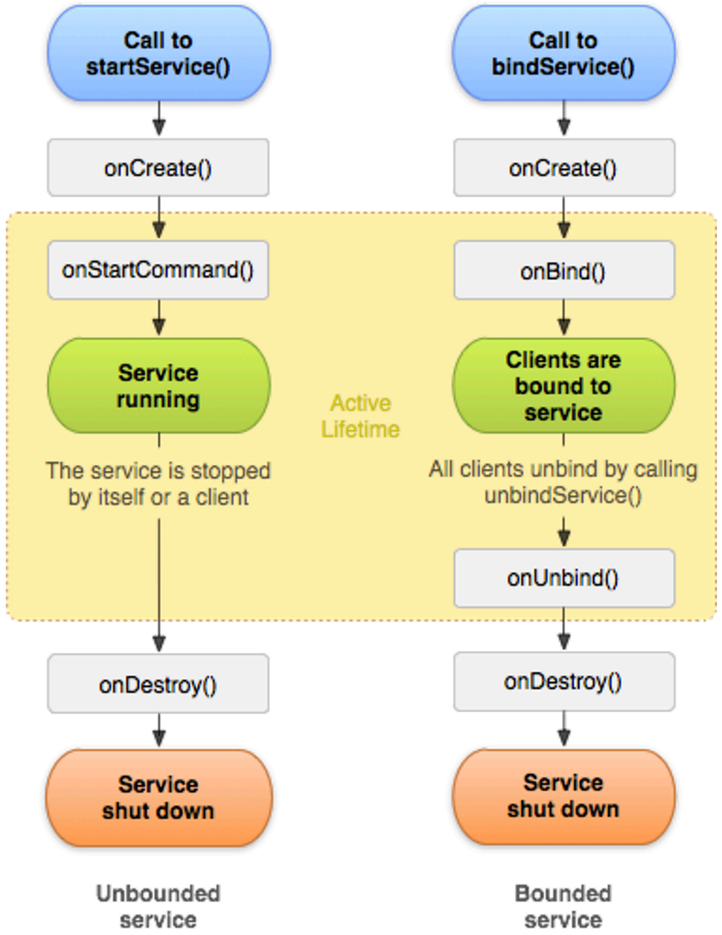

MOBILE APPLICATION DEVELOPMENT

ANDROID (2017)

LECTURE 16: SERVICES

ANDROID COMPONENTS

- The primary building blocks of Android applications.
- Currently exist in four variants:
  - Activities
  - Broadcast Receivers
  - Content Providers
  - Services
- Components serve as entry points from Android into your application.

ACTIVITIES

- The most familiar component in terms of this course.
- Represent single screens with a user interface.
  - Can launch other components, particularly other Activities.
  - Often serve as the primary (and only) entry point into an application.
- Tasked with managing the state of the user's view of an application, and allowing users to interact with the application.

BROADCAST RECEIVERS

- Serve as entry points into applications which are accessed in response to broadcasted events sent by external components.
- Allow applications to respond to events without requiring the applications to monitor for those events in the background.
  - When an applicable broadcast is received, the application 'wakes up'.
  - Due to resource considerations, Android has restricted these components as of Android 8.0 in many cases, but they are still useful for a variety of tasks.
- Example: displaying a notification in response to a certain type of event occurring.

CONTENT PROVIDERS

- Manage access to a set of data controlled by an application.
- Allow other components to obtain and even modify data in a structured and predictable way.
  - Data may be read-only, partially writable, or fully writable.
  - Provides a database-like abstraction over any number of data formats, which may be utilized by components who do not have direct access to the data.
- Examples: providing information to widgets, allowing applications to edit contacts, creating a shared media library with support for metadata editing.

SERVICES

- Components which allow background execution of an application's tasks, even when the application does not appear on the screen. Should create their own internal threads for long-running tasks, because by default they run on the main thread of their application.
- Defined as one of three types of Service:
  - Foreground Service: Displays a status bar icon and does something the user should be able to notice.
  - Background Service: Performs an action that the user does not generally notice. Limited as of Android 8.0.
  - Bound Service: A background Service which is managed by other components.

FOREGROUND SERVICES

- Designed to allow applications to provide some sort of ongoing interaction with the user without the application being open.
- Should perform a useful, but low-impact function and must be apparent to the user (because the point of such a Service is to have the user notice it).
  - Must display a notification and status bar icon which cannot be dismissed while the Service is running.
  - Should be used wisely, as Services place a burden on the system as a whole.
- Examples: media players, fitness trackers, download managers.

BACKGROUND SERVICES

- Services which do not interact with or present themselves to the user, but which run in the background to perform a task that requires no external interaction.
- Should perform a necessary function behind the scenes which the user can not interact with.
  - May be launchable by any component, or private to a specific application.
  - Must terminate once they have completed their intended goals.
- Examples: cache cleanup for an application, ongoing simulation, long-running downloads. Restricted in Android 8.0 largely in favor of using JobScheduler.

BOUND SERVICES

- Similar to background Services, except that such Services must be bound to other components in order to run.
- Should provide decentralized background functionality to other components.
  - When started, provide IBinder instances to allow an interface to other components by which the started Service can be manipulated.
  - Will stop execution if no components are bound to the Service anymore.
- Example: download manager with an open interface, shared data generator, chess engine with multiple clients, etc. As of Android 8.0, JobScheduler is a recommended alternative to bound Services.

JOBSCHEDULER

- A framework which provides a system-level representation of job scheduling.
- Programmers may define 'jobs' for the JobScheduler to run, and the jobs will be run in such a way where the scheduling is done with respect to the whole system.
  - May specify strategies for 'backing off' from failing jobs.
  - Jobs are batched, executed, and finalized strategically by the system.
- Increasingly becoming the recommended way to run tasks in the background, because it allows the system to avoid resource problems as many jobs come in.
- Does not serve (generally) as a replacement for foreground Services, but attempts to replace many other kinds of Services.