2022 SK TECH SUMMIT

셀프서비스 Stream Designer로 나의 시스템을 더욱 강력하게

(ksqlDB기반 사용자 Self Stream Processing 플랫폼 서비스)

SK Hynix 박 재 승

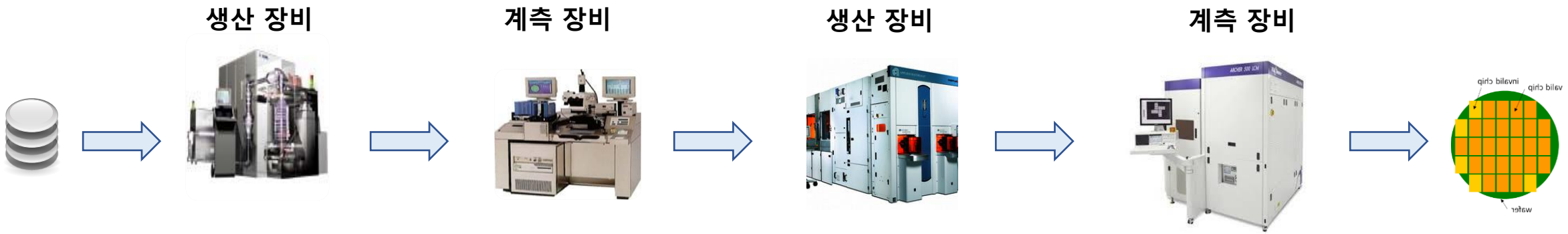
박재승 SK Hynix

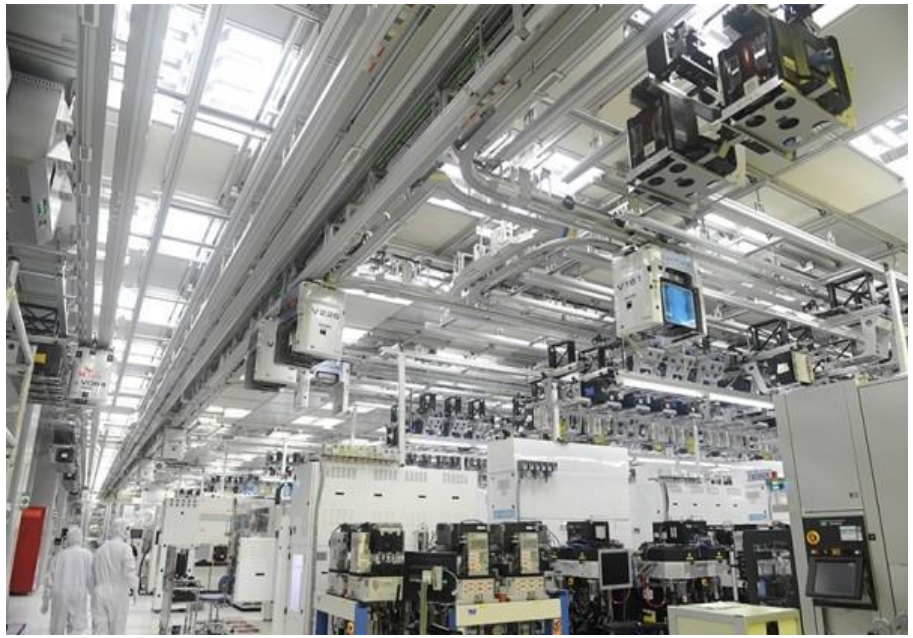
실시간 Data Platform / Data Catalog System 개발 리딩

기술영역 I Data Engineer

Contents

반도체 실시간 데이터



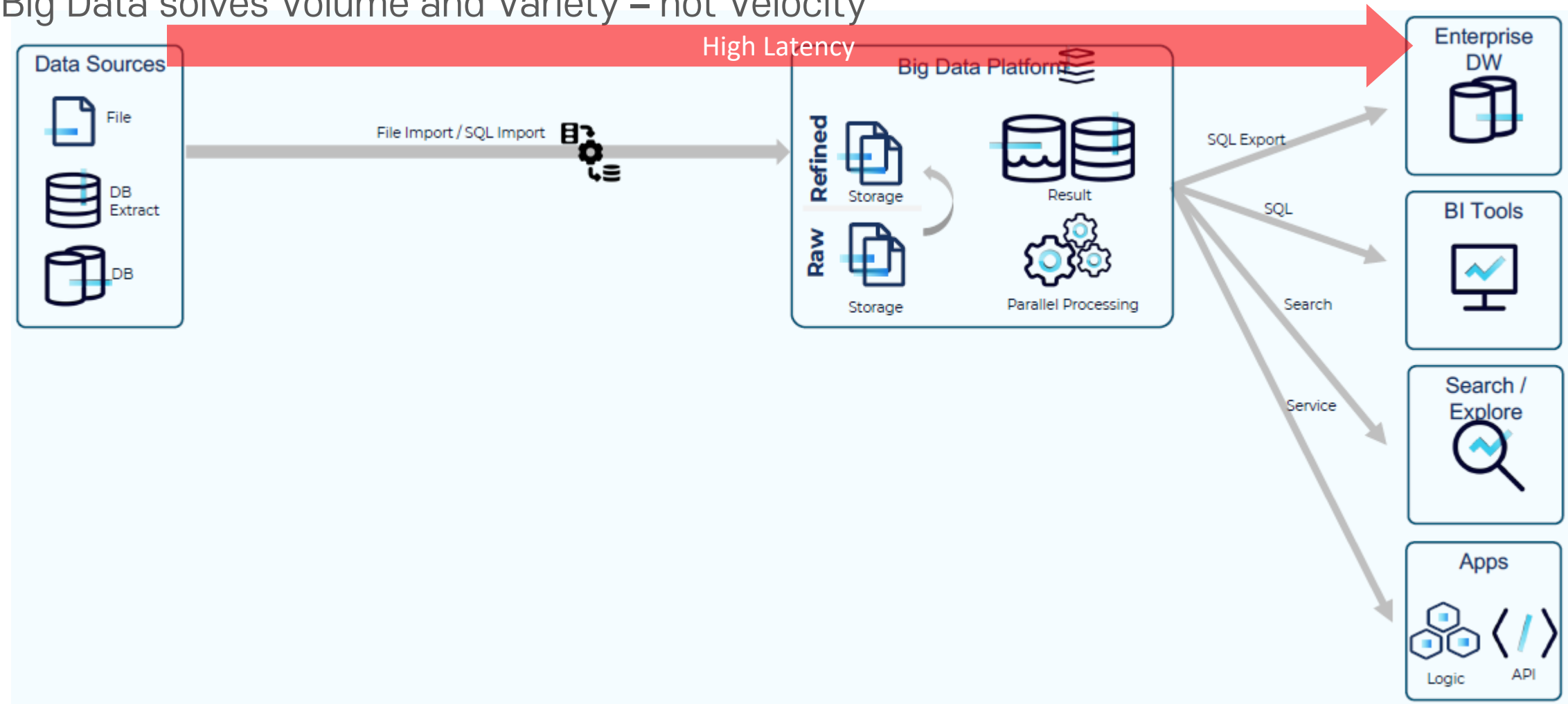


27,000 EPS

54 MB / sec

What's Realtime Platform?

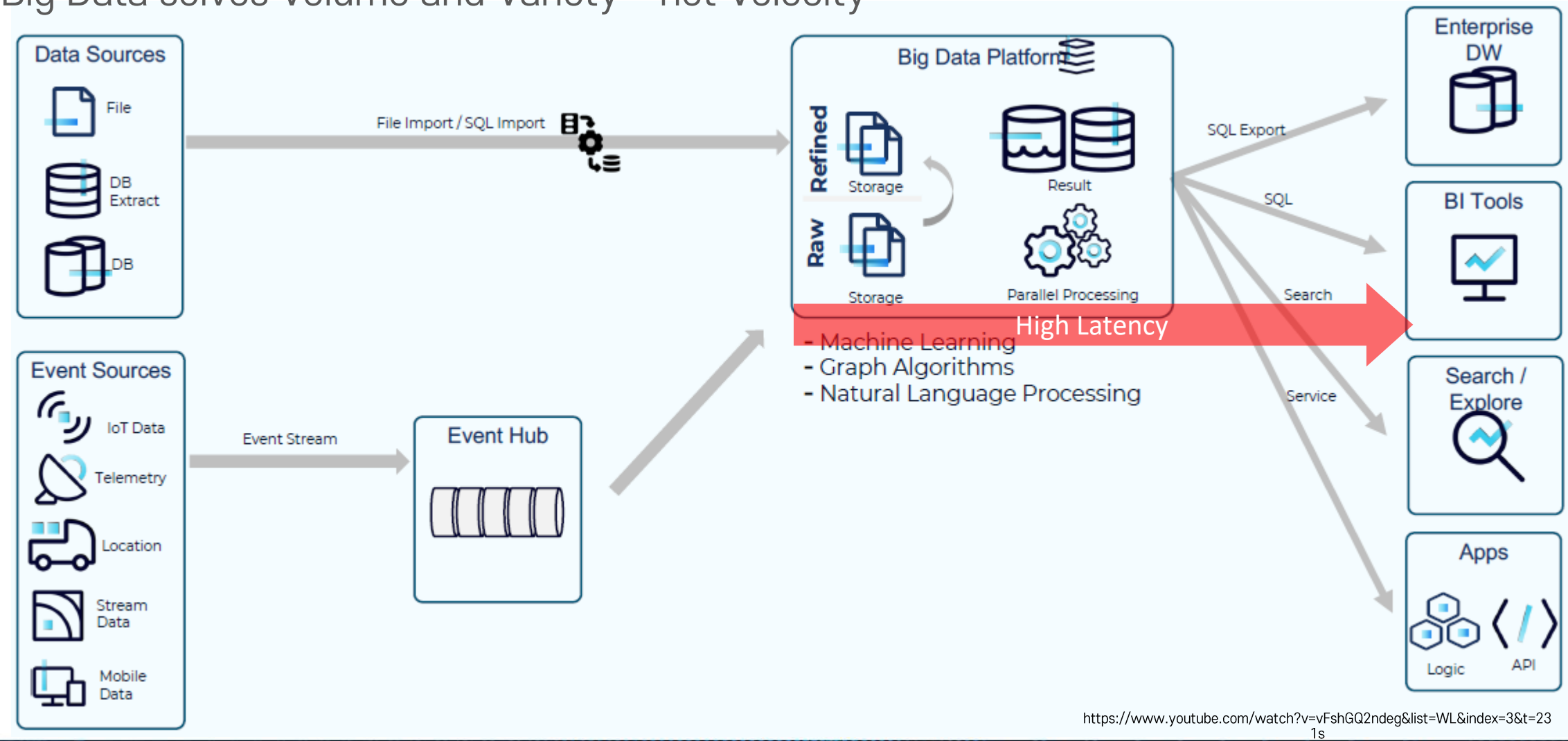
Big Data solves Volume and Variety – not Velocity



https://www.youtube.com/watch?v=vFshGQ2ndeg&list=WL&index=3&t=231s

What's Realtime Platform?

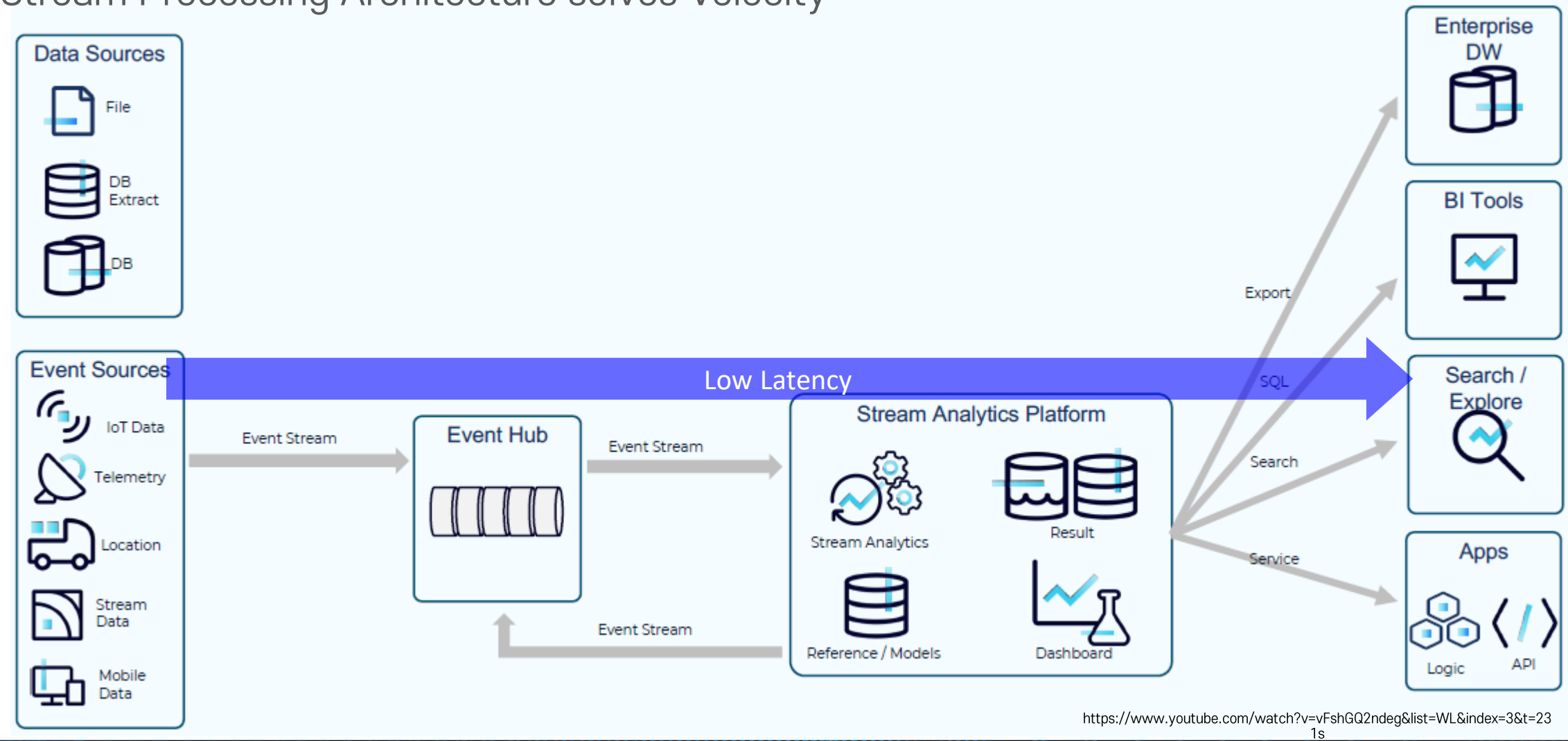
Big Data solves Volume and Variety – not Velocity



https://www.youtube.com/watch?v=vFshGQ2ndeg&list=WL&index=3&t=231s

What's Realtime Platform?

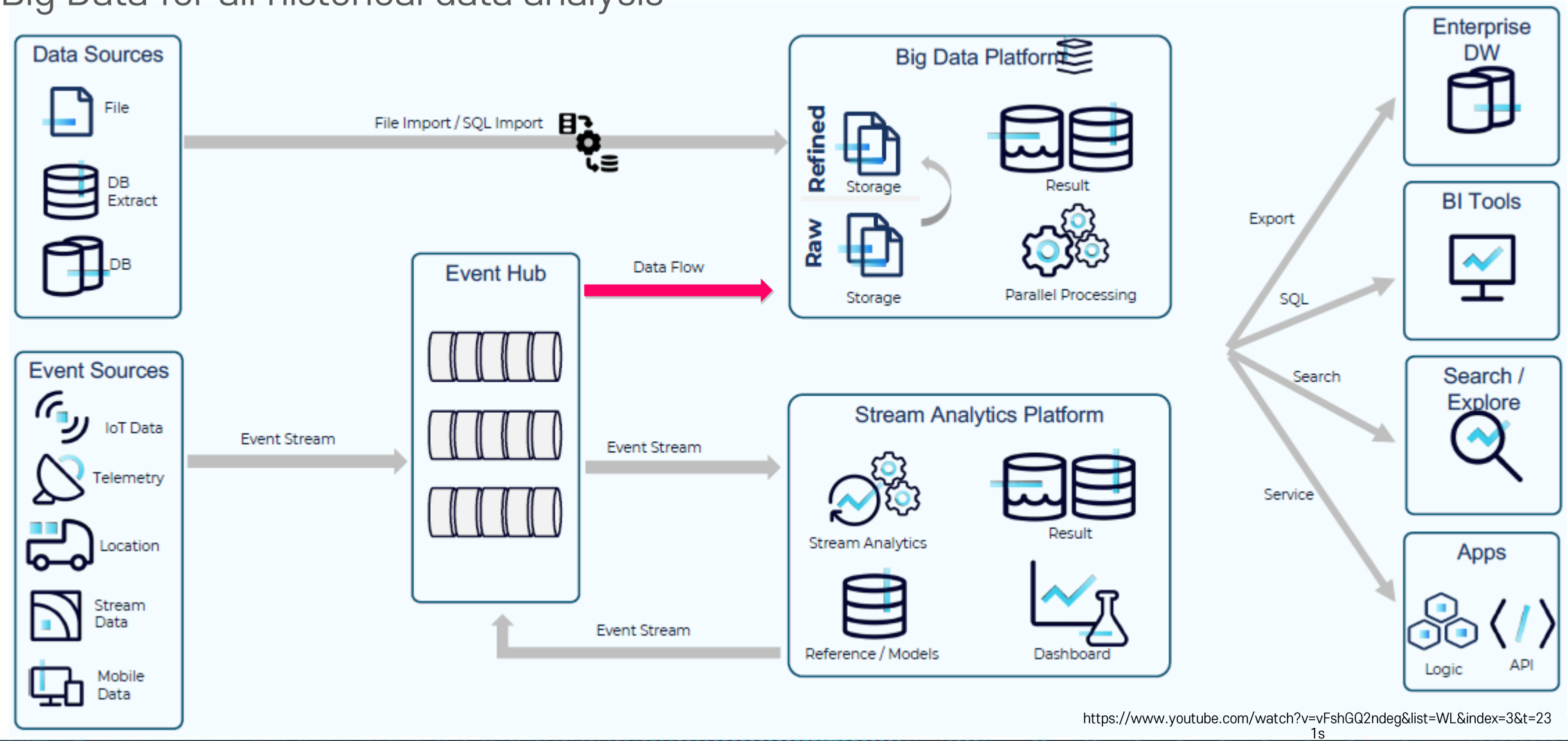
Stream Processing Architecture solves Velocity



https://www.youtube.com/watch?v=vFshGQ2ndeg&list=WL&index=3&t=231s

What's Realtime Platform?

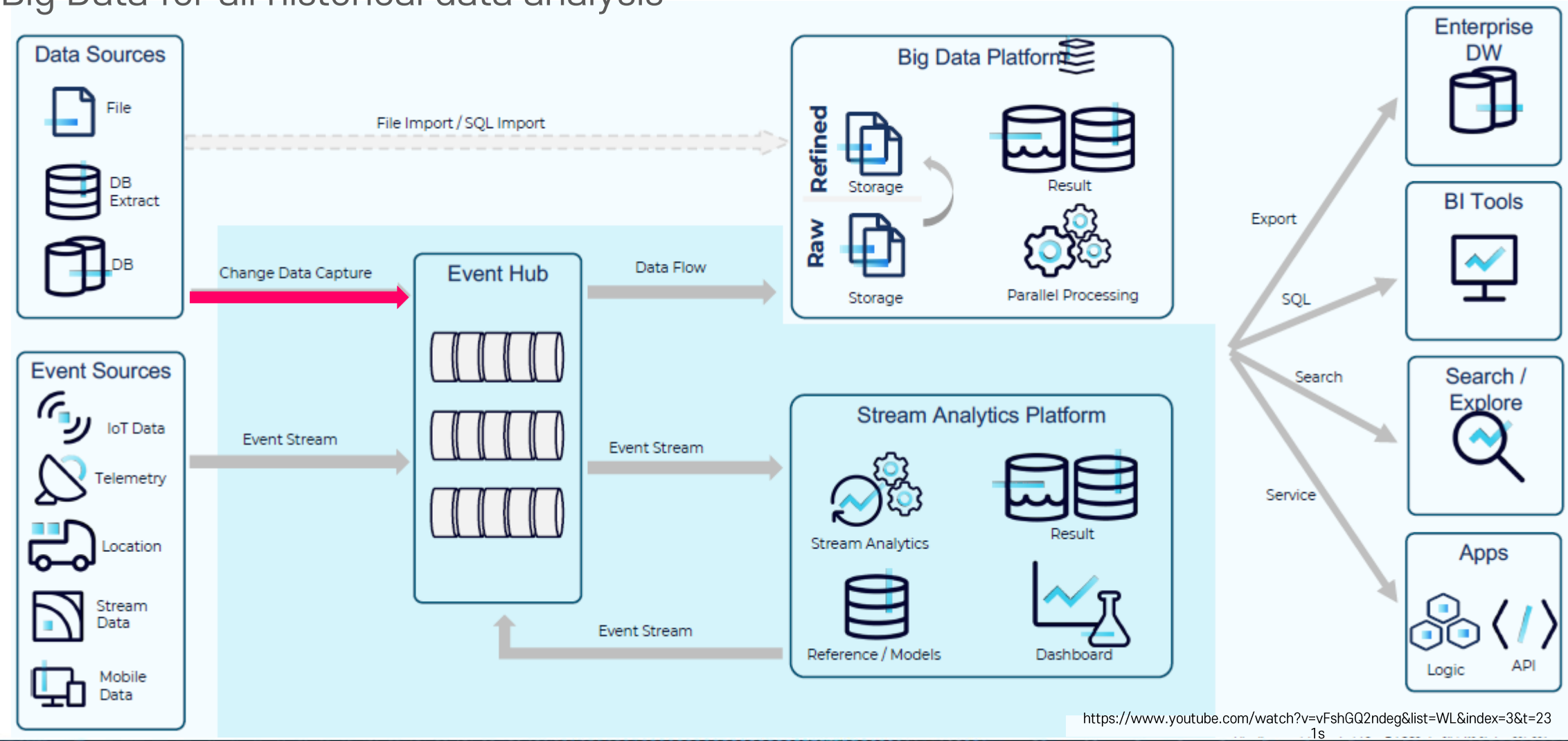
Big Data for all historical data analysis



https://www.youtube.com/watch?v=vFshGQ2ndeg&list=WL&index=3&t=231s

What's Realtime Platform?

Big Data for all historical data analysis



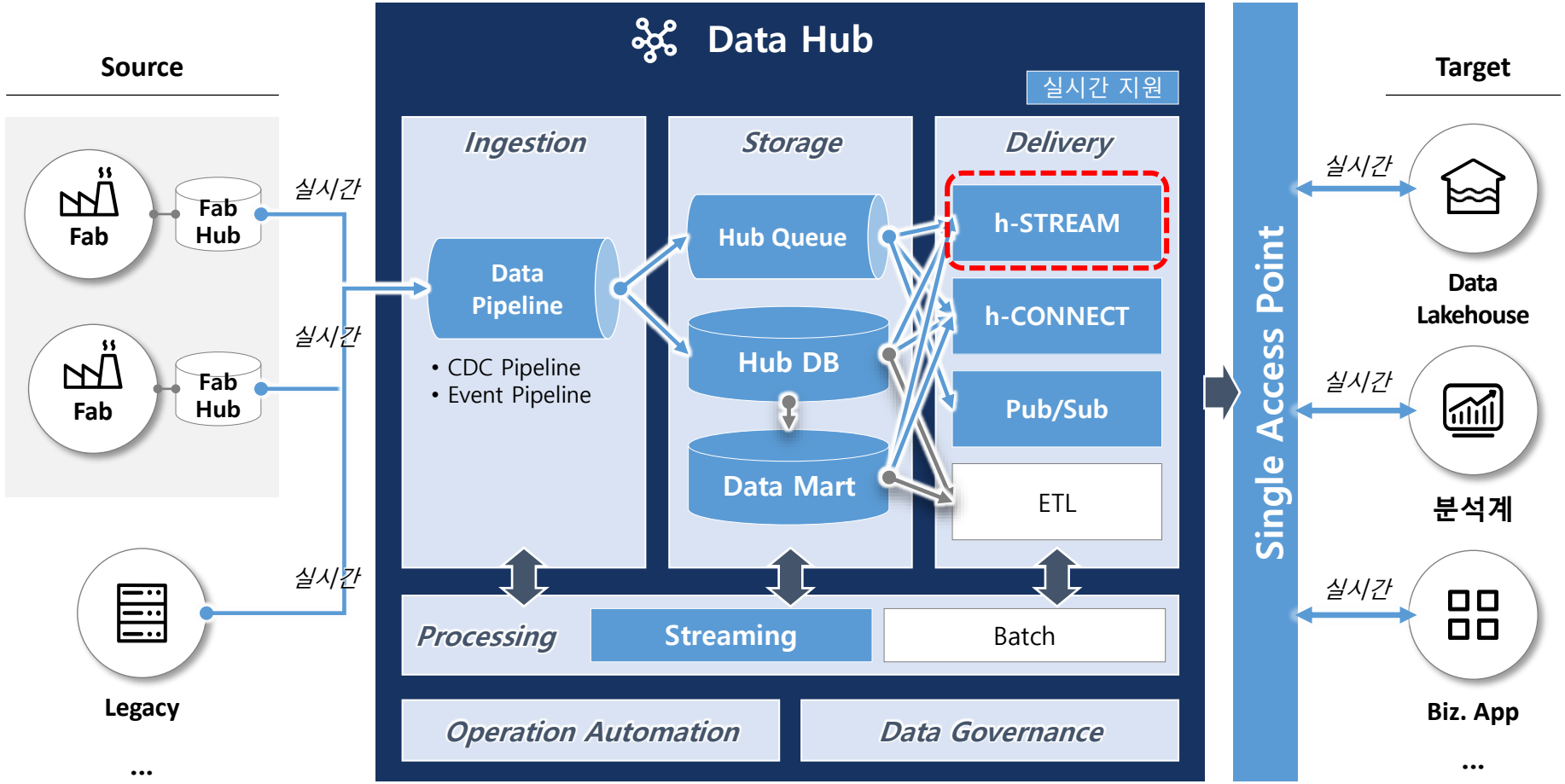
https://www.youtube.com/watch?v=vFshGQ2ndeg&list=WL&index=3&t=231s

Data Hub

Data Hub는 **전사 데이터 연계를 위한** Hub **역할 수행**
Source**와** Target **간 실시간 데이터** Pipeline**으로서 데이터 활용의 신속/적시성을 지원**

Data Hub

Data Hub는 **전사 데이터 연계를 위한** Hub **역할 수행**
Source**와** Target **간 실시간 데이터** Pipeline**으로서 데이터 활용의 신속/적시성을 지원**



h-STREAM

손쉽게 실시간 데이터를 **확인**하고 싶어요

실시간 데이터를 **조작**해서 **나만의 실시간 데이터**를 만들고 싶어요

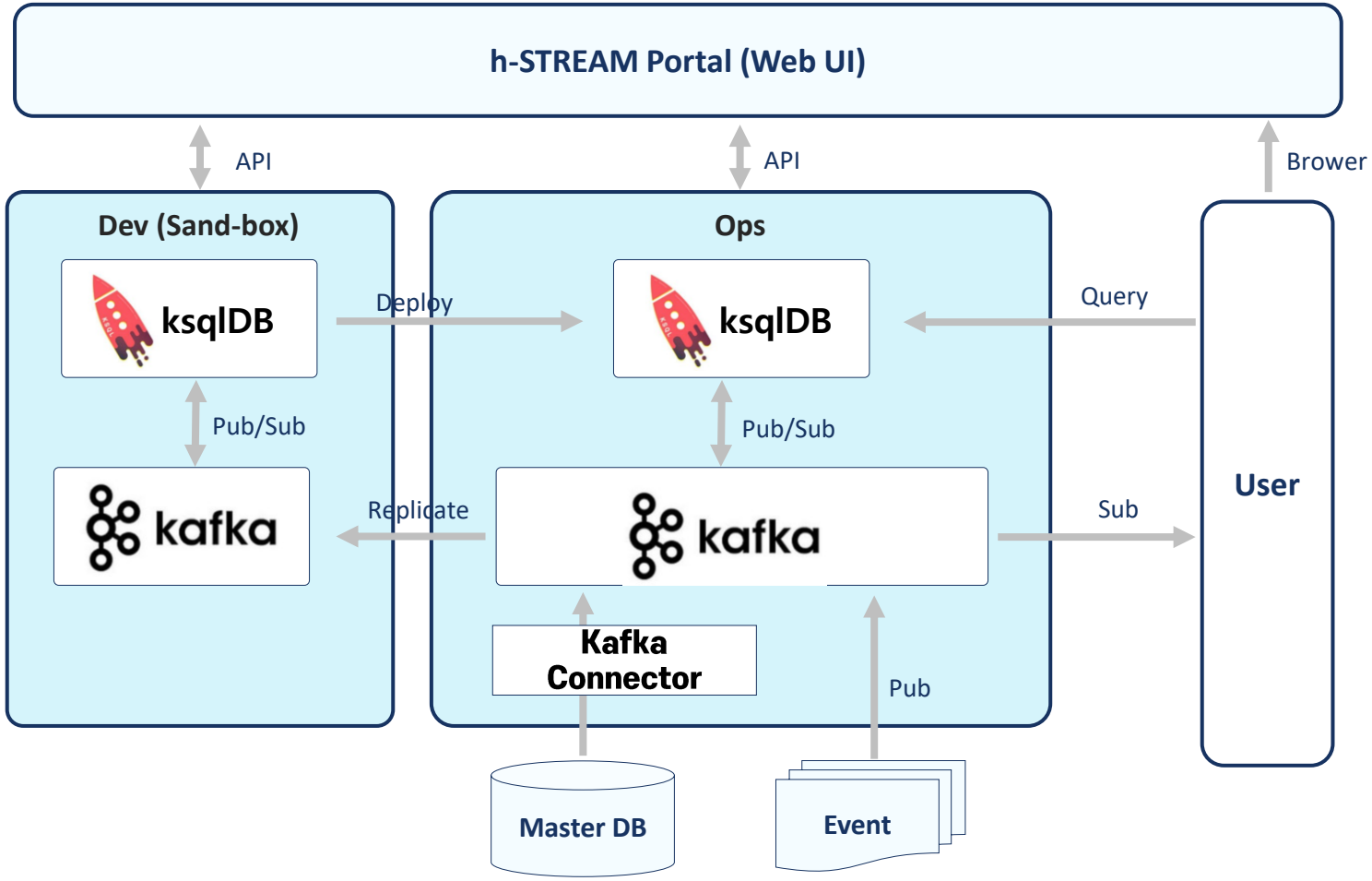
컴퓨팅 자원이 따로 없어 Stream Processing을 못하고 있어요 ㅠㅠ

안정적인 Stream Processing 로직을 운영 시스템에 연계하고 싶어요

Stream Processing **노하우**를 **공유** 받고 싶어요

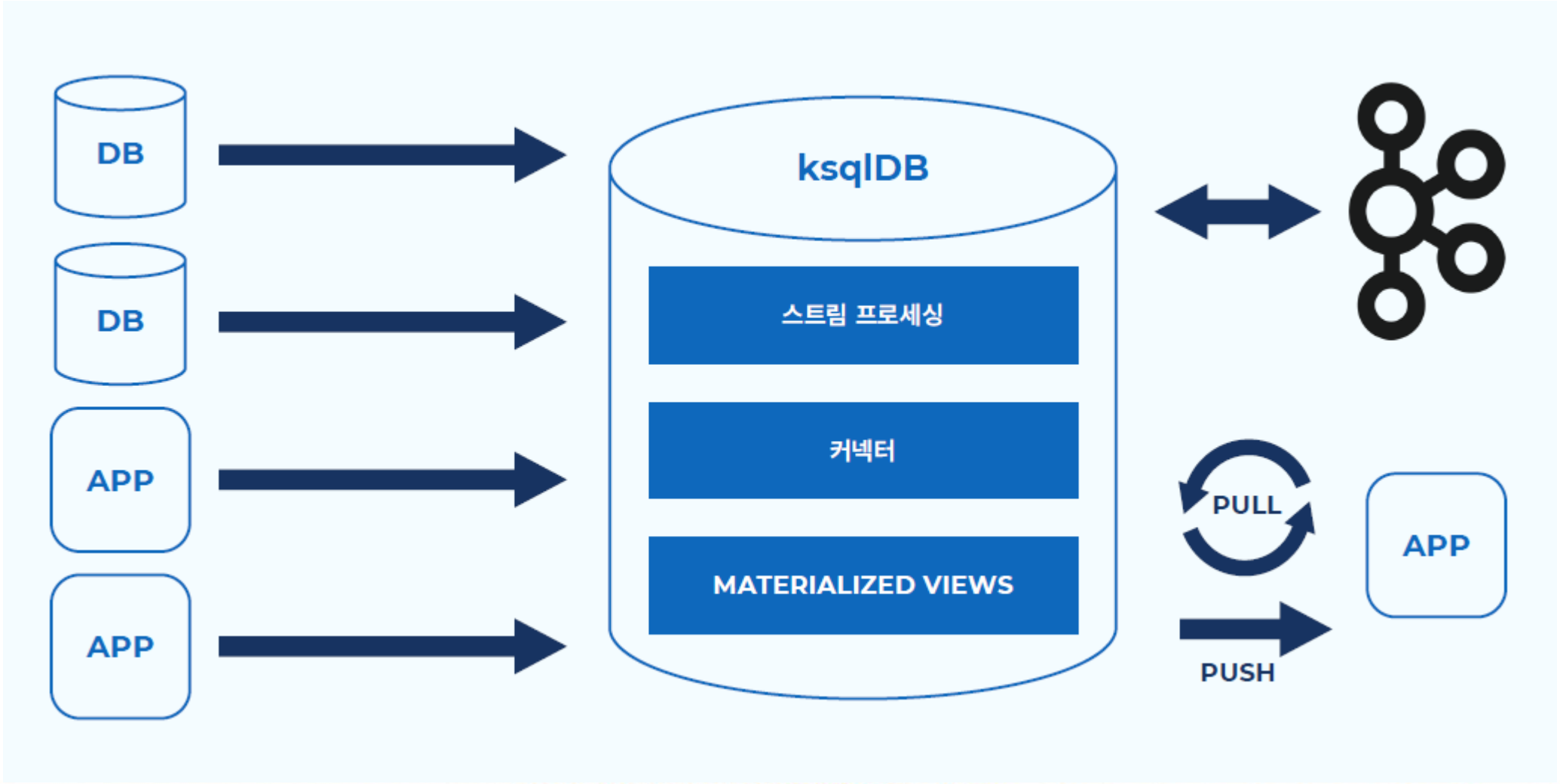
h-STREAM

Architecture



h-STREAM

ksqlDB



h-STREAM

Stream Processing Tools

Kafka Producer & Consumer

```
ConsumerRecords<String, String> records =
consumer.poll(100);
Map<String, Integer> counts = new DefaultMap<String,
Integer>();
for (ConsumerRecord<String, Integer> record : records) {
  String key = record.key();
  int c = counts.get(key)
  c += record.value()
  counts.put(key, c)
}
for (Map.Entry<String, Integer> entry : counts.entrySet()) {
  int stateCount;
  int attempts;
  while (attempts++ < MAX_RETRIES) {
   try {
     stateCount = stateStore.getValue(entry.getKey())
     stateStore.setValue(entry.getKey(), entry.getValue() +
stateCount)
     break;
   } catch (StateStoreException e) {
     RetryUtils.backoff(attempts);
   }
 }
}
```

Kafka Streams

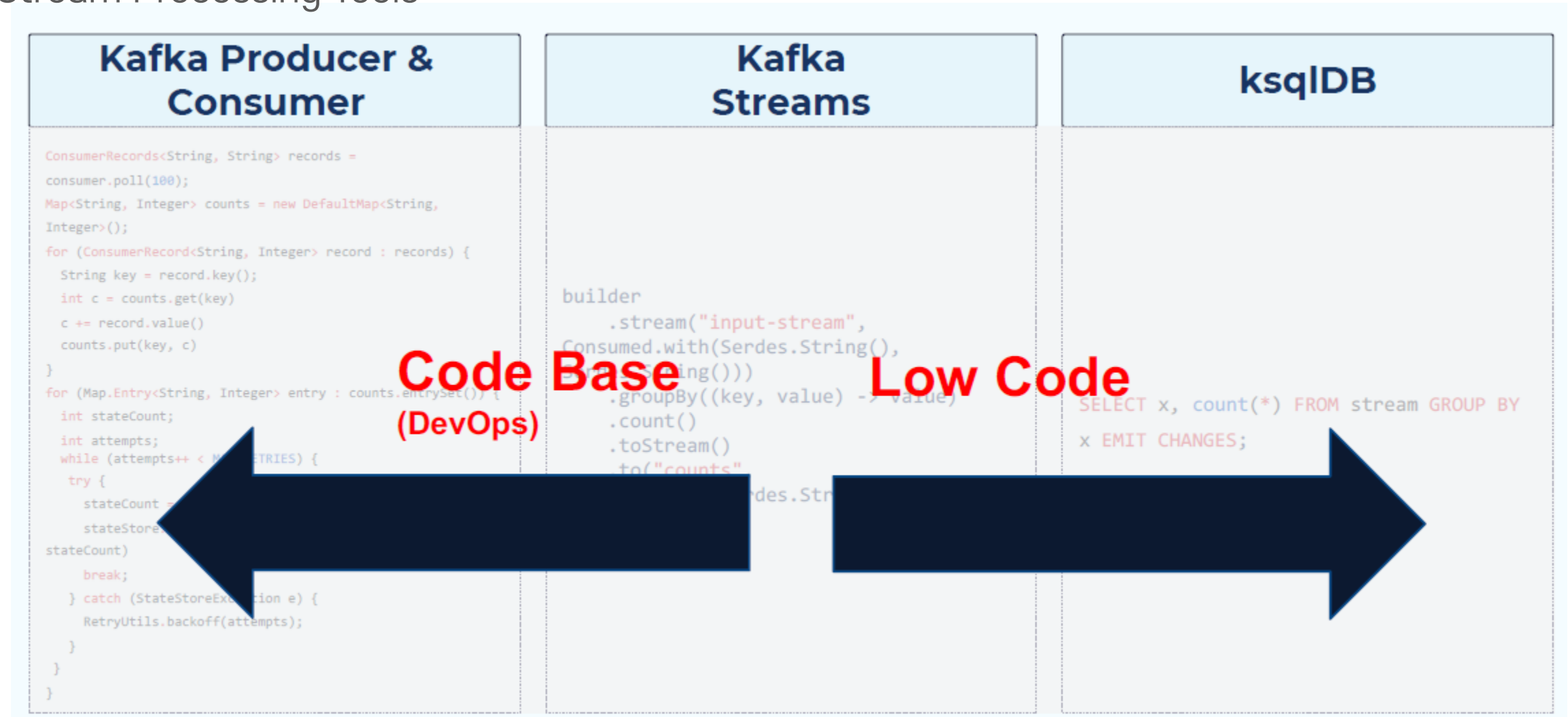
```
builder
    .stream("input-stream",
Consumed.with(Serdes.String(),
Serdes.String()))
    .groupBy((key, value) -> value)
    .count()
    .toStream()
    .to("counts",
Produced.with(Serdes.String(),
Serdes.Long()));
```

ksqlDB

```
SELECT x, count(*) FROM stream GROUP BY
x EMIT CHANGES;
```

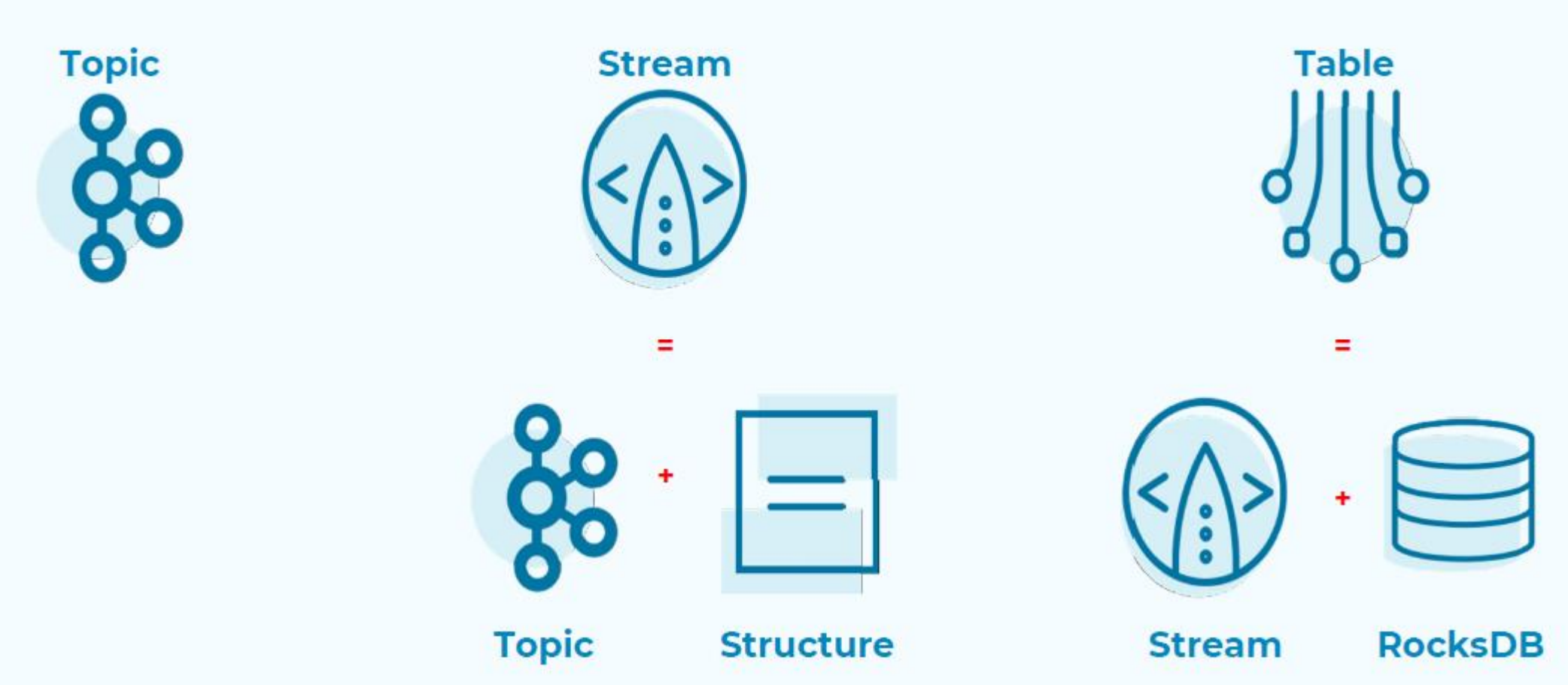
h-STREAM

Stream Processing Tools



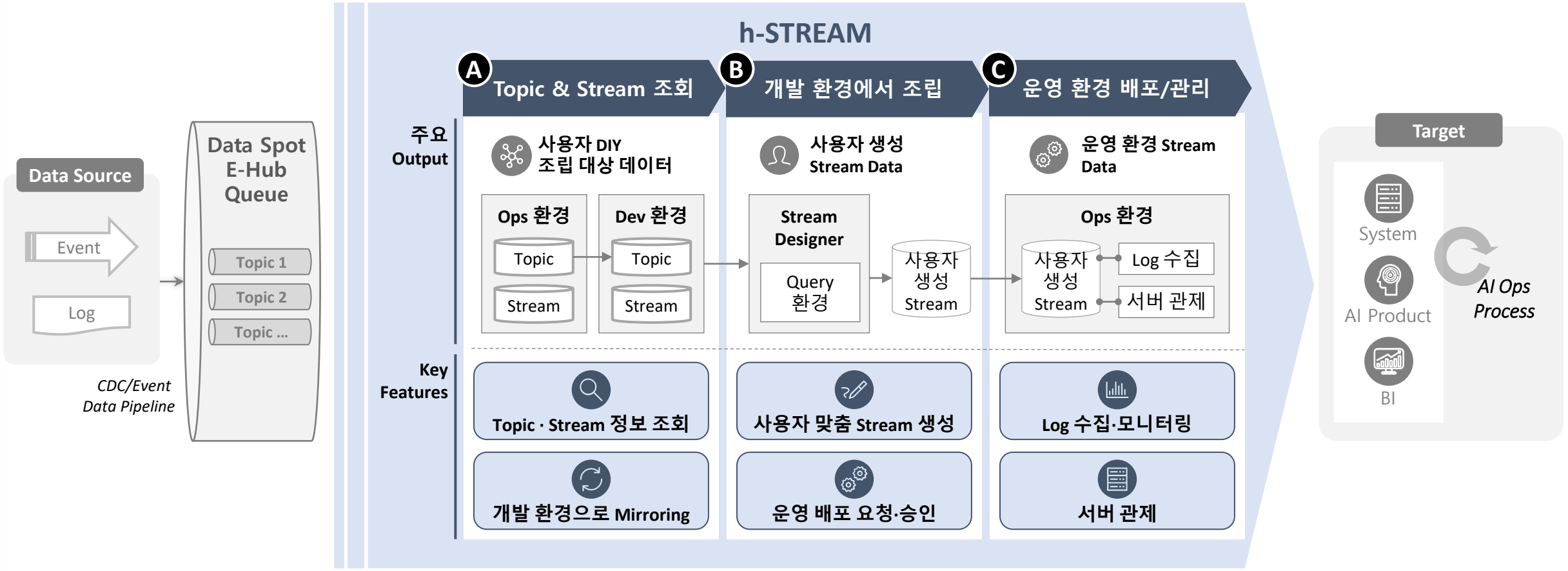
h-STREAM

Topic / Stream / Table



h-STREAM

h-STREAM Features



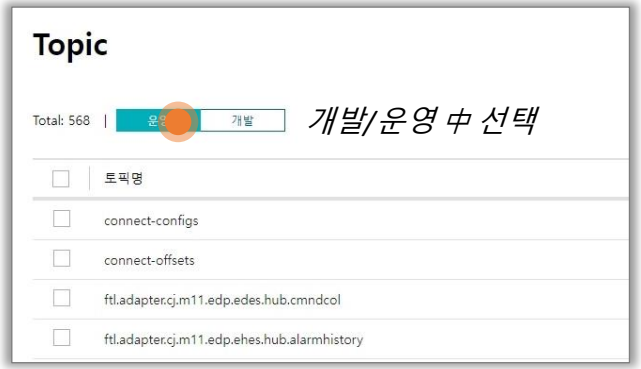
h-STREAM

Topic & Stream Search

Topic · Stream 정보 조회

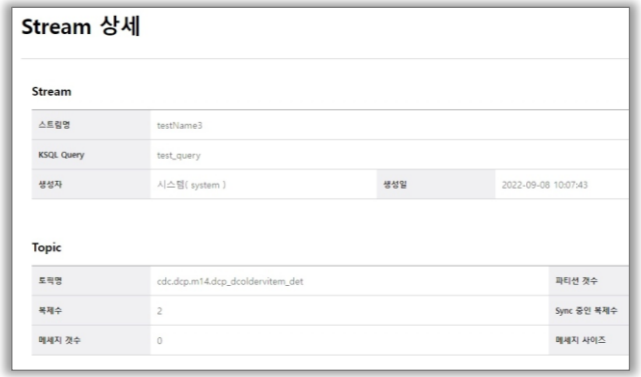
Topic · Stream 목록 조회

- 개발(Sandbox) 및 운영 환경에서 생성/운영 중인 Topic 및 Stream 목록 조회



Topic · Stream 상세 정보 조회

- Topic/Stream 명, 배포 상태, 생성자, 생성일, 최종 수정일 등 해당 Topic/Stream에 대한 상세 정보 조회



개발 환경으로 Mirroring

Topic 복사
(운영 환경 → Sandbox)

- 운영 Topic 조회 후, 사용자가 직접 DIY 조립을 수행할 항목들을 개발(Sandbox) 환경으로 복사 요청
- 관리자는 해당 요청 사항을 확인한 후, 이를 승인하여 개발 환경으로 복제

h-STREAM

Stream Development

사용자 맞춤 Stream 생성

KSQL 기반 Query 작성
(Stream Designer)

- Sandbox 환경에서 KSQL 기반 Query 작성하여 신규 Stream Data 생성
- Query 작성 참고를 위한 Reference Table 및 타 Stream Data 조회 가능

운영 배포 요청

Stream 배포 요청
(Sandbox → 운영 환경)

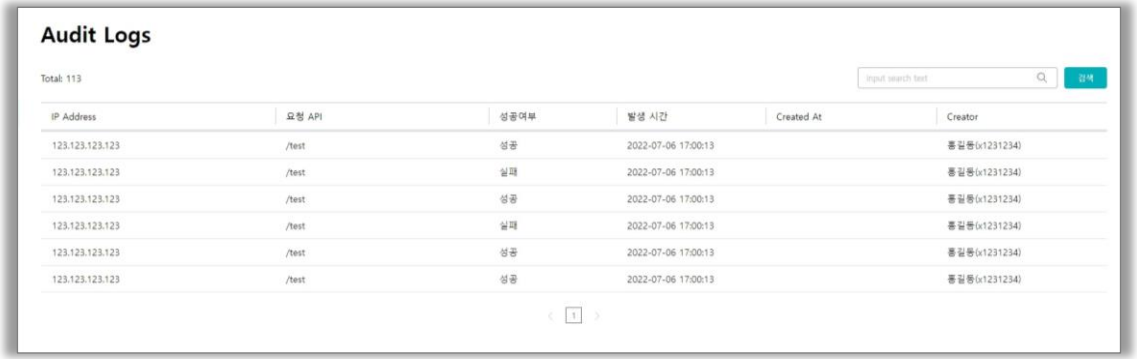
- 개발 환경에서 최종 생성된 Stream Data를 운영 환경에 배포하기 위해서, 관리자에게 '스트림 배포' 요청
- 관리자는 요청 건 확인 후 승인하여 운영 환경 배포 수행

h-STREAM

Stream Deploy

Log 수집·모니터링

Log 확인



- h-STREAM에서 발생한 Log 수집
  - IP Address, 요청 API, 성공 여부, 발생 시간, 생성 시간, 생성자
- Search 기능 활용하여 키워드 기반 Log 검색/조회 가능

서버 관제

서버 목록 조회

- h-STREAM 서버 및 상태 정보 조회

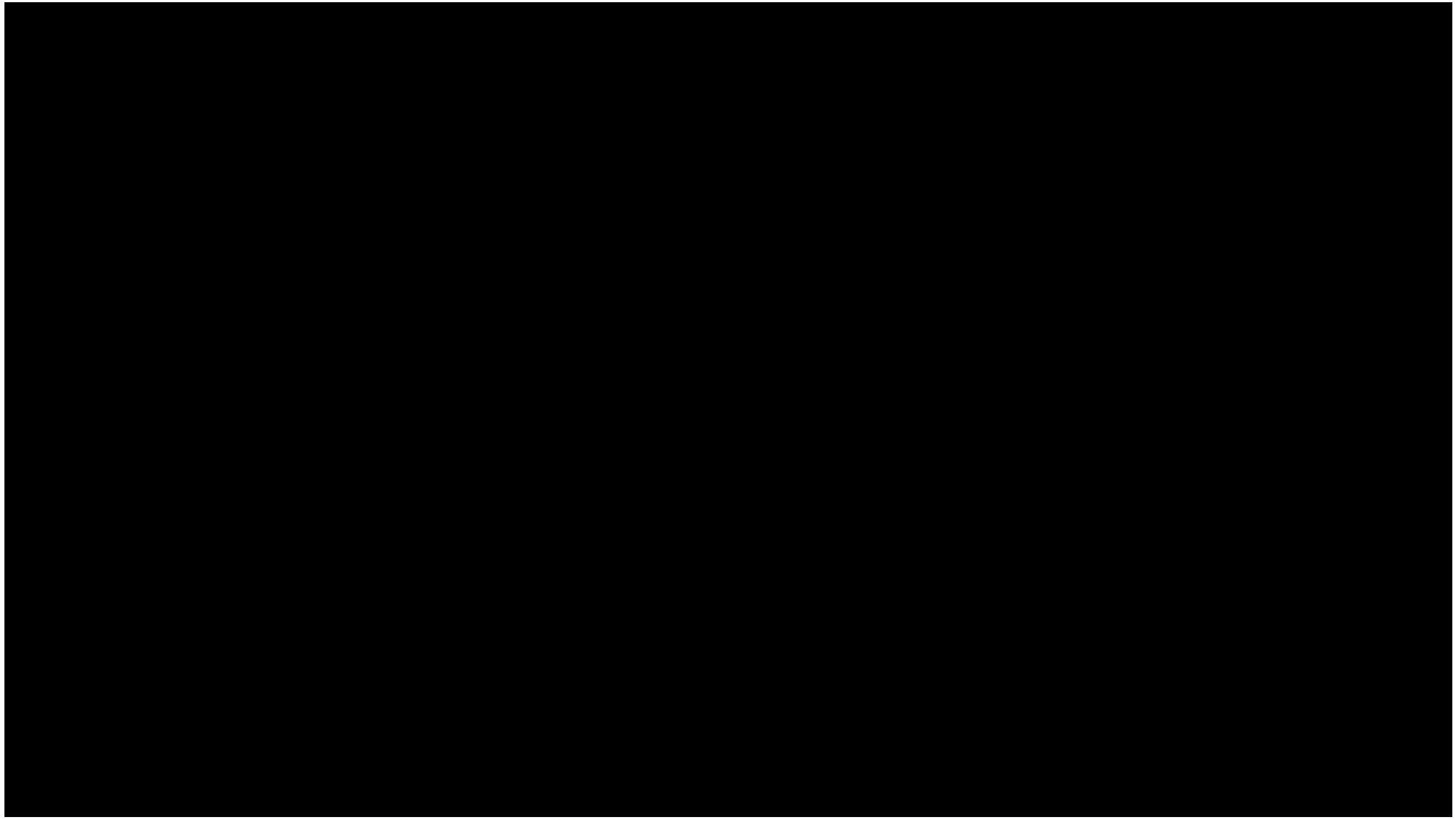
호스트명	IP Address	상태
hostName1	123.123.123.121	성공
hostName2	123.123.123.122	성공
hostName3	123.123.123.123	성공
hostName4	123.123.123.124	실패

운영 관리

- 개별 서버 목록 및 상태 확인 후 각 서버에 '중지', '재가동', 'LOG보기' 명령 수행

h-STREAM

Demo



h-STREAM

Expected Effect

실시간 데이터의 Self-Service 분석 구현

Self-Service 선별 및 조립 ksql 쿼리 기반 손쉬운 topic 조합 및 재구성 지원

Code 작성 최소화 별도의 API 개발 필요 없이, Web UI 기반으로 topic 접근

올바른 의사결정을 위한 데이터 투명성 증대

Data Governance 사용자 배포 Stream이 전사 Catalog와 연동

Data Timeliness 의사결정 시점에 상응하며 시차로 인한 오류 제거

실시간 데이터 기반 비즈니스 Use Case 확산

실시간 데이터 가치 입증 실시간 데이터 적용 사례 발굴 기반 Biz. Impact 실현

적시적 의사결정 지원 핵심 의사결정을 위한 분석 모델에 적시적 Input 제공

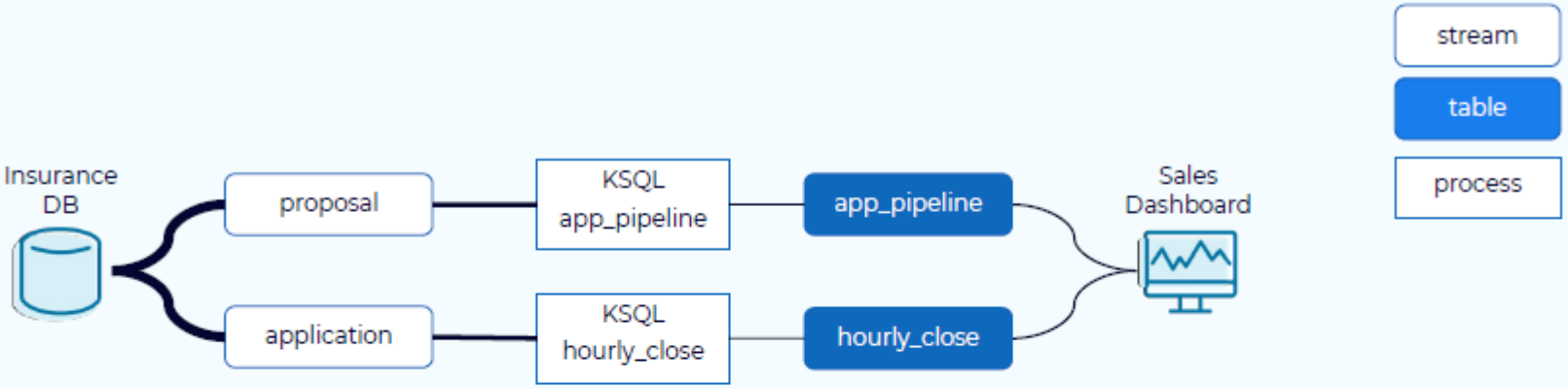
전사 데이터 자산 활용 활성화

CDC 데이터 활용 확산 활용률이 상대적으로 낮은 CDC 데이터의 활용도 강화

분석 문화 고도화에 기여 In-stream Analytics 문화 확산을 위한 기반 제공

h-STREAM

Use Case > Updating Dashboards

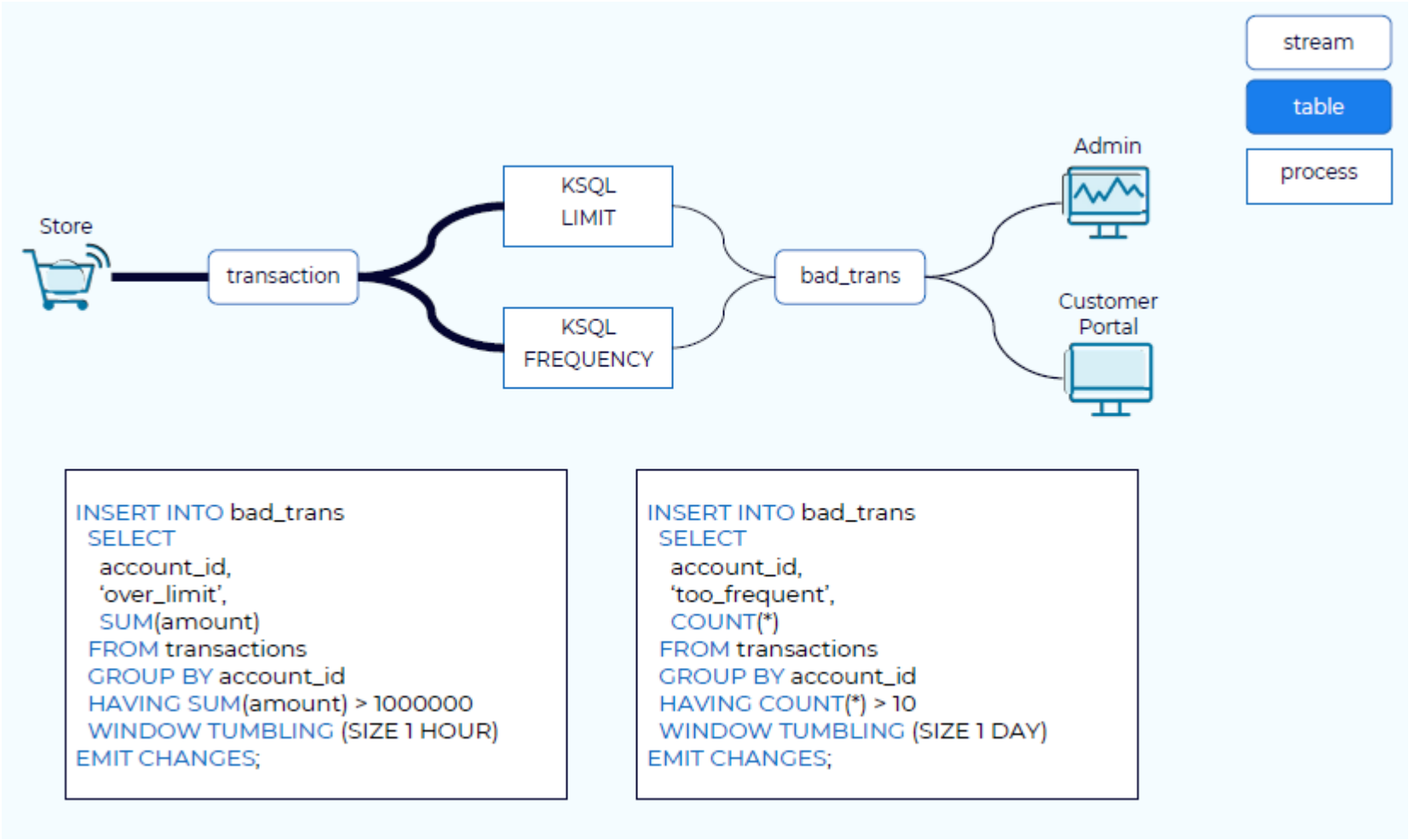



```
INSERT INTO app_pipeline
  SELECT
    COUNT(*)
  FROM proposal prop
  INNER JOIN application app
    ON prop.plan_id = app.plan_id
  WHERE app.status IS NOT 'pending'
  GROUP BY prop.id
  WINDOW TUMBLING (SIZE 1 hour)
EMIT CHANGES;
```

```
INSERT INTO hourly_close
  SELECT
    SUM(amount)
    COUNT(*)
  FROM application
  GROUP BY id
  WHERE app.status IS 'signed'
  WINDOW TUMBLING (SIZE 1 hour)
EMIT CHANGES;
```

h-STREAM

Use Case > Fraud Detection

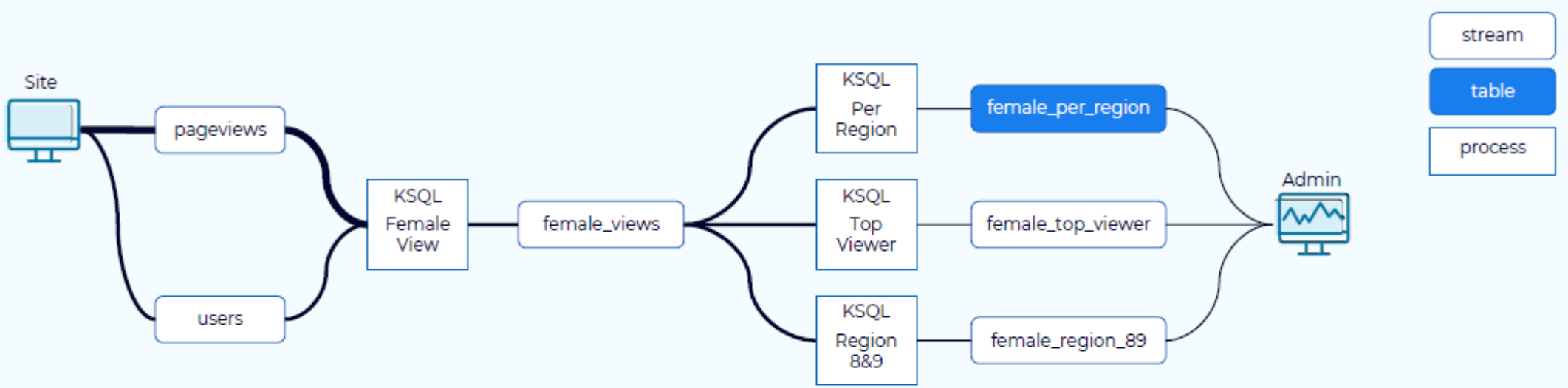



```
INSERT INTO bad_trans
  SELECT
    account_id,
    'over_limit',
    SUM(amount)
  FROM transactions
  GROUP BY account_id
  HAVING SUM(amount) > 1000000
  WINDOW TUMBLING (SIZE 1 HOUR)
EMIT CHANGES;
```

```
INSERT INTO bad_trans
  SELECT
    account_id,
    'too_frequent',
    COUNT(*)
  FROM transactions
  GROUP BY account_id
  HAVING COUNT(*) > 10
  WINDOW TUMBLING (SIZE 1 DAY)
EMIT CHANGES;
```

h-STREAM

Use Case > Site Access Analysis




```
CREATE STREAM female_views AS
  SELECT
    USERS.ID USERID,
    PAGEVIEWS.PAGEID PAGEID,
    USERS.REGIONID REGIONID,
    USERS.GENDER GENDER
  FROM pageviews
  LEFT OUTER JOIN USERS users ON
    ((pageviews.USERID = users.ID))
  WHERE (users.GENDER = 'FEMALE')
EMIT CHANGES;
```

```
CREATE TABLE female_per_region AS
  SELECT
    views.GENDER GENDER,
    views.REGIONID REGIONID,
    COUNT(*) NUMUSERS
  FROM FEMALE_VIEWS views
  WINDOW TUMBLING ( SIZE 30 SECONDS
)
  GROUP BY views.GENDER,
views.REGIONID
  HAVING (COUNT(*) > 1)
;
```

```
CREATE STREAM female_top_viewer AS
  SELECT *
  FROM female_views views
  WHERE (CAST(SPLIT(views.PAGEID,
    '_')[2] AS INTEGER) >= 50)
EMIT CHANGES;
```

```
CREATE STREAM female_region_89 AS
  SELECT *
  FROM female_views views
  WHERE ((views.REGIONID LIKE '%_8') OR
    (views.REGIONID LIKE '%_9'))
EMIT CHANGES;
```

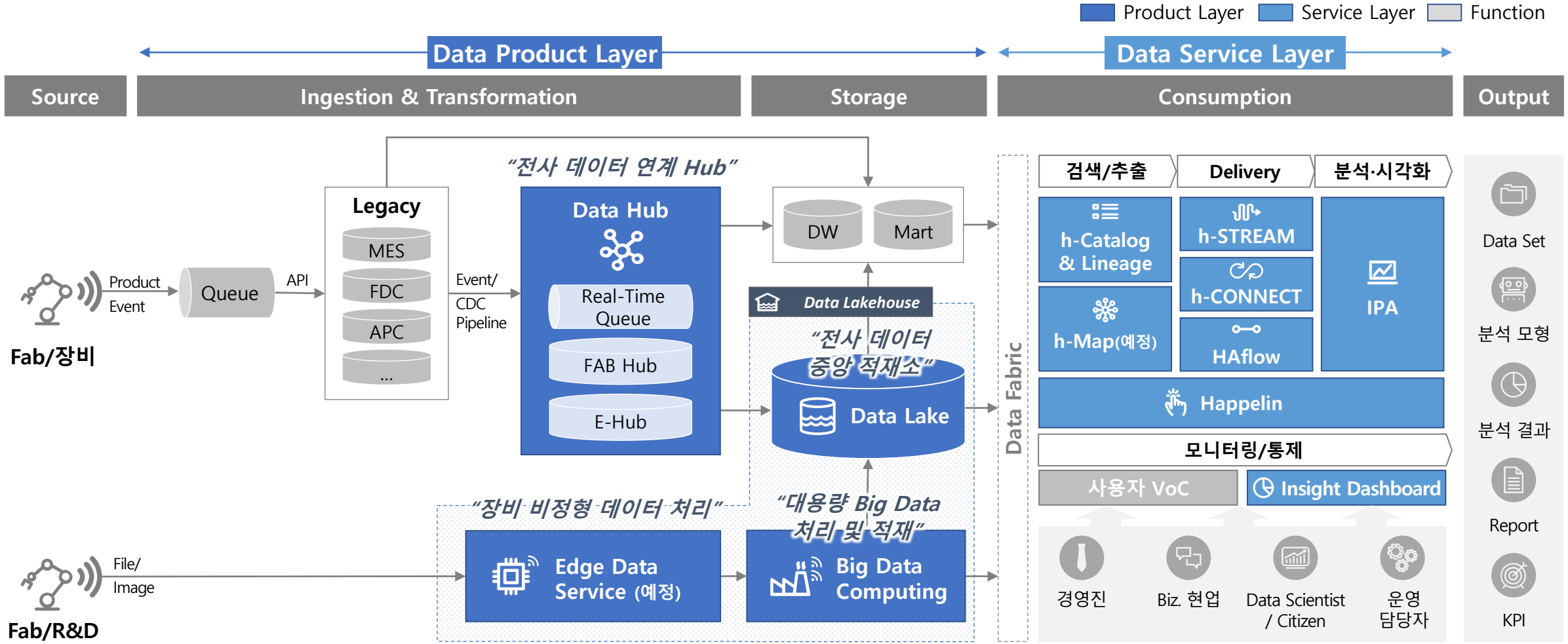
Q&A

E-mail jseung.park@sk.com

Thank you

Data Platform

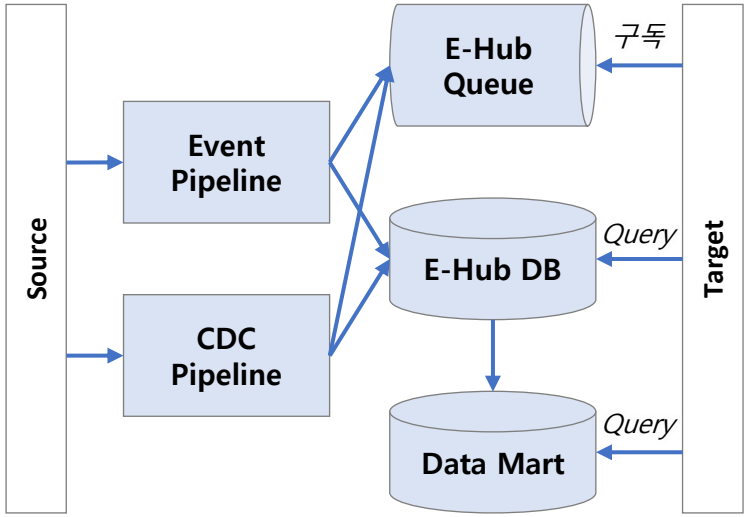
Data Platform Eco System



Data Hub

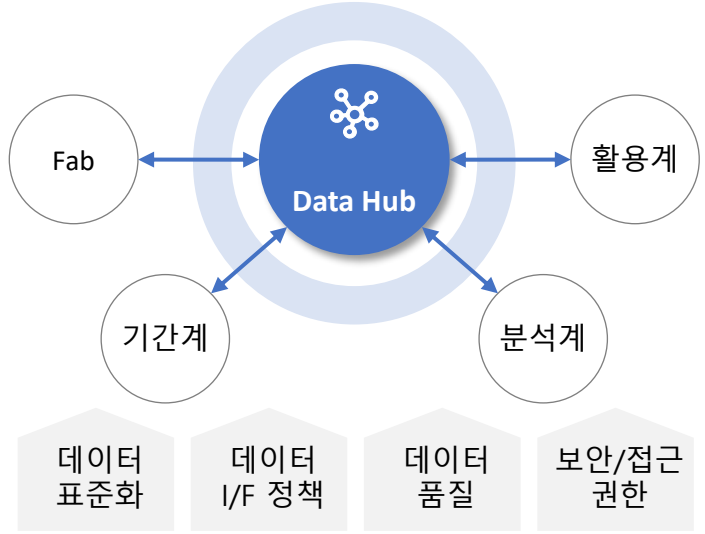
Real-time Data Platform

① 실시간 데이터 연계



- **End-to-End Near Real-Time 데이터 연계** (데이터 발생 시점부터 10초 이내)
- E-Hub Queue 적재 데이터 **실시간 구독**
- 실시간성 **Mart向 Ad-hoc Query** 실행 가능

② 전사 Interface Hub 역할 수행



- 전사 데이터 연계 **Single Access Point**
- I/F 일원화를 통한 **기간계 운영 부담 경감**
- 통일된 I/F 운영 기준 기반 **데이터 품질 및 일관성 확보**

③ 다양한 Delivery Pipeline

h-STREAM (Streaming Data Service)
- Sandbox 환경에서 사용자가 Streaming 데이터를 DIY 조립/생성
- 신규 생성된 데이터를 사용자의 Notebook 환경에 전달

h-CONNECT
- 실시간 변경 데이터를 사용자가 지정한 Table과 지속적으로 자동 동기화

Pub/Sub
- E-Hub Queue 적재 데이터를 Target 시스템에서 구독

- 사용자가 **실시간 데이터를 목적에 맞게 즉시 활용**할 수 있도록 Delivery Pipeline 제공
- VM/Notebook으로 안정적인 적재 지원