

Android

Introduction au développement Android

ISIMA 2012

Plan

1. Contexte

2. Android ?

3. Architecture logicielle

4. Applications

- Cycle de vie d'une application
- Composants principaux d'android
- Intent

5. Structure d'un projet android

Contexte

- Deux grandes plateformes
 - iOS
 - Android
- Suiveurs
 - Blackberry
- Outsiders
 - Windows Phone 7
 - Bada

Smartphones

- android
 - 48 % de part de marché
 - 293 millions de smartphones vendus en 2011
 - 244 % de progression 2011/2010
- iOS
 - 19% de part de marché
 - 93 millions de smartphones vendus en 2011
- Les autres
 - Symbian (16%), Backberry (10%), Bada (2.7%)

Plan

1. Contexte

2. Android ?

3. Architecture logicielle

4. Applications

- Cycle de vie d'une application
- Composants principaux d'android
- Intent

5. Structure d'un projet android

Android?

- Système d'exploitation basé sur le noyau Linux
 - Profondément modifié
- Développé par l'Open Handset Alliance
 - Google, HTC, Intel, Motorola, Qualcomm...
- Libre
 - Licence Apache

Android

- 139 modèles en vente actuellement
- Tablettes
- Mini PC
- Autoradios, cadres photos, voitures...

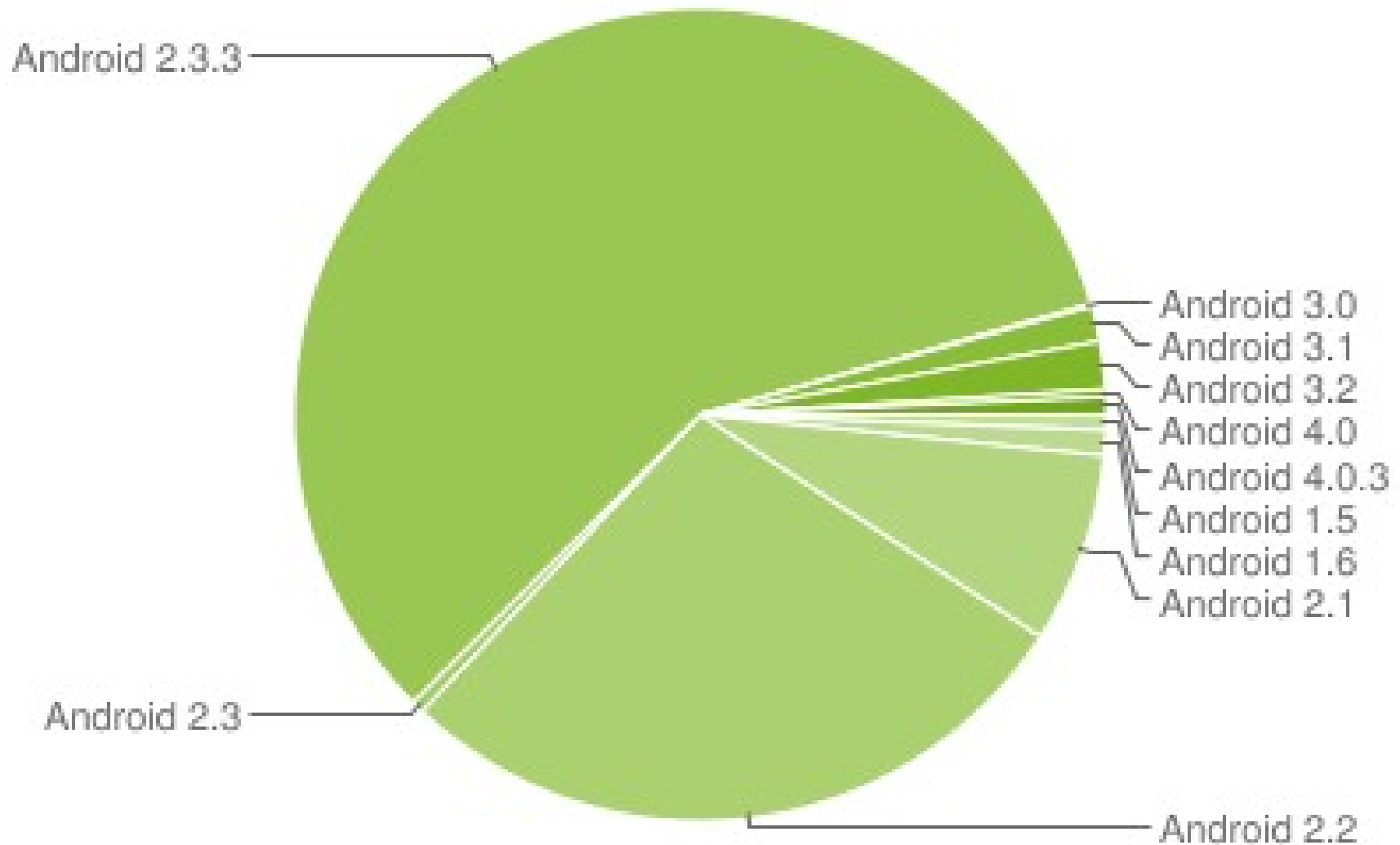
Versions

- 1.5 (API 3)=> Version de base
- 1.6 (API 4)
 - Support d'écrans, Quick Search Box, VPN, TTS
- 2.1 (API 7)
 - LiveWallpaper, exchange, multiple contact accounts, bluetooth, QuickAction, meilleur clavier,...
- 2.2 (API 8)
 - Installation sur SD, tethering, amélioration perf, amélioration bluetooth, HIGHMEM, push, car mode, night mode...

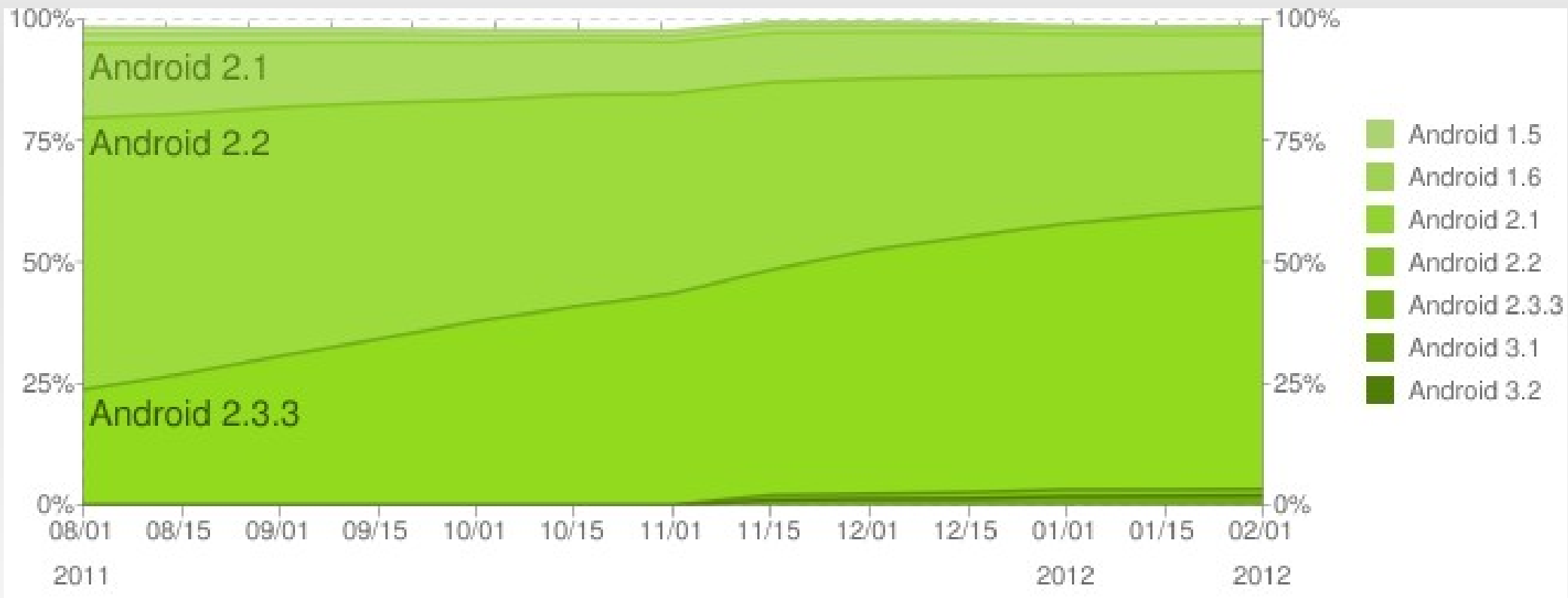
Versions

- 2.3 (API 9)
 - Améliorations graphiques, power management, appels VoIP, NFC, gyroscope,...
- 3.0
 - Spécial tablettes, performances, grands écrans, modifications ergonomiques
- 4.0
 - Réunification de la branche tablette et smartphone
 - Avancées graphiques et ergonomiques

Répartition



Répartition



Accès au market

- CDD
 - Compatibility Definition Document
- CTS
 - Compatibility Test Suite

Plan

1. Contexte

2. Android ?

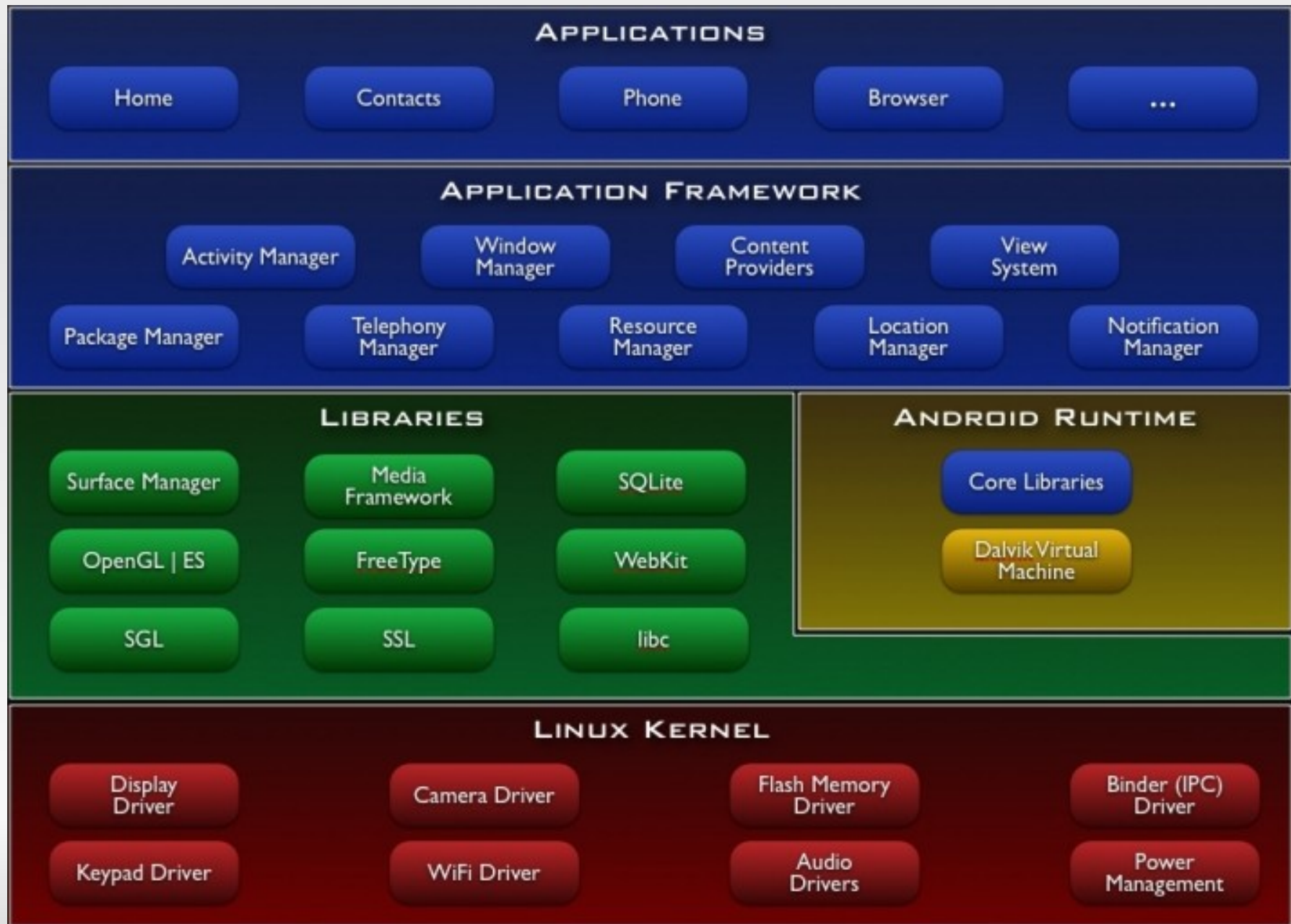
3. Architecture logicielle

4. Applications

- Cycle de vie d'une application
- Composants principaux d'android
- Intent

5. Structure d'un projet android

Architecture logicielle



Applications

- Tout ce qui s'exécute sur android
 - SMS
 - Téléphone
 - Navigateur
 - Maps
 - Spotify
 - Allociné
 - ...

Application framework

- Activity Manager
 - Cycle de vie
 - Pile de fenêtres
- Package Manager
 - Gère ce qui est installé sur la machine
 - Ce que peut faire chaque application



Application framework

- Content Provider
 - Gère le partage de données entre application (contacts, messages...)
- Resource Manager
 - Chaines de caractères spécifique à la langue
 - Images, layouts...



Application framework

- View System
 - View, listes, boutons...
 - Evènements graphiques
 - Layout...
- Location Manager
 - Gestion de position du téléphone
- Notification Manager



Bibliothèques

- Graphique
- Playback audio / video
- Base de donnée
- Langage C



Dalvik

- Machine virtuelle Java android
 - Fichiers Dalvik Executable (.dex)
 - Optimisé pour la consommation mémoire, cpu, utilisation de la batterie...
- 1 instance de machine virtuelle par application

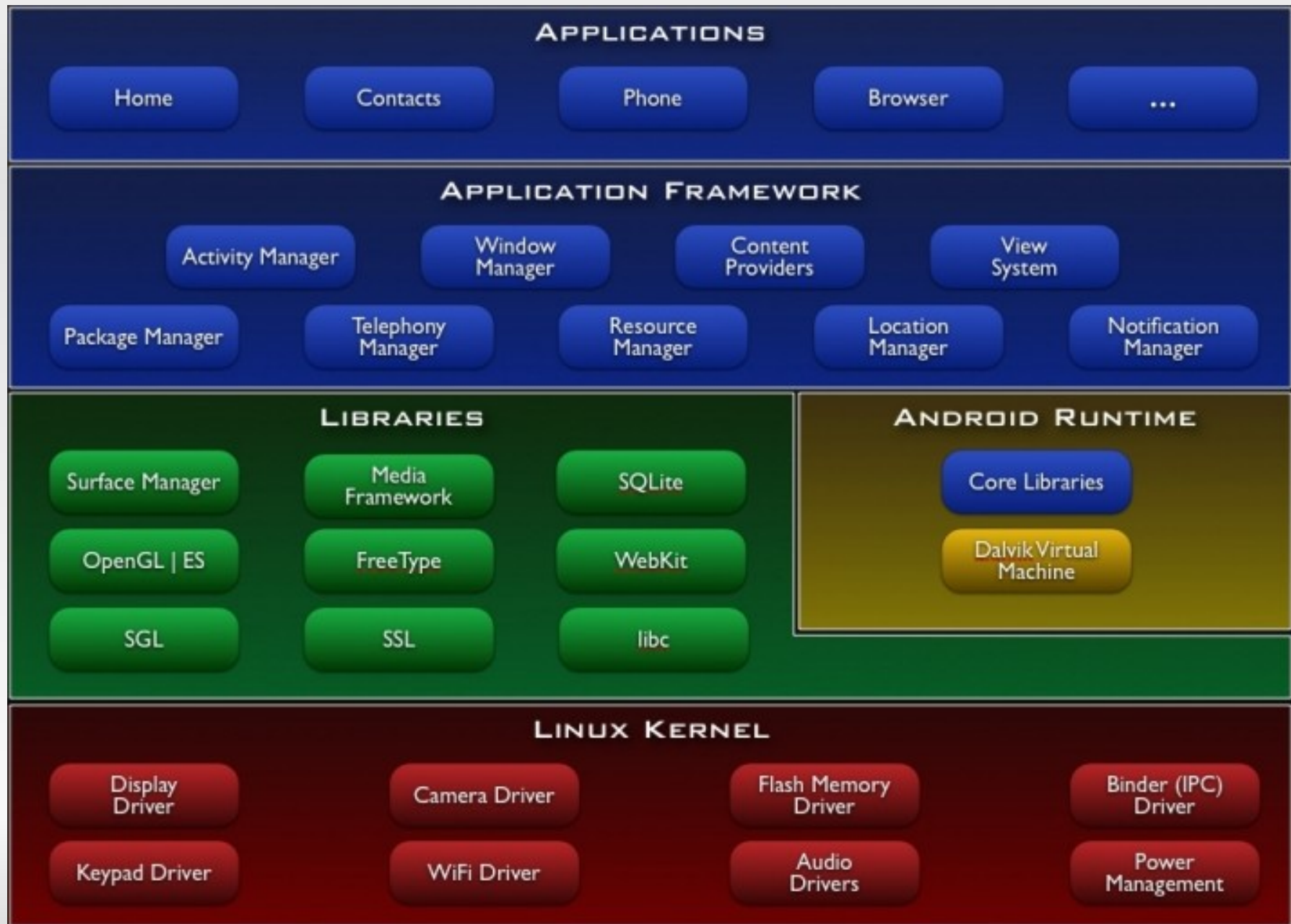


Linux Kernel

- Linux 2.6
- Couche d'abstraction matérielle (HAL)
- Couche à modifier pour développer un nouveau téléphone
 - Drivers (modules)
- Langage C



Architecture logicielle



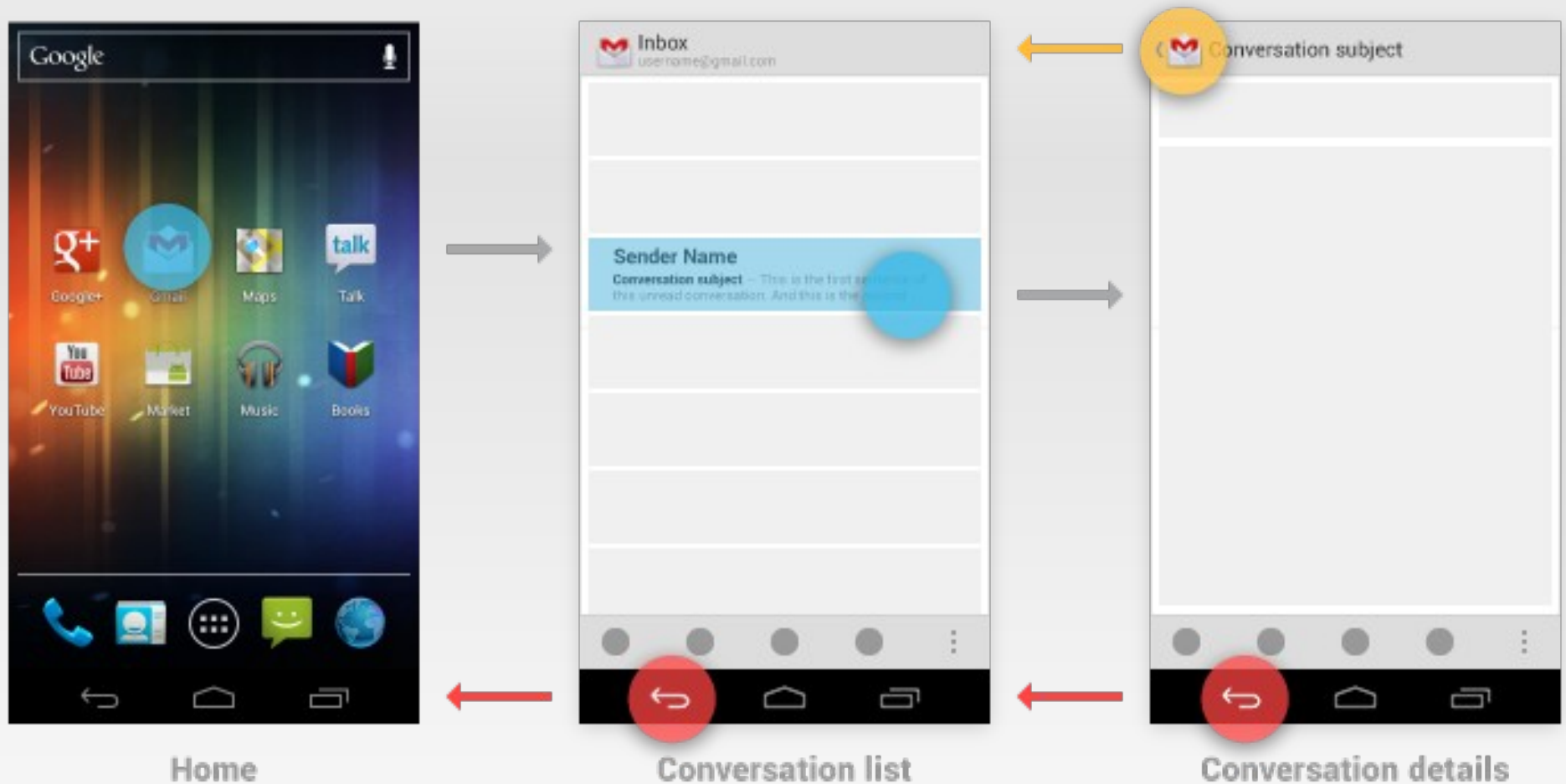
Plan

1. Contexte
2. Android ?
3. Architecture logicielle
4. Applications
 - **Cycle de vie d'une application**
 - Composants principaux d'android
 - Intent
5. Structure d'un projet android

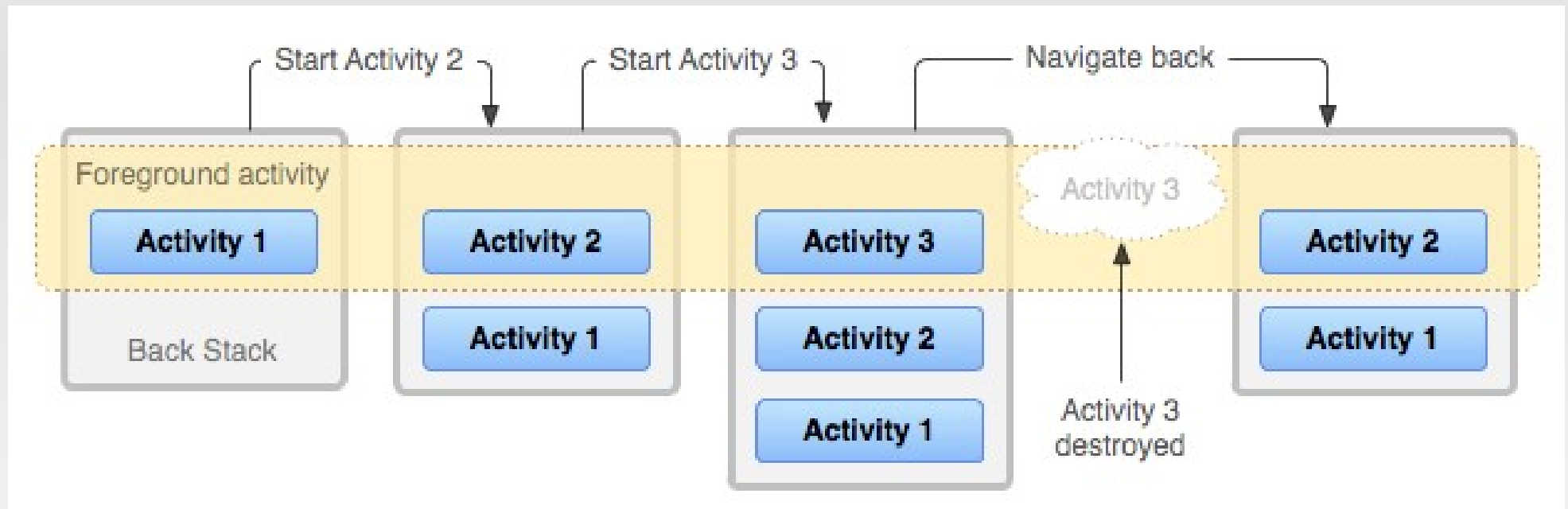
Cycle de vie d'une application

- Chaque application s'exécute dans sa propre machine virtuelle
- Une pile d'activity est maintenue par le système
- Les applications peuvent être tuées par le système pour libérer de la mémoire

Pile d'activity



Pile d'activity



Plan

1. Contexte
2. Android ?
3. Architecture logicielle
4. Applications
 - Cycle de vie d'une application
 - **Composants principaux d'android**
 - Intent
5. Structure d'un projet android

Composants principaux d'android

- Activity
- Service
- Content Providers

Activity

- Fenêtre graphique
- Un écran

A screenshot of a mobile application interface showing a form titled "Serialization". The form contains several input fields for user information: Firstname (with "Scott" entered), Lastname (with "Tiger" entered), Age (with "40" entered), Home address (with "10 bd Google" entered), Line 1, Line 2, Zipcode (with "20010" entered), City (with "Mountain View" entered), and Country (with "USA" entered). At the bottom of the form are two buttons labeled "Get" and "Update". The status bar at the top shows a 3G signal, battery level, and the time 3:45 PM.

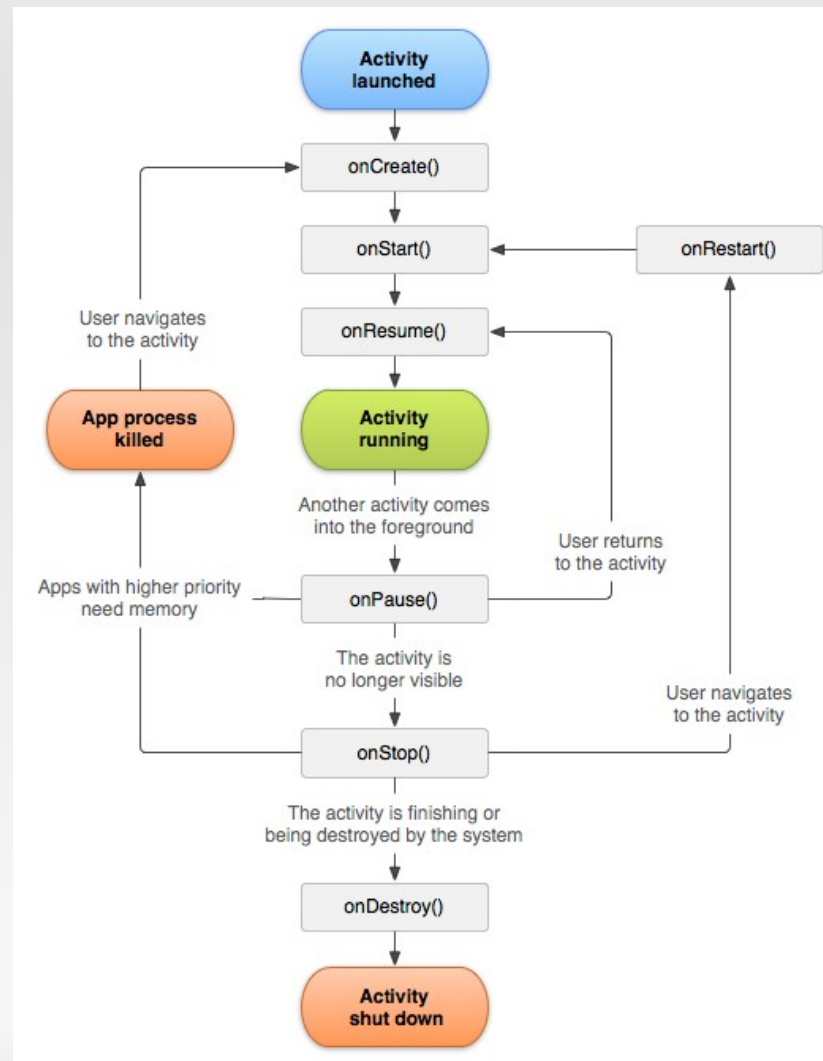
A screenshot of a mobile application interface showing a list of "Upcoming Shows". The header is "Upcoming Shows" in a blue bar. Below the header is a list of shows with their dates, times, names, and locations. The shows are: "The Expendables" (Oct 21, 2008, 7:00 PM, Santa Cruz, CALIFORNIA US), "The Annex" (Oct 21, 2008, 7:00 PM, Madison, Wisconsin), "Station 4" (Oct 23, 2008, 5:30 PM, Saint Paul, Minnesota), "Record Bar" (Oct 24, 2008, 9:00 PM, Kansas City, Missouri), "Gothic Theatre" (Oct 25, 2008, 7:00 PM, Denver, Colorado), and "Belly Up" (Oct 26, 2008, 7:00 PM, Aspen, Colorado). The status bar at the top shows a 3G signal, battery level, and the time 5:02 PM. The bottom of the screen shows a "myspace" logo and the text "Hello, Matt".

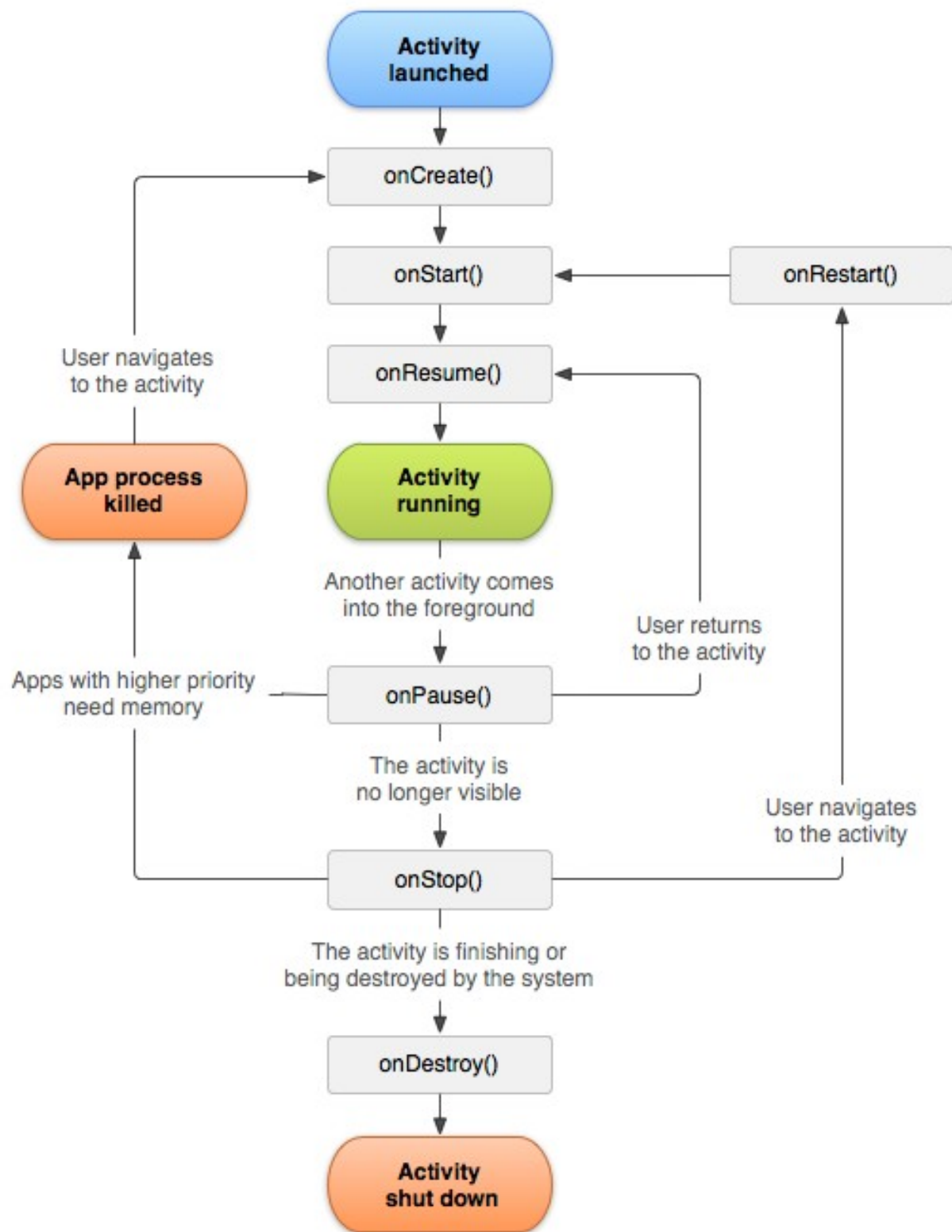
A screenshot of a mobile application interface showing a list of contacts. The header bar contains four icons: TruFriends, Dialer, Call log, and More. Below the header is a list of contacts with their names and status. The contacts are: "Call Test", "Welcome Message", "Voicemail", "Andrew Chang" (Last on-line 2 hrs ago), "Bo Davies" (Last on-line 20 mins ago), and "Dave McCartney" (On-line for 2 hrs). The status bar at the top shows a 3G signal, battery level, and the time 10:08 AM.

Activity

- Classe principale d'une application
- Gère l'affichage des éléments graphiques
- Gère les événements
 - Système
 - Événements utilisateur

Activity : cycle de vie

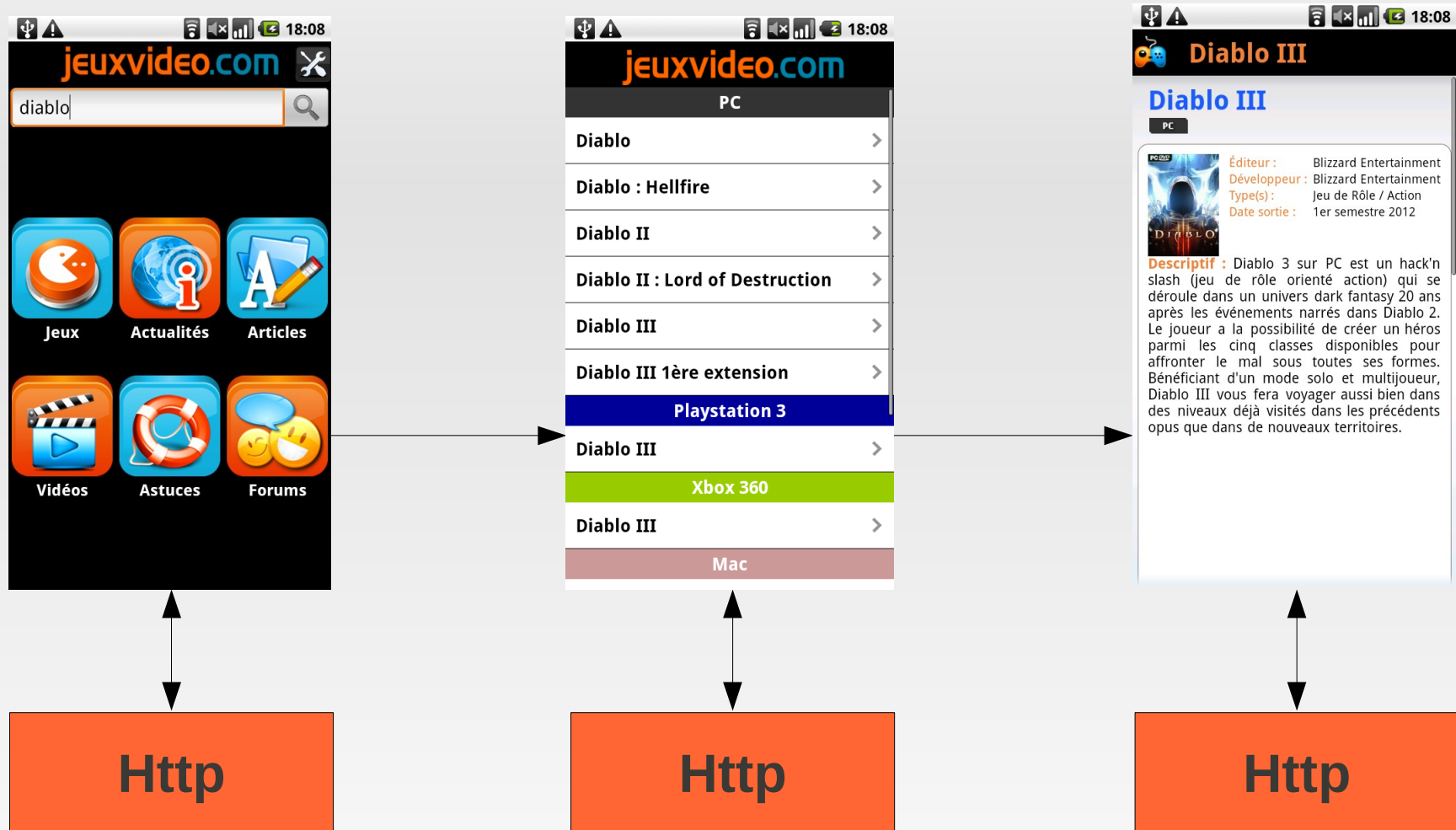




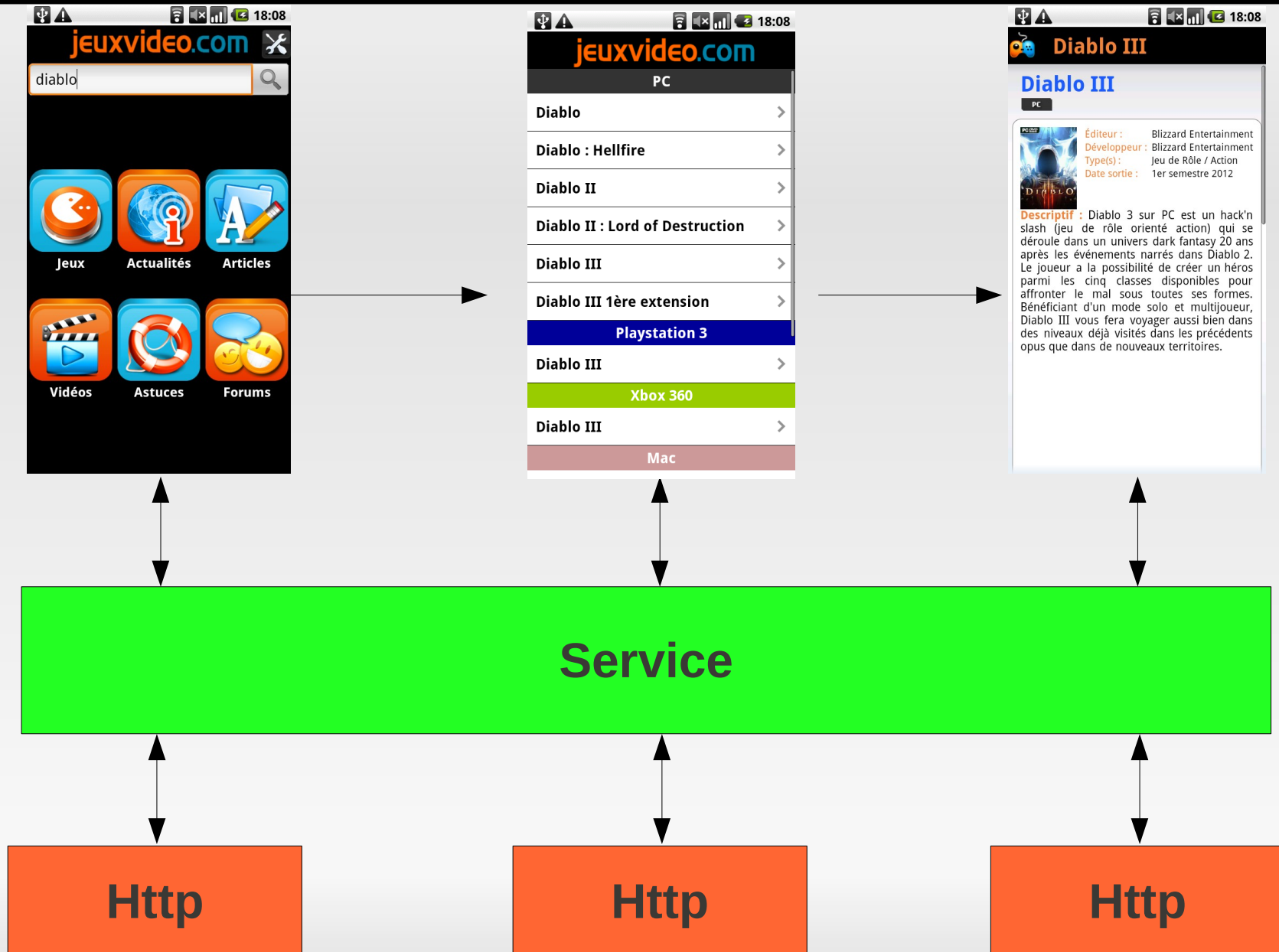
Service

- Application qui n'a pas d'interface graphique
- Tourne en arrière plan
- Exemples
 - Lecture de musique
 - Sport tracker
 - Appels à des services web
 - ...

Service



Service



Content provider

- API d'accès à une base de donnée
- Contient toute sortes de données
 - Contacts
 - Sms
 - Mails
 - Photos
- Permet de partager les données entre plusieurs applications

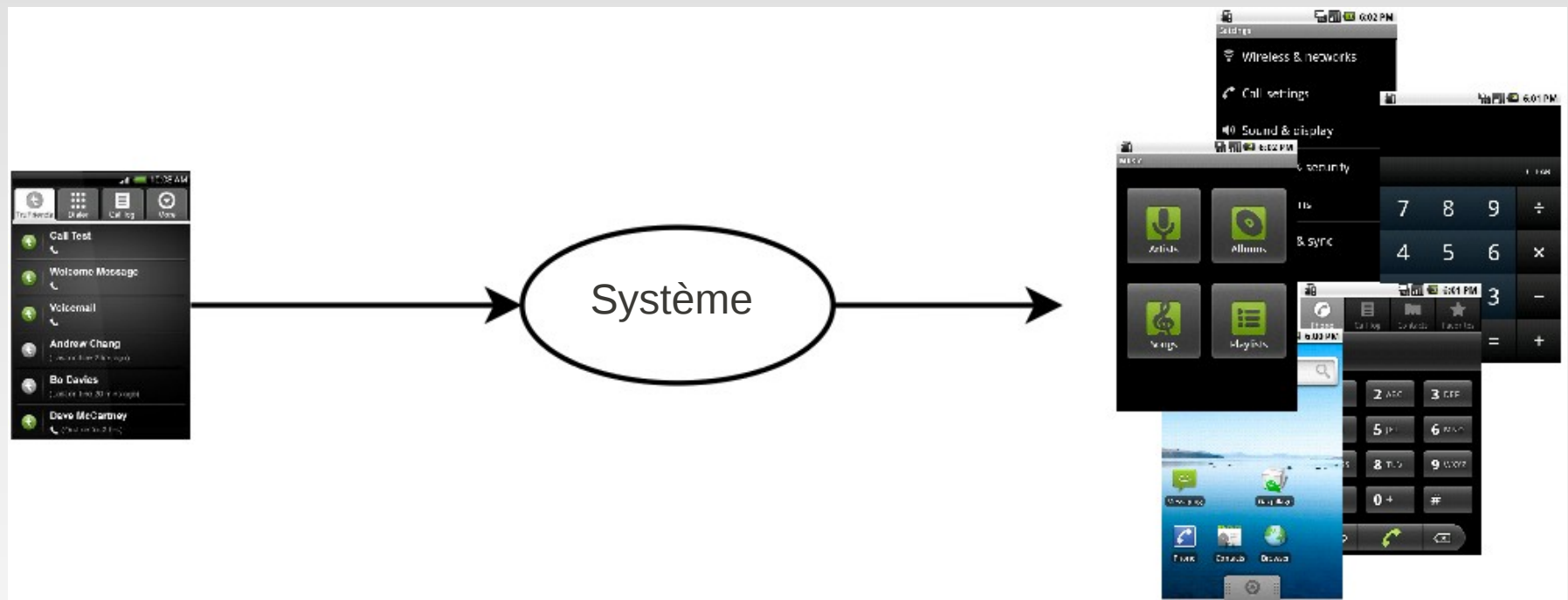
Plan

1. Contexte
2. Android ?
3. Architecture logicielle
4. Applications
 - Cycle de vie d'une application
 - Composants principaux d'android
 - **Intent**
5. Structure d'un projet android

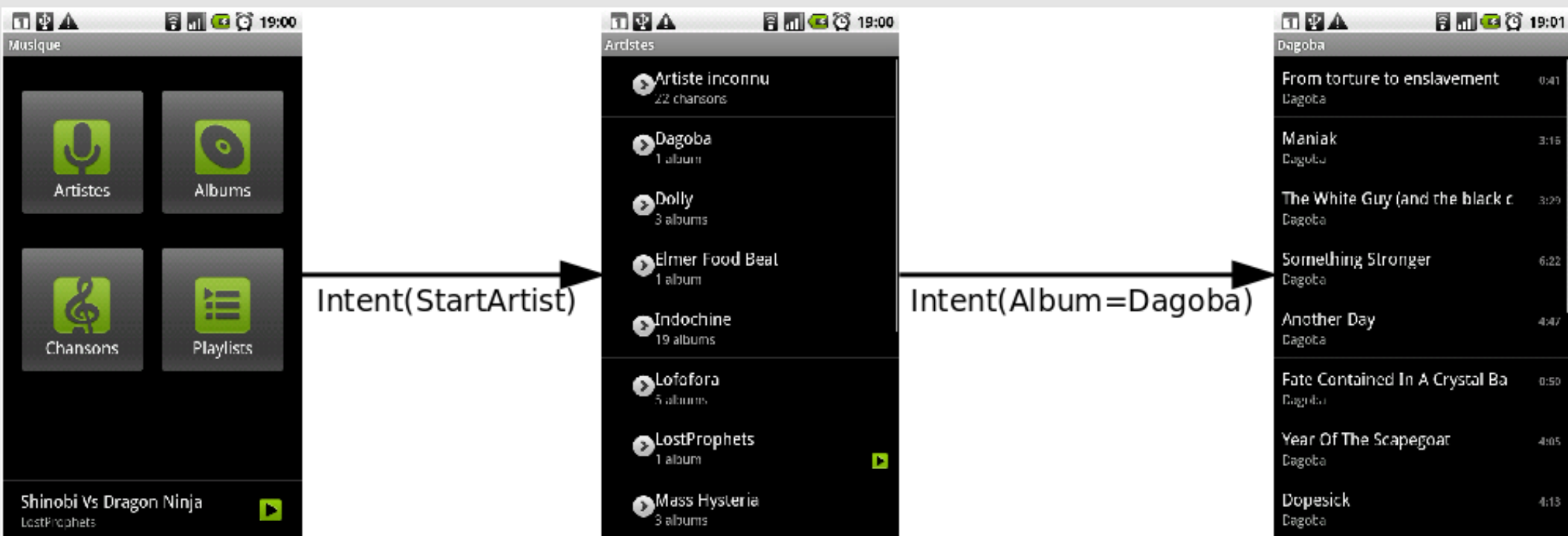
Interaction entre activity

- 1 activity => 1 fenêtre
- Intent à destination
 - D'une autre application
 - D'une autre activity de notre application
 - D'un service
- Il est traité par le système

Résolution d'intent



Interactions entre activity



Intent explicite / implicite

- Intent Explicite

```
Intent i = new Intent(this, ActivityVideo.class);  
startActivity(i);
```

- Intent Implicite

```
Intent i = new Intent("regarderUneVideo");  
startActivity(i);
```

Contenu d'un intent

- Action
 - Ce qu'il doit faire (CALL, EDIT, VIEW,...)
- Data
 - Numéro de téléphone à appeler
 - Numéro du contact à éditer
 - Page Http à visualiser
- Extra
 - Données supplémentaires sous forme de clé/valeur

Intent: les extras

- Un intent peut contenir des données
 - int, boolean, string, ...
 - Objet sérialisable

```
Intent i = new Intent("afficherResultat");  
i.putExtra("id_resultat", 1234);  
i.putExtra("donnees", unObjet);  
startActivity(i);
```

Intent: Récupérer les données

- Récupérer les données venant d'un intent

```
public void onCreate(Bundle savedInstanceState) {  
    ...  
    Intent i = getIntent();  
    int idResultat = i.getIntExtra("id_resultat",-1);  
    Donnee d = (Donnee) i .getSerializableExtra("donnees");  
    ...  
}
```

Plan

1. Contexte

2. Android ?

3. Architecture logicielle

4. Applications

- Cycle de vie d'une application
- Composants principaux d'android
- Intent

5. Structure d'un projet android

Composants d'une application



Values

- Chaines de caractères
 - string.xml
- Localisation en fonction du nom du dossier
 - values-fr/ => Français
 - values-pt/ => Portugal
 - Les mêmes chaines doivent être dans les différents dossier
 - values/ par défaut

String.xml

- ▼ Test
 - src
 - gen [Generated Java Files]
 - Android 2.0.1
 - assets
 - ▼ res
 - drawable-hdpi
 - drawable-ldpi
 - drawable-mdpi
 - layout
 - raw
 - ▼ values
 - strings.xml
 - ▼ values-en
 - strings.xml
 - ▼ values-fr
 - strings.xml
 - AndroidManifest.xml
 - default.properties

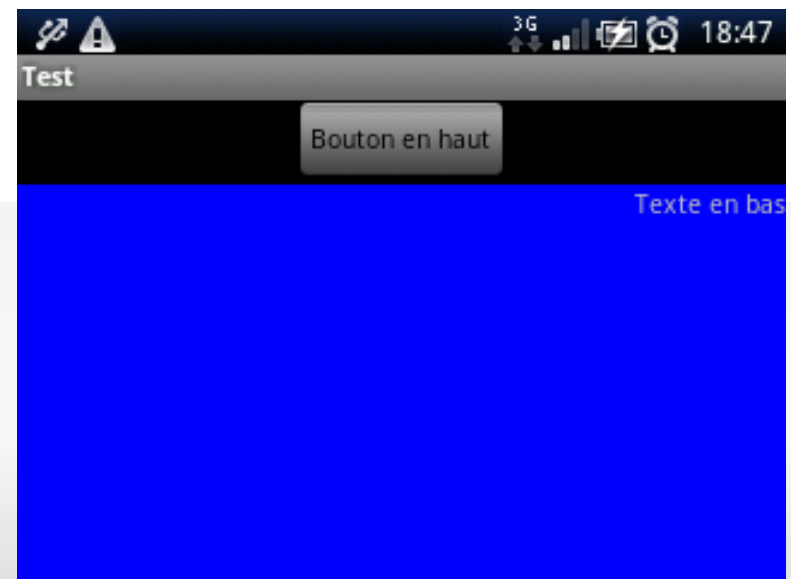
```
1<?xml version="1.0" encoding="utf-8"?>
2<resources>
3    <string name="app_name">Test</string>
4    <string name="dataToDisplay">No Data</string>
5    <string name="boutonPlay">Play</string>
6    <string name="boutonPause">Pause</string>
7</resources>
```


Layout.xml

- Architecture de l'interface utilisateur
 - Définition des éléments et de leur position
- Différents layout de placement
 - Linear Layout
 - Relative Layout
 - Frame Layout
 - Table Layout
 - ...

Exemple 1

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent" android:layout_height="fill_parent">
    <Button
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Bouton en haut">
    </Button>
    <TextView android:id="@+id/MaTextView1"
        android:layout_width="fill_parent" android:layout_height="fill_parent"
        android:text="Texte en bas"
        android:gravity="right"
        android:background="#0000FF">
    </TextView>
</LinearLayout>
```



```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/home_list_item_RelativeLayout"
    android:layout_width="fill_parent" android:layout_height="90dip">
    <ImageView
        android:id="@+id/preview"
        android:layout_width="90dip" android:layout_height="90dip"
        android:layout_alignParentRight="true" android:padding="2dip">
    </ImageView>
    <TextView
        android:id="@+id/title"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/preview"
        android:gravity="left" android:maxLines="2"
        android:textSize="15sp" android:textStyle="bold" >
    </TextView>
    <TextView
        android:id="@+id/date"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/title" android:layout_toLeftOf="@+id/preview"
        android:gravity="left"
        android:singleLine="true">
    </TextView>
    <TextView
        android:id="@+id/home_list_item_TextView_Content"
        android:layout_width="fill_parent" android:layout_height="fill_parent"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/date" android:layout_toLeftOf="@+id/preview"
        android:gravity="left"
        android:maxLines="2" android:textSize="12sp" >
    </TextView>
</RelativeLayout>

```

Clermont-Ferrand et Groland officiellement jumelées [vi...

Le 02/02/12 à 20:48

L'équipe de Groland (Canal Plus) était quasiment au complet pour une céré...



AndroidManifest.xml

- Déclare les éléments de l'application
 - Activity
 - Service
 - Broadcast receiver
- Déclare les autorisation d'accès
- Version de l'API
- Utilisation de tests unitaires
- ...

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.openium.gazpillage"
    android:versionCode="1"
    android:versionName="1.0" >
    <application
        android:icon="@drawable/icon" android:label="@string/app_name" >
        <activity android:name=".ActivityWelcome" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ActivityMain" />
        <activity android:name=".ActivityUserCarStatistics" />
        <activity android:name=".ActivityVideo" >
            <intent-filter >
                <action android:name="regarderUneVideo" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="9" android:targetSdkVersion="14" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
</manifest>
```

Activity class

- Classe clé
- On choisit le layout à afficher
 - `setContentView(R.layout.monLayout);`
- On enregistre les différents listener
- On gère les évènements
- On gère le cycle de vie

Listeners

- Boutons:
 - OnClickListener / onClick
- Listes:
 - OnItemClickListener / onItemClick
 - OnItemSelectedListener / onItemSelected
- Clavier
 - Methode OnKeyUp/ OnKeyDown
- ...

Activity class

- "Récupérer" les éléments graphiques du layout

```
public class test extends Activity implements OnClickListener{

    TextView mTextView;
    Button mButtonPlay;
    Button mButtonPause;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mTextView = (TextView) findViewById(R.id.TextViewData);
        mButtonPlay = (Button) findViewById(R.id.ButtonPlay);
        mButtonPause = (Button) findViewById(R.id.ButtonPause);

        mTextView.setText("Un nouveau texte");

        mButtonPlay.setOnClickListener(this);
        mButtonPause.setOnClickListener(this);
    }

    public void onClick(View v)
    {
        if (mButtonPlay == v)
        {
            // Do something
        }
        else if(mButtonPause == v)
        {
            // Do something else
        }
    }
}
```



```

public class test extends Activity implements OnClickListener{

    TextView mTextView;
    Button mButtonPlay;
    Button mButtonPause;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mTextView = (TextView) findViewById(R.id.TextViewData);
        mButtonPlay = (Button) findViewById(R.id.ButtonPlay);
        mButtonPause = (Button) findViewById(R.id.ButtonPause);

        mTextView.setText("Un nouveau texte");

        mButtonPlay.setOnClickListener(this);
        mButtonPause.setOnClickListener(this);
    }

    public void onClick(View v)
    {
        if (mButtonPlay == v)
        {
            // Do something
        }
        else if(mButtonPause == v)
        {
            // Do something else
        }
    }
}

```

Pour continuer

<http://developer.android.com>