Lab2, Microservice at Amazon
Alex He

Amazon's shift from monolithic architecture to microservice-based system was due to the need of enhanced scalability, speed and reliability. Three of the main core architectures that were switched to microservices were their Search Service, Product Catalog Service, and Shopping Card Service.

**Search Service**
It is responsible for searching for products and this system can be updated independently to improve search results without affecting the rest of the website.

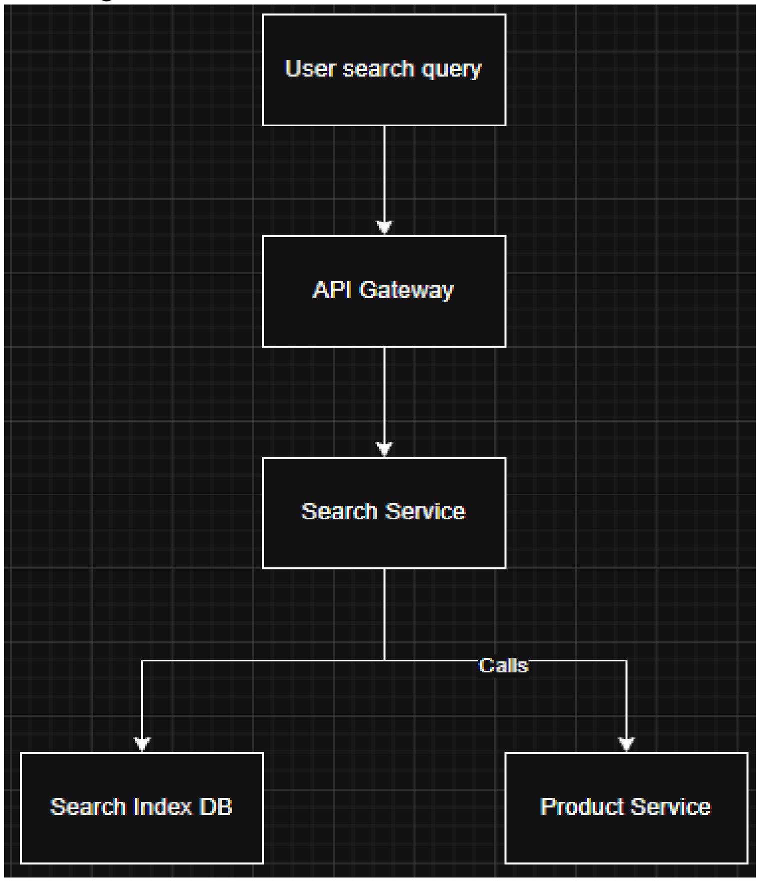


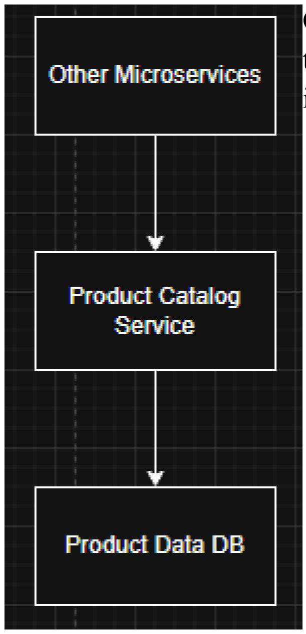
Diagram done by me using draw.io

User search query goes to an API Gateway, which routes it to their Search Service. This service looks up the query inside of its own Search Index Database. Then it could call the Produce Service for more data if needed

| **Pros** | **Cons** |
|---|---|
| • **Independent Scalability**: Search can be scaled independently without needing to scale entire application<br>• **Specialized Technology**: Use specialized search algorithms and databases; Faster more relevant results | • **Increase Complexity:** This service must communicate with other services over a network; Add latency; Robust error handling required |

**Product Catalog Service**

Service that access database to get the products

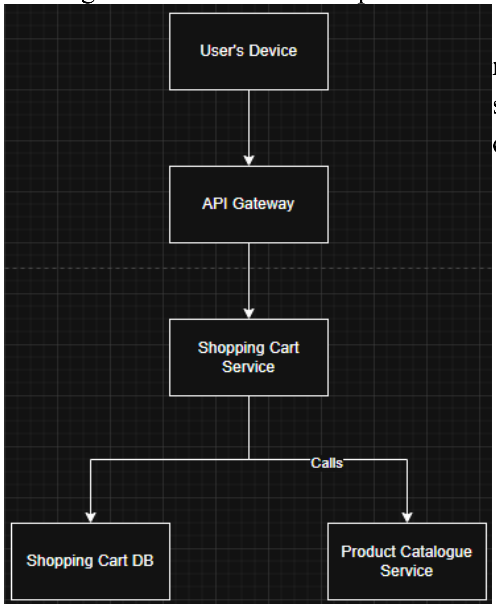


Other services would not store any product information instead they will call this service to retrieve the correct updated data. This service would get the information from a dedicated database

| Pros | Cons |
|---|---|
| • **Reliable Data Source:** It ensures all parts of the site get the exact same, up-to-date product information, eliminating data inconsistencies.<br>• **Fault Isolation:** If a bug affects a different service, product data is unaffected | • **Network Latency:** Data that could be accessed instantly now need to be gotten over a network call which adds small delays |

**Shopping Cart Service**

Manages all items a user has placed in their cart. Critical component



When a user adds an item to their cart, the device sends the request to an API Gateway which passes it to the Shopping Cart service. This service saves them in a Shopping Cart Database and calls Catalogue Service to get the product details

| Pros | Cons |
|---|---|
| **Independent:** The shopping cart remains functional if other parts of the site are down<br>**Focused Development:** Other parts of website are not affected if new features are being developed and deployed | **Distributed Transaction:** Having multiple services creates complex distributed transactions are harder to manage than in-process calls |

[1] AllThingsDistributed. (2020). *Amazon's Monolithic Architecture to Microservices*. Retrieved from https://www.allthingsdistributed.com/2020/09/from-monolith-to-microservices.html

[2] AWS. (n.d.). *Monolith to Microservices: A Journey to Scalability*. Retrieved from https://aws.amazon.com/microservices/

[3] TechTarget. (n.d.). *Microservices at Amazon*. Retrieved from https://www.techtarget.com/searchitoperations/definition/Amazon-Web-Services-microservices

[4] Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media.

Gemini helped with gathering and summarizing resources