



# PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

## WYKŁAD 8 Architektura Aplikacji 2

- Packaging
- MVx

- **Przez typ**
  - Ze względu na język
    - Klasy
    - Interfejsy
    - Obiekty
  - Ze względu na framework
    - Aktywności
    - Fragmenty
    - Service
    - Notification
  - Ze względu na funkcję
    - Kontrolery
    - Widoki
    - Usecasy
- **Przez warstwę**
  - Ui
  - Db
  - Di
  - Net
- **Przez ekran**
  - Autoryzacja
  - Logowanie
  - Koszyk
  - Etc

# Package by Feature

Ta strategia zakłada, że pakiety najwyższego poziomu w aplikacji powinny odpowiadać podstawowym pojęciom z domeny biznesowej aplikacji.

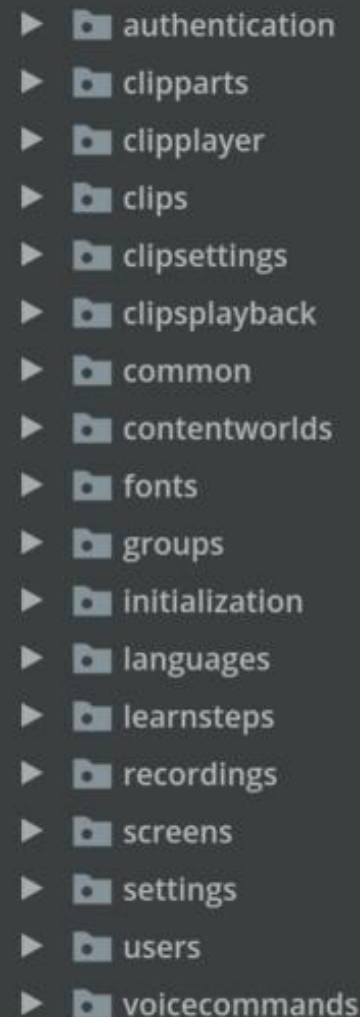
Jeśli pracujesz nad platformą e-commerce, mogą to być niektóre z Twoich funkcji:

- uwierzytelnianie
- użytkownicy
- produkty
- oferty
- finalizacja zakupu

Dla odtwarzacza multimedialnego mogą to być takie elementy:

- albumy
- utwory
- artysta
- oledcje
- W przypadku aplikacji e-commerce domeną biznesową będzie handel elektroniczny, a kluczowe pojęcia to produkty, koszyki zakupowe, zamówienia, użytkownicy, płatności itp.

**Żaden** z wymienionych pakietów nie zawiera logiki związanej z ekranami. Na przykład ekran logowania nie jest częścią funkcji uwierzytelniania. Zwykle umieszczane są w dedykowanym pakiecie na najwyższym poziomie o nazwie „ekrany”, „ui”, który zawiera podpakiety zorganizowane zgodnie z podejściem Package by Screen.



# Package by Feature

Ta strategia zakłada, że pakiet podstawowym pojęciom z domeny

Jeśli pracujesz nad platformą e-commerce, funkcji:

- uwierzytelnianie
- użytkownicy
- produkty
- oferty
- finalizacja zakupu

Dla odtwarzacza multimedialnego:

- albumy
- utwory
- artysta
- oledkcie

- W przypadku aplikacji e-commerce, elektroniczny, a kluczowe zamówienia, użytkownicy, p

**Żaden** z wymienionych pakietów. Na przykład ekran logowania. Zwykle umieszczane są w poziomie o nazwie „ekrany”, „ui” zgodnie z podejściem Package

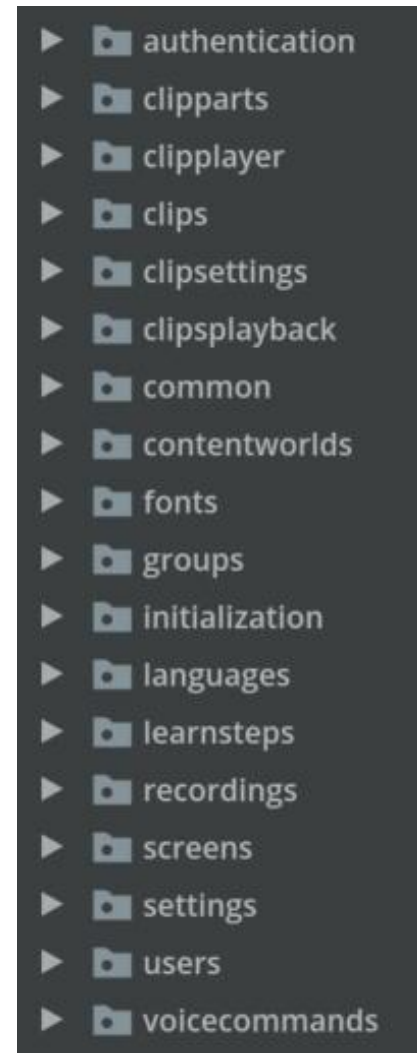
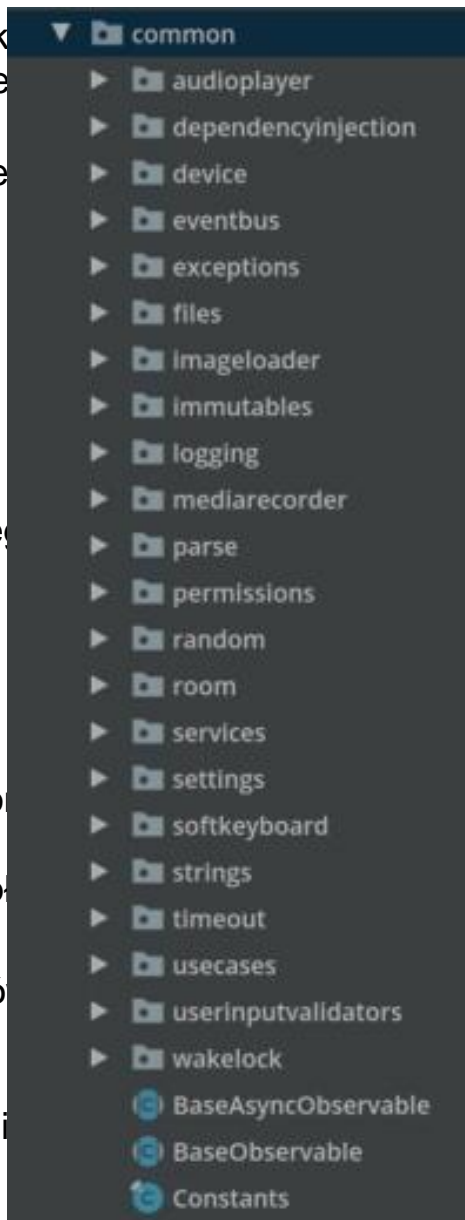
w aplikacji powinny odpowiadać

ektóre z Twoich

ty:

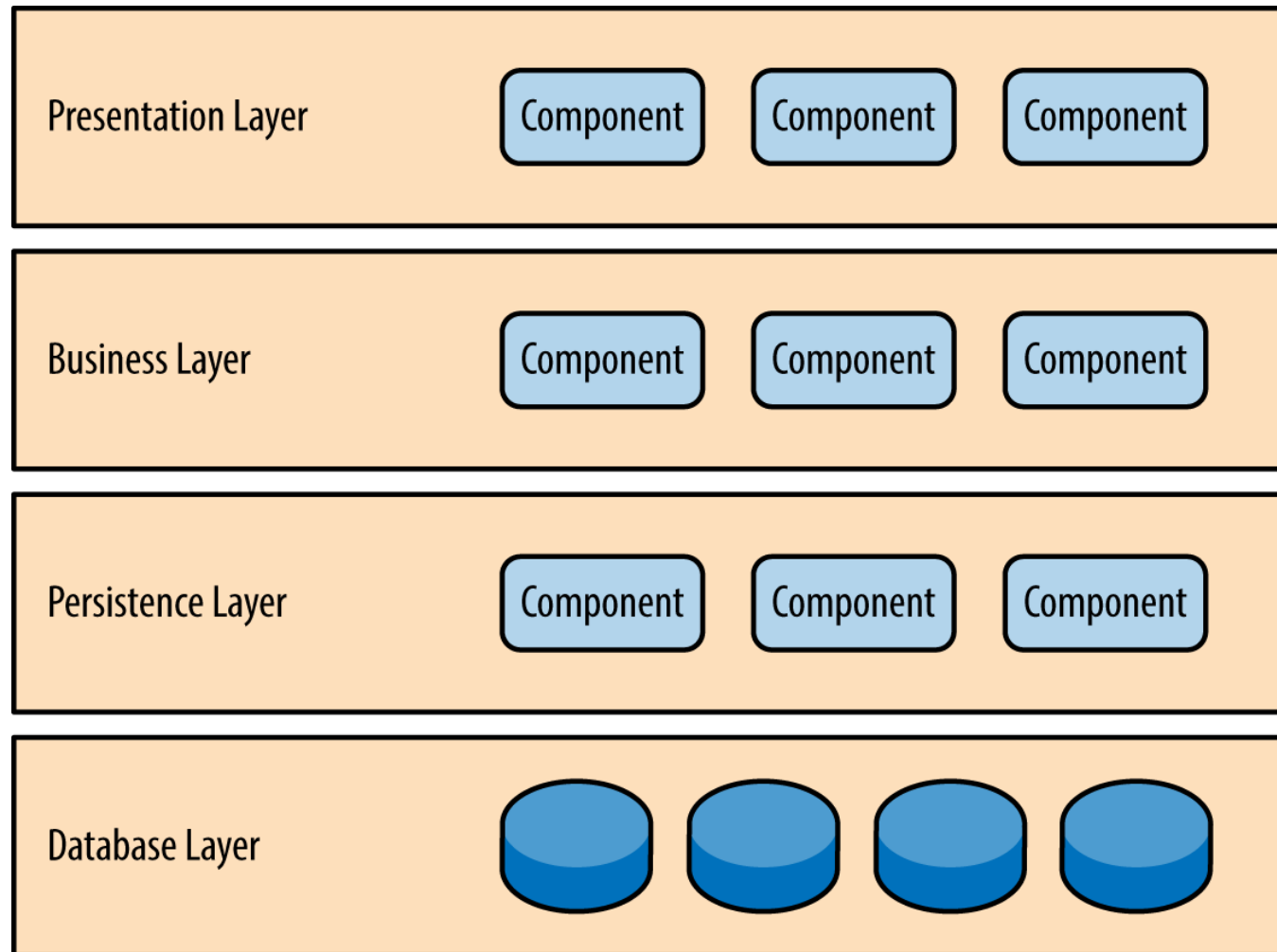
będzie handel  
yki zakupowe,

nej z ekranami.  
wierzytelniania.  
a najwyższym  
zorganizowane

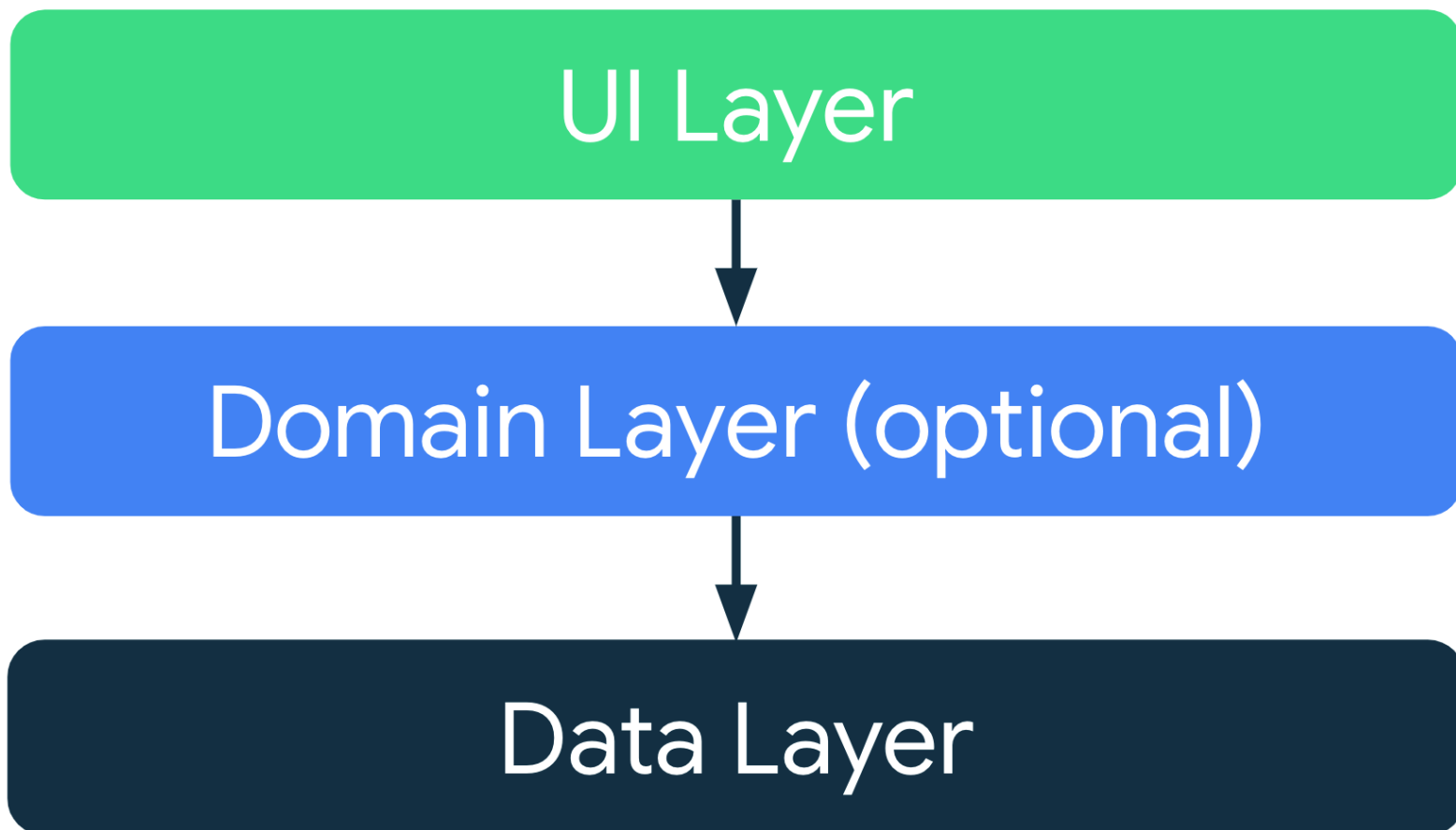


# Presentation Layer Architectural Pattern

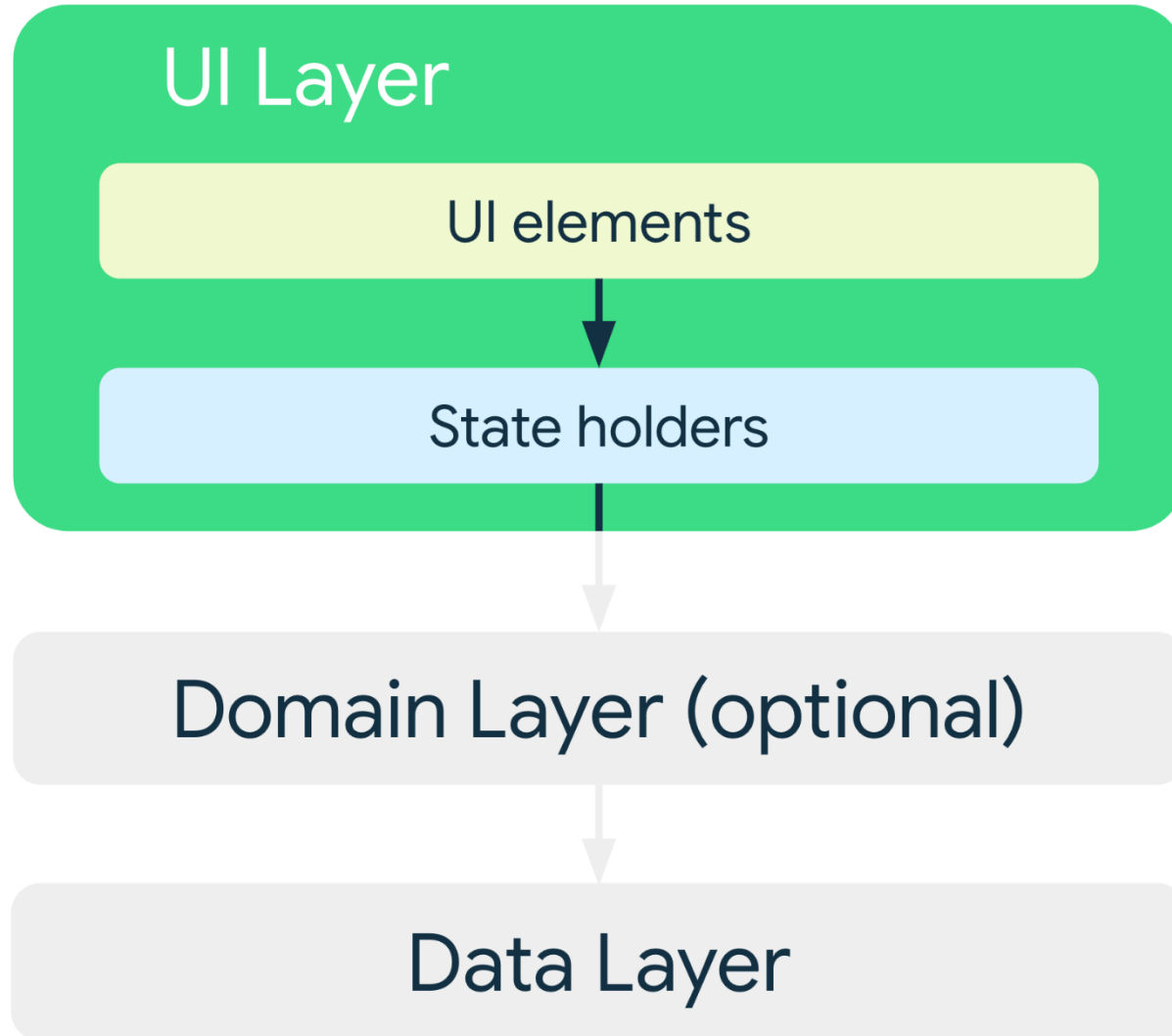
# Presentation Layer Architectural Pattern



## Presentation Layer Architectural Pattern

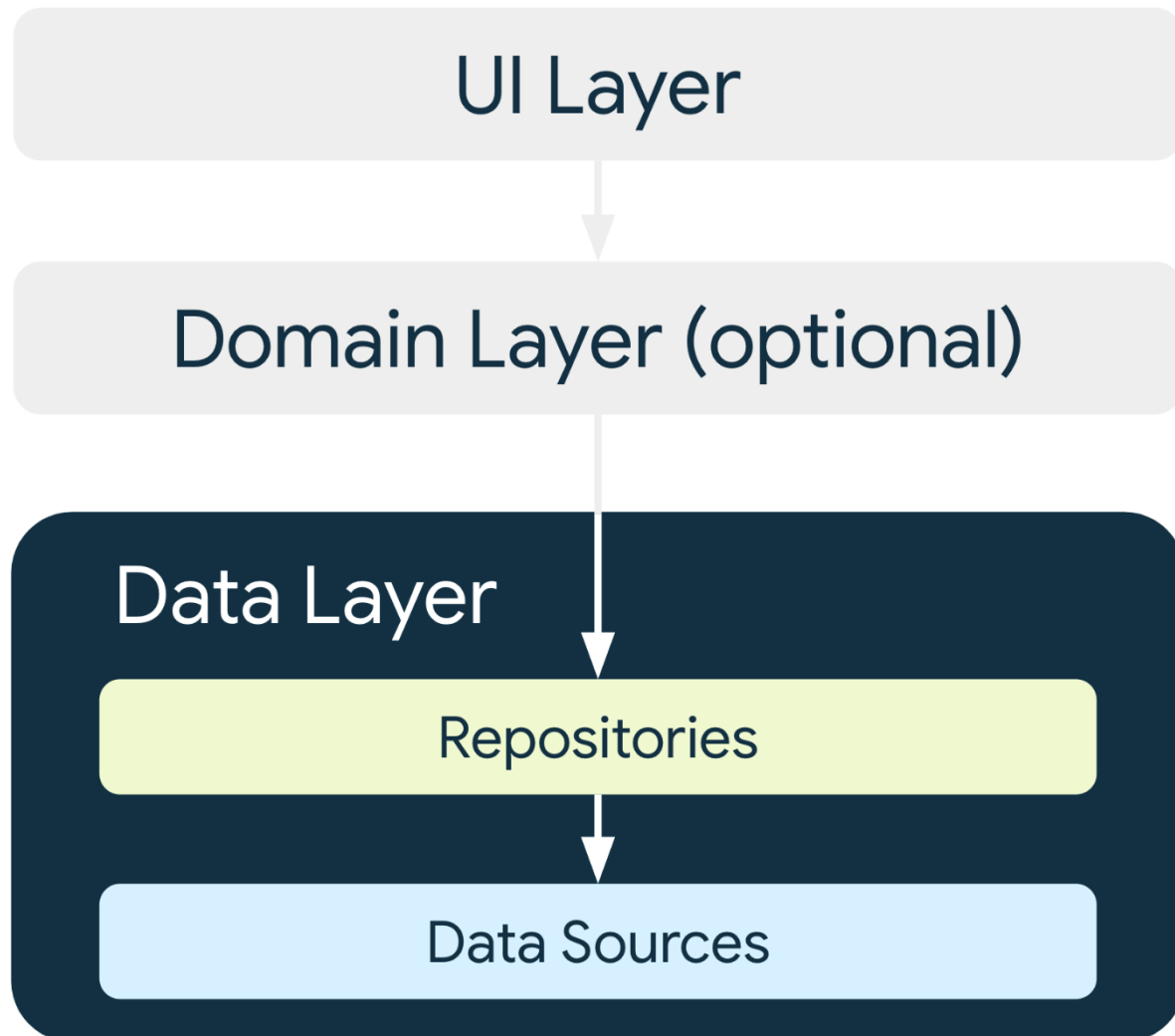


## Presentation Layer Architectural Pattern

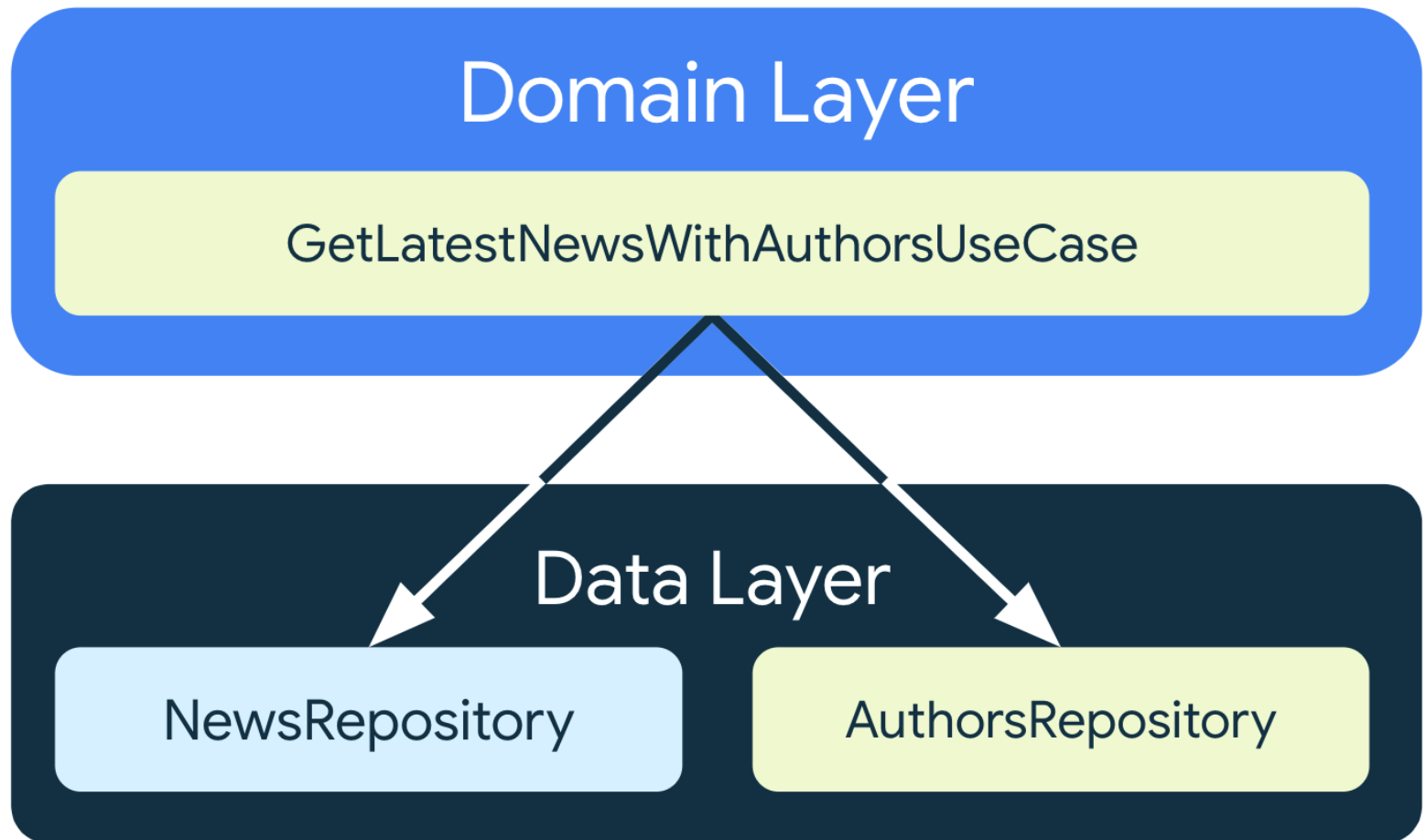




## Presentation Layer Architectural Pattern



## Presentation Layer Architectural Pattern



# Presentation Layer Architectural Pattern

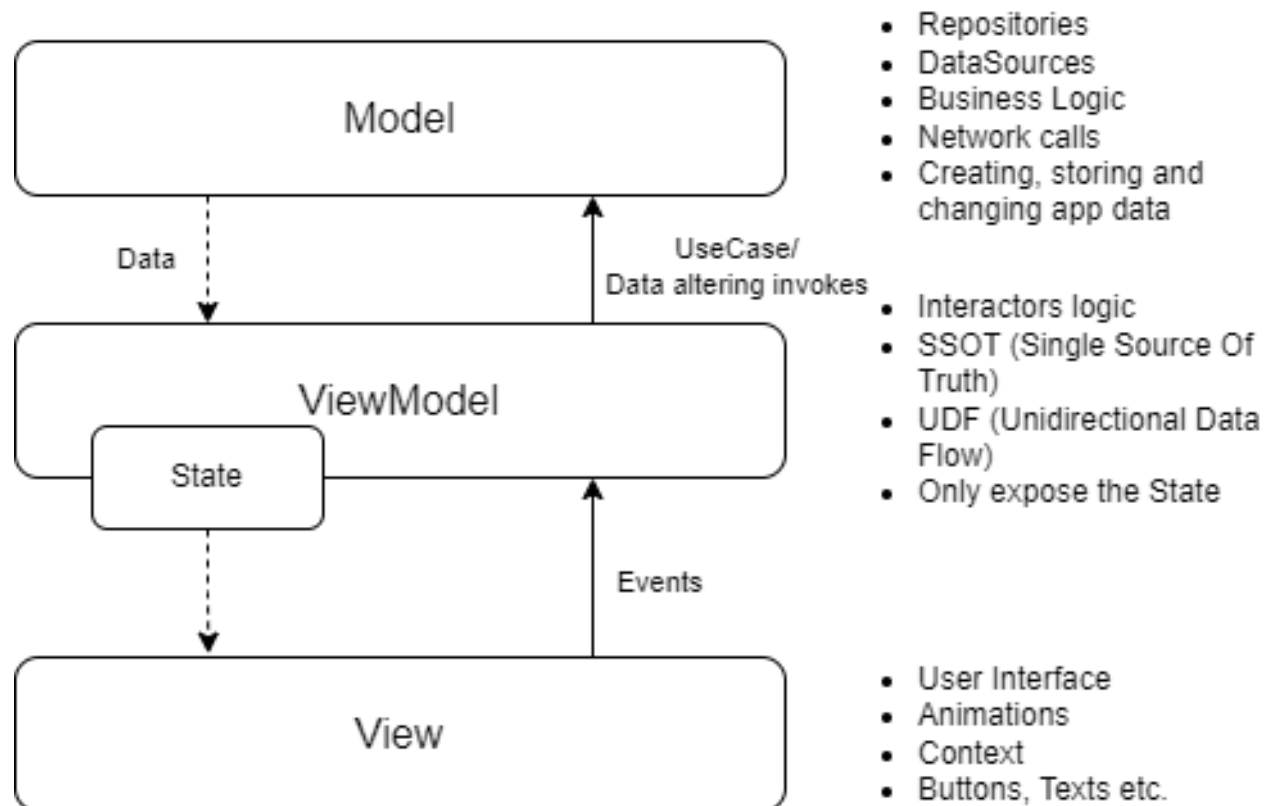
- **Prezentacja danych:**
  - Wyświetla dane, które są dostarczane przez warstwę domeny.
  - Odpowiada za formatowanie i renderowanie danych w odpowiedniej formie, aby użytkownik mógł je zrozumieć i nimi manipulować.
- **Obsługa interakcji użytkownika:**
  - Reaguje na akcje użytkownika, takie jak kliknięcia przycisków, wpisywanie tekstu, przewijanie, itp.
  - Przekazuje informacje o akcjach użytkownika do warstwy domeny lub bezpośrednio do modelu danych w celu zaktualizowania stanu aplikacji.
- **Zarządzanie cyklem życia komponentów UI:**
  - Zajmuje się zarządzaniem stanem i cyklem życia komponentów UI, takich jak Activity, Fragment, ViewModel (w architekturze MVVM) oraz innych elementów UI.
  - Utrzymuje spójność UI podczas zmian, takich jak rotacja ekranu, zamykanie i ponowne otwieranie aplikacji.

# Presentation Layer Architectural Pattern

**Model** – Stan, logika biznesowa, struktury danych

**View** – UI

**X** – Logika biznesowa, kontrola przepływu danych, stan



# Presentation Layer Architectural Pattern

Wysokopoziomowa struktura wielokrotnego użytku przeznaczona do organizacji systemu.

