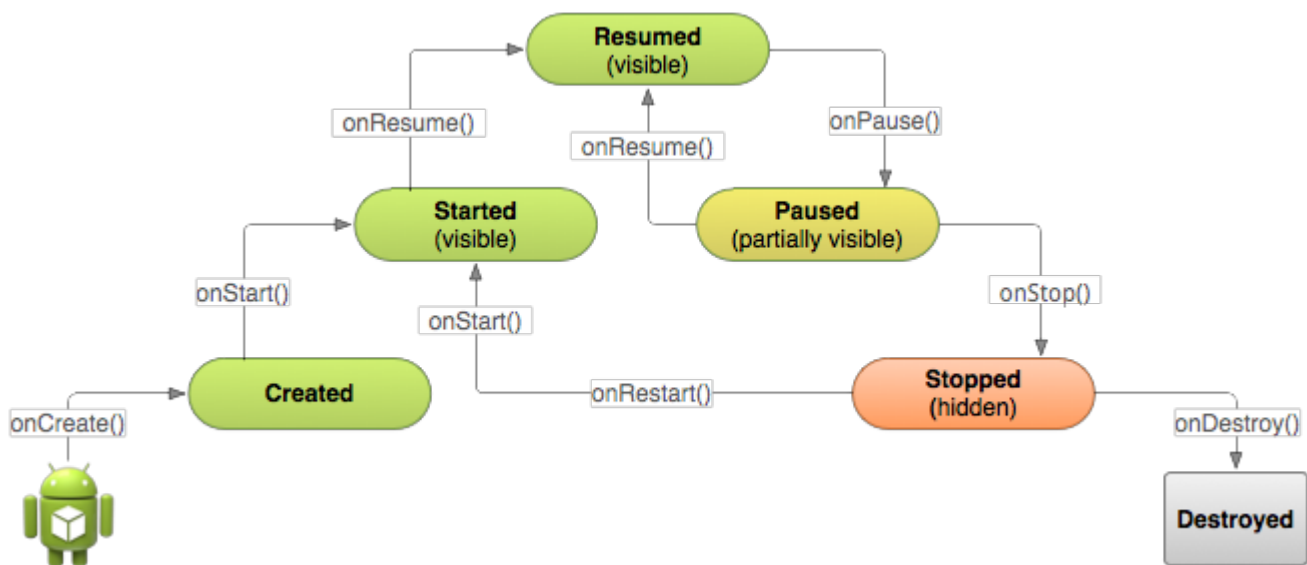


# Retrofit

etrofit

## Ciclo Activity



Quando un'activity va in esecuzione per interagire direttamente con l'utente vengono obbligatoriamente invocati tre metodi:

- **onCreate:** l'activity viene creata. Il programmatore deve assegnare le configurazioni di base e definire quale sarà il layout dell'interfaccia;
- **onStart:** l'activity diventa visibile. È il momento in cui si possono attivare funzionalità e servizi che devono offrire informazioni all'utente;
- **onResume:** l'activity diventa la destinataria di tutti gli input dell'utente. Android pone a riposo l'activity nel momento in cui l'utente sposta la sua attenzione su un'altra attività del sistema, ad esempio apre un'applicazione diversa, riceve una telefonata o semplicemente – anche nell'ambito della stessa applicazione – viene attivata un'altra Activity. Anche questo percorso, passa per tre metodi di callback:
- **onPause:** (l'inverso di onResume) notifica la cessata interazione dell'utente con l'activity;
- **onStop:** (contraltare di onStart) segna la fine della visibilità dell'activity;
- **onDestroy:** (contrapposto a onCreate) segna la distruzione dell'activity.

Il passaggio da un'Activity ad un'altra coinvolge i cicli di vita di entrambe. La prima, quella messa a riposo, dovrà passare almeno per onPause (cessazione interazione con l'utente) e onStop (activity non più visibile) mentre la seconda percorrerà la catena di creazione onCreate–onStart–onResume.

Ma in che ordine avverrà tutto ciò? La priorità del sistema è il mantenimento della fluidità della user-experience. Per questo la consecutio delle operazioni sarà:

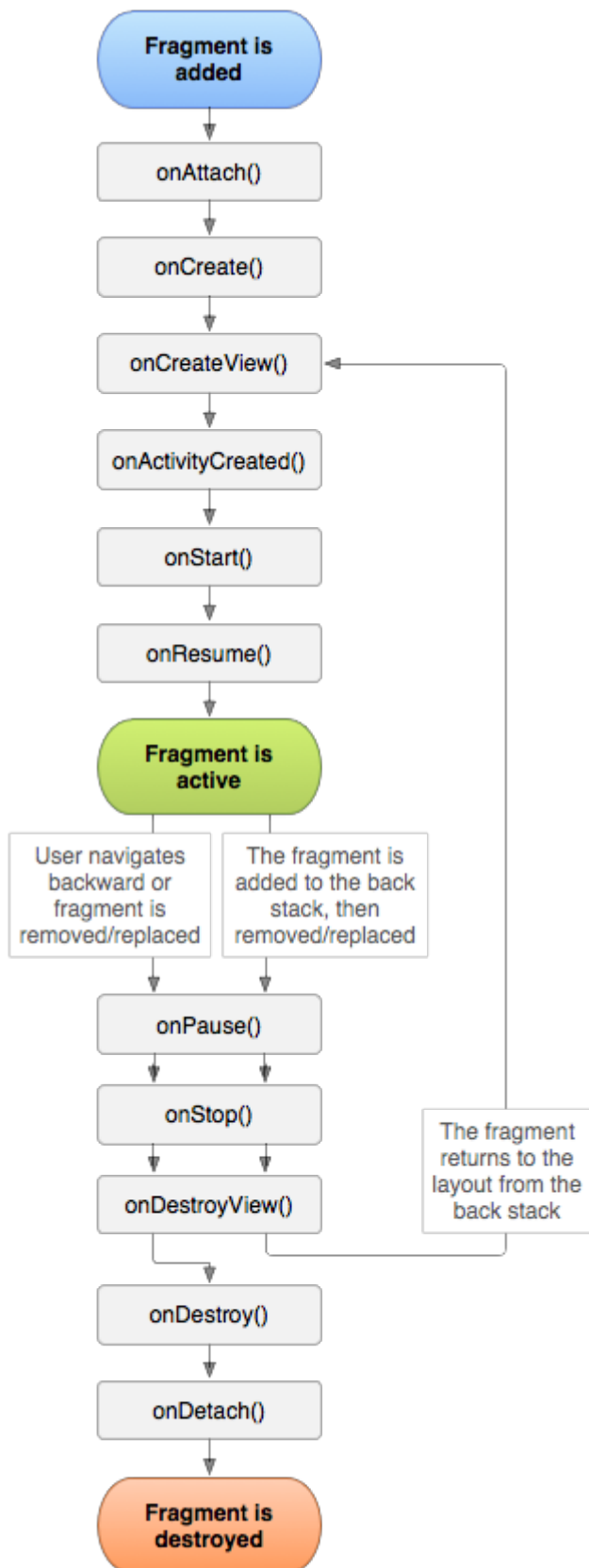
- la prima Activity passa per onPause e viene fermata in stato Paused;

- la seconda Activity va in Running venendo attivata completamente. In tale maniera l'utente potrà usarla al più presto non subendo tempi di ritardo;
- a questo punto, mentre l'utente sta già usando la seconda Activity, il sistema può invocare onStop sulla prima.

## Intent

Un intent è un oggetto (generato dall'omonima classe Intent, nel package android.content) che descrive un'azione da far effettuare a un componente su un insieme di dati. Si tratta, in altre parole, di un meccanismo che consente, attraverso lo scambio di oggetti Intent, di avviare altri componenti della stessa applicazione oppure, cosa assai più interessante, di avviare componenti di altre applicazioni! (poi c'è intent-filter nel manifest...)

## Comunicazione tra fragment



Un Fragment è una porzione di Activity. Ma si faccia attenzione a comprenderne bene il ruolo. Non si tratta solo di un gruppo di controlli o di una sezione del layout. Può essere definito più come una specie di sub-activity con un suo ruolo funzionale molto importante ed un suo ciclo di vita.

Definiamo subito il rapporto tra Fragments e Activity.

Prima di tutto un Fragment non può vivere senza un'Activity. Tipicamente nei nostri programmi

creeremo più Fragments che si alterneranno nel layout mentre di Activity ne sarà sufficiente una (ma possono essere anche di più).

Come detto il Fragment ha il suo ciclo di vita fortemente collegato con quello dell'Activity di appartenenza. Come si vede ricordano molto quelli dell'Activity.

La fase più variegata è l'inizializzazione del fragment:

- **onAttach**: segnala il momento in cui il Fragment scopre l'Activity di appartenenza. Attenzione che a quel punto l'Activity non è stata ancora creata quindi si può solo conservare un riferimento ad essa ma non interagirvi;
- **onCreate**: è la creazione del Fragment in quanto componente;
- **onCreateView**: il programmatore vi lavorerà spesso. È il momento in cui viene creato il layout del Fragment. Solitamente qui si fa uso del LayoutInflater;
- **onActivityCreated**: segnala che la creazione dell'Activity è stata completata, vi si può interagire in tutto e per tutto. Gli altri metodi di callback del ciclo di vita vengono chiamati in corrispondenza degli omonimi metodi dell'Activity.