GEMINI SOLUTIONS

MVVM

Android architecture insight

Mecea Mihai
Gemini Solutions

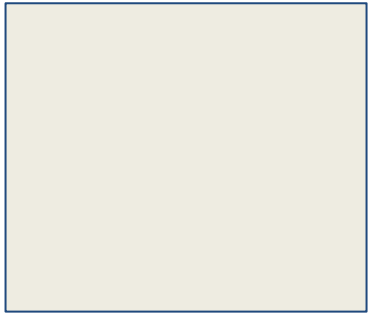
Activity

Everything goes in-here

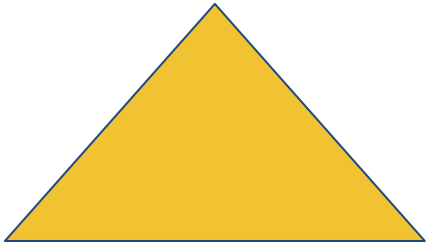
```
4829                    not
4830            }
4831        }
4832
4833    }
4834
```

Duplicates

AnActivity

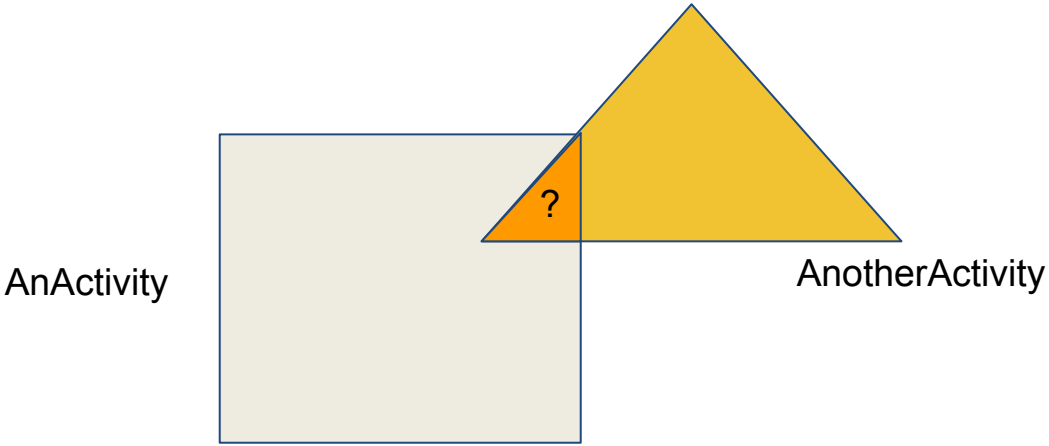


AnotherActivity



Duplicates

What's a code duplicate?

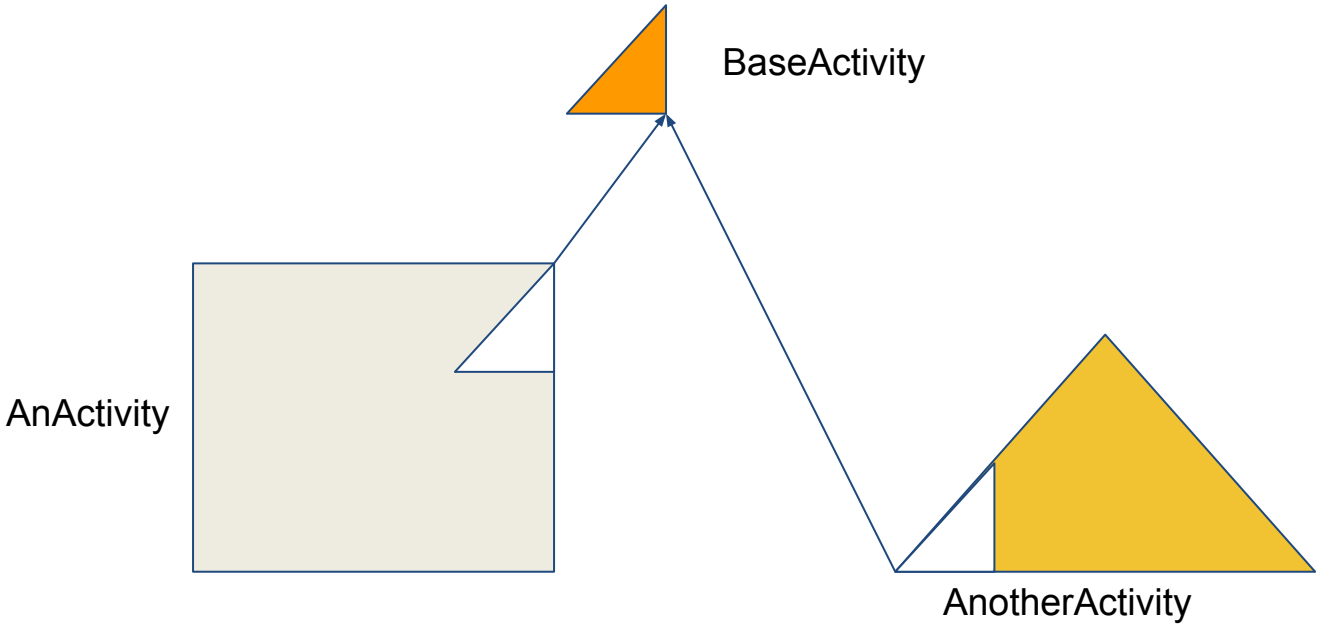


“Code duplicates are like, the worst!”

GEMINI SOLUTIONS

Inheritance FTW

Duplicates



Duplicates

BaseActivity

But what’s actually in it?

Duplicates

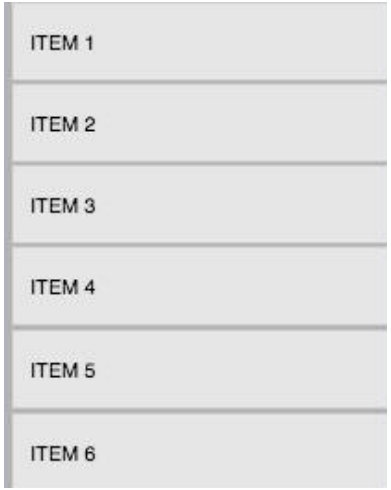
BaseActivity

- Is this code boilerplate?
- Is it a feature?
- Is it the actual feature or it just looks the same?

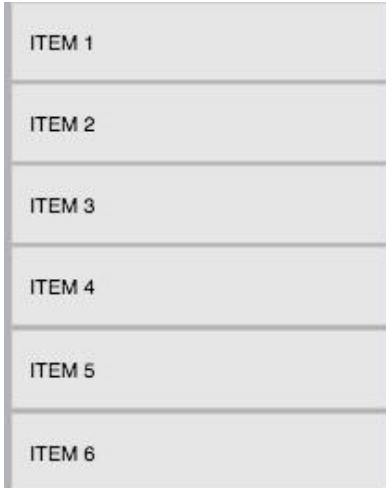
The two look-alike adapters case study

GEMINI SOLUTIONS

Feature A



Feature B

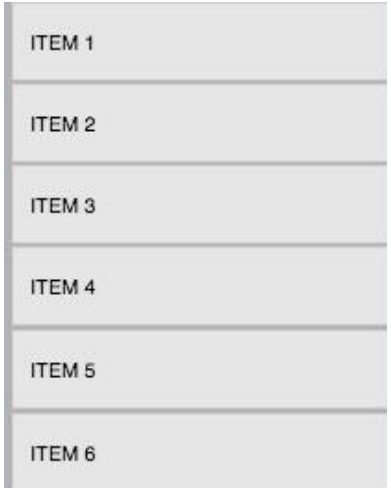



```
FeatureAdapter {
        + text
}
```

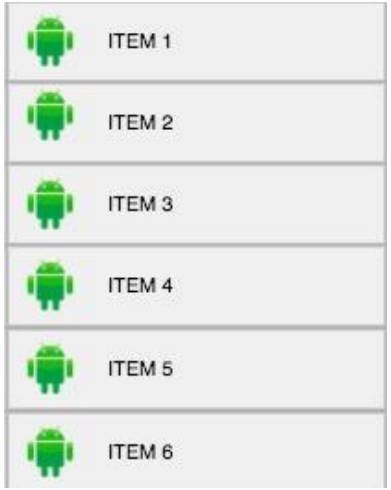
The two look-alike adapters case study

GEMINI SOLUTIONS

Feature A



Feature B



The two look-alike adapters

GEMINI SOLUTIONS

The power of "copy/paste"

```
AFeatureAdapter {
        + text
}


BFeatureAdapter {
        + text
        + image
}
```

The power of "if"

```
THEAdapter {
        + text
        if (this is the one with a picture) {
                + image
        }
}
```

The two look-alike adapters

GEMINI SOLUTIONS

The power of "copy/paste"

```
AFeatureAdapter {
        + text
}

BFeatureAdapter {
        + text
        + image
}
```

The power of "if"

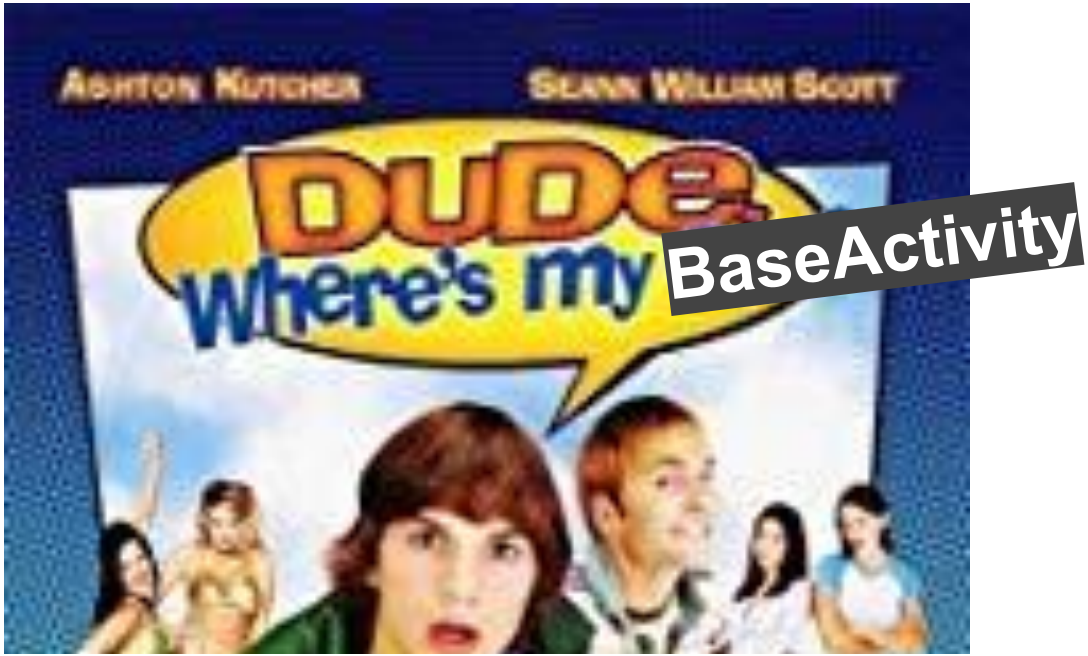
```
THEAdapter {
        + text
        if (this is the one with a picture) {
                + image
        }
}
```

5 years later?

“Prefer composition over inheritance”

GEMINI SOLUTIONS

...

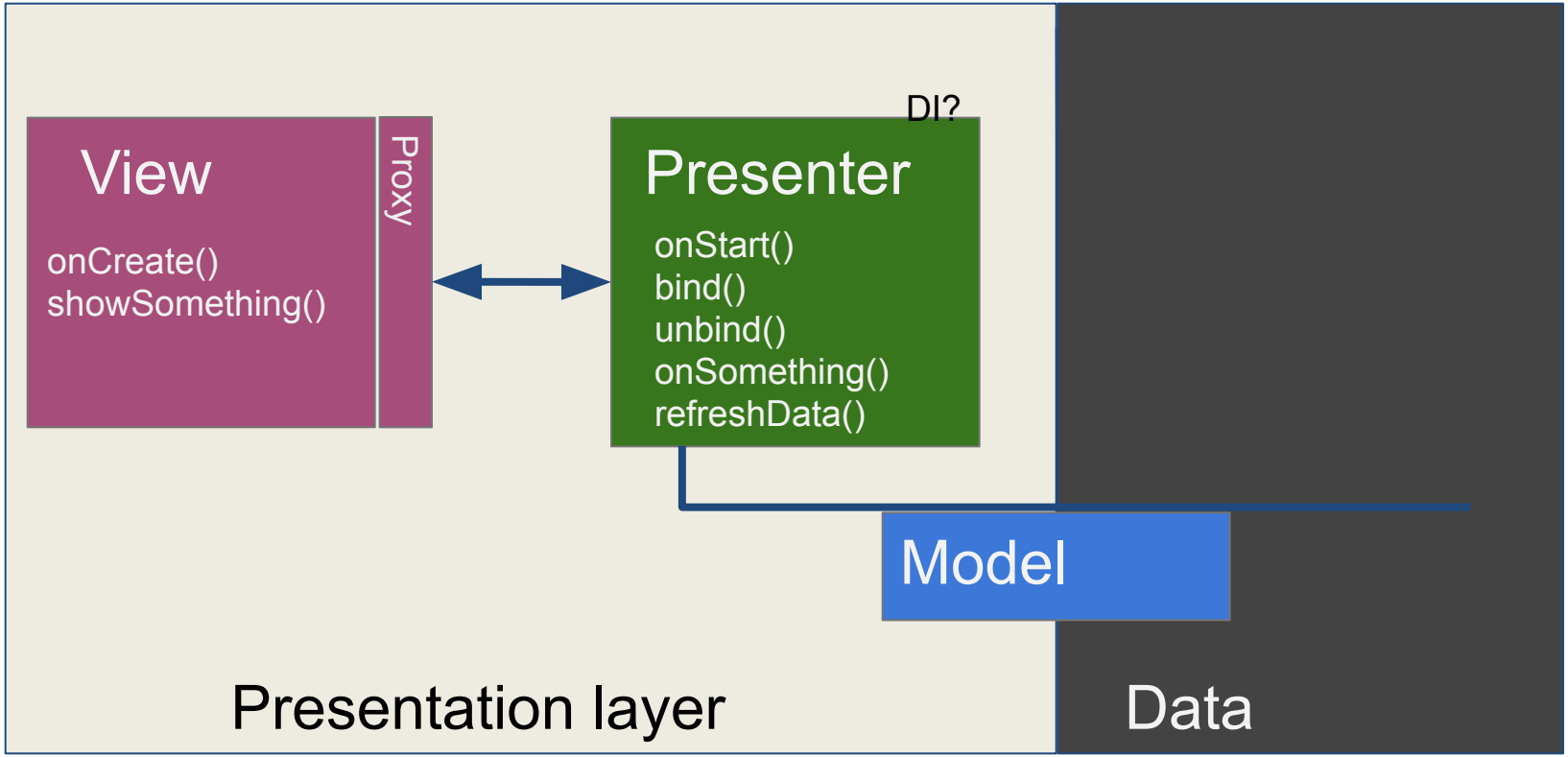

Ashton Kutcher
Seann William Scott
Dude,
Where's my
BaseActivity

MVP

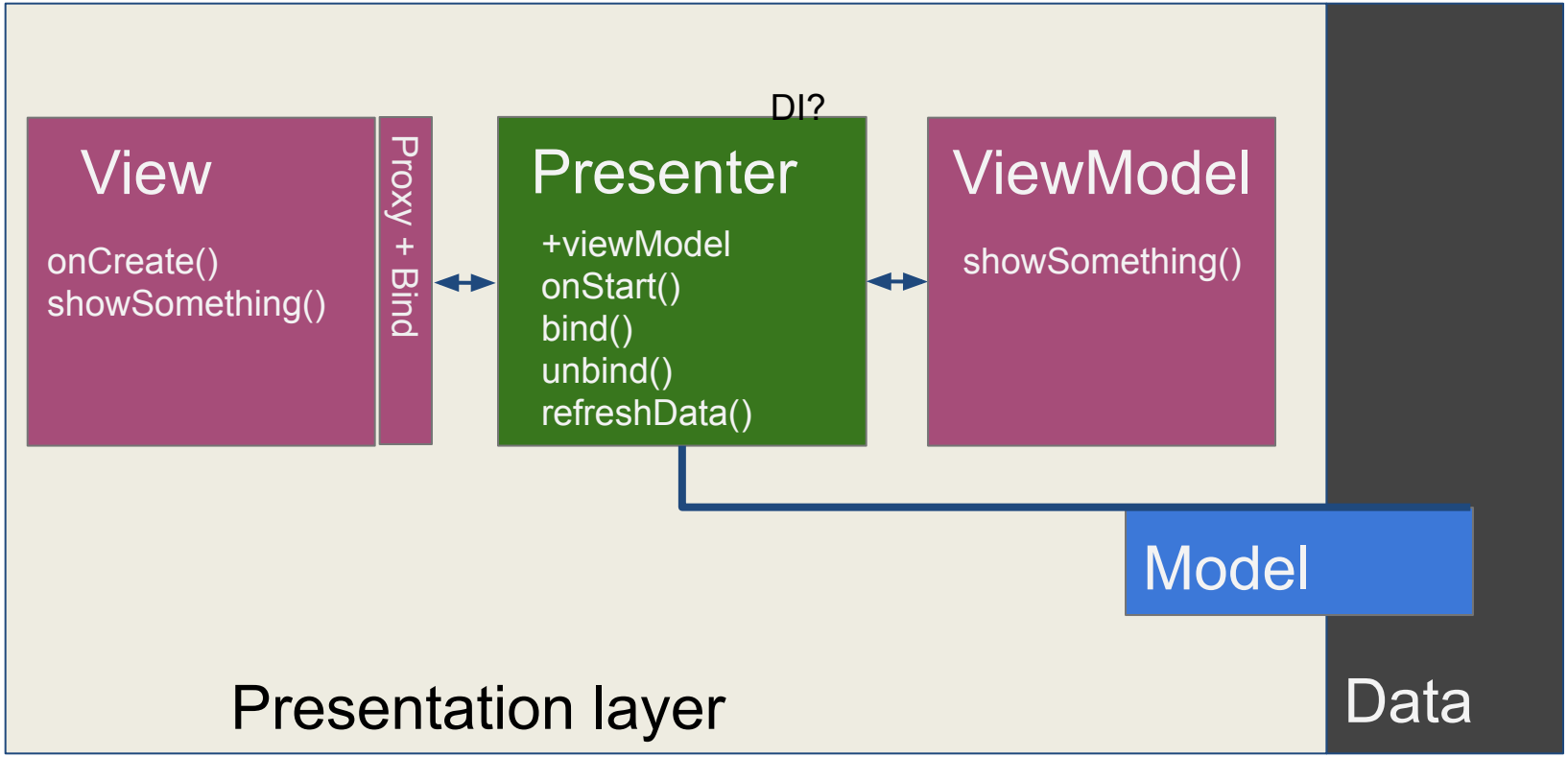
GEMINI SOLUTIONS

Taking a hammer to that “Activity”

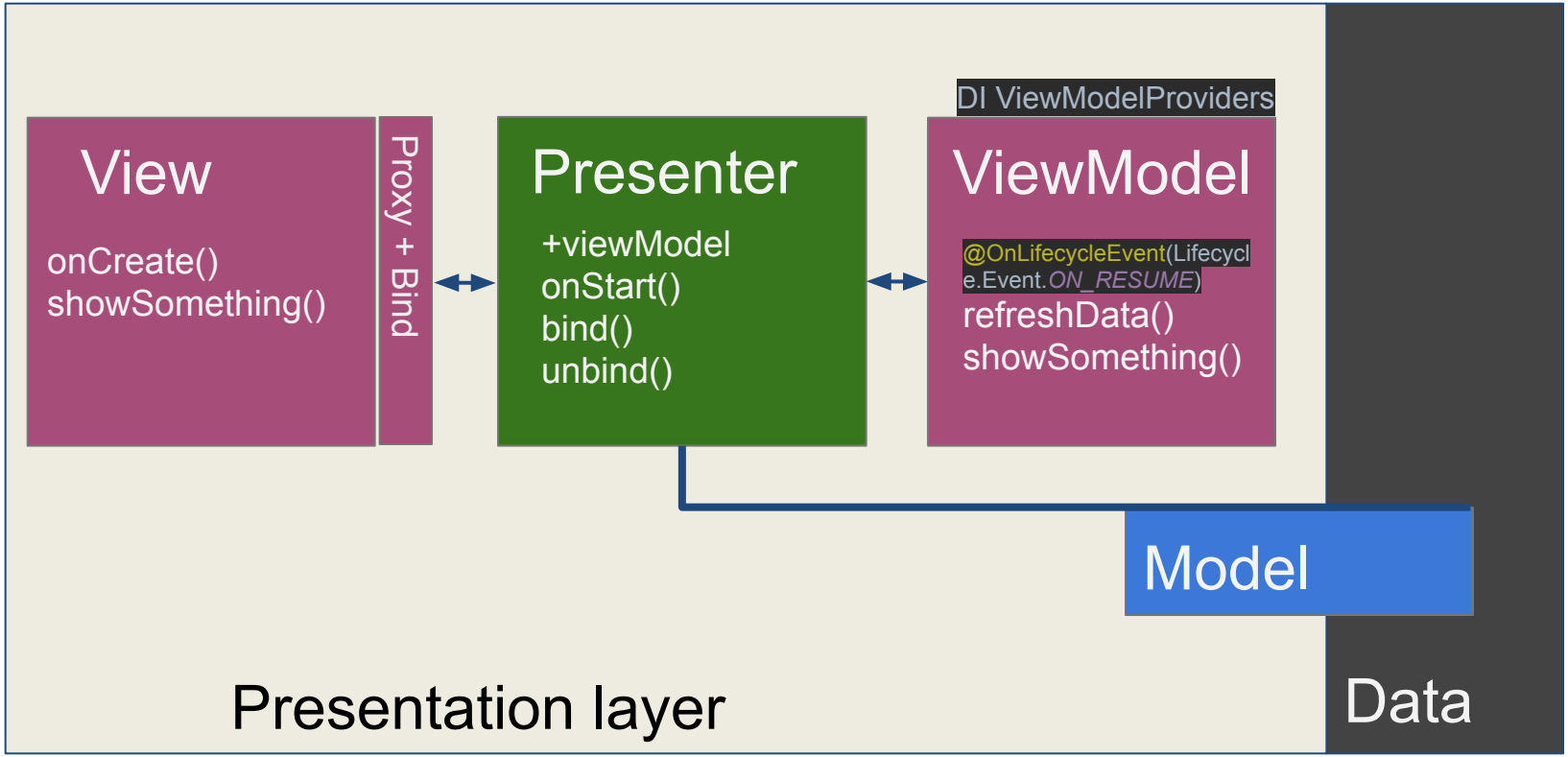
MVP


View
onCreate()
showSomething()
Proxy
DI?
Presenter
onStart()
bind()
unbind()
onSomething()
refreshData()
Model
Presentation layer
Data

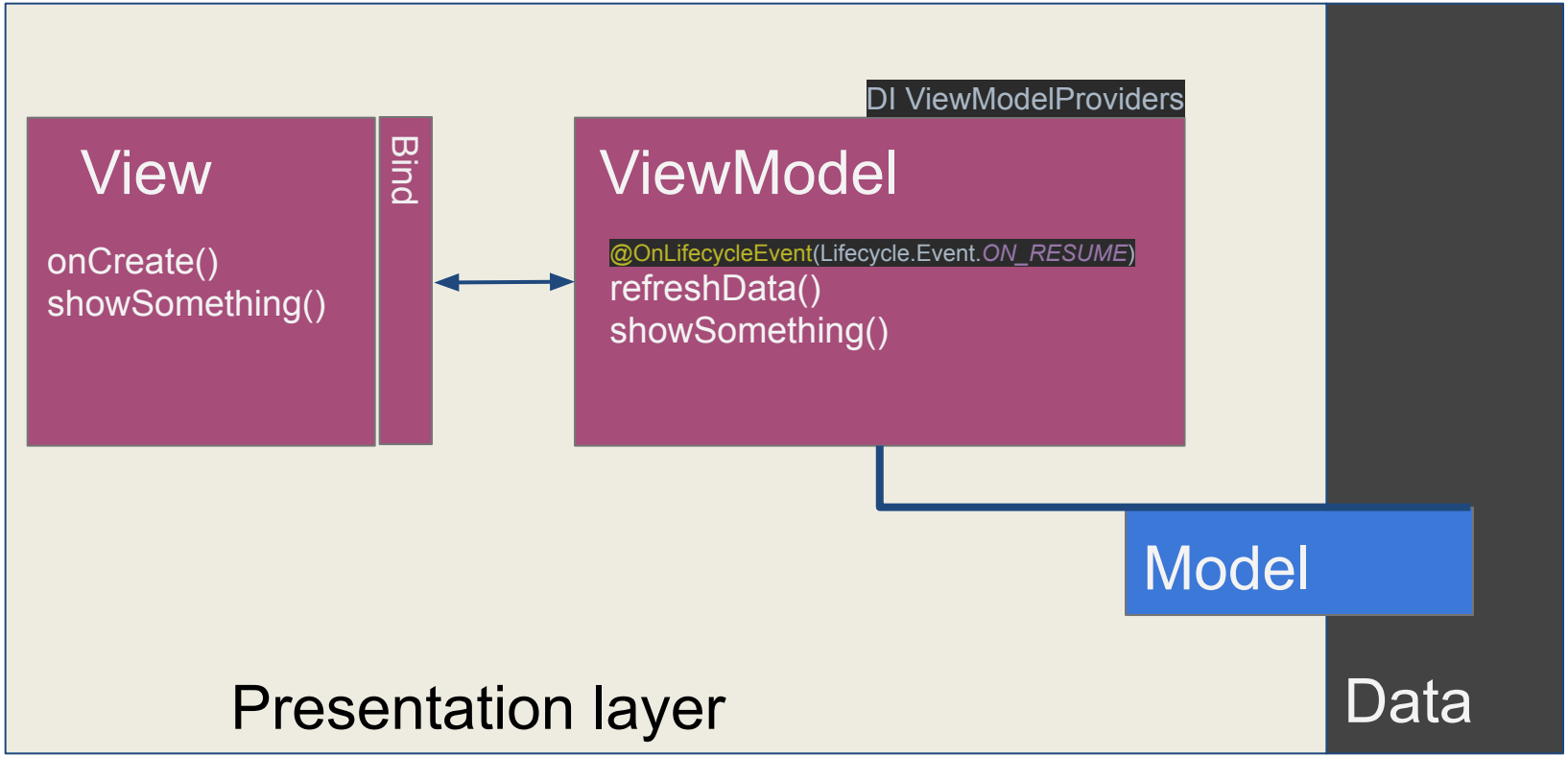
MVVMP


View
onCreate()
showSomething()
Proxy + Bind
DI?
Presenter
+viewModel
onStart()
bind()
unbind()
refreshData()
ViewModel
showSomething()
Model
Presentation layer
Data

MVVMP

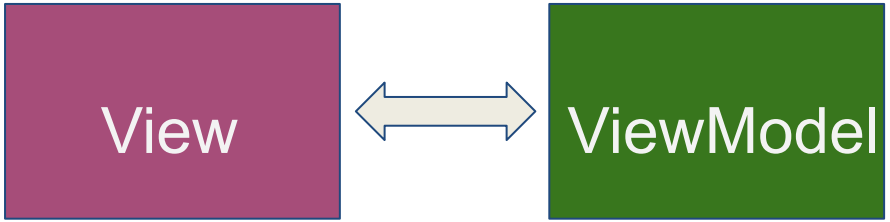

View
onCreate()
showSomething()
Proxy + Bind
Presenter
+viewModel
onStart()
bind()
unbind()
DI ViewModelProviders
ViewModel
@OnLifecycleEvent(Lifecycle.Event.ON_RESUME)
refreshData()
showSomething()
Model
Data
Presentation layer

MVVM

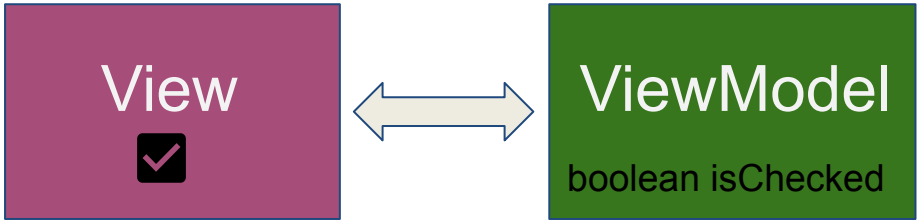

View
onCreate()
showSomething()
Bind
DI ViewModelProviders
ViewModel
@OnLifecycleEvent(Lifecycle.Event.ON_RESUME)
refreshData()
showSomething()
Model
Presentation layer
Data

What is MVVM?

Allows you to **bind UI components** in your layout to **data**.



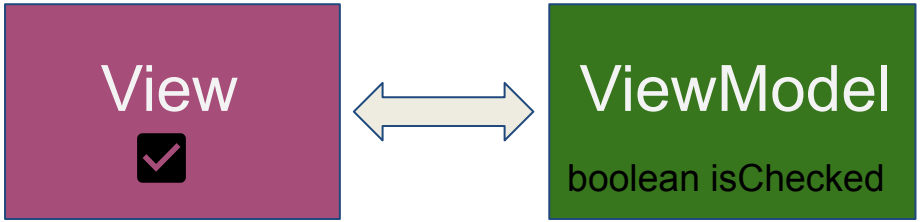
V <-> VM



View and ViewModel mirror each other:

- If a Checkbox view is checked by the user
  - then the ViewModel will have boolean isChecked == true
- If in coding we set the isChecked = true
  - Then the Checkbox view is shown checked

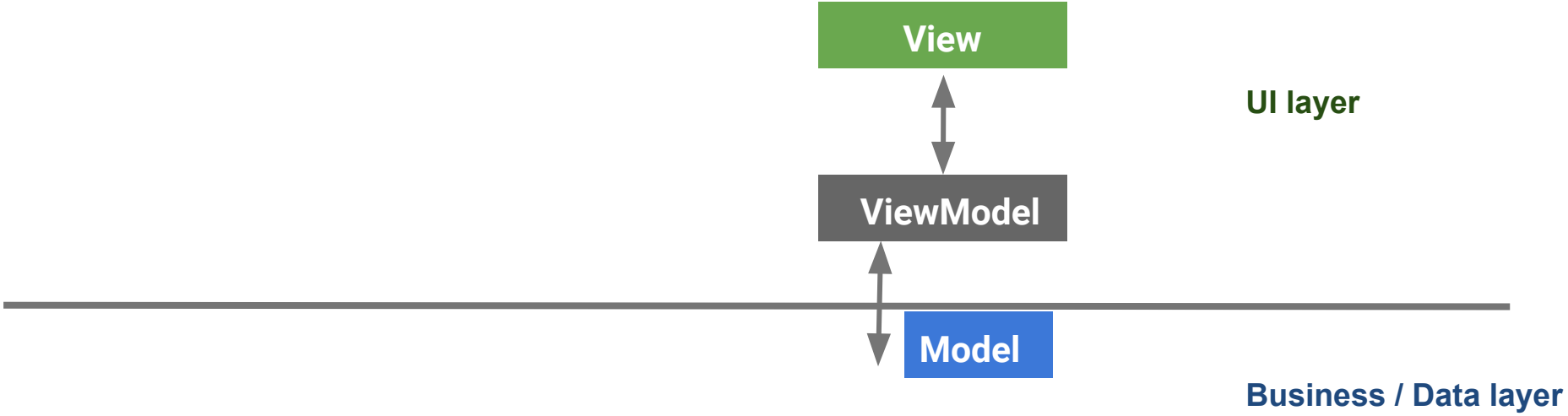
V <-> VM



V <-> VM two way binding

V -> VM one way binding
V <- VM one way binding

MVVM - overview



MVVM

How bad could it go?

MVVM

GEMINI SOLUTIONS

```
public class SomeViewModel extends ViewModel {
...
    void onButtonPressed(int resourceButtonPressed) {
        User user = User.getCurrentUser();
        String token = Preferences.getInstance(getContext()).getString(Preferences.USER_TOKEN, "");
        RetrofitApiService.doRestCallForUser(getContext(), token, new Callback<Map<String,Object>>() {
            @Override
            public void onResponse(Call<Map<String, Object>> call, Response<Map<String, Object>> response) {
                if (response.code() == 200) {
                    Util.logDebugMessage(RetrofitApiService.class.getSimpleName(), response.body().toString());
                } else {
                    prefs.writeSomething...
            }
    }
...
}
```

MVVM

GEMINI SOLUTIONS

Spaghetti integration

What’s wrong with spaghetti?

GEMINI SOLUTIONS

Documentability

GEMINI SOLUTIONS

How you gonna document that?

- and then wrote a callback for request
- and then i saw it had a method called onResponse
- and then i checked for 200
- and then…

If it’s not documentable, it’s probably not the best design

Testability

GEMINI SOLUTIONS

- Don’t have to actually test it
- Mocking comes natural when units are small and well thought

if it’s not testable, it’s probably not the best design

Complexity

GEMINI SOLUTIONS

if it’s not stupid simple, it’s probably not the best design

MVVM

GEMINI SOLUTIONS

```
public class SomeViewModel extends ViewModel {
...
    void onButtonPressed(int resourceButtonPressed) {
        User user = User.getCurrentUser();
        String token = Preferences.getInstance(getContext()).getString(Preferences.USER_TOKEN, "");
        RetrofitApiService.doRestCallForUser(getContext(), token, new Callback<Map<String,Object>>() {
            @Override
            public void onResponse(Call<Map<String, Object>> call, Response<Map<String, Object>> response) {
                if (response.code() == 200) {
                    Util.logDebugMessage(RetrofitApiService.class.getSimpleName(), response.body().toString());
                } else {
                    prefs.writeSomething...
            }
    }
...
}
```

Complexity

GEMINI SOLUTIONS

Richard Branson

@richardbranson

Follow

Complexity is your enemy. Any fool can make something complicated. It is hard to make something simple virg.in/cye

What’s the right architecture?

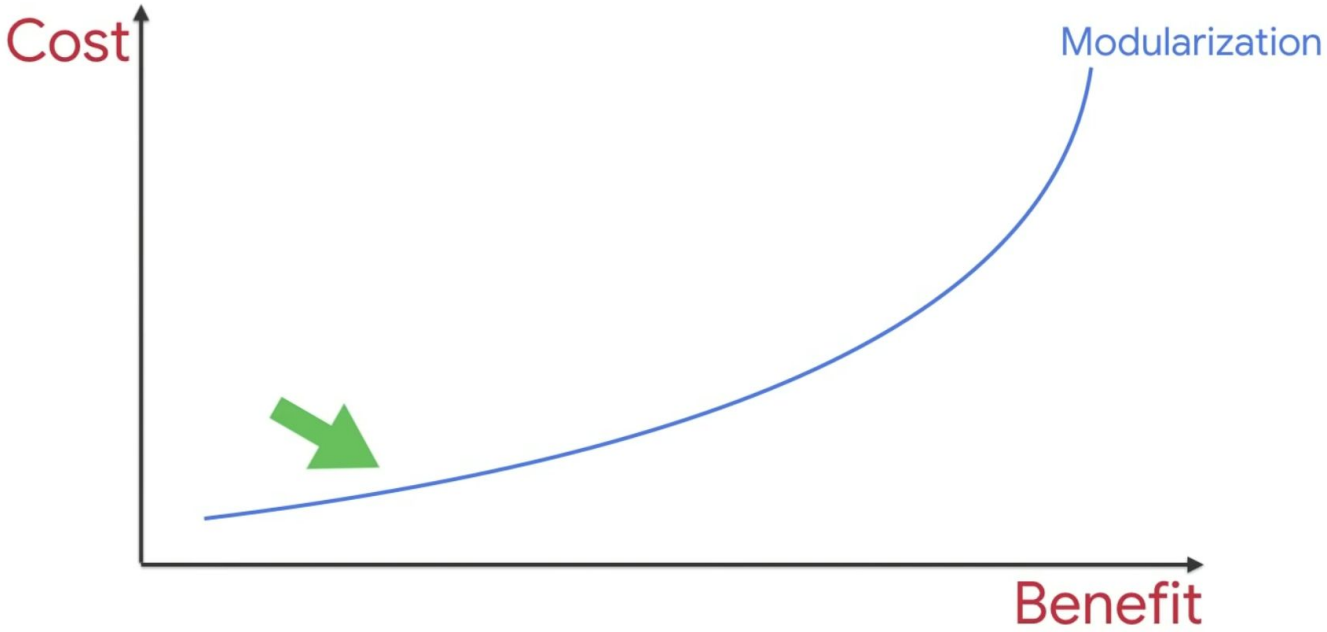
Mobile wise or even Android wise

GEMINI SOLUTIONS

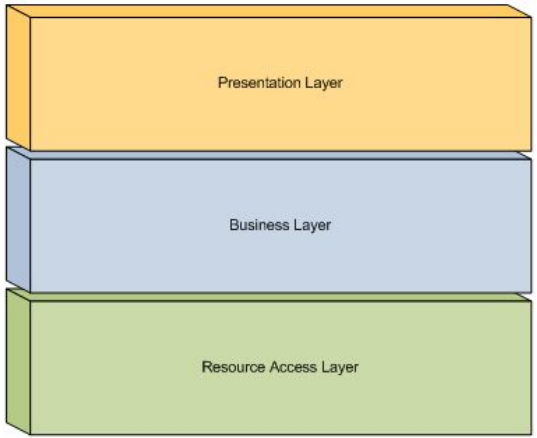
What’s the minimal architecture?

GEMINI SOLUTIONS

Modularization



Google I/O ‘19: Modular Android App


Presentation Layer
Business Layer
Resource Access Layer

Minimal architecture

GEMINI SOLUTIONS

Android Application Module

Dependency

Android Library Module

Minimal architecture

GEMINI SOLUTIONS

Android Application Module

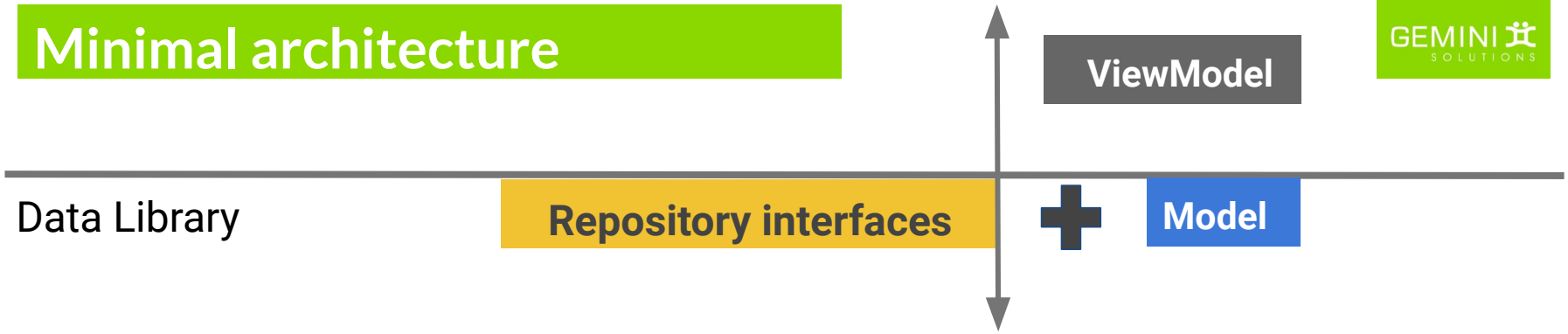
MVVM pattern
Presentation layer

Dependency

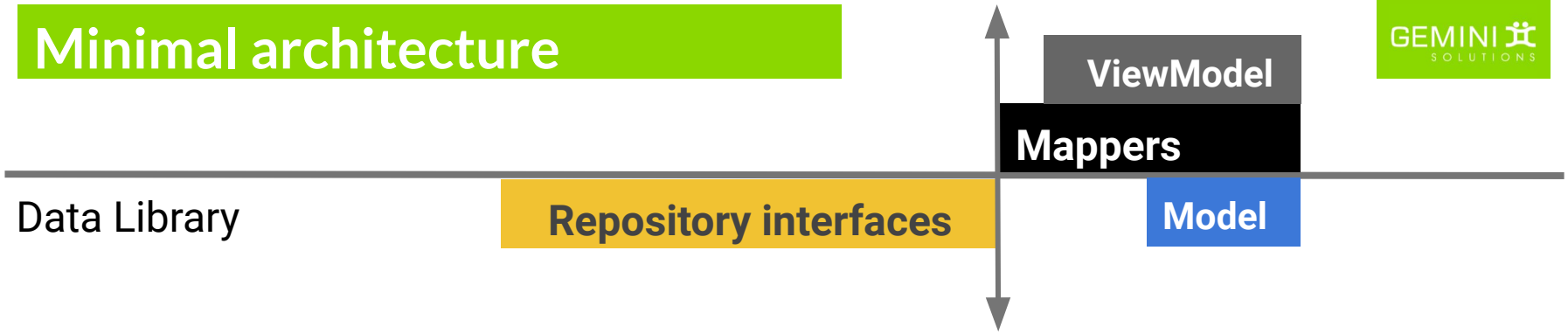
Android Library Module

Repository pattern
Data layer

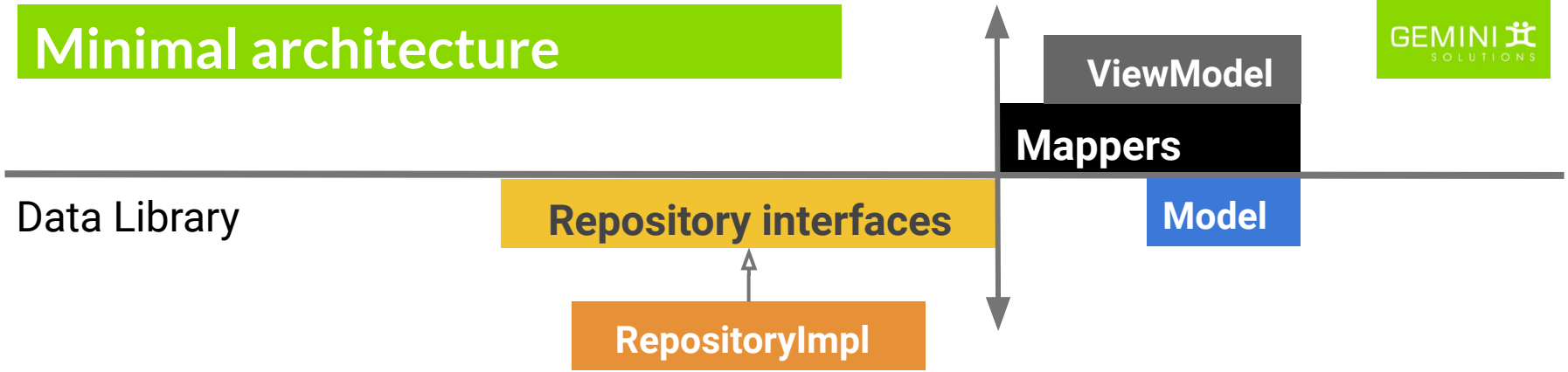
Minimal architecture



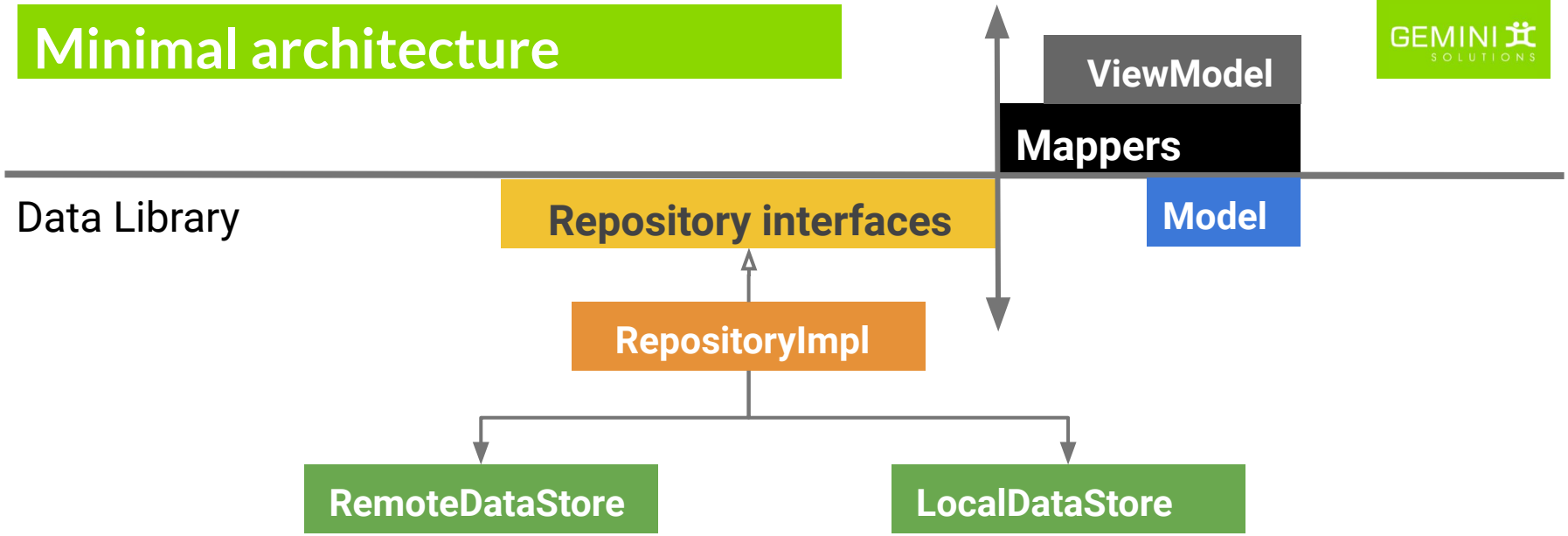
Minimal architecture



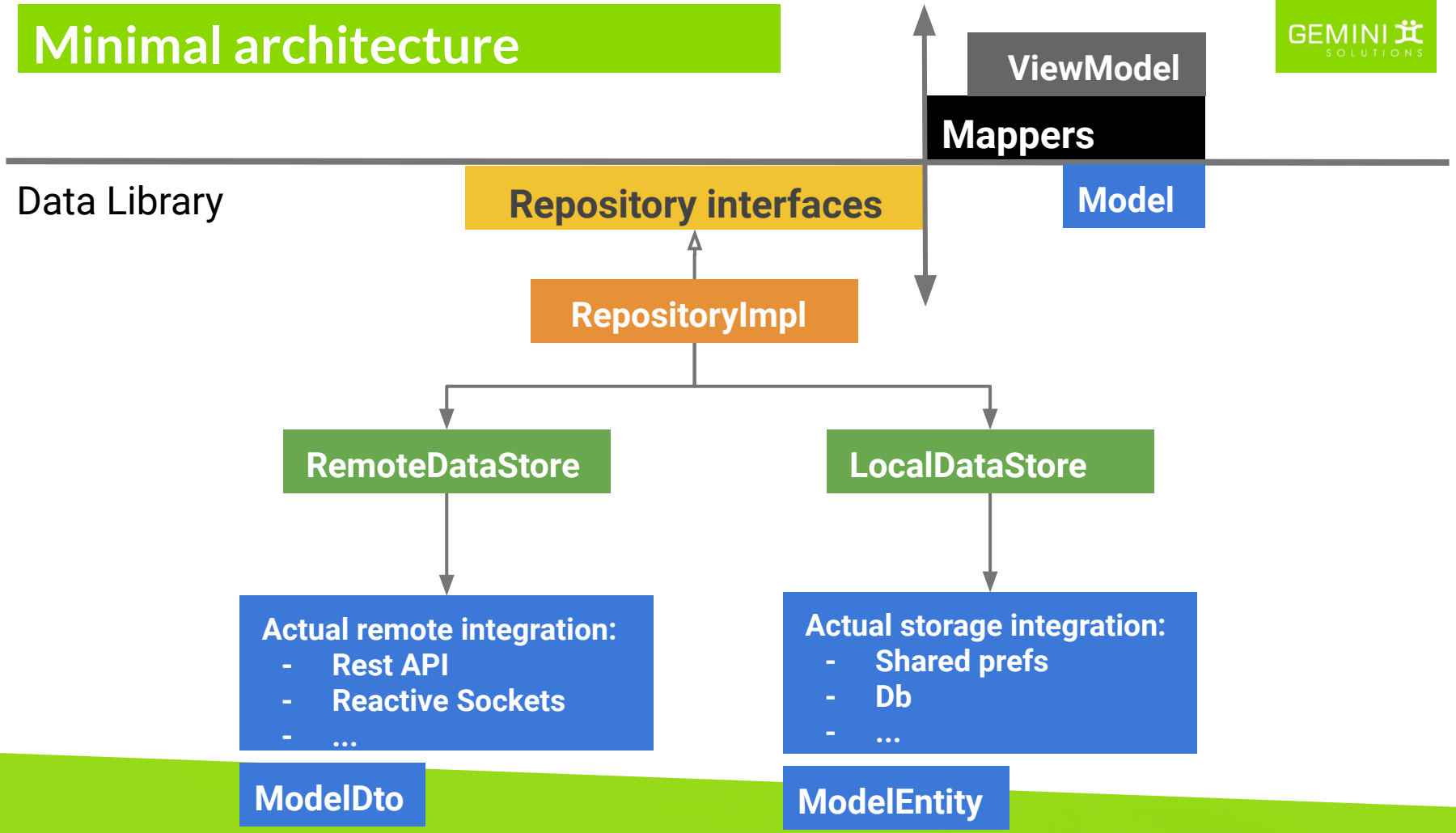
Minimal architecture



Minimal architecture



Minimal architecture



Thanks

GEMINI SOLUTIONS