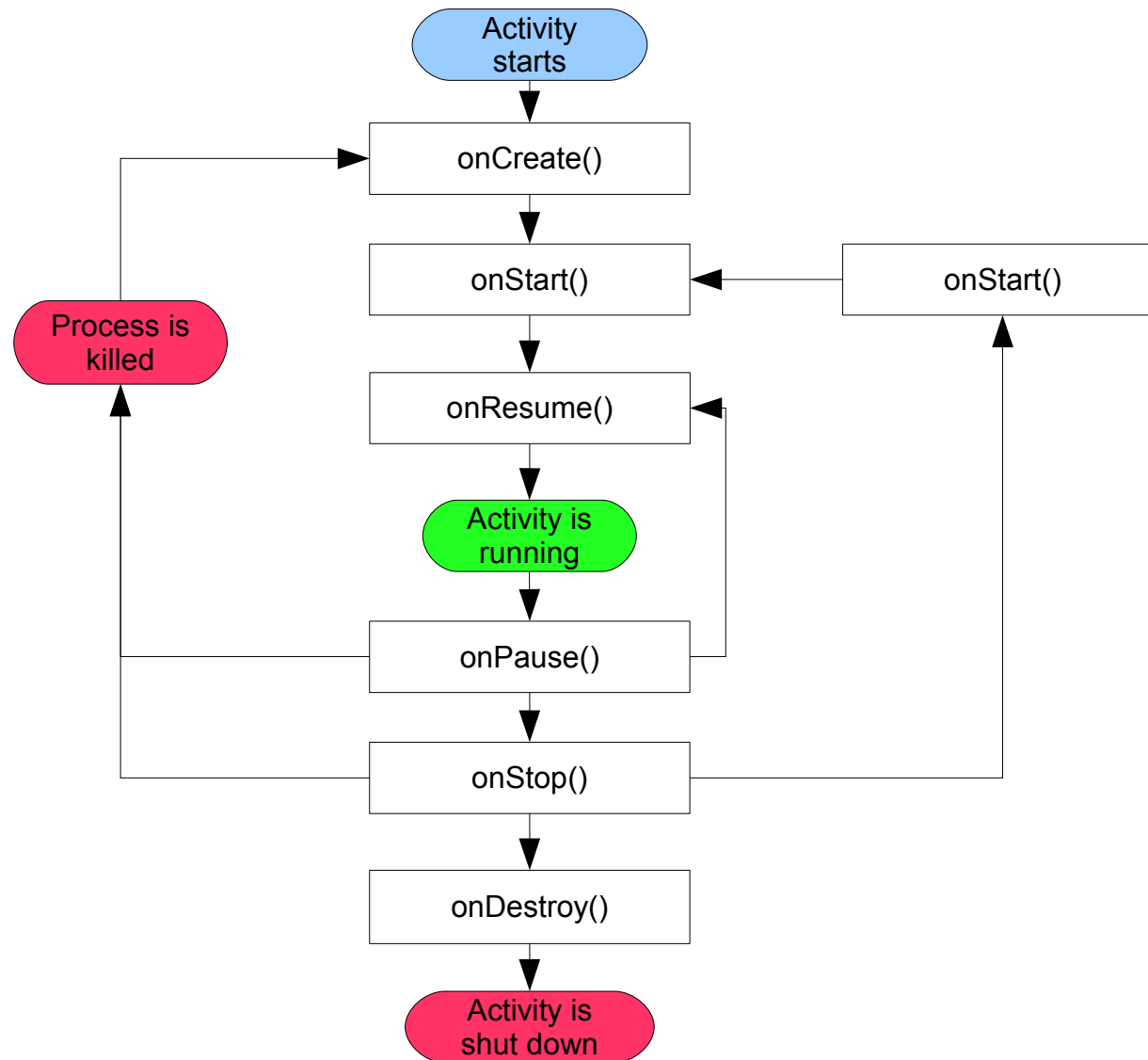


# **Activity / Service & Notification**

# Activity

- アプリケーションの実体
  - **onCreate()** がアプリケーションのエントリポイント
- **Interactive** で視覚的なインターフェース
- **Activity** にはウィンドウが割り当てられる
  - ウィンドウのコンテンツは **View** の階層として提供される
- **1** つのアプリケーションに複数の **Activity** が存在してもよい
  - 各 **activity** はそれぞれ独立している

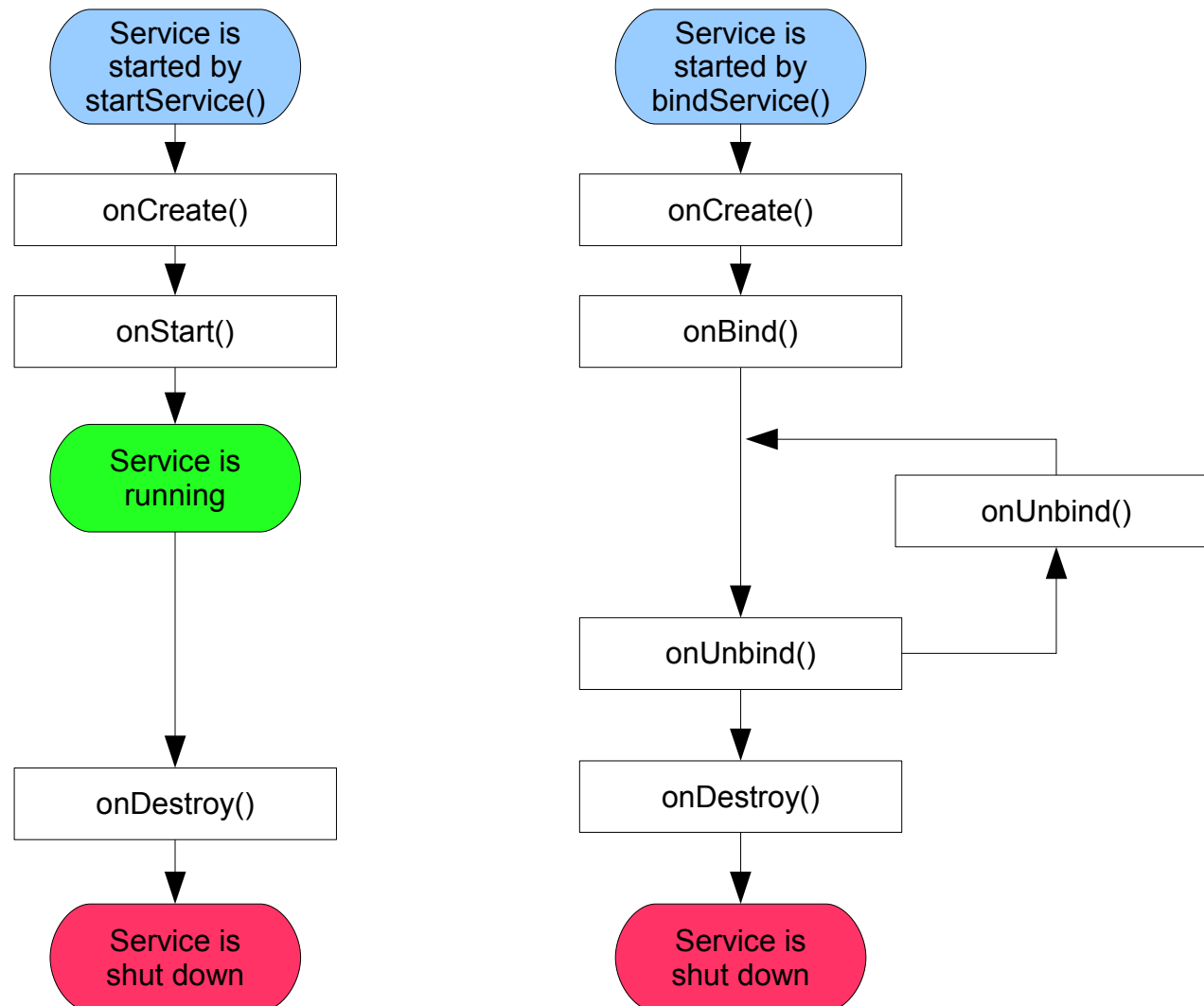
# ライフサイクル(Activity)



# Service

- バックグラウンドで実行されるプロセス
- 画面を占有しない
- **startService()** でサービスを開始する
- **bindService()** で実行中のサービスに接続することもできる
- 例として、音楽再生の **Service** があるとしたら、再生を実行中の **Service** にアプリケーションから接続すると、一時停止や巻き戻しなどの操作ができる

# ライフサイクル(Service)



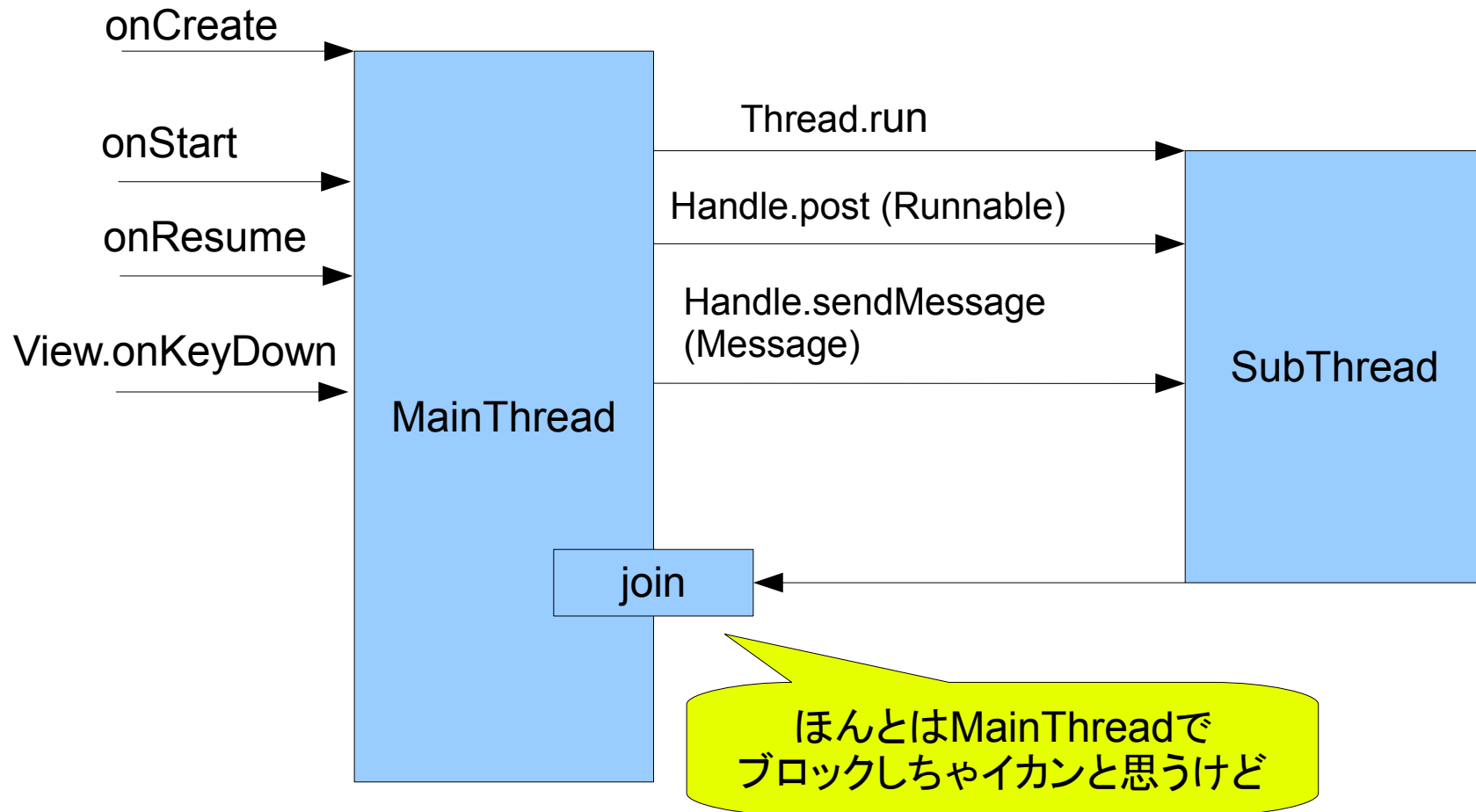
# プロセスとスレッド

- デフォルトでは、アプリケーション **1** つに対して、**Linux** プロセスを **1** つの実行スレッドで開始する
- ただし、他のプロセスで実行されるようにしたり、スレッドを追加したりすることが可能
- **onCreate()** とか **View.onKeyDown()** とかは、必ずプロセスのメインスレッドで実行される

# スレッド

- 時間のかかる処理などは別のスレッドを割り当てて処理をするべき
- メインスレッドがキー操作などを受け付けられないと操作性が悪くなる
- スレッドは **Java** 標準の **Thread** で作成
- **Android** 特有のスレッド管理のためのクラス
  - **Looper** スレッド内でメッセージループを実行
  - **Handler** メッセージを処理する
  - **HandlerThread** メッセージループでスレッドを設定

# スレッド





# Notification

- ユーザへの通知
  - 通知内容は下記の **4** つ
    - アイコン
    - タイトル
    - 内容
    - 発生時間
- **Service** からユーザへの通知などで使う
- ステータスバーに通知アイコンを表示する
  - ステータスバーを”引っ張る”と通知ウィンドウが開く
  - 通知ウィンドウで選択したときの動作を登録可能

# Notification

- **Service** は、直接 **Activity** を起動すべきではない
- その代わりに、**Notification** で通知すべき
- **Service** からしか利用できない、というわけではない
- **Notification** から **Intent** を発行することにより **Activity** の起動などができる
- 通知登録は **Android System Service** の **NotificationManager** に登録する