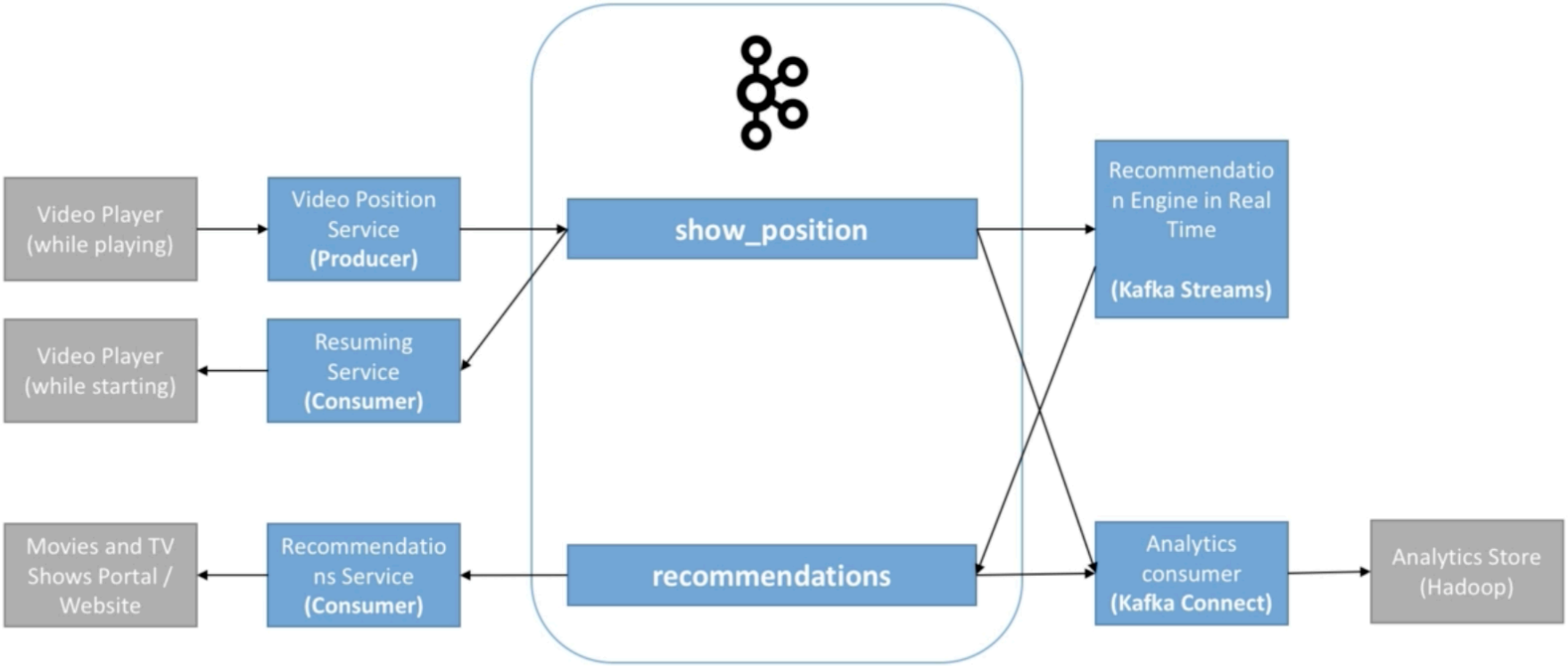
Video Analytics - MovieFlix

- *MovieFlix is a company that allows you to watch TV Shows and Movies on demand. The business wants the following capabilities:*

- Make sure the user can resume the video where they left it off
- Build a user profile in real time
- Recommend the next show to the user in real time
- Store all the data in analytics store

- How would you implement this using Kafka?

Video Analytics – MovieFlix Architecture



Video Analytics – MovieFlix Comments

- show_position topic:
  - is a topic that can have multiple producers
  - Should be highly distributed if high volume > 30 partitions
  - If I were to choose a key, I would choose "user_id"
- recommendations topic:
  - The kafka streams recommendation engine may source data from the analytical store for historical training
  - May be a low volume topic
  - If I were to choose a key, I would choose "user_id"

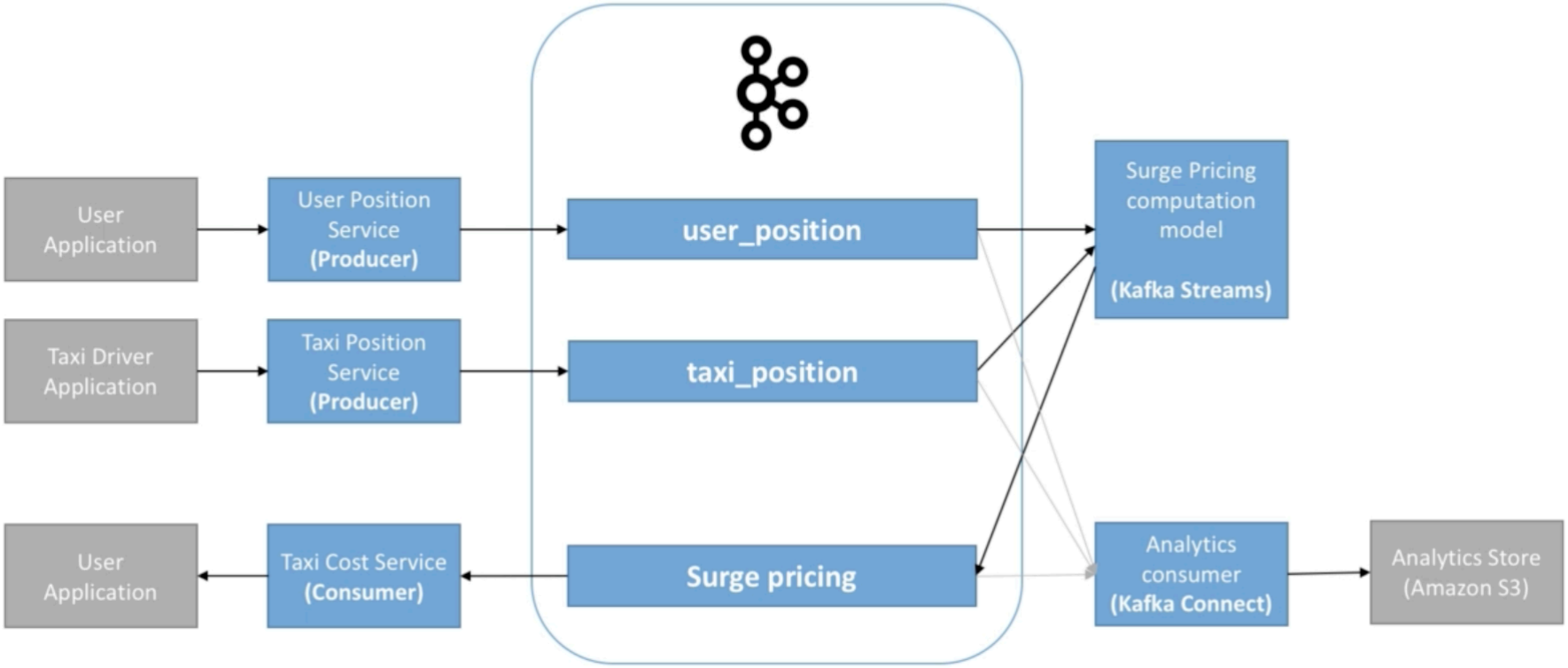
IOT Example - GetTaxi

- *GetTaxi is a company that allows people to match with taxi drivers on demand, right-away. The business wants the following capabilities:*

- The user should match with a close by driver
- The pricing should "surge" if the number of drivers are low or the number of users is high
- All the position data before and during the ride should be stored in an analytics store so that the cost can be computed accurately

- How would you implement this using Kafka?

IOT Example - GetTaxi Architecture



IOT Example - GetTaxi Comments

- taxi_position, user_position topic:s
  - Are topics that can have multiple producers
  - Should be highly distributed if high volume > 30 partitions
  - If I were to choose a key, I would choose "user_id", "taxi_id"
  - Data is ephemeral and probably doesn't need to be kept for a long time

- surge_pricing topic:
  - The computation of Surge pricing comes from the Kafka Streams application
  - Surge pricing may be regional and therefore that topic may be high volume
  - Other topics such as "weather" or "events" etc can be included in the Kafka Streams application

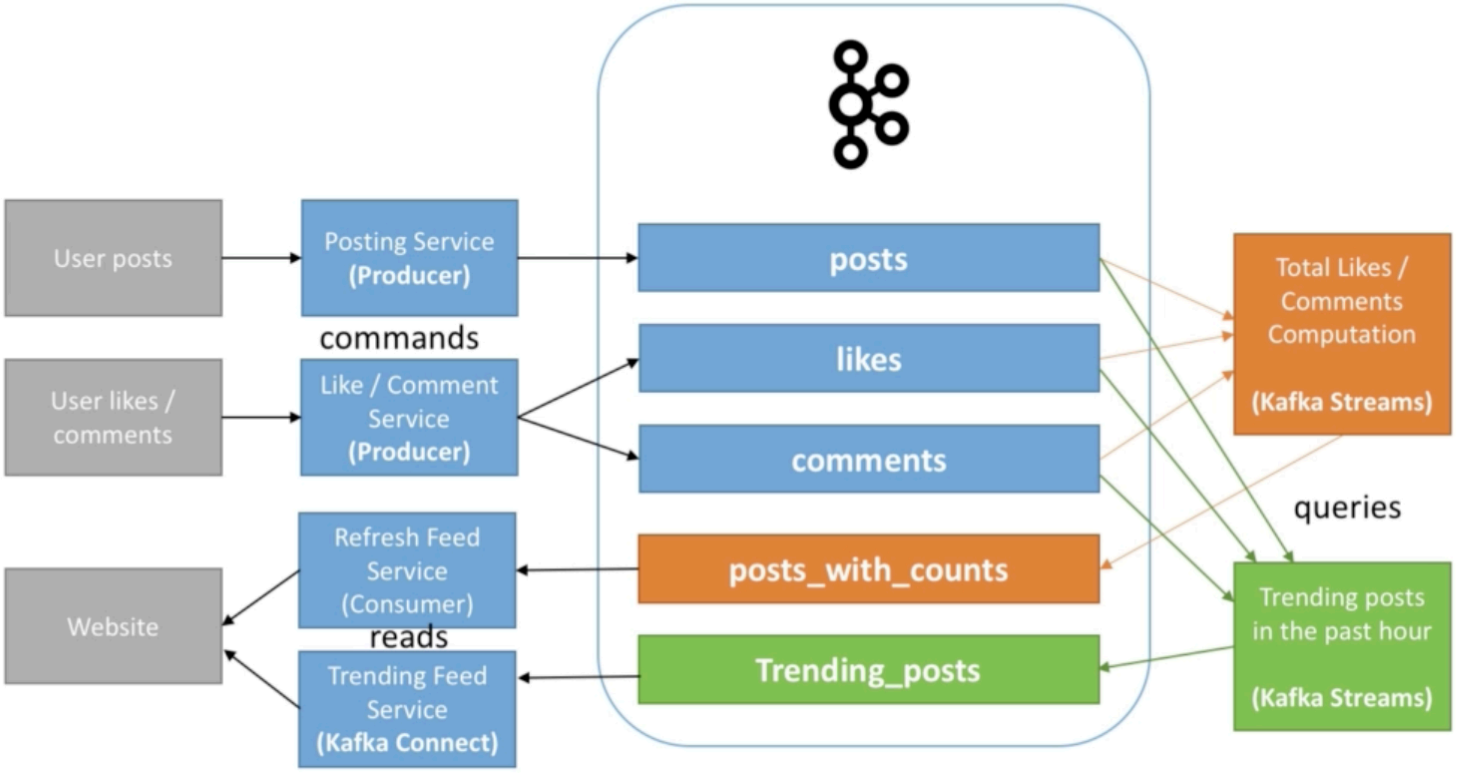
CQRS – MySocialMedia

- *MySocialMedia is a company that allows you people to post images and others to react by using "likes" and "comments". The business wants the following capabilities:*

- Users should be able to post, like and comment
- Users should see the total number of likes and comments per post in real time
- High volume of data is expected on the first day of launch
- Users should be able to see "trending" posts

- How would you implement this using Kafka?

CQRS – MySocialMedia Architecture



CQRS – MySocialMedia Comments

- Responsibilities are "segregated" hence we can call the model CQRS (Command Query Responsibility Segregation)
- Posts
  - Are topics that can have multiple producers
  - Should be highly distributed if high volume > 30 partitions
  - If I were to choose a key, I would choose "user_id"
  - We probably want a high retention period of data for this topic
- Likes, Comments
  - Are topics with multiple producers
  - Should be highly distributed as the volume of data is expected to be much greater
  - If I were to choose a key, I would choose "post_id"
- The data itself in Kafka should be formatted as "events":
  - User_123 created a post_id 456 at 2 pm
  - User_234 liked post_id 456 at 3 pm
  - User_123 deleted a post_id 456 at 6 pm

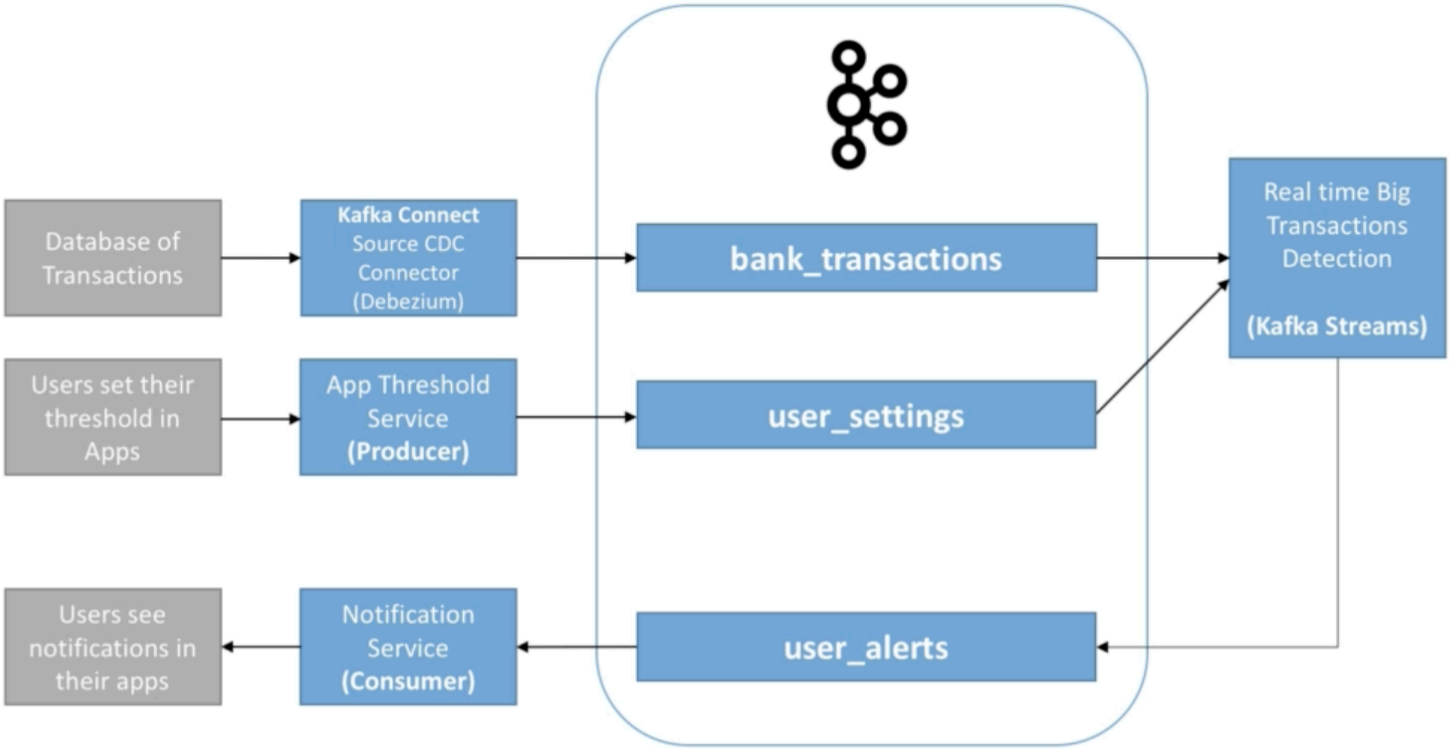
Finance application - MyBank

- *MyBank is a company that allows real-time banking for its users. It wants to deploy a brand new capability to alert users in case of large transactions*

- The transaction data already exists in a database
- Thresholds can be defined by the users
- Alerts must be sent in real time to the users

- How would you implement this using Kafka?

Finance application – MyBank Architecture



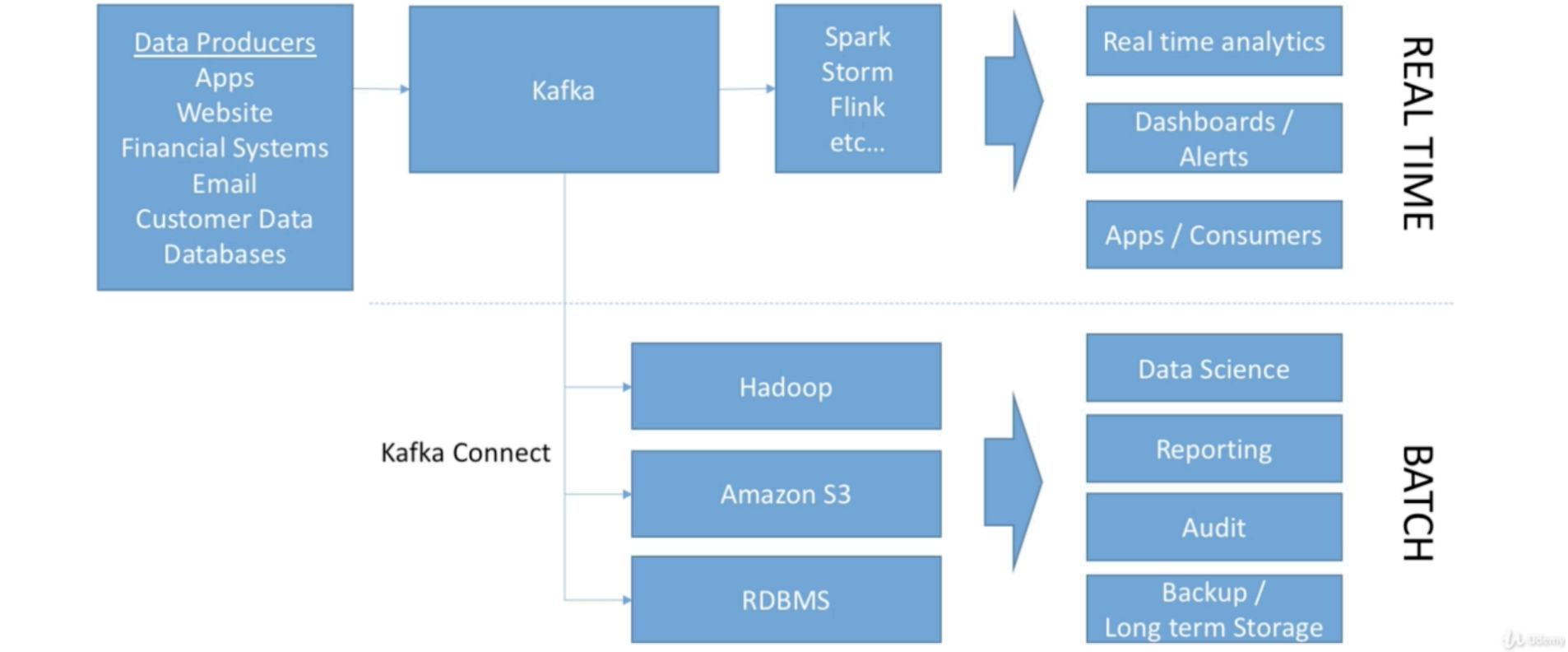
Finance application – MyBank Comments

- Bank Transactions topics:
  - Kafka Connect Source is a great way to expose data from existing databases!
  - There are tons of CDC (change data capture) connectors for technologies such as PostgreSQL, Oracle, MySQL, SQLServer, MongoDB etc…
- Kafka Streams application:
  - When a user changes their settings, alerts won't be triggered for past transactions
- User thresholds topics:
  - It is better to send events to the topic (User 123 enabled threshold at $1000 at 12 pm on July 12th 2018)
  - Than sending the state of the user: (User 123: threshold $1000)

Big Data Ingestion

- It is common to have "generic" connectors or solutions to offload data from Kafka to HDFS, Amazon S3, and ElasticSearch for example
- It is also very common to have Kafka serve a "speed layer" for real time applications, while having a "slow layer" which helps with data ingestions into stores for later analytics
- Kafka as a front to Big Data Ingestion is a common pattern in Big Data to provide an "ingestion buffer" in front of some stores

Big Data Ingestion



Logging & Metrics Aggregation