

Lifecycle Management

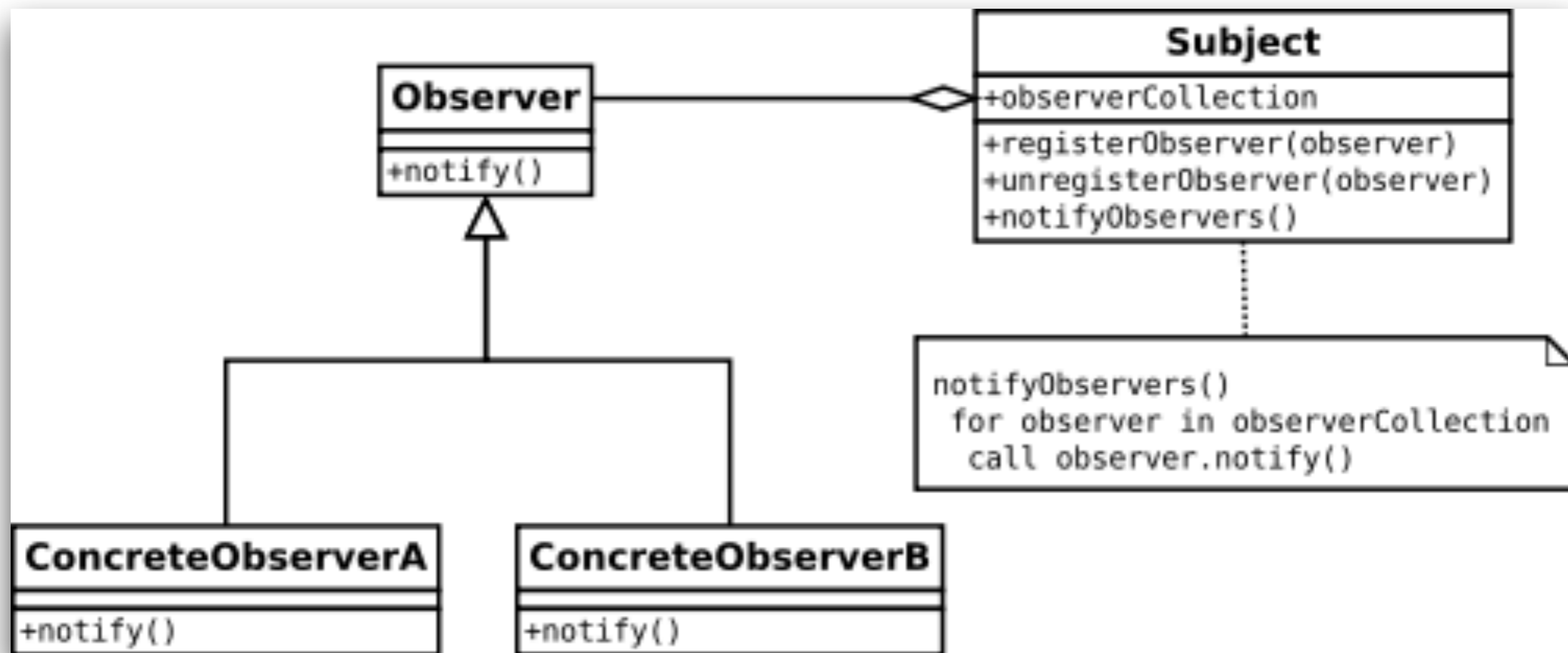
Android

Mobile and Ubiquitous Games

ICS 163

Donald J. Patterson

Callback

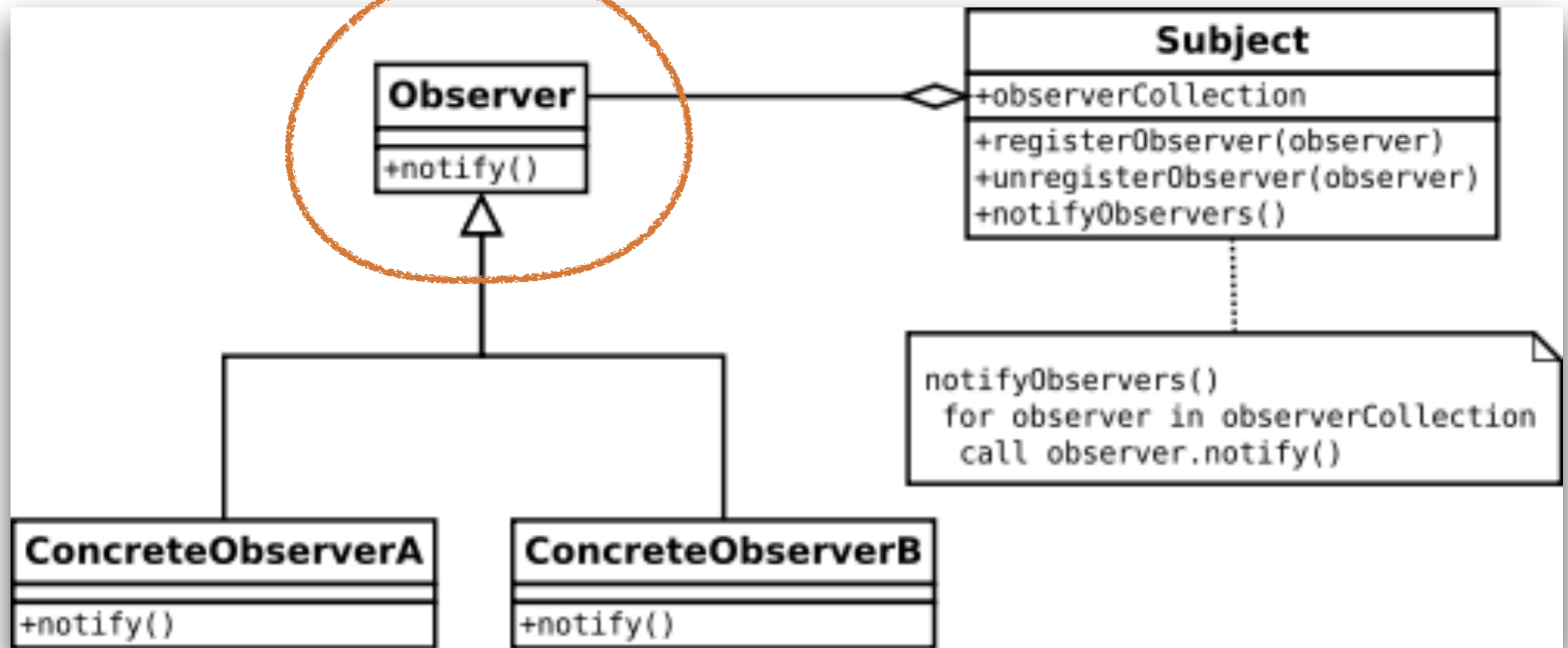


Activity Lifecycle

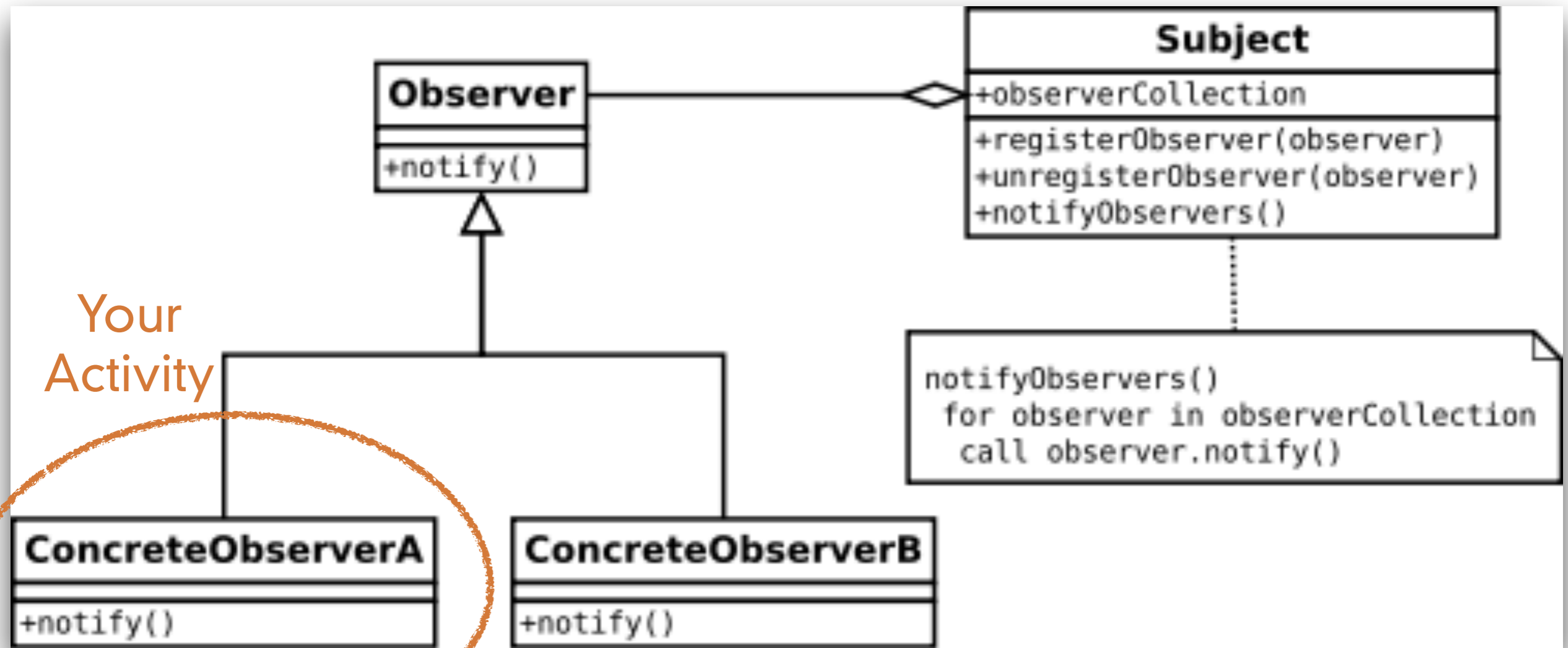
- Unlike traditional Java, Android does not use a “main” function
- It uses a sophisticated set of callbacks
- Each step of the callback corresponds to a step in the **lifecycle** of the app
- This is so that the phone can shut your app down when important things happen, like a phone calls arriving or when a user switches apps
- An implementation of the Activity class contains the callbacks

Callback

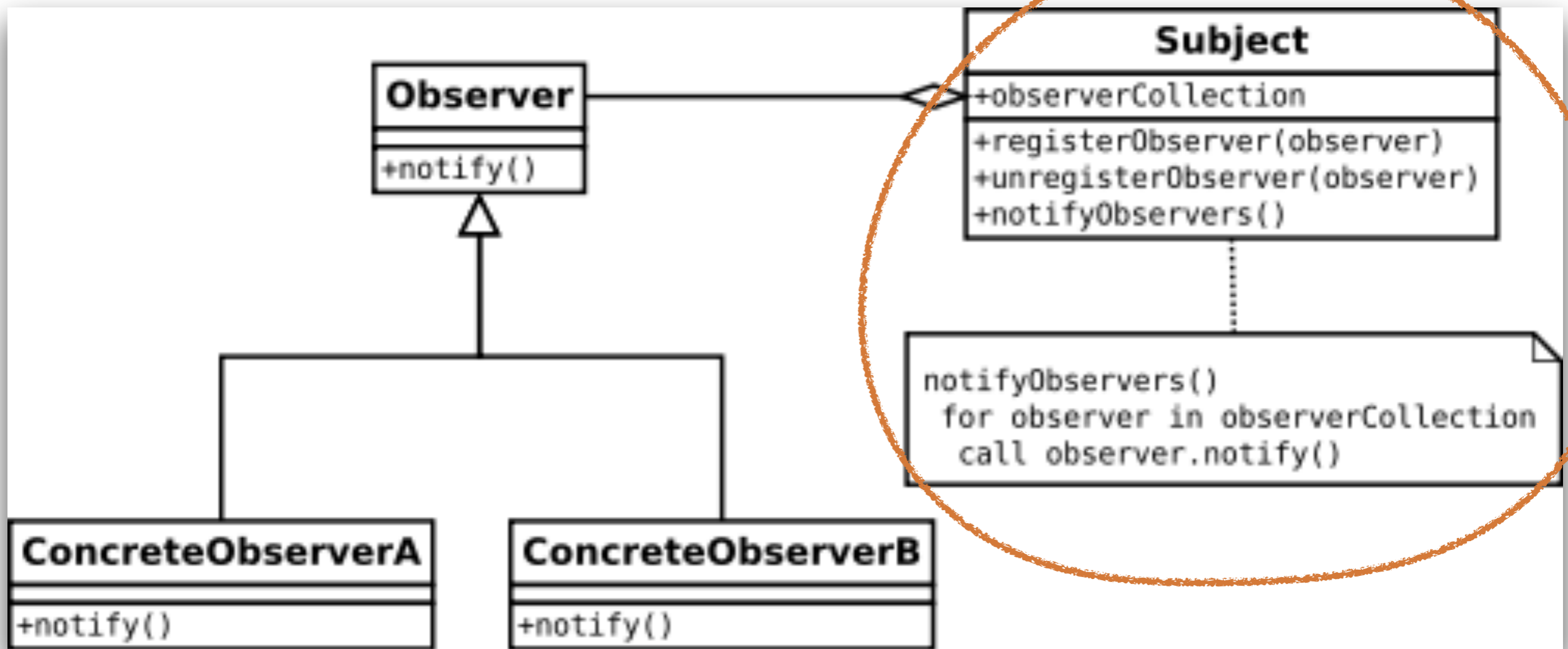
Activity Class



Callback



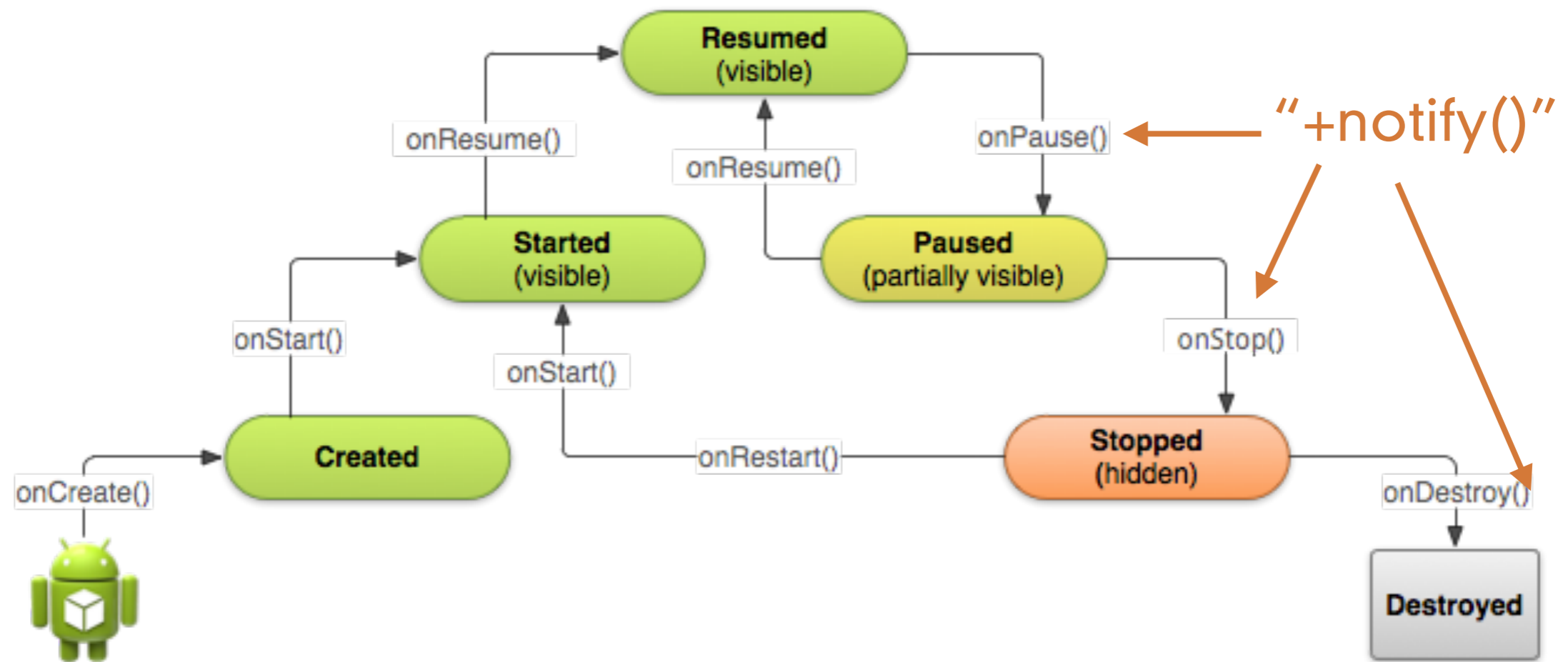
Android OS



Lifecycle Management

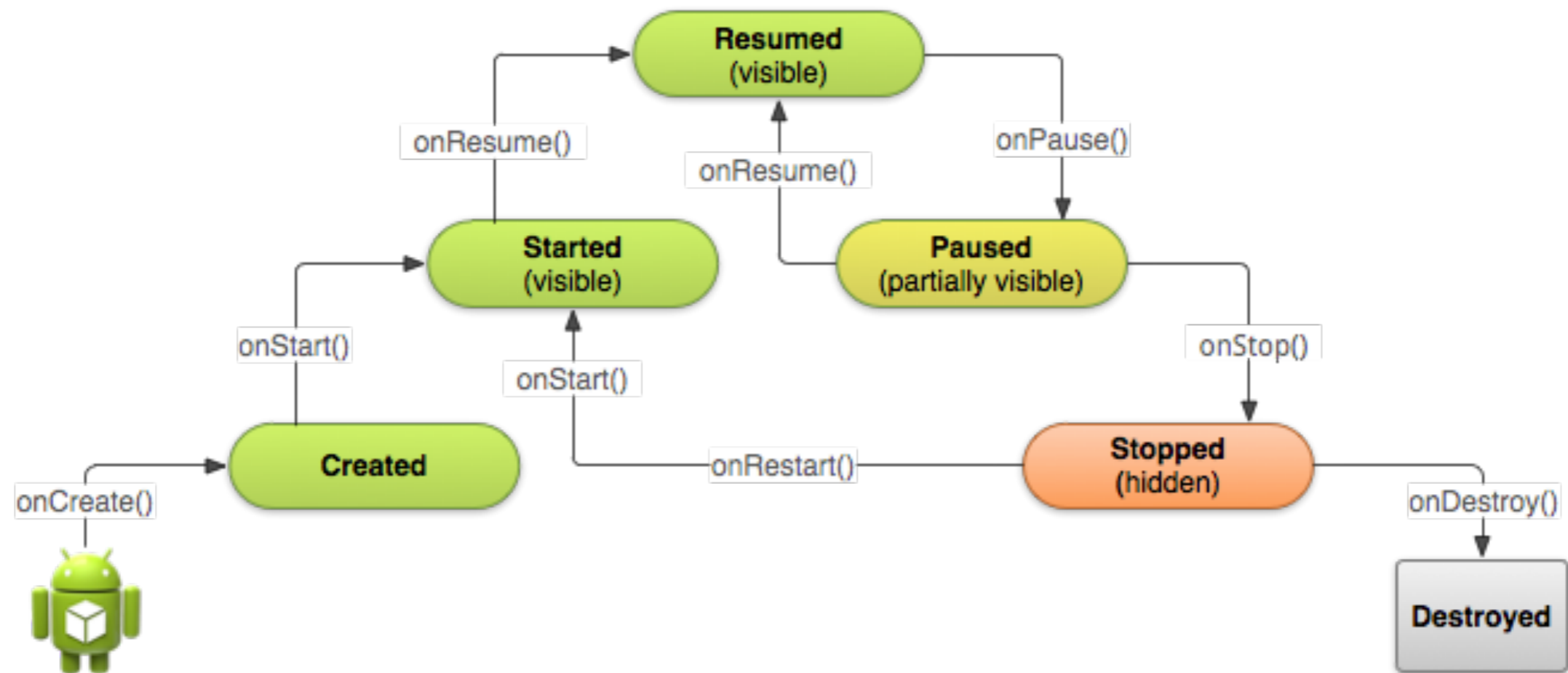


Activity Lifecycle



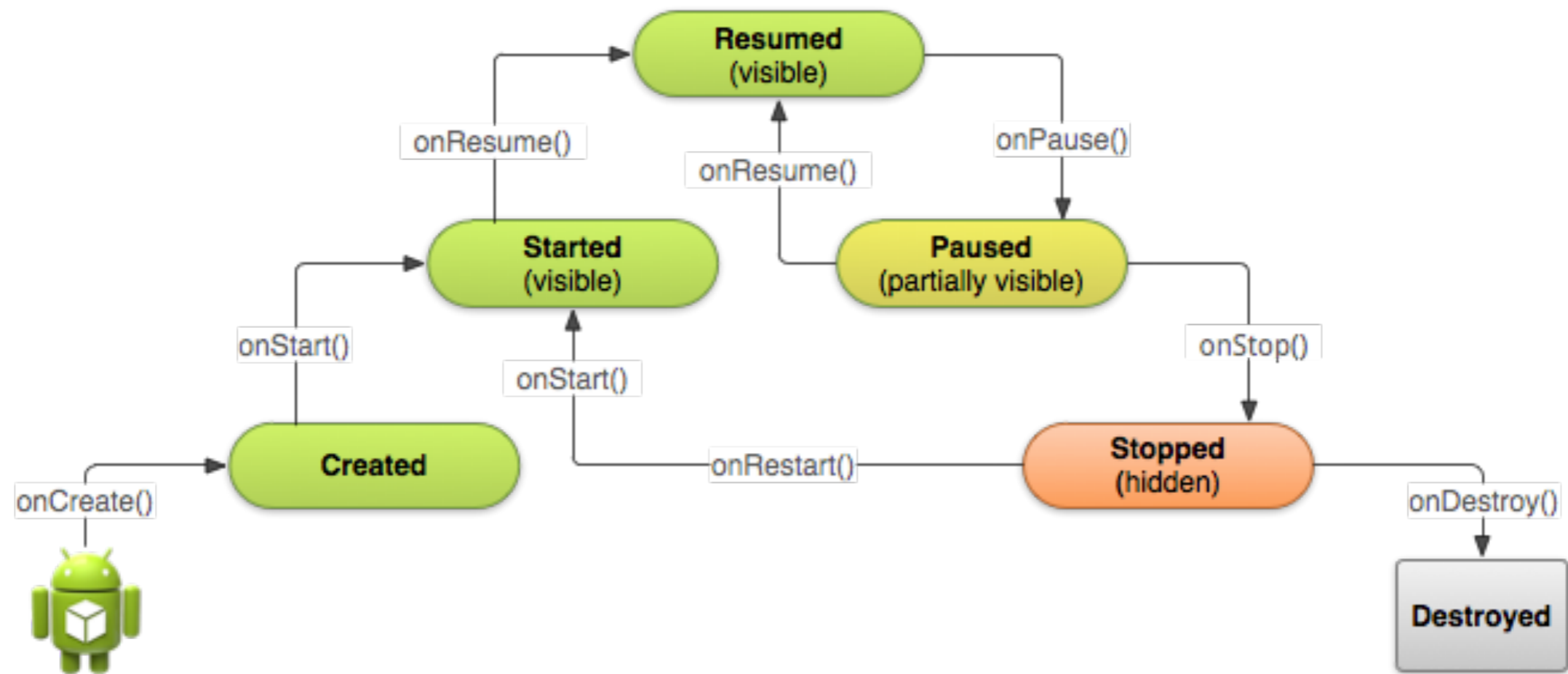
- Key loops
 - Entire Lifetime
 - `onCreate()`- `onDestroy()`

Activity Lifecycle



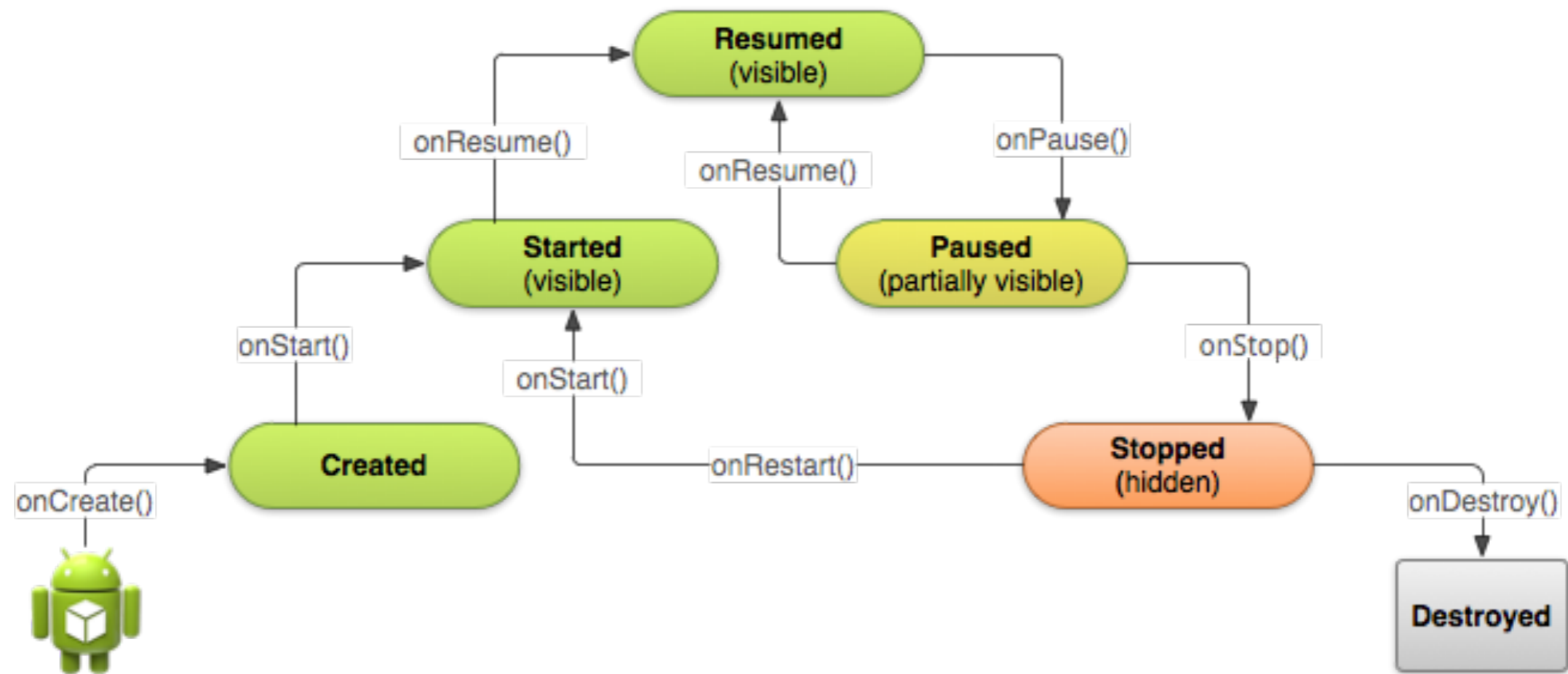
- Key loops
 - Visible Lifetime
 - `onStart()` - `onStop()`

Activity Lifecycle



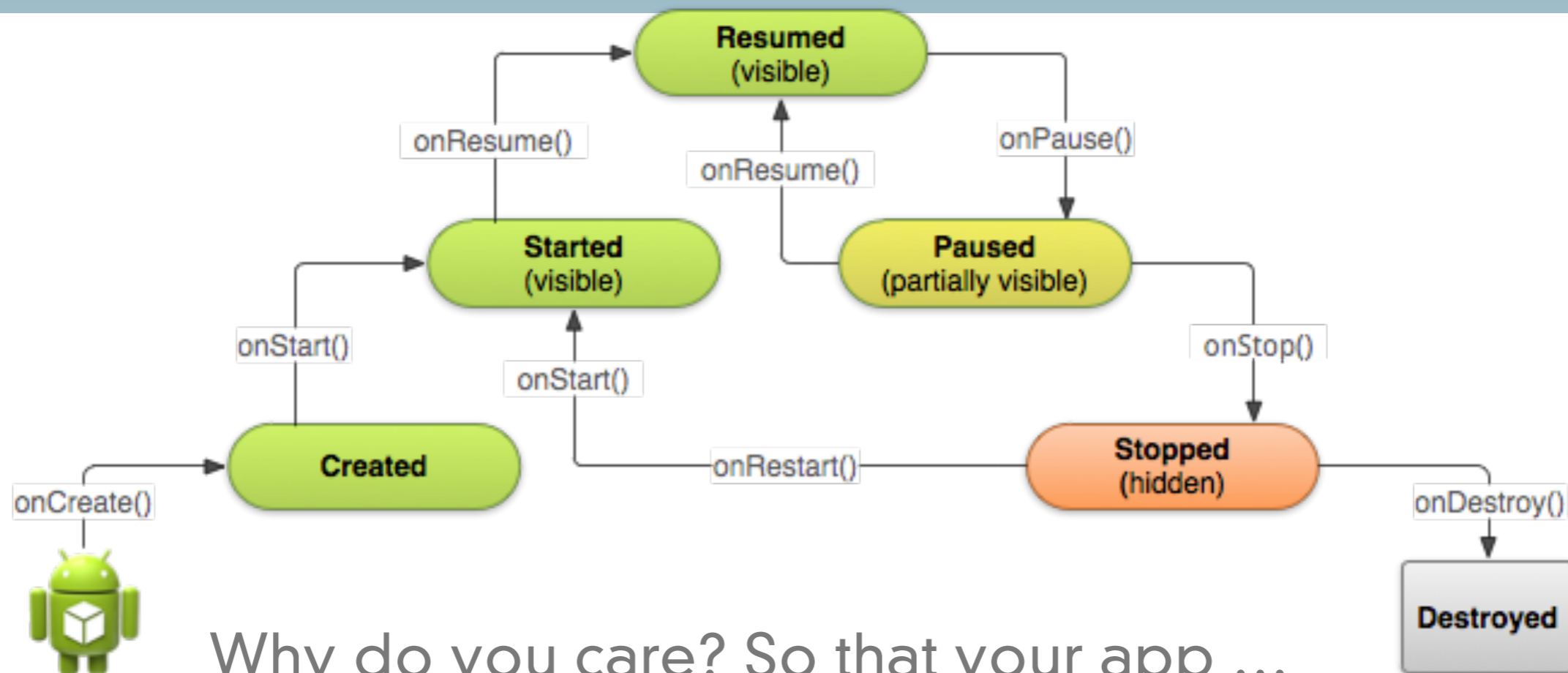
- Key loops
 - Foreground Lifetime
 - `onResume()` - `onPause()`

Activity Lifecycle



- `onPause()` may be followed by kill

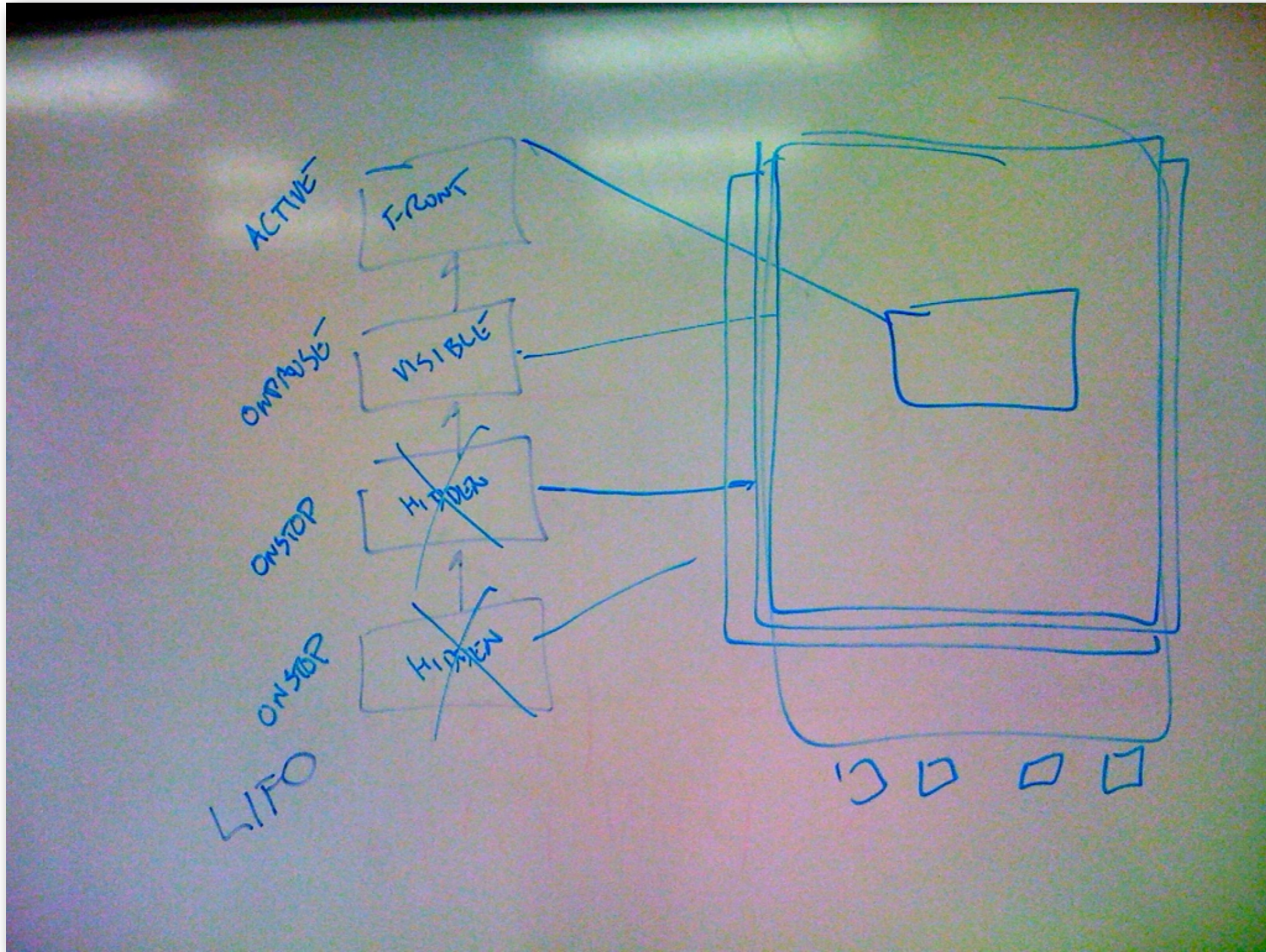
Activity Lifecycle



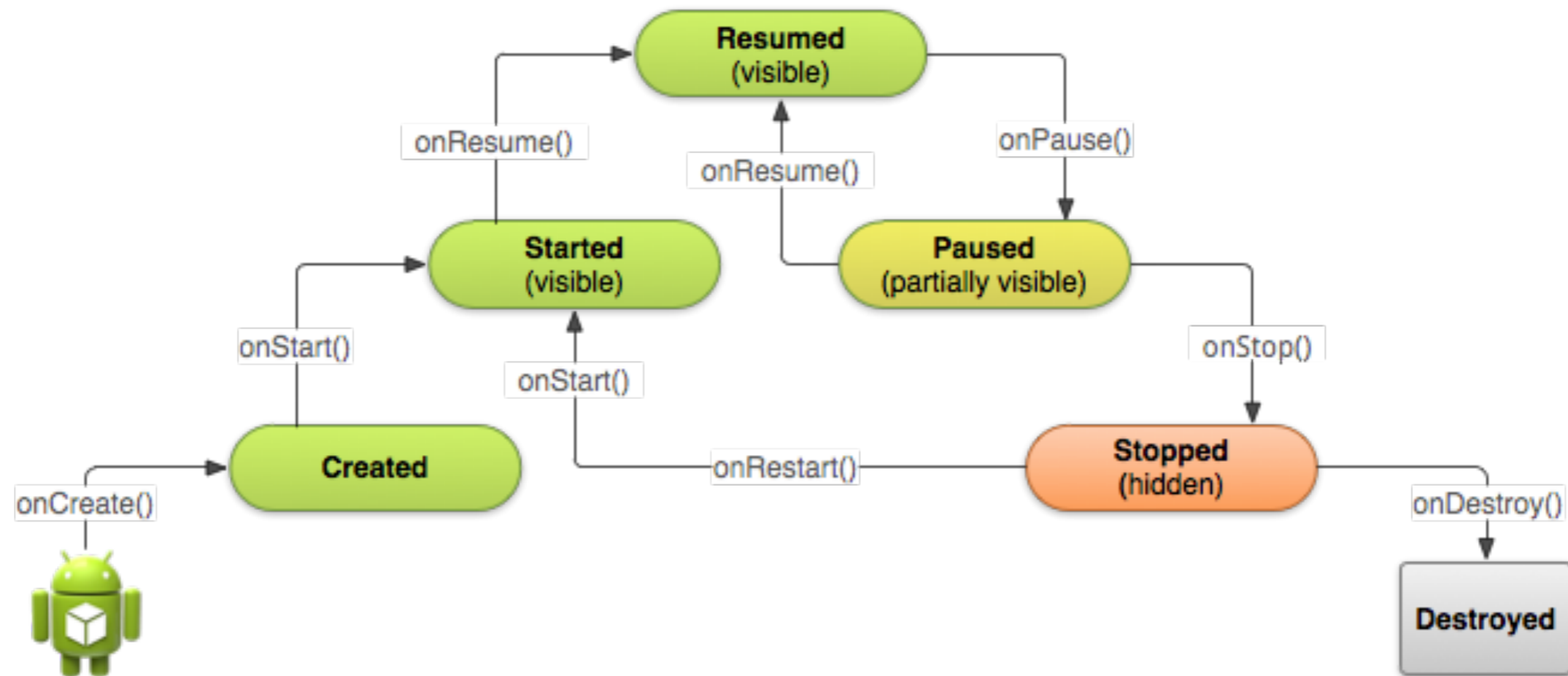
Why do you care? So that your app ...

- Does not crash if the user receives a phone call or switches to another app while using your app.
- Does not consume valuable system resources when the user is not actively using it.
- Does not lose the user's progress if they leave your app and return to it at a later time.
- Does not crash or lose the user's progress when the screen rotates between landscape and portrait orientation.

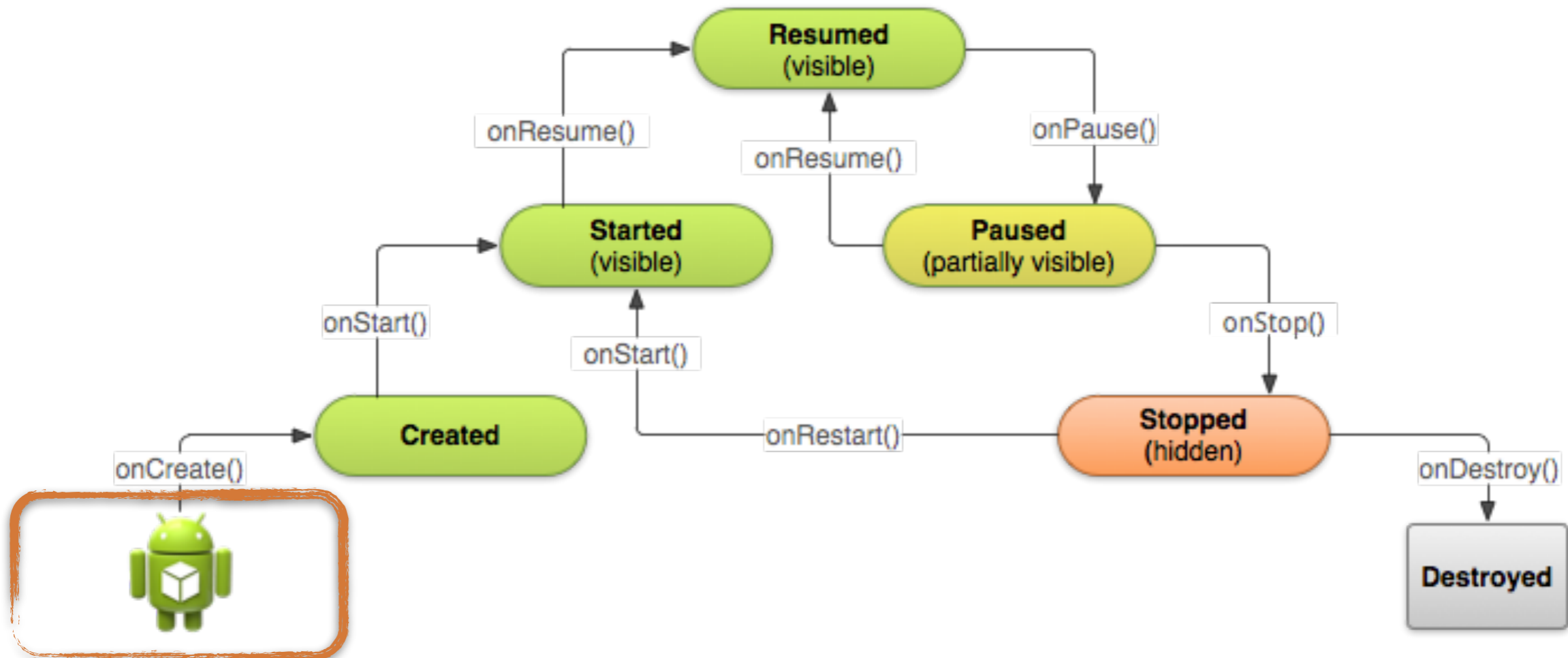
Activity Stack



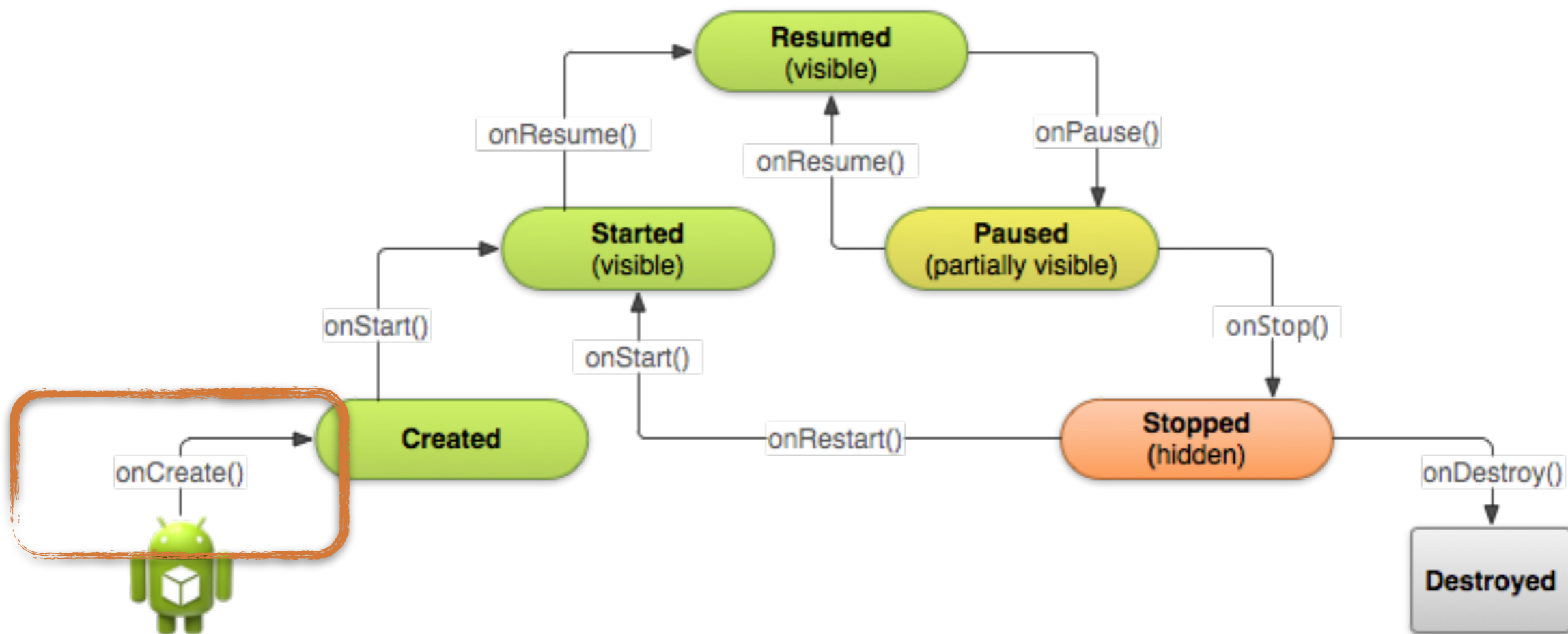
Activity Lifecycle



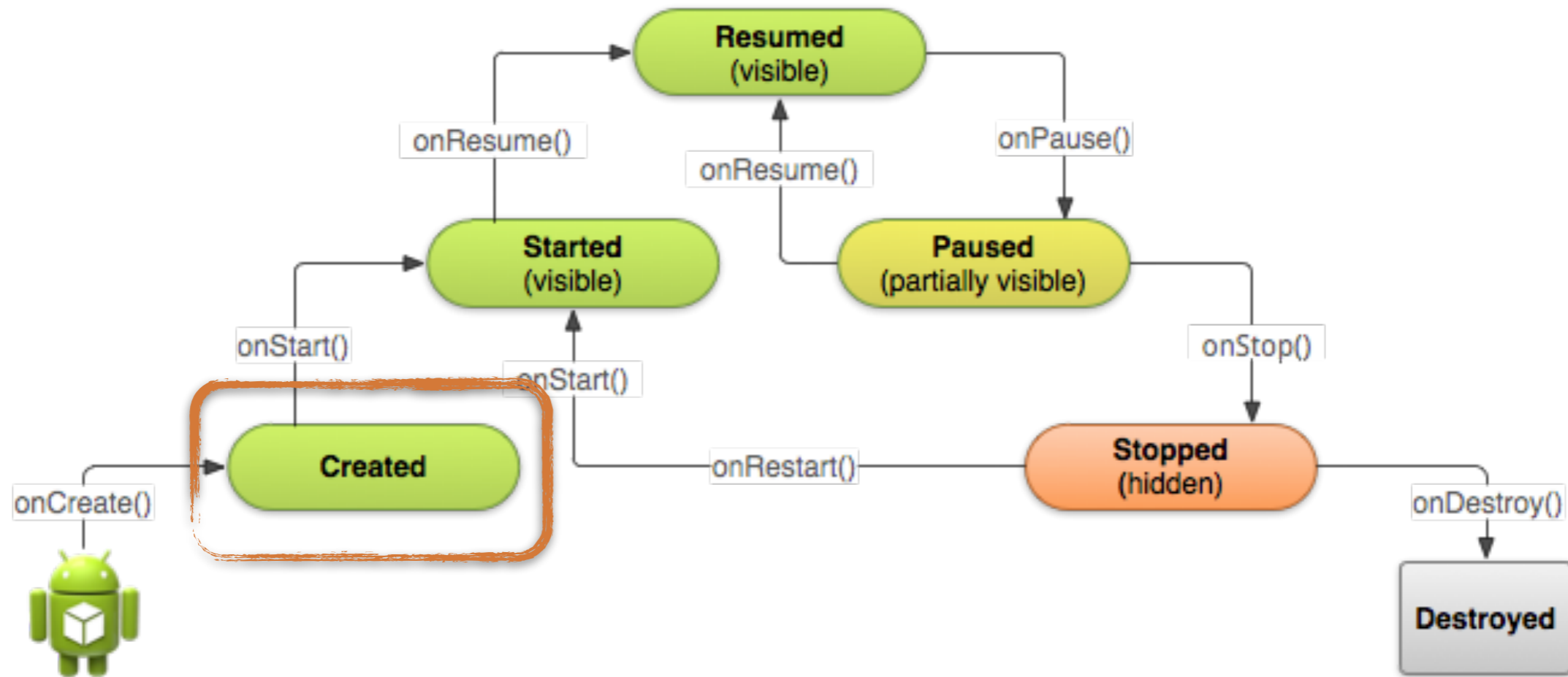
Activity Lifecycle



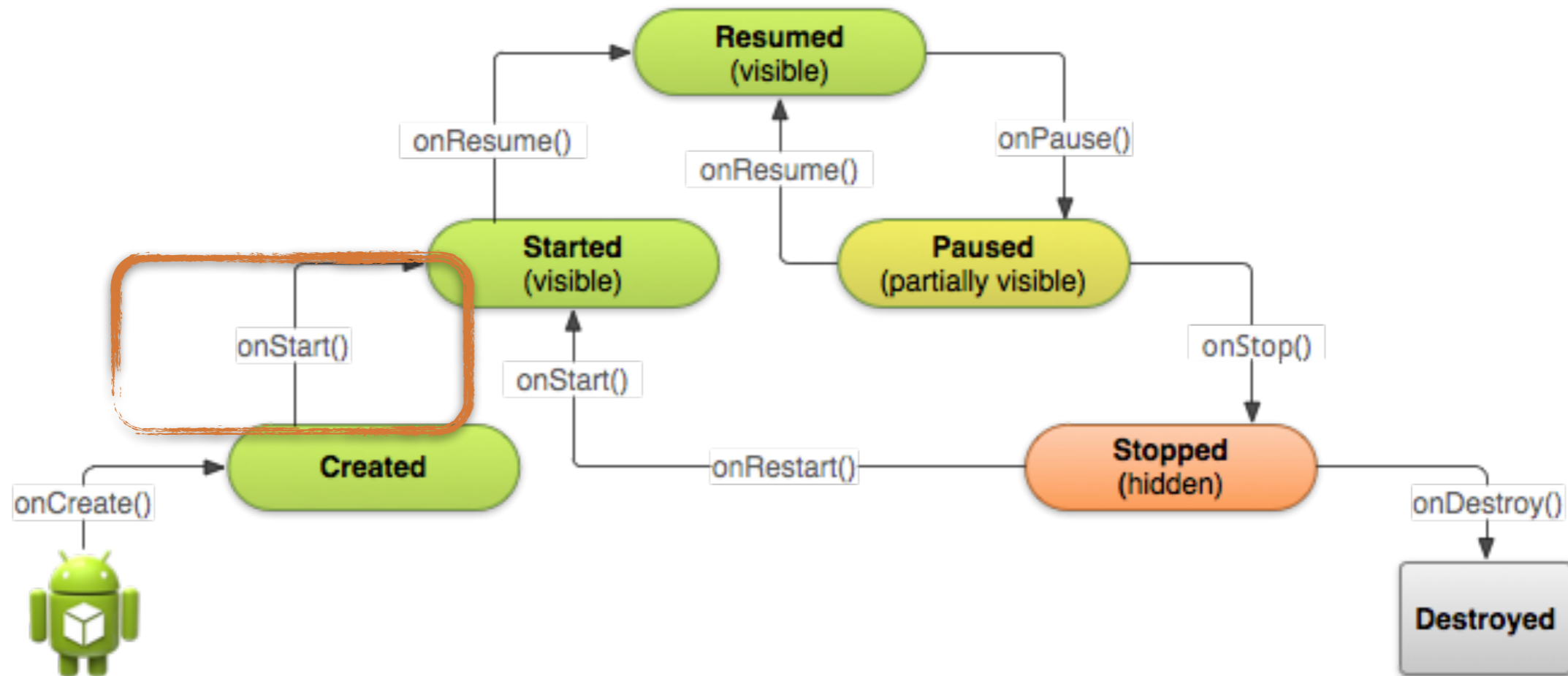
Activity Lifecycle



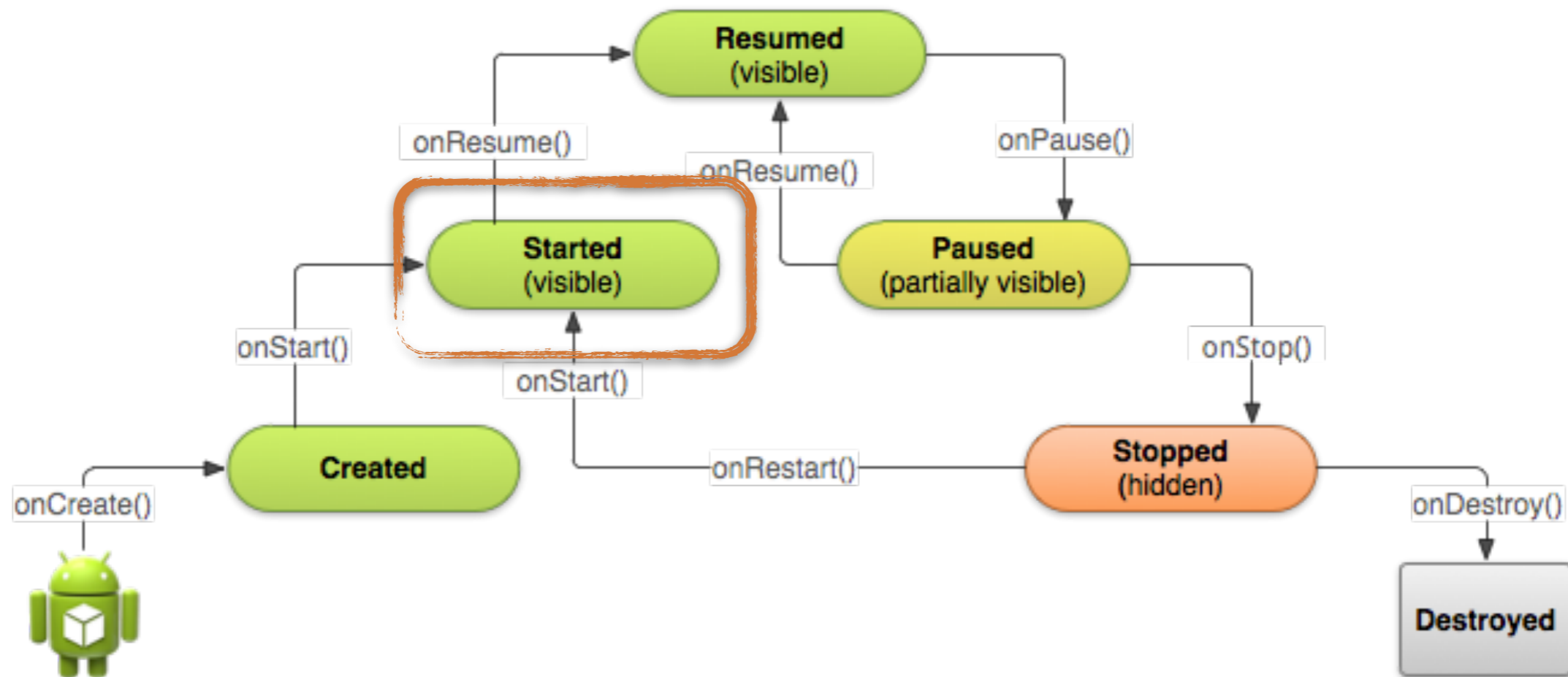
Activity Lifecycle



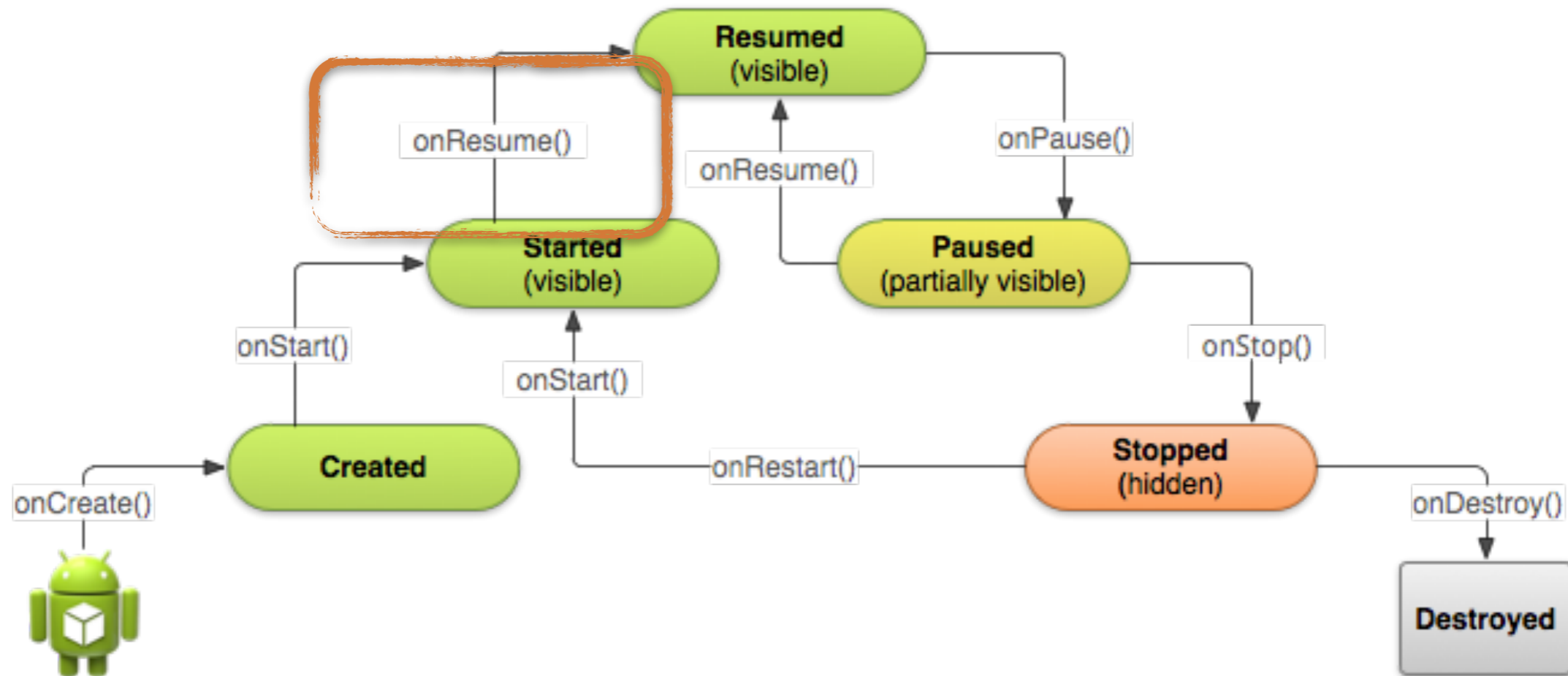
Activity Lifecycle



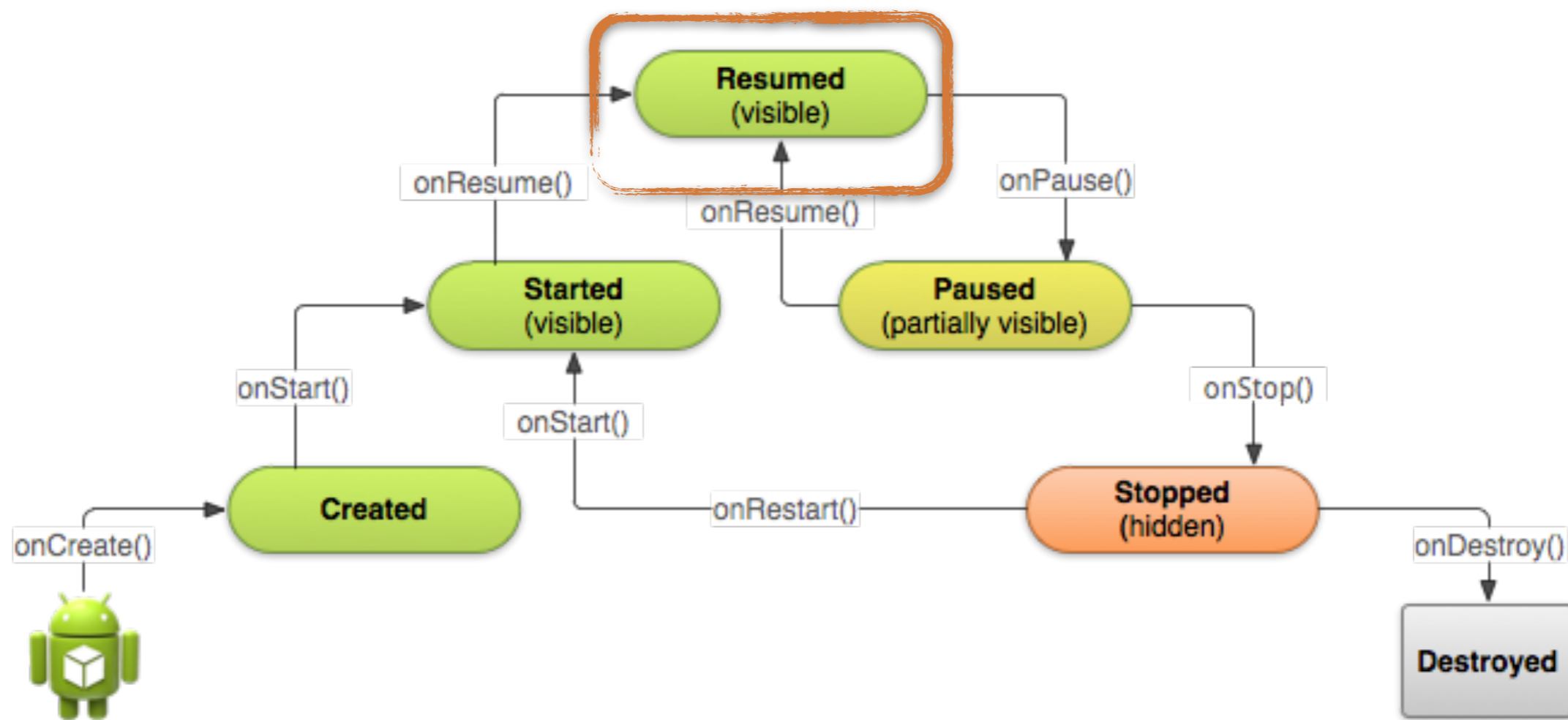
Activity Lifecycle



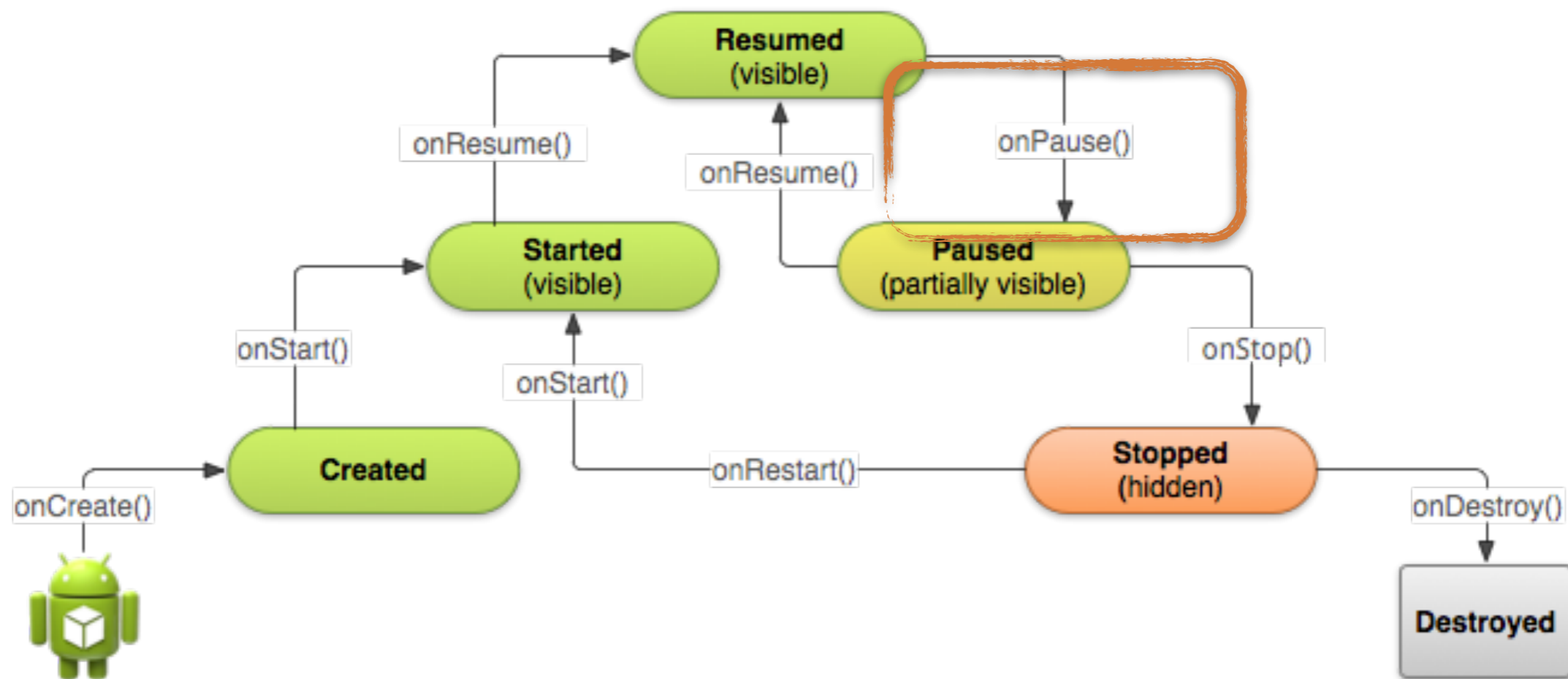
Activity Lifecycle



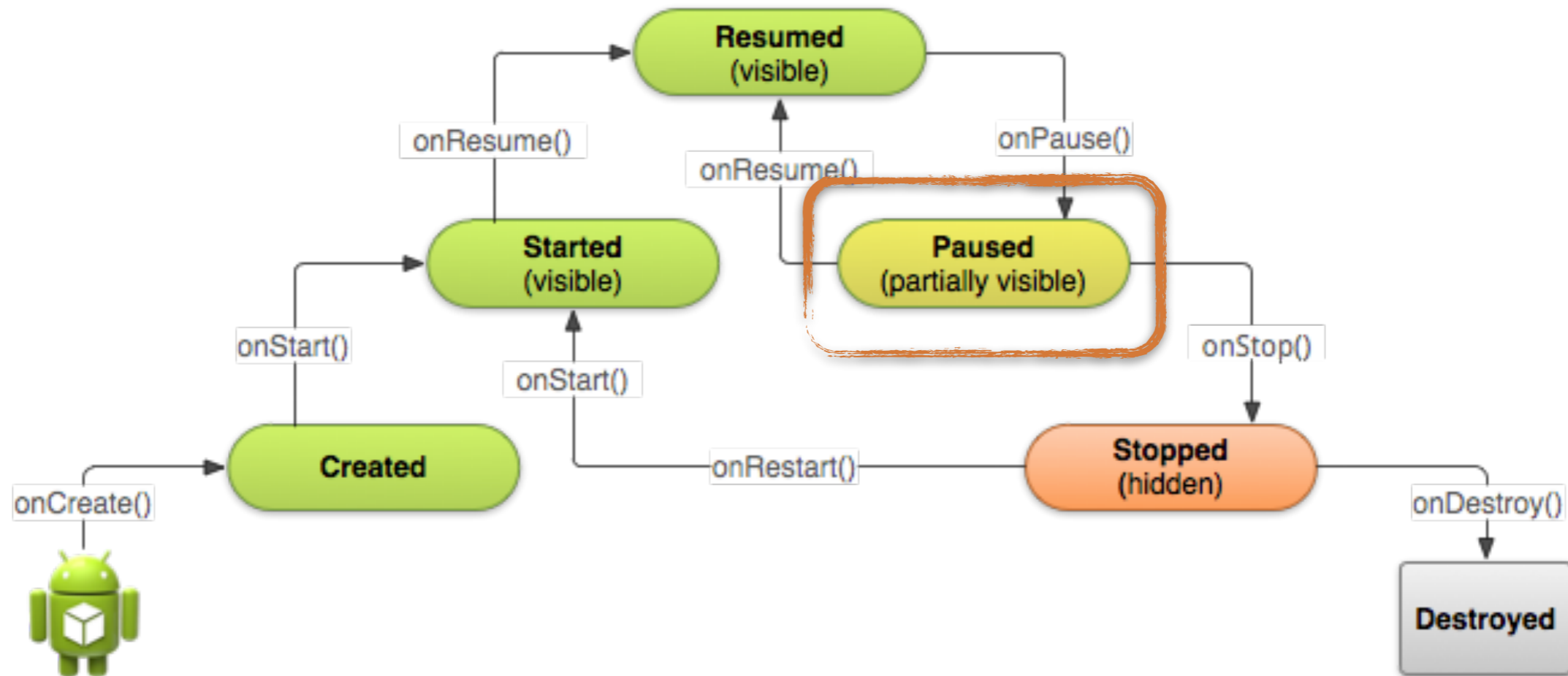
Activity Lifecycle



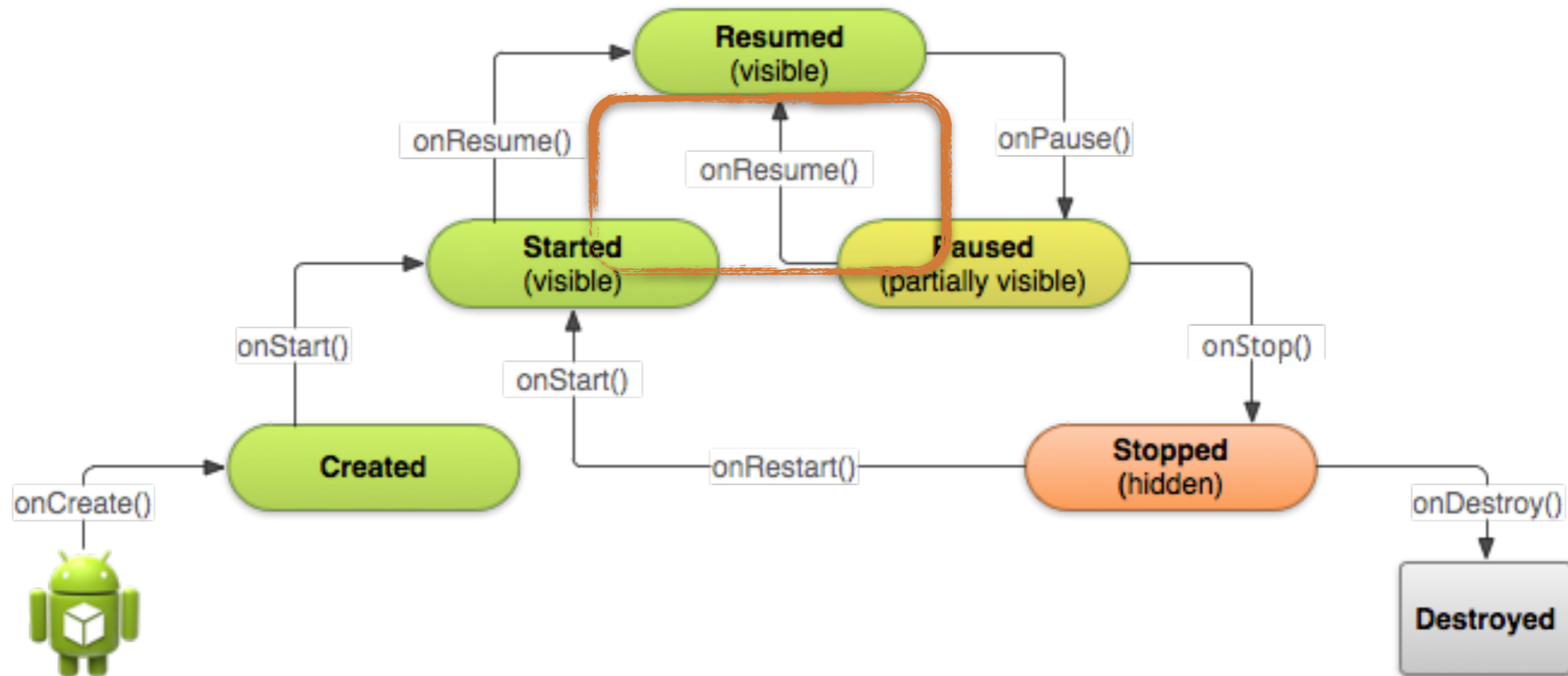
Activity Lifecycle



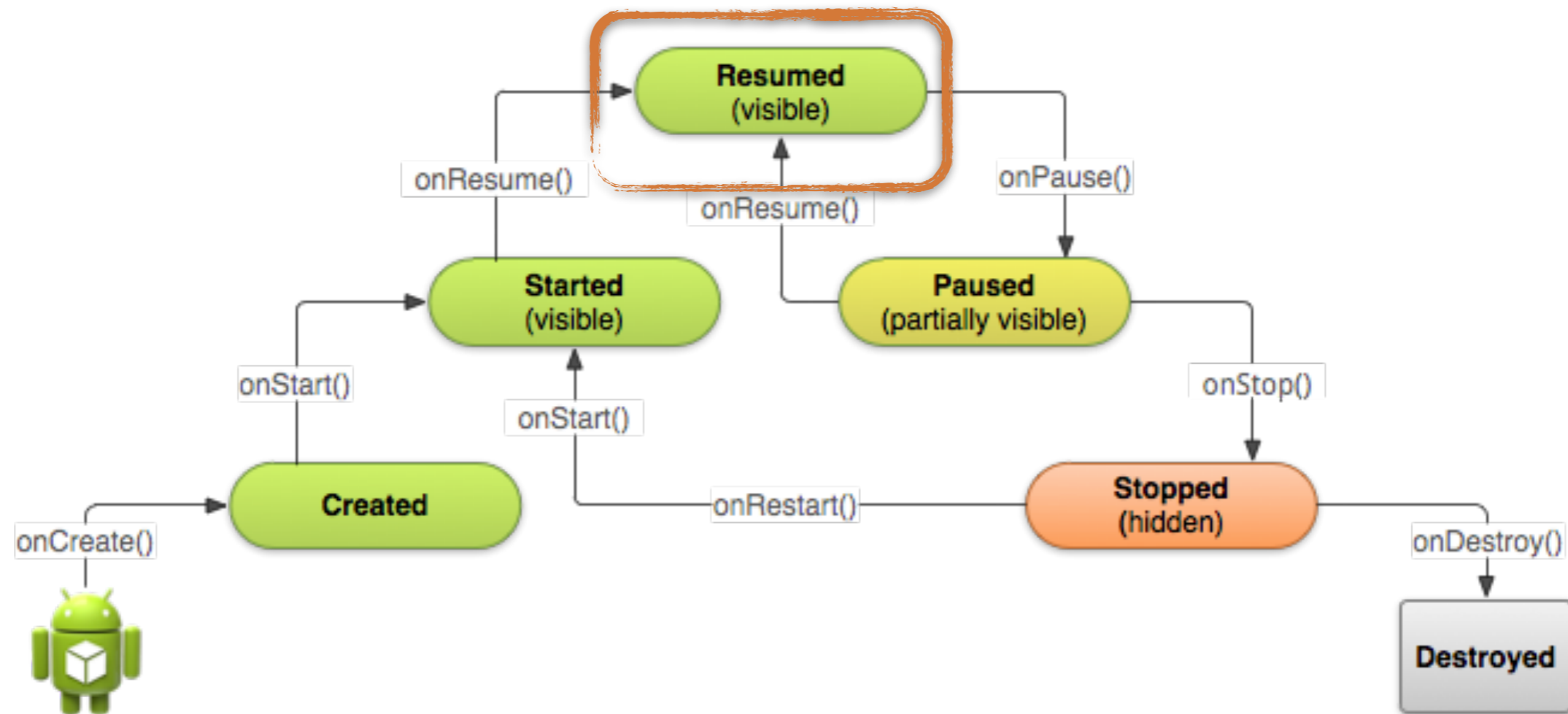
Activity Lifecycle



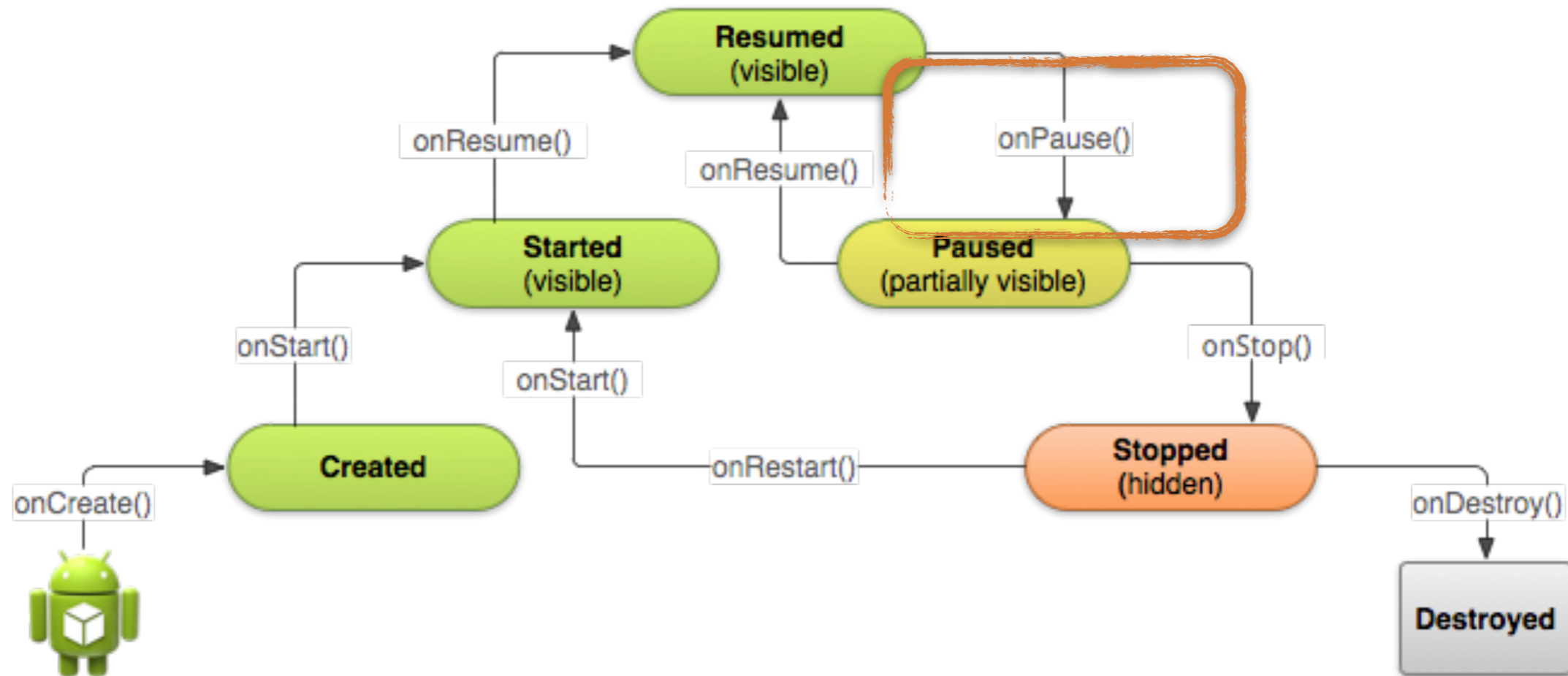
Activity Lifecycle



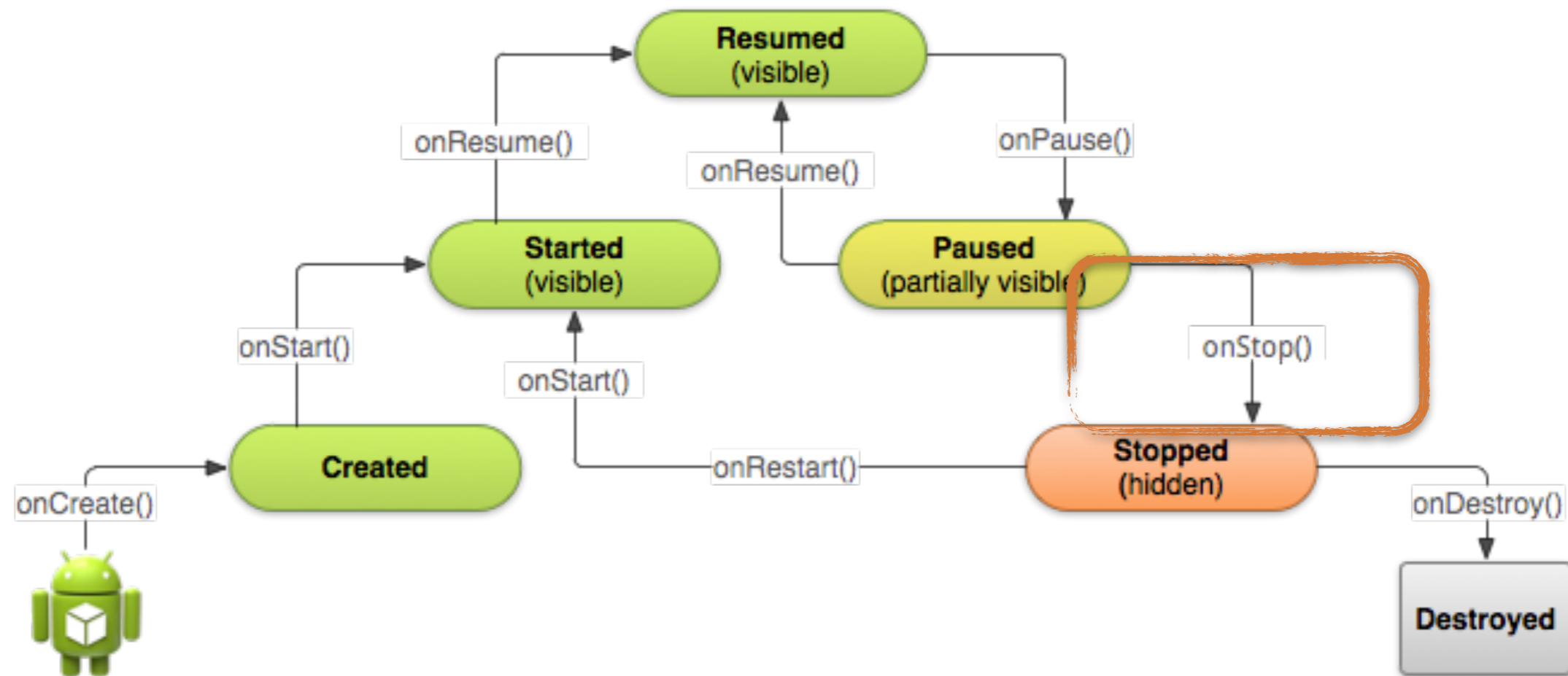
Activity Lifecycle



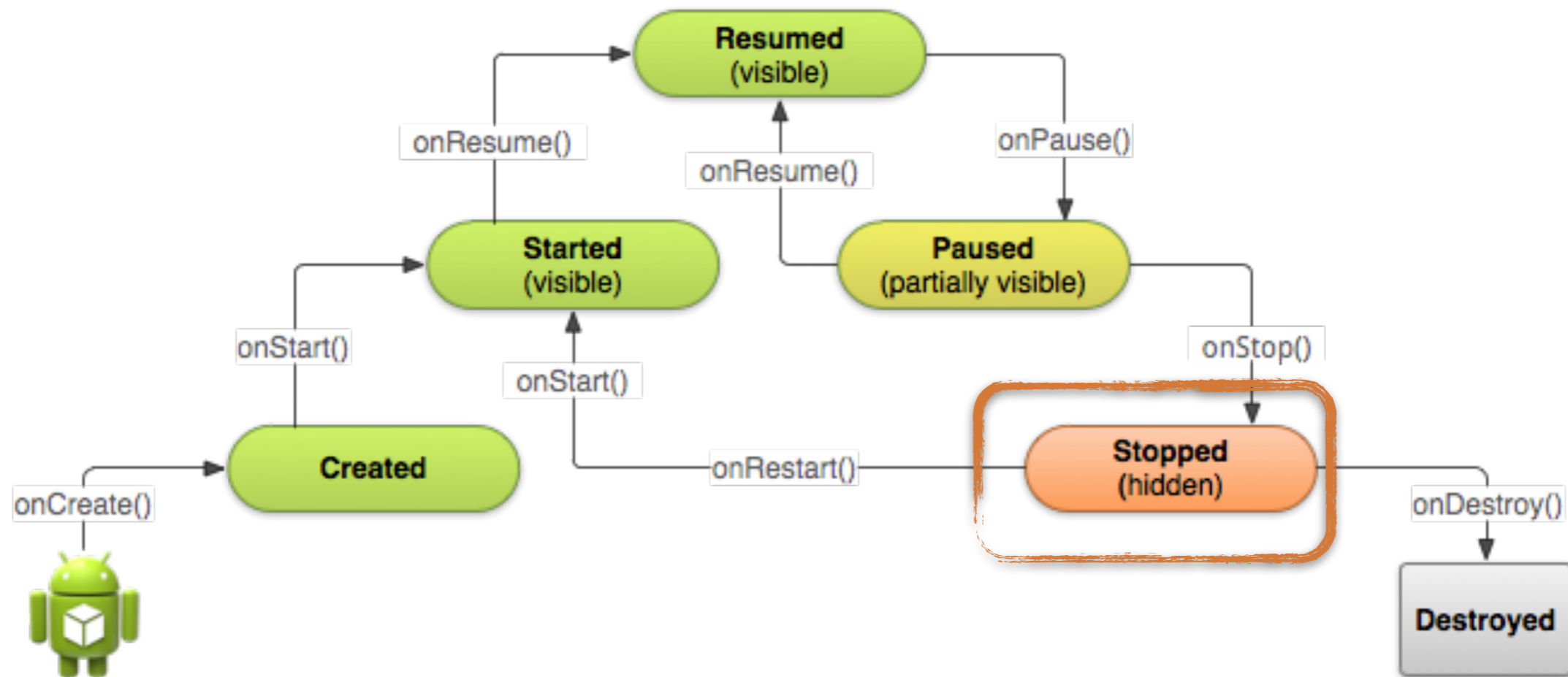
Activity Lifecycle



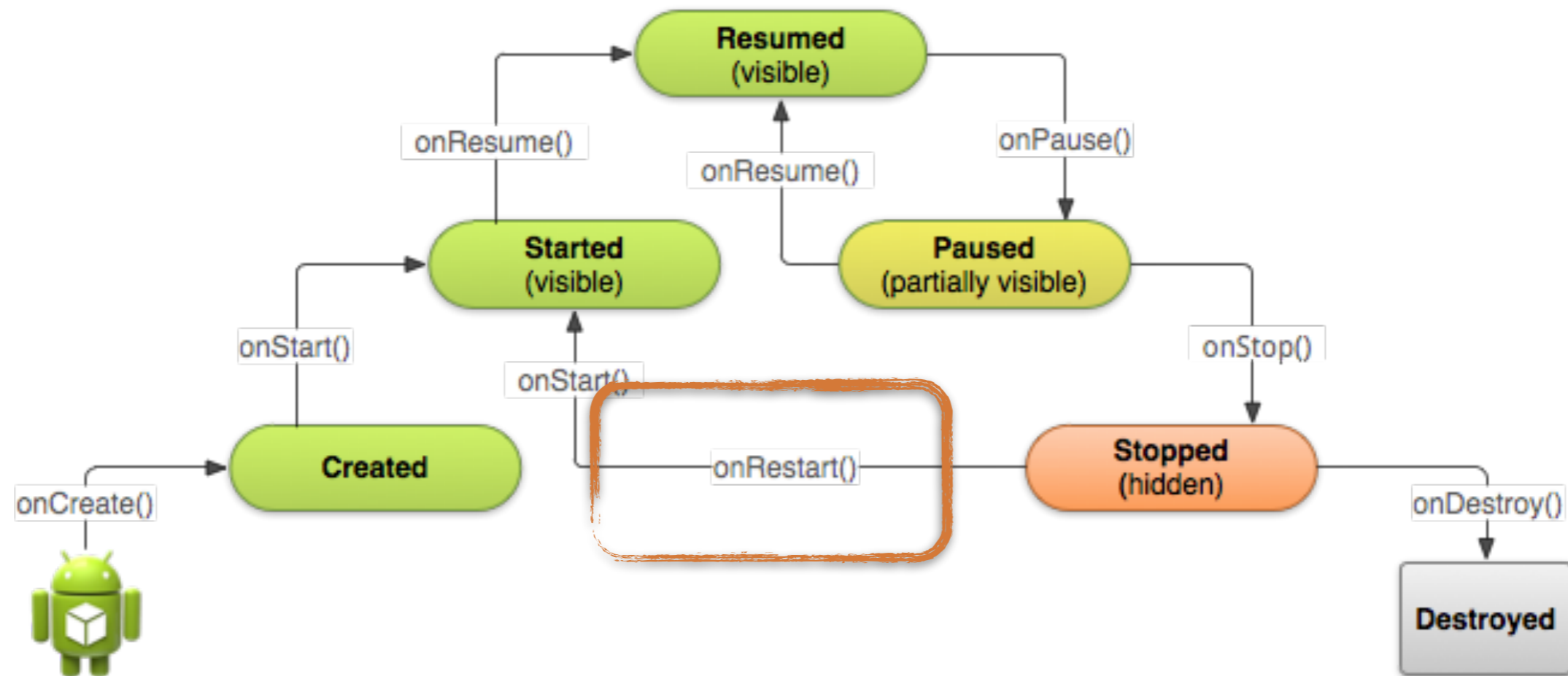
Activity Lifecycle



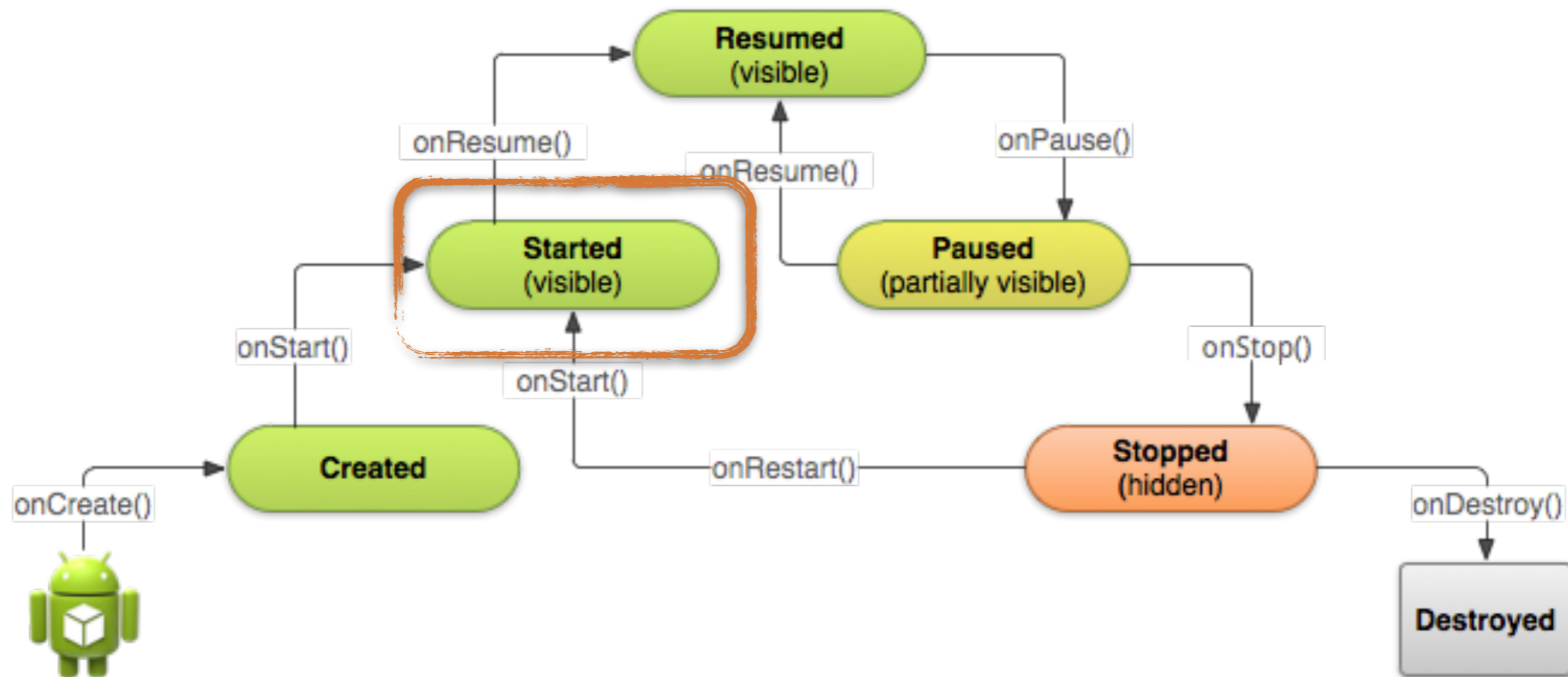
Activity Lifecycle



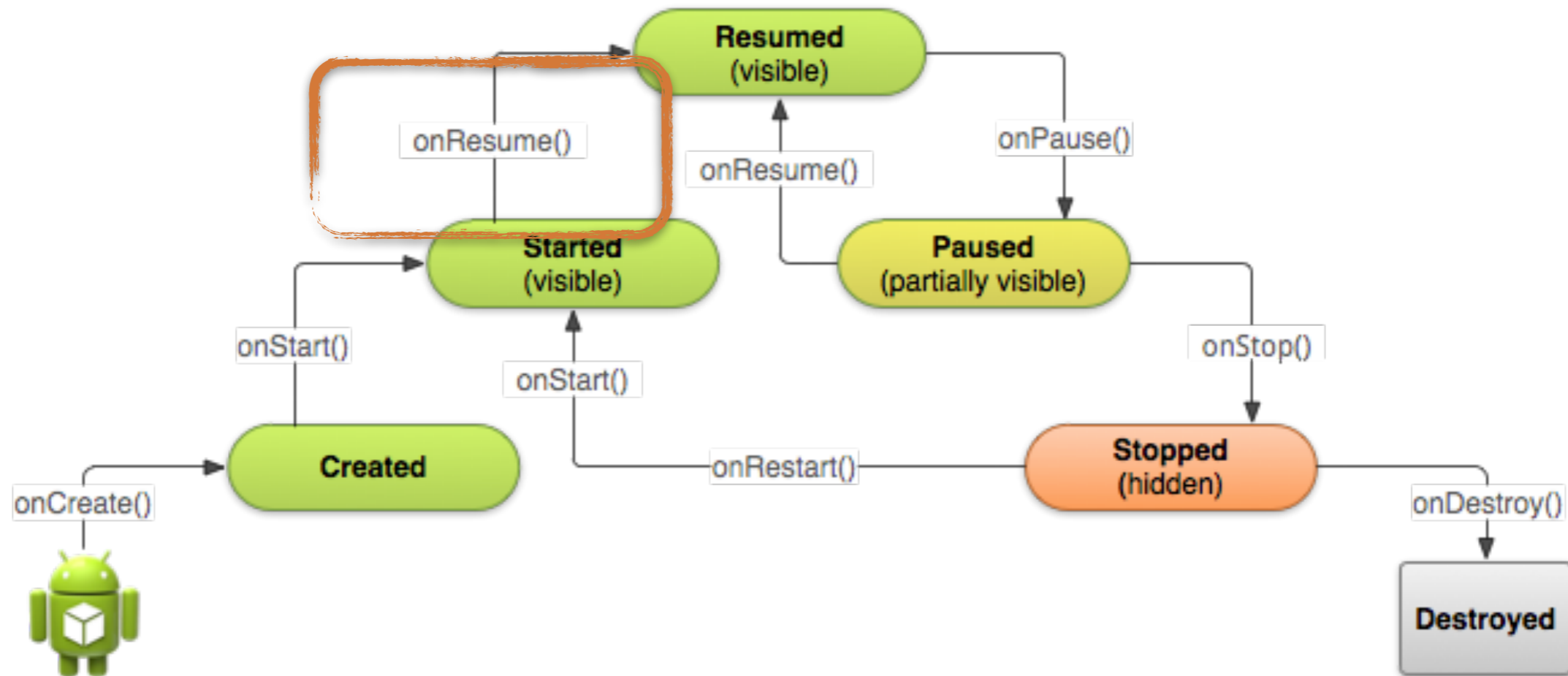
Activity Lifecycle



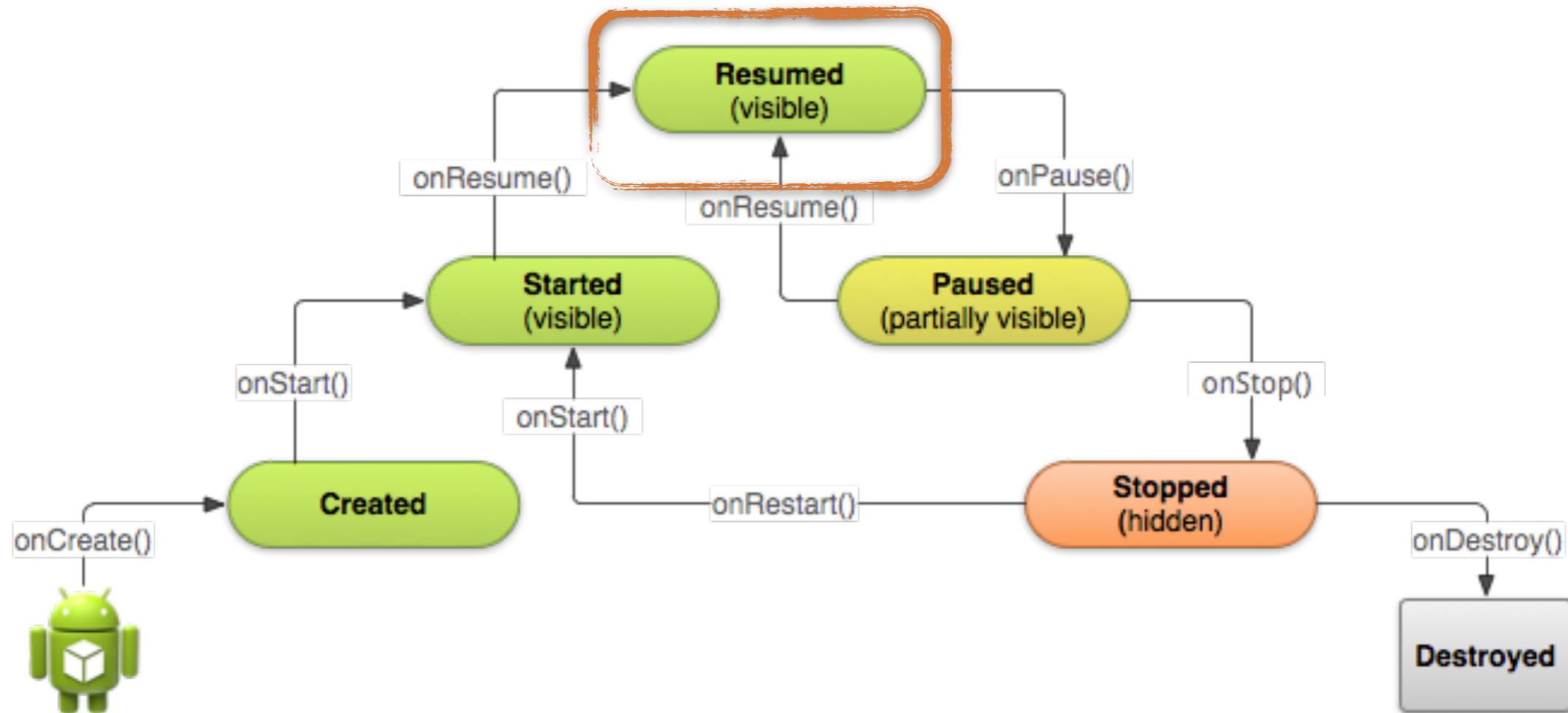
Activity Lifecycle



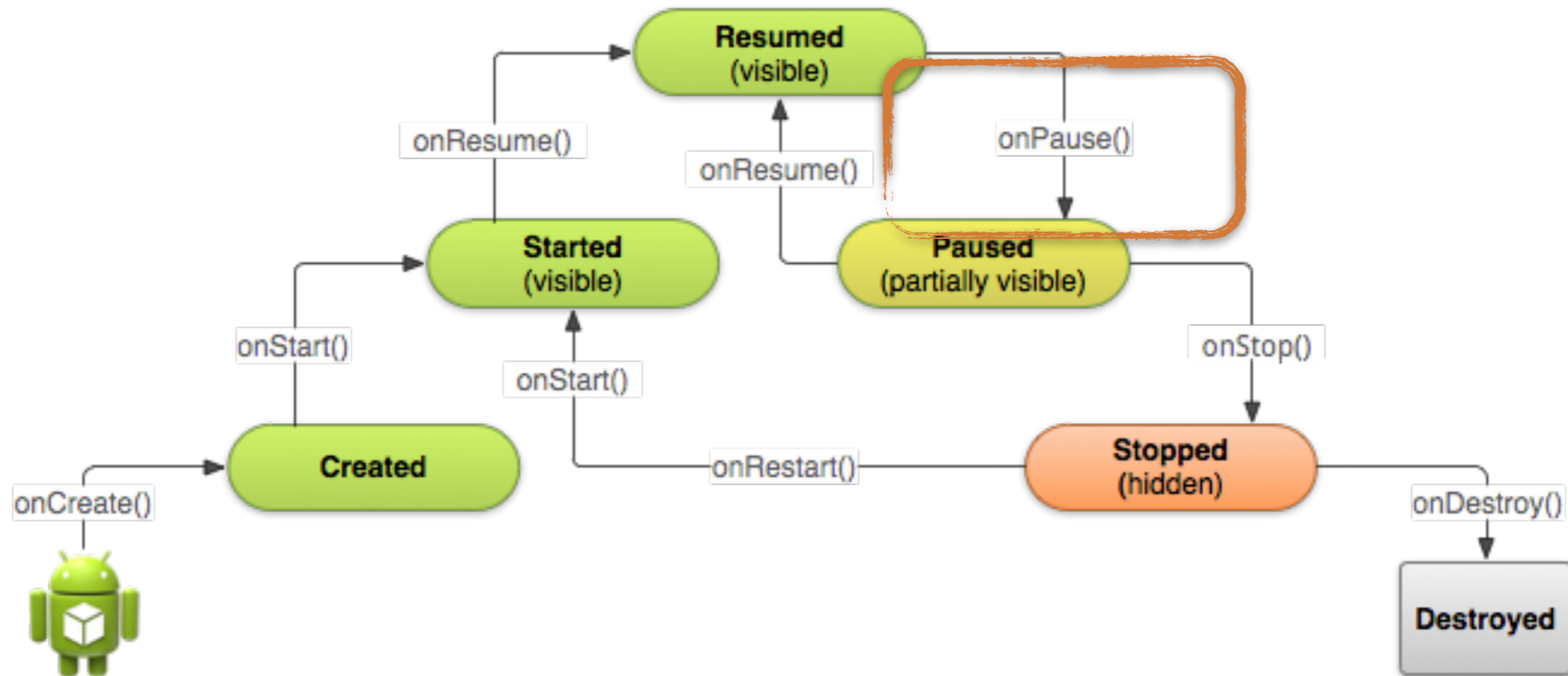
Activity Lifecycle



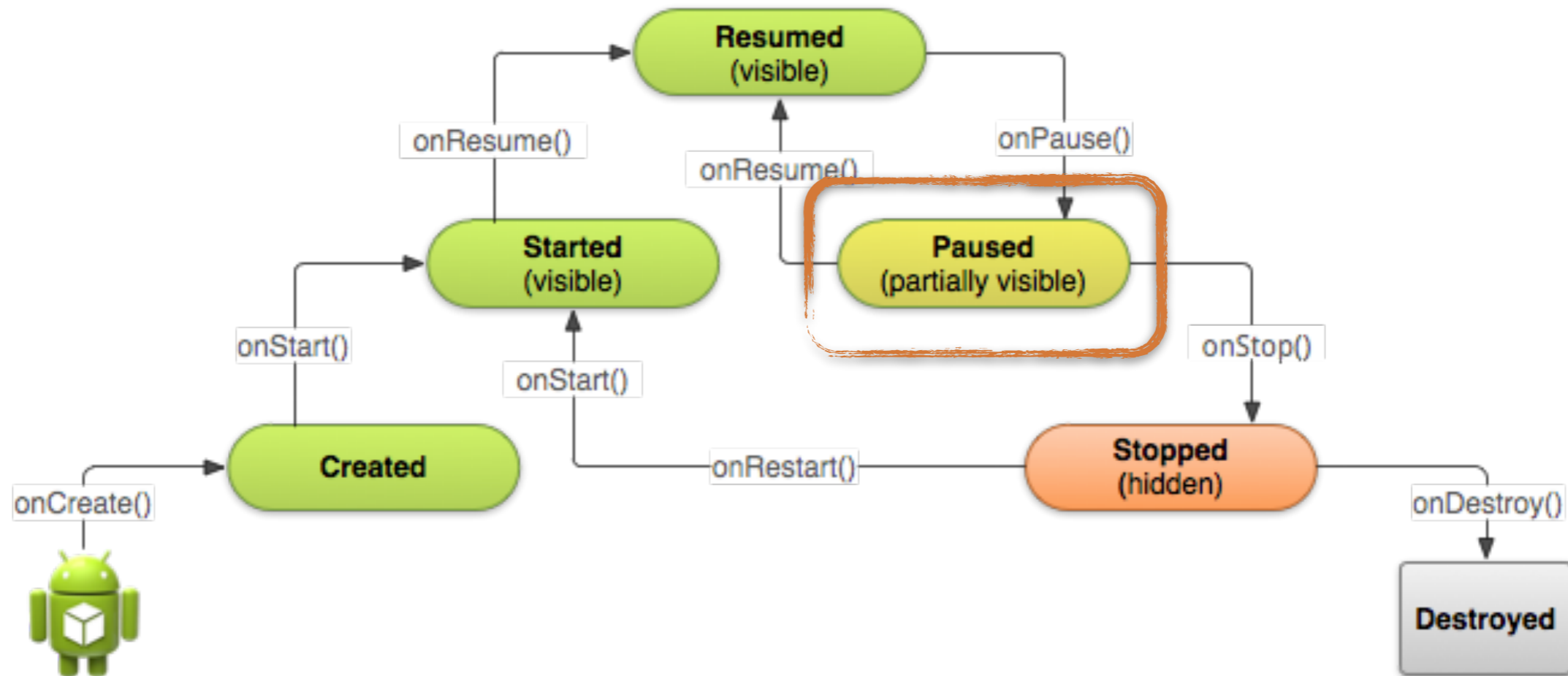
Activity Lifecycle



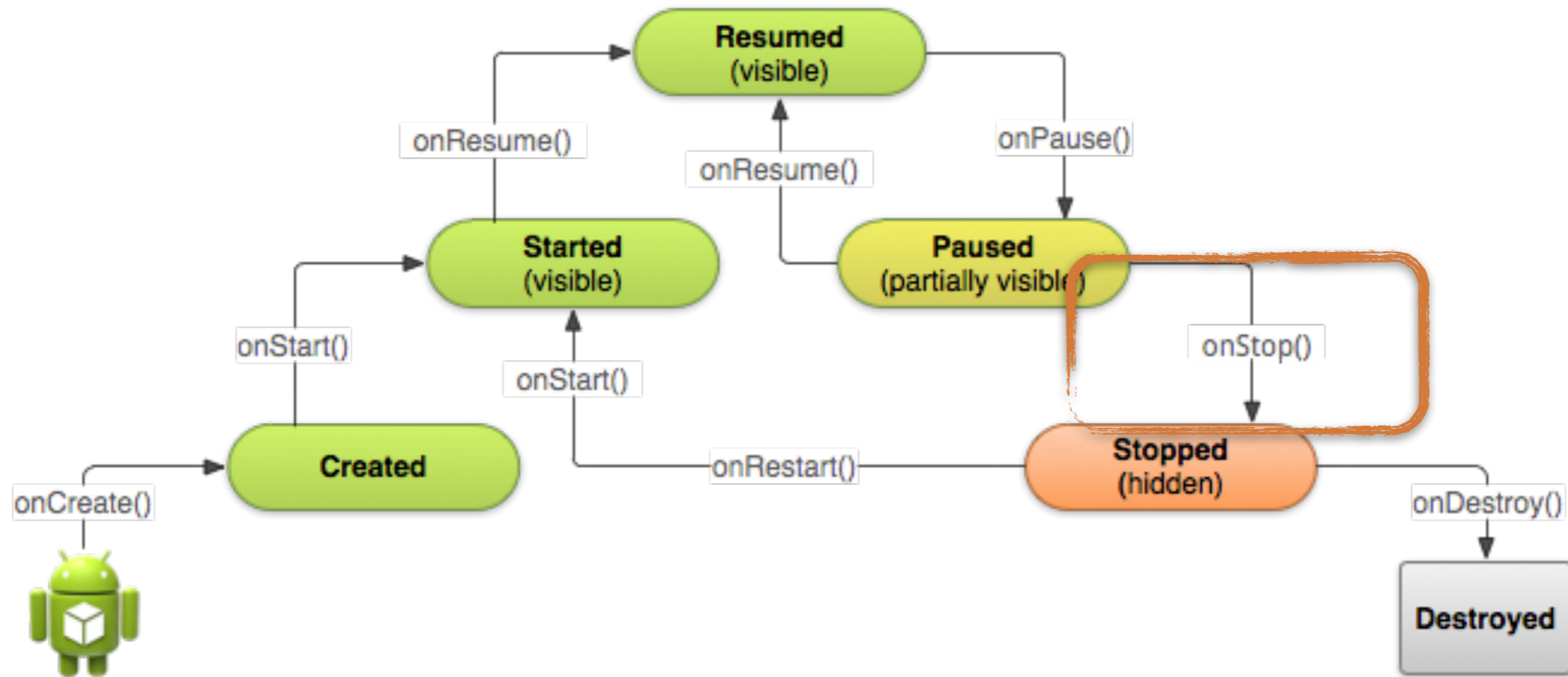
Activity Lifecycle



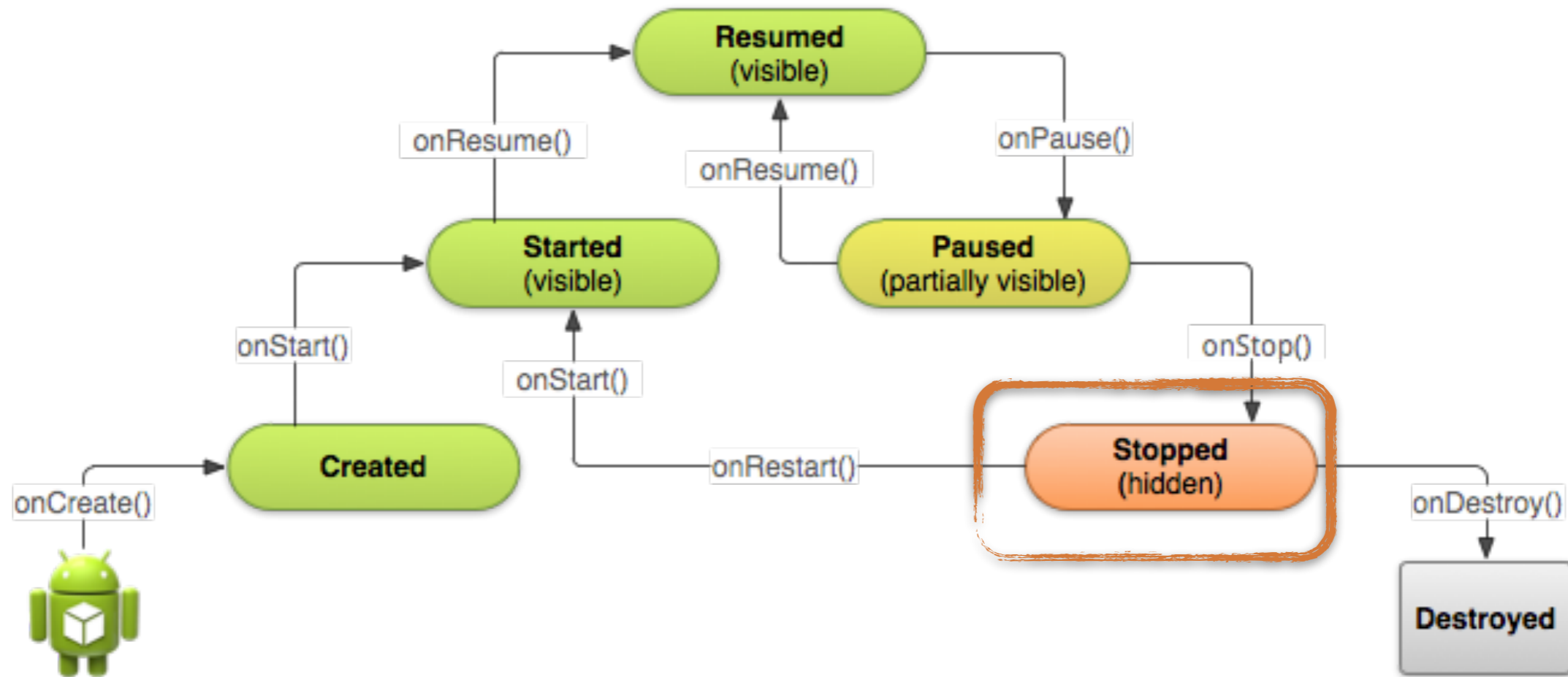
Activity Lifecycle



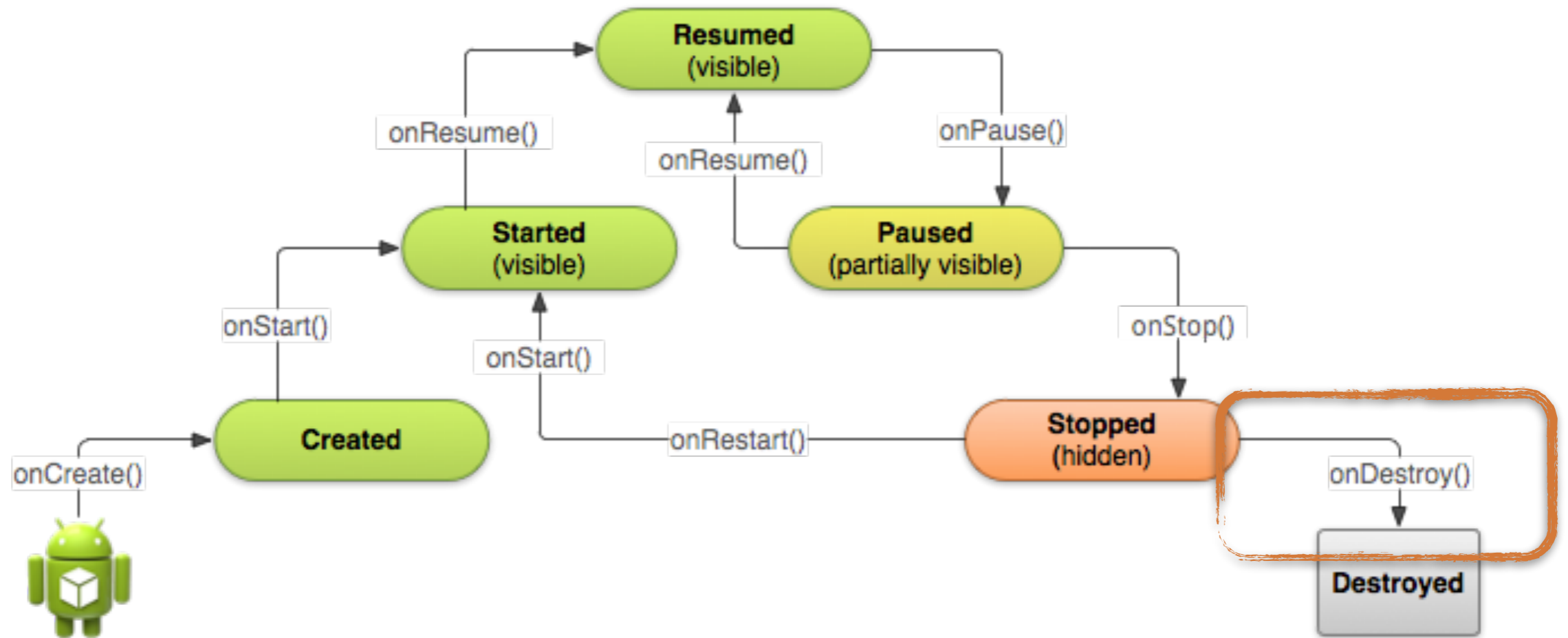
Activity Lifecycle



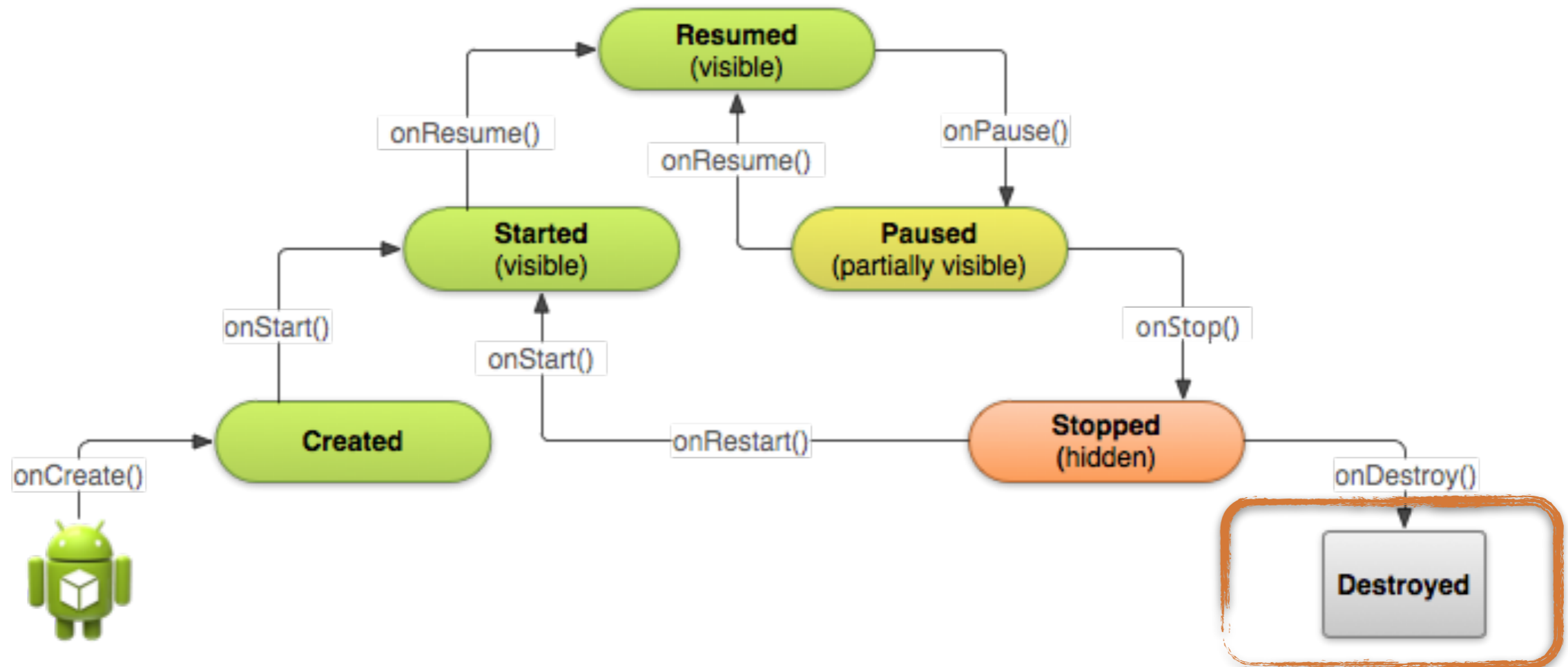
Activity Lifecycle



Activity Lifecycle



Activity Lifecycle



Activity Lifecycle

The screenshot shows an IDE window for an Android application. The title bar indicates the file is MainActivity.java in a project named GPSDrawApp. The breadcrumb navigation shows the path: GPSDrawApp > app > src > main > java > ics163 > luci > ics > uci > edu > gpsdrawapp > MainActivity. The Structure view on the left shows the MainActivity class with methods onCreate, onCreateOptionsMenu, and onOptionsItemSelected. The main editor displays the following Java code:

```
package ics163.luci.ics.uci.edu.gpsdrawapp;

import ...

public class MainActivity extends Activity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

The Manifest



The Manifest

The screenshot shows an IDE window with the following details:

- Title Bar:** AndroidManifest.xml - [app] - GPSDrawApp - [~/Documents/ClassResources/2015_03_ICs163/codeWorkspace/GPSDrawApp]
- Toolbar:** Includes icons for file operations (save, copy, paste, search) and development actions (run, debug, stop, refresh).
- Breadcrumbs:** GPSDrawApp > app > src > main > AndroidManifest.xml
- Project Structure:**
 - app
 - manifests
 - AndroidManifest.xml
 - java
 - ics163.luci.ics.uci.edu.gpsdrawapp
 - MainActivity
 - ics163.luci.ics.uci.edu.gpsdrawapp (android)
 - res
 - drawable
 - layout
 - activity_main.xml
 - menu
 - mipmap
 - values
 - Gradle Scripts

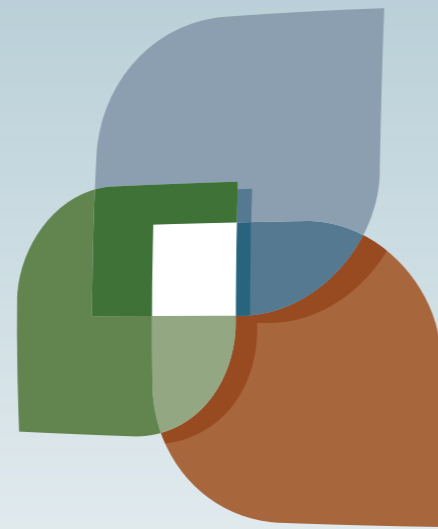
- Code Editor:** Displays the content of `AndroidManifest.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ics163.luci.ics.uci.edu.gpsdrawapp" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="GPSDrawApp"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="GPSDrawApp" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```



L U C I

