Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

실무에서 Kafka 의 활용

1 User Activity Tracking 아키텍쳐

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

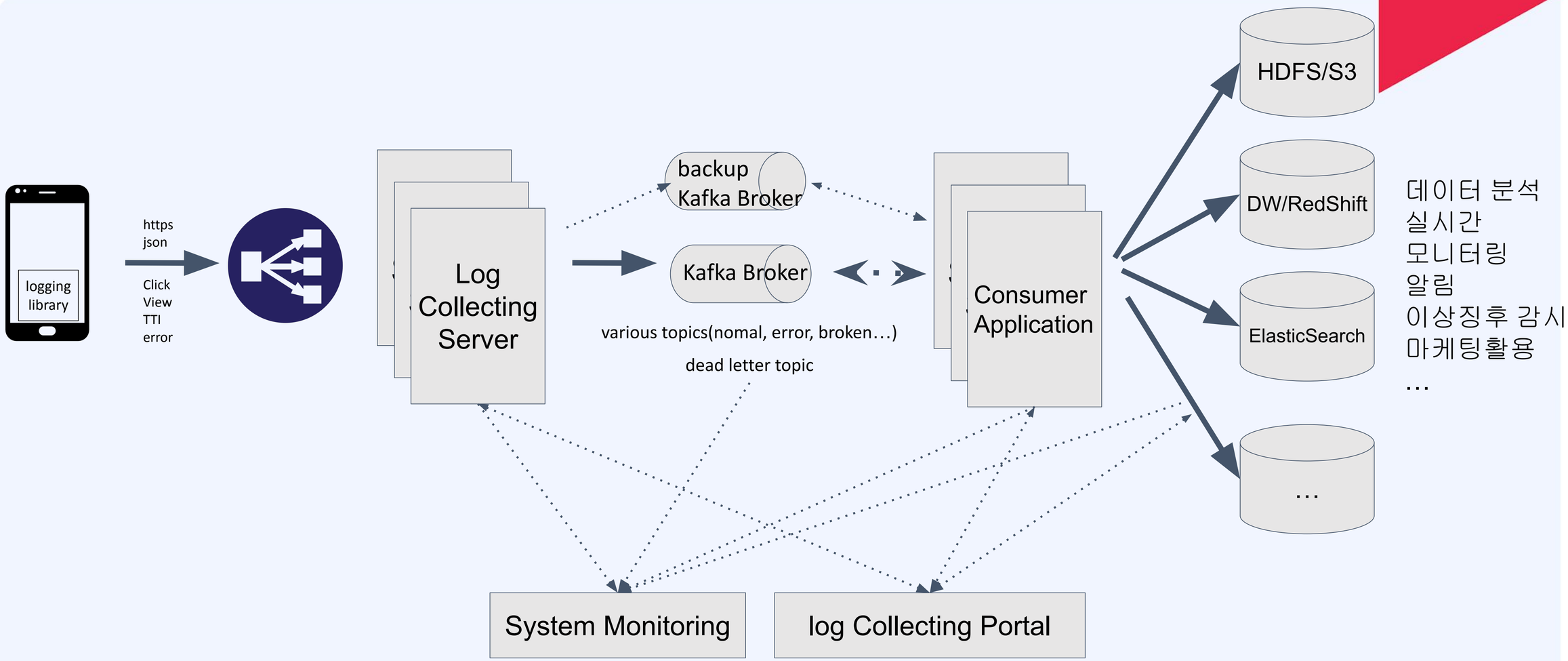
User Activity Tracking

- 고객의 페이지 뷰, 클릭등의 구체적인 행위를 수집하여 고객 행동을 분석/모니터링하고, 이를 통해 기능 개선이나 비즈니스 의사결정의 중요한 데이터로 활용
- 가능한 한 많이 수집하여 저장해 놓고, 이후 필요해 따라 적절히 가공하여 다양한 용도로 사용
- 데이터 수집은 고객에게 제공할 핵심 가치는 아니므로, 데이터 수집을 위해 Application 성능이나 기능에 영향을 끼쳐서는 안됨. 비동기 Batch 전송등을 활용하여 매우 심플하게 처리하는 것이 좋은 선택임.
- 데이터 규모가 매우 크고 폭발적으로 늘어날 수 있음을 고려해서 확장에 유연한 수집/저장 프로세스를 아키텍쳐링 해야함.
- 인터넷 네트워크상의 문제로 수집 서버로 데이터가 전달되지 않을 가능성도 있는만큼, 유실없는 완벽한 수집 보다는 빠르고 지속적인 수집에 좀 더 관심. (acks=1)
- 사용자 활동 추적은 개인 정보 보호에 영향을 미칠 수 있으므로 수집하는 데이터와 사용 방법을 고객에게 투명하게 공개하고 사용자가 원하는 경우 추적을 거부할 수 있는 옵션을 제공하는 것도 중요함.

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

1.

Web Activity Tracking



데이터 분석
실시간
모니터링
알림
이상징후 감시
마케팅활용
…

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

실무에서 Kafka 의 활용

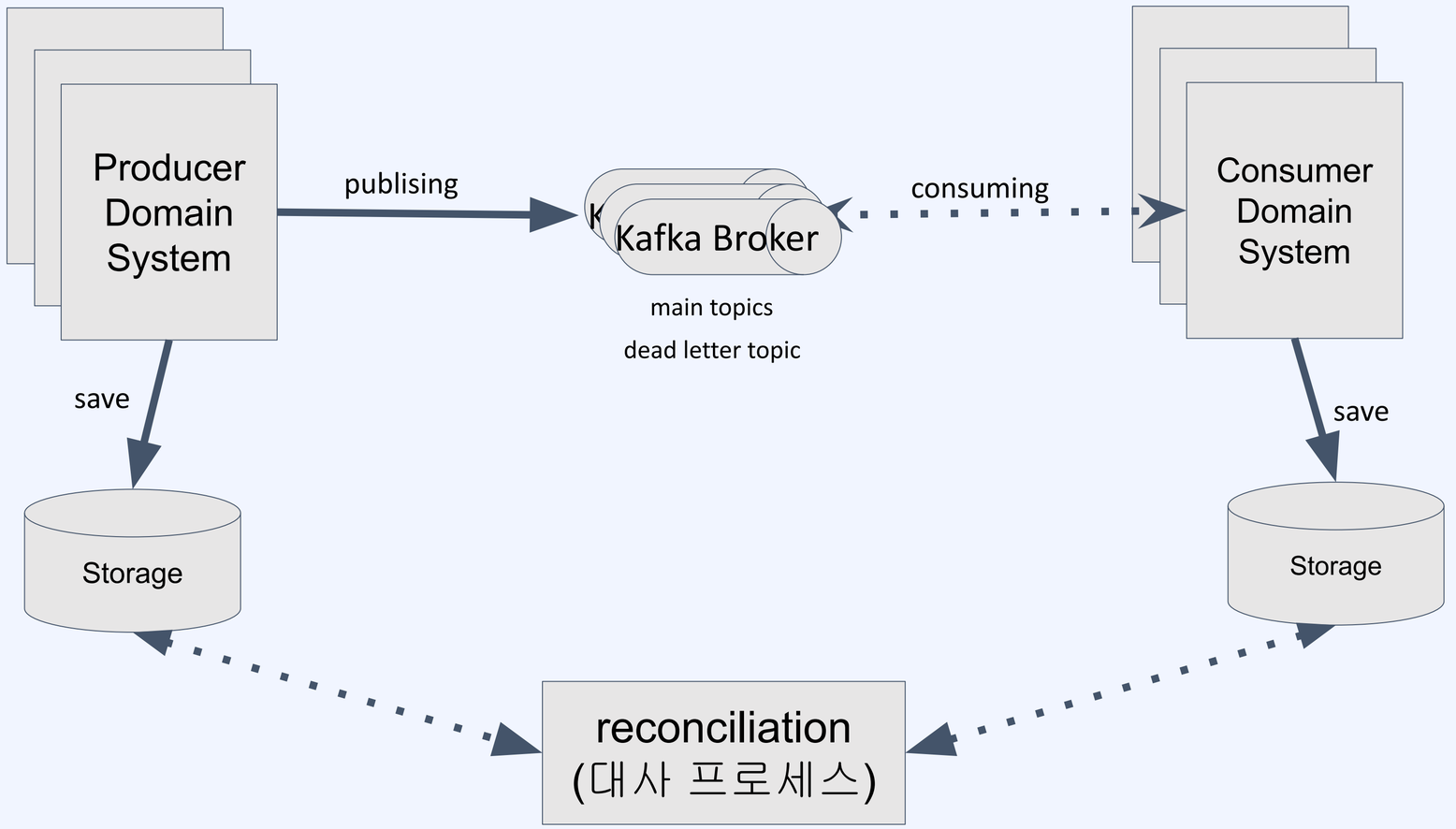
2 비즈니스 Application 간의 Messaging

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Messaging

- 비즈니스 도메인 간의 비동기 프로세스에 사용되는 메세지 브로커로서, ActiveMQ, Rabbit MQ 등을 대체
- 예를 들어, 메세지 발행자 시스템의 트랜잭션이 완료된 후 해당 이벤트에 따른 후속 프로세스가 필요한 다른 시스템에 트랜잭션 결과를 통지.
  - 결재시스템의 프로세스를 완료하고, 배송 도메인에 결과 메세지를 전달.
  - 회원 가입 프로세스를 완료하고, 마케팅 도메인에 회원가입 결과 메세지를 전달.
  - 마케팅 도메인에서 고객에게 발송할 메세지 전송 요청건을 생성하여, 메세지 발송 도메인에 전달 등
- 메세지 유실 가능성을 최소화 할 수 있도록, 강화된 프로듀서 설정(ack=all)을 하거나, 컨슈머에서 dead leatter queue(topic) 를 사용하여 재처리 프로세스를 만들 수 있음. 데이터 누락이나 오류시 매우 크리티컬한 정보라면 별도의 대사 프로세스를 만들어 무결성을 체크하는 경우도 있음.

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved



Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

실무에서 Kafka 의 활용

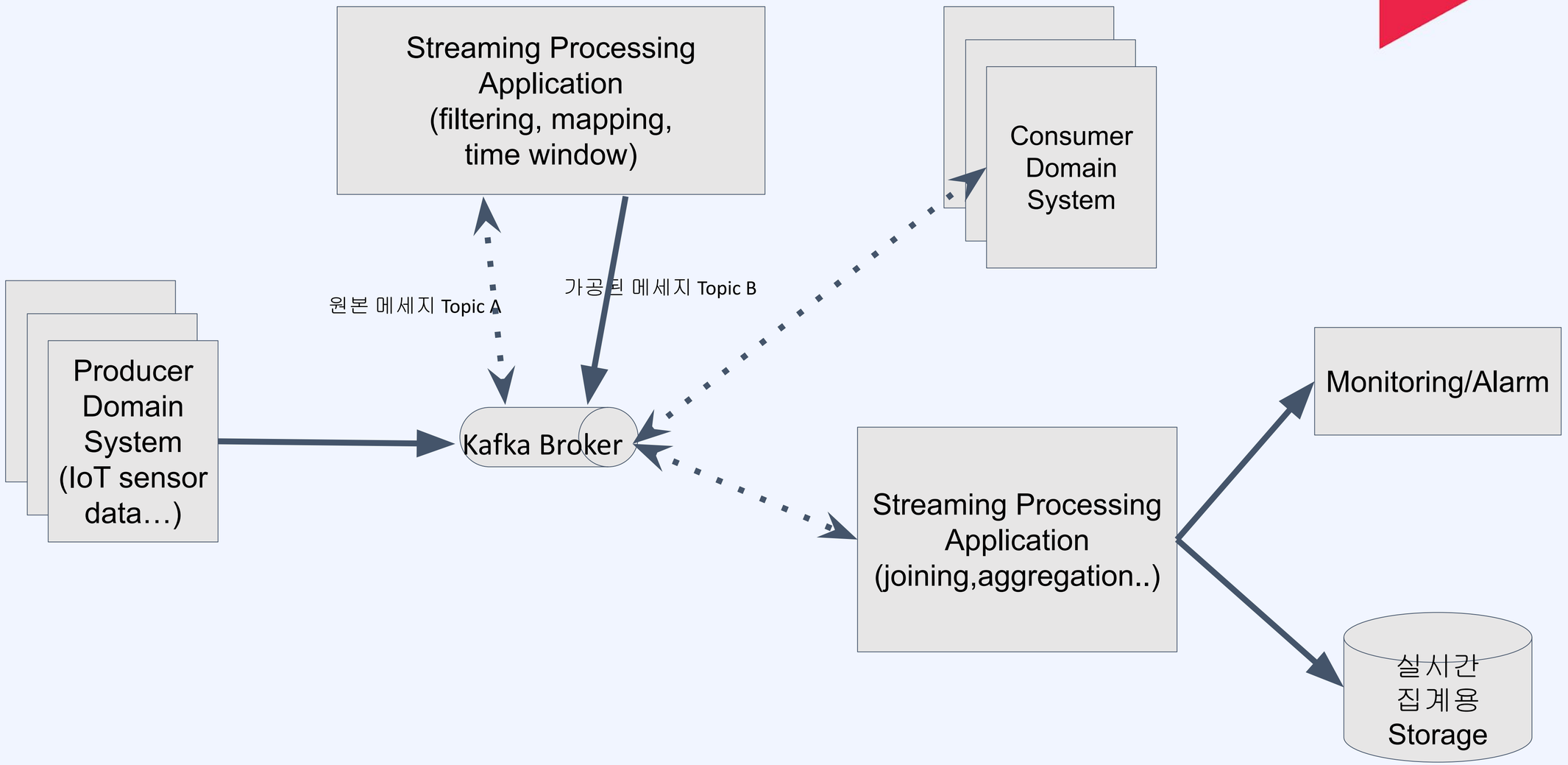
3 Streaming Processing 아키텍쳐

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Streaming Processing

- 지속적으로 토픽에 인입되는 이벤트 메세지를 실시간으로 가공하거나 집계, 분할 하는 등의 프로세싱
- 예를 들어.
  - User Activity Tracking 으로 인입되는 원본 로그 메세지를 재 가공하여 새로운 토픽에 저장
  - IoT 시스템에서 지속적으로 인입되는 이벤트 데이터를 실시간으로 분석
  - Time Window 를 적용하여 최근 10분간의 집계 데이터를 생성하여 슬랙 채널에 자동으로 리포트
  - 시스템의 문제나 비즈니스 데이터의 문제상황을 실시간으로 캐치하려는 Alarm 발생
  - ML Model 에 사용되는 실시간 Feature 를 생성
- Kafka Streams, Apache Storm, Spark Streaming, Apache Flink...

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved



Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Kafka Streams Sample

```
import org.apache.kafka.streams.kstream.TimeWindows;

// Set up a 5-minute time window
final TimeWindows windowSpec =
TimeWindows.of(Duration.ofMinutes(5)).advanceBy(Duration.ofMinutes(1));

// Set up the input and output topics
final StreamsBuilder builder = new StreamsBuilder();
final KStream<String, Long> input = builder.stream("original-topic");
final KTable<Windowed<String>, Long> windowedCounts = input
    .groupByKey()
    .windowedBy(windowSpec)
    .count();

// Write the windowed counts to the output topic
windowedCounts.toStream().to("output-topic", Produced.with(Serdes.String(),
Serdes.Long()));

// Create the Kafka Streams instance and start it
final KafkaStreams streams = new KafkaStreams(builder.build(),
streamsConfiguration);
streams.start();
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

실무에서 Kafka 의 활용

4 Event Sourcing 아키텍쳐

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Event Sourcing

- 어플리케이션의 상태에 대한 모든 변경사항을 일련의 이벤트로 표현되는 디자인 패턴
- 어플리케이션은 상태에 대한 전체 변경 기록을 저장하고 이벤트를 재생하여 현재 상태를 재구성할 수 있음
- 대규모의 MSA 아키텍쳐에서 CQRS 패턴과 결합하여 도입되는 추세
- CQRS 패턴에서 실시간으로 전체 이벤트에 기반하여 현재 상태를 생성하는 것은 성능상의 한계가 있기 때문에, Event Handler 에서 조회시에 사용할 상태값을 구체화된 뷰 (MATERIALIZED VIEW) 에 생성하여 조회시에 사용.
- Kafka는 이벤트 소싱 기반 어플리케이션에서 이벤트 스토어로 활용

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved