Architectural Improvements for a Log Management System

이재익

네이버 서비스플랫폼개발센터

NAVER

목차

- ✓ **네이버 에러/크래시 로그 시스템**
- ✓ **기능 개발 및 아키텍처 개선**
  - 검색 엔진을 활용한 실시간 전문 검색
  - 분산 메시지 큐
  - 실시간 알람

로그의 다양성

Mobile

Desktop

다양성: 포맷/플랫폼

공통성: 수집/저장/검색/분석

```
LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/5, changed state to down
HA_EM-6-LOG: syslog-int-test: Description: Gary
LINK-3-UPDOWN: Interface GigabitEthernet0/5, changed state to down
LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/5, changed state to down
HA_EM-6-LOG: syslog-int-test: Description: Gary
LINK-3-UPDOWN: Interface GigabitEthernet0/5, changed state to up
LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/5, changed state to up
LINK-3-UPDOWN: Interface GigabitEthernet0/14, changed state to down
LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/20, changed state to down
HA_EM-6-LOG: syslog-int-test: Description: Andrew
```

server

NAVER

스케일

일 유입 로그: 15 억건 (1.3 TB)

전체 검색가능 데이타: 960 억건 (126 TB)

7 clusters in production

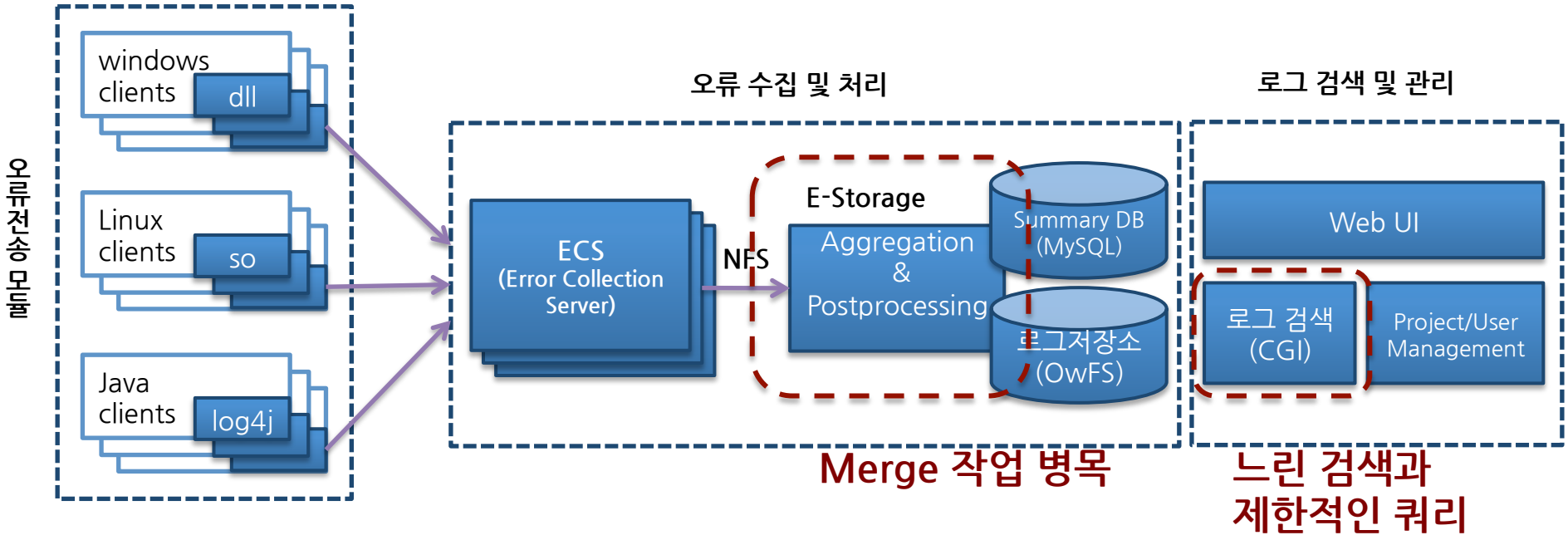
236 서버(H/W)

로그시스템

- ✓ 다양한 SDK
  - 모바일/서버 앱/시스템 로그등 다양한 경로 지원와 플랫폼 지원
- ✓ 전문검색(Full Text Search)
  - 수많은 로그에서 관련 에러를 발견할수 있는 비정형 쿼리 지원
- ✓ 실시간(Real-time) 검색 및 알람
  - 사용자 문의 혹은 시스템 장애에 바로 대처
- ✓ 확장성(Scalability)
  - 로그량 증가에 따른 용량 증설의 용이성
- ✓ 유연한 스키마(Schema free)
  - 다양한 종류의 앱과 플랫폼의 로그를 저장 및 처리할 수 있는 형태
- ✓ 자동 심볼리케이션 및 복호화(deobfuscation)

기존 사내 로그 시스템의 한계

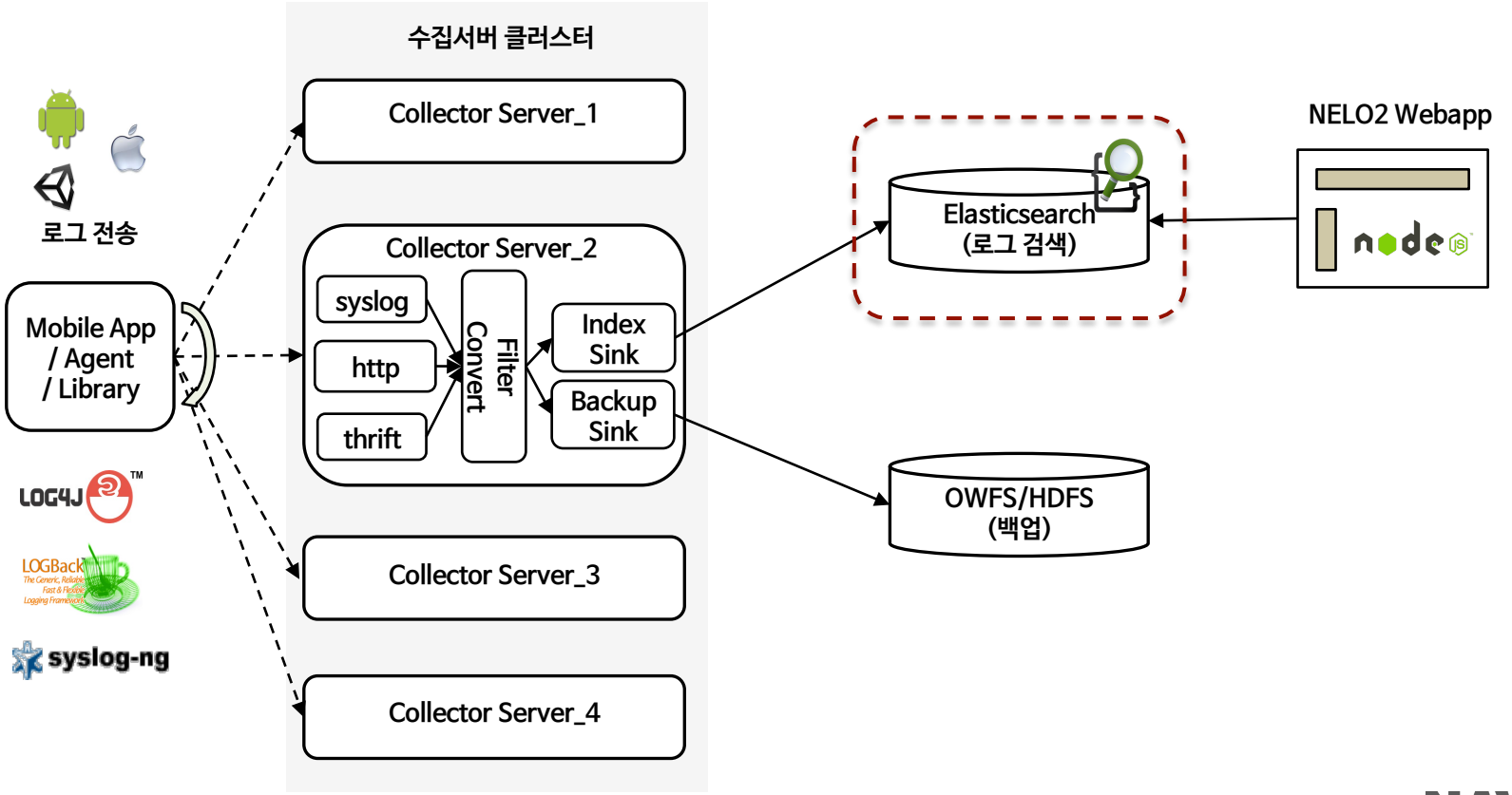
- ✓ 로그량 증가시 수집 서버의 병목
- ✓ 느린 검색성능
- ✓ 제한된 검색 쿼리



NAVER

NELO2

- ✓ Thrift 기반 다양한 플랫폼의 SDK 제공
- ✓ Elasticsearch를 활용한 실시간 전문 검색
- ✓ 로그량 증가에 따른 쉬운 시스템 확장



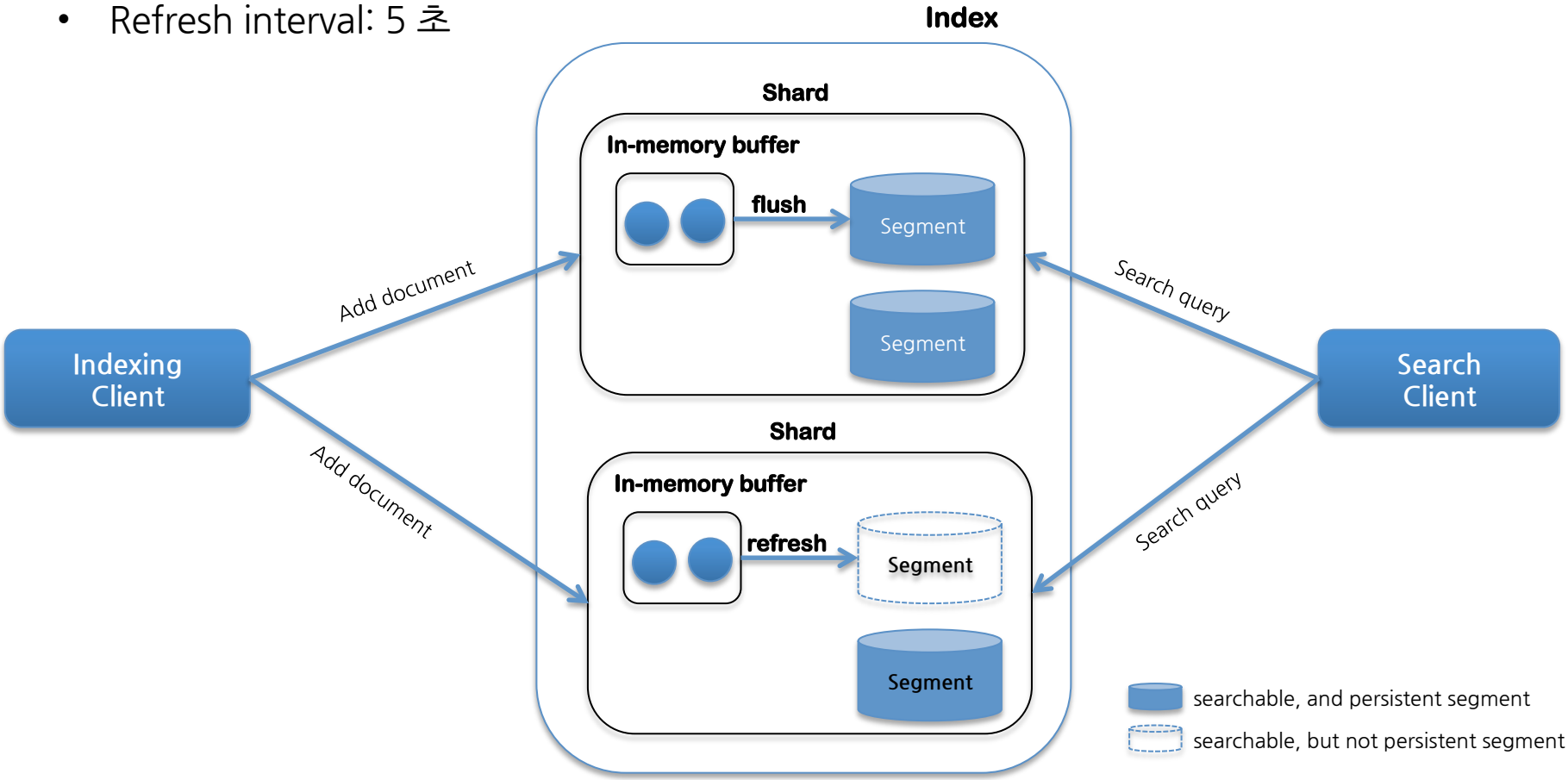
NAVER

Full Text Search

- ✓ Elasticsearch: Apache Lucene 기반 분산 검색 엔진
- ✓ 모든 로그에 대한 Elasticsearch 인덱스 생성 -> Inverted Index 생성
- ✓ Lucene 쿼리 기반으로 다양한 쿼리 제공
  - 필드 기반 검색
    - projectName:"IDSlog" AND body:"Global_POP_BCS"
  - Boolean 쿼리
    - AND, "+"(MUST include), OR, NOT and "-"(MUST NOT include)
  - Wildcard 쿼리
    - projectName:"NELOlog" AND Sensor:LINENcloud*
  - RegEx 쿼리
    - host:/TRNLD100(1|2|3)/
  - Range
    - projectName:"IDSlog" AND port:[0 TO 10080]

Near Real-time Search

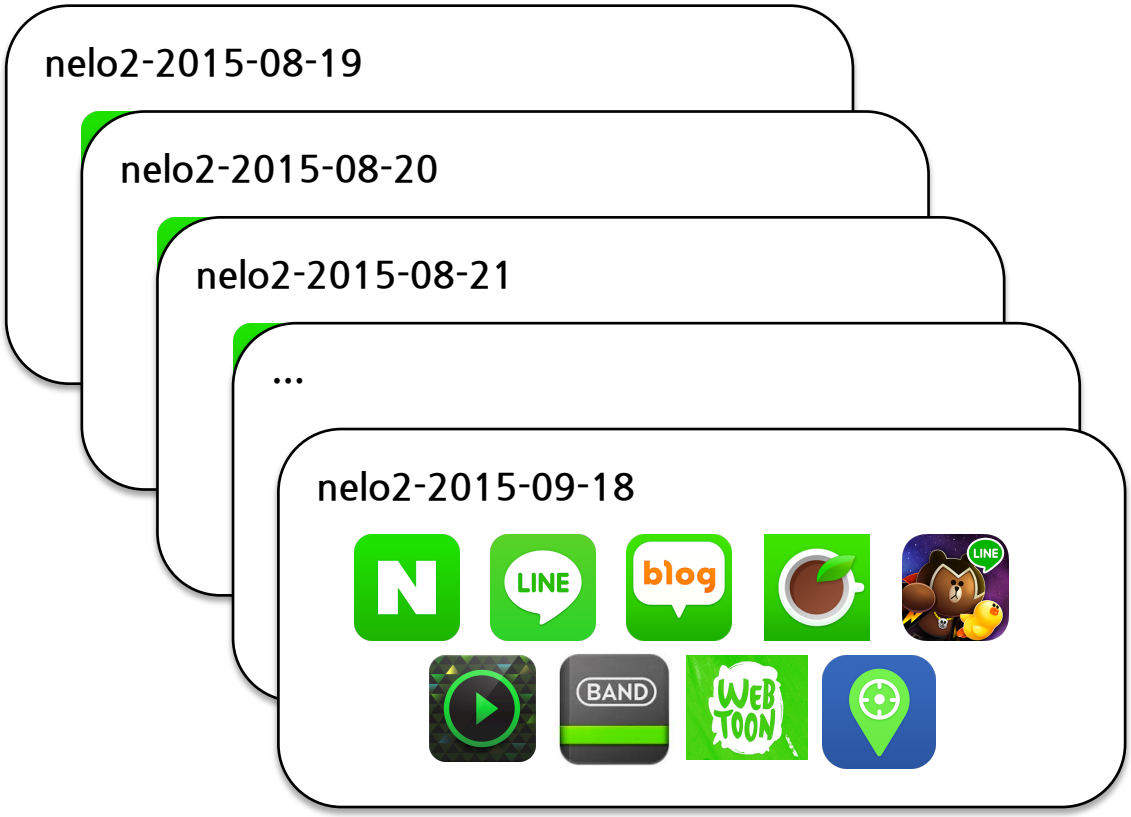
- ✓ Lucene 증분 검색: 인덱스를 Segment로 나누어 증분 검색이 가능
- ✓ Refresh 오퍼레이션을 사용하면 메모리상의 segment를 검색 가능
  - Refresh interval: 5 초



NAVER

인덱스 모델

- ✓ 일별 1개의 인덱스 생성
- ✓ 프로젝트별로 타입 생성
- ✓ 주기적으로 오래된 인덱스 삭제 (인스턴스별: 1달, 3달, 2년, 5년)



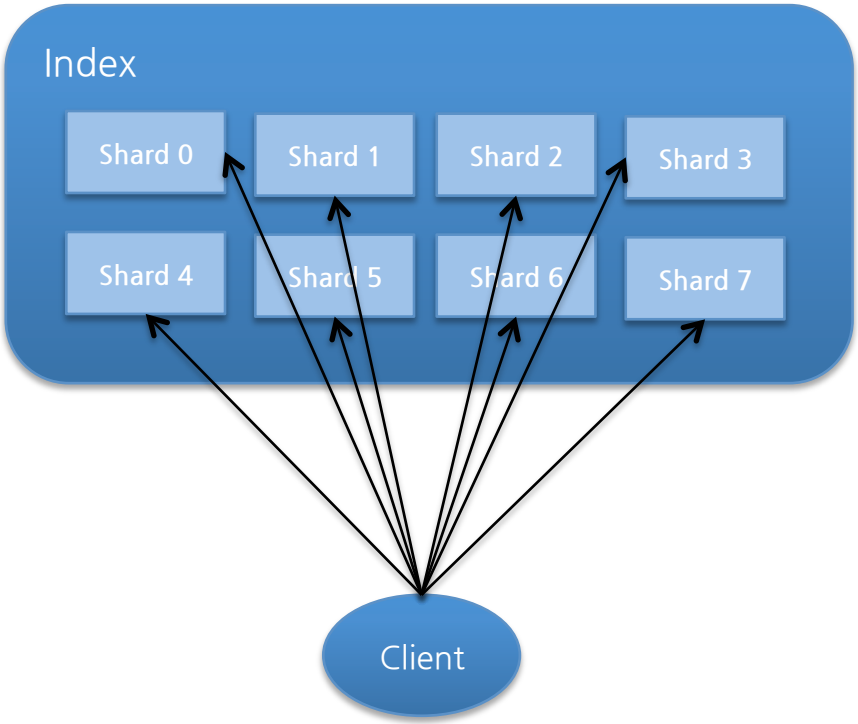
NAVER

성능향상을 위한 라우팅

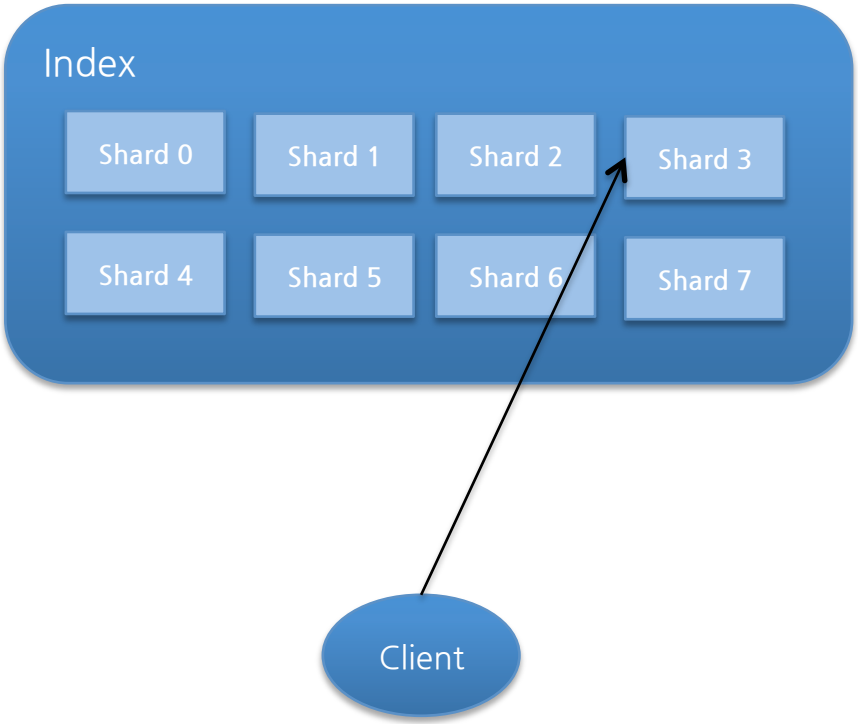
✓ 라우팅

- 샤드 위치 결정
  - shard = hash(routing) % number_of_primary_shards
- Default: document id, “routing” 파라미터: 사용자 지정 라우팅

routing 사용 안함



routing 사용

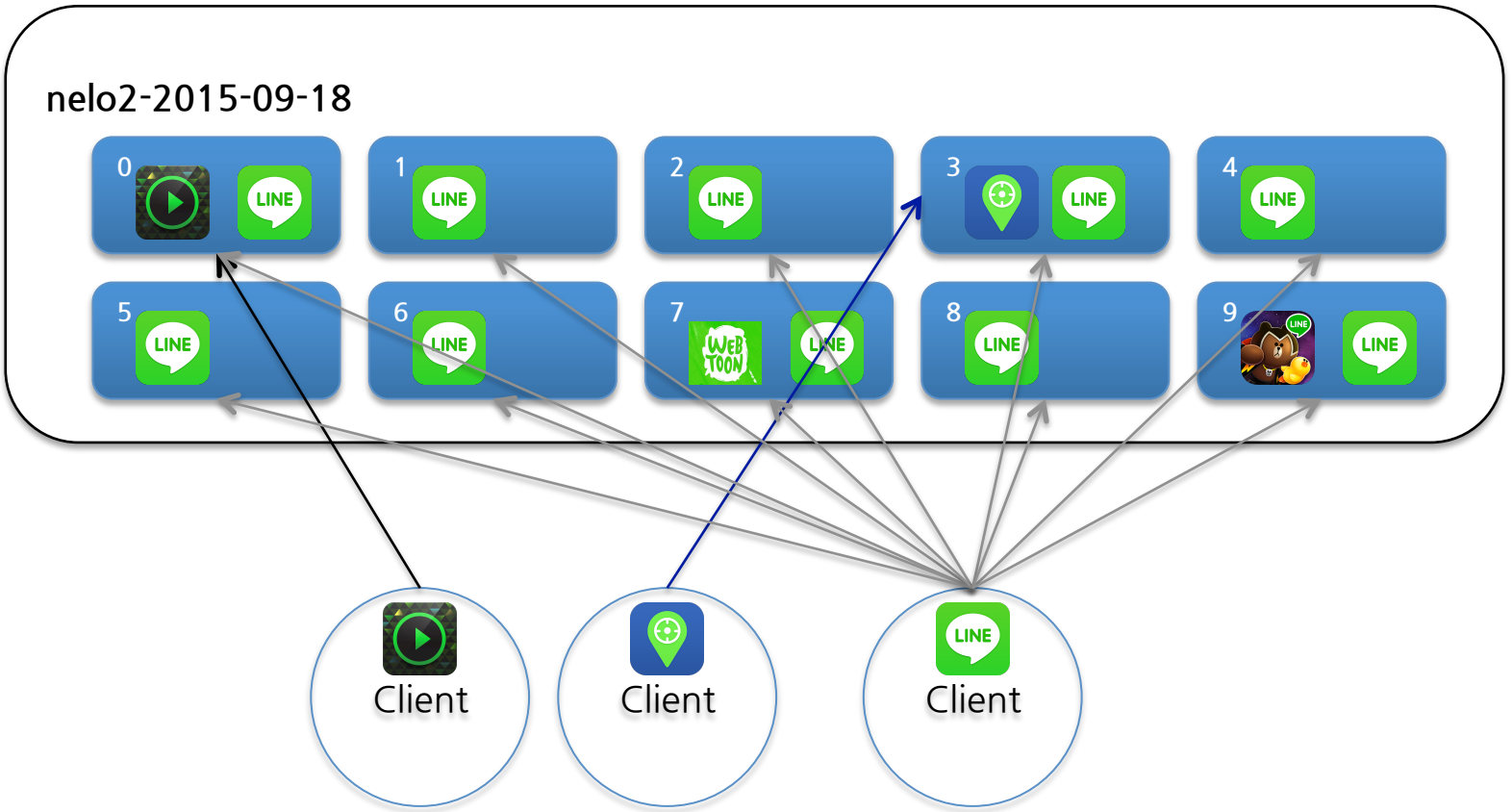


NAVER

사용자 지정 라우팅

✓ 사용자지정 라우팅 (custom routing)

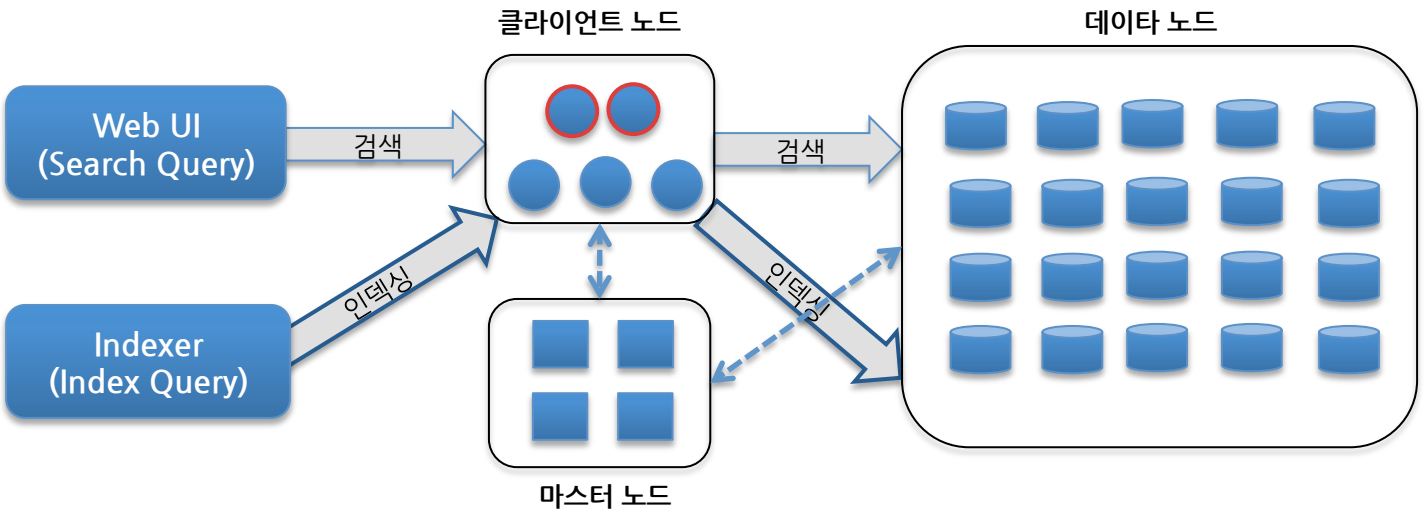
- 작은 프로젝트는 사용: 하나의 샤드에 저장 (라우팅 키: projectName)
- 큰 프로젝트는 사용 안함: 전체 샤드에 나누어 저장 (default routing: doc id)



NAVER

검색 클러스터 구성

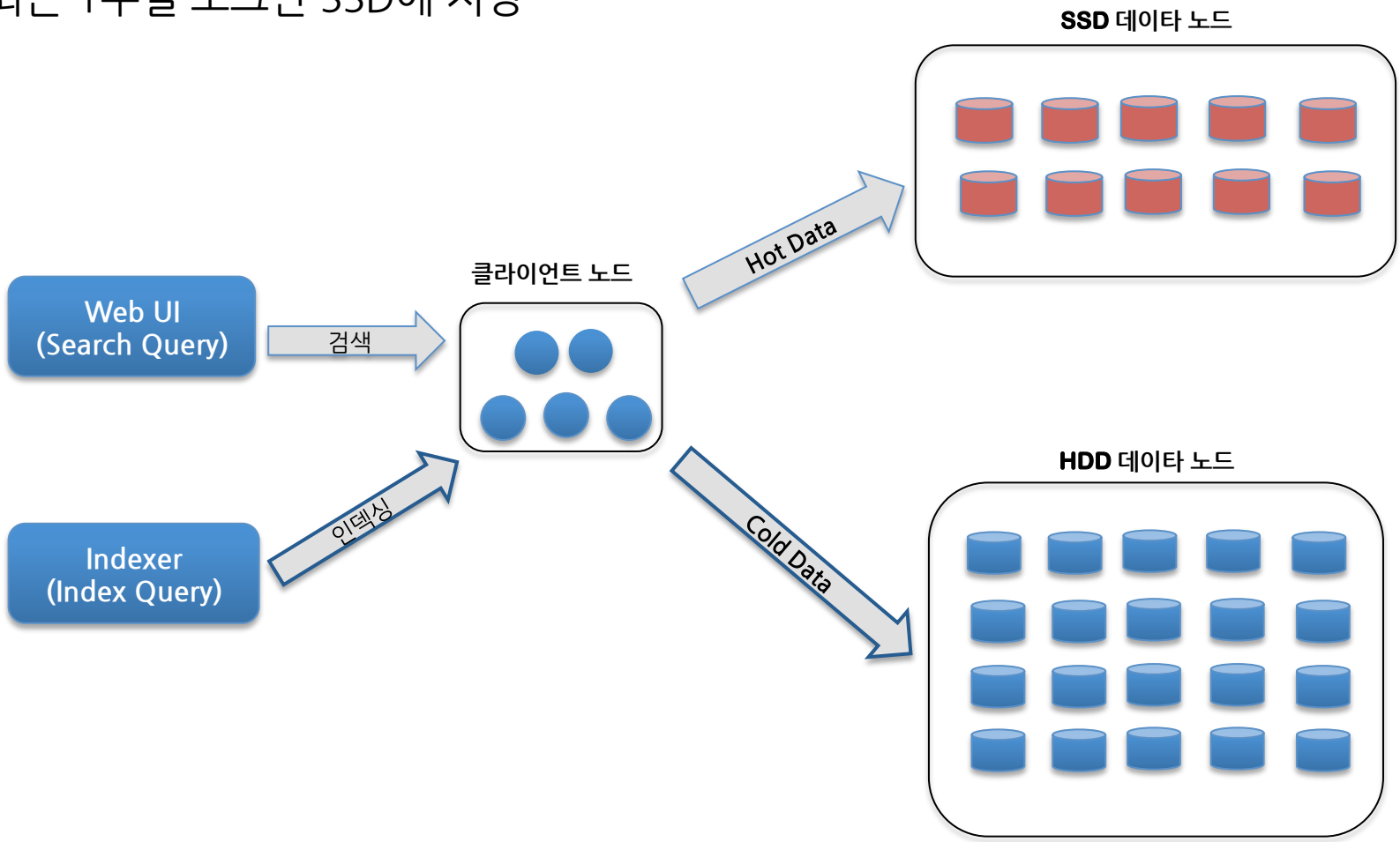
- ✓ 마스터 노드 (node.master: true): 멤버쉽 관리, 메타데이타 저장
- ✓ 데이터 노드 (node.data: true): 데이터 저장 및 프로세싱
- ✓ 클라이언트 노드 (node.master: false, node.data: false): 로드발란서



NAVER

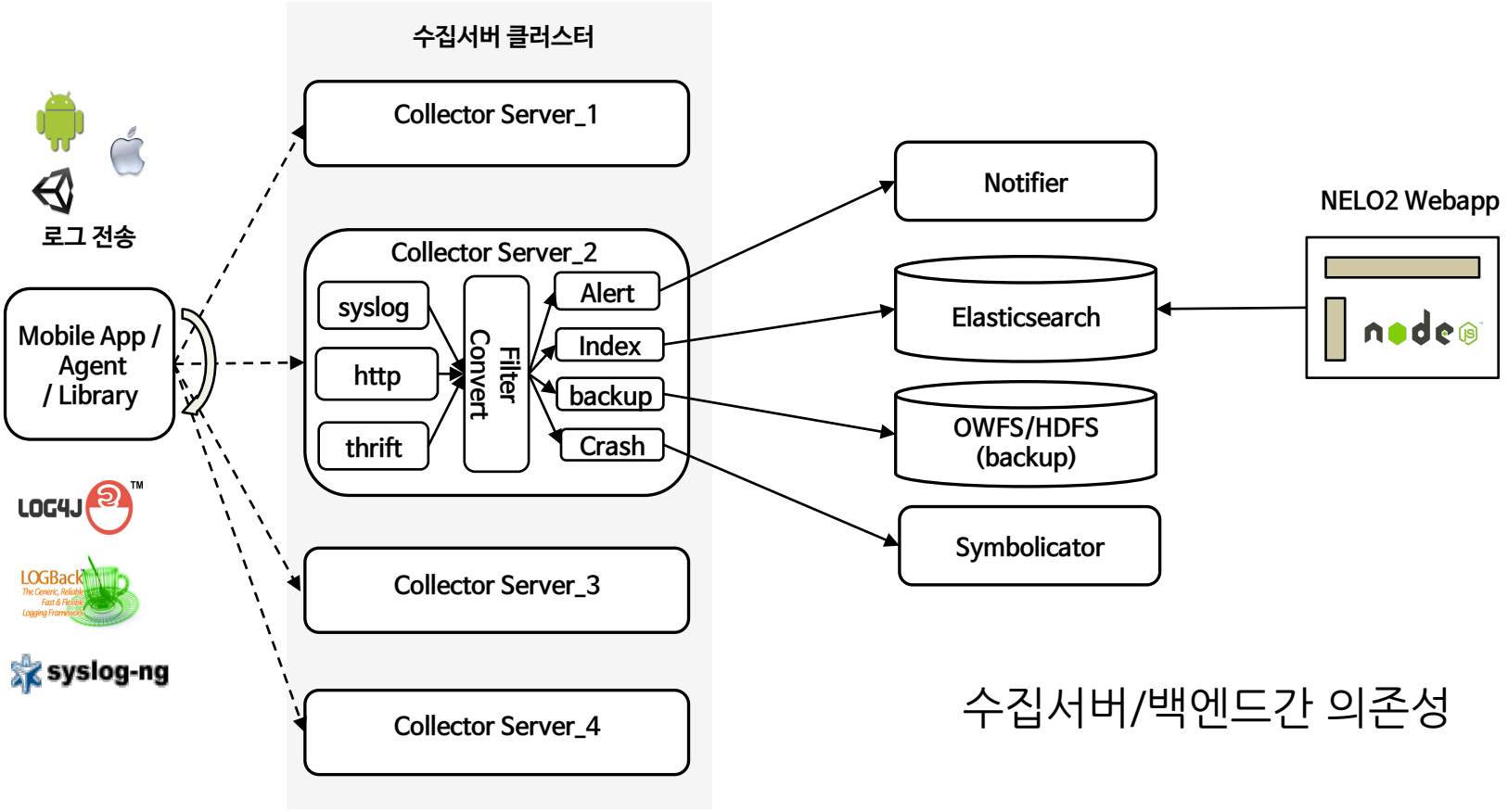
성능 향상을 위한 SSD Layering

- ✓ Tag기반 노드 및 index affinity
- ✓ 최근 1주일 로그만 SSD에 저장



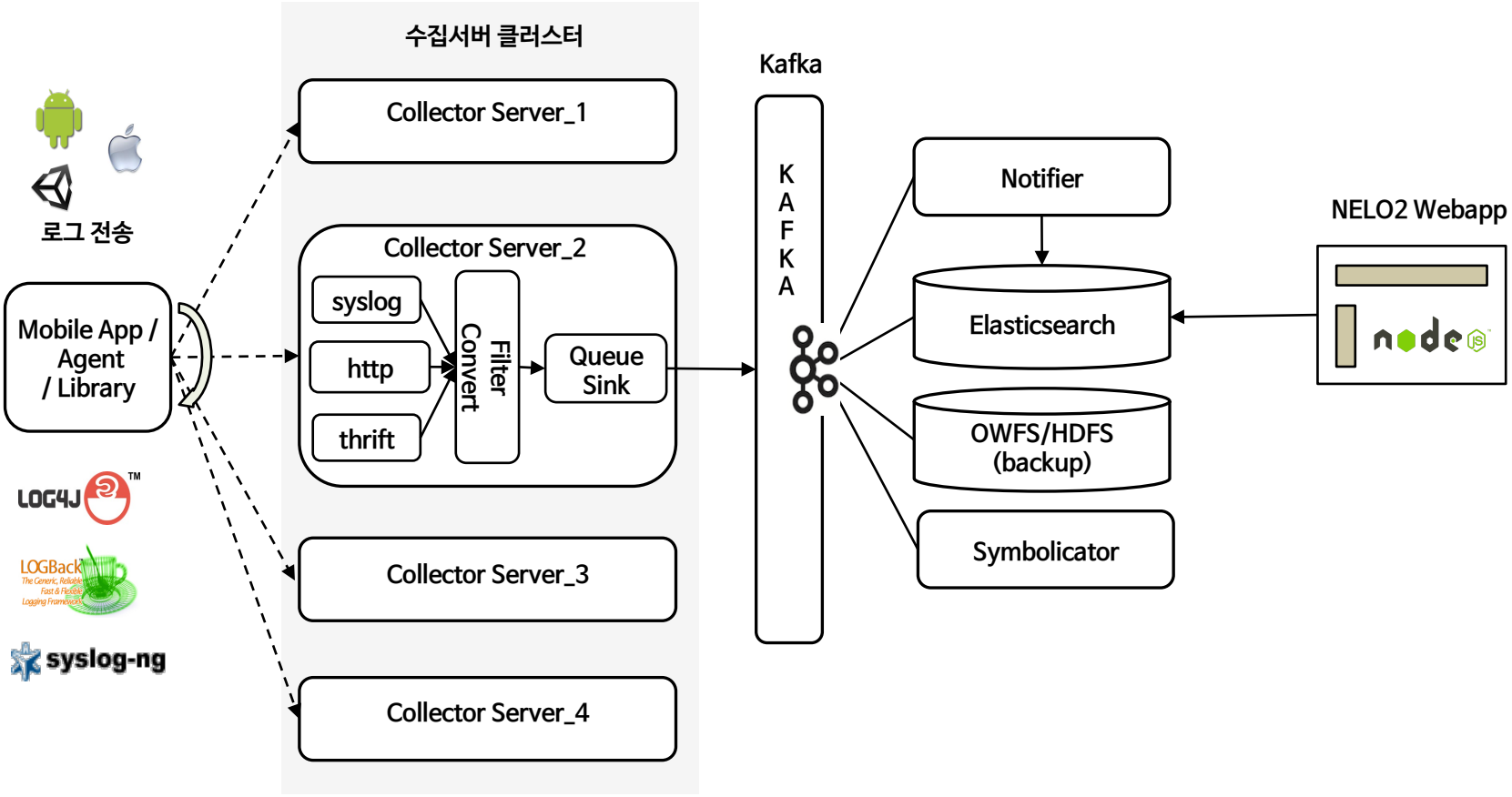
NAVER

백엔드 변경 및 장애시 문제



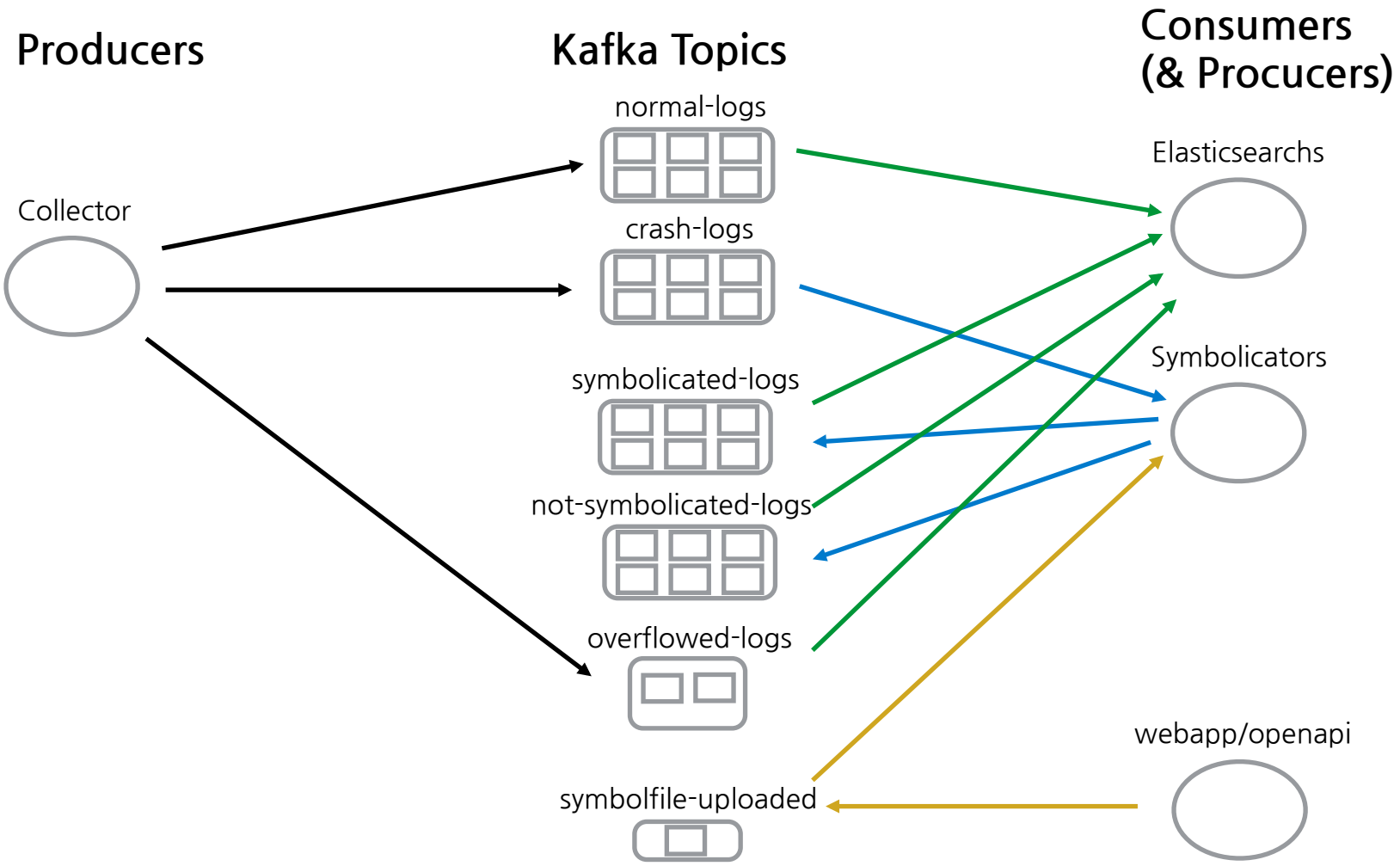
NAVER

분산 메시지 큐 Kafka 적용



NAVER

NELO2의 Kafka Topic 모델



NAVER

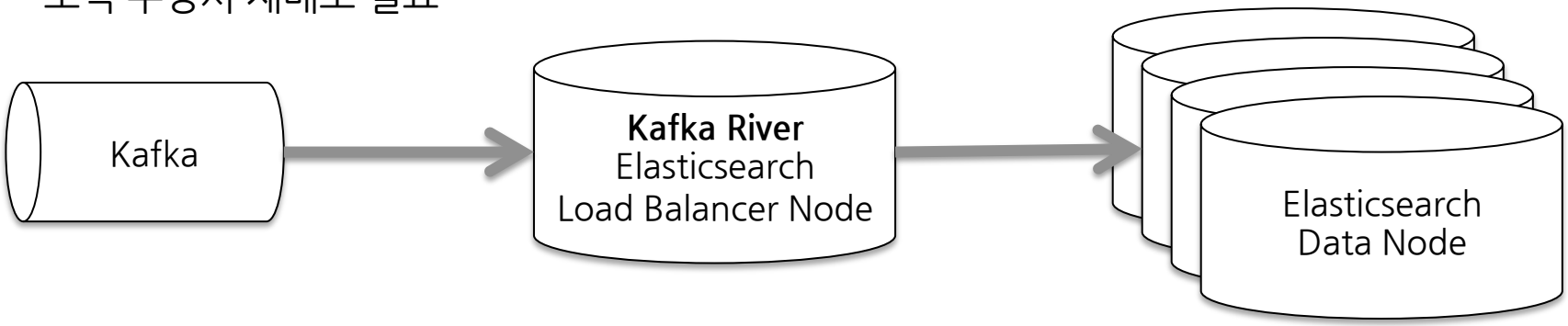
Elasticsearch Kafka River 플러그인 도입과 문제점

✓ Elasticsearch Kafka River 플러그인

- Kafka의 데이타를 읽어서 Elasticsearch에 인덱싱

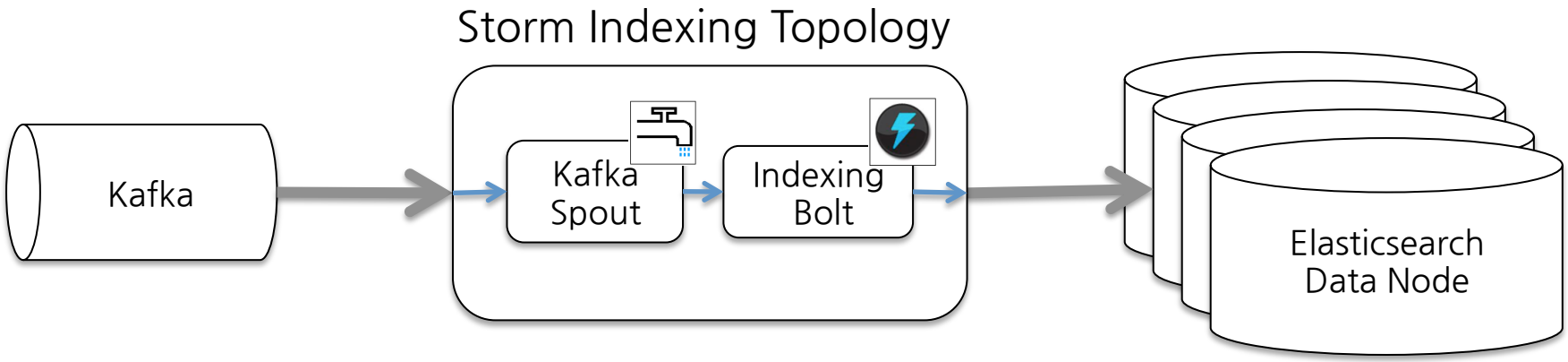
✓ 문제점

- 인덱싱 성능
- 불안정한 동작
- 디버깅의 어려움
- 로직 수정시 재배포 필요



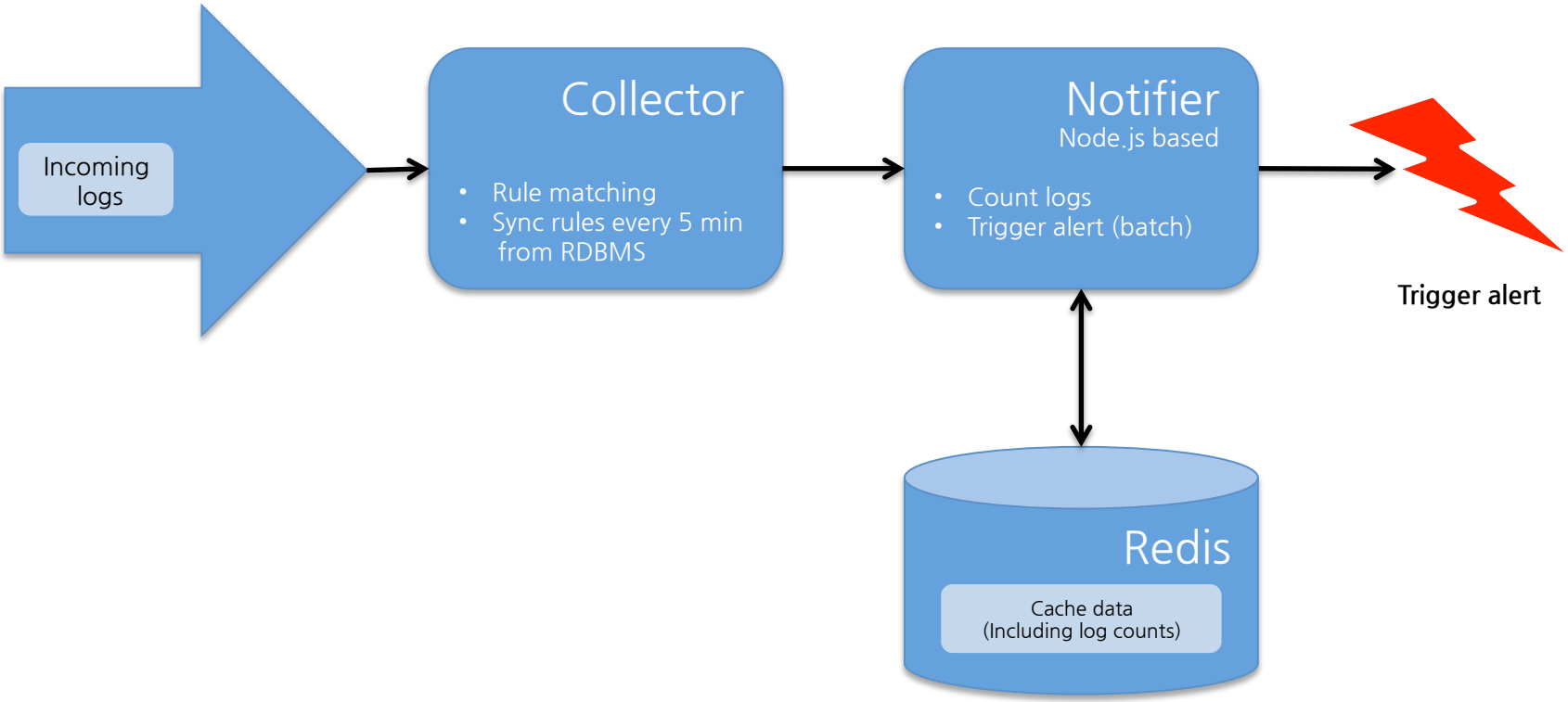
Storm을 활용한 인덱싱

- ✓ High throughput
  - 1,000,000 message/s (100 byte)
- ✓ 메시지 처리 보장
- ✓ 확장성



기존 배치 알람

- ✓ 실시간 알람 전달이 안됨
- ✓ 검색 쿼리와 알람 룰이 다름

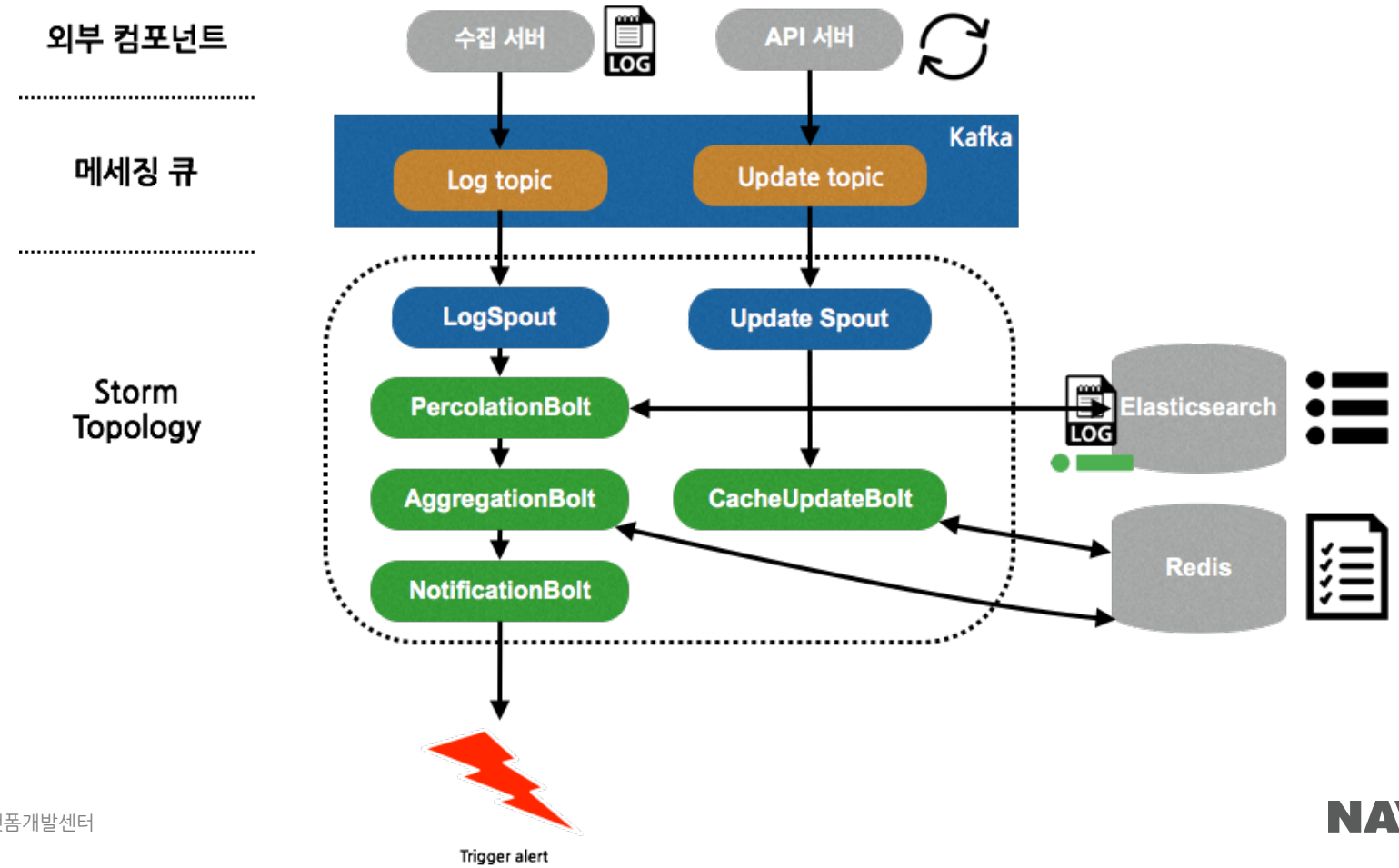


NAVER

배치 알람 -> 실시간 알람

✓ Elasticsearch Percolator와 Storm 활용

- 검색 쿼리를 알람룰로 저장
- 실시간 알람 집계 및 전송



NAVER

요약

- ✓ 검색엔진(Elasticsearch)을 사용하여 실시간으로 Full Text 로그 검색 기능 개발
- ✓ Custom Routing과 SSD Layering을 통해 검색 성능 개선
- ✓ 분산 메시지 큐(Kafka)를 사용해서 시스템의 안정성 확보
- ✓ Percolator와 Storm을 활용해서 실시간 알람 시스템 개발

참고자료

- ✓ Elasticsearch Near-real time 검색
  - https://www.elastic.co/guide/en/elasticsearch/guide/current/near-real-time.html
- ✓ Rafal Kuc의 성능 튜닝 관련된 Berlin Buzzwords 2012 발표 자료
  - http://www.elasticsearch.org/videos/2012/06/05/scaling-massive-elasticsearch-clusters.html
- ✓ Elasticsearch로 로그 검색 시스템 만들기
  - http://d2.naver.com/helloworld/273788
- ✓ NELO2 실시간 알람: Storm과 Elasticsearch를 활용한 로깅 플랫폼의 실시간 알람 시스템 구현
  - http://www.slideshare.net/deview/241-storm-elasticsearch

Q&A

NAVER