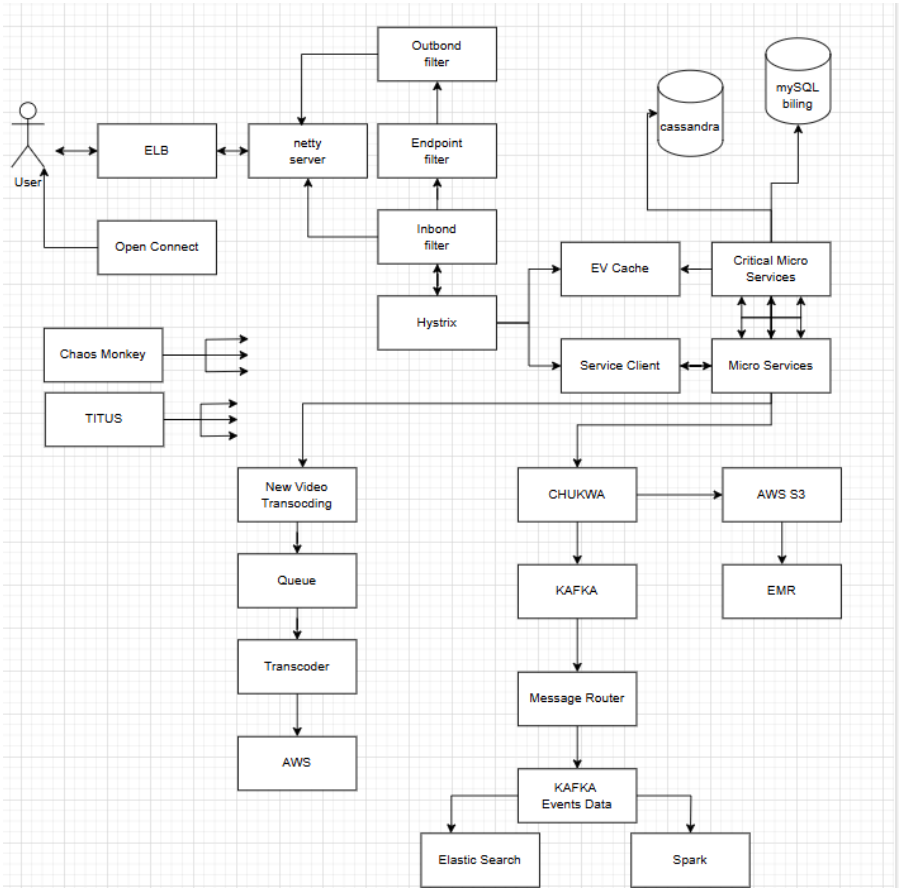
Lab2, Microservices at Netflix

Clement Quach



Open Connect - a Global CDN, serves the video to a customer
ELB - routing traffic to front end. Communicate requests and responses with netty
Netty- a routing service to inbound filter
Inbound- responsible for authentication, send to endpoint
Endpoint- sends a static response, and forwards to the backend.
Outbound - used to zip the content and calculate metrics. The response is sent to Netty to be sent to the client
Hystrix- latency and fault tolerance logic to the database
EV Cache- Several copies of a cache are shared across a network. When a request to read is made, the cache sends it to the requester, and when a write is made, all nodes will be written to.
Critical micro services- Services that route and make calls to the Database
Microservices- like traceback and Error Logs. this connects to Chukwa and Kafka
Chukwe- an open-source data collection system to display and monitor events from the system. It writes in the Hadoop file format and either goes to Kafka or an S3/EMR
Kafka- routing service to the S3, Spark and Elasticsearch
Elastic Search- Service to search logs such as error or issue logs
Spark- Service responsible for giving recommendations
MySQL is used for billing info. A master database replicates itself into read databases. Giving high levels of access and scalability
Cassandra- A NoSQL database used for storing user viewing history.
New Video Transcoding- after the play button is pressed, validate the transcoding
Queue- a Queue for the requests and responses from the AWS server
Transcoder- takes video from an AWS server and converts it to the proper format
AWS server- Worker threads store and serve data
Chaos Monkey- Tests the services for faults

TITUS- an isolated network for tests
Pros
-services are stateless, meaning the same input will always give the same output
-bugs become isolated, because all services are independent, one should not affect the other
-lots of scalability, new features to handle large amounts of new users can be made easier
-failures in the system can be routed

Cons
-More complicated to develop, the more services the harder the entire network is to comprehend
-need for automated deployments
-increased costs, the more service you run the more resources they will consume
-decreased performance, with so many calls to other applications and servers, optimization becomes mandatory

Citations
https://roshancloudarchitect.me/understanding-netflixs-microservices-architecture-a-cloud-architect-s-perspective-5c345f0a70af

https://www.youtube.com/watch?v=57UK46qfBLY

https://medium.com/swlh/a-design-analysis-of-cloud-based-microservices-architecture-at-netflix-98836b2da45f

https://www.geeksforgeeks.org/system-design-netflix-a-complete-architecture/
ChatGPT was used to find information