

Implementing the Secondary Layers: Data and Presentation



Kaushal Dhruw

APP DEVELOPER / AUTHOR

@drulabs

linkedin.com/in/kaushal-dhruw/



Overview



ViewModel & LiveData (AAC)

The presentation layer

Testing presentation layer

The data layer & repository pattern

Testing data layer



Android Architecture Components: A Brief Introduction



Android Architecture Components

LiveData

Room

Paging

Work Manager

ViewModel

...more



Android Architecture Components

LiveData

Room

Paging

Work Manager

ViewModel

...more



ViewModel

Survives configuration changes

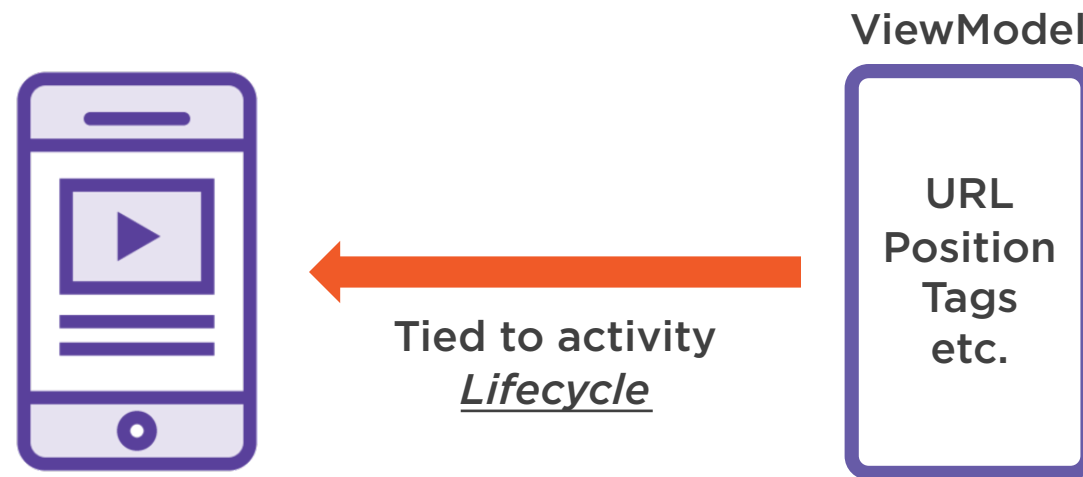
Provides data to view

Manages data in lifecycle conscious way

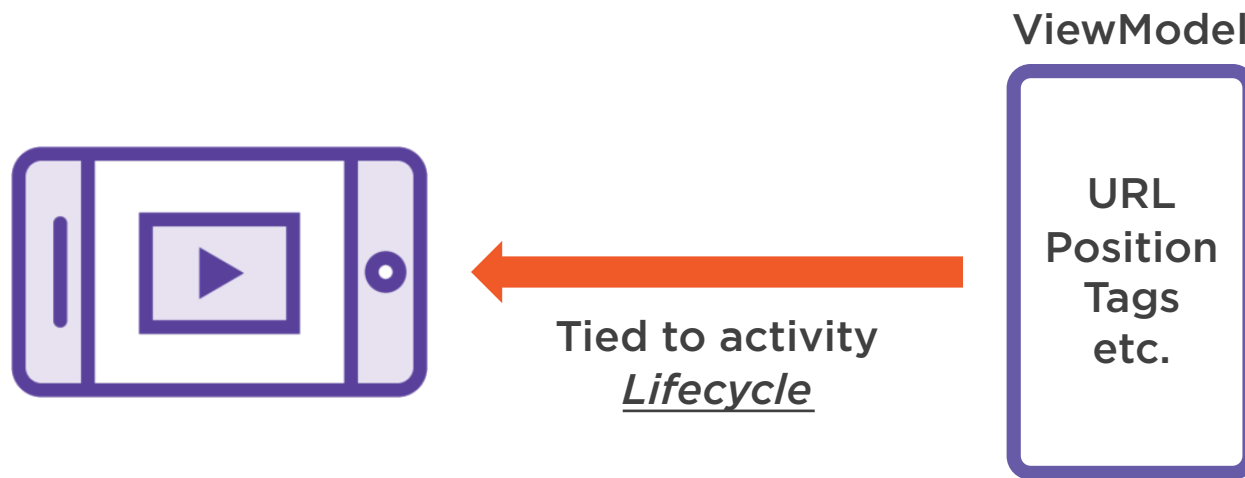
Separates UI from business layer



ViewModel Architecture Component



ViewModel Architecture Component



The saved data persists orientation change



LiveData<T>

Observable data holder

Lifecycle aware

No memory leaks

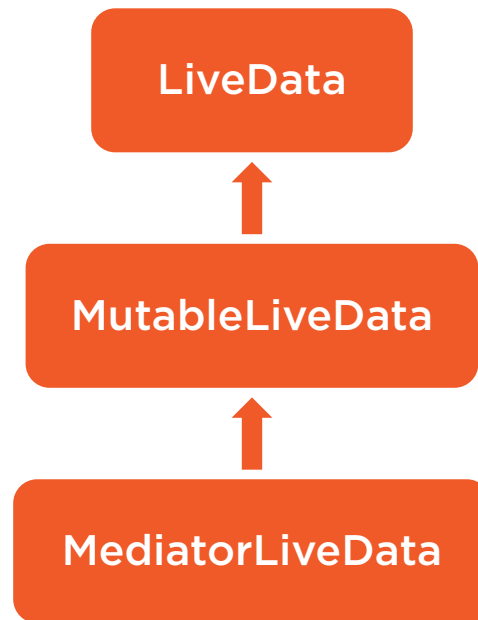
Synced UI state



RxJava Observable<T>
vs.
LiveData<T>



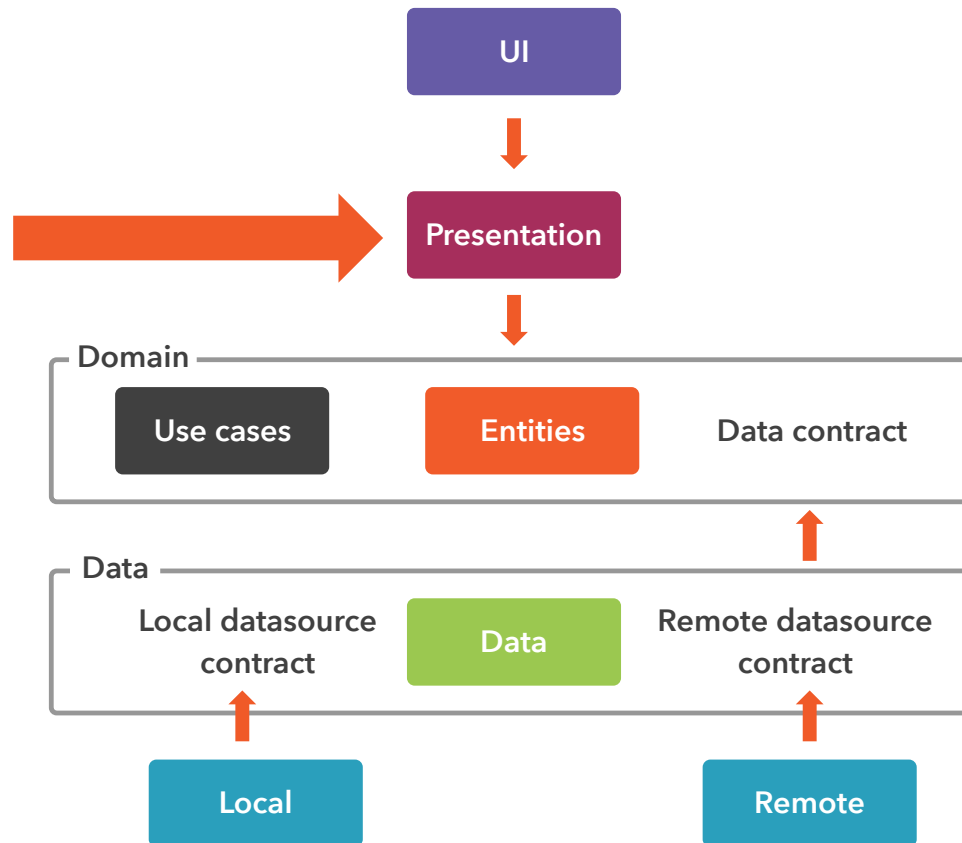
LiveData Hierarchy



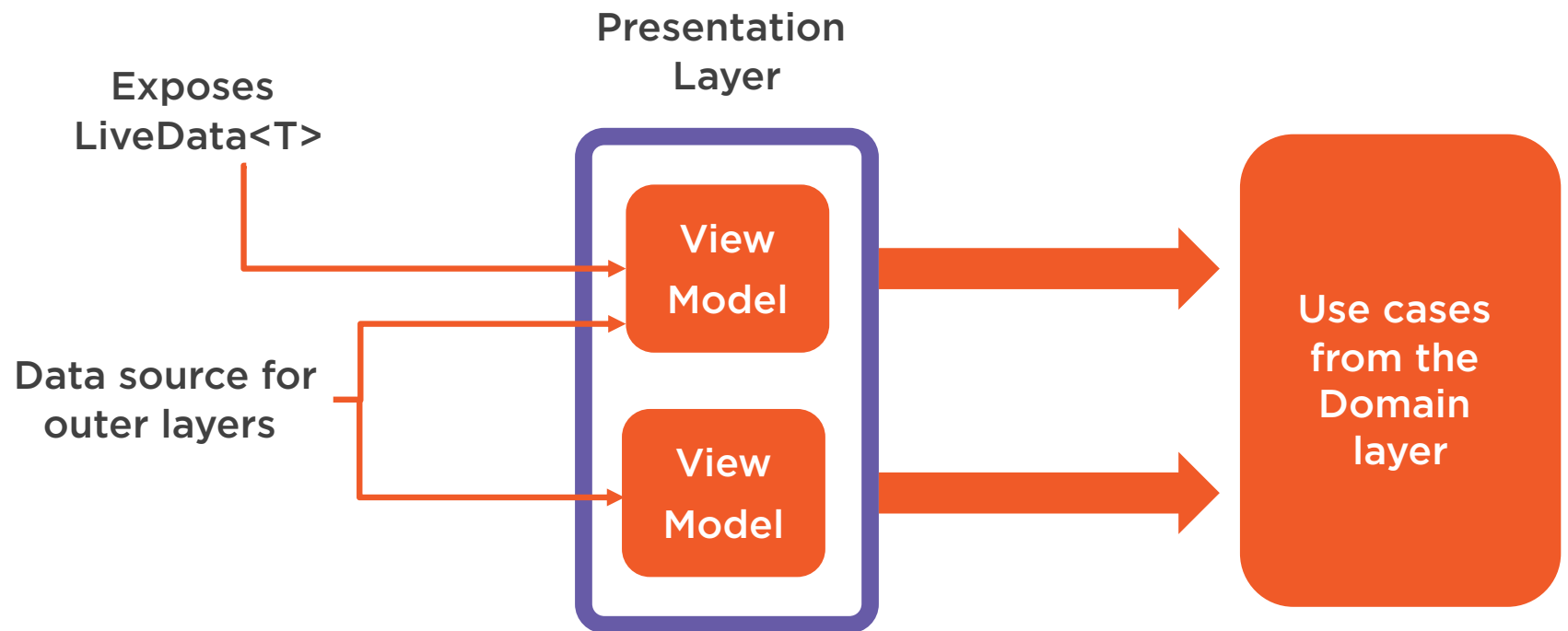
The Presentation Layer



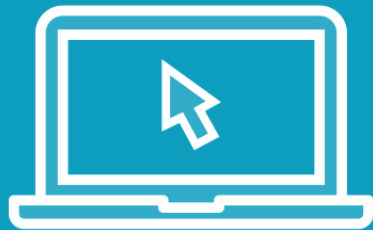
Banking App Layers



Presentation Layer Interaction



Demo



The presentation module

Home and transaction ViewModel

Domain layer interaction

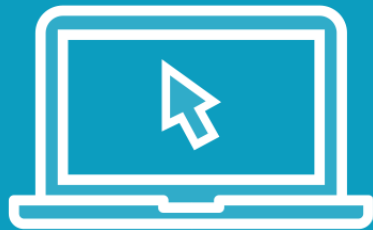
Interface adapters



Testing the Presentation Layer



Demo



Testing the home ViewModel

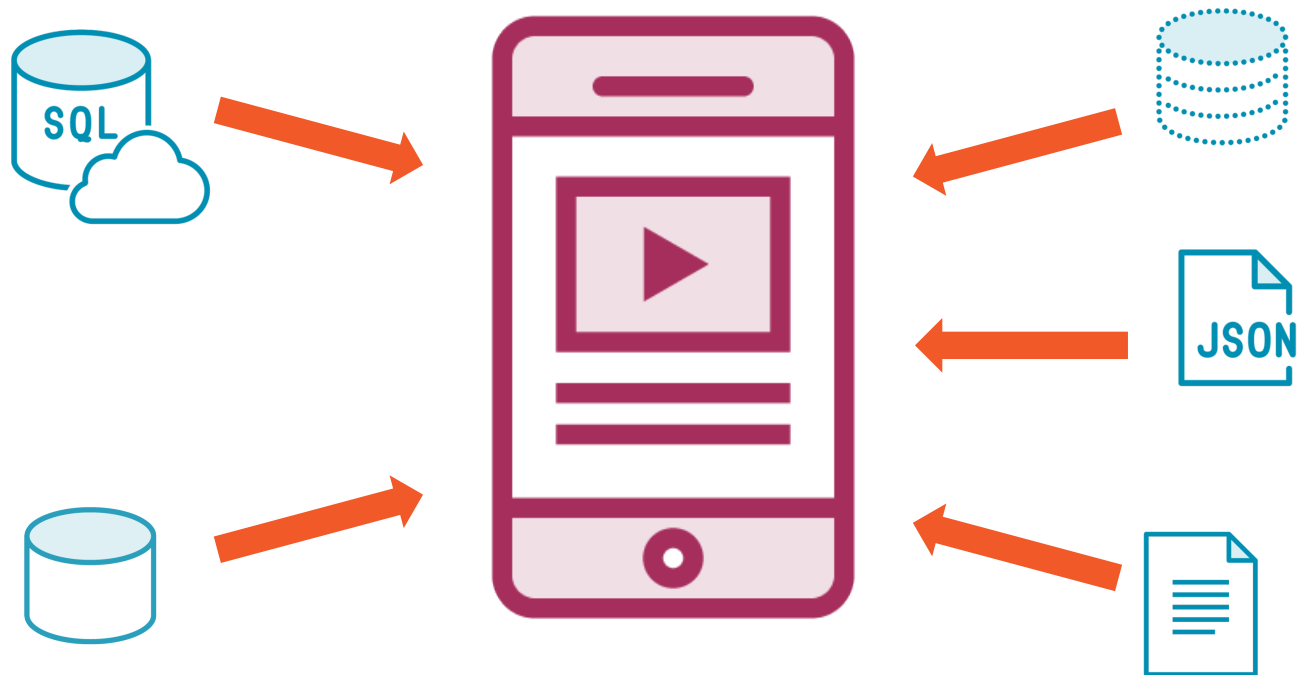
Testing the transactions ViewModel



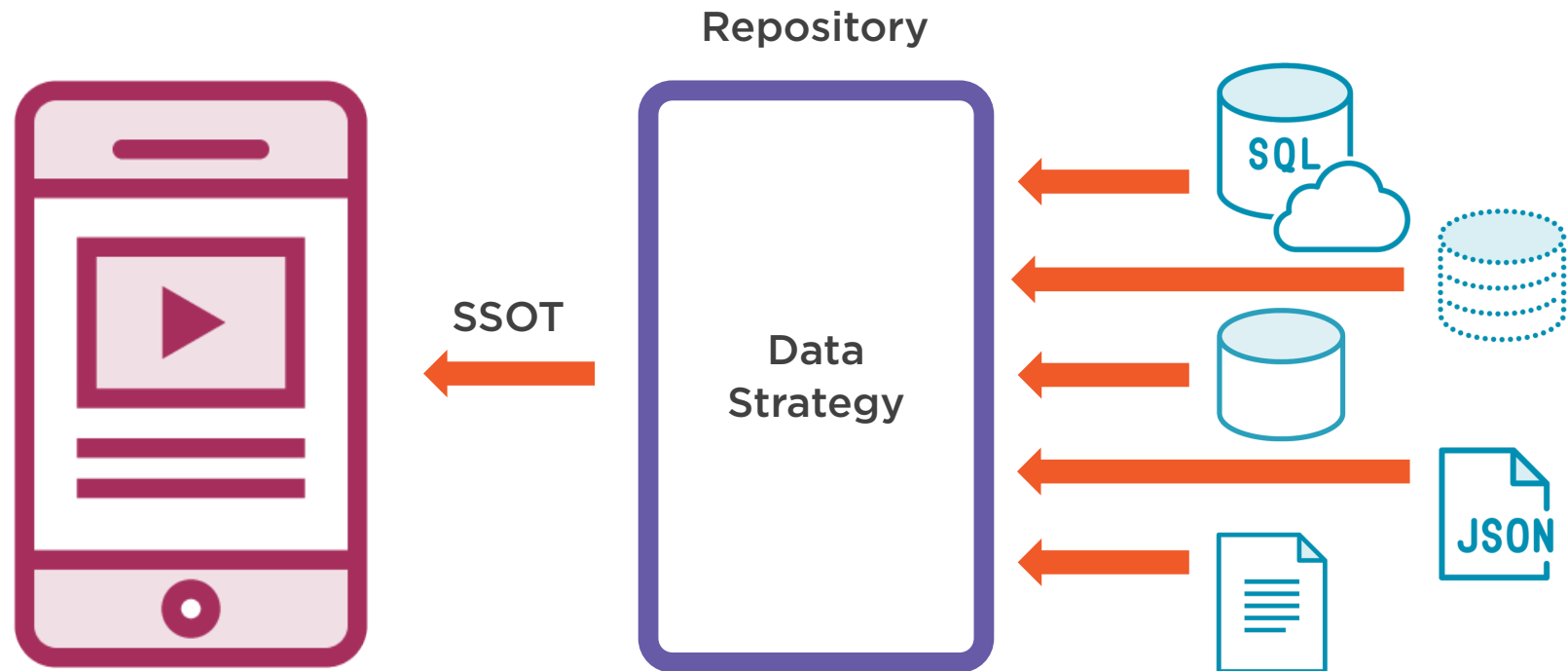
The Repository Pattern



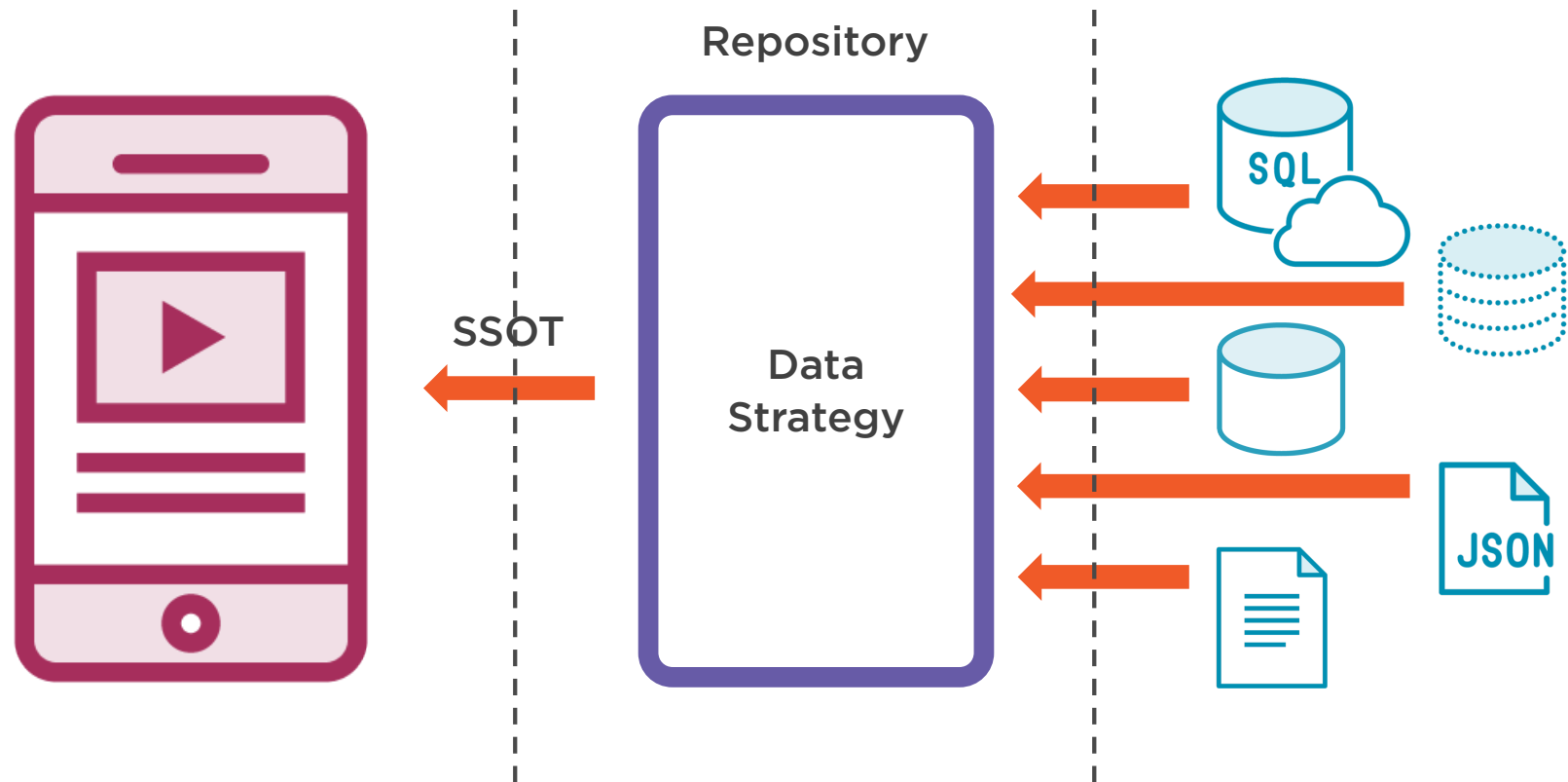
Data Sources



Data Sources



Repository Pattern



Responsibilities of Repository

Interacts with data sources

Isolates user interface and data strategies

Wraps data strategies and policies

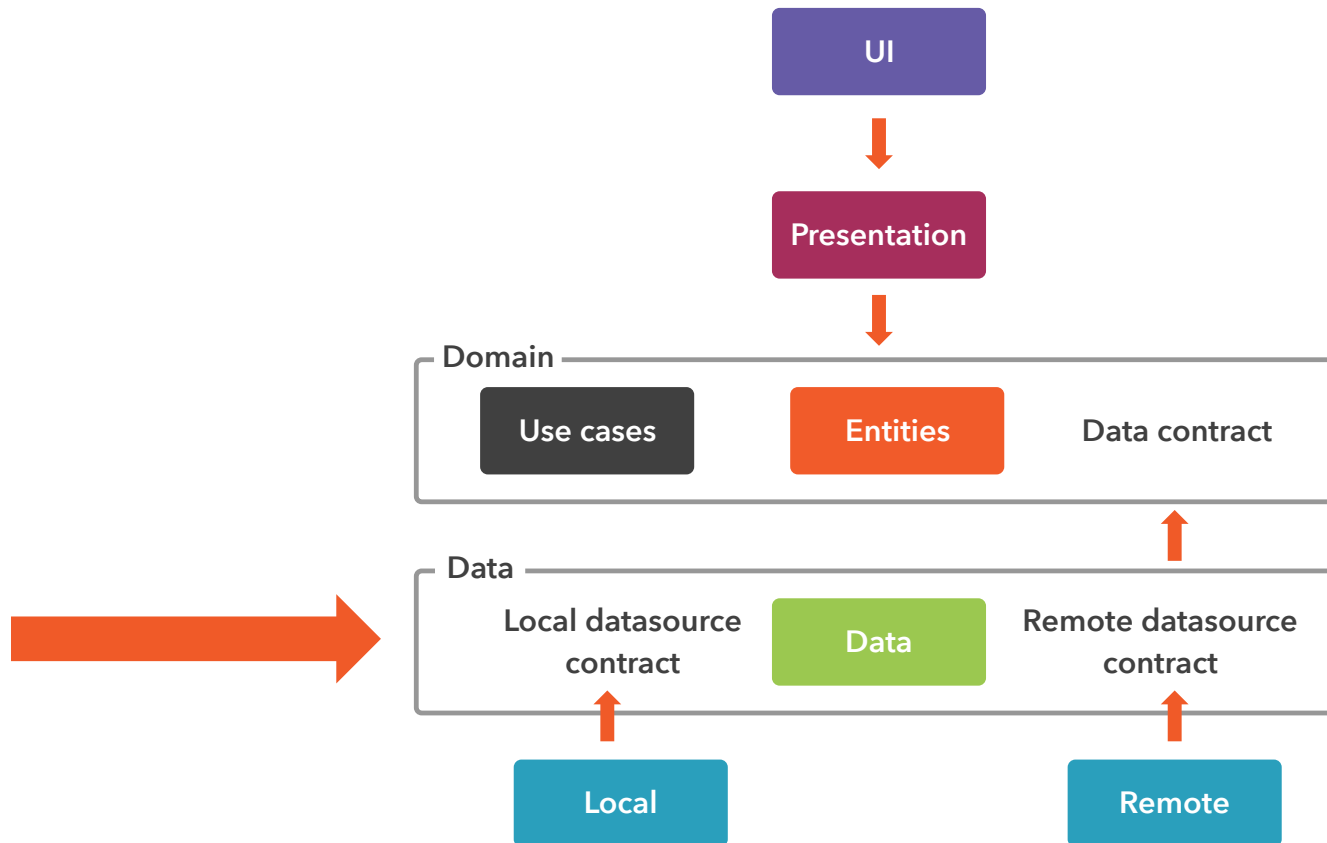
Data conversion



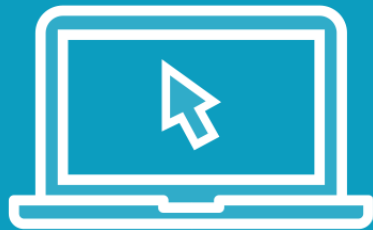
The Data Layer



Banking App Layers



Demo



Implement BankingRepository

Create abstractions for data sources

Implement a data strategy

Test the Data layer



Summary



AAC – ViewModel and LiveData

The presentation layer

Presentation layer testing

The repository pattern

The data layer

Data layer testing

