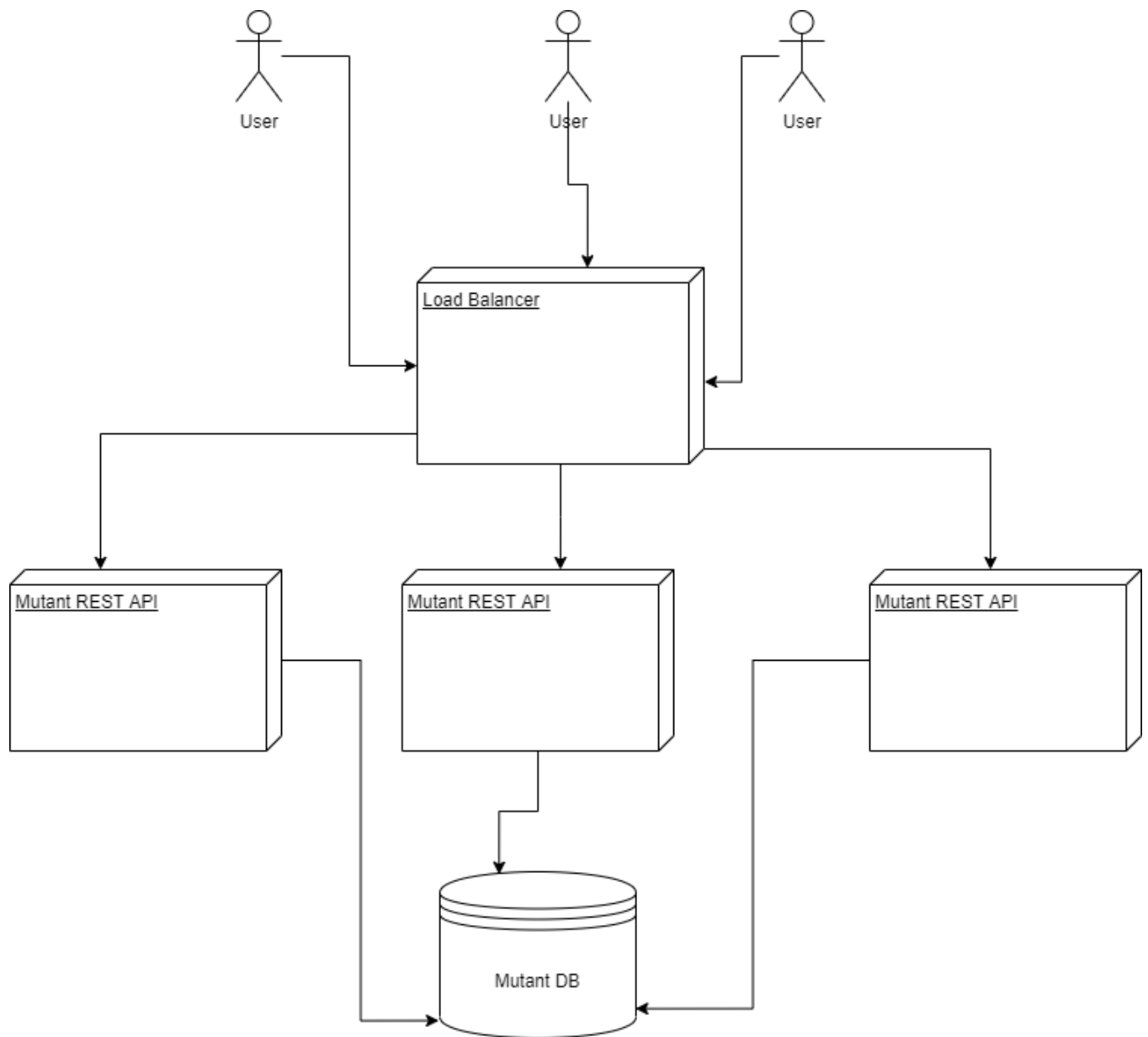


Mutant App

Arquitectura propuesta



Para el desarrollo de esta aplicación se propone usar los diferentes servicios provistos por las nubes más comunes actualmente.

Específicamente, se propone el uso de AWS Auto Scaling que permite definir distintas métricas con las cuales se tomará la decisión de aumentar la capacidad (desplegar más servidores) o de decrementar las máquinas activas. Junto a este servicio, se propone el uso de un balanceador de carga que distribuirá los requests de manera tal que cada servidor tenga un volumen de procesamiento razonable que permita que las peticiones sean procesadas sin que el usuario experimente algún tipo de degradación en el servicio.

Para que lo anterior sea posible, es recomendable manejar una arquitectura de microservicios donde cada parte del negocio tenga definida de manera específica el microservicio al que pertenece.

Para este caso, se considera el validador de mutantes y las estadísticas como un microservicio, el cuál (al ser stateless -requerimiento importante para que cualquier servidor pueda atender cualquier request-) podrá escalar de manera rápida para soportar la carga intensiva de peticiones en cualquier momento.

Así mismo, es necesario contar con una base de datos cuyos recursos estén bien definidos según la carga esperada, esto debido a que de quedarse corta en recursos, la base de datos podría convertirse en un cuello de botella que, aún con los servicios anteriores, impedirá un escalamiento exitoso de la solución.

Será conveniente implementar un caché que permita guardar las cadenas más consultadas o recién agregadas para evitar llamados innecesarios a la base de datos y evitar que esta se congestione.

Casos de uso

A continuación, se presentan dos casos de uso para la aplicación:

1. En el primer caso, el usuario envía una cadena de ADN a la aplicación a través de una petición **POST**. La aplicación recibe esta cadena y determina si ésta pertenece a un mutante o no y la guarda en base de datos si no existía previamente. Tras esto, la aplicación dará una respuesta al usuario que dependerá de si la cadena pertenece a un mutante o no.
2. En el segundo caso, el usuario hace una petición **GET** para obtener las estadísticas almacenadas. La aplicación realizará un conteo del ADN humano y mutante que tenga registrado y retornará estos datos, así como el ratio de los mismo en formato JSON al usuario.

