처리량을 150 req/s → 900 req/s로 6배 확대하였습니다.

Redis 기반 캐싱

- **문제 :** PCM 스트림을 매 요청마다 MySQL로 저장해 네트워크·디스크가 포화되었고, **VPC RTT가 2→40 ms, DB P99 지연이 150 ms로 급증**했습니다.
- **시도 :** 스트림을 먼저 Redis에 적재해 **네트워크 왕복을 한 번**으로 줄이고, 사용자가 저장 버튼을 눌렀을 때에만 S3와 MySQL로 비동기 **Write-Behind** 동기화를 수행하도록 로직을 재구성했습니다.
- **결과 :** VPC RTT를 40 ms에서 **3 ms**로 회복했습니다. DB 쓰기 P99 지연도 150 ms에서 **30 ms**로 개선되었습니다.

CI/CD 고도화

- **문제 :** 모든 서비스가 단일 EC2 인스턴스에 배포되어 장애 시 **전체 서비스가 중단**되었습니다.
- **시도 :** GitHub Actions → CodeDeploy로 ALB 기반 Blue-Green 배포를 구성했습니다. FE·BE를 분리하고 자동 헬스·테스트로 **실패 시 즉시 롤백**됩니다.
- **결과 :** 배포 시간을 **10분 → 4분**으로 단축하고, 배포 실패율을 5 % → 0 %로 낮췄으며 릴리스 중 **가용성을 95 % 이상** 유지했습니다.

Nginx + Fail2Ban 기반 DDoS 방어

- **문제 :** 부트캠프 공개 시연 중 **10 k RPS 이상의 무작위 요청**으로 EC2 인스턴스가 다운되는 문제가 있었습니다.
- **시도 :** Nginx에 **Bot 탐지 로직**을 추가하고 Fail2Ban으로 3분 내 50회 이상 **요청하는 IP를 차단**, CloudFront WAF 규칙으로 L7 공격을 방어했습니다.
- **결과 :** 재현 테스트에서 15 k RPS 공격에도 **P95 응답 210 ms 이하**, CPU 60 % 미만으로 안정적으로 운영되면서 SLA를 충족했습니다.

IT -infra (2024.09 - 2025.06) - github

골목길 보행자와 자동차와의 사고를 비콘을 통해 방지하는 서비스

Android, Jectpack compose, FastAPI, MySQL

검색 서비스 제공

- **문제 :** 디바이스 이름이나 MAC 주소를 LIKE 검색으로 조회할 때 평균 **1 초 안팎의 지연**이 발생하였습니다.
- **시도 :** 경량 검색엔진 **Meilisearch**를 도입하고, 이름·MAC 주소 Prefix 매칭 인덱스를 구축해 정렬 기능을 추가하였습니다.
- **결과 :** 기기 조회 평균 응답 시간을 약 **40 ms로 단축**, DB 읽기 트래픽이 **절반 이하로 감소**하여 DB 부하를 개선하였습니다.

Foreground Service로 BLE 스캔 지속성 확보

- **문제 :** Android Doze·App Standby로 백그라운드 **BLE 스캔이 불가능**하였습니다.
- **시도 :** 시스템 제약을 우회하기 위해 **Foreground Service + Notification 구조**로 스캐너를 재구현했습니다.
- **결과 :** 스캔 성공률을 **62 % → 98 %**로 끌어올리고 배터리 소모를 4.2 %/h → 2.8 %/h로 절감했습니다.

Kalman + EMA 하이브리드 필터로 RSSI 정밀도 향상

- **문제 :** **RSSI ±10 dBm** 노이즈로 거리 추정 오차가 ±5 m까지 확대되었습니다.
- **시도 :** 1차 Kalman 필터로 동적 노이즈를 제거한 뒤 N=4 지수 이동 평균(EMA) 으로 **정적 노이즈를 평활화**하였습니다.
- **결과 :** 결과 위치 오차를 **±1.2 m → ±0.8 m(-35 %)로 감소**하였습니다.

Technical Skills

- C 언어 BSD 소켓 API와 pthread 기반 멀티스레딩을 이용해 수백 명이 동시에 접속하는 할리갈리 게임 서버를 구현한 경험이 있습니다. 구조체 기반 패킷 프로토콜, mutex·cond 동기화로 안정적인 에러 복구 루틴을 직접 설계·최적화하였습니다.
- CI/CD 영역에서는 GitHub Actions → AWS CodeDeploy → Docker Compose를 파이프라인으로 연동하고 ALB Blue/Green 전략을 적용하여 무중단 배포(Zero-Downtime) 를 달성하였습니다. 배포 전후 헬스 체크·Slack 알림·자동 롤백을 탑재해 평균 배포 시간을 10 분 → 4 분으로 단축하고 실패율을 0%로 유지한 경험이 있습니다.
- Node.js(Express) 마이크로서비스를 구축하면서 OpenAI Chat/Speech API의 단일 응답 한계를 해결하기 위해 WebSocket 파이프라인을 설계, JSON chunk 단위로 음성 스트림을 실시간 전송한 경험이 있습니다.
- 모바일 분야에서는 Android Jetpack Compose와 BLE Scanner API를 활용해 iBeacon(UUID·Major·Minor·RSSI) 기반 충돌 예방 앱을 개발했습니다. Foreground Service으로 Doze/App Standby 환경에서도 스캔 지속성을 확보하고, 권한 흐름·배터리 소비 지표를 직접 튜닝하여 스캔 성공률 62 % → 98 %, 소모 4.2 %/h → 2.8 %/h까지 최적화한 경험이 있습니다.

Activity

Techeer 8기 (2024.08 -)

2024 하계 테커 실리콘밸리 소프트웨어 부트캠프 (2023.06.24 - 2024.08.03)

Techeer partners (2024.03.23 - 2024.06.23)

- 실리콘밸리 엔지니어의 SW개발자 스터디 그룹
- 기술 세션, 프론트엔드 React 프레임워크등의 스터디 활동

2023 동계 테커 실리콘밸리 소프트웨어 부트캠프 (2023.12.26 - 2024.02.03)

Techeer partners (2023.09.12 - 2023.12.23)

- 실리콘밸리 엔지니어의 SW개발자 스터디 그룹
- 기술 세션, 백엔드 Spring Boot 프레임워크등의 스터디 활동

Education

- 한국공학대학교 (2023.3 - 2025.8)
  임베디드시스템전공 (3.53 - 4.5)