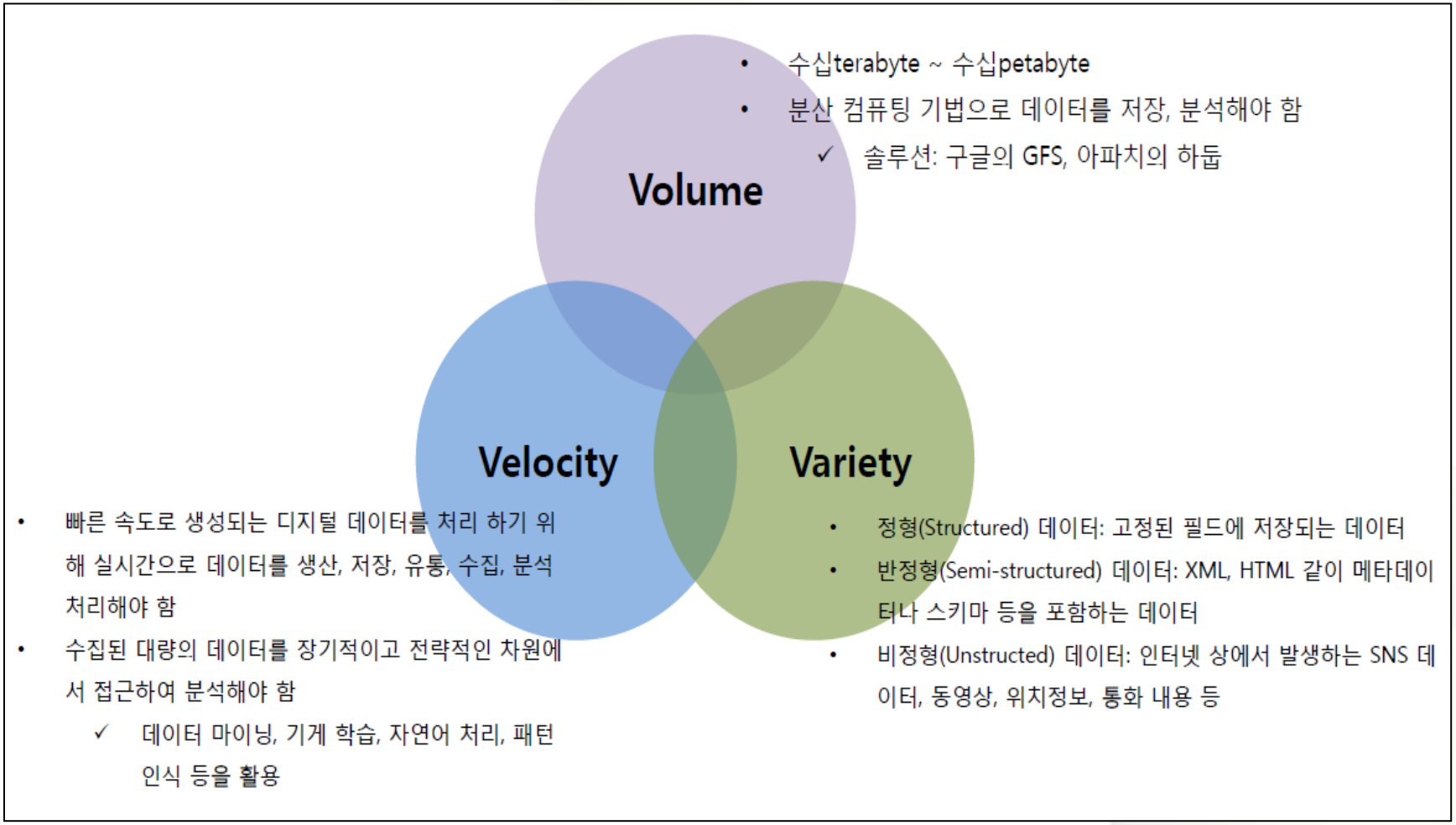
Introduction to Spark

빅데이터

- 빅데이터의 3대 요소
  - 크기 (Volume), 속도 (Velocity), 다양성 (Variety)

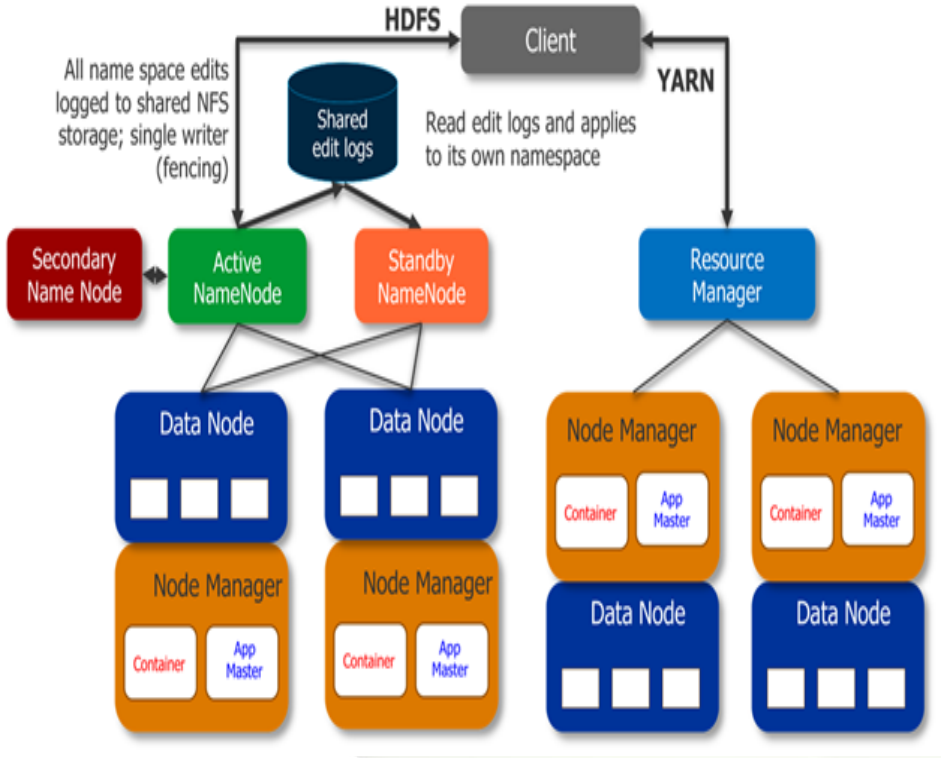


하둡 (Hadoop)

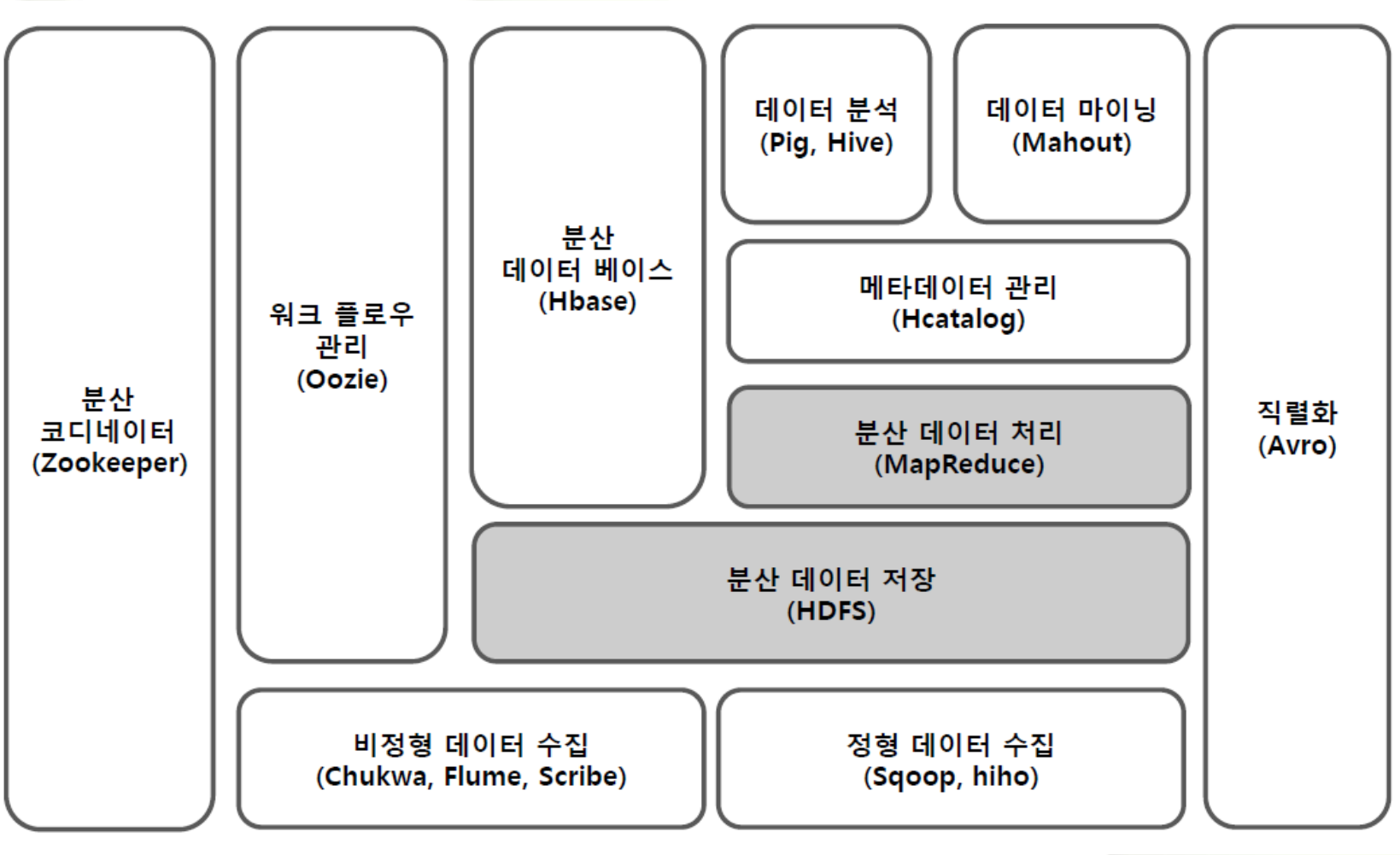
- 빅데이터를 분산 처리할 수 있는 자바 기반의 오픈소스 프레임워크

- 분산 파일 시스템인 HDFS(Hadoop Distributed Files System)에 데이터를 저장하고 분산 처리 시스템인 맵리듀스를 이용해 데이터를 처리

- 2005년에 더그 커팅(Doug Cutting)이 구글이 논문으로 발표한 GFS(Google File System)와 MapReduce를 구현한 결과물



하둡 에코시스템


분산
코디네이터
(Zookeeper)
워크 플로우
관리
(Oozie)
분산
데이터 베이스
(Hbase)
데이터 분석
(Pig, Hive)
데이터 마이닝
(Mahout)
메타데이터 관리
(Hcatalog)
분산 데이터 처리
(MapReduce)
분산 데이터 저장
(HDFS)
직렬화
(Avro)
비정형 데이터 수집
(Chukwa, Flume, Scribe)
정형 데이터 수집
(Sqoop, hiho)

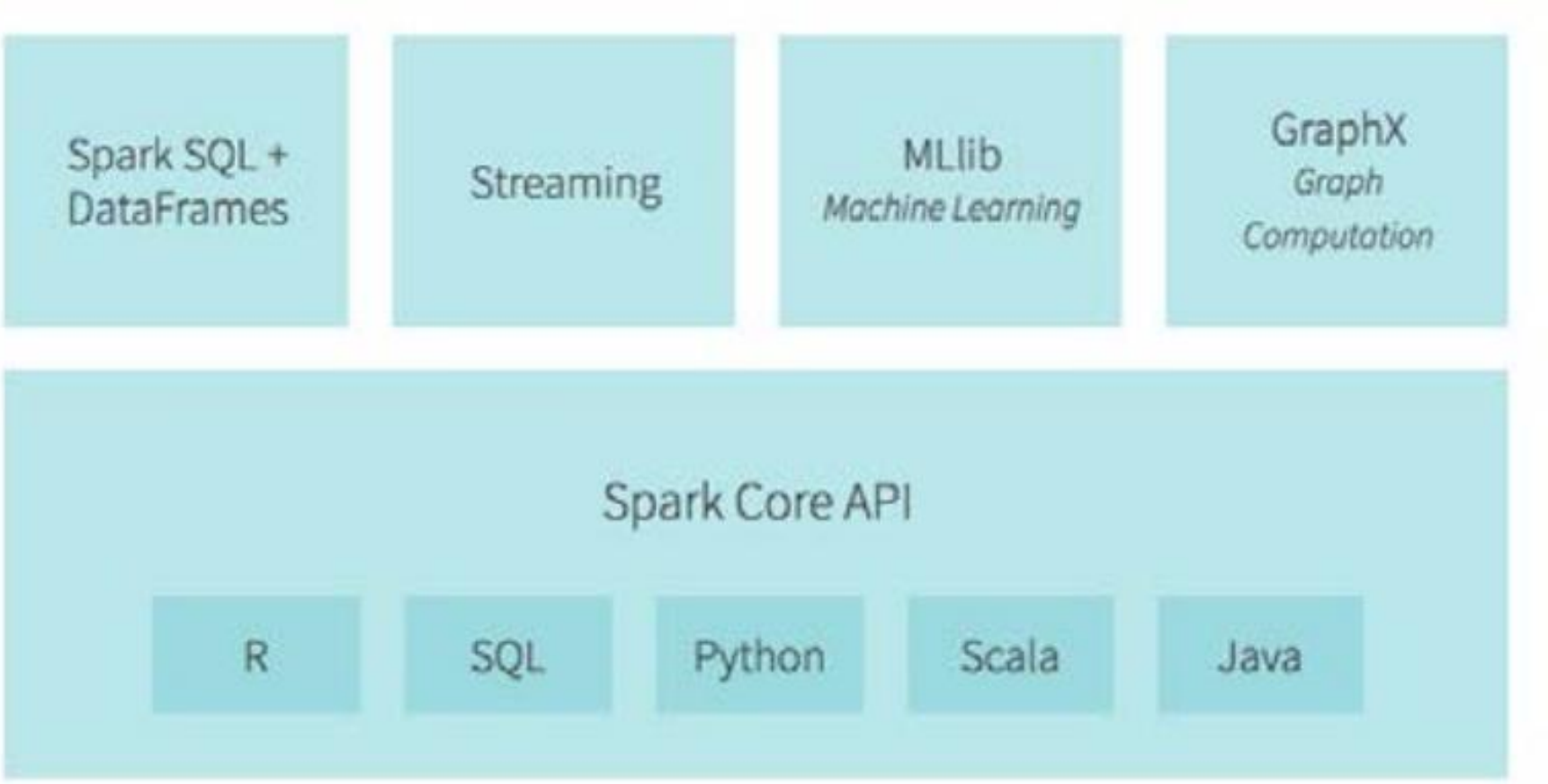
하둡의 단점

- 대부분의 연산 작업을 파일시스템 기반으로 처리 → 상대적으로 낮은 성능
- 복잡한 데이터 분석 요구사항을 맵과 리듀스 패턴만으로 해결하기 어려움
- 자바 언어 기반으로 파이썬, R 등 다른 분석용 도구와 연동이 어려움
- SQL on Hadoop 계열의 도구와 같이 맵리듀스를 편리하게 구현할 수 있는 도구들이 있지만 데이터 분석 요구사항을 충분히 반영하는데 한계가 있음

스파크 (Spark)

- 하둡 기반 맵리듀스의 단점을 보완하기 위해 개발된 분산 데이터 처리 환경
- 메모리를 이용한 데이터 저장 방식을 제공함으로써 머신러닝 등 반복적인 데이터 처리가 필요한 분야에서 높은 성능 구현
- 최적화 과정을 통해 효율적인 데이터 처리 및 성능 향상 가능
- 자연스럽고 강력한 다수의 데이터 처리 함수 제공 → 프로그램의 복잡도를 현저하게 낮춤
- 자바, 스칼라, 파이썬, R을 사용해서 스파크 애플리케이션 개발 가능
- Spark SQL, MLlib 등 다양한 데이터 처리 분야에 특화된 라이브러리 제공

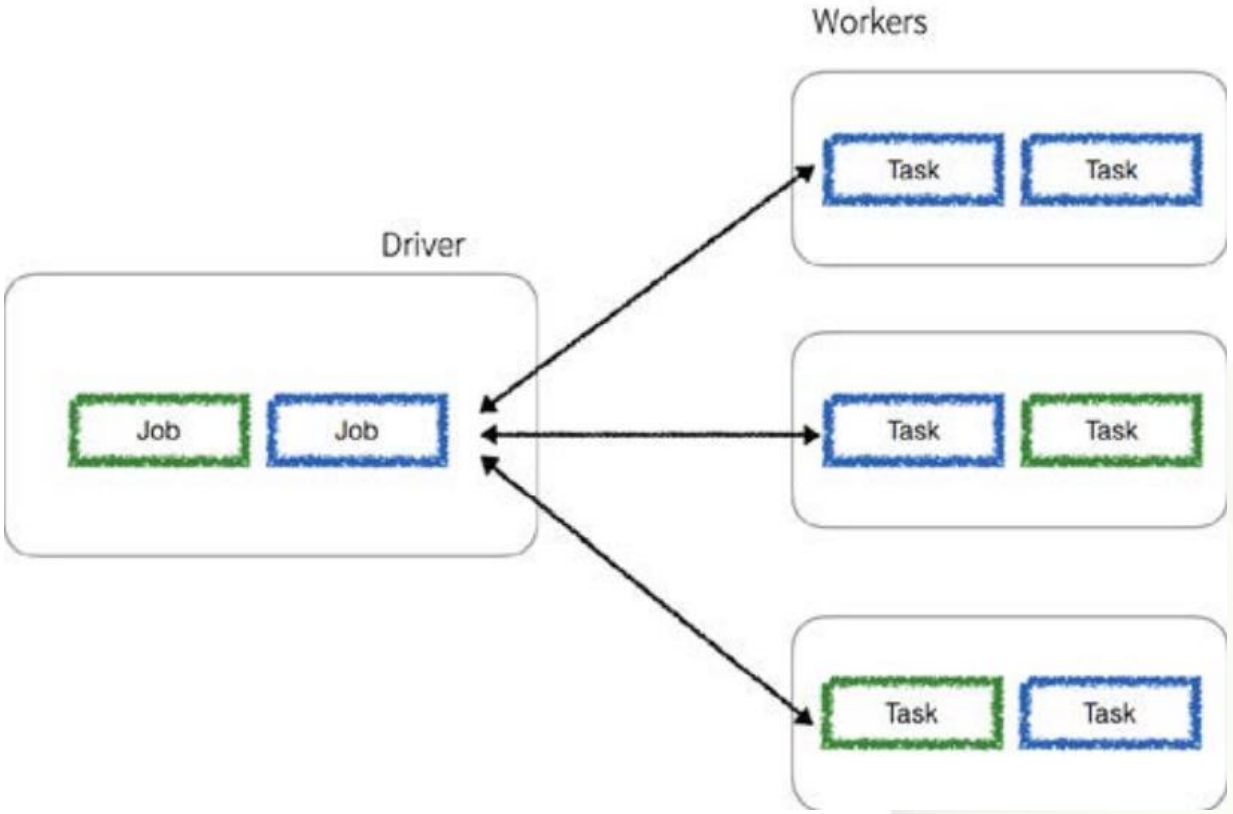
스파크 구성 요소


Spark SQL + DataFrames
Streaming
MLlib
Machine Learning
GraphX
Graph Computation
Spark Core API
R
SQL
Python
Scala
Java

실행 아키텍처

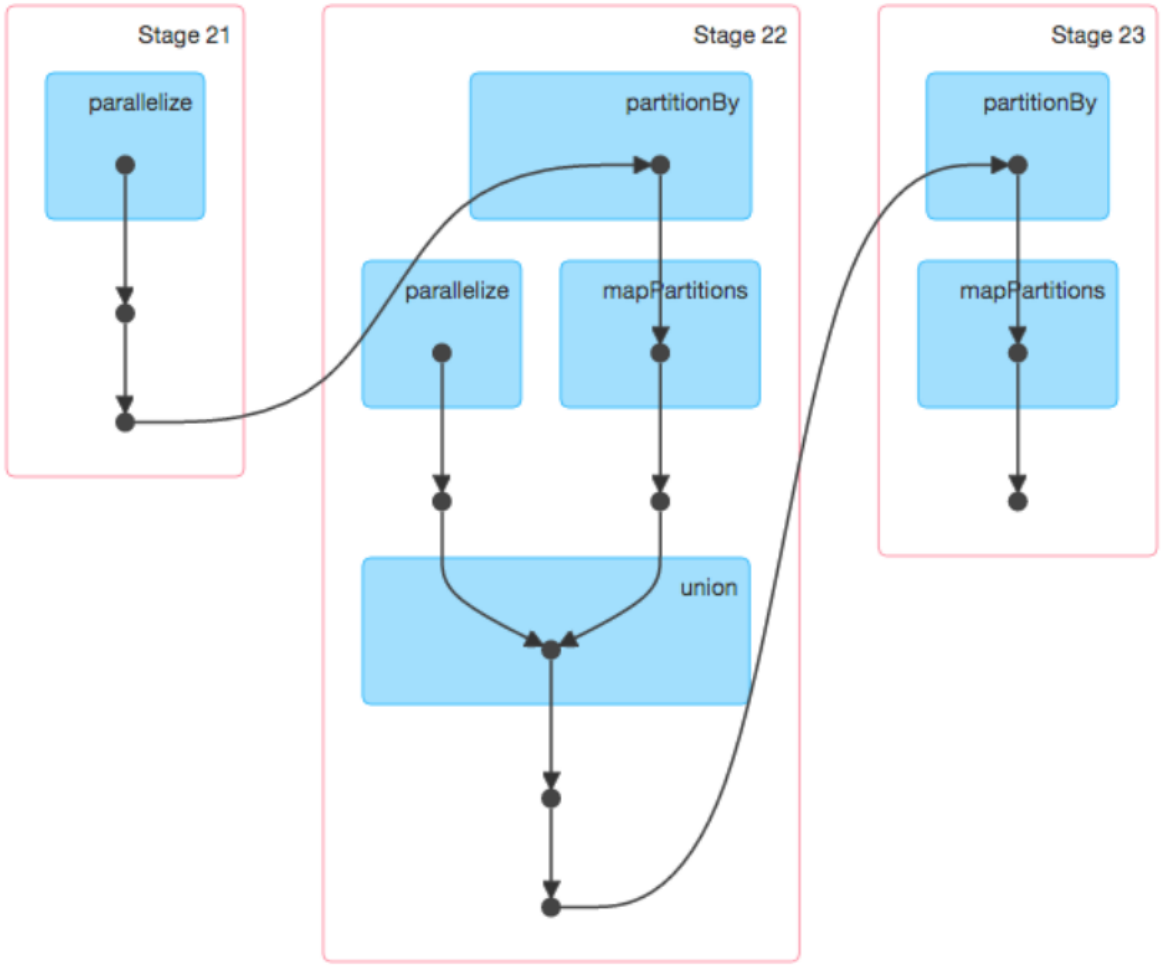
- 모든 스파크 애플리케이션은 여러 개의 잡을 관리하는 드라이버 프로세스를 마스터 노드에서 실행

- 드라이버 프로세스는 태스크의 수와 구성 결정 → 태스크 프로세스는 실행 노드에서 관리



비순환 방향성 그래프 (DAG)

- 스파크 잡에서 객체 의존성은 비순환 방향성 그래프 형태로 구성
- 이 그래프를 기반으로 스케줄링 및 실행 최적화 수행



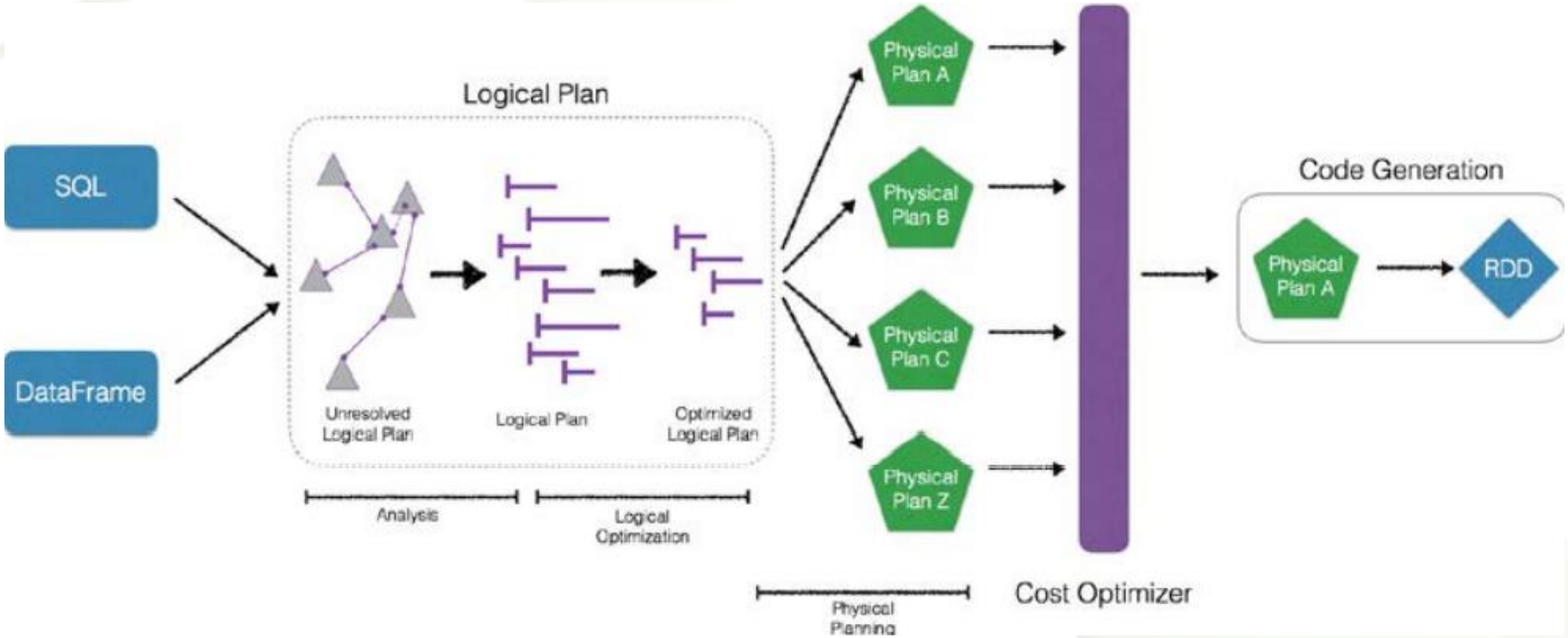
스파크 데이터 모델

- RDD (Resilient Distributed Dataset)
  - 변경 불가능(Immutable)한 자바 가상 머신 객체 집합
  - 스파크 내부에 존재하는 분산 데이터에 대한 모델
  - 다수의 서버에 분산 방식으로 저장된 데이터 요소들의 집합
  - 병렬 처리 및 장애 복구 가능

- DataFrame(after 1.3), DataSet(after 1.6)
  - Column들로 구성된 Schema를 사용하는 데이터 모델
  - 관계형 데이터베이스의 테이블과 유사한 방식의 데이터 처리 모델 제공
  - 향상된 최적화 도구 사용
  - 스파크 2.0부터 DataFrame 클래스는 DataSet 클래스로 통합
    - » R, 파이썬은 DataFrame 형식 사용
    - » 자바는 DataSet 형식 사용
    - » 스칼라는 DataFrame와 DataSet 모두 사용

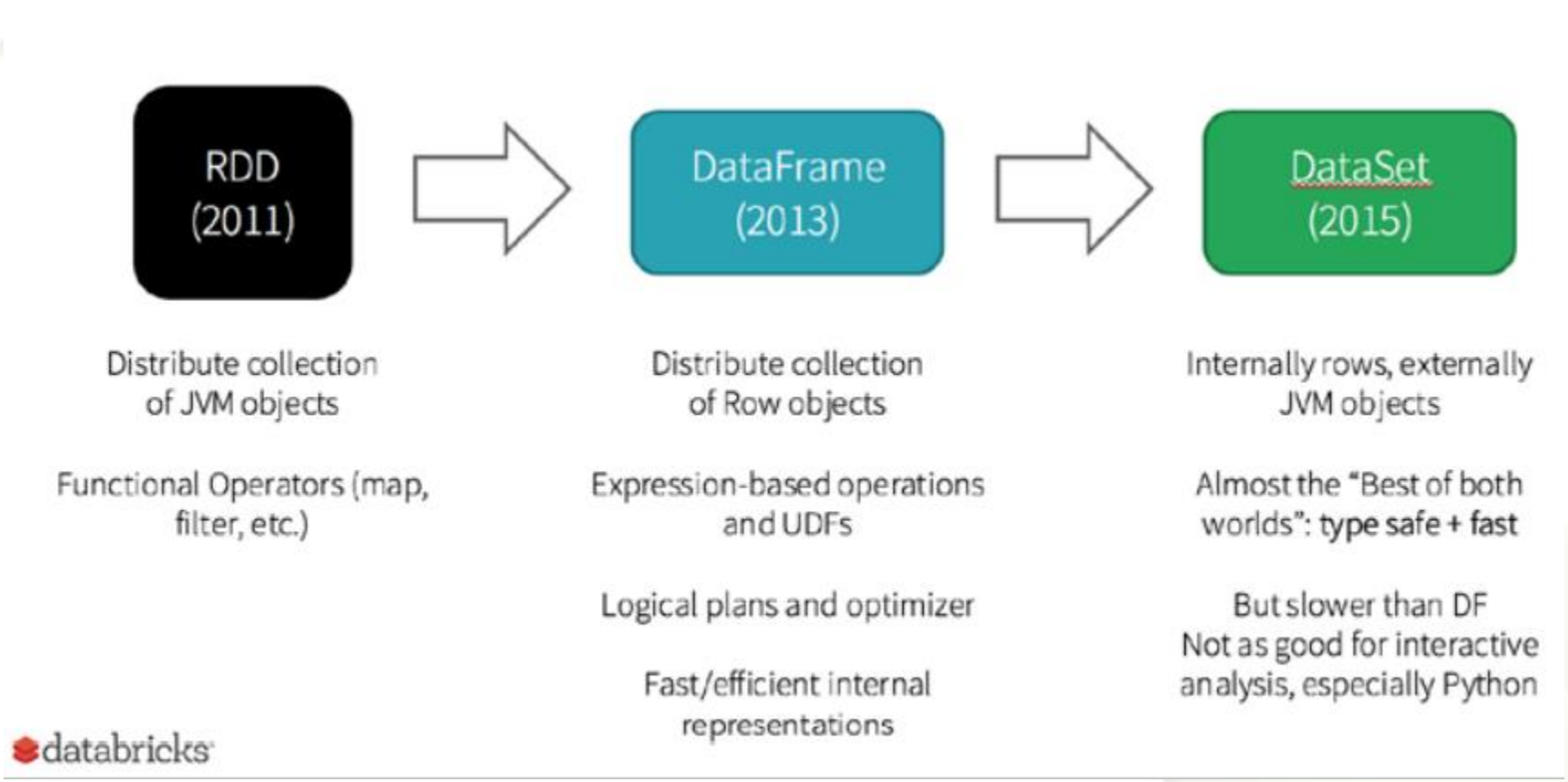
카탈리스트 옵티마이저

- Job의 최적화를 수행하는 엔진



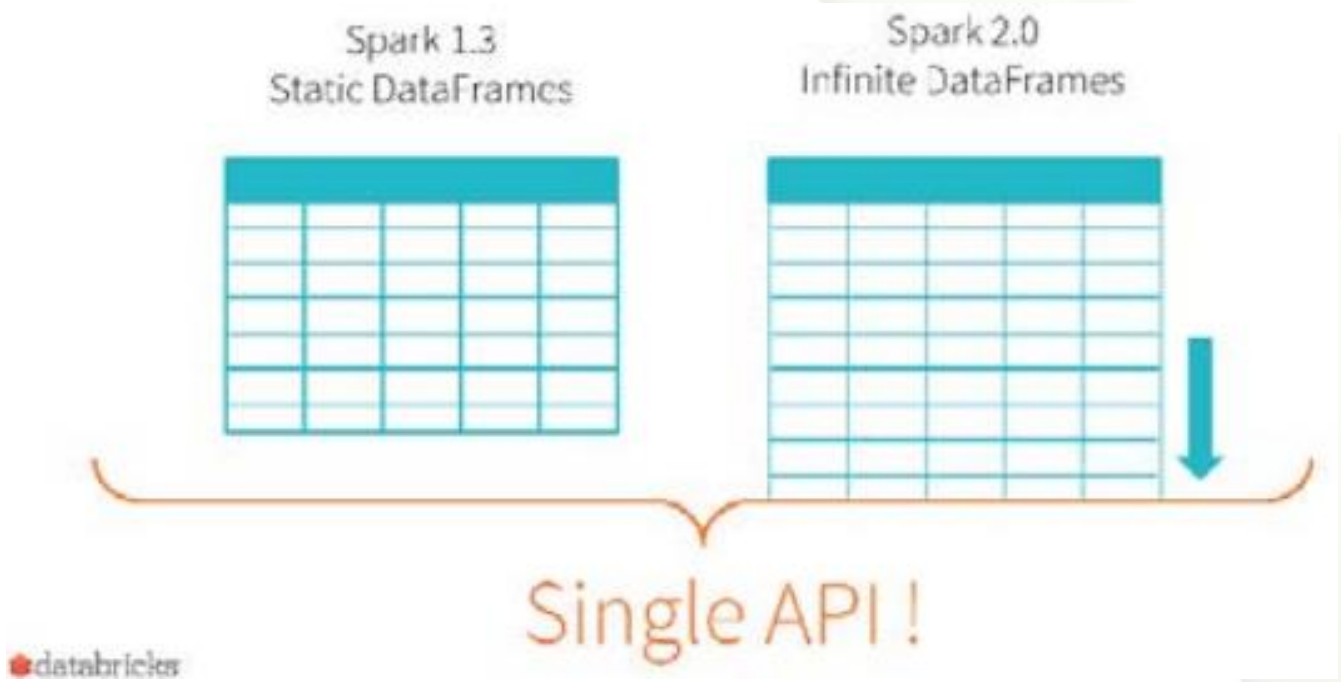
스파크 2.0

- 성능 강화, 구조적 스트리밍, 데이터셋과 데이터프레임 통합 등의 변화



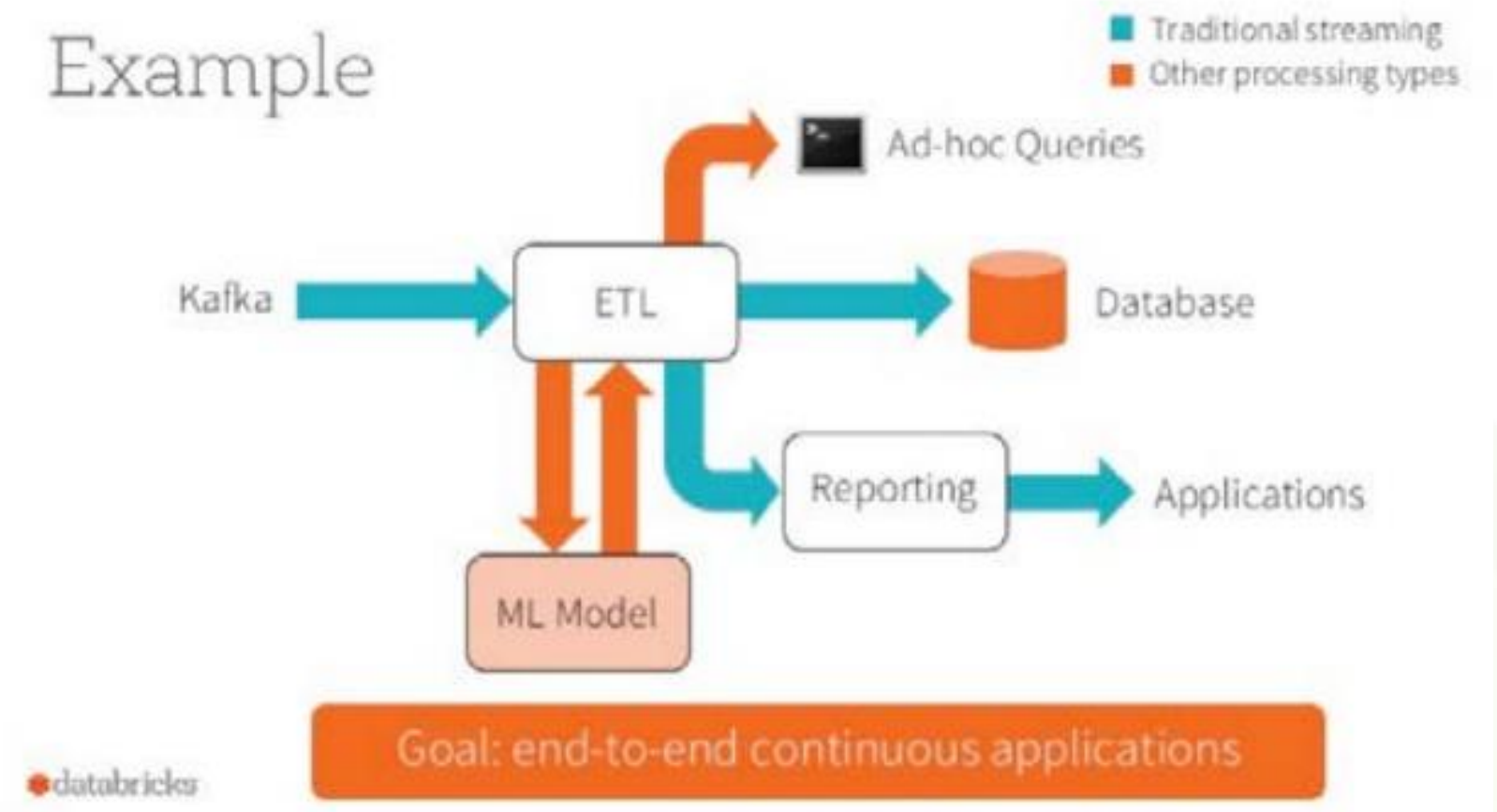
구조적 스트리밍

- 실시간 연속성 데이터에 대한 처리 지원
- 스트리밍 처리를 단순화하기 위해 배치와 스트리밍을 하나의 API로 통합



지속적 애플리케이션

- 스파크를 기반으로 엔드 투 엔드 애플리케이션 개발 가능



스파크 실행 환경 구축

spark-install, zeppelin-install 파일 참고