Department of Information Engineering, CUHK
MScIE – 2nd Semester, 2015/16

IEMS 5722
Mobile Network Programming and Distributed Server Architecture

Lecture 12
Advanced Android Programming

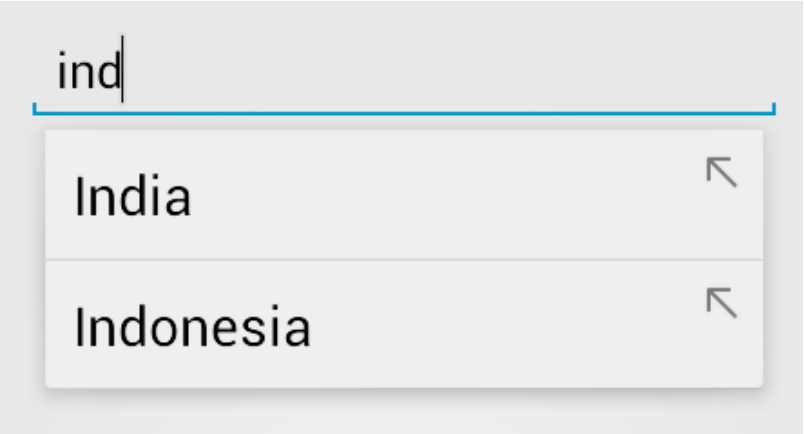
Lecturer: Albert C. M. Au Yeung

31st March, 2016

Auto-complete Text Views

Auto-complete Text Views

- A text view that automatically suggests words or phrases for input while the user is typing
- Useful when you allow the user to perform searching and input specific information (e.g. country names, addresses, etc.)
- You can use one of the following classes depending on your requirements
  - **`AutoCompleteTextView`**
  - **`MultiAutoCompleteTextView`**



Auto-complete Text Views

- Adding AutoCompleteTextView to your Activity

```
<AutoCompleteTextView
    android:id="@+id/ac1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content“/>
```

Auto-complete Text Views

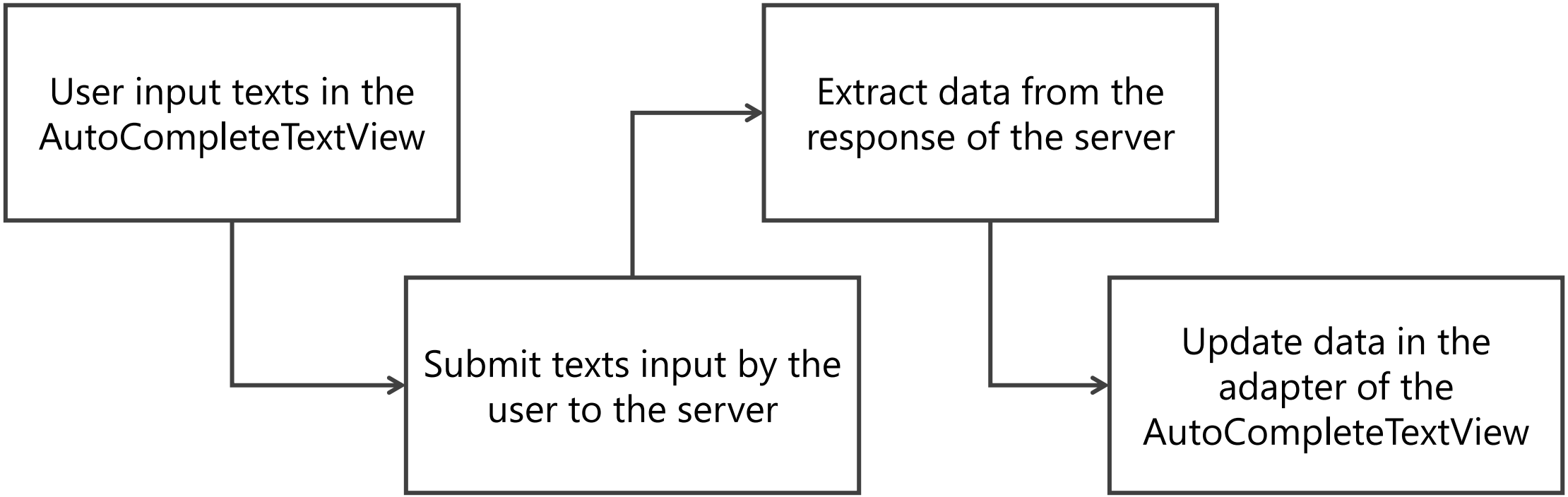
- Auto-complete involves providing a candidate list, this is done by using an adapter (similar to what you do when using ListView)
- For example, the following use a pre-defined string array as a candidate list

```
AutoCompleteTextView ac = (AutoCompleteTextView)
findViewById(R.id.autoCompleteTextView1);

String[] countries = new String[] {
    “Algeria”, “Belgium”, “Canada”, “Denmark”, “Ethiopia”, “France”
};
ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, countries);
ac.setAdapter(adapter);
```

Auto-complete Text Views

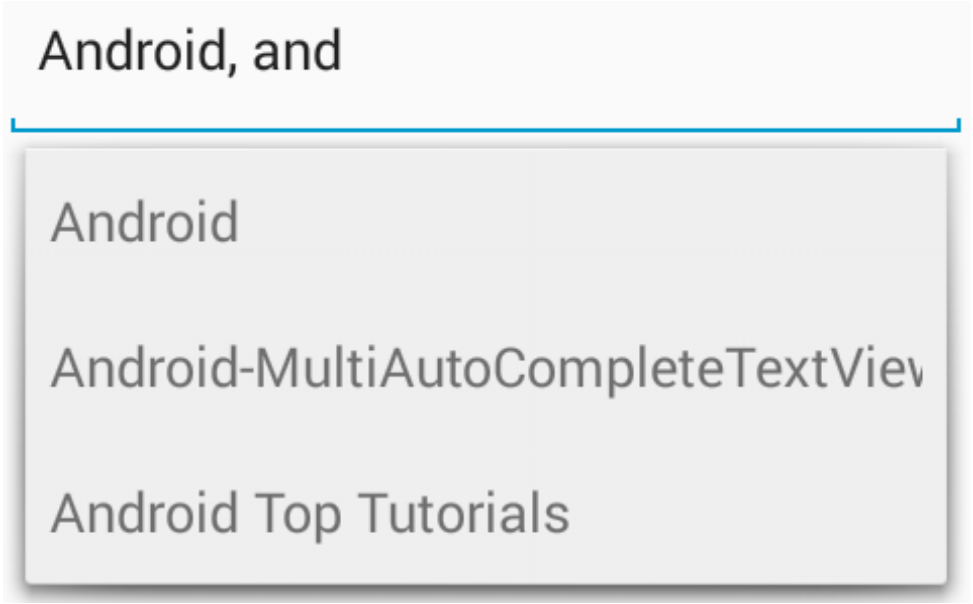
- What if you don’t want to hard-code the list of candidates in the app?
- You need to send the intermediate input of the user to the server, and use the data returned by the server to update the data in the adapter



```
ac.addTextChangedListener(new TextWatcher() {
    @Override
    public void afterTextChanged(Editable s) {
        final String query = s.toString().trim();
        if (query.length() > 1) {
            AsyncHttpClient client = new AsyncHttpClient();
            RequestParams params = new RequestParams();
            params.put("query", query);
            client.get(url, parmas, new TextHttpResponseHandler() {
                @Override
                public void onSuccess(
                    int statusCode, Header[] headers, String response) {
                    // Extract data ...
                    // Update data of the adapter ...
                    adapter.notifyDataSetChanged();
                }
            });
        }
    }
});
```

MultiAutoCompleteTextView

- MultiAutoCompleteTextView allows user to input multiple items separated by a separator (e.g. a comma)
- Suggestions will be provided for the latest item in the text view



Authentication & Authorization

User Authentication

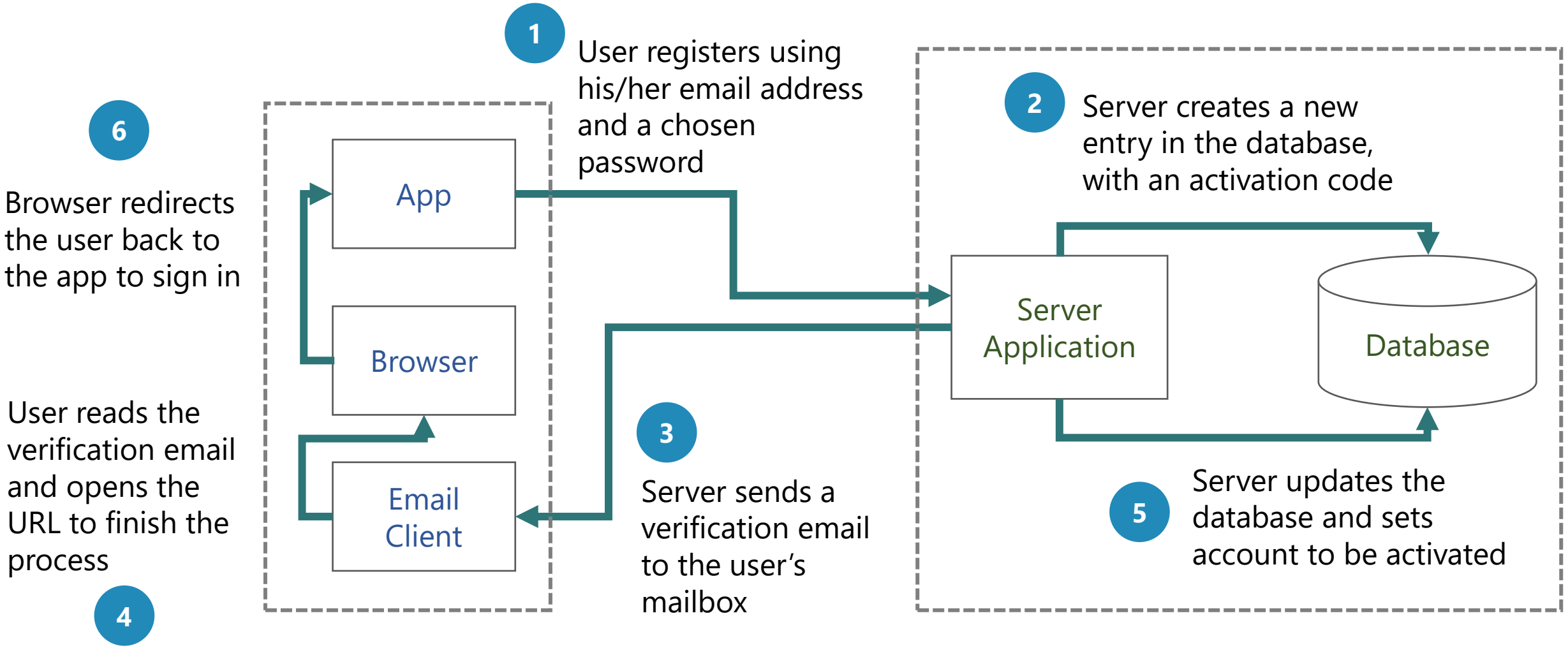
- It is very likely that you would like your users to create an account in your app, and sign in to use its services
- Reasons:
  - It is necessary: if your app needs to uniquely identify every user
  - Track user's usage of the app
  - Allow users to retrieve their data on different devices
  - Present more personalised information and services
  - ...

User Authentication

- How should you implement your system to enable user authentication in your app?
- Basic functions
  - Register an account by email (or phone number)
  - Validate user's email address by sending him/her an activation email
  - Sign in using email (or user name) and chosen password
  - Authenticate the user whenever the app makes API requests

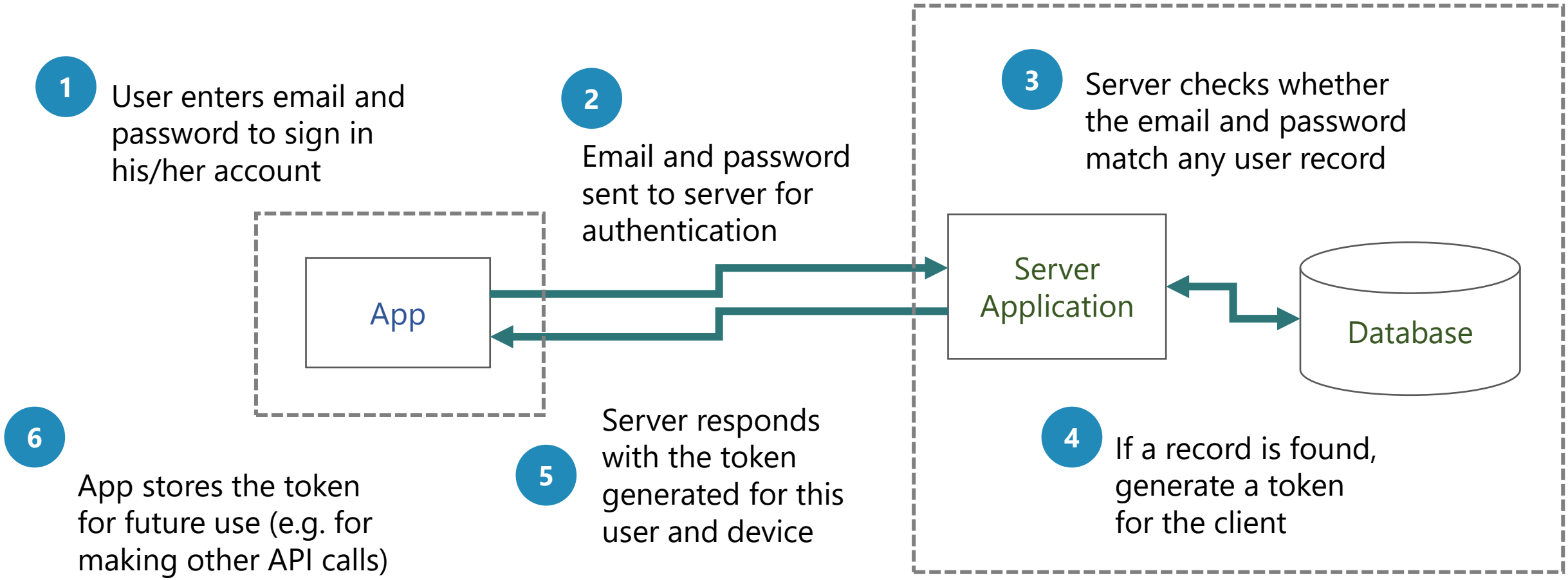
Registration

- A typical registration process:



Sign In

- A typical sign-in process:



User Authentication

Best practices for privacy and security concerns

- Do not store the password clear text in your database, hash it with a salt before sending it out from the app (e.g. using MD5 or SHA1, or more secure ones)
- Do not store user password in the device
- Use HTTPS whenever possible
- Validate user's input (e.g. email and password), before sending them to the server

References:
http://developer.android.com/training/articles/security-tips.html

Authorization

- When using third party libraries, SDKs or APIs in your app, usually it requires the user to authorize your app to use data in another application
- Example:
  - Retrieve your friend list from Facebook
  - Retrieve your name and email address from Google
  - Retrieve your posts from Twitter or Weibo

Authorization

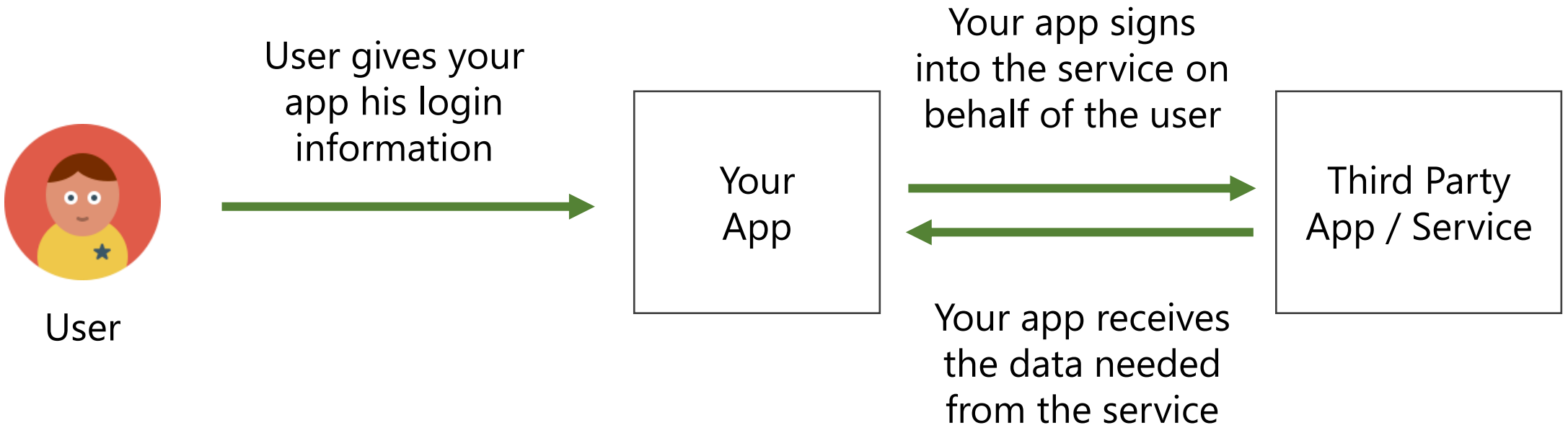
- For most Internet services, SDKs for Android and iOS are available for integrating sign-in process in your app:
  - Facebook
    https://developers.facebook.com/products/login
  - Twitter
    https://dev.twitter.com/twitter-kit/android/twitter-login
  - Google+
    https://developers.google.com/+/mobile/android/sign-in
  - Wechat
    https://open.weixin.qq.com/

Authorization

- For asking for authorization to use data in another app, you should always use their latest SDK whenever possible
- These SDKs wrapped the process of asking authorization from the user
- Most of services perform authorization using OAuth2 (Open Authorization Version 2)
- It is not likely that you will have to implement the flow by yourself, but let's still take a look at it

Open Authorization (OAuth)

- To access a user's account and the data within, you need to have the user's username and password
- Does the following make sense?



Open Authorization (OAuth)

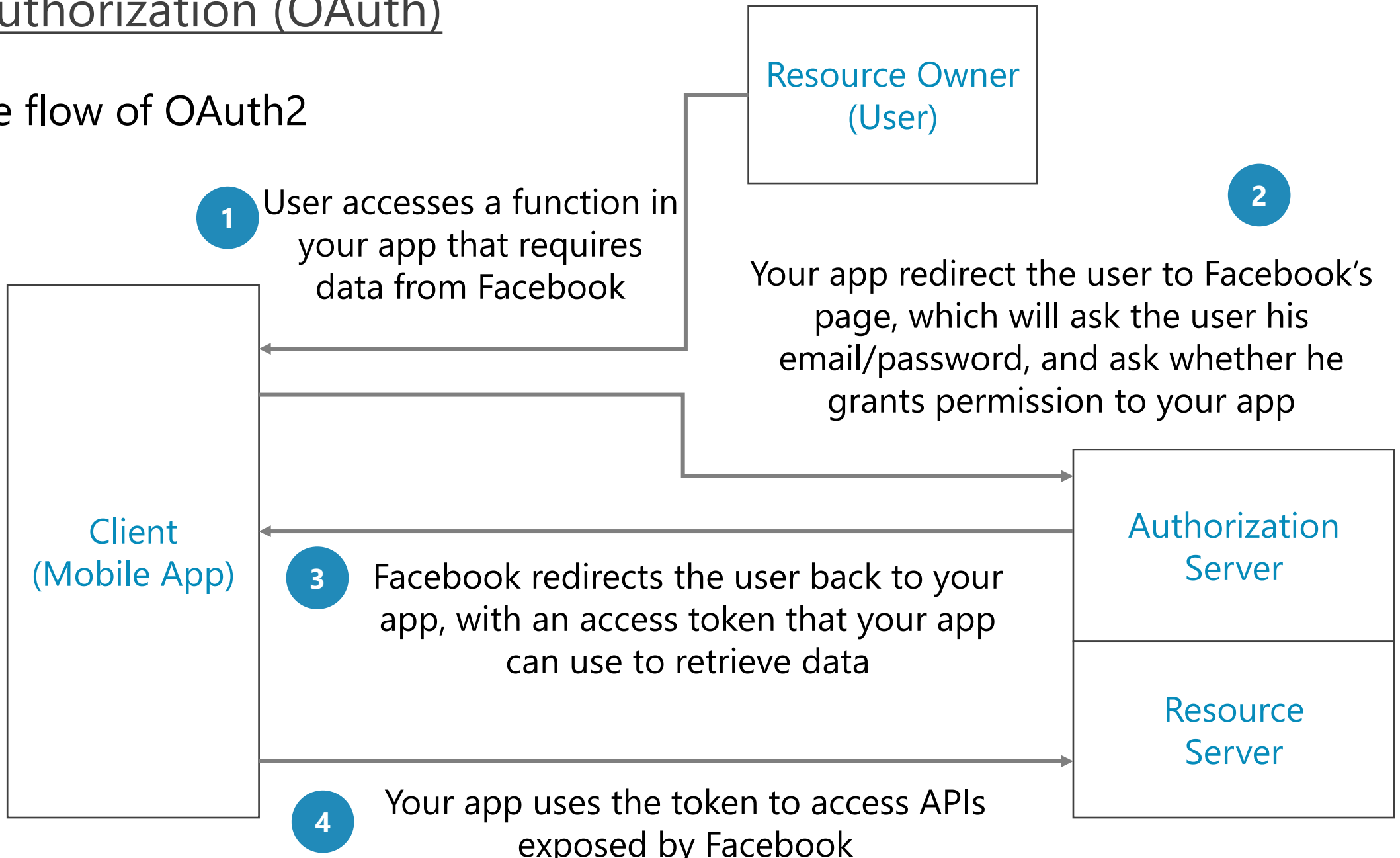
- Never ask users to “give away” their password
- OAuth allows you to get access to user’s data in another service without the need to ask the user for his user account
- Let’s consider a scenario:
  - You have developed a social app, and you want to access the user’s friend list in Facebook, so that you can build up the social network in your app quickly

Open Authorization (OAuth)

- In OAuth, we have three entities:
  - Resource Owner
    This is the user who is using your app
  - Resource and Authorization Server
    In our case this is Facebook's server, which will authorize your app and serve the user's data to your app
  - Client
    Your app, the application that the user is using

Open Authorization (OAuth)

- The flow of OAuth2



Gradle

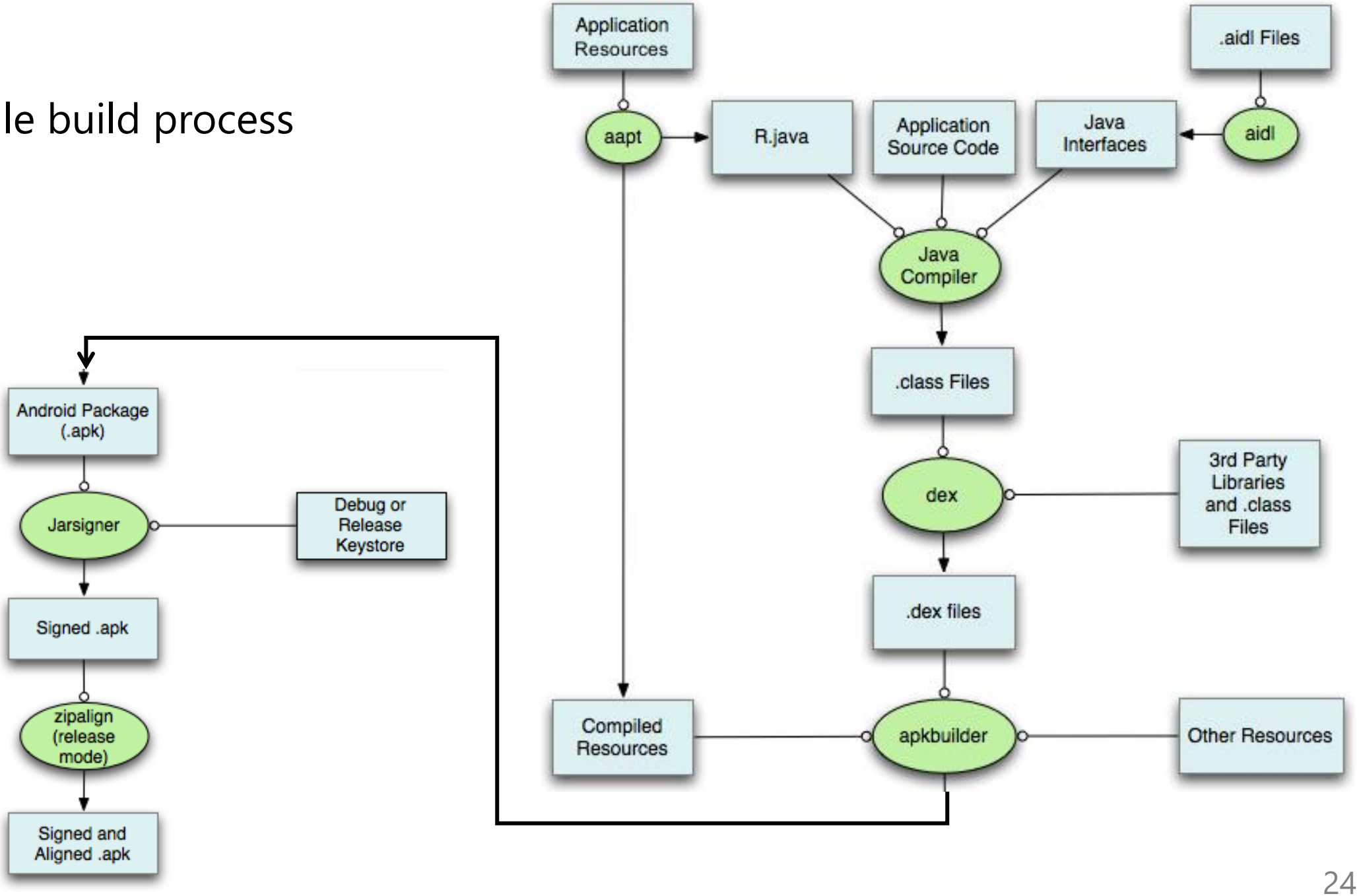
Gradle

- Gradle is the official build tool for Android applications
- It helps you build (compile), test, run and package your apps
- It is integrated into Android Studio through the Android Gradle Plugin
- It can also be executed independently from the command line

Reference: http://gradle.org/

Gradle

- The Gradle build process

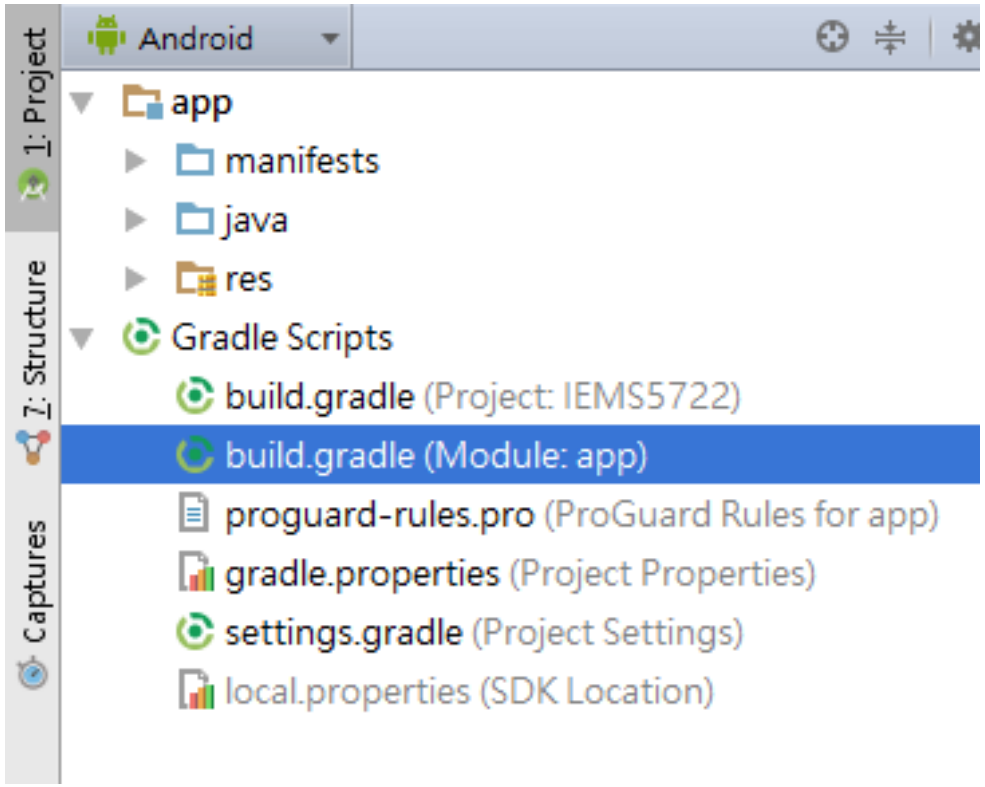


What are the benefits?

- Using Gradle allows you to
  - Configure and customise the build process
  - Create different versions of your app with different features, settings or parameters under the same project
  - Easily incorporate third-party modules into your application

Build Configuration

- An Android Studio project contain a top-level build file and a build file for each module (‘app’ is a module in the project)
- The build files are all named ‘`build.gradle`’
- In most cases, you only need to modify the build files at the module level



Reference: http://developer.android.com/tools/building/configuring-gradle.html

Build Configuration

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.2"

    defaultConfig {
        applicationId "hk.edu.cuhk.ie.iems5722"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:23.1.1'
}
```

Apply the Android plugin for Gradle in this build

Configure the core settings and entries in the AndroidManifest.xml file

Specify different build types. The two default build types are “debug” and “release”.

Declare dependencies to libraries, jar files, remote packages

Product Flavours

- When publishing an app, you might want to publish different variants of the app, which features different functions
- You can specify different "product flavours" in the **build.gradle** file, and specify their individual parameters

```
 productFlavors {
    pro {
        applicationId = "com.iems5722.pro"
    }
    free {
        applicationId = "com.iems5722.free"
    }
}

buildTypes {
    debug {
        applicationIdSuffix ".debug"
    }
}
```

Using BuildConfig

- You can specify constant values in the `build.gradle` file, and use them in your Java code
- When building, different build types or product flavours will use the corresponding values automatically
- Steps:
  1. Specify constant values in `build.gradle` using `buildConfigField`
  2. Use the values in your Java code using the `BuildConfig` class

Using BuildConfig

```
android {
    buildTypes {
        debug {
            buildConfigField ‘int’, ‘CREDITS’, ‘10000’
            buildConfigField ‘String’, ‘APP_NAME’, ‘“APP DEBUG VERSION”’
            buildConfigField ‘boolean’, ‘LOG’, ‘true’
        }

        release {
            buildConfigField ‘int’, ‘CREDITS’, ’10’
            buildConfigField ‘String’, ‘FOO_STRING’, ‘“APP”’
            buildConfigField ‘boolean’, ‘LOG’, ‘false’
        }
    }
}
```

Using BuildConfig

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        int credits = BuildConfig.CREDITS;
        String app_name = BuildConfig.APP_NAME;
        boolean log = BuildConfig.LOG;

        ...
    }

}
```

Using BuildConfig

- In developing client-server systems, one scenario in which you will use BuildConfig is to specify the URL to the APIs for the debug and release build types of your app

```
android {
    buildTypes {
        debug {
            buildConfigField ‘String’, ‘DOMAIN’, ‘dev.myapp.com’
        }

        release {
            buildConfigField ‘int’, ‘DOMAIN’, ‘api.myapp.com’
        }
    }
}
```

ProGuard

ProGuard

- ProGuard is a tool that helps you shrinks, optimises, and ofuscates the codes of your app
- It does so by removing unused code, renaming classes, fields, and methods with semantically obscure names
- Why?
  - Creates an APK file with smaller size
  - Makes the app more difficult to reverse engineer

Reference:
http://developer.android.com/tools/help/proguard.html
http://proguard.sourceforge.net/

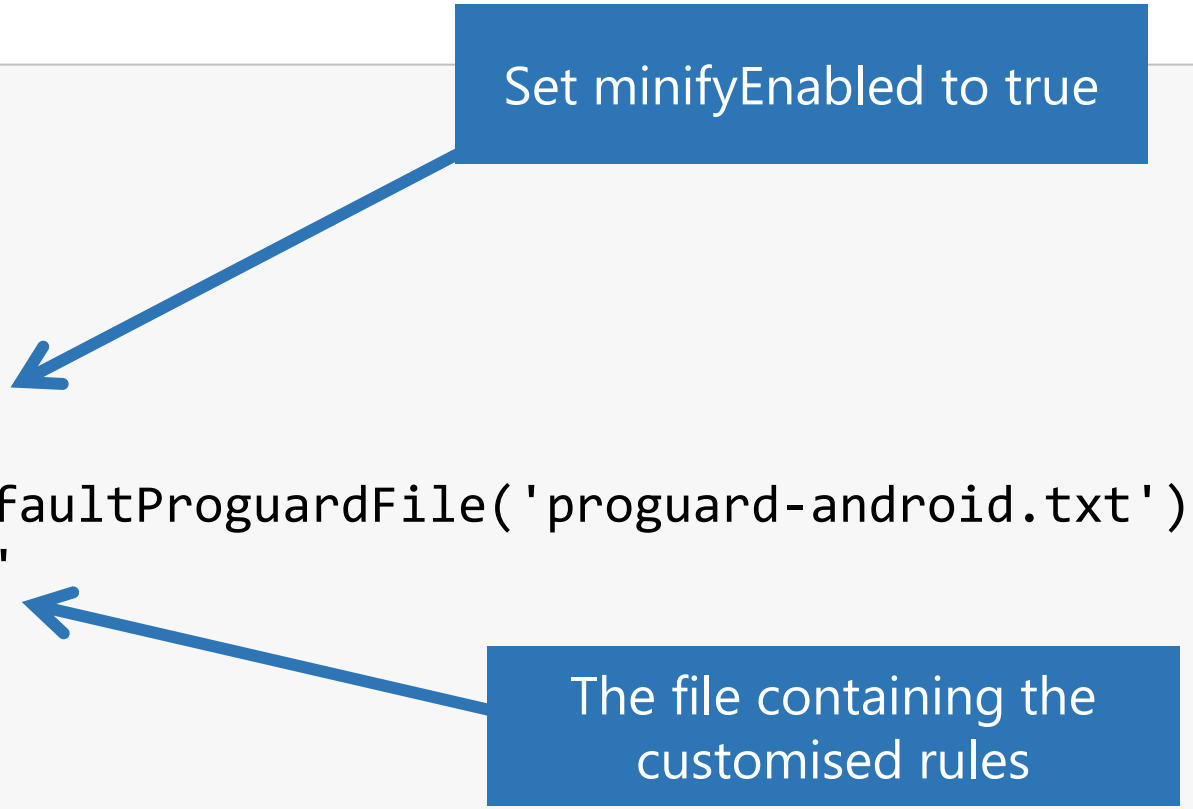
ProGuard

- ProGuard will only be run when you build your application in release mode (and if you have enabled it)
- To enable ProGuard:

```
android {
    ...

    buildTypes {
        release {
            minifyEnabled true
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
            'proguard-rules.pro'
        }

    }
}
```



Configuring ProGuard

- In some cases, ProGuard might not analyse your code correctly, and might remove codes that are actually required in your app
- You should specifically ask ProGuard to not modify some of the codes in your app when necessary
- Whenever you see **`ClassNotFoundException`** when building your app for release, you can add lines in the following format to the **`proguard-rules.pro`** files in the app module

```
-keep public class com.iems5722.app.MyClass001
```

Android + Arduino

Android + Arduino

- Android-Arduino LED Strip Lights
  https://www.youtube.com/watch?v=Hn9KfJQWqgI

Best Practices in Android Programming

Best Practices

References

- Best Practices | Android
  http://developer.android.com/guide/practices/index.html
- Best Practices for Performance | Android
  http://developer.android.com/training/best-performance.html

Project Presentations

Project

- To facilitate testing and marking, please name the package of your app using the following format:
  com.iems5722.groupX (Replace X with your group number)
- Send your app's APK file (or a link to that file if it is too large) to my email (cmauyeung@ie.cuhk.edu.hk) before the end of 27th April, 2016 (Wednesday)
- The APK files will be shared to the class so that we can try your apps during the presentation

Project Presentations

- You will have your project presentations on 28th April, 2016 (Thursday)
- Please check the following page for the project groups
- Presentation order follows the group numbers

https://goo.gl/j6LhGL

Project Presentations

- Each group will be given 8 mins to do the presentation + 2 mins Q&A
- What you should present:
  1. Your idea: Why such idea? What is the problem you want to solve?
  2. Your solution: the features of the app and the system your developed
  3. Technical details of your implementation: flow chart, diagrams, graphics, screenshots are encouraged
  4. How would you further improve it?
- You can use the object projector to demonstrate your app in a real Android device

Project Presentations

- Send you project report to my email (cmauyeung@ie.cuhk.edu.hk) on or before 6th May, 2016 (Thursday)
- Your report should contain descriptions about your project, similar to the points mentioned for the presentation
- Your can hand in your presentation slides if it already contains the descriptions

End of Lecture 12

Looking forward to
your project presentations!