Uniwersytet Wrocławski

Rafał Lewandków

PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

WYKŁAD 1

- PODSTAWOWE INFORMACJE
- PLAN WYKŁADU
- ZASADY ZALICZENIA

Uniwersytet Wrocławski

PODSTAWOWE INFORMACJE

Rafał Lewandków
pokój 075
rafal.lewandkow@uwr.edu.pl
rafal.lewandkow2@uwr.edu.pl

Forma zajęć i liczba godzin:
Wykład 15 godz. /Laboratorium 45 godz.

Literatura obowiązkowa i zalecana:
”Android Programming, The Big Nerd Ranch Guide” 3rd ed. B.Phillips, Ch. Stewart, K. Marsicano.

Nakład pracy studenta:
Praca własna studenta: 65 godz.
Łączna liczba godzin 125
Liczba punktów ECTS: 5

Uniwersytet Wrocławski

PODSTAWOWE INFORMACJE

- **Android Studio**
- **Cykl życia**
- **Android Jetpack**
- **Jetpack Navigation**
- **Jetpack Compose**
- **Bazy danych – ROOM, SQLite**
- **Aktywności, Fragmenty, Composable**
- **Wielowątkowość**
- **Architektura aplikacji – MVVM, MVI**
- **Dependency Injection – Dagger-Hilt**

Uniwersytet Wrocławski

ZASADY ZALICZENIA

Wykład:

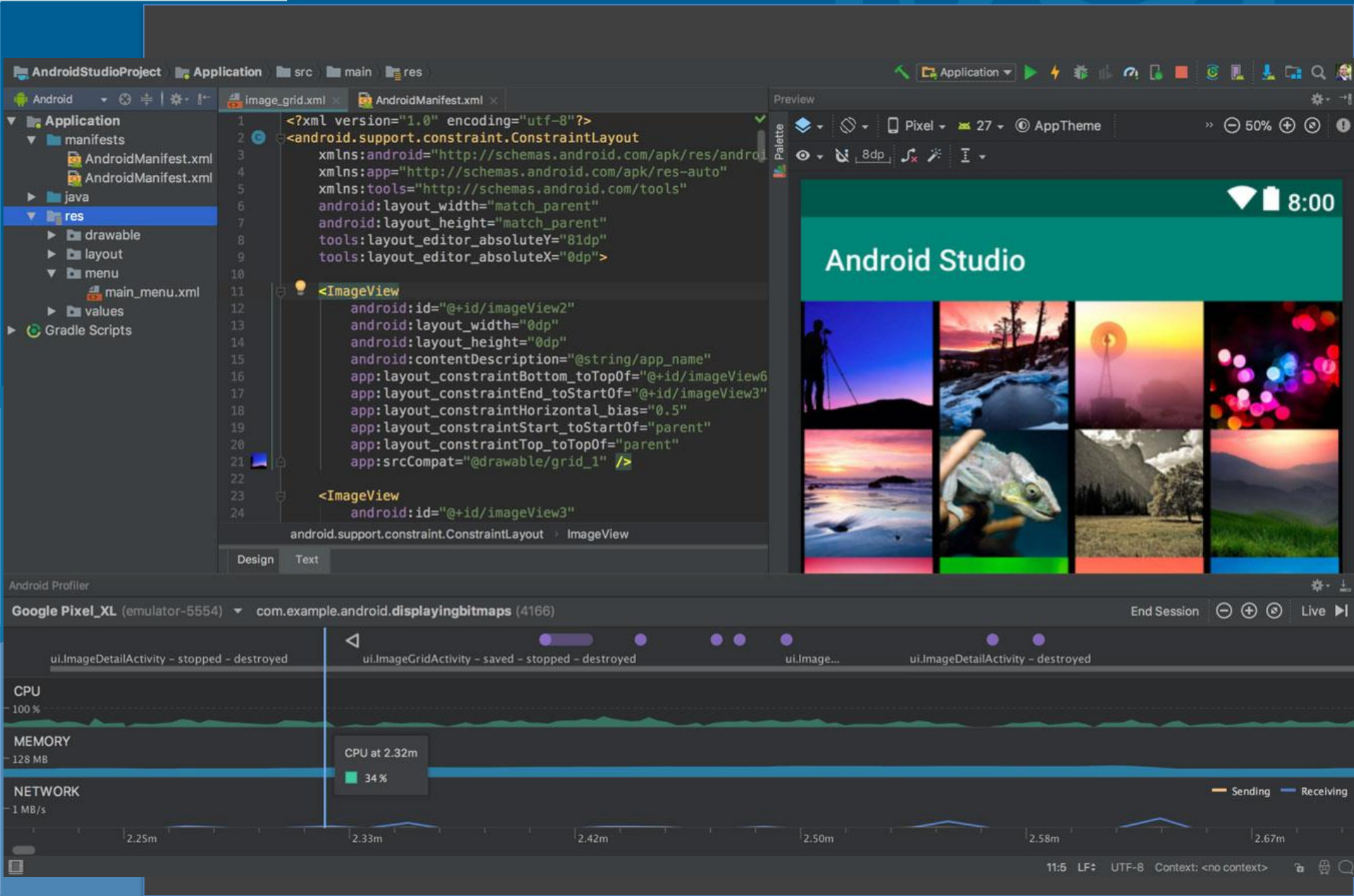
- Wykonanie projektu
- Małe projekty
- Do projektu powinna zostać dołączona dokumentacja:
  - Cel i opis projektu
  - Lista funkcjonalności
- Każdy projekt musi zostać zatwierdzony przez prowadzącego przed wykonaniem

Uniwersytet Wrocławski

Architektura Androida

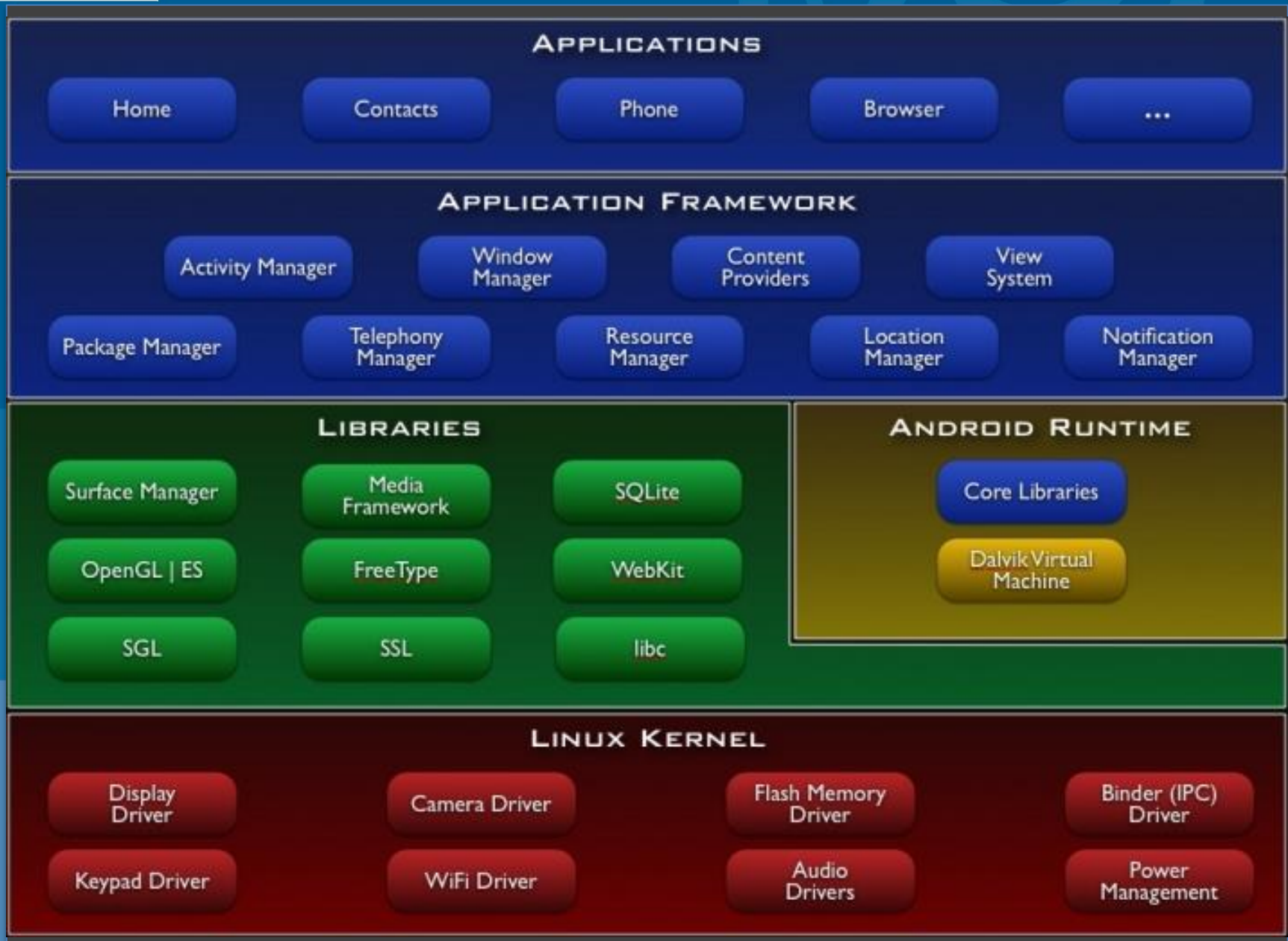
Uniwersytet Wrocławski

Architektura platformy Android



Uniwersytet Wrocławski

Architektura platformy Android



Uniwersytet Wrocławski

Architektura platformy Android

- Sprzętowa warstwa abstrakcji urządzenia (HAL)
- Bazuje na jądrze systemu Linux 2.6 – Android nie jest Linuksem
- Głównie sterowniki do urządzeń obecnych na urządzeniu
- Brak natywnego wsparcia dla „okienek”
- Zarządzanie pamięcią, procesami, sicią, zasilaniem

Uniwersytet Wrocławski

Architektura platformy Android

- Natywne Biblioteki (C/C++)
- Zapewniają interfejs pomiędzy sterownikiem a aplikacją

- **Surface Manager** – rysowanie okien na ekranie działających na różnych procesach
- **OpenGL/ES, SGL** – biblioteki 3D i 2D
- **Media Framework** – kodeki Audio/Video
- **Free Type** – renderer czcionek
- **SQLite** – implementacja baz danych
- **WebKit** – silnik przeglądarki

Architektura platformy Android

- **Core Java Libraries** – zbiór bibliotek języka Java
- **Core Android Libraries** – Interfejsy w języku Java do komponentów z warstwy Libraries

- **Dalvik Virtual Machine – Java Virtual Machine**
- **Android Runtime**

Architektura platformy Android

- Zbiór wysokopoziomowych bibliotek
- Podstawowe komponenty aplikacji

- **Activity Manager** – Kontroler cyklu życia uruchomionych z aktywności
- **Content Providers** – Zapewniają współdzielenie danych pomiędzy aplikacjami
- **Resource Manager** – Udostępnia zasoby
- **Notification Manager** – Umożliwia wyświetlenie powiadomień w telefonie
- **Package Manager** – Kontroluje jakie aplikacja są zainstalowane na telefonie
- **Location Manager** – GPS i inne

Uniwersytet Wrocławski

Podstawowe Elementy Aplikacji

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

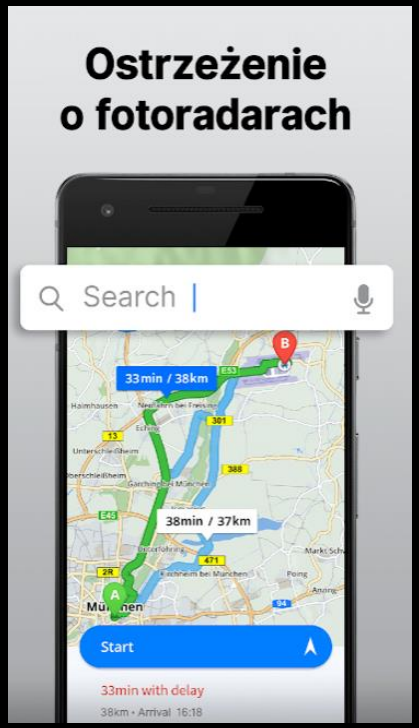
Podstawowe elementy aplikacji

- 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

BACHA Soft

Podstawowe elementy aplikacji

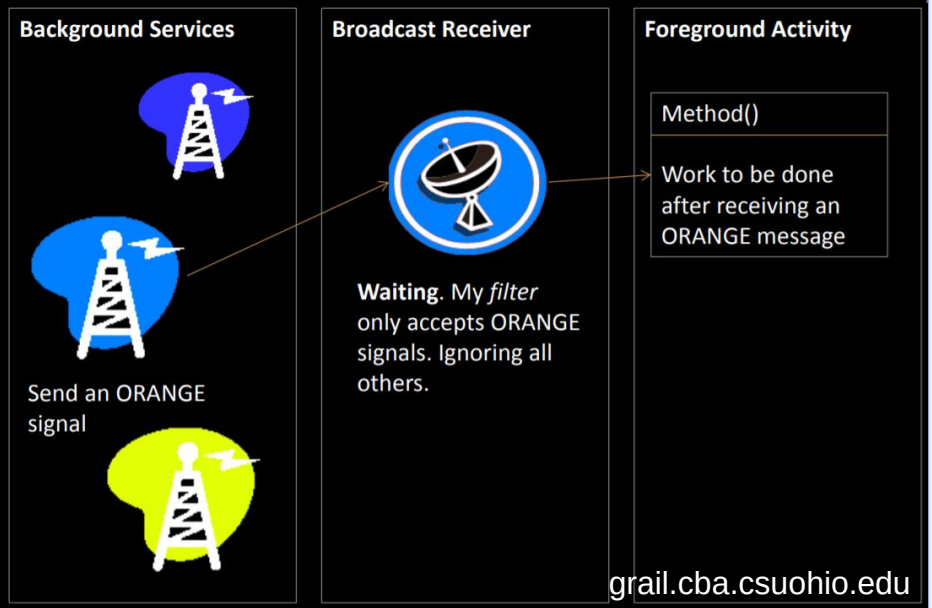
- Service Class – specjalny typ aktywności – nie posiada *Visual User Interface*. Usługa może być aktywna w tle
- Usługi zazwyczaj pracują w tle wykonując „busy-work” przez nieokreślony /nieskończony czas
- Aplikacje rozpoczynają własne usługi lub łączą się z usługami już aktywnymi (GPS, przełączanie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



Maps, GPS Navigation

Podstawowe elementy aplikacji

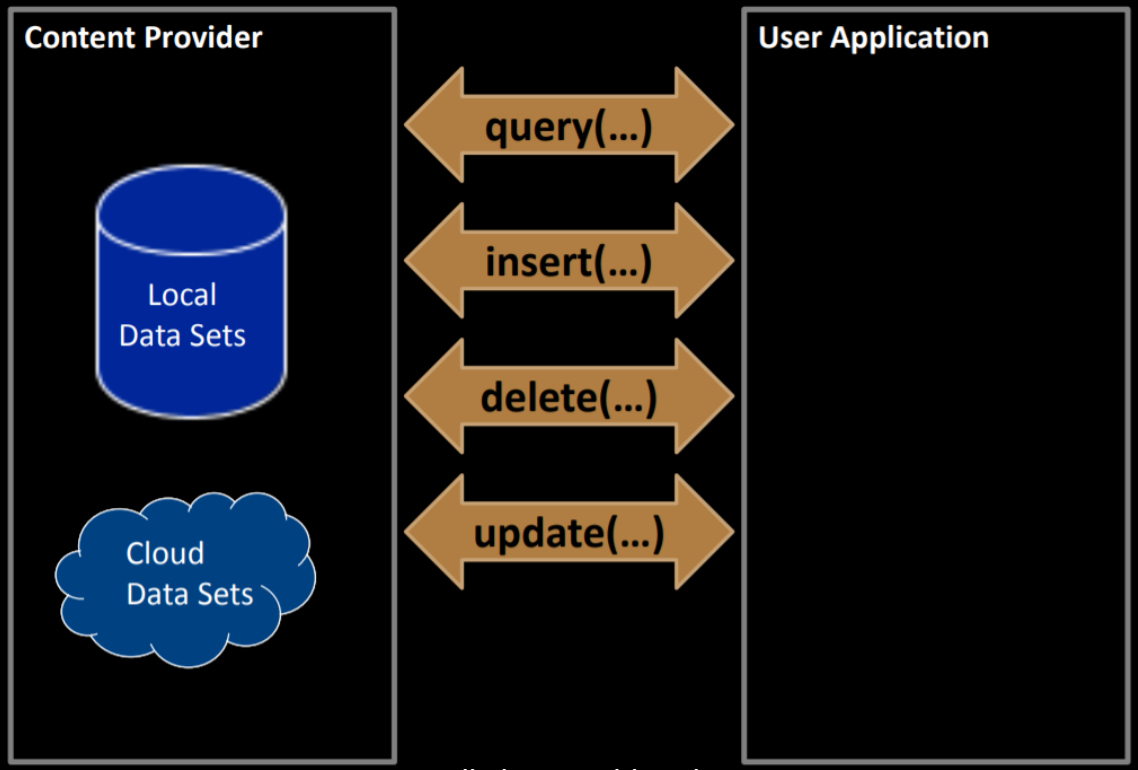
- BroadcasterReceiver Class – dedykowany *listener* oczekujący na wiadomość zwykle typu *system-wide* w celu wykonania pewnych czynności
- Przykładowo: niski poziom baterii, wi-fi dostępne, ostrzeżenie przed radarami
- Nie posiadają interfejsu
- Rejestrowanie wiadomości typowo po kluczu – jeśli wiadomość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



Uniwersytet Wrocławski

Podstawowe elementy aplikacji

- 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

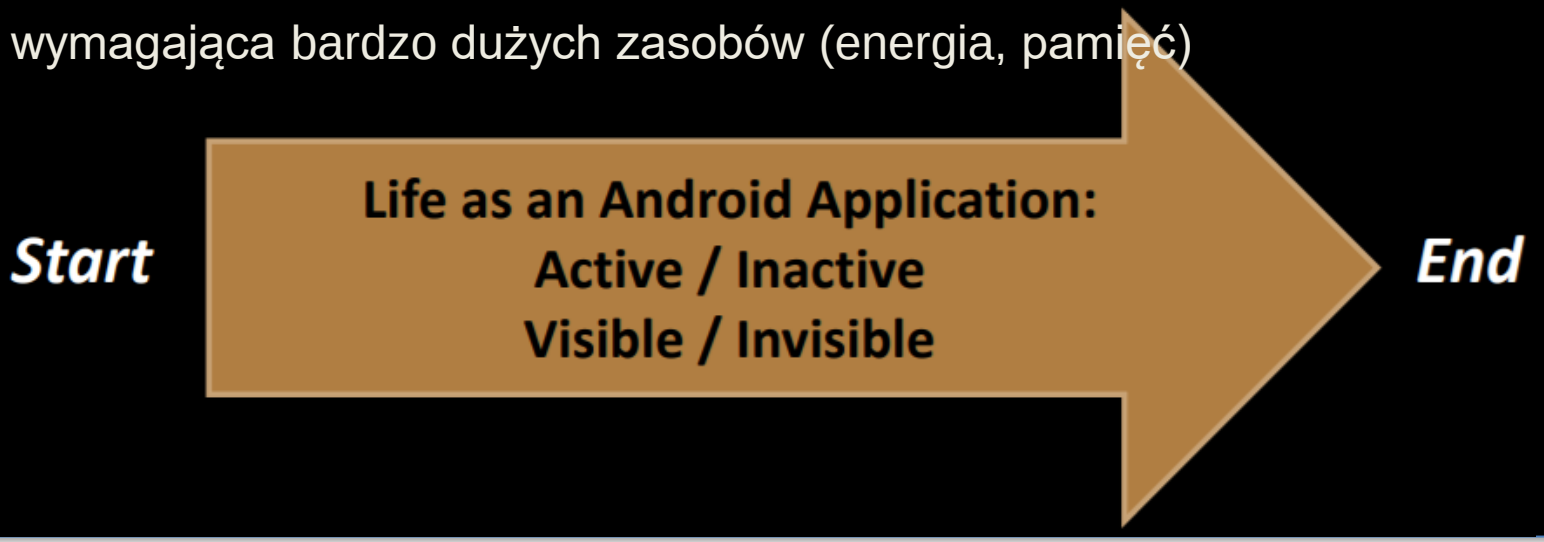
Cykl Życia

Uniwersytet Wrocławski

Cykl Życia

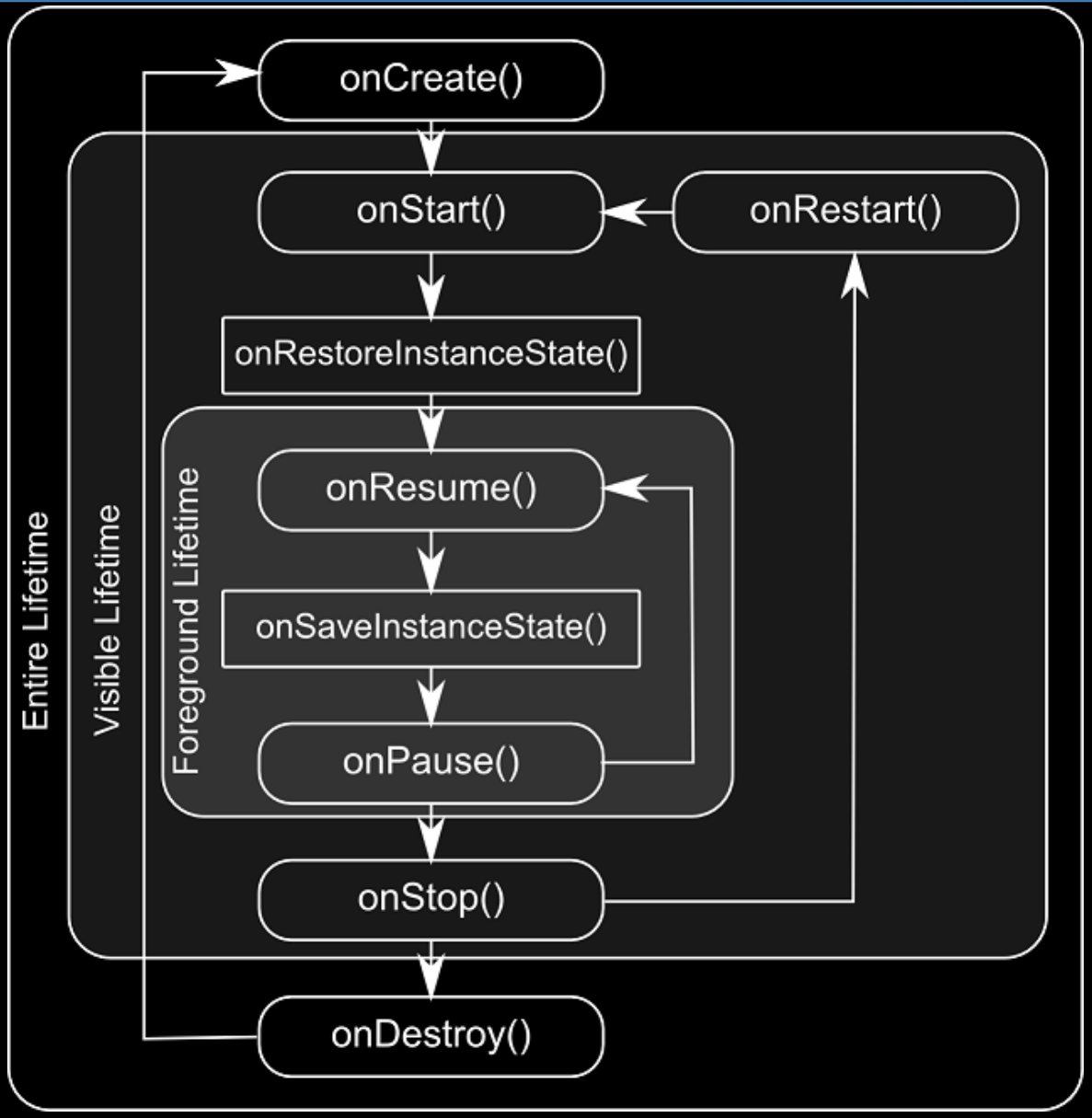
- Każda aplikacja Androida działa w swojej własnej instancji maszyny wirtualnej
- W każdym momencie kilka instancji maszyny wirtualnej może być aktywna (rzeczywista równoległość – nie *task switching*)
- Aplikacja Androida nie kontroluje całkowicie realizacji swojego cyklu życia

- OS może zakończyć każdy proces
  - Zasoby są krytycznie niskie
  - Duża liczba działających aplikacji
  - Aplikacja wymagająca bardzo dużych zasobów (energia, pamięć)



Uniwersytet Wrocławski

Cykl Życia



Uniwersytet Wrocławski

Cykl Życia – Trwałość Danych

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()

https://developer.android.com/reference/android/app/Activity.html

Uniwersytet Wrocławski

Cykl Życia – Trwałość Danych

Bundle Class

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

https://developer.android.com/reference/android/app/Activity.html

Uniwersytet Wrocławski

Cykl Życia – Trwałość Danych

```
@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)`

https://developer.android.com

Uniwersytet Wrocławski

Cykl Życia – Trwałość Danych

```
@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);
    }
```

https://developer.android.com

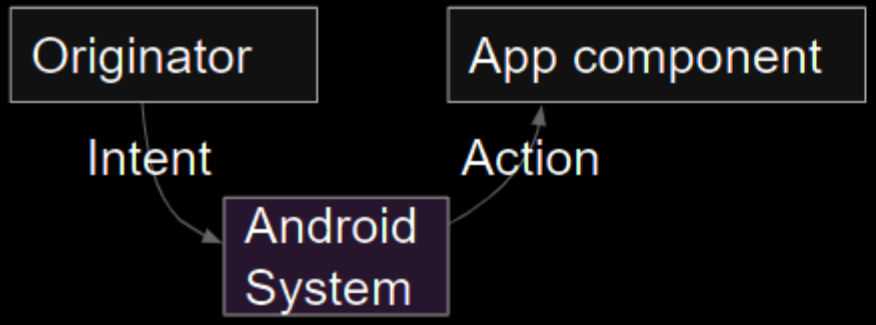
Uniwersytet Wrocławski

Intent

Mechanizm Intencji

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.



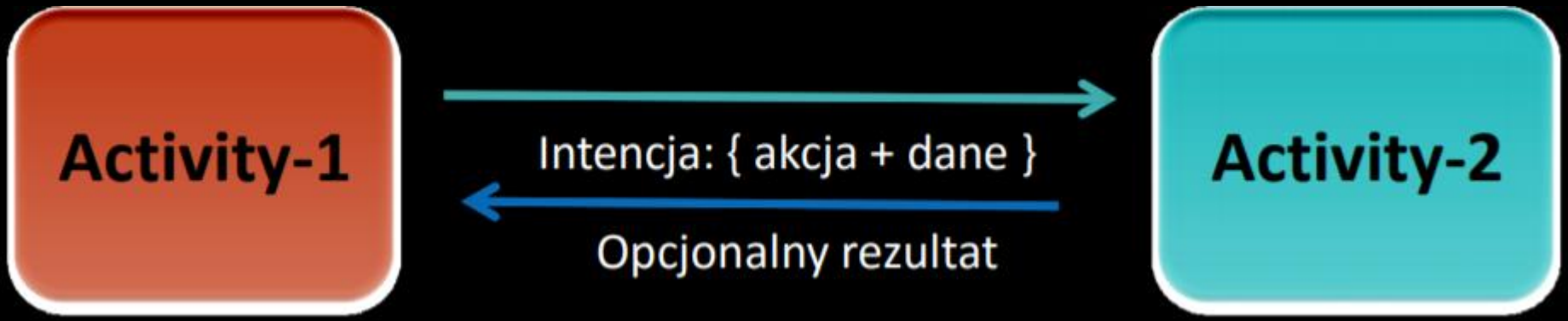
Uniwersytet Wrocławski

Mechanizm Intencji

- Wywoływanie intencji

`startActivity (intent)`	Uruchamia (pokazuje) nową aktywność
`sendBroadcast (intent)`	Wysyła intencję do komponentu typu BroadcastReceiver

`startService (intent)`
`lub`
`bindService (intent, …)`
	Używane są do komunikacji z usługą działającą w tle (osobnym wątku).



Mechanizm Intencji

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

Mechanizm Intencji

- Uruchamianie aktywności jest jednym z głównych zadań intencji
- Intencje uruchomieniowe zostały podzielone na dwa typy
  - Jawne – z jawnie określonym obiektem który chcemy otworzyć
    - Intencja uruchamiająca aktywność `mojaAktywność`

```
Intent intent = new Intent (context,
                    mojaAktywność.class);
startActivity(intent);
```

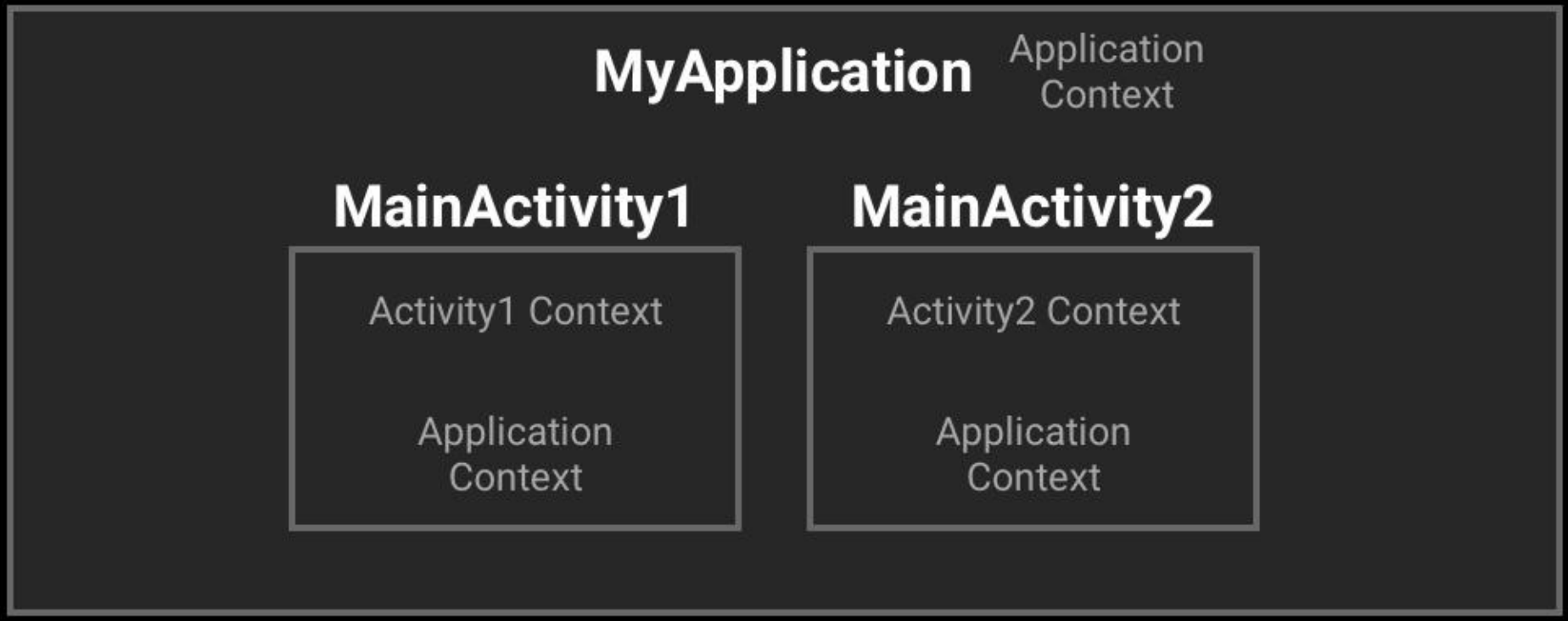
  - Domniemane – intencje zawierające informacje o tym co chcemy zrobić
    - Nie podajemy konkretnych klas które mają zadanie realizować
    - System decyduje która aktywność zostaje uruchomiona
    - Decyzje podejmuje przy pomocy Filtrów Intencji
    - Intencja pokazująca dane z adresu URL

```
Intent intent = new Intent (Intent.ACTION_VIEW,
                    Uri.parse(„www.uwr.edu.pl”));
```

Uniwersytet Wrocławski

Context

Context



https://developer.android.com/reference/android/app/Activity.html