

One Size Doesn't Fit All

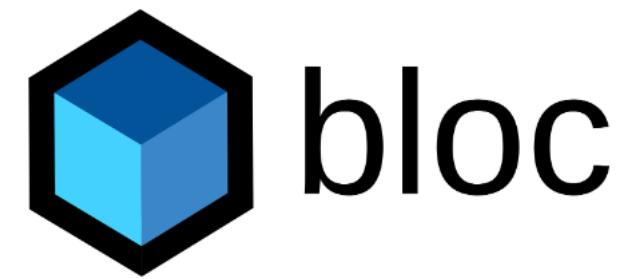
Tailoring Flutter Architectures for Team and Business Success



Jan Stepien

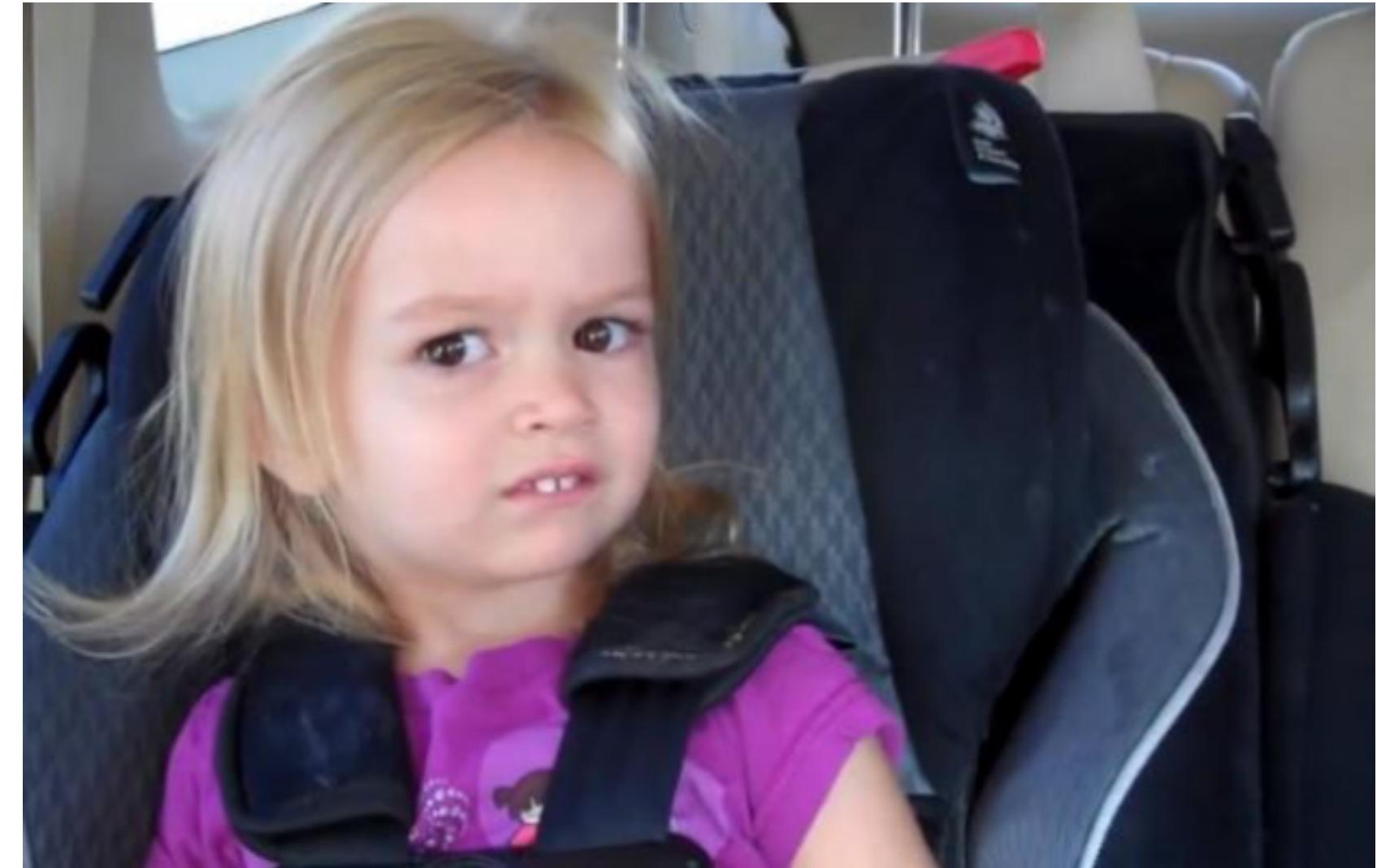
Goals of this talk

- `Bloc or Riverpod type of guy`



Goals of this talk

- `Bloc or Riverpod type of guy`
- `What is my team doing?!`



Goals of this talk

- `Bloc or Riverpod type of guy`
- `What is my team doing?!`
- `I know best why they don't listen`



Goals of this talk

- `Bloc or Riverpod type of guy`
- `What is my team doing?!`
- `I know best why they don't listen`
- `That seems interesting...`

Goals of this talk

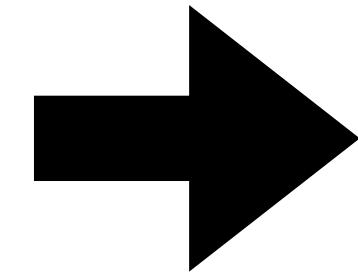
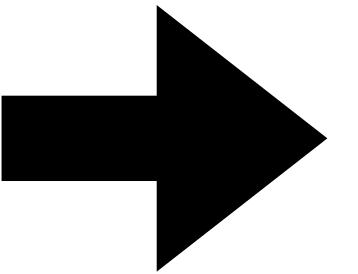
- `Bloc or Riverpod type of guy`
- `What is my team doing?!`
- `I know best why they don't listen`
- `That seems interesting...`



Make you think

How I'm here?

Consulting
&
Corporate

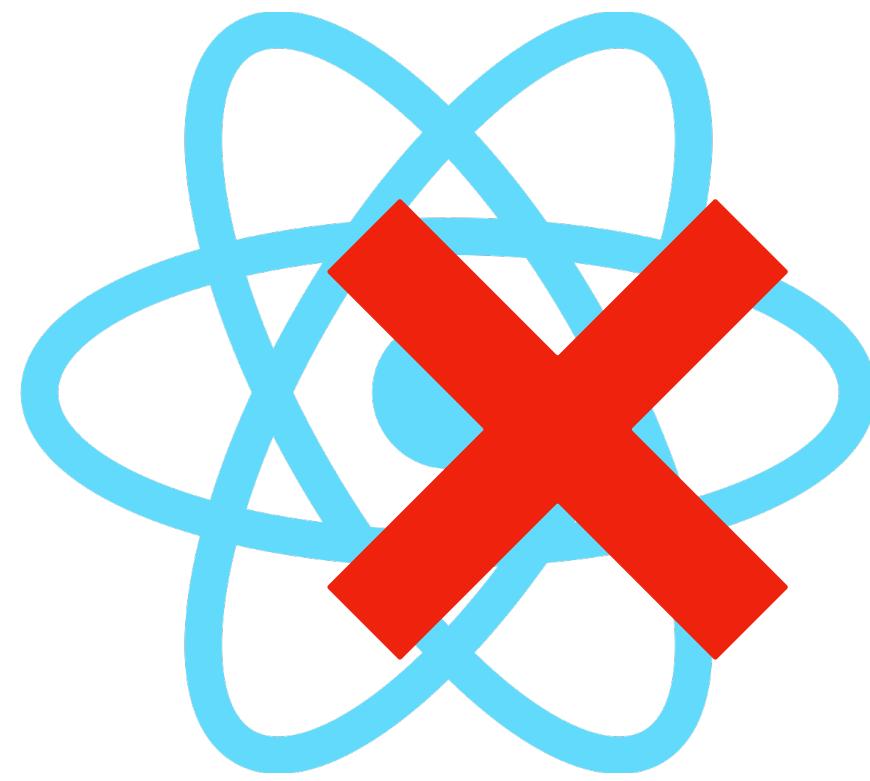


< 10 people startup
&
20k + active users
app

What should I do?

Set the priorities

```
final variable1 = 'Hello world';
```



Set the priorities

Fast, Cheap, and Good... pick two

Fast

Cheap

Good

&

Scalable



Not important

Most important

Set the priorities

Fast, Cheap, and Good... pick two

Cheap

Scalable

Good

Fast



Not important

Most important



Developer Experience

Preparation

Preparation

HEALTHY CODE: A GUIDE TO FLUTTER APP AUDIT



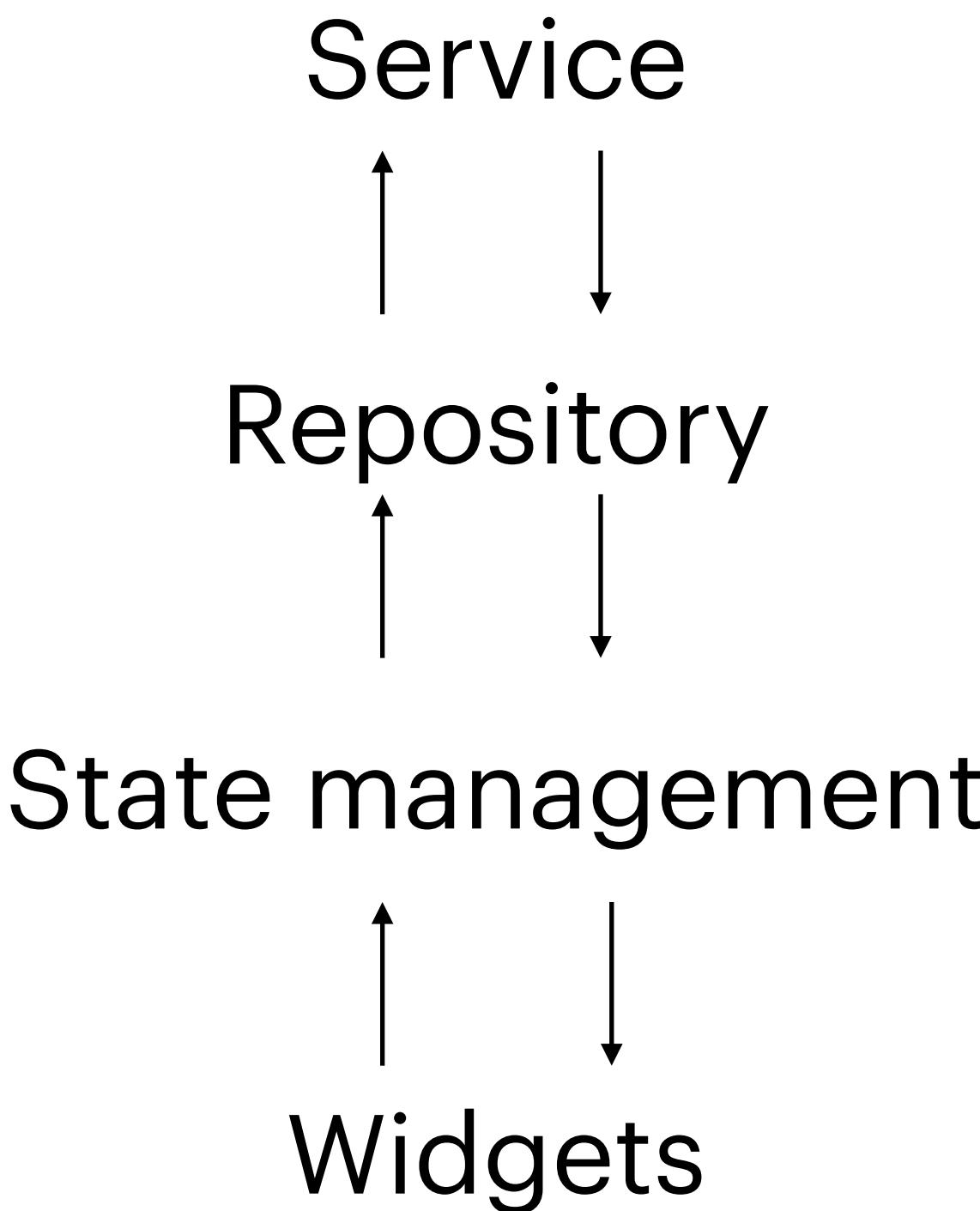
DARIA ORLOVA

Flutter developer and mobile tech lead

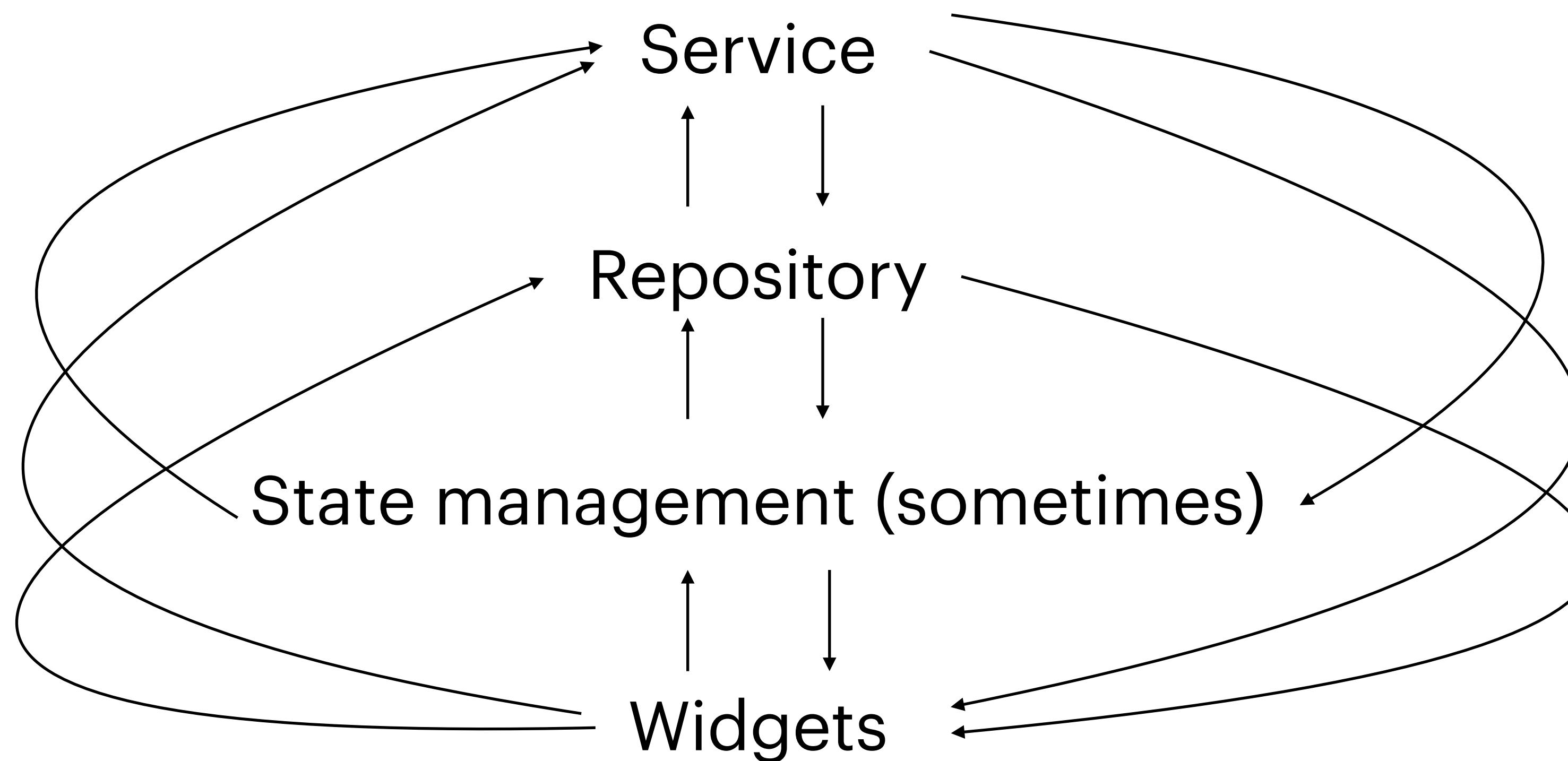
Chili Labs.

<https://www.droidcon.com/2023/08/07/healthy-code-a-guide-to-flutter-app-audit/>

Architectural layers



Architectural layers



Preparation

- **Cleanup project file structure**

Preparation

- **Cleanup project file structure**

- Align file naming strategy

Preparation

- **Cleanup project file structure**

- Align file naming strategy

- Align directory structure

Preparation

- **Cleanup project file structure**

- Align file naming strategy

- Align directory structure

- Use barrel files or not

Preparation

- **Cleanup project file structure**

- Align file naming strategy

- Align directory structure

- Use barrel files or not

- Widget per file or not

- You name it...



Preparation

- Cleanup project file structure

Align file naming strategy



profile.dart

profile_model.dart

Preparation

- **Cleanup project file structure**

Align file naming strategy



profile.dart

profile_model.dart

Align directory structure



<feature>/page

<feature>/pages

Preparation

- **Cleanup project file structure**

Align file naming strategy		profile.dart	profile_model.dart
Align directory structure		<feature>/page	<feature>/pages
Use barrel files or not		Some	None

Preparation

- **Cleanup project file structure**

Align file naming strategy		profile.dart	profile_model.dart
Align directory structure		<feature>/page	<feature>/pages
Use barrel files or not		Some	None
Widget per file or not		No	Not fully

Preparation

- Cleanup project file structure

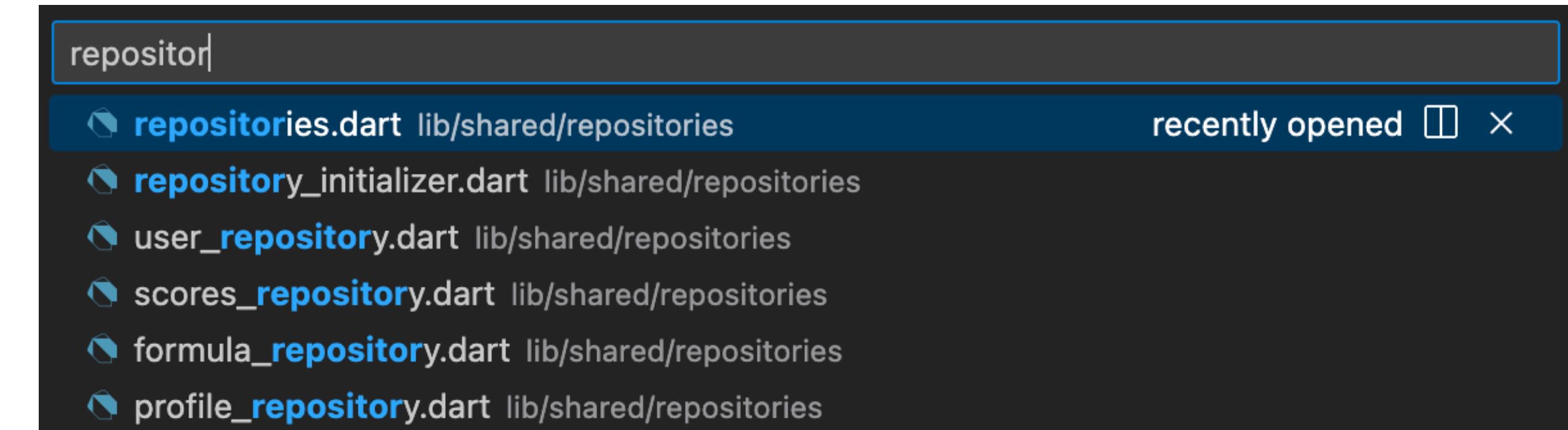
Align file naming strategy

Align directory structure

`cmd + p` or `ctrl + p`

Use barrel files or not

Widget per file or not



Predictable code speeds up development



Preparation

- pubspec.yaml dependency check

```
! pubspec.yaml M X  
!  
! pubspec.yaml  
9   dependencies:  
10  |   image_picker: ^1.0.7  
11  |   images_picker: ^1.2.11
```

Preparation

- **pubspec.yaml dependency check**

ChangeNotifier

Bloc & Cubit

MobX

Preparation

Done



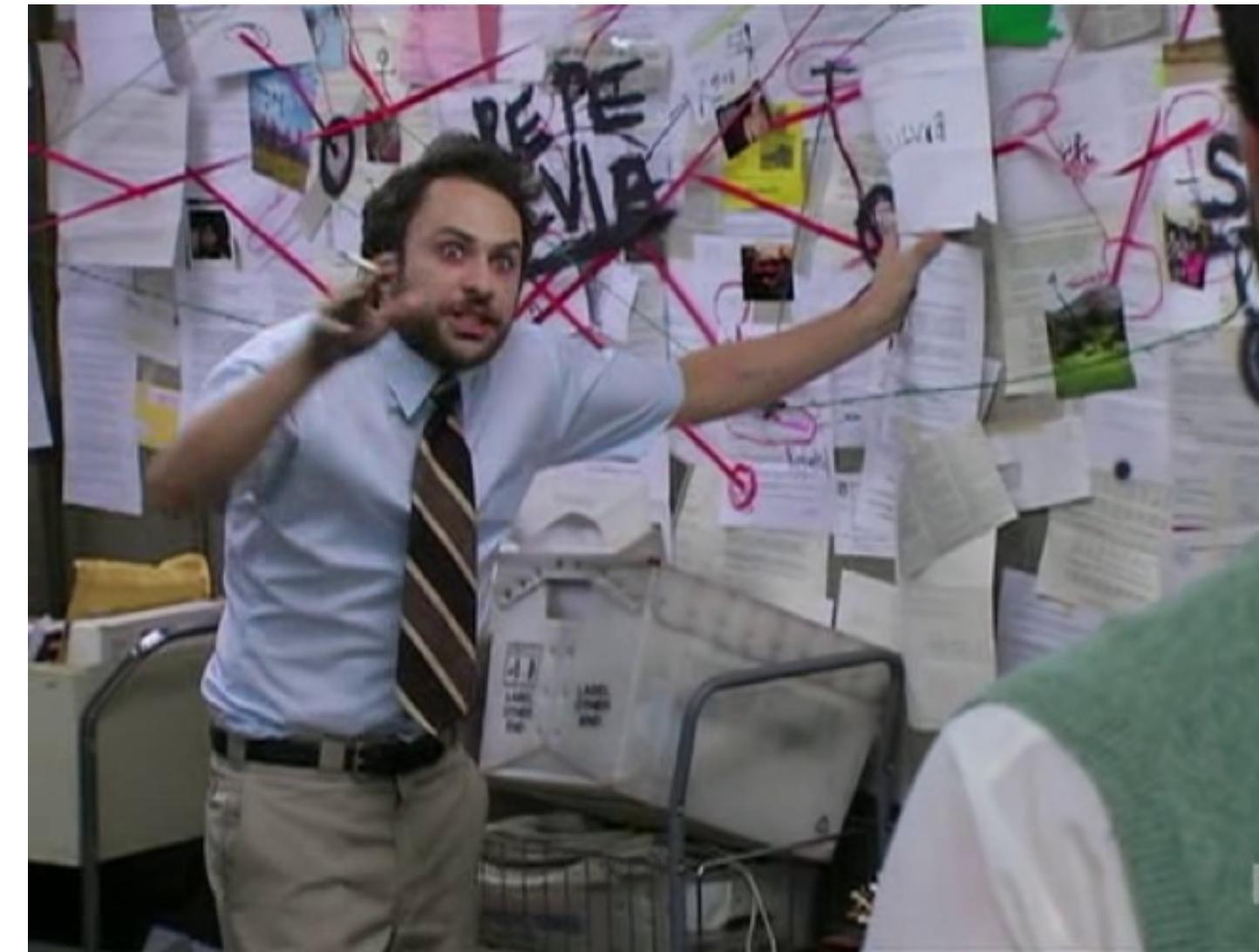
Planning the execution

- Define refactor goal

Planning the execution

- Define refactor goal

Plan before you start coding



Planning the execution

- **Define refactor goal**

Correct split of layers + single state management

Planning the execution

- **Define refactor goal**
Correct split of layers + single state management
- **Definition of done**

Planning the execution

- **Define refactor goal**
Correct split of layers + single state management
- **Definition of done**
DI used across all layers + no flutter_bloc + no ChangeNotifier

Planning the execution

- **Define refactor goal**
Correct split of layers + single state management
- **Definition of done**
DI used across all layers + no flutter_bloc + no ChangeNotifier
- **Split the task**

Planning the execution

- **Define refactor goal**

Correct split of layers + single state management

- **Definition of done**

DI used across all layers + no flutter_bloc + no ChangeNotifier

- **Split the task**

Task each: misused DI, bloc rewrite, per each changeNotifier state

Planning the execution

- **Define refactor goal**

Correct split of layers + single state management

- **Definition of done**

DI used across all layers + no flutter_bloc + no ChangeNotifier

- **Split the task**

Task each: misused DI, bloc rewrite, per each changeNotifier state

- **Communicate**

Keep the tasks status up to date

Just do it

Why we hate MobX and yet we still use it

Planning the execution

- Reason for change

Planning the execution

- Reason for change

We don't like **build_runner** in state management, the developer experience is not great

Planning the execution

- Reason for change

We don't like **build_runner** in state management, the developer experience is not great

Optimise **build_runner**?

Planning the execution

- Reason for change

We don't like **build_runner** in state management, the developer experience is not great

Optimise **build_runner**?

From 2min to <30s on M1 chip

```
! build.yaml ×  
! build.yaml  
1 targets:  
2   $default:  
3     builders:  
4       mobx_codegen:mobx_generator:  
5         generate_for:  
6           - lib/**store.dart  
7     freezed:  
8       generate_for:  
9         - lib/**model.dart  
10    json_serializable:json_serializable:  
11      options:  
12        explicit_to_json: true  
13      generate_for:  
14        - lib/**freezed.dart  
15        - lib/**model.dart  
16        - lib/**store.dart  
17        - test/**context.dart  
18
```

Planning the execution

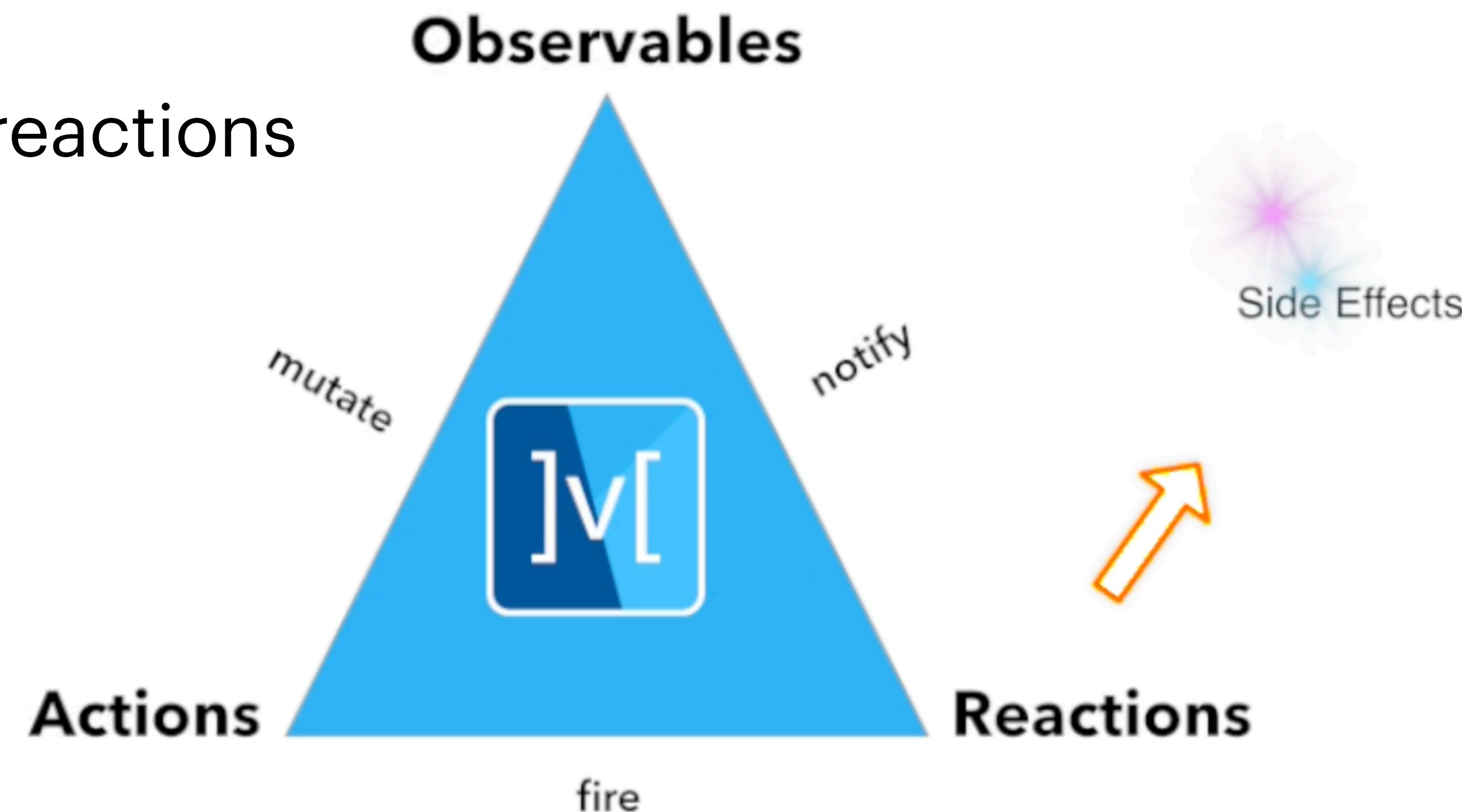
- **Reason for change**

Mobx store does not auto dispose its reactions
(devs are forgetful)

Planning the execution

- **Reason for change**

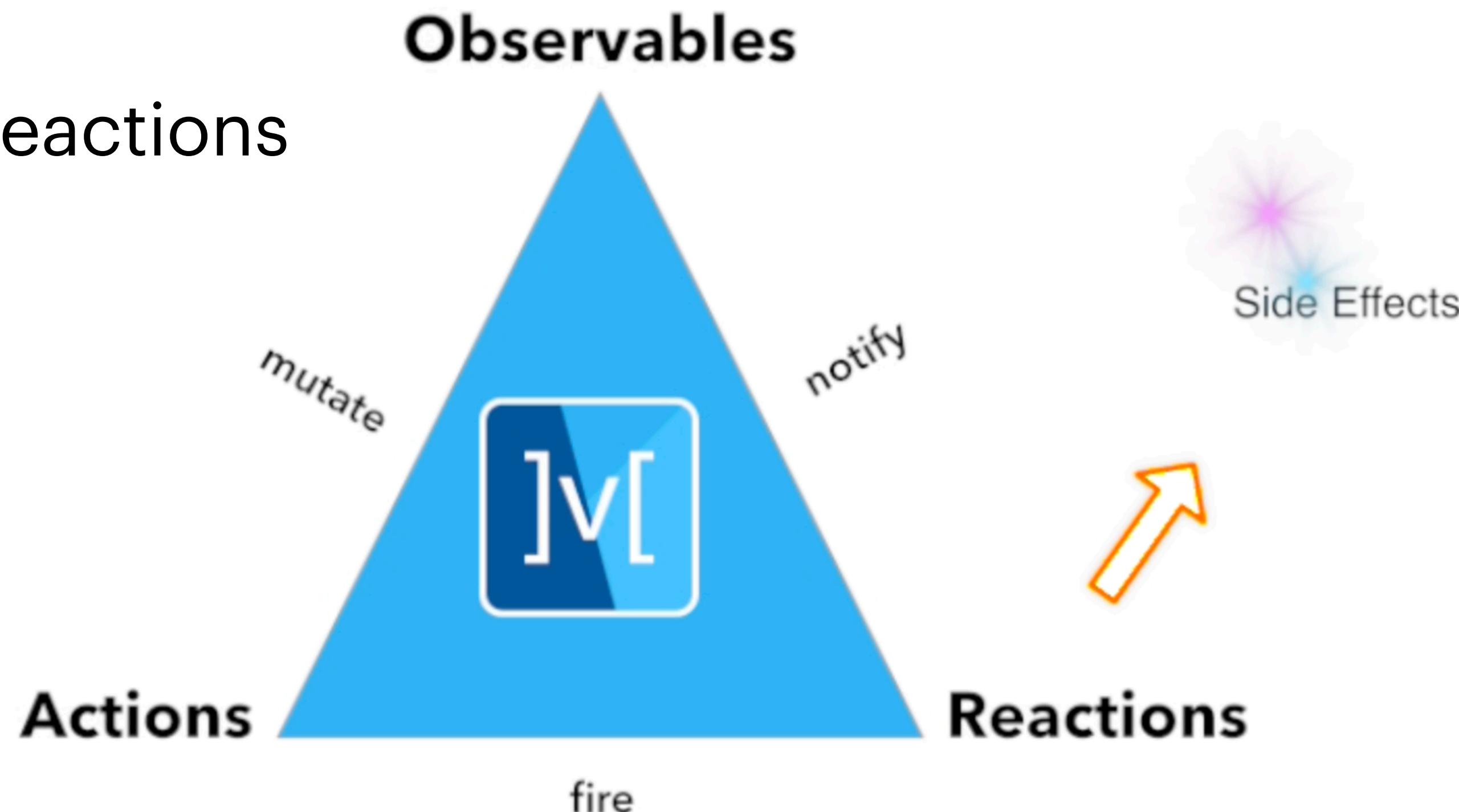
Mobx store does not auto dispose its reactions
(devs are forgetful)



Planning the execution

- **Reason for change**

Mobx store does not auto dispose its reactions
(devs are forgetful)



```
ReactionDisposer autorun(Function(Reaction) fn)
```

Runs the reaction immediately and also on any change in the observables used inside `fn`.

Planning the execution

- Reason for change

Mobx store does not auto dispose its reactions
(devs are forgetful)

```
class StoreAndReactionDisposersWrapper extends StatelessWidget {  
  const StoreAndReactionDisposersWrapper({  
    required this.providers,  
    required this.reactionDisposersBuilder,  
    required this.builder,  
    this.onInit,  
    this.afterInit,  
    this.onDispose,  
    super.key,  
  });
```

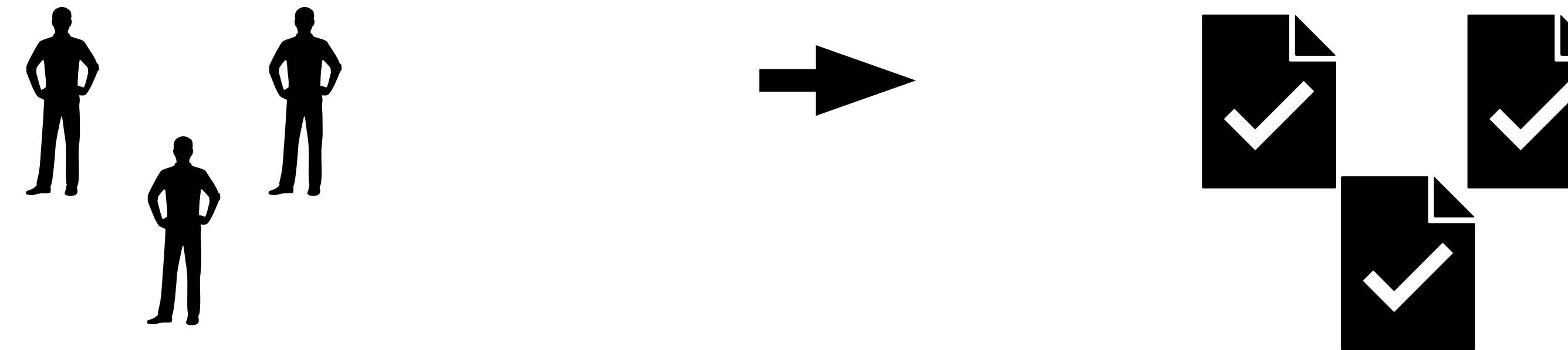
Planning the execution

- Reason for change

Maybe its fine?



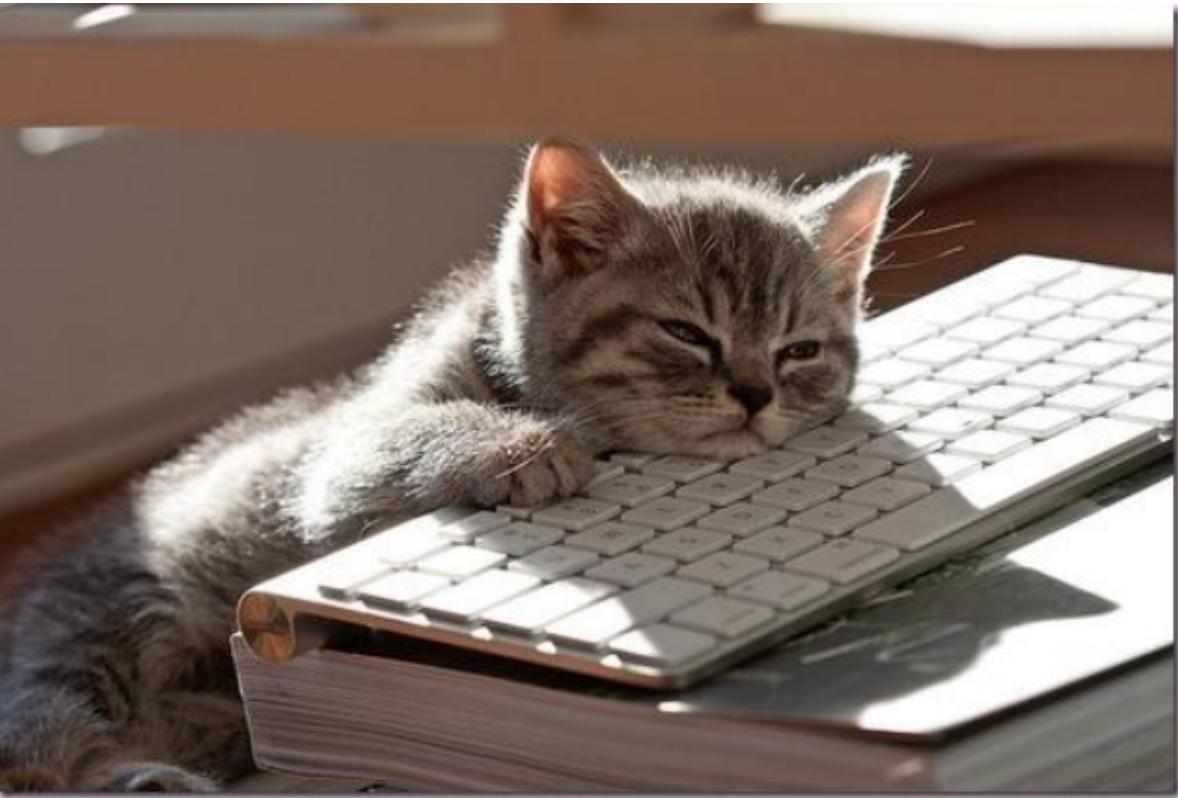
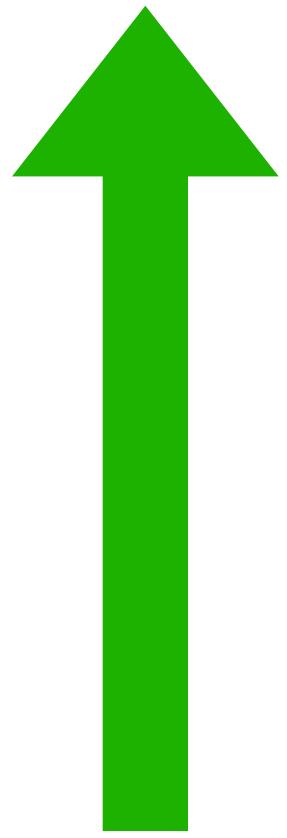
Development alignment



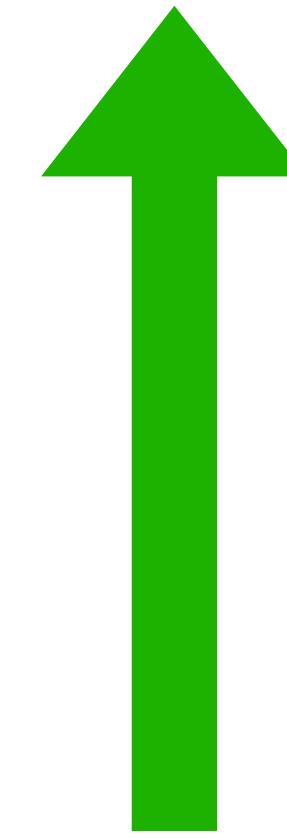
Predictable code speeds up development



Developer alignment



Boring



Speed

Developer alignment

- README file
 - Dev agreement on development structure

The screenshot shows a code editor window with a dark theme. At the top, there's a toolbar with icons for file operations like save, undo, redo, and search. Below the toolbar, the file path is shown as "README.md" with a "M" icon and an "X" icon. The main area contains the following text:

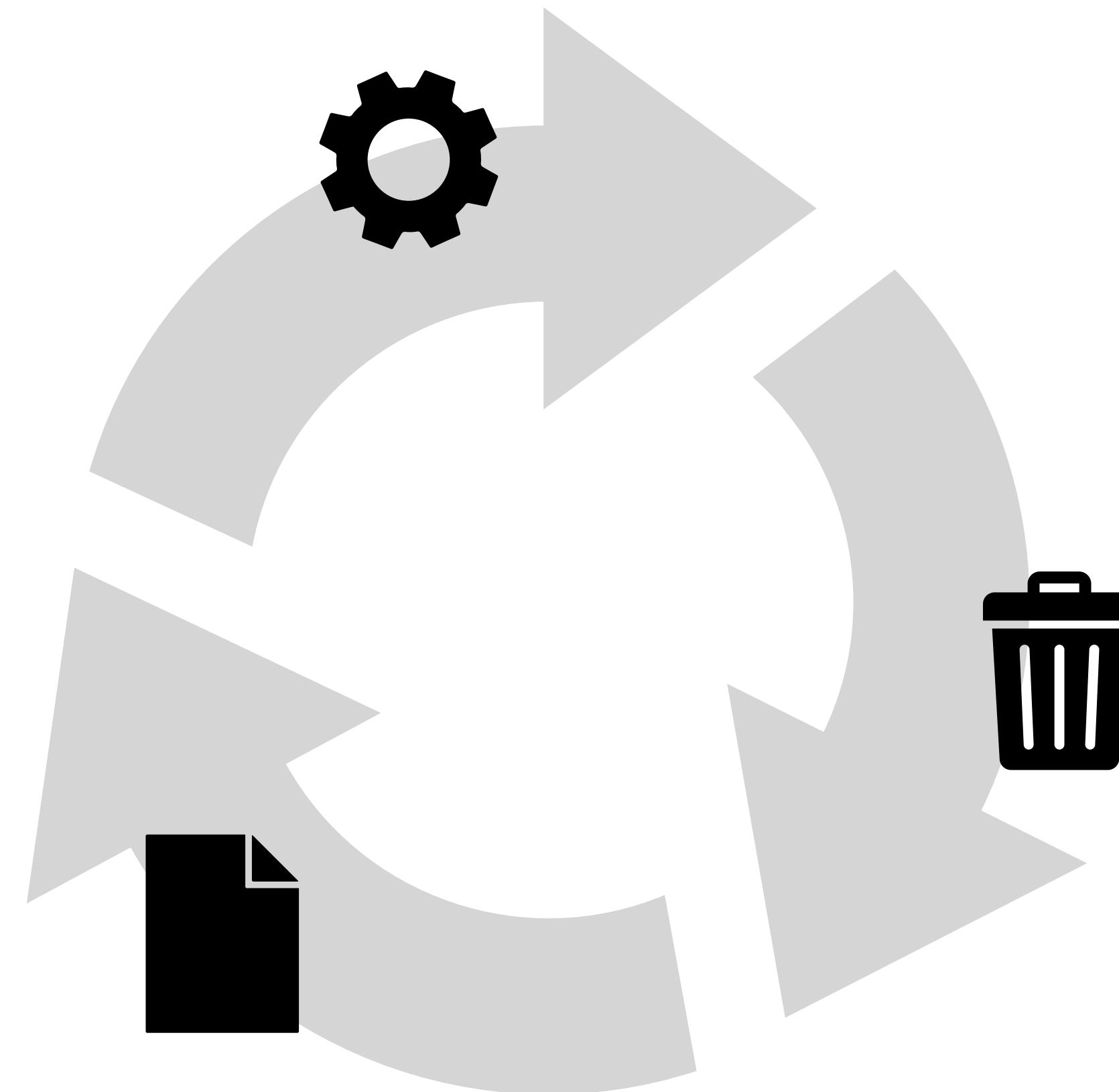
```
① README.md M X
① README.md > abc # Contribution agreement
60 # Contribution agreement
62 This agreement is a document created on mutual agreement from all the developers contributing to this project.
Every team member can change this set of rules after discussing and approval of the other contributing
developers.
63
64 I swear to do my best to develop project according to the following agreement:
65 @jan-stepien, @someoneelse, @thirdfromthepack
66
```

Developer alignment

- **README file**
 - Dev agreement on development structure
- **Dev-only weekly meeting**
 - 5min - 2 hour meeting every single week in calendar



We stopped writing widget tests



We stopped writing widget tests



Set the priorities

Fast, Cheap, and Good... pick two



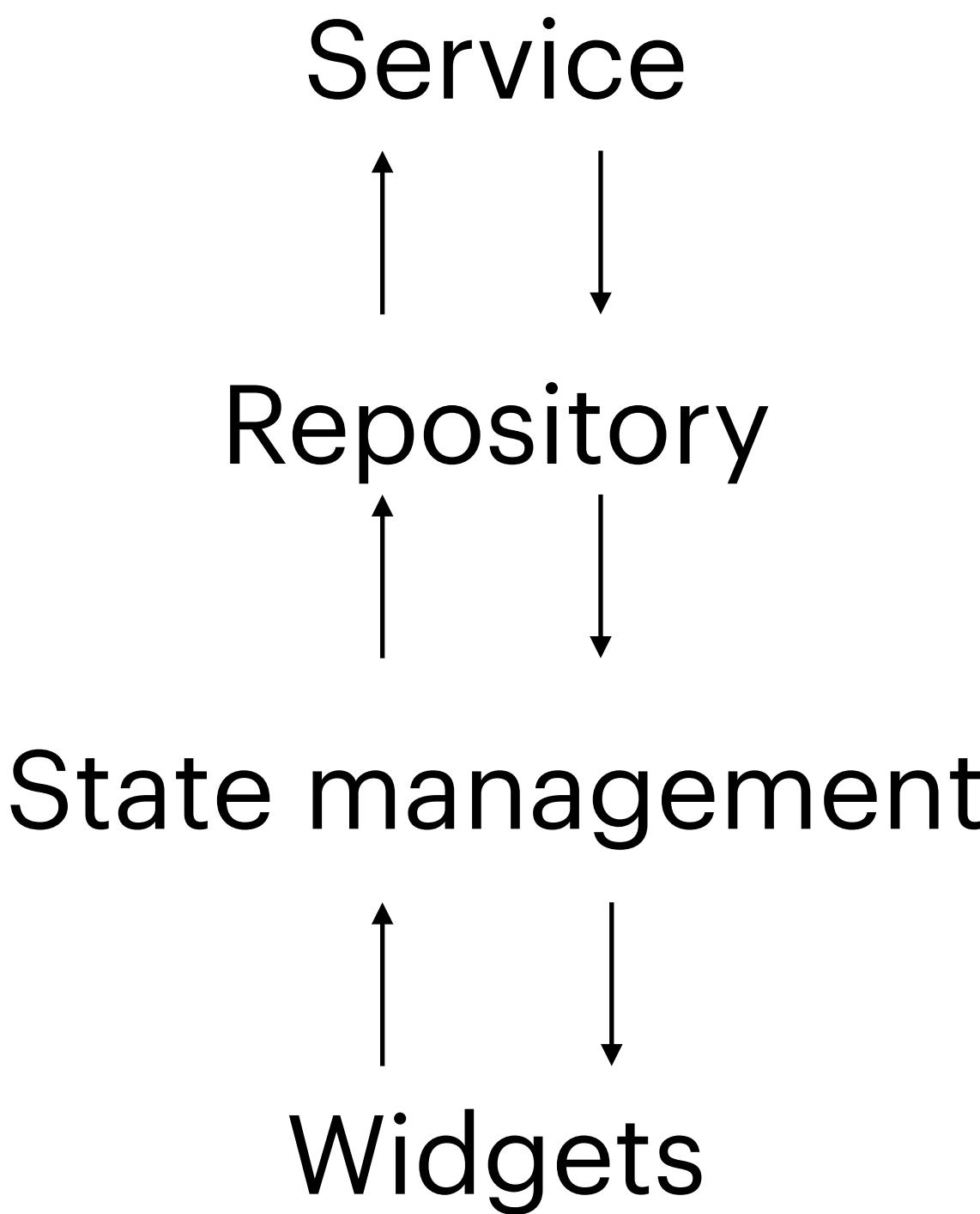
Fast

Not important

Most important

Creating software is a team effort

Architectural layers



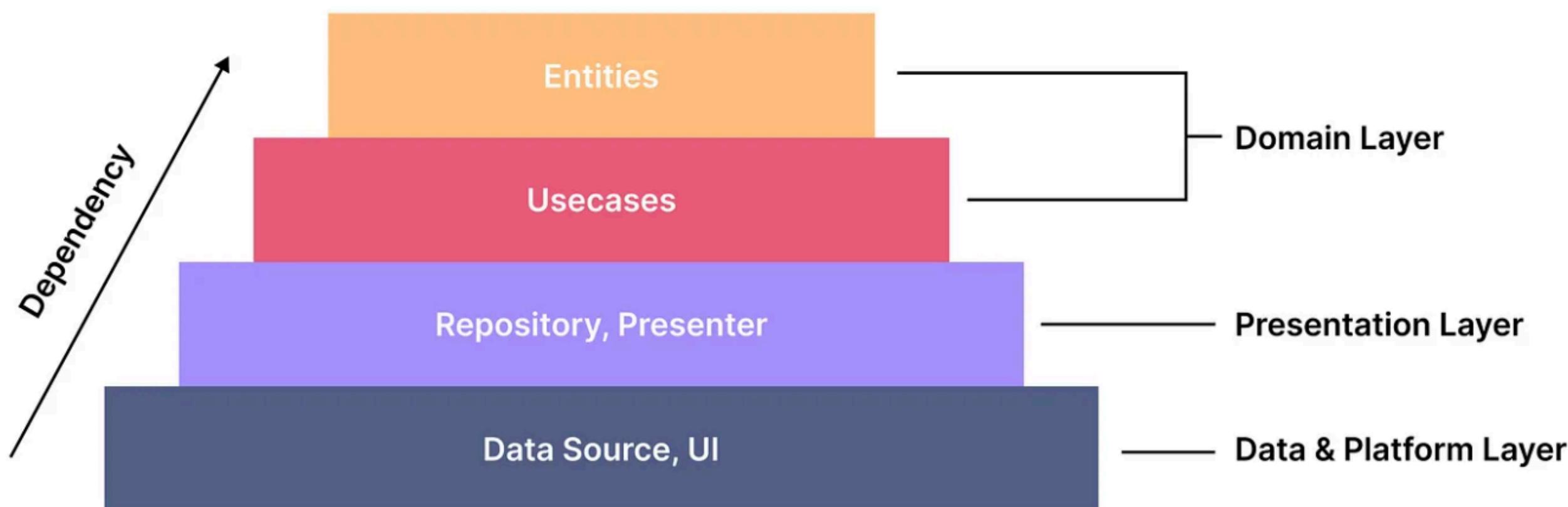
Creating software is a team effort

But it works only when the team shares same goal

The best architecture

- May exist

But its not universal for all projects



The best architecture

- **Is unstable**

Changes with your project needs

The best architecture

- Is unstable

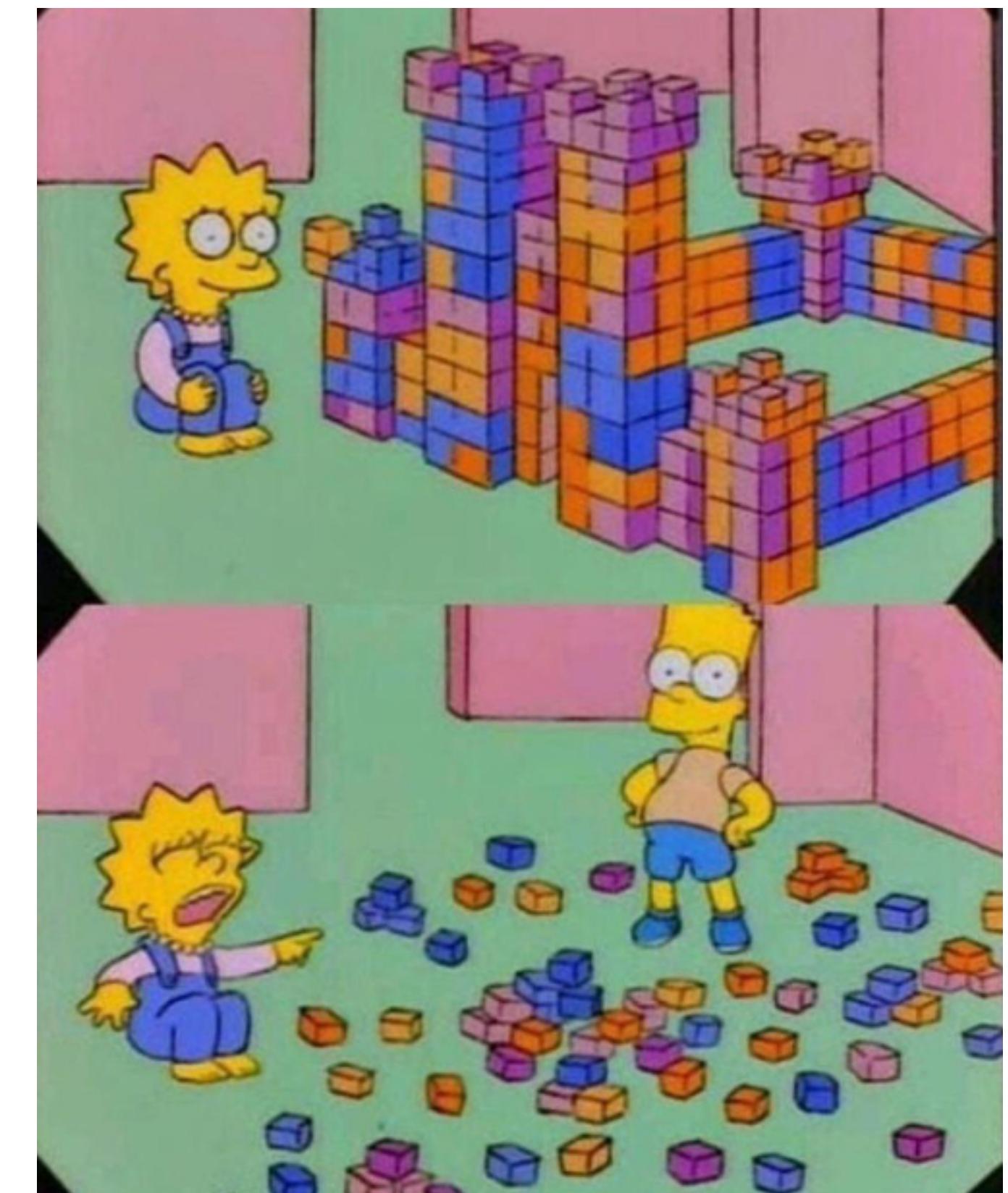
Changes with your project needs



The best architecture

- Is unstable

Changes with your project needs



Don't stay behind

Open up to unknown solutions

Talk to other devs

Be team player

Don't stay behind

Open up to unknown solutions

Talk to other devs

Be team player

Think

Thank you

Presentation



LinkedIn



<https://github.com/Jan-Stepien/fluttercon-architecture-presentation>

<https://www.linkedin.com/in/jan-stepien-developer/>

Q&A

Presentation



LinkedIn



<https://github.com/Jan-Stepien/fluttercon-architecture-presentation>

<https://www.linkedin.com/in/jan-stepien-developer/>