

함께 해봐요. Spring Event Internal

박성훈 (Coupang)





누구를 위한 Hands-On Lab 인가요?



Event를 통한 느슨한 결합을 구현해 보시고 싶으신 분



Spring Event Internal 구현 방식을 알고 싶으신 분





Hands-On Lab의 목표

YOUTHCON '24



1. Event 발행/구독 방식을 직접 구현하며 어떤 요소들이 필요한지 이해합니다.
2. 비동기 Event 구현 방식을 이해합니다.
3. DB Transaction Commit 이후에
Event 구독을 구현하는 방법을 이해합니다.
4. **사실 해보면 별거 아니다** 라는 자신감을 얻습니다.



이런 부분을 유념해 주세요!



- 우리 코드의 많은 구현은 Spring Event의 구현을 따라갑니다.

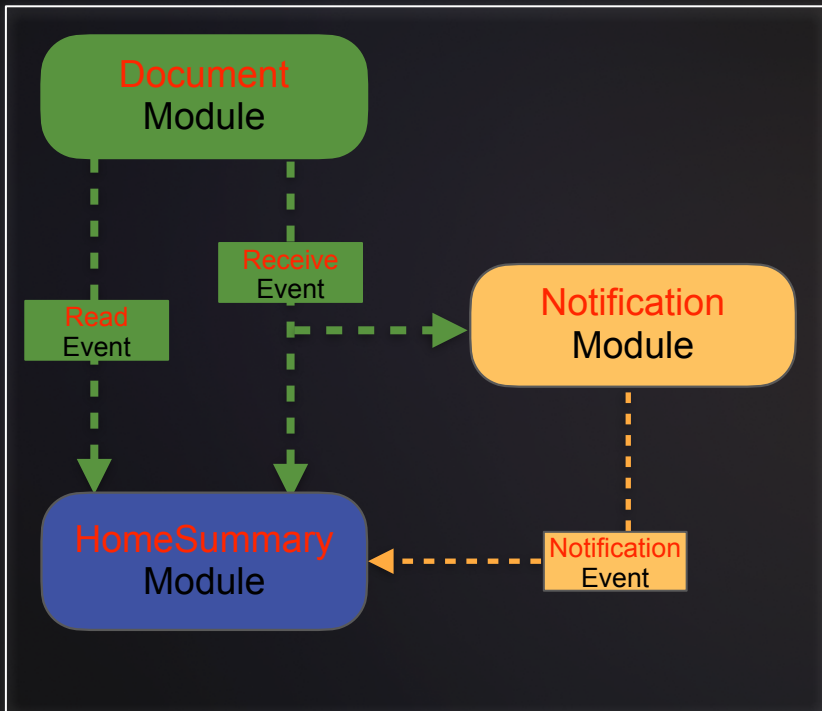
간혹 이상하다고 느껴질 수 있는 부분이 있으면 둘 중에 하나라고 생각해 주세요.

- Spring Event에서 구현하는 방식이거나
- Spring Event에서 구현하는 방식이 너무 어려워서 시간 관계상 대안으로 구현한 것



프로젝트 구조 및 요구사항

YOUTHCON '24



- **Document Module**

- 문서 수신/열람 기능
- 수신 후 **Receive Event** 발행
- 열람 후 **Read Event** 발행

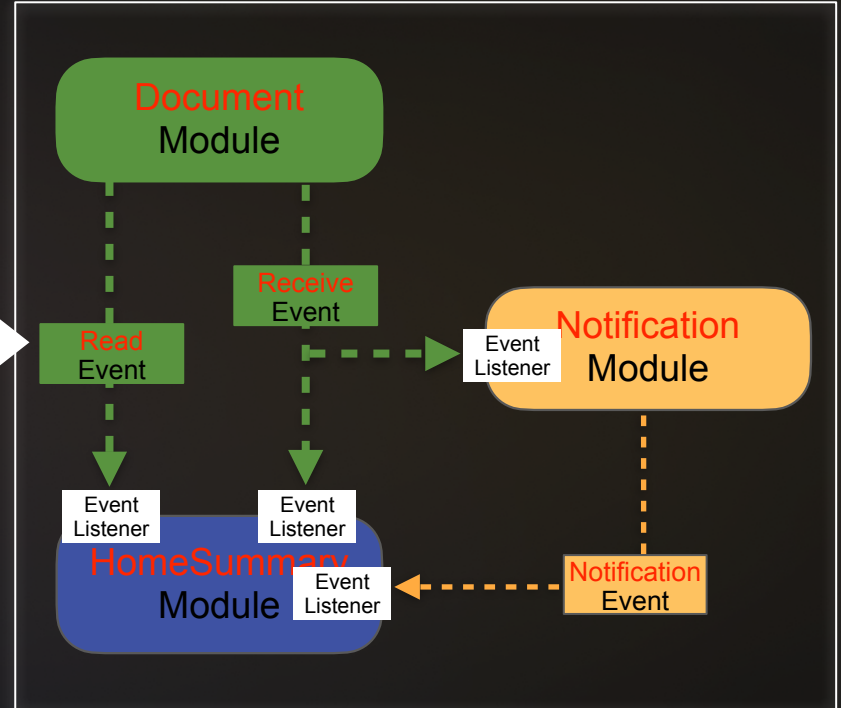
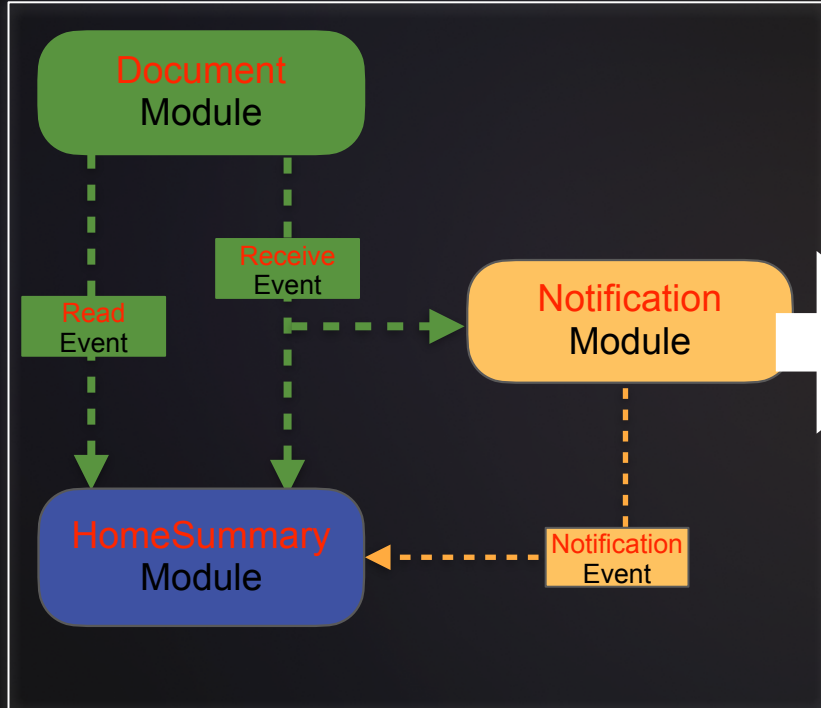
- **Notification Module**

- 문서가 수신 될 때 알람 발송
- 알람 발송 이후 **Notification Event** 발행

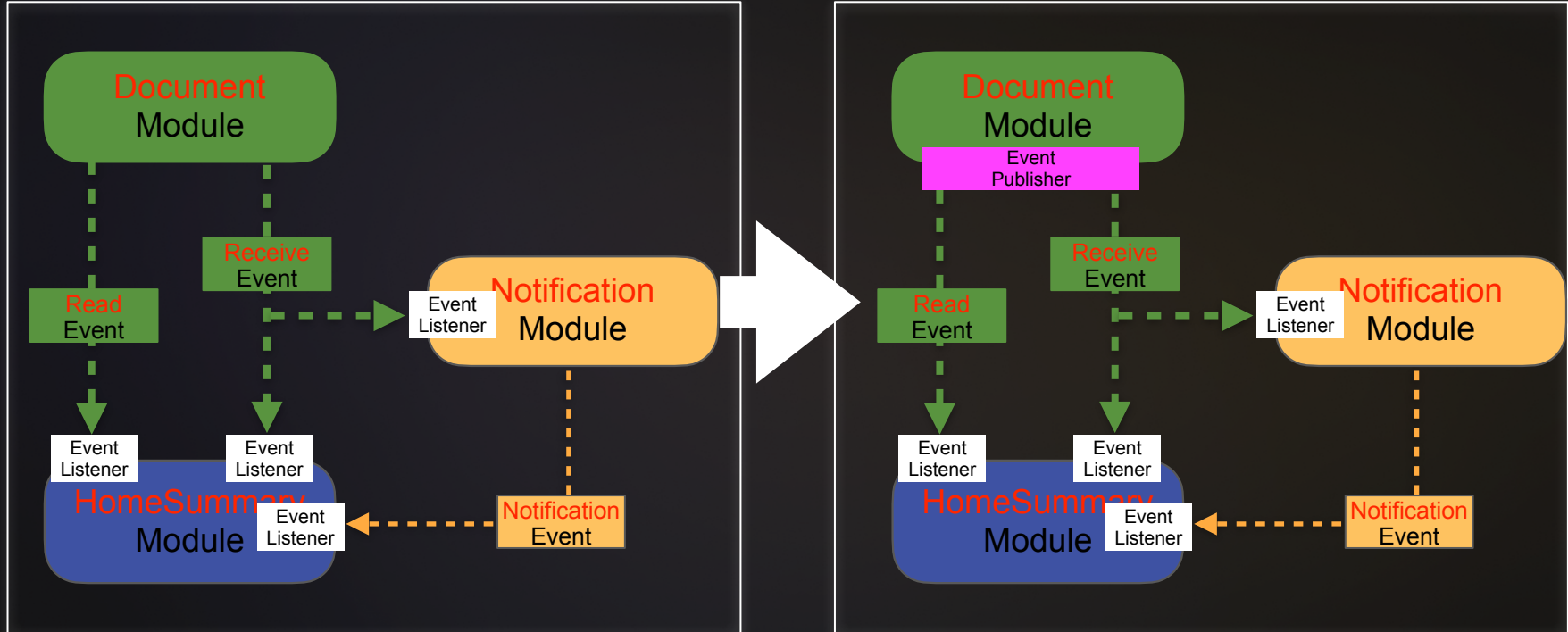
- **HomeSummary Module**

- 홈 화면 요약 기능
- 문서 수신/열람/알람 발송 Event 구독

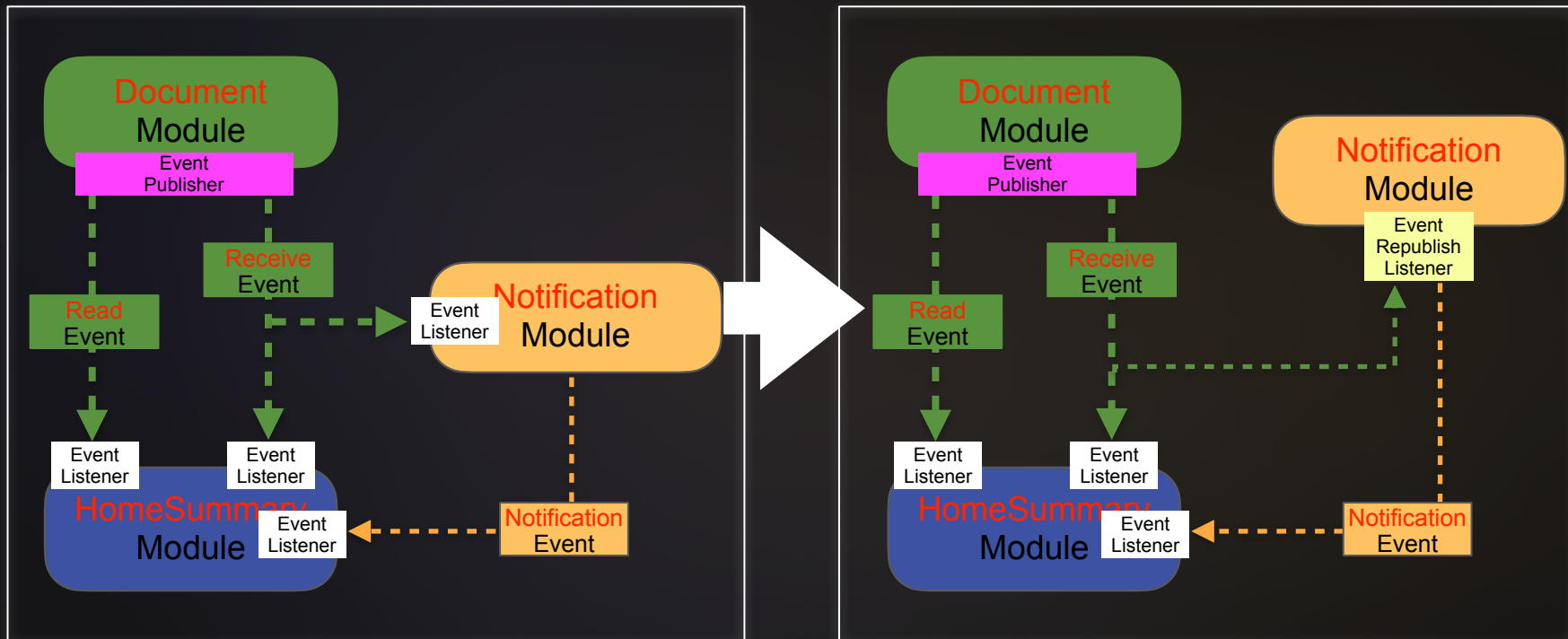
1. Event Listener 구현



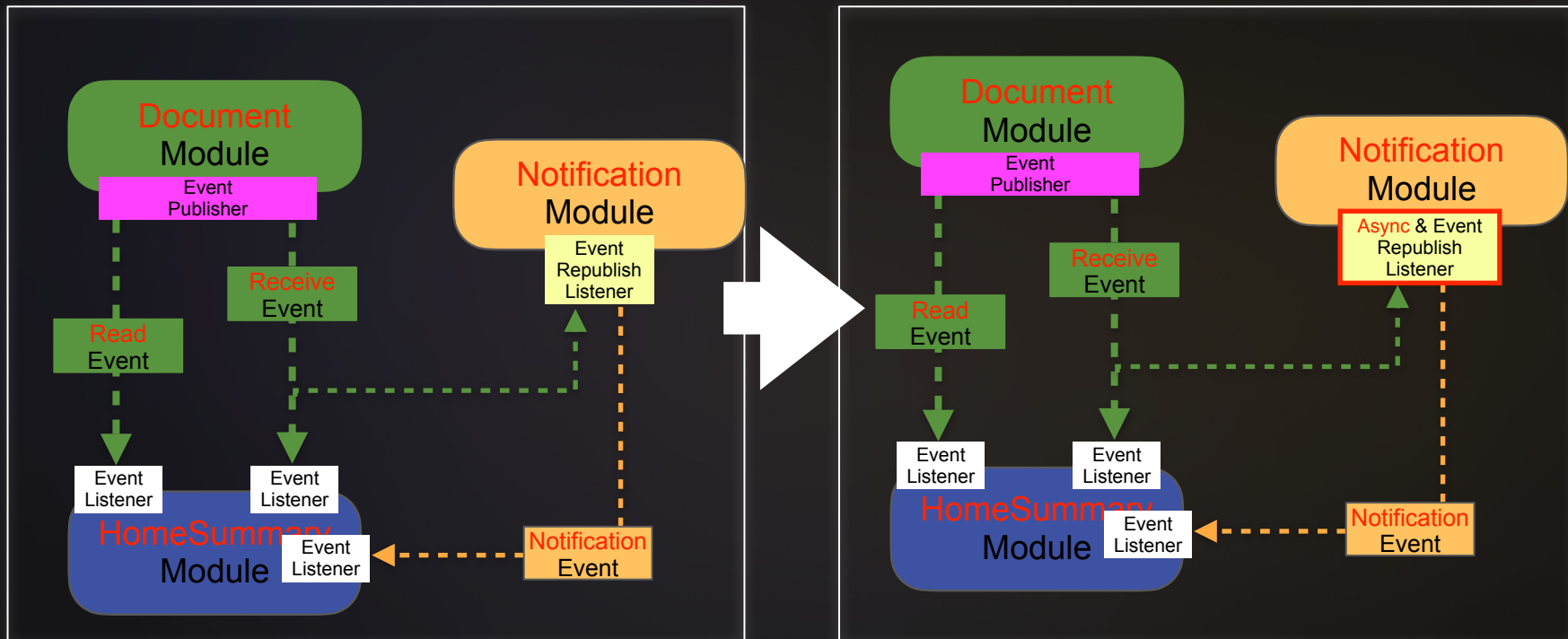
2. Event Publisher 구현



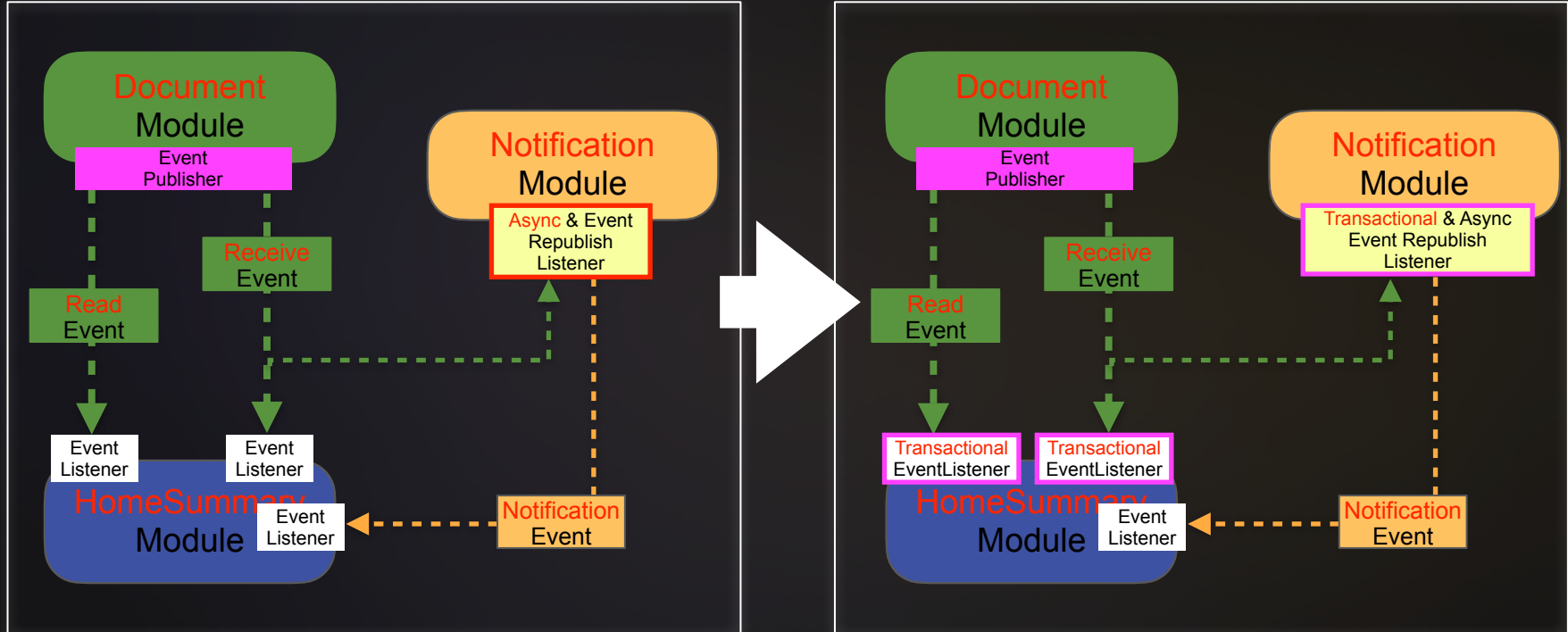
3. Event Republish Listener 구현



4. Async Event Listener 구현



5. Transactional Event Listener 구현



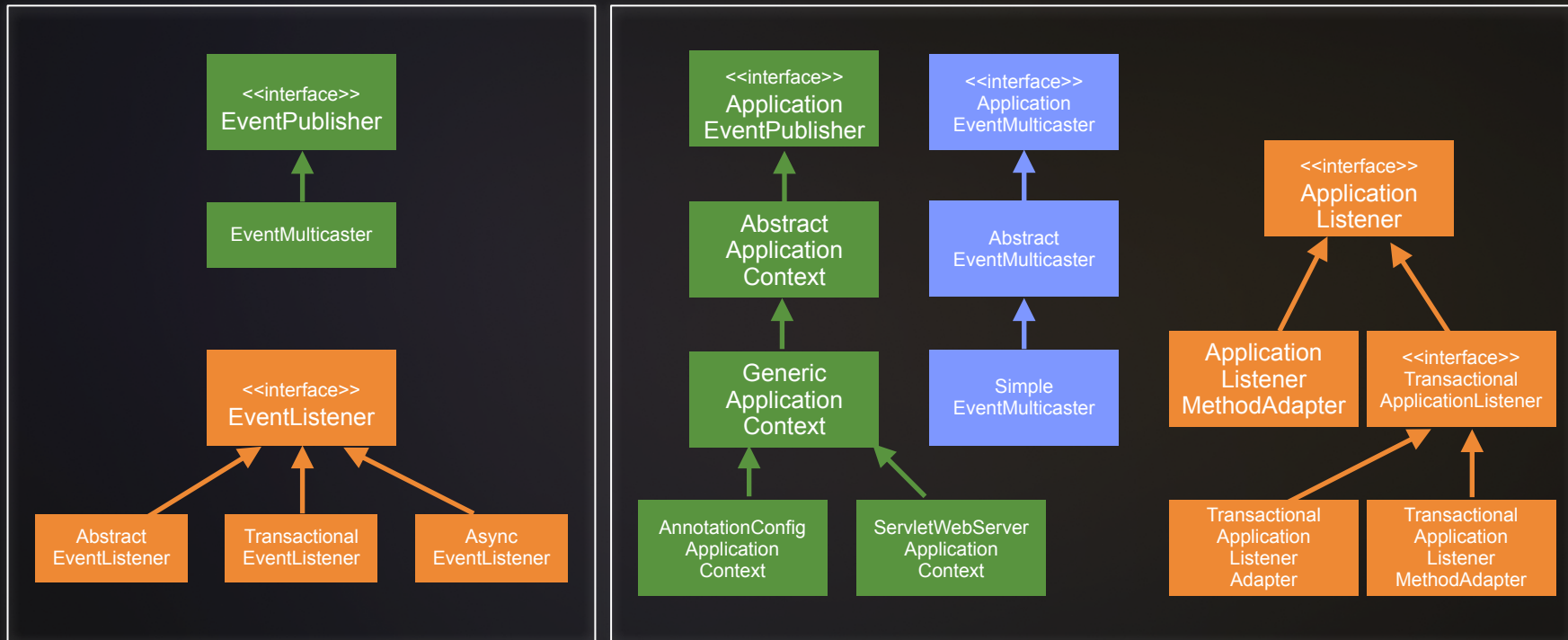


그래서 **Spring**은 어떻게?





실제 Spring Event 구현체





- **혹시라도 따라오시지 못하셨다면..!**
 - 꼭 Github의 step branch들을 따라가시면서 확인해 보시길 바라요
- **Spring이 어떻게 구현했는지도 자세히 봐보세요!**
 - 우리 코드는 굉장히 단순하게 만들었어요
 - 실제 구현체는 굉장히 세밀하고 신중해야 해요
- **Event 오남용을 해보세요! (단 안전하게)**
 - 실수를 해봐야 정도를 알아요
 - 단, 꼭 안전한 곳에서 남용해 보세요



Thank you.

QnA

