Netflix’s scalable and high availability system
+ Adrian Goh Jun Wei + SSP3 + jgoh070@e.ntu.edu.sg

Description

Netflix is a leading video-on-demand subscription service with 110 million subscribers in over 200 countries. To grow its annual revenue of more than $10 billion, it has to provide fast and reliable streaming services. Thus, the architecture is designed with 3 objectives: highly scalable, available and cost-efficient.

The architecture can be described as having a simultaneously heterogeneous style, as it is a combination of Layered and Microservices. Main source of information from http://highscalability.com/, https://netflixtechblog.com/ and https://www.infoq.com/.

It consists of 4 main layers.

1. **Presentation Layer:** Front-end client-side code that displays static or dynamically generated content on the user interfaces (sign-up page, home pages, video streaming). It is a layer which users can access directly, and communicate with other layers through API calls.
2. **Orchestration Layer:** Netflix uses its in-house tool Zuul for this layer (also known as API gateway). It is the front door for all requests from devices and web sites to the backend of the Netflix streaming application.
3. **Application Layer:** Business rules are removed from the client and executed in this middle dynamic content processing and generation level application server. The Java APIs perform their content gathering functions, ensure business rules are processed correctly and deliver the requested content back to the client.
4. **Data Access Layer:** This layer provides access to the data through methods without exposing the data storage mechanisms. One of the main advantages is that the business logic are not dependent on the data sources and protocols, which also means supporting the swapping of data sources without a significant impact or major code rewrites to the databases.

Netflix also has over 1,000 services running across multiple secure and scalable EC2 instances on Amazon Web Services (AWS). Every service is running its process and communicating with lightweight mechanisms, with a database and storage solution that is best suited for the service.

Data storage also uses distributed and scalable databases like DynamoDB and Cassandra. This means that the databases run on multiple cloud servers, and there will be no data loss should some servers experience down-time. Besides, with Netflix being a data-driven company and growing at a very fast pace, the database can handle as much data as you ever want to put into it by adding more computers.

Special techniques or solutions used

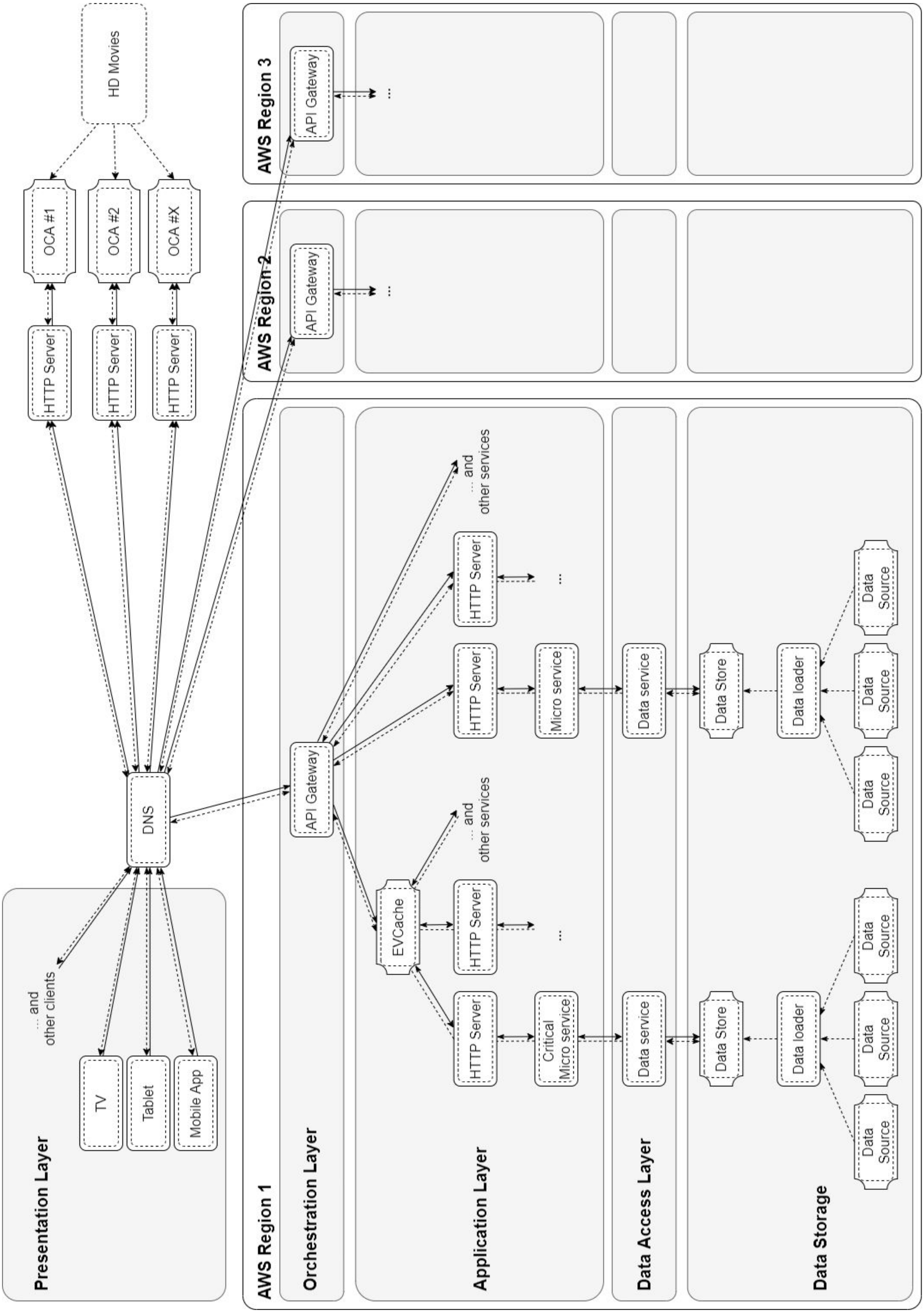
1. **Caching frequently used data with EVCache:** Critical microservices are defined to have basic functionality to work. Netflix cached these frequently used data using a self-developed distributed caching solution called EVCache, and some advantages include:
    a. Faster response time as compared to data being fetched from source/database
    b. Cost-saving by reducing the number of servers needed to handle the requests as most of the requests are served by the cache
    c. Increases the throughput of the services fronted by the cache

2. **Improved reliability by hosting services across multiple regions:** Netflix operates out of 3 AWS regions, and 3 different availability zones within each region. The main advantages of having multiple regions are:
    a. Improvement in the application's reliability because if any region fails, and the other regions will step in to handle all the members in the failed region.
    b. It gives Netflix worldwide coverage with minimal compromisation on streaming quality.
3. **Maximise network efficiency with Open Connect (CDN)**: Video distribution is a core competency for Netflix and thus, they developed Open Connect, its purpose-built CDN, as well as Open Connect Appliances (OCA), a server for video storage that is optimised for delivering large files. OCAs are placed in more than 1,000 locations around the world. Netflix then proceeds to use popularity data to predict which video will have a higher chance of being viewed the following day in each location and copies these predicted videos to one or more OCAs at each location. With these, Netflix benefits from:
    a. Cost-saving as 3rd-party CDNs are more expensive
    b. Improve streaming quality as videos are already stored in OCAs before users request for it.
    c. Reduced cost by only storing videos that will most likely be viewed in the OCAs, instead of wasting unnecessary storage space by storing their entire catalog.
4. **Implementing Data Service and Data Loader:** A Data Monolith means that multiple data sources are loaded in memory even though only a very small percentage of data is being accessed. With the Data Service, the Data Loader only loads whatever data it needs into the Data Store.

Quality attributes addressed

1. **Efficiency**: As caching solutions like EVCache are being used to serve a large portion of frontend requests, the number of requests to the server are largely reduced. This reduced the number of servers required to have a high performing system. Also, since Netflix forecasts which video will most likely be viewed the next day, there's always a one day lead time before a video is required to be on an OCA. This means videos can be copied during quiet, off-peak hours, substantially reducing bandwidth usage. Lastly, Data Service and Data Loader reduces CPU and memory utilization by only loading required data into the Data Store.
2. **Flexibility**: In the event that a new client application will be developed (e.g. new gaming console), Netflix can easily add them into the Presentation Layer without affecting the other layers or even other components in the same layer.
3. **Availability**: By predicting which videos have a higher chance of being viewed and pre-storing them in the OCAs, cache misses (asking for a specific video from an OCA and the OCA saying it doesn't have it) are eliminated since requested videos are always near the users.
4. **Robustness**: With the usage of multiple AWS regions and zones, availability is improved as it requires the systems in all regions and zones to be malfunctioning before the application experiences down-time.
5. **Maintainability**: The Data Access Layer removed dependencies on data sources and protocols from the Application Layer, allowing the team to replace the data sources at any time without affecting the business logics.
6. **Reliability**: Netflix develops Android and iOS apps themselves, and control the software development kit (SDK) for platforms where they don't build the client (e.g. Smart TV). By having control over the applications, Netflix can adapt consistently and transparently to slow networks, failed OCAs, and any other problems that might arise.

Architecture diagram


HD Movies
OCA #1
OCA #2
OCA #X
HTTP Server
HTTP Server
HTTP Server
AWS Region 3
API Gateway
...
AWS Region 2
API Gateway
...
Presentation Layer
TV
Tablet
Mobile App
... and other clients
DNS
AWS Region 1
Orchestration Layer
API Gateway
Application Layer
EVCache
HTTP Server
HTTP Server
Critical Micro service
... and other services
HTTP Server
Micro service
HTTP Server
...
...
... and other services
Data Access Layer
Data service
Data service
Data Storage
Data Store
Data loader
Data Source
Data Source
Data Source
Data Store
Data loader
Data Source
Data Source
Data Source