박경규

목 표

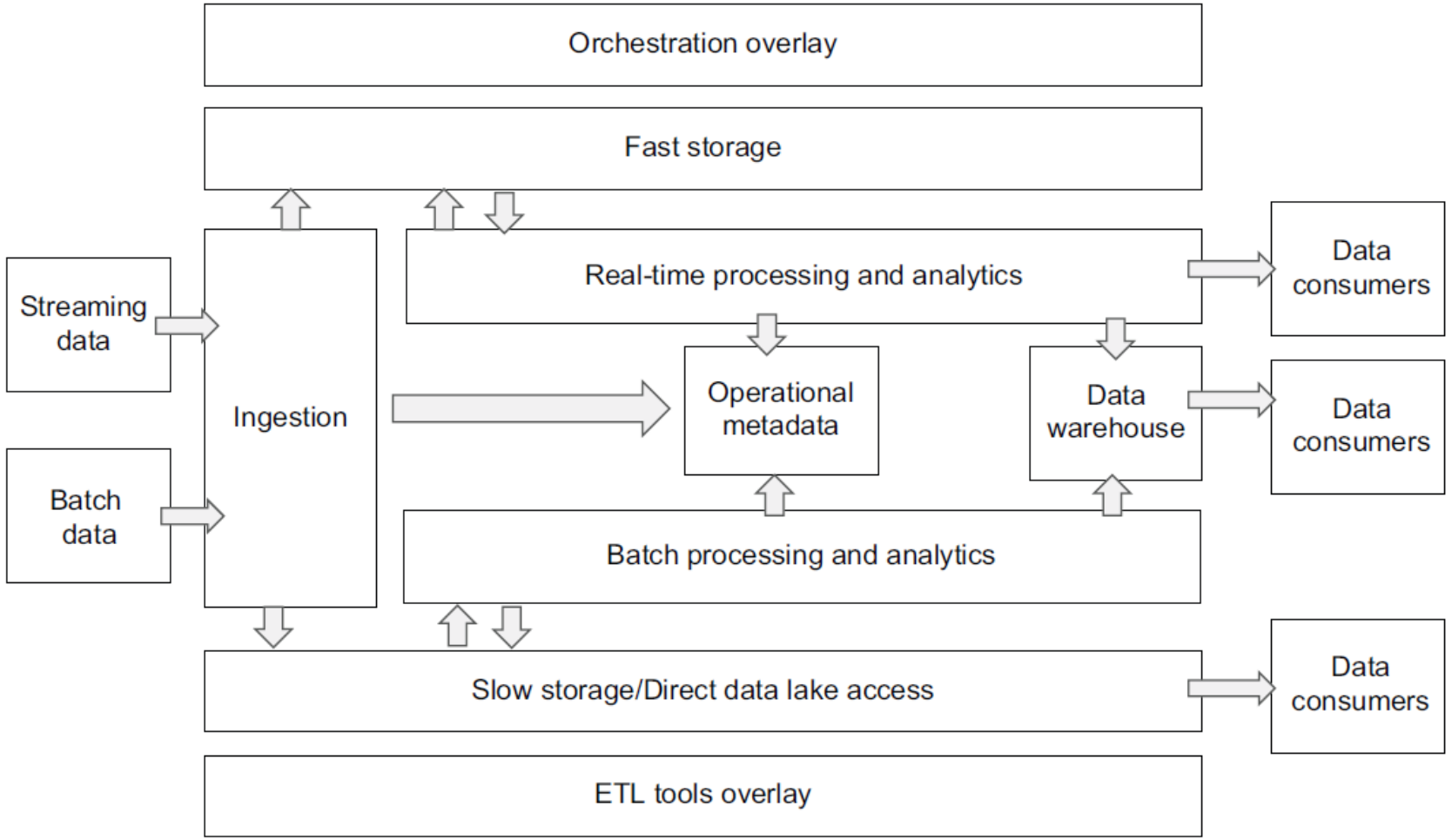
- Design your own data platform using a modular design
- Design for the long term to ensure it is manageable, versatile, and scalable
- Explain and justify your design decisions to others
- Pick the right cloud tools for each part of your design
- Avoid common pitfalls and mistakes
- Adapt your design to a changing cloud ecosystem

주요 내용

- Driving change in the world of analytics data
- Understanding the growth of data volume, variety, and velocity, and why the traditional data warehouse can't keep up
- Learning why data lakes alone aren't the answer
- Discussing the emergence of the cloud data platform
- Studying the core building blocks of the cloud data platform

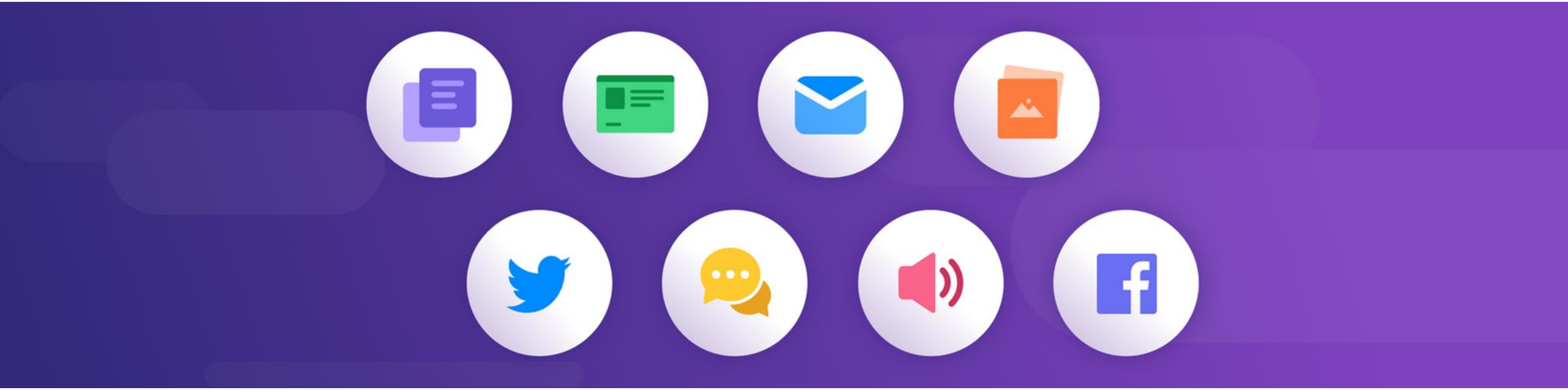
클라우드 데이터 플랫폼

분석 결과를 촉진하기 위해 (facilitate analytics outcomes) 모든 유형의 데이터에 대해,
거의 무제한에 가까운 양의 데이터를 비용 효율적으로 수집, 통합, 변환, 관리할 수 있는 클라우드 네이티브 플랫폼



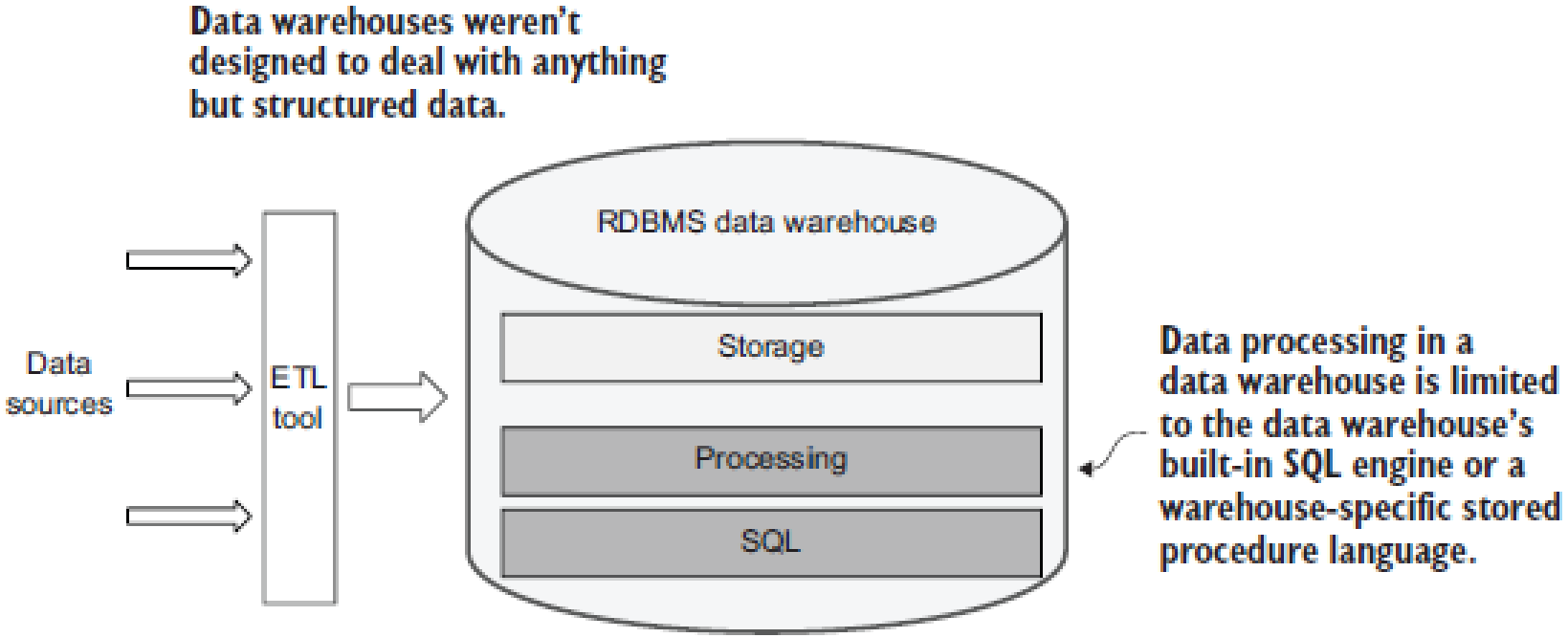
데이터 플랫폼 관련 동향

- SaaS 의 활용이 폭발적으로 증가하면서 수집 데이터의 다양성과 종류가 크게 증가(비정형, 반정형 데이터, 실시간 스트리밍)
- 마이크로서비스 세계에는 데이터를 가져올 중앙 운영 데이터베이스가 없기 때문에, 마이크로서비스에서 메시지를 수집하는 것이 가장 중요한 분석 작업 중 하나가 됨
- 비즈니스 사용자와 데이터 과학자들이 최신 분석툴을 사용해서 원시 데이터에 액세스 하는 경향이 크게 증가하고 있음



데이터웨어하우스의 한계

- SaaS API에 사용되는 반정형 데이터(JSON , Avro, Protocol Buffer)를 처리하지 못함
- 비정형 데이터(binary, image, video, audio data)를 처리하지 못함
- 구조화된 데이터의 빈번한 스키마 변경에 대응이 어려움
- 웨어하우스에 제공하는 SQL엔진과 저장 프로시저만 사용해야 하는 제약이 있음
- 스토리지와 처리영역이 결합되어 있어 확장성과 유연성에 제약이 큼
- 배치 중심 처리 방식으로 스트림 데이터가 처리가 어려움

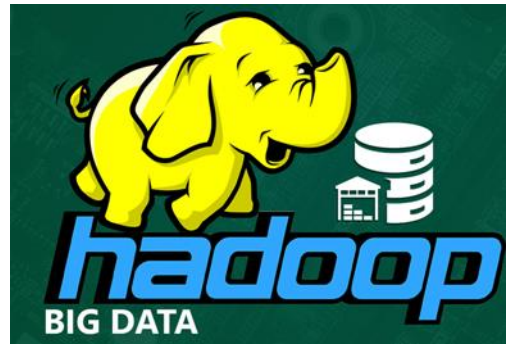


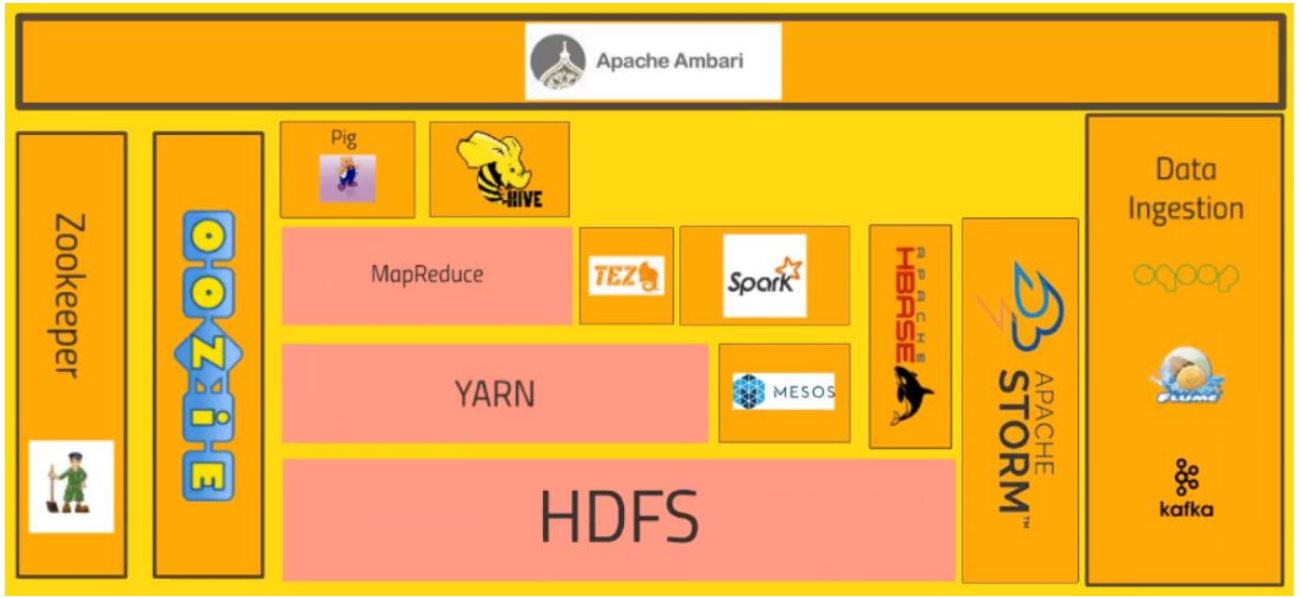
데이터 레이크가 대안이 될 수 있을까?

방대한 양의 원시 데이터를 보관하는 스토리지 리포지토리로, 특정 주제나 목적 중심이 아닌 통합되지 않은 데이터 집합
과거 10년 동안 Hadoop이 데이터 레이크에 대한 실질적인 표준이었음

Hadoop 의 단점

- 매우 복잡한 시스템
- 비즈니스 사용자가 비정형 형태를 이해하고 활용하기에 쉽지 않은 시스템
- 개발자도 잘 사용하려면 많은 노력이 필요
- 스토리지와 컴퓨팅이 분리되어 있지 않음
- 시스템 확장을 위한 하드웨어 추가/변경에 수개월이 소요될 수 있음





퍼블릭 클라우드 활용

하둡의 장점은 살리고 단점들을 보완해서 유연성을 훨씬 더 가져다 주는 솔루션이 클라우드와 함께 등장함

- 온디맨드 스토리지, 컴퓨팅 리소스 프로비저닝, 사용량 기반 요금 지불 모델을 갖춘 퍼블릭 클라우드의 등장으로 데이터 레이크 설계가 Hadoop의 한계를 뛰어 넘음
- 퍼블릭 클라우드를 통해 데이터 레이크는 설계 및 확장성 측면에서 더 많은 유연성을 포함하고 필요한 지원(support)의 양을 대폭 줄이는 동시에 비용 효율성을 높임
- 퍼블릭 클라우드의 출현은 분석 데이터 시스템에 관한 모든 것을 바꾸어 놓음
- AWS EMR은 관리형 서비스로 제공되며 AWS에서 Hadoop 및 Spark 작업을 실행할 수 있음

Elastic resources

Modularity(Storage and compute are separate)

Cloud turns CAPEX into OPEX

Pay per use

Managed services are the norm

Instant availability

A new generation of cloud-only processing frameworks

Faster feature introduction

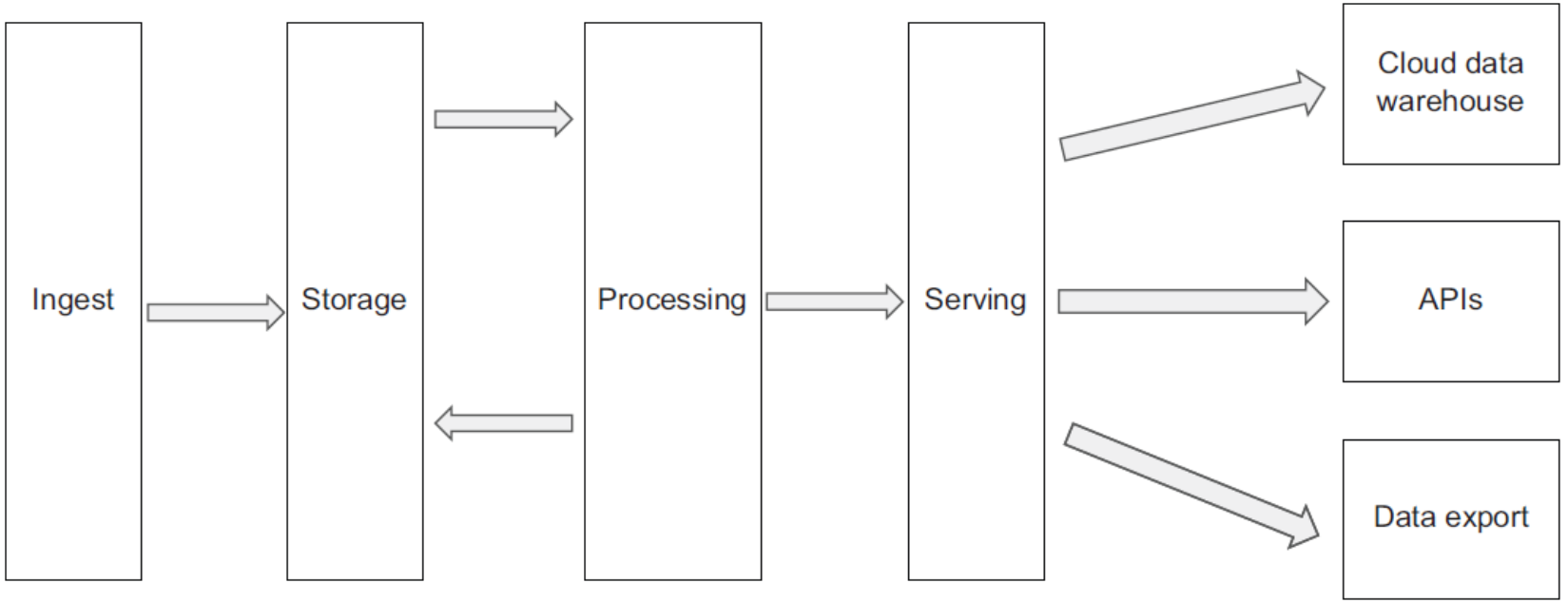
클라우드 플랫폼의 등장

데이터 레이크와 데이터 웨어하우스를 조합하는 방식으르 구성하는 것이 비용 효과적

- 적절하게 관리되며 정제된 데이터를 필요로 하는 요구사항에는 데이터 웨어하우스를 활용
- 모든 데이터에 대한 즉시 액세스하려는 사용자의 요구사항을 충족시키고자 할 경우에는 데이터 레이크 활용
- 비용 측면에서 데이터 레이크가 데이터 웨어하우스 보다 우수함
- 데이터 레이크를 활용할 경우 스트리밍과 같은 새로운 방식의 데이터 처리가 가능
- 클라우드, 클라우드 데이터 웨어하우스 및 클라우드 데이터 레이크에서 사용할 수 있는 새로운 처리 기술의 조합을 통해 클라우드에서 제공되는 모듈성, 유연성 및 탄력성을 더 잘 활용하여 요구 사항을 충족할 수 있음
- 데이터 소비자의 새로운 요구 사항을 해결하기 위해 새로운 기술과 클라우드 서비스를 활용하는 클라우드 데이터 플랫폼을 설계하는 것이 주제

클라우드 플랫폼의 빌딩 블록(building block)

데이터 플랫폼의 목적은 데이터 유형에 관계없이 가능한 가장 비용 효율적인 방식으로 데이터를 수집, 저장, 처리 및 분석에 사용할 수 있도록 하는 것



■ 수집 계층

- 관계형 또는 NoSQL 데이터베이스, 파일 저장소, 내부 또는 타사 API와 같은 다양한 데이터 소스에 접근하고 이들에서 데이터를 추출하는 일을 담당

클라우드 플랫폼의 빌딩 블록(building block)

■ 스토리지 계층

- 스토리지 시스템은 방대한 양의 데이터를 수용할 수 있도록 확장 가능하고 저렴해야 함
- 클라우드 스토리지는 파일 유형에 제한을 두지 않음 : CSV, JSON, Avro, Parquet, 이미지, 비디오 등

■ 클라우드 스토리지 장점

- 클라우드 스토리지는 클라우드 제공자가 완전히 관리

  유지 관리, 소프트웨어 또는 하드웨어 업그레이드 등에 대해 걱정할 필요가 없음
- 클라우드 스토리지는 탄력적

  요구 사항에 따라 볼륨을 늘리거나 줄임

  더 이상 미래의 수요를 예상하여 스토리지 시스템 용량을 과도하게 프로비저닝할 필요가 없음
- 사용한 용량만큼만 비용을 지불
- 클라우드 스토리지와 직접 연결된 컴퓨팅 리소스가 없음

  최종 사용자의 관점에서 보면 클라우드 스토리지에 연결된 가상 머신이 없음

  즉, 유휴 컴퓨팅 용량을 차지하지 않고도 대용량 데이터를 저장할 수 있음

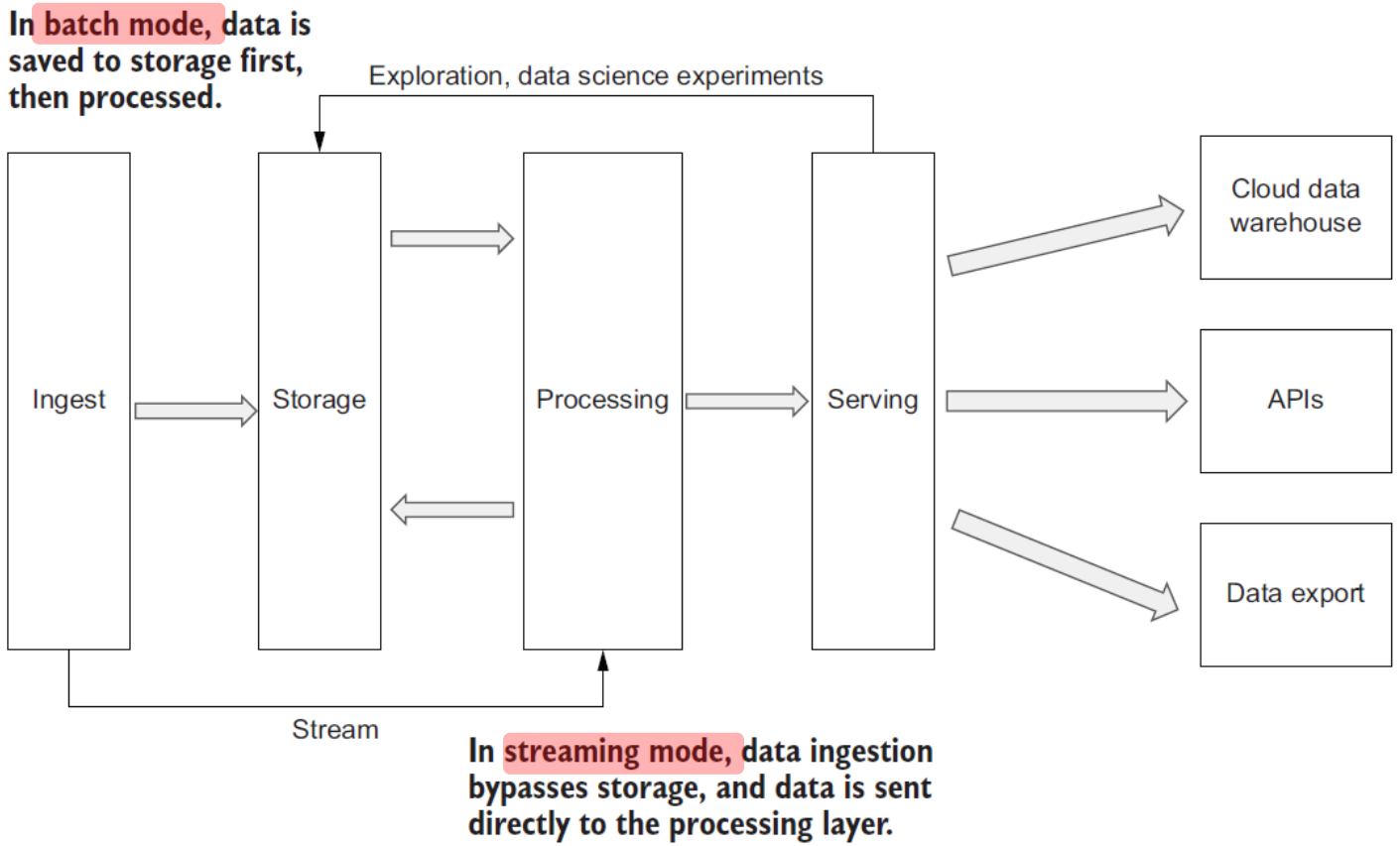
  데이터를 처리할 시간이 되면 쉽게필요에 따라 필요한 컴퓨팅 리소스를 프로비저닝하면 됨

클라우드 플랫폼의 빌딩 블록(building block)

■ 처리 계층

- 확장성과 최신 프로그래밍 언어 지원하고 클라우드 패러다임에 잘 통합되는 여러 처리 프레임워크가 개발되었음
  - Apache Spark, Apache Beam, Apache Flink
- 최신 프로그래밍 언어(Python , Java, Scala) 를 사용하여 데이터 변환, 유효성 검사 또는 정리 작업을 작성

■ 데이터 처리 방식



클라우드 플랫폼의 빌딩 블록(building block)

■ 서비스 계층

- 서빙 레이어의 목표는 최종 사용자가 사람이든 다른 시스템이든 사용할 수 있도록 데이터를 준비하는 것임
- 고급 사용자와 분석가는 애드혹 SQL 쿼리를 실행하고 몇 초 안에 응답을 받기를 원함
- 데이터 과학자와 개발자는 가장 익숙한 프로그래밍 언어를 사용하여 새로운 데이터 변환을 프로토타이핑 하거나 기계 학습 모델을 구축하고 결과를 다른 팀원과 공유하기를 원함
- 궁극적으로, 일반적으로 다양한 액세스 작업에 대해 서로 다른 특수 기술을 사용해야 함
- 그러나 좋은 소식은 클라우드를 통해 단일 아키텍처에서 쉽게 공존할 수 있다는 것
- 예를 들어 빠른 SQL 액세스를 위해 레이크에서 클라우드 데이터 웨어하우스로 데이터를 로드할 수 있음
- 다른 애플리케이션에 대한 데이터 레이크 액세스를 제공하기 위해 레이크에서 빠른 key/value 또는 document store로 데이터를 로드하고 애플리케이션이 이를 가리키도록 할 수 있음
- 클라우드 데이터 레이크는 Spark, Beam, Flink와 같은 프레임워크를 사용하여 클라우드 스토리지 데이터로 직접 작업할 수 있는 환경을 데이터 과학자 및 엔지니어링팀에게 제공함
- Jupyter Notebook, Apache Zeppelin과 같은 관리형 노트북 환경을 사용하여 검토 및 결과를 공유할 수 있는 공동 작업 환경을 구축할 수 있음

데이터 플랫폼 유스케이스 사례

데이터 플랫폼의 다양한 사용 사례를 이해하는 것이 데이터 플랫폼을 설계/계획할 때 중요
이러한 컨텍스트가 없으면 실제로 실제 비즈니스 가치를 제공하지 않는 데이터 늪에 빠질 위험이 있음

Customer 360

- 최고의 경험을 제공하고 고객의 평생 가치를 극대화하려면 고객이 채널을 이동할 때 각각의 행동을 이해하는 것부터 시작한다.
- 견고한 고객 경험을 제공은 Customer 360 애플리케이션을 구축으로 시작한다.
- Customer 360 애플리케이션은 데이터 소스를 통합하여 고객이 필요로 하는 것을 미리 예측하는 데 도움이 되는 동시에 보다 만족스러운 경험을 제공한다.

Thank you