

# Recording in Progress

This class is being recorded

Please turn off your video and/or video if you do not wish to be recorded

# CMSC436: Programming Handheld Systems

# Modern App Architecture

# Today's Topics

Architecture Principles

Notional App Architecture

# Android App Behavior

Apps are made from Android components

Activity, Fragments, Services, etc.

Multiple entry points for app launch, so apps can start in different components

Android can kill components on reconfiguration / low memory

# Some Architectural Considerations

Framework components represent a contract between app and Android framework

- Limit app dependency on framework components

- Don't store app data or state in your app components

- Don't design app components so they depend on each other

# Architectural Principles

Separation of concerns

Drive UI from data models

Single source of truth

Unidirectional Data Flow

# Separation of Concerns

App implemented in separate classes

Each class has a single responsibility



# Drive UI from Data Models

UI and data are separate

Data models encapsulate an app's underlying data

UI presents the data

# Single Source of Truth

Each data type has a single owner

Only the owner can modify/mutate data instances

Owner exposes immutable instances to non-owners

# Unidirectional Data Flow

App operation includes states and events

Each flows in a single direction , e.g.,

- State flows from data towards UI

- Events flow from UI towards data

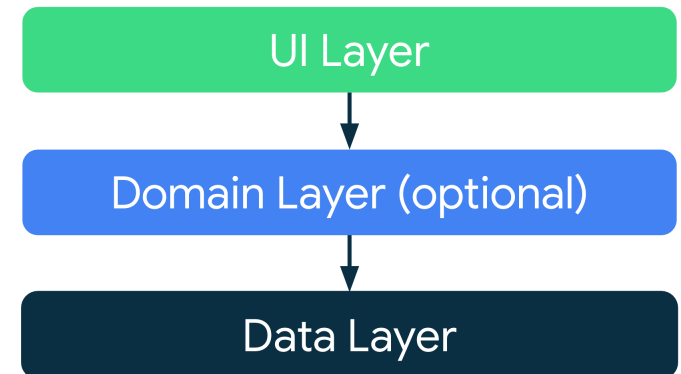
# Architecture Layers

Apps should have at least two layers:

A *UI layer* that displays application data on the screen

A *data layer* that contains the app's business logic and exposes its application data

An optional *domain layer* to simplify and reuse the interactions between the UI and data layers



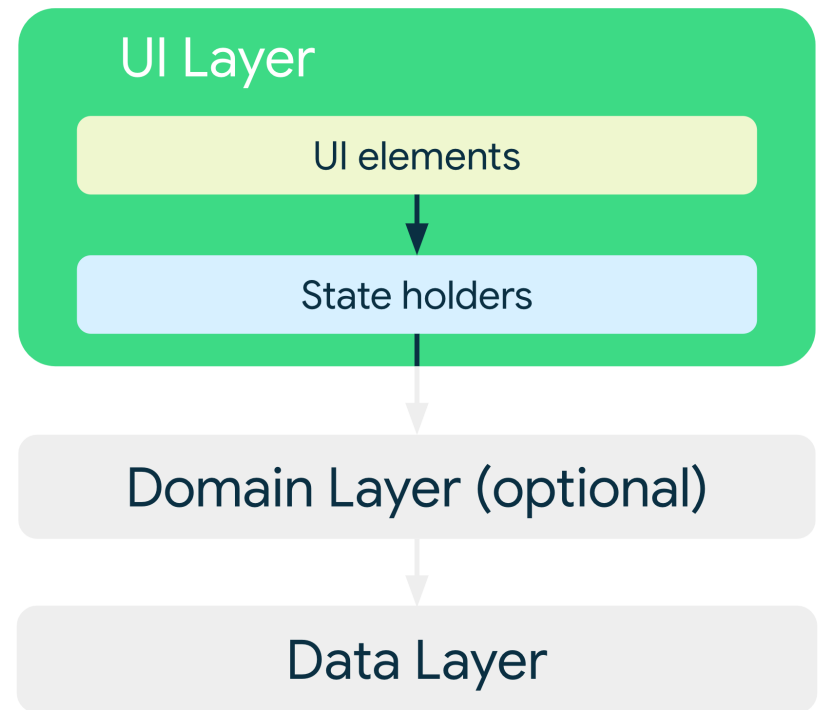
# UI Layer

## UI elements

Displays on screen

## State holders

Hold data, expose it to UI



# Data Layer

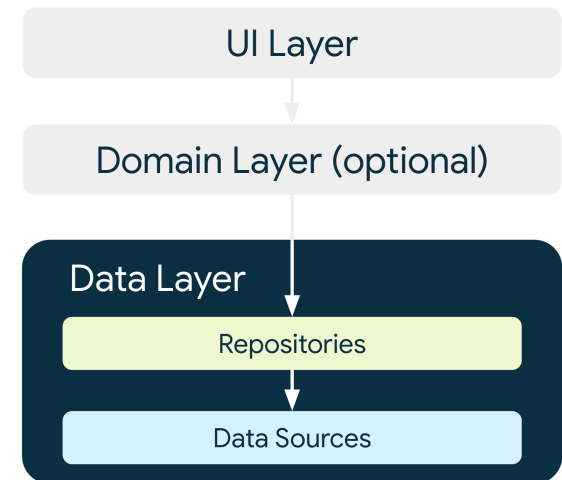
Implements app's business logic

Composed of repositories and data sources

Each data source manages one source of data (e.g., sensor, network server, local database)

Repositories contain zero or more data sources

One repository for each type of data



# Next Time

## User Interface Classes

# Example Applications

FragmentDynamicLiveDataLayout