

# Getting Familiar with Android Architecture Components

---



**Annapurna Agrawal**

AUTHOR

@annapurna\_23 [linkedin.com/in/annapurna-agrawal](https://linkedin.com/in/annapurna-agrawal)



# Overview



**Lifecycle-aware components**

**LifecycleOwners and LifecycleObserver**

**ViewModel and its scope**

**LiveData**

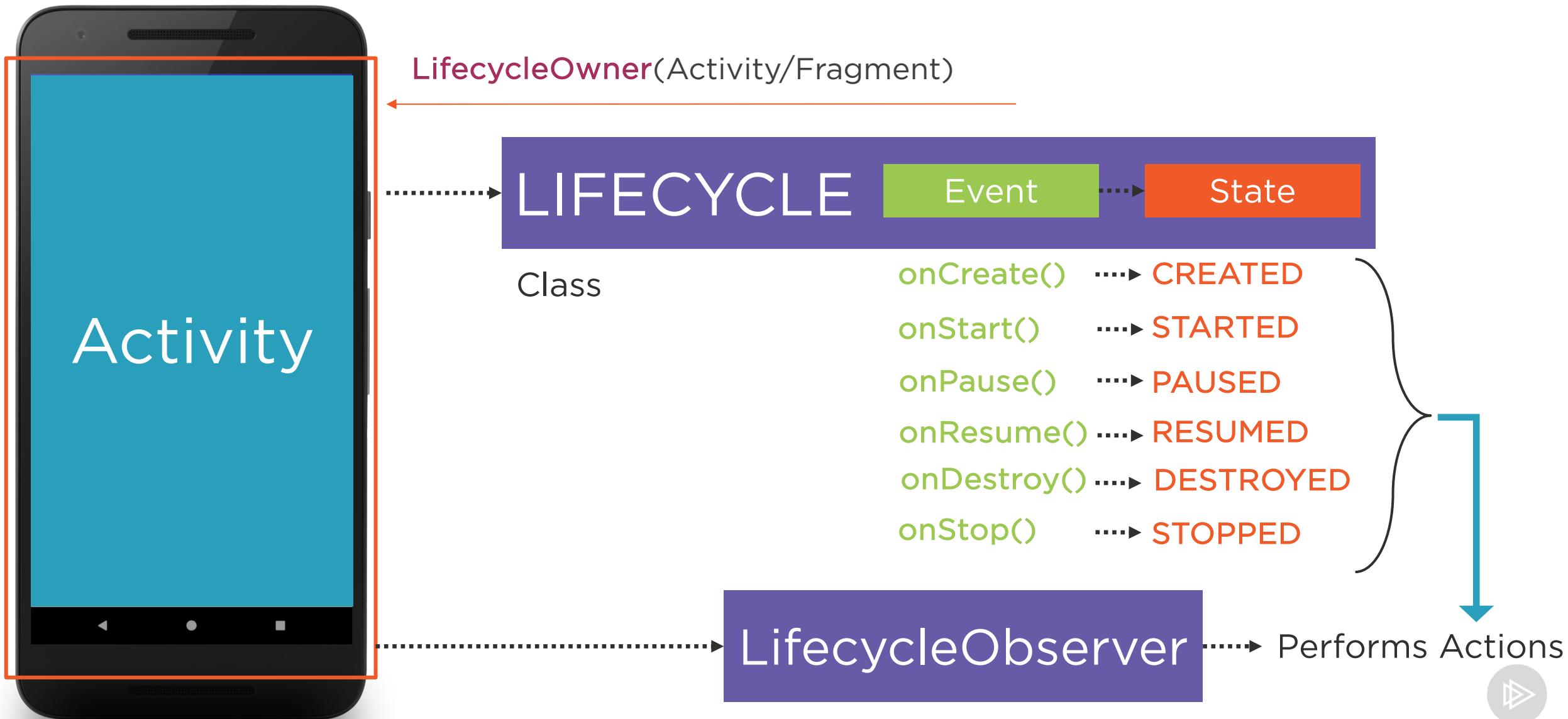


# Exploring Lifecycle-aware Components

---



# Lifecycle-aware Components

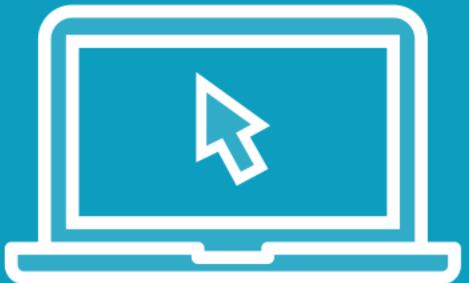


# Lifecycle-aware Components Demo

---



Demo



Exploring how Lifecycle-aware  
components work



# Lifecycle-aware Components

**Responds to the change in lifecycle status of LifecycleOwner**

**Lifecycle class holds info about lifecycle of LifecycleOwner**

**Lifecycle object uses following enumeration to track lifecycle status**

- Event
- State



# Lifecycle-aware Components

**LifecycleOwner provides lifecycle status to  
Lifecycle-aware components**

**LifecycleObserver registers the lifecycle  
status to respond and perform actions**

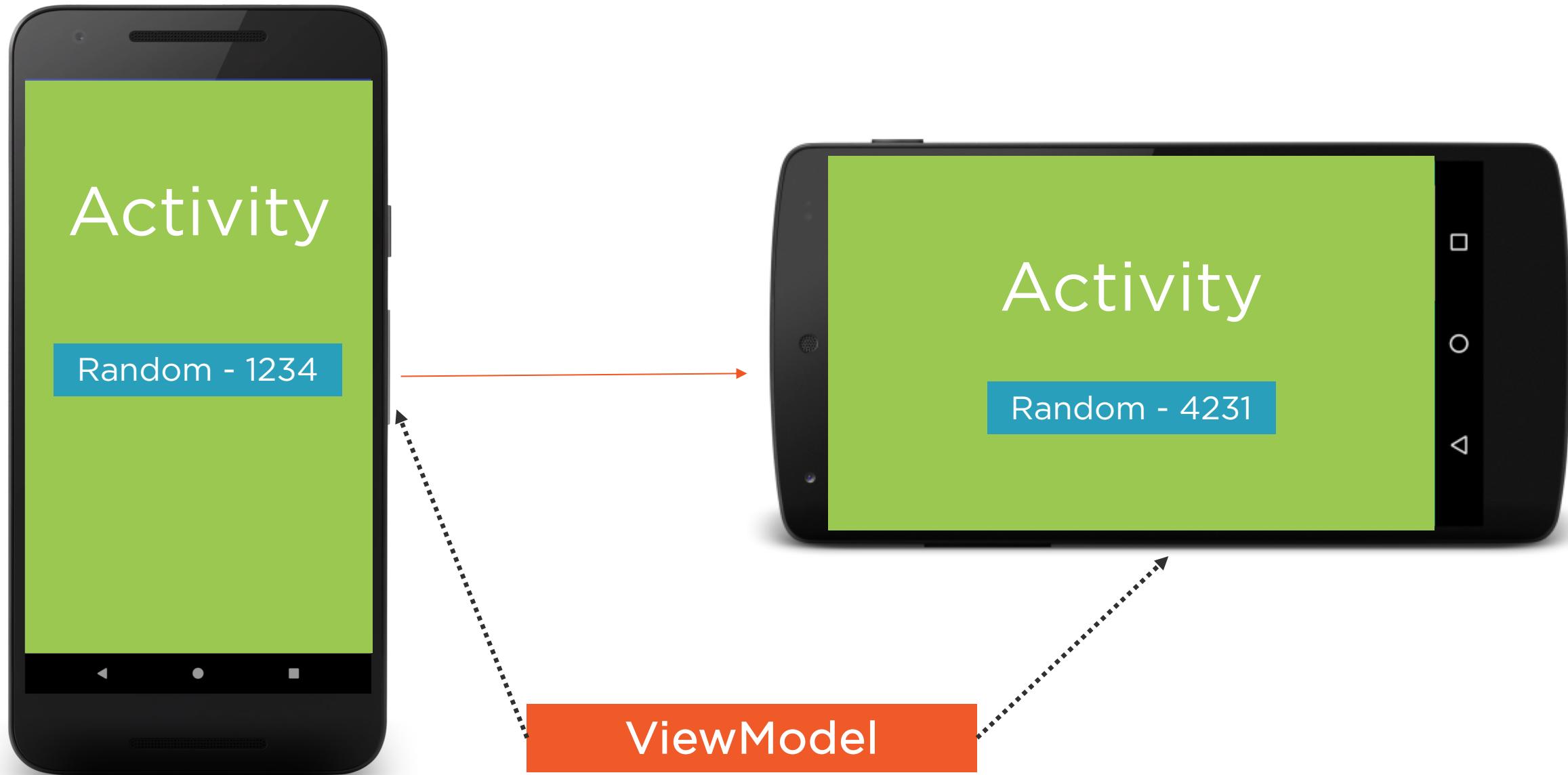


# Why ViewModel?

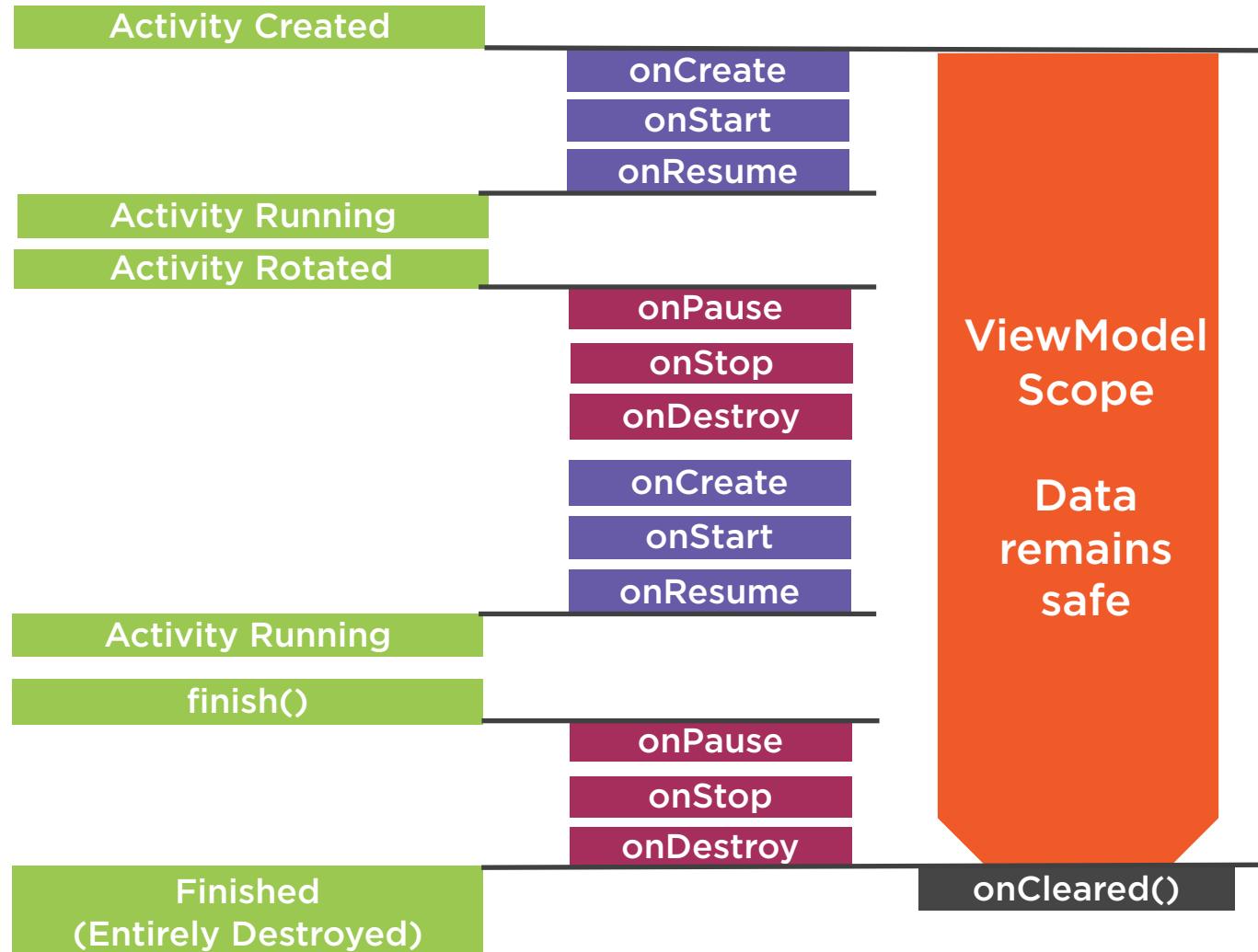
---



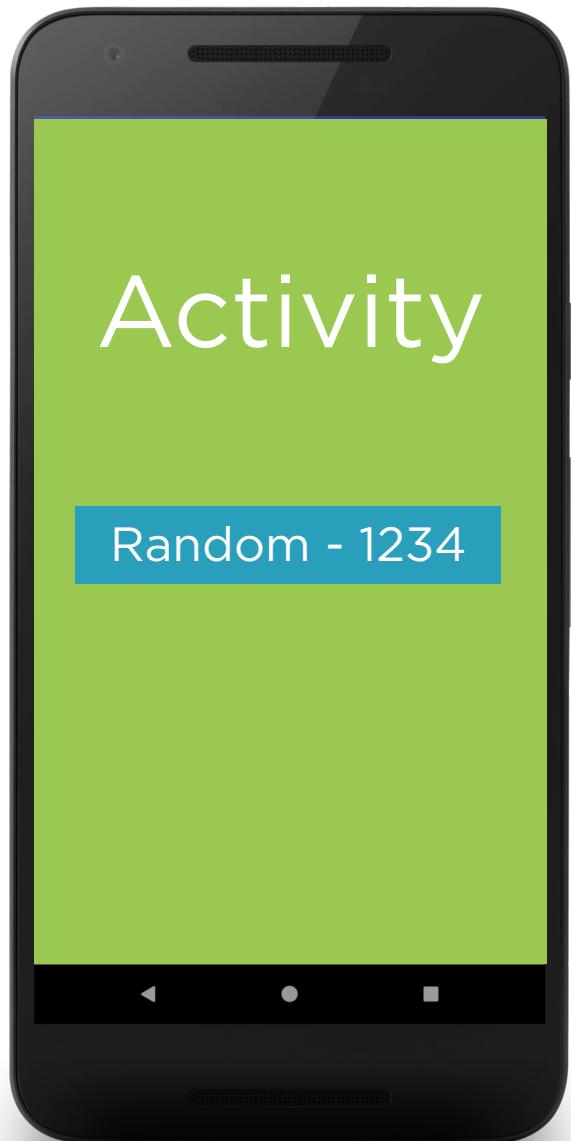
# Why ViewModel?



# Scope of ViewModel



# ViewModel

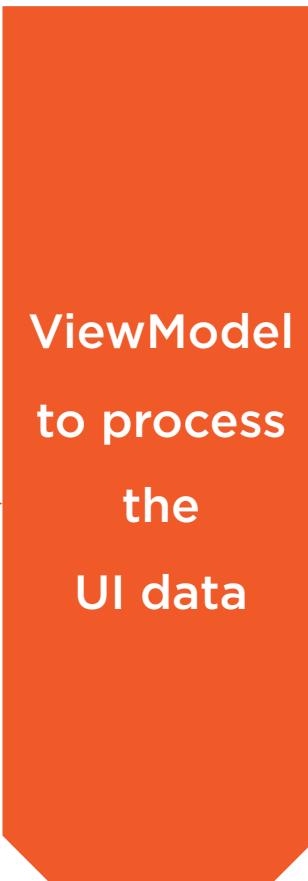
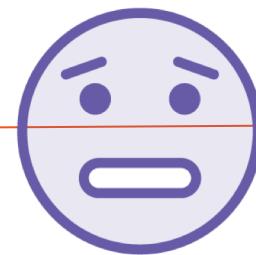


Display UI Data

React to user action

Handle OS communication

Load data from network/DB



# ViewModel

**Survives configuration changes**

- Screen Rotation

**Not same as `onSaveInstanceState()`**

**Used for large data such as bitmap or user list**

**Store and manage UI related data**

**Communication layer between DB and UI**

**Destroyed only if the owner Activity is completely destroyed `onCleared()`**



# ViewModel Demo

---



Demo



**Exploring how ViewModel survives the config changes**

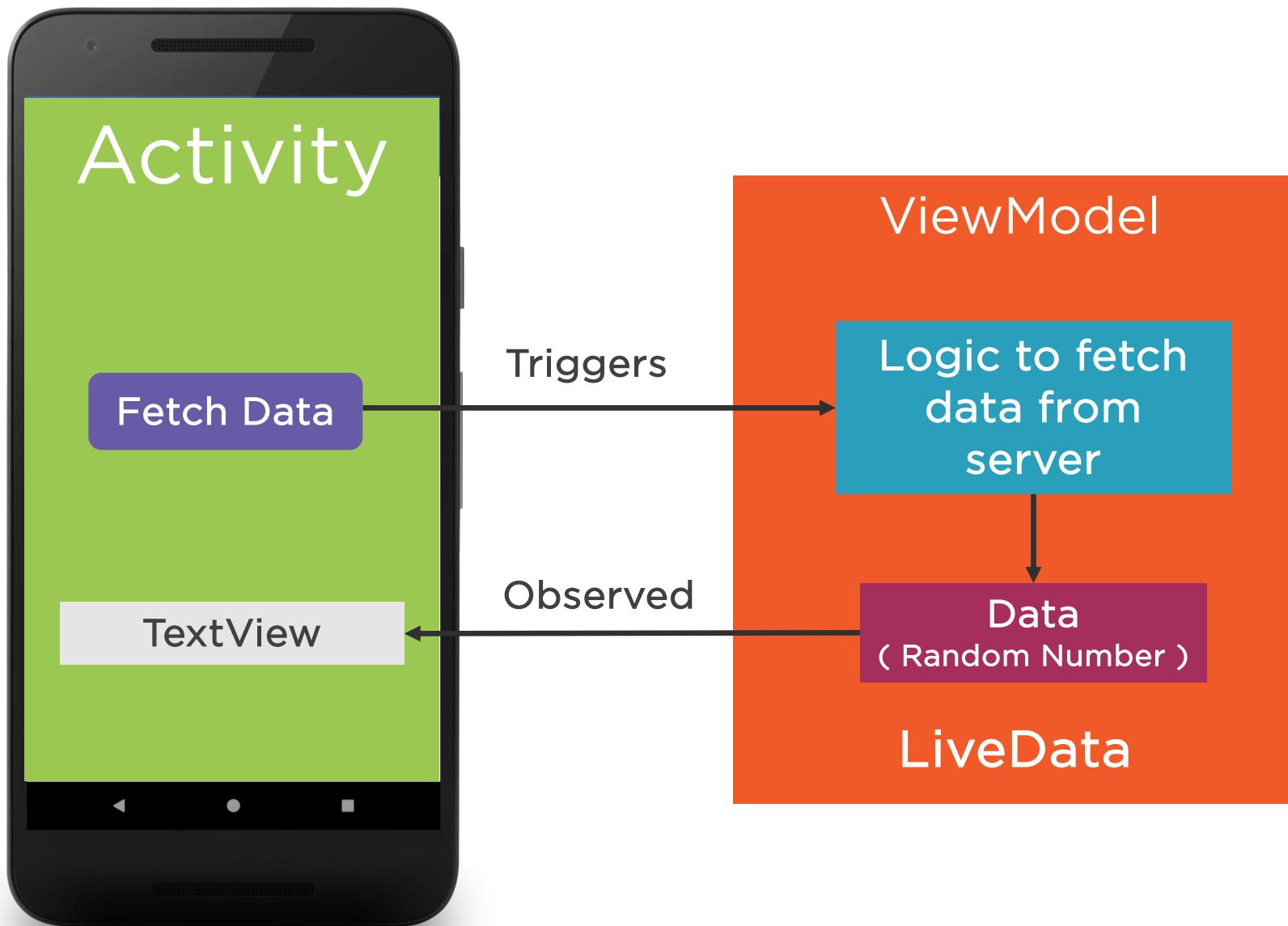


# Understanding LiveData

---



# Problem Statement with Solution

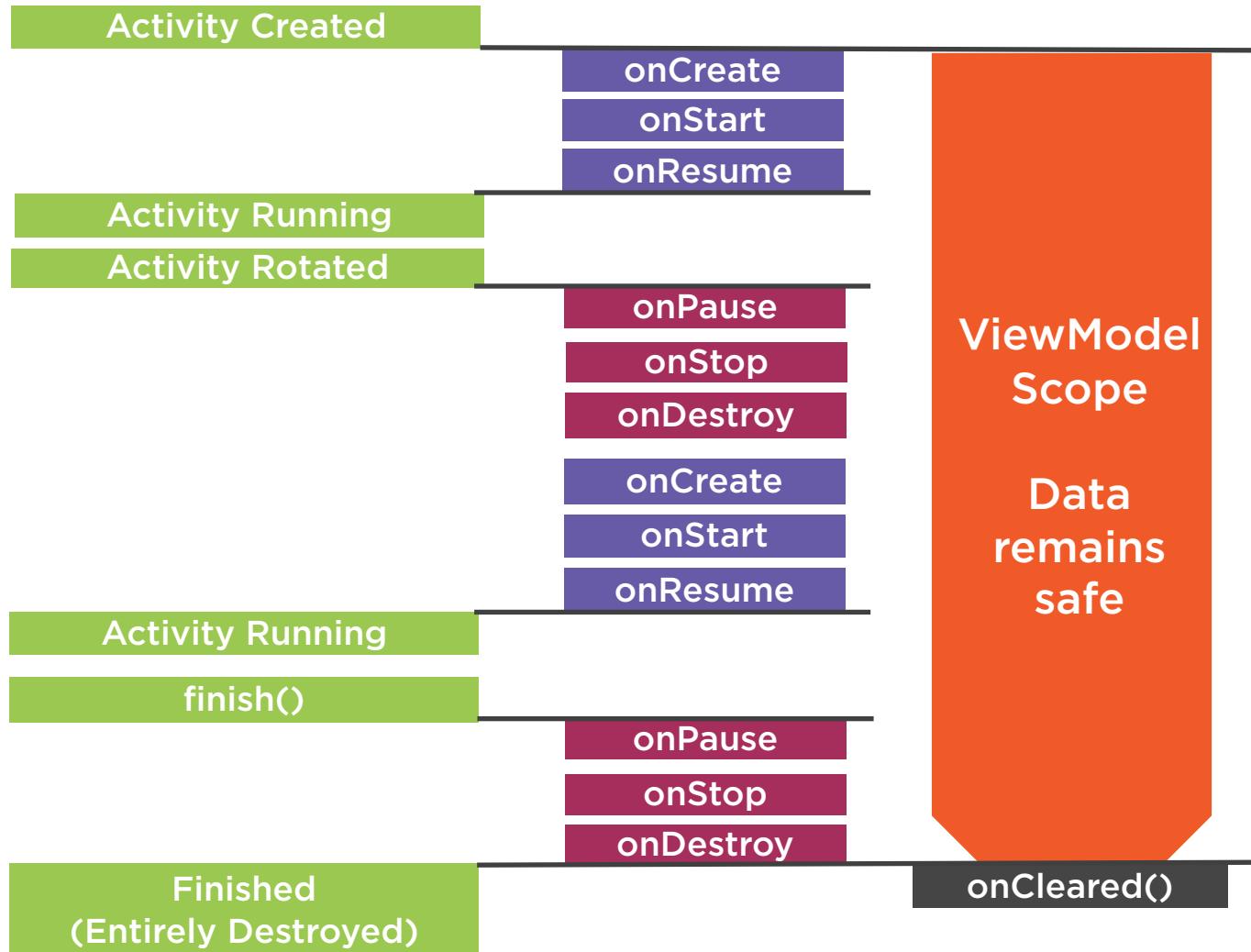


Our traditional approach

- Using Interface
- Using Event Bus such as Otto
- Using LiveData



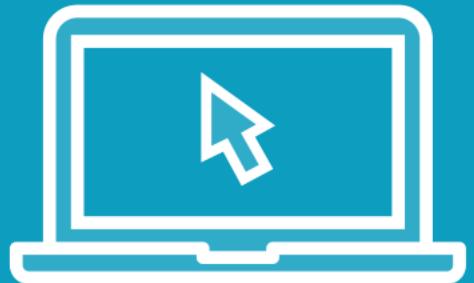
# Scope of ViewModel



# LiveData with ViewModel



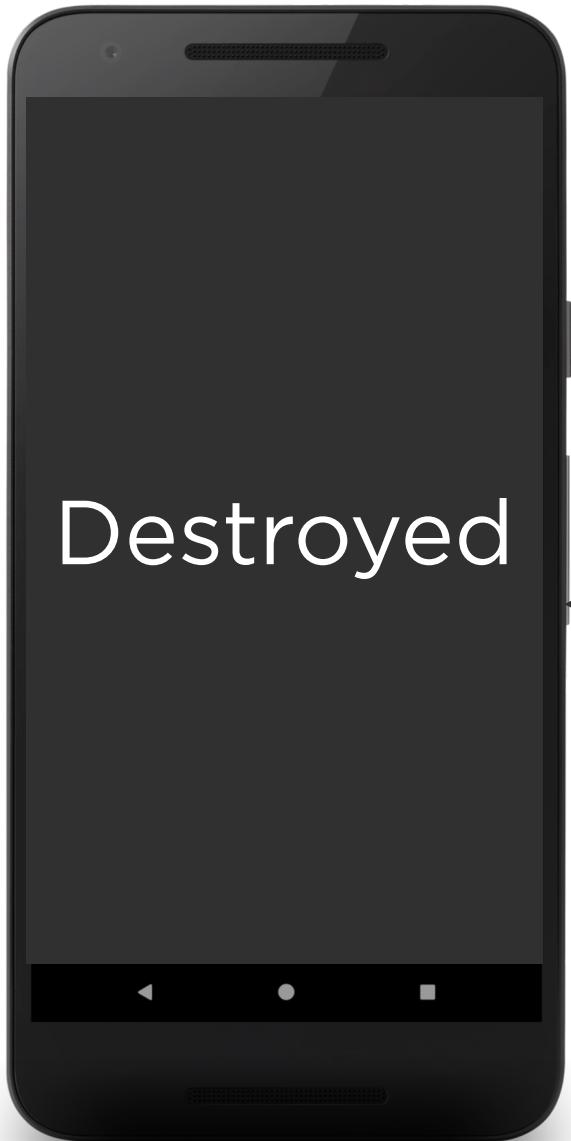
Demo



**Exploring how LiveData works with  
ViewModel**



# LiveData



Activity stops observing LiveData

Case1: When Activity is in **paused/stop state**

Case2: When Activity is **destroyed**



# LiveData

**Observable data holder class**

**Keeps data and allows data to be observed**

**Observe LiveData from app component's  
onCreate() method**



# LiveData

## Benefits of LiveData

- Keeps the UI updated in case of changes
- Automatically destroyed when associated LifecycleOwner is destroyed
- No crashes due to stopped activities
- Can be shared by multiple resources

**Best when used with ViewModel**



# Summary



**Lifecycle-aware components**

**Relation between LifecycleOwners and LifecycleObserver**

**ViewModel and its scope**

**How LiveData works with ViewModel**

