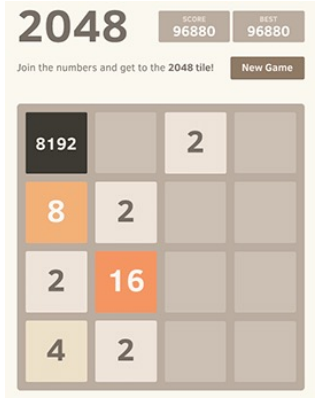
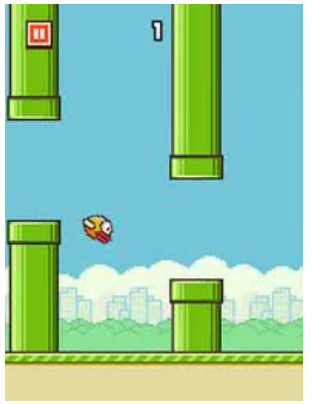
Android Tutorial

陈东尧，2021-03-29

Why do you need to learn Android Programing

- It's fun
  - ***Fast*** prototyping of your idea
- It's useful
  - Smartphones have ever-improving sensing, computing, and communication capabilities
- It's easy
  - No need of extra hardware (everyone has a smartphone)
  - Popular language (Java and some C++)
- (Maybe) A easy way to make money!
  - Publish your app and (possibly) let millions use your app



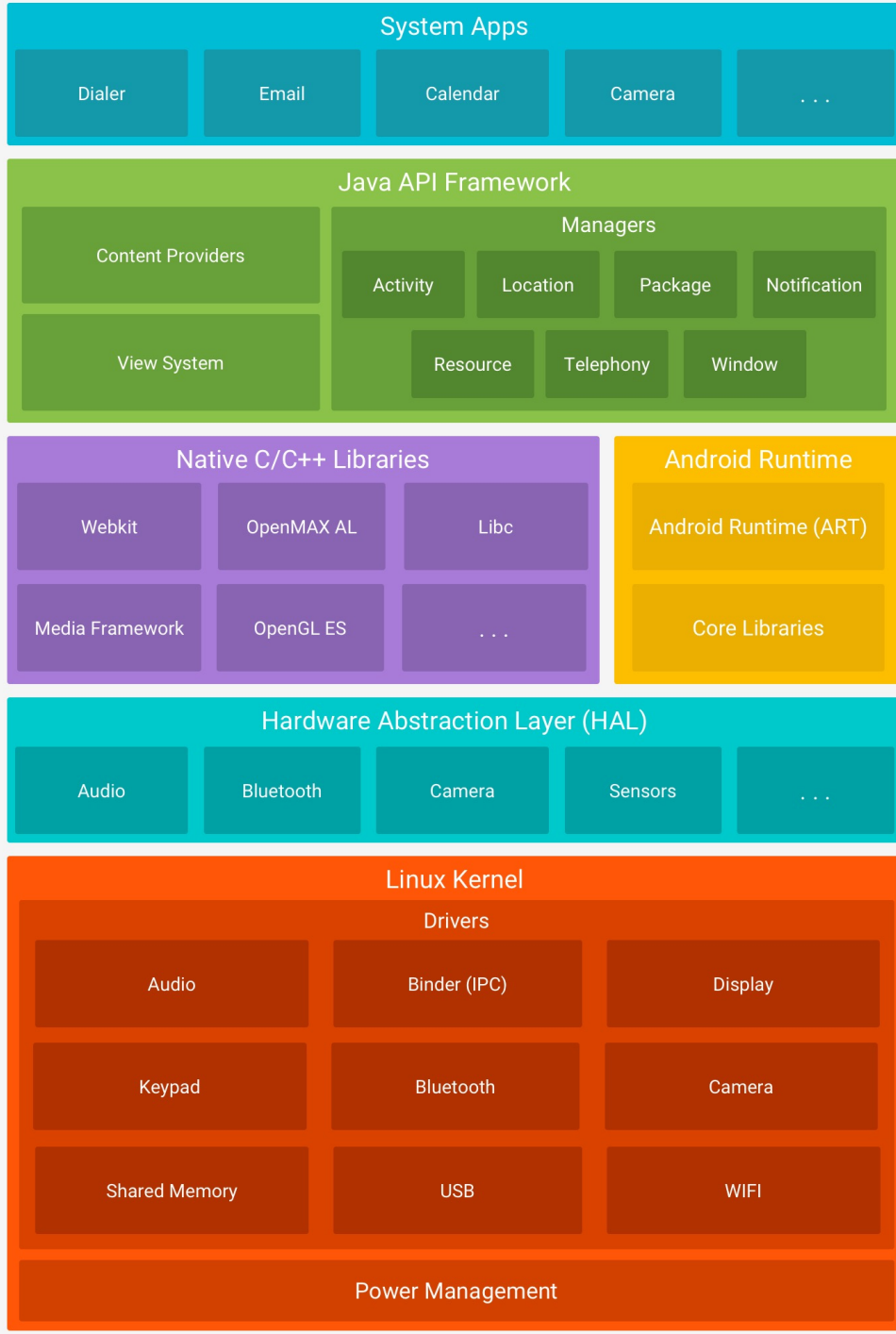


Today's Agenda

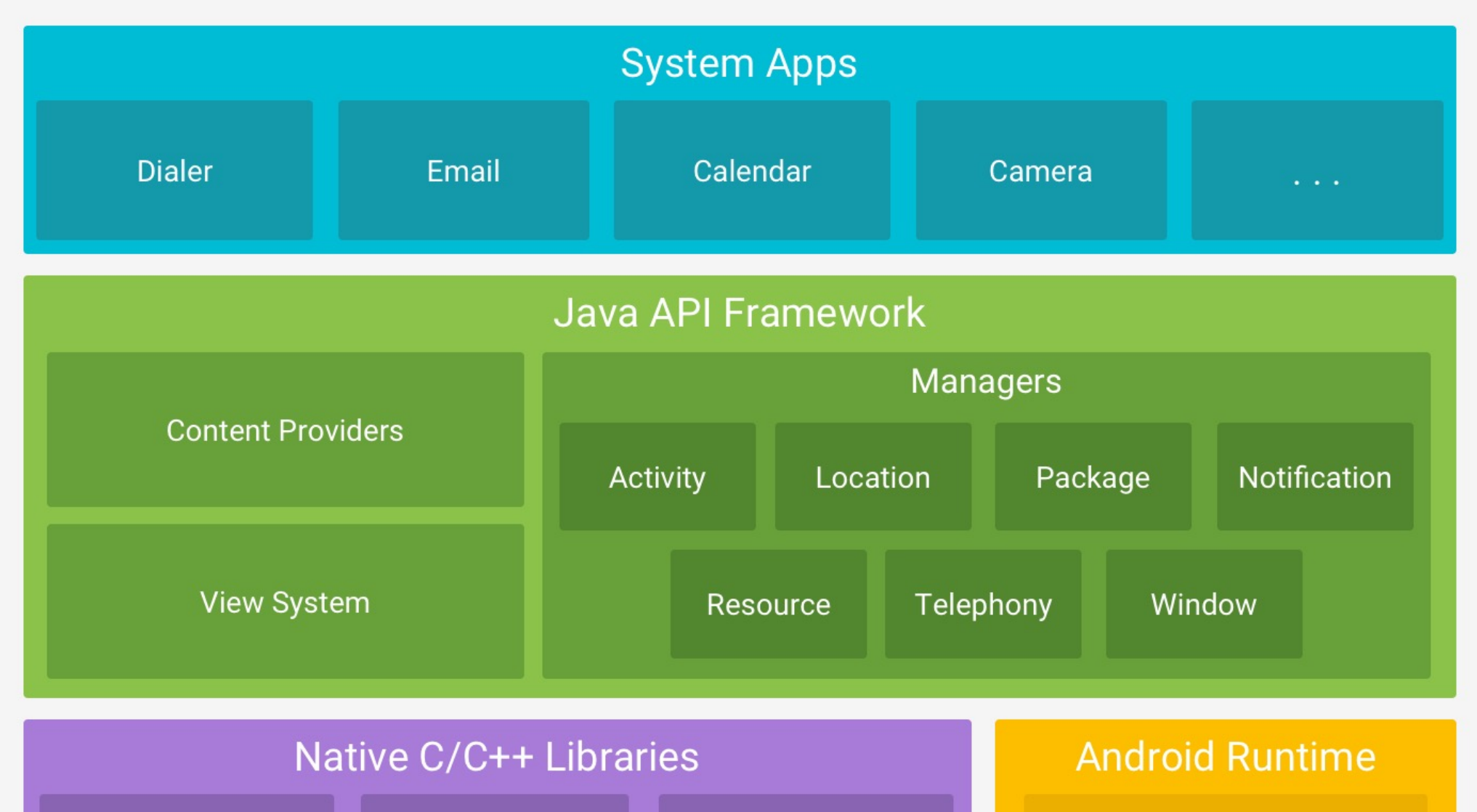
- Android Fundamentals
  - Platform Architecture
- Development principles
  - Your ***first*** Android app
  - Fundamental components of Android App
  - Life Cycle of Android Activity
- Learn by examples
  - Turn signal detection
  - Android sensor API analysis
- Advanced topics
  - Android vs. iOS
  - Cross-platform development

Platform Architecture

- Android is an open source, Linux-based software stack created for a wide array of devices and form factors.



Platform Architecture



Your First Android App

- Get familiar with Android studio
- Overall Android development process
  1. Create app
     - Java files + resource files (for UI design)
  2. Test app
     - Build the *.apk file
     - Install and test on device
  3. Publish the app

Fundamental Components

Component	Description
Activty	Deals with UI aspects. Typically corresponds to **a single screen**
Service	Background tasks (e.g., play music in background while user is web surfing). Typically have no UI interaction.
BroadCastReceiver	Receive messages, e.g., “Low battery” from system/apps and
ContentProvider	Provide an interface to app data. Lets apps share data with each other

“HelloWorld” Coding Demo

- Print “Hello World!” on the screen
- Interact with the UI and change the text.

Example

- MyActivity.java

```
package com.chendy.helloworld1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private TextView text_example;

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

        text_example = (TextView)findViewById(R.id.text_example);
        Button clicky = (Button) findViewById(R.id.clicky);

        clicky.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //code
                text_example.setText("Clicked!");
            }
        });
    }
}
```

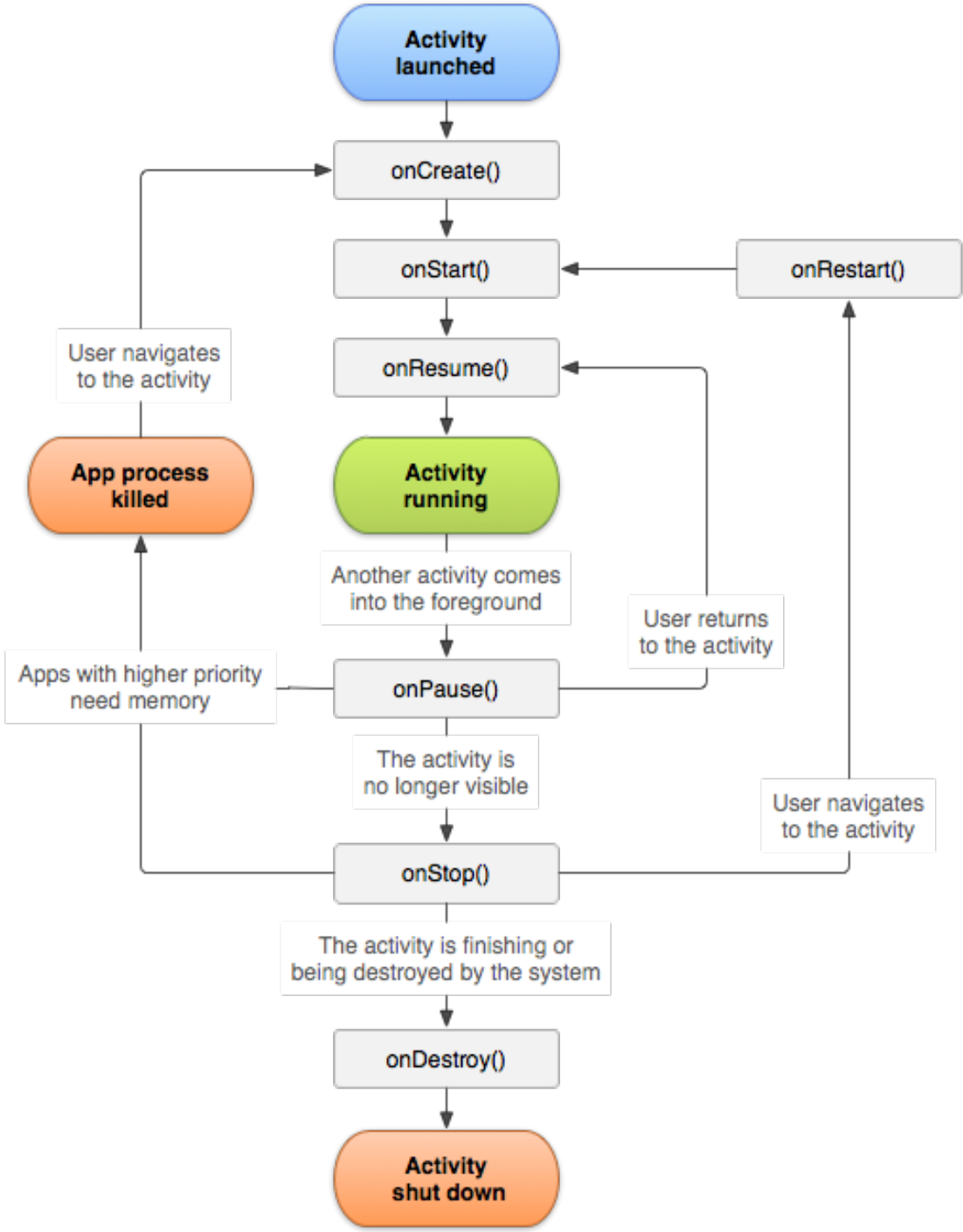
Example

- activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/text_example"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/clicky"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="Click me!" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Understand the Lifecycle of Android Activity

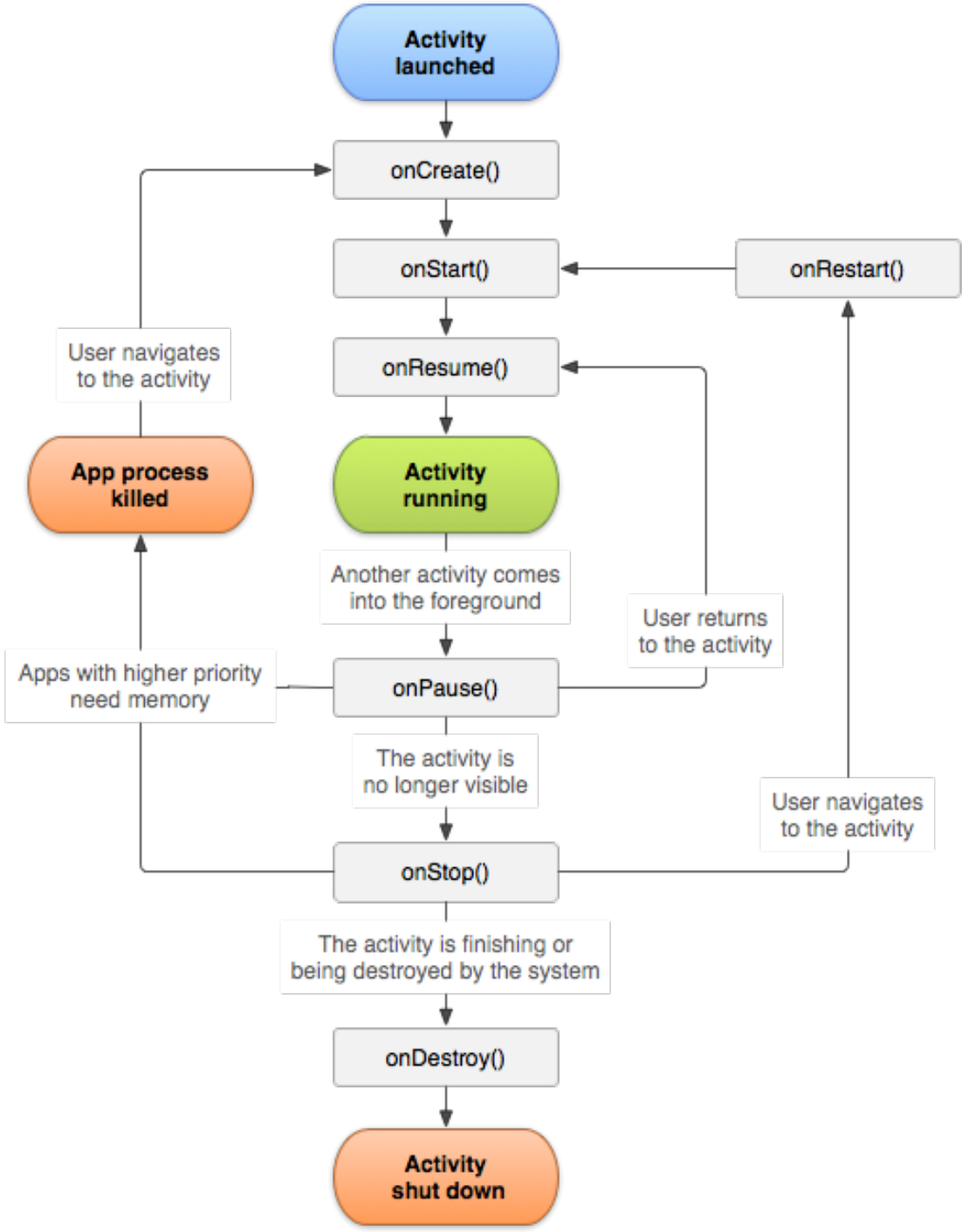
- onCreate()
- onStart()
- onResume()
- onPause()
- onStop()
- onDestroy()



These 6 core activity are implemented as callbacks, since each step may take indefinite amount of time.

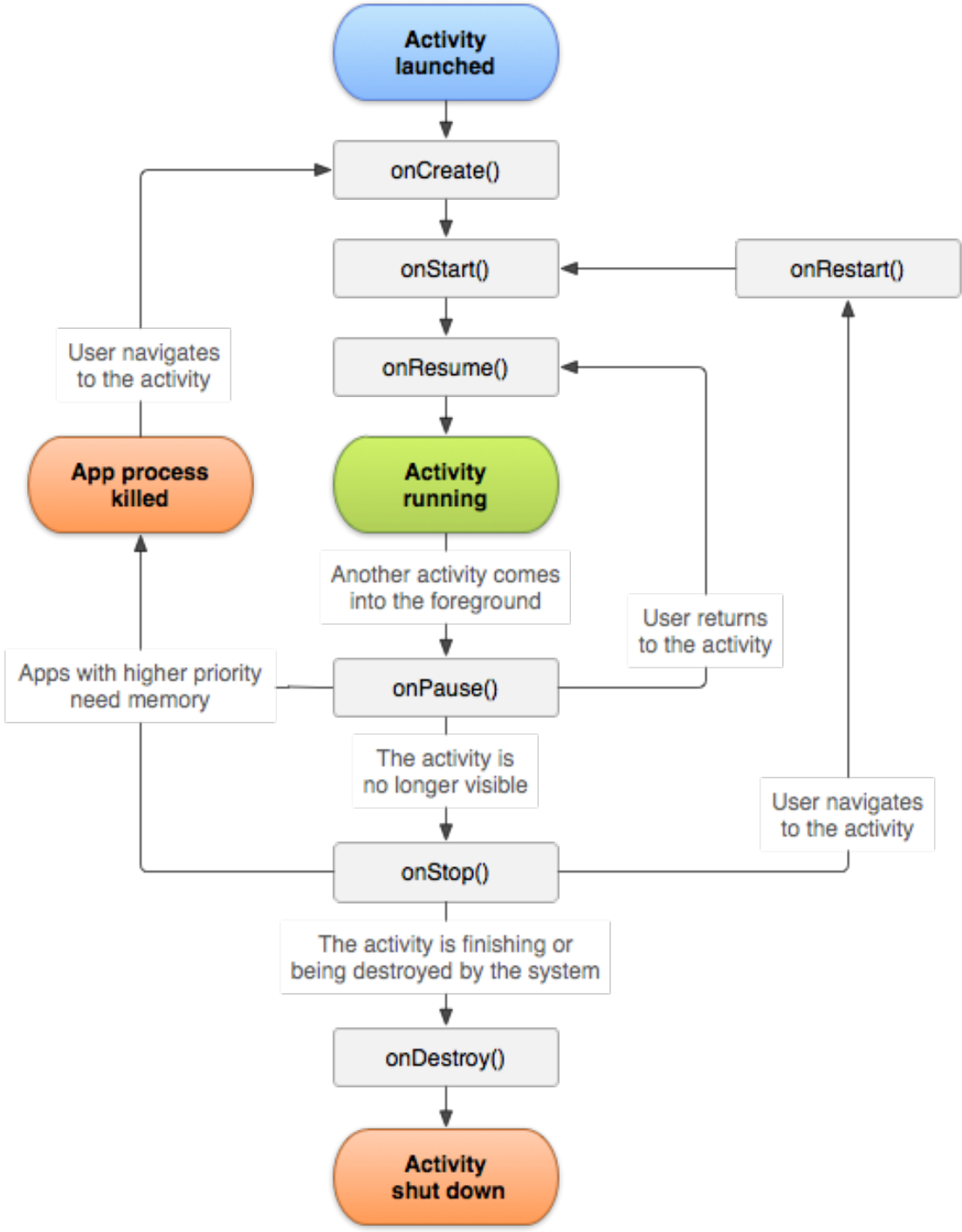
Understand the Lifecycle of Android Activity

- onCreate()
  - Fires when the system first creates the activity. Implement startup logic that should perform only once for the entire life of the activity.
- onStart()
- onResume()
- onPause()
- onStop()
- onDestroy()



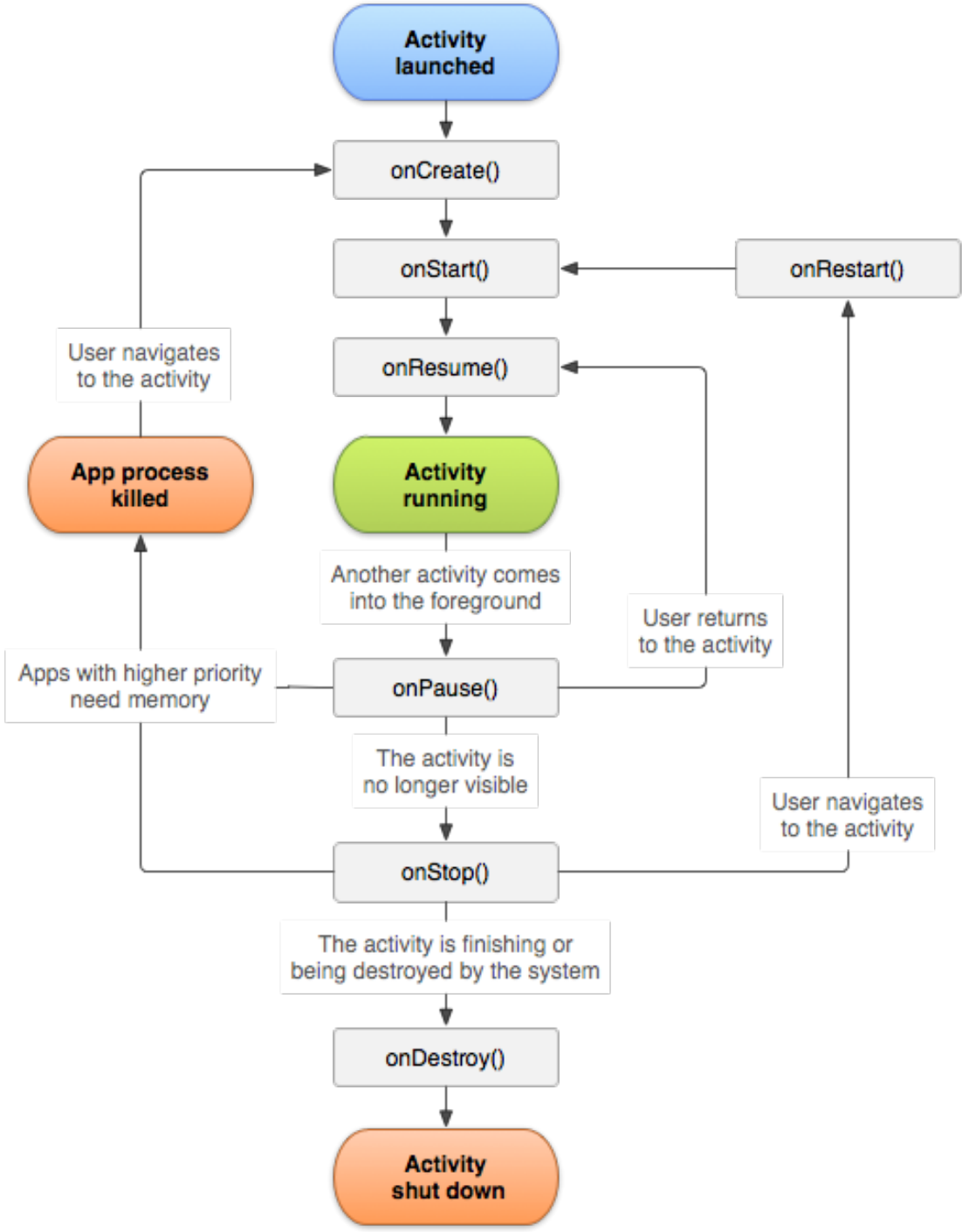
Understand the Lifecycle of Android Activity

- onCreate()
- onStart()
  - This call makes the activity visible to the user, as the app prepares for the activity to enter the foreground and become interactive. **Once finished (usually very fast),** the activity enters the Resumed state
- onResume()
- onPause()
- onStop()
- onDestroy()



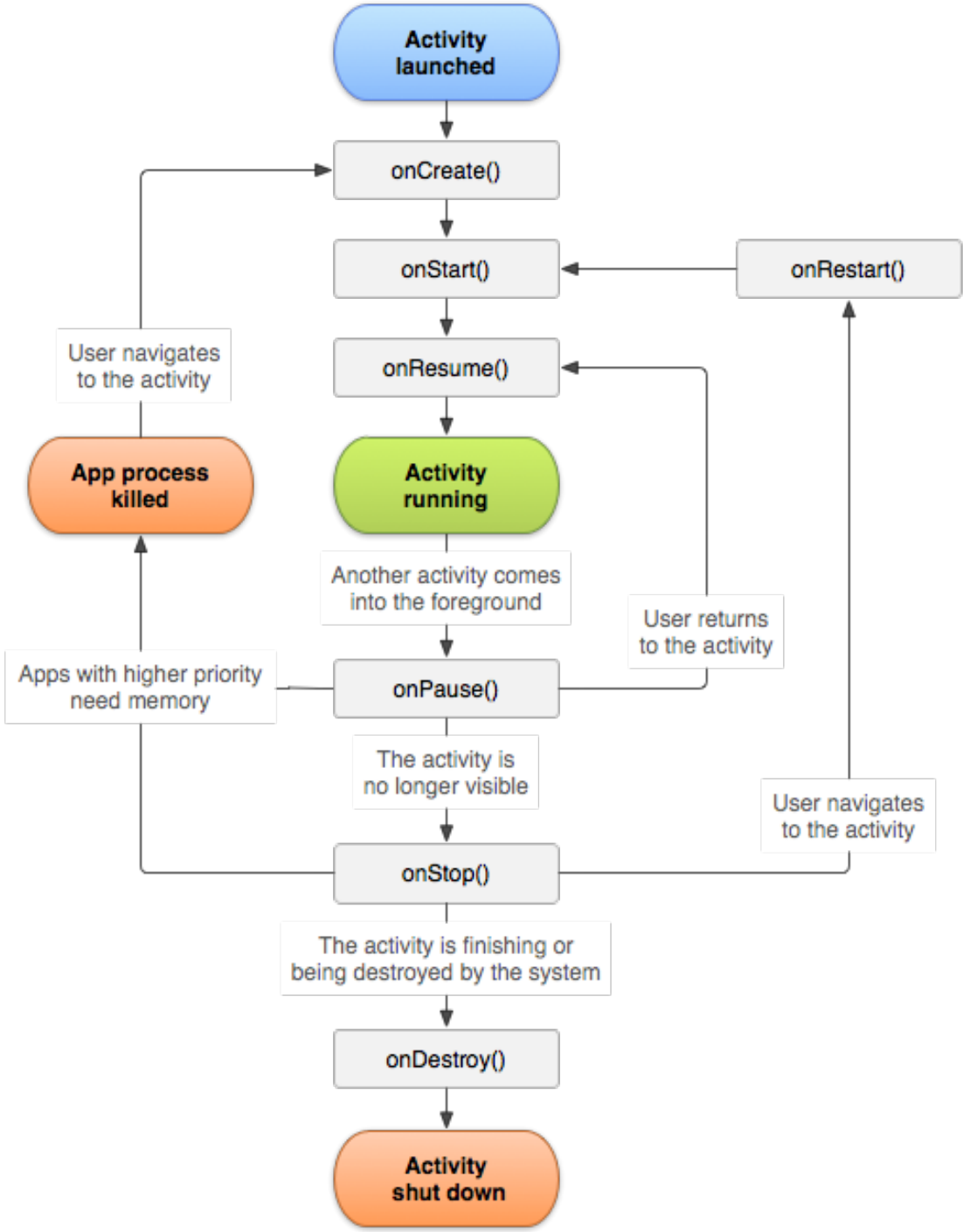
Understand the Lifecycle of Android Activity

- onCreate()
- onStart()
- onResume()
  - This is the state in which the app interacts with the user. The app stays in this state until **something happens to take focus away from the app.** Such an event might be, e.g., receiving a phone call, the user's going to another activity, or screen off.
- onPause()
- onStop()
- onDestroy()



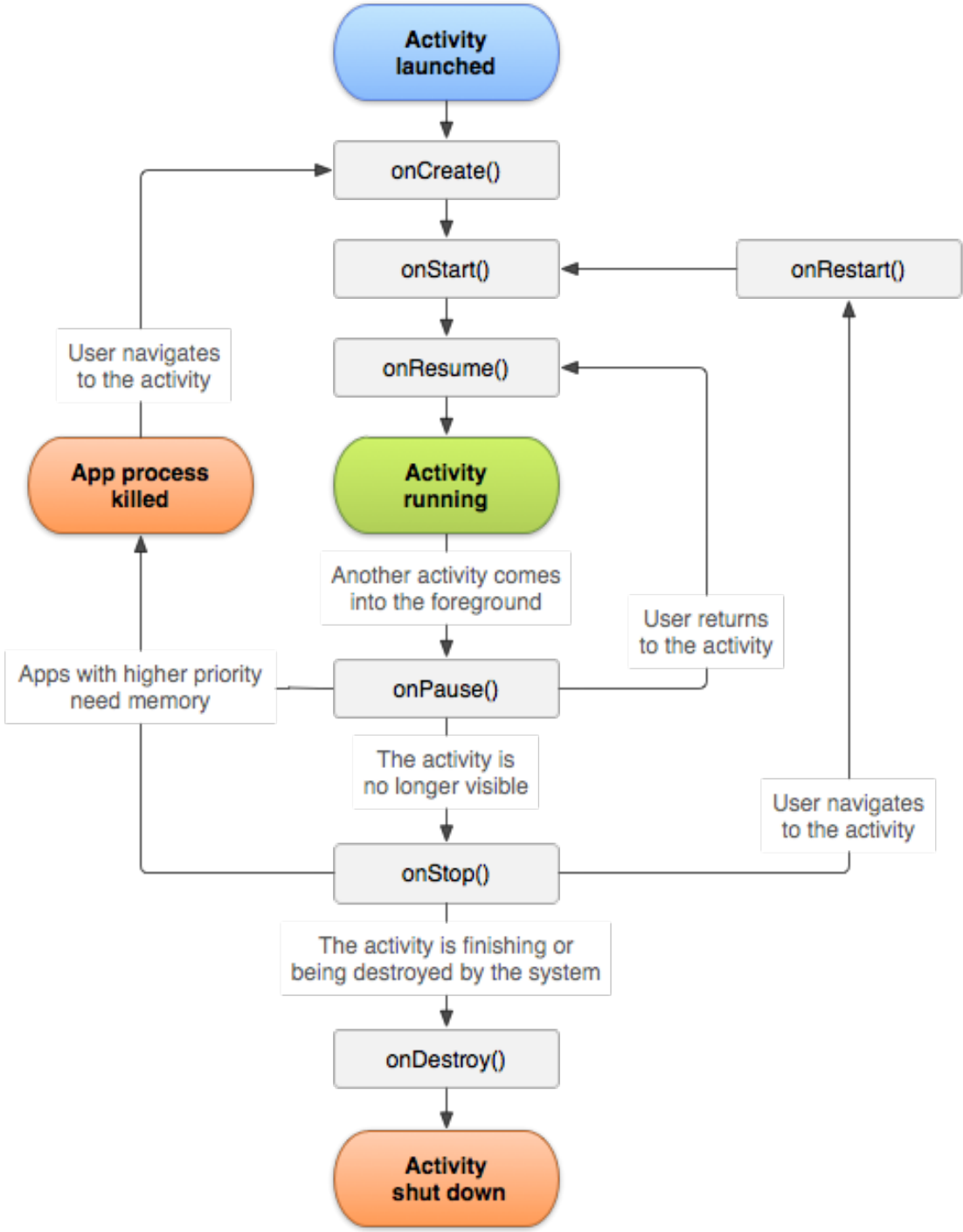
Understand the Lifecycle of Android Activity

- onCreate()
- onStart()
- onResume()
- onPause()
  - This call back used when the activity is no longer in the foreground but not killed yet (e.g., the activity is still visible if the user is in multi-window mode).
  - This method can be used for releasing system resources.
  - Note, onPause() execution is very brief, **time consuming operations (e.g., saving data) needs to perform in onStop()**
- onStop()
- onDestroy()



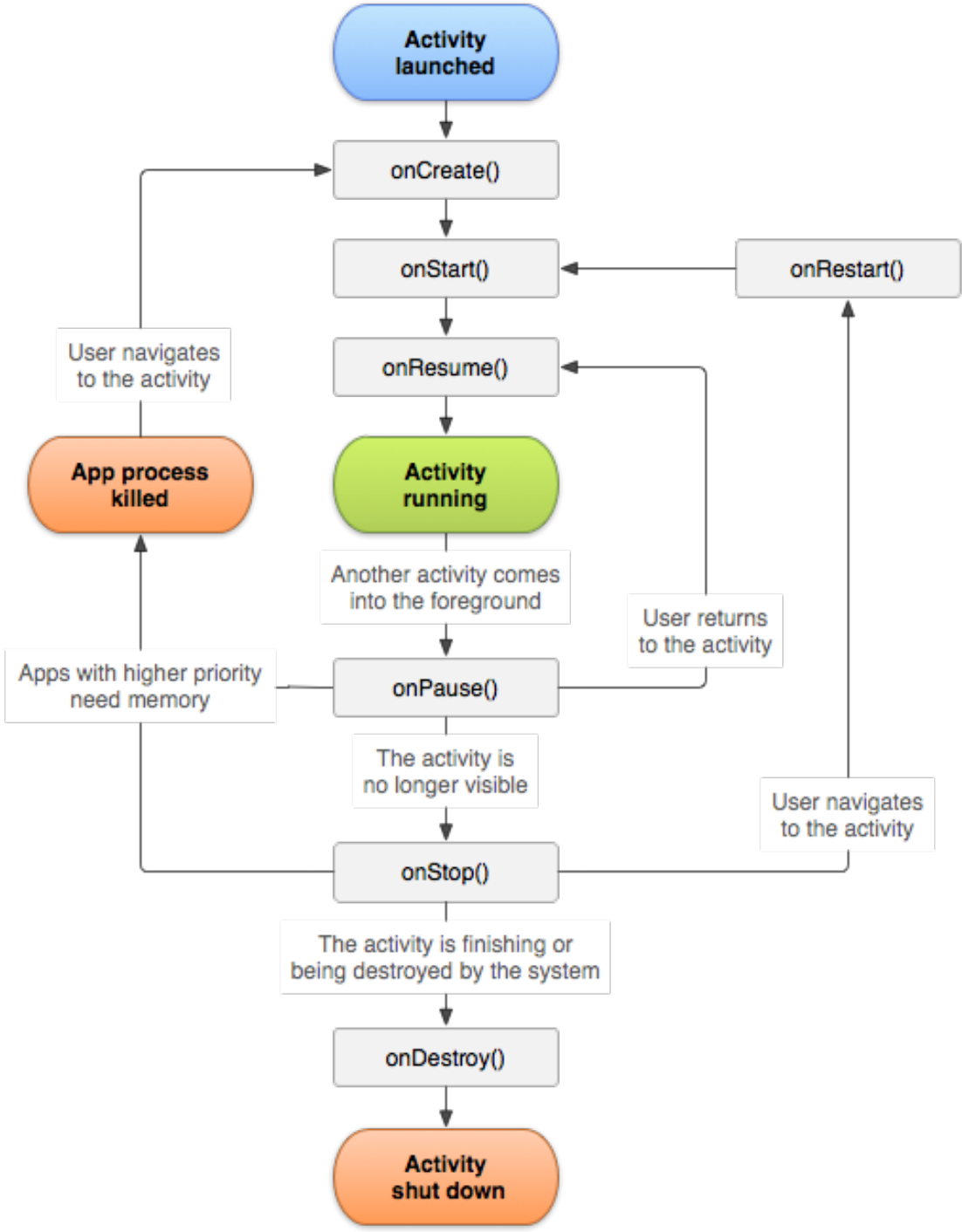
Understand the Lifecycle of Android Activity

- onCreate()
- onStart()
- onResume()
- onPause()
- onStop()
  - System invoke this callback when the activity is no longer visible to the user.
  - Should release or adjust resources while the app is **not visible to user**. For example, switch from fine-grained to coarse-grained location updates.
  - Note, once stopped, the system may destroy the process to recover memory.
- onDestroy()



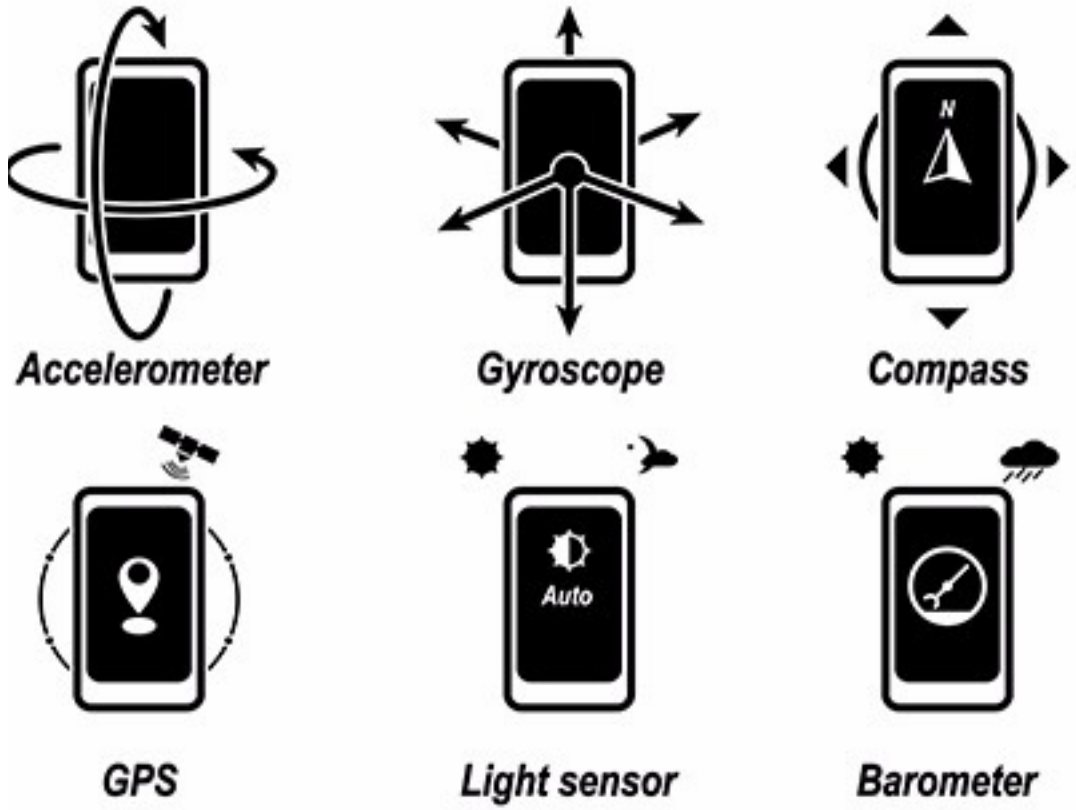
Understand the Lifecycle of Android Activity

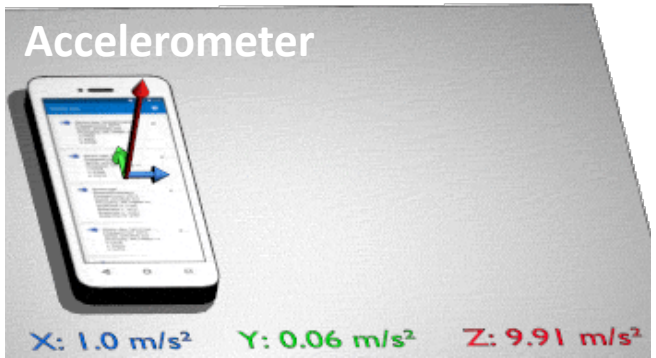
- onCreate()
- onStart()
- onResume()
- onPause()
- onStop()
- onDestroy()
  - Is called when: the activity is finishing

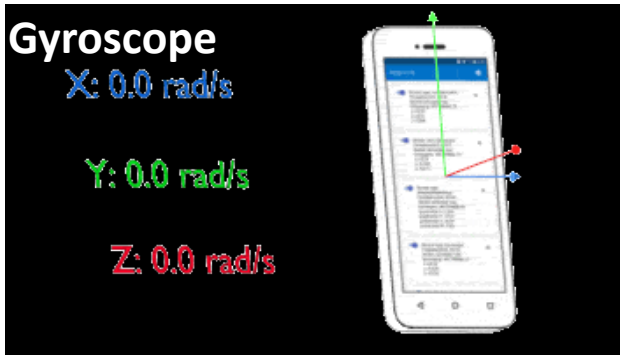


Sensing Tool

- The key aspect of **IoT** is **automatically understand its surrounding environment**.
- Smartphone sensors



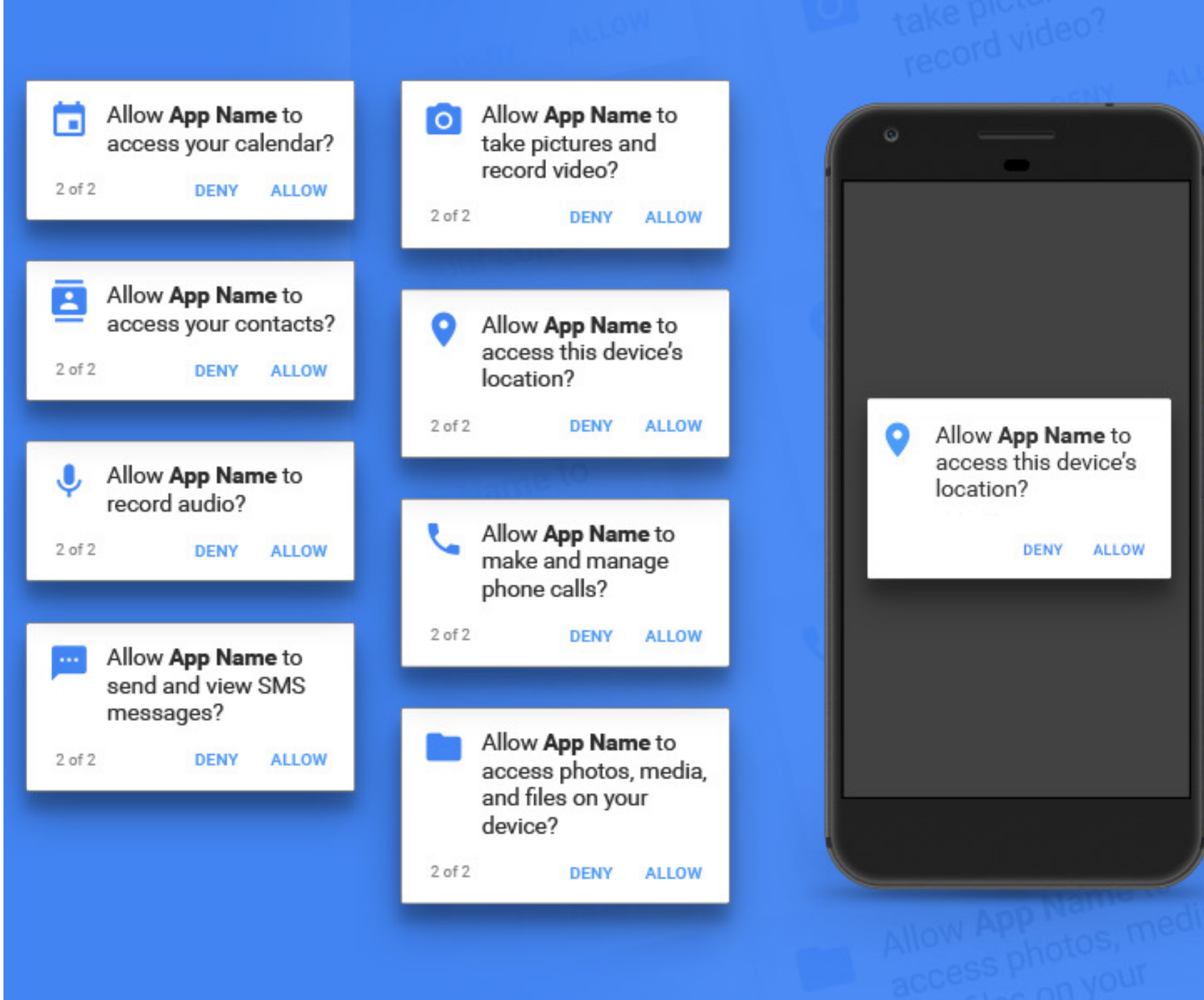




“Location101” Coding Demo

- Read the location data (longitude and latitude) in real-time

Permission Model

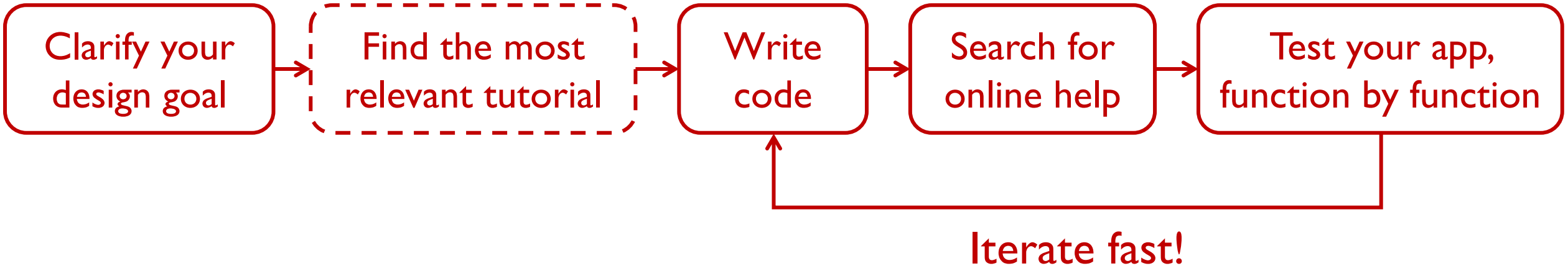


A Real-world Problem

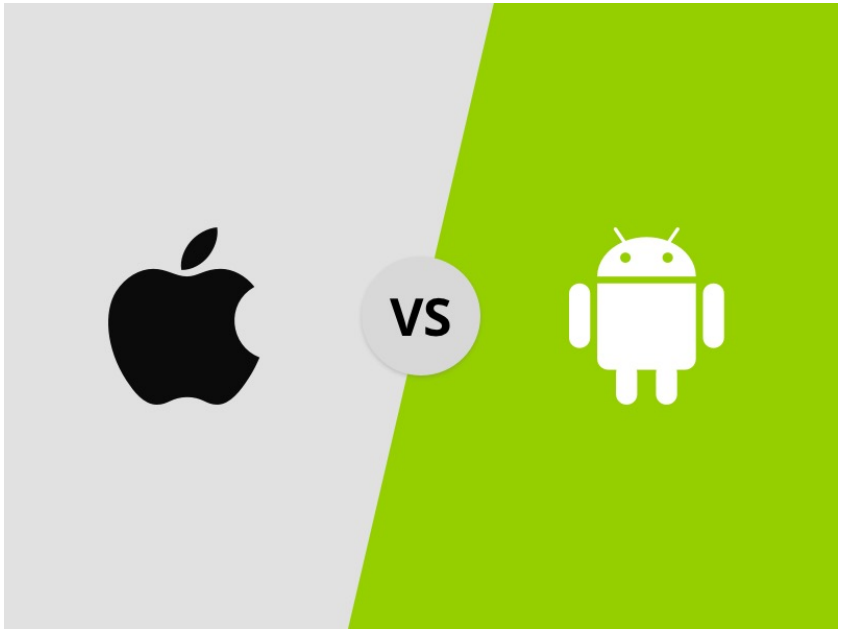
- Can we build an app for better turn signal analysis?
- Tackle this question
  - How to detect turn signal usage?
  - What sensors do we need?
  - How to design the app structure?

Video Demo

Learn to Code by Examples



Compared to iOS Development



- Open source! Open source!! Open source!!!

Thanks!

- 陈东尧 https://chendy.tech/
- Today’s code