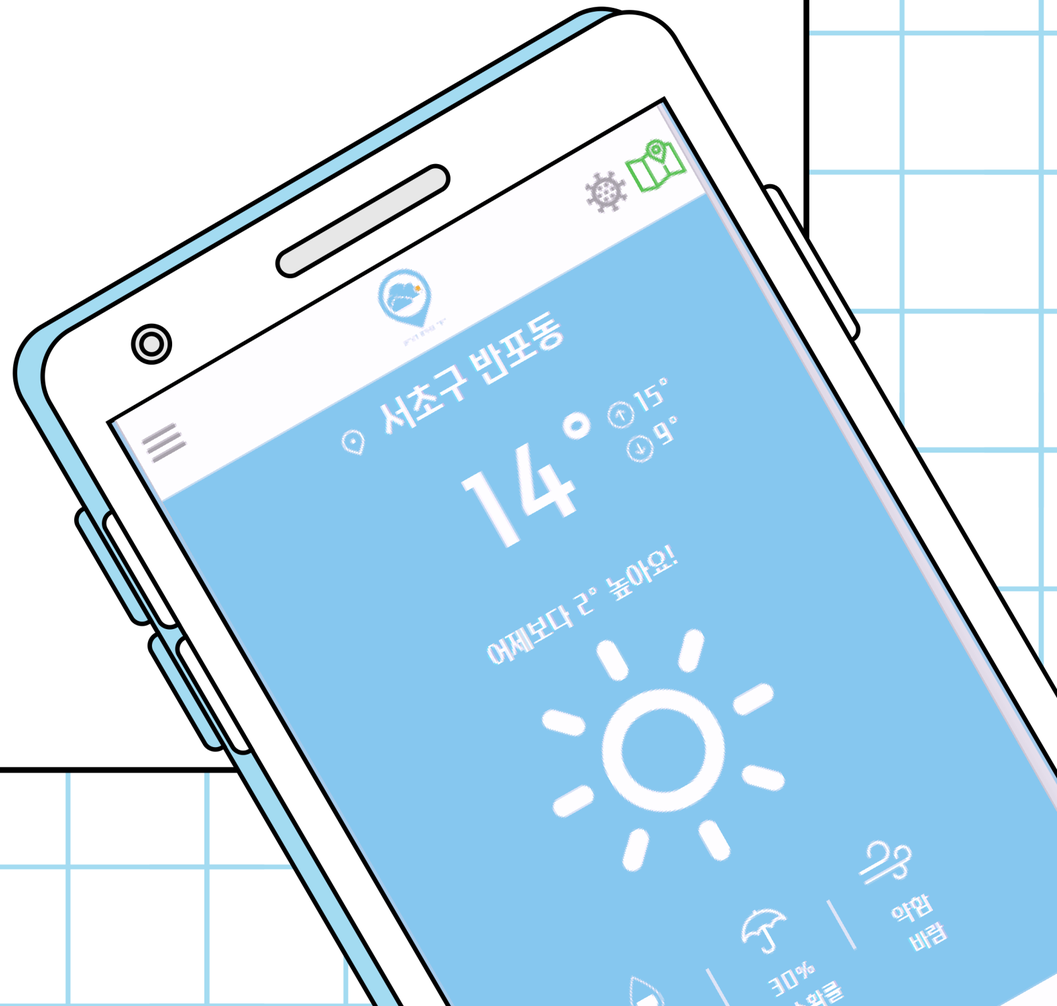
Kakao cloud school 2기

WEATHER WHERE

날씨와 대기정보를 기반으로 한 관광지 추천 서비스

웨하스

https://m.weatherwhere.link/



목차

1. 서비스 기획

2. 서비스 소개

3. 소프트웨어 개발

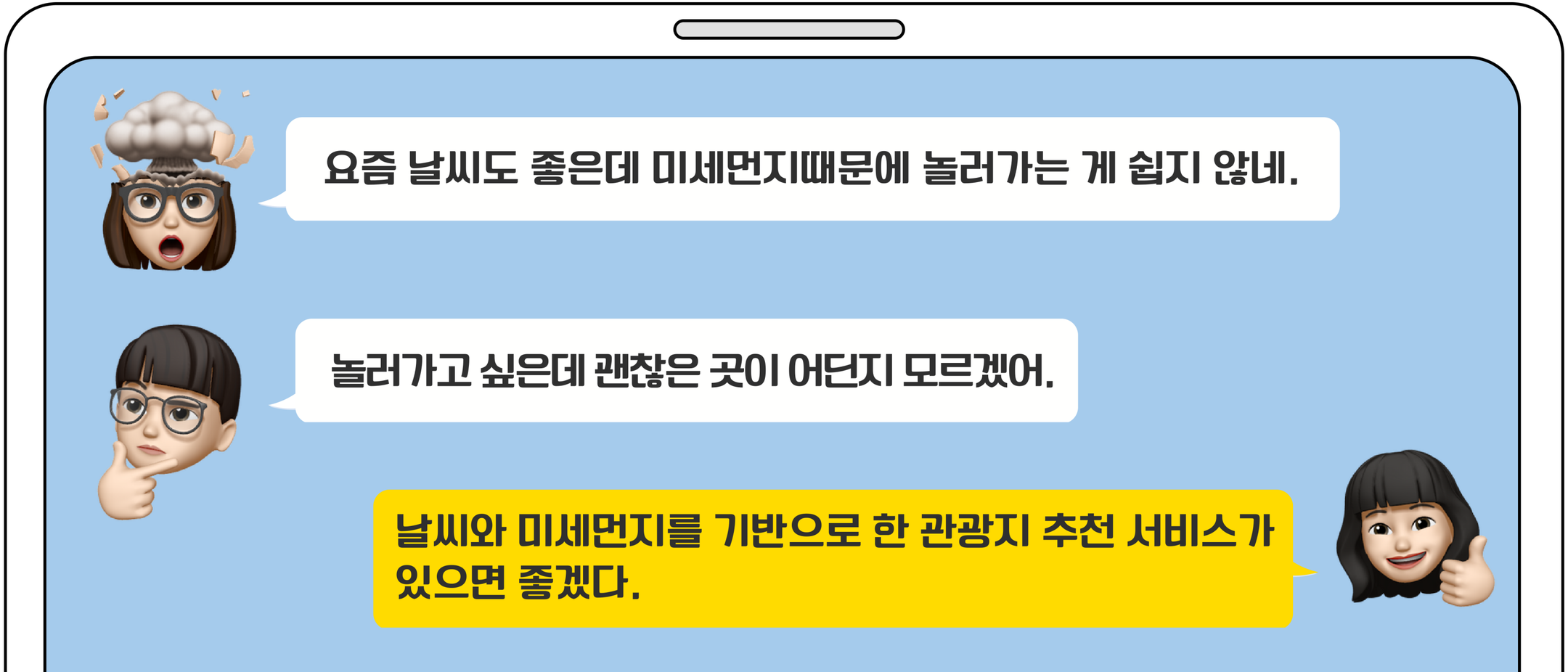
4. 기술적 고민과 해결

5. 프로젝트 관리

1. 서비스 기획

1.1 문제 정의

날씨와 **미세먼지**를 기반으로 한 **관광지 추천** 서비스가 있으면 어떨까?



1. 서비스 기획

1.2 기획 발전 방향

공공데이터포탈에서
날씨와 대기 정보를 수집한 뒤 관광 랭킹 지수로 변환

이를 기반으로 오늘의 추천 관광지를 제공하는 서비스

1. 서비스 기획

1.3 요구사항 정의

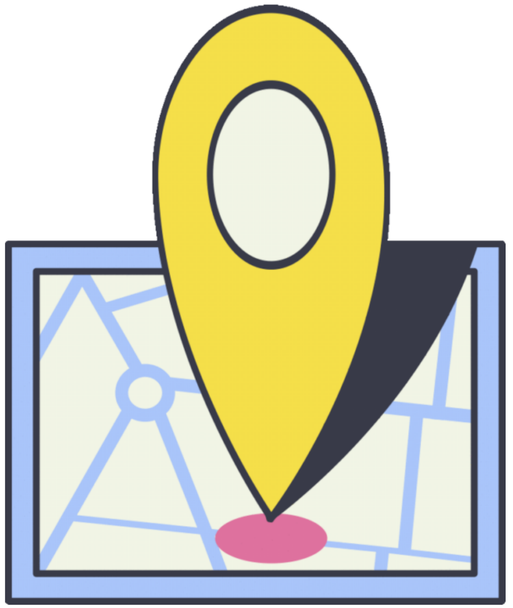
날씨

하루 · 주간 예보 조회

날씨 상세보기

대기

실시간 · 주간 예보 조회



관광

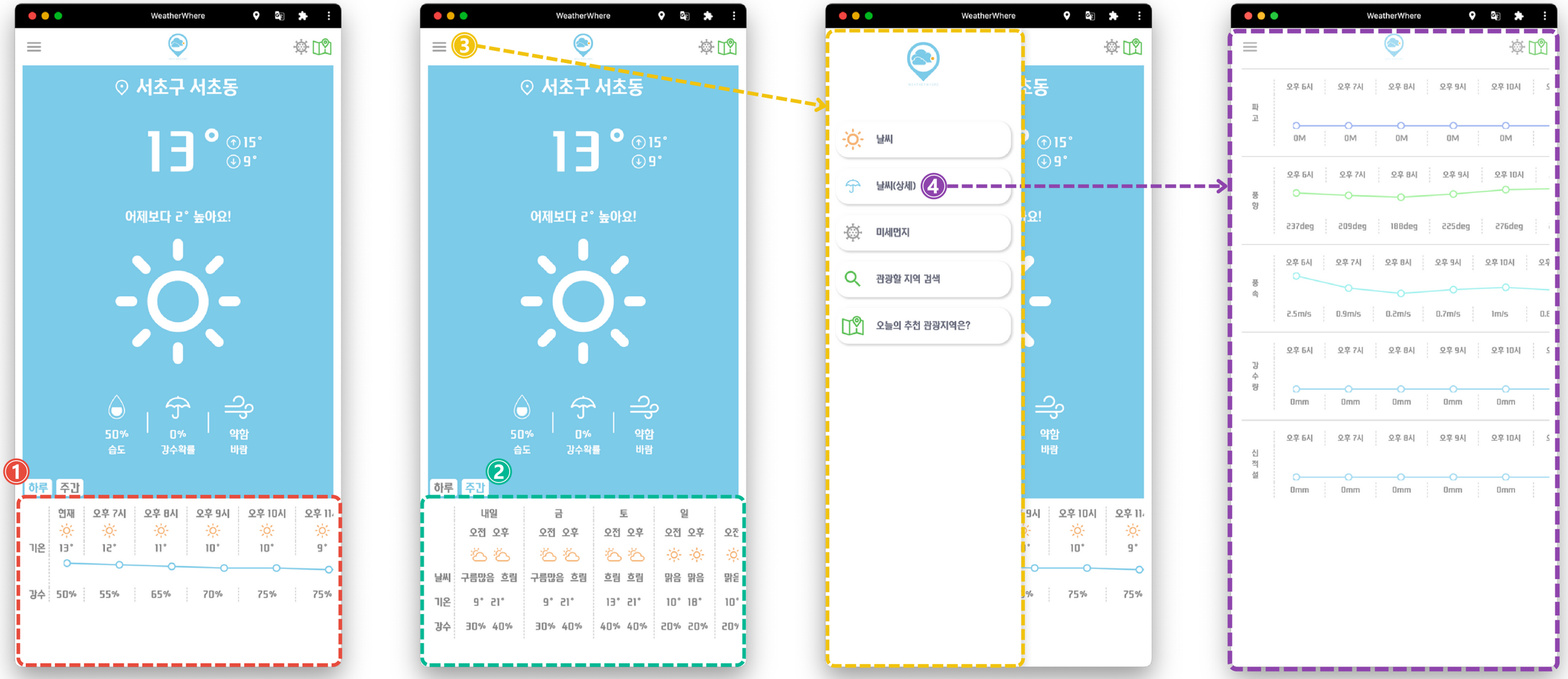
추천 · 검색 지역의

관광지 조회

관광지의 정보 조회

2. 서비스 소개

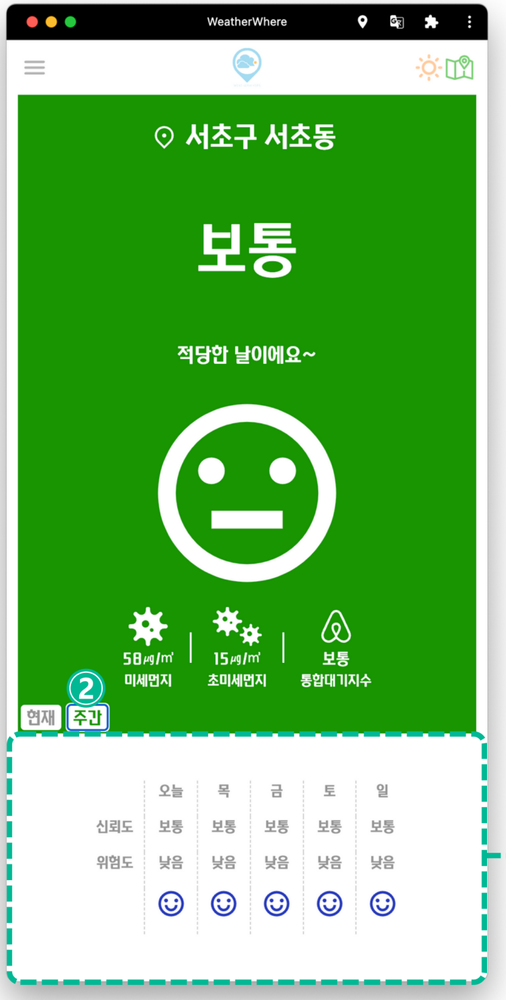
2.1 날씨 서비스 기능 소개



2. 서비스 소개

2.2 대기 서비스 기능 소개

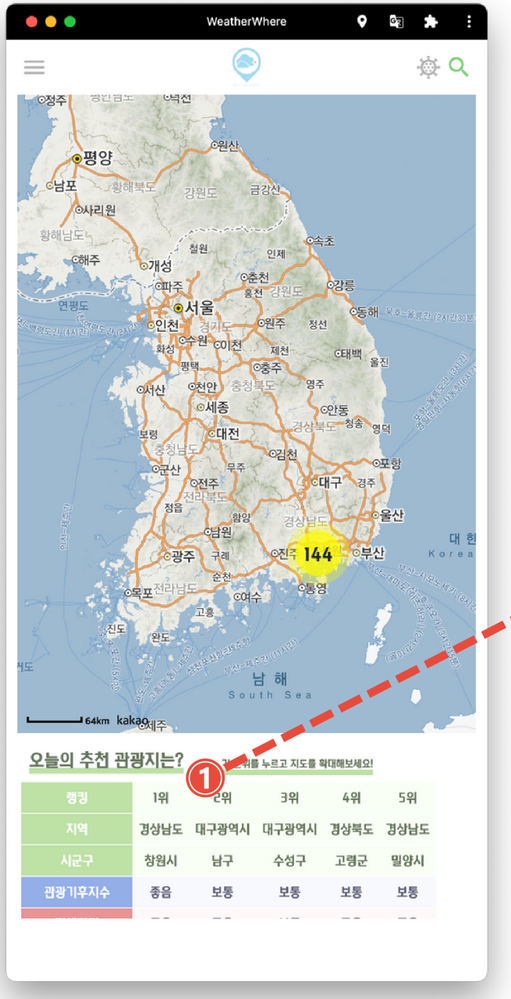
실시간 대기 정보



주간 대기 정보

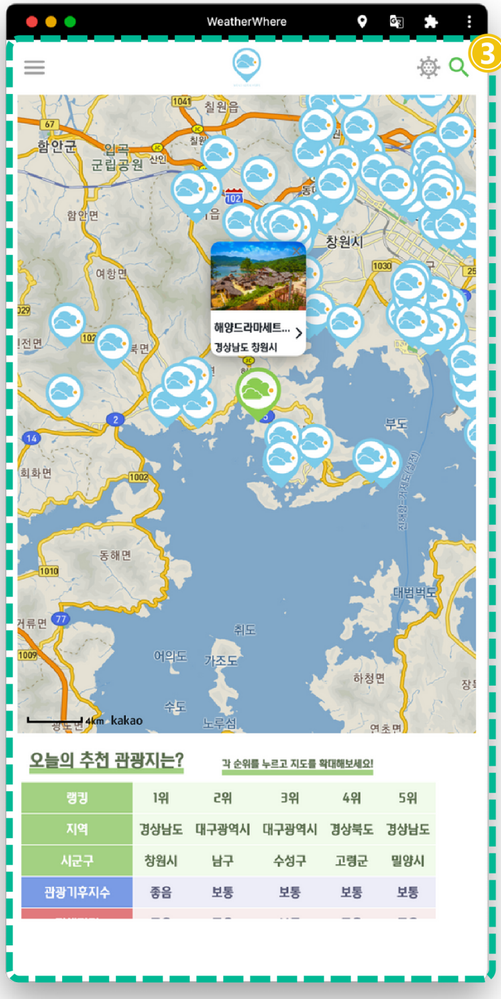
2. 서비스 소개

2.3 관광지 서비스 기능 소개

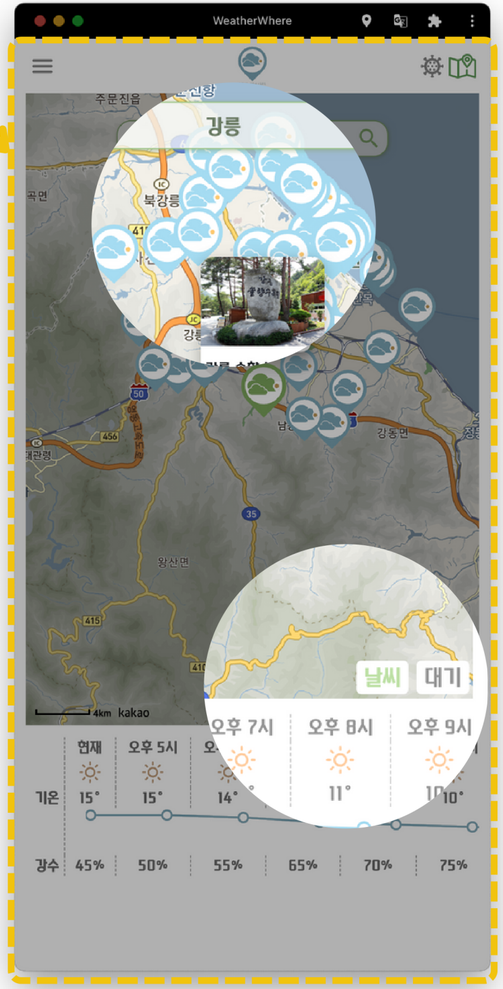


추천 관광지 클릭

추천 관광지의 관광지의 수를
클러스터 형식으로 보여줌



클러스터를 확대, 클릭 하면
관광지를 보여줌



검색 기능을 통해 검색 지역의
관광지와 날씨, 대기 정보 조회 가능

2. 서비스 소개

2.3 관광지 서비스 기능 소개

관광지의 공통 정보

(우편번호, 주소, 개요)

관광지의 상세 날씨, 대기 정보

관광지의 소개 정보

(관광지의 주차장,

반려동물 동반 입장 등등의

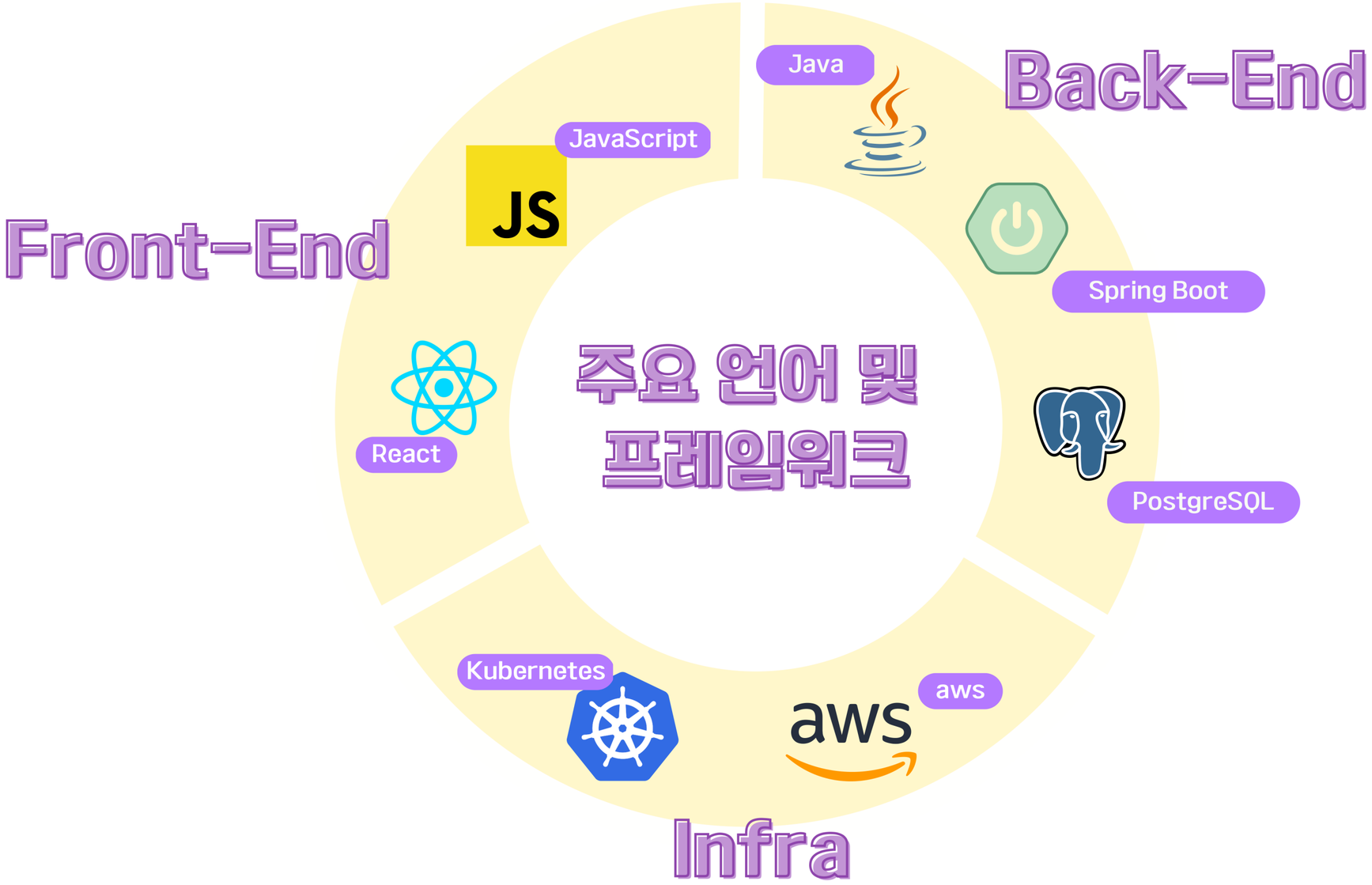
이용 정보)

2. 서비스 소개

2.4 웹 앱 시연

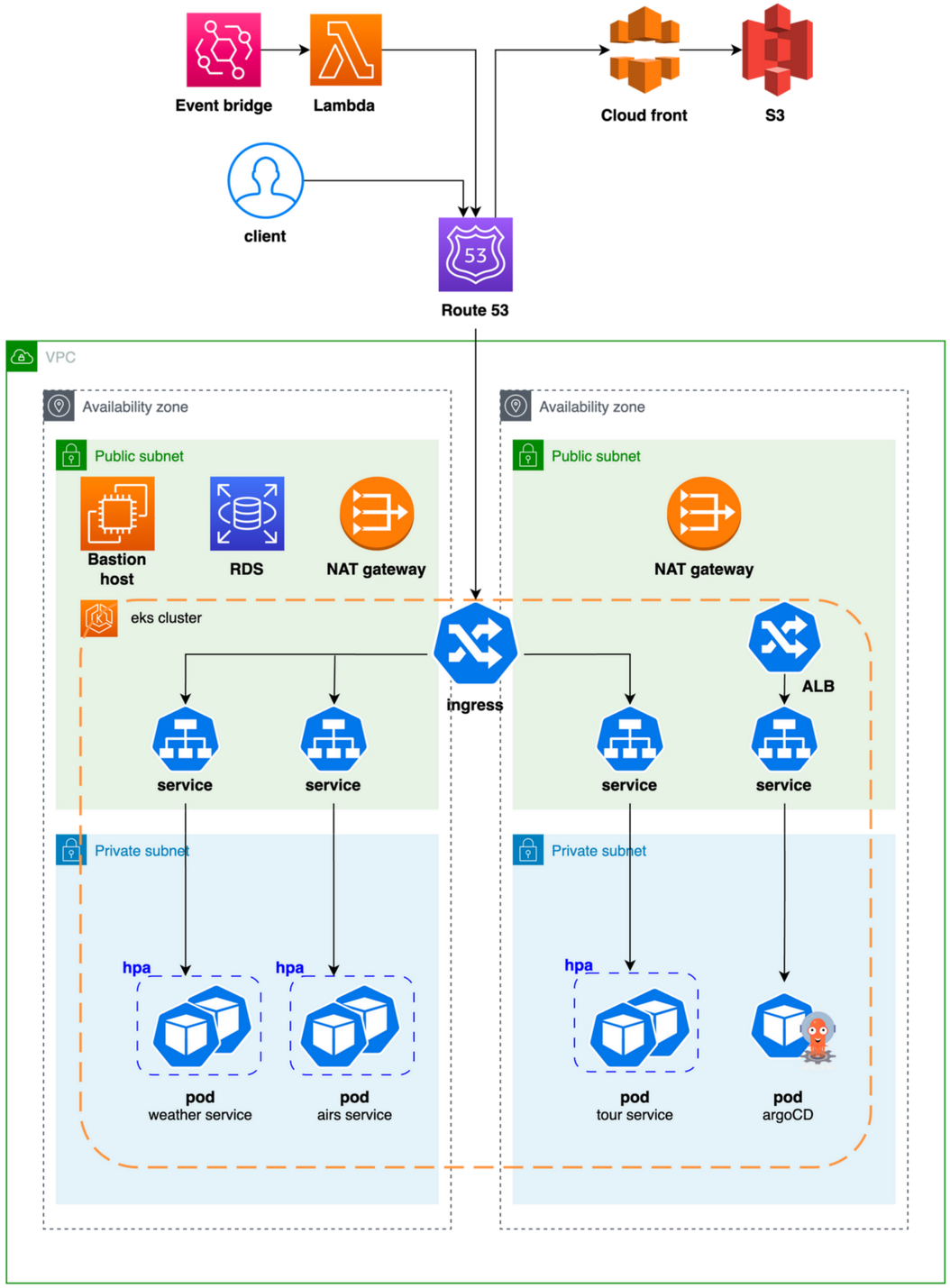
3. 소프트웨어 개발

3.1 기술 스택



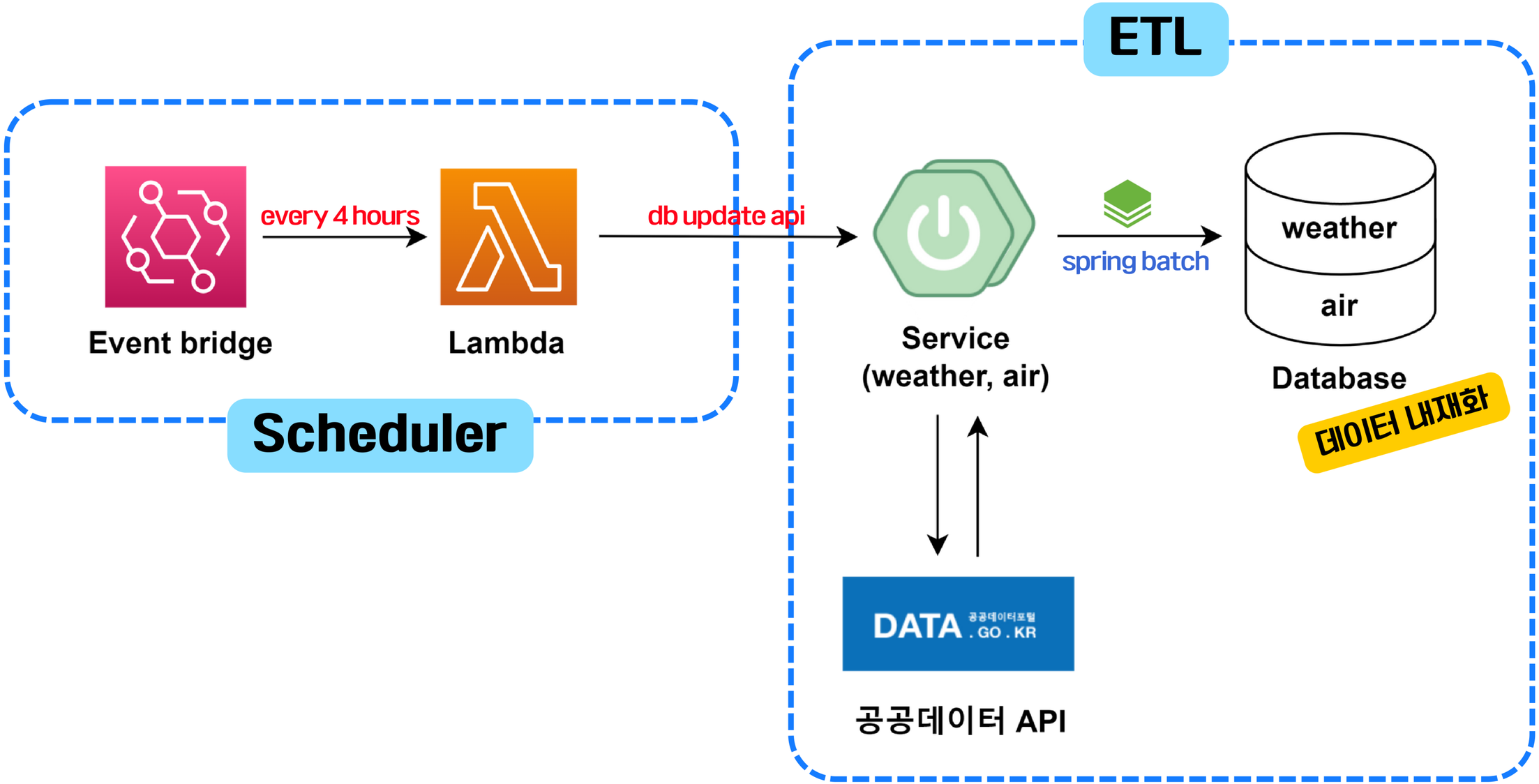
3. 소프트웨어 개발

3.2 소프트웨어 아키텍처



3. 소프트웨어 개발

3.3 데이터 파이프라인 구축(날씨, 대기 원천데이터 ETL)



3. 소프트웨어 개발

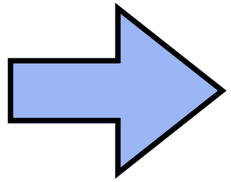
3.4 관광 랭킹 지수 알고리즘

Recommend Logic v1

Min-Max 정규화
날씨 & 미세먼지

✕ 가중치

Tour Rank Value



Recommend Logic v2

TCI 지수 계산
날씨

농도 범위 지정
미세먼지

Tour Rank Value

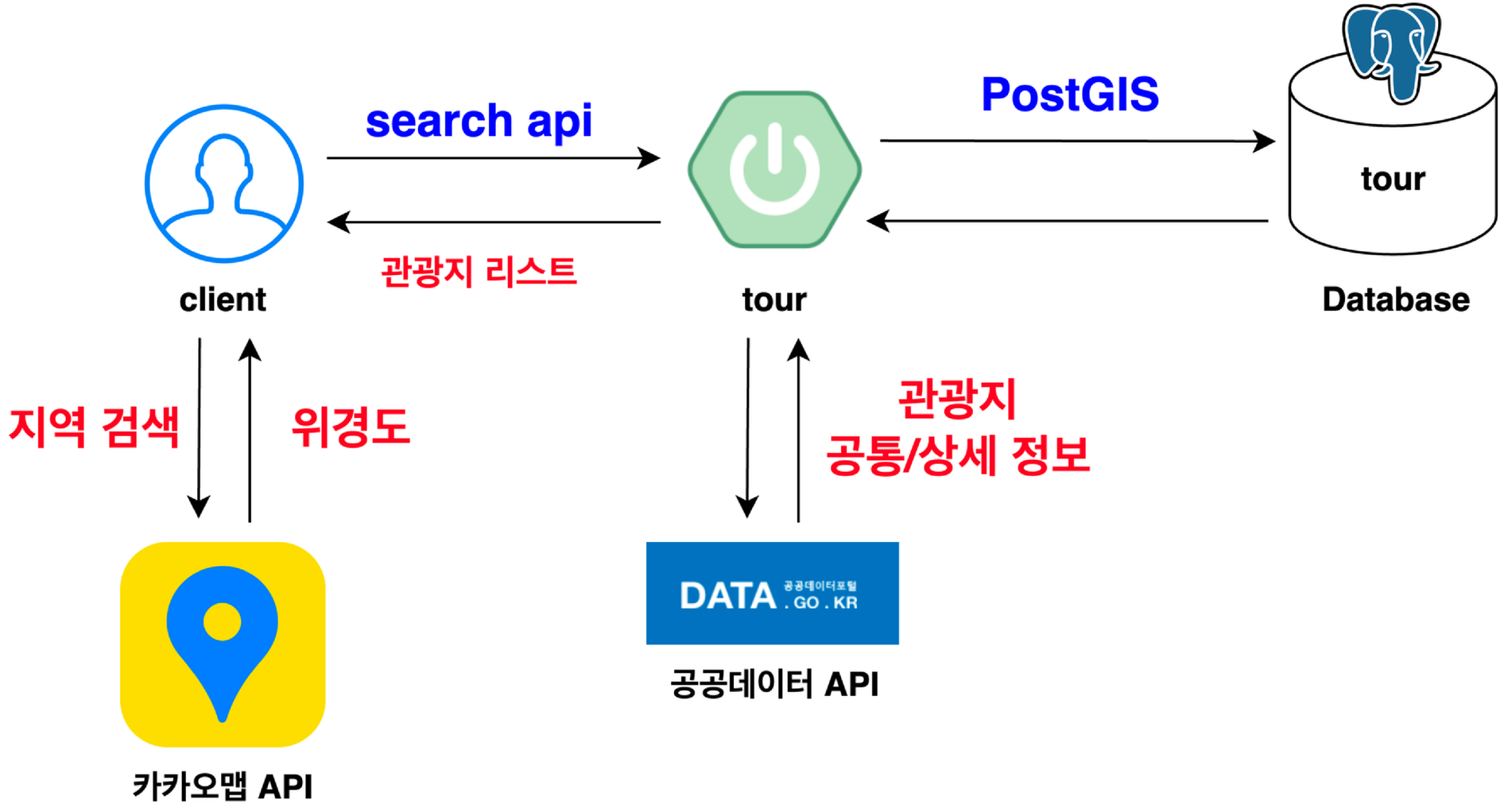
적재된 날씨, 대기 데이터로 관광 랭킹 지수 계산

관광기후지수(TCI)란? 관광에 적합한 기후상황을 지수화한 값.

3. 소프트웨어 개발

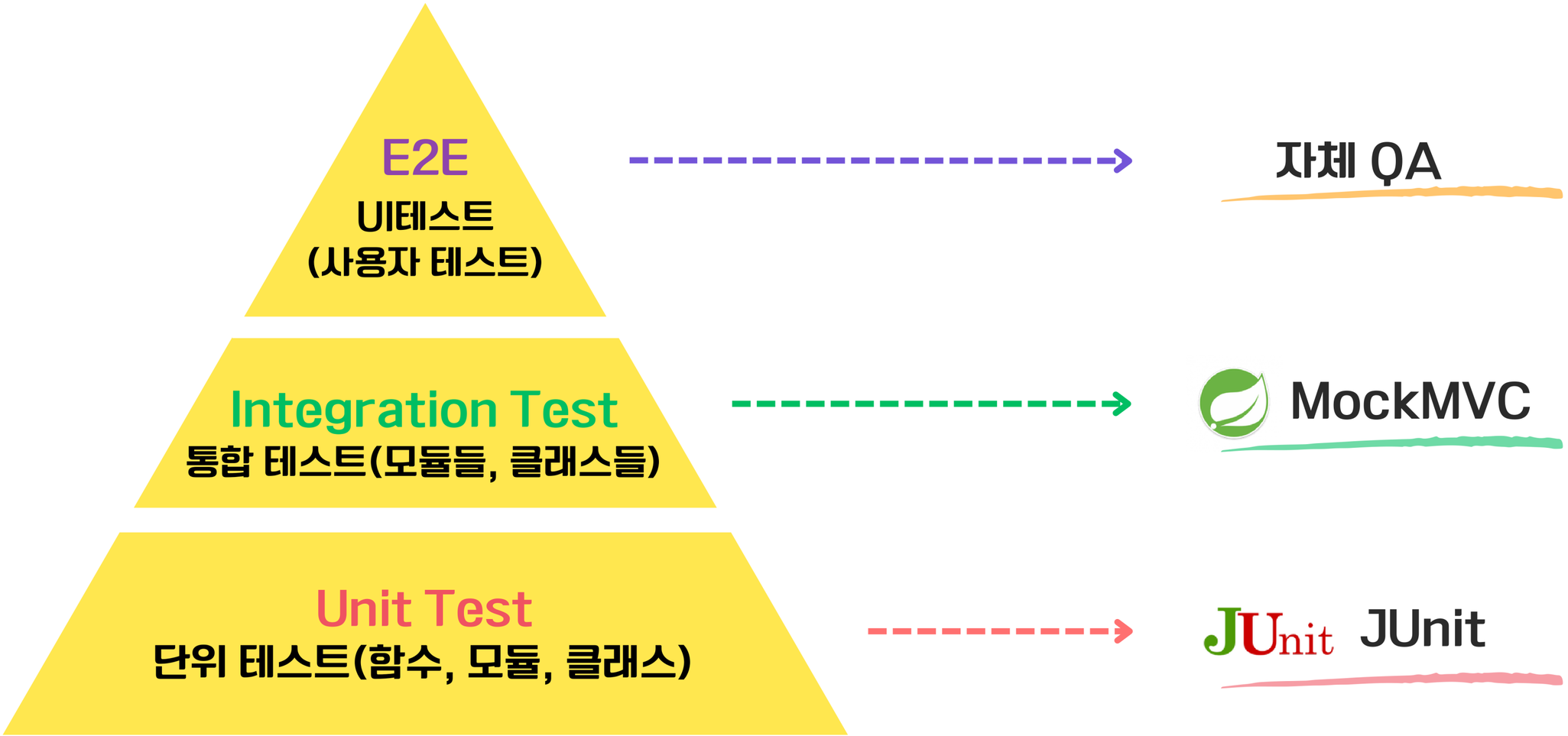
3.5 관광 검색 알고리즘

PostGIS 플러그인을 사용하여
검색 지역 위경도 반경 10km 관광지 조회



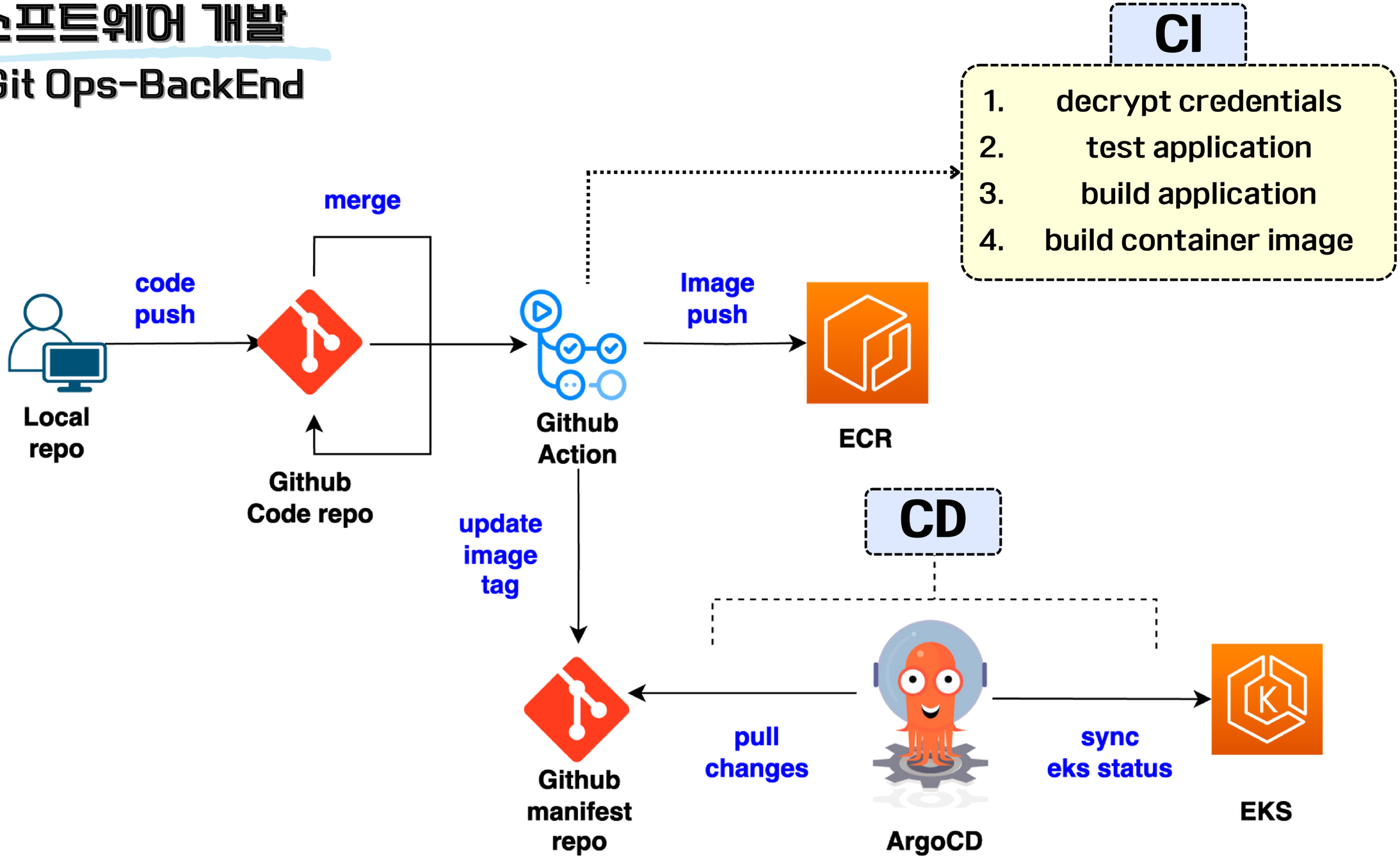
3. 소프트웨어 개발

3.6 소프트웨어 테스팅



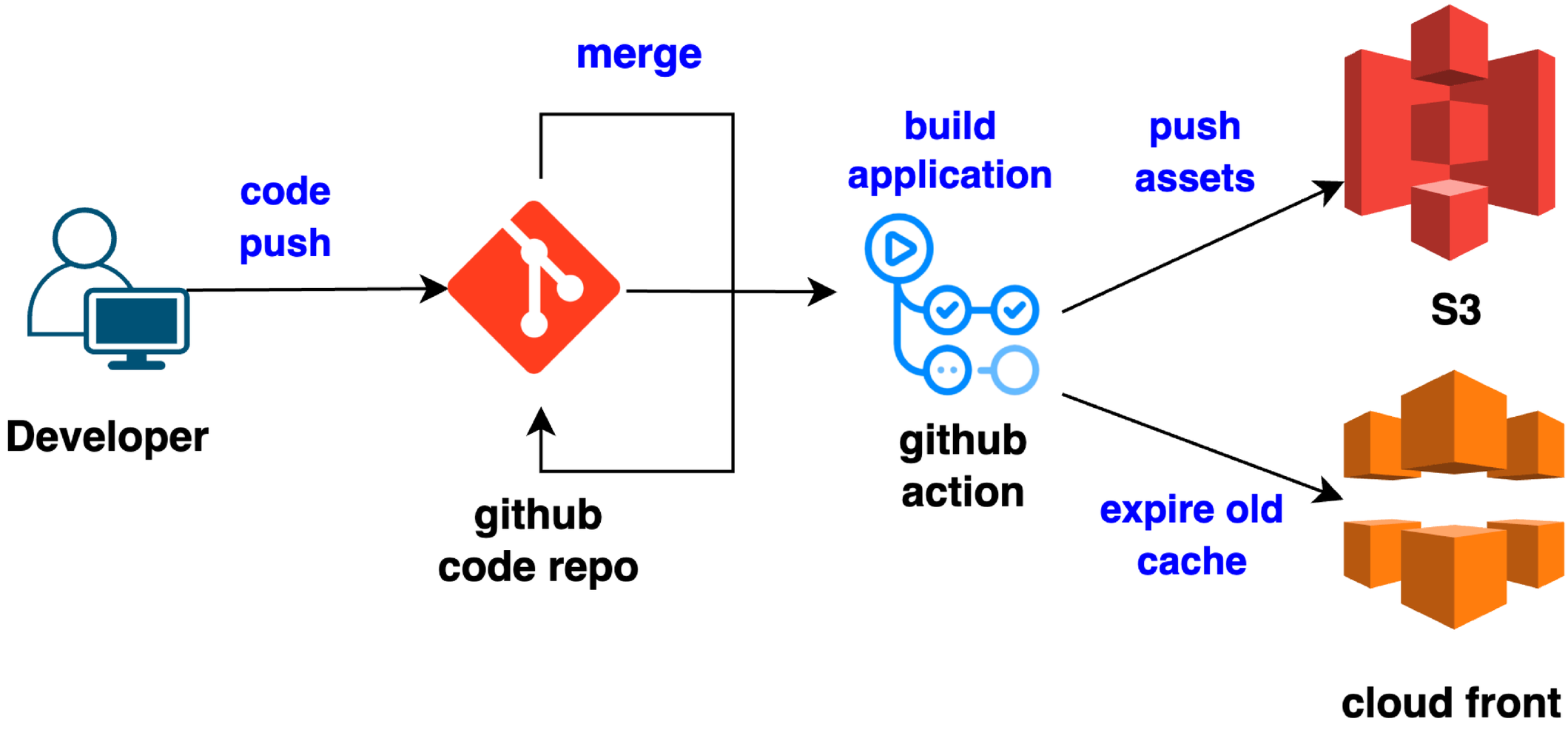
3. 소프트웨어 개발

3.7 Git Ops-BackEnd



3. 소프트웨어 개발

3.8 Dev Ops-FrontEnd



4. 기술적 고민과 해결

4.1 DB 기술 선택에 대한 고민

문제점	해결	
운영 로드 때문에 DB는 일원화해서 사용하지만, 추후 **서비스 별로 분리될 수 있는 구조**가 필요함	**PostgreSQL**	논리적으로 **스키마를 분리**하여 운영함
공간 정보 연산 및 일반 정보를 동시에 다뤄야 하는 비즈니스 요구사항		공간 DB에 특화된 플러그인, **PostGIS** 사용
운영 로드를 줄일 수 있는 부분에서는 최대한 줄이고 싶음	**RDS**	**매니지드 서비스** 사용

4. 기술적 고민과 해결

4.2 운영 및 배포 전략에 대한 고민

문제점	해결	
개발 – 테스트 – 운영 환경이 달라서 개발 단계에 없었던 버그가 테스트 및 운영 단계에서 발견됨	Docker	container 기술을 활용해서 개발–테스트–운영 환경을 일원화
매뉴얼 배포 시 업무 부하가 커지고 휴먼 에러 가능성이 증가함	GitOps	GitOps 도입
배포 시 순단이 일어남	Kubernetes rolling update	무중단 배포

4. 기술적 고민과 해결

4.3 보안 이슈에 대한 고민

문제점	해결	
민감 정보를 평문으로 형상 관리하게 되면 보안 이슈가 발생	Git-Secret	git-secret 및 gpg를 도입 ↓ credential을 암호화하여 형상 관리할 수 있게 됨
• 외부에서 k8s 컨트롤 플레인에 접근할 필요성이 있음 • k8s api를 public으로 노출할 경우 보안 이슈가 발생함	Bastion Host	• k8s 자원들은 private에 구성 • 동일한 VPC의 public 서브넷에 Bastion Host를 배치하여 클러스터 내부에 접근함

4. 기술적 고민과 해결

4.4 모바일 대응에 대한 고민

문제점	해결	
• 초기 개발 단계에서 데스크탑 대응만 되어 있음 • 모바일 환경에 서비스를 제공해야 더 많은 사용자에게 효용을 줄 수 있음	**모바일 웹 브라우저 대응**	모바일 퍼스트 도입하여 모바일일 경우에 최우선으로 초점을 맞춰 디자인
서비스가 크로스 플랫폼에서 연결되는 UX를 중시하기 보다는 유틸리티성이 크기 때문에 다양한 채널로 제공하는 게 더 많은 고객에게 효용을 줄 수 있음	**모바일 네이티브 환경 대응**	PWA를 적용하여 더 부드러운 사용자 경험을 제공함

5. 프로젝트 관리

5.1 협업 및 일정관리

대기 서비스 : 박지우, 박상준
날씨 서비스 : 김규민, 백영서

관광 서비스 : 박지우, 박상준
프론트 개발 : 김규민, 백영서

인프라 : 황주빈

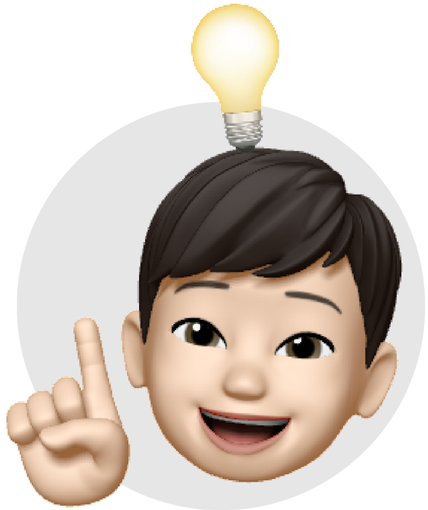
5. 프로젝트 관리

5.3 아쉬운 점 및 차후 개선 방향

아쉬운 점 및 차후 개선 방향

아쉬운 점	차후 개선 방향
인프라 구성에 대한 버전관리가 되지 않음	프로비저닝 도구 도입(Terraform)
Logging 및 Monitoring을 위한 도구 활용이 부족	ELK 도입
현재는 관광 랭킹 지수를 PostgreSQL에서 order by를 사용하여 계산하지만, 이는 불필요한 부하를 일으킬 수 있음	Tour 서비스에서 Redis Sorted Set 활용

팀원 소개



김규민

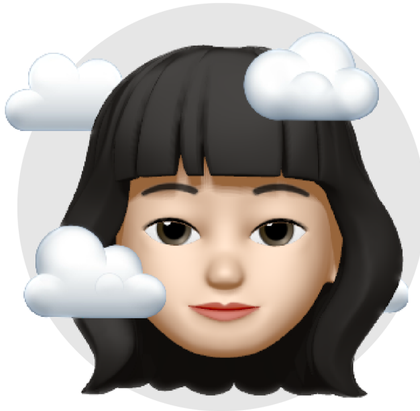
BackEnd
FrontEnd

박상준

BackEnd

박지우

BackEnd



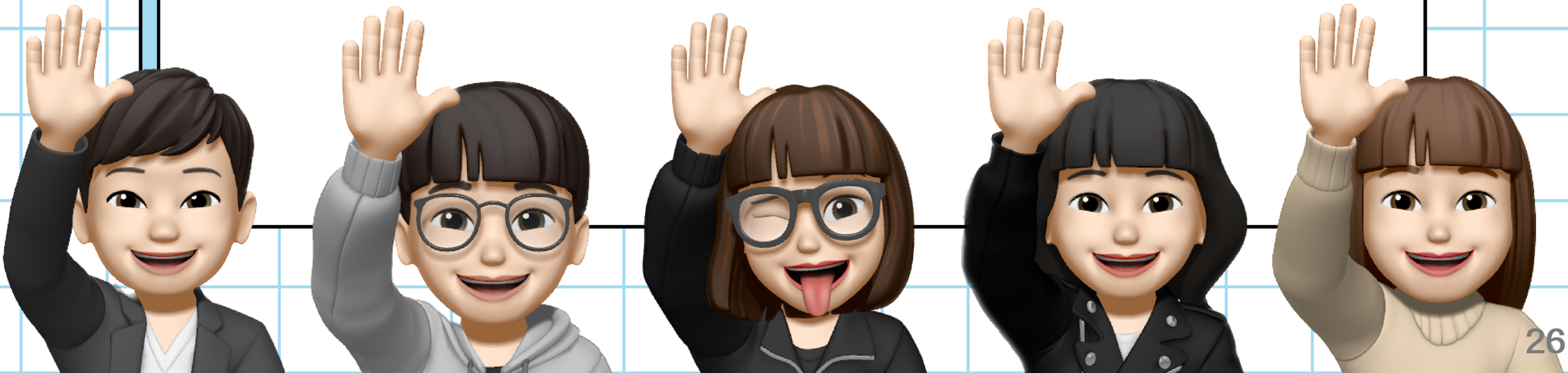
백영서

BackEnd
FrontEnd
Design

황주빈

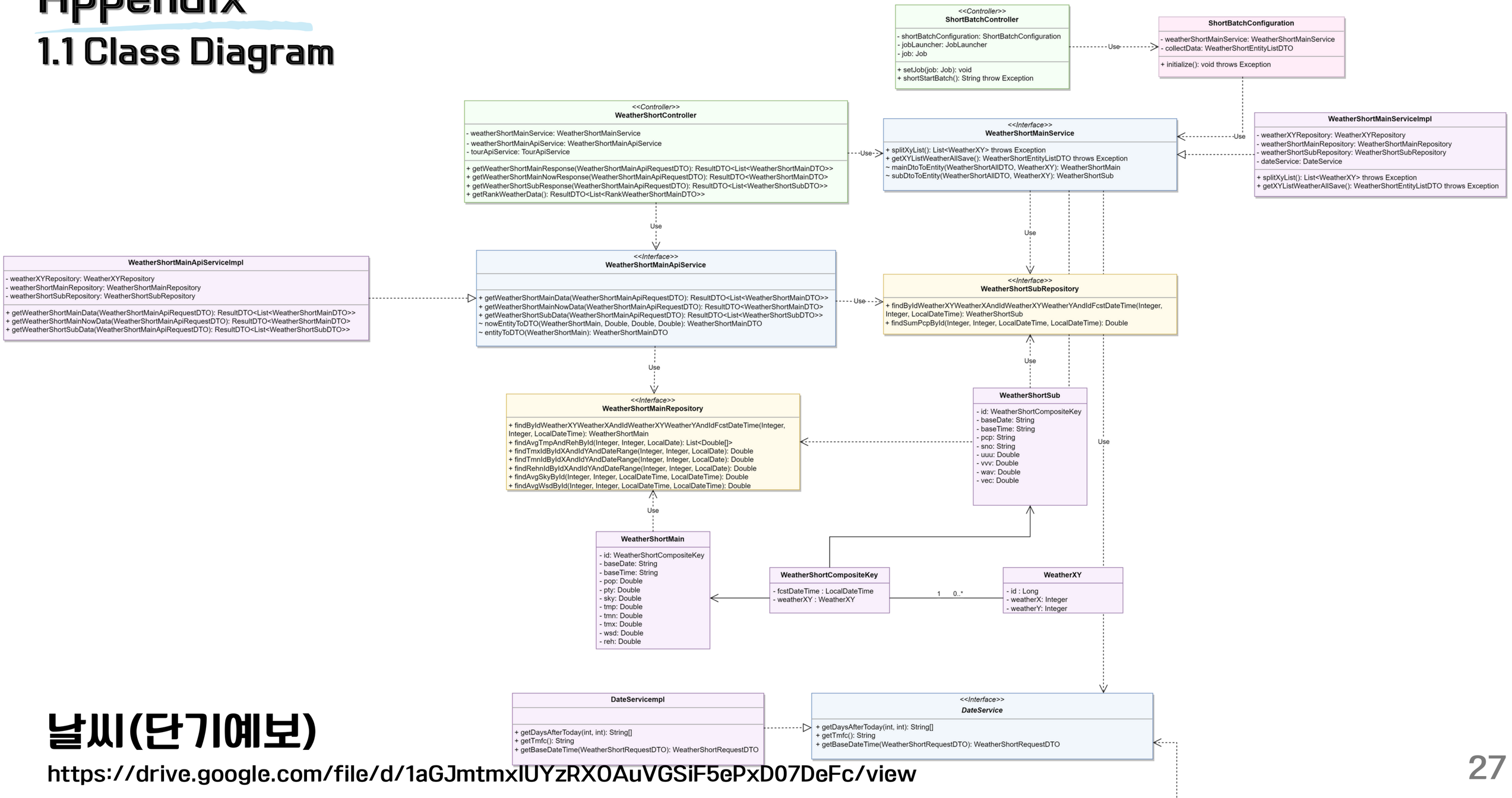
Infra

EOD



Appendix

1.1 Class Diagram

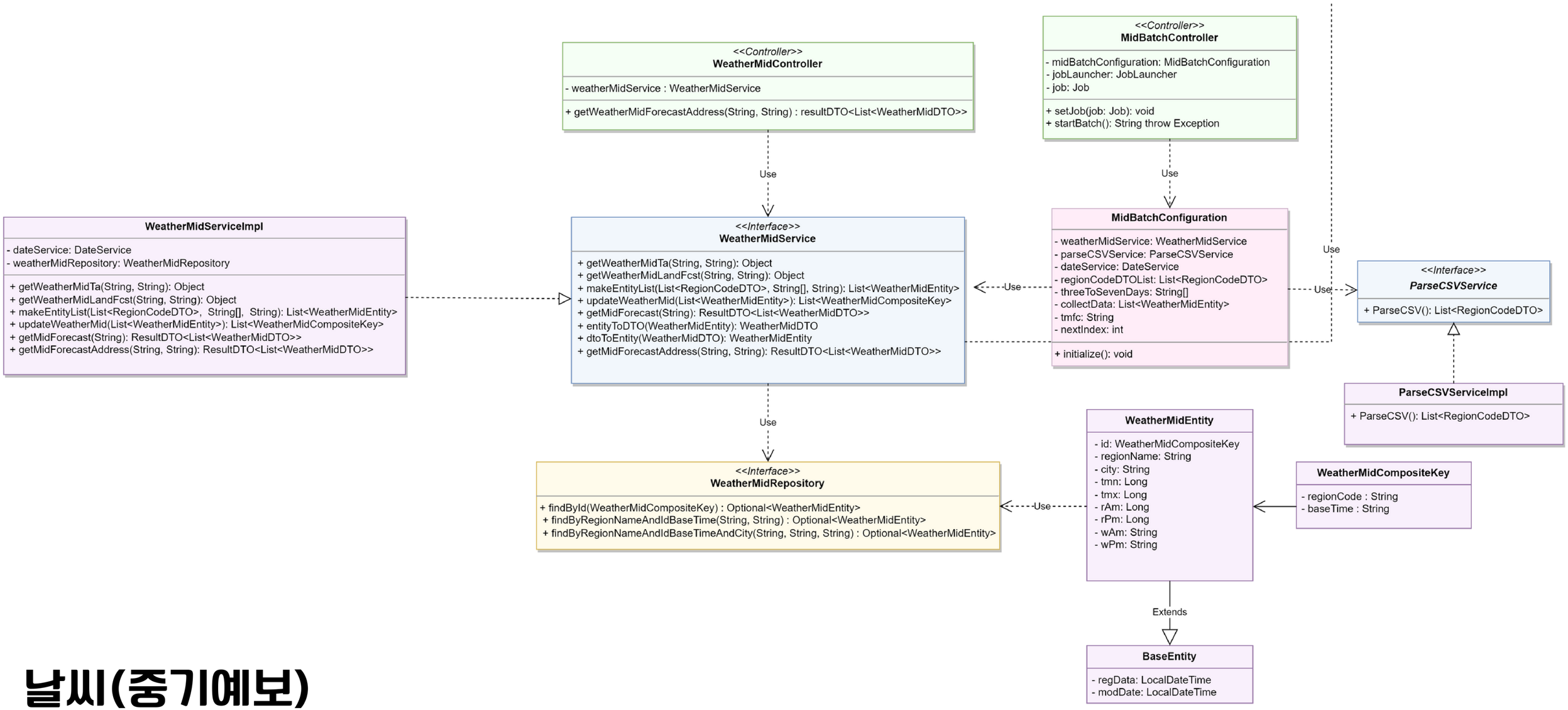


날씨(단기예보)

https://drive.google.com/file/d/1aGJmtmxlUYzRXOAuVGSiF5ePxD07DeFc/view

Appendix

1.1 Class Diagram

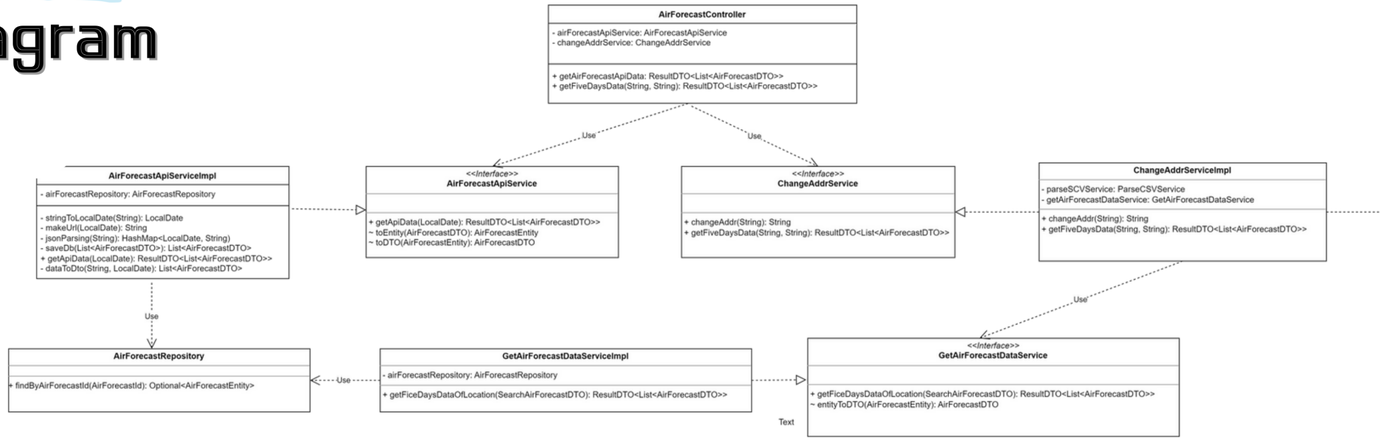


날씨(중기예보)

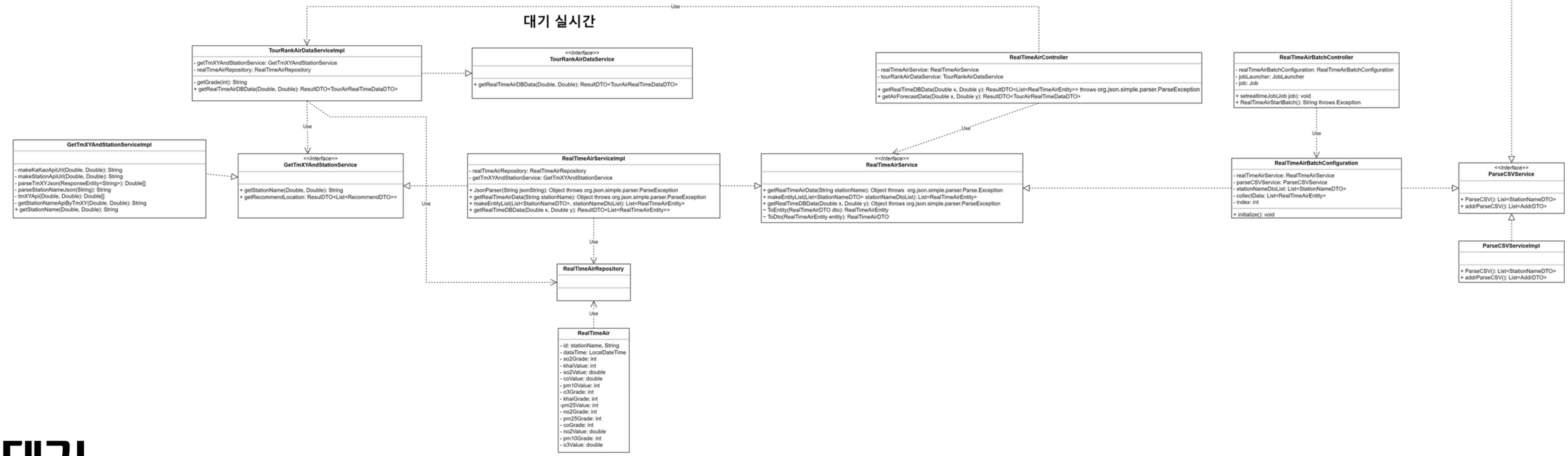
Appendix

1.1 Class Diagram

대기 주간예보



대기 실시간



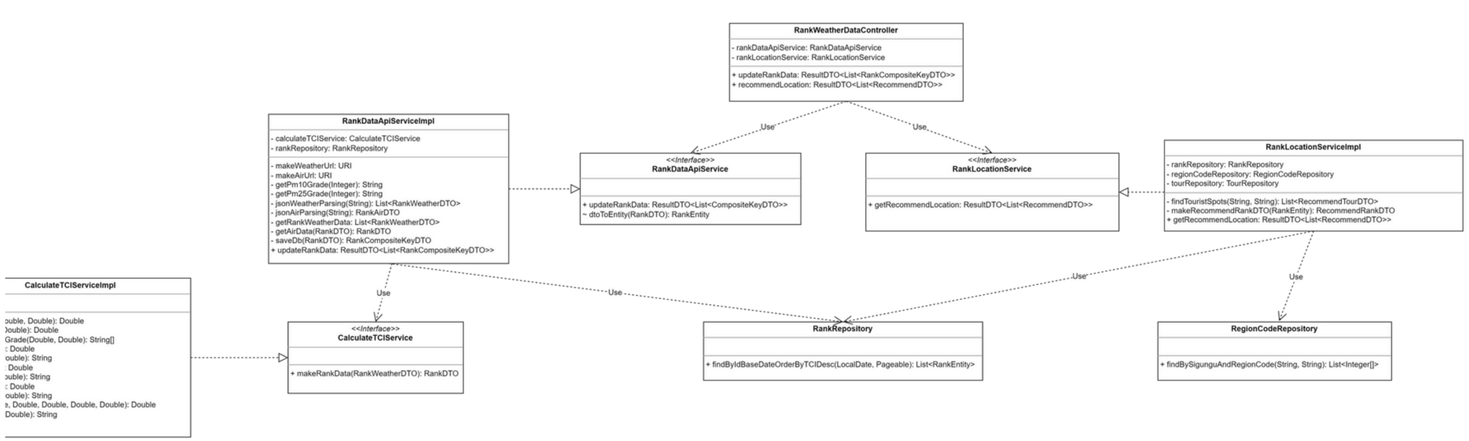
대기

https://drive.google.com/file/d/1DnFl0Hms74qZQo9HtxRjKUAjwP-_OfHH/view?usp=sharing

Appendix

1.1 Class Diagram

Tour Service



관광

https://drive.google.com/file/d/1DnFI0Hms74qZQo9HtxRjKUAjwP-_OfHH/view?usp=sharing