

김규원, 김민태, 김소철, 곽태영, 이승준, 임용덕, 윤서정, 한민, 홍영석

중고 대여 플랫폼 빌리마켓

아키텍처 설명 및 데모

B I L L I 2 M A R K E T

I N D E X

- 01 **기획 의도**
- 02 **개발 환경**
- 03 **데모**
- 04 **구현 방식**
- 05 **결론**

01

기획 의도

- ① 빌리마켓 서비스 목표
- ② 빌리마켓 기본 기능
- ③ 빌리마켓 차별화 기능

01

빌리마켓 기획 의도

중고 거래 플랫폼 빌리마켓 서비스의 목표와 주요 기능

서비스 목표

일반적으로 구매 후 잘 사용하지 않는 물품에 대한 중고 대여 시장 구축으로 사용자의 편의성 증대

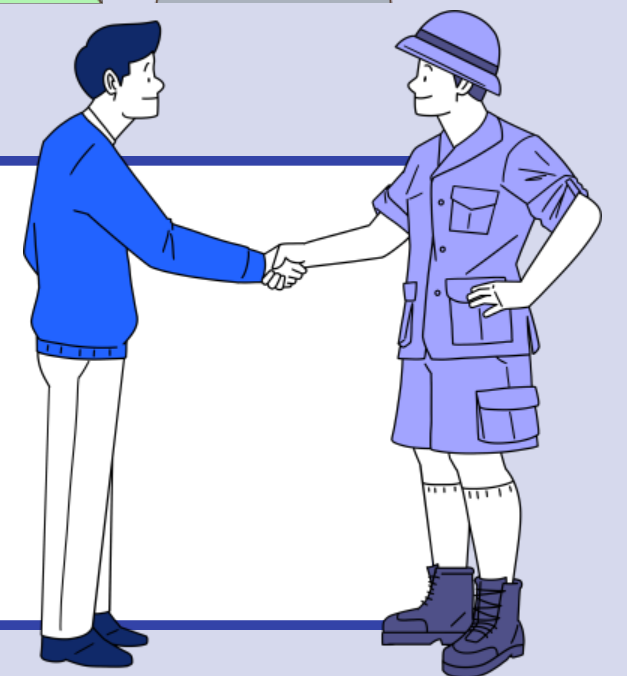
기존 중고 거래 플랫폼

사용자끼리 중고 물품을 판매·구매하는 것에 그치는 경향이 있음.



빌리마켓

일반적인 중고 거래 플랫폼에서 벗어나 대여·반납할 수 있는 새로운 중고 거래 플랫폼으로 사용자에게 더 큰 편의성과 가치를 제공.



01

빌리마켓 기획 의도

중고 거래 플랫폼 빌리마켓 서비스의 목표와 주요 기능

서비스 주요 기능 (기본 기능)

01

회원가입·로그인 기능

회원가입, 네이버 ID를 통해 빌리마켓에 로그인할 수 있습니다.

02

대여 게시글 작성·수정·삭제 기능

게시글에 물품 사진, 설명, 대여 기간을 작성·수정·삭제할 수 있습니다.

03

물품 후기 작성, 사용자 평점 기능

대여한 물품과 사용자의 후기를 작성할 수 있으며, 사용자 평점에 해당 점수가 적용됩니다.

04

물품 검색 기능

사용자 위치 기반, 텍스트, 카테고리 검색을 통해 중고 물품을 검색할 수 있습니다.

05

찜 기능

위치 기반으로 내 주변 6km 반경 내 물품을 보고, 찜 목록에 저장할 수 있습니다.

06

채팅 기능

대여자에게 채팅을 걸어 채팅을 통해 날짜 등을 결정하여 대여를 요청할 수 있습니다.

07

마이페이지 기능

프로필 사진 설정/변경, 올린 물품 목록과 빌린 물품, 찜 목록, 후기를 확인할 수 있습니다.

01

빌리마켓 기획 의도

중고 거래 플랫폼 빌리마켓 서비스의 목표와 주요 기능

서비스 주요 기능 (차별화 기능)

01

보증금 기능

물품에 대한 보증금을 대여 시 대여금과 함께 결제하여
반납 시 물품 상태 훼손·분실 여부에 따라 대여자, 반납자에게 보증금을 다시 지급합니다.

02

물품 대여·예약·반납 기능

채팅을 통해 중고 물품을 대여·예약·반납할 수 있습니다.

03

빌리페이(채팅 결제) 기능

채팅을 통해 계약을 진행하고, 결제를 통해 물품 대여 금액을 지불할 수 있습니다.

02

개발 환경

- ① Agile Process
- ② CI / CD
- ③ Architecture

02

Agile Process

Jira



프로젝트 / 빌리마켓 / NONE 보드
NONE 1 스프린트 2

0일 남음 스프린트 완료

인사이드

할 일 5 진행 중 3 개발완료 & 리뷰대기 1 리뷰검토 완료 0

BILL-14 해야 할 일 2개 하위

회원가입 서버로직 구현

로그인 UI 구현

회원가입 UI 구현

다른 이슈 3개 이슈

프로젝트 / 빌리마켓 / BILL 보드
스프린트 7

0일 남음 스프린트 완료

인사이드

할 일 10 진행 중 3 개발완료 & 리뷰대기 5 리뷰검토 완료 6

BILL-110 해야 할 일 2개 하위 작업 알림 기능 구현

BILL-111 해야 할 일 1개 하위 작업 마이페이지 구현

FE 마이페이지 화면 및 닉네임 수정 기능 구현 BILL-148

BILL-154 진행 중 2개 하위 작업 채팅 심화 기능 구현

BE 계약기간 종료 알림 구현 BILL-159

BE Contract 관리 API 및 서버로직 구현 BILL-155

BILL-112 해야 할 일 3개 하위 작업 내가 올린 물품 보기 기능 구현

FE 내가 올린 물품 화면 UI 구현 BILL-149

DB 내가 올린 물품 불러오기 쿼리 생성 BILL-151

BE GET /items/me API 구현

Epic	2월	3월	4월	5월	6월
BILL-2 첫 Jira 티켓 발급					
BILL-51 기본 기능 구현					
BILL-52 차별 기능 구현					
BILL-53 인수 테스트 완료					
+ 만들기 Epic					

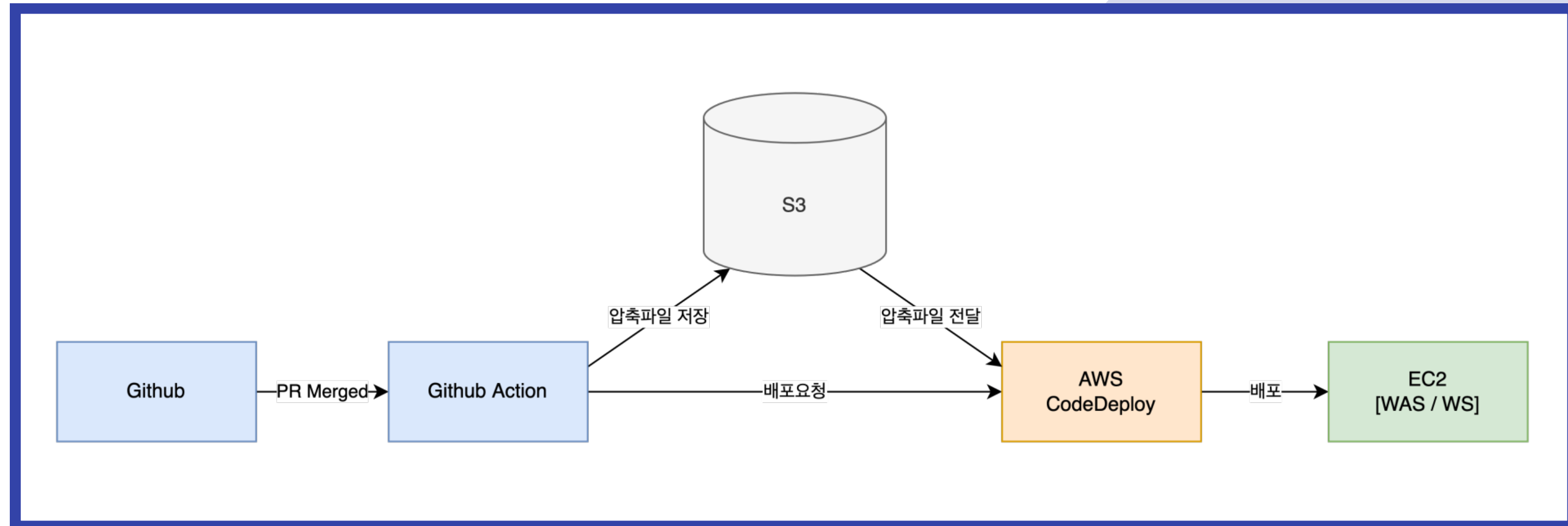
02

CI / CD

Github Action, CodeDeploy

01

Github Action - AWS Code Deploy

Continuous Integration
Continuous Deployment

02

CI / CD

Github Action, CodeDeploy

BILL2MARKET

< Github Action >

<p>✔ Cloud Migration 05/23 Java CI with Gradle #66: Commit b0ec73e pushed by mi2ntae</p>	dev	<p>📅 14 days ago 🕒 1m 32s</p>	...
<p>✔ Update CD_gradle.yml CD with AWS S3 #61: Pull request #2 synchronize by mi2ntae</p>	dev	<p>📅 17 days ago 🕒 2m 41s</p>	...
<p>✔ Update after_install.sh Java CI with Gradle #65: Commit 204b54c pushed by mi2ntae</p>	dev	<p>📅 17 days ago 🕒 1m 33s</p>	...
<p>✔ Update CD_gradle.yml CD with AWS S3 #60: Pull request #2 synchronize by mi2ntae</p>	dev	<p>📅 17 days ago 🕒 2m 1s</p>	...
<p>✔ Update after_install.sh Java CI with Gradle #64: Commit 3ca7387 pushed by mi2ntae</p>	dev	<p>📅 17 days ago 🕒 1m 20s</p>	...
<p>✔ Update CD_gradle.yml CD with AWS S3 #59: Pull request #2 synchronize by mi2ntae</p>	dev	<p>📅 19 days ago 🕒 2m 16s</p>	...
<p>✔ Update after_install.sh Java CI with Gradle #63: Commit 5d6d4cf pushed by mi2ntae</p>	dev	<p>📅 19 days ago 🕒 1m 20s</p>	...
<p>✔ Update CD_gradle.yml CD with AWS S3 #58: Pull request #2 synchronize by mi2ntae</p>	dev	<p>📅 19 days ago 🕒 2m 0s</p>	...
<p>✔ Update after_install.sh Java CI with Gradle #62: Commit a3d8b6d pushed by mi2ntae</p>	dev	<p>📅 19 days ago 🕒 1m 31s</p>	...
<p>✔ Update CD_gradle.yml CD with AWS S3 #57: Pull request #2 synchronize by mi2ntae</p>	dev	<p>📅 19 days ago 🕒 2m 2s</p>	...

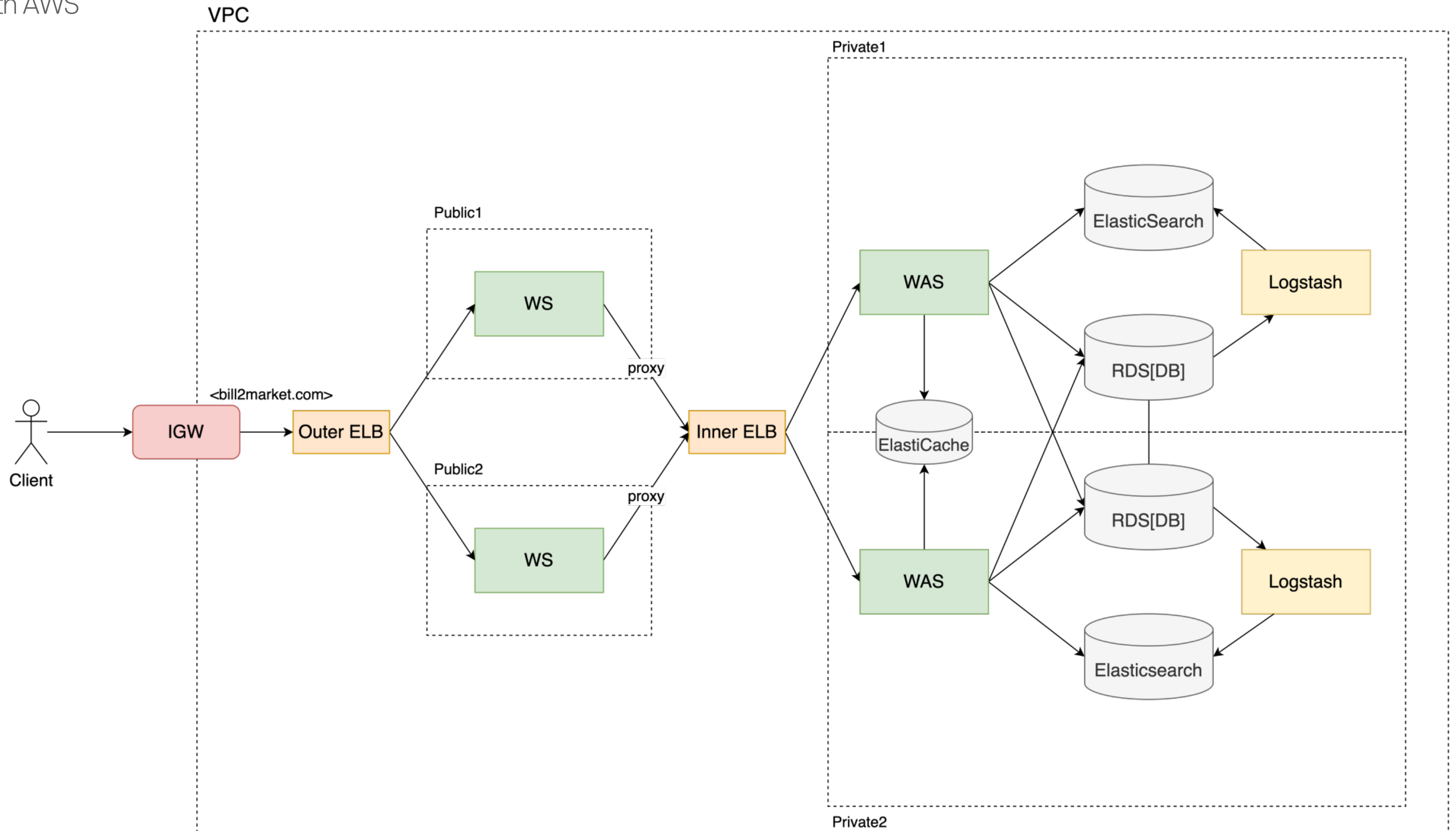
< Code Deploy >

○	d-VK6ETH72H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Back-Deploy	Bill2-WAS2-Deploy	s3://bill2ma...	사용자 작업	6월 4, 2022 2:16 오후 (UTC+9:00)	6월 4, 2022 2:21 오후 (UTC+9:00)
○	d-E5CI7Y72H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Back-Deploy	Bill2-WAS1-Deploy	s3://bill2ma...	사용자 작업	6월 4, 2022 2:16 오후 (UTC+9:00)	6월 4, 2022 2:21 오후 (UTC+9:00)
○	d-EFIN1J51H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Back-Deploy	Bill2-WAS2-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 10:55 오후 (UTC+9:00)	6월 3, 2022 11:00 오후 (UTC+9:00)
○	d-08C92MR1H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Back-Deploy	Bill2-WAS1-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 10:55 오후 (UTC+9:00)	6월 3, 2022 11:00 오후 (UTC+9:00)
○	d-KGI9BMR1H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Back-Deploy	Bill2-WAS2-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 10:16 오후 (UTC+9:00)	6월 3, 2022 10:21 오후 (UTC+9:00)
○	d-XMFM4BS1H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Back-Deploy	Bill2-WAS1-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 10:16 오후 (UTC+9:00)	6월 3, 2022 10:21 오후 (UTC+9:00)
○	d-JLE5FAQ1H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Front-Deploy	Bill2-WS2-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 8:38 오후 (UTC+9:00)	6월 3, 2022 8:44 오후 (UTC+9:00)
○	d-N48G3FQ1H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Front-Deploy	Bill2-WS1-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 8:38 오후 (UTC+9:00)	6월 3, 2022 8:44 오후 (UTC+9:00)
○	d-ZYGV91Q1H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Back-Deploy	Bill2-WAS2-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 8:20 오후 (UTC+9:00)	6월 3, 2022 8:26 오후 (UTC+9:00)
○	d-JK2Q89Q1H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Back-Deploy	Bill2-WAS1-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 8:20 오후 (UTC+9:00)	6월 3, 2022 8:26 오후 (UTC+9:00)
○	d-2LHXY8N1H	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Front-Deploy	Bill2-WS1-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 5:53 오후 (UTC+9:00)	6월 3, 2022 5:59 오후 (UTC+9:00)
○	d-1QDY0U1W4	✔ 성공	현재 위치	EC2/온프레미스	Bill2-Front-Deploy	Bill2-WS2-Deploy	s3://bill2ma...	사용자 작업	6월 3, 2022 5:41 오후 (UTC+9:00)	6월 3, 2022 5:53 오후 (UTC+9:00)

02

Architecture

with AWS



04

구현 방식

주요 기능 구현 방식

- ① 주변 물품 탐색
- ② 카테고리 검색
- ③ 키워드 검색
- ④ 채팅
- ⑤ 빌리페이

04

주변 물품 탐색

- ST_DISTANCE_SPHERE

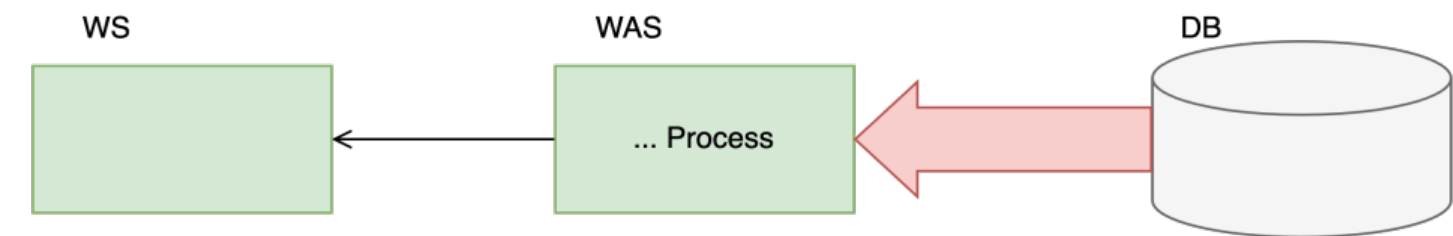
01

ST_DISTANCE_SPHERE

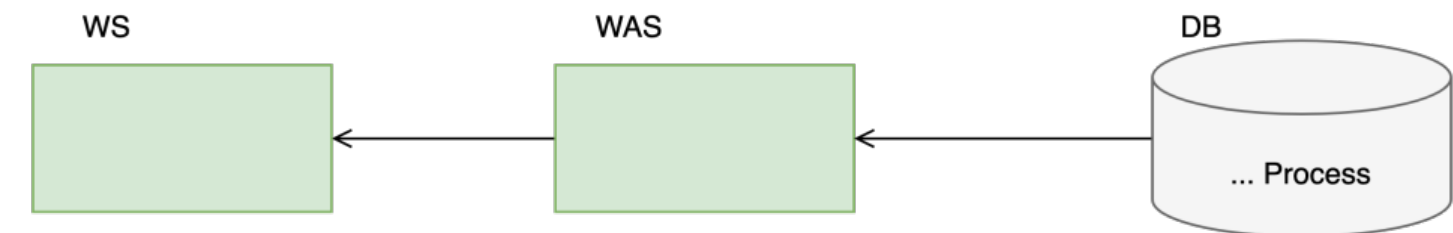
Maria DB에서 제공하는
ST_DISTANCE_SPHERE 함수를 사용하여
사용자의 좌표로부터 6KM이내의 물품만
검색되는 주변 물품 탐색 구현



WAS에서 데이터를
처리할 경우



DB에서 데이터를
전처리할 경우



04

카테고리 검색

- QueryDSL

01

기존 쿼리문의 한계

쿼리문에서 다양한 조건을 처리하는 것에 대해 일반적인 JPA를 사용하는 것은 한계 존재



02

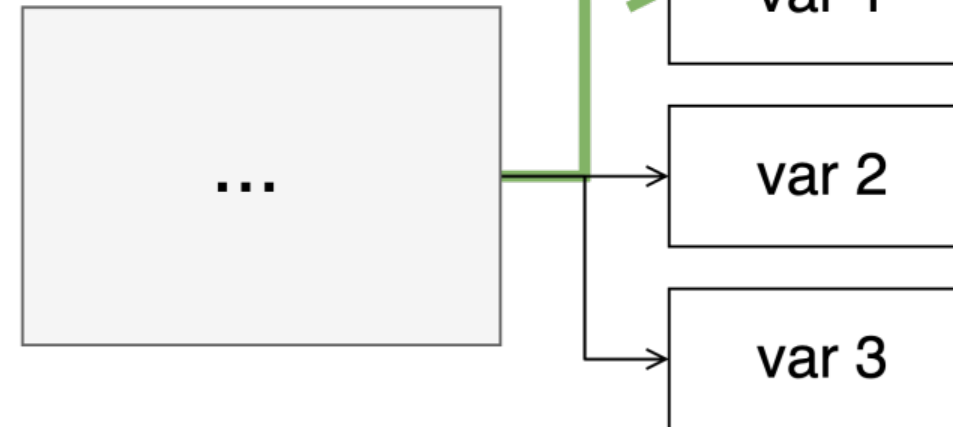
QueryDSL 사용

QueryDSL을 사용하여 동적쿼리를 생성 후 다양한 조건을 처리할 수 있도록 함

Native Query



Query DSL



04

키워드 검색

- ElasticSearch

01

Elastic Search

Elastic Search를 활용하여 검색 기능 구현

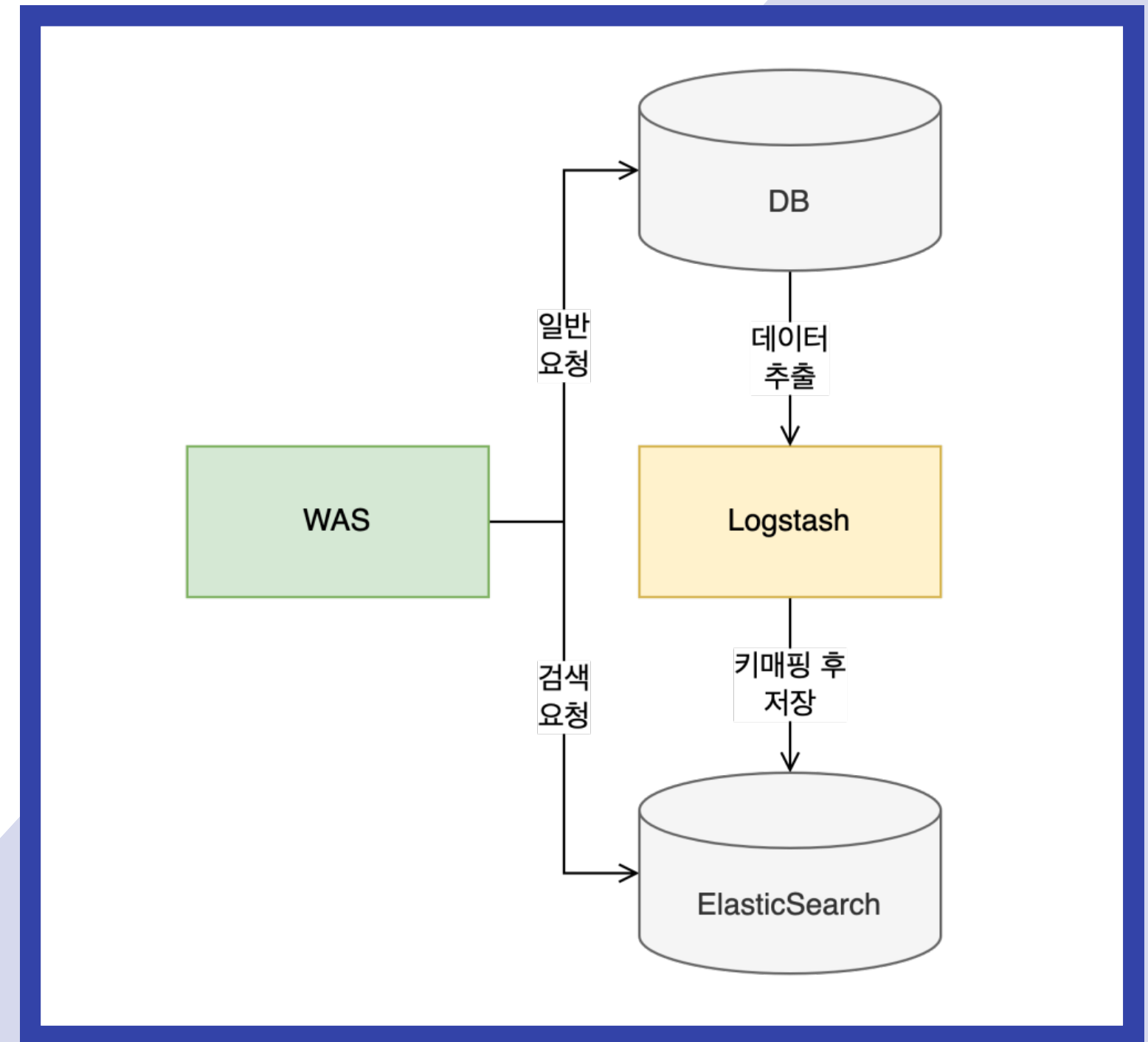
02

한글 형태소 분석

검색 기능 향상을 위한 한글 형태소 분석 사용



elasticsearch



04

채팅

- 웹소켓, STOMP

01

웹 소켓, STOMP를 이용한 채팅기능

실시간 채팅이 가능하도록 웹소켓을 사용한 채팅
기능 구현
STOMP를 사용하여 각 채팅방마다 Chat ID로 구
독하도록 구현

02

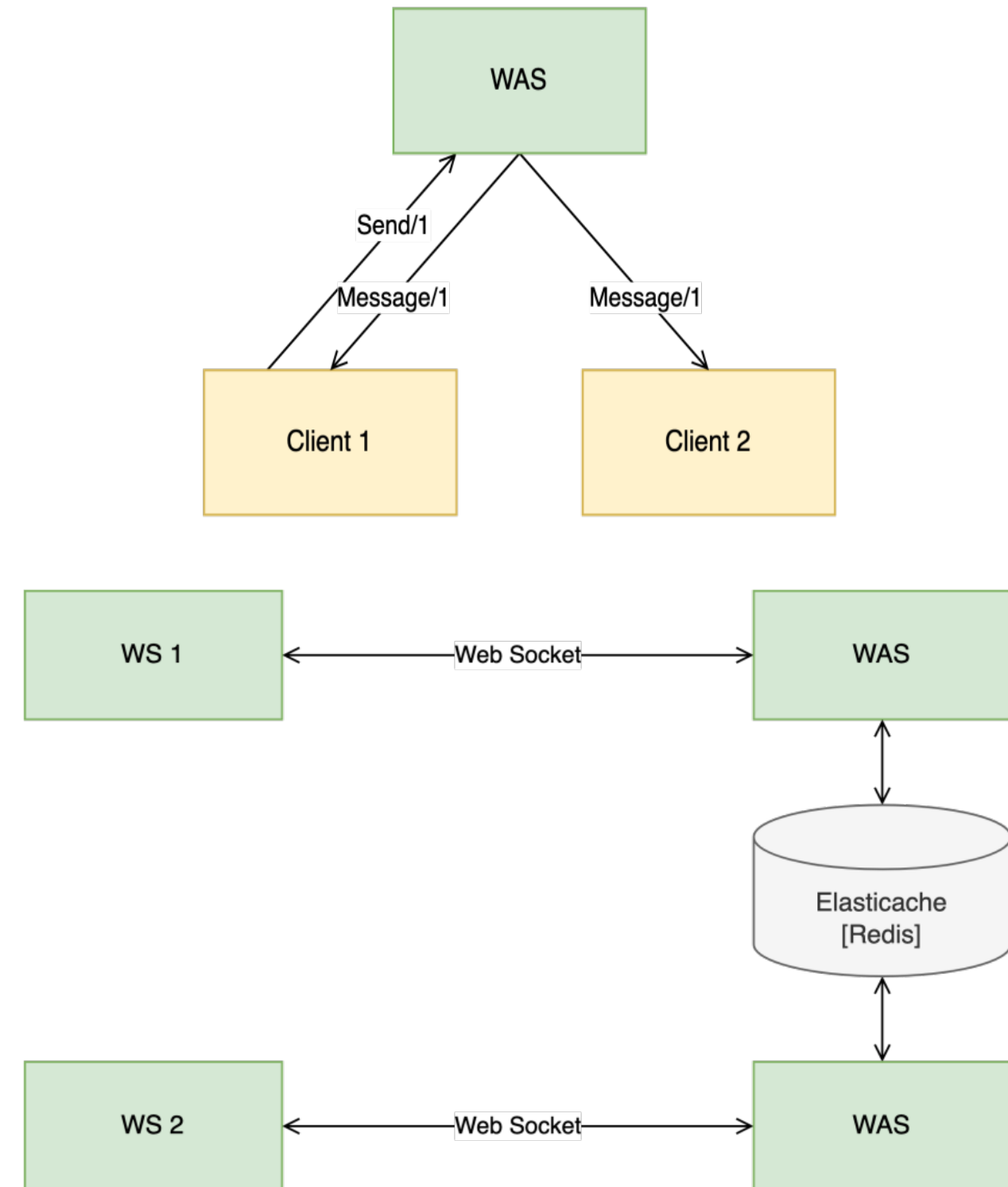
Redis를 사용한 서버 캐시 활용

이중화를 했기 때문에 STOMP의 구독정보를 동기
화 해줄 수 없는 문제점이 발생
Redis를 서버캐시로 사용하여 구독 상태와 채팅창
의 S3 주소를 저장

03

채팅 데이터 S3에 저장

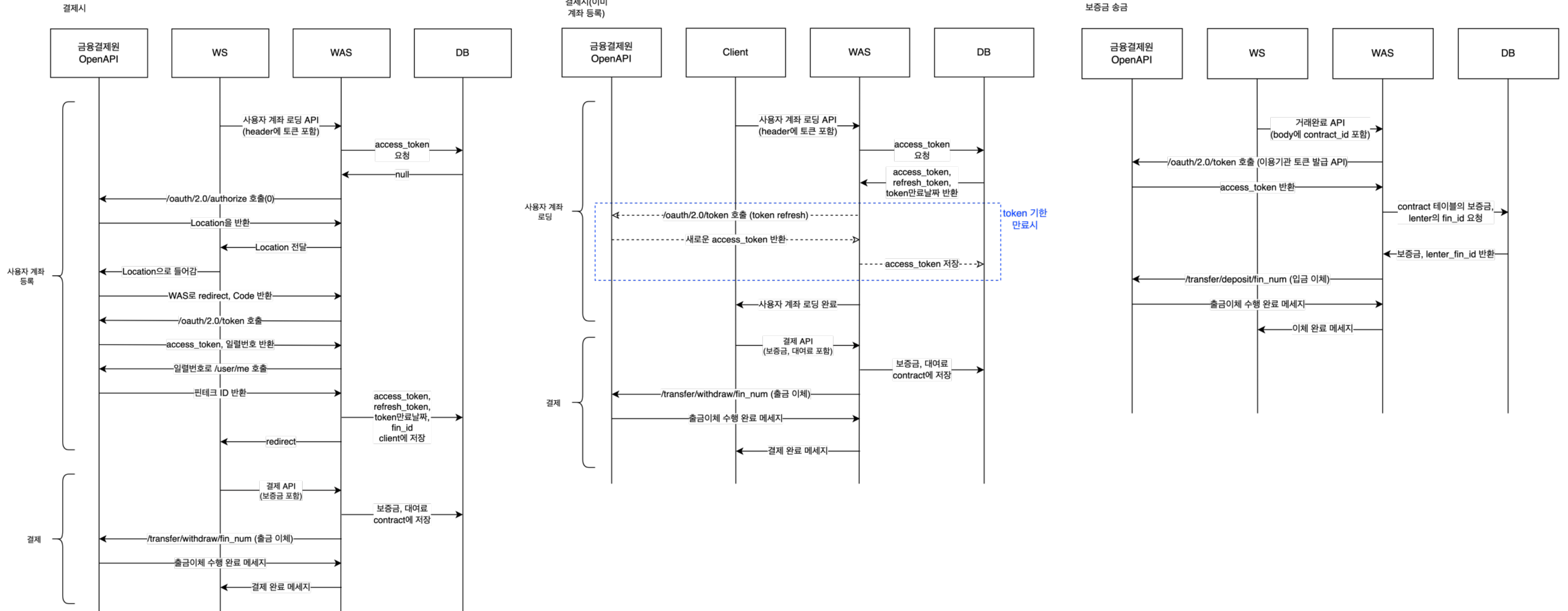
채팅의 메시지 데이터들은 대용량 데이터이기 때문
에 이를 RDS에 그대로 저장할 경우 비용적인 문제
발생
채팅 데이터는 S3에 저장 후 링크만 RDS에 저장



04

빌리페이

- 금융결제원 오픈API



04

빌리페이

- 금융결제원 오픈API

01

OAuth 2.0 인증을 통한 통신

인증과 권한 부여를 동시에 제공하는 인증 프로토콜

02

3-legged

사용자와 빌리마켓, 금융 결제원 사이의 통신을 진행하는 3-legged를 통해 계좌 등록 및 결제를 구현

03

2-legged

빌리마켓과 금융 결제원 사이에서만 통신을 진행하는 2-legged를 통해 출금을 구현



04

빌리페이

- 금융결제원 오픈API

01

CSRF 공격 방지

Redis 캐시에 난수를 저장하여
redirect시 URL을 통한 CSRF 공격 방지

05

결론

개선사항

- 보안을 강화하여 웹브라우저 환경에서 더욱 안전하게 사용할 수 있는 웹서비스 구현
- 디자인 개선 및 캐릭터 추가로 서비스를 더욱 브랜드화
- 사용자 입장에서 빌리마켓을 더욱 편리하게 사용할 수 있는 기능들 추가
- 배너 광고, 제휴 광고 페이지, 구독 기능을 추가해 수익을 창출할 수 있는 수단들을 다양화

목표

웹, iOS, AOS 어느 플랫폼에서나 이용가능한 빌리마켓 구현