Open Infra & Cloud Native Korea 2022

오픈소스 분산 로그 저장소와 메시지큐로의 활용

카카오 클라우드 S&M플랫폼파트 분산기술셀
이영식 den.lee

자기소개

den.lee

Dan이라고 했어야 했는데, Den(동굴...)으로 이름을 잘못 정해 놀림 받고 있습니다.

1년 전 카카오에 합류하여 분산로그저장소 varlog를 활용해 Kov라는 메시지큐를 개발하고 있습니다.

반갑습니다!

1. 분산 로그 저장소

- 로그란?
- 공유 로그와 State Machine Replication
- 분산 로그 저장소와 용도

2. 오픈소스 분산 로그 저장소 varlog

- Varlog 소개
- Varlog 구조
- Global Order의 필요성
- 다양한 Global Order 구현 방식과 문제점

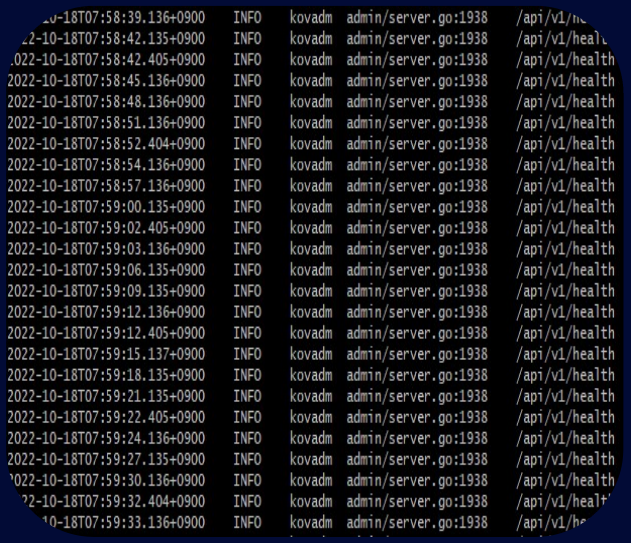
3. 메시지큐 Kov(Kafka on varlog)

- Kov 소개
- Kov의 구조
- Kafka의 Pub-Sub 방식
- Kov의 Kafka Pub-Sub 요청 처리
- Kov 기능 확인

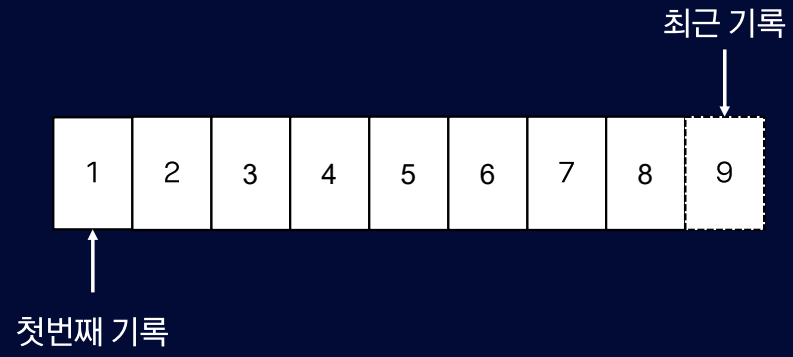
분산 로그 저장소

로그(Log)란?

흔히 어플리케이션 로그만을 떠올릴 수 있으나, 시간 순서대로 연속으로 기록된 데이터는 모두 로그로 볼 수 있습니다.



어플리케이션 로그



로그

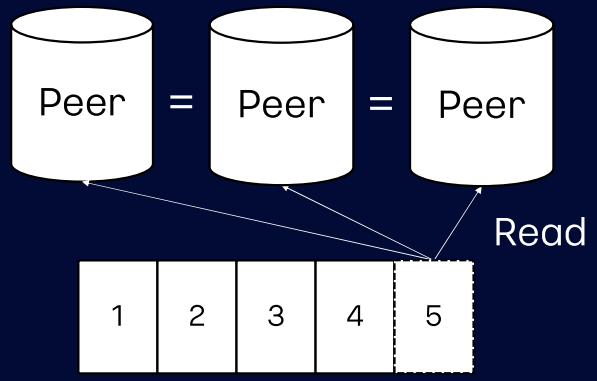
공유 로그와 State Machine Replication

단일 시스템에서 로그는 상태 복구에 이용되지만, 분산 시스템에서는 공유가능한 로그를 상태 동기화에 사용합니다.

같은 프로그램이 같은 명령들을 동일한 순서로 처리하면 같은 상태로 결정되는 것을 이용하는 것입니다.

WAL(Write-Ahead Logging)

e.g. 데이터베이스 장애 복구

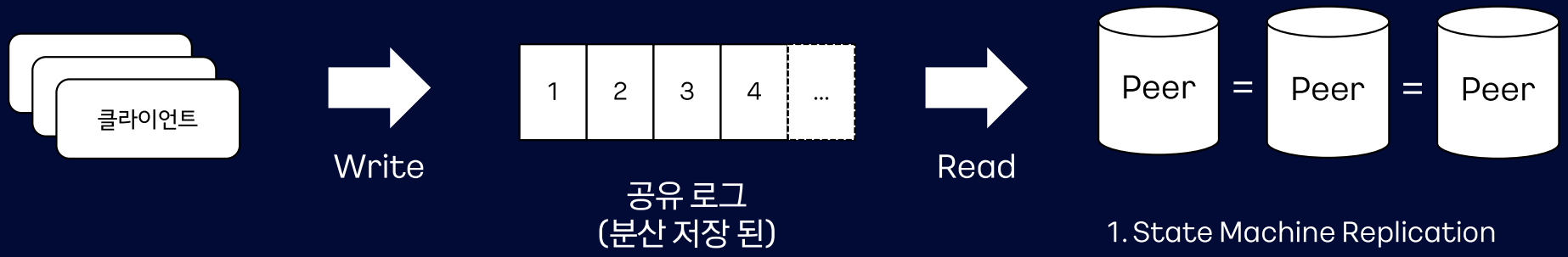


State Machine Replication

e.g. 분산 데이터베이스 인덱스 동기화

분산 로그 저장소와 용도

분산 로그 저장소는 공유가능한 로그를 분산해서 저장하는 시스템입니다.



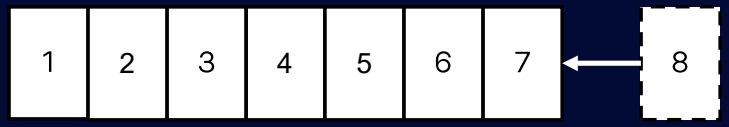
1. State Machine Replication
2. 어플리케이션 로깅
3. 메세지큐
4. WAL(Write-Ahead Logging)
5. 분산 트랜잭션
6. ...

오픈소스 분산 로그 저장소 varlog

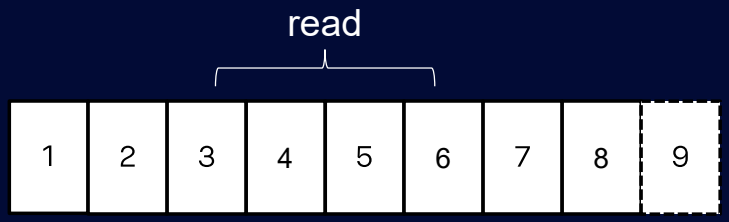
varlog 소개

varlog는 간단한 API로 이용가능한 오픈소스[https://github.com/kakao/varlog] 분산 로그 저장소입니다.

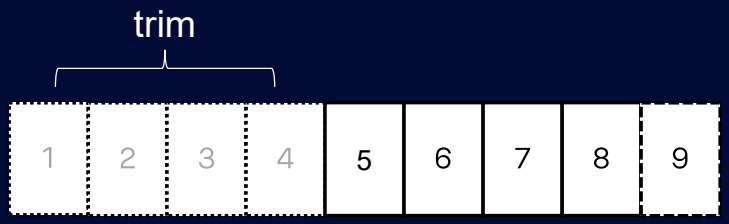
* Append: 로그 끝에 데이터를 이어 붙이기



* Subscribe: 임의의 로그 범위에 저장된 데이터 읽기



* Trim: 로그 앞 부분 삭제



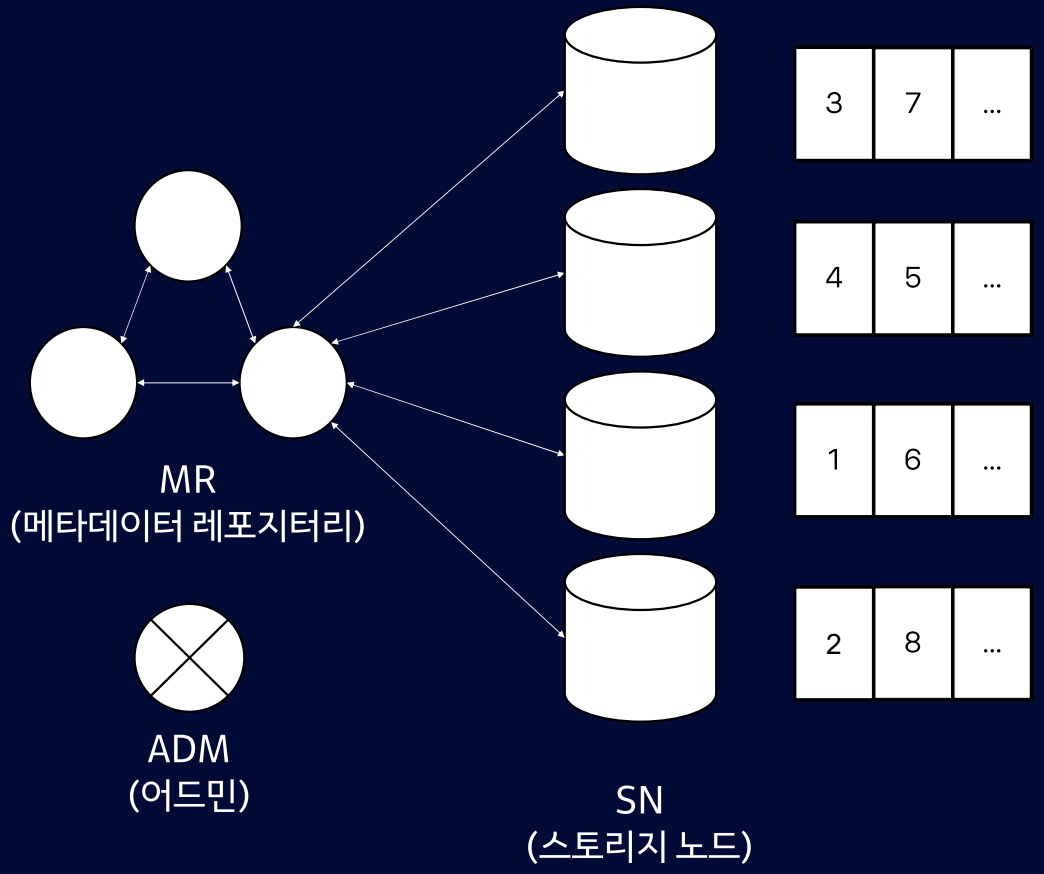
varlog 구조

varlog는 3가지 컴포넌트로 구성됩니다.

SN (Storage Node)
- 로그들을 분산해 저장합니다.
- 클라이언트가 로그를 추가하거나 읽어 감

MR (Metadata Repository)
- 분산 저장 된 로그들에 일관된 순서를 매겨줍니다.

ADM(Admin)
- SN을 관리
- MR을 관리



슬라이드 10

SI0 SN, MR 과 같은 형식으로 적는다면, ADM (Admin) 이 되어야할 것 같아요. 그림에 약자로만 되어 있기 때문에...
Song Injun, 2022-10-25T07:03:51.744

YL0 0 아 그렇네요 이런건 여러번 봐도 제눈에는 왜 안 보이는걸까요... 수정했습니다.
YeongSik Lee, 2022-10-25T07:08:44.706

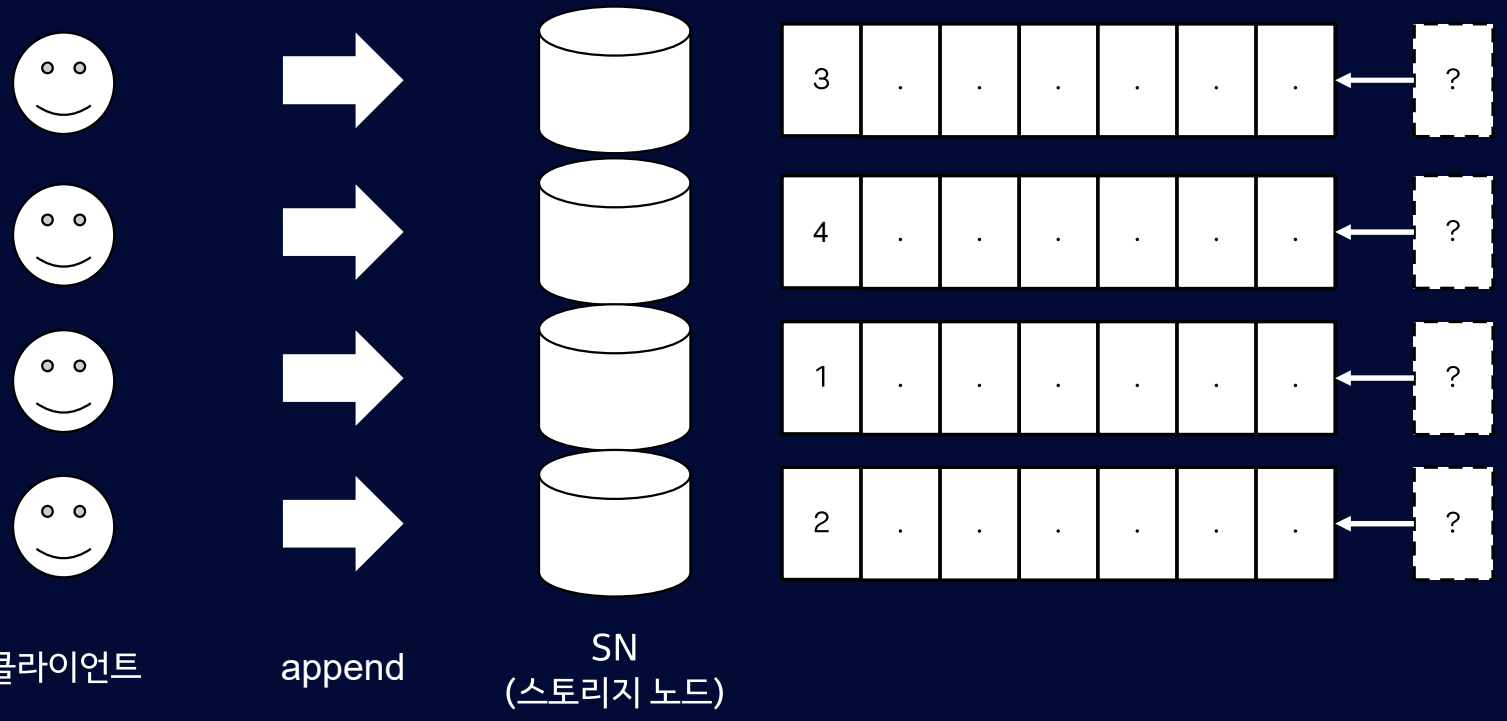
SI1 사소한 것이지만, Varlog 는 로그 순서를 1부터 부여하고 있어요. 그림에는 0이 있어서...
Song Injun, 2022-10-25T07:04:31.282

YL1 0 네 간단한거니 혹시 오해하지않게 1부터 부여하는걸로 변경해야겠네요
YeongSik Lee, 2022-10-25T07:07:30.752

Global Order의 필요성

분산되어 저장되더라도 로그는 시간 순서대로 연속적이어야 하기에 일관된 순서(Global Order)가 필요합니다.

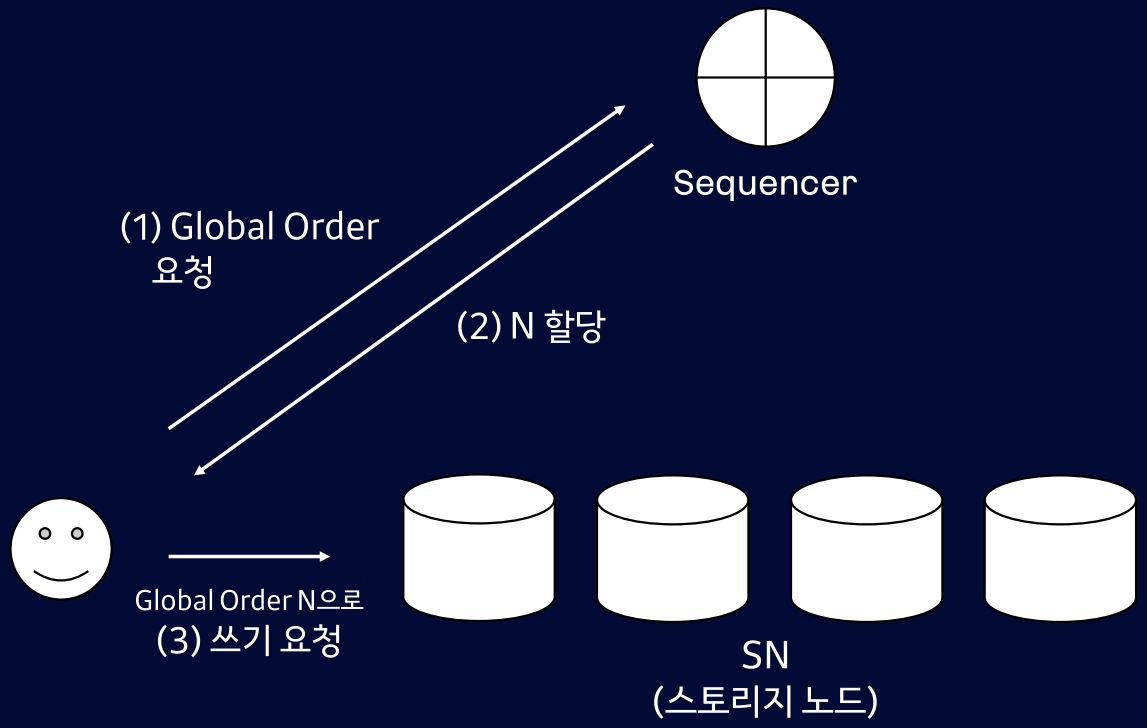
그러나, SN들은 동시에 각자 요청을 받아 로그를 쓰기 때문에 일관된 순서를 매기는데 어려움이 있습니다.



다양한 Global Order 구현 방식과 문제점

기존 분산 로그 저장소들은 Sequencer라는 순서 관리자를 만들어 Global Order를 구현했습니다.

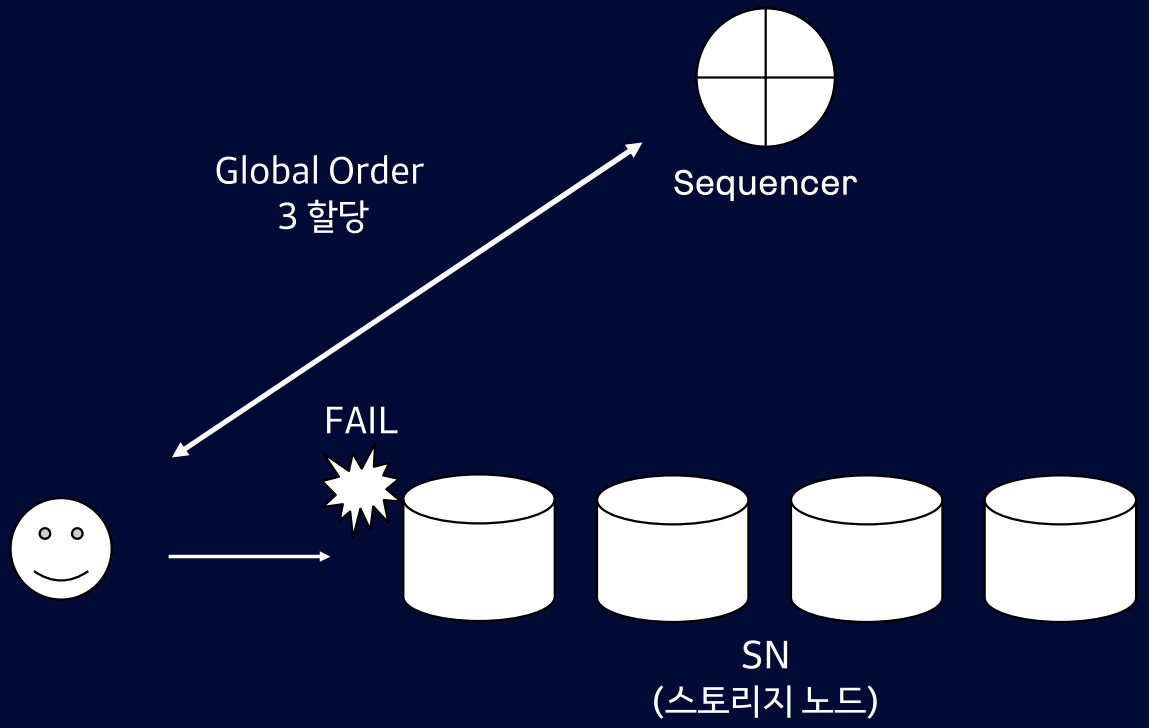
1) 클라이언트가 Sequencer를 통해 로그의 Global Order를 미리 할당 받아, 할당 받은 순서로 SN에 로그 쓰기를 요청

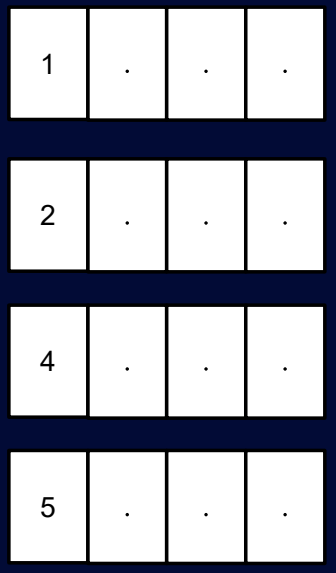


다양한 Global Order 구현 방식과 문제점

기존 분산 로그 저장소들은 Sequencer라는 순서 관리자를 만들어 Global Order를 구현했습니다.

1번 방식은 할당 받은 Global Order로 로그 쓰기에 실패한다면 로그에 Hole이 생기는 문제가 존재합니다.





3번이 없는 불완전한 로그

슬라이드 13

SI0 이전장과 비교하면, 화살표의 설명이 잘못붙어있는 것 같아요.
Global Order 2 할당은 Sequencer 에서 클라이언트 방향
화살표에 붙어야 할 것 같아요.
Song Injun, 2022-10-25T07:08:35.464

YL0 0 이 문구를 위쪽 위치에 둔건

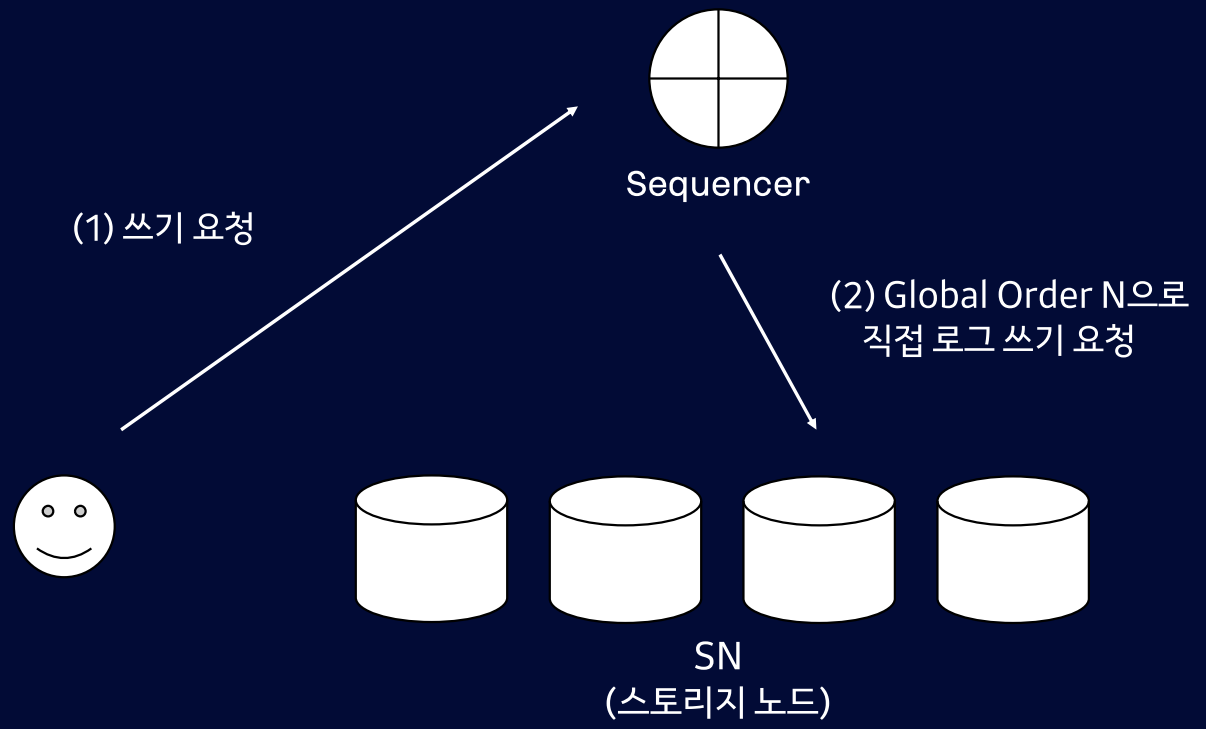
얘기하신대로 아랫 화살표 밑으로
배치했더니 눈에 잘 안들어와서 였는데요

음... 차라리 이 두 단방향 화살표를
양방향 화살표 하나로 변경해보겠습니다.
YeongSik Lee, 2022-10-25T07:11:21.647

다양한 Global Order 구현 방식과 문제점

기존 분산 로그 저장소들은 Sequencer라는 순서 관리자를 만들어 Global Order를 구현했습니다.

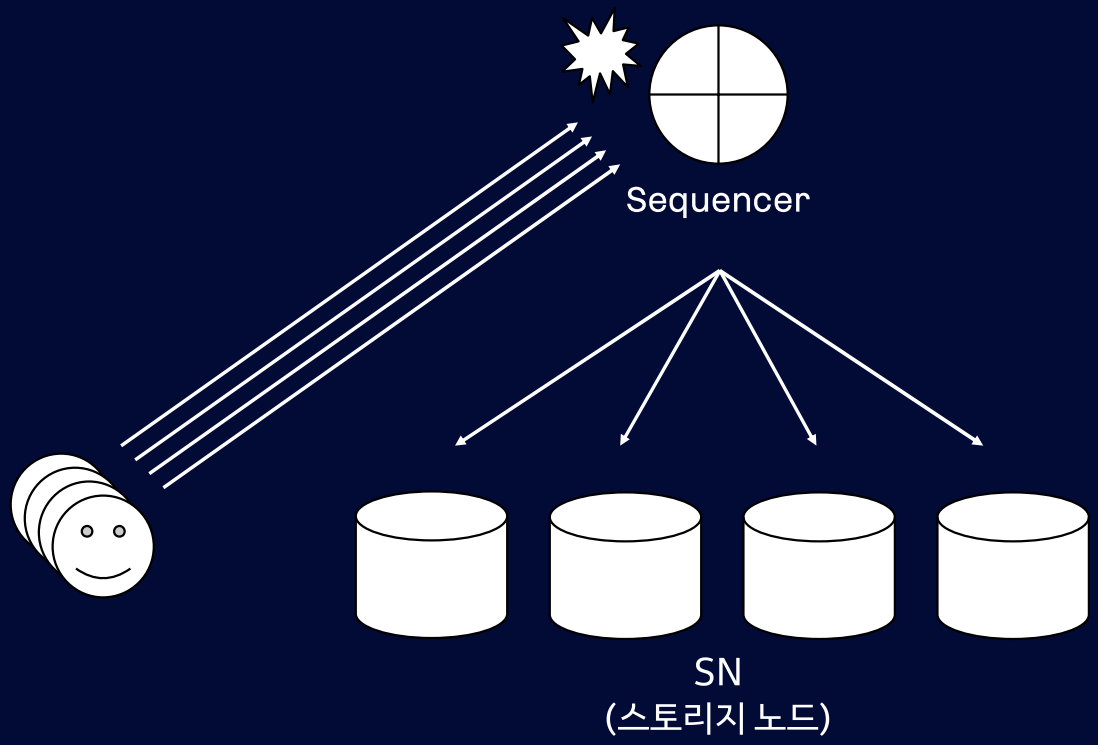
2) 클라이언트가 Sequencer를 통해 로그 쓰기를 요청하면, Sequencer가 알고 있는 Global Order로 직접 쓰기까지 처리



다양한 Global Order 구현 방식과 문제점

기존 분산 로그 저장소들은 Sequencer라는 순서 관리자를 만들어 Global Order를 구현했습니다.

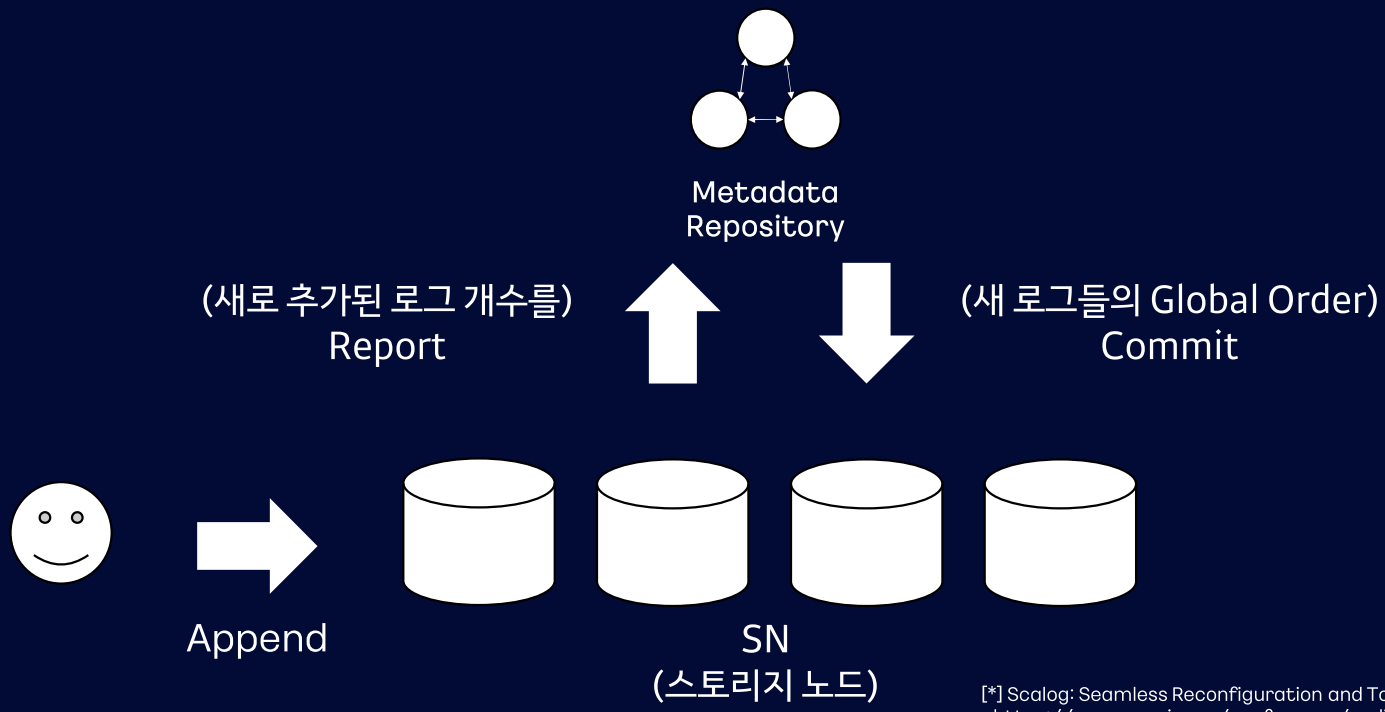
2번 방식은 Sequencer가 쓰기 성능의 병목이 되고, 로그를 추가할 SN을 직접 선택할 수 없다는 문제가 존재합니다.



다양한 Global Order 구현 방식과 문제점

Sequencer 없이 Global Order를 지원하는 새로운 방식이 2020년 Scalog[*]라는 논문을 통해 제안됐습니다.

3) 클라이언트는 로그를 SN에 먼저 쓰고, 그 이후에 로그에 순서가 부여(Commit) 됨



[*] Scalog: Seamless Reconfiguration and Total Order in a Scalable Shared Log)
- https://www.usenix.org/conference/nsdi20/presentation/ding

슬라이드 16

SI0 각주는 슬라이드 하단에 위치하는건 어떨까요? Scalog 근처에 너무 많은 텍스트가 있어 보이는데.. (이건 개인 취향일수도..ㅎ)
Song Injun, 2022-10-25T07:11:32.174

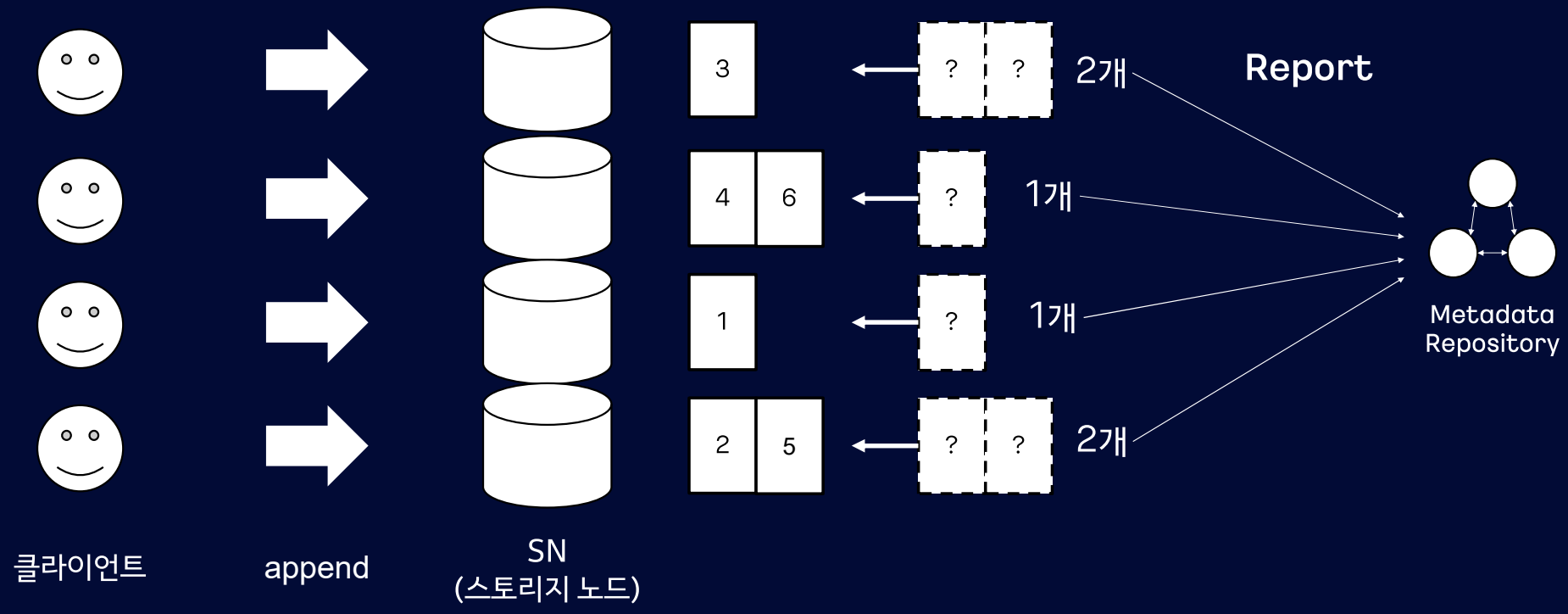
YL0 0 각주가 왠만하면 다 하단에 있긴 하더라구요

저 그림들이 공간을 너무 많이 차지해서 어쩔수 없이 위쪽으로 배치해봤는데요

음... 각주 글자 크기를 더 줄이면 가능할 것 같긴합니다.
YeongSik Lee, 2022-10-25T07:18:35.404

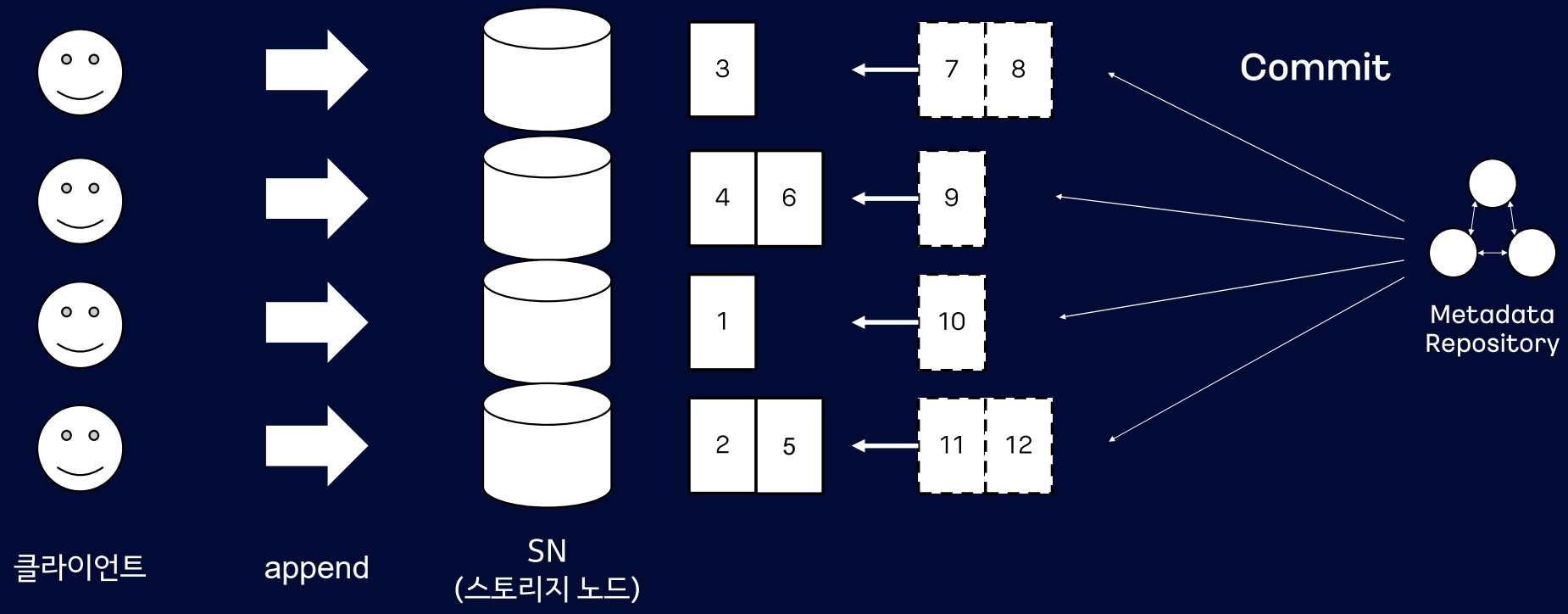
다양한 Global Order 구현 방식과 문제점

모든 SN들은 MR이라는 컴포넌트에 주기적으로 새로 추가된 로그 개수를 Report 합니다.



다양한 Global Order 구현 방식과 문제점

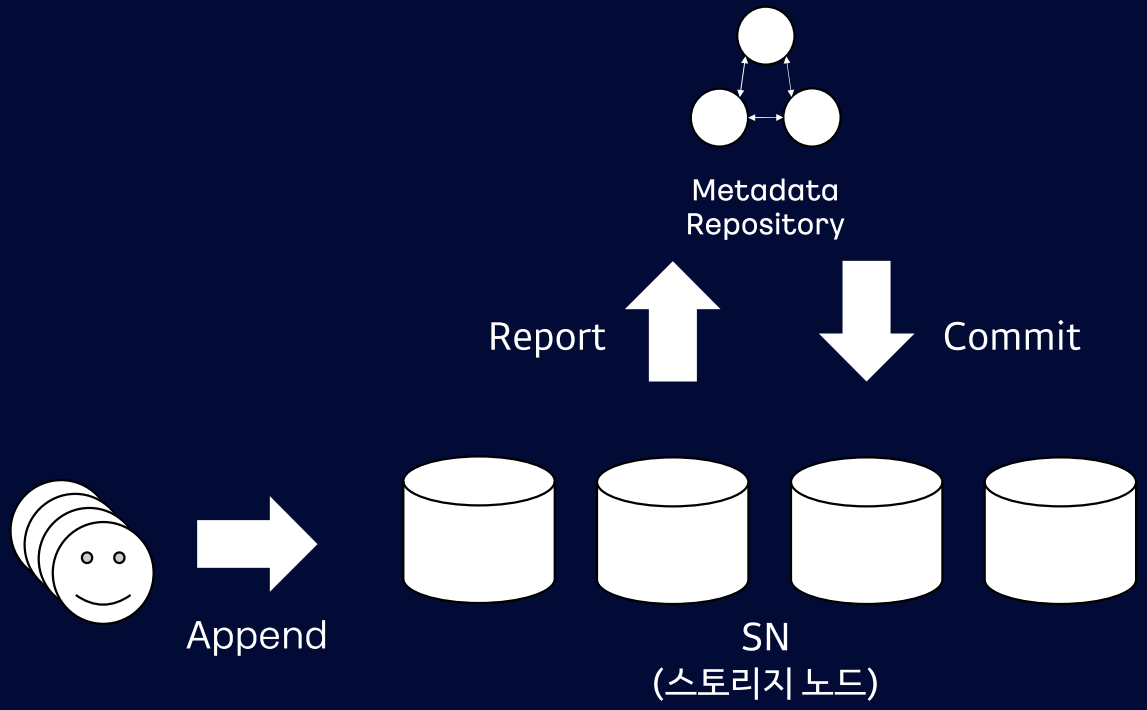
새로 추가된 로그의 수를 취합한 MR이 추가된 로그들에 Global Order를 Commit 하고 로그들의 SN에 알려줍니다.



다양한 Global Order 구현 방식과 문제점

Sequencer 없이 Global Order를 지원하는 새로운 방식이 2020년 Scalog[*]라는 논문을 통해 제안됐습니다.

3번 방식은 쓰기 실패에도 Global Order 부여 전에 실패 처리가 되어서 로그에 Hole이 생기지 않고,
클라이언트들이 각자 원하는 SN에 나눠서 로그를 쓰기 때문에 Sequencer 같은 쓰기 성능 병목이 없습니다.



여러 Global Order 지원 방식과 문제점

3가지 방안을 비교한 결과, varlog는 마지막 방식을 사용하기로 했습니다.

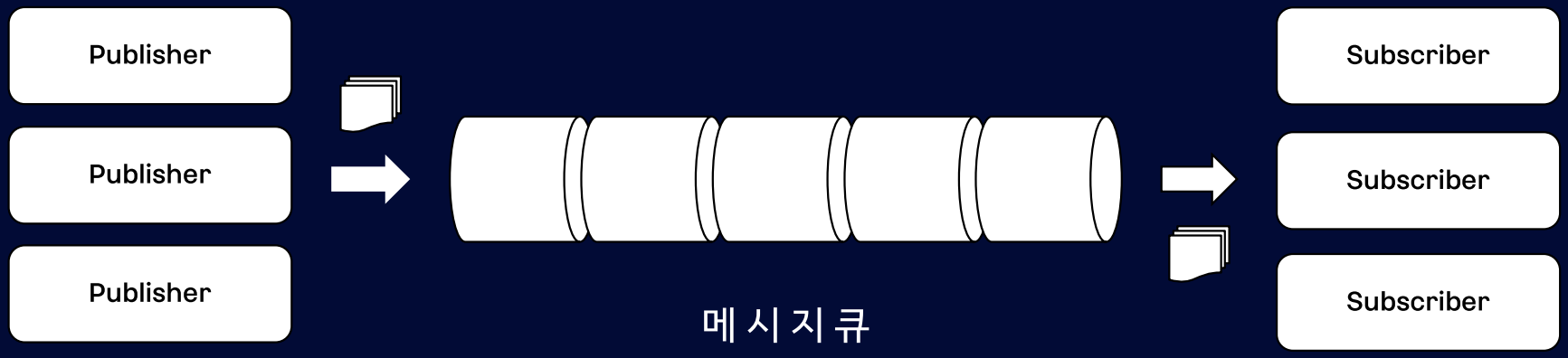
[1] https://research.vmware.com/projects/corfu
[2] https://logdevice.io
[3] https://www.usenix.org/conference/nsdi20/presentation/ding

Global Order 지원 방식	특징	적용 사례
1. Sequencer가 순서만 제공하고 클라이언트가 로그를 쓰는 방식	제공받은 Global Order로 쓰기 실패 시 로그에 Hole이 발생	Corfu[1]
2. Sequencer가 데이터를 받아서 Sequencer가 로그를 쓰는 방식	Sequencer가 쓰기 성능의 병목 + 클라이언트가 원하는 곳에 로그를 쓰지못함	LogDevice[2]
3. 클라이언트가 로그를 직접 쓰고 Global Order를 부여 받는 방식	쓰기 실패 시 로그에 Hole이 발생하지 않음 + 클라이언트가 원하는 곳에 로그를 쓸 수 있음 + 쓰기 성능 병목이 없음	Scalog[3]

메시지큐 Kov (Kafka on varlog)

Kov(Kafka on varlog) 소개

varlog의 기능과 성능에 문제가 없는지 확인하는 용도로 varlog를 메시지큐로 사용하는 서비스를 만들었습니다.



1. State Machine Replication
2. 어플리케이션 로깅
3. 메세지큐
4. WAL(Write-Ahead Logging)
5. 분산 트랜잭션
6. ...

Kov(Kafka on varlog) 소개

메시지큐 중 가장 흔히 사용되는 Apache Kafka 클라이언트를 사용할 수 있도록 Kafka Layer를 구현했습니다.

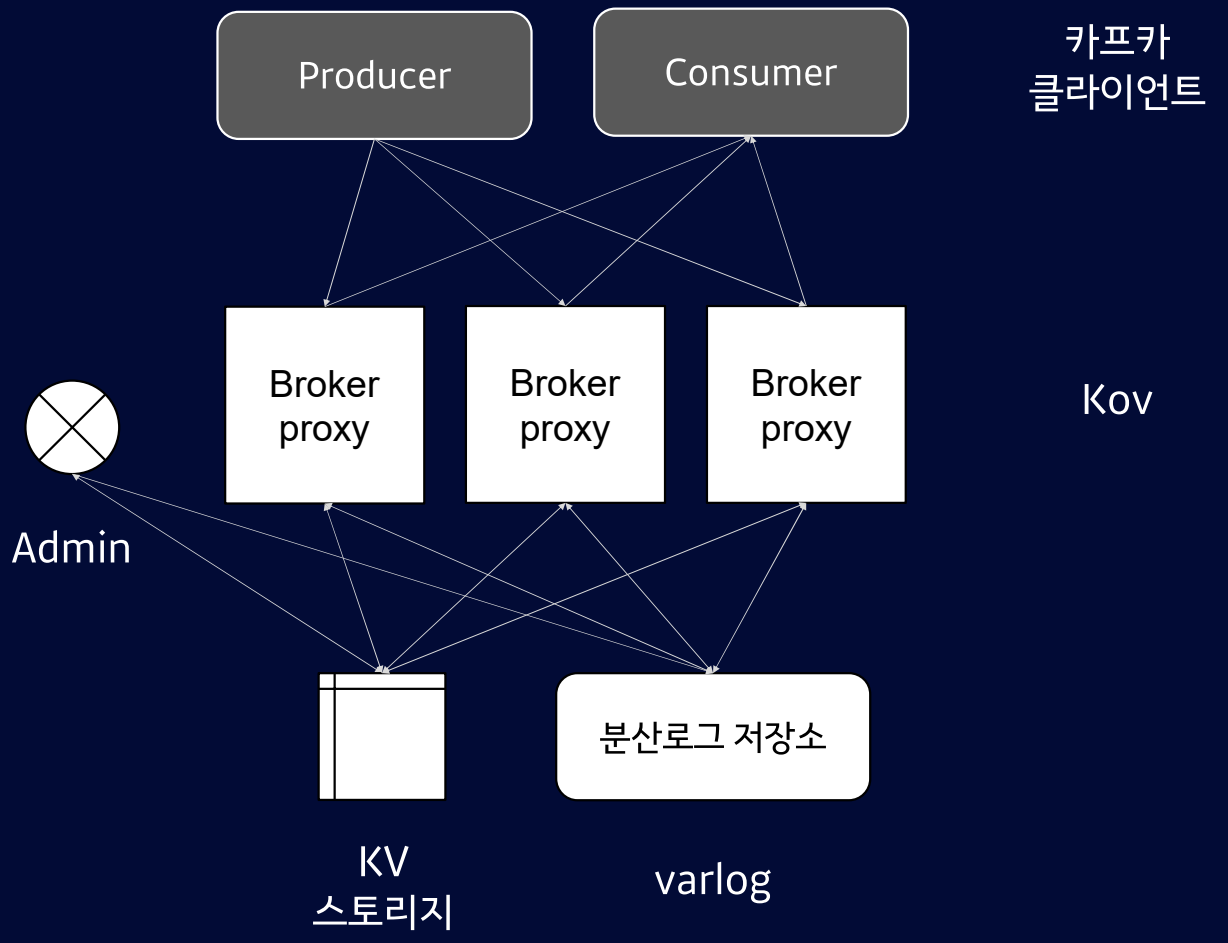
Kov(Kafka on varlog) 구조

Kov는 3가지 컴포넌트로 구성됩니다.

Brokerproxy
- Kafka Layer를 처리
- varlog에 로그를 쓰거나 읽음

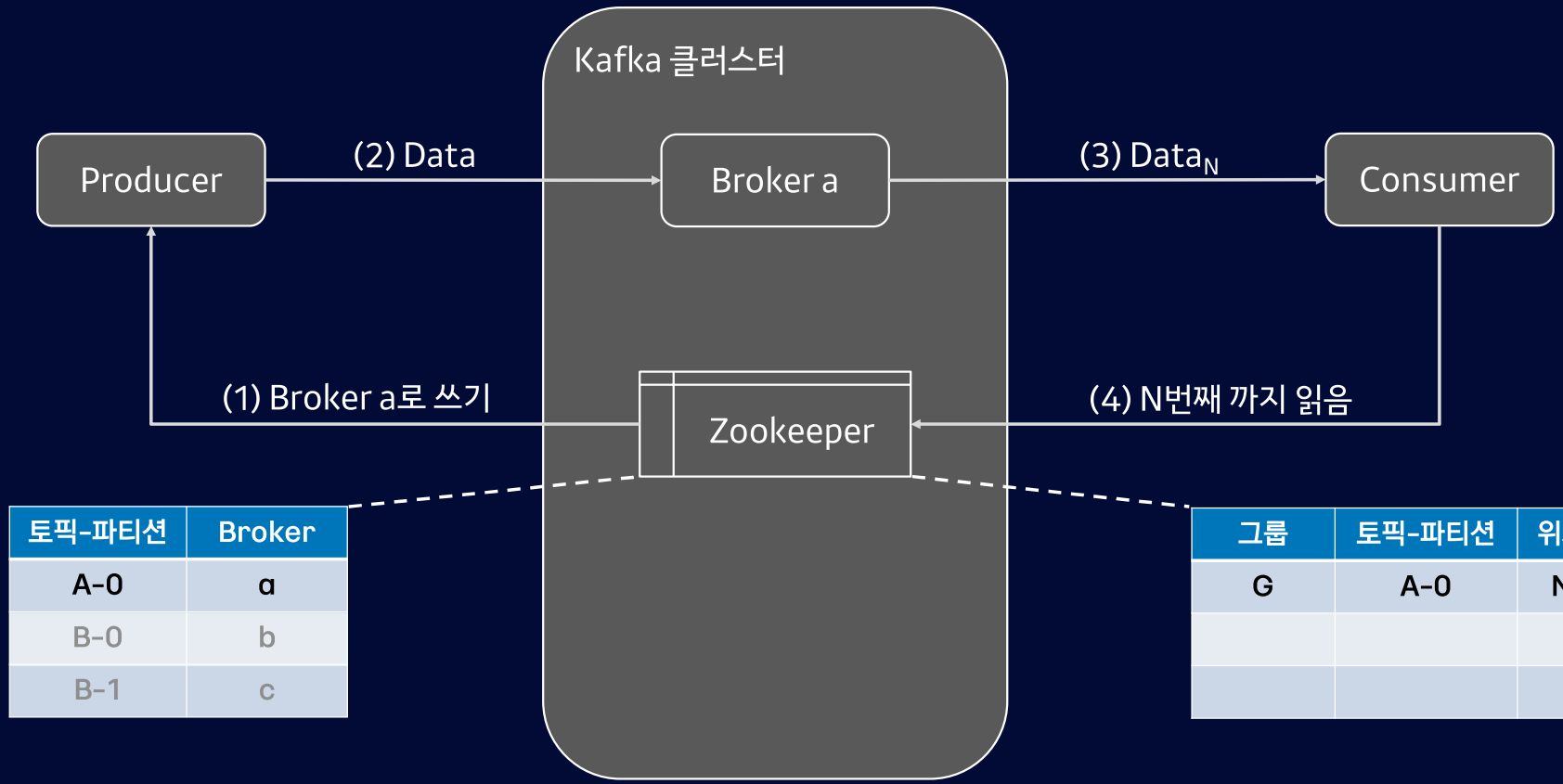
Admin
- Brokerproxy들을 관리
- varlog의 로그 관리

KV(Key-Value) 스토리지
- Kafka 관련 메타데이터를 저장
  (Kafka에서 Zookeeper 역할)



Kafka의 Pub-Sub 방식

Kafka는 메시지를 쓰고, 읽는데 필요한 메타데이터를 Zookeeper에 기록합니다.



토픽-파티션	Broker
A-0	a
B-0	b
B-1	c

그룹	토픽-파티션	위치
G	A-0	N

슬라이드 25

SI0 "Broker a로 쓰기" 라는 말이 어색한것 같아요. 대충 보면 토픽 A-0 의 리더 레플리카를 갖고 있는 브로커 정보를 알아오는 것 같은데요... 이 그림에 어울리는 / 좀더 이해하기 쉬운 말이 있을까요?
-> 우선 이 그림이 이해하기 쉬운지 다른 분들에게도 여쭤봐야겠네요.
Song Injun, 2022-10-25T07:18:10.042

YL0 0 네 이 발표에서 토픽-파티션 같은 복잡한 개념을 얘기안하고 이 부분에서는 어쩔 수 없이 언급을 하다 보니 잘 이해가 안 될 것 같기도 합니다.

이 장표에서도 토픽-파티션 언급을 안하고 비슷한 쉬운 단어로 얘기하고 마는게 나을지 모르겠네요
YeongSik Lee, 2022-10-25T07:20:00.672

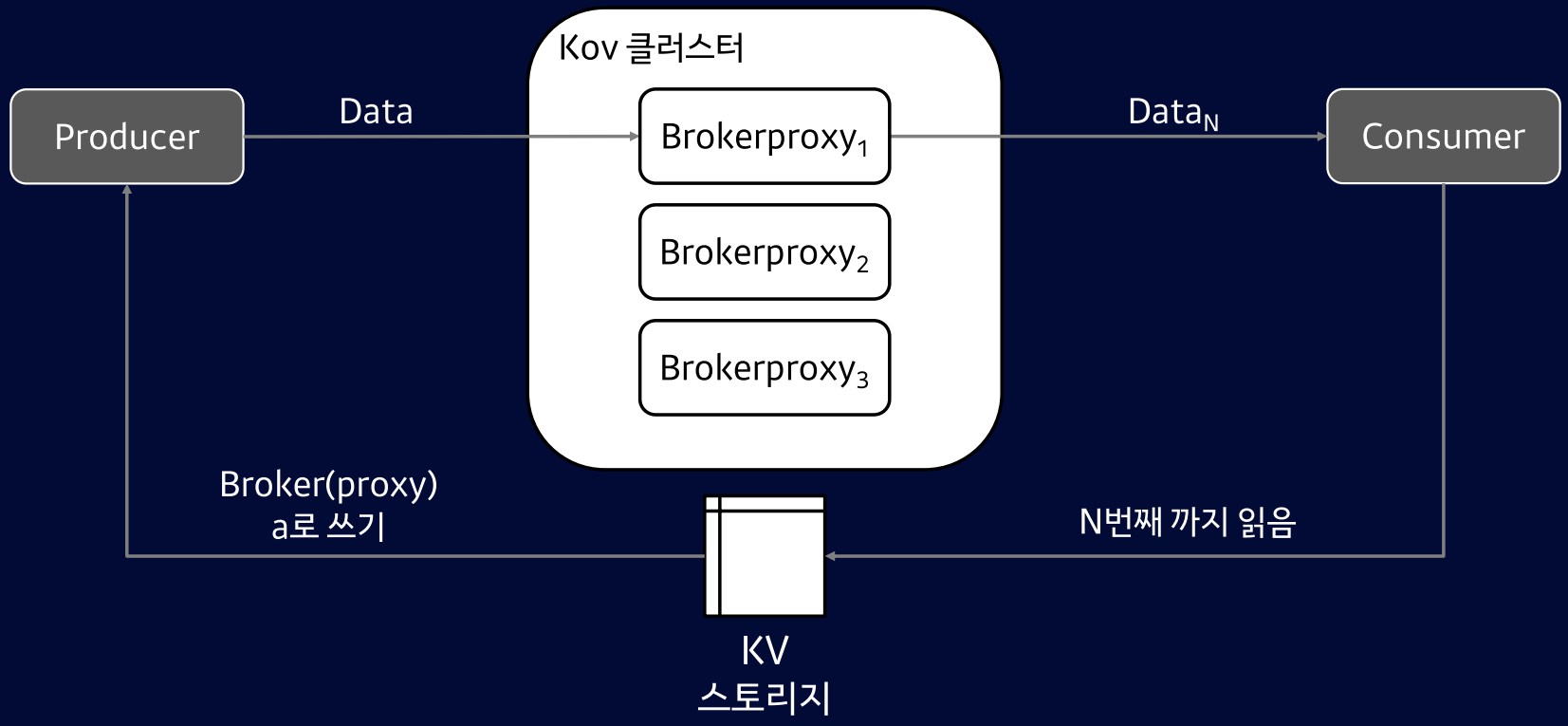
SI1 Consumer -> Zookeeper 방향 화살표는 데이터를 읽은 오프셋을 저장하는 것을로 보이는데요,
현재 표현 "(컨슈머-1)N번째 까지 읽음" 이 이해하기 쉬운 표현인지 모르겠네요..
Song Injun, 2022-10-25T07:18:58.730

YL1 0 N번째 데이터까지 읽었음

이라고 간단히 끊는게 나을려나요
YeongSik Lee, 2022-10-25T07:22:43.315

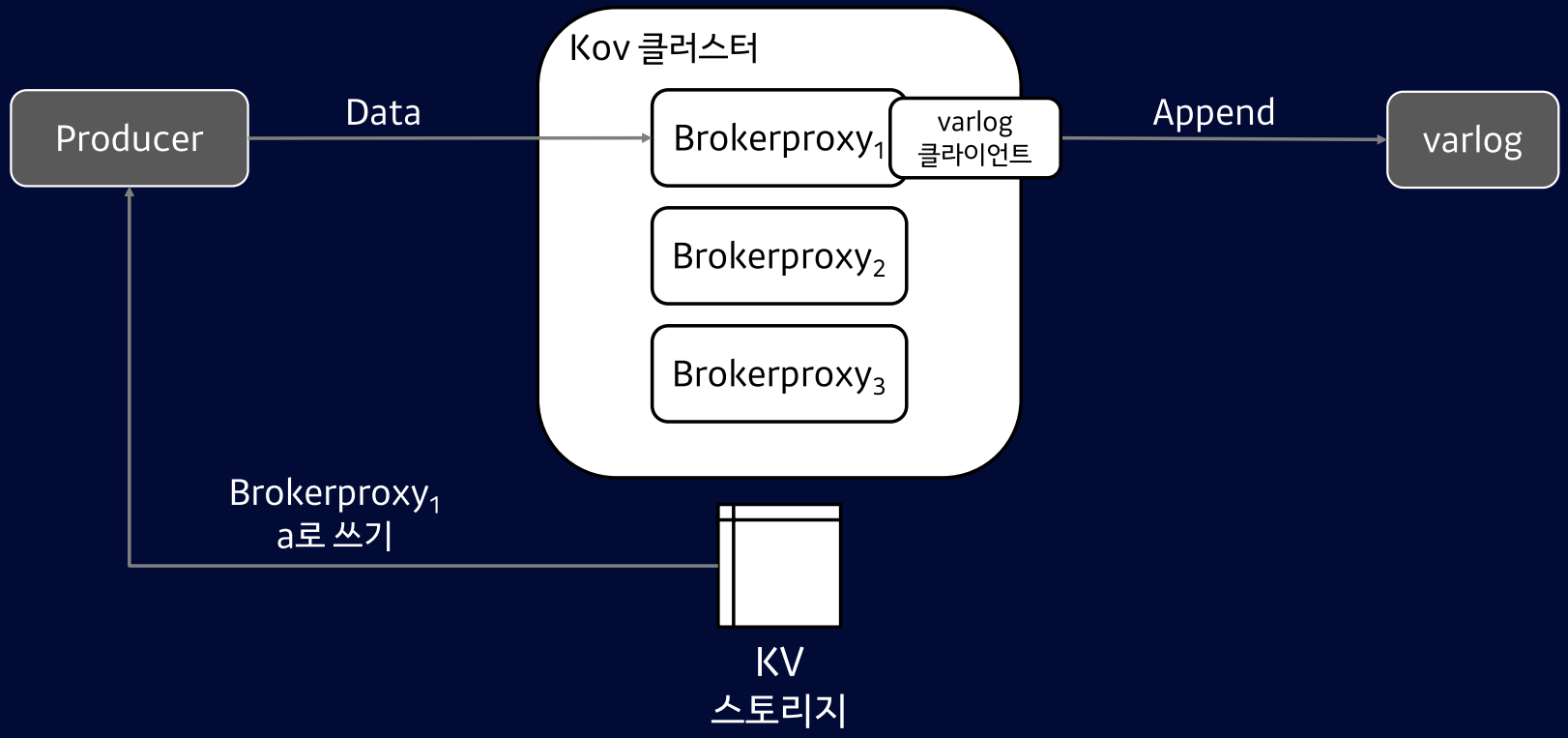
Kov의 Kafka Pub-Sub 요청 처리

Kov는 Pub-Sub에 필요한 메타데이터를 사내 분산 KV 스토리지에 저장시켜 Zookeeper를 대체했습니다.



Kov의 Kafka Pub-Sub 요청 처리

Producer는 KV 스토리지를 통해 어느 Brokerproxy로 Data 쓰기를 요청하면 되는지 알아내 요청하고
요청을 받은 Brokerproxy는 varlog에 데이터를 로그로 남기고 반환 받은 순서를 응답해줍니다.

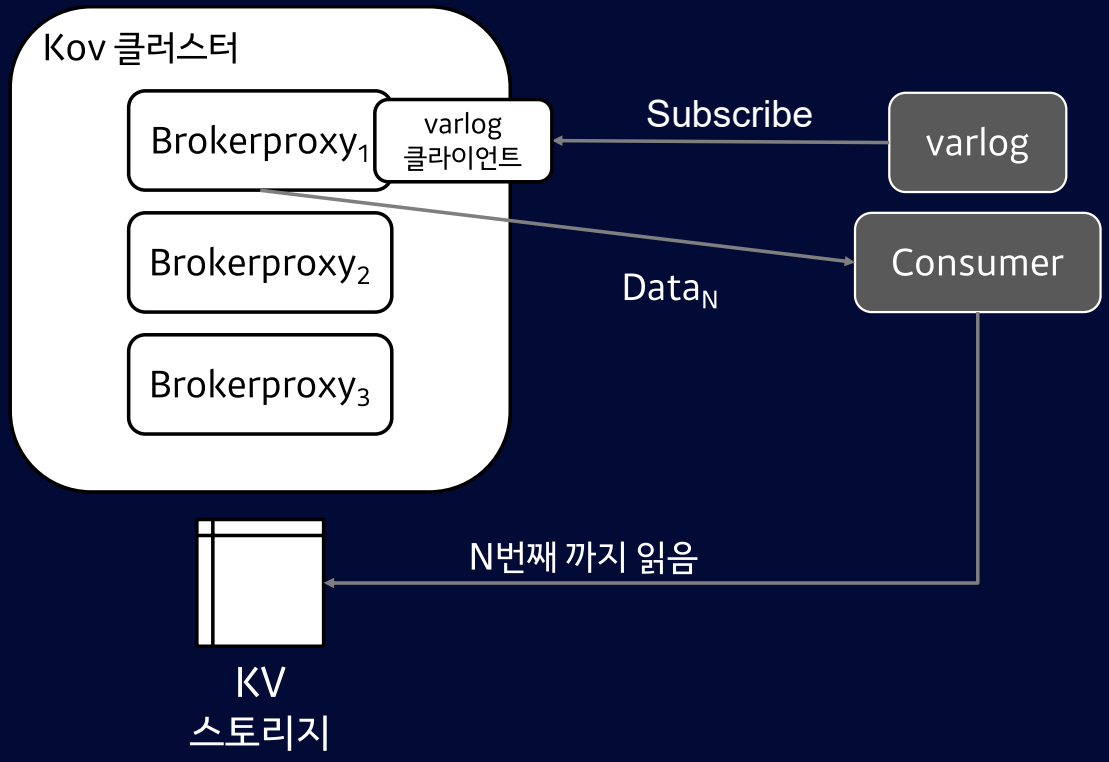


SI0 varlog 클라이언트 에서 나오는 화살표에 append API 사용한다는 표시를 해주는게 좋을까요? (저도 확신이 없네요..)
Song Injun, 2022-10-25T07:21:03.364

YL0 0 음... Data라는 단어 없이 Append라고만 하는게 더 나을 것 같기도 합니다
YeongSik Lee, 2022-10-25T07:23:28.644

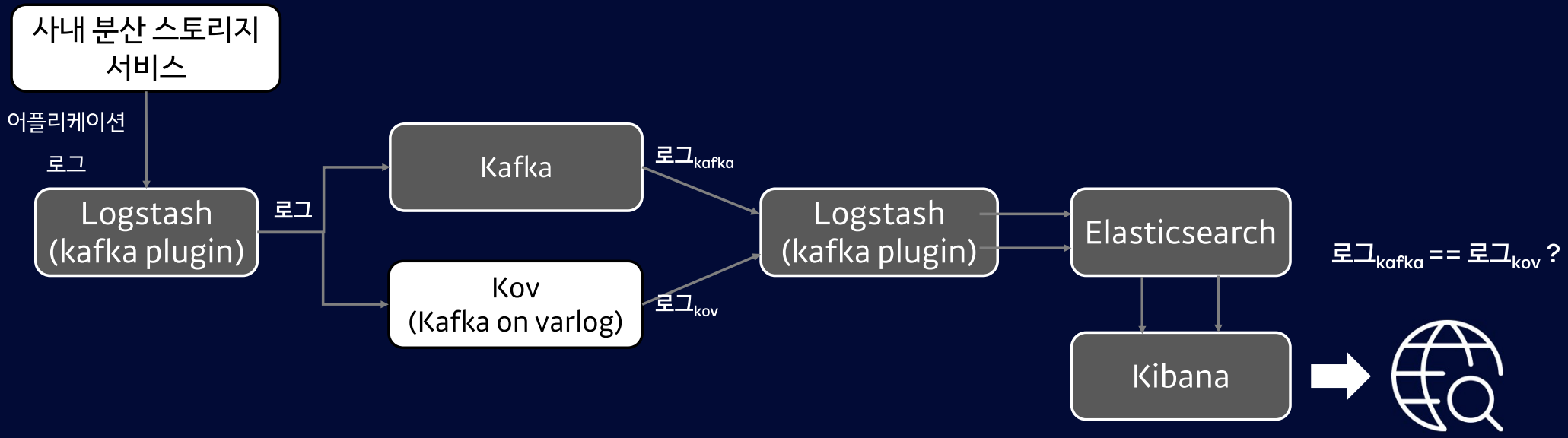
Kov의 Kafka Pub-Sub 요청 처리

Consumer가 Brokerproxy에 N번째 Data 읽기를 요청하면, Brokerproxy는 varlog에서 로그를 읽어 응답해줍니다.
메시지를 읽은 Consumer는 KV 스토리지에 몇 번째 메시지까지 읽었는지 기록합니다.



Kov 기능 확인

사내 분산 스토리지 서비스의 로그를 ELK를 이용해 Kafka와 Kov를 이용해 로그를 두벌로 기록하도록 설정한 후, Kafka와 Kov에 저장된 로그들이 같은 순서로 쓰고 읽혀지는지 확인하는 방식으로 varlog가 문제없이 동작하는지 확인했습니다.



맺음말

분산 로그 저장소를 이용한 서비스 개발을 계획하고 계신다면 varlog[https://github.com/kakao/varlog] 에 많은 관심 부탁드립니다.
varlog에 대한 Contribution 역시 환영합니다!

1. State Machine Replication
2. 어플리케이션 로깅
3. 메세지큐
4. WAL(Write-Ahead Logging)
5. 분산 트랜잭션
6. ...

Any Questions?

Thank you!