Introducing Services

Unlike Activities, which present a rich graphical interface to users, Services run in the background — updating your Content Providers, firing Intents, and triggering Notifications. They are the perfect way to perform regular processing or handle events even after your application's Activities are invisible, inactive, or have been closed.

With no visual interface, Services are started, stopped, and controlled from other application components including other Services, Activities, and Broadcast Receivers. If your application regularly, or continuously, performs actions that don't depend directly on user input, Services may be the answer.

Started Services receive higher priority than inactive or invisible Activities, making them less likely to be terminated by the run time's resource management. The only time Android will stop a Service prematurely is when it's the only way for a foreground Activity to gain required resources; if that happens, your Service will be restarted automatically when resources become available.

Applications that update regularly but only rarely or intermittently need user interaction are good candidates for implementation as Services. MP3 players and sports-score monitors are examples of applications that should continue to run and update without an interactive visual component (Activity) visible.

Further examples can be found within the software stack itself; Android implements several Services including the Location Manager, Media Controller, and the Notification Manager.