

Département d'Informatique, L3, S6
Cours développement mobile

Communication entre les composants
d'une application



Université Badji Mokhtar Annaba

sabri.ghazi@univ-annaba.dz

Email: sabri.ghazi@univ-annaba.dz

Plan

- Le cycle d'une **activity**
 - Détails de la classe Activity.
- La navigation d'une activité à une autre.
 - Le Intent.
 - L'appel d'une activité à partir d'une autre
 - L'appel d'une activité et la récupération d'un résultat.

sabri.ghazi@univ-annaba.dz

2

Communication entre les composant d'une application Android

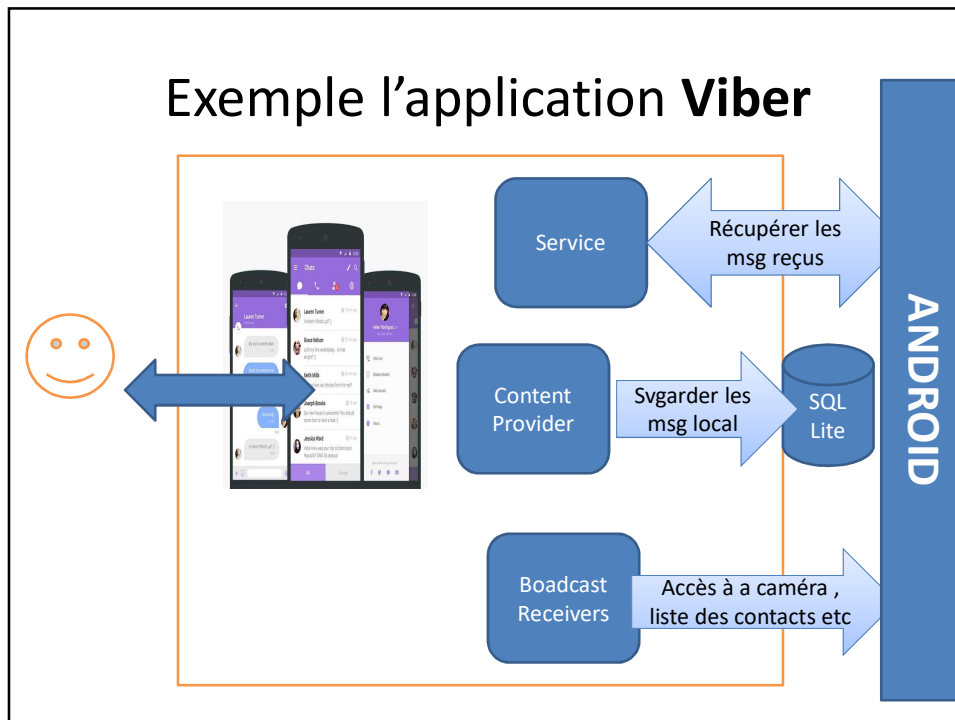
- L'application comme motionné précédemment (voir le cours passé) peut posséder plusieurs **activity**.
- Afin d'assurer la communication entre ces activity, android offre un mécanisme très efficace.
- On peut lancer une **activity** à partir d'une autre (même les **activity** native, qui sont prédéfinies dans le système d'exploitation).

sabri.ghazi@univ-annaba.dz

3

Les composants d'une application Android

- Les composants sont des blocs qui permettent de concevoir des applications.
- Les quatre principaux composants sont :
 - **Activities** : permet de gérer l'interaction de l'utilisateur avec l'application.
 - **Services**: s'occupe des traitements en arrière plan.
 - **Content Providers** : traite l'accès base de données et aussi l'accès aux systèmes de fichiers.
 - **Broadcast Receivers** : gère la communication entre l'application et le système d'exploitation (Android).



D'autres composants

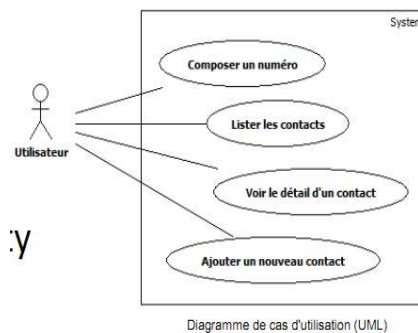
Components	Description
Fragments	Represents a portion of user interface in an Activity.
Views	UI elements that are drawn on-screen including buttons, lists forms etc.
Layouts	View hierarchies that control screen format and appearance of the views.
Intents	Messages wiring components together.
Resources	External elements, such as strings, constants and drawable pictures.
Manifest	Configuration file for the application.

Les Activités

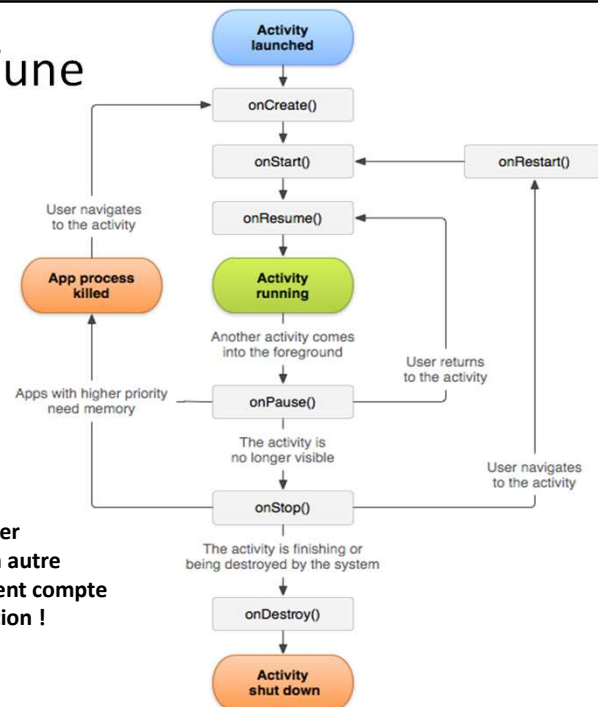
- Une activité est un composant de l'application android qui permet à l'utilisateur de réaliser une tâche:
 - Prendre une photo
 - Faire un appel
 - Envoyer un SMS.
- L'application doit posséder une activité « *main* » principale qui s'exécutera lors de lancement de l'application.
- On peut considérer cette activité comme étant la méthode *main* qui représente le point de départ de l'exécution d'un program.

Les activités

- Une Activité = un écran graphique
 - Incarne souvent un cas d'utilisation (use case UML)
- Une application est formée de *n* activités.
- Exemple : Application (Phone)
 - Numérateur,
 - Annuaire des contacts
 - Fiche d'un contact
 - Ajout d'un nouveau contact.
- Étant
 - la classe **android.app.Activity** ;y



Cycle de vie d'une Activity

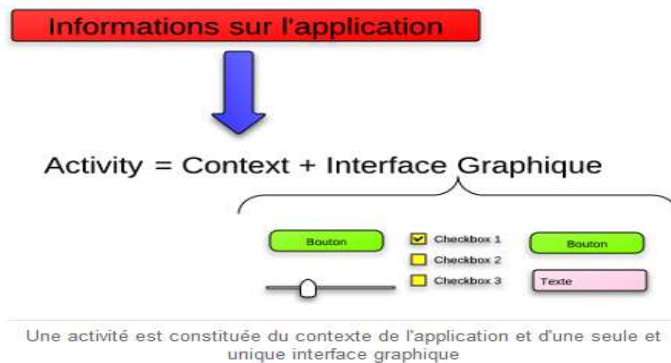


Il est recommandé de bien maîtriser toutes les transitions d'un état à un autre afin d'avoir des activités qui tiennent compte de tous les cas possibles d'interaction ! avec l'utilisateur.

```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

Le contexte d'une activité

- Une activité contient des informations sur l'état actuel de l'application : ces informations s'appellent le *context*.
- Ce *context* constitue un lien avec le système Android ainsi que les autres activités de l'application



Cycle d'une activity

- Pour créer une activité, vous devez créer une sous-classe de **Activity**.
- Dans votre sous-classe, vous avez besoin de mettre en œuvre des méthodes de rappel que le système appelle quand les transitions d'activité entre les différents états de son cycle de vie, tels que :
- Les deux méthodes de rappel les plus importants sont:
 - **onCreate ()**
 - Le système appelle cela lors de la création de l'activité. vous devez initialiser les composants a ce niveau. Plus important encore, ceci est où vous devez appeler **setContentView ()** pour définir l'interface utilisateur de l'activité.
 - **OnPause ()**
 - Le système appelle cette méthode lorsque l'utilisateur quitte l'activité (bien que cela ne signifie pas toujours l'activité est détruite). Cela est généralement le cas, vous devez valider les modifications qui devraient être persisté au-delà de la session de l'utilisateur courant (car l'utilisateur pourrait ne pas revenir)

Cycle d'une activity

- **OnStart()** : après l'exécution de **onCreate**, l'activité est affichée à l'utilisateur, elle est en état « Started »
- **onResume()**: l'activité reçoit maintenant l'interaction de l'utilisateur, elle est dans le sommet de la pile des activités affichées sur l'écran. La plupart des fonctionnalités sont implémentées dans cette méthode.
- **OnStop()**
- **onRestart()**
- **onDestroy()**: important on libère les ressources

sabri.ghazi@univ-annaba.dz

13

Le Intent

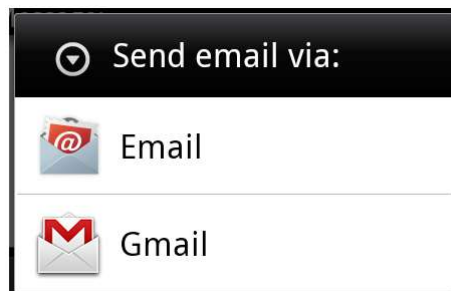
- Un **Intent** est une description abstraite d'une opération qui va être exécutée.
- Exemple, on peut imaginer le cas d'une application qui cherche à ouvrir le gestionnaire des emails via un **intent**, bien sûr le gestionnaire des emails est une autre application, et pour lancer cette dernière on doit procéder comme suit :

sabri.ghazi@univ-annaba.dz

14

Les Intent , exemple envoi d'un email

```
Intent email = new Intent(Intent.ACTION_SEND, Uri.parse("mailto:"));
email.putExtra(Intent.EXTRA_EMAIL, recipients);
email.putExtra(Intent.EXTRA_SUBJECT, subject.getText().toString());
email.putExtra(Intent.EXTRA_TEXT, body.getText().toString());
startActivity(Intent.createChooser(email, "Choose an email client from..."));
```



sabri.ghazi@univ-annaba.dz

15

Intent : exemple lancer une recherche sur le web

```
String q = "photo annaba";
Intent intent = new Intent(Intent.ACTION_WEB_SEARCH);
intent.putExtra(SearchManager.QUERY, q);
startActivity(intent);
```

Ce qui produira comme résultat l'ouverture du navigateur avec la page de recherche avec le terme mis.

sabri.ghazi@univ-annaba.dz

16

Les détails d'un objet Intent

- Un objet **Intent** est une collection d'information qui vont être utilisées par le receveur (une application ou bien le système **Android** lui-même). On peut citer :
 - **Action** : C'est un String qui contient le nom de l'action qui va se lancer, il faut savoir que il existe un nombre d'action prédéfinies par **Android**.
 - **Data**: les paramètres sous format texte de l'action

sabri.ghazi@univ-annaba.dz

17

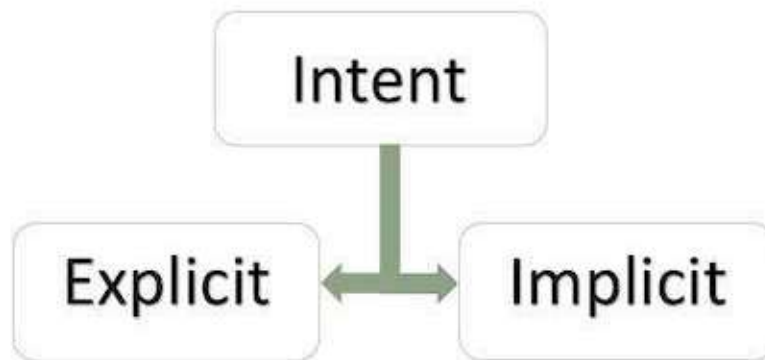
Les détails d'un objet Intent

S.N.	Action/Data Pair & Description
1	ACTION_VIEW content://contacts/people/1 Display information about the person whose identifier is "1".
2	ACTION_DIAL content://contacts/people/1 Display the phone dialer with the person filled in.
3	ACTION_VIEW tel:123 Display the phone dialer with the given number filled in.
4	ACTION_DIAL tel:123 Display the phone dialer with the given number filled in.
5	ACTION_EDIT content://contacts/people/1 Edit information about the person whose identifier is "1".
6	ACTION_VIEW content://contacts/people/ Display a list of people, which the user can browse through.
7	ACTION_SET_WALLPAPER Show settings for choosing wallpaper

sabri.ghazi@univ-annaba.dz

18

Les types d'Intents



sabri.ghazi@univ-annaba.dz

19

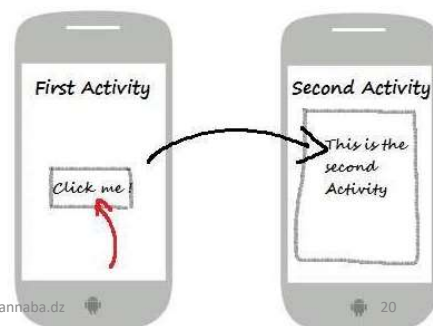
Explicit Intents

- Permet de lier deux **activity** en interne dans l'application.

```

// Explicit Intent by specifying its class name
Intent i = new Intent(FirstActivity.this, SecondActivity.class);

// Starts TargetActivity
startActivity(i);
  
```



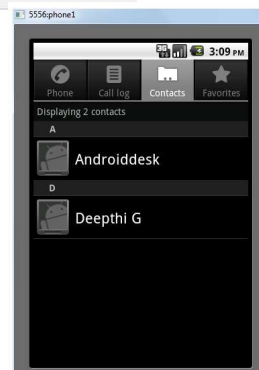
sabri.ghazi@univ-annaba.dz

20

Implicit Intents

- Ce genre **d'intent** n'utilise pas de nom d'activité cible il est utilisé

```
Intent read1=new Intent();  
read1.setAction(android.content.Intent.ACTION_VIEW);  
read1.setData(ContactsContract.Contacts.CONTENT_URI);  
startActivity(read1);
```



sabri.ghazi@univ-annaba.dz

21

Navigation entre activity

- Deux scénarios possibles
 - Naviguer d'une activité A1 vers une activité A2.
 - Afficher une activité A2 à partir d'une activité A1 et récupérer un résultat.

sabri.ghazi@univ-annaba.dz

22

Exemple



Activity A

CHOOSE A CITY?



sabri.ghazi@univ-annaba.dz

23



Activity B

- Annaba
- ALGER
- TAREF
- EL-OUED

BUTTON



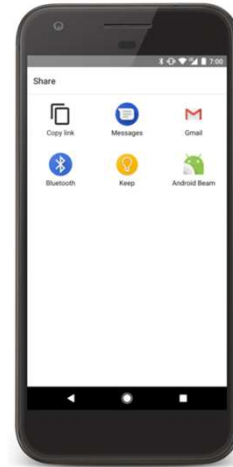
sabri.ghazi@univ-annaba.dz

24

Application Chooser

```
Intent chooser = Intent.createChooser(intent, title);
```

```
try {
    startActivity(chooser);
} catch (ActivityNotFoundException e) {
}
}
```



sabri.ghazi@univ-annaba.dz

25

Ajouter vos activities dans les filter intent du system

- Le système permet aux applications de recter a l'ecoute des certains types d'intent.

Exemple

```
<activity android:name=".ExampleActivity" android:icon="@drawable/app_icon">
  <intent-filter>
    <action android:name="android.intent.action.SEND" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="text/plain" />
  </intent-filter>
</activity>
```

sabri.ghazi@univ-annaba.dz

26