Lab 2 : Microservices at Amazon

Team Members : Sahil Rai, Sarvenaz Mohammadi

Introduction

As microservice architecture is an approach to build applications consisting of multiple short and independent services. Hence enterprises like amazon have adopted microservices to achieve the advantages of scaling, and business agility [1].
The monolith's tightly dependent code made upgrades difficult, time-consuming and expensive. On the other hand, Microservices allowed Amazon to achieve greater business agility, scalability. It also enabled them to make rapid changes and innovations without disrupting the entire system.

Amazon System Architecture

Requirements Considered

1. User can search products
2. User can add products to add to cart
3. User can read/write reviews of the product
4. We need to store pictures, descriptions, reviews, and comments in the database.
5. System should be able to recommend products to users based on their search history and order history.
6. Users should be able to check their order status or previous orders.
7. Users should be able to place the order, using their payment methods.
8. User's credit cards should be stored in a secured database.
9. User should be able to login or sign in
10. Product provider should get a notification that the product is ordered by someone.

Amazon's Main Microservices

1. **Payment Services:** Payment services will be used to process payments for the orders placed. These services will include retrieving credit cards and debit cards previously saved in the user's profile, processing payments for ordered products, and giving users the opportunity to choose the type of payment method they'd like to use. If the payment method is new and has never been used by the user before, the payment services will store that information in the payment database.

2. **Recommendation Services:** Recommendation services will be used to recommend products to users based on their recent search history and their orders. This service will be managed by another child service, which will get the user search history and orders, and based on those child services will add the related products into the Recommended DB. So that the parent service could get those products and show it to the user.

3. **Place Order Services:** Place order service job is, whenever user places an order which was stored in add to cart. Place order service will get the user delivery address, name, phone number, etc. After that it will call the Payment services to get the payment from the user. At last

depending upon the result from payment services, place order services will call Notify Product Provider service.

4. **Search Services:** The job for search service is very easy. Whenever users type anything in the search box to search, store the query in the search database, parallel to that search service will take out the related products from the Product Database, and send it to database services.

5. **Gateway Services:** The point of gateway service is to connect all the services at one place and then using UI display that to the user. This architecture makes sure that users never talk about the services by themselves.

6. **Order Status Service:** Order status service job is to constantly notify the user about their product delivery or show order delivery history. This service also shows users their order receipt, location (if not delivered), when delivered, order date, payment method, Product Provider owner. Once the order is successfully placed, Place order service will save the order in order db and from that order status service will take control of it, to display all the relevant information to the user.

7. **Add to cart services:** Whenever a user wants to store a product for later, they should be able to add that to the cart. The add to cart product structure is very basic, whenever the user clicks add to cart for any product, the product ID of that product is just saved in the database.

8. **User Login/Sign-In services:** Simple service for users to login and sign up.

9. **Notify product provider service:** Whenever the user successfully places an order, the Placed Order service sends a request to the Notify Product Provider service, whose job is to notify the product provider. The notification should include the user's delivery address, user details, and product details.

Advantages and Disadvantages:

Advantages of the Amazon System Architecture:

- Microservices: Microservices make the system more modular and scalable and make it much easier for the developer to maintain the system. It is easier to develop, test and deploy the microservices as a separate service.

- User-Friendly: The Gateway Services provide a user-friendly interface for customers to interact with the system, make the user experience smooth and enjoyable.

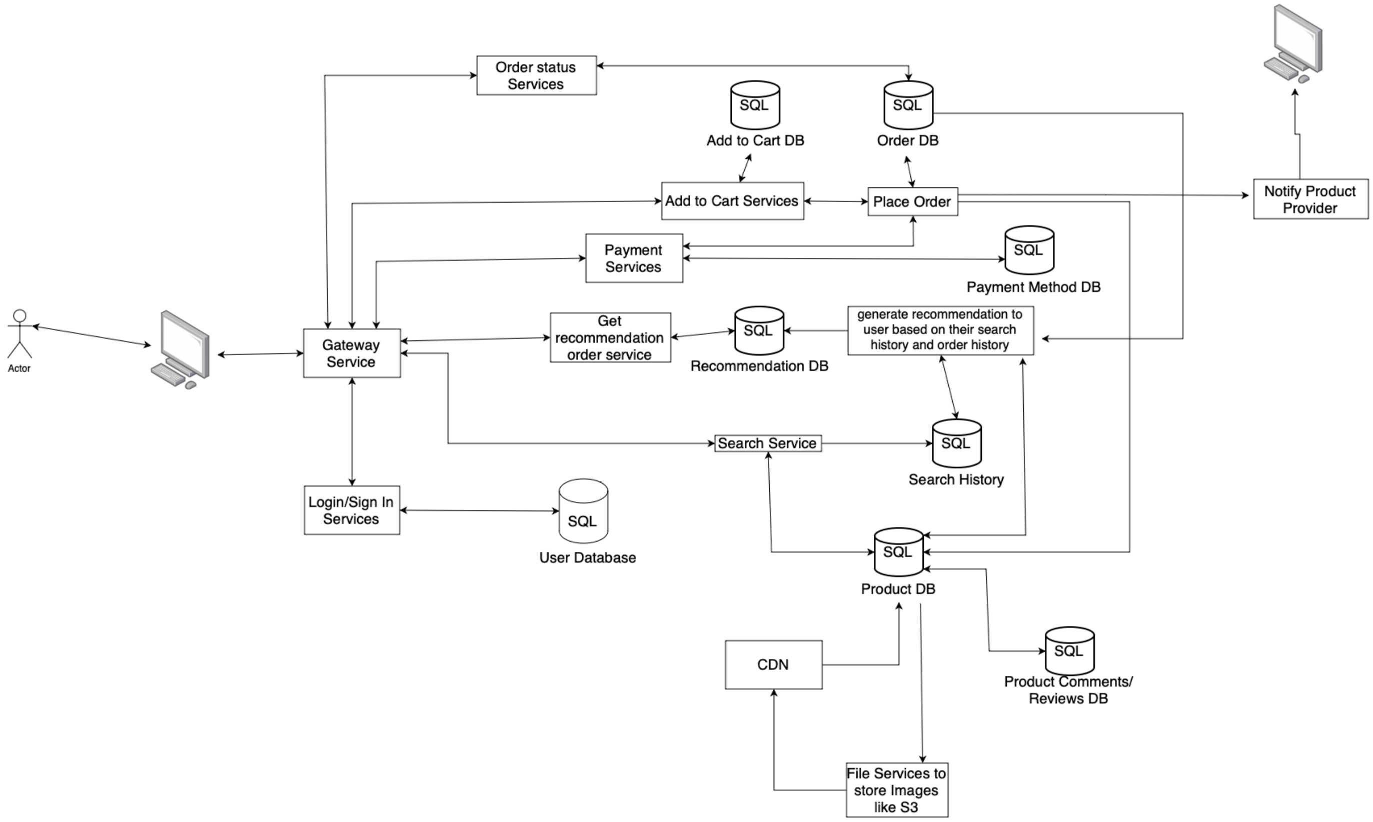
- Personalization: The Recommendation Services use user search history and order history to provide personalized product recommendations. This feature will enhance the user's shopping experience and increase sales.

- Security: Storing credit card information in a secured database helps protect sensitive user data. This architecture includes security measures to protect user information and transactions.

Disadvantages of the Amazon System Architecture:

- Complexity: This architecture is a bit complex considering using multiple microservices and dependencies. Managing and debugging such a system can be challenging.
- Maintenance Overhead: Maintaining a system with many microservices requires lots of effort and can be resource-intensive.
- Potential Latency: When multiple microservices process a user's request, there may be some latency in the system, which could affect user experience.
- Scalability Challenges: While the architecture is designed to be scalable, managing the scaling of individual microservices and maintaining consistency can be complex and time-consuming.

System Architecture Diagram



References

[1] -
https://blog.dreamfactory.com/microservices-examples/#:~:text=Enterprises%20like%20Amazon%2C%20Netflix%2C%20Uber,adapt%20to%20changing%20business%20requirements.