

Part 04. [Chapter 2-2] Feast Feature Store 실습

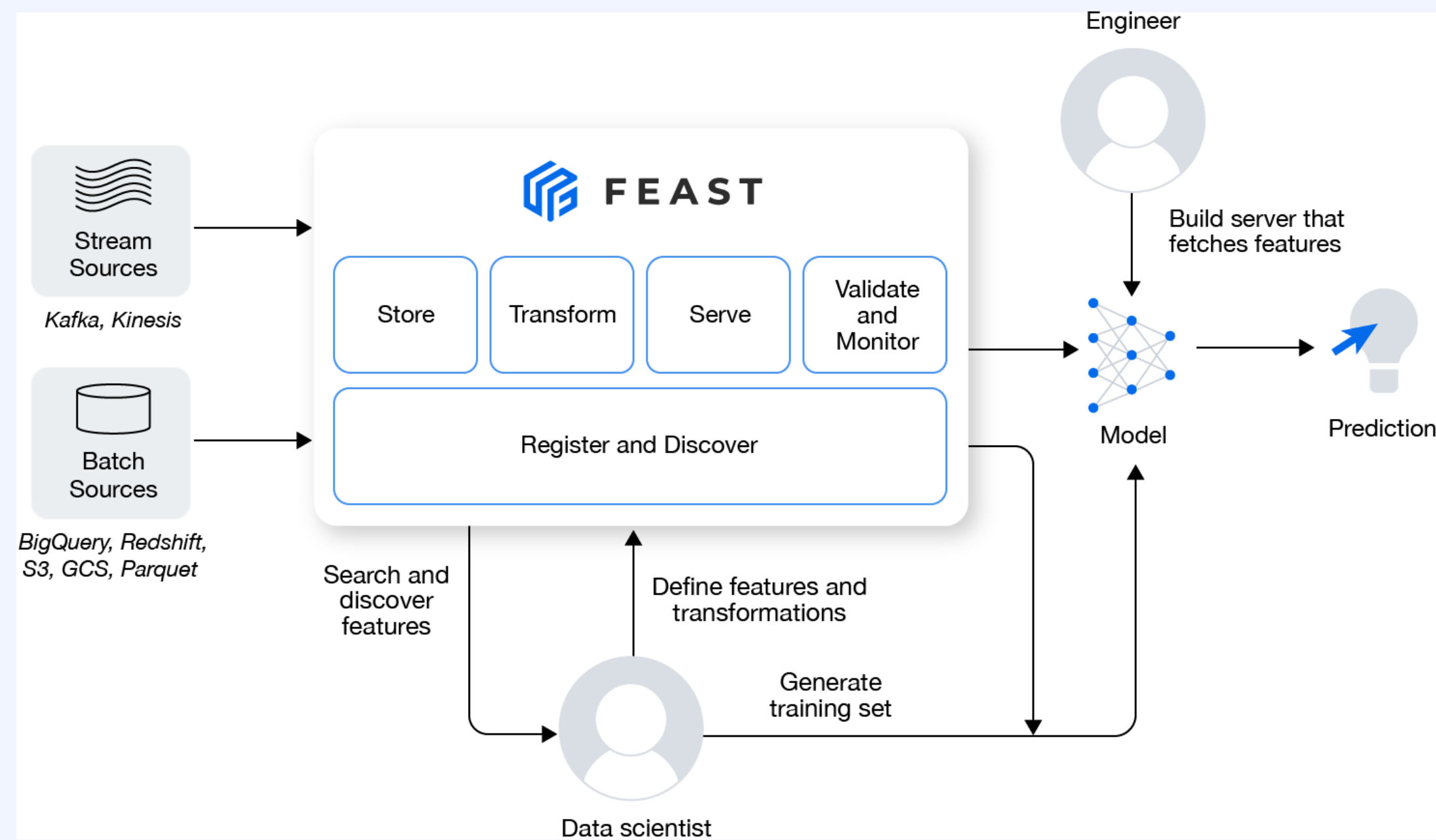
1 Feast 소개

Feast 란?

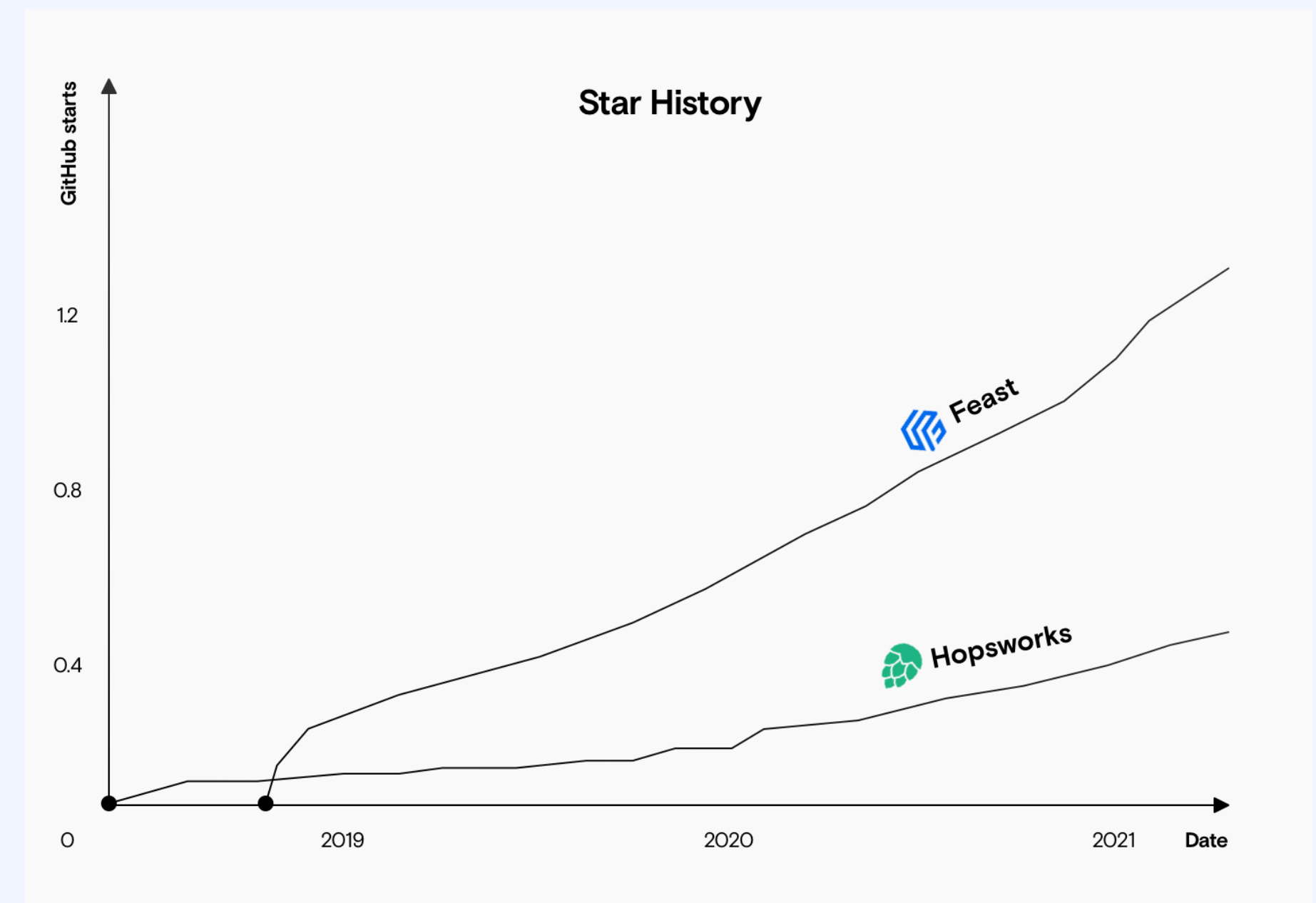
Feast 는 오픈 소스로 사용하기 쉽고 외부 시스템과 통합하기 쉬운 Feature Store 의 한 종류

1.

Feast 소개



출처 : <https://docs.feast.dev/>



출처 : <https://www.datarevenue.com/en-blog/feature-store-feast-vs-hopsworks>

공식 사이트 : <https://docs.feast.dev/>
Github : <https://github.com/feast-dev/feast>

Feast 란?

왜 Feast 를 배워 보려고 하는가?

1.

Feast 소개

Why Feast?



Feast is Open Source

- Compliments the rest of our stack
- Community is active and responsive
- Functionality and code organized and well documented

Feast is Extensible and Pluggable

- Easy to integrate with the rest of our stack

Feast's APIs/SDKs are simple and easy to use

- "We need something like Feast" -- Internal Users
- Desired language support included (Python/Java/Scala)

출처 : Global Feature Store Meetup #7 (https://www.youtube.com/watch?v=n_V3RYFg5Zo)

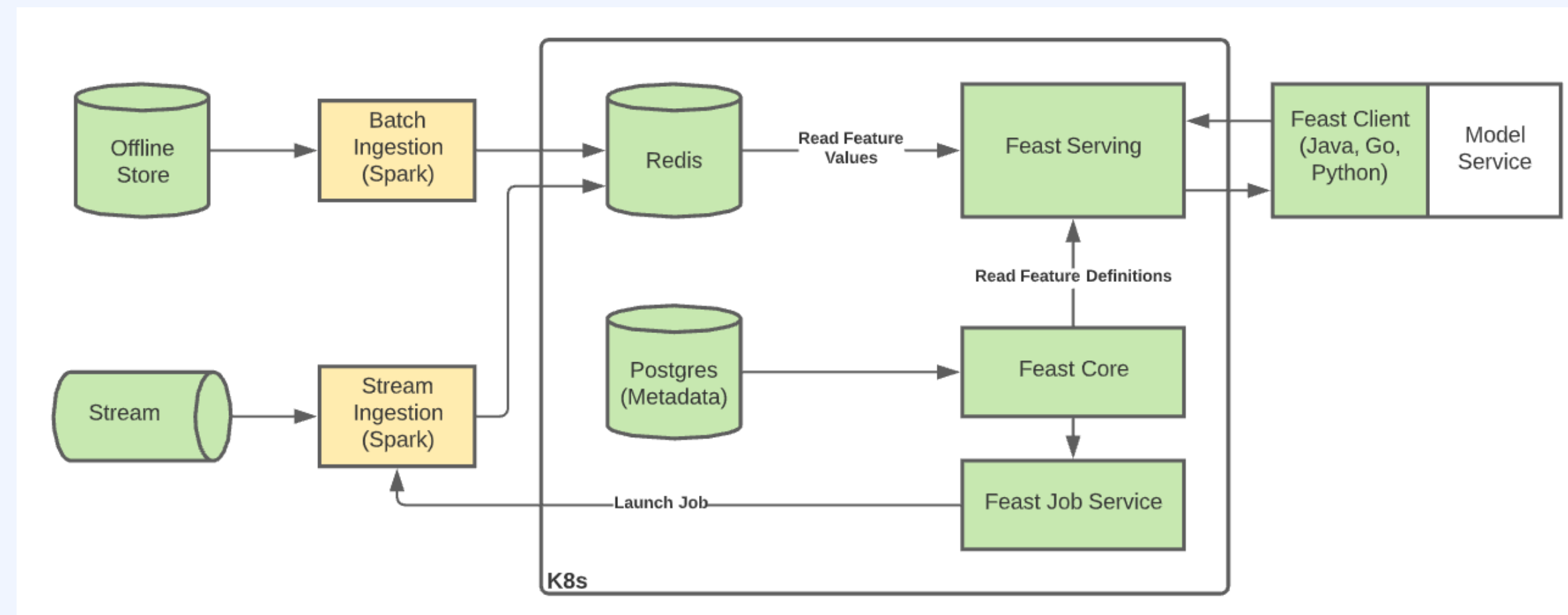
Feast 변화

Feast 는 최근 0.9 에서 0.10+ 로 변화하면서 많은 점들이 변경됨

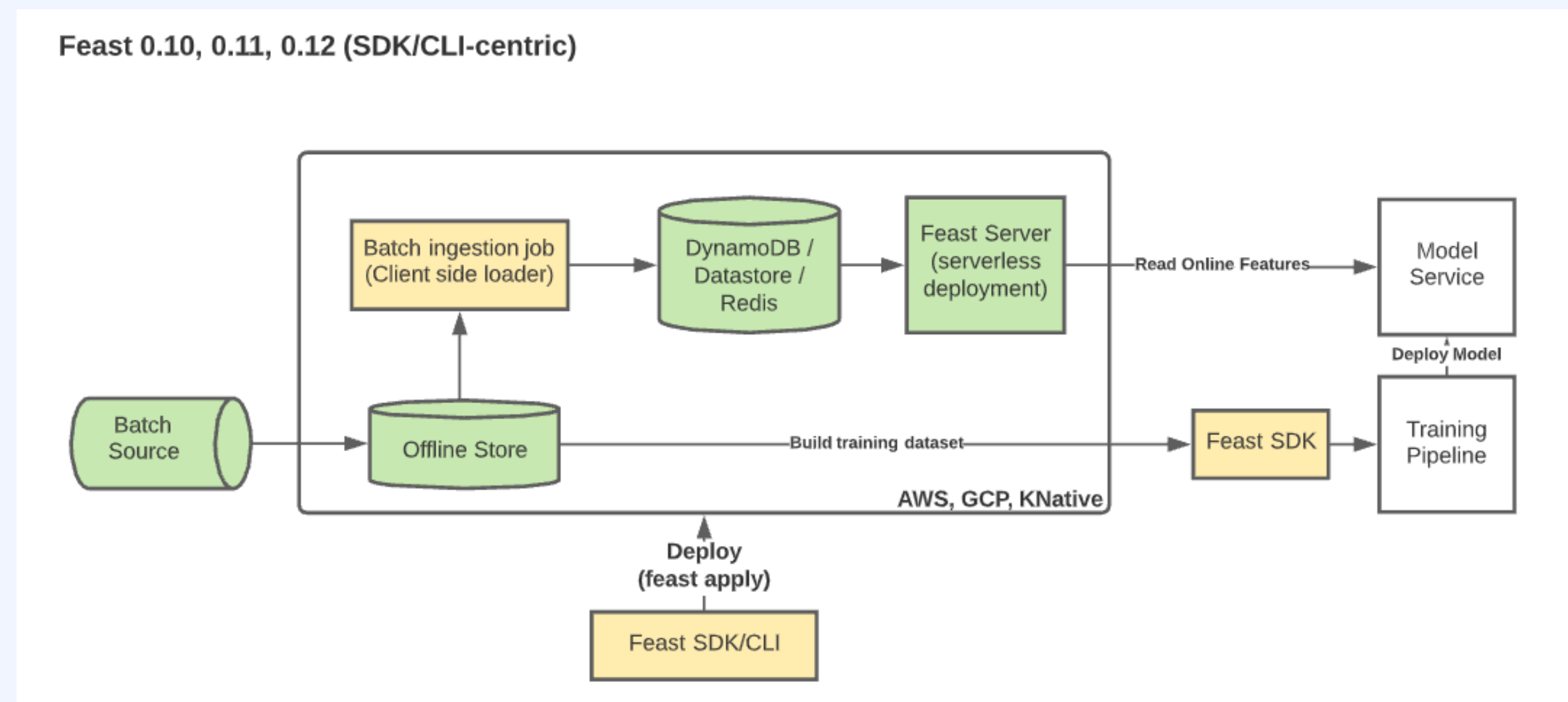
1.

Feast 소개

~ 0.9 →



0.10~ →



Feast 변화

Feast 는 최근 0.9 에서 0.10+ 로 변화하면서 많은 점들이 변경됨

1.
Feast 소개

Component	Feast 0.9	Feast 0.10~
아키텍처	<ul style="list-style-type: none">- 서비스 중심- 쿠버네티스 환경에 배포된 컨테이너 와 서비스	<ul style="list-style-type: none">- SDK/CLI 중심 소프트웨어- Feature store 사용 가능하도록 인프라 설정 및 배포 가능
설치 방법	<ul style="list-style-type: none">- Terraform 또는 Helm 이용	<ul style="list-style-type: none">- SDK/CLI 환경에서 pip 인스톨- apply 명령을 사용한 GCP, AWS 등의 클라우드 배포
필요한 인프라	<ul style="list-style-type: none">- Kubernetes, Postgres, Spark, Docker, Object Store	<ul style="list-style-type: none">- 없음
Batch 계산	<ul style="list-style-type: none">- 가능 (Spark 기반)	<ul style="list-style-type: none">- Python 을 이용해 데이터 다루기- 데이터 웨어하우스 활용
Streaming 지원	<ul style="list-style-type: none">- 가능 (Spark 기반)	<ul style="list-style-type: none">- 계획되어 있음 (apply 이용)

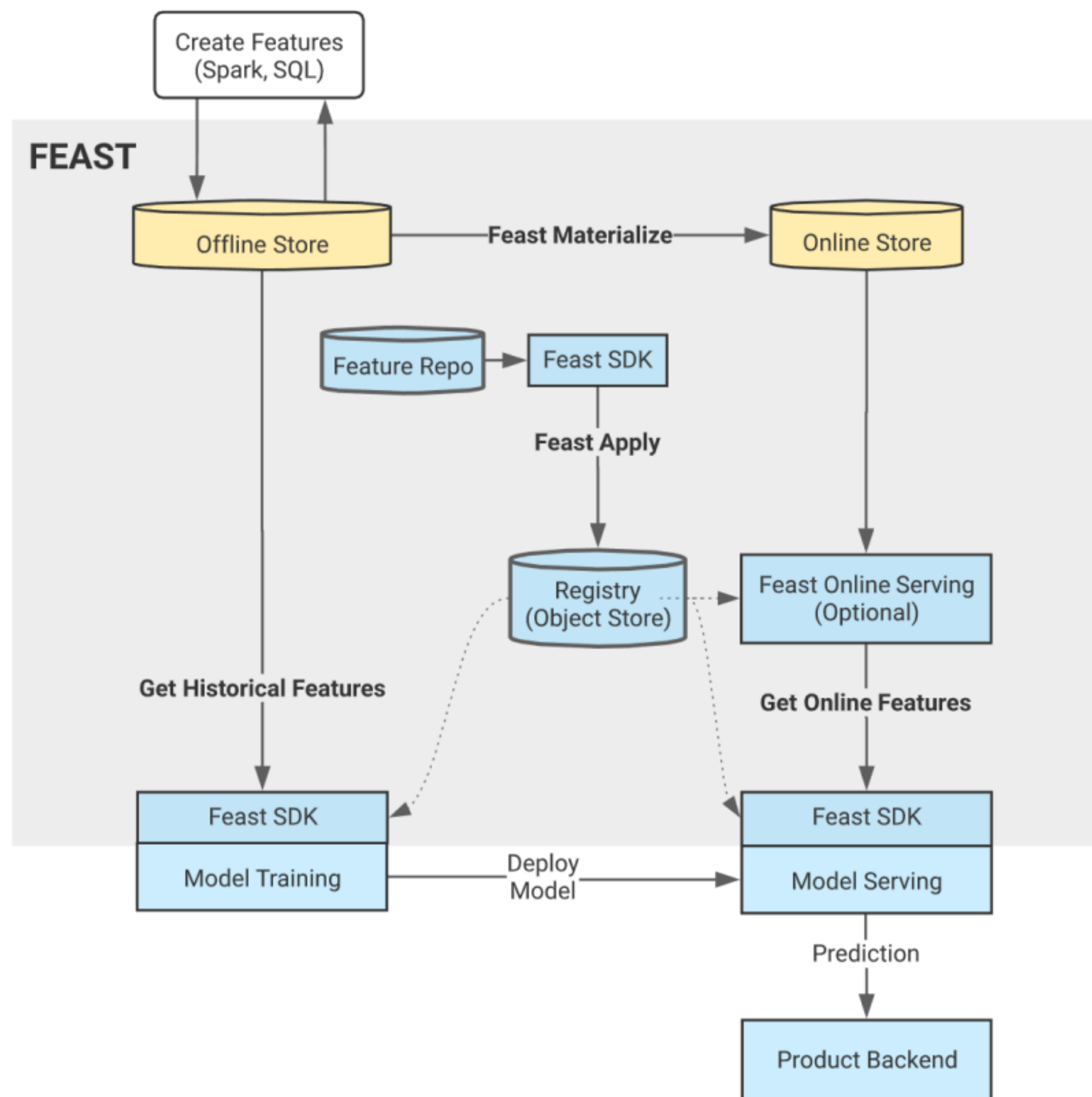
Component	Feast 0.9	Feast 0.10~
오프라인 저장소	<ul style="list-style-type: none">- 없음 (Spark 를 사용한 활용 제한)	<ul style="list-style-type: none">- BigQuery, Redshift, Snowflake, Custom stores
온라인 저장소	<ul style="list-style-type: none">- Redis	<ul style="list-style-type: none">- DynamoDB, Firestore, Redis 이외 다수 계획 중
Job Manager	<ul style="list-style-type: none">- 있음	<ul style="list-style-type: none">- 없음
Registry	<ul style="list-style-type: none">- Postgres 백엔드 기반 gRPC	<ul style="list-style-type: none">- SDK 활용 가능한 파일 기반
Local Mode	<ul style="list-style-type: none">- 없음	<ul style="list-style-type: none">- 있음

Feast Component 소개

Feature Repo 부터 온라인 Serving 까지 SDK 를 이용하여 쉽게 조작 가능

1.

Feast 소개



Part 04. [Chapter 2-2] Feast Feature Store 실습

2 Feast 기본 예제 - (1) Store 생성과 배포 - 1

Part 04. [Chapter 2-2] Feast Feature Store 실습

2 Feast 기본 예제 - (1) Store 생성과 배포 - 2

Part 04. [Chapter 2-2]

Feast Feature Store 실습

3 Feast 기본 예제 – (2) Feature 추출 및 추론

Part 04. [Chapter 2-2]

Feast Feature Store 실습

4 Feast Server 생성

Part 04. [Chapter 2-2]

Feast Feature Store 실습

5 Store 구축과 Minio 연계 실습

Part 04. [Chapter 2-2] Feast Feature Store 실습

6 Feast 와 MLFlow 를 활용한
머신러닝 프로젝트 적용

Part 04. [Chapter 2-2]

Feast Feature Store 실습

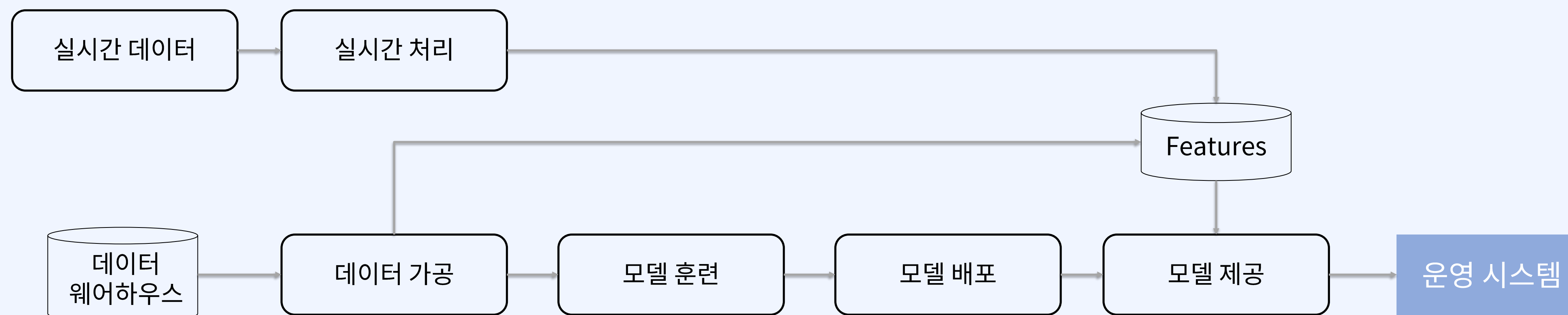
7 Feast 를 활용한 ML 라이프사이클

Feast 를 적용하지 않는 경우

실시간 데이터 처리와 변형을 통한 Feature 생성, 모델 훈련과 제공이
하나의 Pipeline 에서 이루어지는 End-to-End 시스템

7.

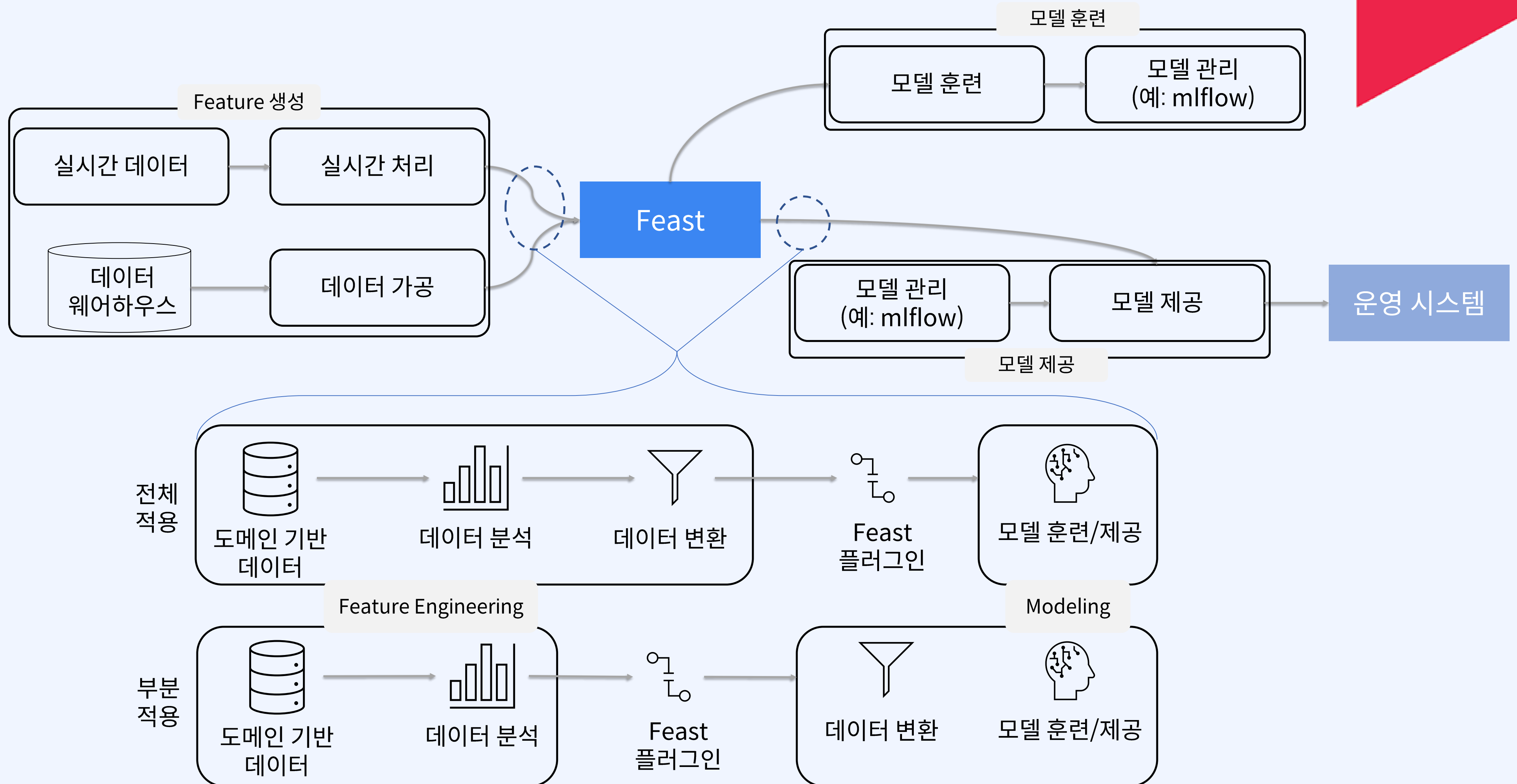
Feast 를 활용한
ML 라이프사이클



Feast 를 적용한 경우

Feature 생성, 모델 훈련, 모델 제공의 3가지 영역

7. Feast 를 활용한 ML 라이프사이클



Feast 를 이용해 할 수 있는 일들

Feast 를 이용해 할 수 있는 일들

7.

Feast 를 활용한
ML 라이프사이클

	모델 제공	데이터 변환	모니터링	레지스트리	저장소
역할	예측이 필요한 때 새로운 Feature 값을 제공하는온라인 제공	새로운 데이터 발생 즉시 Feature 로 변환	ML서비스들의 전체적인 동작 상태를 나타내는 지표들을 모니터링하고 운영 시 파악 가능한 Tool 관리	표준화된 Feature 정의와 메타데이터 전용 레지스트리	S3, BigQuery, Snowflake, Redshift 와 같은 다양한 Feature 제공 시스템의 요구사항을 지원하는 저장 역할
방법	낮은 latency 의 DB 기반 고성능 API 를 통한 Responses	- Batch 변형 - 실시간 변형 - On-demand 변형	Feature store 의 사용성,용량,가용성,불량 또는 Feature 제공시 throughput, 지연, 에러율에 관련된 지표들	데이터 추가를 위한 일정 및 설정, 데이터 변환 및 저장, Feature 제공이 필요한 경우 언제든지 접근 가능한 제공 API 를 위한 자동화된 작업들	DynamoDB, Redis, Cassandra 와 같은 key-value 타입 저장소들이 낮은 latency 로 추론할 수 있는 Feature 값제공

- 실시간/Batch 데이터의 추가와 저장을 관리한다.
- 환경에 관계없는 Feature 정의 표준화가 가능해진다.
- 참조를 통해 Feature 의 공유와 재사용이 쉬워진다.
- 훈련과 제공에 필요한 데이터 일관성이 유지된다.
- 모델 훈련을 위한 Feature 를 적기에 활용할 수 있게 해준다,
- Feature 의 tracking, validating, monitoring 을 통해 모델 성능 유지 가능해진다.

Feast 를 이용해 할 수 없는 일들

Feast 를 이용해 할 수 없는 일들

7.
Feast 를 활용한
ML 라이프사이클

작업 흐름 관리	데이터 웨어하우스 데이터 레이크	데이터 탐색 시스템	파이프라인 생성
Airflow, Luigi	Hive, Snowflake	DataHub	Kubeflow
Feature Engienering Tool	데이터 버전 관리	모델 제공 메타데이터 관리	팀 협업 활동
Pandas, Spark	DVC, Pachyderm	Seldon, Bento	