C🍪🍪KIE-APP

Convergent Capstone Design 1
18th Week Progress Report

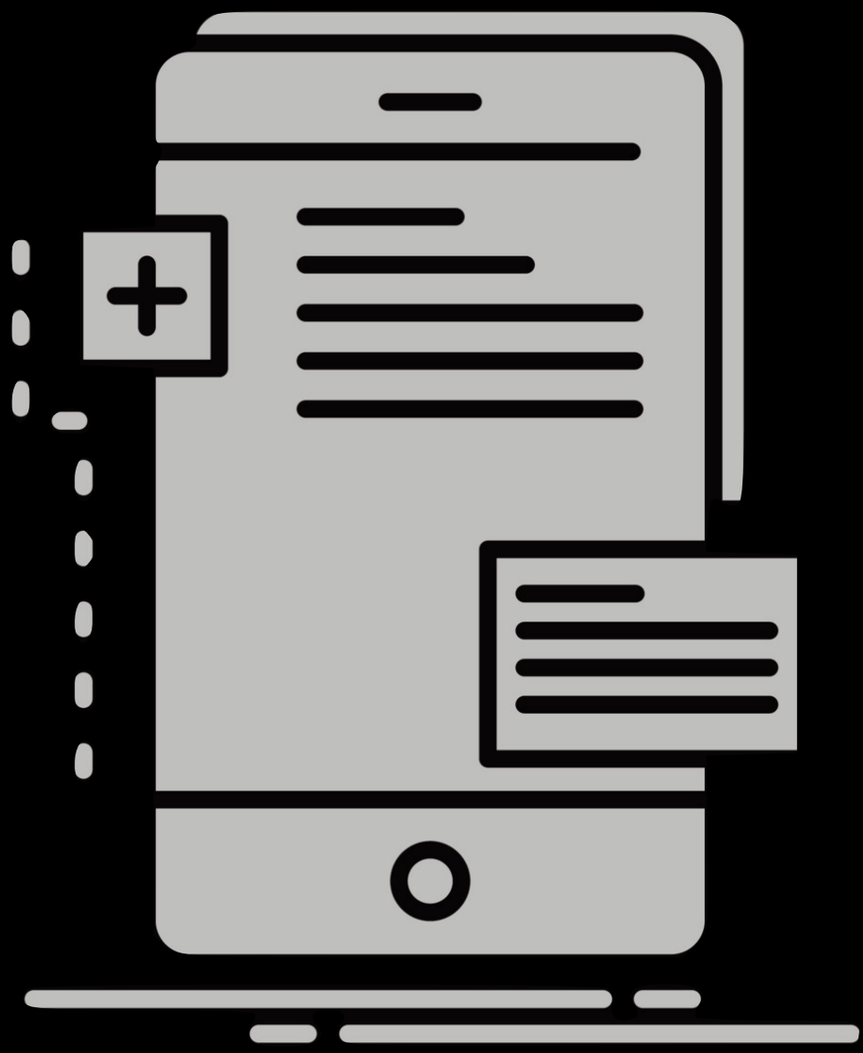
Group 5 Park JongBeum & Baek SeungHeon, June 13rd 2023

Recall

Recall

What we did last week

- Maps tab & UI
- State Management Using Provider
- async/await
- Backend TypeScript Migration

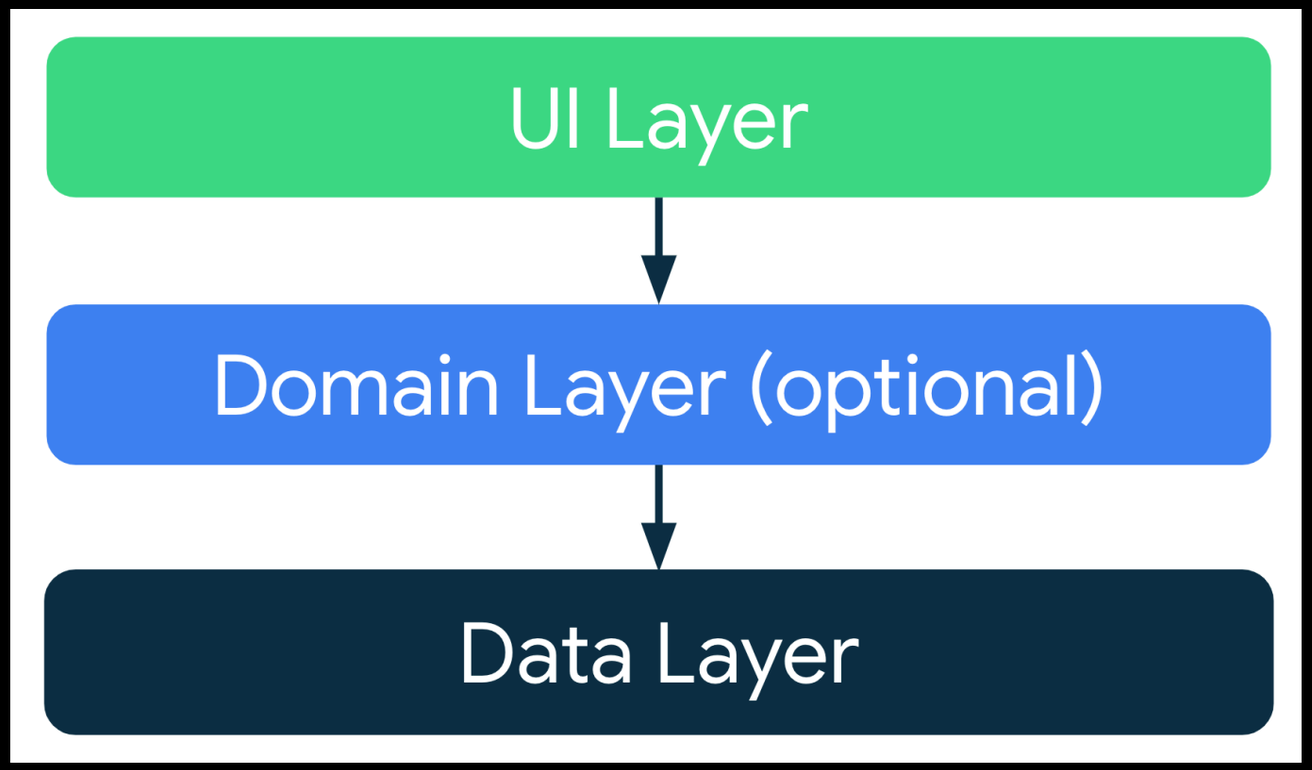


Architecture Pattern

What is Architecture Pattern?

- A structural approach to solving a particular problem
    - Improved Code Maintenance
    - Increased Reusability
    - Clarity of Code Structure
    - Flexibility and Scalability

Guide to app architecture (Android)



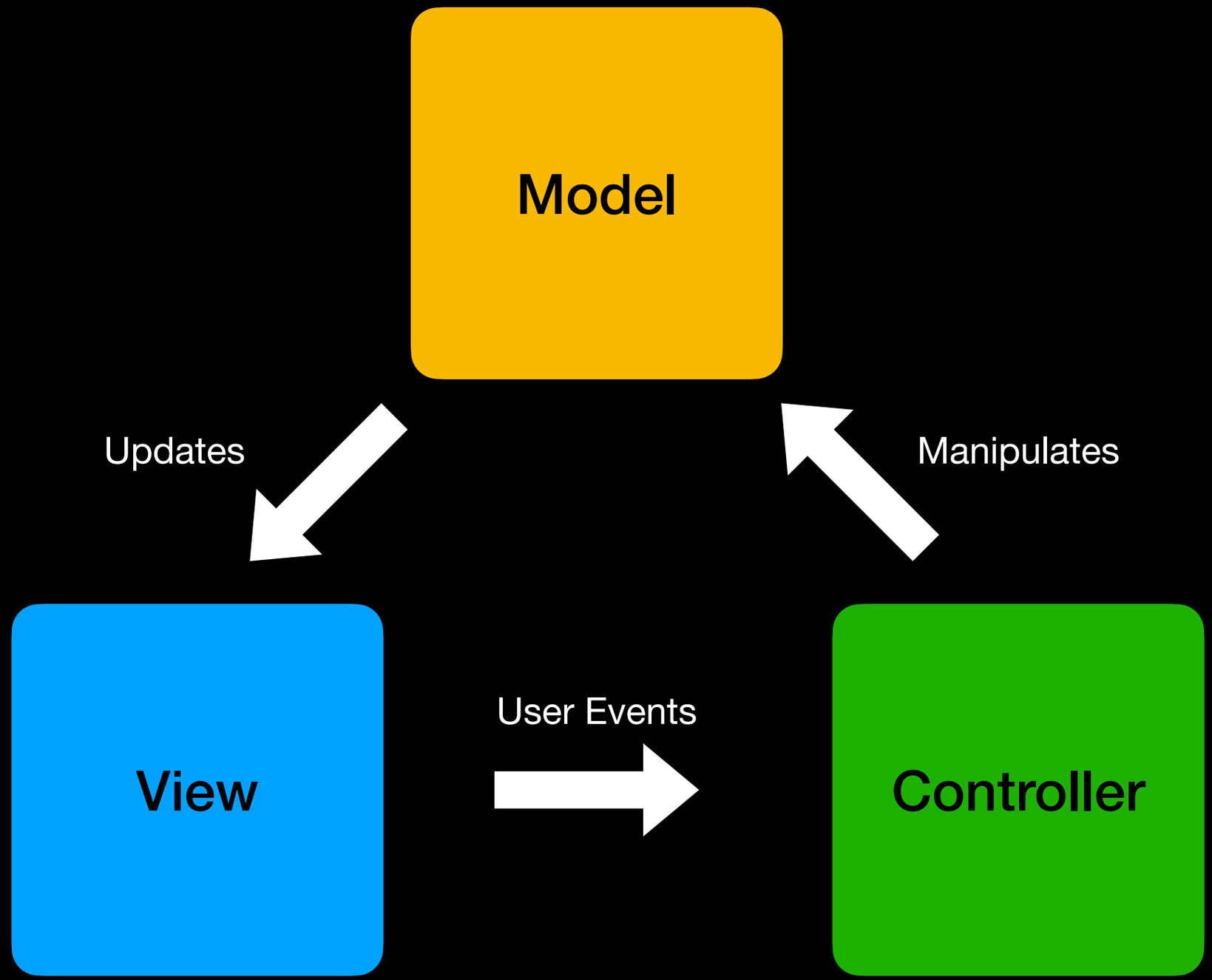
- General best practices
  - Don't store data in app components
  - Create well-defined boundaries of responsibility between various modules in your app

https://developer.android.com/topic/architecture?hl=ko#architecture-benefits

MVC

Model View Controller

Basic Structure



Model View Controller

Pros and Cons

✓ Multiple Components Creation Feasibility

✓ Less Complicated

x Challenging For Modern UI

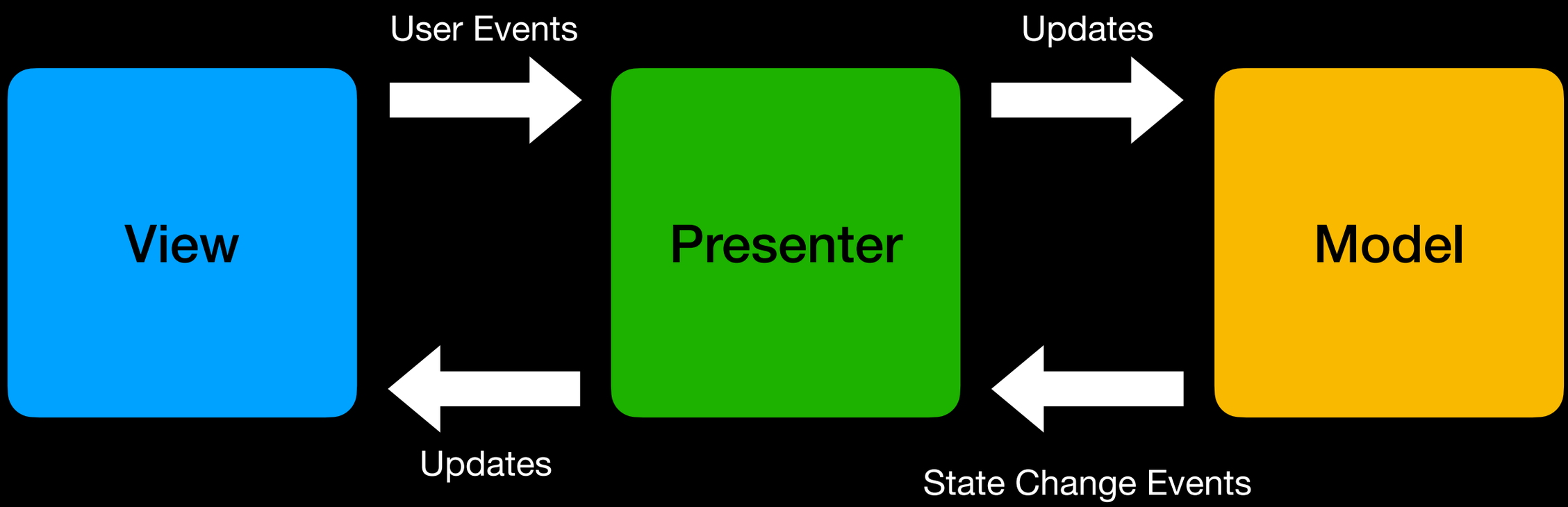
x Hard To Reuse and Run Tests

x View - Model are tightly coupled

MVP

Model View Presenter

Basic Structure



Model View Presenter

Pros and Cons

✓ Easier Debugging

✓ Better Reusability

✓ Reusable Components

x Because of the Lack of Controller, Control flow has also to be handled by the presenter; Responsible for 2 concerns:

  x Updating the Model

  x Presenting the Model

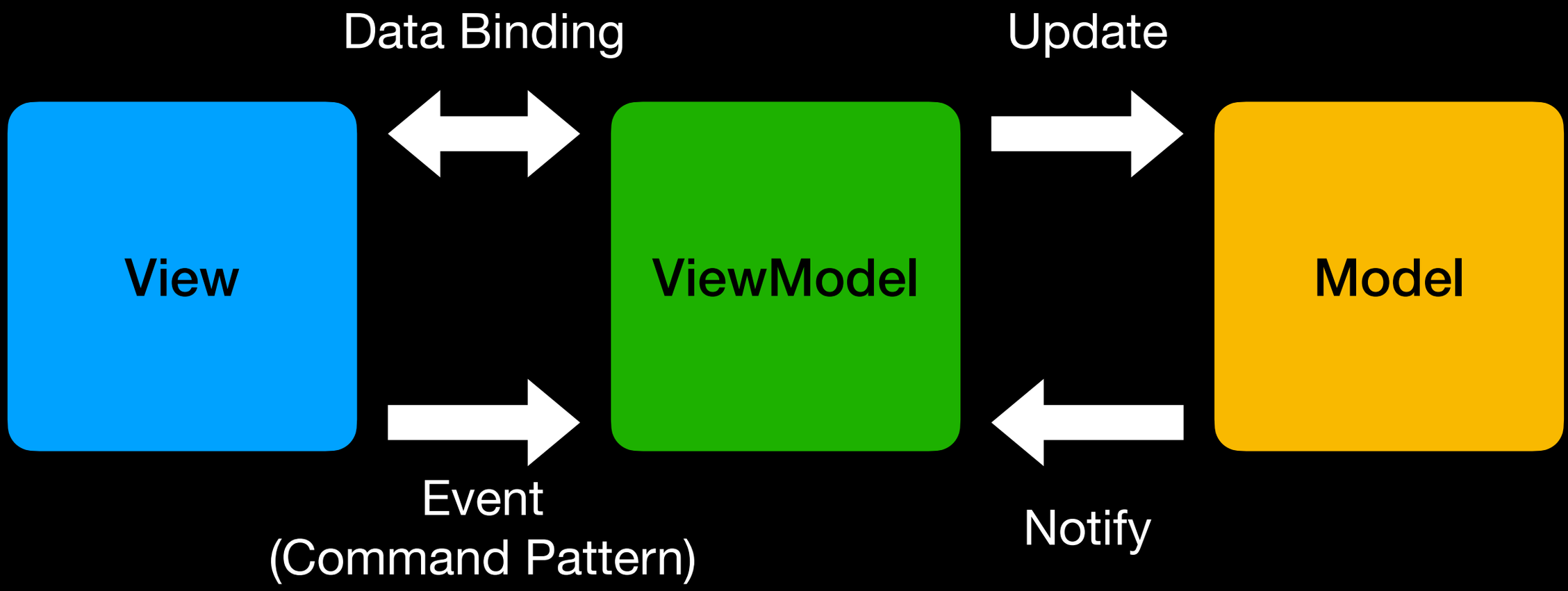
x Cannot Utilize Data Binding

https://www.baeldung.com/mvc-vs-mvp-pattern

MVVM

Model View ViewModel

Basic Structure



Example
MVVM

나이: 3살

체력: 1800

배고픔: 100

산책하기

View

```
DogView {
  build() {
    Button: onPressed => takeWalk();
    Text: “나이: ${
            Consume<DogViewModel>
              .read()
              .나이}”
    Text: “체력:
  }
}
```

ViewModel

```
DogViewModel {
  DogModel model = DogModel();
  나이 = “${Date.now() - model.생일} 살”
  get health: model.health
  get hunger: model.hunger
  takewalk() {
    increaseHunger(10);
    decreaseHealth(20);
  }
}
```

Model

```
DogModel {
  이름: 코코,
  생일: 2020/6/13,
  health: 100,
  hunger: 0,
  increaseHunger()
  decreaseHeatlh()
}
```

Model View ViewModel

Pros and Cons

✓ Faster Screen Loading

✓ Improved Performance

✓ Reusable Components

✓ Testability

x Doesn’t Offer Tight Coupling between ViewModel and Model

x Overkilling for simple UI operation

x Higher Learning Curve

Repository Pattern

```dart
import 'package:cookie_app/repository/api/account.dart';
import 'package:cookie_app/repository/storage/account.storage.dart';
import 'package:cookie_app/types/account/account_info.dart';

class MyInfoRepository {
  final MyInfoRepositoryPattern _repositoryStorage = MyInfoRepositoryStorageImpl();
  final MyInfoRepositoryPattern _repositoryAPI = MyInfoRepositoryApiImpl();

  MyInfoRepository();

  Future<PrivateAccount> getInfo() async {
    final _repository =
        await AccountStorage().isExist() ? _repositoryStorage : _repositoryAPI;
    return _repository.getInfo();
  }
}

abstract class MyInfoRepositoryPattern {
  Future<PrivateAccount> getInfo();
}

class MyInfoRepositoryStorageImpl implements MyInfoRepositoryPattern {
  @override
  Future<PrivateAccount> getInfo() async {
    return PrivateAccount.fromJson(await AccountStorage().readJSON());
  }
}

class MyInfoRepositoryApiImpl implements MyInfoRepositoryPattern {
  @override
  Future<PrivateAccount> getInfo() async {
    return (await AccountAPI.getInfo()).toPrivateAccount();
  }
}

```

DataBinding

Example with Basic Counter

```dart
import 'package:flutter/material.dart';

class Counts with ChangeNotifier {
  int _count = 0;
  int get count => _count;

  void increment() {
    _count++;
    notifyListeners();
  }

  void decrement() {
    _count--;
    notifyListeners();
}

```

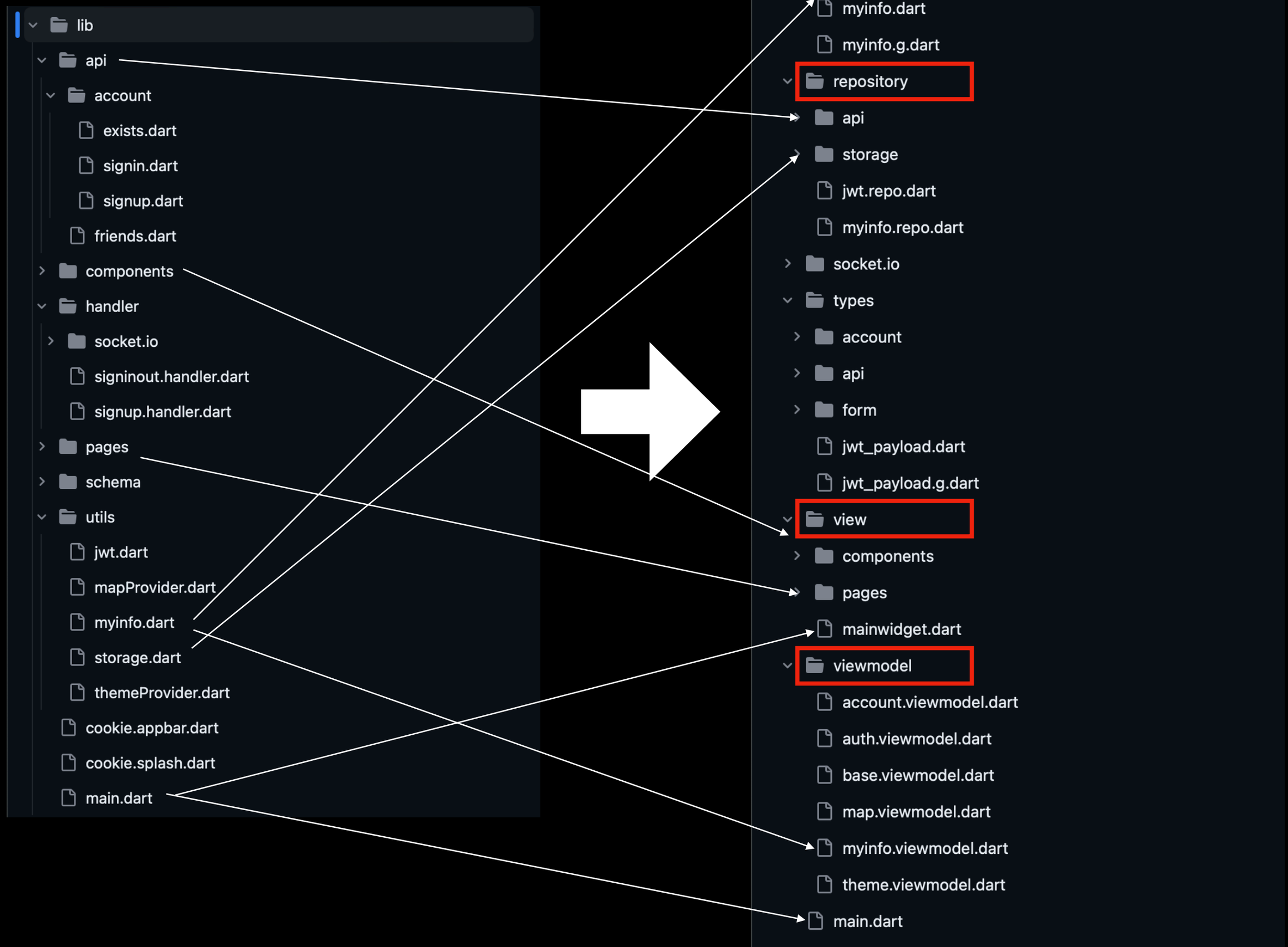
```dart
class IncrementButton extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
        onPressed: () {
          // Provider.of<Counts>(context, listen: false).increment();
          context.read<Counts>().increment();
        },
        child: const Icon(Icons.add));
  }
}

class DecrementButton extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
        onPressed: () {
          // Provider.of<Counts>(context, listen: false).decrement();
          context.read<Counts>().decrement();
        },
        child: const Icon(Icons.remove));
  }
}
```

```dart
import 'package:flutter/material.dart';
import 'package:flutter_provider_example/providers/counts.dart';
import 'package:provider/provider.dart';

class Counter extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Text(
      // Provider.of<Counts>(context).count.toString(),
      context.watch<Counts>().count.toString(),
      style: const TextStyle(fontSize: 20),
    );
  }
}
```

File Structure


lib
api
account
exists.dart
signin.dart
signup.dart
friends.dart
components
handler
socket.io
signinout.handler.dart
signup.handler.dart
pages
schema
utils
jwt.dart
mapProvider.dart
myinfo.dart
storage.dart
themeProvider.dart
cookie.appbar.dart
cookie.splash.dart
main.dart
lib
model
jwt.dart
myinfo.dart
myinfo.g.dart
repository
api
storage
jwt.repo.dart
myinfo.repo.dart
socket.io
types
account
api
form
jwt_payload.dart
jwt_payload.g.dart
view
components
pages
mainwidget.dart
viewmodel
account.viewmodel.dart
auth.viewmodel.dart
base.viewmodel.dart
map.viewmodel.dart
myinfo.viewmodel.dart
theme.viewmodel.dart
main.dart

Thank You

Questions are welcome