Assignment | Tredence | Data Scientist

A. What is Feature Store? What problem it is supposed to solve?

Feature store is a centralized place for creating, managing, and searching for features used in machine learning pipelines. Features are properties that used as input to a machine learning model. Features could be basic or even static properties such as user address, age, education etc. or could be calculated features such as average web clicks in the last hour or max transaction volume per user etc.

When a company embraces feature stores, it allows data professionals across teams to follow the same general workflow for any machine learning use case — regardless of the challenges they're currently addressing (such as classification and regression, time series forecasting etc.). This workflow is typically implementation-agnostic, which means it can be easily adopted for use with new algorithm types and frameworks, such as classical ML algorithm alongside the newer deep learning frameworks.
Another major benefit of using feature stores is the time savings it creates. The stage in any modelling effort where features are created tends to be the most time-consuming; this sensitive process requires that features be calculated correctly, with thousands of features being created at a time and computed in a production environment in the exact same way they were computed offline during research. The use of a feature store makes the process of creating features much more streamlined and efficient.
In that case the best option is to use a centralized feature store.
Ateam can gain much value from building and maintaining a centralized feature store where different data professionals across the company can each create and manage canonical features to be used by other members of the team. This allows data scientists to easily add features they've built into a shared feature store. Once features are there, they are easy to consume both online (in production) and offline (in research), simply by referencing a feature's simple canonical name. Today, we have thousands of features in our feature store that are used in a variety of machine learning projects across the company and across all domains. Our data scientists are adding new features all the time, with new features calculated automatically and updated daily. This has allowed our team members to avoid repeat work, and easily access a wealth of data they need for modelling and research purposes.

B. What are building blocks of feature Store? Define & Research on each of the building blocks.

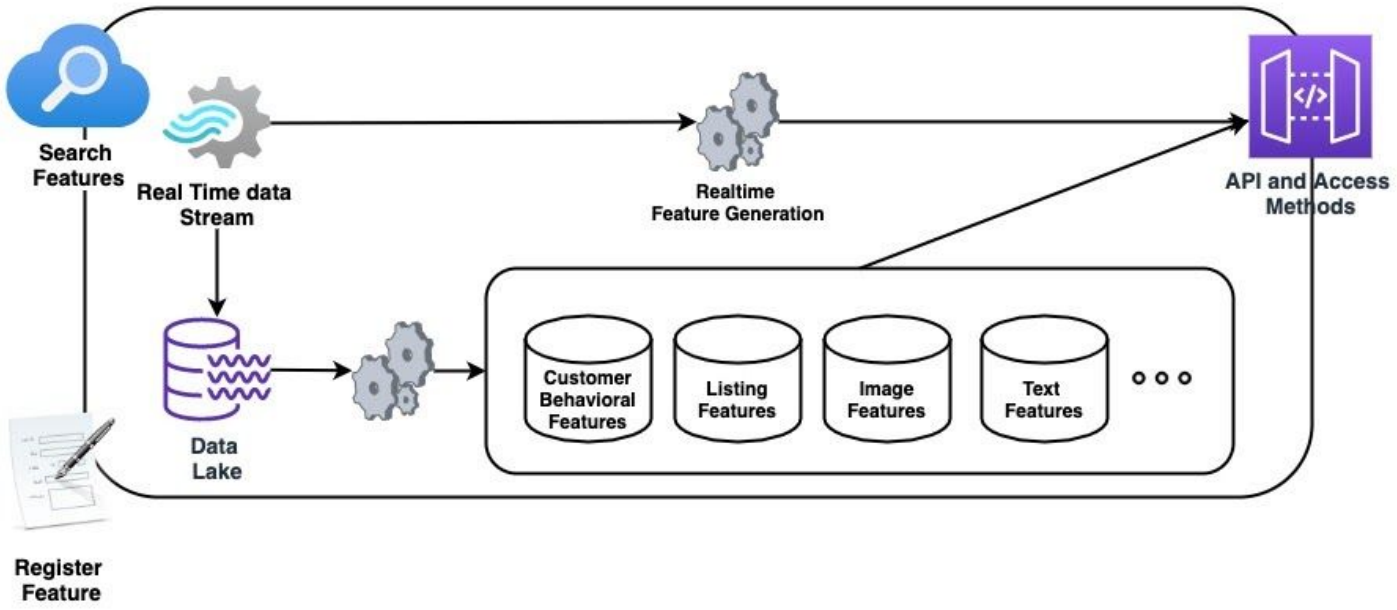
Here are some of the common techniques to generate features:

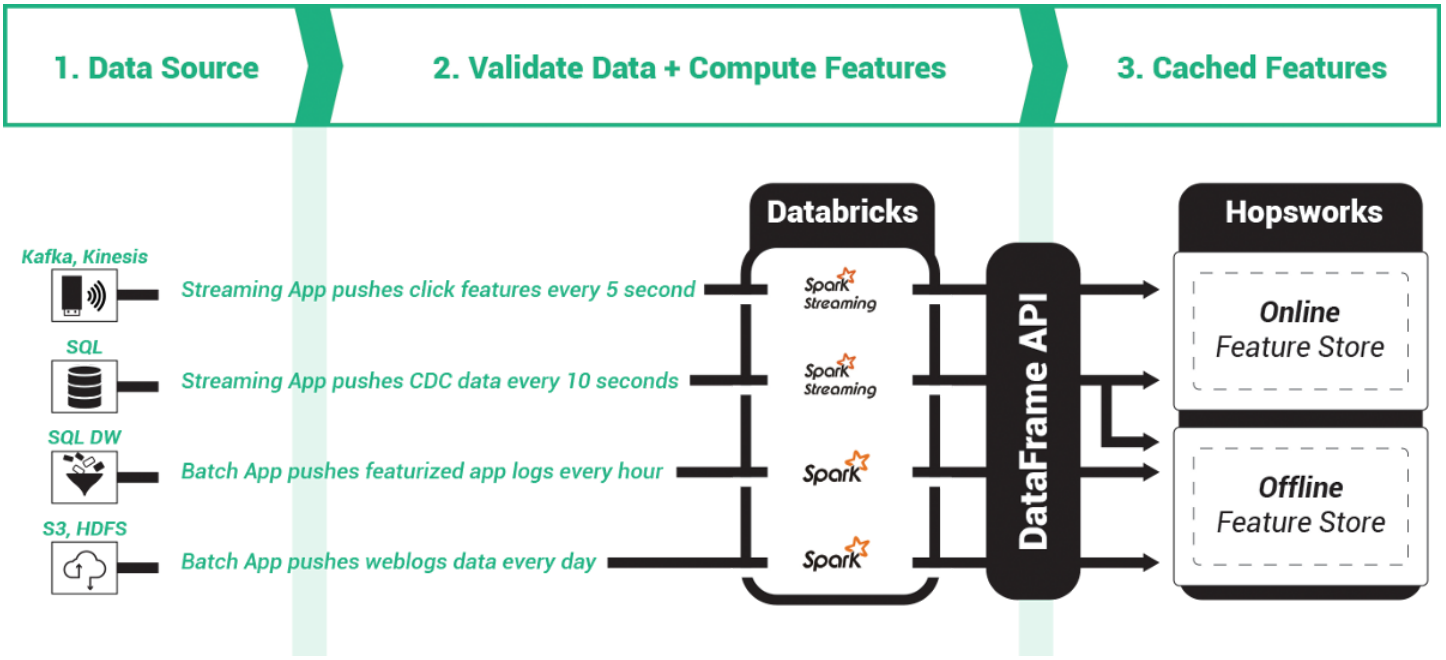
- Converting categorical data into numeric data
- Normalizing data
- One-hot-encoding
- Feature binning (converting continuous features into a discrete value with a different bucket)
- Embedding by reducing high dimensional data into lower space (especially image and text data)
- The statistical description of data and metrics (mean, median, stddev, IQR)

ML Feature store provides a centralized location for registering, searching, and using those important features. The following picture shows the high-level components of the feature store related to real-estate data products. It is composed of the two main layers: ***batch features*** and ***real-time features***.

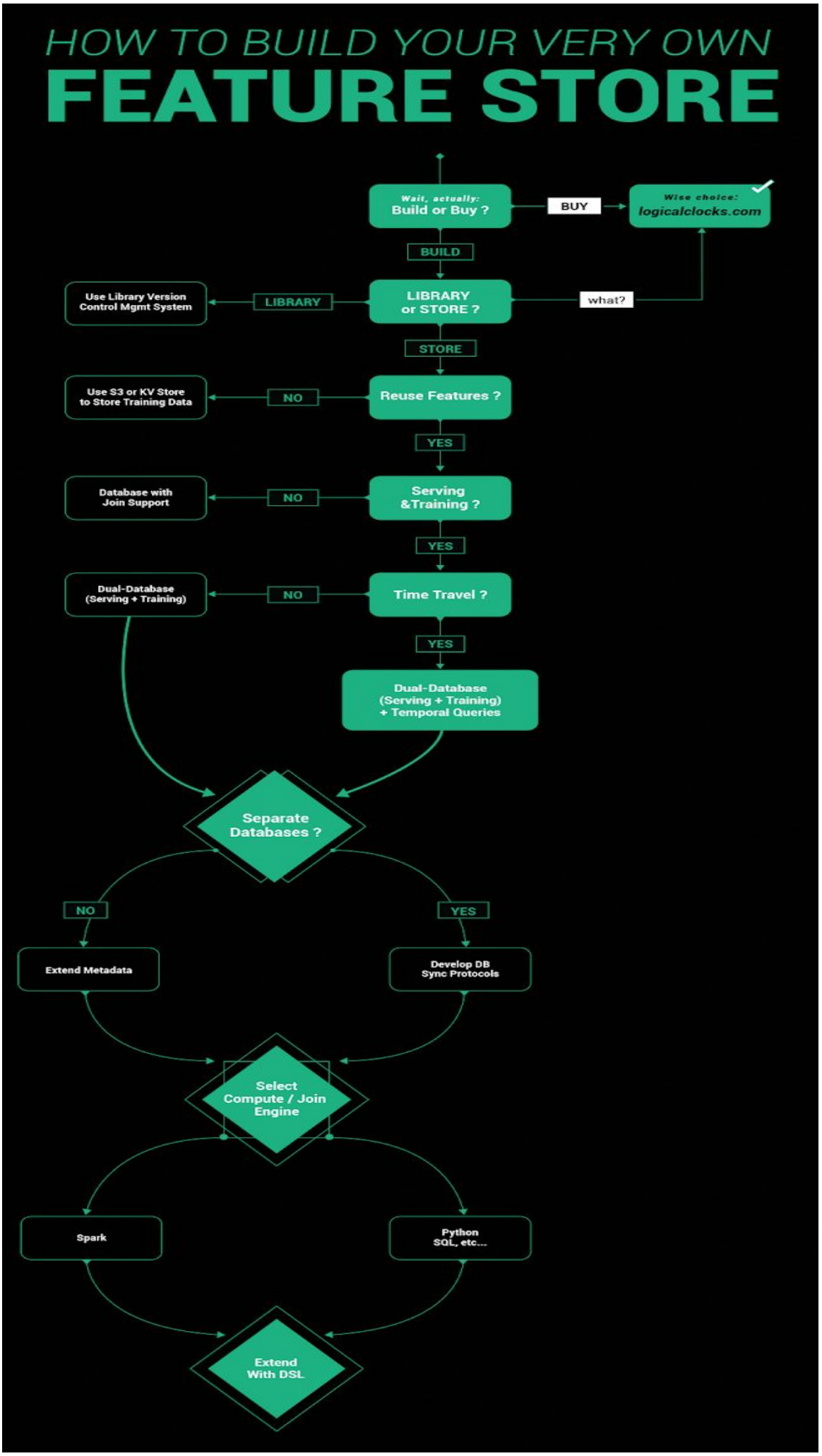
In batch features, all features are extracted from raw data in the data lake through batch processing. These features are usually extracted by using SQL engines like AWS Athena or AWS Glue/Spark. The extracted features are then persisted to a database such as DynamoDB or an object file system such as AWS S3.

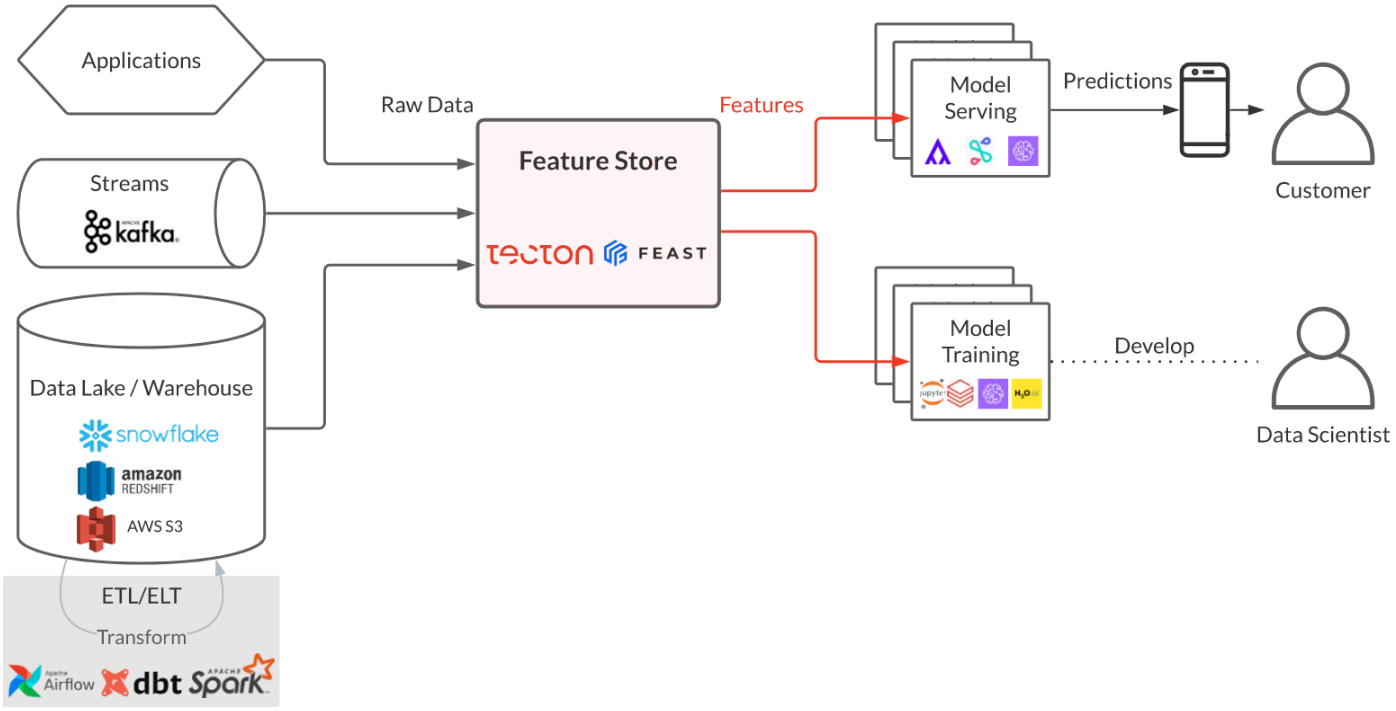
Real-time features are a little bit more complicated due to their SLA. In some applications, such as fraud detection, these need to be completed in less than 100 ms. AWS streaming Glue and Elasticache frameworks are usually used to realize real-time pipelines.





C. **Come up with a process flow block diagram for feature Store.**

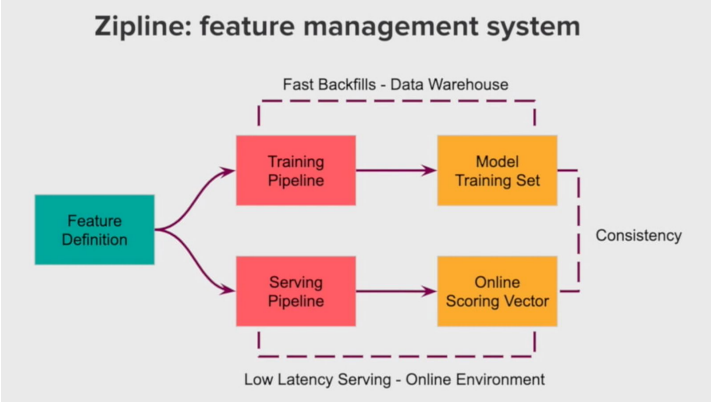


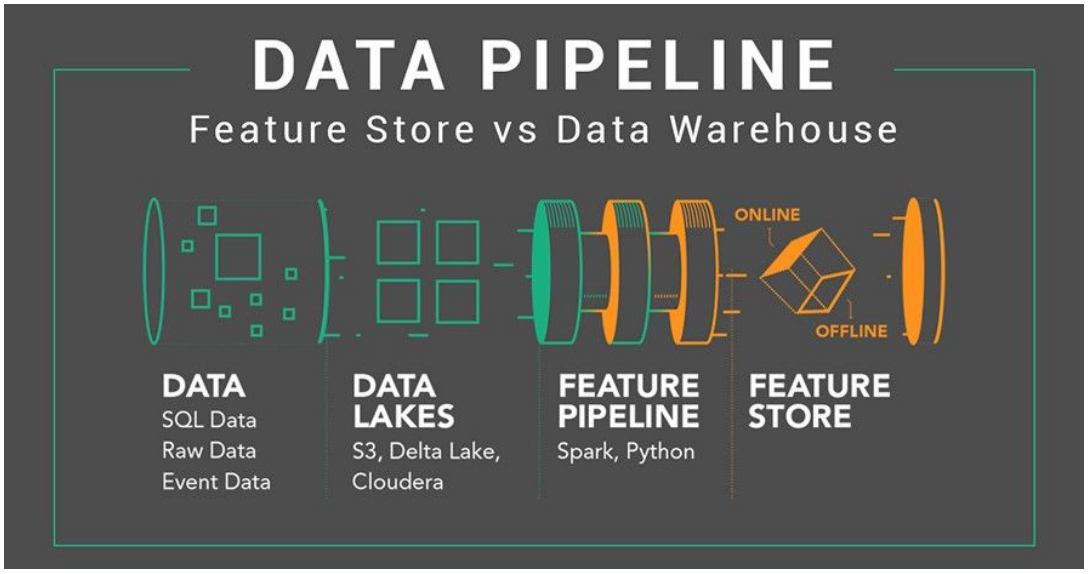


D. Do Competitor Analysis on Feature store and list down their Features & Capabilities.

The feature store is a data warehouse of features for machine learning (ML). Architecturally, it differs from the traditional data warehouse in that it is a dual-database, with one database (row-oriented) serving features at low latency to online applications and the other database (column-oriented) storing large volumes of features, used by Data Scientists to create train/test datasets and by batch applications doing offline model scoring.

- Data recency: The daily exports scenario gives us features made from 1-day stale data, and that may not be acceptable for a system that's sensitive to recent data or biases more towards intra-day events.
- Throughput: Table exports can be extremely large, and to enable newest features right after exports land, the feature store must be able to sustain high throughput to compute offline features.
- Online-offline consistency: The offline system that computes daily features and the online system that gives you the most up-to-date features must output the same result given the same inputs. We should expect identical setup if we were to bring an offline model to an online production environment.
- Feature repository: There are many features that can be shared among different models. This allows collaboration of multiple ML researchers and reduces duplication of effort.
- Monitoring system: If a feature recently changed anomalously, the researchers should be alerted in case their model isn't trained to be robust under novel scenarios.





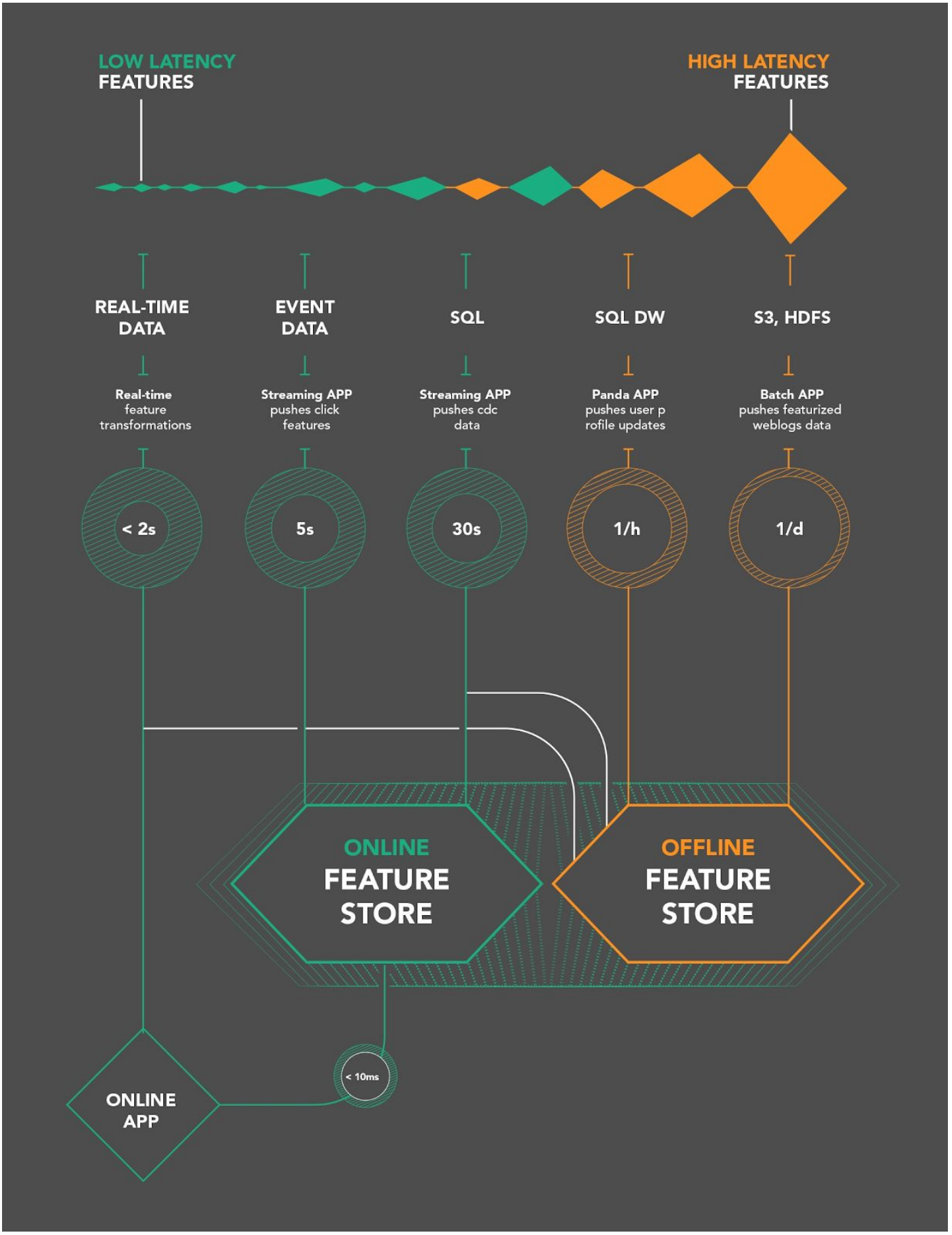
Feature Store can be implemented as Dual Database, find the below image for better understanding.

The main architectural difference between a data warehouse and a feature store is that the data warehouse is typically a single columnar database, while the feature store is typically implemented as two databases:

- an offline feature store for serving large batches of features to (1) create train/test datasets and (2) batch applications scoring models using those batches of features, and
- an online feature store for serving a single row of features (a *feature vector*) to be used as input features for an online model for an individual prediction.

The offline feature store is typically required to efficiently serve and store large amounts of feature data, while the online feature store is required to return feature vectors in very low latency (e.g., < 10ms). Examples of databases used for the offline feature store are Apache Hive and BigQuery and examples of online feature stores include MySQL Cluster, Redis, and DynamoDB.

Note that if you want to reuse features in different train/test datasets for different models, your database or application will need to join features together. This is true for both the offline and online feature stores. If your feature store does not support joining features, that is, you do not reuse features across different models, you (or some system) will need to create a new ingestion pipeline for every new model you support in production.



Reference:

1. https://towardsdatascience.com/the-importance-of-having-a-feature-store-e2a9cfa5619f
2. https://hopsworks.readthedocs.io/en/1.1/featurestore/featurestore.html
3. https://towardsdatascience.com/what-are-feature-stores-and-why-are-they-critical-for-scaling-data-science-3f9156f7ab4
4. https://www.logicalclocks.com/blog/how-to-build-your-own-feature-store
5. https://medium.com/data-for-ai/feature-store-vs-data-warehouse-306d1567c100