



# PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

## WYKŁAD 1

- Podstawowe Elementy Aplikacji
- Cykl Życia
- Intent
- Context



Uniwersytet  
Wrocławski

# Podstawowe Elementy Aplikacji

Podstawowe elementy – komponenty, klasy pierwotne z których zbudowana jest aplikacja (VM + Zasoby = APK (Android Package Kit))

Każdy komponent zapewnia konkretną funkcjonalność i posiada wyodrębniony cykl życia. Komponenty działają kooperacyjnie wspólnie zapewniając ukończenie określonego zadania aplikacji

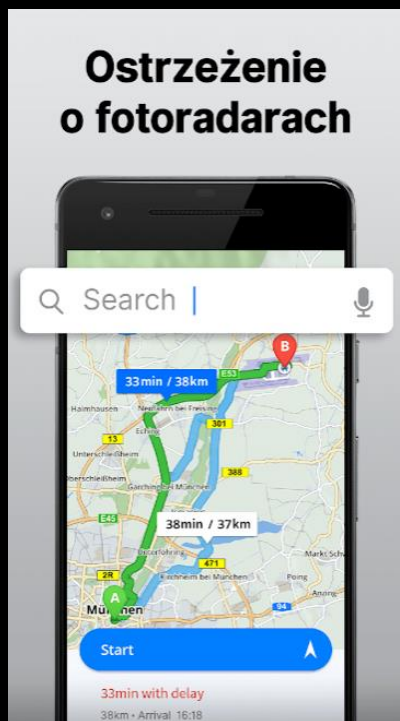
Podstawowe elementy:

1. Activities/Fragments – pojedynczy ekran z UI
2. Services – long running tasks in the background
3. Broadcast Receiver – responds to system-wide announcements
4. Content Provider

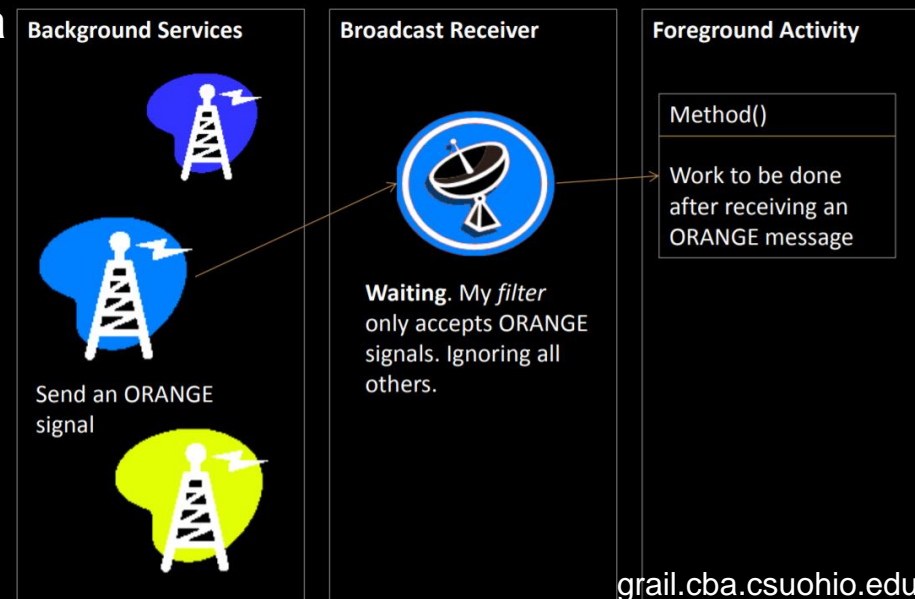
- Activity Class – obiekt, pojedyncze GUI, które poza wyświetlaniem / zbieraniem danych zapewnia pewną funkcjonalność
- Typowo aplikacje zawierają jeden lub więcej obiektów *Activity*
- Aplikacje muszą wyznaczyć jedną czynność jako *main task/entry point*. Ta czynność jest uruchamiana jako pierwsza gdy aplikacja jest włączana
- Czynność może przekazać kontrolę i dane do innej czynności poprzez protokół komunikacji międzyprocesowej *intents* - proces logowania



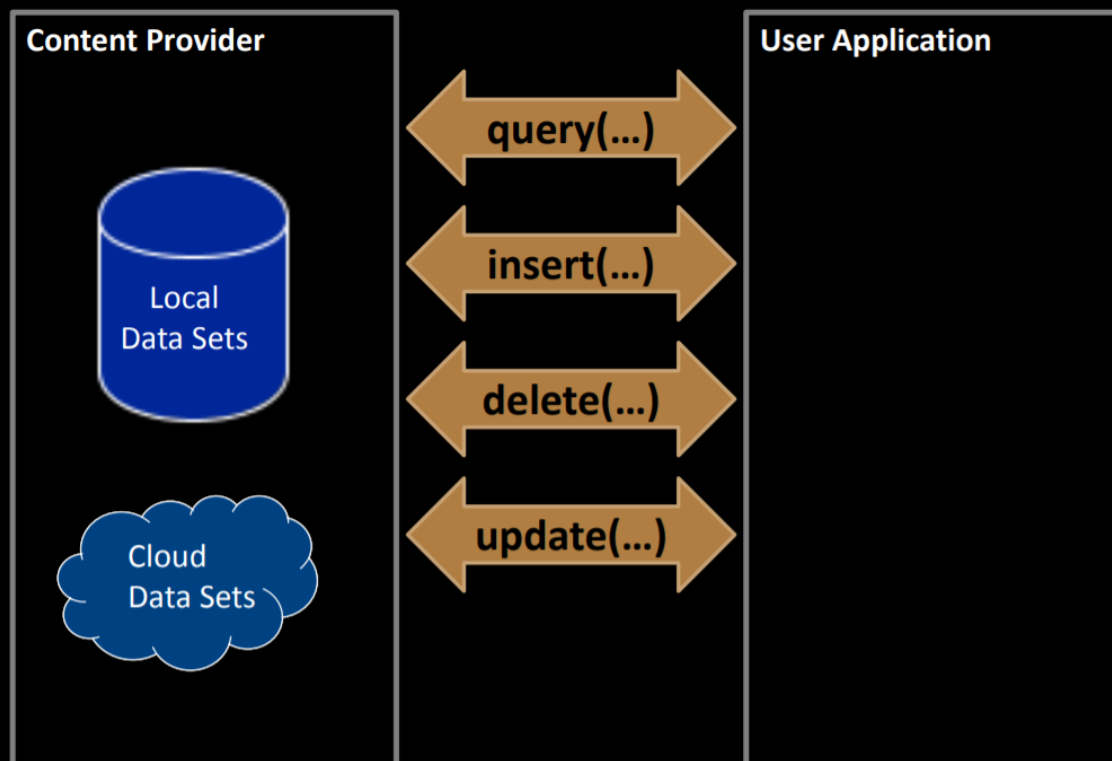
- Service Class – specjalny typ aktywności – nie posiada *Visual User Interface*. Usługa może być aktywna w tle
- Usługi zazwyczaj pracują w tle wykonując „busy-work” przez nieokreślony /nieskończony czas
- Aplikacje rozpoczynają własne usługi lub łączą się z usługami już aktywnymi (GPS, przełączanie między aplikacjami, aktywnościami, brak okienek)



- **BroadcastReceiver Class** – dedykowany *listener* oczekujący na wiadomość zwykle typu *system-wide* w celu wykonania pewnych czynności
- Przykładowo: niski poziom baterii, wi-fi dostępne, ostrzeżenie przed radarami
- Nie posiadają interfejsu
- Rejestrowanie wiadomości typowo po kluczu – jeśli wiadomość odpowiada kluczowi odbiornik jest aktywowany
- Zazwyczaj odpowiada poprzez wykonanie specjalnej czynności lub przekazanie użytkownikowi wiadomości/powiadomienia



- ContentProvider Class – zapewnia dostęp do danych wielu aplikacjom
- Dane globalne: lista kontaktów, zdjęcia, wiadomości, filmy, email
- Dane globalne są zazwyczaj przechowywane w bazie danych (SQLite)
- Klasa oferuje metody innym aplikacjom: retrieve, delete, update, insert
- Wrapper ukrywający właściwe dane. Dostępny interfejs.





Uniwersytet  
Wrocławski

# Cykl Życia

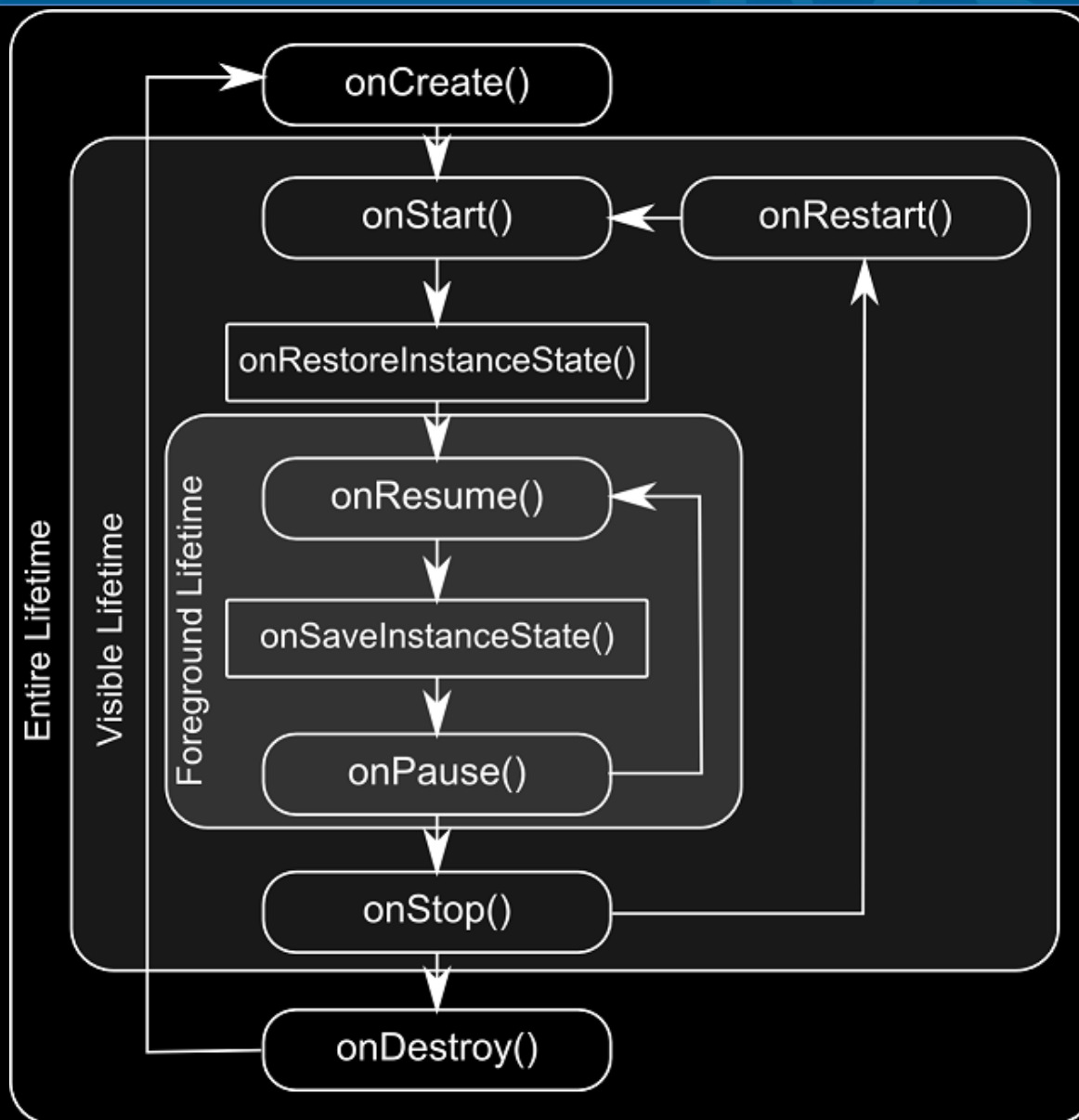


- Każda aplikacja Androida działa w swojej własnej instancji maszyny wirtualnej
- W każdym momencie kilka instancji maszyny wirtualnej może być aktywna (rzeczywista równoległość – nie *task switching*)
- Aplikacja Androida nie kontroluje całkowicie realizacji swojego cyklu życia
- OS może zakończyć każdy proces
  - Zasoby są krytycznie niskie
  - Duża liczba działających aplikacji
  - Aplikacja wymagająca bardzo dużych zasobów (energia, pamięć)

***Start***

**Life as an Android Application:  
Active / Inactive  
Visible / Invisible**

***End***



Zmiana konfiguracji wymaga przeładowania layoutu oraz innych zasobów gdy:

- Następuje zmiana orientacji urządzenia
- Zostaje zmieniony język

Przy zmianie konfiguracji Android:

1. Wyłącza aktywność – `onPause()`, `onStop()`, `onDestroy()`
2. Startuje nową instancję aktywności – `onCreate()`, `onStart()`, `onResume()`

## Bundle Class

- Można zapisać nawet cały obiekt
- Klasa musi implementować interfejs `Parcelable`
  - Konstruktor przyjmuje obiekt typu `Parcel`
  - `writeToParcel(Parcel dest, int flags)`
  - `Parcelable.Creator`

@Override

```
public void onSaveInstanceState(Bundle outState) {  
    super.onSaveInstanceState(outState);  
  
    // Add information for saving HelloToast counter  
    // to the to the outState bundle  
    outState.putString("count",  
        String.valueOf(mShowCount.getText()));  
}
```

Two ways to retrieve the saved Bundle

- in onCreate(Bundle mySavedState)  
Preferred, to ensure that your user interface, including any saved state, is back up and running as quickly as possible
- Implement callback (called after onStart())  
[onRestoreInstanceState\(Bundle mySavedState\)](#)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mShowCount = findViewById(R.id.show_count);

    if (savedInstanceState != null) {
        String count = savedInstanceState.getString("count");
        if (mShowCount != null)
            mShowCount.setText(count);
    }
}
```

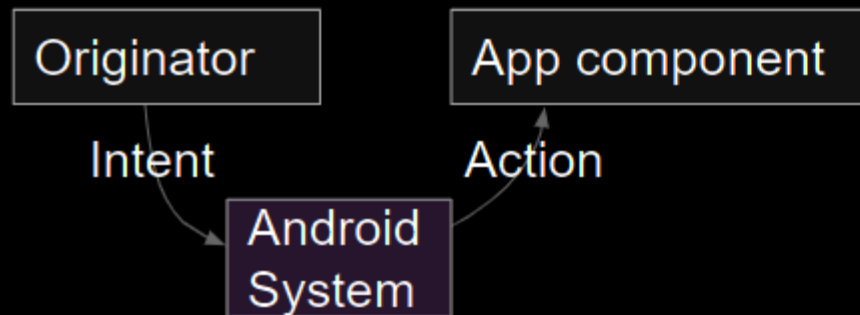


Uniwersytet  
Wrocławski

# Intent

An Intent is a description of an operation to be performed.

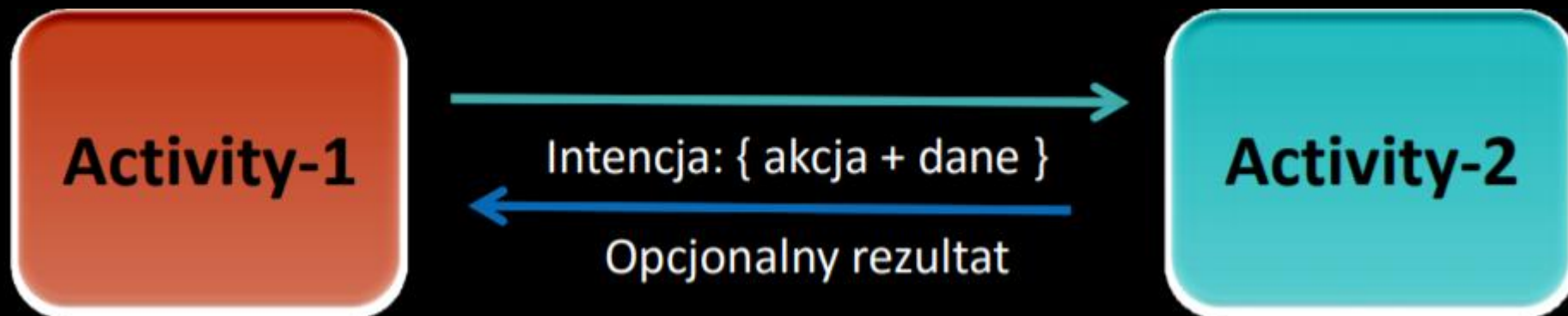
An Intent is an object used to request an action from another app component via the Android system.





- Wywoływanie intencji

<code>startActivity (intent)</code>	Uruchamia (pokazuje) nową aktywność
<code>sendBroadcast (intent)</code>	Wysyła intencję do komponentu typu BroadcastReceiver
<code>startService (intent)</code> lub <code>bindService (intent, ...)</code>	Używane są do komunikacji z usługą działającą w tle (osobnym wątku).



## Explicit Intent

- Starts a specific Activity
  - Request tea with milk delivered by Nikita
  - Main activity starts the ViewShoppingCart Activity

## Implicit Intent

- Asks system to find an Activity that can handle this request
  - Find an open store that sells green tea
  - Clicking Share opens a chooser with a list of apps





Uniwersytet  
Wrocławski

# Context

