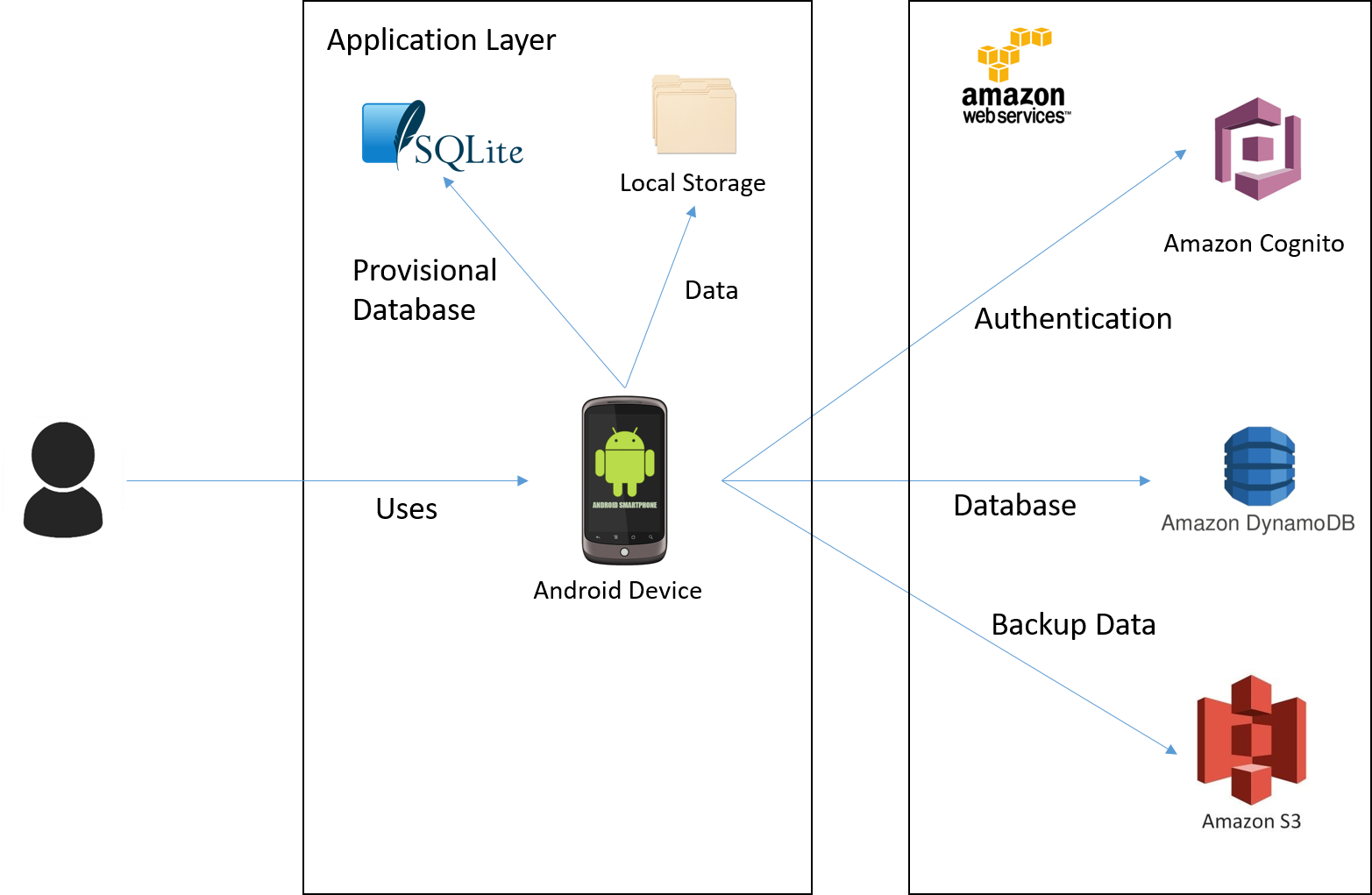
**System Architecture**

The architecture adopted by the application will be a new and sophisticated cloud-client architecture. This architecture works by making use of cloud computing resources to manage user’s data and credentials as well as authenticating user. The proposed application will be using AWS (Amazon Web Services) as its cloud service provider and AWS Mobile SDK [1] for the development of the app. The main advantage this architecture has is that we don’t need to manage the resources in the cloud including the security of it.

**Overview of the architecture:**



**Components of the architecture:**

**Amazon Cognito**

Amazon cognito is Amazon Cognito makes it easy to add user sign-up and sign-in and manage permissions for mobile apps. We can create our own user directory within Amazon Cognito, or qw can authenticate users through social identity providers such as Facebook, Twitter, or Amazon; with SAML identity solutions; or by using our own identity system. With Amazon Cognito, we can focus on creating great app experiences instead of worrying about building, securing, and scaling a solution to handle user management and authentication [2].

**Amazon DynamoDB**

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models. Its flexible data model, reliable performance, and automatic scaling of throughput capacity, makes it a great fit for mobile, web, gaming, ad tech, IoT, and many other applications [3].

**Amazon S3**

Amazon Simple Storage Service (Amazon S3) makes it simple and practical to collect, store, and analyze data - regardless of format – all at massive scale. S3 is object storage built to store and retrieve any amount of data from anywhere – web sites and mobile apps, corporate applications, and data from IoT sensors or devices. It is designed to deliver 99.999999999% durability, and has many customers each storing billions of objects and exabytes of data. You can use it for media storage and distribution, as the “data lake” for big data analytics, as a backup target, and as the storage tier for serverless computing applications. It is ideal for capturing data like mobile device photos and videos, mobile and other device backups, machine backups, machine-generated log files, IoT sensor streams, and high-resolution images, and making it available for machine learning to other AWS services and third party applications for analysis, trending, visualization, and other processing [4].

**SQLite**

SQLite is an embedded SQL database engine that can act as a database in android local system. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format.

Other architecture that have been considered are standalone, client-server, and web-based. Standalone is simpler and more robust, it mostly doesn’t need any internet connection to use all the features in the app, but the number of feature it can have is limited. Client-server opens more possibilities for the application to have more interesting features, however there would be a problem of securing the communication, the cost to set up the server, and the maintenance of the server itself. All of this can be applied to web-based architecture too since it uses a server. Additionally, web-based is less desirable due to its performance issue because there is no local storage for the application, therefore the performance is very dependent on network connection to access the data.

**Design principles of the application:**

**Modularity:** separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality.

**High Cohesion:** each module have functions and elements that are strongly related, only to fulfill one particular purpose or task

**Low Coupling:** modules are loosely coupled and independent so that a change in one module do not affect the other modules

**Standardization:** implementation will conform to standard that has been established and agreed by different parties, this is crucial for things like security

**The proposed application will have the following quality:**

**Extensibility**

The proposed application can add additional functionality without changing or damaging much of the current system. New data types can be added as long as it is supported by the android

**Maintainability**

Following the design principles of high cohesion and low coupling, small modifications will not be a problem. Changing one module will not affect other modules significantly.

**Performance**

The response time will be in acceptable manner even with the huge amount of data that are processed. Efficient encryption algorithm are used as well as other processing algorithm

**Usability**

Adapting KISS (Keep It Simple Stupid) principle in designing interfaces will give user easier times in learning and figuring out the proposed application. It lets user to take less time to perform a certain task.

**Compatibility**

The proposed application will be able to run in various type of android devices as well as different version of android.

**Security**

Data are kept safe by encryption and login is required to have access. Security measures like protection against sql injection or encryption algorithm will follow standard.

# References

|  |  |
| --- | --- |
| [1] | Amazon, "AWS Mobile SDK," Amazon, [Online]. Available: https://aws.amazon.com/mobile/sdk/. |
| [2] | Amazon, "What is Amazon Cognito?," Amazon, [Online]. Available: http://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html. |
| [3] | Amazon, "Amazon DynamoDB," Amazon, [Online]. Available: https://aws.amazon.com/dynamodb/. |
| [4] | Amazon, "Amazon S3," Amazon, [Online]. Available: https://aws.amazon.com/s3/?sc\_channel=PS&sc\_campaign=acquisition\_SG&sc\_publisher=google&sc\_medium=s3\_b&sc\_content=s3\_e&sc\_detail=amazon%20s3&sc\_category=s3&sc\_segment=82556503449&sc\_matchtype=e&sc\_country=SG&s\_kwcid=AL!4422!3!82556503449!e!!g!!amazon%20s3. |