

여행 프로젝트 진행기 (1)

여러 API 지원, 캐싱 적용기

여러 API 프로토콜 지원

여러 API 프로토콜 지원

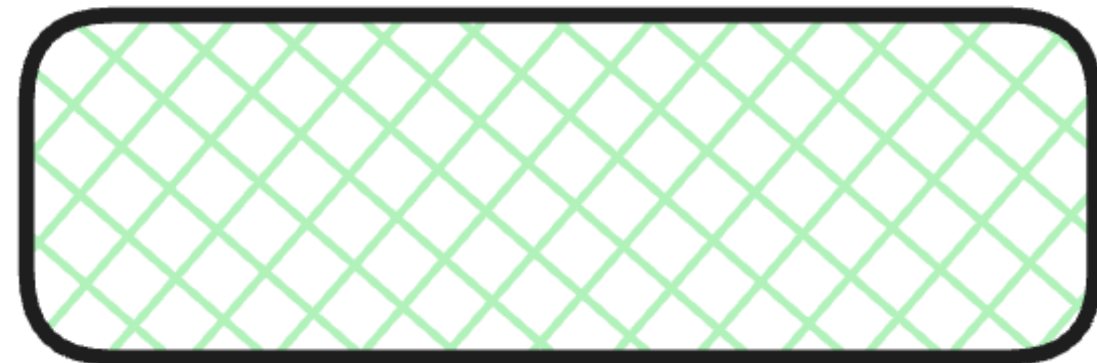
마이크로 서비스간 호출
(미구현)

REST

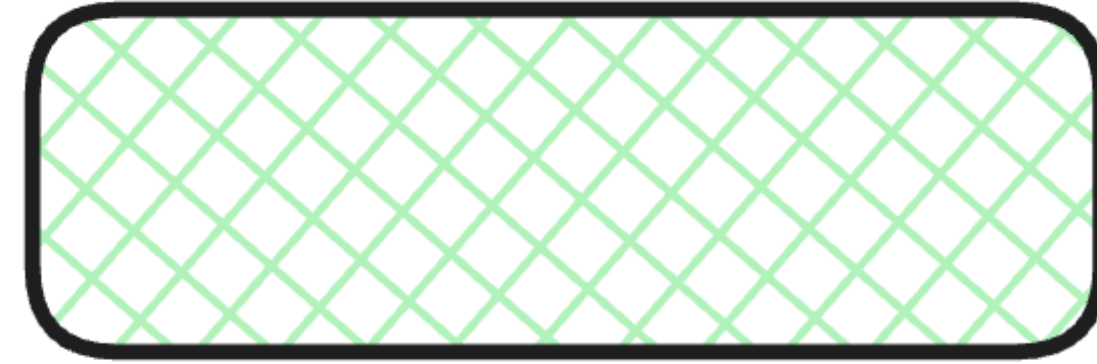
GRPC

GraphQL

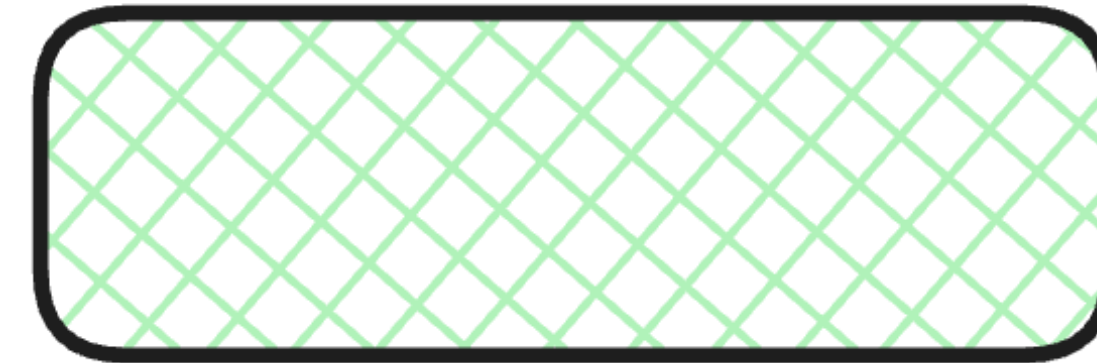
Controller



Service

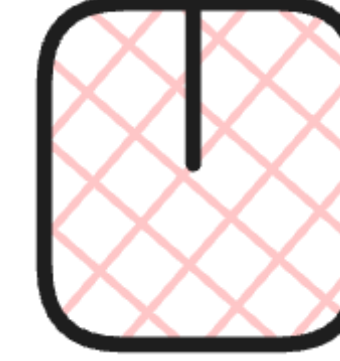
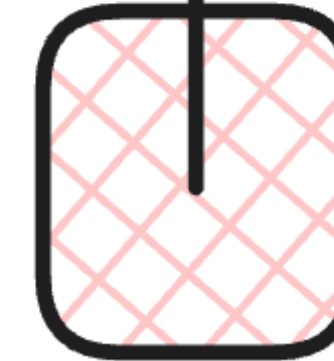


Data Access

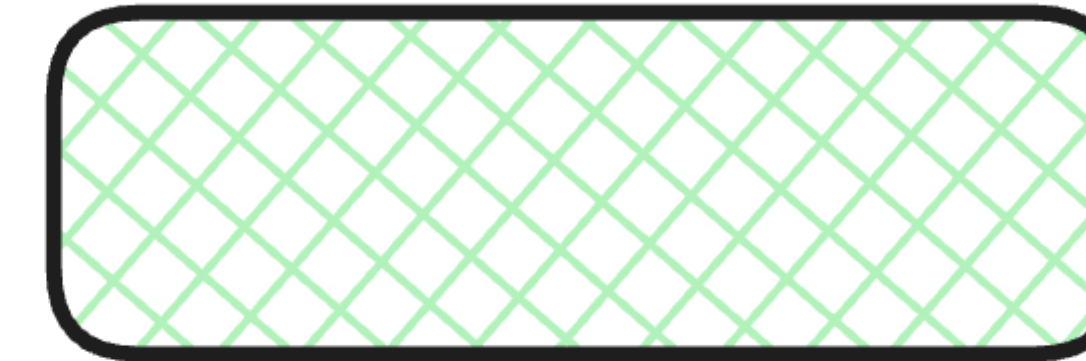


Database

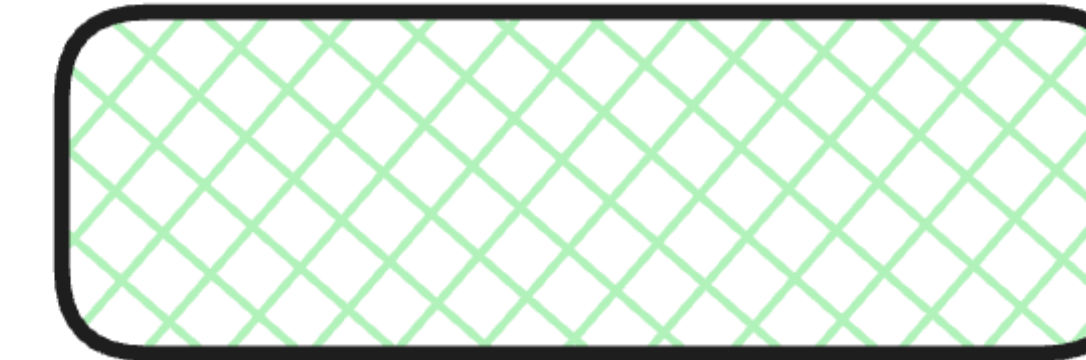
Controller



Service



Data Access



Database

API 노출, 인증 인가 등

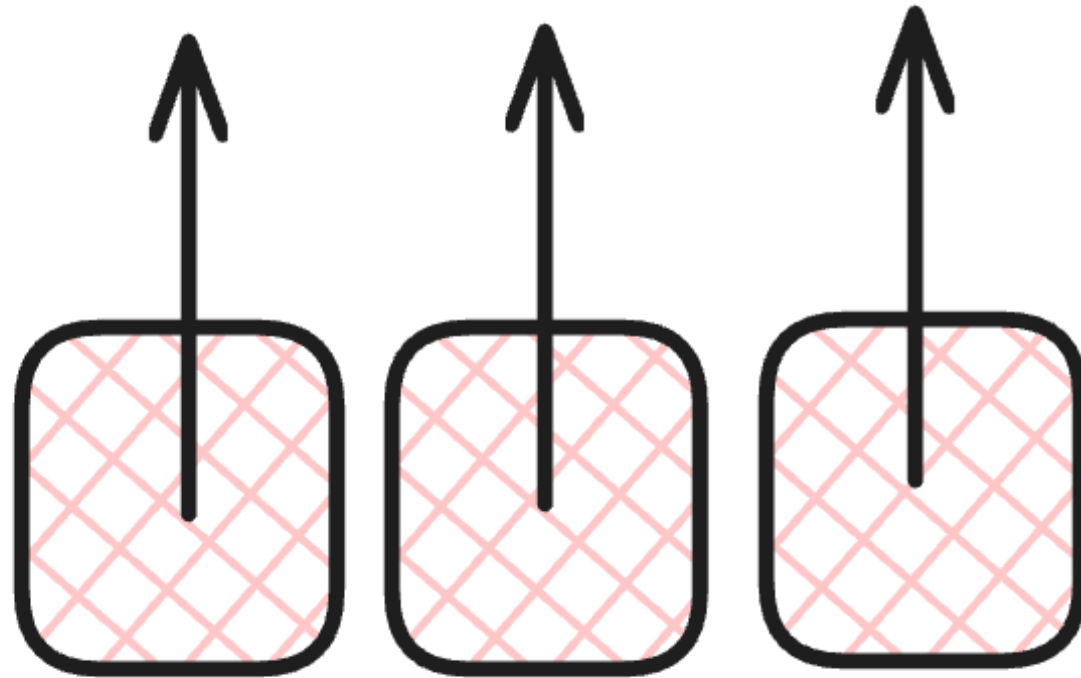
트랜잭션, 핵심 로직

여러 API 프로토콜 지원

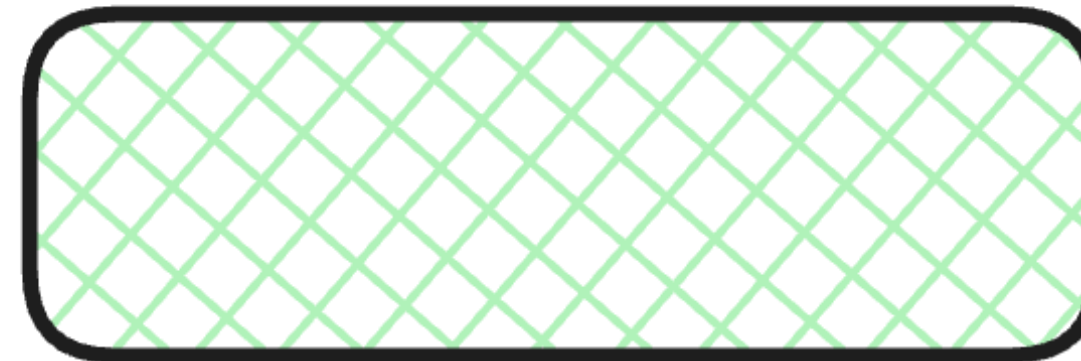
마이크로 서비스간 호출
(미구현)

REST GRPC GraphQL

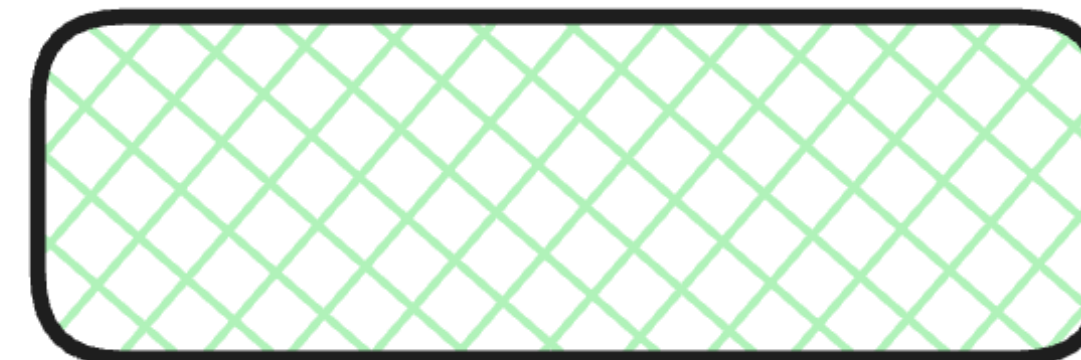
Controller



Service



Data Access



Database

API 노출, 인증 인가 등

트랜잭션, 핵심 로직

```
7  
6 usages 1 implementation Onji Kim  
8 public interface PlaceService {  
9  
10     /**  
11      * 장소를 조회합니다.  
12      *  
13      * @param placeId 장소 id, not null  
14      * @return 조회된 장소  
15      * @throws PlaceRetrieveFailedException 장소 조회에 실패한 경우  
16      * @throws click.porito.place_service.external_api.ExternalApiException 외부  
17      */  
18     PlaceDto getPlace(String placeId) throws PlaceRetrieveFailedException;  
19 }
```

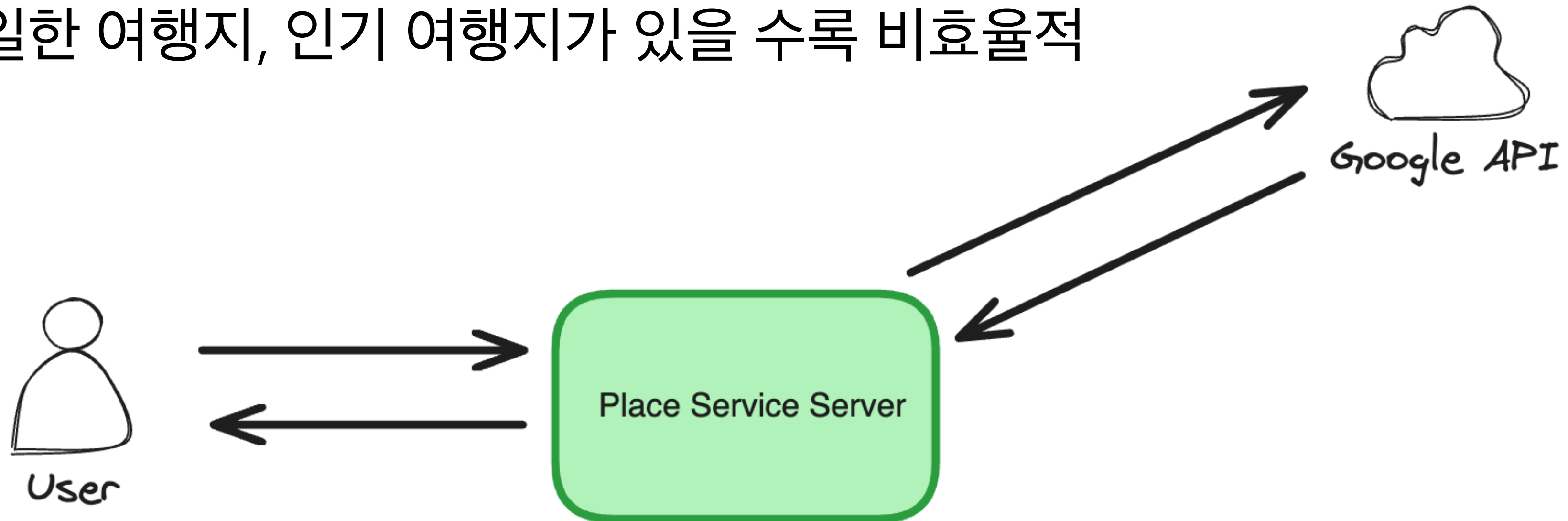
```
Onji Kim  
@GetMapping("/{placeId}")  
public PlaceDto getPlace(@PathVariable String placeId) {  
    return placeService.getPlace(placeId);  
}
```

```
Onji Kim  
@QueryMapping(name = "place")  
public PlaceDto place(@Argument("id") String id) {  
    return placeService.getPlace(id);  
}
```

여행 정보 캐싱 기능 추가

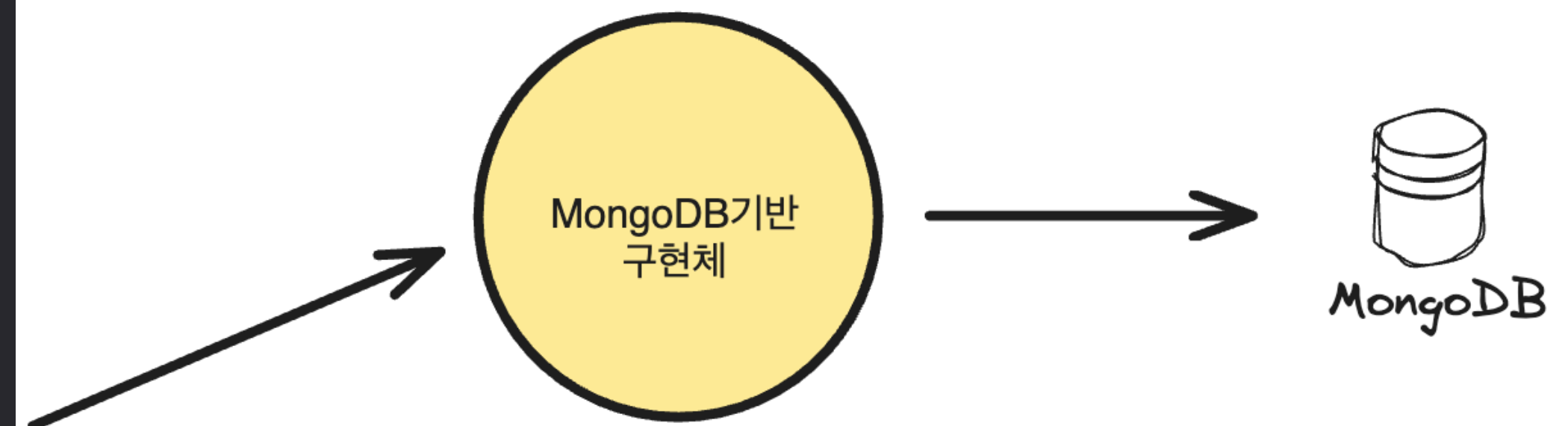
문제점

동일한 여행지, 인기 여행지가 있을 수록 비효율적



캐싱 기능 추가

```
public interface PlaceCacheService {  
    1 usage 1 implementation 👤 Onji Kim  
    Optional<PlaceDto> get(String placeId);  
    1 usage 1 implementation 👤 Onji Kim  
    void put(PlaceDto placeDto);  
    1 usage 1 implementation 👤 Onji Kim  
    void putAll(List<PlaceDto> placeDtos);  
}
```



스키마 없음

빠름

확장 용이

수평적 확장에 대한 강력한 지원

트랜잭션, 정합성 중요하지 않음

캐싱 기능 추가

```
1 usage  👤 Onji Kim
@Data
@Component
@ConfigurationProperties(prefix = "place.cache")
public class PlaceCacheConfiguration {
    private long durationSeconds = 10 * 60 * 24; // 10 days
}
```

```
18 logging:
19   level:
20     root: debug
21 place:
22   cache:
23     duration-seconds: 60
```

환경 변수나, 설정 파일에 정의하면, 캐시 유효기간 설정 가능

이 시간 보다 오래된 경우 캐시를 조회하지 않음

캐싱 기능 추가

2 usages Onji Kim

`@Override`

```
public PlaceDto getPlace(String placeId) {  
    // check cache  
    Optional<PlaceDto> cache = placeCacheService.get(placeId);  
    if (cache.isPresent()) {  
        log.debug("Cache Hit - Get Place From Cache : {}", placeId);  
        return cache.get();  
    }  
    // cache miss  
    // google api  
    log.debug("Cache Miss - Get Place From Google API : {}", placeId);  
  
    final GooglePlace place = googlePlaceApi.placeDetails(placeId)  
        .orElseThrow(() -> new PlaceRetrieveFailedException("Place Not Found", null));  
  
    // cache  
    log.debug("Cache Put - Put Place To Cache : {}", placeId);  
    placeCacheService.put(place);  
    return place;  
}
```

프록시 패턴으로 캐시로직 분리

인터페이스 정의

10 usages 1 implementation Onji Kim

```
public interface PlaceService {
```

```
/**
```

```
 * 장소를 조회합니다.
```

```
 *
```

```
 * @param placeId 장소 id, not null
```

```
 * @return 조회된 장소
```

```
 * @throws PlaceRetrieveFailedException 장소 조회에 실패한 경우(없는 게 아니라, 조회 실패)
```

```
 */
```

3 usages 1 implementation Onji Kim

```
Optional<PlaceView> getPlace(String placeId) throws PlaceRetrieveFailedException;
```

```
/**
```

```
 * 여러 장소를 한번에 조회합니다.
```

```
 * @param placeIds 장소 id 목록, not null
```

```
 * @return 조회된 장소 목록, 장소가 존재하지 않을 경우, Empty List
```

```
 * @throws PlaceRetrieveFailedException 장소 조회에 실패한 경우(없는 게 아니라, 조회 실패)
```

```
 */
```

2 usages 1 implementation Onji Kim

```
List<PlaceView> getPlaces(String[] placeIds) throws PlaceRetrieveFailedException;
```

캐시 로직 분리

```
@Slf4j
@RequiredArgsConstructor
public class PlaceServiceCacheProxy implements PlaceService {

    private final PlaceService placeService;
    private final PlaceCacheService placeCacheService;

    @Override
    public Optional<PlaceView> getPlace(String placeId) throws PlaceRetrieveFailedException {
        // check cache
        Optional<PlaceView> cache = placeCacheService.get(placeId);
        if (cache.isPresent()) {
            log.debug("Cache Hit - Get Place From Cache : {}", placeId);
            return cache;
        }

        // cache miss
        log.debug("Cache Miss - Get Place From : {}", placeService.getClass().getSimpleName());
        final Optional<PlaceView> place = placeService.getPlace(placeId);

        // load to cache
        if (place.isPresent()) {
            // cache
            log.debug("Cache Put - Put Place To Cache : {}", placeId);
            placeCacheService.put(place.get());
        }

        return place;
    }
}
```

```
@Slf4j
@Service
@Validated
@RequiredArgsConstructor
public class PlaceServiceImpl implements PlaceService {

    private final PlaceRepository placeRepository;

    3 usages  Onji Kim
    @Override
    public Optional<PlaceView> getPlace(String placeId) {
        final Optional<PlaceView> place;
        try {
            return placeRepository.getPlace(placeId);
        } catch (DataAccessException e) {
            throw new PlaceRetrieveFailedException("Place Not Found", e);
        }
    }

    2 usages  Onji Kim
    @Override
    public List<PlaceView> getPlaces(String[] placeIds) throws PlaceRetrieveFailedException {
        Assert.notNull(placeIds, message: "placeIds must not be null");
        Assert.noNullElements(placeIds, message: "placeIds must not contain null");
        try {
            return placeRepository.getPlaces(placeIds);
        } catch (DataAccessException e) {
            throw new PlaceRetrieveFailedException("Place Not Found", e);
        }
    }
}
```

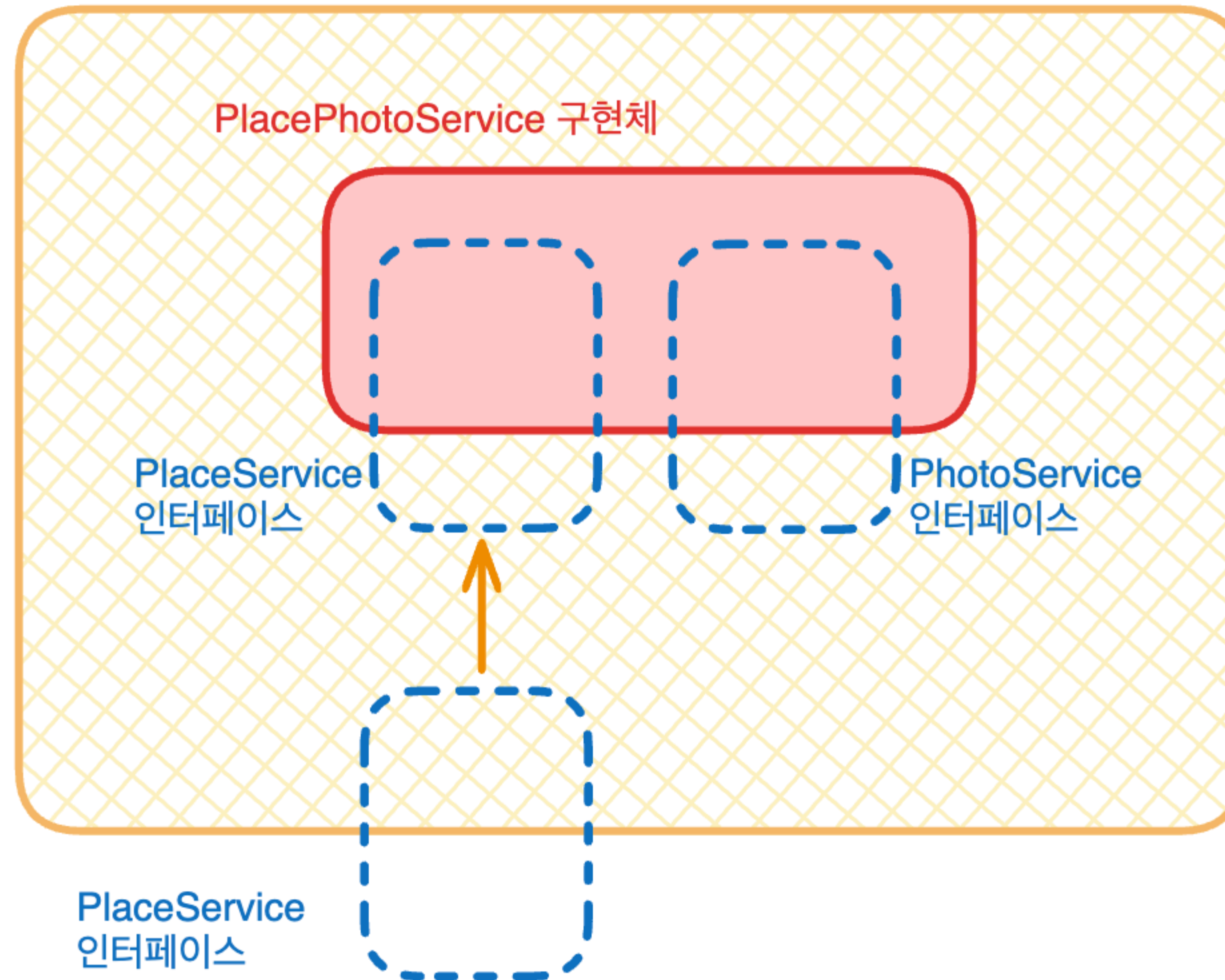
스프링 빈 확장

```
@Component
public class BeanDecorator implements BeanPostProcessor {
    @Override
    public Object postProcessAfterInitialization(Object bean, String beanName) throws BeansException {
        if (bean instanceof PlaceService service) {
            return new PlaceProxy(service);
        }
        return bean;
    }
}
```

문제점

이러한 문제점 발생

Proxy



AOP를 활용해서 캐시 로직 분리,
Data Access Layer로 캐시 역할 변경

Aspect 작성

```
@Component
@Aspect
@RequiredArgsConstructor
public class PlaceRepositoryCacheAspect {
    private final PlaceCacheService placeCacheService;

    /unchecked/
    @Around("execution(java.util.Optional<click.porito.travel_core.place.dto.PlaceView> click.porito.travel_core.place.dao.P")
    public Object aroundGetPlace(ProceedingJoinPoint joinPoint, String placeId) throws Throwable {
        // check cache
        Optional<PlaceView> placeView = placeCacheService.get(placeId);
        if (placeView.isPresent()) {
            log.debug("Cache Hit - Get Place From Cache : {}", placeId);
            return placeView;
        }

        // cache miss
        log.debug("Cache Miss - Get Place From : {}", joinPoint.getTarget().getClass().getSimpleName());
        Optional<PlaceView> result = (Optional<PlaceView>) joinPoint.proceed();

        // load to cache
        if (result.isPresent()) {
            // cache
            log.debug("Cache Put - Put Place To Cache : {}", placeId);
            placeCacheService.put(result.get());
        }

        return result;
    }
}
```

캐시 적용 결과

속도 향상

localhost:29091/v1/places/ChIJFXXzNhPnAGARUifDEqMzXI4

GET localhost:29091/v1/places/ChIJFXXzNhPnAGARUifDEqMzXI4

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (14) Test Results

Status: 200 OK Time: 1164 ms Size: 7.83 kB

```
{
  "id": "ChIJFXXzNhPnAGARUifDEqMzXI4",
  "displayName": {
    "text": "난바",
    "languageCode": "ko"
  }
}
```

```
2024-01-09T14:33:39.039+09:00 DEBUG 6195 --- [io-29091-exec-4] c.p.place_service.service.ServiceImpl : Cache Miss - Get Place From
Google API : ChIJFXXzNhPnAGARUifDEqMzXI4
```

```
2024-01-09T14:33:39.981+09:00 DEBUG 6195 --- [io-29091-exec-4] c.p.place_service.service.ServiceImpl : Cache Put - Put Place To Cache
: ChIJFXXzNhPnAGARUifDEqMzXI4
```


속도 향상(1164 -> 32) 3600 %

Http의 캐시 컨트롤까지 쓰면 더 향상될 것으로 예상

localhost:29091/v1/places/ChIJFXXzNhPnAGARUifDEqMzXl4

GET localhost:29091/v1/places/ChIJFXXzNhPnAGARUifDEqMzXl4

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (14) Test Results

Status: 200 OK Time: 32 ms Size: 5.5 KB

Save as example

Pretty Raw Preview Visualize JSON

```
{
  "id": "ChIJFXXzNhPnAGARUifDEqMzXl4",
  "name": "난바",
  "tags": [
    "subway_station",
```

```
2024-01-09T14:38:22.893+09:00 DEBUG 6195 --- [io-29091-exec-8] c.p.place_service.service.ServiceImpl : Cache Hit - Get Place From
Cache : ChIJFXXzNhPnAGARUifDEqMzXl4
```

Filter {_id : "ChIJFXXzNhPnAGARUifDEqMzXl4"} [Generate query](#) [Explain](#) [Reset](#) [Find](#) [Options](#)

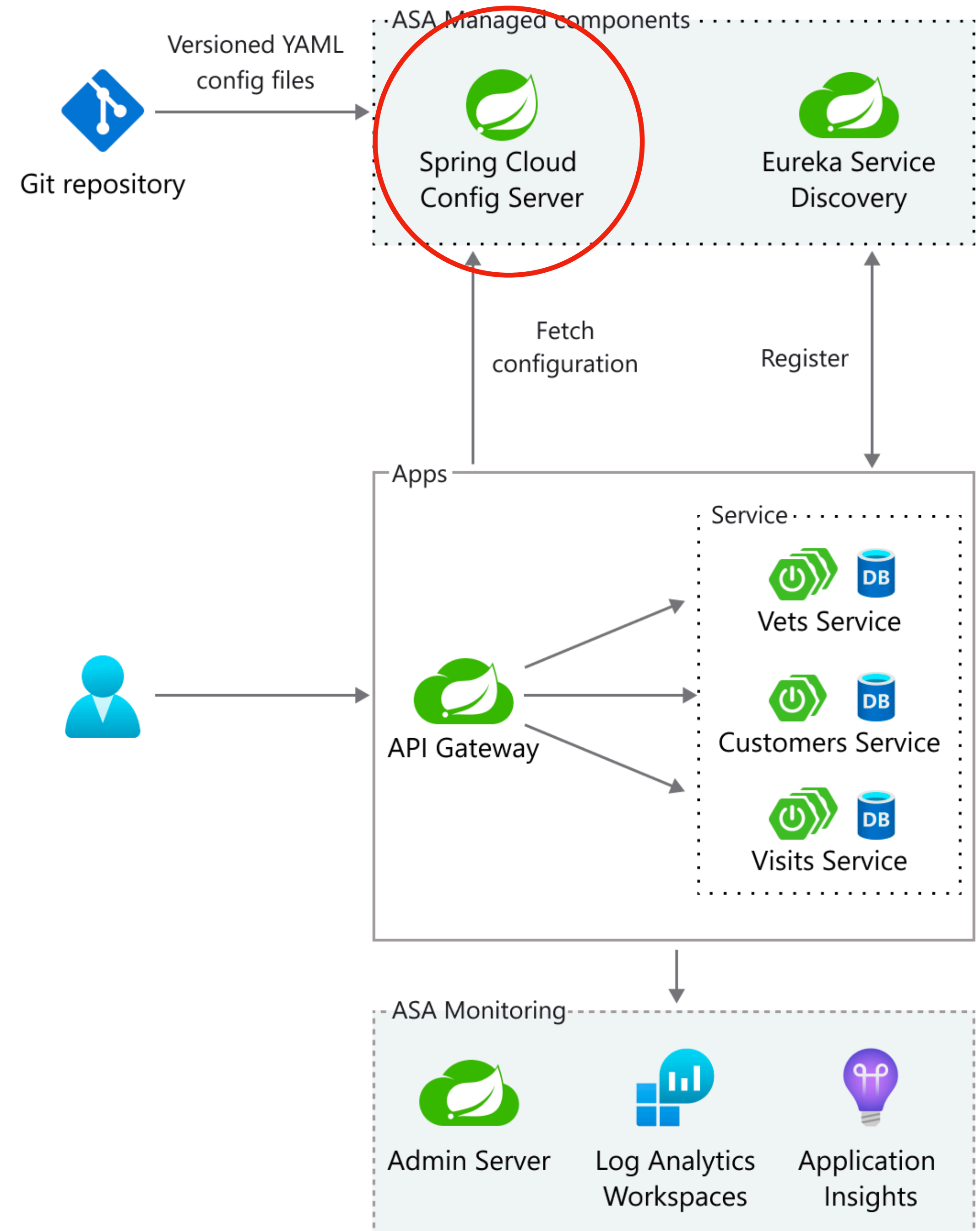
[+ ADD DATA](#) [EXPORT DATA](#)

1 - 1 of 1

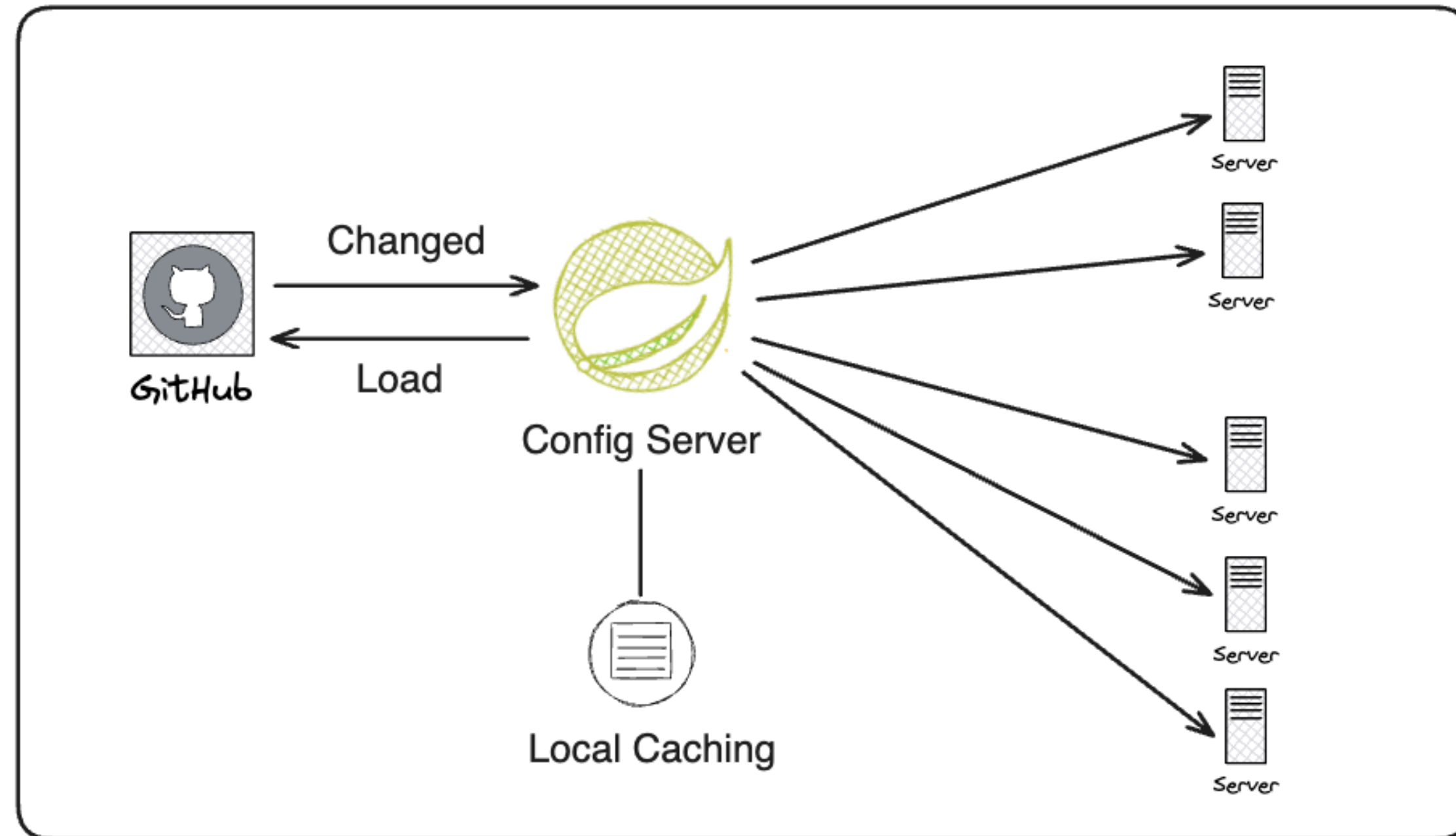
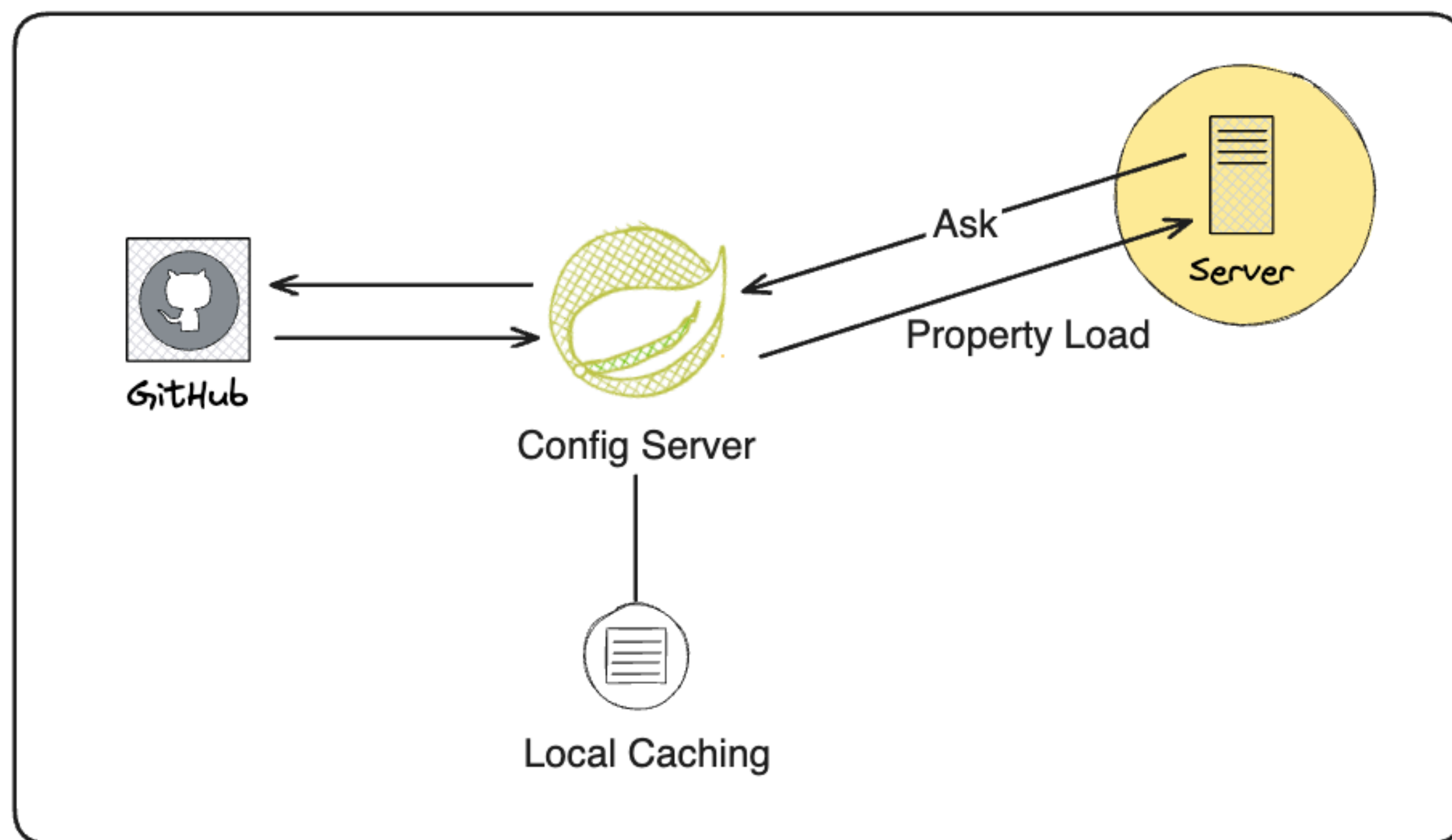
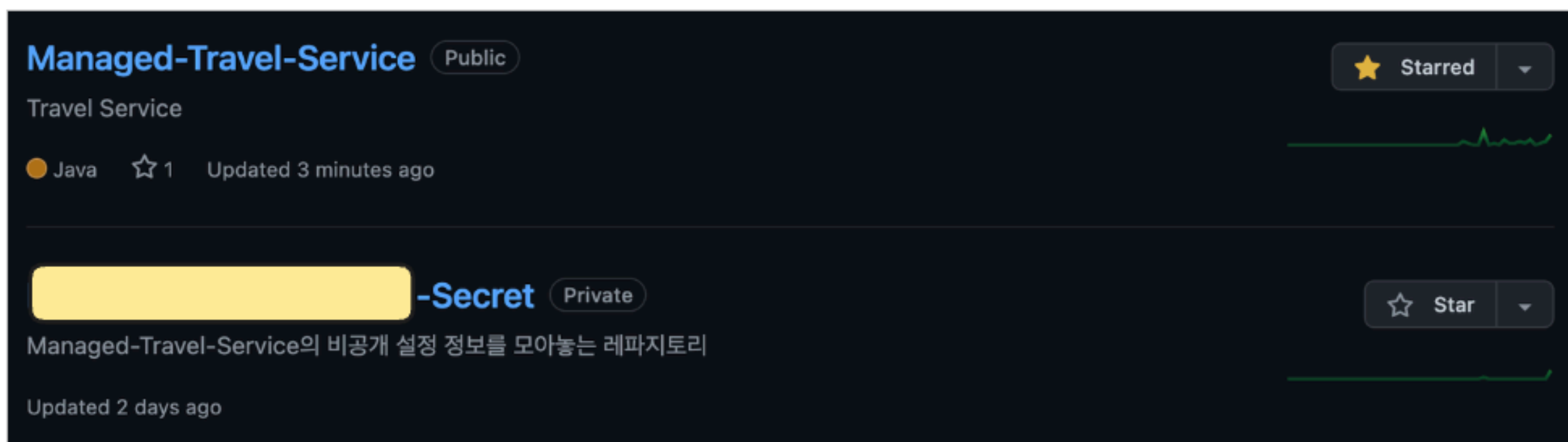
```
_id: "ChIJFXXzNhPnAGARUifDEqMzXl4"
name: "난바"
tags: Array (4)
address: "일본 〒542-0076 오사카부 오사카시 주오구 Namba, 1 조메-9"
location: Object
photos: Array (10)
updatedAt: 2024-01-09T05:33:40.000+00:00
_class: "place"
```


Config Server 도입


Spring Cloud Config



뭐가 좋냐?




Git Submodule로 시크릿 레포 관리


 **Managed-Travel-Service** Public

Unpin Unwatch 1

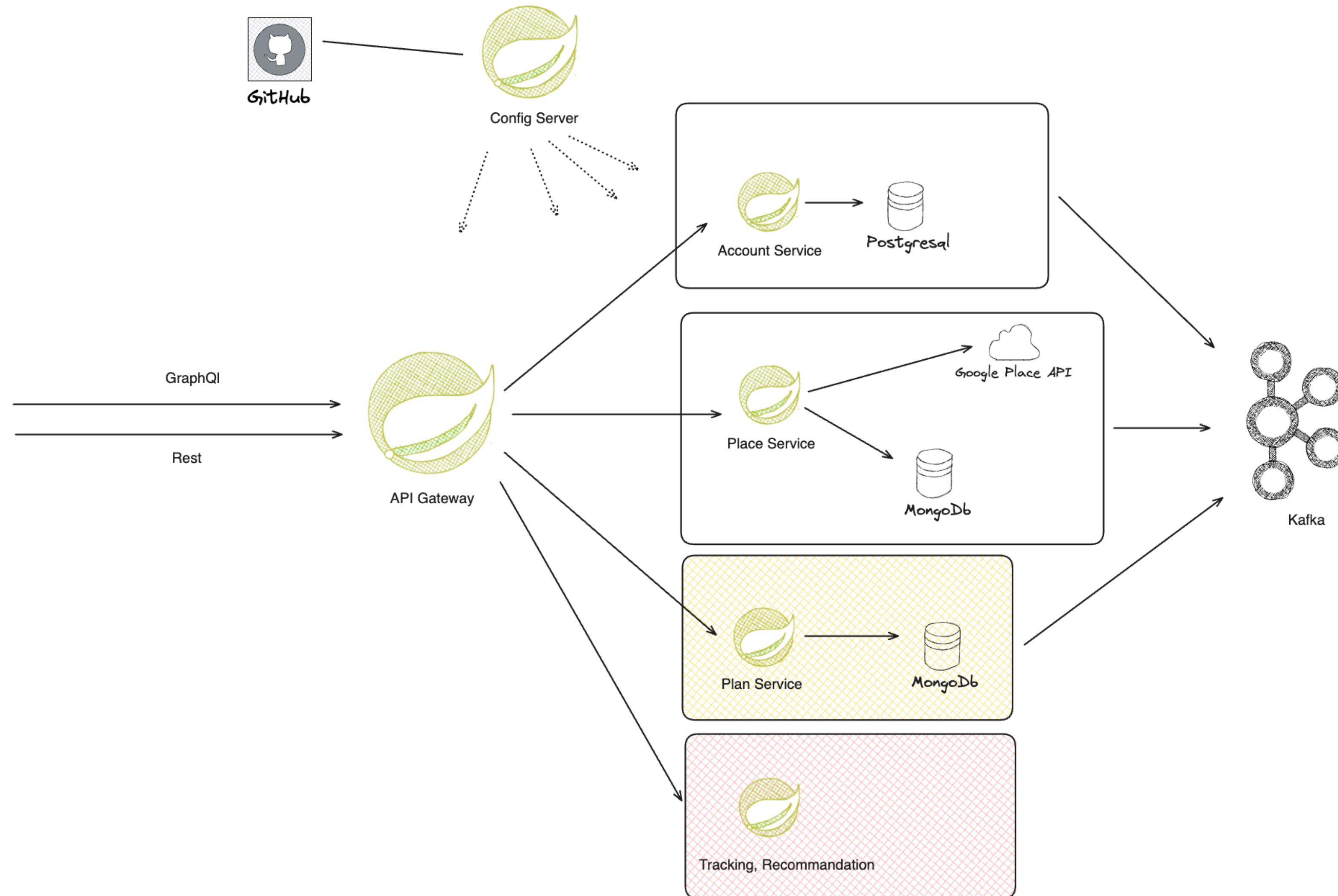
main 1 Branch 0 Tags

t Add file <> Code

 **onjik** fix : 불필요한 PlaceServiceCacheAspect.java 제거 60dc3af · 10 minutes ago 33 Commits

backend	fix : 불필요한 PlaceServiceCacheAspect.java 제거	10 minutes ago
docs/readme	docs : refactor and update readme	3 months ago
 secret @ b5f5500	feat : Spring Cloud Config 기반 설정 관리	2 days ago
.gitignore	init : add gitignore	4 months ago
.gitmodules	chore : manage secret file with submodules	3 months ago
README.md	!BREAKING CHANGE : Change Structure Modular Monoli...	2 months ago

지금까지 구현한 서버



감사합니다.