

# Android 사전세션

AWS IAM

AUSG 2기, 김예본

# 발표자 소개



- 고려대학교 컴퓨터학과
- 안드로이드 개발
- Java, Python
- 前 dodotdo 안드로이드 개발자
- SW Maestro 8기
- AUSG 2기 오거나이저

# 오늘 함께 할 내용

## Android 기초 개념

- **Android 프로젝트 구조**
- **Android 4대 컴포넌트**
- **Android Lifecycle**
- **예시 소스**

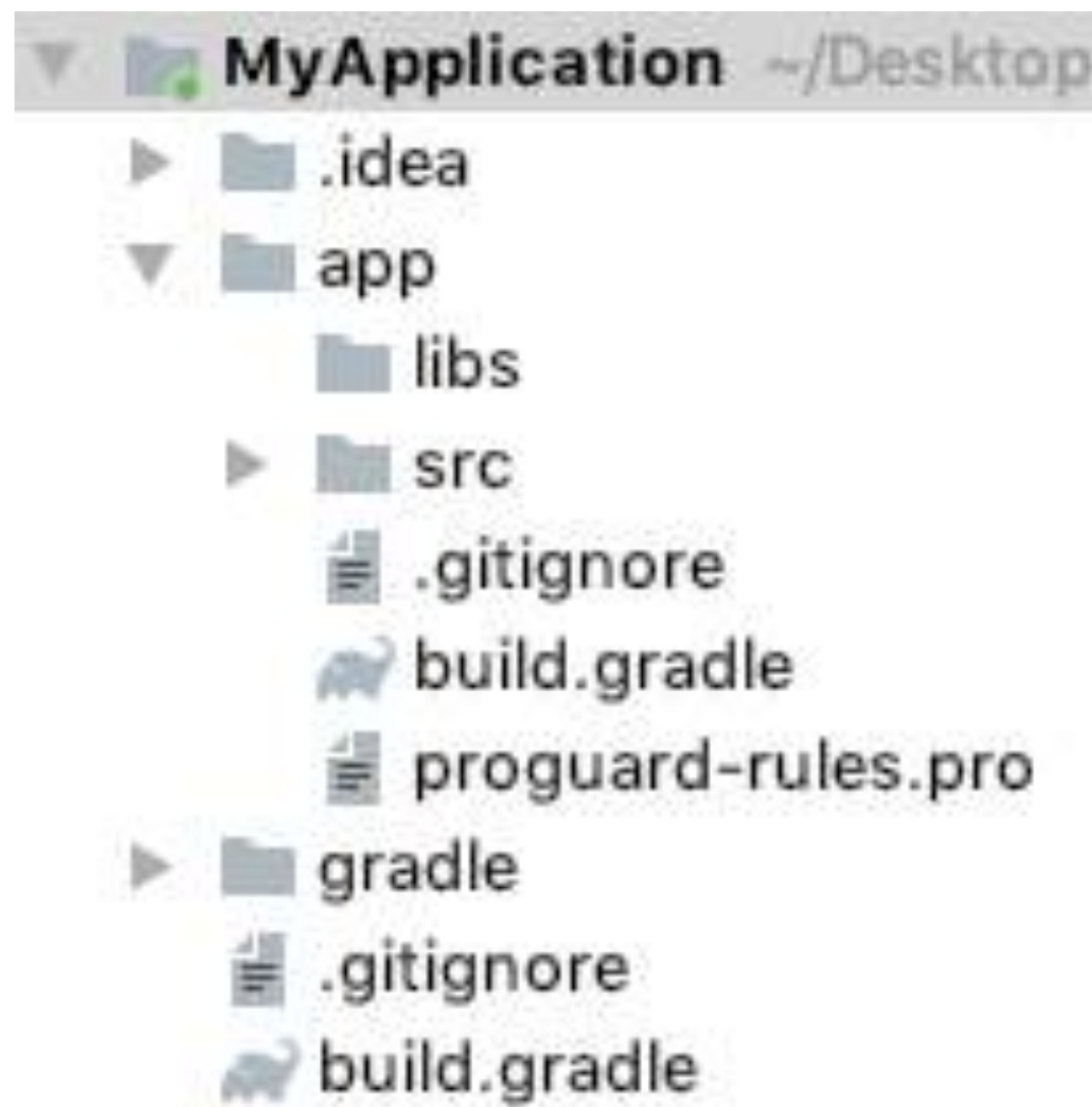
## AWS 기초 개념

- **AWS의 필요성**
- **AWS IAM**

# Android 프로젝트 구조

# Android 프로젝트 구조

## 전체 구조



- **build.gradle**
- **app - src**
- **app - build.gradle**

# Android 프로젝트 구조

## Gradle



**Groovy를 이용한 빌드 자동화 시스템**

# Android 프로젝트 구조

## build.gradle

```
buildscript {  
    repositories {  
        google()  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:3.4.0'  
        // NOTE: Do not place your application dependencies  
        // in the individual module build.gradle files  
    }  
}  
  
allprojects {  
    repositories {  
        google()  
        jcenter()  
    }  
}
```

## 프로젝트 단위의 빌드 설정

- gradle 버전
- Repository 추가



# Android 프로젝트 구조

## app - build.gradle

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

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.0.2'
```

## 모듈 단위의 빌드 설정

- **compileSdkVersion**
- **minSdkVersion**
- **versionCode**
- **dependencies**



# Android 프로젝트 구조

app - src



- **androidTest** - UI 테스트, 기기필요
- **main** - 실제 프로젝트
- **test** - JUnit 테스트

# Android 프로젝트 구조

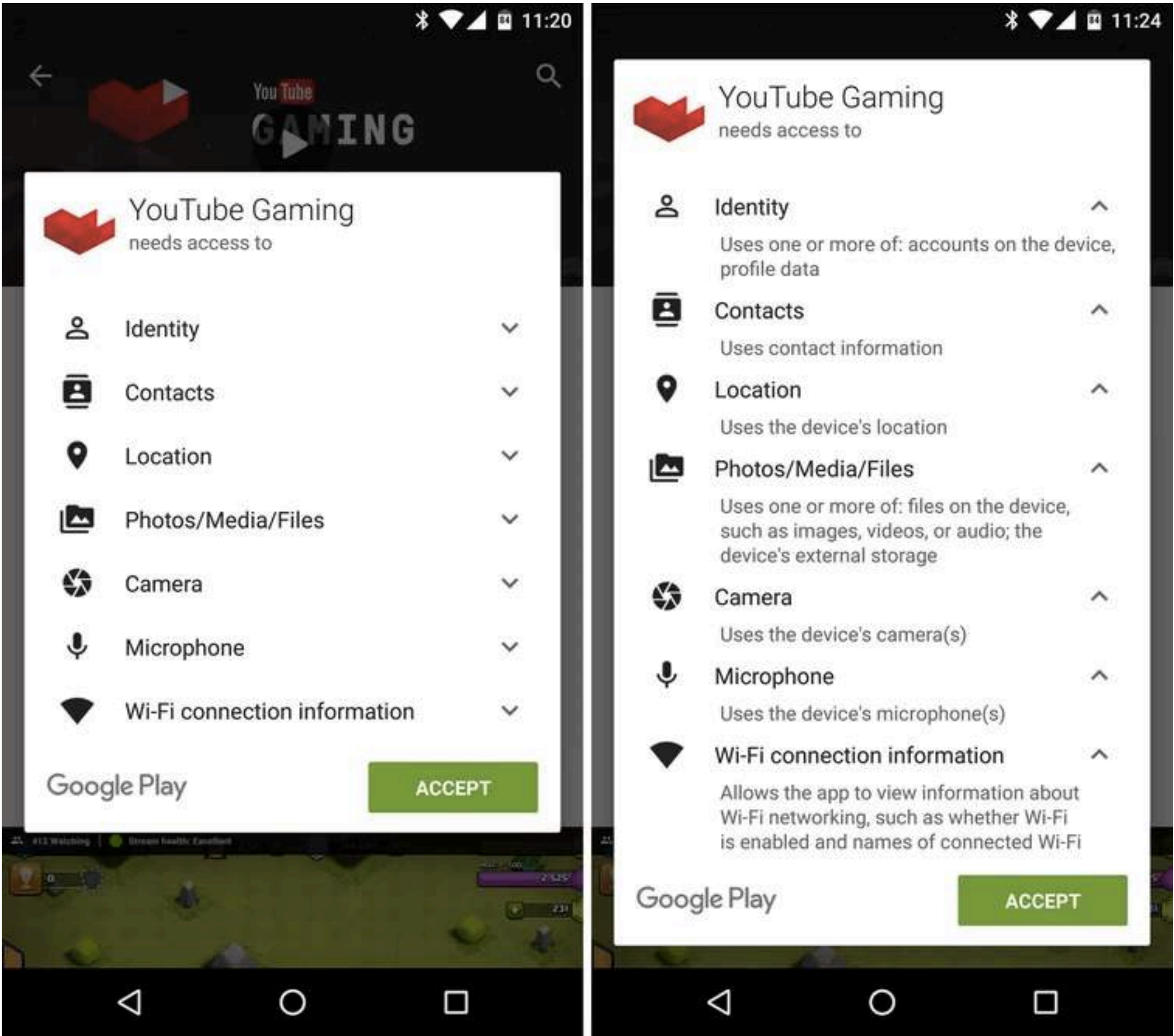
## app - src - main



- java - 기능관련, java 파일들
- res - 프로젝트에 필요한 resources
  - drawable - 이미지, xml 파일들
  - layout - 레이아웃, xml 파일들
  - mipmap - 앱 아이콘
    - colors - 색상 코드
    - strings - 문자열들 국가별 설정가능
    - styles - 뷰에 공통적으로 사용할 styles
- AndroidManifest - component 설정, 권한 설정

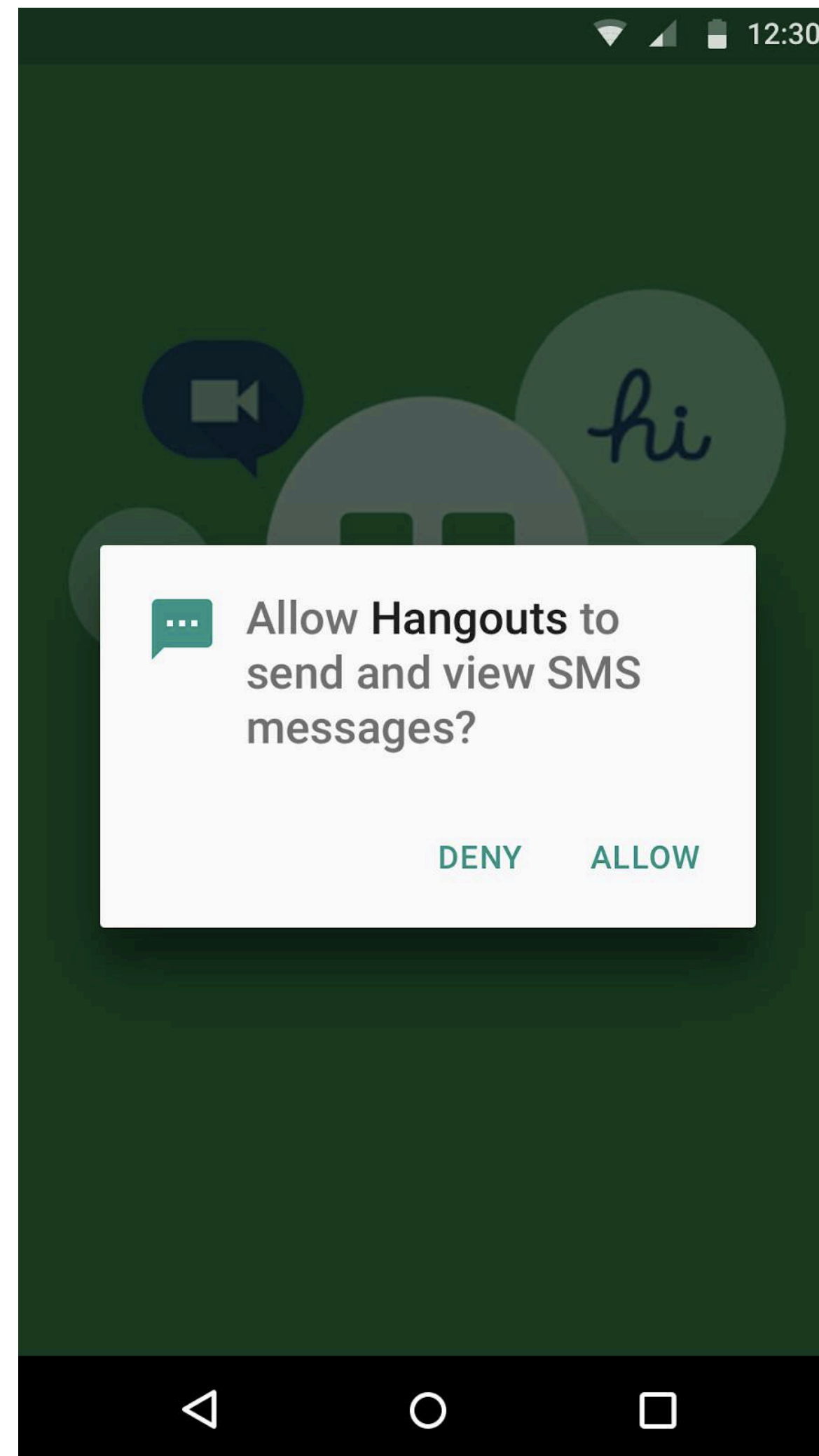
# Android 프로젝트 구조

## App permission



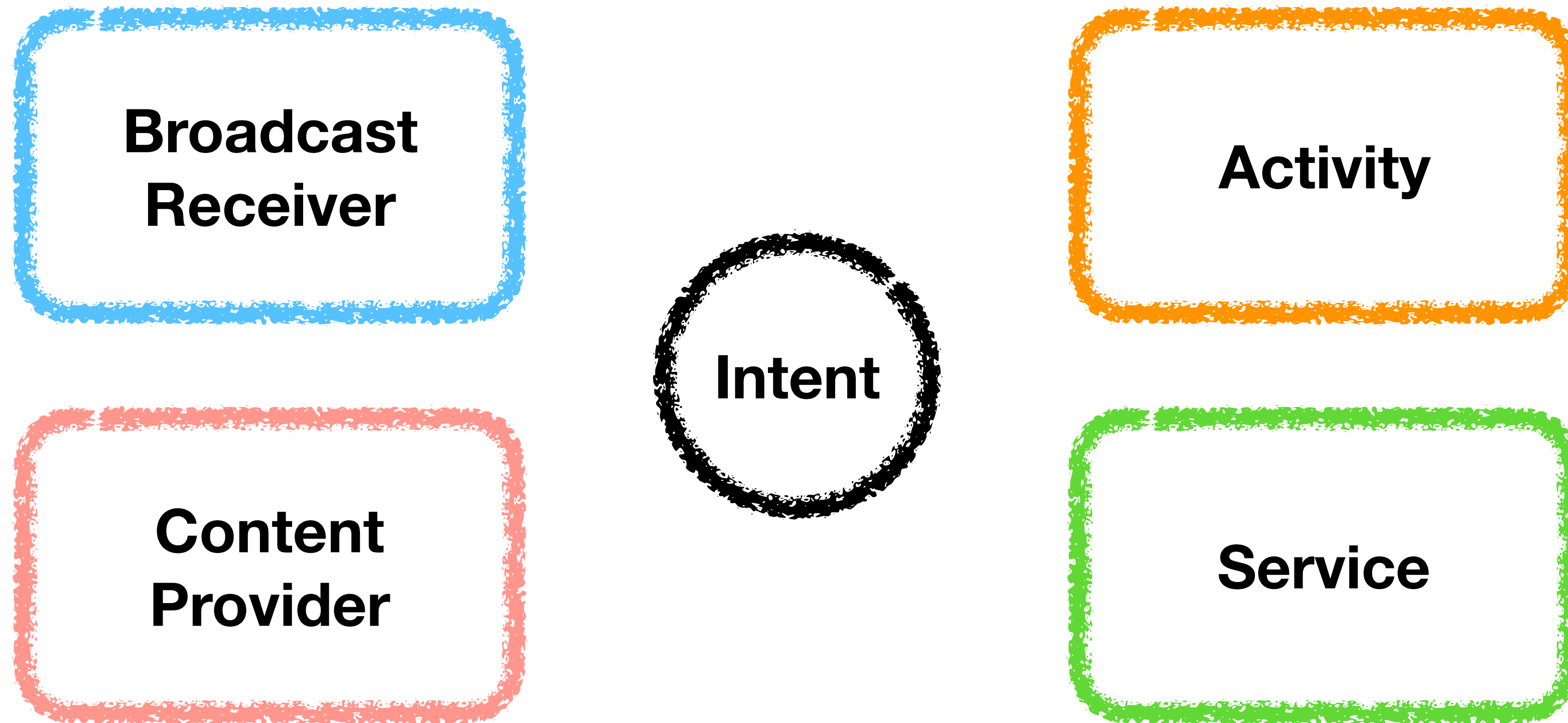
# Android 프로젝트 구조

## App Permission Request



# Android 4대 컴포넌트

# Android 4대 컴포넌트





# Android 4대 컴포넌트

## Activity

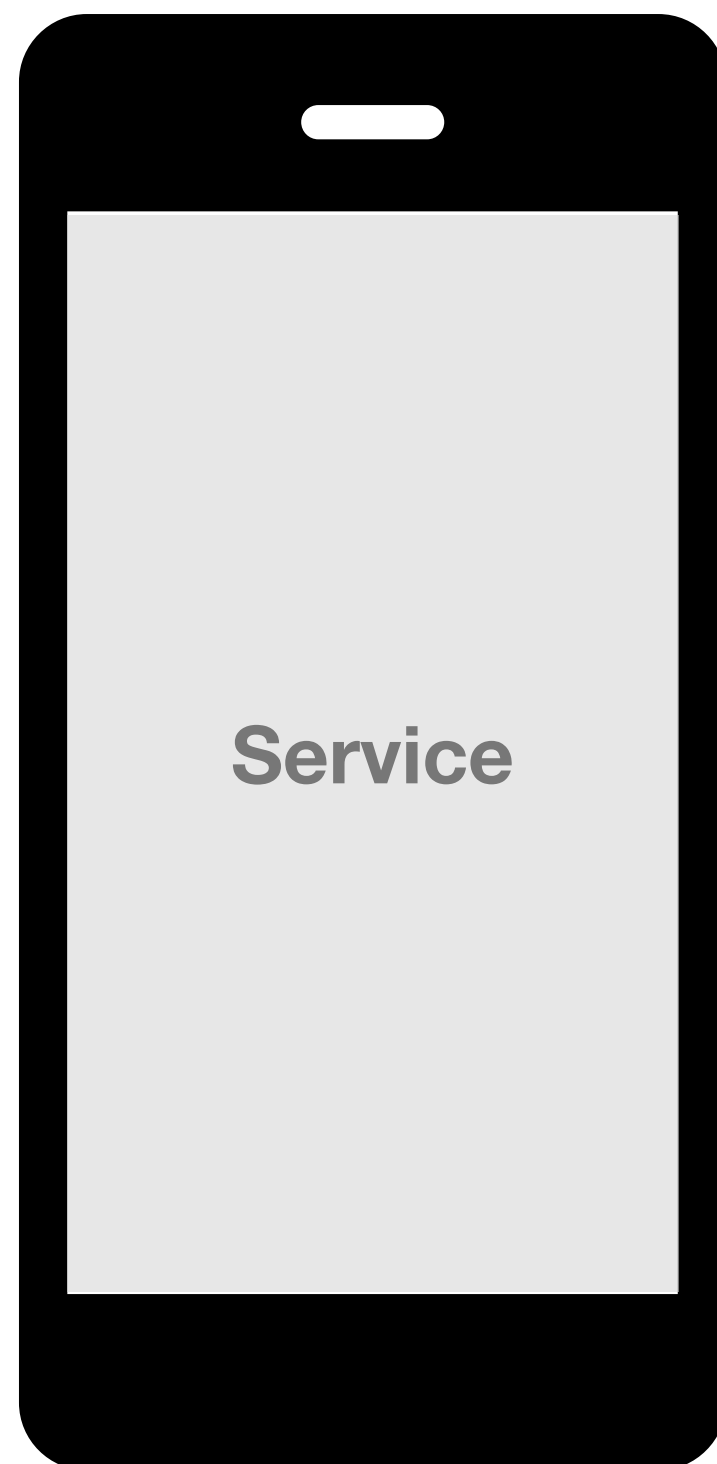


안드로이드 어플리케이션 개개의 화면

- “Main” Activity가 존재
- 보이지 않고 이전에 실행되었던 화면은 **Back Stack**에 저장
  - Back 버튼으로 Pop가능
- Activity LifeCycle 존재

# Android 4대 컴포넌트

## Service



### 안드로이드 어플리케이션 Background 구성 요소

- 화면 없이 핸드폰 Back에서 수행하는 작업
  - ex> 알람, 음악 플레이어
- Foreground 서비스도 존재
- 잠자기 모드로 인하여 WorkManager로 대체되는 중
- Service LifeCycle 존재

# Android 4대 컴포넌트

## 잠자기(Doze) mode



악성 서비스



웹사이트

# Android 4대 컴포넌트

잠자기(Doze) mode



# Android 4대 컴포넌트

## 잠자기(Doze) mode

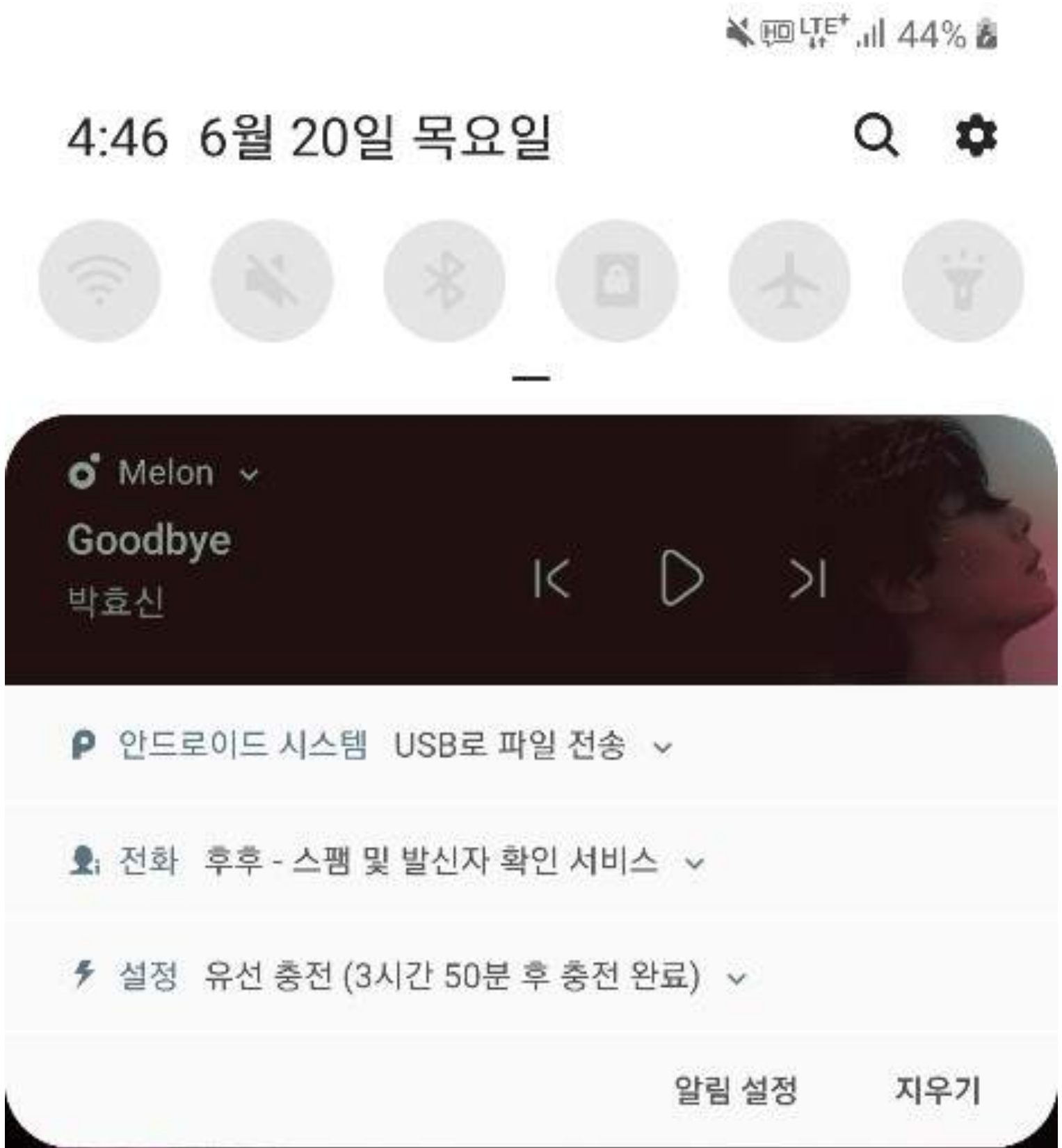


- 안드로이드 6.0부터 적용
- 네트워크, GPS 등 작업 불가
- 배터리가 충전 중이 아닐 때
- 핸드폰이 물리적으로 이동 중이 아닐 때
- 스크린 오프 상태에서 오랫동안 방치되었을 때

# Android 4대 컴포넌트

## Service

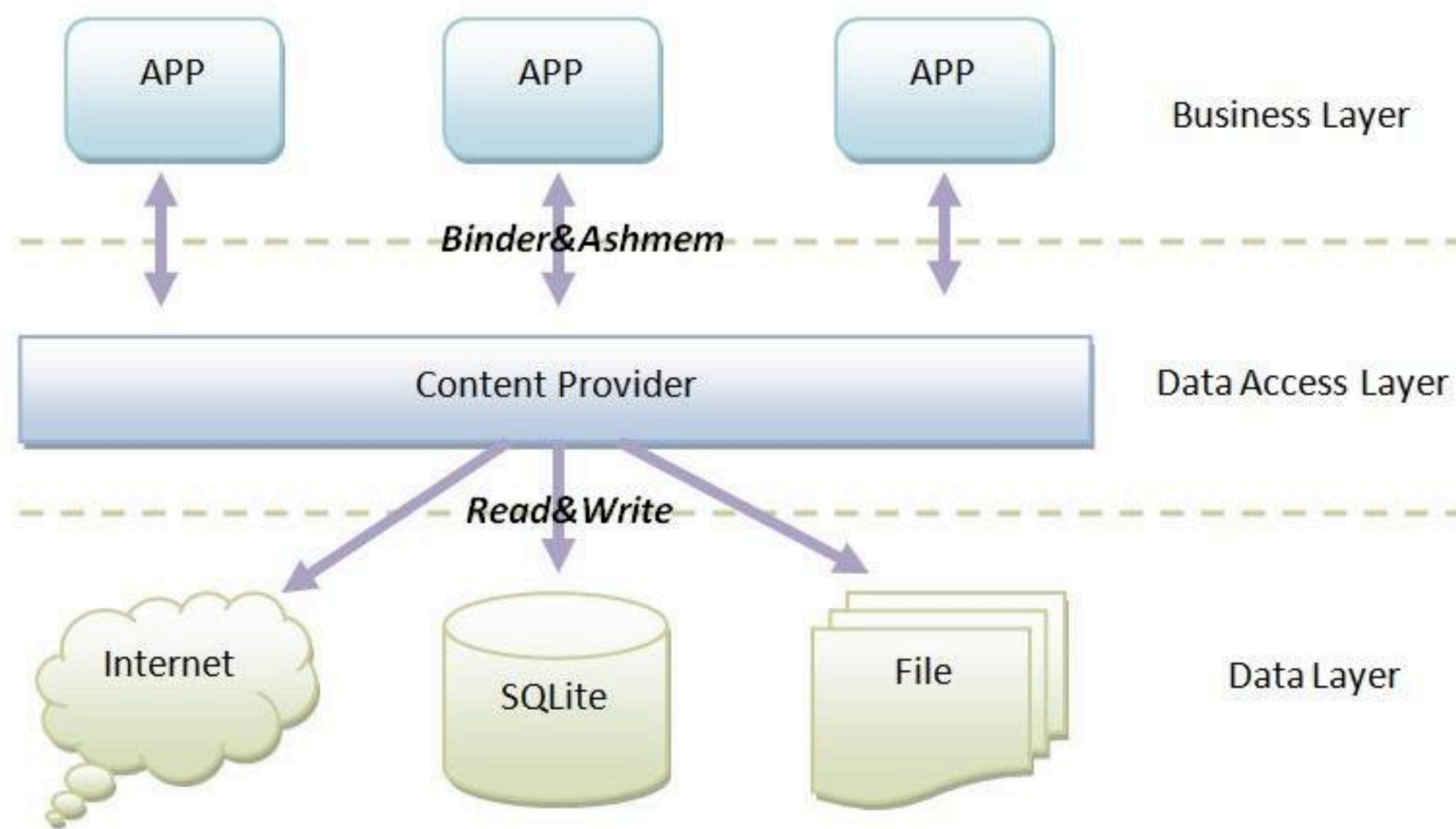
### Foreground 서비스





# Android 4대 컴포넌트

## Content Provider

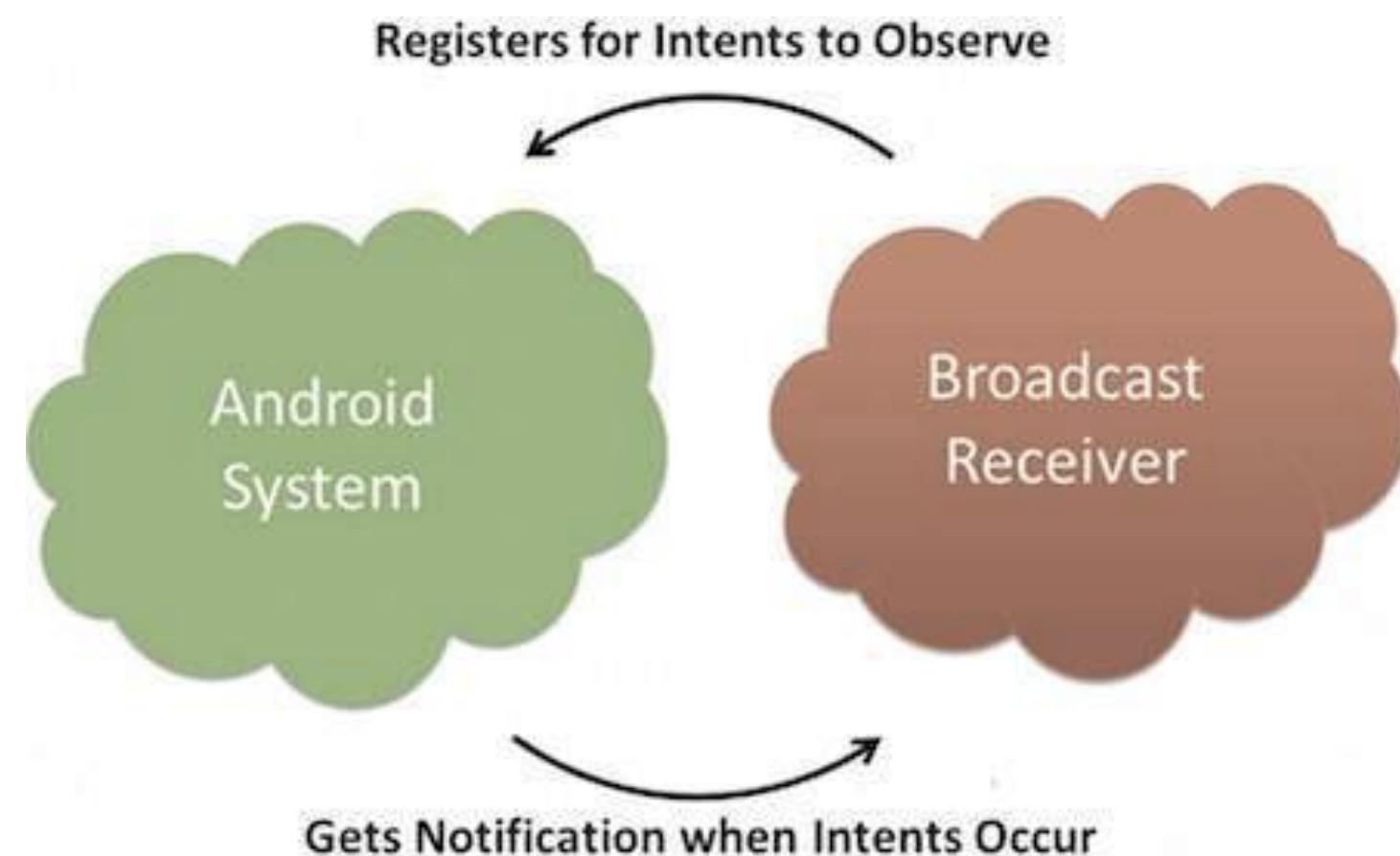


외부 앱이 데이터에 접근할 수 있도록 인터페이스 제공

- 메신저에서 전화번호부에 있는 친구 자동 추가
- 보안에 용이

# Android 4대 컴포넌트

## Broadcast Receiver



시스템 범위의 Broadcast 알림을 받을 수 있는 Component

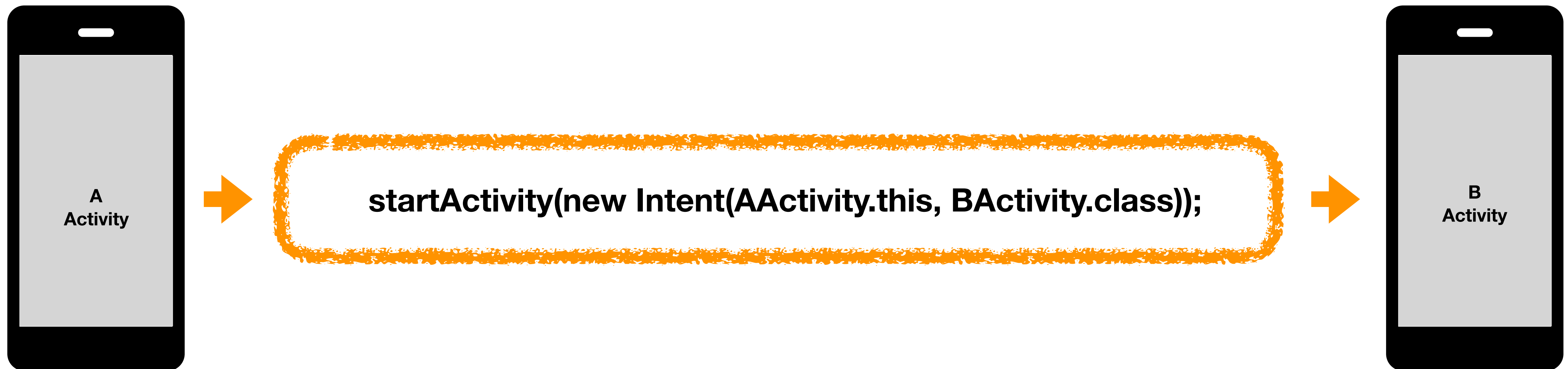
- 핸드폰 켜짐 -> 잠금해제 어플 구현시 사용
- 배터리 % 변화 -> 배터리 위젯 구현시 사용
- 비행기 모드 -> 인터넷 요청 중단 및 캐싱 처리

참고 : [https://chromium.googlesource.com/android\\_tools/+/refs/heads/master/sdk/platforms/android-28/data/broadcast\\_actions.txt](https://chromium.googlesource.com/android_tools/+/refs/heads/master/sdk/platforms/android-28/data/broadcast_actions.txt)

# Android 4대 컴포넌트

## Intent

### Component 들간의 이동 관리

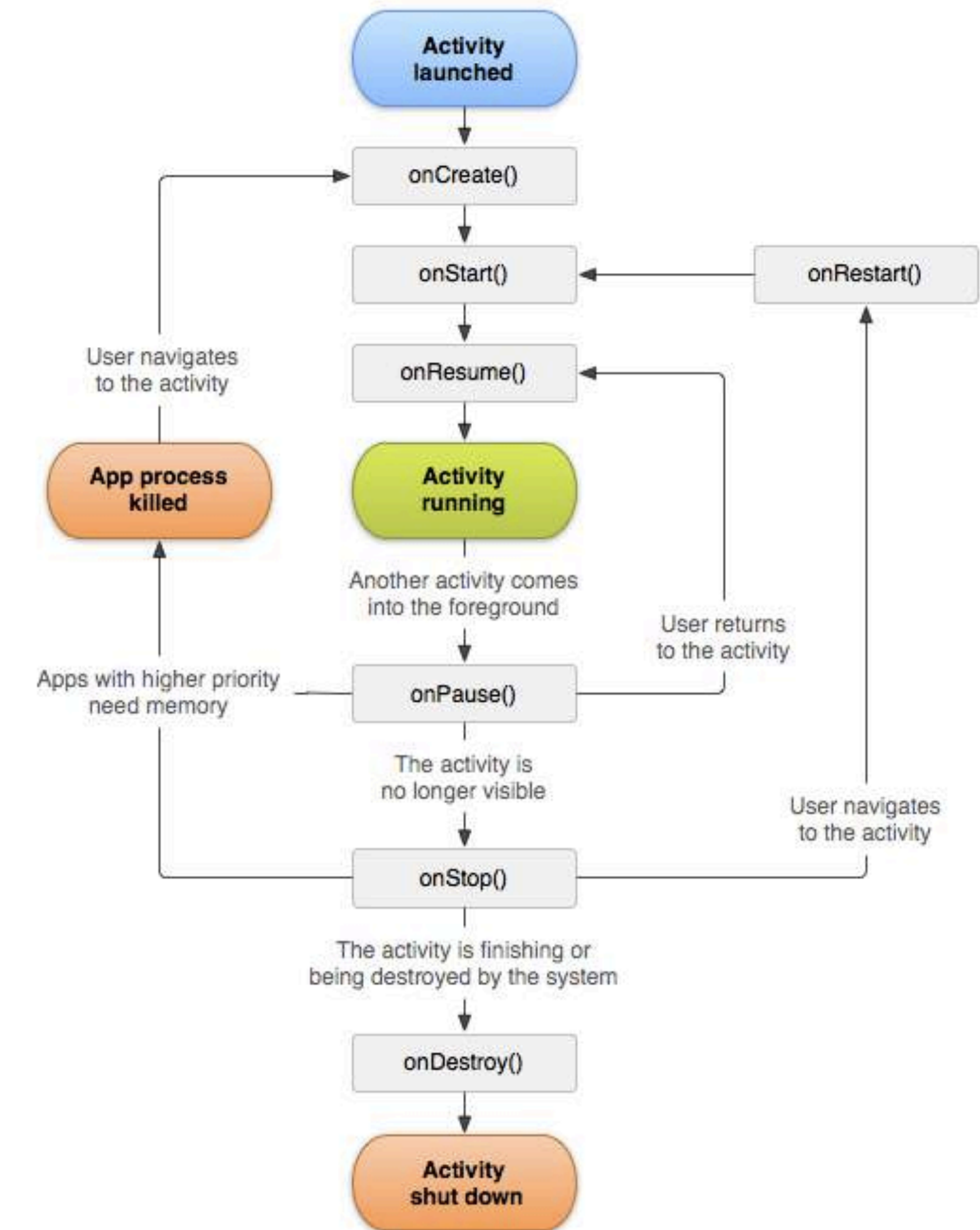


# Android Lifecycle

# Android Lifecycle

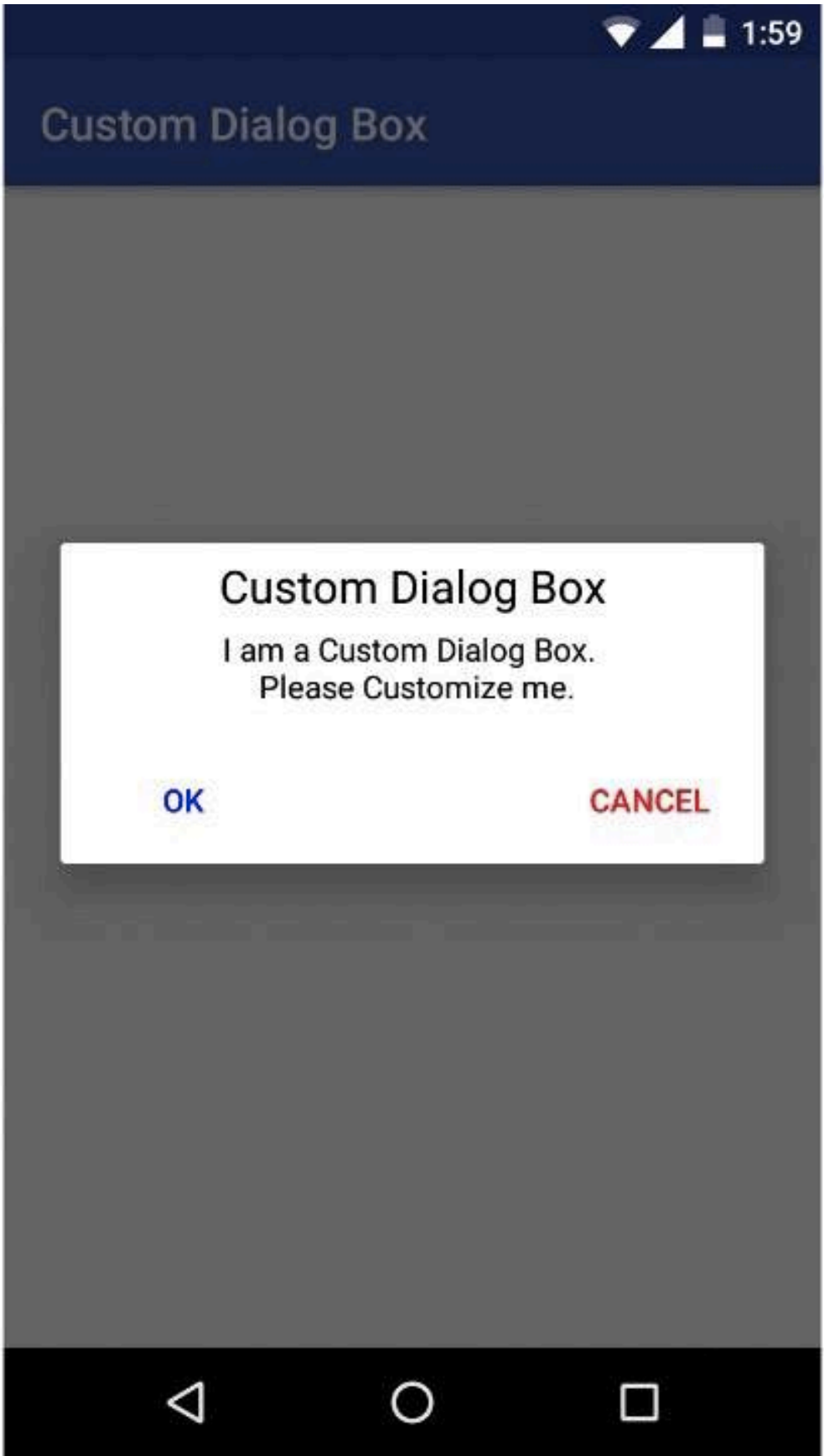
## Activity's lifecycle

- onCreate - activity 첫 생성
- onStart - stop()후 restart()시에만 호출
- onResume - activity 재시작
- onPause - 다른 activity가 foreground로 나올 경우
- onStop - activity가 완전히 스크린에서 보이지 않음
- onDestroy - activity가 완전히 소멸됨



# Android Lifecycle

## onPause





# 예시 소스

AWS

# AWS란?


# AWS란?

내 서비스를 원하는 방식으로 빠르게 배포할 수 있다

- 100가지가 넘는 서비스가 존재
- 필요한 서비스를 사용, 조합하여 서비스를 배포 할 수 있습니다
  - 컴퓨팅 (EC2, Lambda)
  - 스토리지 (S3)
  - 등등...
- 서버를 선금으로 구매 할 필요가 없이, 사용한 만큼 과금됩니다

# AWS란?

어떻게 공부하면 돼요?

- **EC2로 일단 먼저 배포해보기!**
  - 우리에게 익숙한 Ubuntu 인스턴스를 켜서,  
개발환경을 서버에서 켜는 방식으로 배포해봅니다
  - 접속이 안된다면, **Security Group 설정 잊지 말기!**
- **어떻게 하면 더 쉽게 배포하고 관리할까?** 라는 생각이 들면,
  - 즐거운 마음으로 AWS 웹사이트를 살펴봅니다 

# 공부 할 만한 서비스 추천

Erion에서 사용하고 있는 서비스들

- 컴퓨팅
  - **Lambda (웹사이트 SSR로 사용 중)**
  - **Elastic Beanstalk (실시간 통신 서버로 사용 중)**
- 스토리지
  - **S3 (이미지, JS, CSS 등 저장)**
- 데이터베이스
  - **RDS, ElasticSearch Service (관리형 DB로 사용 중)**
  - **DynamoDB (NoSQL)**



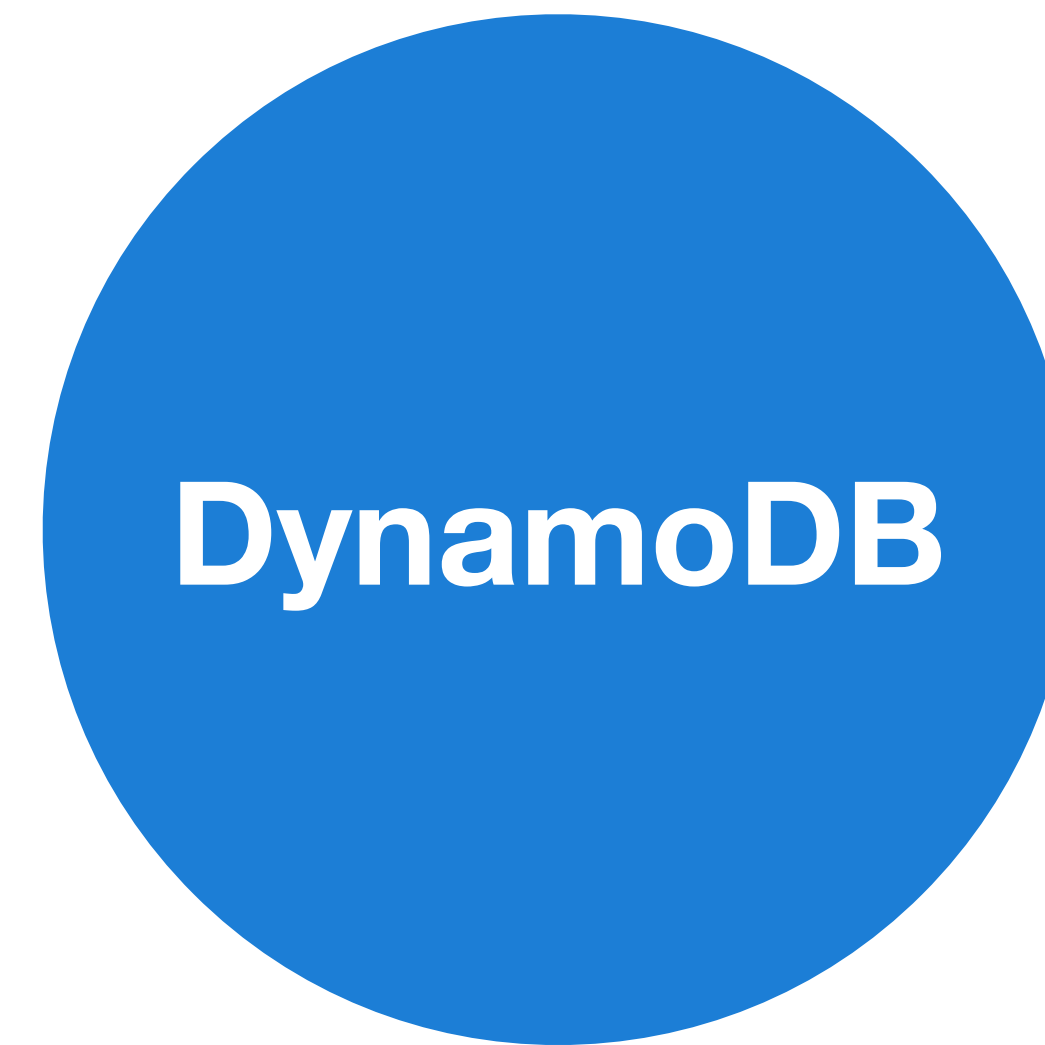
# EC2에서 DynamoDB에 접근하고 싶어요

서비스 섞어 쓰기



# EC2에서 DynamoDB에 접근하고 싶어요

서비스 섞어 쓰기



?? 너가 누군줄 알고???

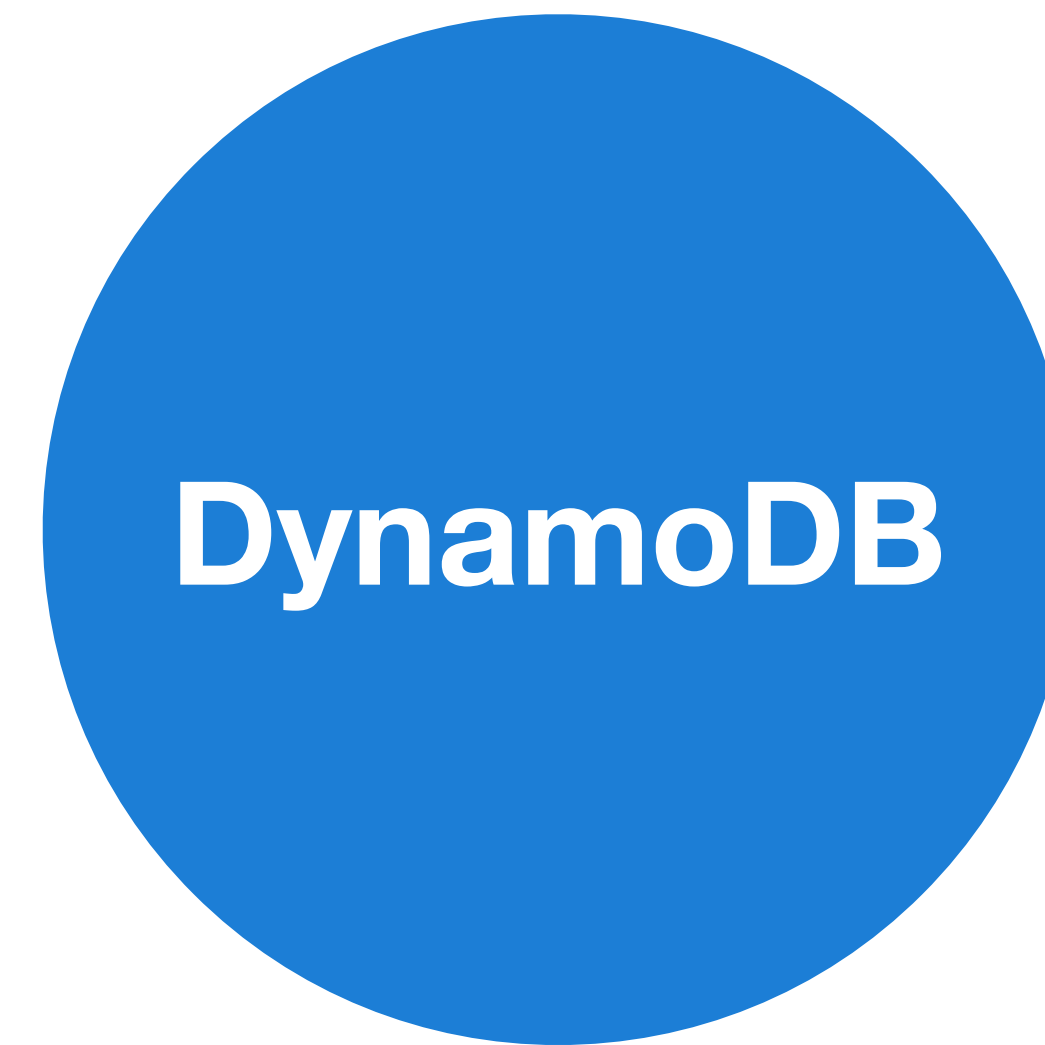
# EC2에서 DynamoDB에 접근하고 싶어요

서비스 섞어 쓰기



# EC2에서 DynamoDB에 접근하고 싶어요

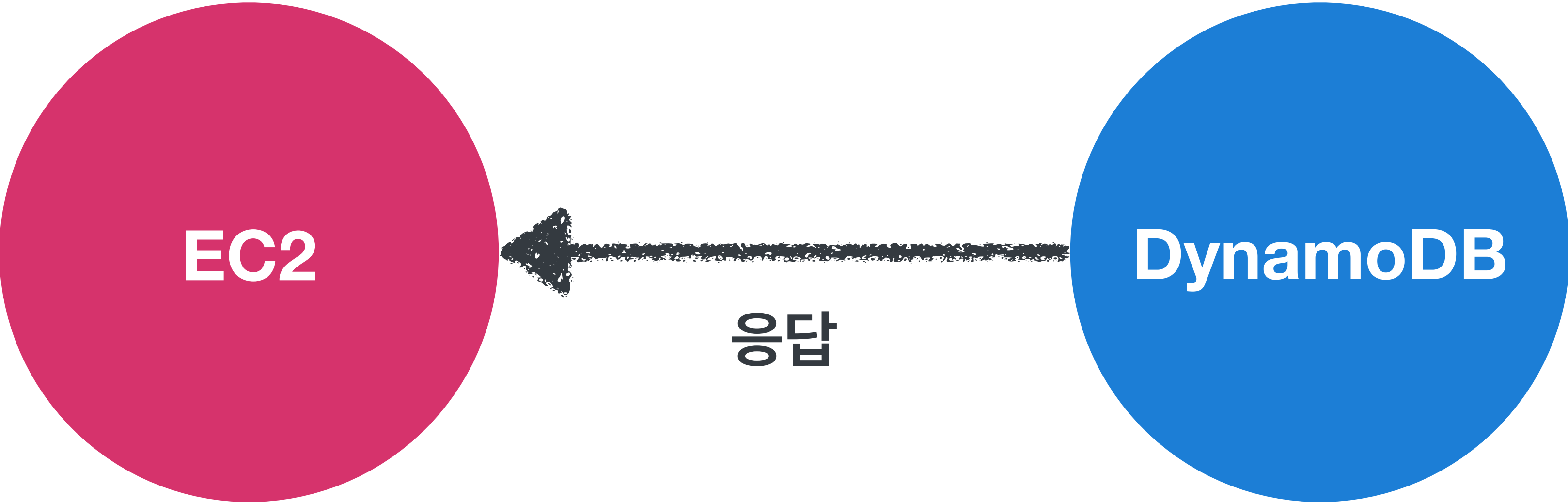
서비스 섞어 쓰기



나한테 접근 가능하구나!

# EC2에서 DynamoDB에 접근하고 싶어요

서비스 섞어 쓰기



# 키는 어디서?

## AWS IAM의 기본



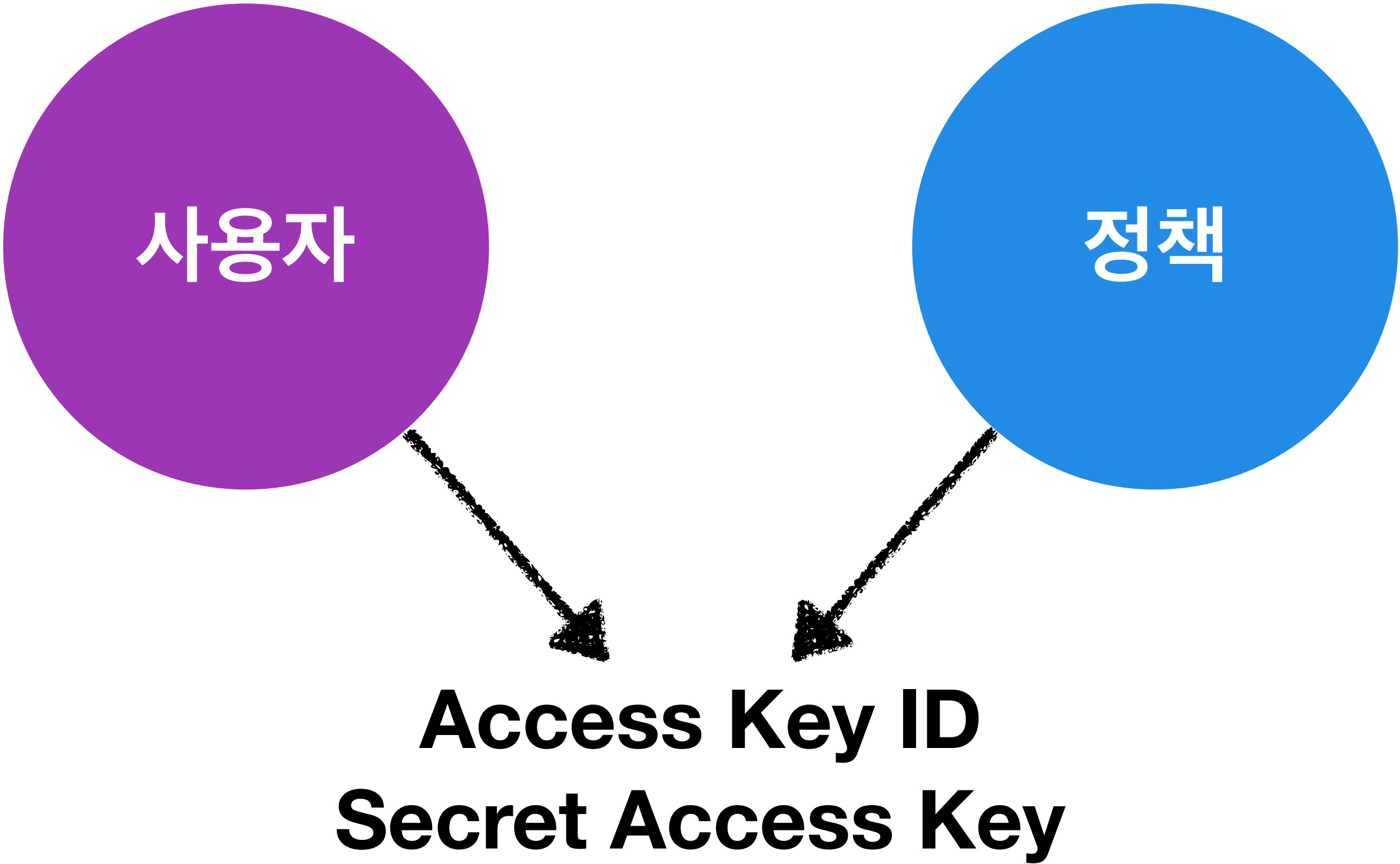
누구에게



어떤 권한을 줄까?



# 키 발급받기



# [실습 3] AWS IAM 사용자 만들기



# 이 설문조사는 영국에서 시작되어... 🕵️

오늘 어떤 모임에 참여하셨나요? → ‘대학생’ 선택

한줄 요약: 완료하시면 부자된대요 😊