



이벤트 프로그래밍 (2021.04.06)



0. 준비

posco ict

0.1 JDK 설치

0.2 maven 설치

0.3 intellij 설치

0.4 Insomnia 설치

✓ <https://github.com/ajdkbuild/ajdkbuild>

Downloads for Windows x86_64

- 1.8.0_282-1 (LTS, supported until [May 2026, announcement](#))
 - [java-1.8.0-openjdk-1.8.0.282-1.b08.ajdkbuild.windows.x86_64.zip](#) (sha256)
 - [java-1.8.0-openjdk-1.8.0.282-1.b08.ajdkbuild.windows.x86_64.msi](#) (sha256)
- 11.0.10.9-1 (LTS, supported until [October 2024, announcement](#))
 - [java-11-openjdk-11.0.10.9-1.windows.ajdkbuild.x86_64.zip](#) (sha256)
 - [java-11-openjdk-11.0.10.9-1.windows.ajdkbuild.x86_64.msi](#) (sha256)

Note: obsolete downloads are available at the bottom of this page

Other downloads

Windows x86 ([issue](#))

- 1.8.0_282-1 ([announcement](#))
 - [java-1.8.0-openjdk-1.8.0.282-1.b08.ajdkbuild.windows.x86.zip](#) (sha256)
 - [java-1.8.0-openjdk-1.8.0.282-1.b08.ajdkbuild.windows.x86.msi](#) (sha256)

✓ <https://maven.apache.org/download.cgi>

Downloading Apache Maven 3.6.3

Apache Maven 3.6.3 is the latest release and recommended version for all users.

The currently selected download mirror is <https://downloads.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available. You may also consult the [complete list of mirrors](#).

Other mirrors:

System Requirements

Java Development Kit (JDK)	Maven 3.3+ require JDK 1.7 or above to execute - they still allow you to build against 1.3 and other JDK versions by Using Toolchains
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts and Windows batch files.

Files


Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.


	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.6.3-bin.tar.gz	apache-maven-3.6.3-bin.tar.gz.sha512	apache-maven-3.6.3-bin.tar.gz.asc
Binary zip archive	apache-maven-3.6.3-bin.zip	apache-maven-3.6.3-bin.zip.sha512	apache-maven-3.6.3-bin.zip.asc
Source tar.gz archive	apache-maven-3.6.3-src.tar.gz	apache-maven-3.6.3-src.tar.gz.sha512	apache-maven-3.6.3-src.tar.gz.asc
Source zip archive	apache-maven-3.6.3-src.zip	apache-maven-3.6.3-src.zip.sha512	apache-maven-3.6.3-src.zip.asc

0.3 IntelliJ 설치

✓ <https://www.jetbrains.com/ko-kr/idea/download/#section=windows>

**IntelliJ IDEA**

Coming in 2021.1 새로운 기능 기능 리소스 구매 [다운로드](#)



버전: 2020.3.3
빌드: 203.7717.56
2021년 3월 16일

[릴리스 노트](#)

[시스템 요구 사항](#)
[설치 안내](#)
[기타 버전](#)

다운로드 IntelliJ IDEA

[Windows](#) macOS Linux

Ultimate

웹 및 엔터프라이즈 개발용

다운로드

.exe

30일 무료 평가판

Community

JVM 및 Android 개발용

다운로드

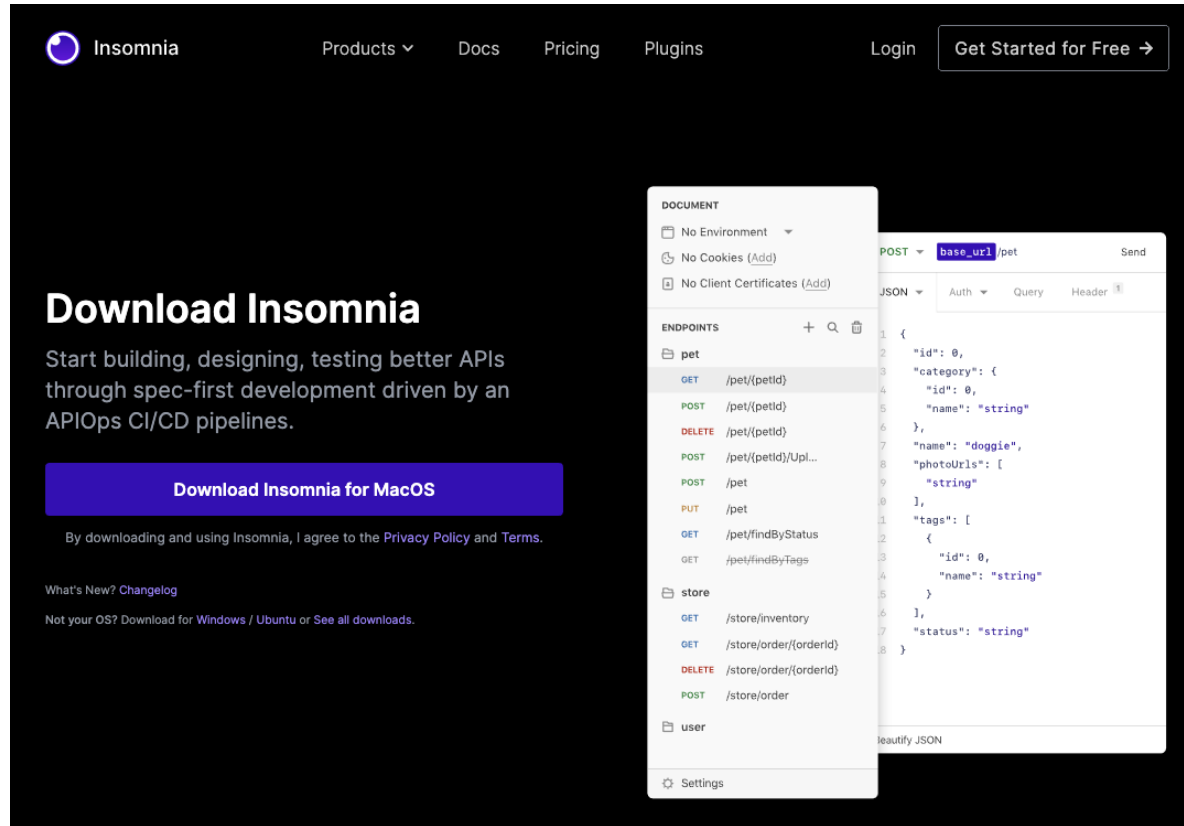
.exe

무료, 오픈 소스

	IntelliJ IDEA Ultimate	IntelliJ IDEA Community 에디션 i
Java, Kotlin, Groovy, Scala	✓	✓
Android i	✓	✓
Maven, Gradle, sbt	✓	✓
Git, SVN, Mercurial	✓	✓

0.4 Insomnia 설치

✓ <https://insomnia.rest/download>





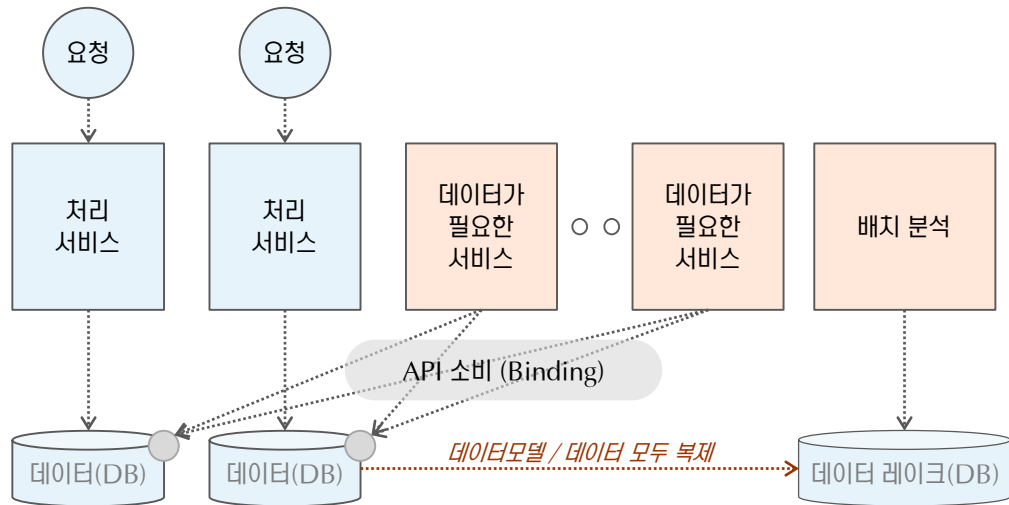
0. 마이크로서비스 기반 애플리케이션 아키텍처 - 이벤트 다시보기

posco ict

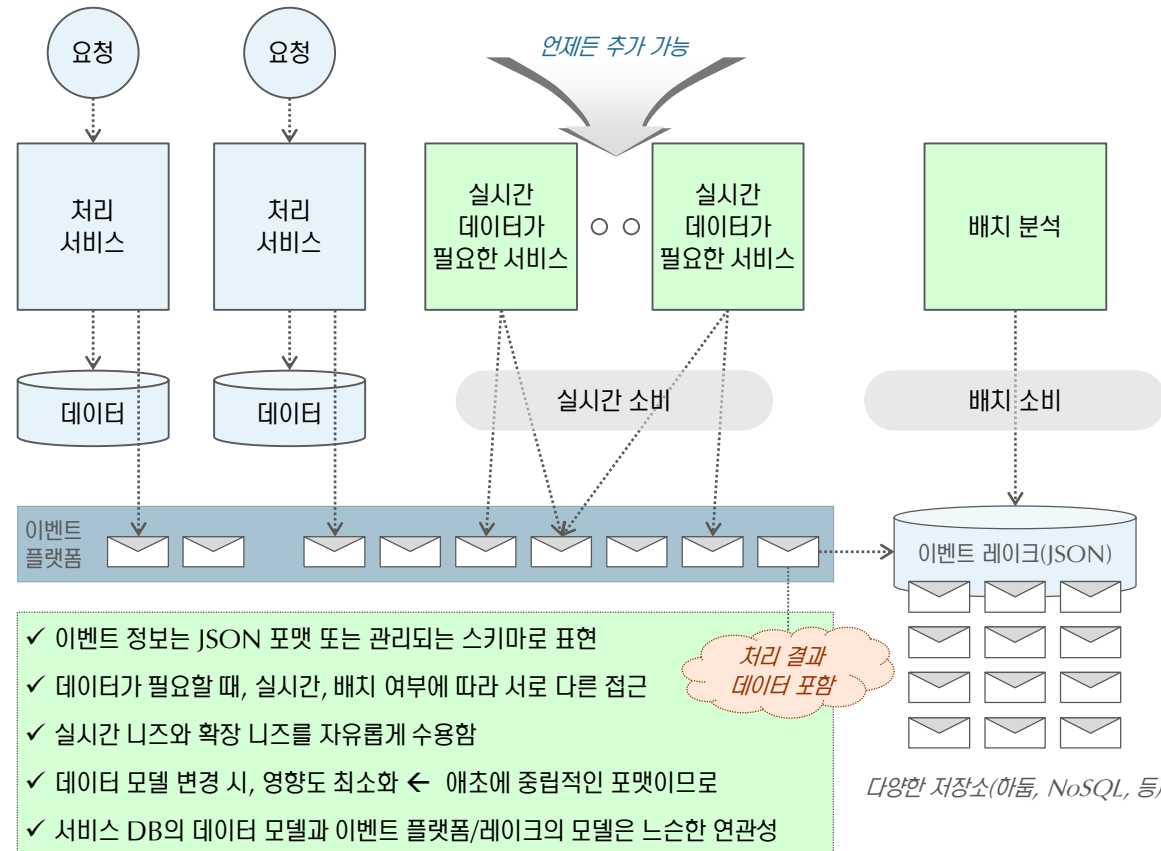
- 0.1 아키텍처 특성(8/9) - 이벤트-드리븐
- 0.2 도메인 이벤트 - Scope
- 0.3 기술 적용(3/7) - Event Sourcing/CQRS
- 0.4 기술 적용(6/7) - 일치성/추적성 1
- 0.5 기술 적용(7/7) - 일치성/추적성 2
- 0.6 Facade 모듈 구조(5/5) - Event consumer

0.1 아키텍처 특성(8/9) – 이벤트-드리븐

- ✓ DB 기반 데이터 처리와 소비 스타일에서 데이터가 필요할 경우, 해당 DB로 접근해야 합니다.
- ✓ 이런 구조는 데이터 소비 서비스의 경우, 데이터 생산 서비스의 데이터 모델에 의존적이며, 접근 제약이 많습니다(성능 등).
- ✓ 배치 분석을 위해 데이터를 모아두는 DB의 경우, 생산 DB와 의존성 때문에 변경이나 확장이 쉽지 않습니다.
- ✓ 이벤트 플랫폼을 바탕으로 독립적인 포맷을 가진 이벤트로 데이터를 소비하는 경우는 위의 제약 조건으로부터 자유롭습니다.



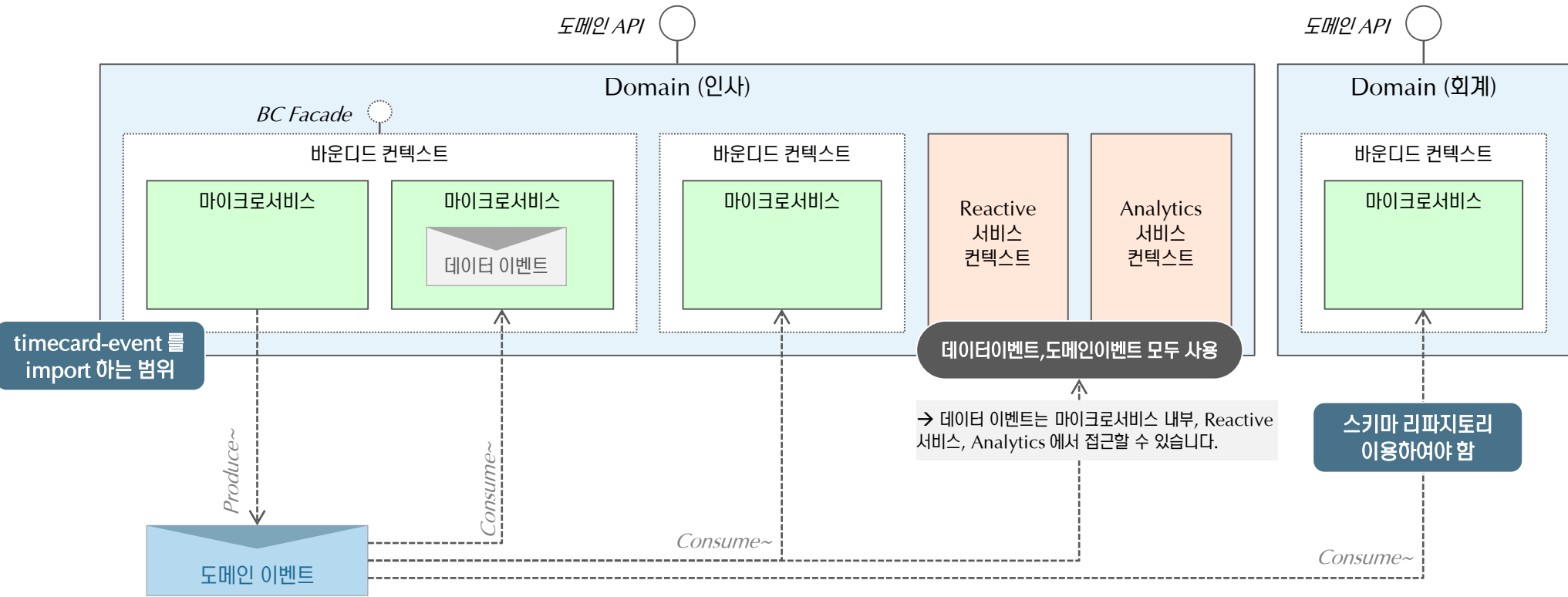
- ✓ 생산자, 소비자 모두 데이터 모델에 의존성이 있음
- ✓ 데이터가 필요할 경우, 데이터 있는 곳으로 접근 → 접근의 어려움, 부하 발생
- ✓ 실시간 니즈와 확장 니즈에 대한 제약 조건 있음
- ✓ 데이터 모델 변경 시, 모든 관련 시스템에 영향을 줌
- ✓ 데이터 레이크와 DB 상호간에 의존성을 갖게 되어 시스템 개선이 어려움



- ✓ 이벤트 정보는 JSON 포맷 또는 관리되는 스키마로 표현
- ✓ 데이터가 필요할 때, 실시간, 배치 여부에 따라 서로 다른 접근
- ✓ 실시간 니즈와 확장 니즈를 자유롭게 수용함
- ✓ 데이터 모델 변경 시, 영향도 최소화 ← 애초에 독립적인 포맷이므로
- ✓ 서비스 DB의 데이터 모델과 이벤트 플랫폼/레이크의 모델은 느슨한 연관성

0.2 도메인 이벤트 - Scope

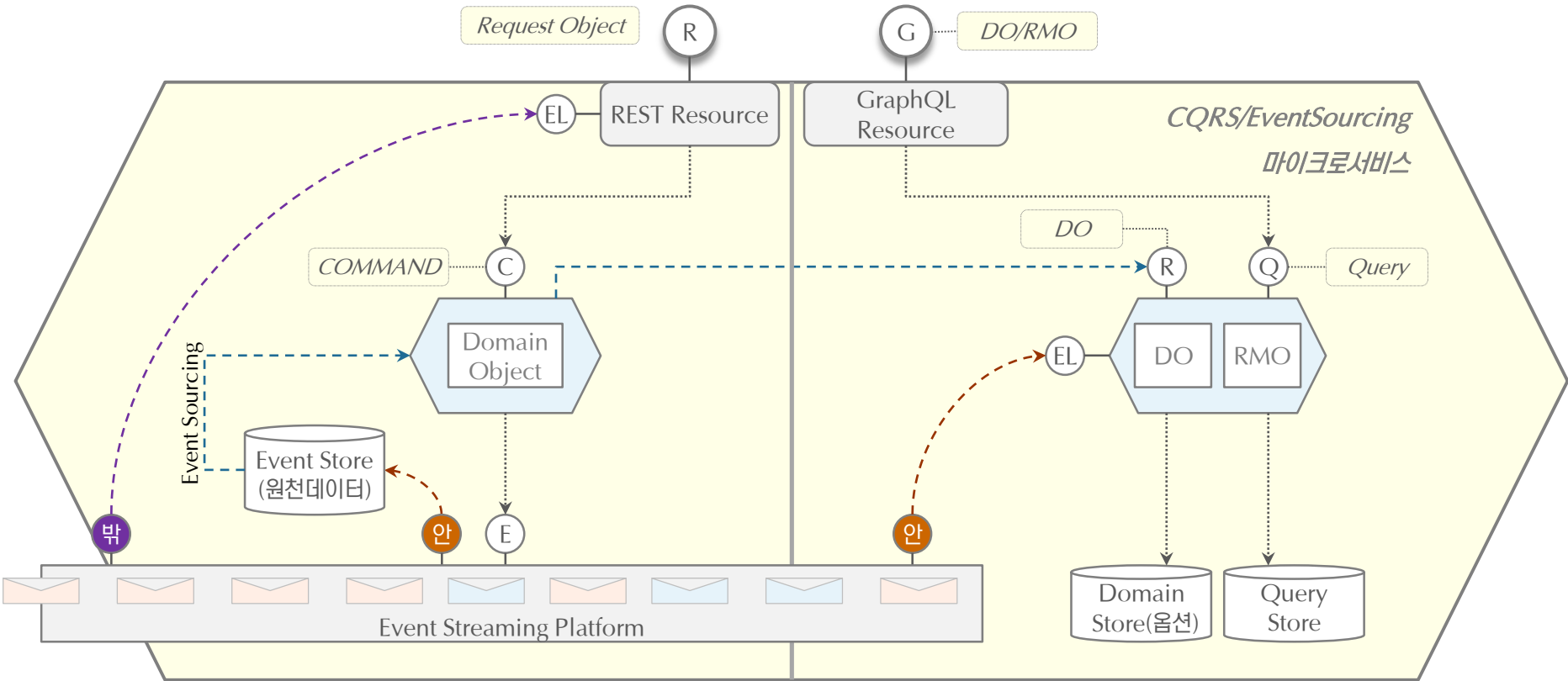
- ✓ 도메인 이벤트는 특정 마이크로서비스 또는 Reactive 서비스에서 발행하며, 전사 범위를 가집니다.
- ✓ 모든 마이크로서비스는 공유를 목적으로 별도의 “마이크로서비스 이름-event”라는 모듈을 별도로 구성하여야 합니다.
- ✓ 도메인 이벤트는 도메인의 First class citizen 입니다. 따라서 신중한 설계와 체계적인 공유가 필요합니다.
- ✓ 도메인 안에서는 “*-event” 모듈을 import하여 이벤트에 접근합니다. 도메인 밖에서는 스키마 리파지토리를 사용합니다.



→ 도메인 이벤트는 전사 범위를 가져야...

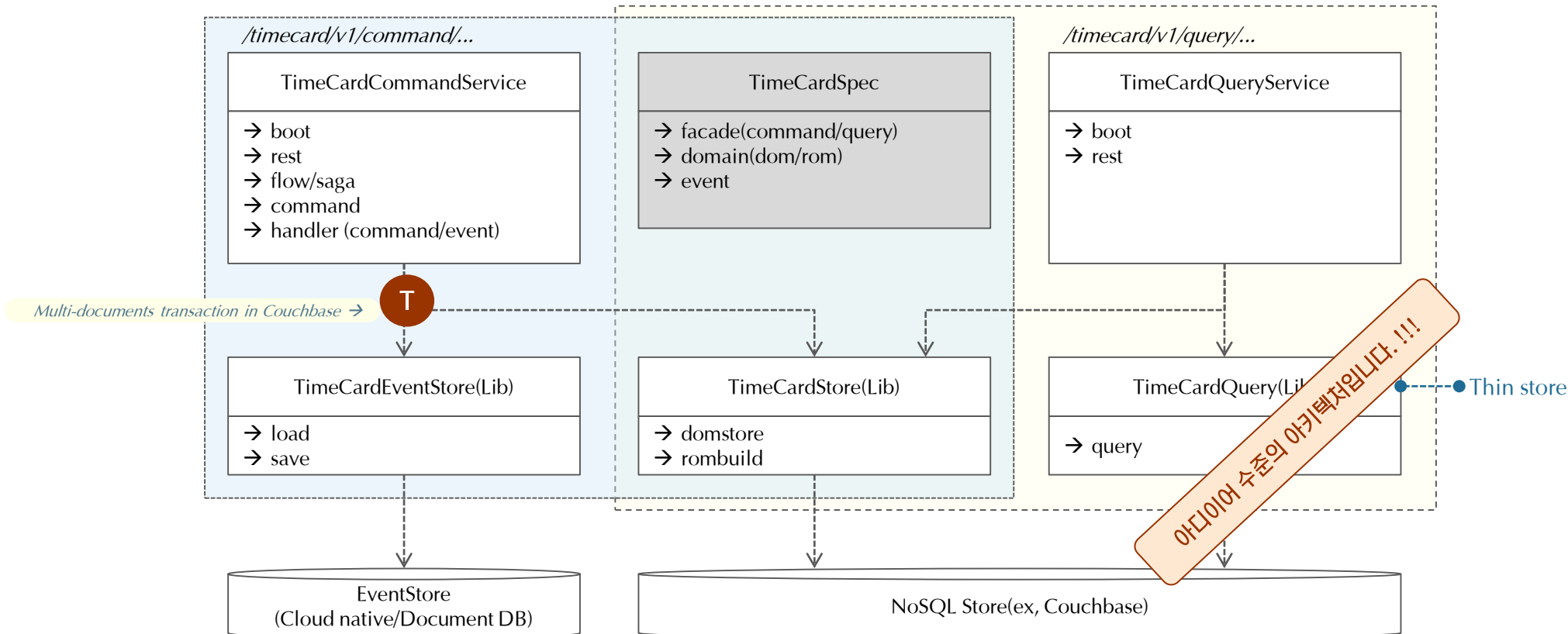
0.3 기술 적용(3/7) – Event Sourcing/CQRS

- ✓ 쓰기(CUD)와 읽기를 나누어 설계할 때 CQRS(Command Query Responsibility Segregation) 패턴을 적용합니다.
- ✓ 쓰기 영역에서는 모든 요청은 Command이며, Command는 도메인 객체에 변화를 가져옵니다.
- ✓ 도메인의 변화는 “이벤트”를 유발합니다. 이벤트를 쌓아서 도메인 상태를 유지하는 방식을 Event Sourcing이라고 합니다.
- ✓ 읽기 영역은 별도의 모델(읽기 전용 모델)을 갖고 있으며, 이 모델은 Query에 최적화되어 있습니다.



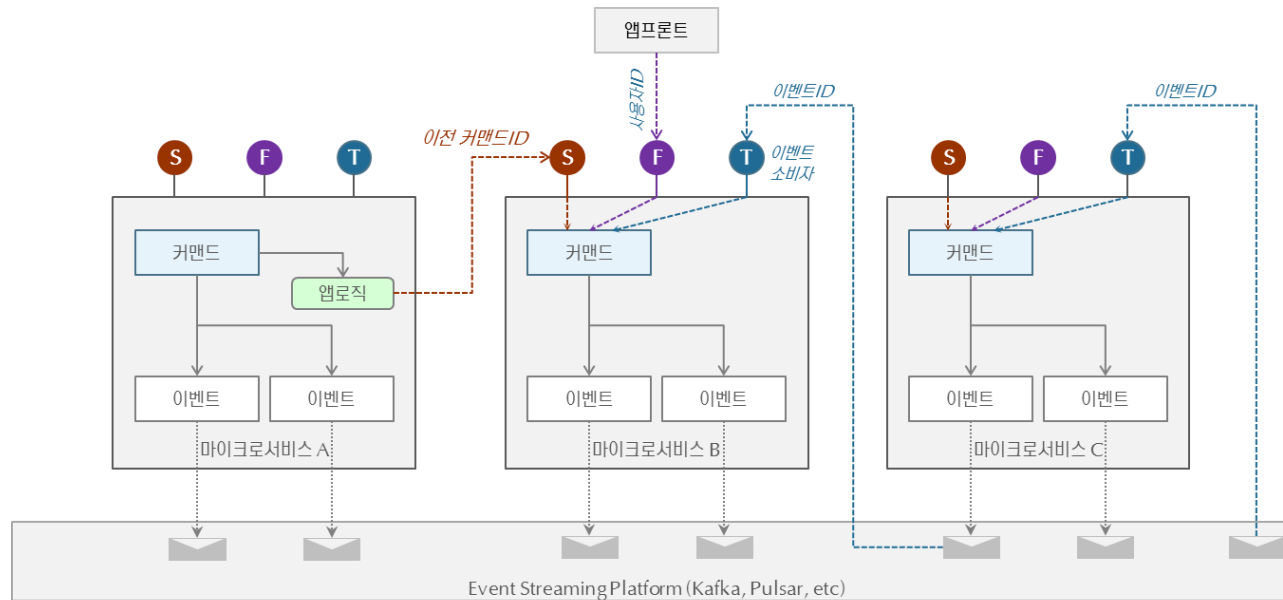
0.4 기술 적용(6/7) – 일치성/추적성 1

- ✓ CQRS는 이벤트 메시지를 기반 처리로 일치성에 대한 보장을 권하지만, 현실은 트랜잭션을 요구합니다.
- ✓ 아이디어 → TimeCardEventStore 호출과 TimeCardStore 호출을 하나의 트랜잭션으로 묶어서 처리합니다.
- ✓ 아이디어 → 이벤트를 던지지 않고 받은 CommandService가 도메인 엔티티/쿼리 엔티티를 저장 합니다.
- ✓ 저장 처리를 완전히 마무리 한 후, 이벤트 메시지를 발행합니다. ← 설계 포인트 !! ← nara.daysman 역할



0.5 기술 적용(7/7) – 일치성/추적성 2

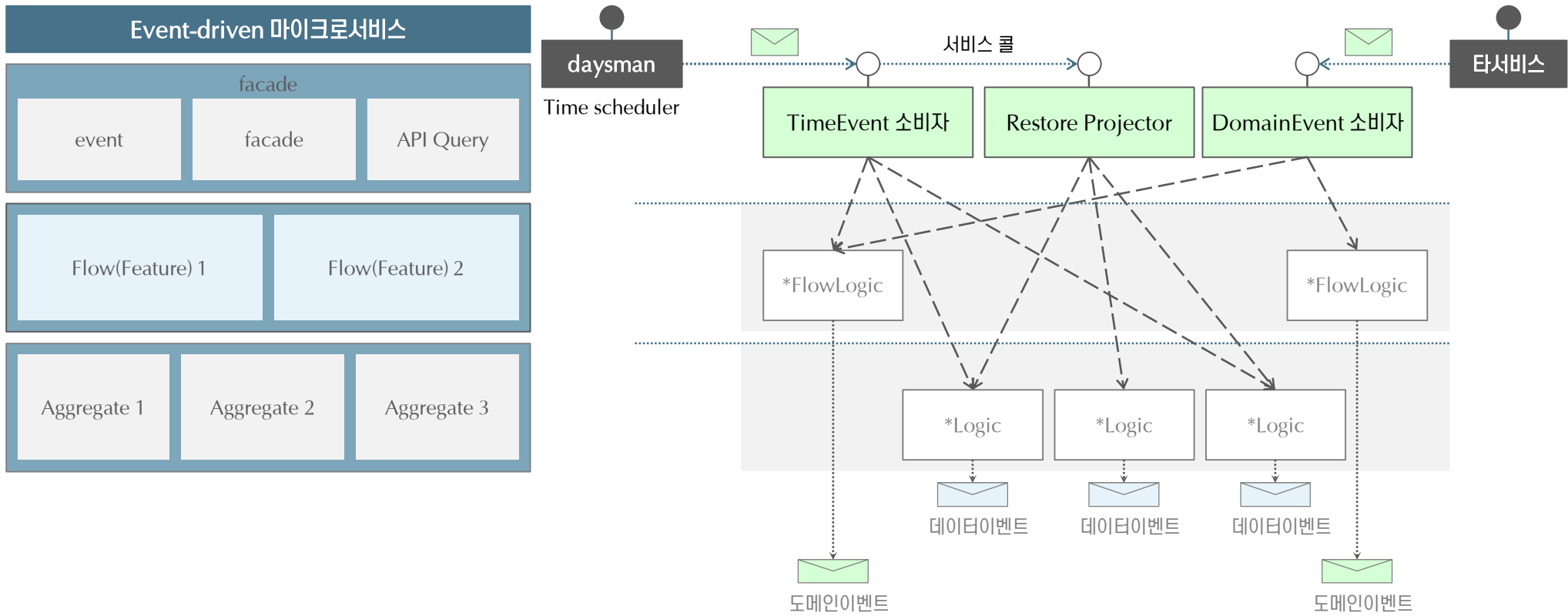
- ✓ 이벤트 소싱과 커맨드 소싱을 모두 할 경우, 두 정보를 이용하여 업무 처리 수준의 추적성을 확보할 수 있습니다.
- ✓ NARA.Accent.cqrs는 CQRS에 대한 Key concept들을 정의하고 추적에 필요한 정보를 미리 정의하여 두었습니다.
- ✓ 모든 커맨드와 이벤트 정보를 이벤트 스트림 형식으로 이벤트 레이크로 보내서 분석과 리포팅에 사용합니다.
- ✓ NARA.AirView는 리포트를, X-Ray는 커맨드와 이벤트, 다시 커맨드로 이어지는 Call-Chain을 분석하여 보여 줍니다.



[로그 수집이 더 이상 필요없는 Command/Query/Event 체인 기반 추적 메커니즘]

0.6 Facade 모듈 구조(5/5) – Event consumer

- ✓ 서비스 외부로부터 발생한 이벤트를 받아서 처리하는 일은 Facade의 중요한 역할입니다.
- ✓ Daysman이 타임 스케줄에 근거하여 발행하는 시간 이벤트를 받아서 처리합니다. → TimeEvent 소비자
- ✓ 타 서비스에서 발행한 도메인 이벤트를 받아서 처리합니다. → DomainEvent 소비자
- ✓ Projection을 통한 DB Restore 실행 시, Daysman 요청을 받아서 처리합니다. → Restore projector 서비스 호출





1. 이벤트의 이해

posco ict

1.1 이벤트란?

1.2 Command와 Event

1.3 Command and Query Responsibility Segregation

1.1 이벤트란? (1/2)

- ✓ 전통적인 접근 방식의 트랜잭션 사용 사례는 데이터베이스에 데이터를 유지하는 것과 관련됩니다.
- ✓ 변경이 수행되면 데이터베이스의 데이터가 새 상태와 일치하도록 업데이트됩니다.
- ✓ 이것은 아주 잘 작동하지만 분명하지 않은 것은 모델을 업데이트하여 최종 상태를 가지고 중간 정보를 잃고 있다는 것입니다.
- ✓ 예를 들어, 고객이 카트를 변경하기 전에 카트에 무엇을 넣었는지 또는 내용이 변경된 시점을 알 수 있는 방법이 없습니다.
- ✓ 일단 이 길을 따라 가면 도메인 모델이 비즈니스가 아닌 요소로 오염되기 쉽고 소프트웨어가 빠르게 복잡해집니다.

ID	SEQ		
001	1		
002	1		
001	2		
003	1		
003	2		
003	3		

최종 상태와 이전 상태가 혼재된 모델

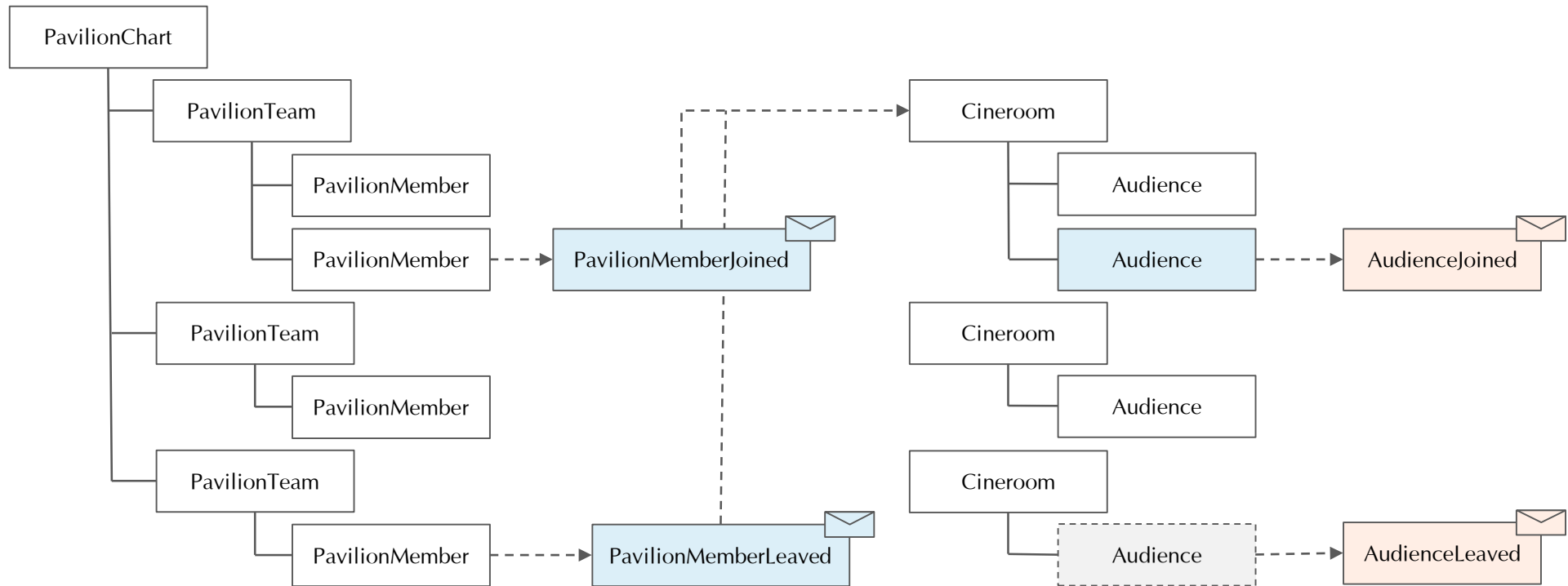
ID		
001		
002		
003		
004		
005		
006		

최종 상태와 이전 상태를 분리한 모델(좌측이 최종 상태)

ID	SEQ	
001	1	
002	1	
001	2	
003	1	
003	2	
003	3	

1.1 이벤트란? (2/2)

- ✓ 도메인 이벤트는 도메인에 영향을 준 과거에 발생한 사실입니다. 따라서, 도메인 이벤트는 불변입니다.
- ✓ 한 Aggregate에서 발생한 이벤트에 관심이 있는 다른 Aggregate는 이 이벤트에 반응합니다.
- ✓ 일부 도메인 이벤트의 경우 기록할 가치는 충분하지만 이 이벤트가 다른 반응을 일으키지는 않습니다.
- ✓ 우리가 역사를 바꿀 수 없듯이 이러한 실현을 소프트웨어 설계에 적용하면 유용한 시스템을 개발할 수 있습니다.



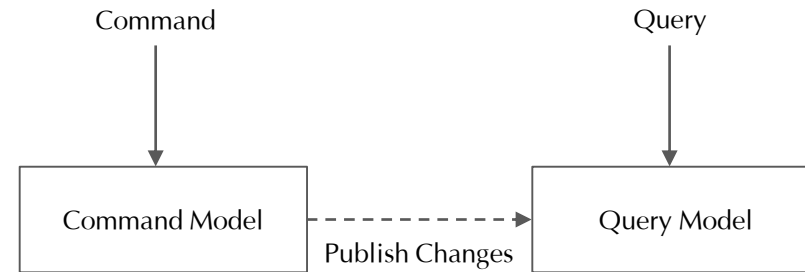
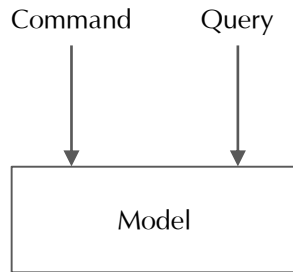
1.2 Command와 Event

- ✓ Command와 Event는 모두 읽기 위한 데이터만 포함하는 단순한 도메인 구조로 동작/비즈니스 논리를 포함하지 않습니다.
- ✓ Command는 명령 처리기에 의해 처리되는 상태 변경을 위해 도메인 또는 도메인 로직으로 전달되는 객체입니다.
- ✓ Event는 도메인 상태가 변경된 사항에 대한 사실에 대한 설명으로 이벤트는 과거이므로 과거형 단어를 포함합니다.
- ✓ 이벤트는 명령을 처리한 결과로 발생하고 일반적으로 이벤트와 1:1 관계를 가지지만 여러 이벤트를 생성하기도 합니다.



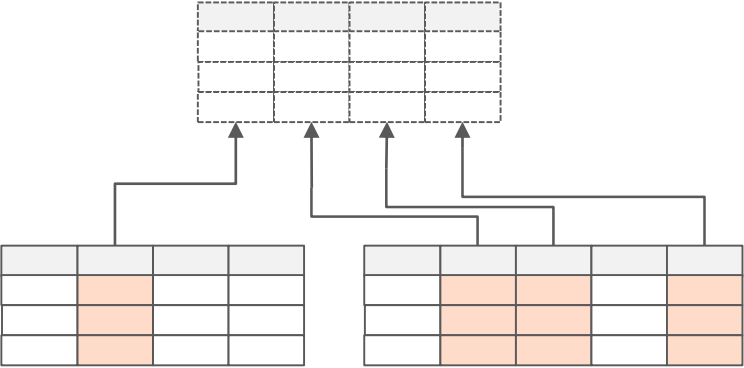
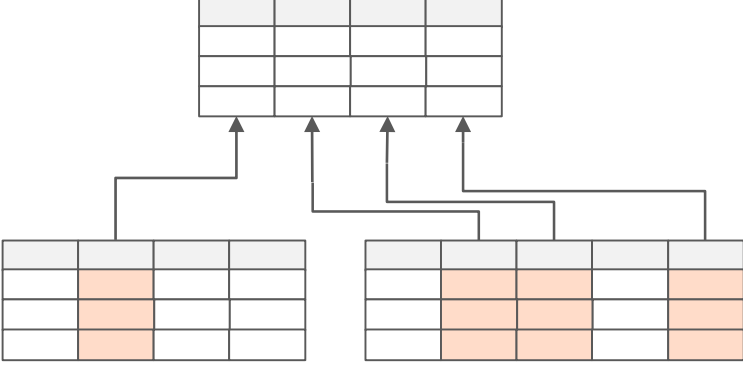
1.3 Command and Query Responsibility Segregation (1/3)

- ✓ CQRS는 Bertrand Meyer의 CQS 원칙으로부터 발전한 개념입니다.
- ✓ Command는 상태를 변경하고 Query는 정보를 반환하며 Query는 사이드 이펙트를 발생시키지 않습니다.
- ✓ CQS가 코드 레벨의 책임인 반면에 Greg Young의 CQRS 패턴은 Command와 Query의 단순한 분리입니다.
- ✓ CQRS는 조회에 대한 성능과 개발에 대한 단순함을 유지할 수 있는 좋은 아키텍처 패턴입니다.



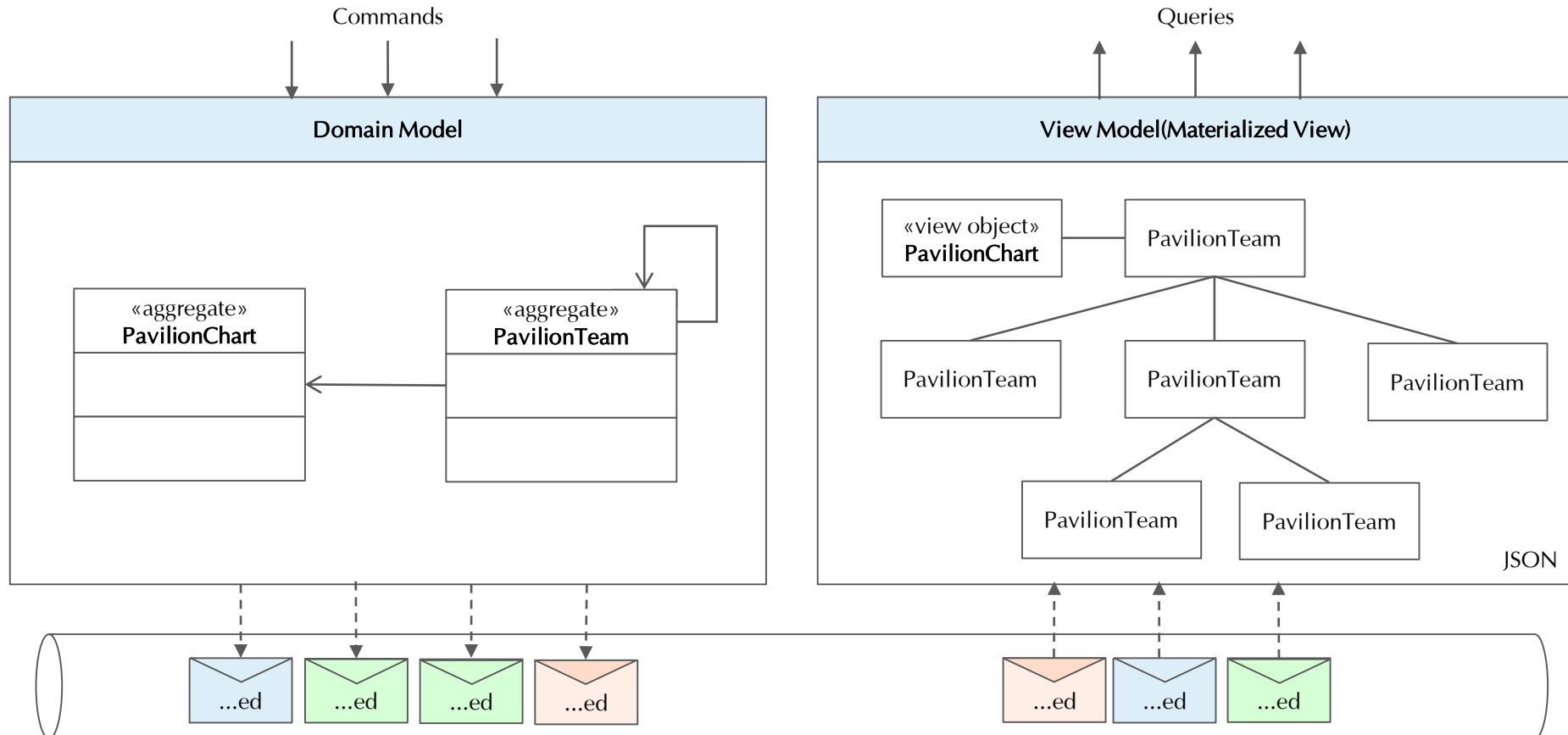
1.3 Command and Query Responsibility Segregation (2/3)

- ✓ View는 사용자에게 접근이 허용된 자료를 제한적으로 보여주기 위해 기본 테이블에서 유도된 가상 테이블입니다.
- ✓ View가 논리적인 테이블인 반면 Materialized View는 물리적으로 존재하는 테이블입니다.
- ✓ Materialized View는 복잡한 조인과 같이 높은 비용의 Query 성능 향상을 목적으로 사용합니다.
- ✓ 집계 함수(SUM, MIN, MAX, AVG 등)가 많은 Query의 결과를 미리 별도의 테이블로 생성합니다.

View	Materialized View
	
<ul style="list-style-type: none">✓ View는 명명된 Query로 어떤것도 저장하지 않는다.✓ View에 질의하면 View의 정의에 기반하여 실제 테이블의 데이터를 조회한다.✓ SELECT Query보다 느리다.	<ul style="list-style-type: none">✓ Materialized View는 물리적으로 데이터를 저장하고 주기적으로 갱신한다.✓ Materialized View에 질의하면 Materialized View의 데이터를 반환한다.✓ SELECT Query를 위한 빠른 성능을 제공한다.

1.3 Command and Query Responsibility Segregation (3/3)

- ✓ Materialized View는 조회에 필요한 속성을 데이터 모델에 최적화 시키는 용도로 주로 사용됩니다.
- ✓ 도메인 모델(Aggregate)에서 발행하는 이벤트에서 조회 전용 Materialized View가 필요한 케이스도 있습니다.
- ✓ PavilionChart/PavilionTeam Aggregate에서 발행하는 도메인 이벤트로부터 조회 전용 Tree 구조를 생성합니다.





2. 이벤트 발행과 소비

posco ict

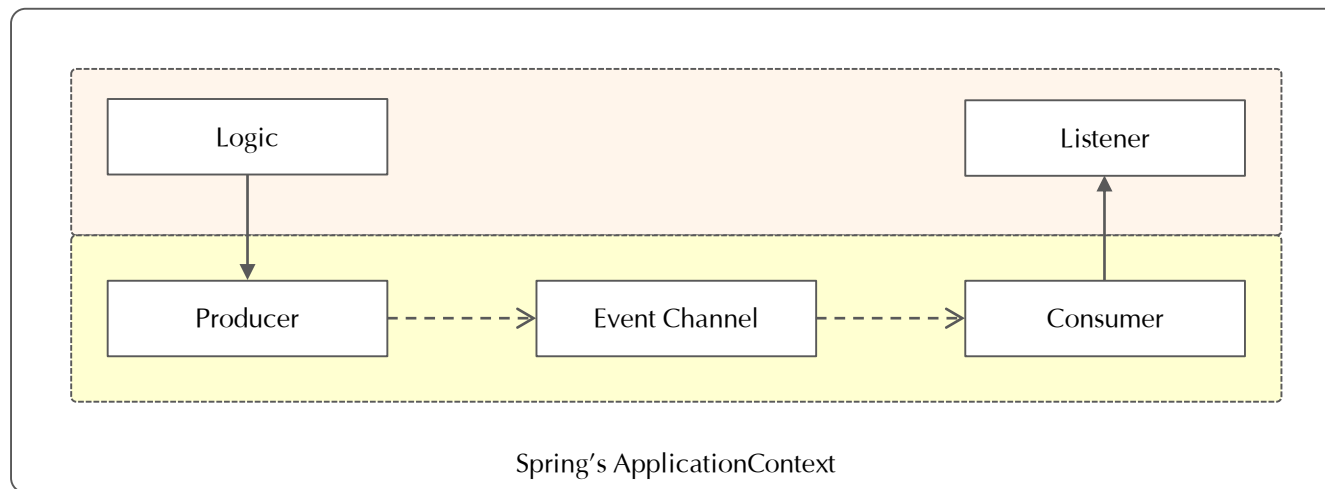
2.1 Spring과 이벤트

2.2 이벤트 발행과 처리

2.3 이벤트 발행과 트랜잭션

2.1 Spring과 이벤트

- ✓ Spring은 이벤트 발행과 소비를 사용할 수 있는 메커니즘을 제공합니다.
- ✓ 이벤트가 전송되는 채널은 Spring의 ApplicationContext입니다.
- ✓ 이벤트 발행은 ApplicationEventPublisher.publishEvent 오퍼레이션을 소비는 @EventListener를 사용합니다.
- ✓ Spring은 publishEvent로 발행된 이벤트의 타입을 보고 @EventListener의 메소드를 호출합니다.



2.2 이벤트 발행과 처리

✓ Spring에서 이벤트 발행과 소비 사례입니다.

이벤트 발행

```
@Service
public class Logic {
    //
    private final ApplicationEventPublisher eventPublisher;

    public Logic(ApplicationEventPublisher eventPublisher) {
        //
        this.eventPublisher = eventPublisher;
    }

    public void doSomething(Message message) {
        //
        // do something
        this.eventPublisher.publishEvent(message);
    }
}
```

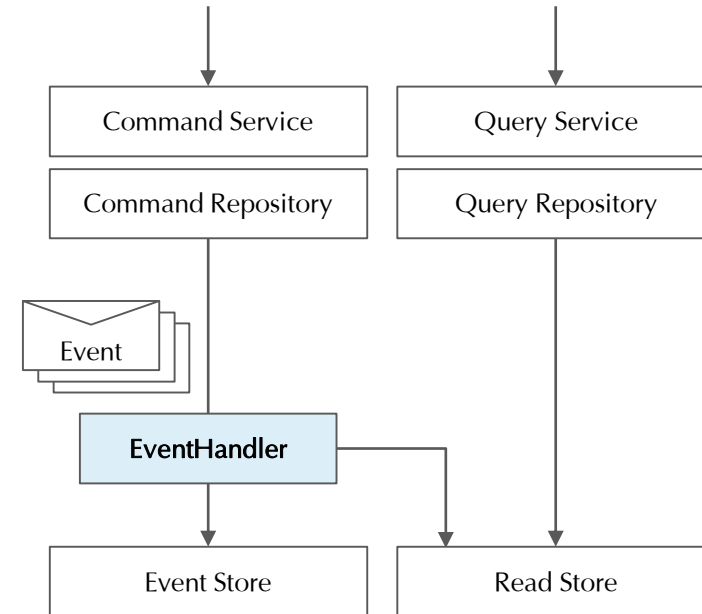
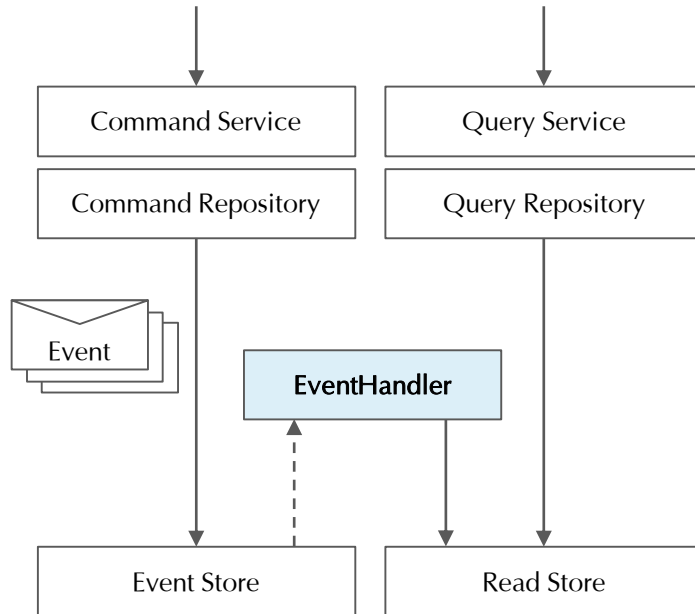
이벤트 소비

```
@Component
public class EventHandler {
    //
    private static final Logger logger = ...

    @EventListener
    public void on(Message message) {
        //
        logger.info("");
        logger.info("@EventListener: {}", message.getName());
        logger.info("");
    }
}
```

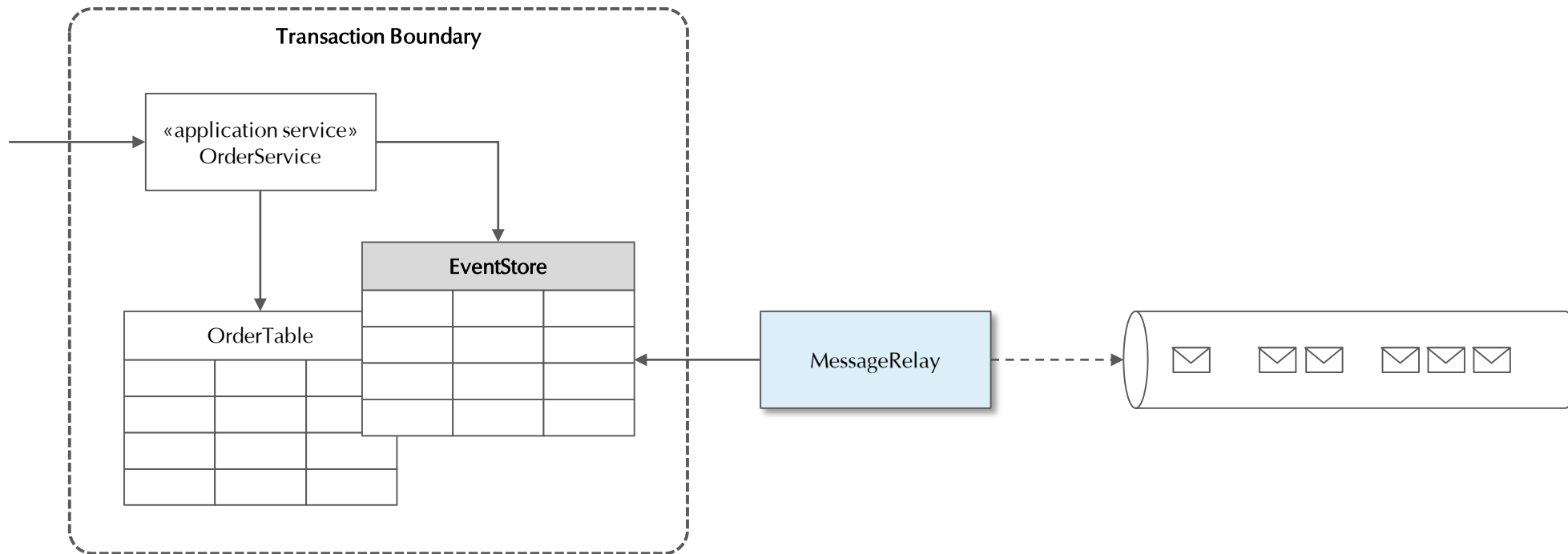
2.3 이벤트 발행과 트랜잭션 (1/2)

- ✓ CQRS는 비즈니스 로직이 적용되는 처리(Command)와 단순 조회(Query)를 분리합니다.
- ✓ 조회 전용 View를 생성/변경하는 EventHandler는 원칙적으로 비동기입니다.
- ✓ Consistency 보장이 필요하면 EventHandler의 책임을 단일 트랜잭션에서 처리되도록 조정합니다.



2.3 이벤트 발행과 트랜잭션 (2/2)

- ✓ 마이크로서비스는 데이터베이스에 변화를 기록하고 메시지/이벤트를 발행해야 합니다.
- ✓ 보상 트랜잭션에 참여하는 서비스는 데이터베이스에 변화를 기록하는 것과 메시지/이벤트 발행이 원자적이어야 합니다.
- ✓ 트랜잭션을 제공하는 저장소를 이용하여 데이터 변화와 메시지/이벤트를 원자적으로 처리할 수 있습니다.
- ✓ MessageRelay는 이벤트 스토어를 주기적으로 조회하므로 이벤트 전달에 추가 지연이 발생합니다.
- ✓ MessageRelay는 이벤트를 두 번 이상 발행할 수 있지만 Idempotent를 적용하여 해결할 수 있습니다.





3. 마이크로 서비스 설계

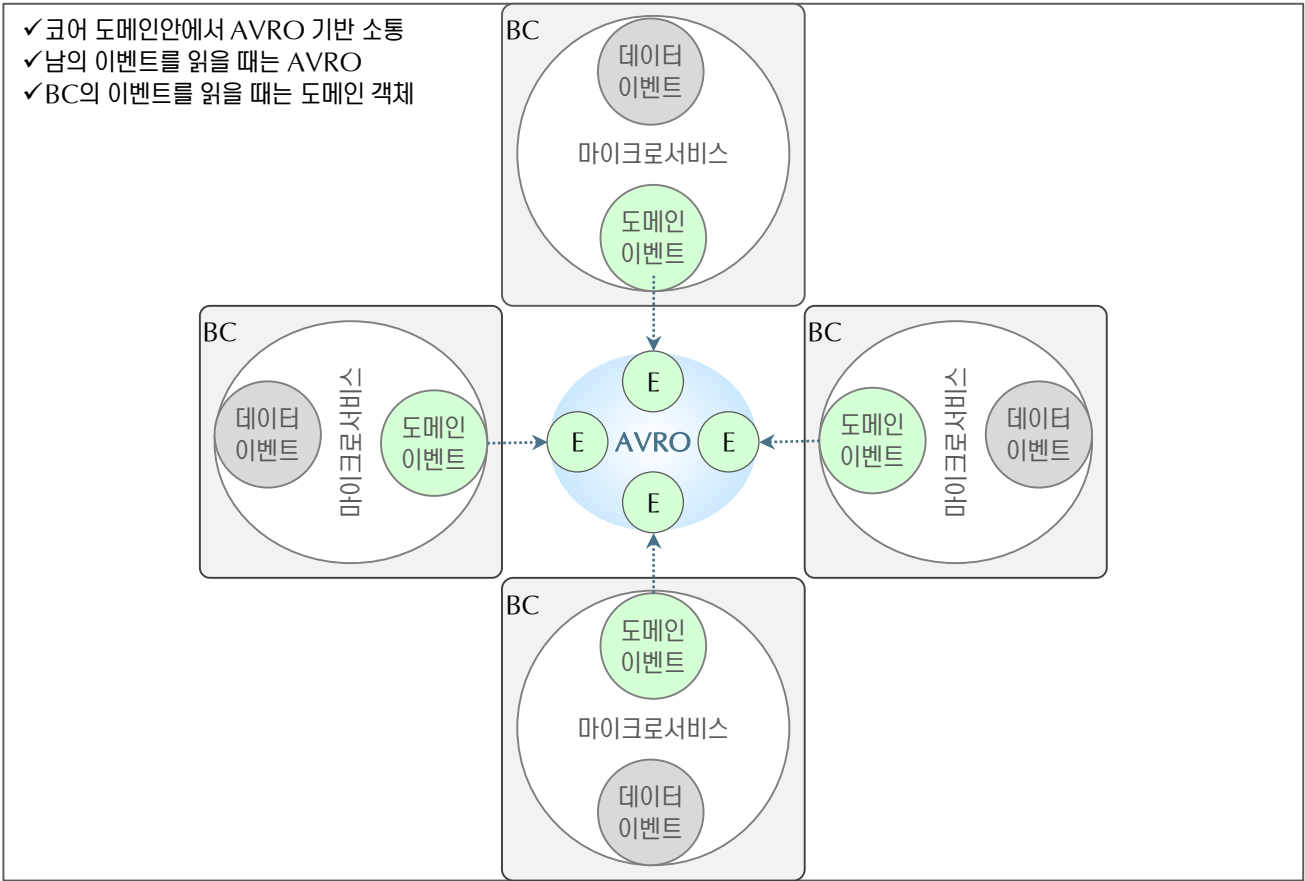
posco ict

3.1 이벤트 객체 공유

3.2 이벤트와 UI

3.1 이벤트 객체 고유

- ✓ 데이터 이벤트는 도메인 객체를 포함하고 있습니다. 데이터 이벤트의 오픈 범위는 마이크로서비스 내부로 국한합니다.
- ✓ 도메인 이벤트는 외부로 전달되는 이벤트로, 이벤트를 소비하는 누구나 읽을 수 있어야 합니다. 이를 위한 장치가 필요합니다.
- ✓ 도메인 객체는 가급적 Primitive 타입으로 정의를 하며, AVRO 를 이용하여 스키마 저장소에 정의하여 둘 수도 있습니다.
- ✓ 1차 스키마 저장소는 daysman 이며, 2차 스키마 저장소는 confluent의 Schema registry가 될 수 있습니다.





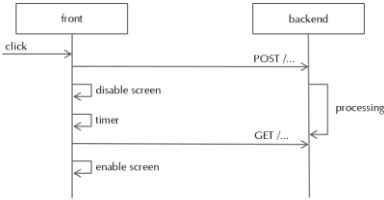
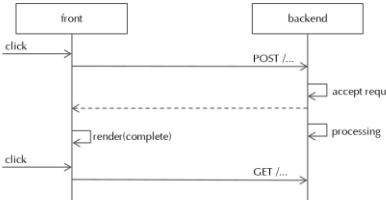
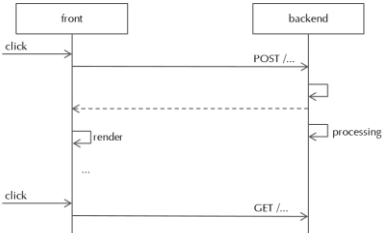
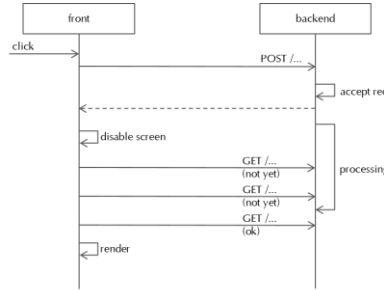
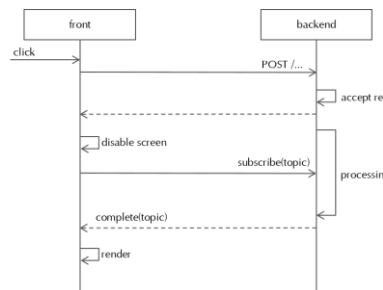
4. 마이크로 서비스 설계 - 이벤트와 UI

posco ict

- 4.1 이벤트와 UI
- 4.2 Deactivation
- 4.3 Confirm UI
- 4.4 Pre-Rendering
- 4.5 Polling
- 4.6 Pushing

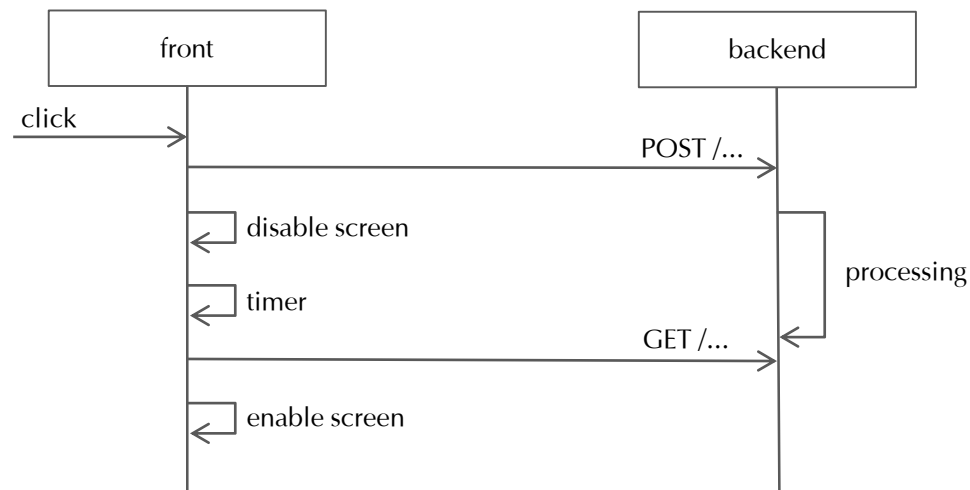
4.1 이벤트와 UI

- ✓ MSA를 이야기하면 backend를 주로 다루며 front에 대해서는 크게 다루지 않습니다.
- ✓ 서비스간 메시지(비동기)를 이용하여 통신하는 경우 사용자의 요청-응답이 비동기가 될 수 밖에 없습니다.
- ✓ front는 요청을 보내고 서버로부터 응답을 받았지만 비즈니스 처리는 아직 완료되지 않았을 수 있습니다.
- ✓ 사용자는 UI의 표현이 실제 처리 완료로 인지하므로 Eventual Consistency와 UI의 관계도 고려해야 합니다.

Deactivation	Confirm UI	Pre rendering	polling	push
				
<div>✓ 기술 난이도: 하</div> <div>✓ 비즈니스 영향도: 상</div>	<div>✓ 기술 난이도: 하</div> <div>✓ 비즈니스 영향도: 상</div>	<div>✓ 기술 난이도: 중</div> <div>✓ 비즈니스 영향도: 중</div>	<div>✓ 기술 난이도: 중</div> <div>✓ 비즈니스 영향도: 하</div>	<div>✓ 기술 난이도: 상</div> <div>✓ 비즈니스 영향도: 하</div>

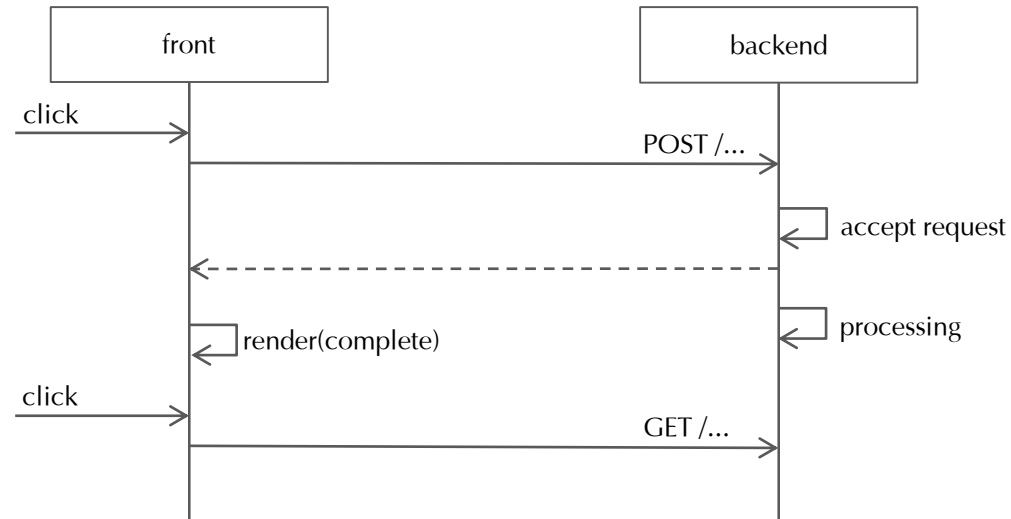
4.2 Deactivation

- ✓ 사용자가 요청을 하면 화면을 비활성화하고 명령의 완료 여부를 표시합니다.
- ✓ 그런 다음 지정된 시간 동안 대기 후 화면을 다시 로드하여 처리 결과를 업데이트합니다.
- ✓ 이 방법은 지정된 시간동안 처리가 완료되지 못했을 수 있으므로 사용자에게 올바른 결과를 보여주지 못할 수 있습니다.
- ✓ 이 접근법은 개발하기 쉽지만 사용자 경험을 트레이드-오프해야 할 수 있습니다.



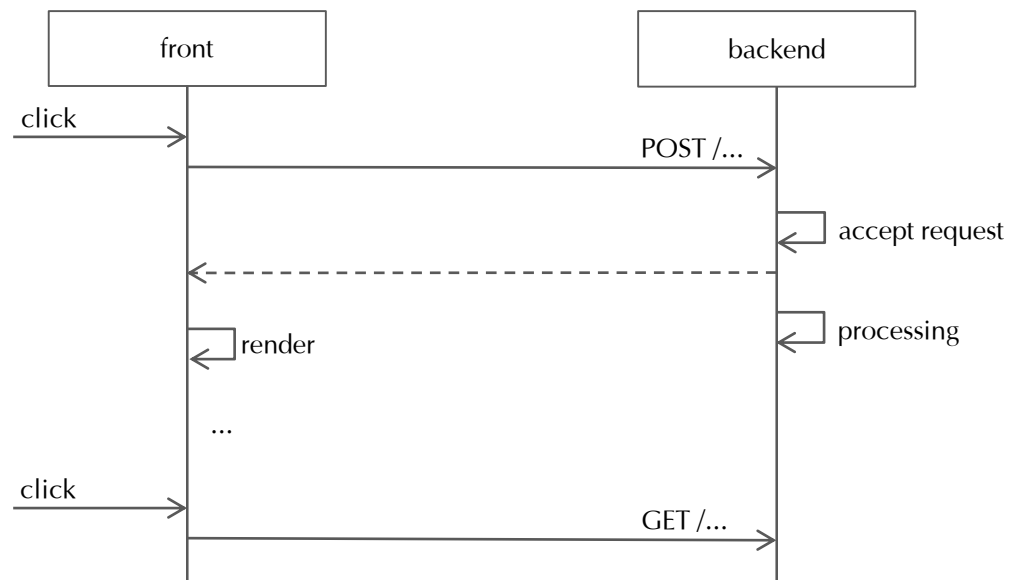
4.3 Confirm UI

- ✓ 사용자에게 '확인' 또는 '감사합니다' 화면을 먼저 표시합니다.
- ✓ 일관성에 대한 일반적인 지연 시간이 주어지고 사용자가 다른 화면으로 이동할 때 최신 상태가 표시됩니다.
- ✓ 이 방법은 판매 완료 또는 티켓 예매와 같은 프로세스 완료 화면(완료되었습니다 -> 예매내역보기 -> 예매내역)에 적합합니다.



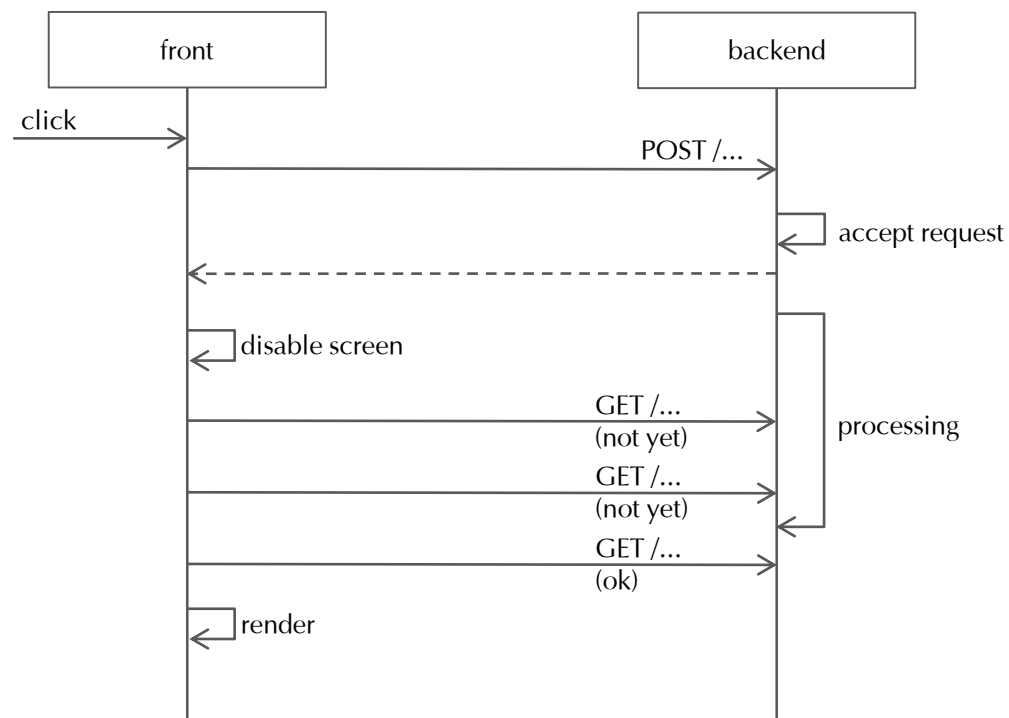
4.4 Pre-Rendering

- ✓ 클라이언트는 명령이 접수되었다는 확인을 받으면 새로운 읽기 모델 상태(성공시 어떻게 표시할지)를 가정 할 수 있습니다.
- ✓ 결과적으로 클라이언트는 뷰의 변경을 예측할 수 있으므로 이 방법을 사용하려면 버전을 관리하는 방법도 함께 고려해야 합니다.
- ✓ 웹 UI는 자바 스크립트의 `promise`를 이용하여 이 프로세스를 비교적 손쉽게 처리할 수 있습니다.
- ✓ 이 방법은 개발하는데 시간이 더 걸리지만 훨씬 자연스러운 사용자 경험을 제공합니다.



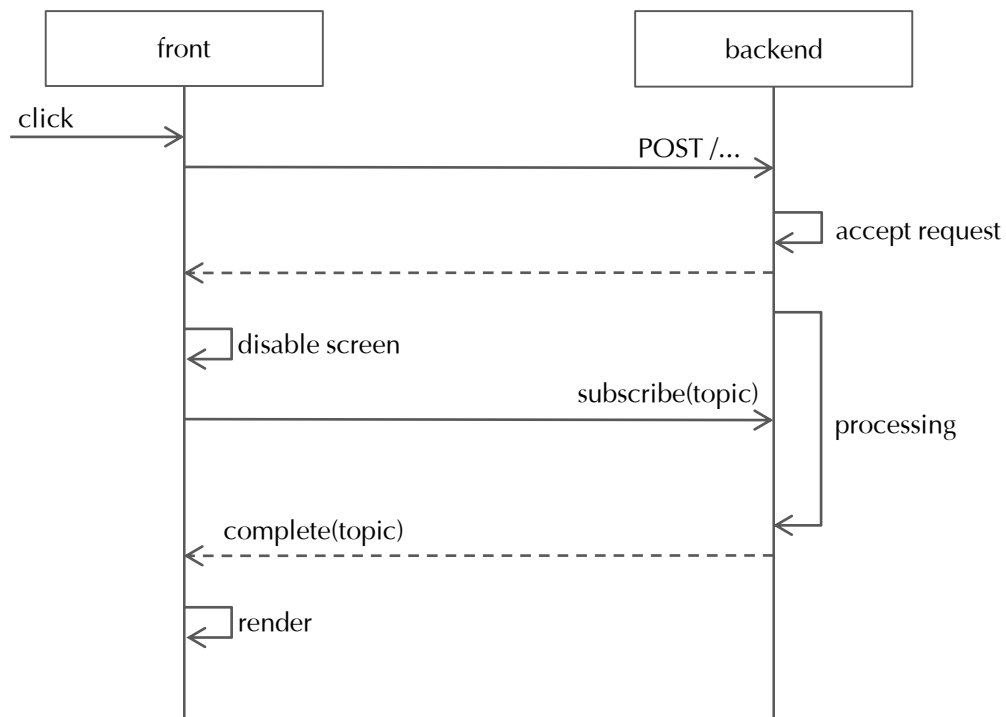
4.5 Polling

- ✓ 화면을 비활성화 시키는 방법은 동일하지만 UI에서 주기적으로 처리 완료 여부를 체크합니다.
- ✓ 주기적 체크에 대한 응답이 처리중이면 화면을 그대로 유지합니다.
- ✓ 주기적 체크에 대한 응답이 완료일 때 화면을 활성화 시키거나 지정된 페이지로 이동(렌더링 포함)합니다.



4.6 Pushing

- ✓ SPA에서 요청을 보내고 기대하는 이벤트를 받을 때까지 화면을 비활성화시킵니다.
- ✓ 서버에서 처리가 완료되면 화면을 활성화 시키거나 지정된 페이지로 이동(렌더링 포함)합니다.
- ✓ 이벤트 방식은 이벤트와 관련된 개발 오버 헤드(복잡도, 복잡도로 인한 개발 시간)가 존재하므로 신중히 접근해야 합니다.
- ✓ WebSocket, SSE(Server-Sent Event)중 선택할 수 있고 실시간 업데이트가 실제로 필요한지 확인하는 것 또한 중요합니다.





4. 실습 환경 구성

posco ict

- 4.1 Kafka
- 4.2 Zookeeper/Kafka 설치
- 4.3 주요 용어
- 4.4 CLI


4.1 Kafka

- ✓ 아파치 카프카(Apache Kafka)는 아파치 소프트웨어 재단이 스칼라로 개발한 오픈 소스 메시지 브로커 프로젝트입니다.
- ✓ 이 프로젝트 목표는 실시간 데이터 피드를 관리하기 위해 통일된, 높은 처리량, 낮은 지연시간을 지닌 플랫폼을 제공합니다.
- ✓ 분산 트랜잭션 로그로 구성되어 상당히 확장 가능한 pub/sub 정의가 가능합니다.
- ✓ 스트리밍 데이터를 처리하기 위한 기업 인프라를 제공합니다.


APACHE KAFKA

*More than **80% of all Fortune 100 companies** trust, and use Kafka.*


Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.




10 OUT OF 10
MANUFACTURING



7 OUT OF 10
BANKS



10 OUT OF 10
INSURANCE



8 OUT OF 10
TELECOM

<http://kafka.apache.org/>

✓ <http://kafka.apache.org/downloads>

DOWNLOAD

2.7.0 is the latest release. The current stable version is 2.7.0.

You can verify your download by following these [procedures](#) and using these [KEYS](#).

2.7.0

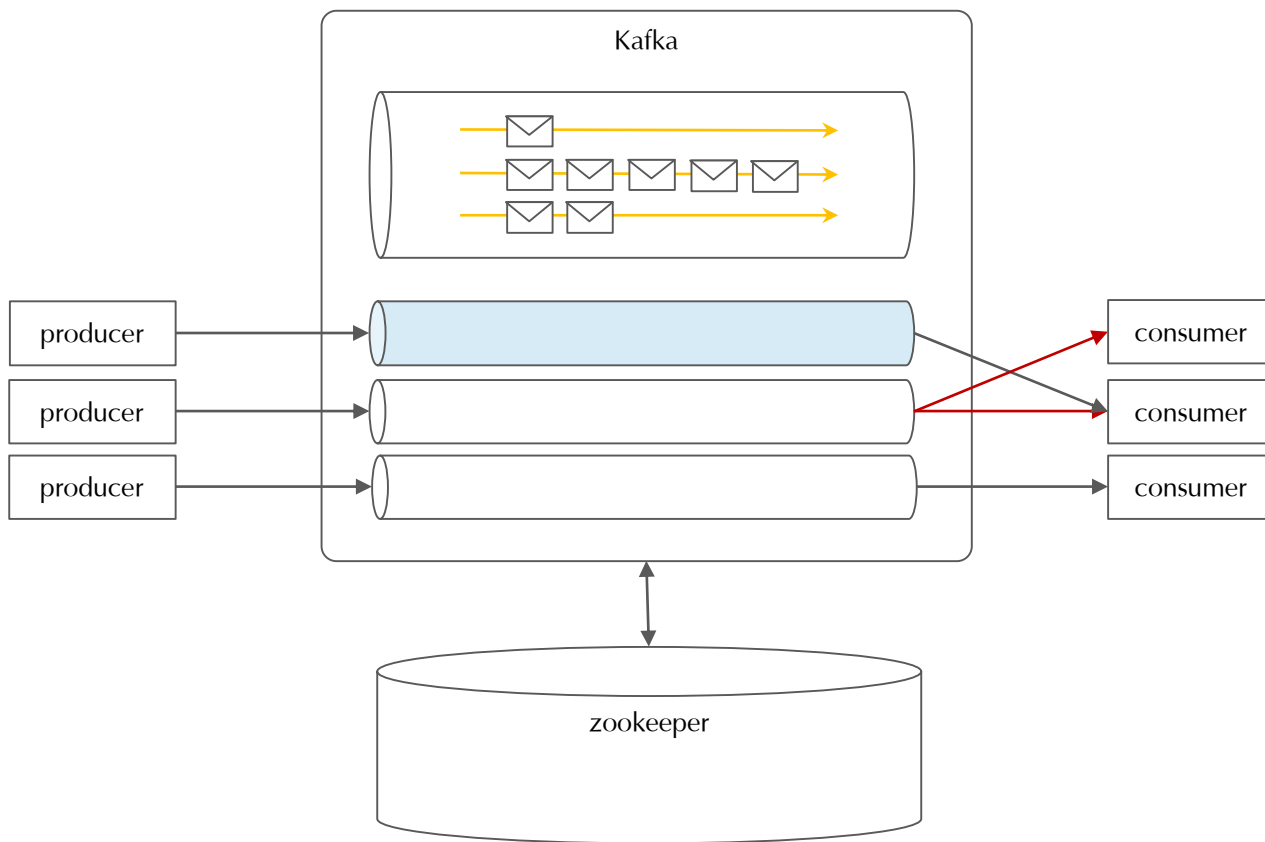
- Released Dec 21, 2020
- [Release Notes](#)
- Source download: [kafka-2.7.0-src.tgz](#) ([asc](#), [sha512](#))
- Binary downloads:
 - Scala 2.12 - [kafka_2.12-2.7.0.tgz](#) ([asc](#), [sha512](#))
 - Scala 2.13 - [kafka_2.13-2.7.0.tgz](#) ([asc](#), [sha512](#))

We build for multiple versions of Scala. This only matters if you are using Scala and you want a version built for the same Scala version you use. Otherwise any version should work (2.13 is recommended).

Kafka 2.7.0 includes a number of significant new features. Here is a summary of some notable changes:

4.3 주요 용어

- ✓ 여러 이벤트를 의미상 하나로 묶고 메시지를 주고 받는 단위를 [Topic]이라고 합니다.
- ✓ 하나의 Topic을 더 세분화 시킨 단위를 [Partition]이라고 합니다.
- ✓ Zookeeper는 Kafka의 Topic, Partition, Consumer Group/Offset을 기록하는 저장소입니다.
- ✓ Kafka를 통해 주고 받은 Event는 Zookeeper에 기록됩니다.



4.4 CLI

- ✓ Command Line Interface를 이용하여 토픽 생성, 이벤트 발행, 이벤트 소비를 실습합니다.
- ✓ 주요 명령어는 \$KAFKA_HOME/bin에 있습니다.

토픽 목록	> kafka-topics.sh --list --zookeeper localhost:2181
토픽 생성	> kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test
토픽 삭제	> kafka-topics.sh --delete --bootstrap-server localhost:9092 --topic test
이벤트 발행	> kafka-console-producer.sh --broker-list localhost:9092 --topic test
이벤트 소비	> kafka-console-consumer.sh --bootstrap-server localhost:9092 --from-beginning --topic test



5. 이벤트 발행과 소비 (Spring Cloud/Event-Driven)

posco ict

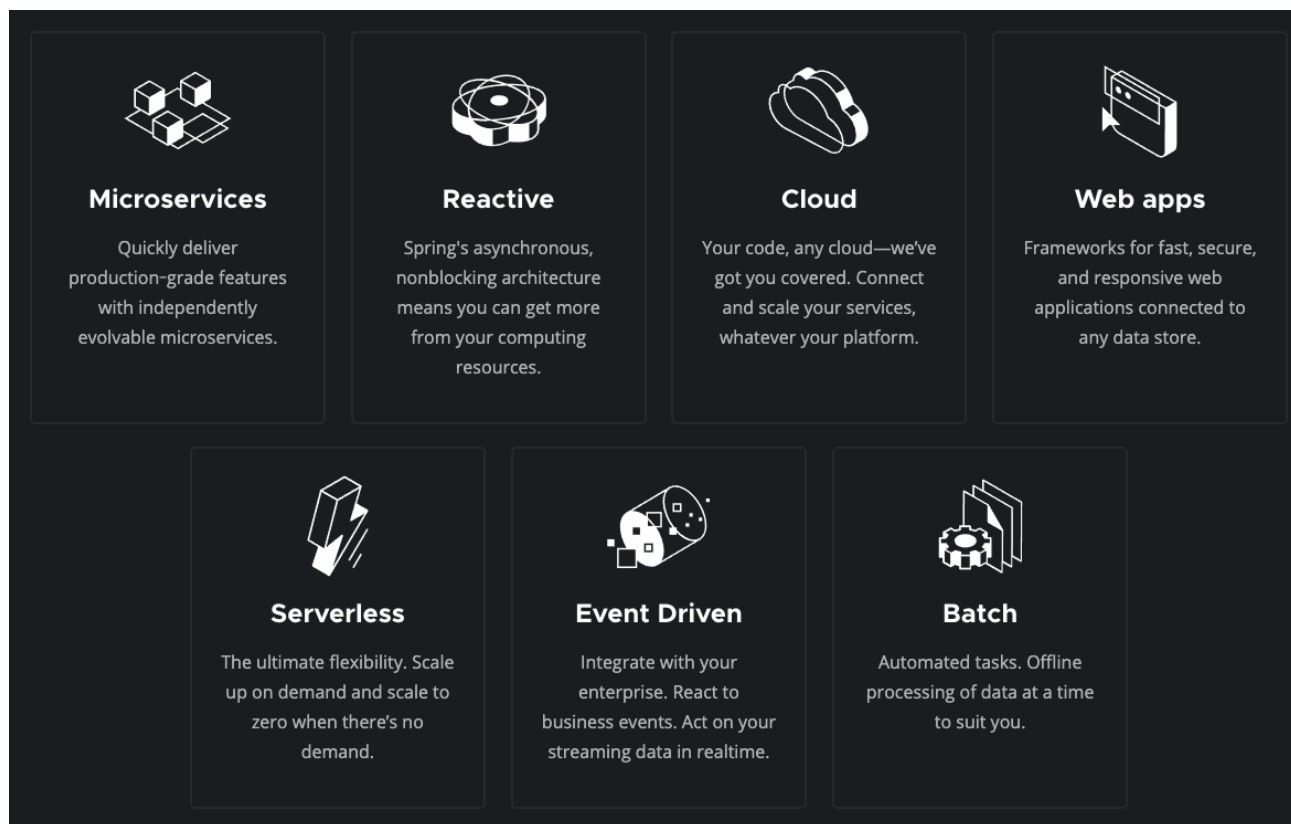
5.1 Spring Cloud/Event-Driven

5.2 (실습) Event Publishing

5.3 (실습) Event Consuming

5.1 Spring Cloud/Event-Driven

- ✓ Spring Cloud 프로젝트 제품군에는 애플리케이션을 클라우드에서 실행하는 데 필요한 많은 서비스를 포함합니다.
- ✓ Spring Event Driven은 이벤트를 통한 통합 및 스트리밍, 데이터의 흐름에 관한 다양한 이벤트 기반 옵션을 제공합니다.
- ✓ 이벤트 기반 마이크로서비스, 스트리밍 데이터 처리를 보다 쉽게 개발할 수 있는 라이브러리를 제공합니다.



5.2 [실습] Event Publishing

- ✓ Spring Boot를 이용하여 Kafka에 이벤트 발행을 실습합니다.

5.3 (실습) Event Consuming

- ✓ Spring Boot를 이용하여 Kafka에 발행한 이벤트 수신을 실습합니다.



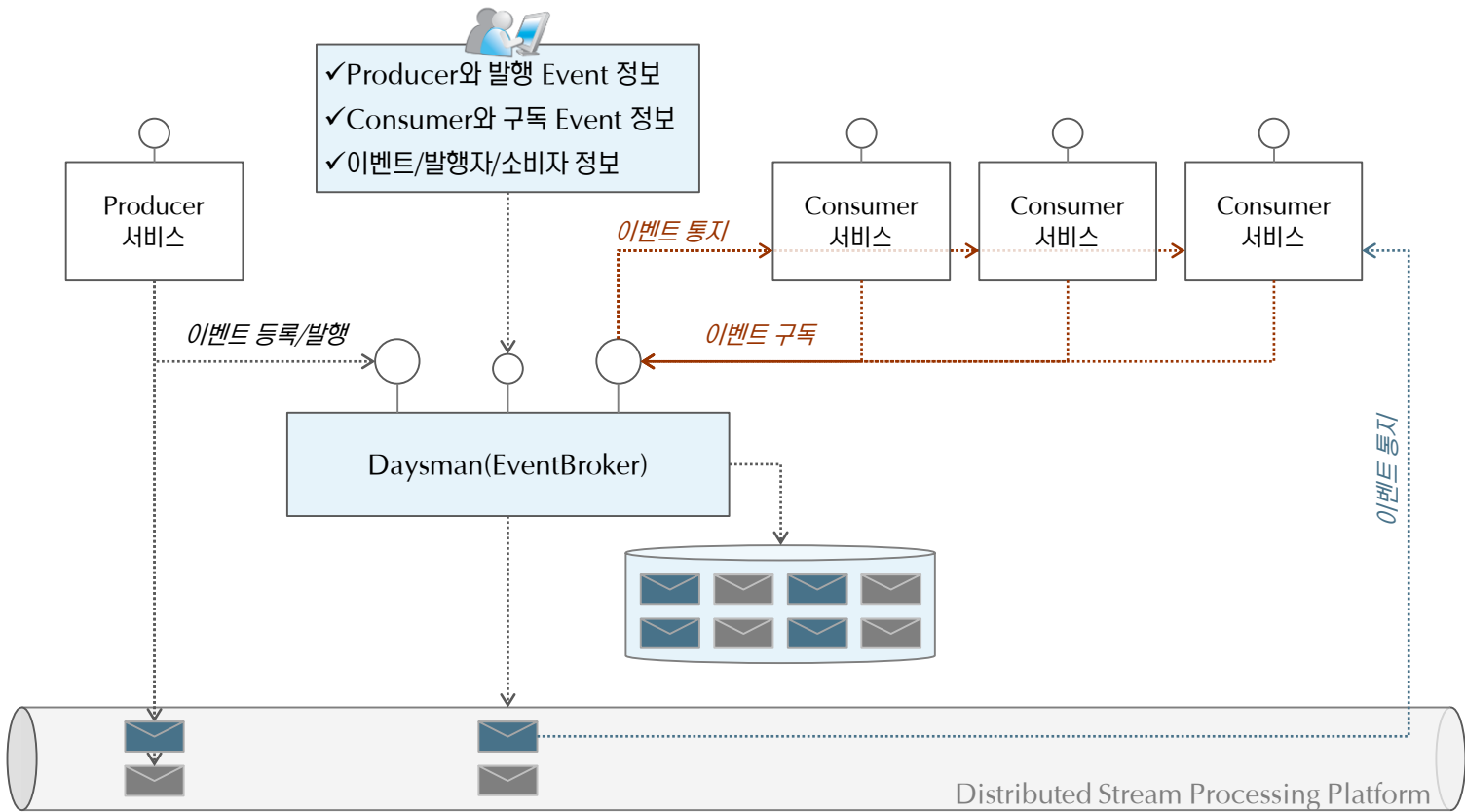
6. 이벤트 발행과 소비 (Daysman/Daysboy)

posco ict

- 6.1 Daysman/Daysboy
- 6.2 (실습) Event Publishing
- 6.3 (실습) Event Consuming

6.1 Daysman/Daysboy

- ✓ 스트림 이벤트는 Consumer를 고려하지 않고, 발행하는 특징이 있습니다.
- ✓ 하지만, 현실 세계의 엔터프라이즈 시스템은 이벤트 발행과 소비를 명확하게 추적할 수 있어야 운영이 편리합니다.
- ✓ Daysman은 Kafka와 같은 이벤트 플랫폼의 Proxy, Wrapper, 등의 역할을 수행하며 이벤트 관련 정보를 관리합니다.
- ✓ 개발과 디버깅 환경에서 Kakfa 없이도 개발과 테스트를 할 수 있도록 지원해주며, 제품화 환경에서는 구독정보를 관리합니다.



6.2 [실습] Event Publishing

- ✓ Daysboy 이용하여 Kafka에 이벤트 발행을 실습합니다.

6.3 [실습] Event Consuming

- ✓ Daysboy를 이용하여 Kafka에 발행한 이벤트 수신을 실습합니다.



7. 이벤트 분석

posco ict

7.1 (실습) Event Analysis

7.1 (실습) Event Analysis

- ✓ 발행한 이벤트(NewQuestionRegisteredEvent, AnswerRegisteredEvent)로 일별 통계를 생성합니다.

✓ 토론

감사합니다...

❖ 손경덕 (kdson@nextree.io)