

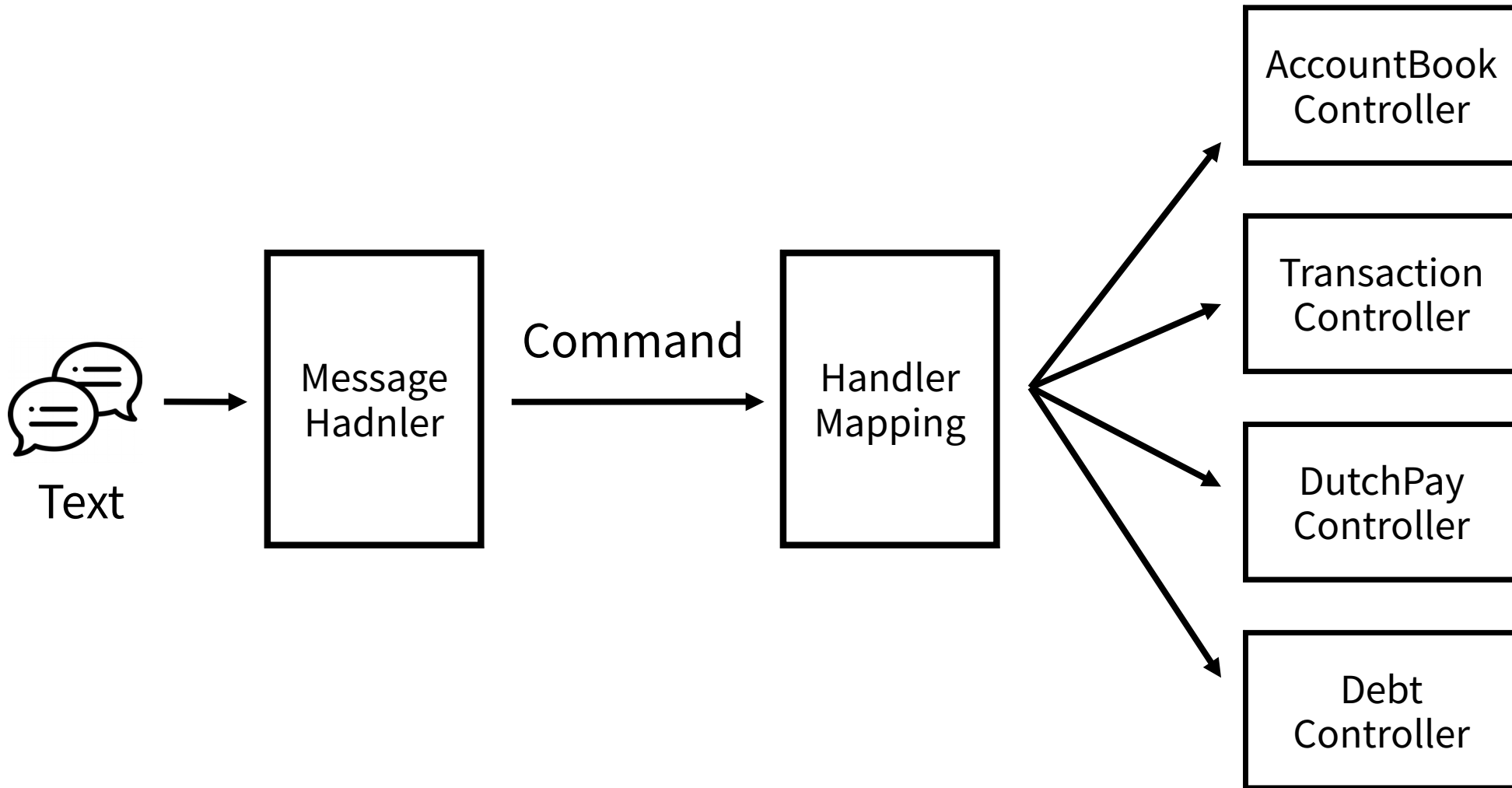
메소드 추출

단일 책임 원칙

명확한 코드

네이밍

```
switch (text) {
    case "조회": {
        reply(replyToken, accountBookDto.toString());
        break;
    }
    case "상세조회": {
        AccountBookDetailDto accountBookDetailDto = botService.getAccountBookDetailDto(accountBookDto);
        reply(replyToken, accountBookDetailDto);
        break;
    }
    case "더치페이": {
        if (accountBookType == AccountBookType.P) {
            reply(replyToken, message: "개인 채팅방에서 이용할 수 없는 기능입니다");
        } else {
            vop.set(accountBookDto.getAccountBookId() + "_d", LocalDateTime.now());
            reply(replyToken, message: "더치 페이 기록 시작\n" + LocalDateTime.now());
        }
        break;
    }
    case "장부기입": {
        hop.put(accountBookId, hashCode: "accountBookSeq", accountBookDto.getAccountBookSeq());
        hop.put(accountBookId, hashCode: "transactionStep", TransactionStep.TRANSACTION_TYPE);
        redisTemplate.expire(accountBookId, timeout: 5L, TimeUnit.MINUTES);
        reply(replyToken, message: "거래 타입을 입력해주세요");
        break;
    }
}
```



```
public enum Command {
```

```
    /*          Default Message          */

    // 메뉴
    MENU(Arrays.asList("메뉴", "메뉴보기", "버튼", "시작", "menu", "No!"), Condition.DEFAULT, AccountBookType.P,
        AccountBookType.G),

    // 수정 및 삭제 url 버튼
    MODIFY_OR_DELETE(Arrays.asList("수정/삭제", "수정", "삭제", "웹", "링크", "modify", "delete", "web", "url", "link"),
        Condition.DEFAULT, AccountBookType.P, AccountBookType.G),

    // 사용법
    HOW_TO_USE(Arrays.asList("사용법", "소개", "help", "introduce", "howtouse"), Condition.DEFAULT, AccountBookType.P,
        AccountBookType.G),

    /*          AccountBook          */

    // 가계부 조회
    ACCOUNTBOOK_INFO(Arrays.asList("조회", "조회해줘", "가계부", "가계부조회", "info"), Condition.DEFAULT, AccountBookType.P,
        AccountBookType.G),

    // 가계부 상세 조회
    ACCOUNTBOOK_DETAIL_INFO(Arrays.asList("상세조회", "상세", "가계부상세", "가계부상세조회", "detail"), Condition.DEFAULT,
        AccountBookType.P, AccountBookType.G),

    /*          Transaction          */

    // 장부 기입 시작
    BOOK_ENTRY_START(Arrays.asList("장부기입", "기입", "거래", "거래입력", "거래추가", "transaction"),
        Condition.DEFAULT, AccountBookType.P, AccountBookType.G),

    // 장부 기입 중간 단계 (장부 기입 요소 입력) : 거래 타입, 금액, 카테고리
    BOOK_ENTRY_WRITING(Collections.EMPTY_LIST, Condition.ADD_TRANSACTION, AccountBookType.P, AccountBookType.G),

    // 장부 기입 완료 : 메모 입력 단계
    BOOK_ENTRY_FINISH(Collections.EMPTY_LIST, Condition.ADD_TRANSACTION, AccountBookType.P, AccountBookType.G),

    // 장부 기입 : 잘못된 입력
    BOOK_ENTRY_WRONG_INPUT(Collections.EMPTY_LIST, Condition.ADD_TRANSACTION, AccountBookType.P, AccountBookType.G),

    // 장부 기입 취소
    BOOK_ENTRY_TERMINATE(Arrays.asList("취소", "장부기입취소", "기입취소", "거래취소", "거래입력취소", "거래추가취소", "terminate"),
        Condition.ADD_TRANSACTION, AccountBookType.P, AccountBookType.G),

    // 장부 기입 생략
    BOOK_ENTRY_SKIP(Arrays.asList("생략", "장부기입생략", "기입생략", "거래생략", "거래입력생략", "거래추가생략", "skip"),
```

CommandMapping Annotation

```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface CommandMapping {

    Command command();

    String description();

}
```



```
@CommandMapping(command = Command.ACCOUNTBOOK_INFO, description = "가계부 조회")
public Object getAccountBookInfo(AccountBookDto accountBookDto) {
    return ReplyMessageSupplier.replyGetMenuForm(accountBookDto.toString());
}
```