**Resume Key words and concepts**

**Cloud Native** – A cloud native application consists of discrete, reusable components known as microservices that are designed to integrate into any cloud environment. These microservices act as building blocks and are often packaged in containers. Microservices work together as a whole to comprise an application, yet each can be independently scaled, continuously improved, and quickly iterated through automation and orchestration processes. The flexibility of each microservice adds to the agility and continuous improvement of cloud-native applications.

**Apache POI** – A Java library for manipulating various file formats. Provides the ability to read and write MS Excel files using Java.

**Spring Boot** – Open-source Java-based framework used to create micro-services.

Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can just run.

Uses annotations for RESTful API endpoints.

@RequestMapping

@PathVariable

@QueryParam

**JWT (JSON Web Tokens)** – “Portable units of identity” identity information in JSON format and can be passed around to services and applications. Any service can verify a JWT itself. Once a JWT is verified, the service or application can use the data within it to act on behalf of the user.

In simple language there is a secret Key used to encrypt the JSON formatted Data which primarily includes the user-id. Now an encryption of data with the Key generates the token that is sent to the client and used in every request. Every time, the client sends in the request with the token the server tries to decrypt it with the Key, if it can, it gets the user-id from the JSON Data which corresponds to the user.

Can be passed around through the headers or cookies.

JWTs expire at specific intervals

When a JWT is created, it is set to expire at a specific expiration time usually in terms of minutes and not hours. JWT expiration time is not extended due to user activity. Instead, they are programmatically replaced by creating a new JWT for the user.

To solve this problem, most applications use refresh tokens. These are used primarily for generating new JWTs. Refresh Tokens also need to expire at some point, but they are more flexible in this because they are persisted in the identity provider.

JWTs are signed

JWTs are cryptographically signed which requires a cryptographic algorithm to verify. These algorithms are purposely designed to be slow. The slower the algorithm, the higher the complexity, and the less likely the algorithm can be creaked using brute force.

If the algorithm used is HMAC to sign the JWT, then all services that want to verify must have the HMAC secret.

JWTs have exploits

This is more a matter of bad coding than flaws that are inherent to JWTs. The “none” algorithm and the “HMAC” hack are both well know exploits of JWTs.

Both exploits have simple fixes. Specifically, you should never allow JWTs that were created using the “none” algorithm. Also, you should not blindly load signing keys using the “kid” header in the JWT. Instead, you should validate that the key is indeed the correct key for the algorithm specified in the header.

Hackers can compromise the secret key through Man in the middle attacks.

Data overhead

The size of the JWT token will be more than that of a normal Session token. The more data you add in the JWT token, the longer it gets linearly. Remember, each request needs the token in it for request verification. So, say, a 1 KB JWT token implies each request will have 1KB over-head upload which is bad in cases of low-speed net connectivity.

**React Redux** – State management tool for React.

Reasons why I am not a big fan of Redux.

Lots of jargon.

Lots of boilerplate code.

Really good with state management at scale.

**Hibernate** – A Java framework that simplifies the development of Java applications that interact with a database.

Lightweight

Open source

ORM (Object Relation Mapping)

Uses JPA (Java Persistence API) for data persistence.

ORM is a technique that maps the object to the data stored in the database.

The ORM tool internally uses JDBC API to interact with the database.

JPA is a Java specification that provides certain functionality and standard ORM tools.

Advantages of Hibernate

Open source and light weight

Fast performance – Utilizes a cache. There are two types of cache: 1st level cache and 2nd level cache. 1st level cache is used by default.

Database Independent Query – HQL (Hibernate Query Language) is object-oriented version of SQL. It generates the database independent queries, so you don’t have to. Without Hibernate, if the DB changes, we would need to change the SQL query which leads to a maintenance problem.

Automatic Table Creation – Hibernate provides the ability to create tables of the database automatically. No need to create tables in the database manually.

Simplified Complex Join – Fetching data from multiple tables is easy

Provides Query Statistics and Database Status – Supports Query cache and provides statistics about query and database status.