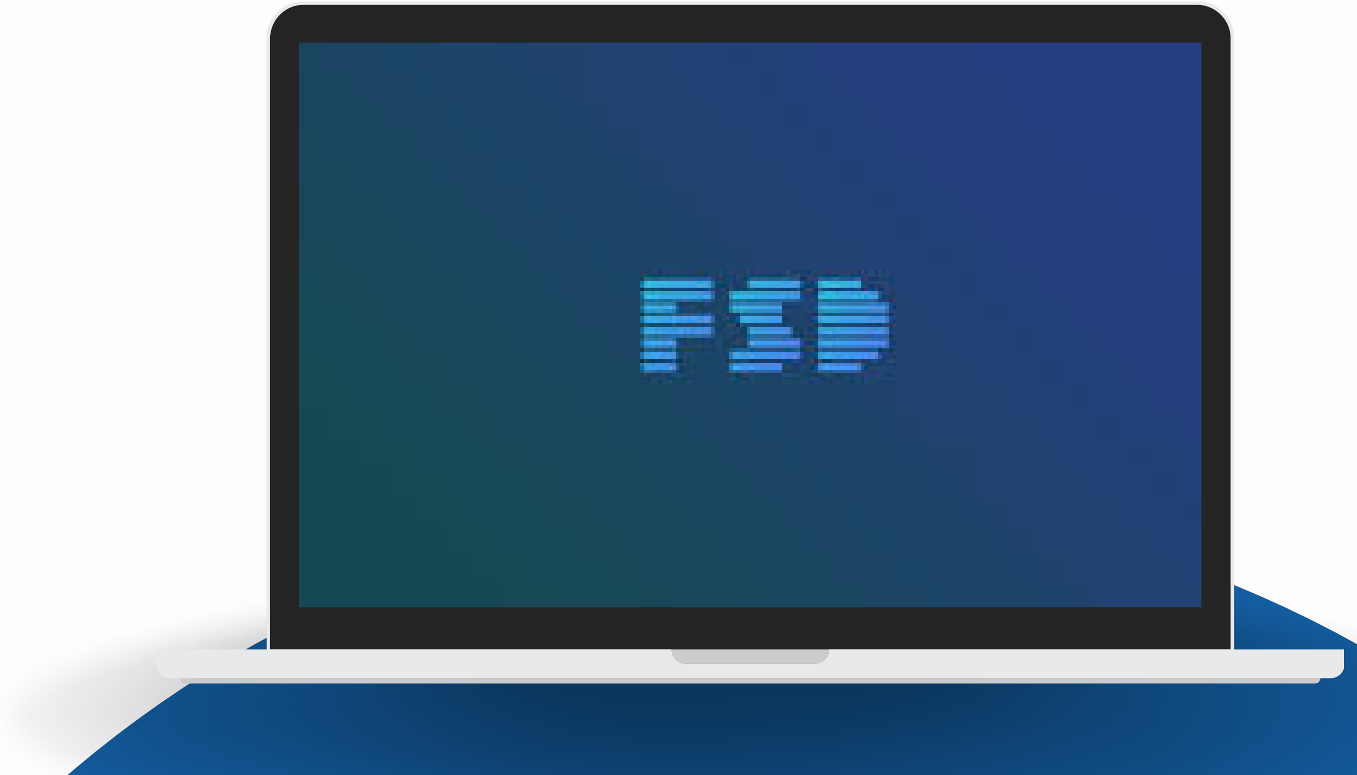
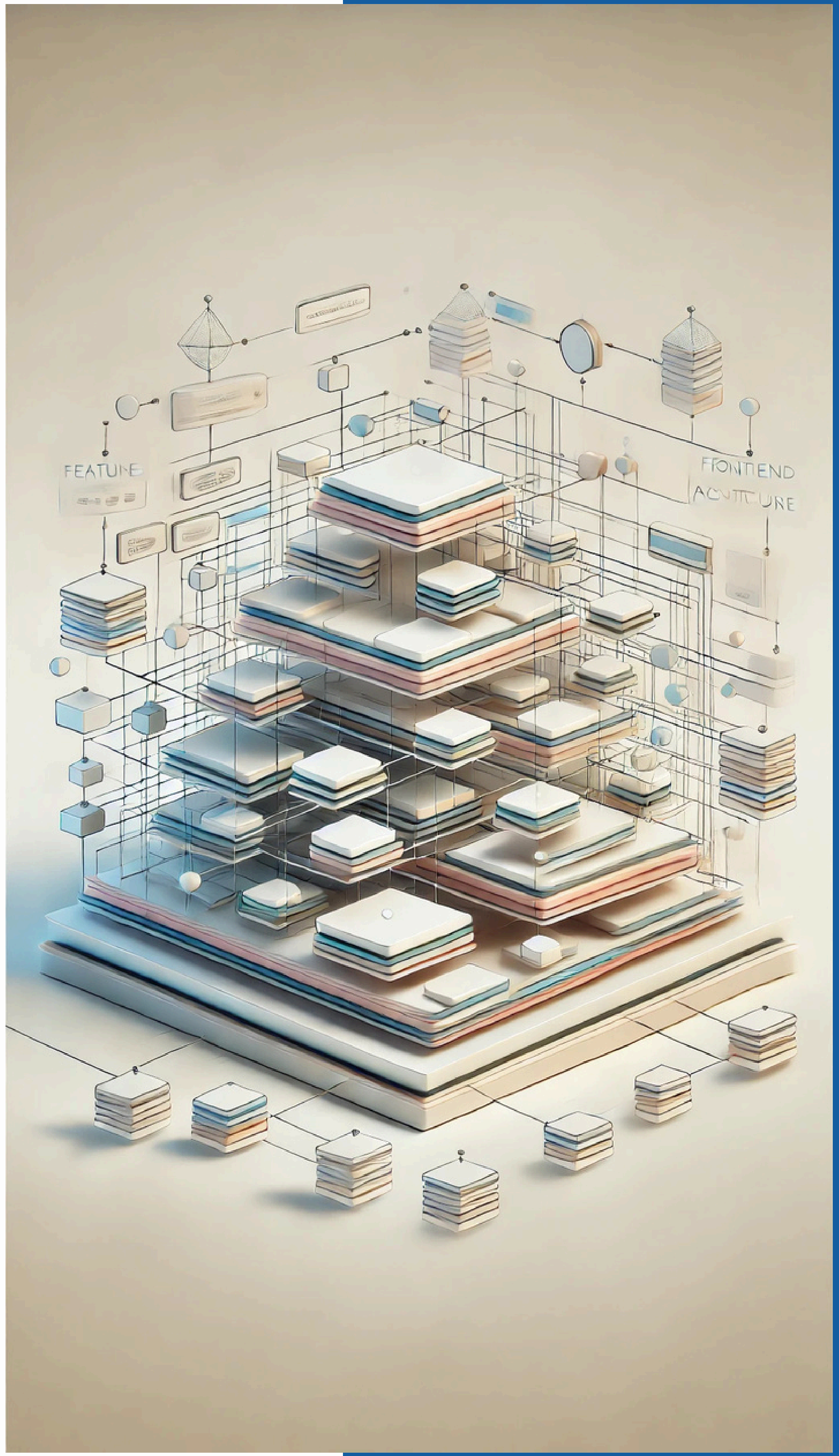
Preventing architectural debt with Feature-Sliced Design: a case for clean code

By: Aleksandr Guzenko



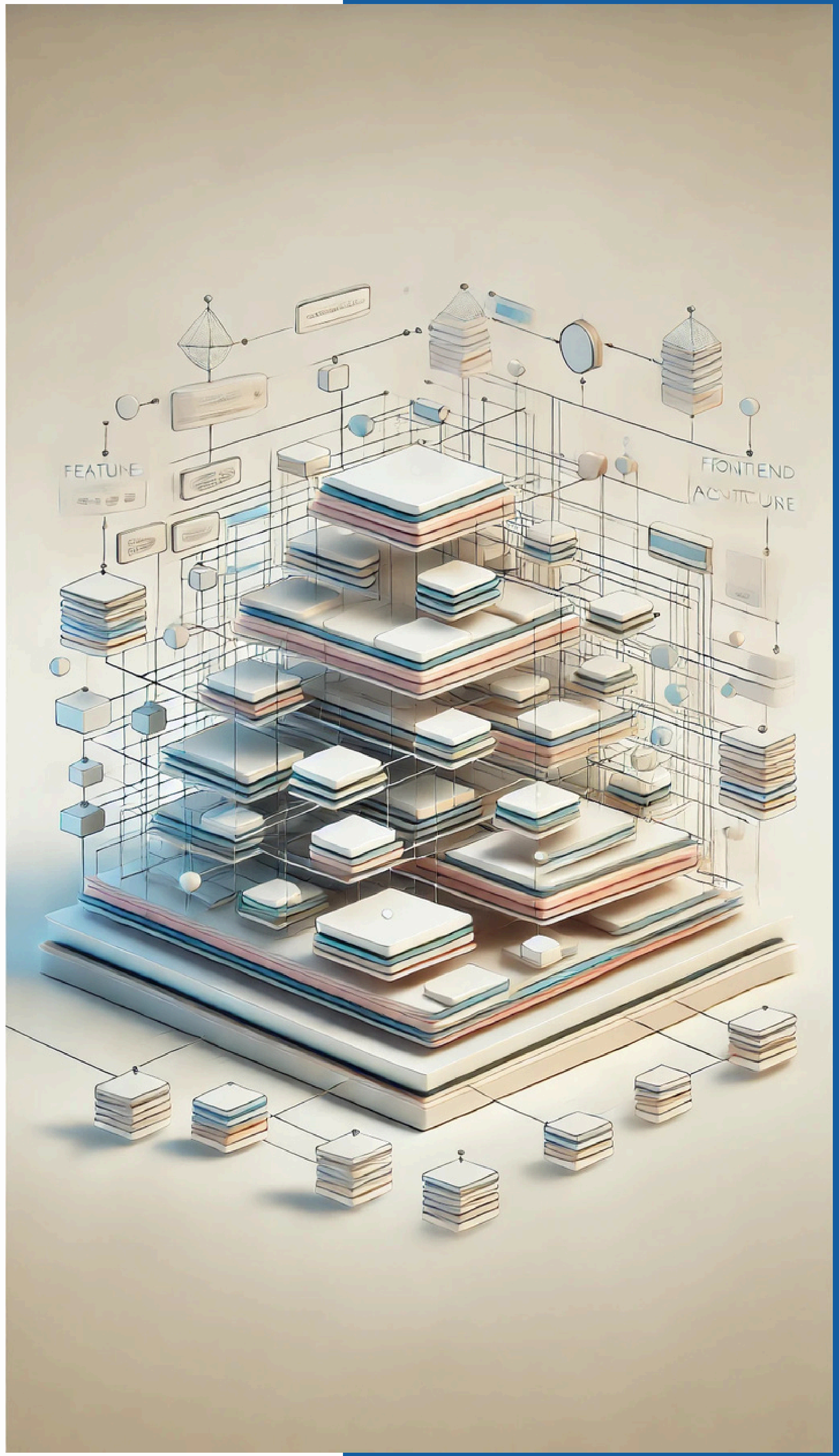
Overview

- Key qualities of a good app



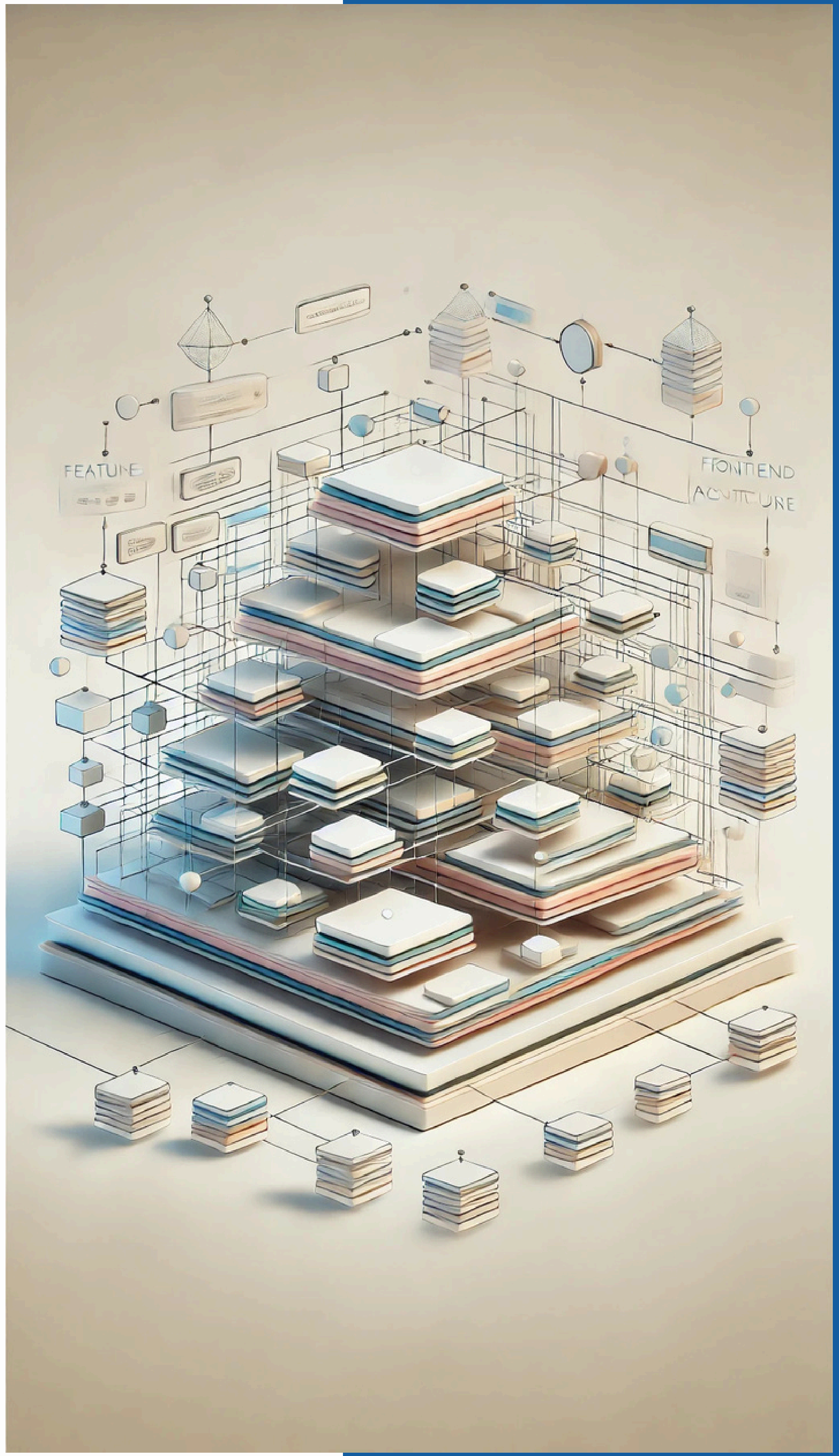
Overview

- Key qualities of a good app
- The main problems of modern frontend



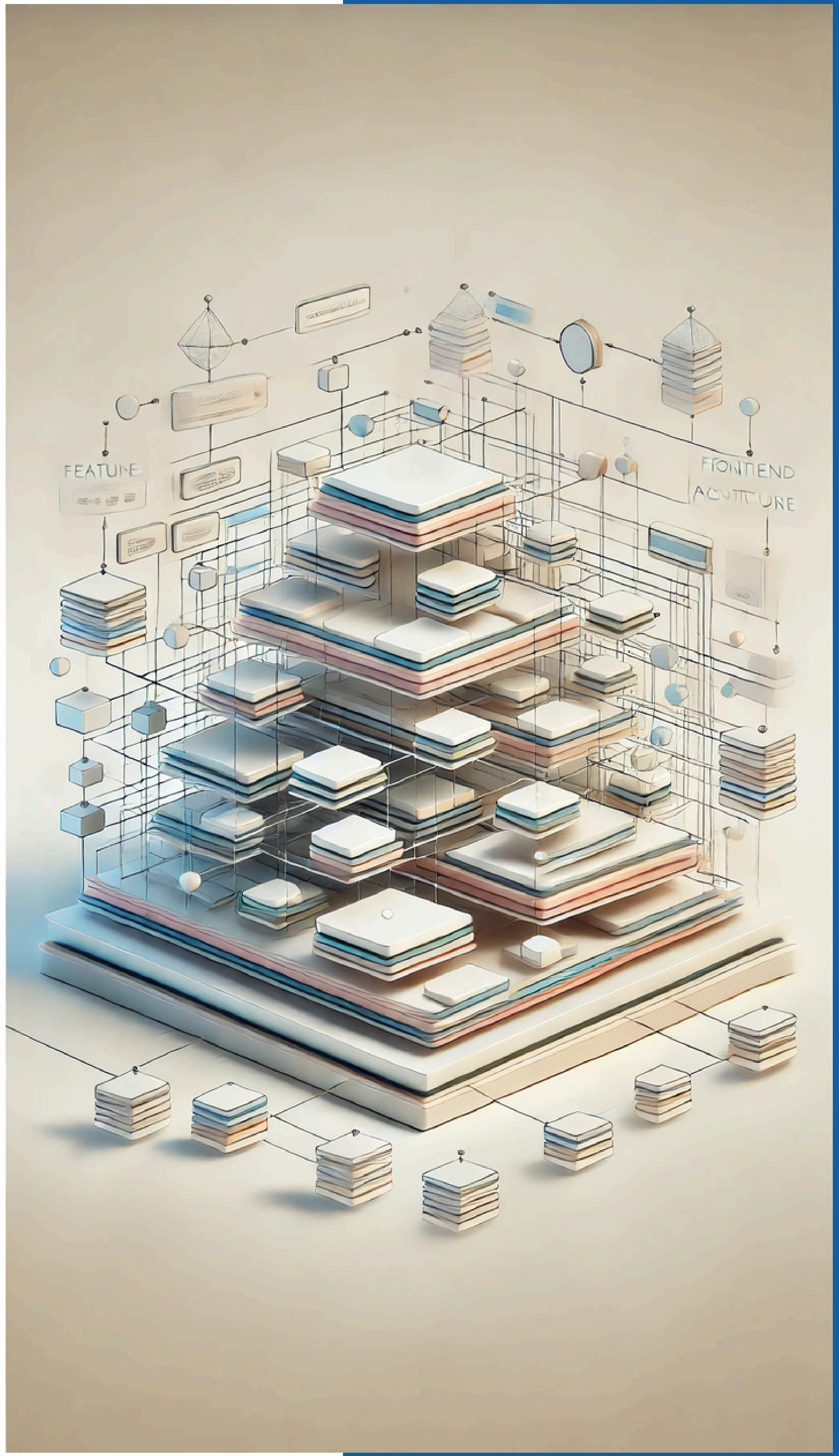
Overview

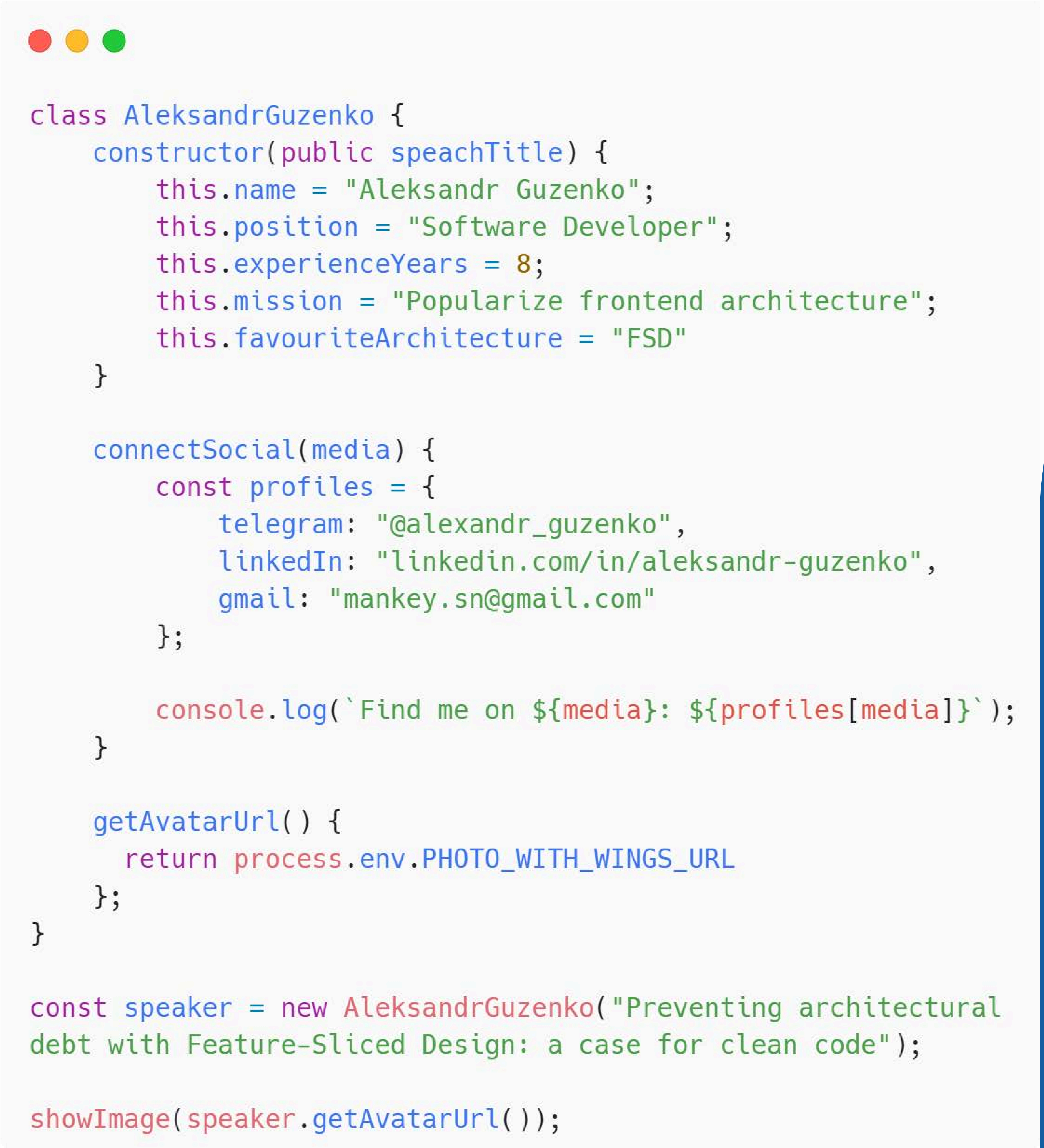
- Key qualities of a good app
- The main problems of modern frontend
- Ways to solve the above problems



Overview

- Key qualities of a good app
- The main problems of modern frontend
- Ways to solve the above problems
- FSD and how it can help you





```
class AleksandrGuzenko {
    constructor(public speachTitle) {
        this.name = "Aleksandr Guzenko";
        this.position = "Software Developer";
        this.experienceYears = 8;
        this.mission = "Popularize frontend architecture";
        this.favouriteArchitecture = "FSD"
    }

    connectSocial(media) {
        const profiles = {
            telegram: "@alexandr_guzenko",
            linkedIn: "linkedin.com/in/aleksandr-guzenko",
            gmail: "mankey.sn@gmail.com"
        };

        console.log(`Find me on ${media}: ${profiles[media]}`);
    }

    getAvatarUrl() {
      return process.env.PHOTO_WITH_WINGS_URL
    };
}

const speaker = new AleksandrGuzenko("Preventing architectural
debt with Feature-Sliced Design: a case for clean code");

showImage(speaker.getAvatarUrl());
```

Key qualities of a good app

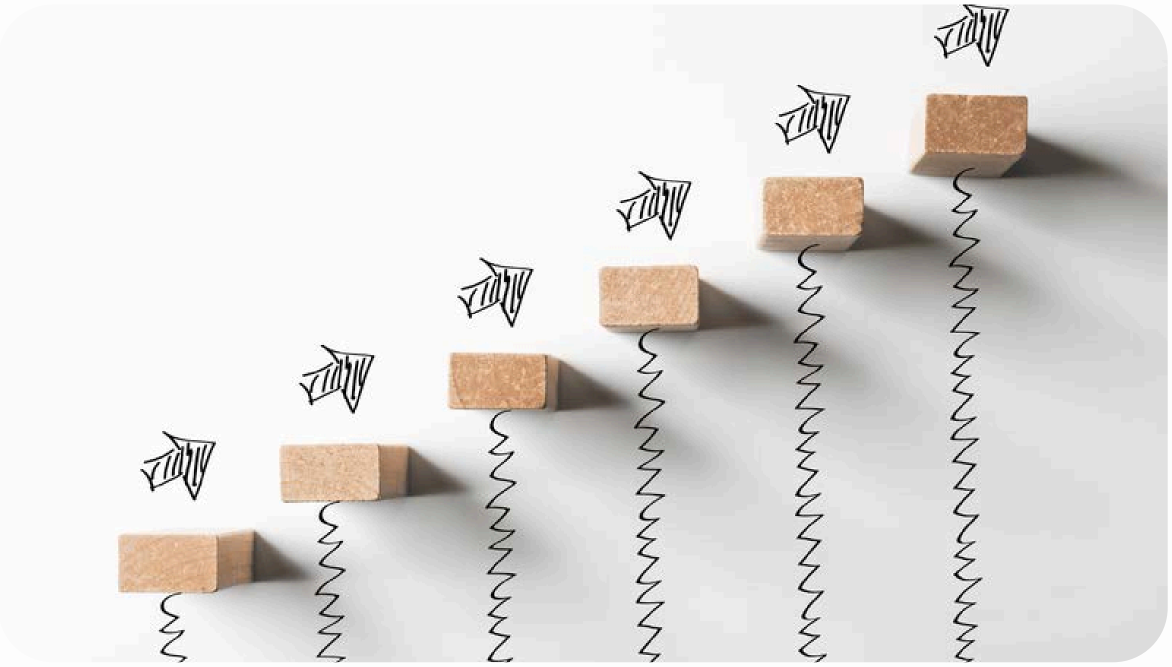
Business efficiency

How well does the app help a business achieve its goals

Key qualities of a good app

Business efficiency

How well does the app help a business achieve its goals



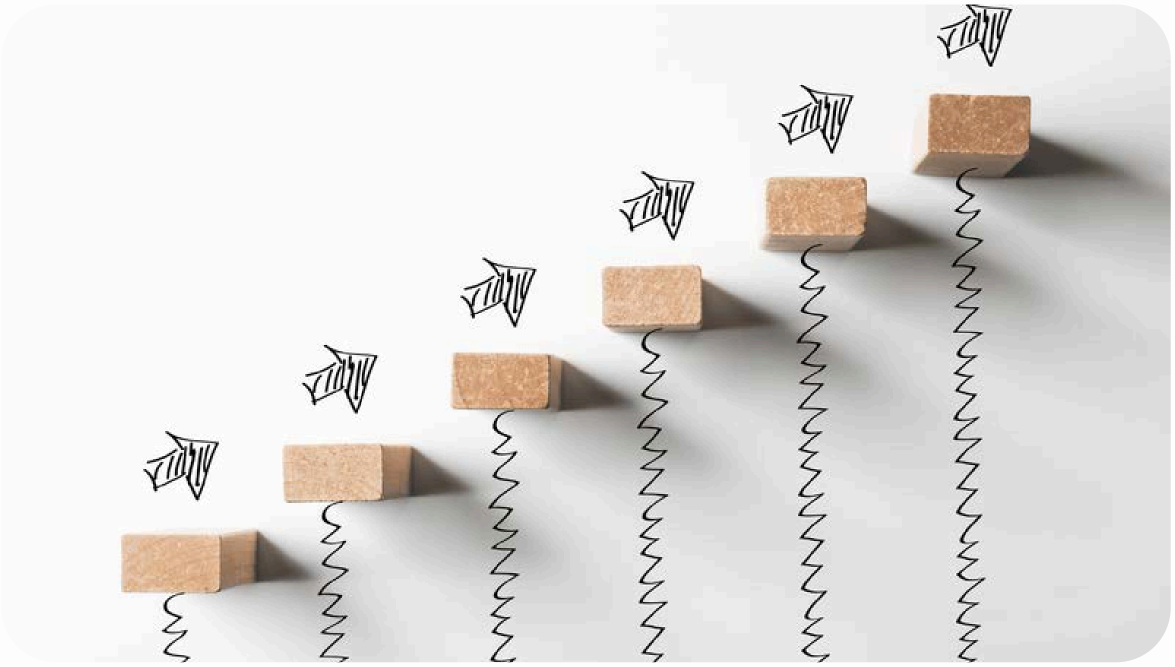
Scalability

How fast can you add a new feature

Key qualities of a good app

Business efficiency

How well does the app help a business achieve its goals



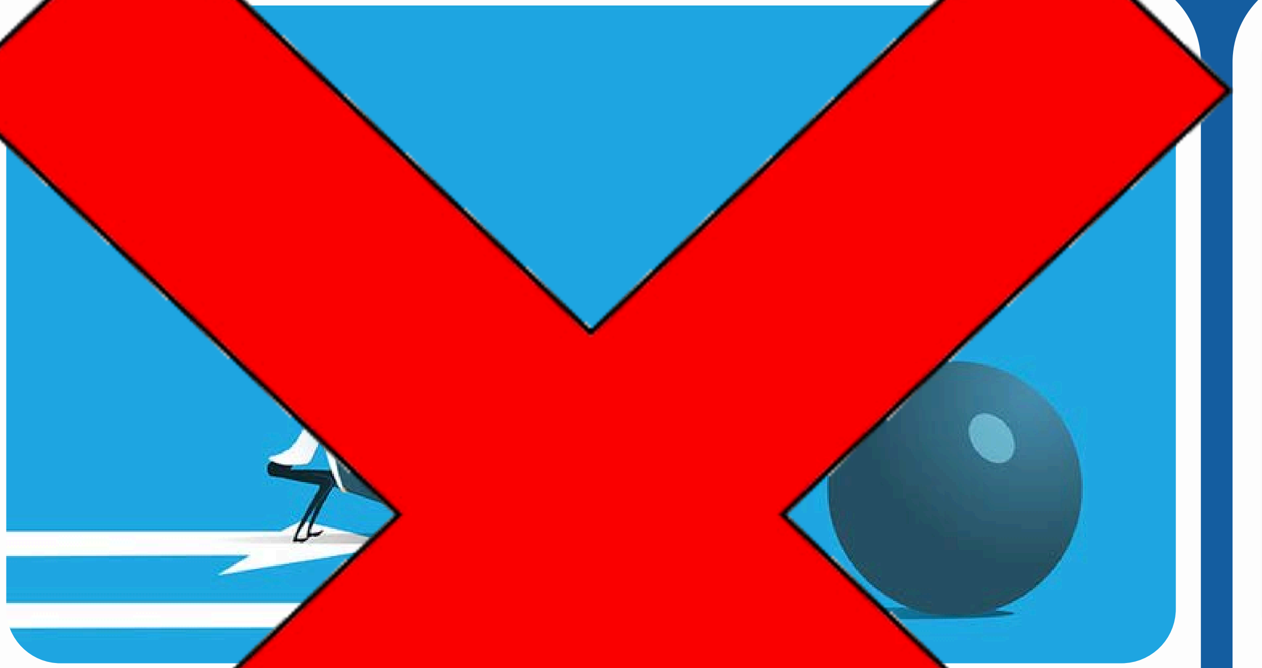
Scalability

How fast can you add a new feature

Maintainability

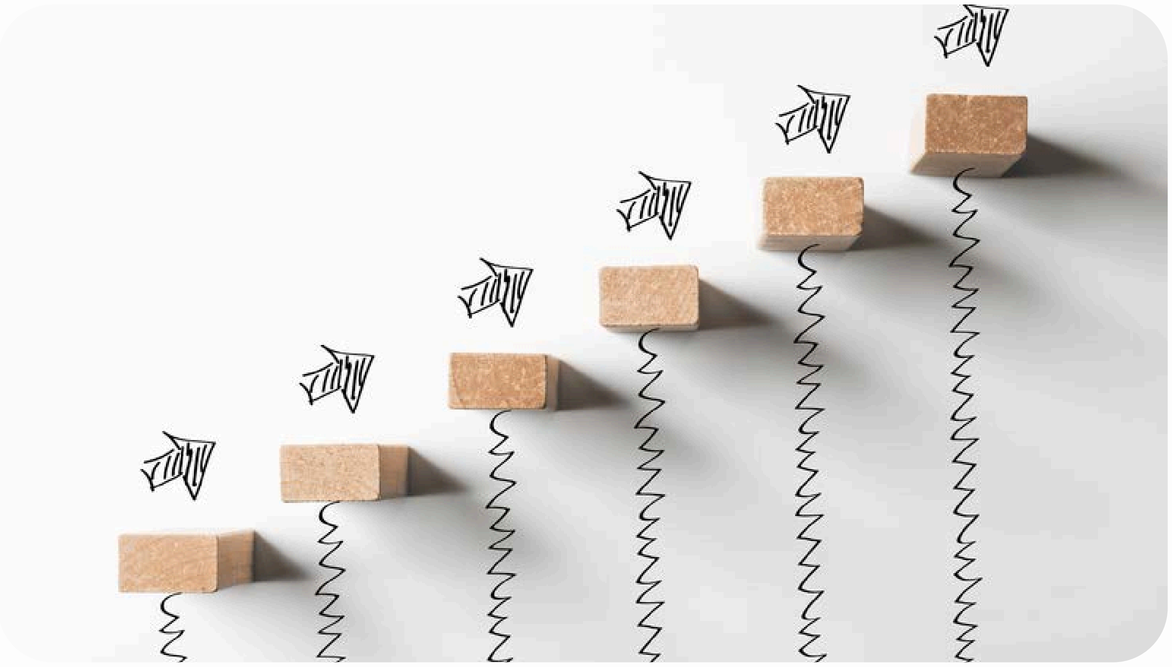
How easy it is to make changes and maintain your app in the long term

Key qualities of a good app



s e

es the app hel
achieve its goals



Scalability

How fast can you add a new feature

Maintainability

How easy it is to make changes and maintain your app in the long term

Why is this important?

When an application **scales** well, adding new functionality and increasing loads is much cheaper.

When an application is **well-maintainable**, the increase in development costs remains small, which allows for the implementation of large applications in the future.

Problems

One of the most difficult tasks is to determine the balance between scalability, maintainability and development speed.

Problems

One of the most difficult tasks is to determine the balance between scalability, maintainability and development speed.

Problem 1

unclear future at
the beggining

Problems

One of the most difficult tasks is to determine the balance between scalability, maintainability and development speed.

Problem 1

unclear future at the beggining

Problem 2

requirements may change too often

Problems

One of the most difficult tasks is to determine the balance between scalability, maintainability and development speed.

Problem 1

unclear future at the beggining

Problem 2

requirements may change too often

Problem 3

lack of time

EVERY APPLICATION IS UNIQUE

The best solution for one application does not guarantee that it will work for another.

Solutions

There may be different best solutions for every person, every project and every company, but here are the most universal ones:

Solutions

There may be different best solutions for every person, every project and every company, but here are the most universal ones:

Become a master in architechture

The longest and the hardest way

Solutions

There may be different best solutions for every person, every project and every company, but here are the most universal ones:

Become a master in architechture

The longest and the hardest way

Use the best-known practices

if your only tool is a hammer, every problem looks like a nail

Solutions

There may be different best solutions for every person, every project and every company, but here are the most universal ones:

Become a master in architechture

The longest and the hardest way

Use the best-known practices

if your only tool is a hammer, every problem looks like a nail

Use out-of-the-box solutions

The most efficient solution

Feature-Sliced Design

Architectural frontend methodology

What is FSD?

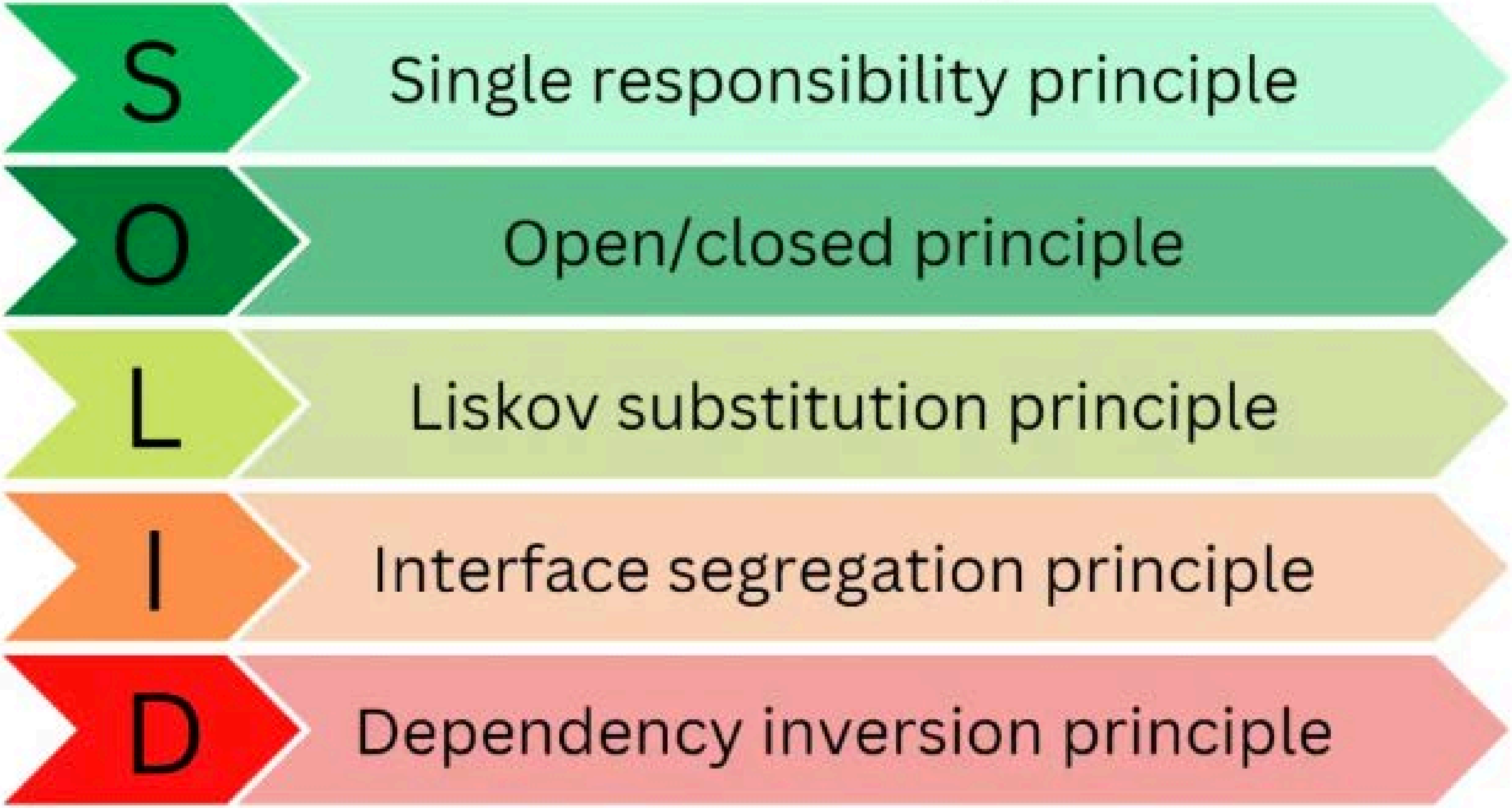
It is a compilation of rules and conventions on organizing code

What is FSD?

It is a compilation of rules and conventions on organizing code

01 SOLID

SOLID Principles



S Single responsibility principle

O Open/closed principle

L Liskov substitution principle

I Interface segregation principle

D Dependency inversion principle

What is FSD?

It is a compilation of rules and conventions on organizing code

01 SOLID

02 Public API

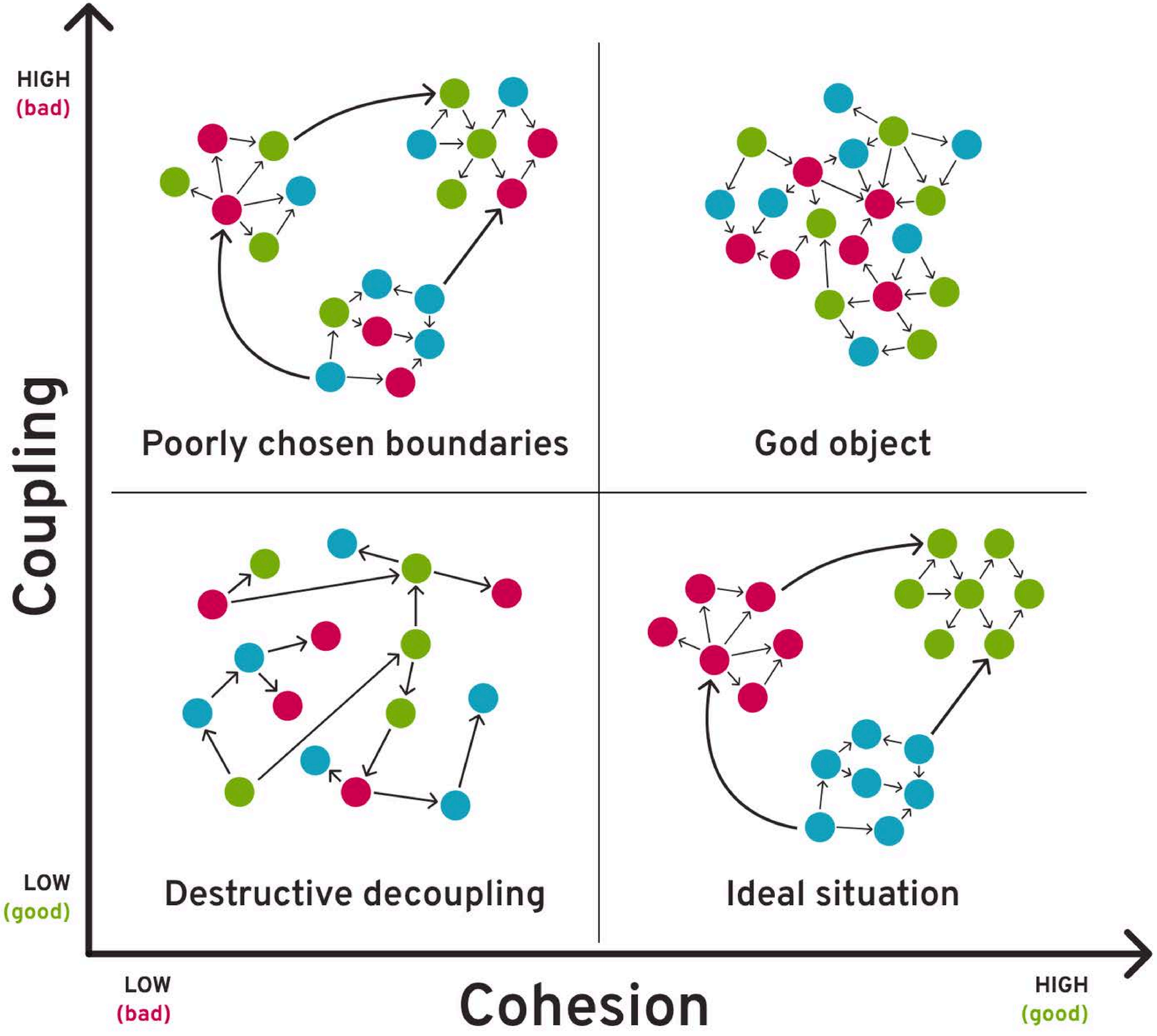
What is FSD?

It is a compilation of rules and conventions on organizing code

01 SOLID

02 Public API

03 Low Coupling & High Cohesion


HIGH
(bad)
Coupling
LOW
(good)
Poorly chosen boundaries
God object
Destructive decoupling
Ideal situation
LOW
(bad)
Cohesion
HIGH
(good)

What is FSD?

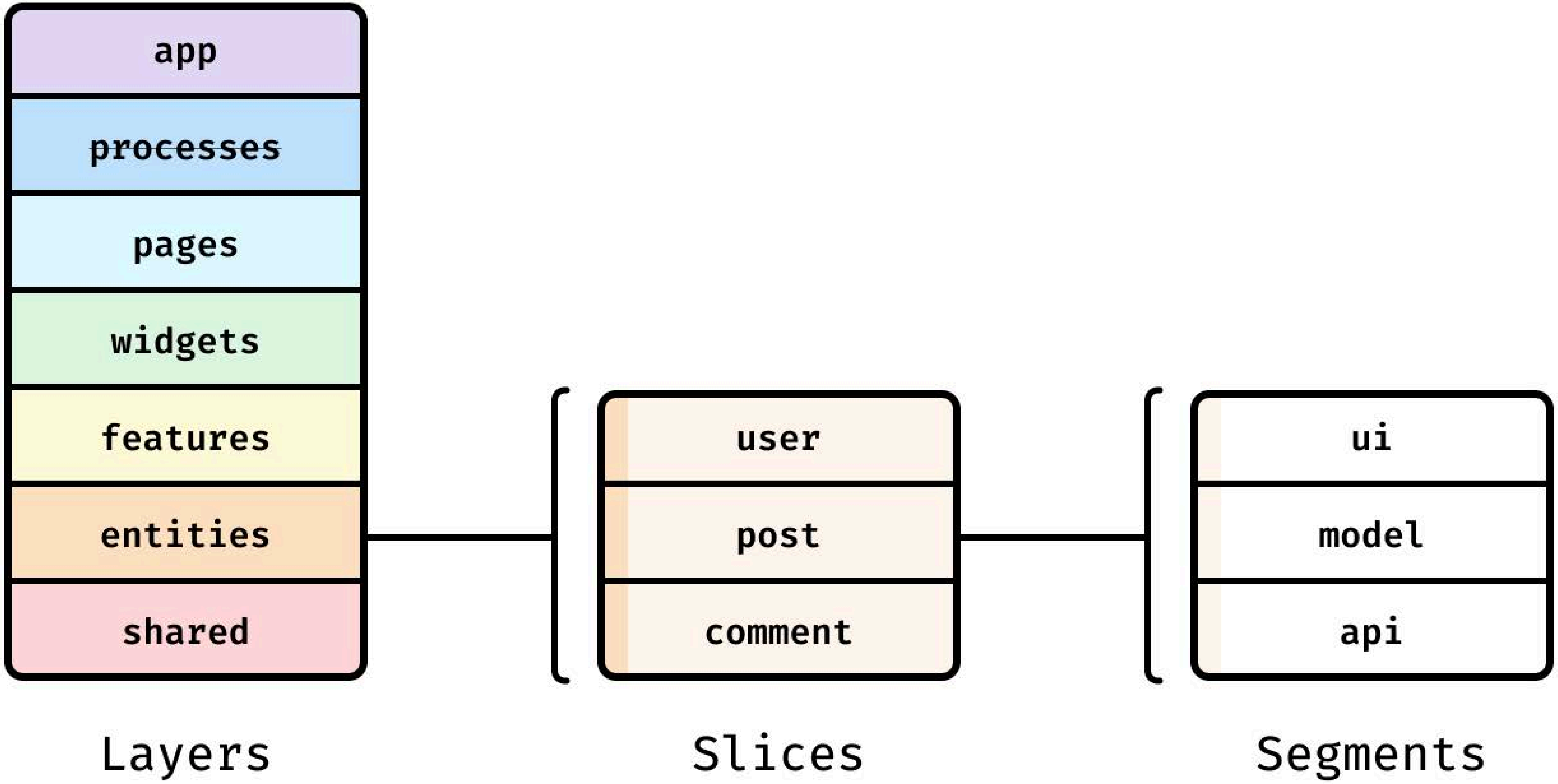
It is a compilation of rules and conventions on organizing code

01 SOLID

02 Public API

03 Low Coupling & High Cohesion

04 FSD own rules



app

~~processes~~

pages

widgets

features

entities

shared

Layers

user

post

comment

Slices

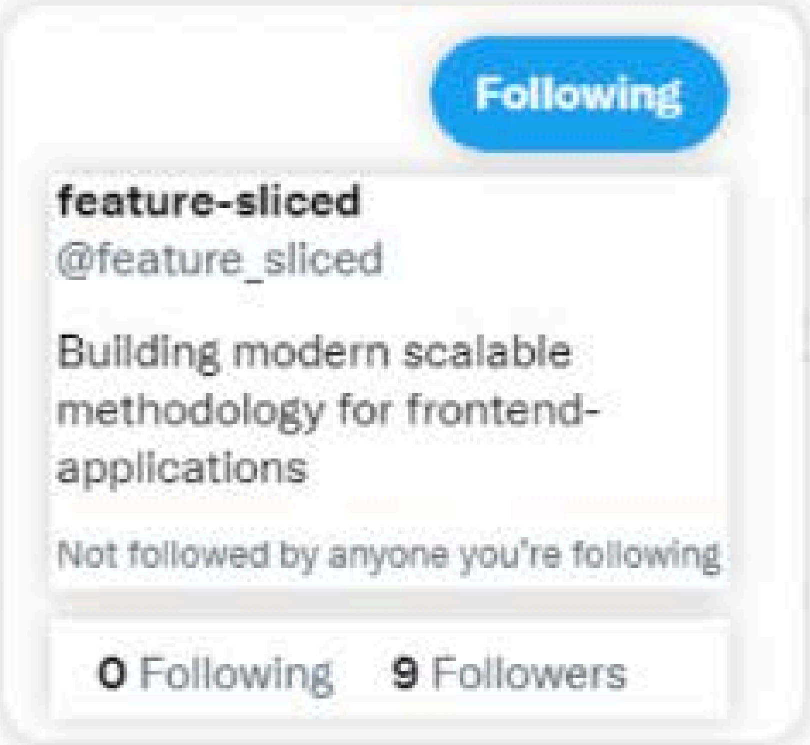
ui

model

api

Segments

SHARED



Following

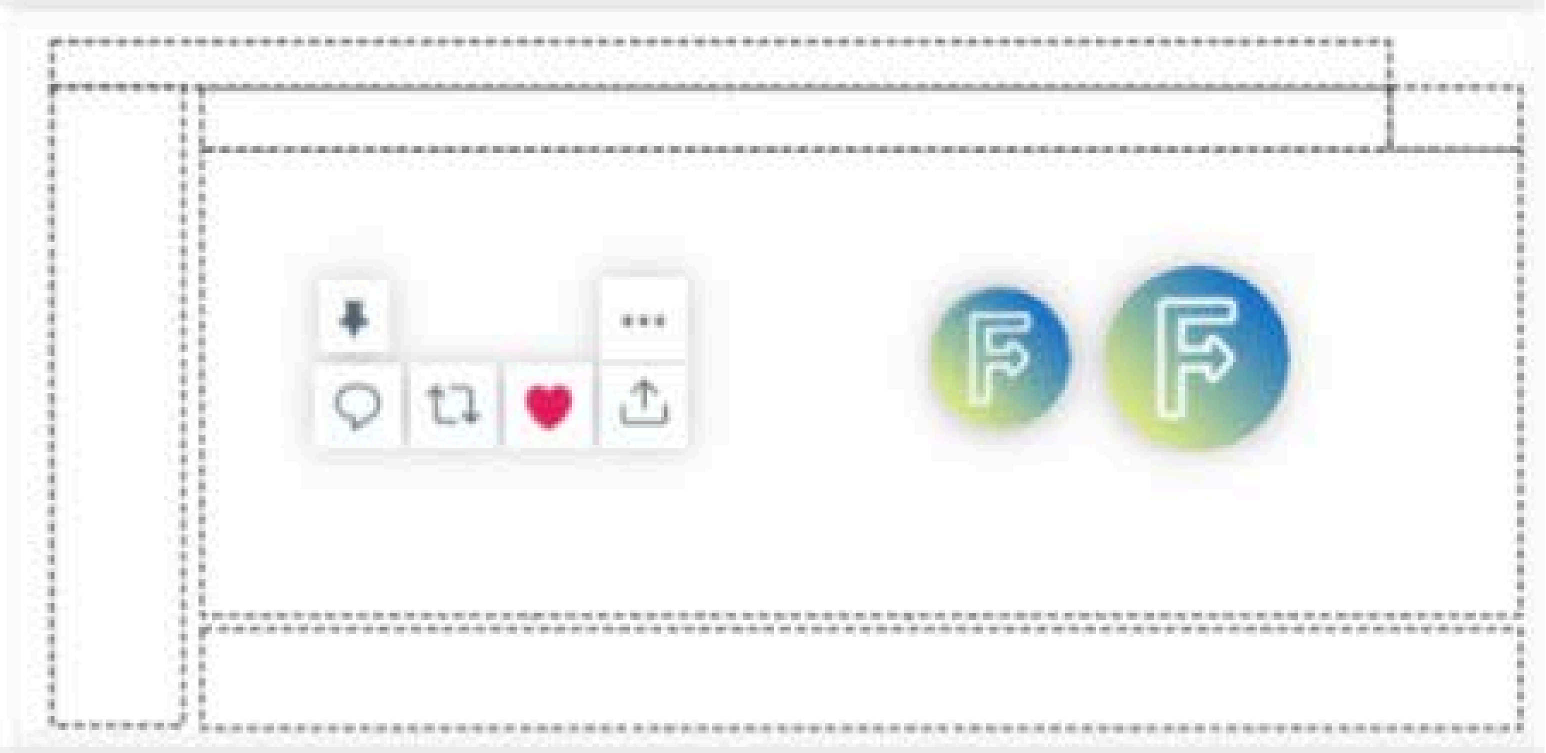
feature-sliced

@feature_sliced

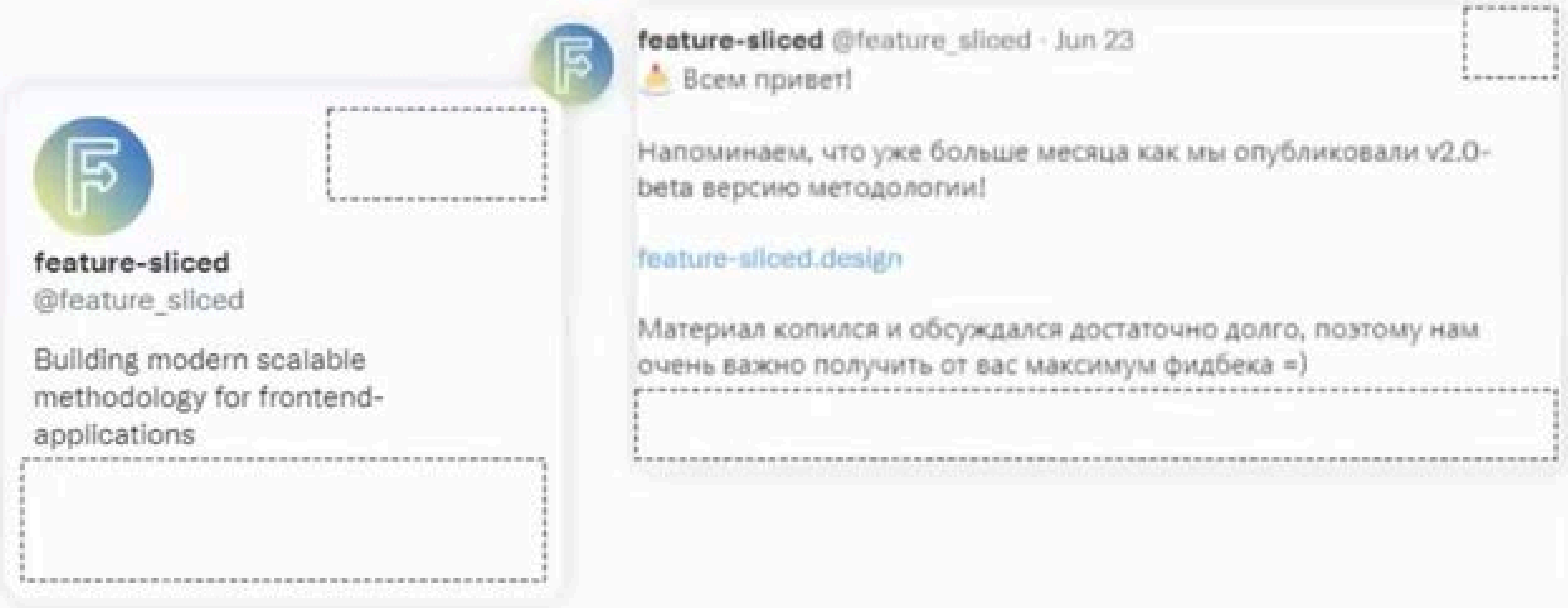
Building modern scalable methodology for frontend-applications

Not followed by anyone you're following

0 Following **9** Followers



ENTITIES



feature-sliced

@feature_sliced

Building modern scalable methodology for frontend-applications

feature-sliced @feature_sliced · Jun 23

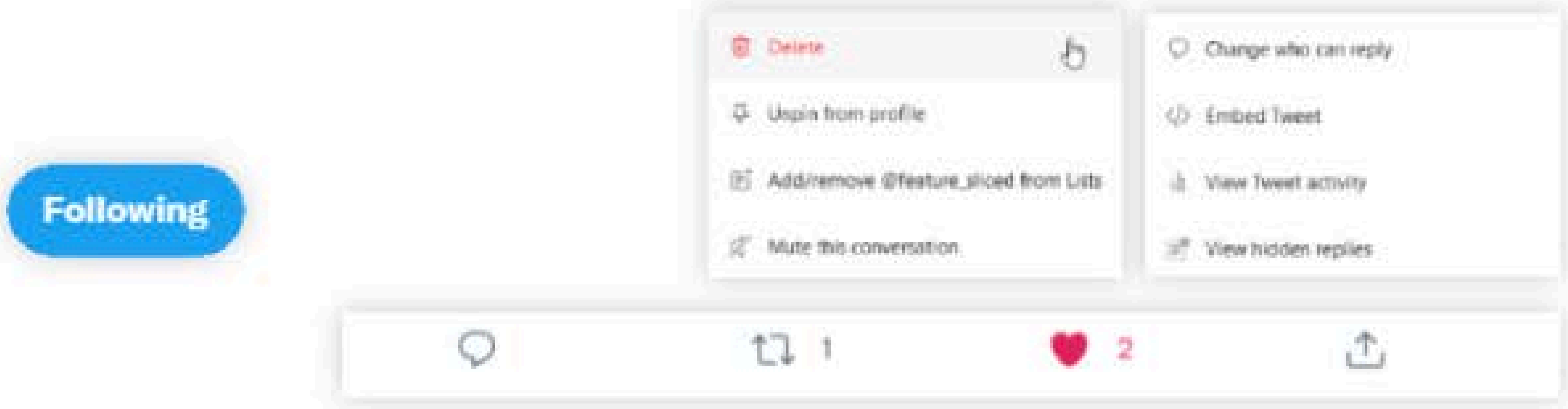
Всем привет!

Напоминаем, что уже больше месяца как мы опубликовали v2.0-beta версию методологии!

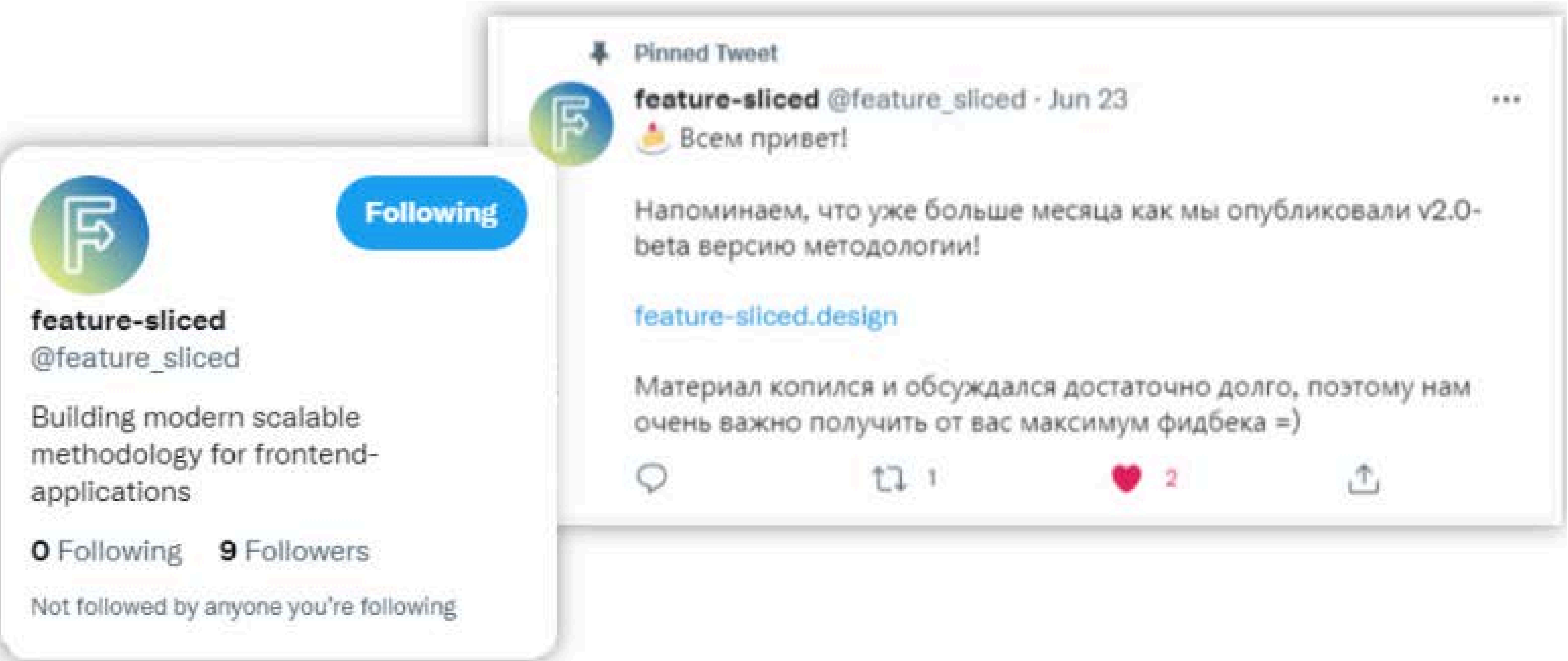
feature-sliced.design

Материал копился и обсуждался достаточно долго, поэтому нам очень важно получить от вас максимум фидбека =)

FEATURES


Following
Delete
Unpin from profile
Add/remove @feature_sliced from Lists
Mute this conversation
Change who can reply
Embed Tweet
View Tweet activity
View hidden replies
1
2

WIDGETS



feature-sliced
@feature_sliced

Building modern scalable methodology for frontend-applications

0 Following 9 Followers

Not followed by anyone you're following

Pinned Tweet

feature-sliced @feature_sliced · Jun 23

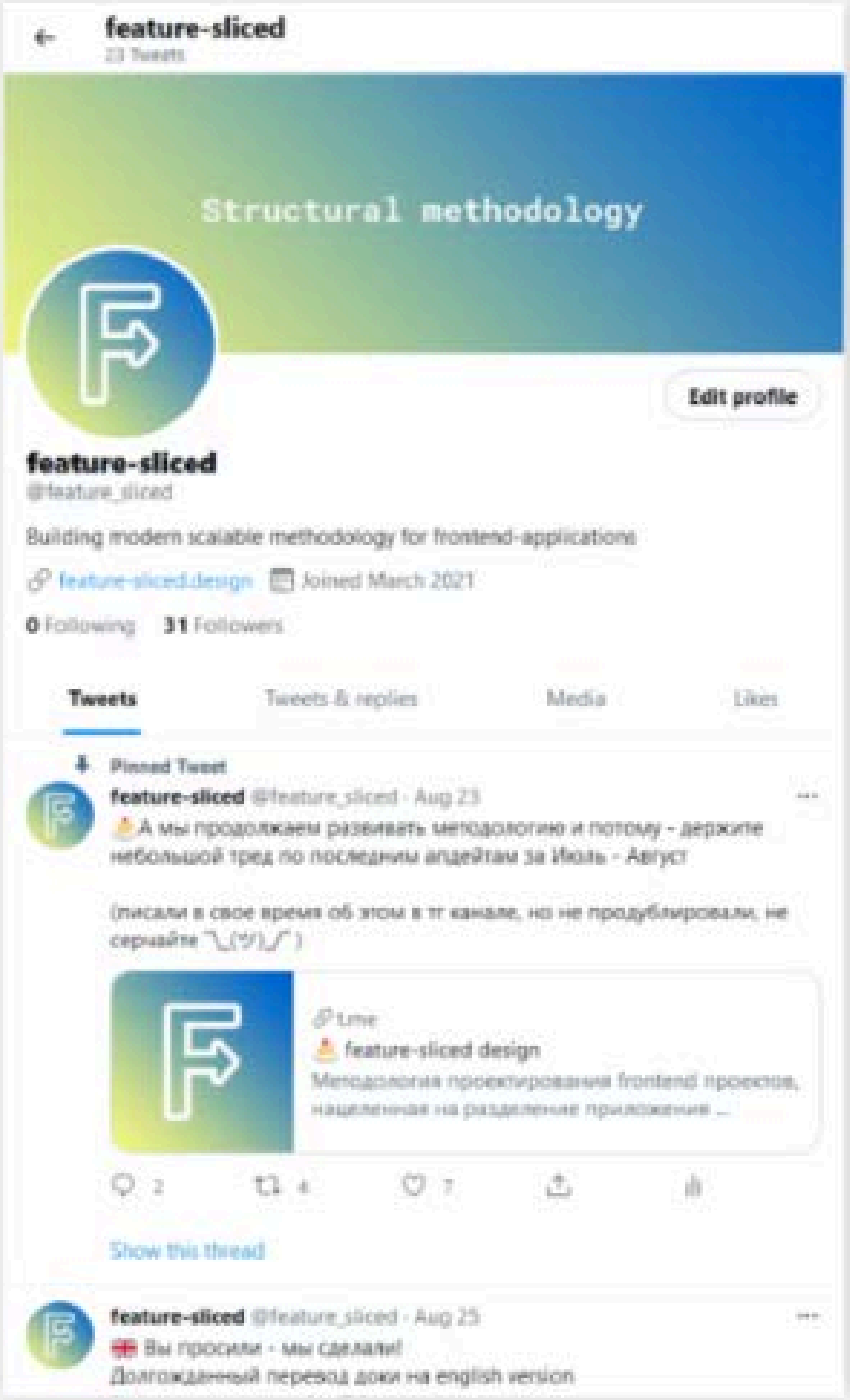
🍰 Всем привет!

Напоминаем, что уже больше месяца как мы опубликовали v2.0-beta версию методологии!

feature-sliced.design

Материал копился и обсуждался достаточно долго, поэтому нам очень важно получить от вас максимум фидбека =)

PAGES



feature-sliced
23 Tweets

Structural methodology

feature-sliced
@feature_sliced

Building modern scalable methodology for frontend-applications

feature-sliced.design Joined March 2021

0 Following **31** Followers

Pinned Tweet

feature-sliced @feature_sliced · Aug 23

🍰А мы продолжаем развивать методологию и потому - держите небольшой тред по последним апдейтам за Июль - Август

(писали в свое время об этом в тг канале, но не продублировали, не серчайте ¯_(ツ)_/¯)

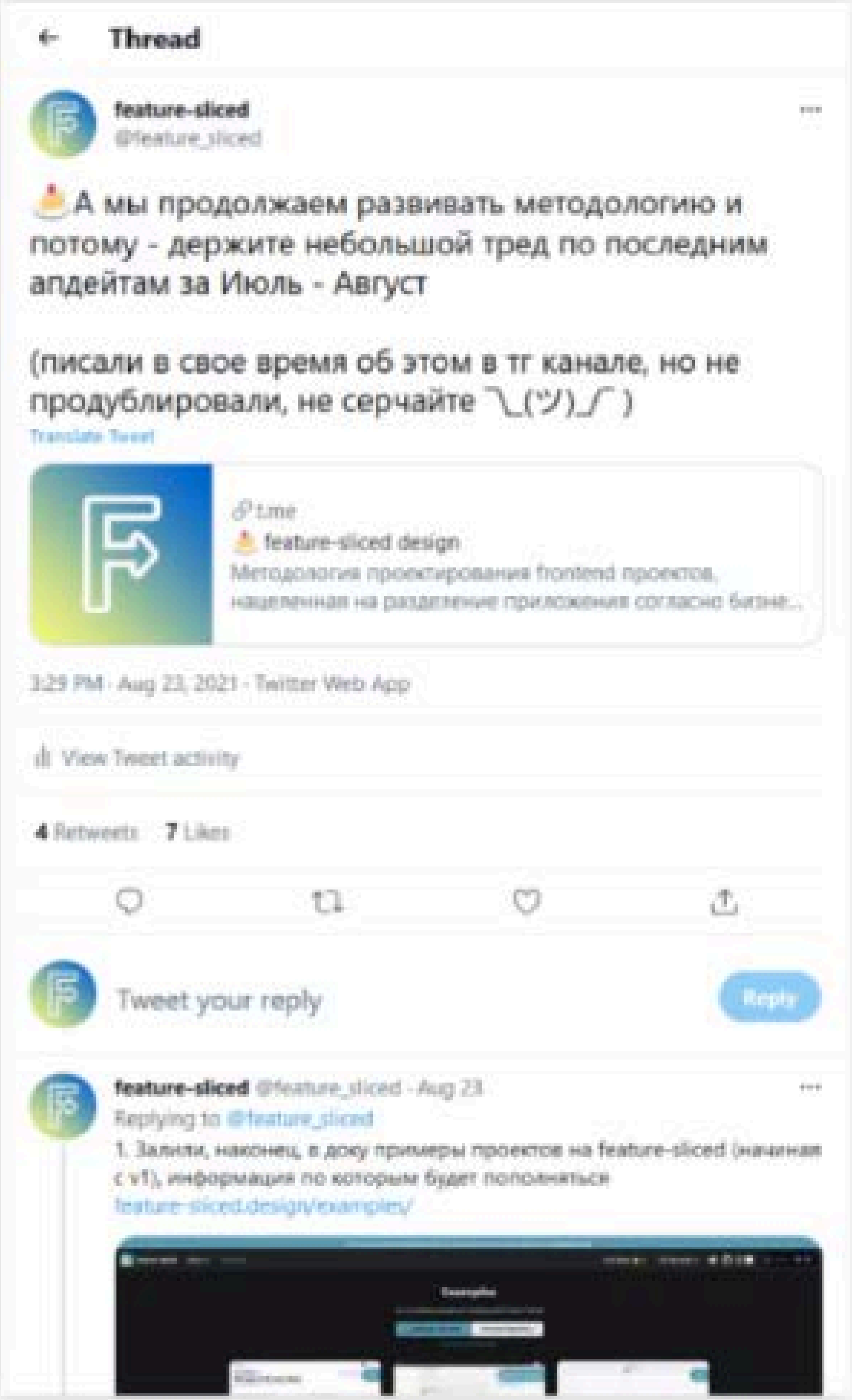
t.me
🍰 feature-sliced design
Методология проектирования frontend проектов, нацеленная на разделение приложения ...

2 4 7

Show this thread

feature-sliced @feature_sliced · Aug 25

🇬🇧 Вы просили - мы сделали!
Долгожданный перевод доки на english version



Thread

feature-sliced
@feature_sliced

🍰А мы продолжаем развивать методологию и потому - держите небольшой тред по последним апдейтам за Июль - Август

(писали в свое время об этом в тг канале, но не продублировали, не серчайте ¯_(ツ)_/¯)

Translate Tweet

t.me
🍰 feature-sliced design
Методология проектирования frontend проектов, нацеленная на разделение приложения согласно бизне...

3:29 PM · Aug 23, 2021 · Twitter Web App

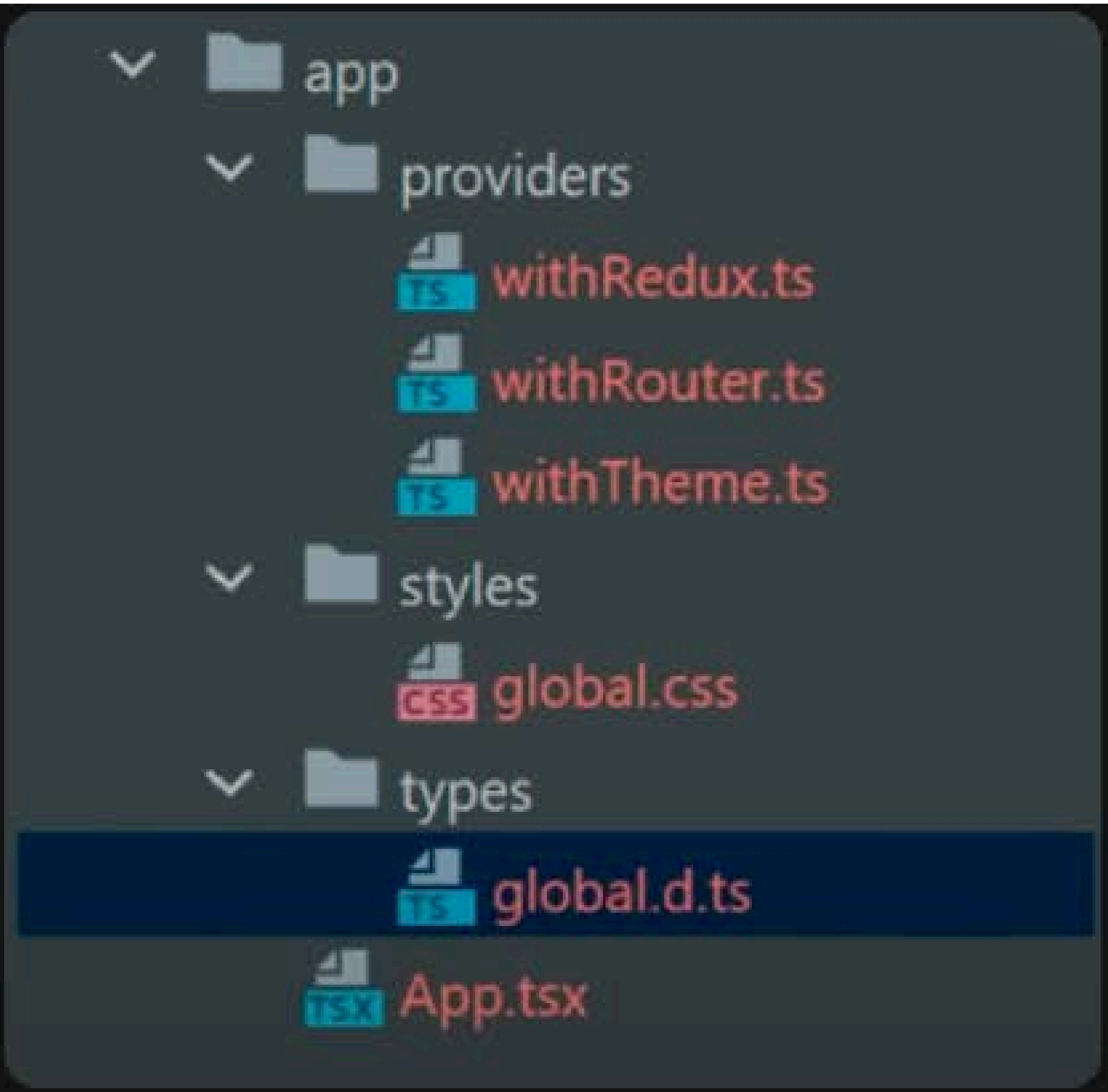
View Tweet activity

4 Retweets **7** Likes

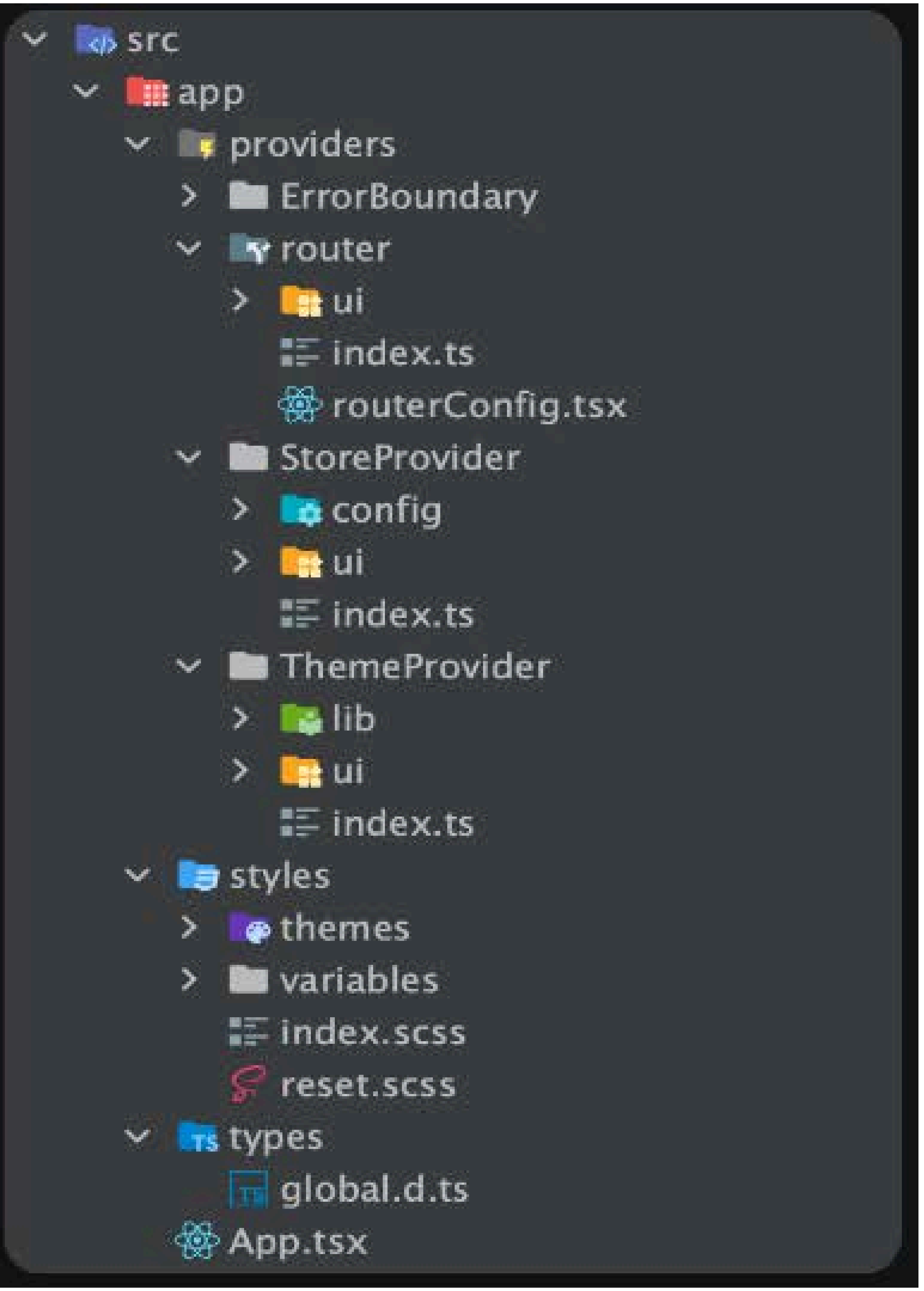
Tweet your reply

feature-sliced @feature_sliced · Aug 23
Replying to @feature_sliced
1. Залили, наконец, в доку примеры проектов на feature-sliced (начиная с v1), информация по которым будет пополняться
feature-sliced.design/examples/

APP



```
app
  providers
    withRedux.ts
    withRouter.ts
    withTheme.ts
  styles
    global.css
  types
    global.d.ts
  App.tsx
```



```
src
  app
    providers
      ErrorBoundary
      router
        ui
        index.ts
        routerConfig.tsx
      StoreProvider
        config
        ui
        index.ts
      ThemeProvider
        lib
        ui
        index.ts
    styles
      themes
      variables
      index.scss
      reset.scss
    types
      global.d.ts
    App.tsx
```

Decomposition cheatsheet

Choosing a layer

Shared

Code that is not specific to your application, code that serves as a foundation.

Self-check question

Can this code be used in a pizza shop app or an online bank?

Example: a dropdown menu can appear in a pizza shop app, a social media post probably can't.

Entities

Code that represents a real-life concept that your app is working with.

Self-check question

When describing your app, does this word appear as a subject or an object? Do your users/clients understand that word?

Example: users can write posts. Clients want to be able to add videos to their posts.

Features

Interactions that provide real-life value to your app's users, the things people want to do with your entities.

Self-check question

When describing to a stranger what your app does, do you mention these actions?

Example: users can write and edit posts. Posts can be configured to auto-delete after 5 minutes.

Widgets

Code that combines the layers below to form meaningful blocks, interactive and complete with data.

Self-check question

When looking at your app's UI from a distance, does this stand out as a complete "block"?

Example: A list of posts with pagination and the header appear as standalone blocks.

Pages

Entire screens of your application, built mostly by combining the layers below. Similar to widgets, but on a larger scale.

Self-check question

Is this code ready to be plugged into the router and work for users as is?

Example: the home page of an online shop with login, fresh deals, categories, search, etc.

App

Infrastructural code that makes your app actually work.

Self-check question

Is this something your framework or technical stack needs for your app to function?

Example: an i18n provider and a router make the app work and display sensible text to the user.

How FSD can help me to achieve maintainability and scalability?

How FSD can help me to achieve maintainability and scalability?

01

Good combination of time tested architectural patterns and principles with it’s own rules to solve modern frontend problems

How FSD can help me to achieve maintainability and scalability?

01

Good combination of time tested architectural patterns and principles with it’s own rules to solve modern frontend problems

02

A big community and a website may help to onboard faster

How FSD can help me to achieve maintainability and scalability?

01 Good combination of time tested architectural patterns and principles with it’s own rules to solve modern frontend problems

02 A big community and a website may help to onboard faster

03 This is the embodiment of the experience of many experienced developers.

How FSD can help me to achieve maintainability and scalability?

01 Good combination of time tested architectural patterns and principles with it’s own rules to solve modern frontend problems

02 A big community and a website may help to onboard faster

03 This is the embodiment of the experience of many experienced developers.

04 Suitable for projects of almost any size

How FSD can help me to achieve maintainability and scalability?

01 Good combination of time tested architectural patterns and principles with it’s own rules to solve modern frontend problems

02 A big community and a website may help to onboard faster

03 This is the embodiment of the experience of many experienced developers.

04 Suitable for projects of almost any size

05 You don’t need to reinvent new solutions every time

THANK YOU!

Aleksandr Guzenko

Software Engineer

mankey.sn@gmail.com

linkedin.com/in/aleksandr-guzenko/

FSD website:

feature-sliced.design