

# 박종혁

phone: +821031663505 email: [jkzombie7@gmail.com](mailto:jkzombie7@gmail.com) blog: <https://devtaco30.github.io/devtaco-blog/>

5년 경력의 Java/Kotlin 기반 백엔드 개발자로서, 유지보수성과 확장성을 고려한 구조 설계에 강점을 갖고 있습니다.

블루밍비트와 웨이브릿지에서 MSA 리빌딩, 고성능 데이터 파이프라인 구축, 그리고 안정적인 서비스 운영에 기여하며 서비스의 유지보수성과 확장성을 높이는 데 주력했습니다. 초기에는 구조적 일관성과 책임 분리를 중시하여 안정적인 시스템 구축에 집중했고, 이후 프론트엔드 및 기획자와의 긴밀한 협업을 통해 사용자 경험과 비즈니스 요구사항을 동시에 고려한 유연한 설계 방식으로 개발 기조를 발전시켜 왔습니다.

AWS 클라우드 환경에서의 서비스 운영 경험과 Jenkins를 활용한 CI/CD 파이프라인 구축을 통해 배포 효율성과 안정성을 높였으며, Redis와 메시지 큐를 활용한 이벤트 기반 아키텍처에도 깊은 관심과 실무 경험을 가지고 있습니다. 최근에는 AI 개발툴을 적극 활용하여 코드 품질과 개발 생산성을 높이고 있습니다.

단순 기능 개발을 넘어 공통 모듈화와 표준화 작업을 수행해 서비스 전반의 일관성을 높여왔으며, 기술적 해결이 단지 기능 구현에 그치지 않고, 궁극적으로는 서비스 운영과 사용자의 만족도에 긍정적인 영향을 줄 수 있도록 구조적인 개선을 추구해왔습니다.

현재는 기술적 깊이를 확장하면서도, 팀과 서비스 전체에 기여할 수 있는 비즈니스 중심의 엔지니어를 지향하고 있습니다.

## [기술 스택 요약]

- Backend: Java, Kotlin, Spring Boot
- Infra: Kafka, Kafka Streams, Redis, MySQL, PostgreSQL, MongoDB, ElasticSearch, RabbitMQ, AWS
- 협업 툴: Jira, Confluence, Slack, Notion

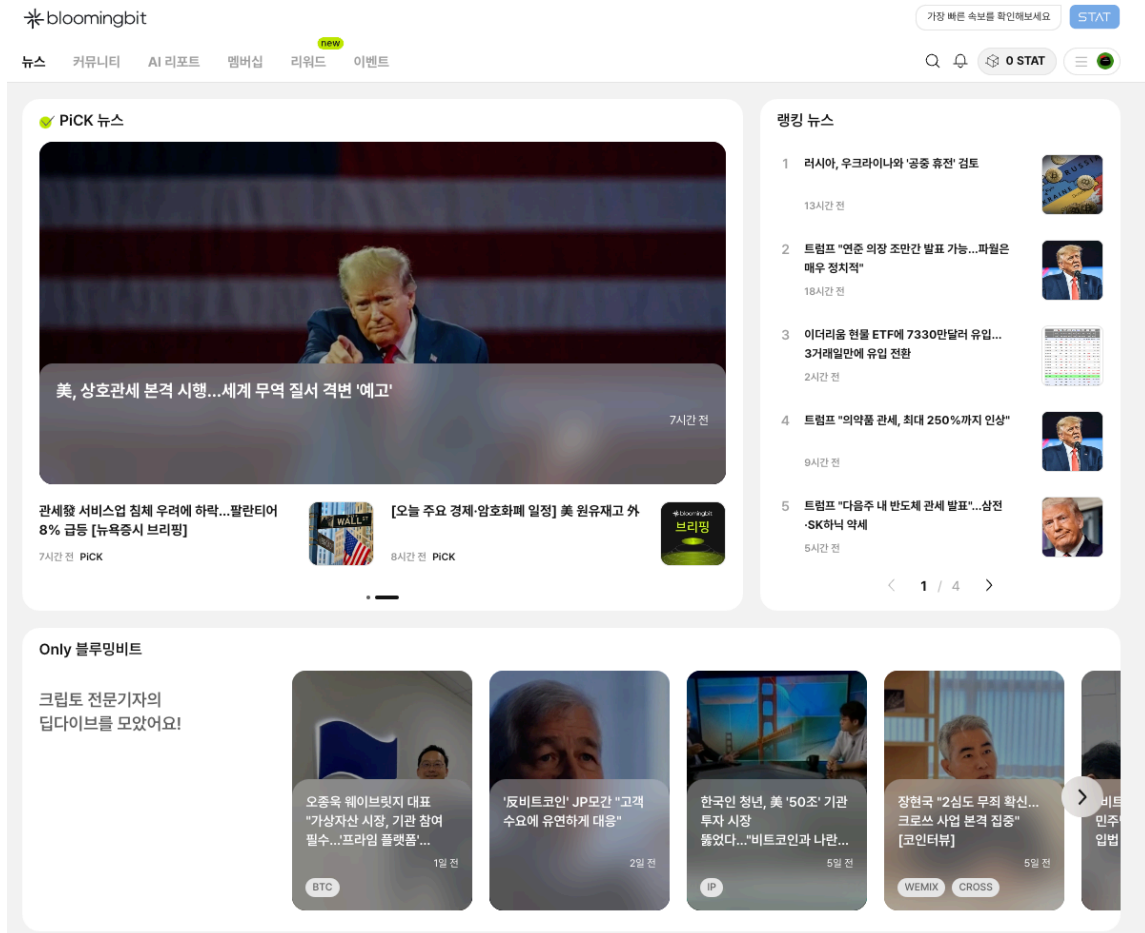


주식회사 블루밍비트

2023.08 - 2025.08 (2년) | 정규직 | 개발 | 팀원

## 메인 프로젝트: 블루밍비트 뉴스/커뮤니티 서비스 개발 및 운영

프로젝트 요약: 국내 블록체인 기반 뉴스 및 커뮤니티 플랫폼인 '블루밍비트'의 백엔드 시스템 개발 및 안정적인 운영을 담당. 사용자 경험 개선과 서비스 확장을 위한 아키텍처 개선 및 기능 구현에 기여.



[ 블루밍비트 WEB 화면 ]

## MSA 리빌딩 및 표준화 주도

2023.10 - 2023.12 개발팀 (팀원)

요약: 기존 구조의 중복 로직, 기술 부채, 통신 혼선을 해소하기 위해, Auth·User·Notification 서비스를 중심으로 전면적인 표준화 및 재설계 수행

구성: BE 3

문제:

- 기존에도 MSA로 서비스가 분리되어 있었으나, 각 서비스가 Auth 관련 코드를 중복으로 보유하고 User 서비스가 JWT 발급 로직까지 포함해 책임이 불명확함
- 서비스마다 core 디렉토리가 존재했지만, 복붙으로 관리되어 코드 규격이 제각각이고 표준화가 불가능함
- internal 통신 방식이 서비스마다 달라 장애 시 공통 대응이 어려움 -> retrofit2로 통일
- Config 관리가 코드로 하드코딩 되어있어 분산 환경 관리와 보안 유지에 부담이 큼
- 기술 부채가 누적되어 신규 기능 개발 속도가 저하

해결:

- User MSA 가 유저 및 인증 까지 관리하도록 통합
- OAuth2 기반 인증 체계로 외부 서비스 연계와 확장성을 고려해 인증 로직을 표준화
- 각 서비스가 들고 있던 core 코드를 별도의 라이브러리로 통합해 Nexus Repository에 배포
  - 공통 Response, Exception handler, internal HTTP client, Redis utils 등을 포함
- Spring Cloud Config Server와 AWS Secrets Manager를 도입해 설정과 보안 관리 체계화
- internal 통신 라이브러리 설계로 통신 규약과 payload 구조를 통일
- CI/CD 파이프라인 구축으로 배포 프로세스를 표준화하고 운영 부담을 완화
- User 도메인 관리, 회원가입·로그인 API를 재설계
- 알람 시스템과 유저 인증을 통합해 연계성을 강화

성과:

- 공통 core 라이브러리 배포로 코드 중복을 제거하고 표준화 달성
- 통신 규약과 에러 처리 표준화로 서비스 간 장애를 감소
- Config 및 보안 관리가 중앙화되어 운영 리스크를 감소
- 배포 속도 향상 및 기술 부채 해소

## Kafka 기반 이벤트 처리 설계 및 구축

2023.02 - 2023.12 | 개발팀 (팀원)

요약: 블루밍비트 전반의 서비스 간 이벤트 처리를 효율화하기 위해 Kafka를 도입, 내부 데이터 연동을 위한 이벤트 기반 아키텍처를 설계 및 구축

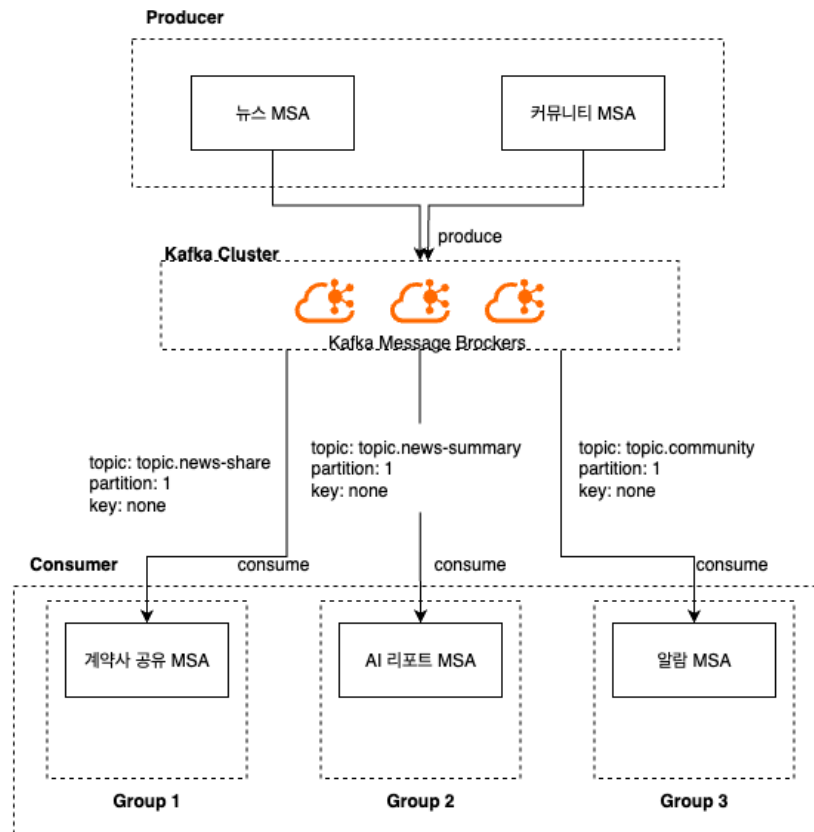
구성: BE 1 DevOps 1

배경:

- 뉴스 정보를 외부 계약사나 가상자산 거래소 등에 효율적으로 전달할 공통 이벤트 파이프라인 구축 필요
- 이후 AI 리포트 생성 시스템, 알람 시스템 등에서 이벤트 기반 파이프라인이 필요

과정 및 해결:

- 뉴스 공유용 MSA 구축과 동시에 Kafka를 도입
- 뉴스 등록 시 Kafka Topic으로 뉴스 데이터를 발행하도록 구현
- 뉴스, 커뮤니티 이벤트 생성시 알람 MSA 가 구독하여 처리
- 전체 아키텍처를 표준 Topic 체계로 설계하여 관리 효율성을 확보



- 본 Kafka 구성은 블루밍비트 서비스 내 뉴스-커뮤니티 이벤트 처리 및 다수 MSA 간 연동을 위해 직접 설계·구현함
- 초기에는 각 Topic 당 Partition 1개, Key 미사용으로 구성하였으나, 향후 트래픽 증가에 대비해 Partition 분할 및 Key 기반 분산 설계를 계획
- 안정성을 위해 replication-factor, HA 구성을 고려

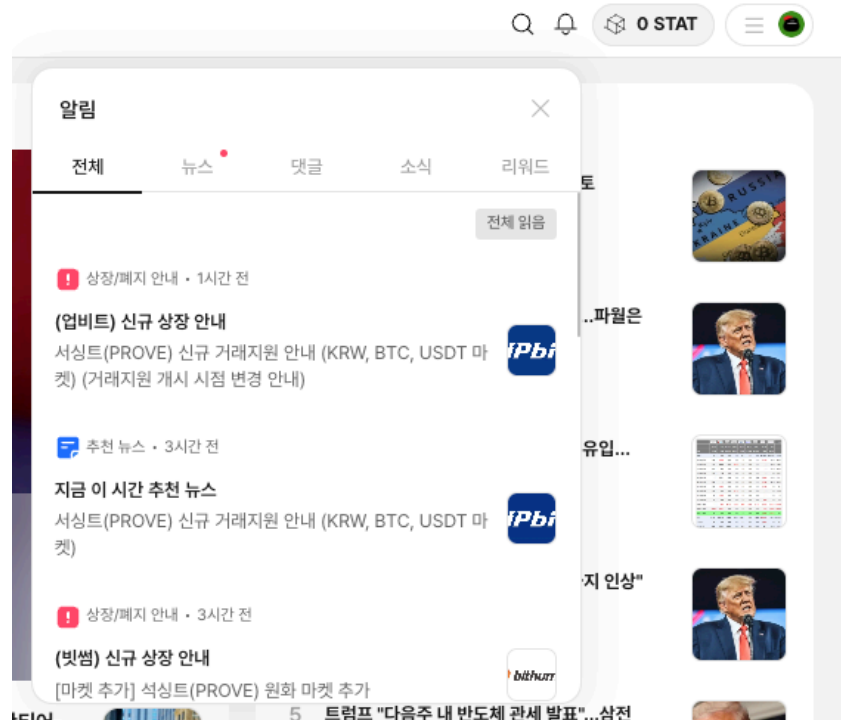
성과:

- Kafka 기반으로 데이터 흐름을 일원화해 서비스 간 연계가 유연해짐
- 기존에는 서비스 간에 REST API를 직접 호출하여 연동했으나, 이 방식은 서비스 간에 엔드포인트, 파라미터, 응답 포맷 등 구체적인 구현에 대한 의존성이 높아 한쪽 서비스의 변경이나 장애가 전파될 가능성이 있음.
- Kafka 도입 후에는 각 서비스가 Kafka에 메시지를 발행하거나 구독하는 방식으로 변경되어, 서비스 간 직접적인 호출/의존이 사라지고, 메시지 포맷만 맞추면 독립적으로 배포·운영이 가능해져 결합도가 감소.

## App Push MSA 구축 및 운영

2024.02 - 2025.04 개발팀 (팀원)

요약: AppPush MSA를 구축하고, 즉발/예약 발송, 묶음 발송, 알람센터 조회 기능을 설계·구현



[ 블루밍비트 Web 알람센터 ]

구성: BE 2, FE 2, PM 1

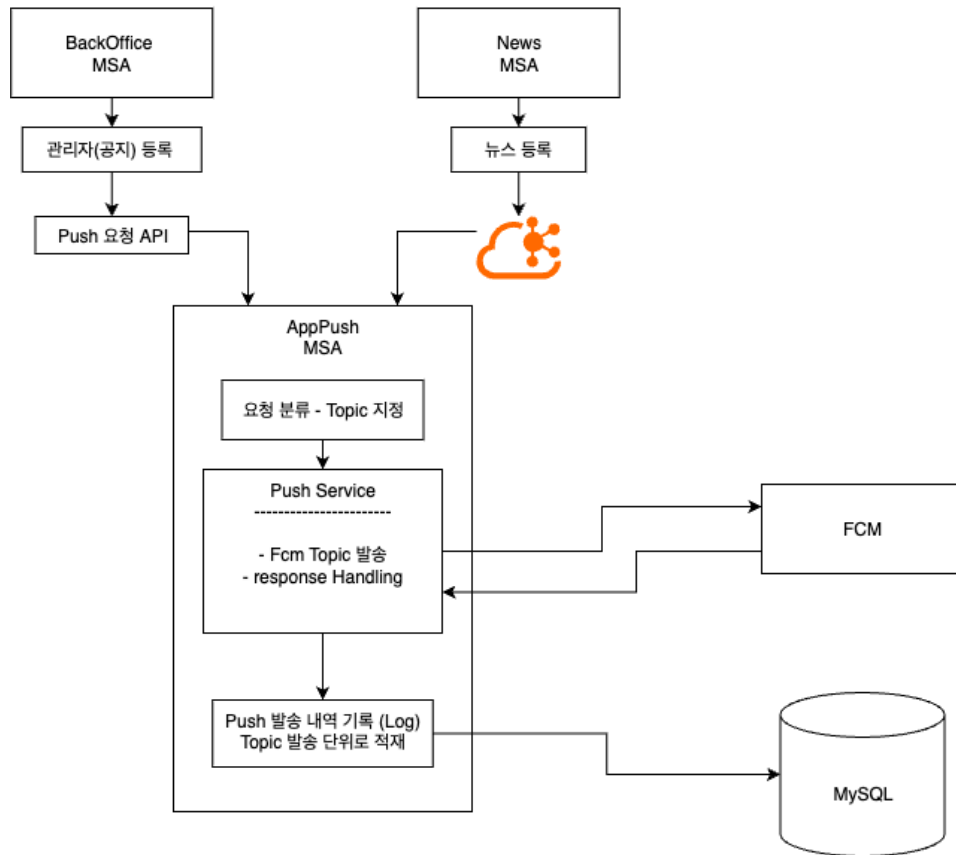
담당 업무 :

- FCM 기반 Topic/Token 발송 로직 구현
- Reserved → Item → Log 테이블 설계 및 데이터 흐름 구축
- 묶음 발송 로직 설계 (같은 유저 대상 푸시 여러 건을 하나로 통합)
- 웹·앱 공통 Push History API 개발
- 알람센터 캐시 테이블 설계 (7일 유지 후 Trigger 삭제)
- 푸시 발송 결과 Log 관리 및 통계 수집

성과 :

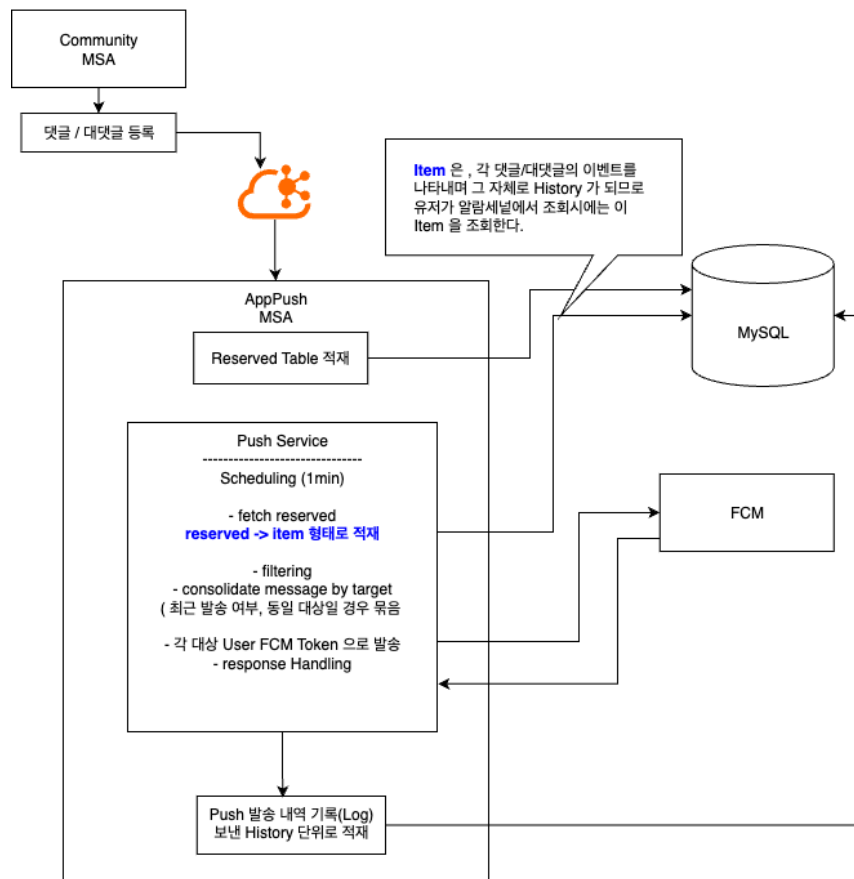
- 다중 기기별 토큰 관리로 단일 사용자 대상 푸시 누락 문제 해결
- 발송 상태 추적으로 재시도 로직을 효율화하고 서버 리소스 절감
- 예약 발송 및 묶음 발송 구현으로 중복 발송 방지 및 트래픽 감소
- Reserved → Item → Log 구조 도입으로 데이터 정합성과 운영 안정성 확보
- 알람센터 접근 시 캐시 테이블 활용으로 DB I/O 부하 완화
- 신규 푸시 타입 추가 시 유지보수 비용 최소화 및 확장성 확보

[즉시 발송]



[ App Push 시 즉시 발송되는 구조의 플로우 ]

[예약 발송]

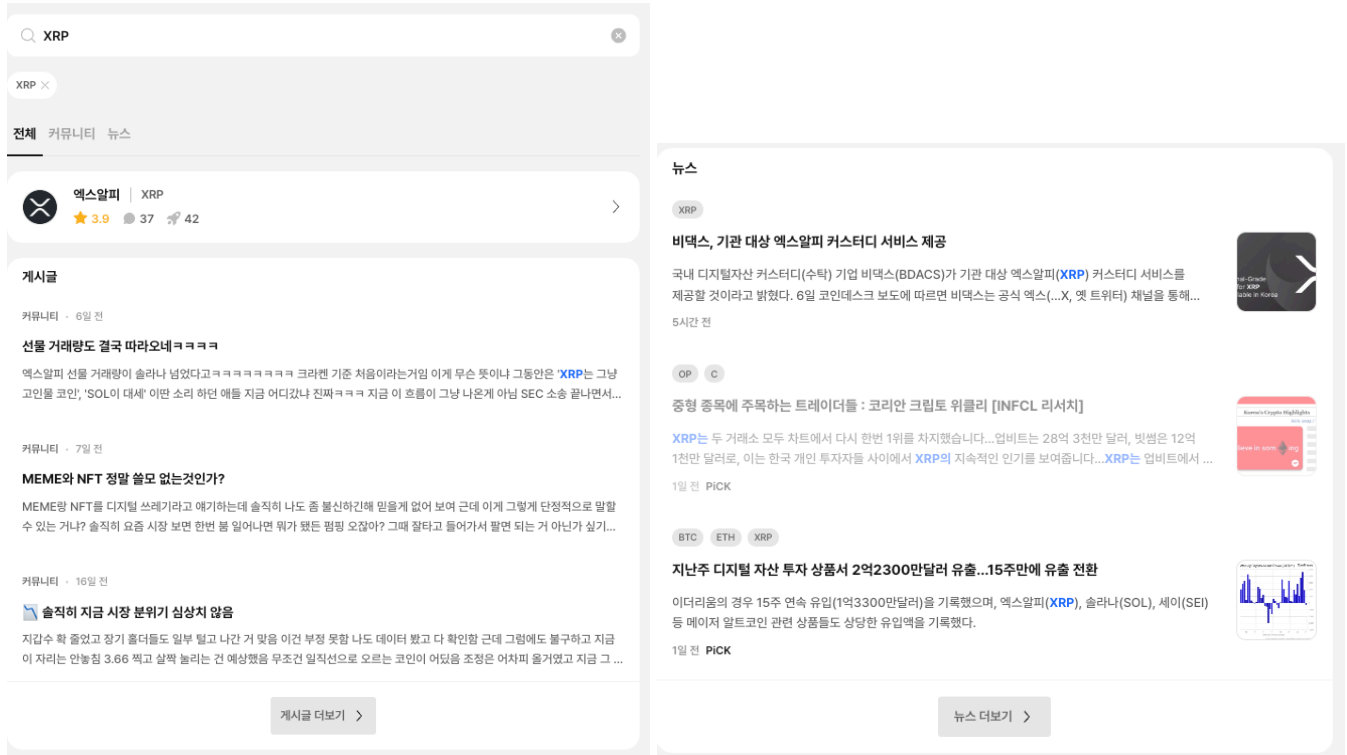


[ App Push 시 예약되어 분단위 스케줄링으로 발송 되는 구조의 플로우 ]

## 검색 전용 MSA 설계 및 속도 이슈 해소

2024.10 - 2024.10 개발팀 (팀원)

요약: 검색 시스템 통합 및 공통 쿼리 모듈 구축을 통해, 도메인 간 중복 구현을 제거



[ 블루밍비트 WEB 검색 결과 페이지 : XRP 로 검색한 게시글과 뉴스 검색 결과 ]

팀 구성: 단독

문제:

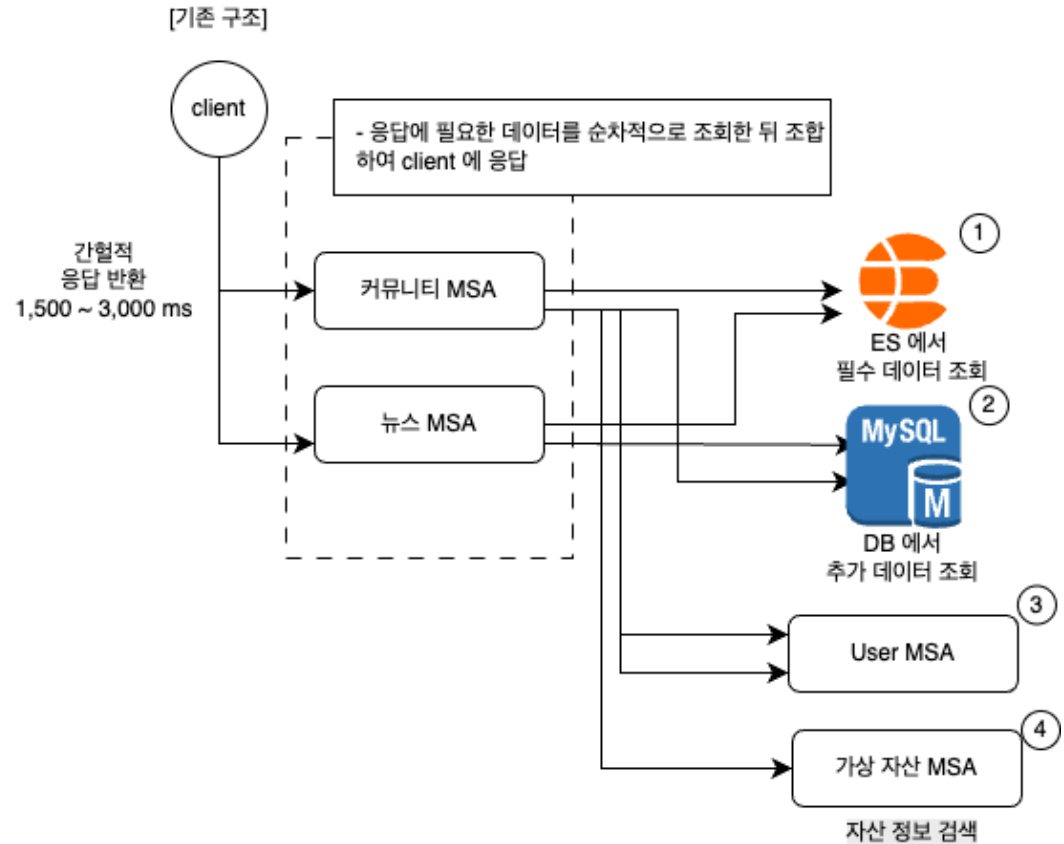
- 서비스 내 커뮤니티, 뉴스, 자산 등 도메인에서 각각 Elasticsearch를 사용해 검색을 처리.
- 검색 요청이 발생하면 각 MSA는 게시글, 댓글, 자산, 작성자 정보를 자체적으로 API 호출을 통해 순차적으로 조회
- 댓글 수가 많은 글이거나 가상자산 태그가 포함된 경우, 응답 시간이 1,500~3,000ms 까지 증가하는 현상이 반복.

해결 :

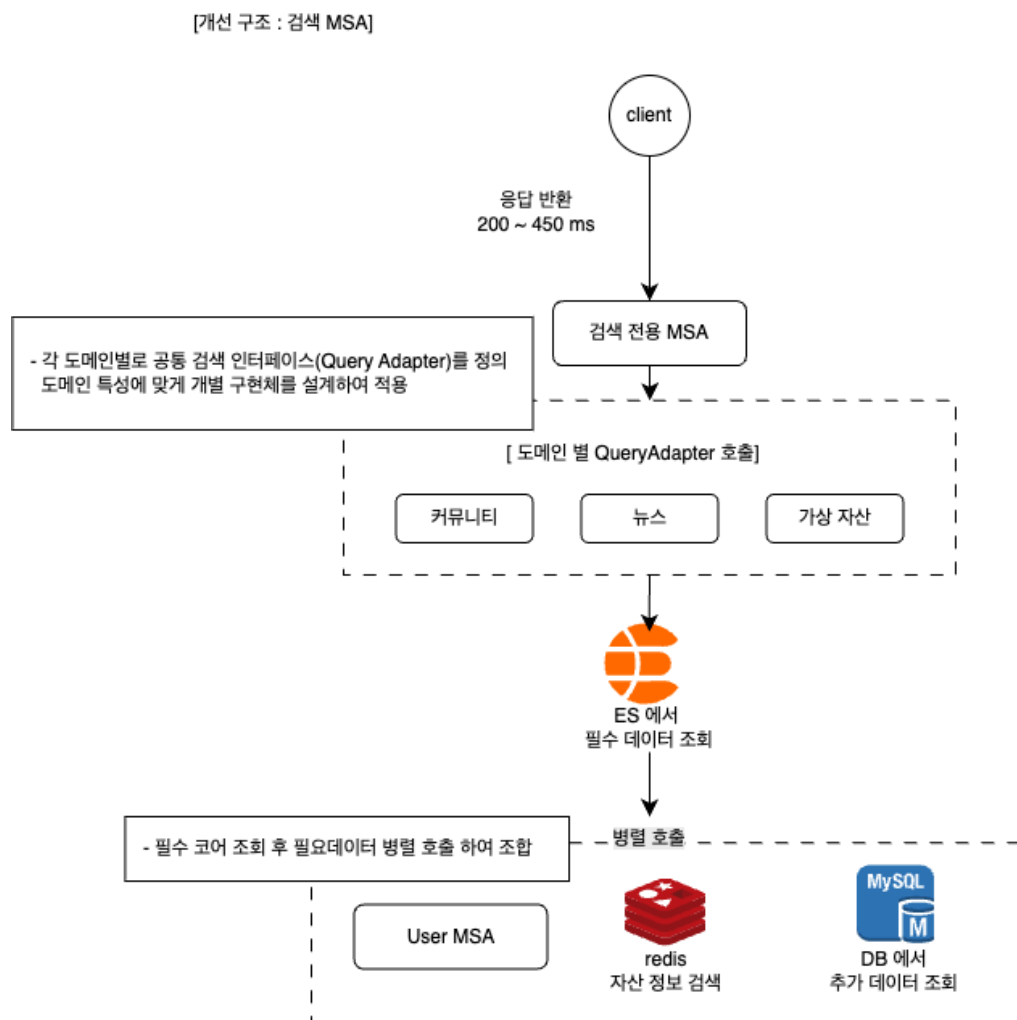
- 검색 요청이 점차 복잡해질 것을 고려하여, 검색 전용 MSA를 신규 구성
- 기존 서비스별 처리 방식을 Query Adapter 인터페이스 기반으로 통합, 각 도메인(커뮤니티, 뉴스, 자산)에 맞게 이를 구현하여 확장성과 재사용성을 확보.
- 각 도메인에 대해 검색 조건을 전달하고, 병렬로 데이터 요청 및 취합할 수 있는 구조로 재설계.
- 검색 응답 조립은 Deferred + Semaphore를 활용해 동시성 제어 및 리소스 안정성을 확보.

성과 :

- 평균 응답 시간을 200~400ms 수준으로 단축하여, 간헐적 지연 문제를 해소.
- 서비스 전반의 검색 로직을 분리하면서, 이후 검색 조건이 추가되거나 확장될 때 각 도메인에 영향을 주지 않고도 검색
- MSA 단독으로 관리·확장 가능하도록 하여 유지보수성과 확장성을 확보.



[블루밍비트 기존 검색 플로우 : 각 MSA로의 검색 요청 후 추가 데이터를 동기로 조회]



[블루밍비트 개선 검색 플로우 : 검색 MSA로의 검색 요청 후 추가 데이터를 병렬로 조회 후 반환]



## 그 외 진행 프로젝트 요약

---

- App Push System OneSignal -> FCM 교체(2025.02 ~ 05)
- WorldCoin 회원 가입 및 로그인 기능 개발 (2024.08) – JWT 기반 인증 및 World ID 연동
- AI 활용センチメンタル 분석 리포트 서비스 구축 (2024.06) – ChatGPT 기반 뉴스 분석 자동화 (Python)
- App Push 시스템 구축 (2024.02 ~ 2024.03) – OneSignal 활용한 메시지 수신 및 전송 구조 설계
- 가상자산 거래소 연결 서비스 개발 (2023.12 ~ 2024.04) – 암호화 자산 포트폴리오 API 구현
- 가상 자산 표준 코드 관리 MSA 개발 (2023.12 ~ 2024.01) – 코드 스냅샷 관리 및 Pub/Sub 동기화
- 계약사 뉴스 제공 MSA 개발 (2023.11 ~ 2023.12) – 뉴스 등록 시 자동 업로드 (Kafka 활용, RSS, FTP, SNS)
- 정보 알람봇 모듈 추가 (2023.09 ~ 2023.10) - RabbitMQ 연동 뉴스 발행 기능 추가
- 블루밍비트 2.0 런칭 (2023.08 ~ 2023.09) - BackEnd Infra 재설계

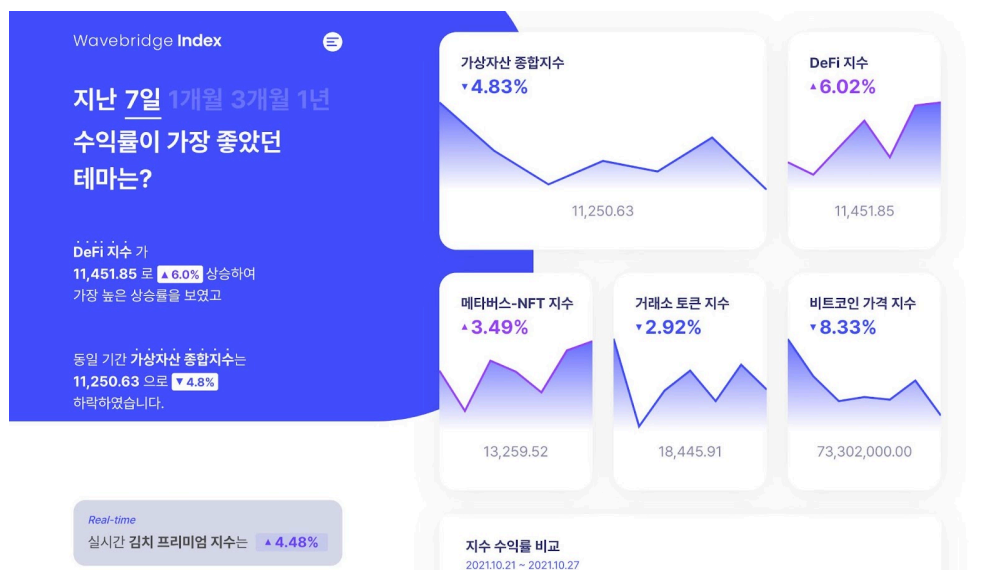
**메인 프로젝트: 웨이브릿지 인덱스(지수) 산출 시스템 개발 및 운영**

팀 구성: (초기) 단독, (중기) BE 1, FE 1, PM 1

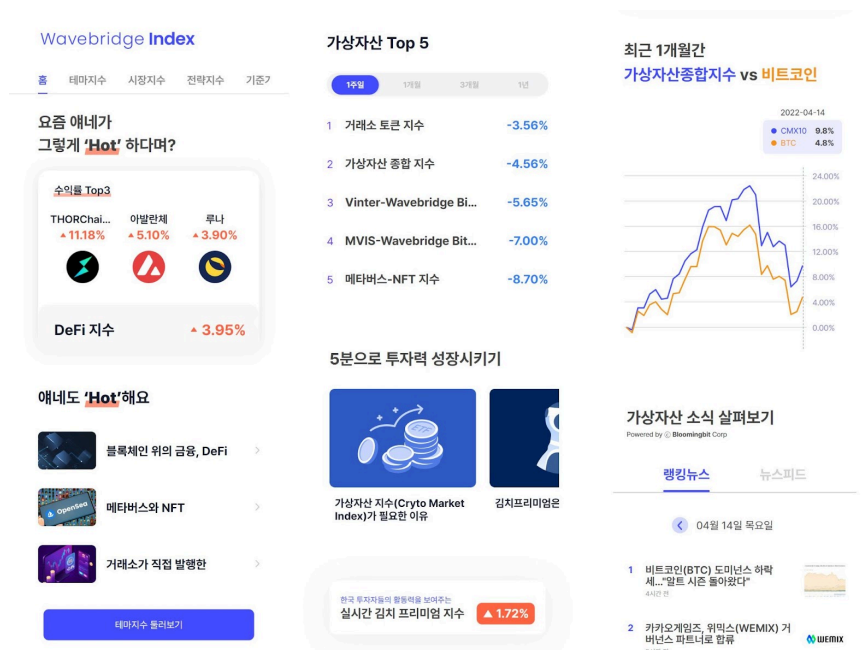
**프로젝트 요약:** 실시간 및 일간 가상 자산 지수를 산출하고 제공하는 핵심 시스템을 개발 및 운영.

고성능 데이터 처리 환경에서 데이터 파이프라인 구축을 담당하며 시스템의 안정성과 정확성 확보에 기여.

프로젝트 초기 프로토타입 단계부터 참여하여, 지수 생성 서버, 웨이브릿지 인덱스 웹페이지(Vue.js), 및 배포 환경(DevOps)까지 직접 구축했습니다. 이후 FE 및 DevOps 전문가가 합류하며 각 역할로 업무를 성공적으로 이관했습니다.



[ 웨이브릿지 인덱스 Web 화면 ]



[ 웨이브릿지 인덱스 Mobile 화면 ]

## 가상 자산 지수 산출 시 오더북 데이터 정합성 이슈

2020.12 - 2021.01 개발팀 (팀원)

요약: 실시간 지수 산출 시스템 내부에서 오더북 데이터의 정합성이 맞지 않는 이슈 해결

팀 구성: 단독

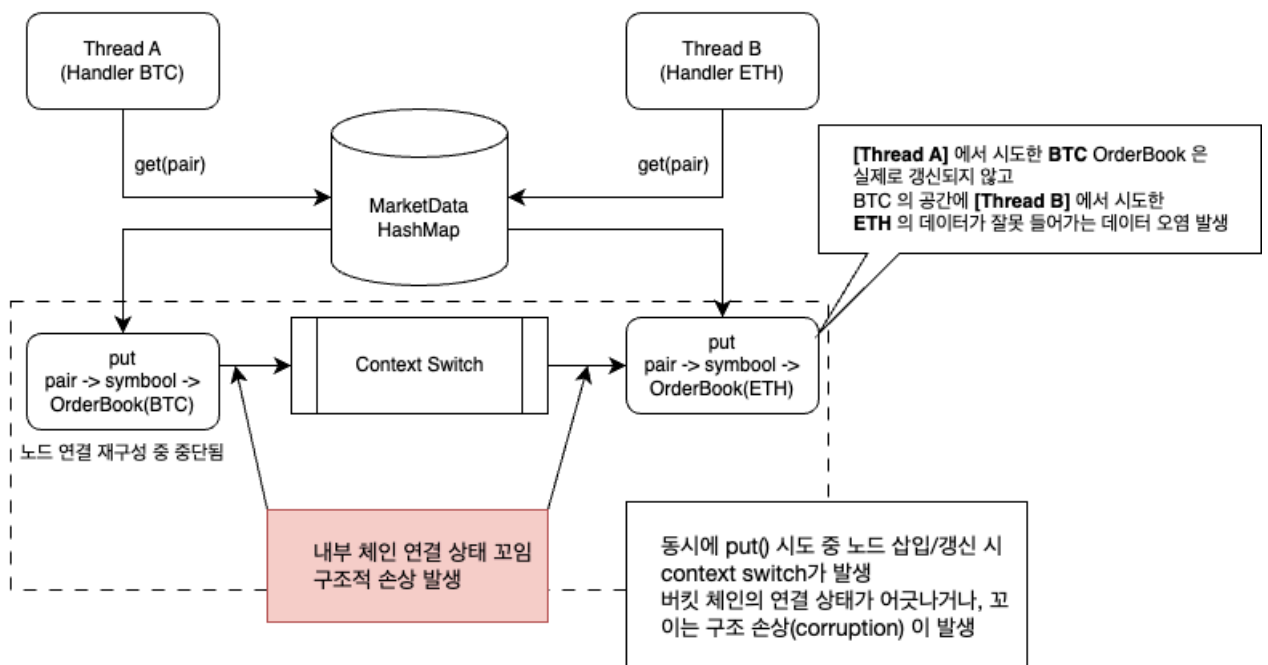
문제:

- 실시간 가상자산 지수를 산출하는 서비스에서, 프론트 차트상 **지수가 간헐적으로 튀는 현상**이 발생
- 당시 시스템은 주력 자산 5종에 대해서만 지수를 산출하고 있었으며, 실시간 가격 데이터를 저장하기 위해 '가상자산 쌍(pair) → 심볼(symbol) → 오더북 객체'의 구조를 가진 HashMap 캐시를 사용.
- 이 구조에서 가장 바깥쪽 Map이 HashMap으로 되어 있었고, 멀티스레드 환경에서 여러 핸들러가 동시에 동일한 자산 쌍(pair)에 접근하며 데이터를 업데이트하는 상황이 발생
- 그러나 Thread Dump 상에서는 병목이나 block 없이 정상 실행 중인 스레드만 관측되어, 단순한 병렬처리 문제로 판단하기 어려운 상황

원인 분석 :

- Thread Dump 기반으로 병목이 감지되지 않았지만, 동일 시점에 산출된 지수 데이터가 사용하는 가격 정보가 서로 상이하다는 점을 확인
- 이를 검증하기 위해 지수 계산에 사용된 가격 데이터를 별도의 임시 테이블에 저장하고, 동시간대의 산출 결과와 비교 분석
- 결과적으로, 멀티스레드 환경에서 HashMap이 스레드 안전하지 않다는 특성 때문에 get → put → replace 등의 메서드가 context switching 중에 중간 끼어들며 수행되면서 정합성이 깨지는 상황이 발생한 것으로 확인
- 특히 멀티스레드 환경에서 여러 스레드가 동시에 내부 캐시에 접근하는 과정에서, 다른 가상 자산의 가격 정보가 잘못된 심볼 키에 저장되는 정합성 오류가 발생. 이는 HashMap의 구조적 한계로 인해, 스레드 간 연산이 중간에 간섭되며 잘못된 데이터가 입력되는 상황.

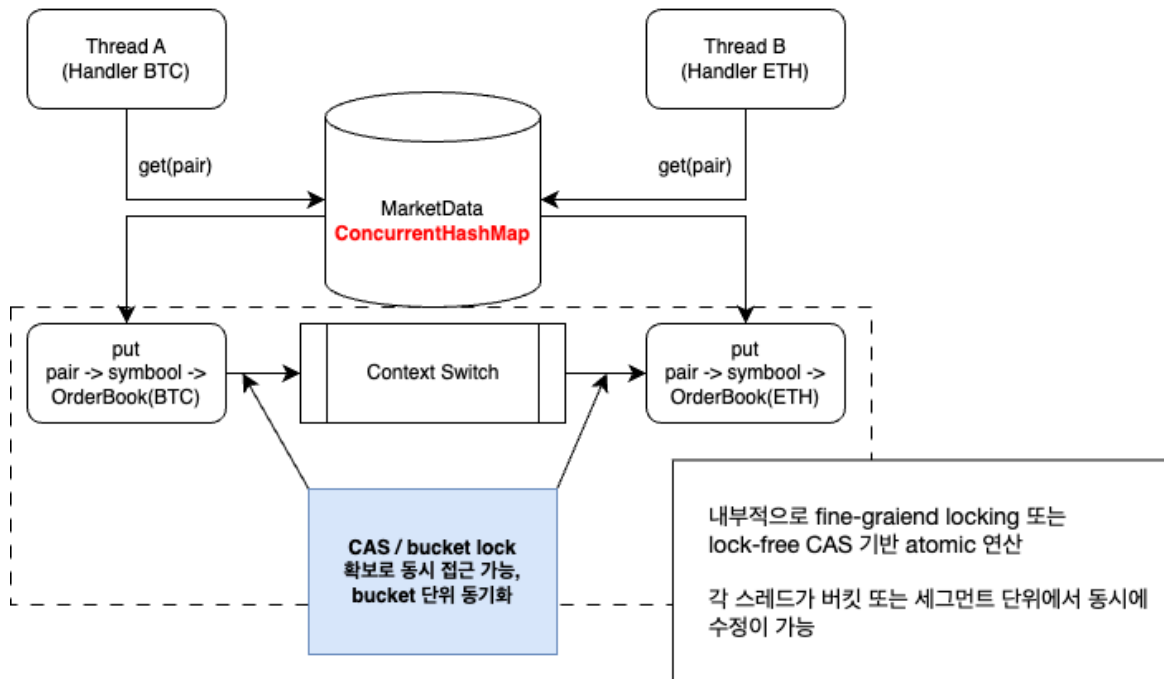
[기존 구조]



구조적으로 가능한 이유:

- get과 put, replace 등 연산은 개별적으로 atomic하지 않으며, context switch가 중간에 발생하면 노드 연결 재구성이 중단되면서 버킷 체인의 연결 상태가 꼬일 수 있음.
- 예를 들어 Thread A가 put을 하려는 순간, Thread B가 동일한 객체에 접근해 내용을 교체하거나 참조하는 경우, 내부 참조가 꼬이거나 오염된 객체 상태가 저장될 수 있음.
- 특히 Map 내부의 노드 연결 구조가 변경되는 타이밍에 교차로 접근하게 되면, 실제로는 서로 다른 키임에도 내부 상태가 공유되는 상황이 발생 가능.

[개선 구조]



해결:

- 바깥쪽 Map을 **ConcurrentHashMap**으로 전면 교체하여 멀티스레드 환경에서 동기화를 보장
- 객체 교체 이후에는 이전 객체를 초기화하여 GC 대상이 되도록 설계

## ElasticJob + Zookeeper 기반 분산 스케줄링 설계

2022.02 - 2022.03 개발팀 (팀원)

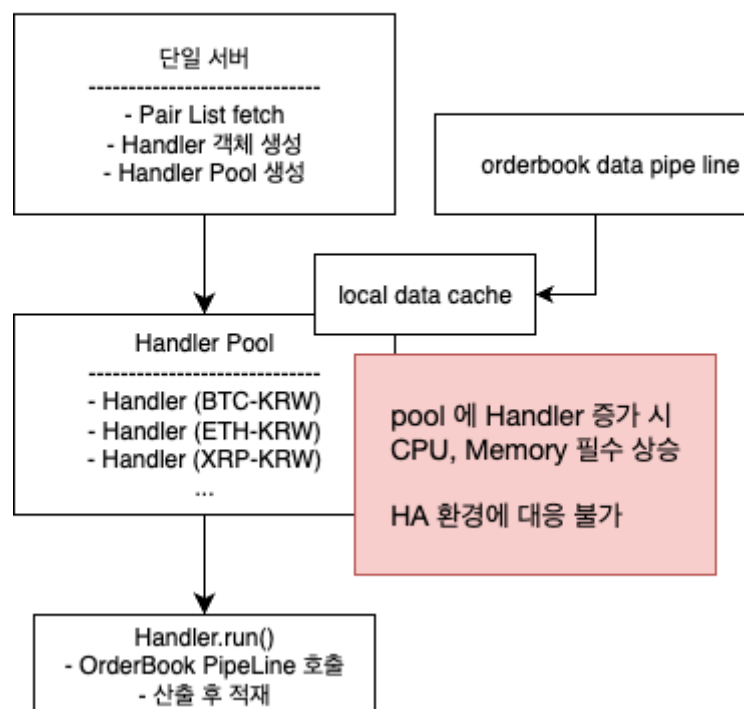
요약: 실시간 지수 산출 시스템의 단일 서버 한계 해결

팀 구성: 단독

문제:

- 기존 시스템은 단일 서버에서 Handler를 생성하고, 지수가 추가되면 새로 Handler 를 계속 추가하는 형태
- 단일 서버 한계로 장애 발생 시 전체 서비스 중단
- Handler 수 증가 시, 멀티 스레드로 병렬처리 되지만, 스레드풀이 차면서 지연 발생
- 하드코딩된 스케줄 주기로 유지보수 부담 증가

[기존 구조]

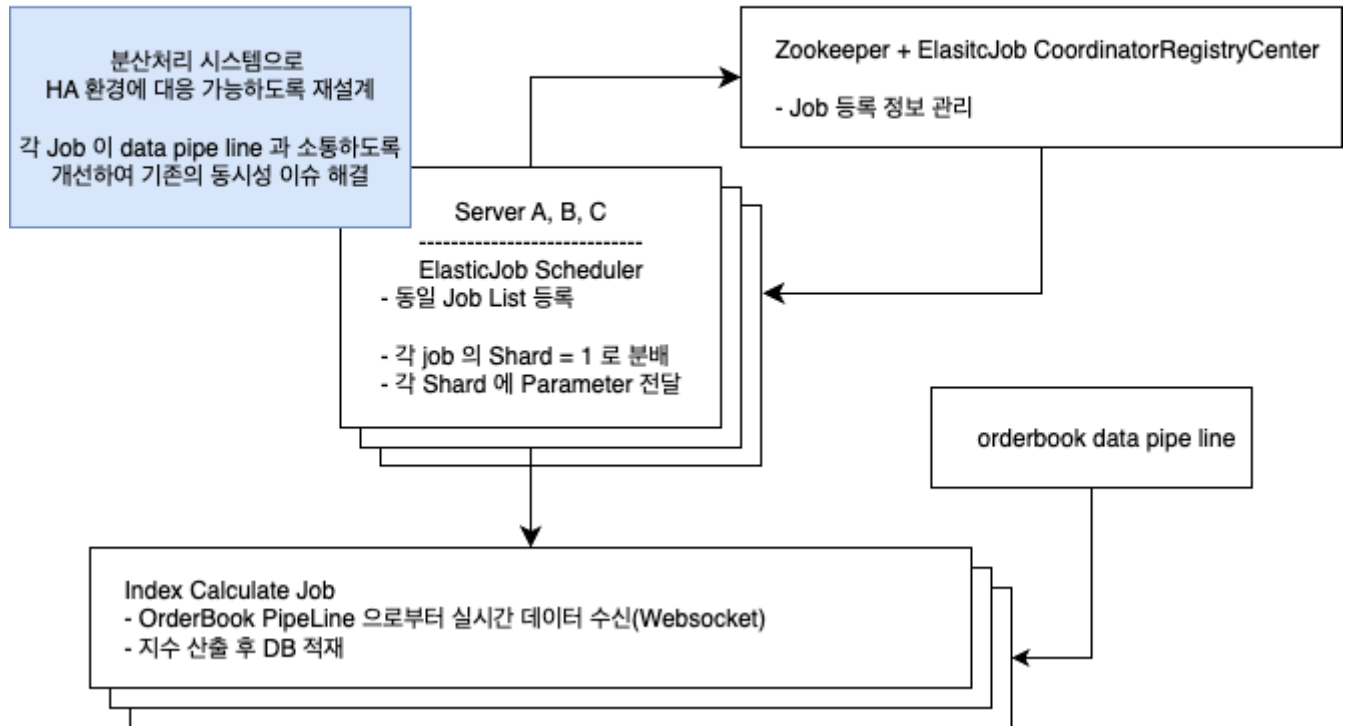


[ 기존 단일 서버에서의 지수 산출 방식 ]

개선 :

- ElasticJob + Zookeeper 기반으로 시스템 전면 재설계
- Job 단위를 지수(Pair) 단위로 정의하고 Shard=1로 설정 → 각 Job이 독립적으로 실행되도록 구성
- Job Parameter로 지수별 산출 주기와 로직을 유연하게 주입
- 기존에 있었던, 동시성 이슈의 구조적 원인 제거 (Local Cache 제거)
- 각 Job이 OrderBook Pipeline과 독립적으로 연결해 데이터를 수신
- Job 간 OrderBook 데이터 공유를 원천적으로 차단 → 데이터 정합성 확보

[개선 구조]



[ HA 환경으로 분산된 환경에서의 지수 산출 방식 ]

성과 :

- HA 확보 및 장애 시 무중단 운영 가능
- Pair 수 증가에도 안정적 성능 유지
- Local Cache 제거로 데이터 정합성 확보
- 신규 Pair 추가 시 무배포 적용 가능
- 시스템 수평 확장성 확보

## 그 외 진행 프로젝트 요약

---

- MarketInfo & MarketData 수집 시스템 (2022.12 ~ 2023.05) – 글로벌 확장성 고려한 모델링 설계
- 표준 코드 운영 및 리팩토링 (2022.12 ~ 2023.12) – 관리 자동화



## JT인터네셔널

2019.06 - 2019.10 (5개월) 정규직

---

### 기업 웹사이트 관리 및 제작

- 웹사이트 퍼블리싱 및 유지보수
- 제품상세페이지 제작
- 카페24 웹페이지 제작
- 오픈마켓, 소셜마켓 관리



## 링크

---

🔗 Github

<https://github.com/devtaco30>