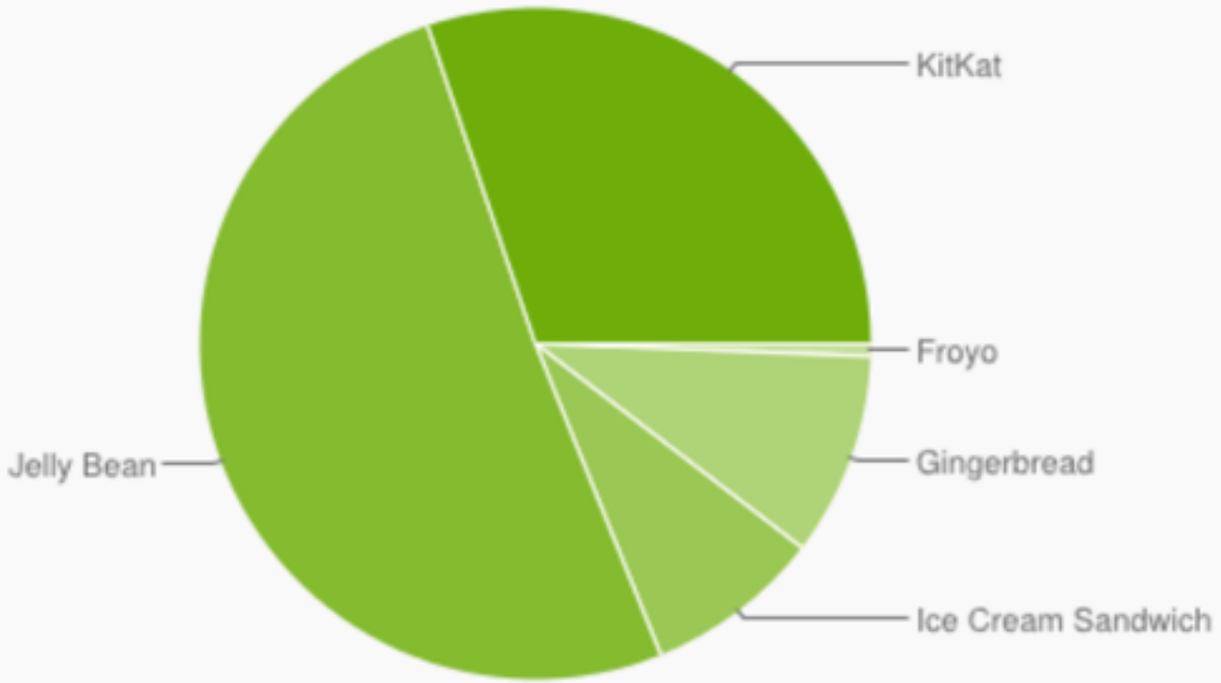
Android Development

introduction to the platform

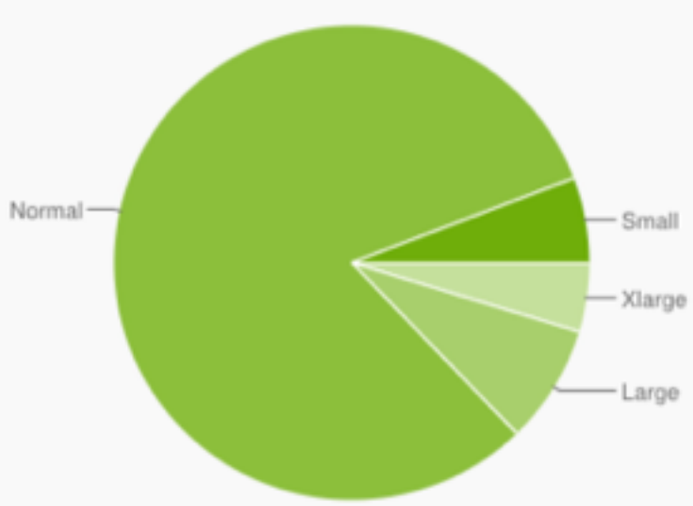
Platform Fragmentation

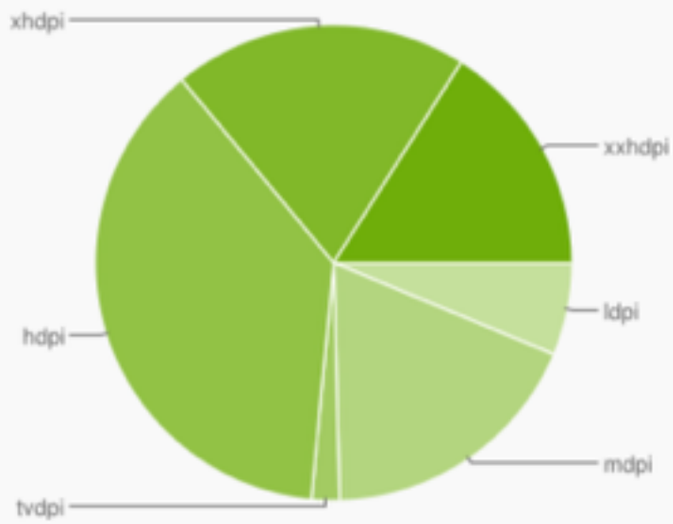
Platform Version	API Level	VERSION_CODE	Notes
Android 5.0	21	LOLLIPOP	Platform Highlights
Android 4.4W	20	KITKAT_WATCH	KitKat for Wearables Only
Android 4.4	19	KITKAT	Platform Highlights
Android 4.3	18	JELLY_BEAN_MR2	Platform Highlights
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	Platform Highlights
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	Platform Highlights
Android 3.0.x	11	HONEYCOMB	Platform Highlights
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1	Platform Highlights
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD	
Android 2.2.x	8	FROYO	Platform Highlights
Android 2.1.x	7	ECLAIR_MR1	Platform Highlights
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONUT	Platform Highlights
Android 1.5	3	CUPCAKE	Platform Highlights
Android 1.1	2	BASE_1_1	
Android 1.0	1	BASE	

Version	Codename	API	Distribution
2.2	Froyo	8	0.6%
2.3.3 - 2.3.7	Gingerbread	10	9.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	8.5%
4.1.x	Jelly Bean	16	22.8%
4.2.x		17	20.8%
4.3		18	7.3%
4.4	KitKat	19	30.2%



	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	5.8%						5.8%
Normal		9.9%		36.6%	18.9%	16.0%	81.4%
Large	0.5%	4.5%	1.9%	0.6%	0.6%		8.1%
Xlarge		3.9%		0.3%	0.5%		4.7%
Total	6.3%	18.3%	1.9%	37.5%	20.0%	16.0%	





Application Fundamentals

- Applications are written in the Java programming language.
- Compiled into an Android package file (.apk).
- Each application runs in its own sandbox and Linux process.
- Applications consist of components, a manifest file and resources.
- Components:
  - Activities
  - Services
  - Content providers
  - Broadcast receivers

Activities

- An activity represents a single screen with a user interface.
- Most applications contain multiple activities.
- When a new activity starts, it is pushed onto the *back stack*.
- User interface can be built with XML or in Java.
- Monitor lifespan through callback methods like onStart(), onPause(), etc.

Services

- Services perform long-running operations in the background.
- Does not contain a user interface.
- Useful for things like network operations, playing music, etc.
- Runs independently of the component that created it.
- Can be bound to by other application components, if allowed.

Content Providers

- Used to store and retrieve data and make it accessible to all applications.
- Are the only way to share data across applications.
- Exposes a public URI that uniquely identifies its data set.
- Data is exposed as a simple table on a database model.
- Android contains many providers for things like contacts, media, etc.

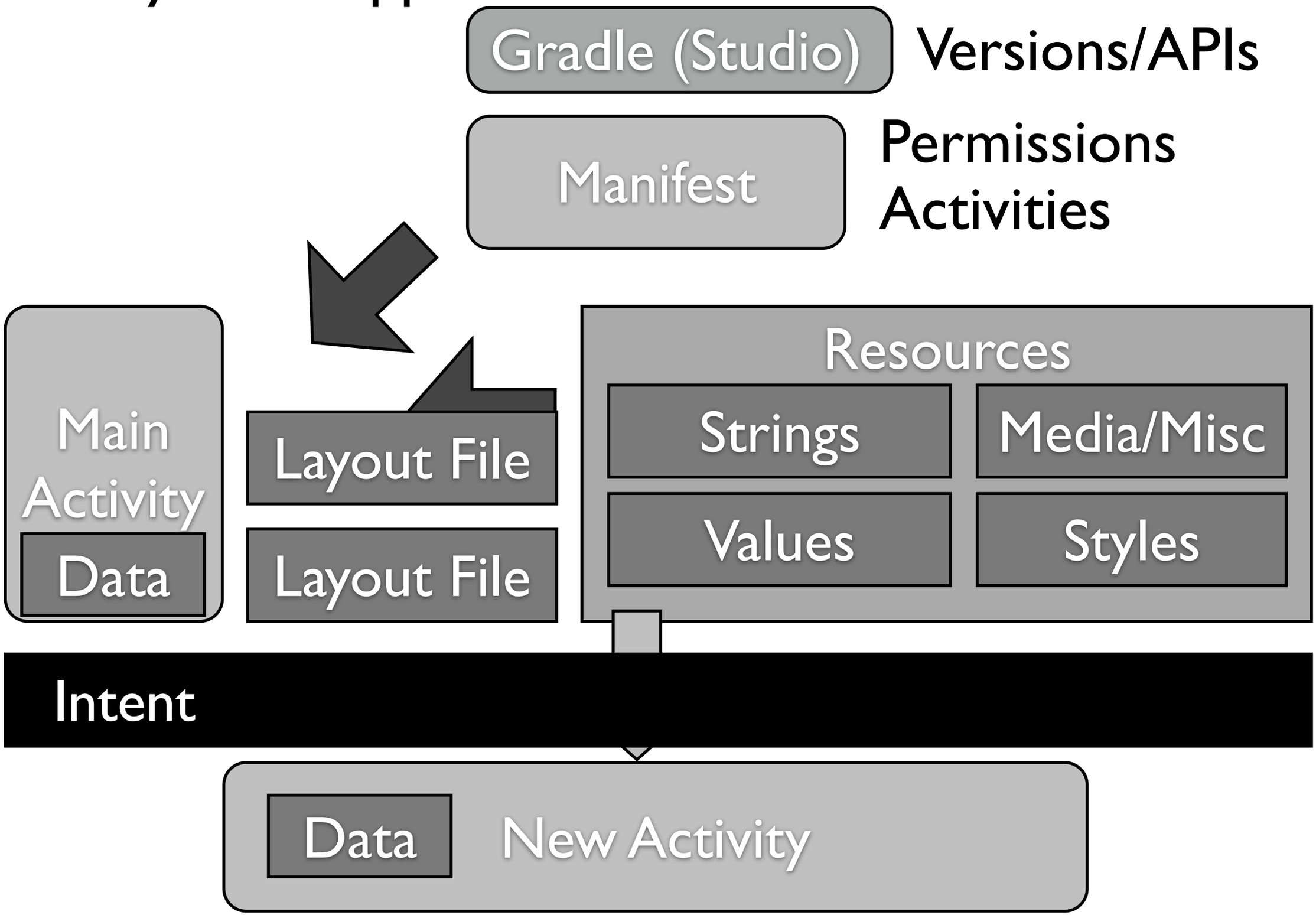
Broadcast Receivers

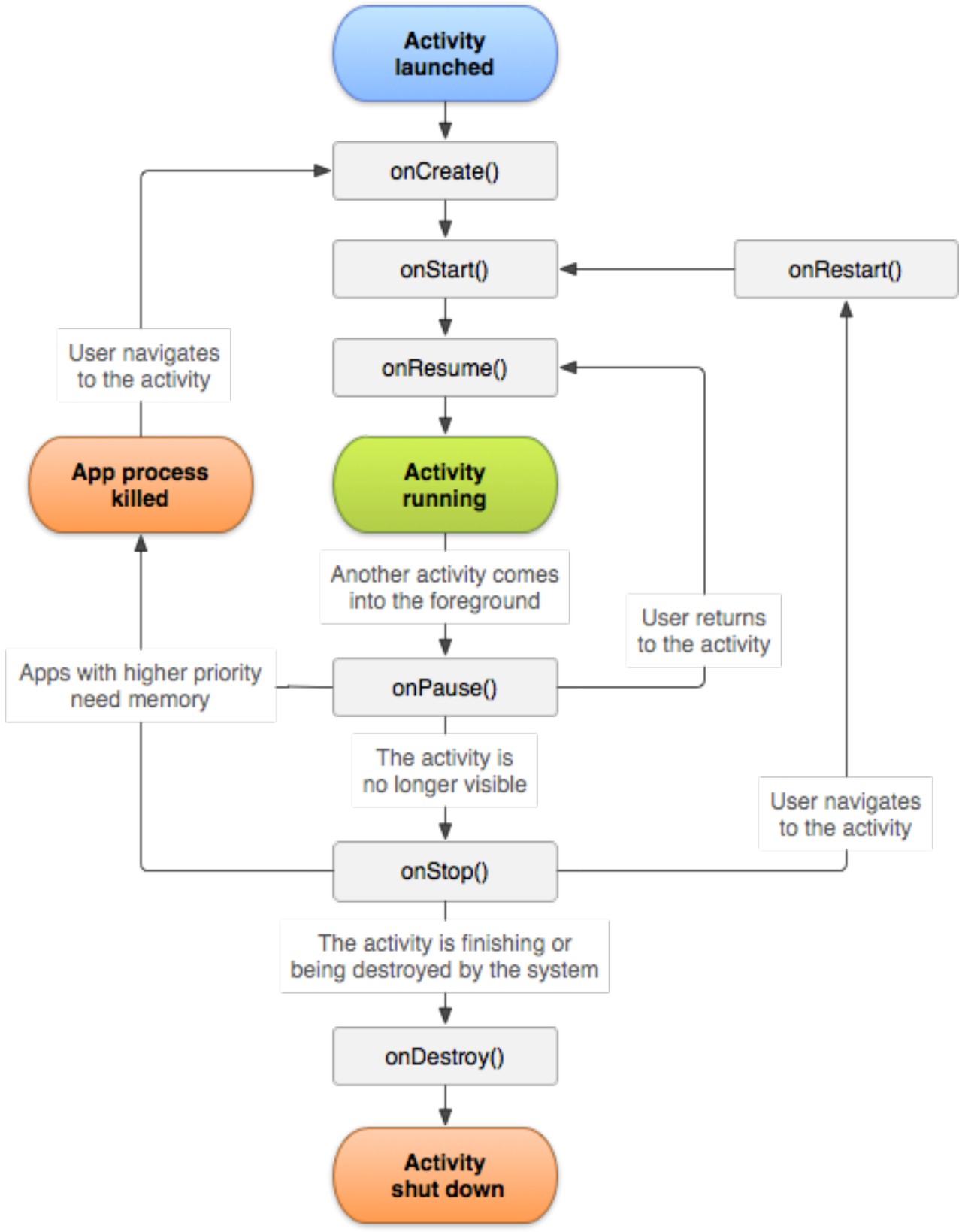
- A component that responds to system-wide broadcast announcements.
- Examples include when the screen turns off, the battery is low, etc.
- Applications can also initiate their own broadcasts.
- Broadcast receivers contain no user interface.
- They can create status bar notifications to alert the user.

Android Manifest File - XML

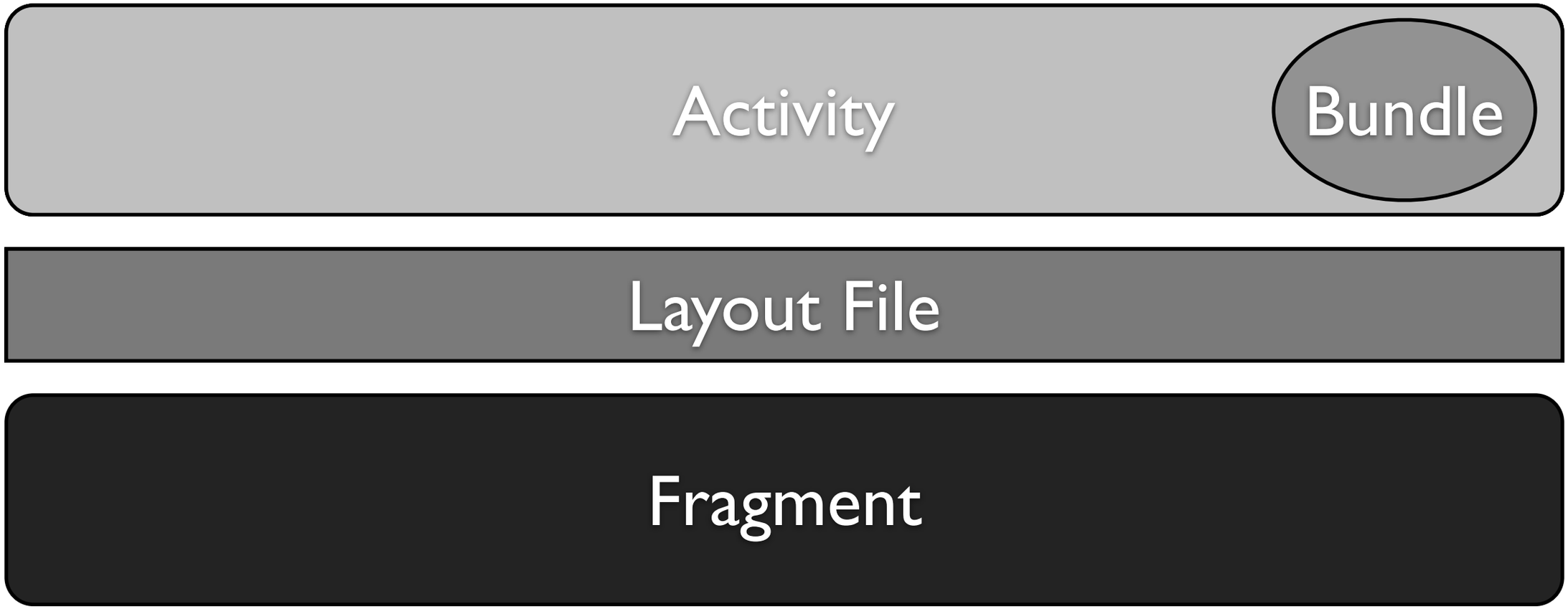
- Applications must have an AndroidManifest.xml file in its root directory.
- Presents information about the application to the Android system.
- Describes the components used in the application.
- Declares the permissions required to run the application.
- Declares the minimum Android API level that the application requires.

Anatomy of an App



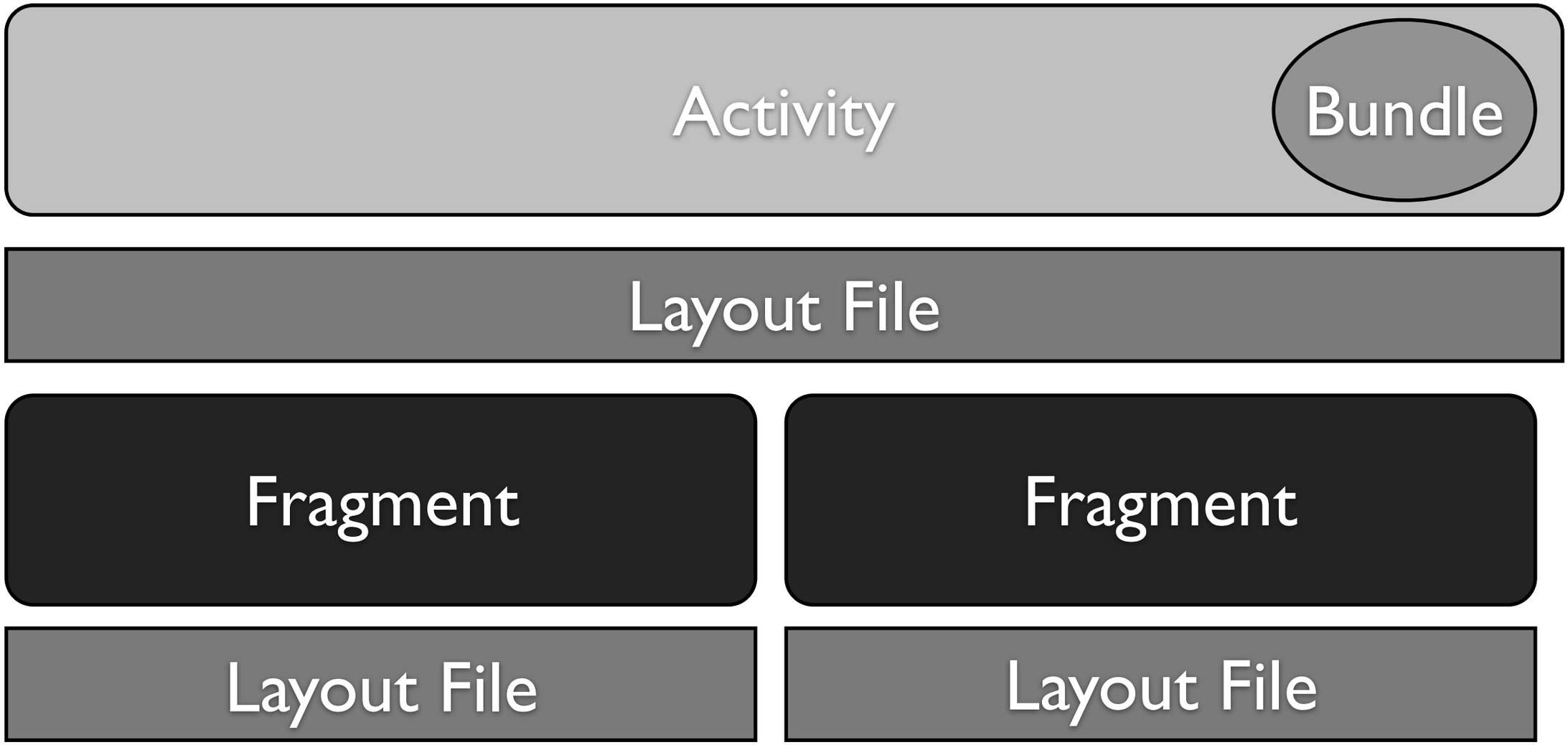

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

fragments

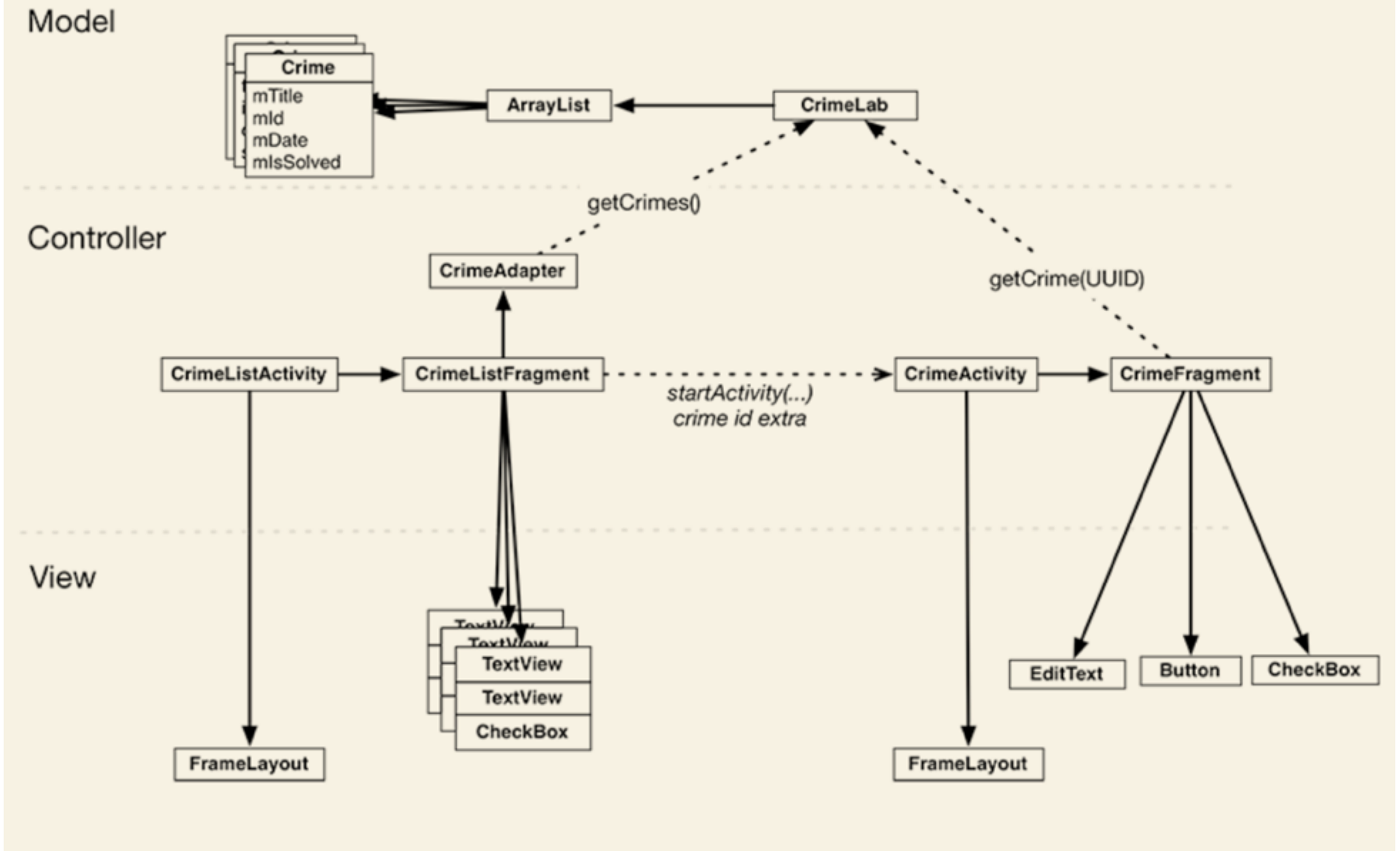


Layout File

Rotation Event


Activity
Bundle
Layout File
Fragment
Fragment
Layout File
Layout File

Model View Control


Model
Crime
mTitle
mId
mDate
mIsSolved
ArrayList
CrimeLab
getCrimes()
Controller
CrimeAdapter
getCrime(UUID)
CrimeListActivity
CrimeListFragment
startActivity(...)
crime id extra
CrimeActivity
CrimeFragment
View
TextView
TextView
CheckBox
EditText
Button
CheckBox
FrameLayout
FrameLayout

Configuration Qualifiers

Screen characteristic	Qualifier
Size	small
	normal
	large
	xlarge
Density	ldpi
	mdpi
	hdpi
	xhdpi
	nodpi
Orientation	land
	port
Aspect ratio	long
	notlong

Configuration Qualifiers

```
res/layout/my_layout.xml             // layout for normal screen size ("default")
res/layout-small/my_layout.xml       // layout for small screen size
res/layout-large/my_layout.xml       // layout for large screen size
res/layout-xlarge/my_layout.xml      // layout for extra large screen size
res/layout-xlarge-land/my_layout.xml // layout for extra large in landscape orientation
```

```
res/drawable-mdpi/my_icon.png        // bitmap for medium density
res/drawable-hdpi/my_icon.png        // bitmap for high density
res/drawable-xhdpi/my_icon.png       // bitmap for extra high density
```