**Chaktim Wong - 50280143**

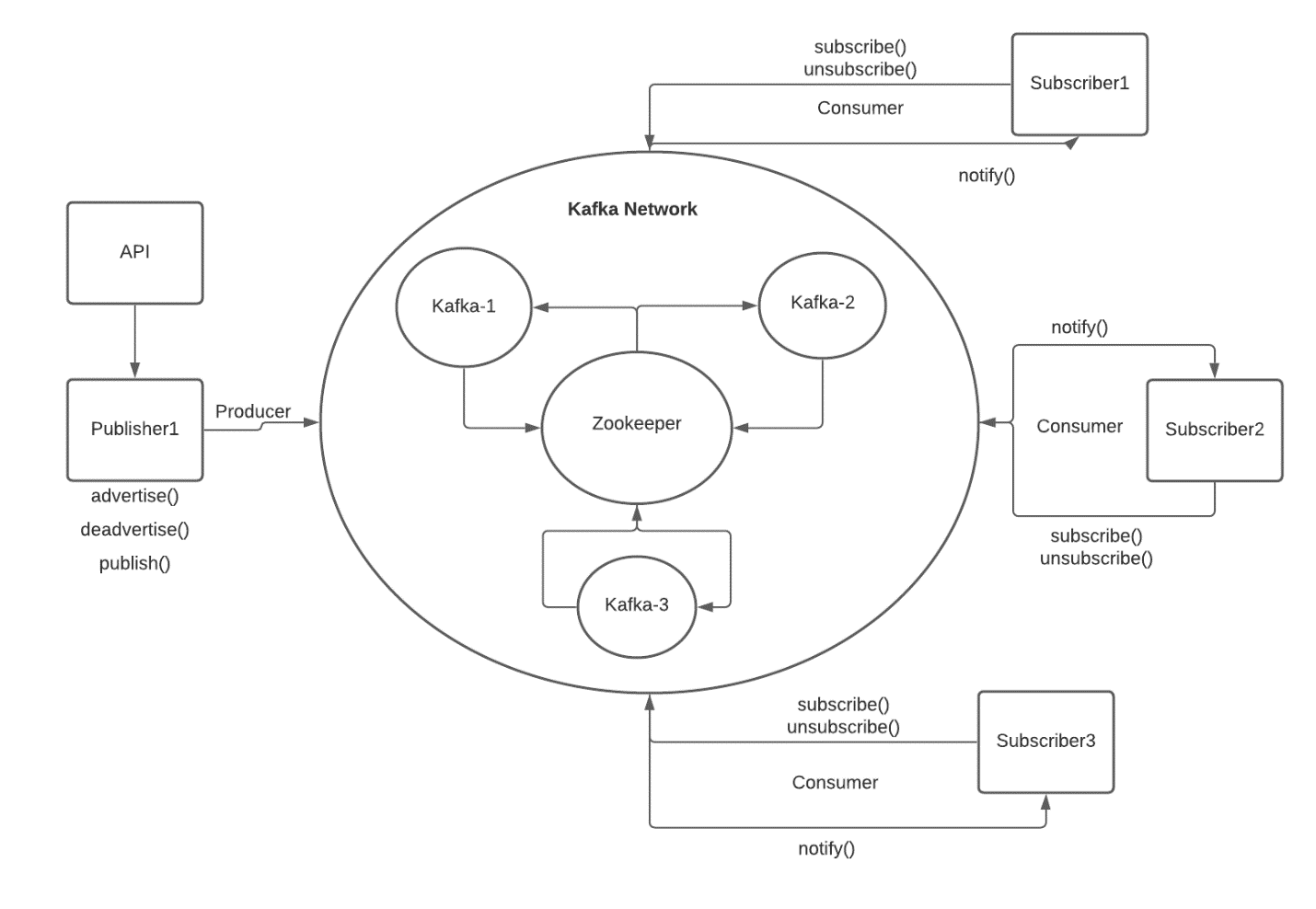
**chaktimw@buffalo.edu**

**Group Number 75**

**API USED: “https://apidocs.cheapshark.com”**

* **Displays games in the Steam Library with their normal prices, current discounts, and more**

**Architectural Diagram:**

****

**Architecture: Sub/Pub System with a simple broker network (topic-based)**

* **Brokers**
  + **Three Kafka brokers are managed by the Zookeeper. This forms the Kafka distributed system which subscribers and publishers can connect to, and exchange data.**
  + **Each broker has three partitions, one for each topic. Data is replicated across brokers.**
* **Publisher**
  + **Publisher container contains a server that retrieves the API data from the link through a get request. The server then creates a Kafka Producer to send data to the Kafka Network.**
  + **Server sends an event automatically every 10 seconds to the Kafka Network. Each partition of the specified topic gets a message.**
  + **Topics are preset ($0-$5, $5-$10, $10-$20)**
* **Subscribers**
  + **Three subscriber containers each has their own server with their own URL (shown in the next section)**
  + **On the subscriber page for each server, a user can subscribe to any of the topics advertised by the publisher. Only advertised topics will be accepted. Once subscribed to a topic, if the publisher publishes a game with a price within the subscribed price topic, the game will display in the newsfeed section along with its price.**
  + **The user will also get all past published events once they subscribe to the topic.**
  + **If the user unsubscribes, these events will remain on their newsfeed, but new events under the unsubscribed topic will not be displayed.**
* **docker-compose**
  + **Creates a container for each broker, publisher, subscriber, and the zookeeper**
  + **Assigns the appropriate dockerfile for each subscriber and publisher**
  + **Connects all the container onto the same docker network**
* **Dockerfiles can be found under dockerfiles in the project directory**
* **Html files are under templates and JS code is under static**
* **Partition management is handled by each broker node and the zookeeper**

**Startup Instructions:**

1. **Run Docker**
2. **Go to the project file directory in the CMD**
3. **Run command “docker-compose up --build”**
4. **Wait until docker finishes starting up each container**
5. **Sometimes, a container may not start properly and crash. Please go to the Docker app and start the container again.**

**Subscriber 1 url:** [**http://localhost:8201**](http://localhost:8201)

**Subscriber 2 url:** [**http://localhost:8202**](http://localhost:8202)

**Subscriber 3 url:** [**http://localhost:8203**](http://localhost:8203)

1. **Open a subscriber URL (each URL is connected to a different container)**
2. **Subscribe to any of the advertised price topics**
3. **Any previously and newly published events that fall under the subscribed topics will be displayed in the Newsfeed**
4. **Each subscriber container has a separate subscription list**