



Systemy operacyjne na platformach mobilnych

Wykład 1

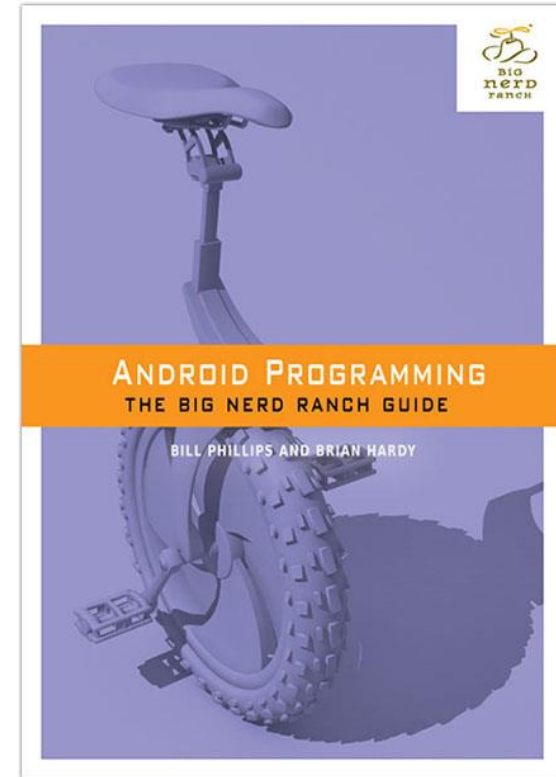
*Grzegorz Jabłoński, Piotr Perek
Katedra Mikroelektroniki i Technik Informatycznych*

Zagadnienia wykładu

- Wprowadzenie do systemów mobilnych
- System operacyjny Android
- Różnice pomiędzy systemami GNU/Linux i Android
- Język Java i wirtualna maszyna Dalvik
- Podstawowe elementy programistyczne systemu Android: Activity, Intent, Content Provider

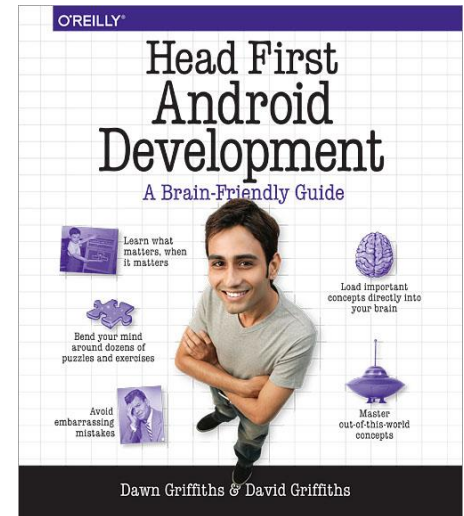
Literatura

- “Android Programming: The Big Nerd Ranch Guide, 2nd ed.”, Bill Phillips, Chris Stewart, Brian Hardy and Kristin Marsicano, 2015



Literatura

- “Head First Android Development”, Dawn Griffiths, David Griffiths, O'Reilly Media, June 2015
- “The Busy Coder's Guide to Android Development”, Mark L. Murphy, CommonsWare, LLC, 2015
- developer.android.com



Mobile Operating Systems

Operating System	4Q15 Shipment Volumes	4Q15 Market Share	4Q14 Shipment Volumes	4Q14 Market Share
Android	325.4	80.7%	279.1	76.0%
iOS	71.5	17.7%	74.8	20.4%
Windows Phone	4.4	1.1%	10.4	2.8%
BlackBerry	0.9	0.2%	1.7	0.5%
Others	0.9	0.2%	1.3	0.4%
Total	403.1	100.0%	367.3	100.0%

Source:<http://www.gartner.com/newsroom/id/3215217>

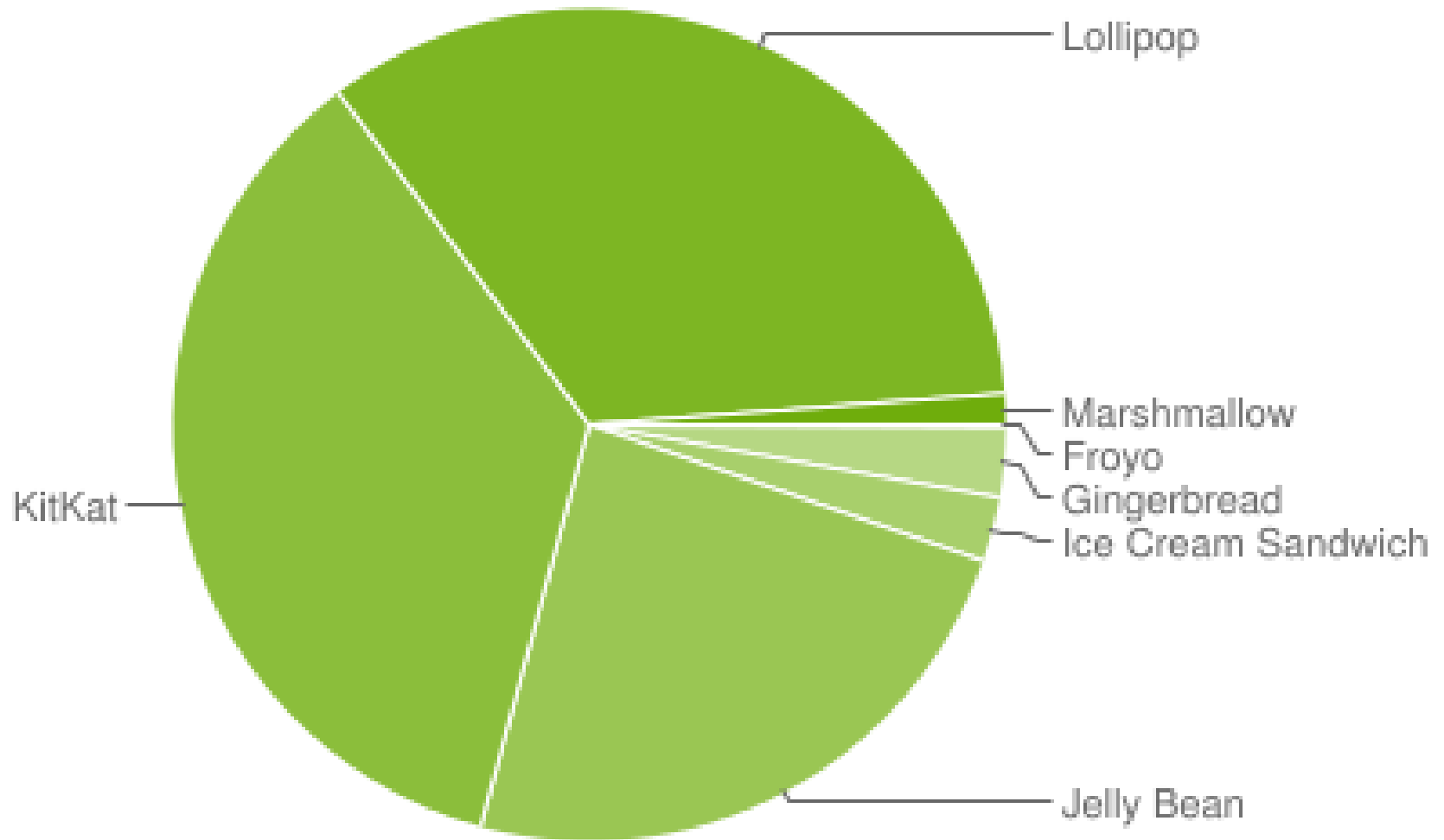
Czym jest Android?

- Android wg Andy'ego Rubina z Google'a:
“The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation”
<http://googleblog.blogspot.com/2007/11/wheres-my-gphone.html>
- Pojęcie “Android” obejmuje:
 - darmowy, system operacyjny na urządzenia mobilne o otwartych źródłach
 - platformę open-source do rozwoju aplikacji mobilnych
 - urządzenia – w tym telefony komórkowe – na których działa system operacyjny Android i napisane dla niego aplikacje

Wersje Androida

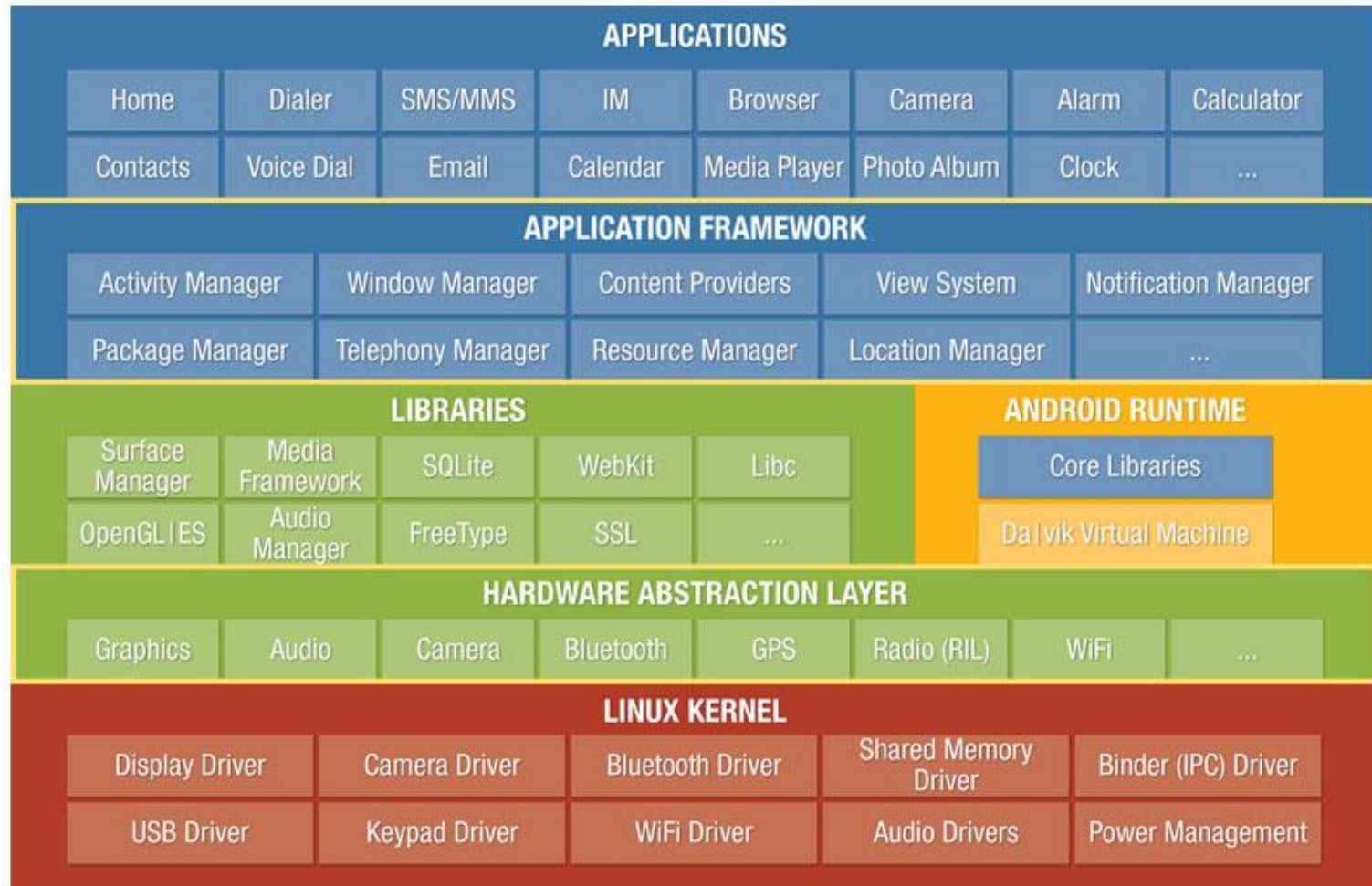
Numer wersji	Data wydania	Nazwa wersji
1.1	luty 2009	
1.5	kwiecień 2009	Cupcake
1.6	wrzesień 2009	Donut
2.0/2.1	październik 2009	Eclair
2.2	maj 2010	Froyo
2.3	grudzień 2010	Gingerbread
3.0/3.1/3.2	styczeń 2011	Honeycomb
4.0	październik 2011	Ice Cream Sandwich
4.1/4.2/4.3	listopad 2012	Jelly Bean
4.4	październik 2013	KitKat
5.0/5.0.1	grudzień 2014	Lollipop
5.0.1	grudzień 2014	Lollipop
6.0 - 6.0.1	październik 2015	Marshmallow

Wersje Androida



<http://developer.android.com/about/dashboards/index.html>

Android – stos programowy



Języki programowania

- Podstawowy język – Java
- Możliwość programowania w języku Scala
- Budowanie aplikacji webowych (JavaScript, CSS, ...)
- Programowanie w języku natywnym C/C++

Podstawowe komponenty aplikacji

- **Activity** – aktywność – interfejs użytkownika
- **Service** – serwis – aplikacja usługi (posiada długi czas życia)
- **BroadcastReceiver** – aplikacja odbierająca zdarzenia globalne i reagująca na nie
- **ContentProvider** – dostawca treści – dostarcza dane dla aplikacji (obiektów Activity i/lub Service)
- Obiekty aktywujące poszczególne komponenty lub przenoszące informacje pomiędzy komponentami nazywane są **intencjami**

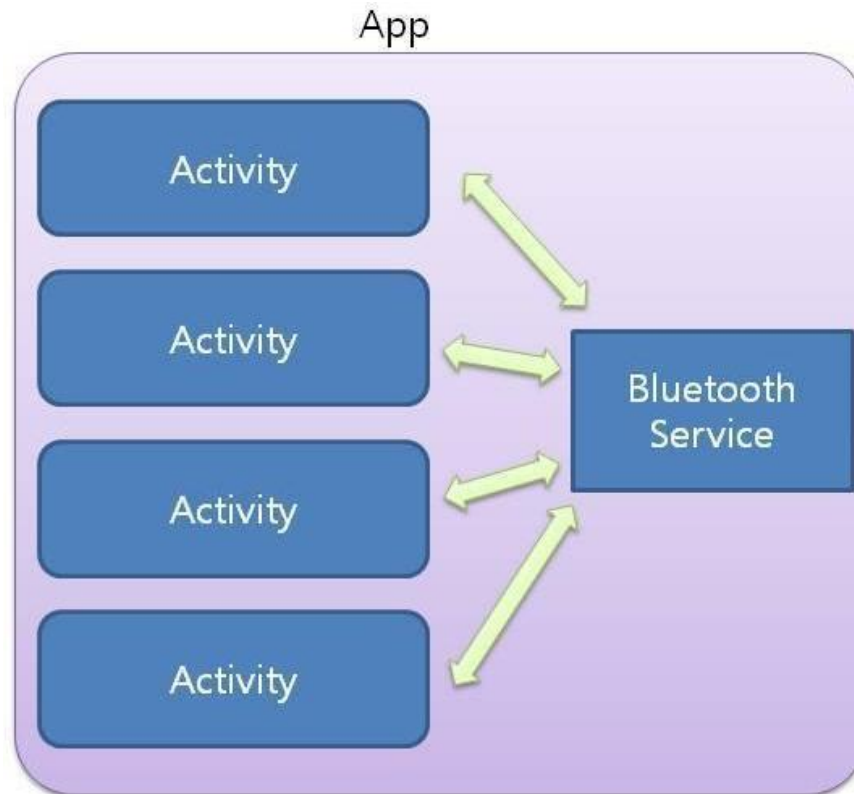
Activity

O Activity należy myśleć jak o widocznym ekranie – reprezentuje ona widoczny komponent aplikacji.

- Aplikacje mogą zawierać więcej niż jedną Activity
- Klasa Activity korzysta z co najmniej jednego obiektu
- View do prezentacji elementów użytkownika (elementy definiowane są w pliku XML)
- Przechodzenie z jednego do drugiego obiektu Activity jest realizowane przez wywołanie metody `startActivity()` lub `startActivityForResult()` z obiektem intencji `Intent` zadany jako argument.

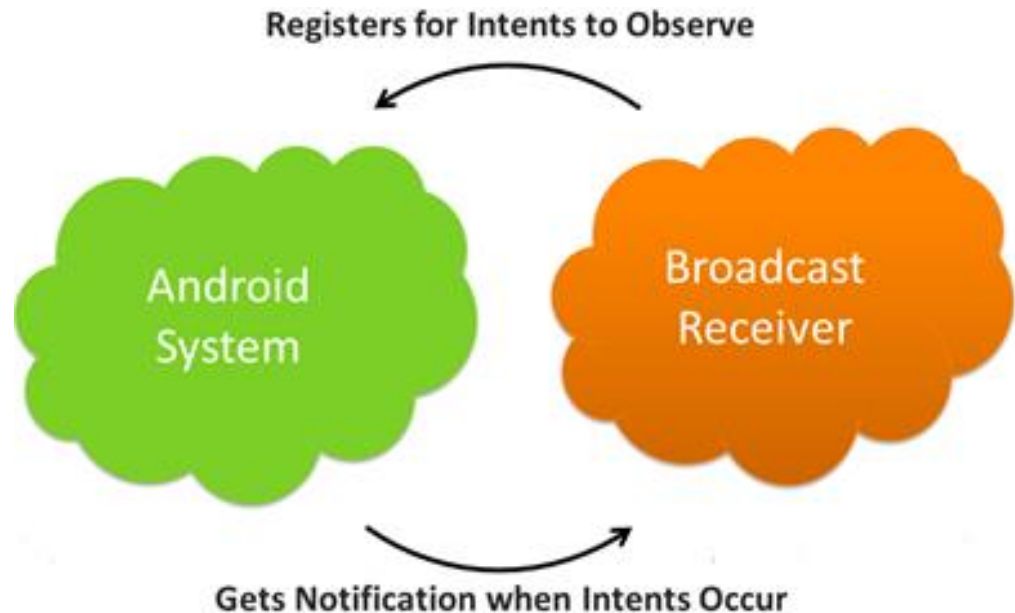
Service

- Serwis wykonuje usługi dla aplikacji



BroadcastReceiver

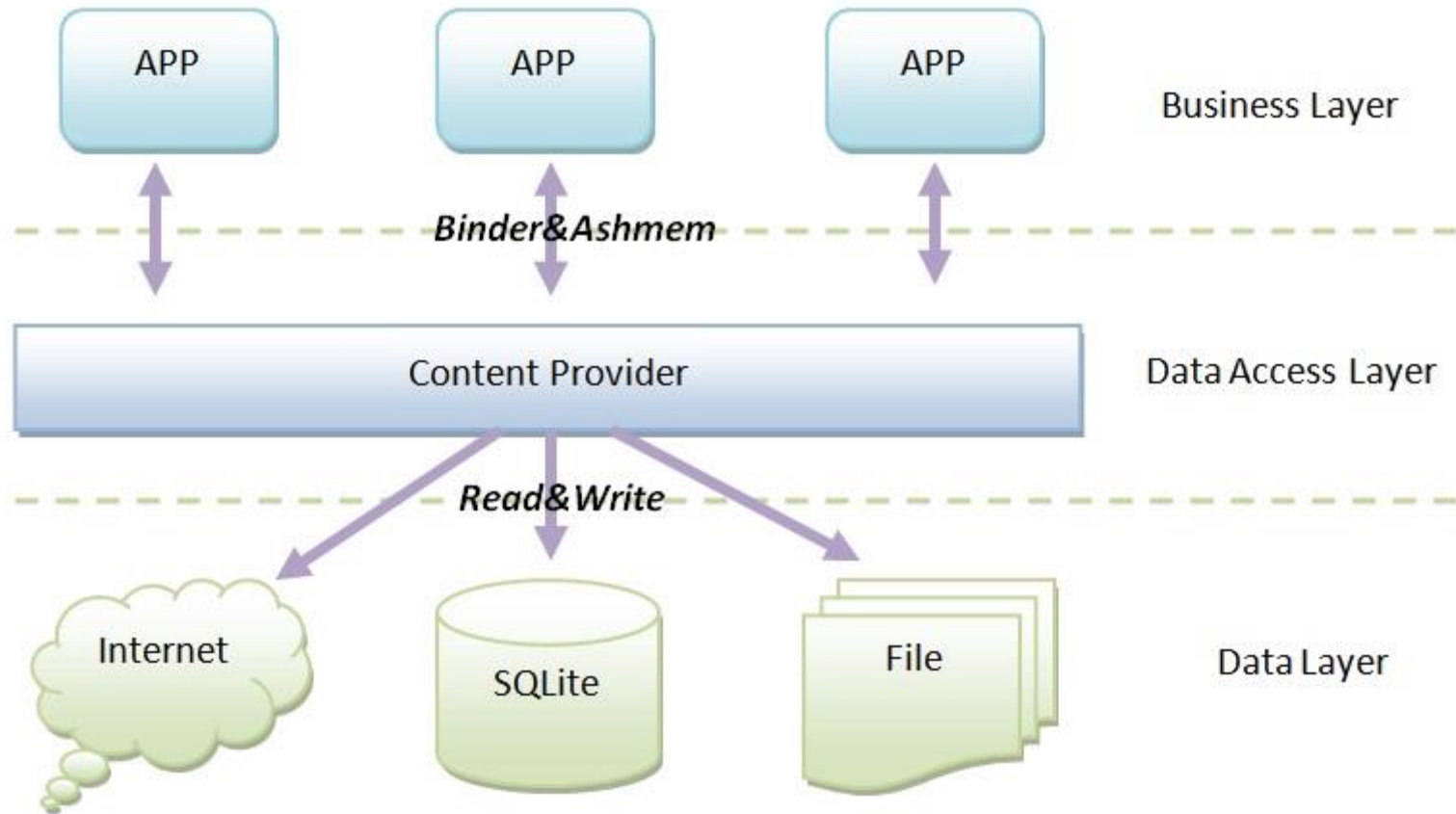
- Dzięki BroadcastReceiver aplikacja może reagować na określone zdarzenia w systemie



ContentProvider

- Implementuje standardowy zbiór metod pozwalających aplikacji na dostęp do magazynu danych.
 - dostęp umożliwia operacje odczytu, zapisu lub obie naraz.
- Może dostarczać dane tej samej aplikacji jak i innym aplikacjom.

ContentProvider



Intencje

- Najważniejszy aspekt kodowania
- Oznacza czynność jaką chcemy wykonać
- Intencje mają postać klasy Intent
- Filtry intencji mają postać klasy IntentFilter
- Intencje przesyłane są do aplikacji androida zarejestrowanych jako IntentFilter

*Jeśli zrozumiesz intencje zrozumiesz
programowanie dla Androida*

Intencje

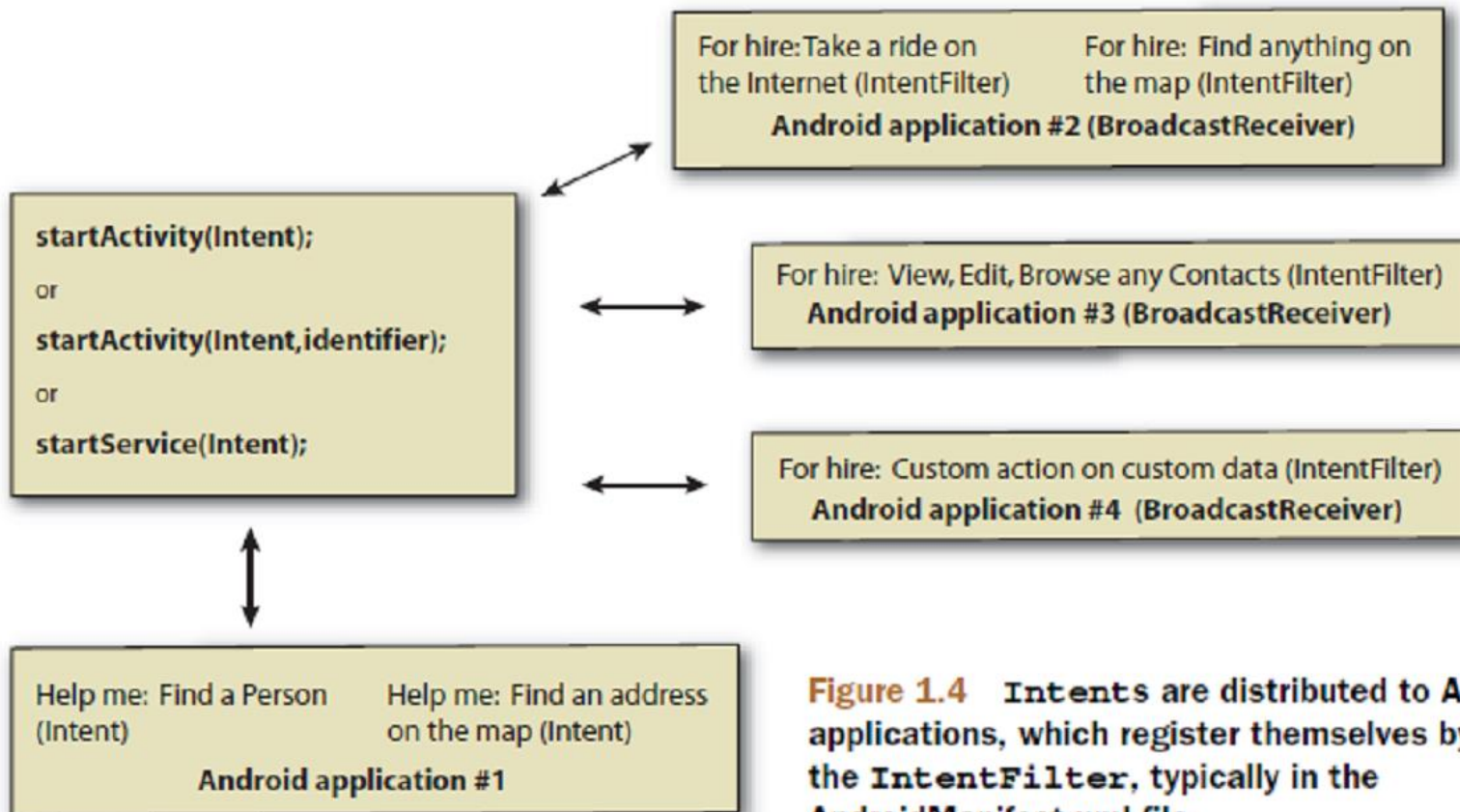


Figure 1.4 Intents are distributed to Android applications, which register themselves by way of the **IntentFilter**, typically in the **AndroidManifest.xml** file.

Plik AndroidManifest.xml

- Pełni rolę mechanizmu spajającego relacje między czterema podstawowymi elementami aplikacji Android.
- Jest deskryptorem aplikacji
- Znajduje się w głównym katalogu aplikacji

Budowa pliku AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.msi.manning.unlockingandroid">
    <application android:icon="@drawable/icon">
        <activity android:name=".Activity1"
android:label="@string/app_name">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Trzy główne podejścia w programowaniu interfejsu użytkownika

- Java-based
 - Używana Java, do definiowania napisu, rozkładu okna, kontrolek GUI i przypisania handlerów zdarzeń
- XML-based
 - Używany plik XML do wykonania powyższych operacji
 - Metody Java odczytują plik i podejmą odpowiednie akcje
- Hybrydowy
 - Używana Java i/lub plik XML

Podjęcie Java, przykład

```
public class SomeName extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        String message = "...";
        LinearLayout window = new LinearLayout(this);
        window.setVariousAttributes(...);
        Button b = new Button(this);
        b.setText("Button Label");
        b.setOnClickListener(new SomeHandler());
        mainWindow.addView(b);
        ...
        setContentView(window);
    }

    private class SomeHandler implements OnClickListener {
        @Override
        public void onClick(View clickedButton) {
            doSomething(...);
        }
    }
}
```

Podójście XML, przykłád

- Java

```
public class SomeClass extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    public void handlerMethod(View clickedButton) {
        String someName = getString(R.string.some_name);
        doSomethingWith(someName);
    }
}
```

- XML

res/values/strings.xml	res/layout/main.xml
<pre><?xml version="1.0" encoding="utf-8"?> <resources> <string name="some_name">...</string> ... </resources></pre>	<pre><?xml version="1.0" encoding="utf-8"?> <LinearLayout ...> <TextView ... /> <Button ... android:onClick="handlerMethod" /> </LinearLayout></pre>

Narzędzia programistyczne

Do budowy aplikacji Androida należy zainstalować:

- JDK (w wersji co najmniej 7)

<http://www.oracle.com/technetwork/java/javase/downloads/>

- Android SDK

<http://developer.android.com/sdk/index.html>

- opcjonalnie środowisko IDE: Android Studio

Podstawowa struktura programu

```
package com.companyname.projectname;
import android.app.Activity;
import android.os.Bundle;
import android.widget.SomeLayoutOrView;
public class SomeName extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        SomeLayoutOrView view = createOrGetView();
        ...
        setContentView(view);
    }
    ...
}
```